KNIME

İLE UÇTAN UCA VERİ BİLİMİ

Şadi Evren ŞEKER

Demet ERDOĞAN

Table of Contents

ÖNSÖZ

Bu kitabın amacı, ülkemizde veri bilimine giriş yapmayı planlayan arkadaşlara yol gösterici bir eser çıkarmaktır. Veri bilimi ile ilgili çok sayıda araç olmasına karşılık, Knime'ı seçmemizin özel hiçbir sebebi yoktur. Kitap ve aslında anlatılan konular veri bilimi yaklaşımına temel bir başlangıç teşkil etmekte olup, herhangi başka bir araç üzerinden de ilerlenebilir (örneğin Rapid Miner, SPSS Modeler veya Weka gibi bir araç çok kolay kullanılabilir).

Kitabın genel yaklaşımı, verinin kaynaktan alınması ve anlamlı bir bilgi üretilmesine kadar geçen süreci bir yolculuk olarak ele almak ve bu yolu önce adım adım sonrasında ise baştan sona sizlere aktarmaktır. Bu amaçla, verinin kaynaktan işlendiği adımlarla kitaba başlanmış ve sonunda da anlamlı uçtan uca örneklerle kitap devam etmiştir. Kitabın, kullandığı bütün modeller ve kitabın video olarak anlatımı da ayrıca udemy üzerinde bulunabilir, burada derslerde anlatılan içeriğe birebir paralel olarak kitap içeriği hazırlanmış, bazı cümlelerde ve örneklerde düzenlemeler yapılmış ancak genel olarak içeriğe bağlı kalınmıştır.

Genel olarak veri bilimine başlayacak birisinin, hiçbir ön eğitimi olmasa bile bu konularla başlamasını faydalı görüyorum, çünkü makine öğrenmesi veya istatistiksel modellerin nerede ve nasıl kullanıldığını öğrenmek, çok daha motive edici ve bilgileri doğru yere koyan bir yaklaşım getiriyor. Dolayısıyla bu alana yeni giren arkadaşlara, aramıza hoş geldiniz diyorum.

Kitabın yazımı sırasında kitabın yazımına büyük bir gayretle başlayan ve emeği geçen Berna Taş'a ve kitabın yazımı için aylarca uğraşarak gece gündüz demeden aralıksız çalışan Demet Erdoğan'a burada emeklerinden dolayı bir kere daha teşekkür ederim.

1.BÖLÜM: GİRİŞ VE KURULUM

1.1 Giriş

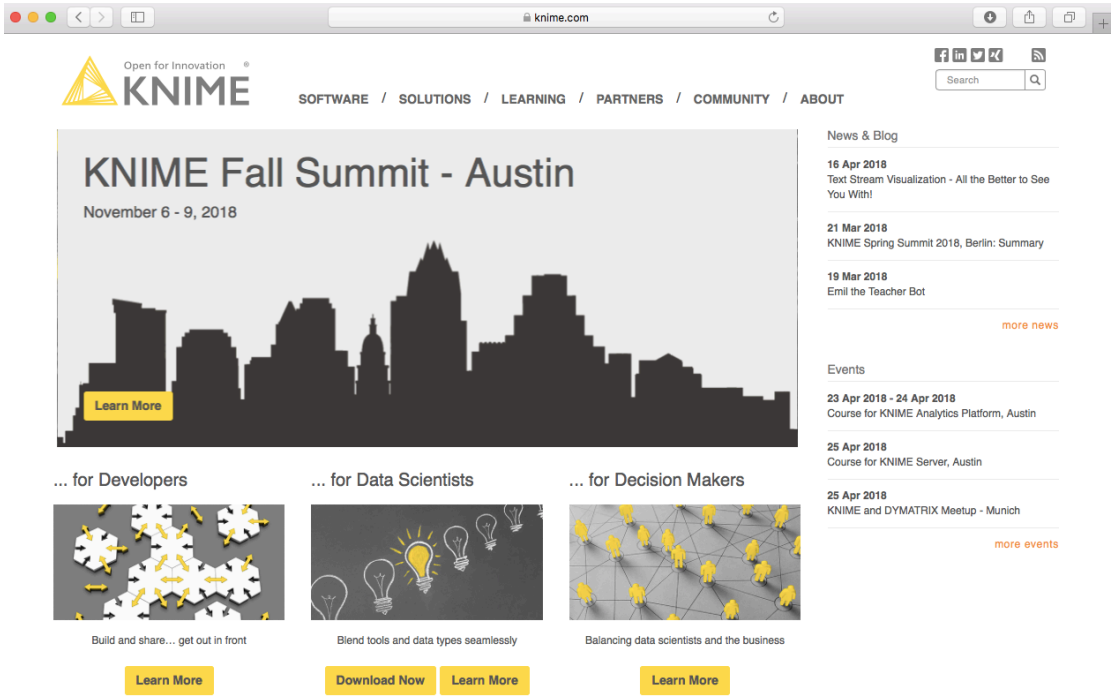
Bu bölümde Knıme giriş yapılacaktır. Bu kitaptaki amaç veri bilimine giriş yapmak ve veri biliminin ne olduğunu, buna bağlı olarak diğer disiplinleri makine öğrenmesi, yapay zeka ve veri tabanı gibi kavramların üzerinden geçerek ve yer yer veri ambarı (DataWareHouse) kavramına da değinmektir. Aslında uçtan uca veriyi alıp değerli, kazanç elde edilebilecek , bir amaca hizmet edebilecek hale getirmektir. Buna KDD denilmektedir. İlerleyen bölümlerde bu kavrama daha detaylıca değinilecektir.

Kullanacak yazılım bahsedildiği üzere KNIME. Açık kaynak kodlu bir yazılım ve kullanılacak dokümantasyonda Knime'ın dokümantasyonuna yakın bir yapıda ilerleyecektir. Bu kitabın amaçlarından bir de veri bilimcisi yetiştirebilmektir. Dolayısıyla sıfırdan veri bilimine giriş yapan ve bu konudaki kavramları öğrenen kişiler kazandırmak amaçlardan biridir. Veri bilimcisi olabilmek için istatistik kökenli olunması ya da bilgisayar mühendisliği/ bilgisayar bilimleri temelli bir kökenin olması bir avantajdır ama şarj değildir. Temel olarak bilgisayar kullanmayı biliyor olmak bu kitaptaki konular için yeterlidir.

Knime'ın analitik kısmını kullanılacaktır. Knime üzerinde kullanılacak bazı eklentiler açık kaynak kodlu ve bazıları ücretsizken bazıları ücretli ve başka amaçlara hizmet eden eklentiler olabilir.

1.2 Knime İndirilmesi ve Kurulumu

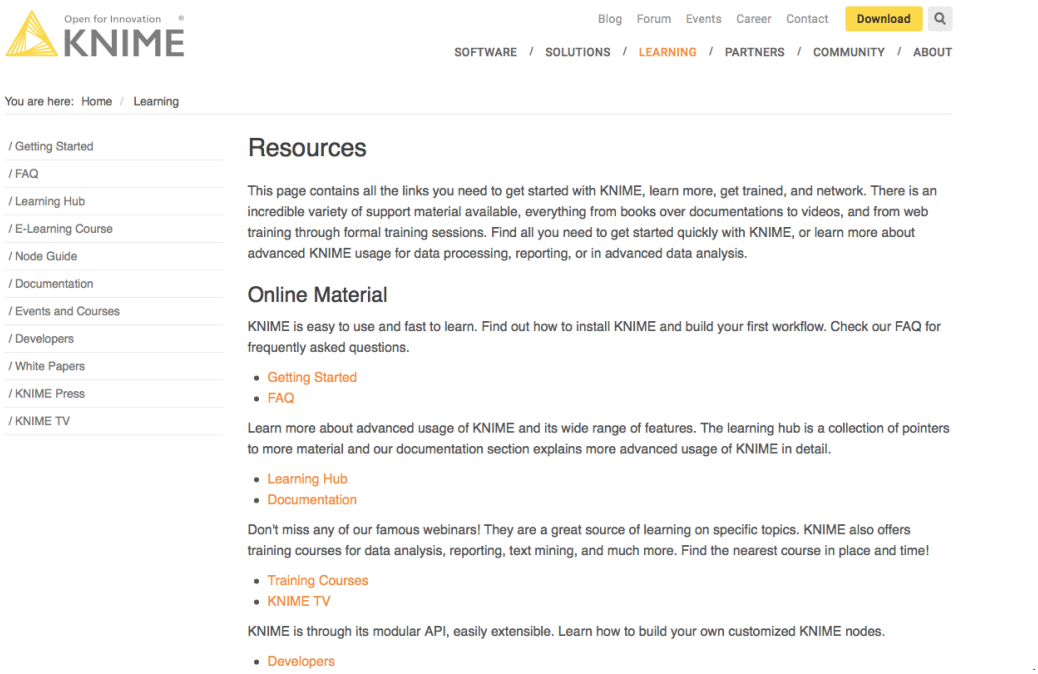
Bu bölümde Knime'ın nasıl indirileceği ve bilgisayara kurulumu açıklanacaktır. Knime.com web sitesinden software sekmesine girerek indirilebilir.



Şekil 1.2.1

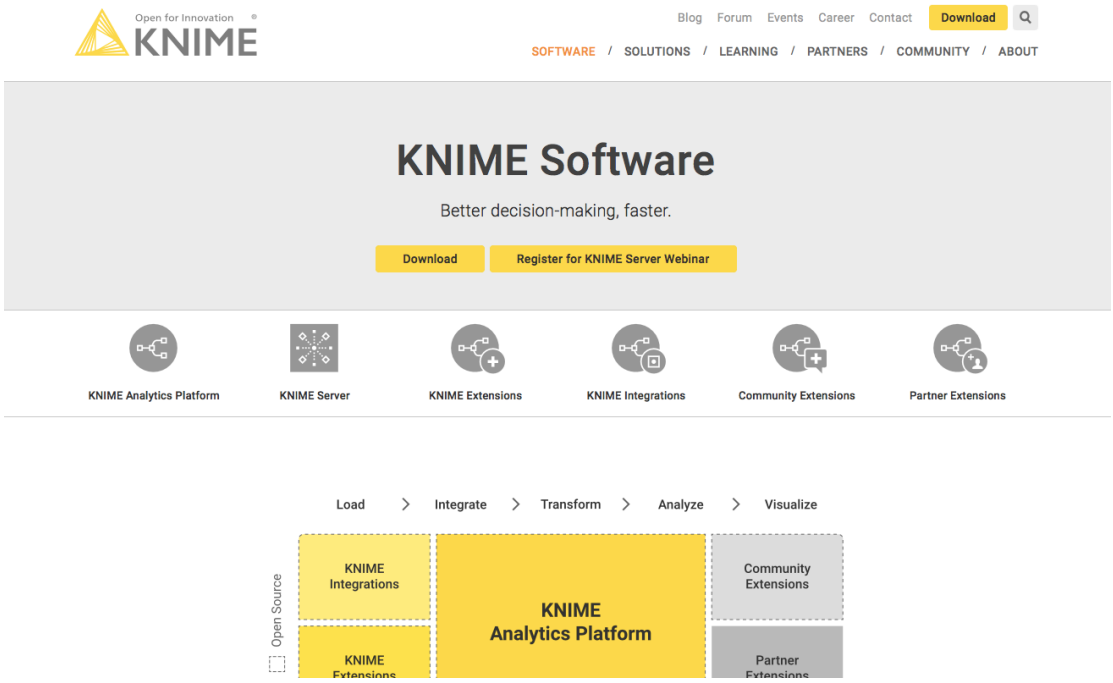
Şekil 1.2.1'de görüldüğü gibi sitede genel seçenek olarak sotfware, solutions, learning, partners, community ve about vardır. Konferanslar, duyurular, yarışmalar gibi etkinliklerin duyuruları bu web sitesinde yayınlanmaktadır.

For developers, for data scientists, for decision makers bölümlerinde ise geliştirici ve karar verici mevkilerdeki kişiler için değişik adımlar bulunmaktadır.



Şekil 1.2.2

Şekil 1.2.2'de en üst kısımda görülen **learning** başlığı altındaki eğitim serisine devam edilecektir. Getting started başlığı seçildikten sonra download, installation screenshots ile devam ederek ve ve daha sonra getting started başlığının altında görülen learning hub başlığı altında ve documentation'ın altındaki değişik eğitimler materyal olarak kullanılacaktır.



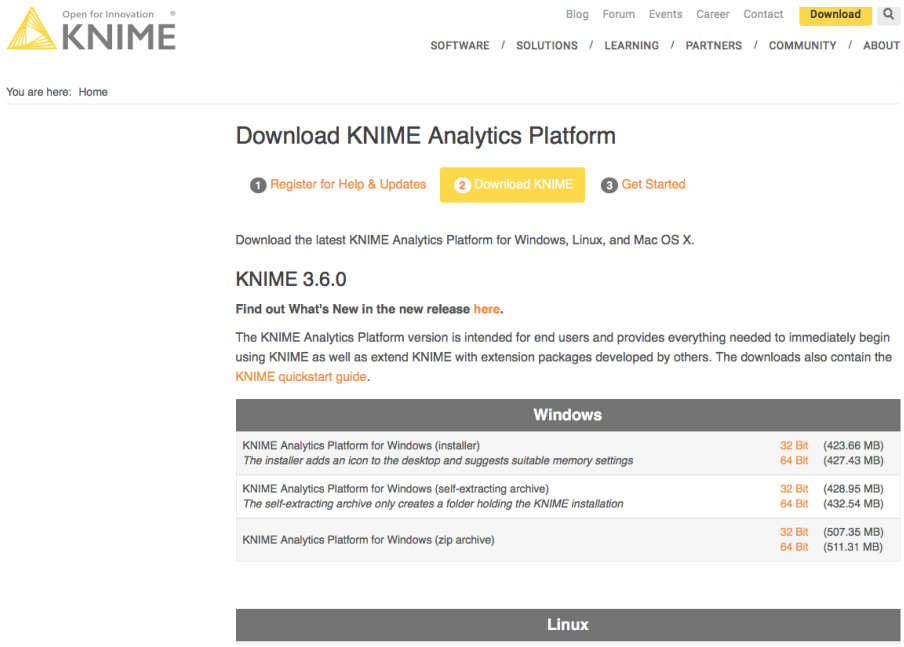
Şekil 1.2.3

Şekil 1.2.3 'de görülen software 'in altında yazılımla ilgili genel bilgiler bulunmaktadır. Data mining süitlerinin tek bilgisayar üzerinde çalışamayacak kadar büyük olmasından

dolayı artık Knime server da görülmektedir. SAS, Rapid miner gibi diğer tool'larında bu yüzden kendi sunucuları bulunmaktadır.

Bu kitapta, şekilde de görülen Knime'ın analytics Platform'u kullanılacaktır. Knime yazılımına gelen open source extansions'lar bulunmakta ve onun dışında değişik partnerler veya community (topluluk) tarafından üretilen extansions'larda bulunmaktadır. Anlamı bazı extansions'ların open source yani ücretsiz olarak kullanılabilmesiyken bazılarının ücretli olmasıdır.

Knime analytics platform, 1500 modül ve yüzlerce örnek içeren oldukça kapsamlı bir paket olup bu kitap için Knime alaytics platform bilgisayara kurulacak ve onun üzerinden devam edecektir.



Şekil 1.2.4

Şekil 1.2.4 de görüldüğü gibi, dowload seçeneği seçildikten sonra ilk pencere register (kayıt) penceresidir ama doldurulması zorunlu değildir. Haberlerin, güncellemelerin vb. Bildirimlerin gelebilmesi için bu bölümde mail ve vb. Bilgiler doldurulmalıdır. İkinci pencere yani download Knime seçilerek bilgisayarın modeli ve işletme sistemine göre uygun versiyonu seçilerek indirilme işlemi başlatılabilir.

1.3 Knime Kurulumu ve İlk Ekranlar (OSX) için

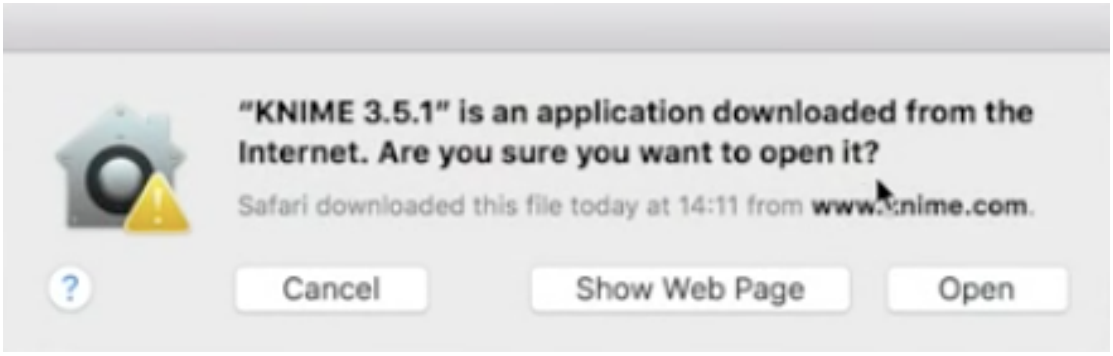
Bu bölümde Knime'ın indirildikten sonra bilgisayara kurulumu gösterilecektir. Knime indirildikten sonra (OSX için) .dmg olarak downloads (indirilenler) bölümüne gelir.

Şekil 1.3.1

Şekil 1.3.1, knime indirildikten sonra dowloads bölümünde görünen şeklidir. Bu aşamadan sonra yazıya çift tıklanır. Çift tıkladıktan sonra ufak bir pencere açılır. Bu sırada makineye bir disk gibi tanıtılıyor demektir. Daha sonra o pencere kapanarak aşağıda şekilde görünen pencere açılır.

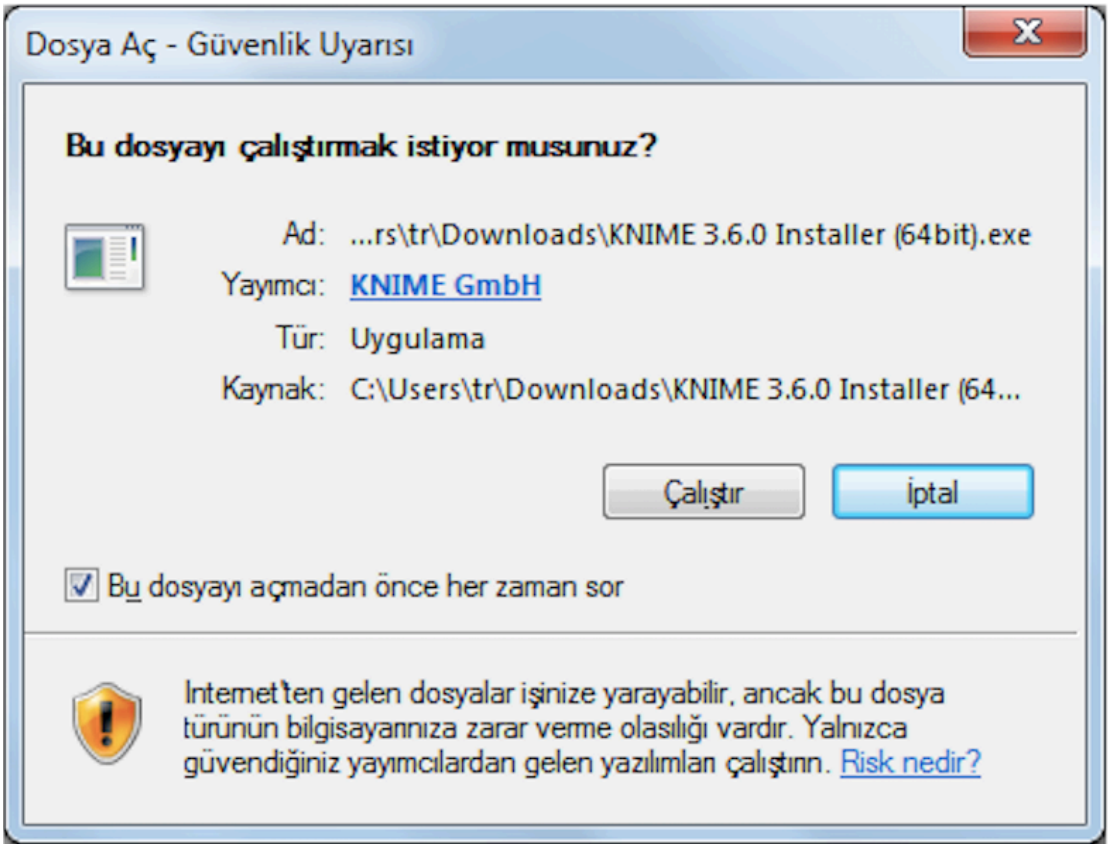
Şekil 1.3.2

Şekil 1.3.2, görülen bu penceredeki KNIME 3.5.1 ikonunu sürüklenerek applications kutucuğuna bırakılmalıdır. Bilgisayarda kurulu kalması için taşınması önemlidir. Daha sonrasında dosya silinebilir. Daha sonra knime bulunarak çalıştırılmaya başlanabilir.



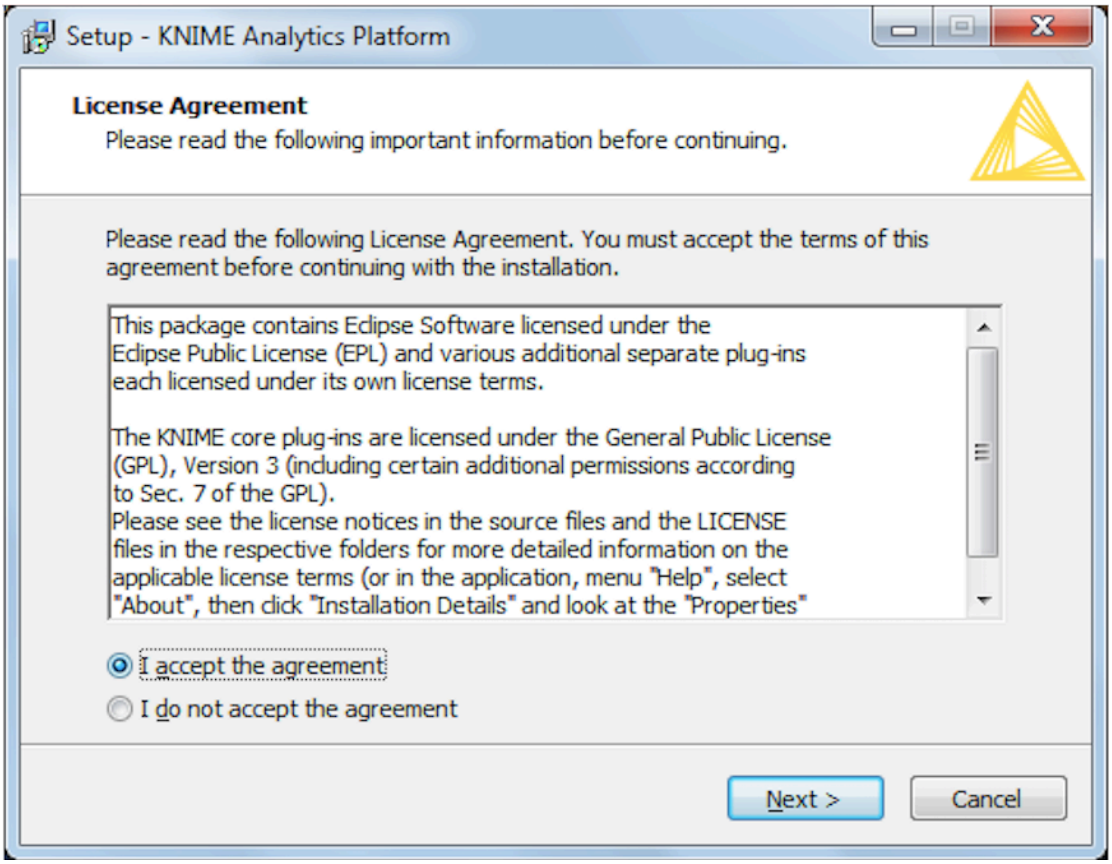
Şekil 1.3.3

Şekil 1.3.3, knime dosyası açılmak için çift tıklandıktan sonra ilk kez açılacağı için "uygulamaya güveniyor musunuz? " sorusunun olduğu pencere açılır. Open butonuna basıldıktan sonra knime açılır.



Şekil 1.3.4

Şekil 1.3.4, Windows bilgisayarlar için, şekil 1.2.4'deki ekranda yani Knime'ın sitesinde bilgisayarın işletme sistemine uygun seçeneğe göre indirme butonuna bastıktan sonra açılan ekrandır.

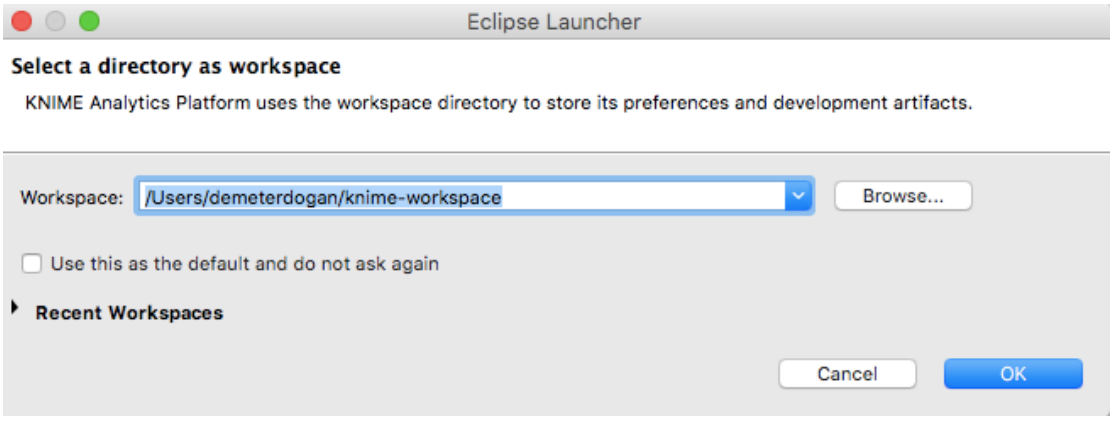


Şekil 1.3.5

Şekil 1.3.5, Windows işletme sistemine sahip bilgisayarda indirme işlemi başlarken sorulan ve I accept the agreement yani şartların kabul edildiği pencere açılır. Daha sonra next tuşuna basılır. Daha sonra şekil 1.3.6'daki ekran açılır.

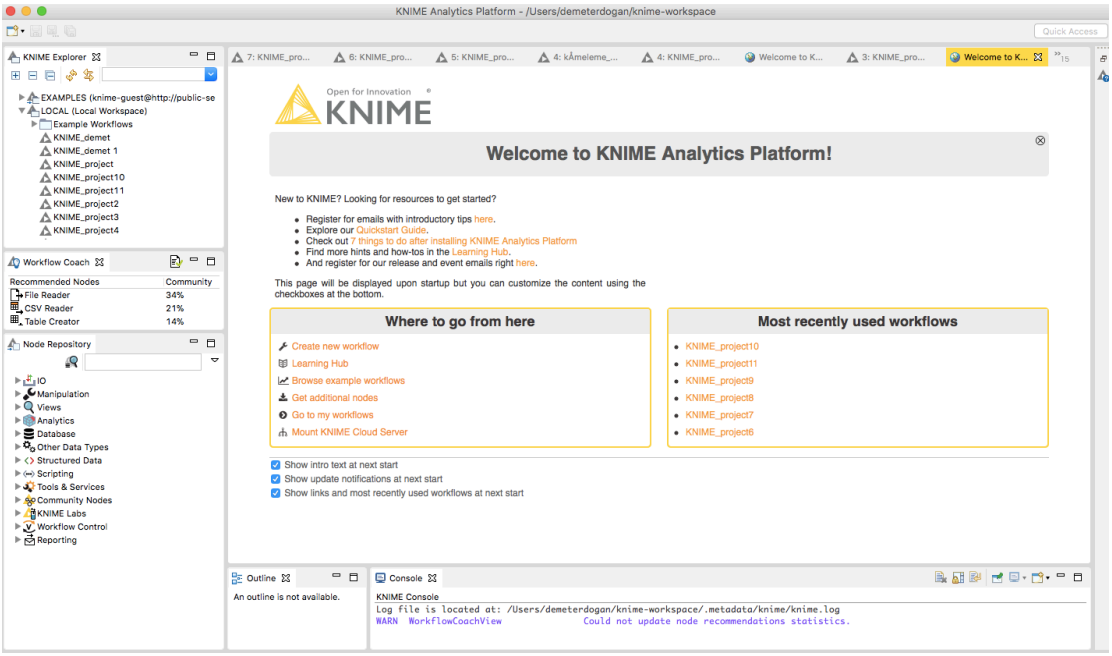
Şekil 1.3.6

Şekil 1.3.6'de görülen pencere ile Knime programı açılmaya başlar. Knime programını açmak için Knime ikonunu her tıklandığında öncelikle bu pencere açılır ve daha sonra şekil 1.3.7'de görülen pencere açılır.



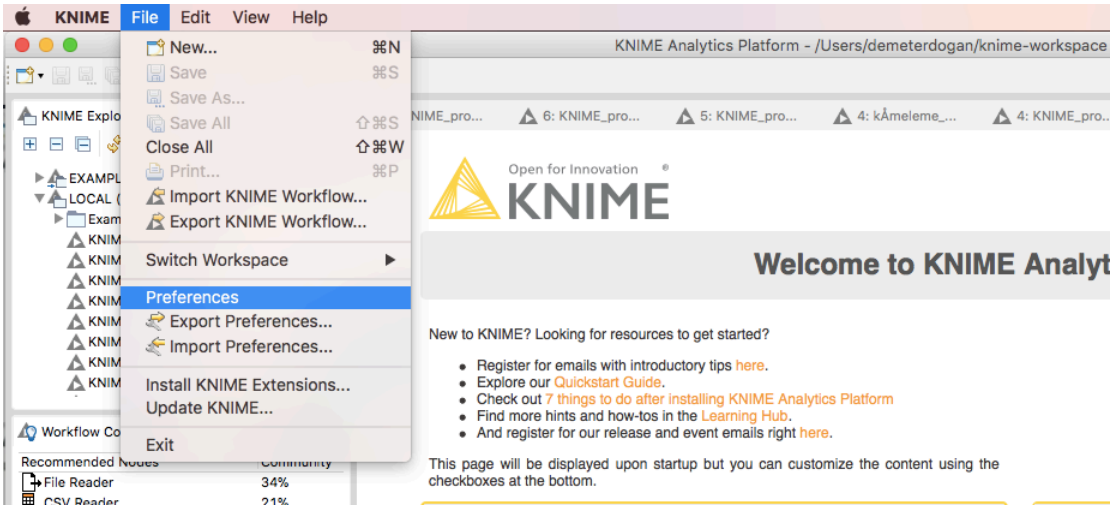
Şekil 1.3.7

Şekil 1.3.7, oluşturulacak olan workspace penceresini göstermektedir. Workspace, çalışmaların konulacağı yer olarak düşünülebilir. Bilgisayardaki herhangi boş bir yer gösterilebilir fakat bu örnekte default yerde bırakılmıştır.



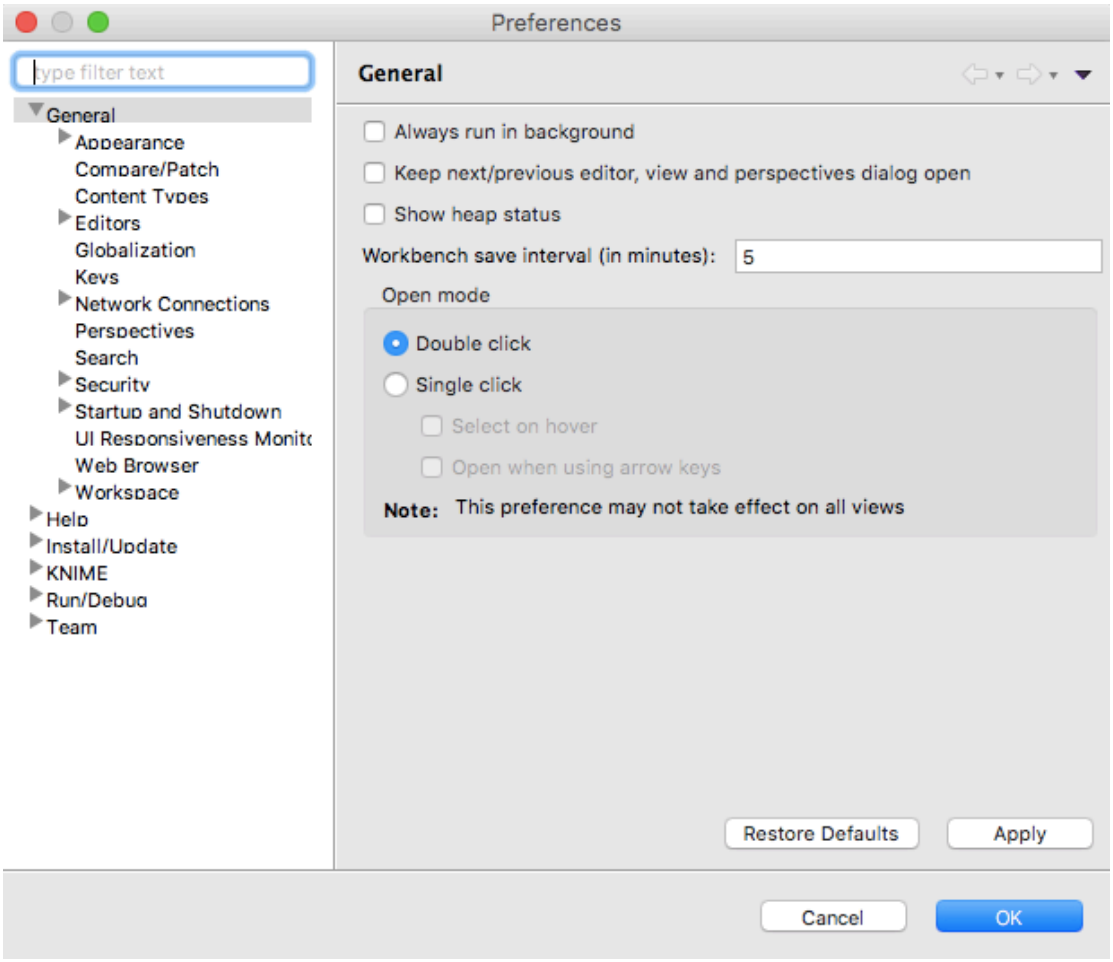
Şekil 1.3.8

Şekil 1.3.8, Knime açıldıktan sonra görülen ilk ekrandır ve bazı alt pencerelerden oluşmaktadır. **Knime explorer** çalışmaların kaydedildiği, örneklerin olduğu bölümdür. Workflow coach, çalışan kişiye knime'ın tavsiyeler verdiği penceredir. En önemli pencere repository penceresidir. Burada yüklü bazı düğümler vardır. Outline, yukarı pencerede çizilecek olanların ufak görüntüsünün gösterildiği penceredir. Node description ise kullanılan node'un tanımının yapıldığı penceredir.



Şekil 1.3.9

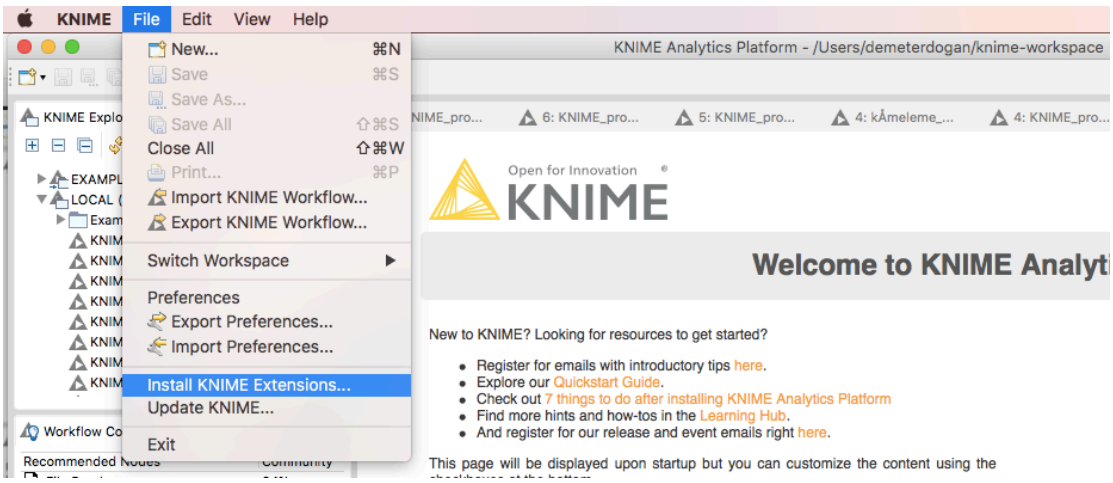
Şekil 1.3.9, Knime'da ayarları yapabilmek için preferences sayfasına giriş yapılacak yeri göstermektedir.



Şekil 1.3.10

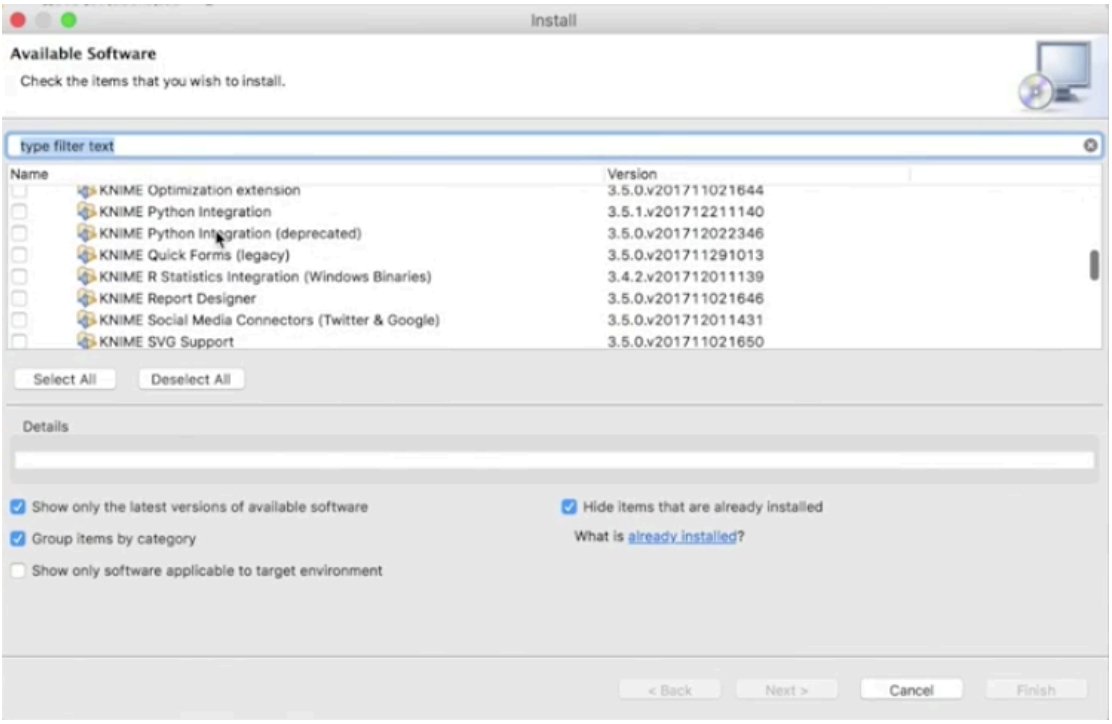
Şekil 1.3.10, preferences penceresini göstermektedir. General yazan bölümde genel durumlarla ilgili ayarlar yapılabilir. Örneğin ekran görüntüsü ile ilgili, editör ile ilgili değişiklikler bu bölümde yapılabilir.

Help (yardım) ile ilgili, install /update ile ilgili, knime (database, explorer, Knime arayüzü) ile ilgili, çalıştırıldıktan sonra (run/debua) console ile ilgili bilgiler preferances bölümünden ulaşılabilir.



Şekil 1.3.11

Şekil 1.3.11, knime'da çok önemli başka bir bölüm olan install KNIME extensions bölümüne nereden girileceğini göstermektedir. Bu seçeneğe tıklayınca Knime'ın kullanıcı için açık olan extensions'ların listesinin olduğu pencere açılacaktır.



Şekil 1.3.12

Şekil 1.3.12, python, R, textprocessing, weka, büyük veri ile ilgili bağlantı modüller vb. Modüller buradan seçilerek knime'a install edilir ve işleme başlanabilir.

2.BÖLÜM: KNIME ORTAMINI TANIMA VE BASİT UYGULAMALAR

2.1 Knime ile Veri Bilimine Giriş

Bu bölümde Knime tanıtımı yapılacak ve veri bilimiyle bağlantısı açıklanacaktır. Knime, data mining suit ismi verilir. Yani veri madenciliği yapılan bir yazılımdır. Bu yazılımlar genellikle bir akış mantığı ile çalışır. Örneğin Rapid Miner, Weka ücretsiz yazılımlardandı fakat Rapid Miner güncelleme ile birlikte artık ücretli olmuştur. Yazılımın open source olması ücretli olmasına engel değildir. 10 000 satıra kadar ücretsiz sonrası için ücretlidir fakat Knime tamamen ücretsizdir.

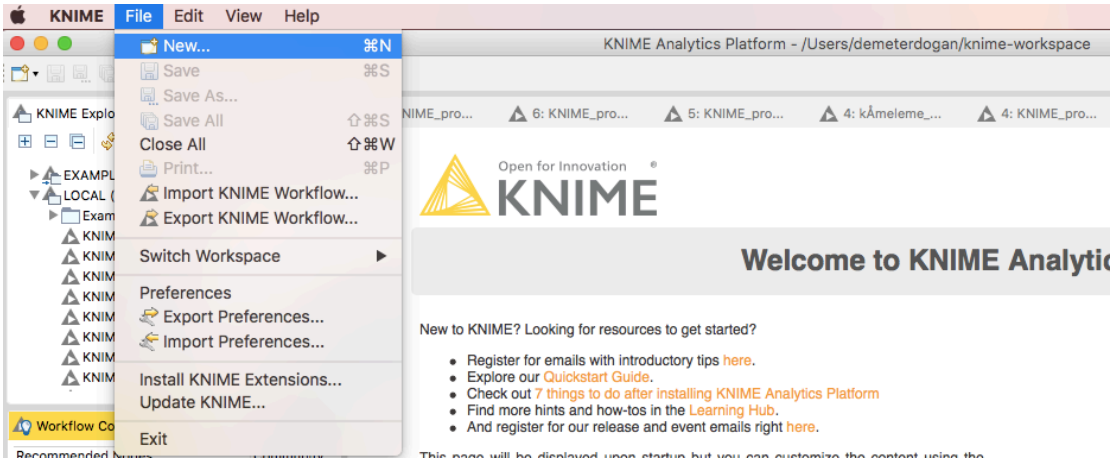
Basitçe amaç bir veri kaynağından bir hedefe/bilgiye akış sağlamaktır. Amaç, müşteri analizi, kampanya analizi, tahminler, farklı kaynaklardan gelen veri birleştirilmesi vb. Olabilir.Knime bunların hepsini karşılayabilen ücretsiz bir data mining suit'tir. Versiyon bu kitap yazılırken 3.5.2 fakat versiyon değişmesi büyük farklılık yaratmamaktadır sadece ufak değişiklikler olabilir.

cinsiyet.xlsx

Boy	Kilo	Cinsiyet
185	85	erkek
174	65	kadın
180	79	kadın
168	58	kadın
175	80	erkek
170	70	erkek
169	50	erkek
183	74	erkek
180	80	erkek
170	60	kadın

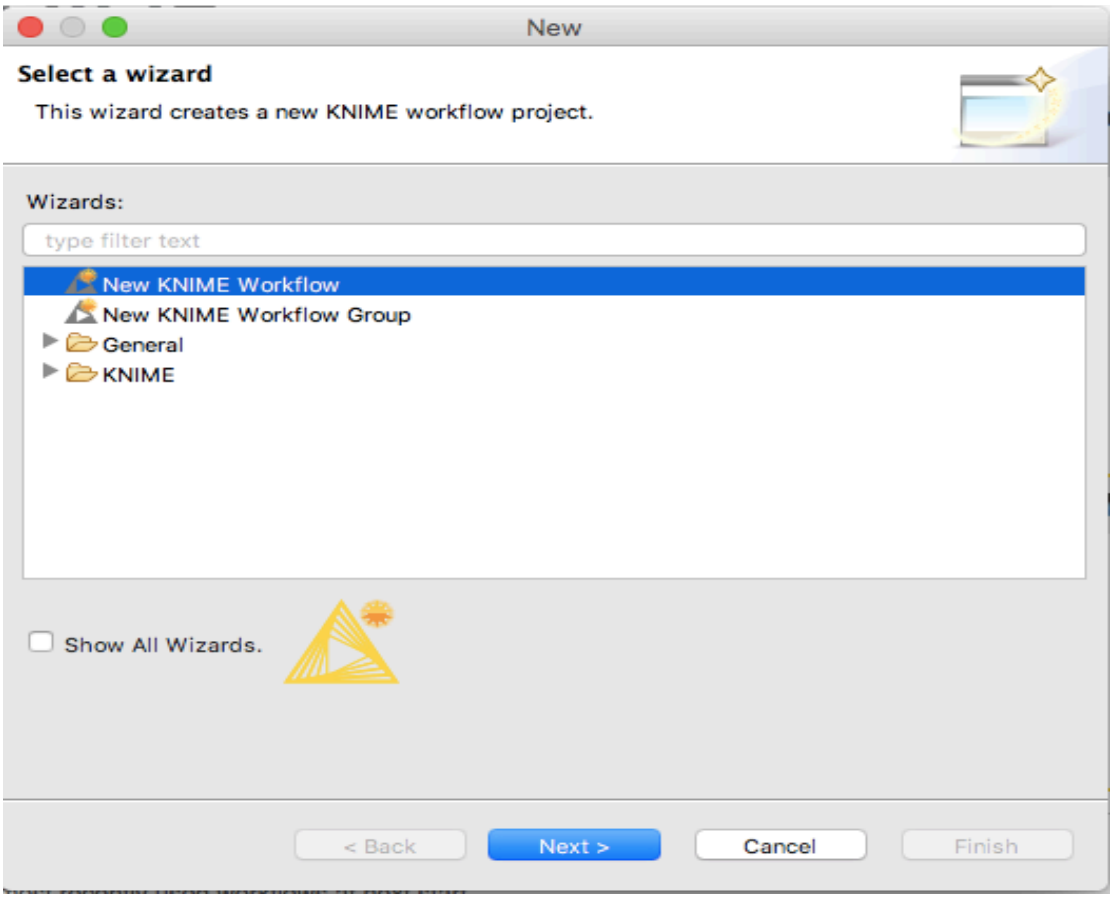
Şekil 2.1.1

Şekil 2.1.1, bu bölümde kullanılacak örnek excel dosyasını göstermektedir. Buradaki sayılar istenildiği gibi değiştirilebilir. Burada amaç, makineye bu bilgiler verilerek onun öğrenmesi sağlanıp cinsiyet tahmini yaptırılmaktır. Gündelik hayatta 190 cm boyu olan 90 kg olan bir kişi için genel olarak tahmin erkek olacağı yönde olması tecrübeye (öğrenilmiş) dayalı bir bilgidir. Makine için de buna benzer bir işlem yapılacaktır.



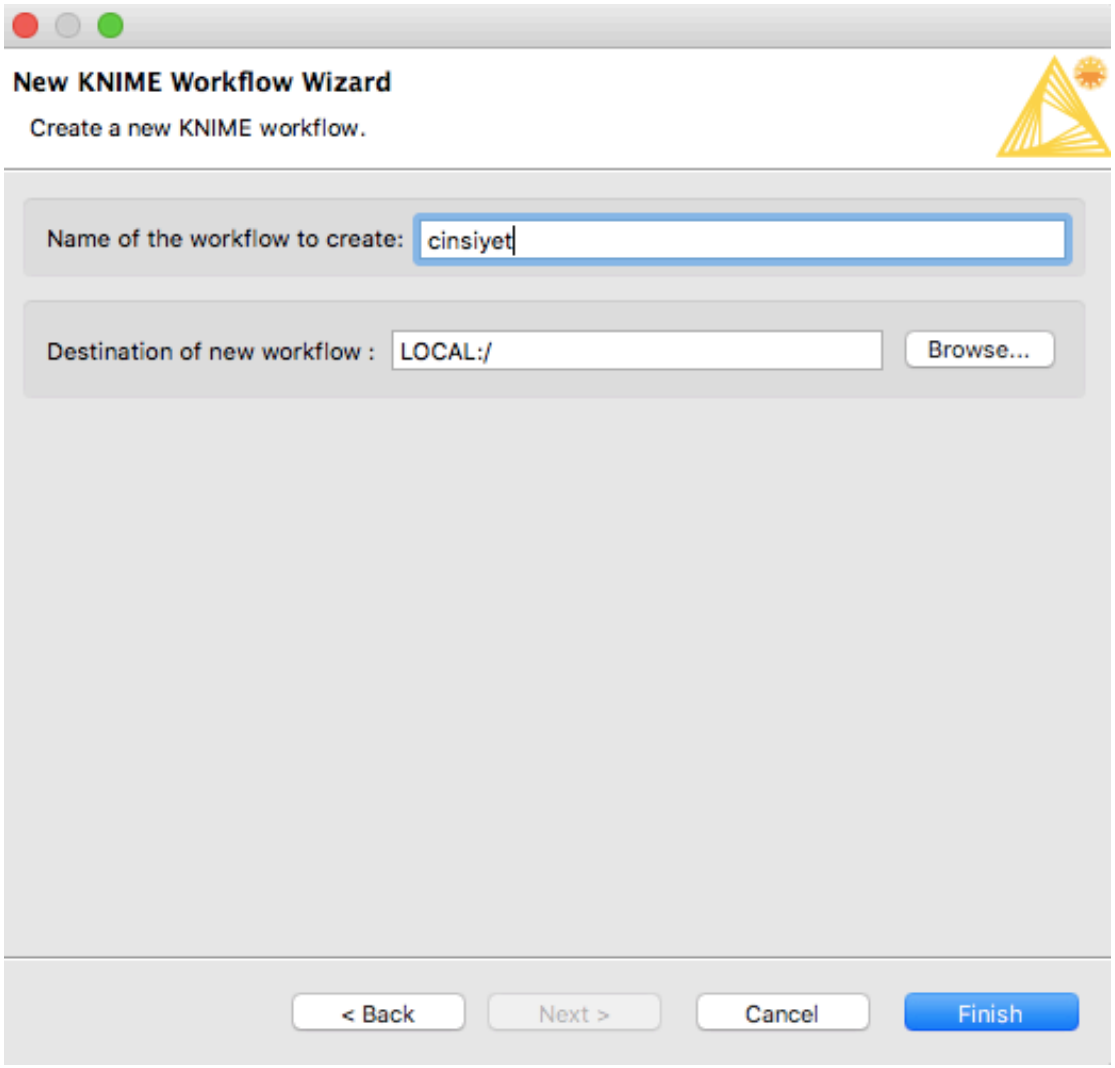
Şekil 2.1.2

Şekil 2.1.2, knime'da çalışmak için yeni bir workflow penceresi açılacak yeri göstermektedir. Burada new seçeneği seçildikten sonra aşağıdaki şekildeki pencere açılır.



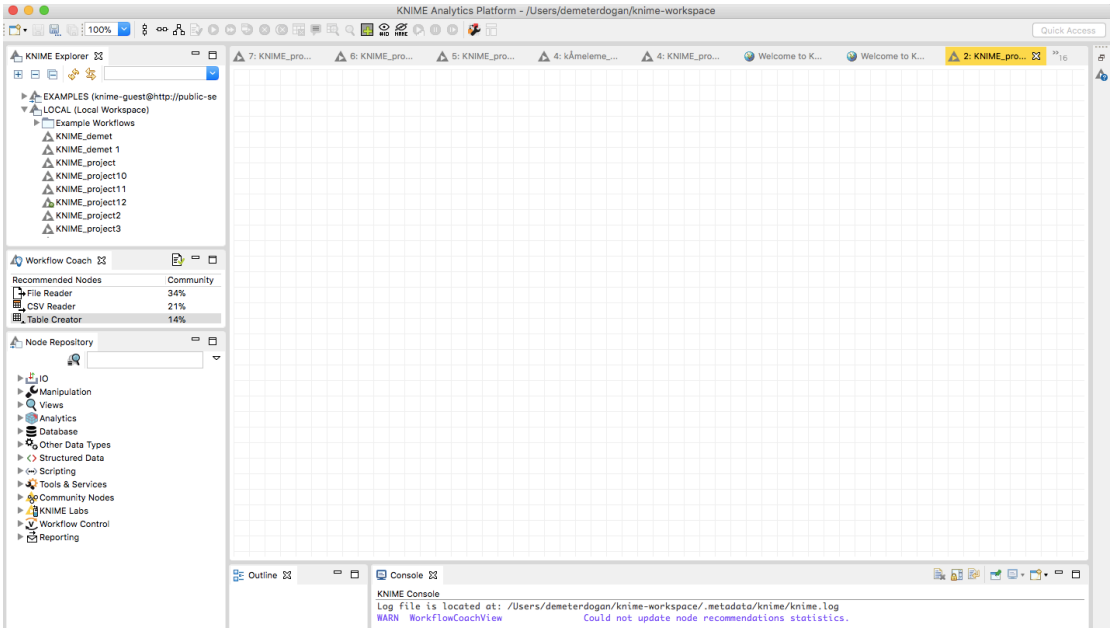
Şekil 2.1.3

Şekil 2.1.3 de görülen new Knime workflow seçeneği seçilerek yeni bir çalışma penceresi açılmalıdır.



Şekil 2.1.4

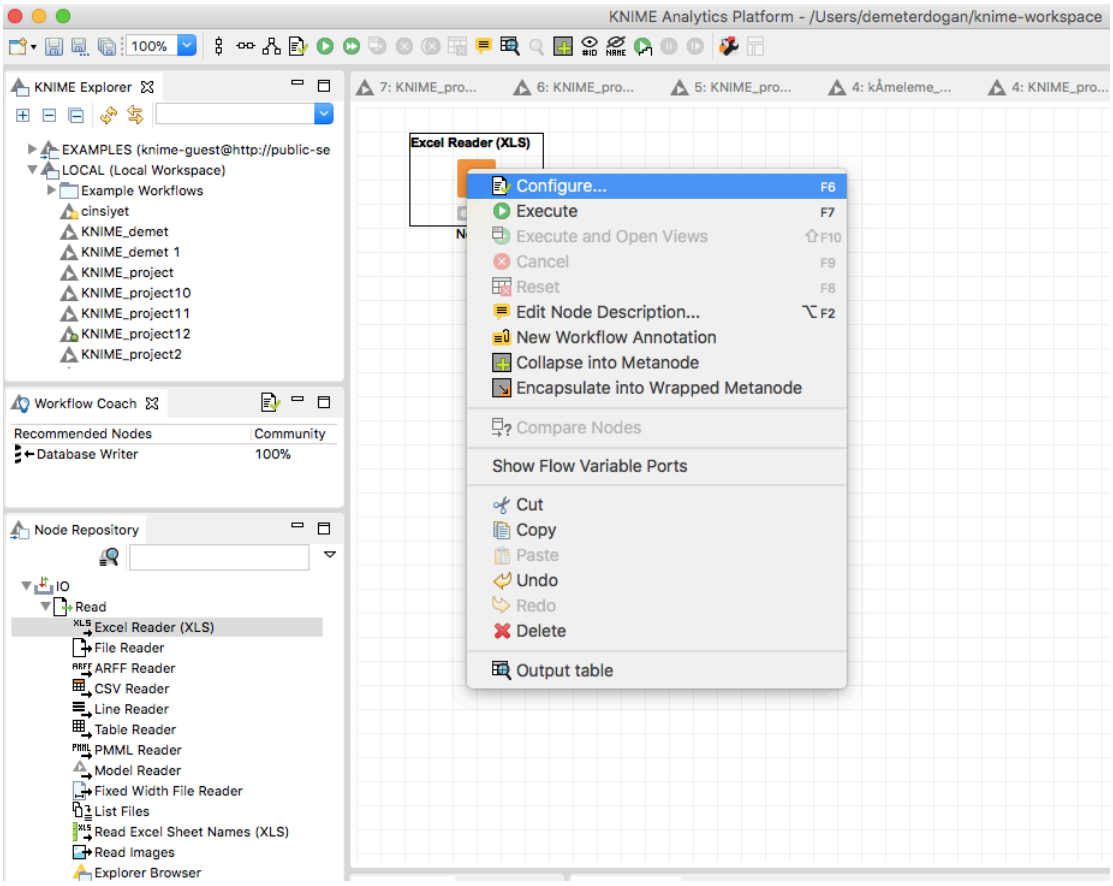
Şekil 2.1.4, yeni açılacak olan workflow ismi ve yeri bu pencerede belirlenir. Bu örnekte workflow ismi cinsiyet olarak yazılmıştır fakar istenilen veya otomatik Knime'ın verdiği isim de kullanılabilir. Bu bölümler geçildikten sonra knime'da yeni boş bir workflow (çalışma penceresi) açılır.



Şekil 2.1.5

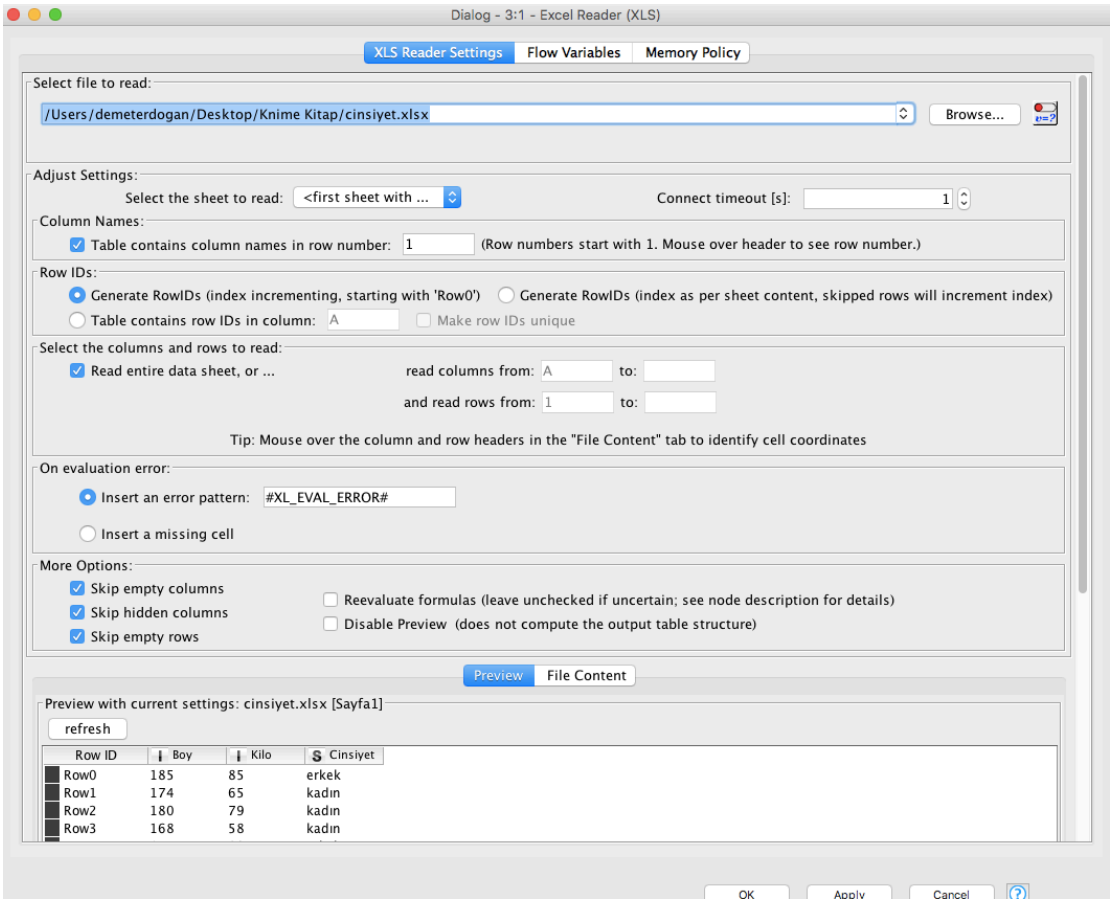
Şekil 2.1.5, açılan boş workflow penceresini göstermektedir. Şekil 2.1.1'da gösterilen excel veri setini Knime'a aktarmak için excel reader node'u (operatörü) kullanılır. Eğer csv dosyası olsaydır o zaman csv reader operatörü kullanılırdı. Aktarılacak dosya tipine göre repository bölümünden IO dosyalarının alt birimi olan read klasöründen seçilmelidir.

Bu örnekte excel kullanılacağı için excel reader seçilip sürüklenerek workflow penceresine bırakılır ya da excel reader operatörüne (node'una) çift tıklanır. Daha sonra o node içerisine cinsiyet veri setini yüklemek için reader operatörüne bir kez sağ tıklanır.



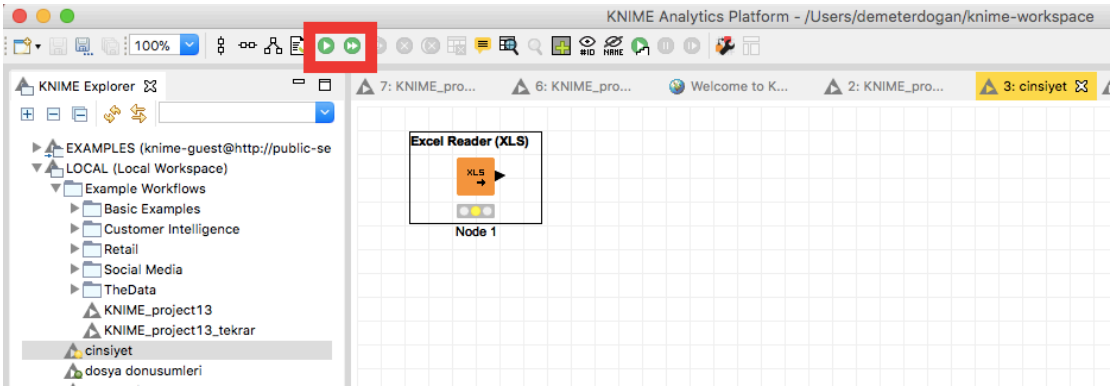
Şekil 2.1.6

Şekil 2.1.6 da görüldüğü gibi sağ tuşla tıkladıktan sonra bir pencere açılır. Burada configure seçeneği seçilmelidir. Bu o operatörle ilgili şekillendirme seçeneğidir. İçerisine dosya ekleme bu bölümden yapılır. Genellikle çoğu node (operatörde) configure bölümünde ayarlamalar yapılmalıdır.



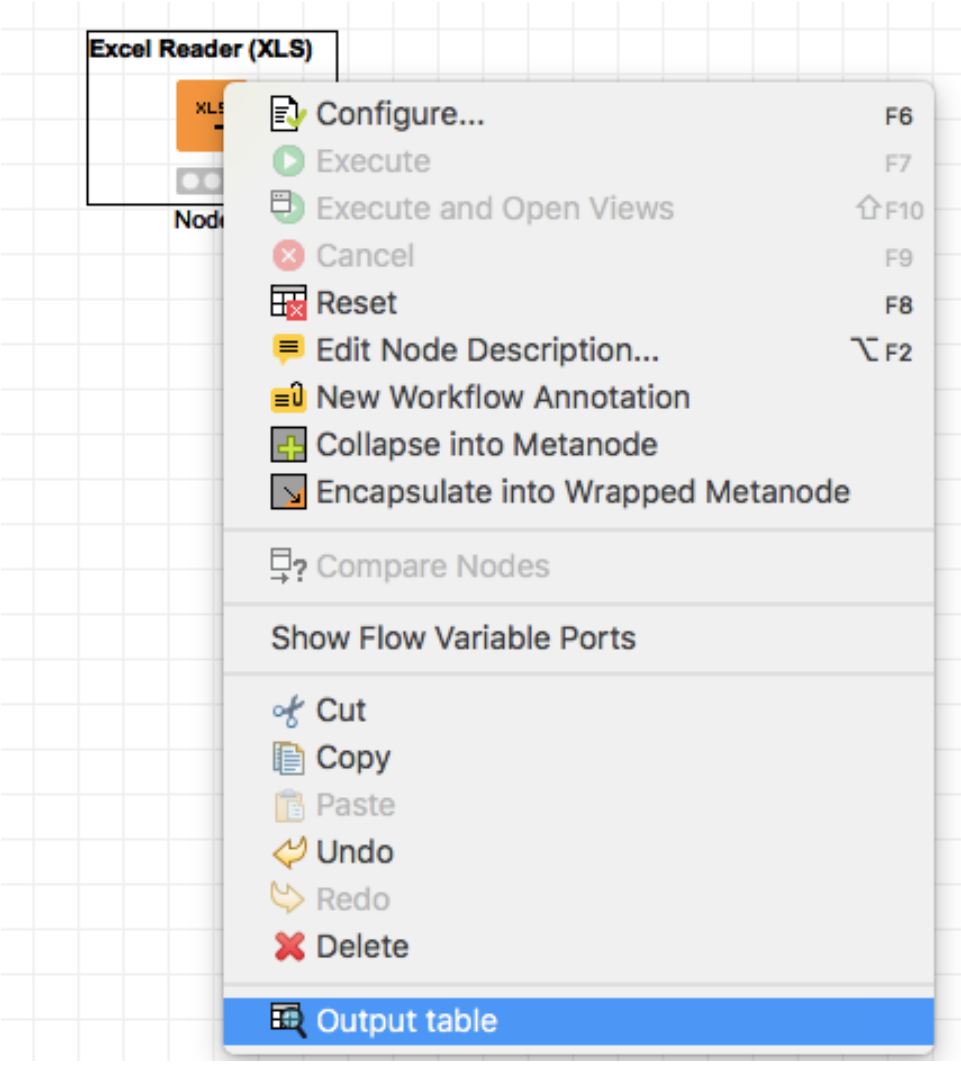
Şekil 2.1.7

Şekil 2.1.7, bir önceki şekilde configure seçtikten sonra açılan penceredir. Burada browse seçeneği seçilerek cinsiyet dosyası bilgisayarda kayıtlı olduğu yerden bulunup seçilmelidir. Table contains column names in row number 1 seçeneği seçilmelidir. Bu kolon başlıklarının başlık ismi bölümü olarak algılanmasını sağlamaktadır. Eğer bu seçilmezse kolon başlıkları birinci row (sıra) gibi algılanır. Bu seçenek seçildikten sonra aşağıda excel görünümlü örnek veri seti göstergesinin üzerinde refresh buonuna basılmalıdır çünkü ancak o zaman kolon başlıkları yenileme durumundan sonra satır başlığı olarak algılanır.



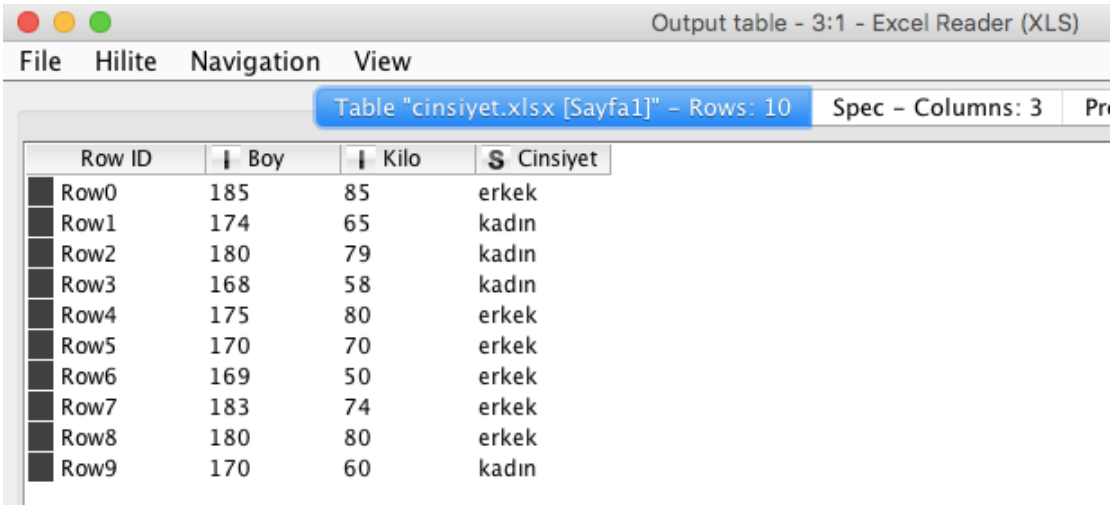
Şekil 2.1.8

Şekil 2.1.8, yukarıdaki şekilde sisteme yüklenen veriden sonra çalıştırılarak verinin sistemde akması sağlanmalıdır. Şekilde görülen kırmızı penceredeki alanda > ve >> işaretlerinden > olan üzerinde çalışılan operatörün execute edilmesini, >> ise tüm operatörlerin execute edilmesini sağlamaktadır.



Şekil 2.1.9

Şekil 2.1.9, sistem execute edildikten (çalıştırıldıktan) sonra oluşan output table'ın nasıl açılacağını göstermektedir.



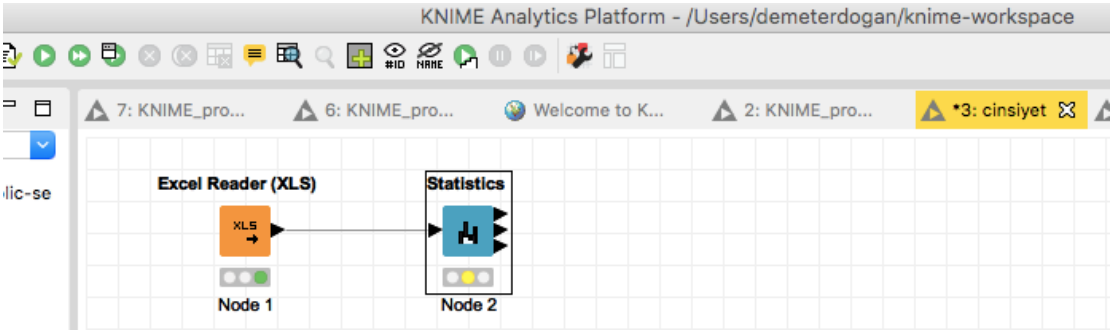
Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	79	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek
Row9	170	60	kadın

Şekil 2.1.10

Şekil 2.1.10, örnek veri setinin sisteme aktarılmasını ve oluşan output table 'ı göstermektedir.

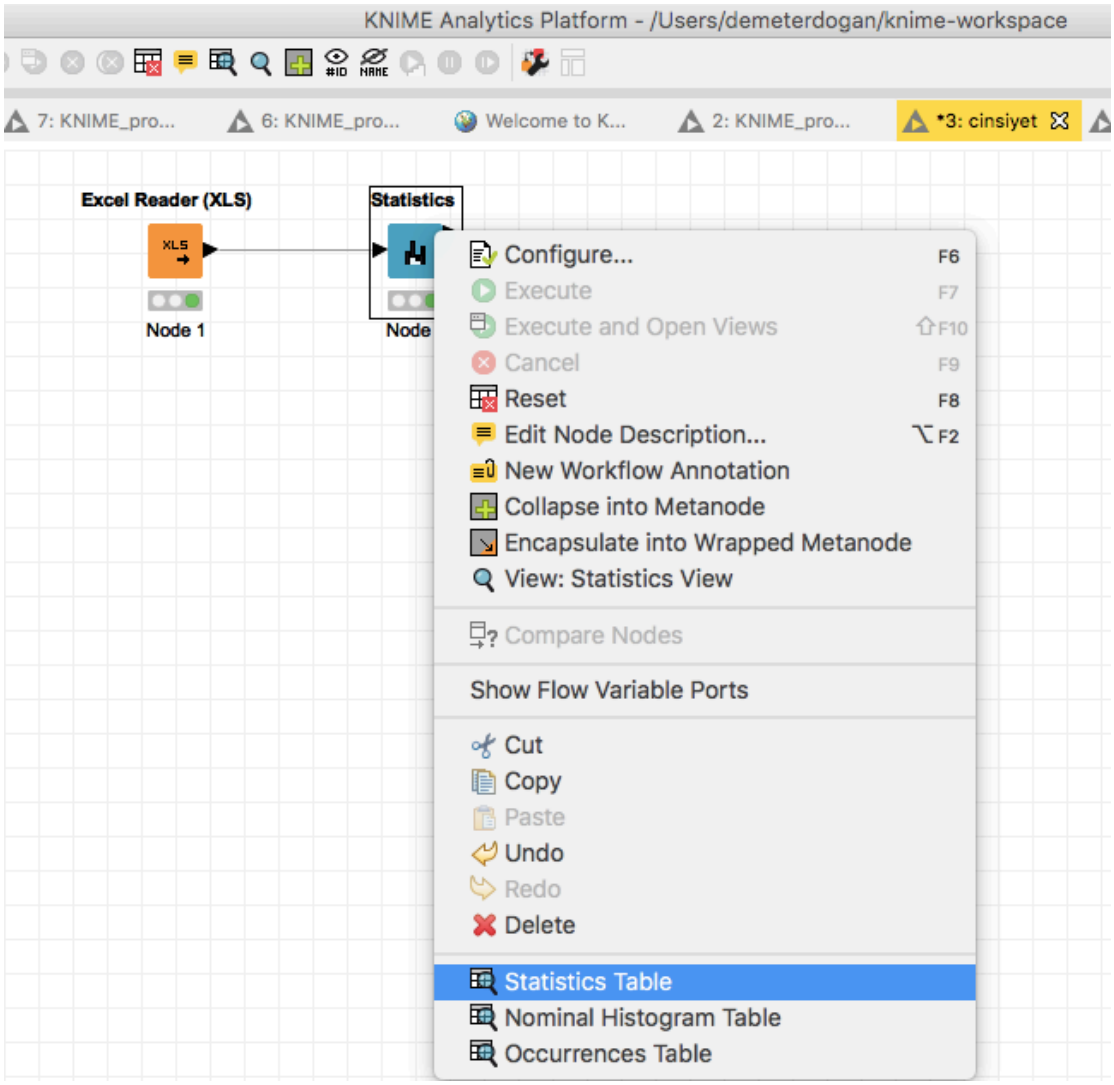
Veri bilimi projesinde iki adım önemlidir. Birincisi, problemin tanınması, ikincisi ise verinin tanınmasıdır. Bu örnekte, makineye boy ve kilo bilgileri verilerek cinsiyetin öğretilmesi. Verinin bunun için uygun olup olmadığı test edilmelidir. Buna yardımcı olacak statistics isimli düğüm (operatör) kullanılabilir.

Node repository bölümünde statistics operatörü aratılıp bulunduktan sonra onun üzerine bir kez tıklayıp tutularak çalışılacak workflow penceresine bırakılır. Excel reader ile statistics bağlantısı şekildeki gibi yapılır. Bu bağlantı ile excel verisi statistics'e akıtılır.



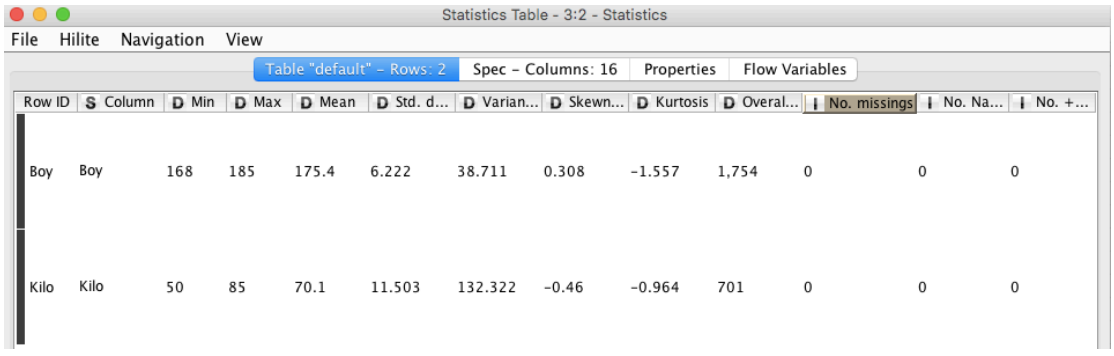
Şekil 2.1.11

Şekil 2.1.11'de görüldüğü gibi bağlantı yapıldıktan sonra program execute edilir. Node'lar yani düğümlerin altındaki yanan kırmızı, sonra sarı sonra yeşil ışıkların anlamları vardır. Kırmızı, problem olduğunu gösterir ve örneğin verilen veride ya da configure'de bir problem olabilir ve bunları değiştirmekle giderilebilir. Sarı, bekleme durumudur yani çalıştırılmayı beklemektedir. Yeşil, herhangi bir problem ypk başarılı biçimde çalıştı demektir.



Şekil 2.1.12

Şekil 2.1.12, program çalıştırıldıktan sonra veri seti hakkında bilgi edinmek için statistics table'ın açılabileceği pencereyi göstermektedir.

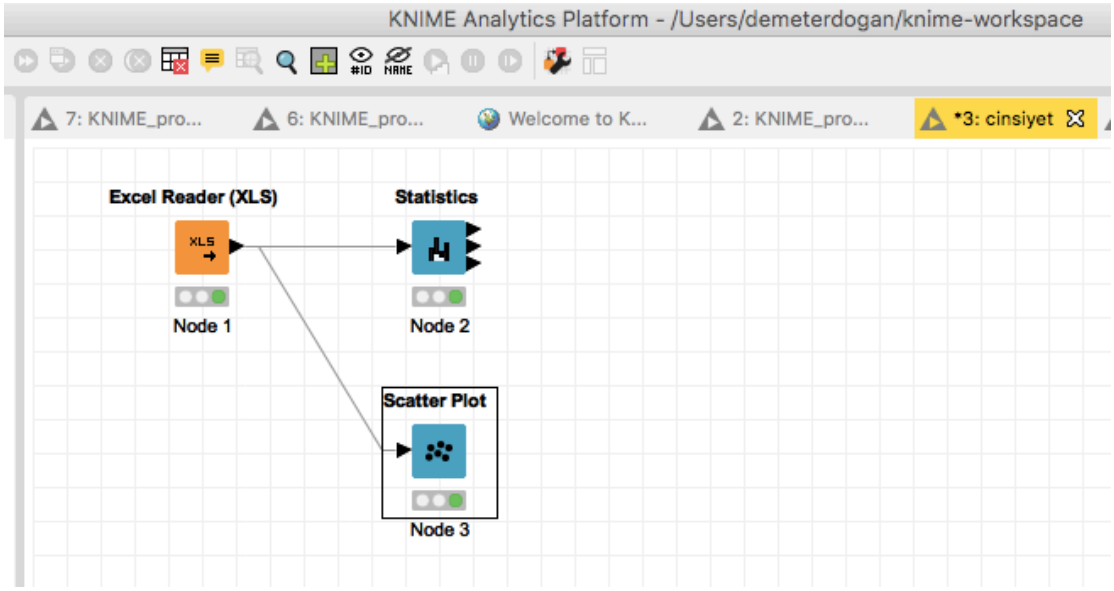


Row ID	Column	Min	Max	Mean	Std. d...	Varian...	Skewn...	Kurtosis	Overal...	No. missings	No. Na...	No. +...
Boy	Boy	168	185	175.4	6.222	38.711	0.308	-1.557	1,754	0	0	0
Kilo	Kilo	50	85	70.1	11.503	132.322	-0.46	-0.964	701	0	0	0

Şekil 2.1.13

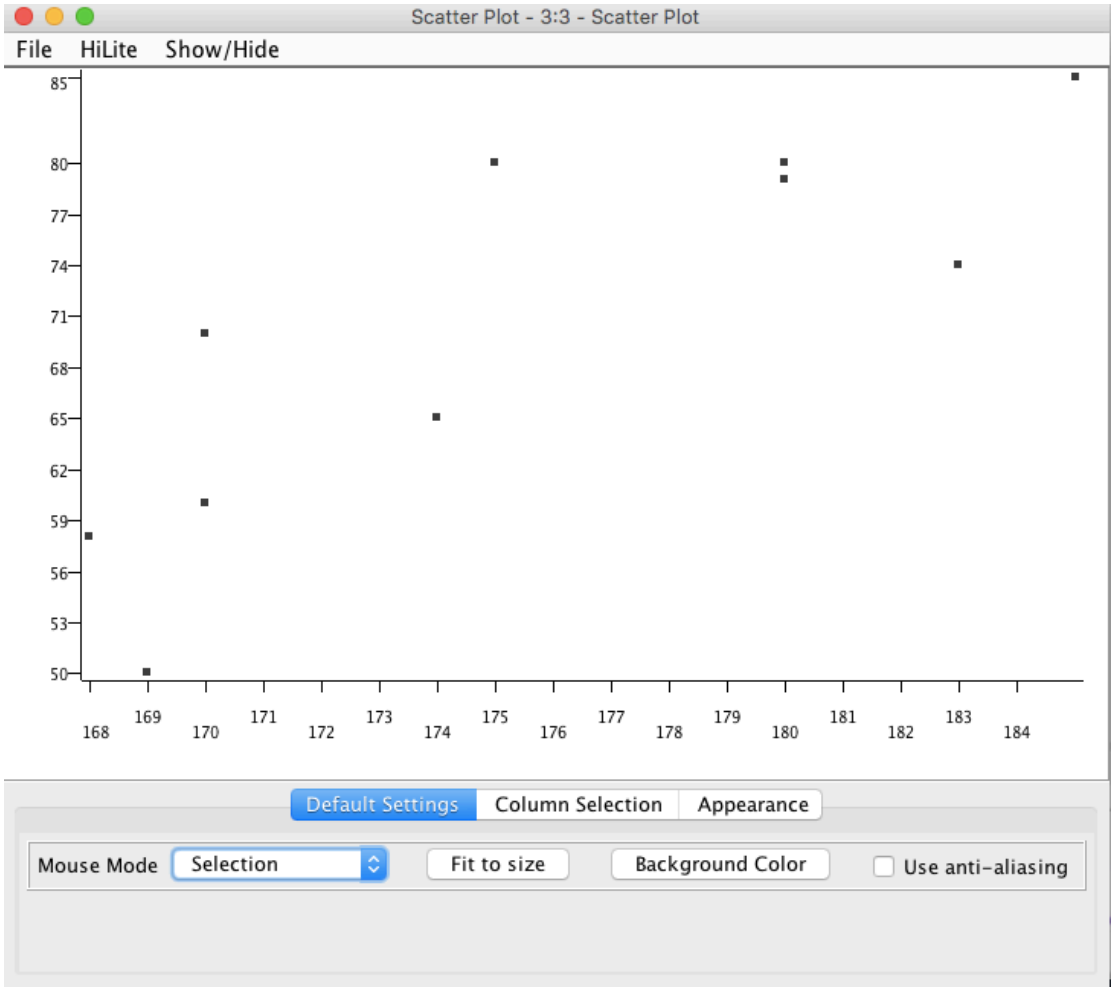
Şekil 2.1.13, kolonları boy ve kilo olarak iki satırda detaylı bilgileri yazılmıştır. Mininmum, maximum değerleri, ortalamaları, number of missings (eksik veri sayısı) vb.

Kolonlar verilmiştir. Örneğin bu veri setinde eksik veri bulunmamaktadır. İstatistiksel olarak veri kümesindeki bilgiler verilmiştir.



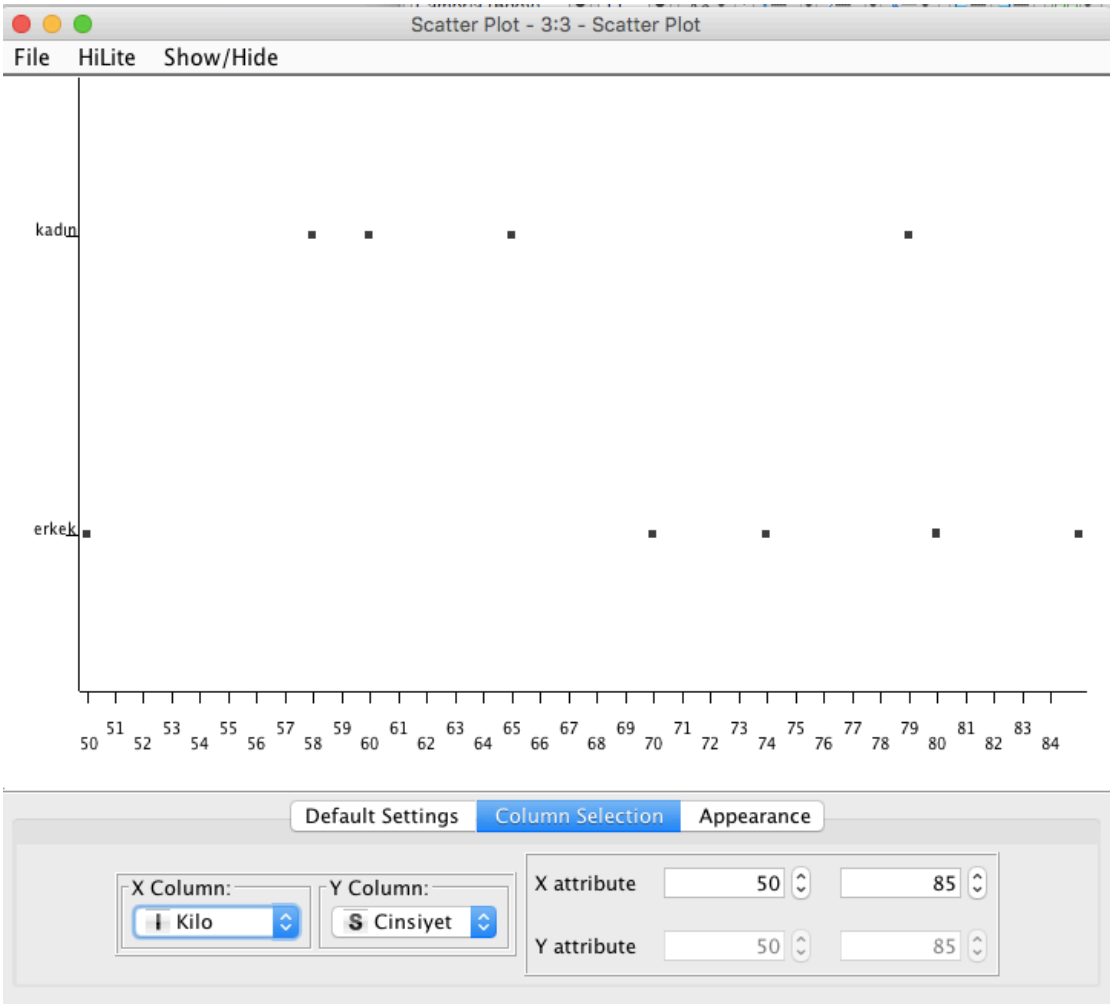
Şekil 2.1.14

Şekil 2.1.14, veri setinin bu sefer de görsel olarak gösterilmesine yardımcı olan scatter plot operatörünün eklenmesini ve sistem ile bağlantısını göstermektedir.



Şekil 2.1.15

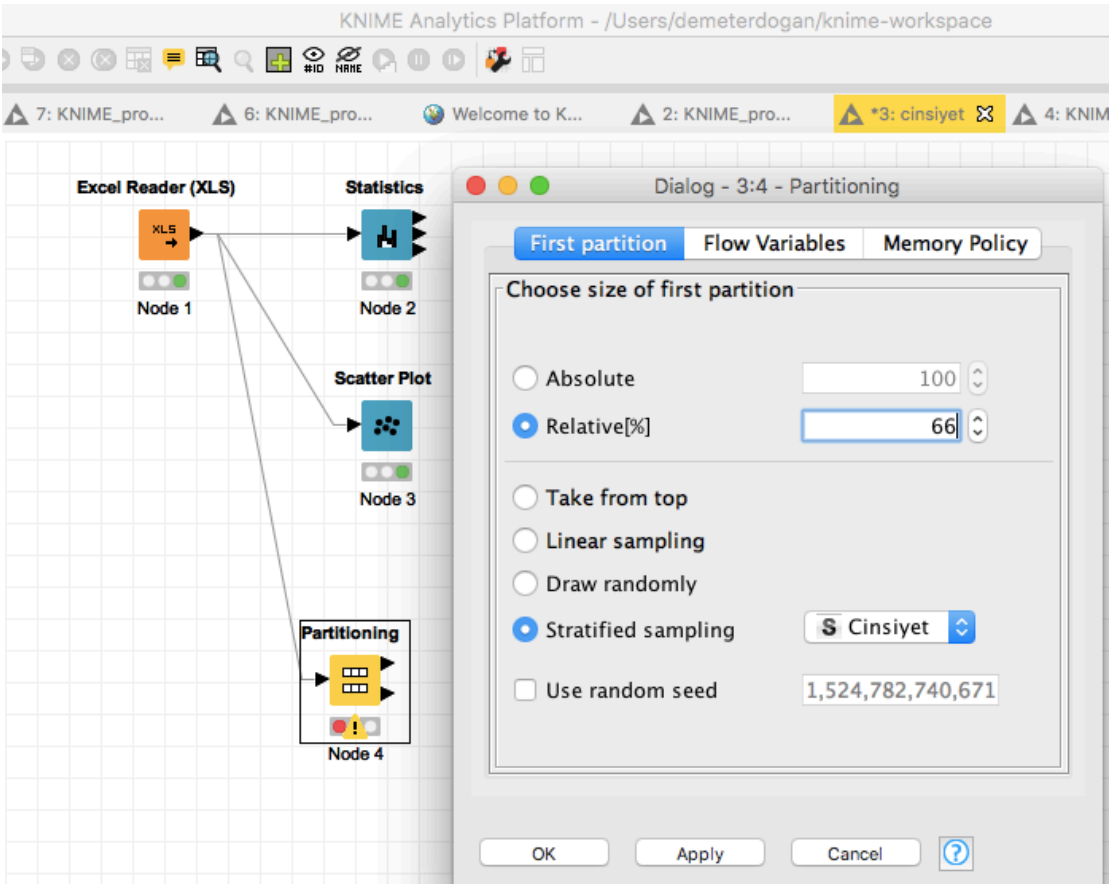
Şekil 2.1.15, iki boyutlu olarak veri setinin grafikleştirilmiş halini göstermektedir. Y ekseninde kilo, x ekseninde ise boy verileri yerleşmiştir.



Şekil 2.1.16

Şekil 2.1.16, kilo ve cinsiyete göre çizilmiş grafiği göstermektedir. Burada 58-65 arası kadın, 69-85 arası genellikle erkek denebilir. Outlier denilen yani bu aralıkların dışında kalan kadın ve erkekler de bulunmaktadır.

Herhangi bir makina algoritması iki aşamadan oluşur. Birincisi eğitim zamanı, ikincisi test zamanıdır. Bu yüzden veri iki parçaya ayrılarak bir kısmı eğitimde bir kısmı da testte kullanılacaktır. Makine öğrenmesi için bir çok algoritma kullanılabilir. Bu yüzden öğrenilen verinin test edilmesi onun verimliliğini ölçmesi açısından önemlidir. Veriyi parçalamak için partitioning operatörü (düğümü) kullanılacaktır.



Şekil 2.1.17

Şekil 2.1.17'de görülen pencerede partitioning operatörü ile verinin 66% ve 34% oranlarında verinin bölünmesi gerektiğini göstermektedir.

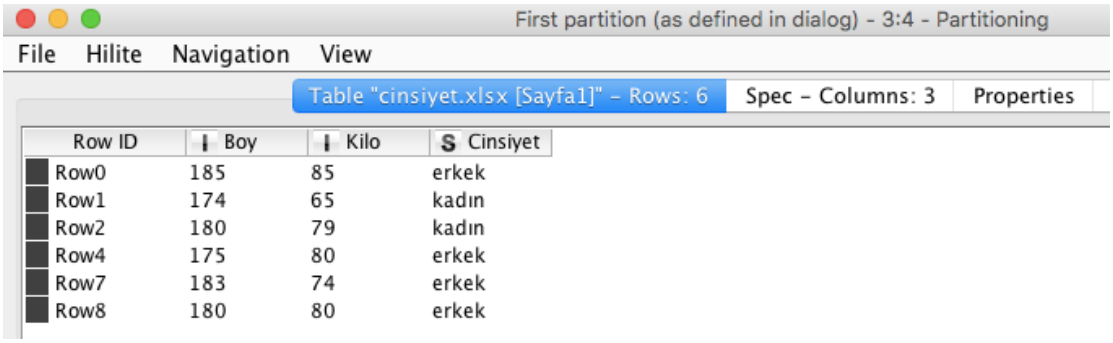
Absolute seçilseydi orada tam olarak kaç verinin test için kaç verinin eğitim için kullanılacağı belirtilmiş olunurdu. Örneğin absolute 100 seçilseydi, 100 örnek train için kullanılacak kalanı train için kullanılacak anlamına gelirdi. 66% ilk partition'dan yani operatörün sağ sarafındaki üstteki oktan çıkar ve bu train için kullanılır. Kalan 34% ise test için kullanılır.

Take from top, veri setinde sırayı bozmadan en baştan verileri al anlamına gelmektedir. Burada sıkıntı eğer veri sıralıysa o zaman test sonucu iyi gelmez. Örneğin veride erkekler en başta sonra kadınlar geliyorsa ve bu seçilirse, erkeklere göre öğrenme gerçekleşir ve testte hiç erkek olmama ihtimali olur.

Linear sampling, veriyi atlamalı olarak alır. Örneğin burada 66% ve 34% e göre 2 train sonraki bir taneyi de test verisi olarak al anlamına gelmektedir.

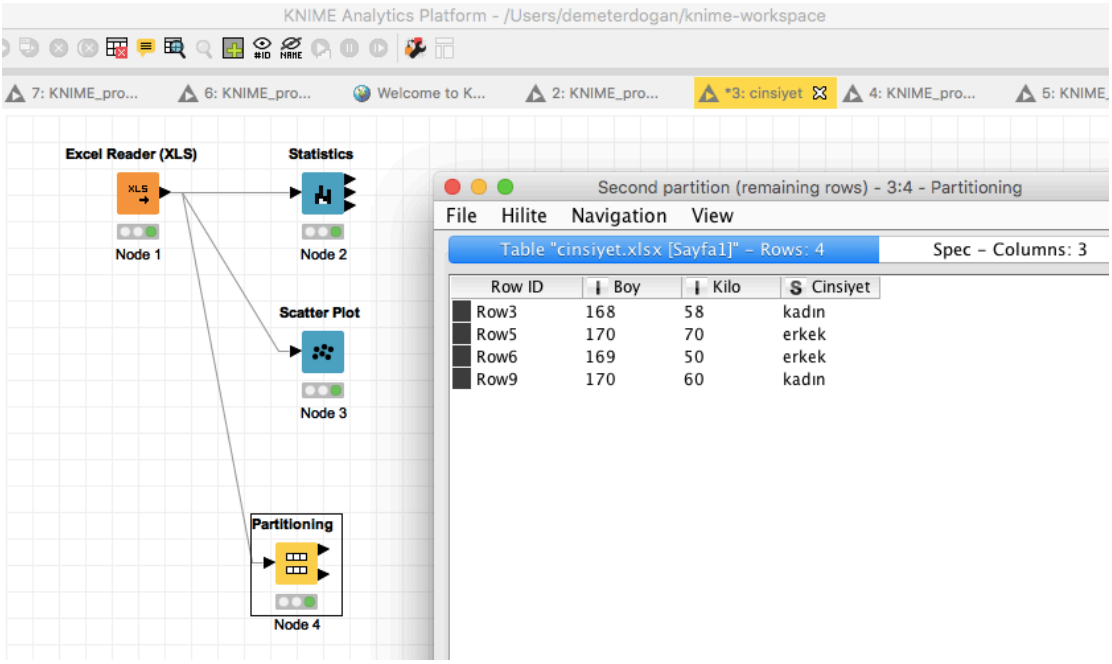
Draw randomly, verileri rastgele bölerek verilen oranda alır. Her çalıştırmada sonuç değişebilir çünkü veri setini karıştırıp rastgele seçim yapar train ve test için.

Stratified sampling, verilen etikete göre veri setindeki yüzdelikten hem train setinde bu yüzdeliği korur hem de test setinde bu yüzdeliği korur. Örneğin 60% erkek ve 40% kadın varsa, train için de total tüm veri setinden 60% erkek ve 40% kadın alır ve totalleri 66% eder ve test için, 60% erkek ve 40% kadın alır ve totalleri 34% kadın eder.



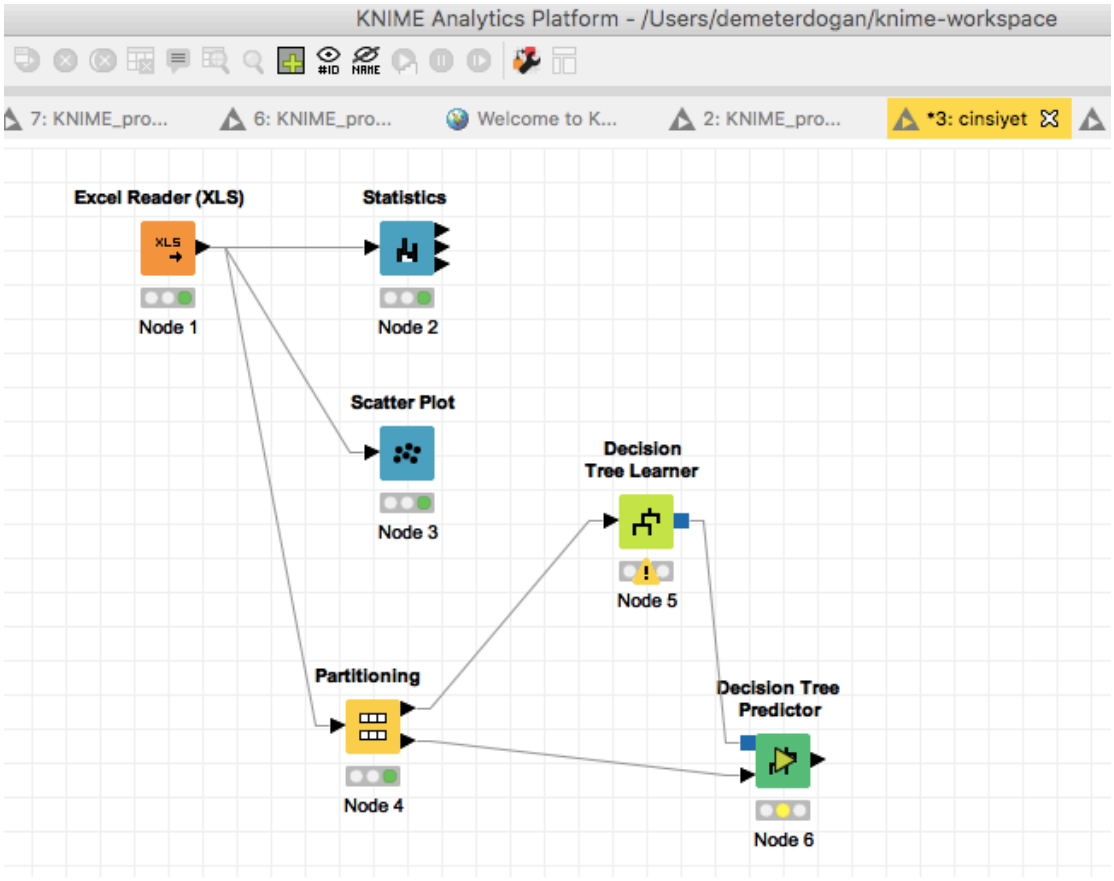
Şekil 2.1.18

Şekil 2.1.18, partitioning'te veri bölünmesinden sonra first partition veri setini göstermektedir. Veri setinin totalde 66% bu kadar veri etmektedir.



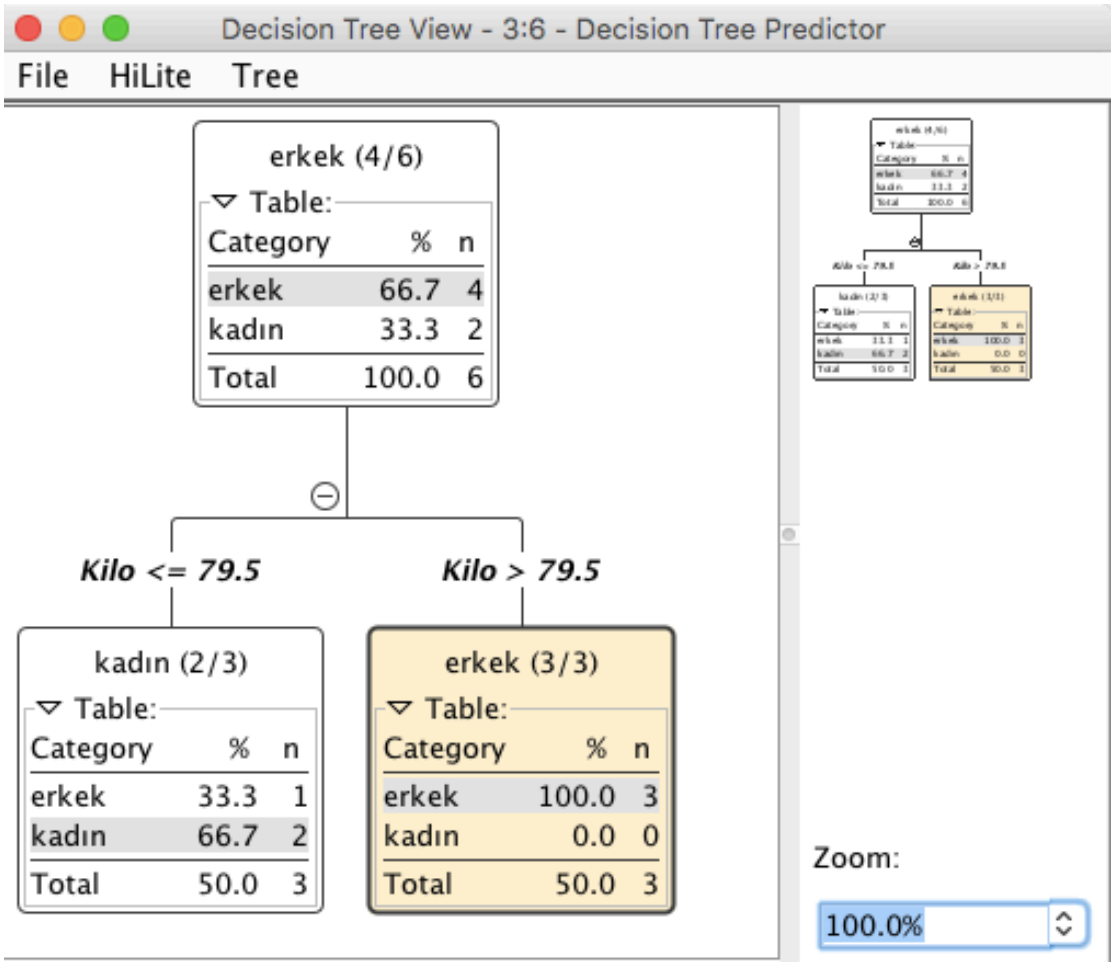
Şekil 2.1.19

Şekil 2.1.19, partitioning'de bölündükten sonraki second partition bölümünü göstermektedir. Üstteki veri kümesi ortalama tüm veri setinin 60% bu veri seti total tüm veri setine göre yaklaşık 40% oluşturmaktadır. Yani verilen orana göre bölündüğü buradan da anlaşılabilir. Makine öğrenme algoritmalarından decision tree kullanılacaktır. Öncelikle öğrenme süreci için learner sonrasında da test edilebilmesi için predictor sisteme eklenmelidir.



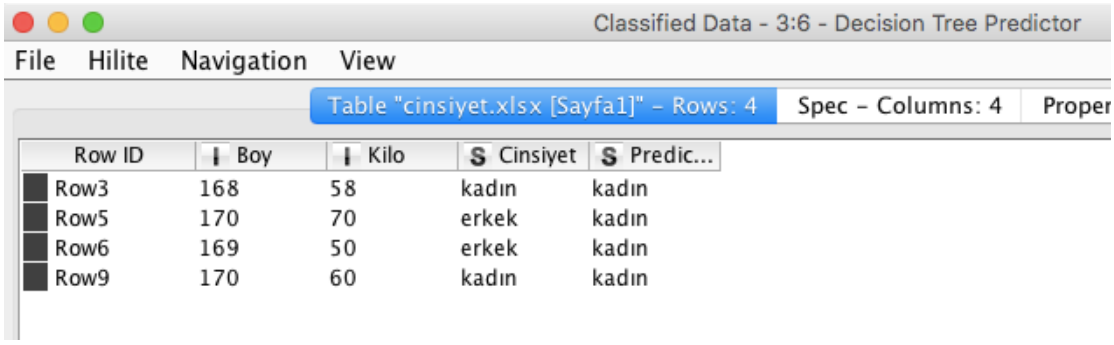
Şekil 2.1.20

Şekil 2.1.10 sisteme decision tree learner ve predictor eklenmesini ve bağlantılarını göstermektedir. Normalde node'ların (operatörlerin) yanlarında siyah üçgenler olur ve bunlar veri akışı olduğu anlamına gelmektedir. Decision tree learner yanında mavi kare bulunması onun modül aktaracağını göstermektedir. Kare'lerden anlaşıldığı gibi learner ile predictor arasında veri aktarması değil de modül aktarması yapılır.



Şekil 2.1.21

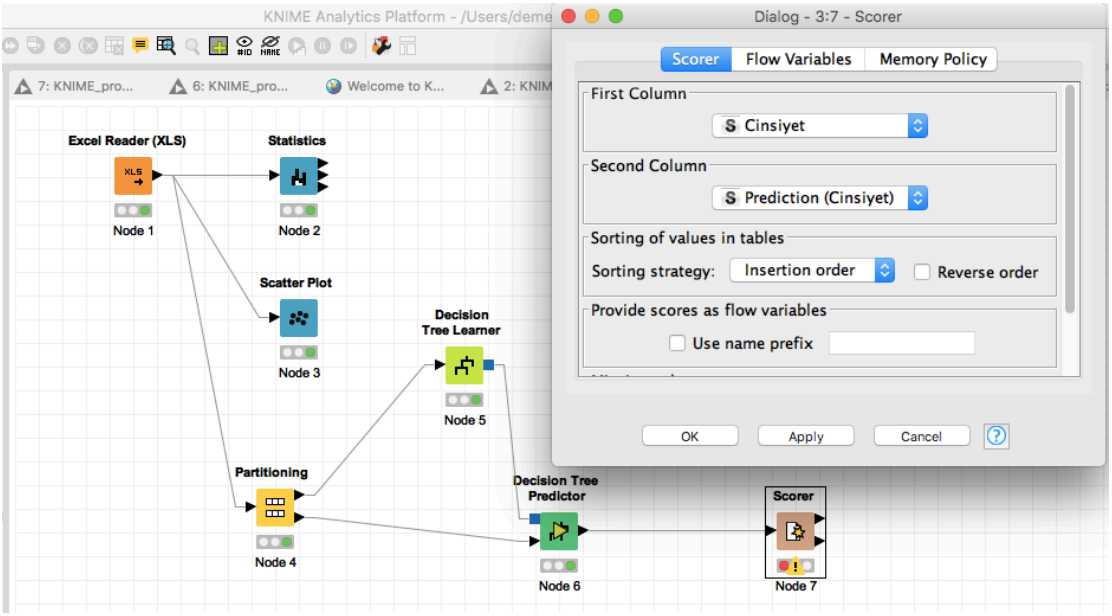
Şekil 2.1.21, decision tree view penceresini göstermektedir. Predicter operatörüne sağ tuşla tıklanarak bu pencereye ulaşılabilir. Erkek (4/6) demek 6 örnek içinde 4 ü erkek ve kiloya göre ayrım yapıldığında 79.5 kg'da az olanlar kadın 79.5kg dan fazla olanlar ise erkektir.



Şekil 2.1.22

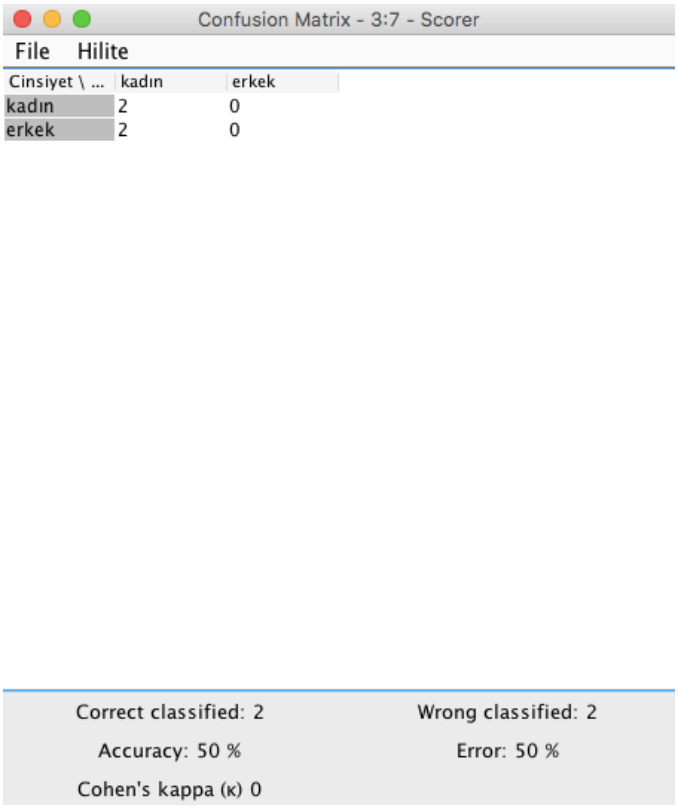
Şekil 2.1.22, yine predictor'a sağ tuşla tıknalanarak ulaşılan classified data penceresini göstermektedir. Görüldüğü üzere, boyu 168cm ve kilosu 58 kg olann kadını kadın olarak doğru tahmin etmiş, boyu 170 cm ve kilosu 70 kg olan erkeği kadın olarak yanlış tahmin etmiş ve boyu 169 cm ve kilosu 50 kg olan erkeği kadın olarak yanlış tahmin etmiştir.

Burada cinsiyet kolonu veri verilirken bilinen cinsiyetlerdir. Prediction kolonu ise makinenin tahmin ettiği cinsiyetlerdir. Bu örnekte veri setindeki veriler azdı fakat binlerce milyonlarca satır olan veri setleri kullanıldığında bu şekilde anlaşılması çok zordur. Bu yüzden test değerlendirmesi scorer düğümüyle (operatörüyle) yapılmaktadır.



Şekil 2.1.23

Şekil 2.1.23, sisteme scorer operatörünün eklenmesini ve configure penceresinde yapılan ayarlamayı göstermektedir. Veri setinde verilen cinsiyet ile tahmin edilen (prediction) cinsiyet kolonlarının karşılaştırılmasına imkan sağlar.

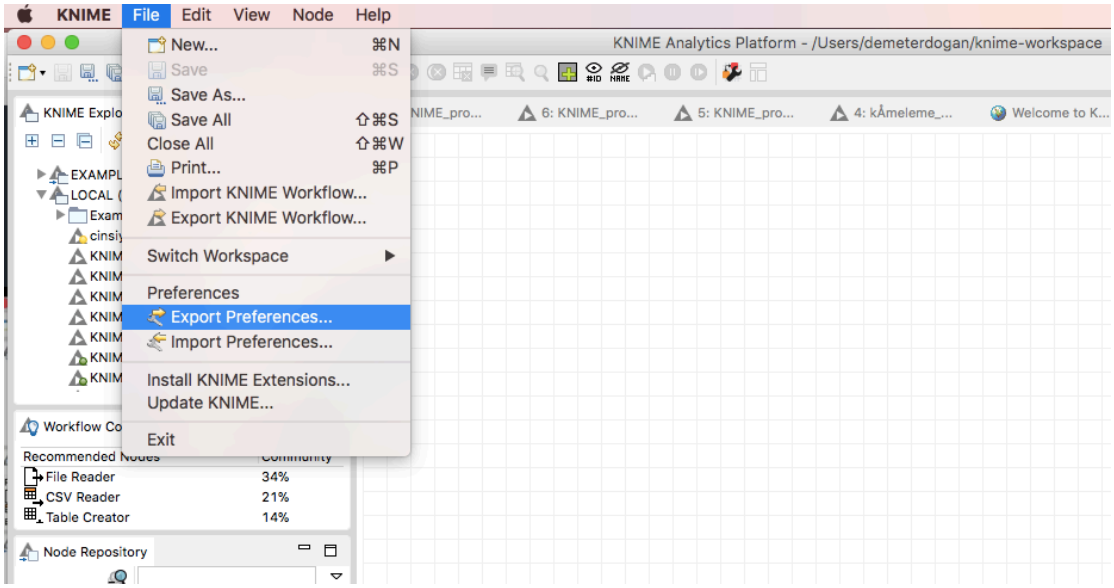


Şekil 2.1.24

Şekil 2.1.24'de confusion matrix görülmektedir. Confusion matrix'te diagon her zaman doğru tahminlerin yapıldığı sayıların yazıldığı yerdir. Örneğin kadın olup kadın tahmin edilen 2 kişidir. Erkek olup erkek tahmin edilen yoktur fakat erkek olup kadın tahmin edilen 2 kişidir. Bu yüzden başarı oranı 50% dir. Burada binominal veri bulunduğu için yani sadece kadın/ erkek tahmin edilmeye çalışıldığı için 2x2 matrix oluşturulmuştur.

2.2 Çalışmaları Kaydetme, Taşıma, Yükleme, Eklenti Kurma, Örnek Uygulamalara Erişim

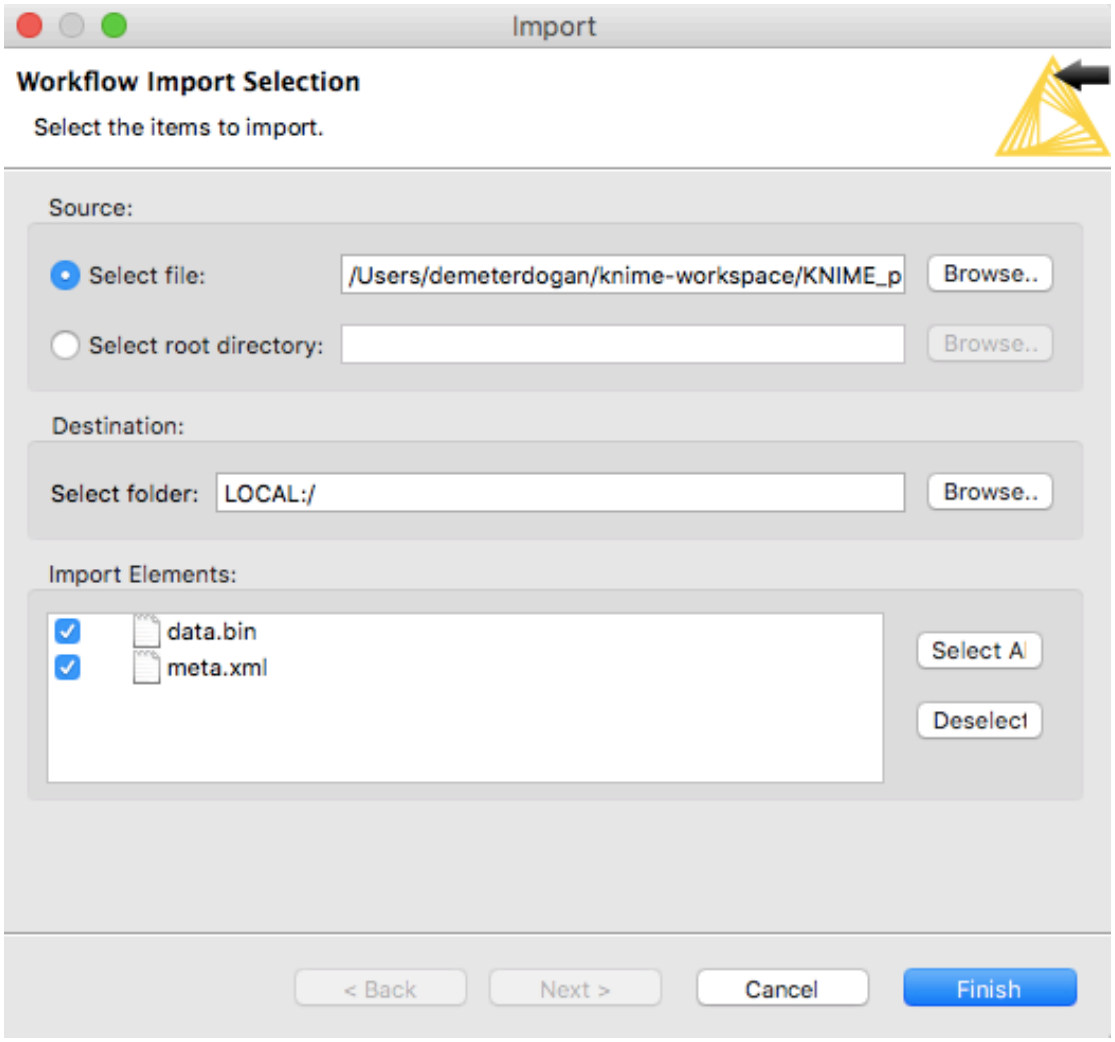
Bu bölümde Knime ortamında yapılmış çalışmaların nasıl kaydedileceği, nasıl geri yükleneceği ve gerekmesi durumunda Knime'a nasıl eklenti yükleneceği gösterilecektir.



Şekil 2.2.1

Şekil 2.2.1, Dosya (File) menüsüne girildiğinde Import ve export Knime workflow seçenekleri bulunmaktadır. Workflow, Knime'da kullanılan ve çalışma akışı anlamına gelen bir kelimedir. Bazı programlarda process vb. Kelimeler kullanılmaktadır.

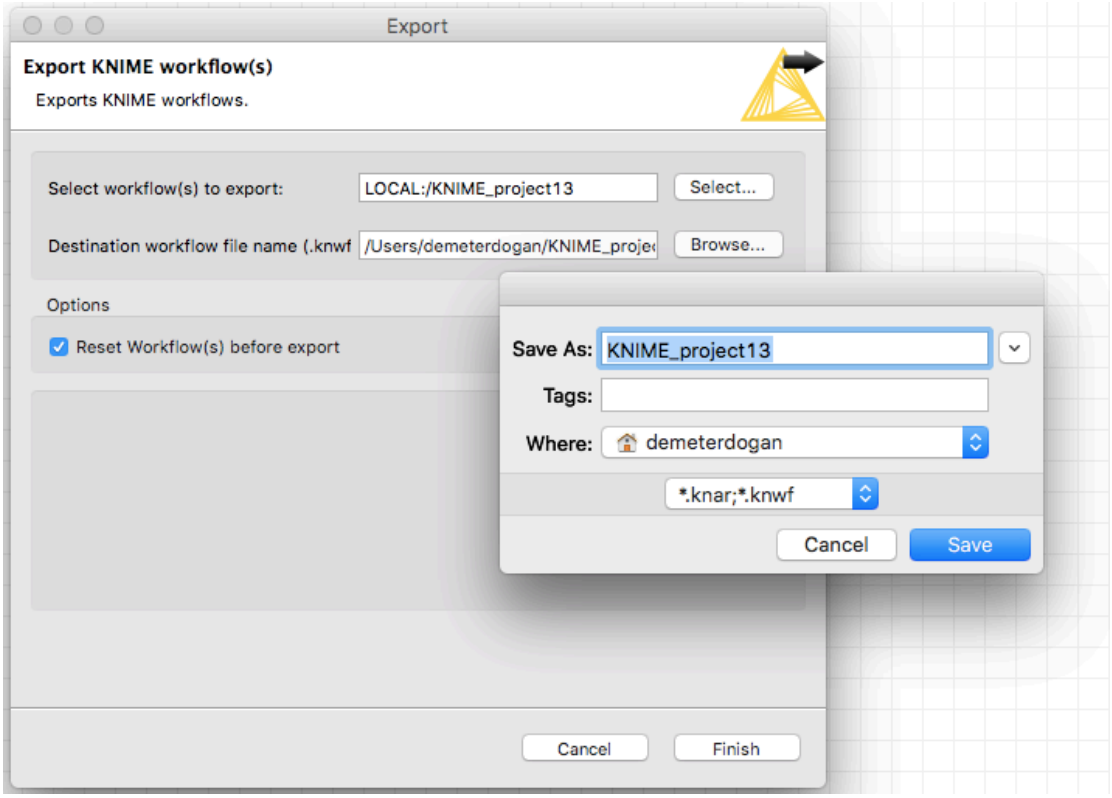
Import Knime Workflow: daha önce kaydedilmiş herhangi bir workflow'un (yapılmış bir çalışma akışının) nasıl tekrardan üzerinde çalışmak için Knime'da yeni bir workflow gibi çağırılacağı yerin seçileceği seçeneğin olduğu yerdir.



Şekil 2.2.2

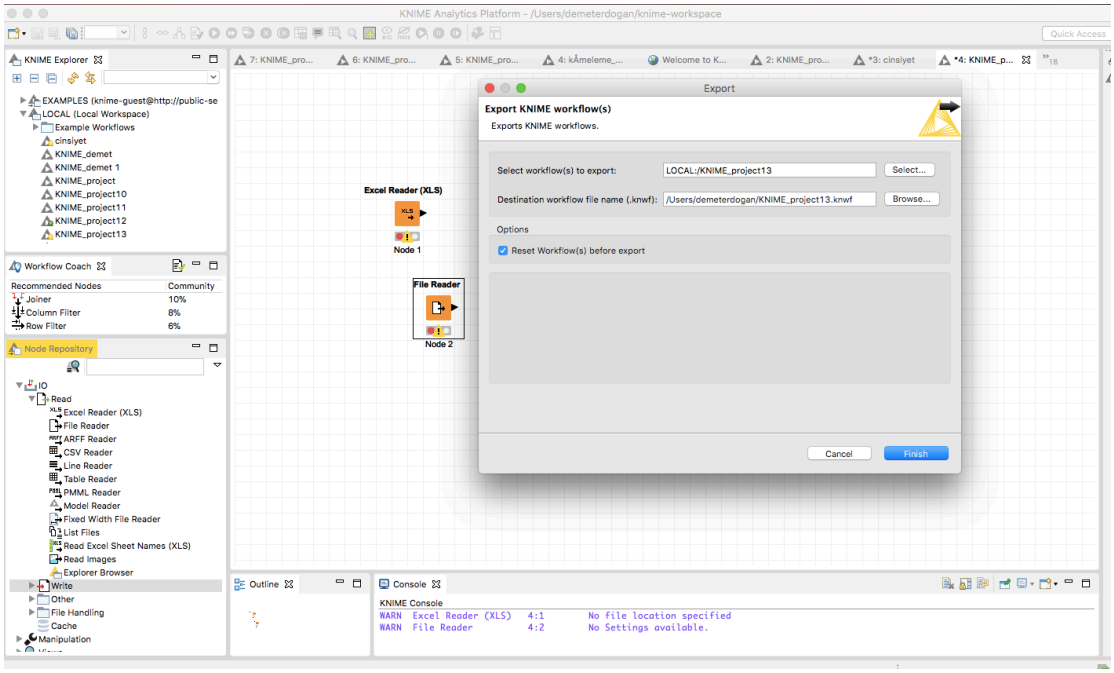
Şekil 2.2.2, import Knime workflow seçeneğinin penceresini göstermektedir. Select file'da browse butonuna basılarak bölümünden hangi workflow tekrardan çalışılacaksa o bölüm seçilir. Select folder'da browse'a basınca Knime explorer bölümü açlır ve daha önceki workflow'lardan biri seçilebilir yer olarak.

Export Knıme Workflow: workflow'un export (dışa aktarılacağı) edileceği yerin belirleneceği seçenektir



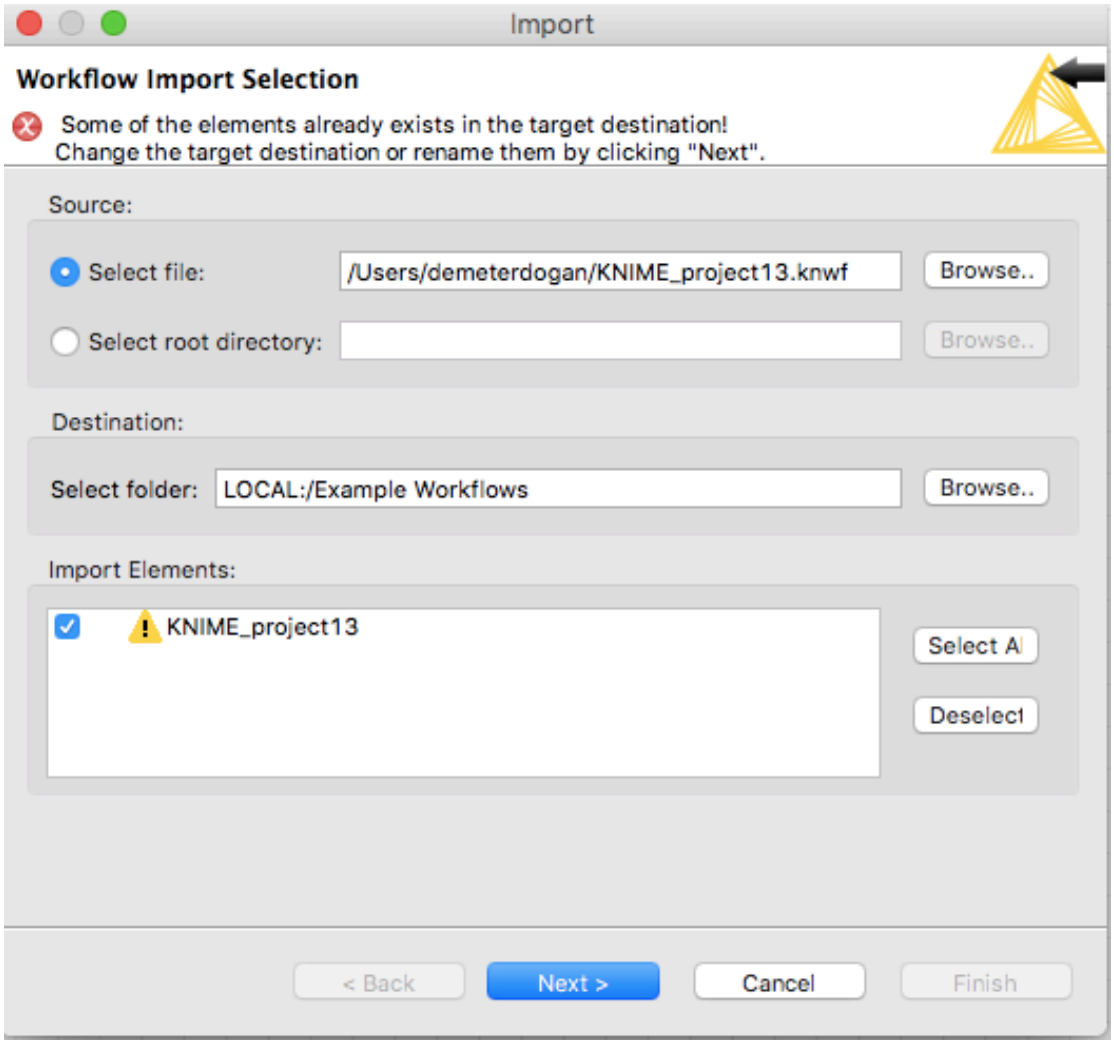
Şekil 2.2.3

Şekil 2.2.3'de görüldüğü gibi öncelikle dışa aktarmak istenen workflow "select workflow(s) to export"yazınının yanındaki browse butonuna basılarak seçilir daha sonra bunun "destination workflow file name" yanındaki browse butonuna basılarak nereye kaydedileceği seçilir. Destination'daki browse'a basıldığında şekilde görülen ufak pencere açılır ve dosya ismi ne istenirse tekrardan yazılır sonra da "where" kısmından bilgisayarda istenilen herhangi bir yere kaydedilebilir.



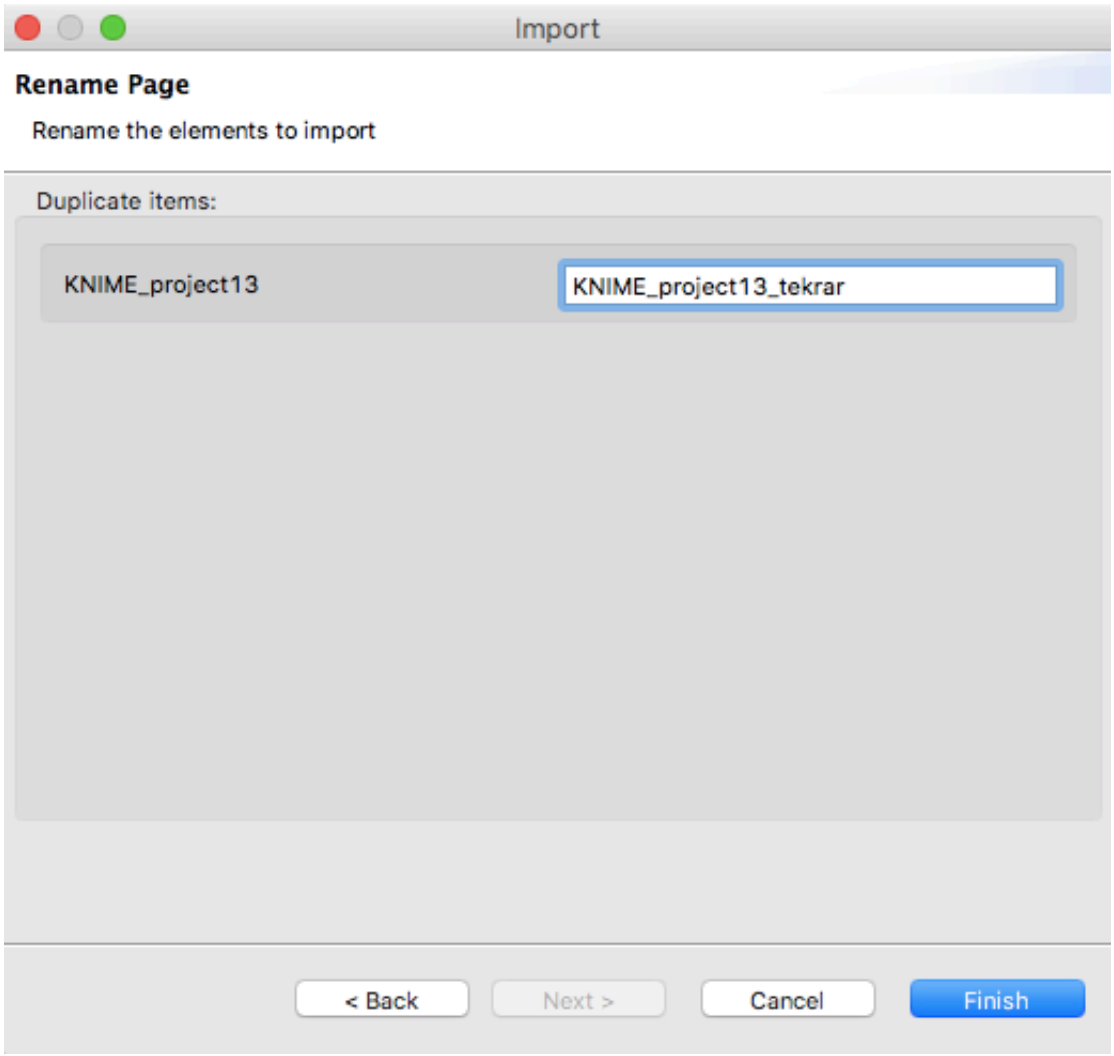
Şekil 2.2.4

Şekil 2.2.4, örnek olması açısından export edilmek üzere rastgele bir project13 isminde workflow oluşturulmuş ve export seçeneğinde şekilde görünen yer ve isim seçilmiştir. Bilgisayarda istenilen yere kaydedilen bu (export edilen) workflow başka yerlerde kullanılmak, flash'a kaydedilmek, başka şekilde bilgisayardan alınmak üzere istenilen şekilde kullanılabilir.



Şekil 2.2.5

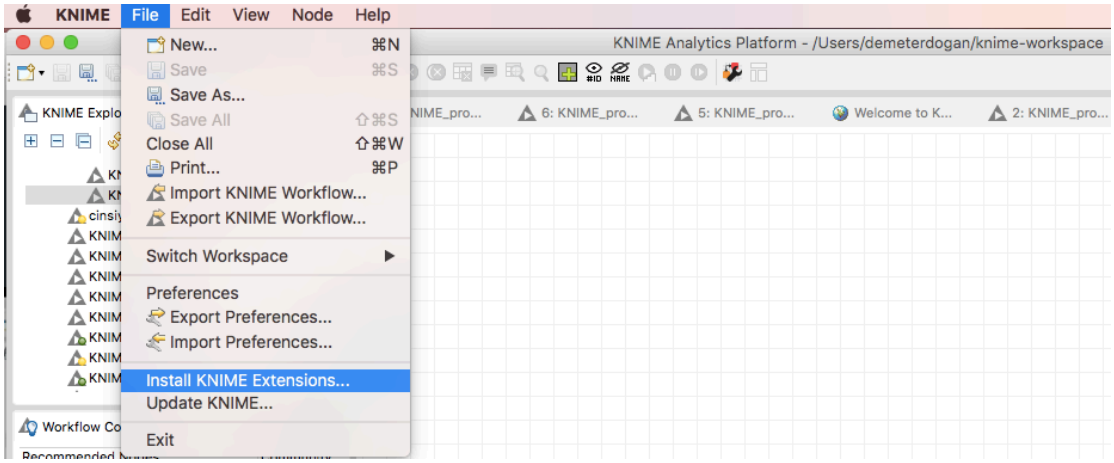
Şekil 2.2.5, örnek olması açısından import edilmek üzere bir önceki şekilde export edilen workflow'un kaydedildiği yerden seçilmesini ve alınan isim tekrarı hatasını göstermektedir. Bir önceki şekilde project13 şeklinde kaydedilen workflow olduğu için tekrardan aynı isimli dosyanın açılmasını kabul edilmez ve import edilen workflow için yeni isim verilmek üzere aşağıdaki pencere açılır.



Şekil 2.2.6

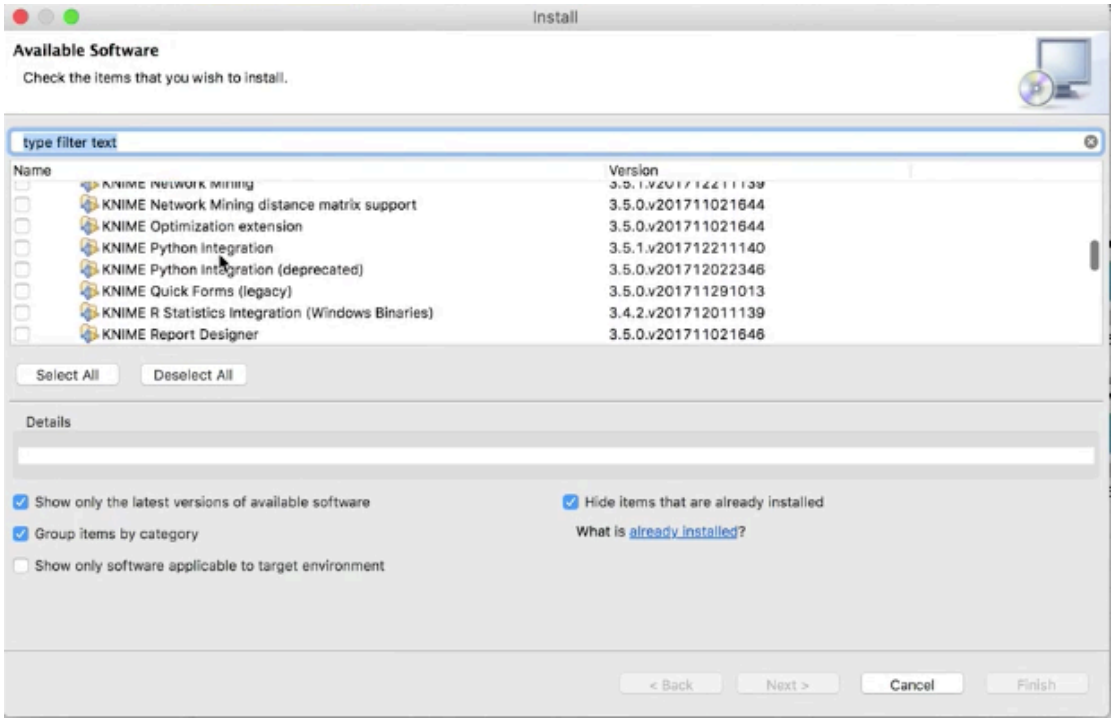
Şekil 2.2.6, import edilen workflow'un isminin daha önceden de kaydedilmiş bir workflow'da kullanılması üzerine başka isim ile sisteme aktarılması istenmektedir. Burada örnek olması açısından KNIME_project13_tekrar ismi verilerek finish tuşuna basıldı. Ve workflow bu isim ile import edilmiş oldu.

Diğer bir özellik de eklentilerdir (extensions). Knime çok sayıda eklenti içermektedir.



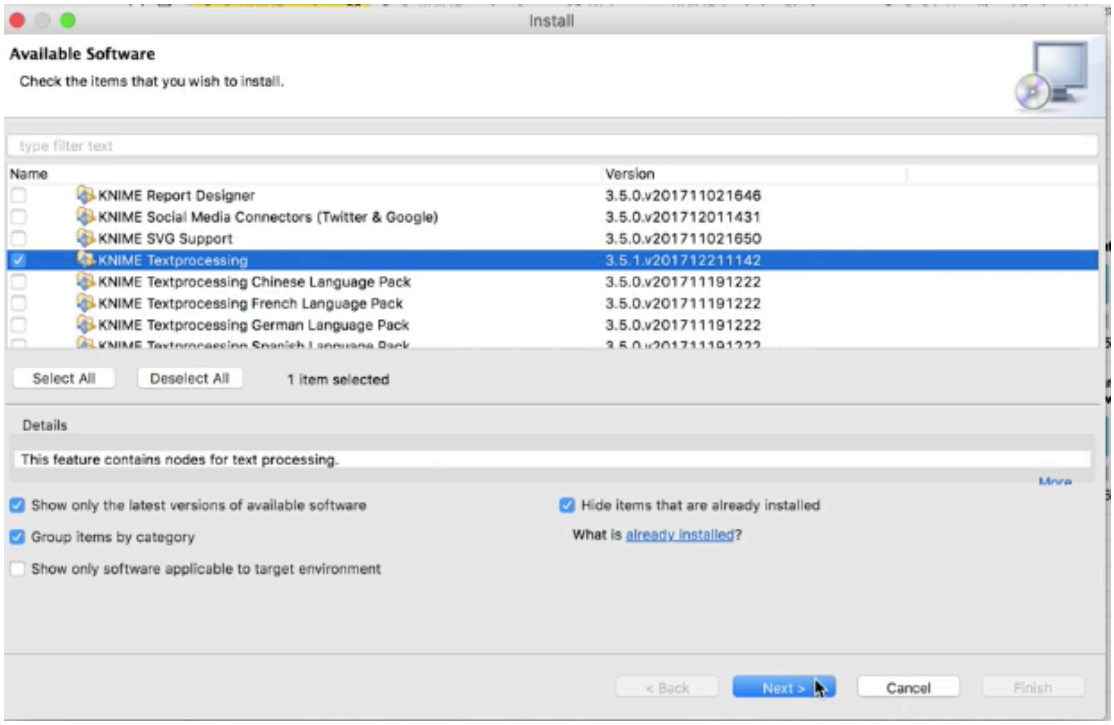
Şekil 2.2.7

Şekil 2.2.7, extensions seçeneğinin Knime'da açılabileceği pencerenin yerini göstermektedir. Buradan extensions install (Knime'a indirmek) edilebilir.



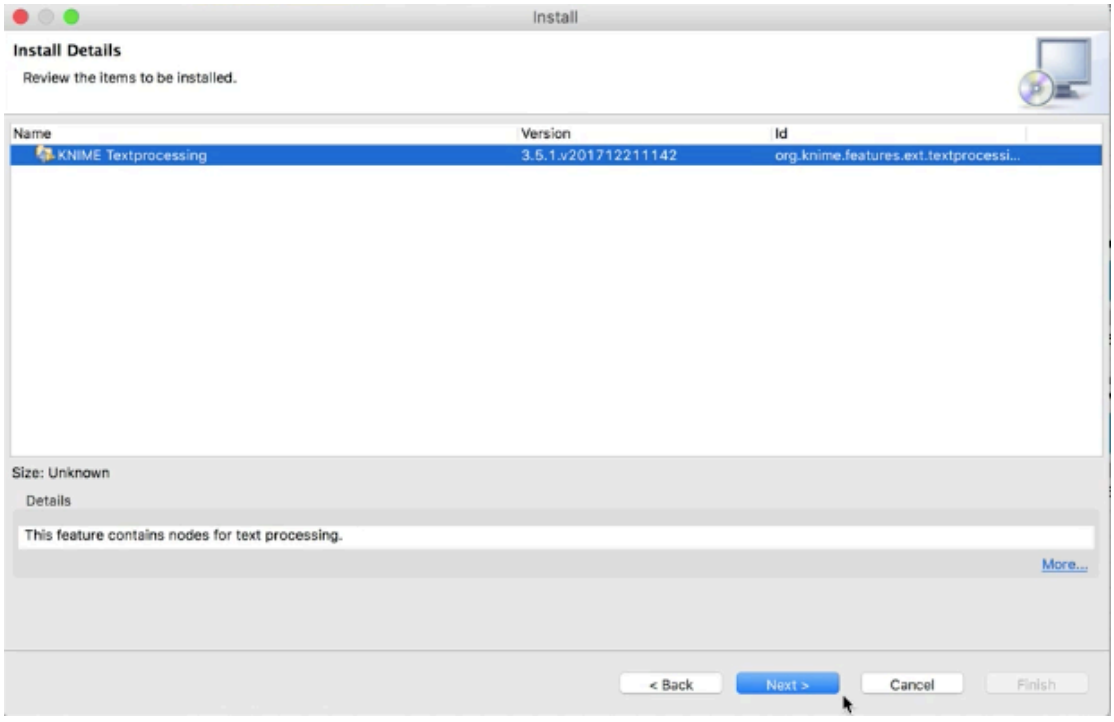
Şekil 2.2.8

Şekil 2.2.8, istenilen extensions listeden seçilerek install edilebilir. Bu liste default olarak knime ile gelir fakat hazır kurulu olmamasının nedeni çok yer kaplayacak olmalarıdır. Bu yüzden ihtiyaç duyulan alana, bölüme veya istenilen konuya göre eklenti (extensions) install edilebilir. Python, R, bioinformatics, cheminformatics vb. Spesifik alanlara yönelik bir çok eklenti bulunmaktadır. Burada Knime'ın kendi extension'ları listelenmektedir ama bu listede olmayan eklentiler üçüncü parti yerlerden de seçilerek install edilebilir.



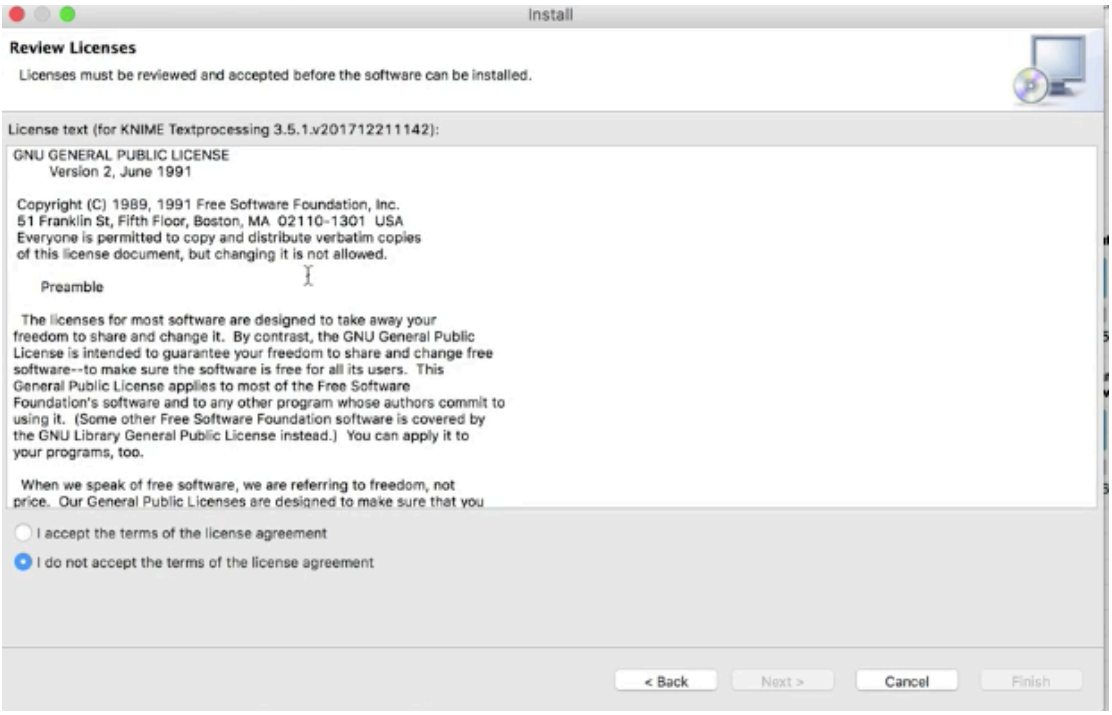
Şekil 2.2.9

Şekil 2.2.9'da örnek olması açısından textprocessing extensions'ı seçilmiş ve install için next tuşuna basılmıştır. Textprocessing, facebook, twitter vb. Programların verileri yani akan yazılar işlenmek isternirse bu eklenti kullanılabilir.



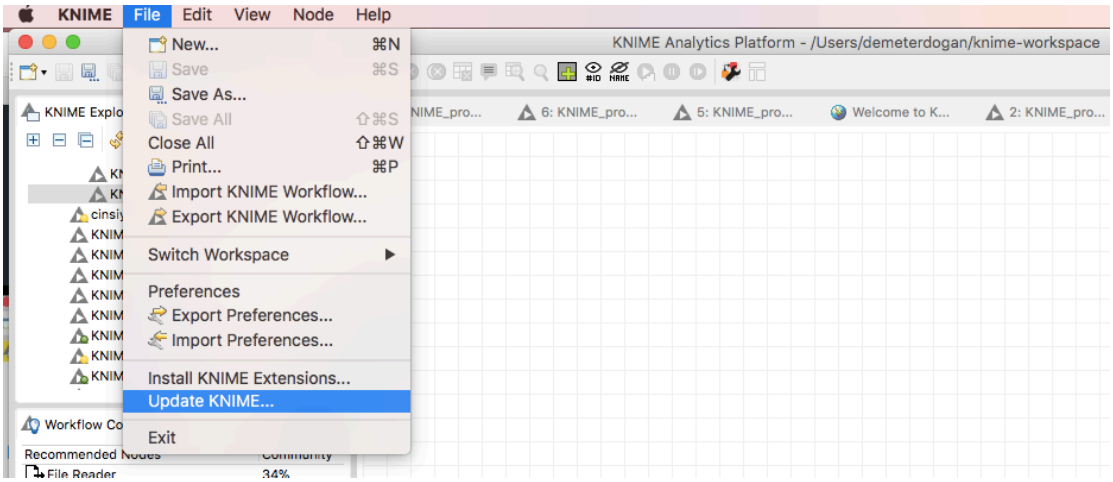
Şekil 2.2.10

Şekil 2.2.10, bir önceki şekilde next dendikten sonra açılan pencereyi göstermektedir. Install edilmek için işaretlenmiş eklentiler bu açılan yeni pencerede tekrardan gösterilir. Eğer istenilen eklentiler seçilmişse next tuşuna basılır.



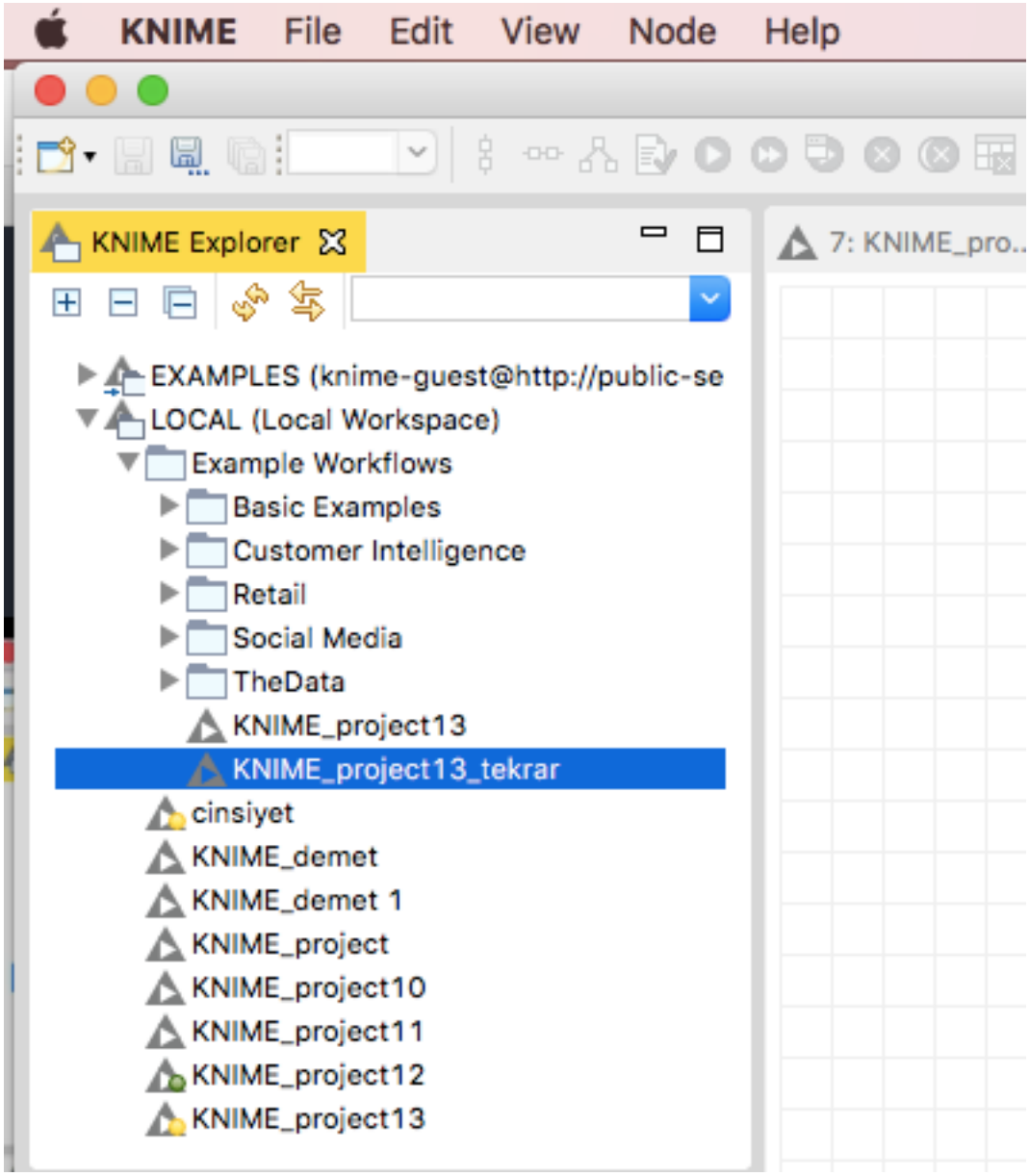
Şekil 2.2.11

Şekil 2.2.11, burada install edilmek istenen eklentiler için "I except the terms of the license agreement" seçeneği seçilerek next tuşuna basılmalıdır. Bu işlemden sonra bir kaç dakika install etme işlemi sürebilir. Install işlemi de bittikten sonra program tekrar çalıştırılması için pencere açılacaktır. O da kabul edildikten sonra Knime yeniden başlatılır (restart) ve install edilen extensions node repository pencereinde **KNIME Labs** bölümüne gelir. Daha önce install edilenler de burada depolanır.



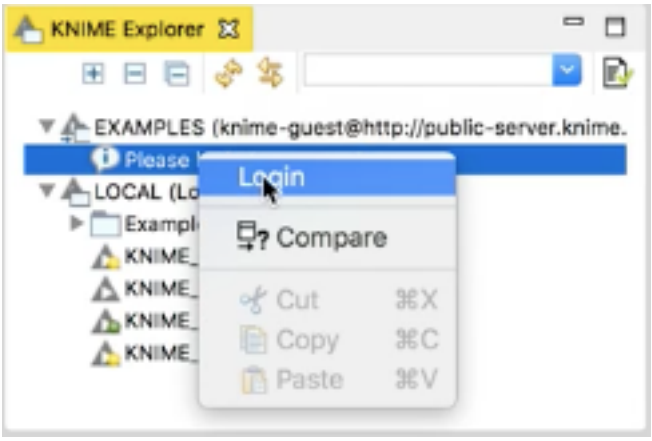
Şekil 2.2.12

Şekil 2.2.12, **Update KNIME** özelliğinin olduğu pencerenin yerini göstermektedir. Bu seçilirse Knime'ın son versiyonu update edilir. Eğer yeni bir versiyon henüz çıkmamışsa bu açılan yeni pencerede belirtilir.



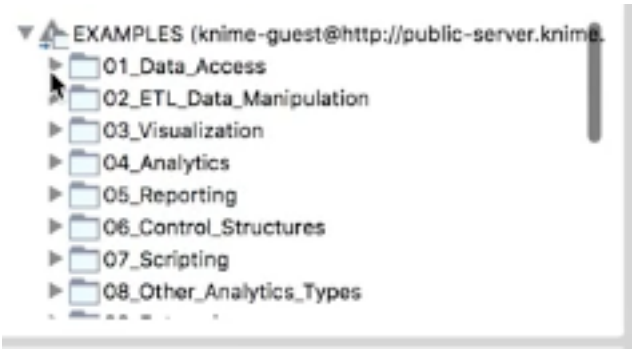
Şekil 2.2.13

Şekil 2.2.13, Knime'ın içinde bulunan örneklerin Knime explorer penceresindeki yeri gösterilmektedir. Bu örnekler belli bir düzene, hıza, bilgi birikimin olması gerektiğine göre dizilmiştir.



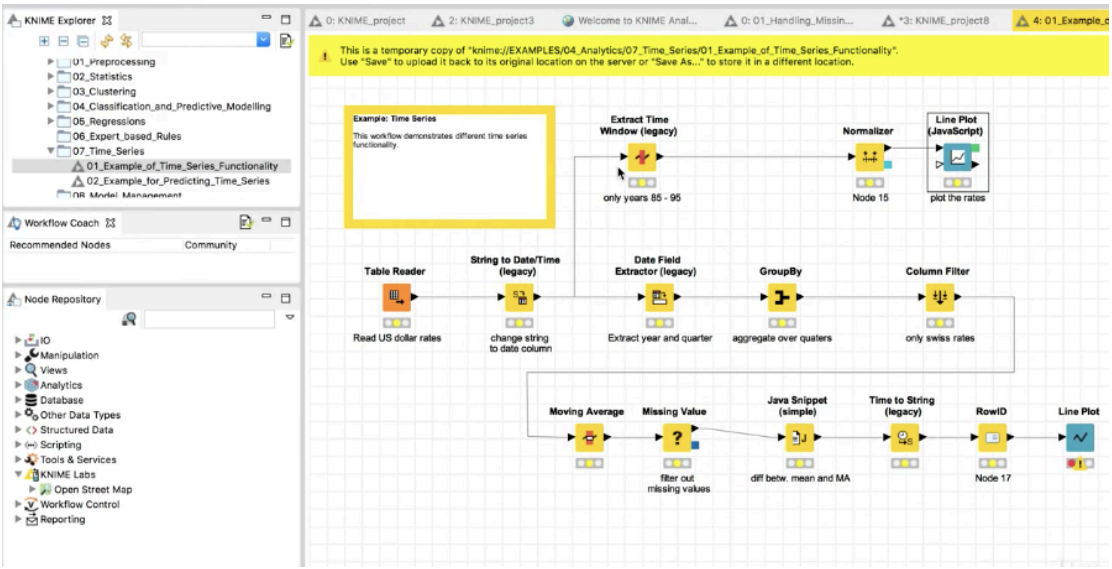
Şekil 2.2.14

Şekil 2.2.14'de de görüldüğü gibi examples seçeneği seçildiğinde Knime'a login olunması istenmektedir.



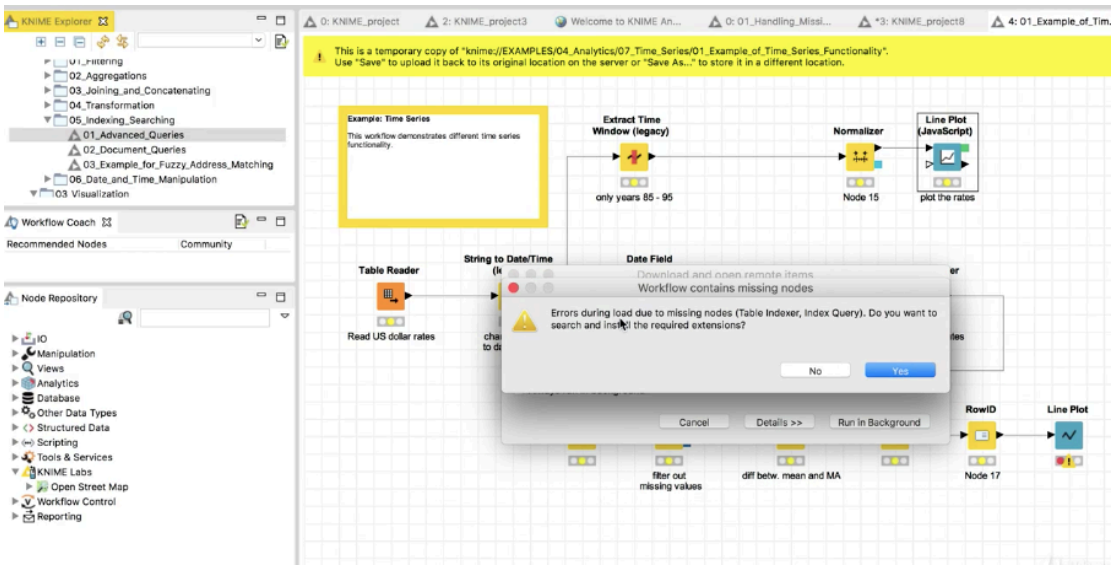
Şekil 2.2.15

Şekil 2.2.15, bir önceki şekilde gösterilen yerde login olunduktan sonra açılan examples listesini göstermektedir. Bu kitapta da buradaki listeye bağlı kalınarak örnekler, açıklamalar yapılacaktır.



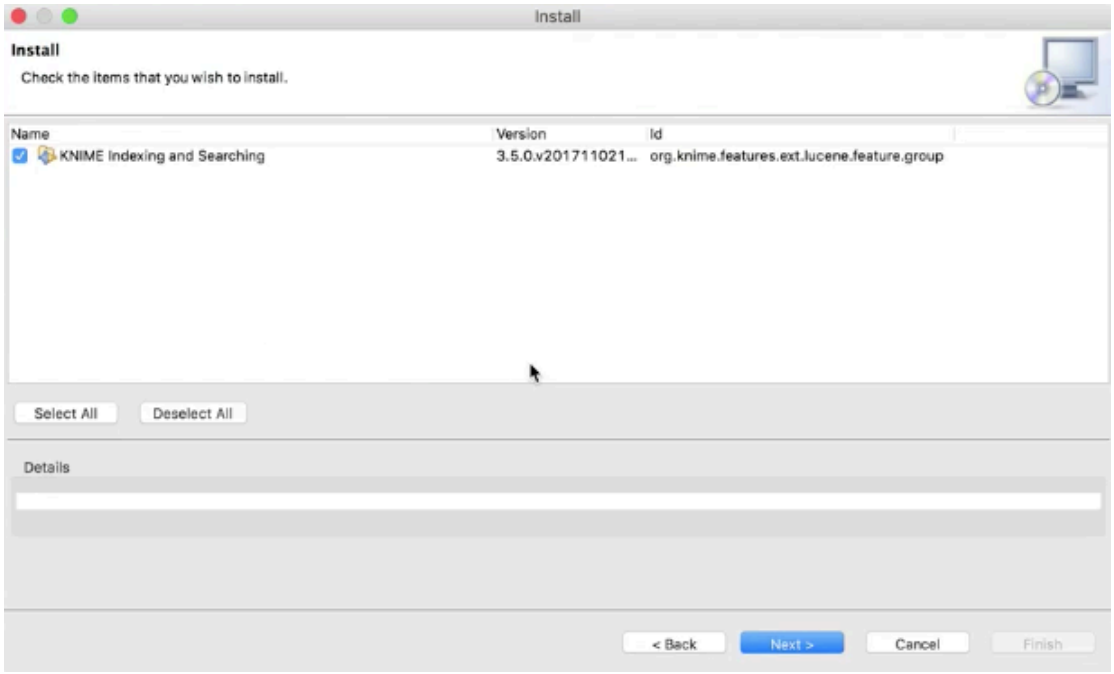
Şekil 2.2.16

Şekil 2.2.16, seçilen bir example'a (örneğe) çift tıklandıktan sonra açılan pencereyi göstermektedir. Burada örnek olması açısından time series örneğine tıklanmıştır. Workflow penceresine uygulamaların sırası, bağlantıları otomatik olarak gelmiştir.



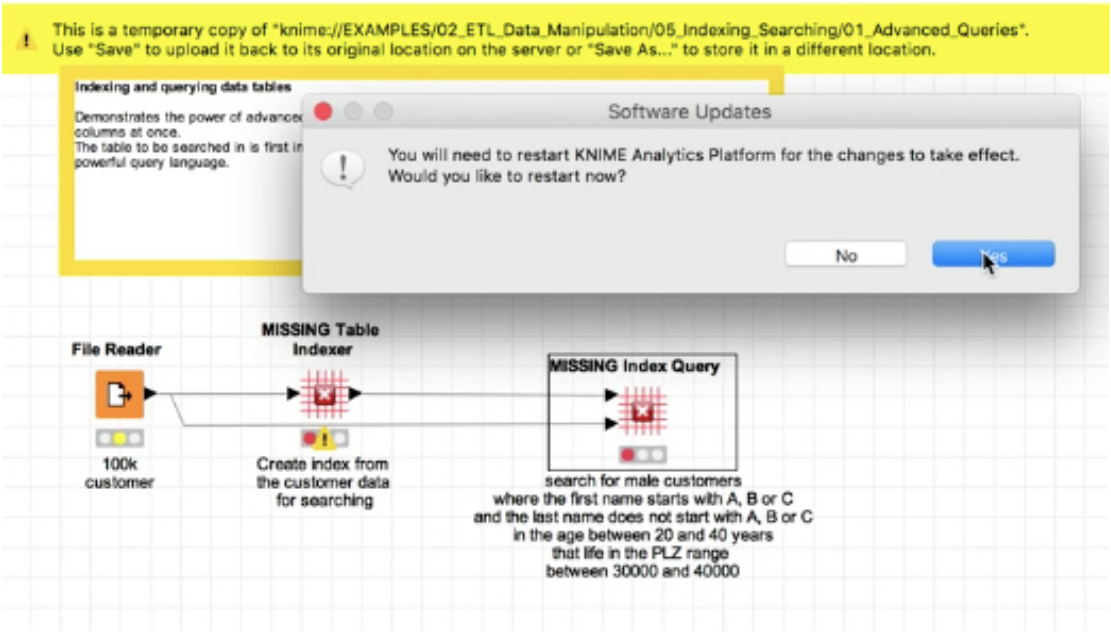
Şekil 2.2.17

Şekil 2.2.17, seçilen başka bir örneğin extension uyarı penceresini göstermektedir. Eğer seçilen örnek için ek eklenti gerekiyorsa install edilmelidir. Açılan pencerede yes tuşuna basılarak otomatik olarak install bölümüne yönlendirilir.



Şekil 2.2.18

Şekil 2.2.18, otomatik yönlendirmeden sonra açılan pencereyi göstermektedir. Next tuşuna basılarak ilerlenmelidir.

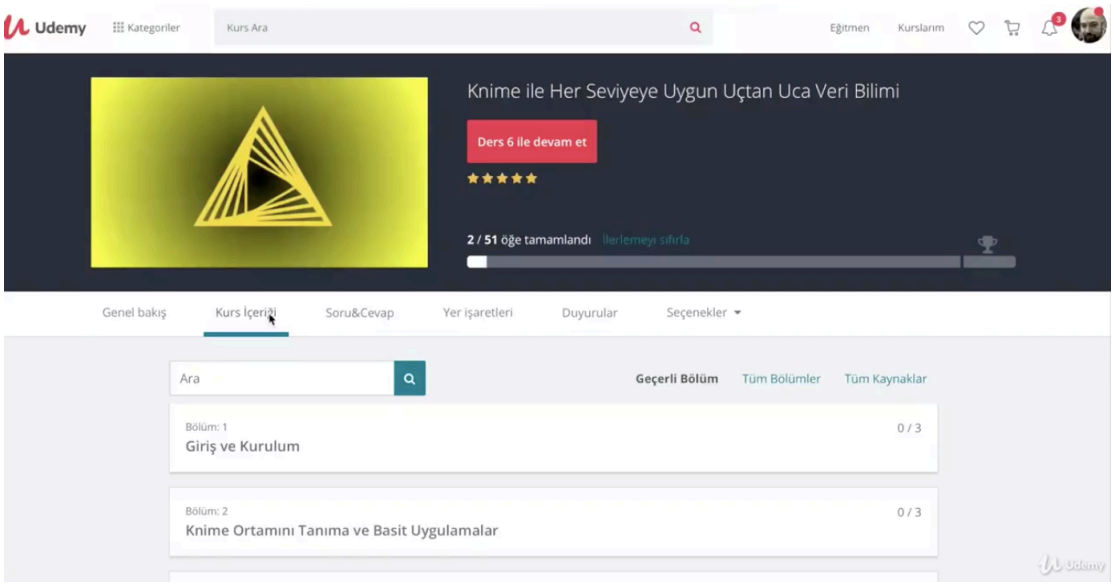


Şekil 2.2.19

Şekil 2.2.19, next tuşuna basıldıktan sonra daha önceki bölümde gösterilen accept penceresinde anlaşmanın kabul edilmesinden sonra install edilen extension kullanılarak otomatik olarak seçilen örnek açılır. Burada da görüldüğü gibi programın yeniden başlatılması sorulacaktır.

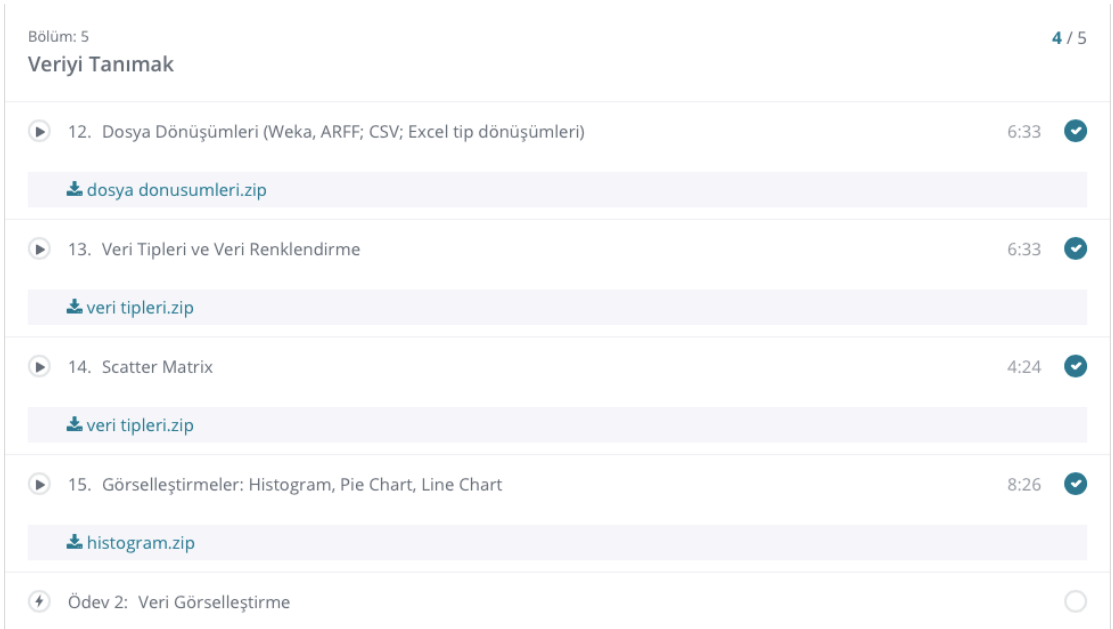
2.3 Ders Ortamındaki Hazır Projelerin Kullanılması

Bu bölümde dersin/kitabın kullanımı ve udemy.com'daki videolar ile ilgili konulardan bahsedilecektir.



Şekil 2.3.1

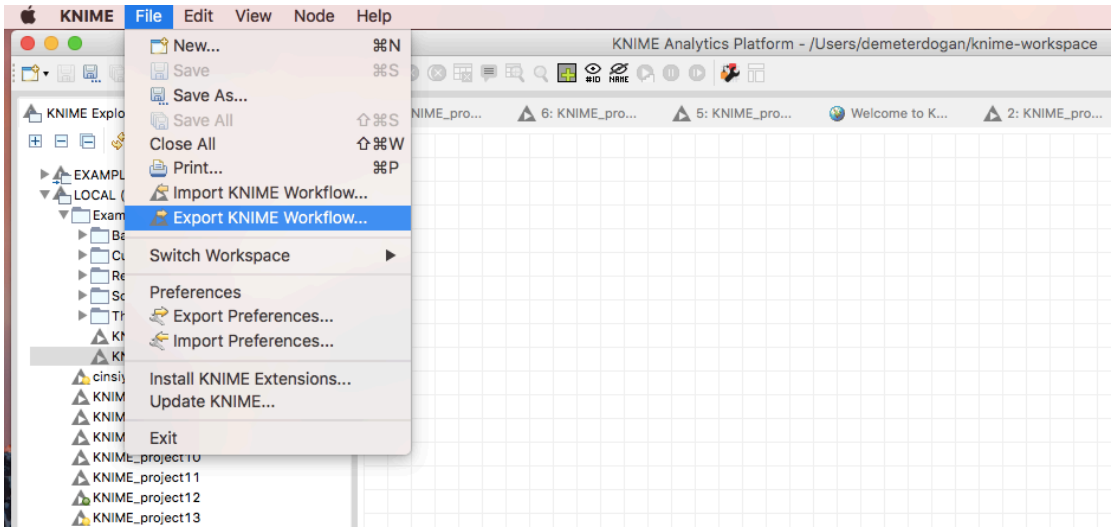
Şekil 2.3.1, bu kitabın tüm bölümlerinin video halinin yüklü olduğu udemy.com'daki "Knime ile Her Seviyeye Ugun Uçtan Uca Veri Bilimi" dersinin ekranını göstermektedir.



Şekil 2.3.2

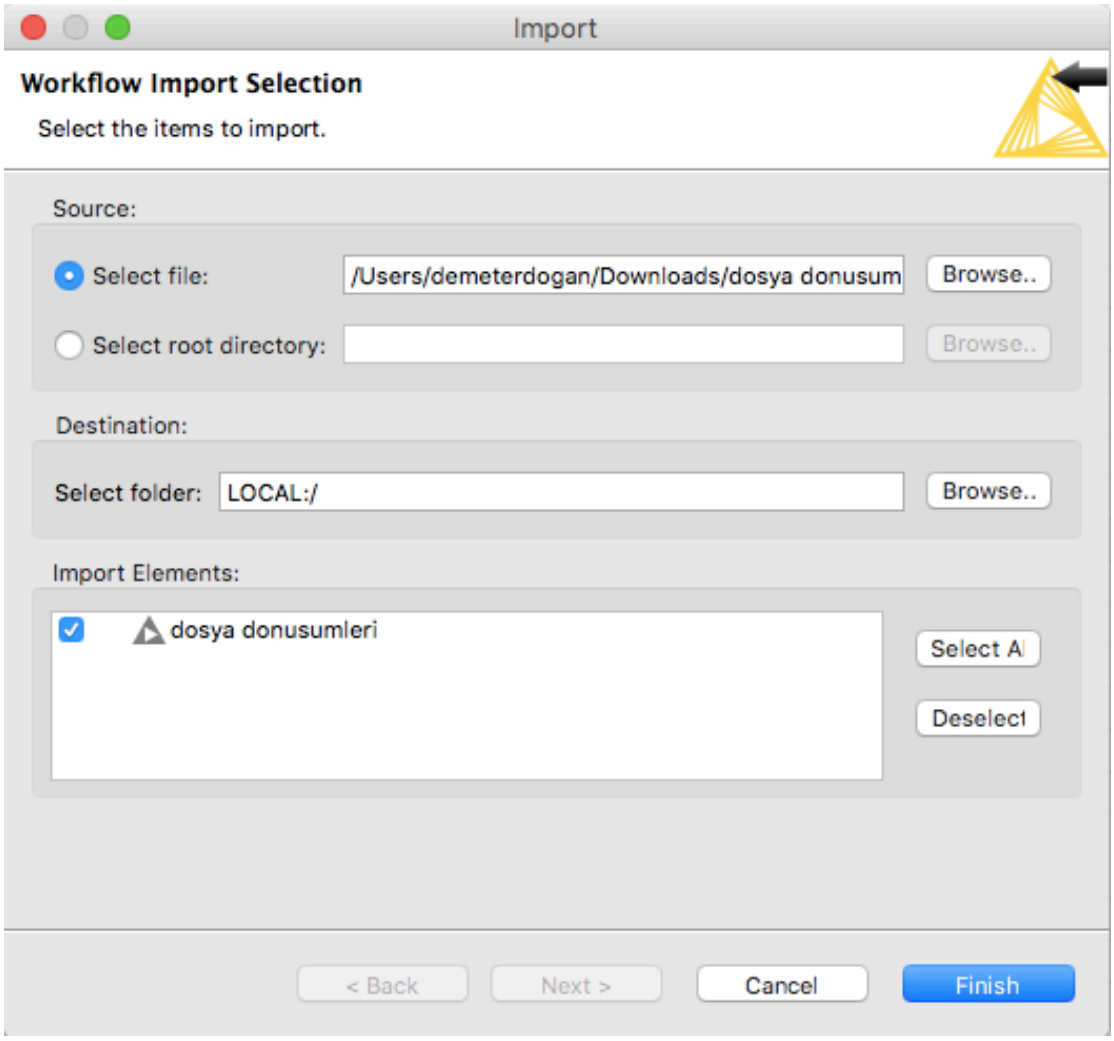
Şekil 2.3.2, udemy'deki ders bölümlerinden bir örnek göstermektedir. Dersler konularına göre bölümlere ayrılmış durumda ve birçok videonun altında workflow'ları zip formatında yüklenmiş durumdadır. Bu dosyalar indirilerek direk kullanılabilir.

Video'da bu workflow'ların oluşturulması detaylı biçimde anlatılmaktadır. Bu yüzden ister indirilip sadece deneme yapılabilir istenirse de baştan workflow oluşturulup video ile eş zamanlı denenebilir. Video serisinden bir workflow zip formatıyla indirilip Knime'da açılması gösterilecektir.



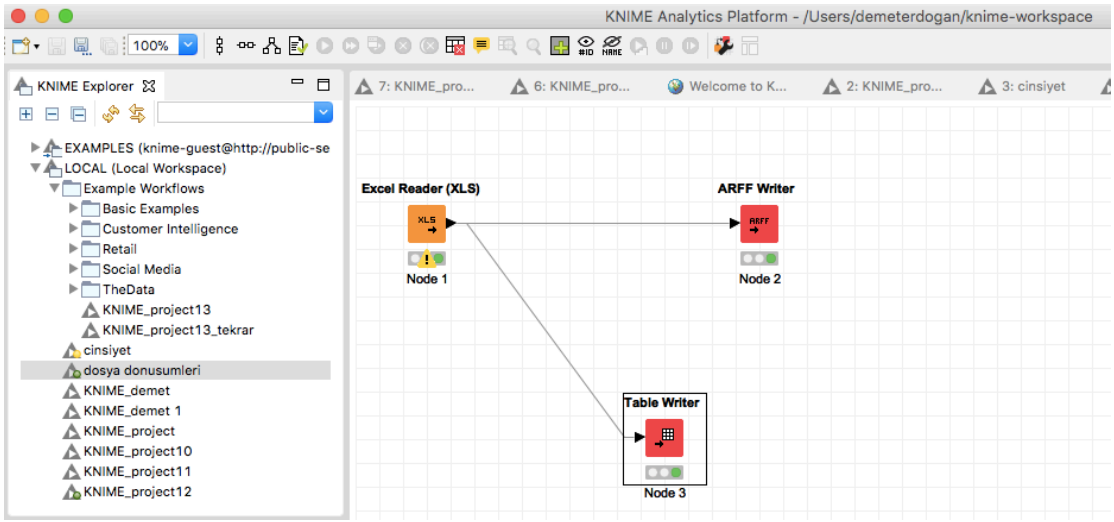
Şekil 2.3.3

Şekil 2.3.3, indirilen hazır bir workflow'un sisteme import edileceği pencereyi göstermektedir. Import KNIME Workflow seçeneği seçildikten sonra aşağıdaki pencere açılır.



Şekil 2.3.4

Şekil 2.3.4, **select file** bölümünden dosyanın indirildiği yerden browse (seçilir). Daha sonra finish tuşuna basılır.



Şekil 2.3.5

İndirilen dosya local workspace bölümüne gelir ve buradaki dosyaya çift tıklayınca çalışma penceresinde otomatik açılır. Açılan dosya Şekil 2.3.5'de görülmektedir. Her operatörün configure bölümlerinden seçenekler değiştirilebilir yani üzerinde oynama yapılabilmektedir. Eğer versiyon farklılığı varsa uyarı penceresi açılır ve güncelleme yapılması sorulur. O pencereden onaylanırsa güncelleme yapılır ve sonrasında workflow açılır. Yüklenen dosyada sorun oluşursa (örnek csv, excel dosyaları) onlar configure bölümünden tekrardan yüklenebilir. Bu kitapta ve video serisinde cinsiyet, iris veri setleri kullanıldı. İris internetten indirilebilir, cinsiyet veri seti de kolayca oluşturulabilecek bir veri setidir.

3.BÖLÜM: VERİ BİLİMİ YÖNTEMLERİ

3.1 Veri Bilimi Yöntemlerine Giriş ve SEMMA

Bu bölümde veri bilimi yöntemlerine giriş yapılacaktır. Aslında literatürde bilinen en fazla kullanılan 3 tip yöntem bulunmaktadır. SEMMA, CRISP-DM ve KDD. Bu yöntemlere sırasıyla değinilecektir. Bir veri bilimi projesinin nasıl yönetileceği ifade edilir. Cevaplanmaya çalışılacak sorular şunlardır:

- Bir veri bilimi projesine nereden başlanacak?
- Hangi adımlar atılacak?
- Ve sonunda nerede bitecek?

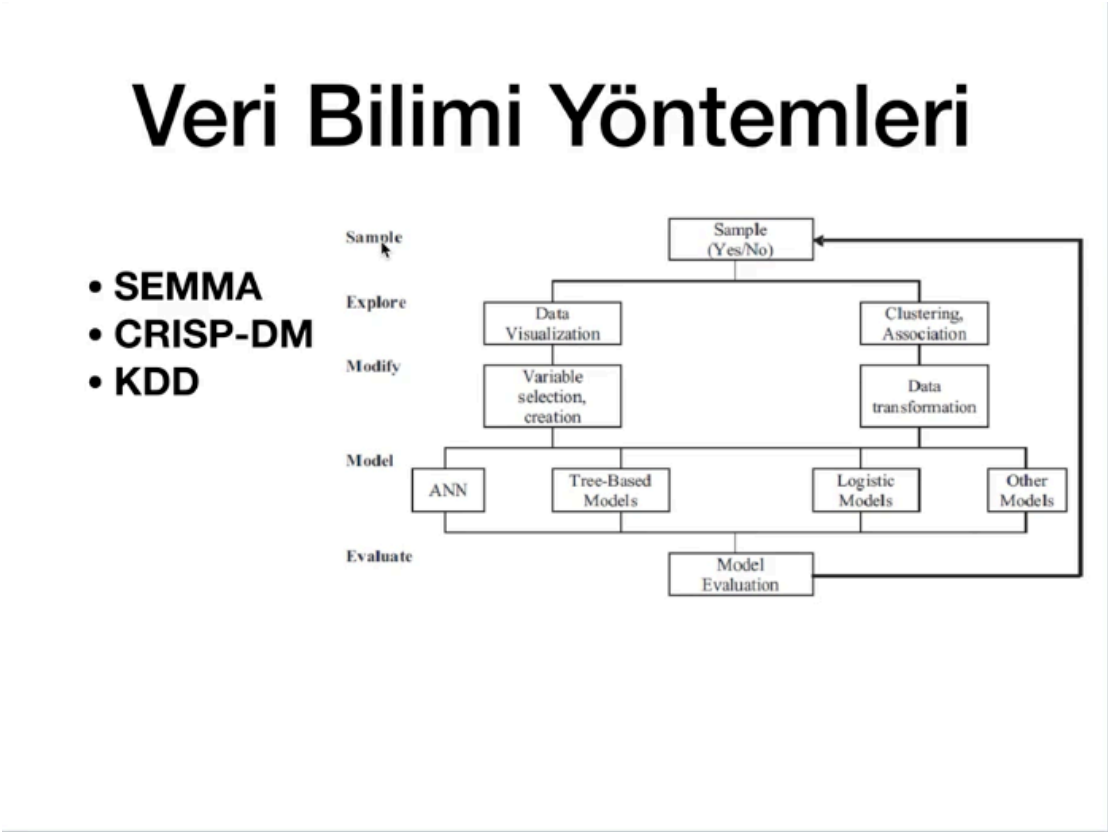
İlk açıklanacak yöntem Semma Yöntemi.

SEMMA YÖNTEMİ

- Sample(Örnekleme),
- Explore(Keşfetme),
- Modify(Dönüştürme),
- Model (makina öğrenmesi Modeli oluşturma) ve
- Evaluate(Değerlendirme)

kelimelerinin baş harflerinden oluşmaktadır. Ve SAS firması tarafından ortaya çıkarılan bir yöntemdir. Şu an modası geçmiş denilebilir. Hepsi aşağı yukarı aynı şeyleri anlatmakla birlikte CRISP-DM ve KDD biraz daha öndedir. Hatta CRISP-DM şu an en çok kullanılan modellerden biridir.

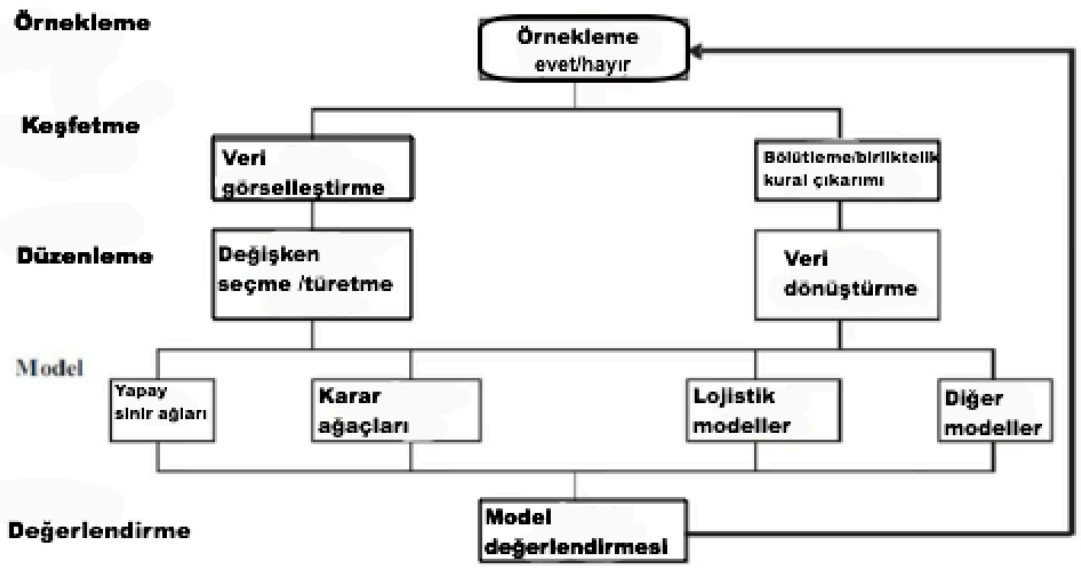
SEMMA, SAS firması tarafından ilk kez ortaya çıkarılmış bir yöntemdir. Uzun yıllar kullanıldıktan sonra artık modası biraz geçmiş bir yöntemdir.



Şekil 3.1.1

Şekil 3.1.1'de görülen şemanın 3.1.2'de Türkçe'ye çevirilmiş hali verilmiştir.

SEMMA yöntemi örnekleme(sample) ile başlamaktadır. Örnekleme evet/hayır cevabına göre ilerlemektedir. Şayet evetse bir veri görselleştirme (data visulization) , hayırsa o zaman bir birliktelik kural çıkarımı veya bir bölütleme (clustering/association rule mining) işlemi yapılacak demektir. Burada aslında veriyi keşfetme süreci, veriyi tanıma süreci var demektir . Sonra veri işlenecek şekle göre dönüştürülmektedir ve sonra model yapılmaktadır. Bütün sistemin sonunda bir değerlendirme süreci vardır. Şayet bu değerlendirme süreci tatmin edici bir boyuttaysa o zaman artık bitti yani SEMMA başarıya ulaştı demektir. Eğer değerlendirme süreci başarılı değilse o zaman örnekleme geri dönülmelidir.



Şekil 3.1.2

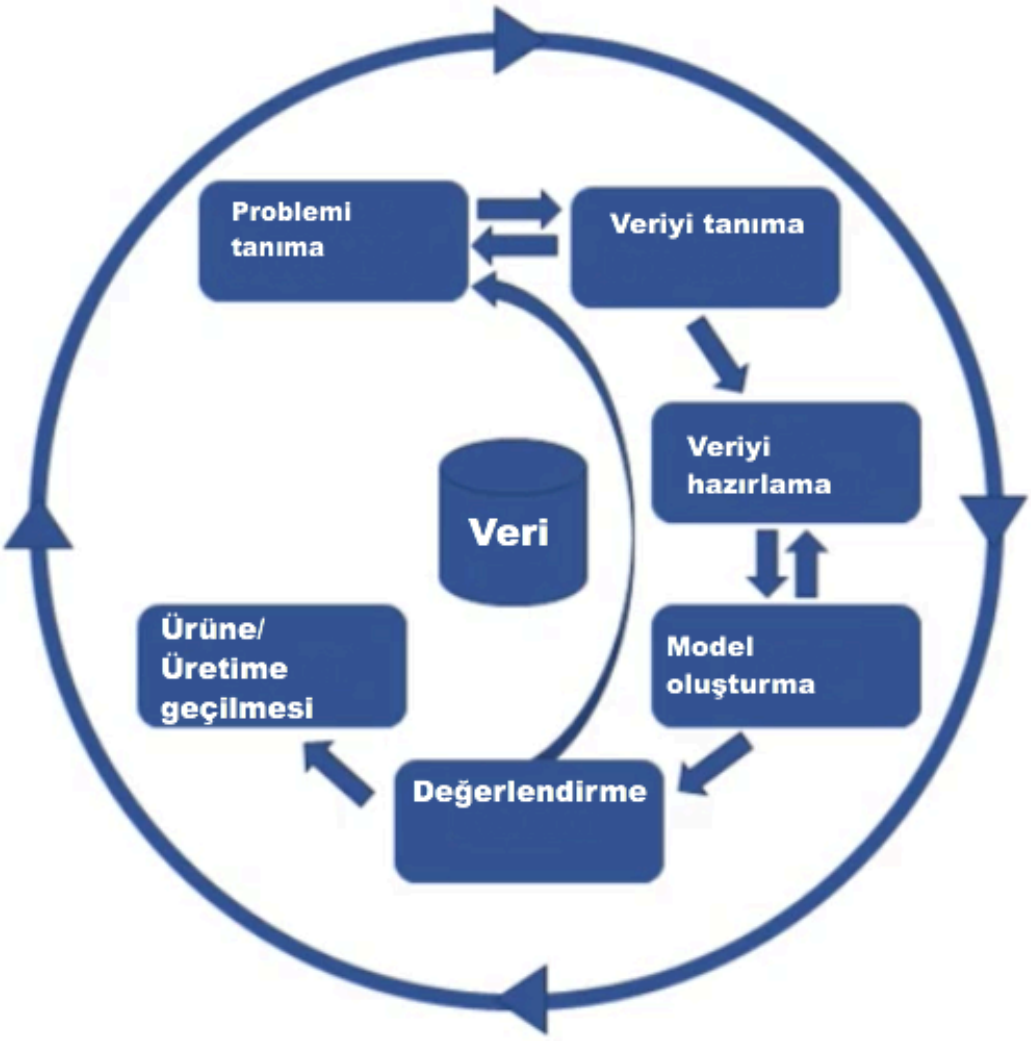
Özetle, öncelik verinin örneklenmesi(sample), verinin keşfedilmesi(explore) daha sonra veriyi uygun hale dönüştürmek (modify) sonra üzerinde bir model oluşturmaya çalışmak ve en sonunda da oluşturulan bu adımların tatmin edip etmediğine bakmak, problemi çözmek için işe yarıyor mu? Yaramıyorsa tekrar başa dönüp baştan başlamak gerekmektedir.

3.2 CRISP-DM

CRISP-DM kelime olarak Cross-industry Standard Process for Data Mining'in kısaltılmışıdır. Veri madenciliği (data mining) için bütün endüstrilerden bağımsız olarak çevrilebilen belli bir standart süreci anlatmaktadır. Veri bilimi yöntemlerinde çok kullanılan yöntemlerden birisidir. Basitçe bir projeye başlarken iki temel aşamadan bahsedilir; **Business Understanding** ve **Data Understanding**. Anlamı; öncelikle problemin tanınması ve problemi tanıdıktan sonra da verinin tanınmasıdır. Bir diğer tabirle de probleme uygun veriyi toplamak denilebilir. Örneğin, bir müşteri segmantasyonu yapılacak ya da kredi kartı sahtekarlığı yakalanacak bunun için öncelikle problem anlaşılmalıdır. Verilen örnek sadece problem başlıklarıdır. Bunların detaylandırılması gerekmektedir. Problem nedir? Nereden başlanıyor? Hangi noktada duruluyor? Gibi soruları sormak gerekir. Daha sonrasında da bu sorulara göre veri toplama aşaması ya da eldeki verinin anlaşılması aşaması başlar. Veri doğru şekilde tanınmıyorsa yapılan şeylerin anlamsız olma ihtimali var demektir. Bu süreçlerin bilinmesi bu anlamda çok önemlidir.

Şekil 3.2.1

Şekil 3.2.1'de görülen şemanın Türkçe çevirisi 3.2.2 de görülmektedir.

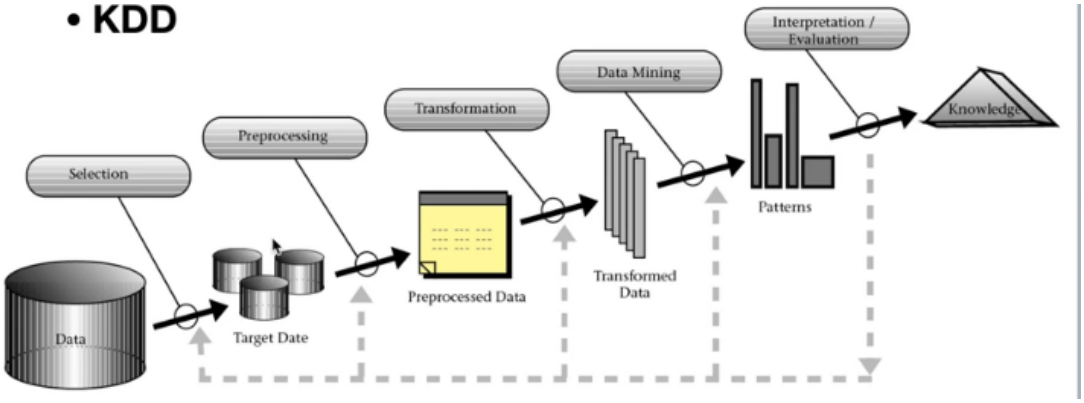


Şekil 3.2.2

Şekilde görülen bussiness understanding ve data understanding aşamaları bittikten sonra da verinin hazırlanması (data preparation) aşaması vardır. Buna verinin ön işlenmesi (preprocessing) , ETL'de dahil edilebilir. Örneğin müşterilerin verileri toplanmışsa ve bir müşterinin adres,yaş vb. bilgileri eksikse bunların bir şekilde giderilmesi ya da kirli verilerin bulunması gerekmektedir. Daha sonra bir modelin oluşturulması gerekmektedir. Kirli verilerin (yaş kısmına -500 yazmak) ya da gürültülü verilerin (yaş kısmına 500 yazmak) temizlenmesi gerekmektedir. Genellikle verinin üzerinden inşa edilen bu model makine öğrenmesi algoritmaları, istatistiksel bir model olabilir. Model oluşturulduktan sonra modelin değerlendirilmesi yani elde edilen sonuçların değerlendirilmesi gerekir. Şayet bu değerlendire tatmin edici bir boyuttaysa o zaman ürüne, üretime geçiliyor ve sistem çalışmaya başlamaktadır. Bu yeterli bir tatmin seviyesi sağlamıyorsa yapılmış olan testler çok büyük hatalar döndürüyorsa o zaman tekrar bütün süreci baştan yapmak gerekir. Modeling kısmı verinin üzerinde biraz model inşaasını kapsar (biraz tasarımı gibidir). Daha sonra da test edilerek üretime geçilmesi düşünülebilir. Bu bir döngüdür. Sürekli olarak tekrar eden bir yapıdır. Genelde endüstride yani gerçek hayat uygulamalarında, projelerinde çok sık tercih edilen bir yöntemdir.

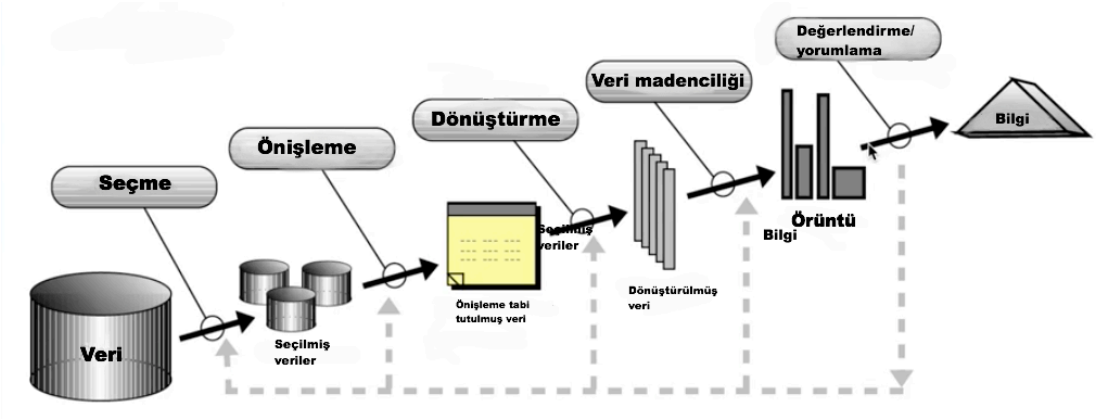
3.3 KDD

KDD, Knowledge and Data Discovery kelimelerinin kısaltılmış halidir. Adından da anlaşılacağı üzere bilgi ve veriyi keşfetme süreci olarak da düşünülebilir. Basitçe 6 adımdan yani 5 işlemden oluşmaktadır. Veriyle başlanır, veri içinden bir seçme işlemi yapılır yani problemde özel olarak hangi verilerin kullanılacağı seçilir. Sonra bu veri üzerinden bir ön işleme yapılır ve veri işleyebilecek hala getirilir (örneğin eksik veriler temizlenir, kirli veriler temizlenir, boyut düşürme yapılır vb.). Ve sonra veri dönüştürülür (örneğin metinden sayısal değerlere dönüştürme, ortalama alınır, özetleme yapılır vb.). Veri madenciliği bu aşamada devreye girmektedir. Model denilen yapı patterns / örüntüler ortaya çıkar. Değerlendirme sürecinin sonunda da bir bilgiye ulaşılmaktadır.



Şekil 3.3.1

Daha önceki bölümlerde açıklanan SEMMA ve CRISP-DM bir döngü içerisindeyken KDD doğrusal bir yapıdadır. Burada da şekil 3.3.1'de de görüldüğü gibi bir aşamadan sonra oklarla belirtilen şekilde daha önceki bir aşamaya dönülüp tekrardan o aşamada çalışılabilir.



Şekil 3.3.2

Şekil 3.3.2, yukarıdaki şekil 3.3.1'in Türkçe'ye çevirilmiş halidir.

KDD, veri bilimi için oldukça önemli bir metottur. Genelde araştırma amaçlı yapılan işler için kullanılmaktadır. Bir şey ilk kez yapılıyorsa yani daha önce hiç karşılaşılmamış bir problemle ilgili ilk kez bir keşif yapılacaksa şayet genellikle KDD kullanılmaktadır. Ürüne yönelik, endüstriye yönelik, piyasaya ve uygulamaya yönelik işler için CRISP-DM daha çok tercih edilmektedir. Aslında genel olarak SEMMA, CRISP-DM ve KDD'de aynı şeyler yapılır. Veriden ya da problemden başlanır, daha sonra veri hazırlanır sonrasında model oluşturulur ve değerlendirilir. Değerlendirme şayet tatmin ediyorsa uygulanır eğer etmiyorsa önceki aşamalara dönülerek tekrardan düzenlemeler yapılır. KDD veri ön işleme ve dönüştürme konusunda biraz daha detaylandırılmıştır. Bu bölümden sonra veriyi hazırlama, model oluşturma gibi adımların hepsi detaylı açıklanacak ve örneklerle çözülecektir.

