



**T.C. İSTANBUL TİCARET
ÜNİVERSİTESİ**

FEN BİLİMLERİ ENSTİTÜSÜ

**VERİ MADENCİLİĞİ ALGORİTMALARI İLE BİRLİKTELİK
KURALLARININ BELİRLENMESİ: PERAKENDE SEKTÖRÜNDE BİR
UYGULAMA**

Ayşe Nur SAĞIN

**Danışman
Dr. Öğr. Üyesi Berk AYVAZ**

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
İSTANBUL - 2018**

KABUL VE ONAY SAYFASI

Ayşe Nur SAĞIN tarafından hazırlanan "Veri Madenciliği Algoritmaları ile Birliktelik Kurallarının Belirlenmesi : Perakende Sektöründe Bir Uygulama" adlı tez çalışması 10/07/2018 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman

Dr. Öğr.Üyesi Berk AYVAZ
İstanbul Ticaret Üniversitesi



Jüri Üyesi

Prof. Dr. Mustafa KÖKSAL
İstanbul Ticaret Üniversitesi



Jüri Üyesi

Dr.Öğr.Üyesi Ali Osman KUŞAKCI
İbn Haldun Üniversitesi

Onay Tarihi : 23.07.2018


Prof. Dr. Necip ŞİMŞEK
Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

Tarih: 10.07.2018

İmza



Ayşe Nur SAĞIN

İÇİNDEKİLER

İÇİNDEKİLER.....	i
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR.....	v
ŞEKİLLER.....	vi
ÇİZELGELER.....	vii
SİMGELER VE KISALTMALAR.....	viii
1. GİRİŞ.....	9
2. LİTERATÜR ÖZETİ.....	11
3. VERİ MADENCİLİĞİ.....	15
3.1. Veri Madenciliği Temel Kavramları	15
3.1.1. Temel Kavramlar; Veri, Malumat, Bilgi, İrfan.....	16
3.1.2. Veri yığını	17
3.1.3. Veri tabanı.....	17
3.1.4. Veri ambarı.....	21
3.2. Verinin Yararlı Bilgiye Dönüşüm Süreci	23
3.3. Veri Madenciliğinin Tanımı.....	24
3.4. Veri Madenciliği Tarihsel Gelişimi	25
3.5. Veri Madenciliği Uygulama Alanları	27
3.7. Veri Madenciliği Modelleri	29
3.7.1. Sınıflama ve Regresyon.....	30
3.7.1.1. Yapay Sinir Ağları (Artificial Neural Networks)	30
3.7.1.2. Genetik Algoritmalar (Genetic Algorithms).....	33
3.7.1.3. K- En Yakın Komşu (K- Nearest Neighbour)	35
3.7.1.4. Naive Bayes Sınıflandırıcısı.....	35
3.7.1.5. Karar Ağaçları (Decision Trees)	37
3.7.1.6. Lojistik Regresyon	38
3.7.2. Kümeleme	39
3.7.3. Birliktelik (İlişki) Kuralları ve Ardışık Zamanlı Örüntüler	40
4. BİRLİKTEKİLİK (İLİŞKİ) KURALLARI.....	42
4.1. Birliktelik Kurallarının Matematiksel Modeli	43

4.2. Birliktelik Kuralı Madenciliği	47
4.3. Birliktelik Kuralları Madenciliği Algoritmaları	48
4.3.1. Apriori Algoritması	48
4.3.2. Apriori Tid Algoritması.....	54
4.3.3. Apriori-Hybrid Algoritması.....	55
4.3.4. AIS Algoritması	56
4.3.5. SETM	56
4.3.6. FP-Growth (Frequent Pattern Growth) Algoritması	57
5. METODOLOJİ	62
6. SONUÇ VE ÖNERİLER.....	79
KAYNAKLAR	82
EKLER.....	88
EK A. Çizelgeler	88
ÖZGEÇMİŞ.....	93

ÖZET

Yüksek Lisans Tezi

VERİ MADENCİLİĞİ ALGORİTMALARI İLE BİRLİKTELİK KURALLARININ BELİRLENMESİ: PERAKENDE SEKTÖRÜNDE BİR UYGULAMA

Ayşe Nur SAĞIN

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Berk AYVAZ

2018, 92 sayfa

Market sepet analizi, müşterilerin tek bir alışverişte satın aldığı ürünler dikkate alınarak, satın alma eğilimlerinin şirket veri tabanlarındaki kayıtlar ile ortaya çıkarılması işlemidir.

Bu çalışmada; perakende sektöründe faaliyet gösteren büyük bir hırdavat şirketinin beş buçuk yıllık verileri üzerinde market sepet analizi uygulanarak ilişkili ürün kategorileri belirlenmiştir. Birliktelik kurallarının belirlenmesinde Apriori ve FP-Growth algoritmalarının her ikisi de ayrı ayrı çalıştırılarak bu tip bir veri setindeki kullanışlılıkları kıyaslanmıştır. Ayrıca veri seti; Veri Seti-1 ve Veri Seti-2 olacak şekilde ikiye bölünmüş, böylece ilk veri setinden çıkarılan kuralların doğruluğu, ardışık zamanlı verileri içeren ikinci veri setinden çıkarılan kurallar ile karşılaştırılarak kuralların tutarlılığı tartışılmıştır.

Anahtar Kelimeler: Apriori algoritması, birliktelik kuralları, fp-growth algoritması, market sepet analizi, veri madenciliği

ABSTRACT

M.Sc. Thesis

DETERMINATION OF ASSOCIATION RULES WITH DATA MINING ALGORITHMS: AN APPLICATION IN RETAIL SECTOR

Ayşe Nur SAĞIN

**İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Industrial Engineering**

Supervisor: Assist. Prof. Dr. Berk AYVAZ

2018, 92 pages

Market basket analysis is the process of extracting purchasing trends from records in company databases, taking into account the products that customers buy in a single trade.

Market basket analysis is the process of extracting purchasing trends from records in company databases, taking into account the products that customers buy in a single transaction. In this study, a market basket analysis was conducted on a five-and-a-half year data of a large hardware company operating in the retail sector, and related product categories were identified. In determining the association rules, both the Apriori and FP-Growth algorithms were run separately and their usefulness in such a set of data was compared. In addition, the data set was divided into Data Set-1 and Data Set-2 so that the consistency of the rules was discussed by comparing the correctness of rules extracted from the first data set with rules derived from the second data set containing consecutive timed data.

Keywords: Apriori algorithm, association rules, data mining, fp-growth algorithm, market basket analysis.

TEŐEKKÜR

Bu arařtırma iin beni ynlendiren, karřılařtıđım zorlukları bilgi ve tecrbesi ile ařmamda yardımcı olan deđerli Danıřman Hocam Dr. đr.yesi, Berk AYVAZ'a, alıřmamın uygulama ařamasında benden yardım ve desteđini esirgemeyen Dr. đr.yesi, Mustafa Cem KASAPBAŐI'na, uygulama blm iin gerek verilere ulařmamı sađlayan End. Mh.Aykut KURTULUŐ'a teŐekkrlerimi sunarım.

Bana her konuda kořulsuz destek veren ve tezimin her ařamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Ayře Nur SAđIN
İSTANBUL, 2018



ŞEKİLLER

	Sayfa
Şekil 3.1. Bilgi Hiyerarşisi.....	16
Şekil 3.2. İlişkisel veri modeli.....	18
Şekil 3.3. VTBK sürecinin aşamaları.....	23
Şekil 3.4. Basit bir üç katmanlı yapay sinir ağı modeli.....	32
Şekil 3.5. Klasik karar ağacı gösterimi.....	37
Şekil 4.1. Sık olamayan nesnekümenin üst kümelerinin de sık olamayacağı_ nın, 5 elemanlı bir kümenin olası altkümeleri üzerinden gösterimi..	49
Şekil 4.2. Örnek veri tabanı üzerinde Apriori algoritması adımları.....	52
Şekil 4.3. Apriori-Tid algoritması adımları.....	55
Şekil 4.4. Adım adım FP ağaçlarının oluşturulması.....	60
Şekil 4.5. Oluşturulan FP ağacının son hali.....	61
Şekil 4.6. FP ağacı üzerindeki budanacak dalların belirlenmesi	61
Şekil 5.1. Akış diyagramı.....	62
Şekil 5.2. VTBK adımları sonucu verinin dönüşümü	63
Şekil 5.3. X Hırdavat alışveriş kayıtları SQL tablosundan bir kesit	64
Şekil 5.4. Düzenlenen alt ürün grubu tablosundan bir kesit.....	65
Şekil 5.5. Veri Seti-1 için Excel Pivot tablosu oluşturulması.....	66
Şekil 5.6. Veri Seti-1'in tabular forma çevrilmiş Excel tablosu	66
Şekil 5.7. Excel2Arff Converter programı çalışma sayfası	67
Şekil 5.8. Kategorilerin .arff formatındaki görüntüsünden bir kesit	68
Şekil 5.9. Arff formatına dönüştürülmüş verilerden bir kesit	68
Şekil 5.10. WEKA programı giriş ekranı	69
Şekil 5.11. WEKA programı 'Preprocess' ekranının görüntüsü.....	69
Şekil 5.12. Veri Seti -1 için Apriori sonuçları	70
Şekil 5.13. Veri Seti -2 için Apriori algoritması sonuçları	73
Şekil 5.14. Veri Seti-1 için FP-Growth Algoritması sonuçları	75
Şekil 5.15. Veri Seti-2 için FP-Growth Algoritması sonuçları.....	76

ÇİZELGELER

	Sayfa
Çizelge 3.1. OLTP ve OLAP sistemlerinin karşılaştırılması	21
Çizelge 3.2. Veri tabanı /Veri ambarı karşılaştırması	22
Çizelge 3.3. Veri madenciliği kavramının tarihsel gelişimi.....	27
Çizelge 4.1. Örnek veri tabanı	45
Çizelge 4.2. Örnek veri tabanı üzerinden 1,2,3 ve 4 elemanlı nesnekümlerin destek değerlerinin hesaplanması.....	46
Çizelge 4.3. Klasik Apriori algoritması özel kodu.....	50
Çizelge 4.4. Apriori-gen fonksiyonu kodu	51
Çizelge 4.5. Apriori algoritması için örnek veri tabanı	52
Çizelge 4.6. FP-Growth algoritması için örnek veri tabanı.....	58
Çizelge 5.1. Veri Seti-1 için Apriori Algoritması sonuçları	71
Çizelge 5.2. Veri Seti-2 için Apriori Algoritması sonuçları	74
Çizelge 5.3. Veri Seti-1 için FP-Growth algoritması sonuçları.....	75
Çizelge 5.4. Veri Seti-2 için FP-Growth Algoritması sonuçları.....	77

SİMGELER VE KISALTMALAR

DIY	Do It Yourself
GB	Gigabyte
KDD	Knowledge Discovery in Databases
OLAP	Çevrimiçi analitik işleme (Online Analytical Processing)
OLTP	Optimize edilmiş sistemler (Online Transaction Processing)
SQL	Yapılandırılmış sorgu dili (Structured Query Language)
VTBK	Veri Tabanından Bilgi Keşfi
VTYS	Veri tabanı yönetim sistemi



1. GİRİŞ

Günlük hayatta gerçekleştirdiğimiz her türlü işlem, gelişmiş veri tabanı sistemleri sayesinde, sayısal ortamlarda kayıt altına alınmaktadır. Bir market alışverişinde birlikte satın aldığımız ürünler ile iade ettiğimiz ürünlerin bilgileri, bankacılık işlemleri, telefon görüşmeleri veya internet üzerinden alışverişler sırasında yaptığımız her işlem eşzamanlı olarak kaydedilmektedir (Silahtaroglu, 2016). Bilişim teknolojisindeki böylesine bir gelişim, şirketler için beraberinde bazı sorunları da getirmiştir. Çok fazla müşteri sayısına sahip ve işlem hareketliliği çok fazla olan işletmelerde, her gün büyük miktarda veri üretilmesiyle devasa veri yığınları oluşmaktadır. Günümüz veri tabanı teknolojisi bu veri yığınlarını sağlıklı bir şekilde saklayacak kadar gelişmiştir fakat önemli olan bu verileri elde tutmak değil, verileri şirket için katma değer üretecek şekilde değerlendirebilmektir.

