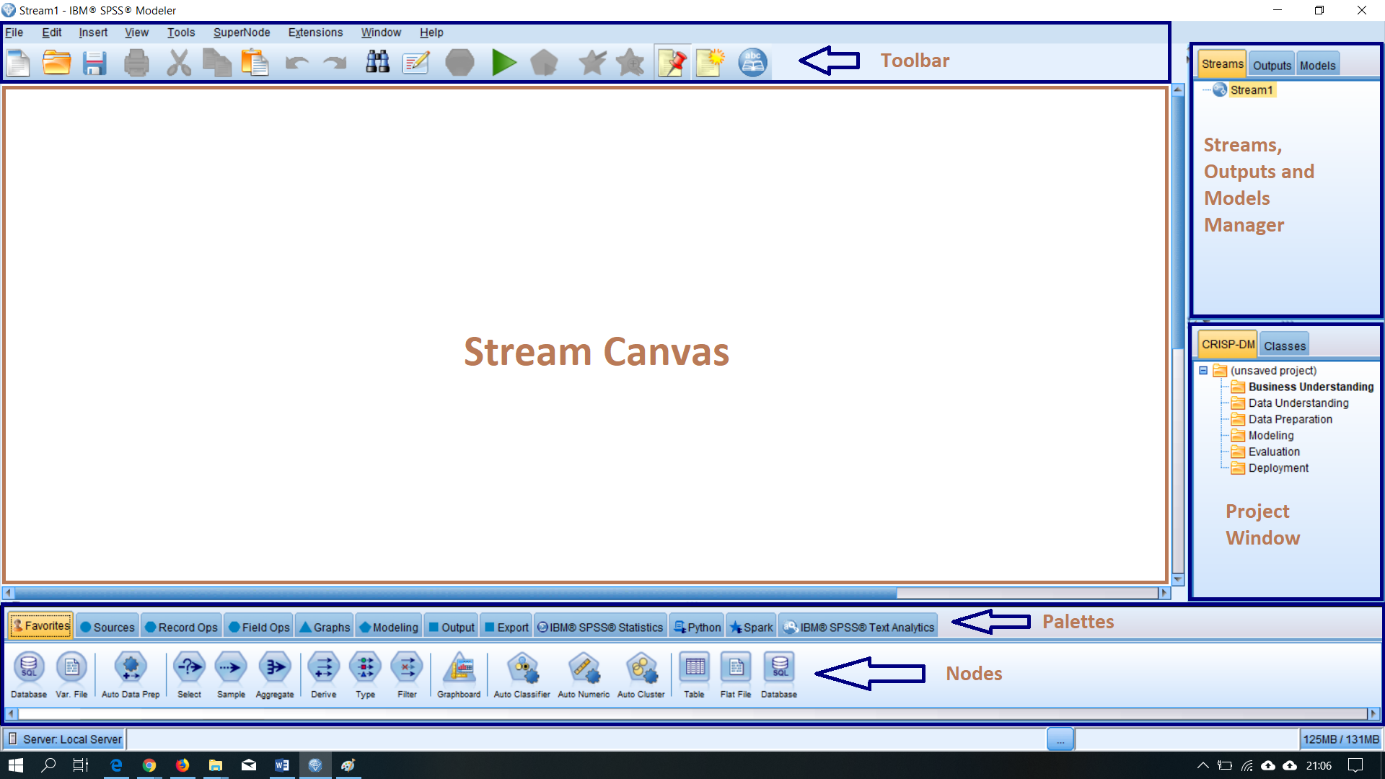
**IBM SPSS Modeler ile Birliktelik Kuralları Analizi (Association Rules Analysis with IBM SPSS Modeler)**

**IBM SPSS Modeler**, SPSS ailesinin veri madenciliği süreçlerinde (bu sürecin **multidisipliner** olduğunu hatırlatmak gerek) içerisinde **Supervised**, **Association** ve **Segmentation** algoritmaları ile bazı **Python** {XGBoost (Linear ve Tree), Random Forest ve t-SEN} ve bazı **Apache Spark** {K-Means, XGBoost, Isotonic} üzerinde kullanılan bazı algoritmalarının yer aldığı (**v18.1.1**) ticari bir yazılımdır. Drag&Drop mantığı ile hızlıca modeller çıkarabileceğiniz kullanışlı kolay bir yazılımdır. Ayrıca **R Project** ve **Python** ile yazmış olduğumuz scriptleri Modeler’ın içerisine implemente edebilirsiniz.

Bu ve benzeri (SAS E-Miner, Knime, RapidMiner) veri madenciliği araçlarının avantajları olduğu gibi dezavantajları var. **Bazı avantajları**: Hızlı ve pratik bir şekilde modelleme yapılması, kişiden bağımsız olarak geliştirilen modelin drag&drop mantığı olarak değiştirilebilmesi, hızlıca production’a alınabilmesi, stabil (paket bağımlılıklarının olmaması sebebiyle) olarak çalışabilmesi sayılabilir. **Dezavantajları ise**; Piyasa da yer alan tüm algoritmaların yer almaması, ticari bir ürün olması nedeniyle maliyeti, …

IBM’in SPSS’i (Statistical Package for the Social Sciences) satın aldıktan sonra yeni adı **IBM SPSS Modeler** (daha önceki adı SPSS Clementine) olarak değiştirilmiştir.



IBM SPSS Modeler;

1. Toolbar,
2. Stream Canvas,
3. Streams, Outputs and Manager,
4. Project Window (CRISP-DM),
5. Palettes ve altında yer alan Nodes kısımlarından oluşmaktadır.

Daha fazla bilgi için: <https://www.ibm.com/products/spss-modeler>

**Birliktelik Kuralları Algoritmaları**

SPSS Modeler’ın altında Association Rules algoritmaları altında **Apriori**, **Carma**, **Sequence** ve **Association Rules** olmak üzere 4 tane algoritması bulunmaktadır.

**Apriori** algoritması kategorik değişkenler çalışmasının yanı sıra diğer algoritmalara göre daha hızlı ve etkili çalışıyorken;

**Carma** algoritmasının çoklu Consequent(RHS - Right Hand Side) çıktısı verbilmekte;

**Sequence** algoritması da Carma algoritmasına benzerliğinin yanı sıra işin zaman boyutunun girmesi ile diğerlerine göre farklılık gösterir.

**Veri Seti: Basket1n**

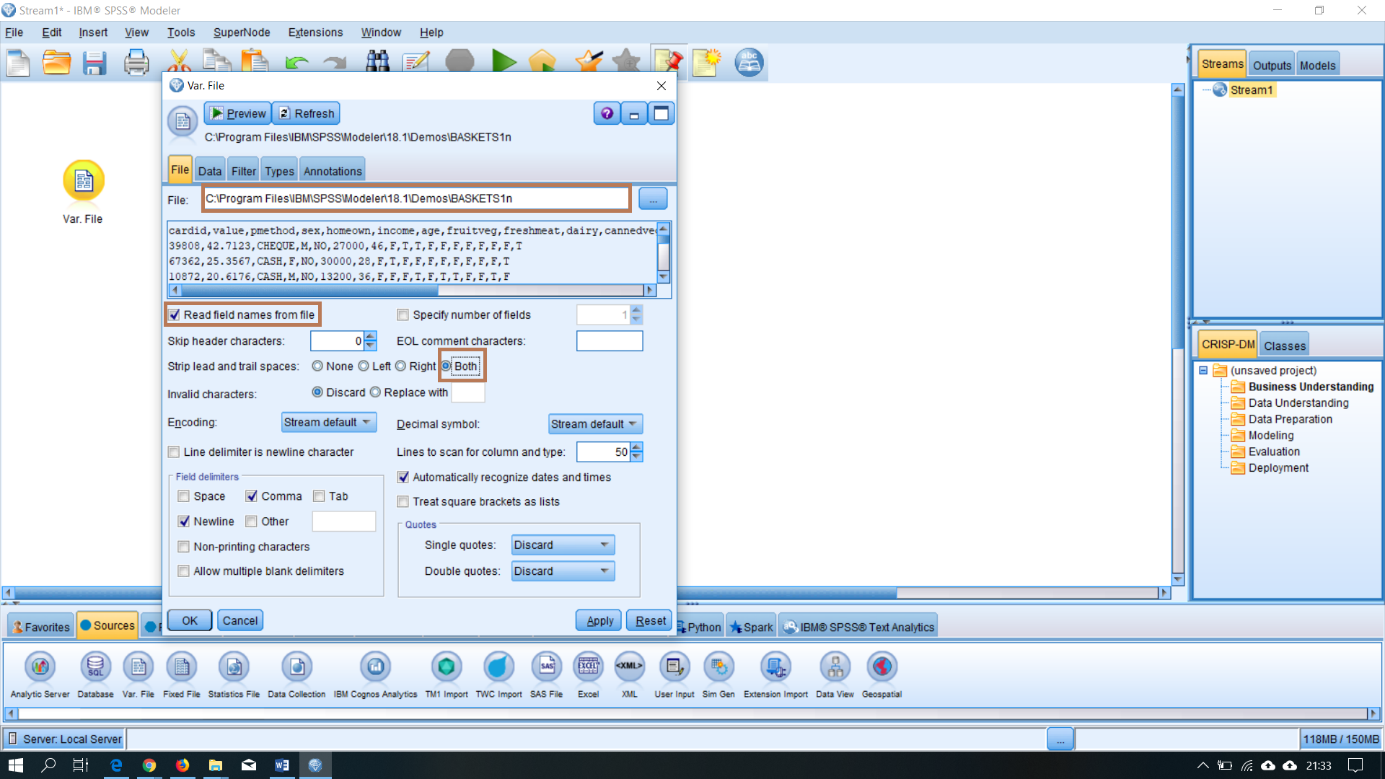
Öncelikle veri setimiz olan **BASKET1n** dosyasına bir göz atalım. Veri setine “KurulumDizini\IBM\SPSS\Modeler\18.1\Demos” dizini içerisinde csv (comma seperated value) formatında yer almaktadır.