3.4 Kavramlara Giriş, Veri Bilimi, Veri Madenciliği, Makine Öğrenmesi, Büyük Veri

Bu bölümde amaç veri bilimi ile ilgili kavramların tanımlanmasıdır ve aşağıdaki sorular gibi sorulara cevap vermektir.

- Veri bilimi nedir?
- Veri madenciliği nedir?
- Makine öğrenmesi nedir?
- Bu kavramların istatistik ile ilişkisi nedir?
- Ne işe yararlar? Nasıl para kazanılır?
- Nereden çıktılar?
- İş analisti kimdir?

Şekil 3.4.1

Şekil 4.3.1'de de görüldüğü gibi süreçle/iş dünyasıyla ilgili 4 katmandan oluşmaktadır. Bu kitap iş analitiği açısından oldukça kullanışlı ve yardımcı olabilecek bir kitaptır. No code yani kod yazmadan da veri bilimi yapılabilecek bir tool olan Knime üzerinden açıklanmaktadır. Kendi verisini kullanacak ya da iş dünyasındaki görevinde verilerini kullanacak toplumdaki herkesin kullanabileceği bir araçtır.

İş analitiği (business analytics) en üst katmandadır. Sahanın doğrudan aksiyonun/süreçlerin içinde yaşayan insanların hitap eden, analitik olarak bu süreçleri

ele alabilen ve bu süreçleri veri üzerinden faydalı işleyebilen kişilere verilen unvana da iş analisti denmektedir.

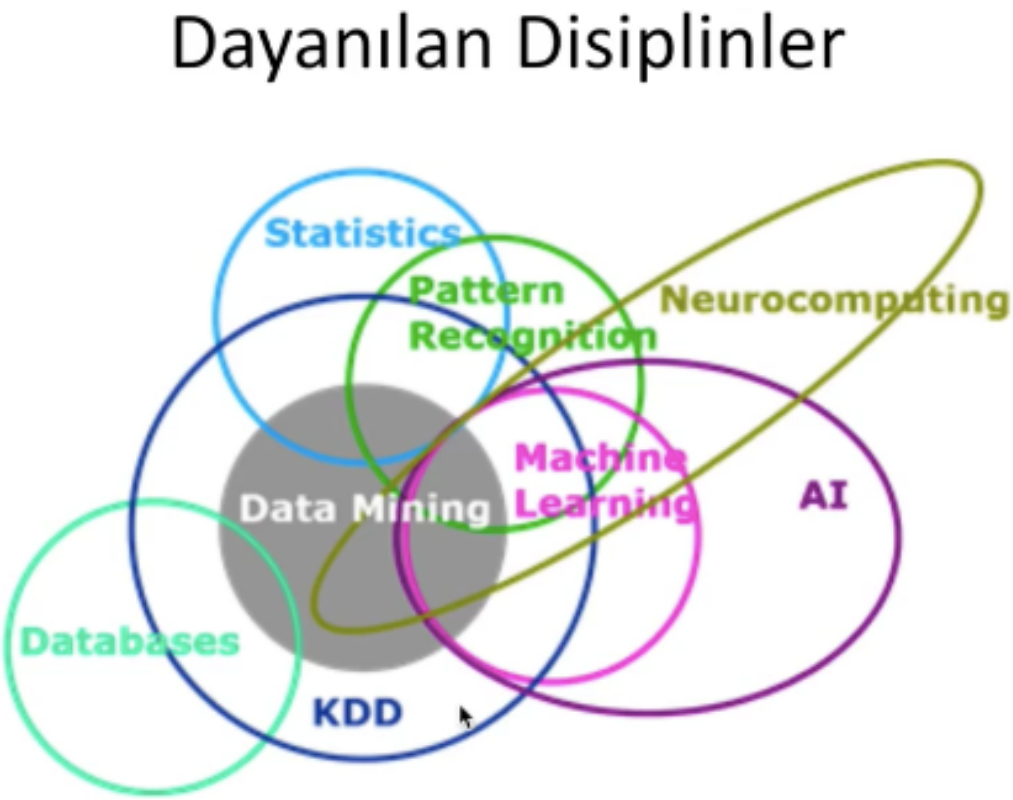
Veri bilimi (data science), veri üzerinden bilim üretilen katmandır. Yukarıdaki katmanda bilim insanı olmayan/biliminden anlamayan , süreçlere yakın kişilerden bahsedilirken burada bilimden bahsedilmektedir. Burada yapılan işin nasıl daha verimli yapılabilir? Yeni algoritmalar neler kullanılabilir veya bu algoritmalar nasıl iyileştirilebilir? gibi soruları cevaplar. Bu kitapta bir çok örnek/ problem gösterilecektir. Bu problemleri çözmek iş analitiğinin derdidir. Ayrıca hangi algoritmanın kullanılacağı da iş analistinin dertlerinden biridir. Veri biliminde hangi algoritmanın kullanılacağından çok nasıl uygulanacağı önemli noktalardan biridir.

Veri madenciliği, istatistiğin ve makine öğrenmesinin çalıştığı alanlardan birisidir. Alt katmanlara indikçe problem çözümü için yeni bir algoritma arayışı olur veya sadece yeni bir algoritma/ model çıkarmış olmak için bile model çıkartılır. İş analitiğindeki katmanda çalışan kişilerin yeni bir algoritma çıkarma çabası yoktur. Veri bilimi katmanında olan kişiler hem sahadaki problemleri bilir hem de bunların çözümü için algoritmaları bilir. Bu kitapta problemin doğru anlaşılması (iş analitiği katmanı) ve bu probleme uygun algoritma kullanımı (veri madenciliği katmanı) gösterilecektir.

Veri bilimcisinin (data scientist) 3 temel kavrama hakimdir. Bunlar;

- Saha bilgisi (domain knowledge). Örneğin bankacılıkla ilgili çalışılıyorsa bankacılık ile ilgili kavramlara hakim olmak
- İstatistik
- Bilgisayar bilimleri (computer science) kodlama biliyor olmak

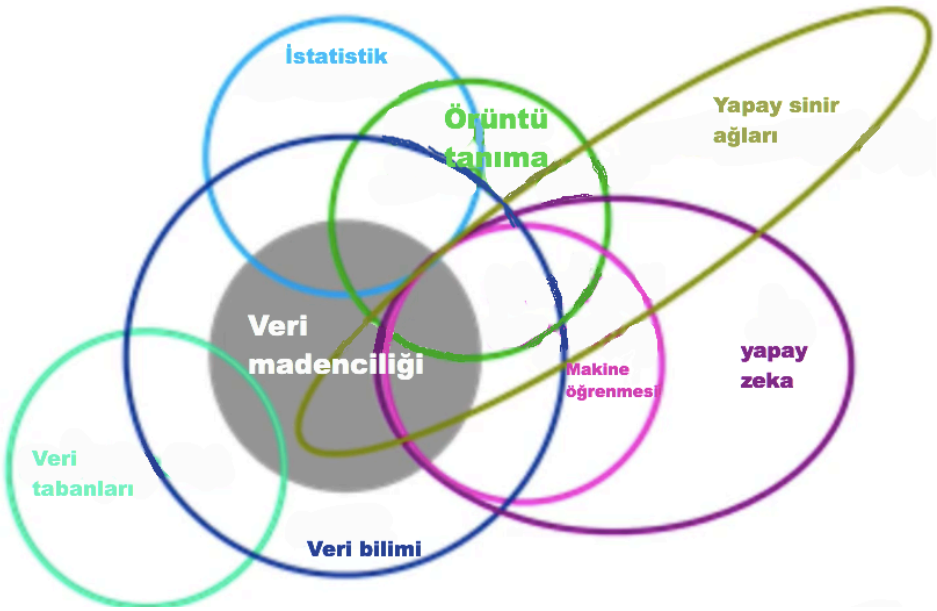
Her veri bilimcisinde bu 3 olası zorunlu olan özellik olamayabiliyor.



3.4.2

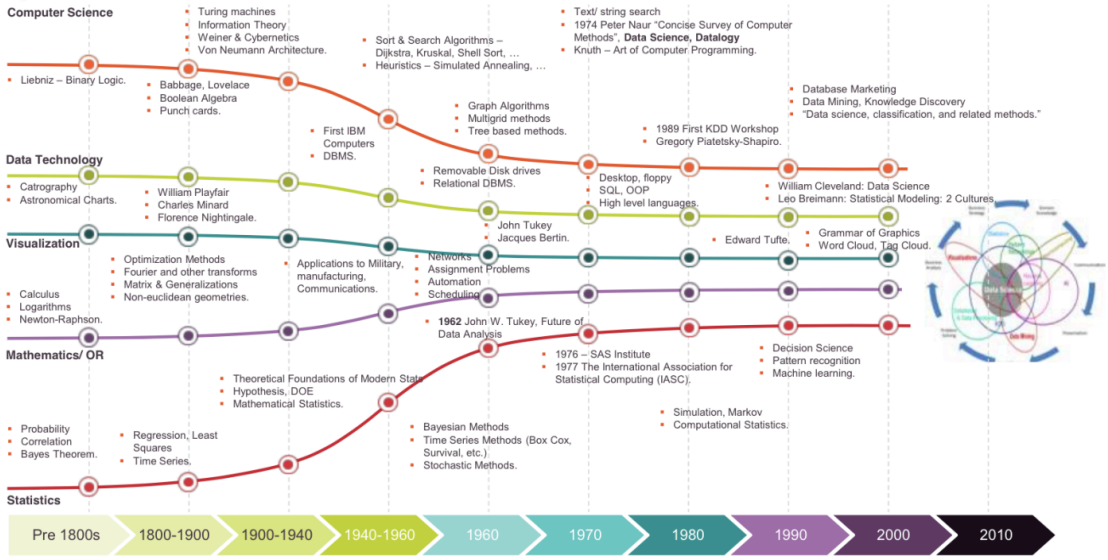
Kavramların grafiği şekil 3.4.2'de daha detaylı görülebilmektedir.

Şekil 3.4.3, bir üstteki şeklin Türkçe'ye çevirilmiş halini göstermektedir.



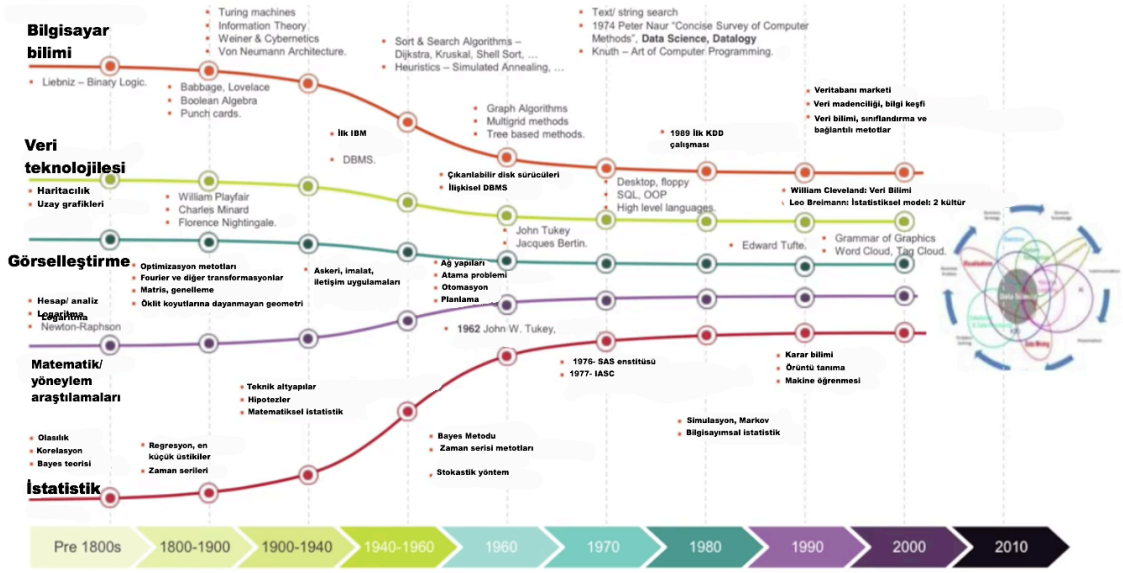
Şekil 3.4.3

Örneğin KDD çemberi data science (veri bilimi) çemberi denebilir. Data mining bu işin kalbinde / çekirdeğinde yer almaktadır. Data mining'in tüm özellikleri data science de (KDD) kullanılır. Machine learning (makine öğrenmesi) yapay zekanın (AI) bir alt dalı olarak görülebilir. Makine öğrenmesi (machine learning) ve yapay zeka (AI) veri madenciliğini (data mining) ve veri bilimini (KDD) besler. Benzer şekilde statistics (istatistik) veri bilimini ve veri madenciliğini besler. Burada dikkat edilmesi gereken şey makine öğrenmesinin istatistik ile kesişimlerinin olmadığıdır. Bazı problemlerde istatistik bazılarında ise makine öğrenmesi kullanılır. Bazen de aynı problemin bir problemin ön işleme kısmında istatistik kullanılırken model kısmında makine öğrenmesi kullanılabilir fakat ikisi aynı anda kullanılmaz. Yapay sinir ağlarıyla popüler hale gelmiş olan neurocomputing veya databases (veri tabanları) data mining ve KDD ile beslenebilir hatta oldukça big data'ya kaymıştır. Bu farklı disiplinler sayesinde big data'da beslenmektedir.



Şekil 3.4.4

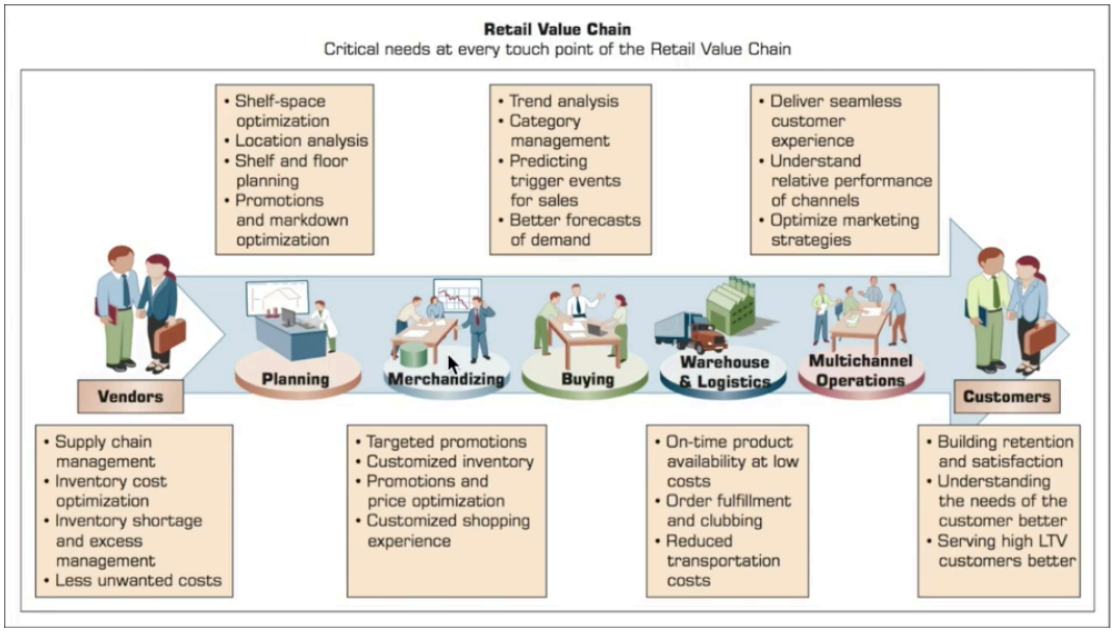
Şekil 3.4.4'de de görüldüğü gibi, 2010 yılı bölümünde bir önceki şekilde açıklanan kavramların grafiği çizilmiştir.



Şekil 3.4.5

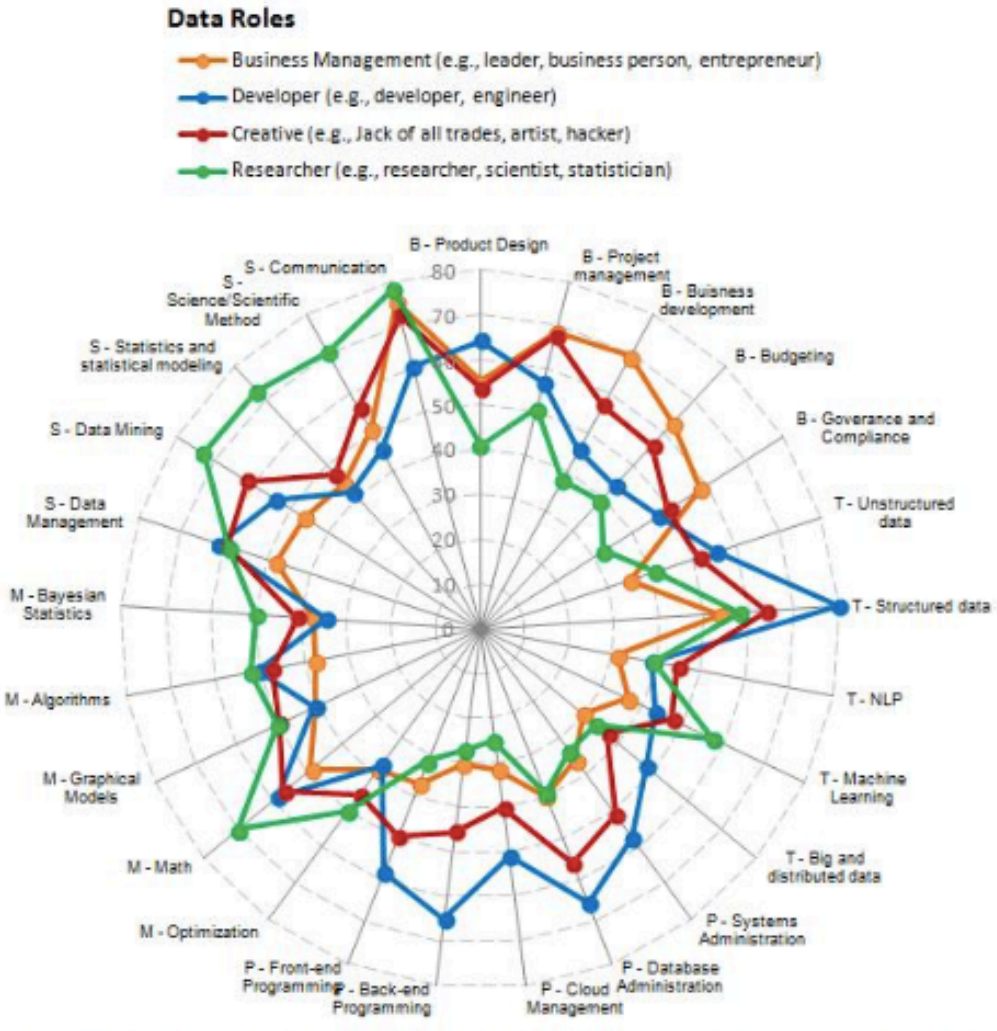
Şekil 3.4.5, bir üstteki şekilin Türkçe'ye çevirilmiş şeklidir. Bu resme giden 5 ana kol bulunmaktadır. Bilgisayar bilimleri (data science), Liebniz 2lik tabandan başlar ve bugüne kadar devam eder. Data technology, verinin nasıl saklanacağı, işleneceğinden bahseder. Visulization, verinin görselleştirilmesidir. Knime veri görselleştirme aracı değil bir data science suittir. Bir sonuca ulaştıktan sonra nasıl görselleştirileceği önemli bir sorundur. Mathematics /OR, yöneylemden gelen bir çalışma vardır. Özellikle son zamanlarda prescriptive analysis (buyrukçu analitik) kavramı girdikten sonra optimizasyon teknikleri önem kazanmaya başlamıştır. Türkiye'de endüstri mühendisliğinde okutulan yön eylem çalışaları bulunmaktadır. Statistics (istatistik), önemli dallardan biridir. Olasılık teoreminin kurulmasından, olaylar arasında ilişki kurulmasına ve karar destek ilişkilerine kadar giden bir süreci vardır.

Data science (veri bilimi) aslında veriyi paraya dönüştürme yöntemidir / sanatıdır. Bir şirketin departmanındaki veriler örneğin gelen çağrılar, şikayetler, satışlar, maaşlar vb. Veriler genellikle depolanıyordu hatta bir çok şirkette depolanmıyordu bile. Fakat artık bu veriler değere dönüştürülüyor. Bir kişi siteye bağlandığında browser'ından, ekran çözünürlüğünden o kişinin gelir düzeyine, demografik yapısına kadar bağlantı kurulabilmektedir. Nereden bağlanıldığına göre mağaza şubesi açılırken bu bilgi kullanılabilmektedir. Bu değer, bazen para, toplum kuruluşu için daha kaliteli hizmet olabilir, bir kamu kuruluşu için işlerin daha işleme süreci olabilir vb. Süreçler olabilir ama önemli olan artık bunların kullanılarak değere dönüştürülmesidir.



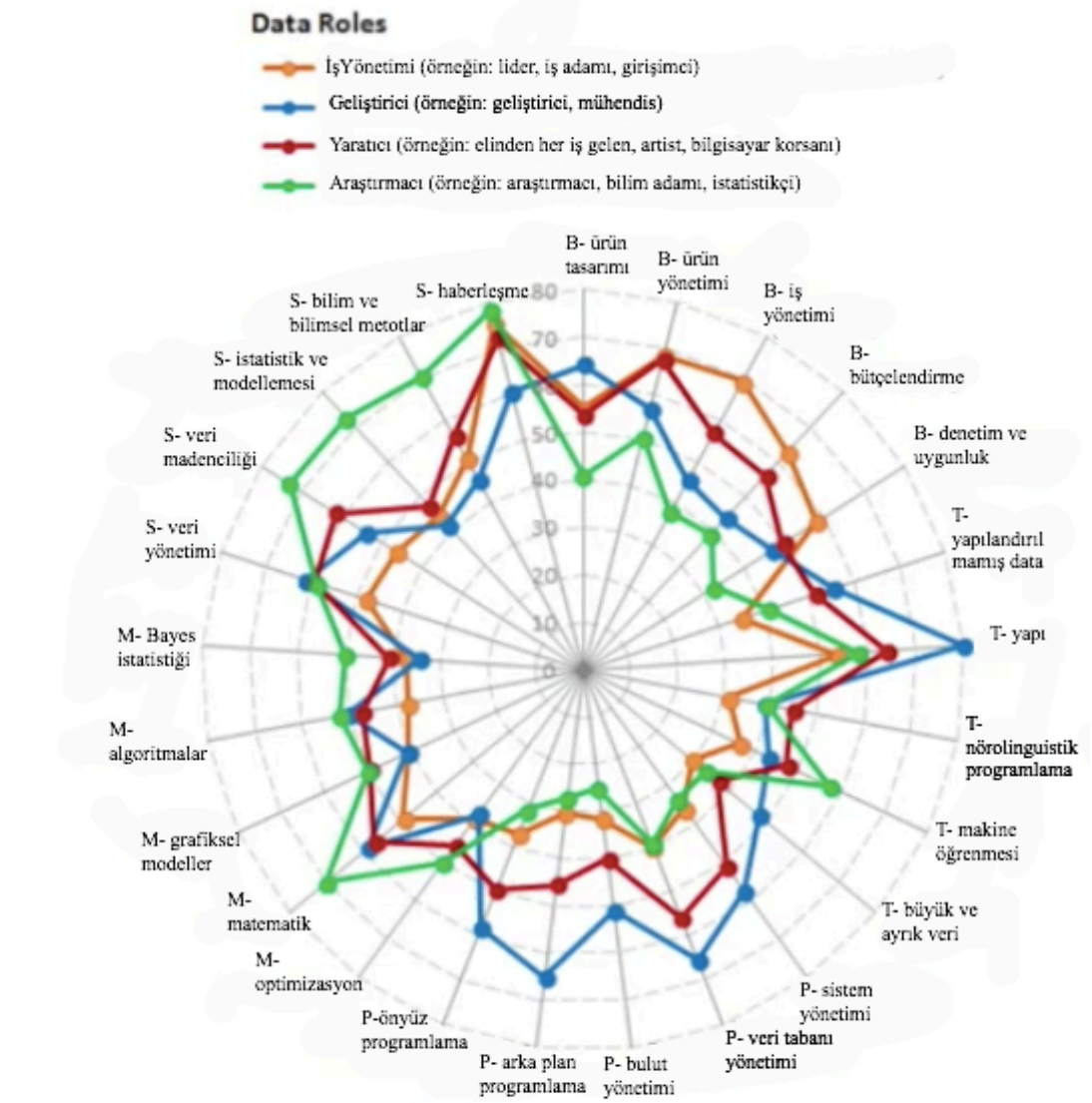
Şekil 3.4.5

Şekil 3.4.5, perakende zincirinin süreçlerini gösteren bir örnektir. Öncelikle tedarikçilerle başlar, bir planlama yapılır (planning) daha sonra da merchandizing (satın alma süreçleri) belirlenir. Satın alma işlemi gerçekleştikten sonra da warehouse (depolaması) ve logistics (tedarik) süreci başlar. Son olarak da multichannel operations (çok kanallı operasyon) süreci olur. Burada satın alma örneği üzerinden açıklandı fakat herhangi başka bir örnek de düşünülebilir.



Şekil 3.4.6

Şekil 3.4.7, şekil 3.4.6'nın Türkçe'ye çevirilmiş halidir.



Şekil 3.4.7

Şekil 3.4.7 'de 4 farklı disiplin gösterilmektedir. Bunlar: iş yönetimi süreci, geliştirme, yaratma süreci, araştıma süreci. Geliştirilen tüm modeller bir ürüne dönüştürülmüştür. Arkasında bir yazılım süreci olan bu ürünler, büyük veri üzerinde bir analitik olabilir, Python'da /Java'da bir kod yazmak olabilir veya hazır bir platformda bu kodun çalıştırılası olabilir. Creative işlere, müşterilerle iletişimin nasıl kurulacağı, hacker'ın çalışması, kampanya üretimi, markanın piyasaya çıkarılması vb. örnekleri gösterilebilir. Researcher (bilim insanları) istatistik, matematik, endüstri mühendisliği kökenli kişilerin daha avantajının olduğu resimde görülmektedir. İş yönetimi, bütçe çıkarımı gibi işlerin iş yönetimi disipliniyle daha çok ilgili olduğu da görülmektedir. Veri bilimi şekilde görülen tüm disiplinleri kullanmaktadır. Örneğin balıkçılık, turizm veya herhangi bir konuda çalışılacaksa o kökenden geliniyor ya da o konudaki terimleri biliyor olmak gerekmektedir. Örneğin hangi ayda müşteri sayısının artacağı, sonraki sene için

beklenen turist sayısı vb bir sürü veriye dayalı sorunun sorulup cevap almasını sağlayan bilim veri bilimidir.

Bu bölümde kısaca;

Veri bilimi; veriyi paraya / değere dönüştürme yöntemi,

Veri madenciliği, veri biliminin özünü oluşturduğu,

Makine öğrenmesi, 150 üzerindeki modülün olduğu, algoritma üretilmesi gibi sorular için bakılacak alan,

İstatistiğin veri bilimi için önemli olduğu, makine öğrenmesi ile birbirine alternatif olduğu açıklanmıştır.

4.BÖLÜM: PROBLEMİ TANIMAK

4.1 Descriptive, Predictive ve Prescriptive Analitik Farkları

Bu bölümde amaç analitik seviyelerini tanıtmaktır. 3 temel analitik seviye vardır. Bunlar tarihsel açıdan ve uygulama seviyesi açısından da sırası tanımlayıcı (descriptive), tahminci (predictive) ve buyrukçu (prescriptive) şeklindedir yani bunlar ardışıktır.

Tanımlayıcı analitik (descriptive), veriyi tanıma sürecidir. Veri üzerinden geçmişe yönelik bilgiler araştırılır. Örneğin, geçmişte bir sahtekarlık yaşandı mı? Geçmişte satışlar nasıldı gibi soruların cevabı burada aranır. Müşteri segmentasyonu, en çok hangi alış veriş yaptı? Gibi sorular da geçmişe yönelik hali hazırda toplanmış verinin tanınması olarak yorumlanır. Ne oldu sorusundan sonra bazı kaynaklarda neden oldu sorusu da eklenebilmektedir.

Tahminci analitik (predictive), geleceğe ne olacağının incelendiği süreçtir. Aslında predictive Türkçe'ye bilinmeye ulaşma şeklinde çevirmek daha doğrudur. Forecasting, tahmin etmek şeklinde çevrilebilir. Örneğin veri setinde müşteri yaşı girilmemişse onun yaşının tahmini de bir prediction (tahmindir). Bu geçmişe yönelik bir tahmindir. Gelecek için yapılan ve öngörü denebilecek olan ise forecasting'tir. Burada predictive ile amaç bilinmeyen / eksik veri için tahminlerin yapılmasıdır. Örneğin, gelecek sene mağazada ne kadar siyah tişört satılır? Sahtekarlık yapma potansiyeli en yüksek müşteri hangisidir? Şirketten bağlılığını koparacak /bırakacak potansiyeli en yüksek müşteri hangisidir? Gibi soruların cevabı burada aranabilir. Tahminci analitiğin çalışabilmesi için verinin tanınması gerekmektedir. Yani öncelikle descriptive (tanımlayıcı) analitik çalışılarak veri tanınır daha sonrasında predictive (tahminci) analitik çalışılır.

Buyrukçu analitik (prescriptive), descriptive ve predictive aşamaları geçildikten sonra alınan aksiyonu gösterir. Örneğin müşteri web sitesine girince o müşteriye en uygun ürünün tavsiye edilmesi, uygun müşteriye uygun ürün mesajının atılması vb. örnekler karar verici adına aksiyon içeren buyrukçu analitiğe giren örneklerdir.

Şekil 4.1.1

Şekil 4.1.1'de görülen şemada en sola insan en sağa makine konulacak şeklinde düşünülürse, veri analitiğinin / veri biliminin evrimi de görülebilir / düşünülebilir. İlk kutucukta insana bilgi veren veriyi tanıtan süreç sonralar geldikçe veri üzerinden değer kazandıran ve aksiyonu paraya çeviren insana dönüşür. Tahminci de ise biraz ileri seviyede artık eksikleri giderecek tahminlerin yapıldığı ve bu yüzden insanın devreden çıkmaya başladığı süreçtir. Buyrukçuda artık neredeyse insanın tamamen süreçten çıktığı ve tamamen makinenin devrede olduğu süreçtir.

Prescriptive'in en çok kullanıldığı alanlardan biri , Türkiye'de de , marketing 'tir (pazarlamadır). Örneğin Amazon gelirinin 35% gelirini bir recommender algoritmasından, Netfilix gelirinin 75% bir recommender algoritması ile sağlanmaktadır. Bu tarz büyük sistemler artık insan gücüyle çalışabilecek sistemler değildir. Örneğin kendi kendine gidecek araç veri toplayarak kendi kendine karar vererek yolda devam edecek bir araçtır.

Genel olarak tanımlayıcı analitiğin çıktıları, raporlar, dashboardlar, skarkartlardır. Tahminci analitiğinde, veri madenciliği, metin madenciliği, web madenciliğinin yoğun kullanıldığı ve örneğin bir sonraki ayda doların ne kadar olacağı gibi tahminlerin yapıldığı analitik seviyesidir. Buyrukçu analitikte, iyileştirme, simülasyon, tavsiye algoritmaları, uzman sistemler yapılır. Hesaplamalar yaparak oyun kuralları belirlenir yani geçmiş veriler yeterince anlaşıldığı için gelecekle ilgili tavsiye algoritmaları yapılabilir ve bu senaryolar iyileştirilir. Problemler iyi anlaşılmalı ve veri iyi tanınmalıdır. Bunun üzerine analitik seviyesi bilinmeli ve ona göre davranılmalıdır.

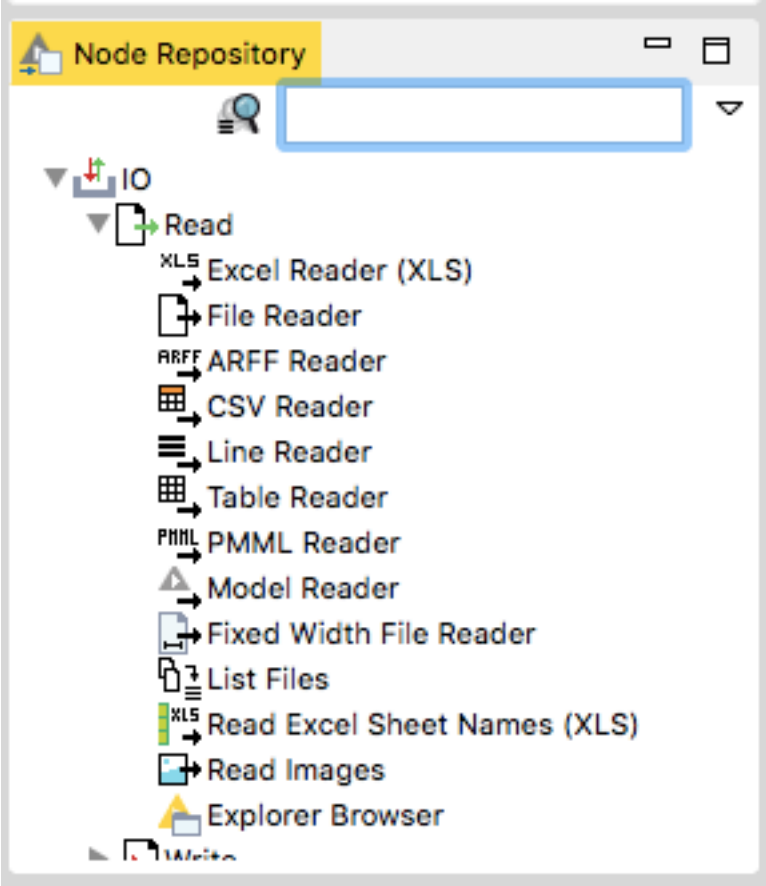
CRISP-DM'de anlatıldığı gibi ilk olarak yapılacak olan iş ve veri anlaşılmalı sonrasında veri hazırlanmalı ve model oluşturulmalıdır.

Bu bölümde amaç problem çeşitlerinin anlaşılabilmesiydi. Bundan sonraki bölümde bilinen problemlerin dört gruptan birine koymak için tanımlamalar ve açıklamalar yapılacaktır.

5.BÖLÜM: VERİYİ TANIMAK

5.1 Dosya Dönüşümleri (Weka, ARFF; CSV; Excel Tip Dönüşümleri)

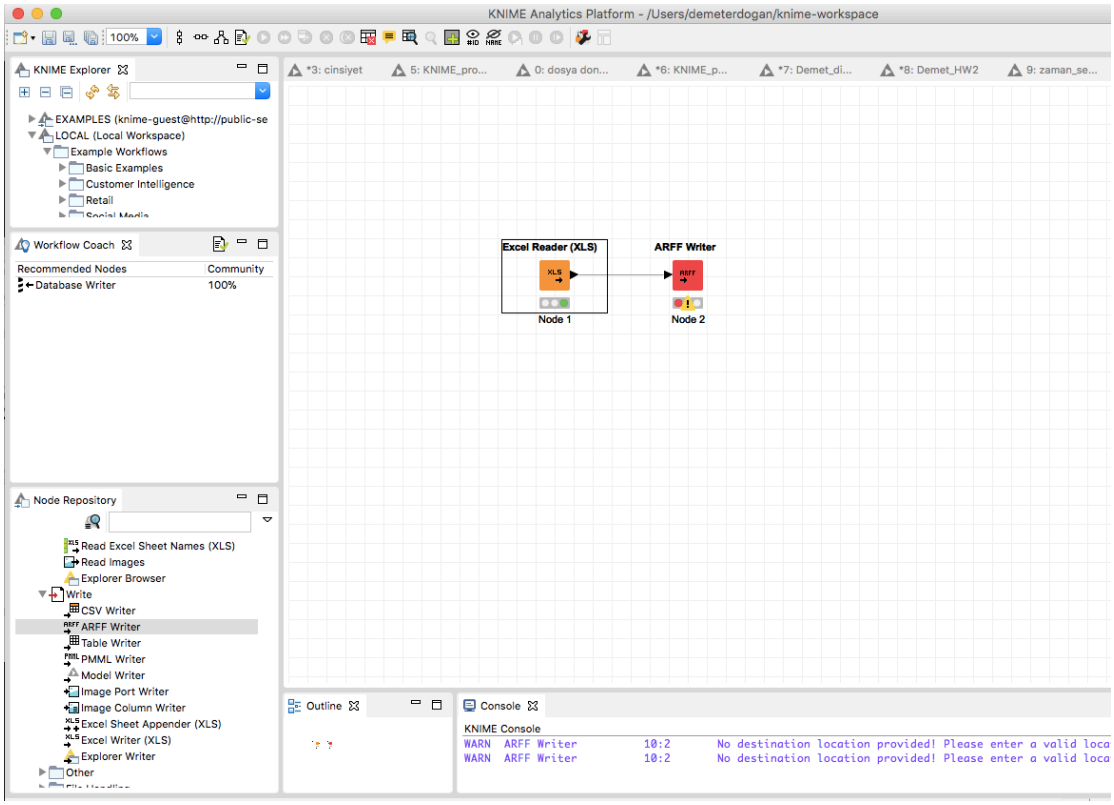
Bu bölümde amaç, Knime üzerinden veri dönüşümleri (dosya tiplerinin) açıklamak ve örnek göstermektir. Node repository altında olan IO klasörü altında olan file reader, ARFF reader, CSV reader, table reader vb. Operatörler dosyaları Knime'a aktarmak için kullanılır.



Şekil 5.1.1

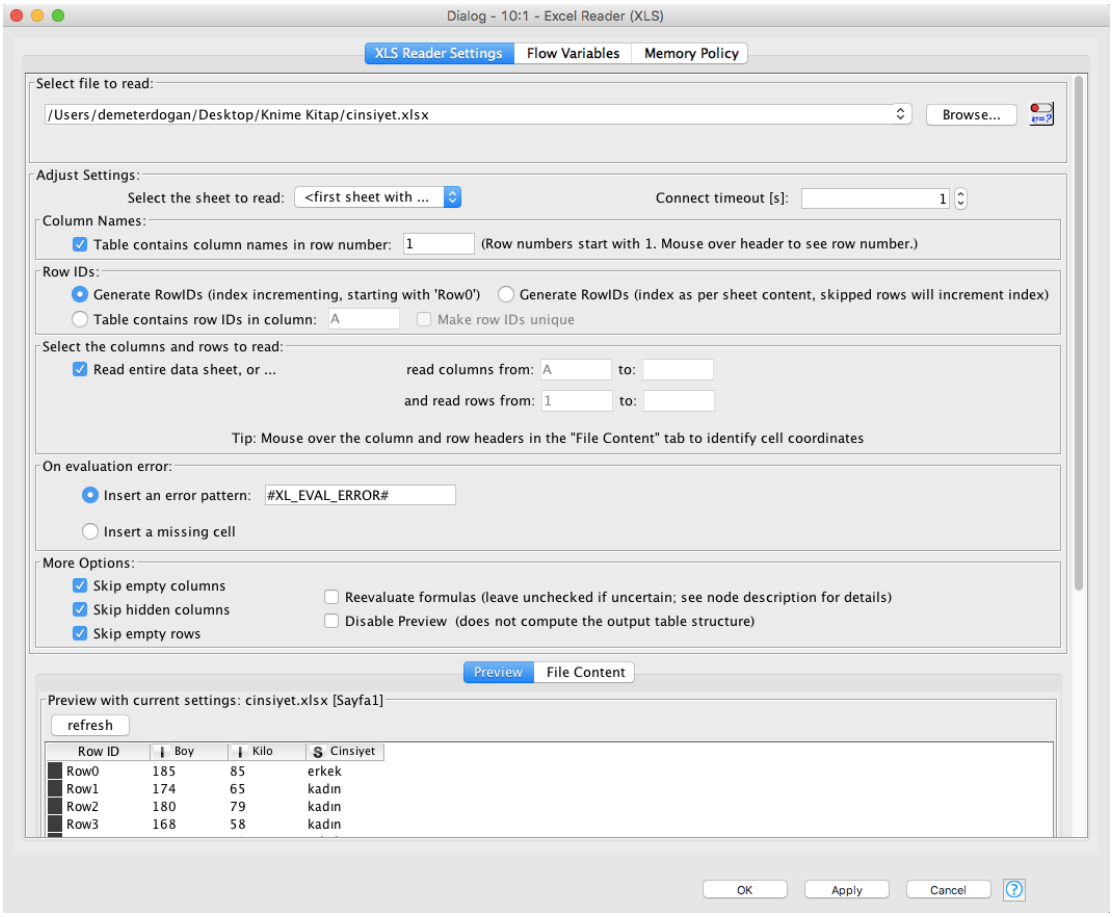
Şekil 5.1.1, IO klasörü altında dosyaların Knime'a aktarılmasını sağlayacak read operatörlerinin listesini göstermektedir.

Örnek olarak excel dosyasını ARFF formatına dönüşümü aşağıdaki şekilde yapılmaktadır.



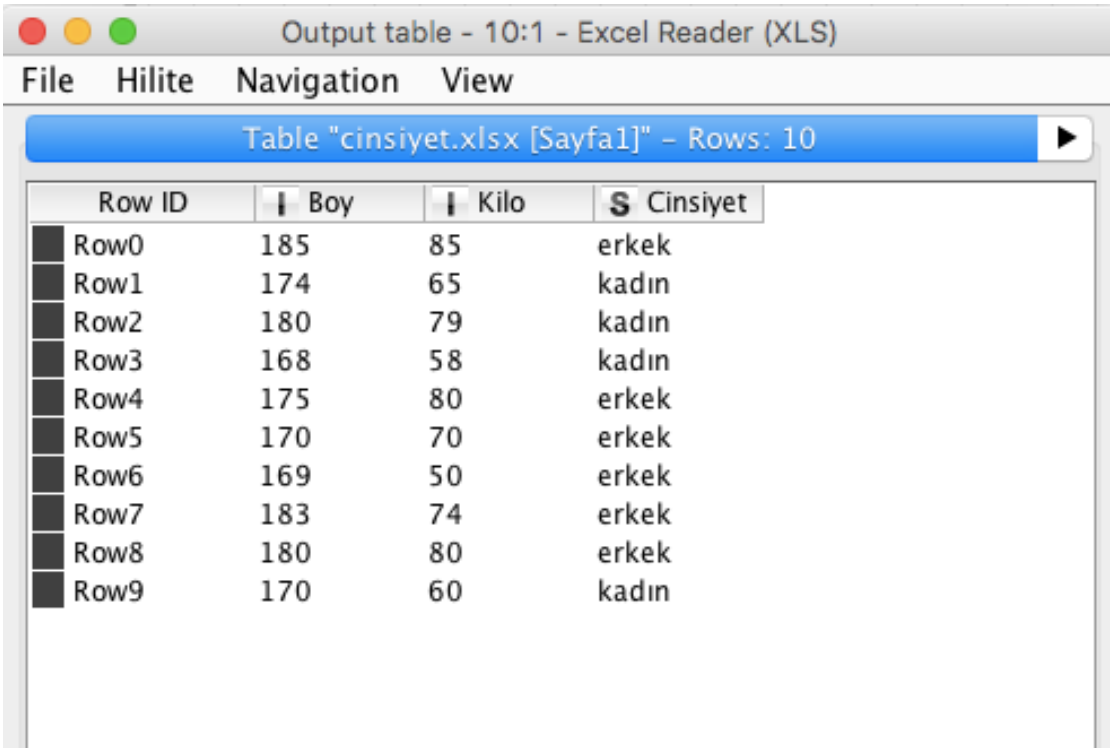
Şekil 5.1.2

Şekil 5.1.2, excel reader ve ARFF writer'ın sisteme eklenmesini ve bağlantısını göstermektedir.



Şekil 5.1.3

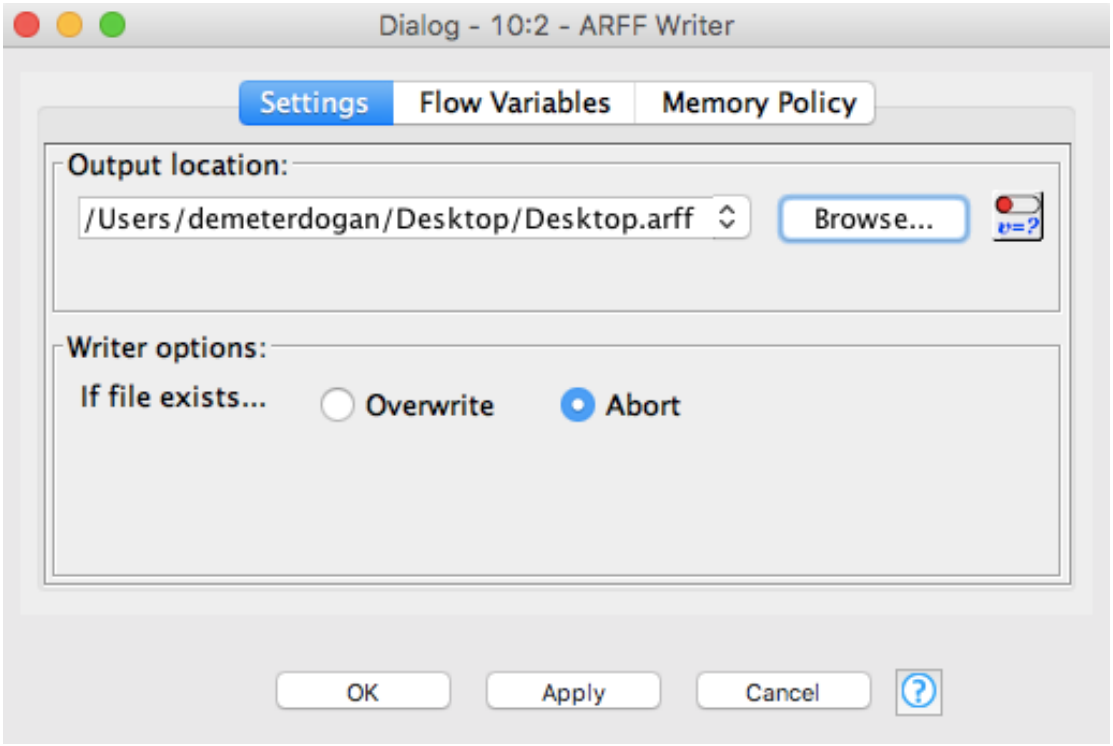
Şekil 5.1.3'de de görüldüğü gibi ilk satırların kolon başlığı olması için table contains column names in row number 1 seçeneği seçilmelidir ve daha sonra refresh tuşuna basılmalıdır. Bu örnekte diğer ayarları ile oynanmamıştır fakat seçenekleri tanıtmak gerekirse; Read entire data sheet, or seçeneği seçilirse read columns from ..... to .... seçenekleri hangi kolonlar arasında seçim yapılacağı ve and read rows from .... to .... ise hani satırların seçileceğinin girildiği alanlardır. Bu alanlar doldurulursa veri setinin tümü alınmayıp sadece bu boşluklardaki alanlara yazılan bölgeler alınır. Error pattern ve insert a missing cell bölümü herhangi bir hata durumunda yapılması istenilenin belirtileceği alandır. Skip empty columns bölümü boş olan kolonların geçilmesi, skip hidden columns gizli kolonların atlanması için seçilen seçeneği ve skip empty rows ise boş satırların atlanması için seçilen seçeneklerdir.



Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	79	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek
Row9	170	60	kadın

Şekil 5.1.4

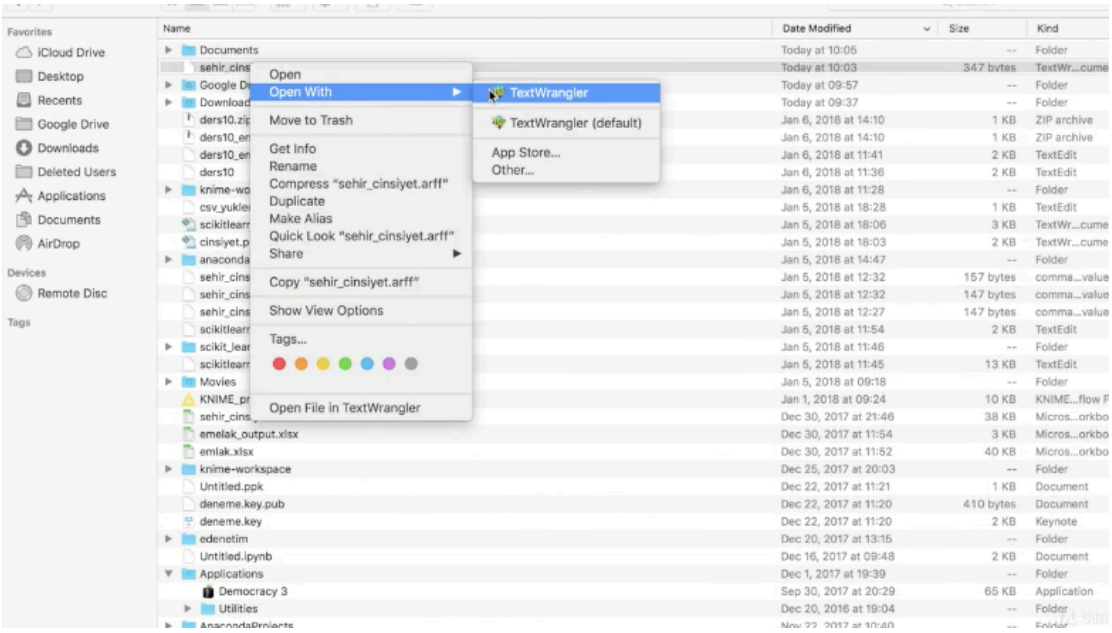
Şekil 5.1.4, kullanılan örnek excel dosyasını göstermektedir. Excel reader operatörüne sağ tuşla tıklanarak output table seçeneği seçilerek ulaşılabilir.



Şekil 5.1.5

Şekil 5.1.5, ARFF writer'ın configure penceresini göstermektedir. Burada amaç excel formatıyla Knime'a aktarılmış olan dosyanın Weka formatına dönüştürülmesidir. Weka

formatı için ARFF writer operatörü kullanılmıştır. Weka, makine öğrenmesi için yaygın kullanılan programlardan biridir. Rapid miner, Knime, Weka birbirne yakın yani çok fark bulunmayan tool'lardır. Browse seçeneğinden dosyanın kaydedilmek istenilen bölgesi seçilir. If file exists.. bölümünde ise bu dosya daha önce kayıtlı ise overwrite seçeneği olan dosyanın üzerine yeniden yeni olanın yazılması, abort ise iptal etmek anlamına gelmektedir.



Şekil 5.1.6

Şekil 5.1.6, kaydedilen dosyanın, kaydedildiği yerden açılmasını göstermektedir. Bu örnekte TextWrangler kullanıldığı için bu şekilde açılması gösterilmektedir.

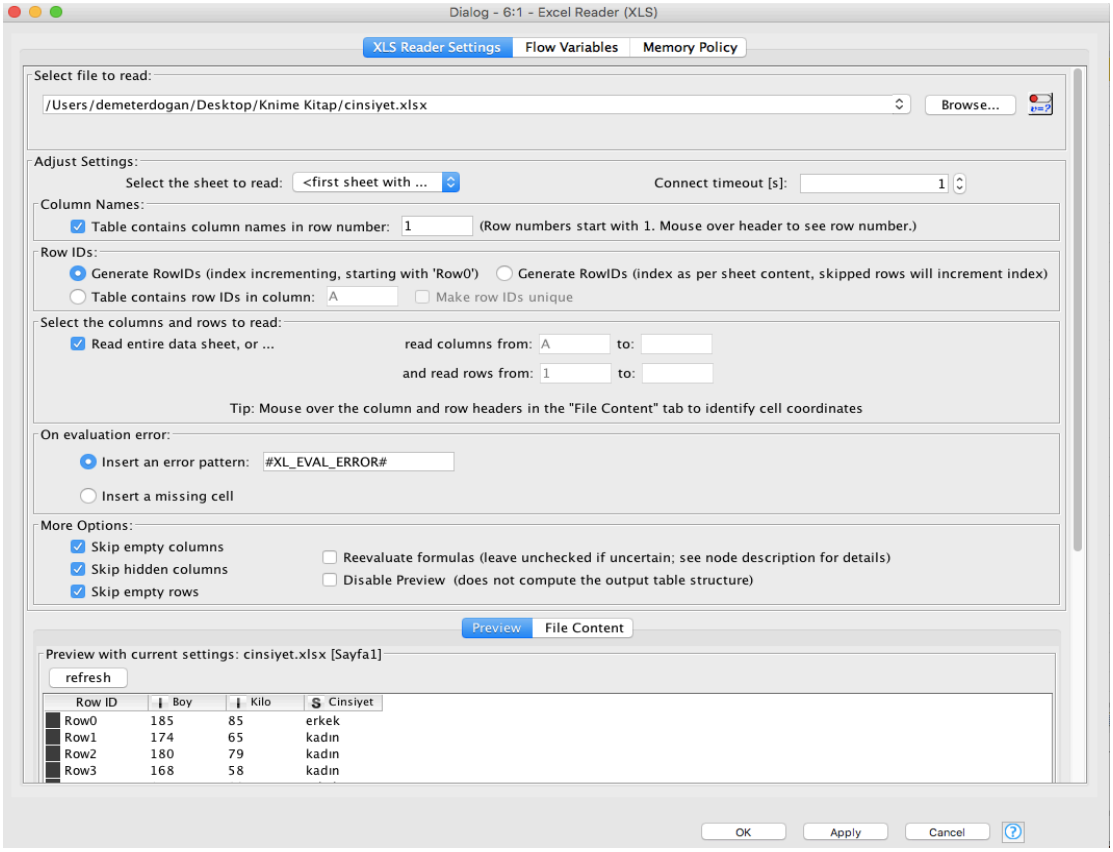
Şekil 5.1.7

Şekil 5.1.7, ARFF olarak kaydedilen dosyanın açılmış halini göstermektedir. Nodepad, windos, lenux kullanılıyorsa farklı yerlerden de açılabilmektedir. Şekilde de görüldüğü gibi, boy, kilo, cinsiyet kolonları belirtildikten sonra veri içerikleri @data yazısından sonra aşağısında verilmektedir. Türkçe karakter sorun oluşturabilir.

Bu bölümde örnek olarak alınan bir dosyanın farklı türe çevrilmesi gösterilmiştir. Çalışılan herhangi bir dosyanın daha sonrasında kullanılabilmesi için direk olarak farklı bir dosya tipi olarak da kaydedilmesi mümkündür.

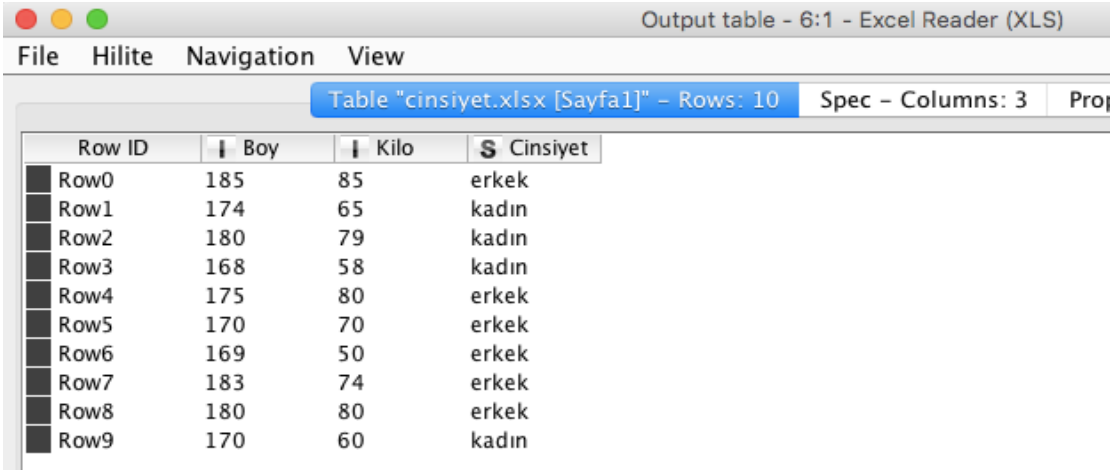
5.2 Veri Tipleri ve Veri Renklendirme

Bu bölümde amaç, veri renklendirmedir. Daha önceki bölümlerde de gösterildiği gibi excel reader operatörü sisteme eklenir ve configure bölümünden cinsiyet veri seti sisteme aktarılır.



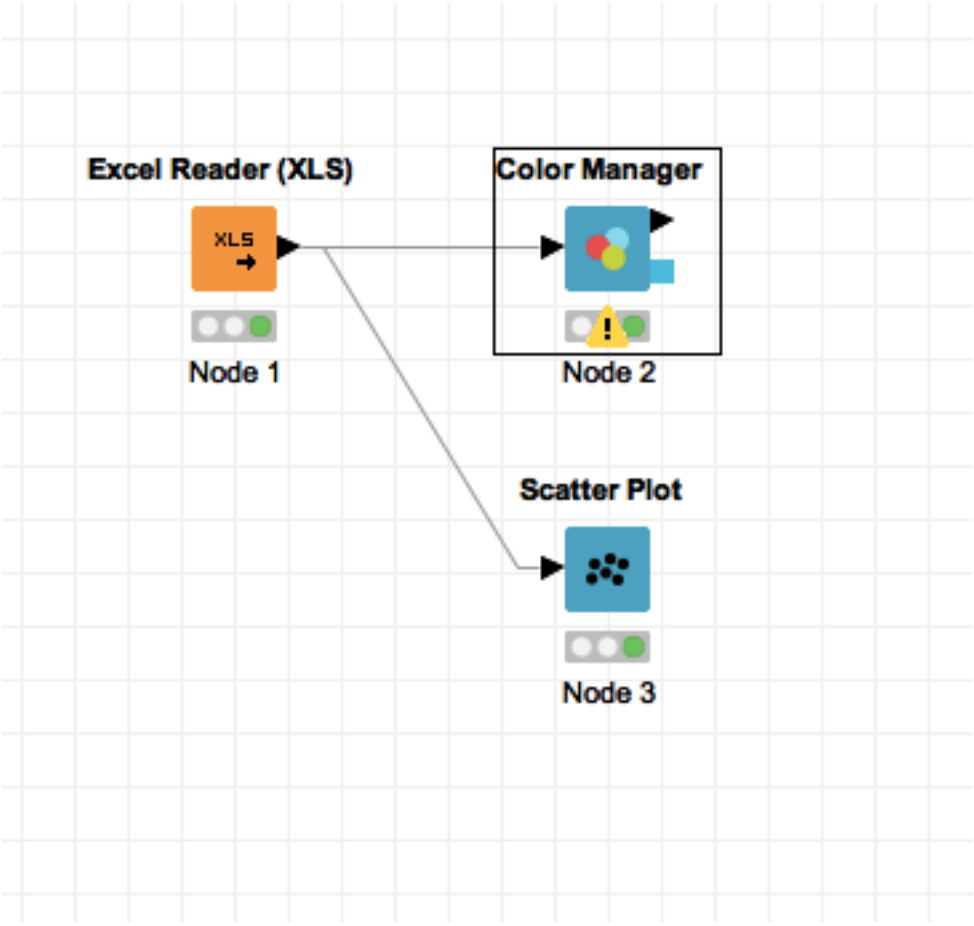
Şekil 5.2.1

Şekil 5.2.1, excel reader operatörünün configure bölümünde cinsiyet veri setinin yüklenmesini ve kolon başlıklarının seçilmesini göstermektedir.



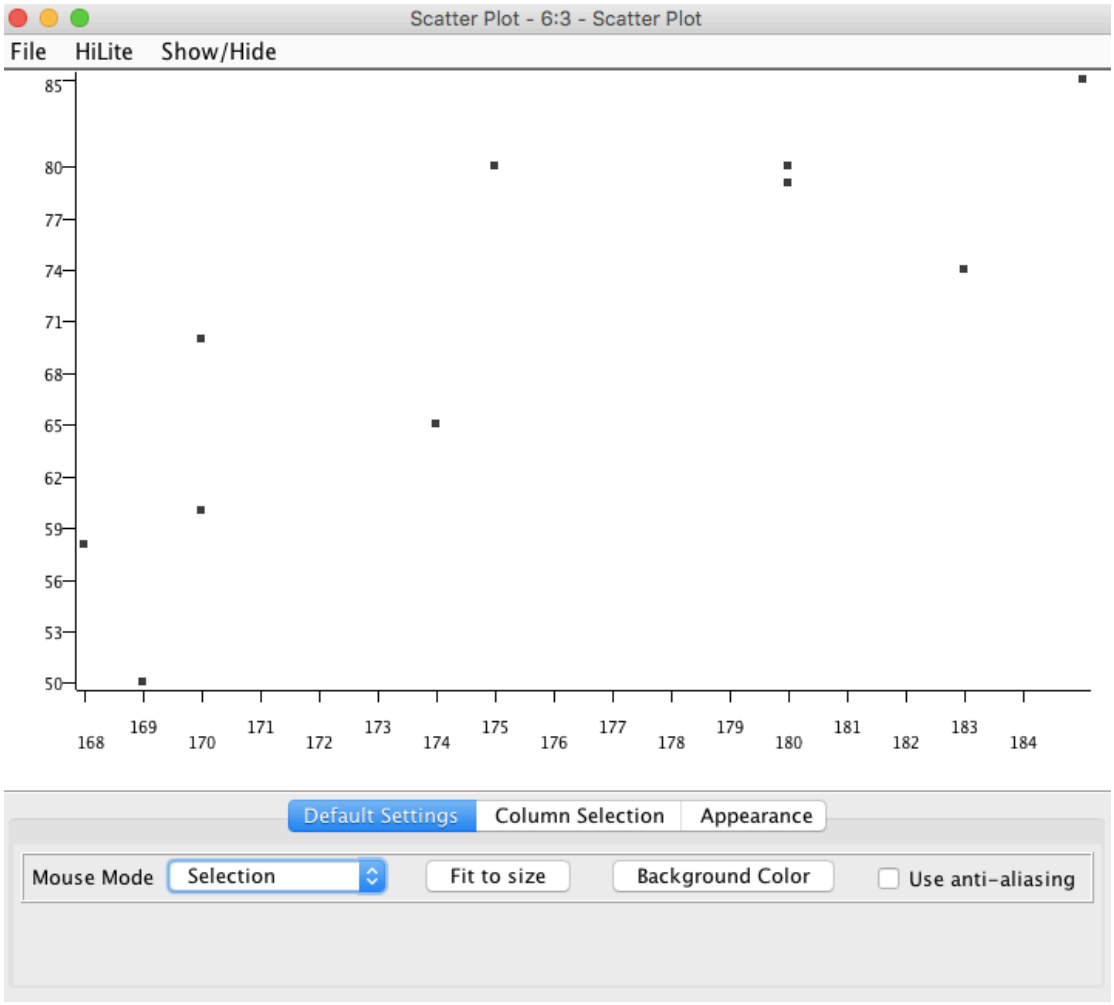
Şekil 5.2.2

Şekil 5.2.2, veri seti yüklenip execute edildikten sonra output table bu şekilde görünebiliyor olmalıdır. Veri içeriği yani boy, kilo ve cinsiyetler değişkenlik gösterebilir.



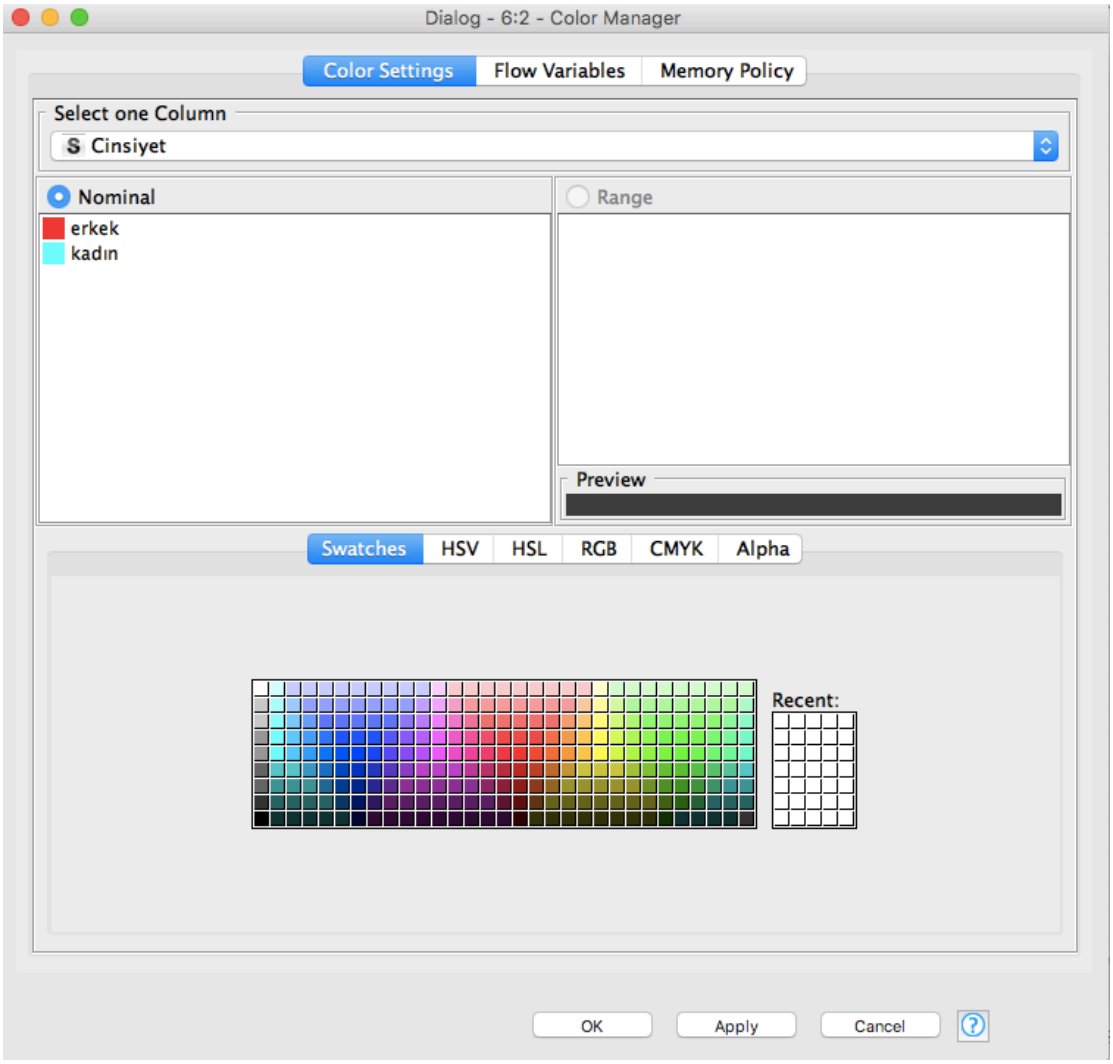
Şekil 5.2.3

Şekil 5.2.3, sisteme color manager ve scatter plot eklenmesini ve bağlantıları göstermektedir. Color manager verilerin renklendirilmesi için kullanılır e bu renklendirme görselleştirme için oldukça yardımcıdır.



Şekil 5.2.4

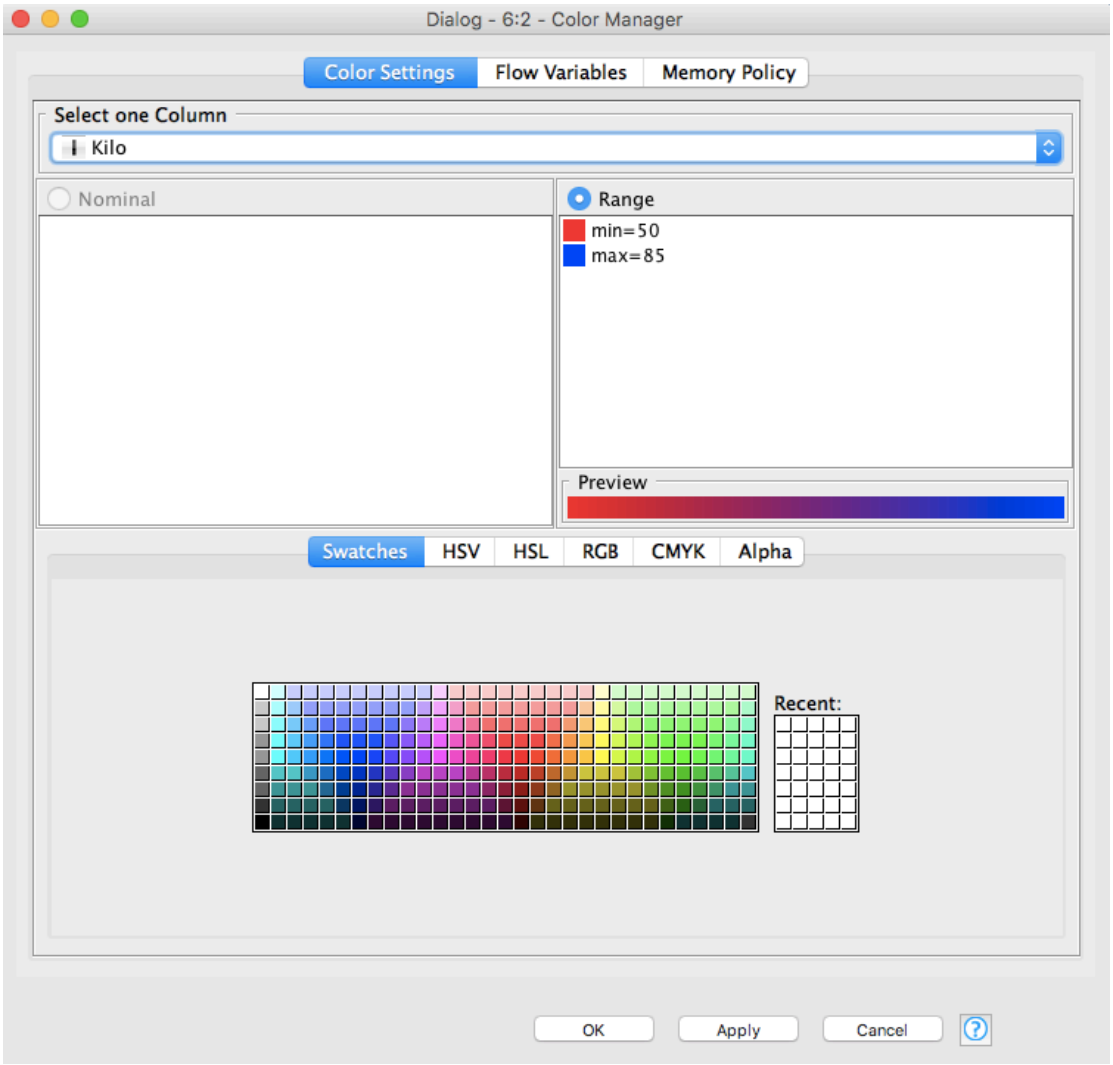
Şekil 5.2.4, Scatter plot daha önceki bölümlerde kullanılmış ve açıklanmıştır. İki boyutlu grafik çizilmesine yardımcı olan scatter plot bu şekilde de görüldüğü gibi renksizdir. Bunu renklendirmek için öncelikle color manager kullanılmalıdır. Color manager eklendiğinde eğer sistem o ana kadar execute edilmemişse "select one column" seçeneğinden renklendirilmek istenilen kolon seçilmelidir. Eğer sistem daha öncesinde çalıştırılmışsa o zaman otamatik olarak sistem cinsiyet kolonunu renklendirmesi gerektiğini bilecektir.



Şekil 5.2.5

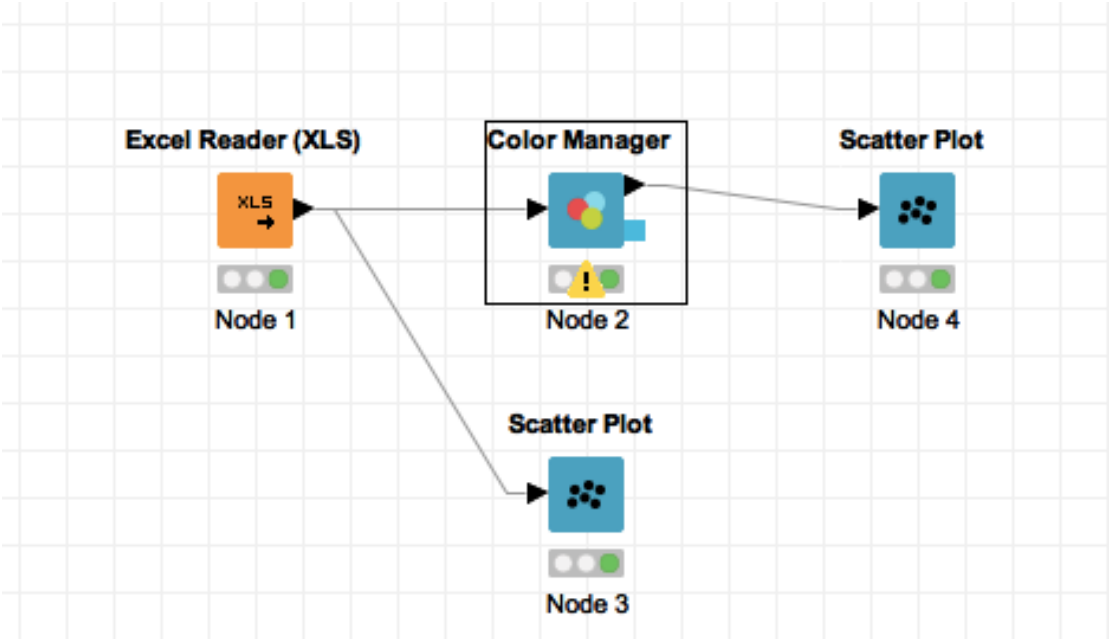
Şekil 5.2.5, color manager operatörünün sistem henüz execute edilmediği için cinsiyet kolonunu otomatik olarak algılamadığı bu yüzden de select one column bölümünden renklendirilmesi gereken kolonun seçileceği ve renk ayarlarının yapılacağı configure penceresini göstermektedir. Erkek kırmızı kadın mavi renk olarak otomatik renk ataması yapmıştır.

Nominal, binominal , polinominal veri tiplerinden buradaki binominal yani iki seçeneği olacak olan örneğin kadın erkek, sigara içiyor içmiyor vb. Şekilde. Veri, nominal ise büyüktür küçüktür ilişkisi, olmaz ve birbirine eklenip çıkarma gibi işlemler yapılamaz. Örneğin kadın küçüktür erkekten denilemez.



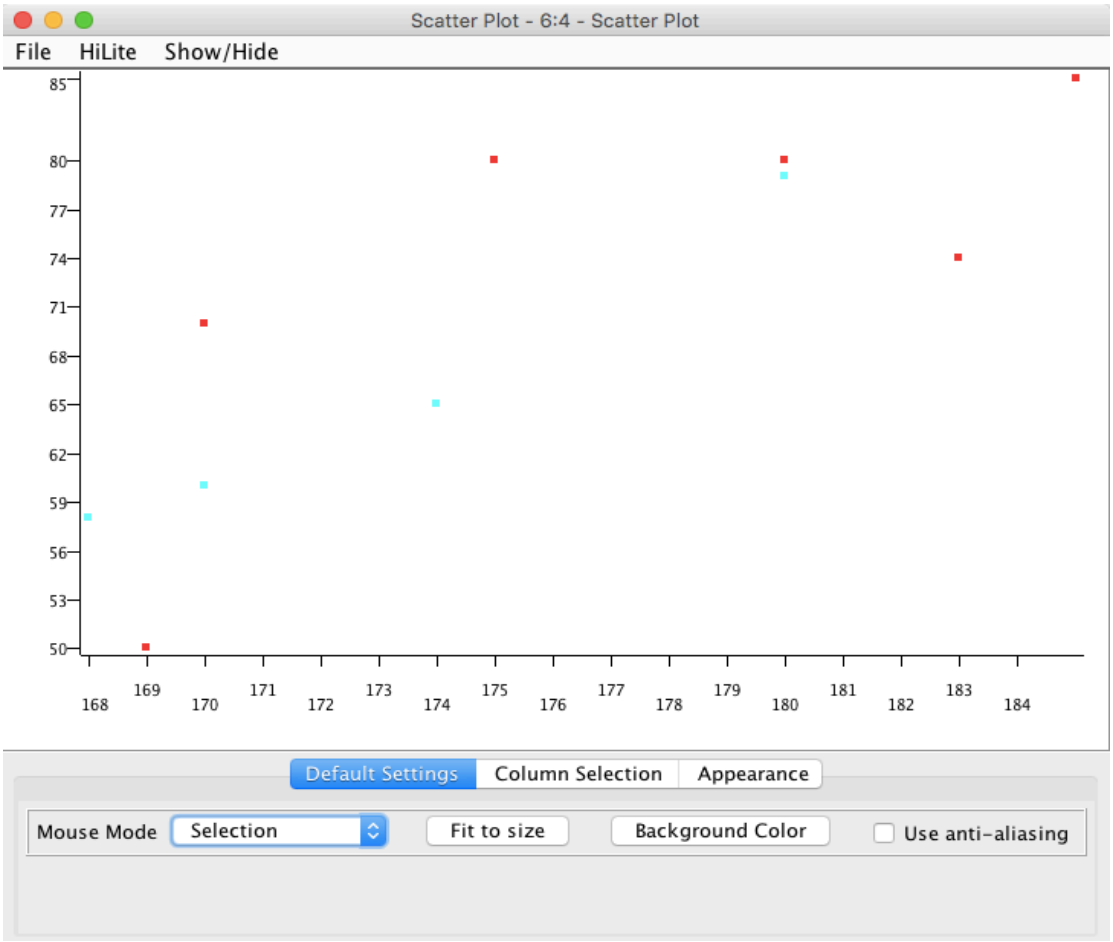
Şekil 5.2.6

Şekil 5.2.6, kilo seçildiği zaman az önceki gibi binominal veri olmaz. Sayısal değerler okunabildiği gibi örneğin 68<75, minimum değeri maximum değeri olabilir. Ayrıca numeric değerler için renk skalası otomatik verilir. Örneğin burada minimum değere (50) kırmızı sonra değer büyüdükçe giderek mavileşen renk alır sonra da maximum değer (85) mavi rengi alır. Binominal 'de ise iki farklı renk olur yani renk geçişi bu şekilde olmaz.



Şekil 5.2.7

Şekil 5.2.7, color manager ile renklendirilmiş cinsiyet kolonundan sonra scatter plot operatörünün eklenmesini ve bağlantılarını göstermektedir.



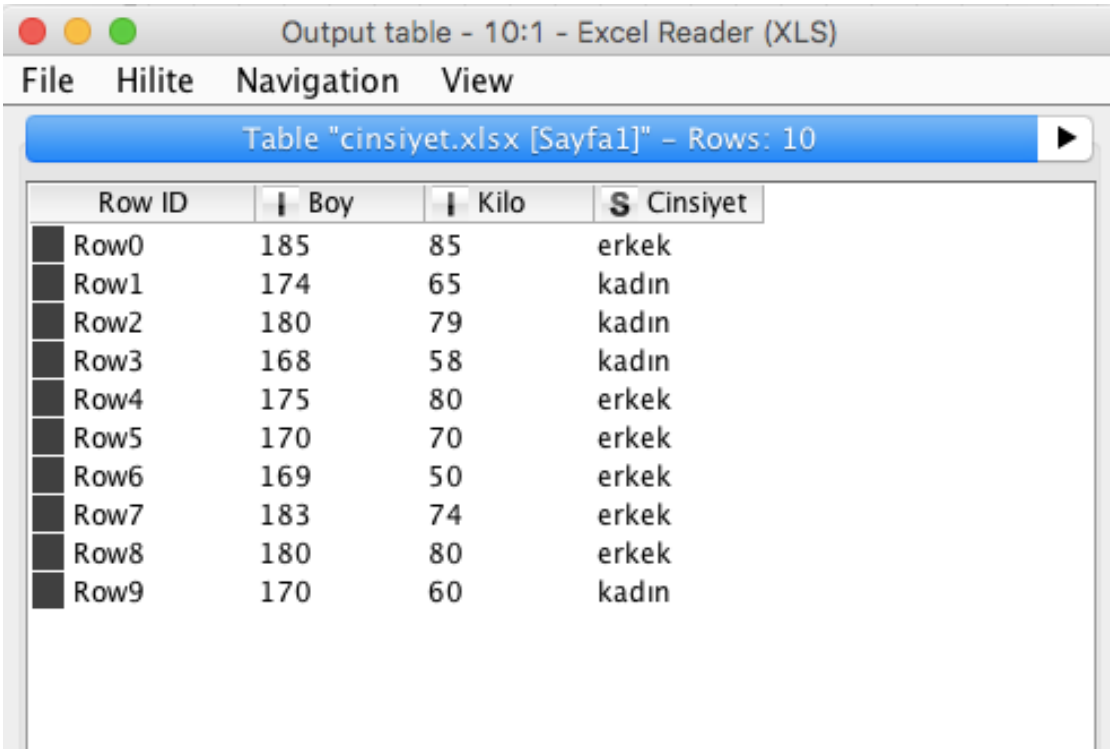
Şekil 5.2.8

Şekil 5.2.8, renklendirilmiş scatter plot grafiğini göstermektedir. Burada maviler kadınları ve kırmızılar erkekleri göstermektedir. Color manager ile bu mavi ve kırmızı renkler cinsiyetlere göre istenildiği şekilde değiştirilebilirdi. Bu örnek basit olduğu için çizilen grafikteki örneklerin renksiz olması çok sorun değil fakat daha karmaşık ve çok sayıda veri içeren bir veri setinde oldukça işe yarayan bir özelliktir.

Bu bölümde iki önemli özellik gösterilmiştir. Color manager renklendirme için, verilerin nasıl renklendirildiği gösterilmiştir. Diğer önemli olan bilgi ise nominal ve numeric verilerin arasındaki farkın ve renklendirilmesinden bahsedilmiştir. Nominal değerler sayı da olabilir. Örneğin İstanbul için 35 ve başka bir il için örneğin Eskişehir için 26 plaka numaraları verilebilir fakat 34 26 dan büyüktür gibi bir kıyaslama yapılamaz. Örneğin erkeğe 1, kadına 0 verilseydi 1, 0'dan büyüktür ya da erkek kadından büyüktür gibi bir kıyaslama yapılamazdı. Numeric değerlerde bu ilişki kurulabilir fakat nominal değerlerde bu ilişki/kıyaslama kurulamaz.

5.3 Scatter Matrix

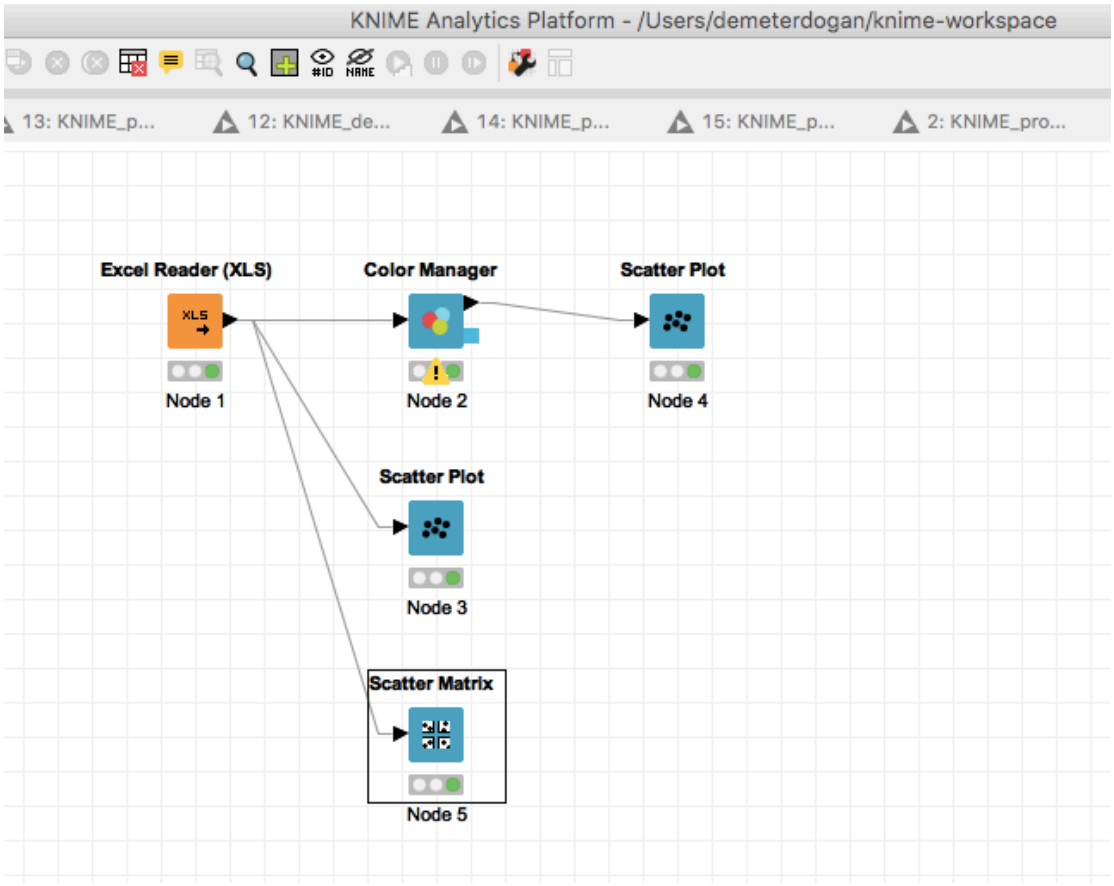
Bu bölümde amaç, veri görselleştirmeye devam edilmesi ve scatter matrix çizilmesinin gösterilmesidir. Bir önceki bölümde kullanılan cinsiyet dosyası kullanılacaktır.



Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	79	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek
Row9	170	60	kadın

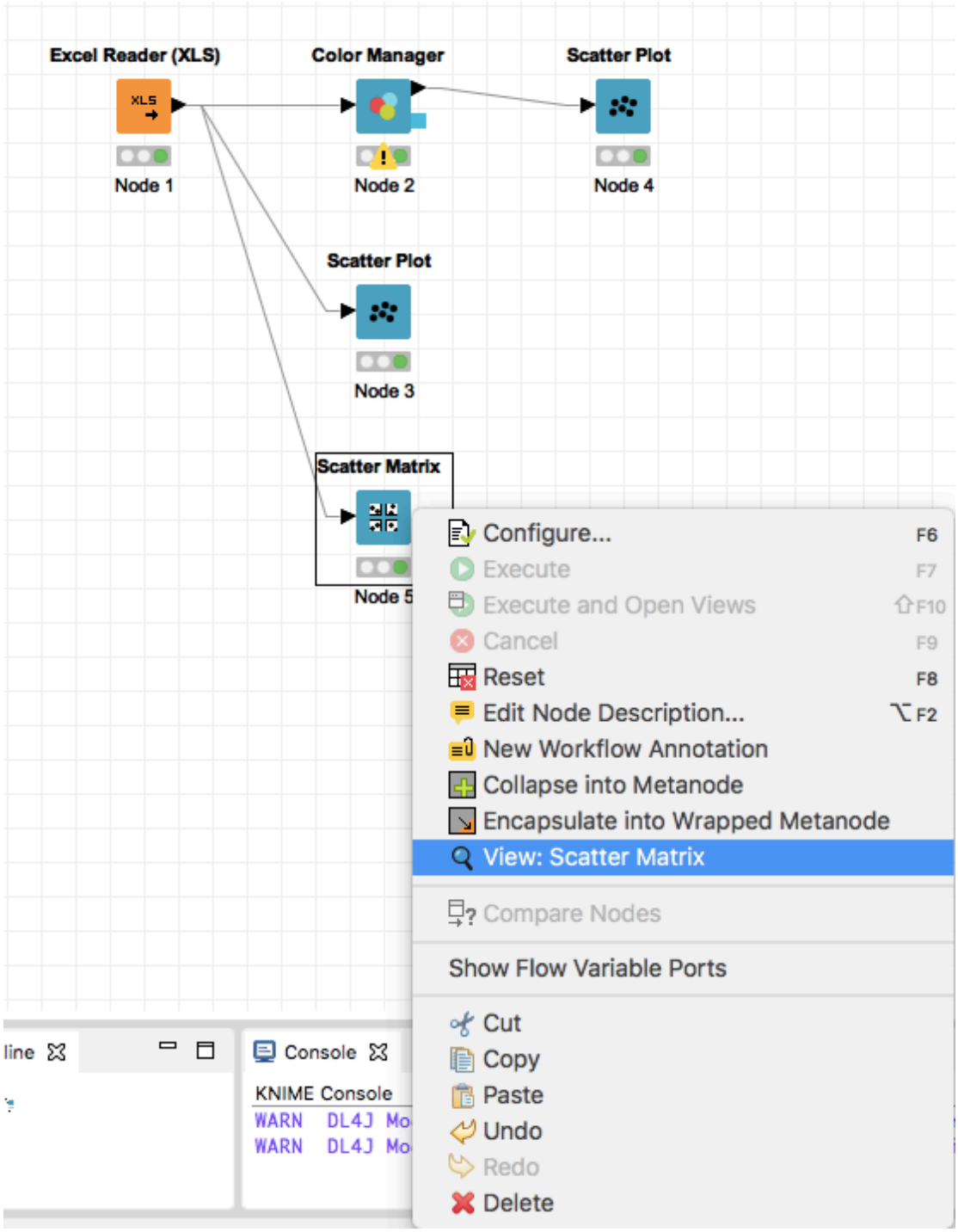
Şekil 5.3.1

Şekil 5.3.1, kullanılan excel dosyasını göstermektedir. Burada değerler istenilen şekilde değiştirilebilir.



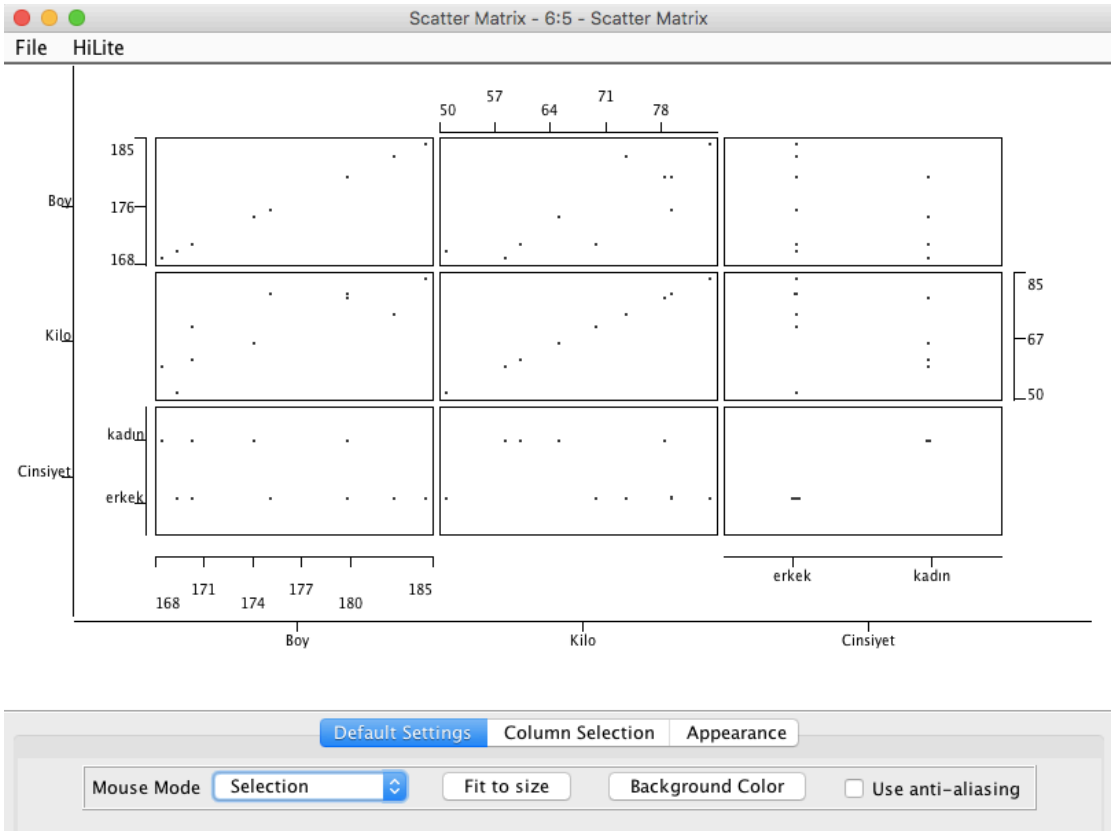
Şekil 5.3.2

Şekil 5.3.2, sisteme scatter matrix operatörünün eklenmesini ve excel reader ile bağlantısını göstermektedir.



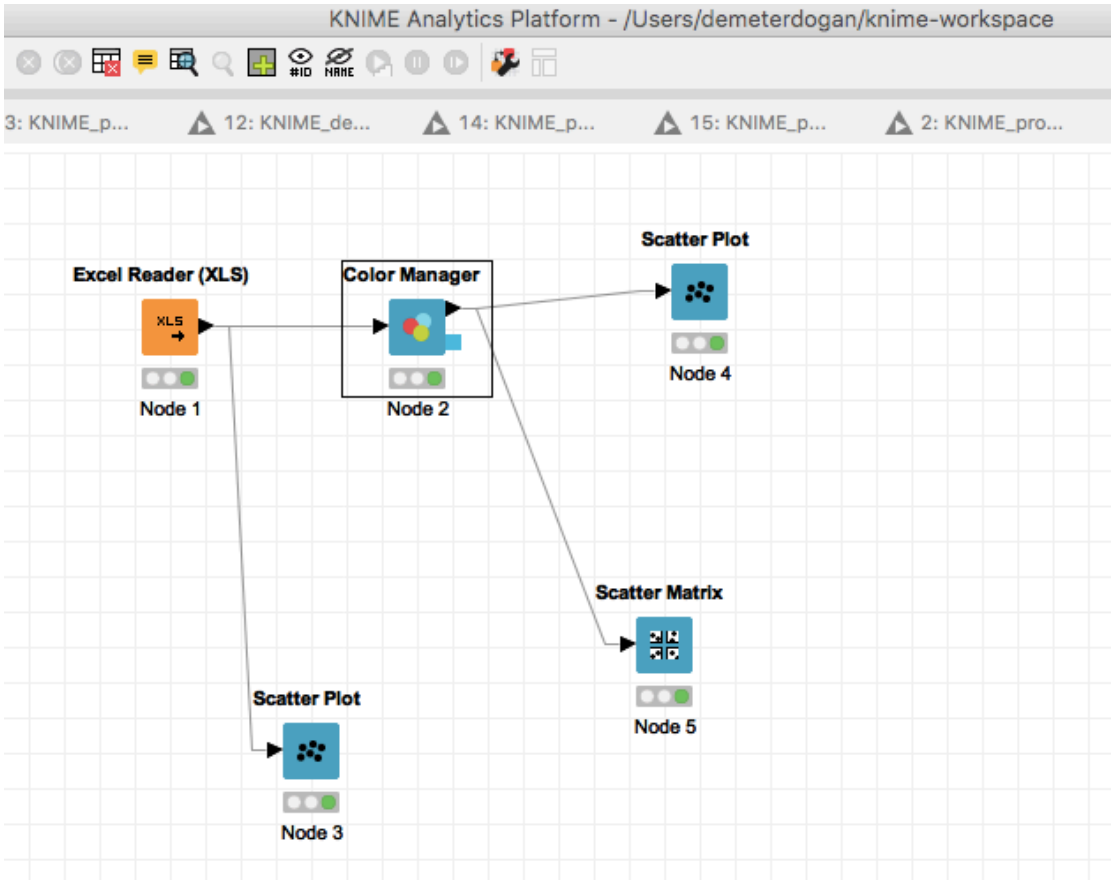
Şekil 5.3.3

Şekil 5.3.3, scatter matrix görsel ekranını yani program run edildikten sonraki sonucunu açmak için basılması gereken tuşun yerini göstermektedir. Scatter matrix node'una sağ tuşla tıklandığında şekilde görülen pencere açılır ve oradan bulunabilir.



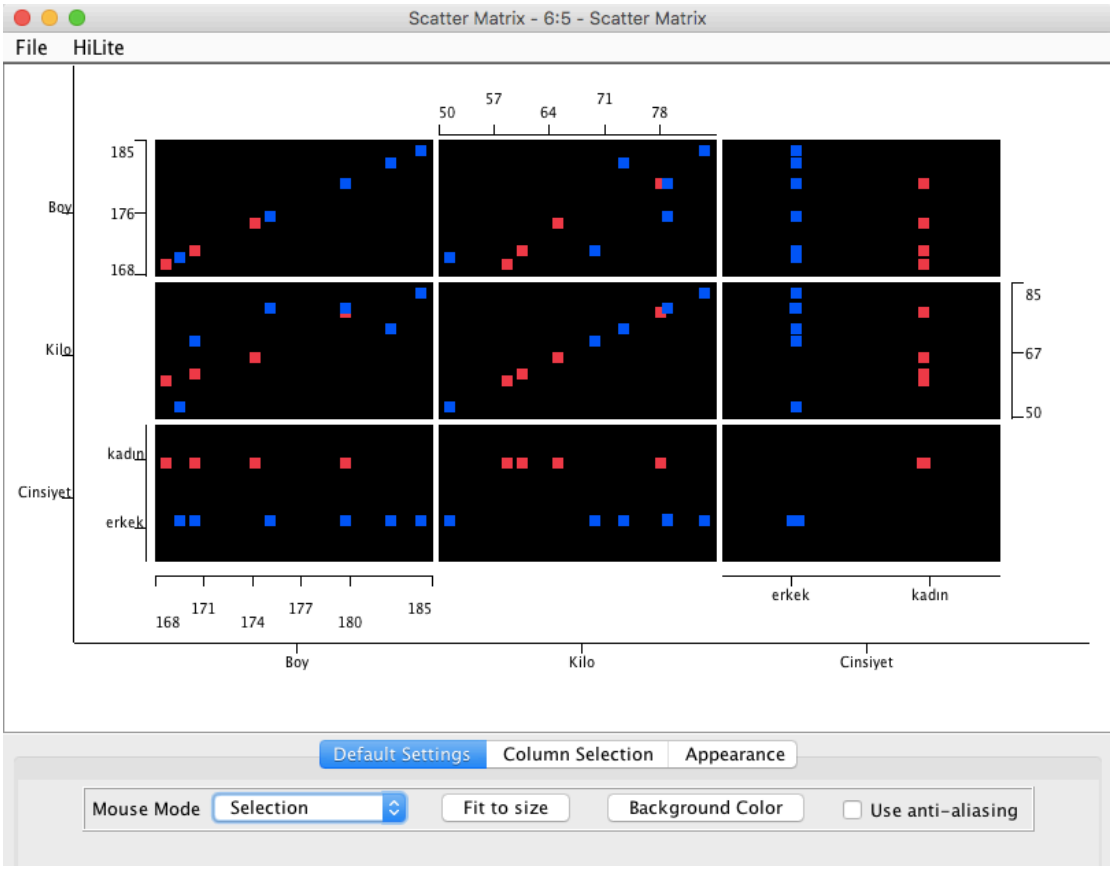
Şekil 5.3.4

Şekil 5.3.4, 'de yatay ve dikey eksene boy, kilo ve cinsiyet kolonlarının yerleştirildiği ve daha sonra bunların birbiriyle olan matrisinin oluşturulduğu görülmektedir. Boy-boy, kilo-kilo, cinsiyet-cinsiyet'e bakıldığında diagonal bir yapı görülmesi gerekmektedir. 65-65, 100-100 şeklinde bir değerlerin birebir olması gerekmektedir. Bu matris verilerin ne şekilde dağıldığını göstermektedir. Örneğin cinsiyet ve boy'da kadın-erkek arasında boy-cinsiyet arasında kadın-erkek ayrışımı belirgin çıkmıştır.



Şekil 5.3.5

Şekil 5.3.5, color manager'a bağlanan scatter plot'ı göstermektedir. Çizilen matrisin renklendirilmiş hali görülmek istenirse color manager kullanılabilir.

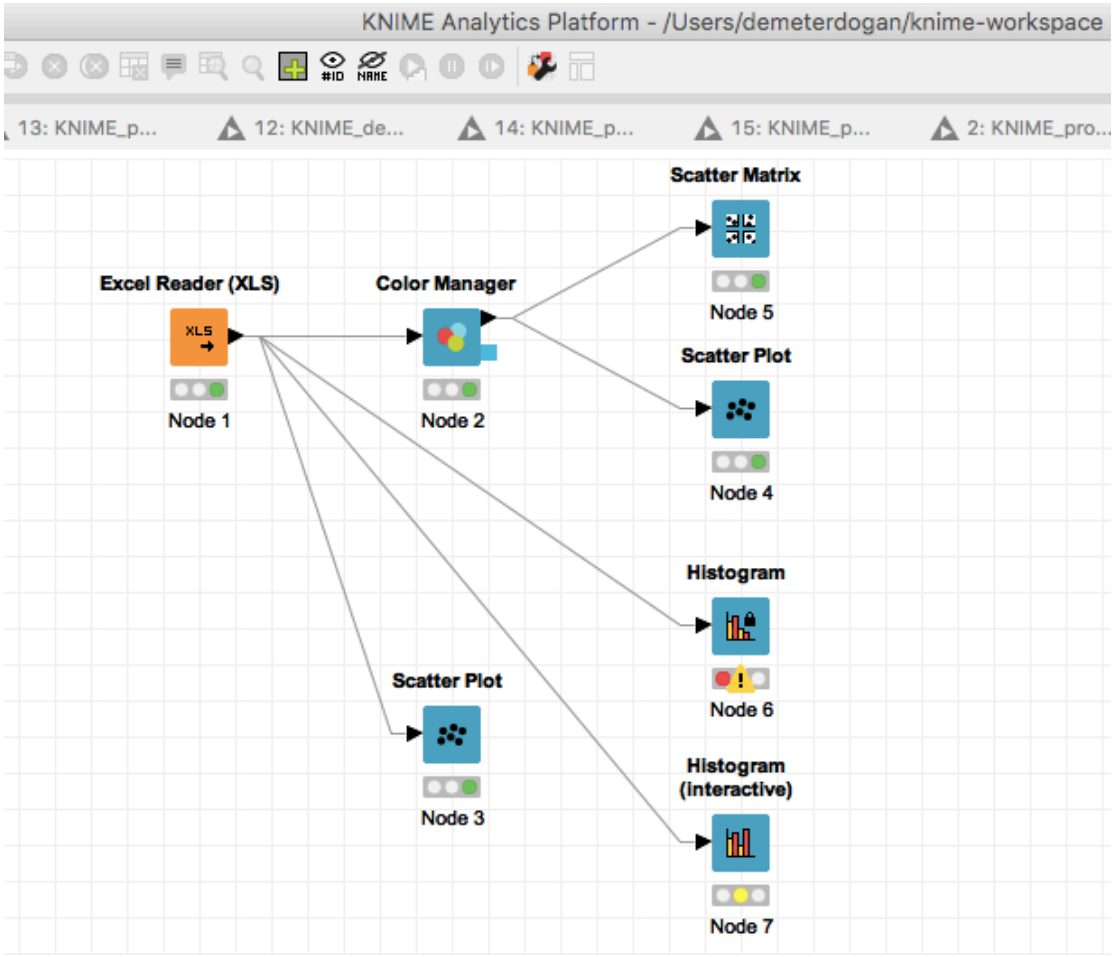


Şekil 5.3.6

Şekil 5.3.6, renklendirilmiş scatter matrix'i göstermektedir. Apperance bölümünden noktaların büyüklüğü değiştirilmiştir. Ayrıca background color da siyah seçilmiş ve noktaların daha da anlaşılması sağlanmıştır. Kırmızılar kadınları, maviler erkekleri belirtmektedir. Bu renklerin seçimi color manager bölünde yapılmıştır. Yukarı bölümde de bahsedildiği gibi renklendirmenin amacı dağılımı görmektir.. Bu örnekte cinsiyet renklendirilmiş olsa da color manager'da boy ya da kilo da seçilerek renklendirilebilir

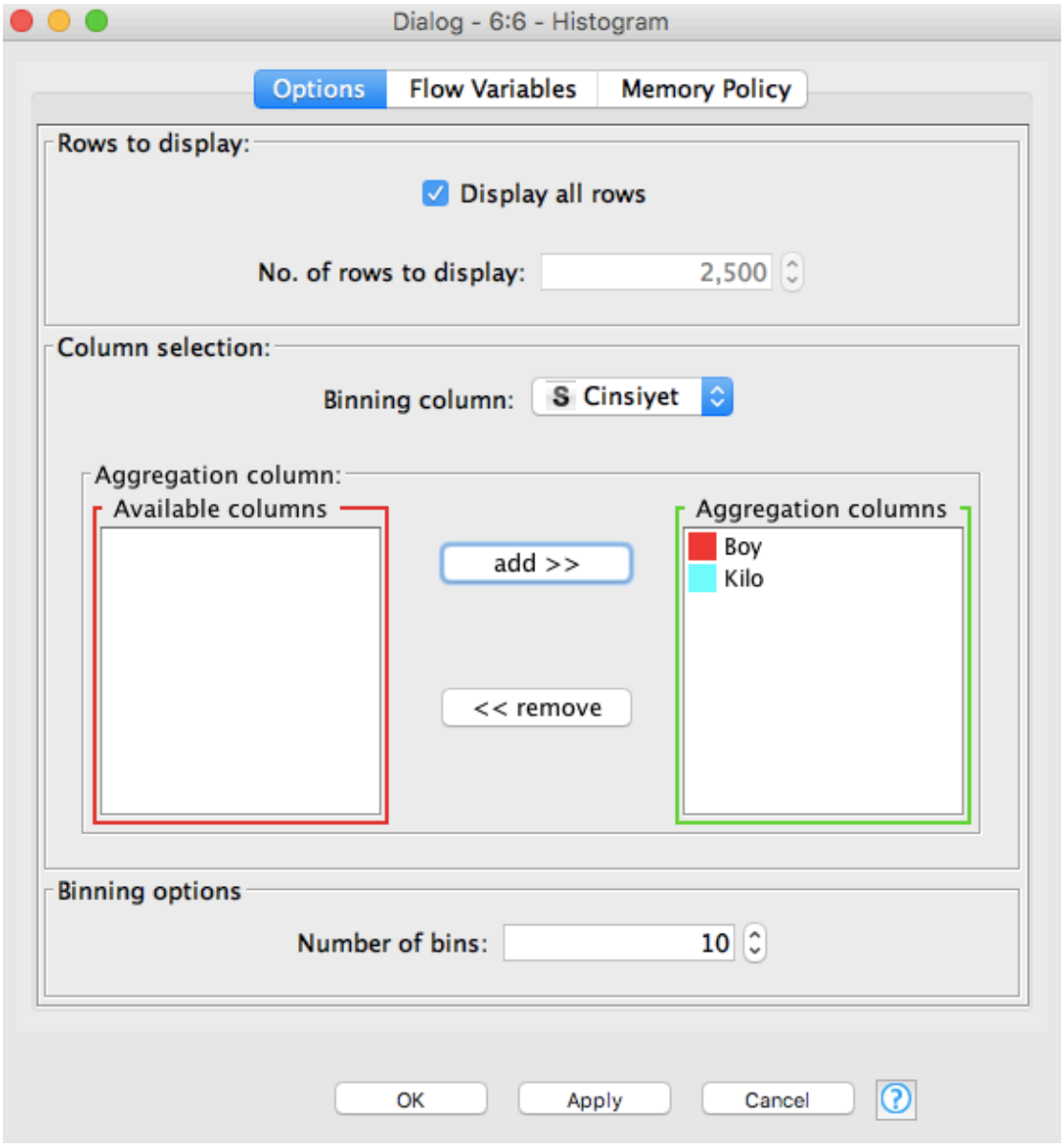
5.4 Görselleştirmeler: Histogram, Pie Chart, Line Chart

Bu bölümde amaç, veri görselleştirmeye devam edilecek ve scatter matrix çizilecektir. Histogram bir diğer görselleştirme araçlarından biridir. Knime'da iki tip histogram bulunmaktadır. Birinci histogram sadece grafik çizilmesine yardım ederken ikinci histogram (interactive) isminden de anlaşıldığı üzere, grafik üzerinde oynama yapılmasına imkan tanımaktadır. Bu yüzden bu örnekte öncelikle normal histogram örneği hızlı bir şekilde gösterildikten sonra interactive olan kullanılacaktır.



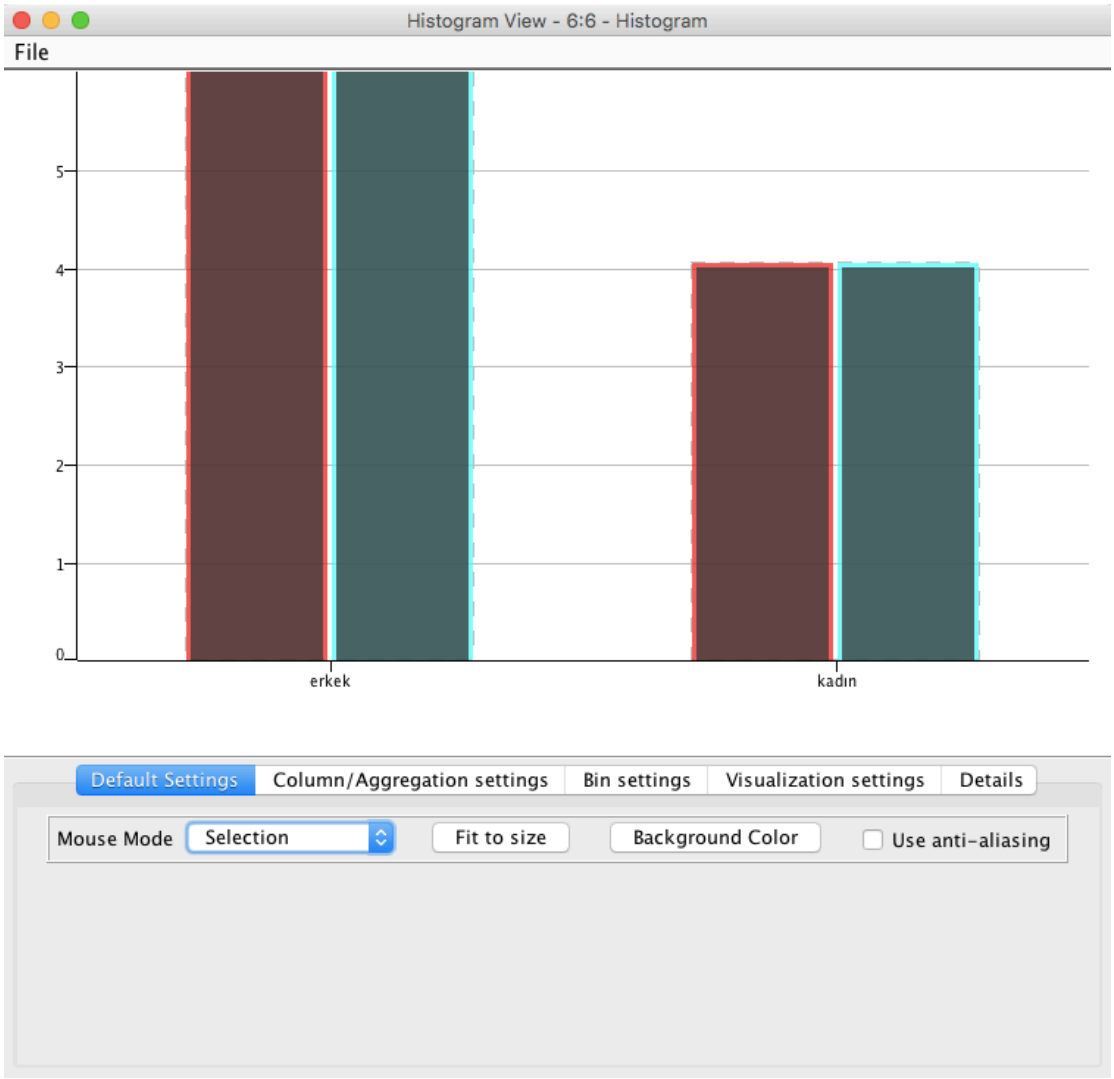
Şekil 5.4.1

Şekil 5.4.1, bir önceki bölümde kullanılan workflow'a histogram ve histogram (interactive) operatörlerinin eklenmiş halini göstermektedir. İnteractive olan bağlantı kurulduktan sonra direk execute edilmeye hazır şekilde altında sarı ışık yanarken histogram configure edilmesi zorunlu olduğunu gösteren kırmızı ışık ve de ünlem işaretini göstermektedir.



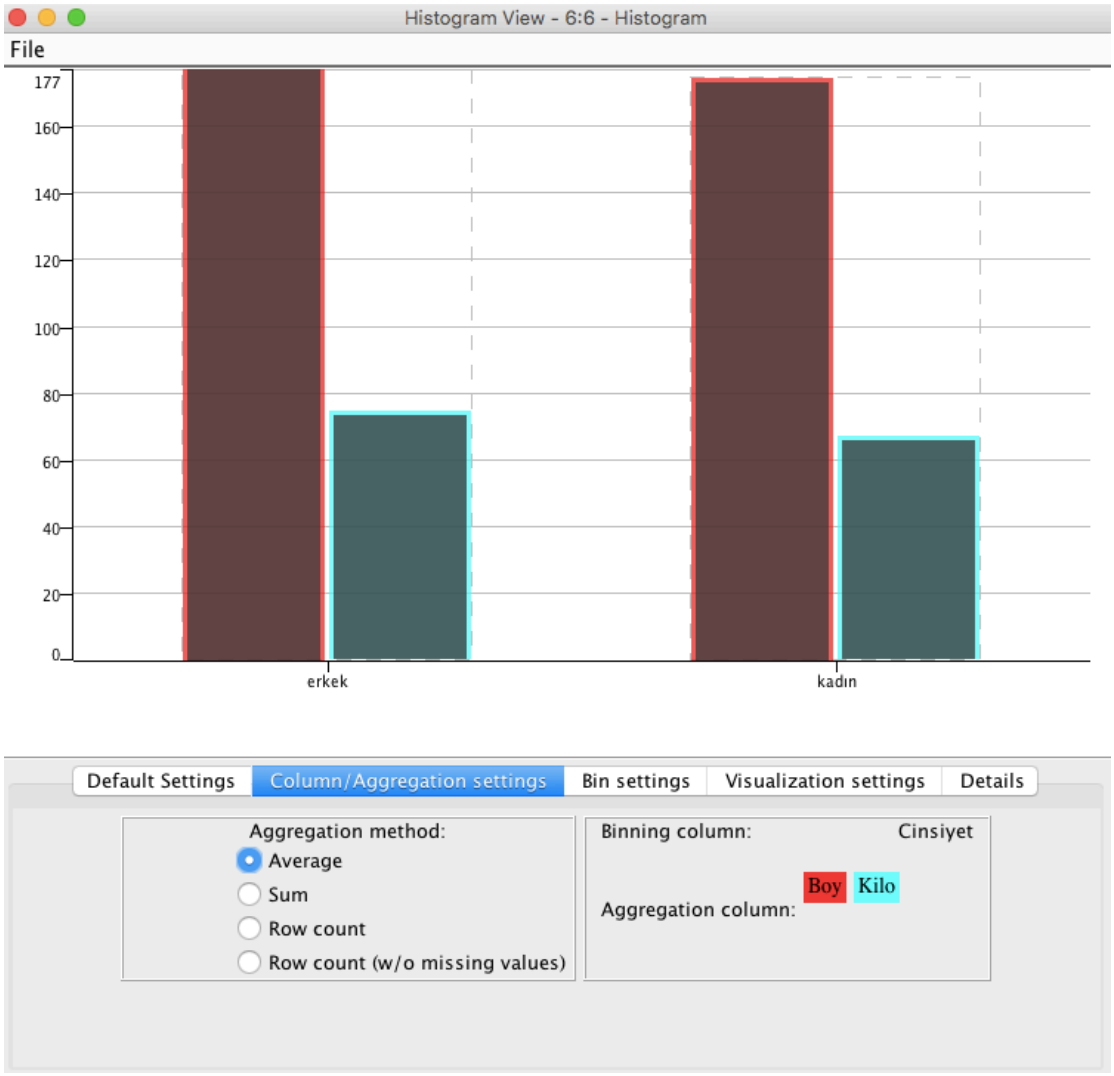
Şekil 5.4.2

Şekil 5.4.2, histogram operatörünün configure penceresini göstermektedir. Binning column bölümünde cinsiyet seçili olması cinsiyete göre birleştirilmesi gerektiği anlamına gelmektedir. Boy ve kilo kolonları dahil edilerek histogram grafiği aşağıda gibi çizilir.



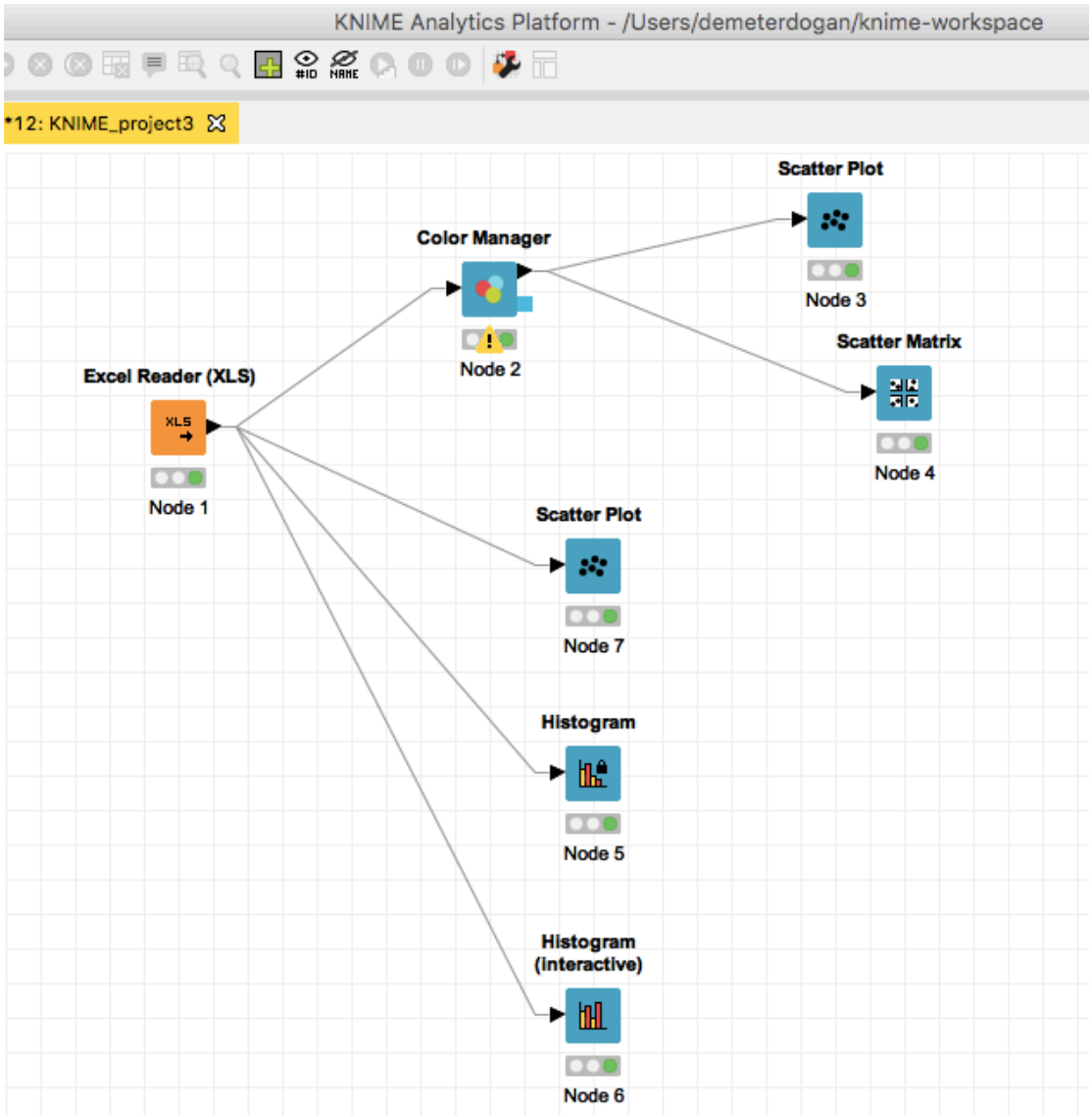
Şekil 5.4.3

Şekil 5.4.3, çizilen histogram grafiğini göstermektedir. Erkek ve kadınlarınki ayrı ayrı çizilmiştir.



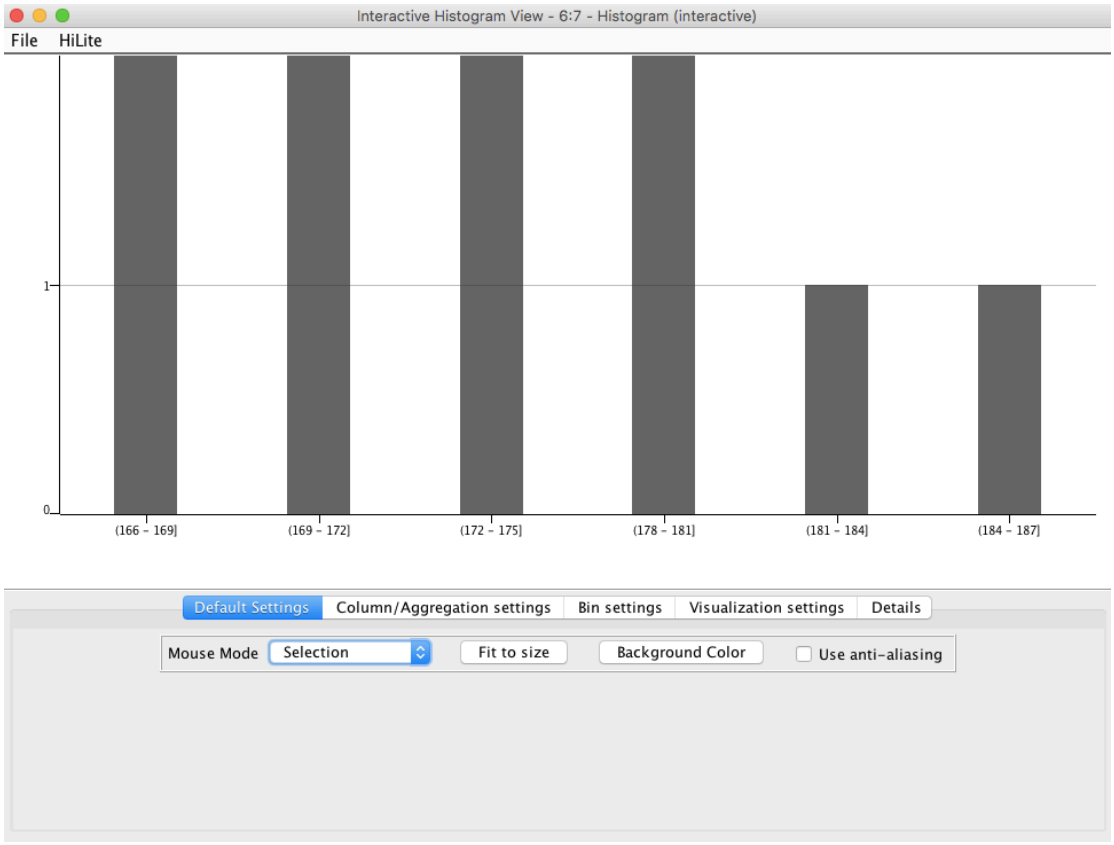
Şekil 5.4.4

Şekil 5.4.4, column setting bölümünden average seçilerek elde edilen grafiği göstermektedir. Bir önceki şekilde row count seçeneği seçili olduğu için erkek ve kadınların boy ve kiloların sayılarının olduğu grafik çizdirilmiştir fakat burada ortalamalarının olduğu değerlerin grafiği çizdirilmektedir. Sum seçeneği boy ve kilolarının toplam değerlerine göre grafiğin çizdirilmesi, row count (w/o missing values) seçeneği ise eksik değerlerle birlikte kadın ve erkekler için boy ve kiloların çizdirilmesi anlamına gelen seçeneklerdir.



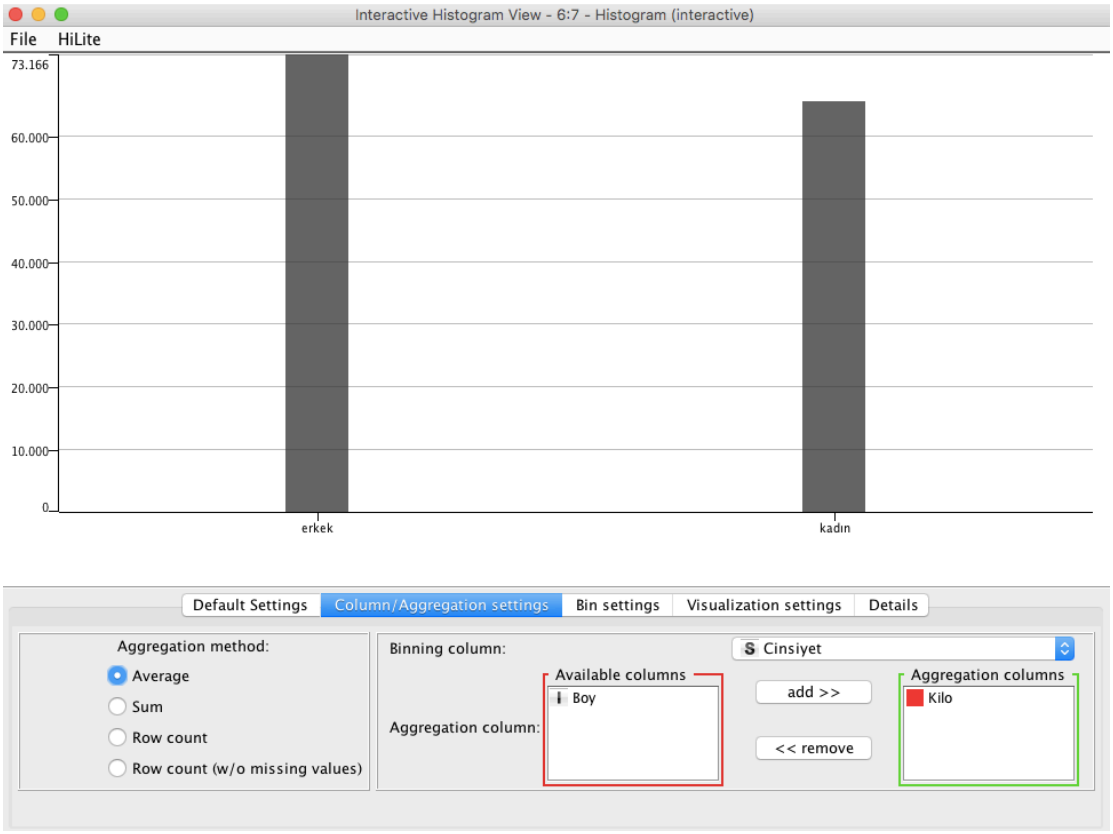
Şekil 5.4.5

Şekil 5.4.5, histogram (interactive) operatörünün sistemdeki bağlantısını göstermektedir.



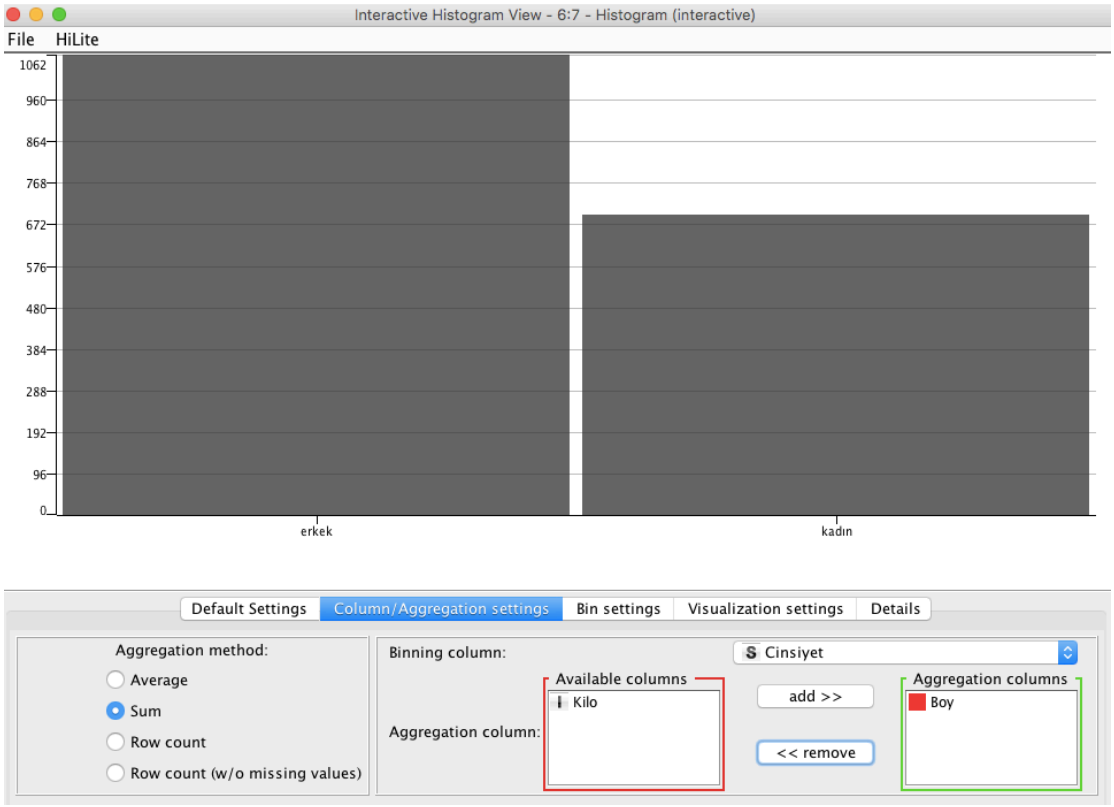
Şekil 5.4.6

Şekil 5.4.6, histogram (interactive) operatöründe herhangi bir ayarlama yapılmadan direk execute edildikten sonra elde edilen grafiği göstermektedir. Ayarları değiştirilerek grafikte çıkan çubuklar (doğrular) da değiştirilebilir. Örneğin burada otomatik olarak bin 10 ve çizilecek kriter olarak "boy" program tarafından seçilmiş ve buna göre aralıklar verilerek çubuklar çizilmiştir.



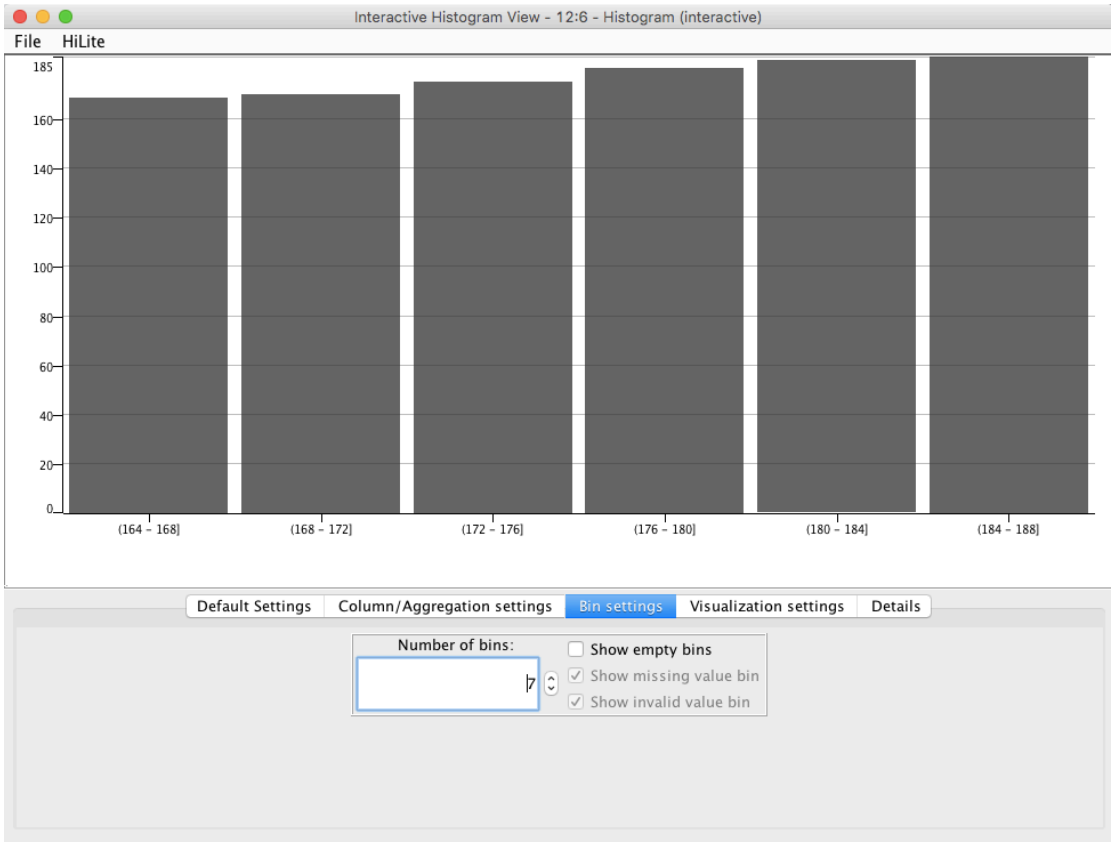
Şekil 5.4.7

Şekil 5.4.7, ayarları değiştirilerek elde edilen grafiği göstermektedir. Binning column olarak cinsiyet seçilmesi oluşacak grafikte cinsiyete göre gruplanması anlamına gelmektedir. Aggregation method olarak da average seçilmesi oluşacak grafikte cinsiyet ile ayrılan grupların seçilecek kolonun ortalamasını vermesi anlamına gelmektedir. Aggregation column olarak da kilo seçilmiştir. Yani kadın ve erkeklerin kilo ortalamalarına göre grafik çizdirilmiştir.



Şekil 5.4.8

Şekil 5.4.8, veriyi cinsiyete göre ayırarak boy toplamlarına göre grafik çizdirilmiştir. Veriyi tanımak için bu şekilde görselleştirme yapılabilir. Veri setinde seçilen kolondaki bilgiye göre bölünerek (örneğin burada cinsiyet kolonundan kadın ve erkek olarak bölünmesi) veri değerlerinin average (ortalaması), sum (toplamı), row count (sıra toplamları) ve row count (w/o missing values) (sıra sayılarının eksik değerlerle ve eksik değersiz toplamı) grafikleri çizdirilebilmektedir.

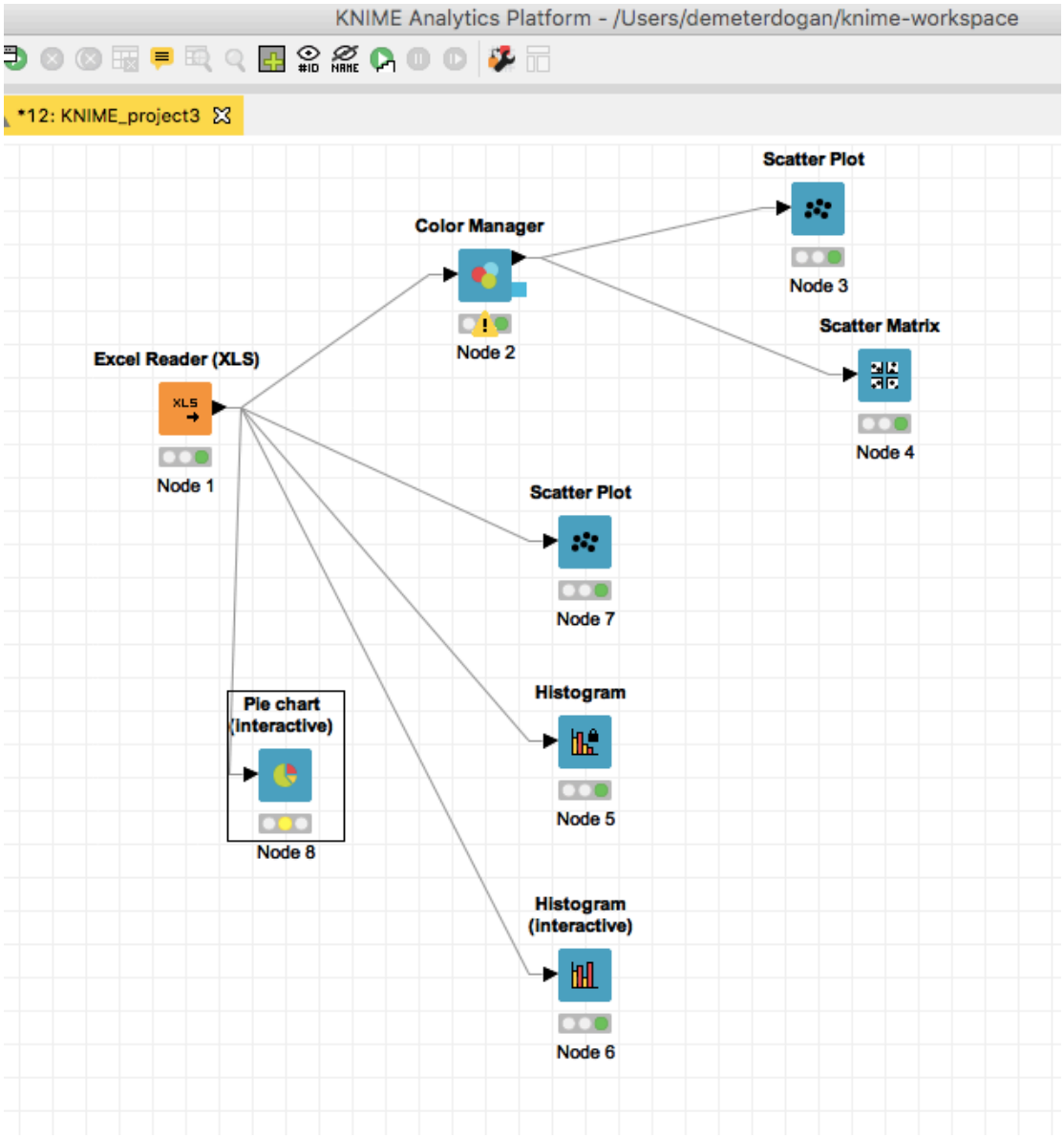


Şekil 5.4.9

Şekil 5.4.9, boy ortalamalarının (average) seçilmesi ve binning kolonu olarak da boy seçilmesiyle oluşturulmuş bir grafiktir. Bin settings bölümünde de görüldüğü gibi 7 seçilmiştir. Boy gruplarındaki artış kiloda da artışa neden olabilmektedir ve bu grafikte çubukların uzunluklarının artmasından anlaşılmaktadır.

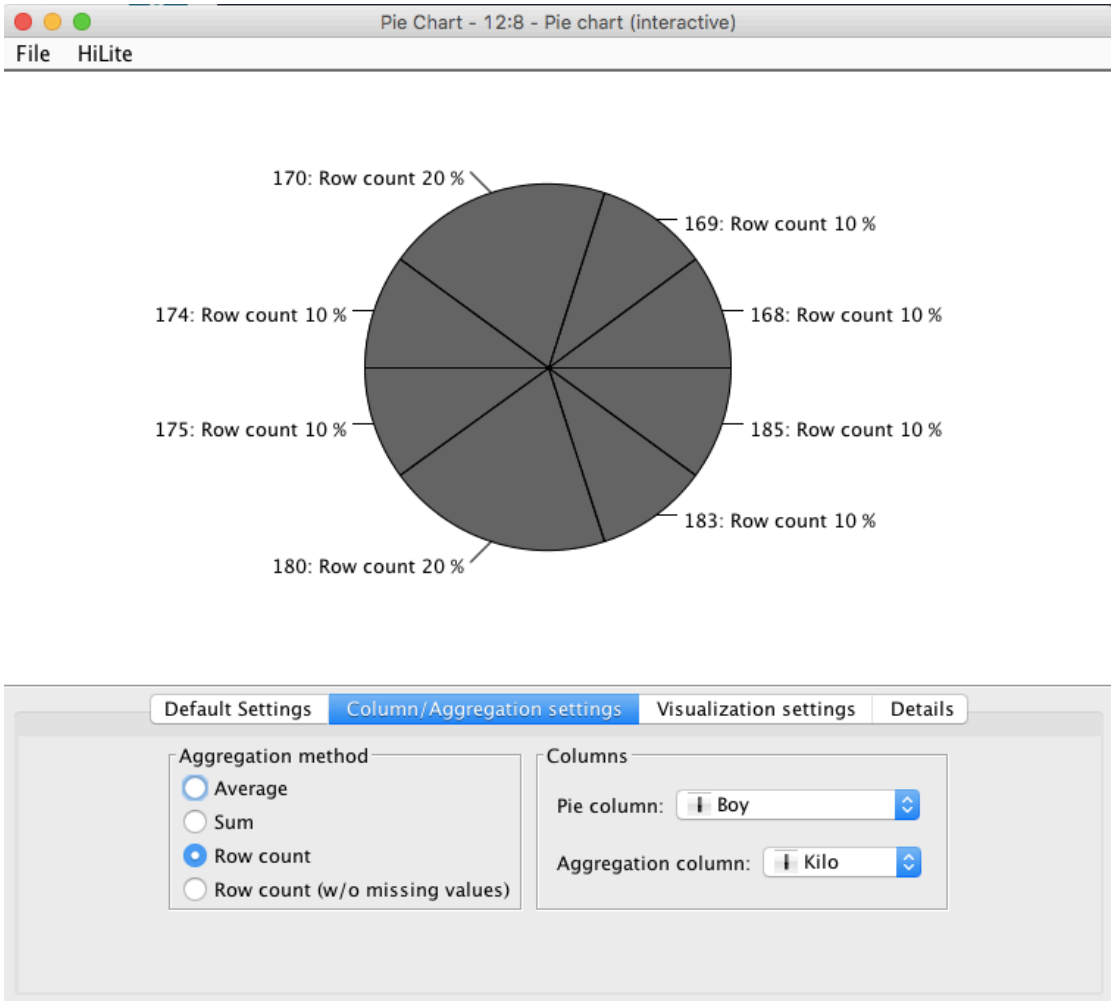
Şekil 5.4.9'da görülen grafikteki kadın ve erkek grafiğinin eşit olmaması durumu imbalance data yani dengesiz verilerin olduğu anlamına gelmektedir. Bu aşamada yani bu bölümde henüz dengesiz veri seti çok önemli değildir. İlerideki bölümlerde açıklama yapılarak anlamlandırılacaktır.

Node repository penceresinden views dosyasının içerisinde utality bölümünde bir çok grafik çizdirme yöntemi bulunmaktadır. Bir tane örnek olarak pie chart (interactive) sisteme eklenerek grafiği gösterilecektir.



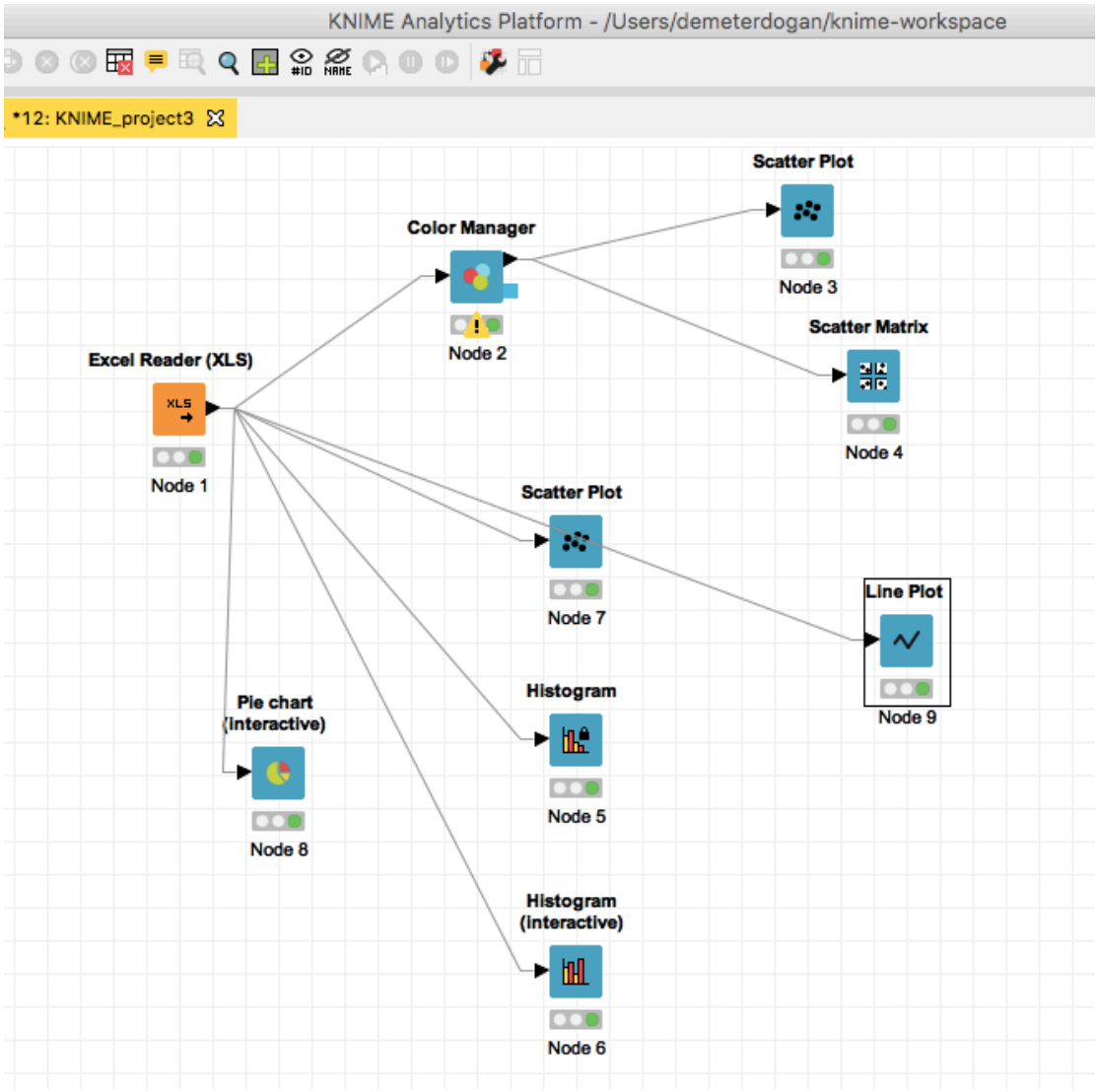
Şekil 5.4.10

Şekil 5.4.10, pie chart (interactive) operatörünün sisteme eklenmiş ve bağlantısının yapılmış halini göstermektedir. İnteractive olduğu için configure etmeden de direk olarak execute edilebilir.



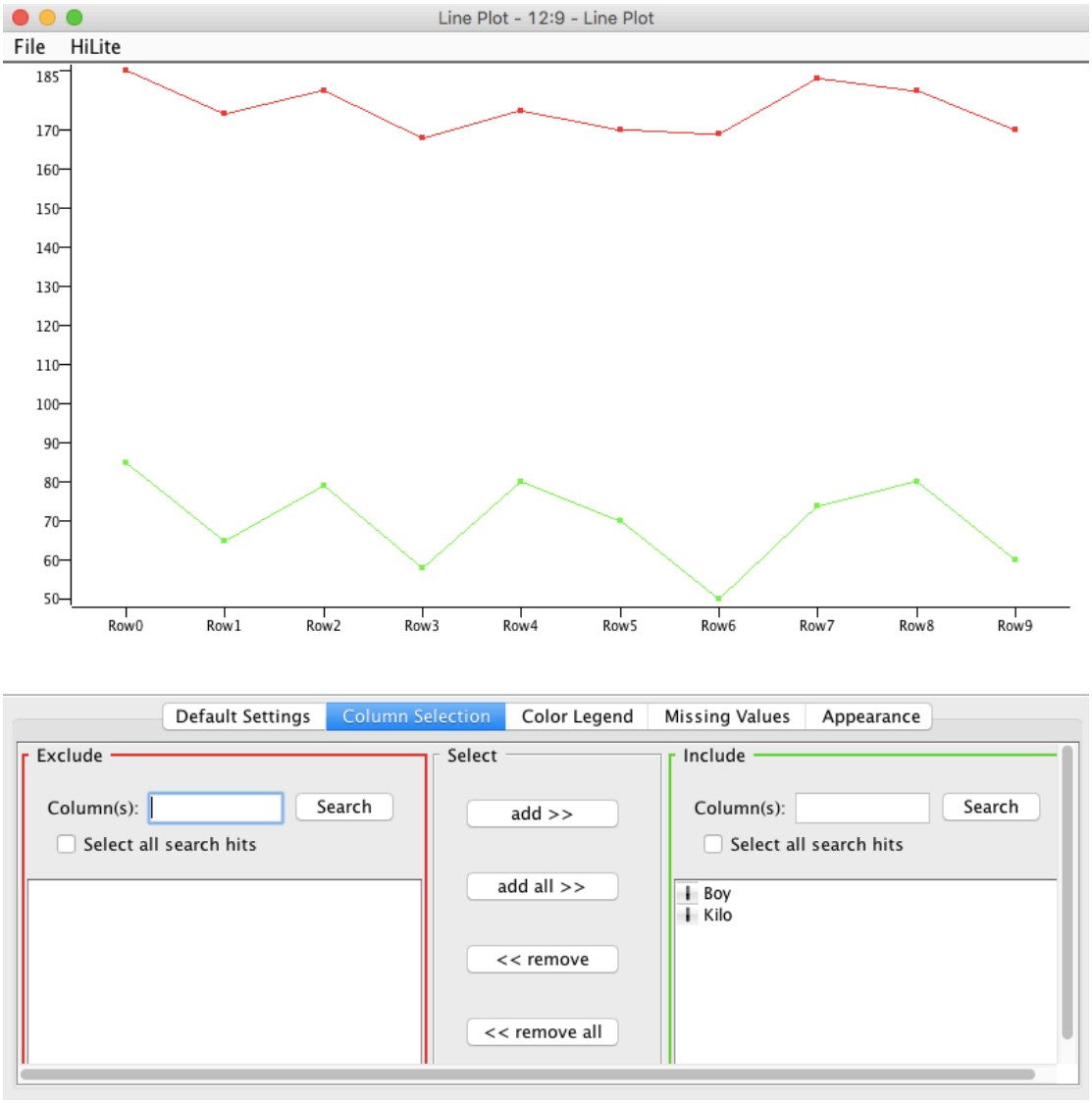
Şekil 5.4.11

Şekil 5.4.11, sistemin execute edildikten sonra otomatik olarak çizdiği pie chart'ı göstermektedir. Satır sayılarına göre çizilen grafik ortalama (average) ya da toplama göre de (sum) çizdirilebilirdir.



Şekil 5.4.12

Şekil 5.4.12, diğer bir örnek olarak line plot operatörünün sisteme eklenmesini ve bağlantısını göstermektedir.



Şekil 5.4.13

Şekil 5.4.13, sistem execute edildikten sonra otomatik oluşan grafiği göstermektedir. Burada görülen grafikte herhangi bir configure (değişim) yapılmamıştır. İstenilirse column selection bölümünden sadece boy ya da sadece kilo seçilerek onun grafiği oluşturulabilir. Yeşil renk kadınların, kırmızı renk ise erkeklerin grafiğidir. Bazı aralıklarda paralel artış görülürken bazı aralıklarda da ters orantının olduğu görülebilir.

Grafik çizme aslında farklı bir uzmanlık alanıdır fakat bu bölümde amaç veri setindeki sınıfların farklarının rahatça grafik üzerinde görülebilmesini göstermekti bu yüzden en çok kullanılan bir kaç grafik örneği gösterildi.

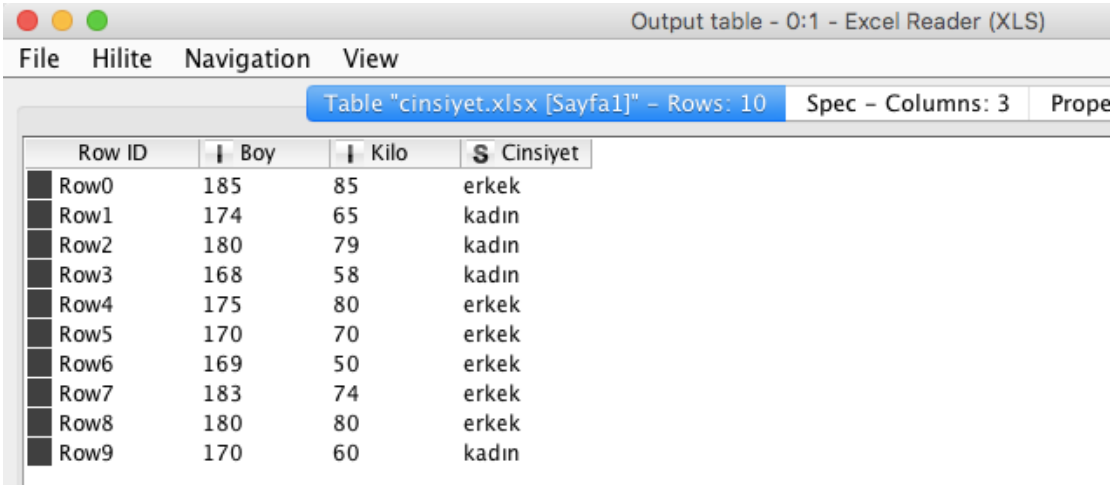
6.BÖLÜM: VERİYİ İŞLEMEK (ETL, PREPROCESSING SÜREÇLERİ)

6.1 SATIR FİLTRELEME (ROW FILTERING)

Bu bölümde amaç veriyi işlemek ve bunun ilk aşamasında yani preprocessing'de veriyi filtrelemeyi göstermektir.

Filtreleme, eksik satırlardan kurtulmak, kirli verileri ya da gürültülü verileri temizlemektir. Temizlenmesindeki amaç makinenin öğrenme başarısını yükseltmek ve sistemdeki bozulmalardan kaçınmadır.

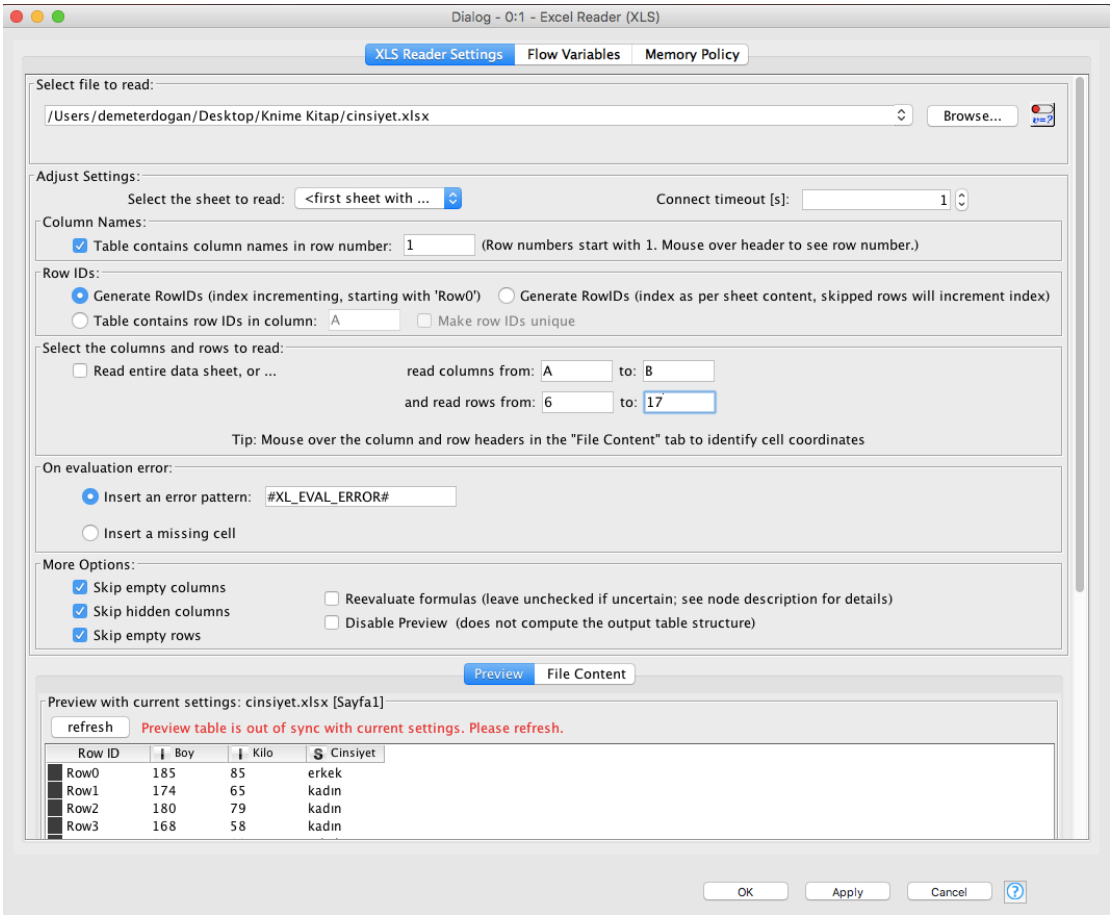
İlk filtreleme veriyi sisteme yüklerken oluşturulabilir. Daha önceki bölümlerde kullanılan cinsiyet veri seti bu bölümde de örnek olarak kullanılacaktır.



Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	79	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek
Row9	170	60	kadın

Şekil 6.1.1

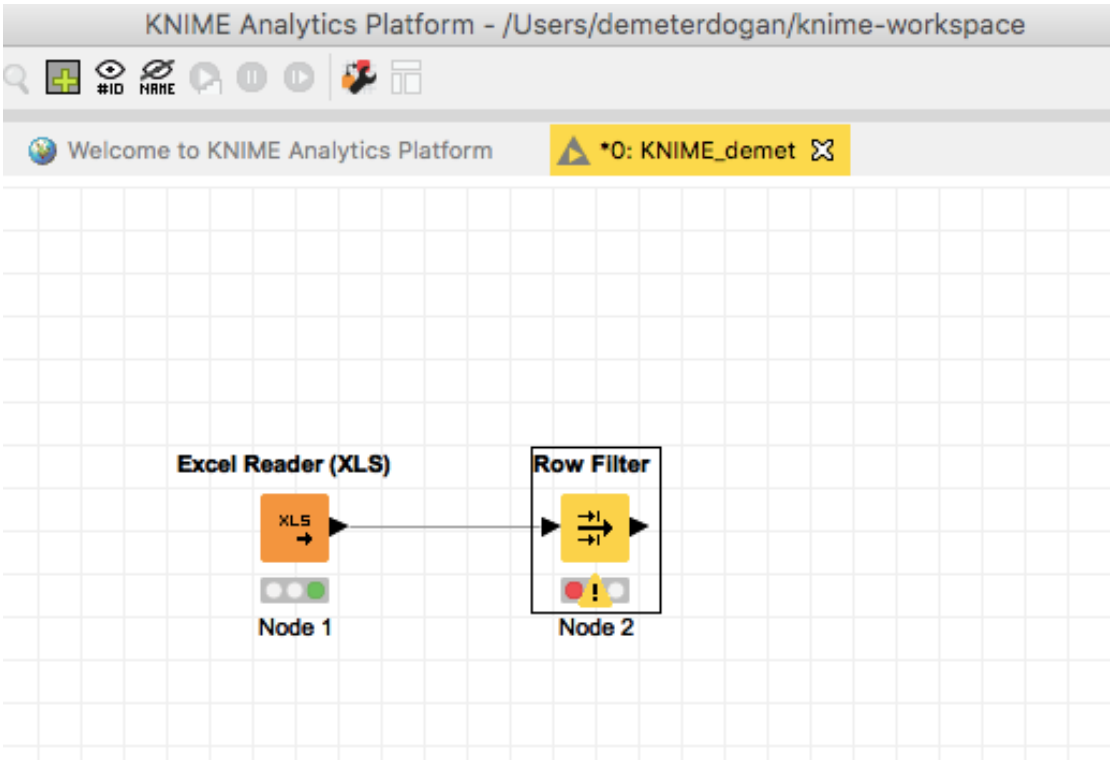
Şekil 6.1.1, bu bölümde kullanılacak örnek cinsiyet veri setini göstermektedir. Değerler istenilen şekilde değiştirilip veri seti uzatılabilir ya da azaltılabilir.



Şekil 6.1.2

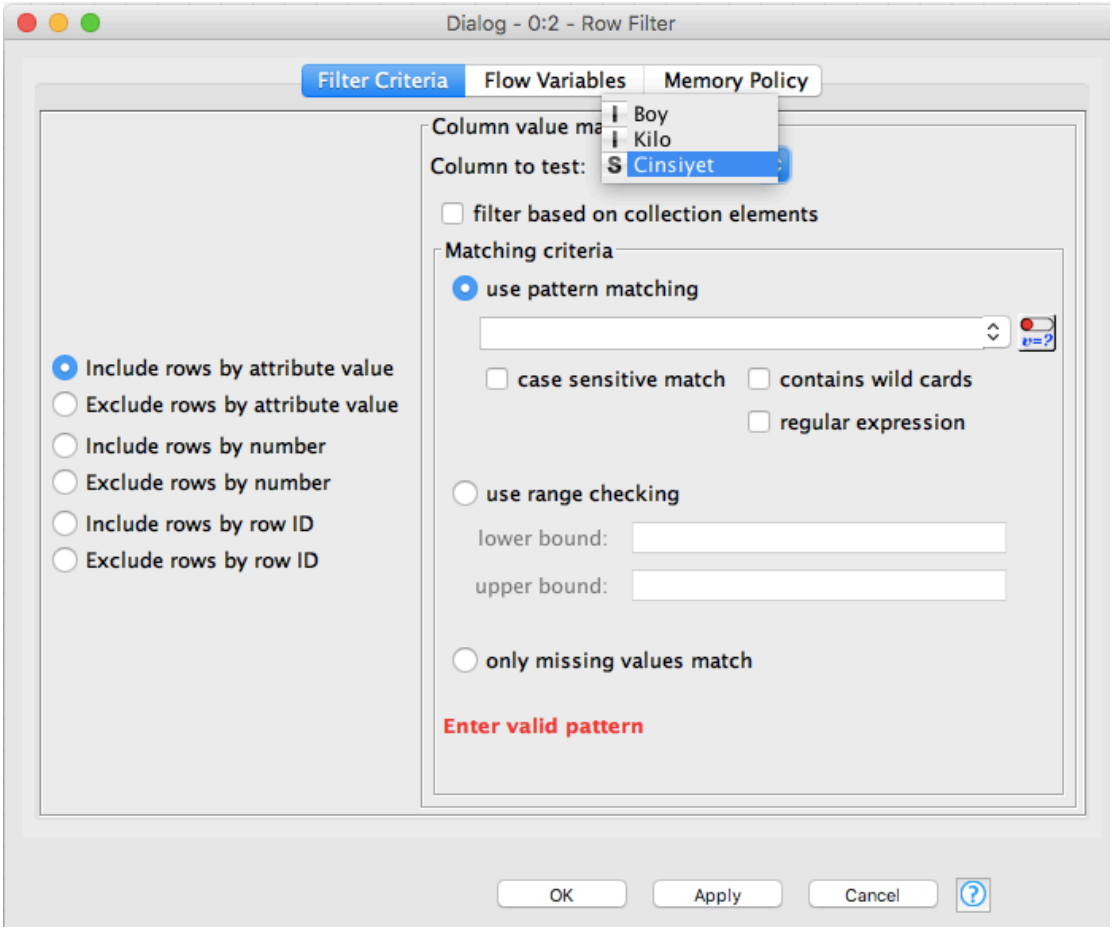
Şekil 6.1.2, cinsiyet veri setinin excel reader ile workflow'a eklemeyi göstermektedir. Select the columns and rows to read: bölümünden read columns from .... to .... bölümünden istenilen kolon seçilebilir. Bu seçim ile örneğin A-C demek sadece A ile C arasındaki tüm kolonları alınması anlamına gelmektedir. Read rows from ... to ... ise alınması istenilen satır aralıklarını göstermektedir. Fakat bu örnekte tüm veri seti alınacaktır. Table contains column names in row number 1 bölümü seçilerek ilk satırdaki isimlerin kolon başlığı olması istenmiştir. Refresh butonuna basarak bu değişiklikten sonra yeniden yüklenen veri execute edilmeye hazır hale gelmiştir.

Knime içerisinde filtrelemek için bazı hazır özellikler bulunmaktadır. Bunlardan biri row filter'dır.



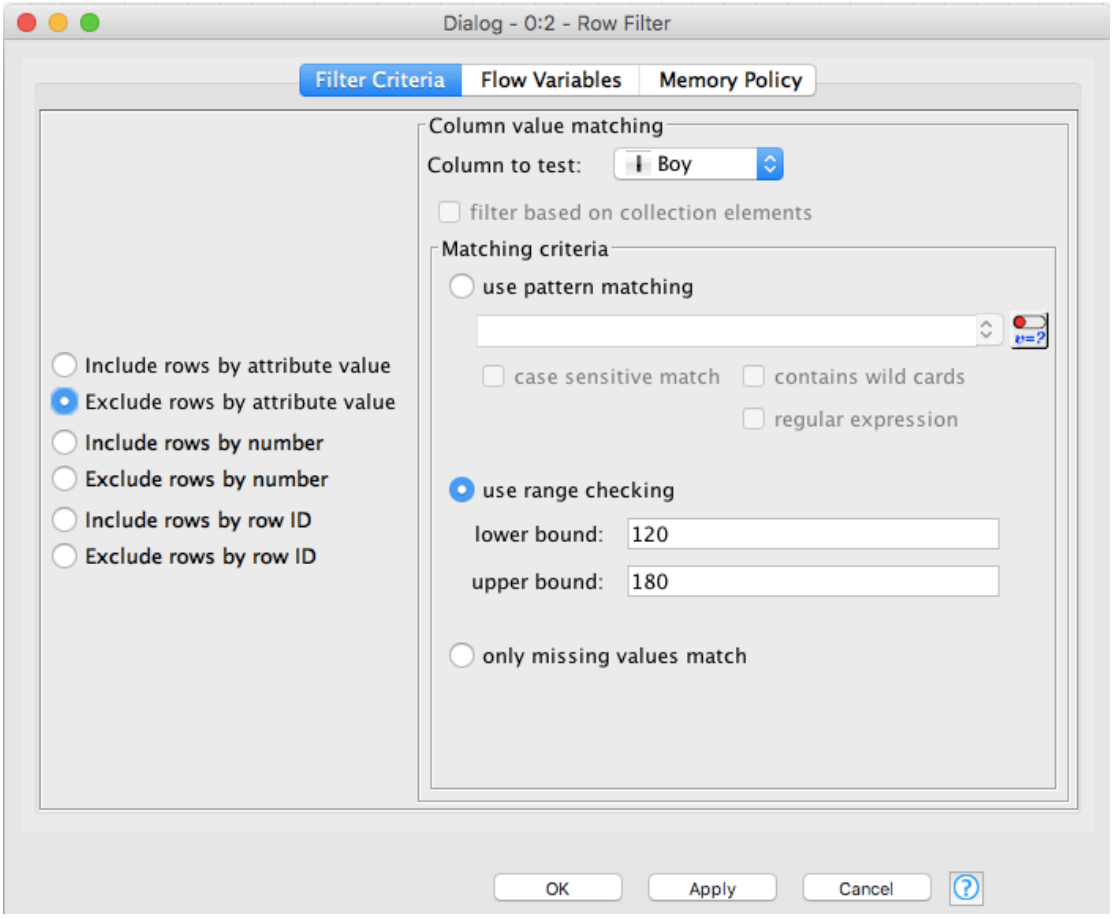
Şekil 6.1.3

Şekil 6.1.3, workflow'a row filter eklenmesini ve bağlantısını göstermektedir. Kutucuğunun üzerinde ünlem işareti ve kırmızı ışık bu operatörde öncelikle configure edilmesinin zorunlu olduğu anlamına gelmektedir.



Şekil 6.1.4

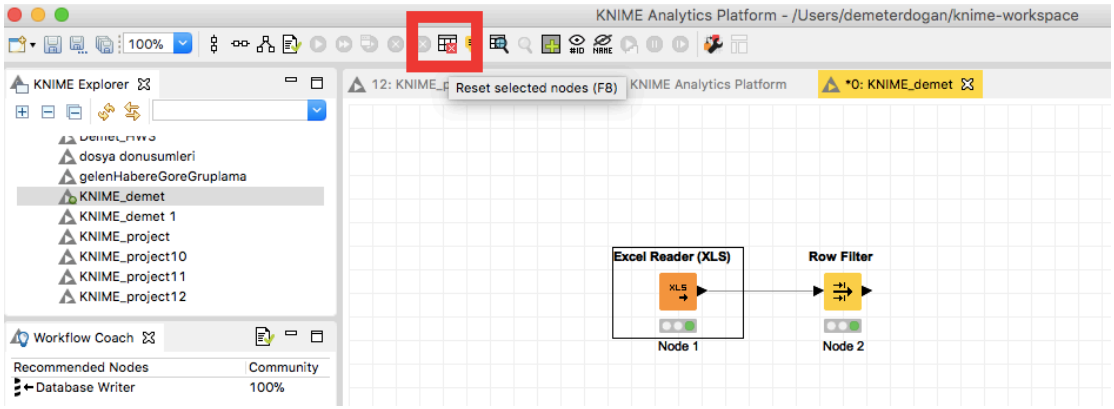
Şekil 6.1.4, row filter operatörünün configure penceresini göstermektedir. Filtrelenecek kolon column to test bölümünden seçilmelidir. Burada 3 kolon bulunduğu için boy, kilo ve cinsiyet kolonlarının isimleri şekilde gösterilmektedir. Use pattern matching bölümünden ise kolonun içinde filtrelenmesi istenilen özellik girilmelidir. Örneğin erkek yazılırsa sadece erkeklerin olduğu satırlar getirilir. Contains wild cards seçilerek e yazılırsa bu harf ile başlayanların olduğu satırlar ekrana getirilir.



Şekil 6.1.5

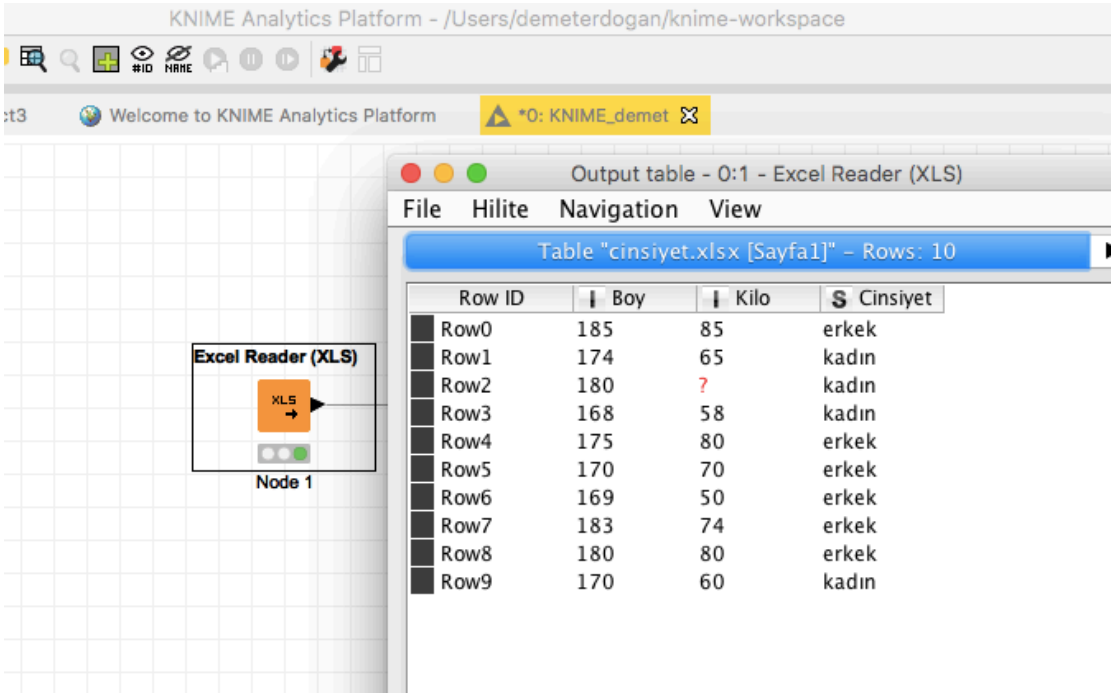
Şekil 6.1.5, yukarıdaki gibi aynı şekilde configure edilmek istenilen kolon seçilerek aralık belirten bir filtrelemeyi göstermektedir. Use range checking bölümünden anlt sınır için lower bound 'a bir değer ve üst limit için upper bound'a bir değer girilmelidir. İnclude rows by attribute seçeneği seçilirse bu verilen değer aralıklarındaki değerlerin olduğu satırlar getirilir. Exclude rows by attribute value seçilirse belirtilen kriter dışarısında kalan veriler getirilir. Only missing values match seçeneği seçilirse eksik verilerin olduğu satırlar getirilir. Fakat hem bu seçenek seçilir hem de exclude rows yazısı ile başlayan bir seçenek sol kısımdan seçilirse eksik verilerin olmadığı satırlar sütünlar getirilir.

Veri setinde eksik veriler ile örnek bir çalışma gösterebilmek adına bir değer silinecek ve sistem reset edilerek veri yüklemesi yenilecektir.



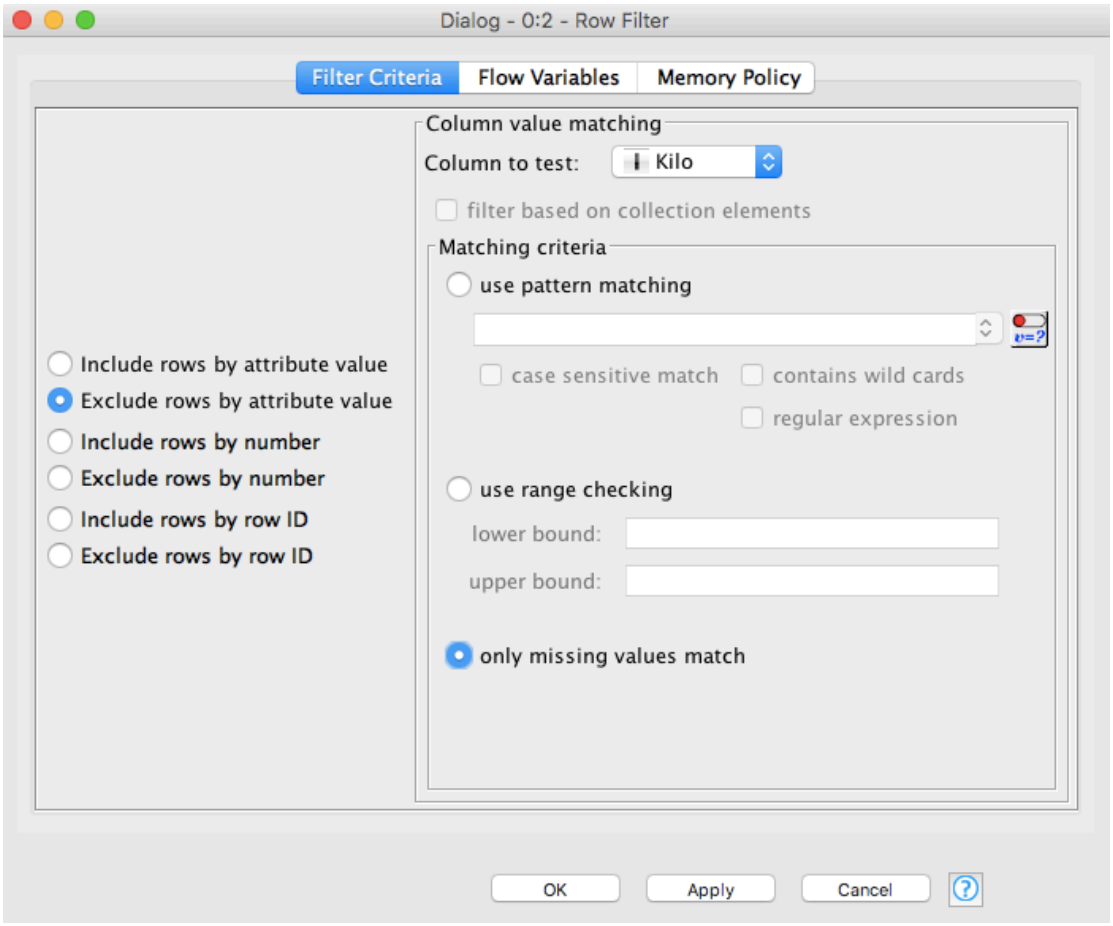
Şekil 6.1.6

Şekil 6.1.6, workflowda yüklenen veri seti üzerinde bir değişiklik yapılırsa şekilde belirtilen alandaki rest butonuna basarak veri setinin yenilenmesini ve güncellemenin sisteme aktarılmasını sağlar.



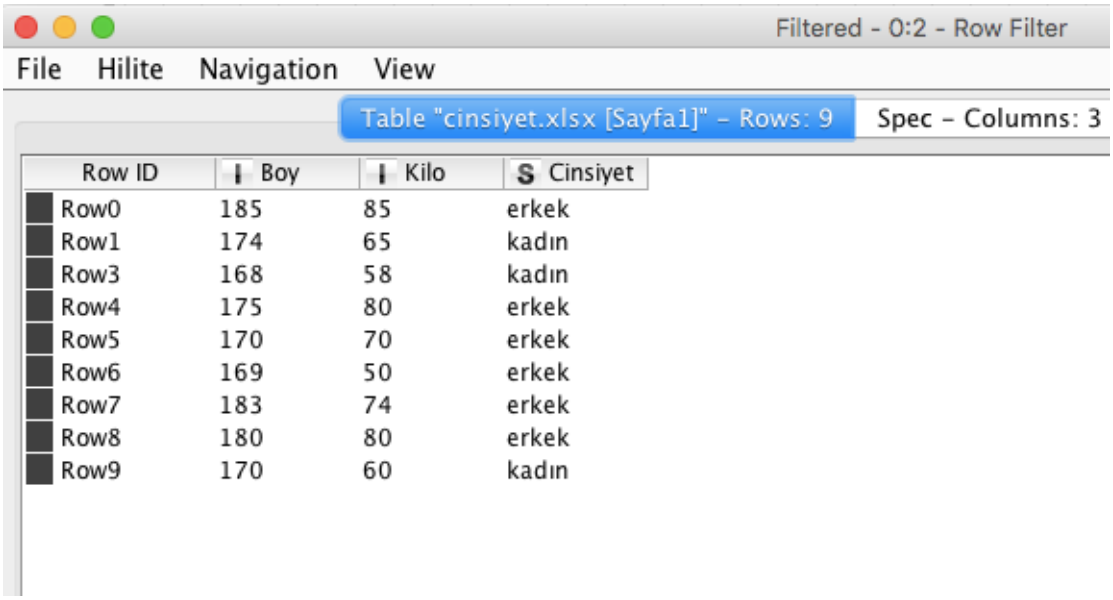
Şekil 6.1.7

Şekil 6.1.7, veri setinde silinmiş yani eksik veri bırakılmış hücreyi ve veri setini göstermektedir. Reset butonuna basarak bu güncelleme knime workflow'una aktarılmıştır.



Şekil 6.1.8

Şekil 6.1.8, eksik verinin olduğu kilo kolonunun configure bölümünde seçilmesini göstermektedir. Ayrıca eksik veriler için only missingg values match seçilmiştir. Bu seçenek seçilirse sistem, sadece eksik verilerin olduğu sıraları baz alır. Fakat aynı zamanda exclude rows by attribute value seçilirse eksik verilerin olduğu sıraların alınmaması anlamına gelmektedir. Bu şekilde kaydedilerek execute edilir.

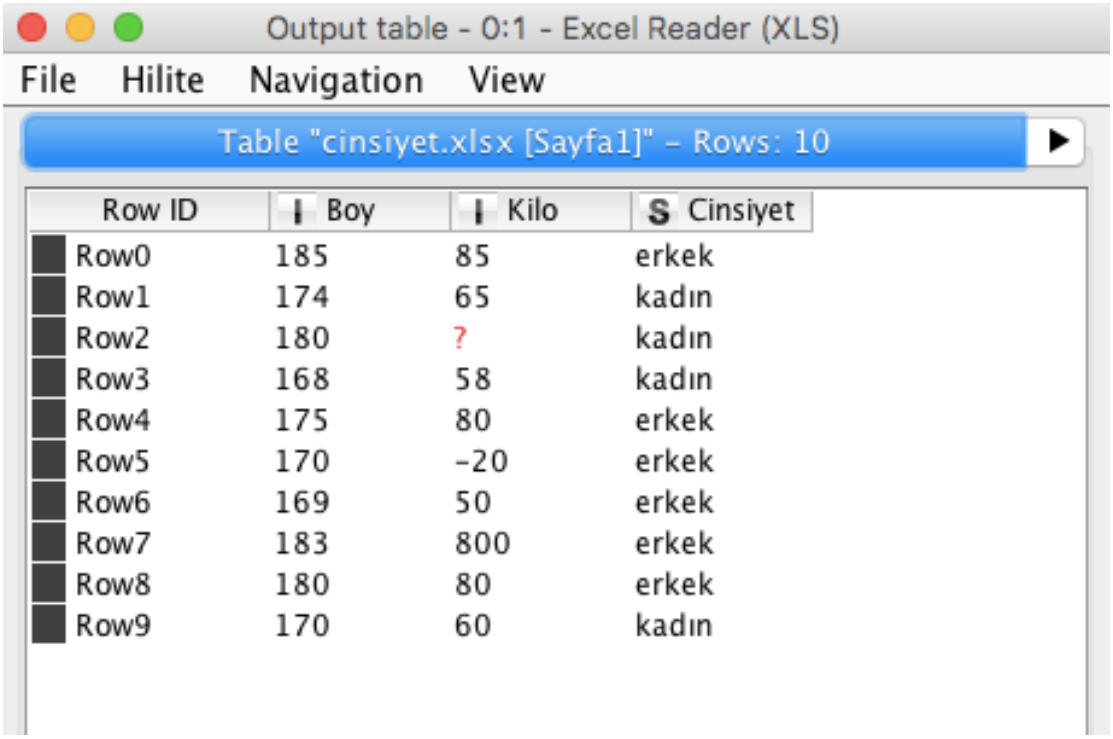


Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek
Row9	170	60	kadın

Şekil 6.1.9

Şekil 6.1.9'da da görüldüğü gibi daha önce eksik veri içeren satır silinmiştir.

Kirli ve gürültülü verilerin temizlenmesi için yine aynı örnek veride öncelikle bazı değerler aşağıdaki gibi değiştirilerek örnek üzerinden açıklanacaktır.

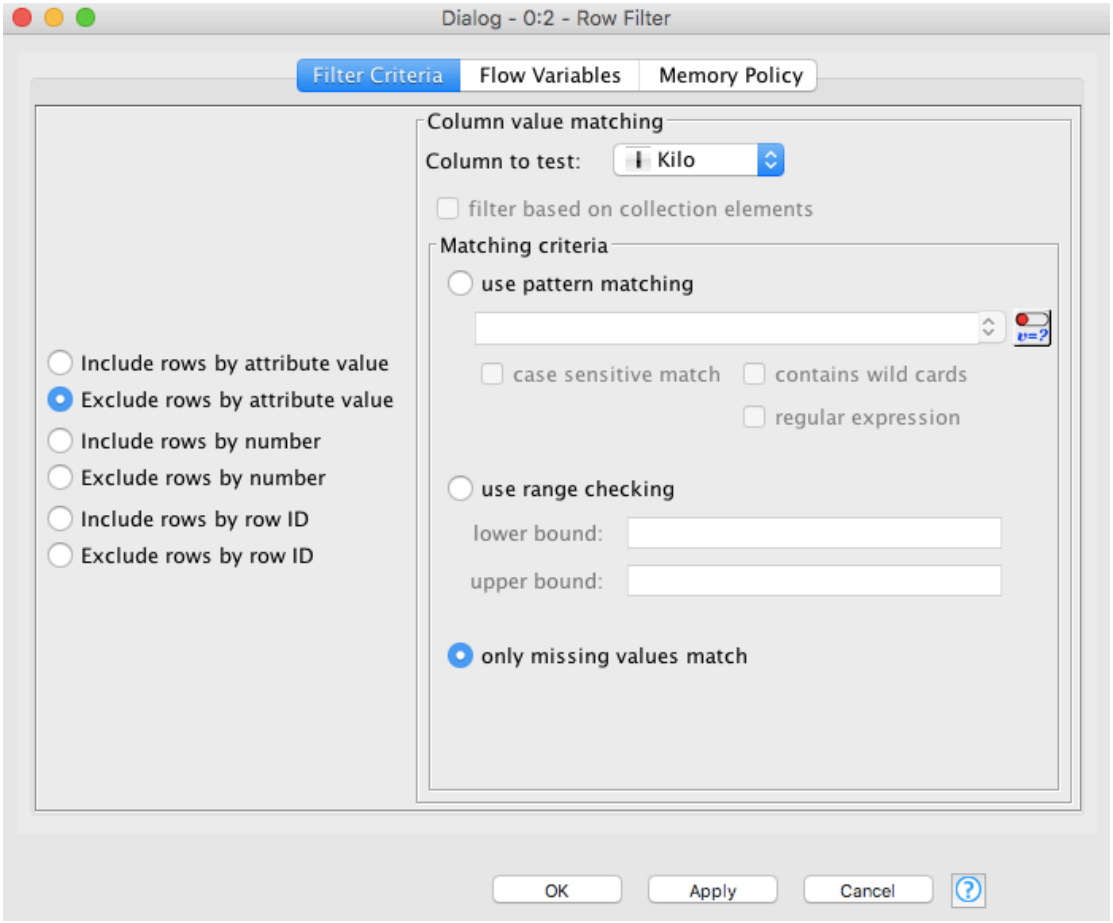


Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	?	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	-20	erkek
Row6	169	50	erkek
Row7	183	800	erkek
Row8	180	80	erkek
Row9	170	60	kadın

Şekil 6.1.10

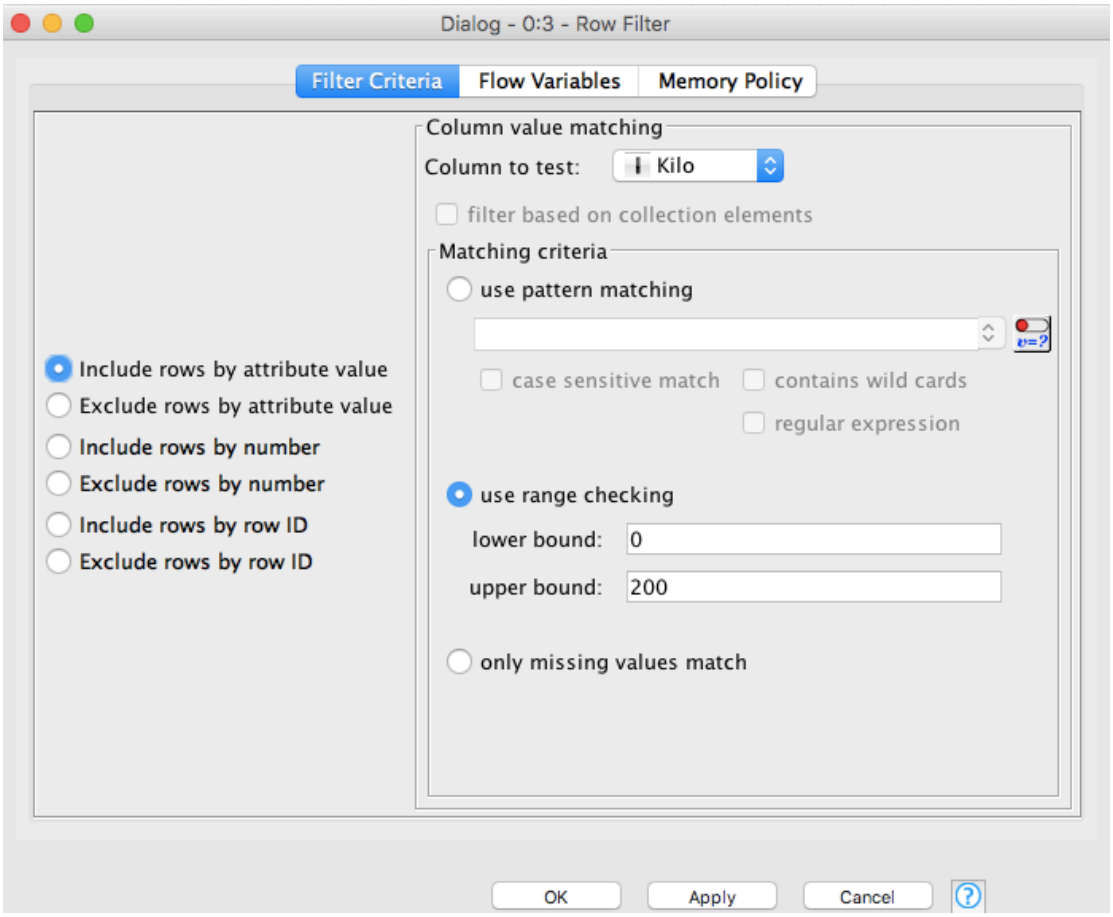
Şekil 6.1.10, veri üzerinde değişiklik yapılan değerleri göstermektedir. Cinsiyet dosyasında değişiklik yapıldıktan sonra excel kaydedilir ve sonrasında workflowda excel reader reset edilir. Sonrasında bu ekranda görülen output table'a ulaşılabilir. Burada görülen yaş kolonundaki -20 değeri imkansızdır. Bir kişinin kilosu 0 ile 200 kg

civarında olabilir. Bu yüzden bu veriye kirli veri denilmektedir. 800 yazan ise abartıdır ve buna da gürültülü veri denilmektedir.



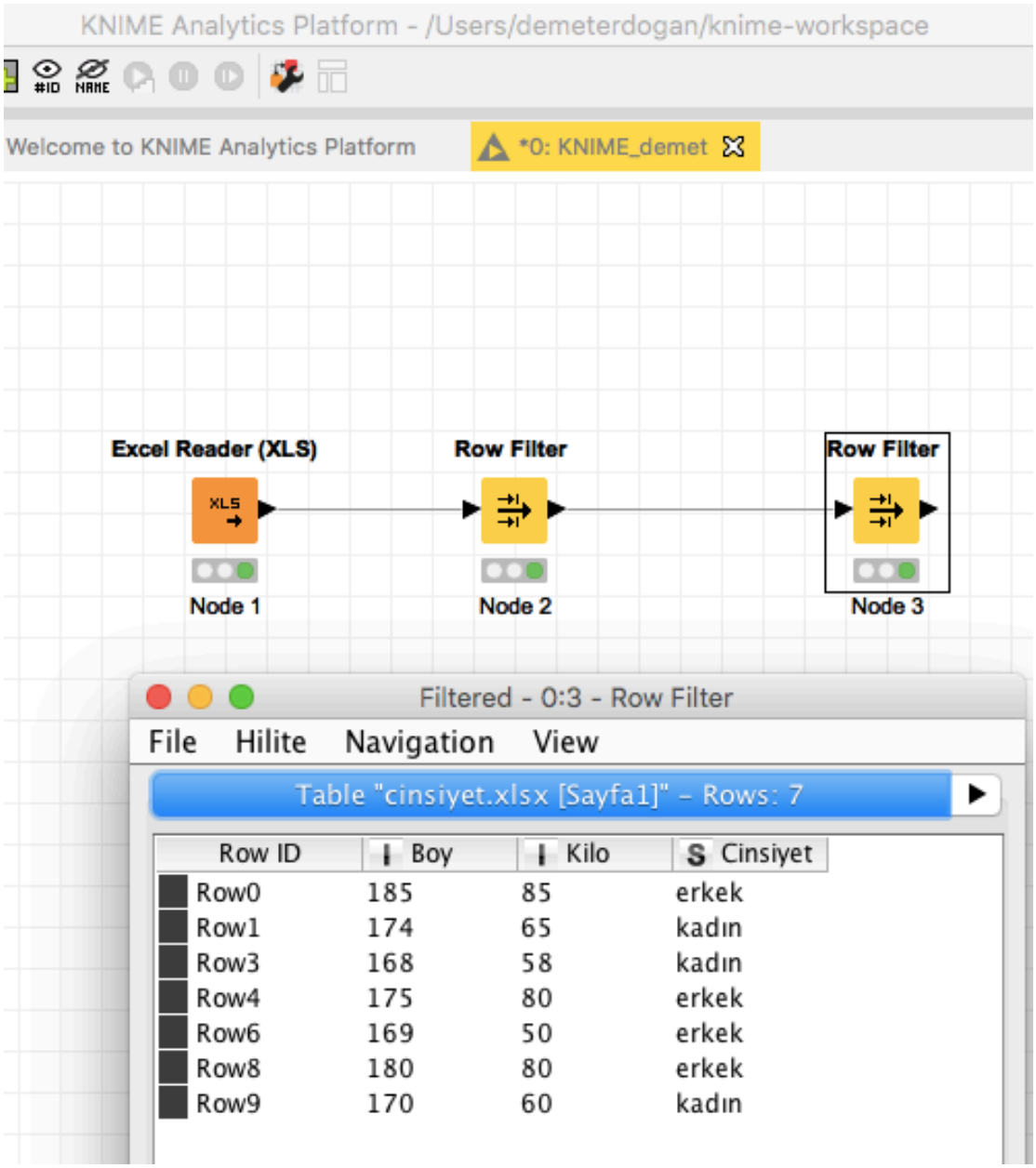
Şekil 6.1.11

Birinci row filter ile öncelikle sistemdeki eksik verilerin olduğu sıralar temizlenir. Şekil 6.1.11'de görüldüğü gibi eksik veriler bu kolonda olduğu bilindiği için kilo kolonu seçilir. Only missing values match seçeneği seçilerek sadece eksik verilerin olduğu satırlar hedef gösterilir. Daha sonrasında asıl istenilen eksik verilerin olduğu sıra alınmamak oldupğu için exclude rows by attribute value seçilir.



Şekil 6.1.12

Şekil 6.1.12, ikinci row filter configure penceresini göstermektedir. Bir row filter'da hem eksik veriler temizlenmesi hem de kirli ve gürültülü verilerin temizlenmesi mümkün değildir. Kilo kolonundaki değerlerin 0-200 arasında olanlarının alınabilmesi için alt ve üst limitler (lower – upper) belirtilmiştir. Bu değerlerin arasındaki değerlerin getirilmesi için include rows by attribute value seçilir ve workflow bu şekilde execute edilir.

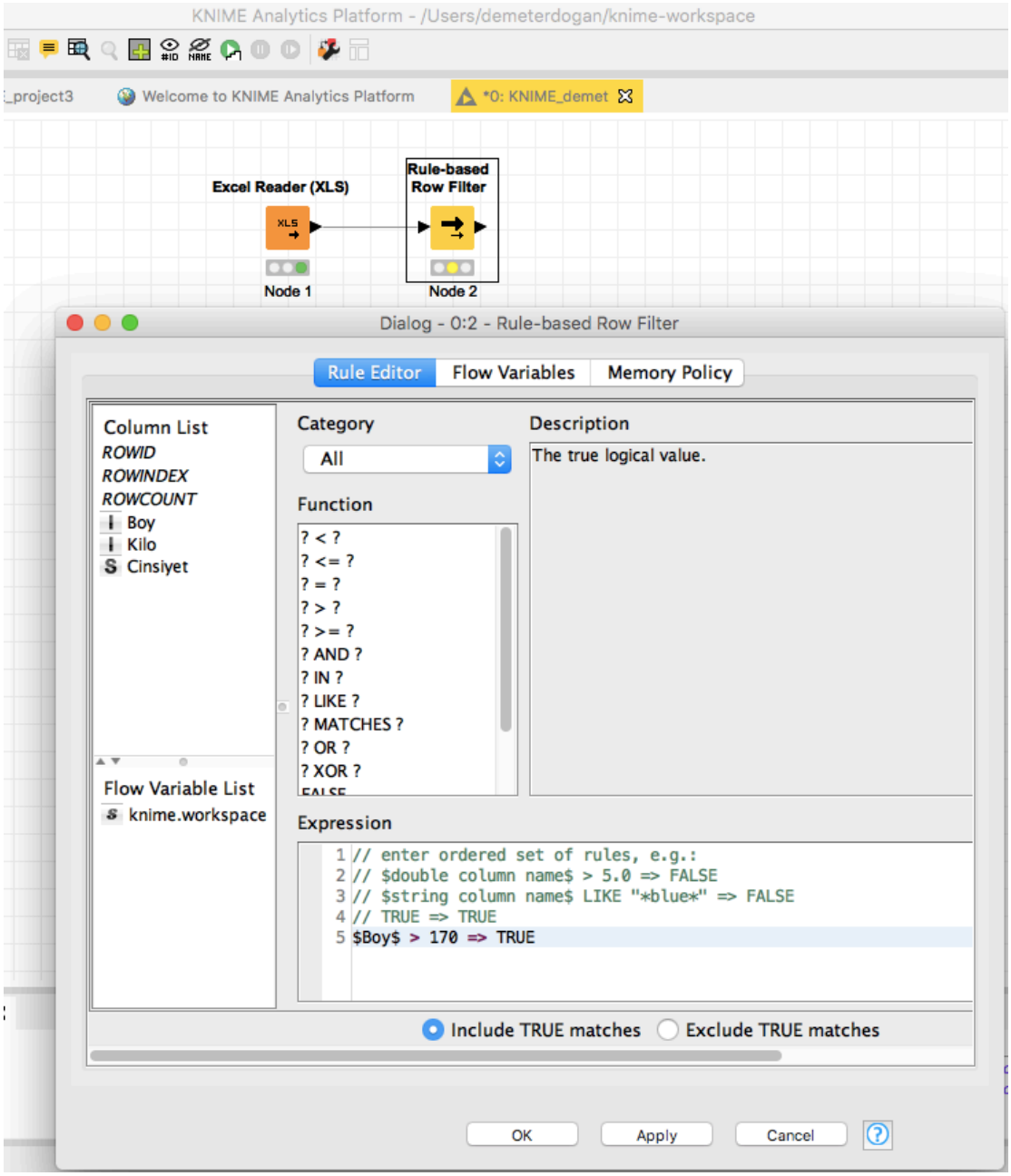


Şekil 6.1.13

Şekil 6.1.13, excel reader ve iki row filter'ın bağlantılarını göstermektedir. Program çalıştırıldıktan sonra elde edilen filtered dosyayı göstermektedir. Görüldüğü gibi eksik verinin olduğu, kirli verinin olduğu ve gürültülü verinin olduğu satırlar silinmiştir. Bu şekilde veri temizleme sayesinde daha sağlıklı öğrenme gerçekleştirilebilir.