İşletmelerin, günümüz tüketici odaklı pazarlarındaki rekabet koşullarında ayakta kalabilmeleri için etkin ve düşük maliyetli pazarlama stratejileri geliştirmeleri gerekir (Emel ve Taşkın, 2002). Stratejik kararların verilebilmesi için ise ilk olarak güvenilir ve mümkün olduğunca kanıta dayalı bilgiye ihtiyaç duyulur. Gelişen teknoloji ile birlikte, veri madenciliği bu ihtiyaca verilebilecek en iyi cevaplardan biri haline gelmiştir. Veri madenciliği, büyük veri tabanlarından değerli bilgiyi süzme tekniğidir (Emel ve Taşkın, 2002). Büyük veri yığınları içerisindeki gizli kalmış bilginin gün yüzüne çıkarılarak şirket için değerli bilgilere dönüştürülmesi veri madenciliği ile mümkündür.

Bu çalışmada veri madenciliği yöntemlerinden olan birliktelik kurallarının belirlenmesi, market sepet analizi ile gerçekleştirilmiştir. Analiz ve uygulama süreci için büyük bir hırdavat şirketinin şubesinden alınan satış verileri kullanılmıştır. Analizin amacı, işletmede belirlenen dönem içerisinde birlikte satılma eğiliminde olan ürün gruplarını belirlemektir. Çalışma sonucunda çıkarılacak kuralların geçerliliğini test edebilmek için ardışık dönemlerdeki verileri içeren veri seti, iki parçaya bölünmüştür. Veri seti çeşitli aşamalardan

geçirilerek, bir veri madenciliği yazılımı olan WEKA'nın anlayabileceği formata dönüştürülmüştür. WEKA programında Apriori ve FP-Growth algoritmaları kullanılarak, veri seti içerisindeki birliktelik kuralları çıkarılmıştır. Algoritmaların, bu boyutta bir veri seti üzerinde çalışma performansı değerlendirilmiştir.

Çalışmanın ilerleyen bölümleri şu şekilde düzenlenmiştir: İkinci bölümde, birliktelik kurallarının çeşitli sektörlerdeki kullanım amacını içeren literatür çalışması yer almaktadır. Üçüncü bölümde, çalışmanın ana konusu olan veri madenciliği ayrıntılı bir şekilde anlatılmıştır. Veri madenciliği temel kavramları, veri madenciliği süreci ve veri madenciliği yöntemlerine ayrı ayrı yer verilerek konu hakkında genel bir bilgilendirme yapılmıştır. Çalışmada, veri madenciliği yöntemlerinden birliktelik kuralları kullanılarak analiz gerçekleştirildiğinden dördüncü bölümde, birliktelik kuralları ayrıntılı bir şekilde ele alınmıştır. Beşinci bölümde, çalışmada kullanılan verinin içeriği ve uygulamanın adımları anlatılmaktadır. Son bölüm olan altıncı bölümde ise yapılan analizlerin sonuçları ve çalışmanın ileride geliştirilebileceği alanlar konusundaki öneriler yer almaktadır.

2. LİTERATÜR ÖZETİ

Poel vd. (2004), bir "Do it yourself" (DIY), perakendecisinden alınan bir yıllık veriler üzerinde market sepet analizi kullanarak ilişkili ürün çiftlerini tayin etmişlerdir. Çeşitli promosyon çalışmalarının bu tamamlayıcı ürün çiftleri üzerindeki etkileri analiz edilmiş, sonuçlar doğrultusunda gerekli tavsiyelerde bulunulmuştur. Chen vd. (2005), mağazalar zincirine sahip işletmelerde, ürünlerin her an tüm mağaza raflarında olduğu varsayımından dolayı önemli satın alma desenlerinin çıkarılamadığını düşünmüşlerdir. Bu sorunu aşmak için benzer çalışmalardan farklı olarak geliştirilen kurallara depo ve zaman bilgilerini de eklemişlerdir. Böylece birliktelik kurallarını; pazarlama stratejisi, ürün tedarigi, envanter ve tüm mağaza zinciri için dağıtım stratejileri geliştirmesinde kullanmayı hedeflemişlerdir. Yun vd. (2006), geleneksel birliktelik kuralları madenciliğinin çok mağazalı ortamlarda sağlıklı sonuçlar veremeyeceğini düşünmüştür. Bu mağazalardaki birliktelik kurallarını incelemek için Apriori algoritmasına benzer bir algoritma geliştirilmiştir. Geliştirilen algoritma ile üretilen kurallar mağaza konumu ve zaman periyodu hakkında da bilgi vermektedir. Boyut bakımından çeşitlilik gösteren ve sürekli değişen ürün karmasına sahip mağazalarda yapılan simülasyon çalışmasına göre önerilen yöntemin, geleneksel Apriori algoritmasının göre daha iyi sonuç verdiği görülmüştür. Kuo vd. (2007), Tayvan Ulusal Sağlık Sigortası Bürosu ile anlaşarak on iki aylık sağlık sigortası verileri üzerinde, hastalıklar arasındaki ilişki kurallarını bulmak için çalışmışlardır. Çalışmalarında karınca koloni sistemi, kümeleme ve birliktelik kurallarından yararlanmışlardır. Erdem ve Özdağoğlu (2008), Ege Bölgesi'nde bir eğitim ve araştırma hastanesinin acil servis verileri üzerinde birliktelik kural analizi gerçekleştirmişlerdir. Apriori algoritması kullanılarak ortaya çıkarılan birliktelik kurallarına bakıldığında; cinsiyet, kalış zamanı, acil servise geliş zamanı ve bazı hastalık tanılarında birliktelikler olduğu gözlemlenmiştir. Bu çalışma sonucunda o bölgedeki hastaların acil servise başvuru nedenlerinin belirlenmesi ve hasta profillerinin belirlenmesi ile acil servis bölümlerinin yeniden yapılanması açısından yol gösterici olması beklenmektedir. Ergün (2008), perakende sektöründe hizmet veren bir mağazada, ilişkili ürün kategorilerinin belirlenmesi için birliktelik

kuralları ve kümeleme analizi kullanılmıştır. Mağaza ürünleri; gıda dışı, hızlı tüketim ve hazır gıda olarak gruplanmış ve bunların alt kategorileri arası ilişkiler, müşteri fiş verilerinden yararlanılarak analiz edilmiştir. Böylece ilişkili kategoriler belirlenmiştir. Timor ve Şimşek (2008), Türkiye’ de perakende sektöründe faaliyet gösteren, Türkiye’nin en büyük marketler zincirinden birinin 2004 yılındaki dört aylık bir sürenin müşteri alışveriş kayıtları üzerinde birliktelik kuralları analizi yapılarak müşterilerin satın alma davranışlarını etkileyen faktörleri karar ağaçları ile belirlemişlerdir.

Li ve Li (2010), bir telekomünikasyon şirketindeki arızalar arasındaki ilişkileri analiz etmek için birliktelik kuralı madenciliğine dayalı yeni bir alarm korelasyon analizi sistemi önermişlerdir. Chiang (2011), Tayvan ‘daki üç farklı internet alışveriş pazarı üzerinde gerçekleştirdiği çalışmasında; değerli müşterilerini öğrenmek için denetimli apriori algoritmasını kullanmış ve RFMDR modelini geliştirmiştir. Nahar vd. (2011), Mesane, meme, prostat, rahim ağzı, akciğer ve cilt kanserlerine neden olan faktörleri belirlemek için birliktelik kurallarını kullanmışlardır. Bu amaç için kullanılan üç farklı algoritma içinden en iyi sonucu verenin apriori olduğunu görmüşlerdir. Durdu (2012), çalışmasında Pehlivanoglu Marketçilik Gıda Paz. San ve Tic A.Ş. ‘nin gıda kategorisindeki ürünler üzerinde birliktelik kural analizi gerçekleştirmiştir. Apriori algoritması kullanarak ilişkili kategorileri belirlemiştir. Erpolat (2012), çalışmasında birliktelik kuralları analizinde yaygın olarak kullanılan Apriori ve FP-Growth algoritmalarının işleyişi hakkında bilgi vermiş, bu algoritmaları bir otomotiv yetkili servisinin müşteri alışveriş verileri üzerinde uygulayarak sonuçlarını karşılaştırmıştır. Çalışması sonucunda FP-Growth algoritmasının bu büyüklükteki veri setinin analiz edilmesinde Apriori algoritmasına göre daha uygun olduğu sonucuna varmıştır. Çil (2012), birliktelik kuralları ve çok boyutlu ölçekleme analizi ile mağaza yerleşim problemi için yeni bir öneride bulunmuştur. Çalışmasında perakende sektöründe önemli bir yere sahip olan Migros Türk süpermarketinin müşteri fişleri ve ürün barkod bilgilerini veri olarak almış; bunu birliktelik kuralları methoduna göre, Apriori algoritması kullanarak analiz etmiştir. Analiz sonucu çıkardığı kurallara göre yeni bir mağaza yerleşim planı önermiştir. Hanguang ve Yu (2012), birliktelik kuralları

madenciliği ve apriori algoritması ile Web tabanlı saldırı tespit sistemi üzerinde çalışılmışlardır. Güngör vd. (2013), Bilecik Şeyh Edebali Üniversitesi Bilgisayar Mühendisliği Bölümü öğrencilerinin teknik seçmeli ders seçimlerinde hangi kriterlere göre seçim yaptıklarının belirlenmesi amacı ile yapılan anket verileri üzerinde Apriori algoritması kullanılarak birliktelik kural analizleri gerçekleştirilmiştir. Mansur ve Kuncoro (2012), Karomah Brass adlı mobilya aksesuarları satan bir şirkette, market sepet analizi ve yapay sinir ağlarını kullanarak müşteri satın alma davranışlarını incelemişlerdir. Bu çalışma ile şirket için, stok planlamada işe yarar sonuçlar elde etmişlerdir.

Paranjape-Voditel ve Deshpande (2013), birliktelik kuralları madenciliğini borsa sektörü üzerinde uygulamışlardır. Pazar sepet analizi ile portföy öneri sistemi geliştirmişlerdir. Dhanabhakya ve Punithavalli (2013), etkili pazar sepet analizi için FRG-AARM yaklaşımını önermişlerdir. Önerilen bu yaklaşımda geleneksel yaklaşımdan farklı olarak Apriori ve Apriori-Tid algoritmalarının en iyi özellikleri Apriori-Hybrid Algoritması ile birleştirilmiştir. Yapılan simülasyon çalışması ile önerilen model, geleneksel modellerle; öngörme doğruluğu ve ortalama çalışma süresi performansı açısından değerlendirilmiştir. Önerilen yaklaşımın çok kısa bir sürede sonuç verdiği görülmüştür. Nahar vd. (2013), Birliktelik kuralları ile erkeklerde ve kadınlarda kalp krizine karşı önemli risk faktörlerini belirlenmiştir. Ceylan (2014), bir eczanede gerçekleştirdiği çalışmada, hasta reçete verilerini kullanarak ilaçlar arasındaki birliktelik ilişkilerini analiz etmiştir. Analiz sonucunda çıkarılan kurallar ile raf düzenlenmesi gerçekleştirilmiştir. Yeni raf düzeni ile eczacının ilaç arama süresi azalmış böylelikle müşteri memnuniyeti artmıştır. Aynı zamanda bu sistem eczanelerdeki hatalı ilaç verme ihtimallerinin de önüne geçmeyi hedeflemektedir. Doğan vd. (2014), bir sigorta şirketinde müşterilerin satın alma alışkanlıklarının belirlenmesi ile müşterilerin sadakatini ve memnuniyetini arttırmanın karı arttırabileceği düşüncesi üzerinde durmuşlardır. Apriori algoritması ile birliktelik kural analizi sonucunda öne çıkan beş kural belirlenmiştir. Kamley vd. (2014), borsa veri tabanında gerçekleştirdikleri çalışmalarında hisse senetleri arasındaki ilişkilerin

belirlenmesi ile borsacılara ve yatırımcılara maksimum kar getirecek yatırım aracının ve uygun zamanın belirlenmesi hedeflenmiştir.