**Drag & Drop Modelleme**

Veri setini içeriye aktarmak için;

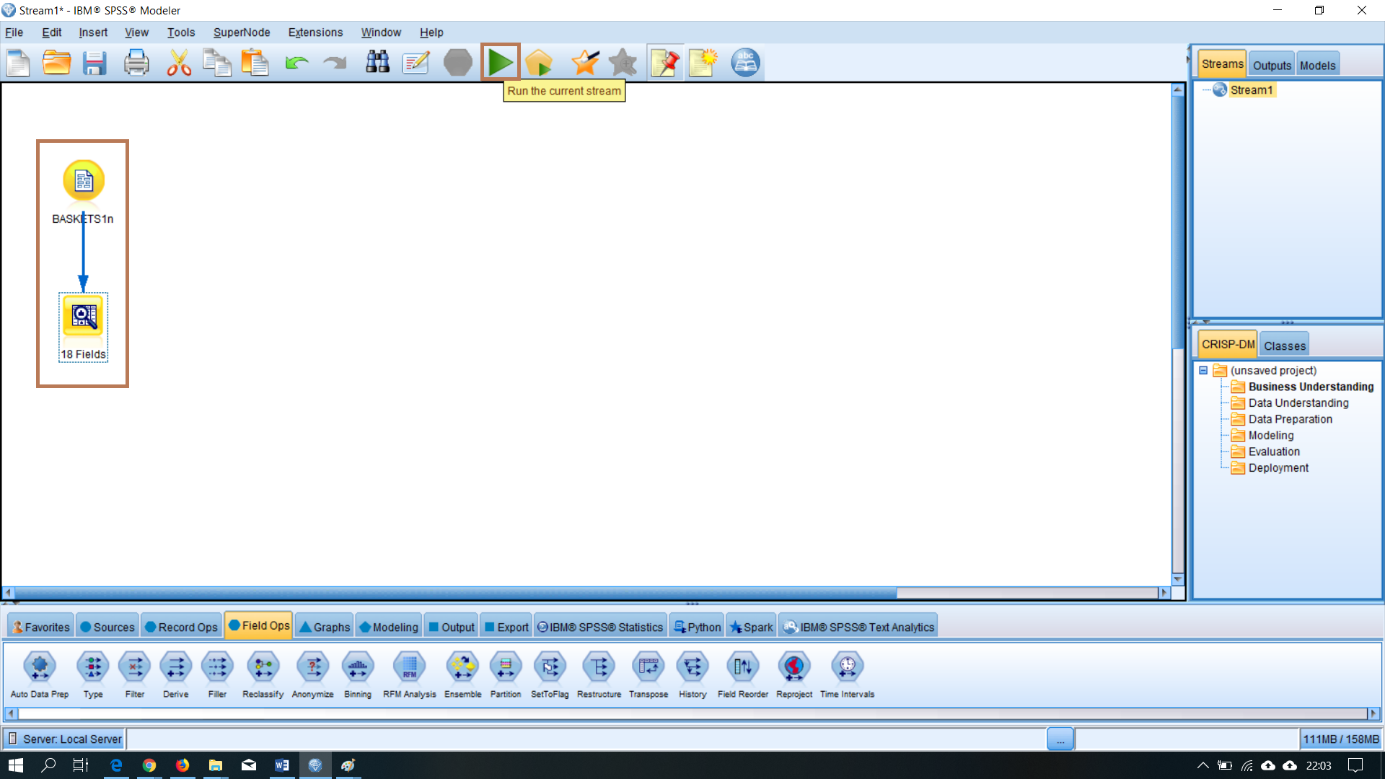
**Sources** Paleti altında **Var.File** node’na çift tıklanarak Stream Canvas eklenir. Stream Canvas üzerinde yer alan **Var.File** node’na çift tıklanarak aşağıdaki adımlar uygulanır.

1. Devamında Basket1n dosyasının **File Path** belirlenir,
2. **Read field names from** file işaretlenir,
3. Strip Lead and trail spaces: **Both** olarak seçilir,
4. Field delimiters: **Comma** ve **Newline** işaretlenir ve Apply ve OK butonlarına tıklanır.



Basket1n veri setini içeriye aktardık. Şimdi ise veri setini biraz inceleyim;

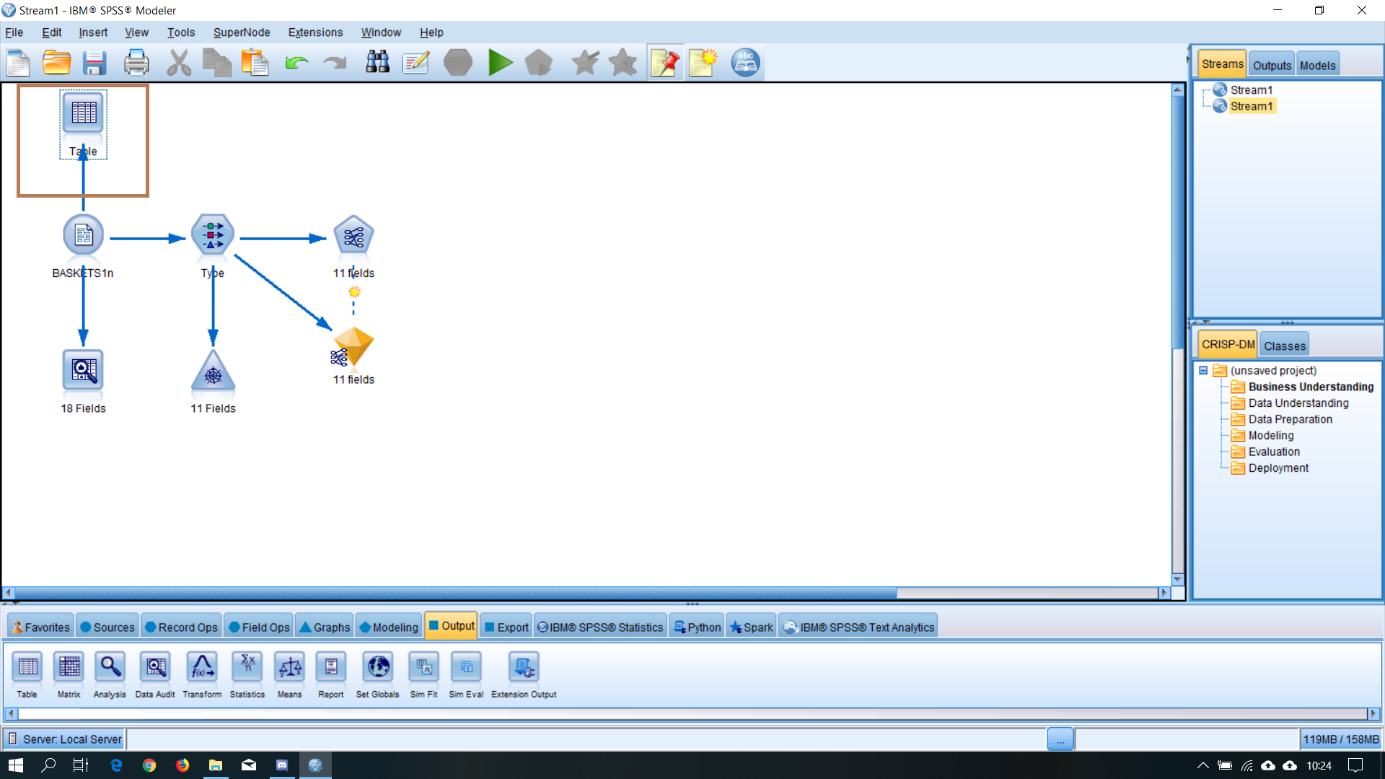
**Output** paleti altında yer alan **Data Audit** çift tıklayarak **Stream Canvas**’a ekliyoruz. Birbirine bağlı olan **Var.File** node’u ile **Data Audit** node’u Mouse yardımı ile seçip Play butonuna tıklıyoruz.