6.2 İleri Satır Filtreleme (Rule Based Row Filtering)

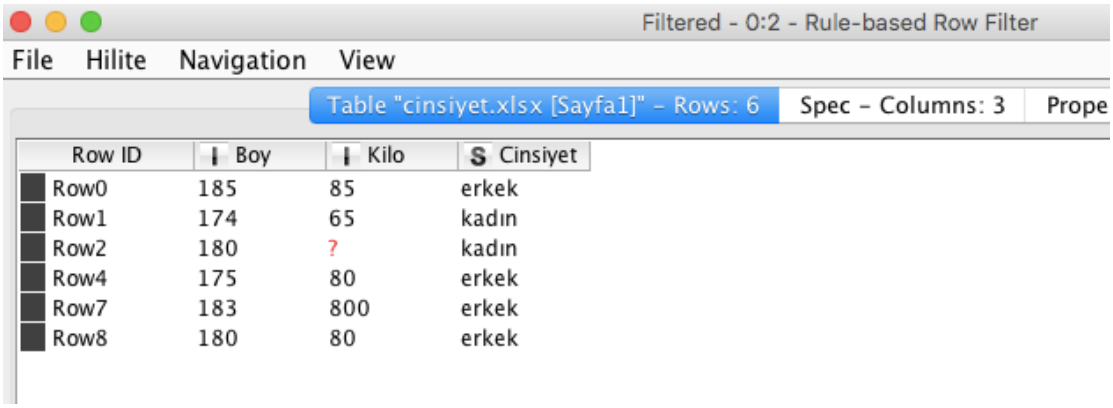
Bu bölümde amaç ileri düzey satır filtrelemeyi göstermektir. Bir önceki bölümde kullanılan yine cinsiyet veri seti ve rule based row filter kullanılacaktır. Rule based row filter 'ın row filter' dan farkı, kural yazılabilmesidir. Row filter'da sadece verilenler arasında include / exclude /only missing values match vb filtremeler yapılırken bu filtrede kural yazılabilmesidir. Rule based row filter'in Türkçesi kural tabanlı satır filtresidir.



Şekil 6.2.1

Şekil 6.2.1, sisteme rule based row filter operatörünün eklenmesini ve excel reader ile bağlantısını göstermektedir. Açık olan pencere ise rule based row filter operatörünün

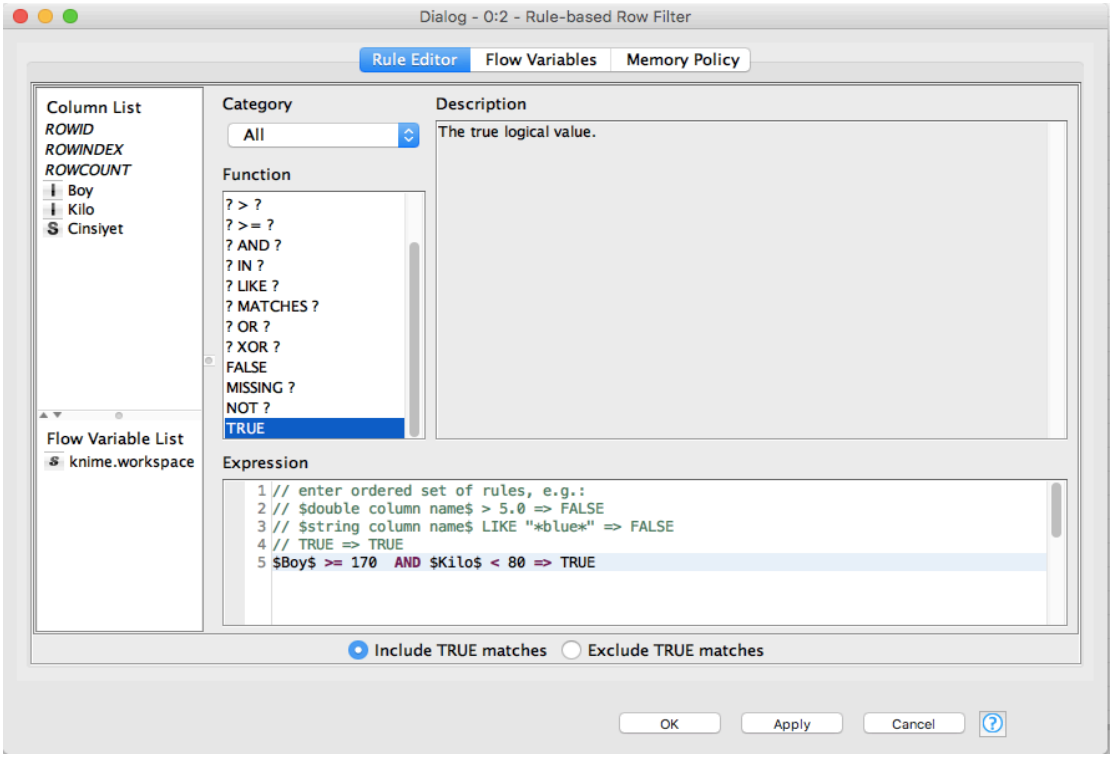
configure penceresini göstermektedir. Operatöre sadece boy, kilo ve cinsiyet kolonları aktığı için sol kısmında bu kolonlarla ilgili isimler çıkmıştır ve bunlarla ilgili function (fonksiyon) yazılabilmektedir. Expression bölümünde örnek fonksiyon kullanımı yazılıdır. Bu örnekte boy >170 => TRUE fonksiyonu sisteme sonradan yazılmıştır. Seçilen klon ismine çift tıklandığı zaman otomatik olarak expression kısmına dolar işaretleri arasında yazılır. Bu örnekte boy ile ilgili fonksiyon yazılacağı için o seçilmiştir. Boyu 170'den büyük olanlar için TRUE yani boyu 170'den büyük olanların seçilmesi anlamına gelmektedir. Alt kısımda da include TRUE matches ise boyu 170 den büyük olanların seçilip getirilmesi, excluede TRUE matches ise boyu 170 den uzun olanların çıkarılarak geri kalanların getirilmesi anlamına gelmektedir.



Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	?	kadın
Row4	175	80	erkek
Row7	183	800	erkek
Row8	180	80	erkek

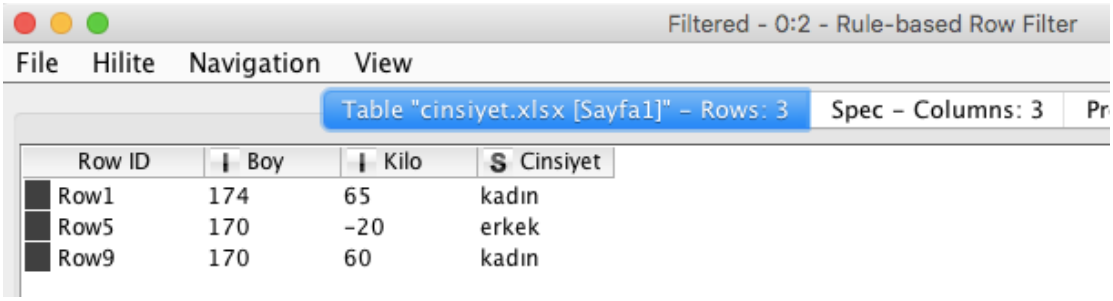
Şekil 6.2.2

Şekil 6.2.2'de de görüldüğü gibi yukarıda yazılan expression'a göre boya göre 170 cm'den büyük olanların tablosunu filtreleyerek onu getirdi.



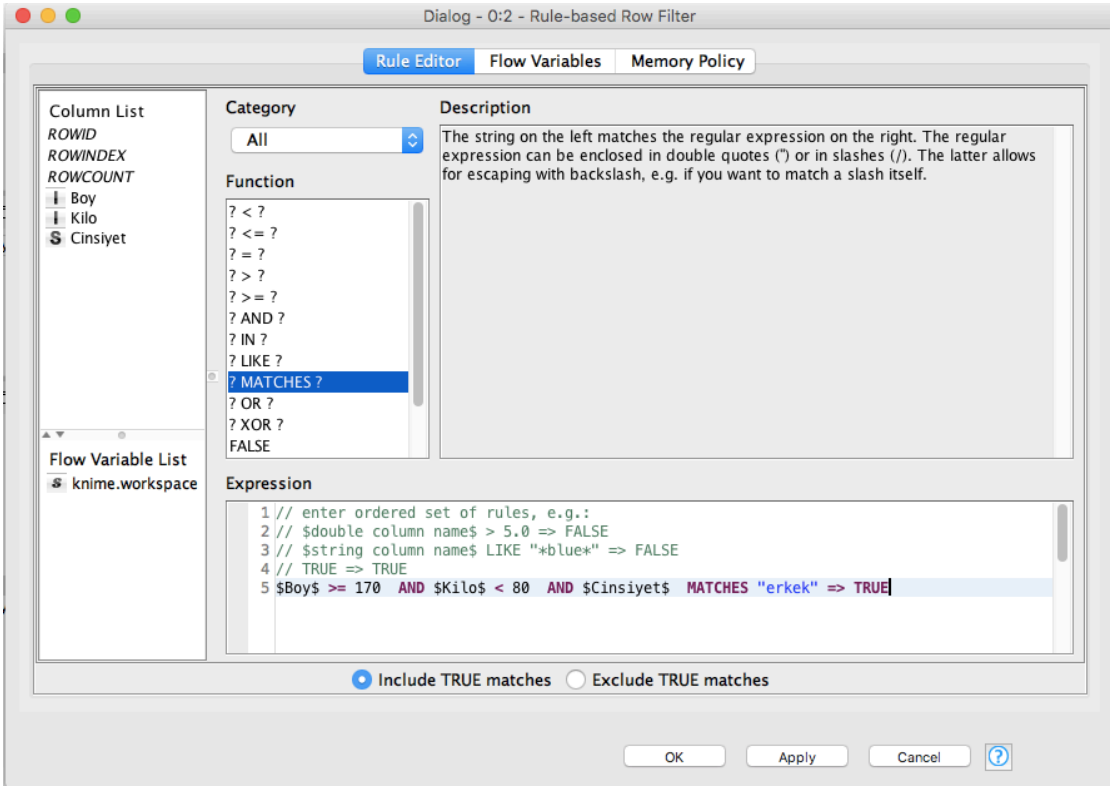
Şekil 6.2.3

Şekil 6.2.3'de görülen configure penceresinde bu sefer de boyu 170 e eşit ve 170'den büyük olanlar ve aynı zamanda kilosu 80'den az olanların getirilmesi için bir komut yazılmıştır. Aynı anda birden çok komut araya "and" gibi bağlaçlarla birleştirilebilir.



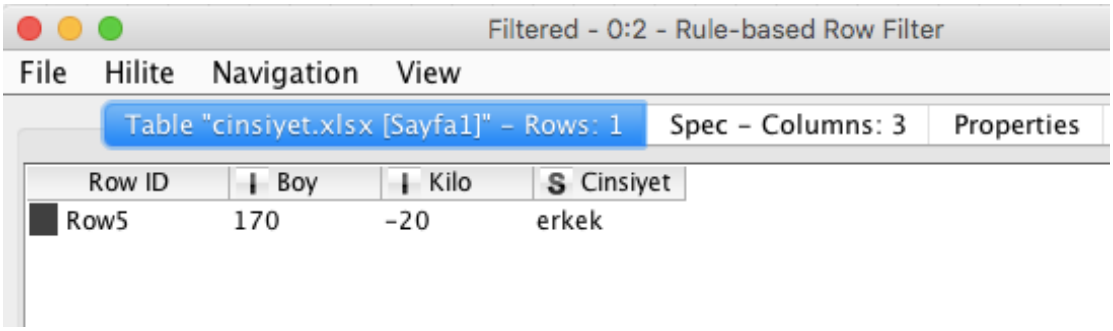
Şekil 6.2.4

Şekil 6.2.4, yukarıdaki komuta göre filtrelenmiş veri setini göstermektedir. Görüldüğü gibi boy 170'e eşit ve daha uzunların olduğu gelirken kilosu da 80'den düşüktür bu kişilerin.



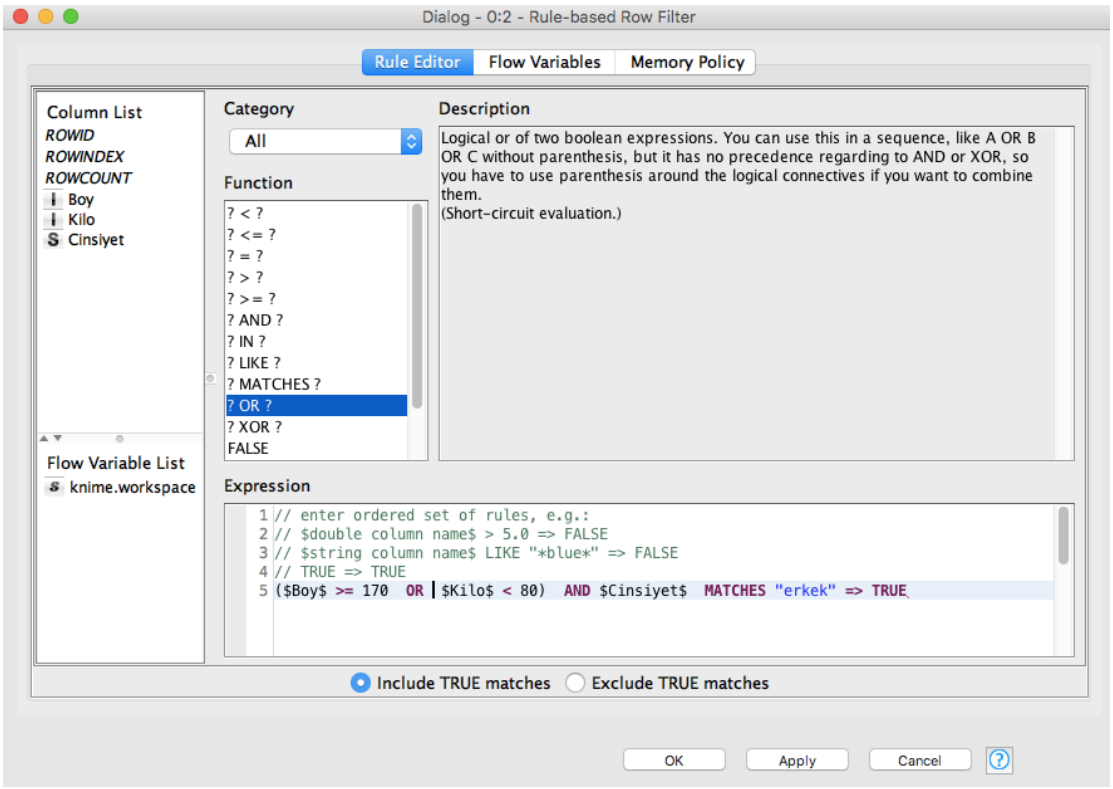
Şekil 6.2.5

Şekil 6.2.5'de de görüldüğü gibi yukarıda yazılmış olan komutlara bir de cinsiyet ile ilgili komutun eklenmesini göstermektedir. Cinsiyeti erkek olanların getirilmesi için matches (eşleştirmek) ile bağlanır. Ayrıca istenilen veri ismi tırnak işareti içerisinde belirtilir. Matches bağlacı string değerler için kullanılır ve eşittir " = " anlamına gelmektedir.



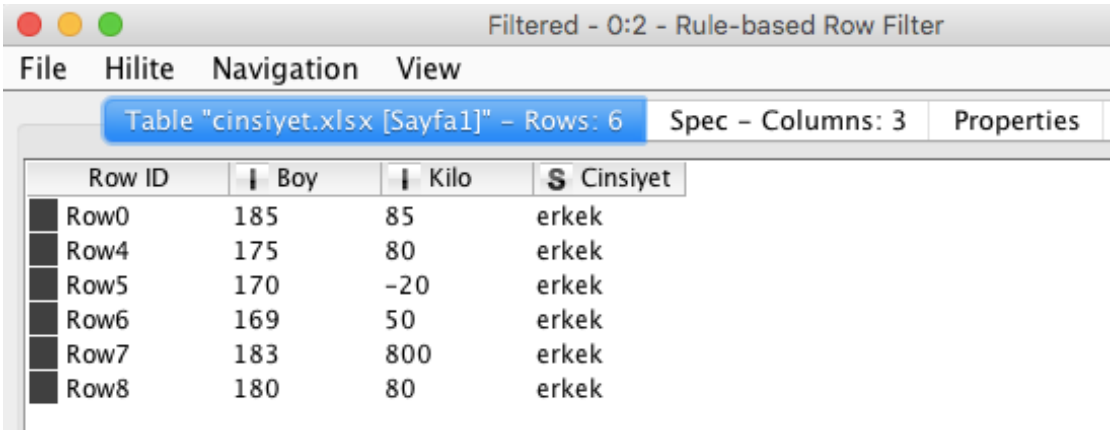
Şekil 6.2.6

Şekil 6.2.6, yukarıda yazılan komuta göre filtrelenmiş veri setini göstermektedir.



Şekil 6.2.7

Şekil 6.2.7, and yerine or bağlacının kullanımını göstermektedir. Burada boyu 170 den büyük veya kilosu 80'den az olan ama sadece erkek olanları filtrelemesi istenmiştir. Parantezin kullanımı içerisindeki ve dışarısındaki komutların önceliği açısından önemlidir.



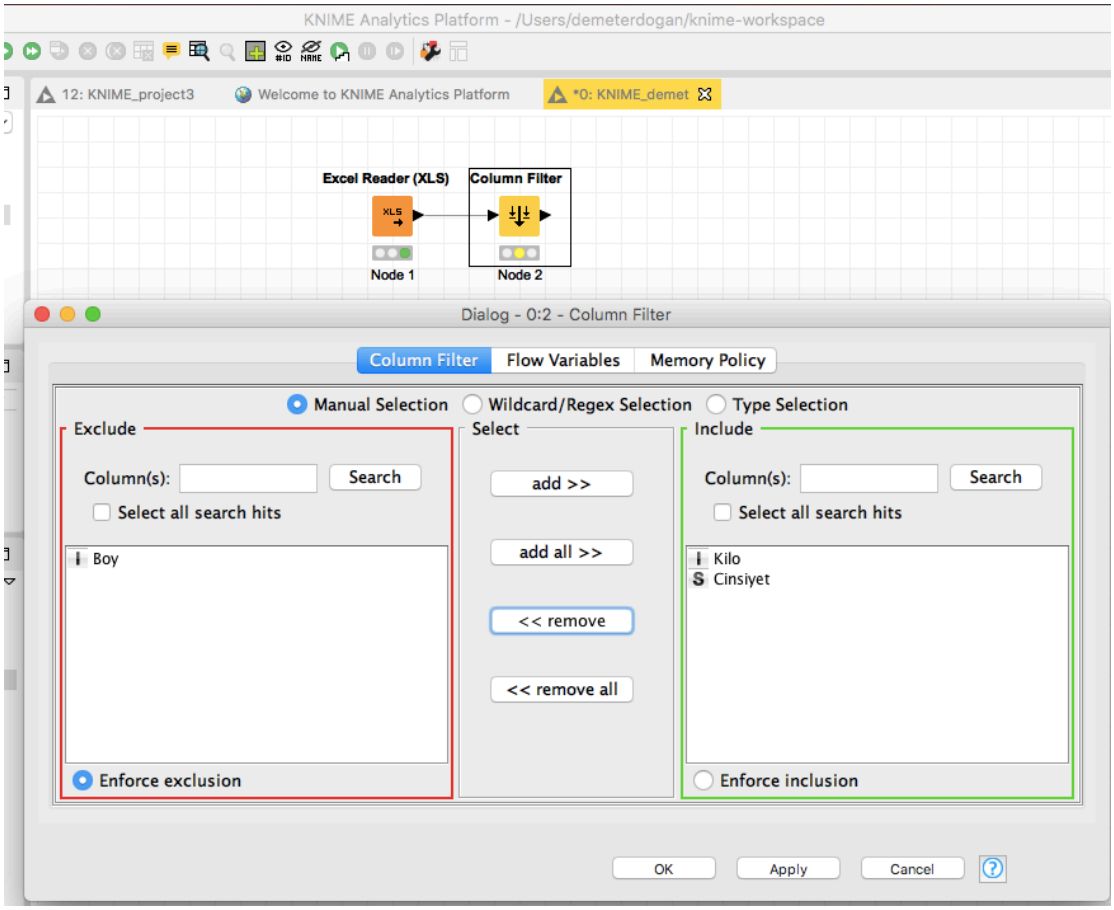
Şekil 6.2.8

Şekil 6.2.8, yukarıdaki komuta göre filtrelenmiş veri setini göstermektedir. 170 uzun olan erkekler ve 85 den az olan erkeklerin hepsini içermektedir.

6.3 Kolon Filtreleme (Column Filtering)

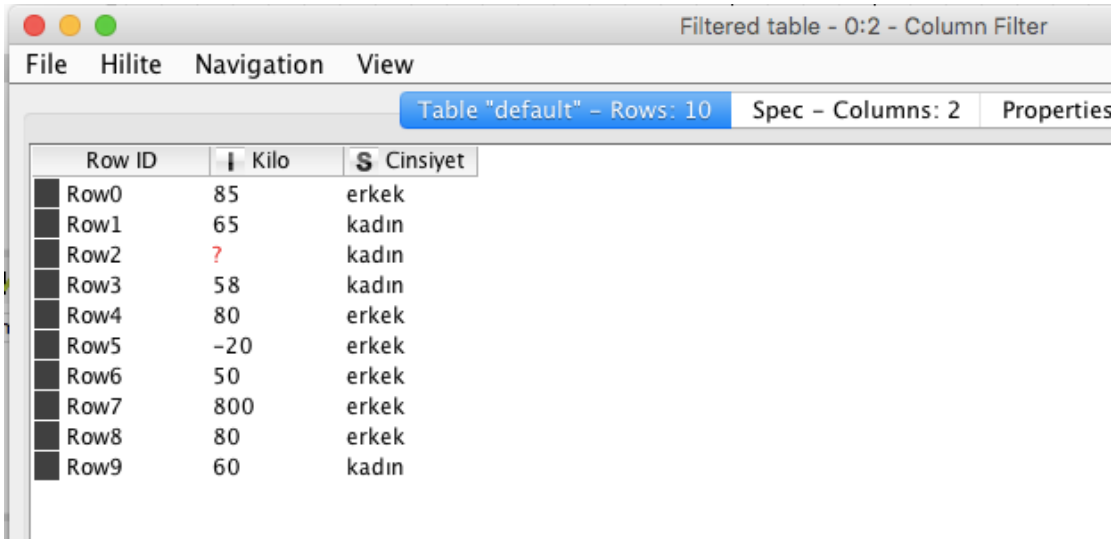
Bu bölümde amaç column filter'ın açıklanmasıdır.

Verinin ön işlemesi (pre-processing) aslında ETL süreci olarak da düşünülebilir. Ön işlemelerden biri de kolon filtrelemedir. Manipulation-Column-Filter-Column filter seçilebilir ya da direkt arama kutucuğuna column filter yazılarak bulunabilir.



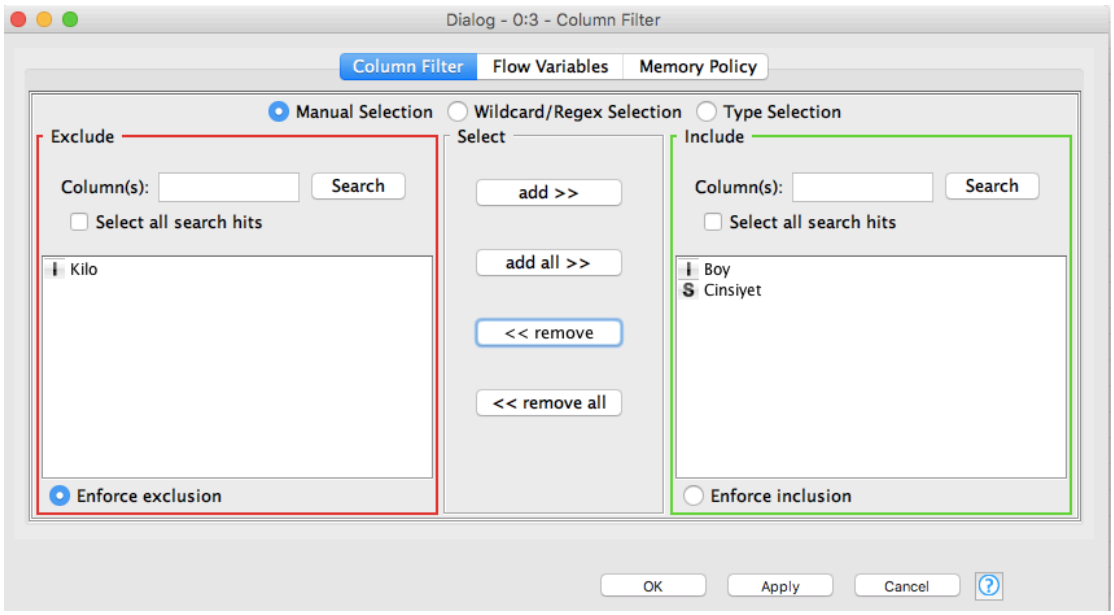
Şekil 6.3.1

Şekil 6.3.1, column filter ve excel reader operatörlerinin bağlantısını ve column filter operatörünün configure penceresini göstermektedir. Amaç row filtering olduğu gibi bir şeyleri filtrelemek fakat bu sefer satır yerine kolon filtrelemek. Veri setinde herhangi bir sebepten dolayı bir veya birden fazla kolon kullanılmayacak ise onları filtrelenebilir. Column filter operatörünün configure penceresinde görüldüğü gibi include kısmında alınmak istenilen kolonlar seçilir. Örneğin sadece kilo ve cinsiyeti kolonları görülsün isteniyorsa boy kolonunu remove edilerek exclude kısmına aktarılmalıdır.



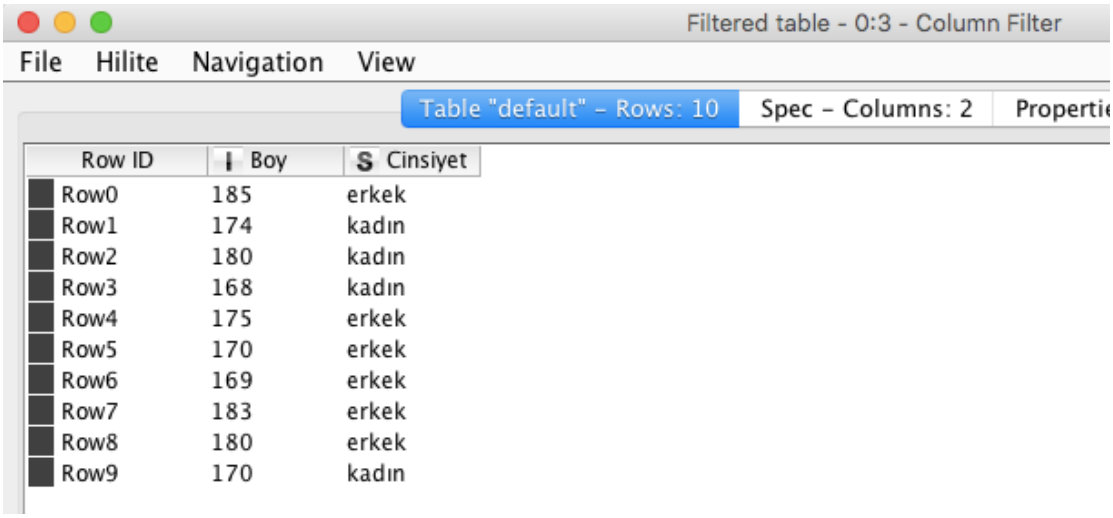
Şekil 6.3.2

Şekil 6.3.2, yukarıda alınan ve elenen kolonların execute edilmiş tablosunu göstermektedir. Görüldüğü gibi kilo ve cinsiyet tabloda yer alırken boy kolonu filtrelenmiştir.



Şekil 6.3.3

Şekil 6.3.3, yukarıdakine benzer bir şekilde column filter'da bu sefer sadece boy ve cinsiyeti seçmeyi göstermektedir.

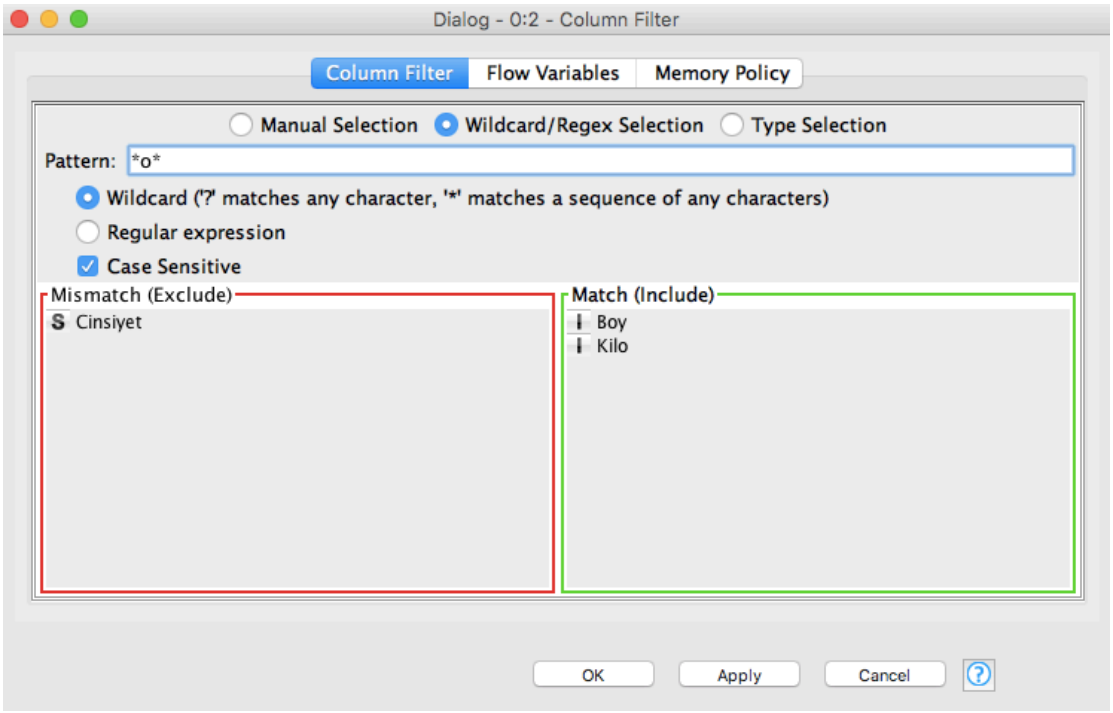


Şekil 6.3.4

Şekil 6.3.4, sadece boy ve cinsiyet kolonlarının alınmış kilo kolonunun elenmiş halini göstermektedir.

Bundan sonra artık makine öğrenmesi yöntemini örneğin desicision tree algoritmasıyla makine öğrenmesi daha önce gösterilmişti. Bu algoritma kullanılarak denenebilir. Hangisinin daha başarılı sonuç çıkardığı test edilebilir.

Kolon filitrelemesinde kolonların her biri aslında birer özellik. Öznitelik olarak da Türkçe'ye çevrilebilir. Kolon filitreleme bazı durumlarda oldukça önemlidir çünkü bazen bir kolon sistemi bozan yapıda olabilir. Örneğin veri setinde kişi isimlerini eklemek makine öğrenmesi olurken bu isimlerin ezberlemesine neden olur. Yani eğer bir şekilde isimden cinsiyeti tahmin ediliyorsa bu da ezbere dayalı bir öğrenmeye girer. Fakat amaç boy ve kilosundan cinsiyetini öğrenmek. Aynı şekilde herkes için unıq olan numaralar sistemin yanlış öğrenmesine sebep olacaktır veya direkt satır numarası, ıd'ler vb. Bilgiler yanlış öğrenmeye sebep olacaktır. Dolayısıyla sistemin doğru öğrenebilmesi için bazen bazı kolonların silinmesi gerekmektedir. Bu durumlarda column filter kullanılabilir. O kolonu komple silme imkanı tanıdığı gibi configure girildiği zaman silinmesini istenilen seçenekler de seçilebilir.



Şekil 6.3.5

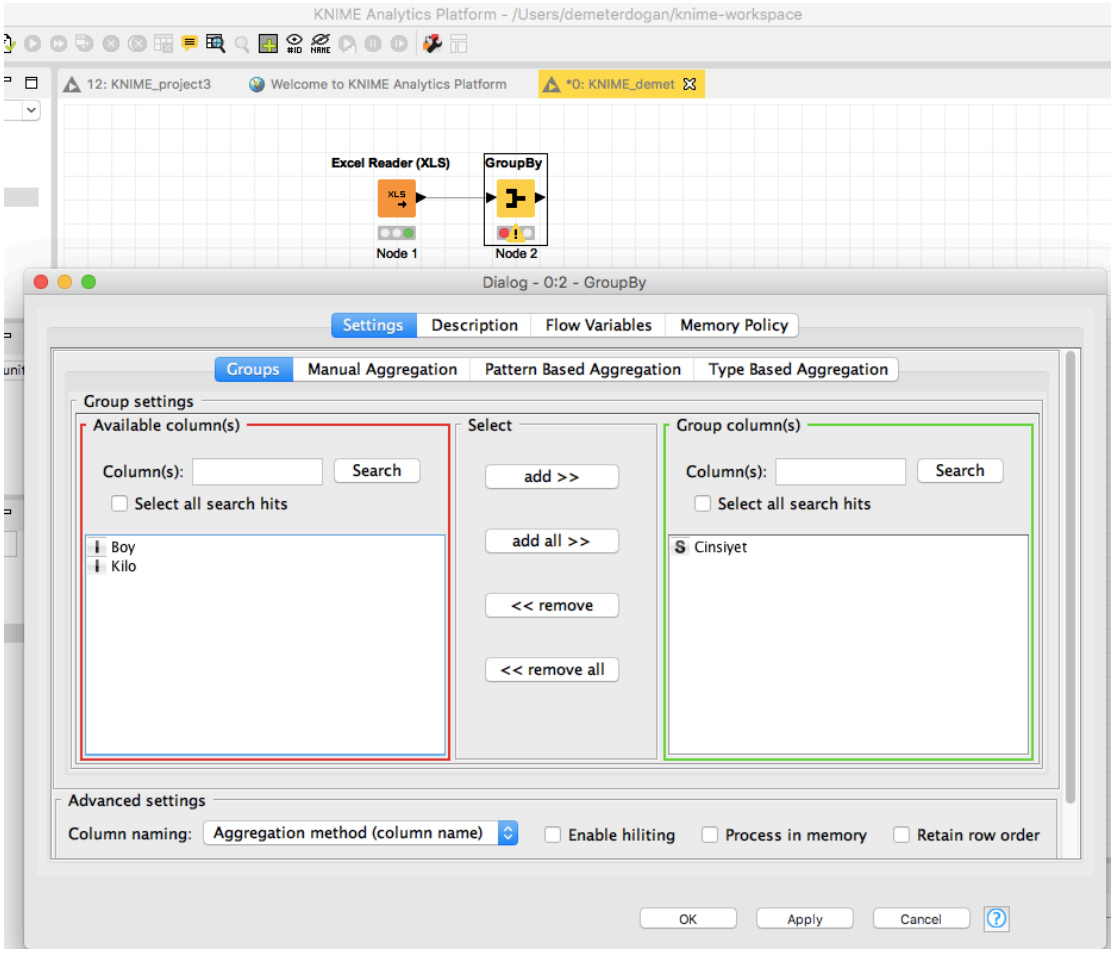
Şekil 6.3.5, column filter'ın wildcard/ Regex selection penceresini göstermektedir.

Wildcard/Regex Selection 'da seçilen kurallara uyan değişkenleri include kısmına ekleyebiliyor. Pattern kısmına b* yazıldığında b ile başlayanları include etmek için direkt match (include) kısmına aktarır. Bu yüzden boy include kısmına eklenir. * işareti sözcüğün geri kalanının önemli olmadığı anlamına gelmektedir. Pattern kısmına b?? yazılması bir karaktere karşılık gelmektedir. Şekilde pattern bölümünün içinde "o" yazarak içinde o harfi geçen değişkenleri include edilmesi istenildi.

Özetle, veri kümesindeki herhangi bir kolon filitrelenmek istenildiğinde veya dışarıda bırakılmak istenildiğinde column filter node düğümü / operatörü kullanılabilir.

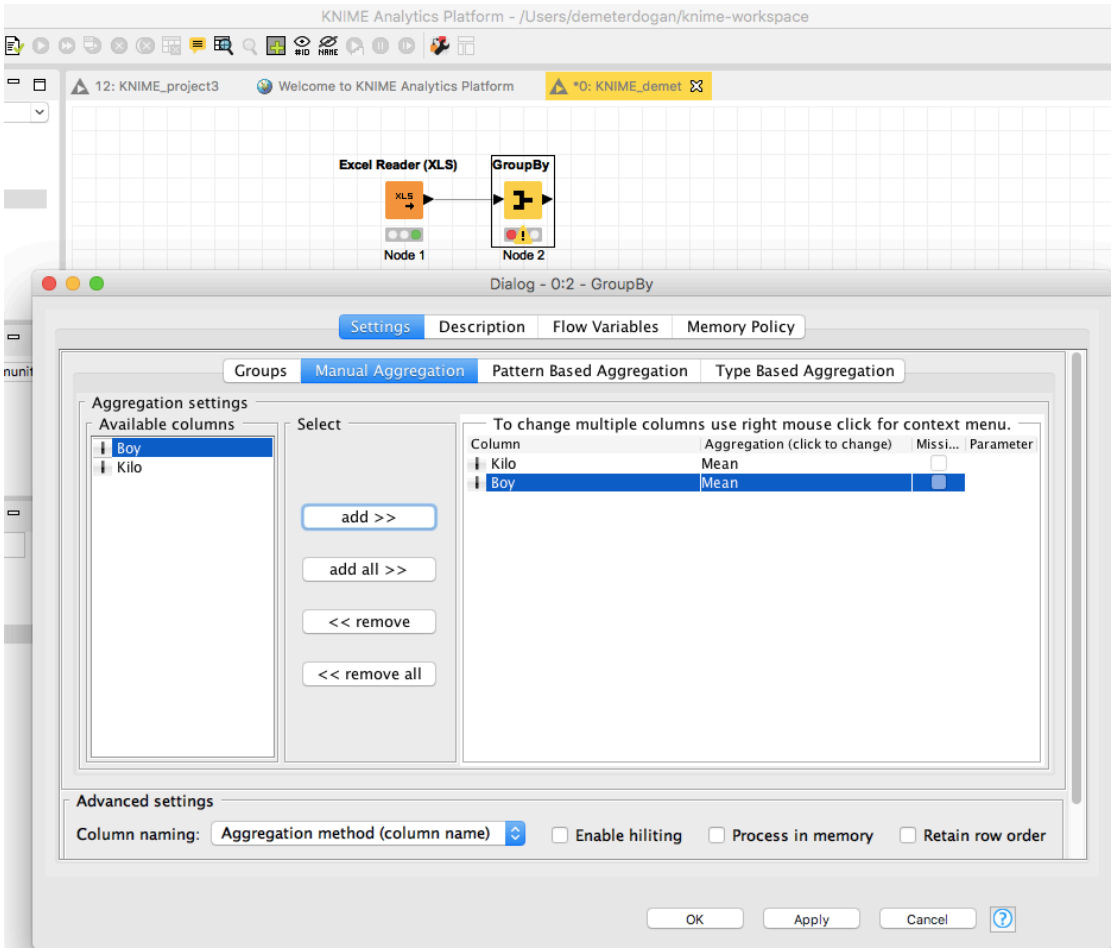
6.4 Gruplama (Group By), Toparlama (Aggregate), Grup Açma (Ungroup) ve Kolon Bölme

Bu bölümde amaç kolon bazlı gruplamanın gösterilmesidir. Bu bölümde de daha önceki bölümlerde kullanılan boy, kilo, cinsiyet verilerinin olduğu örnek dosya kullanılacak ve bu veri setinde nasıl gruplama yapılacağını örnek olarak gösterilecektir.



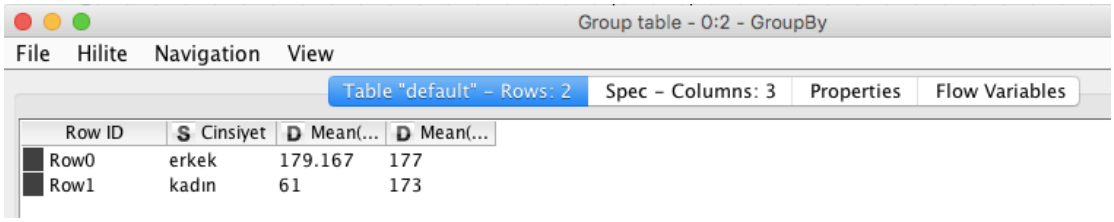
Şekil 6.4.1

Şekil 6.4.1'de görüldüğü gibi excel reader ve GroupBy node'ları/ operatörlerinin bağlantıları yapılır. Sonrasında gruplamak istenilen kolon seçilir. Bu örnekte cinsiyete göre gruplanması gösterileceği için group column(s) bölümüne cinsiyet kolonu aktrılıp diğer kolonlar alınmamıştır. Veri setinde cinsiyet kolonunda kadın ve erkek olduğu için aslında iki satıra indirgenmiş bir sonuç beklenebilir.



Şekil 6.4.2

Şekil 6.4.2'de görüldüğü gibi, manual aggregation penceresi bir önceki pencerede seçilen kolona göre bölünen verilerin burada belirtilecek kurala göre değer döndürmesidir. Örneğin yukarıda cinsiyer seçildiği için kadın ve erkek verileri iki satır olarak gelecektir. Ve burada boy ve kilo seçilip bunların mean değerleri istenildiği için kadın ve erkeklerin ayrı ayrı boy ve kilo ortalamaları çıkması istenildiği anlamına gelmektedir.



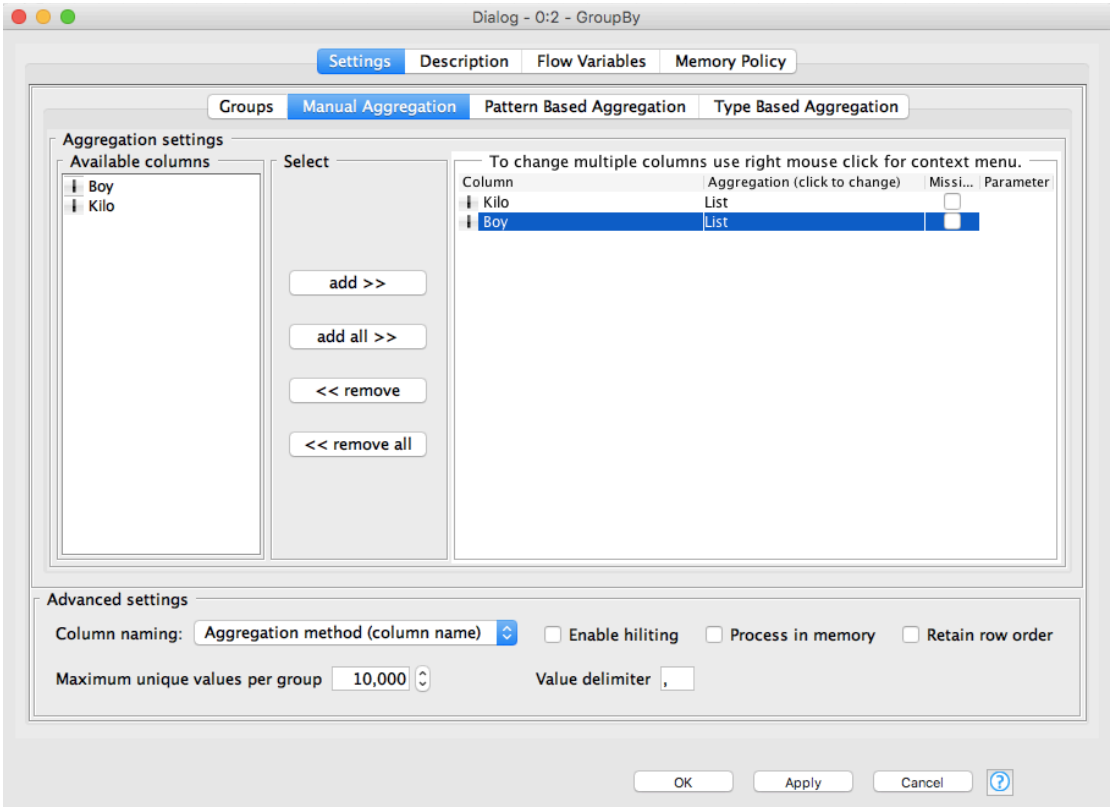
Şekil 6.4.3

Şekil 6.4.3, programı çalıştırıldığında elde edilen group table penceresini göstermektedir. Erkek için ayrı bir satır kadın için ayrı bir satır çıkartılarak bu iki veri kümesinin boylarının ve kilolarının ortalamalarını vermektedir.

Burda iki örnek (kadın/ erkek) olduğu için çok anlamlı gelmeyebilir fakat binlerce farklı etiket olduğunda oldukça işe yarayacak bir işlemdir. Örneğin, her şehirden gelen farklı

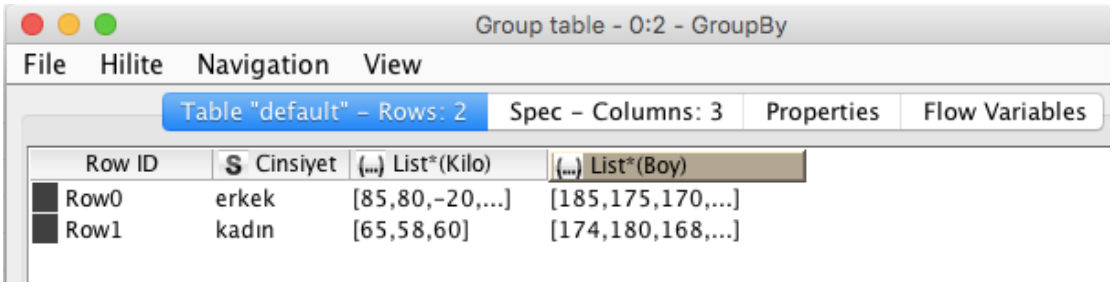
müşteri grupları olduğunda ve müşterilerin şehirlere göre gruplanıp toplam yaptıkları satışları bulunmak istenildiğinde bu konuda işe yarayacaktır. Veya binlerce şubesi olan bir firmanın her şubesinin ortalama satışları bilinmek istenildiğinde, şubelere göre gruplayıp o şubelerin ortalama satışları bu yöntem ile hesaplanabilir.

İsminden de anlaşılabileceği gibi bu group by operatörü verileri gruplamak için kullanılan bir özelliktir. Tek gruplama yöntemi ortalama (mean) almak değil. İstenirse gruplar liste operatörü ile de çalıştırılabilir. Group by düğümünü configure ettikten sonra manual aggregatıon kısmından istenilen operatör seçilebilir.



Şekil 6.4.4

Şekil 6.4.4'de görüldüğü gibi boy ve kilonun listlemesini istenildi.

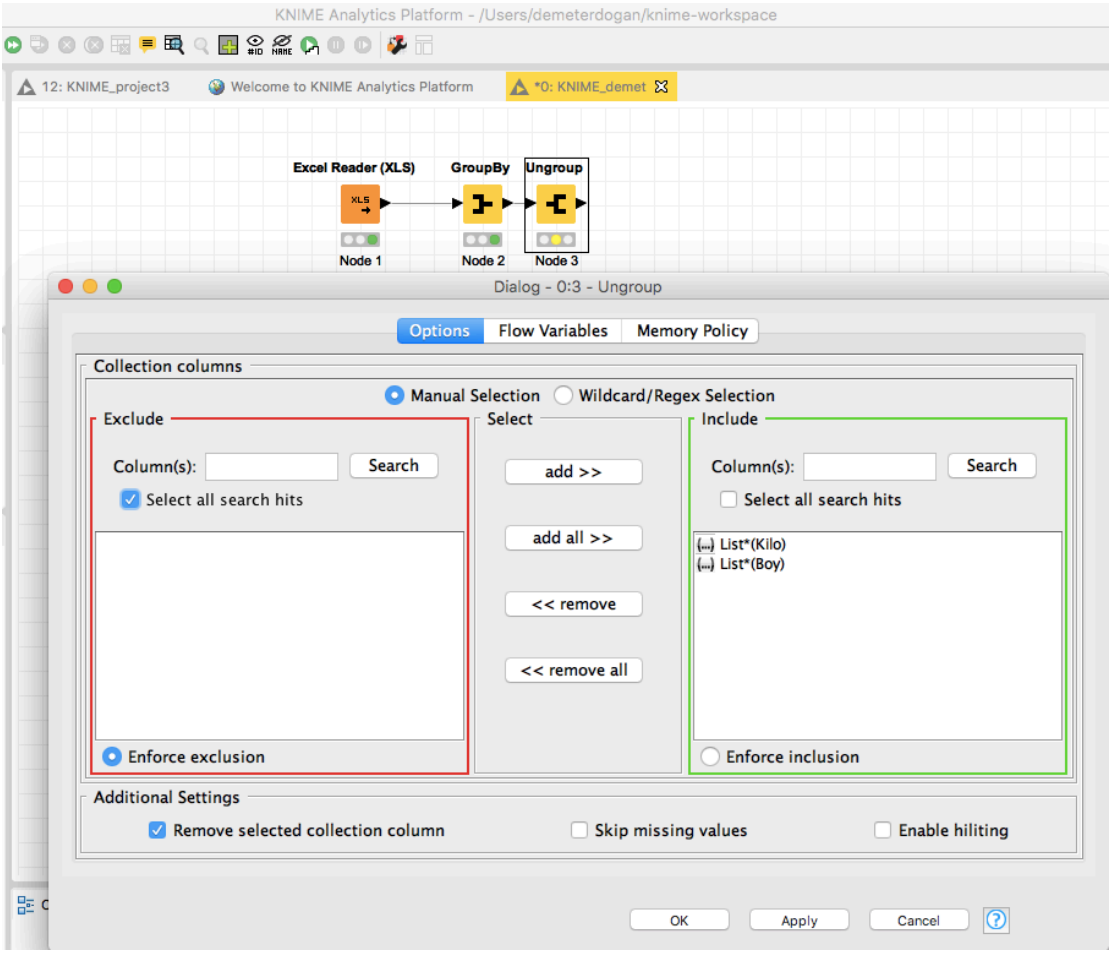


Şekil 6.4.5

Şekil 6.4.5, group by operatörüne sağ tıklayıp group table denildiğinde açılan pencereyi göstermektedir. Aggregatıon aslında toparlamak demektir. Verileri toparlayıp tek bir veriye indirgiyor. Örneğin, erkek için 6 tane satır, kadın için 4 tane satır vardı. Bunları

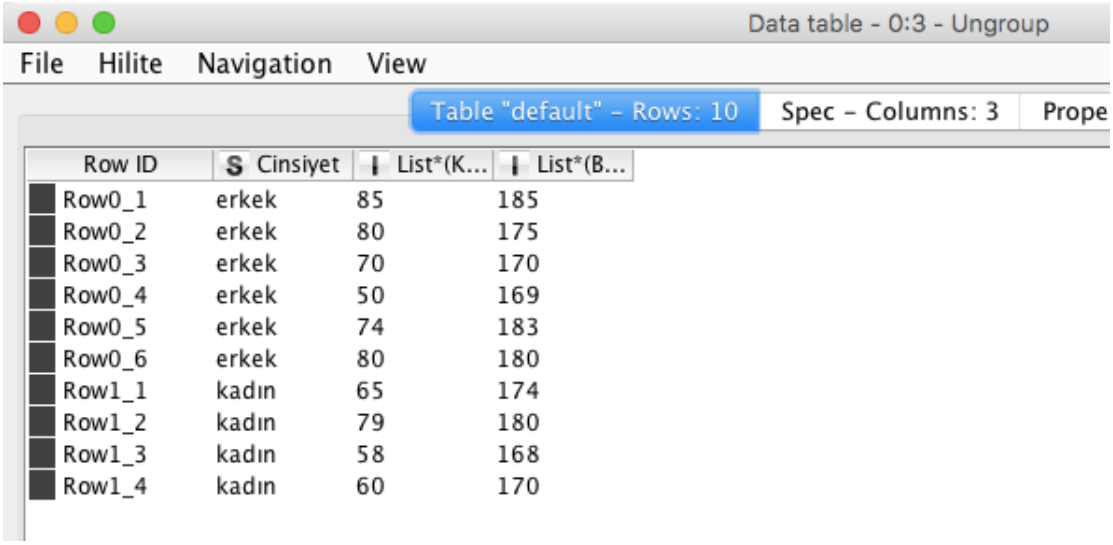
tek bir satıra indirgedi ve buradaki verileri bir listenin içerisine toplayabilir, ortalamasını alabilir, tek bir sayıya toparlayabilir, toplamlarını alabilir, kaç tane olduklarının sayısını alabilirdi. Bütün bunları yapılması manual aggregation penceresinde verilen bilgiyle mümkün kılınır. Bunlar ileride öznitelik çıkartmak için kullanılacaktır.

Group by node/ operatörünün bir tersi ise ungroup operatörüdür. Eğer yukarıdaki şekilde olduğu gibi veri seti liste formatındaysa ya da bir bileşik yapıya sahip ise ungroup ile geri birleştirilmesi mümkündür.



Şekil 6.4.6

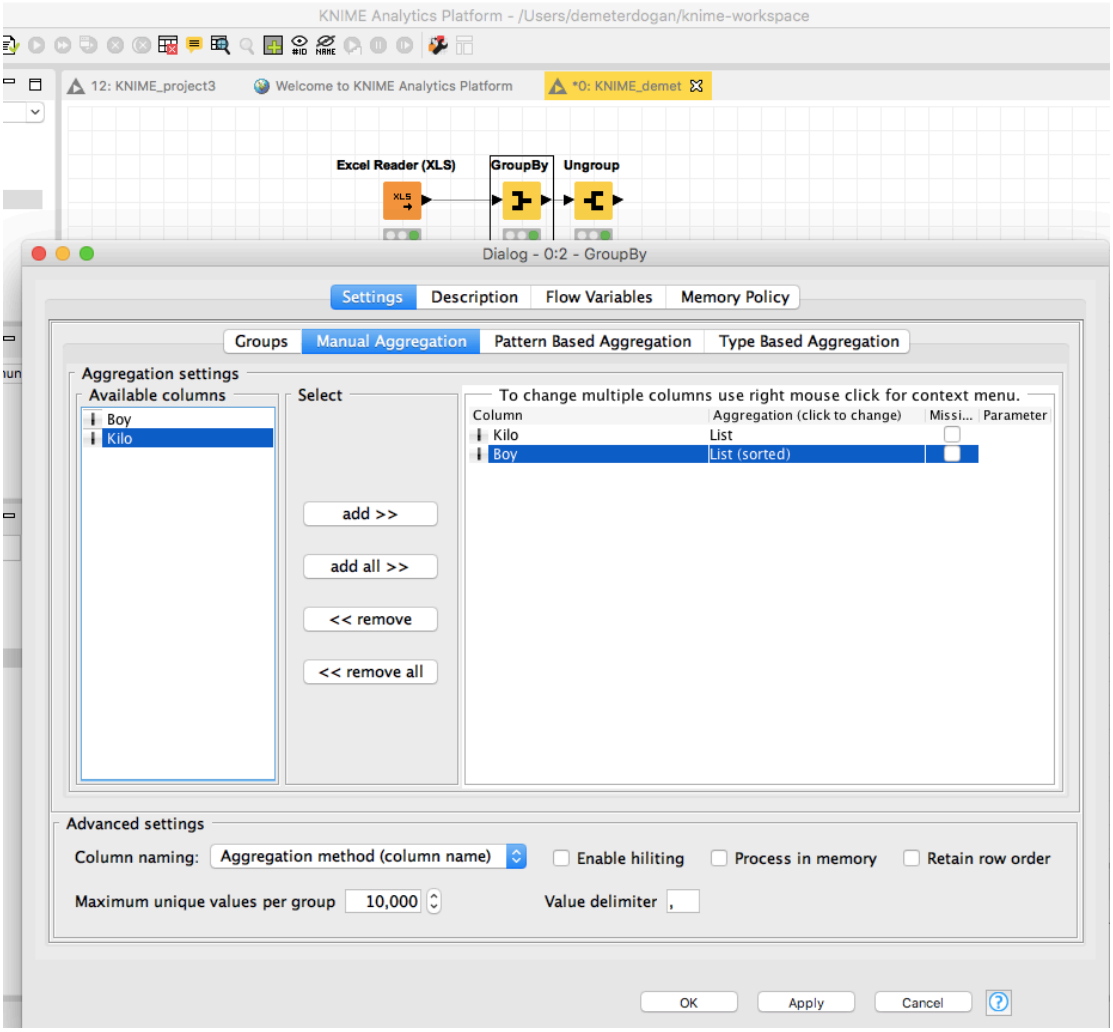
Şekil 6.4.6'da ungroup operatörünün group by operatörü ile bağlantısı ve ungroup node'unun configure penceresi görülmektedir. Ungroup configure edilerek boy listesini ve kilo listesinin ungroup yapılması için ikisini de include penceresine aktarılır ve apply butonuna basılır sonra da program execute edilir.



Row ID	Cinsiyet	List*(K...	List*(B...
Row0_1	erkek	85	185
Row0_2	erkek	80	175
Row0_3	erkek	70	170
Row0_4	erkek	50	169
Row0_5	erkek	74	183
Row0_6	erkek	80	180
Row1_1	kadın	65	174
Row1_2	kadın	79	180
Row1_3	kadın	58	168
Row1_4	kadın	60	170

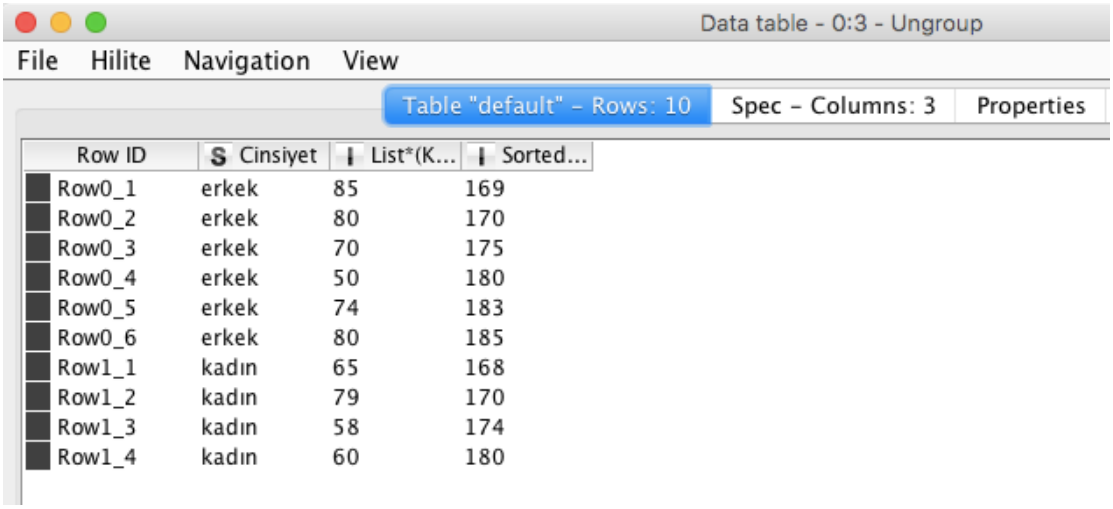
Şekil 6.4.7

Şekil 6.4.7 de görüldüğü gibi veri setindeki liste formatı yeniden eski formatına döner.



Şekil 6.4.8

Şekil 6.4.8'de görüldüğü gibi, group by node'u configure edilerek boy kolonu listesi sorted liste yapıldı. Diğerini yani kilo listesi aynı liste formatında bırakıldı ve kaydedildikten sonra program execute edildi.



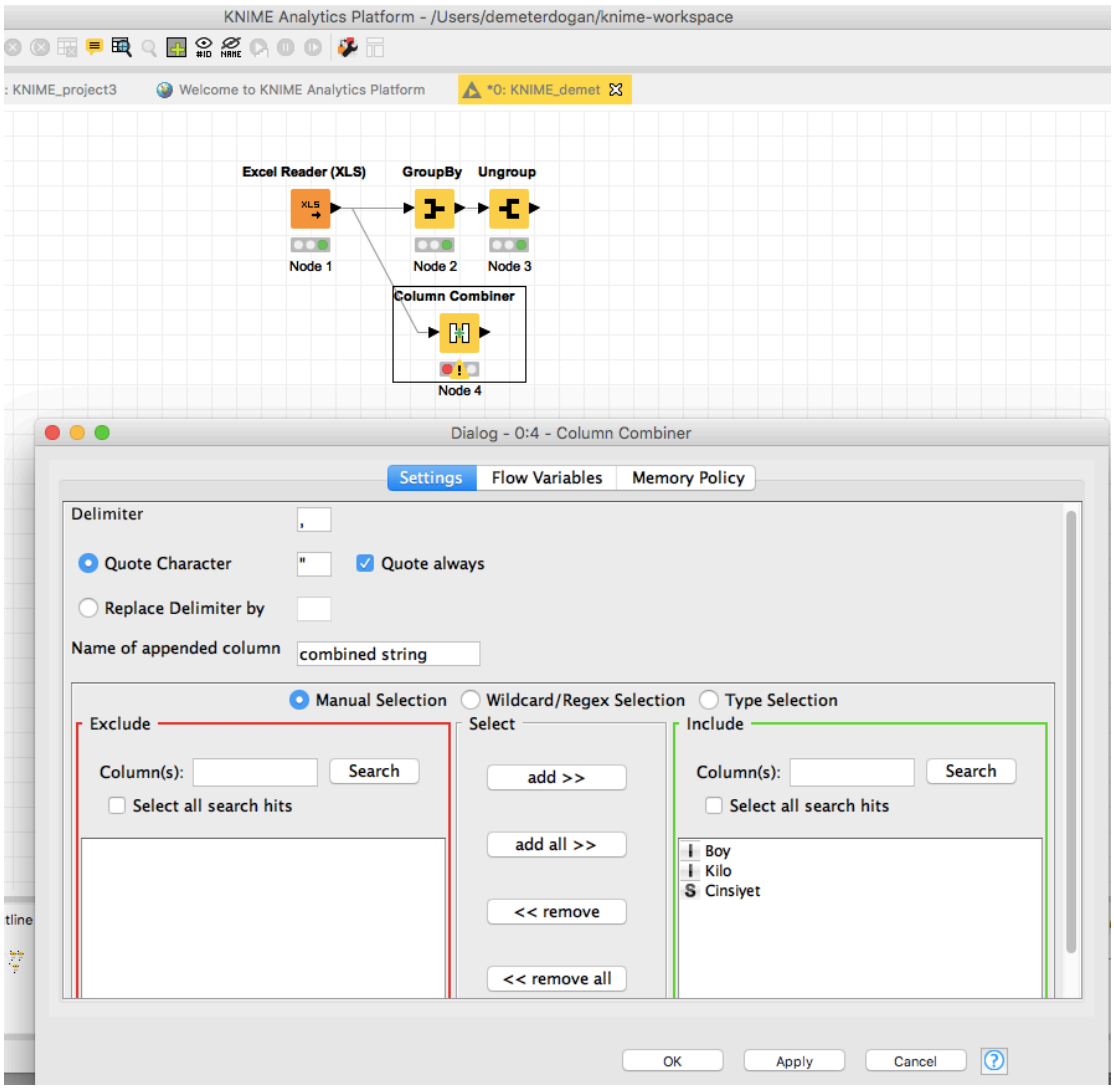
Row ID	S Cinsiyet	List*(K...	Sorted...
Row0_1	erkek	85	169
Row0_2	erkek	80	170
Row0_3	erkek	70	175
Row0_4	erkek	50	180
Row0_5	erkek	74	183
Row0_6	erkek	80	185
Row1_1	kadın	65	168
Row1_2	kadın	79	170
Row1_3	kadın	58	174
Row1_4	kadın	60	180

Şekil 6.4.9

Şekil 6.4.9, ungroup operatörüne sağ tıkladıktan sonra data table denildiği zaman açılan data table' ı göstermektedir. Şekilde de görüldüğü gibi boylar sıralı fakat kilolar sırasızdır.

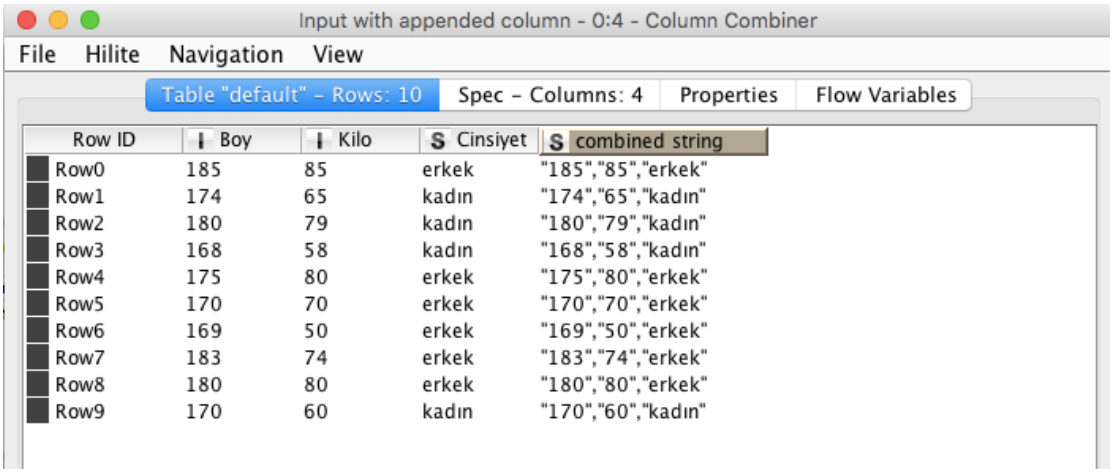
Özetle, veri setinde etiket varsa veriler etikete göre gruplanması, diğer kolonların birleştirilmesi mümkündür. Burada satır bazlı (row based) gruplama veya ungrouptan örnek gösterildi. Benzer şekilde kolon (column) bazlı işlemlerde yapılabilir.

Aynı şekilde group by operatöründeki gibi, sırasıyla node Repository-Manipulation-Column-Split&Combine- Column Combiner node'una ulaşarak veri setine bağlanır. Bu şekilde kolonlar birleştirilecektir.



Şekil 6.4.10

Şekil 6.4.10, sisteme column combiner operatörünün eklenmesini, bağlantılarını ve configure penceresini göstermektedir. Burada birleştirilmesi istenilen kolonlar seçilir yani include penceresine aktarılır. Örnek olması açısından boy, kilo ve cinisyet yani üç kolon da dahil edilmiştir. Delimiter kısmında aralarına virgül koyacağı, quote character ise string değere olduğu gibi tüm karakterleri tırnak(“ ”) içine alarak birleştirecek.

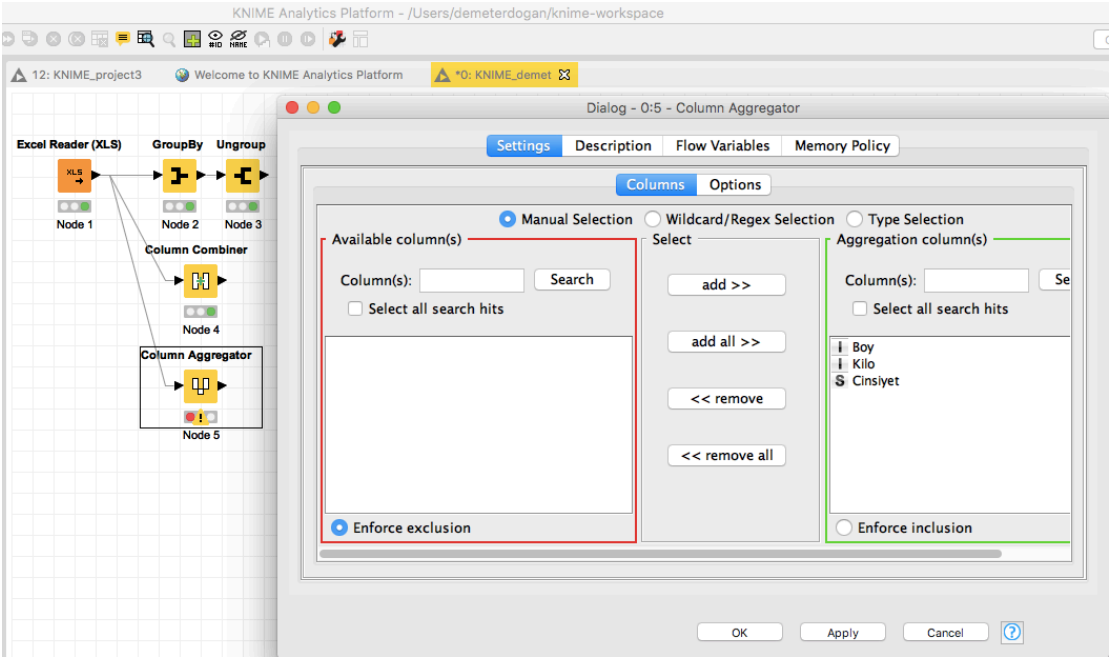


Row ID	Boy	Kilo	Cinsiyet	combined string
Row0	185	85	erkek	"185","85","erkek"
Row1	174	65	kadın	"174","65","kadın"
Row2	180	79	kadın	"180","79","kadın"
Row3	168	58	kadın	"168","58","kadın"
Row4	175	80	erkek	"175","80","erkek"
Row5	170	70	erkek	"170","70","erkek"
Row6	169	50	erkek	"169","50","erkek"
Row7	183	74	erkek	"183","74","erkek"
Row8	180	80	erkek	"180","80","erkek"
Row9	170	60	kadın	"170","60","kadın"

Şekil 6.4.11

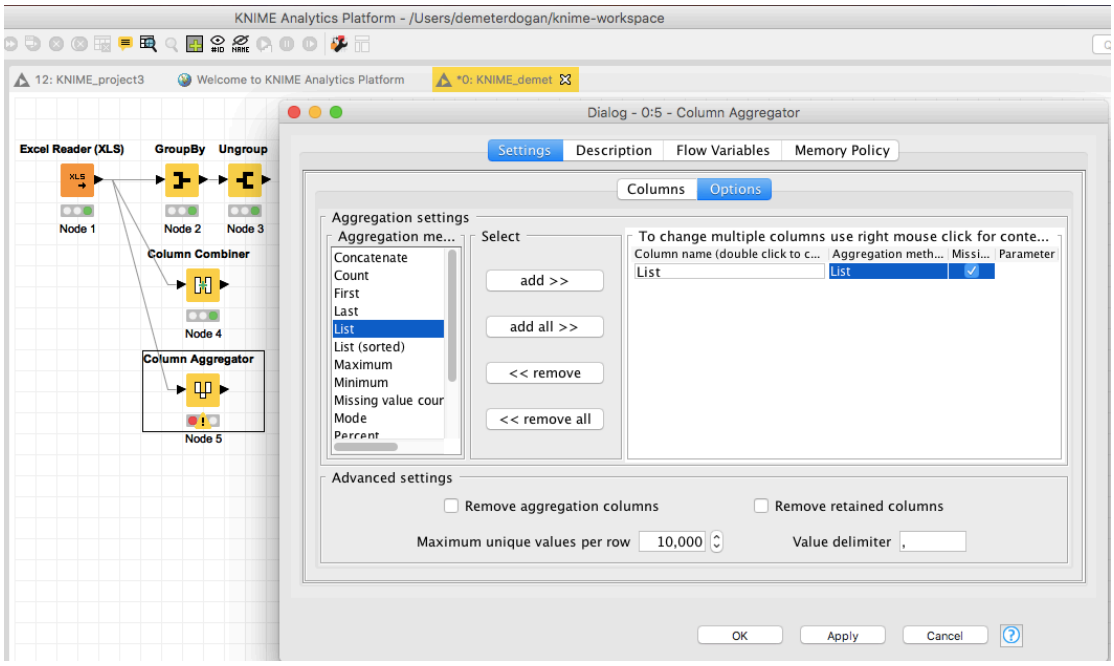
Şekil 6.4.11, Csv (comma separated) formatındaki gibi üç kolonu da birleştirmiş halini göstermektedir.

Diğer bir örnek olarak da column aggregator gösterilebilir. Yukarıdaki bölümde açıklanan gibi burada ki satır yerine kolonları toparlamak anlamına gelmektedir. Column combiner string olarak birleştirirken , column aggregatator veri yapısı olarak birleştirme yapmaktadır.



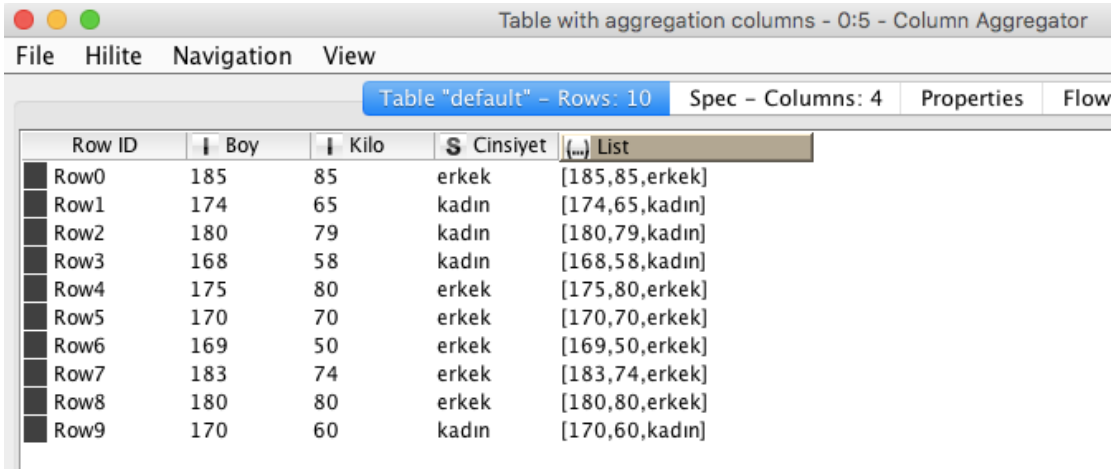
Şekil 6.4.12

Şekil 6.4.12, column aggregator operatörünün sisteme eklenmesi/ bağlantıları ve configure penceresini göstermektedir. Yine üç veriyi (kolon da) seçilmiştir. Önce aggregate metodu seçilmesi gerekmektedir. Bu yüzden bu pencere içerisinde olan options butonuna basılarak bu sekmeye geçerek kolonlar listeye göre birleştirilmeli.



Şekil 6.4.13

Şekil 6.4.13, options penceresinde yapılan değişikliği göstermektedir. Seçilen üç kolonun da liste formatında birleştirilmesi istenmiştir. Program çalıştırıldığında satır satır listeler oluşturulur. Yukarıdaki örnekte bütün kolonu tek bir listeye atarken burada her satırı ayrı bir listeye atar. Bu daha sonra split edilebilir.



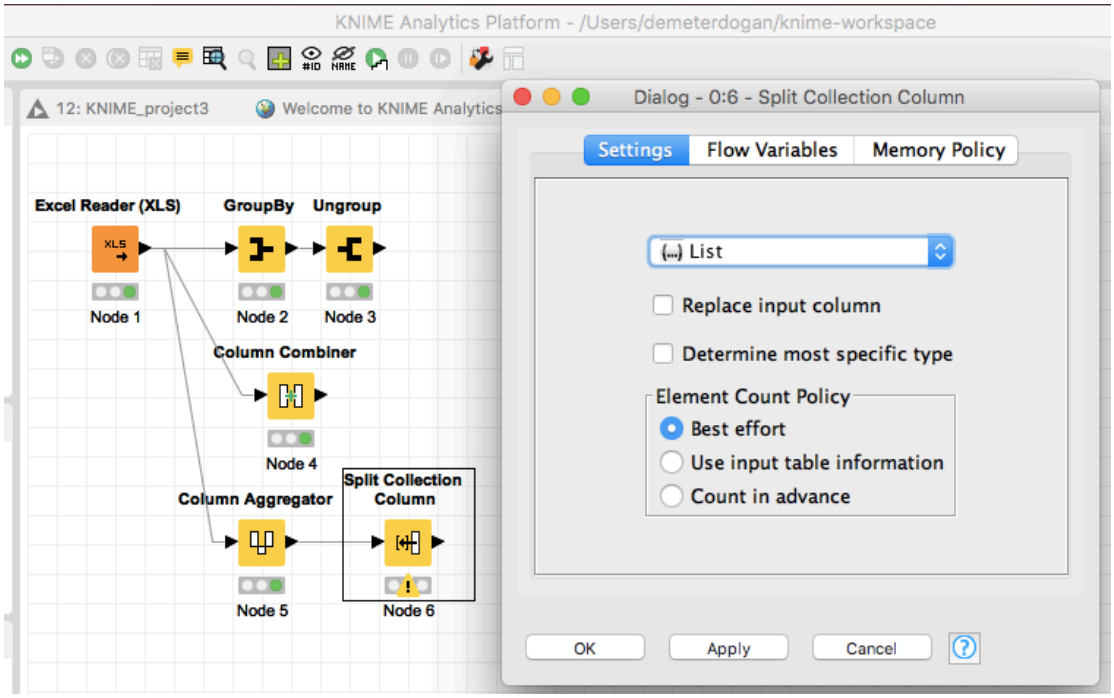
Row ID	Boy	Kilo	Cinsiyet	List
Row0	185	85	erkek	[185,85,erkek]
Row1	174	65	kadın	[174,65,kadın]
Row2	180	79	kadın	[180,79,kadın]
Row3	168	58	kadın	[168,58,kadın]
Row4	175	80	erkek	[175,80,erkek]
Row5	170	70	erkek	[170,70,erkek]
Row6	169	50	erkek	[169,50,erkek]
Row7	183	74	erkek	[183,74,erkek]
Row8	180	80	erkek	[180,80,erkek]
Row9	170	60	kadın	[170,60,kadın]

Şekil 6.4.14

Şekil 6.4.14, üç kolonun birleştirilerek listelenmiş hali dördüncü kolon olrak eklenmiş hali görülmektedir.

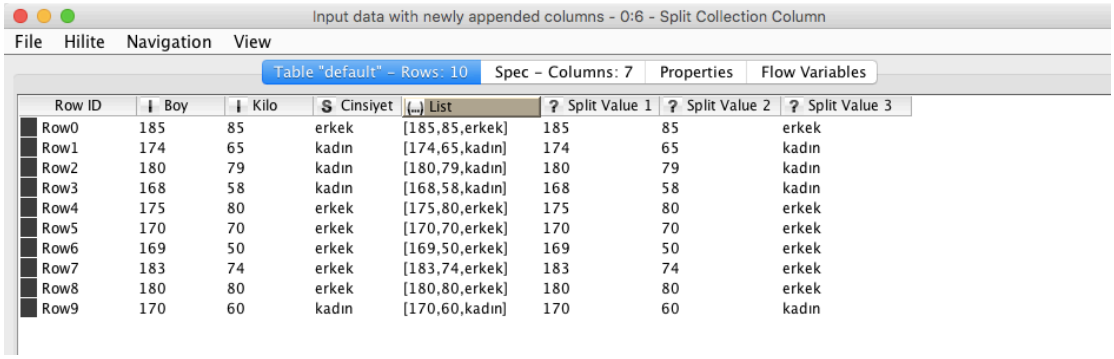
Split etmek içinde “Split Collectıon Column” kullanılır. Bu collection aslında yukarıda yapılan (aggregate) edilen ayırmada kullanılır.

Configure diyerek neyi split edeceğini belirtiyoruz. Liste yapısını split etmesini isteyelim.



Şekil 6.4.15

Şekil 6.4.15, sisteme split collection column node'unun eklenmesini ve bağlantılarını ayrıca bu node'un configure penceresini göstermektedir. Buraya liste seçilerek onun split edilmesi istendiği için program bu şekilde execute edilir.



Row ID	Boy	Kilo	Cinsiyet	List	Split Value 1	Split Value 2	Split Value 3
Row0	185	85	erkek	[185,85,erkek]	185	85	erkek
Row1	174	65	kadın	[174,65,kadın]	174	65	kadın
Row2	180	79	kadın	[180,79,kadın]	180	79	kadın
Row3	168	58	kadın	[168,58,kadın]	168	58	kadın
Row4	175	80	erkek	[175,80,erkek]	175	80	erkek
Row5	170	70	erkek	[170,70,erkek]	170	70	erkek
Row6	169	50	erkek	[169,50,erkek]	169	50	erkek
Row7	183	74	erkek	[183,74,erkek]	183	74	erkek
Row8	180	80	erkek	[180,80,erkek]	180	80	erkek
Row9	170	60	kadın	[170,60,kadın]	170	60	kadın

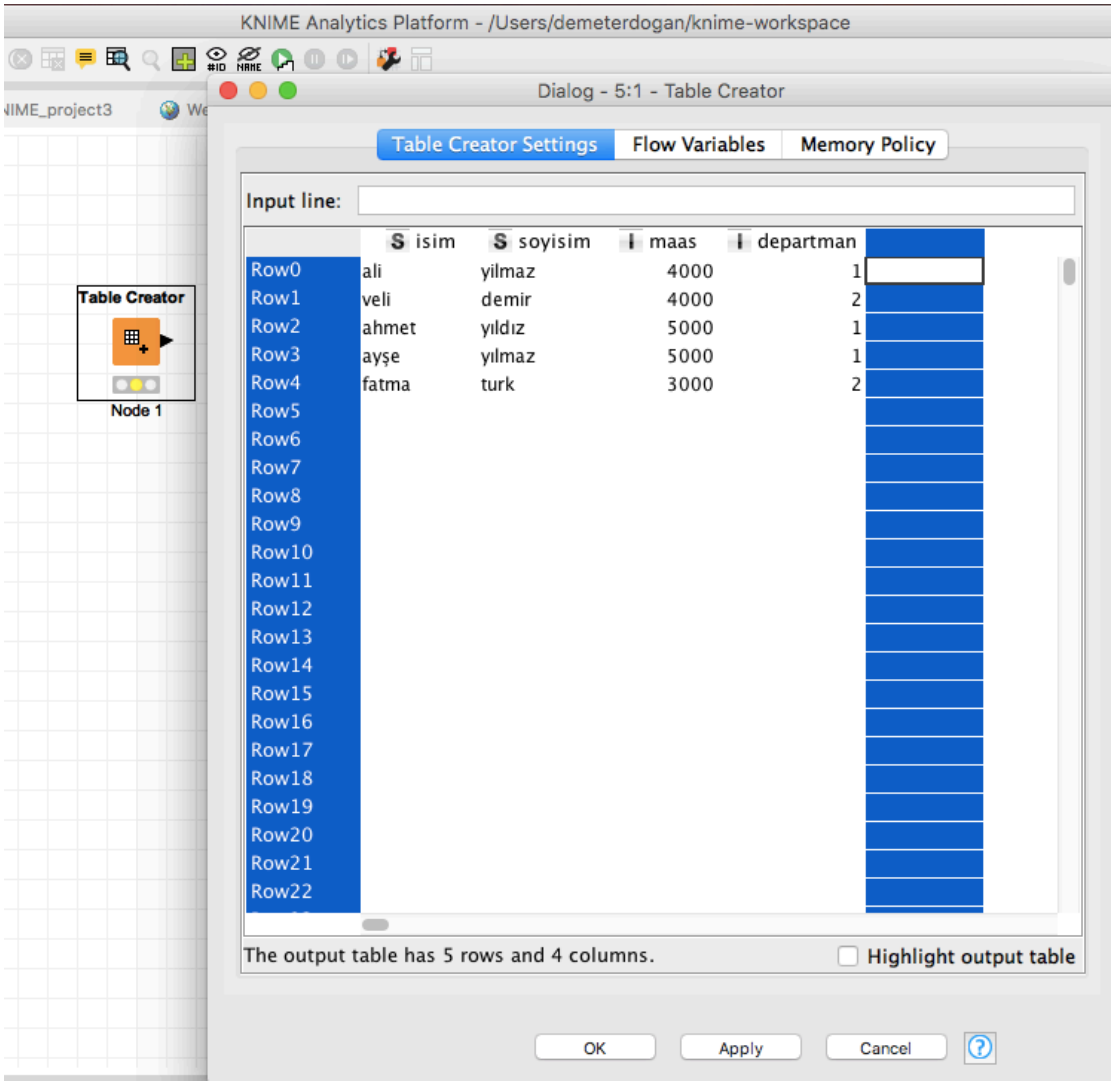
Şekil 6.4.16

Şekil 6.4.16, Yukarıdaki split komutundan sonra execute edilen programın sonuç penceresini göstermektedir. Boy, kilo ve cinsiyetten oluşan liste verisin tekrar bölünerek split value1, split value2, split value3 şeklinde yeni kolonlara yazıldığı görülmektedir..

ETL 'de preprocess için kullanışlı araçlardır. Node Repository kısmındaki diğer araçlar da denenerek kullanılabilir. Çok sayıda tool olmasına rağmen en temelde ve genel olarak işe yarayacak birçok kriter gösterildi ve gösterilecektir. Bu bölümde de veri üzerinde yapılabilecek temel ön işleme işlemlerinden bahsedildi.

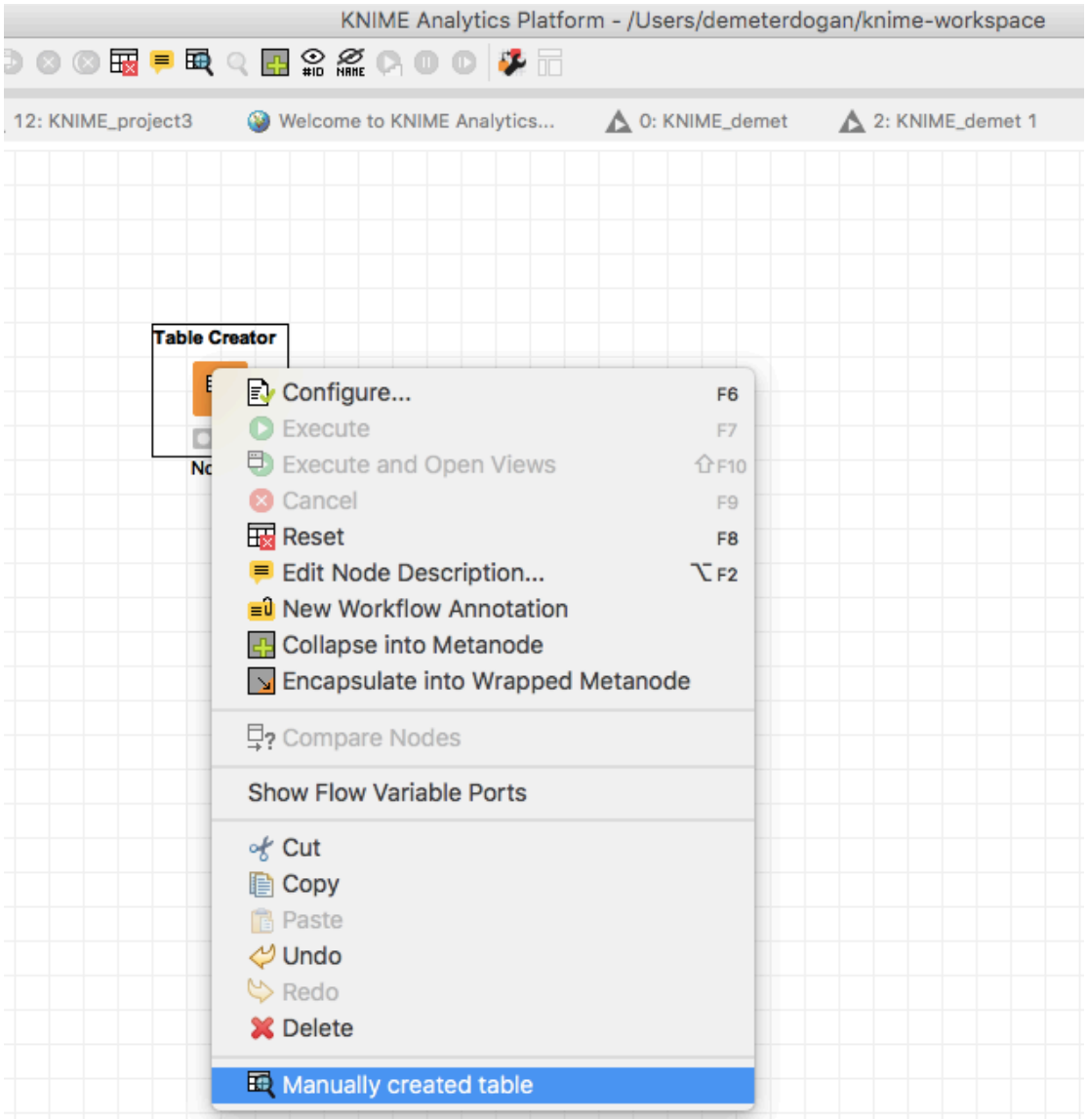
6.5 Birleştirme (Join) ve Üleştirme (Concantenation)

Bu bölümde amaç birden fazla veri kaynağından gelen verilerin ortak bir kümede birleştirilmesini (join) göstermektir. Bu özellik daha çok veritabanı teorisinden gelen bir özelliktir. Burada veri seti workflow üzerinde table creator ile oluşturulacaktır. İstenildiği kadar satır ve kolon girilebilmektedir. Table Creator düğümünü çalışma alanına bıraktıktan sonra configure ederek aşağıdaki şekilde veriler girilebilir.



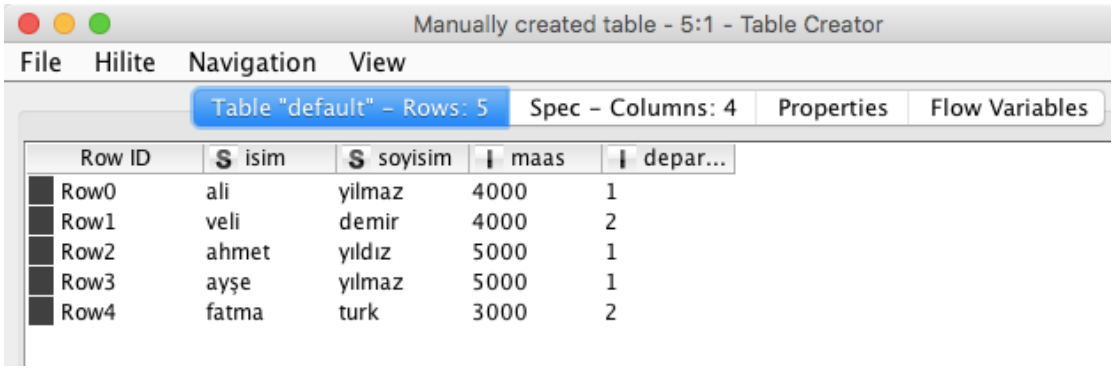
Şekil 6.5.1

Şekil 6.5.1, workflow'a aktarılan table creator node'unu ve onun configure penceresini göstermektedir. Kolonlara çift tıklanarak istenilen değerler girilebilir. Ayrıca başlıklar da çift tıklanarak açılan pencereden güncellenebilir. Açılan ufak pencereden ayrıca kolonun tipi de seçilebilir. Örneğin buraada isim ve soyisim string, maas ve departman ise integer olarak seçilmiştir.



Şekil 6.5.2

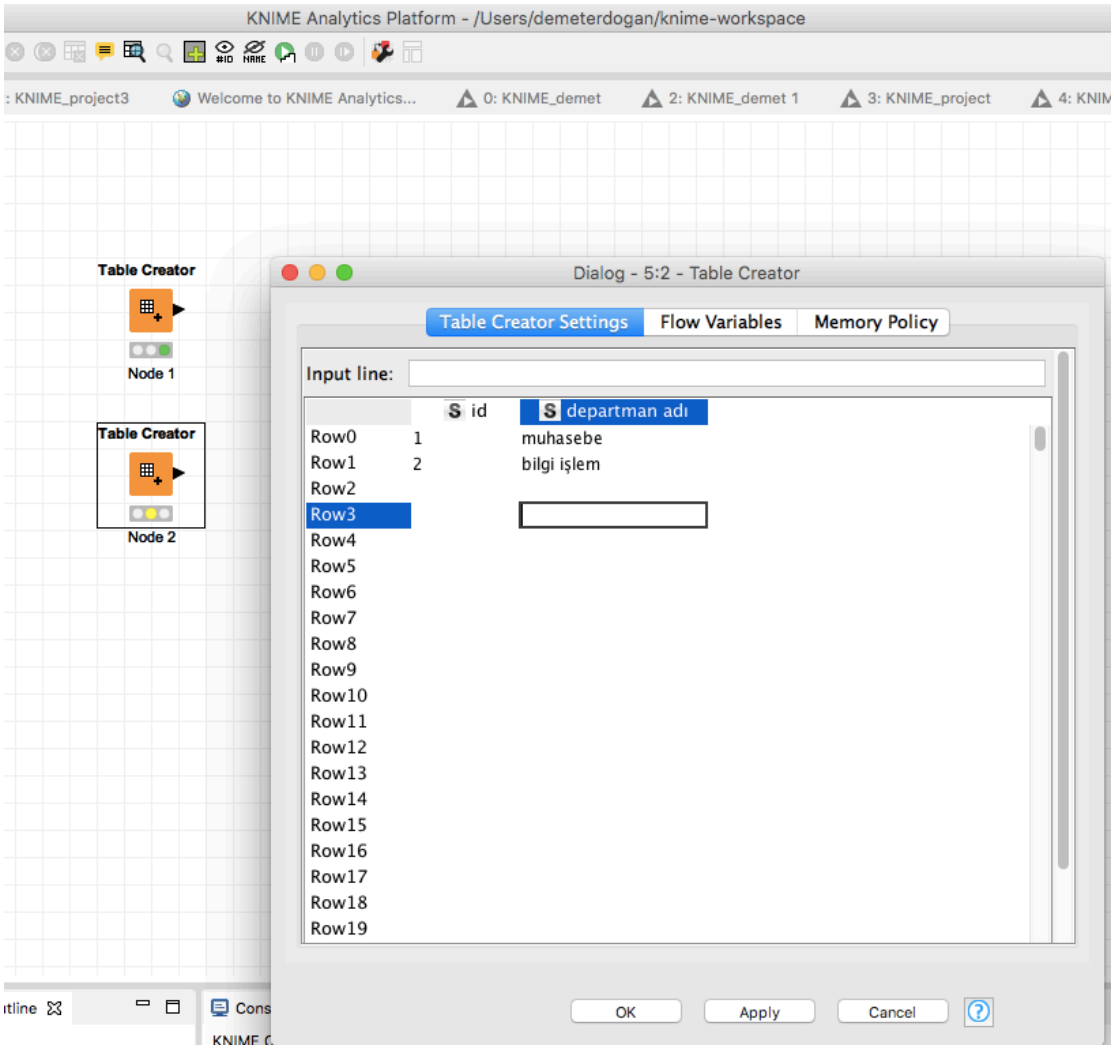
Şekil 6.5.2'de görüldüğü gibi programı çalıştırdıktan sonra sağ tuşa basıp en alttaki "manually created table" seçeneğine basılarak oluşturulan tablo görülebilir.



Row ID	isim	soyisim	maas	depar...
Row0	ali	yilmaz	4000	1
Row1	veli	demir	4000	2
Row2	ahmet	yıldız	5000	1
Row3	ayşe	yılmaz	5000	1
Row4	fatma	turk	3000	2

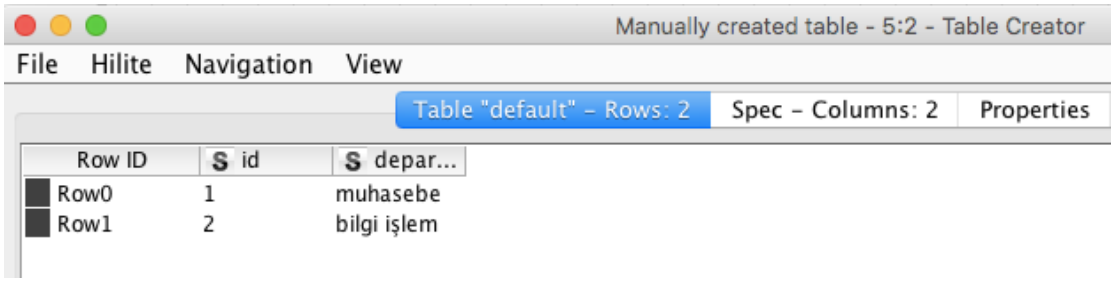
Şekil 6.5.3

Şekil 6.5.3, yukarıda oluşturulan tabloyu göstermektedir.



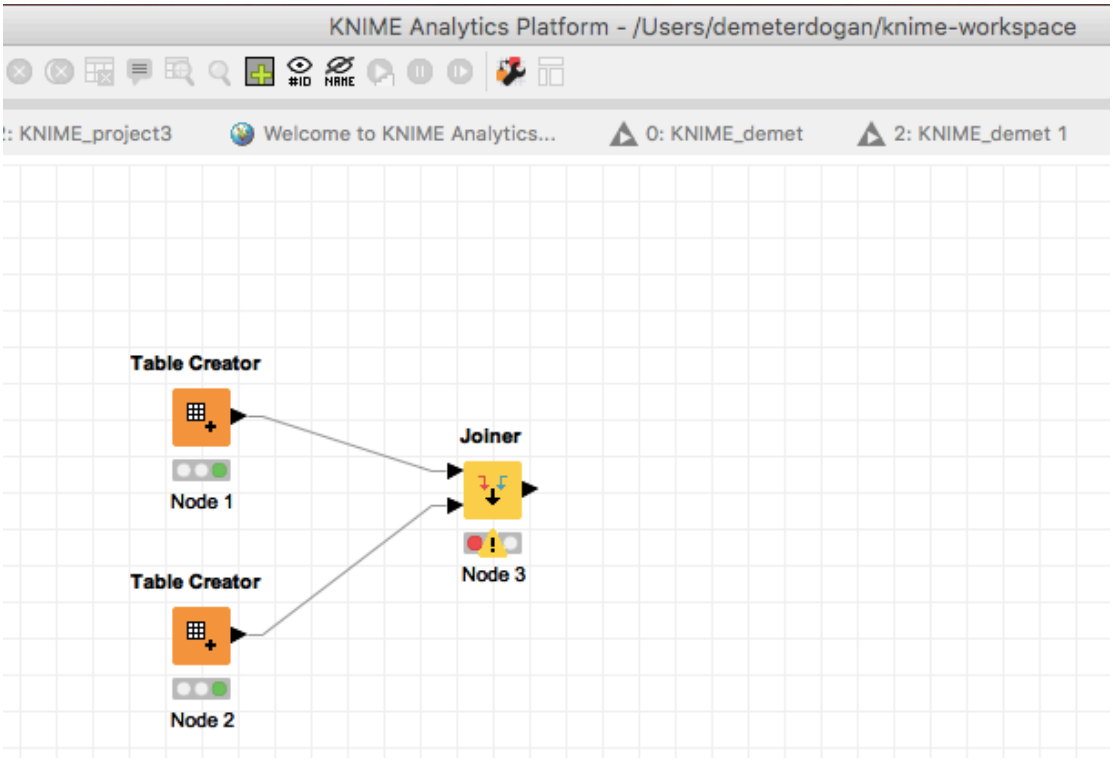
Şekil 6.5.4

Şekil 6.5.4, sisteme bir tane daha Table Creator eklenerek, bu operatörle de departman bilgilerinin tutulması için tablo oluşturulması gösterilmektedir.



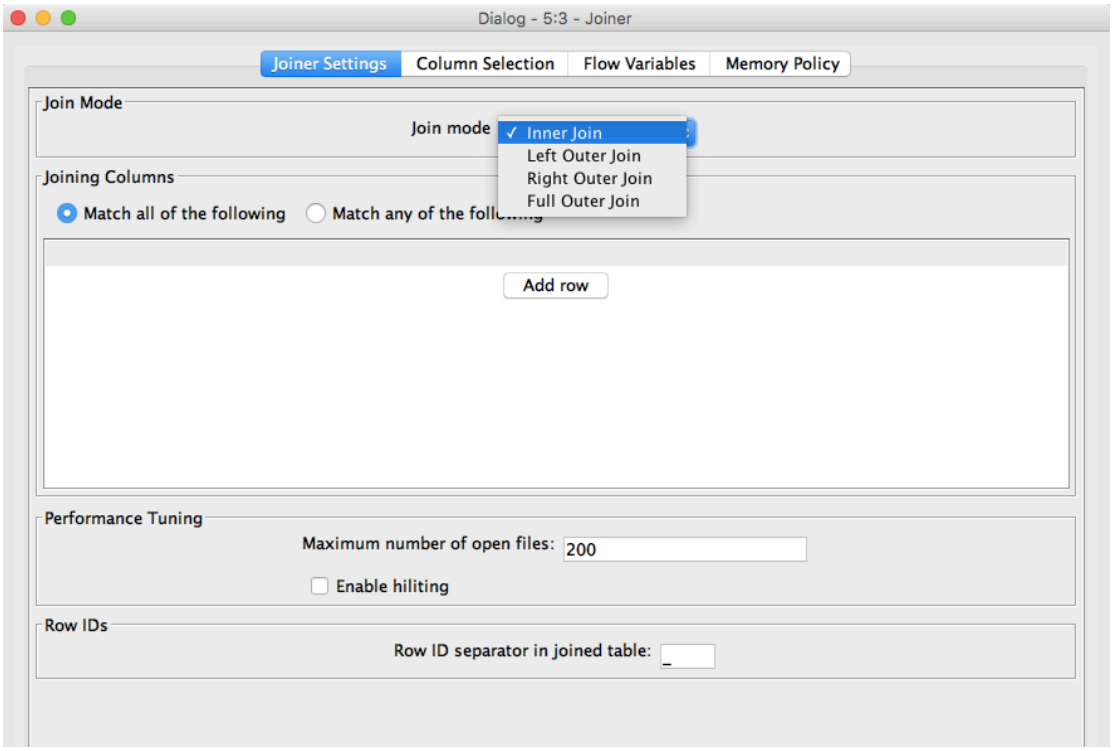
Şekil 6.5.5

Şekil 6.5.5, oluşturulan tablonun manually created table bölümünden açılan penceresini göstermektedir. Bu örnekte yapacak işlem, her çalışanın hangi departmanda olduğunu ismiyle görebilmek. Örneğin 1.tabloya bakıldığında kullanıcıların departmanlarını görülebilir ama isimleri görülemez. Tabloda departman isimlerininde görmesi için join kullanılması gerekecektir.



Şekil 6.5.6

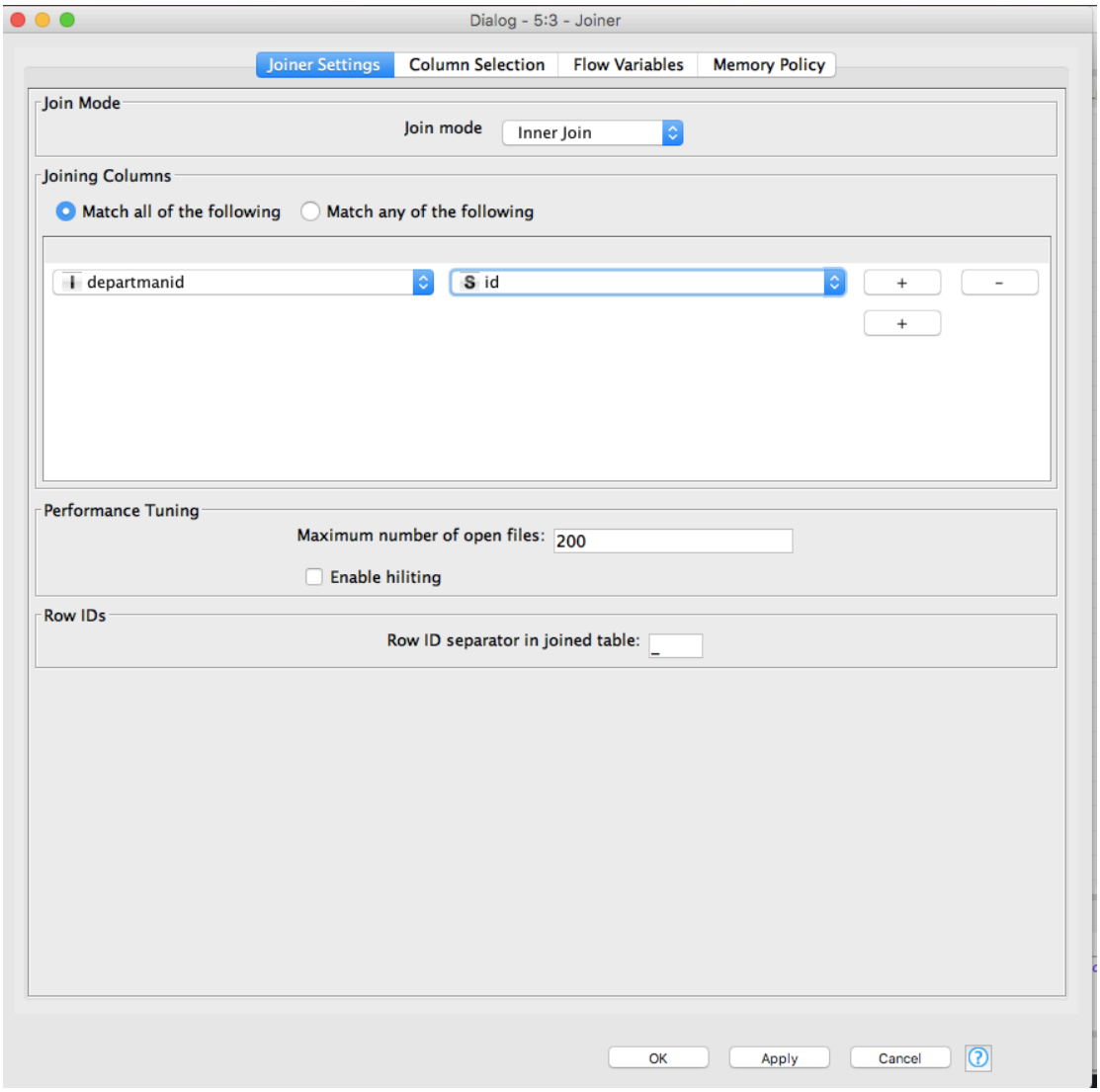
Şekil 6.5.6, sisteme joiner eklenmesini ve diğer iki table creator ile bağlantısını göstermektedir. Joıner configure edilirken Joın mode kısmında dört tip join tipi görülür.



Şekil 6.5.7

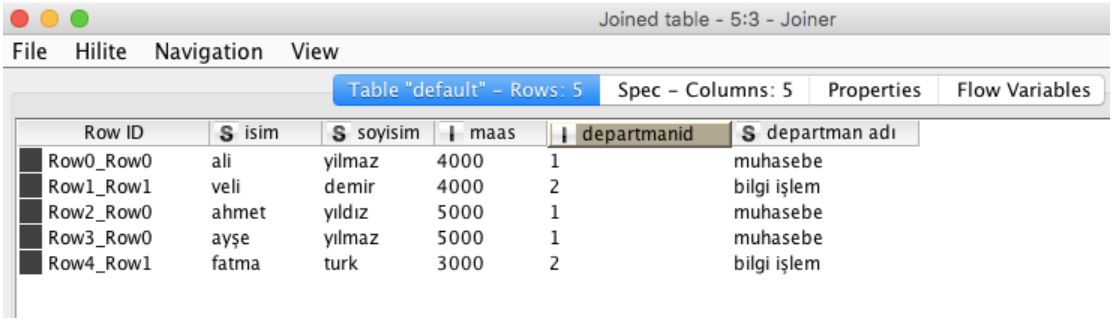
Şekil 6.5.7'de joiner operatörünün configure penceresindeki join mode'da açılan join tiplerini göstermektedir.

Inner joın; iki tabloda da ortak olan değerleri alıp bunları bunları bir kolonda birleştirir.



Şekil 6.5.8

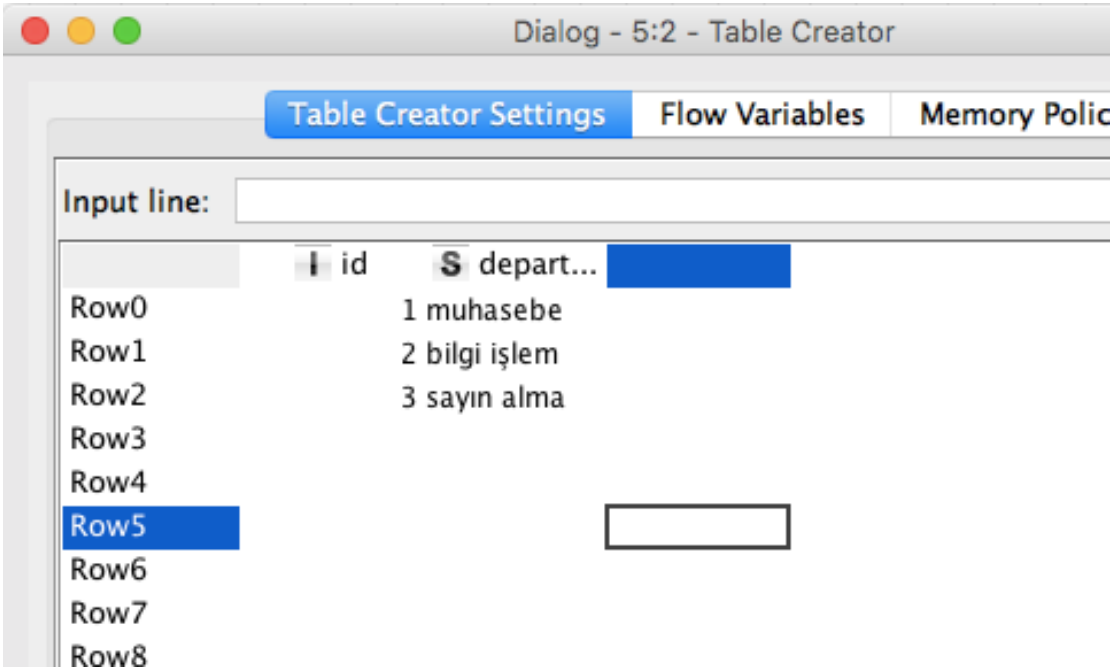
Add row dedikten sonra ilk gelen tablodan departmanid ile ikinci tablodan gelen departmanid birbirine eşitlenmesi gerekecektir. Şekil 6.5.8, bu aşamayı göstermektedir. Buradaki amaç bunların birbirine eşit olduğu durumlarda tabloda bunun gösterilmesidir.



Şekil 6.5.9

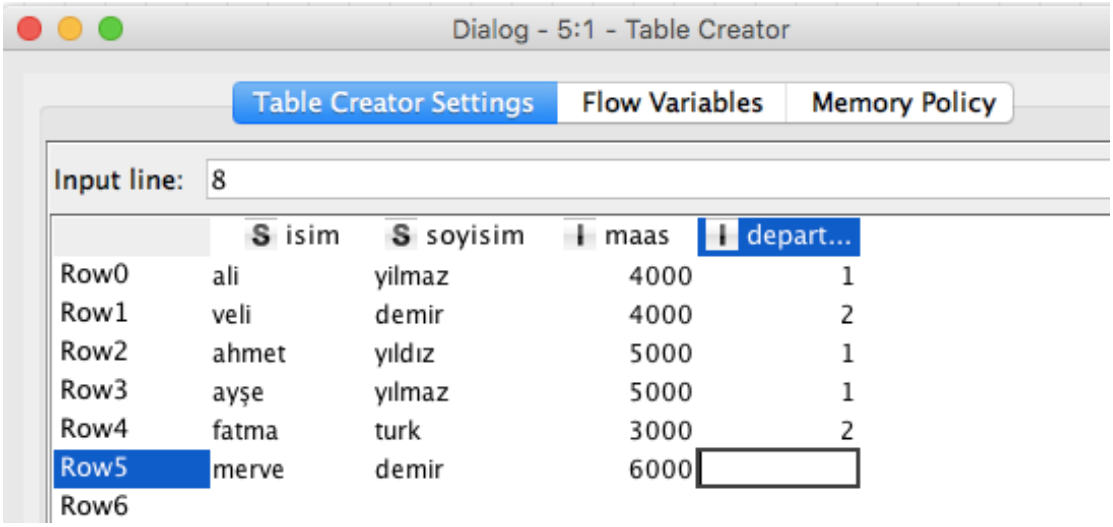
Şekil 6.5.9, program execute edildikten sonra elde edilen sonuç penceresini göstermektedir. Görüldüğü gibi rod ID'lere yani depertman id ve id kolonlarına göre tablolar birleştirilmiştir.

Diğer joın tiplerinin denenebilmesi için veride biraz değişik yapılması gerekecektir. Her iki tabloya da aşağıdaki şekilde son kayıtlar gibi örnek en azından birer satır daha eklenmelidir.



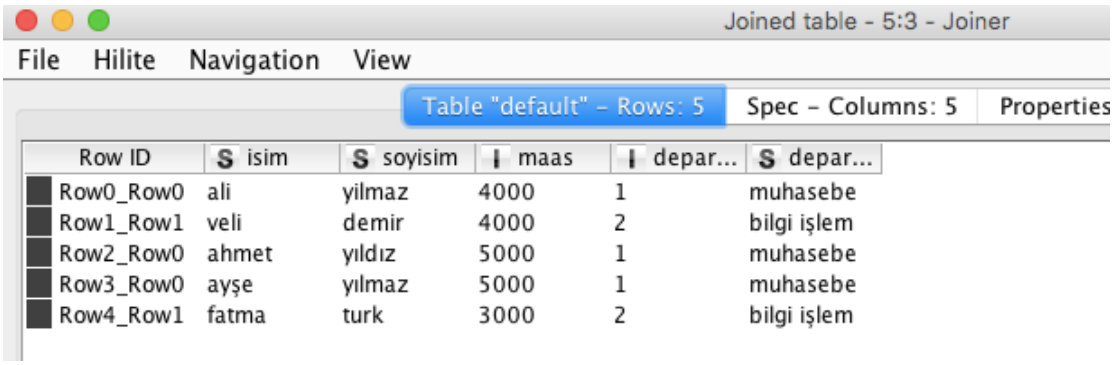
Şekil 6.5.10

Şekil 6.5.10, departman bilgilerinin olduğu tabloya yeni eklenen satırı göstermektedir.



Şekil 6.5.11

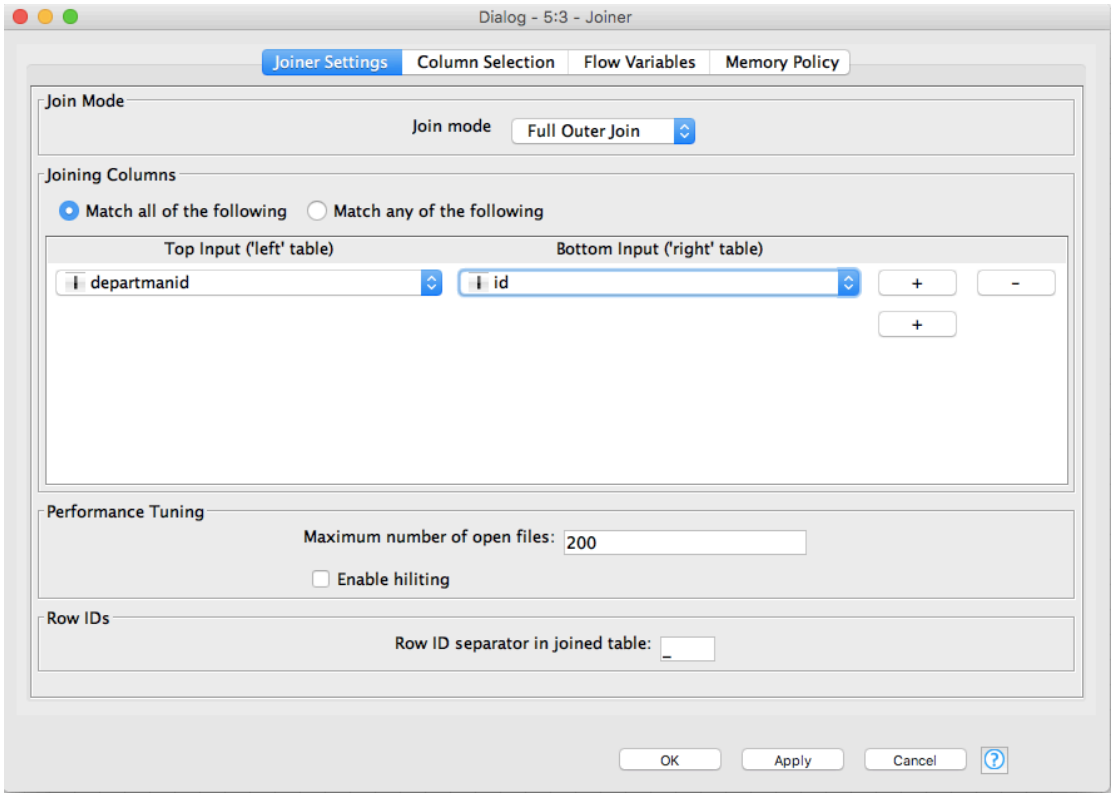
Şekil 6.5.11, ilk oluşturulan çalışan bilgilerinin olduğu tabloya eklenen yeni çalışanın bilgilerini göstermektedir.



Şekil 6.5.12

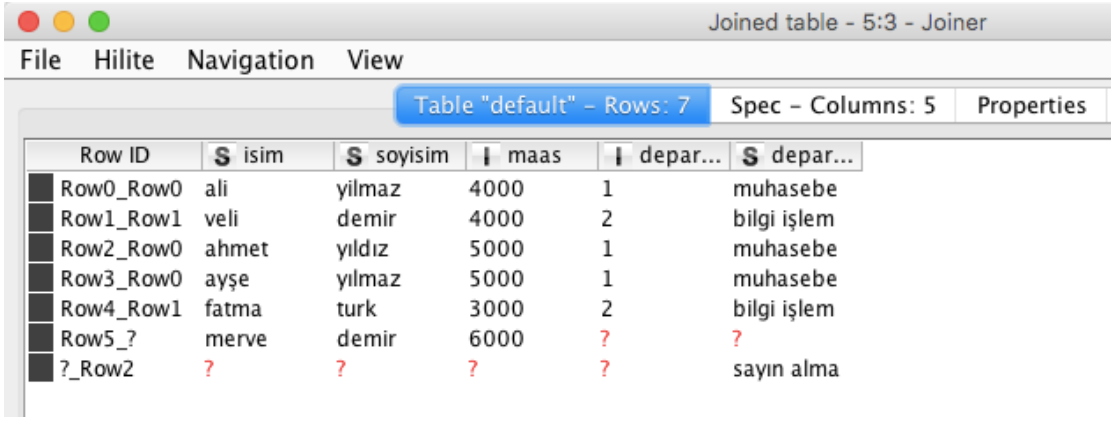
Şekil 6.5.12'de program bu şekilde execute edilip joinner'ın sonucuna bakıldığında 8 numaralı departman ve 3 numaralı departman sonuçta görülmemektedir. Bu departmanların karşılıklarında hiçbir şey olmadığı için sonuç tablosunda da getirilmemiştir. Inner joın kısaca iki tabloda da birbirini karşılayabilen durumlar olduğunda verileri getirmektedir.

Bunları da görülebilmesi için "Full Outer Joın" seçilmelidir. Outer join null (boş) durumları da alır. Yani karşılığının olmadığı durumları da alır.



Şekil 6.5.13

Şekil 6.5.13, full outer join seçilerek yapılan configure penceresini göstermektedir.

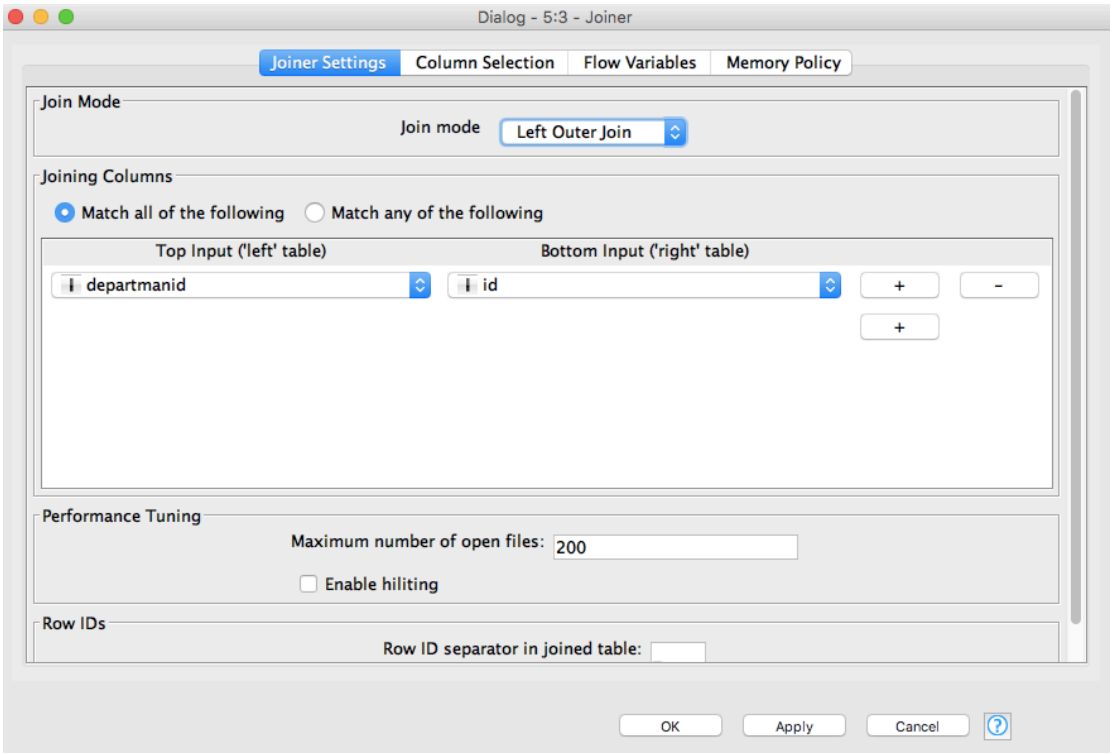


Row ID	isim	soyisim	maas	depar...	depar...
Row0_Row0	ali	yilmaz	4000	1	muhasebe
Row1_Row1	veli	demir	4000	2	bilgi işlem
Row2_Row0	ahmet	yıldız	5000	1	muhasebe
Row3_Row0	ayşe	yılmaz	5000	1	muhasebe
Row4_Row1	fatma	turk	3000	2	bilgi işlem
Row5_?	merve	demir	6000	?	?
?_Row2	?	?	?	?	sayın alma

Şekil 6.5.14

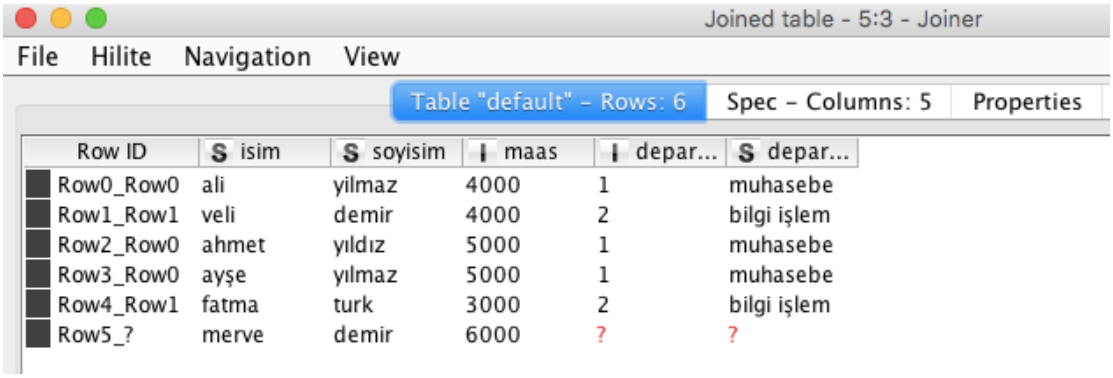
Şekil 6.5.14, full outer join seçilerek elde edilmiş sonuç tablosunu göstermektedir. Örneğin merve demir'in karşılığı yokken onu da ekledi ya da satın alma departmanının da karşılığı yokken onu da ekledi.

Left outer joın ise soldaki tablonun bütün bilgilerini getirir sağdaki tablodaki karşılıklarını boş getirir.



Şekil 6.5.15

Şekil 6.5.15, left outer join seçilerek oluşturulan configure penceresini göstermektedir.



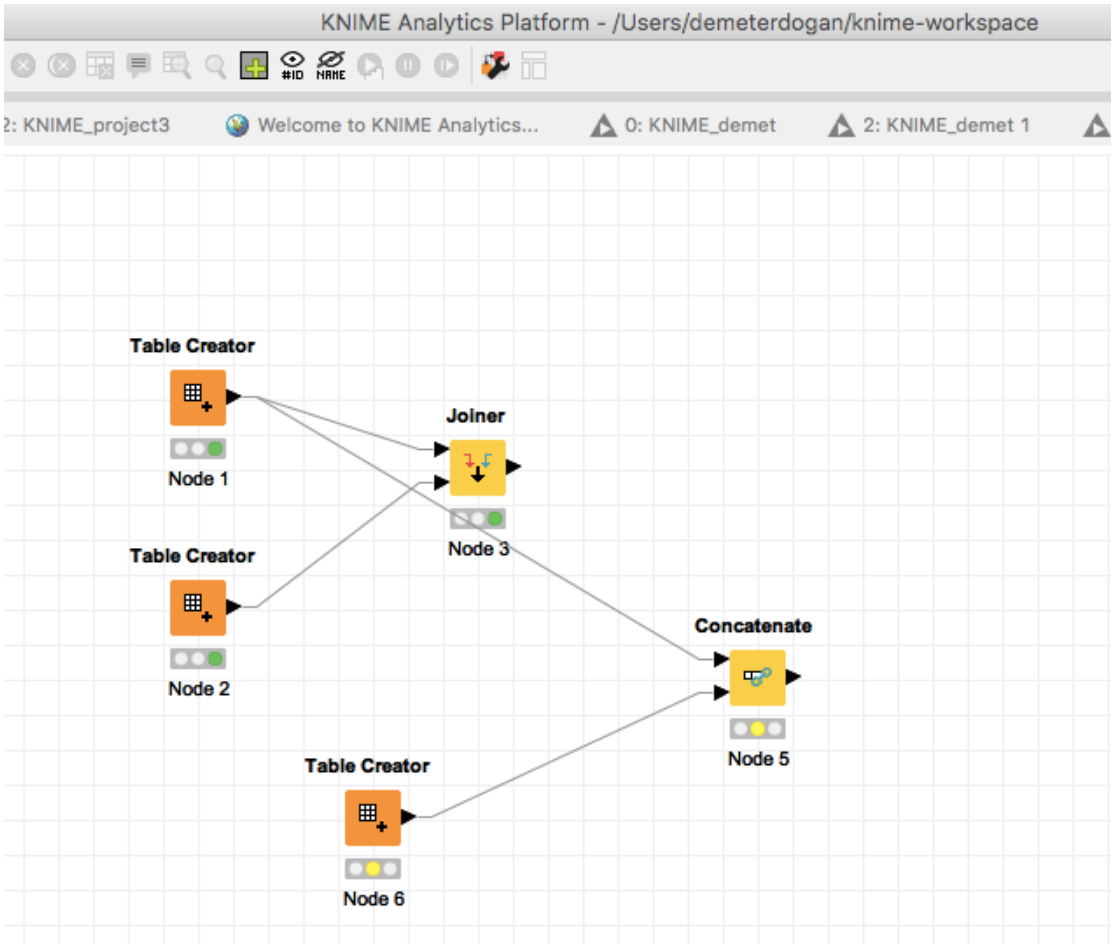
Row ID	isim	soyisim	maas	depar...	depar...
Row0_Row0	ali	yilmaz	4000	1	muhasebe
Row1_Row1	veli	demir	4000	2	bilgi işlem
Row2_Row0	ahmet	yıldız	5000	1	muhasebe
Row3_Row0	ayşe	yılmaz	5000	1	muhasebe
Row4_Row1	fatma	turk	3000	2	bilgi işlem
Row5_?	merve	demir	6000	?	?

Şekil 6.5.16

Şekil 6.5.16, left joiner a göre elde edilen sonuç penceresini göstermektedir. Şekilde de görüldüğü gibi sol tablodaki tüm bilgileri içerirken sağda karşılığı olmayanları boş göstermiş fakat sadece sağda olanları tabloya dahil etmemiştir.

Tam tersini almak için right outer join kullanılmalıdır. Yani sağ tablodaki tüm bilgileri içerirken solda karşılığı olmayanları boş göstermesi ama sağda olanların hepsini tabloya dahil etmesi için right outer join kullanılmalıdır.

Joiner, tablo birleştirmenin bir yöntemi. Diğer bir yöntemde concatenate seçeneğidir. Farklı tablolardan gelen ama aynı kolon yapısına sahip tabloları birleştirmek için concatenate kullanılır. Bu literatürde uç uca eklemek, birleştirmek diye de geçer.

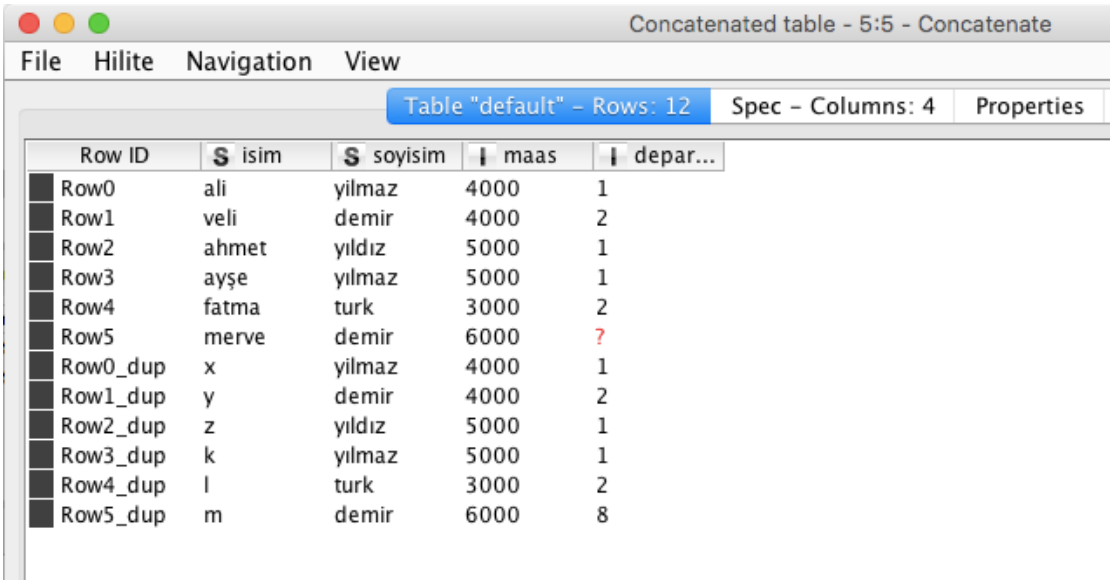


Şekil 6.5.17

Şekil 6..5.17, sisteme yeni table creator ve concatenate node'larının eklenmesini ve bağlantılarını göstermektedir.

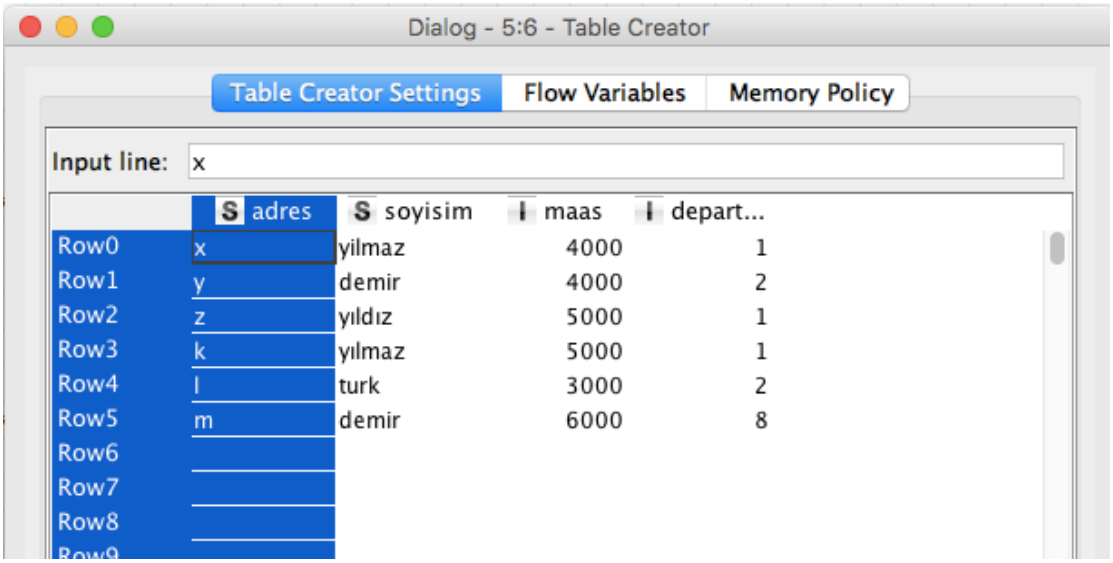
Şekil 6.5.18

Şekil 6.5.18, yeni eklenen table creator'ın configure penceresinde girilen yeni verileri göstermektedir. Örneğin her tabloda farklı çalışanlar olsun ya da bir kısım veri sunucunun üstünde duruyordur diğeri diğer sunucunun üstünde duruyordur ya da değişik şekillerde tablolar bölünmüş olabilir. Bu aynı yapıda olan tabloları birleştirmek istenirse yani karşılığını bulup diğer tablodan bilgi getirip bunları kolon olarak eklemek join idi fakat aynı kolon yapısına sahip farklı tabloları birleştirmek için concatenate (uç uca eklemek, üleştirmek) kullanılmalıdır.



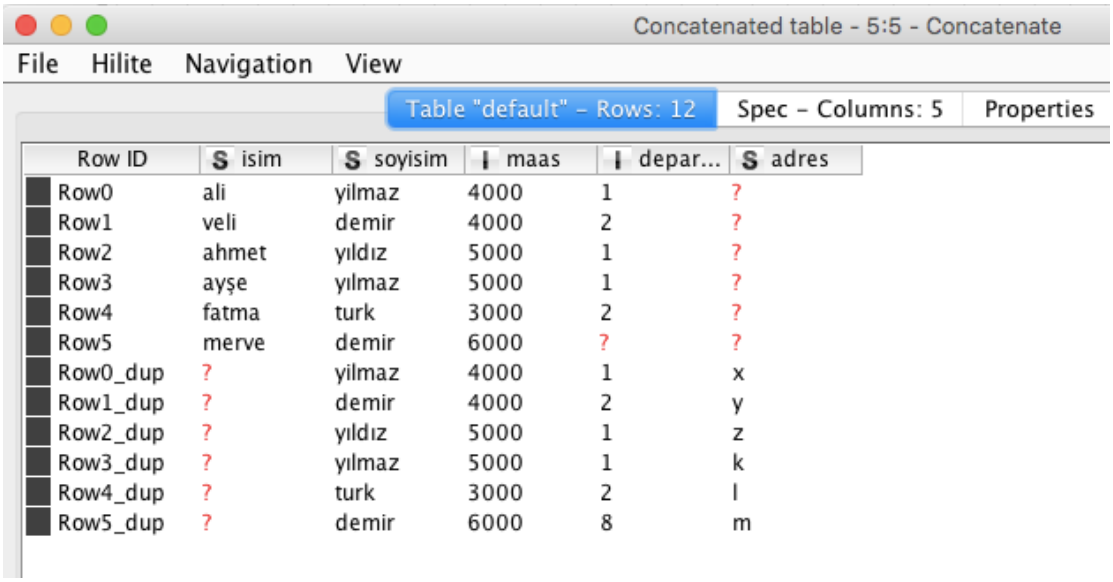
Şekil 6.5.19

Şekil 6.5.19, program çalıştırıldığında elde edilen sonuç tablosunu göstermektedir. Burada da görüldüğü gibi, row0-row5 arasındaki veriler ilk tablodan altındaki kısım ise son oluşturulan tablodan gelmiş ve her şeyin dahil olduğu yeni bir tablo oluşturulmuştur.



Şekil 6.5.20

Şekil 6.5.20, son eklenen table creator configure penceresinde yapılan değişikliği göstermektedir. İsim yazan kolonun adın adres olarak değiştirildi.



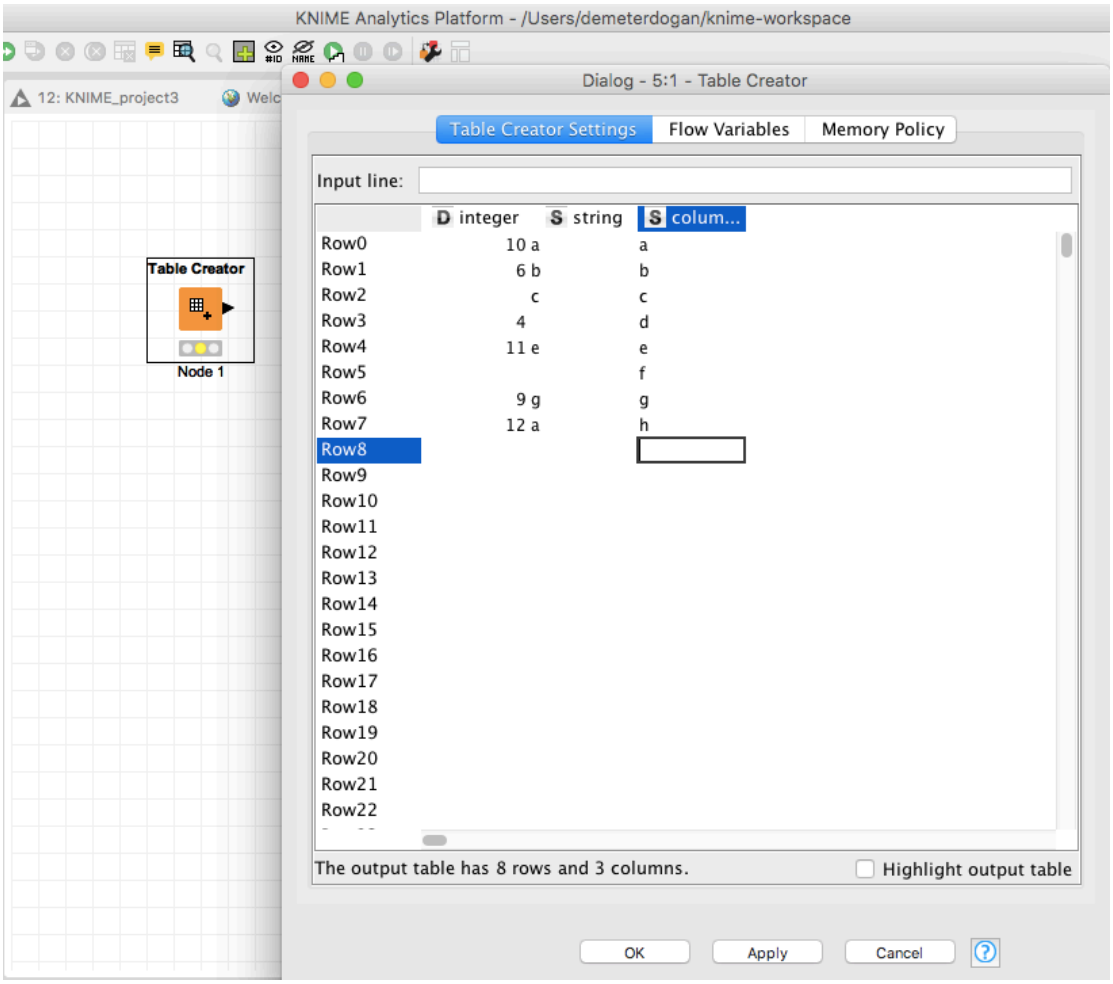
Row ID	isim	soyisim	maas	depar...	adres
Row0	ali	yilmaz	4000	1	?
Row1	veli	demir	4000	2	?
Row2	ahmet	yıldız	5000	1	?
Row3	ayşe	yılmaz	5000	1	?
Row4	fatma	turk	3000	2	?
Row5	merve	demir	6000	?	?
Row0_dup	?	yilmaz	4000	1	x
Row1_dup	?	demir	4000	2	y
Row2_dup	?	yıldız	5000	1	z
Row3_dup	?	yılmaz	5000	1	k
Row4_dup	?	turk	3000	2	l
Row5_dup	?	demir	6000	8	m

Şekil 6.5.21

Şekil 6.5.21'de de görüldüğü gibi, ilk tablodan gelen kısımda adres bilgisi olmadığı için o kolondaki o bölge boş, alttaki kısımda da adres bilgisi olup isim bilgisi olmadığı için o sıralarda da isim bilgileri mising value gelmiştir. Buradan da anlaşılacağı gibi, concatenate sadece tüm verileri uc uca eklemek için kullanılır.

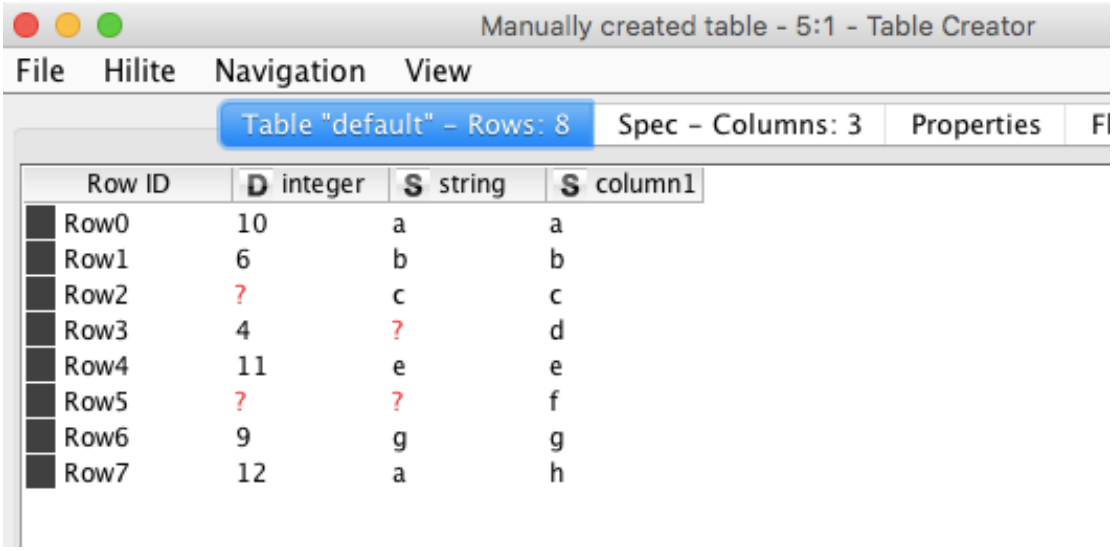
6.6 Eksik Veriler (ilk deneme)

Bu bölümde amaç, eksik verilerin nasıl düzelteceğini /giderileceğini göstermektir. Yukarıdaki bölümlerde olduğu gibi yine Table creator eklenip configure bölümünden içerisinde basitçe birkaç değerin eksik veri (missing value) olarak kaydedildiği kolonlar oluşturulmalı.



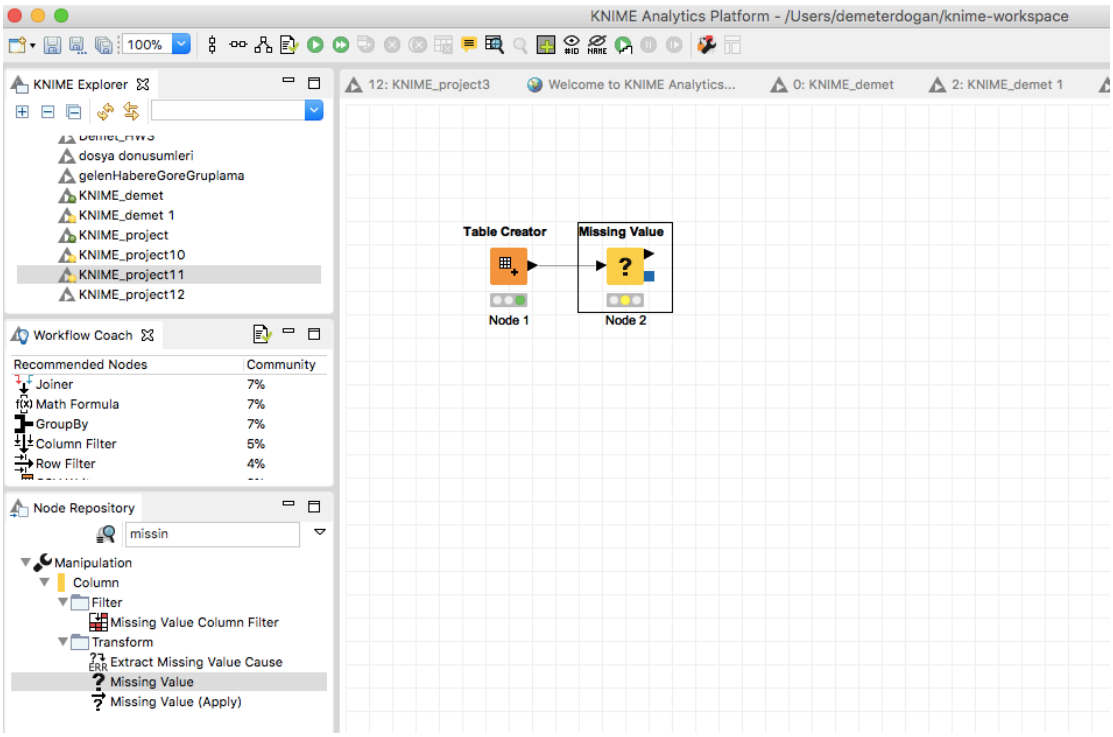
Şekil 6.6.1

Şekil 6.6.1, table creator'ın sisteme eklenmiş halini ve onun configure penceresinde girilen verileri göstermektedir. Boş hücreler bilerek herhangi bir değer girilmemiş hücrelerdir.



Şekil 6.6.2

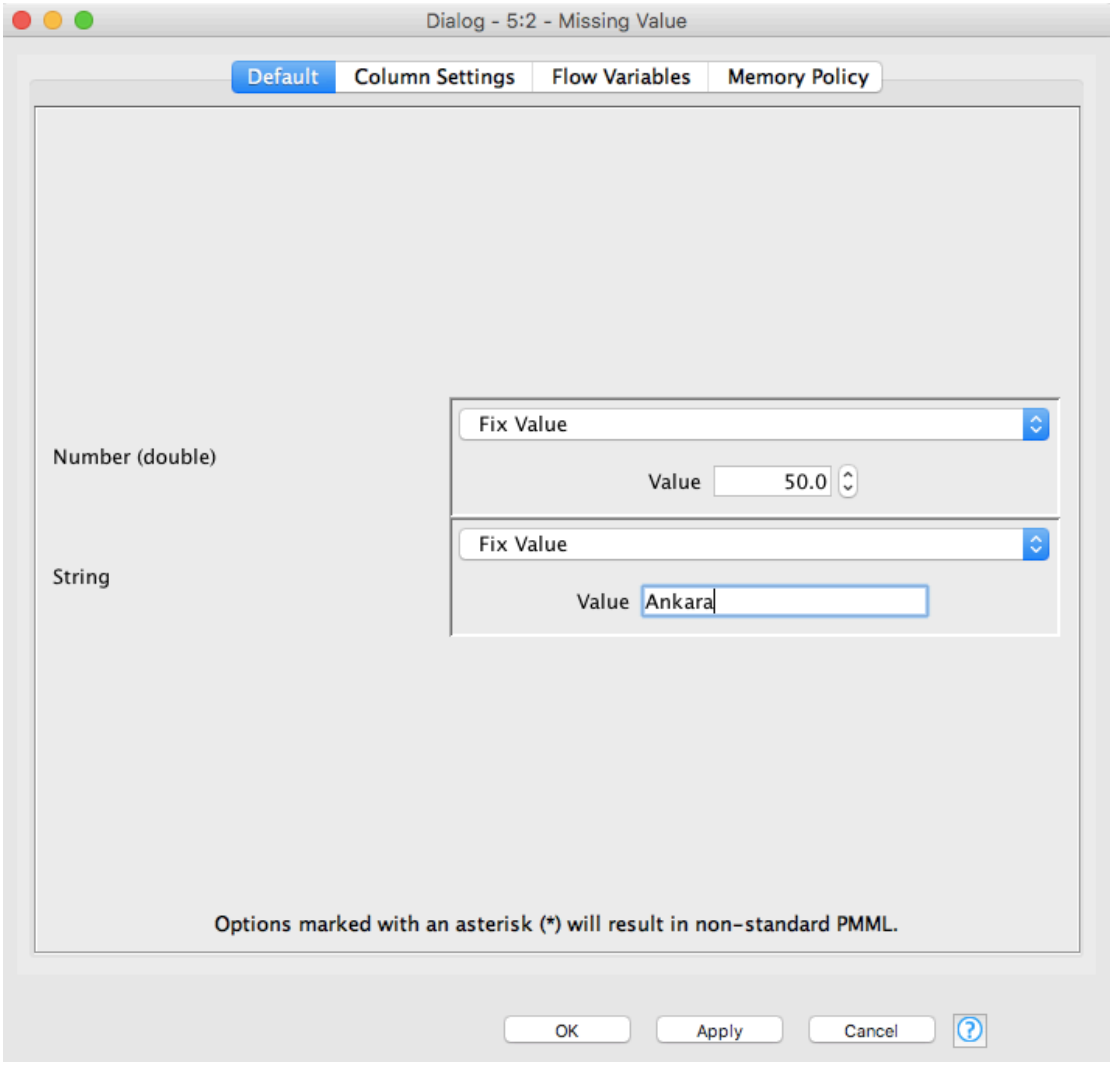
Şekil 6.6.2, yukarıda girilen veriler ile oluşturulmuş tablonun son halini göstermektedir. Soru işareti (?) olarak görülen değerler henüz girilmemiş, bilinmeyen değerler anlamına gelmektedir. Bu eksik veriler nasıl Knime tarafından düzeltilebilir veya veri biliminde, veri madenciliğinde bunlar nasıl ele alınmalı bunlar bu bölümde örnek üzerinden açıklanacaktır.



Şekil 6.6.3

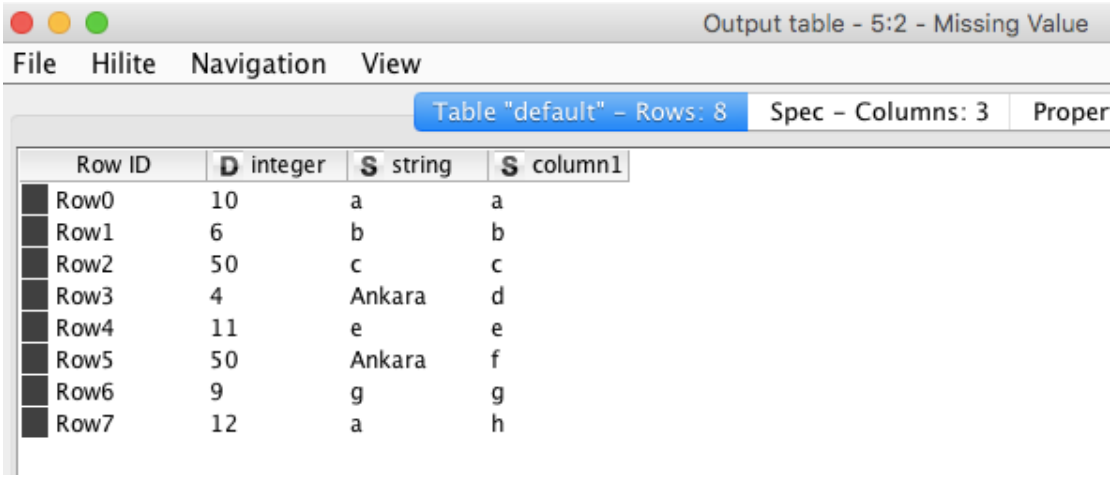
Şekil 6.6.3, Node Repository bölümünden Missing Value şeklinde aratılarak bulunan node'un workflow'a eklenmesini ve table creator ile bağlantısını göstermektedir. Daha önce filitreleme kısmında bahsedildiği üzere row filtering ve column filtering ile eksik veri (missing value) içeren herhangi bir satırı direkt silebilmek mümkündü. Bir diğer

yöntem ise eksik kısımların tahmin edilmesidir. Knime'da tahmin etmesi için değişik yöntemler bulunmaktadır.



Şekil 6.6.4

Şekil 6.6.4'de de görüldüğü gibi Missing Value'nun configure bölümünde iki tip veriden bahsedilmektedir. Bunlar: Sayı (number) veya dizgi (string)'dir. Eksik kısımlarda Sayı olan durumlarda fix value (sabit değer) koyarak örneğin burada 50 ile doldurması, String içinde bir fix value örneğin Ankara yazılması ile çözülebilir.

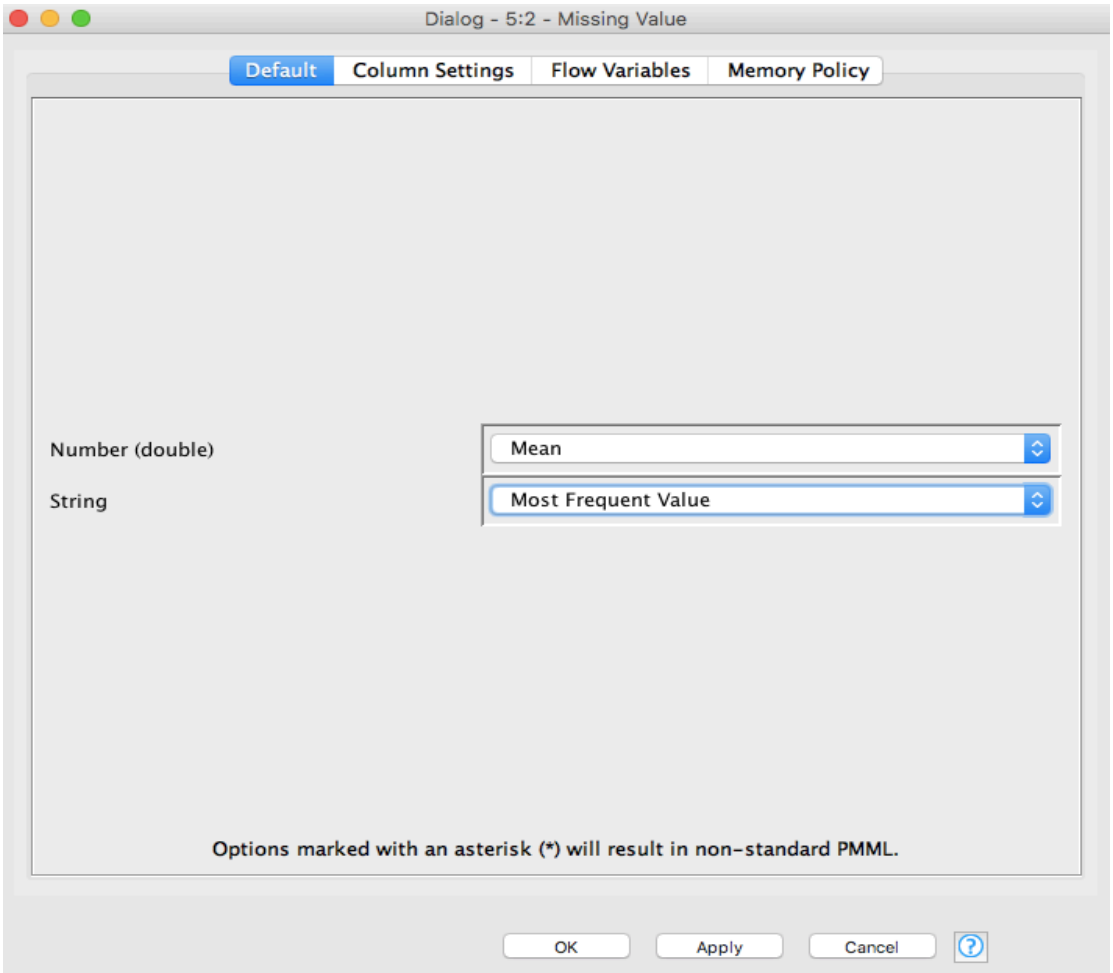


Row ID	integer	string	column1
Row0	10	a	a
Row1	6	b	b
Row2	50	c	c
Row3	4	Ankara	d
Row4	11	e	e
Row5	50	Ankara	f
Row6	9	g	g
Row7	12	a	h

Şekil 6.6.5

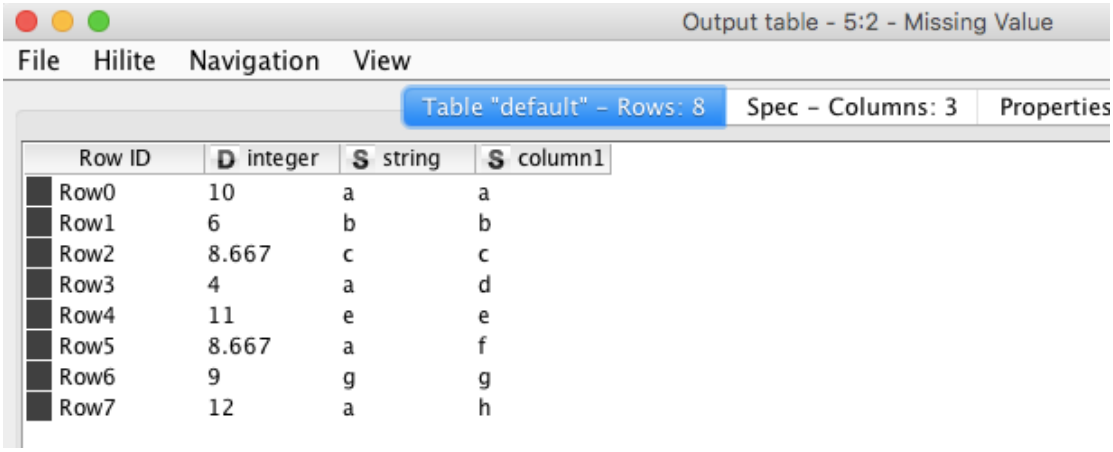
Şekil 6.6.5, Fix değerleri koyup çalıştırıldığı zaman oluşan output table'ı göstermektedir. Görüldüğü gibi eksik olan değerler bir önceki şekilde verilmiş olan fix value'larla yani sabit değerlerle doldurulmuştur.

Başka şekilde de bu eksik verileri tamamlamak mümkün.



Şekil 6.6.6

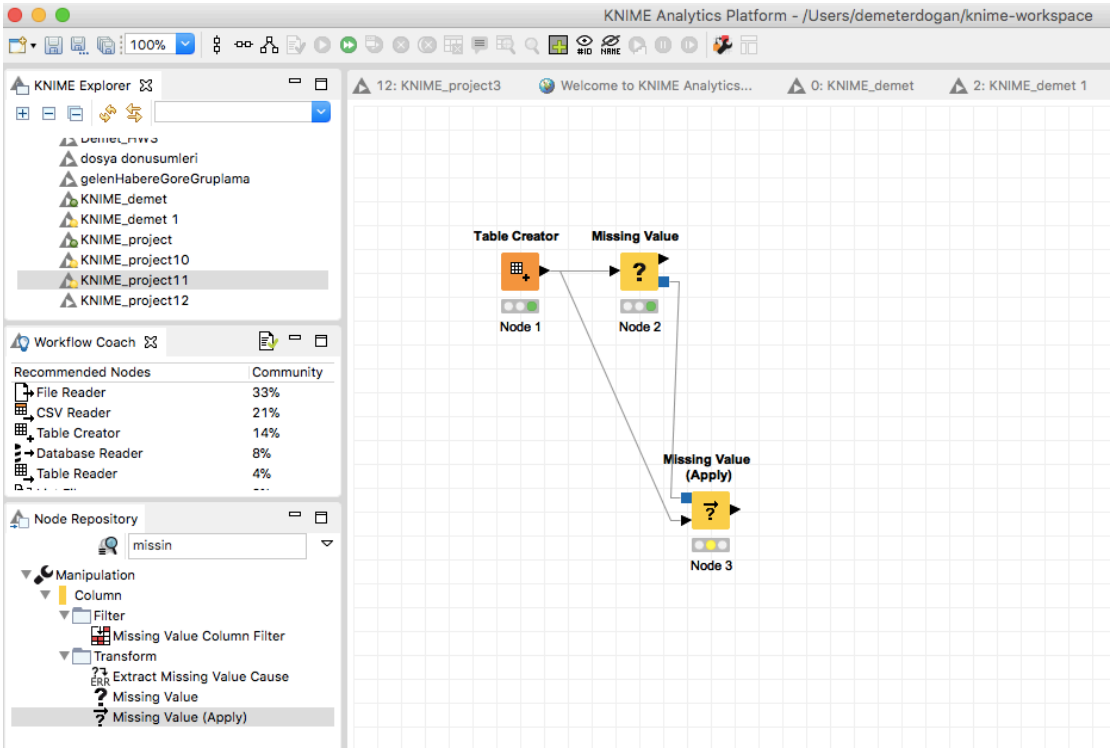
Şekil 6.6.6, missing value operatörünün configure penceresinde sayısal değerler için mean(ortalama) ile ve stringler için de (most frequent value) en fazla tekrar eden değeri ile tamamlanması komutunun girilmesini göstermektedir. Dolayısıyla herhangi bir string eksikse o kolona bakılacak o kolon içerisinde en fazla tekrarlı değer eksik verilere yazılacaktır. Şayet bir sayı eksikse kolona bakılacak ve o kolondaki bütün sayıların ortalama değeri eksik değerlerin yerine yazılacaktır. Ayrıca, number için ortanca, minimum, maximum vb. Değerler ve string için ise bir önceki, bir sonraki vb. Değerler ile eksik verilerin giderilme seçenekleri de seçilebilir.



Row ID	integer	string	column1
Row0	10	a	a
Row1	6	b	b
Row2	8.667	c	c
Row3	4	a	d
Row4	11	e	e
Row5	8.667	a	f
Row6	9	g	g
Row7	12	a	h

Şekil 6.6.7

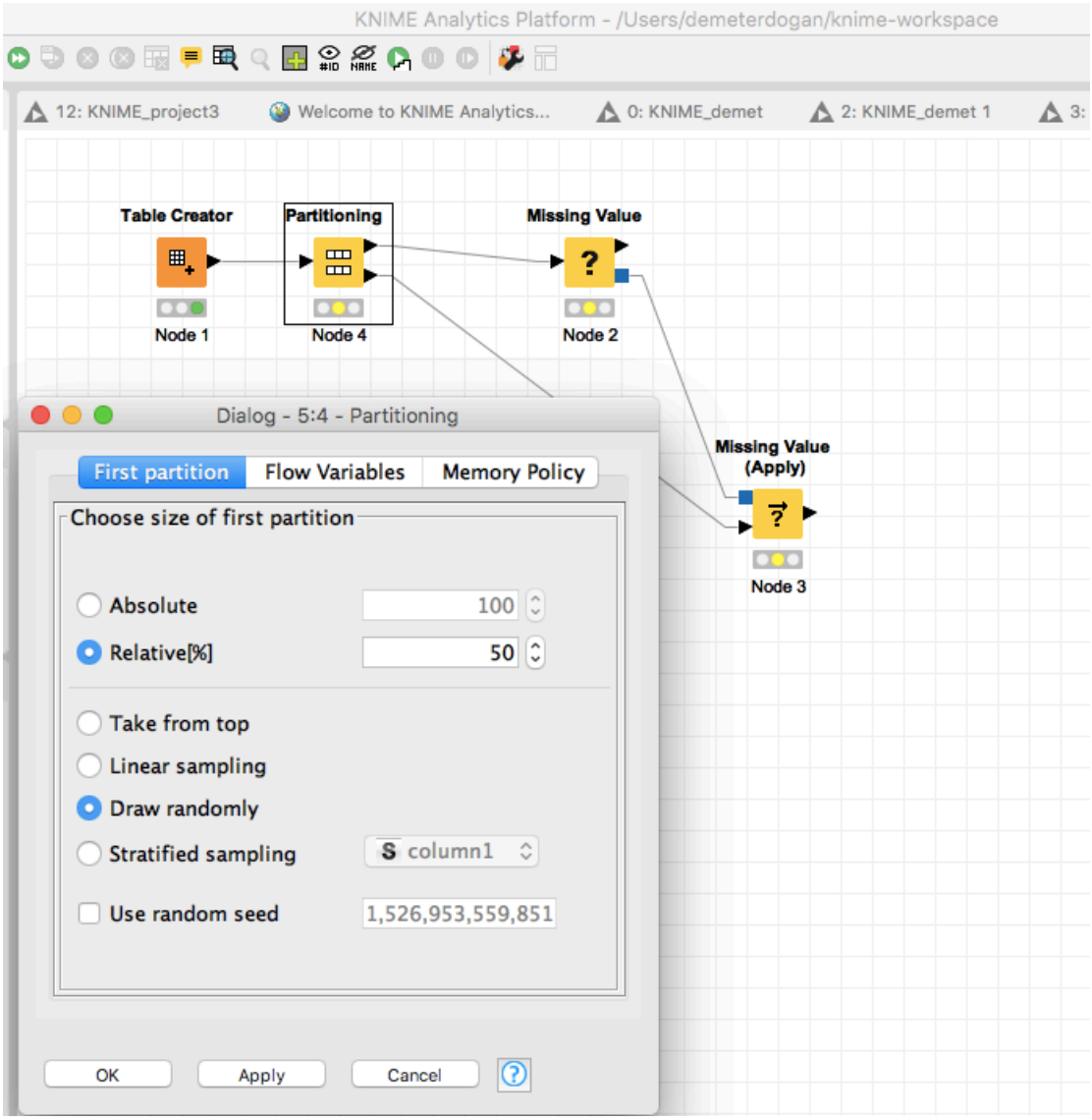
Şekil 6.6.7, oluşan output table'ı göstermektedir. Buradan yola çıkarak, ortalama değerin 8.667 ve en fazla tekrar eden değerin ise "a" harfi olduğu eskik veri olan bölgelere bu değerlerin yazılmasıyla anlaşıldı.



Şekil 6.6.8

Şekil 6.6.8, Sisteme missing value (apply) operatörünün eklenişini ve bağlantıları göstermektedir. Görüldüğü gibi missing value node'unun bir de apply özelliği vardır. Mavi köşeli olan kareler aslında bir PMML outputu verir. Yani bir makine öğrenmesi sonucu bir model oluşturur. İstatistiksel bir model veya makine öğrenmesi ile gelen bir model ve bu çıktılar daha sonra başka programlarda da kullanma imkanı sağlar. Örneğin Knime'da öğretilen bir model daha sonrasında rapid miner'da kullanılabilir. Şuan örnek olması açısından bu örnekte kullanılacak fakat aynı veri kümesi içinde kullanılması normalde çok anlamlı değildir.

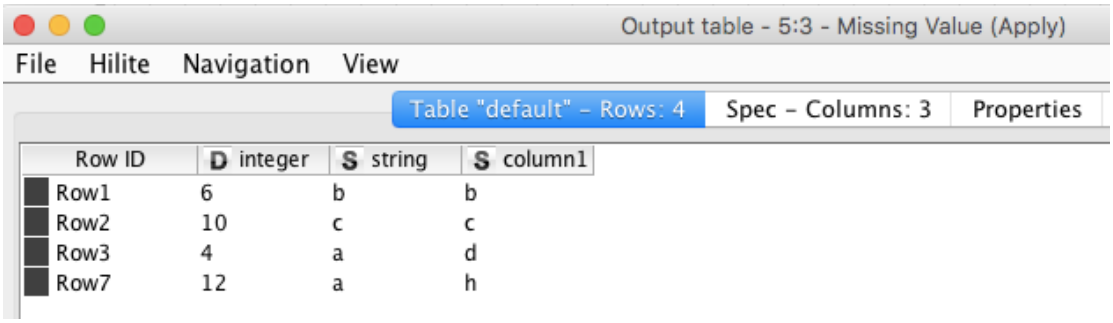
Gelen verileri partitioning ile parçalara bölünerek bir kısmı öğrenmesi bir kısmı da öğrenilenin test edilmesi için kullanılabilir.



Şekil 6.6.9

Şekil 6.6.9, sisteme partitioning operatörünün eklenmesini ve bağlantılarını ve ayrıca bu operatörün configure penceresini göstermektedir. Program çalıştırıldığı zaman veri ikiye bölünmüş olur. İlk %50 lik grup missing value kısmına ikinci %50 lik kısım missing

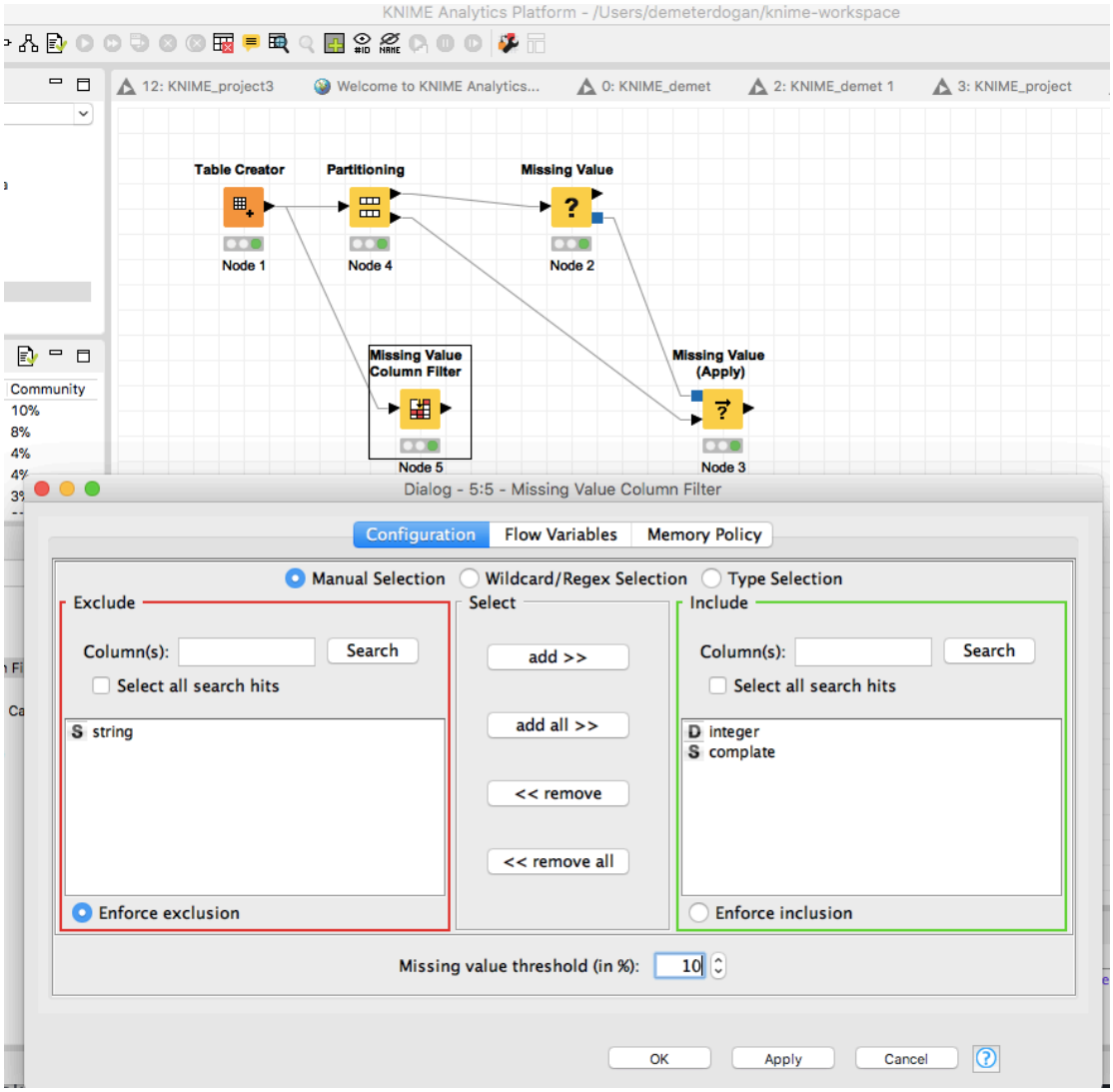
value (apply) uygulamasına gider. Buradaki sistemde öğrenme farklı verilerden, uygulama farklı verilerden yapılmaktadır.



Şekil 6.6.10

Şekil 6.6.10, program çalıştırıldıktan sonra missing value (apply) dan elde edilen output table'ı göstermektedir. Veri seti yarıya bölündüğü için çıkan sonuçta da daha az satır bulunur. İlk 50% kısımdan öğrendiği bölümü uygulayarak ikinci kısımdaki eksik verileri bu öğrenmeye göre tamamlamaktadır.

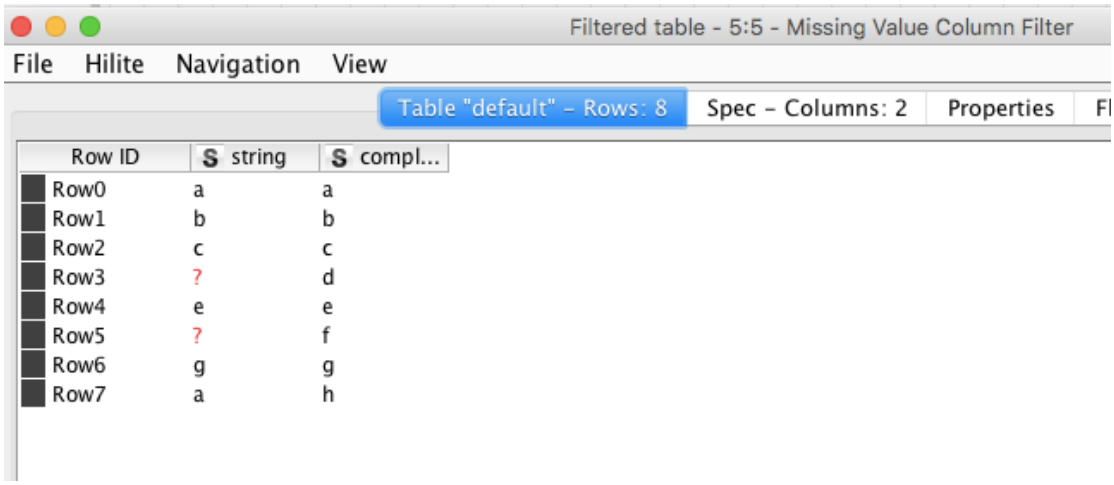
Bir diğer yöntem ise eksik verilerin kolon bazlı olarak filitrelenmesi için "Missing Value Column Filter" kullanılmasıdır.



Şekil 6.6.11

Şekil 6.6.11, sisteme missing value column filter operatörünün eklenmesini, bağlantılarını ve ayrıca bu operatörün configure penceresini göstermektedir. Configure girildiği zaman ayarlar kısmından hangi kolonlara bakılmasını isteniyorsa o kolonlar include penceresine aktarılırken diğerleri exclude penceresine aktarılır. Üç kolonlu veri seti vardı bu üç kolonun tamamına bakılsın ve eksik değerler silinsin ya da sadece isim eksikse silsin ama yaş eksikse silmesin gibi durumlar için bir tanesini hariç (exclude) tutulabilir veya sadece bir tanesi include edilebilir.

Yukarıdaki örnekte sadece stringteki eksiklleri alınmaması fakat integerdaki eksiklikleri silmesi yani integer herhangi bir eksik değer varsa bunu göstermemesi için integer ve complate kolonları alınıp string kolonu alınmamıştır. Missing value threshold kısmını %10 olarak belirlendi. Bu şekilde sistemin %10 üzerindekileri exclude etmesini beklenir.



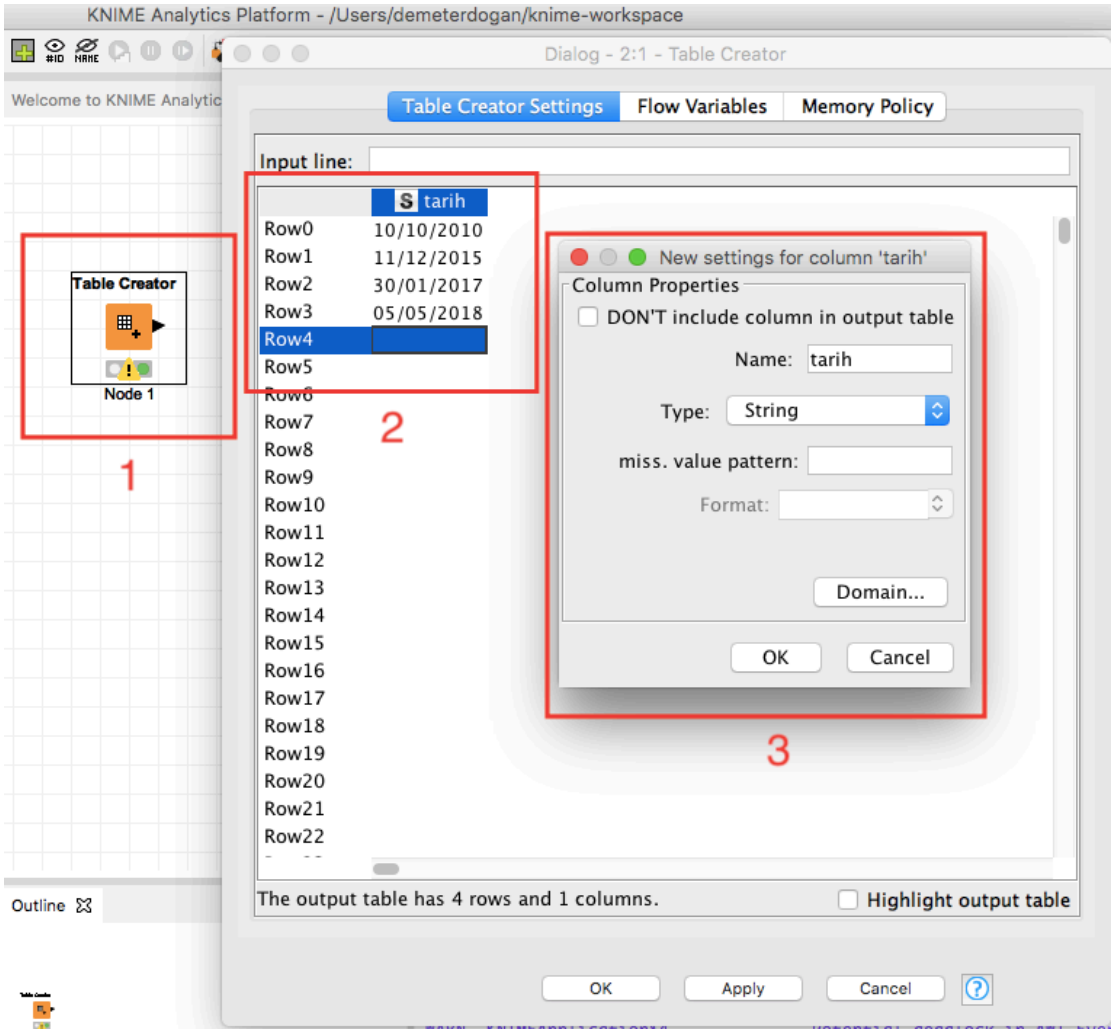
Şekil 6.6.12

Şekil 6.6.12, program çalıştırıldığında elde edilen çıktıyı göstermektedir. Complete exclude(hariç) 'u hiç eksik veri olmamasından dolayı hadil edilmedi.

Özetle bu bölümde; bir şekilde eksik değerlerin sistemin öğrenmesini sağlanaması, öğreneceği ayrı bir veri kümesinden eksik verilerle ilgili bir yöntem çıkartmasını ve sonra bunu başka bir veri kümesi üzerine uygulamasının nasıl yapılacağını tek bir veri kümesi varsa hem ordan öğrenip hem de oradaki veriler üzerinde uygulayacaksa missing value düğümünün (operatörünün) yeterli olacağı veya kolon bazlı filitreleme, eksik değerli olan kolonların tamamen kaldırılması gibi yöntemler gösterildi.

6.7 Tarih ve Zaman İşlemleri

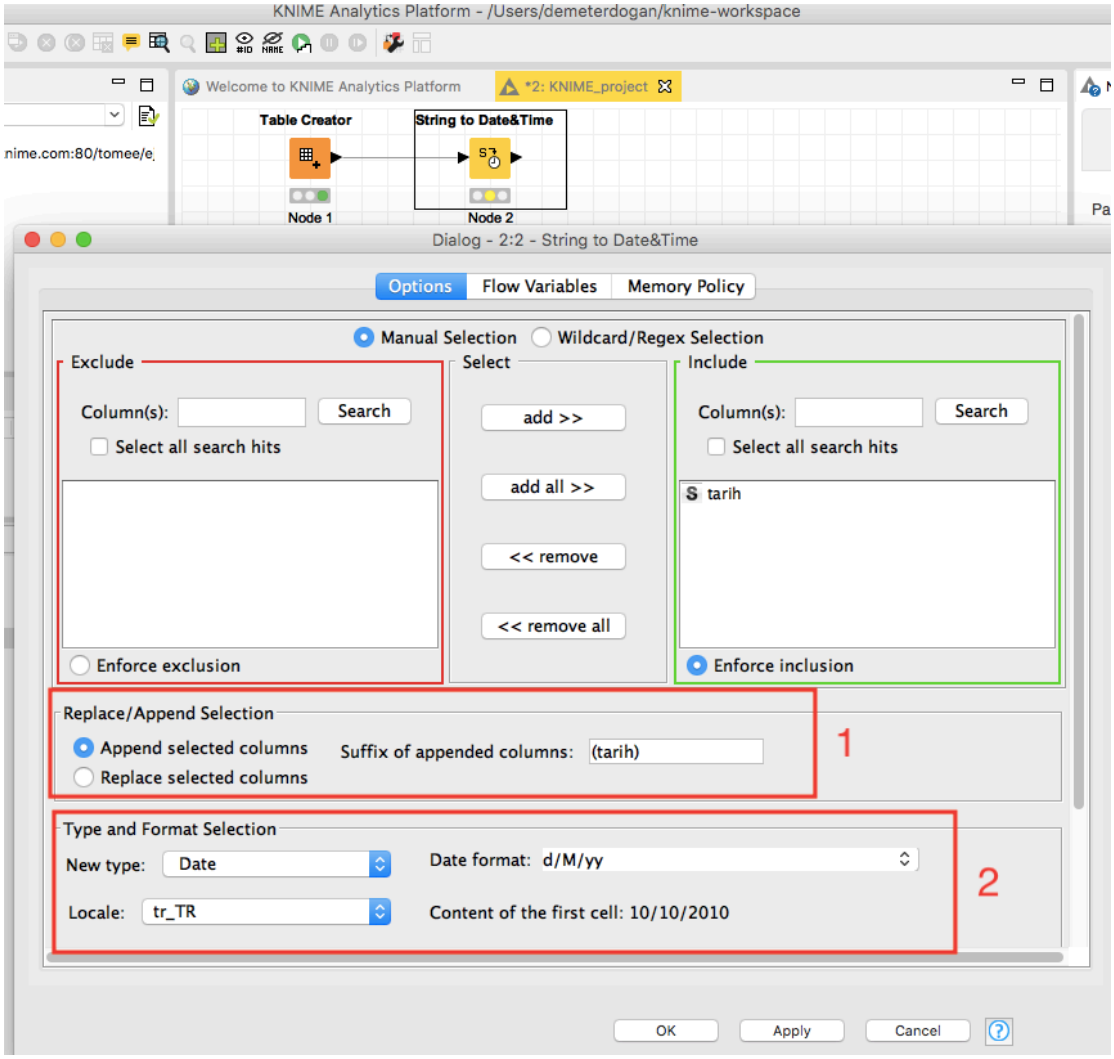
Bu bölümde tarih ve saat verileri ile ilgili çalışma yapılacaktır. Şekil 6.7.1 de görüldüğü gibi **table creator** kullanılarak manual olarak tarih verileri örnek olarak işlenecektir.



Şekil 6.7.1

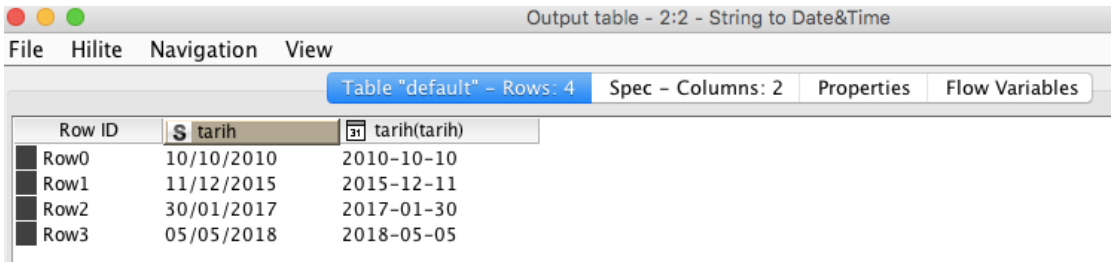
Şekil 6.7.1'de görülen 1. Penceredeki operatör configure edileceğinde 2. Pencere açılmaktadır. 2. Penceredeki tarih yazan kısıma sağ tuklandığında 3. Pencere açılmaktadır. **Name** kısmına tarih yazıldığı için 2. Pencerede görülen tarih kolonu o ismi almış ve string seçili olduğu için de türü string görünmektedir.

Bu aşamadan sonra girilen tarihin Knime dilinde tarih olarak algılanması için **string to date &time** operatörü kullanılacaktır.



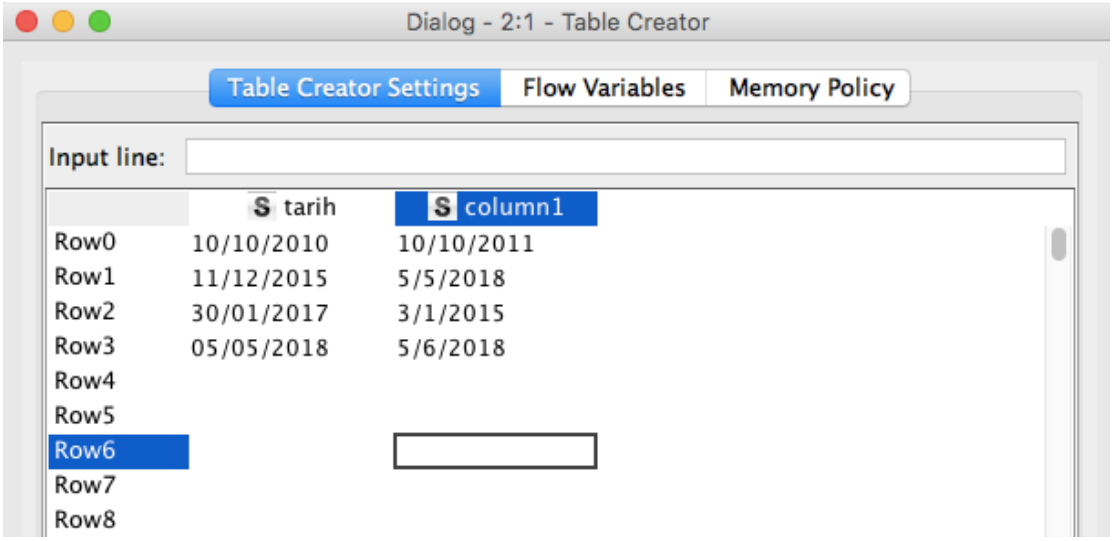
Şekil 6.7.2

Şekil 6.7.2'de görüldüğü gibi string to date&time operatörün içinde 1. Penceredeki gibi **append selected columns** seçilerek **suffix of appended columns** bölümüne tarih yazılmıştır. Ayrıca 2. Penceredeki bölümde de new type bölümünden date seçilip locale de ise **tr_TR** seçilmiştir. Date format olarak ise d/M/yy yazılmıştır. Burada d→day yani günü, M→ month yani ayı, yy ise year yeni yılı belirtmektedir. dd/MM/yyyy seklinde de yazılabilir ama bu durumda 5. Ay gibi tek haneleri yazılmak istendiğinde hata vereceği için başına 0 koymak yani 05 yazmak gerekmektedir. Bu şekilde çalıştırıldığında hata verecektir çünkü örnek verilen tarihlerde yıl hep 4 haneli işlenmiştir bu yüzden **d/M/yyyy** şeklinde yazılmalıdır.



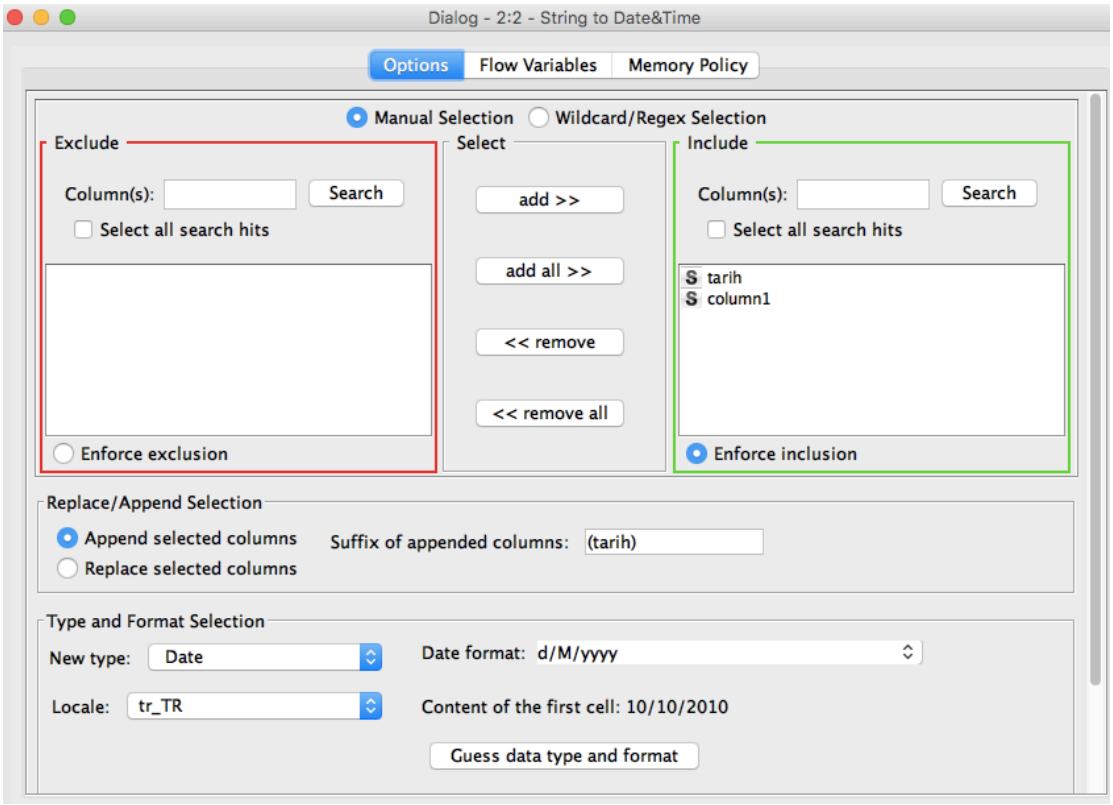
Şekil 6.7.3

Şekil 6.7.3 program çalıştırıldıktan sonra elde edilen output table'ı göstermektedir. Tarihler arasında fark almak hesap yapmak gibi avantajlar sağlamaktadır. Date&time difference operatörü ekleyerek hesap yapılabilir. Şekil 7.1.4 sisteme girme (üye olma) ve çıkma tarihleri arasındaki gün bazlı farkın hesaplanması için örnek göstermektedir.



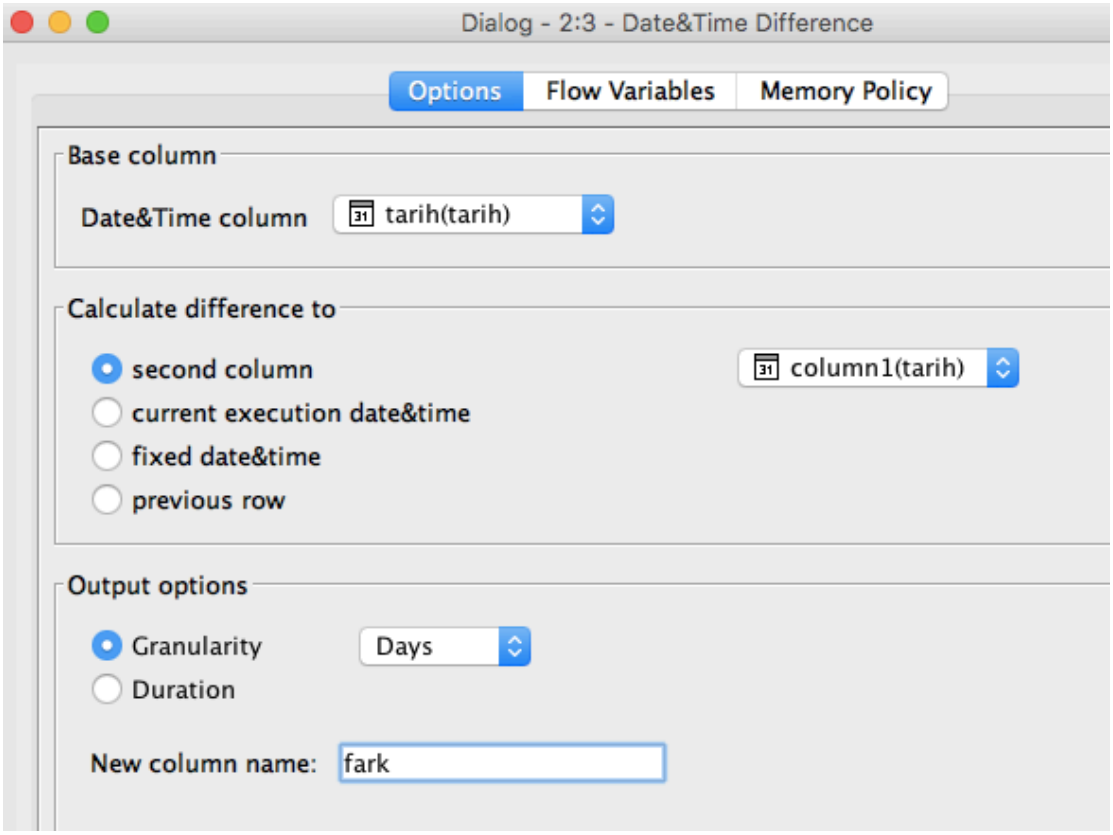
Şekil 6.7.4

Şekil 6.7.4 table creator operatöründe sistemden ayrılanların tarihleri için oluşturulan column1 kolonunu göstermektedir.



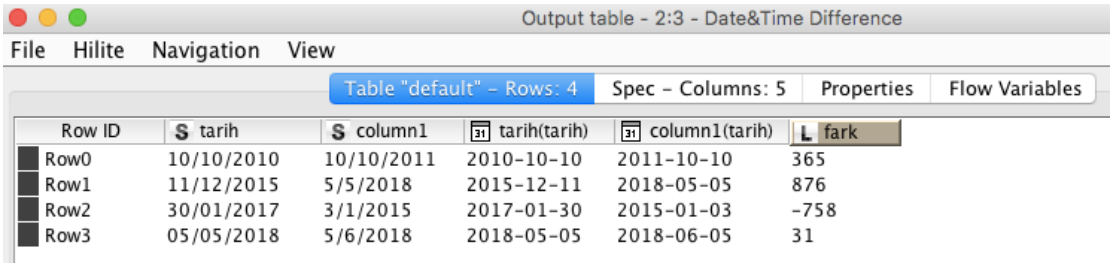
Şekil 6.7.5

Şekil 6.7.5 string to date&time operatöründeki column1'in de include bölüme aktarılmasını göstermektedir.



Şekil 6.7.6

Şekil 6.7.6 date&time difference operatöründe tarihler arasında fark bulunabilmesi için yapılan değişiklikleri göstermektedir. Tarih(tarih) yazan temel alınan kolonu yani sisteme üyelik tarhini, column1(tarih) ise sistemden çıkma tarihlerini ve fark yazan ise yeni oluşturulacak kolonun adını belirtmektedir.