Kaur ve Kang (2016), birliktelik kural madenciliğini dinamik veriler üzerinde gerçekleştirecek bir algoritma önermişlerdir. Değişimin modellenmesi konseptiyle çalışarak periyodik madencilik gerçekleştirmişlerdir. Solnet vd. (2016), Avustralya'da lüks bir otel grubunun verileri üzerinde market sepet analizi gerçekleştirmişlerdir. Otel misafirleri için en uygun ek hizmetlerin belirlenmesi ile otelin karını arttırılabileceği konusunda çalışmışlardır. Yazgan vd. (2016), çalışmalarında bir ecza deposunda sipariş toplama sürecini iyileştirmek üzerinde çalışmışlardır. Depolanacak ürünlerin hacimsel olarak küçük olduğu bu problemin çözümünde sipariş yığınlama stratejisini önermişlerdir. Müşteri siparişleri arasındaki birliktelik kuralları tespit edilerek farklı ecza depolarında aralarında benzerlik tespit edilen sipariş fişleri aynı yığına denk gelecek şekilde genetik algoritma ile yığınlar oluşturulmuştur. Sipariş toplayıcının bir turda birden fazla siparişi toplaması ile tur sayısı azalmış ve bununla birlikte de zaman tasarrufu sağlanmış olur. Özçalıcı (2017), ikinci el otomobil satış sitesi olan sahibinden.com adresinden aldığı 211,109 adet ilan üzerinde Apriori algoritmasını kullanarak satışa sunulan araçların genel özellikleri belirlenmiştir. Apriori algoritması böyle bir amaç için ilk kez kullanılmıştır. Yıldırım vd. (2017), bir tekstil firmasının verileri üzerinde gerçekleştirdikleri birliktelik kuralları madenciliği çalışması ile iplikler ve kumaş özellikleri arasındaki ilişkileri belirlemeye çalışmışlardır. Birliktelik kurallarının belirlenmesinde FP-Growth algoritması kullanılmıştır.

3. VERİ MADENCİLİĞİ

Günlük hayatta gerçekleştirdiğimiz her türlü işlem, gelişmiş veri tabanı sistemleri sayesinde, sayısal ortamlarda kayıt altına alınmakta ve bu kayıtlar gün geçtikçe büyük veri yığınlarına dönüşmektedir. William ve arkadaşları 1992 yılında gerçekleştirdikleri çalışmaları sonucunda; Dünya üzerindeki veri miktarının her yirmi ayda ikiye katlandığı tahmininde bulunmuşlardır (Frawley vd.,1992). Aylar geçtikçe üstel olarak artan bu veriler büyük veri yığınlarının oluşmasına neden olmaktadır. Bu veri yığınlarının belirli işlemlerden geçirilerek kullanılabilir bilgiye dönüştürülmesi sürecinin tamamı, 1989 yılında gerçekleştirilen KDD (Knowledge Discovery in Databases) çalışması ile 'Veri tabanından bilgi keşfi' olarak ilk kez hayatımıza girmiştir. Veri madenciliği de, belirlenen veri kümesi üzerinde; sınıflama, kümeleme, birliktelik kuralları vb. yöntemler ve bu yöntemlere uygun algoritmaların kullanılarak çeşitli örüntülerin elde edildiği, VTBK (Veri Tabanından Bilgi Keşfi) sürecinin bir aşamasıdır.

Veri madenciliği; elimizdeki büyük veri yığınları içerisinde gizlenmiş, önceden bilinmeyen, sezgi ve tecrübelerimizle tahmin edemediğimiz ayrıca öğrenilmesi durumunda işletmeye büyük yarar sağlayacak değerli bilginin gün yüzüne çıkarılması işlemidir (Silahtaroglu, 2016 ; Özkan, 2016). Bu çalışmanın ana konusu veri madenciliği yöntemlerinden olan birliktelik kurallarının market sepet analizi ile belirlenmesi olduğundan, ilerleyen kısımlarda bu konunun detaylarına yer verilmiştir.

3.1. Veri Madenciliği Temel Kavramları

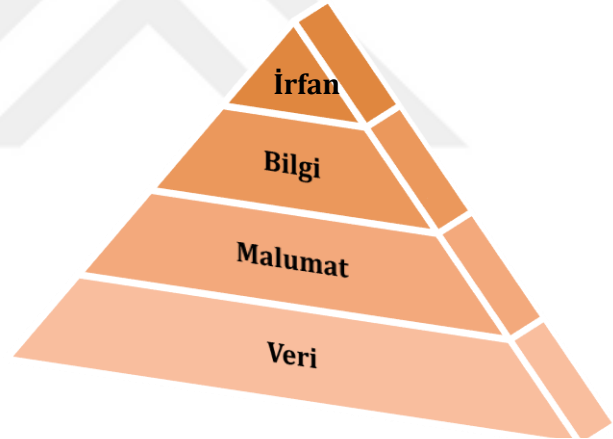
Bu çalışmanın ve ilgili çalışmaların daha iyi anlaşılması açısından veri madenciliğinde sıkça kullanılan veri, malumat, bilgi, irfan, veri tabanı, veri yığını vb. kavramlar açıklanmalıdır.

3.1.1. Temel Kavramlar; Veri, Malumat, Bilgi, İrfan

Veri kelimesi, çoğumuzun zihninde sayısal ifadelerle karşılık bulur. Oysa işlenmemiş her türlü anlamlı sözcük, sayı veya nota birer veridir. Verilerin bilgisayarın anlayabileceği şekle dönüştürülüp kaydedilmesi, daha sonra çeşitli veri madenciliği yöntemleri uygulanmasıyla ulaşılan değer, bilgidir. Örneğin notalar veri, müzik parçası ise bilgidir. Sözcükler veri, cümleler ile paragraflar bilgidir. Büyük hacimdeki veriler içerisinden çıkarılan, çok daha küçük hacimli bilgi, karar vermede kullanılan önemli bir araçtır. Buna karşılık büyük hacimli verilerin karar vermede kullanım değeri daha düşüktür (Gürsaka, 2001).

Veri madenciliği alanındaki İngilizce çalışmalarda sıkça karşılaştığımız; 'Data', 'Information', 'Knowledge', 'Wisdom' terimleri, Türkçe tercümelerinde genellikle 'Bilgi' olarak çevrilmektedir. Şekil 3.1'de terimler arasındaki hiyerarşi gösterilmektedir. Terimlerin doğru tercümei ise aşağıdaki gibidir (Akpınar, 2014);

- Data: Veri
- Information: Malumat
- Knowledge: Bilgi
- Wisdom: İrfan, Bilgelik



Şekil 3.1. Bilgi Hiyerarşisi

Yukarıda, terimler aynı zamanda mertebelerine göre de sıralanmıştır. Veri işlenerek malumata dönüşür, malumat işlenerek bilgiye, bilgi de işlenerek aralarında en değerli sonuç olan irfana dönüşür. Bunu somut bir örnekle açıklayalım: Bir aracın hızını veri olarak alalım. Bu hızın anlamı, mesela tehlikeli bir hız veya randevumuza geç kalmamıza sebep olacak bir hız olması malumat olur. Hızın arttırılması, azaltılması ve geçmiş tecrübeler ile karşılaştırılması eylemlerinin sonucu bilgi olarak kabul edilebilir. Şoförlük bilgisi de irfan olarak kabul edilebilir. Bu örnek sadece kavramlar arasındaki ilişkinin daha rahat

anlaşılması içindir. Bu kavramları doğrusal bir yapıya oturtmak yanlış olacaktır (Şekeroğlu, 2010 ; Şeker, 2013).

3.1.2. Veri yığını

Ergün E., henüz işlem görmemiş, taşınabilen, güncellenebilen, yönetilebilen veri topluluğunu 'veri yığını' olarak tanımlamıştır(Ergün, 2008).

Veri yığınları henüz hiçbir işlem görmedikleri için fazlaca gürültülü veri içerebilirler. Bu sebeple veri yığınlarına direkt olarak veri madenciliği teknikleri uygulanamaz. Veri madenciliği süreci öncelikle bu veri yığınlarındaki amaca uygun verileri seçmek ve gereksiz verileri temizlemek ile başlar. Eğer verilerdeki gürültü miktarı çok fazla ise gerekli veri madenciliği teknikleri uygulansa dahi istenilen sonuç alınamayabilir (Ergün, 2008).

3.1.3. Veri tabanı

Veri tabanı (database) , ilişkili verilerin tutulduğu , kullanıcıya sistematik erişim imkanı sunan ,yönetilebilen , güncellenebilen, istenilen yere taşınabilen ,kullanım amaçlarına uygun bir şekilde düzenlenmiş bilgi depolarıdır (Döşlü, 2008).

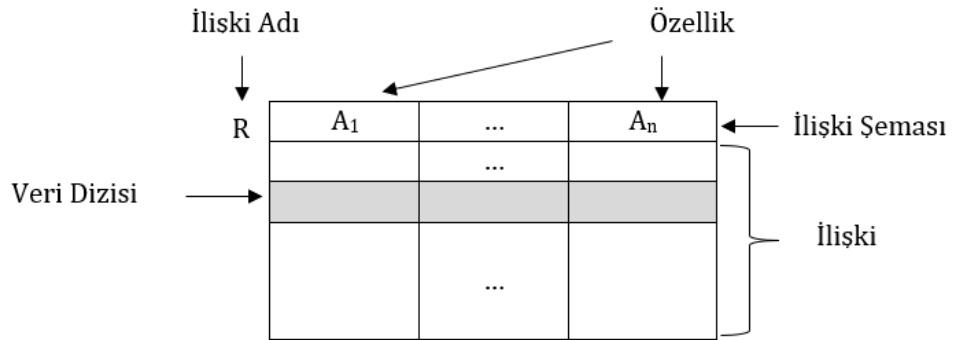
Veri tabanı; süpermarkelerden hastanelere, bankalardan devlet dairelerine kadar birçok kuruluşun sorunsuz bir şekilde işleyebilmesi için gereken uygulama ve programların kullanıldığı, çeşitli işlemsel verilerin saklandığı yerdir. Bir süpermarketin veri tabanı barkod kodu, müşteri numarası, kasa numarası vb. gibi bilgileri içerirken, bir hastane veri tabanı hasta adı, hasta yaşı, doktoru vb. gibi bilgileri içermektedir. Veri tabanları, güncel verilerden oluşmaları ve kullanıcının istediği anda uygun sorgularla hedeflenen sonuçlara ulaşılabilmesi avantajı ile kurumlara rekabet üstünlüğü sağlar. Ayrıca işletme içerisinde ortak bir veri tabanı kullanıldığı için her departman istediği güncel veriye kendi sisteminden kolaylıkla ulaşabilir.

İşlemsel veri tabanları:

Dinamik bir yapıya sahiptirler, yani her birim işlemde kendilerini güncellerler ve sürekli değişirler. Her veri girişinde güncellenmesi, gerektiğinde silme ve veri ekleme imkanı sunması ile birçok kuruluşun ilk tercihi olma sebeplerindedir.

İlişkisel veri tabanları:

Bu tür veri tabanları bir tablo grubundan oluşurlar. Modelde ilişkiler tablolarla temsil edilir. Veriler konuya ya da görevlerine göre bu tablolarda depolanırlar. Her tablo yalnızca tek bir konu veya görev özelliklerini içerir. İlişkili tablolar birbirine bağlanarak birbirleri ile ilişkilendirilir. Şekil 3.2, ilişkisel veri modelini göstermektedir.