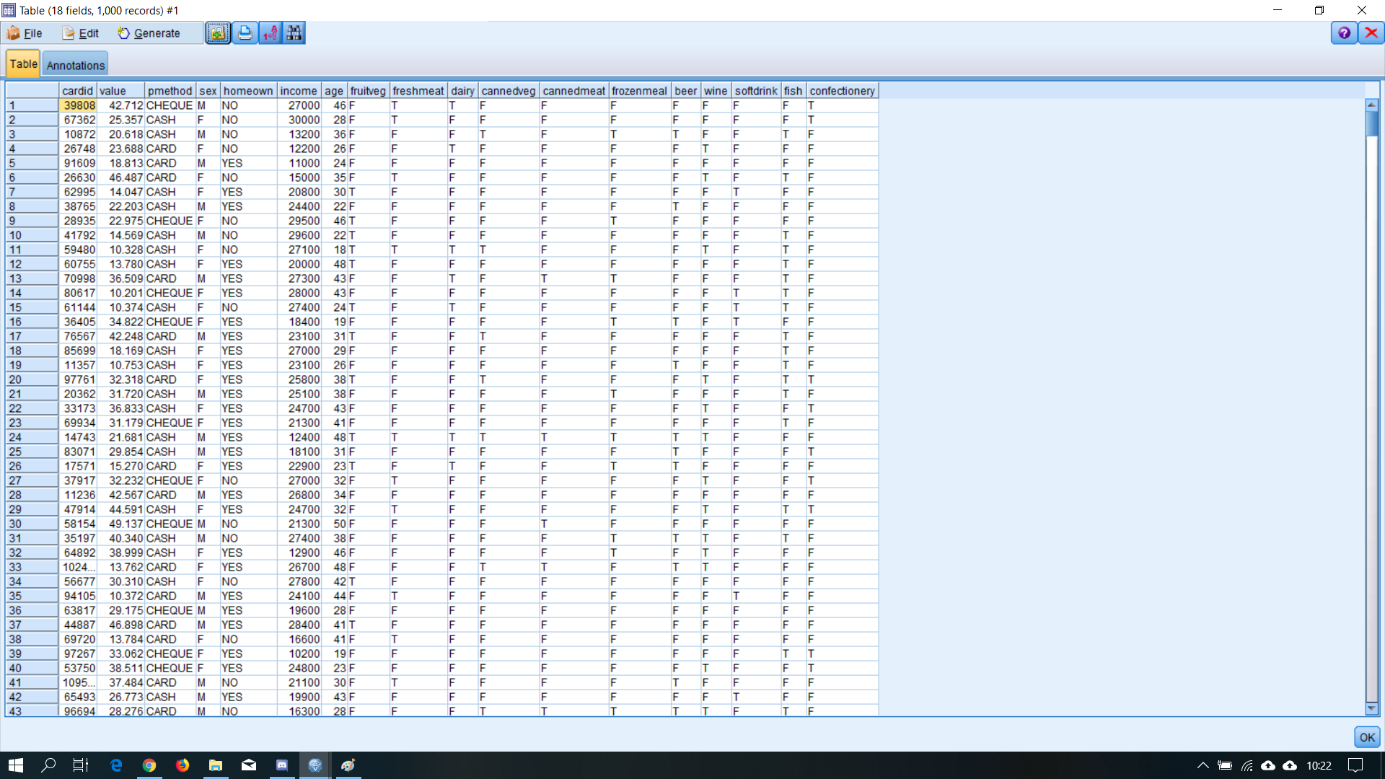




Veri seti ait **Field**(Değişken), **Graph**(Değişkenlere Değerlerime ait Grafik), **Measurement**(Değişkenlerin Tipi), **Min**(En Küçük Değer), **Max**(En Büyük Değer), **Mean**(Ortalama), **Std. Dev.**(Std. Sapma), **Skewness**(Çarpıklık), **Unique**(Aldığı Farklı Değer Tipi), **Valid**(Değişkene ait Gözlem Sayısı) istatistiksel bilgilerileri içermektedir. **Data Audit** node’u Buradan aykırı değer olup olmadığı, eksik değer olup olmadığı gibi durumların farkına varmamızı sağlar.

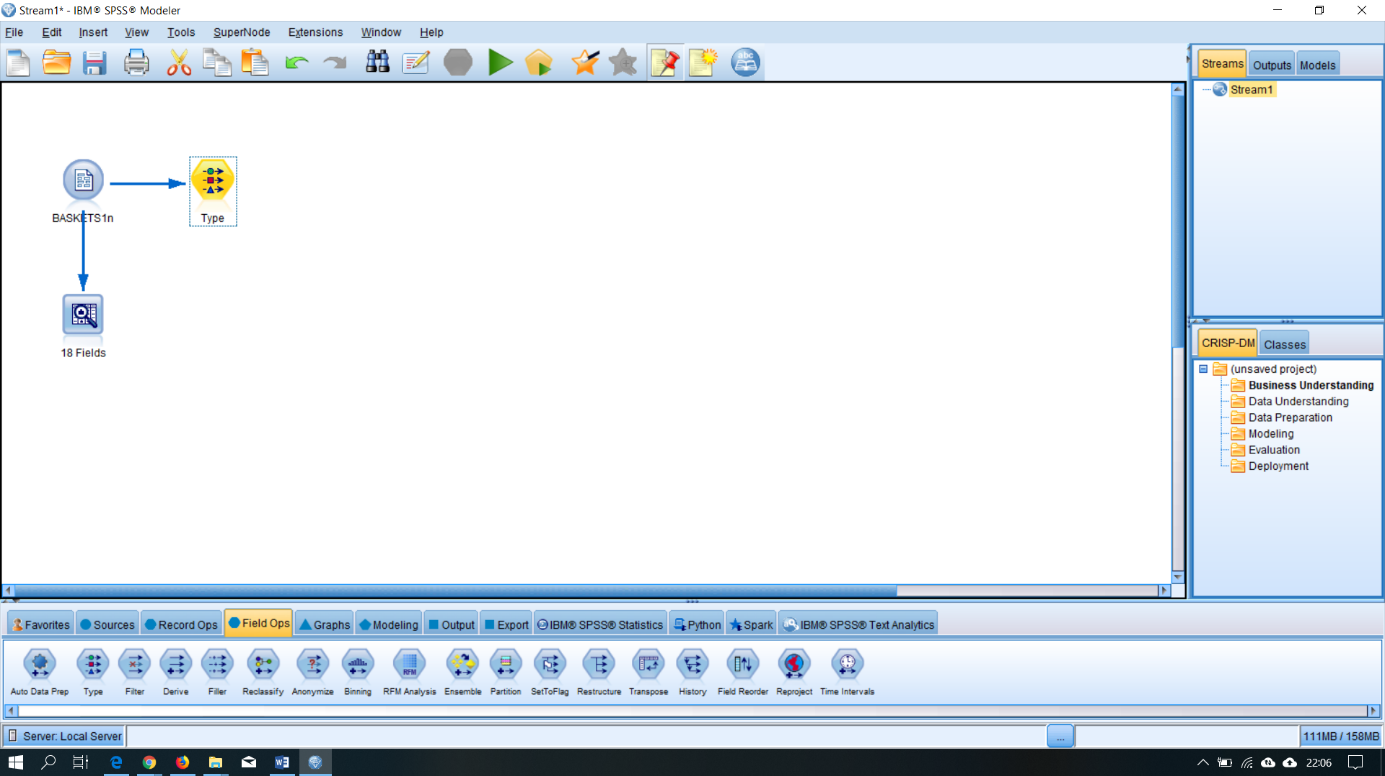
**Output** paleti altında yer alan **Table** node’unu çift tıklayarak **Var.File** node’una ekliyoruz. Devamında node’un üzerine ise çift tıklayıp özellerinden **Run** butonuna basıyoruz. Veri setimize ait değişkenler ve gözlemler karşımızda.