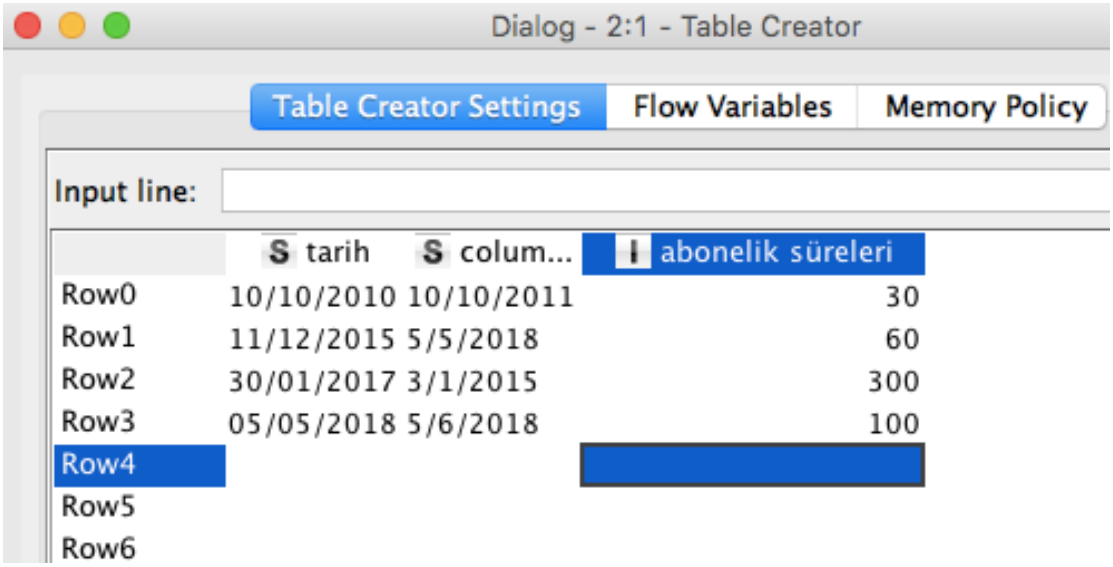


Row ID	tarih	column1	tarih(tarih)	column1(tarih)	fark
Row0	10/10/2010	10/10/2011	2010-10-10	2011-10-10	365
Row1	11/12/2015	5/5/2018	2015-12-11	2018-05-05	876
Row2	30/01/2017	3/1/2015	2017-01-30	2015-01-03	-758
Row3	05/05/2018	5/6/2018	2018-05-05	2018-06-05	31

Şekil 6.7.7

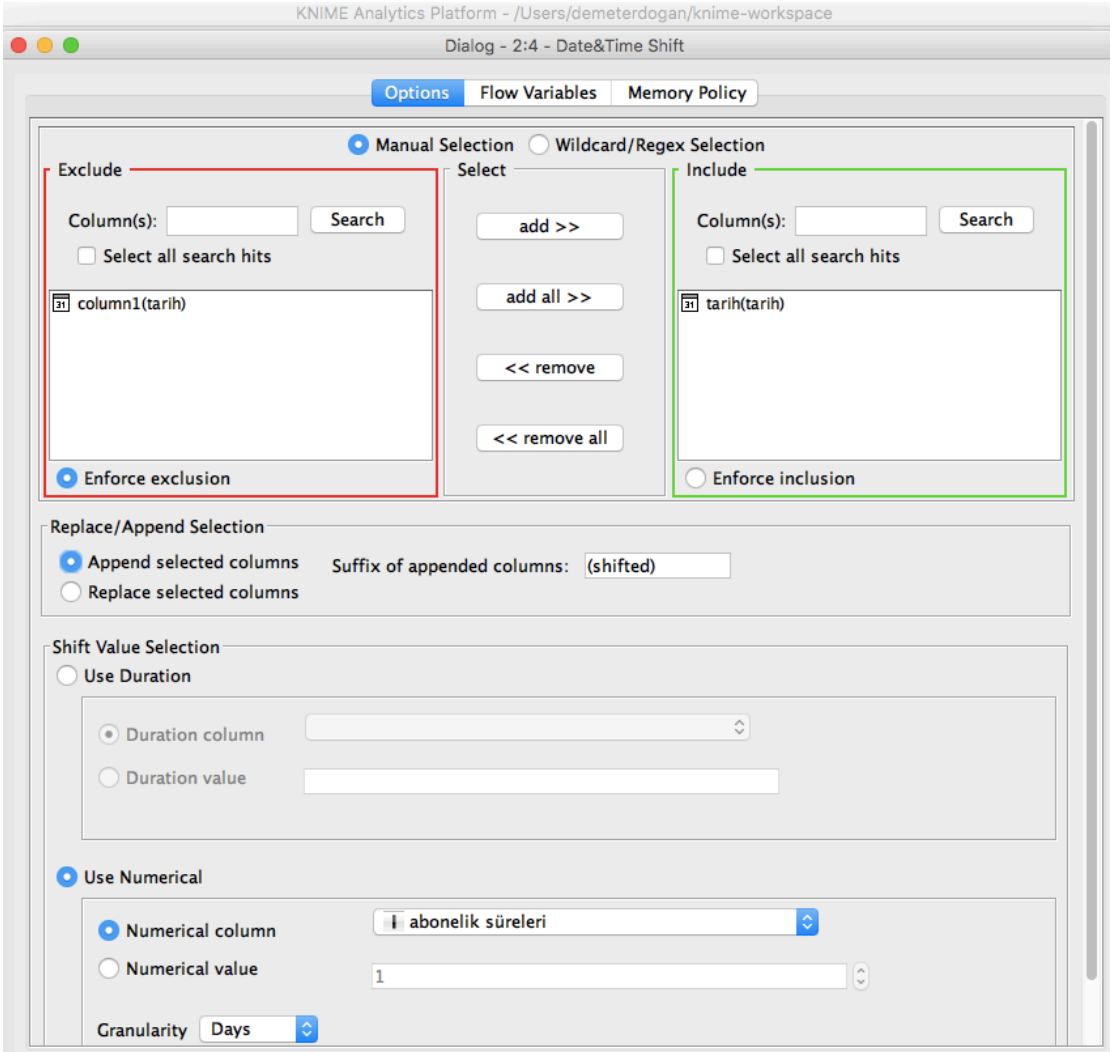
Şekil 6.7.7 sistem çalıştırıldıktan sonraki output table'ı göstermektedir. Burada Row2 yazan satırda eksi değer kirli veri için örnek verilebilir. Şekil 6.7.4'te de görüldüğü gibi sistemden çıkma tarihi sisteme üyelik tarihinden örnek olması açısından erken yazılmıştır. Bu gibi durumlarda verilerin temizlenmesi/filtrelenmesi açısından faydalı olacaktır.

Tarihlerde biraz geciktirme/ ileri/geri alma gibi durumları date&time shift operatörü ile yapılmaktadır.



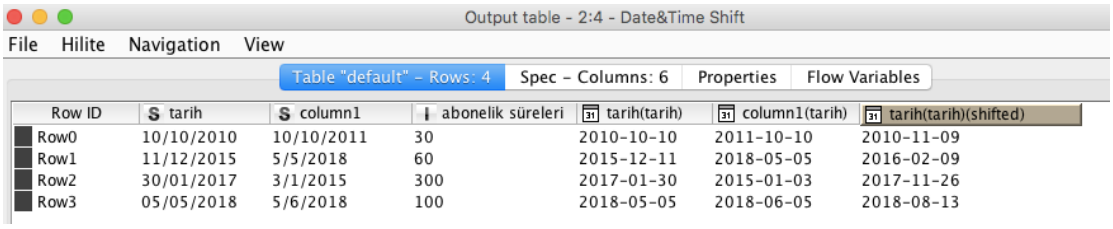
Şekil 6.7.8

Şekil 6.7.8 table creator de abonelik süreleri ismiyle integer olarak manuel eklenen müşterilerin sistemde kalma sürelerini göstermektedir.



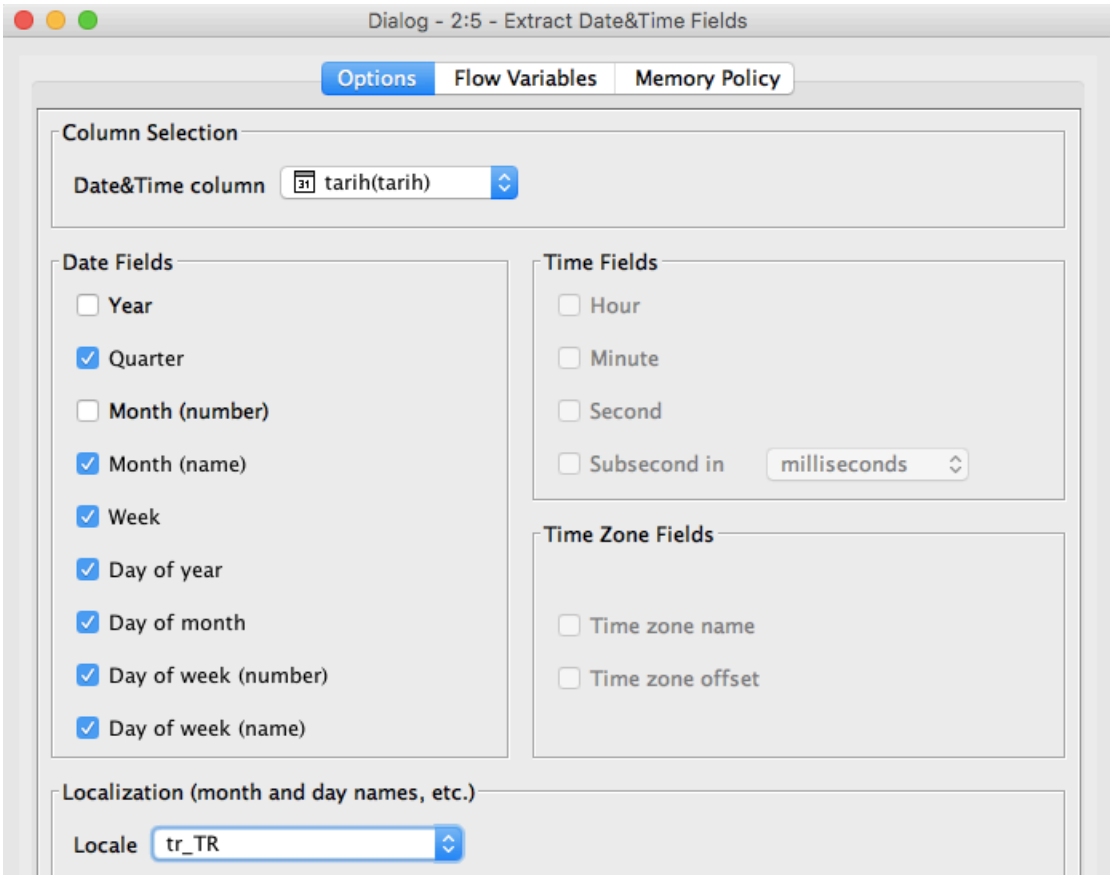
Şekil 6.7.9

Şekil 6.7.9 tarih kolonundaki yani sisteme üye olanların üyeliklerinin bitiş tarihlerinin gösterilebilmesi için verilen gün kadar kaydırmanın (shift) yapılabilmesi için date&time shift operatöründeki yapılan değişiklikleri göstermektedir.



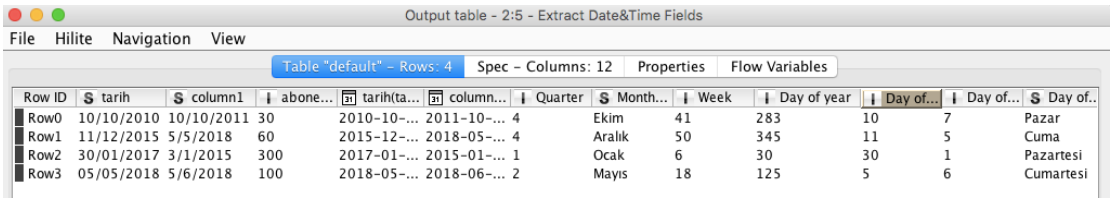
Şekil 6.7.10

Şekil 7.10 program çalıştırıldıktan sonra sisteme üye olanların abonelik sürelerine göre sistemden ayrılma tarihlerini tarih(tarih)(shifted) olarak sonucu göstermektedir.



Şekil 6.7.11

Şekil 6.7.11 verilen tarihlerin detaylarının görünmesini sağlayacak operatör configure'ünü göstermektedir. Örneğin burada sisteme giriş tarihinin yılın hangi haftası, ayı, günü gibi detayların çıkması istenmiştir.



Row ID	tarih	column1	abone...	tarih(ta...	column...	Quarter	Month...	Week	Day of year	Day of...	Day of...	Day of..
Row0	10/10/2010	10/10/2011	30	2010-10-...	2011-10-...	4	Ekim	41	283	10	7	Pazar
Row1	11/12/2015	5/5/2018	60	2015-12-...	2018-05-...	4	Aralık	50	345	11	5	Cuma
Row2	30/01/2017	3/1/2015	300	2017-01-...	2015-01-...	1	Ocak	6	30	30	1	Pazartesi
Row3	05/05/2018	5/6/2018	100	2018-05-...	2018-06-...	2	Mayıs	18	125	5	6	Cumartesi

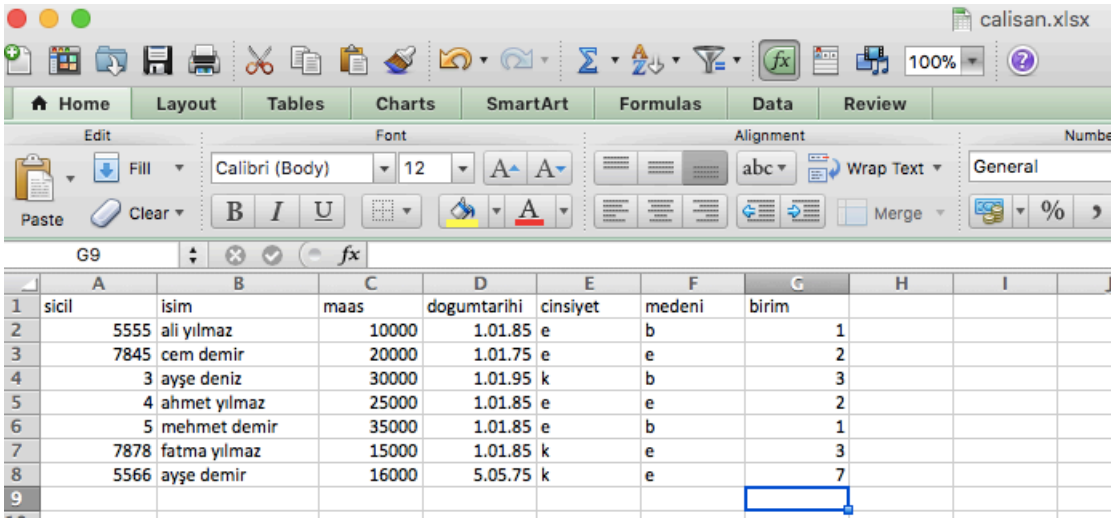
Şekil 6.7.12

Şekil 7.12 sistem çalıştırıldıktan sonraki output table'ı göstermektedir. Örneğin 10/10/2010 da üye olan kişi yılın 41. Haftasında, 283. Gününde, Pazar günü sisteme üye olmuştur.

Bu operatör örneğin bir restoranın haftanın hangi günü daha çok yoğun olduğu, daha çok ciro yaptığı gibi detayların anlaşılabilmesi için yaygın kullanılan bir operatördür.

6.8 GROUP VE JOIN UYGULAMASI

Bu bölümde amaç join ve group by uygulaması yapmak ve veri ön işleme, ETL sürecindeki işlemleri göstermektir.

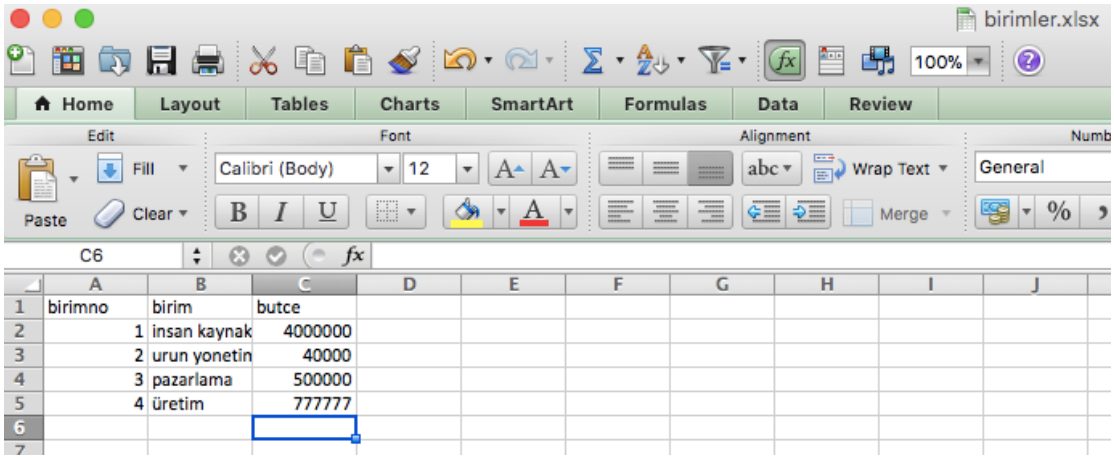


calisan.xlsx

sicil	isim	maas	dogumtarihi	cinsiyet	medeni	birim
5555	ali yılmaz	10000	1.01.85	e	b	1
7845	cem demir	20000	1.01.75	e	e	2
3	ayşe deniz	30000	1.01.95	k	b	3
4	ahmet yılmaz	25000	1.01.85	e	e	2
5	mehmet demir	35000	1.01.85	e	b	1
7878	fatma yılmaz	15000	1.01.85	k	e	3
5566	ayşe demir	16000	5.05.75	k	e	7

Şekil 6.8.1

Şekil 6.8.1 bu bölümde kullanılacak çalışanların bilgilerini içeren excel dosyasını göstermektedir.



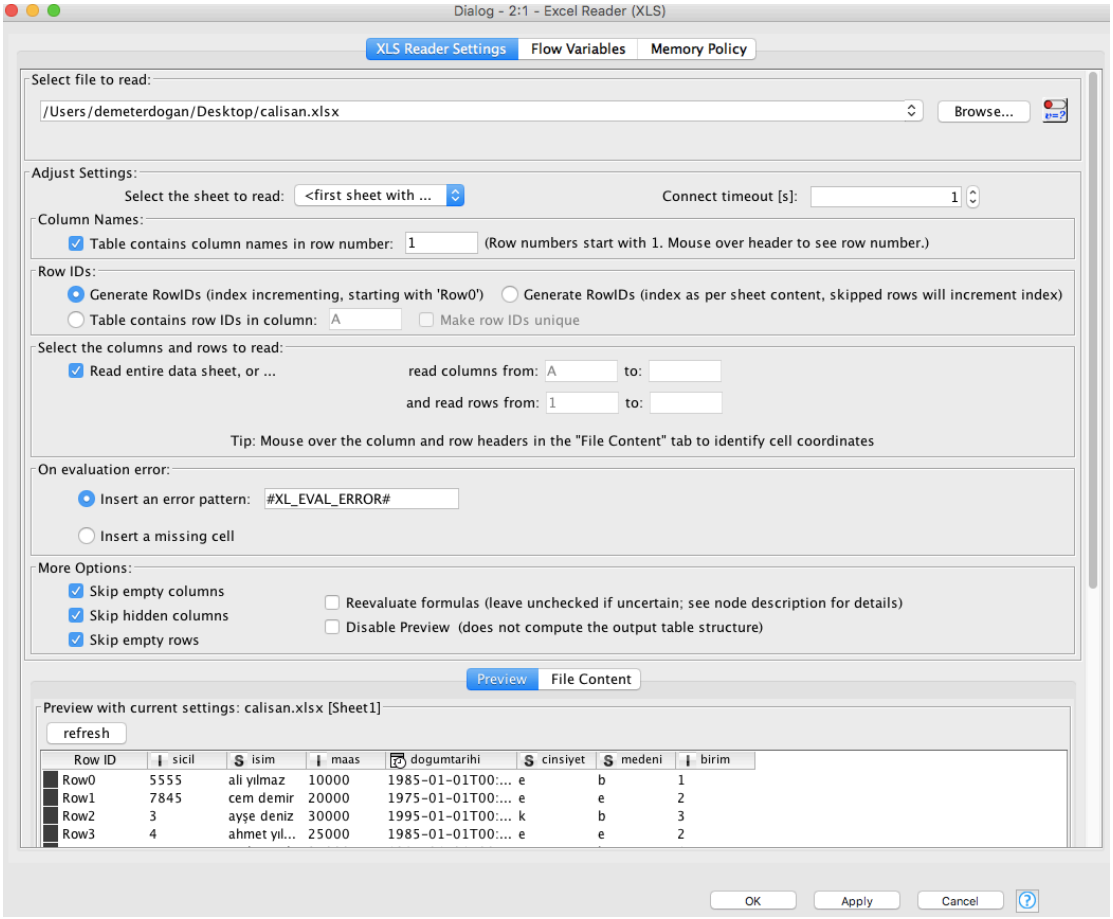
birimler.xlsx

birimno	birim	butce
1	insan kaynak	4000000
2	urun yonetin	40000
3	pazarlama	500000
4	üretim	777777

Şekil 6.8.2

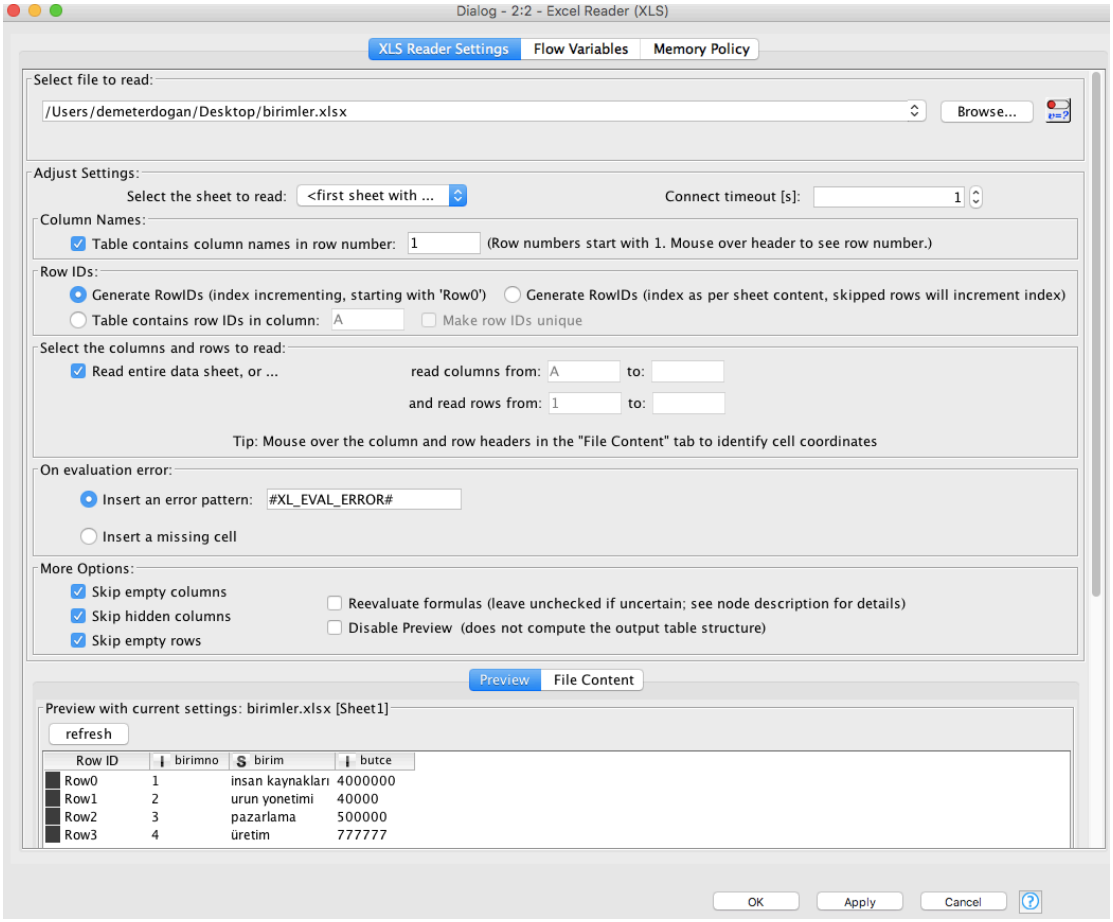
Şekil 6.8.2 bu bölümde kullanılacak departman isimlerini, birim no larını ve bu departmanların sahip oldukları bütçeleri göstermektedir. Bu iki excel dosyası oluşturulup sonrasında aşağıdaki uygulamalar yapılmalıdır.

Excel reader'lar ile bu excel dosyaları knime a aktarılır.



Şekil 6.8.3

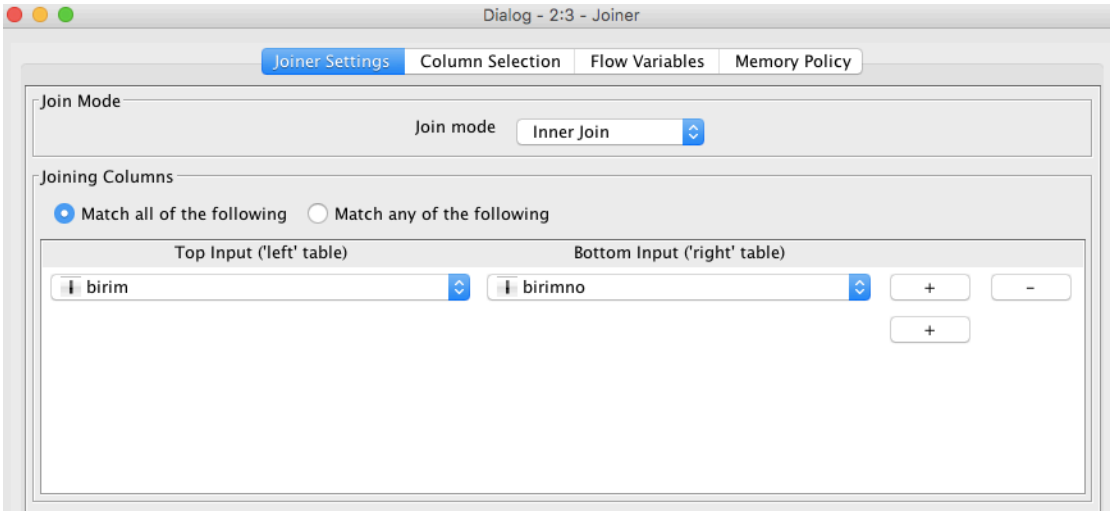
Şekil 6.8.3 çalışan bilgilerini içeren excel'in birinci excel reader ile sisteme tanıtılmasını göstermektedir.



Şekil 6.8.4

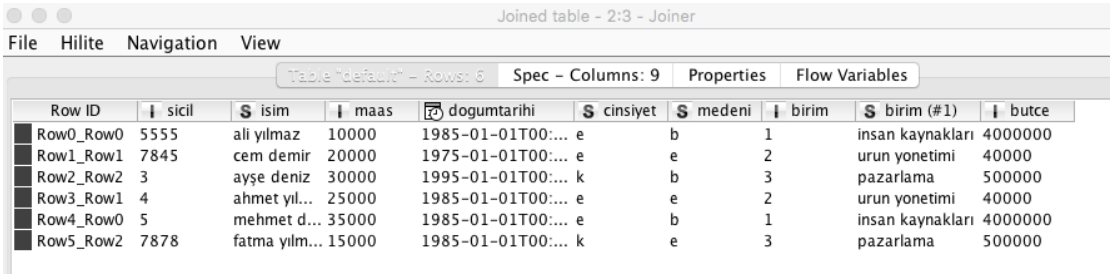
Şekil 6.8.4 birimlerin bilgilerini içeren excel'in ikinci excel reader ile sisteme tanıtılmasını göstermektedir.

Şimdi amaç departmanda en yüksek maaşlı çalışan kişilerin isimleriyle birlikte listelenmesi. Bunun için öncelikle joiner ekleyerek Şekil 6.8.5 deki gibi ayarlanmalıdır.



Şekil 6.8.5

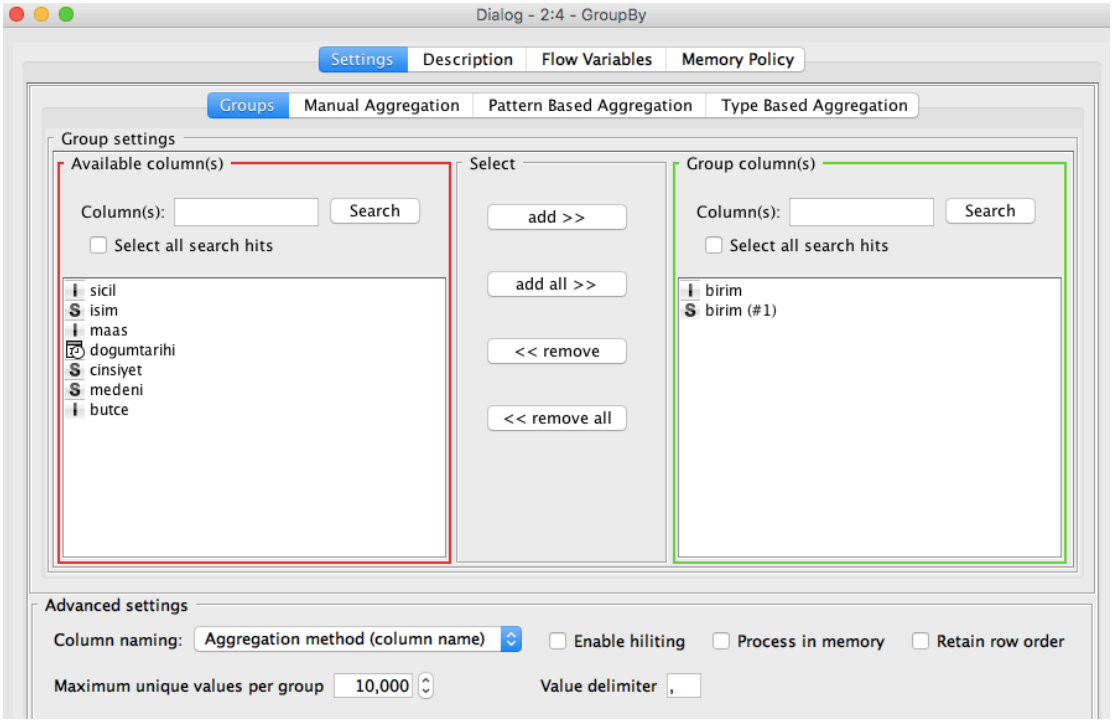
Şekil 6.8.5'te görüldüğü için iki excelde de olan bilgilerin alınabilmesi için inner join seçilmelidir. Diğer join türleri yukarıdaki bölümlerde açıklamıştır ve istenilen başka bir join de kullanılabilir.



Row ID	sicil	isim	maas	dogumtarihi	cinsiyet	medeni	birim	birim (#1)	butce
Row0_Row0	5555	ali yılmaz	10000	1985-01-01T00:...	e	b	1	insan kaynakları	4000000
Row1_Row1	7845	cem demir	20000	1975-01-01T00:...	e	e	2	urun yonetimi	40000
Row2_Row2	3	ayşe deniz	30000	1995-01-01T00:...	k	b	3	pazarlama	500000
Row3_Row1	4	ahmet yıl...	25000	1985-01-01T00:...	e	e	2	urun yonetimi	40000
Row4_Row0	5	mehmet d...	35000	1985-01-01T00:...	e	b	1	insan kaynakları	4000000
Row5_Row2	7878	fatma yılm...	15000	1985-01-01T00:...	k	e	3	pazarlama	500000

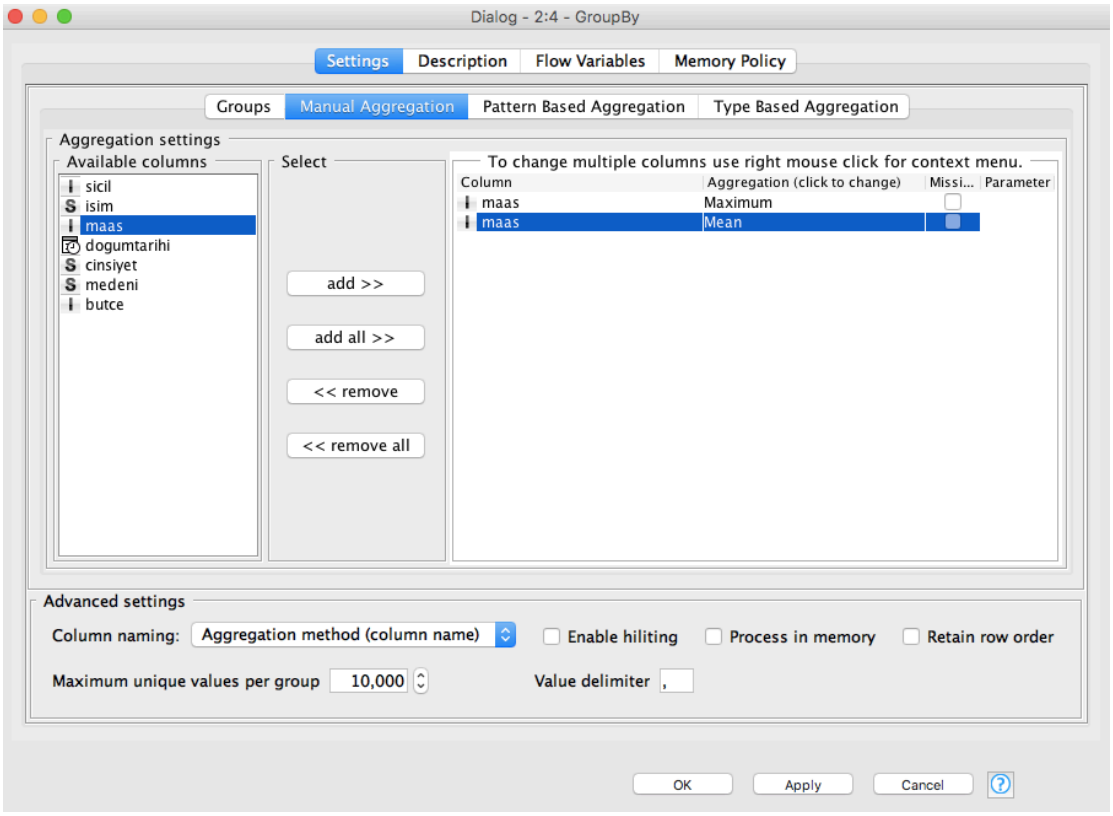
Şekil 6.8.6

Şekil 6.8.6 program çalıştırıldıktan sonra elde edilen joined table'ı göstermektedir.



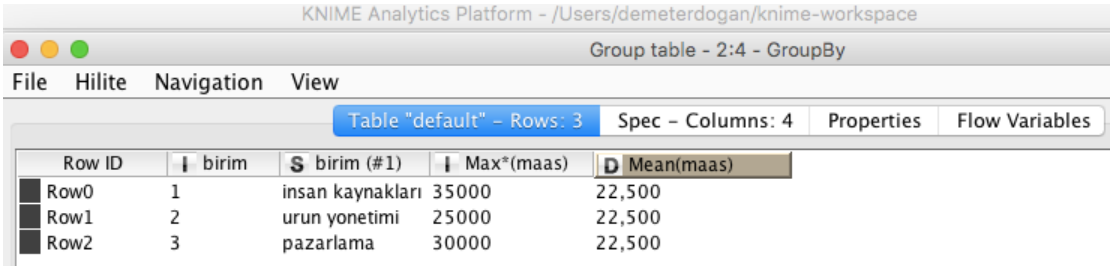
Şekil 6.8.7

Şekil 6.8.7 groupby operatörü içerisinde seçilen birim ve birim no kolonlarının gruplanmasını göstermektedir.



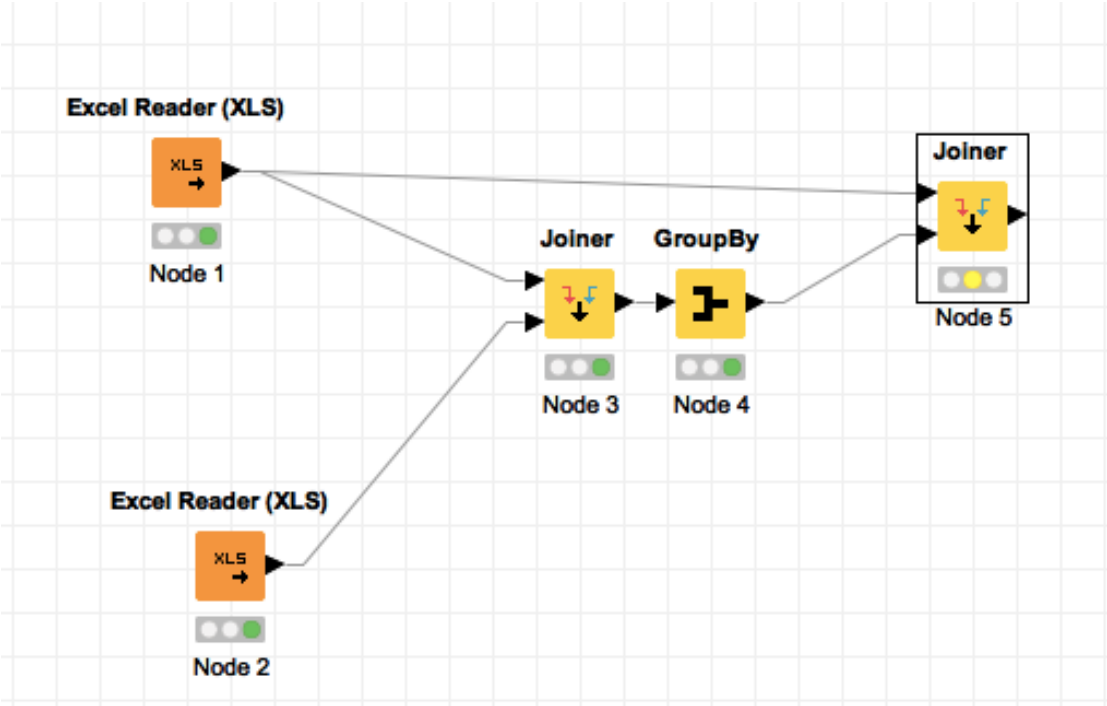
Şekil 6.8.8

Şekil 6.8.8 yine group by operatörünün içerisinde maaşların maximum değerinin ve ortalamalarının gösterilmesi için manual aggregation penceresinde yapılan ayarları göstermektedir.



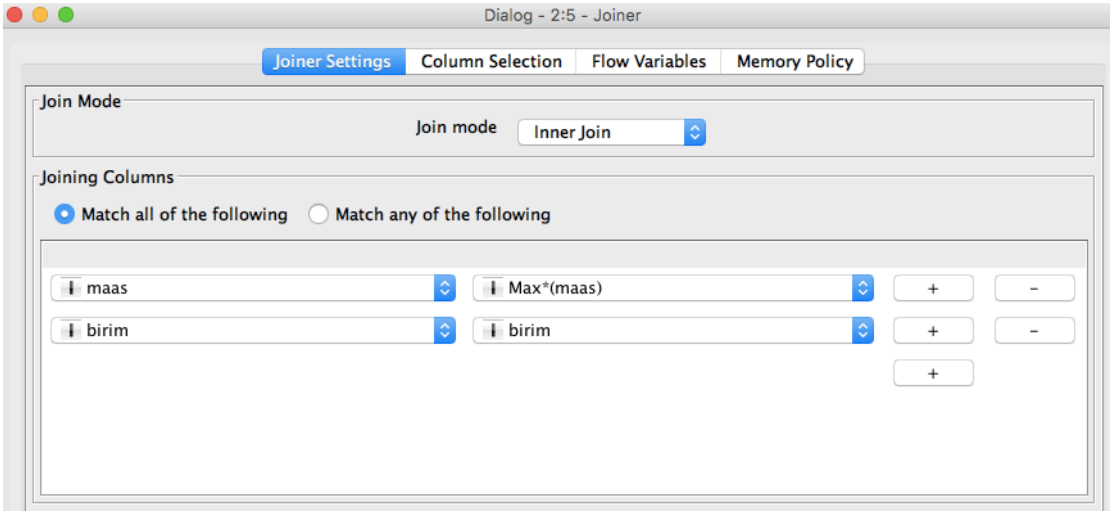
Şekil 6.8.9

Şekil 6.8.9 program çalıştırıldıktan sonra elde edilen group table'ı göstermektedir. Burada da görüldüğü gibi en yüksek maaş alan çalışan detayları henüz belli değildir. Bunun için joiner operatörü kullanılmalıdır.



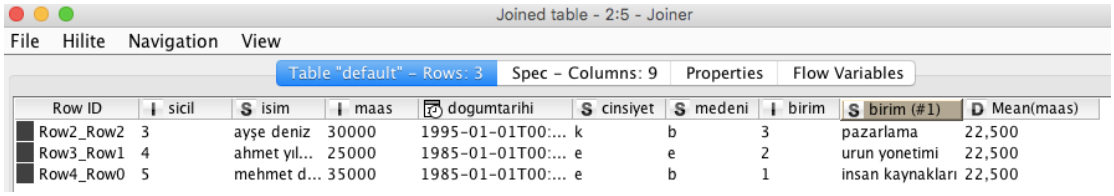
Şekil 6.8.10

Şekil 6.8.10 birimler içerisinde en yüksek maaş alan kişilerin bilgilerini getirmesi için eklenen joiner operatörünü göstermektedir.



Şekil 6.8.11

Şekil 6.8.11 joiner operatörünün nasıl configure edildiğini göstermektedir. Örneğin 35000 maaş alan kişi belki insan kaynaklarında çalışmıyordur ve bu bilgi farklı birimden getirilecektir. Bu durum olmaması için birimler ve maaş durumları seçilerek program çalıştırılmalıdır.



Row ID	sicil	isim	maas	dogumtarihi	cinsiyet	medeni	birim	birim (#1)	Mean(maas)
Row2_Row2	3	ayşe deniz	30000	1995-01-01T00:...	k	b	3	pazarlama	22,500
Row3_Row1	4	ahmet yıl...	25000	1985-01-01T00:...	e	e	2	urun yonetimi	22,500
Row4_Row0	5	mehmet d...	35000	1985-01-01T00:...	e	b	1	insan kaynakları	22,500

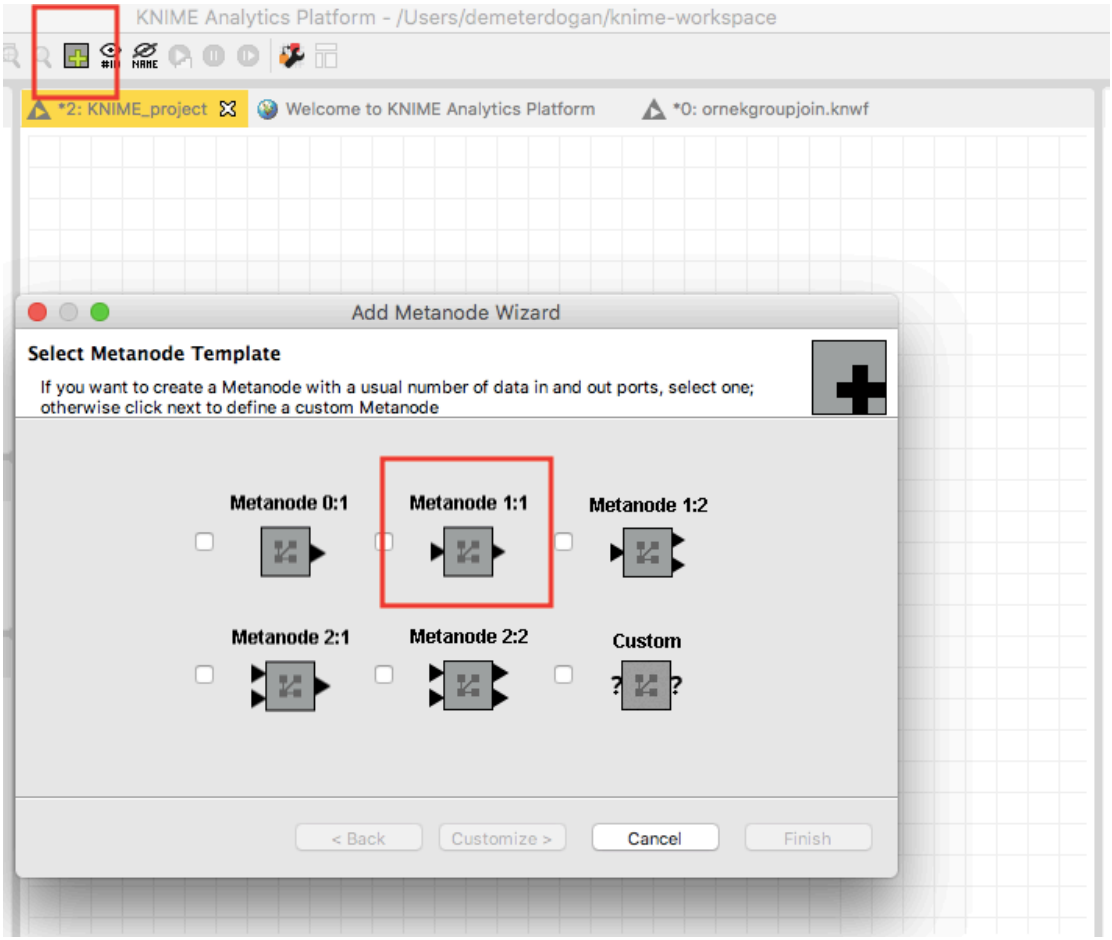
Şekil 6.8.12

Şekil 6.8.12 program çalıştırıldıktan sonra elde edilen joined table'ı göstermektedir. Sonuca göre örneğin; birim numarası 3 olan pazarlama departmanından 30000 ile en yüksek maaşı alan kişi Ayşe Deniz'dir.

7.BÖLÜM: İLERİ KNIME KULLANIMI

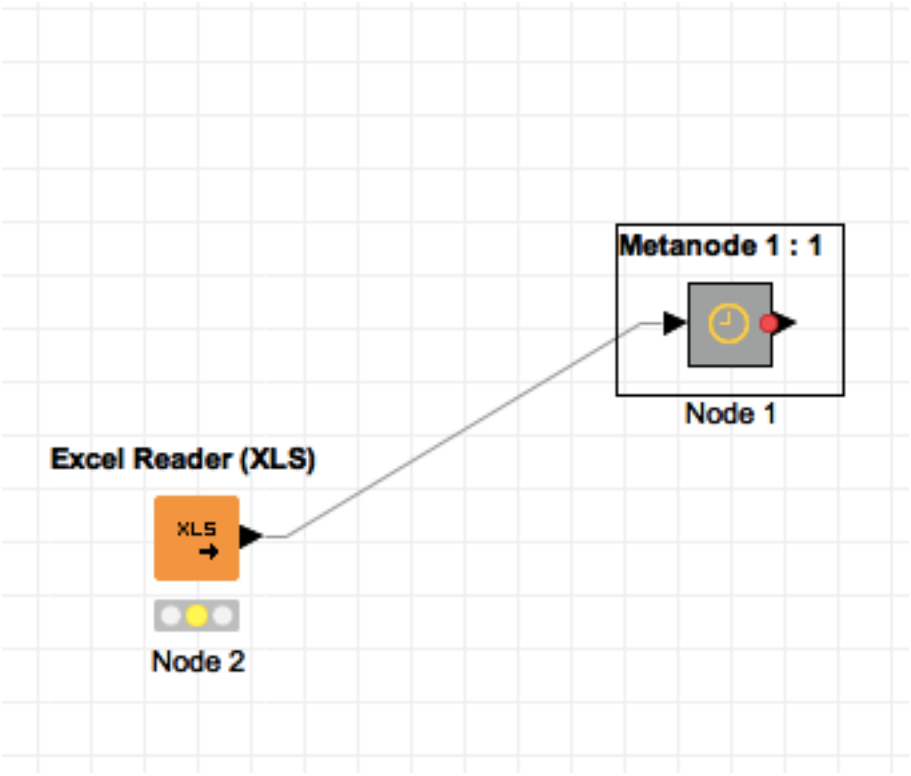
7.1 MetaNode Yapısı

Bu bölümde amaç Knime'ın ileri özelliklerinden olan MetaNode aracının kullanımı gösterilecektir. MetaNode, birden fazla node'u kapsayan ve bunları blocklayan bir araçtır.



Şekil 7.1.1

Şekil 7.1.1'de MetaNode aracının Knime penceresinin üst kısmında görülmektedir. Bu bölümde 1 input 1 output olacak şekilde MataNode 1:1 seçilerek ilerlenecektir.



Şekil 7.1.2

Şekil 7.1.2'de Excel Reader dosyası ile MetaNode aracının bağlantısı görülmektedir. Excel Reader operatörü içerisine daha önce de kullanılan cinsiyet, boy ve kilo kolonları olan excel dosyası tanıtılmıştır.

Output table - 2:2 - Excel Reader (XLS)

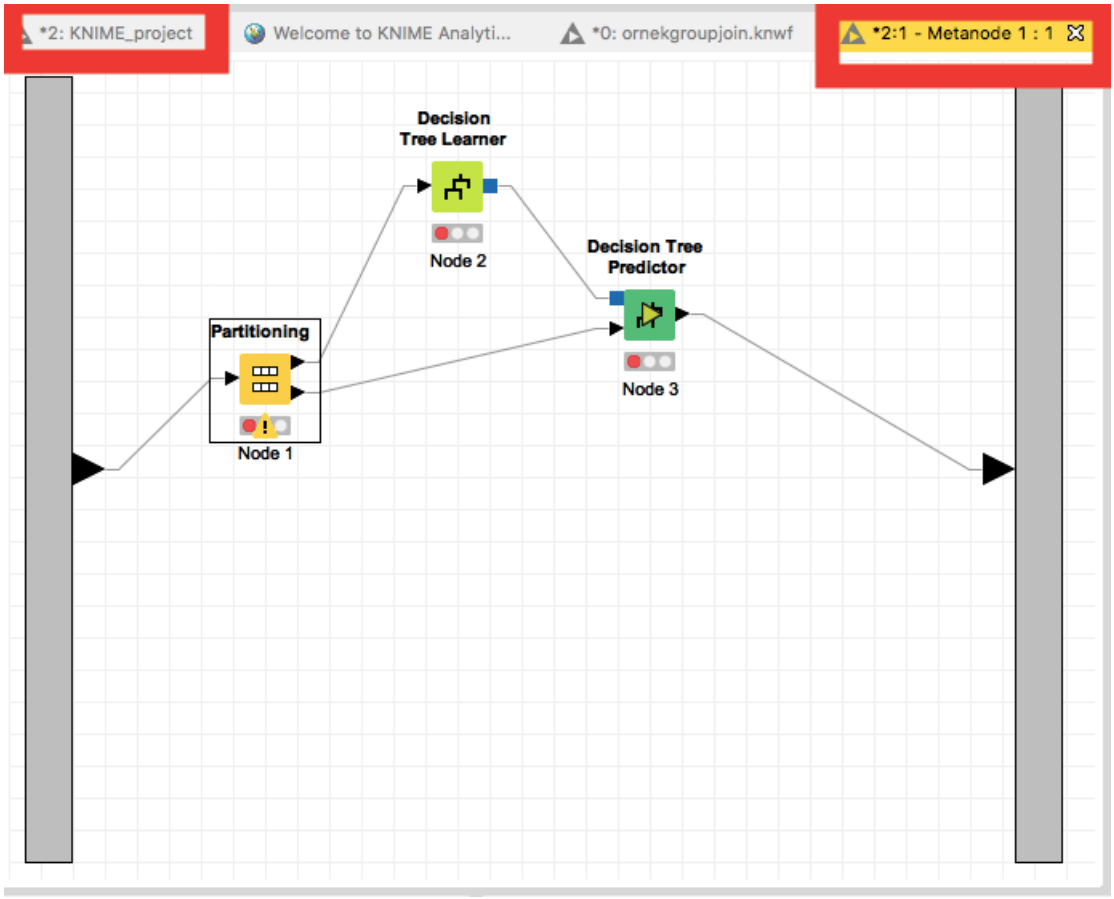
File Hilite Navigation View

Table "cinsiyet.xlsx [Sheet1]" - Rows: 10 | Spec - Columns: 3 | P

Row ID	boy	kilo	cinsiyet
Row0	185	85	erkek
Row1	174	65	kadın
Row2	180	79	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek
Row9	170	60	kadın

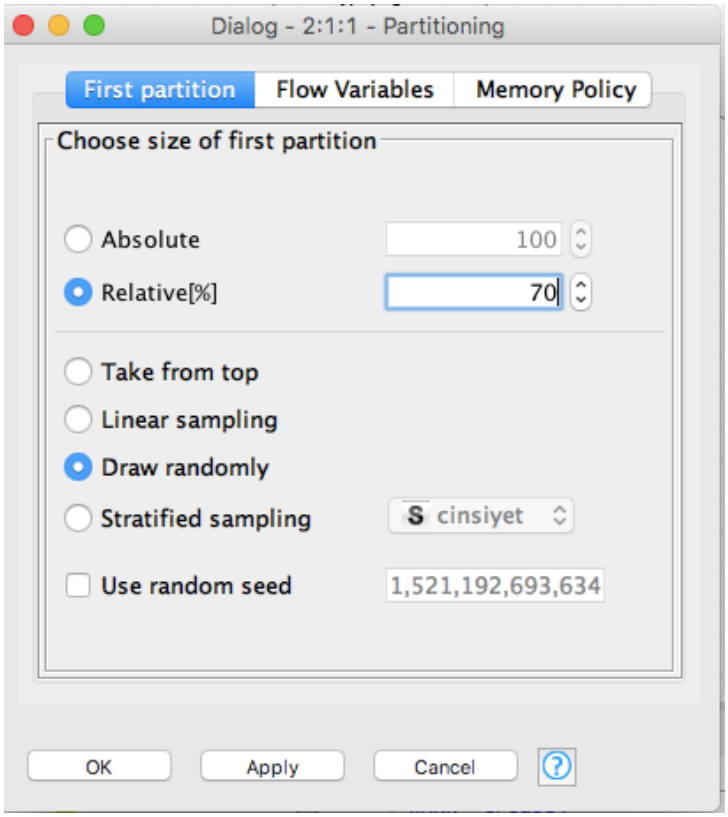
Şekil 7.1.3

Şekil 7.1.3 Excel Reader'a tanıtılan örnek excel dosya içeriğini göstermektedir.



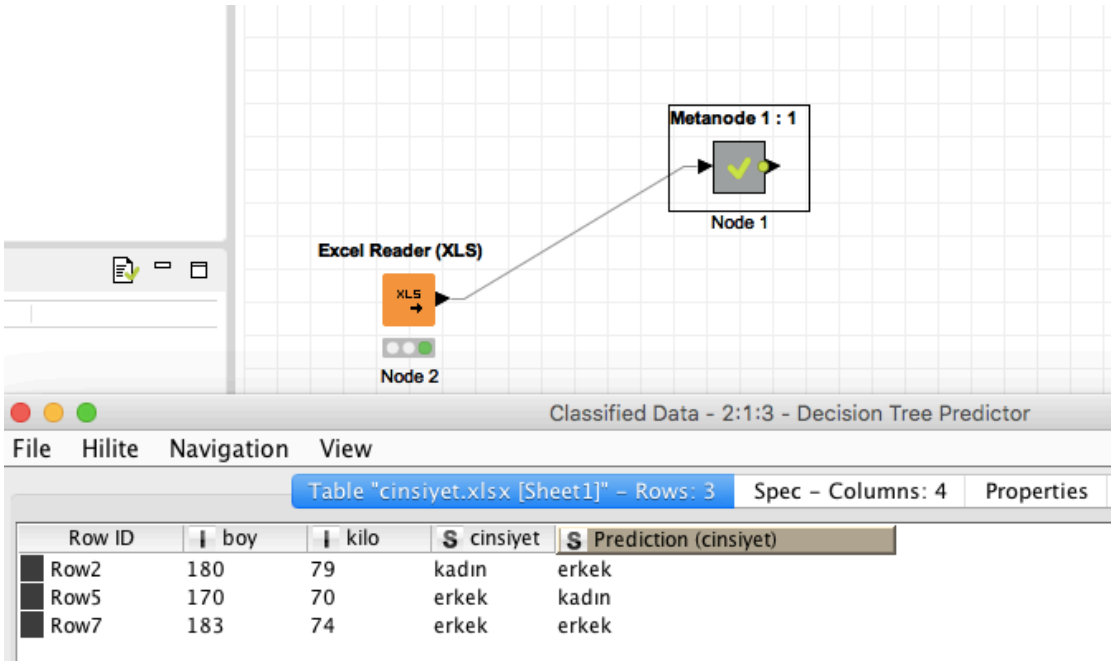
Şekil 7.1.4

Şekil 7.1.4 MetaNode'un içini göstermektedir. MetaNode'a çift tıklandığında boş bir pencere açılır. Daha sonrasında içerisine eğitim ve test için verinin bölünmesini sağlayan partitioning ve öğrenmesi için decision tree learner ayrıca test için decision tree predictor operatörleri eklendi. MetaNode içerisinden çıkmak için şekilde sol üstte görülen kırmızı penceredeki üzerinde çalışılan Knime-project penceresine tıklanmalıdır. Sağ kısımda görülen MetaNode 1:1 penceresi, metanode içerisini gösteren sayfadır.



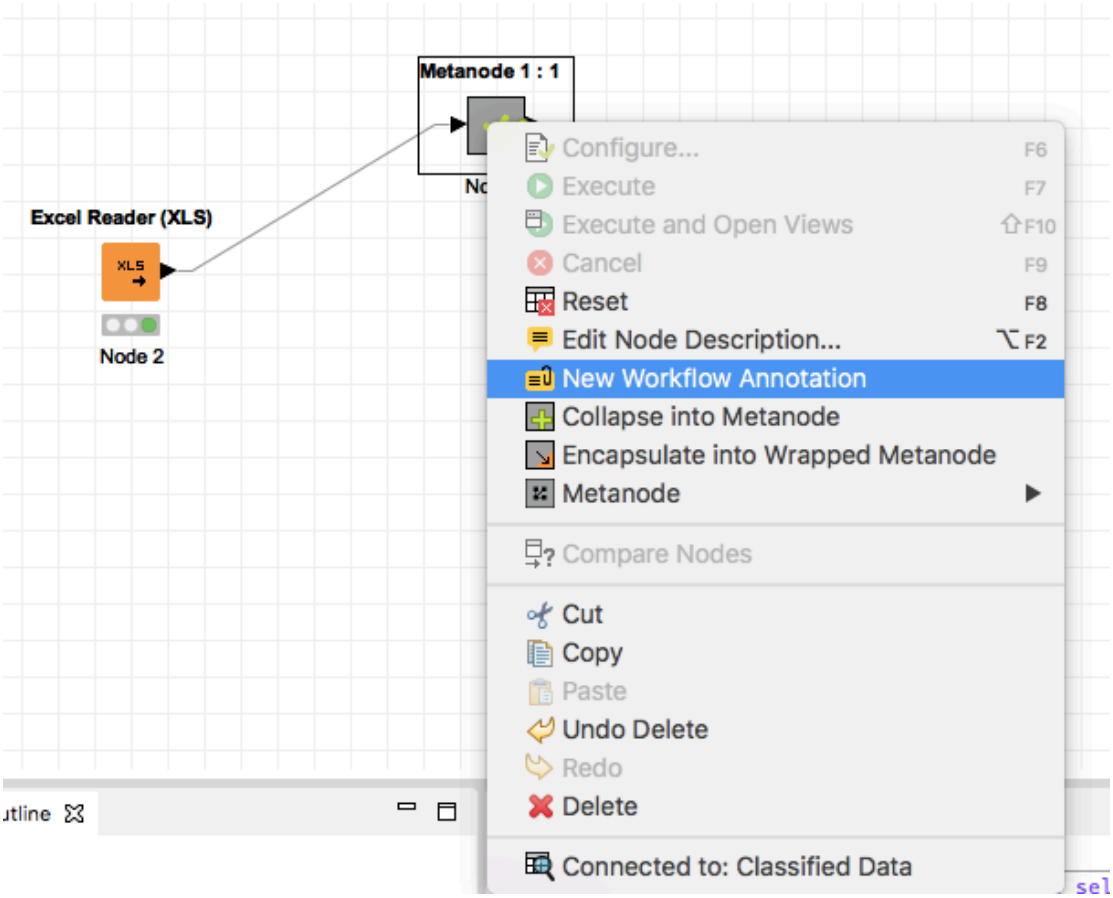
Şekil 7.1.5

Şekil 7.1.5, partitioning operatörü içerisinde verinin hangi oranda bölündüğünü göstermektedir. Burada veri 70% ve 30% oranlarında bölünmüştür.



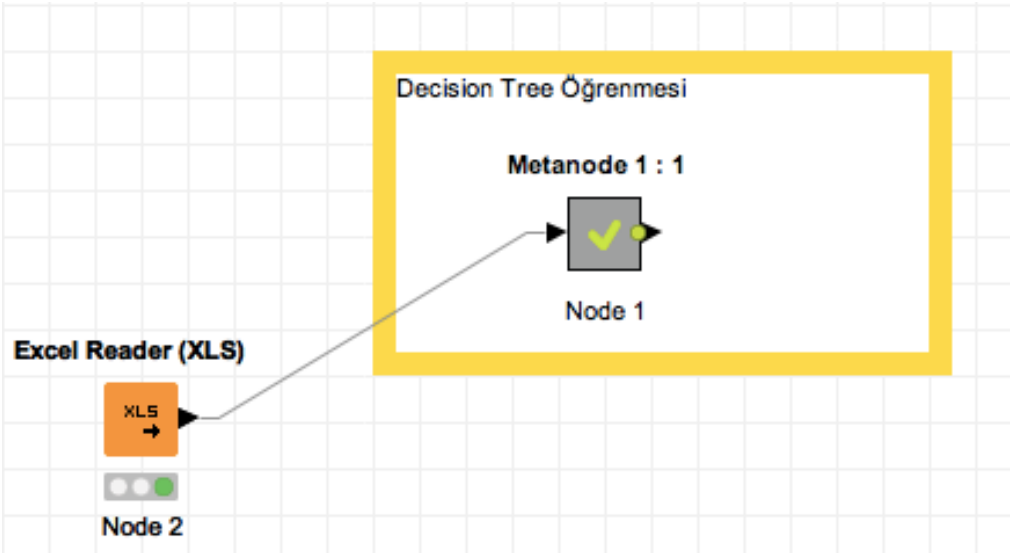
Şekil 7.1.6

Şekil 7.1.6, Knime genel penceresini göstermektedir. Bir önceki şekilden çıkılıp ana pencereye dönülmelidir. Program çalıştırıldığında output table'dan sonuca ulaşılabilir. Şekilde de görülen decision tree predictor penceresi sonucu göstermektedir.



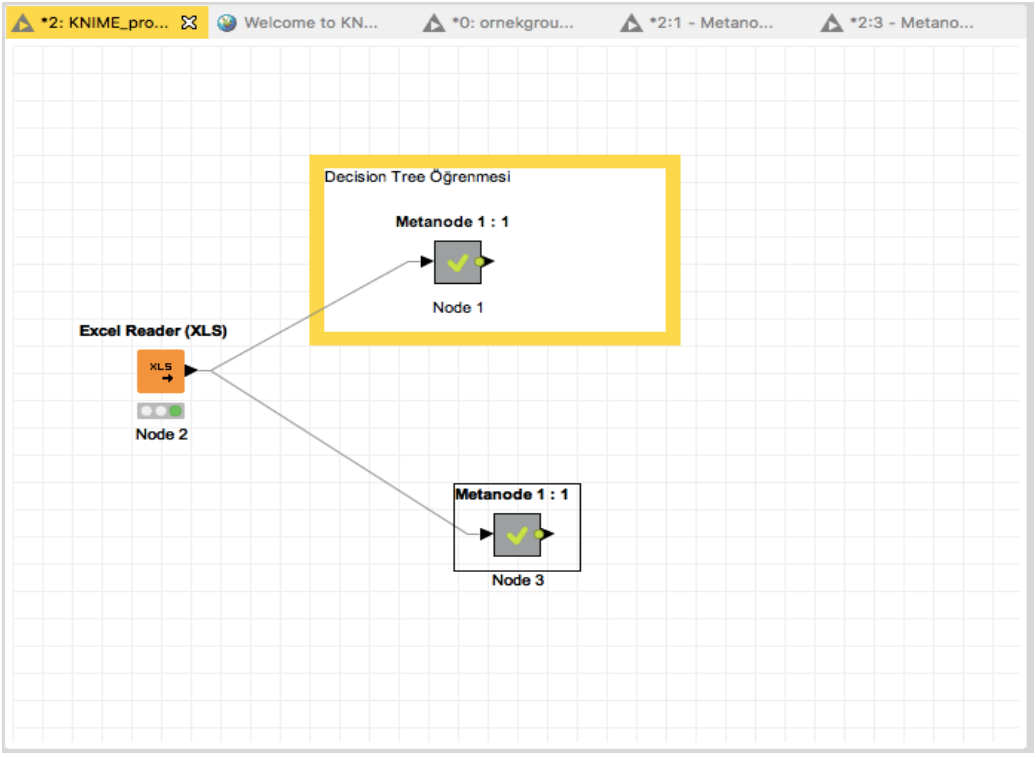
Şekil 7.1.7

Şekil 7.1.7 MetaNode'a sağ tıklanarak penceresi üzerine nasıl not eklenebileceğini (annotation) göstermektedir.



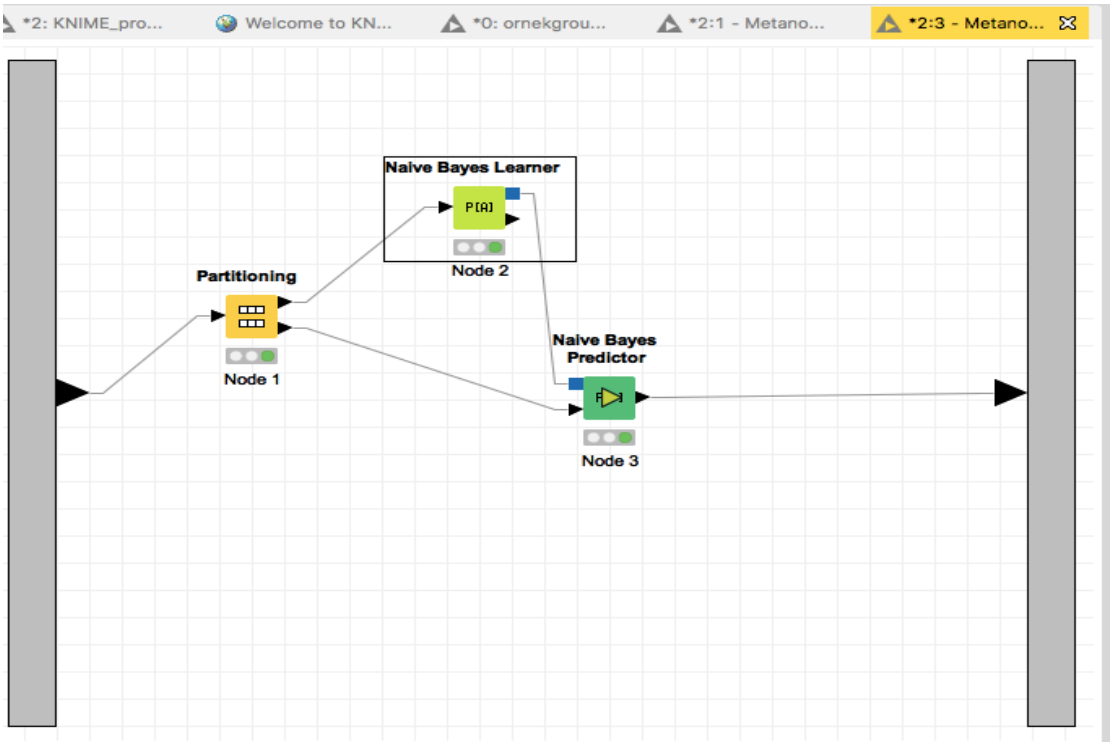
Şekil 7.1.8

Şekil 7.1.8, not eklendikten sonraki halini göstermektedir.



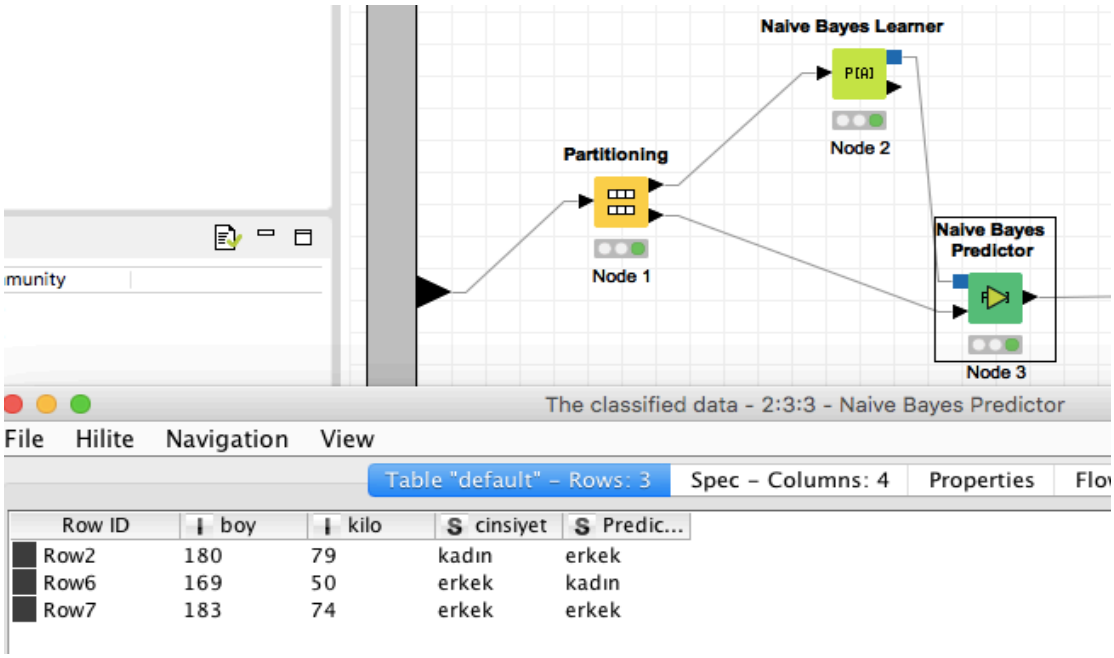
Şekil 7.1.9

Şekil 7.1.9, 1. MetaNode'un kopyalanıp aşağı yapıştırılmış halini göstermektedir. Kopyalama işlemi içeriğinin de kopyalanmasını sağlamıştır.



Şekil 7.1.10

Şekil 7.1.10, yeni oluşturulan MetaNode içeriğini göstermektedir. Burada ilk kopyalama yapıldığında Decision Tree Learner ve Decision Tree Predictor operatörleri vardı fakat onlar silinip Naive Bayes Learner ve Naive Bayes Predictor eklendir. Partitioning hiç değiştirilmedi. Bu şekilde program çalıştırıldığında Şekil 7.1.11'deki gibi bir sonuç elde edilir.



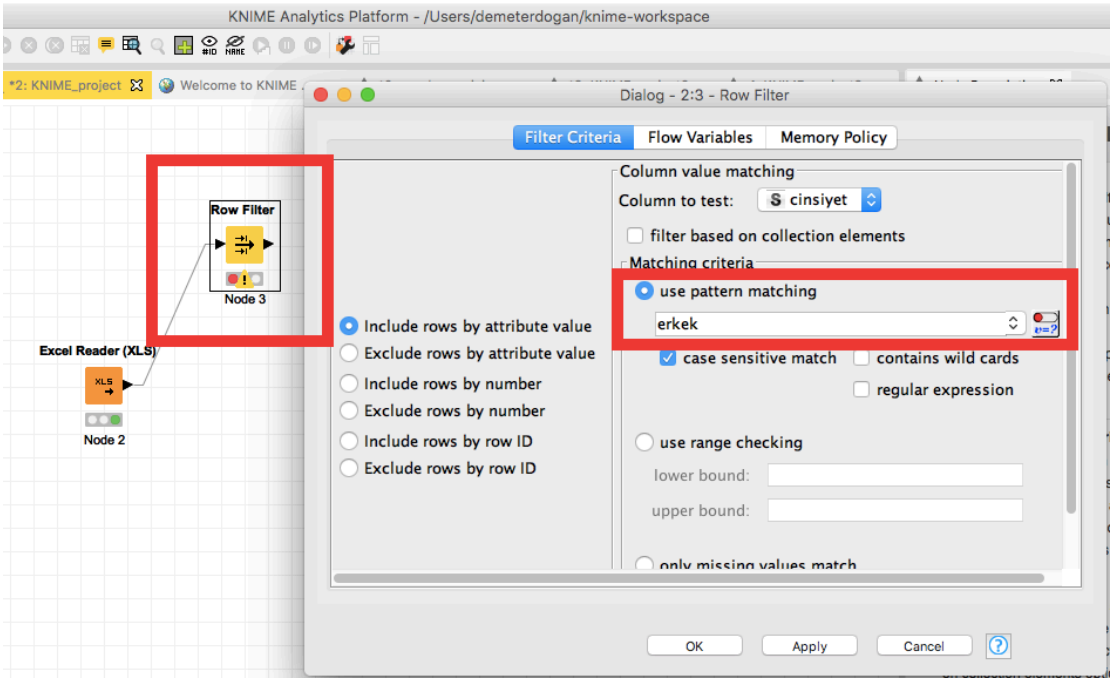
Şekil 7.1.11

Şekil 7.1.11 Naive Bayes ile yapılan işlemin sonucunu göstermektedir.

MetaNode üst düğüm olduğu için alt düğümlerin toplanmasında yani operatörlerin kapsanmasında oldukça yardımcı olmaktadır. Bu sadece çok karmaşık projelerde bile kolay ulaşıp anlaşılabilir bir görüntü elde edilebilmektedir.

7.2 Knime Değişkenleri Ve Değişken Akışı (Flow Variables)

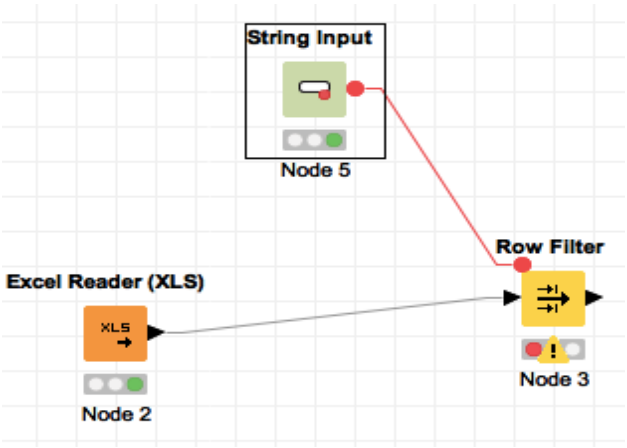
Bu bölümde knime üzerinden akış değişkenlerinin nasıl kullanılacağı gösterilecektir. Daha önceki bölümlerde kullanılan boy, kilo ve cinsiyet kolonlarını içeren excel veri seti kullanılacaktır.



Şekil 7.2.1

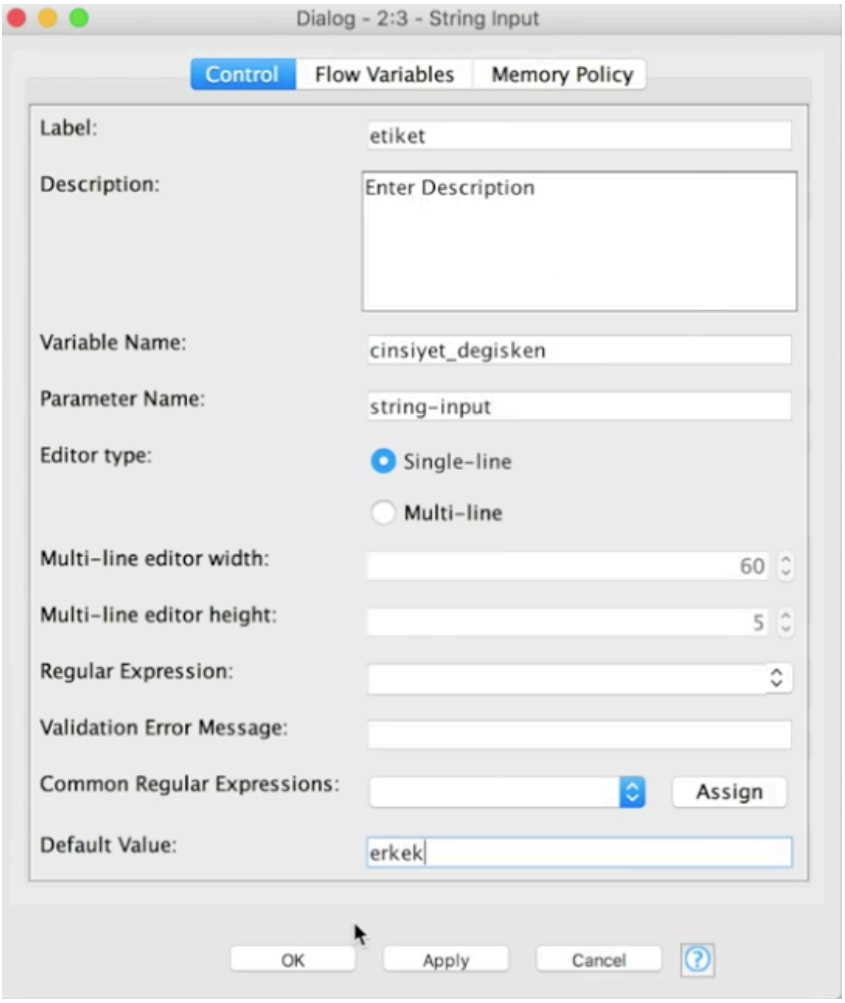
Şekil 7.2.1 row filter operatörü ile erkek bazlı kolon filtrelemeyi göstermektedir.

Row filter'da özel olarak seçilen kolonun kullanılabilmesi/düğümler üzerinde kontrol sağlanabilmesi için string input operatörü kullanılacaktır. String input kısaca değişken tanımlamaya yardımcı olan operatördür.



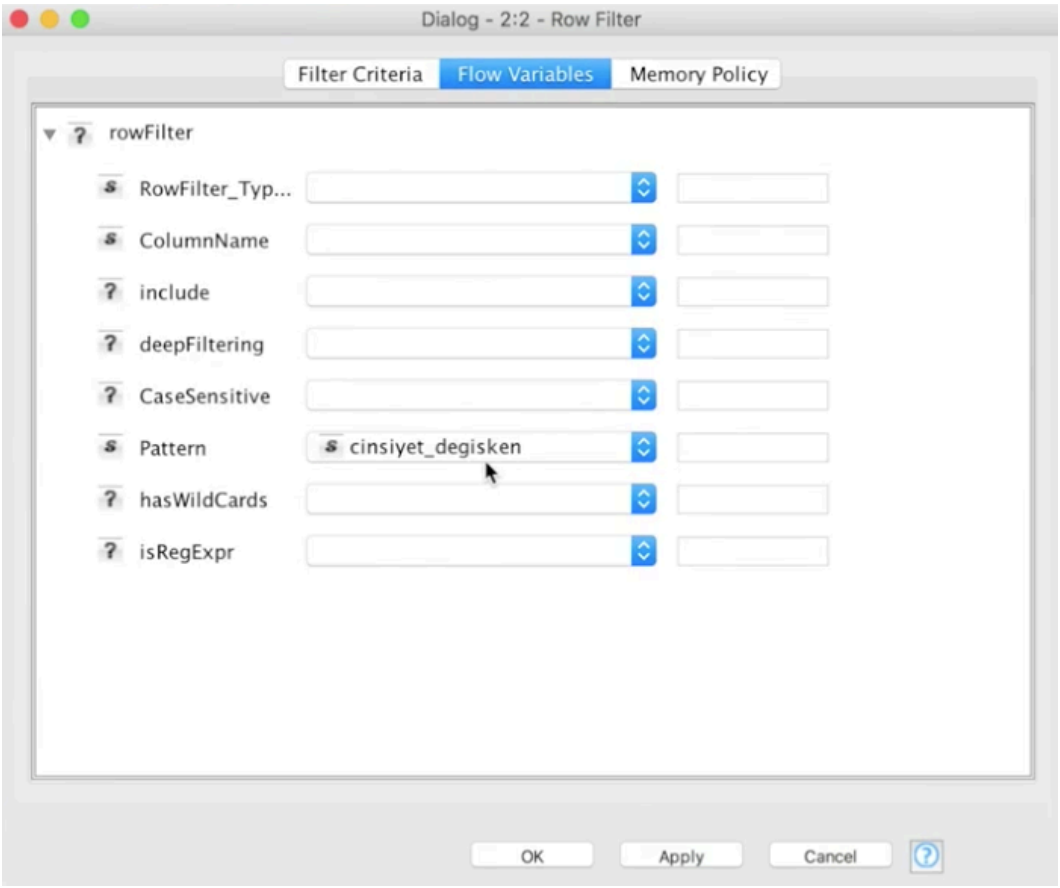
Şekil 7.2.2

Şekil 7.2.2 string input operatörünün row filter ile bağlantısını göstermektedir. Bu aşamadan sonra string input aşağıdaki gibi configure edilecektir.



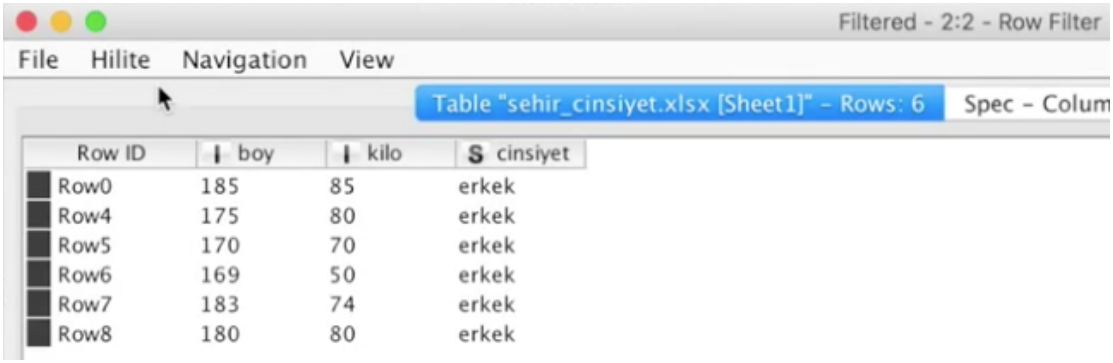
7.2.3

Şekil 7.2.3 string input içerisindeki Label, variable name ve default value değerlerinin ne olarak görüneceği belirtilmiştir.



Şekil 7.2.4

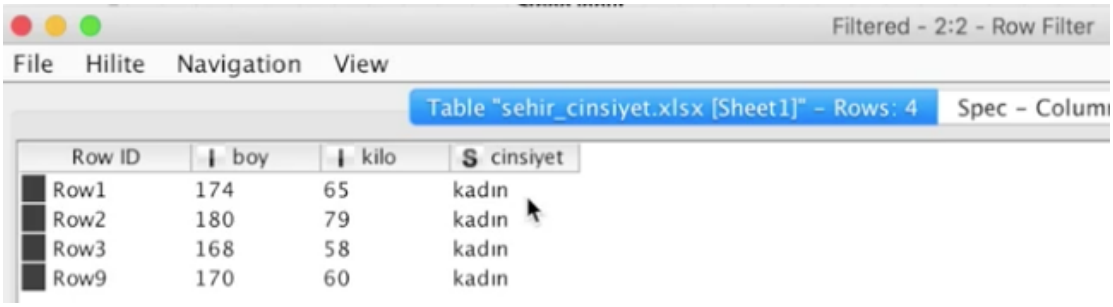
Şekil 7.2.4 pattern'e cinsiyet_değişken'inin akması için row filter operatörü configure bölümünü göstermektedir. Default olarak erkekler seçilmişti şekil 7.2.3'te. Bu şekilde run edildiğinde aşağıdaki sonuç penceresine ulaşılır.



Row ID	boy	kilo	cinsiyet
Row0	185	85	erkek
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row7	183	74	erkek
Row8	180	80	erkek

Şekil 7.2.5

Şekil 7.2.5 row filter içerisindeki filtered sonucunu göstermektir. Default value olarak erkek seçildiği için sadece erkekleri filtrelemiştir.

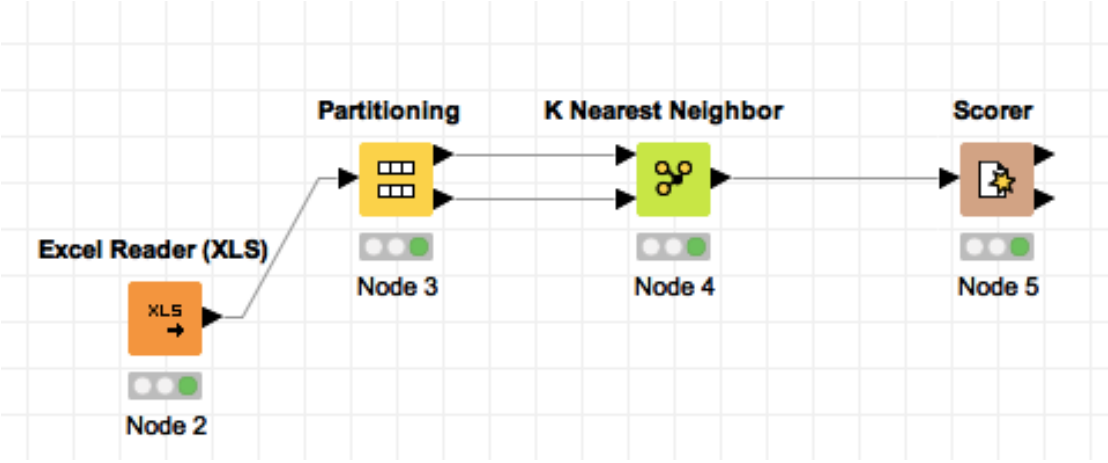


Şekil 7.2.6

Şekil 7.2.6, Şekil 7.2.3'te default value'da erkek yerine kadın seçilirse filtered sonucu bu şekilde sadece kadınların olduğu row'lar getirilir.

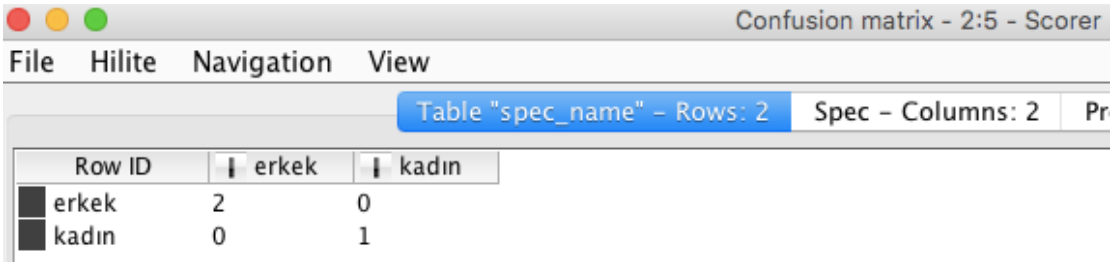
7.3 Döngüler (Loop) ve Model Parametrelerinin Test Edilmesi ve İyileştirilmesi

Bu bölümde knime üzerinden döngüler gösterilecektir.



Şekil 7.3.1

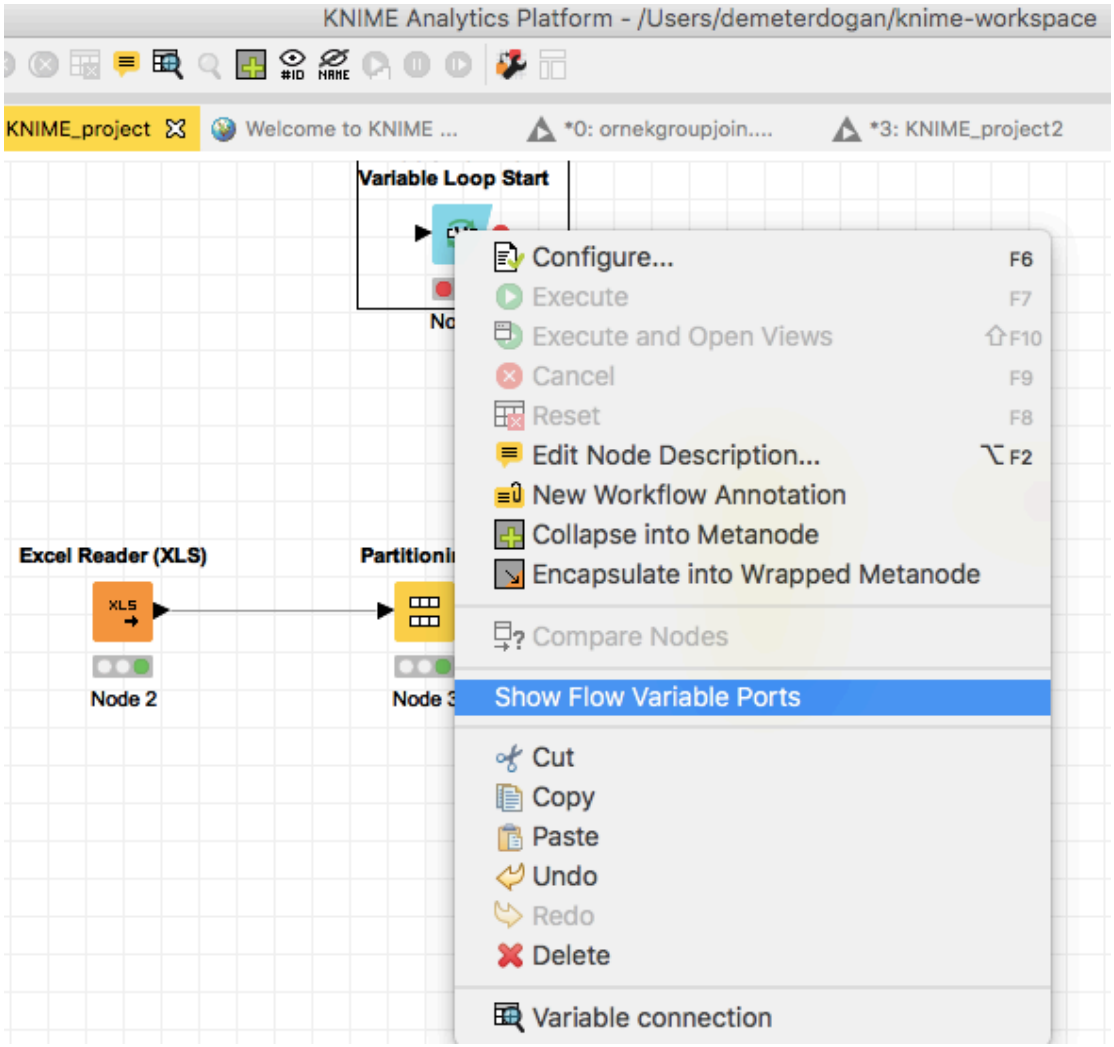
Knime'da klasik K-NN algoritma kullanımını göstermektedir. K-NN diğer algoritmalar gibi veriyi iki parçadan almaz. Partitioning'ten bölünen veriyi aynı şekilde iki parça halinde bir operatör içine alabilmektedir.



Şekil 7.3.2

Şekil 7.3.2 program bu şekildeyken run edildiğinde Confusion matrix 'i göstermektedir.

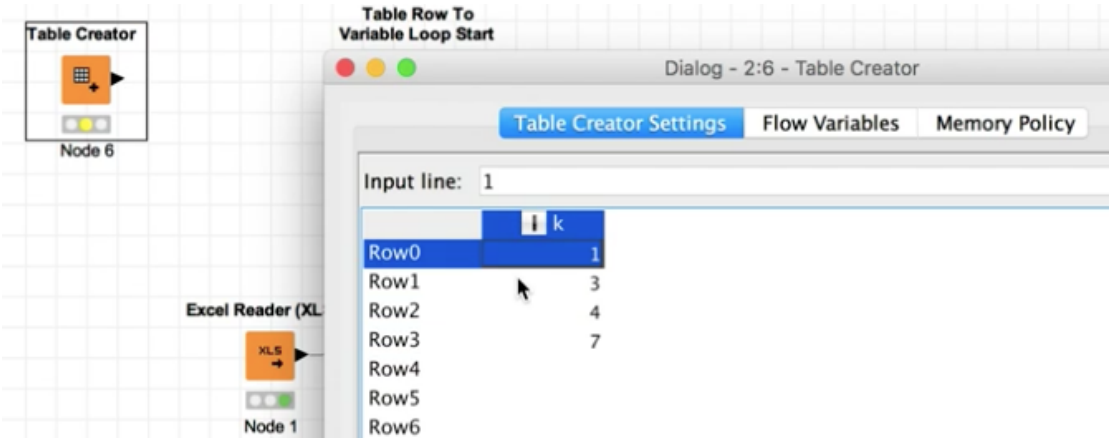
Table row to variable loop start operatörünü kullanarak hangi değişkenin akmasını yani kullanılması istenirse onun seçilmesini sağlayan operatördür.



Şekil 7.3.3

Şekil 7.3.3 table row to variable'a tıklayıp Show flow variable ports seçilerek değişkenlerin akmasını sağlayabilecek başların çıkmasını sağlamaktadır. Bu operatör bir döngü başlatmaktadır ve bu döngü satır satır okunarak değişkene çevirir.

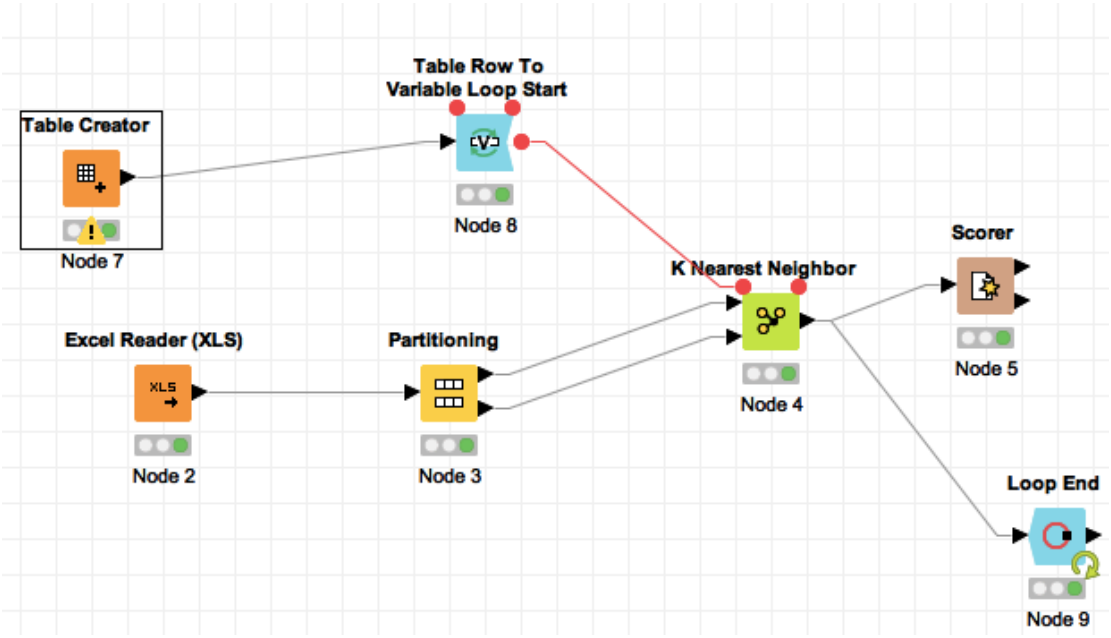
K 1 komşu, 3 komşu ve 7 komşu değerlerini deneyerek test edilmesi için k isminde kolon oluşturuldu ve aldına 1, 3, 5 ve 7 değerleri yazılır. Burada 1, 3, 5 ve 7 sadece örnek olması açısından yazılmış rakamlardır. İstenilen rakamlar kullanılabilir.



Şekil 7.3.4

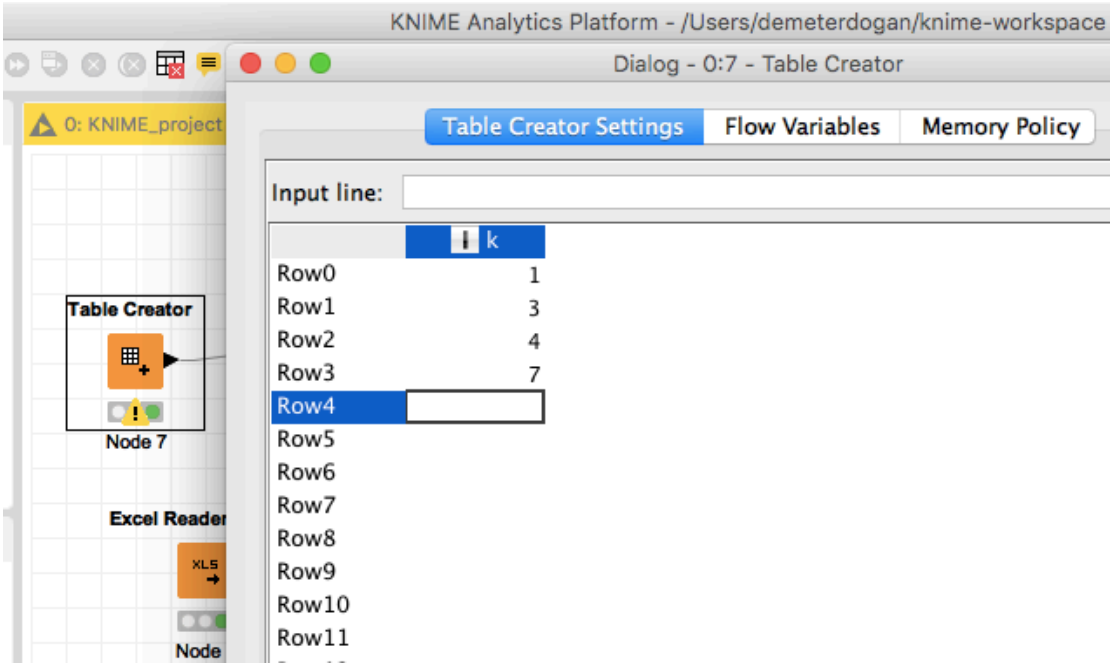
Şekil 7.3.4 table creator operatörü içerisinde manuel olarak tablo oluşturulma aşamasını göstermektedir.

Daha sonra döngünün bitmesi için loop end operatörü aşağıdaki şekilde sisteme eklenebilir.



Şekil 7.3.5

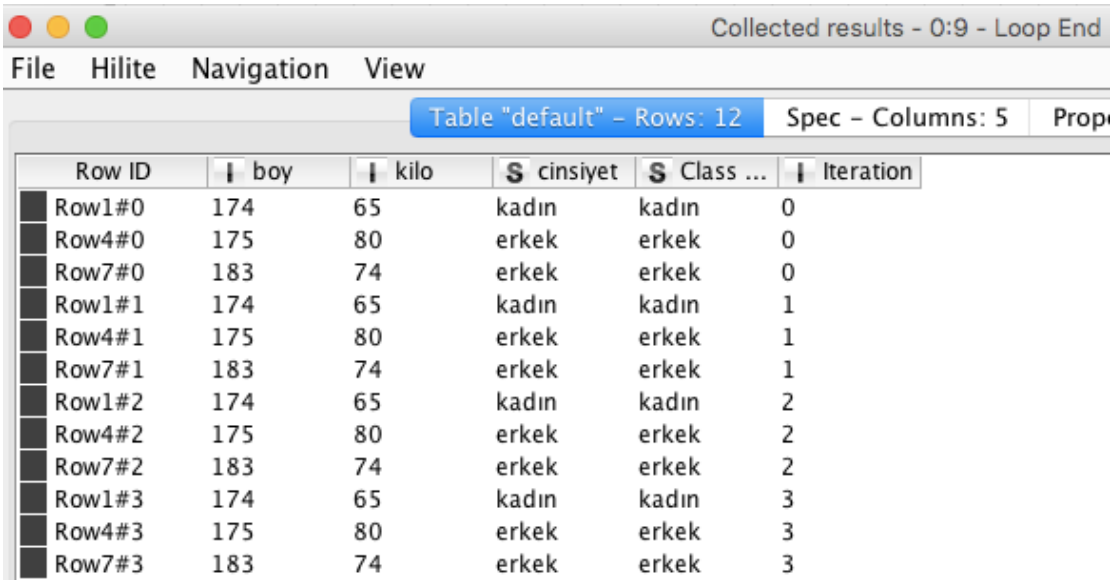
Şekil 7.3.5 tüm operatörlerin birbiri ile bağlanmasını göstermektedir.



Şekil 7.3.6

Şekil 7.3.6 table creator'da configure'de kullanılacak değerleri ve oluşturulan k kolonuna göstermediler.

Bir loop (döngünün) bitebilmesi için başka bir operatör kullanılacaktır.

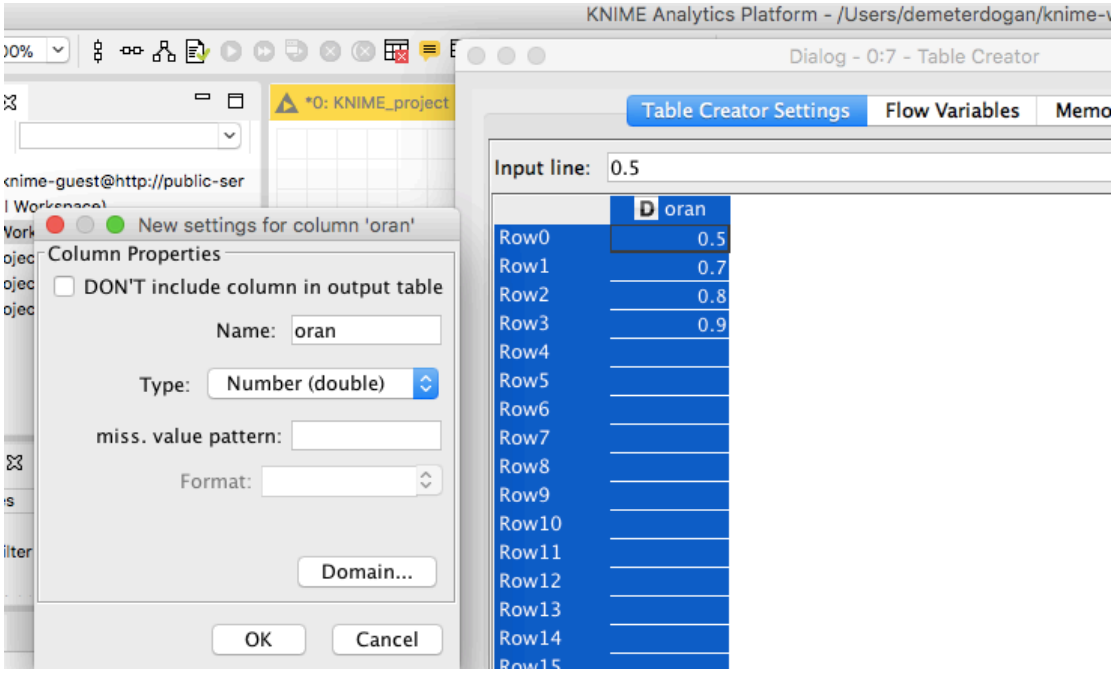


Row ID	boy	kilo	cinsiyet	Class ...	Iteration
Row1#0	174	65	kadın	kadın	0
Row4#0	175	80	erkek	erkek	0
Row7#0	183	74	erkek	erkek	0
Row1#1	174	65	kadın	kadın	1
Row4#1	175	80	erkek	erkek	1
Row7#1	183	74	erkek	erkek	1
Row1#2	174	65	kadın	kadın	2
Row4#2	175	80	erkek	erkek	2
Row7#2	183	74	erkek	erkek	2
Row1#3	174	65	kadın	kadın	3
Row4#3	175	80	erkek	erkek	3
Row7#3	183	74	erkek	erkek	3

Şekil 7.3.7

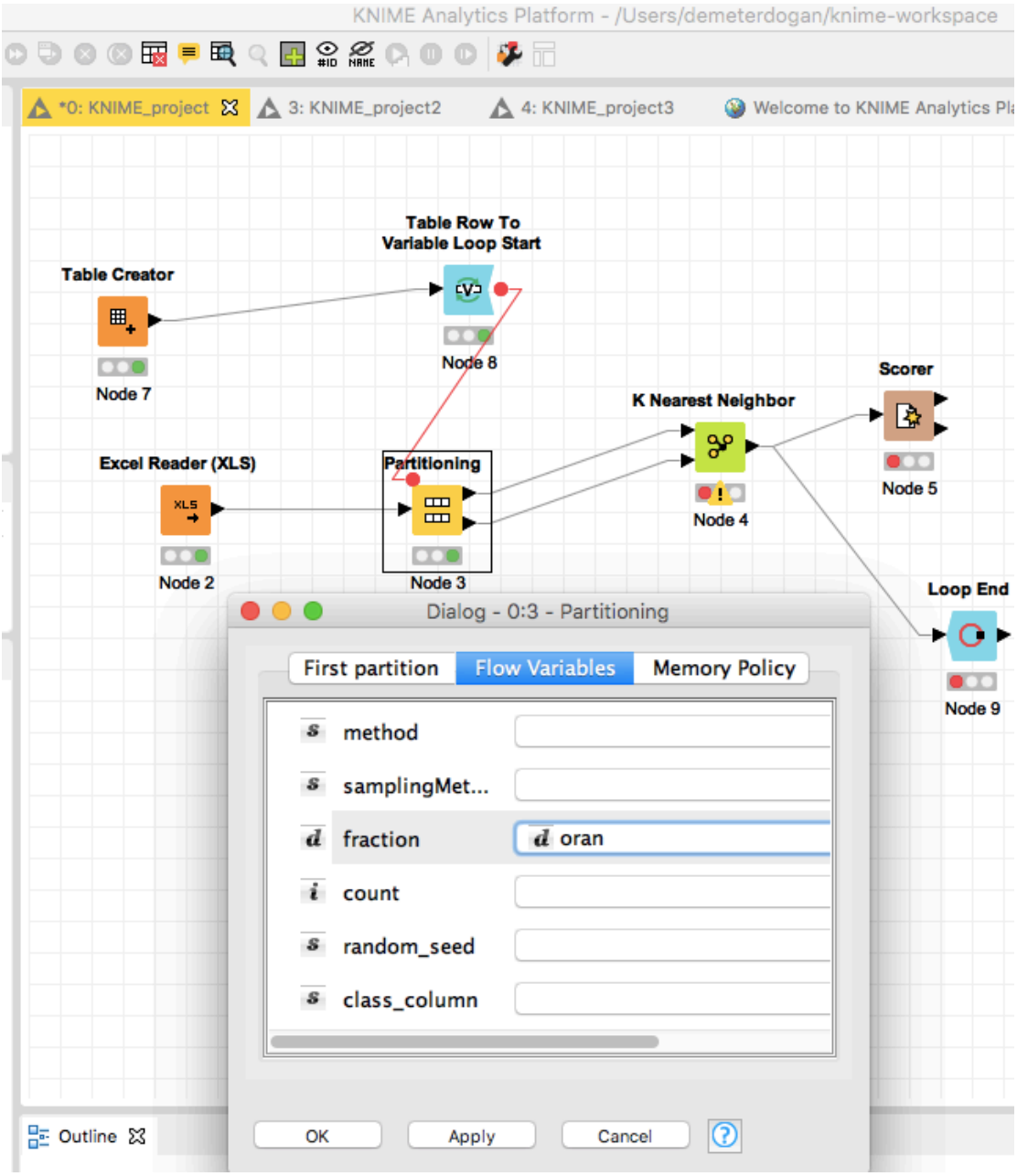
Şekil 7.3.7 loop end operatöründen ulaşılan collected result bölümündeki sonuçlardır. 1,3,4,7 baştan seçildiği için 4 loop oluşturulmuştur. Örneğin row0 döngülerinde kadın-kadın, erkek-erkek ve erkek-erkek olarak üçünü de doğru tahmin etmiş ve hata yapmamıştır.

Şekil 7.3.4'te olduğu gibi tam değerler yerine oran kullanılması için aşağıdaki ayarlamalar yapılmalıdır.



Şekil 7.3.8

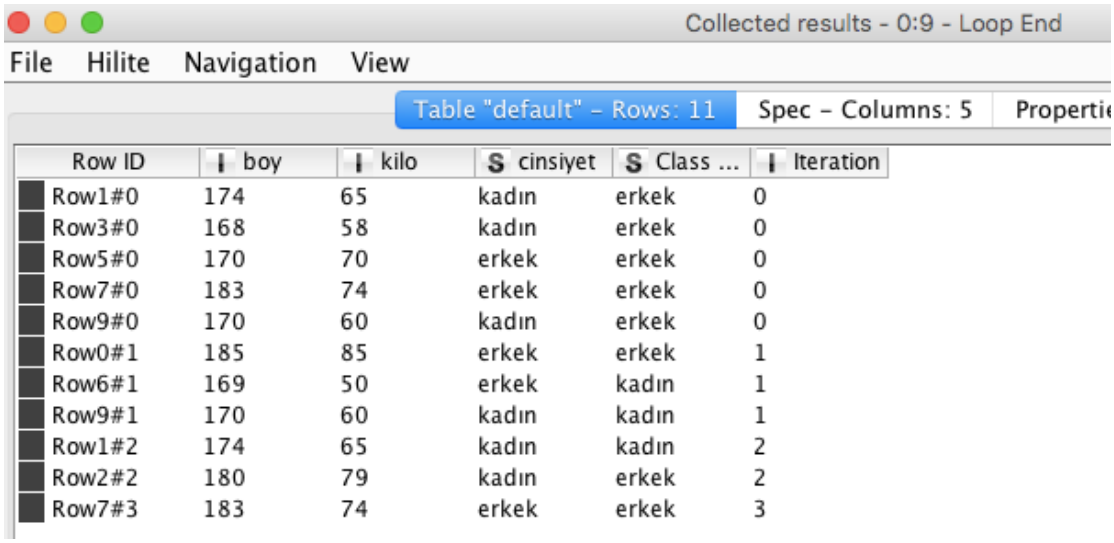
Şekil 7.3.8 Table creator'da configure ederken ondalık değerlerin kullanılması için örnek verilen sayıları göstermektedir. Type olarak ondalıklı sayı kullanılacağı için bu sefer double seçilmelidir.



Şekil 7.3.9

Şekil 7.3.9, table row to variable loop start ile partitioning arasındaki bağlantıyı ve partitioning configure bölümündeki flow variable yerinde fraction için oran seçilmesini göstermektedir.

Table creator'ı configure ettikten sonra bağlantı noktaları değiştirilmesli ve program bir kez run edildikten sonra partitioning configure edilmelidir.



Row ID	boy	kilo	cinsiyet	Class ...	Iteration
Row1#0	174	65	kadın	erkek	0
Row3#0	168	58	kadın	erkek	0
Row5#0	170	70	erkek	erkek	0
Row7#0	183	74	erkek	erkek	0
Row9#0	170	60	kadın	erkek	0
Row0#1	185	85	erkek	erkek	1
Row6#1	169	50	erkek	kadın	1
Row9#1	170	60	kadın	kadın	1
Row1#2	174	65	kadın	kadın	2
Row2#2	180	79	kadın	erkek	2
Row7#3	183	74	erkek	erkek	3

Şekil 7.3.10

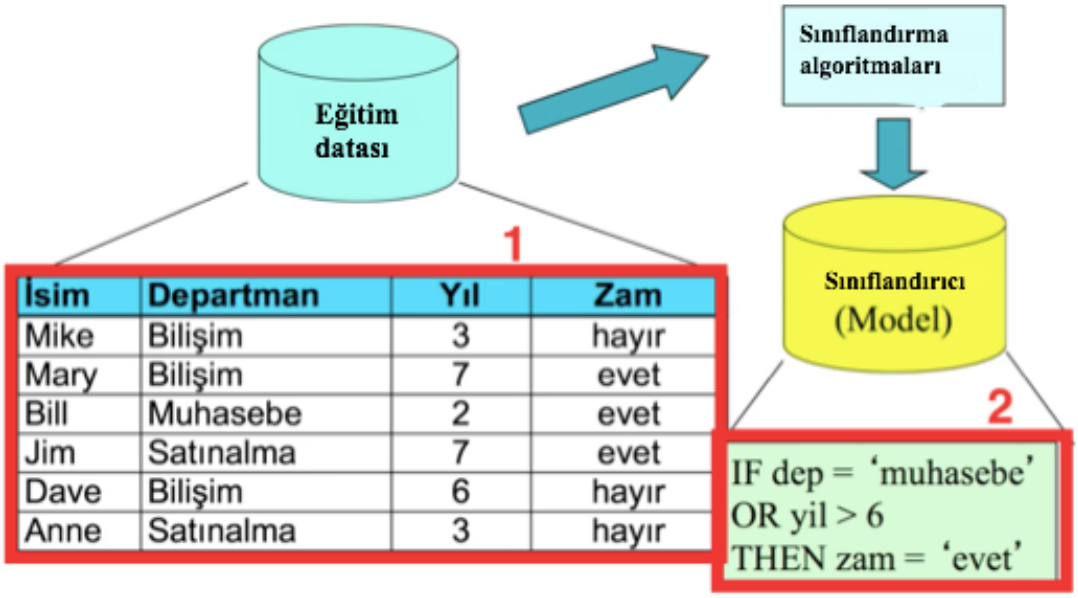
Şekil 7.3.10, program run edildikten sonra loop end collected results'ı göstermektedir. 7.3.7 count'a göre sonucu gösterirken 7.3.10 fraction'a göre sonucu göstermektedir. Ayrıca 7.3.10'da, 0.5 yani 50% seçildiği için 5 örnek row0 için, 70% öğrenme test 30% olduğu için row1 3 örnek vb. 7.3.8'de görülen değerlerin sonuca yansıması görülmektedir.

Amaç döngü oluşturmak ve bu döngüyle herhangi bir nodun parametrelerini kontrol etmek ve bu parametrelere göre en iyi bölme noktasını belirlemekti.

8. BÖLÜM: MODEL OLUŞTURMAK (MAKİNE ÖĞRENMESİ, VERİ MADENCİLİĞİ VE İSTATİKSEL MODELLER)

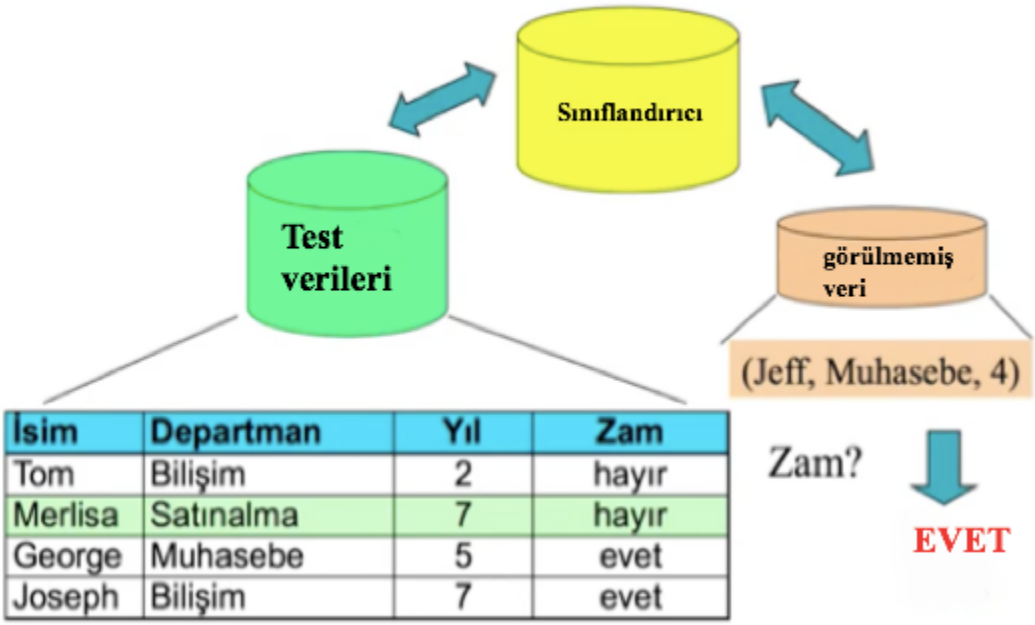
8.1 Makine Öğrenmesine Giriş: Test ve Eğitim Kümeleri, Ezberleme (Overfitting)

Bu bölümde veri kümelerinden (train, test, validation), makine öğrenmesinden, ezber / aşırı öğrenmeden (overfitting) ve rule based learning kavramından bahsedilecektir.



Şekil 8.1.1

Şekil 8.1.1 de görülen 1. Penceredeki alan makine öğrenmesi için kullanılan örnek veri setini göstermektedir. Burada makine öğrenmesi altında sınıflandırma algoritması gösterilmektedir. Burada amaç çalışılan departman ve yıl sayısına göre zam alınıp alınmayacağıdır. 2. Penceredeki kısımda basitçe kural oluşturulmuştur. IF ile ifade edilen kısım eğer 'muhasebe' departmanında veya 6 yıldan fazla çalıştıysa zam alır şeklinde okunabilir. Bu veri setindeki tüm satırlar için bu kural geçerlidir.



İsim	Departman	Yıl	Zam
Tom	Bilişim	2	hayır
Merlisa	Satınalma	7	hayır
George	Muhasebe	5	evet
Joseph	Bilişim	7	evet

Şekil 8.1.2

Şekil 8.1.2, test edilebilmesi için verilen test kümesini göstermektedir. Amacı, öğrenilmiş olan algoritmanın sınanmasıdır. Örneğin Tom'un zam alıp almadığına bakıldığında, departmandan dolayı (bilişim) almaz, ayrıca 6 yıldan az çalıştığı için de almaz sonucu çıkar. Bu yüzden makine öğrenmesi sonucu almaz şeklinde döner. Merlisa için bakıldığında, departmanından dolayı zam almaz ama 6 yıldan fazla çalıştığı için zam aldı sonucu çıkmalıydı. Fakat makine öğrenmesinden çıkan kural maalesef gerçeği bilemedi. Buradan da anlaşılacağı gibi, makine örnek bir veri seti ile eğitilir. Sonra daha önce görmediği bir veri seti üzerinden test edilir. İsim bazlı öğrenerek de (kural bazlı öğrenme) yapılabilirdi. Fakat 6 örnekte 6 farklı isim olduğu için makine ezberleyecekti ve sonucunda hiç görülmeyen isimler için sağlıklı sonuçlar çıkmayacaktı.

Yukarıda yapıldığı gibi genellikle verinin 2/3'ü eğitim, 1/3'ü test için kullanılır. Bu oran duruma göre değişmekle birlikte bazen veri bölünmeyebilir de. Burada önemli olan makineyi test etmektir. Diğer önemli durum ise, tahmin'dir. Eğer nominal değer tahmin edilmeye çalışılıyorsa buna classification adı verilir.

Bu bölümde veri kümesinin bölünmesinden, makine öğrenmesinin en temeli olan rule based öğrenmeden ve ezber (overfitting) öğrenmeden basit bir örnek ile bahsedildi.

8.2 Naive Bayes ve Bayes Teoreminin Veri Biliminde Kullanımı

Bu bölümden itibaren makine öğrenme algoritmalarına giriş yapılacaktır. Kural tabanlı algoritmalarda genellikle gerçek hayatta ortak kurallar çıkarılamadığı için kullanımı kısıtlıdır. Olasılıksal/ istatiksel yöntemlerden olan Naive Bayes teoremi Bu bölümde açıklanacaktır. Teorem koşullu olasılığı açıklamaktadır yani B'in gerçekleşmesi durumunda A'nin olasılığını sorgulamaktadır.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

P(A|B) → sonsal olasılık

P(B|A) → benzerlik

P(A) → önsel olasılık

Bu örnek koşul değiştirme durumunu anlamına da gelmektedir. Örneğin B'nin gerçekleştiği durumda A'nın olma olasılığı sorgulanırken bu A'nın gerçekleştiği durumda B'nin olma olasılığı olarak da sorgulanabilir. Tabii bu durumda ayrı ayrı A ve B nin olasılıkları bilindiğinde cevaplanabilir.

Örneğin, bilgisayar satan bir dükkan olsa ve gelen öğrencinin buradan bilgisayar alma olasılığı merak edildiğinde öncelikle öğrenci olduğu kabul edilmiş olunur. İkinci olarak da öğrencilerden kaç tanesi satın alma işlemi yapmış, öğrenci olup satın alma (conditional), satın almış kişilerin kaçı öğrenci hepsinin birbiriyle yer değiştirmesi incelenebilir. Satılmış bir ürünün mesela satın alma işlemi yapanların kaçı öğrencidir bilgisine ulaşmak kolaydır. Ama önemli olan öğrenci olan birinin bu mağazadan ürün alma ihtimalini hesaplamak önemlidir. Bayesian teoremi bu ihtimali sorgulama özelliğini sunar.

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Şekil 8.2.1

Şekil 8.2.1 görülen tablo, Jiawei Han'ın Data Mining: Concepts and Tecchniques kitabından alınmıştır. Türkçe'ye çevirilmiş Tablo 8.2.2 de görülmektedir.

Yaş	Gelir Düzeyi	Öğrencilik Durumu	Kredi notu	Bilgisayar alımı
<= 30	Yüksek	Hayır	Uygun	Hayır
<= 30	Yüksek	Hayır	Çok iyi	Hayır
31...40	Yüksek	Hayır	Uygun	Evet
>40	Orta	Hayır	Uygun	Evet
>40	Düşük	Evet	Uygun	Evet
>40	Düşük	Evet	Çok iyi	Hayır
31...40	Düşük	Evet	Çok iyi	Evet
<= 30	Orta	Hayır	Uygun	Hayır
<= 30	Düşük	Evet	Uygun	Evet
>40	Orta	Evet	Uygun	Evet
<= 30	Orta	Evet	Çok iyi	Evet
31...40	Orta	Hayır	Çok iyi	Evet
31...40	Yüksek	Evet	Uygun	Evet
>40	Orta	Hayır	Çok iyi	Hayır

Tablo 8.2.2

Şekil 8.2.2 yaş gruplarına (age), gelir düzeyine (income), öğrenci olup olmama (student), kredi skoruna (credit_rating) göre kategorileri olan ve sonucunda mağazadan bilgisayar alıp almadığını veren örnek bir veri setini göstermektedir. Buradaki tüm veriler nominal verilerdir. Naive bayes için bu önemlidir. Yaş normalde sayısal olmasına rağmen bin'lere bölünerek nominal yapılmıştır. Veriler bu şekilde açıkça belirtildiğinde daha sonrasında buradan hangi müşterilere ne gibi kampanyalar yapılmalı gibi soruların cevapları bulunabilir. Yani genelde satış departmanlarının genellikle önemsedikleri alan alma ihtimali olacak müşterileri saptamak ve onlara odaklanmaktır.

Classification yöntemi genel olarak rule based learning ile de çalışır. Örneğin burada 30..40 yaş aralığındaki kişiler bilgisayar alır denirse bu veri seti için doğru sonuç çıkar. Fakat yaş grubu artarsa ya da farklı bir özelliğe sahip müşteri gelirse başarısız olma ihtimali de yükselebilir.

Naive bayes için öncelikle müşterinin alma ve almama ihtimali hesaplanır. Daha sonrasında almış müşterilerin grupları ve olasılıkları çıkarılır. Örneğin bilgisayar almış

ve <30 (30 yaş altı) müşterilerin olasılığı vb. tüm olasılıklar çıkartılır. Hatta bilgisayar almış, <30 ve düşük gelir, yüksek gelir düzeyine sahip müşterilerin de çıkartılabilir.

Bu veri setinden yola çıkarak, yaşı 30'dan küçük, gelir düzeyi orta, öğrenci ve kredi durumu idare eden (fair) bir kişinin bilgisayar alıp almayacağı merak edilirse

$P(C_i)$: P(bilgisayar_alımı = "evet") = $9/14$ = 0.643

P(bilgisayar_alımı = "hayır") = $5/14$ = 0.357

$P(X|C_i)$ ile tüm sınıflandırmalar için hesaplamalar:

$P(yaş = " \leq 30" \mid bilgisayar\_alım = "evet")$ = $2/9$ = 0.222

$P(yaş = " \leq 30" \mid bilgisayar\_alım = "hayır")$ = $3/5$ = 0.6

$P(gelir\_düzeyi = "orta" \mid bilgisayar\_alım = "evet")$ = $4/9$ = 0.444

$P(gelir\_düzeyi = "orta" \mid bilgisayar\_alım = "hayır")$ = $2/5$ = 0.4

$P(öğrencilik\_durumu = "evet" \mid bilgisayar\_alım = "evet")$ = $6/9$ = 0.667

$P(öğrencilik\_durumu = "evet" \mid bilgisayar\_alım = "hayır")$ = $1/5$ = 0.2

$P(kredi\_notu = "vasat" \mid bilgisayar\_alım = "evet")$ = $6/9$ = 0.667

$P(kredi\_notu = "vasat" \mid bilgisayar\_alım = "hayır")$ = $2/5$ = 0.4

X = (yaş <= 30, gelir_düzeyi = orta, öğrencilik_durumu = evet, kredi_notu = Uygun**)**

$\boldsymbol{P(X|C_i)}$**:** $P(X \mid bilgisayar\_alım = "evet")$ = 0.222 x 0.444x 0.667= 0.044

$P(X \mid bilgisayar\_alım = "hayır") = 0.6 \; x \; 0.4 \; x \; 0.2 \; x \; 0.4 = 0.019$

$\boldsymbol{P(X|C_i) * P(C_i)}$**:** $P(X \mid bilgisayar\_alım = "evet")$*P(bilgisayar_alımı="evet")=0.028

$P(X \mid bilgisayar\_alım = "hayır")$*P(bilgisayar_alımı="evet")=0.007

Bu yüzden X, ("bilgisayar_alımı="evet") e bağlı olur.

Şekil 8.2.3

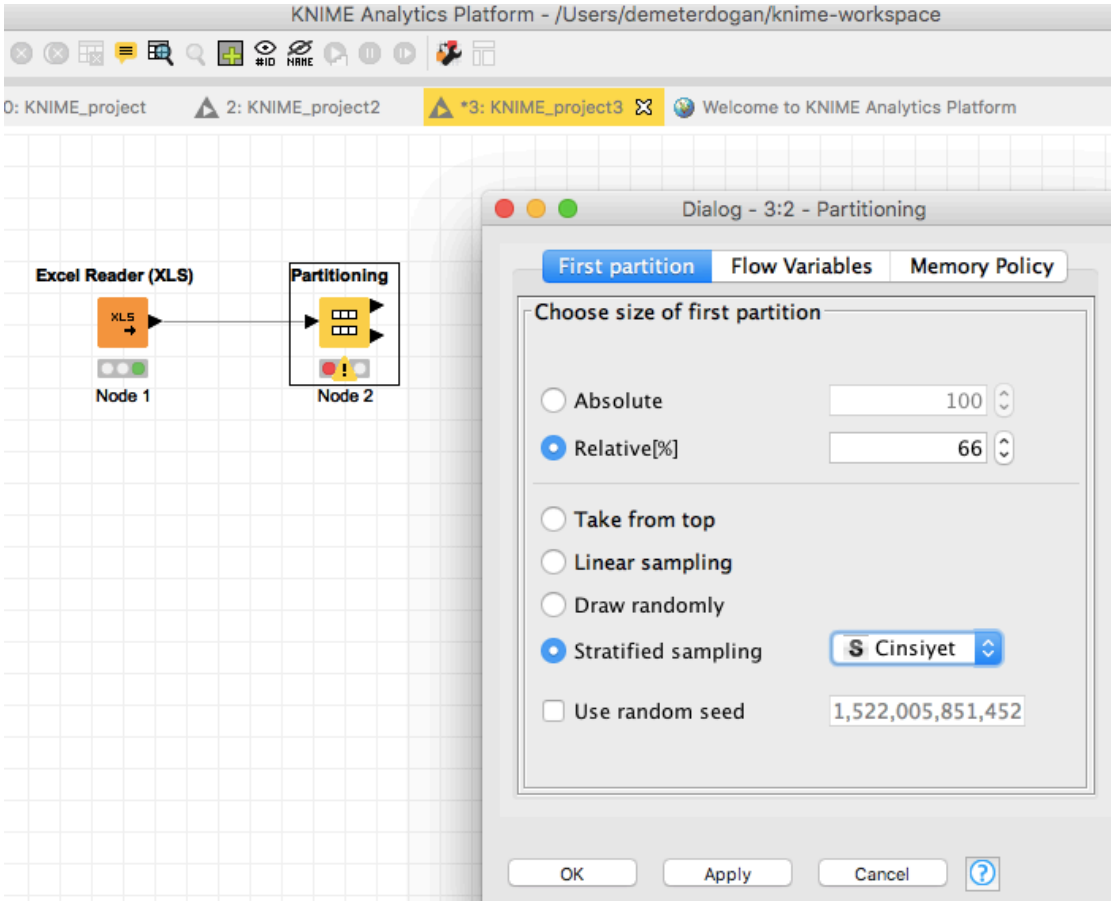
Şekil 8.2.3 'de görülen X, ulaşılmak istenen değerdir. Bu örnekte 4 boyut yukarıda açıklanmıştır. Bunlar; kredi durumu (credit_rating) , öğrencilik durumu(student yes/no) , yaş durumu (age) ve gelir durumu (income). Olasılıkları Şekil 8.2.3'te görülmektedir. Sonuçtan da anlaşıldığı gibi yaşı 30'dan küçük, gelir düzeyi orta, öğrenci ve kredi

durumu idare eden (fair) bir kişinin bilgisayar alma olasılığı (0.028) almama olasılığından (0.007) dört kat fazladır. Bu kişi için kampanya yapılıp bilgisayar satılabilir sonucu çıkarılabilir. Şekiller Jiawei Han'ın Data Mining: Concepts and Tecchniques kitabından alınmıştır.

8.3 Numerik Verilerin Kutulanması (binning) ve Naive Bayes Uygulaması (Knime ile)

Bu bölümde amaç bir önceki bölümde açıklanan teorinin Knime üzerinden uygulamasını göstermektir.

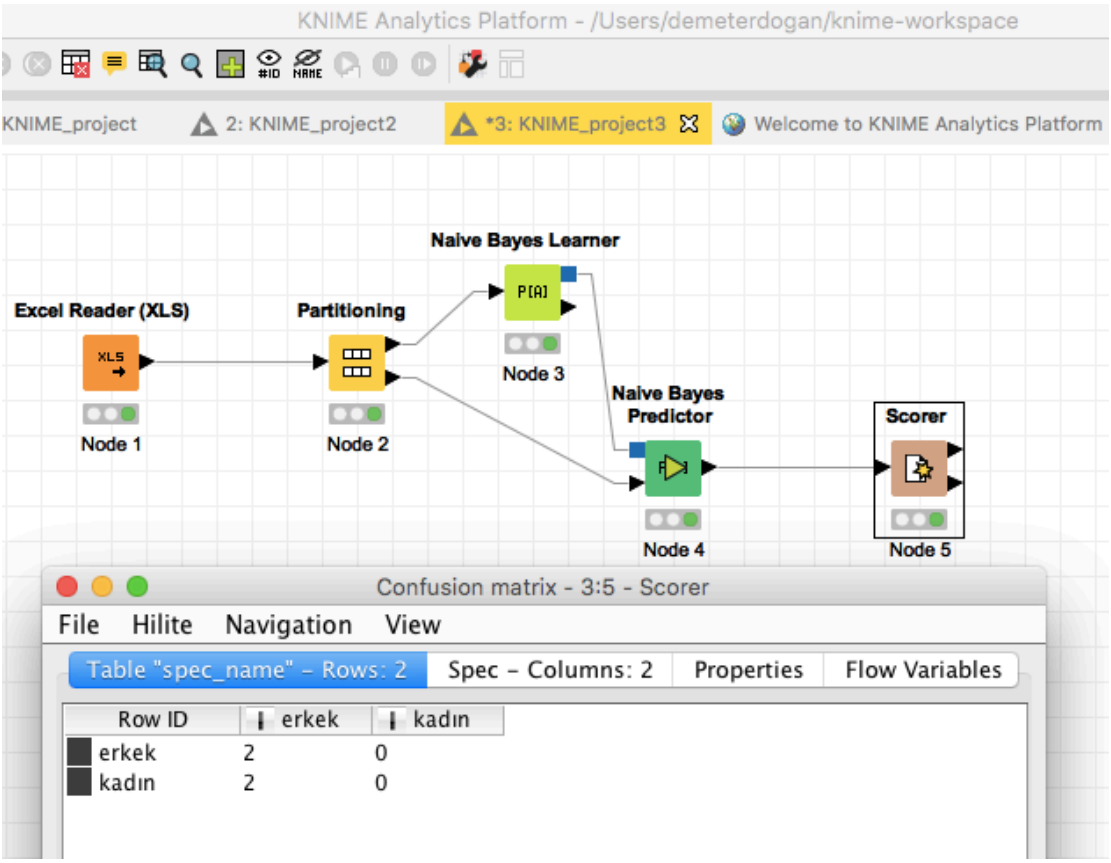
Daha önceki bölümlerde kullanılan boy ve kilodan cinsiyete ulaşmaya çalışılan veri seti bu bölümde de kullanılacaktır. Veriyi yine eğitim ve test için bölmek gerekeceği için aşağıdaki gibi partitioning operatörü bağlanmıştır.



Şekil 8.3.1

Şekil 8.3.1'de görülen relative ile verilerin 66% ve 34% şeklinde bölüneceği ve stratified sampling ile de veri içerisindeki erkek ve kadınların aynı oranda bölünmüş veri ile bölüneceği anlamına gelmektedir. Örneğin veri setinde 80% erkek varsa seçilen 66% yüzde içerisinde de 80% erkek ve 20% kadın oranı korunur.

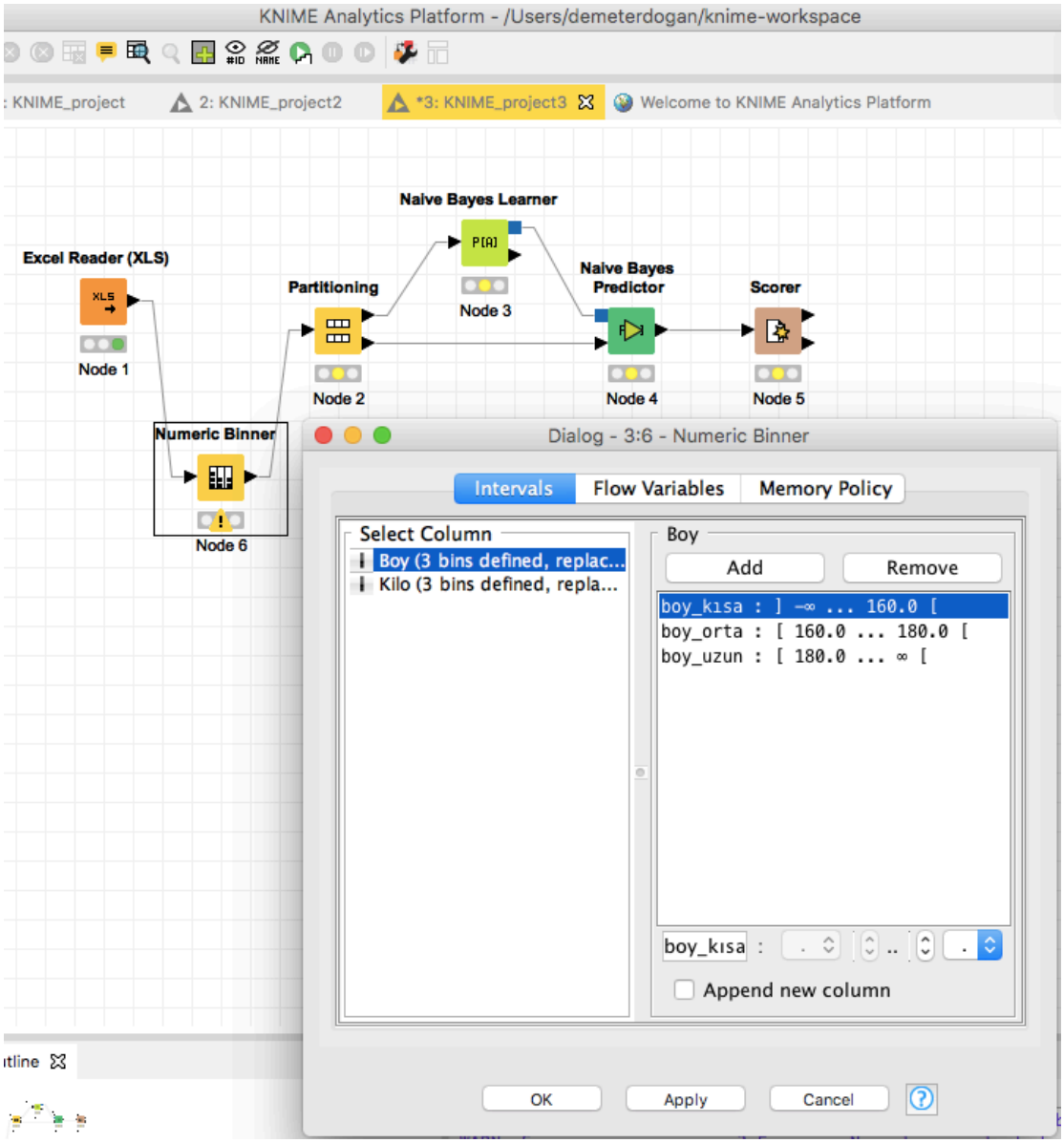
Naive Bayes learner yukarıda da açıklandığı gibi veriden condition probability değerlerini çıkarmak için eklenecektir.



Şekil 8.3.2

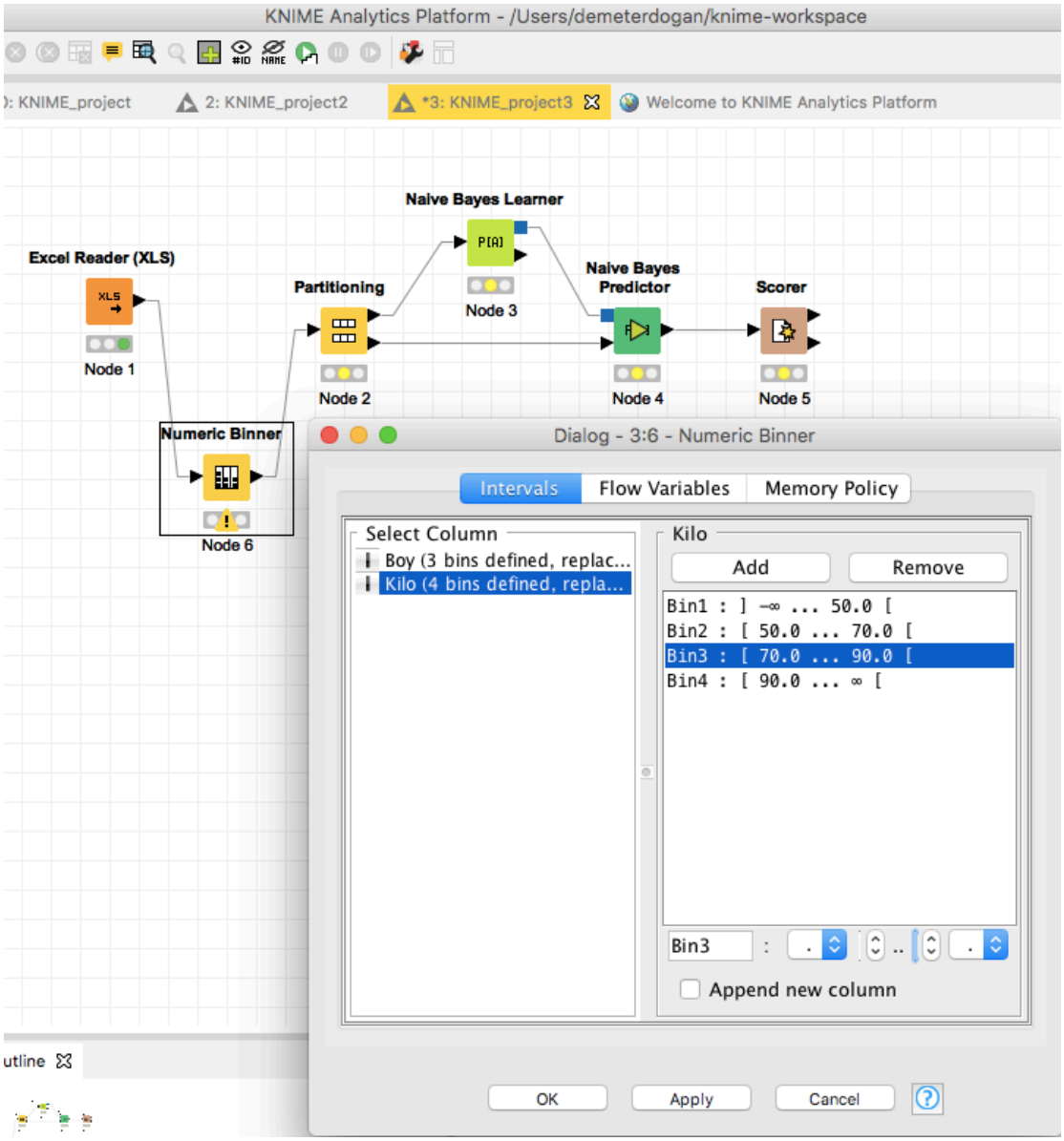
Şekil 8.3.2 Naive bayes learner, naive bayes predictor ve scorer bağlantılarını göstermektedir. Scorer'dan confusion matrix sonucuna göre diagonal düzleme bakıldığında; erkekler 2'de 2 yani 100% başarı ile tahmin edilirken kadınların tamamı yanlış tahmin edilmiştir.

Önceki bölümde belirtilen numeric değerin olmaması bunun yerine nominal değerlerin olması durumu Knime için numeric binner operatörü ile sağlanabilir.



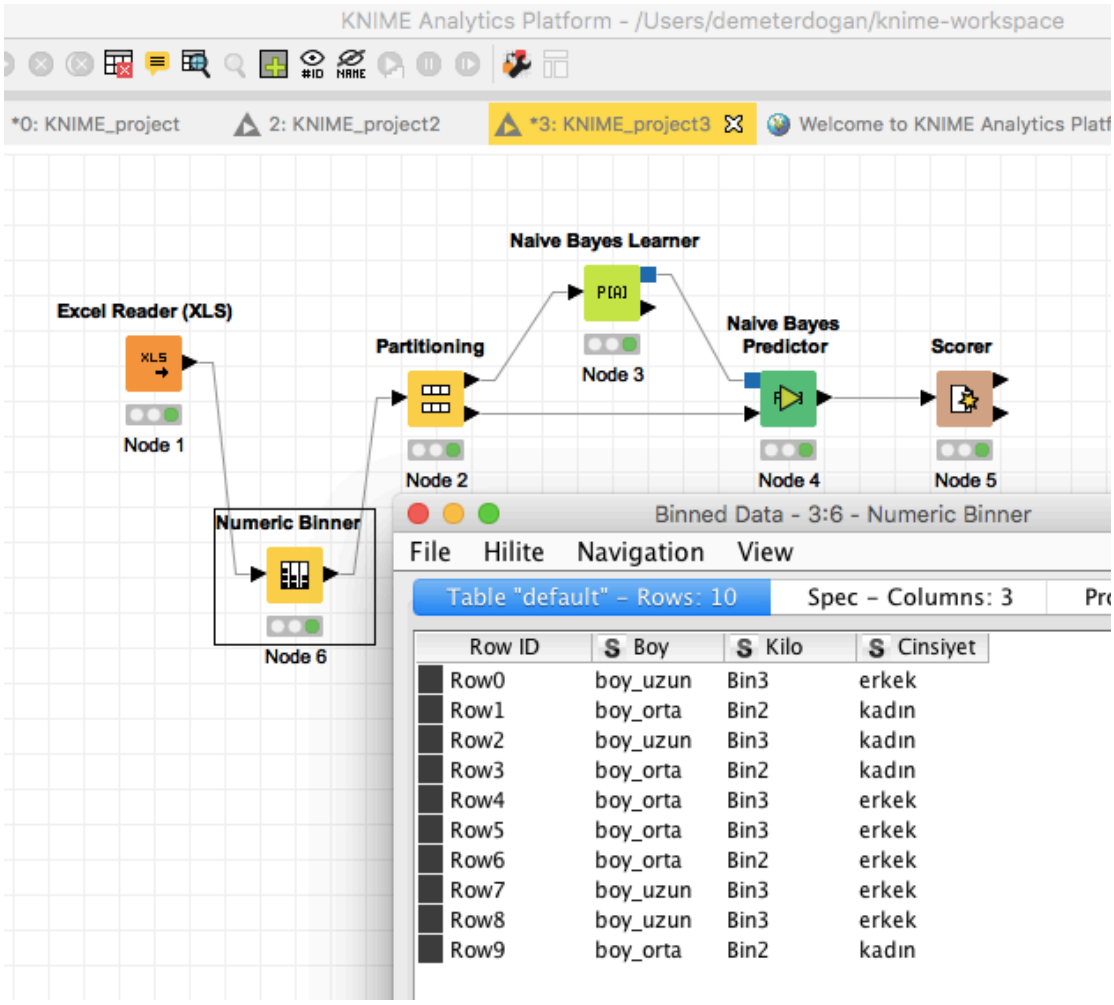
Şekil 8.3.3

Şekil 8.3.3 numeric binner operatör bağlantısını ve içerisine manuel olarak add tuşu ile 3 girdi verilen, boy_kısa, boy_orta, boy_uzun ve yine sonradan eklenmiş 160, 180 sınırlarını göstermektedir.



Şekil 8.3.4

Şekil 8.3.4, yukarıda olduğu gibi yine sonradan sınırları eklenmiş 4 sınırı göstermektedir. Bin1 eksi sonsuzdan 50'te kadar olanları, Bin2 50 ile 70 arasını, Bin3 70 ile 90 arasını ve Bin4 90 ile artı sonsuz aralığındağı değerleri belirtmektedir. Sağ taraftaki sınırlar dahil değildir. Örneğin Bin1 50'ye dahil değildir. 50 Bin2'nin, Bin 2 70'i dahil etmemiş ve 70 Bin3'ün sınırları altındadır.



Şekil 8.3.5

Şekil 8.3.5, program çalıştırıldıktan sonra numeric binner sonucunu göstermektedir. Eskiden boy kolonunda 160-170-180 gibi değerler yazarken şu anda artık verilen değer aralıklarındaki isimler yani boy_uzun, boy_orta gibi değerler yazmaktadır. Bu şekilde diğer tüm operatörlerde de sayısal değerler yerine aynı şekilde verilen yeni isimli nominal değerler verilmektedir.

8.4 Karar Ağacı (Decision Tree) Öğrenmesi

Bu bölümde amaç karar ağacı öğrenmesini göstermektir.

Prediction sayısal veriler üzerinde çalışırken classification ise label tahmini yapmaktadır. İkisi de sınıflandırma gibi görünse de literatürdeki farklılıkları buradandır. Decision tree ise sınıflar üzerinde etiketlendirme yapmaktadır.

Yaş	Gelir Düzeyi	Öğrencilik Durumu	Kredi notu	Bilgisayar alıp almadığı
<= 30	Yüksek	Hayır	Uygun	Hayır
<= 30	Yüksek	Hayır	Çok iyi	Hayır
31...40	Yüksek	Hayır	Uygun	Evet
>40	Orta	Hayır	Uygun	Evet
>40	Düşük	Evet	Uygun	Evet
>40	Düşük	Evet	Çok iyi	Hayır
31...40	Düşük	Evet	Çok iyi	Evet
<= 30	Orta	Hayır	Uygun	Hayır
<= 30	Düşük	Evet	Uygun	Evet
>40	Orta	Evet	Uygun	Evet
<= 30	Orta	Evet	Çok iyi	Evet
31...40	Orta	Hayır	Çok iyi	Evet
31...40	Yüksek	Evet	Uygun	Evet
>40	Orta	Hayır	Çok iyi	Hayır

Tablo 8.4.1

Tablo 8.4.1, daha önceki bölümlerde de kullanılan veri setidir. Decision treenin normalde birden fazla algoritması vardır fakat bu örnekte ID3/C4.5 algoritması gösterilecektir. Bu algoritmadaki amaç veriden oluşturulan ağacın minimum derinlikte tutulmasıdır. Bu yaggıyla yola çıkarak bir özellik seçilir ve en tepede o olur. Decision tree'de en yukarıda görülen age (yaşa göre) sınıflandırmasının nedeni entropy'dir.

Entropy (expected informarion) (dağıntı/entropi):

Bir tanımlama grubu içinde (D) sınıflandırmak için gerekli entropi:

$$Info(D) = -\sum_{i=l}^{m} p_i log_2(p_i)$$

Info →bilgi,

gain →kazanç demektir.

"D" sınıflandırma için kullanılan bilgiler:

$$Info(D) = -\sum_{i=l}^{m} p_i log_2(p_i)$$

Öznitelik üzerindeki A dallanma tarafından elde edilen bilgi:

$$\text{Gain(A)} = \text{Info(D)} - Info_A(D)$$

Yukarıdaki formüller decision tree'deki dalların ve tepedeki elemanın (entropy) hesaplanması için formülleri göstermektedir.

Sınıf P:

bilgisayar_alımı = "yes"

$$Bilgi_{yaş}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = (0.694)$$

Sınıf N:

bilgisayar_alımı = "no"

$$Bilgi(D) = I(9,5) = -\frac{9}{14} log_2\left(\frac{9}{14}\right) - \frac{5}{14} log_2\left(\frac{5}{14}\right) = 0.940$$

yaş	p_i	n_i	I(p_i, n_i)
<=30	2	3	0.971
31....40	4	0	0
>40	3	2	0.971

$\frac{5}{14}$ $I(2,3)$ anlamı; 14 kişiden 5 kişi “yaş <=30” ve bunların 2 si “evet” 3’ü “hayır” çıkmıştır.

Dolayısıyla,

Kazanç(yaş) = Bilgi(D) – $Bilgi_{yaş}(D)$ = 0.246

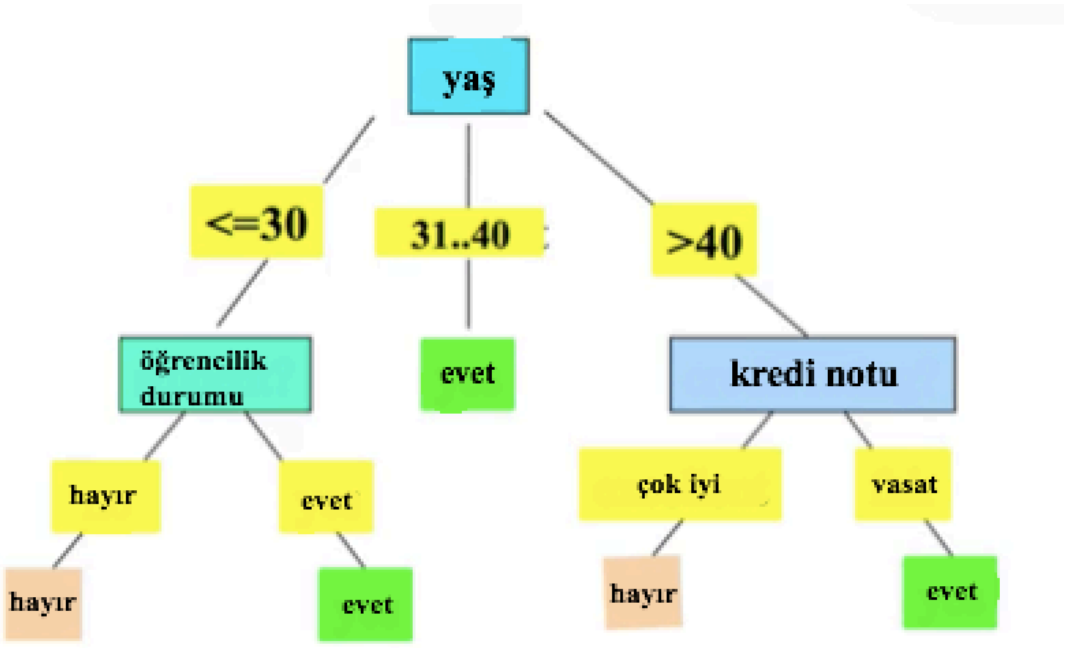
Benzer şekilde;

Kazanç (gelir) =0.029

Kazanç (öğrencilik durumu) = 0.151

Gain (kredi notu) = 0.048

Yukarıda, yaş, gelir durumu, öğrenci olma durumu ve kredi notu durumları için ayrı ayrı kazanç hesaplamasını gösterilmektedir. Sonuca göre sırasıyla 0.246, 0.029, 0.151 ve 0.048 arasından en yüksek değer yaş’ın olduğu için ağaçta en yukarıdaki bölüme yaş yazılır ve buna göre dallanma oluşturulur. Bu şekilde bir sonraki dal oluşturulur.

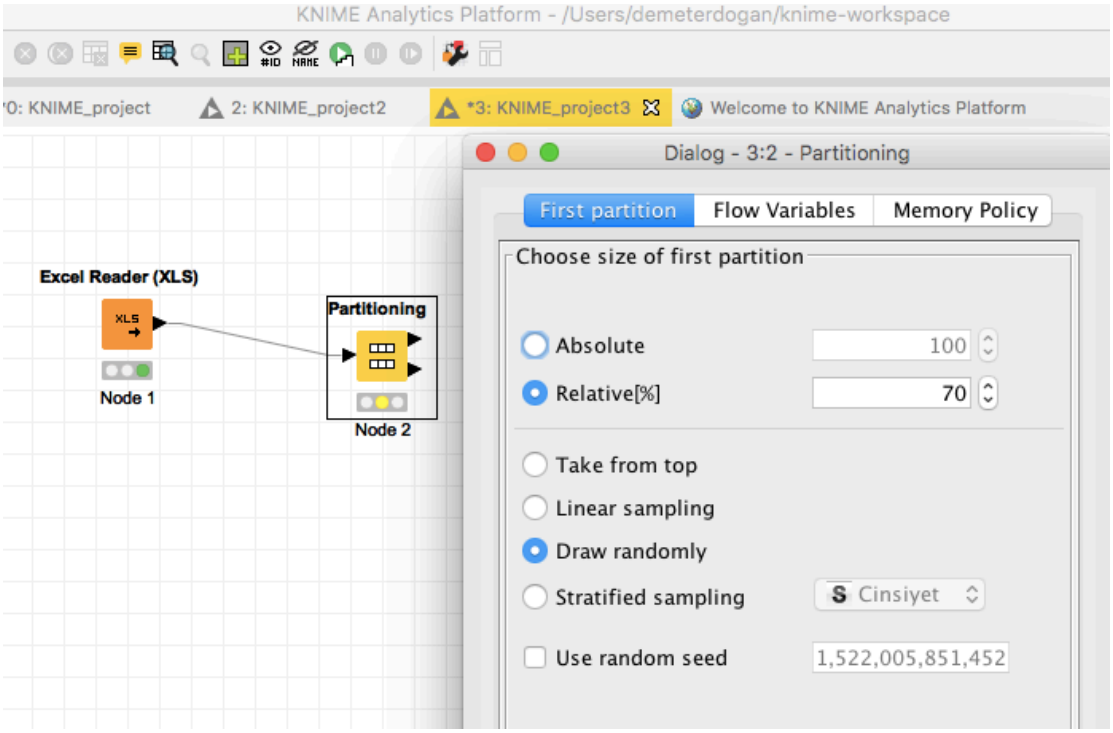


Şekil 8.4.4

Şekil 8.4.4 decision tree (karar ağacı) sonucunu göstermektedir. Decision tree'deki her bir devam dalı rule based learning kuralı olarak yazılabilir. Örneğin, 30 yaşından küçükse, öğrenciyse bilgisayar alır veya 31-40 yaş aralığındaysa bilgisayar alır ya da 40 yaşından büyükse, kredi durumu uygun ise (fair) bilgisayar alır sonuçlarına varılabilir. Rule based tabanlı sistem decision tree'ye teorik olarak çevrilebilmesi gerekir ama genelde bu yapılamaz. Bu çevirmenin aslında çok dezavantajı vardır. Nedeni; kural tabanlı sistem karar ağacına çevrildiğinde çok fazla karar dalı (devam yolu) oluşmasıdır.

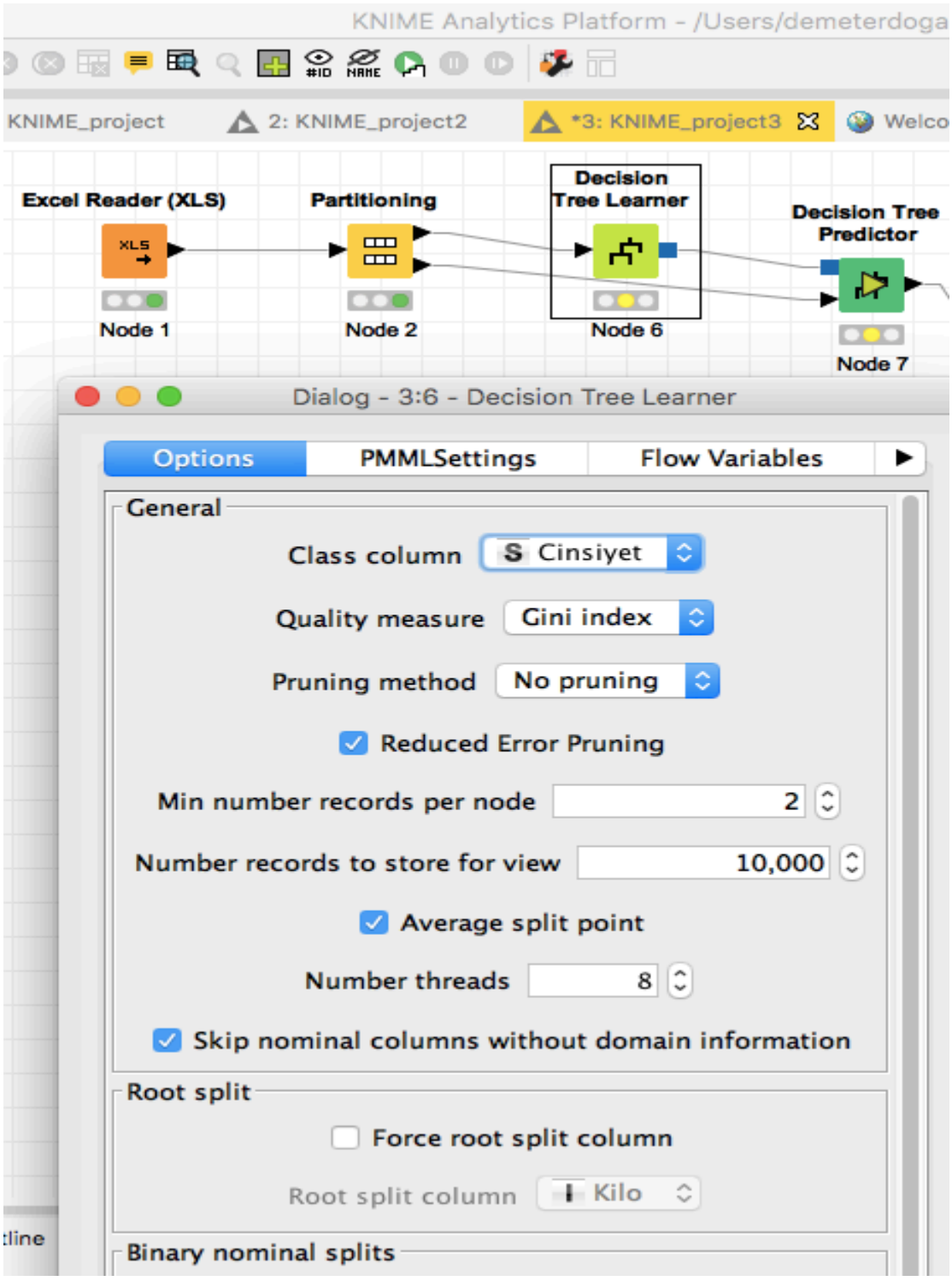
8.5 PMML Dosya Kullanımı ve Knime ile Decision Tree (Karar Ağacı) Uygulaması

Bu bölümde amaç, bir önceki bölümde açıklanan decision tree teorisinin uygulamasını göstermektir. Daha önceki bölümlerde kullanılan boy, kilo cinsiyet veri kümesi kullanılacaktır.



Şekil 8.5.1

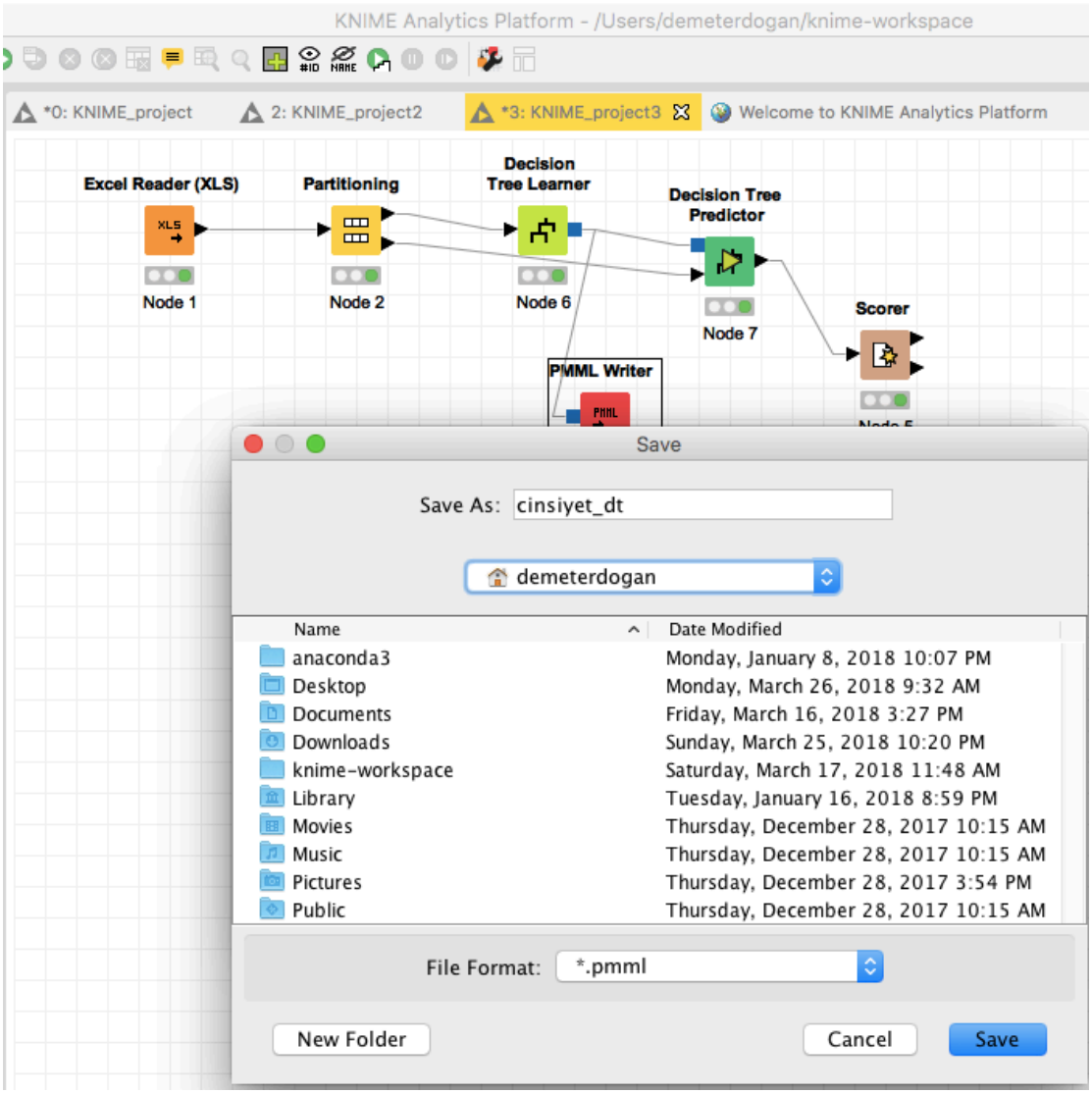
Şekil 8.5.1, partitioning'deki configure bölümündeki seçimleri göstermektedir. Veri seti relative 70% yani 70%'i train 30% test için parçalandı ve test'e ve train'e alınacak veriler randomly (rastgele) seçilecek anlamına gelmektedir.



Şekil 8.5.2

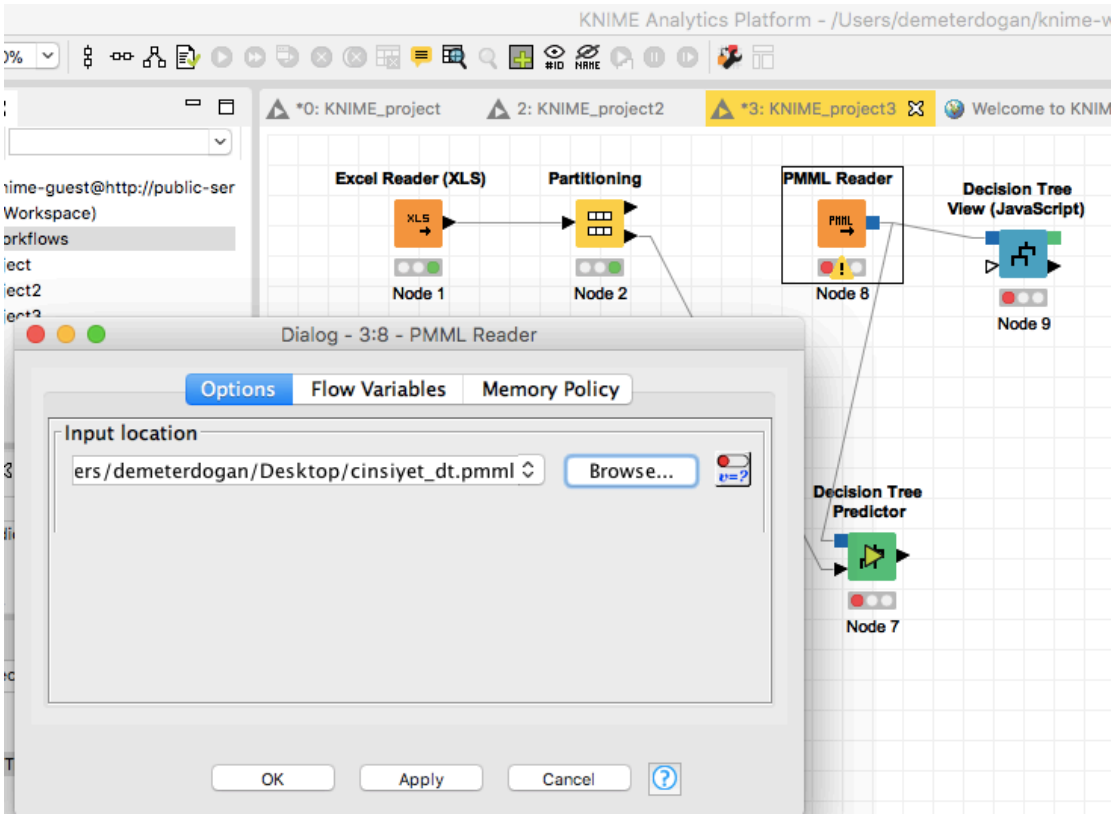
Şekil 8.5.2, decision tree learner configure bölümünü göstermektedir. Class column yazan bölüm tahmin edilmesi istenen (label) bölümüdür. Burada kendi otomatik çıkmaktadır fakat bazen birden fazla seçenek olabilmektedir.

PMML writer, öğrenilen makine öğrenmesi veya istatistiksel modeli diske bir dosya olarak kaydetmeye yarar. Bu sırada da dünyaca standart hale gelmiş PMML standardını kullanır.



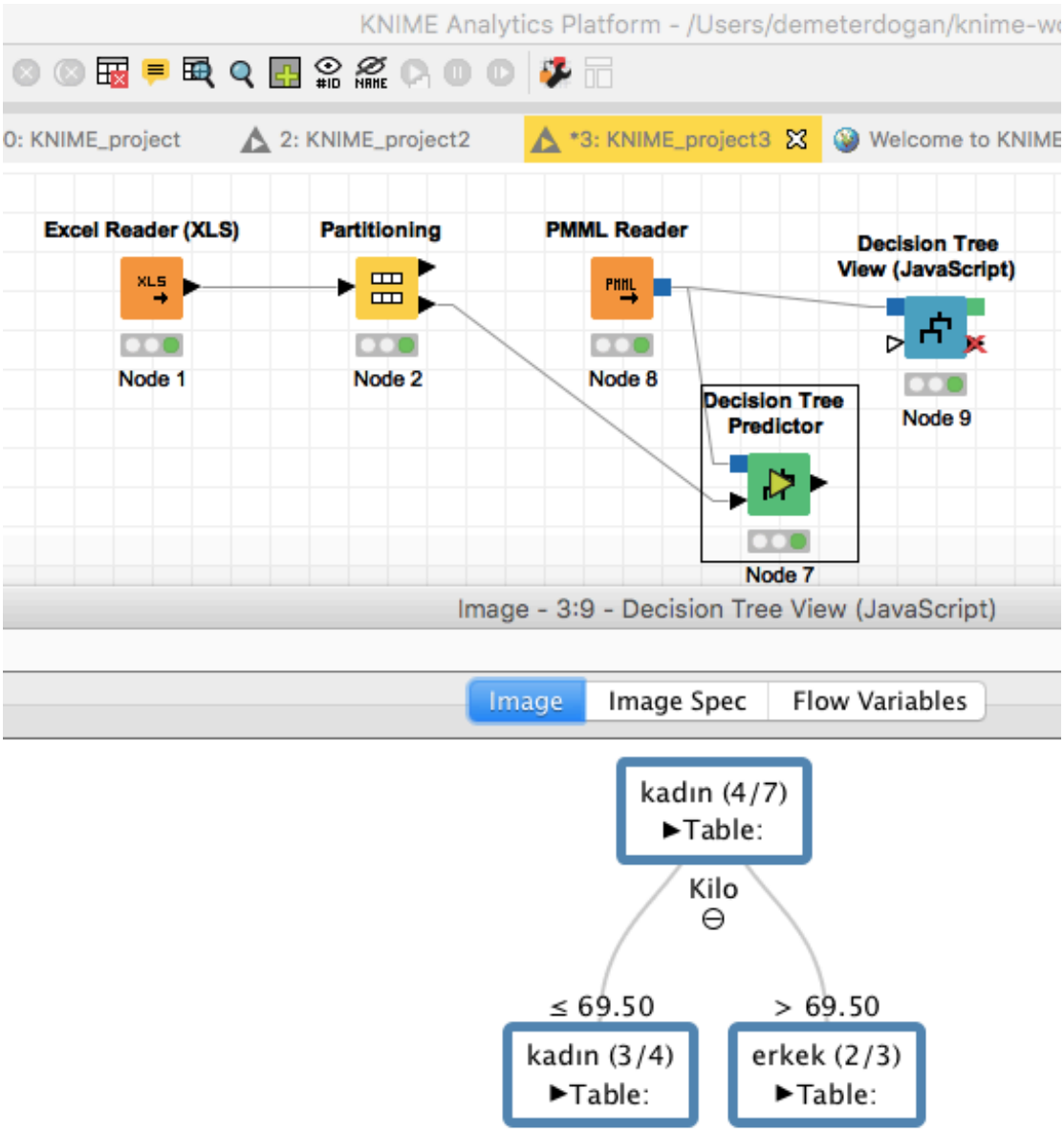
Şekil 8.5.3

Şekil 8.5.3 PMML'in ne şekilde kayıt edileceğini göstermektedir. Burada örnek olarak cinsiyet_dt ismi verildi.



Şekil 8.5.4

Şekil 8.5.4, yukarıda yazılan PMML dosyasının ardından decision tree learner'ın silinip PMML reader ile daha önce öğretilen modelin kullanılması için sisteme yüklenmesini göstermektedir. PMML configure bölümünden daha önce kaydettiğimiz cinsiyet_dt browse edilerek sisteme dahil edilmelidir.



Şekil 8.5.5

Şekil 8.5.5, program çalıştırıldıktan sonra decision tree view içerisinden girilerek kontrol edilebilinen image penceresini göstermektedir. Görüldüğü üzere kiloya göre bölünmüş ve 69.50 kg altındaysa kadın, 69.50 üzerindeyse erkek şeklinde dallandırılmıştır. Burada ¾ oranı, yani 100% garanti etmediğini göstermektedir. 69.50 altında erkek olma olasılığı da vardır.

8.6 Apriori Algoritması ve Birliktelik Kural Çıkarımı (Association Rule Mining)

Bu bölümde amaç birliktelik kural çıkarımını teorik olarak ve daha sonrasında da örnek üzerinden göstermektir.

Birliktelik kural çıkarımı gerçek hayatta normalde kampanyalarda kullanılır. Bir ürün alan bir kişiye diğer ürünü tavsiye etmektir amaç. Diğer bir seçenekler de sosyal ağlarda arkadaş tavsiyesi, eğitim tavsiyesi, olaylar arasında vb. birliktelik çıkarımı kullanılabilir. Algoritmalardan apriori ve FP-growth'dan da bahsedilecektir.

Burada en klasik problem olarak fiş üzerinden hangi ürünleri aldıkları incelenerek elde edilmiş veriler ile işlemler yapılacaktır.

Örnek olarak;

M yazılanlar müşterileri, sayılar ise müşterinin aldığı ürünleri temsil etmektedir. Bu değerler numerik değil nominal değerlerdir. Yani örneğin 1 dometesi, 2 elmayı temsil edebilmektedir.

M1: [1,2,3,4]

M2: [2,5,3]

M3: [1,5]

M4: [2]

M5:[1,2]

Öncelikle frekans tablosu çıkarılmalı. Burada tüm ürünler yazılmalı.

1 numaralı ürün toplamda 3 kez satılmış, 2 numaralı ürün 4 kez satılmış, 3 numaralı 3 vb şekilde oluşturulmalı.

Tek ürün frekansı (histogramı):

1→3

2→4

3→3

4→1

5→2 şeklinde oluşturulur.

İkili ürün frekansı (histogramı):

Burada ilişki bidirectional (iki yönlü) yani (1,2) yazıldığı zaman (2,1) yazılmasına gerek yoktur. Yani 1. Ve 2. Ürünlerin birlikte satılması anlamına gelmektedir.

(1,2) -2 birinci ürünün ikinci ürün ile birlikte satımı ve bu satın alım 2 kez yapılmış

(1,3) -2 birinci ürünün üçüncü ürün ile birlikte satımı ve bu 2 kez yapılmış

(1,4) -1 birinci ürünün dördüncü ürün ile birlikte satımı ve bu 1 kez yapılmış

(1,5) -1 birinci ürünün beşinci ürün ile birlikte satımı ve bu 1 kez yapılmış

(2,3) -2 ikinci ürünün üçüncü ürün ile birlikte satımı bu 2 kez yapılmış

(2,4) -1 ikinci ürünün dördüncü ürün ile birlikte satımı ve bu 1 kez yapılmış

(2,5) -1 ikinci ürünün beşinci ürün ile birlikte satımı ve bu 1 kez yapılmış

(3,4) -1 üçüncü ürünün dördüncü ürün ile birlikte satımı ve bu 1 kez yapılmış

(3,5) -2 üçüncü ürünün beşinci ürün ile birlikte satımı bu 2 kez yapılmış

(4,5) -1 dördüncü ürünün beşinci ürün ile birlikte satımı ve bu 1 kez yapılmış

Bu her problemde iki yönlü olmaz. Örneğin kahve almış birinin kahveyi beğendikten sonra o kahveden satın alması durumu bu örnekten farklıdır. Zaman kavramı yoktur. Hizmet eş zamanlı alında şeklinde düşünülmektedir. Ayrıca burada ürünlerin alınan miktarları eşit olarak düşünülmektedir.

Tablodan bir kez satılan ürünler ya da hiç birlikte satılmayan ürünler kaldırılabilir. (1,4) (2,4) (3,4) (4,5) bu ürünler birlikte bir kez satıldığı için tablodan silinebilir. Burada örnek olması açısından tablodan silinmemiştir. (1,2) (1,3) (2,3) (3,5) ürünleri birlikte aynı sayıda yani 2şer kez satılmış. Bu yüzden;

1→2

1→3

Burada 1 numaralı ürünü alan müşteriye 2 numara ya da 3 numaralı ürünler tavsiye edilebilir.

Üçlü ürün frekansı tablosu:

(1,2,3) → 1 Bu üç ürün bir kez birlikte satılmış

(1,3,5) → 1 Bu üç ürün bir kez birlikte satılmış

(2,3,5) → 1 Bu üç ürün bir kez birlikte satılmış

Aynı şekilde 4 lü ürün frekansı çıkarılabilir. Burada üstünlük elde edilebilecek satış sayıları olmadığı için hepsi aynı şekilde üçlü ya da ikili satış için önerilebilir. Burada kullanılan apriori algoritmasıdır.

8.7 FP- Growth Algoritması ve Birliktelik Kural Çıkarımı

Bu bölümde amaç FP growth algoritmasını göstermek ve apriori algoritması gibi örneğini göstermektir.

Yukarıdaki bölümde kullanılan örnek bu bölümde de kullanılacaktır.

M yazılanlar müşterileri, sayılar ise müşterinin aldığı ürünleri temsil etmektedir. Bu değerler numerik değil nominal değerlerdir. Yani örneğin 1 dometesi, 2 elmayı temsil edebilmektedir.

M1: [1,2,3,4]

M2: [2,5,3]

M3: [1,5]

M4: [2]

M5:[1,2]

Tek ürün frekansı (histogramı):

1-3

2-4

3-3

4-1

5-2 şeklinde oluşturulur.

1→3 birinci üründen 3 kez satılmış

2→4 ikinci üründen 4 kez satılmış

3→3 üçüncü üründen 3 kez satılmış

4→1 dördüncü üründen 1 kez satılmış

5→2 beşinci üründen 2 kez satılmış şeklinde oluşturulur.

Daha sonra;

(2,4)

(1,3)

(3,3)

(5,2)

(4,1)

şeklinde satış sayısına göre sıralanır. 1. Ve 3. Ürünlerin satış sayıları aynı olduğu için hangisinin önce yazıldığının önemi yoktur. Sonuç her şekilde aynı olacaktır.

Satış sayısına göre müşteri listelerin önceliklendirilmiş satış listeleri şu şekildedir:

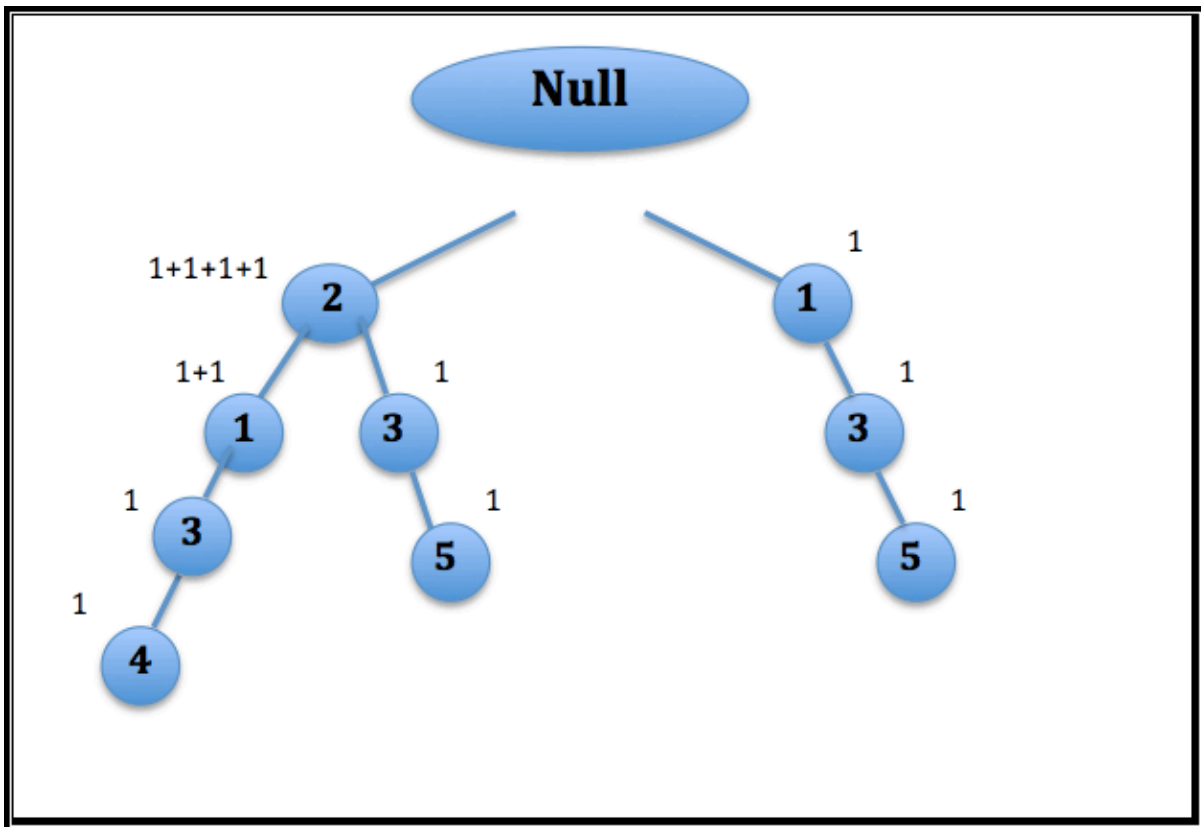
M1: [2,1,3,4]

M2: [2,3,5]

M3: [1,3,5]

M4: [2]

M5: [2,1]



Şekil 8.7.1

Şekil 8.7.1 müşterilerin alış veriş listesine göre oluşturulmuş ağacı göstermektedir. 2.üründen totalde 4 kez alınmış ve 2. Ürünü alanlar 1. Ürünü totalde 2 kez almış vb. şekilde oluşturularak devam etmektedir. Burada yukarıdaki dizilimin aksine sıra önemlidir.

Ağaçta kesme işlemi uygulanabilir. Örneğin 30% (0.3) oranında satılmış olması gibi minimum değer belirterek ağaç budanabilir. 5 ürün olduğu için 0.3x5=1.5 bunu yaklaşık olarak 2 şeklinde

alınırsa yani en az 2 ürün satılmış olunsun denirse, bu durumda ağaçtaki 4 silinmelidir. Yani ağacın en altındaki dalına ihtiyaç yoktur.

Şimdi de en düşük frekanslıdan en yüksek frekanslıya doğru gidilecek. 4. Üründen 2. Ürüne doğru gidilecek fakat 2. Ürün dahil edilmeyecektir. 4 e giden yolu bulmak gerekmektedir.

Conditional Based Pattern ismi verilen bu sıralama aşağıdaki gibi gösterilebilir:

4→ (2,1,3:1) Burada 4. Ürünü alan tek kişi olduğu için tek gidiş yolu vardır o da; 2-1-3-4. Ürünleri sırasıyla almaktır.

Ayrıca 2:1, 1:1, 3:1

2'den 1 sonra 1'den 1 ve en son 3'den 1 kez alınmış. Treshold yani limit olarak 2 belirlenmişti yukarıda. Bu yüzden 2 nin altında olan (2. Ve 1. ürünler) değerler elenecektir.

5→ (2,3:1)

veya

(1,3:1)

2:1, 3:2, 1:1

Burada 5. Ürün için gidiş yolu vardır o da; 2-3 veya 1-3'tür ve birer kez gidilmiştir.

2'den 1 sonra 1'den 1 ve en son 3'den 1 kez alınmış. Treshold yani limit olarak 2 belirlenmişti yukarıda. Bu yüzden 2 nin altında olan (2. Ve 1. ürünler) değerler elenecektir.

3→ (2,1:1)

→ (2:2)

→ (1:1)

2'den 1 sonra 1'den 1 ve en son 3'den 1 kez alınmış. Limit olarak 2 belirlenmişti yukarıda. Bu yüzden 2 nin altında olan (1. Ürün) değerler elenecektir.

1→ (2:2) Burada 2 satış yapıldığı için eleme yapılmayacaktır.

Sonuç olarak;

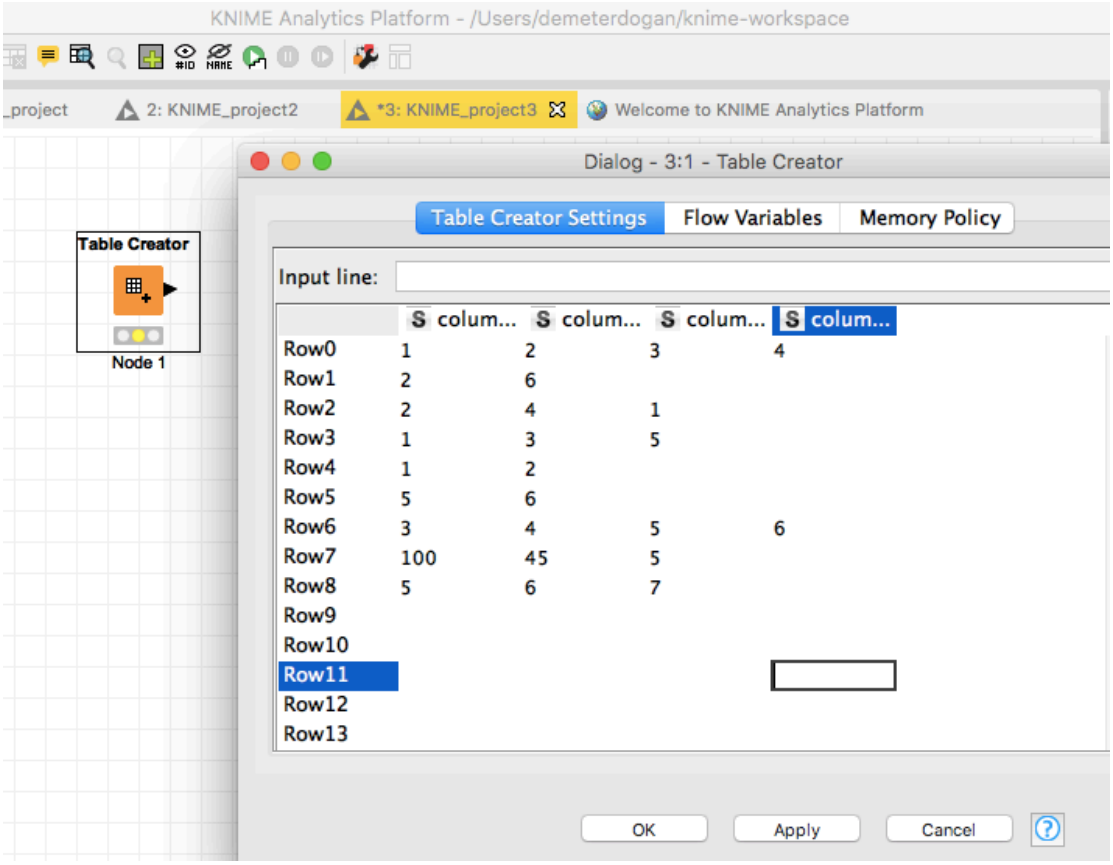
5 → 3:2

3→ 2:3

1→ 2:2 kalmıştır. Kritik ürün olarak bu ürünler denebilir.

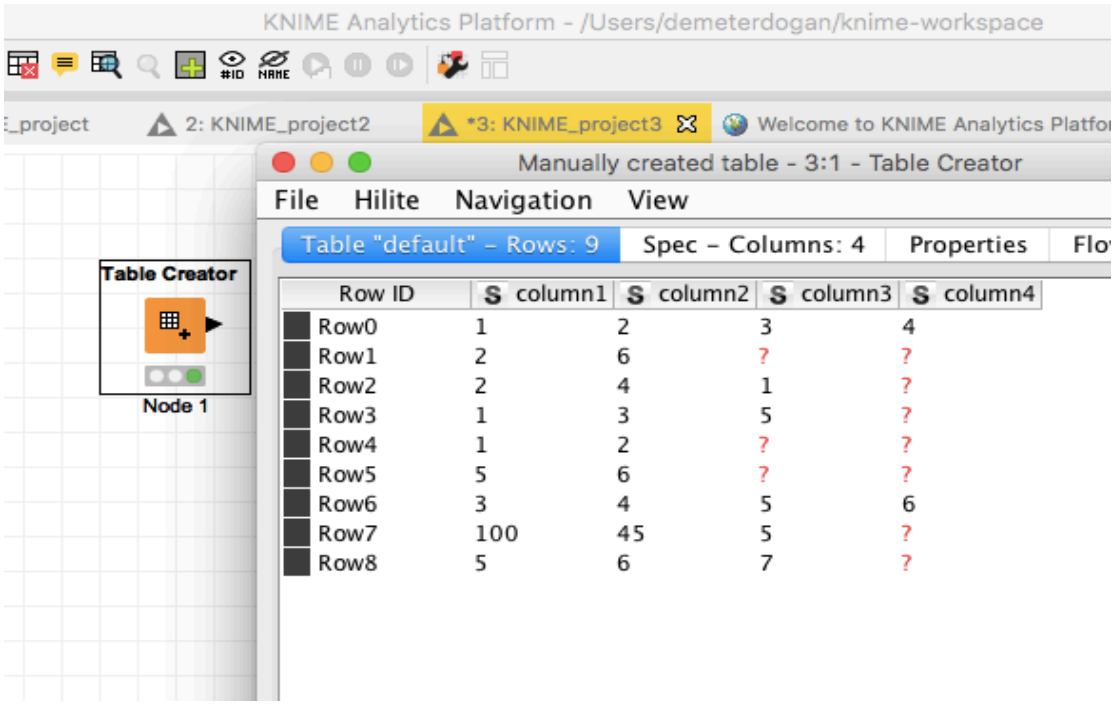
8.8. Knime Üzerinden ARM (Association Rule Mining) Uygulaması

Bu bölümde amaç, yukarıdaki bölümlerde verilen ARM teorisinin uygulamasını bu bölümde göstermektir.



Şekil 8.8.1

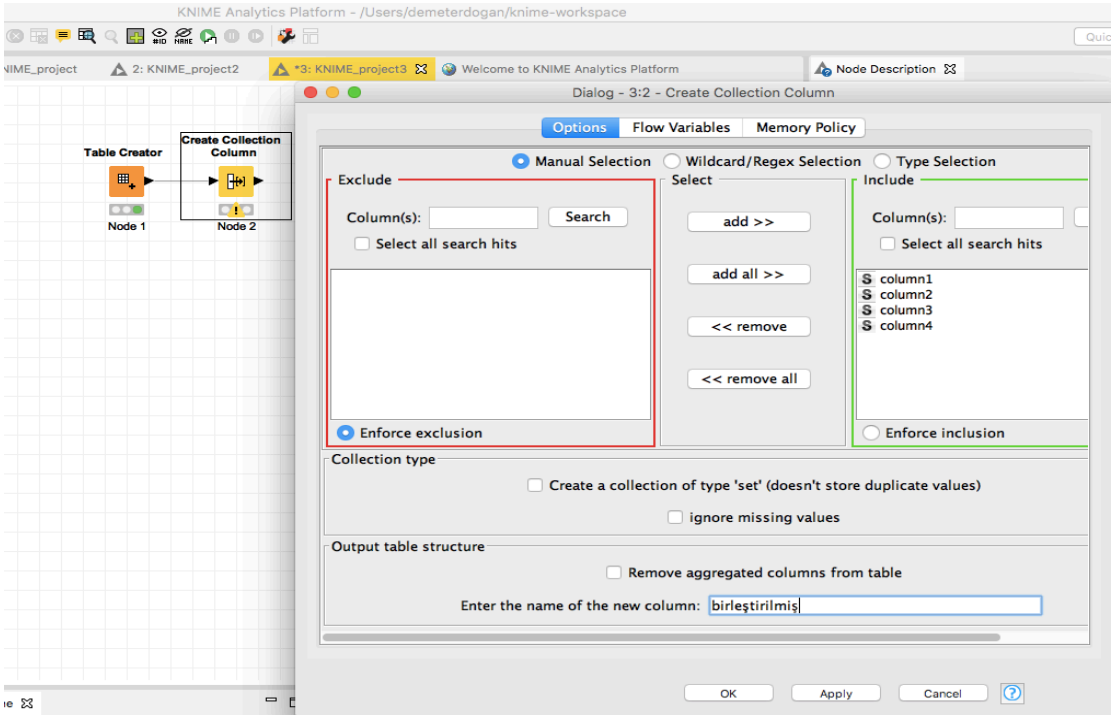
Şekil 8.8.1, table creator kullanılarak oluşturulan kolonları ve altındaki değerleri de ürün kodlarını göstermektedir. Örneğin Row0 birinci müşteriyi, 1 sütü, 2, bebek mamasını vb. şekilde düşünülebilir.



Şekil 8.8.2

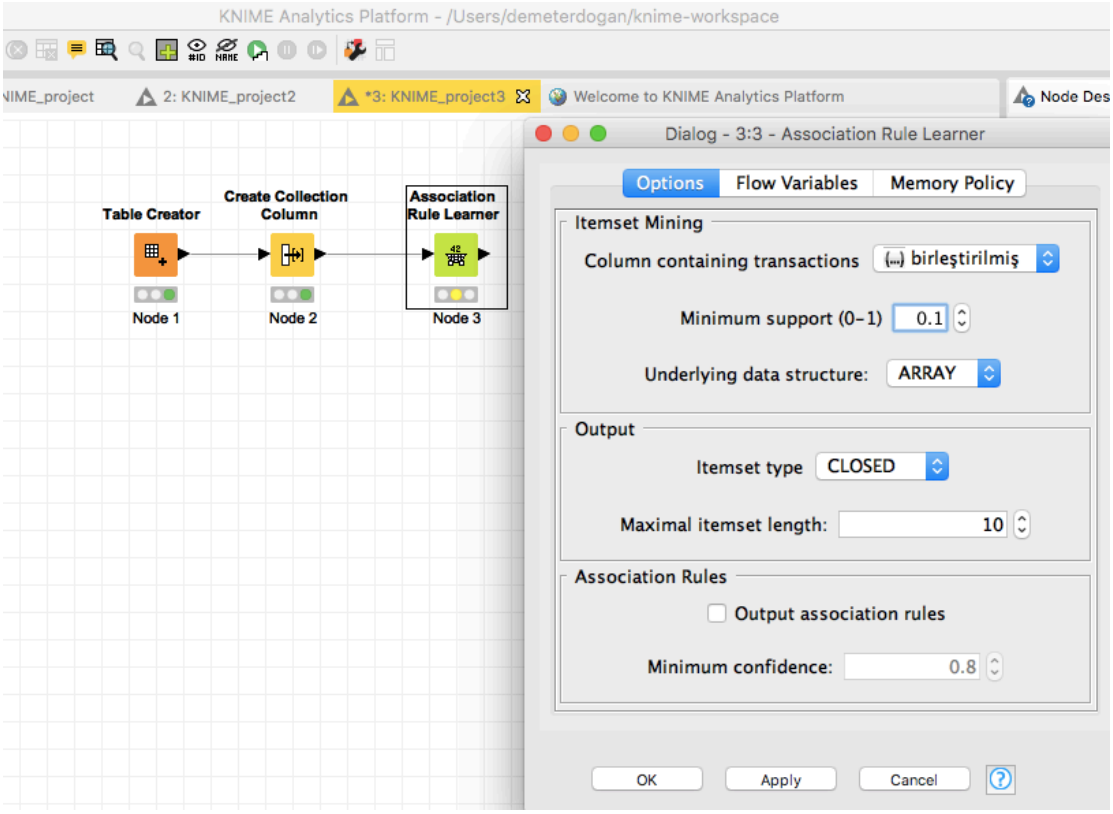
Şekil 8.8.2, program çalıştırıldıktan sonra table creator içerisinde oluşan oluşan created table'ı göstermektedir. Knime bazı kolonlarda ? işaretini kendi koymaktadır nedeni veri girilirken oraların boş bırakılmasıdır.

ARM operatörü çoklu kolonda çalışamaz bu yüzden tek sıraya indirgeyebilmek için create collection column operatörü kullanılmalıdır.



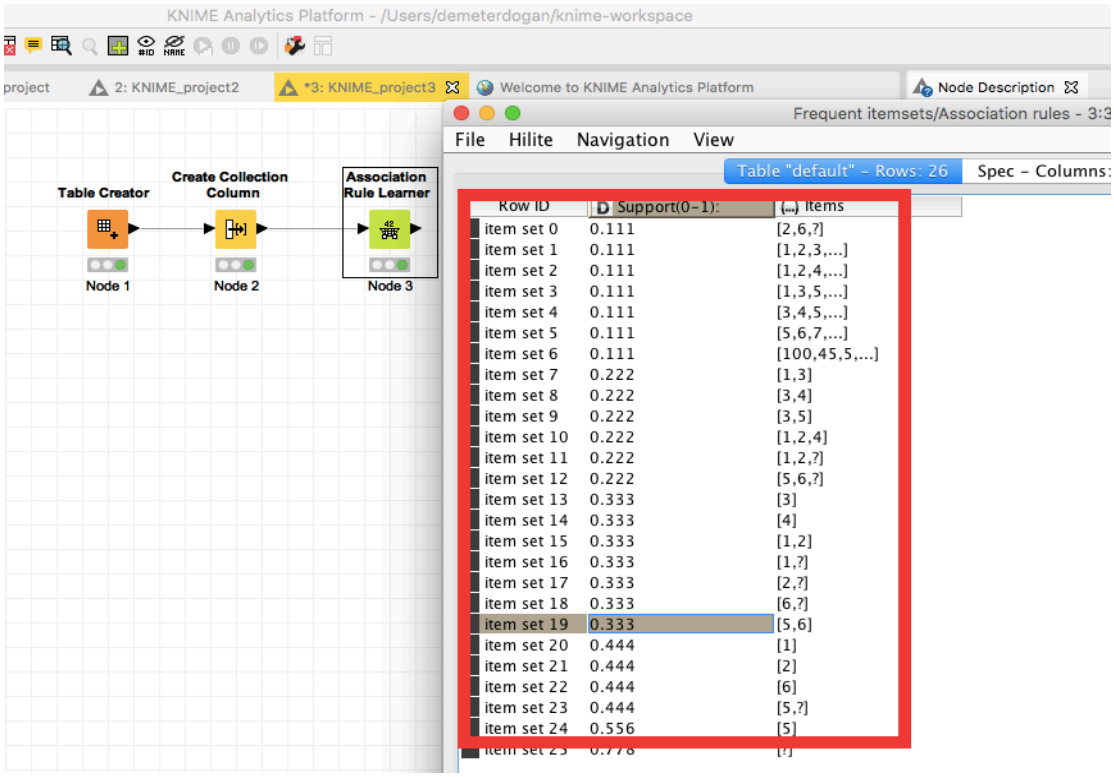
Şekil 8.8.3

Şekil 8.8.3, sisteme create collection column operatörünün eklenmesini ve oluşturulacak yeni kolonun isminin "birleştirilmiş" olacağını göstermektedir. Bu birleştirilmiş isim örnek olması açısından verilmiştir. Oraya herhangi başka bir isim de yazılabilir.