Şekil 3.2.İlişkisel veri modeli (Hoche vd., 2006)

Şekil 3.2’de görülebileceği gibi bahsedilen tablolar satır ve sütunlardan oluşur. Tablonun bir satırına, *tuple* adı verilir. Tüm tupleler bir araya gelerek tablonun gövde kısmını oluştururlar. İlişki, gövde kısmından çıkarılmaktadır. Her özellik farklı bir sütunda belirtilir. Belirtilen özelliğin ismi tablo başlığında verilir. Sütun girdilerini, özelliğin değerleri oluşturur ve bu şekilde depolanır. İlişkilendirme sayesinde bir tabloya ait bilgiler, anahtar sütundan hareketle başka bir tablodan takip edilebilir (Hoche vd., 2006) .

Tablolar arasındaki ilişkiler üç şekilde tanımlanabilir;

Bire-Bir İlişki: İlişkilendirilmek istenilen tablolar arasında yalnızca tek bir kaydın eşleştirilebildiği ilişki türüdür.

Bire-Çok İlişki: Birinci tablodan bir kaydın, ikinci tablodan birçok kayıtle eşleştirilebildiği ilişki şeklindedir. En fazla tercih edilen ilişkilendirme şeklindedir.

Çok-Çok İlişki: Bire-çok ilişkilendirme şeklinin çift yönlü işlediği ilişkilendirme şeklindedir. İki tablo arasındaki ilişkiyi oluşturmak için bu iki tablo ile de ilişkili üçüncü bir tabloya ihtiyaç duyulur.

Veri Tabanı Yönetim Sistemi:

Veri tabanlarındaki veriler üzerinde merkezi bir denetim olduğundan, veri tabanı kullanıcıları genel amaçlı programlama dilleri veya işletim sistemi komutları ile yazılan uygulama programlarını kullanarak doğrudan veri tabanındaki verilere erişemezler. Veri tabanındaki veriler üzerinde herhangi bir işlem yapamazlar. Veri tabanı kullanımı için gerekli olan Veri Tabanı Yönetim Sistemi-VTYS (Data Base Management System) olarak adlandırılan yazılım sistemleridir (Yarımağan, 2010).

VTYS yazılımları en pahalı yazılım programlarından biridir. Buna rağmen, kurumlara getirisinin yanında maliyeti çoğu işletme tarafından, katlanmaya değer bir maliyet olarak değerlendirilmektedir. Günümüzde verinin kullanıldığı alanlardaki uygulamalar, veri tabanı yönetim sistemi olmadan kullanılamaz hale gelmiştir.

VTYS 'leri, sayesinde veri tabanlarına yeni tablo eklenebilir, veri tabanındaki tablolar üzerinde silme ve güncelleme işlemleri gerçekleştirilebilir. Veriler yedeklenebilir. Yine bu sistem üzerinden gerekli sorgular yapılarak raporlar oluşturulabilir ve analizler yapılabilir.

Özkan Y., VTYS 'ni kısaca; kümelenmiş verilerin düzenli olarak tutulduğu ve bu verilerin çeşitli yazılımlar aracılığı ile yönetilebildiği ortam olarak tanımlamıştır (Özkan, 2016) .

Veri tabanları günümüzde yaygın olarak iki amaç için kullanılır. Bunlardan ilki, optimize edilmiş sistemler (Online Transaction Processing) ; literatürde kısaca OLTP olarak geçen ve işletmelerin operasyonel süreçlerine dahil olan işleri iyi bir şekilde yürütülmesinde katkı sağlayan , birçok kullanıcı tarafından eş zamanlı olarak ekleme, silme ,güncelleme işlemlerinin yapılabildiği veri tabanı türüdür (Gözcü, 2015).

İkincisi ise bu çalışmanın konusu olan 'Veri Madenciliği' için, büyük öneme sahip Çevrimiçi Analitik İşleme (Online Analytical Processing-OLAP) adıyla bilinir. OLAP, daha çok raporlama ve karar destek amacı ile kullanılan bir veri tabanı yapısıdır.

İlk kez 1993 yılında, Dr. E. F. Codd ve arkadaşları tarafından, kullanıcıya dinamik veri analizi gerçekleştirme imkanı veren bir mimari olarak geliştirilmiştir. OLAP araçları, genellikle çok boyutlu veri analizi gerçekleştirmek için kullanılır. Bu araçlar, etkileşimli veri analizini basitleştirmeye ve desteklemeye yöneliktir. OLAP araçları, kullanıcıların kurumsal analitik veri tabanlarına erişimini sağlar (Fayyad vd., 1996). Böylece kurumsal değişikliklerde oluşabilecek problemlere karşı taktik ve operasyonel kararlar üretilebilir.

Silahtaroglu G. OLAP teknolojisini, bilgisayar üzerinde akıl yürüterek işlem yapma olarak tanımlamıştır.

OLAP teknolojisi sayesinde veriler yalnızca ihtiyaç anında değil sürekli olarak depolanır ve ihtiyaç anında tüm veriler üzerinden ya da gerekli olan kısmından analiz için yararlanılabilir. Ayrıca OLAP teknolojisi 'Kim' , 'Ne Zaman' soruları ile birlikte 'Neden' ve 'Eğer şöyle olursa ..' sorularına da yanıt verebildiği için ileriye dönük karar vermede de şirketler için oldukça faydalı olabilmektedir (Döşlü, 2008).

OLTP ve OLAP arasındaki başlıca farklar, Han ve Kamber tarafından Çizelge 3.1 ile gösterildiği şekilde özetlenmiştir(Han ve Kamber, 2006).

Çizelge 3.1. OLTP ve OLAP sistemlerinin karşılaştırılması(Han ve Kamber, 2006)

Özellik	OLTP	OLAP
Nitelik	Hareketsel işleme	Bilgisel işleme
Yönelim	Hareket	Analiz
Kullanıcı	Tezgahtar, DBA, Veri tabanı profesyoneli	Bilgi çalışanı, yönetici, analist
Fonksiyon	Güncül hareketler	Uzun terimli bilgi gereksinimleri, karar destek
Veri tabanı tasarımı	E-R tabanlı, uygulamaya dayalı	Yıldız/kar tanesi, özneye dayalı
Veri	Güncel, güncellik garantisi	Tarihsel kesinlik zamanla sağlanır
Özetleme	İlkel, çok detaylı	Özetlenmiş, birleştirilmiş
Görünüm	Detaylı, düz ilişki	Özetlenmiş, çok boyutlu
Çalışma Birimi	Kısa, basit hareketler	Karmaşık sorgu
Erişim	Oku/Yaz	Çoğunlukla oku
Odaklanma	Gelen veri	Çıkan bilgi
İşlemler	Birincil anahtardaki indeks/hash	Çoklu tarama
Erişilen Kayıt Sayısı	Onlarca	Milyonlarca
Kullanıcı Sayısı	Binlerce	Yüzlerce
Veri tabanı Boyutu	100 MB'tan GB'ya kadar	100 GB'tan TB'ya kadar
Öncelik	Yüksek başarı, yüksek kullanılabilirlik	Yüksek esneklik, sonkullanıcı yönetimi
Ölçüm	Üretilen hareketsel iş	Üretilen sorgusal iş, cevap süresi

3.1.4. Veri ambarı

Veri ambarları, veri madenciliği sürecinin gerçekleştirilebilmesi için uygun verileri hazırlayan özel bir veri tabanıdır. Farklı kaynaklardan gelen bu veriler genellikle farklı yapıdadırlar. Bu verilerin, aynı amaç için kullanılması ümidiyle oluşturulan topluluklar veri ambarlarıdır. Veri ambarları sayesinde farklı kaynaklardan gelen bu farklı yapıdaki veriler, tek bir çatı altında analiz edilebilir (Fayyad vd., 1996).

Veri ambarı, temelde organizasyonel kararlarda kullanılan, konu yönelimli, birleştirilmiş, zaman değişimli, kararlı veri depolarıdır. Veri ambarının konu yönelimli olması; belli konular için düzenlenmiş olması demektir, birleştirilmiş olması; farklı kaynaklardan gelen veriler olsalar dahi aralarında tutarlı bir ilişki

olduğunu , zaman değişkenli olması ; ortalama 5-10 yıllık verileri depolayabilme özelliğinden gelmektedir (Şentürk , 2006).

Veri ambarları için önemli olan DataMart ve MetaData kavramlarını da tanımlayalım.

DataMart: Yaklaşık 1-10 GB kapasiteli birimsel ambarlardır. İşletmelerin belirli kullanıcılarına ait verileri içeren, veri ambarlarının alt kümeleridir.

MetaData: Veri ambarlarındaki kayıtlı veriler hakkında bilgilerin verildiği kısımdır.

Veri ambarları operasyonlar için kritik öneme sahip olan analizler ve sorgular gerçekleştirilebilmesi için tasarlanmış ve optimize edilmiştir (Imhoff vd. , 2003). Genellikle 6 saat aralıkla, güncel veri tabanları taranır ve veri tabanlarında olan herhangi bir değişme veri ambarına aktarılır. Dolayısıyla güncel bilgiye veri ambarlarından ulaşılamaz. Çizelge 3.2 de veri tabanı ve veri ambarı bazı açılardan karşılaştırılması görülmektedir.

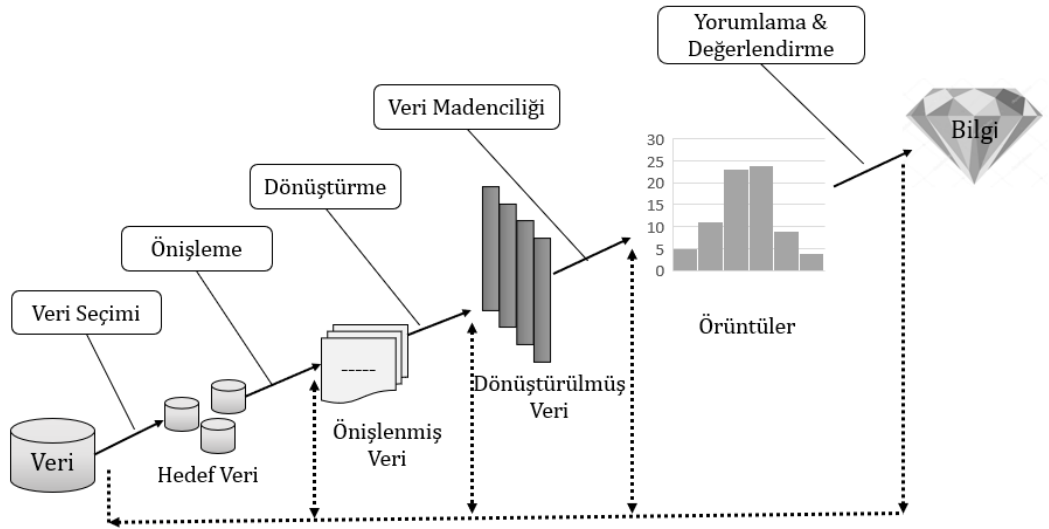
Çizelge 3.2. Veri tabanı /Veri ambarı karşılaştırması (Anon ,2018)

Veri tabanı	Veri ambarı
Genellikle sık yapılan güncellemelere bağlı olarak değişir. Bu nedenle analiz ya da karara varmak için kullanılmaz.	Verileri çıkarır, analiz edilmesi ve karara varılması için raporlar.
Çeşitli sorgularla istenilen sonuçlara hızlı bir şekilde ulaşılabilir.	Veri madenciliği gibi uzun ve karmaşık süreçler sonunda analizler yapılabilmektedir.
Detaylı verileri içerir.	Özet verilerden oluşur.
Son kullanıcı sayısı fazladır.	Son kullanıcı sayısı azdır (<100).
Çevrimiçi çalışır.	Çevrimdışı çalışır.
Kullanıcıya her aşamada, ihtiyacı olan veriye erişim imkanı sağlar.	Üst yönetime hitap eden karar destek sistemleridir.