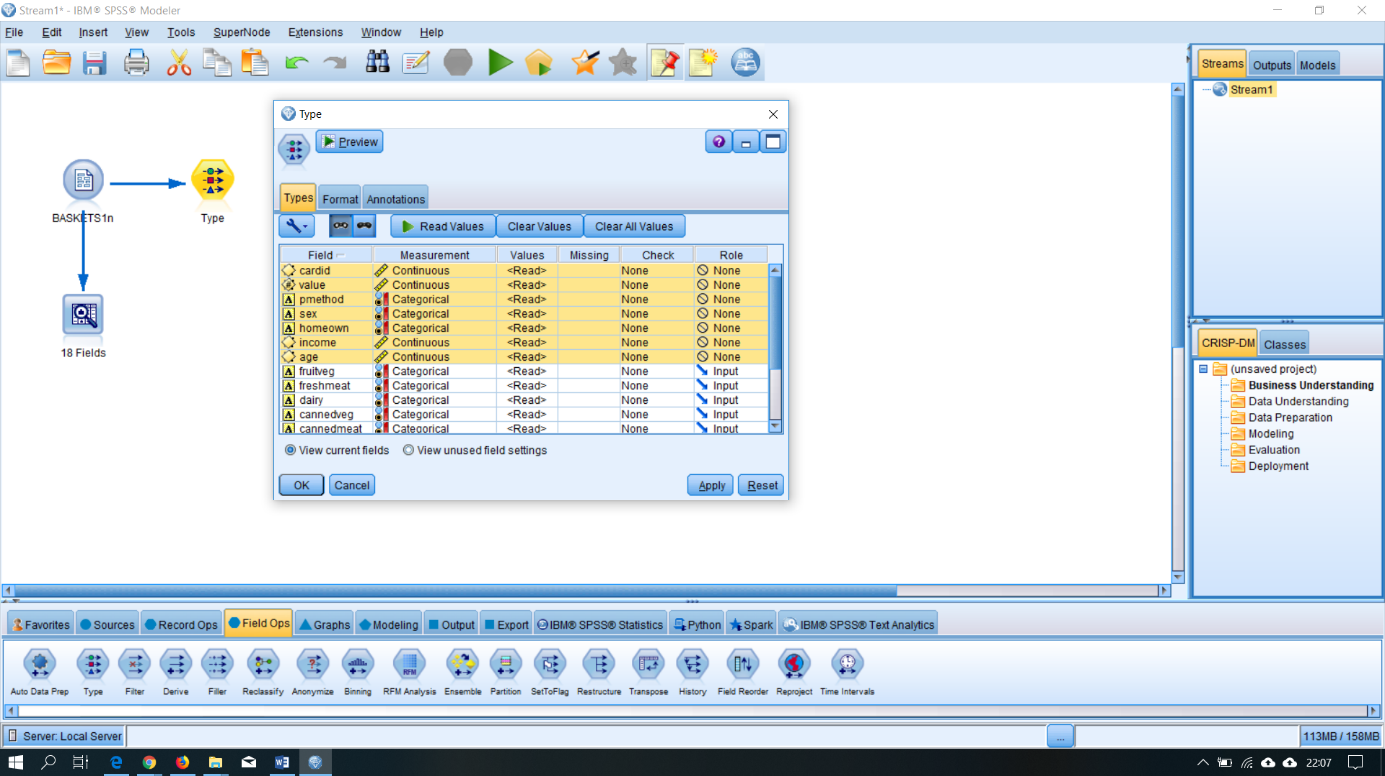


Birliktelik Kuralları için veri seti içerisinde yer alan bazı ilgisiz değişkenleri veri seti içerisinden çıkaralım.

**Field Ops** paleti altında yer alan **Type** node'nu **Stream Canvas**’a ekliyoruz.

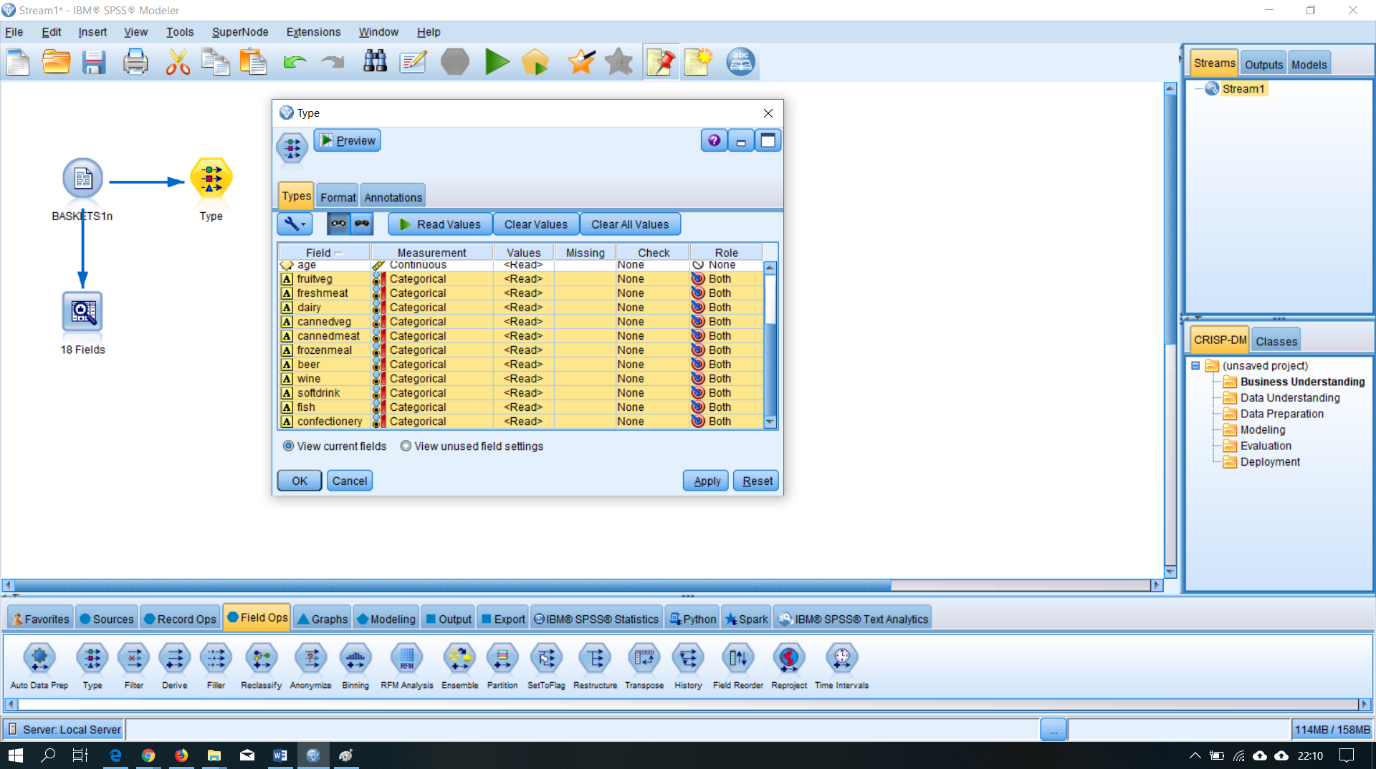


**Type** Node’a çift tıklayıp, **cardid**, **value**, **pmethod**, **sex**, **homeown**, **income** ve **age** değişkenlerini seçip Role kısmında modele girmemesi için **None** olarak işaretliyoruz.