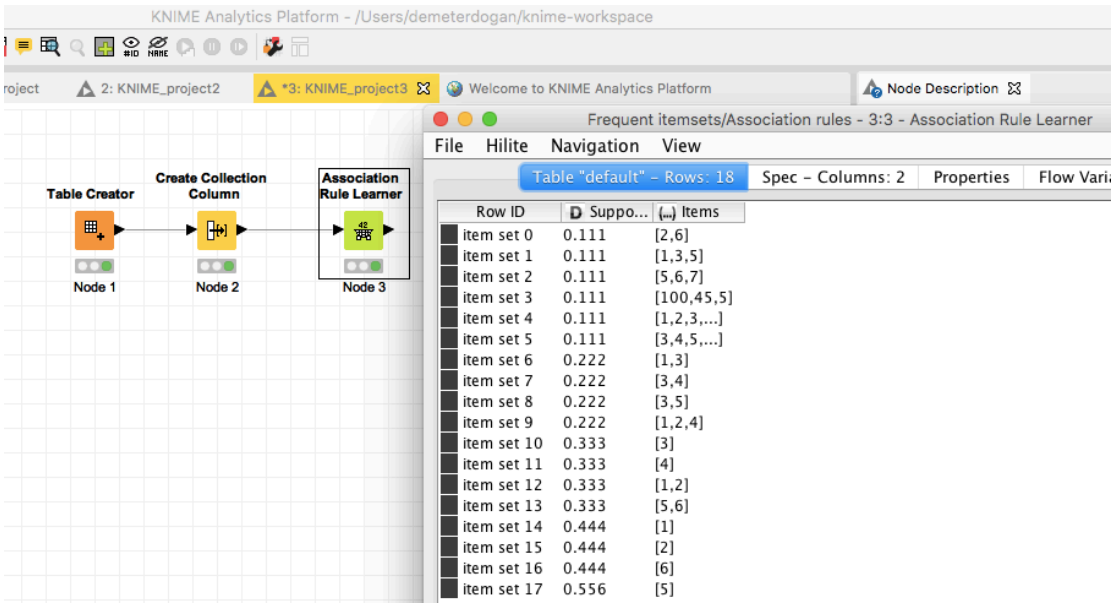
Şekil 8.8.4

Şekil 8.8.4'te de görüldüğü üzere, mümkün olduğunca fazla ürün alınabilsin diye association rule learner operatöründe configure ederken minimum support için 0.1 değeri verildi.



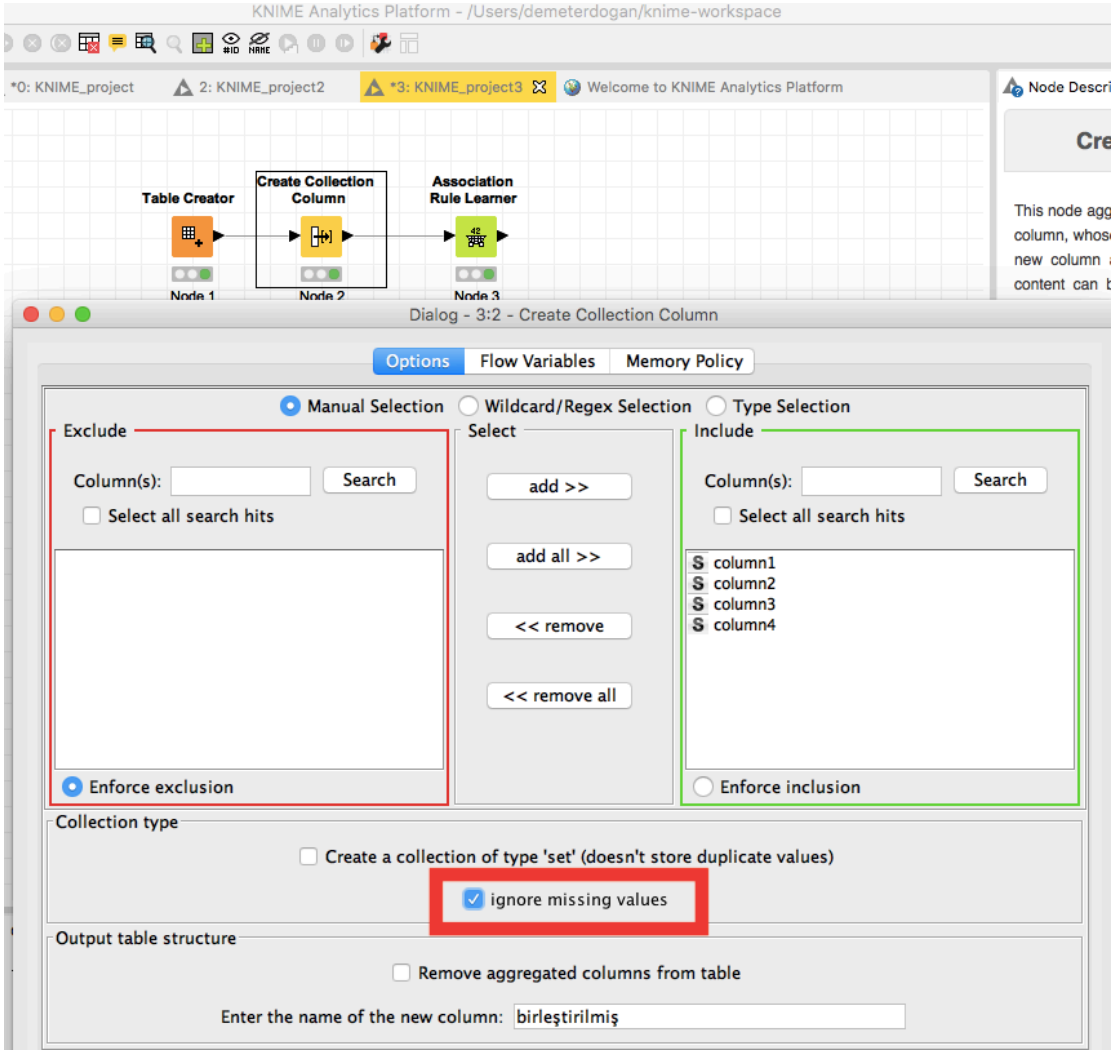
Şekil 8.8.5

Şekil 8.8.5 program çalıştırıldıktan sonra elde edilen sonuç penceresini göstermektedir. Görüldüğü üzere daha önce girilmemiş verilerin yerine soru işareti konmuş ve bunlar sanki bir ürünmüş gibi algılanmıştır. İtem set 25 de yani en alttaki sırada 0.778 oranında satış ile soru işareti sanki bir ürün kodu gibi en çok orana sahip olmuştur.



Şekil 8.8.7

Şekil 8.8.7 program çalıştırıldıktan sonraki çıktıyı göstermektedir. Soru işaretleri bir önceki şekildeki anlatıldığı gibi giderildiği için programda artık sadece öncden girilmiş ürün listelerine göre sıralama yapılmıştır. Yukarıdaki Şekil 8.8.4'te 0.1 olarak girilen support value değeri yüzünden 0.1'in altındaki değere sahip olan satışlar sistemden elenmiştir. Sonuca göre örneğin, kampanya yapılacaksa item set 15 'te görüldüğü gibi 5 ve 6. Ürünler birlikte satılması için bir paket yapılabilir.



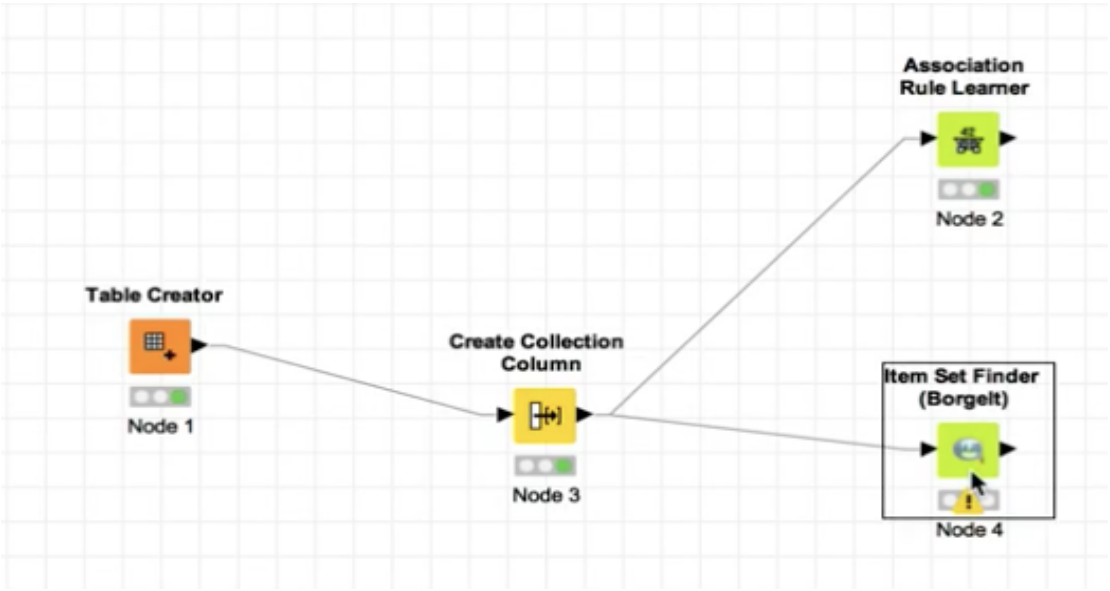
Şekil 8.8.6

Şekil 8.8.6 create collection column için configure ederken yazılmamış hücreler için ignore (yoksay) seçeneğinin seçilmesini göstermektedir. Böylece bir önceki şekildeki sonuçta çıkan problem giderilecektir.

Bu örnekte bir üründen sadece bir kez alınmış gibi davranılmıştır fakat daha fazla alındıysa örneğin Row0 diye geçen birinci müşteri 3. Üründen 2 tane almışsa Şekil 8.8.1 de 5. Kolon eklenip oraya tekrardan 3 yazılmalıdır. Ve Şekil 8.8.6 da görülen create a collection of type 'set' seçeneği seçilmelidir. Bu sayede Knime 3. Ürünleri bir küme yaparak ona göre ağırlık hesaplaması oluşturmaktadır.

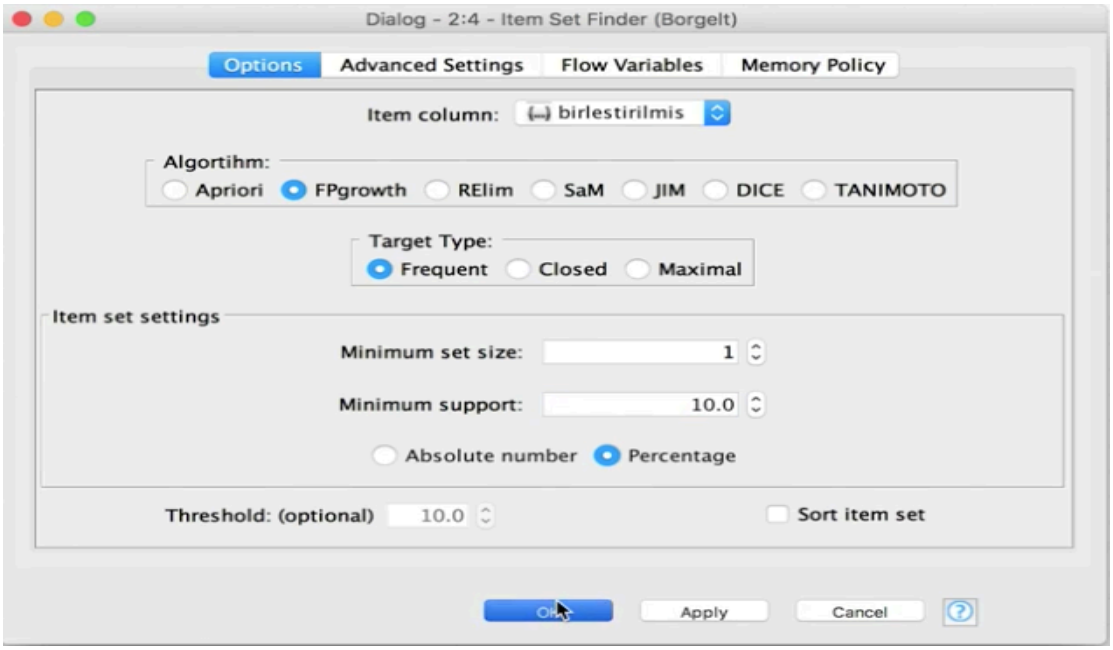
8.9. Knime Üzerinden Apriori veya FPGrowth Algoritmaları

Bu bölümde amaç association rule mining ile ilgili bir örnek ile daha detay işlem yaptırılması amaçlanmıştır. Bir önceki bölümde Şekil 8.8.4'te oluşturulan table creator, create collection column ve association rule learner bağlantıları ile başlanacaktır.



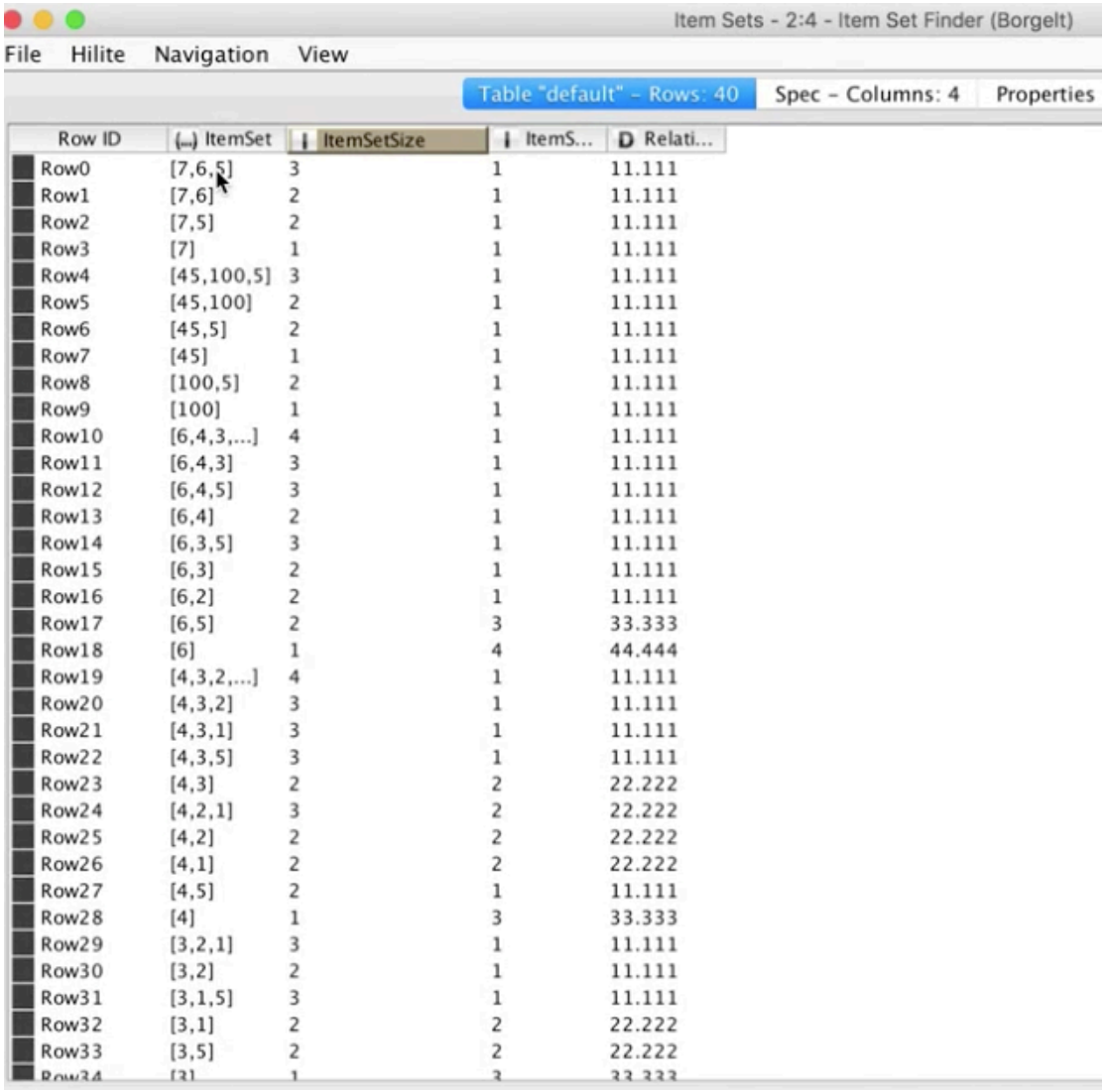
Şekil 8.9.1

Şekil 8.91 'de item set finder (borgelt) operatörünün sisteme bağlantısı gösterilmektedir.



Şekil 8.9.2

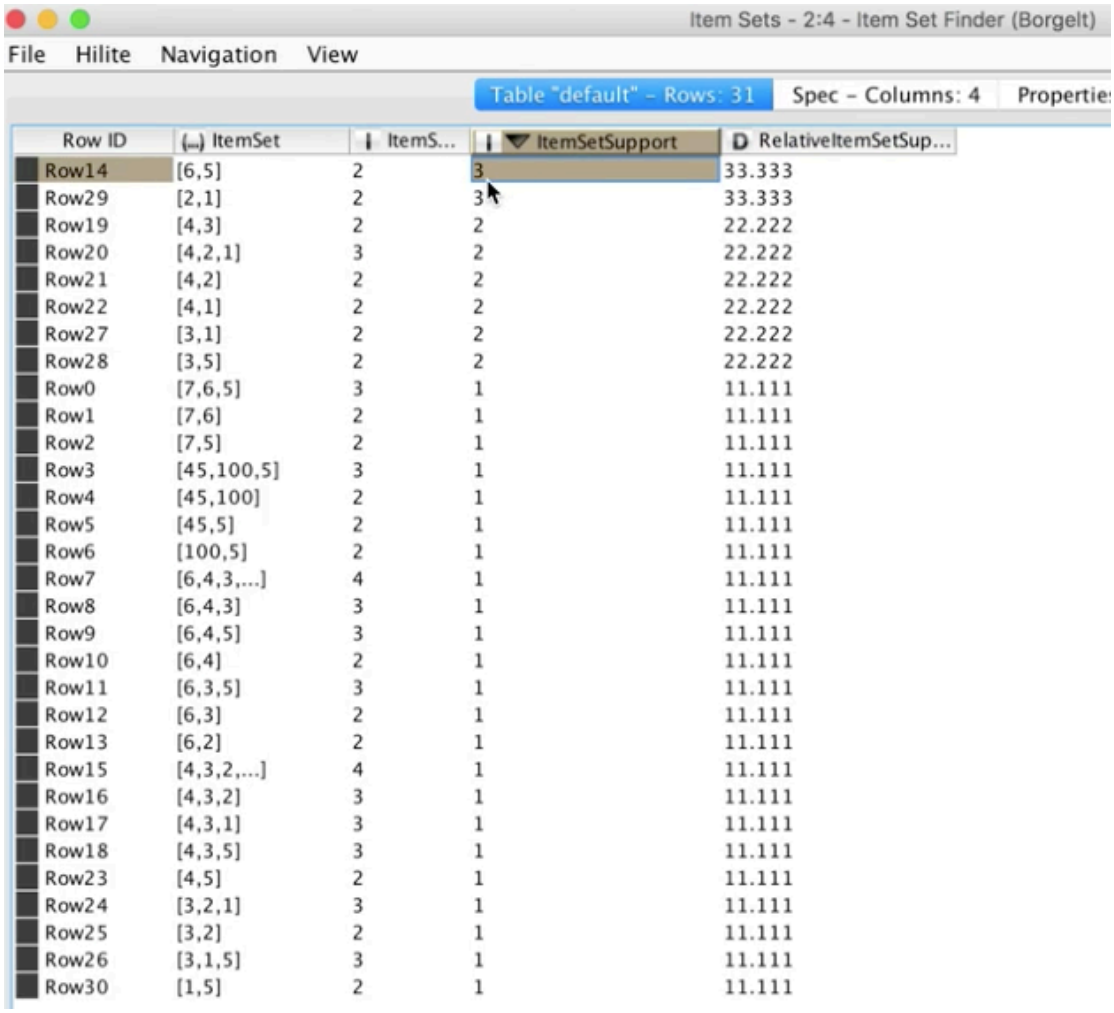
Şekil 8.9.2 item operatörünün configure bölümündeki ayarlamayı göstermektedir. FPgrowth algoritması seçilmiş ve bu şekilde program çalıştırılmıştır.



Row ID	ItemSet	ItemSetSize	ItemS...	Relati...
Row0	[7,6,5]	3	1	11.111
Row1	[7,6]	2	1	11.111
Row2	[7,5]	2	1	11.111
Row3	[7]	1	1	11.111
Row4	[45,100,5]	3	1	11.111
Row5	[45,100]	2	1	11.111
Row6	[45,5]	2	1	11.111
Row7	[45]	1	1	11.111
Row8	[100,5]	2	1	11.111
Row9	[100]	1	1	11.111
Row10	[6,4,3,...]	4	1	11.111
Row11	[6,4,3]	3	1	11.111
Row12	[6,4,5]	3	1	11.111
Row13	[6,4]	2	1	11.111
Row14	[6,3,5]	3	1	11.111
Row15	[6,3]	2	1	11.111
Row16	[6,2]	2	1	11.111
Row17	[6,5]	2	3	33.333
Row18	[6]	1	4	44.444
Row19	[4,3,2,...]	4	1	11.111
Row20	[4,3,2]	3	1	11.111
Row21	[4,3,1]	3	1	11.111
Row22	[4,3,5]	3	1	11.111
Row23	[4,3]	2	2	22.222
Row24	[4,2,1]	3	2	22.222
Row25	[4,2]	2	2	22.222
Row26	[4,1]	2	2	22.222
Row27	[4,5]	2	1	11.111
Row28	[4]	1	3	33.333
Row29	[3,2,1]	3	1	11.111
Row30	[3,2]	2	1	11.111
Row31	[3,1,5]	3	1	11.111
Row32	[3,1]	2	2	22.222
Row33	[3,5]	2	2	22.222
Row34	[3]	1	3	33.333

Şekil 8.9.3

Şekil 8.9.3, program çalıştırıldıktan sonraki item sets penceresini göstermektedir. Item set size ürünlerin kaç elemanlı satıldığını, item set support value ise destek değerini yani ne kadar önemli olduğunu belirten değerdir.



Row ID	ItemSet	ItemS...	ItemSetSupport	RelativeItemSetSup...
Row14	[6,5]	2	3	33.333
Row29	[2,1]	2	3	33.333
Row19	[4,3]	2	2	22.222
Row20	[4,2,1]	3	2	22.222
Row21	[4,2]	2	2	22.222
Row22	[4,1]	2	2	22.222
Row27	[3,1]	2	2	22.222
Row28	[3,5]	2	2	22.222
Row0	[7,6,5]	3	1	11.111
Row1	[7,6]	2	1	11.111
Row2	[7,5]	2	1	11.111
Row3	[45,100,5]	3	1	11.111
Row4	[45,100]	2	1	11.111
Row5	[45,5]	2	1	11.111
Row6	[100,5]	2	1	11.111
Row7	[6,4,3,...]	4	1	11.111
Row8	[6,4,3]	3	1	11.111
Row9	[6,4,5]	3	1	11.111
Row10	[6,4]	2	1	11.111
Row11	[6,3,5]	3	1	11.111
Row12	[6,3]	2	1	11.111
Row13	[6,2]	2	1	11.111
Row15	[4,3,2,...]	4	1	11.111
Row16	[4,3,2]	3	1	11.111
Row17	[4,3,1]	3	1	11.111
Row18	[4,3,5]	3	1	11.111
Row23	[4,5]	2	1	11.111
Row24	[3,2,1]	3	1	11.111
Row25	[3,2]	2	1	11.111
Row26	[3,1,5]	3	1	11.111
Row30	[1,5]	2	1	11.111

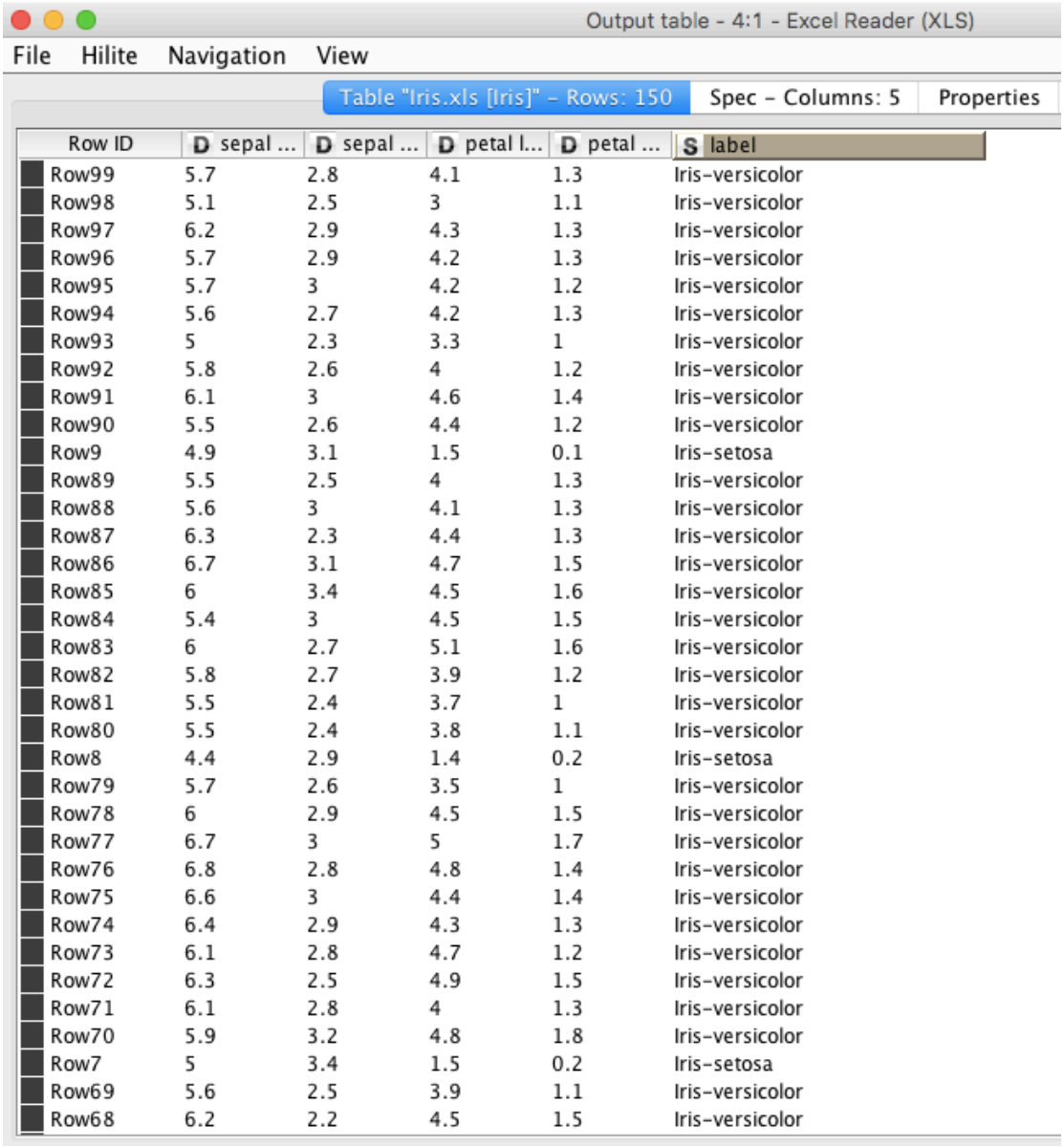
Şekil 8.9.4

Şekil 8.9.4, minimum set size 2 seçilerek program çalıştırıldığındaki sonucu göstermektedir. Bu seçim Şekil 8.9.2 de görülen pencereden yapılmaktadır. Burada en az ikili satışların listesi çıkarılmıştır.

8.10. Bölütleme (Kümeleme, Clustering) ve K-Means Algoritması

Bu bölümde amaç kümeleme kavramını, supervised-unsupervised learning kavramını tanıtmaktır.

Örnek olarak bu bölümde iris veri seti kullanılacaktır. Iris veri seti, 4 yaprak özelliği verilerek yaprak çeşidinin bulunmaya çalışıldığı meşhur bir veri setidir.



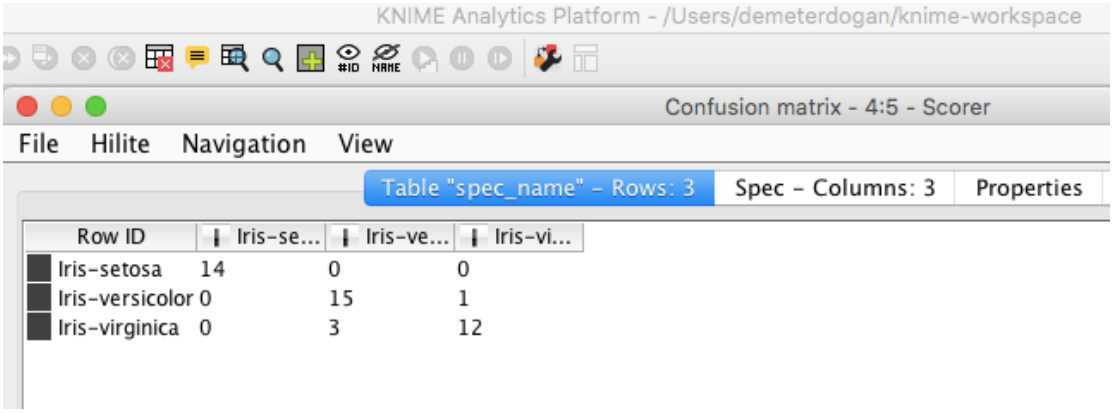
Row ID	D sepal ...	D sepal ...	D petal l...	D petal ...	S label
Row99	5.7	2.8	4.1	1.3	Iris-versicolor
Row98	5.1	2.5	3	1.1	Iris-versicolor
Row97	6.2	2.9	4.3	1.3	Iris-versicolor
Row96	5.7	2.9	4.2	1.3	Iris-versicolor
Row95	5.7	3	4.2	1.2	Iris-versicolor
Row94	5.6	2.7	4.2	1.3	Iris-versicolor
Row93	5	2.3	3.3	1	Iris-versicolor
Row92	5.8	2.6	4	1.2	Iris-versicolor
Row91	6.1	3	4.6	1.4	Iris-versicolor
Row90	5.5	2.6	4.4	1.2	Iris-versicolor
Row9	4.9	3.1	1.5	0.1	Iris-setosa
Row89	5.5	2.5	4	1.3	Iris-versicolor
Row88	5.6	3	4.1	1.3	Iris-versicolor
Row87	6.3	2.3	4.4	1.3	Iris-versicolor
Row86	6.7	3.1	4.7	1.5	Iris-versicolor
Row85	6	3.4	4.5	1.6	Iris-versicolor
Row84	5.4	3	4.5	1.5	Iris-versicolor
Row83	6	2.7	5.1	1.6	Iris-versicolor
Row82	5.8	2.7	3.9	1.2	Iris-versicolor
Row81	5.5	2.4	3.7	1	Iris-versicolor
Row80	5.5	2.4	3.8	1.1	Iris-versicolor
Row8	4.4	2.9	1.4	0.2	Iris-setosa
Row79	5.7	2.6	3.5	1	Iris-versicolor
Row78	6	2.9	4.5	1.5	Iris-versicolor
Row77	6.7	3	5	1.7	Iris-versicolor
Row76	6.8	2.8	4.8	1.4	Iris-versicolor
Row75	6.6	3	4.4	1.4	Iris-versicolor
Row74	6.4	2.9	4.3	1.3	Iris-versicolor
Row73	6.1	2.8	4.7	1.2	Iris-versicolor
Row72	6.3	2.5	4.9	1.5	Iris-versicolor
Row71	6.1	2.8	4	1.3	Iris-versicolor
Row70	5.9	3.2	4.8	1.8	Iris-versicolor
Row7	5	3.4	1.5	0.2	Iris-setosa
Row69	5.6	2.5	3.9	1.1	Iris-versicolor
Row68	6.2	2.2	4.5	1.5	Iris-versicolor

Şekil 8.10.1

Şekil 8.10.1, iris veri setinin output table yani nasıl bir veri seti olduğunu göstermektedir. Label kolonu makine öğrenmesinden sonra ulaşılmak istenilen kolondur. Burada 4 özellikten sonra yaprak çeşidinin tahmin edilmesi ulaşılmak istenilen kolon olarak yorumlanabilir. Buna gözetimli öğretim (supervised learning)

denir. Eğer etiket verilmeseydi makineye bırakılsaydı ve bu verileri kendisine göre ayrılması istenseydi clustering yapılmış olunurdur.

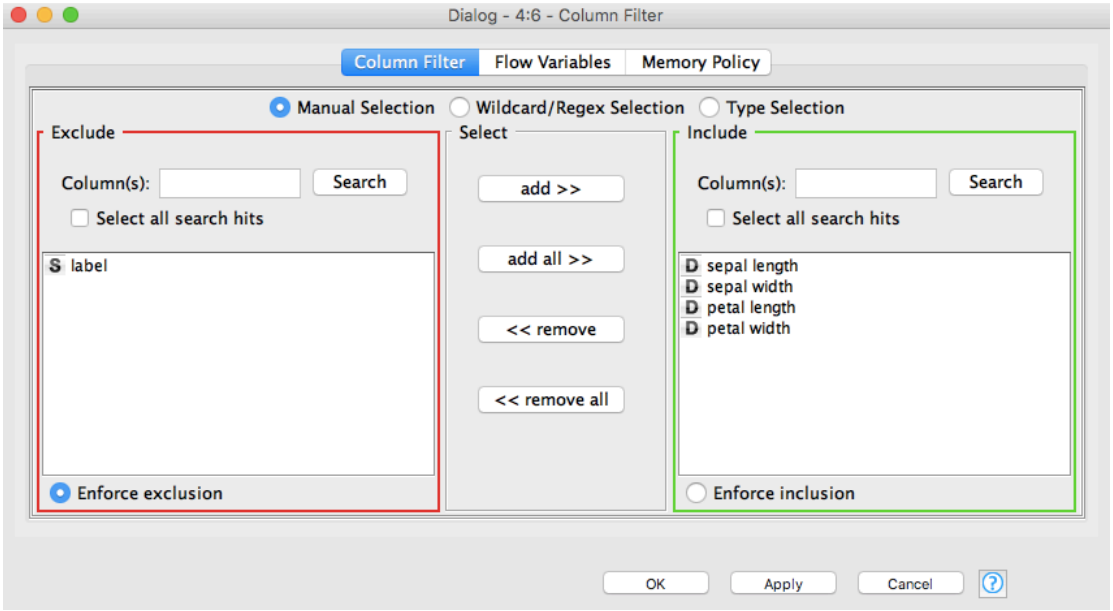
Veri öncelikle eğitim için sonra da test için partitioning bölünür.



Şekil 8.10.2

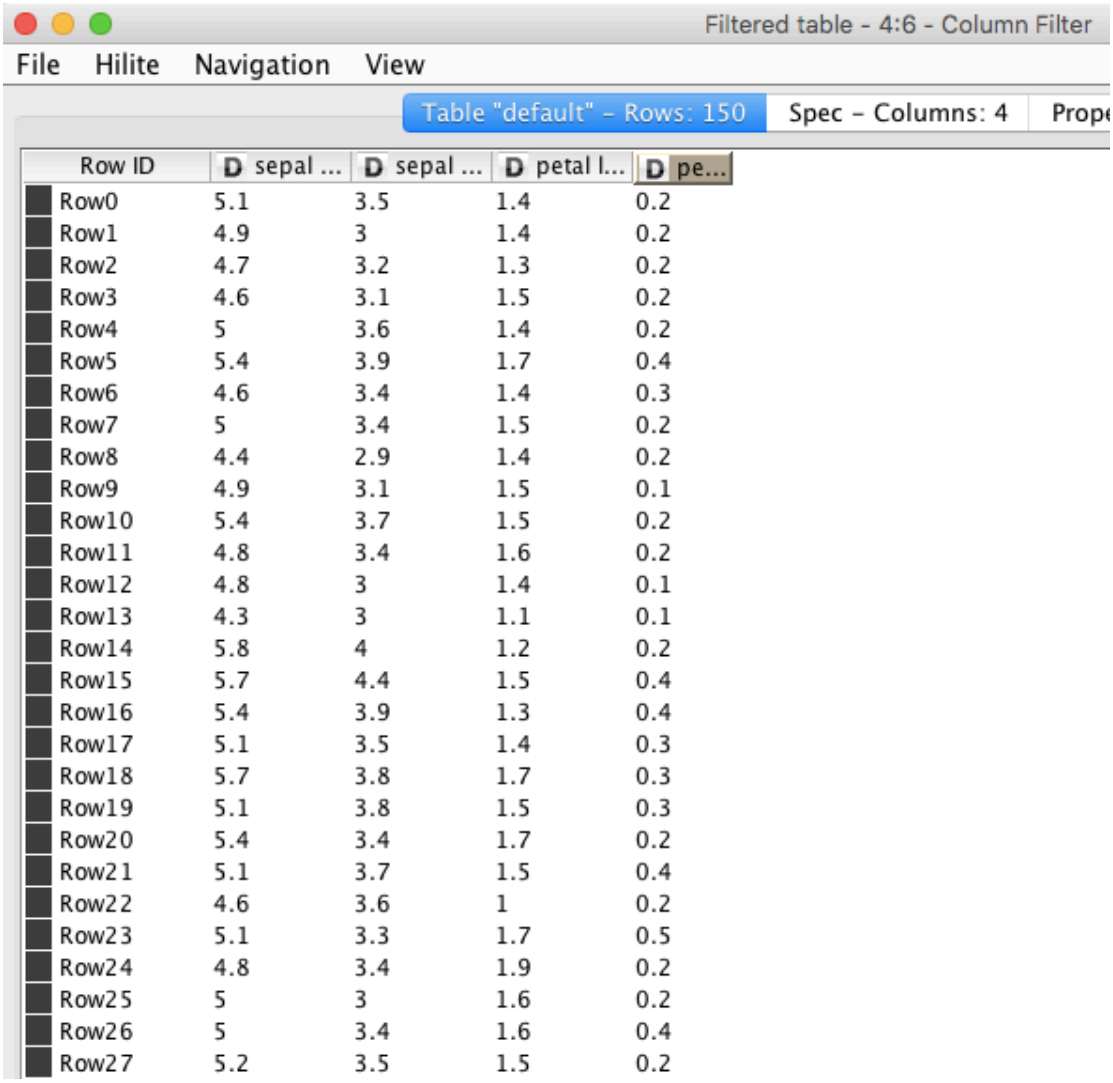
Şekil 8.10.2, decision tree learner ve decision tree predictor kullanıldıktan sonra program çalıştırıldığında scorerdaki sonuç penceresini göstermektedir. Iris setosa 14, iris vertica 15 ve iris virginica 12 tanesi doğru bilinmiştir. Bu yöntem gözetimli öğretme yöntemi kullanılarak yapılmıştır.

Gözetimli eğitimden çıkarılarak makinenin kendisinin gruplayıp sonuca ulaşabilmesi için label kolonu kaldırılacaktır. Bunun için column filter operatörü kullanılmıştır.



Şekil 8.10.3

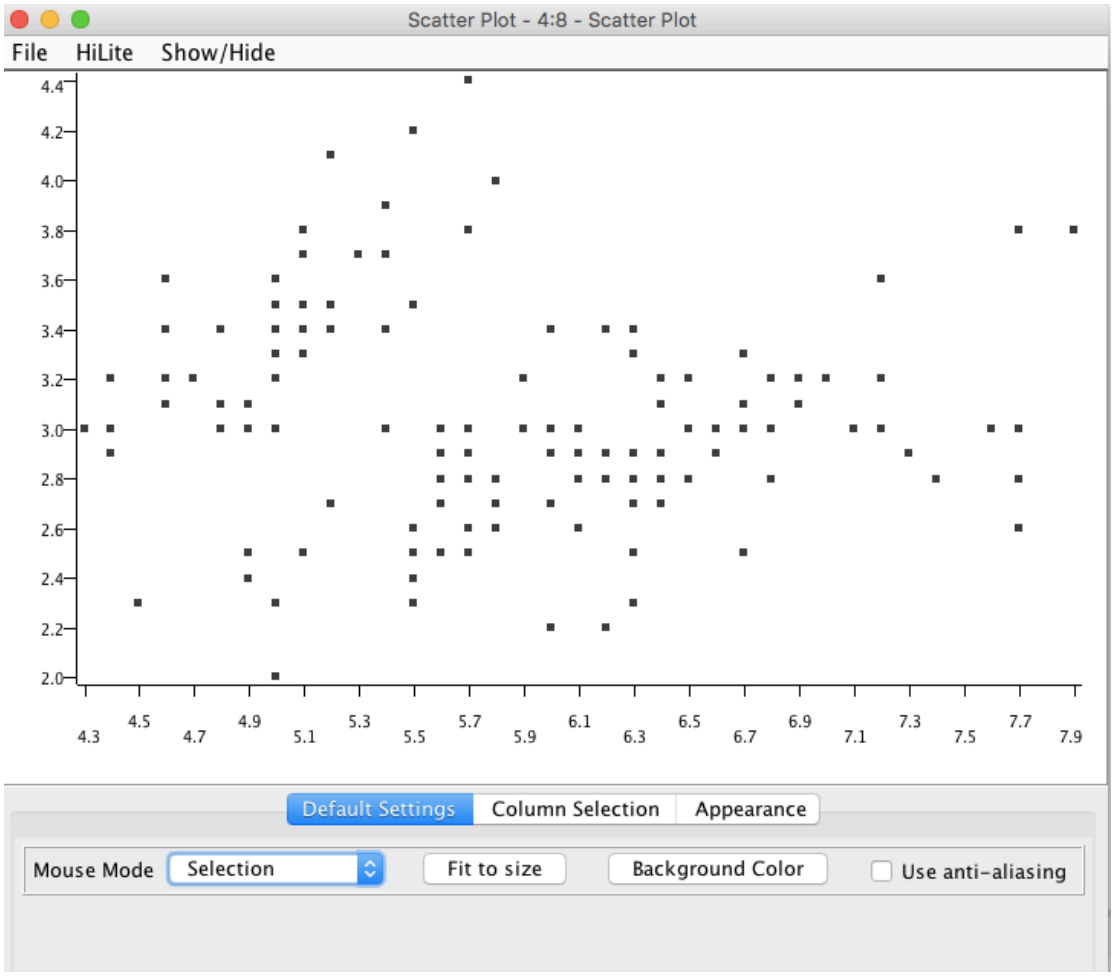
Şekil 8.10.3, column filter operatörü için yapılan configure'ü göstermektedir. Label kolonu kaldırılmak istendiği için o exclude edilmiştir.



Row ID	sepal ...	sepal ...	petal l...	pe...
Row0	5.1	3.5	1.4	0.2
Row1	4.9	3	1.4	0.2
Row2	4.7	3.2	1.3	0.2
Row3	4.6	3.1	1.5	0.2
Row4	5	3.6	1.4	0.2
Row5	5.4	3.9	1.7	0.4
Row6	4.6	3.4	1.4	0.3
Row7	5	3.4	1.5	0.2
Row8	4.4	2.9	1.4	0.2
Row9	4.9	3.1	1.5	0.1
Row10	5.4	3.7	1.5	0.2
Row11	4.8	3.4	1.6	0.2
Row12	4.8	3	1.4	0.1
Row13	4.3	3	1.1	0.1
Row14	5.8	4	1.2	0.2
Row15	5.7	4.4	1.5	0.4
Row16	5.4	3.9	1.3	0.4
Row17	5.1	3.5	1.4	0.3
Row18	5.7	3.8	1.7	0.3
Row19	5.1	3.8	1.5	0.3
Row20	5.4	3.4	1.7	0.2
Row21	5.1	3.7	1.5	0.4
Row22	4.6	3.6	1	0.2
Row23	5.1	3.3	1.7	0.5
Row24	4.8	3.4	1.9	0.2
Row25	5	3	1.6	0.2
Row26	5	3.4	1.6	0.4
Row27	5.2	3.5	1.5	0.2

Şekil 8.10.4

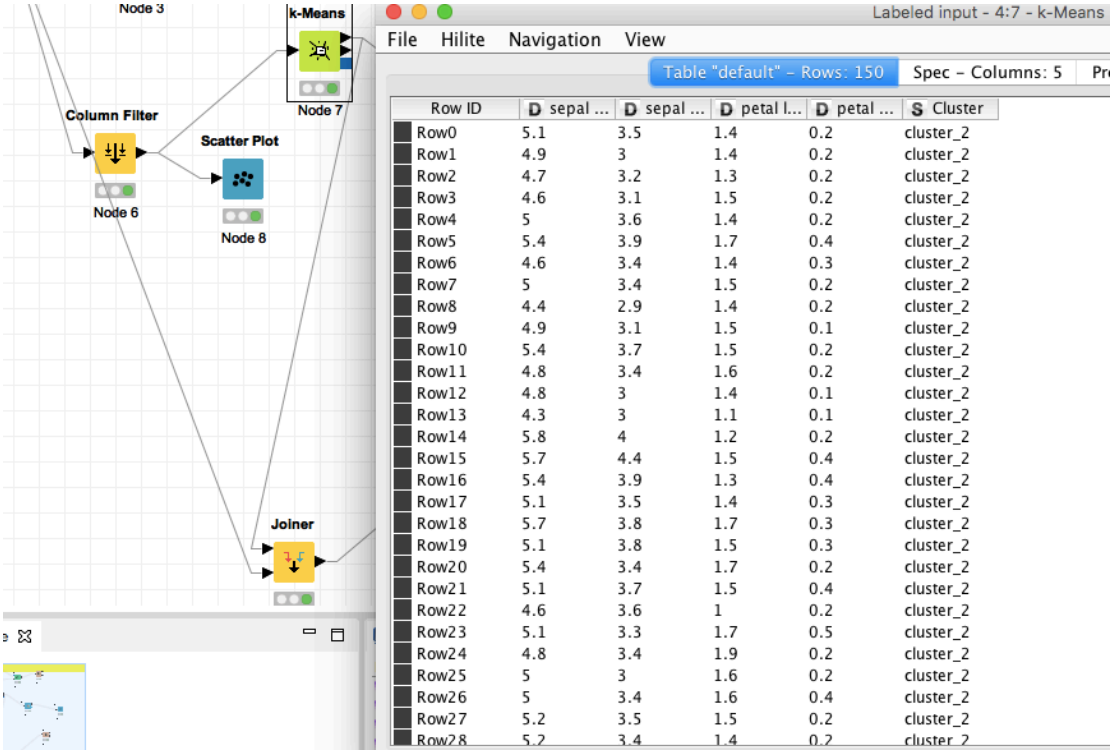
Şekil 8.10.4, label kolonu column filter node'u ile kaldırıldıktan sonraki veri setini göstermektedir. K-means algoritması kullanılacaktır. K tane orta bulur ve bu orta noktalara göre bölüt oluşturur.



Şekil 8.10.5

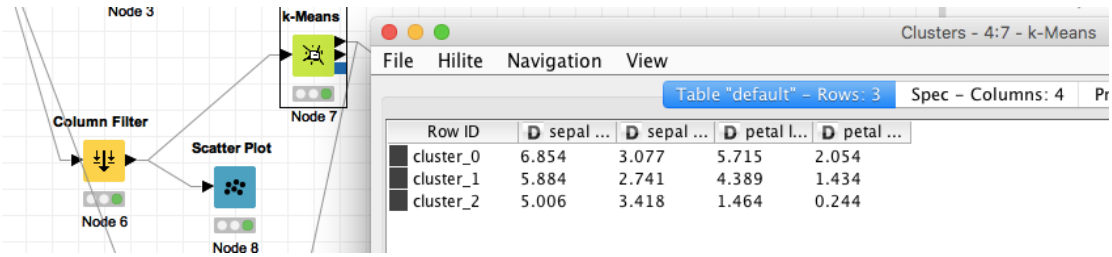
Şekil 8.10.5, verilerin uzayda iki boyutlu olarak dağıtılmış halini göstermektedir. Sisteme column filter'a scatter plot operatörü bağlanarak bu şekle ulaşılmıştır. Verilerdeki gruplaşma kolayca fark edilmektedir. Bu label etiketi bilinmeden yapılmış kümelemedir.

K-means algoritması verilere bakarak gruplamayı kendi yapacaktır. K-meanste kaç cluster olması isteniyorsa o miktarda bölüt bulur.



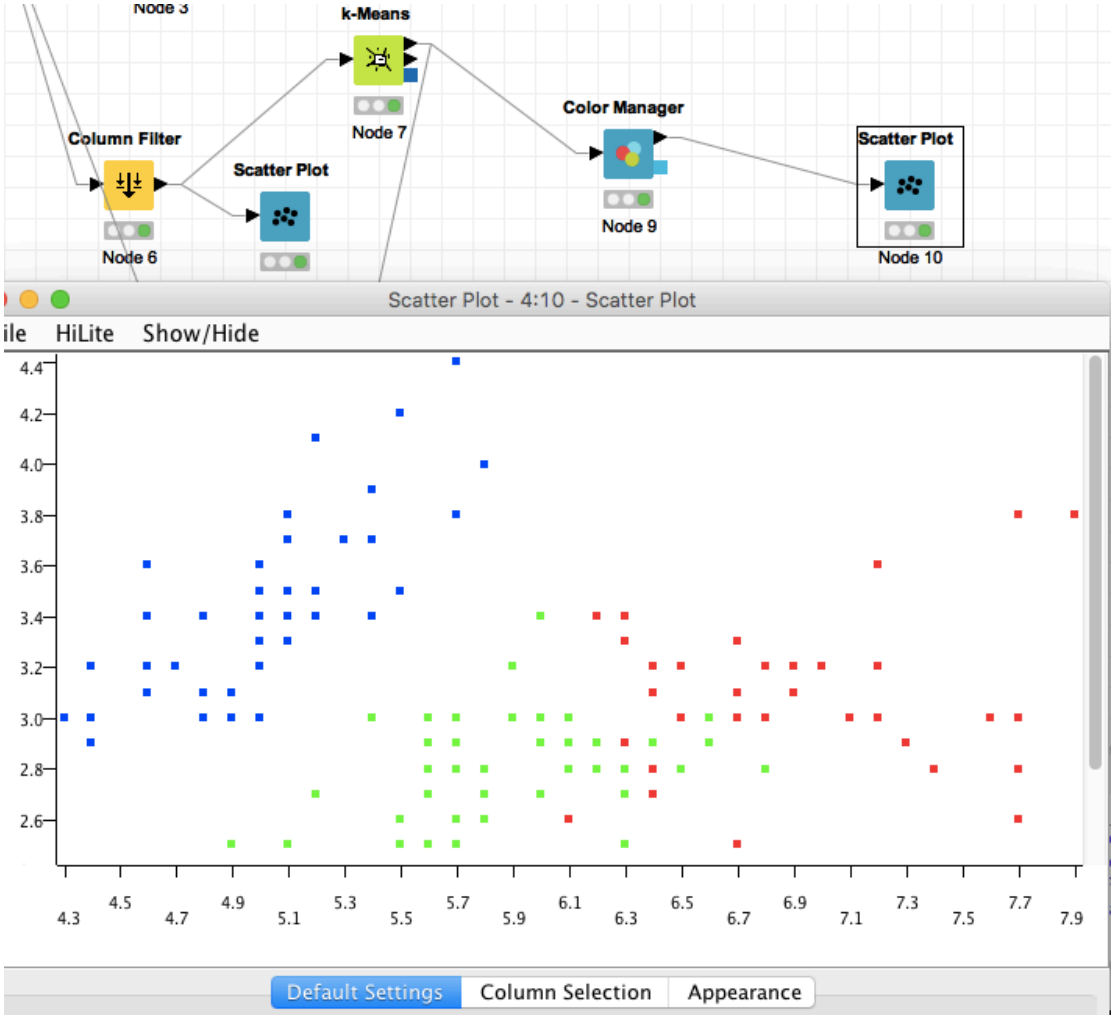
Şekil 8.10.6

Şekil 8.10.6, k-Means operatörü eklenerek oluşturulmuş labeled sonucunu göstermektedir. Cluster_0, cluster_1, cluster2 şeklinde 3 cluster oluşturulmuştur.



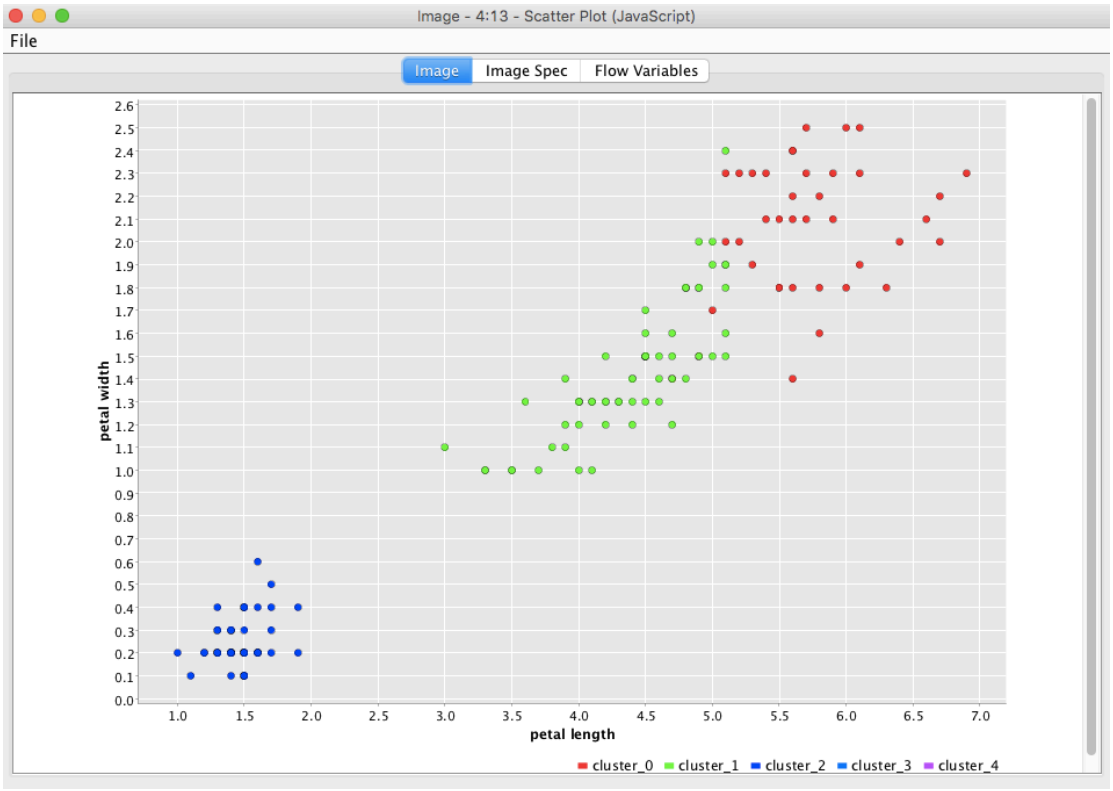
Şekil 8.10.7

Şekil 8.10.7,oluşturulan cluster'ların orta noktalarını göstermektedir. K-means ile birlikte cluster bilgisi oluşturulduğu için artık color manager operatörü ve scatter plot eklenerek bu gruplama renklendirilebilir.



Şekil 8.10.8

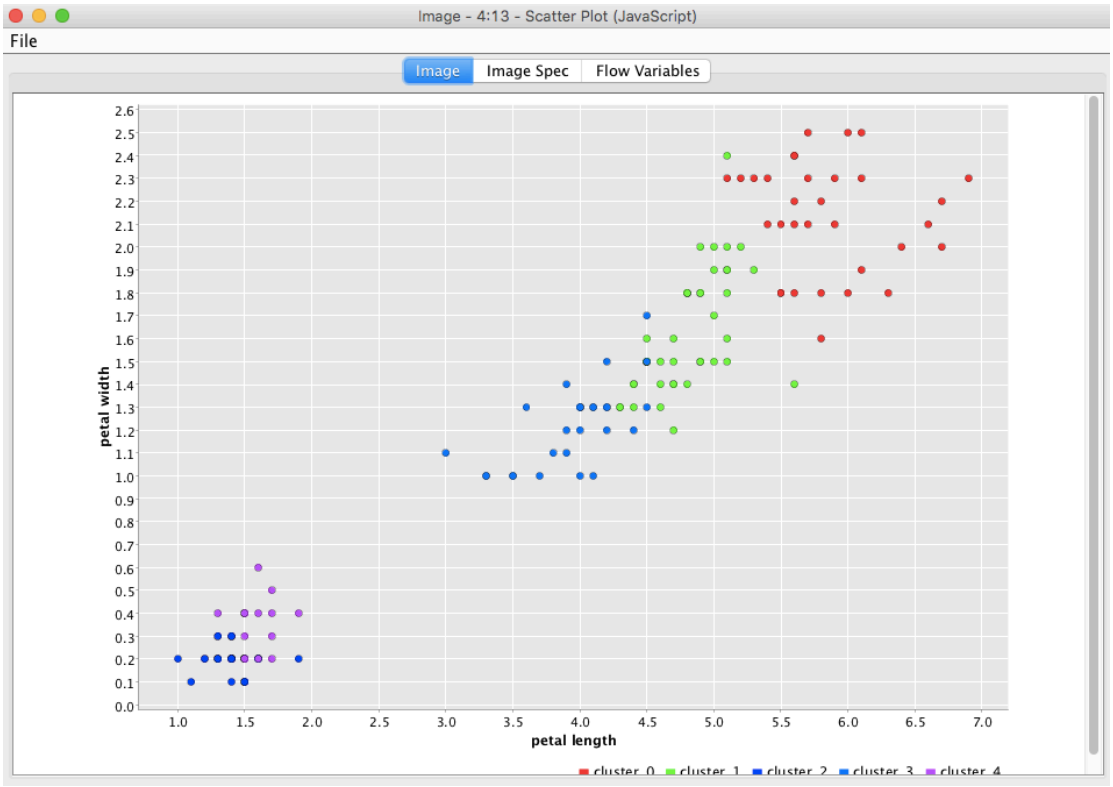
Şekil 8.10.8, color manager ve scatter plot eklendikten sonra renklendirilmiş ve gruplanmış veri setini göstermektedir.



Şekil 8.10.9

Şekil 8.10.9, x ve y koordinatları için scatter plot configure bölümünden seçilen farklı özelliklerin kullanılmasıyla oluşan gruplaşmayı göstermektedir.

Clustering'te etiket verilmeyen bir kolonun makine tarafından kendi içinde ayırt edici özellikleri fark ederek gruplamayı yapmasıdır.

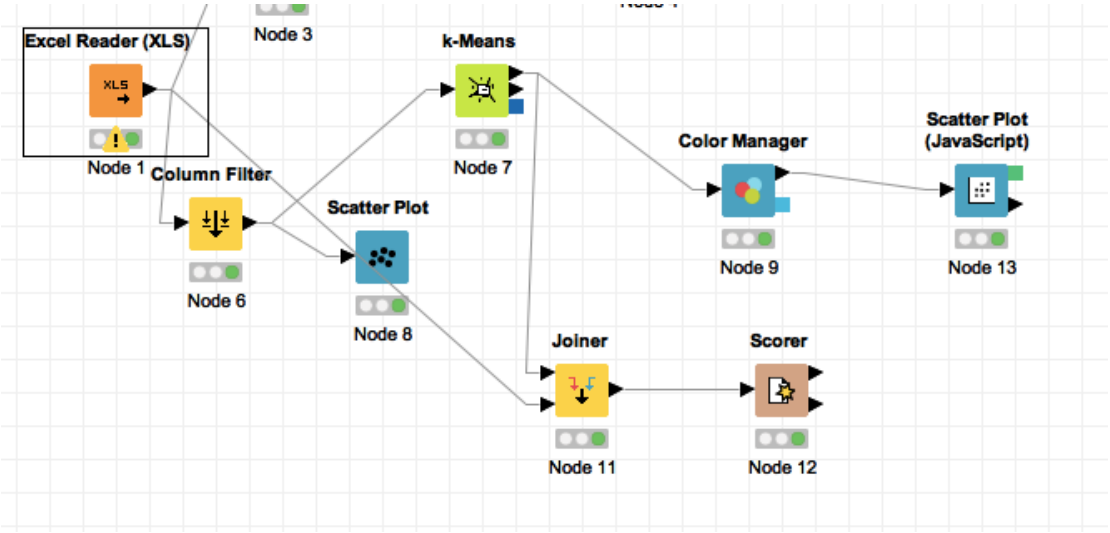


Şekil 8.10.10

Şekil 8.10.10, k-means configure bölümünden 5 cluster istenirse scatter plot için vereceği sonuçtur. 5 cluster olduğu için 5 gruplama yapmış ve bunları farklı renklendirmiştir.

Aşağıda, k-means yine 3 cluster yapılarak daha önce decision treeden gelen bilgiyi classification yapılabilir mi diye bakılacaktır. Clustering Classification algoritması gibi kullanılmaya çalışılacaktır.

Kolonlar joiner operatörü ile birleştirilecektir. Orjinal veri k-means'ten gelen veri birleştirilmesi Şekil 8.10.11 de gösterilmektedir.



Şekil 8.10.11

Şekil 8.10.11, joiner node'u ve diğer node'ların bağlantılarını göstermektedir. Şekil 8.10.12 ise joiner yapıldıktan sonra sonucu göstermektedir. ,

Joined table - 4:11 - Joiner

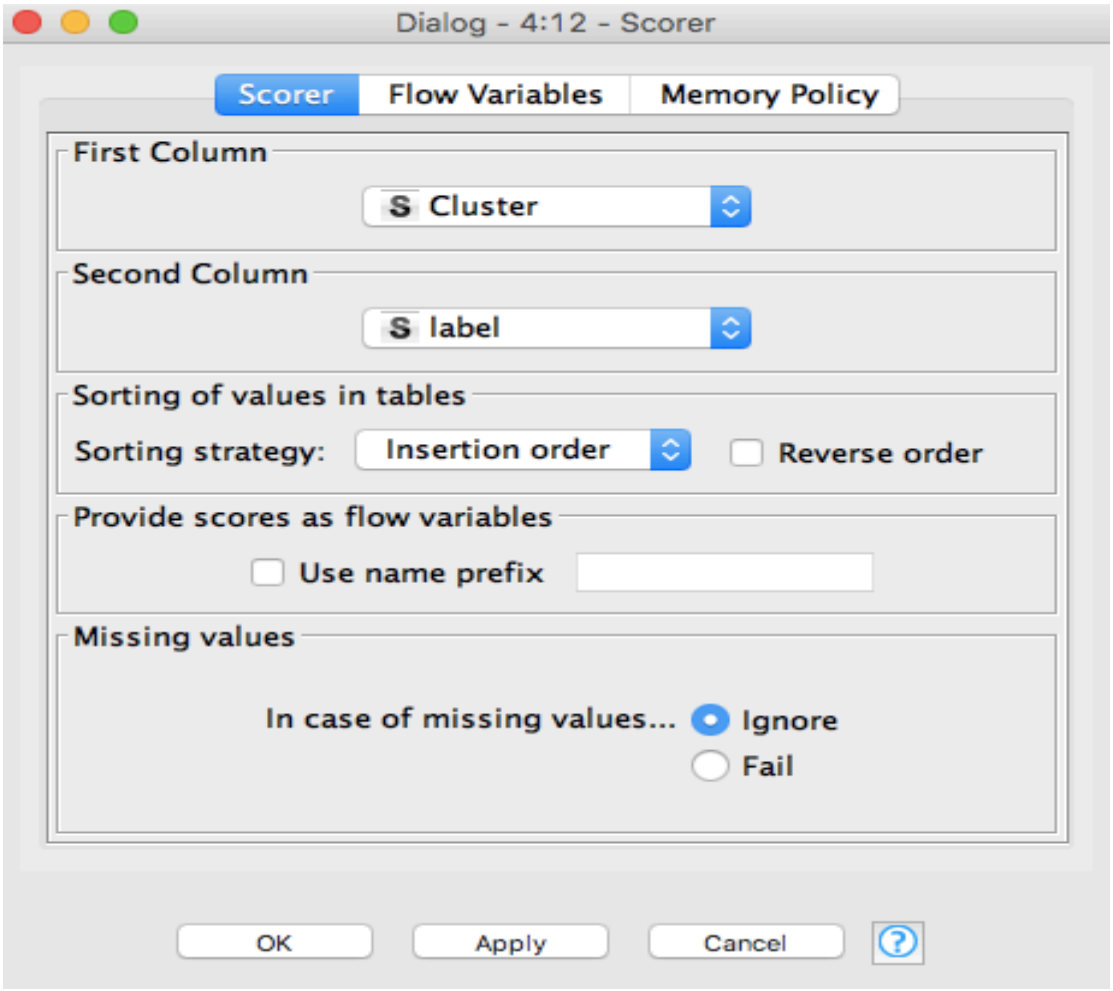
File Hilite Navigation View

Table "default" - Rows: 150 | Spec - Columns: 10 | Properties | Flow Variables

Row ID	D sepal ...	D sepal ...	D petal l...	D petal ...	S Cluster	D sepal ...	D sepal ...	D petal l...	D petal ...	S label
Row0	5.1	3.5	1.4	0.2	cluster_4	5.1	3.5	1.4	0.2	Iris-setosa
Row1	4.9	3	1.4	0.2	cluster_2	4.9	3	1.4	0.2	Iris-setosa
Row2	4.7	3.2	1.3	0.2	cluster_2	4.7	3.2	1.3	0.2	Iris-setosa
Row3	4.6	3.1	1.5	0.2	cluster_2	4.6	3.1	1.5	0.2	Iris-setosa
Row4	5	3.6	1.4	0.2	cluster_4	5	3.6	1.4	0.2	Iris-setosa
Row5	5.4	3.9	1.7	0.4	cluster_4	5.4	3.9	1.7	0.4	Iris-setosa
Row6	4.6	3.4	1.4	0.3	cluster_2	4.6	3.4	1.4	0.3	Iris-setosa
Row7	5	3.4	1.5	0.2	cluster_4	5	3.4	1.5	0.2	Iris-setosa
Row8	4.4	2.9	1.4	0.2	cluster_2	4.4	2.9	1.4	0.2	Iris-setosa
Row9	4.9	3.1	1.5	0.1	cluster_2	4.9	3.1	1.5	0.1	Iris-setosa
Row10	5.4	3.7	1.5	0.2	cluster_4	5.4	3.7	1.5	0.2	Iris-setosa
Row11	4.8	3.4	1.6	0.2	cluster_2	4.8	3.4	1.6	0.2	Iris-setosa
Row12	4.8	3	1.4	0.1	cluster_2	4.8	3	1.4	0.1	Iris-setosa
Row13	4.3	3	1.1	0.1	cluster_2	4.3	3	1.1	0.1	Iris-setosa
Row14	5.8	4	1.2	0.2	cluster_4	5.8	4	1.2	0.2	Iris-setosa
Row15	5.7	4.4	1.5	0.4	cluster_4	5.7	4.4	1.5	0.4	Iris-setosa
Row16	5.4	3.9	1.3	0.4	cluster_4	5.4	3.9	1.3	0.4	Iris-setosa
Row17	5.1	3.5	1.4	0.3	cluster_4	5.1	3.5	1.4	0.3	Iris-setosa
Row18	5.7	3.8	1.7	0.3	cluster_4	5.7	3.8	1.7	0.3	Iris-setosa
Row19	5.1	3.8	1.5	0.3	cluster_4	5.1	3.8	1.5	0.3	Iris-setosa
Row20	5.4	3.4	1.7	0.2	cluster_4	5.4	3.4	1.7	0.2	Iris-setosa
Row21	5.1	3.7	1.5	0.4	cluster_4	5.1	3.7	1.5	0.4	Iris-setosa
Row22	4.6	3.6	1	0.2	cluster_2	4.6	3.6	1	0.2	Iris-setosa
Row23	5.1	3.3	1.7	0.5	cluster_4	5.1	3.3	1.7	0.5	Iris-setosa
Row24	4.8	3.4	1.9	0.2	cluster_2	4.8	3.4	1.9	0.2	Iris-setosa
Row25	5	3	1.6	0.2	cluster_2	5	3	1.6	0.2	Iris-setosa
Row26	5	3.4	1.6	0.4	cluster_4	5	3.4	1.6	0.4	Iris-setosa
Row27	5.2	3.5	1.5	0.2	cluster_4	5.2	3.5	1.5	0.2	Iris-setosa
Row28	5.2	3.4	1.4	0.2	cluster_4	5.2	3.4	1.4	0.2	Iris-setosa

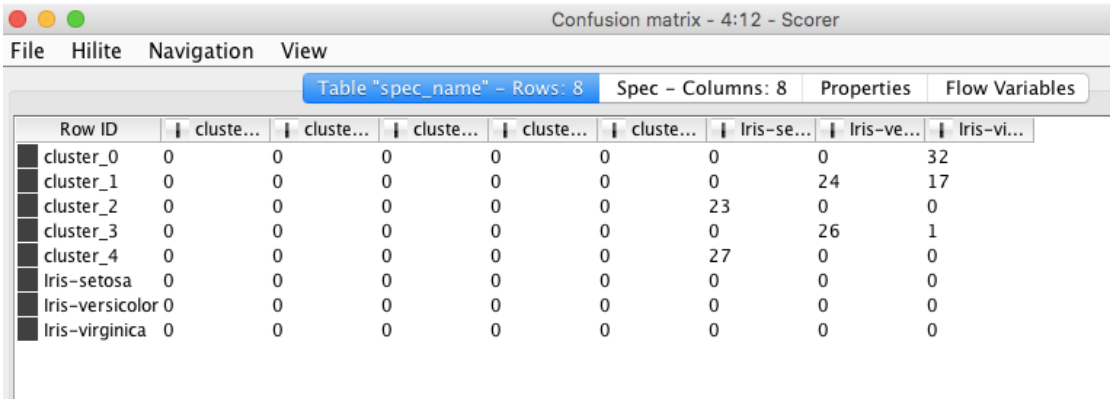
Şekil 8.10.12

Şekil 8.10.12 de görüldüğü üzere sol taraf ile sağ taraf yani inner join ile birleştirilmiştir. Yani sol taraf k-means'ten gelen, sağ taraf ise labeled olan yani orjinal veriden gelen verileri inner join ile birleştirilmiş halini göstermektedir. Benzerliğin ne kadar başarılı yapıldığına bakmak için scorer operatörü kullanılmalıdır.



Şekil 8.10.13

Şekil 8.10.13 scorer için configure bölümünü göstermektedir. Yani sol taraf k-means'ten gelen, sağ taraf ise labeled olan yani orjinal veriden gelen verilerin birleştirilmiş hallerinin sayısal değerlerler sonucunun gösterilebilmesi için eklenen scorer node'unun ayarlama ekranıdır ve bahsedilen kolonlar seçilmiştir.



Row ID	cluste...	cluste...	cluste...	cluste...	cluste...	Iris-se...	Iris-ve...	Iris-vi...
cluster_0	0	0	0	0	0	0	0	32
cluster_1	0	0	0	0	0	0	24	17
cluster_2	0	0	0	0	0	23	0	0
cluster_3	0	0	0	0	0	0	26	1
cluster_4	0	0	0	0	0	27	0	0
Iris-setosa	0	0	0	0	0	0	0	0
Iris-versicolor	0	0	0	0	0	0	0	0
Iris-virginica	0	0	0	0	0	0	0	0

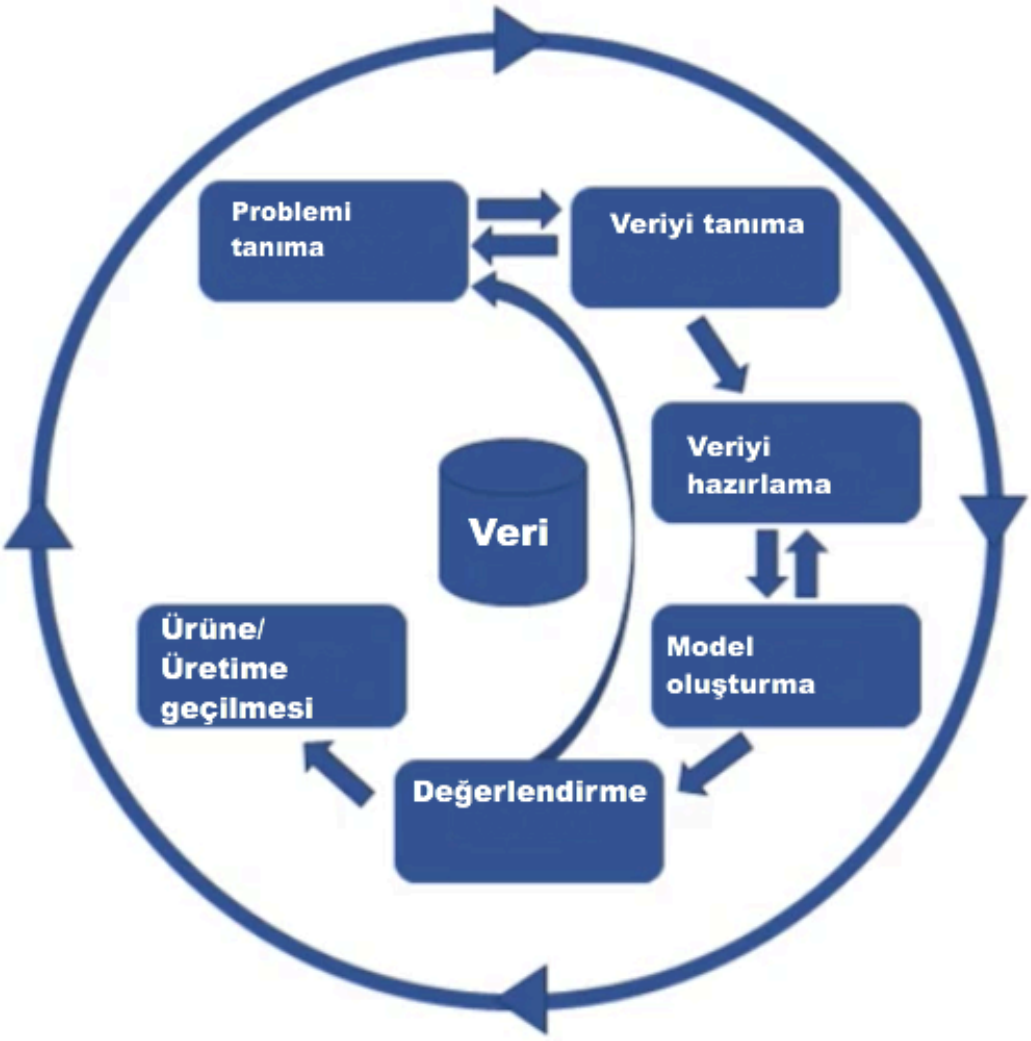
Şekil 8.10.14

Şekil 8.10.14 de görüldüğü gibi iris virginica çok rahat biçimde diğer gruplardan ayrılmış ve hepsi doğru etiketlenmiştir. Burada orjinal veriden gelen label ile k-mean'ten gelen teiket yapılmamış veri kıyaslaması yapılmaktadır. Cluster_1 yani iris versicolor için 17 tane veride hata yapılmıştır.

Bu bölümde, K-mean veri özelliklerine bakarak kendi gruplama yaparken decision tree'de label yani istenilen kolon belirtilerek sonuca ulaşılmaya çalışılmıştır ve bunların karşılaştırılması da yapılmıştır. Kısaca, supervised (k-nn/decision tree vb.) - unsupervised (k-means) algoritmalarının farkı gösterilmiştir.

8.11. Tahmin (Prediction) ve Doğrusal Regresyon (Linear Regression)

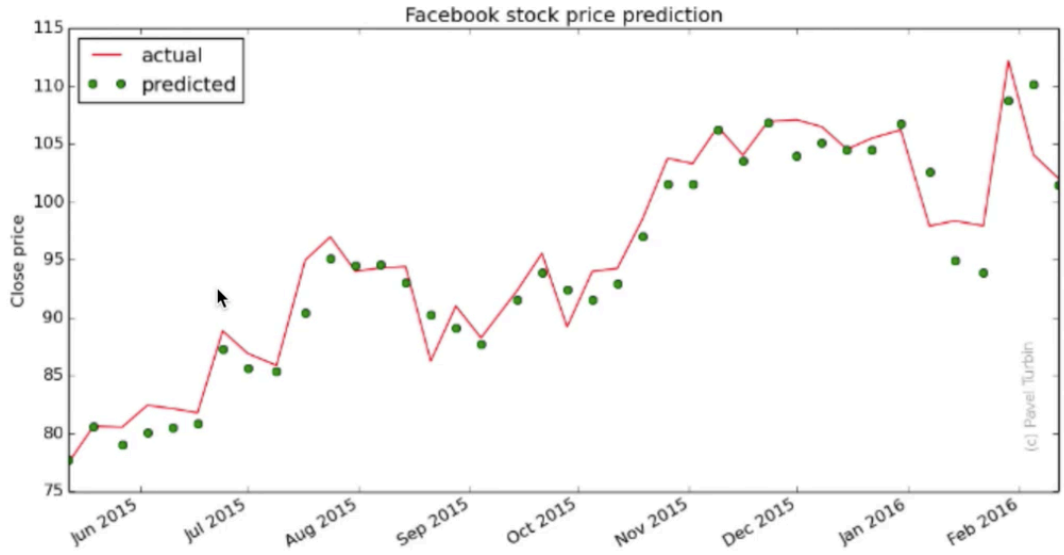
Bu bölümde amaç prediction (tahmin) ve linear regression kavramlarını tanıtmak.



Şekil 8.11.1

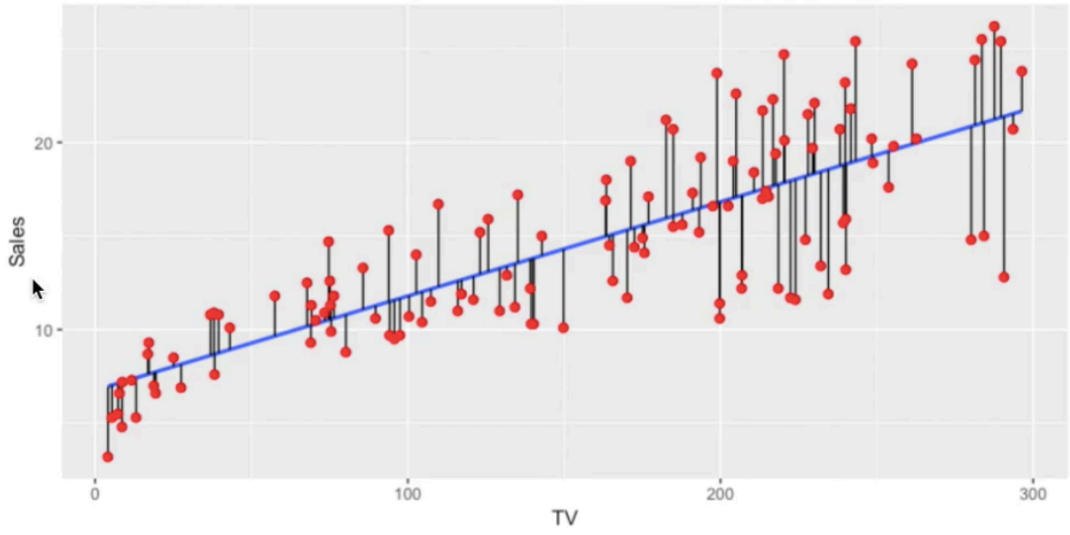
Şekil 8.11.1, CRISP-DM yönteminde problem anlaşıldıktan, veri anlaşıldıktan ve sonra veri hazırlandıktan sonra modelleme'nin yapıldığı daha önceki bölümlerde açıklanmıştı. Modellemede classification, clustering, ARM ve prediction'dan oluşmakta ve daha önceki bölümlerde classification, clustering ve ARM açıklanmıştır. Bu bölümde ise prediction açıklanacaktır. Bu problemler birbirine dönüştürülebilir modellerdir. Daha önceki bölümde işlenen sınıflandırma da bir tahmin çeşididir. Fakat burada tahmin ile kastedilen farklıdır. Bir önceki bölümde yaprak çeşitlerinden iris_setosa, iris_ versicolor ve iris_virginica bulunmaya çalışması bir etiketlemedir. Tahmin'den kastedilen ve bulunmaya çalışılanlar sayısal değerlerdir. Yaprak örneğindekiler nominal değerlerdir.

Toplanıp çıkarılamayan yani dört işlem yapılamayacak değerlerdir. Bunlar sınıflandırma için kullanılan değerleridir. Tahmin için sayısal yani numeric değerler kullanılır ve dört işlem yapılabilir.



Şekil 8.11.2

Şekil 8.11.2'de kırmızı çizgi Facebook'un borsadaki kapanış gerçek değerlerini , yeşil noktalar ise tahmin değerleri göstermektedir. Bu değerler yani veriler sayısal (numeric) değerlerdir. Yanlış tahmin veya doğru tahmin yoktur. Tahminin hata payı vardır. Örneğin örnekte bazı noktalarda doğruya çok yakın yani çizgi üzerinde çıkmıştır tahmin değerleri. Bazı noktalar ise çizgiden uzak yani hata payı yüksek çıkmış tahminlerdir.



Şekil 8.11.3

Şekil 8.11.3Doğrusal regresyonu (linear regression)'u göstermektedir. Grafik zamana bağlı satış değerleri göstermektedir. Kırmızı noktalar gerçek satış değerlerini ve mavi çizgi ise linear regresion doğrusunu (y=ax+b) göstermektedir. Amaç, en az hataya sahip doğruyu çizmek yani noktalara en yakın çizgiyi çizebilmek.

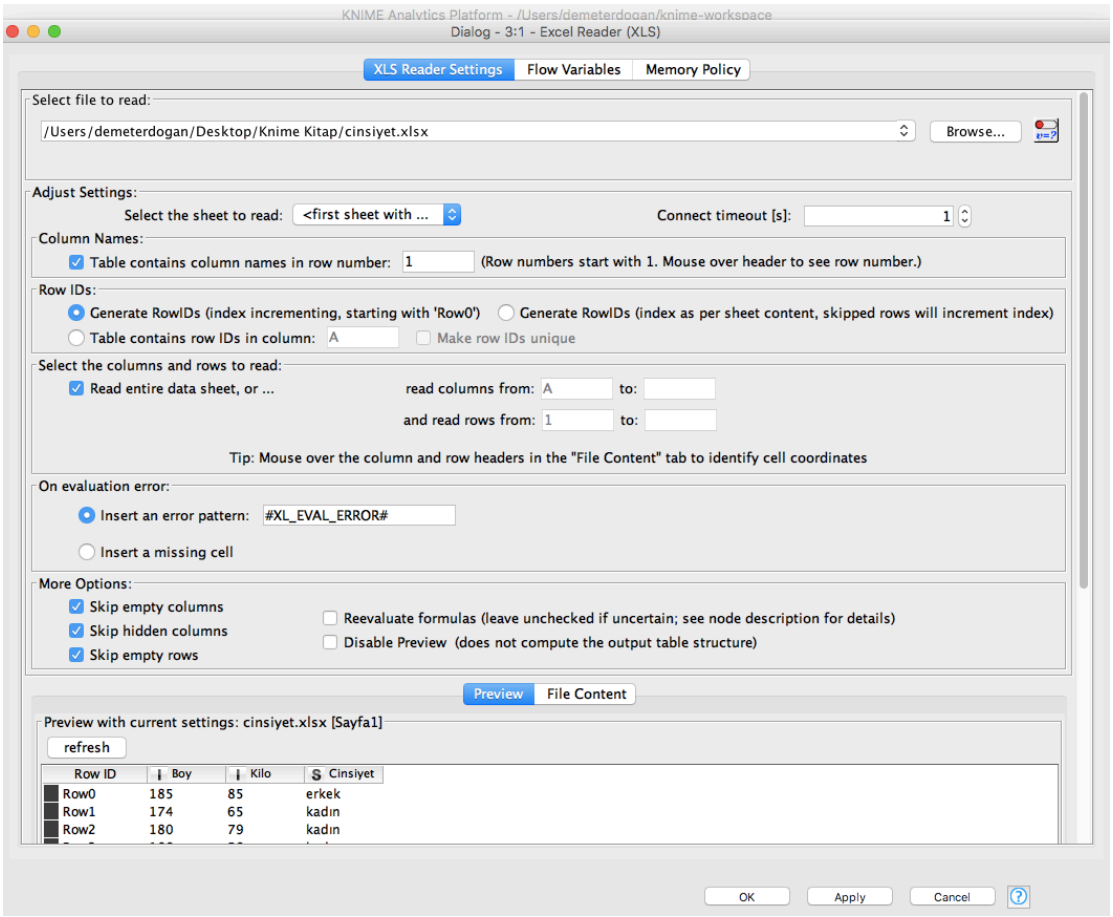
Prediction bir modellemede kullanılan bir tahmin çeşididir. Veri madenciliği ve makine öğrenmesinin temel problemlerinden biridir.

Prediction ve forecasting tahmin demektir. Forecasting'e öngörü de denebilir. Yani 300 günlük veri bulunuyor 301 veya 600. Vb günlerdeki veri ne olabilir sorusunu cevaplamaya çalışır. Prediction ise geçmiş veya gelecekteki bir veriyi tahmin etmeye çalışır.

8.12. Knime ile Tahmin (Prediction) ve Doğrusal Regresyon (Linear Regression) Örneği

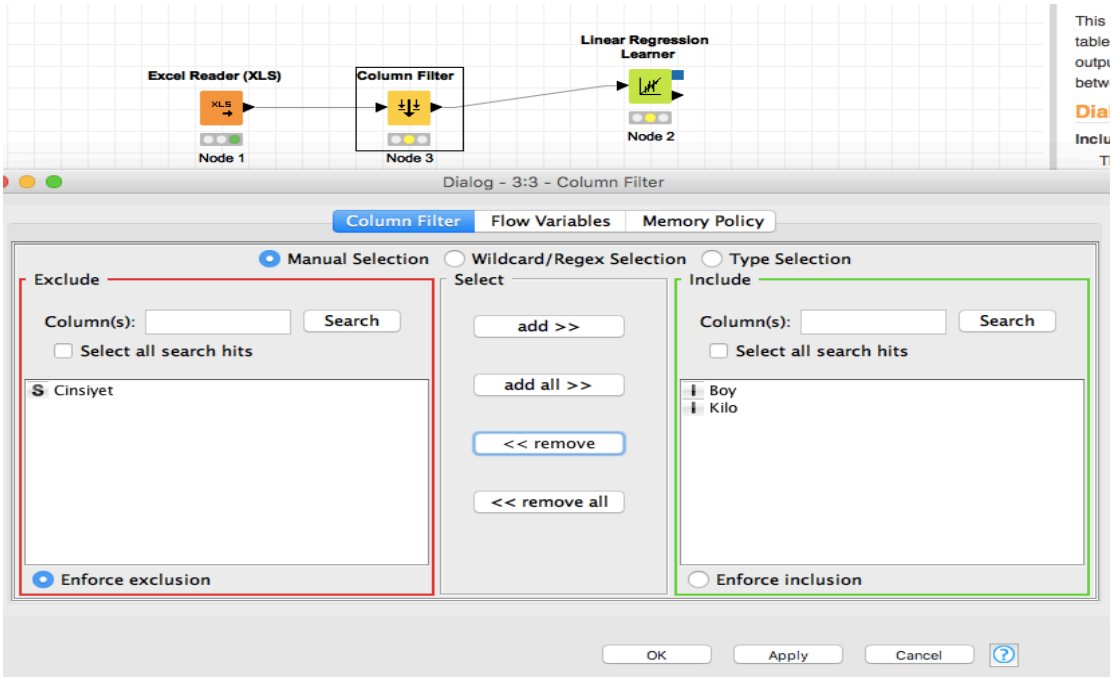
Bu bölümde amaçi yukarıdaki bölümlerde teorisi verilmiş kavramların örnek ile gösterilmesidir.

Bu bölümde de daha önceki bölümlerde kullanılan cinsiyet veri seti kullanılacaktır.



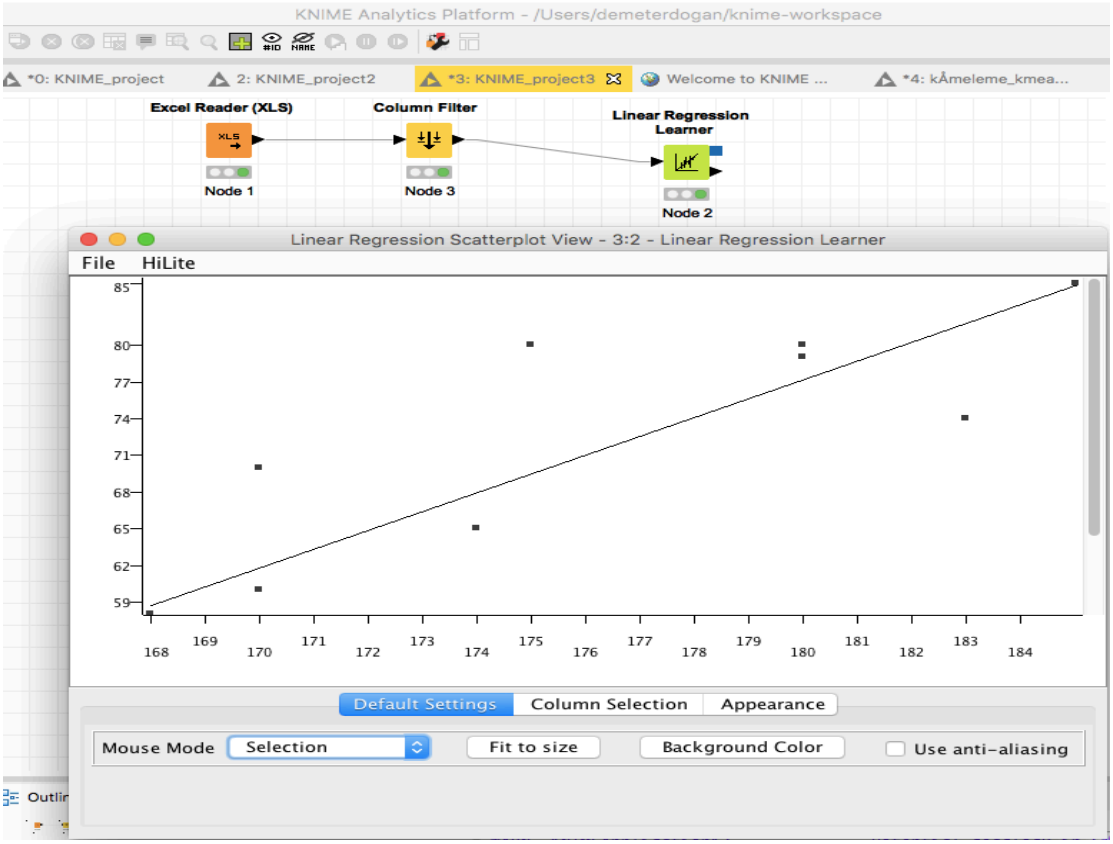
Şekil 8.12.1

Şekil 8.12.1 Veri setinin excel reader opratörüne yüklenmesini ve ilk row'un kolon isimlerini içeren sıra olduğunun seçim ekranını göstermektedir. Boy arttıkça kilo artar gibi genel anlamda doğrusal bir ilişki olduğu anlaşılabilir. Sayısal tahmin amaçlı boydan kiloyu ya da kilodan boya ulaşmaya çalışılacaktır.



Şekil 8.12.2

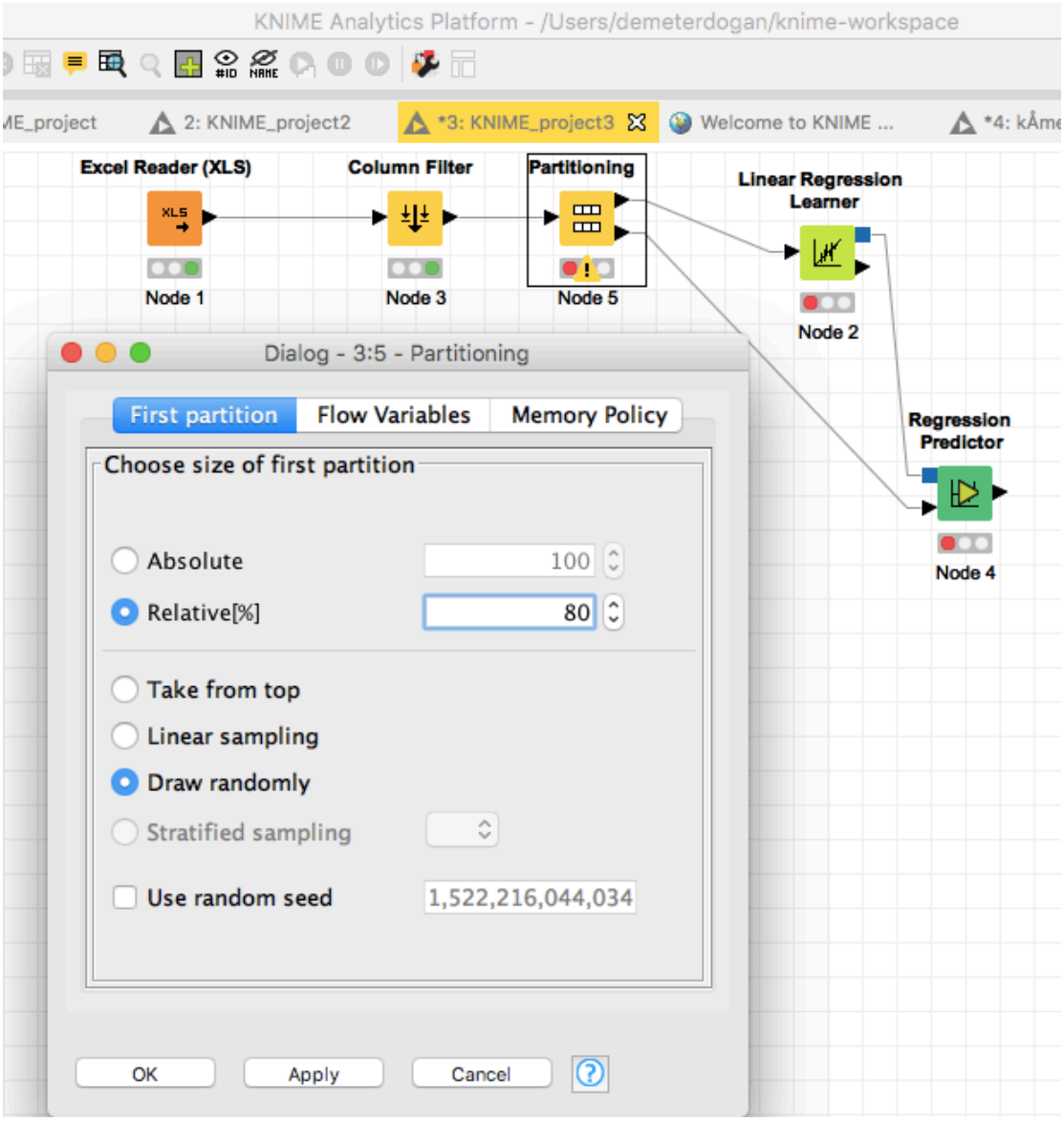
Şekil 8.12.2, excel reader, column filter ve linear regression learner operatörlerinin bağlantılarını göstermektedir. Label nominal değer olduğu ve amaç boy-kilo arasında ilişki olunduğu için cinsiyet label değeri listeden çıkarılmıştır.



Şekil 8.12.3

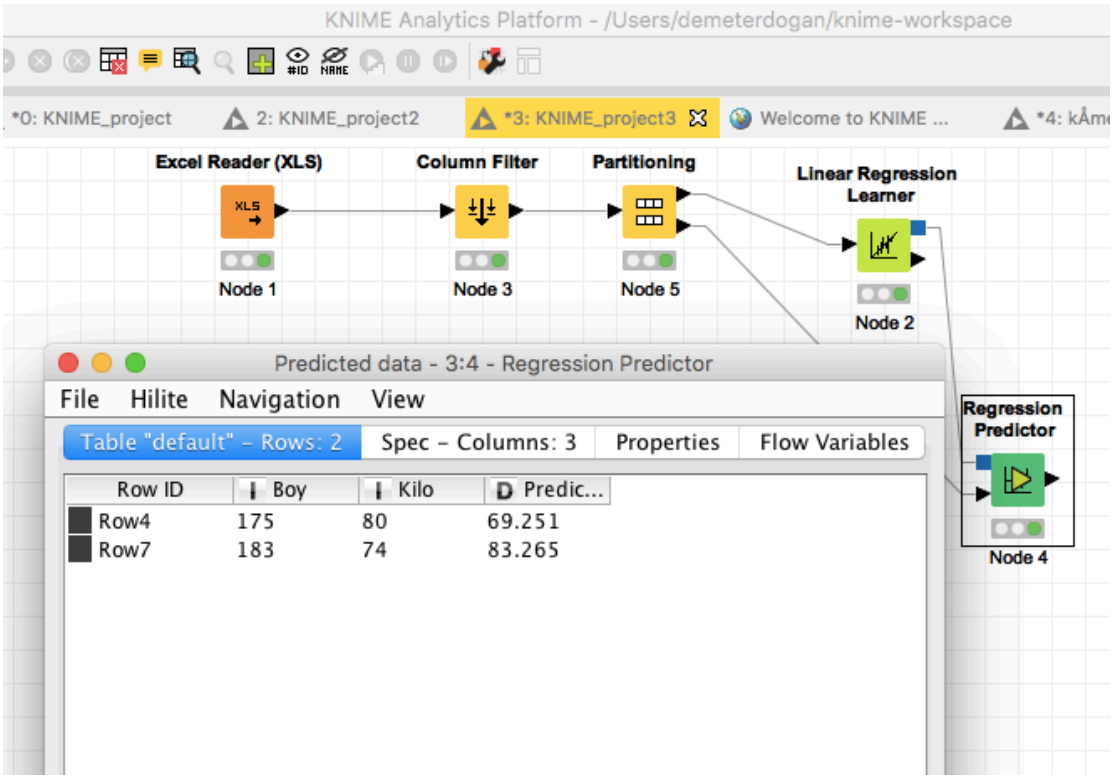
Şekil 8.12.3, Linear Regression Learner'ın programı çalıştırdıktan sonra oluşturduğu scatterplot'u göstermektedir. Bu şekildeki doğru olabilecek en minimum hata miktarına sahip doğrudur.

Sayısal değerlerin kullanımı için öncelikle veriyi bölmek gerekir. Daha sonra da regression predictor ekleyerek öğrenmesinin doğruluğunun testi için kullanılır.



Şekil 8.12.4

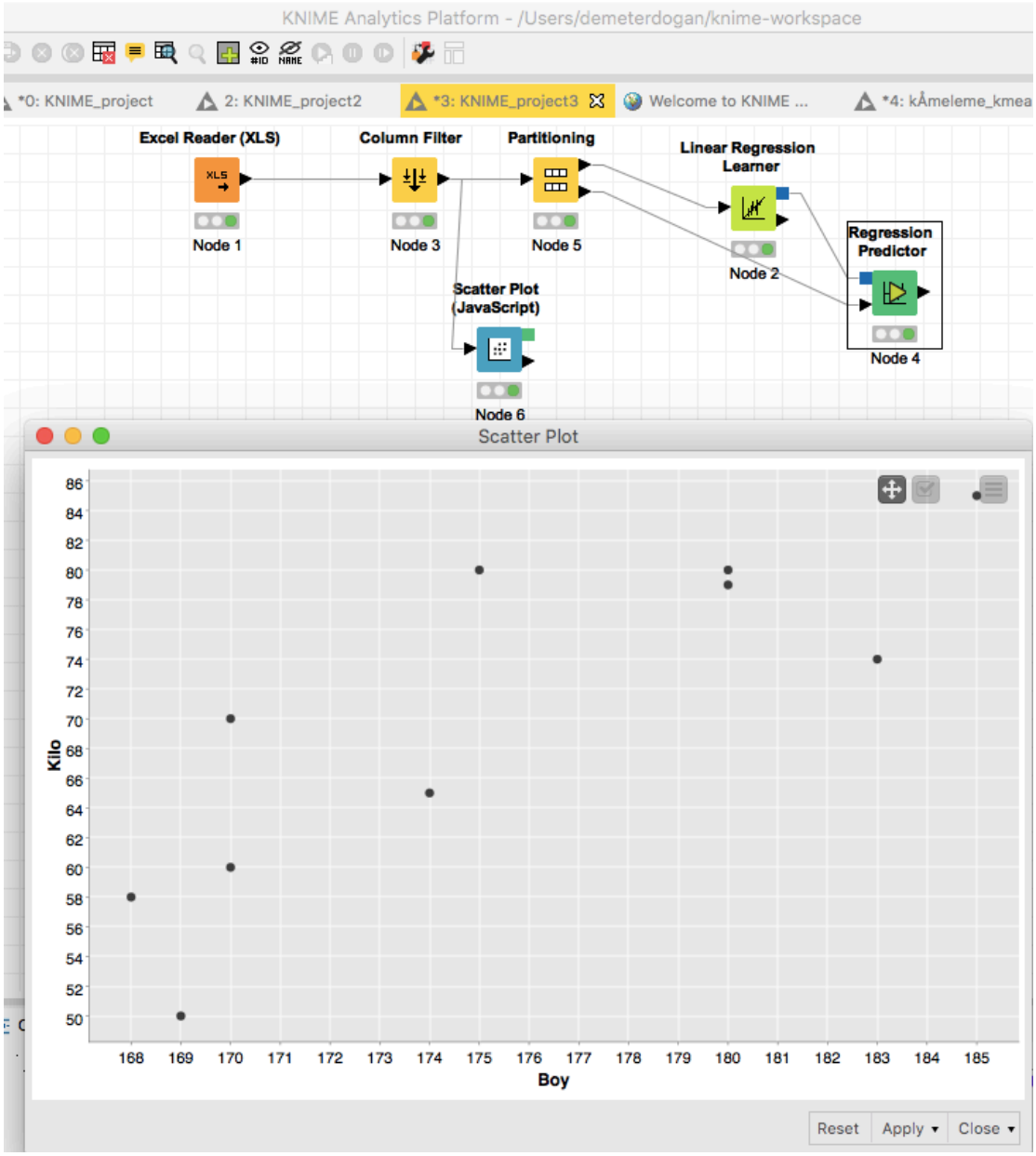
Şekil 8.12.4, sisteme partitioning ve regression predictor eklenmiş halini ve partitioning configure bölümündeki yapılan değişikliği göstermektedir.



Şekil 8.12.5

Şekil 8.12.5, program çalıştırıldıktan sonraki predicted data penceresini göstermektedir. Buradan da görüldüğü gibi, 175 boyundaki 80 kg olan bir kişiyi 69.251kg ve 183 boyundaki 74 kg birini de 83 kg olarak tahmin etmiştir.

Boy-kiilo doğru orantı kurulabilecek iyi bir örnektir fakat bazen çok daha karmaşık örneklerle de karşılaşılabilir.



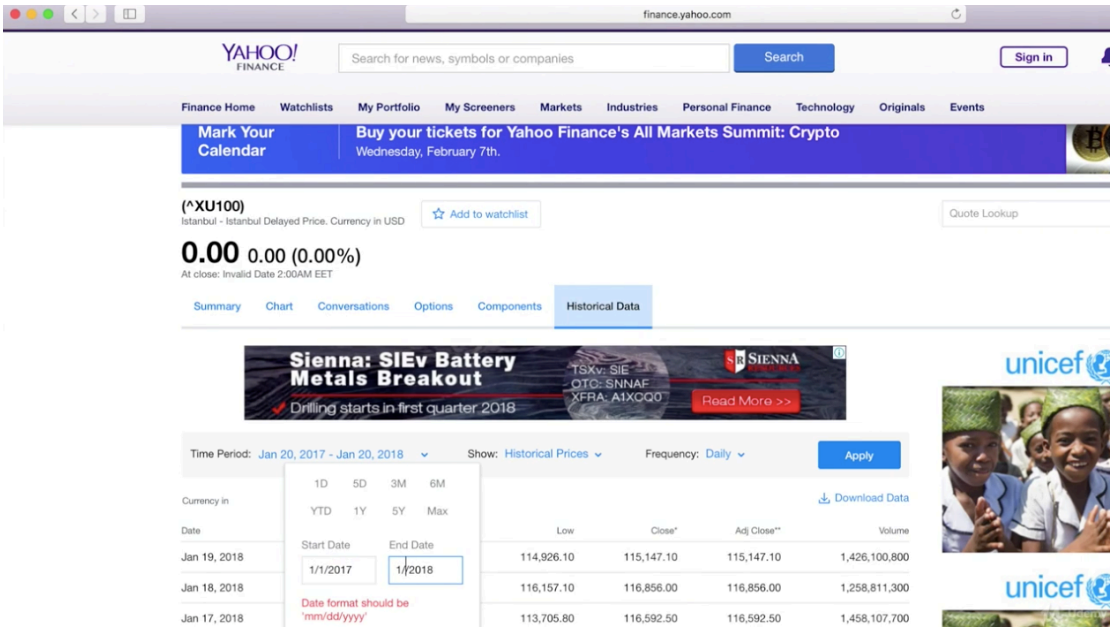
Şekil 8.12.6

Şekil 8.12.6, en ilkel biçimde oluşturulan scatter plot sonucunu göstermektedir. Burada da görüldüğü gibi örneğin 169 boyundaki birinin 50 kg olduğu, 174 birinin 65 kg olduğudur. Linear regression ilkelleme demektir. Karmaşık verilerin doğru çizerek birbiriyle oranına bakılmasını sağlar.

8.13. Tahmin Örneği: Borsa Verisi

Bu bölümde amaç, linear regression yöntemi kullanılarak borsadaki değerlerin tahmin edilmesidir.

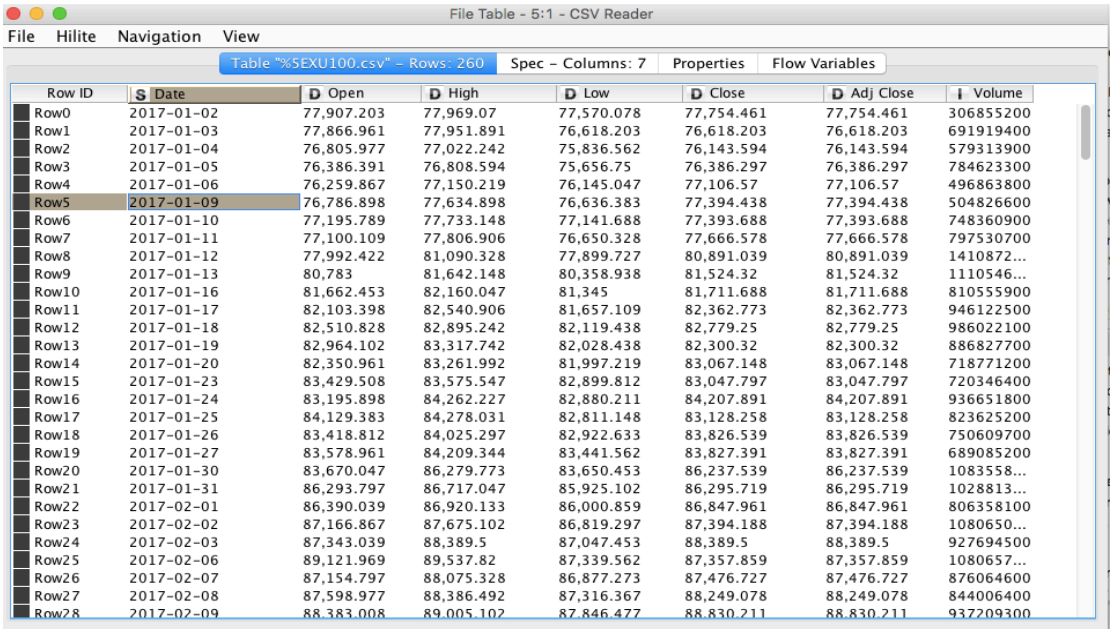
Bu bölümde kullanılacak veriler Yahoo finans sitesinden indirilebilir.



Şekil 8.13.1

Şekil 8.13.1'de görüldüğü gibi XU'lu veriler Türkiye'ye ait borsa verilerini içerir. Tarih aralığı istenilen bir aralık girilerek apply (uygula) denerek sonra download (indirmek) edilir.

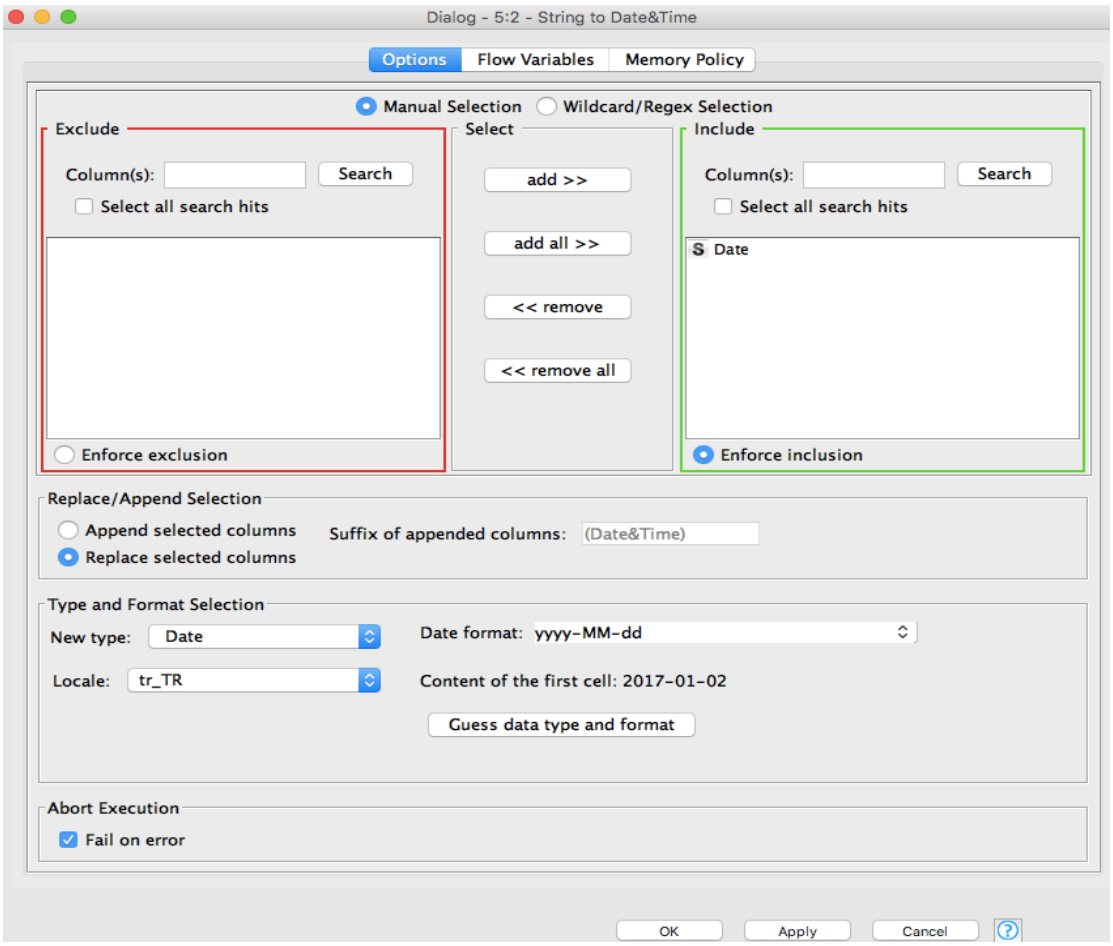
İnen veriler csv formatındadır ve bu verilerin sisteme tanıtılması için csv reader operatörü kullanılır. Configure bölümünde has row reader bölümündeki kutucuk işaretli olmamalıdır.



Row ID	Date	Open	High	Low	Close	Adj Close	Volume
Row0	2017-01-02	77,907.203	77,969.07	77,570.078	77,754.461	77,754.461	306855200
Row1	2017-01-03	77,866.961	77,951.891	76,618.203	76,618.203	76,618.203	691919400
Row2	2017-01-04	76,805.977	77,022.242	75,836.562	76,143.594	76,143.594	579313900
Row3	2017-01-05	76,386.391	76,808.594	75,656.75	76,386.297	76,386.297	784623300
Row4	2017-01-06	76,259.867	77,150.219	76,145.047	77,106.57	77,106.57	496863800
Row5	2017-01-09	76,786.898	77,634.898	76,636.383	77,394.438	77,394.438	504826600
Row6	2017-01-10	77,195.789	77,733.148	77,141.688	77,393.688	77,393.688	748360900
Row7	2017-01-11	77,100.109	77,806.906	76,650.328	77,666.578	77,666.578	797530700
Row8	2017-01-12	77,992.422	81,090.328	77,899.727	80,891.039	80,891.039	1410872...
Row9	2017-01-13	80,783	81,642.148	80,358.938	81,524.32	81,524.32	1110546...
Row10	2017-01-16	81,662.453	82,160.047	81,345	81,711.688	81,711.688	810555900
Row11	2017-01-17	82,103.398	82,540.906	81,657.109	82,362.773	82,362.773	946122500
Row12	2017-01-18	82,510.828	82,895.242	82,119.438	82,779.25	82,779.25	986022100
Row13	2017-01-19	82,964.102	83,317.742	82,028.438	82,300.32	82,300.32	886827700
Row14	2017-01-20	82,350.961	83,261.992	81,997.219	83,067.148	83,067.148	718771200
Row15	2017-01-23	83,429.508	83,575.547	82,899.812	83,047.797	83,047.797	720346400
Row16	2017-01-24	83,195.898	84,262.227	82,880.211	84,207.891	84,207.891	936651800
Row17	2017-01-25	84,129.383	84,278.031	82,811.148	83,128.258	83,128.258	823625200
Row18	2017-01-26	83,418.812	84,025.297	82,922.633	83,826.539	83,826.539	750609700
Row19	2017-01-27	83,578.961	84,209.344	83,441.562	83,827.391	83,827.391	689085200
Row20	2017-01-30	83,670.047	86,279.773	83,650.453	86,237.539	86,237.539	1083558...
Row21	2017-01-31	86,293.797	86,717.047	85,925.102	86,295.719	86,295.719	1028813...
Row22	2017-02-01	86,390.039	86,920.133	86,000.859	86,847.961	86,847.961	806358100
Row23	2017-02-02	87,166.867	87,675.102	86,819.297	87,394.188	87,394.188	1080650...
Row24	2017-02-03	87,343.039	88,389.5	87,047.453	88,389.5	88,389.5	927694500
Row25	2017-02-06	89,121.969	89,537.82	87,339.562	87,357.859	87,357.859	1080657...
Row26	2017-02-07	87,154.797	88,075.328	86,877.273	87,476.727	87,476.727	876064600
Row27	2017-02-08	87,598.977	88,386.492	87,316.367	88,249.078	88,249.078	844006400
Row28	2017-02-09	88,383.008	89,005.102	87,846.477	88,830.211	88,830.211	937209300

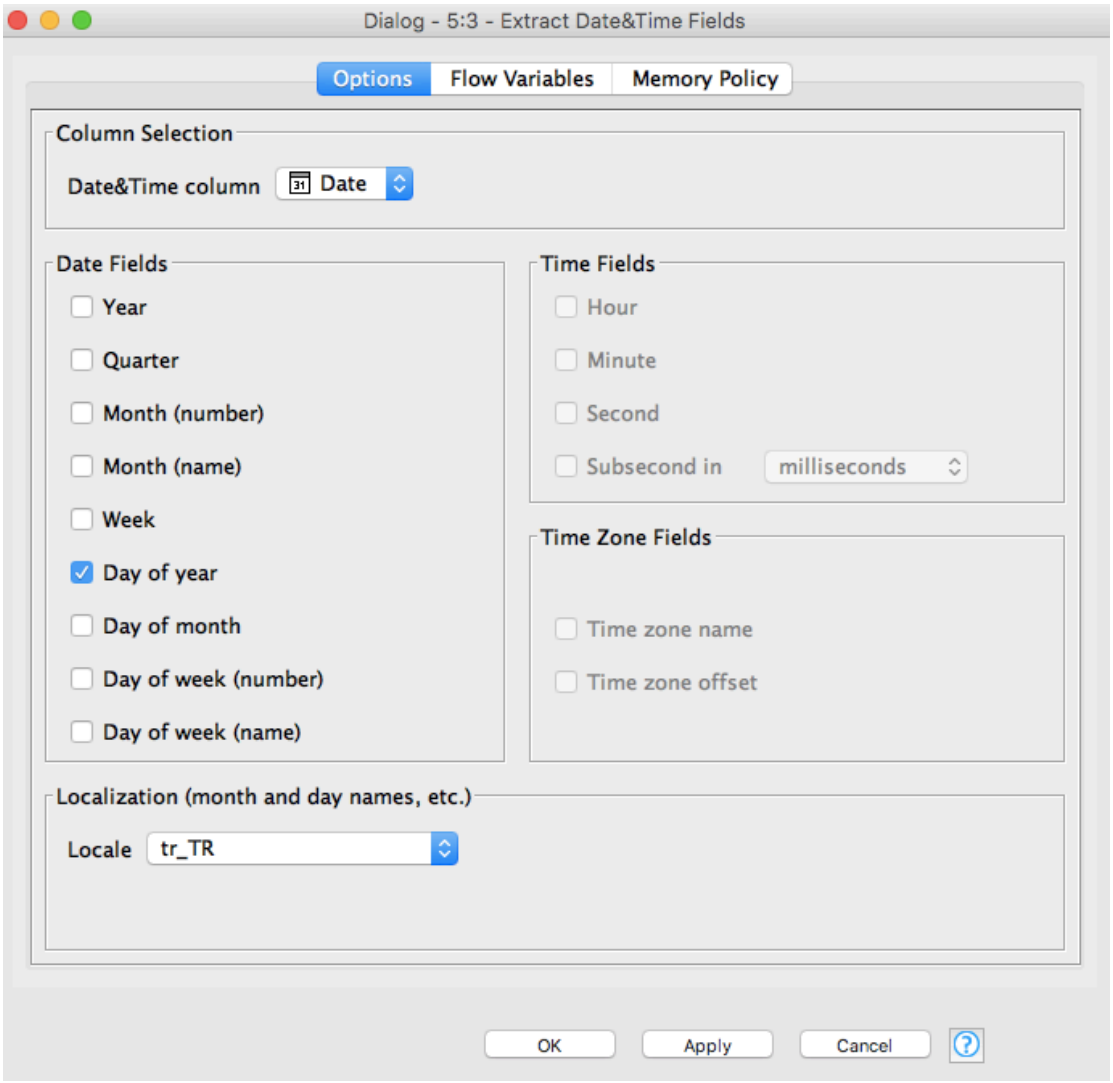
Şekil 8.13.2

Şekil 8.13,2'de görüldüğü gibi date kolonundaki bilgiler string formatındadır ve bu fomat date formatı ile değiştirilmelidir.



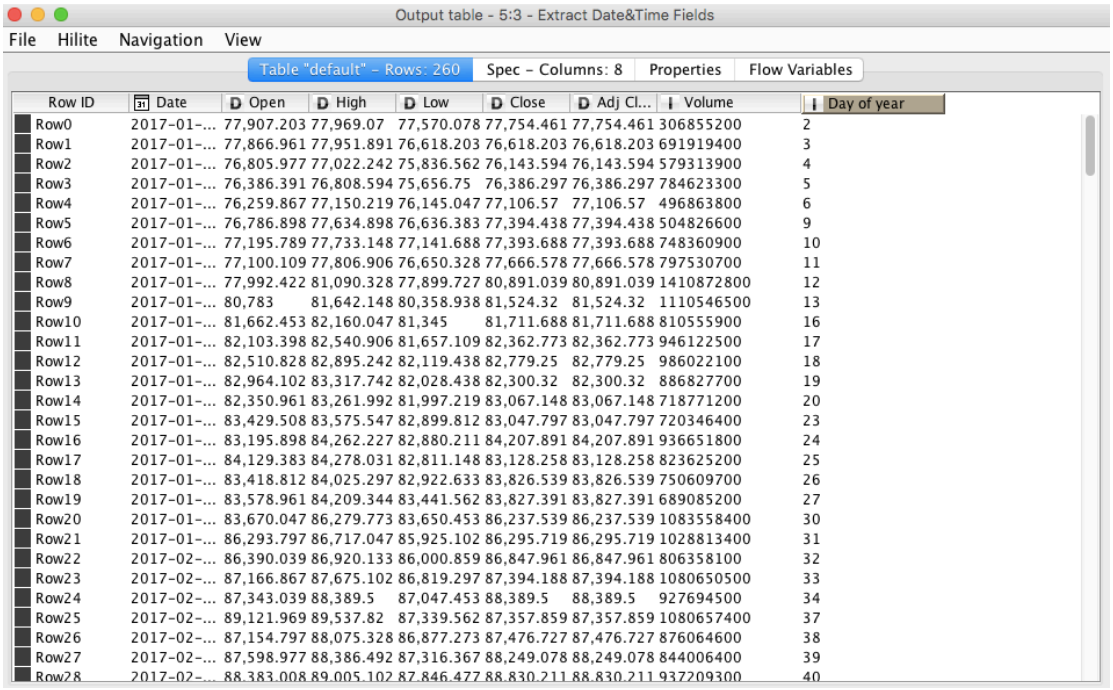
Şekil 8.13.3

Şekil 8.13.3'de string formata sahip date'lerin string'ten yyyy-MM-dd date formatına çevrilecektir. Locale tr_TR olarak seçilmiş ve new type'da date olarak belirlenmiştir.



Şekil 8.13.4

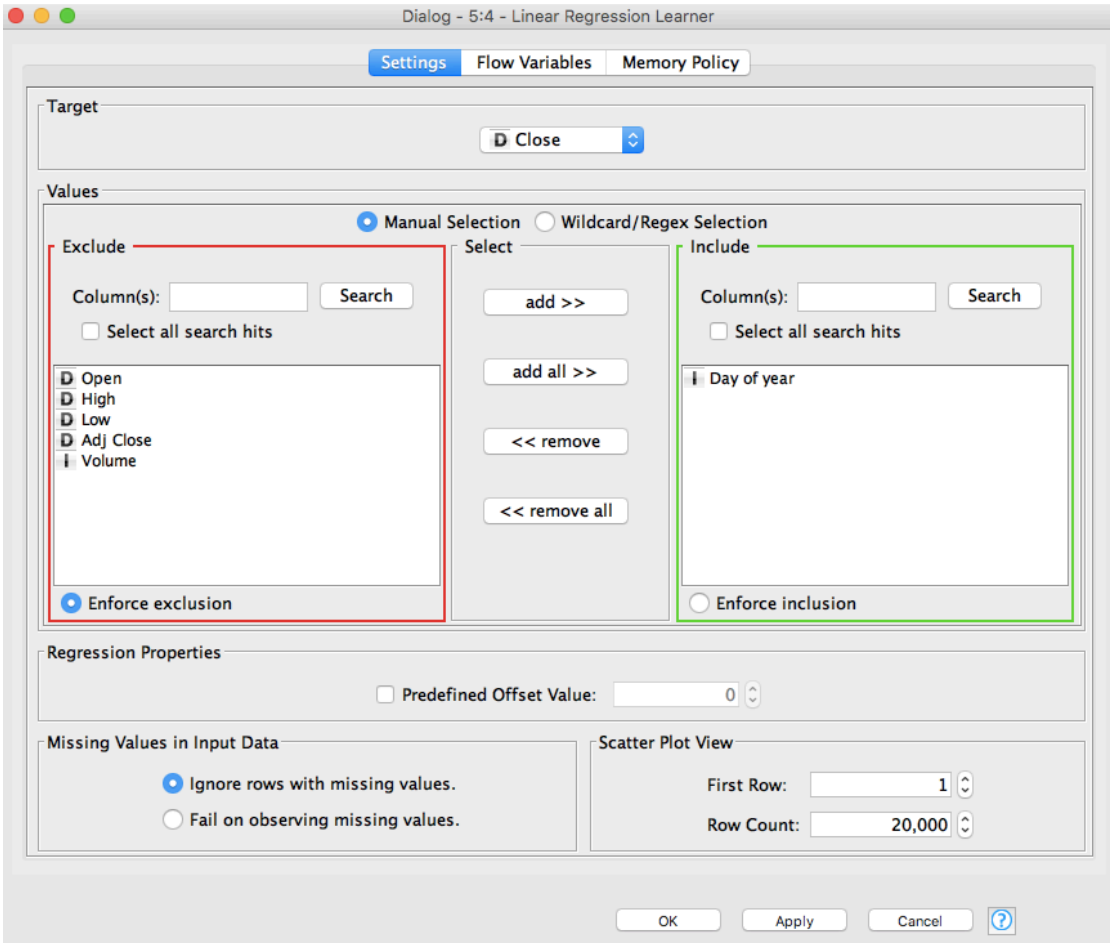
Şekil 8.13.4, date için yani tarih için bir formata dönüştürebilmeyi sağlayan ve bunu yeni oluşturacağı kolonda gösteren extract date & time field operatörü sisteme eklenerek dday of year dönüşümünün seçimini göstermektedir.



Row ID	Date	Open	High	Low	Close	Adj Cl...	Volume	Day of year
Row0	2017-01-...	77,907.203	77,969.07	77,570.078	77,754.461	77,754.461	306855200	2
Row1	2017-01-...	77,866.961	77,951.891	76,618.203	76,618.203	76,618.203	691919400	3
Row2	2017-01-...	76,805.977	77,022.242	75,836.562	76,143.594	76,143.594	579313900	4
Row3	2017-01-...	76,386.391	76,808.594	75,656.75	76,386.297	76,386.297	784623300	5
Row4	2017-01-...	76,259.867	77,150.219	76,145.047	77,106.57	77,106.57	496863800	6
Row5	2017-01-...	76,786.898	77,634.898	76,636.383	77,394.438	77,394.438	504826600	9
Row6	2017-01-...	77,195.789	77,733.148	77,141.688	77,393.688	77,393.688	748360900	10
Row7	2017-01-...	77,100.109	77,806.906	76,650.328	77,666.578	77,666.578	797530700	11
Row8	2017-01-...	77,992.422	81,090.328	77,899.727	80,891.039	80,891.039	1410872800	12
Row9	2017-01-...	80,783	81,642.148	80,358.938	81,524.32	81,524.32	1110546500	13
Row10	2017-01-...	81,662.453	82,160.047	81,345	81,711.688	81,711.688	810555900	16
Row11	2017-01-...	82,103.398	82,540.906	81,657.109	82,362.773	82,362.773	946122500	17
Row12	2017-01-...	82,510.828	82,895.242	82,119.438	82,779.25	82,779.25	986022100	18
Row13	2017-01-...	82,964.102	83,317.742	82,028.438	82,300.32	82,300.32	886827700	19
Row14	2017-01-...	82,350.961	83,261.992	81,997.219	83,067.148	83,067.148	718771200	20
Row15	2017-01-...	83,429.508	83,575.547	82,899.812	83,047.797	83,047.797	720346400	23
Row16	2017-01-...	83,195.898	84,262.227	82,880.211	84,207.891	84,207.891	936651800	24
Row17	2017-01-...	84,129.383	84,278.031	82,811.148	83,128.258	83,128.258	823625200	25
Row18	2017-01-...	83,418.812	84,025.297	82,922.633	83,826.539	83,826.539	750609700	26
Row19	2017-01-...	83,578.961	84,209.344	83,441.562	83,827.391	83,827.391	689085200	27
Row20	2017-01-...	83,670.047	86,279.773	83,650.453	86,237.539	86,237.539	1083558400	30
Row21	2017-01-...	86,293.797	86,717.047	85,925.102	86,295.719	86,295.719	1028813400	31
Row22	2017-02-...	86,390.039	86,920.133	86,000.859	86,847.961	86,847.961	806358100	32
Row23	2017-02-...	87,166.867	87,675.102	86,819.297	87,394.188	87,394.188	1080650500	33
Row24	2017-02-...	87,343.039	88,389.5	87,047.453	88,389.5	88,389.5	927694500	34
Row25	2017-02-...	89,121.969	89,537.82	87,339.562	87,357.859	87,357.859	1080657400	37
Row26	2017-02-...	87,154.797	88,075.328	86,877.273	87,476.727	87,476.727	876064600	38
Row27	2017-02-...	87,598.977	88,386.492	87,316.367	88,249.078	88,249.078	844006400	39
Row28	2017-02-...	88,383.008	89,005.102	87,846.477	88,830.211	88,830.211	937209300	40

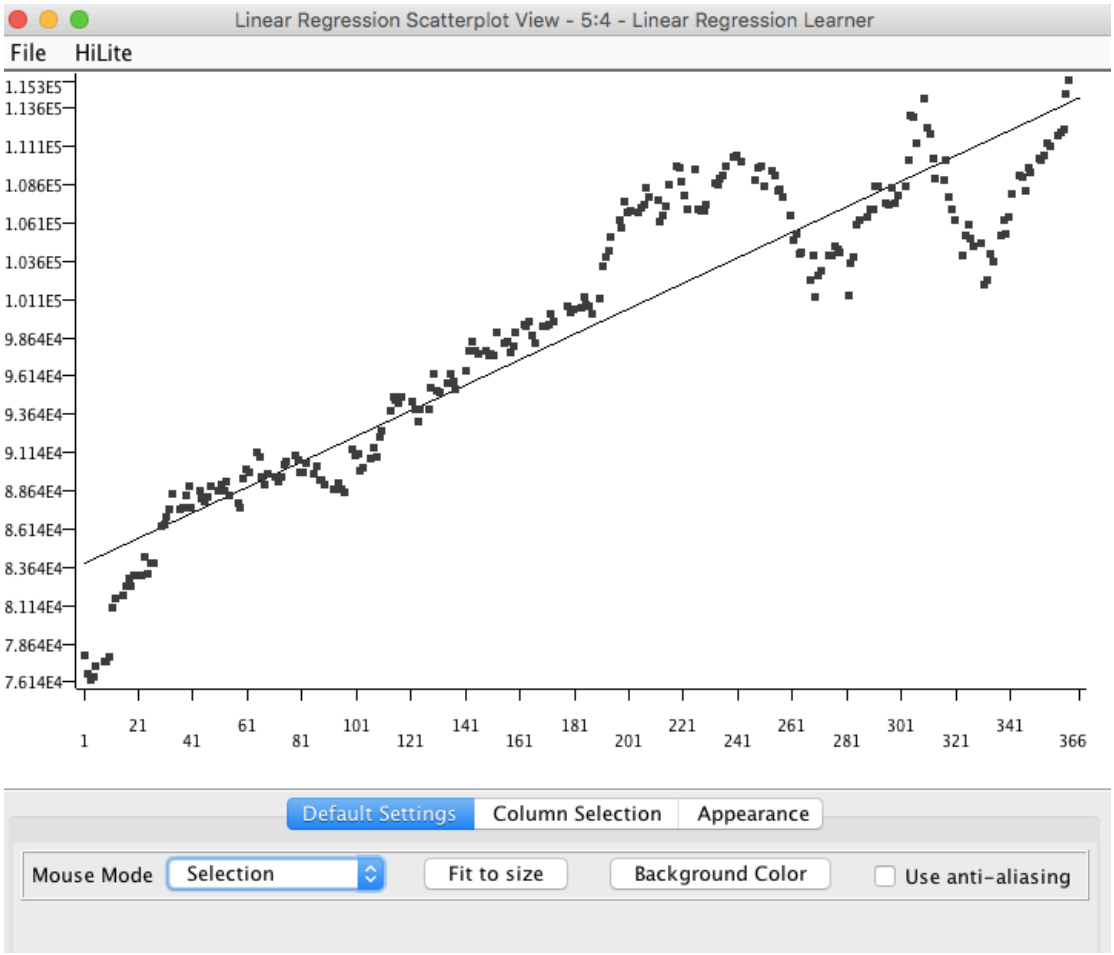
Şekil 8.13.5

Şekil 8.13.5'te görüldüğü gibi son kolon "day of year" bir önceki şekilde seçildiği için sisteme eklenmiştir. Burada date kolonunda yazan tarihin yılın kaçıncı günü olduğuna dönüştürmektedir. Örneğin 2,3,4,5,6. Günlerden sonra 9. Güne geçmiştir. Yani aradaki iki gün haftasonu tatili olduğu için borsa kapalıdır. Bu bilgi ile prediction yapılabilir. Örneğin borsa açık olsaydı o iki günde nasıl veriler olurdu gibi bilgiler için hesaplama yapılabilir.



Şekil 8.13.6

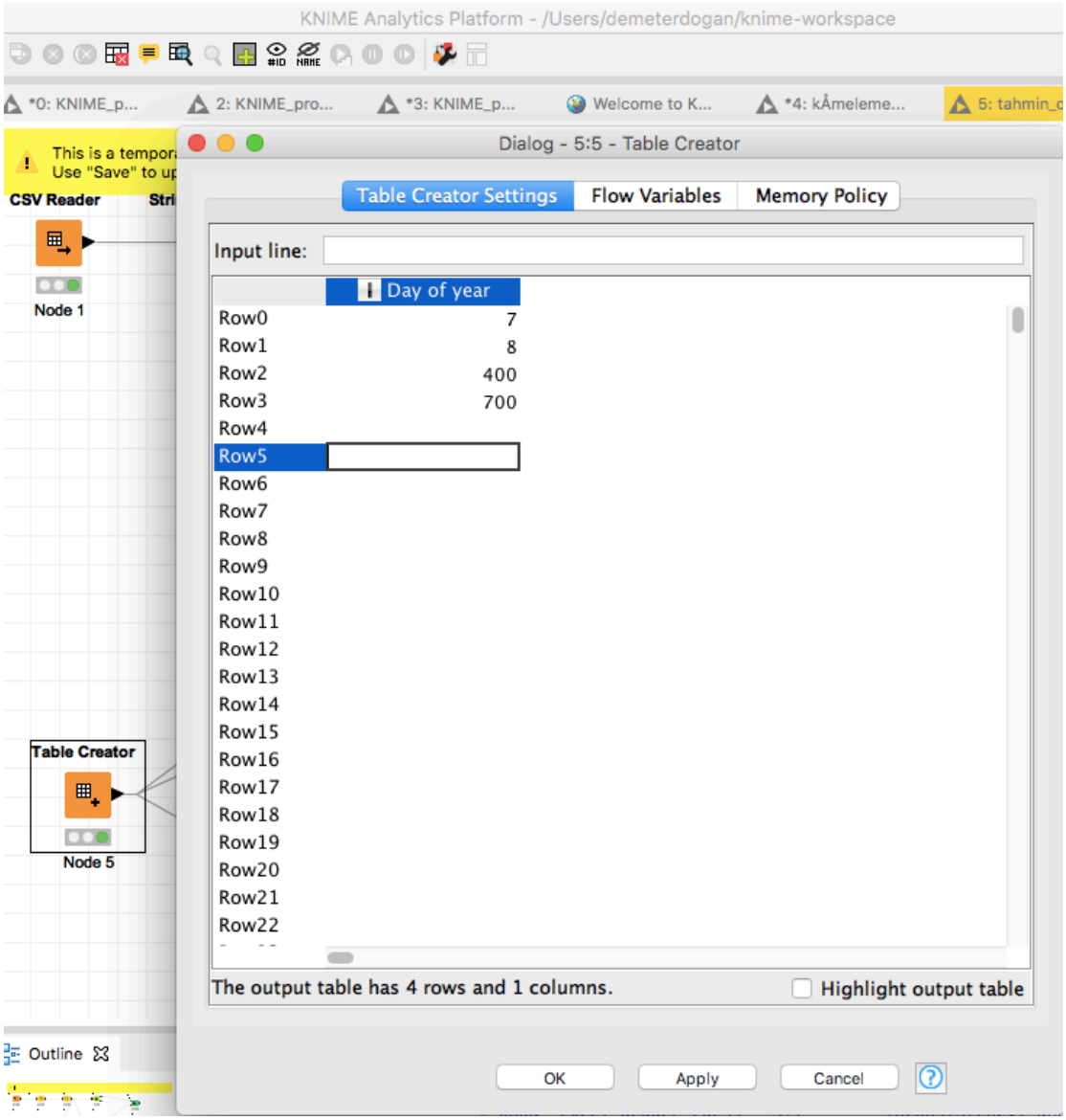
Şekil 8.13.6 linear regression learner'ın sisteme eklenmesinden sonra configure penceresinde yapılan değişikliği göstermektedir. Target olarak close yani linear regression da kapanış değerleri ve yılın hangi günü olduğunun grafiğinin çıkarılması istenmiştir. Bu yüzden diğer tüm kolonlar exclude edilmiştir. Ayrıca eksik rowlar yok sayılması için ignore rows with missing values seçeneği seçilmiştir.



Şekil 8.13.7

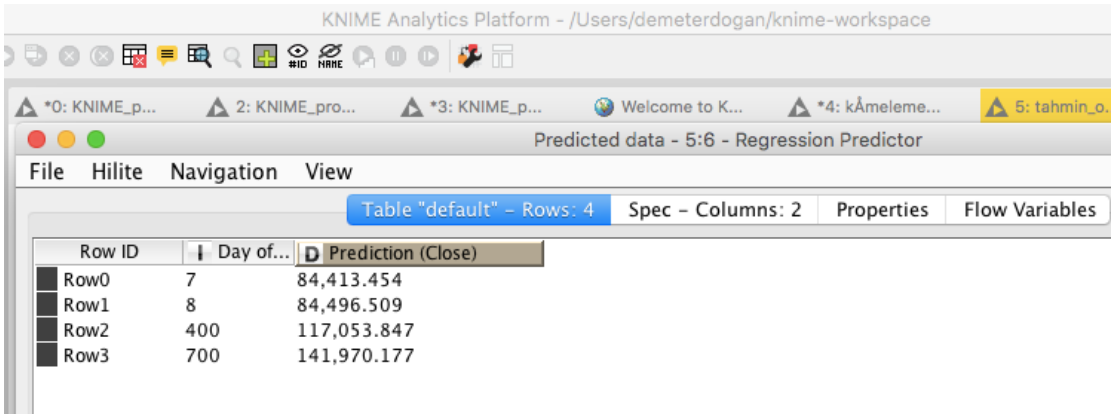
Şekil 8.13.7'de oluşturulan linear regression scatterplot grafiğini gösterilmektedir. X ekseninde yılın günleri y ekseninde ise borsanın kapanış değerleri verilmiş ve ilkel bir doğru oluşturulmuştur. Doğrunun üzerine denk gelen noktalarda hata oldukça az, doğrudan uzakta kalan noktaların ise hata payları yüksektir. Burada doğrusal linear ile öğrenme gerçekleştirilmiş olundu.

7., 8., 400. Ve 700. Günleri tahmin edilsin istenirse table creator ve regression predictor eklenerek bakılabilir.



Şekil 8.13.8

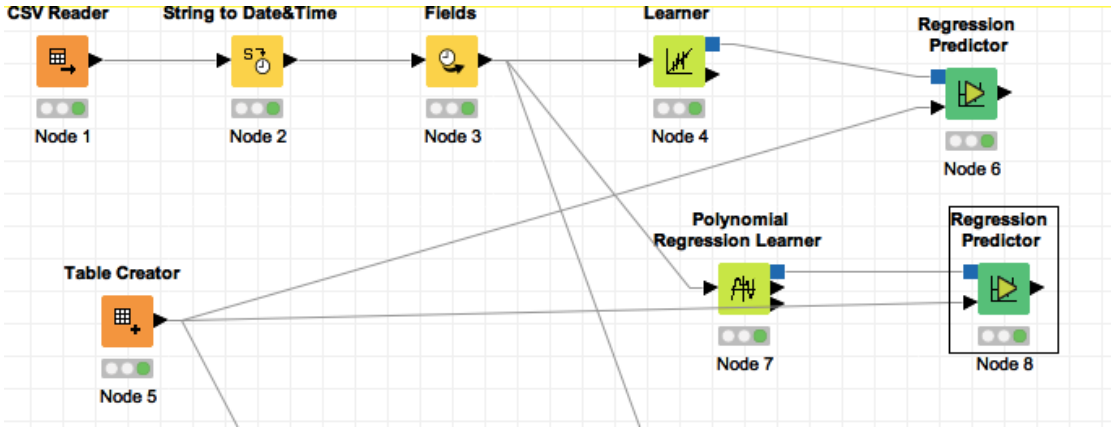
Şekil 8.13.8 table creator'da bulunmak istenilen 7,8,400 ve 700. Günler yazıldığı görülmektedir.



Şekil 8.13.9

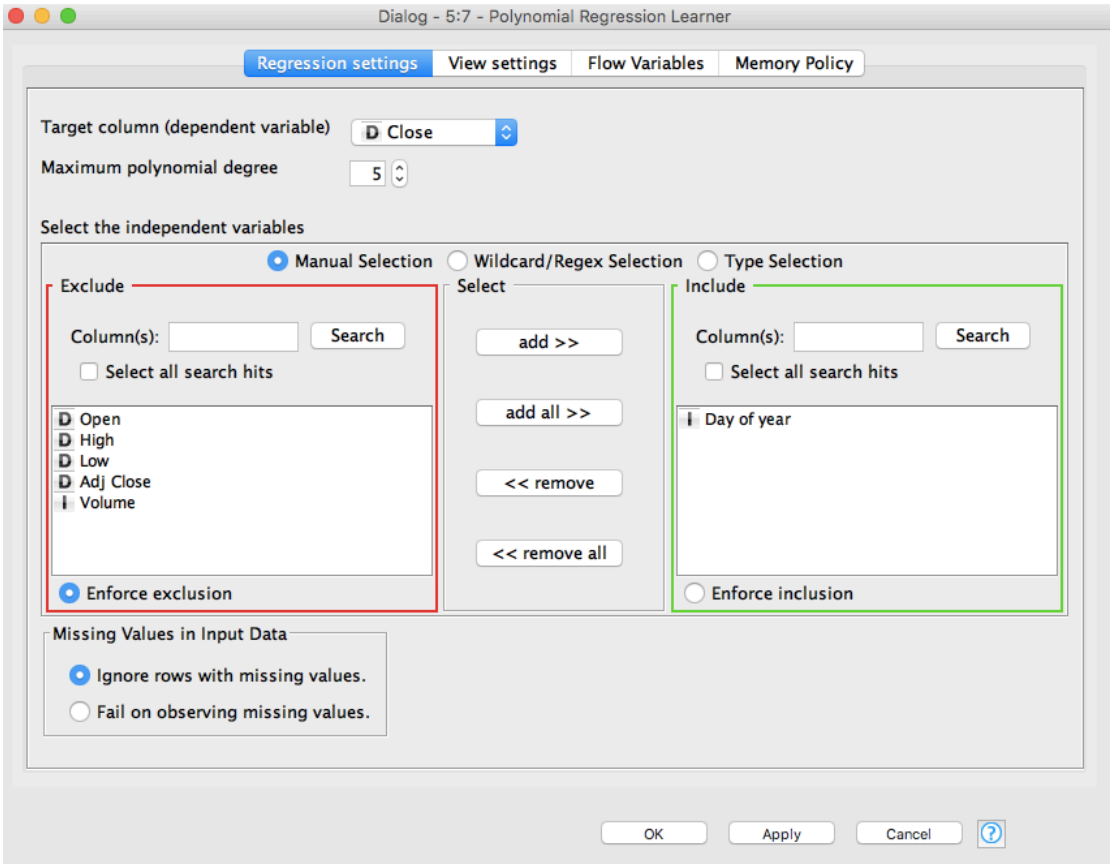
Şekil 8.13.9'da regression predictor sonucu görülmektedir. 7. Gün için 84 413, 8. Gün için 84 496 vb. değerleri tahmin edilmiştir.

Farklı regression modelleri de denenebilir.



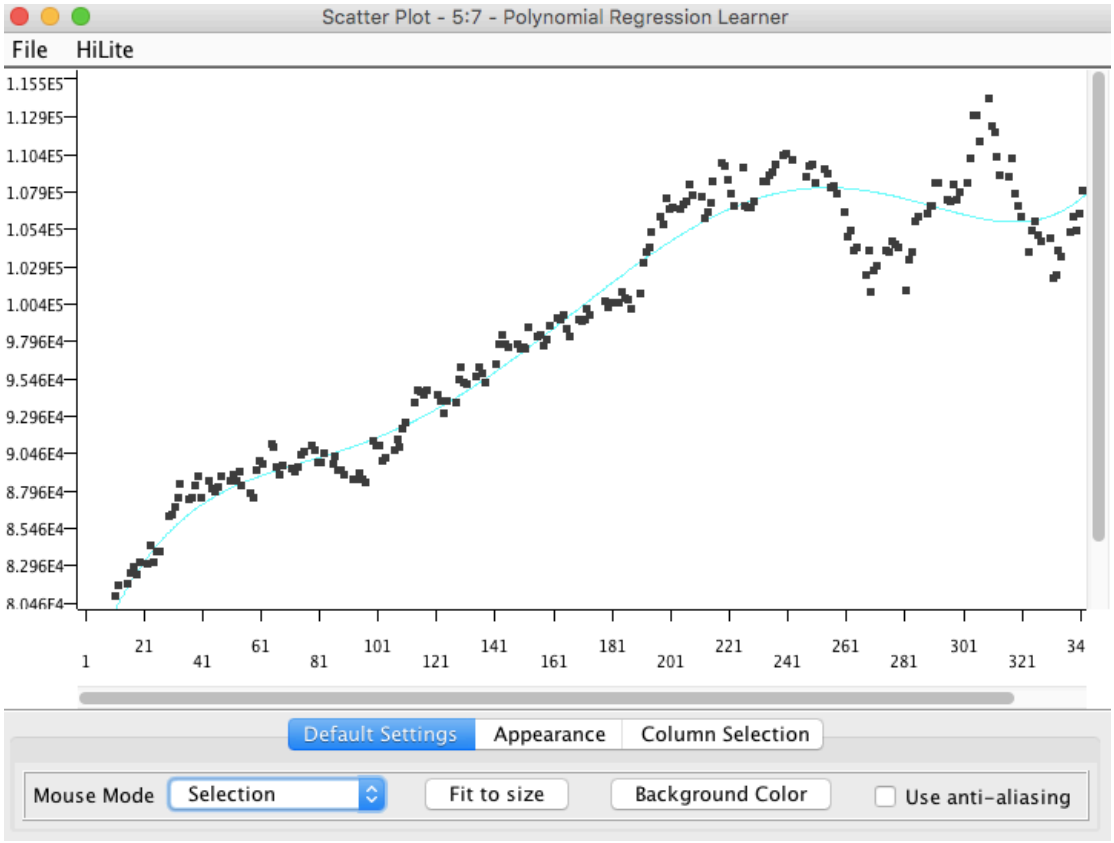
Şekil 8.13.10

Şekil 8.13.10, sisteme polynomial regression learner eklenmesini ve bağlantılarını göstermektedir.



Şekil 8.13.11

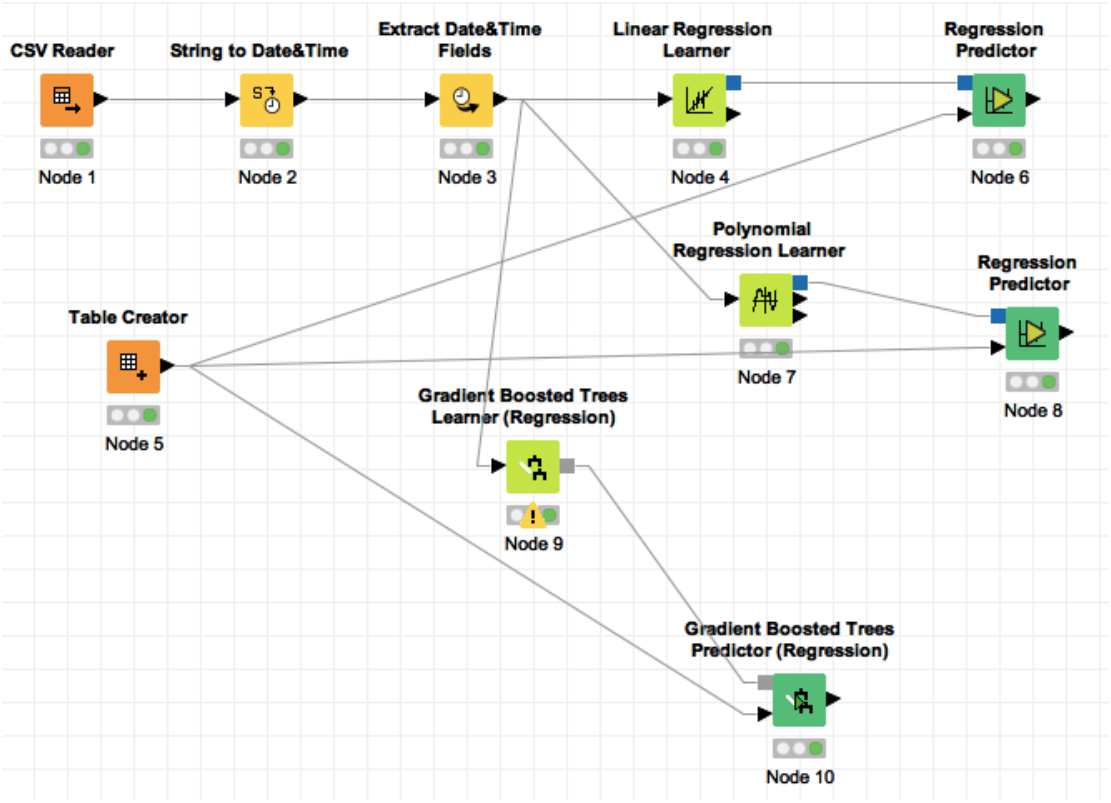
Şekil 8.13.11 bu regression için yapılan configure'ü göstermektedir. Program bu şekilde çalıştırıldığında aşağıdaki scatter plot oluşmaktadır.



Şekil 8.13.12

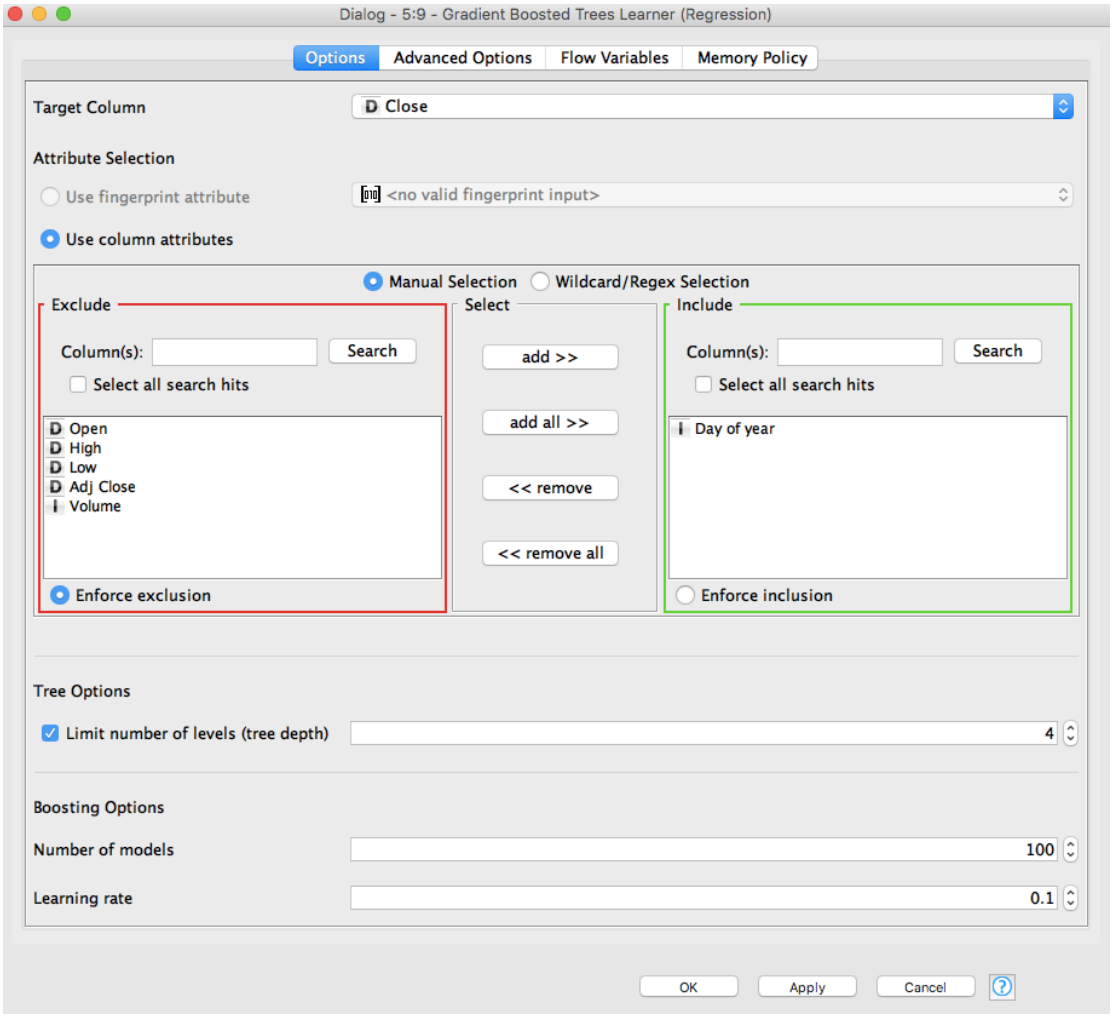
Şekil 8.13.12 polynomial regression learner'dan elde edilen scatterplot'ı göstermektedir. Noktaların çizgiye yakınlığından dolayı linear regression'a göre daha az hata payı görülmektedir.

Bir diğer deneme de gradient coosted tree üzerinde yapılabilir.



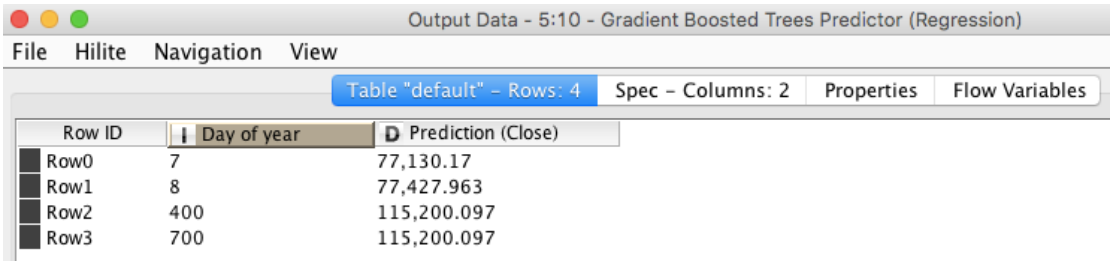
Şekil 8.13.13

Şekil 8.13.13, sisteme gradient boosted trees learner ve predictor eklenmesini ve bağlantılarını göstermektedir.



Şekil 8.13.14

Şekil 8.13.14 gradient boosted trees learner için configure bölümünü göstermektedir. Sadece day of year kolonu dahil edilip diğer kolonlar exclude yani dahil edilmemiştir. Target kolon olarak da close seçilmiştir.



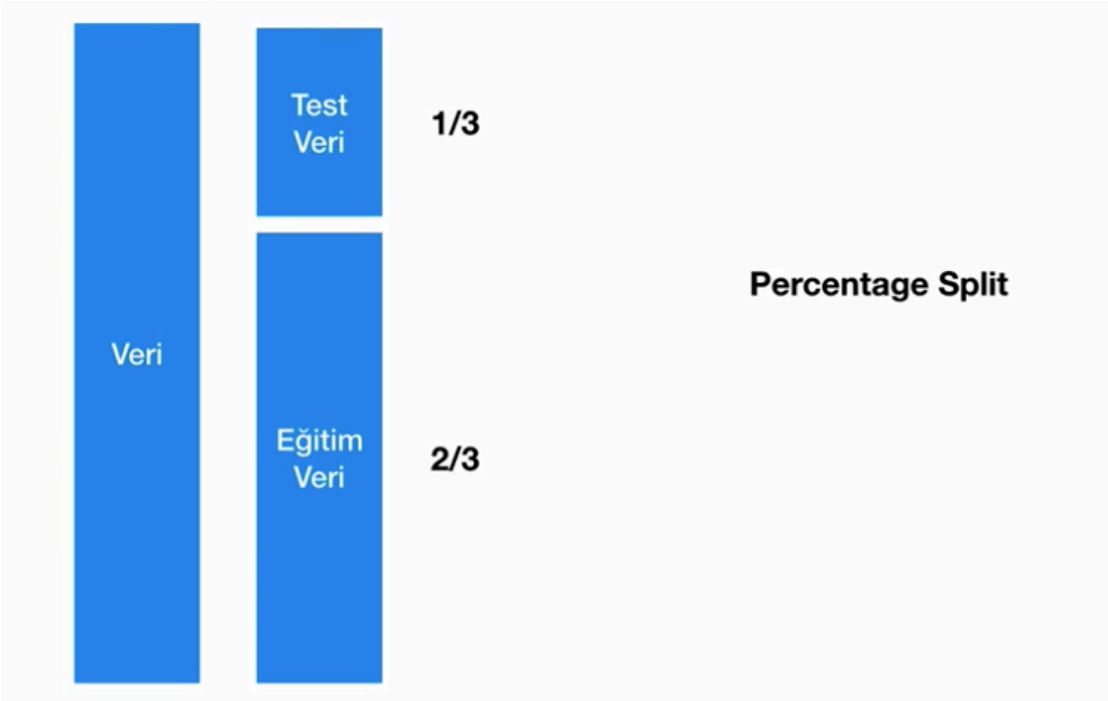
Şekil 8.13.15

Şekil 8.13.15 gradient boosted trees predictor'ın sonucunu göstermektedir. Burada da görüldüğü gibi 7. Gün için 77 130, 8. Gün için 77 427 ve son iki gün için ise aynı değerleri tahmin etmiştir.

9.BÖLÜM: BAŞARI DEĞERLENDİRME (EVALUATION)

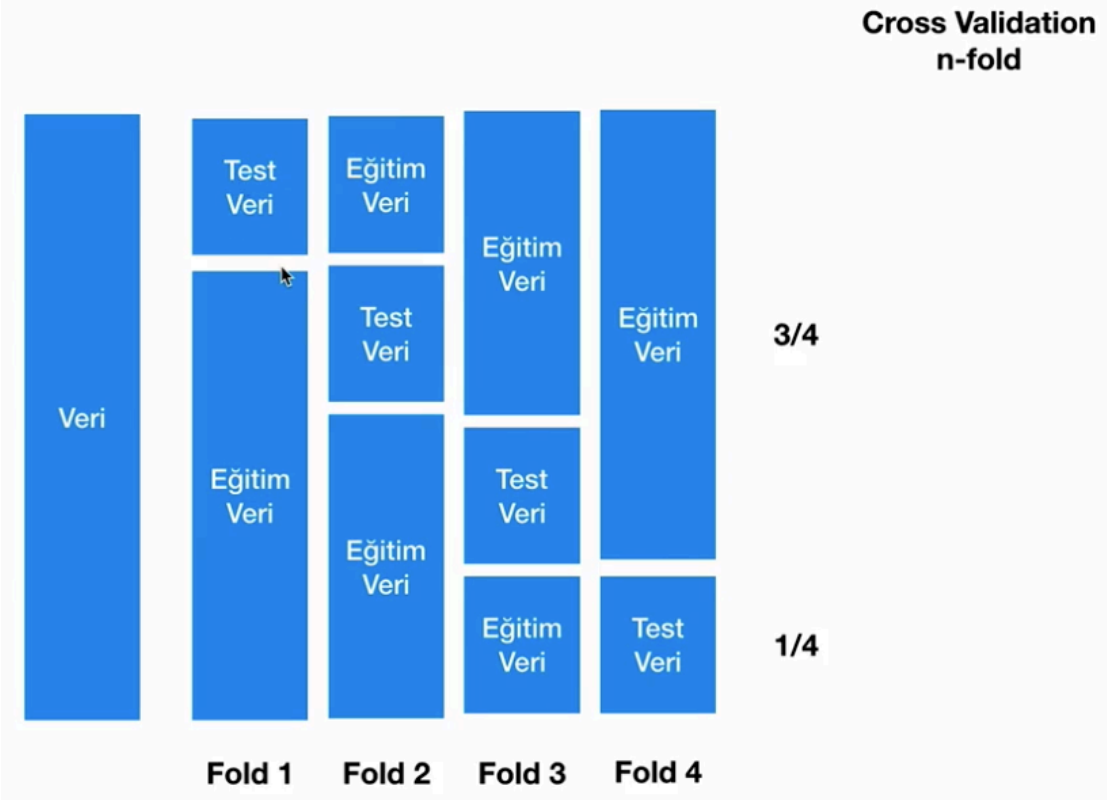
9.1 k-Katlamalı Çapraz Doğrulama (k-fold Cross Validation)

Bu bölümde amaç, sahip olunan verinın nasıl kullanılacağı ve başarısının ölçümü için teorik bilgi vermek.



Şekil 9.1.1

Şekil 9.1.1, percentage splite'e göre veriyi bölmenin ve kullanmanın oranını göstermektedir. Genellikle veri seti 1/3'e 2/3 oranında bölünür. 2/3 oranındaki veriler ile makine öğrenmesi gerçekleştirilir. 1/3 oranındaki verilerde ise bu öğrenme test edilir.



Şekil 9.1.2

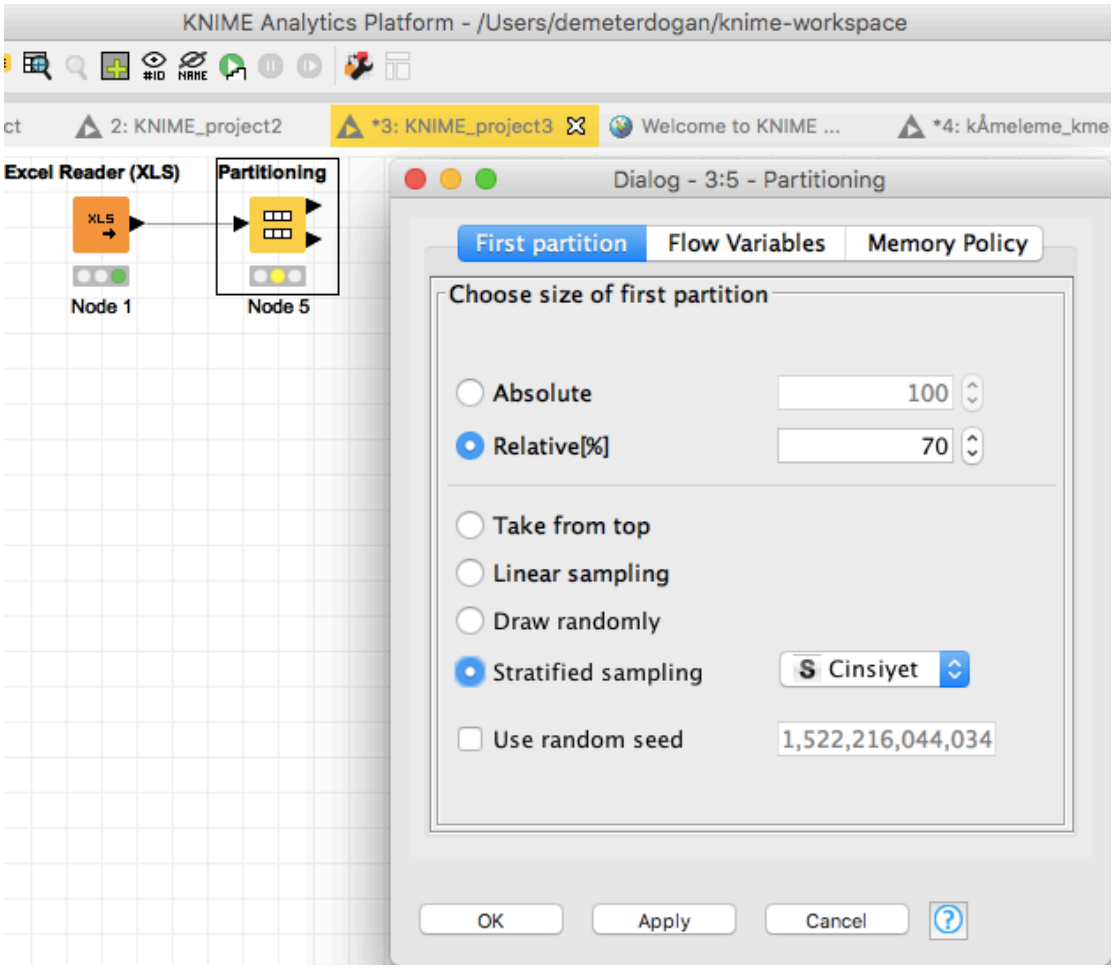
Şekil 9.1.2 Cross validation 4-fold için eğitim ve test oranlarını ve kullanım adımlarını göstermektedir. Cross validation'nın percentage splite'ten farkı verinin tümünü hem eğitimde hem de testte kullanmasıdır. Cross validation n fold 'da n kaç parçaya bölüneceğini göstermektedir. Burada n=4 olarak alınmıştır. İlk olarak tüm veri 4 parçaya bölünmüm ve ¼ test ¾ ü eğitim için kullanılmış daha sonraki adımda eğitim içerisindeki bölümden ve tüm veri setine oranı ¼ olacak şekilde veri seti test için kalan ¾ ise eğitim için kullanılmıştır. Yani veriler baştan sona kayarak hepsi ayrı ayrı test ve eğitim için kullanılır. N=4 olduğu için 4 kez bu işlem tekrarlanır ve başarı oranları çıkarılır. Sistemin başarısı bu dört başarının ortalaması alınarak bulunur.

Burada n=10 alındığında fakat 9 veri varsa cross validation kullanılamaz. Ya da test için çok az oran olursa test oranın düşüklüğü de sorun yaratabilir. Literatürde genelde en çok n değeri 10'dur.

9.2 k-Katlamalı Çapraz Doğrulama (k-fold Cross Validation) Uygulaması

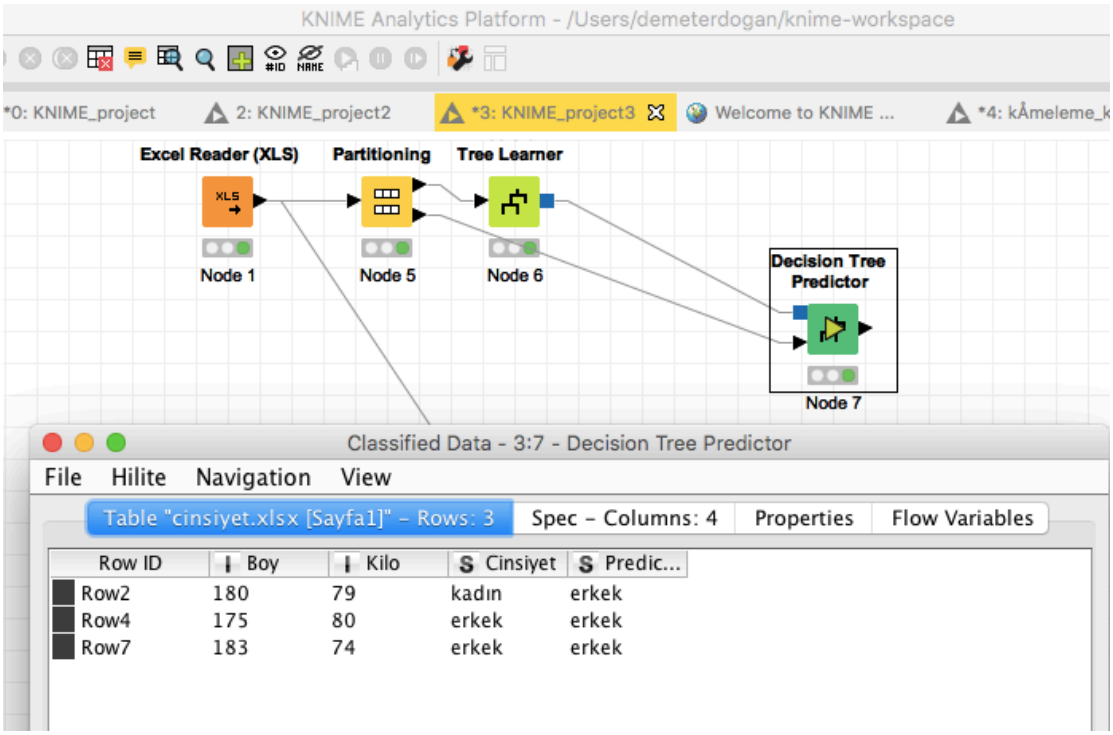
Bu bölümde amaç, bir önceki bölümde verilen teorinin Knime üzerinden uygulamasını göstermek.

Bu bölümde de yine daha önce kullanılan cinsiyet veri seti kullanılacak.



Şekil 9.2.1

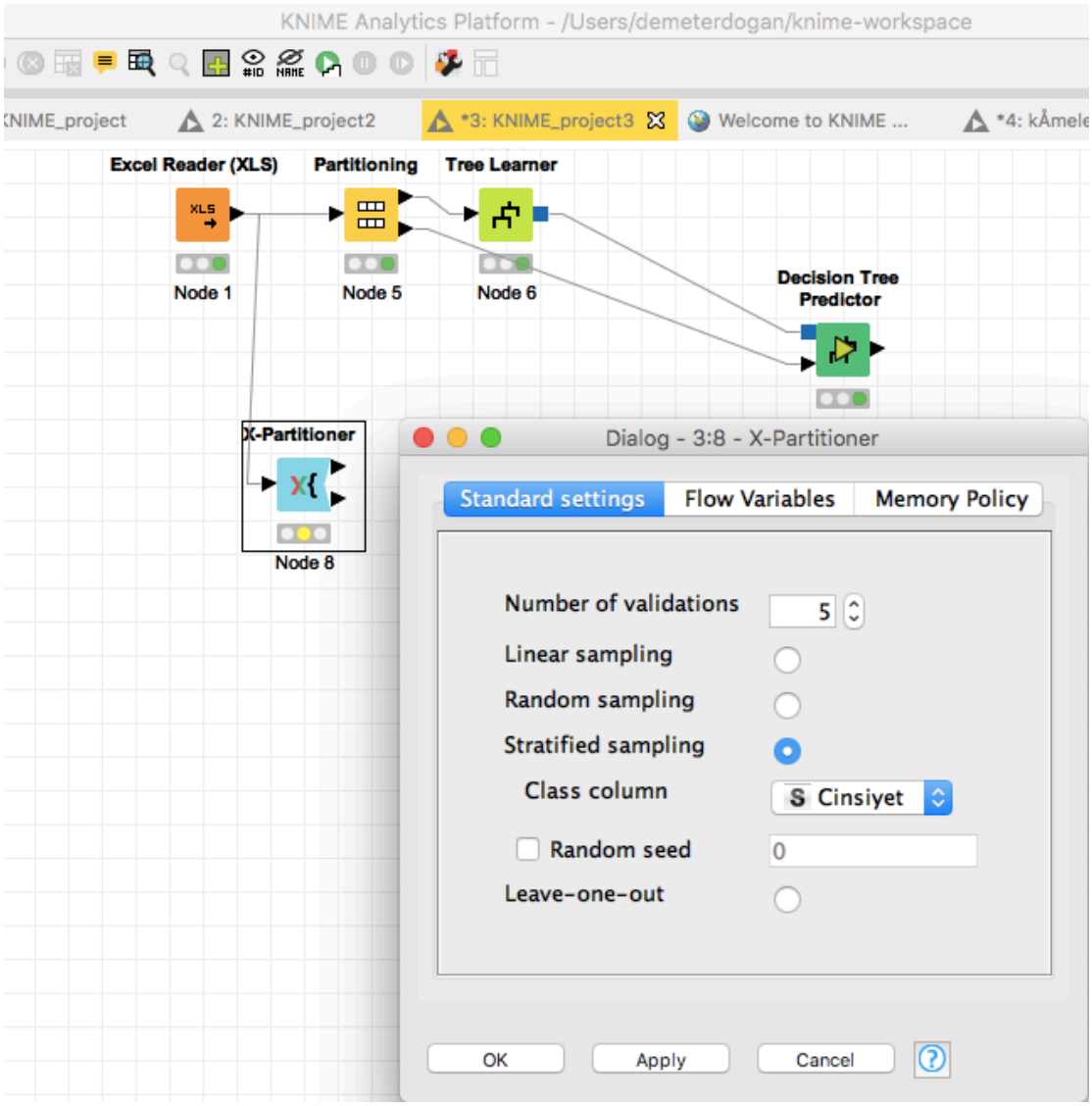
Şekil 9.2.1, excel reader'a yüklenmiş cinsiyet veri setinin partitioning operatörü ile bağlantısını ve configure penceresinde yapılan değişiklikler gösterilmiştir.



Şekil 9.2.2

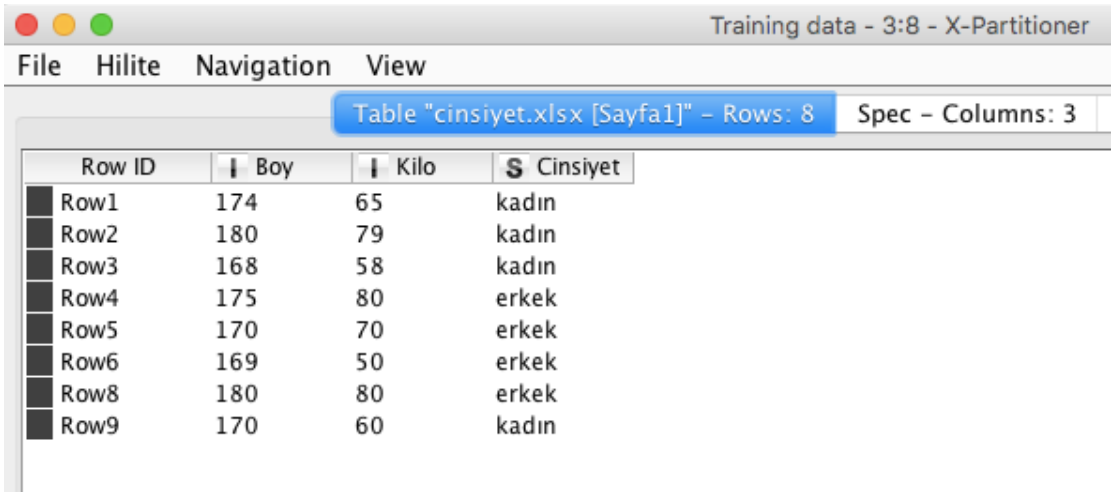
Şekil 9.2.2 decision tree learner ve predictor'ın sisteme eklenmesi ve classified sonucunu göstermektedir. Görüldüğü üzere row2 için gerçekten kadına ait olan verileri öğrenme sonucunda yanlış tahmin ederek erkek demiş ve diğer tahminleri doğru yapmıştır.

Şekil 9.2.3, bir önceki bölümde anlatılan cross validation örneğini göstermektedir.



Şekil 9.2.3

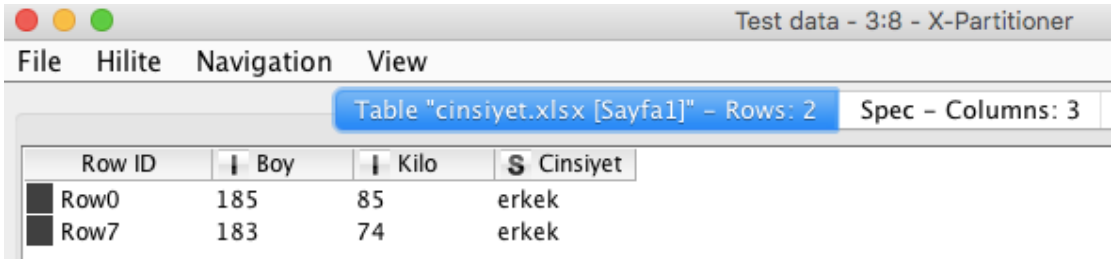
Şekil 9.2.3 x partitioner'ın sisteme eklenmesini ve configure bölümündeki değişikliği göstermektedir. Number of validation bir önceki bölümde bahsedilen n değerini yani kaç kez katlanacağını kaç işlem yapılacağını gösterir. Burada 5 değerinin verilmesi 20% oranında test ve 80% oranında öğrenme için veri setini böleceği ve 5 kez bu veri setindeki verileri kaydırarak kullanacağı yani tüm verilerin testte ve öğrenmede kullanılacağı anlamına gelir.



Row ID	Boy	Kilo	Cinsiyet
Row1	174	65	kadın
Row2	180	79	kadın
Row3	168	58	kadın
Row4	175	80	erkek
Row5	170	70	erkek
Row6	169	50	erkek
Row8	180	80	erkek
Row9	170	60	kadın

Şekil 9.2.4

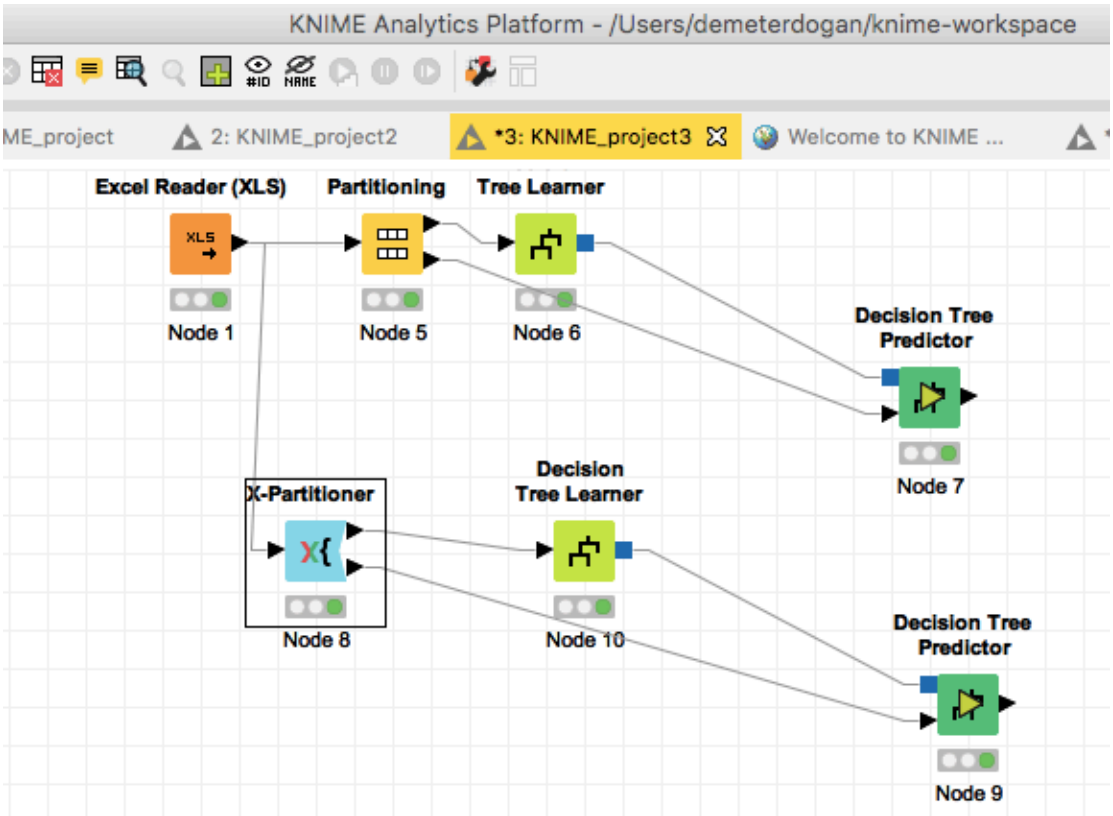
Şekil 9.2.4'teki tablo x partitioning'in training data setini göstermektedir. N=5 seçilidiği için veri setinin 80%'i training e kullanılmıştır.



Row ID	Boy	Kilo	Cinsiyet
Row0	185	85	erkek
Row7	183	74	erkek

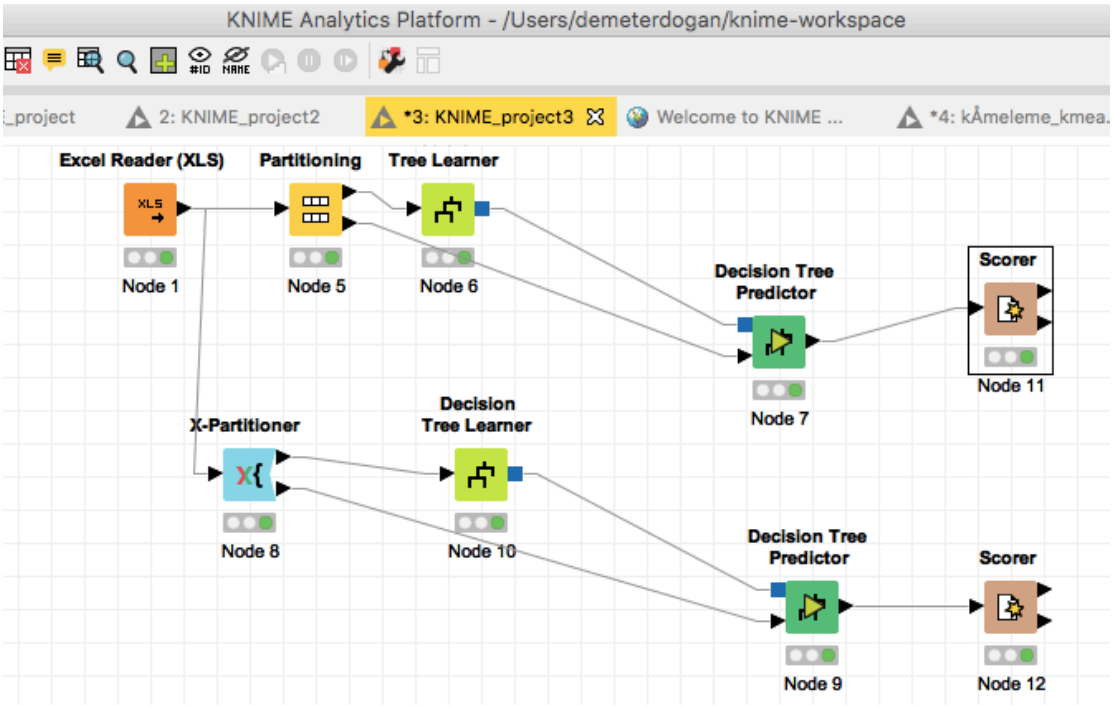
Şekil 9.2.5

Şekil 9.2.5'teki tablo x partitioning'in test data setini göstermektedir. N=5 seçilidiği için veri setinin 20%'i teste kullanılmıştır.



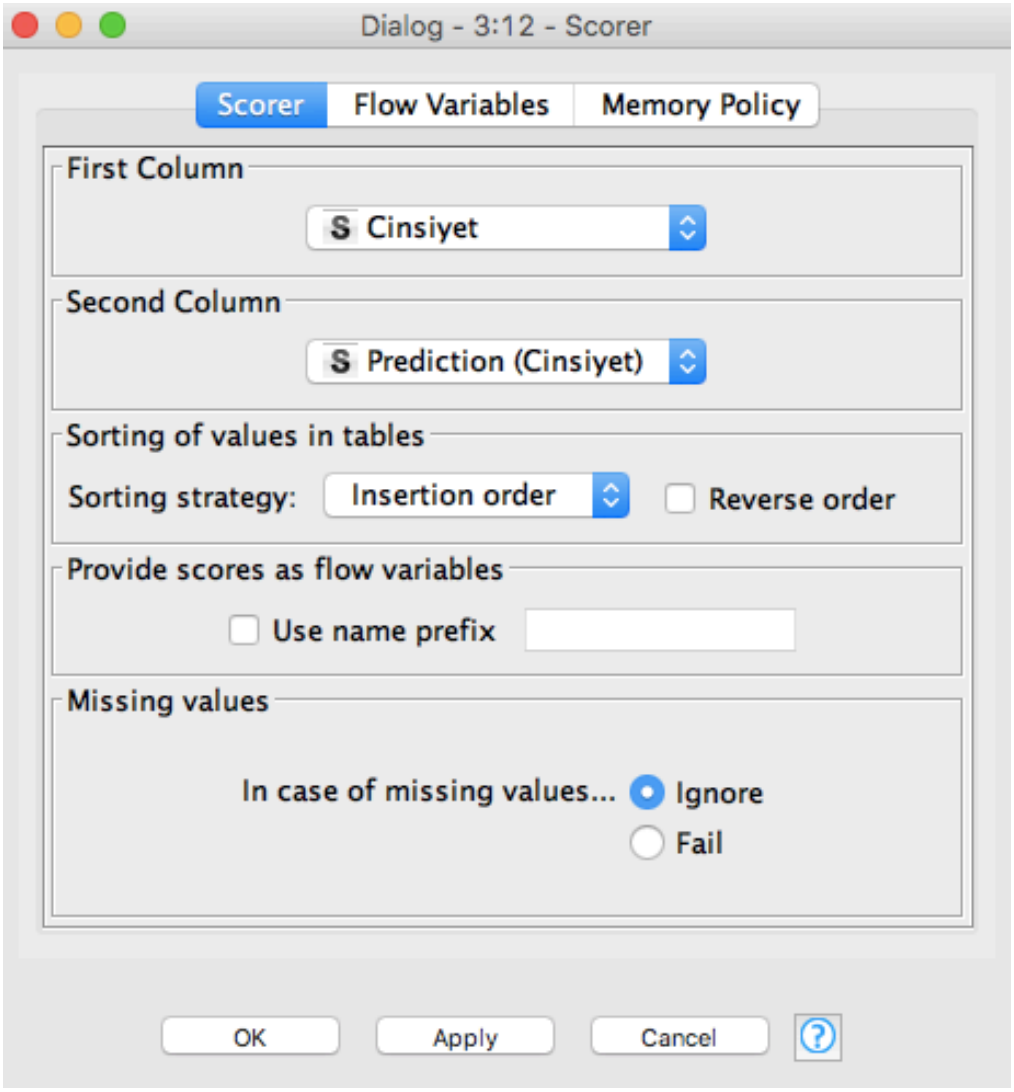
Şekil 9.2.6

Şekil 9.2.6 tüm veri kullanılarak decision tree learner ve predictor eklenip öyle öğrenme ve test için kullanılması için bağlantıları göstermektedir.



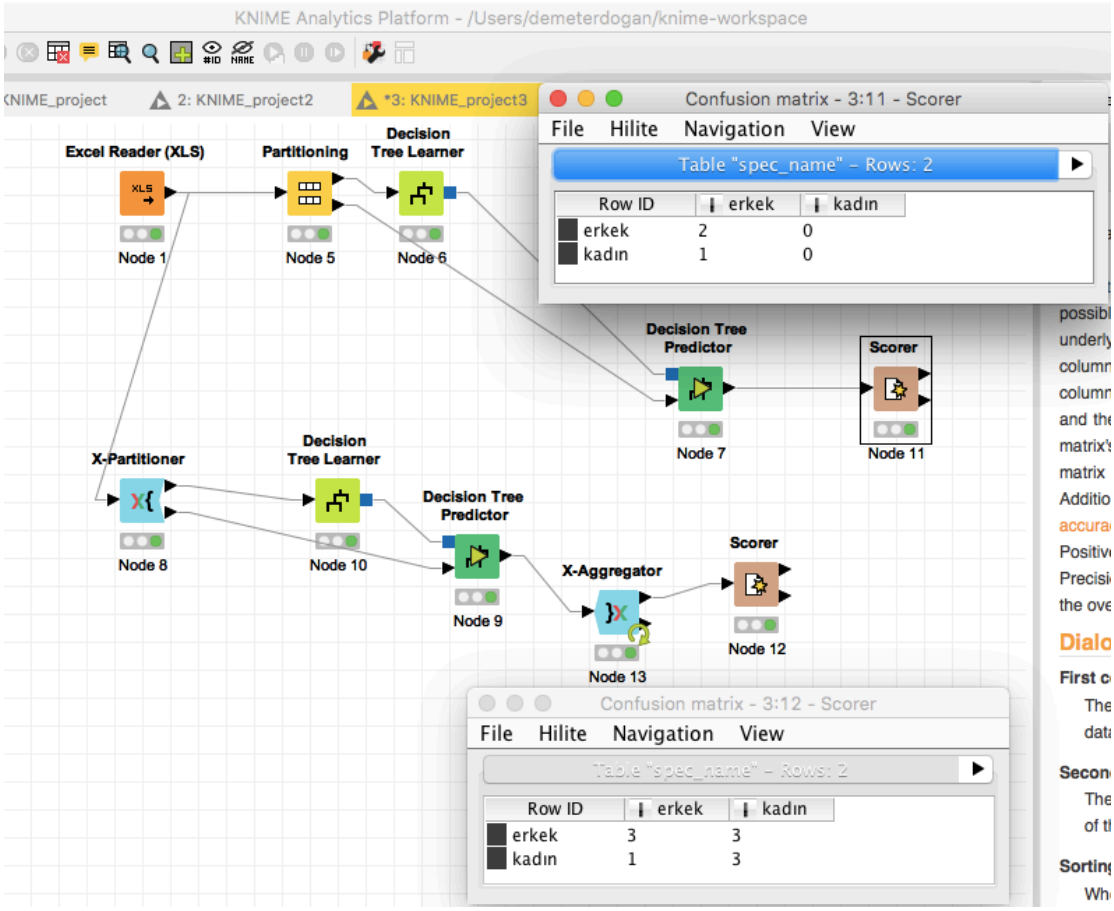
Şekil 9.2.7

Şekil 9.2.7 karşılaştırma yapılabilmesi sistemde birinci bölüme ve x partitionerdan sonraki bölümlere scorer eklenmesini göstermektedir.



Şekil 9.2.8

Şekil 9.2.8 scorer'ların configure seçeneğini göstermektedir. Cinsiyeti ve tahmin edilen cinsiyeti vermesi için first ve second column seçeneklerine cinsiyet ve prediction (cinsiyet) seçilmelidir.



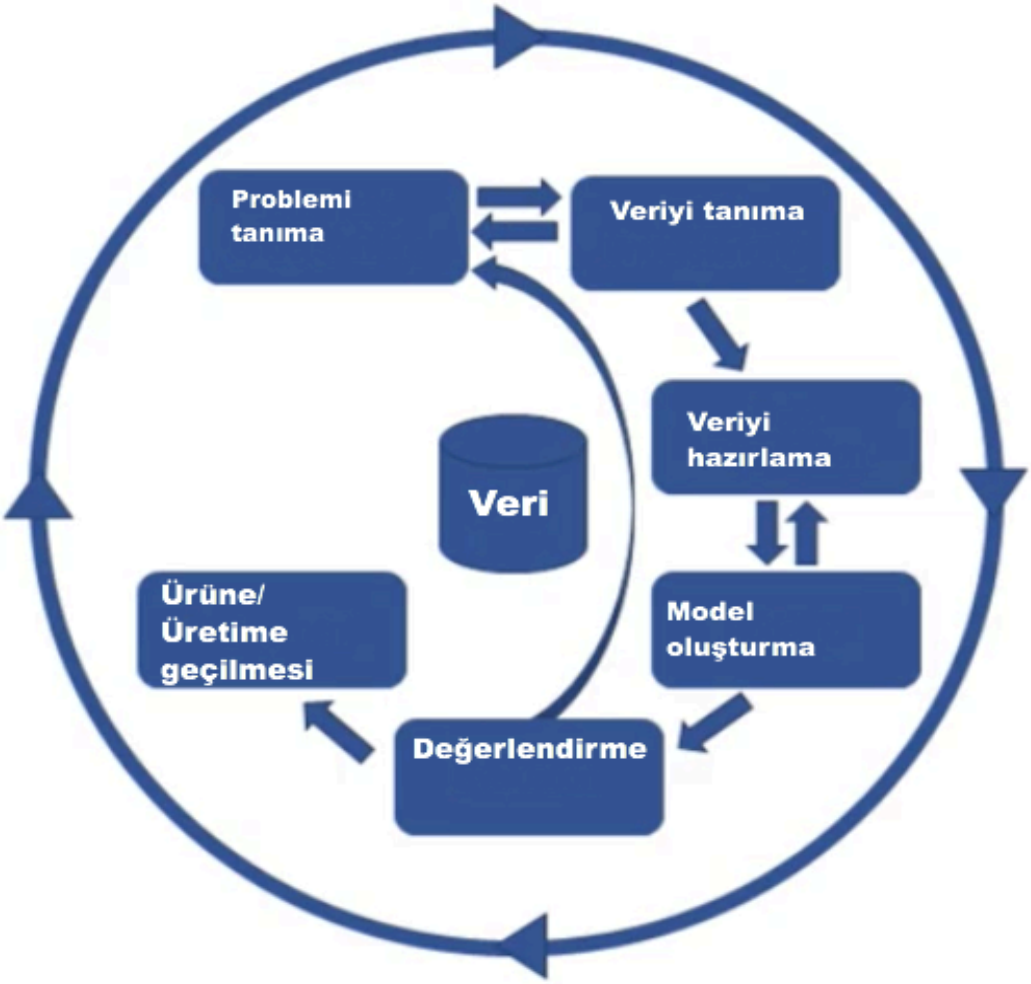
Şekil 9.2.9

Şekil 9.2.9 da üstte görülen confusion matrix partitioningten sonra yapılan decsision tree predictor sonucunu, altta görülen confusion matrix x partitioner'dan sonra yapılan predictor score'larını göstermektedir. X partitiner'daki avantaj, tüm veriler hem testtte hem de öğrenmede kullanılmıştır. Partitioning bölümünde ise veri bölünmüş ve öğrenmedeki veri testte testteki veri öğrenmede kullanılmıştır.

Ayrıca x aggregator kullanımı önemlidir ve unutulmamalıdır çünkü 5 fold seçildiği için 5 kez hesaplama yapılır. Aggregator bu 5 sonucun ortalamasını alır ve başarıyı bu ortalamadan belirler.

9.3 Confusion Matrix, Precision, Recall, Sensitivity, Specificity

Bu bölümde amaç Crisp-Dm'in son aşamalarından olan evaluation aşamaasından bahsetmek.



Şekil 9.3.1

Şekil 9.3.1'de daha önceki bölümlerde açıklanan adımlardan ve avaluation aşaması gösterilmektedir. Problemi anladıktan, veriyi anladıktan, veriyi hazırladıktan ve modellemesinden sonra bu aşamaların değerlendirilmesi gerekmektedir. Bu değerlendirme dünyadaki bu işi bilen herkesin anlayabileceği ortak bir değerlendirme formatıyla olmalıdır.

	C_1	C_2
C_1	Gerçek pozitif	Yanlış negatif
C_2	Yanlış pozitif	Doğru negatif

Tablo 9.3.2

Tablo 9.3.2, daha önce bahsedilen diagonal matrix'i göstermektedir. Yani sol yukarıdan sağ aşağı doğru çapraz şeklinde gelen hücrelerdeki bilgiler doğru tahmin edilenlerin sayısını vermektedir. Örneğin C1 olup C1 tahmin edilene true positive, C1 olup C2 şeklinde yanlış tahmin edilene False negative, C2 olup C1 şeklinde yanlış tahmin edilene False positive ve son olarak C2 olup C2 tahmin edilene true negatif olarak sınıflandırılır.

Tp → true positive,

FN → false negative,

FP → false positive,

TN → true negative şeklinde kısaltılabilir.

sınıflar	Bilgisayar_alımı=evet	Bilgisayar_alımı=hayır	toplam	Teşhis(%)
Bilgisayar_alımı=evet	6954	46	7000	99.34
Bilgisayar_alımı=hayır	412	2588	3000	86.27
toplam	7366	2634	10000	95.52

Tablo 9.3.3

Tablo 9.3.3 daha önceki bölümlerde verilen bilgisayar satın alma örneğinin sonuç tablosunu göstermektedir.

Yaş	Gelir Düzeyi	Öğrencilik Durumu	Kredi notu	Bilgisayar alımı
<= 30	Yüksek	Hayır	Uygun	Hayır
<= 30	Yüksek	Hayır	Çok iyi	Hayır
31...40	Yüksek	Hayır	Uygun	Evet
>40	Orta	Hayır	Uygun	Evet
>40	Düşük	Evet	Uygun	Evet
>40	Düşük	Evet	Çok iyi	Hayır
31...40	Düşük	Evet	Çok iyi	Evet
<= 30	Orta	Hayır	Uygun	Hayır
<= 30	Düşük	Evet	Uygun	Evet
>40	Orta	Evet	Uygun	Evet
<= 30	Orta	Evet	Çok iyi	Evet
31...40	Orta	Hayır	Çok iyi	Evet
31...40	Yüksek	Evet	Uygun	Evet
>40	Orta	Hayır	Çok iyi	Hayır

Tablo 9.3.4

Tablo 9.3.4 bilgisayar satın alma örneğinin Naive Bayesian için kullanılmadan önceki halini göstermektedir.

Tablo 9.3.3 bilgisayar alması beklenenler bilgisayar_alımı=evet şeklinde gösterilmektedir.

Accuracy (doğruluk) → toplam örneklerin kaçının doğru sınıflandırıldığıdır.

Örneğin; toplamda 10 000 örnek var ve toplam

doğruluk = (6954+ 2588)/ 10000 = 95.52% 'dir.

Sensitivity (duyarlılık)→ gerçek pozitif / (gerçek positif + yanlış negatif) oranıdır.

Örneğin, (6954+46)/ 7000 = 99.34 % sensitivity'i vermektedir.

Bilgisayar alma olasılığı yüksek kişilerin ne oranda doğru tahmin edildiğidir.

Specificity (belirginlik) → gerçek negatif / (gerçek negatif + yanlış positif) oranıdır.