3.2. Verinin Yararlı Bilgiye Dönüşüm Süreci

Günlük hayatta gerçekleştirdiğimiz her türlü işlem, gelişmiş veri tabanı sistemleri sayesinde, sayısal ortamlarda kayıt altına alınmaktadır. Bu kayıtlar gün geçtikçe büyük veri yığınlarına dönüşmektedir. William ve arkadaşları 1992 yılında Al Magazine dergisinde yayımlanan çalışmalarında; Dünya üzerindeki veri miktarının her yirmi ayda ikiye katlandığı tahmininde bulunmuşlardır (Frawley vd. ,1992). Aylar geçtikçe üstel olarak artan bu veriler büyük veri yığınlarının oluşmasına neden olmuştur. Bu veri yığınlarının belirli işlemlerden geçirilerek kullanılabilir bilgiye dönüştürülmesi sürecinin tamamı, ilk kez 1989 yılında gerçekleştirilen KDD çalışması ile 'Veri tabananından bilgi keşfi' olarak hayatımıza girmiştir.

Ham verinin belirli adımlardan geçirilerek bilgiye dönüştürüldüğü bu sürecin adımları, Şekil 3.3'te gösterilmiştir.



Şekil 3.3. VTBK sürecinin aşamaları (Fayyad vd., 1996)

Şekil 3.3' de görüldüğü gibi birçok araştırmacının VTBK ile aynı anlamda kullandığı 'veri madenciliği', bu sürecin bir adımıdır. Sürecin başlıca adımlarını şu şekilde açıklayabiliriz:

Veri Seçimi (Data Selection): Belirlenen hedef doğrultusunda, amaca ulaşmak için uygun olan verinin seçildiği adımdır. Sonraki aşamalar, bu aşamada belirlenmiş olan veri kümesi üzerinden gerçekleştirilir.

Ön İşleme (Data Preprocessing): VTBK süreci sonunda ulaşılabilecek bilginin güvenilirliği, başlangıçtaki verinin kalitesine bağlıdır. Başlangıçta seçilen veri kümesinin kalitesini arttırmak için ön işleme aşamasında küme içerisindeki hatalı, tutarsız veya eksik veriler belirlenir ve gürültü olarak ifade edilen bu olumsuzluklar düzeltilir. Veri temizleme işlemi de bu adımda gerçekleştirilir.

Dönüştürme (Transformation): Ön işleme aşamasında temizlenmiş verinin uygun dönüşüm işlemleri sonucunda veri madenciliği için uygun formata dönüştürüldüğü aşamadır. Doğru indirgeme işlemi, veri madenciliği işleminin çalışma süresinin iyileşmesine katkı sağlar.

Veri Madenciliği (Data Mining): Belirlenen veri kümesi üzerinde; sınıflama, kümeleme, birliktelik kuralları vb. yöntemler ve bu yöntemlere uygun algoritmaların kullanılarak çeşitli örüntülerin elde edildiği aşamadır.

Yorumlama (Interpretation): Veri madenciliği aşaması sonucu çıkarılan örüntünün yorumlandığı aşamadır. Bu örüntü ile birlikte keşfedilen bilginin yararlılığı ve geçerliliği bu aşamada değerlendirilir.

3.3. Veri Madenciliğinin Tanımı

Veri madenciliği; elimizdeki büyük veri yığınları içerisinde gizlenmiş, önceden bilinmeyen, sezgi ve tecrübelerimizle tahmin edemediğimiz ayrıca öğrenilmesi durumunda işletmeye büyük yarar sağlayacak değerli bilginin gün yüzüne çıkarılması işlemidir (Özkan, 2016; Silahtaroglu, 2016).

Literatüre bakıldığında, veri madenciliği için kalıplaşmış tek bir tanımın olmayıp, farklı birçok tanımın yapıldığı görülmüştür. Bu tanımlardan bazılarını aşağıda yer verilmiştir.

Verideki gizli, önceden bilinmeyen ve potansiyel olarak faydalı enformasyonun önemsiz olmayanlarının açığa çıkarılmasıdır (Frawley vd., 1992).

Veri madenciliği, geniş veri tabanlarından bilgiyi çıkarmak için; makine öğrenimi, örüntü tanıma, istatistik, veri tabanı ve görüntüleme tekniklerini bir araya getiren disiplinler arası alandır (Cabena vd., 1998).

Veri madenciliği, tek başına ham verinin sunamadığı bilgiyi ortaya çıkaran veri analizi sürecidir (Jacobs, 1999).

Veri madenciliği, oldukça tahminci anahtar değişkenlerin binlerce potansiyel değişkenden izole edilmesini sağlama yeteneğidir (Kittler ve Wang, 1999).

Veri madenciliği, büyük miktardaki veri setlerinde saklı durumda bulunan örüntü ve eğilimleri keşfetme işlemidir (Özekes ve Çamurcu, 2003).

Veri madenciliği, önceden bilinmeyen, veri içinde gizli, anlamlı ve yararlı örüntülerin büyük ölçekli veri tabanlarından otomatik biçimde elde edilmesini sağlayan veri tabanlarında bilgi keşfi süreci içinde bir adımdır (Han ve Kamber, 2006).

3.4. Veri Madenciliği Tarihsel Gelişimi

Veri madenciliğinin, bilişim teknolojilerindeki doğal gelişmenin bir sonucu olarak ortaya çıktığı söylenebilir (Han ve Kamber, 2006). Bilişim teknolojilerindeki gelişimi ise şu şekilde özetleyebiliriz:

1950'li yıllarda bilgisayar yalnızca basit sayma işlemleri için kullanılıyordu. Bilgisayarların veri analiz problemlerinin çözümü için kullanılmaya başlandığı yıllar olan 1960'lı yıllarda, veri madenciliği kavramsal olarak ortaya çıkmaya başlamıştır. İstenilen bilgiye çeşitli sorgularla bilgisayarlardan ulaşılabileceği düşünülmüştür. Ancak o yıllarda veri madenciliği ismi yerine, veri yakalaması, veri taraması gibi isimler kullanılmıştır (Köktürk vd. , 2009).

1970'lerde ilişkisel veri tabanı yönetim sistemleri uygulamaları kullanılmaya başlanmıştır. Böylece hiyerarşik veri tabanı yapılarından ilişkisel veri tabanı, veri modelleme araçları ve indeksleme yapısına geçişi sağlanmıştır. Kullanıcıların, sorgulama dilleri ve arayüzler sayesinde veriye erişim imkanı artmıştır (Kocabaş, 2010). 1980'lerde veri tabanı yönetim sistemleri yaygınlaşarak işletmelerde kullanılmaya başlamıştır. Şirketlerin müşterileri, rakipleri ve ürünleri ile ilgili bilgileri içeren veri tabanları oluşturulmuştur. Büyük miktarda veri içeren bu veri tabanlarına ulaşabilmek için SQL ve benzeri sorgulama dilleri kullanılmıştır (Savaş vd., 2012)

1990'larda bilgisayar mühendisleri, geleneksel istatistiksel yöntemler yerine algoritmik bilgisayar modülleri ile veri analizinin gerçekleştirilebileceğini vurgulamış ve veri madenciliği ismini kullanmışlardır (Savaş vd. ,2012) . 1990'lı yıllarda teknolojinin gelişmesi ile birlikte artan veri miktarı sonucunda genişleyen veri tabanlarından, bilgiye nasıl ulaşılabileceği düşüncesi, üzerinde durulmuş ve bu konuda çalışmalar yaygınlaşmıştır. 1991 yılındaki "Knowledge Discovery in Real Databases: A Report on the IJCAI89 Workshop" başlıklı makalesinin bilgi keşfi ve veri madenciliği ile ilgili temel tanım ve kavramları ortaya koyması ile süreç daha da hızlanmıştır. 1992 yılında veri madenciliği için ilk yazılım gerçekleştirilmiştir.

2000'li yılların başlarından günümüze kadar veri madenciliği birçok alanda kullanılmaya başlanmıştır. Çizelge 3.3'te veri madenciliğinin tarihsel gelişimi yer almaktadır.

Çizelge 3.3. Veri madenciliği kavramının tarihsel gelişimi (Kocabaş , 2010)

Gelişim Adımları	Cevap Aranılan Sorular	Kullanılan Teknolojiler	Özellikler
Veri Toplama (1960'lar)	"Şirketin son 3 yıldaki satışları toplamı ne kadar?"	Bilgisayarlar, Teypler, Diskler	Geçmişe dönük, Statik veri dağılımı
Veri Erişimi (1980'ler)	"Ankara'dan geçen Ocak ayında birim satışları ne kadar?"	İlişkisel veri tabanları, SQL, ODBC	Kayıt bazında geçmişe dönük dinamik veri dağılımı
Veri Ambarları ve Karar Destek Sistemleri (1990'lar)	"Ankara'da geçen Ocak ayında birim satışlarının bir önceki yılın Ocak ayındaki birim satış oranı kaç?"	OLEP, Çok Boyutlu Veri tabanı Sistemleri, Veri Ambarları	Çoklu bazda geçmişe dönük dinamik veri dağılımı
Veri Madenciliği (Bugün)	"Gelecek ay Ankara'da satışlar ne kadar olabilir, neden?"	İleri düzeyli algoritmalar, Çok işlemcili bilgisayarlar, Gelişmiş veri tabanları	Geleceğe dönük, öngörülü bilgi dağılımı

3.5. Veri Madenciliği Uygulama Alanları

Literatür özetleri kısmında da değinildiği gibi veri madenciliği günümüzde çok farklı alanlarda uygulanmaktadır. Pazarlama, finans, e-ticaret, sağlık ve sosyal medya alanlarında veri madenciliği uygulamalarının birçok örneğini görebiliriz. Bu alanlardaki başlıca örnekleri şu şekilde sıralayabiliriz (Özkan, 2016) .

Pazarlama

- Müşterilerin satın alma alışkanlıklarının belirlenmesi
- Kaybedilen müşterilerin analizi
- Ürünlerin birlikte satın alınma eğilimlerinin belirlenmesi
- Satışlardaki normal olmayan gelişmelerin saptanması
- Müşteri segmentasyonu
- Müşteri memnuniyeti araştırması
- Satış tahmini

Finans

- Sigorta alanındaki dolandırıcılıkların tespit edilmesi

- Kredi kartı dolandırıcılıklarının tespiti
- Kredi kartı harcama bilgilerinden yararlanarak müşteri gruplarını belirleme
- Kredi taleplerinin değerlendirilmesi
- Personel kayıp analizi
- Finansal göstergeler arasında gizli ilişkilerin belirlenmesi
- Riskli müşteri gruplarının belirlenmesi

Elektronik Ticaret

- Elektronik mağazalar için saldırıların tespiti
- Web sayfalarına yapılan ziyaretlerin çözümlenmesi
- Web sayfalarına yapılan saldırıların belirlenmesi
- Müşteri görüşlerinin çözümlenmesi
- Müşteri memnuniyet araştırması
- Okuyucu görüş analizi

Sağlık

- DNA mikrodizi analizleri
- Hastalık teşhisi
- İlaç analizi
- İlaç yan etki analizi
- Sağlıkta dolandırıcılık analizleri

Sosyal Medya

- Kutuplaşma analizleri
- Duygu analizleri
- Politik gelişmelerin analizi
- Seçim kampanyalarının analizi
- Seçim sonuçları öngörülü çalışma
- Politikacılar popülerlik araştırması
- Sanatçılar popülerlik araştırması
- TV yayınları popülerlik araştırması
- İnternet gazetelerindeki okuyucu görüşleri

- Elektronik mağazalardaki tüketici görüşleri
- Tüketici memnuniyet araştırması

3.7. Veri Madenciliği Modelleri

Literatürde veri madenciliği modelleri; denetimli (supervised) ve denetimsiz (unsupervised) olmak üzere iki farklı kategoride incelenir (Koyuncugil ve Özgülbaş, 2009). Denetimli öğrenme, örnekten öğrenme şeklinde de düşünülebilir. Bu öğrenme şeklinde, denetçi tarafından önceden belirlenen kriterlere göre sınıflar ayrılır, her sınıfa ilişkin örnekler verilir. Buradaki amaç, örneklerden yola çıkarak her bir sistemin kendi içerisindeki kuralını belirlemek ve daha sonraki örneklerin sınıfını bu kurallar sayesinde belirlemektir. Denetimsiz öğrenmede, denetimli öğrenmenin tersine önceden belirlenmiş sınıflar olmadan verilerden model üretilir. Denetimli ve denetimsiz öğrenme arasındaki temel fark, denetimsiz öğrenimde eğitim sürecinin bulunmaması yani verilerin bir model oluşturmak için eğitilmemesidir.