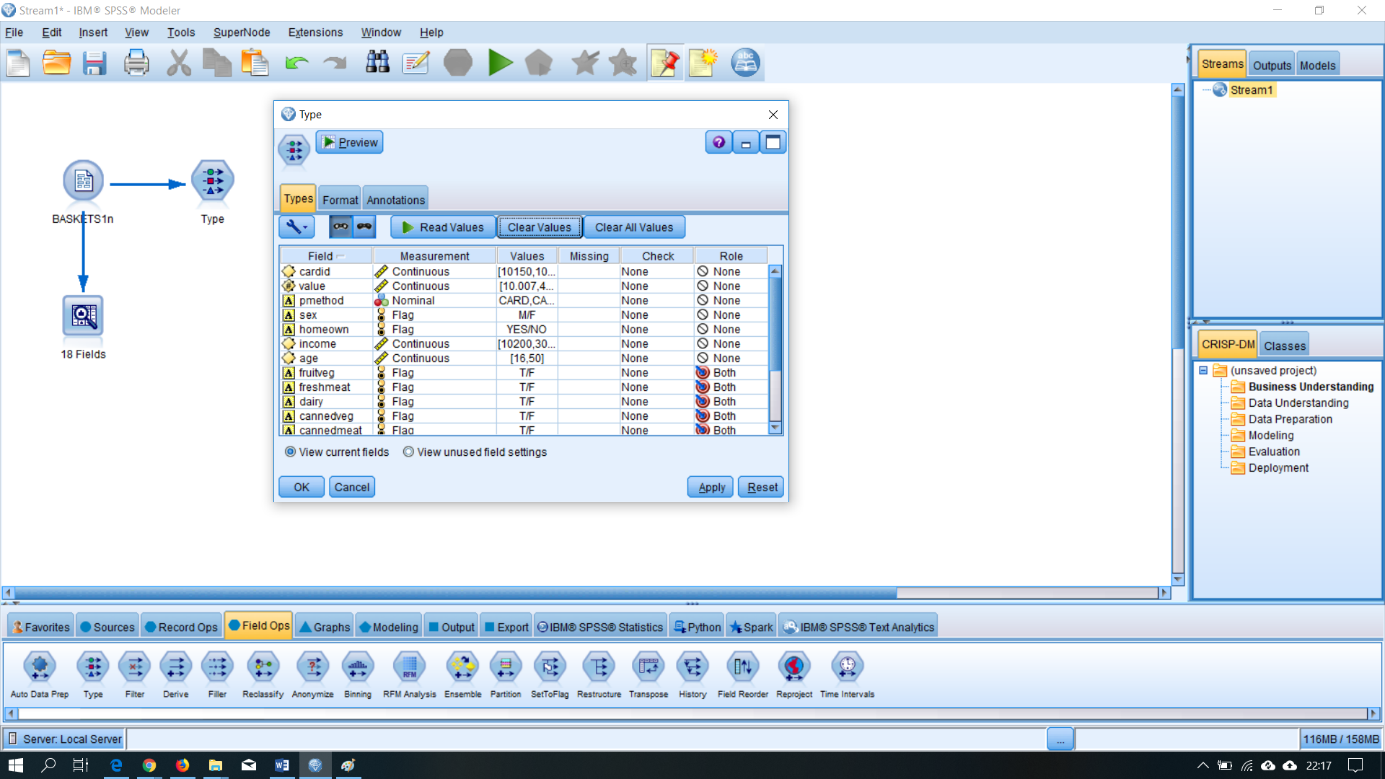


Diğer değişkenlerimiz olan **fruitveg**, **freshmeat**, **dairy**, **cannedveg**, **cannedmeal**, **frozenmeal**, **beer**, **wine**, **softdrink**, **fish**, **confectionery** değişkenleri seçip **Role** bilgilerini **Both** olarak işaretliyoruz. Ne demek istedik;

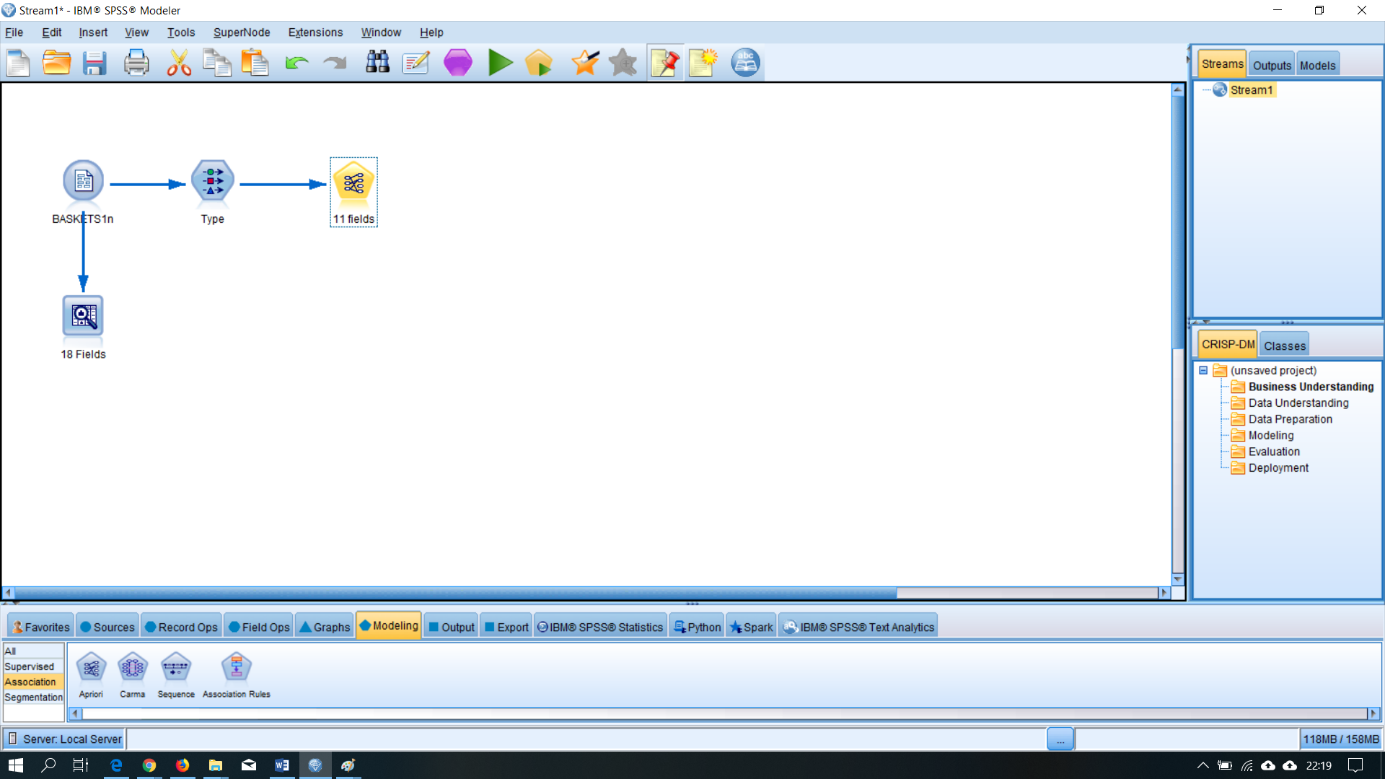
İlk yazımız olan Birliktelik Kuralları Analizi <https://www.veribilimi.co/associationrulesanalysis/> değişken yönlerinin In, Out ve Both olmak üzere 3 farklı değer alabileceğini belirtmiştik. Burada yer alan değişkenlerin Apriori algoritması içerisinde hem **input** hem de **output** olarak yani kuralın hem sağında hem de solunda yer almasını istediğimiz için **Both** olarak işaretliyoruz.

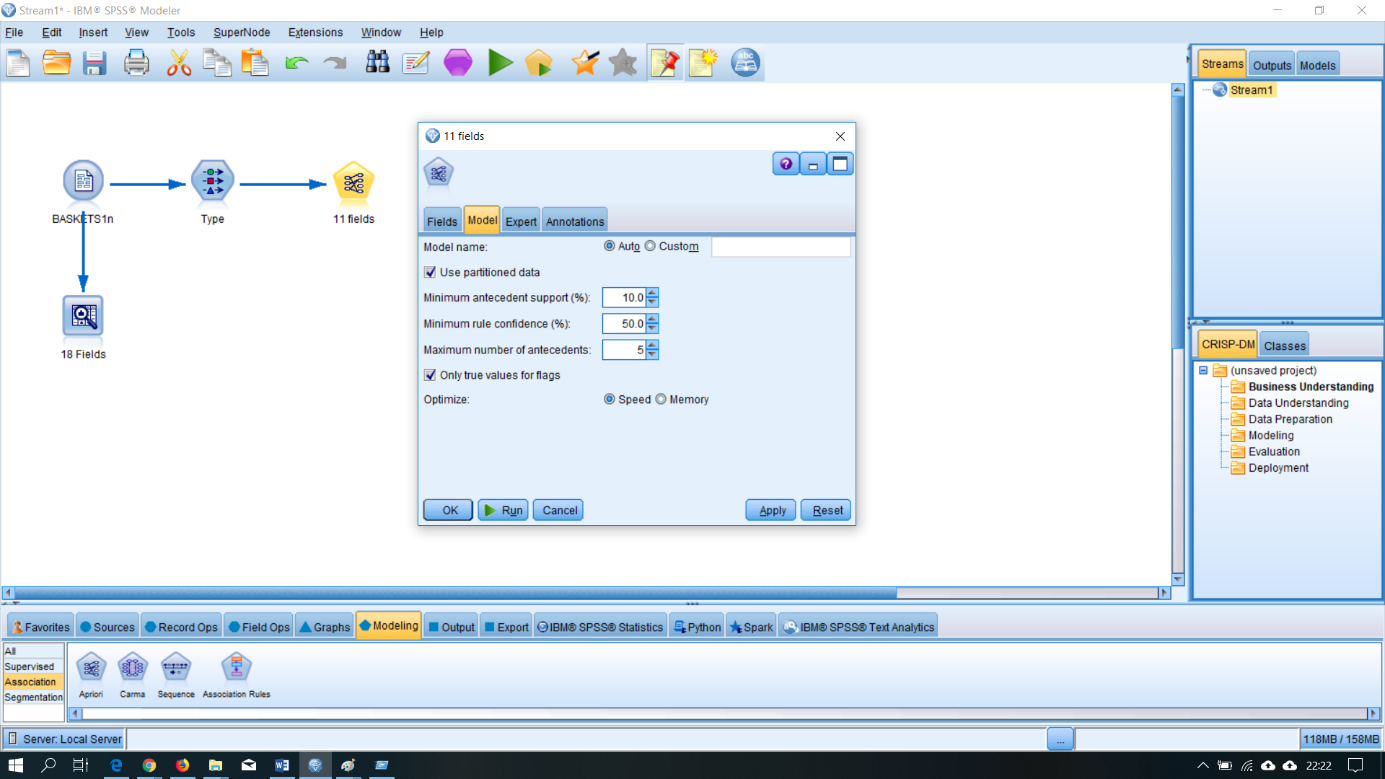


Type node’undaki son işlemimiz ise, Clear Values, Read Values butonlarına tıklıyoruz ve değişkenlere ait **Measurement**(tip) ve **Values**(değer) bilgilerini almasını sağlıyoruz.