Örneğin, 2588/(412+2588) = 86.27%

Bilgisayar almama olasılığı yüksek kişilerin ne oranda doğru tahmin edildiğidir.

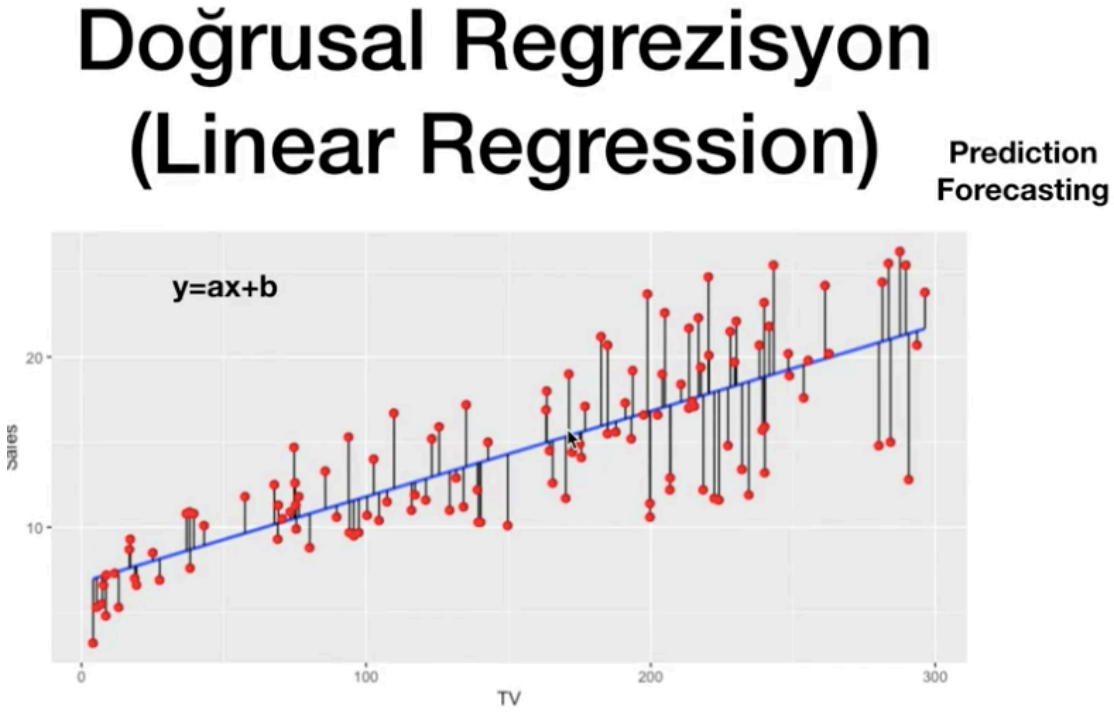
Precision (duyarlılık)→ gerçek pozitif / (gerçek pozitif + yanlış positif)

Örneğin, 6954/(6954+412)

Sınıflandırma sonucunun ne kadar doğru olduğunu göstermektedir.

Herkesin anladığı terimlerden kullanılabilme amacıyla sensitivity, precision, specifity kullanılır.

Prediction hatalarında farklı yöntemler kullanılır.



Şekil 9.3.5

Şekil 9.3.5 linear regression için hataları göstermektedir. Örneğin bu modelde hatalar, verilerin (noktaların) ortadan çizilen doğruya ne kadar uzaklıkta olduğu ile ilgilidir. Burada problem eksi ve artı hataların olduğudur. Örneğin doğru üzerinde kalması ya da altında kalması yüzünden + ve – ler çıkar ve bunları toplayaınca birbirini sıfırlayabilir. Bu yüzden mutlak değer ile ifade edilmeli ve öyle hesaplamalara katılmalıdır.

Mutlak hata (absolute error): $|y_i - y_i'|$

Hatanın karesi: $(y_i - y_i')^2$

Test hatası (genelleme hatası) : test kümesi üzerinde ortalama kayıp

Ortalama mutlak değer: $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$

Ortalama hata kareleri: $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$

Göreli mutlak hata: $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \ddot{y}_i|}$

Göreli hata kareleri: $\frac{\sum_{i=1}^{d^d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - y_i')^2}$

Ortalama hatanın karesi aykırı değerin varlığını abartır

Yukarıda formülleri verilmiş error tipleri bu artı ve eksi problemini gidermek için kullanılan formüllerdir.

Bu bölümde, iki türlü hata ölçümünden bahsedilmiştir. Birincisi classifier (sınıflandırmada) ikincisi ise prediction'da (tahminde) hata nasıl ölçülür örnek ile açıklanmıştır.

9.4 Bölütleme (Kümeleme, Clustering) Değerlendirilmesi: Saflık (Purity), Rendindex

Bu bölümde amaç clustering (bölütleme, kümeleme) algoritmalarını açıklamak hem de clustering'in nasıl yapıldığını göstermektir. Clusterin uzaya dağılmış verilerin kümelenmesidir. Veri kümesinde kaç kolon var ise o sayıda boyut düzlemde dağılır. Örneğin boy, kilo ayak numarasına göre cinsiyet örneğinde 3 boyutlu düzlem kullanılmalı.

Clustering algoritmaları sınıflandırmaları;

Monothetic: Bir ortaklık üzerinden gider. Örneğin yaşa göre olunca gençler, orta yaşlılar ve yaşlılar şeklinde gruplanma.

Polythetic: Aradaki mesafeler önemlidir.

Şekil 9.4.1

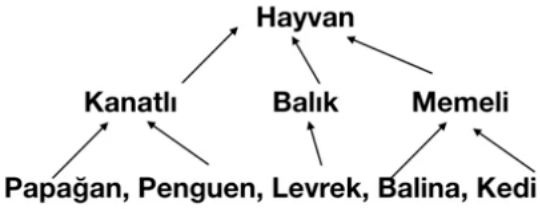
Şekil 9.4.1'de görülen grafikteki noktalar arasındaki uzaklık polythetic sınıflandırma için örnek verilebilir. Noktaların birbirine ve düzlemlere uzaklığına bakılır.

Hard Clustering: Düzlemde verilerin kesin olarak birbirinden ayrıldığı clustering'tir. Kesişim kesinlikle yoktur.

Soft Clustering: Bir üyenin birden fazla gruba dahil olabileceği clustering çeşididir.

Düz: Belli bir gruba dahil olma. Örneği yeni gelen verinin bir gruba dahil edilmesi.

Hiyerarşik: Taxonomy



Şekil 9.4.2

Şekil 9.4.2'de hiyerarşik gruplama örneği gösterilmektedir.

Papağan, penguen → Kanatlı

Levrek → Balık

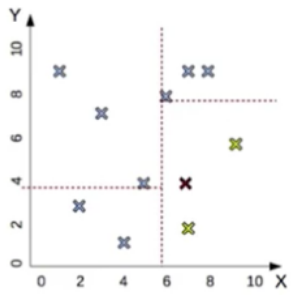
Balina, kedi → Memeli

Kanatlı, Balık, Memeli → Hayvan

Clustering Algoritmaları;

KD-Tree: Monothetic, hard, hierarchical

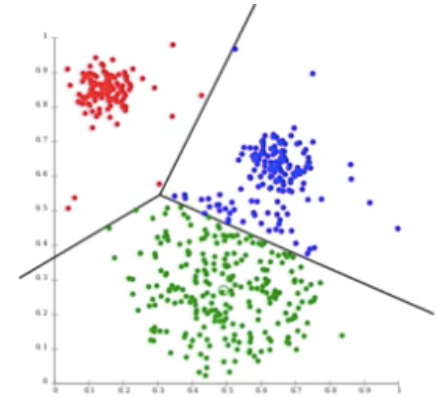
Komşuluk yakınlıklarına göre ağaç oluşturulur. Sınırlar öğrenilir. Örnekler arasında sabit bir özellik üzerinden işlem çalışmaktadır.



Şekil 9.4.3

Şekilde düzlem, x ekseninden 2 düzleme, y'den totalde 3 düzleme bölünmüştür.

K-Means: En çok kullanılan algoritmalardan biridir. Polythetic (noktaların arasındaki mesafelere göre çalışır), hard (grupllaşma arasındaki sınırlar net bir şekilde belli), flat (üyeler arasında bir hiyerarşi yok)



Şekil 9.4.4

Şekil 9.4.4 K-means algoritması için örnek bir düzlemdir.

Gaussian (EM): Polythetic, (noktaların arasındaki mesafelere göre çalışır) hard (grupllaşma arasındaki sınırlar net bir şekilde belli), hierarchical

Cluster Evaluation:

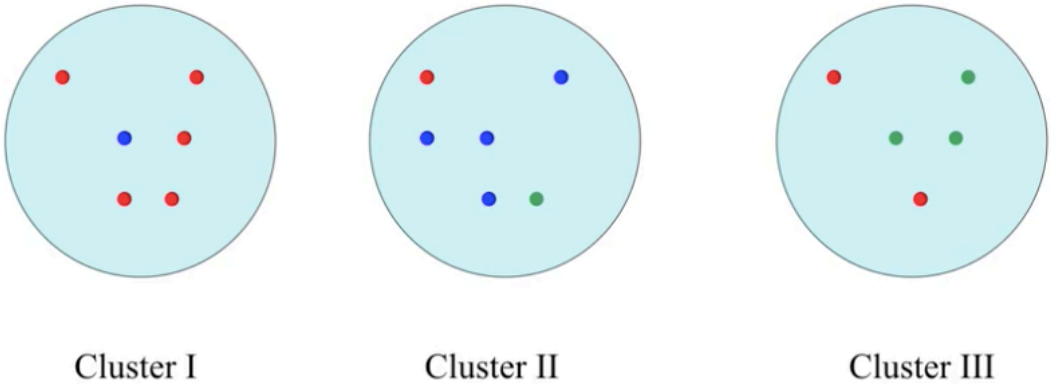
Cluster evaluationda önemli 2 temel vardır.

Harici (external)

Dahili (internal): Küme içerisindeki elemanların mesafesine göre

- Aynı küme içerisindeki elemanların birbirine olan mesafelerinin toplamının minimum olması
- Farklı küme içerisindeki elemanların birbirine olan mesafelerinin toplamının maksimum olması

Ölçümleri; Purity, Entropy, F-Measure



Şekil 9.4.5

Şekil 9.4.5 cluster I, II ve III için purity şemalarını göstermektedir.

Cluster I: 1/6 (max(5,1,0) =5/6

Cluster II: 1/6 (max(1,4,1) =4/6

Cluster III: 1/6 (max(2,3,0) =3/6

6 renk olduğu için 6 çeşit veri tipi, birincisinde 5 kırmızı 1 mavi olduğu için max(5,1,0) 1/6 ile çarpılır. Diğerleri de bu şekilde hesaplanır. Her bir claster için olan püre değerleri bunlardır.

Nokta sayıları	Kümelemede aynı grupta olan	Kümelemede farklı grupta olan
Zemin bazlıda aynı grupta	A	C
Zemin bazlıda farklı grupta	B	D

Tablo 9.4.6

Şekil 9.4.6, Rand index formülü için tabloyu göstermektedir.

R= A/A+C

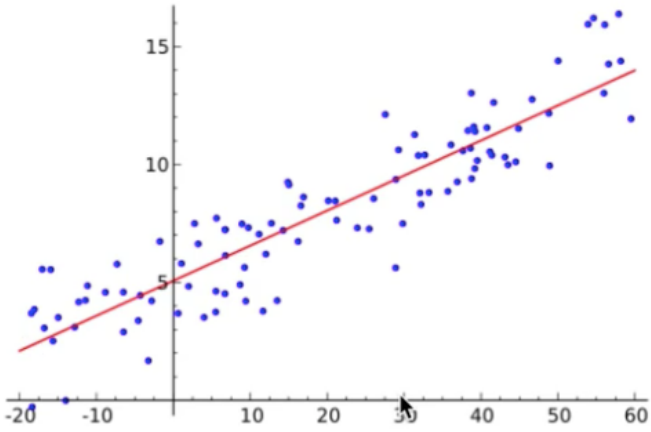
Precision (hassasiyet):

P= A/A+B

RI= A+D /A+B+C+D

9.5 Tahmin (Prediction) Değerlendirmesi, RMSE, RMAE, MSE, MAE

Bu bölümde amaç prediction'nın (tahminin) değerlendirilmesini açıklamaktır. Burada bahsi geçen tahmin numeric değerlerin tahminidir. Dört yöntem kullanılmaktadır.



Şekil 9.5.1

Şekil 9.5.1'de görülen kırmızı çizgi tahmini oluştural değerlerin oluşturduğu doğrudur. Mavi noktalar ise gerçek değerleri göstermektedir. Linear regression'a tabi tutularak çıkarılmış bir çizgidir.

Örneğin; şekle göre 20 değeri için model 8'i (prediction value) göstermekte fakat geçeği ise 8.2 görülmektedir. Tahmin ve gerçek arasındaki fark hatayı oluşturmaktadır.

Mean Absolute Error:

$$\frac{|p_1 - a_1| + \ldots\ldots + |p_n - a_n|}{n}$$

Formüle göre p→predicted (tahmin), a→actual (gerçek) Değerlerin farkının mutlak değerlerde toplamı / toplam sayıdır. Daha öncede bahsedildiği gibi eksi ve artı değerler çıkacağı için nötrlemesin diye mutlak almak bu sorunu çözer.

Roor Mean Square Error:

$$\sqrt{\frac{(p_1 - a_1)^2 + \ldots\ldots + (p_n - a_n)^2}{n}}$$

Yukarıda ile aynı gibi düşünülebilir. Mutlak değer alınması yerine karesi alınıp sonra da kareden kurtulmak için karekök içerisine alınır.

Relative Absolute Errror:

$$\frac{|p_1 - a_1| + \ldots\ldots + |p_n - a_n|}{|a_1 - \hat{a}| + \ldots\ldots |a_n - \hat{a}|}$$

Görece olarak hata hesaplanır. Örneğin 8'den geçen ortalama bir değer (tüm verilerin toplamının ortalaması), gerçek veriler arasında bu 8'e en yakın değer ile en uzak değer arasındaki fark aynı değildir.

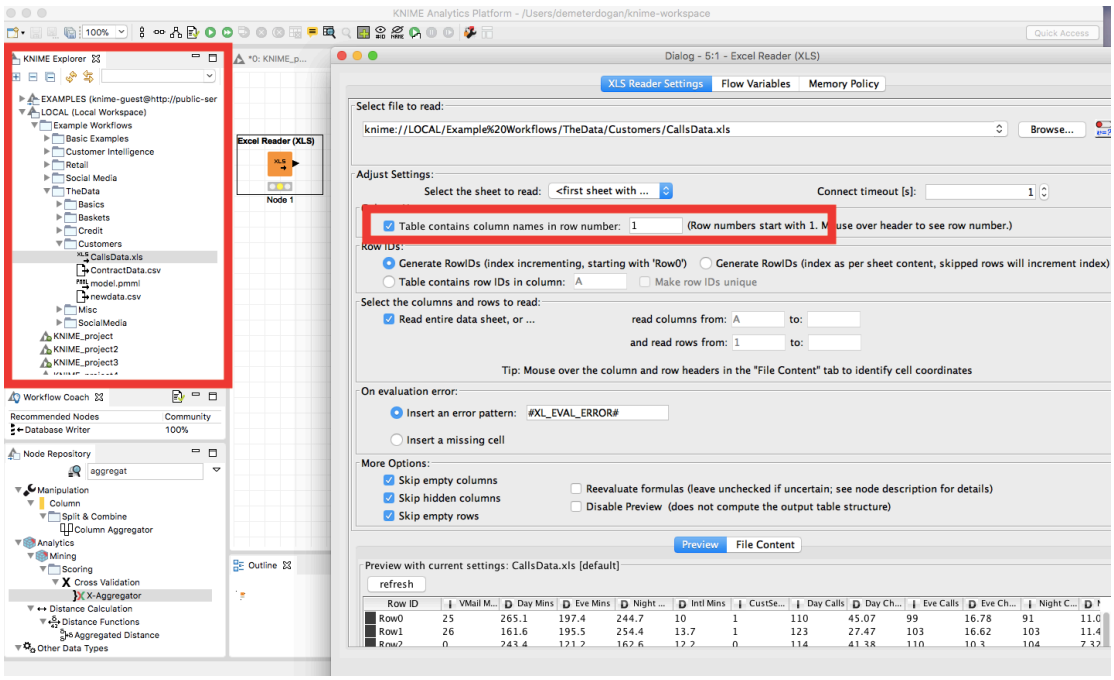
Root Relative Squared Error:

$$\sqrt{\frac{(p_1 - a_1)^2 + \ldots\ldots + (p_n - a_n)^2}{(a_1 - \hat{a})^2 + \ldots\ldots (a_n - \hat{a})^2}}$$

Bir yukarıdaki ile yanıdır fakat burada sadece ortalama değere olan uzaklık mutlak içerisine alınmak yerine negatiflerden kurtulmak için karesi alınmaktadır ve daha sonra da karekökü alınmaktadır.

9.6 Knime ile Tahmin (Prediction) Değerlendirilmesi (Evaluation)

Bu bölümde amaç, regression algoritmasının sayısal değer üzerinde nasıl değerlendirildiğini göstermek.

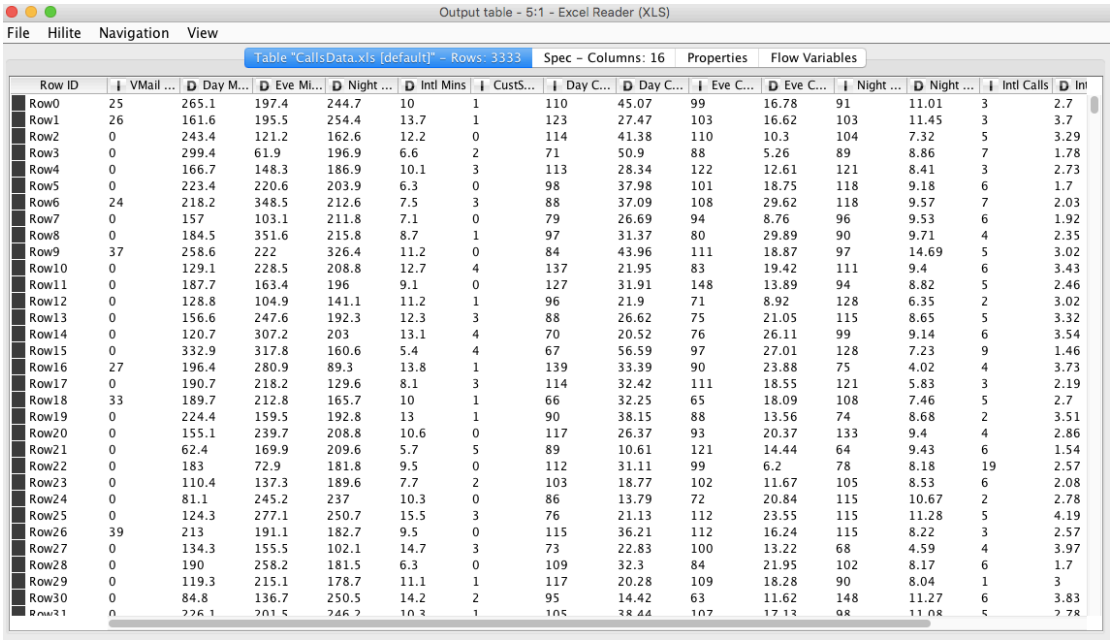


Şekil 9.6.1

Şekil 9.6.1, bu bölümde kullanılacak örnek veri setinin sisteme nereden aktarılacağını ve configure'ünü göstermektedir.

Knime explorer→ Local→ example workflows → theData→ customers→ callsData

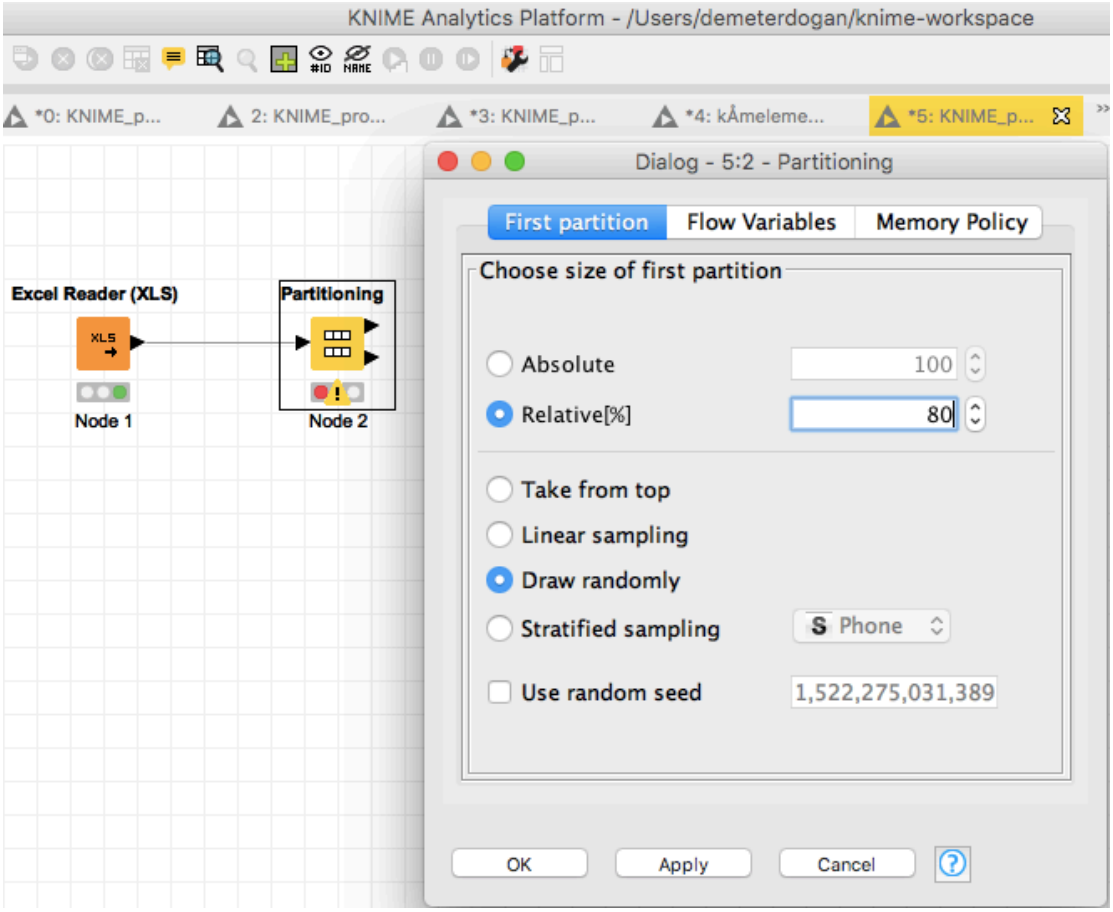
Adımları sırasıyla takip ederek callsData ismindeki excel formatındaki veri setine çift tıklandığında veri seti sisteme direk aktarılmış olur. Daha sonra configure seçeneğinde ilk row başlık olabilmesi için seçim yapılır.



Row ID	VMail ...	Day M...	Eve Mi...	Night ...	Intl Mins	CustS...	Day C...	Day C...	Eve C...	Eve C...	Night ...	Night ...	Intl Calls	Int
Row0	25	265.1	197.4	244.7	10	1	110	45.07	99	16.78	91	11.01	3	2.7
Row1	26	161.6	195.5	254.4	13.7	1	123	27.47	103	16.62	103	11.45	3	3.7
Row2	0	243.4	121.2	162.6	12.2	0	114	41.38	110	10.3	104	7.32	5	3.29
Row3	0	299.4	61.9	196.9	6.6	2	71	50.9	88	5.26	89	8.86	7	1.78
Row4	0	166.7	148.3	186.9	10.1	3	113	28.34	122	12.61	121	8.41	3	2.73
Row5	0	223.4	220.6	203.9	6.3	0	98	37.98	101	18.75	118	9.18	6	1.7
Row6	24	218.2	348.5	212.6	7.5	3	88	37.09	108	29.62	118	9.57	7	2.03
Row7	0	157	103.1	211.8	7.1	0	79	26.69	94	8.76	96	9.53	6	1.92
Row8	0	184.5	351.6	215.8	8.7	1	97	31.37	80	29.89	90	9.71	4	2.35
Row9	37	258.6	222	326.4	11.2	0	84	43.96	111	18.87	97	14.69	5	3.02
Row10	0	129.1	228.5	208.8	12.7	4	137	21.95	83	19.42	111	9.4	6	3.43
Row11	0	187.7	163.4	196	9.1	0	127	31.91	148	13.89	94	8.82	5	2.46
Row12	0	128.8	104.9	141.1	11.2	1	96	21.9	71	8.92	128	6.35	2	3.02
Row13	0	156.6	247.6	192.3	12.3	3	88	26.62	75	21.05	115	8.65	5	3.32
Row14	0	120.7	307.2	203	13.1	4	70	20.52	76	26.11	99	9.14	6	3.54
Row15	0	332.9	317.8	160.6	5.4	4	67	56.59	97	27.01	128	7.23	9	1.46
Row16	27	196.4	280.9	89.3	13.8	1	139	33.39	90	23.88	75	4.02	4	3.73
Row17	0	190.7	218.2	129.6	8.1	3	114	32.42	111	18.55	121	5.83	3	2.19
Row18	33	189.7	212.8	165.7	10	1	66	32.25	65	18.09	108	7.46	5	2.7
Row19	0	224.4	159.5	192.8	13	1	90	38.15	88	13.56	74	8.68	2	3.51
Row20	0	155.1	239.7	208.8	10.6	0	117	26.37	93	20.37	133	9.4	4	2.86
Row21	0	62.4	169.9	209.6	5.7	5	89	10.61	121	14.44	64	9.43	6	1.54
Row22	0	183	72.9	181.8	9.5	0	112	31.11	99	6.2	78	8.18	19	2.57
Row23	0	110.4	137.3	189.6	7.7	2	103	18.77	102	11.67	105	8.53	6	2.08
Row24	0	81.1	245.2	237	10.3	0	86	13.79	72	20.84	115	10.67	2	2.78
Row25	0	124.3	277.1	250.7	15.5	3	76	21.13	112	23.55	115	11.28	5	4.19
Row26	39	213	191.1	182.7	9.5	0	115	36.21	112	16.24	115	8.22	3	2.57
Row27	0	134.3	155.5	102.1	14.7	3	73	22.83	100	13.22	68	4.59	4	3.97
Row28	0	190	258.2	181.5	6.3	0	109	32.3	84	21.95	102	8.17	6	1.7
Row29	0	119.3	215.1	178.7	11.1	1	117	20.28	109	18.28	90	8.04	1	3
Row30	0	84.8	136.7	250.5	14.2	2	95	14.42	63	11.62	148	11.27	6	3.83
Row31	0	226.1	201.5	246.2	10.3	1	105	38.44	107	17.13	98	11.08	5	2.78

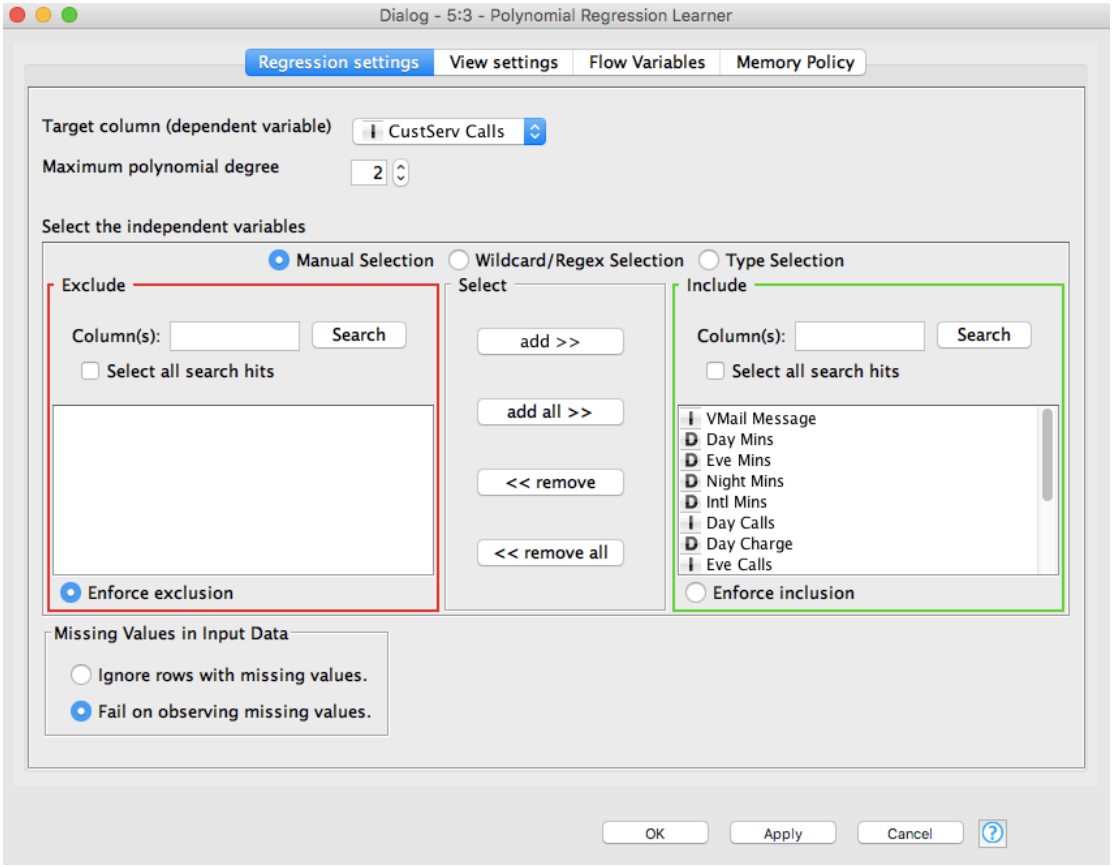
Şekil 9.6.2

Şekil 9.6.2, müşteri yardım masasına gelen aramaları /call center verilerini göstermektedir. Kaç dk aradıkları, gece kaç kez aradığı, gece ödediği, gündüz ödediği vb. bilgiler bulunmaktadır.



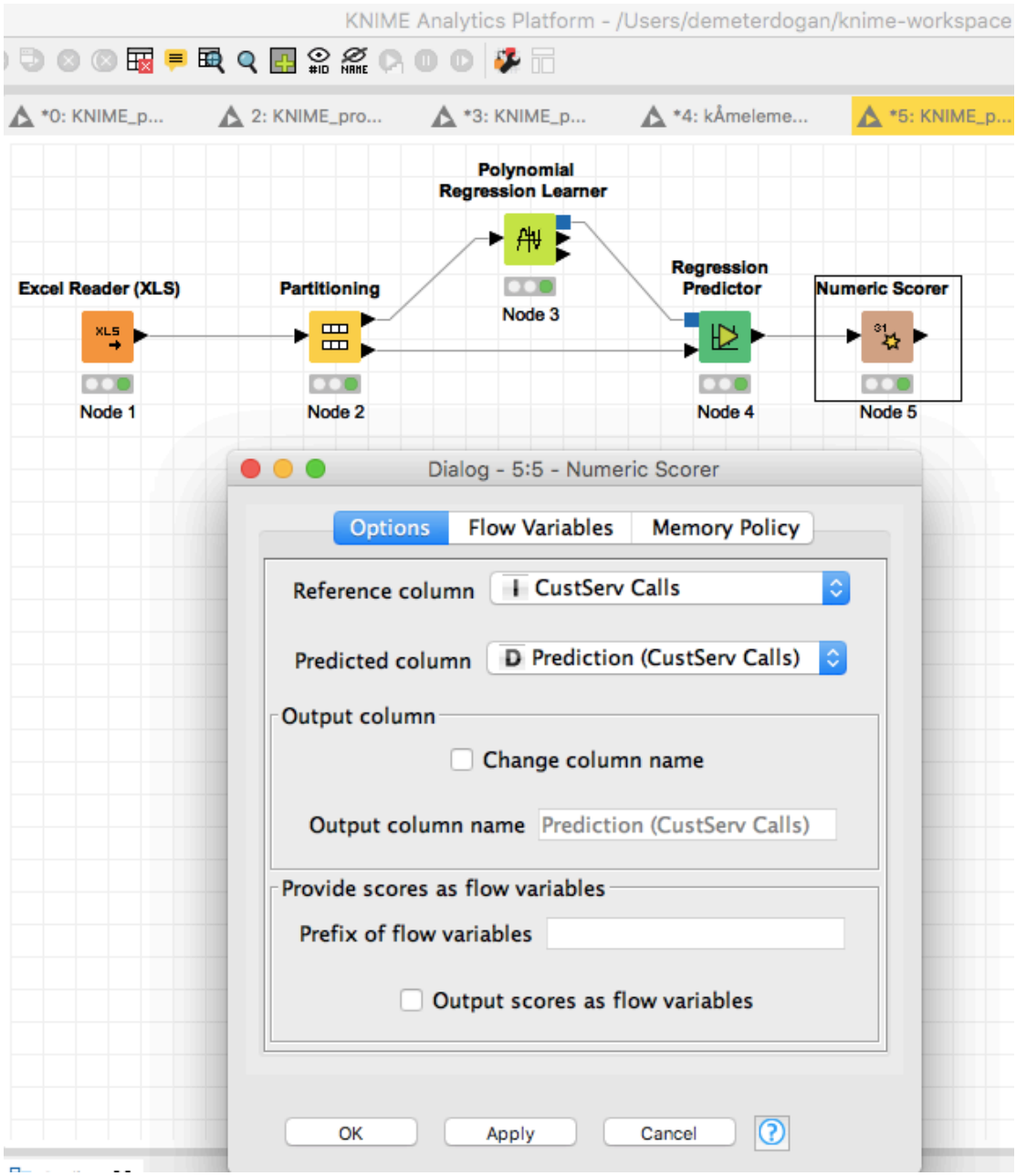
Şekil 9.6.3

Şekil 9.6.3, sisteme partitioning operatörünün eklenmesi ve configure ile veriyi 80% ve 20% oranında bölünmesi için yapılan işlem penceresini göstermektedir.



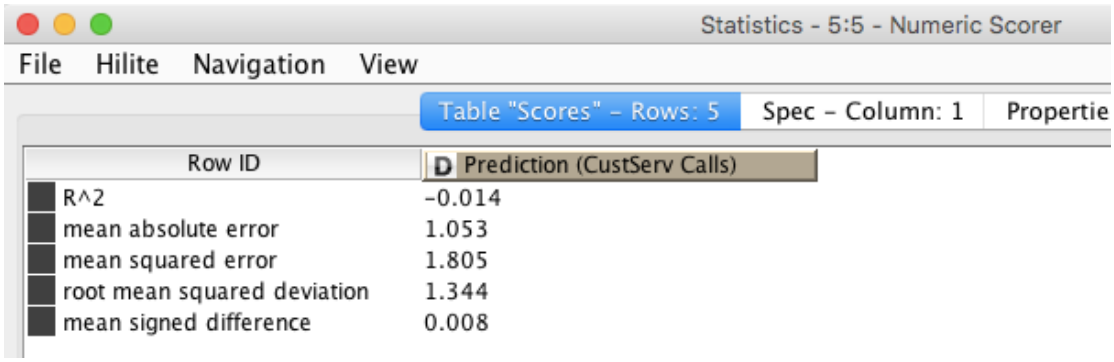
Şekil 9.6.4

Şekil 9.6.4 sisteme eklenen polynomial regression learner için configure penceresinde yapılan değişikliği göstermektedir.



Şekil 9.6.5

Şekil 9.6.5 numeric scorer için configure penceresini göstermektedir. Önceki bölümlerde nominal değerlerin prediction (tahmini) için scorer kullanılırken bu bölümde sayısal değer tahmini yapılacağı için numeric scorer kullanılmalıdır.



Şekil 9.6.6

Şekil 9.6.6, progrsm çalıştırıldıktan sonra statistics penceresini göstermektedir. R^2, mean absolute error, mean squared error, root mean squared deviation ve mean signed difference sonuçları daha önceki bölümlerde açıklanmıştır. Açıklamalara göre bu tablo yorumlanabilmektedir.

9.7 Birliktelik Kural Çıkarımı Değerlendirmesi (ARM Evaluation)

Bu bölümdeki amaç, birliktelik kural çıkarımının nasıl değerlendirileceğini açıklamak. Birliktelik kural çıkarımında iki kavram önemlidir; antecedent ve consequent. antecedent olayı tetikleyen ya da aksiyon öncesi şeklinde açıklanabilir. Consequent ise olay gerçektleştikten sonra ortaya çıkan durumlar şeklinde açıklanabilir.

Örnek olarak süt ve bebek bezi almış kişilerin bira alma oranı incelenecektir.

Değerlendirme Ölçütleri (Evaluation Metrics)

Destek (Support (s)) → Tüm işlemlerde X ve Y'nin birlikte geçme oranı. Örneğin müşterilerin kaçı süt, bebekbezi ve birayı birlikte almıştır?

$$s = \frac{\sigma(süt, bebek\ bezi, bira)}{|T|} = \frac{2}{5} = 0.4$$

Güven (Confidence (c)) → İlk aksiyon Y'yi yapanların kaçı X'i de yaptı? Örneğin, süt ve bebekbezi alanların kaçı birayı da almıştır?

$$c = \frac{\sigma(süt, bebek\ bezi, bira)}{\sigma(süt, bebek\ bezi)} = \frac{2}{3} = 0.67$$

Kaldıraç (Lift (l)) → İki aksiyon Y ve X'in bağımsız olması durumunda ne kadar birlikte tekrarlandığını gösterir. Örneğin, süt ve bebekbezi alanların ne kadarı birayı da birlikte almıştır?

$$I = \frac{s(X,Y)}{s(X), s(Y)} = \frac{0.4}{\frac{3}{5} x \frac{3}{5}} = 1.1111$$

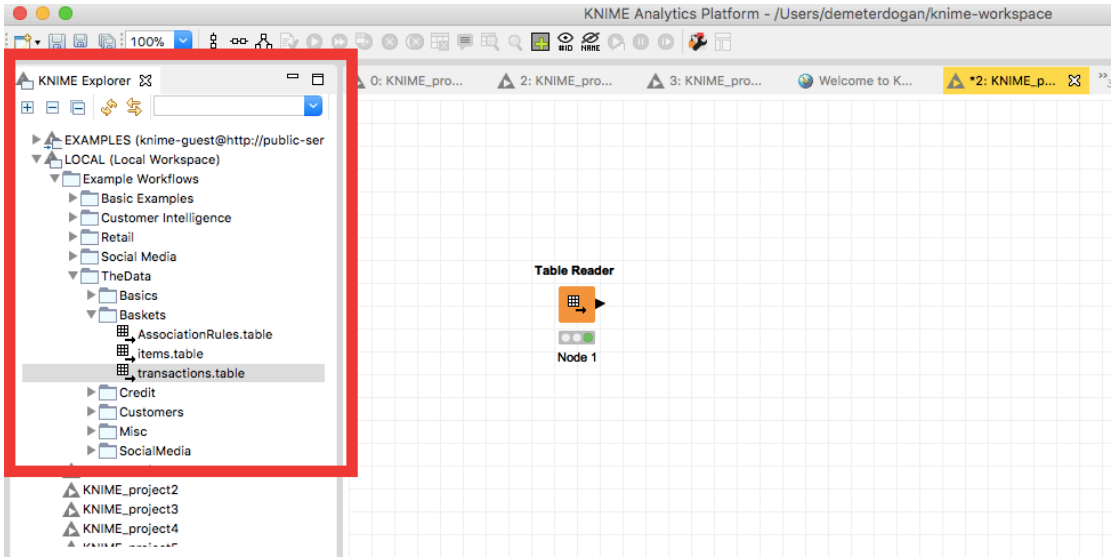
T-numarası	Durum
1	Ekmek, süt
2	Ekmek, bebek bezi, bira, yumurta
3	Süt, bebek bezi, kola
4	Ekmek, süt, bebek bezi, bira
5	Ekmek, süt, bebek bezi, kola

Tablo 9.7.1

Tablo 9.7.1 bir markette örnek 5 satış işleminin kaydını göstermektedir.

9.8 Birliktelik Kural Çıkarımı Değerlendirmesi (ARM Evaluation)

Bu bölümde amaç, birliktelik kural çıkarımının Knime üzerinden uygulaması göstermek.

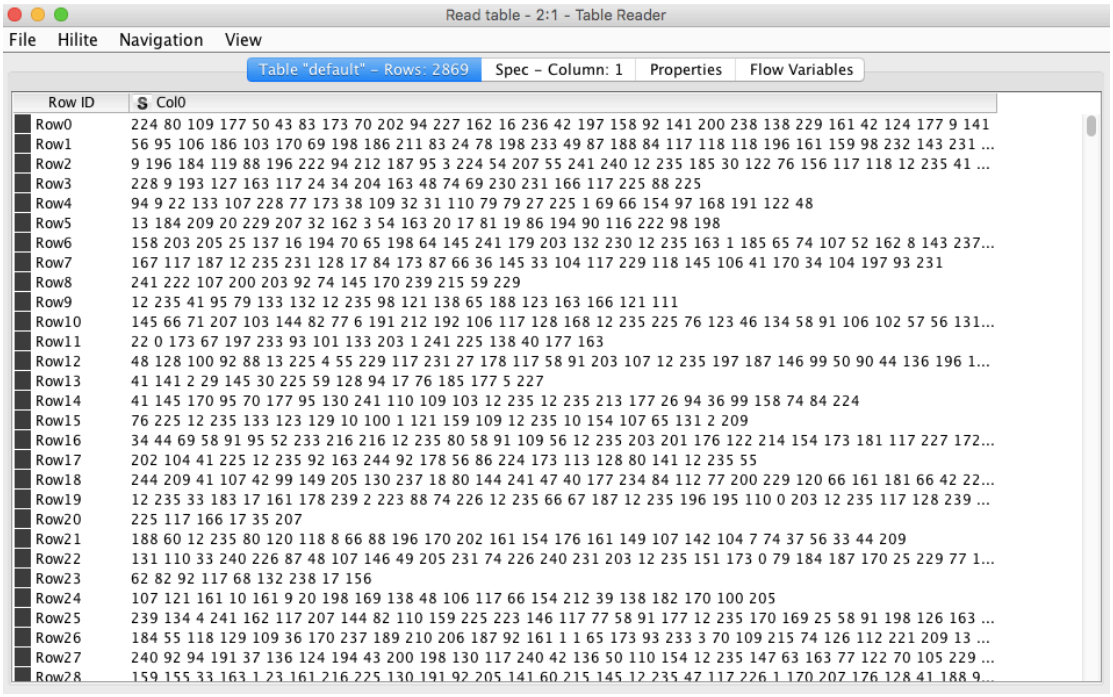


Şekil 9.8.1

Şekil 9.8.1, bu bölümde kullanılacak örnek veri setinin sisteme nereden aktarılacağını ve configure'ünü göstermektedir.

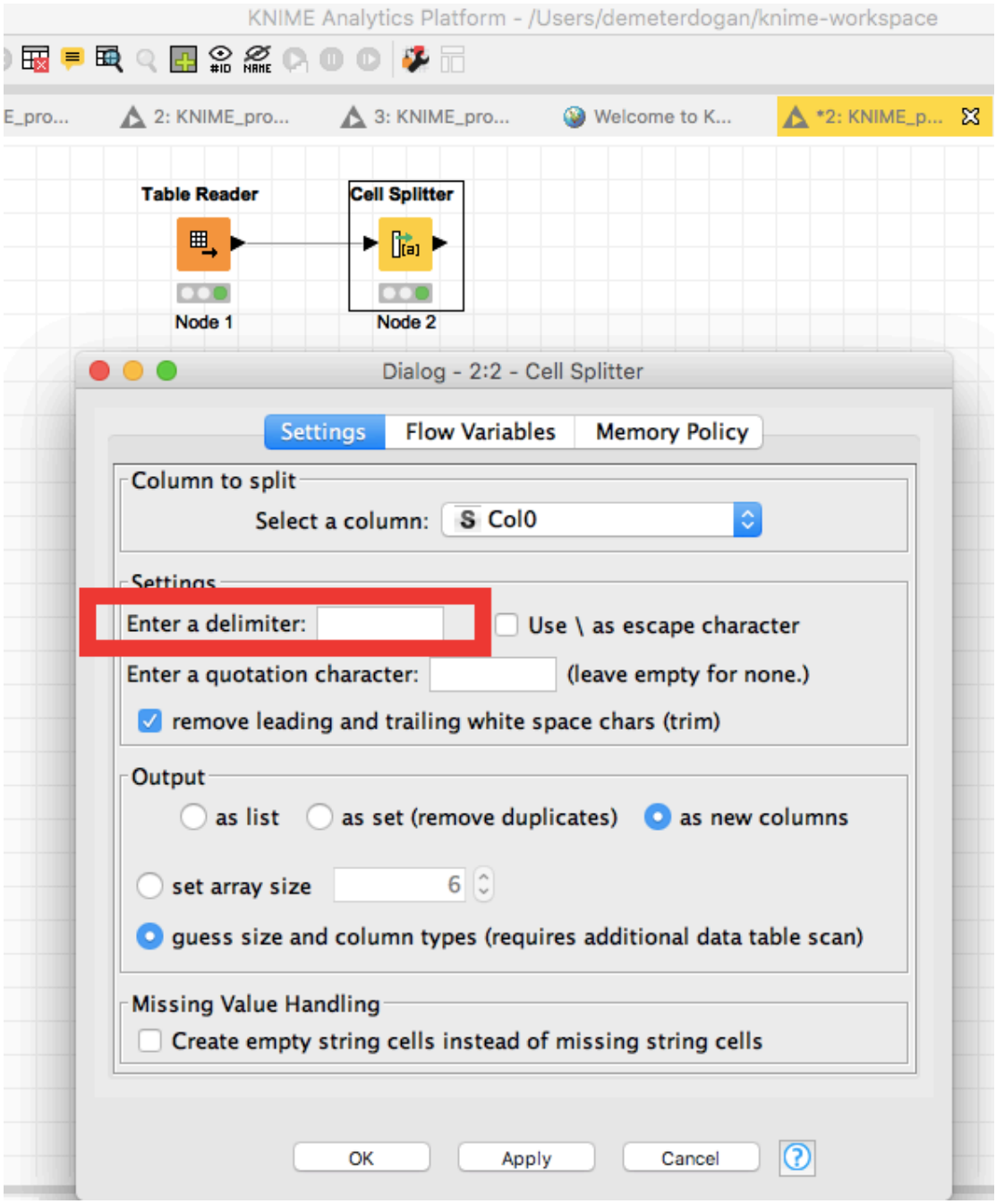
Knime explorer→ Local→ example workflows → theData→ Baskets→ transactions

Adımları sırasıyla takip ederek transactions ismindeki excel formatındaki veri setine çift tıklandığında veri seti sisteme direk aktarılmış olur. Daha sonra configure seçeneğinde ilk row başlık olabilmesi için seçim yapılır.



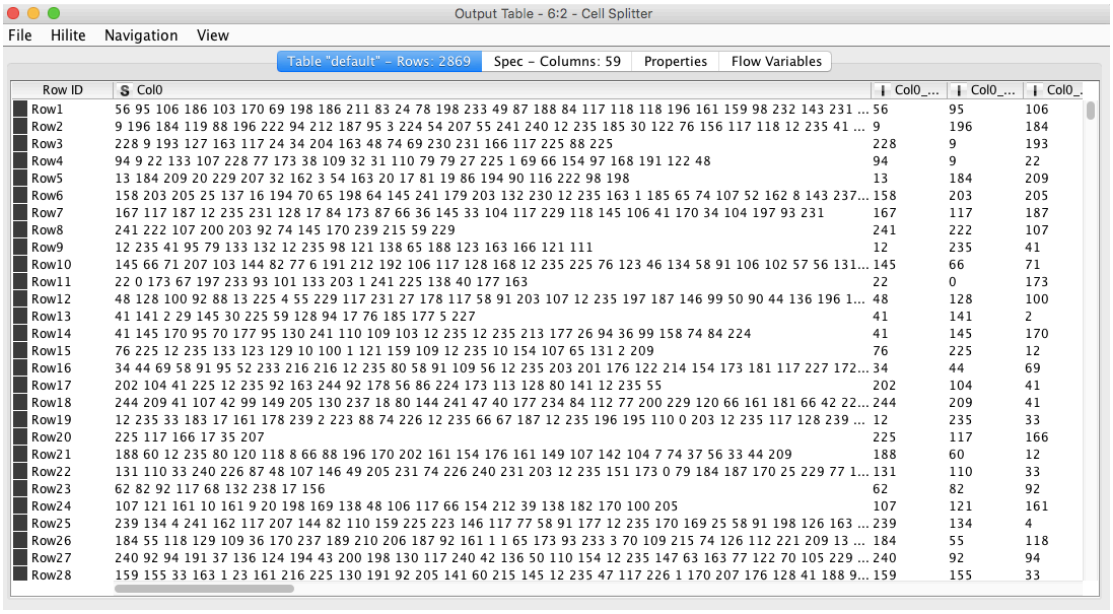
Şekil 9.8.2

Şekil 9.8.2, birlikte satılan ürün kodlarını göstermektedir. Ayıraç olarak boşluk kullanılmıştır. Bunları ayırmak için cell splitter operatörü kullanılacaktır.



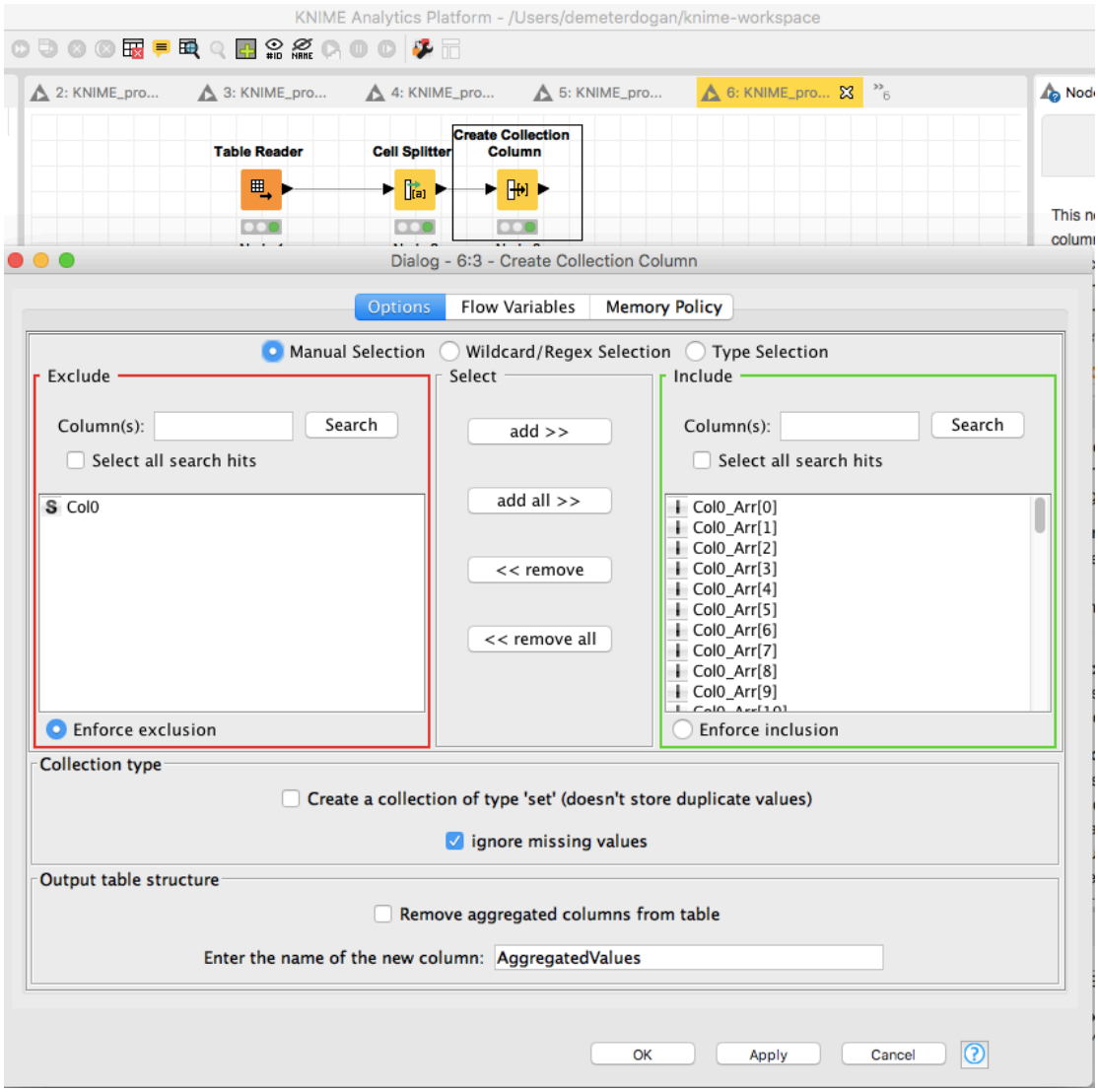
Şekil 9.8.3

Şekil 9.8.3, sisteme cell splitter eklenmesini ve configure penceresini göstermektedir. Enter a delimineter bölümünde girilen karaktere göre kolondaki hücrelerin ayrılması demektir.



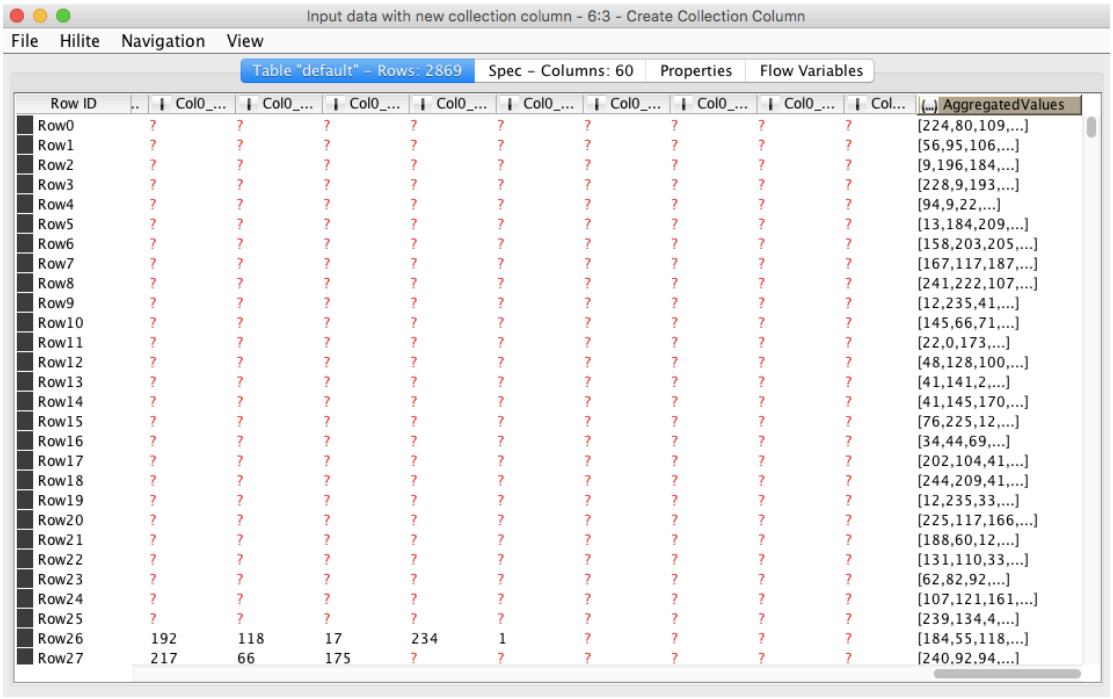
Şekil 9.8.4

Şekil 9.8.4 Kolonlardaki rakamların aralarındaki boşluk'lardan ayrılarak hepsinin ayrı bir kolonda toplanmasını göstermektedir. Daha sonra sisteme create collection column operatörü eklenecektir.



Şekil 9.8.5

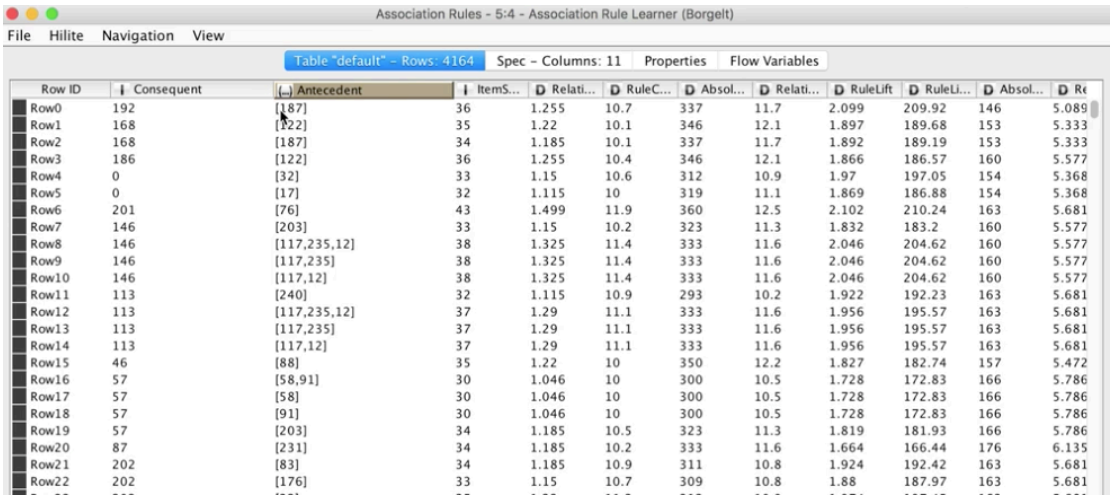
Şekil 9.8.5 sisteme create collection column eklenmesini ve onun configure'ünü göstermektedir. Burada amaç kolonları liste şeklinde birleştirmektir.



Row ID	Col0_...	Col0_...	Col0_...	Col0_...	Col0_...	Col0_...	Col0_...	Col0_...	Col...	AggregatedValues
Row0	?	?	?	?	?	?	?	?	?	[224,80,109,...]
Row1	?	?	?	?	?	?	?	?	?	[56,95,106,...]
Row2	?	?	?	?	?	?	?	?	?	[9,196,184,...]
Row3	?	?	?	?	?	?	?	?	?	[228,9,193,...]
Row4	?	?	?	?	?	?	?	?	?	[94,9,22,...]
Row5	?	?	?	?	?	?	?	?	?	[13,184,209,...]
Row6	?	?	?	?	?	?	?	?	?	[158,203,205,...]
Row7	?	?	?	?	?	?	?	?	?	[167,117,187,...]
Row8	?	?	?	?	?	?	?	?	?	[241,222,107,...]
Row9	?	?	?	?	?	?	?	?	?	[12,235,41,...]
Row10	?	?	?	?	?	?	?	?	?	[145,66,71,...]
Row11	?	?	?	?	?	?	?	?	?	[22,0,173,...]
Row12	?	?	?	?	?	?	?	?	?	[48,128,100,...]
Row13	?	?	?	?	?	?	?	?	?	[41,141,2,...]
Row14	?	?	?	?	?	?	?	?	?	[41,145,170,...]
Row15	?	?	?	?	?	?	?	?	?	[76,225,12,...]
Row16	?	?	?	?	?	?	?	?	?	[34,44,69,...]
Row17	?	?	?	?	?	?	?	?	?	[202,104,41,...]
Row18	?	?	?	?	?	?	?	?	?	[244,209,41,...]
Row19	?	?	?	?	?	?	?	?	?	[12,235,33,...]
Row20	?	?	?	?	?	?	?	?	?	[225,117,166,...]
Row21	?	?	?	?	?	?	?	?	?	[188,60,12,...]
Row22	?	?	?	?	?	?	?	?	?	[131,110,33,...]
Row23	?	?	?	?	?	?	?	?	?	[62,82,92,...]
Row24	?	?	?	?	?	?	?	?	?	[107,121,161,...]
Row25	?	?	?	?	?	?	?	?	?	[239,134,4,...]
Row26	192	118	17	234	1	?	?	?	?	[184,55,118,...]
Row27	217	66	175	?	?	?	?	?	?	[240,92,94,...]

Şekil 9.8.6

Şekil 9.8.6 program çalıştırıldıktan sonraki sonucu göstermektedir. En sağdaki kolonda diğer kolonların collection hali görülmektedir.



Row ID	Consequent	Antecedent	ItemS...	Relati...	RuleC...	Absol...	Relati...	RuleLift	RuleLi...	Absol...	Re
Row0	192	[187]	36	1.255	10.7	337	11.7	2.099	209.92	146	5.089
Row1	168	[122]	35	1.22	10.1	346	12.1	1.897	189.68	153	5.333
Row2	168	[187]	34	1.185	10.1	337	11.7	1.892	189.19	153	5.333
Row3	186	[122]	36	1.255	10.4	346	12.1	1.866	186.57	160	5.577
Row4	0	[32]	33	1.15	10.6	312	10.9	1.97	197.05	154	5.368
Row5	0	[17]	32	1.115	10	319	11.1	1.869	186.88	154	5.368
Row6	201	[76]	43	1.499	11.9	360	12.5	2.102	210.24	163	5.681
Row7	146	[203]	33	1.15	10.2	323	11.3	1.832	183.2	160	5.577
Row8	146	[117,235,12]	38	1.325	11.4	333	11.6	2.046	204.62	160	5.577
Row9	146	[117,235]	38	1.325	11.4	333	11.6	2.046	204.62	160	5.577
Row10	146	[117,12]	38	1.325	11.4	333	11.6	2.046	204.62	160	5.577
Row11	113	[240]	32	1.115	10.9	293	10.2	1.922	192.23	163	5.681
Row12	113	[117,235,12]	37	1.29	11.1	333	11.6	1.956	195.57	163	5.681
Row13	113	[117,235]	37	1.29	11.1	333	11.6	1.956	195.57	163	5.681
Row14	113	[117,12]	37	1.29	11.1	333	11.6	1.956	195.57	163	5.681
Row15	46	[88]	35	1.22	10	350	12.2	1.827	182.74	157	5.472
Row16	57	[58,91]	30	1.046	10	300	10.5	1.728	172.83	166	5.786
Row17	57	[58]	30	1.046	10	300	10.5	1.728	172.83	166	5.786
Row18	57	[91]	30	1.046	10	300	10.5	1.728	172.83	166	5.786
Row19	57	[203]	34	1.185	10.5	323	11.3	1.819	181.93	166	5.786
Row20	87	[231]	34	1.185	10.2	333	11.6	1.664	166.44	176	6.135
Row21	202	[83]	34	1.185	10.9	311	10.8	1.924	192.42	163	5.681
Row22	202	[176]	33	1.15	10.7	309	10.8	1.88	187.97	163	5.681

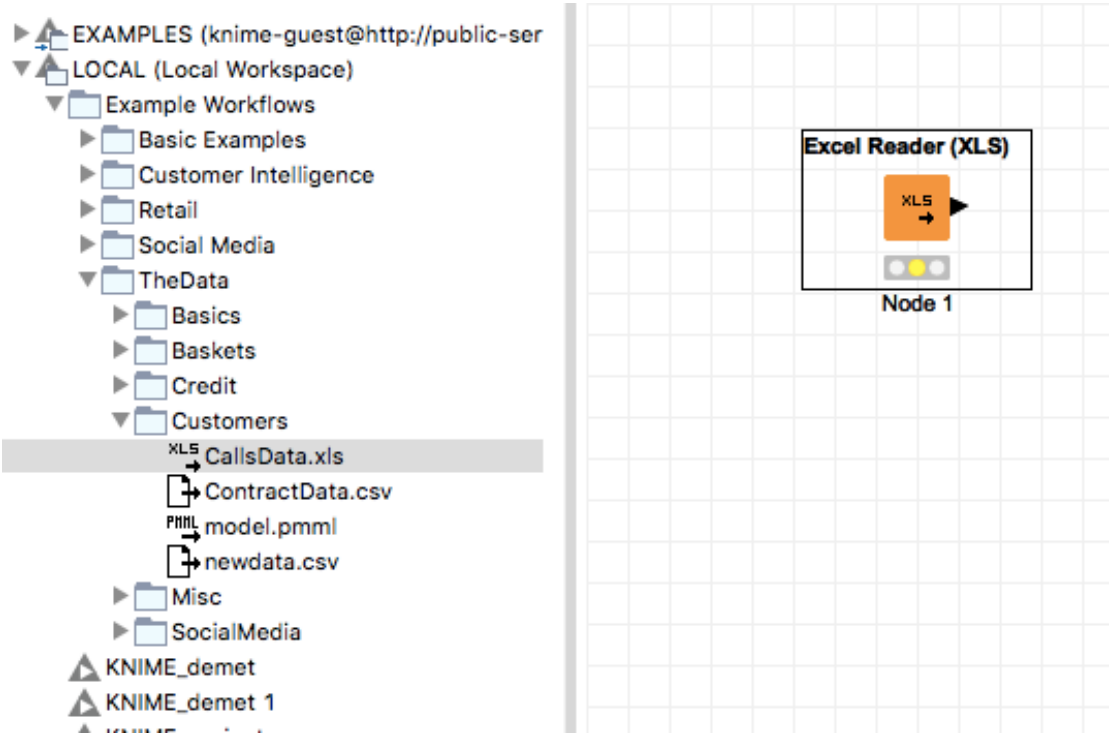
Şekil 9.8.7

Şekil 9.8.7 sisteme association rule learner (borgelt) eklendikten sonra programın çalıştırılıp sonuç elde edilmiş halini göstermektedir. Antecedent kolonu olayın gerçekleşmesi durumu consuquent ise olay gerçekleşmesinden sonra gerçekleşecek durumu göstermektedir. Örneğin 187 nolu ürünü alan kişinin 192 nolu ürünü alması beklenir şeklinde yorumlanabilir.

10. KNIME İLE DİĞER DİLLERİN BAĞLANMASI

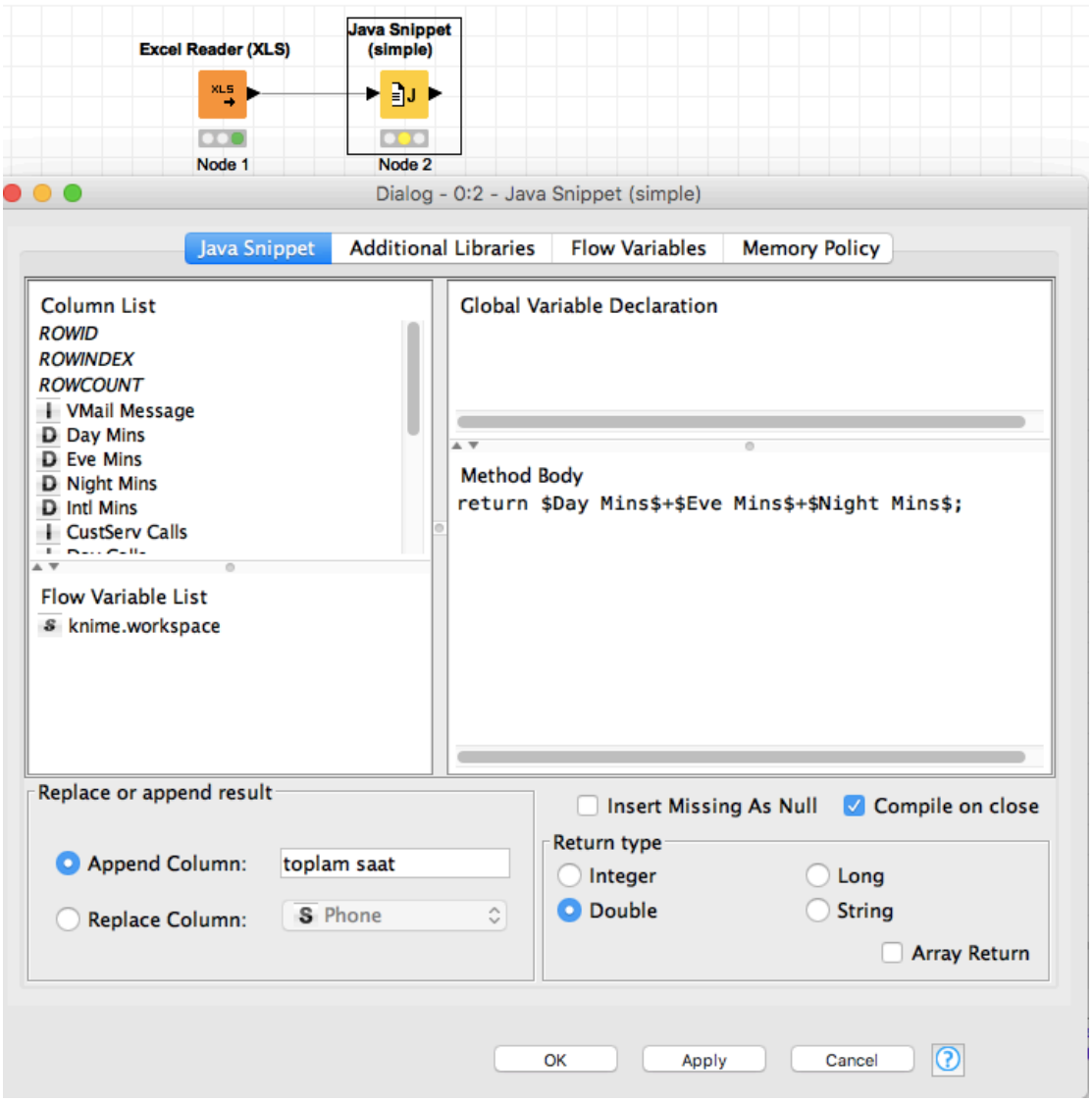
10.1. Java Snippet

Bu bölümde Java kodlarını Knime ile nasıl birleştirilebileceği gösterilecektir.



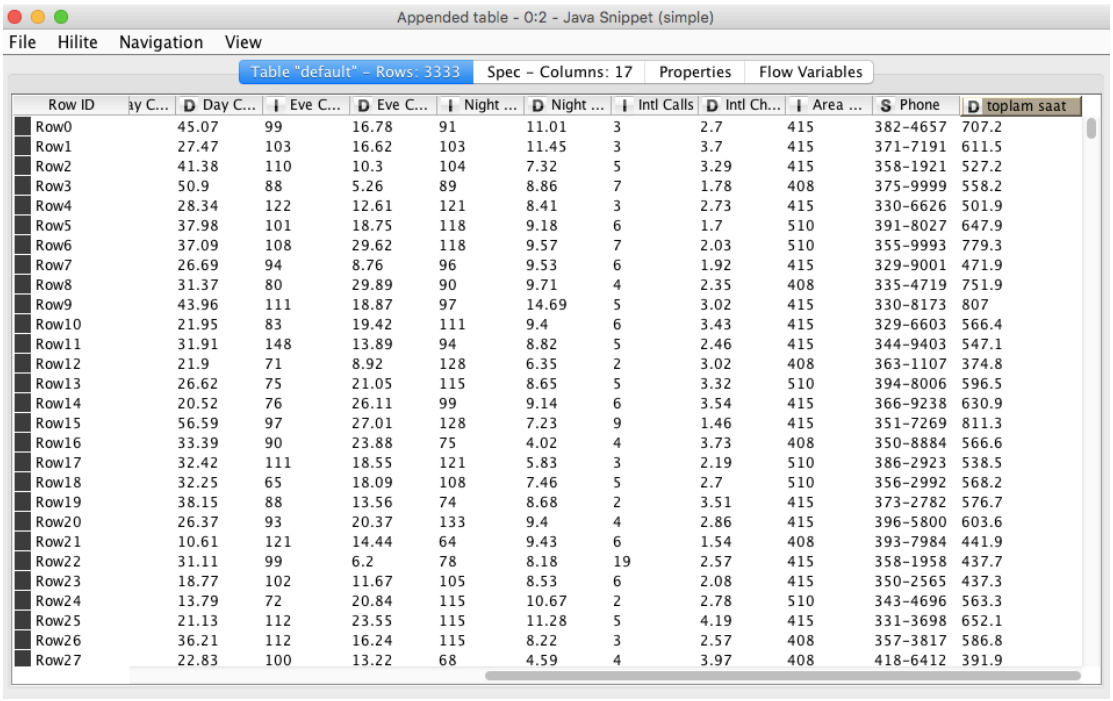
Şekil 10.1.1

Şekil 10.1.1 Knime içerisinde olan CallsData veri setinin sisteme aktarılmasını göstermektedir. CallsData seçilip sürüklenerek Knime işlem penceresine bırakılarak sisteme aktarılmış olunur. Configure bölümünden ilk row'larının "table contains column names in row number" ile başlık olarak seçilmesi gerekmektedir.



Şekil 10.1.2

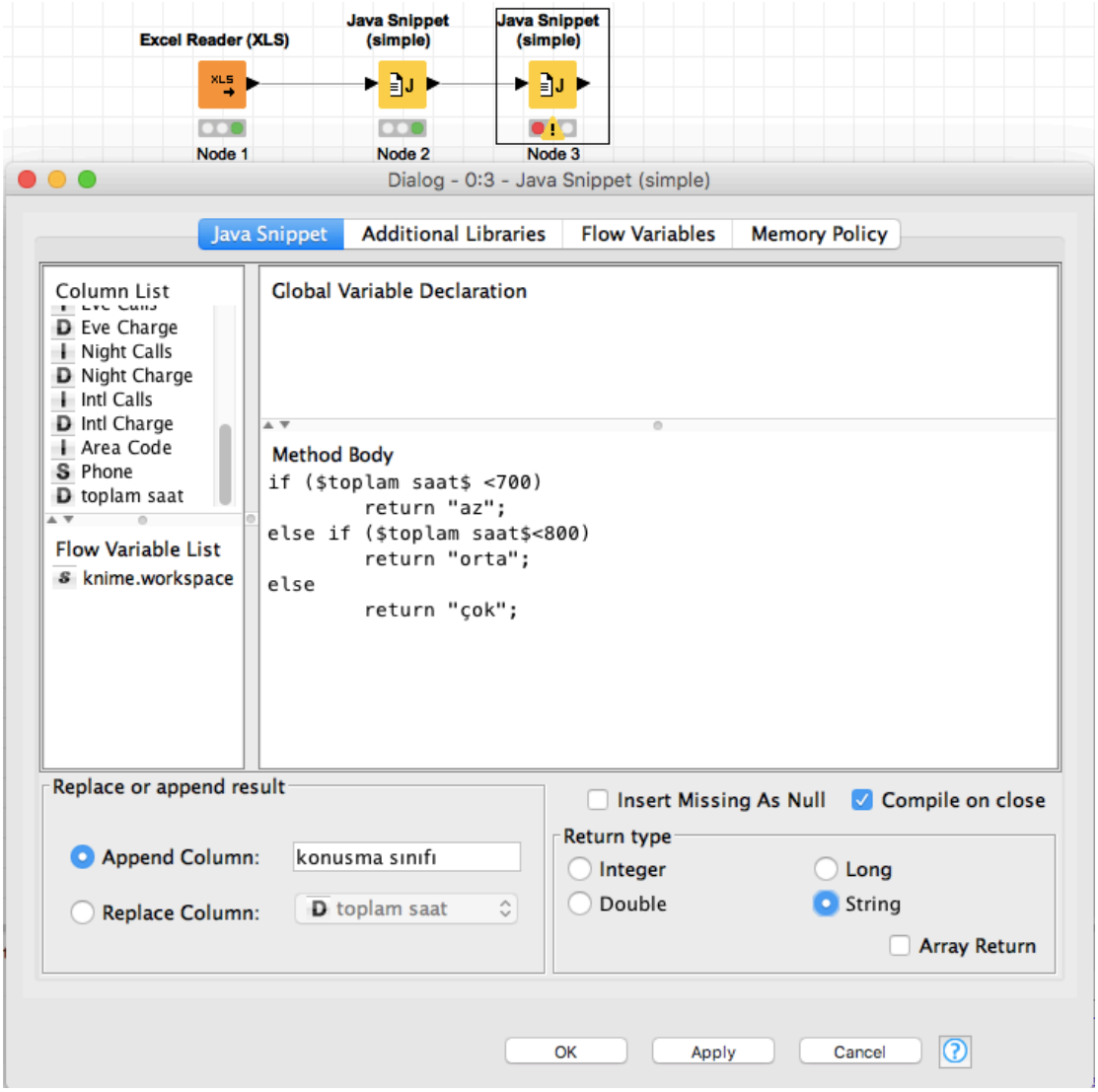
Şekil 10.1.2 sisteme Java snippet (simple) operatörünün eklenmesini ve configure bölümünü göstermektedir. Sol taraftaki pencereden day mins, eve mins ve night mins yazılarının üzerine çift tıklandığında onları sağdaki penceredeki method body penceresine aktarır. Burası kodun yazıldığı kısımdır. Başlarına return ve sonuna ; işareti konulur. Append column ise yeni oluşacak kolonun ismini göstermektedir. Oraya bu örnek için toplam saat yazılmıştır. Program bu şekilde çalıştırılır.



Row ID	ay C...	Day C...	Eve C...	Eve C...	Night ...	Night ...	Intl Calls	Intl Ch...	Area ...	Phone	toplam saat
Row0		45.07	99	16.78	91	11.01	3	2.7	415	382-4657	707.2
Row1		27.47	103	16.62	103	11.45	3	3.7	415	371-7191	611.5
Row2		41.38	110	10.3	104	7.32	5	3.29	415	358-1921	527.2
Row3		50.9	88	5.26	89	8.86	7	1.78	408	375-9999	558.2
Row4		28.34	122	12.61	121	8.41	3	2.73	415	330-6626	501.9
Row5		37.98	101	18.75	118	9.18	6	1.7	510	391-8027	647.9
Row6		37.09	108	29.62	118	9.57	7	2.03	510	355-9993	779.3
Row7		26.69	94	8.76	96	9.53	6	1.92	415	329-9001	471.9
Row8		31.37	80	29.89	90	9.71	4	2.35	408	335-4719	751.9
Row9		43.96	111	18.87	97	14.69	5	3.02	415	330-8173	807
Row10		21.95	83	19.42	111	9.4	6	3.43	415	329-6603	566.4
Row11		31.91	148	13.89	94	8.82	5	2.46	415	344-9403	547.1
Row12		21.9	71	8.92	128	6.35	2	3.02	408	363-1107	374.8
Row13		26.62	75	21.05	115	8.65	5	3.32	510	394-8006	596.5
Row14		20.52	76	26.11	99	9.14	6	3.54	415	366-9238	630.9
Row15		56.59	97	27.01	128	7.23	9	1.46	415	351-7269	811.3
Row16		33.39	90	23.88	75	4.02	4	3.73	408	350-8884	566.6
Row17		32.42	111	18.55	121	5.83	3	2.19	510	386-2923	538.5
Row18		32.25	65	18.09	108	7.46	5	2.7	510	356-2992	568.2
Row19		38.15	88	13.56	74	8.68	2	3.51	415	373-2782	576.7
Row20		26.37	93	20.37	133	9.4	4	2.86	415	396-5800	603.6
Row21		10.61	121	14.44	64	9.43	6	1.54	408	393-7984	441.9
Row22		31.11	99	6.2	78	8.18	19	2.57	415	358-1958	437.7
Row23		18.77	102	11.67	105	8.53	6	2.08	415	350-2565	437.3
Row24		13.79	72	20.84	115	10.67	2	2.78	510	343-4696	563.3
Row25		21.13	112	23.55	115	11.28	5	4.19	415	331-3698	652.1
Row26		36.21	112	16.24	115	8.22	3	2.57	408	357-3817	586.8
Row27		22.83	100	13.22	68	4.59	4	3.97	408	418-6412	391.9

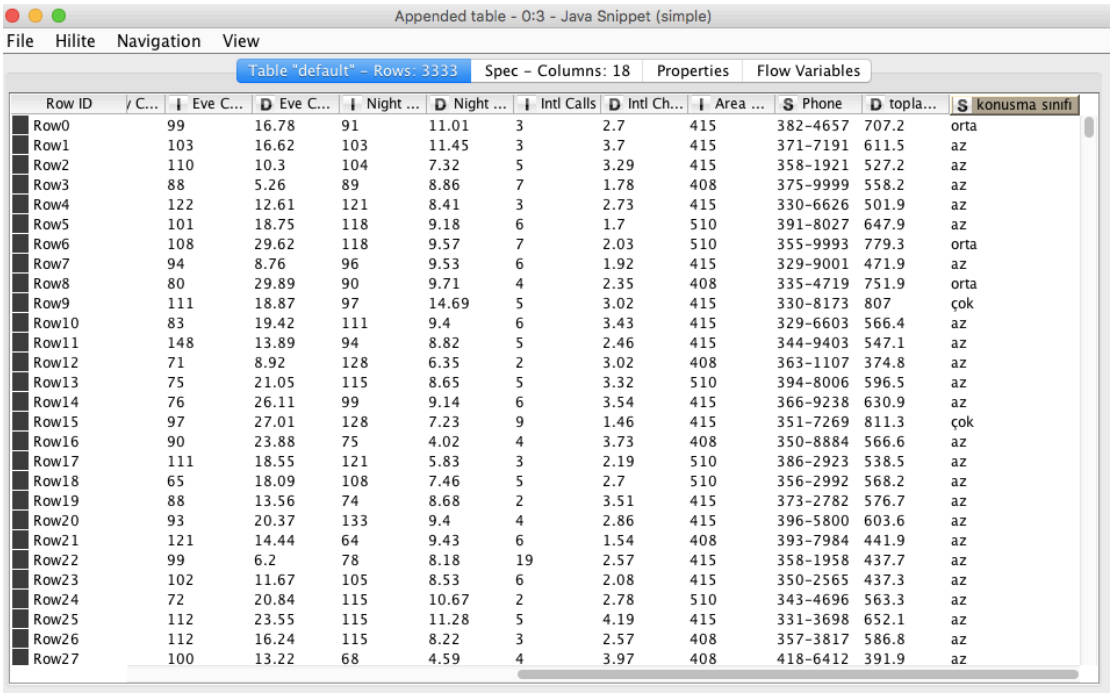
Şekil 10.1.3

Şekil 10.1.3 program çalıştırıldıktan sonra oluşan sonuç ekranını göstermektedir. En sağdaki kolon yeni oluşturulan “toplam saat” kolonudur.



Şekil 10.1.4

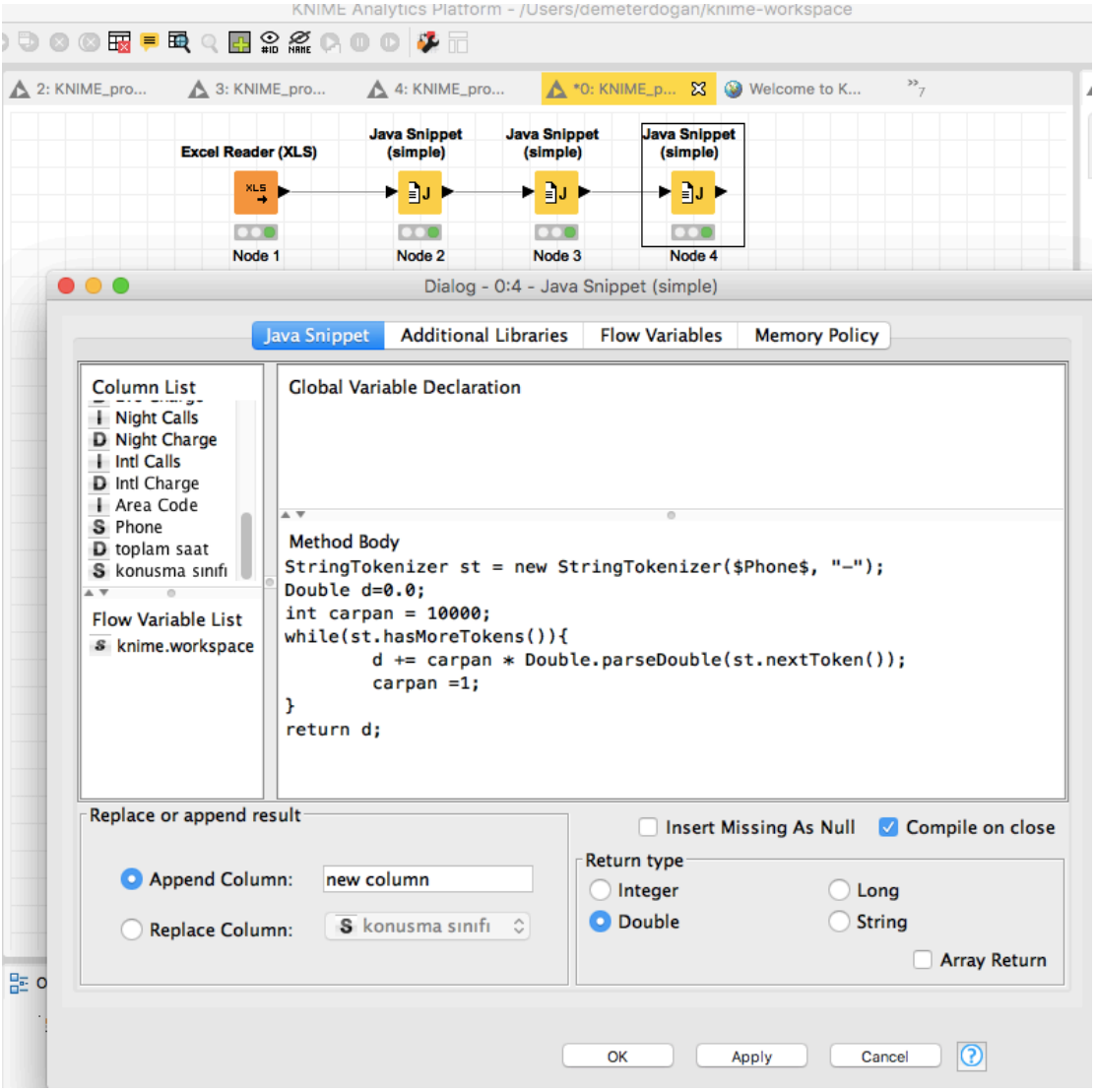
Şekil 10.1.4 sisteme ikinci bir java snippet eklenmesini ve yeni bir kod yazılmasını göstermektedir. Bir önceki işlemde oluşturulan toplam saat kolonuna göre görüşme yapan müşterileri sınıflandırmak için bu işlem yapılmıştır. Kodlara göre; 700 saatten az konuşan az, 800'dan az 700'den fazla için orta ve 800 üzerinde olanlar için çok yazan konuşma sınıfı isimli bir kolon oluşturulacaktır.



Row ID	/ C...	Eve C...	Eve C...	Night ...	Night ...	Intl Calls	Intl Ch...	Area ...	Phone	topla...	konusma sınıfı
Row0	99	16.78	91	11.01	3	2.7	415	382-4657	707.2	orta
Row1	103	16.62	103	11.45	3	3.7	415	371-7191	611.5	az
Row2	110	10.3	104	7.32	5	3.29	415	358-1921	527.2	az
Row3	88	5.26	89	8.86	7	1.78	408	375-9999	558.2	az
Row4	122	12.61	121	8.41	3	2.73	415	330-6626	501.9	az
Row5	101	18.75	118	9.18	6	1.7	510	391-8027	647.9	az
Row6	108	29.62	118	9.57	7	2.03	510	355-9993	779.3	orta
Row7	94	8.76	96	9.53	6	1.92	415	329-9001	471.9	az
Row8	80	29.89	90	9.71	4	2.35	408	335-4719	751.9	orta
Row9	111	18.87	97	14.69	5	3.02	415	330-8173	807	çok
Row10	83	19.42	111	9.4	6	3.43	415	329-6603	566.4	az
Row11	148	13.89	94	8.82	5	2.46	415	344-9403	547.1	az
Row12	71	8.92	128	6.35	2	3.02	408	363-1107	374.8	az
Row13	75	21.05	115	8.65	5	3.32	510	394-8006	596.5	az
Row14	76	26.11	99	9.14	6	3.54	415	366-9238	630.9	az
Row15	97	27.01	128	7.23	9	1.46	415	351-7269	811.3	çok
Row16	90	23.88	75	4.02	4	3.73	408	350-8884	566.6	az
Row17	111	18.55	121	5.83	3	2.19	510	386-2923	538.5	az
Row18	65	18.09	108	7.46	5	2.7	510	356-2992	568.2	az
Row19	88	13.56	74	8.68	2	3.51	415	373-2782	576.7	az
Row20	93	20.37	133	9.4	4	2.86	415	396-5800	603.6	az
Row21	121	14.44	64	9.43	6	1.54	408	393-7984	441.9	az
Row22	99	6.2	78	8.18	19	2.57	415	358-1958	437.7	az
Row23	102	11.67	105	8.53	6	2.08	415	350-2565	437.3	az
Row24	72	20.84	115	10.67	2	2.78	510	343-4696	563.3	az
Row25	112	23.55	115	11.28	5	4.19	415	331-3698	652.1	az
Row26	112	16.24	115	8.22	3	2.57	408	357-3817	586.8	az
Row27	100	13.22	68	4.59	4	3.97	408	418-6412	391.9	az

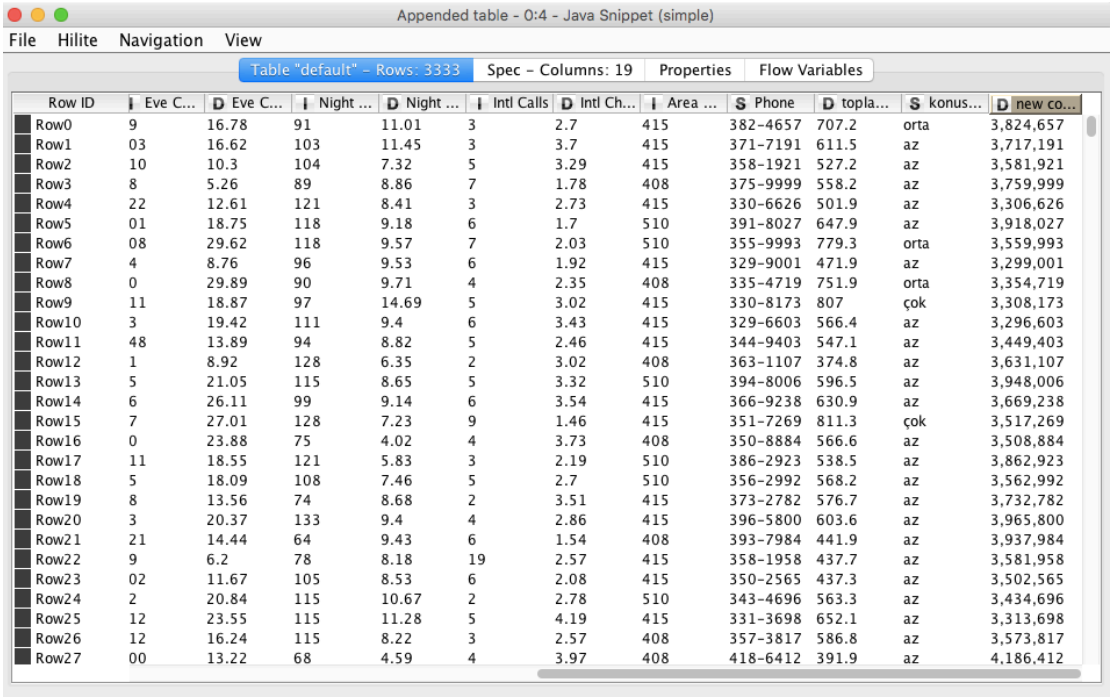
Şekil 10.1.5

Şekil 10.1.5 program çalıştırıldıktan sonra toplam saat kolonuna göre oluşan konuşma sınıfı kolonunu göstermektedir.



10.1.6

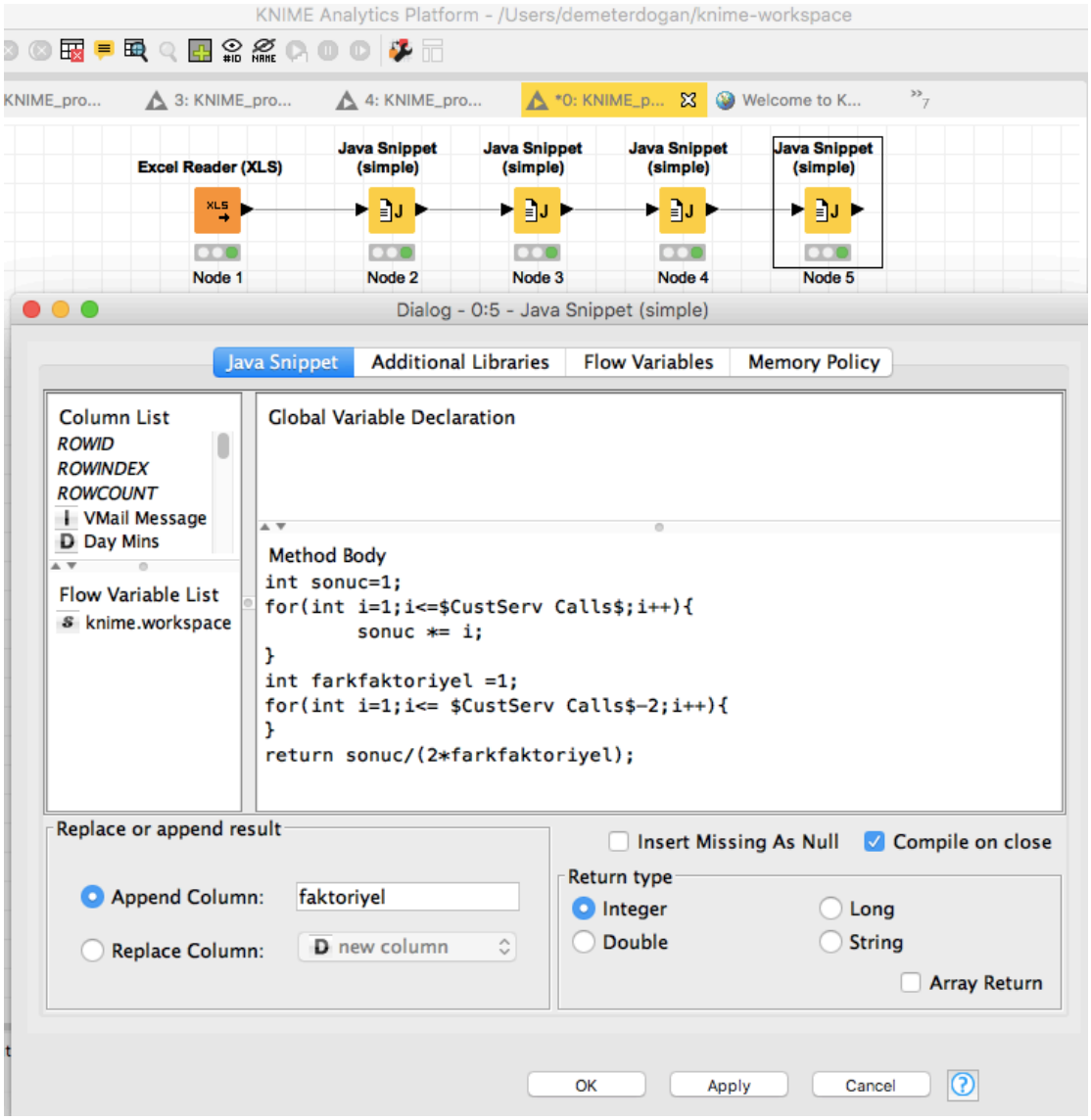
Şekil.10.1.6 phone (telefon) numaralarının olduğu kolondaki numaraların parçalanması için sisteme yeni bir Java snippet eklenmesini ve configure bölümünde yazılan kodu göstermektedir. Telefon numarası xxx-xxxx formatında olduğu için ilk üç haneyi 10000 ile çarpıp, sonra aradaki işareti (-) kaldırıp tel bir sayı formunda yazabilmenin örnek kodu gösterilmiştir.



Row ID	Eve C...	Eve C...	Night ...	Night ...	Intl Calls	Intl Ch...	Area ...	Phone	topla...	konus...	new co...
Row0	9	16.78	91	11.01	3	2.7	415	382-4657	707.2	orta	3,824,657
Row1	03	16.62	103	11.45	3	3.7	415	371-7191	611.5	az	3,717,191
Row2	10	10.3	104	7.32	5	3.29	415	358-1921	527.2	az	3,581,921
Row3	8	5.26	89	8.86	7	1.78	408	375-9999	558.2	az	3,759,999
Row4	22	12.61	121	8.41	3	2.73	415	330-6626	501.9	az	3,306,626
Row5	01	18.75	118	9.18	6	1.7	510	391-8027	647.9	az	3,918,027
Row6	08	29.62	118	9.57	7	2.03	510	355-9993	779.3	orta	3,559,993
Row7	4	8.76	96	9.53	6	1.92	415	329-9001	471.9	az	3,299,001
Row8	0	29.89	90	9.71	4	2.35	408	335-4719	751.9	orta	3,354,719
Row9	11	18.87	97	14.69	5	3.02	415	330-8173	807	çok	3,308,173
Row10	3	19.42	111	9.4	6	3.43	415	329-6603	566.4	az	3,296,603
Row11	48	13.89	94	8.82	5	2.46	415	344-9403	547.1	az	3,449,403
Row12	1	8.92	128	6.35	2	3.02	408	363-1107	374.8	az	3,631,107
Row13	5	21.05	115	8.65	5	3.32	510	394-8006	596.5	az	3,948,006
Row14	6	26.11	99	9.14	6	3.54	415	366-9238	630.9	az	3,669,238
Row15	7	27.01	128	7.23	9	1.46	415	351-7269	811.3	çok	3,517,269
Row16	0	23.88	75	4.02	4	3.73	408	350-8884	566.6	az	3,508,884
Row17	11	18.55	121	5.83	3	2.19	510	386-2923	538.5	az	3,862,923
Row18	5	18.09	108	7.46	5	2.7	510	356-2992	568.2	az	3,562,992
Row19	8	13.56	74	8.68	2	3.51	415	373-2782	576.7	az	3,732,782
Row20	3	20.37	133	9.4	4	2.86	415	396-5800	603.6	az	3,965,800
Row21	21	14.44	64	9.43	6	1.54	408	393-7984	441.9	az	3,937,984
Row22	9	6.2	78	8.18	19	2.57	415	358-1958	437.7	az	3,581,958
Row23	02	11.67	105	8.53	6	2.08	415	350-2565	437.3	az	3,502,565
Row24	2	20.84	115	10.67	2	2.78	510	343-4696	563.3	az	3,434,696
Row25	12	23.55	115	11.28	5	4.19	415	331-3698	652.1	az	3,313,698
Row26	12	16.24	115	8.22	3	2.57	408	357-3817	586.8	az	3,573,817
Row27	00	13.22	68	4.59	4	3.97	408	418-6412	391.9	az	4,186,412

Şekil 10.1.7

Şekil 10.1.7 program çalıştırıldıktan sonraki sonuç sayfasını göstermektedir. En sağdaki new column isimli kolan phone kolonunda yazan numaraların formatını parçalamayı ve Double formunda sayı olarak yazılmışını göstermektedir.



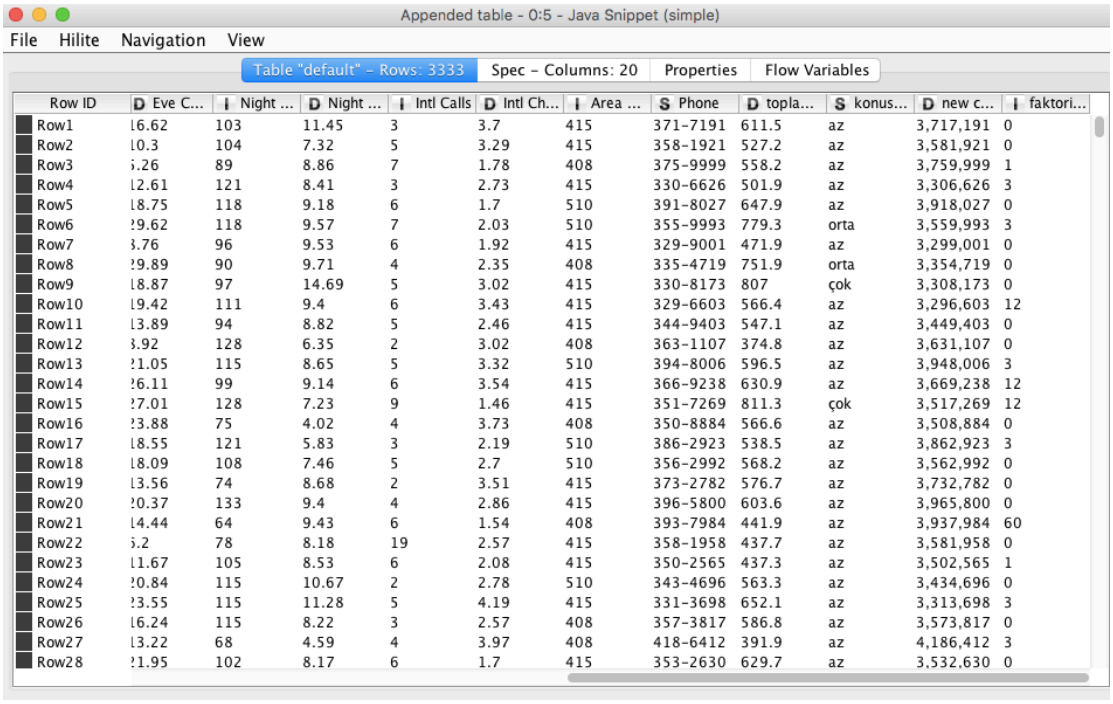
Şekil 10.1.8

Şekil 10.1.8'de kombinasyon kodu yazılmıştır.

Cust serv call kolonunda yazan kolonu ;

$$C(n,r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

formülü ile yazılmıştır.



Row ID	Eve C...	Night ...	Night ...	Intl Calls	Intl Ch...	Area ...	Phone	topla...	konus...	new c...	faktori...
Row1	16.62	103	11.45	3	3.7	415	371-7191	611.5	az	3,717,191	0
Row2	10.3	104	7.32	5	3.29	415	358-1921	527.2	az	3,581,921	0
Row3	5.26	89	8.86	7	1.78	408	375-9999	558.2	az	3,759,999	1
Row4	12.61	121	8.41	3	2.73	415	330-6626	501.9	az	3,306,626	3
Row5	18.75	118	9.18	6	1.7	510	391-8027	647.9	az	3,918,027	0
Row6	29.62	118	9.57	7	2.03	510	355-9993	779.3	orta	3,559,993	3
Row7	8.76	96	9.53	6	1.92	415	329-9001	471.9	az	3,299,001	0
Row8	29.89	90	9.71	4	2.35	408	335-4719	751.9	orta	3,354,719	0
Row9	18.87	97	14.69	5	3.02	415	330-8173	807	çok	3,308,173	0
Row10	19.42	111	9.4	6	3.43	415	329-6603	566.4	az	3,296,603	12
Row11	13.89	94	8.82	5	2.46	415	344-9403	547.1	az	3,449,403	0
Row12	8.92	128	6.35	2	3.02	408	363-1107	374.8	az	3,631,107	0
Row13	21.05	115	8.65	5	3.32	510	394-8006	596.5	az	3,948,006	3
Row14	26.11	99	9.14	6	3.54	415	366-9238	630.9	az	3,669,238	12
Row15	27.01	128	7.23	9	1.46	415	351-7269	811.3	çok	3,517,269	12
Row16	23.88	75	4.02	4	3.73	408	350-8884	566.6	az	3,508,884	0
Row17	18.55	121	5.83	3	2.19	510	386-2923	538.5	az	3,862,923	3
Row18	18.09	108	7.46	5	2.7	510	356-2992	568.2	az	3,562,992	0
Row19	13.56	74	8.68	2	3.51	415	373-2782	576.7	az	3,732,782	0
Row20	20.37	133	9.4	4	2.86	415	396-5800	603.6	az	3,965,800	0
Row21	14.44	64	9.43	6	1.54	408	393-7984	441.9	az	3,937,984	60
Row22	5.2	78	8.18	19	2.57	415	358-1958	437.7	az	3,581,958	0
Row23	11.67	105	8.53	6	2.08	415	350-2565	437.3	az	3,502,565	1
Row24	20.84	115	10.67	2	2.78	510	343-4696	563.3	az	3,434,696	0
Row25	23.55	115	11.28	5	4.19	415	331-3698	652.1	az	3,313,698	3
Row26	16.24	115	8.22	3	2.57	408	357-3817	586.8	az	3,573,817	0
Row27	13.22	68	4.59	4	3.97	408	418-6412	391.9	az	4,186,412	3
Row28	21.95	102	8.17	6	1.7	415	353-2630	629.7	az	3,532,630	0

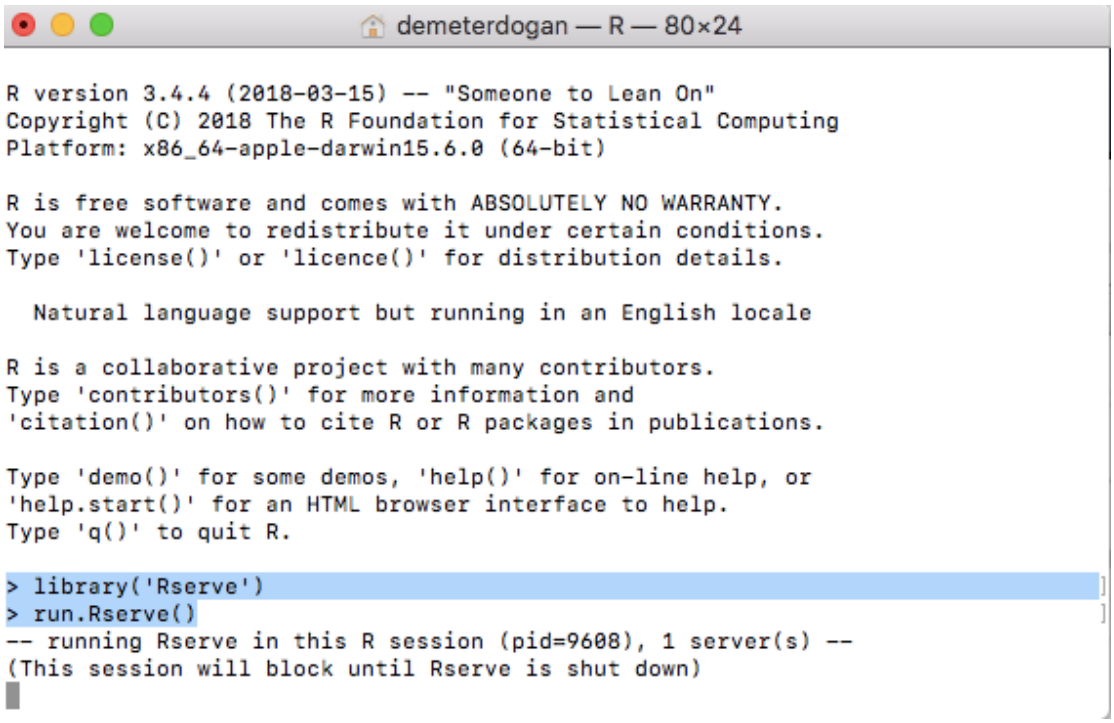
Şekil 10.1.9

Şekil 10.1.9 kod yazıldıktan sonra programın çalıştırılmasıyla elde edilen yeni kolonu göstermektedir.

Knime'da Java snippet kullanılarak çok kompleks kodlar da yazılabilir. Ama genel olarak amaç Knime içerisinde hazır bulunan operatörleri kullanmaktır.

10.2 R Snippet

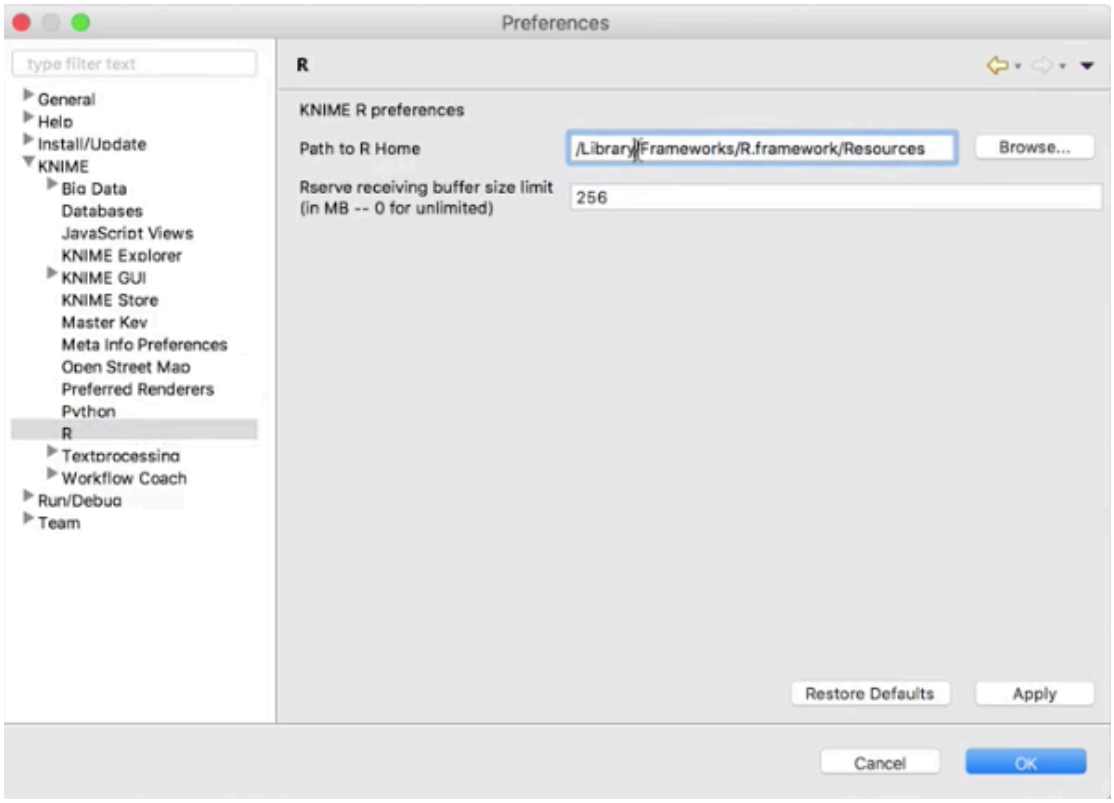
Bu bölümde R script'lerinin Knime ile nasıl birleştirilebileceği gösterilecektir. R'ın splitlerinin sağlıklı çalışabilmesi için öncelikle R kurulumunun olması gerekmektedir.



Şekil 10.2.1

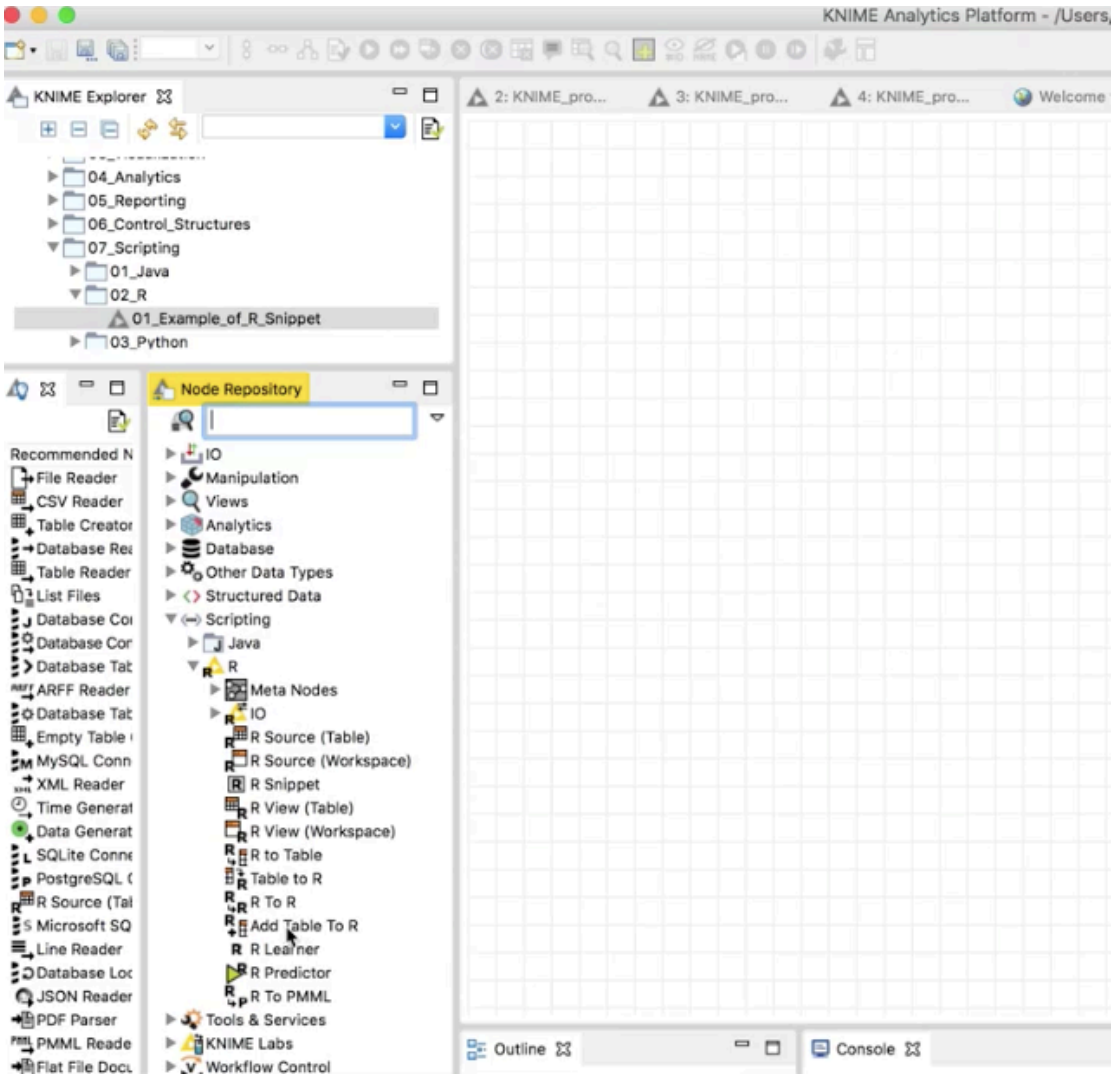
Öncelikle Rserve 'ün çalışabiliyor olması gerekmektedir. Daha sonra herhangi bir R konsolunu açarak Şekil 10.2.1, görülen R'a Rserve library'sinin kurulması ve run edilmesiyle başlanacaktır.

Ayrıca Knime preferences bölümünde kurulumlar bulunmaktadır. Örneğin, Python, R vb. Burada R'ın kurulumunun olduğu yerin dizilimi doğru verilmelidir.



Şekil 10.2.2

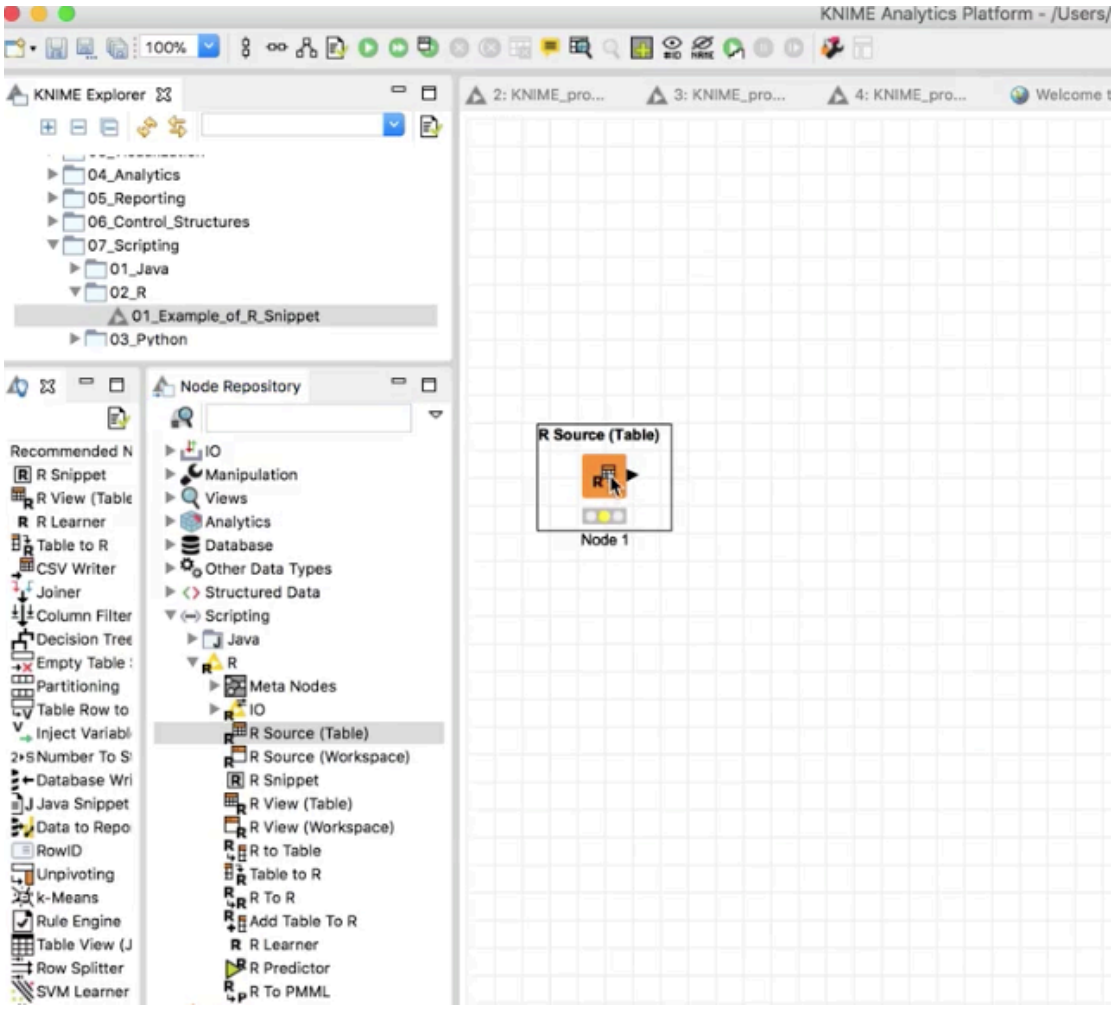
Şekil 10.2.2 'de Knime preference bölümünden açılan ve Knime seçeneği altında bulunan R penceresinde path to R home bölümünde R'ın kurulduğu yerin dizilimi görülmektedir. R, bilgisayarda hangi bölüme kurulduysa ona göre bu dizilim değişmektedir.



Şekil 10.2.3

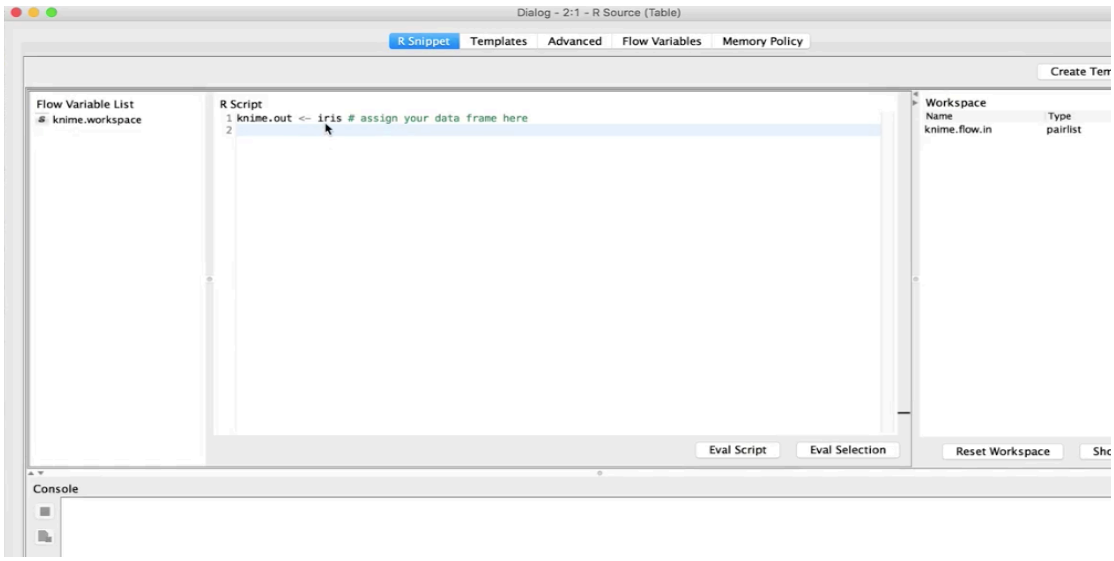
Şekil 10.2.3, bir önceki şekilde belirtilen dizilim onaylandıktan sonra R paketinin node repository bölümünde belirmesini göstermektedir. Node repository bölümünde scripting klasörü altında beliren R ve alt node'lar şekildeki gibi görülecektir.

R predictor ve R learner ile R'ın içindeki kütüphaneleri kullanılıp eğitilebilir.



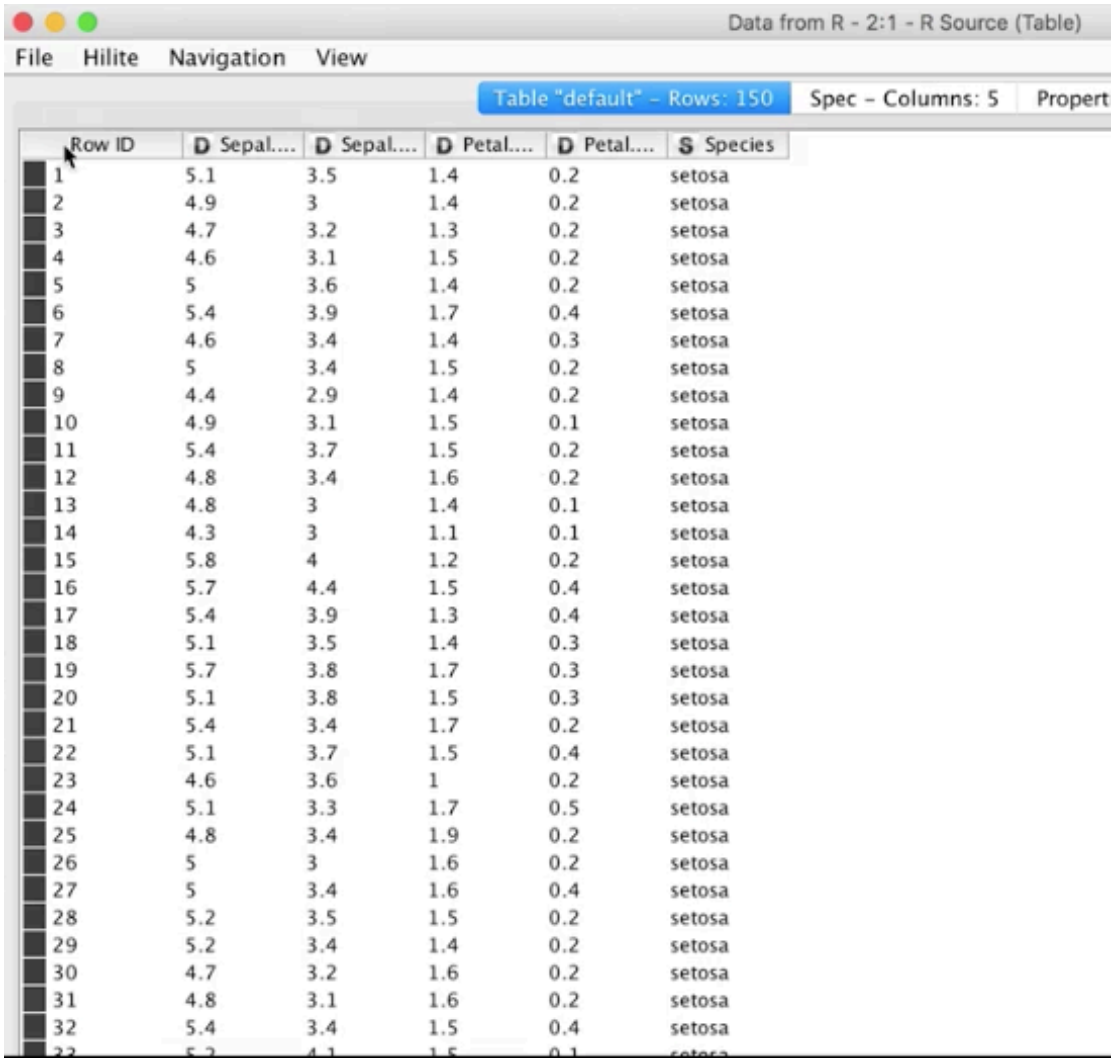
Şekil 10.2.4

Şekil 10.2.4'de bir R kaynağı yükleyebilmek için kullanılan R source node'unun workflow'a eklenmiş halini göstermektedir.



Şekil 10.2.5

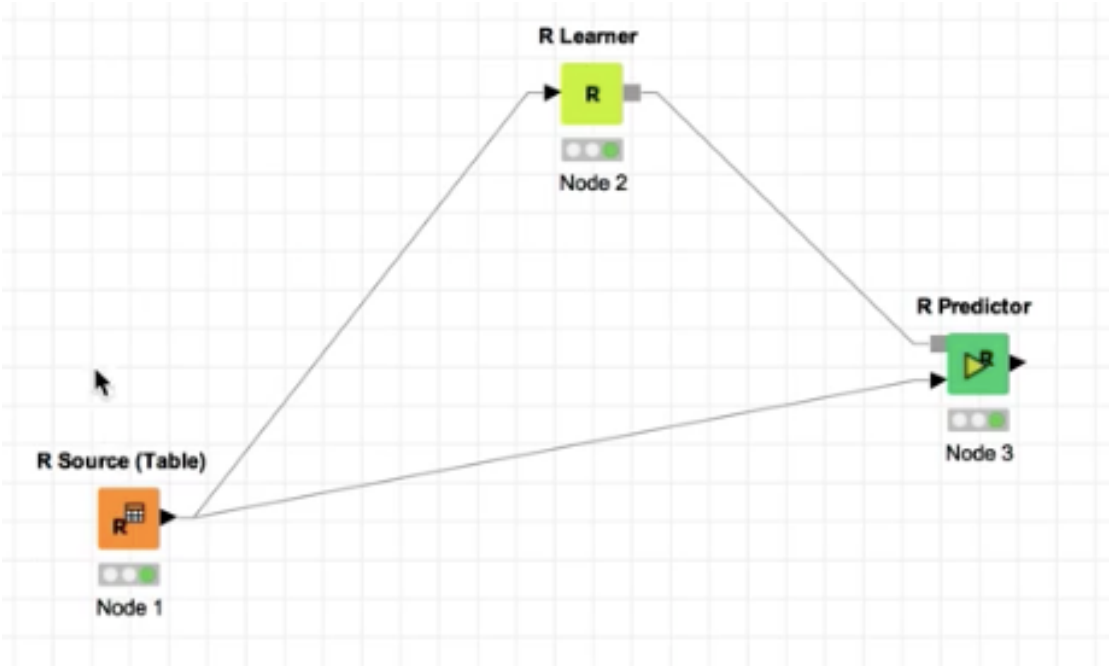
Şekil 10.2.5, R source node'unun içerisini göstermektedir. Configure bölümündeki bu bölümde knime.out yazısı kullanılan parametrenin output'unu ifade etmektedir. Yani R source node'unun veriyi akıttığı, verinin çıktığı kodu yani çıktısını (output) belirtmektedir. İris ise daha önceki bölümlerde de belirtildiği gibi hazır bir veri setidir. R'ın içerisinde de bu veri seti hazır bulunmaktadır. Ok seçeneği seçildikten sonra bir kez çalıştırılıp sonuca bakılabilir. R kullanıldığı için normalde iris veri setinin file reader ile çalıştırmaktan çok daha yavaş çalışacaktır.



Row ID	Sepal....	Sepal....	Petal....	Petal....	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3	1.4	0.1	setosa
14	4.3	3	1.1	0.1	setosa
15	5.8	4	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa
20	5.1	3.8	1.5	0.3	setosa
21	5.4	3.4	1.7	0.2	setosa
22	5.1	3.7	1.5	0.4	setosa
23	4.6	3.6	1	0.2	setosa
24	5.1	3.3	1.7	0.5	setosa
25	4.8	3.4	1.9	0.2	setosa
26	5	3	1.6	0.2	setosa
27	5	3.4	1.6	0.4	setosa
28	5.2	3.5	1.5	0.2	setosa
29	5.2	3.4	1.4	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa
31	4.8	3.1	1.6	0.2	setosa
32	5.4	3.4	1.5	0.4	setosa

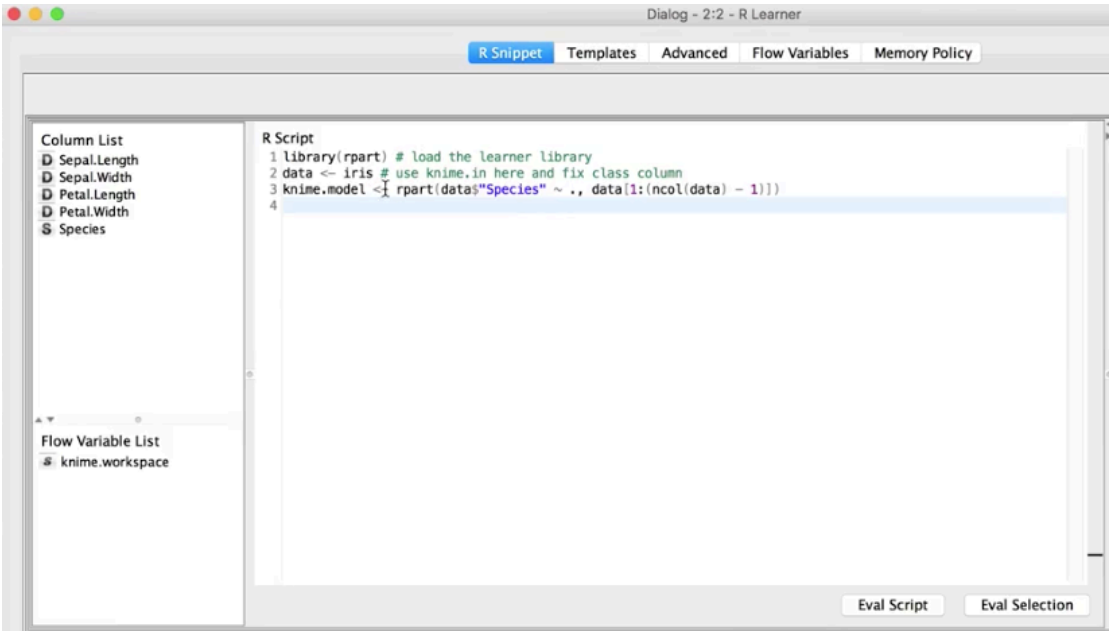
Şekil 10.2.6

Şekil 10.2.6, R source run ediltikten sonra data from R seçeneği ile ulaşılan sonuç, iris veri setinin kolonları ve türlerini göstermektedir.



Şekil 10.2.7

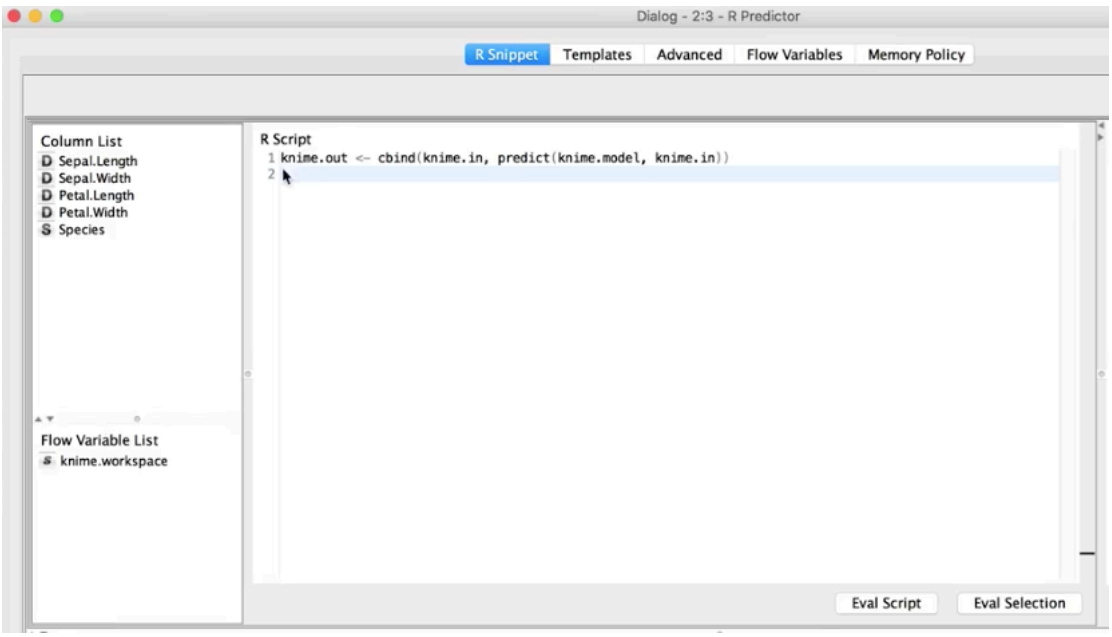
Şekil 10.2.7, R'ın kendi scripting'ini kullanarak R learner ve R predictor eklenmiş penceresini göstermektedir.



Şekil 10.2.8

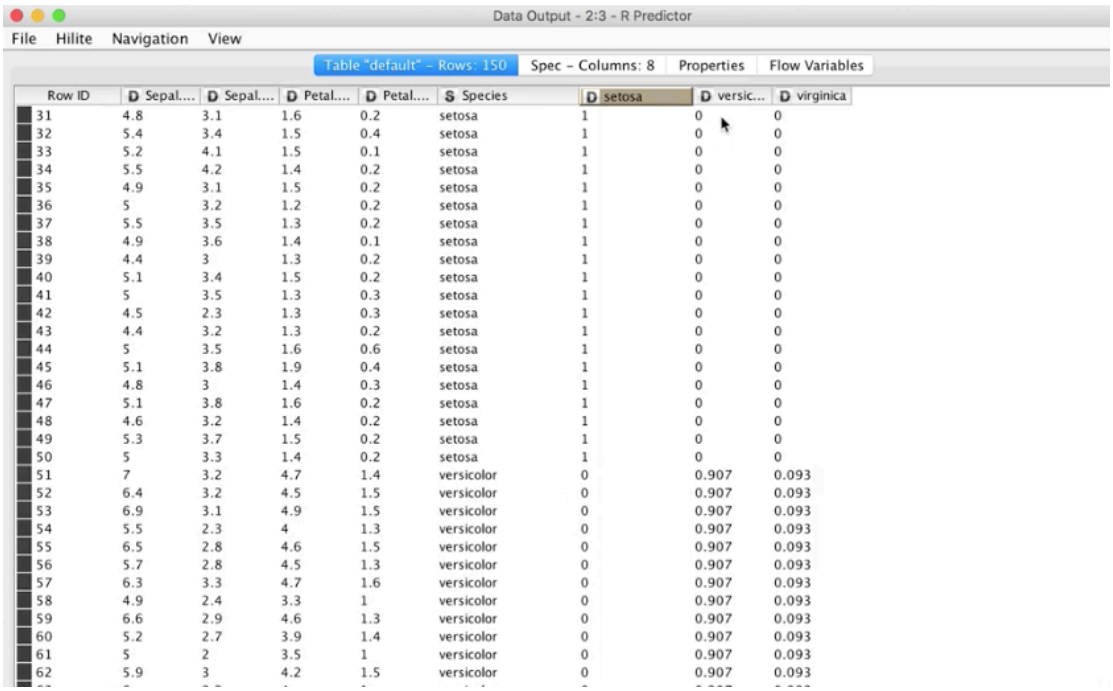
Şekil 10.2.8, R learner'ın configure penceresini göstermektedir. Data <-- iris bölümü iris veri setini aldığını, rpart veriyi böldüğünü ve data$"species" species yani türlerin yazılı olduğu kolonu kullandığı anlamına gelmektedir. Knime.model, yukarıda bahsedildiği gibi burada R learner'ın çıktısının olacağı model anlamına gelmektedir. İstenilen herhangi bir eğitim kümesiyle kullanılabilir. R içinde k-nn, naive bayes vb. Herhangi

başka bir yöntem kullanmak mümkün fakat burada herhangi bir yöntem kullanılmadı. Bu şekildeyken tamamlanıp pencere kapatılabilir.



Şekil 10.2.9

Şekil 10.2.9, R predictor'un configure penceresini göstermektedir. Burada modelden gelecek bilgi ile inputtan gelecek bilgi bind (bağlanması) edileceği cbind ile ve predict edileceği belirtilmektedir. Bu pencerede de değişiklik yapmadan direk onaylanıp run edilir.

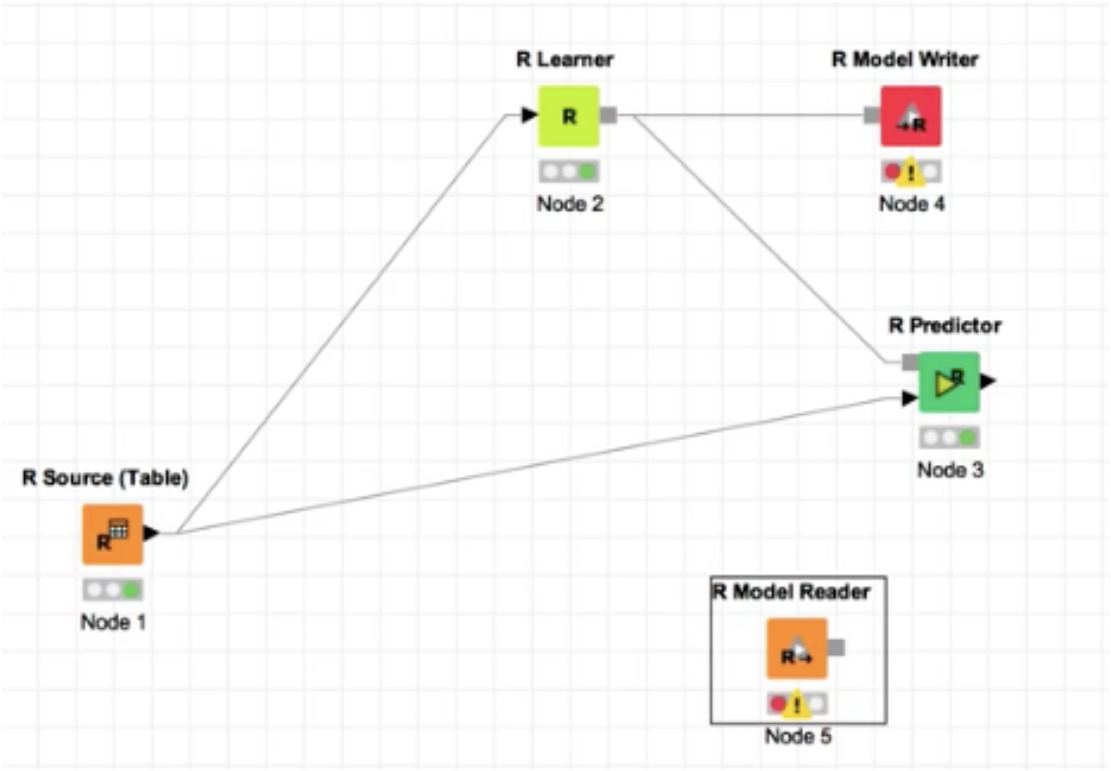


Row ID	Sepal....	Sepal....	Petal....	Petal....	Species	setosa	versic...	virginica
31	4.8	3.1	1.6	0.2	setosa	1	0	0
32	5.4	3.4	1.5	0.4	setosa	1	0	0
33	5.2	4.1	1.5	0.1	setosa	1	0	0
34	5.5	4.2	1.4	0.2	setosa	1	0	0
35	4.9	3.1	1.5	0.2	setosa	1	0	0
36	5	3.2	1.2	0.2	setosa	1	0	0
37	5.5	3.5	1.3	0.2	setosa	1	0	0
38	4.9	3.6	1.4	0.1	setosa	1	0	0
39	4.4	3	1.3	0.2	setosa	1	0	0
40	5.1	3.4	1.5	0.2	setosa	1	0	0
41	5	3.5	1.3	0.3	setosa	1	0	0
42	4.5	2.3	1.3	0.3	setosa	1	0	0
43	4.4	3.2	1.3	0.2	setosa	1	0	0
44	5	3.5	1.6	0.6	setosa	1	0	0
45	5.1	3.8	1.9	0.4	setosa	1	0	0
46	4.8	3	1.4	0.3	setosa	1	0	0
47	5.1	3.8	1.6	0.2	setosa	1	0	0
48	4.6	3.2	1.4	0.2	setosa	1	0	0
49	5.3	3.7	1.5	0.2	setosa	1	0	0
50	5	3.3	1.4	0.2	setosa	1	0	0
51	7	3.2	4.7	1.4	versicolor	0	0.907	0.093
52	6.4	3.2	4.5	1.5	versicolor	0	0.907	0.093
53	6.9	3.1	4.9	1.5	versicolor	0	0.907	0.093
54	5.5	2.3	4	1.3	versicolor	0	0.907	0.093
55	6.5	2.8	4.6	1.5	versicolor	0	0.907	0.093
56	5.7	2.8	4.5	1.3	versicolor	0	0.907	0.093
57	6.3	3.3	4.7	1.6	versicolor	0	0.907	0.093
58	4.9	2.4	3.3	1	versicolor	0	0.907	0.093
59	6.6	2.9	4.6	1.3	versicolor	0	0.907	0.093
60	5.2	2.7	3.9	1.4	versicolor	0	0.907	0.093
61	5	2	3.5	1	versicolor	0	0.907	0.093
62	5.9	3	4.2	1.5	versicolor	0	0.907	0.093

Şekil 10.2.10

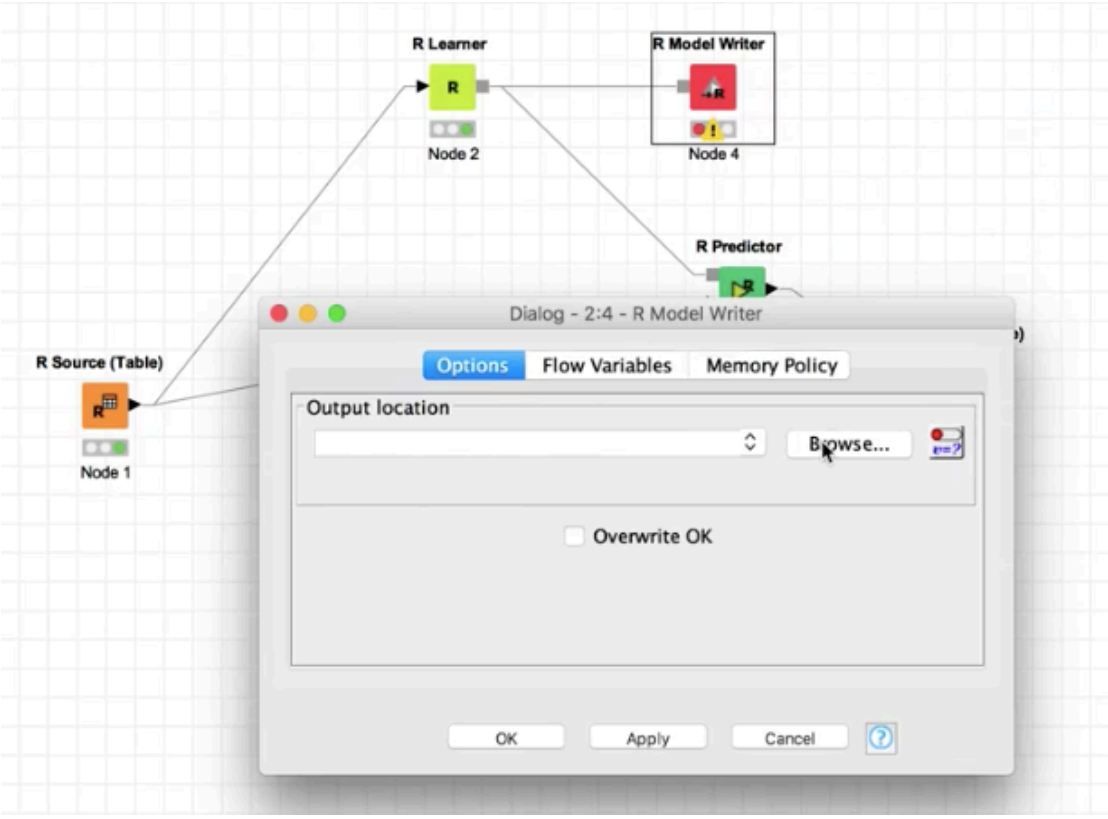
Şekil 10.2.10, program run ediltikten sonra R predictor 2ın data output penceresini göstermektedir. En sağda görülen 3 kolon; setosa, versicolor, virginica yaprak türleri model çıktısında yeni oluşan kolonlar ve tahmin edilme oranlarını belirtmektedir. Species, veri setinden gelen ve yaprakların türünün yazılı olduğu kolonu belirmektedir. Setosa, versicolor, virginica model çıktısında o yaprağın tahmin edilen tür olup olmadığını belirtmektedir. Örneğin setosa'lar 1 olarak belirtilmesi onların direk ayırt edildiği ama versicolor'ın 0.907 oranında versicolor, 0.093 oranında ise virginica olabileceği tahmin edildiğini göstermektedir.

R doğrudan makine öğrenmesi ve istatistiksel modellere daha fazla müdahale edebildiği için bir önceki bölümde gösterilen Java'ya göre Knime içerisinde daha fazla node'a (palet seçeneği) sahiptir.



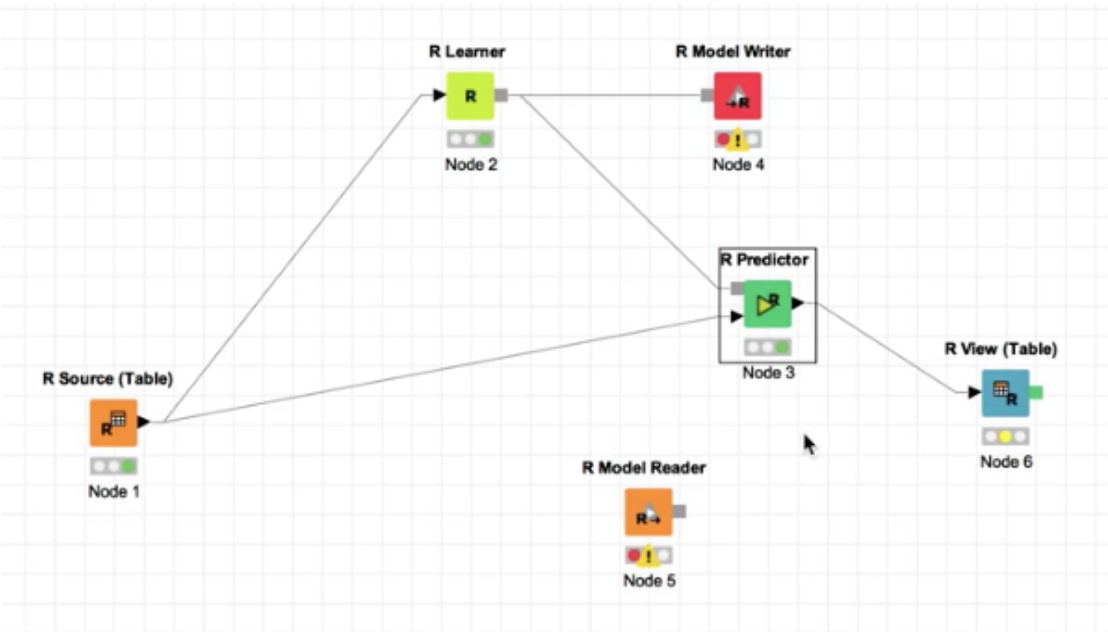
Şekil 10.2.11

Şekil 10.2.11, öğrenilen ya da oluşturulan modelin R model Writer ve R model reader ile yazılabileceği ve kaydedilerek daha sonra da bu modeli kullanmak için okutulabileceği node'ları göstermektedir.



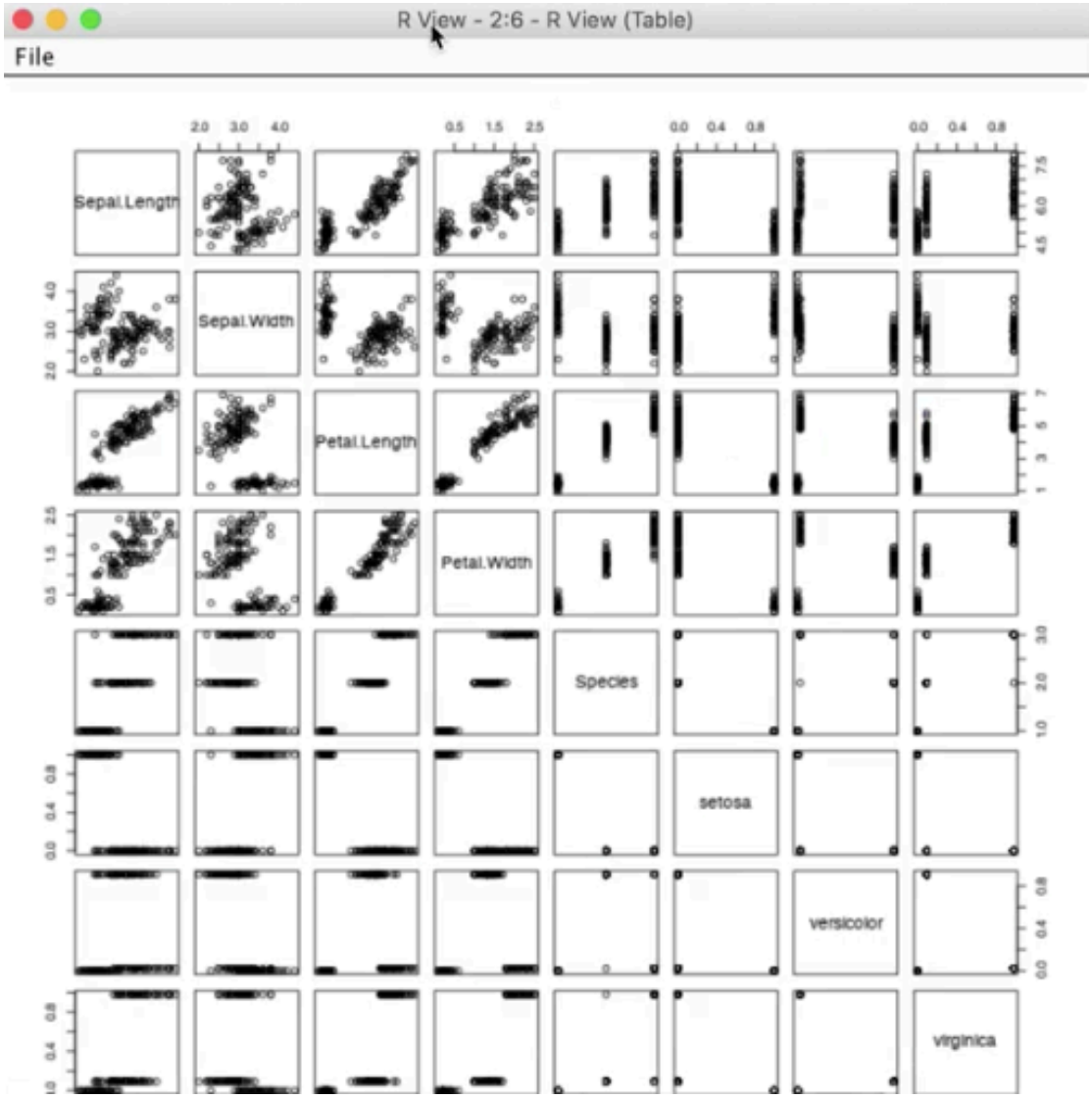
Şekil 10.2.12

Şekil 10.2.12, R Model Write'ın yani oluşturulan modelin bilgisayarda nereye yazılacağı bilgisinin verilebilmesi için configure penceresini göstermektedir. Browse seçeneğine tıklanarak açılan pencerede save as bölümüne istenilen bir isim verilerek, örneğin "kayıt" olabilir, kaydedilmelidir.



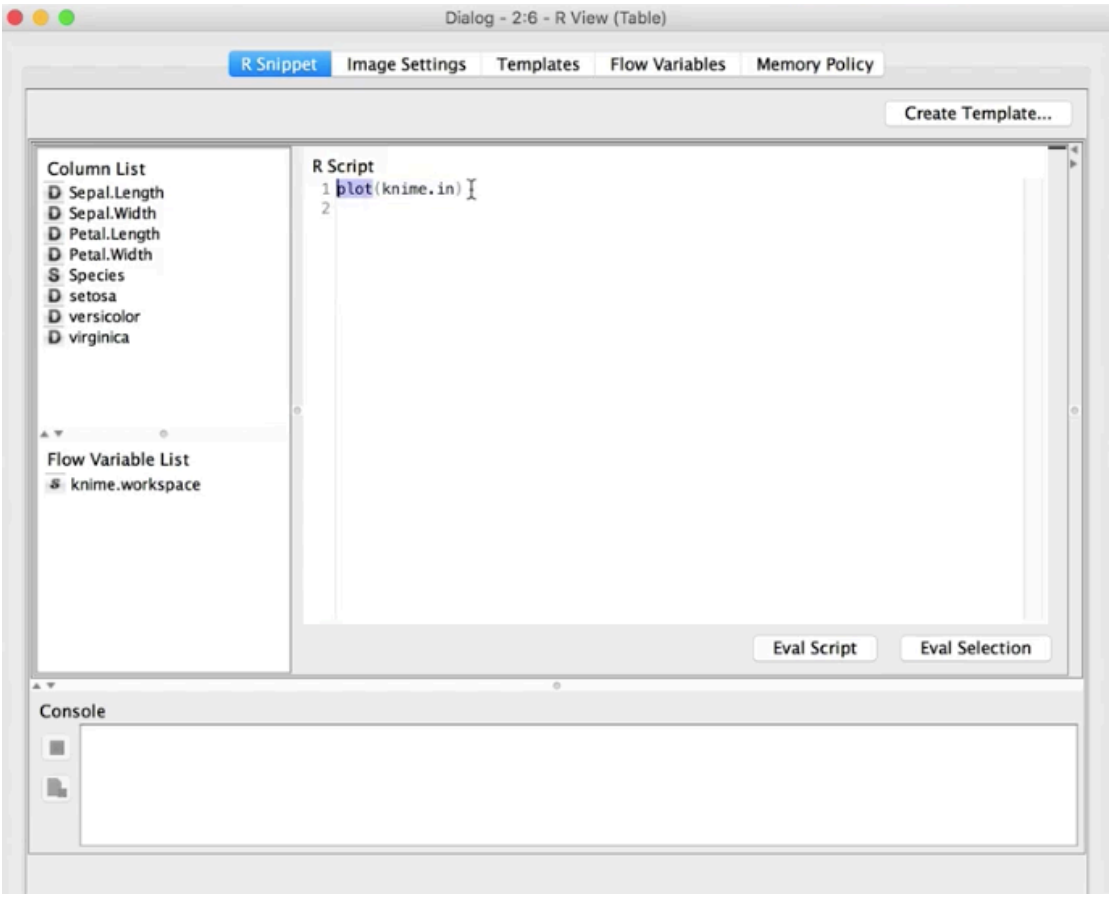
Şekil 10.2.13

Şekil 10.2.13, oluşturulan modelin table ile gösterilebilmesi için workflow'a R view (table) node'unun eklenmiş halini ve bağlantısını göstermektedir.



Şekil 10.2.14

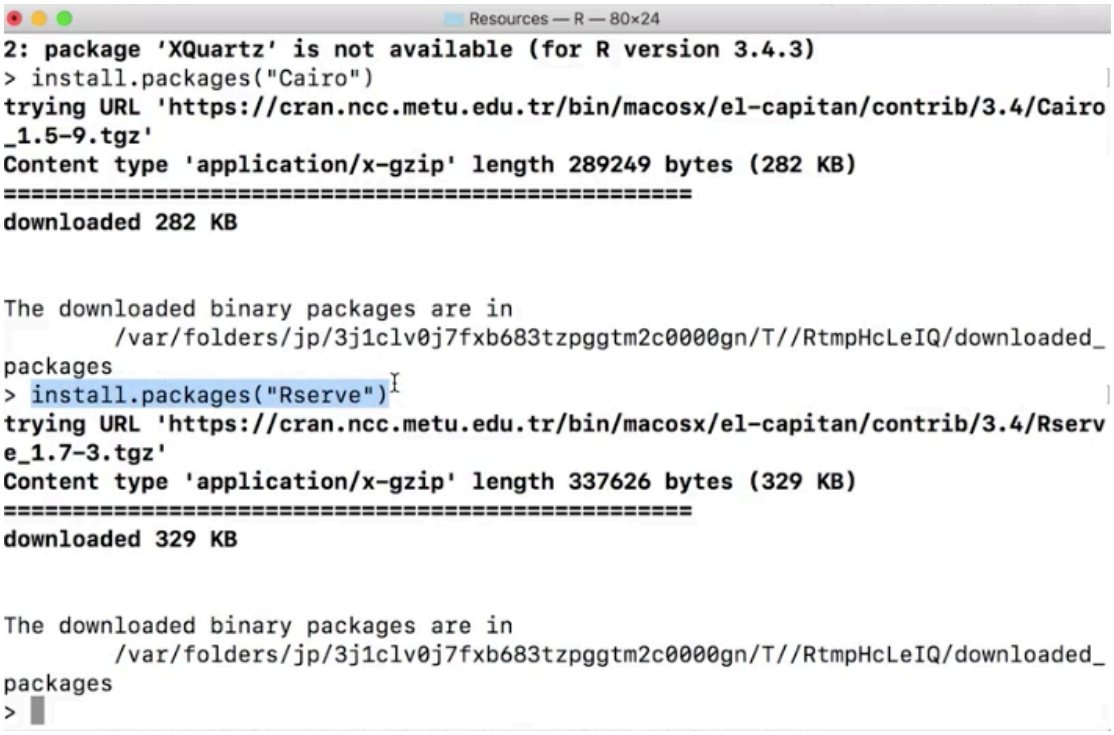
Şekil 10.2.14, R view (table) operatörünün **view:R view** bölümünün çıktısını göstermektedir. İris setosa, versicolor, virginica'nın farklı boyutlara göre görselleştirilmeleri yapılmıştır.



Şekil 10.2.15

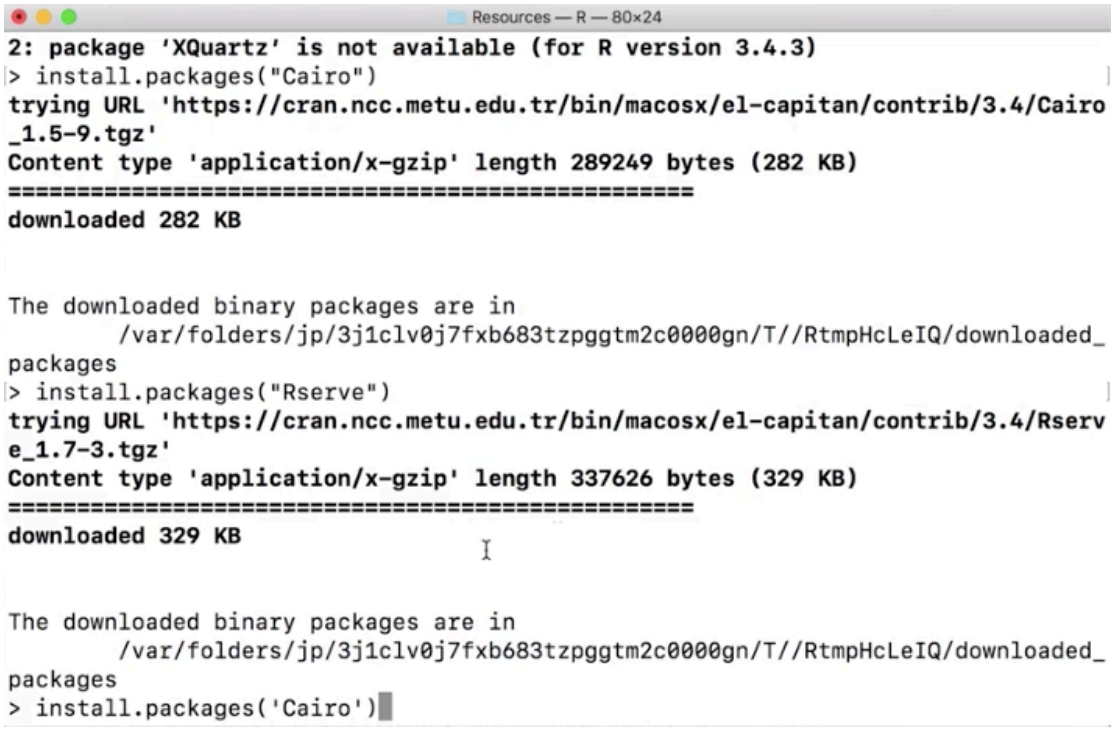
Şekil 10.2.15 R view (table) node'unun configure penceresini göstermektedir. Burada istenilen bir formatta tablo uluşturulması için bilginin verilmesi gerekmektedir. R'ın içerisindeki visualization (görselleştirme araçları) hangisi istenirse o kullanılabilir. Sonuç belirtilen formatta çıktı olarak görülebilecektir.

Burada en önemli durum R serve'ün çalışabiliyor olması gerekmektedir. Görselleştirme gibi özelliklerin çalışabilmesi için cairo 'nun install edilmiş olması gerekmektedir.



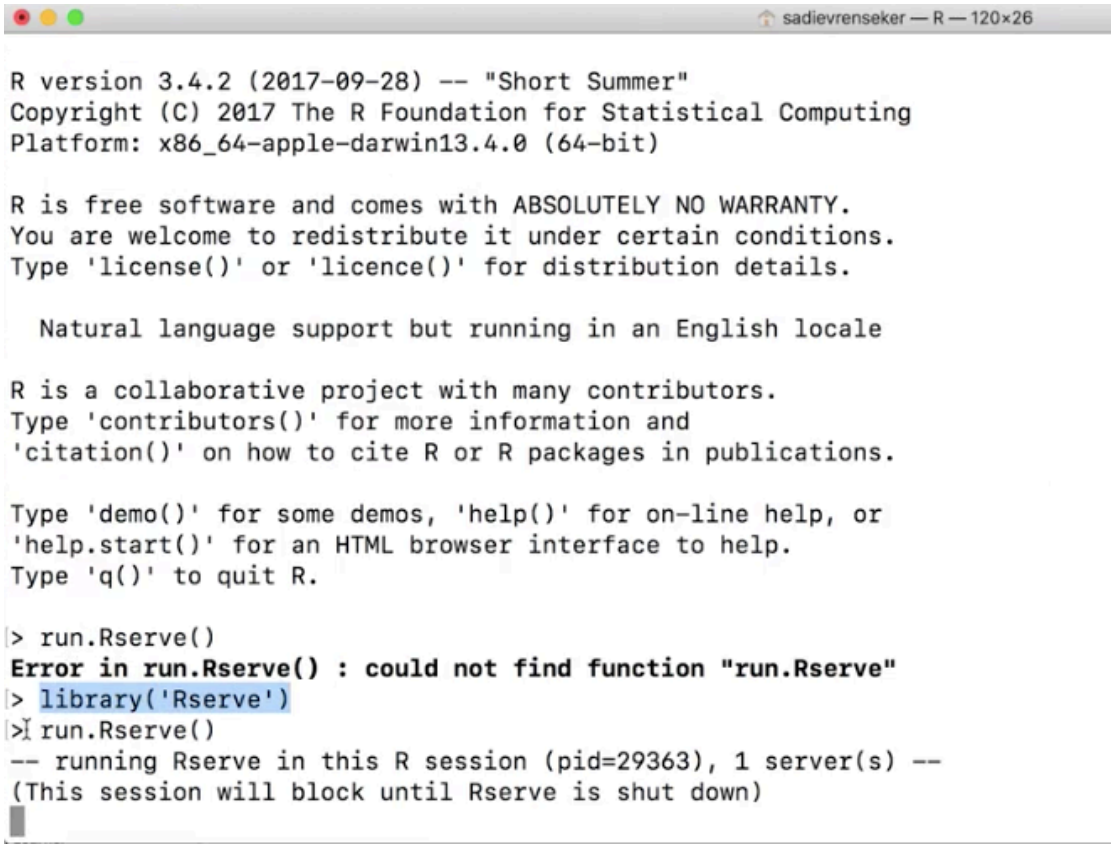
Şekil 10.2.16

Şekil 10.2.16, R için install edilebilmesi için komutun yazıldığı pencereyi göstermektedir.



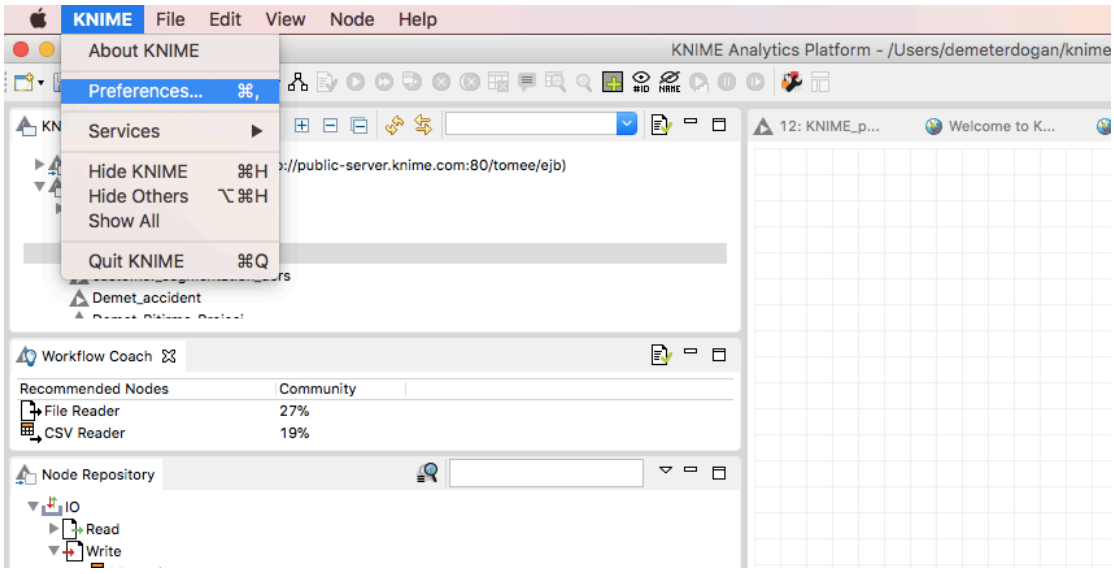
Şekil 10.2.17

Şekil 10.2.17, eğer cairo paketi indirilmek istenirse direk olarak install.packages('cairo') yazılarak da install edilebileceğini göstermektedir.



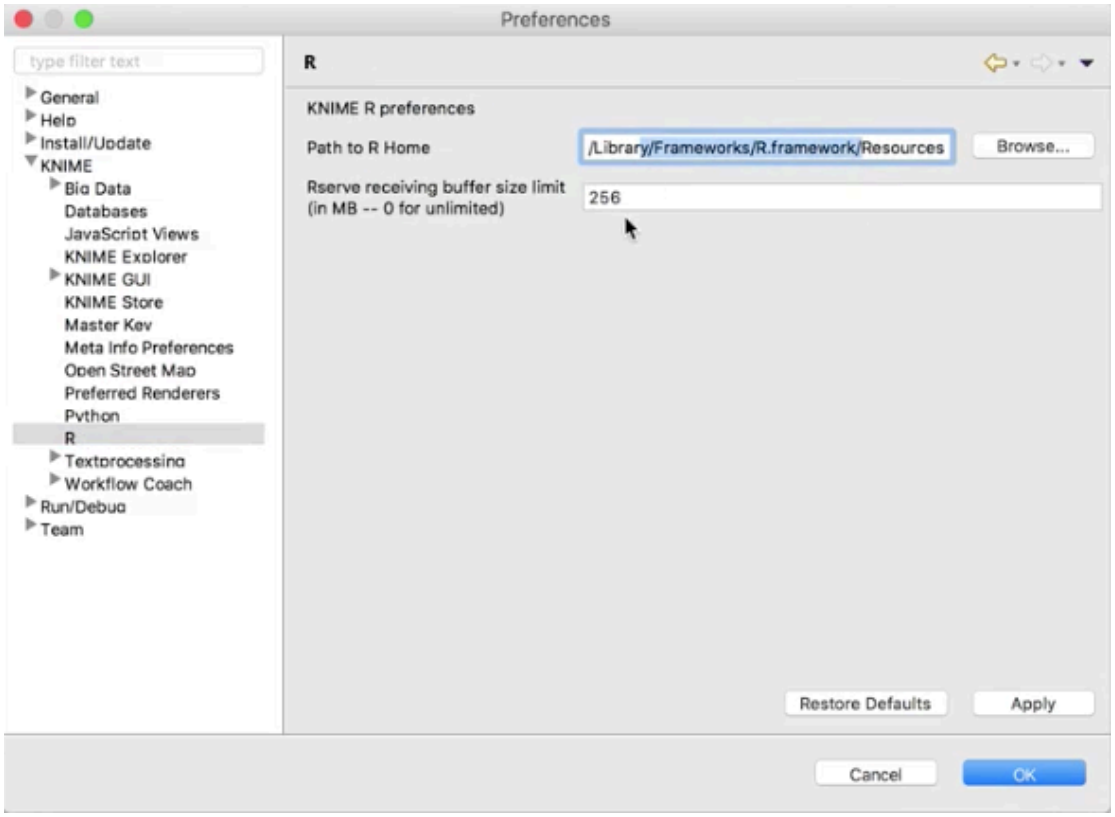
Şekil 10.2.18

Şekil 10.2.18, indirilen paketin library('Rserve') ve run.Rserve() diyerek yüklenmesi ve çalıştırılması gereken ve komutun yazıldığı pencereyi göstermektedir.



Şekil 10.2.19

Şekil 10.2.19, R console da indirilen ve run edilen paketlerden sonra son olarak Knime penceresinden üst sol kısımda olan Knime bölümünden preferences a girilmelidir.



Şekil 10.2.20

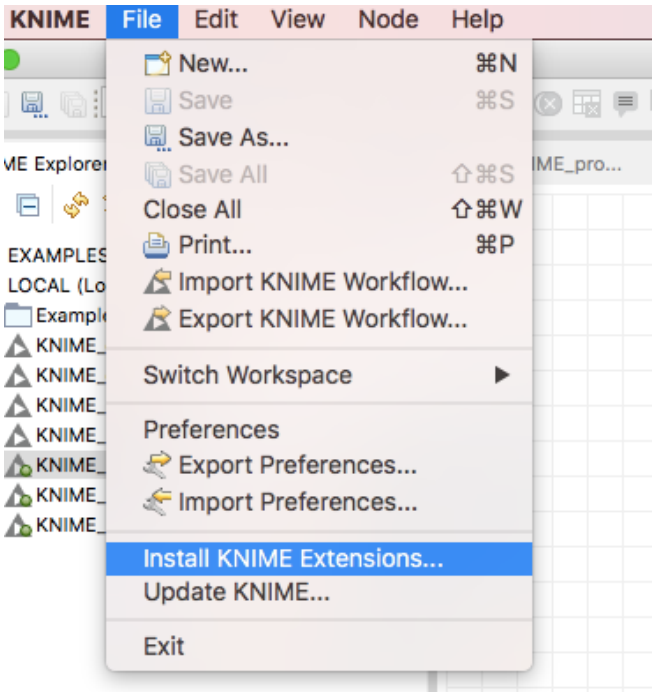
Şekil 10.2.20, bir önceki şekilde preferences seçildikten sonra açılan pencereyi göstermektedir. Burada Knime klasörü seçilerek Knime GUI başlığı altında R seçilmelidir. Burada daha önce de bahsedildiği gibi Path to R home bölümüne R 'ın olduğu path'in yani dizilimi doğru verilmiş olması gerekmektedir.

Rserve seceiving buffer size limit'de ise ram'in buffer için ne kadar ayrılacağının belirtildiği bölümü göstermektedir. Bzen büyük dosyalarla çalışırken ram yetersiz gelebilir, çok büyük veriler için bilgisayar da yetersiz gelebilir fakat buradan biraz daha büyültülerek veri üzerinde çalışılabilir.

Aslında önerilen R'da yapılacaklar bölümü R da yapılmalı Knime da yapılacak şeyler Knime da yapılmalı ve taşınmak istenilen durum varsa daha sonra R model writer ve r model reader ile bağlanılarak devam edilmelidir. Nedeni, R ın yaptıklarını Knime'a node'lar aracılığıyla taşımak performansı olumsuz etilemesidir.

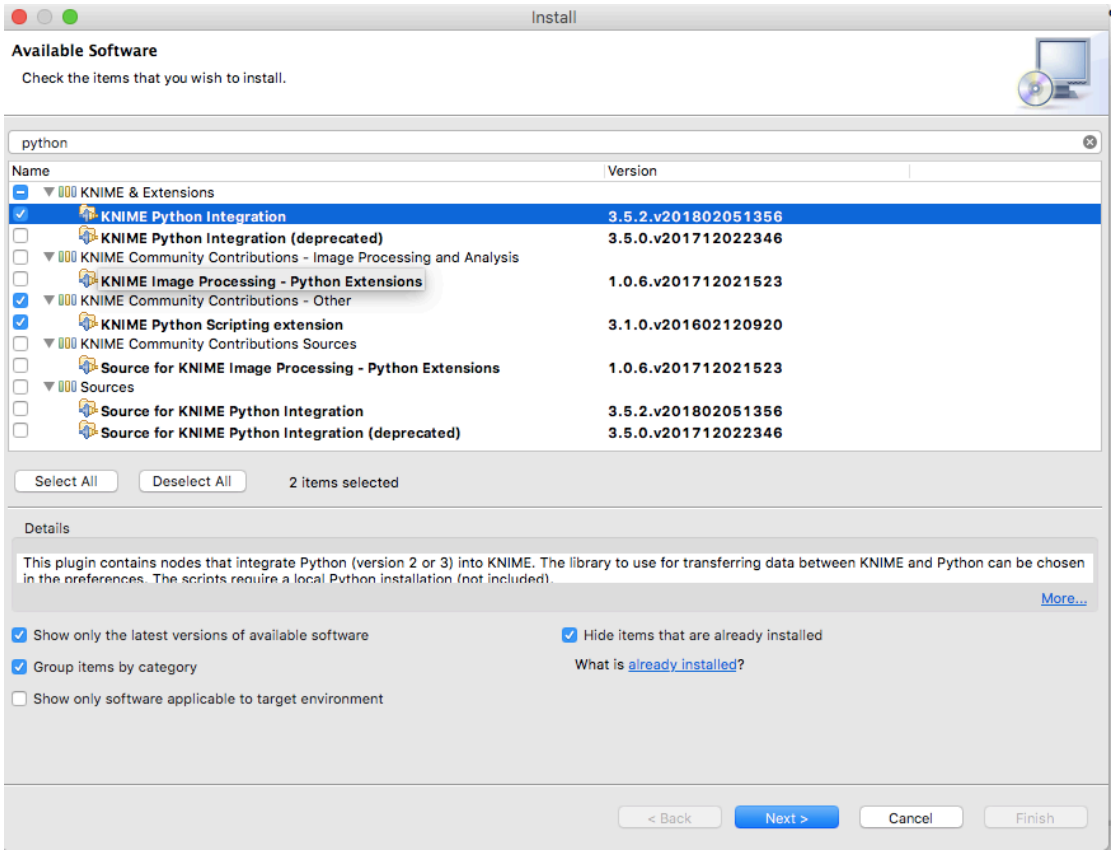
10.3 Python Snippet

Bu bölümde amaç Knime'da python kullanımını göstermek. Knime'da python kurulu gelmez.



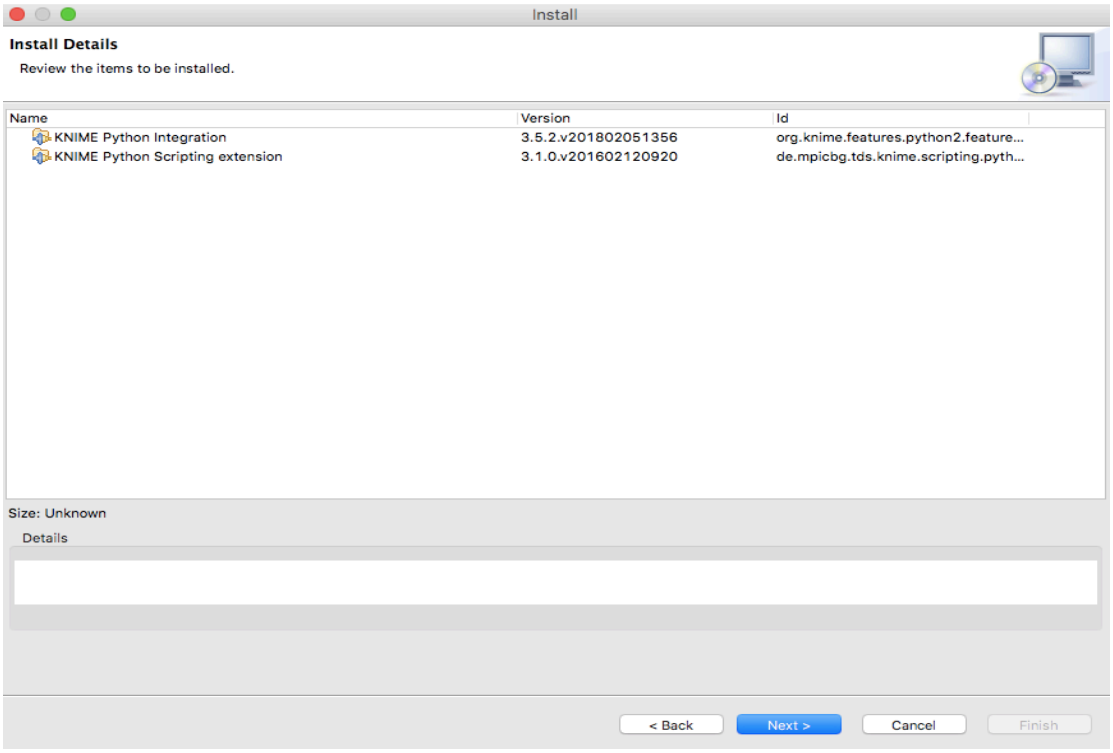
Şekil 10.3.1

Şekil 10.3.1 Knime'da Python kurmak için paketlerin indirileceği "Install Knime extensions" butonunun yerini göstermektedir.



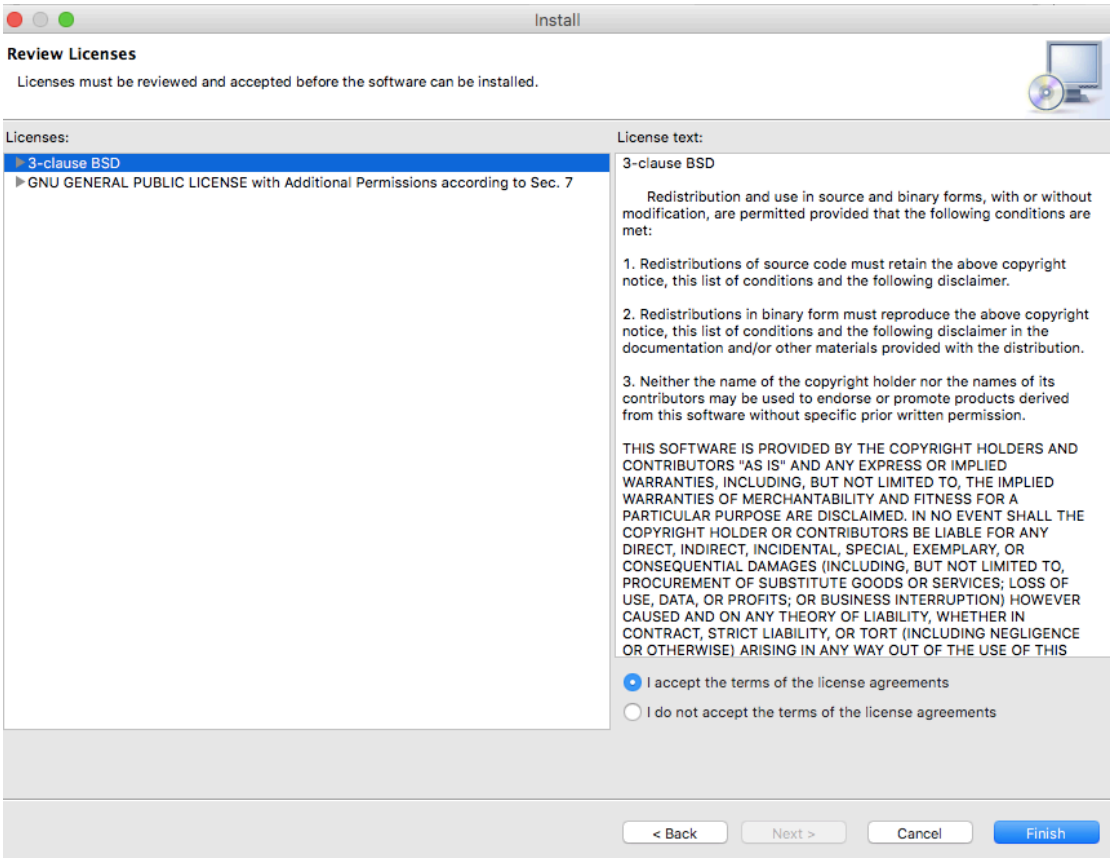
Şekil 10.3.2

Şekil 10.3.2, Install Knime extensions butonuna bastıktan sonra açılan pencereyi ve arama butonuna Python yazılarak python ile ilgili klasörlerin gelmesi bulmak açısından faydalı olacaktır. “knime python integration” ve “knime community contributions-other” seçenekleri seçilmelidir. Daha sonrasında da next tuşuna basılmalıdır.



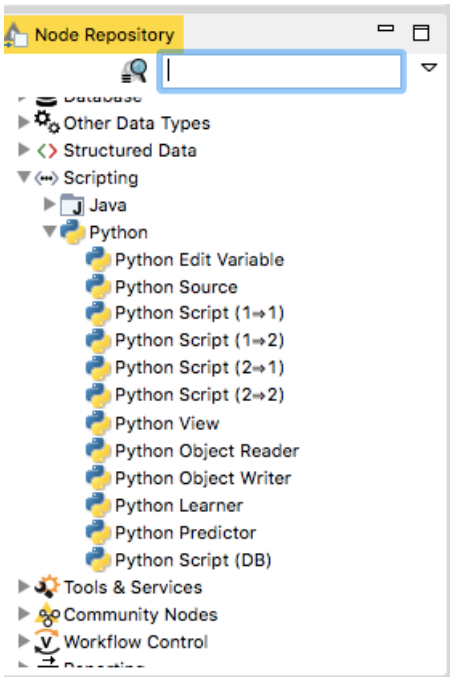
Şekil 10.3.3

Şekil 10.3.3 Bir önceki şekilde seçilen klasörlerin listesini göstermektedir.



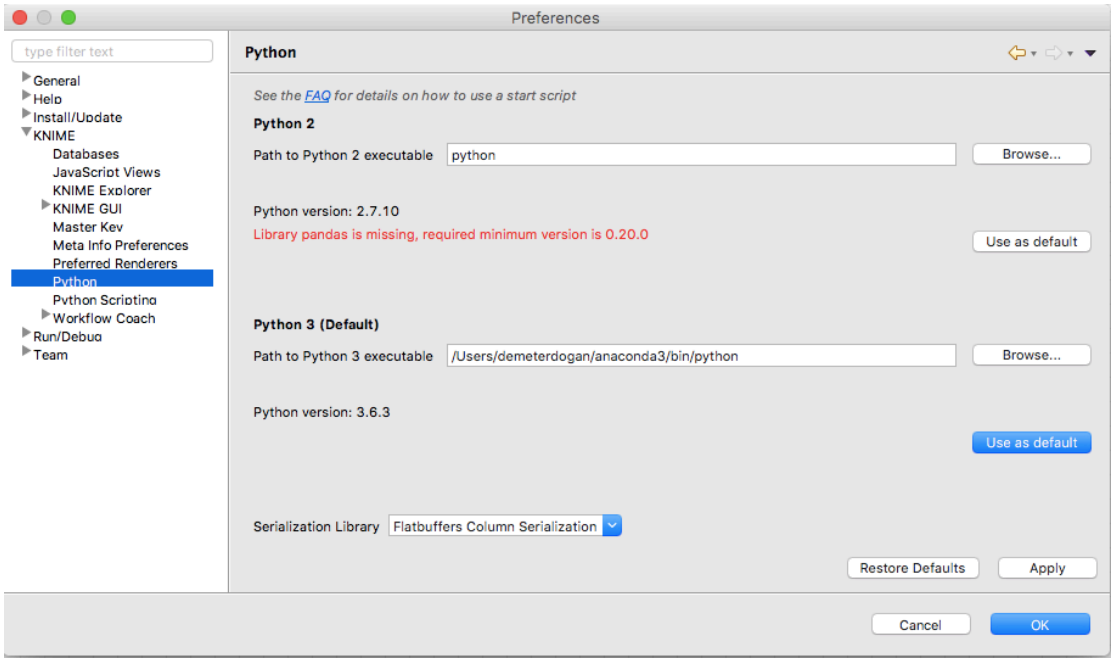
Şekil 10.3.4

Şekil 10.3.4 De görülen işlemi tamamladıktan sonra Pythonla ilgili paketler kurulmuş olacaktır.



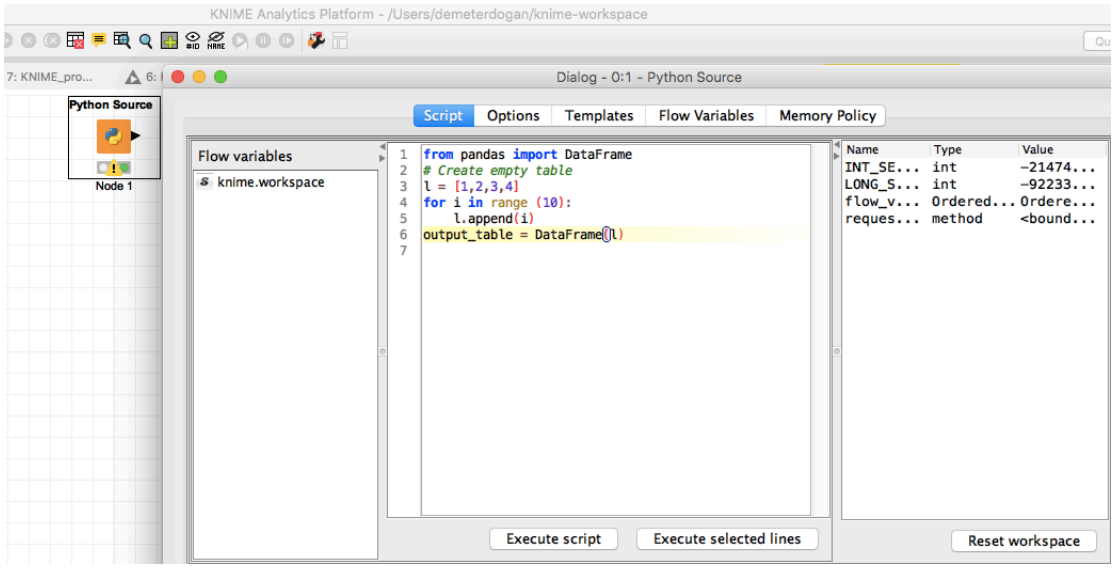
Şekil 10.3.5

Şekil 10.3.5, kurulum tamamlandıktan sonra repository bölümünde scripting altında python ile ilgili klasörler çıkmalıdır.



Şekil 10.3.6

Şekil 10.3.6 preferences' girildiğinde KNIME sekmesi altında python seçeneğine girilerek python kurulumunun yapıldığı yerden browse edilmelidir. Python 2 versiyonları kullananlar için 2, python 3 versiyonunu kullananlar python 3 bölümünden browse etmelidirler.



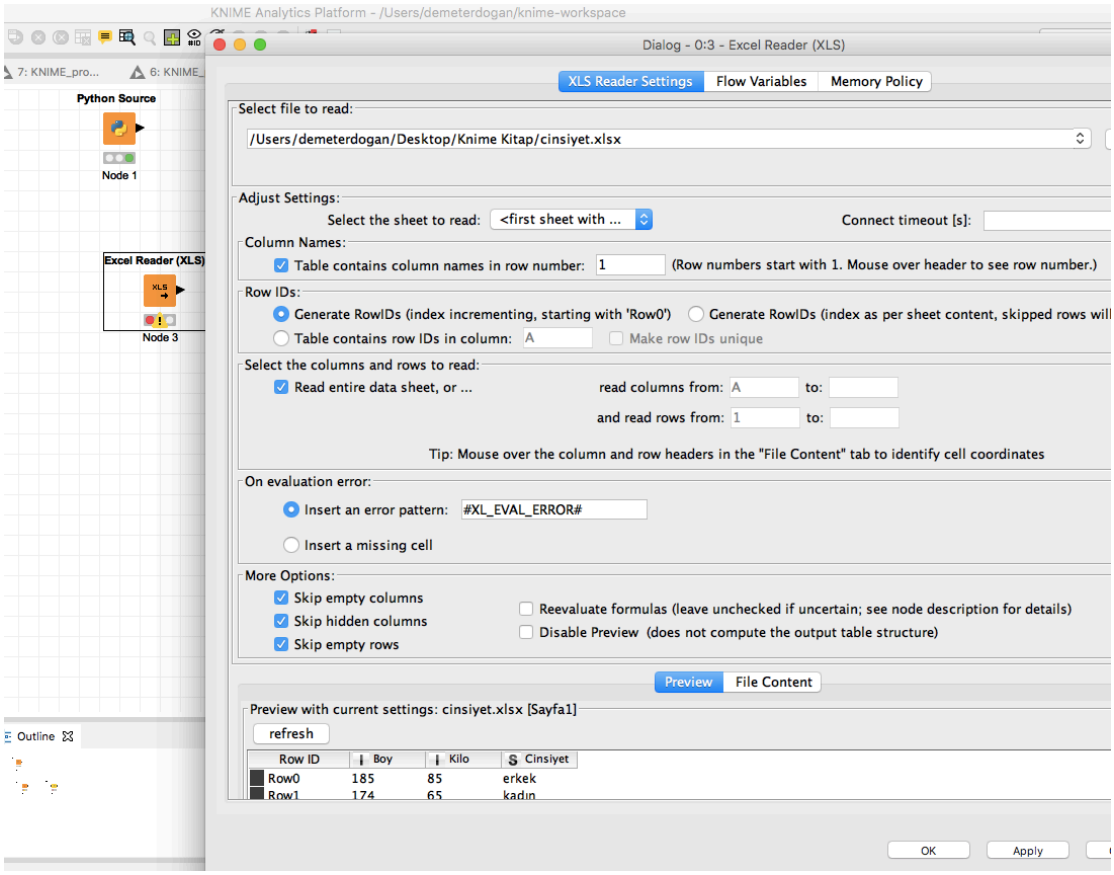
Şekil 10.3.7

Şekil 10.3.7, sisteme python source operatörünün eklenmesini ve configure bölümünde yazılan kodu göstermektedir.

Şekil 10.3.8

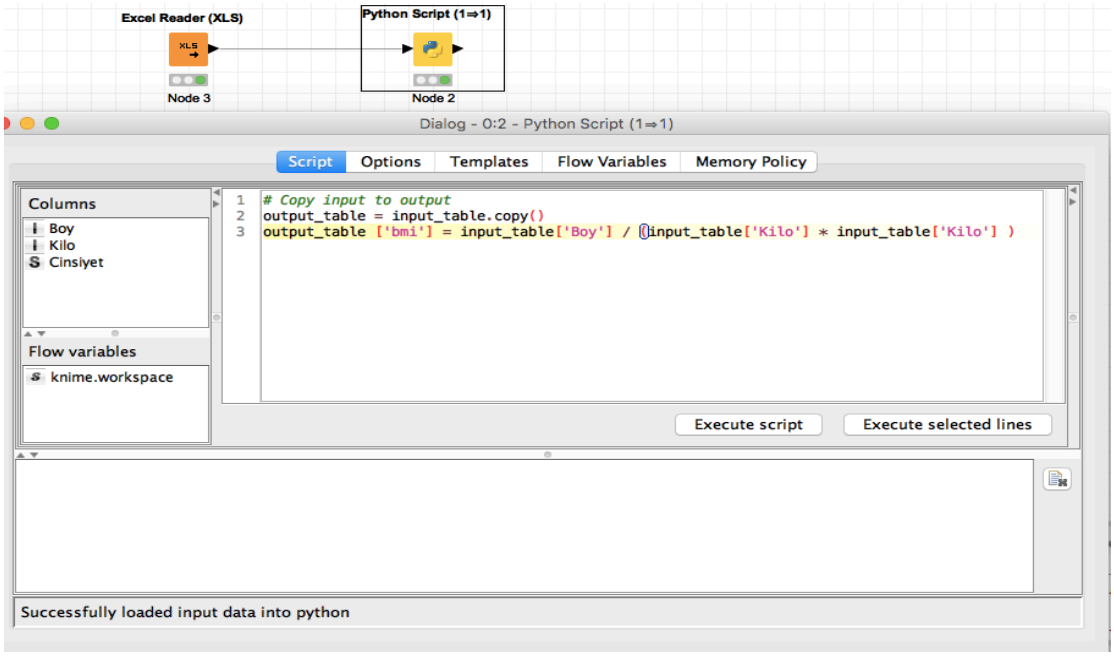
Şekil 10.3.8, yukarıda yazılan kodu run edildiğinde elde edilecek tabloyu göstermektedir. 1'den 4 e kadar sayılar ve sonrasında da 0'dan 9 a kadar sayıları döndürmüştür.

(1→1)



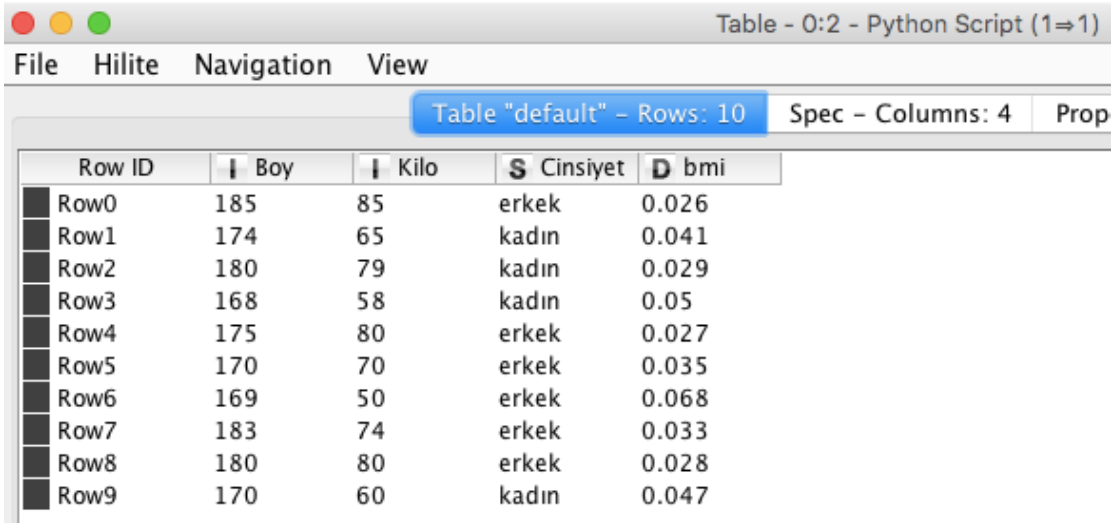
Şekil 10.3.9

Şekil 10.3.9 sisteme excel reader eklenmesini ve daha önce kullanılan cinsiyet dosyasının browse edilmesini göstermektedir. İçeriğinde boy, kilo ve cinsiyet kolonları olan bir örnek excel dosyası istenilen rakamlarla oluşturulabilir.