Geleneksel olarak veri madenciliği modelleri 3 ana başlık altında toplanmaktadır.

- Sınıflandırma ve Regresyon (Classification and Regression)
- Kümeleme (Clustering)
- Birliktelik Kuralları (Association rules) ve Ardışık Zamanlı Örüntüler (Sequential patterns)

Bu modellerden; sınıflandırma, denetimli öğrenim kategorisinde iken kümeleme, birliktelik kuralları ve ardışık zamanlı örüntüler denetimsiz öğrenme modelleri kategorisinde yer almaktadır (Hastie vd. ,2004) .

3.7.1. Sınıflama ve Regresyon

Sınıflama ve regresyon modelleri, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen veri analiz yöntemleridir. Örneğin; bir grup sosyal medya kullanıcısı arasındaki kutuplaşma ölçülürken kullanıcıların 'pozitif' veya 'negatif' biçimdeki kategorik değerlerin tahmin edilmesinde sınıflama kullanılır. Regresyon modeli ise yaşı ve geliri verilen potansiyel müşterinin bir bilgisayar ürününü alırken yapacağı harcama miktarını tahmin etmek için kurulabilir (Han ve Kamber, 2006).

Sınıflama ve regresyon modellerinde en çok kullanılan yöntemler şunlardır (Akpınar, 2000);

1. Yapay Sinir Ağları (Artificial Neural Networks)
2. Genetik Algoritmalar (Genetic Algorithms)
3. K- En Yakın Komşu (K- Nearest Neighbour)
4. Naive - Bayes sınıflayıcısı
5. Karar Ağaçları (Decision Trees)
6. Lojistik Regresyon'dur.

3.7.1.1. Yapay Sinir Ağları (Artificial Neural Networks)

Yapay sinir ağı, insan beyninin soyut bir hesaplama modelidir. İnsan beyninde 'nöron' olarak isimlendirilen, yaklaşık 10^{11} küçük birim vardır. Bu nöronların 10^{15} bağlantı ile birbirine bağlandığı tahmin edilmektedir. İnsan beynine benzer bir şekilde bir yapay sinir ağı, yapay nöronlardan ve ara bağlantılardan oluşmaktadır (Kantardzic, 2002).

Genel olarak yapay sinir ağları, gelişmiş matematiksel yapıların hesaplanmasını içeren, beynin bir işlevini yerini getirmek için tasarlanmış sistemlerdir. Bu metot biyolojik sinir ağlarından esinlenerek geliştirilmiştir. Yapay sinir ağları,

katmanlar şeklinde düzenlenir ve ağırlıklı bağlantılar denilen tek yönlü iletişim kanalları ile birbirleri ile haberleşirler.

Yapay sinir ağlarını oluşturan nöronların *aktivasyon veya aktivasyon* seviyesi denilen bir iç hali vardır. Aktivasyon seviyesi, gelen girdileri tanımlayan bir fonksiyondur. Yapay sinir ağındaki nöronlar, insan beyinde olduğu gibi diğer nöronlara sinyal gönderirler. Sinir ağlarında, nöron olarak isimlendirilen düğümlerin birden fazla girdisi olabilir ancak ağdaki diğer düğümlere gönderebileceği tek bir çıktısı olmalıdır.

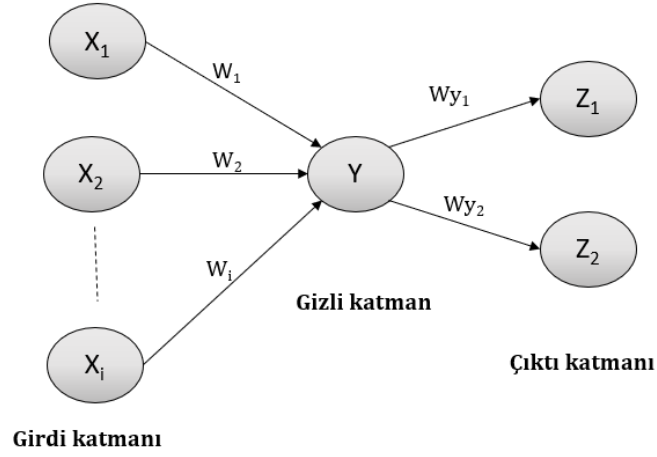
Her nöron tek seferde yalnızca bir tane sinyal gönderebilir. Bir nöron bir seferde birden fazla nörona sinyal gönderebilir. Nöronların gönderdikleri bu sinyaller, sinyali alan nöron için giriş fonksiyonu özelliği taşırlar. Örneğin gizli katmandaki bir Y nöronu için girdi fonksiyonu, (3.1) eşliğinde gösterildiği gibi; Y nöronuna genel sinyallerle (x_i), bir önceki katmanlar ile olan bağlantılardaki ağırlıkların (w_i) çarpımının toplamıdır (Silahtaroglu, 2016).

$$y - girdi = w_1x_1 + w_2x_2 + \dots + w_ix_i \quad (3.1)$$

Y nöronunun etkin hale gelebilmesi için y-girdi değerinin bir fonksiyon eşliğinde belli bir eşik değerine ulaşması gerekir.

Yapay sinir ağları iki ya da üç katmandan oluşurlar. Üç katmanlı yapay sinir ağı; girdi katmanı, gizli katman ve çıktı katmanından oluşur. İki katmanlı olan sinir ağında gizli katman olmaz. Basit şekliyle her bir nöron bir önceki katmanlardaki diğer nöronlara, sinaptik ağırlıkları yoluyla bağlanmaktadır (Kalogirou, 2000).

İnsan beyinde öğrenme, nöronlar arasındaki sinaptik bağlantıların ayarlanması ile gerçekleşmektedir. Yapay sinir ağlarında ise eğitime algoritması girdi ve çıktı verilerini kullanarak bağlantı ağırlıklarında bir yakınsama sağlanıncaya kadar, tekrarlı bir şekilde ayarlanması ile gerçekleşir. Üç katmanlı yapay sinir ağı modeli Şekil 3.4'te gösterilmiştir.



Şekil 3.4. Basit bir üç katmanlı yapay sinir ağı modeli

Yapay sinir ağlarının güçlü yönleri şu şekilde sıralanabilir (Oğuzlar , 2004);

- Çok sayıda gürültülü veri içeren veri kümelerinde bile iyi sonuçlar verebilmesi,
- Hem sayısal hem de kategorik çıktılarının alınmasına ve tahmin edilmesine imkân sağlıyor olması,
- Veri kümesinde zaman faktörünün gerekli olduğu uygulamalarda da kullanılabilmesi,
- Farklı alanlardaki çalışmalara kolay entegre olabilmesi.

Yapay sinir ağlarının zayıf yönleri ise şu şekilde sıralanabilir.

- Yapay sinir ağlarının davranışlarının açıklanmasındaki eksiklik, sinir ağlarının bu konuda olumsuz eleştirilere maruz kalmasına neden olmaktadır.
- Yapay sinir ağlarında kullanılan öğrenim algoritmaları sonucunda, optimal sonuca ulaşıp ulaşılmadığı kesin olarak bilinmemektedir.
- Sinir ağları test verileri üzerinde çalıştırıldığında, eğitim verileri üzerinde çalıştığı kadar başarılı olamamaktadır.

3.7.1.2. Genetik Algoritmalar (Genetic Algorithms)

Genetik algoritma tekniđi 1970'li yıllarda John Holland ve arkadaşlarının liderliđindeki çalışmalar sonucu ilk kez Michigan Üniversitesi'nde ortaya çıkmıştır.1975'te Holland," Doğal ve Yapay Sinirlerin Uygulanması" adlı kitabını yayımlamıştır. Mekanik öğrenme konusunda çalışan Holland, Darwin 'in evrim kavramından etkilenererek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirilmeyi düşünmüştür ve genetik algoritma bu düşüncenin bir sonucu olarak ortaya çıkmıştır. Genetik algoritmalar evrimsel hesaplamalarda en fazla kullanılan tekniktir denilebilir. Bu algoritmalar doğal süreci gözlemleyerek canlıların yapılarında var olan birtakım özellikleri sanal ortamlarda taklit ederek modeller geliştirir. Geliştirilen modellerle karşılaşılan problemlere çözüm bulmaya çalışılır. Bir optimizasyon probleminin karar değişkenleri, doğal genetik sistemlerde kromozomlara benzeyen bir veya daha fazla dizinin yapısına göre kodlanır. Kodlama dizeleri, genlere benzer özelliklerden oluşur. Özellikler dizede farklı konumlarda bulunur. Her bir özellik önerilen kodlama yöntemi ile uyumlu olan kendi konumunun ve kesin bir alel değerinin bulunduğu yerdir(Kantardzic, 2002). Kromozomlardaki dize yapıları daha iyi alternatif değerler üretebilmek için doğal seleksiyon sürecine uygun olarak farklı işlemlerden geçer.

Algoritmada ilk olarak olası çözümlerin kodlandığı, popülasyon olarak adlandırılan bir çözüm grubu oluşturulur. Çözümlerin kodları kromozom olarak adlandırılır. Bu adımda popülasyondaki birey sayısı belirlenmiş olur. Sonrasında eldeki jenerasyon kullanılarak yeni bir jenerasyon elde edilir. Beklenen, yeni jenerasyonun öncekinden daha iyi olmasıdır. İstenilen durma kriterine ulaşıncaya kadar yeni jenerasyonların üretimine devam edilir (Silahtarođlu, 2016).

Veri tabanındaki her kayıt kromozom ifadesine karşılık gelmektedir. Kromozomlar üzerinde genlerden oluşmaktadır ve bu genler üretilecek sonuçlar hakkındaki bilgileri içermektedir. Bu bilgilerin kullanılabilmesi için kromozomlar çözümlenerek, kullanılabilir bir forma sokulmalıdır. Çözümleme işlemi; ikili, permütasyon, değer çözümü ve ağaç çözümü olmak üzere dört farklı

şekilde yapılabilir. Kromozomun istenilen yöntemler ile çözümlenmesinden sonraki adım üretim işlemidir. Üretim iki ya da daha fazla kromozomdan yararlanılarak yeni bir jenerasyon üretilmesidir. Bu aşamada ilk akla gelen, anne ve babayı temsil eden rastgele olarak iki kromozom seçilerek, seçilen kromozomlardan herhangi birinin özelliklerini yeni kromozoma vermektir. Başka bir yöntem ise tek nokta üretimi ile anne ve baba kromozomlarının bir kısmını alarak yeni bireyler üretmektir. Yine farklı bir yöntem olan iki noktalı üretimde hem anne hem de baba kromozomlarından iki ayrı parça alınarak üretim gerçekleştirilir. Ve son olarak en basit olan, anne ve babadan rasgele alınan genlerle de yeni bireyler oluşturulabilir (Silahtaroglu, 2016).

Üretim işleminden sonra mutasyon işlemine geçilir. Mutasyondaki asıl amaç, jenerasyon içerisindeki tüm sonuçların kısır döngüye girip aynı türler üzerinde gezinmesini engellemektir. Mutasyon ile bir önceki adımda üretilmiş olan kromozomlar üzerindeki genler rastgele değiştirilir. Üretim ve mutasyon genetik algoritmaların sonuçlarını önemli ölçüde etkilemektedir. Ayrıca jenerasyon büyüklüğü de önemli bir parametredir. Jenerasyon içerisindeki kromozom sayısı ve yeni neslin kromozom sayısı algoritmanın hızı açısından önemlidir. Çok büyük miktarda nesiller üretmek algoritmanın hızı açısından bir dezavantaj olabilir. Az miktarda kromozom ile de sağlıklı sonuçlar elde edilemeyebilir (Silahtaroglu, 2016).

