



CUMHURİYET ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Anabilim Dalı

BİRLİKTELİK KURALLARI MADENCİLİĞİ VE BİR
UYGULAMA

Yüksek Lisans Tezi

Yusuf TAŞ

Sivas
Ocak 2018

CUMHURİYET ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Anabilim Dalı

BİRLİKTELİK KURALLARI MADENCİLİĞİ VE BİR
UYGULAMA

Yüksek Lisans Tezi

Yusuf TAŞ

Tez Danışmanı
Doç. Dr. Mehmet Ali ALAN

Sivas
Ocak 2018

KABUL VE ONAY

Üniversite: : Cumhuriyet Üniversitesi
Enstitü : Sosyal Bilimler Enstitüsü
Ana Bilim Dalı : Yönetim Bilişim Sistemleri Ana Bilim Dalı
Bilim Dalı :
Tezin Başlığı : BİRLİKTELİK KURALLARI MADENCİLİĞİ VE BİR UYGULAMA
Savunma Tarihi : 11/01/2018
Danışmanı : Doç. Dr. Mehmet Ali ALAN

Unvanı - Adı Soyadı

İmza

Jüri Başkanı : Prof. Dr. Aslan GÜLCÜ

Üye : Doç. Dr. Mehmet Ali ALAN

Üye : Yrd. Doç. Dr. Ali Rıza İNCE

Oy Birliği

Oy Çokluğu

Yusuf TAŞ tarafından hazırlanan BİRLİKTELİK KURALLARI MADENCİLİĞİ VE BİR UYGULAMA başlıklı tez, kabul edilmiştir.

.../.../.....

Prof. Dr. Ahmet ŞENGÖNÜL
Enstitü Müdürü

ETİK İLKELERE UYGUNLUK BEYANI

Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü bünyesinde hazırladığım bu Yüksek Lisans/Doktora/Sanatta Yeterlik tezinin bizzat tarafımdan ve kendi sözcüklerimle yazılmış orijinal bir çalışma olduğunu ve bu tezde;

- 1- Çeşitli yazarların çalışmalarından faydalandığımda bu çalışmaların ilgili bölümlerini doğru ve net biçimde göstererek yazarlara açık biçimde atıfta bulunduğumu;
- 2- Yazdığım metinlerin tamamı ya da sadece bir kısmı, daha önce herhangi bir yerde yayımlanmışsa bunu da açıkça ifade ederek gösterdiğimi;
- 3- Başkalarına ait alıntılanan tüm verileri (tablo, grafik, şekil vb. de dahil olmak üzere) atıflarla belirttiğimi;
- 4- Başka yazarların kendi kelimeleriyle alıntıladığım metinlerini, tırnak içerisinde veya farklı dizerek verdiğim yine başka yazarlara ait olup fakat kendi sözcüklerimle ifade ettiğim hususları da istisnasız olarak kaynak göstererek belirttiğimi,

beyan ve bu etik ilkeleri ihlal etmiş olmam halinde bütün sonuçlarına katlanacağımı kabul ederim.

15.01/2018

Yusuf TAŞ

TEŐEKKÜR METNİ

Bu tez alıőmasını hazırlamadan önce ders aőamasında bilgilerinden yararlandıđım bölüm hocalarıma, tezin her aőamasında ilgi ve desteđini esirgemeyen, özellikle analiz ve yorumlama kısmında tecrübelerinden yararlandıđım, yönlendirme ve bilgilendirmeleriyle alıőmamı bilimsel temeller ışıđında őekillendiren sayın hocam Do. Dr. Mehmet Ali ALAN' a teőekkürlerimi sunarım.

Sevgili annem, babam, kardeőime, Do. Dr. Selahattin KO hocama, Arő. Gör. Yasin Görmez, Aziz BAĐCI ve ailesi, Kasım Can IŐIK ve Deniz KASAP' a manevi hibir yardımı esirgemedен yanımda oldukları için tüm kalbimle teőekkür ederim.

İÇİNDEKİLER

İÇİNDEKİLER	i
ŞEKİL LİSTESİ	v
ÖZET	vii
ABSTRACT	ix
GİRİŞ	1
BİRİNCİ BÖLÜM	3
1. VERİ MADENCİLİĞİ TANIMI VE TARİHÇESİ	3
1.1. Veri Tabanlarında Bilgi Keşfi Süreci.....	5
1.1.1. Veri Tabanlarında Bilgi Keşfi Aşamaları	7
1.2. Veri Kaynakları	9
1.2.1. İlişkisel Veri Tabanları	11
1.2.2. İşlemsel Veri Tabanları.....	11
1.2.3. Gelişmiş Veri ve Bilgi Sistemleri	12
1.2.4. Veri Ambarları.....	12
1.2.5. İşlemsel Veri Tabanları ile Veri Ambarları Arasındaki Farklar	15
1.2.6. Veri Ambarı ile Veri Tabanı Arasındaki Farklar.....	16
1.3. Veri Madenciliği, Makine Öğrenmesi ve İstatistik	17
1.4. Veri Madenciliğinin Kullanım Alanları	17
1.5. Veri Madenciliği Sistemlerinin Sınıflandırılması	20
1.5.1. Veri Tabanına Göre	20
1.5.2. Bilgiye Göre.....	20
1.5.3. Yararlanılan Tekniğe Göre	20
1.6. Veri Madenciliğinde Karşılaşılan Zorluklar	21

1.6.1. Veri Tabanı Boyutu	21
1.6.2. Gürültü	21
1.6.3. Eksik ve Artık Veriler	22
1.6.4. Dinamik Veri Yapısı	22
1.7. Modellerine Göre Veri Madenciliği	22
1.7.1. Tanımlayıcı Modeller	23
1.7.2. Tahmin Edici Modeller	23
1.8. Veri Madenciliği Teknikleri	23
1.8.1. Sınıflandırma ve Regresyon Sorgusu	24
1.8.1.1. Yapay Sinir Ağları	24
1.8.1.2. Genetik Algoritmalar	25
1.8.1.3. Bulanık Veri Madenciliği	25
1.8.1.4. Karar Ağaçları	26
1.8.2. Kümeleme	26
1.8.3. Ardışık Zamanlı Örüntüler	28
1.8.4. Birliktelik Kuralları Madenciliği	29
1.8.5. Anormallik Tespiti	30
1.8.6. Görselleştirme	31
1.8.7. Veri Madenciliğinde İstatistiksel Teknikler	31
İKİNCİ BÖLÜM	33
2. BİRLİKTELİK KURALLARI MADENCİLİĞİ	33
2.1. Market Sepet Analizi	34
2.2. Birliktelik Kuralları	35
2.2.1. Güven (Confidence) ve Destek (Support)	39
2.3. Birliktelik Analizinde Kullanılan Algoritmalar	40

2.3.1.AIS Algoritması.....	40
2.3.2.SETM Algoritması	41
2.3.3.Apriori Algoritması	42
2.3.4.Apriori-TID Algoritması	45
2.3.5.FP-Growth (Frequent Pattern Growth) Algoritması.....	46
2.3.6.Tertius Algoritması.....	50
2.3.7.Diğer Algoritmalar.....	50
ÜÇÜNCÜ BÖLÜM	55
3. UYGULAMA.....	55
3.1. Uygulamanın Amacı	55
3.2. Verilerin Hazırlanması	57
3.2.1.Verilerin Kapsamı.....	57
3.2.2.Verit Temizleme	57
3.2.3.Verit Seçme ve Dönüştürme	57
3.3. Kullanılan Algoritma ve Programlar.....	57
3.3.1.Weka Programı	57
3.4. Ürün Grupları Özellikleri Analizi	60
SONUÇ.....	71
KAYNAKÇA	73
EKLER.....	81
Ek: 1 FP-Growth sonuçları.....	81
ÖZ GEÇMİŞ.....	95

ŞEKİL LİSTESİ

Şekil 1.1. Veri Tabanlarında Bilgi Keşfi Sürecini Oluşturan Adımlar	6
Şekil 1.2. Veri Ambarının Temel Yapısı	14
Şekil 2.1. Apriori Algoritması	45
Şekil 2.2. FP-Growth Algoritmasının Genel Yapısı	48
Şekil 2.3. Growth Algoritmasının Genel Yapısı.....	49
Şekil 3.1. Weka Programı Ara yüzü.....	58
Şekil 3.2. Birliktelik Analizi Paneli	59
Şekil 3.3. Değişkenler ve Değerlerinin Gösterimi	60
Şekil 3.4. FP-Growth Algoritması Birliktelik Analiz Sonucu.....	61
Şekil 3.4. FP-Growth Algoritması Birliktelik Analiz Sonucu (Devamı).....	62
Şekil 3.4. FP-Growth Algoritması Birliktelik Analiz Sonucu (Devamı).....	63

ÖZET

Bilişim teknolojilerinin hızlı gelişimine bağlı olarak her sektör ve sistemde hızla artan veri yığınları, veri madenciliği alanındaki çalışmalara çeşitlilik ve önem kazandırmaktadır.

Veri madenciliği çalışmaları, perakendecilik sektöründe müşterilerin ihtiyaçlarının tespiti ve onlara uygun çözümler bulma, sunulan hizmetlerin geliştirilmesi bunlara dayalı olarak yapılacak yatırımların ve stratejilerin tespitinde, günümüzde karar verme sürecinde yer alanlara oldukça yardımcı olmaktadır.

Bu çalışmada Veri Madenciliği teknikleri ile perakende sektöründe hizmet veren bir firmaya ait 10000 satırlık verileri üzerinde, uygulama bölümünde WEKA programı aracılığıyla, FP-Growth algoritması kullanılarak veri madenciliği birliktelik analizi yapılmıştır. Yapılan bu çalışma sonucunda birlikte satılma eğilimi gösteren ürünler hakkında bilgiler verilmiştir.

Üç bölümden oluşan tez, Veri Madenciliği hakkında genel bilgiler, analizde kullanılan “Birliktelik Analizi” hakkında detaylı teorik bilgiler ve son olarak da uygulama bölümlerini içermektedir.

Anahtar Kelimeler: Veri Madenciliği, Apriori, FP-Growth, Birliktelik Kuralları, Veri tabanlarında Bilgi Keşfi Süreci.

ABSTRACT

Rapidly growing data stacks in every sector and systems give importance and diversity to studies on data mining depending on rapid development of information technologies.

Data mining studies are quite helpful for decision makers take part in retail industry in determining of customer needs and finding proper solutions for them and developing services offered and building strategies and investment policies according to them.

In this study, data mining association analysis were conducted using FR-Growth algorithm through WEKA software on 10,000 rows of data which belong to business firm serve in retail industry. In consequence of this study, information was given about products that tend to be sold together.

The thesis which consist of three chapters contains general information about data mining, elaborative theoretical information about “Association Analysis” used in analysis and sections of applications conducted in this research.

Keywords: Data Mining, Apriori, FP-Growth, Association Rules, Knowledge Discovery Process in Databases.

GİRİŞ

Günümüz insanının yapmış olduğu her alışverişte, her telefon aramasında ve her bankacılık işleminde, uydulardan ve uzaktan algılayıcılardan elde edilerek kaydedilen, devlet ve işletme yönetimi altında gerçekleştirilen işlemler neticesinde depolanan veriler her geçen an olağanüstü boyutlarda artış göstermektedir (Akpınar 2000).

Son dönemlerde bilişim sistemleri ve teknolojinin hızlı gelişmesine bağlı olarak; büyük marketler, şirketler ve diğer kuruluşlar veri tabanlarında kuruluşun yapısına ve amacına endeksli olarak farklı kategorilerde veri toplamaktadırlar. Şirketlerin depolamış olduğu veriyi faydalı bilgiye çevirme isteği ve uygun yazılımların gelişimi, depolanan bu veriyi işleyerek, verinin içerisinde saklı olan kullanılabilir ilginç ilişkilerin, birlikteliklerin ve örüntülerin açığa çıkarılmasını kaçınılmaz hale getirmiştir. Fakat müşteri özellikleri ve müşterilerin satın alma alışkanlıklarına yönelik yararlı ve kullanışlı bilgiler çoğunlukla saklı ve henüz işlenmemiş durumdadır.

Saklı ve ham bilgiye karşı olan bu inanılmaz ilgi, veri madenciliği ve veri tabanlarında bilgi keşfi gibi yeni alanların bulunmasıyla daha anlaşılabilir ve kolay yorumlanabilir bir duruma gelmiştir. Özellikle bilişim teknolojilerinde dikkati çekecek ölçüde yaşanan gelişmeler ile depolanan verilerin boyutlarının artması, veri madenciliği ve veri tabanlarında bilgi keşfi' nin gelişmesine sebep olmuştur. Bu yeni alanlar ve bilgisayarın kullanımı ile binlerce veri yığını arasından kullanışlı bilgi otomatik olarak keşfedilmektedir.

Veri madenciliği çalışmalarında ilgi çekici sonuçları elde etmenin en önemli alanlarından biri birliktelik kuralları madenciliğidir. Elde edilen kurallar ile nesne veya nesne grupları arasındaki ilişkiler tespit edilebilmekte, alınacak kararlar ve yapılacak olan tanımlamalarda yol gösterici olarak kullanılabilir. Birliktelik kurallarının belirlenmesinde birçok farklı algoritma kullanılmaktadır. Bunların en başında Apriori algoritması gelmektedir. Apriori algoritması, performansı ile daha önce tanımlanan algoritmaları geride bırakmış, daha sonra gelen paralel algoritmaların da temelini oluşturmuştur. Veri kümelerinin durumuna göre Apriori

temelinde AprioriTID ve AprioriHybrid algoritmaları da tanımlanmıştır. Performans boyutunda kullandığı bellek tabanlı yöntemlerle çok büyük veri kümelerinde göz dolduran FP-Growth algoritması apriori algoritmasını geride bırakmıştır.

Veri madenciliğinde birliktelik kuralları madenciliği çalışmaları çeşitli alanlarda yer bulmuştur. Bu çalışmaların en başında market sepet analizi çalışmaları gelmektedir. Perakende sektöründe müşteri satın alma alışkanlıklarının belirlenmesi ve bu doğrultuda satış ve pazarlama stratejilerinin belirlenmesi konusunda etkili sonuçlara ulaşılmıştır. Market sepet analizi çalışmalarında kullanılan veriler market- mağaza çerçevesinde yoğunlaşmaktadır. Ancak, farklı alanlarda da uygulamalar yararlı sonuçlara erişim sağlayacaktır.

Bu çalışmada birinci bölümde veri madenciliği alanı tanıtılmış, veri madenciliğinin ilk adımı olan bilginin keşfi süreci hakkında bilgiler verilmiştir. Ayrıca veri madenciliğinde kullanılan veri kaynakları, veri madenciliğinin kullanım alanları, veri madenciliğinin benzer hedefler için çalışan bilim dallarından farklılıklarının yanı sıra veri madenciliğinin ilgilendiği problemlerden de kısaca bahsedilmiştir.

İkinci bölümde çalışmanın temelini oluşturan birliktelik kuralları madenciliği hakkında teorik bilgi verilmiştir. Analizde kullanılan ölçüm değerlerinin yanı sıra kullanılan algoritmalar hakkında bilgi sunulmuştur.

Üçüncü ve son bölümde market satış verileri üzerinde birliktelik kuralları madenciliği yapılmıştır. Uygulamada verilerin elde edilmesi ile kuralların belirlenmesi arasında izlenen adımlar hakkında bilgiler verilmiştir. Bu adımlar sırasında kullanılan teknolojiler, birliktelik kurallarının belirlenmesi ve bu kuralların yardımcı veriler ile yorumlanması bu bölümde ele alınan konu başlıkları arasındadır.

BİRİNCİ BÖLÜM

1. VERİ MADENCİLİĞİ TANIMI VE TARİHÇESİ

İlk bilgisayarlar 1950’li yıllarda sayımlar için kullanılmaya başlanmıştır. 1960’lı yıllara gelindiğinde veri tabanı ve verilerin tutulması kavramı teknoloji dünyasındaki yerini almıştır. Bilim adamları 1960’ların sonunda basit öğrenmeli bilgisayarlar geliştirebilmişlerdir. Günümüzde sinir ağları olarak bilinmekte olan perseptron’ların yalnızca çok basit olan kuralları öğrenebileceğini Minsky ve Papert göstermişlerdir (Adriaans, Zantinge 1997).

1960’lardan beri veri tabanı ve bilgi teknolojisi, ilkel dosya işleme sistemlerinden daha karmaşık ve güçlü veri tabanı sistemlerine doğru sistematik bir şekilde gelişim göstermeye başlamıştır. 1970’lerde veri tabanı sistemleri ile ilgili araştırma ve geliştirmeler hiyerarşik yapılardan ilişkisel veri tabanı yapılarına doğru ilerlemeye başlamıştır. İlişkisel veri tabanı sistemlerinin getirdiği veri modelleme araçları, indeksleme ve veri organizasyon yöntemleri ile tablolar birbirleri ile ilişkili olarak saklanmaya başlamıştır. Bunlara ek olarak kullanıcılar, sorgulama dilleri ve kullanıcı ara yüzlerini kullanarak optimize sorgu işleme ve işlem yönetimi aracılığıyla veriye daha esnek ve daha rahat erişme imkânı elde etmişlerdir.

1980’lerde yaygınlaşmış olan veri tabanı yönetim sistemleri, mühendisliklerde ve bilimsel alanlarda uygulanmaya başlanmıştır. Bu yıllarda işletmeler, müşterileri, ürünleri ve rakipleri hakkındaki verilerin bulunmuş olduğu veri tabanları hazırlamışlardır. Hazırlanan bu veri tabanlarında çok büyük boyutlarda veri bulunmaktadır ve bu verilere SQL (Structured Query Language) veri tabanı sorgulama dili ya da benzeri diller kullanarak ulaşılabilmektedir.

1990’lı yıllarda içerisindeki veri boyutu hızla katlanarak büyüyen veri tabanlarından, yararlı bilgilerin ne şekilde bulunabileceği düşünülmeye başlanmış ve bunun üzerine yayınlara ve çalışmalara başlanmıştır. 1989, KDD (IJCAI)-89 Veri Tabanlarında Bilgi Keşfi Çalışma Grubu toplantısı ve 1991, KDD (IJCAI)-89’un sonuç bildirgesi sayılabilecek “Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop” makalesinin KDD (Knowledge Discovery and Data

Mining) ile ilgili temel tanım ve kavramları ortaya koyması ile süreç daha da hızlanmış ve sonunda veri madenciliği için 1992 yılında ilk yazılım gerçekleştirilmiştir (Savaş, Topaloğlu ve Yılmaz 2012).

Veri madenciliği 2000'li yıllarda sürekli olarak gelişme göstermiş ve neredeyse tüm alanlarda uygulanmaya başlanmıştır. Bu alana olan ilgi alınan sonuçların yararları görüldükçe artmıştır. Veri madenciliği, verilerdeki kalıpları keşfetme süreci olarak tanımlanır. Süreç otomatik veya yarı otomatik olmalıdır. Keşfedilen modeller, genellikle ekonomik bir avantaj sağlayan bazı avantajlara neden oldukları için anlamlı olmalıdır (Ian, Eibe 2005: 4).

Veri Madenciliği, büyük miktardaki veri yığınlarından önemli bilgilerin ortaya çıkmasını sağlar. Böylece normal şartlarda fazla zaman alan çalışmalarla kesin doğru olmayacak biçimde tespit edilebilen bilgi, veri madenciliği sayesinde kısa sürede ve doğru olarak tespit edilir. Tespit edilmiş olan bu bilgiler tarafsız değerlendirmelerin yapılmasında ya da önemli kararların alınmasında kullanılır ve iş dünyasındaki yaklaşımlara ilişkin tahminlerde bulunulmasına ve kurumsal veri kaynaklarının iyi analiz edilmesine yardımcı olur. Özetle veri madenciliği ile işletmeler stratejik kararlar alırken kendilerine devasa veri yığınları içerisinde yol gösterecek hayati verileri çıkararak analiz edebilir (Alpaydın 2000).

Veri madenciliği temel olarak, veri setleri arasındaki düzenin ya da desenlerin tespit edilmesi, yazılım ve verinin analizi tekniklerinin kullanılması ile alakalıdır. Bilgisayar verileri arasındaki bağın, özelliklerin ve kuralların tespit edilmesinden sorumludur. Hedef, önceden fark edilmemiş olan veri desenlerini belirleyebilmektir (Arslan 2008: 4).

Veri Madenciliği, beklenmeyen veya tahmin edilemeyen ilişkileri tespit etmek, anlaşılabilir ve kullanışlı bilgileri elde etmek için büyük boyutlardaki verilerin çeşitli yöntemlerle analiz edilmesi ve özetlenmesidir.

Veri madenciliği ve veri tabanlarında bilgi keşfi (Knowledge Discovery in Databases - KDD) süreci kavramları kaynaklar incelendiğinde zaman zaman birbirinin yerine kullanıldığı görülmektedir. Veri tabanlarında bilgi keşfi aşamasında kritik bir adım olan veri madenciliği, birçok çalışmada sürecin tamamını anlatmak için kullanıldığı görülmektedir. Veri madenciliği çalışması ile büyük veri tabanı

sistemleri içerisinde saklı kalmış olan bilgilerin ortaya çıkarılması sağlanır. Bu işlem istatistik ve matematik disiplinleri, yapay zekâ uygulamaları, modelleme teknikleri, çeşitli bilgisayar uygulamaları ve veri tabanı teknolojisi kullanılarak yapılır (Arslan 2008: 3).

1.1. Veri Tabanlarında Bilgi Keşfi Süreci

Geleneksel olarak, veri toplama işlemi, veri analizinin en önemli aşamalardan biri olarak kabul edilmektedir. Bir analist (örneğin, bir istatistikçi veya veri bilimcisi), toplanacak değişkenleri seçmek için mevcut alan bilgisini kullanır. Seçilen değişkenlerin sayısı genellikle kısıtlı olur ve değerlerinin toplanması el ile yapılabilirdi. Bilgisayar destekli analiz durumunda, analist toplanan verileri istatistiksel bir bilgisayar paketi veya elektronik tabloya girmek zorunda kalırdı. Yüksek veri toplama maliyetlerinden ötürü insanlar, kısıtlı bilgi temelinde kararlar vermeyi öğrendiler.

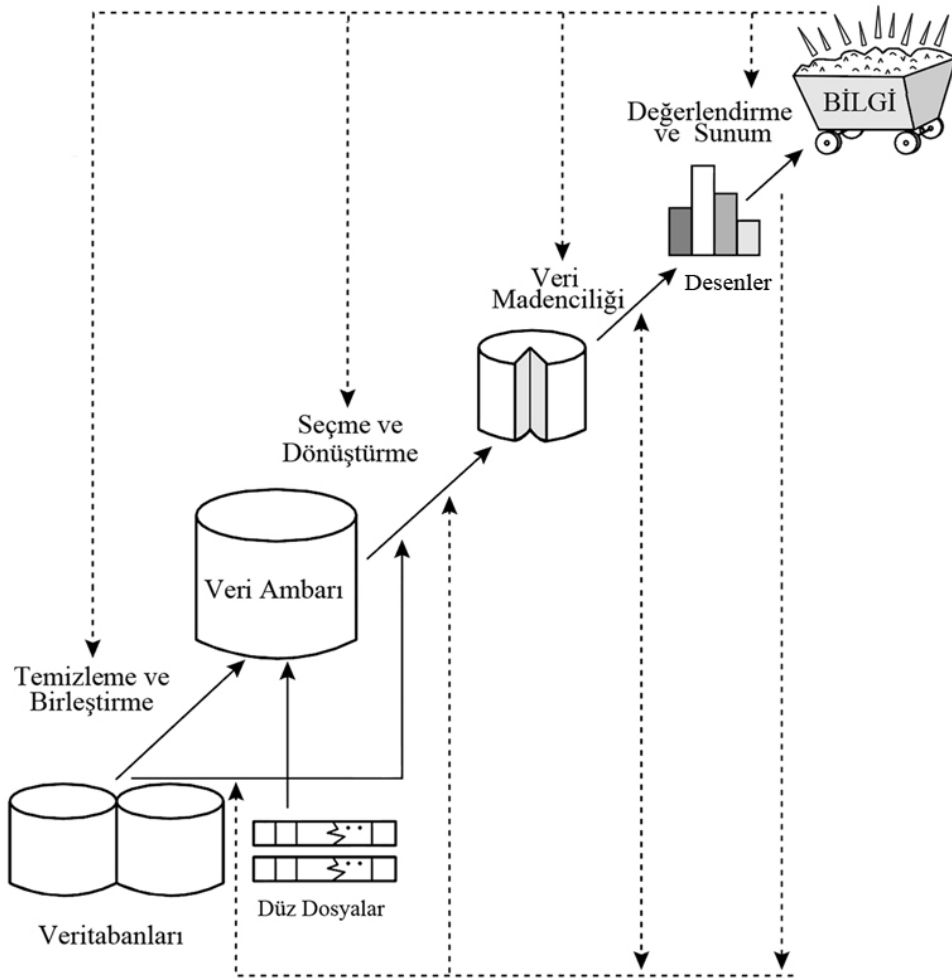
Kayıtlı verilerden anlamlı bilgilere ulaşım sürecine Veri tabanlarında Bilgi Keşfi (VTBK) olarak nitelendirilmektedir. Veri tabanlarında bilgi keşfi, depolanmış veri içerisindeki yeni, geçerli, faydalı ve sonuç olarak anlaşılabilir örüntülerin elde edilmesi sürecidir. Bu sürecin ilk adımı, uygulama alanının öğrenilmesi ile başlar. Veri tabanlarında bilgi keşfinin son adımı ise, elde edilmiş olan bilginin görüntüleme ve bilgi gösterimi yöntemleri kullanılarak kullanıcıya sunulması şeklindedir. Bazı araştırmacılar veri tabanlarında bilgi keşfi ile veri madenciliğini eş anlamlı olarak kabul etmelerine rağmen, genel görüş veri madenciliği veri tabanlarında bilgi keşfi sürecinin bir aşaması şeklindedir (Şen 2008: 3).

Bilgi Çağının başlangıcından bu yana, veri toplama ve saklama süreçleri daha kolay ve daha ucuz hale gelmiştir. Her 20 ayda bir saklanan bilginin iki katına çıktığı öngörülmektedir (Rokach, Maimon 2008: 3).

Winter Corporation tarafından yapılan, dünyadaki en büyük işletme veri tabanlarının tespit edilmesi hedefli bir araştırmada, 1998 yılında Sears, Roebuck and Co.'nun yalnızca karar destek amaçlı kullanılan veri tabanının 4630 gigabayta eriştiği görülmektedir. Veri tabanı sistemlerinin yaygınlaşan kullanımı ve boyutlarındaki bu devasa artış işletmeleri, saklanan bu verilerden hangi şekilde

yararlanabileceği sorunu ile karşı karşıya bırakmıştır. Raporlama araçları veya geleneksel sorgu (query) dili'nin oluşan veri yığınları karşısında yetersiz kalması, Veri Tabanlarında Bilgi Keşfi gibi yeni arayışların bulunmasına neden olmaktadır. Veri Tabanlarında Bilgi Keşfi süreci içerisinde, modelin kurulması ve değerlendirilmesi aşamalarından meydana gelen Veri Madenciliği, sürecin en önemli bölümünü oluşturmaktadır (Akpınar 2000).

Veri tabanı keşiflerini izleyen adımların aşağıdaki şekilde sunulmuştur:



Şekil 1.1. Veri Tabanlarında Bilgi Keşfi Sürecini Oluşturan Adımlar

İnsanların birçoğu veri tabanlarında bilgi keşfi terimini, veri madenciliği ile eş anlamlı olarak kullanmaktadır.

1.1.1. Veri Tabanlarında Bilgi Keşfi Aşamaları

Veriden bilgiye ulaşım sürecindeki Veri Tabanlarında Bilgi Keşfi aşamaları şu şekildedir:

- **Veri Seçimi (Data Selection):** Bu aşama, birden çok veri kümesi arasından, üzerinde sorgu yapılmasına uygun örnek bir veri kümesi oluşturma aşamasıdır. Veri toplama (data collection) ve farklı kümelerdeki verilerin birleştirilmesi işlemi de bu süreçte yer alır. Toplama, belirlenen problem için gerekli olduğuna inanılan verilerin ve bu verilerin toplanacağı veri kaynaklarının tespit edilmesi adımıdır.

Veri seçimi aşamasında yapılması gerekenler;

- Farklı ortamlardaki verilerin mevcut yapılarının incelenmesi ve tablo yapılarının ortaya çıkarılması,
- Veri madenciliği ile hedeflenen sonuca ulaşmak için gerekli verilerin, uygulama için belirlenen veri depolama ortamına aktarılması olarak sıralanabilir.

- **Veri Önleme (Data Preprocessing):** Veri seçimi işlemi ile elde edilen örnek veri kümesinde bulunan hata ve eksikliklerin düzenlendiği ve ayıklandığı aşamadır. Veri temizleme (data cleaning) ve veri dönüştürme (data transformation) veri önleme işlemleridir. Veri temizlemenin amacı gürültülü ve ilgisiz verinin veri setinden çıkarılmasıdır. Veri dönüşümünün amacı ise, kaynak veri içindeki farklı biçimdeki veri tip ve değerlerini yapılacak veri madenciliği çalışması doğrultusunda değiştirmektir.

Modelde kullanılan veri tabanının çok büyük olması halinde örnekleme yapılması daha uygun olabilir. Hesaplama imkânı, günümüzde ne kadar gelişmiş olursa olsun, çok büyük veri tabanları üzerinde çok fazla sayıda modelin denenmesi fazla zaman alması sebebiyle mümkün olamamaktadır. Bu yüzden veri tabanının tamamını kullanarak bir kaç modelin denenmesi yerine, rastgele örneklenmiş bir veri tabanı bölümü üzerinde birçok modelin denenmesi ve bunlar içerisinde en güvenilir ve en güçlü modelin kullanılması daha uygun olacaktır.

Veri tipi dönüşümü, basit olarak veri tipi değişimidir. Örnek olarak, integer tipteki bir veriyi boolean tipine dönüştürme işlemi verilebilir. Bu dönüşümün sonucunda, sorgulama yapılacak veri tabanı boyutu azaltılabilir ve sorgularda hız artışı sağlanabilir.

Bazı veri tabanlarında bir kolon içinde sürekli tekrarlayan benzer veriler bulunmaktadır. Bu verileri bir kaç grup içine yerleştirme işlemi uygulanarak verinin kalitesi artırılır. Gruplama tekniği ile yorumlamanın daha kolay olması sağlanabilir.

Farklı veri tabanlarından gelen veriler tek bir tablo içinde birleştirildiğinde veri alanlarının bazıları boş kalabilir. Bu durumu düzeltmek için, kayıp değerler en çok kullanılan değerler ile doldurabilir, bir kayıta çok fazla kayıp değer varsa kayıt tamamen silinebilir, en olası ortalama değer ile doldurulabilir.

- **Veri İndirgeme (Data Reduction):** Seçilmiş olan örnek veri kümesindeki gereksiz ve tekrarlı olan verilerin ayıklandığı aşamadır. Bu işlem ile verinin boyutu indirgiğinden veri madenciliği uygulanırken çalıştırılacak sorguların daha hızlı sonuç üretmeleri sağlanır.

- **Veri Madenciliği (Data Mining):** Bu aşama veri madenciliği yöntemlerinin ve algoritmalarının uygulandığı adımdır. Veri Madenciliği; veri tabanı sistemleri, verilerin saklanması, makine öğrenimi, istatistik gibi alanların kombinasyonundan oluşan disiplinler arası bir yöntemdir. Veri madenciliği istatistikçiler için yeni bir konu değildir. İstatistik ve Veri madenciliği ortak amaçlara sahiptir, her ikisi de verilerin yapılarının keşfedilmesiyle ilgilidir. İstatistik ve veri madenciliği kavramsal olarak farklı alanlar olsa da ikisinin ortak özellikleri vardır. Her iki alan da veriden öğrenme, verinin bilgiye dönüştürülmesi, veriyi analiz etme, verinin anlamını çözme, belirsizlikleri ortadan kaldırma, olayı etkileyen faktörleri belirleme, ön görüde bulunma amaçlarını yerine getirirler. Genel amaçlarına bakıldığında her iki alanın da odaklandığı nokta veriden öğrenme ya da veriyi bilgiye dönüştürmedir. Bu açıdan, her ikisinin de edindiği görev birbirine çok yakındır. Veri Madenciliği, klasik istatistiksel uygulamalarla çok fazla benzerlik göstermektedir, fakat klasik istatistiksel uygulamalar yeterli derecede düzenlenmiş ve genellikle özet veriler üzerinde çalıştırılır ve analiz edilen veri sayısı binler, yüz binler düzeyinde iken Veri Madenciliğinde bu sayı milyon hatta milyarlar seviyesinde olmaktadır.