Şekil 10.3.10

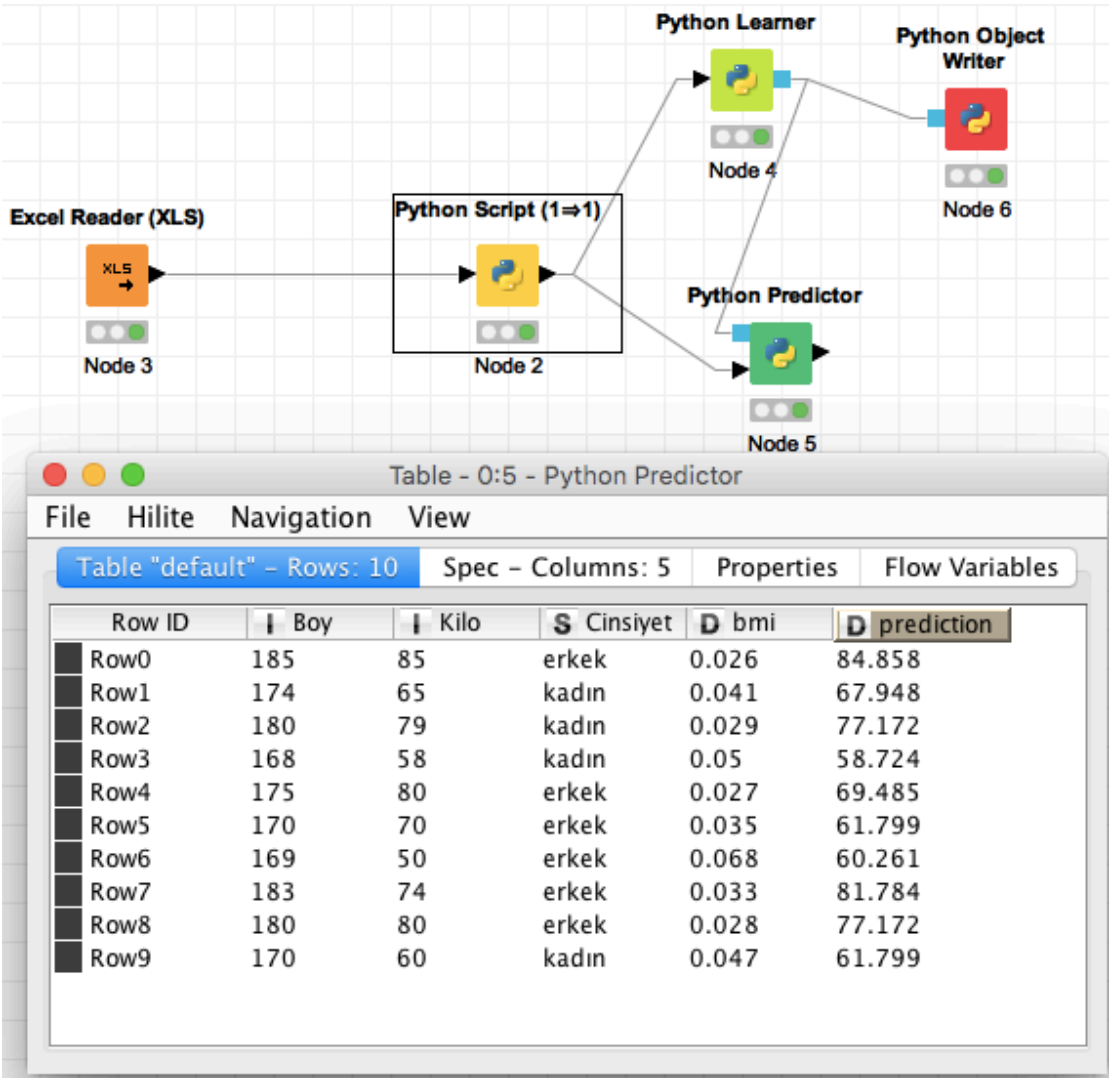
Şekil 10.3.10 sisteme python script (1→1) operatörünün eklenmesini ve configure bölümüne yazılan kodu göstermektedir. (2 →1) 2 input bir outputu, (1→1) bir input bir outputu göstermektedir.



Row ID	Boy	Kilo	Cinsiyet	bmi
Row0	185	85	erkek	0.026
Row1	174	65	kadın	0.041
Row2	180	79	kadın	0.029
Row3	168	58	kadın	0.05
Row4	175	80	erkek	0.027
Row5	170	70	erkek	0.035
Row6	169	50	erkek	0.068
Row7	183	74	erkek	0.033
Row8	180	80	erkek	0.028
Row9	170	60	kadın	0.047

Şekil 10.3.11

Şekil 10.3.11 program çalıştırıldıktan sonra elde edilen sonuç tablosunu göstermektedir. Bmi, yukarıda ismi verilen kolon ve yazılan formül sonucu altındaki değerler de hesaplanan değerlerdir. Boy, Kilo ve cinsiyet kolonları ise sisteme aktarılan örnek veri kümesinden gelen kolonlardır.



Row ID	Boy	Kilo	Cinsiyet	bmi	prediction
Row0	185	85	erkek	0.026	84.858
Row1	174	65	kadın	0.041	67.948
Row2	180	79	kadın	0.029	77.172
Row3	168	58	kadın	0.05	58.724
Row4	175	80	erkek	0.027	69.485
Row5	170	70	erkek	0.035	61.799
Row6	169	50	erkek	0.068	60.261
Row7	183	74	erkek	0.033	81.784
Row8	180	80	erkek	0.028	77.172
Row9	170	60	kadın	0.047	61.799

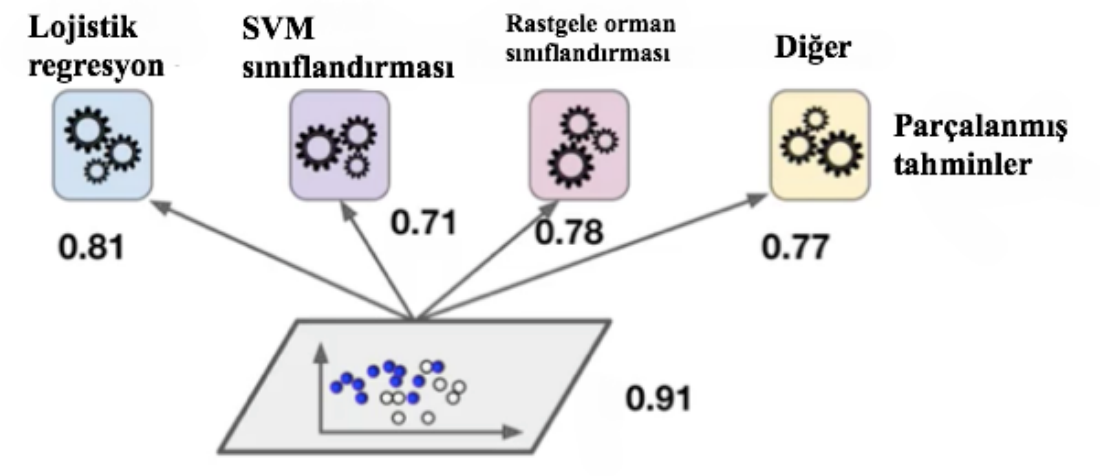
Şekil 10.3.12

Şekil 10.3.12 sisteme daha önceki bölümlerde olduğu gibi python learner, python predictor ve python writer eklenmesini ve direk bu haliyle program çalıştırıldıktan sonra da python predictor'de oluşan sonuç tablsunu göstermektedir. Boy'dan kilo tahmini yapan algoritma otomatik içeriğinde yazılı geldiği için prediction kolonunda kilo tahminleri bulunmaktadır.

11. ÜST ÖĞRENME ALGORİTMALARI (META LEARNER)

11.1 Ensemble Yöntemleri ve Bagging, Boosting ve Fusion Kavramlarına Giriş

Bu bölümde ensemble yöntemleri gösterilecektir. Ensemble learning Türkçe'de kollektif öğrenme olarak da geçmektedir. Bir algoritma ile işlem yapıldığında bunun bir sonucunda bir başarı oranı oluşur. Boosting, Bagging (random forest), AdaBoost, Stacking (blending, MAVL) bu bölümde açıklanmaya çalışılacaktır.

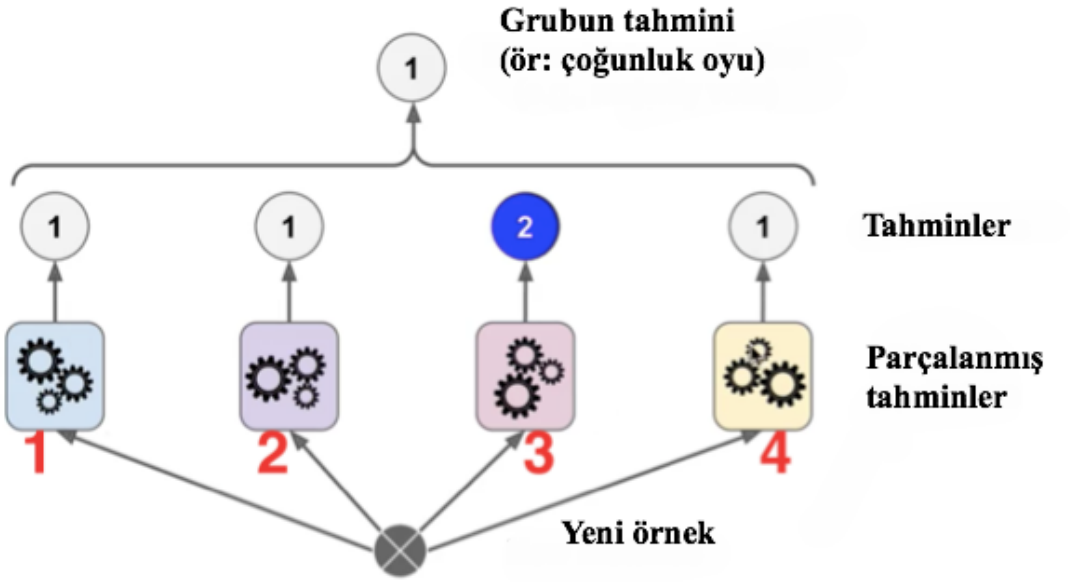


Şekil 11.1.1

Şekil 11.1.1, logistic regression başarı oranını 0.81 (81%), SVM classifier başarı oranını 0.71, random forest classifier başarı oranını 0.78, other (diğer yöntemlerinkini) 0.77 olarak görülmektedir. Ensemble learning bu tüm algoritmaları kullanarak başarı oranını yukarı çekmeyi amaçlar.

Çoğunluk Oylaması (Majority Voiting) (MAVL)

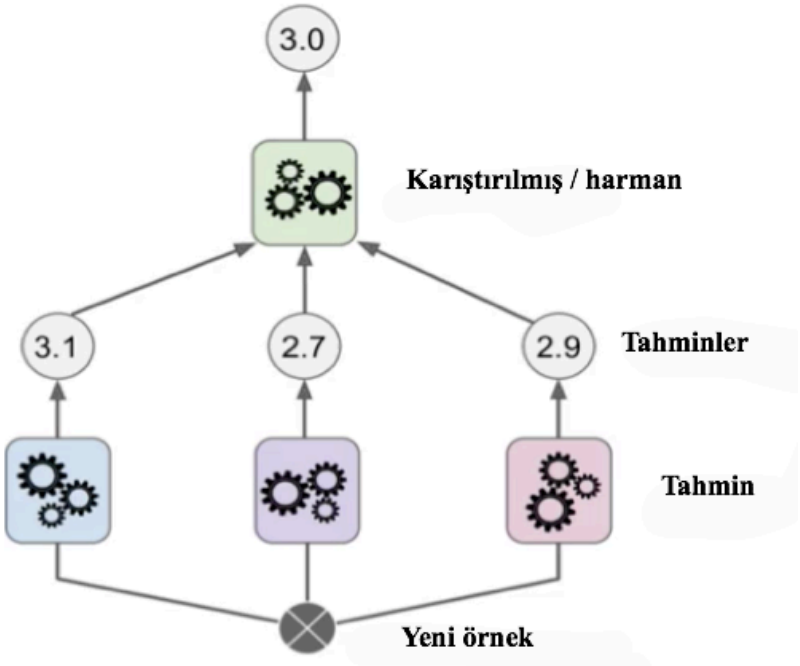
Çoğunluğun dediğinin olduğu oylama yöntemi demektir. En basit ve en etkili yöntemlerden biri sayılabilir.



Şekil 11.1.2

Şekil 11.1.2 çoğunluk oylaması örneğini göstermektedir. Örneğin birinci algoritmadan 1. Sınıf, 2. Algoritmadan 1. Sınıf, 3. Algoritmadan 1.sınıf ve 4. Algoritmadan 1. Sınıf sonucu çıkmıştır. Bu yüzden sonuç olarak 1. Sınıf seçilir. Eğer 2 tane 2. Sınıf seçilseydi algoritmaların başarı yüzdeliklerine bakılırdı eğer her şey aynı olsa o zaman iki seçenekten biri seçilirdi. Algoritmaların ağırlıklarına da bakılarak sonuca varılabilir.

Stacking (istifleme)

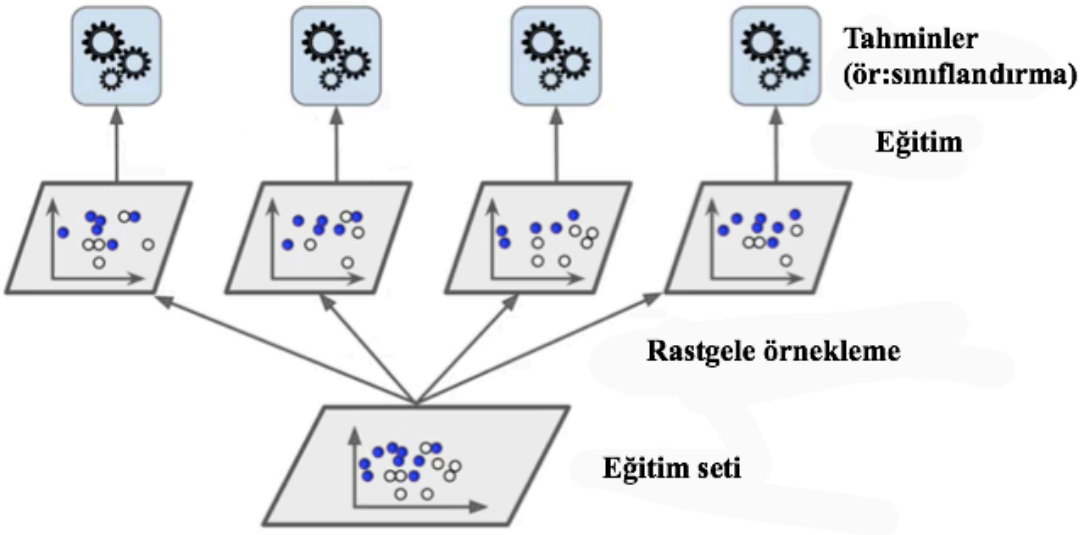


Şekil 11.1.3

Şekil 11.1.3, yukarıda örnekte açıklandığı gibi sınıflandırma algoritmasının sonuçlarının ağırlıklarıyla verilmesini göstermektedir. Bu ağırlıklar blend edilir yani orta noktası bulunur. Farkları, yukarıdaki classification için kullanılırken bu prediction için kullanılmaktadır.

Bagging (BootStarp Aggregation)

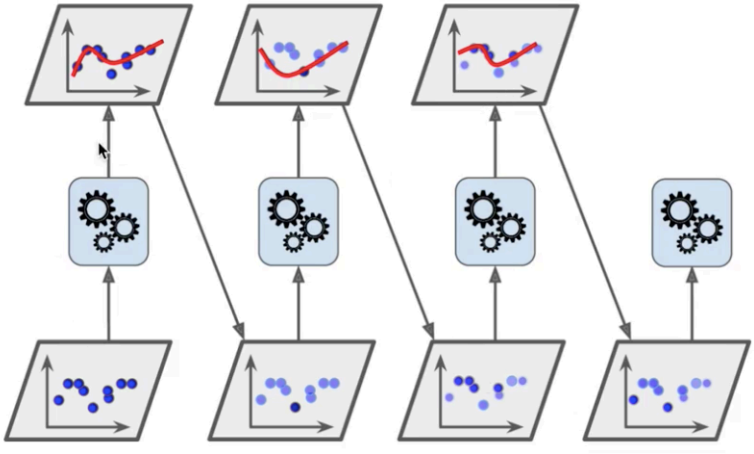
Burada bir eğitim kümesiyle algoritmalar öncelikle train edilir. Veri setinin tamamı algoritmalara verilmez, her algoritmaya farklı kısmı verilerek eğitilir. Algoritmaların testlerinden çıkan sonuçlar da her algoritmanın train için verilen veri setinin bölgesi için tahminde bulunur.



Şekil 11.1.4

Şekil 11.1.4 bir veri setinin parçalar halinde algoritmalarda training için kullanılmasını ve sonra test edilmesini göstermektedir.

AdaBoost



Şekil 11.1.5

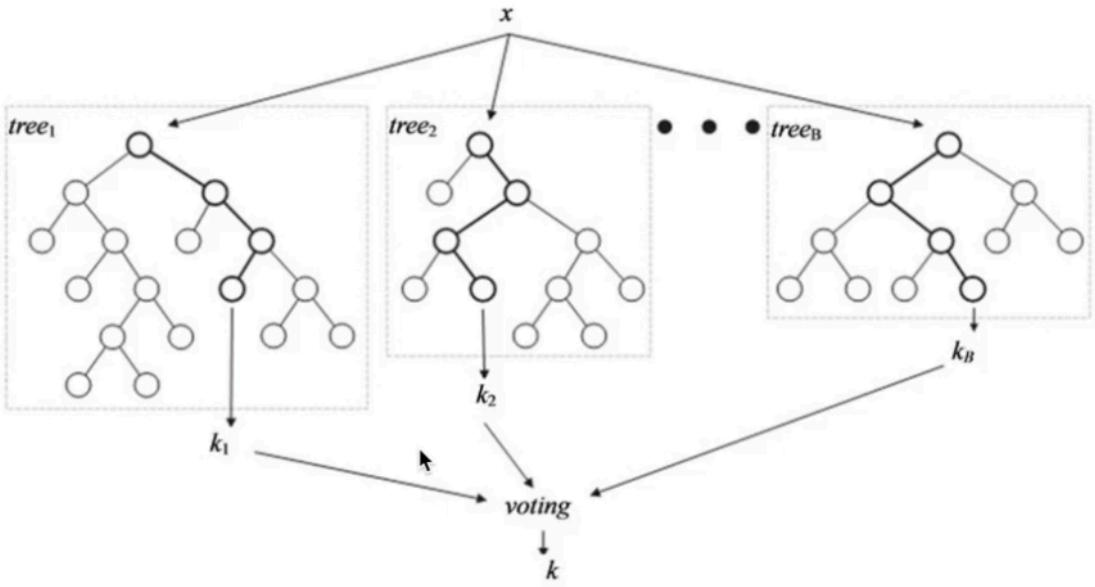
Şekil 11.1.5 AdaBoost örneğini göstermektedir. Bir veri setinde önce training yapılır sonra test edilir. Test edilenlerde yanlış tahmin edilenler bu sefer başka bir algoritma ile train edilir ve sonra test edilir. Sonra Burada yanlış olanlar başka bir algoritmada train edilir ve sonra test edilir şeklinde devam eder. Her seferinde sistemde başarılı olduğu örnekleri alır başarısız olunanlarla işlem devam eder.

Decision Forest (Karar Ormanları)

Farklı karar ağaçları oluşturur ve dallanmalar farklı oluşur yani bir soruyu çözmek için birden fazla ağaç üretilir.

Rassal Ağaç (Random Forest)

Rassal ağaç karar ormanlarına örnek verilebilir.

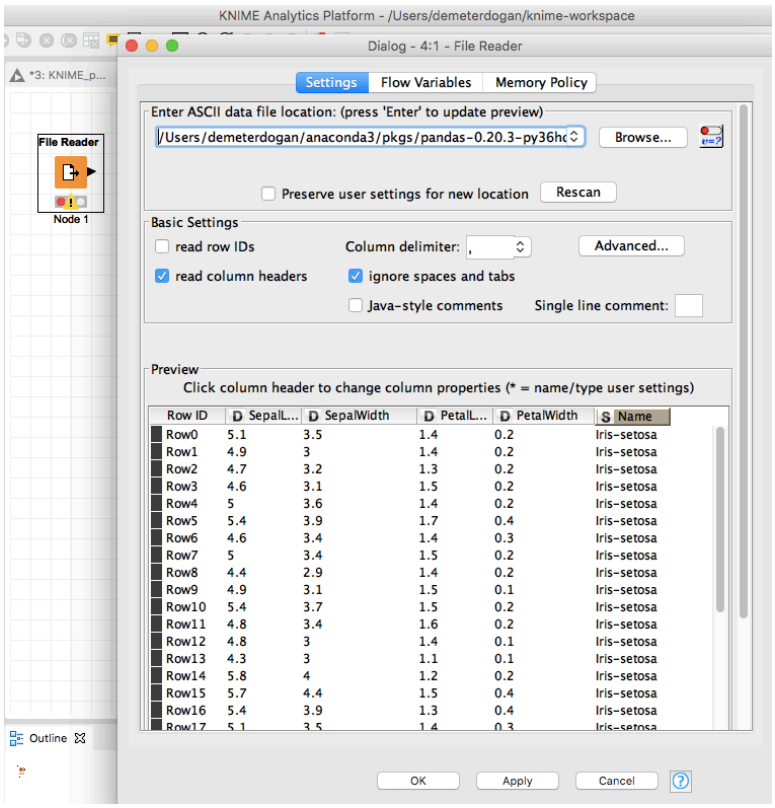


Şekil 11.1.6

Şekil 11.1.6'da bir aynı veri setinden birden fazla karar ağacı oluşturulmuştur. Hepsinden çıkan sonuç farklıdır. Örneğin birincisinden k1, ikincisinden k2 ve üçüncüsünden k3 çıkmıştır. Son olarak bunlar kendi aralarında oylanacaktır (voting). Tek bir algoritmaymış gibi hareket eder. Farklı algoritmaymış gibi işlem yapılmasına gerek yoktur.

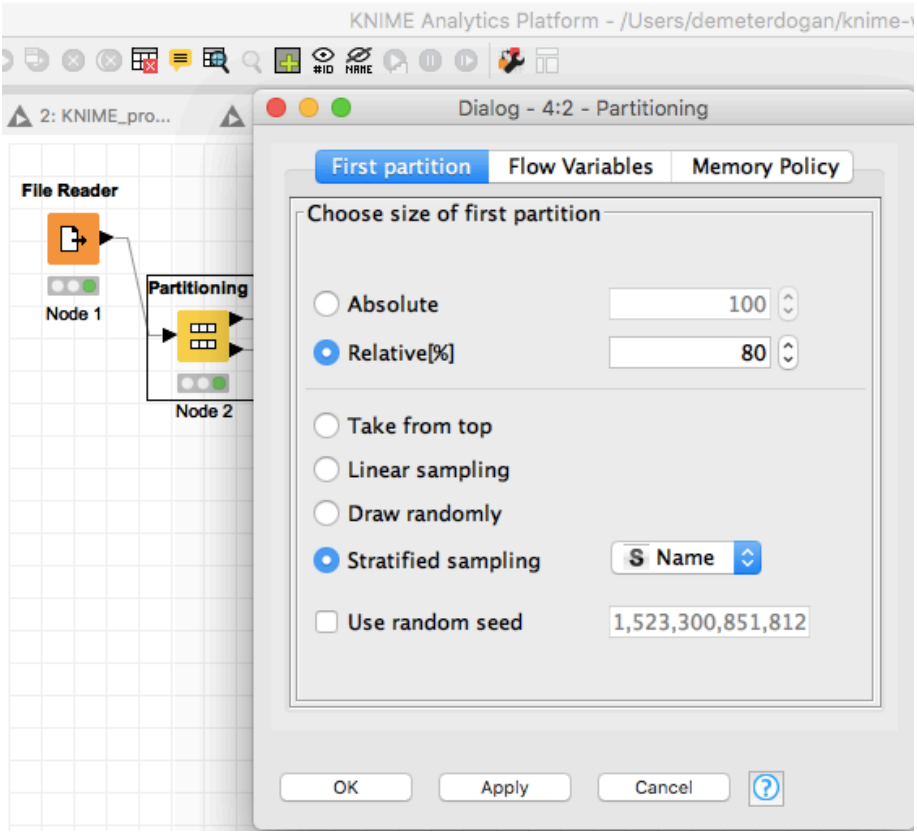
11.2 Örnek Üzerinden MAVL ve Prediction Fusion Uygulaması

Bu bölümde amaç, yukarıda teorik olarak açıklanan algoritmaların uygulamasını göstermektir.



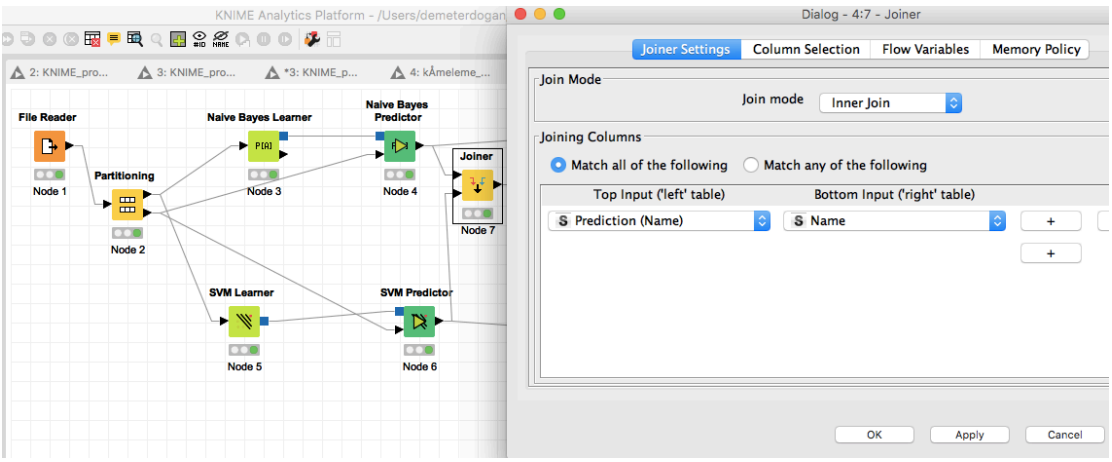
Şekil 11.2.1

Şekil 11.2.1, file reader operatörünün sisteme eklenmesini ve iris veri setinin tanıtılmasını göstermektedir. Anlaşıldığı üzere bu bölümde iris veri seti kullanılacaktır.



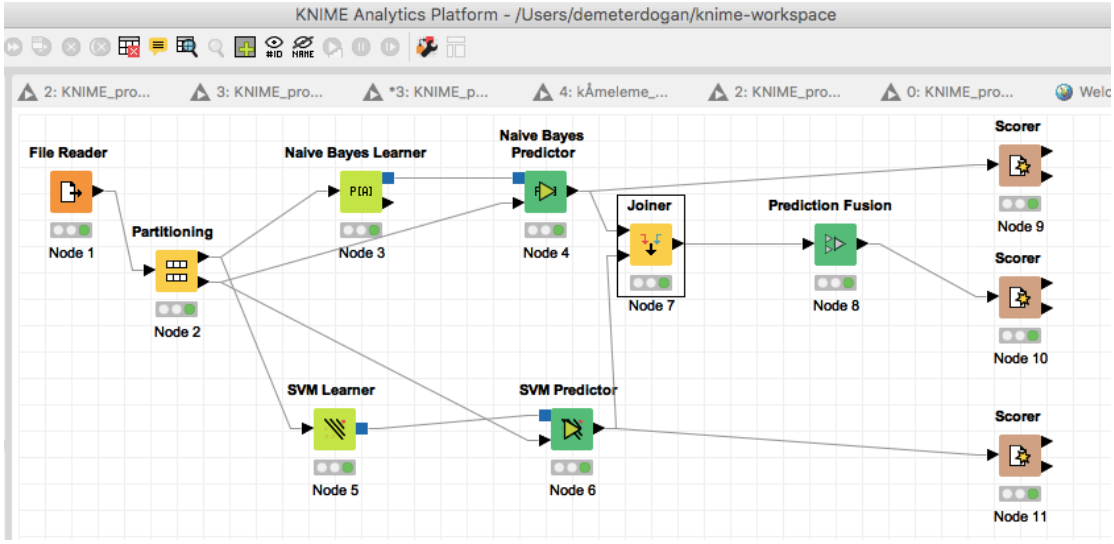
Şekil 11.2.2

Şekil 11.2.2, sisteme partitioning eklenmesiyle veri setinin 80% ve 20% olarak training ve test kümelerine ayrılması için configure penceresini göstermektedir.



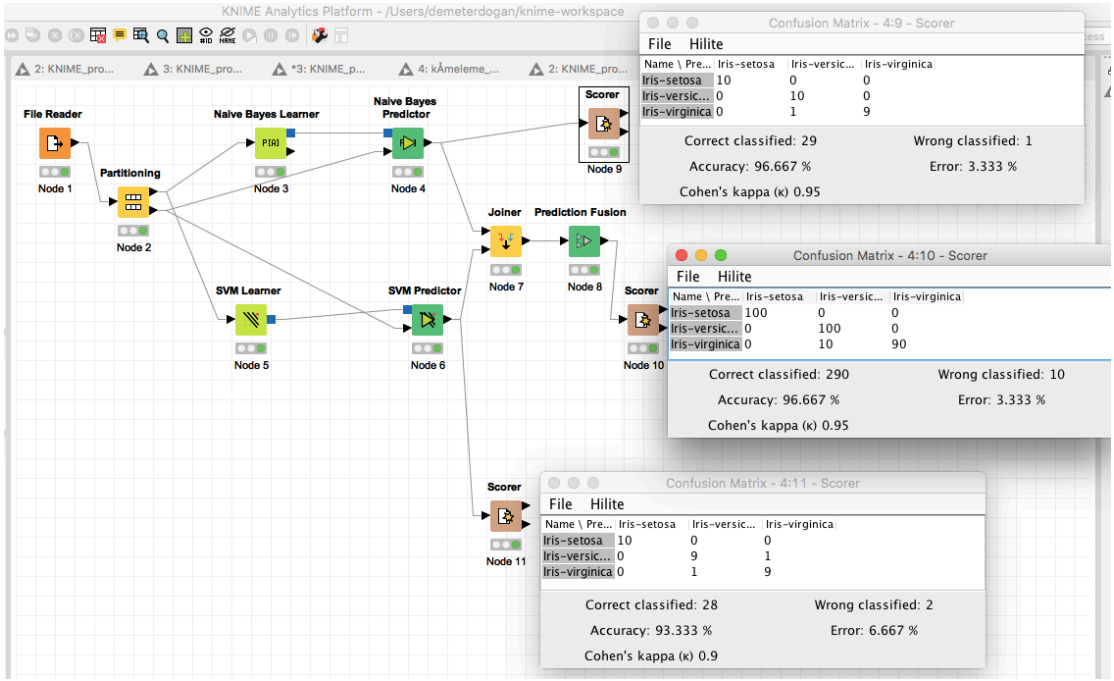
Şekil 11.2.3

Şekil 11.2.3, sisteme naive bayes learner ve predictor, SVM learner ve predictor, ve bunların birleştirilmesi için joiner operatörünün eklenmesini ve joiner operatörünün configure penceresini göstermektedir.



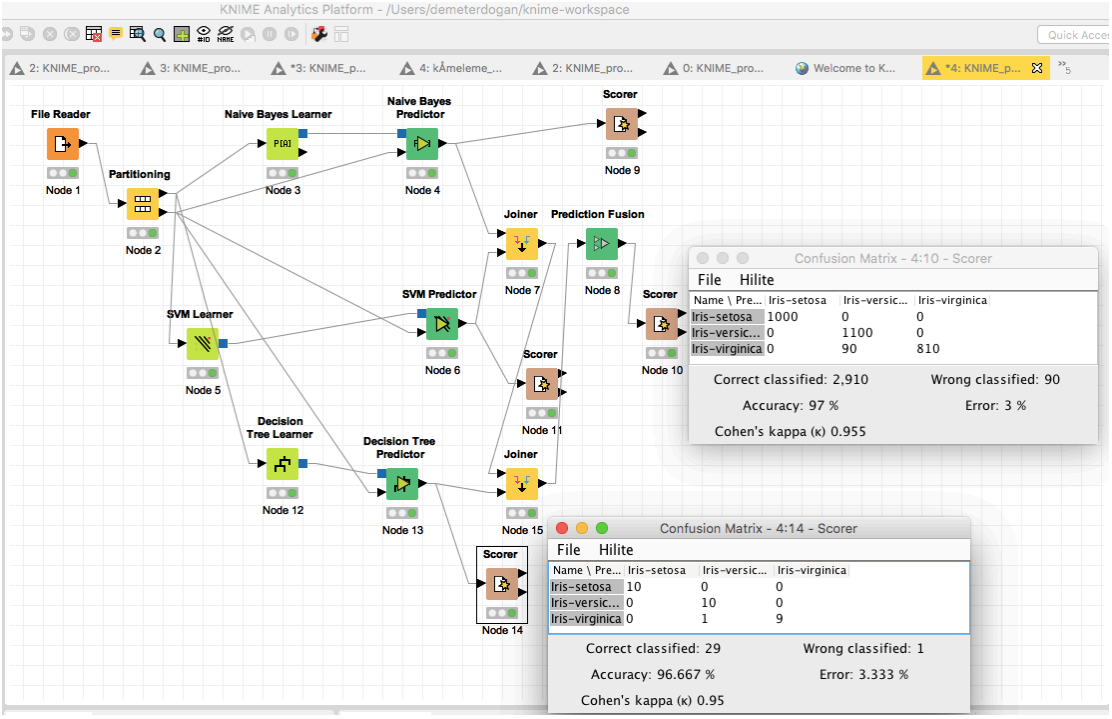
Şekil 11.2.4

Şekil 11.2.4, sisteme prediction fusion eklenmesini ve sonra sonuçların net görülebilmesi için hepsine ayrı ayrı scorer operatörlerinin bağlanmasını göstermektedir.



Şekil 11.2.5

Şekil 11.2.5, scorer'ların confusion matrix'lerini göstermektedir. Naive bayes'den gelen sonuç 96.67% ile, prediction fusion'dan gelen 96.67% ile, SVM'den gelenin ise 93.333% ile accuracy'e sahip olduğunu göstermektedir. Naive bayse 96.67% olduğu için prediction'da değerini ondan almaktadır. Bazen SVM ile birleşmesinden başarısı artabilir de.



Şekil 11.2.6

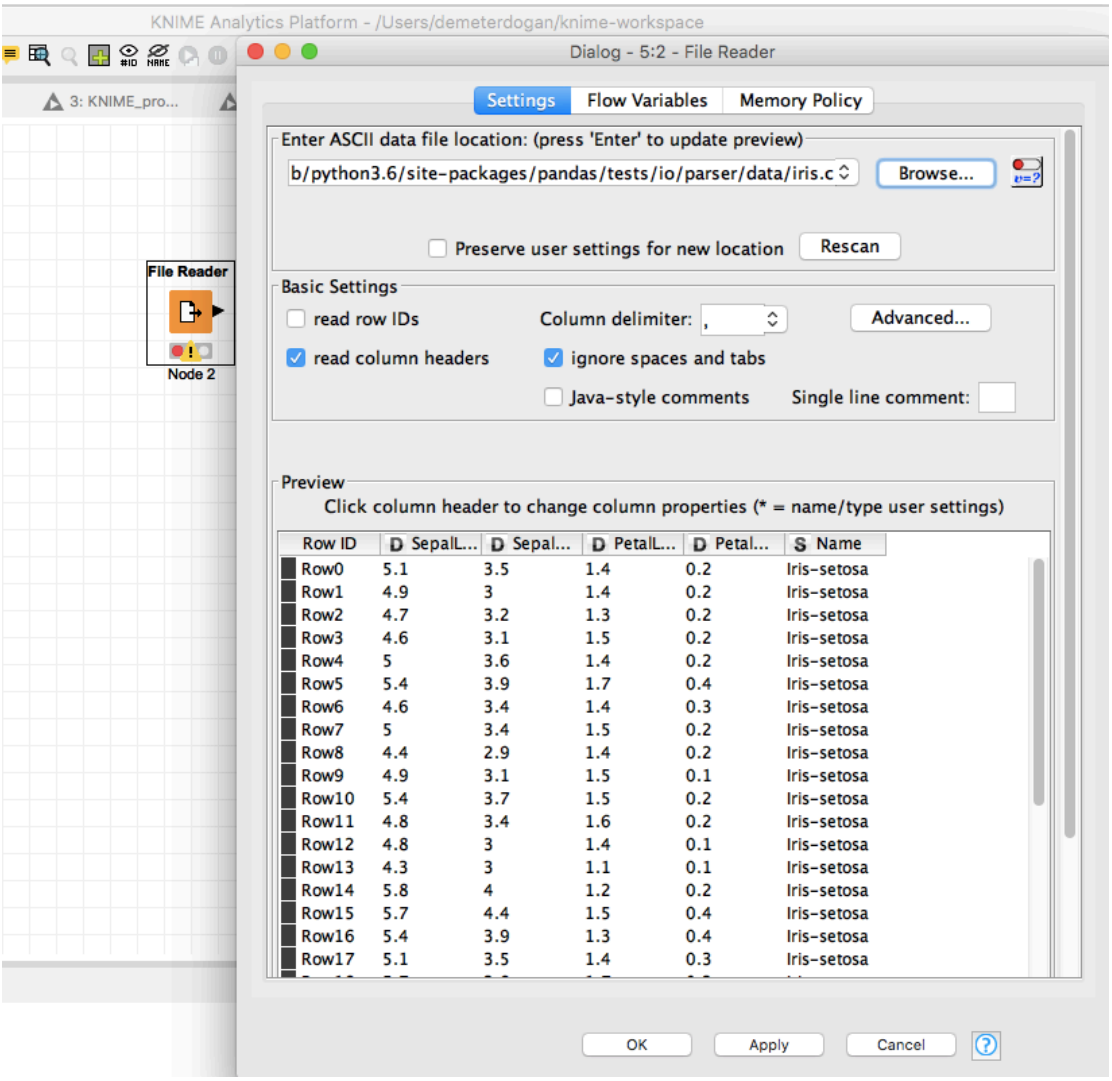
Şekil 11.2.6, sisteme yeni bir algoritma eklenmesini ve bununla birlikte oluşan değişimi göstermektedir. Decision tree learner, decision tree predictor, joiner ve scorer operatörlerinin bağlantıları şekilde görülmektedir. Bağlantılar yapıldıktan sonra, sadece decision tree accuracy oranı 96.67%, joiner ile birleştirildikten sonra ise 97% olmuş yani başarısı artmıştır.

Bu bölümde algoritmaların kullanımı gösterilmiştir.

11.3 Random Forest (Rassal Orman) Yöntemi ile Sınıflandırma ve Tahmin

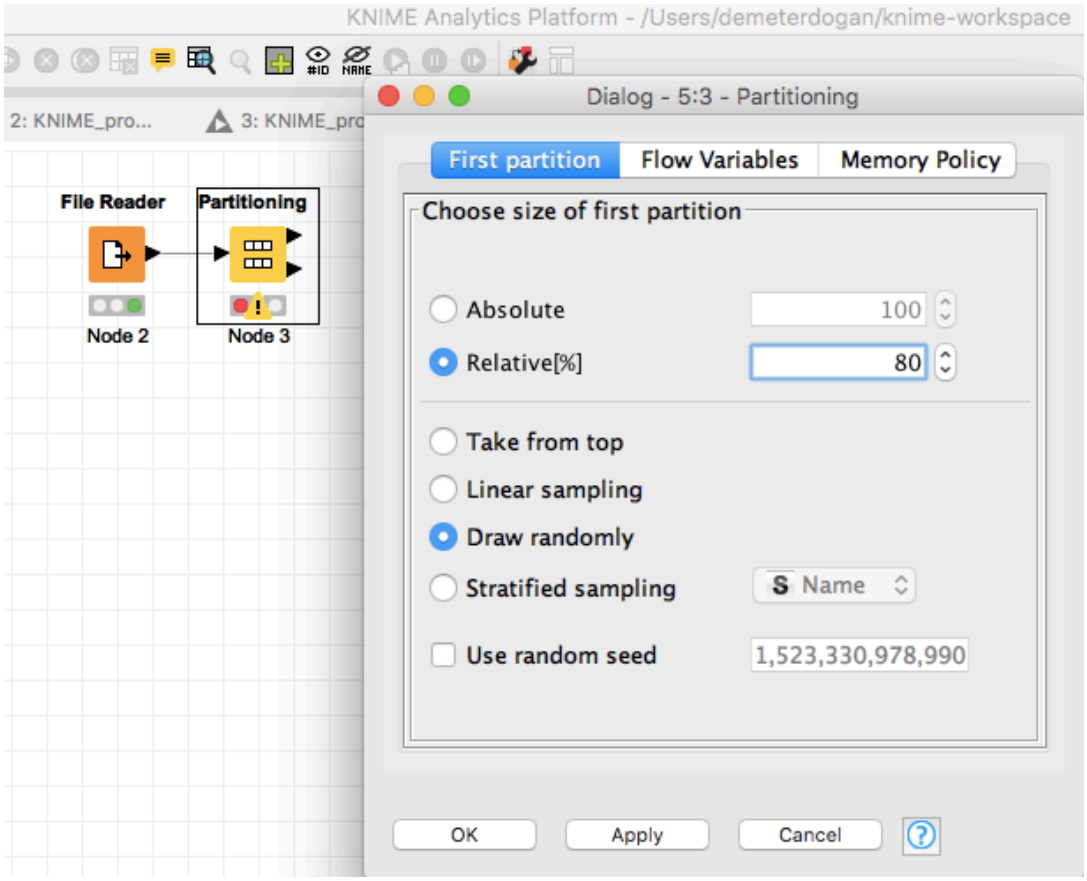
Bu bölümde amaç, 1. bölümde teorik olarak açıklanan random forest ‘ın örnek üzerinden göstermektir.

Bu bölümde iris veri seti kullanılacaktır.



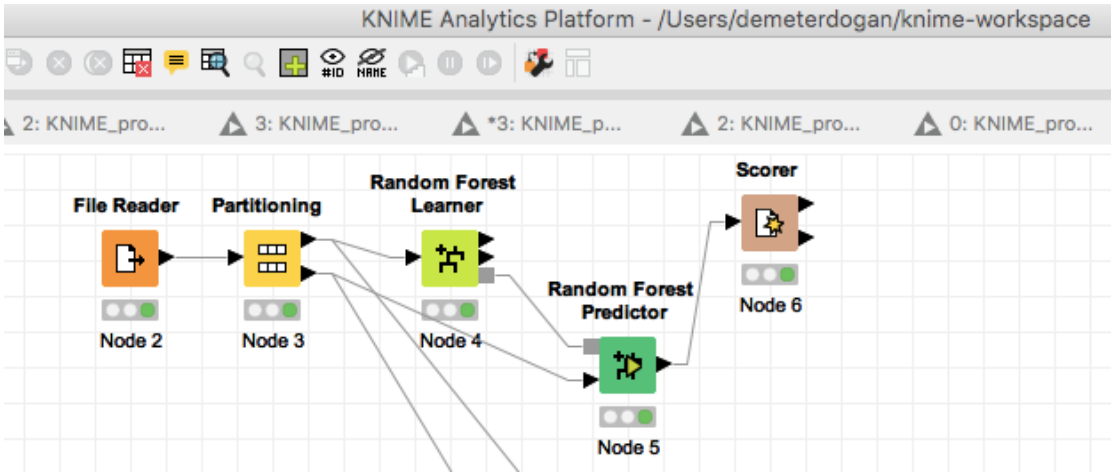
Şekil 11.3.1

Şekil 11.3.1, file reader operatörünün sisteme eklenmesini ve iris veri setinin configure bölümünden browse edilmesini göstermektedir. İris veri kümesi eşhur bir veri seti olmakla birlikte internetten kolayca bulunabilir.



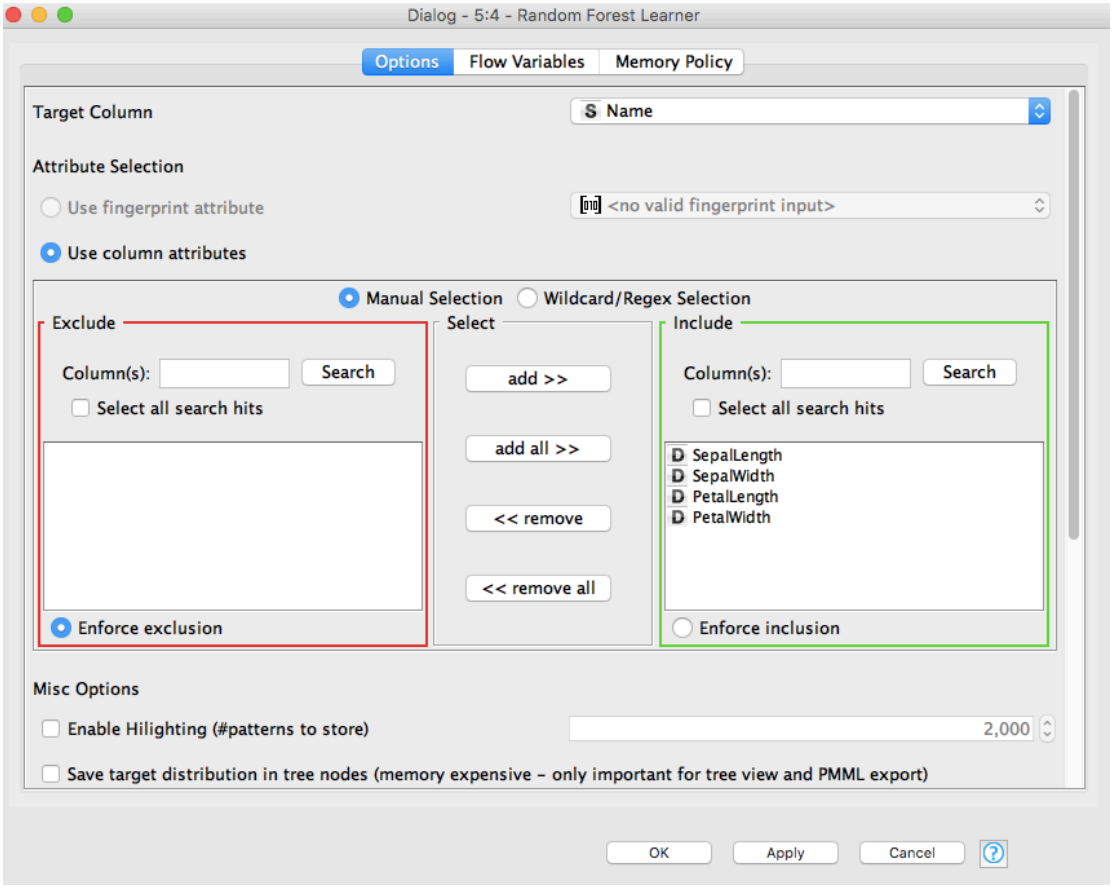
Şekil 11.3.2

Şekil 11.3.2, sisteme partitioning operatörünün eklenmesini ve configure penceresini göstermektedir. Veri set 80% ve 20% olarak ikiye bölünmüştür.



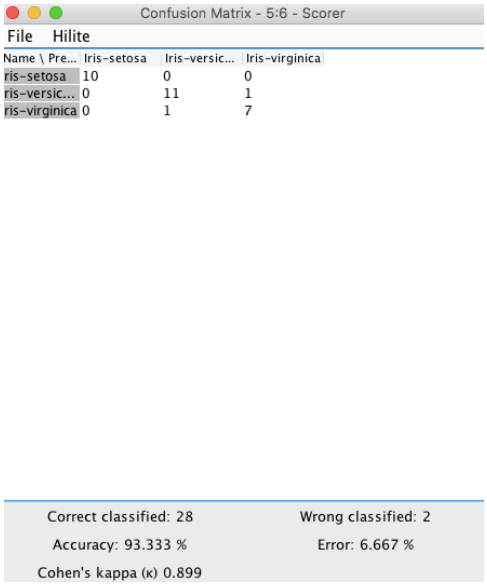
Şekil 11.3.3

Şekil 11.3.3, sisteme random forest learner, random forest predictor ve scorer eklenmesini ve aralarındaki bağlantıyı göstermektedir.



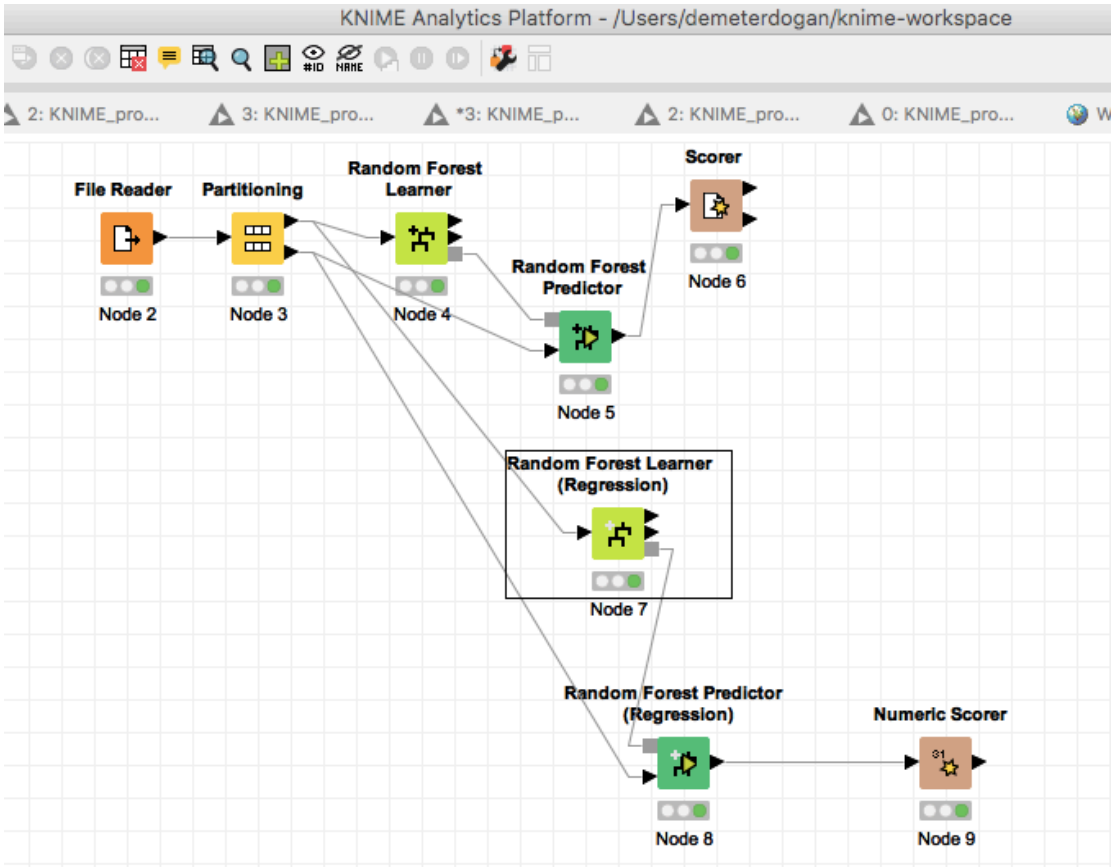
Şekil 11.3.4

Şekil 11.3.4, random forest learner'ın configure penceresini göstermektedir. Burada değişiklik yapılmamıştır ama yine de programı çalıştırabilmek için bir kez bile olsun configure penceresi açılmalıdır.



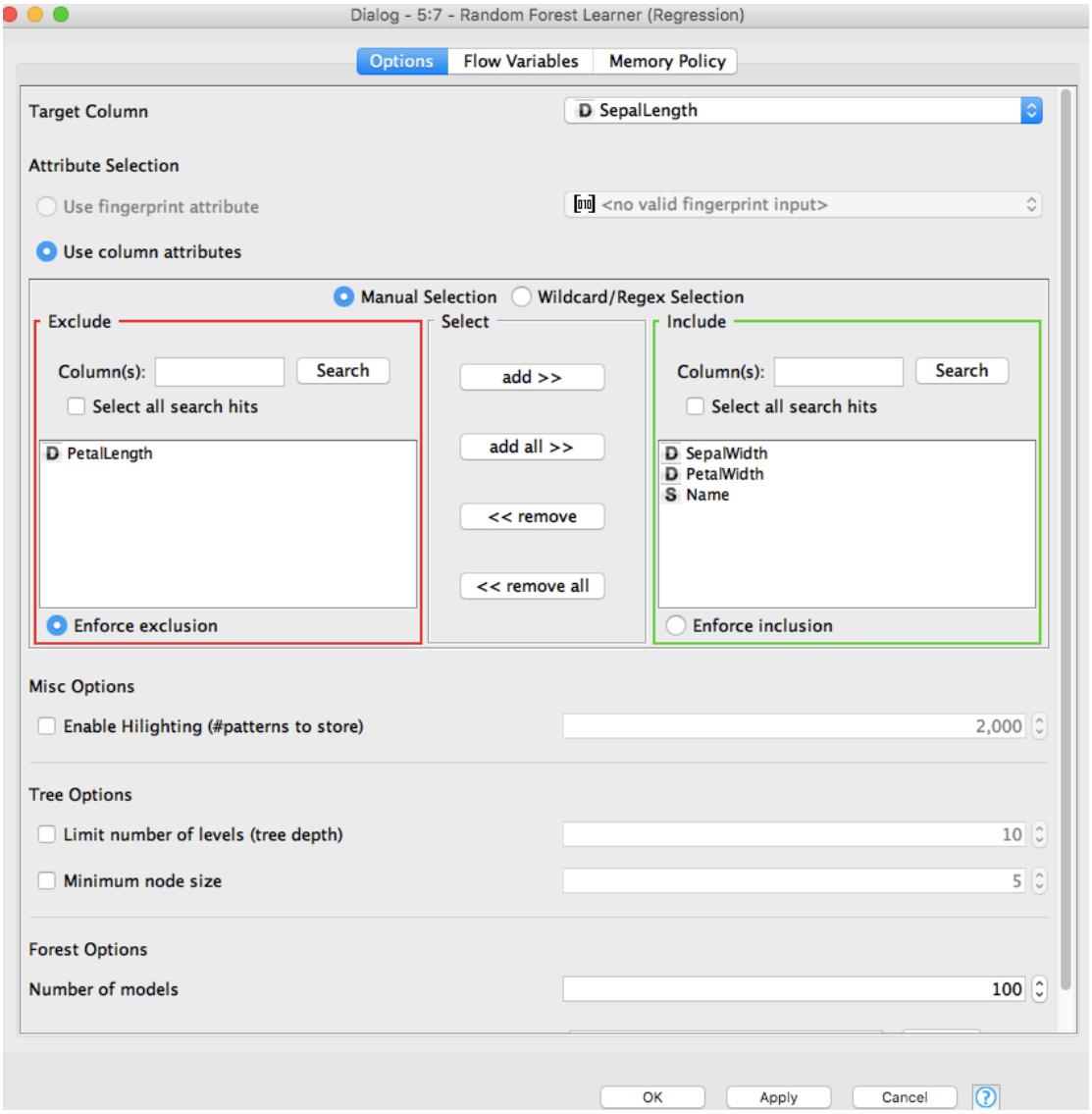
Şekil 11.3.5

Şekil 11.3.5, program çalıştırıldıktan sonra elde edilen confusion matrix'i göstermektedir. Accuracy 93.3% gelmiştir.



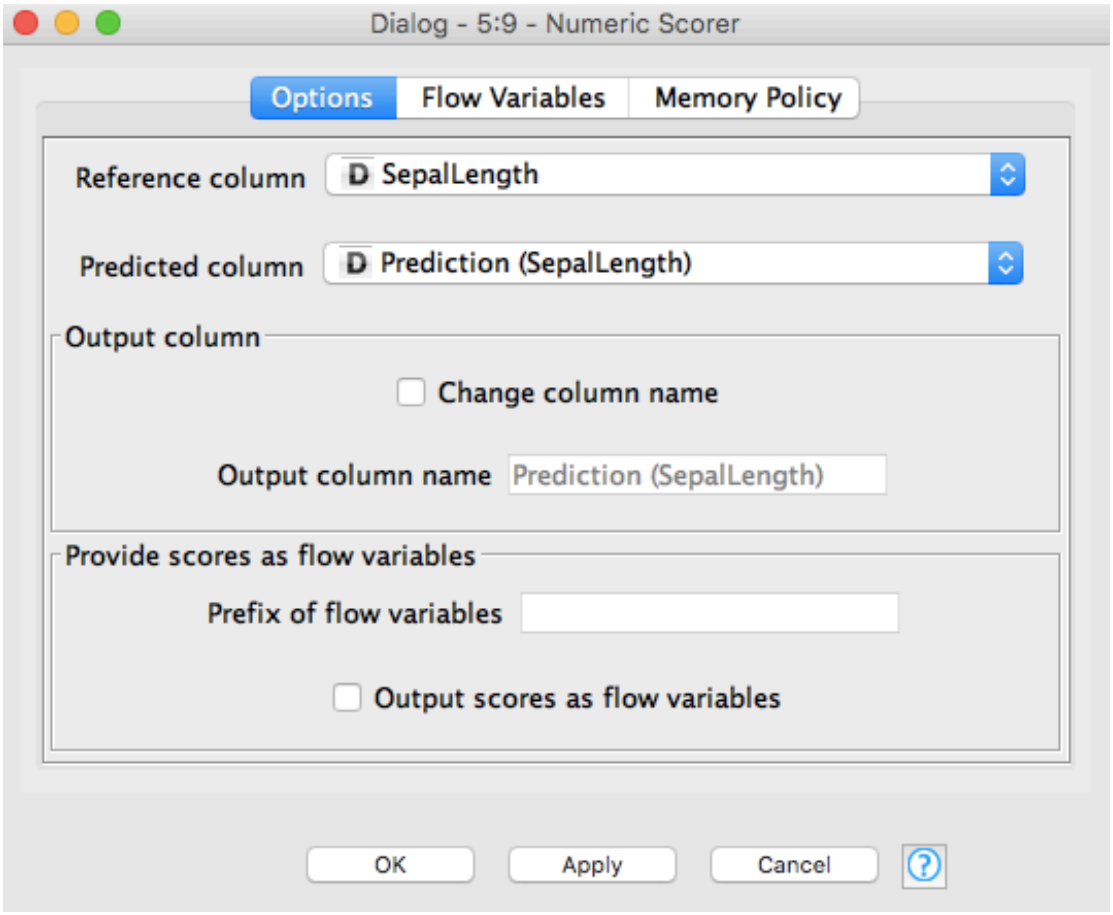
Şekil 11.3.6

Şekil 11.3.6, sisteme random forest learner regresssiion predictori, random forest predictor (regression) ve numaric scorer eklnmesimi, diğerleriyle bağlantıyı göstermektedir.



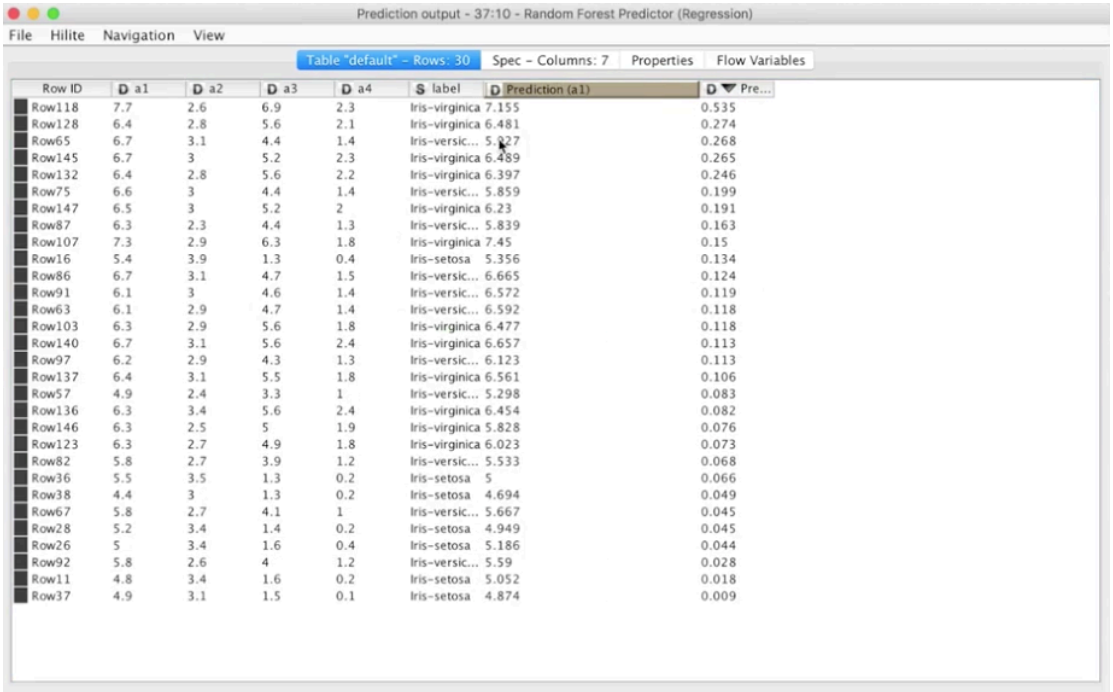
Şekil 11.3.6

Şekil 11.3.6, random forest Learner (regression) configure penceresini göstermektedir. Bu sefer yukarıdakinden fark, tahmin edilmek istenen değer, diğer yaprak özelliklerini bakılarak başka bir yaprak uzunluğu/genişliğini tahmin etmektir.



Şekil 11.3.7

Şekil 11.3.7, Numeric score configure penceresini göstermektedir.



Şekil 11.3.8

Şekil 11.3.8 random forest learner (regression)'ın prediction outputunu göstermektedir. Örneğin yaprağın uzunluğu (a1 kolonu) 7.7 iken makine 7.15 tahmin etmiştir.

12. BÖLÜM: UÇTAN UCA GERÇEK HAYAT ÖRNEKLERİ

12.1. İş İlanları, Web Siteleri, Kaynaklar, Yarışmalar ve Örnek Veri Kümesi

Bu bölümde gerçek bir uygulamaya giriş gösterilecek ve bazı önemli kaynaklar tanıtılacaktır.

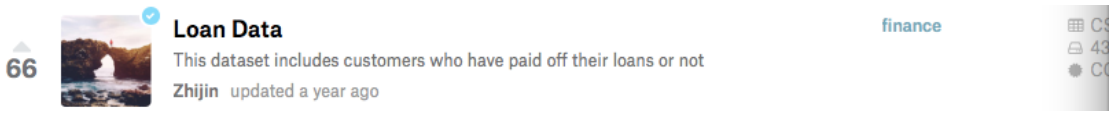
Kaggle önemli kaynaklardan biridir. Competitions yani yarışmalara katılarak, oraya yüklenmiş veri setlerini indirip kullanarak ya da oraya veri seti oluşturup yükleyerek, eğitim (tutorial) videoları izleyerek ve iş ilanlarına başvurarak kaggle'dan faydalanılabilir.

Kdnuggets, mldata.org gibi kaynaklardan da faydalanılabilinir.

Bu bölümde ve sonraki bölümlerde veri kaynağı olarak kullanılacak yer kaggle'dır.

Kaggle'a giridkten sonra sırasıyla;

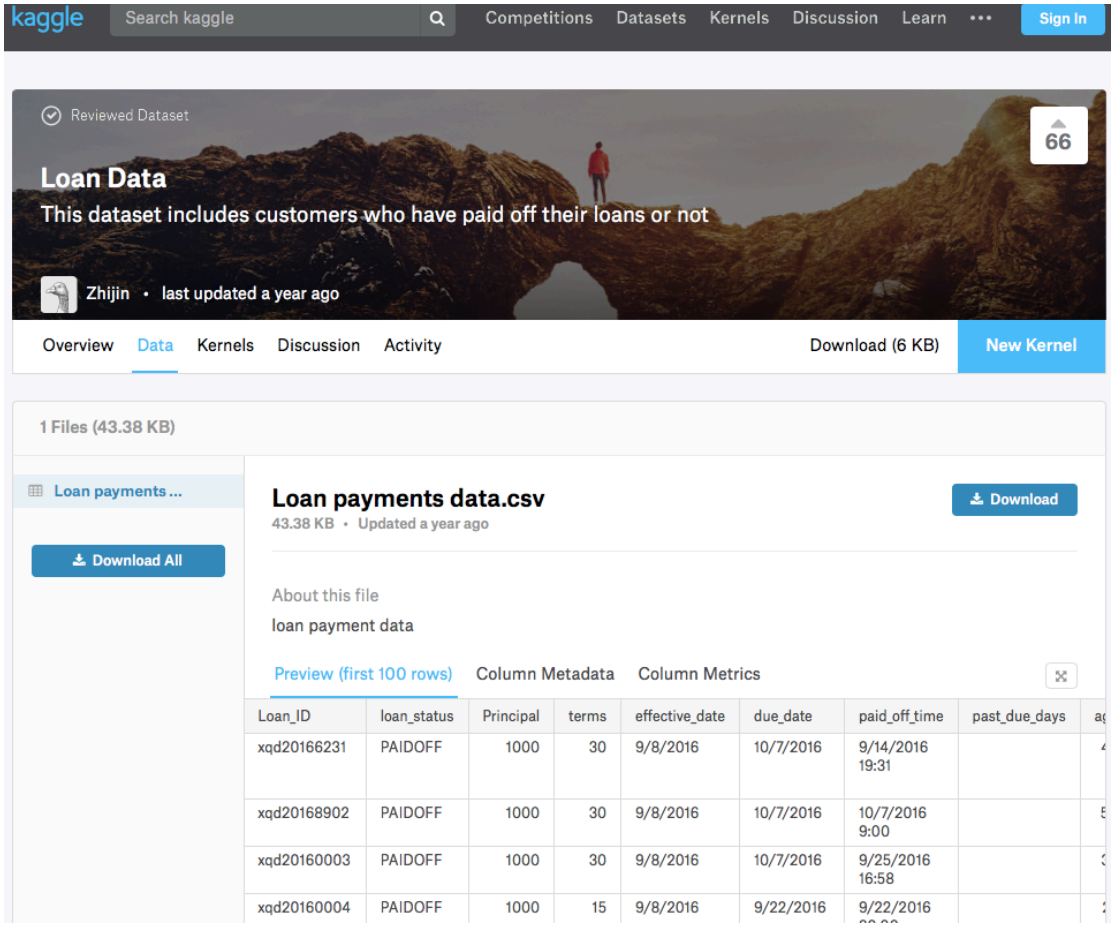
1. Datasets
2. Arama butonuna loan payment yazılarak veri seti aratılmalı
3. Loan data isimli çıkan projeye girilmeli



Şekil 12.1.1

Şekil 12.1.1 girilmesi gereken projenin görselini göstermektedir.

4. Top contributors projeye katılanları , kernels kimlerin neler denediği ve discussion da tartışma platformunu göstermektedir.
5. Data kısmına girilerek download butonuna basılıp veri seti indirilmelidir.



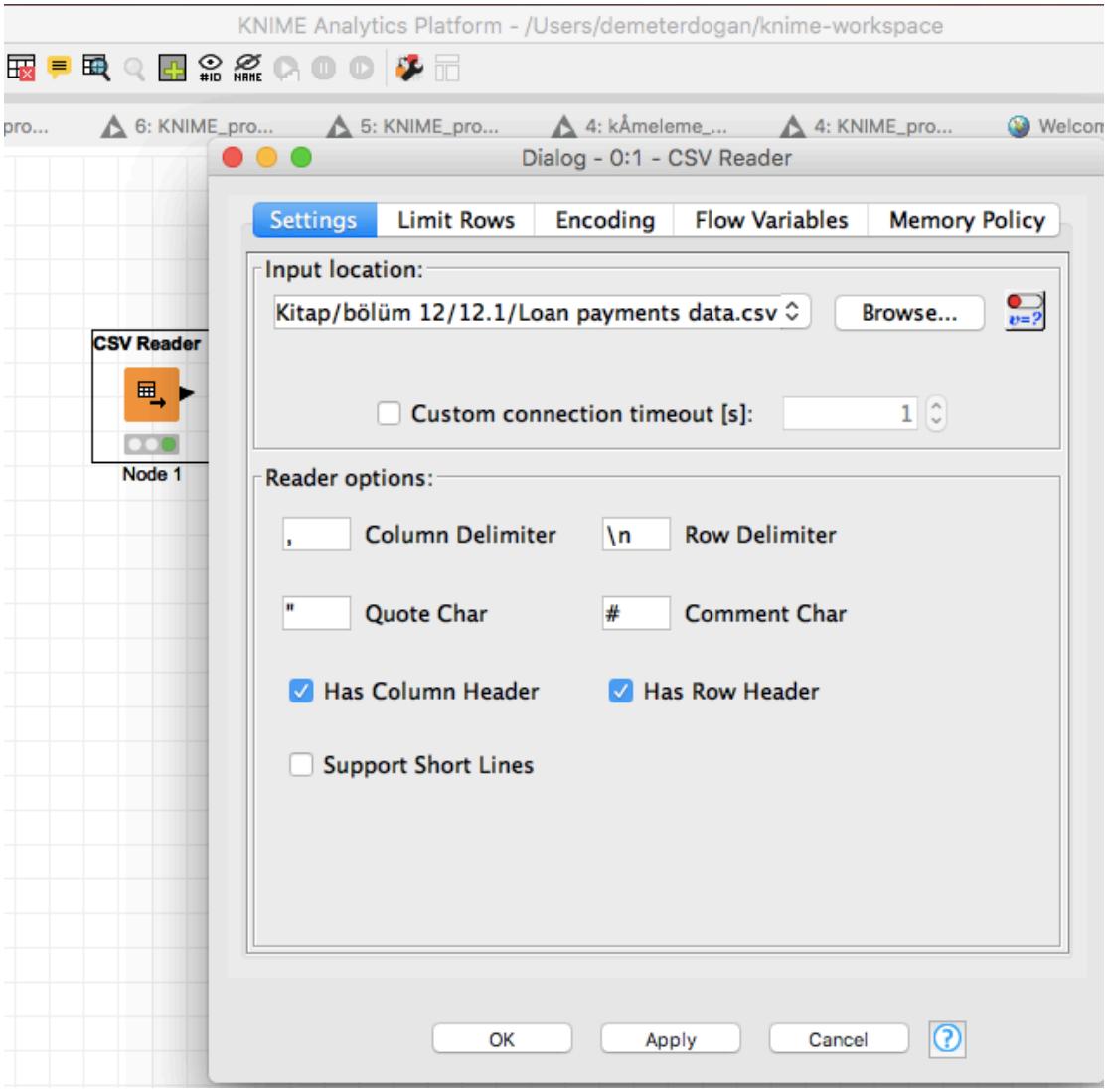
Şekil 12.1.2

Şekil 12.1.2 loan data projesinin data kısmını ve download buton yerini göstermektedir. Ayrıca veri setinin de kolon isimleri ve veriden örnek bir kaç sıra görülmektedir. Bu veri seti ilerideki bölümlerde kullanılacaktır.

Bu bölümde faydalı olabilecek birkaç kaynak ve kaggle'dan örnek veri seti indirilmesinden bahsedilmiştir.

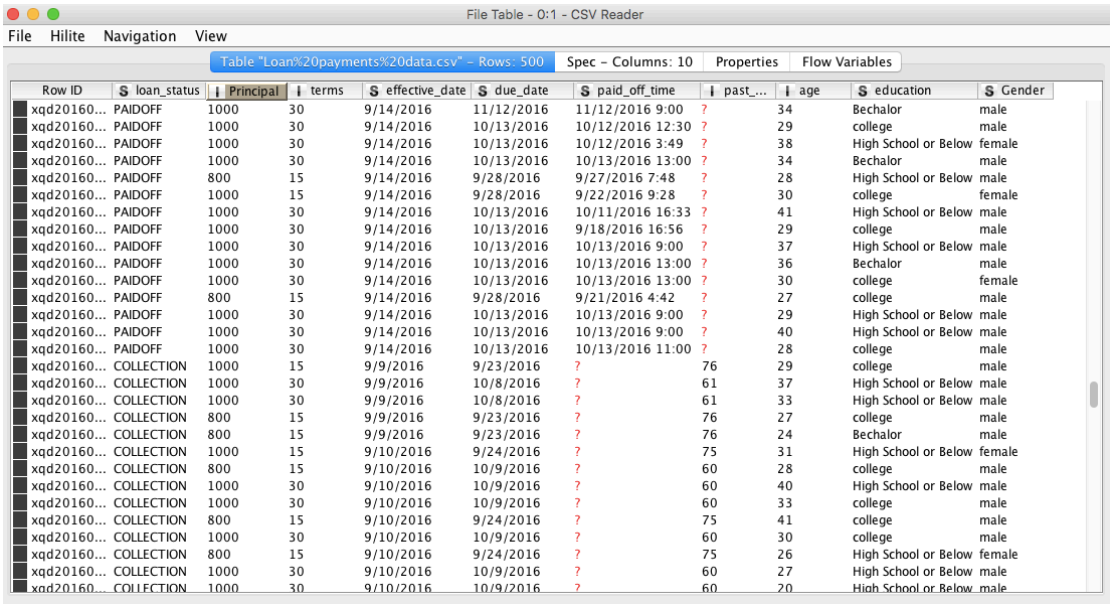
12.2. Müşterinin Borcunu Ödeyip Ödemeyeceğinin Tahmini

Bu bölümde, bir önceki bölümde indirilen veri kümesi ile müşterimin borcunun ödeyip ödemeyeceğinin tahmininin yapılabilmesi gösterilecektir.



Şekil 12.2.1

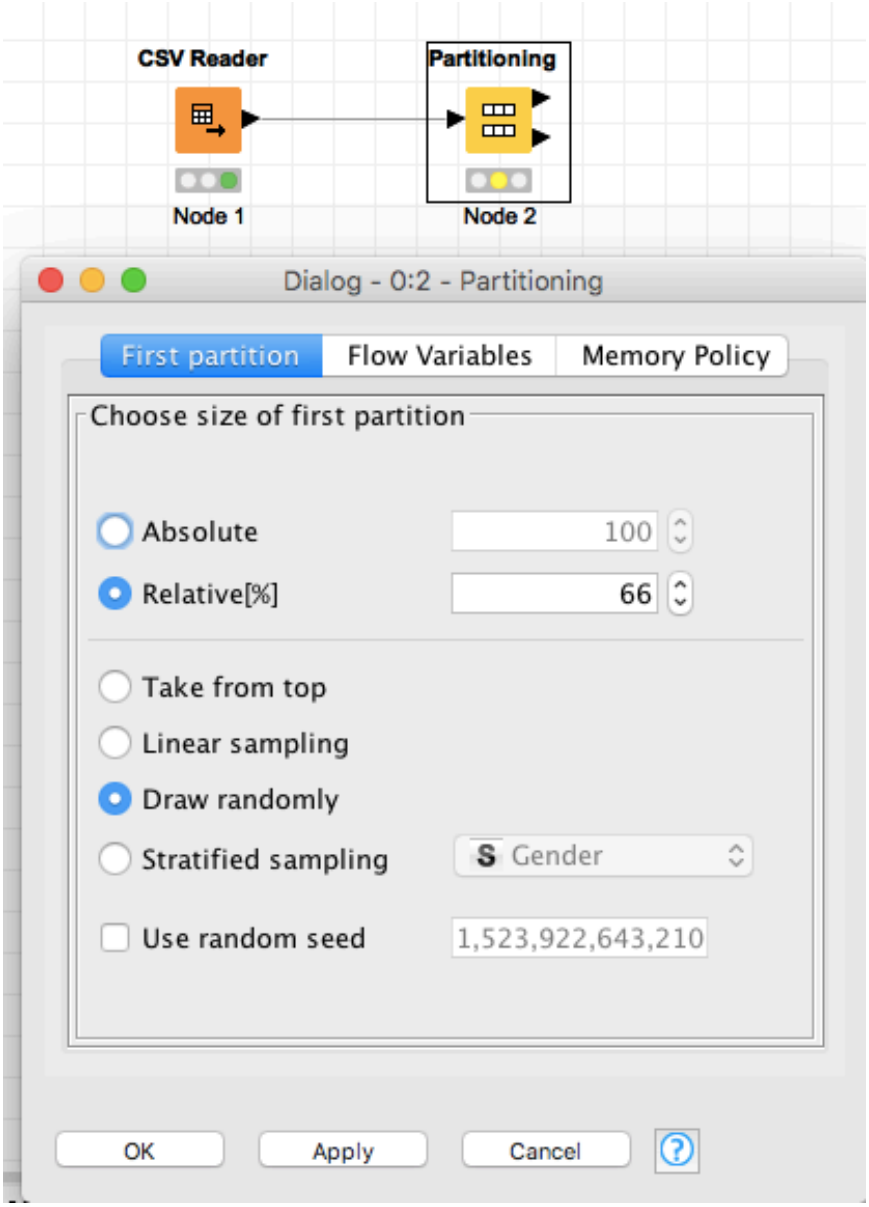
Şekil 12.2.1, sisiteme csv reader operatörünün eklenmesini ve bir önceki bölümde kaggle'dan indirilen loan payments csv dosyasının sisteme bu operatör ile eklenmesini göstermektedir.



Row ID	loan_status	Principal	terms	effective_date	due_date	paid_off_time	past_...	age	education	Gender
xqd20160...	PAIDOFF	1000	30	9/14/2016	11/12/2016	11/12/2016 9:00	?	34	Bechalor	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/12/2016 12:30	?	29	college	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/12/2016 3:49	?	38	High School or Below	female
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 13:00	?	34	Bechalor	male
xqd20160...	PAIDOFF	800	15	9/14/2016	9/28/2016	9/27/2016 7:48	?	28	High School or Below	male
xqd20160...	PAIDOFF	1000	15	9/14/2016	9/28/2016	9/22/2016 9:28	?	30	college	female
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/11/2016 16:33	?	41	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	9/18/2016 16:56	?	29	college	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 9:00	?	37	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 13:00	?	36	Bechalor	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 13:00	?	30	college	female
xqd20160...	PAIDOFF	800	15	9/14/2016	9/28/2016	9/21/2016 4:42	?	27	college	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 9:00	?	29	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 9:00	?	40	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/2016	10/13/2016 11:00	?	28	college	male
xqd20160...	COLLECTION	1000	15	9/9/2016	9/23/2016	?	76	29	college	male
xqd20160...	COLLECTION	1000	30	9/9/2016	10/8/2016	?	61	37	High School or Below	male
xqd20160...	COLLECTION	1000	30	9/9/2016	10/8/2016	?	61	33	High School or Below	male
xqd20160...	COLLECTION	800	15	9/9/2016	9/23/2016	?	76	27	college	male
xqd20160...	COLLECTION	800	15	9/9/2016	9/23/2016	?	76	24	Bechalor	male
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	?	75	31	High School or Below	female
xqd20160...	COLLECTION	800	15	9/10/2016	10/9/2016	?	60	28	college	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	40	High School or Below	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	33	college	male
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	?	75	41	college	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	30	college	male
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	?	75	26	High School or Below	female
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	27	High School or Below	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	20	High School or Below	male

Şekil 12.2.2

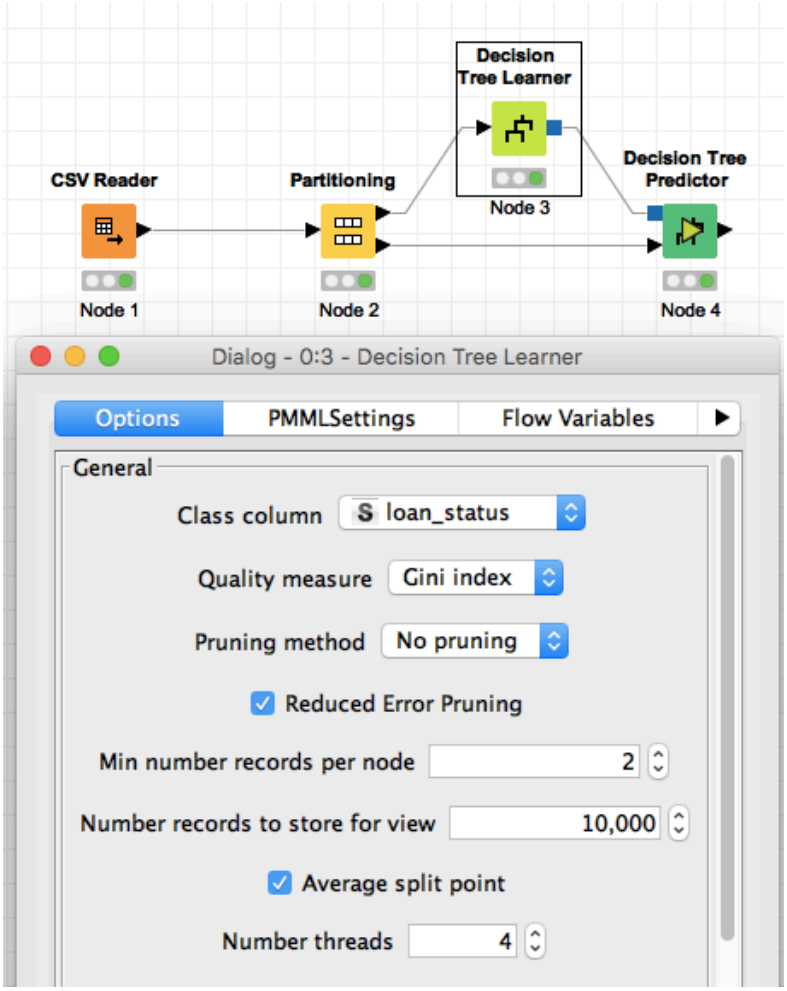
Şekil 12.2.2, loan payments veri setinin içeriğini göstermektedir. Loan_status ile borcunu ödenip ödenmediğini gösteren kolon bulunmaktadır. Paidoff ödedi, collection ödemedi ve collection_paidoff gecikmeli ödeme anlamına gelmektedir. Principal anapara, dua_date ödeme tarihi, kişinin yaşı, eğitim durumu ve cinsiyeti vb. bilgileri veren kolonlar da bulunmaktadır.



Şekil 12.2.3

Sınıflandırma algoritması kullanılacağından dolayı veriyi öğrenme ve test için bölebilmek için validation yöntemlerinden biri olan split validation kullanılacaktır. Şekil 12.2.3, bunun için sisteme partitioning operatörünün eklenmesini ve configure penceresinde yapılan değişikliği göstermektedir. Veri seti 66% ve 34% olarak bölünmektedir.

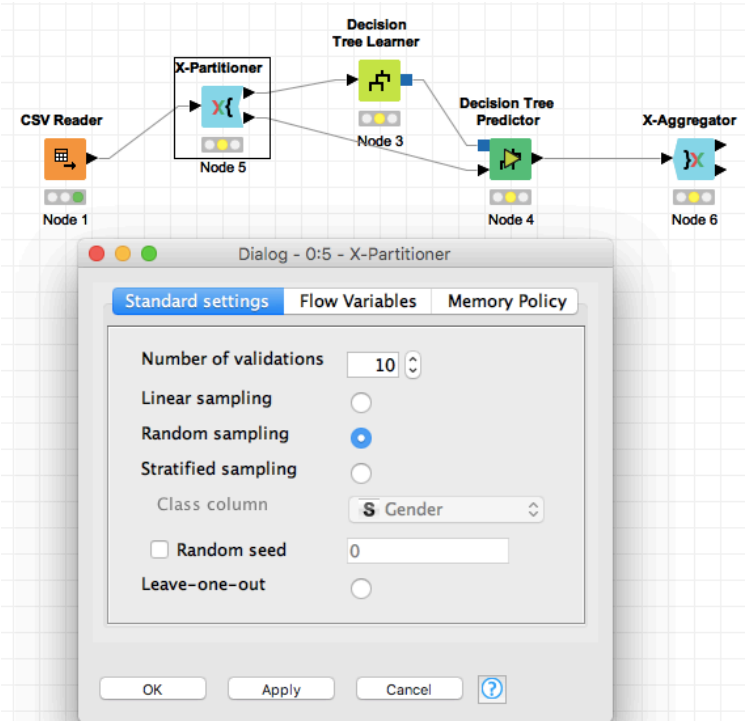
Bu validation'ın sorunu bazı veriler sadece test bazıları sadece training için kullanılacaktır.



Şekil 12.2.4

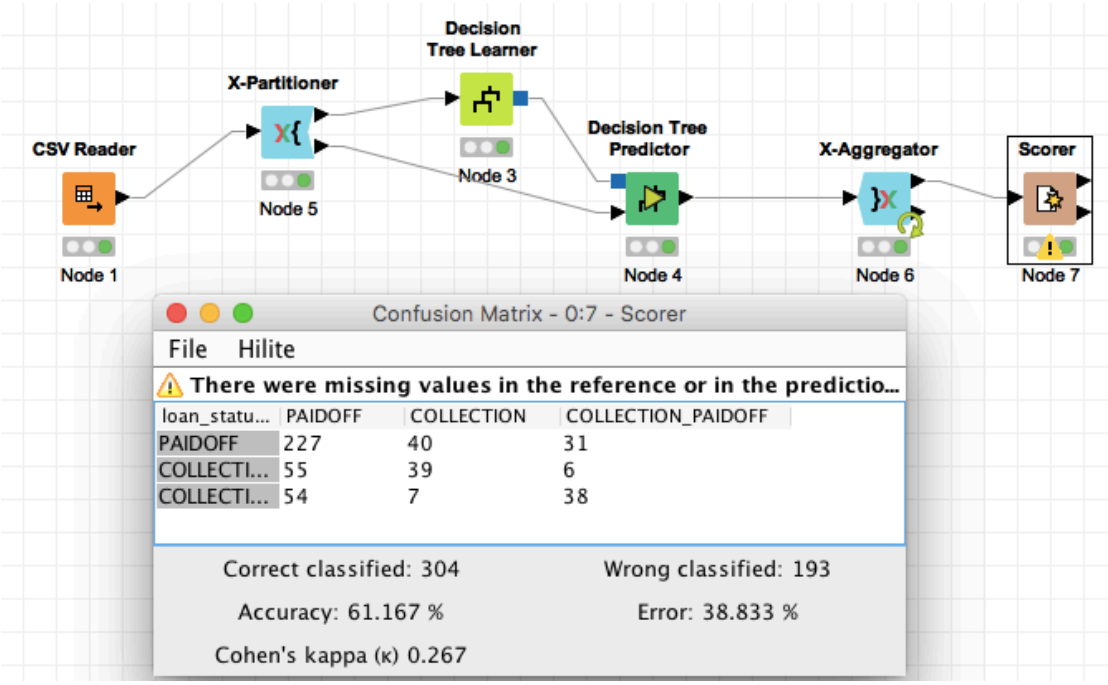
Şekil 12.2.4, sisteme decision tree learner, decision tree predictor operatörlerinin eklenmesini ve bağlantılarını ayrıca decision tree learner configure penceresinde yapılan değişikliği göstermektedir. Class column seçeneği tahmin edilmek istenilen kolonu verir. Bu örenekte loan_statüs predict edilecek kolon olarak seçildi.

Fakat yukarıda bahsedilen partitioning'in veri setindeki tüm verileri training ve test için kullanamadığından dolayı o operatör değiştirilecektir.



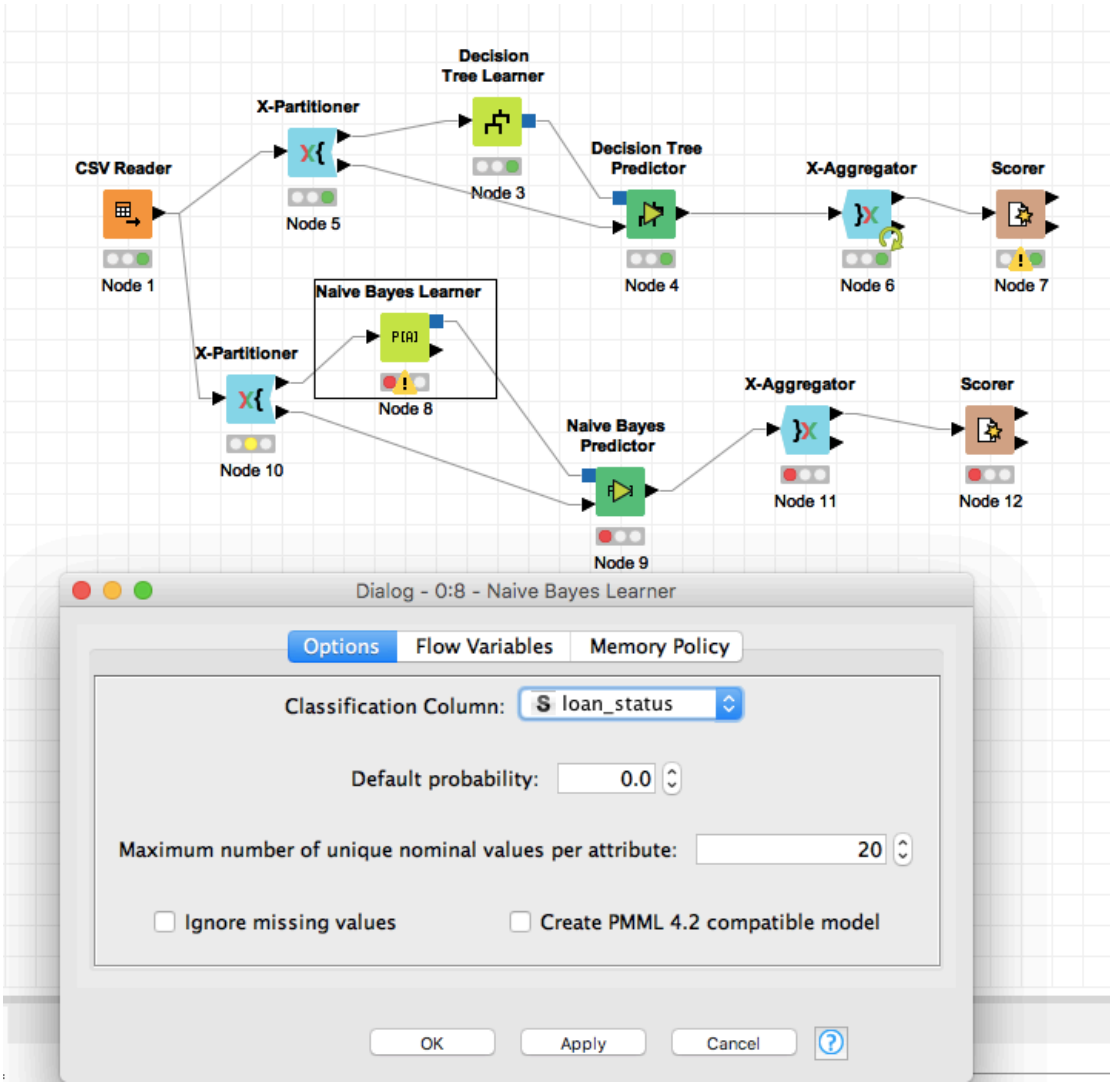
Şekil 12.2.5

Şekil 12.2.5, partitioning yerine x-partitioner operatörünün konulmasını ve configure penceresinde yapılan değişikliği göstermektedir. Tüm verilerin hem testte hem öğrenme aşamasında kullanılabilmesi için bu validation yöntemi tercih edilebilir. Veri seti 10 parçaya bölünmüş ve tüm parçaları kullanılacaktır sonunda da bu parçaların birleştirilebilmesi için x-aggregator operatörü eklenmiştir.



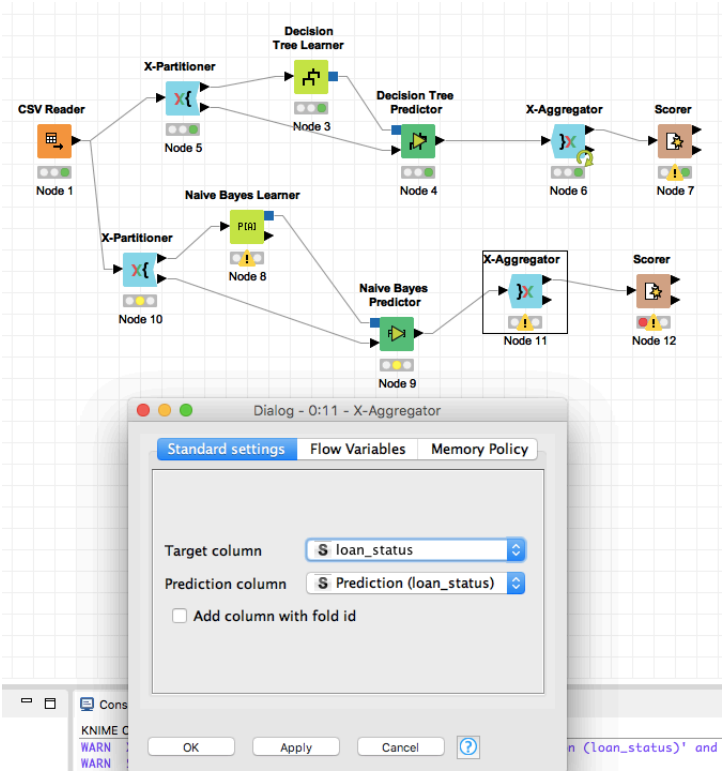
Şekil 12.2.7

Şekil 12.2.7, sonucu görebilmek ve değerlendirebilmek açısından sisteme her zamanki gibi scorer operatörünün eklenmesini ve çalıştırıldıktan sonra oluşan confusion matrix'i göstermektedir. Accuracy görüldüğü gibi 61%'dir. Diagonal'a bakıldığında, paidoff olup sistemin paidoff tahmini 227 kişi, collection olup sistemin collection tahmini 39 kişi ve collection_paidoff olup sistemin collection_paidoff tahmini 38 kişidir.



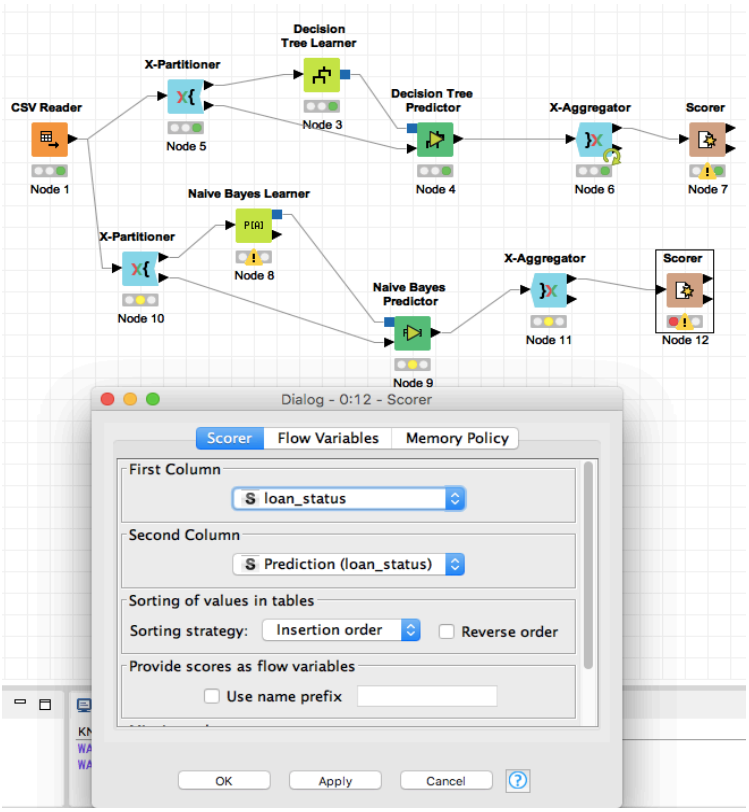
Şekil 12.2.8

Şekil 12.2.8, başka örnek bir operatör denemesini göstermektedir. Decision tree yerine naive bayes kullanılmıştır. Yine tahmin edilmek istenilen kolon loan_status'dür.



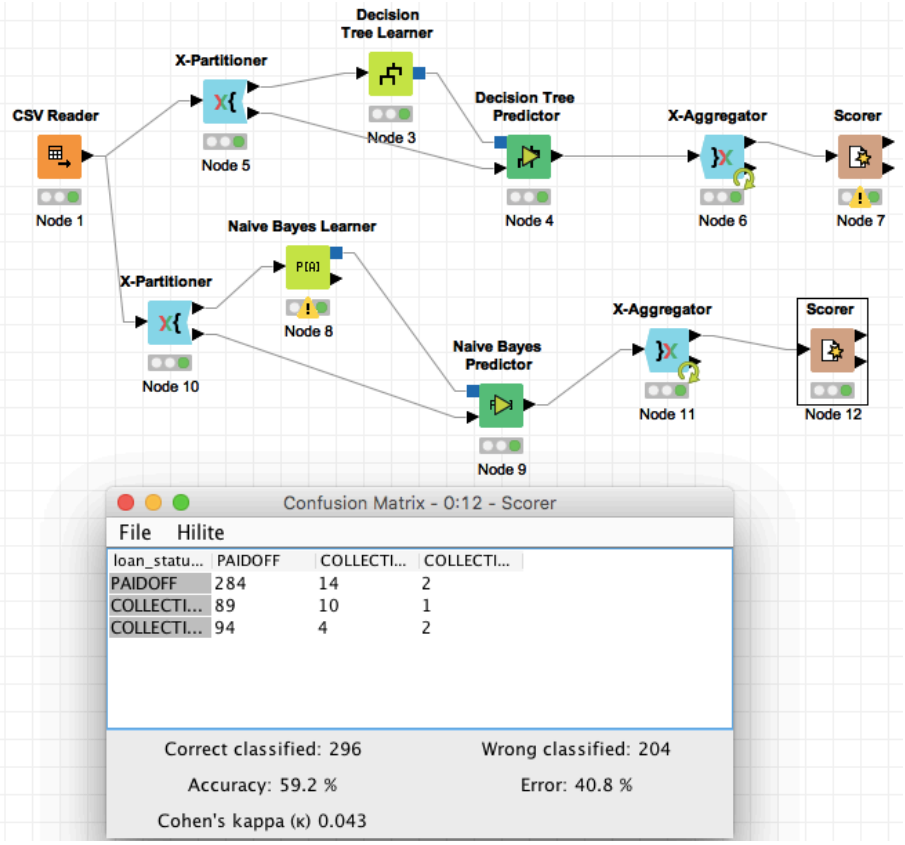
Şekil 12.2.9

Şekil 12.2.9, x-aggregator operatörünün configure penceresini göstermektedir.



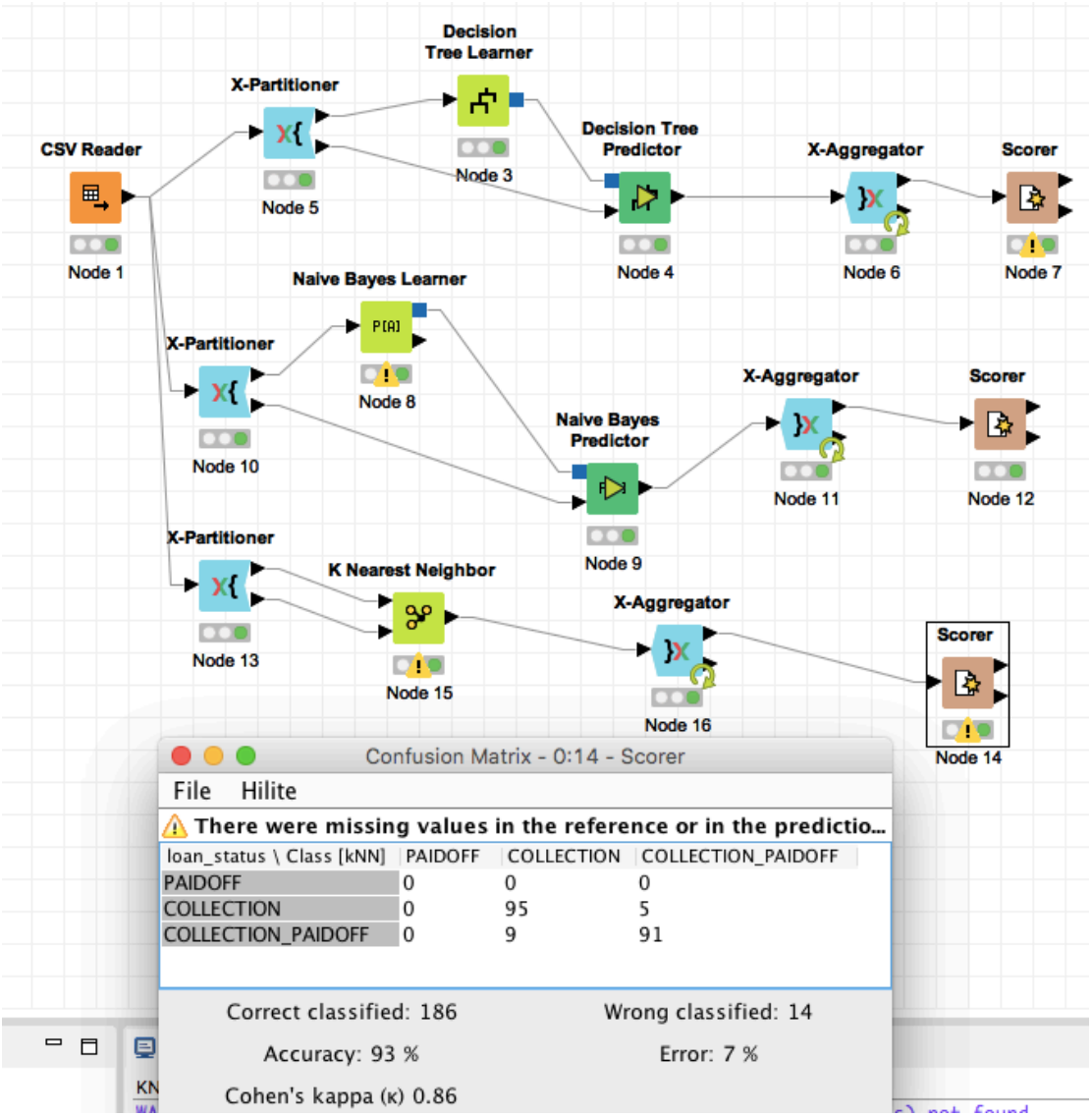
Şekil 12.2.10

Şekil 12.2.10, scorer operatörünün configure penceresini göstermektedir.



Şekil 12.2.11

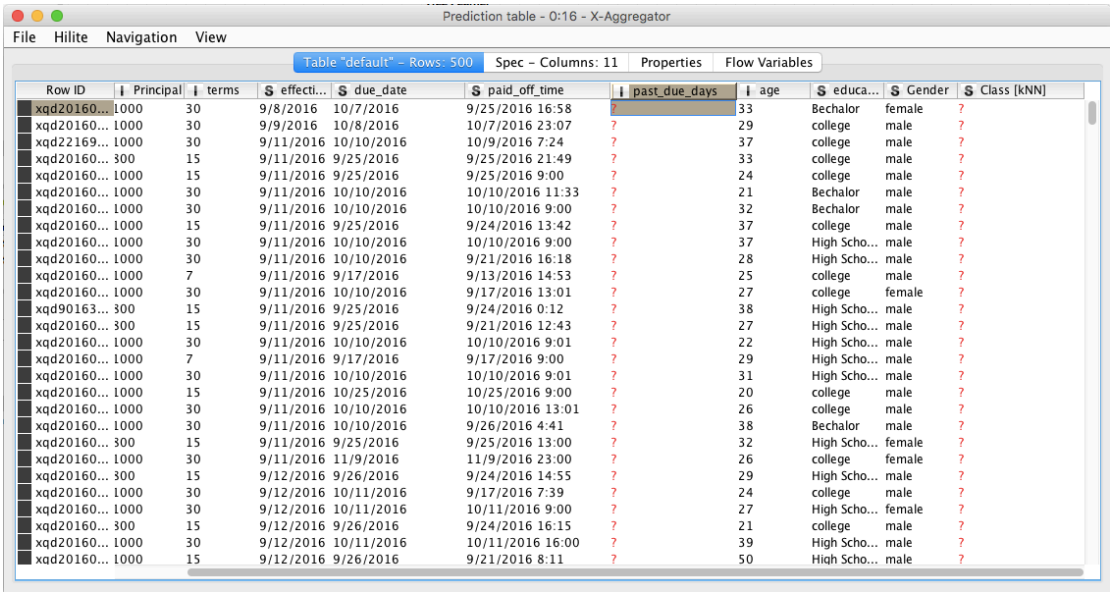
Şekil 12.2.11, sistem çalıştırıldıktan sonraki naive bayes operatörünün başarını gösteren confusion matrix penceresini göstermektedir. Bu seferki başarı 59.2%'dir.



Şekil 12.2.12

Şekil 12.2.12, başka bir operatörün de örnek olarak denenmesi için k nearest neighbor (k-NN) algoritmasının ve diğer operatörlerin eklenmesini göstermektedir. Configure bölümlerinde yukarıda yapıldığı gibi loan_status kolonu seçilmelidir hepsinde.

Aslında başarı 93% görülse de sistem paidoff'ları dahil etmediği için bu başarı tüm sistemin verilerini kullanarak elde edilmiş bir başarı değildir.



Row ID	Principal	terms	effecti...	due_date	paid_off_time	past_due_days	age	educa...	Gender	Class [kNN]
xqd20160...	1000	30	9/8/2016	10/7/2016	9/25/2016 16:58	?	33	Bechalor	female	?
xqd20160...	1000	30	9/9/2016	10/8/2016	10/7/2016 23:07	?	29	college	male	?
xqd22169...	1000	30	9/11/2016	10/10/2016	10/9/2016 7:24	?	37	college	male	?
xqd20160...	300	15	9/11/2016	9/25/2016	9/25/2016 21:49	?	33	college	male	?
xqd20160...	1000	15	9/11/2016	9/25/2016	9/25/2016 9:00	?	24	college	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	10/10/2016 11:33	?	21	Bechalor	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	10/10/2016 9:00	?	32	Bechalor	male	?
xqd20160...	1000	15	9/11/2016	9/25/2016	9/24/2016 13:42	?	37	college	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	10/10/2016 9:00	?	37	High Scho...	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	9/21/2016 16:18	?	28	High Scho...	male	?
xqd20160...	1000	7	9/11/2016	9/17/2016	9/13/2016 14:53	?	25	college	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	9/17/2016 13:01	?	27	college	female	?
xqd90163...	300	15	9/11/2016	9/25/2016	9/24/2016 0:12	?	38	High Scho...	male	?
xqd20160...	300	15	9/11/2016	9/25/2016	9/21/2016 12:43	?	27	High Scho...	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	10/10/2016 9:01	?	22	High Scho...	male	?
xqd20160...	1000	7	9/11/2016	9/17/2016	9/17/2016 9:00	?	29	High Scho...	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	10/10/2016 9:01	?	31	High Scho...	male	?
xqd20160...	1000	15	9/11/2016	10/25/2016	10/25/2016 9:00	?	20	college	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	10/10/2016 13:01	?	26	college	male	?
xqd20160...	1000	30	9/11/2016	10/10/2016	9/26/2016 4:41	?	38	Bechalor	male	?
xqd20160...	300	15	9/11/2016	9/25/2016	9/25/2016 13:00	?	32	High Scho...	female	?
xqd20160...	1000	30	9/11/2016	11/9/2016	11/9/2016 23:00	?	26	college	female	?
xqd20160...	300	15	9/12/2016	9/26/2016	9/24/2016 14:55	?	29	High Scho...	male	?
xqd20160...	1000	30	9/12/2016	10/11/2016	9/17/2016 7:39	?	24	college	male	?
xqd20160...	1000	30	9/12/2016	10/11/2016	10/11/2016 9:00	?	27	High Scho...	female	?
xqd20160...	300	15	9/12/2016	9/26/2016	9/24/2016 16:15	?	21	college	male	?
xqd20160...	1000	30	9/12/2016	10/11/2016	10/11/2016 16:00	?	39	High Scho...	male	?
xqd20160...	1000	15	9/12/2016	9/26/2016	9/21/2016 8:11	?	50	High Scho...	male	?

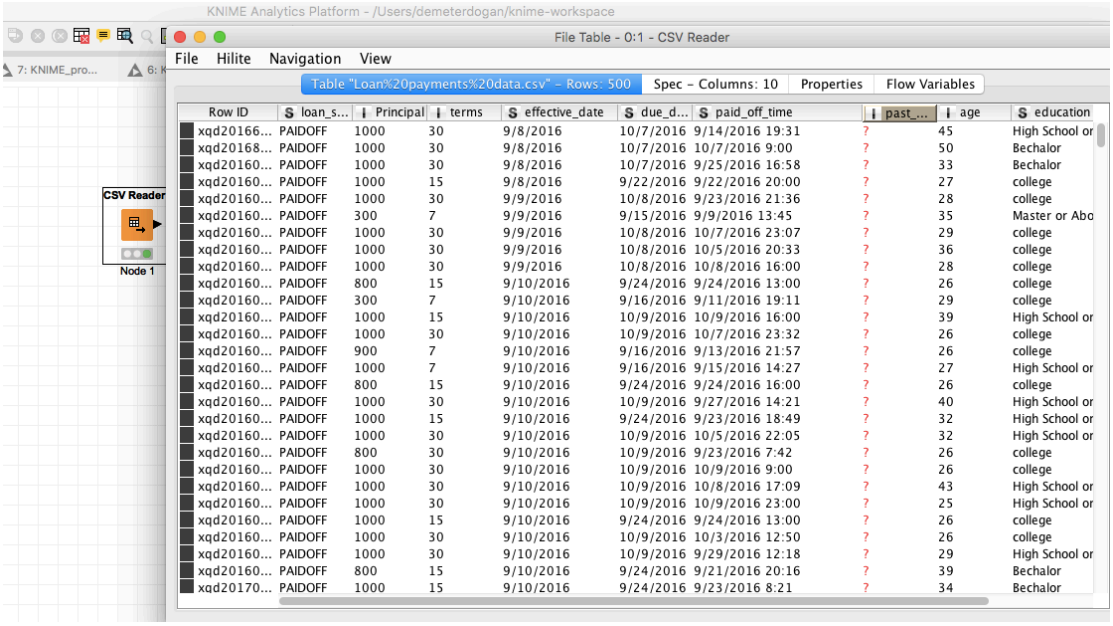
Şekil 12.2.13

Şekil 12.2.13, paidoff kolonlarının sisteme neden dahil edilmediğini gösteren x-aggregator sonuç penceresidir. Hali hazırda borç ödenmiş olduğu için paid_due_days (ödenecek gün) kolonunda soru işareti vardır. K-NN numaric değerleri kullandığı için oradaki soru işaretini missing value olarak almış ve onları işlemlere katmamıştır. O soru işaretli yerler 0 ile değiştirilebilir ya da numeric başka belirlenen bir değer verilirse o sıradaki değerler de işleme katılabilir.

12.3. Müşteri Ödeme Vade Tahmini

Bu bölümde, 1. bölümde indirilen veri kümesi (loan_status) ile müşterimin ödeme vade tahmininin yapılabilmesi gösterilecektir.

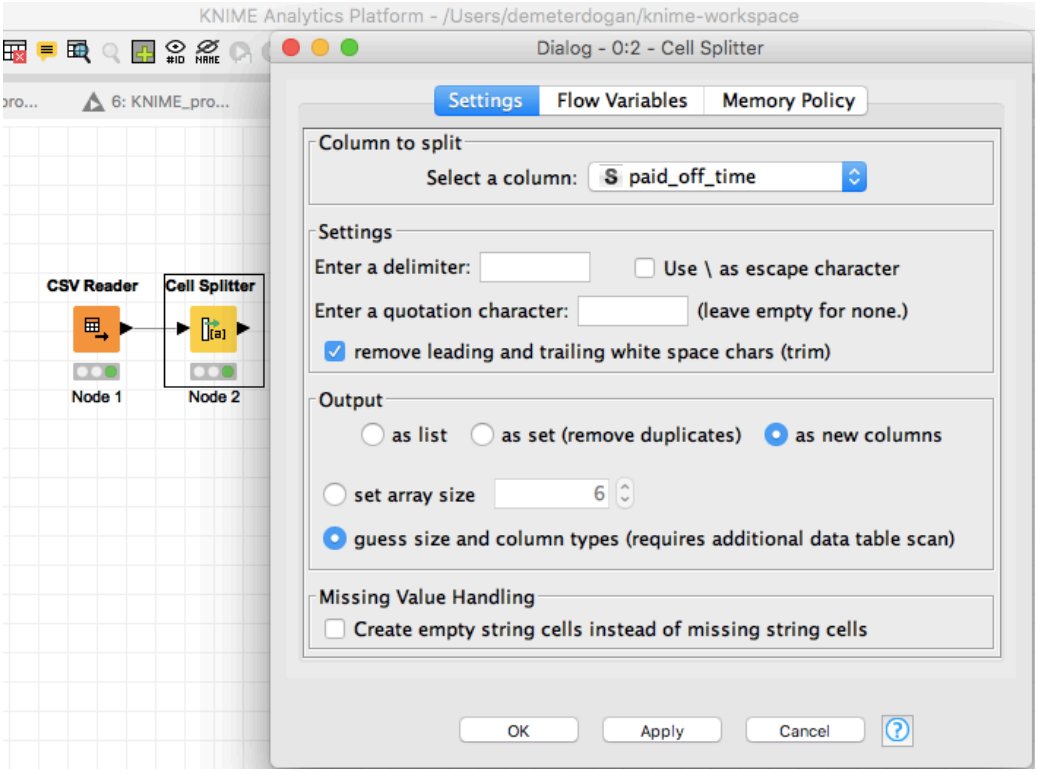
Prediction sayısal (numeric), classification da (nominal) değerler kullanılmaktadır yani bu bölümde prediction yapılacaktır.



Row ID	loan_s...	Principal	terms	effective_date	due_d...	paid_off_time	past_...	age	education
xqd20166...	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/14/2016 19:31	?	45	High School or
xqd20168...	PAIDOFF	1000	30	9/8/2016	10/7/2016	10/7/2016 9:00	?	50	Bechalor
xqd20160...	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/25/2016 16:58	?	33	Bechalor
xqd20160...	PAIDOFF	1000	15	9/8/2016	9/22/2016	9/22/2016 20:00	?	27	college
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	9/23/2016 21:36	?	28	college
xqd20160...	PAIDOFF	300	7	9/9/2016	9/15/2016	9/9/2016 13:45	?	35	Master or Abo
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/7/2016 23:07	?	29	college
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/5/2016 20:33	?	36	college
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/8/2016 16:00	?	28	college
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 13:00	?	26	college
xqd20160...	PAIDOFF	300	7	9/10/2016	9/16/2016	9/11/2016 19:11	?	29	college
xqd20160...	PAIDOFF	1000	15	9/10/2016	10/9/2016	10/9/2016 16:00	?	39	High School or
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/7/2016 23:32	?	26	college
xqd20160...	PAIDOFF	900	7	9/10/2016	9/16/2016	9/13/2016 21:57	?	26	college
xqd20160...	PAIDOFF	1000	7	9/10/2016	9/16/2016	9/15/2016 14:27	?	27	High School or
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 16:00	?	26	college
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	9/27/2016 14:21	?	40	High School or
xqd20160...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/23/2016 18:49	?	32	High School or
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/5/2016 22:05	?	32	High School or
xqd20160...	PAIDOFF	800	30	9/10/2016	10/9/2016	9/23/2016 7:42	?	26	college
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/9/2016 9:00	?	26	college
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/8/2016 17:09	?	43	High School or
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/9/2016 23:00	?	25	High School or
xqd20160...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/24/2016 13:00	?	26	college
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/3/2016 12:50	?	26	college
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	9/29/2016 12:18	?	29	High School or
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/21/2016 20:16	?	39	Bechalor
xqd20170...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/23/2016 8:21	?	34	Bechalor

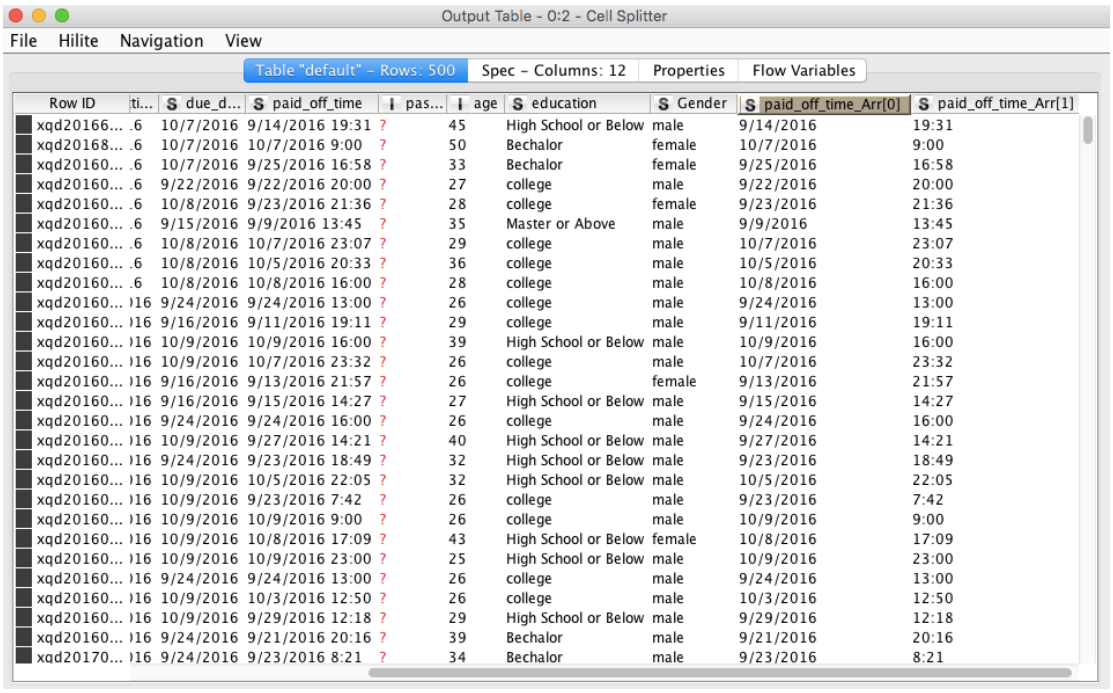
Şekil 12.3.1

Şekil 12.3.1 sisteme iki önceki bölümde kaggle'dan yüklenen veri setinin sisteme csv operatörü ile yüklenmesini ve veri seti içeriğini göstermektedir. Effective date ve paid off time bu bölümde kullanılmak istenilen kolonlardır. Aralarındaki fark ödemesi gereken zaman ve ödedikleri zaman olan kolonlar olduğu için bu kolonların date (tarih) formatında olmaları gerekmektedir. Şuan string formatında görülmekteler. Ayrıca paid off time kolonunda saat de yazmaktadır bunun da silinmesi ya da ayrılması gerekmektedir.



Şekil 12.3.2

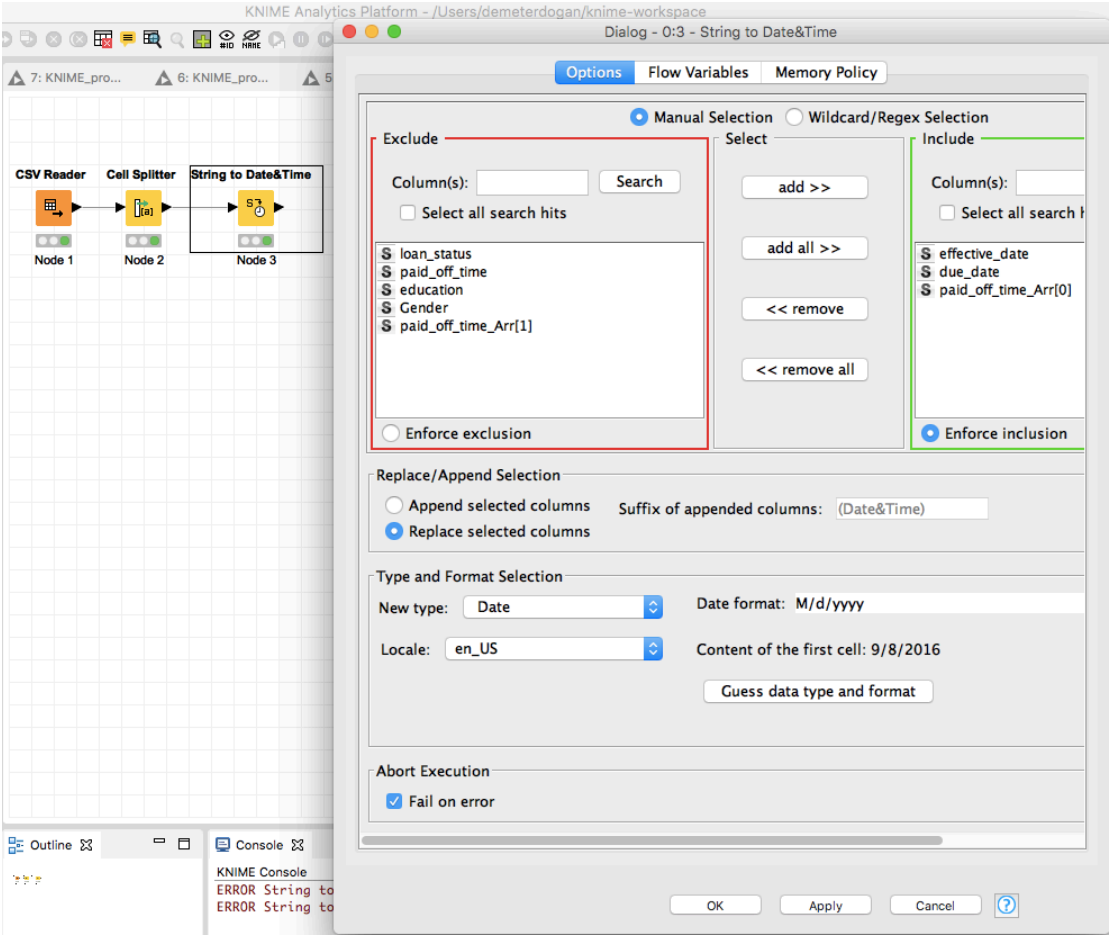
Şekil 12.3.2 sisteme cell splitter operatörünün eklenmesini ve configure bölümünü göstermektedir. Yukarıdaki şekilde bahsedilen kolonda saat bölümünün ayrılması için bu operatör kullanılmıştır. Enter a delimiter bölümüne bir boşluk (space) bırakılmıştır fakat ekran görüntüsünde fark edilmemektedir. Bu; boşluk olan yerden tarih ve saat bölümünü ayır demektir.



Row ID	ti...	due_d...	paid_off_time	pas...	age	education	Gender	paid_off_time_Arr[0]	paid_off_time_Arr[1]
xqd20166...	.6	10/7/2016	9/14/2016 19:31	?	45	High School or Below	male	9/14/2016	19:31
xqd20168...	.6	10/7/2016	10/7/2016 9:00	?	50	Bechalor	female	10/7/2016	9:00
xqd20160...	.6	10/7/2016	9/25/2016 16:58	?	33	Bechalor	female	9/25/2016	16:58
xqd20160...	.6	9/22/2016	9/22/2016 20:00	?	27	college	male	9/22/2016	20:00
xqd20160...	.6	10/8/2016	9/23/2016 21:36	?	28	college	female	9/23/2016	21:36
xqd20160...	.6	9/15/2016	9/9/2016 13:45	?	35	Master or Above	male	9/9/2016	13:45
xqd20160...	.6	10/8/2016	10/7/2016 23:07	?	29	college	male	10/7/2016	23:07
xqd20160...	.6	10/8/2016	10/5/2016 20:33	?	36	college	male	10/5/2016	20:33
xqd20160...	.6	10/8/2016	10/8/2016 16:00	?	28	college	male	10/8/2016	16:00
xqd20160...	16	9/24/2016	9/24/2016 13:00	?	26	college	male	9/24/2016	13:00
xqd20160...	16	9/16/2016	9/11/2016 19:11	?	29	college	male	9/11/2016	19:11
xqd20160...	16	10/9/2016	10/9/2016 16:00	?	39	High School or Below	male	10/9/2016	16:00
xqd20160...	16	10/9/2016	10/7/2016 23:32	?	26	college	male	10/7/2016	23:32
xqd20160...	16	9/16/2016	9/13/2016 21:57	?	26	college	female	9/13/2016	21:57
xqd20160...	16	9/16/2016	9/15/2016 14:27	?	27	High School or Below	male	9/15/2016	14:27
xqd20160...	16	9/24/2016	9/24/2016 16:00	?	26	college	male	9/24/2016	16:00
xqd20160...	16	10/9/2016	9/27/2016 14:21	?	40	High School or Below	male	9/27/2016	14:21
xqd20160...	16	9/24/2016	9/23/2016 18:49	?	32	High School or Below	male	9/23/2016	18:49
xqd20160...	16	10/9/2016	10/5/2016 22:05	?	32	High School or Below	male	10/5/2016	22:05
xqd20160...	16	10/9/2016	9/23/2016 7:42	?	26	college	male	9/23/2016	7:42
xqd20160...	16	10/9/2016	10/9/2016 9:00	?	26	college	male	10/9/2016	9:00
xqd20160...	16	10/9/2016	10/8/2016 17:09	?	43	High School or Below	female	10/8/2016	17:09
xqd20160...	16	10/9/2016	10/9/2016 23:00	?	25	High School or Below	male	10/9/2016	23:00
xqd20160...	16	9/24/2016	9/24/2016 13:00	?	26	college	male	9/24/2016	13:00
xqd20160...	16	10/9/2016	10/3/2016 12:50	?	26	college	male	10/3/2016	12:50
xqd20160...	16	10/9/2016	9/29/2016 12:18	?	29	High School or Below	male	9/29/2016	12:18
xqd20160...	16	9/24/2016	9/21/2016 20:16	?	39	Bechalor	male	9/21/2016	20:16
xqd20170...	16	9/24/2016	9/23/2016 8:21	?	34	Bechalor	male	9/23/2016	8:21

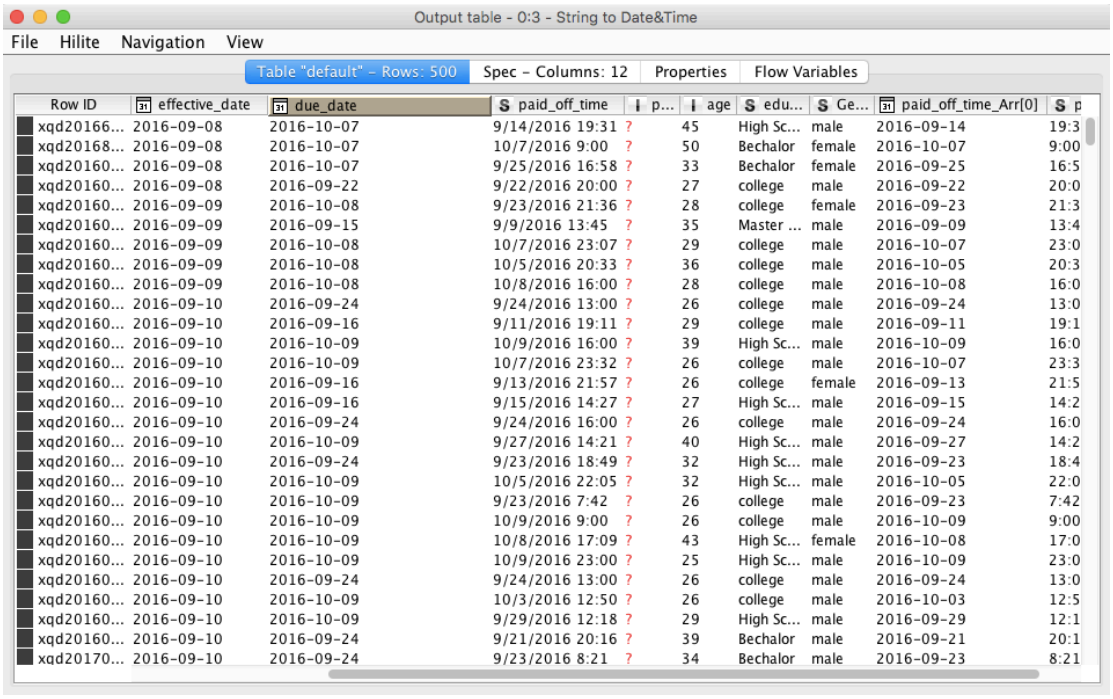
Şekil 12.3.3

Şekil 12.3.3, paid off time kolonunun saat ve tarih bölümlerinin 2 ayrı kolona ayrılmış halini göstermektedir. Paid off time arr[0] ilk bölümü yani ilk elemanı (tarih bölümünü) paid off time arr[1] ise 2. Elemanı yani saat kısmının olduğu yeri yeni kolon olarak açmıştır.



Şekil 12.3.4

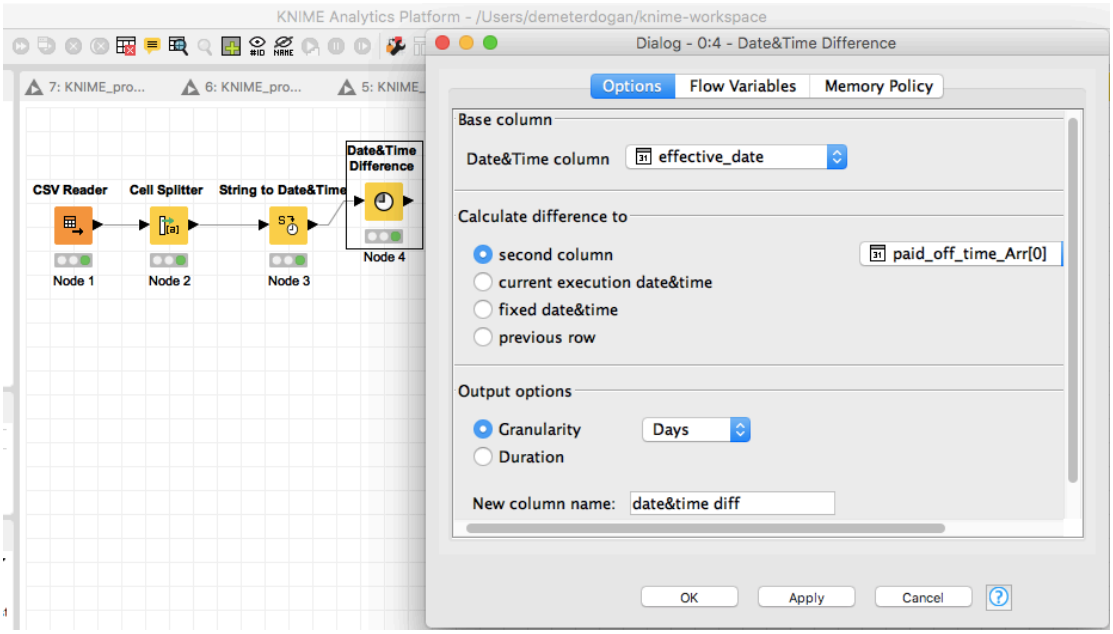
Şekil 12.3.4 sisteme string to date&times operatörünün eklenmesini ve configure penceresinde yapılan değişikliği göstermektedir. Effective date, due date, paid off time arr[0] kolonlarının string'ten date formatına dönüştürülmesi istenmiş ve M/d/yyy yani tek haneli olarak ay/gün/yıl olacak şekilde date formate (tarih formatı) belirlenmiştir.



Row ID	effective_date	due_date	paid_off_time	p...	age	edu...	Ge...	paid_off_time_Arr[0]	p
xqd20166...	2016-09-08	2016-10-07	9/14/2016 19:31	?	45	High Sc...	male	2016-09-14	19:3
xqd20168...	2016-09-08	2016-10-07	10/7/2016 9:00	?	50	Bechalor	female	2016-10-07	9:00
xqd20160...	2016-09-08	2016-10-07	9/25/2016 16:58	?	33	Bechalor	female	2016-09-25	16:5
xqd20160...	2016-09-08	2016-09-22	9/22/2016 20:00	?	27	college	male	2016-09-22	20:0
xqd20160...	2016-09-09	2016-10-08	9/23/2016 21:36	?	28	college	female	2016-09-23	21:3
xqd20160...	2016-09-09	2016-09-15	9/9/2016 13:45	?	35	Master ...	male	2016-09-09	13:4
xqd20160...	2016-09-09	2016-10-08	10/7/2016 23:07	?	29	college	male	2016-10-07	23:0
xqd20160...	2016-09-09	2016-10-08	10/5/2016 20:33	?	36	college	male	2016-10-05	20:3
xqd20160...	2016-09-09	2016-10-08	10/8/2016 16:00	?	28	college	male	2016-10-08	16:0
xqd20160...	2016-09-10	2016-09-24	9/24/2016 13:00	?	26	college	male	2016-09-24	13:0
xqd20160...	2016-09-10	2016-09-16	9/11/2016 19:11	?	29	college	male	2016-09-11	19:1
xqd20160...	2016-09-10	2016-10-09	10/9/2016 16:00	?	39	High Sc...	male	2016-10-09	16:0
xqd20160...	2016-09-10	2016-10-09	10/7/2016 23:32	?	26	college	male	2016-10-07	23:3
xqd20160...	2016-09-10	2016-09-16	9/13/2016 21:57	?	26	college	female	2016-09-13	21:5
xqd20160...	2016-09-10	2016-09-16	9/15/2016 14:27	?	27	High Sc...	male	2016-09-15	14:2
xqd20160...	2016-09-10	2016-09-24	9/24/2016 16:00	?	26	college	male	2016-09-24	16:0
xqd20160...	2016-09-10	2016-10-09	9/27/2016 14:21	?	40	High Sc...	male	2016-09-27	14:2
xqd20160...	2016-09-10	2016-09-24	9/23/2016 18:49	?	32	High Sc...	male	2016-09-23	18:4
xqd20160...	2016-09-10	2016-10-09	10/5/2016 22:05	?	32	High Sc...	male	2016-10-05	22:0
xqd20160...	2016-09-10	2016-10-09	9/23/2016 7:42	?	26	college	male	2016-09-23	7:42
xqd20160...	2016-09-10	2016-10-09	10/9/2016 9:00	?	26	college	male	2016-10-09	9:00
xqd20160...	2016-09-10	2016-10-09	10/8/2016 17:09	?	43	High Sc...	female	2016-10-08	17:0
xqd20160...	2016-09-10	2016-10-09	10/9/2016 23:00	?	25	High Sc...	male	2016-10-09	23:0
xqd20160...	2016-09-10	2016-09-24	9/24/2016 13:00	?	26	college	male	2016-09-24	13:0
xqd20160...	2016-09-10	2016-10-09	10/3/2016 12:50	?	26	college	male	2016-10-03	12:5
xqd20160...	2016-09-10	2016-10-09	9/29/2016 12:18	?	29	High Sc...	male	2016-09-29	12:1
xqd20160...	2016-09-10	2016-09-24	9/21/2016 20:16	?	39	Bechalor	male	2016-09-21	20:1
xqd20170...	2016-09-10	2016-09-24	9/23/2016 8:21	?	34	Bechalor	male	2016-09-23	8:21

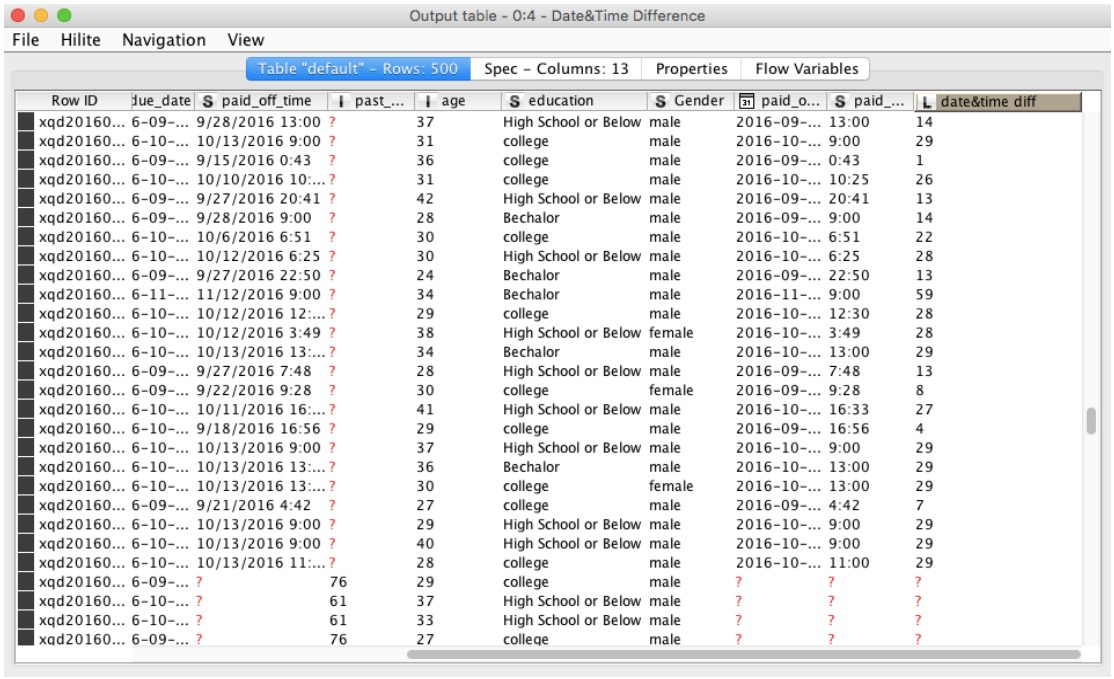
Şekil 12.3.5

Şekil 12.3.5, daha öncesinde string olan 3 kolonun yukarıdaki işlem sonucunda date formatında olduğunu göstermektedir.



Şekil 12.3.6

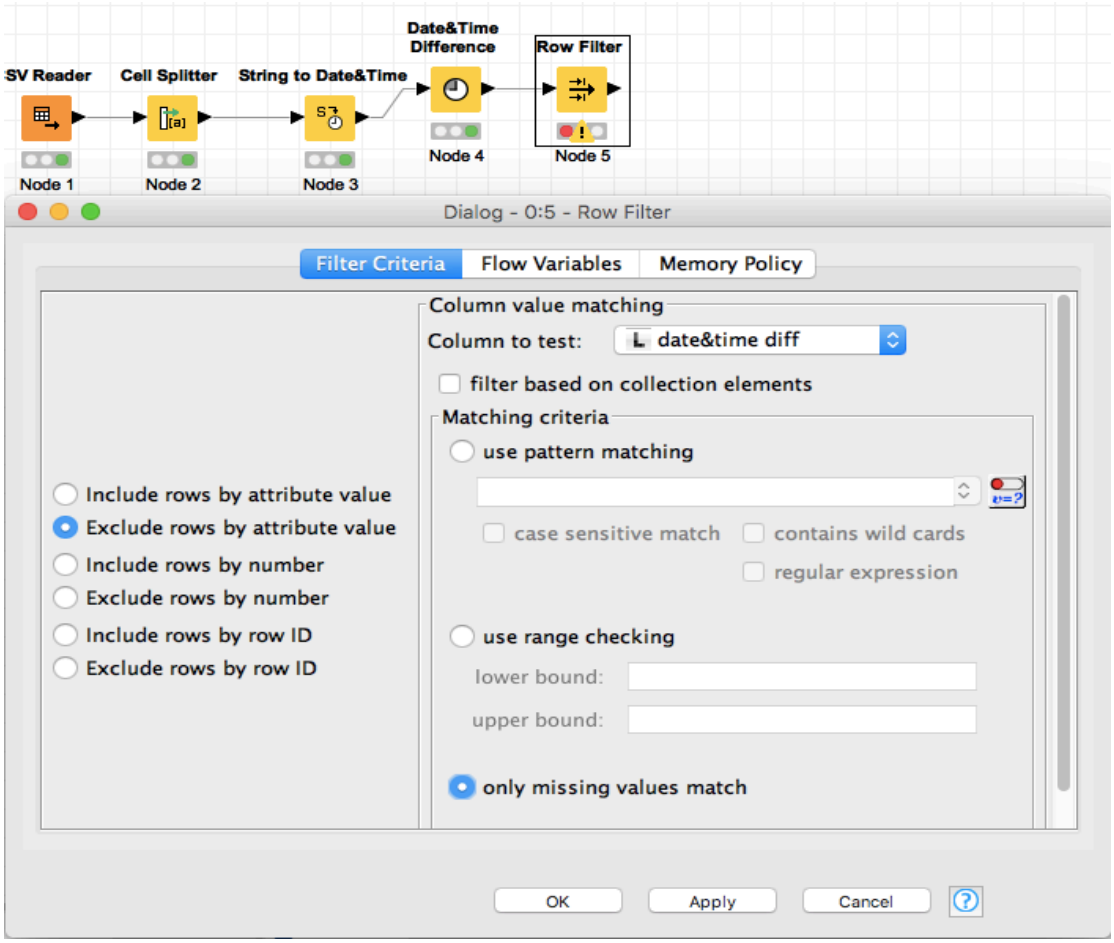
Şekil 12.3.6, tarihler arasında fark alınabilsin diye sisteme date&time difference operatörünün eklenmesini göstermektedir. Granularity ile bu farkın hangi cinsten yazılması istendiği belirtilir. Bu örnekte gün (days) olarak istenmiştir.



Row ID	due_date	paid_off_time	past_...	age	education	Gender	paid_o...	paid_...	date&time diff
xqd20160...	6-09-...	9/28/2016 13:00	?	37	High School or Below	male	2016-09-...	13:00	14
xqd20160...	6-10-...	10/13/2016 9:00	?	31	college	male	2016-10-...	9:00	29
xqd20160...	6-09-...	9/15/2016 0:43	?	36	college	male	2016-09-...	0:43	1
xqd20160...	6-10-...	10/10/2016 10:...	?	31	college	male	2016-10-...	10:25	26
xqd20160...	6-09-...	9/27/2016 20:41	?	42	High School or Below	male	2016-09-...	20:41	13
xqd20160...	6-09-...	9/28/2016 9:00	?	28	Bechalor	male	2016-09-...	9:00	14
xqd20160...	6-10-...	10/6/2016 6:51	?	30	college	male	2016-10-...	6:51	22
xqd20160...	6-10-...	10/12/2016 6:25	?	30	High School or Below	male	2016-10-...	6:25	28
xqd20160...	6-09-...	9/27/2016 22:50	?	24	Bechalor	male	2016-09-...	22:50	13
xqd20160...	6-11-...	11/12/2016 9:00	?	34	Bechalor	male	2016-11-...	9:00	59
xqd20160...	6-10-...	10/12/2016 12:...	?	29	college	male	2016-10-...	12:30	28
xqd20160...	6-10-...	10/12/2016 3:49	?	38	High School or Below	female	2016-10-...	3:49	28
xqd20160...	6-10-...	10/13/2016 13:...	?	34	Bechalor	male	2016-10-...	13:00	29
xqd20160...	6-09-...	9/27/2016 7:48	?	28	High School or Below	male	2016-09-...	7:48	13
xqd20160...	6-09-...	9/22/2016 9:28	?	30	college	female	2016-09-...	9:28	8
xqd20160...	6-10-...	10/11/2016 16:...	?	41	High School or Below	male	2016-10-...	16:33	27
xqd20160...	6-10-...	9/18/2016 16:56	?	29	college	male	2016-09-...	16:56	4
xqd20160...	6-10-...	10/13/2016 9:00	?	37	High School or Below	male	2016-10-...	9:00	29
xqd20160...	6-10-...	10/13/2016 13:...	?	36	Bechalor	male	2016-10-...	13:00	29
xqd20160...	6-10-...	10/13/2016 13:...	?	30	college	female	2016-10-...	13:00	29
xqd20160...	6-09-...	9/21/2016 4:42	?	27	college	male	2016-09-...	4:42	7
xqd20160...	6-10-...	10/13/2016 9:00	?	29	High School or Below	male	2016-10-...	9:00	29
xqd20160...	6-10-...	10/13/2016 9:00	?	40	High School or Below	male	2016-10-...	9:00	29
xqd20160...	6-10-...	10/13/2016 11:...	?	28	college	male	2016-10-...	11:00	29
xqd20160...	6-09-...	?	76	29	college	male	?	?	?
xqd20160...	6-10-...	?	61	37	High School or Below	male	?	?	?
xqd20160...	6-10-...	?	61	33	High School or Below	male	?	?	?
xqd20160...	6-09-...	?	76	27	college	male	?	?	?

Şekil 12.3.7

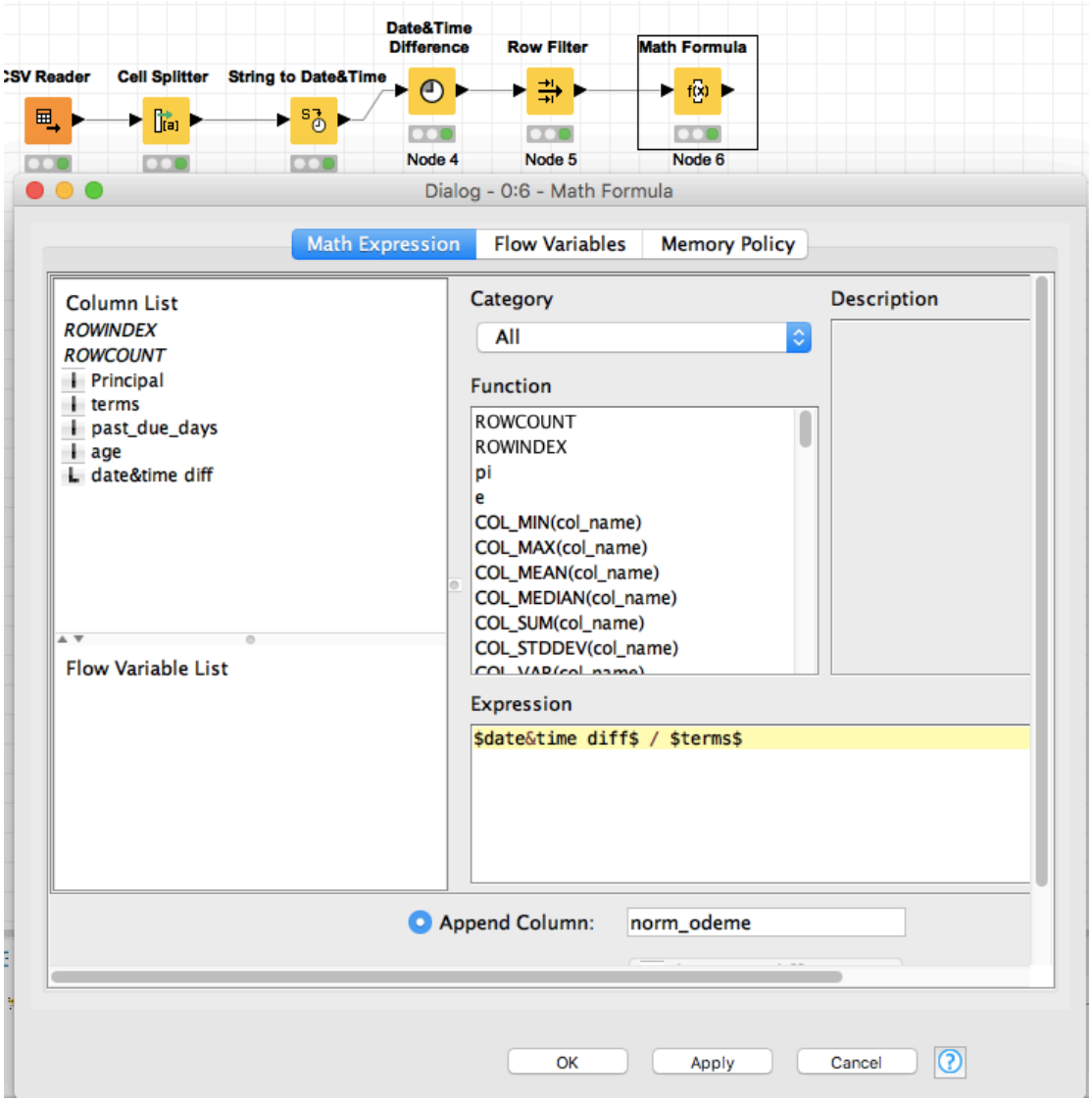
Şekil 12.3.7, effective date ile paid off time [0] arasındaki gün farkı date&time diff kolonunda gösterilmektedir. Soru işaretli sıralar eksik verilerin olduğu ve işlem yapılamamış yerleri göstermektedir.



Şekil 12.3.8

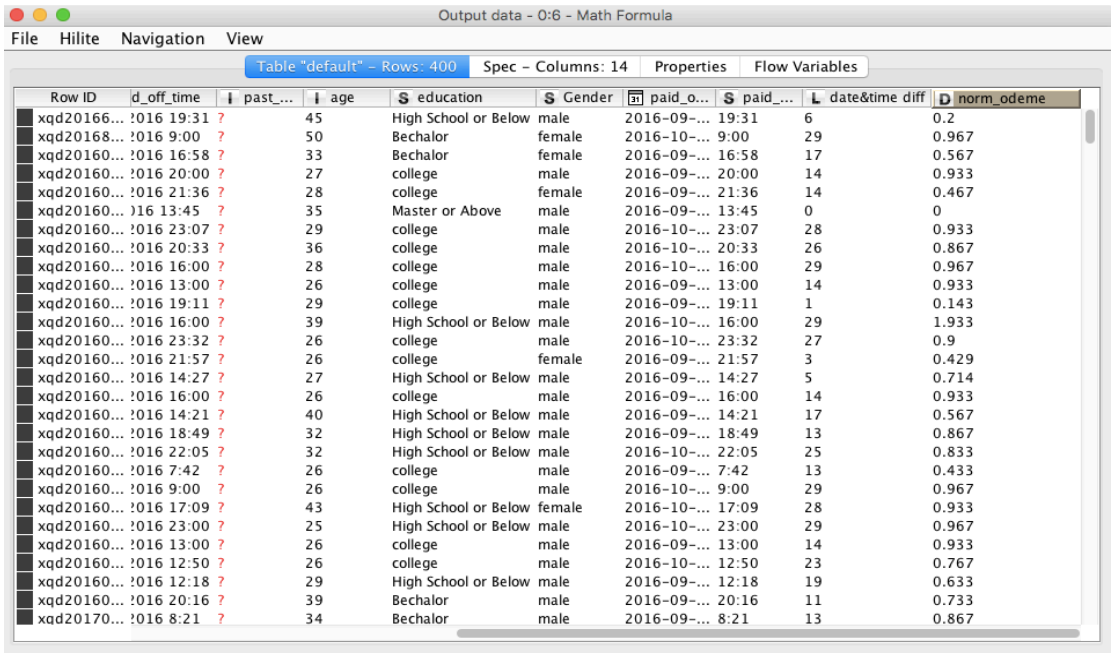
Şekil 12.3.8, yukarıda eksik verileri gösteren sıraların silinmesi için sisteme row filter operatörünün eklenmesini ve configure penceresini göstermektedir.

Ödeme alışkanlığı, düzeni koşullara göre hesaplama yapmak ve değerlendirmek gerekir. Terms kolonunda belirtilen rakamlar kaç günlüğüne borç alındığını belirtmektedir. Bir kişinin 30 günlük borç alıp 10 günde ödemesi ile 10 günlük borç alıp 3 günde ödemesi arasında fark vardır. Sistemde normalize etmek için borcu aldığı gün ödeyenlere 0, borcu zamanında ödeyenlere 1 ve borcunun zamanını geçirip ödeyenlere 2 değerleri verilsin. Normalize etmek için yeni oluşturulan date&time diff kolonunu terms kolonuna bölünmelidir.



Şekil 12.3.9

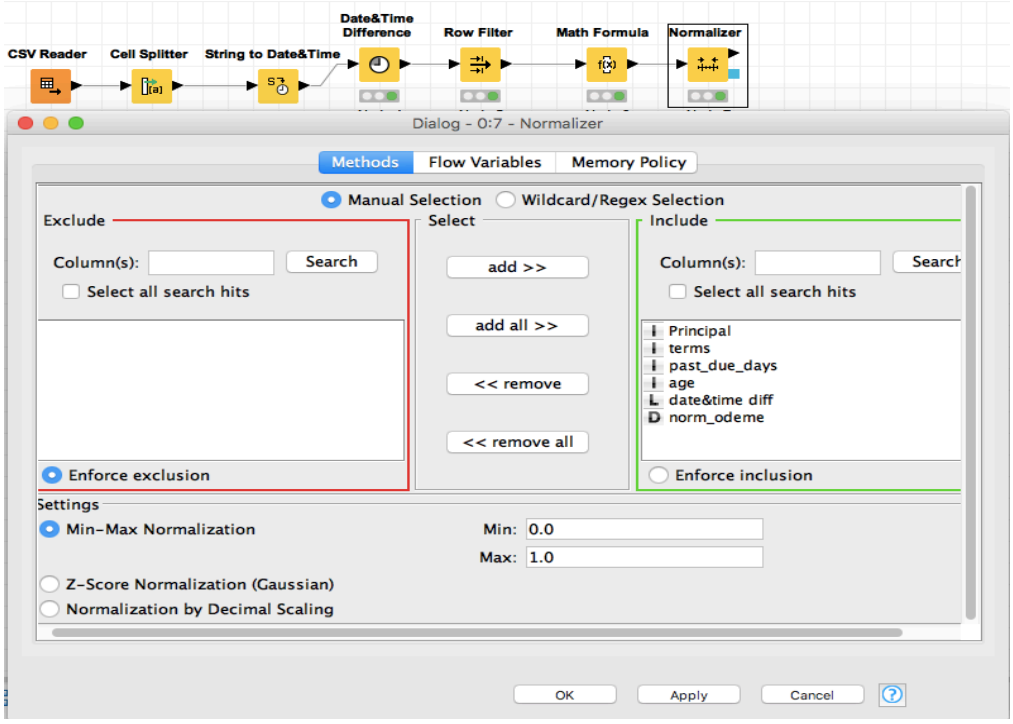
Şekil 12.3.9, math operatörünün sisteme eklenmesini ve içeriğinde yazılan formülü göstermektedir. Aslında bu aşama tam olarak normalize etmemektedir fakat normalize için ilk aşamadır.



Row ID	d_off_time	past_...	age	education	Gender	paid_o...	paid_...	date&time diff	norm_odeme
xqd20166...	?016 19:31	?	45	High School or Below	male	2016-09-...	19:31	6	0.2
xqd20168...	?016 9:00	?	50	Bechalor	female	2016-10-...	9:00	29	0.967
xqd20160...	?016 16:58	?	33	Bechalor	female	2016-09-...	16:58	17	0.567
xqd20160...	?016 20:00	?	27	college	male	2016-09-...	20:00	14	0.933
xqd20160...	?016 21:36	?	28	college	female	2016-09-...	21:36	14	0.467
xqd20160...	)16 13:45	?	35	Master or Above	male	2016-09-...	13:45	0	0
xqd20160...	?016 23:07	?	29	college	male	2016-10-...	23:07	28	0.933
xqd20160...	?016 20:33	?	36	college	male	2016-10-...	20:33	26	0.867
xqd20160...	?016 16:00	?	28	college	male	2016-10-...	16:00	29	0.967
xqd20160...	?016 13:00	?	26	college	male	2016-09-...	13:00	14	0.933
xqd20160...	?016 19:11	?	29	college	male	2016-09-...	19:11	1	0.143
xqd20160...	?016 16:00	?	39	High School or Below	male	2016-10-...	16:00	29	1.933
xqd20160...	?016 23:32	?	26	college	male	2016-10-...	23:32	27	0.9
xqd20160...	?016 21:57	?	26	college	female	2016-09-...	21:57	3	0.429
xqd20160...	?016 14:27	?	27	High School or Below	male	2016-09-...	14:27	5	0.714
xqd20160...	?016 16:00	?	26	college	male	2016-09-...	16:00	14	0.933
xqd20160...	?016 14:21	?	40	High School or Below	male	2016-09-...	14:21	17	0.567
xqd20160...	?016 18:49	?	32	High School or Below	male	2016-09-...	18:49	13	0.867
xqd20160...	?016 22:05	?	32	High School or Below	male	2016-10-...	22:05	25	0.833
xqd20160...	?016 7:42	?	26	college	male	2016-09-...	7:42	13	0.433
xqd20160...	?016 9:00	?	26	college	male	2016-10-...	9:00	29	0.967
xqd20160...	?016 17:09	?	43	High School or Below	female	2016-10-...	17:09	28	0.933
xqd20160...	?016 23:00	?	25	High School or Below	male	2016-10-...	23:00	29	0.967
xqd20160...	?016 13:00	?	26	college	male	2016-09-...	13:00	14	0.933
xqd20160...	?016 12:50	?	26	college	male	2016-10-...	12:50	23	0.767
xqd20160...	?016 12:18	?	29	High School or Below	male	2016-09-...	12:18	19	0.633
xqd20160...	?016 20:16	?	39	Bechalor	male	2016-09-...	20:16	11	0.733
xqd20170...	?016 8:21	?	34	Bechalor	male	2016-09-...	8:21	13	0.867

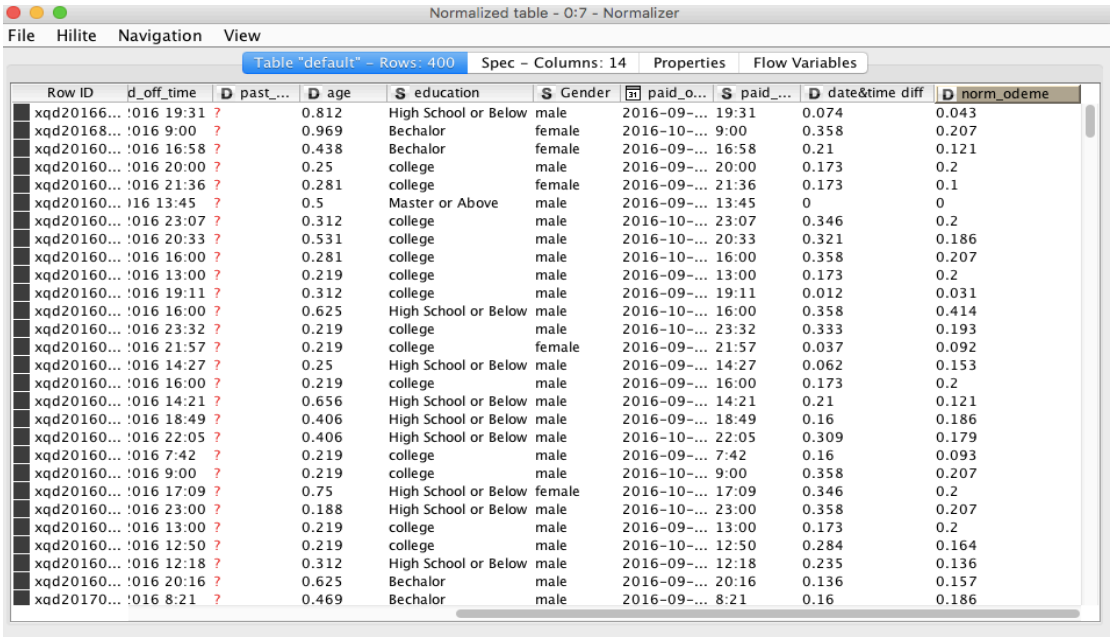
Şekil 12.3.10

Şekil 12.3.10, yukarıdaki işlem sonucunda oluşan sonuç penceresini (norm_odeme kolonunu) göstermektedir. Normalize edilmiş veriler belli bir değer aralığında olur. Burada belirli bir değer aralığı olmadığı için tam anlamıyla normalize edilmiş sayılamaz.



Şekil 12.3.11

Şekil 12.3.11, sisteme normalizer operatörünün eklenmesini ve configure penceresini göstermektedir. Min-max normalization bu örnekte kullanılacak normalizasyon yöntemidir. Herhangi bir sayının o kolonda bulunan minimum değer ile farkını alıp o kolondaki min ve max sayı farkına bölerek normalize eder.

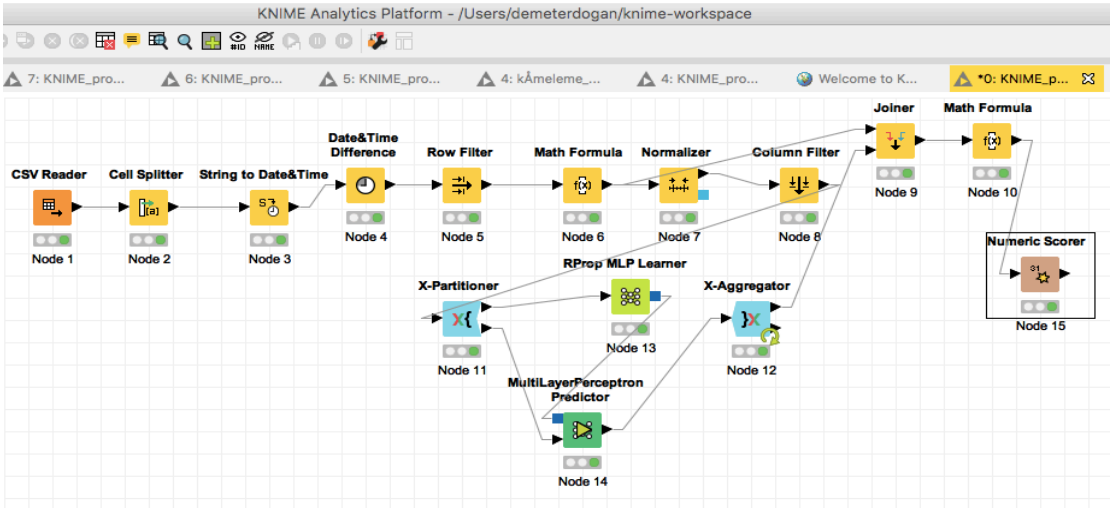


Row ID	d_off_time	past_...	age	education	Gender	paid_o...	paid_...	date&time diff	norm_odeme
xqd20166...	'016 19:31	?	0.812	High School or Below	male	2016-09-...	19:31	0.074	0.043
xqd20168...	'016 9:00	?	0.969	Bechalor	female	2016-10-...	9:00	0.358	0.207
xqd20160...	'016 16:58	?	0.438	Bechalor	female	2016-09-...	16:58	0.21	0.121
xqd20160...	'016 20:00	?	0.25	college	male	2016-09-...	20:00	0.173	0.2
xqd20160...	'016 21:36	?	0.281	college	female	2016-09-...	21:36	0.173	0.1
xqd20160...	)16 13:45	?	0.5	Master or Above	male	2016-09-...	13:45	0	0
xqd20160...	'016 23:07	?	0.312	college	male	2016-10-...	23:07	0.346	0.2
xqd20160...	'016 20:33	?	0.531	college	male	2016-10-...	20:33	0.321	0.186
xqd20160...	'016 16:00	?	0.281	college	male	2016-10-...	16:00	0.358	0.207
xqd20160...	'016 13:00	?	0.219	college	male	2016-09-...	13:00	0.173	0.2
xqd20160...	'016 19:11	?	0.312	college	male	2016-09-...	19:11	0.012	0.031
xqd20160...	'016 16:00	?	0.625	High School or Below	male	2016-10-...	16:00	0.358	0.414
xqd20160...	'016 23:32	?	0.219	college	male	2016-10-...	23:32	0.333	0.193
xqd20160...	'016 21:57	?	0.219	college	female	2016-09-...	21:57	0.037	0.092
xqd20160...	'016 14:27	?	0.25	High School or Below	male	2016-09-...	14:27	0.062	0.153
xqd20160...	'016 16:00	?	0.219	college	male	2016-09-...	16:00	0.173	0.2
xqd20160...	'016 14:21	?	0.656	High School or Below	male	2016-09-...	14:21	0.21	0.121
xqd20160...	'016 18:49	?	0.406	High School or Below	male	2016-09-...	18:49	0.16	0.186
xqd20160...	'016 22:05	?	0.406	High School or Below	male	2016-10-...	22:05	0.309	0.179
xqd20160...	'016 7:42	?	0.219	college	male	2016-09-...	7:42	0.16	0.093
xqd20160...	'016 9:00	?	0.219	college	male	2016-10-...	9:00	0.358	0.207
xqd20160...	'016 17:09	?	0.75	High School or Below	female	2016-10-...	17:09	0.346	0.2
xqd20160...	'016 23:00	?	0.188	High School or Below	male	2016-10-...	23:00	0.358	0.207
xqd20160...	'016 13:00	?	0.219	college	male	2016-09-...	13:00	0.173	0.2
xqd20160...	'016 12:50	?	0.219	college	male	2016-10-...	12:50	0.284	0.164
xqd20160...	'016 12:18	?	0.312	High School or Below	male	2016-09-...	12:18	0.235	0.136
xqd20160...	'016 20:16	?	0.625	Bechalor	male	2016-09-...	20:16	0.136	0.157
xqd20170...	'016 8:21	?	0.469	Bechalor	male	2016-09-...	8:21	0.16	0.186

Şekil 12.3.12

Şekil 12.3.12, normalize edilmiş değerlerin olduğu kolonu ve diğer kolonları göstermektedir. 0-1 arası belirtildiği için sadece bu aralıktaki değerler bulunabilir.

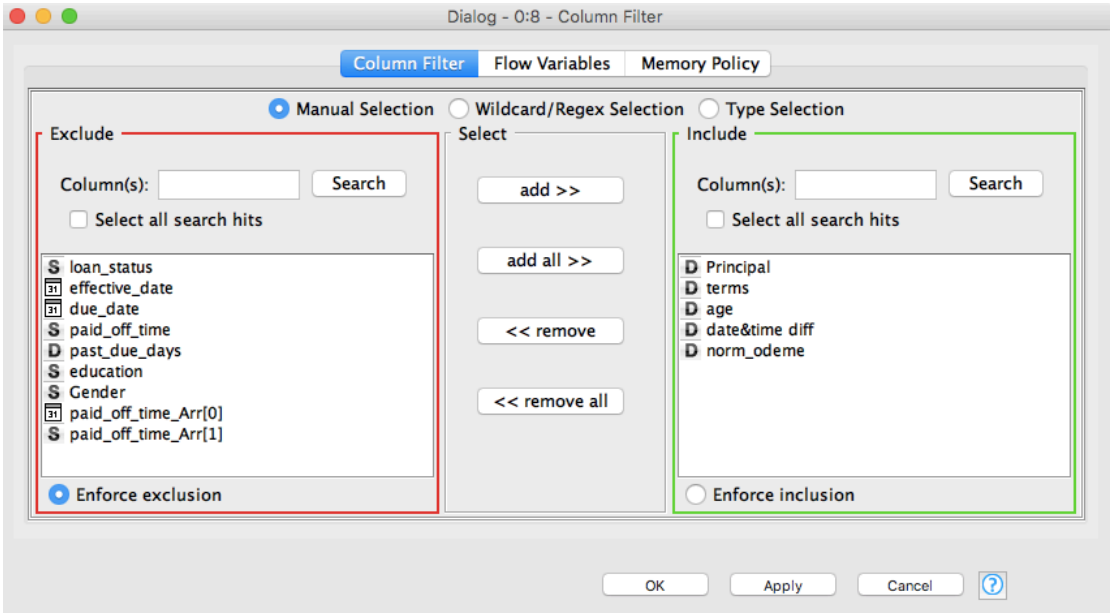
Daha öncesinde tarih bilgisi bulunuyordu veri setinde fakat bu aşamadan sonra sayısal değere çevrildiği için prediction edilebilir. Trainin veri setinin kullanılması test verisi olarak da kullanılması makinenin ezberlemesi riskini taşıdığı için burada kullanılmayacaktır. Bu yüzden validate ettirilmeli yani train için kullanılmayan bölümler test edilmeli ve bu şekilde tüm veri setindeki veriler sıra sıra hem testte hem de train de kullanılmalı.



Şekil 12.3.13

Şekil 12.3.13, sistemin son halini göstermektedir.

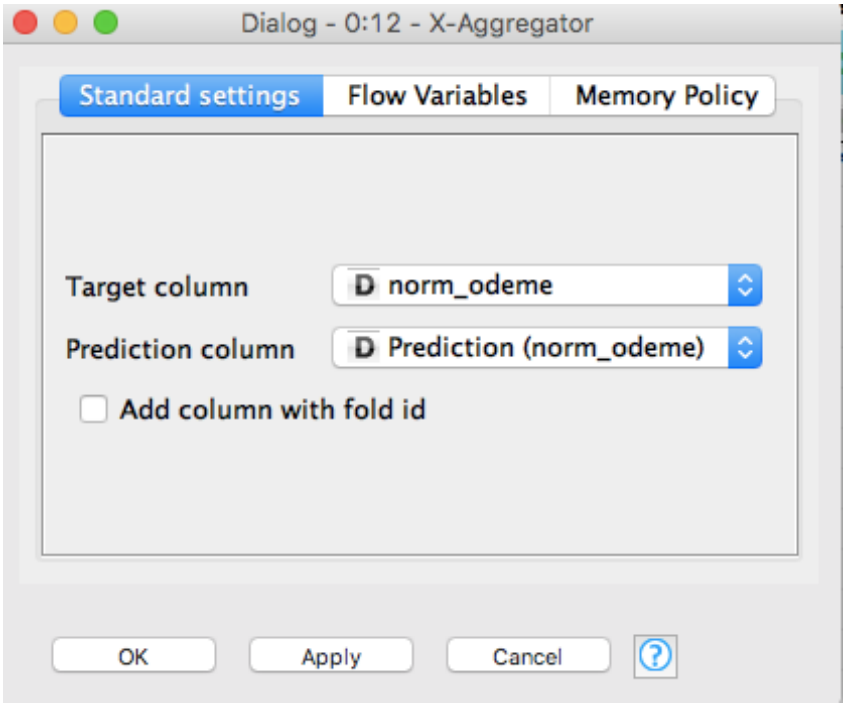
Normalizer operatöründen sonra column filter eklenmesindeki amaç RProp MLP Learner operatörü çalıştırılırken sadece double olan kolonları kullanabilmesidir.



Şekil 12.3.14

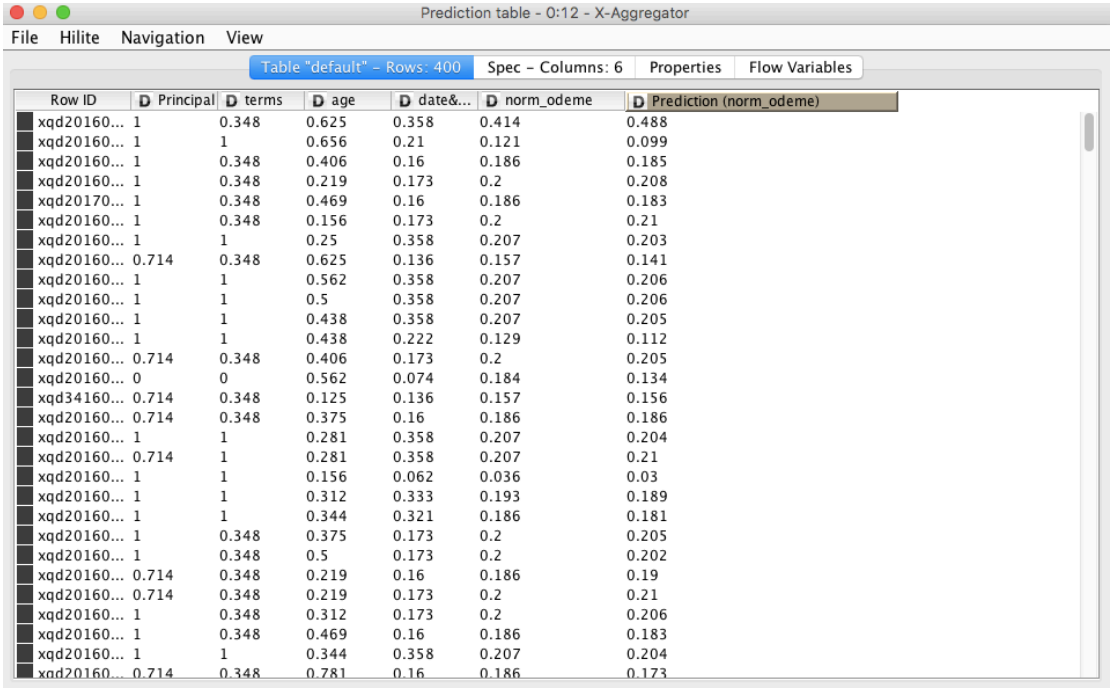
Şekil 12.3.14, column filter operatörünün configure penceresinde double olan özelliklerin seçimini göstermektedir.

X-partitioner ile veri seti validate edilerek tüm veriler bölüm bölüm hem trainingte hem de testte kullanılmak için sisteme eklendi. Past_due_date kolonunun kullanılmamasının nedeni, veri setinde zaten bulunmak istenilen bilginin o olmasıdır. Bir kişinin ne kadar geç ödeyeceğinin öğrenilmek istendiği modelde, kişinin yaşından, almak istediği paradan, kaç günlük vade ile almak istemesinden vb. Bilgiler ile bulunmalıdır. Ayrıca bu değerler kullanılsaydı o zaman bir kişinin borç alıp borcu zamanında ödemeyip kesin olarak geciktirmiş olunacağı var sayılıp hesaplama yapılmaya çalışması anlamına gelirdi. Kolon kullanılacak olsaydı bazı sıralarında (row) eksik veriler vardı ve o da RProp MLP Learner operatörü çalıştırılırken sorun oluşturacağı için sisteme missing value operatörü eklenerek bu kolonda eksik olan bilgiler tüm kolonun ortalamasını buralara yazarak ya da kolonda bulunan en küçük değeri buralara yazarak vb. Şekilde eksik bilgileri tamamlama yöntemlerinden biri bu missing value operatöründen seçilerek tamamlanabilirdi.



Şekil 12.3.15

Şekil 12.3.15, X-aggregator operatörünün configure penceresindeki düzenlemeyi göstermektedir. X-partitioner ile parçalanan training ve test için kullanılan verilerin bilgilerinin bu operatör ile birleşimi yapılması için sisteme eklenmiştir.

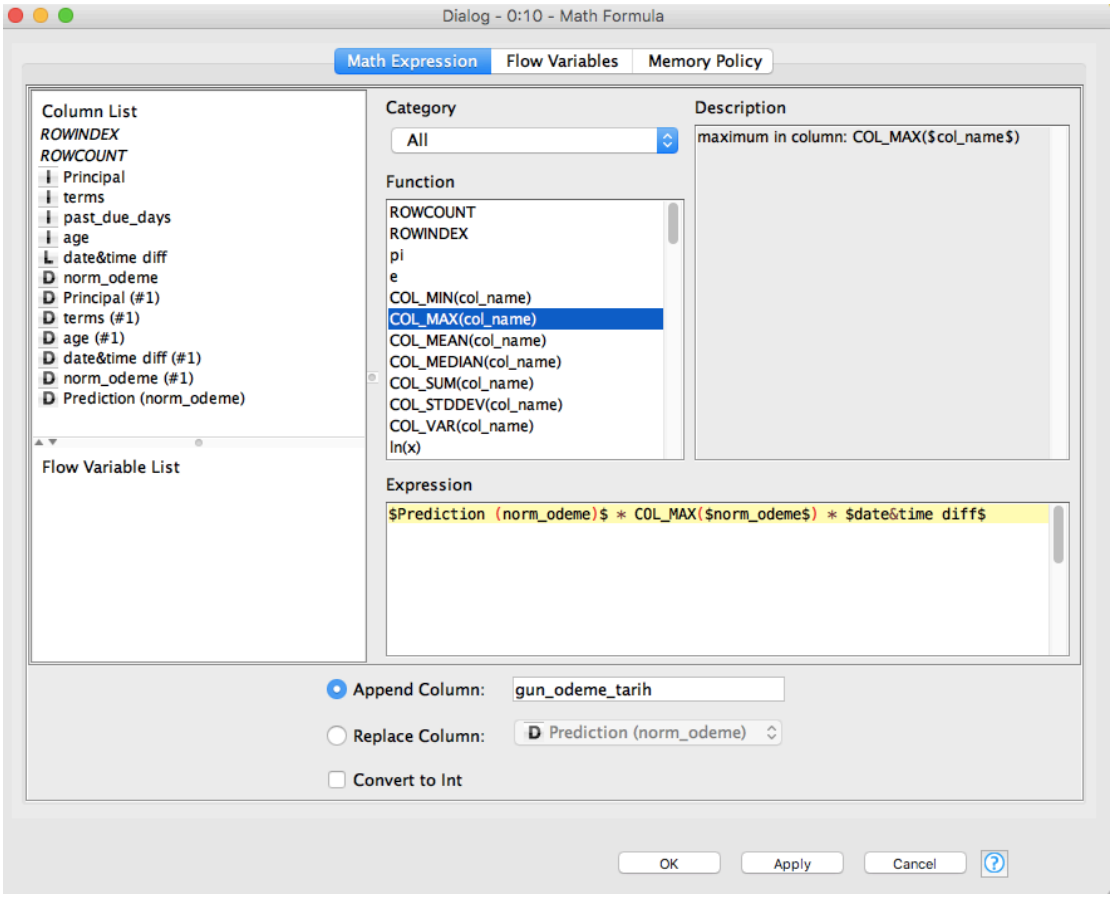


Row ID	Principal	terms	age	date&...	norm_odeme	Prediction (norm_odeme)
xqd20160...	1	0.348	0.625	0.358	0.414	0.488
xqd20160...	1	1	0.656	0.21	0.121	0.099
xqd20160...	1	0.348	0.406	0.16	0.186	0.185
xqd20160...	1	0.348	0.219	0.173	0.2	0.208
xqd20170...	1	0.348	0.469	0.16	0.186	0.183
xqd20160...	1	0.348	0.156	0.173	0.2	0.21
xqd20160...	1	1	0.25	0.358	0.207	0.203
xqd20160...	0.714	0.348	0.625	0.136	0.157	0.141
xqd20160...	1	1	0.562	0.358	0.207	0.206
xqd20160...	1	1	0.5	0.358	0.207	0.206
xqd20160...	1	1	0.438	0.358	0.207	0.205
xqd20160...	1	1	0.438	0.222	0.129	0.112
xqd20160...	0.714	0.348	0.406	0.173	0.2	0.205
xqd20160...	0	0	0.562	0.074	0.184	0.134
xqd34160...	0.714	0.348	0.125	0.136	0.157	0.156
xqd20160...	0.714	0.348	0.375	0.16	0.186	0.186
xqd20160...	1	1	0.281	0.358	0.207	0.204
xqd20160...	0.714	1	0.281	0.358	0.207	0.21
xqd20160...	1	1	0.156	0.062	0.036	0.03
xqd20160...	1	1	0.312	0.333	0.193	0.189
xqd20160...	1	1	0.344	0.321	0.186	0.181
xqd20160...	1	0.348	0.375	0.173	0.2	0.205
xqd20160...	1	0.348	0.5	0.173	0.2	0.202
xqd20160...	0.714	0.348	0.219	0.16	0.186	0.19
xqd20160...	0.714	0.348	0.219	0.173	0.2	0.21
xqd20160...	1	0.348	0.312	0.173	0.2	0.206
xqd20160...	1	0.348	0.469	0.16	0.186	0.183
xqd20160...	1	1	0.344	0.358	0.207	0.204
xqd20160...	0.714	0.348	0.781	0.16	0.186	0.173

Şekil 12.3.16

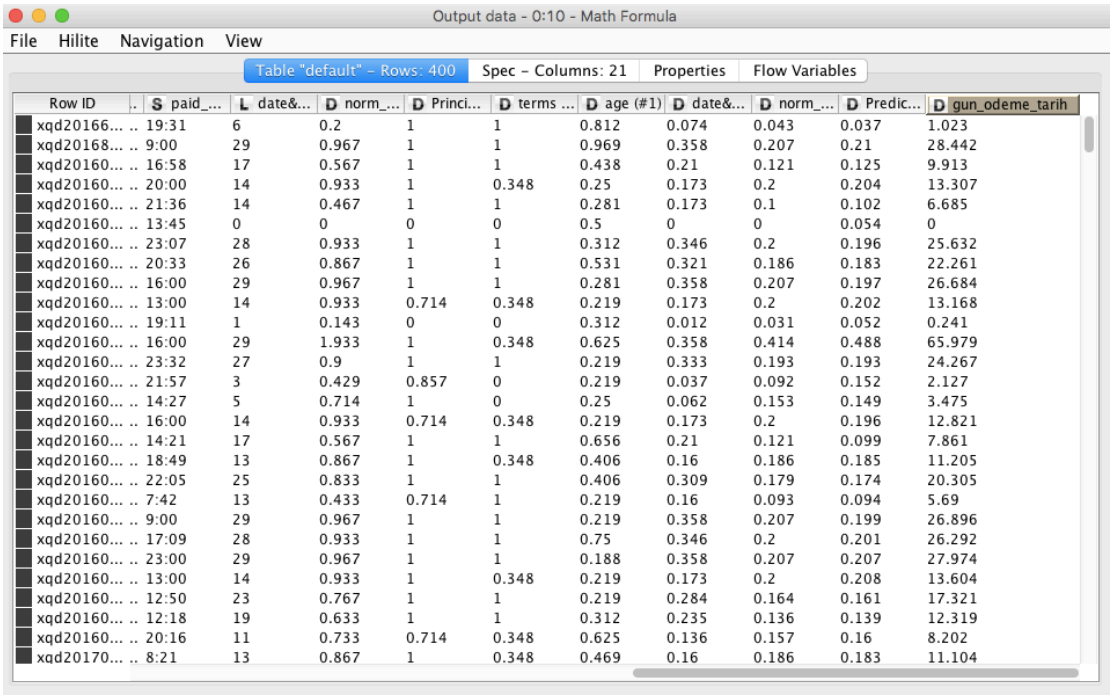
Şekil 12.3.16 X-aggregator operatörünün prediction table'ını göstermektedir. Yani normalize edilmiş verinin predict (tahmin edilmiş) değerlerini o kolonun yanında çıkarmıştır. Örneğin principal kolonunda olan en yüksek değer 1 bu 1000 TL gibi düşünülürse, 1000TL alan biri 1 aylık borç aldığında (terms kolonunda 1 ise) yaşı da 0.656 (kolondaki max yaş ile çarpılarak normalize edilmemiş gerçek değeri bulunur) ödemesi gereken zaman ile ödemesi beklenen zaman aralığında 0.21 birim vardır (normalize edilmiş tarih değeri).

Normalize edilmemiş değerleri alınarak math operatörü ile join edilerek (ıd'ler ile eşleştirilirken missing value olanlar da elenmiş olur) işlem yapılarak bu normalize edilmiş değer gerçek sayı değerine dönüştürülür.



Şekil 12.3.17

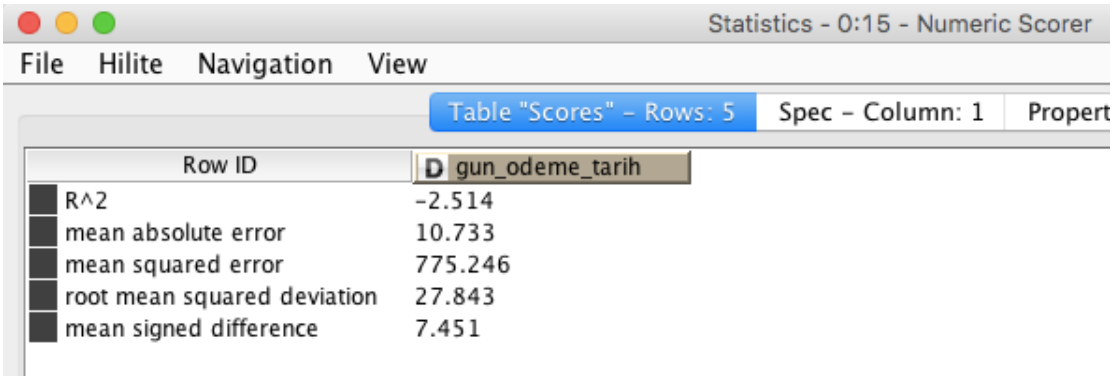
Şekil 12.3.17, math formula operatörünün configure penceresini göstermektedir. Normalize edilmiş ve kolondaki maximum değer ve tarihler arasındaki gün farkı ile çarpılarak gün bazında ödemesi çıkarılır.



Row ID	paid_...	date&...	norm_...	Princi...	terms ...	age (#1)	date&...	norm_...	Predic...	gun_odeme_tarih
xqd20166...	19:31	6	0.2	1	1	0.812	0.074	0.043	0.037	1.023
xqd20168...	9:00	29	0.967	1	1	0.969	0.358	0.207	0.21	28.442
xqd20160...	16:58	17	0.567	1	1	0.438	0.21	0.121	0.125	9.913
xqd20160...	20:00	14	0.933	1	0.348	0.25	0.173	0.2	0.204	13.307
xqd20160...	21:36	14	0.467	1	1	0.281	0.173	0.1	0.102	6.685
xqd20160...	13:45	0	0	0	0	0.5	0	0	0.054	0
xqd20160...	23:07	28	0.933	1	1	0.312	0.346	0.2	0.196	25.632
xqd20160...	20:33	26	0.867	1	1	0.531	0.321	0.186	0.183	22.261
xqd20160...	16:00	29	0.967	1	1	0.281	0.358	0.207	0.197	26.684
xqd20160...	13:00	14	0.933	0.714	0.348	0.219	0.173	0.2	0.202	13.168
xqd20160...	19:11	1	0.143	0	0	0.312	0.012	0.031	0.052	0.241
xqd20160...	16:00	29	1.933	1	0.348	0.625	0.358	0.414	0.488	65.979
xqd20160...	23:32	27	0.9	1	1	0.219	0.333	0.193	0.193	24.267
xqd20160...	21:57	3	0.429	0.857	0	0.219	0.037	0.092	0.152	2.127
xqd20160...	14:27	5	0.714	1	0	0.25	0.062	0.153	0.149	3.475
xqd20160...	16:00	14	0.933	0.714	0.348	0.219	0.173	0.2	0.196	12.821
xqd20160...	14:21	17	0.567	1	1	0.656	0.21	0.121	0.099	7.861
xqd20160...	18:49	13	0.867	1	0.348	0.406	0.16	0.186	0.185	11.205
xqd20160...	22:05	25	0.833	1	1	0.406	0.309	0.179	0.174	20.305
xqd20160...	7:42	13	0.433	0.714	1	0.219	0.16	0.093	0.094	5.69
xqd20160...	9:00	29	0.967	1	1	0.219	0.358	0.207	0.199	26.896
xqd20160...	17:09	28	0.933	1	1	0.75	0.346	0.2	0.201	26.292
xqd20160...	23:00	29	0.967	1	1	0.188	0.358	0.207	0.207	27.974
xqd20160...	13:00	14	0.933	1	0.348	0.219	0.173	0.2	0.208	13.604
xqd20160...	12:50	23	0.767	1	1	0.219	0.284	0.164	0.161	17.321
xqd20160...	12:18	19	0.633	1	1	0.312	0.235	0.136	0.139	12.319
xqd20160...	20:16	11	0.733	0.714	0.348	0.625	0.136	0.157	0.16	8.202
xqd20170...	8:21	13	0.867	1	0.348	0.469	0.16	0.186	0.183	11.104

Şekil 12.3.18

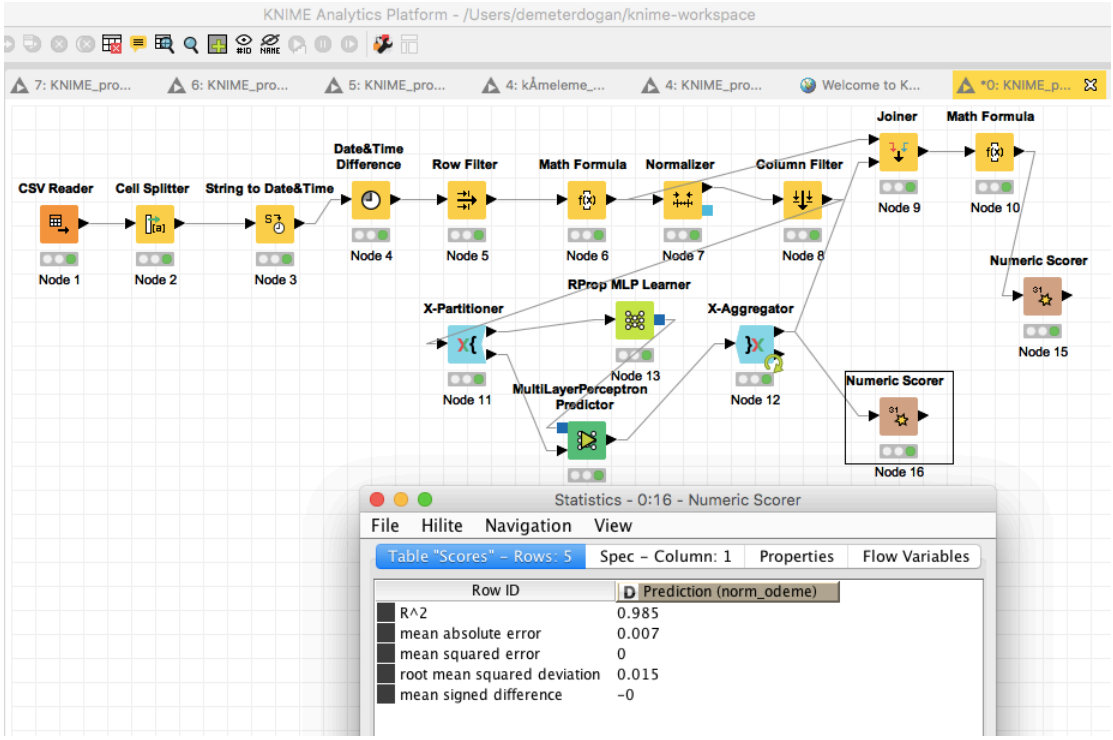
Şekil 12.3.18, math formula operatöründe yapılan işlemden sonra elde edilen output data'yı göstermektedir. Son kolonda da görüldüğü gibi örneğin birinci kişinin 1 günde ikinci kişinin 28 günde üçüncü kişinin 9.9 günde ödemesi beklenmektedir. Daha önce normalize ederken bölünün değer bu aşamada tekrar eski formuna dönüştürülebilmesi için çarpıldı.



Row ID	gun_odeme_tarih
R^2	-2.514
mean absolute error	10.733
mean squared error	775.246
root mean squared deviation	27.843
mean signed difference	7.451

Şekil 12.3.19

Sayısal olarak karşılaştırmak için sisteme numeric scorer kullanılır. Şekil 12.3.19 numeric scorer'ın statistic penceresini göstermektedir. Normalde rood mean squared 0-1 arasında değer olduğunda daha anlamlıydı. Şuan bu örnekte 27 gün çıkmış olması ödemesi gereken zaman ile ödeme yapacağı zaman arasında 27 gün olacağı anlamına gelmektedir.

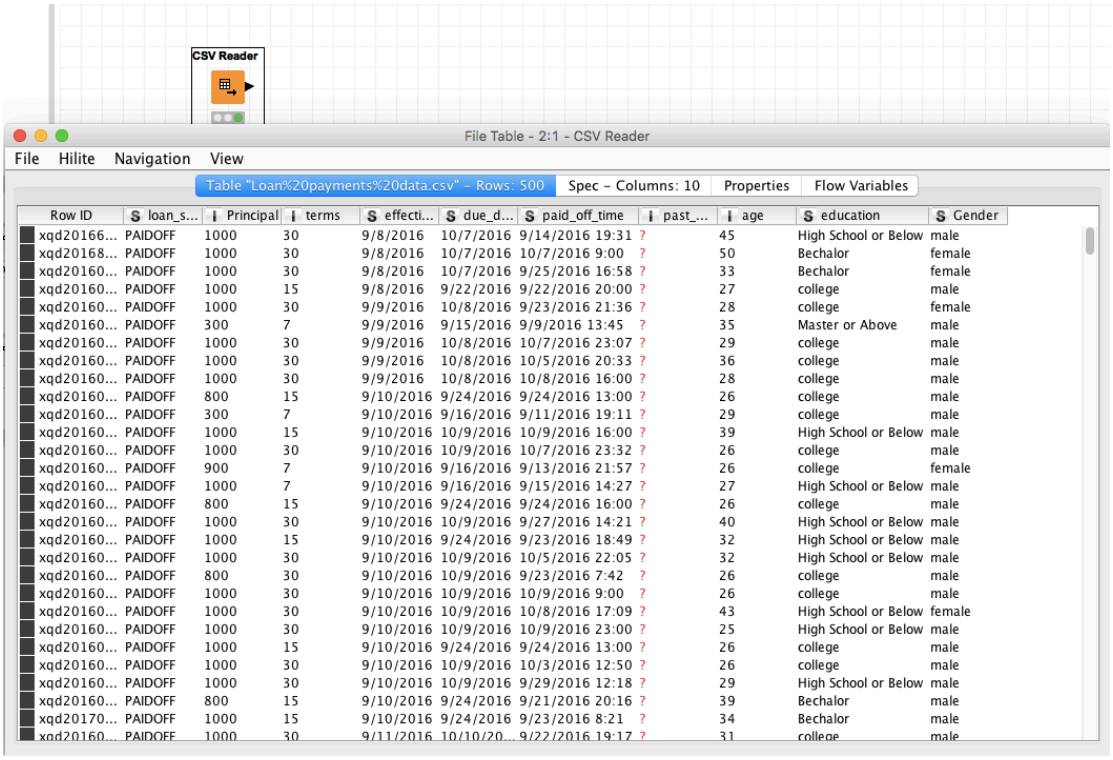


Şekil 12.3.20

Şekil 12.3.20, sistemin normalize edilmiş veri bölümüne direk numeric scorer eklenerek run edildiğinde elde edilen statistics'i göstermektedir. Az önce 27 gün fark olan değer tekrar gün formatına çevirilmiş sonuçtu. Burada ise ödemesi gereken zaman ile ödeme yapması planlanan zaman arasında 0.015'lik fark bulunmaktadır. Bu değer 0 ile 1 arasında olması gerektiği için 0.015 daha anlaşılabilir bir sonuçtur. 2% hata payı vardır denebilir. Rood mean squered error, hata paylarının karelerinin toplamının kaç tane eleman olduğuna bölünmesinin kare köküdür.

12.4. Müşteri Ödeme Vade Tahmini

Bu bölümde, 1. bölümde indirilen veri kümesi (loan_status) ile müşterilere verilebilecek krediyi tahmin eden sistemin oluşturulması gösterilecektir.

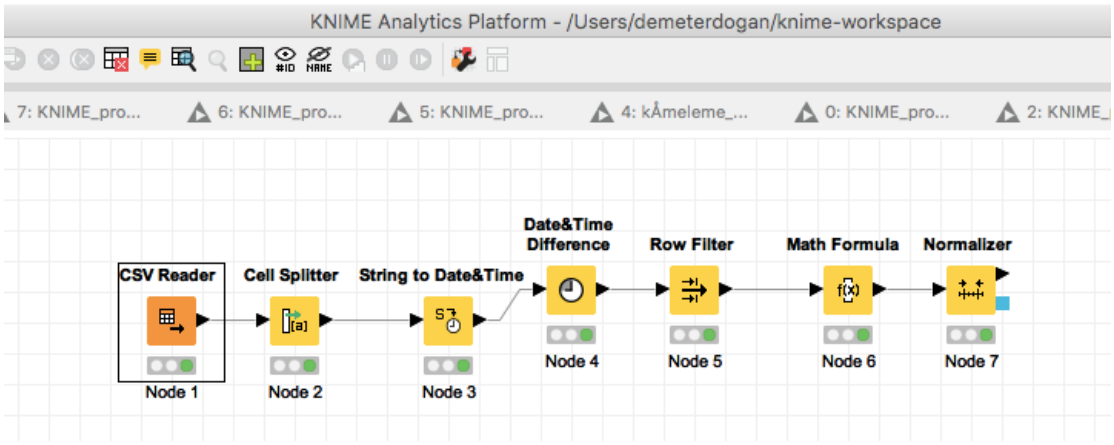


Row ID	loan_s...	Principal	terms	effecti...	due_d...	paid_off_time	past_...	age	education	Gender
xqd20166...	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/14/2016 19:31	?	45	High School or Below	male
xqd20168...	PAIDOFF	1000	30	9/8/2016	10/7/2016	10/7/2016 9:00	?	50	Bechalor	female
xqd20160...	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/25/2016 16:58	?	33	Bechalor	female
xqd20160...	PAIDOFF	1000	15	9/8/2016	9/22/2016	9/22/2016 20:00	?	27	college	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	9/23/2016 21:36	?	28	college	female
xqd20160...	PAIDOFF	300	7	9/9/2016	9/15/2016	9/9/2016 13:45	?	35	Master or Above	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/7/2016 23:07	?	29	college	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/5/2016 20:33	?	36	college	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/8/2016 16:00	?	28	college	male
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 13:00	?	26	college	male
xqd20160...	PAIDOFF	300	7	9/10/2016	9/16/2016	9/11/2016 19:11	?	29	college	male
xqd20160...	PAIDOFF	1000	15	9/10/2016	10/9/2016	10/9/2016 16:00	?	39	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/7/2016 23:32	?	26	college	male
xqd20160...	PAIDOFF	900	7	9/10/2016	9/16/2016	9/13/2016 21:57	?	26	college	female
xqd20160...	PAIDOFF	1000	7	9/10/2016	9/16/2016	9/15/2016 14:27	?	27	High School or Below	male
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 16:00	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	9/27/2016 14:21	?	40	High School or Below	male
xqd20160...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/23/2016 18:49	?	32	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/5/2016 22:05	?	32	High School or Below	male
xqd20160...	PAIDOFF	800	30	9/10/2016	10/9/2016	9/23/2016 7:42	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/9/2016 9:00	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/8/2016 17:09	?	43	High School or Below	female
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/9/2016 23:00	?	25	High School or Below	male
xqd20160...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/24/2016 13:00	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/3/2016 12:50	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	9/29/2016 12:18	?	29	High School or Below	male
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/21/2016 20:16	?	39	Bechalor	male
xqd20170...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/23/2016 8:21	?	34	Bechalor	male
xqd20160...	PAIDOFF	1000	30	9/11/2016	10/10/20...	9/22/2016 19:17	?	31	college	male

Şekil 12.4.1

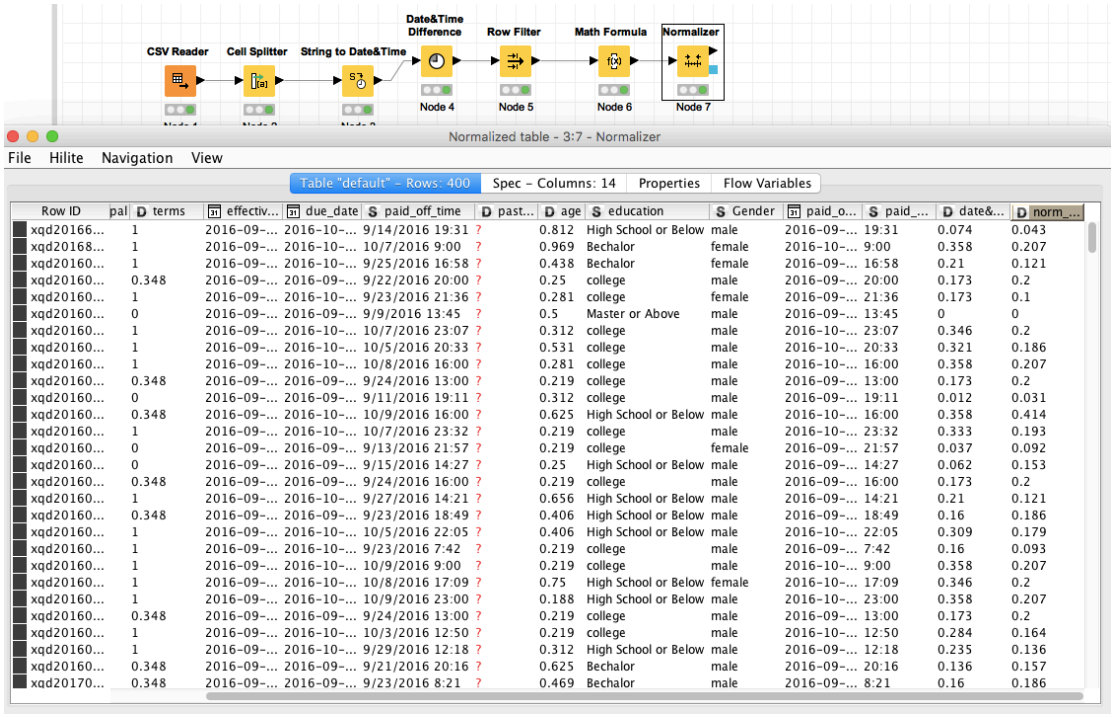
Şekil 12.4.1 birinci bölümde kullanılan loan payments datasının sisteme cvs reader ile yüklenmesini ve veri seti içeriğini göstermektedir. Burada amaç, yaşı, eğitim durumu, cinsiyeti vb. Özellikleri bilinen kişinin ne kadar borç istediği ile ne kadar sürelik borç istediği arasındaki ilişki ile ne kadar borç verilebileceğini bulunmak. Örneğin terms kolonunda 7 olan (7 günlük borç) bir kişinin 300 TL almak istediği biliniyor. Bu bilgi işlenerek aslında principal'ı (istediği miktarın) ne olabileceğini makine öğrenmesi ile bulmak sonuçta elde edilmek istenilen bilgi. Sayısal (numeric) bir değere ulaşılmak istendiği için predicton uygulanmalıdır. Örnek olarak neural network yöntemi kullanılacaktır.