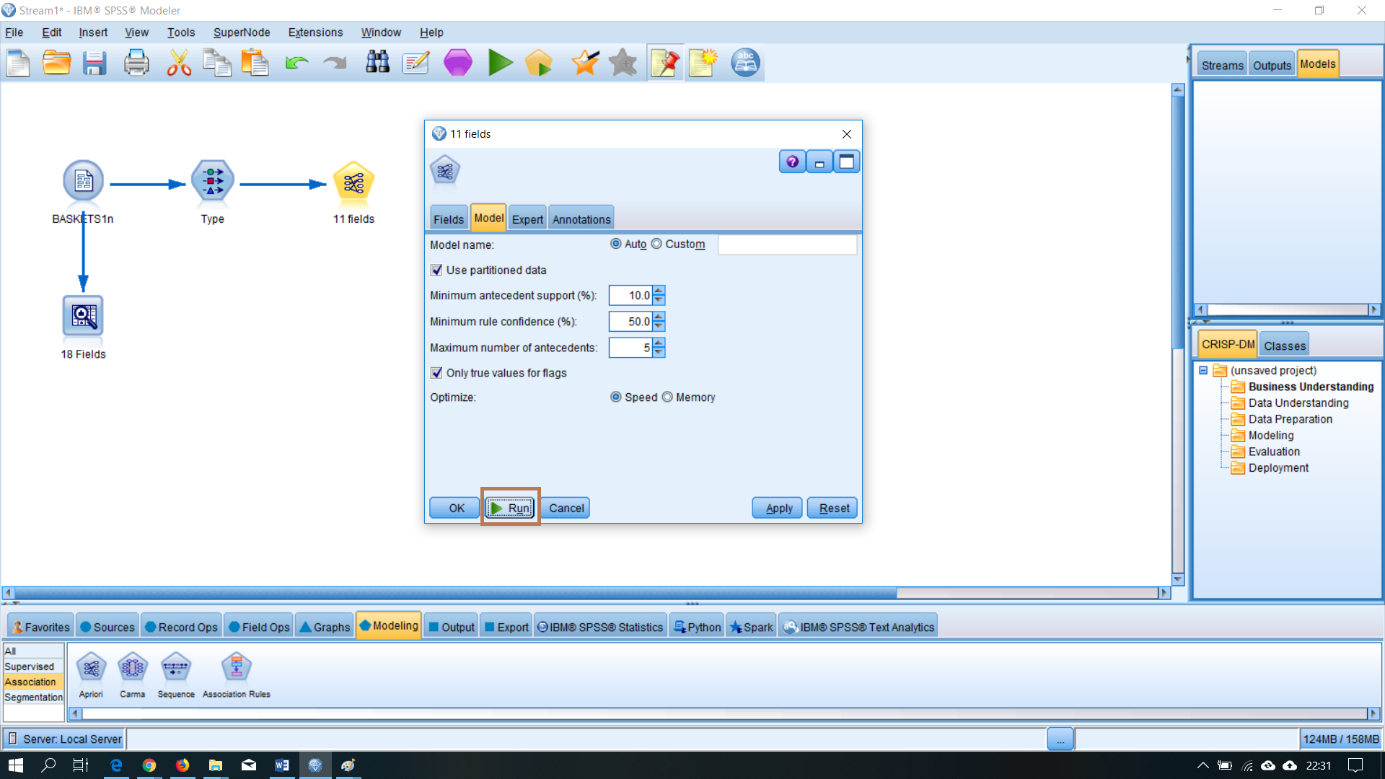


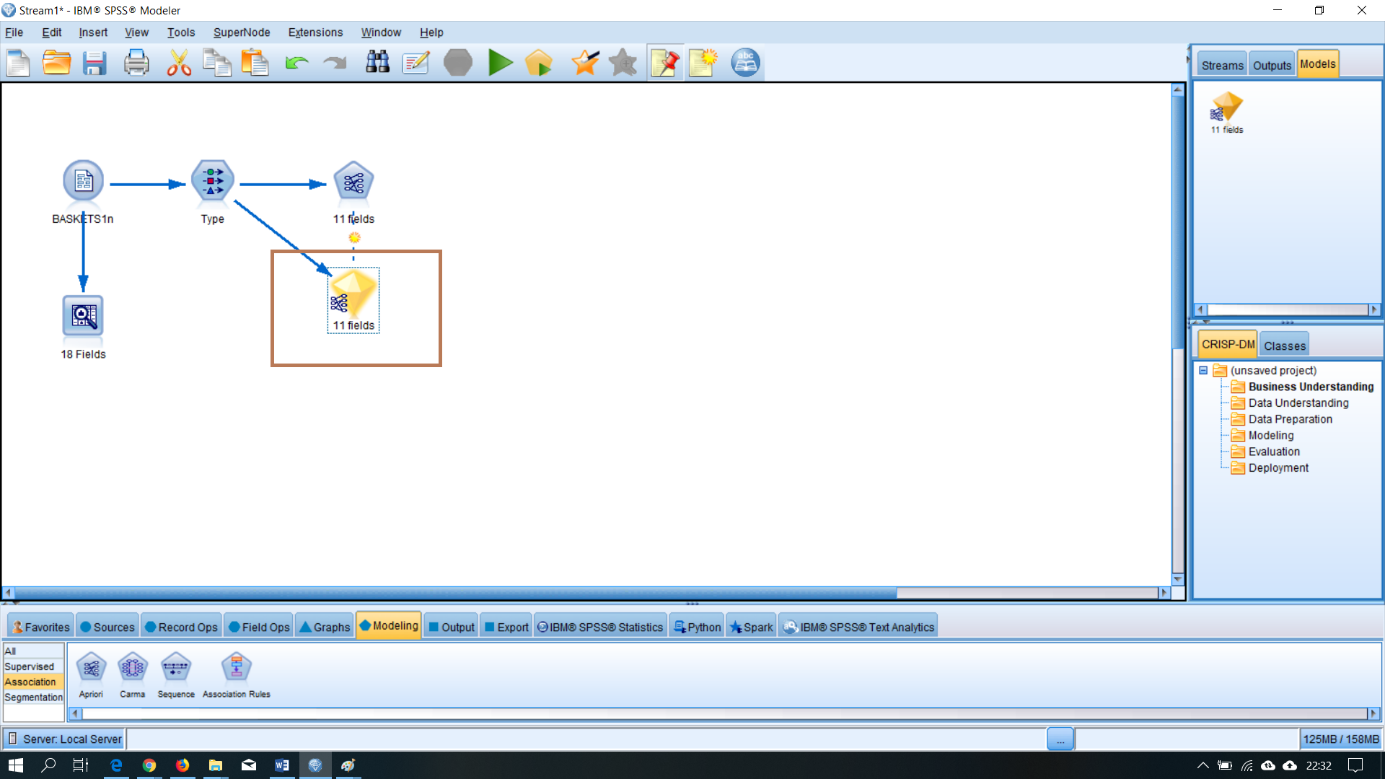
Modelleme de sona doğru yaklaşıyoruz. Şimdi **Modeling** paleti altında yer alan **Association** alt kısmı içerisindeli **Apriori** node’unu **Stream Canvas**’a çift tıklayarak ekliyoruz.