Dolayısıyla deęişken sayısı da çok fazla olduğundan klasik istatistiksel yöntemler, bu verileri analiz etmeye yeterli gelmemeye başlamış ve yeni tekniklere ihtiyaç duyulmaya başlanmıştır.

Veri Madencilięi ile istatistik uygulamaları arasındaki en önemli iki farktan birincisi; Veri Madencilięi terabaytlarla ölçülen milyarlarca veri üzerinden hesaplamalar yaparak ortaya çıkmamış, gizli kalmış bilgileri, desenleri açığa çıkarmaya çalışırken, istatistik ise çok büyük veriler üzerinden analiz yapmaz. İkinci fark ise; Veri Madencilięi, geçmiş verileri kullanarak geleceęe yönelik hiç akla dahi gelmeyecek örüntüleri, gizli kalmış bilgileri keşfetmeye yönelikken, istatistikte ise veriler akıldaki bazı sorular için toplanır, anketler yapılır ve bu sorulara yanıt bulmak için veriler analiz edilir (Demirel 2010: 40).

- Deęerlendirme (Evaluation): Bilgi keşfi sürecinde bu aşamadan önceki aşamalar neticesinde elde edilen bilginin yenilik, geçerlilik, yararlılık ve basitlik kriterlerine göre deęerlendirilmesi aşamasıdır (Fayyad, vd. 1996: 28).

Veri madencilięi süreci, bilgi tabanı ve kullanıcıyla etkileşim içindedir. Kullanıcıya elde edilen ilginç örüntüler sunulur ve eęer istenirse bilgi tabanına da kaydedilebilir. Yukarıdaki adımlar içerisinde veri madencilięi adımı olarak tanımlanan 4. adımın nerede başlayıp nerede bittięi bulanıktır. Çoğunluk itibari ile önceki ve sonraki adımları kapsamaktadır. Veri madencilięi noktasında çalışan firmaların veri madencilięi adına bu adımların tümünü yaptığını söylenebilir.

1.2. Veri Kaynakları

Veri kaynakları, verinin saklandığı alanları ifade eder. Veri tabanları, Excel dosyaları (her türlü veriyi, özellikle sayısal verileri tablolar ya da listeler halinde tutma ve bu verilerle ilgili ihtiyaç duyacağınız tüm hesaplamaları ve analizleri yapma imkânı sunan bir uygulama programı), XML dosyaları (interneti kullanarak veri alışverişi yapan sistemler ve platformlar arasındaki veri iletişimini standart hale getirmek için tasarlanan bir işaretleme dili), metin dosyaları (hiçbir formatlama kontrolü içermeyen ve sadece metin karakterlerinden oluşan bilgisayar dosyaları) vb. dosyalar veri kaynakları olarak nitelendirilirler. Veri kaynakları, bilhassa de açık kaynaklı bilgisayar programlarının artmasıyla çok fazla çeşitlilięe sahip olmuştur.

Ücretsiz veya ticari veri analiz programlarının hepsi standart olarak kabul edilen veri tabanı, xml ve text veri kaynaklarına bağlanabilmektedir.

Veri tabanı (Database), birbirleriyle ilişkisi olan verilerin tutulduğu, kullanım amacına uygun olarak düzenlenmiş veriler topluluğunun mantıksal ve fiziksel olarak anlamlarının olduğu ve bunların sayısal ortamlarda saklandığı ve gerektiğinde tekrar bir erişime olanak sağlayan, büyük boyutlarda veriler barındıran bilgi depolarıdır. Veri tabanları gerçekte var olan ve birbirleriyle ilişkileri olan nesnelere ve ilişkilerini modeller. Veri tabanı; üniversite, banka, seyahat şirketi, okul, hastane vb. kuruluşların çalışıp işleyebilmesi için gereken uygulama programlarının kullandığı operasyonel çok çeşitli verilerin toplamıdır. Ticari bir şirket için satış bilgileri, müşteri bilgileri, ödeme bilgileri, ürün bilgileri vb., okul için açılan dersler, öğrenci bilgileri, öğretmen bilgileri, okula kaydedilmiş öğrenciler, sınav sonuçları, sınav tarihleri vb., hastane için doktor bilgileri, hasta bilgileri, mali bilgiler, teşhis-tedavi bilgileri vb. kullanılan çok çeşitli operasyonel verilere örnek olarak verilebilir.

Belirli bir konu hakkında toplanmış veriler bir veri tabanı programı altında toplanırlar. Bu verilerden istenildiğinde; toplanan bilgilerin tümü veya istenilen özelliklere uyanları görüntülenebilir, yazdırılabilir ve hatta bu bilgilerden yeni bilgiler üretilerek bunlar çeşitli amaçla kullanılabilir.

Özelliklerine göre veri yığınlarının sınıflandırılması yapısal, yarı yapısal ve yapısal olmayan şeklindedir. İstatistiklere göre bütün verilerin %80'i yarı yapısal verilerdir. Bu veriler, içerisinde metin, grafik, resim vs. bulunduran dokümanlardır (Güven, Bozkurt ve Kalıpsız 2007).

Veri madenciliği uygulamalarının yapılabileceği farklı veri havuzları vardır. Esas itibarıyla, veri madenciliği her türden veri havuzuna ve veri akışları gibi geçici veriler de dâhil herhangi bir veri kaynağı üzerine uygulanabilir. Dolayısıyla, veri depolarının incelenmesinin kapsamı, işlemsel veri tabanları, ilişkisel veri tabanları, gelişmiş veri tabanı sistemleri, veri ambarları ve World Wide Web'i içermektedir (Han, Kamber 2006: 3).

1.2.1. İlişkisel Veri Tabanları

İlişkisel veri, grafik kullanıcı ara yüzleri ya da SQL gibi yapısal sorgu dilleri vasıtasıyla yazılan sorgular ile erişilebilir veridir. Kullanıcı ara yüzü ya da kullanılan sorgu dilinin sağladığı olanaklar doğrultusunda, veriler ile iletilen soruların yanıtları alınmaktadır (Dolgun 2006: 9).

İlişkisel veri tabanlarına bakıldığında veri özelliğine göre her biri farklı isimlerde ayrılmış olan tablolarla karşılaşılmaktadır. Ayrılan bu tablolar özellik (kolonlar, alanlar) kümesinden oluşmakta ve “kayıt” olarak adlandırılan satırlardan meydana gelmektedir. İlişkisel veri tabanları, kendisini oluşturan varlıklar ve bu varlıkların ilişkilerinden meydana gelen veri modelleri ile tanımlanır. İlişkisel veri tabanlarına veri madenciliği uygulandığında veri desenleri ve eğilimler araştırılarak ileri yönde adımlar atılabilir. İlişkisel veri tabanları, tüm özellikleri ile veri madenciliği çalışmalarında en yoğun kullanılan, zengin ve en gözde veri kaynaklarından biridir (Delibaş 2010: 7).

Veri madenciliği, ilişkisel veri tabanlarındaki verilere ait eğilimleri çözümlmek için ya da verideki örüntüleri ortaya çıkarmak için kullanılabilir. İlişkisel veri tabanları en çok kullanılan veri madenciliği uygulama platformlarından birisidir (Han, Kamber 2000: 10).

1.2.2. İşlemsel Veri Tabanları

Genel olarak, bir işlemsel veri tabanı, her kaydın bir işlemi temsil ettiği bir dosyadan oluşur. Her işlem genellikle benzersiz bir işlem kimlik numarası (trans ID) ve işlemi oluşturan öğelerin bir listesi (bir mağazada satın alınan öğeler gibi) içerir (Han, Kamber 2006: 15).

İşlemsel veri tabanı, satışın yapıldığı şube veya kasa, satışı gerçekleştiren personel bilgileri, müşteri bilgileri vb. bilgileri, işlemle ilgili verilerin depolandığı başka tabloları da içerebilmektedir. Bu tür veri tabanlarına bir analist tarafından “X müşterisinin satın aldığı tüm ürünlerin listesi” veya “Y2 kodlu ürüne ait kaç adet satış işlemi gerçekleştirilmiştir?” gibi sorular yöneltilebilir. Bu gibi sonuçları elde etmek için işlemsel veri tabanlarında tarama yapacak basit sorgular yazarak sonuçlar elde edilebilecektir. Daha derin inceleme yapılmak istendiğinde “Hangi ürünler

birlikte daha çok satılmıştır?” sorusu sorulabilir. Bu gibi işlemlerde de sıradan veri sorgulama işlemleri ile değil de sık görülen ürün birliktelikleri analizi gibi veri madenciliği sistemleri kullanılmaktadır (Delibaş 2008: 10).

1.2.3. Gelişmiş Veri ve Bilgi Sistemleri

İlişkisel veri tabanı sistemleri iş yaşamındaki uygulamalarda yaygın olarak kullanılmaktadır. Veri tabanı teknolojisindeki gelişmelerle birlikte, çeşitli gelişmiş veri ve bilgi sistemleri ortaya çıkmıştır.

Yeni veri tabanı uygulamaları, mekânsal verilerin (haritalar gibi) işlenmesi, mühendislik tasarım verileri(binaların tasarımı, sistem bileşenleri veya entegre devreler gibi), hipermetin ve çoklu ortam verilerini (metin, resim, video ve ses verileri dâhil), zamanla ilgili veriler (tarihi kayıtlar veya borsa verileri gibi), sürekli veri akışı içeren verileri (video kayıt verileri, sensör verileri gibi sürekli veri akışının söz konusu olduğu veriler) ve internetten yapılan her türlü erişimin kayıt altında tutulduğu büyük bilgi depoları haline gelmiş web verilerini kapsamaktadır (Tiryaki 2006: 37).

Bu uygulamalar, karmaşık nesne yapılarını işlemek için verimli veri yapıları ve ölçeklenebilir yöntemler gerektirir; değişken uzunluktaki kayıtlar; yarı yapılandırılmış veya yapılandırılmamış veriler; metin, zaman - mekânsal ve multimedya verileri; karmaşık yapılar ve dinamik değişikliklerle veri tabanı şemaları gibi. Bu ihtiyaçlara cevap olarak veri tabanı sistemleri ve özel uygulama odaklı veri tabanı sistemleri geliştirilmiştir.

Bu tür veri tabanları veya bilgi depoları, büyük miktarda karmaşık veriyi verimli bir şekilde depolamak, almak ve güncellemek için gelişmiş olanaklar gerektirirken, veri madenciliği için verimli bir zemin oluşturur ve birçok zorlu araştırma ve uygulama sorununu ortaya çıkarırlar (Han, Kamber 2006: 16).

1.2.4. Veri Ambarları

Basit anlamda veri madenciliği işleminin uygulanacağı verilerin oluşturulduğu özel veri tabanlarıdır. Veri tabanlarında bulunan veriler ile analiz yapmak ve karar destek sürecinde yararlanmak, veri madenciliği ile mümkün olabilmektedir. Madenciliği yapılacak verinin de bazı niteliklere sahip olması

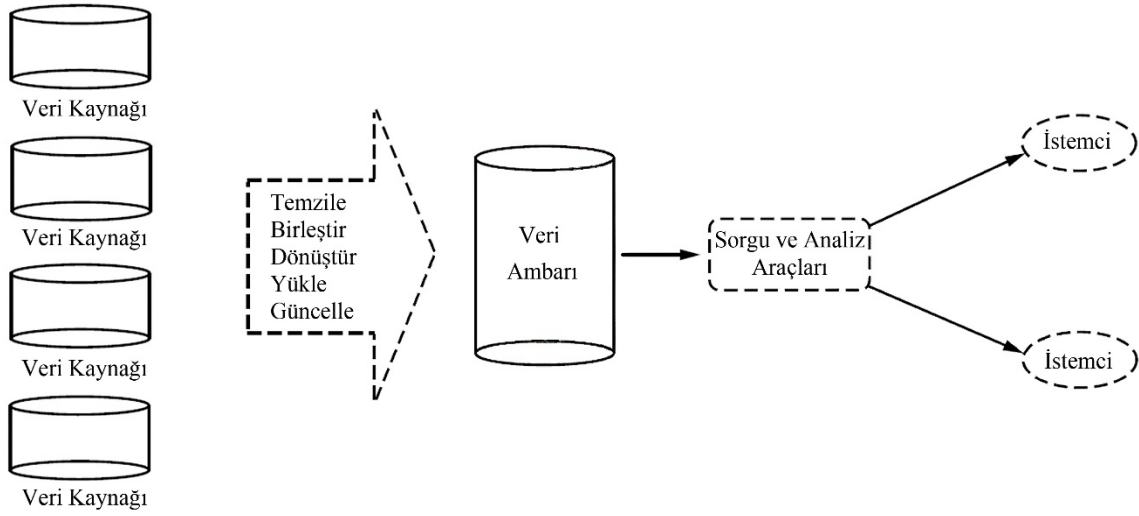
gerekir. Bu nitelikler veri ambarı (Data Warehouse) ile sağlanmaktadır. Veri ambarlarının hazırlanması işlemi, verinin çeşitli kaynaklardan toplanarak, veriler içerisinde bulunan hatalar ve uyumsuzlukların belirlenerek ortadan kaldırılmasıdır.

Veri ambarları, bir kurumun ya da işletmenin farklı bölümleri tarafından canlı sistemler sayesinde toplanan verilerin, daha sonra incelemeye alınabilecek durumda olanlarının arka planda yer alan bir sistemde birleştirilmesiyle oluşturulan büyük ölçekli veri depolarıdır (Tantuğ 2002: 8).

Veri ambarları, coğrafi bilişim sistemlerinden sağlık sektörlerine, şirketlerin pazarlama biriminden üretime, geleceğe yönelik tahminler yapmak, sonuçlar çıkarmak ve şirketlerin yönetim stratejilerini belirlemek için kullanılan bir sistemdir. Masraflı bir yatırım maliyetine sahip olsa bile getirisi ve faydası bu maliyetleri fazlasıyla aşmaktadır (Döşlü 2008: 6).

Veri ambarları, organizasyonel verinin kolay bir biçimde elde edilmesini sağlayan yapılardır. 1990'lı yıllarda ortaya çıkan veri ambarları verileri kullanılabilir ilişki ve profillerde sınıflandırmazlar, yalnızca potansiyel bilgiye sahip olan veri tabanlarıdır. Veride saklı olan bilginin keşfedilmesini sağlayan ise veri madenciliği gibi tekniklerdir. Veri ambarından veriyi çekebilmek için hangi veriye ihtiyaç olduğunu ve bu verinin nerede bulunduğunu belirlemek önemlidir. Genellikle gerekli olan veri, ayrı sistemler üzerinde olup, başka formatlardadır (Gülce 2010: 4).

Bir veri ambarı, birden fazla kaynaktan toplanan, birleştirilmiş bir şema altında saklanan ve genellikle tek bir sitede bulunan bir bilgi deposudur. Veri ambarları, veri temizleme, veri birleştirme, veri dönüştürme, veri yükleme ve periyodik veri yenileme işlemleri ile oluşturulur. Şekil 2'de veri ambarının temel yapısı sunulmuştur (Han, Kamber 2006: 106).



Şekil 1.2. Veri Ambarının Temel Yapısı

Bir veri ambarı genellikle çok boyutlu bir veri tabanı yapısı ile modellenir; burada her boyut, şemadaki bir nitelik veya bir nitelik kümesine karşılık gelir ve her hücre, sayım veya satış miktarı gibi bazı toplam ölçütün değerini depolar. Bir veri ambarının gerçek fiziksel yapısı, ilişkisel bir veri deposu veya çok boyutlu bir veri küpü olabilir. Bir veri küpü çok boyutlu bir veri görünümü ve özetlenen verilerin önceden hesaplanmasına ve hızlı bir şekilde erişilmesini sağlar (Han, Kamber 2006: 107).

Bir veri ambarının temel yapısı; organizasyon içinde bulunan son kullanıcıların tamamına işlem sonuçlarını ve verileri sağlayan, en gelişmiş iletişimi sunan, birbiriyle bütünleşik aşağıdaki alt bileşenlerden oluşur (Ergün 2008: 33):

- Enformasyon Ulaşım katmanı,
- Operasyonel Veri Tabanı / Harici Veri Tabanı Katmanı,
- Uygulama Haberleşmesi Katmanı,
- Veri Ulaşım Katmanı,
- Veri Dizin (Metadata) Katmanı,
- İşlem (Process) Yönetim Katmanı,
- Veri Sunum Katmanı,
- Veri Ambarı Katmanıdır.

1.2.5. İşlemsel Veri Tabanları ile Veri Ambarları Arasındaki Farklar

İşlemsel veri tabanları çevrimiçi olarak verilerin işlenmesi ve bunların sorgulanması ilkesine göre çalışmaktadır. Bu özelliği ile bu tür sistemlere çevrim içi hareket işleme (OLTP - Online Transaction Processing) ismi verilmektedir. Bu sistemlerde işletmelerin satın alma, stok, üretim, bankacılık işlemleri, bordro, tescil ve muhasebe gibi gün-gün yapılan işlem kayıtları tutulur. Diğer tarafta veri ambarları ise, karar verme veya veri analizi rollerini üstlenmiş kullanıcılara veya bilgi çalışanlarına hizmet sunmaktadır. Bu sistemler veriyi farklı kullanıcıların çeşitli ihtiyaçlarını karşılama durumuna göre düzenlemekte veya sunmaktadır. Bu tür sistemler de çevrimiçi analitik işlem süreci (OLAP – Online Analytical Processing) olarak adlandırılır (Han, Kamber 2006: 108).

Online Analytical Processing ve Online Transaction Processing sistemlerin özellikleri arasındaki temel farklılıklar şu şekilde özetlenebilir (Delibaş 2010: 10):

- **Kullanıcı ve Sistem Yönelimi:** Bir Online Transaction Processing sistem müşteriye yöneliktir ve çalışanların, istemcilerin ve bilgi teknolojileri uzmanların sorgulama işlemlerinde kullanılır. Online Analytical Processing ise pazara yöneliktir ve idarecilerle analistleri içeren bilgi çalışanlarının kullanımına uygundur.
- **Veri İçeriği:** Bir Online Transaction Processing sistem karar vermek için kolayca kullanılabilir çok fazla detaya sahip hali hazırda veriyi yönetir. Online Analytical Processing sistem ise farklı detay seviyesindeki bilginin özetlenmesi, birleştirilmesi, depolanması ve kaydedilmesi için kolaylıklar sunan, büyük boyutlardaki geçmiş verileri yönetir. Bu özellikler veriyi karar vermede kolaylıkla kullanılabilir hale getirmektedir.
- **Veri Tabanı Tasarımı:** Online Transaction Processing sistem genellikle uygulama tabanlı veri tabanı tasarımını ve varlık-ilişki veri modelini benimsemektedir. Online Analytical Processing sistem ise genelde yıldız veya kar tanesi modelini ve konu tabanlı veri tabanı tasarımını benimsemektedir.

- **Odak Noktası:** Online Transaction Processing sistem esasen geçmiş verilerini içermeksizin bir kuruluşun veya departmanın mevcut verilerine odaklanır. Tam tersi olarak Online Analytical Processing sistem ise bir kuruluşun gelişim sürecinden oluşan veri tabanı şemasının birçok örneğine sık sık göz atar. Online Analytical Processing sistemler aynı zamanda farklı organizasyonlardan meydana gelen veya birçok veri kaynağından birleştirilmiş bilginin üstesinden gelebilmektedir. Büyük kapasitesinden dolayı Online Analytical Processing verisi, çoklu depolama ortamlarında saklanır.
- **Erişim Deseni:** Bir Online Transaction Processing sistemin erişim deseni, kısa, atomik hareketlerden oluşmaktadır. Bu tür bir sistem kurtarma mekanizmaları ve uyumluluk kontrolü gerektirmektedir. Oysa Online Analytical Processing sistemlere erişim çoğunlukla salt okunurdur. Çünkü birçok veri ambarı güncel veriden çok geçmişe dönük verilerle çalışır.

1.2.6. Veri Ambarı ile Veri Tabanı Arasındaki Farklar

Veri tabanı içerisinde bulunan bilgiler çoğunlukla anlık bilgilerdir. Yani o an için güncel olan fakat belirli bir zaman geçtikten sonra güncelliğini yitirecek olan bilgilerdir. Ancak veri ambarı içerisinde bulunan veriler genel olarak yığılarak birikirler ve verilerin geçerliliği çok daha uzun süre olmaktadır. Veri ambarı içerisinde ne kadar çok kayıt bulunursa yapılan incelemelerin sonucu da o kadar doğru olacaktır. Ancak veri tabanı içerisinde bulunan kayıt sayısının çok fazla olması durumunda, veri tabanını kullanan canlı sistemlerin performansları düşecek dolayısı ile verilere erişim çok yavaşlayacaktır ki bu, canlı sistemlerde istenmeyen durumların başında gelmektedir (Tantuğ 2002: 10).

Veri ambarları; off-line çalışır, veri değişiminden çok sorgulama yapılı, eski veriler saklandığı için veri miktarı çoktur, kullanıcı sayısı azdır ve veri madenciliği gibi uzun ve karmaşık süreçler sonucunda analiz yapılabilir.

Veri tabanları ise; on-line çalışır, veri değişimleri yoğunluktadır, güncel veri saklandığı için veri miktarı daha azdır, kullanıcı sayısı çok ve sorgularla istenilen sonuçlara anında ulaşılabilir.

1.3. Veri Madenciliği, Makine Öğrenmesi ve İstatistik

Veri madenciliği, yapay zekâ (Artificial Intelligence) ve istatistik alanlarındaki gelişmelerden faydalanmaktadır. Her iki disiplin de desen tanıma ve sınıflandırma problemleri üzerinde çalışmaktadır. İki alan da sinir ağları ve karar ağaçları uygulamalarına büyük katkılar sağlamaktadır.

Veri madenciliği geleneksel istatistik tekniklerinin yerine geçmemektedir. Aksine istatistik alanındaki büyük değişikliğin sonucunun parçası içerisinde olan istatistiksel metotların bir gelişimidir. Birçok istatistiksel metodun gelişimi, analiz edilecek mütevazı miktarlardaki veri üzerinde iyi çalışan teorik yapıları ve analitik metotları esas almaktadır. Bilgisayarın gücünün artması ve maliyetlerinin azalması milyonlarca satırdan oluşan veri kümelerini doğurmuştur. Bu yığınlar olası çözümlerin keşfine dayanan yeni tekniklerin gelişimine zemin hazırlamıştır (Delibaş 2010: 13).

Yeni teknikler, diskriminant analizi gibi eski algoritmalarındaki yeni yaklaşımlar, yapay sinir ağları, bulanık mantık, genetik algoritmalar ve karar ağaçları gibi nispeten yeni algoritmaları içermektedir. Büyük miktardaki mevcut veriler üzerinde artan bilgisayar gücünün kullanılmasıyla birlikte, bu teknikler herhangi bir fonksiyonel yapı veya ilişkiyi neredeyse tam olarak tahmin edebilmektedir. Geleneksel istatistik teknikleri, fonksiyonel yapı ve ilişkileri niteleyen modellere dayanmaktadır.

Esas nokta; veri madenciliğinin, bunlar ve diğer yapay zekâ ve istatistiksel tekniklerin uygulaması olması ve eğitilmiş istatistik uzmanlarının yanı sıra nitelikli bilgi işçilerinin de kullanılabileceği bir tarzda yaygın ticari problemlere uygulayabilmesidir. Veri madenciliği tahmin edici modeller ortaya çıkarmaya çalışan insanların üretkenliğini artıran bir yoldur (Edelstein 1999: 4).

1.4. Veri Madenciliğinin Kullanım Alanları

Veri kaynaklarının hızla artmasına bağlı olarak verinin kendisi de hızla artmaktadır. Artık günümüzde devasa veri yığınlarının insanlar tarafından analiz edilebilmesi mümkün olmamaktadır. Bu işlemin otomatikleşmiş hali olarak

değerlendirilebilecek olan veri madenciliği çalışmaları gün geçtikçe daha fazla alanda kullanım alanı bulmaktadır (Güven, Bozkurt ve Kalıpsız 2007).

Veri madenciliğinin kullanım alanları konu başlıkları ile aşağıdaki gibi sınıflandırılabilir (Akpınar 2000):

Bankacılık:

- Riskli müşteri örüntülerinin belirlenmesi,
- Farklı finansal göstergeler arasında gizli korelasyonların bulunması,
- Sigorta dolandırıcılıklarının tespiti,
- Yeni poliçe isteyecek olan müşterilerin tahmin edilmesi,
- Kredi kartı dolandırıcılıklarının tespiti,
- Risk analizleri Sigortacılık,
- Usulsüzlük tespiti,
- Müşteri dağılımı,
- Kredi taleplerinin değerlendirilmesi,
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi.

Pazarlama:

- Çeşitli müşteri analizleri,
- Pazar sepeti analizi,
- Posta kampanyalarında cevap verme oranının artırılması,
- Çapraz satış analizleri,
- Mevcut müşterilerin elde tutulması için geliştirilecek pazarlama stratejilerinin oluşturulması,
- Çeşitli pazarlama kampanyalarında,
- Müşteri dağılımı,
- Müşteri değerlendirme,
- Müşteri ilişkileri yönetimi,

- Müşterilerin satın alma örüntülerinin belirlenmesi,
- Satış tahmini,
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması.

Perakendecilik:

- Alım-satım stratejilerinin optimizasyonu,
- Genel piyasa analizleri,
- Hisse senedi fiyat tahmini,
- Satış noktası veri analizleri,
- Tedarik ve mağaza yerleşim optimizasyonu,
- Alış-veriş sepeti analizleri.

Telekomünikasyon:

- Hatların yoğunluk tahminleri,
- Hisse tespitleri,
- Kalite ve iyileştirme analizleri.

Endüstri - Kalite kontrol analizleri

- Üretim süreçlerinin optimizasyonu,
- Lojistik.

Sağlık ve İlaç:

- Ürün geliştirme,
- Test sonuçlarının tahmini,
- Semptomlara göre hastalık tespiti,
- Tedavi sürecinin belirlenmesi,
- Tıbbi teşhis.

1.5. Veri Madenciliği Sistemlerinin Sınıflandırılması

Birçok veri madenciliği sistemi tasarlanmıştır ve geliştirilmiştir. Bu sistemler veri tabanına, keşfedilecek bilgiye ve faydalanılan tekniklere göre sınıflandırılabilir (Lee, Siau 2001).

1.5.1. Veri Tabanına Göre

İlişkisel, işlemsel, mekânsal, nesne-tabanlı, çoklu-ortam, kalıtım ve web veri tabanları gibi birçok veri tabanı sistemi bulunmaktadır. Bir veri madenciliği sistemi veri tabanlarının tasarlandığı amaca göre sınıflandırılabilir. Örneğin; ilişkisel veri tabanından bilgi keşfi yapan bir sistemi ilişkisel veri madenciliği sistemi, nesne-tabanlı veri tabanından bilgi keşfi yapan bir sistem de nesne-tabanlı veri madenciliği sistemi olarak adlandırılabilir (Delibaş 2010: 15).

1.5.2. Bilgiye Göre

Veri madenciliği sistemleri birliktelik kuralları, karakteristik kurallar, sınıflandırma kuralları, kümeleme, yayılma veya sapma analizleri gibi farklı türden bilgiyi keşfedebilir. Veri madenciliği keşfedilmiş bilgiyi soyutlama düzeylerine göre de sınıflandırılabilir. Genel bilgi, basit düzeyli bilgi, çoklu düzeyde bilgi gibi sınıflandırmalar yapılabilir. Esnek bir veri madenciliği sistemi çoklu soyutlama düzeylerinde bilgi keşfi yapabilmelidir.

1.5.3. Yararlanılan Tekniğe Göre

Veri madenciliği sistemleri, veri madenciliği tekniklerine göre de sınıflandırılabilir. Örneğin; otonom bilişim madenciliği, veriye dayalı veri madenciliği, sorguya dayalı veri madenciliği gibi dayalı olduğu metoda göre ve interaktif veri madenciliği teknikleri şeklinde sınıflandırılabilir. Bunlara alternatif olarak, genelleme temelli madencilik, desen temelli madencilik, istatistiksel veya matematiksel temelli madencilik gibi üzerinde durduğu madencilik gelişimine göre ve bağlı gelişmelere göre de sınıflandırılabilir (Chen, Han ve Yu 1996).

1.6. Veri Madenciliğinde Karşılaşılan Zorluklar

Veri madenciliği sistemi, girdi olarak kullanılacak olan işlenmemiş veriyi veri tabanlarından alır. Bu durumda veri tabanlarının eksiksiz, geniş, net ve dinamik veriler içermemesi durumunda sıkıntılar meydana gelir (Aydoğan 2003: 13).

Küçük veri setlerinde doğru ve hızlı bir şekilde çalışan bir veri madenciliği sistemi, çok daha büyük veri setlerine uygulandığında tamamen farklı biçimde davranabilir. Tutarlı veri üzerinde bir veri madenciliği sistemi mükemmel çalışırken, benzer veriye gürültü eklendiğinde sistem önemli ölçüde kötüleşebilir.

Günümüzde veri madenciliği sistemlerinin karşılaştığı sorunlar şu şekildedir:

1.6.1. Veri Tabanı Boyutu

Veri tabanı hacimleri olağanüstü bir hızla artış göstermektedir. Makine öğrenimi algoritmalarından birçoğu birkaç yüz tutanaklık oldukça küçük örneklemeleri ele alabilecek şekilde geliştirilmiştir. Örüntülerin gerçekten var olduğunu göstermesi açısından, örnekleme büyük olması bir avantajdır fakat böyle bir örneklemeden elde edilebilecek olan olası örüntü sayısı da çok büyük olacaktır. Bu nedenle veri tabanı boyutunun çok büyük olması, veri madenciliği sistemlerinin karşı karşıya olduğu en önemli sorunlardan biridir. Bu yüzden veri madenciliği yöntemleri ya örnekleme ya yatay/dikey olarak indirgemeli ya da sezgisel bir yaklaşımla arama uzayını taramalıdır. Dikeyde indirgeme özelliklerin bulunduğu kolonların azaltılması, yatayda indirgeme ise veri alanının örnekleme çalışmasıdır (Şen 2008: 13).

1.6.2. Gürültü

Birçok verinin değeri büyük veri tabanlarında yanlış olabilir. Bu yanlış, girilen değerlerin yanlış ölçülmesi veya veri girişi esnasında yapılan insan hatalarından kaynaklanır. Gürültü, verinin elde edilmesi veya girişi esnasında oluşan sistem dışı hatalar olarak adlandırılır. Günümüzde kullanılmakta olan ticari ilişkisel veri tabanları, veri girişi esnasında oluşan hataları otomatik şekilde gidermek konusunda az bir destek sağlamaktadır. Hatalı veri gerçek dünya veri tabanları üzerinde çok

ciddi sıkıntılar oluşturabilir. Karşılaşılabilecek olan bu durum, bir veri madenciliği yönteminin kullanılan veri setinde bulunan gürültülü verilere karşı daha fazla duyarlı olmasını gerektirir (Quinlan 1986).

1.6.3. Eksik ve Artık Veriler

Kullanılan veri seti, artık nitelikler veya eldeki probleme uygun olmayan veriler içerebilir. Bir değer bilinmiyor ya da yanlışlıkla girilmemiş olabilir. Veri madenciliğindeki birçok yöntem, her veri nesnesi için sabit bir özellik sayısı gerektirdiğinden, eksik veriler sorun yaratır. Artık veri oluşumunu engellemek için özellik seçimi yapılmalıdır.