Daha önceki bölümde de açıklandığı gibi neural network kullanmak için verilerin double formatında ve normalize edilmiş veriler olmalı. Bu yüzden daha önceki bölümdeki yöntemlerin aynısı kullanılacaktır.



Şekil 12.4.2

Şekil 12.4.2, veri setinin normalize edilene kadarki kullanılan operatörleri ve bağlantılarını göstermektedir. Bir önceki bölümde hepsinin kullanım nedeni detaylı açıklanmıştır.

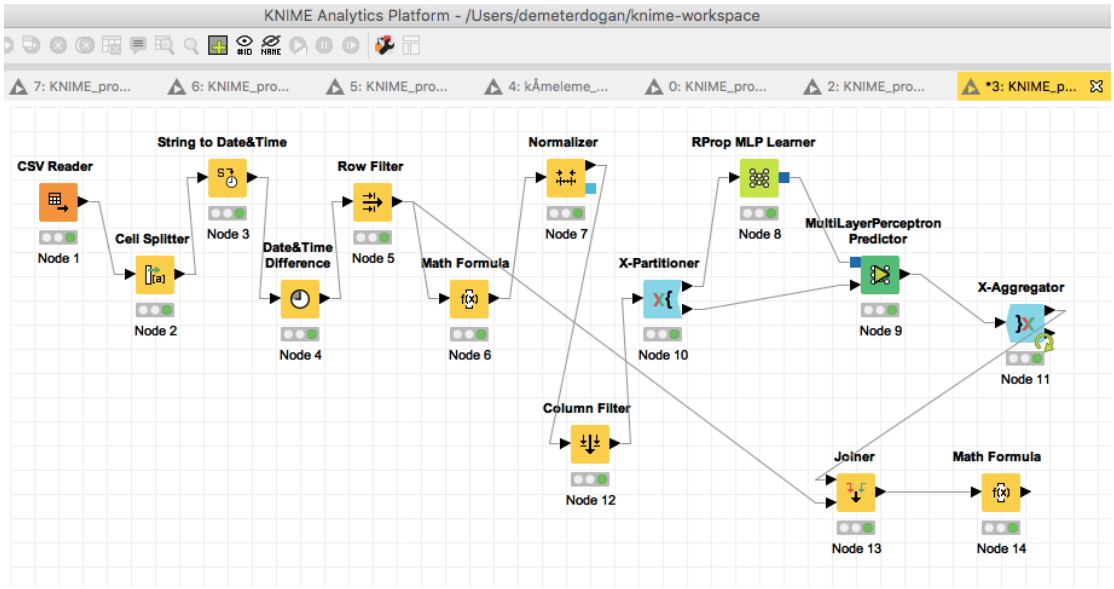


Row ID	pal	terms	effectiv...	due_date	paid_off_time	past...	age	education	Gender	paid_o...	paid_...	date&...	norm_...
xqd20166...		1	2016-09-...	2016-10-...	9/14/2016 19:31	?	0.812	High School or Below	male	2016-09-...	19:31	0.074	0.043
xqd20168...		1	2016-09-...	2016-10-...	10/7/2016 9:00	?	0.969	Bechalor	female	2016-10-...	9:00	0.358	0.207
xqd20160...		1	2016-09-...	2016-10-...	9/25/2016 16:58	?	0.438	Bechalor	female	2016-09-...	16:58	0.21	0.121
xqd20160...		0.348	2016-09-...	2016-09-...	9/22/2016 20:00	?	0.25	college	male	2016-09-...	20:00	0.173	0.2
xqd20160...		1	2016-09-...	2016-10-...	9/23/2016 21:36	?	0.281	college	female	2016-09-...	21:36	0.173	0.1
xqd20160...		0	2016-09-...	2016-09-...	9/9/2016 13:45	?	0.5	Master or Above	male	2016-09-...	13:45	0	0
xqd20160...		1	2016-09-...	2016-10-...	10/7/2016 23:07	?	0.312	college	male	2016-10-...	23:07	0.346	0.2
xqd20160...		1	2016-09-...	2016-10-...	10/5/2016 20:33	?	0.531	college	male	2016-10-...	20:33	0.321	0.186
xqd20160...		1	2016-09-...	2016-10-...	10/8/2016 16:00	?	0.281	college	male	2016-10-...	16:00	0.358	0.207
xqd20160...		0.348	2016-09-...	2016-09-...	9/24/2016 13:00	?	0.219	college	male	2016-09-...	13:00	0.173	0.2
xqd20160...		0	2016-09-...	2016-09-...	9/11/2016 19:11	?	0.312	college	male	2016-09-...	19:11	0.012	0.031
xqd20160...		0.348	2016-09-...	2016-10-...	10/9/2016 16:00	?	0.625	High School or Below	male	2016-10-...	16:00	0.358	0.414
xqd20160...		1	2016-09-...	2016-10-...	10/7/2016 23:32	?	0.219	college	male	2016-10-...	23:32	0.333	0.193
xqd20160...		0	2016-09-...	2016-09-...	9/13/2016 21:57	?	0.219	college	female	2016-09-...	21:57	0.037	0.092
xqd20160...		0	2016-09-...	2016-09-...	9/15/2016 14:27	?	0.25	High School or Below	male	2016-09-...	14:27	0.062	0.153
xqd20160...		0.348	2016-09-...	2016-09-...	9/24/2016 16:00	?	0.219	college	male	2016-09-...	16:00	0.173	0.2
xqd20160...		1	2016-09-...	2016-10-...	9/27/2016 14:21	?	0.656	High School or Below	male	2016-09-...	14:21	0.21	0.121
xqd20160...		0.348	2016-09-...	2016-09-...	9/23/2016 18:49	?	0.406	High School or Below	male	2016-09-...	18:49	0.16	0.186
xqd20160...		1	2016-09-...	2016-10-...	10/5/2016 22:05	?	0.406	High School or Below	male	2016-10-...	22:05	0.309	0.179
xqd20160...		1	2016-09-...	2016-10-...	9/23/2016 7:42	?	0.219	college	male	2016-09-...	7:42	0.16	0.093
xqd20160...		1	2016-09-...	2016-10-...	10/9/2016 9:00	?	0.219	college	male	2016-10-...	9:00	0.358	0.207
xqd20160...		1	2016-09-...	2016-10-...	10/8/2016 17:09	?	0.75	High School or Below	female	2016-10-...	17:09	0.346	0.2
xqd20160...		1	2016-09-...	2016-10-...	10/9/2016 23:00	?	0.188	High School or Below	male	2016-10-...	23:00	0.358	0.207
xqd20160...		0.348	2016-09-...	2016-09-...	9/24/2016 13:00	?	0.219	college	male	2016-09-...	13:00	0.173	0.2
xqd20160...		1	2016-09-...	2016-10-...	10/3/2016 12:50	?	0.219	college	male	2016-10-...	12:50	0.284	0.164
xqd20160...		1	2016-09-...	2016-10-...	9/29/2016 12:18	?	0.312	High School or Below	male	2016-09-...	12:18	0.235	0.136
xqd20160...		0.348	2016-09-...	2016-09-...	9/21/2016 20:16	?	0.625	Bechalor	male	2016-09-...	20:16	0.136	0.157
xqd20170...		0.348	2016-09-...	2016-09-...	9/23/2016 8:21	?	0.469	Bechalor	male	2016-09-...	8:21	0.16	0.186

Şekil 12.4.3

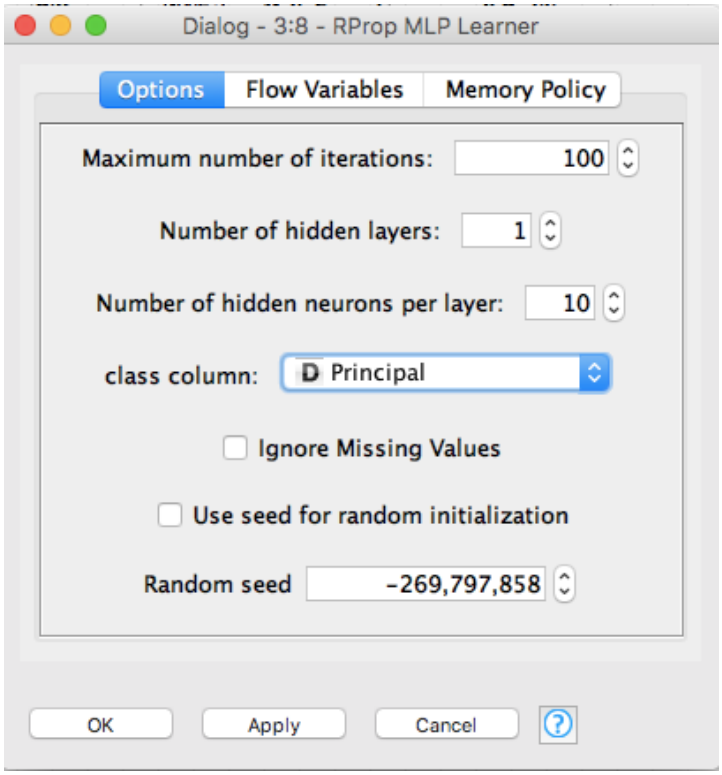
Şekil 12.4.3, veri setinin normalize edilmiş halini göstermektedir. Principal, terms, age, paid, date&time diff vb. Kolonlar normalize edilmiş olduğu görülebilir.

Verinin öğrenme (training) bölümü ve testi için RProp MLP learner ve multiLayer Percetron Predictor ve bunların başarı oranını ölçmek için de x-partitioner ve X-aggregator kullanılacaktır.



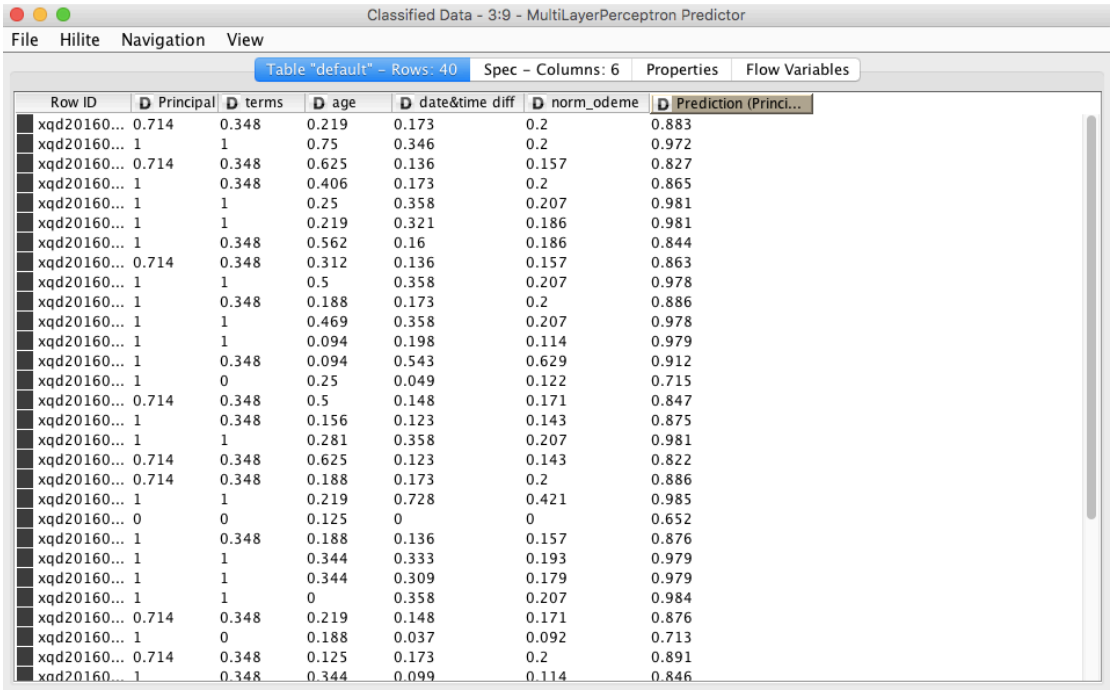
Şekil 12.4.4

Şekil 12.4.4, amaca ulaşmak için kullanılacak tüm operatörleri ve bağlantılarını göstermektedir.



Şekil 12.4.5

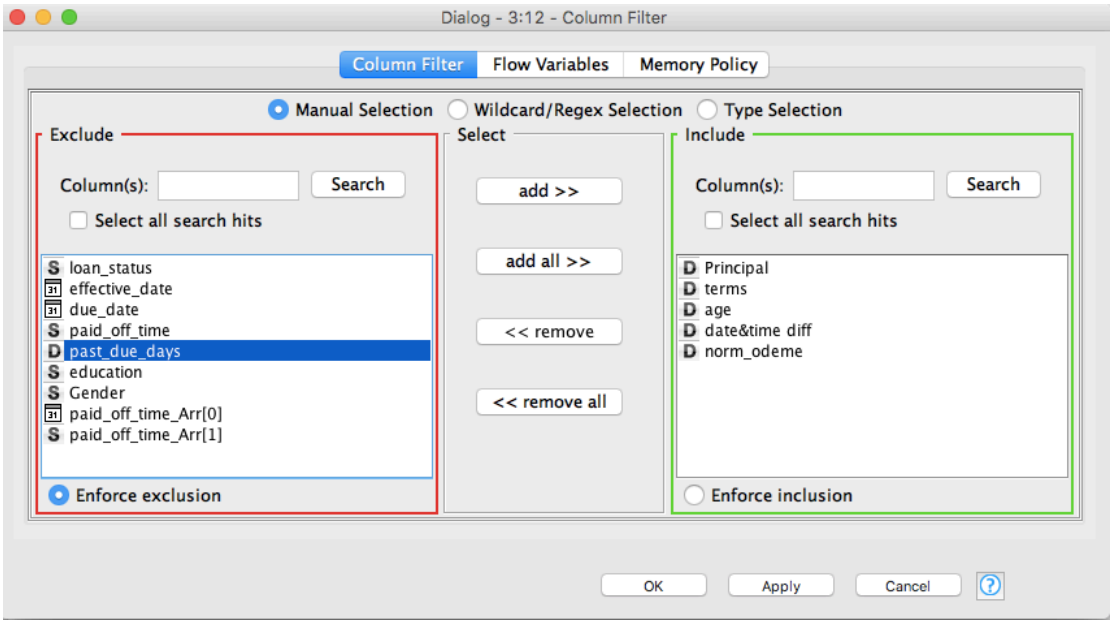
Şekil 12.4.5, RProp MLP Learner operatörünün configure penceresini göstermektedir. Burada class column olarak principal seçilmiştir. Yani label (bulunmak istenilen) olarak predict edilecek kolon principal olduğu anlaşılabilir.



Row ID	Principal	terms	age	date&time diff	norm_odeme	Prediction (Princi...
xqd20160...	0.714	0.348	0.219	0.173	0.2	0.883
xqd20160...	1	1	0.75	0.346	0.2	0.972
xqd20160...	0.714	0.348	0.625	0.136	0.157	0.827
xqd20160...	1	0.348	0.406	0.173	0.2	0.865
xqd20160...	1	1	0.25	0.358	0.207	0.981
xqd20160...	1	1	0.219	0.321	0.186	0.981
xqd20160...	1	0.348	0.562	0.16	0.186	0.844
xqd20160...	0.714	0.348	0.312	0.136	0.157	0.863
xqd20160...	1	1	0.5	0.358	0.207	0.978
xqd20160...	1	0.348	0.188	0.173	0.2	0.886
xqd20160...	1	1	0.469	0.358	0.207	0.978
xqd20160...	1	1	0.094	0.198	0.114	0.979
xqd20160...	1	0.348	0.094	0.543	0.629	0.912
xqd20160...	1	0	0.25	0.049	0.122	0.715
xqd20160...	0.714	0.348	0.5	0.148	0.171	0.847
xqd20160...	1	0.348	0.156	0.123	0.143	0.875
xqd20160...	1	1	0.281	0.358	0.207	0.981
xqd20160...	0.714	0.348	0.625	0.123	0.143	0.822
xqd20160...	0.714	0.348	0.188	0.173	0.2	0.886
xqd20160...	1	1	0.219	0.728	0.421	0.985
xqd20160...	0	0	0.125	0	0	0.652
xqd20160...	1	0.348	0.188	0.136	0.157	0.876
xqd20160...	1	1	0.344	0.333	0.193	0.979
xqd20160...	1	1	0.344	0.309	0.179	0.979
xqd20160...	1	1	0	0.358	0.207	0.984
xqd20160...	0.714	0.348	0.219	0.148	0.171	0.876
xqd20160...	1	0	0.188	0.037	0.092	0.713
xqd20160...	0.714	0.348	0.125	0.173	0.2	0.891
xqd20160...	1	0.348	0.344	0.099	0.114	0.846

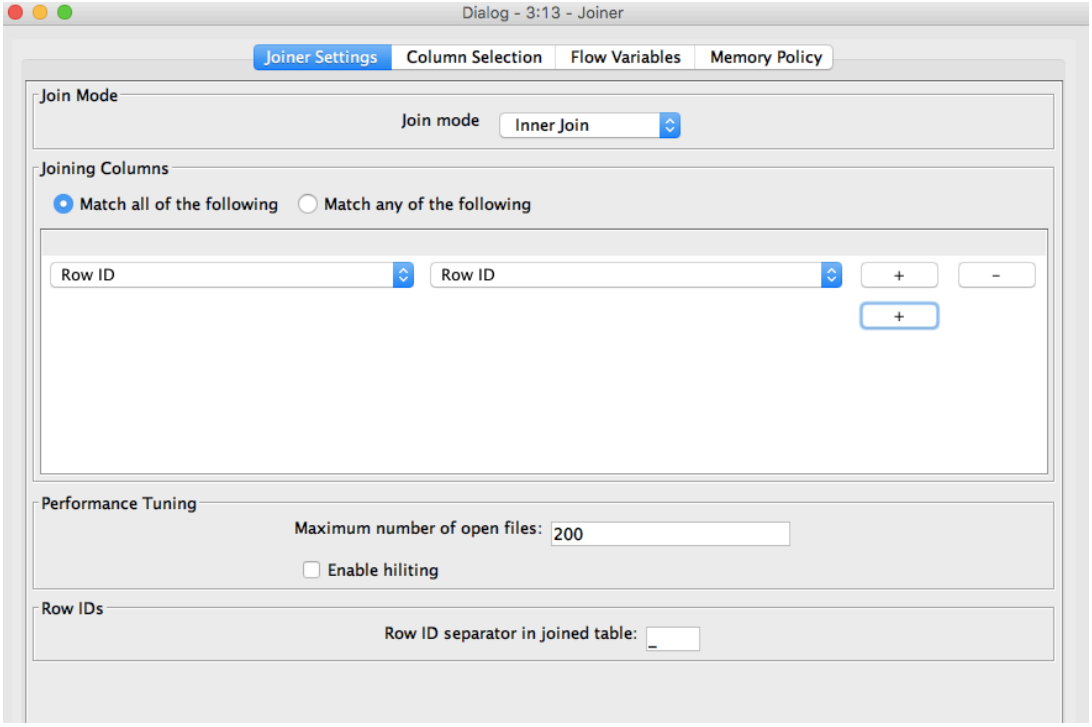
Şekil 12.4.6

Şekil 12.4.6, program çalıştırıldıktan sonra elde edilen multilayer perceptron predictor 'ın classified data penceresini göstermektedir. Normalize edilmiş fakat maximum vade tarihi ile çarpılmamış prediction principal değerleri son kolona eklenmiştir.



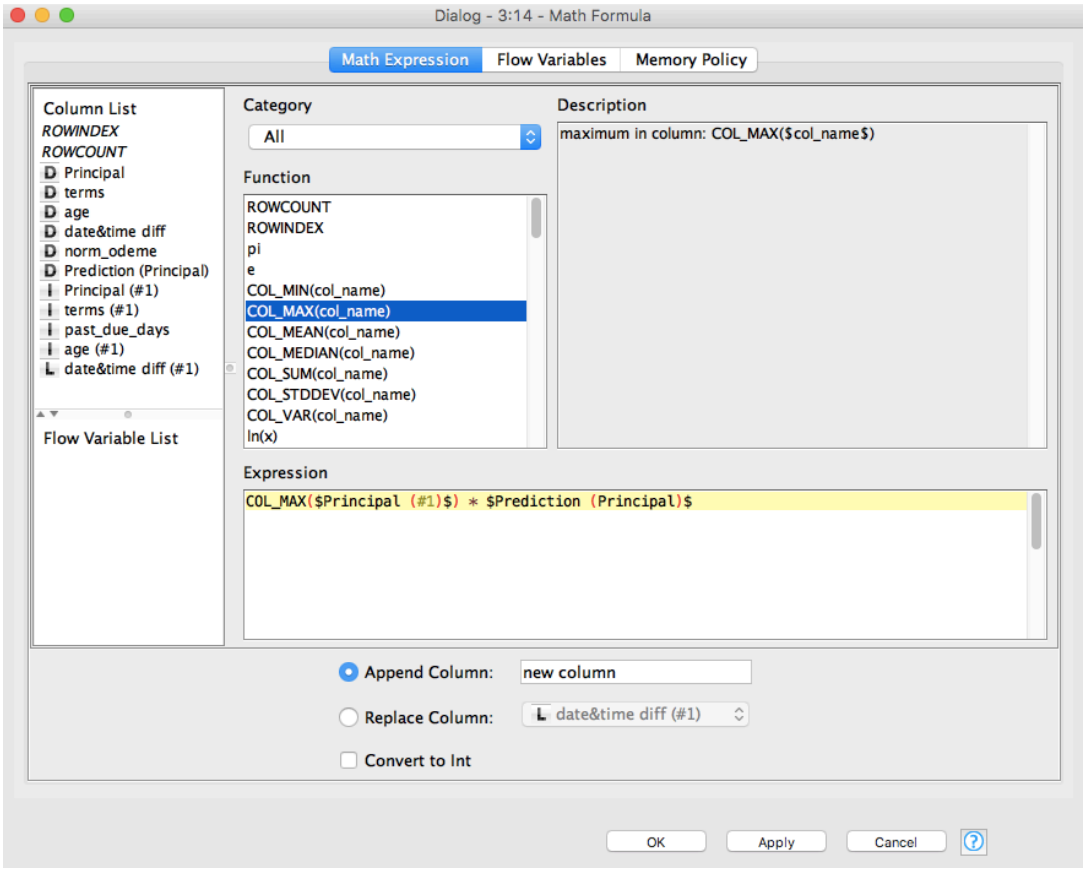
Şekil 12.4.7

Şekil 12.4.7, column filter operatörünün configure penceresinde seçilen ve seçilmeyen kolonları göstermektedir. Neural network kullanıldığı için seçilen kolonların double özellikte olması gerekmektedir. Ayrıca eksik değerin (missing value) olmaması gerektiği için Past_due_days kolonu işleme alınmamıştır.



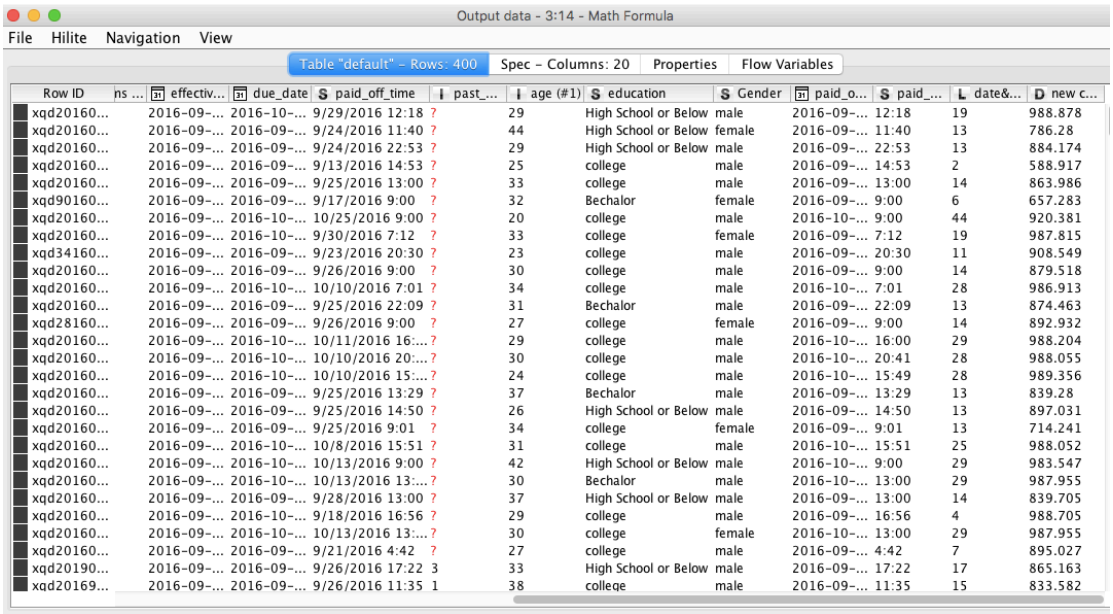
Şekil 12.4.8

Şekil 12.4.8, join operatörünün configure penceresini göstermektedir. Normalize edilmemiş veri ile normalize edilmiş ve öğrenmesi testi yapılmış veri setinin inner join ile birleştirilmesi için bu operatör kullanılmıştır.



Şekil 12.4.9

Şekil 12.4.9, math formula operatörünün configure penceresinde yazılan işlemi göstermektedir. Principal (normalize edilmemiş değerlerinki) max değeri ile predict edilen principal değerinin çarpımı istenilen sonucu verecektir.



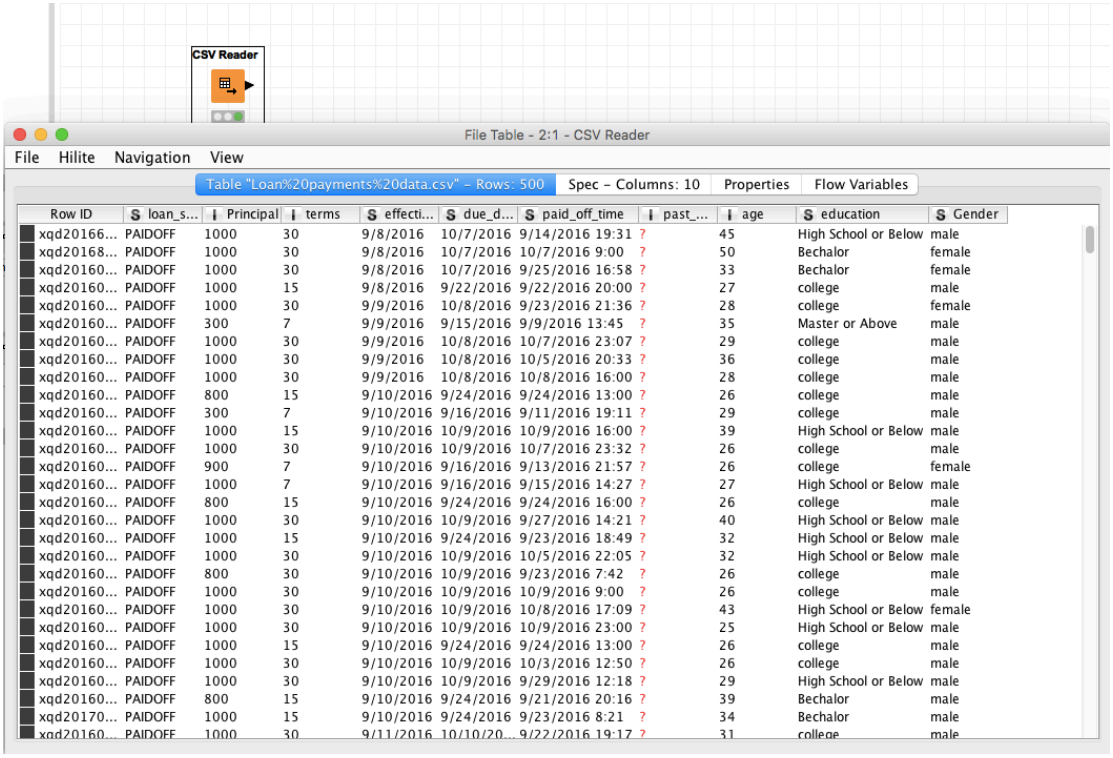
Row ID	ns ...	effectiv...	due_date	paid_off_time	past_...	age (#1)	education	Gender	paid_o...	paid_...	date&...	new c...
xqd20160...		2016-09-...	2016-10-...	9/29/2016 12:18	?	29	High School or Below	male	2016-09-...	12:18	19	988.878
xqd20160...		2016-09-...	2016-09-...	9/24/2016 11:40	?	44	High School or Below	female	2016-09-...	11:40	13	786.28
xqd20160...		2016-09-...	2016-09-...	9/24/2016 22:53	?	29	High School or Below	male	2016-09-...	22:53	13	884.174
xqd20160...		2016-09-...	2016-09-...	9/13/2016 14:53	?	25	college	male	2016-09-...	14:53	2	588.917
xqd20160...		2016-09-...	2016-09-...	9/25/2016 13:00	?	33	college	male	2016-09-...	13:00	14	863.986
xqd90160...		2016-09-...	2016-09-...	9/17/2016 9:00	?	32	Bechalor	female	2016-09-...	9:00	6	657.283
xqd20160...		2016-09-...	2016-10-...	10/25/2016 9:00	?	20	college	male	2016-10-...	9:00	44	920.381
xqd20160...		2016-09-...	2016-10-...	9/30/2016 7:12	?	33	college	female	2016-09-...	7:12	19	987.815
xqd34160...		2016-09-...	2016-09-...	9/23/2016 20:30	?	23	college	male	2016-09-...	20:30	11	908.549
xqd20160...		2016-09-...	2016-09-...	9/26/2016 9:00	?	30	college	male	2016-09-...	9:00	14	879.518
xqd20160...		2016-09-...	2016-10-...	10/10/2016 7:01	?	34	college	male	2016-10-...	7:01	28	986.913
xqd20160...		2016-09-...	2016-09-...	9/25/2016 22:09	?	31	Bechalor	male	2016-09-...	22:09	13	874.463
xqd28160...		2016-09-...	2016-09-...	9/26/2016 9:00	?	27	college	female	2016-09-...	9:00	14	892.932
xqd20160...		2016-09-...	2016-10-...	10/11/2016 16:...	?	29	college	male	2016-10-...	16:00	29	988.204
xqd20160...		2016-09-...	2016-10-...	10/10/2016 20:...	?	30	college	male	2016-10-...	20:41	28	988.055
xqd20160...		2016-09-...	2016-10-...	10/10/2016 15:...	?	24	college	male	2016-10-...	15:49	28	989.356
xqd20160...		2016-09-...	2016-09-...	9/25/2016 13:29	?	37	Bechalor	male	2016-09-...	13:29	13	839.28
xqd20160...		2016-09-...	2016-09-...	9/25/2016 14:50	?	26	High School or Below	male	2016-09-...	14:50	13	897.031
xqd20160...		2016-09-...	2016-09-...	9/25/2016 9:01	?	34	college	female	2016-09-...	9:01	13	714.241
xqd20160...		2016-09-...	2016-10-...	10/8/2016 15:51	?	31	college	male	2016-10-...	15:51	25	988.052
xqd20160...		2016-09-...	2016-10-...	10/13/2016 9:00	?	42	High School or Below	male	2016-10-...	9:00	29	983.547
xqd20160...		2016-09-...	2016-10-...	10/13/2016 13:...	?	30	Bechalor	male	2016-10-...	13:00	29	987.955
xqd20160...		2016-09-...	2016-09-...	9/28/2016 13:00	?	37	High School or Below	male	2016-09-...	13:00	14	839.705
xqd20160...		2016-09-...	2016-10-...	9/18/2016 16:56	?	29	college	male	2016-09-...	16:56	4	988.705
xqd20160...		2016-09-...	2016-10-...	10/13/2016 13:...	?	30	college	female	2016-10-...	13:00	29	987.955
xqd20160...		2016-09-...	2016-09-...	9/21/2016 4:42	?	27	college	male	2016-09-...	4:42	7	895.027
xqd20190...		2016-09-...	2016-09-...	9/26/2016 17:22	3	33	High School or Below	male	2016-09-...	17:22	17	865.163
xqd20169...		2016-09-...	2016-09-...	9/26/2016 11:35	1	38	college	male	2016-09-...	11:35	15	833.582

Şekil 12.4.10

Şekil 12.4.10, output date'yı göstermektedir. İstenilen sonuç kolonu new column ismiyle pelirtilen kolondur. Örneğin male (erkek), collage mezunu, yaşı 25 olduğu bilinen birine 588 TL borç verilebileceği, female (kadın), barchalor (üniversite mezunu), 32 yaşında olduğu bilinen birine de 657 TL verileceği öngörülmektedir.

12.5. Müşteri Segmentasyon (Customer Segmentation)

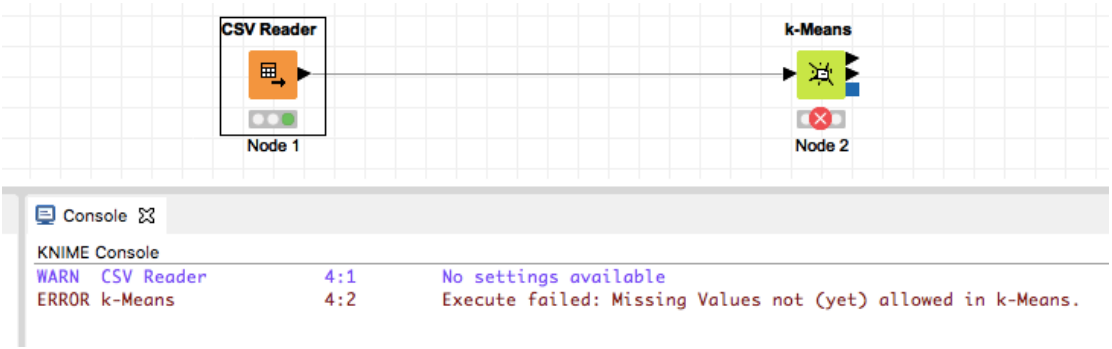
Bu bölümde, 1. bölümde indirilen veri kümesi (loan_status) ile müşteri segmentasyonu yapılması gösterilecektir. Segmentasyonda yani daha doğrusu clustering'te önemli olan unlabed çalışılmasıdır. Yani herhangi bir etikete göre sınıflandırma yapılmamasıdır.



Row ID	loan_s...	Principal	terms	effecti...	due_d...	paid_off_time	past_...	age	education	Gender
xqd20166...	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/14/2016 19:31	?	45	High School or Below	male
xqd20168...	PAIDOFF	1000	30	9/8/2016	10/7/2016	10/7/2016 9:00	?	50	Bechalor	female
xqd20160...	PAIDOFF	1000	30	9/8/2016	10/7/2016	9/25/2016 16:58	?	33	Bechalor	female
xqd20160...	PAIDOFF	1000	15	9/8/2016	9/22/2016	9/22/2016 20:00	?	27	college	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	9/23/2016 21:36	?	28	college	female
xqd20160...	PAIDOFF	300	7	9/9/2016	9/15/2016	9/9/2016 13:45	?	35	Master or Above	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/7/2016 23:07	?	29	college	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/5/2016 20:33	?	36	college	male
xqd20160...	PAIDOFF	1000	30	9/9/2016	10/8/2016	10/8/2016 16:00	?	28	college	male
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 13:00	?	26	college	male
xqd20160...	PAIDOFF	300	7	9/10/2016	9/16/2016	9/11/2016 19:11	?	29	college	male
xqd20160...	PAIDOFF	1000	15	9/10/2016	10/9/2016	10/9/2016 16:00	?	39	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/7/2016 23:32	?	26	college	male
xqd20160...	PAIDOFF	900	7	9/10/2016	9/16/2016	9/13/2016 21:57	?	26	college	female
xqd20160...	PAIDOFF	1000	7	9/10/2016	9/16/2016	9/15/2016 14:27	?	27	High School or Below	male
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/24/2016 16:00	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	9/27/2016 14:21	?	40	High School or Below	male
xqd20160...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/23/2016 18:49	?	32	High School or Below	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/5/2016 22:05	?	32	High School or Below	male
xqd20160...	PAIDOFF	800	30	9/10/2016	10/9/2016	9/23/2016 7:42	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/9/2016 9:00	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/8/2016 17:09	?	43	High School or Below	female
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/9/2016 23:00	?	25	High School or Below	male
xqd20160...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/24/2016 13:00	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	10/3/2016 12:50	?	26	college	male
xqd20160...	PAIDOFF	1000	30	9/10/2016	10/9/2016	9/29/2016 12:18	?	29	High School or Below	male
xqd20160...	PAIDOFF	800	15	9/10/2016	9/24/2016	9/21/2016 20:16	?	39	Bechalor	male
xqd20170...	PAIDOFF	1000	15	9/10/2016	9/24/2016	9/23/2016 8:21	?	34	Bechalor	male
xqd20160...	PAIDOFF	1000	30	9/11/2016	10/10/20...	9/22/2016 19:17	?	31	college	male

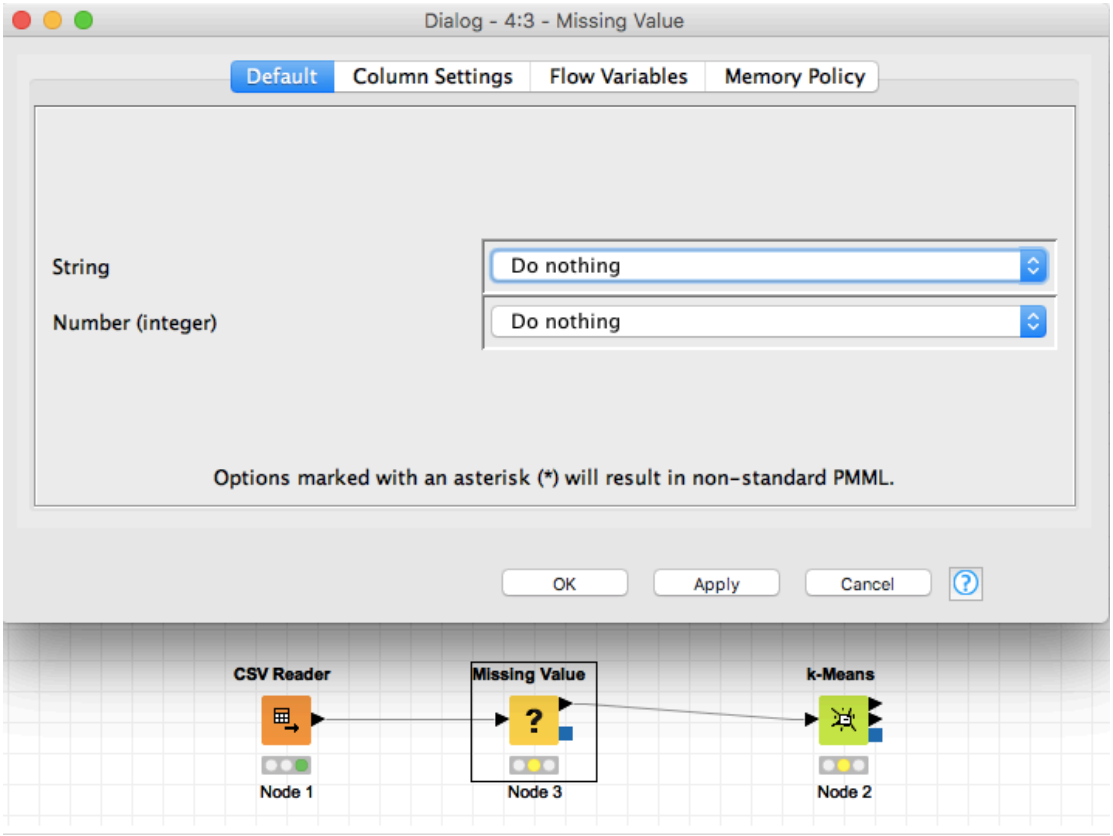
Şekil 12.5.1

Şekil 12.5.1 birinci bölümde kullanılan loan payments datasının sisteme cvs reader ile yüklenmesini ve veri seti içeriğini göstermektedir. Müşteriler sisteme verilerek makinenin kendisine göre segmente etmesi beklenmektedir. Daha önceden kullanılan k-means clustering operatörü burada örnek olması açısından kullanılacak.



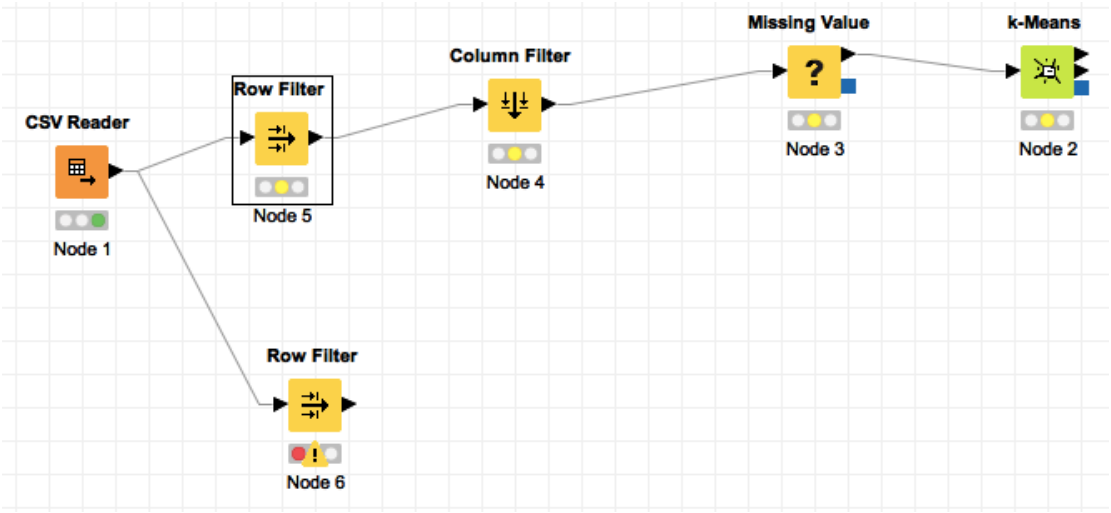
Şekil 12.5.2

Şekil 12.5.2'de k-means operatörünün sisteme eklenmesini göstermektedir. Sistem execute edildikten sonra error meydana gelmiştir. Nedeni şekilde de görüldüğü gibi veri seti içerisinde eksik verilerin olmasıdır. Bunları gidermek için birden fazla yol vardır.



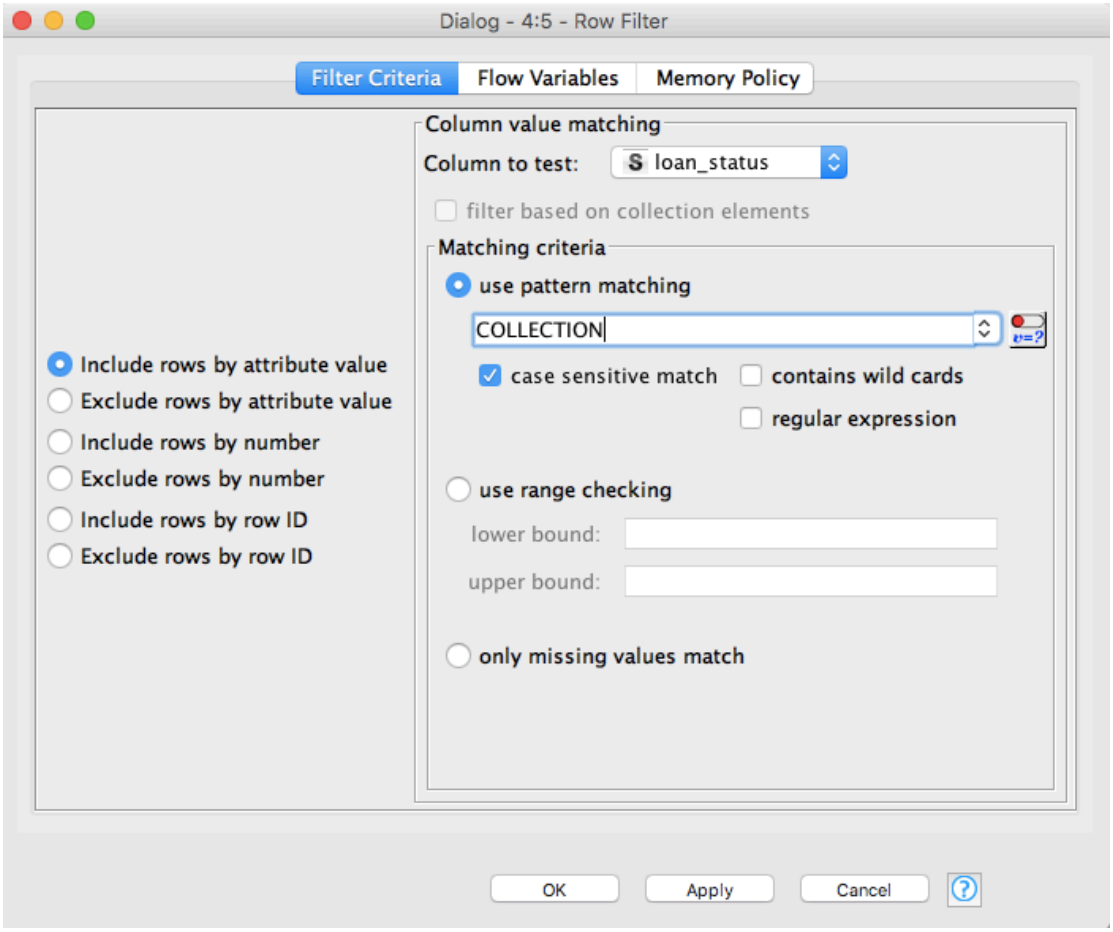
Şekil 12.5.3

Şekil 12.5.3 sisteme missing value operatörünün eklenmesini göstermektedir. Eksik verilerin giderilmesine yönelik string özellikte olanlar ve number özellikte olanlar için (integer özellikte olanlar) 2 çeşit veri tipi için de ayrımı vardır. String için; fix value (sabit bir değer), most frequent value (en çok tekrar edilen değer), next value (kendinden sonra gelen değer), previous value (önceki gelen değer), remove row (eksik verinin olduğu sıranın silinmesi) yöntemleri ile eksik veri giderilebilir. Fakat şuan tarih kolonu da string görünmekte ve bu yöntemlerle giderilmesi anlamlı olmaz. Örneğin paid_off_time yani ödeme yapılan tarih kolonunda, kişi henüz ödeme yapmadığı için eksik bilgiler var. Bunun için örneğin o kolonun alınmaması gibi bir çözüm düşünülebilir. Past_due_dates yani gecikmiş ödeme bilgisinin bulunduğu kolonda da eksik değerler var. Böyle bir durumda clustering'i ağaçlara bölünebilir. Ödeyenler kendi arasında segmente edilirken ödeme yapmayanlar kendi arasında segmente edilebilir.



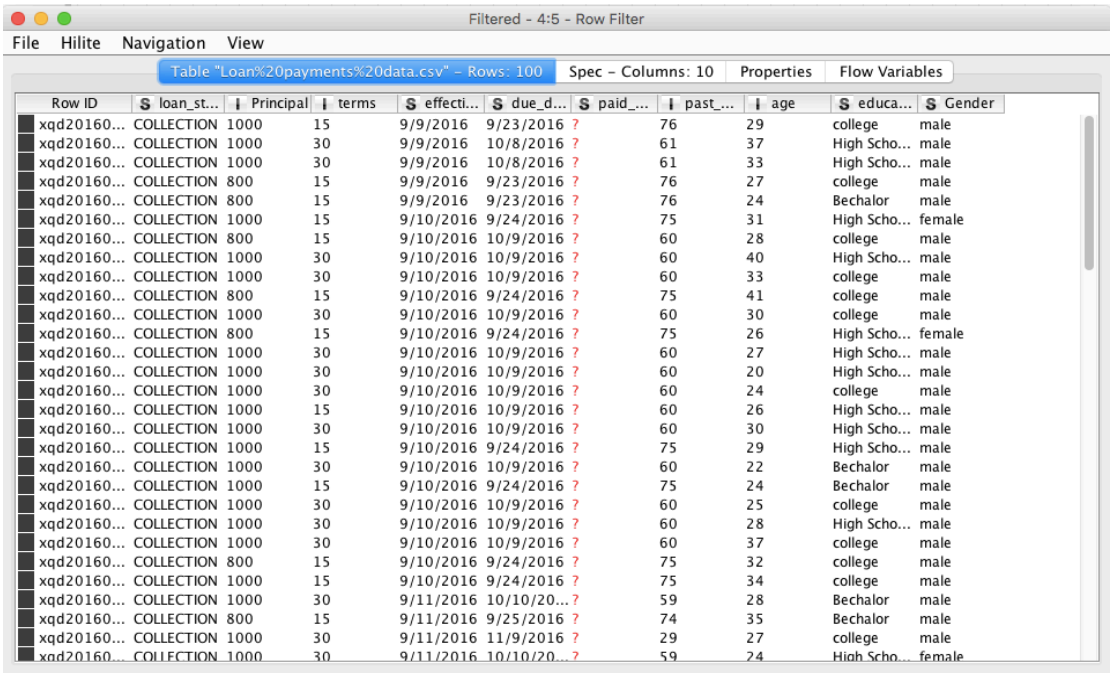
Şekil 12.5.4

Şekil 12.5.4, sisteme 2 row filter eklenmesini göstermektedir. Teki borcunu ödeyenler arasında segmente yapılabilmesi için, diğeri borcunu ödemeyenler arasında gruplama yapılabilmesi için dallandırılmıştır.



Şekil 12.5.5

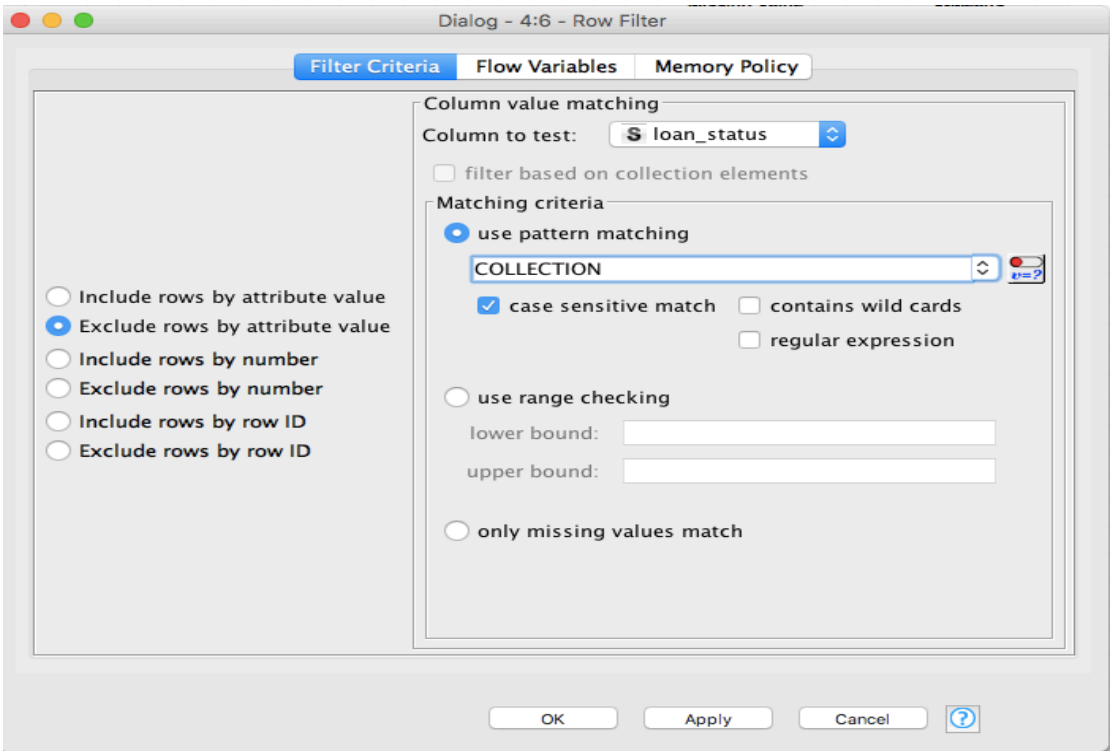
Şekil 12.5.5, üst kısımda olan row filter'in configure penceresini göstermektedir. Burada ödeme yapmamış olanların bilgileri filtrelenecektir.



Row ID	loan_st...	Principal	terms	effecti...	due_d...	paid_...	past_...	age	educa...	Gender
xqd20160...	COLLECTION	1000	15	9/9/2016	9/23/2016	?	76	29	college	male
xqd20160...	COLLECTION	1000	30	9/9/2016	10/8/2016	?	61	37	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/9/2016	10/8/2016	?	61	33	High Scho...	male
xqd20160...	COLLECTION	800	15	9/9/2016	9/23/2016	?	76	27	college	male
xqd20160...	COLLECTION	800	15	9/9/2016	9/23/2016	?	76	24	Bechalor	male
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	?	75	31	High Scho...	female
xqd20160...	COLLECTION	800	15	9/10/2016	10/9/2016	?	60	28	college	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	40	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	33	college	male
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	?	75	41	college	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	30	college	male
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	?	75	26	High Scho...	female
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	27	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	20	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	24	college	male
xqd20160...	COLLECTION	1000	15	9/10/2016	10/9/2016	?	60	26	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	30	High Scho...	male
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	?	75	29	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	22	Bechalor	male
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	?	75	24	Bechalor	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	25	college	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	28	High Scho...	male
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	?	60	37	college	male
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	?	75	32	college	male
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	?	75	34	college	male
xqd20160...	COLLECTION	1000	30	9/11/2016	10/10/20...	?	59	28	Bechalor	male
xqd20160...	COLLECTION	800	15	9/11/2016	9/25/2016	?	74	35	Bechalor	male
xqd20160...	COLLECTION	1000	30	9/11/2016	11/9/2016	?	29	27	college	male
xqd20160...	COLLECTION	1000	30	9/11/2016	10/10/20...	?	59	24	High Scho...	female

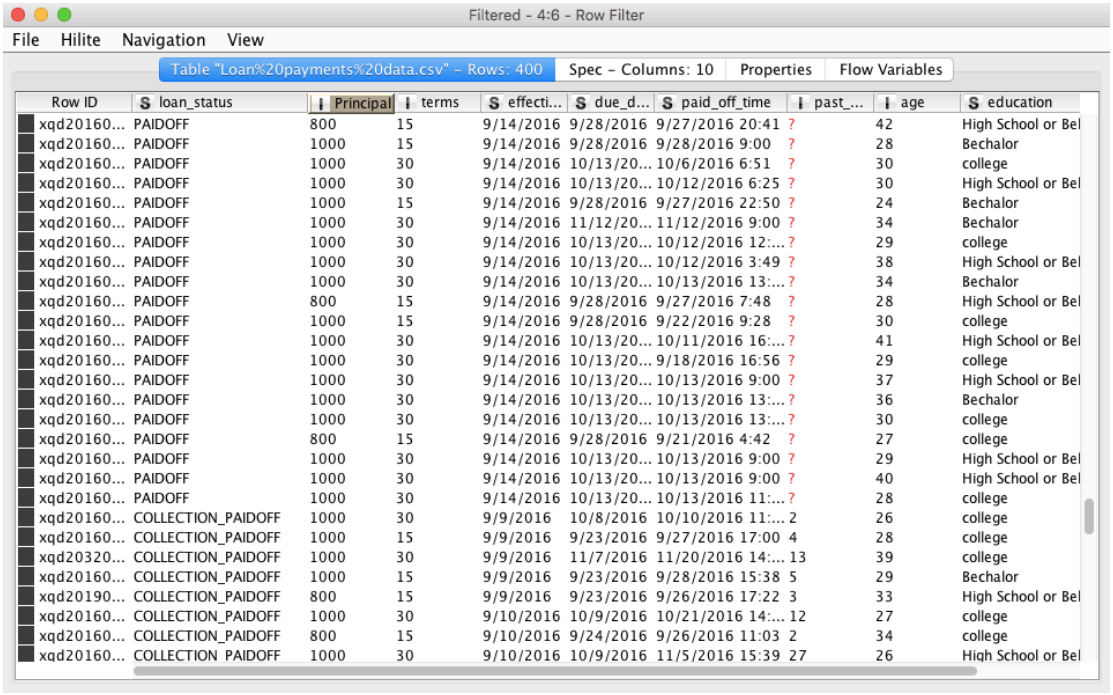
Şekil 12.5.6

Şekil 12.5.6, bir önceki şekil çalıştırıldıktan sonra elde edilen yani ödeme yapmamış olanlara göre filtrelenenlerin sonucunu göstermektedir.



Şekil 12.5.7

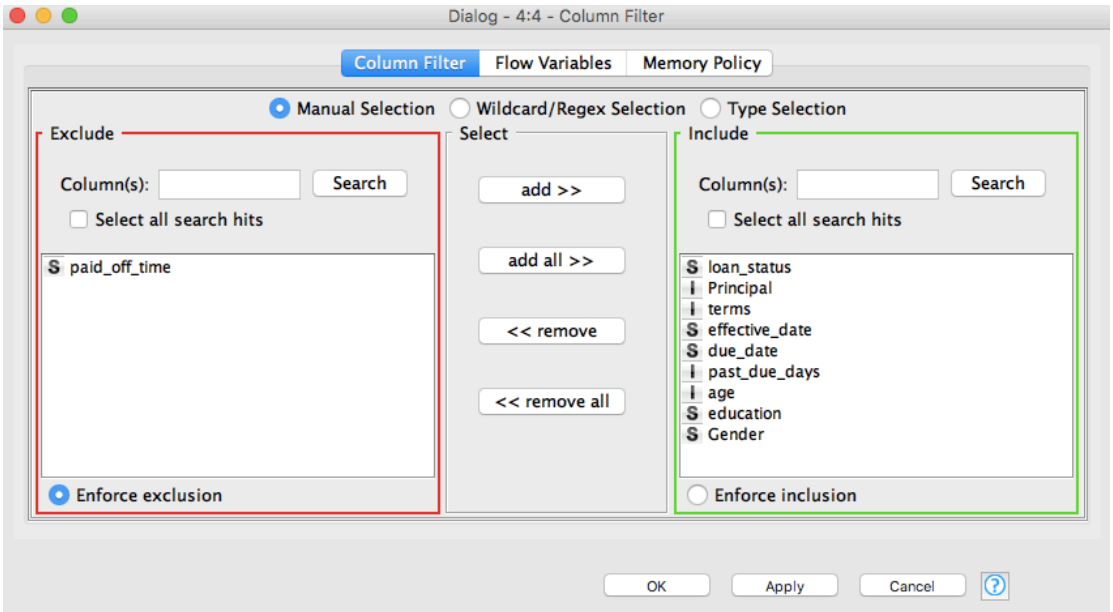
Şekil 12.5.7, ödeme yapmamış olanları dahil edilmediği geri kalanlarının alındığı verilerin seçimini göstermektedir (Şekil 12.5.4 deki altta kalan row filter'ın içeri). Ödeme sürecinde olanlar ve ödeme yapmış olanların filtrelenmesi beklenmektedir.



Row ID	loan_status	Principal	terms	effecti...	due_d...	paid_off_time	past_...	age	education
xqd20160...	PAIDOFF	800	15	9/14/2016	9/28/2016	9/27/2016 20:41	?	42	High School or Bel
xqd20160...	PAIDOFF	1000	15	9/14/2016	9/28/2016	9/28/2016 9:00	?	28	Bechalor
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/6/2016 6:51	?	30	college
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/12/2016 6:25	?	30	High School or Bel
xqd20160...	PAIDOFF	1000	15	9/14/2016	9/28/2016	9/27/2016 22:50	?	24	Bechalor
xqd20160...	PAIDOFF	1000	30	9/14/2016	11/12/20...	11/12/2016 9:00	?	34	Bechalor
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/12/2016 12:...	?	29	college
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/12/2016 3:49	?	38	High School or Bel
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 13:...	?	34	Bechalor
xqd20160...	PAIDOFF	800	15	9/14/2016	9/28/2016	9/27/2016 7:48	?	28	High School or Bel
xqd20160...	PAIDOFF	1000	15	9/14/2016	9/28/2016	9/22/2016 9:28	?	30	college
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/11/2016 16:...	?	41	High School or Bel
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	9/18/2016 16:56	?	29	college
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 9:00	?	37	High School or Bel
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 13:...	?	36	Bechalor
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 13:...	?	30	college
xqd20160...	PAIDOFF	800	15	9/14/2016	9/28/2016	9/21/2016 4:42	?	27	college
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 9:00	?	29	High School or Bel
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 9:00	?	40	High School or Bel
xqd20160...	PAIDOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 11:...	?	28	college
xqd20160...	COLLECTION_PAIDOFF	1000	30	9/9/2016	10/8/2016	10/10/2016 11:...	2	26	college
xqd20160...	COLLECTION_PAIDOFF	1000	15	9/9/2016	9/23/2016	9/27/2016 17:00	4	28	college
xqd20320...	COLLECTION_PAIDOFF	1000	30	9/9/2016	11/7/2016	11/20/2016 14:...	13	39	college
xqd20160...	COLLECTION_PAIDOFF	1000	15	9/9/2016	9/23/2016	9/28/2016 15:38	5	29	Bechalor
xqd20190...	COLLECTION_PAIDOFF	800	15	9/9/2016	9/23/2016	9/26/2016 17:22	3	33	High School or Bel
xqd20160...	COLLECTION_PAIDOFF	1000	30	9/10/2016	10/9/2016	10/21/2016 14:...	12	27	college
xqd20160...	COLLECTION_PAIDOFF	800	15	9/10/2016	9/24/2016	9/26/2016 11:03	2	34	college
xqd20160...	COLLECTION_PAIDOFF	1000	30	9/10/2016	10/9/2016	11/5/2016 15:39	27	26	High School or Bel

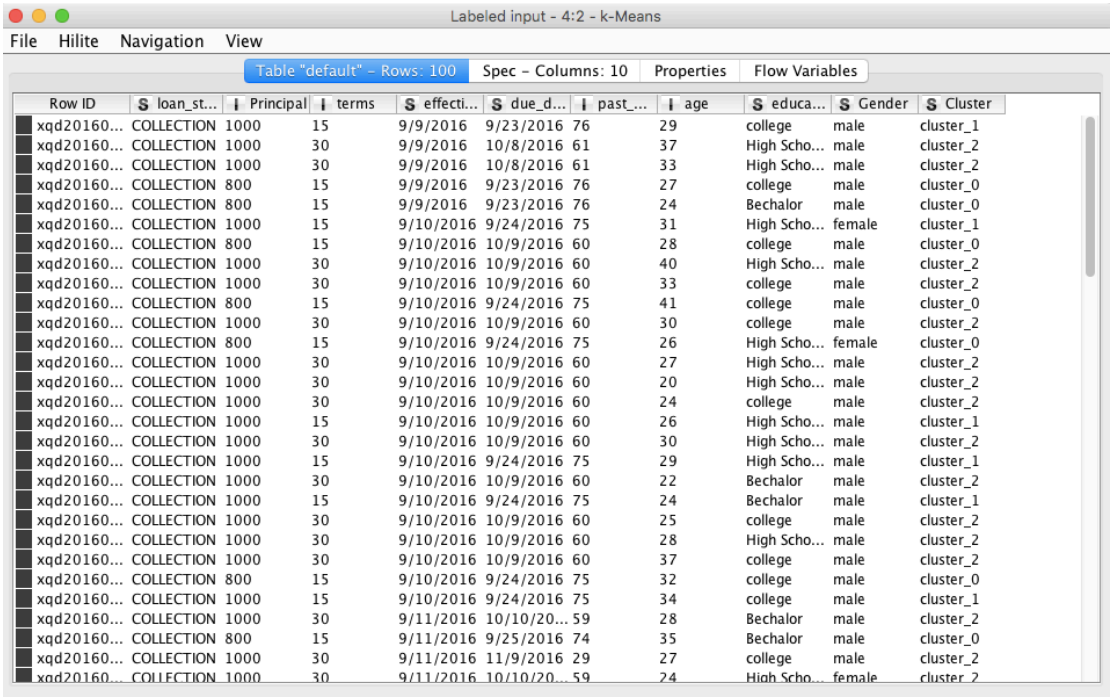
Şekil 12.5.8

Şekil 12.5.8, program çalıştırıldıktan sonra bir önceki şekilde yapılan işleme göre gelen filtrelenmiş veri setini göstermektedir. Burada henüz ödeme sürecinde olanlar ya da ödeme yapmamış olanlar yani borcunun ödemesi tam anlamıyla bitmemiş olanların bilgileri görülmektedir.



Şekil 12.5.9

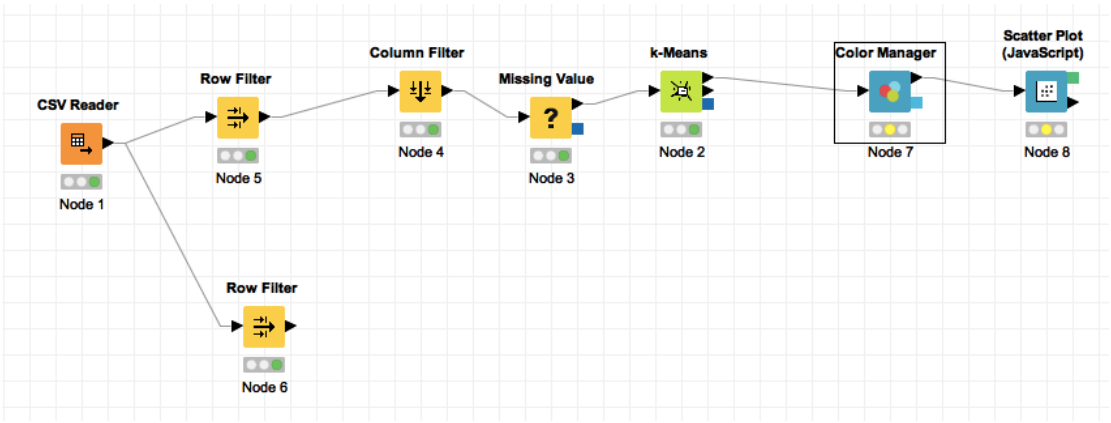
Şekil 12.5.9, column filter operatörünün configure içeriğini göstermektedir. K-means operatörü çalışırken eksik veri olan sıralar yüzünden hata vermişti bu yüzden eksik veri olan kolonun kaldırılması bu aşamada işlemleri bozmayacağı için sistemden filtrelenmiştir. Şekil 12.5.4'de görülen column filter, ödeme yapmamış müşterilerin olduğu dalda olduğu için paid_off_time zaten önemsiz bir kolon olacaktı.



Row ID	loan_st...	Principal	terms	effecti...	due_d...	past_...	age	educa...	Gender	Cluster
xqd20160...	COLLECTION	1000	15	9/9/2016	9/23/2016	76	29	college	male	cluster_1
xqd20160...	COLLECTION	1000	30	9/9/2016	10/8/2016	61	37	High Scho...	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/9/2016	10/8/2016	61	33	High Scho...	male	cluster_2
xqd20160...	COLLECTION	800	15	9/9/2016	9/23/2016	76	27	college	male	cluster_0
xqd20160...	COLLECTION	800	15	9/9/2016	9/23/2016	76	24	Bechalor	male	cluster_0
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	75	31	High Scho...	female	cluster_1
xqd20160...	COLLECTION	800	15	9/10/2016	10/9/2016	60	28	college	male	cluster_0
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	40	High Scho...	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	33	college	male	cluster_2
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	75	41	college	male	cluster_0
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	30	college	male	cluster_2
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	75	26	High Scho...	female	cluster_0
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	27	High Scho...	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	20	High Scho...	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	24	college	male	cluster_2
xqd20160...	COLLECTION	1000	15	9/10/2016	10/9/2016	60	26	High Scho...	male	cluster_1
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	30	High Scho...	male	cluster_2
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	75	29	High Scho...	male	cluster_1
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	22	Bechalor	male	cluster_2
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	75	24	Bechalor	male	cluster_1
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	25	college	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	28	High Scho...	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/10/2016	10/9/2016	60	37	college	male	cluster_2
xqd20160...	COLLECTION	800	15	9/10/2016	9/24/2016	75	32	college	male	cluster_0
xqd20160...	COLLECTION	1000	15	9/10/2016	9/24/2016	75	34	college	male	cluster_1
xqd20160...	COLLECTION	1000	30	9/11/2016	10/10/20...	59	28	Bechalor	male	cluster_2
xqd20160...	COLLECTION	800	15	9/11/2016	9/25/2016	74	35	Bechalor	male	cluster_0
xqd20160...	COLLECTION	1000	30	9/11/2016	11/9/2016	29	27	college	male	cluster_2
xqd20160...	COLLECTION	1000	30	9/11/2016	10/10/20...	59	24	High Scho...	female	cluster_2

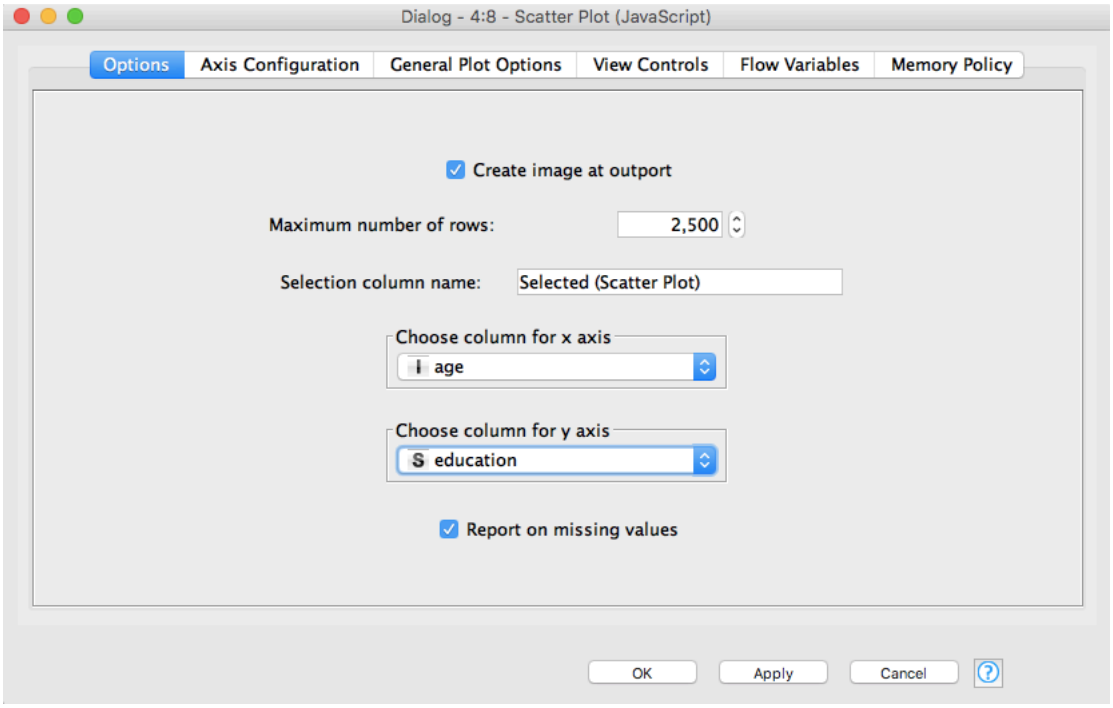
Şekil 12.5.10

Şekil 12.5.10, ödeme yapmamış müşterilerin segmente edilmesinden oluşan sonucu göstermektedir. Sistem bu müşteriler ile 3 cluster oluşturmuştur ve bunlar son kolonda görülmektedir.



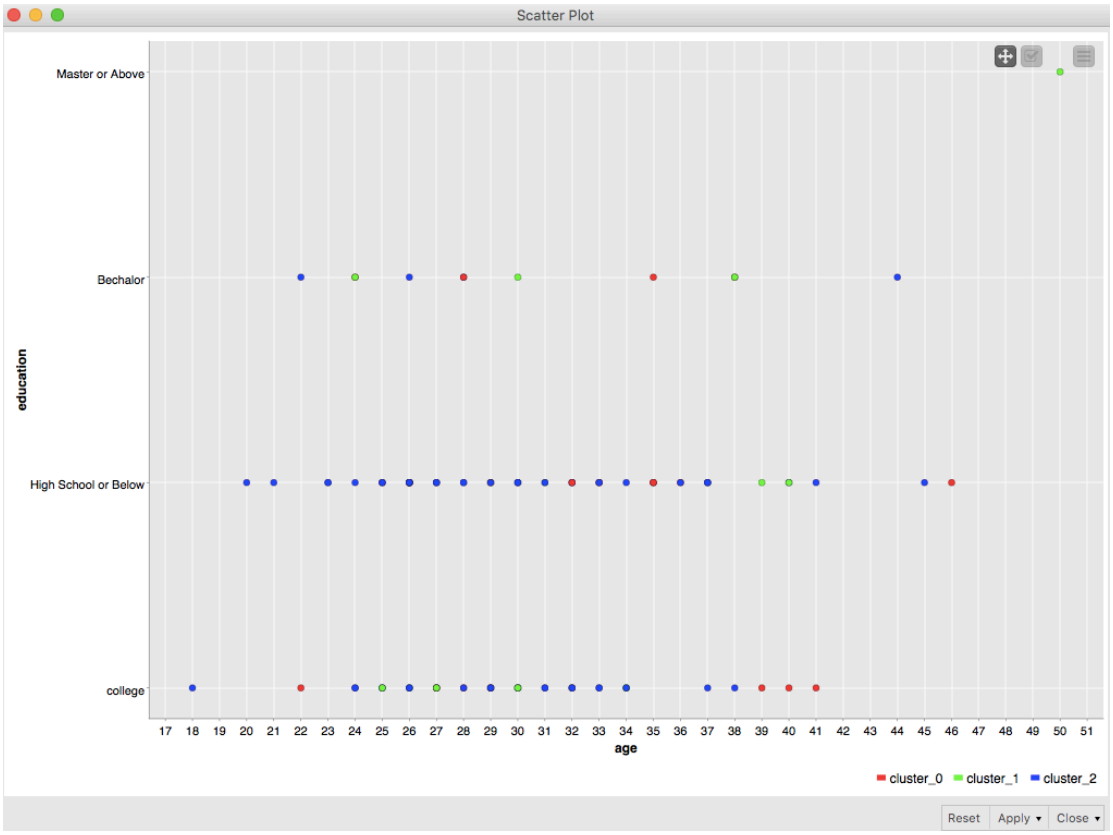
Şekil 12.5.11

Şekil 12.5.11, oluşturulan 3 cluster'ın daha anlamlı bir şekilde gösterilmesi için üçününde ayrı renklendirilmesi ve bunlarla bir grafik çizilerek gösterilmesi için sisteme color manager ve scatter plot operatörlerinin eklenmesini göstermektedir.



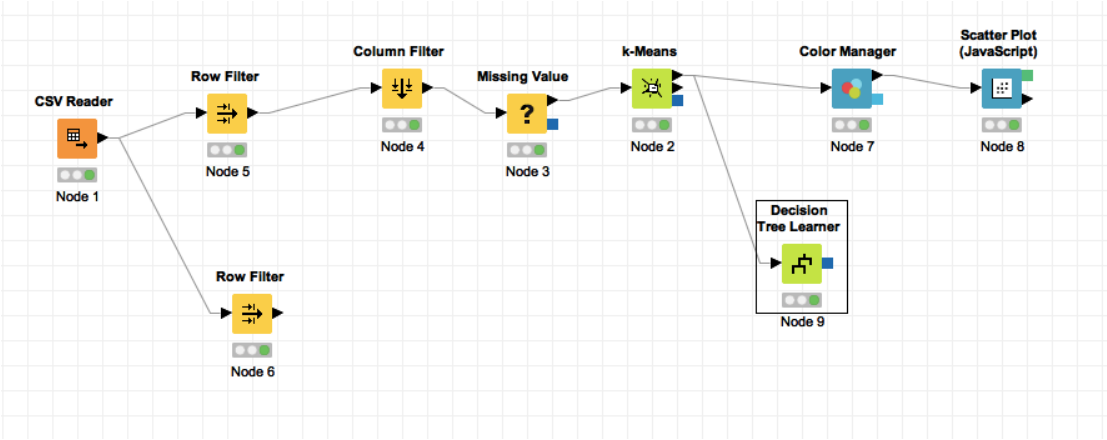
Şekil 12.5.12

Şekil 12.5.12, scatter plot configure içerisindeki yapılan ayarı göstermektedir. Grafikte yaş ve eğitim durumu arasındaki bağlantı grafiği oluşması beklenmektedir.



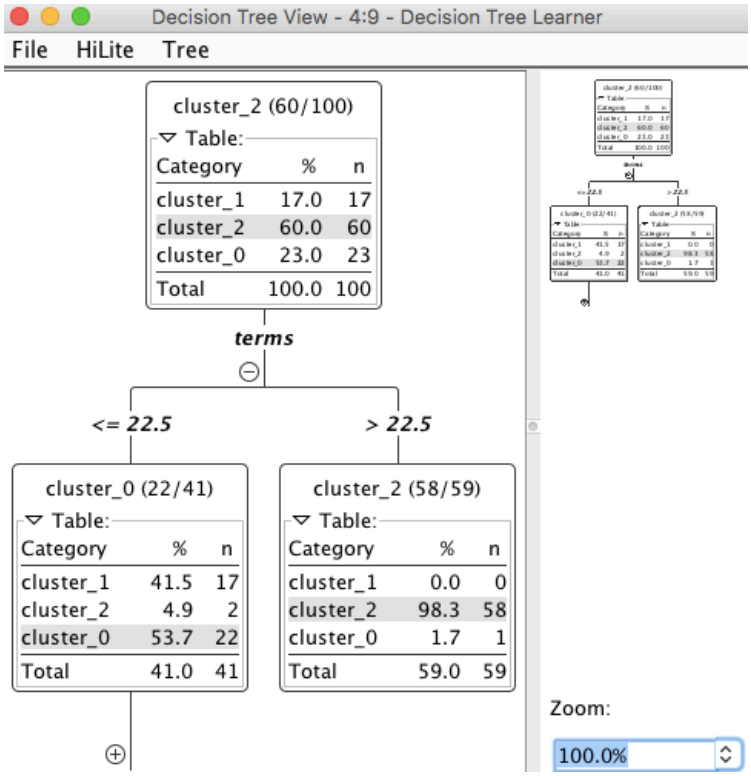
Şekil 12.5.13

Şekil 12.5.13, scatter plot'ta bir şeyler çizilmiş durumda fakat bu ilişkileri bulmak için tek tek tüm ilişkilere bakmak gerekmektedir. Bu yüzden decision tree bağlayıp neye göre ağaç dallanması yaptığına bakılabilir.



Şekil 12.5.14

Şekil 12.5.14, sisteme decision tree eklenmesini göstermektedir.

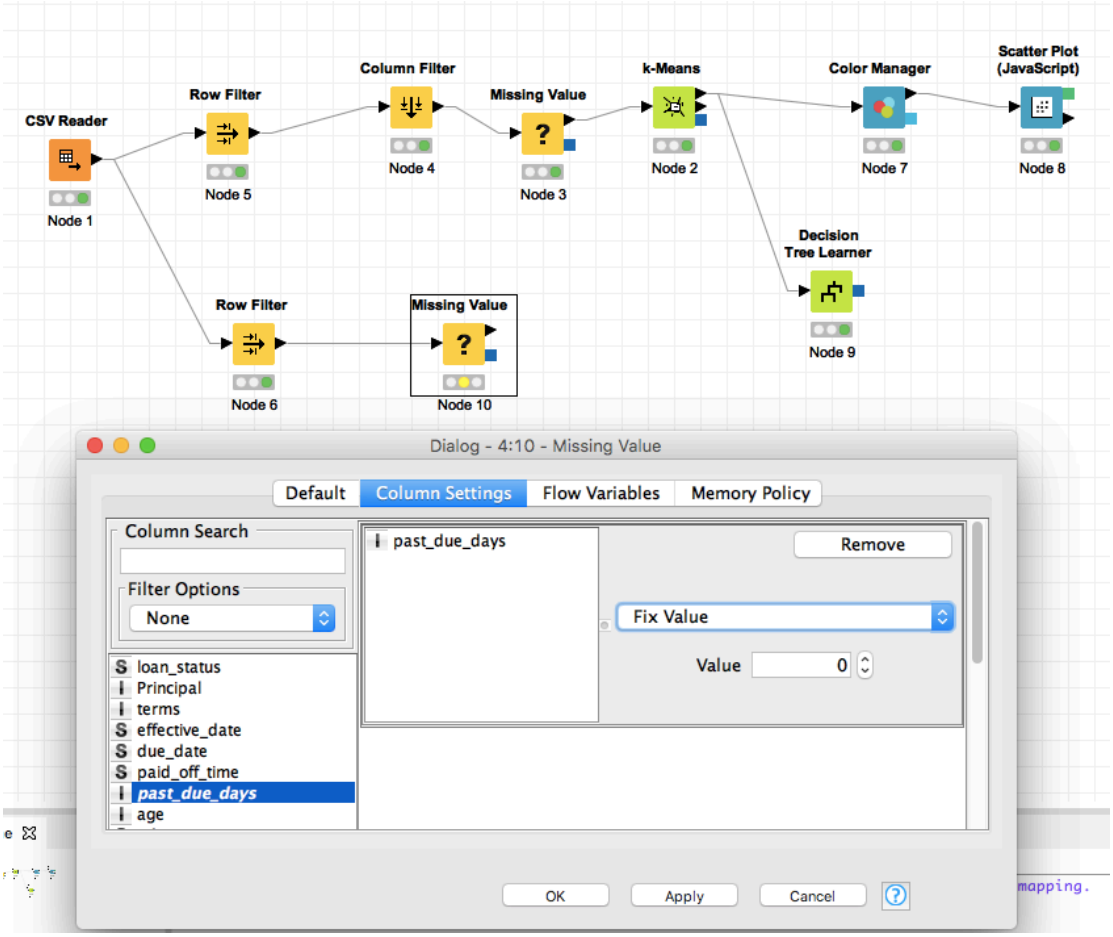


Şekil 12.5.15

Şekil 12.5.15, decision tree view penceresini göstermektedir. Decision tree'ye göre terms (borcu ne kaç gün sonra ödeyeceğine göre) 22.5 dan az olanlar ve çok olanlara göre dallanmıştır. Scatter plotta terms bilgisini kullanmak cluster'lar için daha ayırt

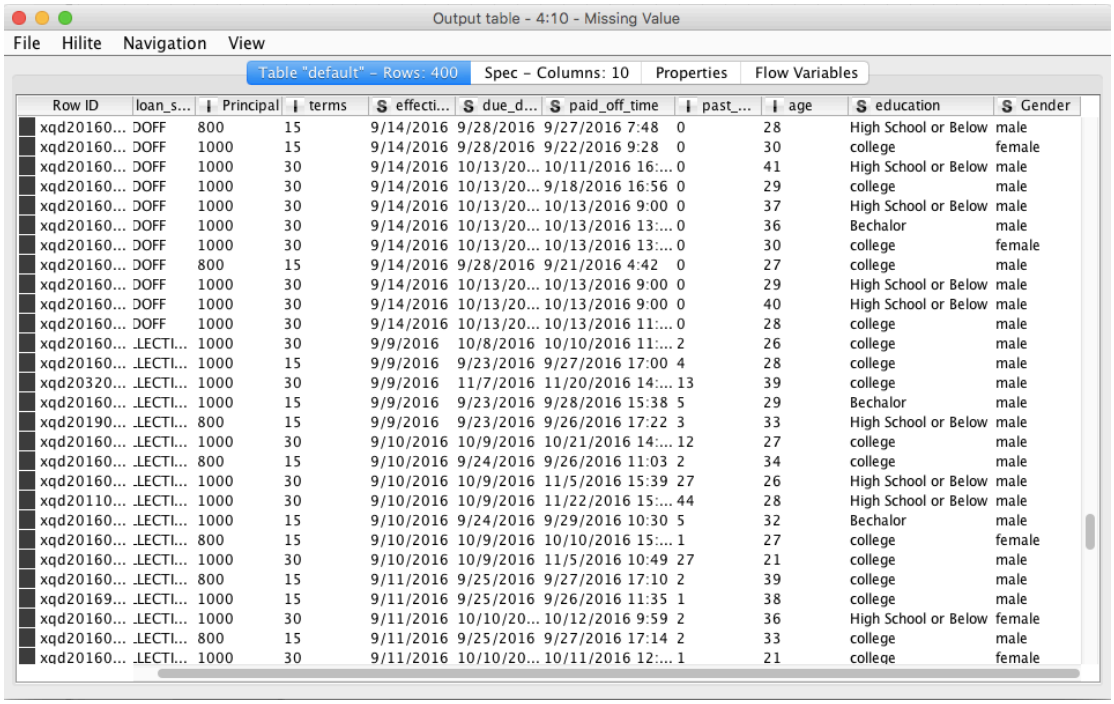
edici olacaktır. Örneğin cluster 2 için bir çok müşteri 22.5 günden sonra ödeme yapacağı çıkmıştır.

Şimdi ikinci row filter uygulananlar yani ödeme yapıyor olan ve yapmış olan müşteriler için cluster işlemi yapılacaktır.



Şekil 12.5.16

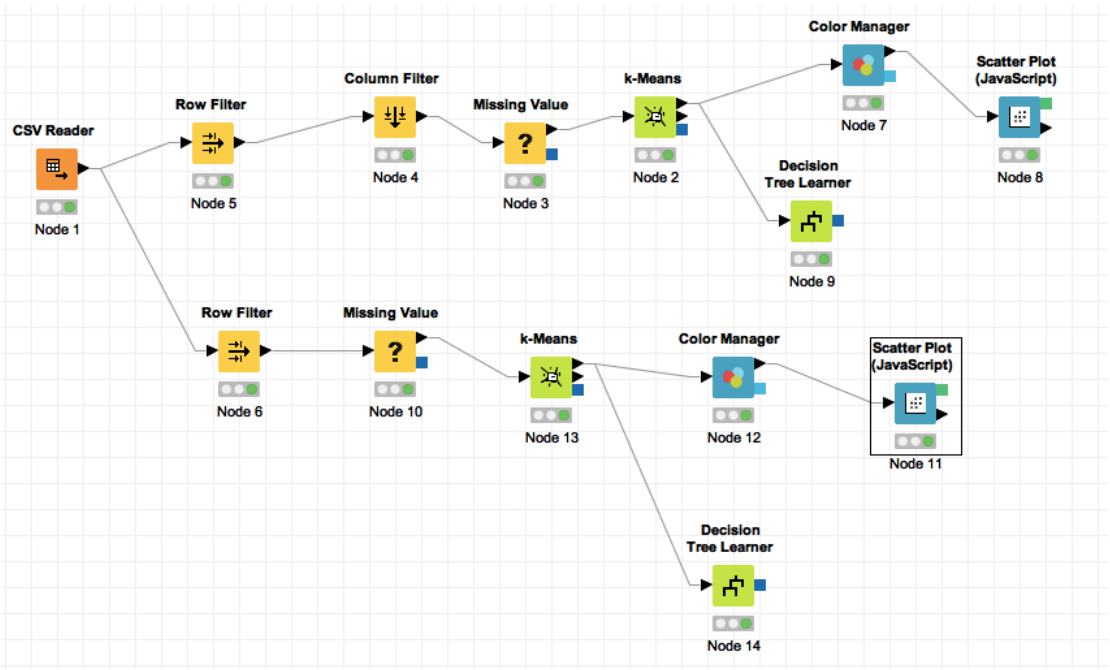
Şekil 12.5.16, sisteme missing value operatörünün eklenmesini ve configure penceresini göstermektedir. Şekil 12.5.6'da görüldüğü gibi past_due_days kolonunda ? işaretleri bulunmaktadır. Bunun anlamı o müşterilerin ödemelerini zamanında yaptığı yani ödemeyi geciktirmediğidir. Bu yüzden bu soru işaretli yerler 0 ile değiştirilebilir. Şekilde de görüldüğü gibi fix value olarak yani sabit değer (0) kolondaki bütün bu soru işaretli yerler ile yer değiştirecektir.



Row ID	loan_s...	Principal	terms	effecti...	due_d...	paid_off_time	past_...	age	education	Gender
xqd20160...	DOFF	800	15	9/14/2016	9/28/2016	9/27/2016 7:48	0	28	High School or Below	male
xqd20160...	DOFF	1000	15	9/14/2016	9/28/2016	9/22/2016 9:28	0	30	college	female
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/11/2016 16:...	0	41	High School or Below	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	9/18/2016 16:56	0	29	college	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 9:00	0	37	High School or Below	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 13:...	0	36	Bechalor	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 13:...	0	30	college	female
xqd20160...	DOFF	800	15	9/14/2016	9/28/2016	9/21/2016 4:42	0	27	college	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 9:00	0	29	High School or Below	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 9:00	0	40	High School or Below	male
xqd20160...	DOFF	1000	30	9/14/2016	10/13/20...	10/13/2016 11:...	0	28	college	male
xqd20160...	LECTI...	1000	30	9/9/2016	10/8/2016	10/10/2016 11:...	2	26	college	male
xqd20160...	LECTI...	1000	15	9/9/2016	9/23/2016	9/27/2016 17:00	4	28	college	male
xqd20320...	LECTI...	1000	30	9/9/2016	11/7/2016	11/20/2016 14:...	13	39	college	male
xqd20160...	LECTI...	1000	15	9/9/2016	9/23/2016	9/28/2016 15:38	5	29	Bechalor	male
xqd20190...	LECTI...	800	15	9/9/2016	9/23/2016	9/26/2016 17:22	3	33	High School or Below	male
xqd20160...	LECTI...	1000	30	9/10/2016	10/9/2016	10/21/2016 14:...	12	27	college	male
xqd20160...	LECTI...	800	15	9/10/2016	9/24/2016	9/26/2016 11:03	2	34	college	male
xqd20160...	LECTI...	1000	30	9/10/2016	10/9/2016	11/5/2016 15:39	27	26	High School or Below	male
xqd20110...	LECTI...	1000	30	9/10/2016	10/9/2016	11/22/2016 15:...	44	28	High School or Below	male
xqd20160...	LECTI...	1000	15	9/10/2016	9/24/2016	9/29/2016 10:30	5	32	Bechalor	male
xqd20160...	LECTI...	800	15	9/10/2016	10/9/2016	10/10/2016 15:...	1	27	college	female
xqd20160...	LECTI...	1000	30	9/10/2016	10/9/2016	11/5/2016 10:49	27	21	college	male
xqd20160...	LECTI...	800	15	9/11/2016	9/25/2016	9/27/2016 17:10	2	39	college	male
xqd20169...	LECTI...	1000	15	9/11/2016	9/25/2016	9/26/2016 11:35	1	38	college	male
xqd20160...	LECTI...	1000	30	9/11/2016	10/10/20...	10/12/2016 9:59	2	36	High School or Below	female
xqd20160...	LECTI...	800	15	9/11/2016	9/25/2016	9/27/2016 17:14	2	33	college	male
xqd20160...	LECTI...	1000	30	9/11/2016	10/10/20...	10/11/2016 12:...	1	21	college	female

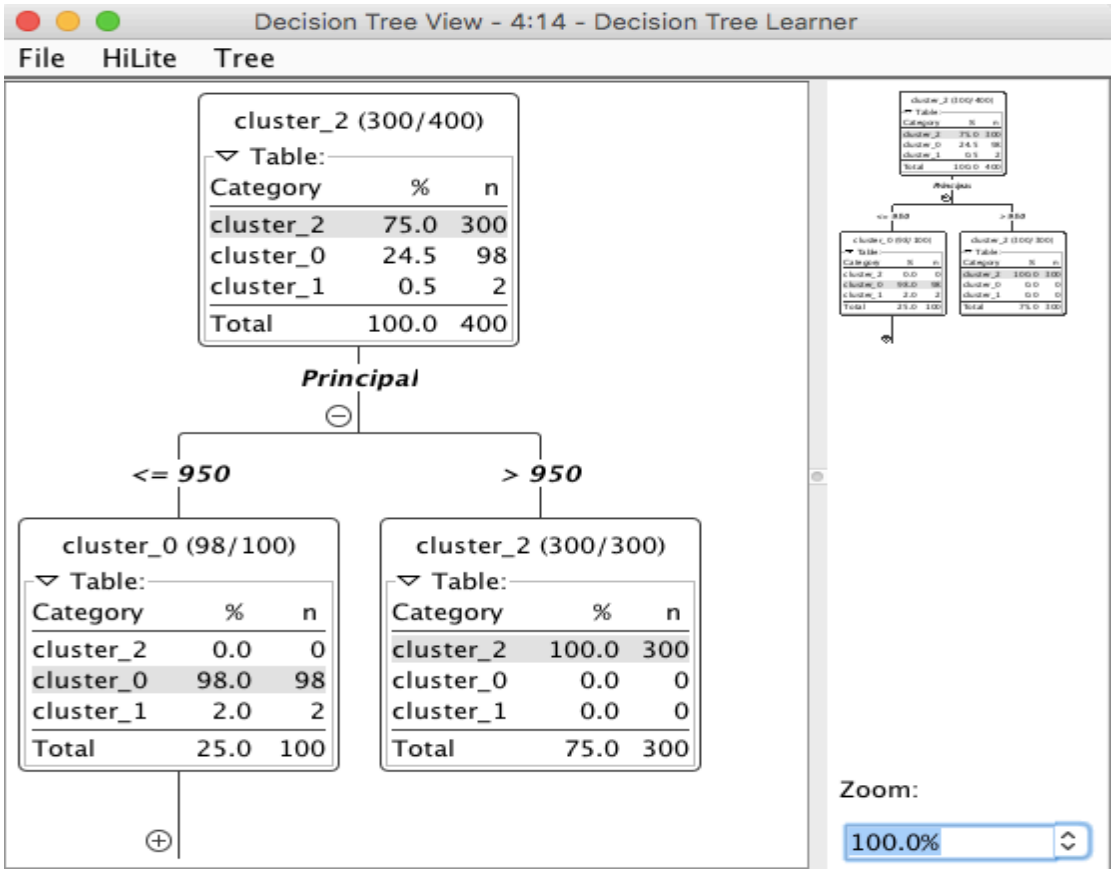
Şekil 12.5.17

Şekil 12.5.17, bir önceki işlemden sonra sistem çalıştırılmış halinin output table'ını göstermektedir. Görüldüğü üzere past_due_days kolonunda soru işaretli sıralar 0 değerini almıştır.



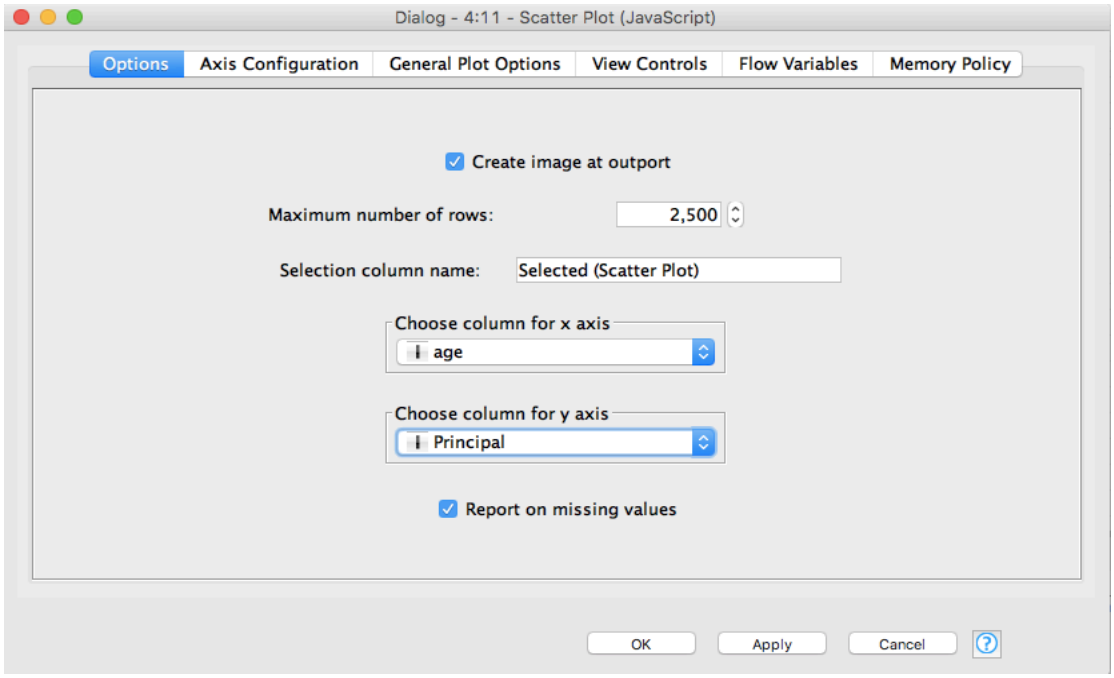
Şekil 12.5.18

Şekil 12.5.18, Daha önce yapılan işlemlerin aynısı ödeme yapmış müşteriler içinde yapılacak. Bu yüzden aynı operatörler şekilde görüldüğü gibi eklendi.



Şekil 12.5.19

Şekil 12.5.19, decision tree view penceresini göstermektedir. Sonuca göre en uygun bölünme principal bulunmuştur. Bu yüzden scatter plot'da principal kullanılması en uygun durumlardan biridir.



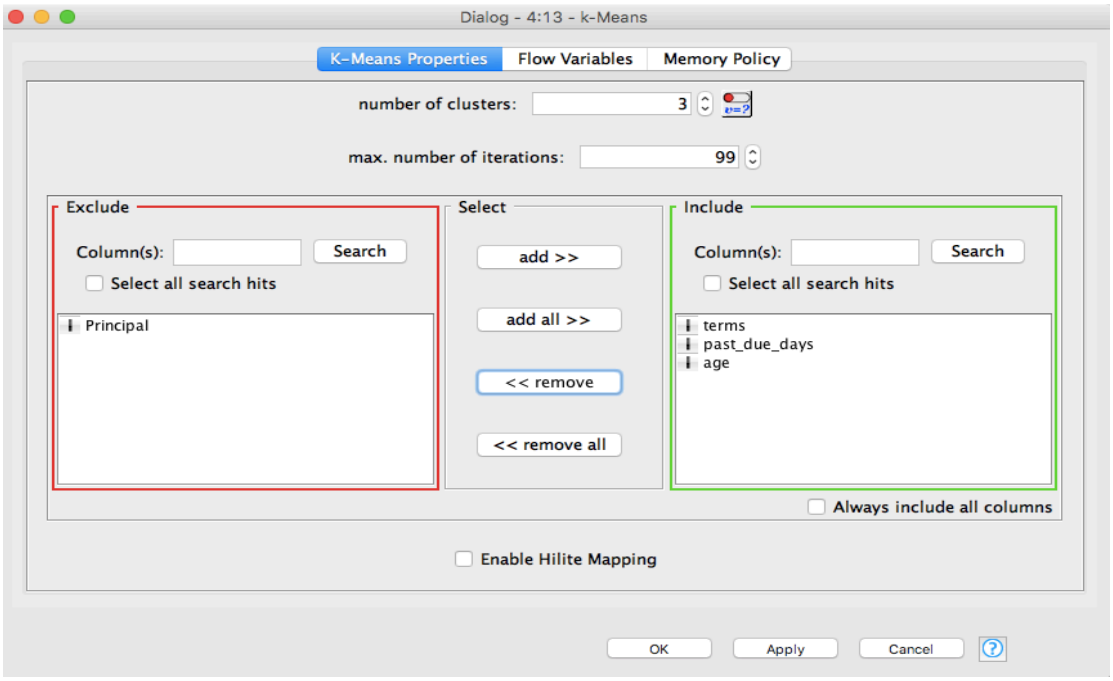
Şekil 12.5.20

Şekil 12.5.20, scatter plot configure penceresinden seçilen iki kolonu göstermektedir.

Şekil 12.5.21

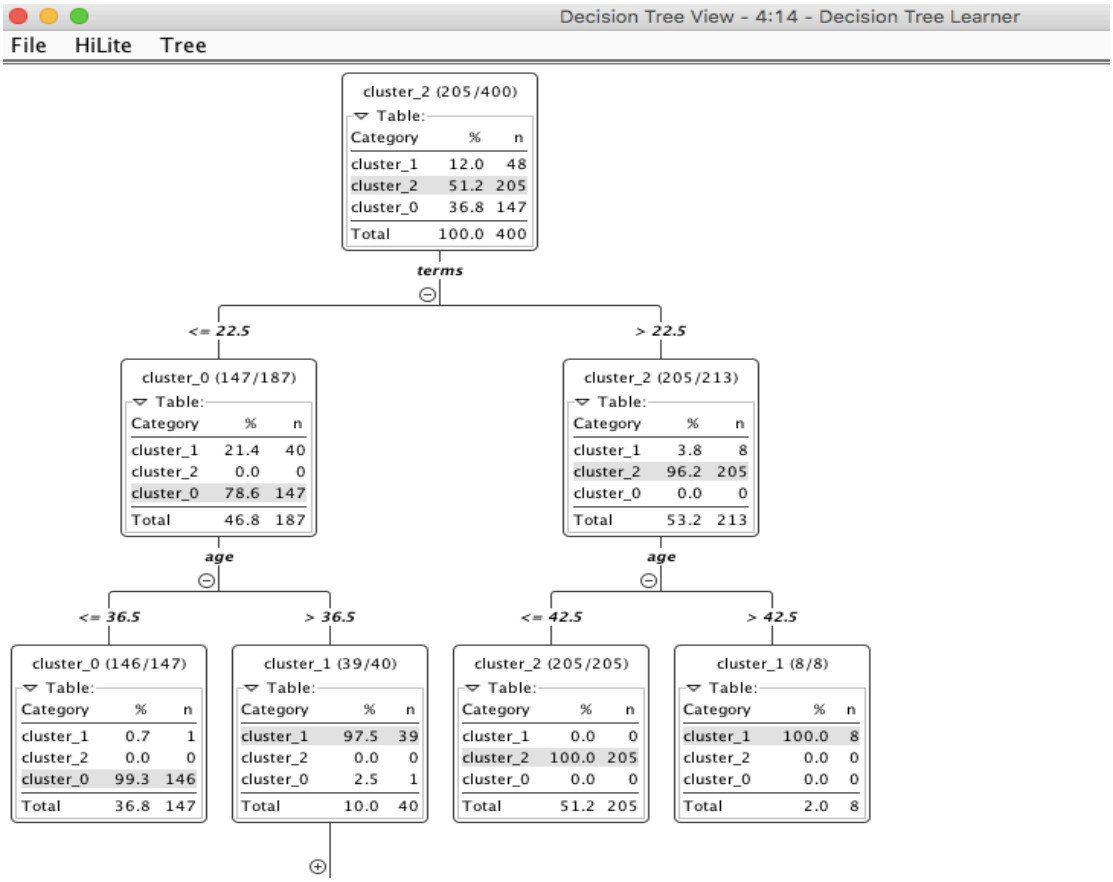
Şekil 12.5.21, yukarıda seçilen kolonlara göre program execute edildikten sonra elde edilen scatter plot'u göstermektedir. Decision tree'de de principal 950'den çok ya da az olmasına göre ilk dallanma çıkmıştı. Burada da görüldüğü gibi 950 ve yukarısı cluster 2 deki müşteriler çıkmıştır.

Karar verici kişi eğer principal a göre cluster istemez ise o zaman bu kolon işlemden çıkarılarak aynı işlemler tekrarlanabilir.



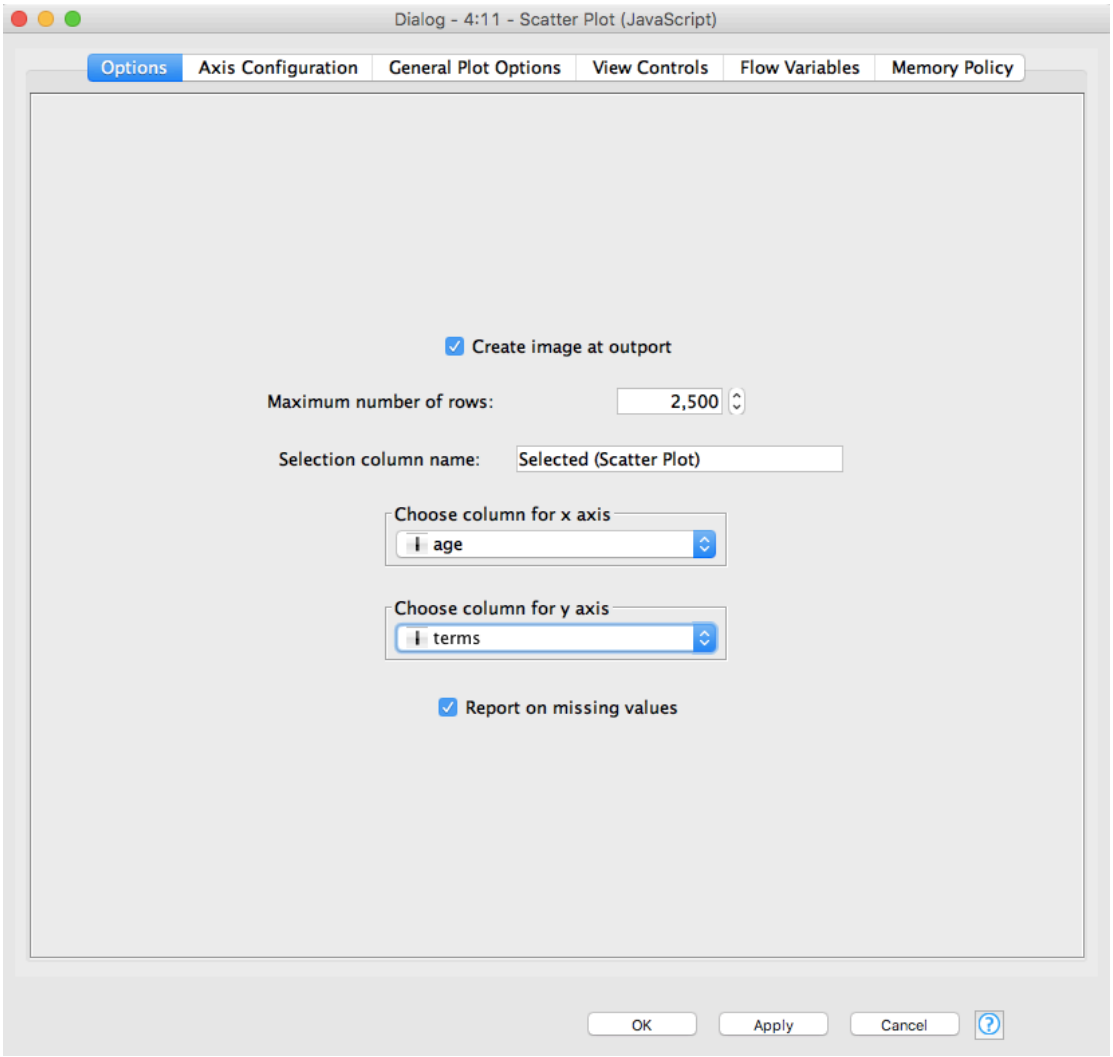
Şekil 12.5.22

Şekil 12.5.22, k-means operatöründen principal'ı işlemden çıkarmak için configure penceresini göstermektedir.



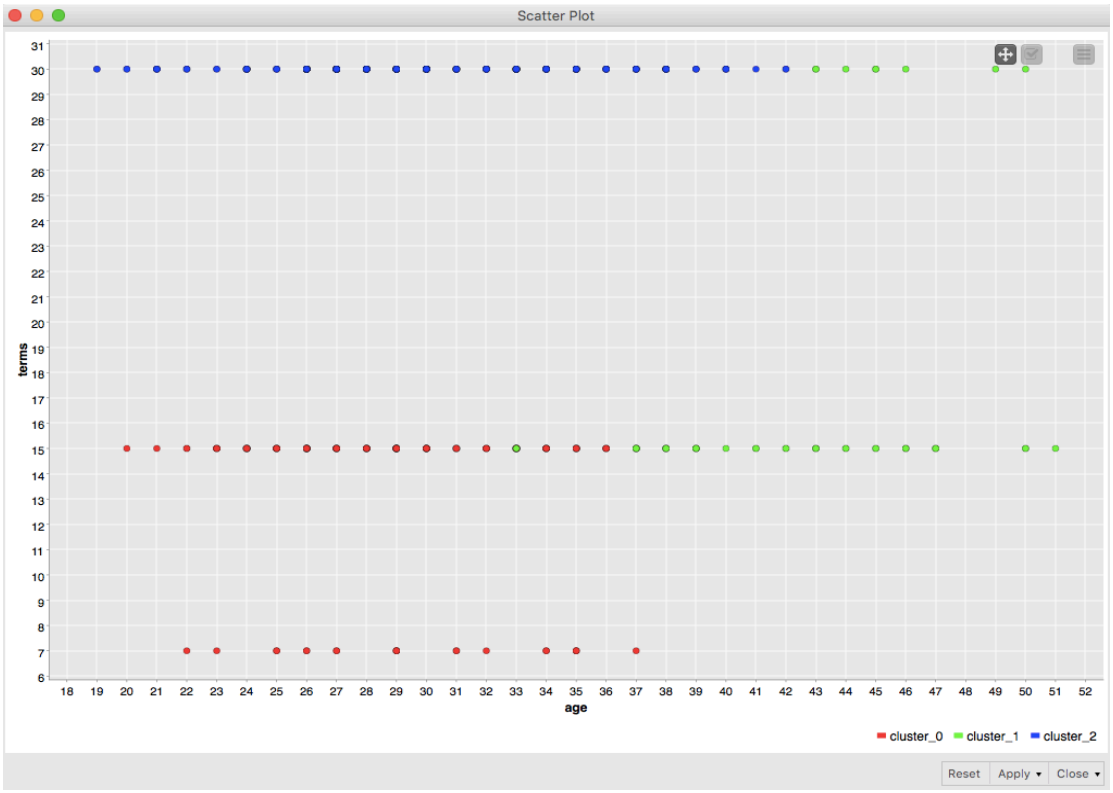
Şekil 12.5.23

Şekil 12.5.23, principal kolonu elenip program tekrardan çalıştırıldıktan sonra decision tree'nin dallanmasını göstermektedir. Bu sefer önce term sonra da age e göre dallanma olmuştur.



Şekil 12.5.24

Şekil 12.5.24, decision treeden alınan sonuca göre scatter plot için kullanılacak kolon seçimlerinin yapıldığı scatter plot configure penceresini göstermektedir.



Şekil 12.5.25

Şekil 12.5.25, terms ve age e göre scatter plot'ı göstermektedir. Örneğin decision tree sonucuna göre terms 22.5 dan büyük olanlar genellikle cluster 2 çıkmıştı. Burada da görüldüğü gibi cluster 2 terms 30 çıkmış.

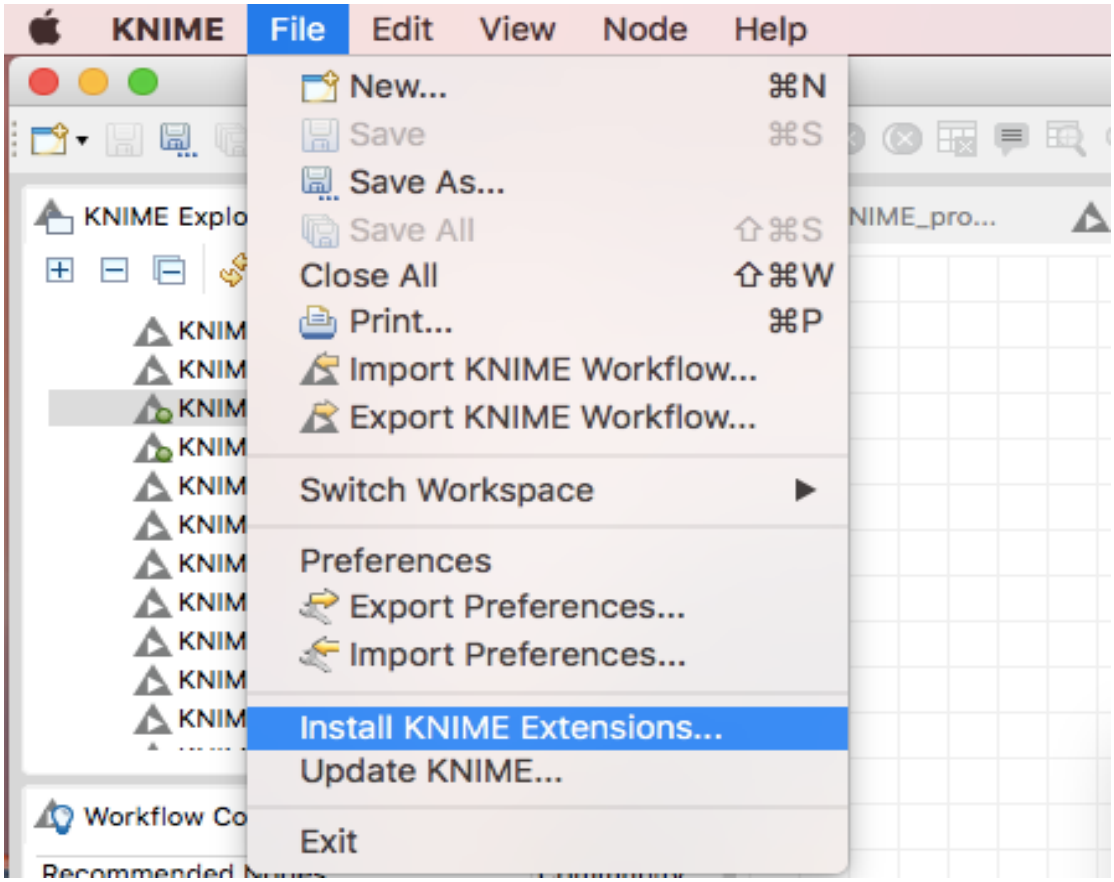
Eğer vadesi belli bir borç verme düzeni oluşturulmak istenirse o zaman k-means'den terms kolonu da exclude edilir ve tüm işlemler tekrardan sırasıyla yapılır ona göre scatter çizilebilir. Burada önemli olan hangi kolonların işleme katılacağının karar verici merci tarafından önceden belirlenmesi ona göre işlem yapılmasıdır. Eğer hiç bir eleme yapılmazsa makine kendisine göre decision tree'den sonuç çıkarır ve bu sonuca göre scatter plot'da kullanılacak kolonlar seçilebilir.

13. BÖLÜM: DERİN ÖĞRENME (DEEP LEARNING)

13.1 DL4j ile Knime Üzerinden Derin Öğrenme Uygulaması

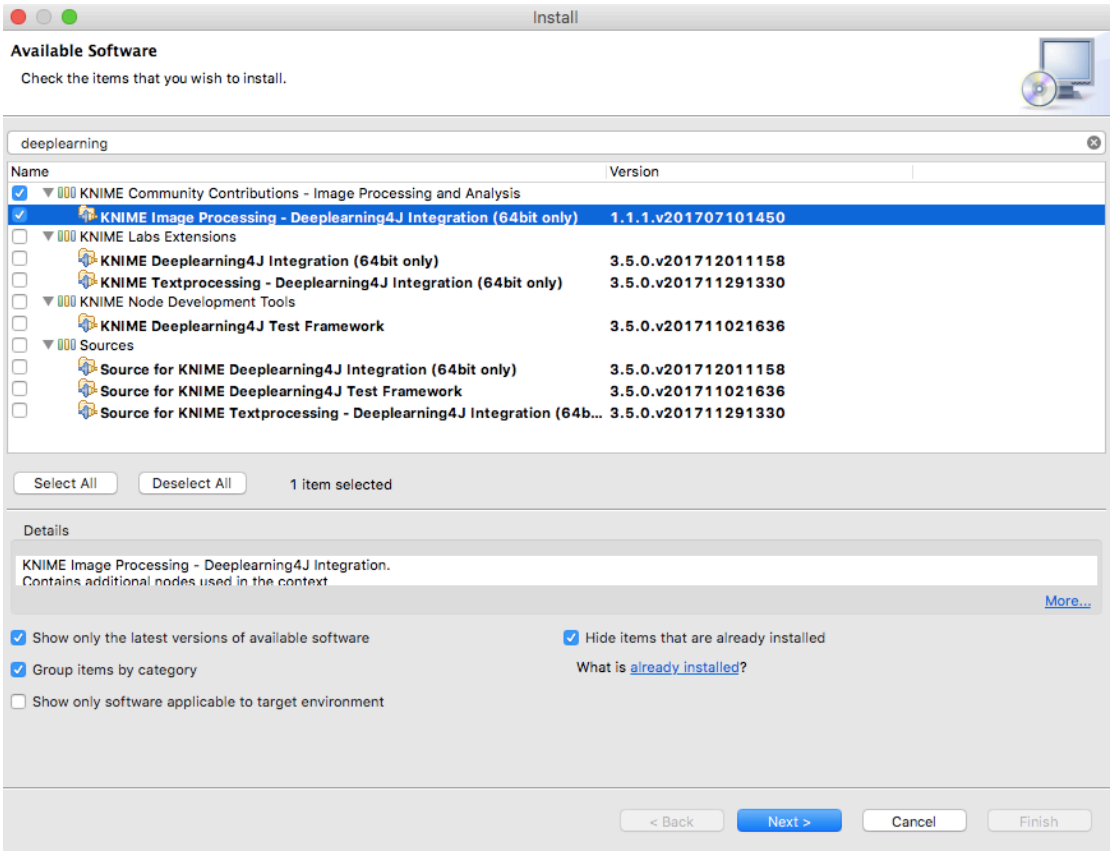
Bu bölümde amaç, Knime'da deep learning kütüphanelerinin kullanımını göstermektir.

Deep learning ile ilgili kütüphaneler Knie'da hazır kurulu gelmemektedir. Bu yüzden daha önceki bölümlerde yapıldığı gibi extensions bölününden seçilip Knime'a kurulmalıdır.



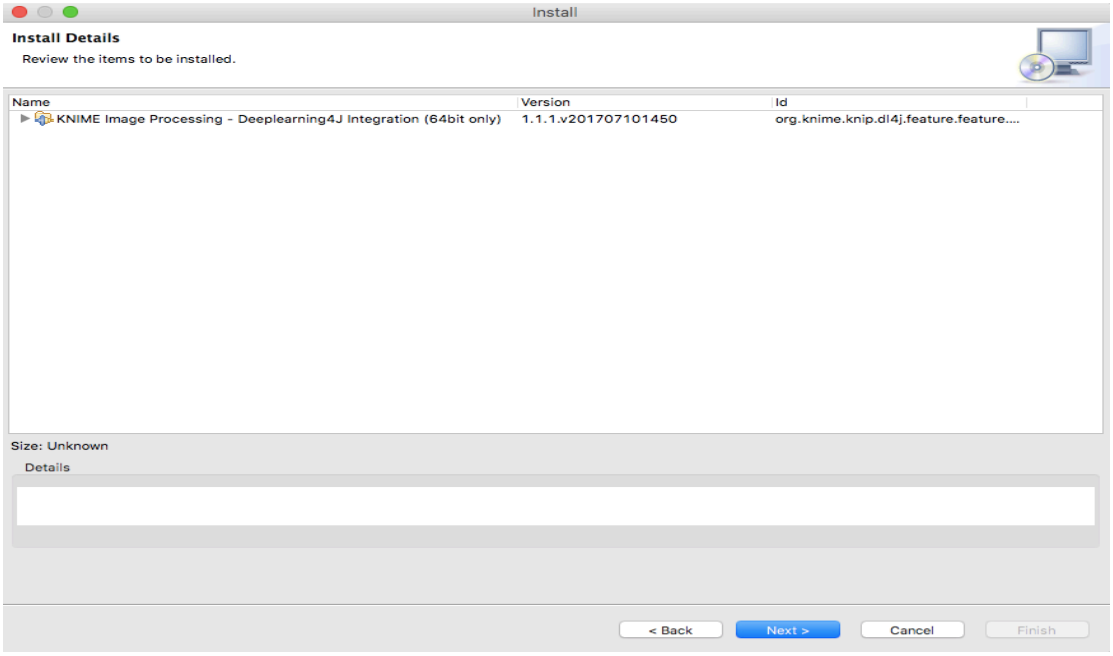
Şekil 13.1

Şekil 13.1, DL4J kütüphanesinin Knime'a install edileceği bölümü göstermektedir. Install Knime extensions sekmesine girdikten sonra aşağıdaki şekildeki pencere açılır.



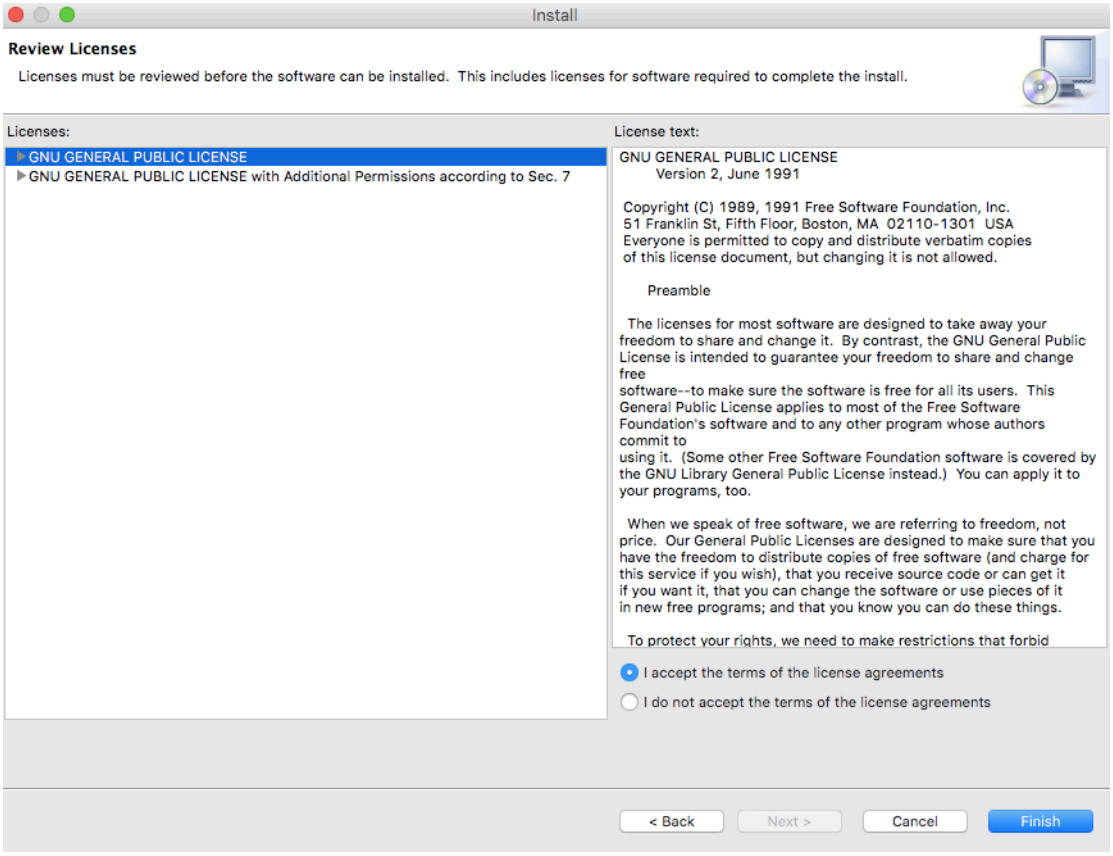
Şekil 13.2

Şekil 13.2, istenilen kütüphanenin arama butonuna yazılması gerekmektedir. Deeplearning kelimesi bitişik yazılarak aratılmalıdır. Şekilde görülen deeplearninf4J seçeneği seçilerek next tuşuna basılmalıdır.



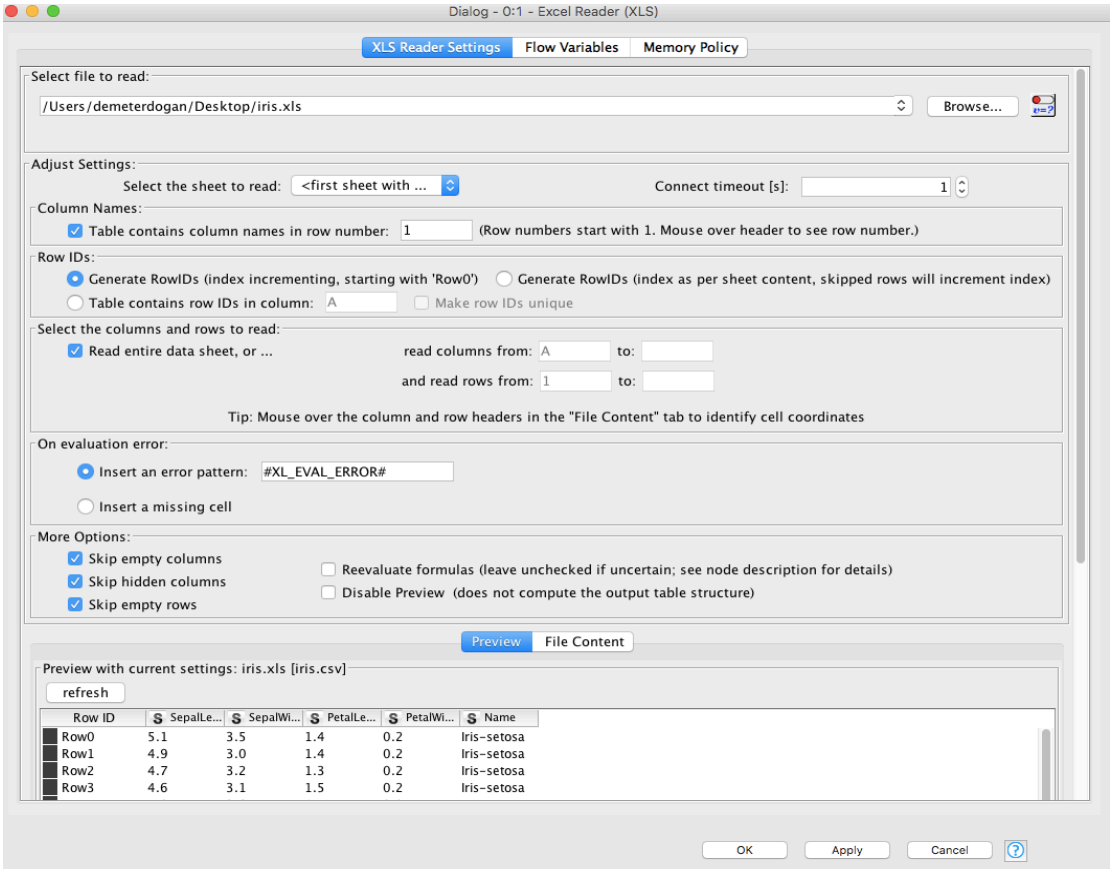
Şekil 13.3

Şekil 13.3, bir üstteki şekilde seçilen kütüphanelerin listesini gösterir. Next tuşuna basıldıktan sonra onay penceresi açılır.



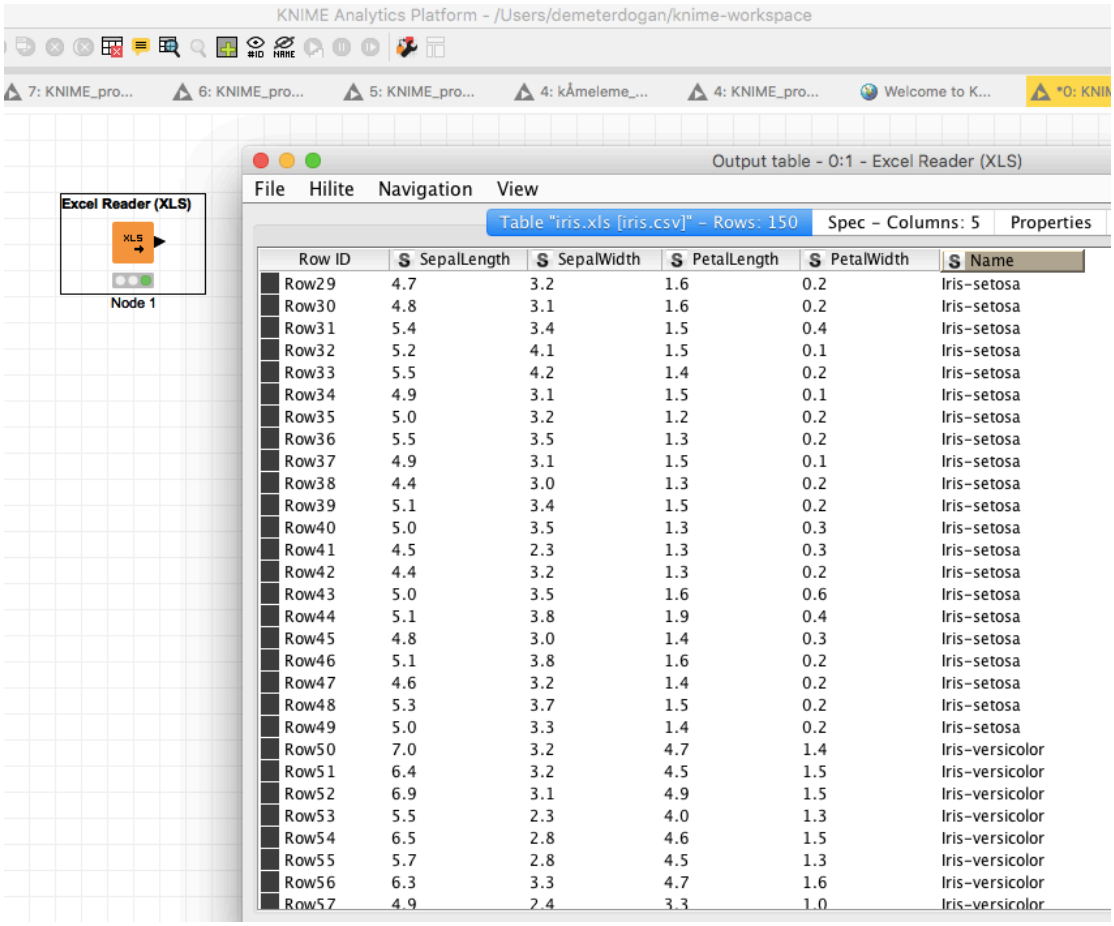
Şekil 13.4

Bu pencerede ı accept the terms of the license agreements seçeneği seçildikten sonra finish'e basılmalıdır. Bu aşamadan sonra kütüphanenin yüklenmesi bir kaç dakikayı bulabilir. Daha sonra yeni bir pencere açılarak Knime'ın yeniden başlatılması gerektiğini söyler. Üzerinde çalışılan pencereler varsa önce kaydedilmeli daha sonrasında yeniden başlatılmalıdır.



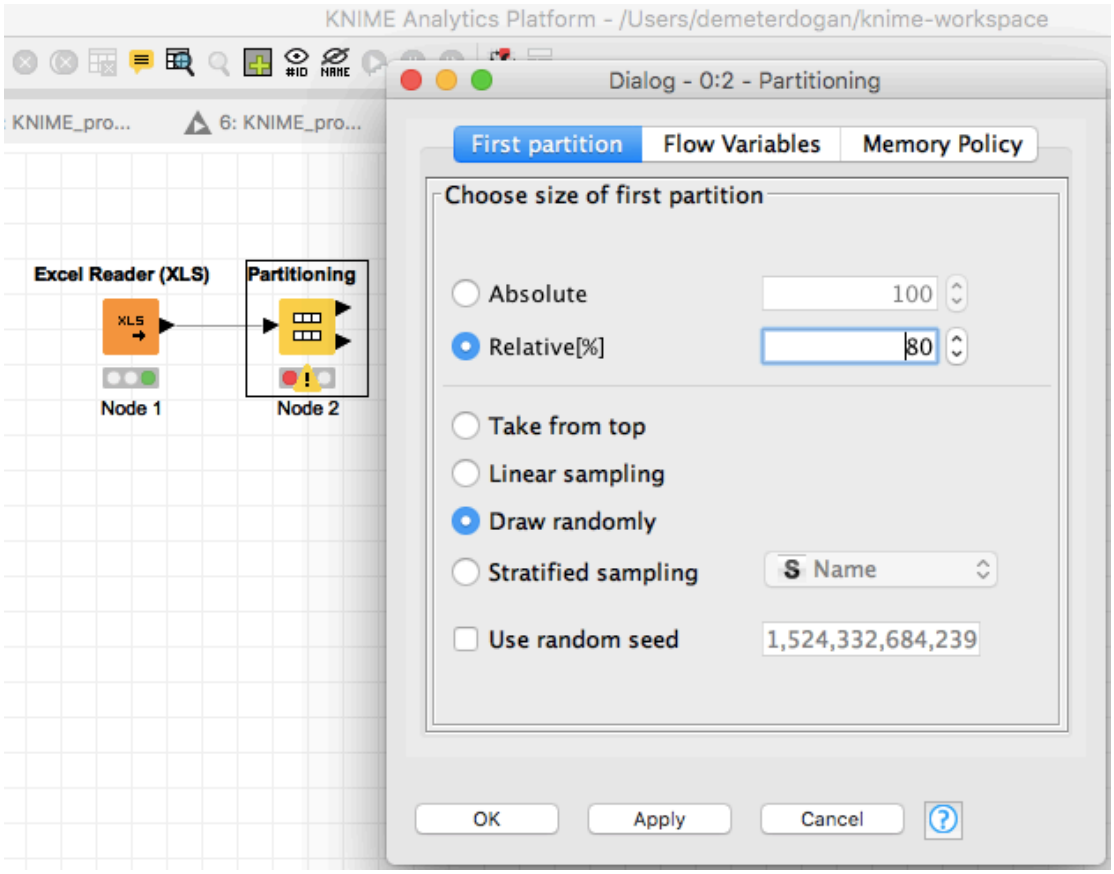
Şekil 13.5

Şekil 13.5, bu örnekte iris veri seti kullanılacağı için excel reader operatörüne bu veri setinin kayıtlı olduğu yerden seçilip yüklenmelidir. “table contains column names in row number 1” butonu seçilmelidir kolon başlıklarının ayrı olması için seçilmelidir.



Şekil 13.6

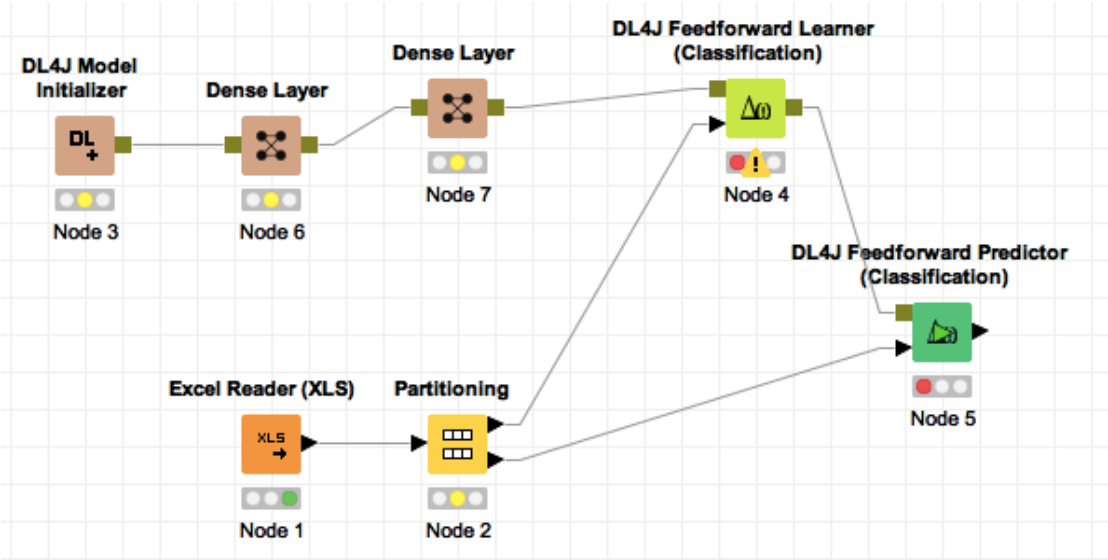
Şekil 13.6, excel reader'a iris veri seti yüklenip execute edildikten sonra elde edilen output table'ı göstermektedir. Görüldüğü gibi 4 çeşit yaprak genişliği ve 3 çeşit yaprak vardır. Bu veri üzerinde deep learning yapısı oluşturulmaya çalışılacaktır. Veriyi training ve test için bölmeye yardımcı partitioning operatörü kullanılarak veri parçalanacaktır.



Şekil 13.7

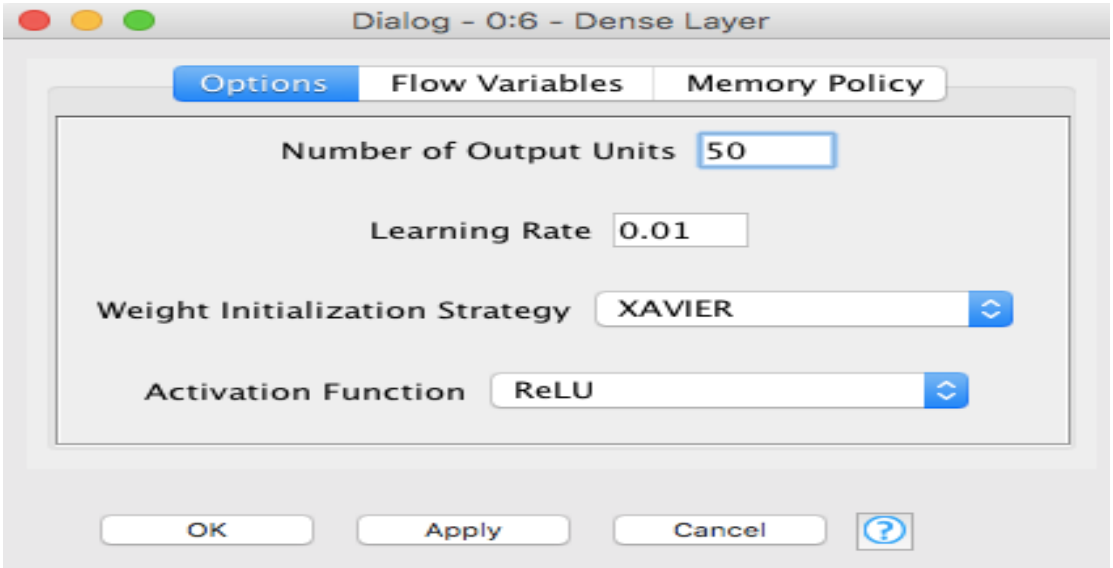
Şekil 13.7, partitioning operatörünün sisteme eklenmesini ve configure penceresinde 80% 20% oranında verinin rastlantısal olarak test ve eğitim için bölünmesini göstermektedir.

Yapay sinir ağı için network oluştururken DL4J initializer ile başlanır. Fakat buradaki yapıyı öğrenmek için henüz uygun değil çünkü herhangi bir network barındırmıyor. Kullanıcı neural netwrok detayını belirtmek zorundadır.



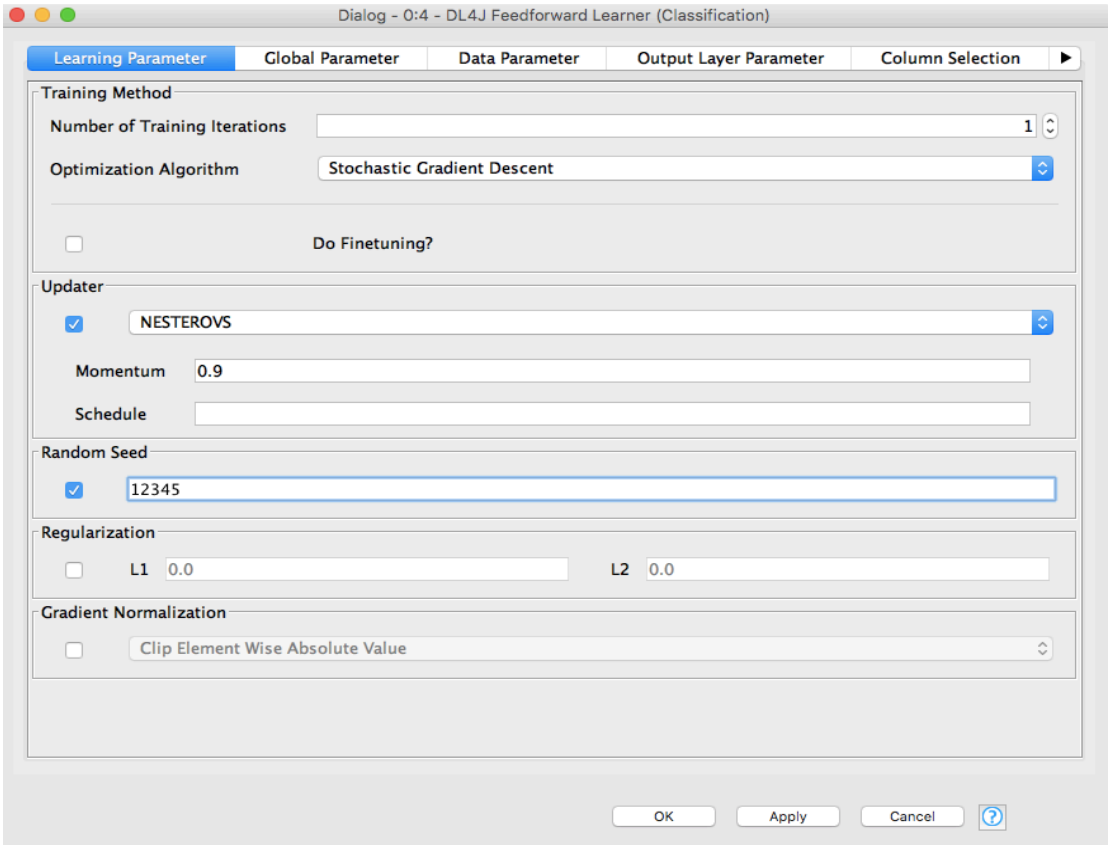
Şekil 13.8

Şekil 13.8, neural network için kullanılacak yapıyı (operatörleri) ve bağlantıları göstermektedir.



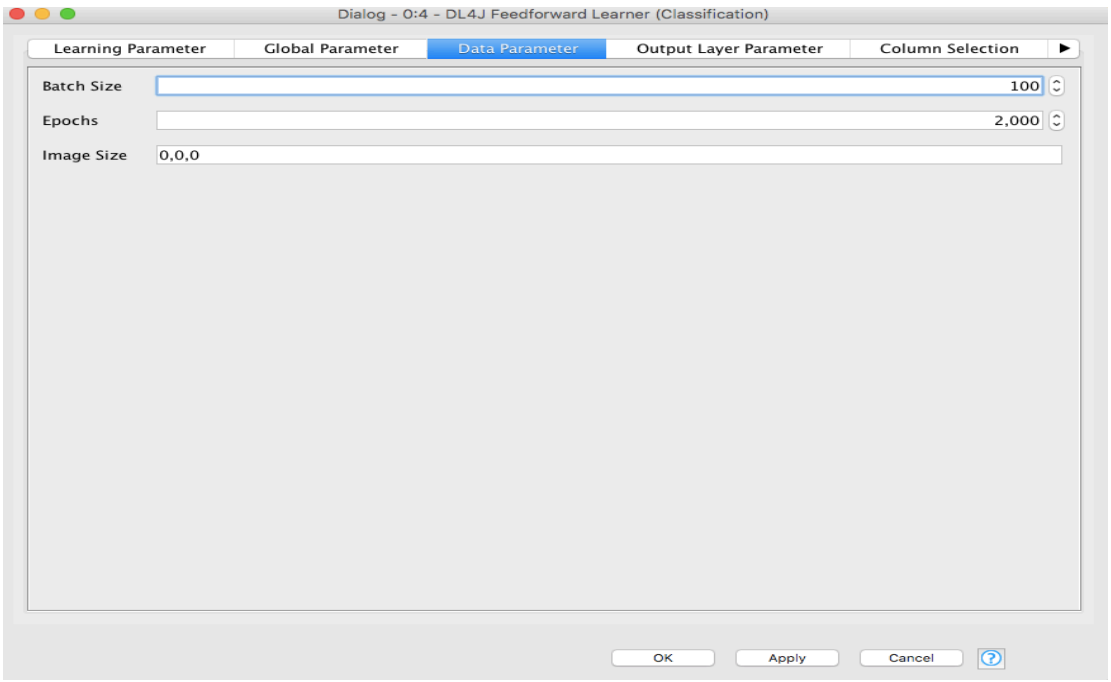
Şekil 13.9

Şekil 13.9, dense layer operatörlerinin configure için yapılması gereken değişikliği göstermektedir. Dense layer kaç katmanlı olacağını gösterir. Bu örnekte 2 dense layer kullanıldığı için iki katmanlı olacağı anlamına gelmektedir. Ve her katman için 50 neural olması istenmiştir.



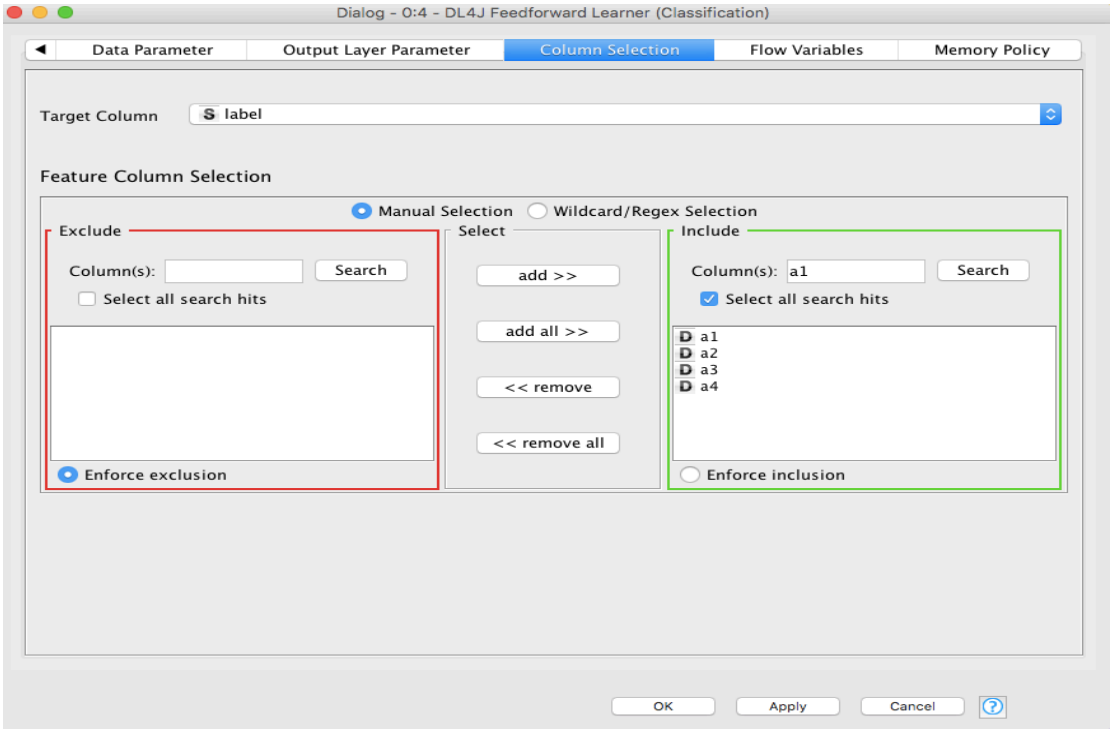
Şekil 13.10

Şekil 13.10, DL4J feedforward learner configure penceresindeki learning parameter bölümünü göstermektedir. Random seed başlayacağı yer buraya örnek olarak 12345 rakamı yazıldı. Bir iteration olsun istendiği için orası değiştirilmedi.



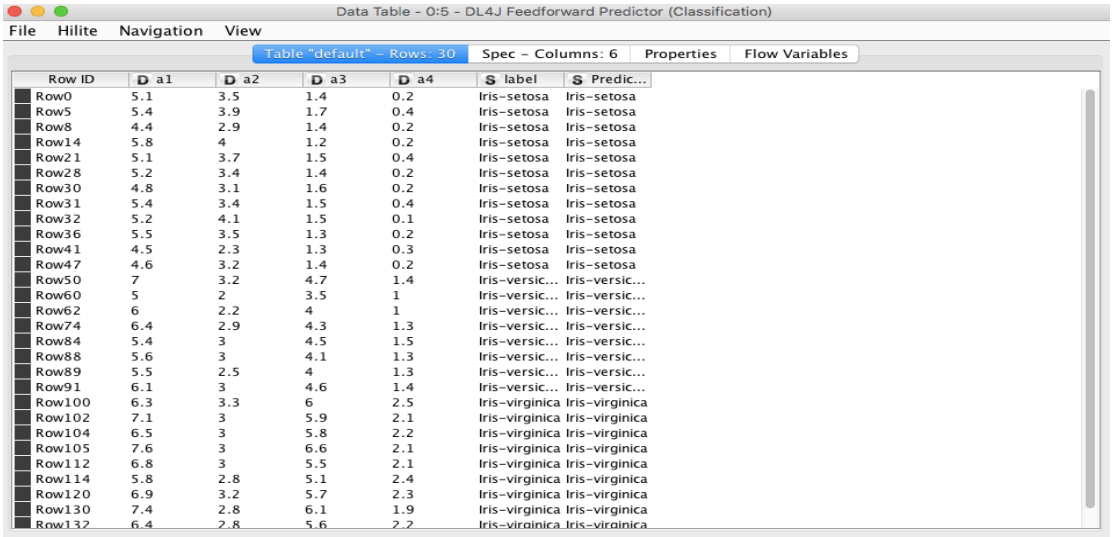
Şekil 13.11

Şekil 13.11, DL4J feedforward learner configure penceresindeki data parameter bölümünü göstermektedir. Epochs 2000 yapılmasının anlamı her bir öğrenme işleminin 2000 kez tekrar edilmesi anlamına gelmektedir.



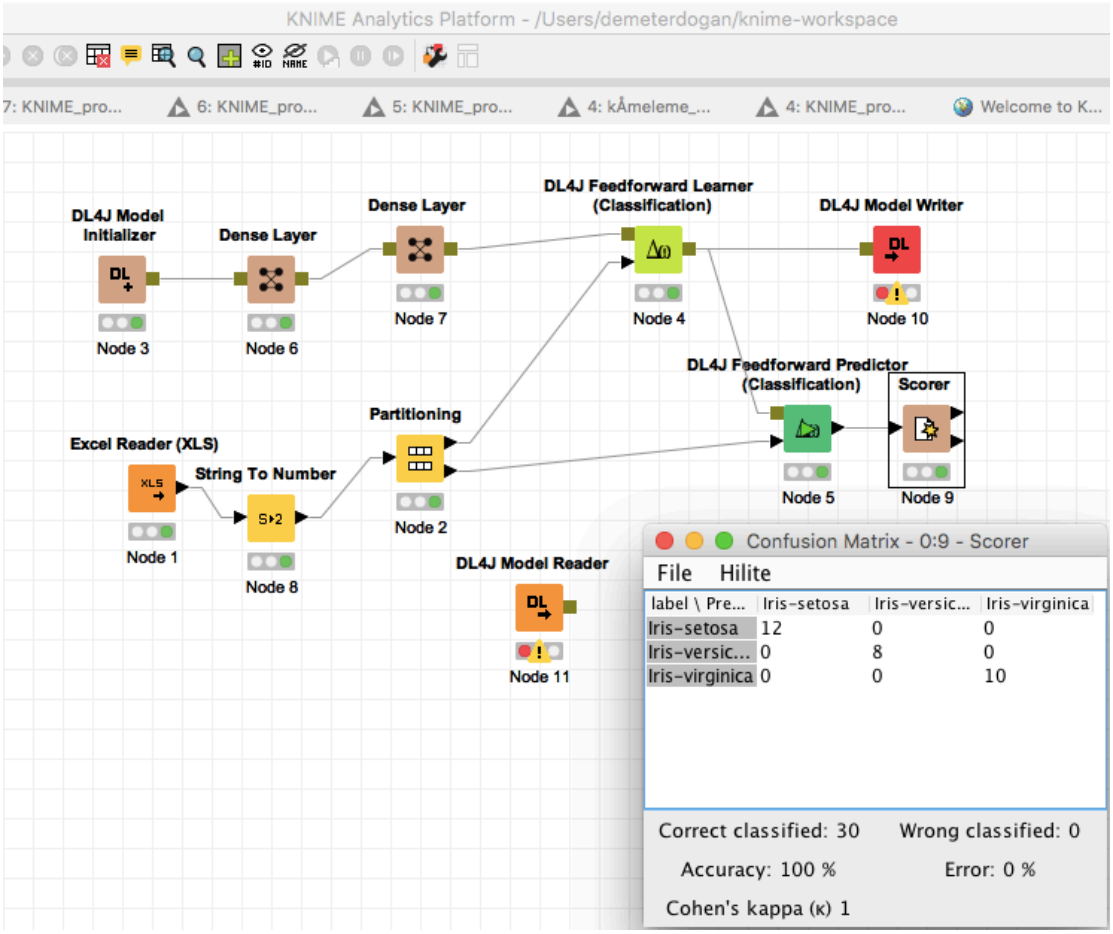
Şekil 13.12

Şekil 13.12, DL4J feedforward learner configure penceresindeki column selection bölümünü göstermektedir. Burada target column label seçilmiştir. Yani bulunmak istenilen kolon label kolonudur (yaprak isimlerinin olduğu kolon) ve tüm uzunluklar (a1, a2, a3, a4) kullanılmak üzere include penceresine aktarılmıştır.



Şekil 13.13

Şekil 13.13, program çalıştırıldıktan sonra elde edilen DL4J feedforward predictor classification penceresini göstermektedir. Label olan kolon veri setinden gelen ve yaprak isimlerinin bulunduğu kolon. Prediction (label) kolonu ise makinenin tahmini olan kolondur. Burada gözle kontrolde hata görülmemektedir fakat sisteme scorer operatörü ekleyerek daha detaylı değer incelemesi yapılabilir.



Şekil 13.14

Şekil 13.14, sisteme scorer eklenmiş ve çalıştırıldıktan sonraki sonuç penceresini göstermektedir. Accuracy 100% yani diagonal olarak da görüldüğü gibi iris setosa olan 12 verinin hepsi iris setosa olarak doğru bilinmiş, iris versicolor olan 8 verinin hepsi versicolor olarak bilinmiş ve iris virginica olan 10 verinin hepsi iris virginica olarak doğru bilinmiştir.

Ayrıca şekilde görüldüğü gibi DL4J Model Writer sisteme eklenip feedward learner ile bağlantı yapılırsa makine öğrenmesi bu şekilde kaydedilebilir. Bir sonraki sefer kullanılmak istenirse DL4J model reader sisteme eklenir ve predictora bağlantısı yapılarak direk program çalıştırılabilir.

Burada amaç deep learning'i açıklamak değildi. Burada amaç deep learningi veri bilimcisinin kod yazmadan bir tool ile (knime) basit şekilde kullanımı göstermekti.

14. Knime Nodes (Operatörleri)

1.IO:

i. Read (okuma)

- **Excel reader (XLS):**
  Üzerinde çalışılması hedeflenen excel formatındaki dosyaların Knime'a yüklenmesini ve Knime üzerinden okunmasını sağlar.

- **File Reader:**
  Knime'a yüklenmesini ve Knime üzerinden okunmasını sağlar.

- **ARFF Reader:**
  Üzerinde çalışılması hedeflenen ARFF formattaki dosyaların Knime'a yüklenmesini ve Knime üzerinden okunmasını sağlar.

- **CSV Reader:**
  Üzerinde çalışılması hedeflenen CSV (comma separated (virgülle ayrılmış metin dosyası)) formattaki dosyaların Knime'a yüklenmesini ve Knime üzerinden okunmasını sağlar.

- **PMML Reader:**
  XML uyumlu dosyalardaki PMML modellerinin Knime'a yüklenmesini ve Knime üzerinden okunmasını sağlar.

ii. Write (yazma)

- **Csv writer:**
  Üzerinde çalışılan veri tablosunu CSV dosyası (comma separated (virgülle ayrılmış metin dosyası)) olarak bilgisayarda istenilen yere kaydedilebilmesini sağlar.

- **ARFF writer:**
  Üzerinde çalışılan verilerin ARFF dosyası olarak bilgisayarda istenilen yere kaydedilebilmesini sağlar.

- **Table writer:**
  Bir iç tabloyu kullanarak bir dosyaya bir veri tablosu yazılabilmesini sağlar.

- **PMML writer:**
  Öğrenilen makine öğrenmesi veya istatistiksel modeli diske bir dosya olarak kaydetmeye yarar. Bu sırada da dünyaca standart hale gelmiş PMML standardını kullanır.

- **Excel writer (XLS):**
  Veri tablosunun excel dosyası olarak bilgisayarda istenilen yere kaydedilebilmesini sağlar.

iii. Other (Diğer)

- **Tablo creator:**
  Herhangi bir sayıda satır ve sütun içeren bir veri tablosunun manuel olarak oluşturulmasına izin verir.

- **Data generator:**
  Kümeler ile rastgele veri oluşturulmasına izin verir.

2. Manipulation:

a) Column

i. Binning

- **Auto-Binner:**
  Sayısal verileri aralıklarla otomatik olarak gruplandırmayı sağlar.

- **Numeric Binner:**
- Sayısal verileri aralıklarla otomatik olarak gruplandırmayı sağlar. Fakat auto binner node'undan farkı burada gruplara aralık değerleri de belirtilmelidir.

ii. Convert & Replace

- **Category to Number:**
  Bir sütunun her bir kategorisini nominal verilerle bir tamsayıya eşler.

- **Cell replacer:**
  Sözlük tablosuna göre bir sütundaki hücreleri 2. Girilen değer ile değiştirir.

- **Column Rename:**
  Sütun adlarının yeniden adlandırılması veya türlerinin değiştirilmesini sağlar.

- **Constant Value Column:**
  Her satıra sabit bir hücre içeren sütun ekler.

- **Math Formula:**
  Matematik formülünü değerlendirir, sonucunu yeni bir sütun olarak veya bir giriş sütununu değiştirerek değerlendirir.

- **Number to String:**
  Bir sütundaki sayıları dizelere dönüştürür.

- **String to Number:**
  Bir sütundaki dizeleri sayılara dönüştürür.

- **Double to Int:**
  Bir sütundaki ondalık sayıları tam sayılara dönüştürür.

- **Round Double:**
  Belirtilen ondalık basamağı yuvarlar.

- **String Manipulation:**
  Dizeleri, arama, değiştirme, önde gelen ve sondaki boşlukları kaldırma, büyük harfe çevirme gibi işlemler yapar.
- **String Replacer:**
  Belirli bir özel sembol eşleşiyorsa, dize hücrelerindeki değerleri değiştirir.

- **Edit Numeric Domain:**
  Seçilen sayısal sütunların etki alanını ayarlar.

- **Edit Nominal Domain:**
  Alanların olası değerlerini yönetir.

iii. Filter

- **Column Filter:**
  Sütun fitresi, sütunların giriş tablosundan çıkarılmasına izin verir.

- **Reference Column Filter:**
  Referans sütun filtresi referans olarak ikinci tabloyu kullanarak sütunların ilk tablodan filtrelenmesini sağlar.

- **Missing Value Column Filter:**
  Belirli bir yüzdeden daha fazla eksik değer içeren tüm sütunları kaldırır.

iv. Split & Combine

- **Cell Splitter:**
  Tabloların bir sütununda hücrelerin dize temsilini ayrı sütunlara veya belirtilen bir ayırıcıya dayalı olarak hücre koleksiyonunu içeren bir sütuna ayırır.

- **Column Aggregator:**
  Satır başına seçilen sütunları gruplandırır ve seçili kümeleme yöntemini kullanarak hücrelerini toplar.

- **Column Combiner:**
  Bir sütun kümesinin içeriğini birleştirir ve birleştirilmiş dizeyi giriş tablosuna ayrı sütun olarak ekler.

- **Column Merger:**
  Eksik olmayan hücreyi seçerek iki sütunu birleştirir.

- **Column Appender:**
  Bölücü düğümün çalışmasını tersine çevirmenin hızlı bir yoludur.

- **Create Collection Column:**
  Birden çok sütunu yeni bir koleksiyon sütununa birleştirir.

- **Joiner:**
  İki tabloyu birleştirir.

- **Cross Joiner:**
  İki tablonun çapraz birleştirilmesini gerçekleştirir.

v. Transform

- **Denormalizer:**
  Bir tablonun niteliklerini bir modele göre dengeler.

- **Normalizer:**
  Bir tablonun özelliklerini standartlaştırır.

- **Missing value:**
  Tablodaki eksik değerleri filtreler veya değiştirir.

- **One to Many:**
  Bir sütunun değerlerini eklenmiş sütunlara dönüştürür.

- **Many to One:**
  Birden çok sütunu tek bir sütuna toplar.

- **Set Operator:**
  Seçilen iki tablo sütununda bir set işlemi gerçekleştirir.

- **SMOTE:**
  SMOTE algoritmasını kullanarak öğrenme kalitesini arttırmak için yapay veriler ekler.

b) Row

i. **Filter**

- **Nominal Value Row Filter:**
  Nominal özellik değeri üzerinde satırları filtreler

- **Nominal Value Row Splitter:**
  Nominal özellik değerinde satırları böler

- **Row Filter:**
  Veri satırlarının satır kimliği, özellik değeri ve satır numarası aralığı gibi belirli kriterlere göre filtrelenmesine izin verir.

- **Row splitter:**
  Giriş tablosunun satır kimliği, özellik değeri ve satır numarası aralığı gibi belirli kriterlere bölünmesini sağlar.

- **Rule-based Row Filter:**
  Kullanıcı tanımlı iş kurallarına göre giriş tablosunu filtreler.

- **Rule-based Row Splitter:**
  Kullanıcı tanımlı iş kurallarına göre giriş tablosunu böler.

ii. **Transform**

- **Concatenate:**
  Sıralı iki tabloyu birleştirir.

- **GroupBy:**
  Seçili sütun (lar) ile tabloyu gruplandırır ve seçili kümeleme/birleştirme yöntemini kullanarak kalan sütunları toplar.

- **Ungroup:**
  Koleksiyon değerlerinin her bir listesi için bir sütundaki koleksiyonun değerlerine ve orijinal satırdan verilen diğer tüm sütunlara sahip satırların listesini oluşturur.

- **Partitioning:**
  Tabloyu istenilen oranda ikiye bölüme ayırır.

- **Pivoting:**
  Döndürme ve gruplama için seçili sütunların giriş tablosunu döndürür ve gruplar; sütun kümeleme/gruplama ile geliştirilmiştir.

- **Unpivoting:**
  Seçilen sütunları giriş tablosundan satırlara döndürür ve kalan giriş sütunlarını aynı anda ilgili çıktı satırlarına ekleyerek çoğaltır.

- **Rank:**
  Seçilen sıralama özniteliklerine ve sıralama moduna göre seçilen grupların sıralamasını hesaplar.

- **Row Sampling:**
  Giriş verilerinden bir örnek (bir grup satır) çıkarır.

- **Sorter:**
  Satırları kullanıcı tanımlı kriterlere göre sıralar.

iii. Other

- **Add Empty Rows:**
  Girdi tablosuna eksik değerler veya belirli bir sabitle belirli sayıda boş satır ekler.

- **Extract Column Header:**
  Tek bir satır ile sütun adlarını içeren yeni bir tablo oluşturur.

- **Insert Column Header:**
  İkinci tablonun tablosundaki eşleştirmeye göre diğer tablonun sütun isimlerini günceller.

- **RowID:**
  RowID'yi değiştirir ve / veya geçerli RowID'nin değerleriyle bir sütun oluşturur.

- **Rule Engine:**
  Kullanıcı tanımlı iş kurallarını giriş tablosuna uygular.

c) Table

- **Extract Table Dimension:**
  Giriş tablosunun satır ve sütun sayısını çıkarır ve bunları veri tablosu ve akış değişkenleri olarak çıkarır.

- **Extract Table Spec:**
  Düğüm, girdi tablosundan meta bilgileri ayıklar (sütun adları, türleri, vb.).

- **Transpose:**
  Satırları ve sütunları değiştirerek bir tabloyu aktarır.

d) PMML

- **Normalizer (PMML):**
  Bir tablonun özelliklerini normalleştirir.

- **Number to String (PMML):**
  Bir sütundaki sayıları dizelere dönüştürür.

- **Numeric Binner (PMML):**
  Sayısal sütunların grup değerleri dize tipinde kategorize eder.

- **String to Number (PMML):**
  Bir sütundaki dizeleri sayılara dönüştürür.

- **XML to PMML:**
  XML belgelerini PMML belgelerine dönüştürür

- **Cell to PMML:**
  PMML hücresini ilk satırdaki PMML bağlantı noktasına dönüştürür.

- **PMML to Cell:**
  PMML Portunu PMML hücresini içeren bir tabloya dönüştürür.

3. Views

a) **Property**
   - **Color Manager:**
     Seçilen bir nominal veya sayısal sütuna renkler atar.

   - **Size Manager:**
     Bir sayısal sütunun değerlerine karşılık gelen boyutlar atar.

b) **Utility**
   - **Image to Table:**
     Verilen bir görüntüyü hücreye sahip bir tabloya dönüştürür.

   - **Tablet o Image:**
     Tablo satırında bulunan görüntüyü bir görüntü nesnesine dönüştürür.

c) **Box Plot:**
   Bir kutu çiziminde, sayısal nitelikler için sağlam istatistiksel parametreler görüntülenir ve aşırı aykırı değerler tanımlanır.

d) **Histogram:**
   Verileri histogram görünümünde görüntüler.

e) **Lift Chart:**
   Bir yükseltme grafiği oluşturur.

f) **Line Plot:**
   Sayısal sütunları satır olarak çizer.

g) **Pie Chart:**

Verileri bir pasta grafiğinde görüntüler.

h) Scatter Matrix:
Her sütunun diğerleriyle karşılaştırıldığı bir dağılım matrisi oluşturur.

i) Scatter Plot:
Seçilen iki özniteliğin dağılım grafiğini oluşturur.

4. Analytics

a. **Mining**

i. **Bayes**

- **Naive Bayes Learner:**
  Verilen sınıflandırılmış verilerden Naif Bayes modeli oluşturur.

- **Naive Bayes Predictor:**
  Girdi verilerindeki her satırın sınıf üyeliğini öngörmek için, Naif Bayes öğrencisinden PMML Naif Bayes modelini kullanır.

ii. **Clustering**

- **Fuzzy c-Means:**
  Bulanık c-kümeleri kümeleme gerçekleştirir.

- **k-Means:**
  Yeni bir merkez tabanlı kümeleme oluşturur.

- **Hierarchical Clustering:**
  Hiyerarşik Kümeleme gerçekleştirir.

- **SOTA Learner:**
  SOTA ile sayısal verileri küme.

- **SOTA Predictor:**
  SOTA modelini kullanarak satırlar için sınıfları tahmin eder.

- **Hierarchical Cluster View:**
  Hiyerarşik kümeleme sonuçlarını gösterir.

- **Hierarchical Cluster Assigner:**
  Kümeleri, bir hiyerarşik kümeleme temelinde satırlara atar.

iii. **Rule Induction**

- **Fuzzy Rules Learner:**
  Etiketli sayısal veriler üzerinde Bulanık Kuralı Modelini öğrenir.

- **Fuzzy Rule Predictor:**

Sayısal verilere bir Bulanık Kural Modeli uygular ve her test örneği için bir tahmin çıkarır.

iv. Neural Network

a. MLP

- **MultiLayerPerceptron Predictor:**
  Eğitimli bir çok katmanlı algılayıcıya bağlı olarak çıktı değerlerini tahmin eder.

- **RProp MLP Learner:**
  Esnek geri yayılımlı bir çok katmanlı algılayıcı oluşturur ve öğrenir.

b. PNN

- **PNN Learner:**
  Etiketli verilerde Olasılıksal Sinir Ağı'nı (PNN) eğitir.

- **PNN PRedictor:**
  Sayısal verilere bir PNN Modeli uygular ve bir sınıflandırma çıkarır.

v. Decision Tree

- **Decision Tree Learner:**
  Veri madenciliğinde kullanılan 3 önemli yöntemden birisidir. Seçilen sütunu etkileyen faktörlere göre dallanarak oluşturulan bir nevi ağaç yaratır.

- **Decision Tree Predictor:**
  Bu node'da iki giriş bulunur çünkü birisi gelen veriler için diğeriyse çıkartılan kuraldı. Bu node'u kullanabilmek için kural gereklidir vw bu kuralı learner'dan alır.

vi. Decision Tree Ensemble

c. Gradient Boosting

I. Classification

- **Gradient Boosted Trees Learner:**
  Güçlendirilmiş gradyan ağaç modelini öğrenir.
- **Gradient Boosted Trees Predictor:**
  Güçlendirilmiş gradyan ağaç modelini kullanarak verileri sınıflandırır.

II. **Regression**

- **Gradient Boosted Trees Learner (Regression):**
  Güçlendirilmiş gradyan ağaç modelini öğrenir.
- **Gradient Boosted Trees Predictor (Regression):**
  Güçlendirilmiş gradyan ağaç modelinden regresyon uygular.

d. **Random Forest**

I. **Classification**

- **Random Forest Learner:**
  Sınıflandırma için rasgele bir orman öğrenir.

- **Random Forest Predictor:**
  Rasgele bir orman modelinde tek tek ağaçların tahminlerinin bir araya getirilmesine göre kalıpları tahmin eder.

II. **Regression**

- **Random Forest Learner (Regression):**
  Regresyon için rasgele bir orman öğrenir.

- **Random Forest Predictor (Regression):**
  Bireysel tahminlerin ortalamasını kullanarak rasgele bir orman modelinden regresyon uygular.

vii. **Misc Classifiers**

- **K Nearest Neighbor:**
  Eğitim verilerini kullanarak k En Yakın Komşu algoritmasına dayalı bir dizi test verilerini sınıflandırır.

viii. **Item Sets / Association Rules**

- **Association Rule Learner:**
  Birtakım işlemlerde belirli bir minimum destekle sık kullanılan itemleri arar ve isteğe bağlı olarak onlardan önceden tanımlanmış bir güven değeriyle ilişkilendirme kuralları oluşturur.

ix. **Linear / Polynomial Regression**

- **Linear Regression Learner:**
  Çok değişkenli bir doğrusal regresyon gerçekleştirir.

- **Polynomial Regression Learner:**
  Giriş verilerinden bir polinom regresyon modeli oluşturan eğitim modülü

- **Regression Predictor:**
  Bir regresyon modeli kullanarak yanıtı tahmin eder.

x. **Logistic Regression**

- **Logistic Regression Learner:**
  Çok terimli bir lojistik regresyon gerçekleştirir.

- **Logistic Regression Predictor:**
  Bir lojistik regresyon modeli kullanarak yanıtı tahmin eder.

xi. **SVM**

- **SVM Learner:**
  Bir destek vektör makinesini eğitir.

- **SVM Predictor:**
  Verilen parametrelerin çıktısını tahmin etmek için SVM öğrenici nodu tarafından üretilen bir SVM modelini kullanır.

xii. **Feature Selection**

- **Feature Selection Loop Start (1:1):**
  Özellik seçim döngüsünün başlangıcıdır. Özellik seçim döngüsü, giriş veri kümesindeki tüm özelliklerden, model yapımı için en iyi özelliklerin alt kümesini seçilebilmesini sağlar. Bu düğüm ile (i) seçim sürecinde hangi özelliklerin / sütunların sabitleneceğini belirlenir. Bu sabit veya "statik" özellikler / sütunlar her döngü yinelemesinde bulunur ve elemeden muaftır; (ii) diğer (değişken) özellikler / sütunlarda hangi seçim stratejisinin kullanılacağını; ve (iii) değişken özelliklerin eşik sayısında seçim sürecinin sonlanmasıdır.

- **Feature Selection Loop Start (1:2):**
  Bu düğüm, özellik seçim döngüsünün başlangıcıdır. Özellik seçim döngüsü, giriş veri kümesindeki tüm özelliklerden, model yapımı için en iyi özelliklerin alt kümesini seçmenizi sağlar. Bu düğüm ile (i) seçim sürecinde hangi özelliklerin / sütunların sabitleneceğini belirlernir. Bu sabit veya "statik" özellikler / sütunlar her döngü yinelemesinde bulunur ve elemeden muaftır; (ii) diğer (değişken) özellikler / sütunlarda hangi

seçim stratejisinin kullanılacağını; ve (iii) değişken özelliklerin eşik sayısında seçim sürecinin sonlanmasıdır. Bu düğümde iki giriş ve çıkış portu var. İlgili ilk port, eğitim verisi ve test verileri için ikinci port için tasarlanmıştır. Aynı filtre her iki tabloya da uygulanır ve bu nedenle her zaman aynı sütunları içerir.

xiii. Scoring

a. X Cross Validation

- **X-Partitioner:**

  Bu düğüm, çapraz doğrulama döngüsündeki ilktir. Döngünün sonunda her iterasyondan sonuçları toplamak için bir X-Toplayıcı bulunmalıdır. Bu iki düğüm arasındaki tüm düğümler, yinelemelerin yapılması gerektiğinden defalarca yürütülür.

- **X-Aggregator:**

  Bu düğüm çapraz geçerlilik döngüsünün sonu olmalı ve bir X-Partitioner düğümü izlemelidir. Bir tahminci düğümden sonucu toplar, öngörülen sınıfı ve gerçek sınıfı karşılaştırır ve tüm satırlar için tahminleri ve yineleme istatistiklerini çıkarır.

b. Scorer:

Öznitelik değer çiftlerine göre iki sütunu karşılaştırır ve karışıklık matrisini gösterir, yani kaç tane öznitelik sırası ve sınıflandırması eşleşir. Pencere, karşılaştırma için iki sütun seçilmesine izin verir; Seçilen ilk sütundan gelen değerler, hata matrisinin satırlarında ve ikinci sütundan gelen hata matrisi sütunları ile temsil edilir. Düğümün çıktısı her hücredeki eşleşme sayısı ile karışıklık matrisidir. Ek olarak, ikinci çıkış noktası, Doğru-Pozitif, Yanlış-Pozitif, Doğru Negatifler, Yanlış-Negatifler, Geri Çağırma, Hassasiyet, Hassasiyet, Özgünlük, F-ölçümü ve genel doğruluk gibi bir dizi doğruluk istatistiklerini rapor eder.

c. Numeric Scorer:

Sayısal bir sütunun değerleri ve tahmini değerler arasındaki belirli istatistikleri hesaplar.

d. ROC Curve:

Eğrilik yarıçapı eğrilerini gösterir.

xiv. Weka

Bir zip dosyasına/dosyasından weka sınıflandırma modeli yazar ve okur.

b. Statistics

Hypothesis Testing

- **Single Sample t-test:**
  Tek örneklem t-testi, popülasyon ortalamasının belirli bir sayıya eşit olduğu sıfır hipotezini test eder.

- **Independent Groups t-test:**
  Bu test, iki grup arasında aynı sütundaki gözlem araçlarını karşılaştırmak için tasarlanmıştır.

- **Paired t-test:**
  Eşleştirilmiş (veya "bağımlı") t-testi, birbirinden bağımsız olmayan gözlem araçlarının karşılaştırılması için kullanılır.

- **One-way ANOVA**
  Tek yönlü varyans analizi (ANOVA), çeşitli araçlardan herhangi birinin birbirinden farklı olup olmadığını test etmeyi sağlar.

5. Other Data Types

Time Series

a. **Manipulation**
   - **Create Date & Time Range:**
     Tarih ve saat değerlerini üretir.

   - **Date & Time Difference :**
     İki tarih ve saat hücresi arasındaki farkları hesaplar.

   - **Date & Time Shift:**
     Bir süreye veya ayrıntıya göre bir tarih veya saat kaydırır.

   - **Modify Date:**
     Bir tarih ve saat hücresinin tarih bilgilerini değiştirir.

   - **Modify Time:**
     Bir tarih ve saat hücresinin zaman bilgisini değiştirir.

   - **Modify Time Zone:**
     Bir saat dilimini değiştirir.

b. **Transform:**
- **Date & Time to String:**
  Tarih ve Saat hücrelerini dizeleri tutan hücrelere dönüştürür.

- **String to Date & Time:**
  Tarih ve Saat hücrelerine tarih ve / veya zaman dizeleri ayrıştırır.

- **Duration to String:**
  Bir süreyi bir dizeye dönüştürür.

- **Duration to Number:**
  Süresi hücreleri tek süreli alanlara dönüştürür.

- **String to Duration:**

Bir dizeyi bir süreye dönüştürür.

- **Date & Time to legency Date & Time:**
  Yeni tarih & saati eskisine dönüştürür.

6. Scripting

a. Java

- **Java Snippet:**
  Java kod parçacıklarına dayalı yeni bir sütun veya akış değişkenleri hesaplar.

- **Java Snippet Row Filter:**
  Java parçaları tabanlı sıra filtresi

b. Python

- **Python Edit Variable:**
  Yerel bir Python yüklemesinde bir Python betiğinin yürütülmesine izin verir. Python yürütülebilir dosyasının yolu Tercihler (preferences) → KNIME → Python'da yapılandırılmalıdır. Bu düğüm Python 2 ve 3'ü destekler.

7. Community Nodes

KNIME Topluluk Katkıları, kimya ve biyoinformatik, görüntü işleme veya bilgi alma gibi farklı uygulama alanlarından çok çeşitli KNIME düğümleri sunar.

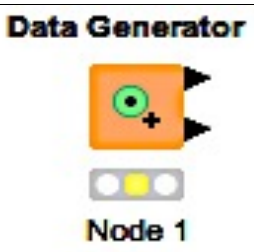
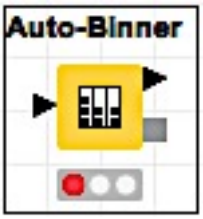
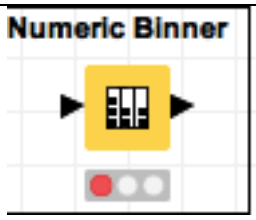
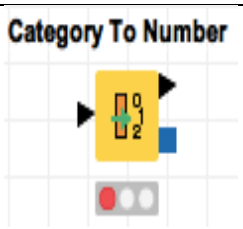
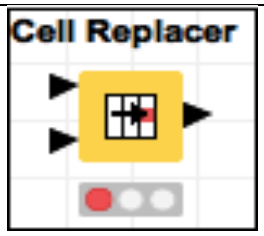
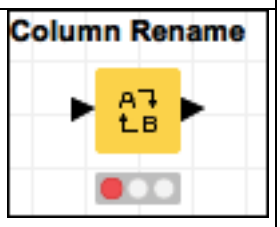
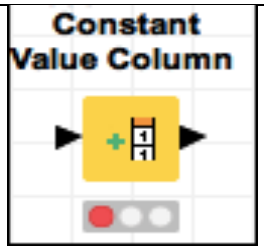
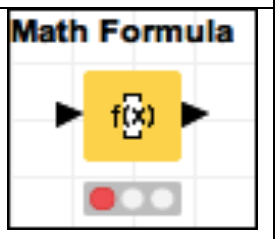
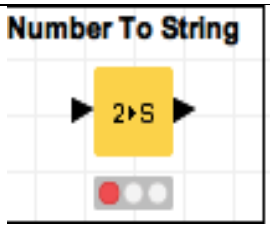
Standart KNIME Güncelleme Sitesi aracılığıyla mevcut uzantıların aksine, çeşitli topluluk geliştiricileri tarafından sağlanır ve korunur.

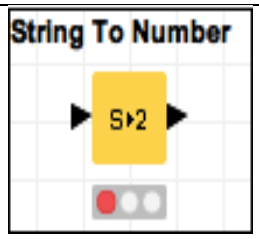
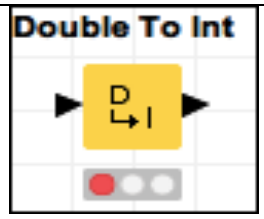
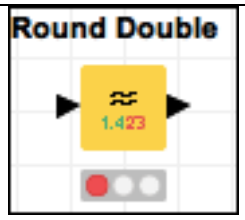
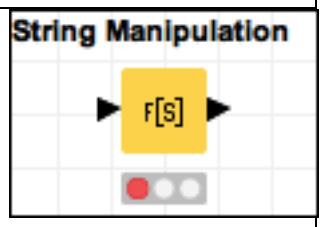
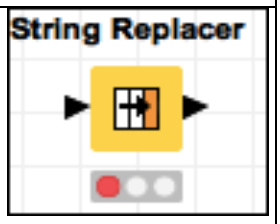
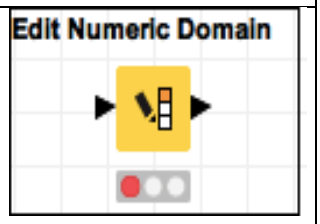
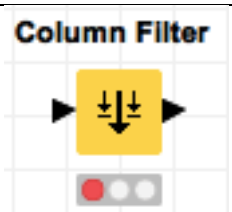
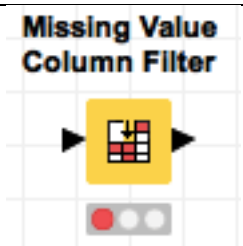
Güvenilir Topluluk Katkıları, KNIME'de Dosya (file) --> KNIME Uzantılarını Yükle'yi (install Knime extensions in Knime) seçerek kolayca yüklenebilir.

Ek Uzantıları, Dosya (file) --> Tercihler (preferences) --> Yükleme (install) / Güncelleme (update) --> Kullanılabilir Güncelleme Siteleri aracılığıyla (available update sites) KNIME'de Güncelleme Sitesini etkinleştirerek kullanılabilir.

8.Nodes İkonları (Görselleri)

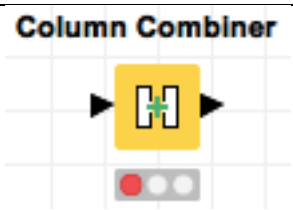
Excel Reader	Excel Reader (XLS) XLS Node 1		CSV Writer	CSV Writer Node 1	
File Reader	File Reader Node 1		ARFF Writer	ARFF Writer ARFF Node 1	
ARFF Reader	ARFF Reader ARFF Node 1		Table Writer	Table Writer Node 1	
CSV Reader	CSV Reader Node 1		PMML Writer	PMML Writer PMML Node 1	
PMML Reader	PMML Reader PMML Node 1		Excel Writer	Excel Writer (XLS) XLS Node 1	

Tablo Creator		Data Generator	
Auto Binner		Numeric Binner	
Category to Number		Cell Replacer	
Column Rename		Constant Value Column	
Math Formula		Number to String	

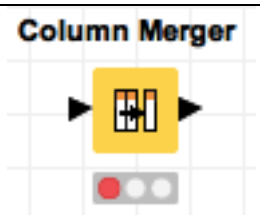
String to Number	
Double to Int	
Round Double	
String Manipulation	
String Replacer	
Edit Numeric Domain	
Column Filter	
Reference Column Filter	
Missing Value Column Filter	
Cell Splitter	

Column Aggregator

Column Combiner



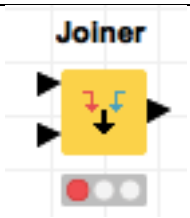
Column Merger



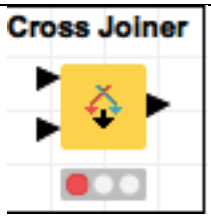
Column Appender

Create Collection Column

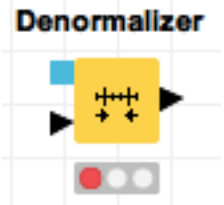
Joiner



Cross Joiner

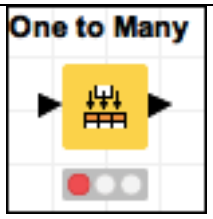
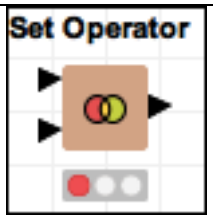
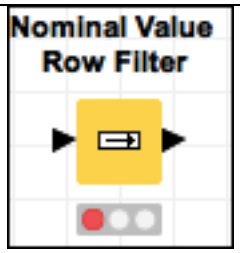
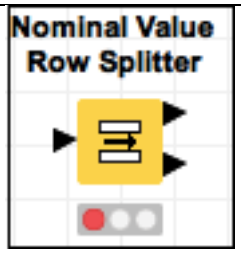
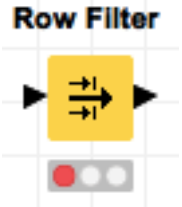
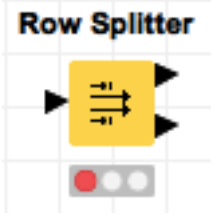
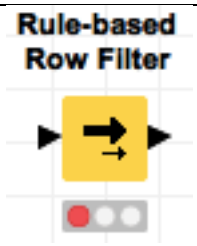
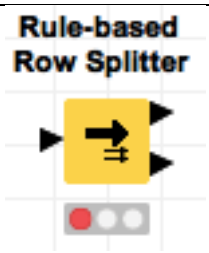


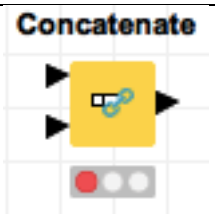
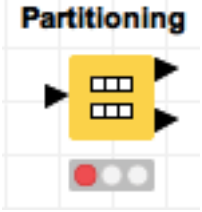
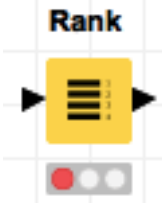
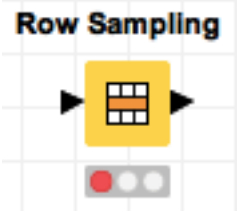
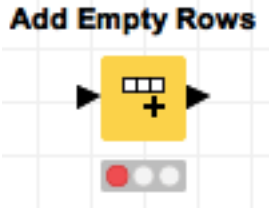
Denormalizer

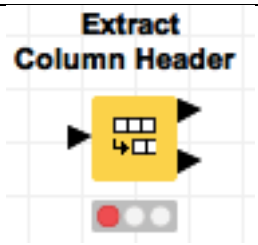
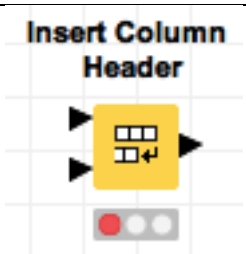
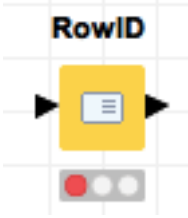
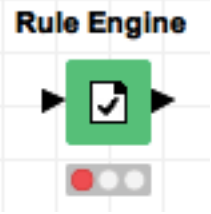
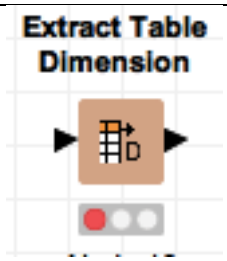
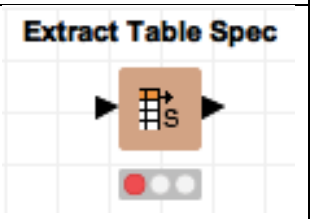
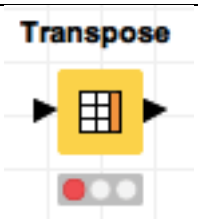
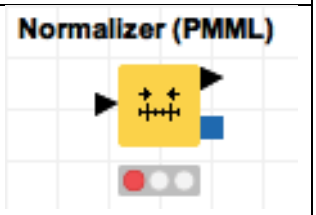
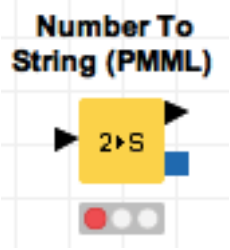
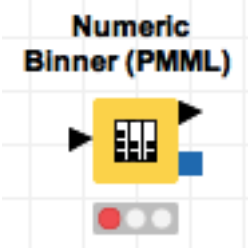


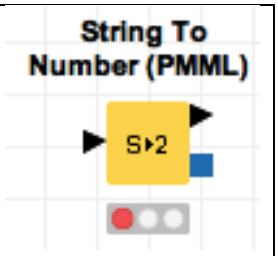
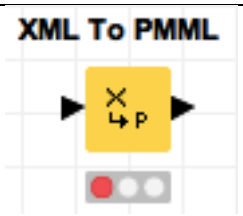
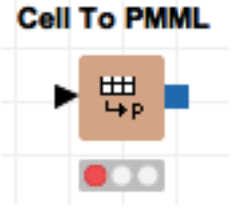
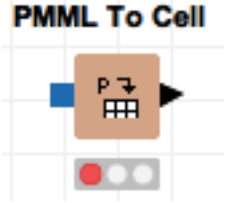
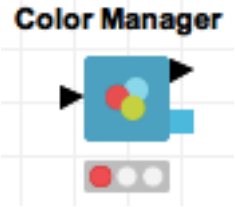
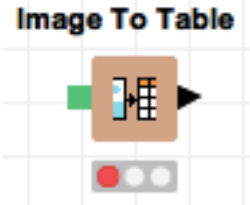
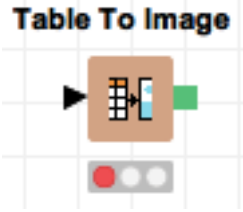
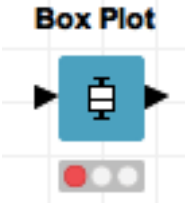
Normalizer

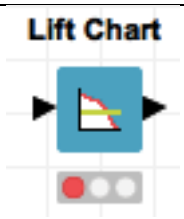
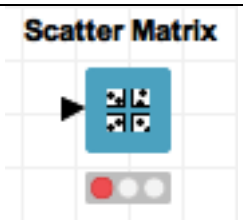
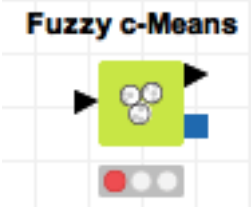
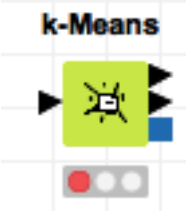
Missing Value

One to Many	
Many to One	
Set Operator	
SMOTE	
Nominal Value Row Filter	
Nominal Value Row Splitter	
Row Filter	
Row Splitter	
Rule-Based Row Filter	
Rule-Based Row Splitter	

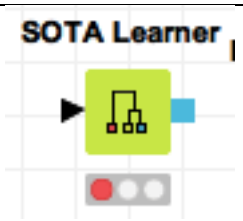
Concatenate 	GroupBy 
Ungroup 	Partitioning 
Pivoting 	Unpivoting 
Rank 	Row Sampling 
Sorter 	Add Empty Rows 

Extract Column Header	
Insert Column Header	
RowID	
Rule Engine	
Extract Table Dimension	
Exract Table Spec	
Transpose	
Normalizer (PMML)	
Number to String (PMML)	
Numeric Binner (PMML)	

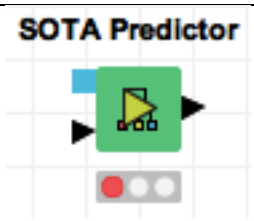
String to Number (PMML)	
XML to PMML	
Cell to PMML	
PMML to Cell	
Color Manager	
Size Manager	
Image to Table	
Tablet to Image	
Box Plot	
Histogram	

Lift Chart 	Line Plot 
Pie Chart 	Scatter Matrix 
Scatter Plot 	Naive Bayes Learner 
Naive Bayes Predictor 	Fuzzy c-Means 
k-means 	Hierarchical Clustering 

SOTA Learner



SOTA Predictor



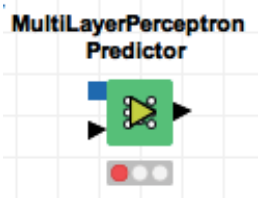
Hierarchical Cluster View

Hierarchical Cluster Assigner

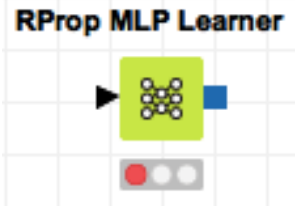
Fuzzy Rules Learner

Fuzzy Rule Predictor

MultiLayer Perceptron Predictor

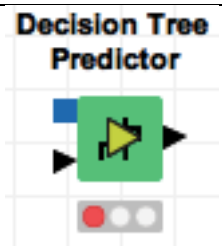
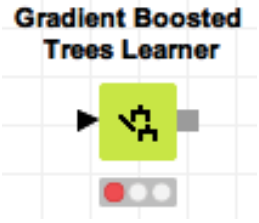
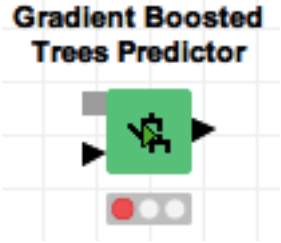
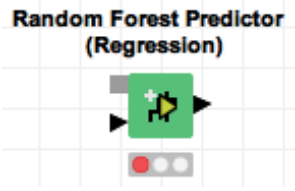


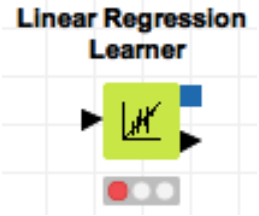
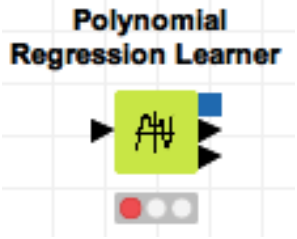
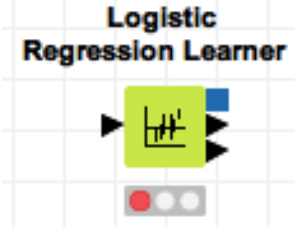
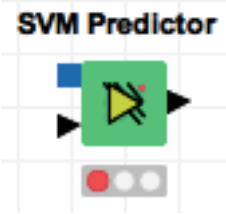
RProp MLP Learner



PNN Learner

PNN Predictor

Decision Tree Learner	
Decision Tree Predictor	
Gradient Boosted Trees Learner	
Gradient Boosted Trees Predictor	
Gradient Boosted Trees Learner (Regression)	
Gradient Boosted Trees Predictor (Regression)	
Random Forest Learner	
Random Forest Predictor	
Random Forest Learner (Regression)	
Random Forest Predictor (Regression)	

K Nearest Neighbor	
Association Rule Learner	
Linear Regression Learner	
Polynomial Regression Learner	
Regression Predictor	
Logistic Regression Learner	
Logistic Regression Predictor	
SVM Learner	
SVM Predictor	
Feature Selection Loop Start (1:1)	

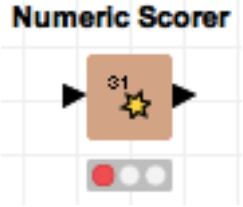
Feature Selection Loop Start (1:2)

X-Partitioner

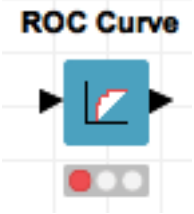
X-Aggregator

Scorer

Numeric Scorer

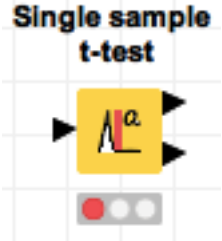


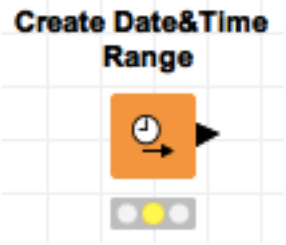
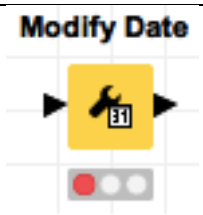
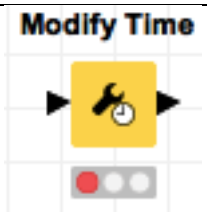
ROC Curve

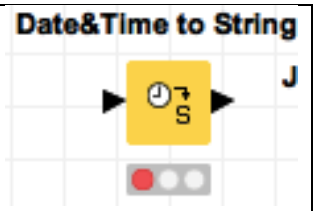
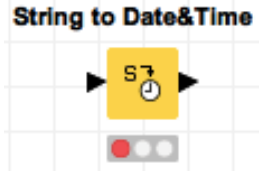
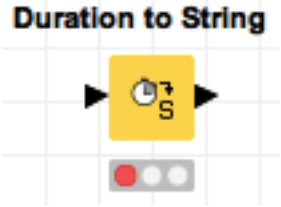
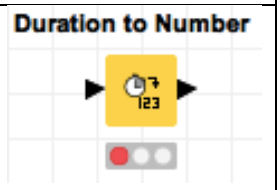
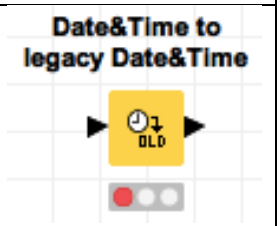


Weka

Single Sample t-Test



Independent Groups t-Test	
Paired t-Test	
One-way ANOVA	
Create Date&Time Range	
Date&Time Difference	
Date&Time Shift	
Modify Date	
Modify Time	

Modify Time Zone 	Date&Time to String 
String to Date&Time 	Duration to String 
Duration to Number 	String to Duration 
Date&Time to legency Date&Time 	Java Snippet 
Java Snippet Row Filter 	Python Edit Variable