Üretim ve mutasyona ek olarak jenerasyon büyüklüğü de genetik algoritmalarında etkili bir parametredir. Tüm jenerasyonda kaç adet kromozom bulunmakta, her bir üretilen yeni nesilde kaç adet kromozom bulunacak vs. sayılar genetik algoritmanın çalışma hızını da arttırıp azaltacaktır.

Genetik algoritmanın avantajları şu şekilde sıralanabilir (İnan, 2018);

- Kavramların kolay tasarlanması,
- Çok amaçlı eniyileme problemleri ile kullanılabilmesi,
- Oldukça karmaşık problemlerin çözümünde bile iyi sonuçlar verebilmesi,
- İyi tanımlanamayan problemlerin çözümünde ideal çözüm yöntemi olması,

- Kısa sürelerde iyi sonuçlar verebimesi.

Dezavantajları ise;

- Elde edilen sonuçlar her zaman en iyi çözüm olmama ihtimali,
- Problemleri ve verileri genetik algoritmaların kullanılabileceği forma sokmak oldukça zor olabilir.

3.7.1.3. K- En Yakın Komşu (K- Nearest Neighbour)

Yabancı kaynaklarda k-nearest neighbour olarak geçen en yakın komşu algoritması, veri madenciliği yöntemleri içerisinde en basit sınıflama yaklaşımlarından birisidir. Bu algortmada sınıflandırma, her nesnenin (kaydın) diğer nesnelere olan uzaklığının hesaplanması ile gerçekleştirilir. Sınıflandırılacak nesne, özniteliklerine göre kendisine en yakın olan komşularının sınıfına yerleştirilir (Beyer vd. ,1999). k-en yakın komşu algoritma ismindeki 'k' ,sınıflandırmada aktif olarak kullanılan komşuların sayısını ifade etmektedir. Örneğin k=7 için yeni bir eleman sınıflandırılmak istenir ise daha önceki sınıflandırılmış elemanlardan en yakın 7 tanesine bakılarak yeni eleman bu 7 elemanın olduğu sınıfa dahil edilir. Mesafe hesaplamasında genellikle Öklid 'in mesafe hesabından yararlanır.

3.7.1.4. Naive Bayes Sınıflandırıcısı

Naive Bayes sınıflandırıcısı, temeli Bayes teoremine dayanan istatistiki sınıflandırma yöntemlerindedir. Bu teknik ile elde sınıflandırılmış olarak bulunan veriler kullanılarak yeni bir verinin bu sınıflardan hangisine dahil edileceği hesaplanır. Çoğu algortmada olduğu gibi Naive Bayes algortmasında da hesapları kolaylaştırmak için bazı kabullerde bulunulur. Bunlar; niteliklerin birbirlerinden bağımsız olması ve niteliklerin aynı önem derecesine sahip olmasıdır. Naive Bayes kavramını şu şekilde açıklayabiliriz (Han ve Kamber, 2006):

X, sınıfı bilinmeyen veri öğelerini temsil etsin. $X = \{x_1, x_2, x_3, \dots, x_n\}$ öznelik değerlerinden oluşsun. X sınıfında m tane sınıf olduğunu varsayalım. C_1, C_2, \dots, C_n sınıf değerlerini temsil etsin. Sınıfı belirlenecek olan örneğin olasılıkları (3.2) 'de verilen denklem ile hesaplanır.

$$P(C_i | X) = \frac{P(X | C_i) + P(C_i)}{P(X)} \quad (3.2)$$

Hesaplama işlemlerini azaltmak amacıyla $P(X | C_i)$ olasılığı için basitleştirilir. Birbirinden bağımsız olduğu kabul edilen, x_i değerleri için (3.3) bağıntısı kullanılabilir.

$$P(X | C_i) = \prod_{k=1}^n P(X_k | C_i) \quad (3.3)$$

Örnek X 'i sınıflandırmak için (3.2)'de, $P(C_i | X)$ içinde yer alan paydalar birbirlerine eşit olduğundan pay değerleri karşılaştırılması yeterli olur. Bu değerler içinden en büyük olan seçilir ve bilinmeyen örneğin sınıfı, bu sınıf olarak belirlenir.

$$\underset{C_i}{\operatorname{argmax}} \{P(X | C_i)\} \quad (3.4)$$

Sonuç olarak Bayes sınıflandırıcısı olarak (3.5)'teki bağıntı elde edilmiş olur:

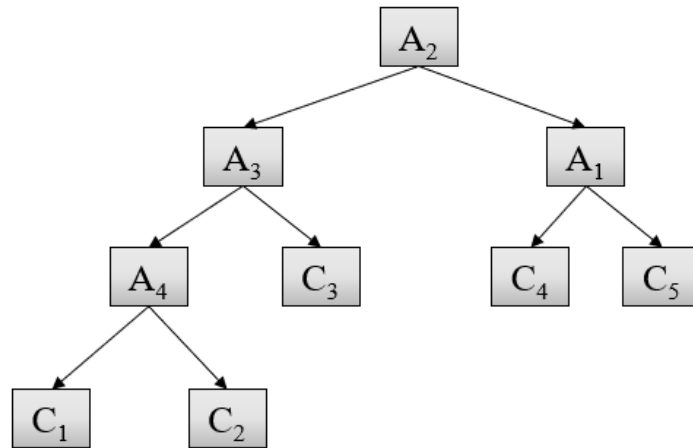
$$C_{MAP} = \underset{C_i}{\operatorname{argmax}} \prod_{k=1}^n \{P(X_k | C_i)\} \quad (3.5)$$

3.7.1.5. Karar Ağaçları (Decision Trees)

Veri madenciliği sınıflama yöntemlerinden olan karar ağaçları; görece düşük maliyeti ve veri tabanı sistemleri ile entegrasyonunun kolay olması sebebiyle en yaygın kullanılan sınıflandırma yöntemlerinden biridir. Bu yöntemle, bir kararın önceden belirlenmiş olasılıklarla başka karar noktalarını etkilediği, ardışık karar verme süreci kolaylıkla görselleştirilip hesaplamalar yapılabilir (Akpınar, 2014).

Karar ağaçları, gerçek ağaçlar gibi kök, dal ve yapraklardan oluşur. Yapı itibarıyla akış şemalarına benzerler. Nitelikler düğümlerle temsil edilir. Dallar ve yapraklar, ağaç yapısının elemanlarıdır. Ağaç yapısı, doğadaki ters çevrilmiş bir ağaca benzer. En üst düğüm kök, en alt düğüm yaprak ve bunların arasında kalan düğümler ise dal olarak adlandırılır. Kök düğümünden başlanarak her hiyerarşide veri dizisi belirli kriterlere göre bölünür. Seçilen karar ağacı algoritmasına göre bölünme ikili, üçlü veya çoklu olabilir (Özkan, 2016).

Şekil 3.5, klasik bir karar ağacı gösterimidir. Düğümler A_i 'ler ile gösterilmiştir. Her düğüm kendinden sonra iki dala ayrılmaktadır. Ayırma işleminde, cevabı veri tabanında bulunan bir soru sorularak verilen cevaba göre bu iki yoldan biri tercih edilmektedir. Yapraklar C_i 'ler ile gösterilmiştir ve her biri farklı bir sınıfı temsil etmektedir (Silahtaroglu, 2016).



Şekil 3.5. Klasik karar ağacı gösterimi (Silahtaroglu, 2016)

Karar ağaçları öğreniminde iki ana hedef vardır. Bunlar (Şeker, 2013);

Sınıflandırma problemi: Sınıflandırma ağaçları (Classification Tree): Hedef kümeyi çeşitli sınıflardan birine yerleştirmeyi amaçlayan ve sınıf tanımı yapan problemlerdir.

İkelleme problemleri: İkelleme ağaçları (Regression Trees): Sonuçta bir sınıf yerine bir değer döndüren veri madenciliği problemleridir.

En yaygın kullanılan karar ağacı algoritmaları;

- CHAID (Chi-Squared Automatic Interaction Detector),
- C&RT (Classification and Regression Trees),
- ID3 (Iterative Dichotomiser 3),
- C4.5 (Extension of ID3 Algorithm) 'tir.

Bu algoritmalar şu özellikleri ile birbirinden ayrılabilir (Akpınar, 2014).

- ✓ Kullanılan ölçü skalası,
- ✓ Her düğümden ortaya çıkan yeni düğümlerin sayısı,
- ✓ Ağacın büyümesini durdurma kriteri,
- ✓ En iyi bölen özneliğin seçilmesi,
- ✓ Budama sürecidir.

3.7.1.6. Lojistik Regresyon

Bağımlı değişkenin binom dağılımına uygun olduğu durumlarda, bağımlı değişken üzerindeki önemli risk faktörü olan, bağımsız değişkenleri belirlemek için lojistik regresyon kullanılır. Farklı bir şekilde ifade edersek; lojistik regresyon, bağımlı değişkenin ikili (binary) ya da sıralı olduğu durumlarda, bağımlı ve bağımsız değişken değerlerinin birbirlerine olan etkileri arasındaki neden sonuç ilişkisinin belirlenmesinde kullanılır. Lojistik regresyon yöntemi, olasılık kurallarına uygun bir şekilde verilerin sınıflandırılmasını sağlar (Kaya ve Yeşilova, 2011). Bu yöntemin kullanım amacı, en az değişkenle en iyi uyumu

yakalayacak şekilde, deęişkenler arasındaki ilişkiyi tanımlayabilen bir model kurmaktır (Bircan, 2004).

Lojistik regresyon, bağımlı deęişkenin deęerini tahmin etmekten ziyade bağımlı deęişkenin belirli bir deęere sahip olma olasılığını tahmin etmeye çalışır. Örneğin bir banka müşterisinin kredi notunun iyi veya kötü bir derece olup olmadığını tahmin etmeye çalışır. Olayın gerçekleşme durumu, bağımlı deęişkenin tahmini olasılığına bakılarak belirlenir. Örneğin; tahmin edilen olasılık 0.50'den büyükse tahmin EVET'e (iyi bir kredi notu) daha yakındır. Olasılığının dięer deęerleri için ise tahmin HAYIR'a (kötü bir kredi notu) daha yakındır. Bu sebeple lojistik regresyonda olasılık p ,başarı olasılığı olarak adlandırılır (Kantardzic, 2002).

3.7.2. Kümeleme

Kümeleme, elimizdeki veri topluluęu içerisinde birbirine benzeyen veri parçalarını ayırma işlemidir. Sınıflandırmada olduęu gibi veri gruplara ayrılır. Sınıflama ile farkı ise; kümelemede sınıfların önceden belirli olmamasıdır. Verilerin kaç farklı gruba ayrılacağı, verilerin birbirine olan benzerliklerine göre belirlenir (Dumhan, 2003). Kümeleme modellerinde amaç, tanımlanan verileri kullanarak nesnelere birbirlerine olan benzerlik ve farklılıklarına göre gruplara ayrılmasıdır. Belirlenen her bir gruba küme adı verilmektedir. Başlangıç aşamasında veri tabanındaki kayıtların hangi kümelere ayrılacağı veya kümelemenin hangi deęişken özelliklerine göre yapılacağı bilinmediğinden, kümeleme denetimsiz öğrenme yöntemidir.

Kümeleme yöntemlerinin çoğunda veriler arasındaki uzaklıklardan yararlanılarak kümelere ayırma işlemi gerçekleştirilir. Çoğunlukla Euclid, Manhattan, Mahalanobis, Minkowski veya Canberra uzaklık ölçütleri kullanılır (Arslan, 2008). Birimler arası uzaklıklara göre çeşitli algoritmalar kullanılarak veri tabanı alt kümelere ayrılır. Her bir kümede yer alan elemanlar dahil oldukları grubu dięer gruplardan ayıran ortak özelliklere sahiptir. Kümeleme analizinde aynı grup elemanlarının olabildiğince birbirine benzer yani homojen,

farklı grup elemanlarının birbirinden farklı yani heterojen olması istenmektedir. Bu analiz yöntemi biyoloji, tıp, antropoloji, pazarlama, ekonomi, telekomünikasyon gibi farklı birçok alanda kullanılmaktadır. Kullanım alanı ve kullanıcının amacına göre kümeleme analizi; doğru türlerin belirlenmesi, model oluşturma, gruplara dayalı tahmin, hipotez testi, veri araştırma (inceleme), hipotez oluşturma, veri indirgeme gibi amaçlar için kullanılabilir (Çakmak vd., 2005).