1.6.4. Dinamik Veri Yapısı

Online veri tabanları içerisindeki veriler periyodik olarak değişir. Bu durum da, bilgi keşfi metotları için bazı dezavantajlara neden olmaktadır. İlk olarak uzun süre çalışan ve sadece okuma yapan bilgi keşfi metodu, bir veri tabanı uygulaması olarak var olan veri tabanı ile beraber çalıştırıldığında uygulamanın da performansı önemli ölçüde düşer. İkinci olarak, veri tabanının içerisinde bulunan verilerin kalıcı olduğu düşünülüp, çevrim dışı veri tabanı üzerinde bilgi keşif metodu çalıştırıldığında, değişen verinin elde edilen örüntülere yansımaları gerekmektedir. Burada kuralların değişip değişmediği ve istikrarlılığı sorunu ortaya çıkar (Hulten, Spencer ve Domingos 2001).

1.7. Modellerine Göre Veri Madenciliği

Veri madenciliğinde kullanılan modelleri tahmin edici (Predictive) ve tanımlayıcı (Descriptive) olmak üzere iki ana başlık altında toplayabiliriz. Tahmin edici modeller ile tanımlayıcı modeller arasındaki ayrım çok keskin değildir. Tahmin edici modeller anlaşılabilir olduğu ölçüde tanımlayıcı model olarak, tanımlayıcı modeller de tahmin edici model olarak birbirlerinin yerine kullanılabilirler (Velickov, Solomatine 2000).

1.7.1. Tanımlayıcı Modeller

Analiste daha önceden bir hipoteze sahip olmaksızın, veri setinin içinde ne tür ilişkiler olduğunu anlama fırsatı sunar. Analizcinin büyük veri tabanlarındaki bilgileri incelemek, örüntüleri bulmak için doğru soruları sorup hipotezler geliştirmesi uygulamada zor olduğundan, ilginç örüntüleri bulma önceliği veri madenciliği programına bırakılır. Elde edilen bilginin kalitesi ve zenginliği, uygulamanın kullanılabilirliğini ve gücünü oluşturur (Güvenç 2001: 120).

Tanımlayıcı modellerde karar vermeyi, rehberlik etmede kullanılabilir olan mevcut verilerdeki örüntülerin tanımlanması sağlamaktadır. 25 yaş altı bekâr kişiler ile 25 yaş üstü evli kişiler üzerinde yapılan ve ödeme performanslarını gösteren bir analiz tanımlayıcı modellere örnek olarak verilebilir (Tiryaki 2006: 15). Kümeleme, birliktelik kuralları, çok kullanılan tanımlayıcı modellerdir.

1.7.2. Tahmin Edici Modeller

Tahmin, geçmişteki deneyimlerden elde edilen bilgiler ile mantık kullanılarak, gelecekte bir olay, bir durum hakkında öngöründe bulunmaktır.

Tahmin edici modellerde, sonuçları daha önceden bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır (Akpınar 2000). Örneğin bir sınıftaki öğrencilerin bir dersle ilgili almış oldukları vize ve ödev notları gibi veriler bir veri tabanında toplanabilir. Bu verilere uygun olarak kurulan model öğrencilerin o dersin sonunda finalden alacağı notun tahmininde kullanılmaktadır.

1.8. Veri Madenciliği Teknikleri

Gerek tanımlayıcı gerekse tahmin edici modellerde yaygın olarak kullanılan belli başlı teknikler; Sınıflama ve Regresyon Sorgusu, Kümeleme Sorgusu, Ardışık Zamanlı Örüntüler, Birliktelik Kuralları Madenciliği olarak sıralanabilir. Kümeleme, birliktelik kuralları ve ardışık örüntü modelleri tanımlayıcı, Sınıflama ve Hipotez Testi modelleri tahmin edici modellerdir (Bramer 2007: 9).

1.8.1. Sınıflandırma ve Regresyon Sorgusu

Sınıflama veri madenciliğinde yoğun olarak kullanılan bir yöntem olup, veri tabanlarındaki gizli örüntülerin ortaya çıkartılmasında kullanılır. Verilerin sınıflandırması için belirli bir süreç takip edilir. İlk olarak mevcut veri tabanının bir bölümü eğitim amacıyla kullanılarak sınıflandırma kurallarının oluşturulması sağlanır. Daha sonra bu kurallar sayesinde yeni bir durum ortaya çıktığında nasıl karar verilmesi gerektiği belirlenir (Özkan 2008: 158).

Sınıflama sorgusu kullanılarak bir kaydın daha önceden nitelikleri belirlenmiş bir sınıfa girmesi hedeflenmektedir. Sınıflama algoritması öğrenme verilerini kullanarak hangi sınıfların var olduğu ve bu sınıflara girebilmek için kayıtların hangi niteliklere sahip olması gerektiğini otomatik olarak keşfeder (Şen 2008: 16).

1.8.1.1. Yapay Sinir Ağları

Yapay sinir ağları tekniği, veri madenciliğinde geniş bir kullanım alanına sahiptir. Desen tanıma, makine öğrenmesi ve yapay sinir ağları gibi teknikler dikkat çekmektedir. Bilgi edinme, bilgi sunma ve araştırma gibi yapay zekâda kullanılan diğer teknikler veri madenciliğindeki sürecin çeşitli adımları ile uyuşmaktadır (Lee, Siau 2001).

Sınıflandırma problemine bir çözüm olarak yapay sinir ağları da gösterilebilir. Lu, Setiono ve Liu (1996)'ya göre sinir ağları tabanlı veri madenciliği yaklaşımı üç aşamadan meydana gelmektedir:

- *Ağ yapısı ve eğitimi:* Bu aşamada nitelik sayısı, sınıf sayısı ve seçilen giriş kodlama metodunun sayısına dayanan katmanlandırılmış sinir ağı eğitilir ve yapılandırılır.
- *Ağ budaması:* Gereksiz bağlantılar ve birimler ağın sınıflandırma hata aralığını artırmadan silmeyi amaçlamaktadır. Budama neticesinde ağda az miktarda birim ve bağlantının kalması bizlere kısa ve anlaşılır kurallar çıkarmamıza imkân sağlamaktadır.
- *Kural çıkarımı:* Sınıflandırma kuralları bu aşamada ortaya çıkar.

Diğer yapay zekâ teknikleri duruma dayalı nedenleme ve zekâ faktörlerini içeren veri madenciliği için kullanılabilir. Duruma dayalı nedenleme desenleri tanımak için geçmiş durumları kullanır, zekâ faktörü ise veri içerisinde eleme yapmak için bir bilgisayar programı kullanır.

1.8.1.2. Genetik Algoritmalar

Genetik algoritmalar, nispeten yeni denebilecek bir yazılım modelidir. Her biri bir problemin olası çözümünü sunan kurallar popülasyonu başlangıçta rastgele oluşturulur. Daha sonra yeni nesiller için yavrular oluşturmak üzere kurallar çifti – genellikle en güçlüleri ebeveyn olarak seçilecek şekilde kombine edilir. Bir mutasyon işlemi her yeni neslin bazı üyelerinin genetik yapısını rastgele değiştirmek için kullanılır. Sistem onlarca veya yüzlerce nesil çalıştırılır. Bu işlem kabul edilebilir veya optimum sonuç bulunana kadar veya belirlenen bir zaman kısıtına kadar devam eder. Genetik algoritmalar bazı hesaplanabilir kriterlere göre optimizasyon gerektiren problemler için uygundur. Bu model veri madenciliğine uygulanabilir özelliktedir. Minimize edilecek sayı çoğunlukla bir eğitim verisi üzerindeki sınıflandırma hatalarının sayısı olmaktadır. Büyük ve karmaşık problemlerin çözümünde makul sürelerde sonuç almak için hızlı bilgisayarlara ihtiyaç duyulmaktadır. Büyük boyutlardaki veri kümelerinde bu teknikte veri madenciliği yapmak yalnızca kullanılabilir en hızlı bilgisayarlara kullanışlı hale gelecektir (Lee, Siau 2001).

1.8.1.3. Bulanık Veri Madenciliği

Bulanık veri madenciliği, araştırmanın aktif bir alanıdır. Gerçek ve sayısal veriyle çalışırken, bulanık küme teorisinin gücü takdir edilmekte ve bilgi keşfi sürecine sağlamlık ve anlaşılabilirlik getirmektedir. Bilinen bir bulanık veri madenciliği algoritması, bulanık karar ağaçlarının oluşturulmasına dayanır. Bulanık karar ağaçları, klasik karar ağaçlarının sayısal ve bulanık verileri ele alan bir uzantısıdır (Bouchon-Meunier, Marsala 1999).

Veri madenciliği ifadesi pek çok uygulamayla verilen sonuç çıkarımı üzerine vurgu yaparken, belirsizlik muhakemesi ifadesi belirsizlik söz konusu olduğunda yeni bilginin çıkarımı anlamı üzerine odaklanır. Buna karşın, belirsizlik muhakemesi,

veri madenciliği hedefi üzerinde etkin bir başarıya ulaşır. Aynı zamanda belirtmek lazım ki, bu belirsizlik muhakemesi teknikleri veri tabanı çevresinde analizler yapmak üzere de geliştirilmiştir (Koyuncugil 2006: 93).

1.8.1.4. Karar Ağaçları

Veri madenciliği kullanışlı bilgi desenlerini keşfetmek amacıyla büyük ve karmaşık veri bloklarını araştıran bir teknoloji ve bilim dalıdır. Teorisyenler ve pratisyenler bu işlemleri daha etkili, uygun maliyetli ve hassas bir süreç haline getirmek için sürekli gelişmiş teknikler üzerinde çalışmaktadır. Bu bağlamda en umut verici ve popüler yaklaşımlardan biri karar ağaçlarının kullanımınıdır. Karar ağaçları bir nesne ile hedef değeri arasındaki ilişkilerin tahmin edilmesi ve açıklanması konusunda basit ama başarılı bir tekniktir.

Aslında mantık, yönetim ve istatistik alanlarından türeyen karar ağaçlarının bu günlerde veri madenciliğinde kullanımı metin madenciliği, bilgi çıkarımı, desen tanıma ve makine öğrenmesi gibi alanlarda etkin bir şekilde görülmektedir (Rokach, Maimon 2008: 77).

Karar ağaçları, her bir düğümü bir nitelik değerinin kriterine denk gelen, her bir dalı bir kriterin sonucunu gösteren ve yaprakları da sınıflar veya sınıf dağılımlarını temsil eden akış diyagramı benzeri ağaç yapılarıdır. Karar ağaçları kolaylıkla sınıflandırma kurallarına dönüştürülebilir. Ayrıca regresyon, kümeleme ve özellik seçimi gibi veri madenciliği işlemlerinde de geniş bir uygulama alanına sahiptir. Karar ağaçlarının kolay okunabilirliği; nominal, nümerik veya metin girdileri kullanabilme esnekliği; hata veya kayıp değer içeren veri kümelerini işlemedeki uyum yeteneği; küçük işlem gücü ile yüksek tahmin performansları gösterebilmesi ve büyük veri kümeleri için kullanışlılığı sayabileceğimiz belli başlı özellikleri arasındadır (Han, Kamber 2006: 292; Rokach, Maimon 2008: 77).

1.8.2. Kümeleme

Kümeleme işlemi yönteminde amaç üyelerinin birbirlerine çok benzediği, fakat özellikleri bakımından birbirlerinden çok farklı olan kümelerin ortaya çıkarılması ve veri tabanındaki kayıtların bu farklı olan kümelere ayrılmasıdır. Kümeleme işlemi yönteminde; veri tabanındaki verilerin hangi kümelere bölüneceği

veya kümeleme işleminin hangi değişken özelliklerine göre yapılacağı, bu konuda uzman olan bir kişi tarafından belirtilebileceği gibi, verilerin hangi kümelere ayrılacağını geliştirilen yazılımlar sayesinde de yapabilmektedir. Kümeleme işlemi; istatistik ve makine öğrenmesi, web madenciliği, biyoloji gibi pek çok alanda kullanılır. Kümeleme işlemi yönteminde, sınıflama işlemi yönteminde olan veri sınıfları yoktur (Han, Kamber 2000: 25).

Kümeleme işlemi tekniği sınıflandırma işleminin tersine, veri setini önceden sınıflara ayırmak yerine veriler dağılımlarına göre değerlendirilerek doğal sınıflandırmalar oluşturur. Kümeleme işlemi tekniğinin sınıflandırma işleminden en temel farkı önceden belirlenmiş sınıflar ya da sınıf tanımları kullanmamasıdır. Kümeleme işleminde ana kural sınıf içi benzerliği en yüksek, sınıflar arası benzerliği en düşük yapmaktır (Gürgen 2008: 14).

Kümeleme ile sınıflamayı birbirinden ayıran en temel fark, kümeleme işleminin sınıflama işlemi gibi önceden belirlenmiş sınıflara göre bölme yapmamasıdır. Kümeleme işleminde önceden tanımlanmış sınıflar ya da örnek sınıflar bulunmamaktadır. Sınıflama işleminde ise her bir veri, önceden sınıflandırılmış olan bir takım sınıflar üzerinde yapılan bir eğitim sonucunda ortaya çıkan bir modele göre önceden belirlenmiş bir sınıfa atanmaktadır. Verilerin kümelenebilmesi, verilerin birbirlerine olan benzerliklerine göre yapılmaktadır. Oluşan sınıfların taşıdığı anlamların belirlenmesi tamamen çözümlemeyi yöneten kişiye kalmıştır (Ergün 2008: 53).

En yaygın kullanılan kümeleme algoritması k-ortalamlar algoritmasıdır. Diğer kümeleme teknikleri ile karşılaştırıldığında k-ortalamlar algoritması (k-means) büyük veri tabanlarının kümelenebilmesinde oldukça etkin bir algoritmadır. Yeni bir olay ortaya çıktığında; algoritma tüm veriyi inceleyerek buna en çok benzeyen olayların bir altkümesini oluşturur ve onları sonucu tahmin etmek için kullanır (Anand 2003: 21).

Kümeleme çalışmalarında birçok algoritma kullanılmaktadır. Algoritmalar kümelemenin oluşturuluş şekline göre de ayrılabilir, kullanılan veri türüne, yapılacak çalışmanın amacına göre de ayrılabilir. Genel anlamda kümeleme algoritmaları,

hiyerarşik ve bölümlenmeli olarak iki grupta incelenebilir. Bu iki grup altında da farklı algoritma ve yöntemler söz konusudur (Silahtaroglu 2008: 40):

- Hiyerarşik Yöntemler
 - Aglomeratif kümeleme algoritmaları
 - Bölünür (Divisive) kümeleme algoritmaları
- Bölümlenmeli Yöntemler
 - Yer değiştiren algoritmalar
 - Olasılıklı algoritmalar
 - K-medoid yöntemler
 - K-means yöntemler
 - Yoğunluğa dayalı algoritmalar
- Grid temelli yöntemler
- Kısıtlara dayanan yöntemler.
- Kategorik verinin yinelenmesine dayalı yöntemler.

1.8.3. Ardışık Zamanlı Örüntüler

Ardışık zamanlı örüntü, birbirleri ile ilişkisi olan ve birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Belli bir olayın gerçekleşmesinden sonra aynı olayla ilintili bir başka olayın gerçekleşmesi ardışık zamanlı örüntü kavramını oluşturur.

Örneğin;

- A ameliyatı yapıldığında 20 gün içerisinde %40 olasılıkla B enfeksiyonu oluşacaktır.
- Çamaşır makinesi alan bir müşteri 6 ay içerisinde, %70 olasılıkla kurutma makinesi de alacaktır. Mağaza %70 olasılığı arttırmak için çamaşır makinesinin alımından 4 ay sonra, kurutma makinelerinde %10 indirim uygulayabilir (Ergün 2008: 56).

1.8.4. Birliktelik Kuralları Madenciliği

Birliktelik kuralları madenciliği (Birliktelik kuralları) ürün katalog tasarımından, satış-pazarlamaya kadar birçok farklı alanda kullanım alanı bulmaktadır. Örneğin, herhangi bir ürünü satın alırken, bu ürünün yanında farklı bir ürün ya da ürünlerin satın alınması durumu, bu ürünler arasındaki ilişkiyi ortaya koymaktadır. Bu bağlantıların ortaya çıkarılması ve bunun bir kural olarak ortaya konması ise birliktelik kuralları konusuna girer. Literatürde bu tarz çalışmalara market sepeti analizi denilmektedir. Market sepeti analizi müşterilerin alışveriş alışkanlıklarının veri tabanındaki bilgiler aracılığıyla ortaya çıkartılması işlemidir. Bu alışveriş alışkanlıklarının ortaya çıkartılması alış-veriş merkezindeki ürünlerin yerleşmesi, marketin yerinin tasarımı, markette sunulacak ve satılacak olan ürünlerin belirlenmesinde yardımcı olmaktadır (Silahtaroglu 2008: 83).

Birliktelik kuralları, bir arada gerçekleşen olayların ya da özelliklerin belirlenmesi süreci işlemidir. Birliktelik kuralları genellikle “eğer şu olursa daha sonra bu olur” şeklinde ifade edilir. Çoğunlukla açıklayıcı veri analizinde, veri ön işlemede, ayrıık değerleri tespit etmede, eğilim ve ilişkilerin bulunmasında kullanılır (Fayyad, vd. 1996: 28).

Bir birliktelik algoritması oluşturmadan önce kurallar belirlenmelidir. Büyük veri tabanında ilişkileri bulacak algoritmalar geliştirmek çok zor değildir. Fakat geliştirilen algoritmalar önemli ilişkileri ortaya çıkaracağı gibi önemsiz birçok ilişkiyi de ortaya çıkarır. Bu yüzden, büyük veri tabanlarında küçük alt kümeler bulunmalıdır.

İşlemsel veya ilişkiyel veri tabanlarından birliktelik kurallarının çıkarılması, veri tabanı alanındaki en dikkat çekici konu başlığını teşkil etmektedir. Asıl amaç, veri tabanlarındaki birleştirilmiş veri kümelerinden, nitelik değer kümesi olan A_i ($i \in \{1, \dots, m\}$) ve B_j ($j \in \{1, \dots, m\}$) içinden “ $A_1 \wedge \dots \wedge A_m \Rightarrow B_1 \wedge \dots \wedge B_n$ ” şeklindeki güçlü birliktelik kurallarını açığa çıkarmaktır. Birliktelik kurallarını çıkarma işi büyük bir işlemsel veri tabanı içinde defaten tarama yapmayı gerektirdiğinden, işlem gücünün muazzam seviyede olması zorunludur (Lee, Siau 2001).

Birliktelik analizine 2. bölümde daha detaylı olarak değinilecektir.

1.8.5. Anormallik Tespiti

Çoğu zaman, verilerin genel davranışına veya modeline uymayan veri nesnelere vardır. Geriye kalan veri kümesiyle büyük ölçüde farklı olan veya bunlarla tutarlı olmayan bu tür veri nesnelere “anormal” olarak adlandırılır. Anormallikler, ölçüm veya yürütme hatasından kaynaklanabilir. Örneğin, bir kişinin yaşının -999 olarak gösterilmesi, kayıtlı bir yaşın program varsayılan ayarından kaynaklanıyor olabilir. Alternatif olarak, anormal değerler belki de doğal veri değişkenliğinin bir sonucudur. Örneğin bir şirketin yöneticisinin maaşı, doğal olarak firmanın diğer çalışanların maaşlarına göre anormal olarak göze çarpıyor olabilir.

Birçok veri madenciliği algoritmaları, anormallik etkisini en aza indirmeye veya tamamını ortadan kaldırmaya çalışmaktadır. Ancak bu, önemli gizli bilgilerin kaybolmasına neden olabilir çünkü bir kişinin sesi bir başka kişinin sinyali olabilir. Başka bir deyişle, anormalliklerin hileli faaliyet gösterebileceklerin sahtekârlık tespiti durumunda olduğu gibi anormallik özel bir ilgiden kaynaklanabilir. Böylece, anormallik tespit ve analizi “anormallik madenciliği” olarak anılan ilginç bir veri madenciliği çalışmasıdır (Delibaş 2010: 19).

Anormallik madenciliğinin çok geniş bir uygulama alanı vardır. Örneğin, kredi kartlarının veya telekomünikasyon servislerinin olağandışı kullanımını tespit ederek, dolandırıcılık tespitinde kullanılabilir. Ayrıca, özelleştirilmiş pazarlamada, aşırı derecede düşük veya aşırı yüksek gelirli müşterilerin harcama davranışlarını belirlemek için veya çeşitli tıbbi tedavilere alışılmadık cevaplar bulmak için tıbbi analizde yararlıdır.

Anormallik madenciliği şu şekilde tarif edilebilir:

n veri noktaları veya nesnelere ile k adet anormallik beklenen sayısı verildiğinde, kalan verilere göre dikkate değer derecede benzemeyen, farklı ve tutarsız veriler ile ilgili k adet uç nokta bulunur. Anormallik madenciliği iki alt problem olarak görülebilir:

- Verilen veri kümesinde hangi veri tutarsız olarak kabul edilebileceğini tanımlamak,

- Belirlenen anormallik çıkarmak için etkin bir yöntem bulmaktır (Han, Kamber 2006: 451).

1.8.6. Görselleştirme

Görsel veri madenciliği büyük veri kümelerinden üstü kapalı ve kullanışlı bilgiyi veri / bilgi görüntüleme teknikleri ile keşfetmektedir. İnsanda görsel sistem, gözler ve sonraki aşamada ise büyük bir bilgi tabanı içeren güçlü bir paralel işlem ve sonuçlandırma mekanizmasına sahip olan beyin ile kontrol edilmektedir. Görsel veri madenciliği temel olarak bu bileşenlerin gücünü, veri dağılımı, desenler, kümeler ve sapkın değerlerin kavranması için oldukça ilgi çekici ve etkili bir araç haline getirerek birleştirmektedir. Görsel veri madenciliği, veri madenciliği ve veri görüntüleme disiplinlerinin birleşimi olarak görülebilir. Aynı zamanda bilgisayar grafikleri, çoklu-ortam sistemleri, desen tanıma ve yüksek performanslı hesaplama sistemleri ile yakından ilişkilidir (Han, Kamber 2006: 667).

1.8.7. Veri Madenciliğinde İstatistiksel Teknikler

Lee ve Siau (2001)'ya göre istatistik veri seçme, örnekleme, veri madenciliği ve ortaya çıkan bilginin değerlendirilmesinin kaçınılmaz bir bileşenidir. İstatistik veri madenciliğinin sonuçlarının değerlendirilmesinde iyi ile kötünün ayrılmasında kullanılmaktadır. Veri temizleme sürecinde istatistik verideki aykırılıkların tespiti, gerektiğinde veriyi düzeltmek ve gürültüyü kestirebilmek için teknikler sunmaktadır.

Kümeleme ve deneylerin dizaynındaki teknikler keşif amaçlı veri analizi için oyuna katılmaktadır. İstatistikte çalışma genel olarak teknik ve modellerin teorik yönünün üzerinde durur. Sonuç olarak veri madenciliğinde önem arz eden araştırma, küçük çapta dikkate alınır. Buna ek olarak, veri tabanı tarafında, büyük veri kümeleri ile ilgilenen teknikler ve etkili veri yönetimi için kullanılan teknikler veri madenciliği için çok önemli konulardır. Ancak bu konular istatistikte henüz dikkate alınmaya başlanmıştır (Lee, Siau 2001).

İKİNCİ BÖLÜM

2. BİRLİKTELİK KURALLARI MADENCİLİĞİ

Veri madenciliği, veri sahibi için hem anlaşılabilir hem de faydalı olma ilkesiyle beklenmeyen ilişkileri bulmak ve yeni yöntemlerle veriyi özetlemek için genellikle büyük boyutlardaki veri kümelerinin analiz edilmesidir. Bir veri madenciliği uygulamasından elde edilen ilişkiler ve özetler genellikle modeller ve ya desenler olarak adlandırılmaktadır. Doğrusal denklemler, kurallar, kümeler, grafikler, ağaç yapıları ve zaman serilerindeki tekrarlı desenler bunların bazılarıdır. Birliktelik kuralları veri madenciliğindeki bölgesel desenlerin en popüler temsilcileri arasındadır (Kumar, Wahidabanu 2008).

Birliktelik kuralı madenciliği keşfi, sınıflandırma kuralı ile birçok ortak özelliği paylaşmaktadır. Her ikisi de bir veri kümesinde düzenlilikleri karakterize etmek için kurallar kullanır. Fakat bu iki kural keşfi modeli amaçları açısından büyük farklılıklar göstermektedir. Sınıflandırma kuralı tahmin yapma kapasitesi edinme odaklanırken, birliktelik kural keşfi kullanıcıya bilgi sağlamaya odaklanmaktadır. Özellikle, veri öğeleri arasındaki beklenmedik ilişkileri saptama ve karakterize etme üzerine odaklanmaktadır (Webb 2003).

Birliktelik kuralları madenciliğinin en yaygın uygulama alanı perakende satışlarda müşterilerin satın alma yönelimlerinin belirlenmesi için uygulanmaktadır. Müşterilerin, bir seferde satın aldığı tüm ürünlerin ele alınması ile satın alma eğilimlerinin ortaya koyulduğu bu uygulamalara "Pazar sepet çözümü" denilmektedir (Coşlu 2013).

Son dönemlerde, veri toplama ve otomatik tanıma uygulamalarında yaşanan gelişmeler ile işletmelerin barkot sistemleri kullanımı yaygınlaşmaya başlamıştır. Yaşanan bu gelişmeler sayesinde bir işleme ait satış verilerinin elektronik ortamlara aktarılması sağlanmıştır. Çoğunlukla büyük süpermarketlerde elde edilen bu tür verilere market sepet verisi denilmektedir. Kuruluşların çoğu, bu verileri kullanarak verilerden avantajlar sağlamayı amaçlamaktadır (Karabatak 2008: 24).

2.1. Market Sepet Analizi

İlk olarak 1993 yılında, market sepet verisi üzerine birliktelik kuralı problemi ele alınmıştır. Amaç ürün satışları arasındaki bağıntıyı bulmaktır. Bu bağıntının ortaya çıkarılması şirket kârını arttırmak için kullanılabilir. Market sepet analizi günlük işlemler sonucunda elde edilen verilerden anlamlı ilişkiler çıkarmada kullanılır. “Eğer A malını alıyorsa % x ihtimalle B malını almaya da meyillidirler” şeklinde ki bir sonuç A malını satan bir mağaza için çok yararlı bir bilgi olabilmektedir. Market sepet analizi uygulamaları; mağaza raflarının düzenlenmesi (layout), çapraz satış (crossselling), katalog tasarımı ve fiyatlandırma (pricing) gibi çeşitli alanlarda kullanılmaktadır (Karabatak 2008: 24).

Veri Madenciliğinde, müşterilerin satın alma eğilimlerinin tanımlanmasında kullanılan ardışık zamanlı örüntüler ve birliktelik kuralları Pazar Sepet Analizi adı ile yaygın olarak kullanılmaktadır. Yapılan alışverişlerde müşterinin hangi ürün veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi için kullanılan bu yöntem, bir müşteriye daha fazla ürünün satılması yöntemlerinden biridir. Birliktelik kuralları müşteri alışveriş davranışlarının belirlenmesinde etkili olan bir yöntemdir (Timor, Ezerçe ve Gürsoy 2011: 130).

Büyük miktarda veri sürekli toplanıp depolanırken, pek çok endüstri tarafından araştırılarak işe yarar sonuçlar alınması konusunda ilgi çekici hale gelmektedir. Büyük miktarda ticari işlem kayıtları arasında ilginç korelasyon ilişkilerinin keşfi, katalog tasarımı, pazarlar arası ticaret ve müşteri alışveriş davranış analizi gibi birçok iş karar verme sürecinde yardımcı olabilmektedir. Sık görülen ürün birliktelikleri madenciliğini için tipik bir örnek pazar sepeti analizidir. Bu süreç, müşterilerin "alışveriş sepetlerine" yerleştirdikleri farklı ürünler arasındaki birliktelikleri bularak müşterinin satın alma alışkanlıklarını analiz etmektedir. Bu tür ilişkilerin keşfi, perakendecilerin hangi öğeleri müşterilerin sıklıkla birlikte satın aldığına dair fikir edindiği şekilde pazarlama stratejileri geliştirmelerine yardımcı olabilir. Örneğin, müşteriler süt satın alıyorsa, süpermarkete aynı seferde ekmek (ve ne tür ekmek) almaya ne kadar ihtimal verilecekler? Bu bilgiler, perakendecilere seçici pazarlama yapmalarına ve raf alanlarını planlamalarına yardım ederek satışların artmasına neden olabilir (Han, Kamber 2006: 228).

2.2. Birliktelik Kuralları

Birliktelik kuralları, veri madenciliği alanında üzerinde çok fazla araştırma ve uygulama yapılmış bir konudur. Benzer işlem içinde genellikle birlikte görülen nesnelere barındıran kurallardır.

Birliktelik kurallarının uygulandığı en bilinen örnek market sepet uygulamasıdır. Bu uygulama, müşterilerin yapmış oldukları alışverişlerde ürünler arasındaki ilişkileri bularak müşterilerin satın alma davranışlarını belirler. Bu tip ilişkilerin belirlenmesi, müşterilerin hangi ürünleri birlikte aldıkları bilgisini ortaya çıkarır ve market sahipleri de bu bilgi dâhilinde etkili satış stratejileri geliştirebilir, raf düzenlerini belirleyerek satış oranlarını artırabilirler. Market sepeti analizinin son dönemlerde çok fazla ilgi ile karşılaşmasının nedeni kullanım kolaylığı ve kolay anlaşılmasıdır. İlk olarak 1993 yılında Agrawal ve diğerleri tarafından ele alınmıştır.

Agrawal, Imielinski ve Swami (1993) birliktelik kurallarını şu şekilde tanımlamışlardır:

$I = \{I_1, I_2, \dots, I_m\}$ ürünler diyeceğimiz bir binary nitelikler kümesi olsun.

$T = \{t_1, t_2, \dots, t_n\}$ ise veri tabanındaki işlemleri gösterecek. Her bir t_k nin alacağı değer 0 veya 1'dir. Eğer $t_k = 0$ ise I_k satın alınmamış, eğer $t_k = 1$ ise I_k satın alınmış demektir. Her bir işlem için veri tabanında ayrı bir kayıt bulunmaktadır. Şimdi $X \subseteq I$ için X 'teki her bir I_k ya karşılık gelen t_k değeri $t_k = 1$ 'dir.

Bir birliktelik kuralı ile şu ifade edilmektedir:

$X \Rightarrow I_j$, X, I 'nin bir alt kümesidir. I_j ise I içindeki herhangi bir elemandır ve bu eleman X içinde yer almamaktadır. $X \Rightarrow I_j$ kuralının T için uygun olduğunun söylenebilmesi için belli bir güven seviyesinden söz etmek gerekecektir. Yani, T içindeki tüm X 'lerin ne kadarının I_k 'yı sağladığı %c değeriyle ifade edilmelidir. Bu durumda birliktelik kuralını $0 \leq c \leq 1$ güven seviyesiyle birlikte şöyle ifade edebiliriz: $X \Rightarrow I_j | c$. Güven seviyesi, kuralın gücünü de ifade etmektedir.

Bahsedilen ilişkilerin ifade edilmesinde kullanılan iki değer bulunmaktadır: *Destek ve Güven*. Bu değerler sayısal değerlerdir ve bunları tanımlamak için bazı

nümerik terimler tanımlamamız gerekmektedir. D işlemlerin veri tabanı, N de D deki işlemlerin sayısı olsun. Her D_i işlemi bir ürün kümesidir. $Destek(X)$ de X ürün kümesini içeren işlemlerin oranı olsun:

$$Destek(X) = \frac{|\{I \mid I \in D \wedge I \supseteq X\}|}{N}$$

I bir eleman kümesi ve $|\cdot|$ de kümenin eleman sayısını göstermektedir.