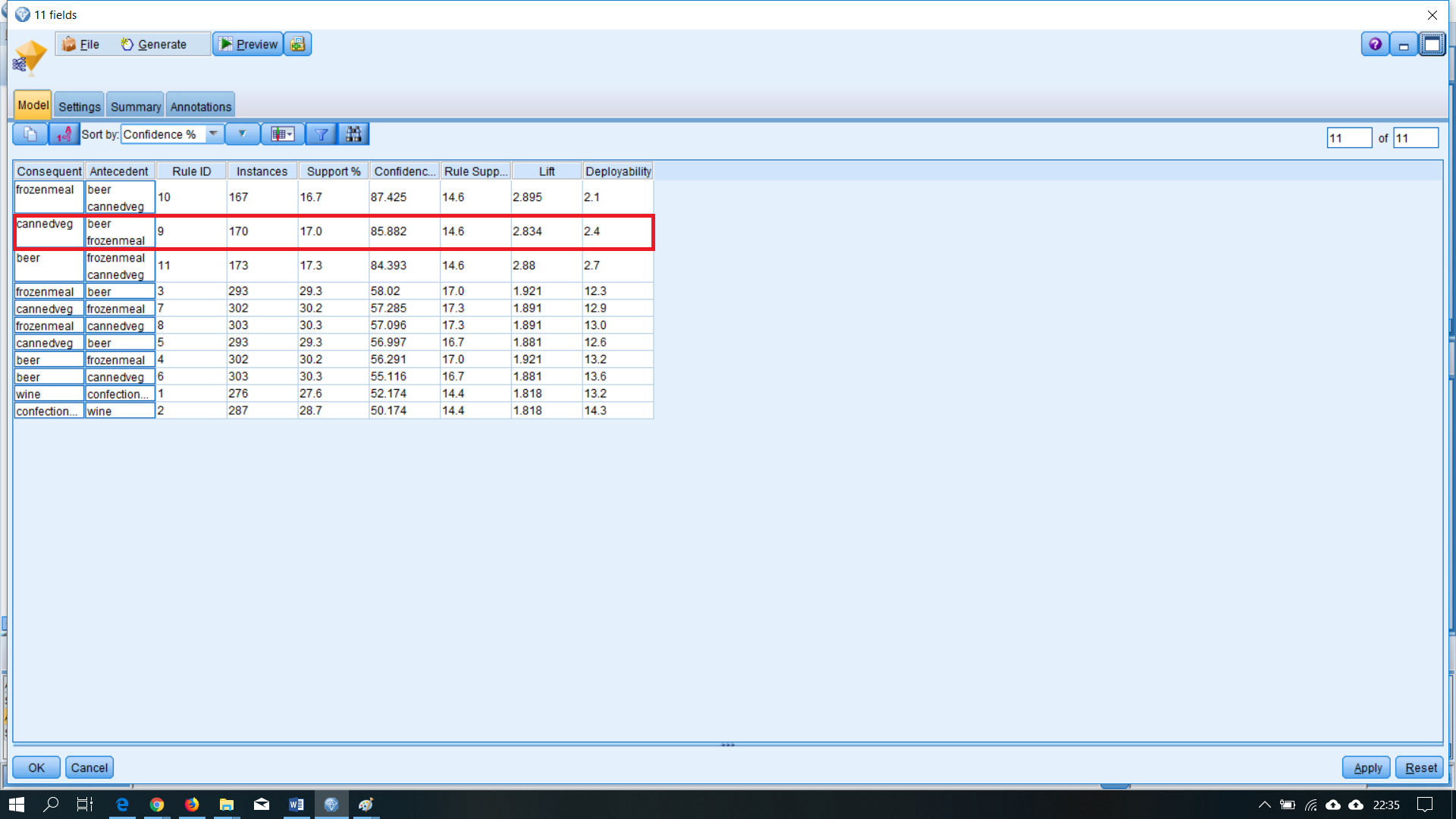
Devamında **Apriori** node’una çift tıklayarak node ait özellikler penceresine giriyoruz. Burada Apriori algoritmasının Min. Support ve Min. Confidence başlangıç değerlerini belirliyoruz. Min. Support değerini **Apriori** default(varsayılan) değeri olan %10, Min. Confidence değerini %50 olarak ayarlıyoruz. 

Diğer ayarlara gelecek olursak; **Use partitioned data** default olarak işaretli gelmekte ve **Maximum number of antecedents** değerini yani oluşan kuralın RHS(Right Hand Side) kısmında en fazla 5 tane items olabileceğini belirtmektedir. **Only true values for flags** burada ise veri seti içerisinde yer alan değişkenlere ait **Measurement** tipinin **Flag** (İkili) olduğu yani Satın Alındığı (**True**) ya da Satın Alınmadığı (**False**) değerine sahip olduğumuzu bildiğimiz üzere işaretli olmasında hiçbir sakınca yok. Varsayılan olarak işaretli gelmektedir. Optimize kısmında **Speed** ya da Memory tercihimiz default olarak Speed olarak seçili gelmektedir. Son olarak **Run** butonuna tıklıyoruz ve başlangıç değerleri ile kurallar oluşmaya başlıyor.

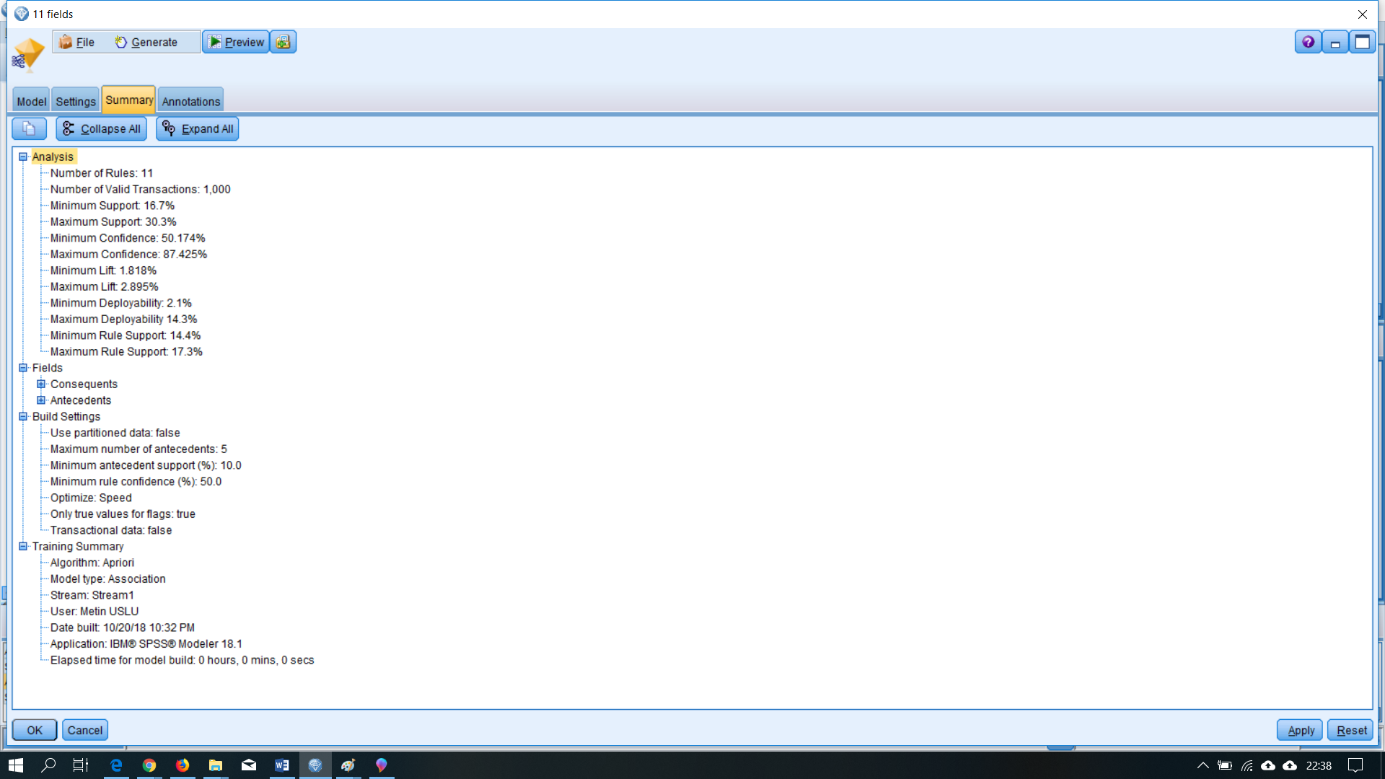




Algoritma çalıştı ve nihayetinde bir **elmas** oluştu. Elmasın üzerine çift tıkladığımızda **Model** sekmesinde oluşan kuralları görebiliriz.



Aynı şekilde modele ait bilgileri **Summary** sekmesinden ulaşabiliriz.