Bir birliktelik kuralının destek değeri, önceki ve sonrakinin her ikisinin de bulunduğu işlemlerin toplam işlem sayısına oranıdır. Güven değeri ise, öncekini içeren işlemlerin aynı zamanda sonrakini de buldurma oranıdır. $A \Rightarrow C$ birlikteliği için destek ve güven değerleri aşağıdaki gibidir (Webb 2003):

$$\begin{aligned} destek(A \Rightarrow C) &= destek(A \cup C) \\ güven(A \Rightarrow C) &= destek(A \cup C) / destek(A) \end{aligned}$$

Yüksek güven ve destek değerlerine sahip kurallara güçlü kurallar denir. Kullanıcı tarafından minimum destek eşik değeri (min_destek) ve minimum güven eşik değeri ($min_güven$) belirlenir. Belirlenen eşik değerlerinden yüksek değerdeki birliktelik kuralları dikkate alınır ve ilginç olarak ifade edilir. İlginç bir desen, bilgi olarak ifade edilir. Genel olarak bu değerler 0 - 1 aralığından çok 0% - 100% aralığında ifade edilmektedir (Han, Kamber 2006: 230). Verilen bir işlemler kümesinde birliktelik kurallarının amacı, kullanıcı tarafından belirlenen minimum destek ve minimum güven değerinden büyük $A \Rightarrow C$ kurallarının bulunmasıdır.

Örneğin bir A ürününü satın alan müşterilerin B ürününü de satın alma birliktelikleri aşağıdaki kural ile ifade edilir.

$$A \Rightarrow B[destek = \%5, güven = \%70]$$

Buradaki destek ve güven değerleri, kuralın ifadesi için kullanılan değerlerdir ve şu şekilde anlamlandırılır: Analiz edilen satış işlemlerinin %5'inde A ve B ürünleri birlikte alınmıştır ve A ürününü satın alan müşterilerin %70'i B ürününü de satın almıştır.

Birliktelik kurallarının yararlı olması için hem kolay anlaşılabilir hem de konu ile ilgili olması gerekir. Birliktelik kuralları sezgisel ve simgesel yapıda olduğundan kolay anlaşılabilirlik daima birliktelik kurallarının baskın yönünü

oluşturmuştur. Birliktelik kurallarında, kullanıcının kuralların sayısını ve tipini kontrol edebileceği farklı yollar vardır. En yaygın olarak kullanılanı, eşik değerleri olarak da bilinen minimum güven ve minimum destek değerlerinin belirlenmiş olduğu yöntemdir. Bu yöntemde yalnızca kullanıcı tarafından belirlenen eşik değerlerinden yüksek olan güven ve destek değerlerine sahip olan kurallar bulunur ve kullanılır. Bir diğer yöntemde kullanıcının sınırlanmış nesne tanımlamasıdır. Tanımlanan bu sınırlanmış nesne, kuralların içeriğinin sınırlanmasında kullanılmış olan mantıksal bir ifadedir. Örneğin, sınırlanmış nesne hamburger, cips ve kola olsun. Sadece hamburger, cips ve kola içeren kurallar ile ilgilenilir (Döşlü 2008: 27).

Birliktelik kurallarındaki bir işlemin ve bir nesnenin tanımı uygulamaya bağlıdır. Market sepeti analizinde; nesnelere, müşterilerin aldığı ürünler ve yapmış olduğu işlem, beraber alınan bütün nesnelere kümesini oluşturur. Birliktelik kurallarında yoğun olarak kullanılan birkaç önemli terim vardır. Bu terimler; kuralın sol tarafını ifade eden öncül (antecedent), kuralın sağ tarafını ifade eden sonuç (consequent), destek değeri, güven değeri, minimum destek değeri (min_destek), minimum güven değeri (min_güven), nesne kümesi, yaygın nesne kümesi ve aday nesne kümesidir (Dolgun 2006: 36).

Birliktelik kural analizi iki adımlı bir işlem olarak incelenebilir:

1. Tüm sık görülen ürün kümelerini bul. Belirlenen bu ürün kümelerinin sıklığı önceden belirlenmiş minimum destek değeri ile tanımlanır.
2. Sık görülen ürün kümelerinden güçlü birliktelik kurallarını oluştur. Bu kurallar minimum destek ve güven değerini sağlayan kurallar olarak tanımlanır (Han, Kamber 2006: 345).

Birliktelik kurallarında kullanılan bir değer hesaplama ise “lift” değeridir. Lift değeri şu şekilde hesaplanır:

$$Lift(A \Rightarrow C) = \frac{güven(A \Rightarrow C)}{destek(C)}$$

İşlem sonucunun 1’den küçük olması A’nın görülmesinin C’nin görülmesi üzerinde negatif korelasyona sahip olduğunu; 1’den büyük olması A’nın görülmesinin C’nin görülmesi üzerinde pozitif korelasyona sahip olduğunu ifade

eder ki bunun anlamı, birinin görülmesi ile diğerinin görülmesi ilişkilidir. Eğer lift değeri 1 çıkmış ise bu iki tarafın birbirinden bağımsız olduğu anlamına gelir.

Şimdi destek, güven ve lift değerlerini bir örnek üzerinde uygulayarak açıklayalım. Bilgisayar ürünleri satışı yapan bir firmada yapılan 10000 satışın analizi neticesinde veriler, müşteri işlemlerinin 7500 adetinde video film satın alınırken 6000 adetinde bilgisayar oyunu alındığını, satış işlemlerinin 4000 adetinde ise her iki ürünün de satıldığını göstermektedir. Buradaki birliktelik kurallarının keşfi için yapılan işlemde, min_destek değerinin %30, min_güven değerinin %60 verilerek aşağıdaki kural elde edilmiştir:

$$oyun \Rightarrow video [destek = \%40, güven = \%66]$$

Elde edilen kural min_destek ve min_güven eşik değerlerinin her ikisini de aşan değerlere sahip olduğundan güçlü bir birliktelik kuralıdır. Fakat bu değer yanıltıcıdır. Çünkü video filmlerin satış olasılığı olan %75, %66 güven değerinden büyüktür. Aslında bilgisayar oyunlarının satışı, birinin satışı diğerinin satış olasılığını düşürdüğü için olumsuz bir birliktelik oluşturmuştur. Bu noktada korelasyon hesaplaması için kullanabileceğimiz lift değerini hesaplırsak;

$$lift(oyun \Rightarrow video) = \frac{güven(oyun \Rightarrow video)}{destek(video)} = \frac{0,66}{0,75} = 0,88$$

sonucu bulunacaktır. Lift değeri 1'den küçük çıktığından bilgisayar oyunu ve video film arasında negatif korelasyon olduğu ifade edilebilmektedir.

Sonuç olarak birliktelik kural analizinin gelişimi sürecinde birçok çalışma ele alınmış, teknikler geliştirilmiş ve şu anki temeller meydana getirilmiştir. Bunlar arasında fonksiyonel bağımlılık keşfi (Mannila, Raiha 1987), güçlü kurallar (Agrawal, Srikant 1994), sınıflandırma kuralları (Han, Cai ve Cercone 1993; Quinlan 1992), kümeleme (Fisher 1987), disk tabanlı, büyük işlemsel veri setlerinde birliktelik kuralları madenciliği için verimli metotlar (Agrawal, Imelinski ve Swami 1993; Agrawal, Srikant 1994; Agrawal, Srikant 1995; Mannila, Toivonen ve Verkamo 1994; Park, Chen ve Yu 1995) gibi teknikler bunlardan bazılarıdır. Birliktelik kurallarının keşfi veri madenciliğinin önemli bir sınıfıdır ve verideki

nitelikler arasındaki dikkate değer ilişkilerin deşifre edilmesini amaçlamaktadır (Houtsma, Swami 1995; Quinlan 1992).

Birliktelik kurallarının belirlenmesi için sık tekrarlanan öğelerin bulunması, bulunan bu öğelerin önceden belirlenen minimum destek sayısı kadar tekrarlanmış olması gerekir. Daha sonra tekrarlanan bu öğelerden güçlü birliktelik kuralları oluşturulur. Oluşturulan bu kurallar minimum güven ve minimum destek değerlerini karşılamalıdır (Arabacı 2007: 36).

2.2.1. Güven (Confidence) ve Destek (Support)

Destek ve güven değerleri, sırasıyla keşfedilen kuralların yararlılığını (kullanışlılığını) ve kesinliğini (doğruluğunu) ifade eder.

Güven ve destek değerlerinin örnek bir formülü şu şekildedir:

$$A \Rightarrow B \text{ [destek} = \% 2, \text{güven} = \% 60] \quad (2.1)$$

$(A \Rightarrow B)$ güveni aşağıdaki gibi hesaplanır:

$$\Rightarrow \text{güven} (A \Rightarrow B) = (\text{A ve B'nin bulunduğu satır sayısı}) / (\text{A'nın bulunduğu satır sayısı}) \quad (2.2)$$

Güven değerinin %60 olduğu (2.2) den çıkan sonuç; A ürünü satın alanların %60'ı B ürününü de almışlardır. Güven değerinin %100 olması demek A ürünün alan her kişi B ürünün de almıştır anlamına gelir ve böyle kurallara kesin kural adı verilir.

$(A \Rightarrow B)$ desteği ise şu şekildedir:

$$\text{destek} (A \Rightarrow B) = (\text{A ve B'nin bulunduğu satır sayısı}) / (\text{toplam satır sayısı}) \quad (2.3)$$

Destek değeri %2 olan (2.1) den çıkan sonuç; Satılan tüm satışların %2'sinde A ve B birlikte bulunmaktadır.

$$\text{Destek} - \text{support} = \frac{(\text{Kola, Cips, Su})}{\text{Toplam hareket}} = \frac{2}{5} = 0,4$$

$$\text{Güven} - \text{confidence} = \frac{(\text{Kola, Cips, Su})}{(\text{Kola, Cips})} = \frac{2}{4} = 0,5$$

Bu eşitliklerden de anlaşılacağı gibi, $\{\text{Kola, Cips}\} \Rightarrow \text{Su}$ kuralı %40 destek, %50 güven ölçülerine sahiptir

Örnek verilecek olursa market sepet analizi yöntemi farklı raf dizilimlerinin olabilmesine imkân tanır. Bir taktikte, birlikte sık olarak alınan ürünler raflarda yakın yerlere dizilebilirler. Bilgisayar alan müşterilerin çoğunluğu yazılım ürünlerini de alma eğilimdedir ise bu ürünlerin birbirine yakın yerlere konulması iki ürünün satış oranlarını da artırabilir. Diğer alternatif bir taktikte, bilgisayar ve yazılım ürünlerini markete ait bir rafın başlangıcına ve sonuna yerleştirmek, müşteriye kandırma yollarından birisi olabilir. Çünkü müşteri genelde raf boyunca başka ürünlere bakarak ilerler ve bunları satın alma olasılığı doğar (Han, Kamber 2000: 51).

2.3. Birliktelik Analizinde Kullanılan Algoritmalar

Agrawal, Imielinski and Swami tarafından 1993 yılında birliktelik kurallarının keşfi problemi ilk olarak ortaya atıldıktan sonra bu işlemlerin gerçekleştirilebilmesi için farklı algoritmalar geliştirilmiştir. Bu algoritmalarından bazılarını aşağıda değinilecektir. Bu algoritmalarından en ilke olanı AIS (Agrawal, Imielinski ve Swami 1993) ve en çok bilineni Apriori (Agrawal, Srikant 1994) algoritmasıdır.

2.3.1. AIS Algoritması

AIS algoritması, veri tabanındaki tüm yaygın ürün kümelerini oluşturmak için 1993 yılında Agrawal, Imielinski ve Swami tarafından geliştirilmiş ve yayınlanmış olan ilk algoritmadır. Algoritma karar destek sorgulamaları yapmak için veri tabanlarının fonksiyonlarını artırmaya odaklanmıştır. Veri tabanında bulunan ürün isimlerinin A'dan Z'ye sıralanması kısıtını taşır (Delibaş 2010: 26).

AIS algoritması veri tabanını birçok kez tarar ve her tarama sırasında tüm işlemleri okur. Yapılan ilk tarama da veri tabanındaki ürünleri, teker teker sayarak hangilerinin sık tekrarlanan ürünler olduğunu belirler. Bunlardan sık tekrar edenler aday ürün kümeleri olarak işaretlenirler. Bu işlem tamamlandıktan sonra, bir önceki taramada sık tekrar ettikleri belirlenen ürün kümeleriyle, o işlemin ürünleri arasındaki ortak ürün kümeleri tespit edilir. Tespit edilen bu ortak ürün kümeleri işlemde var olan diğer ürünlerle birleştirilerek yeni aday kümeler oluşturulur. Herhangi bir I ürün kümesi, bir işlemdeki ürünlerle birleşip aday kümelerden birini

oluşturabilmesi için, birleşeceği kümenin hem sık tekrar ediyor olması hem de harf sırası bakımından ürün kümesi üçündeki diğer tüm ürünlerden sonra geliyor olması gerekmektedir (Silahtaroglu 2008: 85).

AIS algoritması aday kümelerin belirlenmesi adımını gerçekleştirebilmek için bir budama tekniği kullanır. Budama tekniğinin temelinde, aday kümeler içindeki gereksiz kümelerin belirlenerek silinmesi vardır. Bu adım tamamlandıktan sonra, her aday kümesinin desteği hesaplanır. Hesaplanan destek seviyeleri minimum destek seviyesine eşit veya bu seviyeden yüksek çıkanlar, yaygın ürün kümesi olarak işaretlenir. Bir sonraki taramada bu yaygın işareti taşıyan kümeler, daha sonraki aday kümelerin belirlenmesinde kullanılır (Agrawal, Imielinski ve Swami 1993).

2.3.2. SETM Algoritması

SETM algoritması, 1995 yılında Houtsma ve Swami tarafından yaygın ürün kümelerinin hesaplanmasında SQL kullanılmasının temel alınması için geliştirilmiş algoritmadır. SETM algoritmasında yaygın nesne kümelerinin her üyesi \overline{L}_k , TID (transaction ID) birincil anahtar olmak üzere $\langle \text{TID}, \text{ürün kümesi} \rangle$ biçimindedir. Aynı şekilde aday kümelerinin her bir üyesi, \overline{C}_k da $\langle \text{TID}, \text{ürün kümesi} \rangle$ biçimindedir (Delibaş 2010: 27).

SETM algoritması da veri tabanı üzerinde birçok tarama yapar. Yapılan ilk taramada, ayrı ayrı her bir ürünün destek sayısını sayar ve veri tabanında hangilerinin büyük veya yaygın olduğunu bulur. Daha sonra, bir önceki taramada geniş olarak işaretlenen ürün kümelerini kullanarak aday kümeleri belirler. SETM algoritması farklı olarak aday kümelerle birlikte üzerinde çalışılan işlemlerin TID bilgisini de saklı tutar. Bundan sonra aday ürün kümeleri ürün ismine göre sıraya dizilir ve küçük nesne kümeleri silinir. Eğer veri tabanı TID numarasına göre sıralanmışsa, bir sonraki tarama esnasında herhangi bir işlemdeki yaygın ürün kümeleri L_k 'nin TID numarasına göre sıralanmasıyla elde edilir. Bu şekilde veri tabanı birkaç kez taranır. Artık farklı herhangi bir yaygın ürün kümesi bulunamadığında algoritma sonlandırılır (Silahtaroglu 2008: 85).

2.3.3. Apriori Algoritması

1994 yılında Agrawal ve diğeri tarafından geliştirilen Apriori algoritması, veri madenciliği tarihinde birliktelik kurallarının çıkarılması konusunda elde edilmiş büyük bir başarı ve birliktelik kuralları çıkarımında en çok bilinen algoritma olmuştur. Yaygın nesnelere önsel bilgilerini kullanmasından yani bilgileri bir önceki adımdan almasından dolayı algoritmanın ismi “önceki (prior)” anlamında aprioridir (Döşlü 2008: 34).

Yaygın ürün kümelerini keşfetmek için kullanılan algoritmalar eldeki tüm verileri birçok kez tararlar. Yapılan ilk taramada, tek tek öğelerin destek değeri hesaplanarak kullanıcı tarafından başlangıçta girilen minimum destek değeri ile karşılaştırılır ve her bir ürünün yaygın olup olmadığına bakılır. Daha sonraki her tarama bir önceki taramada yaygın olarak belirlenmiş ürünlerden başlar ve yaygın ürün kümeleri oluşturulur. Bu yaygın ürün kümelerine aday ürün kümeleri denir. Taramanın sonunda ise hangi aday ürün kümesinin gerçekten yaygın olduğuna bakılır. Daha önce de belirtildiği gibi bir ürün kümesinin yaygın olarak adlandırılabilmesi için o ürün kümesinin kullanıcı tarafından verilen minimum destek seviyesinin üzerinde bir destek seviyesine sahip olması gerekir. Bir sonraki taramada, yine bir önceki taramada yaygın olarak seçilen ürün kümelerinden başlanır ve veri tabanının sonuna kadar bu ürün kümelerinin destekleri hesaplanır. Bu işlem, başka yeni yaygın ürün kümeleri bulunamayana kadar sürer (Agrawal, Srikant 1994).

Apriori algoritması, daha önceden ortaya atılmış olan AIS ve SETM algoritmalarından temelde her bir taramada aday ürün kümelerinin sayılma ve bu aday kümelerinin üretilme şekliyle ayrılır. Hem AIS algoritmasında, hem de SETM algoritmasında, aday nesne kümeleri tarama esnasında veriler okunurken üretilir. Özellikle bir işlem okunduktan sonra, yaygın ürün kümelerinin bu işlemlerde olup olmadığına da bakılır. Yeni aday ürün kümelerinin üretilmesi ise işlemlerdeki diğer ürünlerle elde edilen yaygın ürün kümelerinin birleştirilmesiyle genişletilerek üretilir (Agrawal, Imielinski ve Swami 1993; Agrawal, Srikant 1994).

Tabi bu durum da gereksiz yere, aslında küçük ürün kümesi olan birçok aday ürün kümesinin sanki yaygın ürün kümesiymiş gibi üretilmesi ve sayılması sonucunu doğurur. Bu da algoritmanın zaman karmaşıklığını arttırır (Silahtaroglu 2008: 87).

Apriori algoritması ise aday ürünleri üretirken veri tabanındaki işlemleri hiç işin içine sokmadan, yalnızca bir önceki taramada yaygın olduğu tespit edilmiş ürün kümelerini kullanarak oluşturur. Apriori algoritmasının yaygın bir ürün kümesinin herhangi bir alt kümesinin de yaygın olacağı kabulüne dayandığından bahsedilmiştir. Böylece k adet üründen oluşmuş bir ürün kümesi, $k-1$ adet ürüne sahip yaygın ürün kümelerinin birleştirilmesi ve alt kümeleri yaygın olmayanların silinmesiyle elde edilebilir. Bu birleşme ve silme işlemi sonunda daha az sayıda aday ürün kümeleri oluşacaktır (Delibaş 2010: 28).

Apriori algoritması, bir hareket veya ürün kümesi içindeki ürünlerin alfabetik şekilde sıralandığını varsaymaktadır. F_k , k boyutlu sık görülen bir ürün kümesi ve C_k da onların adayları olsun. Apriori ilk önce veri tabanını tarar ve her ürünün sayısını artırarak ve onları gereken minimum destek değeri sağlanıncaya kadar toplayarak boyutu 1 olan sık görülen ürün kümelerini arar. Daha sonra aşağıdaki üç adımı tekrar ederek bütün sık görülen ürün kümelerini bulur (Wu, vd. 2008).

- k boyutlu sık görülen ürün kümelerinden, C_{k+1} adet, $k+1$ boyutlu aday sık görülen ürün kümesi üret.
- Veri tabanını tara ve sık görülen ürün kümelerinin her adayı için destek değerini hesapla.
- Gereken minimum destek seviyesi sağlanıncaya kadar bu ürün kümelerini F_{k+1} 'e ekle.

Apriori algoritması Şekil 2.1'de gösterilmiştir. 3. Satırdaki apriori-gen fonksiyonu aşağıdaki iki adımlık süreci uygulayarak F_k 'dan C_{k+1} üretir (Wu, vd. 2008).

Birleşme adımı: k boyutlu iki sık görülen, ilk $k-1$ elemana sahip P_k ve Q_k ürün kümelerini birleştirerek $k+1$ boyutlu sık görülen ürün kümelerinin başlangıç adayları R_{k+1} i belirle.

$$R_{K+1} = P_k \cup Q_k \{ürün_1, \dots, ürün_{k-1}, ürün_k, ürün_k\}$$

$$P_k = \{ürün_1, \dots, ürün_{k-1}, ürün_k, ürün_k\}$$

$$Q_k = \{ürün_1, \dots, ürün_{k-1}, ürün_k, ürün_k\}$$

Burada $ürün_1 < ürün_2 < \dots < ürün_k, ürün_k$ 'dir.

- **Budama adımı:** R_{K+1} 'deki k boyutlu tüm ürün kümelerinin sık görülen olup olmadığını kontrol et ve bu gerekliliği sağlamayanları R_{K+1} 'den çıkararak C_{K+1} 'i üret. Çünkü C_{K+1} 'in sık görülen olmayan k boyutlu hiçbir alt kümesi $k+1$ boyutlu sık görülen ürün kümesinin alt kümesi olamaz.

5. satırdaki alt küme fonksiyonu t hareketindeki sık görülen ürün kümelerinin tüm adaylarını belirler. Daha sonra Apriori veri tabanını tarayarak sadece bu yolla üretilen adayların frekanslarını hesaplar.

Apriori' nin sık görülen ürün kümelerinin maksimum boyutu k_{max} iken veri tabanını en fazla k_{max+1} kere tarar.

Apriori aday kümelerin boyutunu indirgemedeki iyi performans gösterir (Şekil 2.1). Bununla birlikte, çok fazla sık görülen nesne kümesi, büyük nesne kümeleri ya da minimum destek değerinin çok düşük olduğu durumlarda, çok fazla sayıda aday küme üretme ve çok sayıda aday ürün kümesini kontrol etmek için veri tabanını tekrar tekrar kontrol etme gibi sorunlarla karşı karşıya kalır.

```

Apriori Algoritması

 $F_1 = \{1 \text{ elemanlı sık görülen ürün kümeleri}\};$ 

For (k= 1;  $F_k \neq \phi$ ; k++) do begin

     $C_{k+1} = \text{apriori-gen}(F_k); // \text{Yeni adaylar}$ 

    for all işlemler t  $\in$  Database do begin

         $C'_t = \text{altküme}(C_{k+1}, t); // t \text{ içindeki adaylar}$ 

        For all adaylar c  $\in C'_t$  do

            c.count++;

        End

         $F_{k+1} = \{c \in C_{k+1} \mid \text{c.count} \geq \text{minimum support}\}$ 

    End

End

Answer  $\bigcup_k F_k$ ;

```

Şekil 2.1. Apriori Algoritması

Veri madenciliğinde kullanılan karar ağaçları, sınıflandırma kuralları ve kümeleme teknikleri gibi desen bulma algoritmalarının pek çoğu makine öğrenmesi araştırma topluluğunda geliştirilmiştir. Sık rastlanan desen ve birliktelik kuralı madenciliği bu geleneğin az sayıda istisnalarından biridir. Bu tekniğin girişi veri madenciliği araştırmalarını canlandırmış ve etkisi muazzam olmuştur. Algoritma oldukça basit ve uygulaması kolaydır. Veri madencilerinin ilk yapmaya çalıştıkları şey Apriori ve benzeri algoritmaları denemektir (Delibaş 2010: 30).

2.3.4. Apriori-TID Algoritması

Birliktelik kurallarında, algoritmalar destek değerlerini hesaplamak için her geçişte tüm veri tabanını tarar; ancak her geçişte veri tabanının tamamının taranması gereksizdir. Bu varsayıma dayanarak Agrawal ve Srikant 1994 yılında, AprioriTID algoritmasını sunmuştur.

Apriori-TID algoritması da taramaya başlamadan önce aday nesne kümelerini belirlemek için Apriori 'nin aday üretim fonksiyonunu kullanır. Apriori-TID

algoritmasının Apriori algoritmasından en büyük farkı ilk geçişten sonra destek seviyesini bulmak için veri tabanının taranmamasıdır. Bu iş için önceki geçişte kullanılan aday kümelerinin şifrelenmiş şekli olan C_k kullanılır. SETM algoritmasında da olduğu gibi C_k 'nin her elemanı $\langle TID, \{X_k\} \rangle$ biçimindedir. Burada X_k , TID numaralı işlemde bulunan potansiyel yaygın k-nesne kümesidir, $k=1$ iken C_1 veri tabanına benzer. Bununla beraber her nesne, nesne kümesi tarafından yer değiştirilir, $k>1$ iken C_k algoritmanın onuncu adımında olduğu gibi üretilir, t işlemdeki C_k bir elemanı $\langle TID, c \rangle$ şeklindedir. Burada c , t işlemdeki C_k ya ait bir aday elemanıdır. $\{c \in C_k \mid c\}$. Eğer bir işlemin, herhangi bir k nesne kümesi adayı yoksa bu durumda C_k 'nin bu işlem için herhangi bir girdisi, elemanı olmayacaktır. Daha doğru bir ifade ile bu işlemin TID numarasını taşıyıyor olacaktır. Böylece C_k 'daki girdi sayısı, özellikle bu k değerleri için, veri tabanındaki işlem sayısından daha küçük olabilir. Ayrıca yine büyük k değerleri için her girdi kendisine karşılık gelen işlemde daha küçük olabilir. Çünkü o işlemde çok az sayıda aday barınıyor olabilir. Fakat küçük k değerleri için bunun tersi olacaktır; yani girdiler kendilerine karşılık gelen işlemlerden daha büyük olabileceklerdir (Agrawal, Srikant 1994).

2.3.5. FP-Growth (Frequent Pattern Growth) Algoritması

FP-Growth algoritması, yaygın olarak görülen nesne kümelerini bulmak için kullanılan, sistem kaynaklarını verimli olarak kullanabilen ve büyük veri kümelerinde hızlı çalışabilen bir birliktelik kuralları algoritmasıdır. Önceki çoğu algoritmadan daha etkili bir biçimde çalışarak maliyeti azalttığı görülmüştür. Bu durumun en temel sebebi olarak, tüm veri tabanını daha yoğun ve daha küçük bir veri yapısı, sık desen ağacı (Frequent Pattern Tree), içinde tutuyor olması gösterilmektedir.

FP-Growth içinde tüm veri tabanı, apriori tabanlı algoritmalarından farklı olarak yalnızca iki kez taranır. İlk tarama tüm öğelerin destek değerinin hesaplanması için, ikinci tarama ise ağaç veri yapısının oluşturulması içindir. Yeni aday nesne kümesi üretimine ve her defasında veri tabanının taranmasına gerek kalmadığı için FP_Growth algoritması büyük veri tabanları için bir kazanımdır. Algoritmanın çalışma prensibi şu şekildedir: Veri tabanı bir kez taranarak nesnelere teker teker destek değerleri hesaplanır. Hesaplanan destek değerleri, algoritmaya

girdi olarak verilen destek eşik değerinden, büyük ve eşit olan nesnelere büyükten küçüğe sıralanarak bir liste içerisine dâhil edilmektedir. Bu eleme ile yaygın olmayan nesnelere ağaca eklenmesi engellenmiş olur. Benzer şekilde veri tabanında bulunan her bir işlem kaydı içerisindeki nesnelere destek değerine göre büyükten küçüğe doğru sıralanır. FP-Tree oluşturmak için ilk olarak “root” adında yeni bir düğüm oluşturulur. Daha sonra her bir hareket sıralı şekilde ağaç içerisine yerleştirilir. Bu aşama da şu şekilde gerçekleşir: İşlem içerisinde yer alan bir öge eğer ağaçta yoksa o öge için yeni bir düğüm oluşturulur ve destek değeri 1 yapılarak ögelerle beraber tutulur. Eğer o öge daha önce oluşturulmuşsa sadece o düğümün destek değeri 1 artırılır. Düğümler arasındaki ilişkiyi tutmak için de bir başlık tablosu (*header table*) tutularak, her düğümün başlangıç noktası işaretlenir ve ağaç içerisindeki aynı düğümler birbirine işaretçilerle bağlanır. Ağaç oluşturulduktan sonra üzerinde Fp-Growth algoritması çalıştırılır. Sıklığı en az olan nesneden başlanarak, içinde o nesnenin geçtiği yollar belirlenir. Her yol için de, o ögenin destek değeri o yolun destek değeri olarak atanır. O nesne için şartlı desen temelini (*conditional pattern base*) belirleyen bu yollar oluşturur. Her bir şartlı desen temelinden şartlı desen ağacı (*conditional pattern tree*) oluşturulur. Daha sonra bu şartlı desen ağacı üzerinde algoritma özyinelemeli olarak yeniden çalışır. Tablo içindeki her bir öge için bu süreç tekrarlanır ve böylece sık nesnelere kümesi belirlenir (Özdoğan, Abul ve Yazıcı 2009).

Şekil 2.2’de gösterilmekte olan FP-Growth algoritmasının genel yapısıdır. Algoritmada ilk olarak veri tabanındaki her bir nesnenin destek değerleri hesaplanmaktadır. Hesaplanan destek değerleri, algoritmaya girdi olarak verilen destek eşik değerinden, büyük ve eşit olan nesnelere büyükten küçüğe sıralanarak bir liste içerisine dâhil edilmektedir. Bu eleme ile yaygın olmayan nesnelere ağaca eklenmesi engellenmiş olur. Veri tabanındaki her bir işlem kaydı nesnelere destek değerlerine uygun sıralanarak ağaca sıkıştırılmış şekilde eklenir. Sıralama işlemi ile destek değeri daha büyük olan nesnelere köke daha yakın olması sağlanır. Sıkıştırma işlemi çok tekrar eden nesnelere ilk ekler olarak birleştirilmesi ile gerçekleştirilir. Bu metot arama işlemi maliyetini büyük ölçüde azaltır. Hareket kaydı içerisinde yer alan bir nesne ağaçta bulunmuyorsa o nesne için yeni bir düğüm oluşturulur ve destek değeri 1 olarak atanır. Fakat o nesne daha önce ağaçta

oluşturulmuş ise yalnızca o düğümün destek değeri 1 arttırılır. Nesnelerin ağaçtaki başlangıç noktaları bir başlık tablosu içerisinde saklanır. Bununla birlikte ağaç içerisinde aynı nesneyi içeren düğümler birbirine bağlanır. Ağaç oluşturma işlemi tamamlandıktan sonra üzerinde Growth algoritması çalıştırılır.

Algoritma FPGrowth(VT, mindestek)

Boş liste tanımla: F[];

foreach Hareket H_i in VT **do**

foreach Nesne n_j in H_i **do**

 F[n_j] ++;

End

End

Foreach Nesne n in F **do**

if F[n] < mindestek **then**

 N nesnesini F listesinden sil

End

End

Sırala F[];

FPtree ağaç yapısının kök düğümünü tanımla: kök;

Foreach Hareket H_i in VT **do**

H_i kaydını F listesine göre sırala;

 AgacaEkle (H_i , kök);

End

Foreach nesne n_i in N **do**

 Growth (kök, n_i , mindestek);

End

End

Şekil 2.2. FP-Growth Algoritmasının Genel Yapısı

Şekil 2.3'te Growth algoritmasının her bir nesne için çalıştırılan genel yapısı gösterilmektedir. İlk olarak içerisinde o nesnenin geçtiği yollar belirlenir. Belirlenen yolda eğer tek bir dal varsa yaygın nesneler kümesi, dalı oluşturan nesnelerin kombinasyonudur. Fakat birden fazla yol varsa, destek değeri o yoldaki minimum destek değeri olarak belirlenir. Daha sonra bu yollar o nesne için koşullu örüntü temelini (conditional pattern base) oluşturur. Her bir koşullu örüntü temelinden

koşullu örüntü ağacı (conditional pattern tree) oluşturulur. Daha sonra bu şartlı örüntü ağacı üzerinde algoritma özyinelemeli olarak yeniden çalıştırılır.

```
Algoritma Growth(dugum, n, mindestek)
  if dugum tek bir yol, Y, içeriyorsa then
    foreach düğümlerin_kombinasyonu dkin Y do
      oruntu o= dk  $\cup$  n
      destek = min (dk düğümlerinin destek değerleri )
      if o.destek > mindestek then
        Cikti (o);
      End
    End
  Else
    foreach  $a_i$  in dugum do
      oruntu o=  $a_i \cup n$ 
      destek = min ( $a_i$  düğümlerinin destek değerleri )
      if o.destek > mindestek then
        Cikti (o);
      End
      Nesne-koşullu örüntüleri oluştur;
      Nesne-koşullu FPtree ağacı oluştur;
      if FPtree  $\neq \phi$  then
        Growth (FPtree, o, mindestek);
      End
    End
  End
End
```

Şekil 2.3. Growth Algoritmasının Genel Yapısı

Fp-Growth algoritması tamamlandığında birlikte yoğun olarak görülen nesnelere kümesi belirlenmiş olur. Algoritma böl ve yönet yaklaşımına uygun olarak, ana görevin kendi içinde daha küçük görevlere ayrılmasına imkân vermektedir. Bununla birlikte oluşturulan FPtree veri yapısı asıl veri kümesinden daha büyük olmamaktadır (Birant, vd. 2010).

2.3.6. Tertius Algoritması

Peter A. Flach Nicolas Lachiche tarafından geliştirilen Tertius algoritması 7500 satırlık C kodundan oluşmaktadır. Tertius, k adet en çok onaylanmış hipotezin bulunmasında, uygun en iyi birinci değer aramasını kullanan bir makine öğrenme aracıdır. Bu durum kuralın yeniliği ve kullanılabilirliği ölçülerinin dengeye getirilmesini sağlamaktadır. Arama esnasındaki tekrarların önlenmesi için artıksız düzeltme operatörü kullanılmaktadır. Bölümlere ait verinin analizinde Wickens tarafından düzenlenen olasılık tablosu kullanılmaktadır. Gözlemlenen ve tahmin edilen değerler arasındaki ilişkinin ortaya çıkarılmasında Pearson X^2 istatistiksel yöntemi kullanılmaktadır. Tertius diğer tümcesel kural bulma yöntemlerine yenilik olarak doğrulama değeri kavramını getirmiştir. Tertius algoritmasında doğrulama değeri buluşsal olarak ölçülmektedir. Bulunan örneklerin budama işlemi karşıt örneklerin incelenmesi ile yapılır (Peter, Nicolas 2001).

2.3.7. Diğer Algoritmalar

Birliktelik kuralları konusunda bugüne kadar pek çok algoritma geliştirilmiştir. Bu algoritmalarından bazıları birbirinden bağımsız ve habersiz olan, eş zamanlı olarak farklı gruplar tarafından da geliştirilmiştir. Literatürdeki diğer algoritmalar önceki birliktelik kuralları algoritmalarına benzer mantık yürütmektedirler. Bu algoritmalarından bazılarının kısa bir özetine aşağıda yer verilmiştir (Silahtaroglu 2008: 90):

- Apriori-Hybrid algoritması (Agrawal, Srikant 1994).

Apriori ve Apriori_TID algoritmalarının ikisini beraber kullanan karma bir algoritmadır. Algoritma, çalışması sırasında her tekrarda aynı algoritmanın kullanılması yerine Apriori algoritmasının avantajlı olduğu durumlarda Apriori algoritmasını, Apriori_TID algoritmasının avantajlı olduğu durumlarda da ise Apriori_TID algoritma yapısını kullanarak çalışan bir yapıya sahiptir. Apriori-Hybrid algoritması birçok durumda Apriori algoritmasından daha iyi sonuçlar vermektedir (Karabatak 2008: 34).

- OCD (Off-line Candidate Determination - Sıradışı Aday Belirleme) algoritması (Mannila, Toivonen ve Verkamo 1994).

Geniş nesne kümelerinin belirlenmesi için veri tabanından alınmış küçük örneklerin çok daha iyi sonuçlar verebileceği düşüncesine dayanır.

OCD algoritması, gereksiz aday kümelerini ortadan kaldırmak için önceki döngülerden elde edilen bilgilerin birleşik analiz sonuçlarını kullanır. Bir $Y \subseteq I$ altkümesinin seyrek olup olmadığını bilmemiz için, işlemlerin en az $(1-s)$ kadarının taranması gerekmektedir (Burada s destek eşiğidir). Bu yüzden s ' nin düşük değerleri için neredeyse bütün ilişkili veriler okunmalıdır. Açıkça ifade edilebilir ki eğer veri tabanı çok büyükse, veri üzerinden mümkün olduğunca az döngü yapılması önem taşımaktadır (Gürgen 2008: 46).

- DMA (Distributed Mining Algorithm - Dağıtılmış Madencilik Algoritması) (Cheung, vd. 1996).

DMA algoritması, iletişim mesajı azaltma tekniklerinin ve aday budama tekniğinin eklenmiş olduğu veri paralelleştirme modeli bazlı bir tekniktir. Bir nesne kümesinin hem tüm veri tabanında küresel olarak yaygın hem de bir veri tabanı bölmesinde yerel olarak yaygın yani yoğun olup olmadığına karar vermek ve sonrasında yoğun yaygın nesne kümelerinden adayları oluşturmak için her bir işlemcideki yaygın nesne kümelerinin kendi sayılarını kullanır (Döşlü 2008: 52).

- Partitioning Algoritması (Savasere 1995).

Veri tabanını küçük parçalara ayırarak, bellekte kullanılan alanı azaltıp daha hızlı sonuca ulaşmayı sağlayan bölümlenme tekniğidir.

Partition algoritması, veri tabanı üzerindeki en iyi tarama özelliğine sahip olup bilinen diğer algoritmalarından biraz farklıdır. Veri tabanını üzerinde en fazla iki tarama işlemi gerçekleştirerek veri tabanını n adet parçaya ayırmakta ve her parça için yoğun nesne kümelerini hesaplamaktadır. Daha sonra tüm parçaların yoğun nesne kümelerini

birleştirilmekte ve süper bir yoğun nesne kümesi hesaplamaktadır. Fakat tüm veri tabanında yoğun olan nesne kümelerinin, her parça setinde de yoğun olması gerekmektedir. Partition algoritmasının temel avantajı Girdi/Çıktı maliyetini azaltmak ve yoğun kümeleri sayarken ana hafızayı kullanmaktır (Karabatak 2008: 35).

- Sampling Algoritması (Toivonen 1996).

Sampling algoritması, en iyi durumda veri tabanı tarama sayısını bire, en kötü durumda ikiye düşürür. Öncelikle ana belleğe sığacak bir örnek veri tabanından alınır. Örnekteki büyük ürün kümelerinin kümesi, Apriori benzeri yüksek seviye bir algoritma mantığı kullanılarak bu örnekten bulunur (Gürgen 2008: 49).

- HPA (Hash-based Parallel Mining of Association Rules – Bağlantı Kurallarının Çırpı Temelli Paralel Madenciliği) (Takahiko, Kitsuregawa 1996).

HPA algoritması, hashing tekniği kullanarak adayları farklı işlemcilerle dağıtır. Örneğin; her bir işlemci, kendisine verilmiş olan adayları hesaplamak için aynı hash fonksiyonunu kullanır. Yapılan bu hesaplamada, işlemciler arasında bölünmüş veri tabanlarında hareket etmek yerine, aynı hash tekniğiyle hareketlerin alt nesne kümelerini hedef işlemcilerle doğru hareket ettirirler. Bu sayede, bir hareketteki alt nesne kümesi n adet işlemci yerine yalnızca bir işlemciye gider. HPA algoritması, çarpık isleme tarzı (skew handling) tekniğiyle daha da geliştirilmiştir (Döşlü 2008: 54).

- CARMA (Continuous Association Rule Mining Algorithm - Sürekli Bağlantı Kuralı Madenciliği) (Hidber 1999).

CARMA algoritması, kullanıcıya her taramadan sonra oluşan kuralları gösterip, minimum destek ve güven seviyelerini değiştirme olanağı verir.

Birliktelik kurallarının çevrimiçi olarak hesaplanması için önerilen bir algoritma olup veri tabanı üzerinden en çok iki tarama işlemi

gerçekleştirmektedir. Veri tabanının ilk taraması esnasında potansiyel yoğun nesne kümeleri ile ilgili bir bölüm oluşturulur ve istenirse kullanıcının minimum güven ve minimum destek parametrelerini değiştirme olanağı sağlar. Veri tabanının ikinci taramasında ise algoritma her bölümdeki nesne kümesinin destek değerini hesaplamakta ve yoğun olmayan nesne kümelerini temizlemektedir. Bölümler oluşturulurken her nesne kümesi ile ilgili alt ve üst sınır değerlerine göre yeni adaylar bu bölümlere eklenir veya çıkarılır. Ayrıca, sayma işlemi ikinci taramada yer almaktadır (Karabatak 2008: 37).

- CCPD (Common Candidate Partitioned Database - Ortak Aday Bölünmüş Veri tabanı) Algoritması (Zaki 1996).

CCPD algoritması, bazı güncellemelerle SGI Power Challenge paylaşımı bir bellek üzerinde CD'nin yürütülmesidir. Paylaşım bir bellek ortamında etkili bir şekilde aday sayılması ve üretilmesi için teknikler önermektedir. Yaygın nesne kümelerini ortak öneklere (çoğunlukla ilk nesne) göre eşdeğer sınıflar şeklinde gruplandırır ve her bir eşdeğer sınıftan adayları oluşturur. Yaygın nesne kümelerinin gruplandırılması aday sayısını azaltmazken aday üretim süresini düşürür. Her bir harekette adayların etkili bir şekilde sayılması için kısa-döngülü (short-circuited) bir altküme kontrol metodunu önermektedir (Döşlü 2008: 52).

- IDD (Intelligent Data Distribution) Algoritması (Han, Karypis ve Kumar 1997).

IDD algoritması, aday ürün kümelerindeki ilk ürüne dayanarak adayları işlemciler arasında bölümlere ayırır. Böylelikle aynı ilk ürüne sahip aday ürün kümeleri aynı bölümde bulunurlar. Bundan dolayı, her işlemci sadece işlemciye atanmış ürünlerden biriyle başlayan alt kümeleri kontrol etmesi gerekir (Gürgen 2008: 52).

- PAR (Parallel Association Rules - Paralel Bağlantı Kuralları) (Zaki 1997).

PAR algoritması, doğal yatay veri tabanı bölmelerinin tersine dikey veri tabanı bölmeleriyle çalışır. Veri tabanları için dikey organizasyonu kullanılarak nesne kümesindeki nesnelerin TID listelerinin kesişimi ile bir nesne kümesinin hesaplanması daha kolay yapılmaktadır. Buna rağmen, veri tabanı yatay olarak düzenlenmişse, dikey bölmelere dönüşümü gerekmektedir (Döşlü 2008: 54).

ÜÇÜNCÜ BÖLÜM

3. UYGULAMA

3.1. Uygulamanın Amacı

Marketlerde satılan ürünlerin birbirleri ile olan ilişkileri yani hangi ürünün hangi ürünle birlikte satın alınabileceği günümüzde market stratejilerinin belirlenmesinde oldukça önemli rol oynamaktadır. Bu maksatla market yöneticileri birliktelik analizi yapılan ürünlerin yerleşimini yan yana ya da birliktelikleri tespit edilen ürünlerin arasına görülmesi veya fark edilmesi istenen başkaca ürünler koyarak farklı yöntemlere göre yapabilmektedirler.

Çalışmanın amacı; veri madenciliği, veri tabanlarında bilgi keşfi ve birliktelik kurallarını detaylı olarak incelemek, veri madenciliğinde istatistiksel çözümlmeye ağırlık vererek bir pazar sepeti çözümlemesi uygulaması gerçekleştirip sonuçları değerlendirmektir.

Yapılan literatür incelemesinde;

Emel, Taşkın ve Tok (2005) yapmış oldukları bir çalışmada, birliktelik analizi için SuperQuery veri madenciliği yazılımı kullanmışlardır. En iyi satış performansına sahip müşteriler İstanbul'da, en kötü satış performansına sahip müşterilerin ise Erzurum'da olduğu görülmüştür.

Özçakır ve Çamurcu (2007) yapmış oldukları bir çalışmada, bir firmanın pastane satış verileri üzerinde veri madenciliği uygulamış ve birliktelik kuralları ile bir yazılım tasarlamışlardır. Tasarlanan yazılımda, Apriori algoritması kullanılmışlardır. Uygulanan Apriori algoritması ile farklı zaman dilimi, farklı satış lokasyonu girdi değerleri doğrultusunda birlikte satın alınan ürünler ile ilgili bağıntılar olduğu gözlemlenmiştir. Sonuç olarak, genelde aynı ürün grubuna ait ürünlerin, en sık birlikte satın alınan ürünler olduğunu görmüşlerdir.

Cavique (2007) yapmış olduğu çalışmasında birliktelik kuralı bulma algoritmalarının bazılarının büyük veri setlerinde çok uzun zamanda sonuçlanacağını

vurgulamaktadır. Çalışmalarında büyük veri setlerinde hızlı bir şekilde birliktelik kurallarını bulacak bir algoritma tanıtılmaktadır. Çalışmalarında önerdikleri yöntemin başarılı olduğu vurgulanmaktadır.

Timor ve Şimşek (2008) yaptıkları çalışmada tüketici davranışlarını modellemek için birliktelik kurallarını kullanmışlardır. Büyük bir perakende zincirine ait veriler kullanılmıştır. Çalışma sonunda elde edilen birliktelik kurallarının pazarlama ve raf diziminde faydalı olabileceği raporlanmaktadır.

Ay ve Çil (2008) yaptıkları çalışmada bir perakende zincirinde birliktelik kuralları oluşturmuşlardır. Çalışmada oluşturulan birliktelik kurallarından yola çıkmak suretiyle aynı zamanda yerleşke düzeni de önerilmektedir.

Erpolat (2012) yapmış olduğu bir çalışmada, Türkiye’de otomotiv sektöründe faaliyet gösteren bir yetkili servisin müşterilerine ait alış-veriş verileri, Apriori ve FP-Growth Algoritmaları kullanılarak analiz etmiştir. Sonuç olarak ele alınan ürün gruplarının birliktelik kurallarını belirlemede, Apriori Algoritması iki farklı çözüm verirken FP-Growth Algoritması ise tek bir çözüm sağladığı görülmüştür.

Aytaç ve Bilge (2013) yaptıkları çalışmada tele pazarlama verilerini kullanmış ve birliktelik kuralları oluşturmuşlardır. Çalışmalarında sayısal değişkenler, apriori algoritmasının işleyebilmesi için kategorik hale getirilmiştir. Çalışma sonunda kurallar oluşturulmuştur ve pazarlama stratejileri için değerli olan bazı bilgilerin ortaya çıktığı ifade edilmektedir.

Doğan, Erol ve Buldu (2014) yaptıkları çalışmada bir sigorta şirketinin müşterilerine ilişkin toplanan bir veri seti üzerinde birliktelik kurallarını uygulamışlardır. Çalışma sonunda birlikte satın alınan sigorta çeşitleri belirlenmiş ve pazarlama faaliyetlerine yön gösterebilecek bilgilerin ortaya çıktığı raporlanmaktadır.

Doğrul, Akay ve Kurt (2015) yaptıkları çalışmada trafik kazası verilerini analiz etmek için birliktelik kurallarından faydalanmışlardır. Çalışmada kazaların sıklıkla yaşandığı yerler ve zamanlar belirlenmiştir ve bu gibi durumlarda önlemlerin arttırılması ile birlikte kazaların önlenebileceği ifade edilmektedir.

Kurt Pehlivanoglu ve Duru (2015) çalışmalarında bir ortaokulda okuyan öğrencilerin sosyal medya kullanma alışkanlıkları üzerine veri toplamış ve sonuçları birliktelik kuralları yardımıyla analiz etmişlerdir. Çalışmanın sonucunda ortaya çıkan kurallar raporlanmaktadır.

3.2. Verilerin Hazırlanması

3.2.1. Verilerin Kapsamı

Parekende sektöründe hizmet veren bir firmaya ait 10000 faturalık veri seti kullanılmıştır. Her faturada, satın alınan ürünler yapılan bir alış-veriş işlemi temsil etmektedir. Fatura bir müşterinin sepetine giren ürünlerin göstergesidir. Faturada her satır topluluğu satın alınan ürünleri temsil etmektedir.

3.2.2. Veri Temizleme

Excel'de satır halinde alt alta bulunan verilerde toplam bulunan farklı ürün sayısı tespit edilerek, bu ürünler sütun halinde yazılmıştır. Daha sonra teker teker her faturayı temsil eden satırlarda bulunan ürünlerin karşılığı olarak 1, bulunmayan ürünlere 0 yazılması Python programında kod yazılarak sağlanmıştır.

3.2.3. Veri Seçme ve Dönüştürme

Bu aşamada, verilerden sağlıklı bir birliktelik sonucu almak için bir adet ürün bulunan faturalar elenerek 7678 satırlık, iki adet ürün bulunan faturaların elenerek 6035 satırlık, üç adet ürün bulunan faturaların elenerek 4736 satırlık, dört adet ürün bulunan elenerek 3731 satırlık veri setleri elde edilmiş ve arff formatına Python[<https://www.python.org/>] kullanılarak çevrilmiştir.

3.3. Kullanılan Algoritma ve Programlar

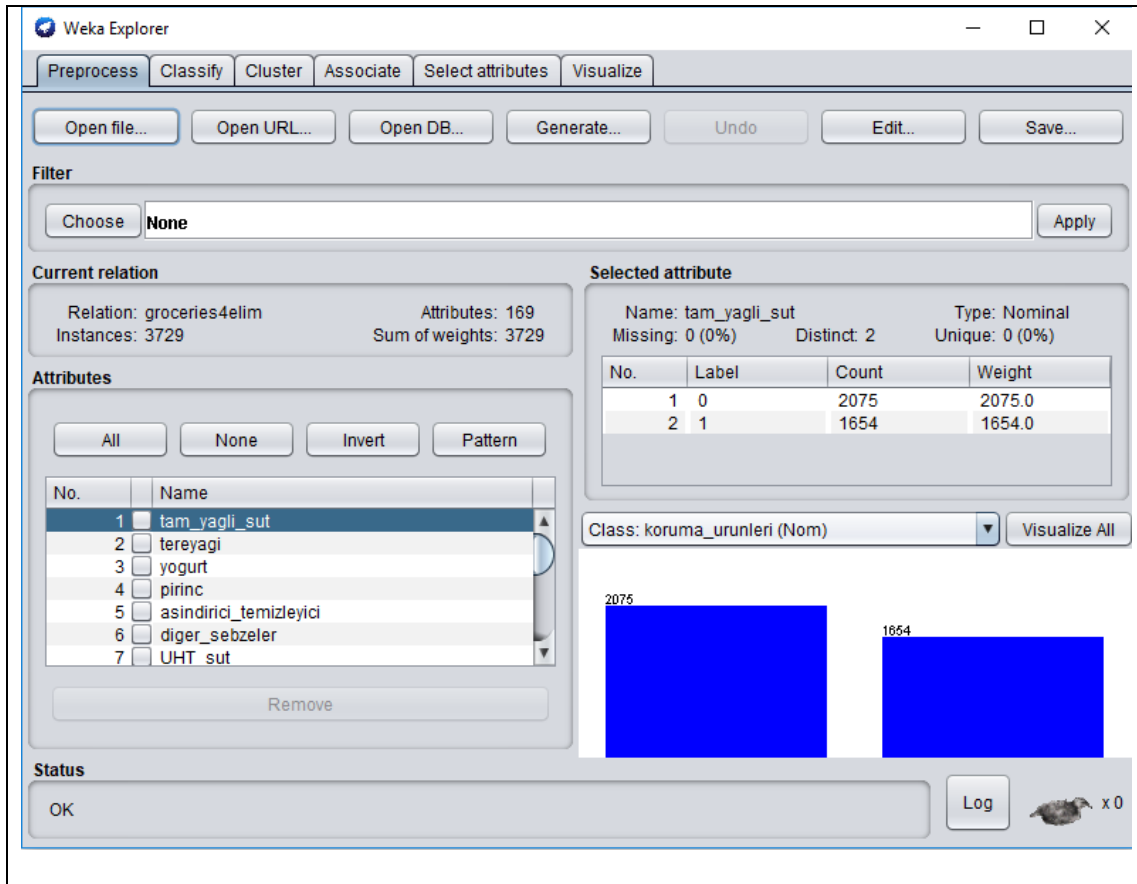
3.3.1. Weka Programı

Weka veri ön işleme araçlarının ve makine öğrenme algoritmalarının veri madenciliği görevleri için bir araya getirildiği, akademik çevrelerde sıklıkla kullanılan, açık kaynak kodlu bir yazılımdır. Yeni Zelanda'da bulunan Waikato Üniversitesinde Java yazılım dili ile geliştirilmiştir. Büyük veya dağıtık veri

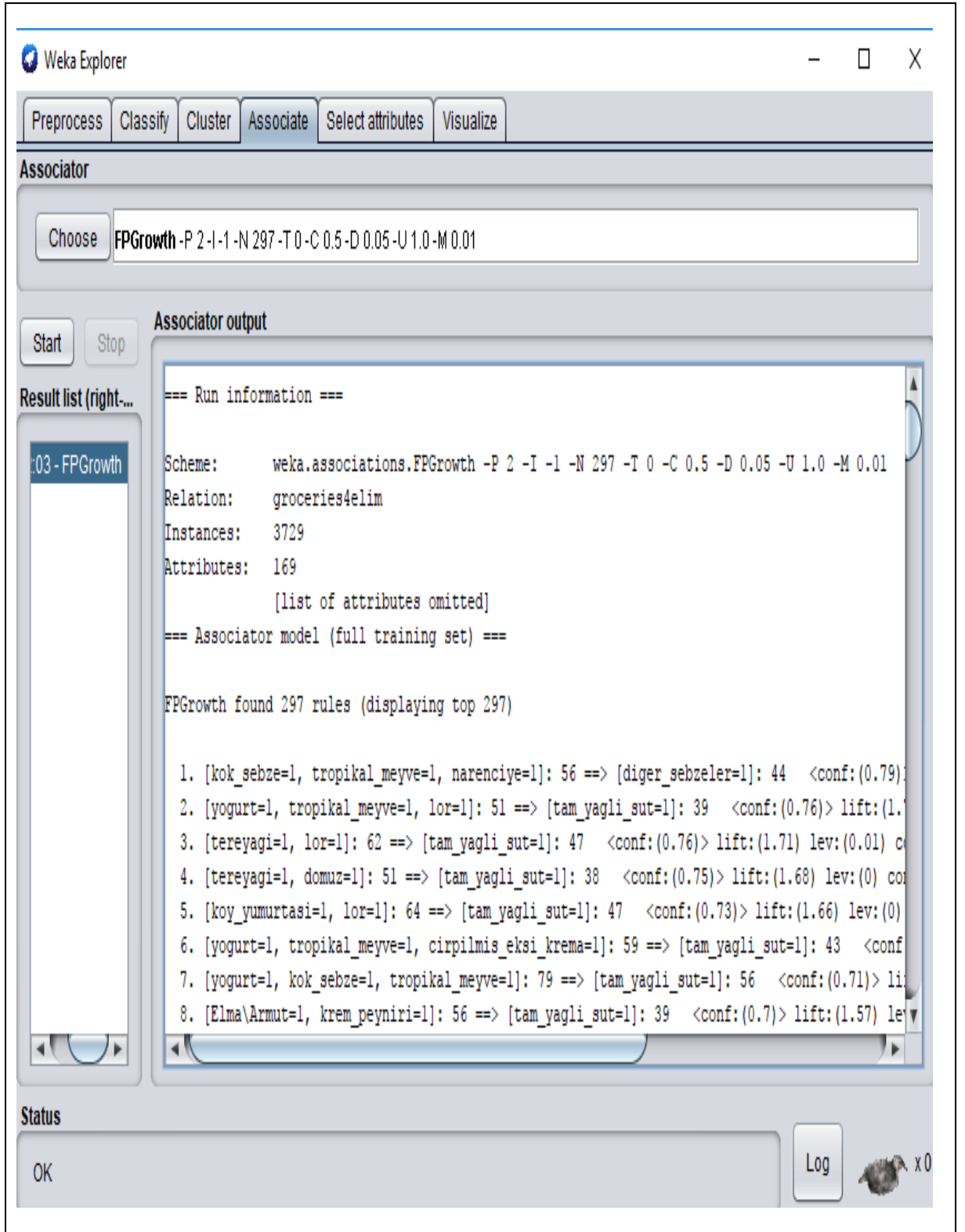
tabanlarında kullanılabilir. Weka ile veri ön işleme, sınıflama, birliktelik analizi, kümeleme, nitelik değerlerinin seçilmesi yapılabilmektedir. Arff, csv, C4.5 uzantılı dosyalar kullanılabilir (David, Peter 2007: 3).

Weka, bir makine öğrenme algoritması topluluğudur.

Algoritmalar doğrudan bir veri kümesine uygulanabilir veya kendi Java kodunuzdan çağrılabilir. Weka, veri ön işleme, sınıflandırma, gerileme, kümeleme, ilişki kuralları ve görselleştirme araçları içerir. Aynı zamanda yeni makine öğrenme planları geliştirmek için de uygundur.

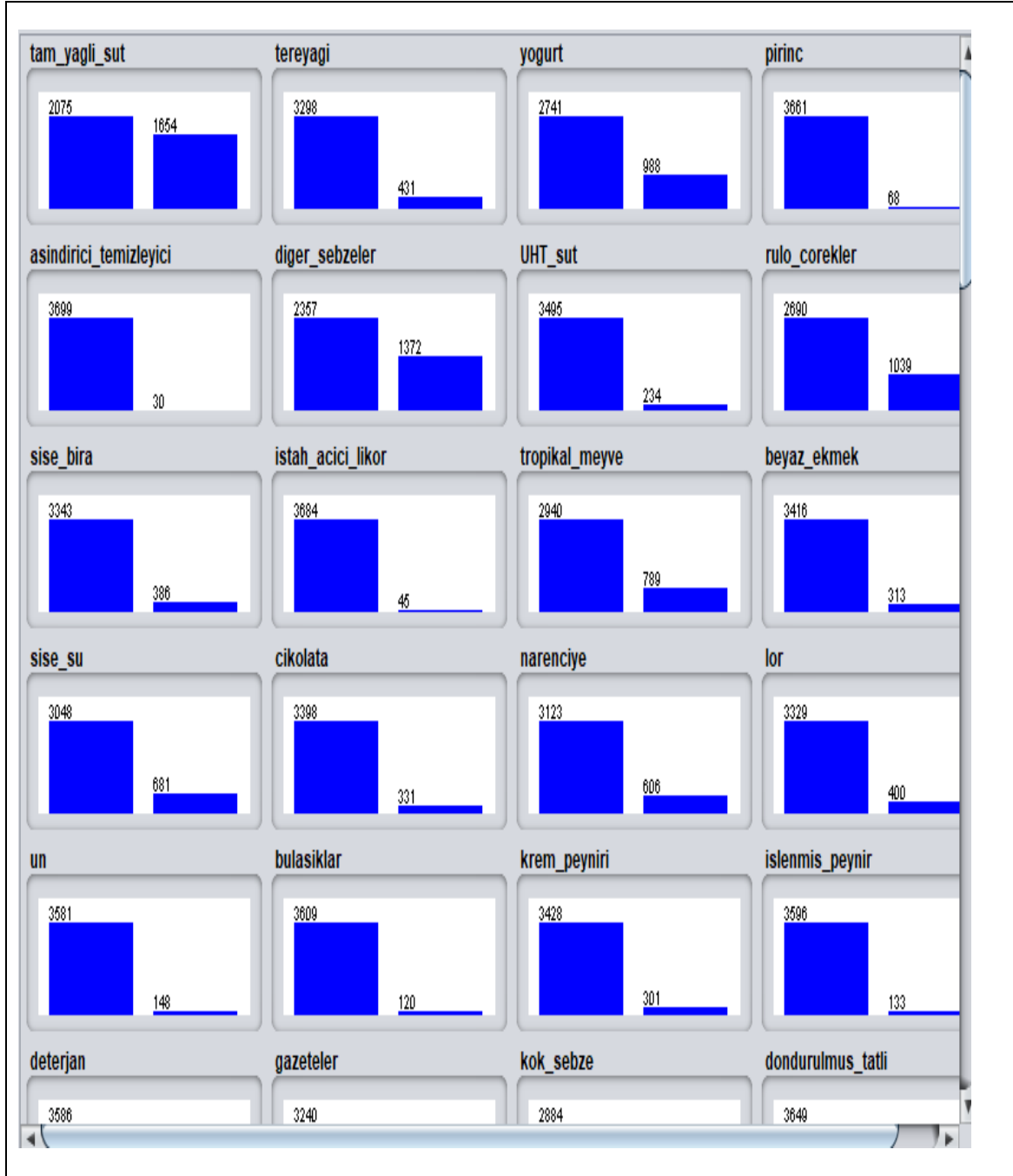


Şekil 3.1. Weka Programı Ara yüzü



Şekil 3.2. Birliktelik Analizi Paneli

3.4. Ürün Grupları Özellikleri Analizi



Şekil 3.3. Değişkenler ve Değerlerinin Gösterimi

1. [kok_sebzelerler=1, tropikal_meyve=1, narenciye=1]: 56 ==> [diger_sebzeler=1]: 44 <conf:(0.79)> lift:(2.14) lev:(0.01) conv:(2.72)
2. [yogurt=1, tropikal_meyve=1, lor_peyniri_peyniri=1]: 51 ==> [tam_yagli_sut=1]: 39 <conf:(0.76)> lift:(1.72) lev:(0) conv:(2.18)
3. [tereyagi=1, lor_peyniri_pemiri=1]: 62 ==> [tam_yagli_sut=1]: 47 <conf:(0.76)> lift:(1.71) lev:(0.01) conv:(2.16)
4. [tereyagi=1, domuz_eti=1]: 51 ==> [tam_yagli_sut=1]: 38 <conf:(0.75)> lift:(1.68) lev:(0) conv:(2.03)
5. [koy_yumurtasi=1, lor_peyniri_peyniri=1]: 64 ==> [tam_yagli_sut=1]: 47 <conf:(0.73)> lift:(1.66) lev:(0) conv:(1.98)
6. [yogurt=1, tropikal_meyve=1, cirpilmis_eksi_elma=1]: 59 ==> [tam_yagli_sut=1]: 43 <conf:(0.73)> lift:(1.64) lev:(0) conv:(1.93)
7. [yogurt=1, kok_sebzelerler=1, tropikal_meyve=1]: 79 ==> [tam_yagli_sut=1]: 56 <conf:(0.71)> lift:(1.6) lev:(0.01) conv:(1.83)
8. [Elma\Armut=1, krem_peyniri=1]: 56 ==> [tam_yagli_sut=1]: 39 <conf:(0.7)> lift:(1.57) lev:(0) conv:(1.73)
9. [diger_sebzeler=1, cirpilmis_eksi_elma=1, tereyagi=1]: 57 ==> [tam_yagli_sut=1]: 39 <conf:(0.68)> lift:(1.54) lev:(0) conv:(1.67)
10. [diger_sebzeler=1, yogurt=1, tereyagi=1]: 63 ==> [tam_yagli_sut=1]: 43 <conf:(0.68)> lift:(1.54) lev:(0) conv:(1.67)
11. [cirpilmis_eksi_elma=1, tereyagi=1]: 97 ==> [tam_yagli_sut=1]: 66 <conf:(0.68)> lift:(1.53) lev:(0.01) conv:(1.69)
12. [diger_sebzeler=1, kok_sebzelerler=1, Elma\Armut=1]: 78 ==> [tam_yagli_sut=1]: 53 <conf:(0.68)> lift:(1.53) lev:(0) conv:(1.67)
13. [yogurt=1, sert_peynir=1]: 61 ==> [tam_yagli_sut=1]: 41 <conf:(0.67)> lift:(1.52) lev:(0) conv:(1.62)
14. [tropikal_meyve=1, lor_peyniri_peyniri=1]: 96 ==> [tam_yagli_sut=1]: 64 <conf:(0.67)> lift:(1.5) lev:(0.01) conv:(1.62)
15. [koy_yumurtasi=1, margarin=1]: 77 ==> [tam_yagli_sut=1]: 51 <conf:(0.66)> lift:(1.49) lev:(0) conv:(1.59)
16. [kok_sebzelerler=1, tereyagi=1]: 121 ==> [tam_yagli_sut=1]: 80 <conf:(0.66)> lift:(1.49) lev:(0.01) conv:(1.6)
17. [koy_yumurtasi=1, tavuk=1]: 59 ==> [tam_yagli_sut=1]: 39 <conf:(0.66)> lift:(1.49) lev:(0) conv:(1.56)
18. [yogurt=1, tereyagi=1]: 138 ==> [tam_yagli_sut=1]: 91 <conf:(0.66)> lift:(1.49) lev:(0.01) conv:(1.6)
19. [kok_sebzelerler=1, sivi_yag=1]: 67 ==> [tam_yagli_sut=1]: 44 <conf:(0.66)> lift:(1.48) lev:(0) conv:(1.55)
20. [kok_sebzeler=1, hamburger_koftesi=1]: 58 ==> [tam_yagli_sut=1]: 38 <conf:(0.66)> lift:(1.48) lev:(0) conv:(1.54)