Şimdi kırmızı çerçeve içerisine alınan kuralı kısmı yorumlayalım;

1. **Instances**: **Beer** ve **FrozenMeal** items’ları veri seti içerisinde 170 kere birlikte görüldüğü;
2. **Support**: **Beer** ve **FrozenMeal** items'ları birlikte görülme olasılığı %17 olduğu;

**Nasıl**: 170/1000 = 0,17 yani %17 olduğu;

1. **Confidence**: **Beer** ve **FrozenMeal** items'ları birlikte satın alan kişilerin %85,882 olasılıkla **CannedVeg** items'ını da satın aldığı;
2. **Rule Support**: **Beer**, **FrozenMeal**, ve **CannedVeg** items'larının bir veri seti içerisinde beraber görülme olasılığı %14,6 olduğu;
3. **Lift**: **Beer** ve **FrozenMeal** items’larının yer aldığı alışveriş sepetlerinde **CannedVeg** items’ının satışı 2,834 kat arttığı;
4. **Deployment**: Veri seti içerisinden yer alan Antecedent(Öncül)’ların ne kadarının karşılandığı ve Consequent(Soncul) ‘ların ne kadarının karşılandığını ölçüsüdür. Yani müşterilerin ne kadar Antecedent items’larına sahip olduğu ancak Consequent items’larına sahip olmadığını belirtir.

**Nasıl**: (Support – Rule Support)\*100 = (17,0 – 14,6) = 2,4 olduğu görülmüştür.

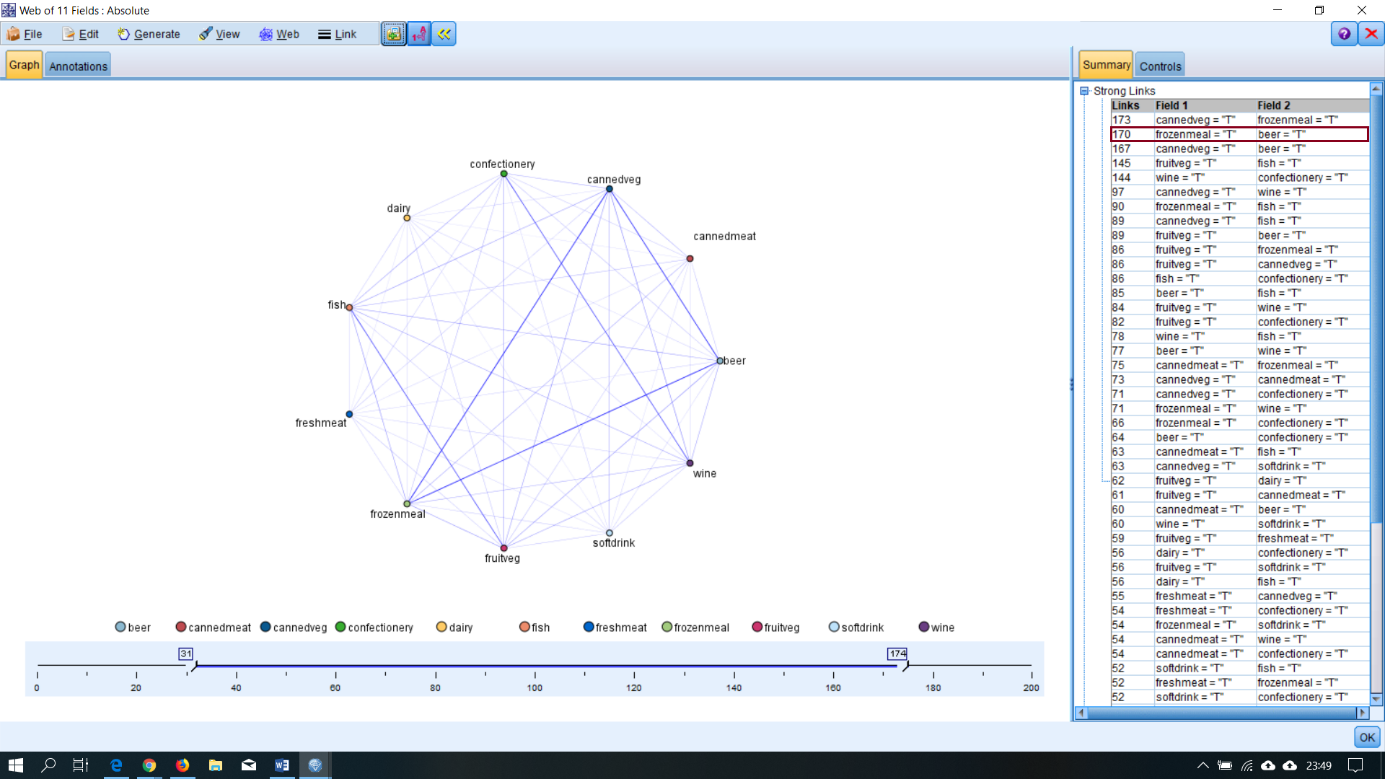
**Graph** paleti altında yer alan **Web** node'u **Type** node'na bağlıyoruz. Devamında Web node’na çift tıklayıp veri seti içerisinde yer alan kategorik Flag tipindeki (True ve False) değişkenlerimizi ekliyoruz.



Devamında **Run** butonuna tıklayarak **Web** node’umuzu çalıştırıyoruz.



Web node’umuza ait çıktılar aşağıdaki gibidir.



Yukarıdaki (Resim 17) resimde sağ kısımda kırmızı renk ile çerçeve içerisine alınan kısım; FrozenMeal ve Beer item’ların birlikte veri seti içerisinde geçme sıklığının 170 olduğu ve en yüksek adete sahip 2 Links olduğu gözükmektedir.

Şimdi de oluşan modelimizi **PMML** (Predictive Model Markup Language) olarak kaydediyoruz. PMML, **Data Mining Group** (**DMG**) tarafından desteklenen ve geliştirilmeye devam edilen ve şu anda 4.3 versiyonuna sahip olan **XML** (Extensible Markup Language) tabanlı olup model ait oluşan çıktıları (sistemlerimize entegre edip tavsiyelerde bulunabilmemiz için) dışarıya aktarıp eder. Oluşan **XML** çıktısı;

<?xml version="1.0" encoding="UTF-8"?><PMML xmlns="http://www.dmg.org/PMML-4\_2" version="4.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<Header copyright="(C) Copyright IBM Corp. 1994, 2015">

<Application name="IBM SPSS Modeler" version="18.1"/>

</Header>

<DataDictionary numberOfFields="11">

<DataField dataType="string" name="fruitveg" optype="categorical">

<Extension extender="spss" name="storageType" value="string"/>

<Extension extender="spss" name="flagTypeTrueValue" value="T"/>

<Value property="valid" value="F"/>

<Value property="valid" value="T"/>

</DataField>

<DataField dataType="string" name="freshmeat" optype="categorical">

<Extension extender="spss" name="storageType" value="string"/>

<Extension extender="spss" name="flagTypeTrueValue" value="T"/>

<Value property="valid" value="F"/>

<Value property="valid" value="T"/>

</DataField>

**…**

**…**

**…**

<Itemset id="7" numberOfItems="2" support="0.167">

<ItemRef itemRef="3"/>

<ItemRef itemRef="5"/>

</Itemset>

<AssociationRule antecedent="1" confidence="0.5217391304347826" consequent="2" id="1" lift="1.8179063778215423" support="0.144"/>

<AssociationRule antecedent="2" confidence="0.5017421602787456" consequent="1" id="2" lift="1.8179063778215423" support="0.144"/>

<AssociationRule antecedent="3" confidence="0.5802047781569966" consequent="4" id="3" lift="1.9212078746920414" support="0.17"/>

<AssociationRule antecedent="4" confidence="0.5629139072847682" consequent="3" id="4" lift="1.9212078746920418" support="0.17"/>

<AssociationRule antecedent="3" confidence="0.5699658703071673" consequent="5" id="5" lift="1.8810754795616083" support="0.167"/>

<AssociationRule antecedent="5" confidence="0.5511551155115512" consequent="3" id="6" lift="1.8810754795616078" support="0.167"/>

<AssociationRule antecedent="4" confidence="0.5728476821192053" consequent="5" id="7" lift="1.8905864096343412" support="0.173"/>

<AssociationRule antecedent="5" confidence="0.570957095709571" consequent="4" id="8" lift="1.8905864096343412" support="0.173"/>

<AssociationRule antecedent="6" confidence="0.8588235294117647" consequent="5" id="9" lift="2.8344010871675405" support="0.146"/>

<AssociationRule antecedent="7" confidence="0.874251497005988" consequent="4" id="10" lift="2.894872506642345" support="0.146"/>

<AssociationRule antecedent="8" confidence="0.8439306358381503" consequent="3" id="11" lift="2.880309337331571" support="0.146"/>

</AssociationModel>

</PMML>

PMML hakkında daha fazla bilgi için; <http://dmg.org/pmml/v4-3/GeneralStructure.html> ve <https://en.wikipedia.org/wiki/Predictive_Model_Markup_Language> bağlantılarını inceleyebilirsiniz.