Şekil 3.4. FP-Growth Algoritması Birliktelik Analiz Sonucu

21. [Elma\Armut=1, cirpilmis_eksi_elma=1]: 87 ==> [tam_yagli_sut=1]: 57 <conf:(0.66)> lift:(1.48) lev:(0) conv:(1.56)
22. [diger_sebzeler=1, yogurt=1, narenciye=1]: 72 ==> [tam_yagli_sut=1]: 47 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.54)
23. [rulo_corekler=1, yogurt=1, kok_sebzeler=1]: 69 ==> [tam_yagli_sut=1]: 45 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.54)
24. [koy_yumurtasi=1, tereyagi=1]: 89 ==> [tam_yagli_sut=1]: 58 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.55)
25. [diger_sebzeler=1, kok_sebzeler=1, tereyagi=1]: 63 ==> [tam_yagli_sut=1]: 41 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.52)
26. [pirinc=1]: 68 ==> [tam_yagli_sut=1]: 44 <conf:(0.65)> lift:(1.46) lev:(0) conv:(1.51)
27. [diger_sebzeler=1, kok_sebzeler=1, dondurulmus_sebzeler=1]: 59 ==> [tam_yagli_sut=1]: 38 <conf:(0.64)> lift:(1.45) lev:(0) conv:(1.49)
28. [tam_yagli_sut=1, kok_sebzeler=1, narenciye=1]: 87 ==> [diger_sebzeler=1]: 56 <conf:(0.64)> lift:(1.75) lev:(0.01) conv:(1.72)
29. [Elma\Armut=1, tereyagi=1]: 69 ==> [tam_yagli_sut=1]: 44 <conf:(0.64)> lift:(1.44) lev:(0) conv:(1.48)
30. [diger_sebzeler=1, yogurt=1, tropikal_meyve=1]: 118 ==> [tam_yagli_sut=1]: 75 <conf:(0.64)> lift:(1.43) lev:(0.01) conv:(1.49)
31. [Elma\Armut=1, lor_peyniri=1]: 74 ==> [tam_yagli_sut=1]: 47 <conf:(0.64)> lift:(1.43) lev:(0) conv:(1.47)
32. [diger_sebzeler=1, kok_sebzeler=1, koy_yumurtasi=1]: 71 ==> [tam_yagli_sut=1]: 45 <conf:(0.63)> lift:(1.43) lev:(0) conv:(1.46)
33. [diger_sebzeler=1, yogurt=1, Elma\Armut=1]: 79 ==> [tam_yagli_sut=1]: 50 <conf:(0.63)> lift:(1.43) lev:(0) conv:(1.47)
34. [diger_sebzeler=1, yogurt=1, meyve_sebze_suyu=1]: 79 ==> [tam_yagli_sut=1]: 50 <conf:(0.63)> lift:(1.43) lev:(0) conv:(1.47)
35. [tropikal_meyve=1, koy_yumurtasi=1]: 106 ==> [tam_yagli_sut=1]: 67 <conf:(0.63)> lift:(1.43) lev:(0.01) conv:(1.47)
36. [yogurt=1, lor_peyniri=1]: 152 ==> [tam_yagli_sut=1]: 96 <conf:(0.63)> lift:(1.42) lev:(0.01) conv:(1.48)
37. [Elma\Armut=1, koy_yumurtasi=1]: 81 ==> [tam_yagli_sut=1]: 51 <conf:(0.63)> lift:(1.42) lev:(0) conv:(1.45)
38. [tropikal_meyve=1, sigir_eti=1]: 70 ==> [tam_yagli_sut=1]: 44 <conf:(0.63)> lift:(1.42) lev:(0) conv:(1.44)
39. [tropikal_meyve=1, sigir_eti=1]: 70 ==> [diger_sebzeler=1]: 44 <conf:(0.63)> lift:(1.71) lev:(0) conv:(1.64)
40. [tropikal_meyve=1, tereyagi=1]: 94 ==> [tam_yagli_sut=1]: 59 <conf:(0.63)> lift:(1.42) lev:(0) conv:(1.45)
41. [kok_sebzeler=1, yesillikler=1]: 64 ==> [tam_yagli_sut=1]: 40 <conf:(0.63)> lift:(1.41) lev:(0) conv:(1.42)

Şekil 3.4. FP-Growth Algoritması Birliktelik Analiz Sonucu (Devamı)

42. [tropikal_meyve=1, hijyen_malzemeleri=1]: 64 ==> [tam_yagli_sut=1]: 40 <conf:(0.63)> lift:(1.41) lev:(0) conv:(1.42)
43. [kok_sebzeler=1, koy_yumurtasi=1]: 133 ==> [tam_yagli_sut=1]: 83 <conf:(0.62)> lift:(1.41) lev:(0.01) conv:(1.45)
44. [cirpilmis_eksi_elma=1, krem_peyniri=1]: 61 ==> [tam_yagli_sut=1]: 38 <conf:(0.62)> lift:(1.4) lev:(0) conv:(1.41)
45. [tam_yagli_sut=1, kok_sebzeler=1, dondurulmus_sebzeler=1]: 61 ==> [diger_sebzeler=1]: 38 <conf:(0.62)> lift:(1.69) lev:(0) conv:(1.61)
46. [diger_sebzeler=1, yogurt=1, kok_sebzeler=1]: 124 ==> [tam_yagli_sut=1]: 77 <conf:(0.62)> lift:(1.4) lev:(0.01) conv:(1.44)
47. [Elma\Armut=1, cirpilmis_eksi_elma=1]: 87 ==> [diger_sebzeler=1]: 54 <conf:(0.62)> lift:(1.69) lev:(0.01) conv:(1.62)
48. [yogurt=1, kok_sebzeler=1, tropikal_meyve=1]: 79 ==> [diger_sebzeler=1]: 49 <conf:(0.62)> lift:(1.69) lev:(0.01) conv:(1.61)
49. [cirpilmis_eksi_elma=1, koy_yumurtasi=1]: 89 ==> [tam_yagli_sut=1]: 55 <conf:(0.62)> lift:(1.39) lev:(0) conv:(1.41)
50. [kok_sebzeler=1, soganlar=1]: 86 ==> [diger_sebzeler=1]: 53 <conf:(0.62)> lift:(1.68) lev:(0.01) conv:(1.6)
51. [diger_sebzeler=1, yogurt=1, unlu_mamuller=1]: 65 ==> [tam_yagli_sut=1]: 40 <conf:(0.62)> lift:(1.39) lev:(0) conv:(1.39)
52. [diger_sebzeler=1, kok_sebzeler=1, cirpilmis_eksi_elma=1]: 83 ==> [tam_yagli_sut=1]: 51 <conf:(0.61)> lift:(1.39) lev:(0) conv:(1.4)
53. [diger_sebzeler=1, kabartma_tozu=1]: 70 ==> [tam_yagli_sut=1]: 43 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.39)
54. [rulo_corekler=1, lor_peyniri=1]: 93 ==> [tam_yagli_sut=1]: 57 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.4)
55. [koy_yumurtasi=1, esmer_ekmek=1]: 62 ==> [tam_yagli_sut=1]: 38 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.38)
56. [sise_su=1, tereyagi=1]: 85 ==> [tam_yagli_sut=1]: 52 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.39)
57. [yogurt=1, hamburger_koftesi=1]: 64 ==> [tam_yagli_sut=1]: 39 <conf:(0.61)> lift:(1.37) lev:(0) conv:(1.37)
58. [cirpilmis_eksi_elma=1, margarin=1]: 64 ==> [tam_yagli_sut=1]: 39 <conf:(0.61)> lift:(1.37) lev:(0) conv:(1.37)
59. [tam_yagli_sut=1, kok_sebzeler=1, meyve_sebze_suyu=1]: 64 ==> [diger_sebzeler=1]: 39 <conf:(0.61)> lift:(1.66) lev:(0) conv:(1.56)
60. [tam_yagli_sut=1, tropikal_meyve=1, lor_peyniri=1]: 64 ==> [yogurt=1]: 39 <conf:(0.61)> lift:(2.3) lev:(0.01) conv:(1.81)

Şekil 3.4. FP-Growth Algoritması Birliktelik Analiz Sonucu (Devamı)

Diğer kurallar ekte sunulmuştur.

Bu veriler doğrultusunda;

Üretilen birinci kurala göre; Köklü Sebzeler, Tropikal Meyveler ve Narenciye ürünlerini alan 56 müşterinin 44'ü Diğer Sebzeler ürünlerini de almıştır. Köklü Sebzeler, Tropikal Meyveler ve Narenciye ürünlerini alan müşterilerin % 79 olasılıkla Diğer Sebzeler ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 2.14, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 2.72 olarak bulunmuştur.

Üretilen ikinci kurala göre; Yoğurt, Tropikal Meyveler ve Lor Peyniri ürünlerini alan 51 müşterinin 39'u Tam Yağlı Süt ürünlerini de almıştır. Yoğurt, Tropikal Meyveler ve Lor Peyniri ürünlerini alan müşterilerin % 76 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.72, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 2.18 olarak bulunmuştur.

Üretilen üçüncü kurala göre; Tereyağı ve Lor Peyniri ürünlerini alan 62 müşterinin 47'si Tam Yağlı Süt ürünlerini de almıştır. Tereyağı ve Lor Peyniri ürünlerini alan müşterilerin % 76 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.71, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 2.16 olarak bulunmuştur.

Üretilen dördüncü kurala göre; Tereyağı ve Domuz Eti ürünlerini alan 51 müşterinin 38'i Tam Yağlı Süt ürünlerini de almıştır. Tereyağı ve Domuz Eti ürünlerini alan müşterilerin % 75 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.68, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 2.03 olarak bulunmuştur.

Üretilen beşinci kurala göre; Köy Yumurtası ve Lor Peyniri ürünlerini alan 64 müşterinin 47'si Tam Yağlı Süt ürünlerini de almıştır. Köy Yumurtası ve Lor Peyniri ürünlerini alan müşterilerin % 73 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.66, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.98 olarak bulunmuştur.

Üretilen altıncı kurala göre; Yoğurt, Tropikal Meyve ve Çırpılmış Ekşi Elma ürünlerini alan 59 müşterinin 43'ü Tam Yağlı Süt ürünlerini de almıştır. Yoğurt, Tropikal Meyve ve Çırpılmış Ekşi Elma ürünlerini alan müşterilerin % 73 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.64,

kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.93 olarak bulunmuştur.

Üretilen yedinci kurala göre; Yoğurt, Kök Sebzeler ve Tropikal Meyve ürünlerini alan 79 müşterinin 56'sı Tam Yağlı Süt ürünlerini almıştır. Yoğurt, Kök Sebzeler ve Tropikal Meyve ürünlerini alan müşterilerin % 71 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.60, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 1.83 olarak bulunmuştur.

Üretilen sekizinci kurala göre; Elma/Armut ve Krem Peynir ürünlerini alan 56 müşterinin 39'u Tam Yağlı Süt ürünlerini almıştır. Elma/Armut ve Krem Peynir ürünlerini alan müşterilerin % 70 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.57, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.73 olarak bulunmuştur.

Üretilen dokuzuncu kurala göre; Diğer Sebzeler, Çırpılmış Ekşi Elma ve Tereyağı ürünlerini alan 57 müşterinin 39'u Tam Yağlı Süt ürünlerini almıştır. Diğer Sebzeler, Çırpılmış Ekşi Elma ve Tereyağı ürünlerini alan müşterilerin % 68 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.54, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.67 olarak bulunmuştur.

Üretilen onuncu kurala göre; Diğer Sebzeler, Yoğurt ve Tereyağı ürünlerini alan 63 müşterinin 43'ü Tam Yağlı Süt ürünlerini almıştır. Diğer Sebzeler, Yoğurt ve Tereyağı ürünlerini alan müşterilerin % 68 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.54, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.67 olarak bulunmuştur.

Üretilen on birinci kurala göre; Çırpılmış Ekşi Elma ve Tereyağı ürünlerini alan 97 müşterinin 66'sı Tam Yağlı Süt ürünlerini almıştır. Çırpılmış Ekşi Elma ve Tereyağı ürünlerini alan müşterilerin % 68 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.68, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 1.69 olarak bulunmuştur.

Üretilen on ikinci kurala göre; Diğer Sebzeler, Kök Sebzeler ve Elma/Armut ürünlerini alan 78 müşterinin 53'ü Tam Yağlı Süt ürünlerini almıştır. Diğer Sebzeler, Kök Sebzeler ve Elma/Armut ürünlerini alan müşterilerin % 68 olasılıkla

Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.53, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.67 olarak bulunmuştur.

Üretilen on üçüncü kurala göre; Yoğurt ve Sert Peynir ürünlerini alan 61 müşterinin 41'i Tam Yağlı Süt ürünlerini de almıştır. Yoğurt ve Sert Peynir ürünlerini alan müşterilerin % 67 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.52, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.62 olarak bulunmuştur.

Üretilen on dördüncü kurala göre; Tropikal Meyve ve Lor Peyniri ürünlerini alan 96 müşterinin 64'ü Tam Yağlı Süt ürünlerini de almıştır. Tropikal Meyve ve Lor Peyniri ürünlerini alan müşterilerin % 67 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.5, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 1.62 olarak bulunmuştur.

Üretilen on beşinci kurala göre; Köy Yumurtası ve Margarin ürünlerini alan 77 müşterinin 51'i Tam Yağlı Süt ürünlerini de almıştır. Köy Yumurtası ve Margarin ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.49, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.59 olarak bulunmuştur.

Üretilen on altıncı kurala göre; Kök Sebzeler ve Tereyağı ürünlerini alan 121 müşterinin 80'i Tam Yağlı Süt ürünlerini de almıştır. Kök Sebzeler ve Tereyağı ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.49, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 1.6 olarak bulunmuştur.

Üretilen on yedinci kurala göre; Köy Yumurtası ve Tavuk ürünlerini alan 59 müşterinin 39'u Tam Yağlı Süt ürünlerini de almıştır. Köy Yumurtası ve Tavuk ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.49, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.56 olarak bulunmuştur.

Üretilen on sekizinci kurala göre; Yoğurt ve Tereyağı ürünlerini alan 138 müşterinin 91'i Tam Yağlı Süt ürünlerini de almıştır. Yoğurt ve Tereyağı ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir.

Burada ilgi (lift) değeri 1.49, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 1.6 olarak bulunmuştur.

Üretilen on dokuzuncu kurala göre; Kök Sebzeler ve Sıvı Yağ ürünlerini alan 67 müşterinin 44'ü Tam Yağlı Süt ürünlerini almıştır. Kök Sebzeler ve Sıvı Yağ ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.48, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.55 olarak bulunmuştur.

Üretilen yirminci kurala göre; Kök Sebzeler ve Hamburger Köftesi ürünlerini alan 58 müşterinin 38'i Tam Yağlı Süt ürünlerini almıştır. Kök Sebzeler ve Hamburger Köftesi ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.48, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.54 olarak bulunmuştur.

Üretilen yirmi birinci kurala göre; Elma/Armut ve Çırpılmış Ekşi Elma ürünlerini alan 87 müşterinin 57'si Tam Yağlı Süt ürünlerini almıştır. Elma/Armut ve Çırpılmış Ekşi Elma ürünlerini alan müşterilerin % 66 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.48, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.56 olarak bulunmuştur.

Üretilen yirmi ikinci kurala göre; Diğer Sebzeler, Yoğurt ve Narenciye ürünlerini alan 72 müşterinin 47' si Tam Yağlı Süt ürünlerini almıştır. Diğer Sebzeler, Yoğurt ve Narenciye ürünlerini alan müşterilerin % 65 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.47, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.54 olarak bulunmuştur.

Üretilen yirmi üçüncü kurala göre; Rulo Çörekler, Yoğurt ve Kök Sebzeler ürünlerini alan 69 müşterinin 45'i Tam Yağlı Süt ürünlerini almıştır. Rulo Çörekler, Yoğurt ve Kök Sebzeler ürünlerini alan müşterilerin % 65 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.47, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.54 olarak bulunmuştur.

Üretilen yirmi dördüncü kurala göre; Köy Yumurtası ve Tereyağı ürünlerini alan 89 müşterinin 58'i Tam Yağlı Süt ürünlerini almıştır. Köy Yumurtası ve Tereyağı ürünlerini alan müşterilerin % 65 olasılıkla Tam Yağlı Süt ürünlerini de

aldığı söylenebilir. Burada ilgi (lift) değeri 1.47, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.55 olarak bulunmuştur.

Üretilen yirmi beşinci kurala göre; Diğer Sebzeler, Kök Sebzeler ve Tereyağı ürünlerini alan 63 müşterinin 41'i Tam Yağlı Süt ürünlerini de almıştır. Diğer Sebzeler, Kök Sebzeler ve Tereyağı ürünlerini alan müşterilerin % 65 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.47, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.52 olarak bulunmuştur.

Üretilen yirmi altıncı kurala göre; Pirinç ürününü alan 68 müşterinin 44'ü Tam Yağlı Süt ürünlerini de almıştır. Pirinç ürününü alan müşterilerin % 65 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.46, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.51 olarak bulunmuştur.

Üretilen yirmi yedinci kurala göre; Diğer Sebzeler, Kök Sebzeler ve Dondurulmuş Sebzeler ürünlerini alan 59 müşterinin 38'i Tam Yağlı Süt ürünlerini de almıştır. Diğer Sebzeler, Kök Sebzeler ve Dondurulmuş Sebzeler ürünlerini alan müşterilerin % 64 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.45, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.49 olarak bulunmuştur.

Üretilen yirmi sekizinci kurala göre; Tam Yağlı Süt, Kök Sebzeler ve Narenciye ürünlerini alan 87 müşterinin 56'sı Diğer Sebzeler ürünlerini de almıştır. Tam Yağlı Süt, Kök Sebzeler ve Narenciye ürünlerini alan müşterilerin % 64 olasılıkla Diğer Sebzeler ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.75, kaldıraç (leverage) değeri 0.01 ve kanaat (conviction) değeri ise 1.72 olarak bulunmuştur.

Üretilen yirmi dokuzuncu kurala göre; Elma/Armut ve Tereyağı ürünlerini alan 69 müşterinin 44'ü Tam Yağlı Süt ürünlerini de almıştır. Elma/Armut ve Tereyağı ürünlerini alan müşterilerin % 64 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.44, kaldıraç (leverage) değeri 0 ve kanaat (conviction) değeri ise 1.48 olarak bulunmuştur.

Üretilen otuzuncu kurala göre; Diğer Sebzeler, Yoğurt ve Tropikal Meyveler ürünlerini alan 118 müşterinin 75'i Tam Yağlı Süt ürünlerini de almıştır. Diğer

Sebzeler, Yoğurt ve Tropikal Meyveler ürünlerini alan müşterilerin % 64 olasılıkla Tam Yağlı Süt ürünlerini de aldığı söylenebilir. Burada ilgi (lift) değeri 1.43, kaldıraç (leverage) değeri 0,01 ve kanaat (conviction) değeri ise 1.49 olarak bulunmuştur.

Üretilen diğer kurallar için de, yukarıdaki kurallara yapılmış olan yorumlara benzer yorumlar her bir sonuç dikkate alınmak suretiyle yapılabilir.

SONUÇ

Son yıllarda veri tabanı teknolojisindeki gelişmeler, saklanan veri miktarlarında önemli bir artışa yol açmıştır. Bu artış veri madenciliğini şirketlerin stratejilerini belirlemeleri açısından önemli bir konu haline getirmiştir. Market veya mağaza verileri ile yapılan birliktelik analizi çalışmalarında müşterilerin ürün tercihlerini belirlemeye yönelik sonuçlar üzerinde durulmakta ve bu sonuçlara göre pazarlama stratejileri geliştirilmektedir.

Rekabetin yoğun olduğu ortamlarda yer alan işletmeler kendilerini diğer işletmelerden farklılaştırabilmek için bilgiyi etkili ve verimli bir biçimde kullanma yolunda veri madenciliğini tercih etmektedirler. Bilginin elde edilmesinde çok eski zamanlardan beri kullanılan yöntemlerin en önemlileri arasında olan istatistiksel yöntemler, yavaş yavaş hak ettikleri değere kavuşmaktadırlar. Sağlamış olduğu faydanın farkına varan iş dünyası istatistik uygulamalarını yeni baştan keşfetmeye başlamışlardır. Bu maksatla en yaygın kullanılan yöntemlerden bir tanesi veri madenciliğidir. Veri madenciliği, devasa veri tabanlarından, önceden keşfedilmemiş, kullanılabilir ve geçerli bilginin ortaya çıkarılması işlemi olarak ifade edilebilir. Veri madenciliğinin en popüler modellerinden birisi birliktelik kuralları madenciliğidir. Birliktelik kuralları madenciliği, ilişkisel veri tabanları, işlemsel veri tabanları ve diğer veri havuzları biçimleri gibi çeşitli veri tabanlarında bulunan veri kümelerinden, sık rastlanan örüntüleri, korelasyonları, ilişkilendirmeleri veya nedensel yapıları bulmak için kullanılan bir yöntemdir.

Verilerin ve veri işleme araçlarının gelişmesi sonucunda gerek ticari gerekse akademik alandaki çalışmalar hız kazanmış ve veri analizi birçok alandan daha önemli hale gelmiştir. Sayısal ortamda biriken verilerin artık kullanılmalarının gerekliliği fark edilmeye başlanmış ve geçmiş bilgiler değerlendirilmeye alınmıştır. Özellikle gelecekteki verilerin tahmin için kullanılan bu veriler üzerinde işlem yapmak üzere birçok bilim dalı bir araya gelerek çalışmalar gerçekleştirmişlerdir. Veri madenciliği, bu noktada çok disiplinli bir kavram olarak ortaya çıkmış ve hızla popülerlik kazanmıştır. Birçok alanda uygulanabilir olması ve ortaya koyduğu sonuçların ilgili faaliyet alanına katkılarının fark edilmesi de yaygınlaşmasında büyük rol oynamıştır.

Müşteriye sunulacak herhangi bir ürün-hizmet teklifinin müşteri tarafından kabul edilip edilmemesi, kişi hakkındaki bir demografik veri veya onun bir diğer tutum ve davranışıyla ilişkilendirilebilirse işletmeler için pazarlama faaliyetleri bakımından önemli bir bilgi birikimi elde edilmiş olur. Elde edilen bu bilgi birikimi, işletmelerin müşterileri hakkında toplamış oldukları veriler üzerinde birliktelik kuralları madenciliği uygulaması sayesinde kazanılır.

Bu çalışmada bir markete ait veriler kullanılarak birliktelik kuralları madenciliği yapılmış ve satılan ürünler arasında birliktelik kurallarının olduğu ortaya konmuştur. Bu sonuçlara göre; işletmenin müşterilerin birlikte satın almaya eğilimli oldukları ürünleri yakın raflara koyarak satışlarını artırıcı ürün planlaması yapılmasına imkân verirken aynı zamanda müşteriye zamandan tasarruf sağlayarak müşteri memnuniyetine katkı sağlamış olur. En az beş ürün alan müşterilerin faturaları dikkate alınarak analizler yapılmıştır. Mevcut verilerle yapılan birliktelik kuralları madenciliği sonucunda 297 adet kural üretilmiştir. En yüksek güven seviyesi en üstte olacak şekilde yapılan sıralamaya göre; en üstte “ Köklü Sebzeler, Tropikal Meyveler ve Narenciye “ ürünlerinin olduğu ortaya konmuştur.

KAYNAKÇA

- ADRIAANS Pieter, ZANTINGE Dolf, *Data Mining*, , Boston, MA, USA Addison Wesley Longman Publishing 1997.
- AGRAWAL Rakesh, SHAFER John C., “Parallel Mining of Association Rules”. *IEEE Transactions on Knowledge and Data Engineering*, S: (1996), s. (962-969).
- AGRAWAL Rakesh, IMIELINSKI Tomasz, SWAMI Arun, “Mining Association Rules Between Sets Of Items In Large Databases”, In Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD’93), Washington, DC, May 1993, s. (207–216).
- AGRAWAL Rakesh, SRIKANT Ramakrishnan, “Mining Sequential Patterns”, *International Conference Data Engineering*, Taipei, Taiwan 1995.
- AGRAWAL Rakesh, SRIKANT Ramakrishnan, “Fast algorithms for mining association rules”, In Proceedings of the 20th International Conference on Very Large Databases, San Francisco: Morgan Kaufmann, 1994, s. (487–499).
- AKPINAR Haldun, “Veritabanlarında Bilgi Keşfi ve Veri Madenciliği”, İstanbul Üniversitesi İşletme Fakültesi Dergisi, S: (2000), s. (1-22).
- ALPAYDIN Ethem, “Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri”, Bilişim 2000 Eğitim Semineri, 2000.
- ANAND Anuraç, “A Study And Comparison Of Data Clustering Techniques”, Faculty of The Graduate School of The University of Texas, Master Thesis, El Paso 2003.
- ARABACI Gültekin, *Veri Madenciliğinde Apriori, Tahminci Apriori Ve Tertius Algoritmalarının Weka Ve Yale Programları İle Karşılaştırılması Ve Bir Uygulama*, İstanbul Ticaret Üniversitesi Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, İstanbul 2007.

- ARSLAN Halil, *Sakarya Üniversitesi Web Sitesi Erişim Kayıtlarının Web Madenciliği İle Analizi*, Sakarya Üniversitesi Fen Bilimleri Enstitüsü / Elektronik-Bilgisayar Eğitimi Anabilim Dalı, Yüksek Lisans Tezi, Sakarya 2008.
- AYDOĞAN Fatih, *E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı Ve Gerçekleştirimi*, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara 2003.
- AYTAÇ Muhammed B., BİLGE Hasan Ş., *Tele Pazarlama Verilerinin Birliktelik Kurallarıyla ve Crisp-Dm Yöntemiyle Analiz Edilmesi*, Aksaray Üniversitesi İİBF Dergisi 2013/2, s. (25-40).
- BİRANT Derya, KUT Alp, VENTURA Medi, ALTINOK Hakan, ALTINOK Benal, ALTINOK Elvan, IHLAMUR Murat, “İş Zekâsı Çözümleri için Çok Boyutlu Birliktelik Kuralları Analizi”, Akademik Bilişim’10 - XII. Akademik Bilişim Konferansı Bildirileri 10 - 12 Şubat 2010, Muğla Üniversitesi S: 2010, s. (257-263).
- BRAMER Max, *Principles of Data Mining*, Undergraduate Topics in Computer Science ISSN 1863-7310, Springer-Verlag London Limited 2007.
- BOUCHON-MEUNIER Bernadette, MARSALA Christophe, “Learning Fuzzy Decision Rules”, The Handbooks of Fuzzy Sets Series book series S: 1999, s. (279-304).
- CAVIQUE Luis, *A Scalable Algorithm for the Market Basket Analysis*, Journal of Retailing and Consumer Services 2007/14, s. (400-407).
- CHEN Ming-Syan, HAN Jiawei ve YU Philip S., “Data Mining: An Overview From A Database Perspective”, *IEEE Transactions On Knowledge And Data Engineering*, S: (1996/6), s. (866-883).
- CHEUNG David W., HAN Jiawei, NG Vicent T., FU Ada W., FU Yongjian, “A Fast Distributed Algorithm for Mining Association Rules”, The fourth international conference on Parallel and distributed information systems, 1996, s. (31-42).

- COŞLU Eda, “Veri Madenciliği”, Akademik Bilişim 2013 – XV. Akademik Bilişim Konferansı Bildirileri 23-25 Ocak 2013 – Akdeniz Üniversitesi, Antalya, 2013, s. (615-619).
- DAVID Scuse, REUTEMANN Peter, *WEKA Experimenter Tutorial for Version 3-4*, Yeni Zelanda Waikato Üniversitesi 2006.
- DELİBAŞ Emre, *Birliktelik Analizi ile Reçeteli İlaç Satışları Üzerinde Bir Uygulama*, Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü İşletme Ana Bilim Dalı / Sayısal Yöntemler Bilim Dalı, Yüksek Lisans Tezi, Sivas 2010.
- DEMİREL Betül, *Veri Madenciliğinde Chaid Algoritmasının Sosyal Güvenlik Kurumu Veri Tabanına Uygulanması*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara 2010.
- AY Derya, ÇİL İbrahim, *Migros Türk. A.Ş. de Birliktelik Kurallarının Yerleşim Düzeni Planlamada Kullanılması*, Endüstri Mühendisliği Dergisi YA/EM 2008/2 Özel Sayısı, s. (14-29).
- DOĞAN Buket, EROL Bahar ve BULDU Ali, *Sigortacılık Sektöründe Müşteri İlişkileri Yönetimi için Birliktelik Kurallarının Kullanılması*, Marmara Fen Bilimleri Dergisi 2014/3, s. (105-114).
- DOĞRUL Gürdoğan, AKAY Diyar ve KURT Mustafa, *Trafik Kazalarının Birliktelik Kuralları ile Analizi*, Gazi Mühendislik Bilimleri Dergisi 2015/2, s. (265-284).
- DOLGUN Muhsin Ö., *Büyük Alışveriş Merkezleri için Veri Madenciliği Uygulamaları*, Hacettepe Üniversitesi İstatistik Anabilim Dalı, Yüksek Lisans Tezi, Ankara 2006.
- DÖŞLÜ Ayhan, *Veri Madenciliğinde Market Sepet Analizi ve Birliktelik Kurallarının Belirlenmesi*, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul 2008.
- EDELSTEIN Helbert A., *Introduction to Data Mining and Knowledge Discovery*, Maryland USA:Two Crows Corporation 1999.

- EMEL Gül G., TAŞKIN Çağatay ve TOK Arif, “Pazarlama Stratejilerinin Oluşturulmasında Bir Karar Destek Aracı: Birliktelik Kuralı Madenciliği ”, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi 2005/3, s. (30-59).
- ERGÜN Ertuğrul, *Ürün Kategorileri Arasındaki Satış İlişkisinin Birliktelik Kuralları Ve Kümeleme Analizi İle Belirlenmesi Ve Perakende Sektöründe Bir Uygulama*, Afyon Kocatepe Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı, Doktora Tezi, Afyon 2008.
- ERPOLAT Semra, “Otomobil Yetkili Servislerinde Birliktelik Kurallarının Belirlenmesinde Apriori ve FP-Growth Algoritmalarının Karşılaştırılması”, Anadolu Üniversitesi Sosyal Bilimler Dergisi 2012/2, s. (137-146)
- FAYYAD Usama M., PIATETSKY-SHAPIRO Gregory, SMYTH Padhraic ve UTHURUSAMY Ramasamy , *Advances in Knowledge Discovery And Data Mining*, Menlo Park California: AAAI, 1996.
- FISHER Douglas, “Improving Inference Through Conceptual Clustering”, *AAAI Conference*, Seattle, Washinton 1987, s. (461-465).
- GÜLCE Ali C., *Veri Madenciliğinde Apriori Algoritması Ve Apriori Algoritmasının Farklı Veri Kümelerinde Uygulanması*, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Edirne 2010.
- GÜRGEN Güneş, *Birliktelik Kuralları Sepet Analizi ve Uygulanması*, Marmara Üniversitesi Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, İstanbul 2008.
- GÜVEN Ahmet, BOZKURT Özgür Ö. ve KALIPSIZ Oya, “Veri Madenciliğinin Geleceği”, Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü Üniversite Dergisi, S: 2007, s: (1-40).
- GÜVENÇ Eda, *Yüksek Öğretimde Öğrenci Performansının Veri Madenciliği Teknikleri ile Belirlenmesi*, Boğaziçi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği ABD, Yüksek Lisans Tezi, İstanbul 2001.
- HAN Eui-Hong, KARYPIS George, KUMAR Vipin, “Scalable Parallel Data Mining for Association Rules”, 1997 ACM SIGMOD International Conference On Management Of Data 1997, s. (277-288).

- HAN Jiawei, KAMBER Micheline, *Data Mining Concepts And Techniques Second Edition*, San Francisco:Morgan Kaufman, 2006.
- HAN Jiawei, KAMBER Micheline, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers,2000.
- HAN Jiawei, CAI Yandong, CERCONE Nick, “Data-Driven Discovery Ofquantitative Rules İn Relational Databases”, IEEE Trans.Knowledge and Data Engineering, S: (1993), s. (29-40).
- HIDBER Christian, “Online Association Rule Mining”, ACM SIGMOD International Conference on Management of Data 1999, s. (145-156).
- HULTEN Geoff, SPENCER Laurie, DOMINGOS Pedro, “Mining Time-Changing Data Streams”, 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Fransisco CA: ACM Pres 2001, s. (97-106).
- HOUTSMA Maurice, SWAMI Arun, “Set-Oriented Mining for Association Rules İn relational Databases”, Proceedings of 11th International Conference 1995, s. (25-33).
- IAN Witten H., EIBE Frank, *Data Mining Practical Machine Learning Tools and Techniques, Second Edition*, Morgan Kaufmann Publishers ISBN: 0-12088407-0 2005.
- KARABATAK Murat, *Özellik Seçimi, Sınıflama ve Öngörü Uygulamalarına Yönelik Birliktelik Kuralı Çıkarımı ve Yazılım Geliştirilmesi*, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, Elazığ 2008.
- KUMAR Senthil A.V., WAHIDABANU R.S.D., “Data Mining Association Rules for Making Knowledgeable Decisions”, *Data Mining Applications for Empowering Knowledge Societies*, S: 2008 , s. (43-53).
- KURT PEHLİVANOĞLU Meltem, DURU Nevcihan, *Veri Madenciliği Teknikleri Kullanılarak Ortaokul Öğrencilerinin Sosyal Ağ Kullanım Analizi: Kocaeli İli Örneği*, Düzce Üniversitesi Bilim ve Teknoloji Dergisi 2015/3, s. (508-517).

- KOYUNCUGİL Ali S., *Bulanık Veri Madenciliği ve Sermaye Piyasalarına Uygulanması*, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, Ankara 2006.
- LEE Jun S., SIAU Keng, “A Review Of Data Mining Tecniques”, *Industrial Management & Data Systems*, S: (2001/1), s. (41-46).
- LU Hongjun, SETIONO Rudy And LIU Huan, “Effective Data Mining Using Neural Networks”, *IEEE Transactions On Knowledge And Data Engineering*, S: (1996/6),s. (957-961).
- MANNILA Heikki, RAIHA Kari-Jouko, “Dependency Inference”, The 1987 International Conference Very Large Data Bases, Brighton, England 1987, s. (155-158).
- MANNILA Heikki, TOIVONEN Hannu, VERKAMO Inkeri A., “Efficient Algorithms for Discovering Association Rules”, AAI’94 Workshop Knowldege Discovery in Databases, 1994, s. (181-192).
- ÖZKAN Yalçın, *Veri madenciliği yöntemleri*, Papatya Yayınları, İstanbul 2008.
- ÖZDOĞAN Gülistan Ö., ABUL Osman, YAZICI Ali, “Paralel Veri Madenciliği Algoritmaları”, BAŞARIM’09, 1. Ulusal Yüksek Başarım ve Grid Konferansı, ODTÜ-KKM, Ankara, 15-18 Nisan 2009, s. (1-26).
- PARK Jong S., CHEN Ming-Syan, YU Philip S., “Aneffective hash based algorithm for mining association rules”, ACM-SIGMOD International Conference Management of Data. San Jose,California 1995, s. (175-186).
- PETER Flach A., NİCOLAS Lachiche, “Confirmation-Guided Discovery of First-Order Rules with Tertius”, *Machine Learning*, C: (2001/42), s.(61-95).
- QUINLAN Ros J., *Programs for machine learning. Programs for Machine Learning by J. Ross Quinlan*, Morgan Kaufmann Publishers, Inc 1992.
- QUINLAN Ross J., “The Effect Of Noise On Concept Learning”, San Mateo, CA: Morgan Kauffmann Inc., C: 1986, s. (149-166).

- ÖZÇAKIR Cemal F., ÇAMURCU Yılmaz A., “Birliktelik Kuralı Yönetimi İçin Bir Veri Madenciliği Yazılımı Tasarımı ve Uyhulaması”, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi 2007/2, s. (21-37).
- ROKACH Lior, MAIMON Oded, *Data Mining With Decision Trees Theory And Applications*. Singapore: World Scientific, 2008.
- SAVASERE Ashok, OMIECINSKI Edward, NAVATHE Shamkant, “An Efficient Algorithm For Mining Association Rules In Large Databases”, 21st Int. Conf. Very Large Data Bases, , 1995, s. (432—444).
- Serkan, TOPALOĞLU Nurettin, YILMAZ Mithat, “Veri Madenciliği ve Türkiye’deki Uygulama Örnekleri” İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, S:2012, s. (1-23).
- SİLAHTAROĞLU Gökhan, “*Kavram ve algoritmalarıyla temel veri madenciliği*”, Papatya Yayınları, İstanbul, 2008.
- ŞEN Fatih, *Veri Madenciliği İle Birliktelik Kurallarının Bulunması*. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya 2008.
- TAKAHIKO Shintani, KITSUREGAWA Masaruta, “Hash Based Parallel Algorithms for Mining Association Rules”, 4th Int. Conf. on Parallel and Distributed Information Systems, 1996, s. (19-30).
- TİMOR Mehpere, EZERÇE Ayşegül ve GÜRSOY Tuğba U., “Müşteri Profili ve Alışveriş Davranışlarını Belirlemede Kümeleme ve Birliktelik Kuralları Analizi: Perakende Sektöründe Bir Uygulama”, İstanbul Üniversitesi İşletme Fakültesi Sayısal Yöntemler Anabilim Dalı Üniversite Dergisi için Makale, S: 2011 s. (128-147).
- TİMOR Mehpere, ŞİMŞEK Umman T., *Veri Madenciliğinde Sepet Analizi İle Tüketici Davranışı Modellemesi*, Yönetim 2008/19, s. (3-10).
- TANTUĞ Ahmet C., *Veri Madenciliği ve Demetleme*, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul 2002.

- TİRYAKİ Sevcan, *Lojistik Alanında Bir Veri Madenciliği Uygulaması*, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul 2006.
- TOIVONEN Hannu, “Sampling Large Databases for Association Rules”, *22nd International Conference on Very Large Data Bases (VLDB)*. 1996, s. 134-145
- VELICKOV Slavco, SOLOMATINE, Dimitri, *Predictive Data Mining: Practical Examples*, In: AI methods in Civil Engineering Applications, Cottbus 2000.
- WEBB Geoffrey I., “Association Rules”, *The Handbook Of data Mining*, S: (2003), s. (26-39).
- WU Xindong, KUMAR Vipin , QUINLAN Ross J., GHOSH Joydeep, YANG Qiang, MOTODA Hiroshi, MCLACHLAN Geoffrey J., NG Angus, LIU Bing, YU Philip S., ZHOU Zhi-Hua, STEINBACH Michael , HAND David J. ve STEINBERG Dan, “Top 10 algorithms in data mining”, *Knowledge And Information System*, S: (2008/1), s. (1-37).
- ZAKI Mohammed J., PARTHASARATHY Srinivasan, OGIHARA Mitsunori ve LI Wei, “New Algorithms For Fast Discovery Of Association Rules”, *Data Mining And Knowledge Discovery*, S: (1997/4), s. (343-373).
- ZAKI Mohammed J., “Parallel and Distributed Association Mining : A Survey”, *IEEE*, S: (1999), s. (14-25).

EKLER

Ek: 1 FP-Growth sonuçları

1. [kok_sebzelerler=1, tropikal_meyve=1, narenciye=1]: 56 ==> [diger_sebzeler=1]: 44 <conf:(0.79)> lift:(2.14) lev:(0.01) conv:(2.72)
2. [yogurt=1, tropikal_meyve=1, lor_peyniri_peyniri=1]: 51 ==> [tam_yagli_sut=1]: 39 <conf:(0.76)> lift:(1.72) lev:(0) conv:(2.18)
3. [tereyagi=1, lor_peyniri_peyniri=1]: 62 ==> [tam_yagli_sut=1]: 47 <conf:(0.76)> lift:(1.71) lev:(0.01) conv:(2.16)
4. [tereyagi=1, domuz_eti=1]: 51 ==> [tam_yagli_sut=1]: 38 <conf:(0.75)> lift:(1.68) lev:(0) conv:(2.03)
5. [koy_yumurtasi=1, lor_peyniri_peyniri=1]: 64 ==> [tam_yagli_sut=1]: 47 <conf:(0.73)> lift:(1.66) lev:(0) conv:(1.98)
6. [yogurt=1, tropikal_meyve=1, cirpilmis_eksi_elma=1]: 59 ==> [tam_yagli_sut=1]: 43 <conf:(0.73)> lift:(1.64) lev:(0) conv:(1.93)
7. [yogurt=1, kok_sebzelerler=1, tropikal_meyve=1]: 79 ==> [tam_yagli_sut=1]: 56 <conf:(0.71)> lift:(1.6) lev:(0.01) conv:(1.83)
8. [Elma\Armut=1, krem_peyniri=1]: 56 ==> [tam_yagli_sut=1]: 39 <conf:(0.7)> lift:(1.57) lev:(0) conv:(1.73)
9. [diger_sebzeler=1, cirpilmis_eksi_elma=1, tereyagi=1]: 57 ==> [tam_yagli_sut=1]: 39 <conf:(0.68)> lift:(1.54) lev:(0) conv:(1.67)
10. [diger_sebzeler=1, yogurt=1, tereyagi=1]: 63 ==> [tam_yagli_sut=1]: 43 <conf:(0.68)> lift:(1.54) lev:(0) conv:(1.67)
11. [cirpilmis_eksi_elma=1, tereyagi=1]: 97 ==> [tam_yagli_sut=1]: 66 <conf:(0.68)> lift:(1.53) lev:(0.01) conv:(1.69)
12. [diger_sebzeler=1, kok_sebzelerler=1, Elma\Armut=1]: 78 ==> [tam_yagli_sut=1]: 53 <conf:(0.68)> lift:(1.53) lev:(0) conv:(1.67)
13. [yogurt=1, sert_peynir=1]: 61 ==> [tam_yagli_sut=1]: 41 <conf:(0.67)> lift:(1.52) lev:(0) conv:(1.62)
14. [tropikal_meyve=1, lor_peyniri_peyniri=1]: 96 ==> [tam_yagli_sut=1]: 64 <conf:(0.67)> lift:(1.5) lev:(0.01) conv:(1.62)
15. [koy_yumurtasi=1, margarin=1]: 77 ==> [tam_yagli_sut=1]: 51 <conf:(0.66)> lift:(1.49) lev:(0) conv:(1.59)
16. [kok_sebzelerler=1, tereyagi=1]: 121 ==> [tam_yagli_sut=1]: 80 <conf:(0.66)> lift:(1.49) lev:(0.01) conv:(1.6)
17. [koy_yumurtasi=1, tavuk=1]: 59 ==> [tam_yagli_sut=1]: 39 <conf:(0.66)> lift:(1.49) lev:(0) conv:(1.56)
18. [yogurt=1, tereyagi=1]: 138 ==> [tam_yagli_sut=1]: 91 <conf:(0.66)> lift:(1.49) lev:(0.01) conv:(1.6)
19. [kok_sebzelerler=1, sivi_yag=1]: 67 ==> [tam_yagli_sut=1]: 44 <conf:(0.66)> lift:(1.48) lev:(0) conv:(1.55)
20. [kok_sebzeler=1, hamburger_koftesi=1]: 58 ==> [tam_yagli_sut=1]: 38 <conf:(0.66)> lift:(1.48) lev:(0) conv:(1.54)

21. [Elma\Armut=1, cirpilmis_eksi_elma=1]: 87 ==> [tam_yagli_sut=1]: 57 <conf:(0.66)> lift:(1.48) lev:(0) conv:(1.56)
22. [diger_sebzeler=1, yogurt=1, narenciye=1]: 72 ==> [tam_yagli_sut=1]: 47 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.54)
23. [rulo_corekler=1, yogurt=1, kok_sebzeler=1]: 69 ==> [tam_yagli_sut=1]: 45 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.54)
24. [koy_yumurtasi=1, tereyagi=1]: 89 ==> [tam_yagli_sut=1]: 58 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.55)
25. [diger_sebzeler=1, kok_sebzeler=1, tereyagi=1]: 63 ==> [tam_yagli_sut=1]: 41 <conf:(0.65)> lift:(1.47) lev:(0) conv:(1.52)
26. [pirinc=1]: 68 ==> [tam_yagli_sut=1]: 44 <conf:(0.65)> lift:(1.46) lev:(0) conv:(1.51)
27. [diger_sebzeler=1, kok_sebzeler=1, dondurulmus_sebzeler=1]: 59 ==> [tam_yagli_sut=1]: 38 <conf:(0.64)> lift:(1.45) lev:(0) conv:(1.49)
28. [tam_yagli_sut=1, kok_sebzeler=1, narenciye=1]: 87 ==> [diger_sebzeler=1]: 56 <conf:(0.64)> lift:(1.75) lev:(0.01) conv:(1.72)
29. [Elma\Armut=1, tereyagi=1]: 69 ==> [tam_yagli_sut=1]: 44 <conf:(0.64)> lift:(1.44) lev:(0) conv:(1.48)
30. [diger_sebzeler=1, yogurt=1, tropikal_meyve=1]: 118 ==> [tam_yagli_sut=1]: 75 <conf:(0.64)> lift:(1.43) lev:(0.01) conv:(1.49)
31. [Elma\Armut=1, lor_peyniri=1]: 74 ==> [tam_yagli_sut=1]: 47 <conf:(0.64)> lift:(1.43) lev:(0) conv:(1.47)
32. [diger_sebzeler=1, kok_sebzeler=1, koy_yumurtasi=1]: 71 ==> [tam_yagli_sut=1]: 45 <conf:(0.63)> lift:(1.43) lev:(0) conv:(1.46)
33. [diger_sebzeler=1, yogurt=1, Elma\Armut=1]: 79 ==> [tam_yagli_sut=1]: 50 <conf:(0.63)> lift:(1.43) lev:(0) conv:(1.47)
34. [diger_sebzeler=1, yogurt=1, meyve_sebze_suyu=1]: 79 ==> [tam_yagli_sut=1]: 50 <conf:(0.63)> lift:(1.43) lev:(0) conv:(1.47)
35. [tropikal_meyve=1, koy_yumurtasi=1]: 106 ==> [tam_yagli_sut=1]: 67 <conf:(0.63)> lift:(1.43) lev:(0.01) conv:(1.47)
36. [yogurt=1, lor_peyniri=1]: 152 ==> [tam_yagli_sut=1]: 96 <conf:(0.63)> lift:(1.42) lev:(0.01) conv:(1.48)
37. [Elma\Armut=1, koy_yumurtasi=1]: 81 ==> [tam_yagli_sut=1]: 51 <conf:(0.63)> lift:(1.42) lev:(0) conv:(1.45)
38. [tropikal_meyve=1, sigir_eti=1]: 70 ==> [tam_yagli_sut=1]: 44 <conf:(0.63)> lift:(1.42) lev:(0) conv:(1.44)
39. [tropikal_meyve=1, sigir_eti=1]: 70 ==> [diger_sebzeler=1]: 44 <conf:(0.63)> lift:(1.71) lev:(0) conv:(1.64)
40. [tropikal_meyve=1, tereyagi=1]: 94 ==> [tam_yagli_sut=1]: 59 <conf:(0.63)> lift:(1.42) lev:(0) conv:(1.45)
41. [kok_sebzeler=1, yesillikler=1]: 64 ==> [tam_yagli_sut=1]: 40 <conf:(0.63)> lift:(1.41) lev:(0) conv:(1.42)

42. [tropikal_meyve=1, hijyen_malzemeleri=1]: 64 ==> [tam_yagli_sut=1]: 40 <conf:(0.63)> lift:(1.41) lev:(0) conv:(1.42)
43. [kok_sebzeler=1, koy_yumurtasi=1]: 133 ==> [tam_yagli_sut=1]: 83 <conf:(0.62)> lift:(1.41) lev:(0.01) conv:(1.45)
44. [cirpilmis_eksi_elma=1, krem_peyniri=1]: 61 ==> [tam_yagli_sut=1]: 38 <conf:(0.62)> lift:(1.4) lev:(0) conv:(1.41)
45. [tam_yagli_sut=1, kok_sebzeler=1, dondurulmus_sebzeler=1]: 61 ==> [diger_sebzeler=1]: 38 <conf:(0.62)> lift:(1.69) lev:(0) conv:(1.61)
46. [diger_sebzeler=1, yogurt=1, kok_sebzeler=1]: 124 ==> [tam_yagli_sut=1]: 77 <conf:(0.62)> lift:(1.4) lev:(0.01) conv:(1.44)
47. [Elma\Armut=1, cirpilmis_eksi_elma=1]: 87 ==> [diger_sebzeler=1]: 54 <conf:(0.62)> lift:(1.69) lev:(0.01) conv:(1.62)
48. [yogurt=1, kok_sebzeler=1, tropikal_meyve=1]: 79 ==> [diger_sebzeler=1]: 49 <conf:(0.62)> lift:(1.69) lev:(0.01) conv:(1.61)
49. [cirpilmis_eksi_elma=1, koy_yumurtasi=1]: 89 ==> [tam_yagli_sut=1]: 55 <conf:(0.62)> lift:(1.39) lev:(0) conv:(1.41)
50. [kok_sebzeler=1, soganlar=1]: 86 ==> [diger_sebzeler=1]: 53 <conf:(0.62)> lift:(1.68) lev:(0.01) conv:(1.6)
51. [diger_sebzeler=1, yogurt=1, unlu_mamuller=1]: 65 ==> [tam_yagli_sut=1]: 40 <conf:(0.62)> lift:(1.39) lev:(0) conv:(1.39)
52. [diger_sebzeler=1, kok_sebzeler=1, cirpilmis_eksi_elma=1]: 83 ==> [tam_yagli_sut=1]: 51 <conf:(0.61)> lift:(1.39) lev:(0) conv:(1.4)
53. [diger_sebzeler=1, kabartma_tozu=1]: 70 ==> [tam_yagli_sut=1]: 43 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.39)
54. [rulo_corekler=1, lor_peyniri=1]: 93 ==> [tam_yagli_sut=1]: 57 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.4)
55. [koy_yumurtasi=1, esmer_ekmek=1]: 62 ==> [tam_yagli_sut=1]: 38 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.38)
56. [sise_su=1, tereyagi=1]: 85 ==> [tam_yagli_sut=1]: 52 <conf:(0.61)> lift:(1.38) lev:(0) conv:(1.39)
57. [yogurt=1, hamburger_koftesi=1]: 64 ==> [tam_yagli_sut=1]: 39 <conf:(0.61)> lift:(1.37) lev:(0) conv:(1.37)
58. [cirpilmis_eksi_elma=1, margarin=1]: 64 ==> [tam_yagli_sut=1]: 39 <conf:(0.61)> lift:(1.37) lev:(0) conv:(1.37)
59. [tam_yagli_sut=1, kok_sebzeler=1, meyve_sebze_suyu=1]: 64 ==> [diger_sebzeler=1]: 39 <conf:(0.61)> lift:(1.66) lev:(0) conv:(1.56)
60. [tam_yagli_sut=1, tropikal_meyve=1, lor_peyniri=1]: 64 ==> [yogurt=1]: 39 <conf:(0.61)> lift:(2.3) lev:(0.01) conv:(1.81)

61. [tam_yagli_sut=1, kok_sebzeler=1, Elma\Armut=1]: 87 ==> [diger_sebzeler=1]: 53 <conf:(0.61)> lift:(1.66) lev:(0.01) conv:(1.57)
62. [kok_sebzeler=1, narenciye=1]: 161 ==> [diger_sebzeler=1]: 98 <conf:(0.61)> lift:(1.65) lev:(0.01) conv:(1.59)
63. [yogurt=1, tavuk=1]: 79 ==> [diger_sebzeler=1]: 48 <conf:(0.61)> lift:(1.65) lev:(0.01) conv:(1.56)
64. [diger_sebzeler=1, tereyagi=1]: 186 ==> [tam_yagli_sut=1]: 113 <conf:(0.61)> lift:(1.37) lev:(0.01) conv:(1.4)
65. [kok_sebzeler=1, krem_peyniri=1]: 71 ==> [diger_sebzeler=1]: 43 <conf:(0.61)> lift:(1.65) lev:(0) conv:(1.55)
66. [tropikal_meyve=1, cirpilmis_eksi_elma=1]: 129 ==> [tam_yagli_sut=1]: 78 <conf:(0.6)> lift:(1.36) lev:(0.01) conv:(1.38)
67. [cirpilmis_eksi_elma=1, tavuk=1]: 68 ==> [tam_yagli_sut=1]: 41 <conf:(0.6)> lift:(1.36) lev:(0) conv:(1.35)
68. [diger_sebzeler=1, seker=1]: 100 ==> [tam_yagli_sut=1]: 60 <conf:(0.6)> lift:(1.35) lev:(0) conv:(1.36)
69. [yogurt=1, jambon=1]: 65 ==> [tam_yagli_sut=1]: 39 <conf:(0.6)> lift:(1.35) lev:(0) conv:(1.34)
70. [unlu_mamuller=1, lor_peyniri=1]: 70 ==> [tam_yagli_sut=1]: 42 <conf:(0.6)> lift:(1.35) lev:(0) conv:(1.34)
71. [diger_sebzeler=1, kok_sebzeler=1, meyve_sebze_suyu=1]: 65 ==> [tam_yagli_sut=1]: 39 <conf:(0.6)> lift:(1.35) lev:(0) conv:(1.34)
72. [kok_sebzeler=1, esmer_ekmek=1]: 92 ==> [tam_yagli_sut=1]: 55 <conf:(0.6)> lift:(1.35) lev:(0) conv:(1.35)
73. [meyve_sebze_suyu=1, koy_yumurtasi=1]: 77 ==> [tam_yagli_sut=1]: 46 <conf:(0.6)> lift:(1.35) lev:(0) conv:(1.34)
74. [kok_sebzeler=1, tropikal_meyve=1]: 201 ==> [diger_sebzeler=1]: 120 <conf:(0.6)> lift:(1.62) lev:(0.01) conv:(1.55)
75. [yogurt=1, sise_bira=1]: 84 ==> [tam_yagli_sut=1]: 50 <conf:(0.6)> lift:(1.34) lev:(0) conv:(1.34)
76. [narenciye=1, tereyagi=1]: 84 ==> [tam_yagli_sut=1]: 50 <conf:(0.6)> lift:(1.34) lev:(0) conv:(1.34)
77. [tam_yagli_sut=1, tropikal_meyve=1, Elma\Armut=1]: 79 ==> [diger_sebzeler=1]: 47 <conf:(0.59)> lift:(1.62) lev:(0) conv:(1.51)
78. [yogurt=1, yagli_sut=1]: 74 ==> [tam_yagli_sut=1]: 44 <conf:(0.59)> lift:(1.34) lev:(0) conv:(1.33)
79. [domuz=1, sigir_eti=1]: 69 ==> [diger_sebzeler=1]: 41 <conf:(0.59)> lift:(1.62) lev:(0) conv:(1.5)
80. [rulo_corekler=1, yogurt=1, kok_sebzeler=1]: 69 ==> [diger_sebzeler=1]: 41 <conf:(0.59)> lift:(1.62) lev:(0) conv:(1.5)
81. [kok_sebzeler=1, gofretler=1]: 64 ==> [tam_yagli_sut=1]: 38 <conf:(0.59)> lift:(1.34) lev:(0) conv:(1.32)
82. [tam_yagli_sut=1, soganlar=1]: 108 ==> [diger_sebzeler=1]: 64 <conf:(0.59)> lift:(1.61) lev:(0.01) conv:(1.52)

83. [narenciye=1, cirpilmis_eksi_elma=1]: 103 ==> [tam_yagli_sut=1]: 61 <conf:(0.59)> lift:(1.34) lev:(0) conv:(1.33)
84. [tam_yagli_sut=1, cirpilmis_eksi_elma=1, tereyagi=1]: 66 ==> [diger_sebzeler=1]: 39 <conf:(0.59)> lift:(1.61) lev:(0) conv:(1.49)
85. [kabartma_tozu=1]: 144 ==> [tam_yagli_sut=1]: 85 <conf:(0.59)> lift:(1.33) lev:(0.01) conv:(1.34)
86. [yogurt=1, kok_sebzeler=1]: 234 ==> [tam_yagli_sut=1]: 138 <conf:(0.59)> lift:(1.33) lev:(0.01) conv:(1.34)
87. [tam_yagli_sut=1, kok_sebzeler=1, tropikal_meyve=1]: 117 ==> [diger_sebzeler=1]: 69 <conf:(0.59)> lift:(1.6) lev:(0.01) conv:(1.51)
88. [cirpilmis_eksi_elma=1, lor_peyniri=1]: 95 ==> [tam_yagli_sut=1]: 56 <conf:(0.59)> lift:(1.33) lev:(0) conv:(1.32)
89. [tropikal_meyve=1, cirpilmis_eksi_elma=1]: 129 ==> [diger_sebzeler=1]: 76 <conf:(0.59)> lift:(1.6) lev:(0.01) conv:(1.51)
90. [unlu_mamuller=1, cirpilmis_eksi_elma=1]: 68 ==> [tam_yagli_sut=1]: 40 <conf:(0.59)> lift:(1.33) lev:(0) conv:(1.3)
91. [unlu_mamuller=1, cirpilmis_eksi_elma=1]: 68 ==> [diger_sebzeler=1]: 40 <conf:(0.59)> lift:(1.6) lev:(0) conv:(1.48)
92. [cirpilmis_eksi_elma=1, tereyagi=1]: 97 ==> [diger_sebzeler=1]: 57 <conf:(0.59)> lift:(1.6) lev:(0.01) conv:(1.5)
93. [diger_sebzeler=1, lor_peyniri=1]: 162 ==> [tam_yagli_sut=1]: 95 <conf:(0.59)> lift:(1.32) lev:(0.01) conv:(1.33)
94. [kok_sebzeler=1, tavuk=1]: 99 ==> [tam_yagli_sut=1]: 58 <conf:(0.59)> lift:(1.32) lev:(0) conv:(1.31)
95. [yogurt=1, soganlar=1]: 65 ==> [tam_yagli_sut=1]: 38 <conf:(0.58)> lift:(1.32) lev:(0) conv:(1.29)
96. [yogurt=1, soganlar=1]: 65 ==> [diger_sebzeler=1]: 38 <conf:(0.58)> lift:(1.59) lev:(0) conv:(1.47)
97. [yogurt=1, dilimlenmis_peynir=1]: 77 ==> [tam_yagli_sut=1]: 45 <conf:(0.58)> lift:(1.32) lev:(0) conv:(1.3)
98. [tropikal_meyve=1, dondurulmus_sebzeler=1]: 84 ==> [tam_yagli_sut=1]: 49 <conf:(0.58)> lift:(1.32) lev:(0) conv:(1.3)
99. [hindi=1]: 67 ==> [diger_sebzeler=1]: 39 <conf:(0.58)> lift:(1.58) lev:(0) conv:(1.46)
100. [kok_sebzeler=1, tropikal_meyve=1]: 201 ==> [tam_yagli_sut=1]: 117 <conf:(0.58)> lift:(1.31) lev:(0.01) conv:(1.32)
101. [rulo_corekler=1, margarin=1]: 129 ==> [tam_yagli_sut=1]: 75 <conf:(0.58)> lift:(1.31) lev:(0) conv:(1.31)
102. [sosis=1, cirpilmis_eksi_elma=1]: 86 ==> [tam_yagli_sut=1]: 50 <conf:(0.58)> lift:(1.31) lev:(0) conv:(1.29)
103. [Elma\Armut=1, tereyagi=1]: 69 ==> [diger_sebzeler=1]: 40 <conf:(0.58)> lift:(1.58) lev:(0) conv:(1.45)
104. [narenciye=1, koy_yumurtasi=1]: 95 ==> [tam_yagli_sut=1]: 55 <conf:(0.58)> lift:(1.31) lev:(0) conv:(1.29)

105. [diger_sebzeler=1, yogurt=1, cirpilmis_eksi_elma=1]: 95 ==> [tam_yagli_sut=1]: 55 <conf:(0.58)> lift:(1.31) lev:(0) conv:(1.29)
106. [diger_sebzeler=1, tropikal_meyve=1, cirpilmis_eksi_elma=1]: 76 ==> [tam_yagli_sut=1]: 44 <conf:(0.58)> lift:(1.31) lev:(0) conv:(1.28)
107. [rulo_corekler=1, yogurt=1, tropikal_meyve=1]: 83 ==> [tam_yagli_sut=1]: 48 <conf:(0.58)> lift:(1.3) lev:(0) conv:(1.28)
108. [yogurt=1, baharatli_alman_sosisi=1]: 104 ==> [tam_yagli_sut=1]: 60 <conf:(0.58)> lift:(1.3) lev:(0) conv:(1.29)
109. [diger_sebzeler=1, tropikal_meyve=1, narenciye=1]: 85 ==> [tam_yagli_sut=1]: 49 <conf:(0.58)> lift:(1.3) lev:(0) conv:(1.28)
110. [kok_sebzeler=1, Elma\Armut=1]: 151 ==> [tam_yagli_sut=1]: 87 <conf:(0.58)> lift:(1.3) lev:(0.01) conv:(1.29)
111. [kok_sebzeler=1, cirpilmis_eksi_elma=1]: 158 ==> [tam_yagli_sut=1]: 91 <conf:(0.58)> lift:(1.3) lev:(0.01) conv:(1.29)
112. [diger_sebzeler=1, kok_sebzeler=1, tropikal_meyve=1]: 120 ==> [tam_yagli_sut=1]: 69 <conf:(0.57)> lift:(1.3) lev:(0) conv:(1.28)
113. [rulo_corekler=1, tavuk=1]: 87 ==> [tam_yagli_sut=1]: 50 <conf:(0.57)> lift:(1.3) lev:(0) conv:(1.27)
114. [tropikal_meyve=1, tereyagi=1]: 94 ==> [diger_sebzeler=1]: 54 <conf:(0.57)> lift:(1.56) lev:(0.01) conv:(1.45)
115. [yogurt=1, hijyen_malzemeleri=1]: 68 ==> [tam_yagli_sut=1]: 39 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.26)
116. [cirpilmis_eksi_elma=1, peceteler=1]: 68 ==> [tam_yagli_sut=1]: 39 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.26)
117. [sosis=1, tereyagi=1]: 82 ==> [tam_yagli_sut=1]: 47 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.27)
118. [yogurt=1, cirpilmis_eksi_elma=1]: 185 ==> [tam_yagli_sut=1]: 106 <conf:(0.57)> lift:(1.29) lev:(0.01) conv:(1.29)
119. [rulo_corekler=1, domuz=1]: 103 ==> [tam_yagli_sut=1]: 59 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.27)
120. [diger_sebzeler=1, koy_yumurtasi=1]: 208 ==> [tam_yagli_sut=1]: 119 <conf:(0.57)> lift:(1.29) lev:(0.01) conv:(1.29)
121. [kok_sebzeler=1, unlu_mamuller=1]: 98 ==> [tam_yagli_sut=1]: 56 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.27)
122. [kok_sebzeler=1, unlu_mamuller=1]: 98 ==> [diger_sebzeler=1]: 56 <conf:(0.57)> lift:(1.55) lev:(0.01) conv:(1.44)
123. [diger_sebzeler=1, kok_sebzeler=1, narenciye=1]: 98 ==> [tam_yagli_sut=1]: 56 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.27)
124. [tereyagi=1]: 431 ==> [tam_yagli_sut=1]: 246 <conf:(0.57)> lift:(1.29) lev:(0.01) conv:(1.29)
125. [yogurt=1, krem_peyniri=1]: 114 ==> [tam_yagli_sut=1]: 65 <conf:(0.57)> lift:(1.29) lev:(0) conv:(1.27)

126. [tropikal_meyve=1, esmer_ekmek=1]: 95 ==> [tam_yagli_sut=1]: 54 <conf:(0.57)> lift:(1.28) lev:(0) conv:(1.26)
127. [yogurt=1, koy_yumurtasi=1]: 134 ==> [tam_yagli_sut=1]: 76 <conf:(0.57)> lift:(1.28) lev:(0) conv:(1.26)
128. [kok_sebzeler=1, sivi_yag=1]: 67 ==> [diger_sebzeler=1]: 38 <conf:(0.57)> lift:(1.54) lev:(0) conv:(1.41)
129. [yogurt=1, Elma\Armut=1]: 164 ==> [tam_yagli_sut=1]: 93 <conf:(0.57)> lift:(1.28) lev:(0.01) conv:(1.27)
130. [deterjan=1]: 143 ==> [tam_yagli_sut=1]: 81 <conf:(0.57)> lift:(1.28) lev:(0) conv:(1.26)
131. [kok_sebzeler=1, tavuk=1]: 99 ==> [diger_sebzeler=1]: 56 <conf:(0.57)> lift:(1.54) lev:(0.01) conv:(1.42)
132. [domuz=1, sigir_eti=1]: 69 ==> [tam_yagli_sut=1]: 39 <conf:(0.57)> lift:(1.27) lev:(0) conv:(1.24)
133. [diger_sebzeler=1, kok_sebzeler=1, sise_su=1]: 69 ==> [tam_yagli_sut=1]: 39 <conf:(0.57)> lift:(1.27) lev:(0) conv:(1.24)
134. [kok_sebzeler=1, margarin=1]: 101 ==> [diger_sebzeler=1]: 57 <conf:(0.56)> lift:(1.53) lev:(0.01) conv:(1.42)
135. [tam_yagli_sut=1, tropikal_meyve=1, cirpilmis_eksi_elma=1]: 78 ==> [diger_sebzeler=1]: 44<conf:(0.56)> lift:(1.53) lev:(0) conv:(1.41)
136. [tam_yagli_sut=1, tropikal_meyve=1, narenciye=1]: 87 ==> [diger_sebzeler=1]: 49 <conf:(0.56)> lift:(1.53) lev:(0) conv:(1.41)
137. [yogurt=1, uzun_omurlu_firin_urunu=1]: 80 ==> [tam_yagli_sut=1]: 45 <conf:(0.56)> lift:(1.27) lev:(0) conv:(1.24)
138. [saksi_bitkileri=1]: 105 ==> [tam_yagli_sut=1]: 59 <conf:(0.56)> lift:(1.27) lev:(0) conv:(1.24)
139. [tropikal_meyve=1, baharatli_alman_sosisi=1]: 89 ==> [tam_yagli_sut=1]: 50 <conf:(0.56)> lift:(1.27) lev:(0) conv:(1.24)
140. [diger_sebzeler=1, dondurulmus_sebzeler=1]: 164 ==> [tam_yagli_sut=1]: 92 <conf:(0.56)> lift:(1.26) lev:(0.01) conv:(1.25)
141. [tam_yagli_sut=1, kok_sebzeler=1, cirpilmis_eksi_elma=1]: 91 ==> [diger_sebzeler=1]: 51 <conf:(0.56)> lift:(1.52) lev:(0) conv:(1.4)
142. [kok_sebzeler=1, meyve_sebze_suyu=1]: 116 ==> [diger_sebzeler=1]: 65 <conf:(0.56)> lift:(1.52) lev:(0.01) conv:(1.41)
143. [lor_peyniri=1]: 400 ==> [tam_yagli_sut=1]: 224 <conf:(0.56)> lift:(1.26) lev:(0.01) conv:(1.26)
144. [kok_sebzeler=1, dondurulmus_sebzeler=1]: 109 ==> [tam_yagli_sut=1]: 61 <conf:(0.56)> lift:(1.26) lev:(0) conv:(1.24)
145. [pirinc=1]: 68 ==> [diger_sebzeler=1]: 38 <conf:(0.56)> lift:(1.52) lev:(0) conv:(1.39)
146. [sosis=1, cirpilmis_eksi_elma=1]: 86 ==> [diger_sebzeler=1]: 48 <conf:(0.56)> lift:(1.52) lev:(0) conv:(1.39)

147. [tam_yagli_sut=1, yogurt=1, kok_sebzeler=1]: 138 ==> [diger_sebzeler=1]: 77 <conf:(0.56)> lift:(1.52) lev:(0.01) conv:(1.41)
148. [kok_sebzeler=1, lor_peyniri=1]: 104 ==> [tam_yagli_sut=1]: 58 <conf:(0.56)> lift:(1.26) lev:(0) conv:(1.23)
149. [kok_sebzeler=1, beyaz_ekmek=1]: 70 ==> [tam_yagli_sut=1]: 39 <conf:(0.56)> lift:(1.26) lev:(0) conv:(1.22)
150. [tam_yagli_sut=1, soda=1, kok_sebzeler=1]: 79 ==> [diger_sebzeler=1]: 44 <conf:(0.56)> lift:(1.51) lev:(0) conv:(1.39)
151. [diger_sebzeler=1, Elma\Armut=1]: 234 ==> [tam_yagli_sut=1]: 130 <conf:(0.56)> lift:(1.25) lev:(0.01) conv:(1.24)
152. [diger_sebzeler=1, sivi_yag=1]: 90 ==> [tam_yagli_sut=1]: 50 <conf:(0.56)> lift:(1.25) lev:(0) conv:(1.22)
153. [tam_yagli_sut=1, yesillikler=1]: 72 ==> [diger_sebzeler=1]: 40 <conf:(0.56)> lift:(1.51) lev:(0) conv:(1.38)
154. [diger_sebzeler=1, yesillikler=1]: 72 ==> [tam_yagli_sut=1]: 40 <conf:(0.56)> lift:(1.25) lev:(0) conv:(1.21)
155. [tam_yagli_sut=1, yesillikler=1]: 72 ==> [kok_sebzeler=1]: 40 <conf:(0.56)> lift:(2.45) lev:(0.01) conv:(1.69)
156. [rulo_corekler=1, hamburger_koftesi=1]: 81 ==> [diger_sebzeler=1]: 45 <conf:(0.56)> lift:(1.51) lev:(0) conv:(1.38)
157. [diger_sebzeler=1, hijyen_malzemeleri=1]: 92 ==> [tam_yagli_sut=1]: 51 <conf:(0.55)> lift:(1.25) lev:(0) conv:(1.22)
158. [rulo_corekler=1, gofretler=1]: 83 ==> [tam_yagli_sut=1]: 46 <conf:(0.55)> lift:(1.25) lev:(0) conv:(1.22)
159. [sise_su=1, koy_yumurtasi=1]: 85 ==> [tam_yagli_sut=1]: 47 <conf:(0.55)> lift:(1.25) lev:(0) conv:(1.21)
160. [cirpilmis_eksi_elma=1, domuz=1]: 76 ==> [tam_yagli_sut=1]: 42 <conf:(0.55)> lift:(1.25) lev:(0) conv:(1.21)
161. [yogurt=1, kahve=1]: 87 ==> [tam_yagli_sut=1]: 48 <conf:(0.55)> lift:(1.24) lev:(0) conv:(1.21)
162. [kok_sebzeler=1, meyve_sebze_suyu=1]: 116 ==> [tam_yagli_sut=1]: 64 <conf:(0.55)> lift:(1.24) lev:(0) conv:(1.22)
163. [rulo_corekler=1, tavuk=1]: 87 ==> [diger_sebzeler=1]: 48 <conf:(0.55)> lift:(1.5) lev:(0) conv:(1.37)
164. [rulo_corekler=1, cirpilmis_eksi_elma=1]: 136 ==> [tam_yagli_sut=1]: 75 <conf:(0.55)> lift:(1.24) lev:(0) conv:(1.22)
165. [tam_yagli_sut=1, tropikal_meyve=1, cirpilmis_eksi_elma=1]: 78 ==> [yogurt=1]: 43 <conf:(0.55)> lift:(2.08) lev:(0.01) conv:(1.59)
166. [yogurt=1, tropikal_meyve=1]: 265 ==> [tam_yagli_sut=1]: 146 <conf:(0.55)> lift:(1.24) lev:(0.01) conv:(1.23)
167. [cirpilmis_eksi_elma=1, koy_yumurtasi=1]: 89 ==> [diger_sebzeler=1]: 49 <conf:(0.55)> lift:(1.5) lev:(0) conv:(1.37)
168. [diger_sebzeler=1, soda=1, kok_sebzeler=1]: 80 ==> [tam_yagli_sut=1]: 44 <conf:(0.55)> lift:(1.24) lev:(0) conv:(1.2)

169. [tam_yagli_sut=1, kok_sebzeler=1, sise_su=1]: 71 ==> [diger_sebzeler=1]: 39 <conf:(0.55)> lift:(1.49) lev:(0) conv:(1.36)
170. [kok_sebzeler=1, domuz=1]: 122 ==> [diger_sebzeler=1]: 67 <conf:(0.55)> lift:(1.49) lev:(0.01) conv:(1.38)
171. [kok_sebzeler=1, alisveris_cantolari=1]: 113 ==> [diger_sebzeler=1]: 62 <conf:(0.55)> lift:(1.49) lev:(0.01) conv:(1.37)
172. [meyve_sebze_suyu=1, tereyagi=1]: 73 ==> [tam_yagli_sut=1]: 40 <conf:(0.55)> lift:(1.24) lev:(0) conv:(1.19)
173. [hamburger_koftesi=1]: 252 ==> [tam_yagli_sut=1]: 138 <conf:(0.55)> lift:(1.23) lev:(0.01) conv:(1.22)
174. [yogurt=1, unlu_mamuller=1]: 159 ==> [tam_yagli_sut=1]: 87 <conf:(0.55)> lift:(1.23) lev:(0) conv:(1.21)
175. [tropikal_meyve=1, unlu_mamuller=1]: 119 ==> [tam_yagli_sut=1]: 65 <conf:(0.55)> lift:(1.23) lev:(0) conv:(1.2)
176. [diger_sebzeler=1, yogurt=1]: 390 ==> [tam_yagli_sut=1]: 213 <conf:(0.55)> lift:(1.23) lev:(0.01) conv:(1.22)
177. [yesillikler=1]: 132 ==> [tam_yagli_sut=1]: 72 <conf:(0.55)> lift:(1.23) lev:(0) conv:(1.2)
178. [yesillikler=1]: 132 ==> [diger_sebzeler=1]: 72 <conf:(0.55)> lift:(1.48) lev:(0.01) conv:(1.37)
179. [koy_yumurtasi=1]: 496 ==> [tam_yagli_sut=1]: 270 <conf:(0.54)> lift:(1.23) lev:(0.01) conv:(1.22)
180. [yogurt=1, meyve_sebze_suyu=1]: 171 ==> [tam_yagli_sut=1]: 93 <conf:(0.54)> lift:(1.23) lev:(0) conv:(1.2)
181. [narenciye=1, cirpilmis_eksi_elma=1]: 103 ==> [diger_sebzeler=1]: 56 <conf:(0.54)> lift:(1.48) lev:(0) conv:(1.36)
182. [kok_sebzeler=1, sosis=1]: 138 ==> [tam_yagli_sut=1]: 75 <conf:(0.54)> lift:(1.23) lev:(0) conv:(1.2)
183. [yogurt=1, sigir_eti=1]: 105 ==> [tam_yagli_sut=1]: 57 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.19)
184. [tam_yagli_sut=1, kok_sebzeler=1, koy_yumurtasi=1]: 83 ==> [diger_sebzeler=1]: 45 <conf:(0.54)> lift:(1.47) lev:(0) conv:(1.35)
185. [rulo_corekler=1, sigir_eti=1]: 120 ==> [tam_yagli_sut=1]: 65 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.19)
186. [tropikal_meyve=1, lor_peyniri=1]: 96 ==> [diger_sebzeler=1]: 52 <conf:(0.54)> lift:(1.47) lev:(0) conv:(1.35)
187. [cirpilmis_eksi_elma=1, dondurulmus_sebzeler=1]: 72 ==> [diger_sebzeler=1]: 39 <conf:(0.54)> lift:(1.47) lev:(0) conv:(1.34)
188. [kok_sebzeler=1, gazeteler=1]: 109 ==> [diger_sebzeler=1]: 59 <conf:(0.54)> lift:(1.47) lev:(0.01) conv:(1.35)
189. [kok_sebzeler=1, dondurulmus_sebzeler=1]: 109 ==> [diger_sebzeler=1]: 59 <conf:(0.54)> lift:(1.47) lev:(0.01) conv:(1.35)
190. [dondurulmus_balik=1]: 85 ==> [tam_yagli_sut=1]: 46 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.18)

191. [yogurt=1, beyaz_ekmek=1]: 85 ==> [tam_yagli_sut=1]: 46 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.18)
192. [kok_sebzeler=1, domuz=1]: 122 ==> [tam_yagli_sut=1]: 66 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.19)
193. [kok_sebzeler=1, narenciye=1]: 161 ==> [tam_yagli_sut=1]: 87 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.19)
194. [narenciye=1, unlu_mamuller=1]: 87 ==> [tam_yagli_sut=1]: 47 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.18)
195. [soganlar=1]: 239 ==> [diger_sebzeler=1]: 129 <conf:(0.54)> lift:(1.47) lev:(0.01) conv:(1.36)
196. [yogurt=1, domuz=1]: 89 ==> [tam_yagli_sut=1]: 48 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.18)
197. [diger_sebzeler=1, uzun_omurlu_firin_urumu=1]: 102 ==> [tam_yagli_sut=1]: 55 <conf:(0.54)> lift:(1.22) lev:(0) conv:(1.18)
198. [diger_sebzeler=1, cirpilmis_eksi_elma=1]: 260 ==> [tam_yagli_sut=1]: 140 <conf:(0.54)> lift:(1.21) lev:(0.01) conv:(1.2)
199. [tam_yagli_sut=1, yogurt=1, Elma\Armut=1]: 93 ==> [diger_sebzeler=1]: 50 <conf:(0.54)> lift:(1.46) lev:(0) conv:(1.34)
200. [tam_yagli_sut=1, yogurt=1, meyve_sebze_suyu=1]: 93 ==> [diger_sebzeler=1]: 50 <conf:(0.54)> lift:(1.46) lev:(0) conv:(1.34)
201. [tropikal_meyve=1, beyaz_ekmek=1]: 84 ==> [tam_yagli_sut=1]: 45 <conf:(0.54)> lift:(1.21) lev:(0) conv:(1.17)
202. [kok_sebzeler=1, krem_peyniri=1]: 71 ==> [tam_yagli_sut=1]: 38 <conf:(0.54)> lift:(1.21) lev:(0) conv:(1.16)
203. [unlu_mamuller=1, tereyagi=1]: 73 ==> [tam_yagli_sut=1]: 39 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.16)
204. [cirpilmis_eksi_elma=1, cilek=1]: 73 ==> [tam_yagli_sut=1]: 39 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.16)
205. [kok_sebzeler=1, koy_yumurtasi=1]: 133 ==> [diger_sebzeler=1]: 71 <conf:(0.53)> lift:(1.45) lev:(0.01) conv:(1.33)
206. [meyve_sebze_suyu=1, dondurulmus_sebzeler=1]: 75 ==> [tam_yagli_sut=1]: 40 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.16)
207. [rulo_corekler=1, kok_sebzeler=1]: 227 ==> [tam_yagli_sut=1]: 121 <conf:(0.53)> lift:(1.2) lev:(0.01) conv:(1.18)
208. [unlu_mamuller=1, meyve_sebze_suyu=1]: 79 ==> [tam_yagli_sut=1]: 42 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.16)
209. [tropikal_meyve=1, lor_peyniri=1]: 96 ==> [yogurt=1]: 51 <conf:(0.53)> lift:(2.01) lev:(0.01) conv:(1.53)
210. [rulo_corekler=1, hamburger_koftesi=1]: 81 ==> [tam_yagli_sut=1]: 43 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.16)
211. [yogurt=1, cikolata=1]: 81 ==> [tam_yagli_sut=1]: 43 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.16)
212. [tropikal_meyve=1, sosis=1]: 130 ==> [tam_yagli_sut=1]: 69 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.17)

213. [diger_sebzeler=1, jambon=1]: 83 ==> [tam_yagli_sut=1]: 44 <conf:(0.53)> lift:(1.2) lev:(0) conv:(1.15)
214. [yogurt=1, kok_sebzeler=1]: 234 ==> [diger_sebzeler=1]: 124 <conf:(0.53)> lift:(1.44) lev:(0.01) conv:(1.33)
215. [sosis=1, Elma\Armut=1]: 102 ==> [tam_yagli_sut=1]: 54 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.16)
216. [cirpilmis_eksi_elma=1]: 543 ==> [tam_yagli_sut=1]: 287 <conf:(0.53)> lift:(1.19) lev:(0.01) conv:(1.18)
217. [sosis=1, meyve_sebze_suyu=1]: 89 ==> [tam_yagli_sut=1]: 47 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.15)
218. [diger_sebzeler=1, tropikal_meyve=1, Elma\Armut=1]: 89 ==> [tam_yagli_sut=1]: 47 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.15)
219. [cirpilmis_eksi_elma=1, dondurulmus_sebzeler=1]: 72 ==> [tam_yagli_sut=1]: 38 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.14)
220. [yogurt=1, margarin=1]: 129 ==> [tam_yagli_sut=1]: 68 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.16)
221. [un=1]: 148 ==> [tam_yagli_sut=1]: 78 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.16)
222. [yogurt=1, yagli_sut=1]: 74 ==> [diger_sebzeler=1]: 39 <conf:(0.53)> lift:(1.43) lev:(0) conv:(1.3)
223. [kok_sebzeler=1, baharatli_alman_sosisi=1]: 93 ==> [tam_yagli_sut=1]: 49 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.15)
224. [yogurt=1, esmer_ekmek=1]: 131 ==> [tam_yagli_sut=1]: 69 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.16)
225. [diger_sebzeler=1, rulo_corekler=1, tropikal_meyve=1]: 76 ==> [tam_yagli_sut=1]: 40 <conf:(0.53)> lift:(1.19) lev:(0) conv:(1.14)
226. [kok_sebzeler=1, cirpilmis_eksi_elma=1]: 158 ==> [diger_sebzeler=1]: 83 <conf:(0.53)> lift:(1.43) lev:(0.01) conv:(1.31)
227. [sise_su=1, cirpilmis_eksi_elma=1]: 80 ==> [tam_yagli_sut=1]: 42 <conf:(0.53)> lift:(1.18) lev:(0) conv:(1.14)
228. [diger_sebzeler=1, yogurt=1, soda=1]: 82 ==> [tam_yagli_sut=1]: 43 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
229. [rulo_corekler=1, kok_sebzeler=1]: 227 ==> [diger_sebzeler=1]: 119 <conf:(0.52)> lift:(1.42) lev:(0.01) conv:(1.32)
230. [rulo_urunler=1]: 84 ==> [tam_yagli_sut=1]: 44 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
231. [diger_sebzeler=1, kok_sebzeler=1]: 420 ==> [tam_yagli_sut=1]: 220 <conf:(0.52)> lift:(1.18) lev:(0.01) conv:(1.16)
232. [unlu_mamuller=1, esmer_ekmek=1]: 84 ==> [tam_yagli_sut=1]: 44 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
233. [kok_sebzeler=1, soganlar=1]: 86 ==> [tam_yagli_sut=1]: 45 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
234. [yumusak_peynir=1]: 130 ==> [diger_sebzeler=1]: 68 <conf:(0.52)> lift:(1.42) lev:(0.01) conv:(1.3)

235. [Elma\Armut=1, meyve_sebze_suyu=1]: 90 ==> [tam_yagli_sut=1]: 47 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
236. [yogurt=1, peceteler=1]: 113 ==> [tam_yagli_sut=1]: 59 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
237. [yogurt=1, dondurulmus_sebzeler=1]: 113 ==> [tam_yagli_sut=1]: 59 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
238. [diger_sebzeler=1, sise_bira=1]: 138 ==> [tam_yagli_sut=1]: 72 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.15)
239. [tropikal_meyve=1, peceteler=1]: 94 ==> [tam_yagli_sut=1]: 49 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.14)
240. [diger_sebzeler=1, sigir_eti=1]: 165 ==> [tam_yagli_sut=1]: 86 <conf:(0.52)> lift:(1.18) lev:(0) conv:(1.15)
241. [kok_sebzeler=1, tereyagi=1]: 121 ==> [diger_sebzeler=1]: 63 <conf:(0.52)> lift:(1.42) lev:(0) conv:(1.3)
242. [unlu_mamuller=1, tereyagi=1]: 73 ==> [diger_sebzeler=1]: 38 <conf:(0.52)> lift:(1.41) lev:(0) conv:(1.28)
243. [tam_yagli_sut=1, kok_sebzeler=1, sigir_eti=1]: 73 ==> [diger_sebzeler=1]: 38 <conf:(0.52)> lift:(1.41) lev:(0) conv:(1.28)
244. [kok_sebzeler=1, lor_peyniri=1]: 104 ==> [diger_sebzeler=1]: 54 <conf:(0.52)> lift:(1.41) lev:(0) conv:(1.29)
245. [tam_yagli_sut=1, yogurt=1, cirpilmis_eksi_elma=1]: 106 ==> [diger_sebzeler=1]: 55 <conf:(0.52)> lift:(1.41) lev:(0) conv:(1.29)
246. [diger_sebzeler=1, domuz=1]: 187 ==> [tam_yagli_sut=1]: 97 <conf:(0.52)> lift:(1.17) lev:(0) conv:(1.14)
247. [diger_sebzeler=1, rulo_corekler=1, yogurt=1]: 112 ==> [tam_yagli_sut=1]: 58 <conf:(0.52)> lift:(1.17) lev:(0) conv:(1.13)
248. [diger_sebzeler=1, tropikal_meyve=1, narenciye=1]: 85 ==> [kok_sebzeler=1]: 44 <conf:(0.52)> lift:(2.28) lev:(0.01) conv:(1.57)
249. [diger_sebzeler=1, dilimlenmis_peynir=1]: 87 ==> [tam_yagli_sut=1]: 45 <conf:(0.52)> lift:(1.17) lev:(0) conv:(1.13)
250. [rulo_corekler=1, tereyagi=1]: 118 ==> [tam_yagli_sut=1]: 61 <conf:(0.52)> lift:(1.17) lev:(0) conv:(1.13)
251. [kok_sebzeler=1, Elma\Armut=1]: 151 ==> [diger_sebzeler=1]: 78 <conf:(0.52)> lift:(1.4) lev:(0.01) conv:(1.29)
252. [sise_su=1, margarin=1]: 93 ==> [tam_yagli_sut=1]: 48 <conf:(0.52)> lift:(1.16) lev:(0) conv:(1.12)
253. [diger_sebzeler=1, krem_peyniri=1]: 126 ==> [tam_yagli_sut=1]: 65 <conf:(0.52)> lift:(1.16) lev:(0) conv:(1.13)
254. [diger_sebzeler=1, meyve_sebze_suyu=1]: 194 ==> [tam_yagli_sut=1]: 100 <conf:(0.52)> lift:(1.16) lev:(0) conv:(1.14)
255. [rulo_corekler=1, esmer_ekmek=1]: 97 ==> [tam_yagli_sut=1]: 50 <conf:(0.52)> lift:(1.16) lev:(0) conv:(1.12)
256. [rulo_corekler=1, dondurulmus_sebzeler=1]: 97 ==> [tam_yagli_sut=1]: 50 <conf:(0.52)> lift:(1.16) lev:(0) conv:(1.12)

257. [yogurt=1, narenciye=1]: 194 ==> [tam_yagli_sut=1]: 100 <conf:(0.52)> lift:(1.16) lev:(0) conv:(1.14)
258. [tam_yagli_sut=1, dondurulmus_sebzeler=1]: 179 ==> [diger_sebzeler=1]: 92 <conf:(0.51)> lift:(1.4) lev:(0.01) conv:(1.29)
259. [diger_sebzeler=1, tropikal_meyve=1]: 323 ==> [tam_yagli_sut=1]: 166 <conf:(0.51)> lift:(1.16) lev:(0.01) conv:(1.14)
260. [kok_sebzeler=1, gazeteler=1]: 109 ==> [tam_yagli_sut=1]: 56 <conf:(0.51)> lift:(1.16) lev:(0) conv:(1.12)
261. [tam_yagli_sut=1, yogurt=1, tropikal_meyve=1]: 146 ==> [diger_sebzeler=1]: 75 <conf:(0.51)> lift:(1.4) lev:(0.01) conv:(1.28)
262. [narenciye=1, sigir_eti=1]: 74 ==> [tam_yagli_sut=1]: 38 <conf:(0.51)> lift:(1.16) lev:(0) conv:(1.11)
263. [yogurt=1, cirpilmis_eksi_elma=1]: 185 ==> [diger_sebzeler=1]: 95 <conf:(0.51)> lift:(1.4) lev:(0.01) conv:(1.28)
264. [Elma\Armut=1, lor_peyniri=1]: 74 ==> [yogurt=1]: 38 <conf:(0.51)> lift:(1.94) lev:(0) conv:(1.47)
265. [soda=1, sigir_eti=1]: 74 ==> [kok_sebzeler=1]: 38 <conf:(0.51)> lift:(2.27) lev:(0.01) conv:(1.55)
266. [diger_sebzeler=1, rulo_corekler=1, kok_sebzeler=1]: 119 ==> [tam_yagli_sut=1]: 61 <conf:(0.51)> lift:(1.16) lev:(0) conv:(1.12)
267. [tam_yagli_sut=1, kok_sebzeler=1, tereyagi=1]: 80 ==> [diger_sebzeler=1]: 41 <conf:(0.51)> lift:(1.39) lev:(0) conv:(1.26)
268. [kok_sebzeler=1]: 845 ==> [tam_yagli_sut=1]: 433 <conf:(0.51)> lift:(1.16) lev:(0.02) conv:(1.14)
269. [rulo_urunler=1]: 84 ==> [diger_sebzeler=1]: 43 <conf:(0.51)> lift:(1.39) lev:(0) conv:(1.26)
270. [narenciye=1, tereyagi=1]: 84 ==> [diger_sebzeler=1]: 43 <conf:(0.51)> lift:(1.39) lev:(0) conv:(1.26)
271. [seker=1]: 256 ==> [tam_yagli_sut=1]: 131 <conf:(0.51)> lift:(1.15) lev:(0) conv:(1.13)
272. [rulo_corekler=1, sise_bira=1]: 96 ==> [tam_yagli_sut=1]: 49 <conf:(0.51)> lift:(1.15) lev:(0) conv:(1.11)
273. [kok_sebzeler=1, sigir_eti=1]: 149 ==> [diger_sebzeler=1]: 76 <conf:(0.51)> lift:(1.39) lev:(0.01) conv:(1.27)
274. [diger_sebzeler=1, tavuk=1]: 157 ==> [tam_yagli_sut=1]: 80 <conf:(0.51)> lift:(1.15) lev:(0) conv:(1.12)
275. [diger_sebzeler=1, esmer_ekmek=1]: 171 ==> [tam_yagli_sut=1]: 87 <conf:(0.51)> lift:(1.15) lev:(0) conv:(1.12)
276. [diger_sebzeler=1, hamburger_koftesi=1]: 120 ==> [tam_yagli_sut=1]: 61 <conf:(0.51)> lift:(1.15) lev:(0) conv:(1.11)
277. [tam_yagli_sut=1, kok_sebzeler=1]: 433 ==> [diger_sebzeler=1]: 220 <conf:(0.51)> lift:(1.38) lev:(0.02) conv:(1.28)
278. [tavuk=1]: 313 ==> [tam_yagli_sut=1]: 159 <conf:(0.51)> lift:(1.15) lev:(0.01) conv:(1.12)

279. [yumusak_peynir=1]: 130 ==> [tam_yagli_sut=1]: 66 <conf:(0.51)> lift:(1.14) lev:(0) conv:(1.11)
280. [koy_yumurtasi=1, margarin=1]: 77 ==> [diger_sebzeler=1]: 39 <conf:(0.51)> lift:(1.38) lev:(0) conv:(1.25)
281. [yogurt=1]: 988 ==> [tam_yagli_sut=1]: 500 <conf:(0.51)> lift:(1.14) lev:(0.02) conv:(1.12)
282. [tam_yagli_sut=1, kabartma_tozu=1]: 85 ==> [diger_sebzeler=1]: 43 <conf:(0.51)> lift:(1.37) lev:(0) conv:(1.25)
283. [tam_yagli_sut=1, rulo_corekler=1, soda=1]: 85 ==> [diger_sebzeler=1]: 43 <conf:(0.51)> lift:(1.37) lev:(0) conv:(1.25)
284. [yogurt=1, domuz=1]: 89 ==> [diger_sebzeler=1]: 45 <conf:(0.51)> lift:(1.37) lev:(0) conv:(1.25)
285. [koy_yumurtasi=1, tereyagi=1]: 89 ==> [diger_sebzeler=1]: 45 <conf:(0.51)> lift:(1.37) lev:(0) conv:(1.25)
286. [diger_sebzeler=1, cilek=1]: 93 ==> [tam_yagli_sut=1]: 47 <conf:(0.51)> lift:(1.14) lev:(0) conv:(1.1)
287. [pasta_dilimi=1]: 97 ==> [tam_yagli_sut=1]: 49 <conf:(0.51)> lift:(1.14) lev:(0) conv:(1.1)
288. [rulo_corekler=1, yogurt=1]: 295 ==> [tam_yagli_sut=1]: 149 <conf:(0.51)> lift:(1.14) lev:(0) conv:(1.12)
289. [yagli_sut=1]: 200 ==> [tam_yagli_sut=1]: 101 <conf:(0.51)> lift:(1.14) lev:(0) conv:(1.11)
290. [rulo_corekler=1, domuz=1]: 103 ==> [diger_sebzeler=1]: 52 <conf:(0.5)> lift:(1.37) lev:(0) conv:(1.25)
291. [tam_yagli_sut=1, rulo_corekler=1, kok_sebzeler=1]: 121 ==> [diger_sebzeler=1]: 61 <conf:(0.5)> lift:(1.37) lev:(0) conv:(1.25)
292. [diger_sebzeler=1, kahve=1]: 123 ==> [tam_yagli_sut=1]: 62 <conf:(0.5)> lift:(1.14) lev:(0) conv:(1.1)
293. [tropikal_meyve=1, meyve_sebze_suyu=1]: 129 ==> [diger_sebzeler=1]: 65 <conf:(0.5)> lift:(1.37) lev:(0) conv:(1.25)
294. [tam_yagli_sut=1, tavuk=1]: 159 ==> [diger_sebzeler=1]: 80 <conf:(0.5)> lift:(1.37) lev:(0.01) conv:(1.26)
295. [tropikal_meyve=1, narenciye=1]: 173 ==> [tam_yagli_sut=1]: 87 <conf:(0.5)> lift:(1.13) lev:(0) conv:(1.11)
296. [diger_sebzeler=1, narenciye=1]: 247 ==> [tam_yagli_sut=1]: 124 <conf:(0.5)> lift:(1.13) lev:(0) conv:(1.11)
297. [tavuk=1]: 313 ==> [diger_sebzeler=1]: 157 <conf:(0.5)> lift:(1.36) lev:(0.01) conv:(1.26)

ÖZ GEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Yusuf TAŞ
Uyruğu : Türk
Doğum Tarihi ve Yeri : 21/10/1990 Seyhan
e-posta : yusuf_tas_35@hotmail.com

EĞİTİM

Derece	Kurum	Mezuniyet Yılı
Lisans	Cumhuriyet Üniversitesi, Yönetim Bilişim Sistemleri Bölümü	2013
Yüksek Lisans	Cumhuriyet Üniversitesi, Yönetim Bilişim Sistemleri Bölümü	2018

İŞ TECRÜBESİ

Tarih	Kurum	Görev
-------	-------	-------

YABANCI DİL BİLGİSİ

Yabancı Dilin Adı KPDS () ÜDS () TOEFL () IELTS ()