Kümeleme yöntemleri hiyerarşik ve hiyerarşik olmayan (bölümlemeli) kümeleme yöntemleri olmak üzere iki bölüme ayrılırken yapılan literatür incelendiğinde bu algoritmaların daha alt bölümlere ayrılacağı görülür. Hiyerarşik kümeleme yöntemleri, kümelerin bir ana küme olarak ele alınması ve sonrasında aşamalı olarak içerdiği alt kümelere ayrılması veya ayrı ayrı ele alınan kümelerin aşamalı olarak bir küme biçiminde birleştirilmesi esasına dayanır. Hiyerarşik kümeleme yöntemleri ile en sık kullanılan algoritmalar, en yakın komşu ve en uzak komşu algoritmalarıdır. Hiyerarşik olmayan yöntemlerde n tane nokta, önceden verilen k küme sayısına ($k < n$) göre kümelere ayrılır. Hiyerarşik yöntemlerin aksine kullanıcı tarafından belirlenen bazı kriterlere uygun kümeler oluşturulurken, oluşturulacak küme sayısı önceden belirlidir. Kullanıcı algoritmaya kümeler arasındaki minimum/maksimum mesafe ve kümelerin iç benzerlik kriterlerini de vermek zorundadır (Guidici, 2004). Hiyerarşik olmayan yöntemler ile en sık kullanılan algoritmalar ise k -metadoits ve k -ortalamalar (means) algoritması olarak gösterilebilir.

3.7.3. Birliktelik (İlişki) Kuralları ve Ardışık Zamanlı Örüntüler

Büyük veri kümelerindeki gizli ilişkileri keşfetmek için kullanılan birliktelik kuralları; kümeleme yöntemi gibi denetimsiz veri madenciliği yöntemlerindedir. Bu yöntem ile veri tabanlarında yığınlar halinde bulunan kayıtlar üzerine bir dizi işlemler uygulanarak, kayıtlar arasındaki ilişkiyi açıklayan kurallar çıkarılır.

Birliktelik kuralları işletmeler tarafından genellikle karar vermede kullanılan önemli bilgi kaynaklarıdır. Bu yöntemin bazı uygulama alanları; pazar sepet analizi, çapraz pazarlama, katalog tasarımı, kayıp analizleri, öneri sistemleri oluşturulması ve bunun gibidir. Market sepet analizinde, müşteri satın alma modellerini keşfetmek için çaba gösterilir. Örneğin bir markette süt ve ekmeğin birlikte satın alınıyor olması sezgisel tecrübeler sayesinde tahmin ediliyor olabilir. Ancak birliktelik kuralları analizi sayesinde hangi süt çeşidi ya da markası ile hangi ekmek çeşidi ya da markasının sıklıkla tercih edildiği bilgisine ulaşılabilir. Analiz sonucunda aşağıdaki şekilde kurallar çıkarılır;

- “Tam buğday ekmeği alan müşterilerin %45’i aynı alışverişte diyet süt te almıştır.”

Ardışık zamanlı örüntüler; ilişkilerin, birbirini izleyen dönemler üzerinden tanımlandığı analiz yöntemidir. Örneğin (Akpınar, 2014);

- “Çekiç satın alan bir müşterii ilk üç ay içerisinde %15, bu dönemi izleyen 3 ay içerisinde %10 ihtimalli çivi satın alırlar.”
- “X ameliyatı yapıldığında, 15 gün içinde %45 ihtimalle Y enfeksiyonu oluşacaktır. “

şeklindeki kurallar, ardışık zamanlı örüntü analizleri sonucunda çıkarılan kurallardır.

Kurallar sayesinde, farkına varılan birliktelik ilişkileri işletme tarafında stratejik kararların verilmesinde kullanılarak kar artışı sağlanabilir (Cios vd., 2007) .

Birliktelik kuralları madenciliği bu çalışmanın temel konusu olduğundan, bu konuya 4. bölümde ayrıntılı olarak yer verilmiştir.

4. BİRLİKTELİK (İLİŞKİ) KURALLARI

Büyük veri kümelerindeki gizli ilişkileri keşfetmek için kullanılan birliktelik kuralları; veri madenciliğinin tanımlayıcı modellerindendir. Bu yöntem ile veri tabanlarında yığınlar halinde bulunan kayıtlar üzerine bir dizi işlemler uygulanarak, kayıtlar arasındaki ilişkiyi açıklayan kurallar çıkarılır.

Birliktelik kuralları rahat anlaşılması ve kullanışlılığı ile pazarlama, finans, eğitim, sağlık, e-ticaret, perakendecilik ve telekomünikasyon gibi geniş kullanım alanlarına sahiptir (Çil, 2012).

Birliktelik kuralları tekniği ile bir dizi olayın meydana gelişinin, belirli bir olayın ortaya çıkmasını ne ölçüde etkilediği değerlendirilerek özet olarak sunulur. Analiz yapacak kişi tarafından uygulamanın başında belirlenen başarı oranına eşit ve başarı oranından yüksek olan kurallar çıkarılır (Koyuncugil ve Özgülbaş , 2009).

Bu teknikle veri tabanındaki tüm olası ilginç örüntülere erişilebilir. Tüm noktalara erişilebilmesi, veri tabanındaki kayıtların üstel olarak artması sonucunda büyüyen veri yığınlarında bile uygulanabilmesi ve hızlı hesaplama mantığı sayesinde ticari veri tabanlarının madenciliğinde kullanımı giderek yaygınlaşmaktadır. Literatürde genellikle market sepet analizi olarak da geçen birliktelik kuralları, tek bir alışverişte birlikte satın alınan ya da bir ürün alındıktan sonra belirli bir süre içerisinde alınan ilişkili ürünlerin belirlenmesinde kullanılır. Böylece müşterilerin çapraz satın alma davranışları hakkında bilgi edinilmiş olur (Yang ve Lai, 2006).

Birliktelik kuralları analizi ile ilişkili örüntülerin çıkarılması konusunda literatürde sıkça karşılaşılan bir hikâye vardır. Perakende sektöründe faaliyet gösteren Walmart ismindeki süpermarket zincirinde yapılan çalışma, cuma günleri bebek bezi alan müşterilerin büyük bir kısmının bira da aldığını ortaya çıkartmıştır. Bebek bezi ve bira, ekmek ve süt gibi zihnimize bütünleşmiş iki ürün değildirler. Ancak bu birliktelik ilişkisine getirilen yorum şu şekildedir; “Küçük çocukları olan ebeveynler cuma günleri eğlenmek istediklerinde partiye gitmek yerine evlerine bira alarak evlerinde eğlenmek zorunda kalıyorlar.

Böylece bebek bezi alan baba çoğunlukla yanında bira da alır,” şeklinde bir birlikteliğin doğmasına neden oluyor (Rao ve Shield, 1998). Böylesine ilginç birlikteliklerin farkında olunması ve bu doğrultuda karı arttırmaya yönelik çalışmalar yapılması, günümüz piyasasında işletmelere rekabet üstünlüğü sağlayacaktır.

Market sepet analizinde amaç ilişkili kartegorileri veya ürünleri belirlemektir. Belirlenen ilişkilerin doğru şekilde kullanılması ile şirketin karında artış sağlanabilir. Örneğin X ürününü alan müşterilerinin Y ürününü de büyük olasılıkla aldıklarını biliyorsak ve eğer müşteri X malını alıyor ama Y malını almıyorsa o müşteri potansiyel Y müşterisidir. Müşterileri bu şekilde kategorize etmek bile firma karlılığı için büyük önem taşır.

4.1. Birliktelik Kurallarının Matematiksel Modeli

Birliktelik kuralının matematiksel modeli Agrawal, Imielinski ve Swami tarafından 1993 yılında ortaya çıkarılmıştır.

Modelde; nesnelar kümesi, $I = \{i_1, i_2, i_3, \dots, i_m\}$ ve işlemler kümesi D şeklinde ifade edilir. Her i, farklı bir nesneye(ürüne) karşılık gelir. D veri tabanındaki her hareket (transaction) T, $T \subset I$ şeklinde tanımlanan nesneların kümesi (nesneküme) olsun. TID her harekete ait tekil bir numaradır ve m öğelerin sayısıdır. A ve B nesnekümleri temsil etsin. Bir T işlemler kümesi ancak ve ancak $A \subset T$ ise yani A, T'nin alt kümesi ise A'yı kapsıyor denir. A ve B'nin $A \subset I, B \subset I$ ve $A \cap B = \emptyset$ koşullarını sağlayacak şekilde nesnekümler olduğu varsayıldığında birliktelik kuralı $A \Rightarrow B$ şeklinde ifade edilir. Burada A öncül, B sonuç olarak adlandırılır (Cios vd., 2007).

Birliktelik kuralları madenciliğinde sıkça kullanılan kavramdan bahsedelim (Öğüdücü, 2018):

Öğeler kümesi \ Nesneküme (itemset): Bir veya daha fazla öğeden oluşan kümedir. 'k' yerine küme öğe seti içerisindeki eleman sayısı yazılarak 'k-nesneküme' şeklinde ifade edilir.

Destek sayısı (support count): Öğeler kümesinin veri kümesinde görülme sıklığıdır.

Yaygın öğeler \ Sık nesneküme (frequent itemsets): Destek değeri, kullanıcı tarafından belirlenen destek değerinden büyük veya eşit olan nesnekümelere k -nesnekümenin sık geçen nesnekümelere L_k şeklinde ifade edilir.

Destek(support) ve Güven(confidence) Ölçütleri:

Sepet analizi gerçekleştirilecek veri seti yalnızca ürünün satılıp satılmadığına dair bilgi içeriyorsa , ürünler arasındaki bağıntı destek(support) ve güven(confidence) değerleri aracılığı ile hesaplanır (Alpaydın, 2000).

A ve B ürün gruplarını birlikte içeren alışveriş sayısını ifade etmek için *destek sayısı* kullanılır ve *sayı* ($A \Rightarrow B$) şeklinde gösterilir.

Kural destek ölçütü bir ilişkinin yapılan tüm alışverişler içerisindeki oranını ifade etmek için kullanılır;

$$destek(A \Rightarrow B) = \frac{\text{sayı}(A,B)}{\text{Tüm alışverişler}} \quad (4.1)$$

Kural güven ölçütü, A ürün grubunu alan müşterilerin B ürün grubunu da alma olasılığını ortaya koymaktadır. (4.2) 'de gösterilen şekilde hesaplanır.

$$güven(A \Rightarrow B) = \frac{\text{sayı}(A,B)}{\text{sayı}(A)} \quad (4.2)$$

Destek ve güven ölçütlerini karşılaştırmak için eşik değeri kullanılır. Çıkarılan kuralı anlamlı ve ilginç sayılabilmesi için destek ve güven ölçütlerinin belirlenen eşik değerinin üzerinde olması gerekir. Örneğin süt ve ekmek alan müşterileri tanımlayan ilişki kuralı şu şekilde ifade edilir:

süt \Rightarrow ekmek [*destek* = %25, *güven* = %60]

Bu gösterimde soldan sağa doğru olan yönlü ilişki, süt alımının ekmek alımını tetiklediğini göstermektedir. Örnekteki %25 oranındaki destek değeri ile tüm alışverişlerin %25'inde süt ile ekmeğin birlikte satın alındığı anlaşılır. %60'lık güven değerinden ise aynı alışverişte süt alan müşterilerin %60'ının ekmek te satın aldığı anlaşılır. Hesaplanan destek ve güven değerlerinin büyüklüğü, çıkarılacak birliktelik kuralının güçlü olduğunu gösterir. Başta belirlenen destek ve güven değerini aşan birliktelikler, kural olarak kabul edilir. Destek değeri çıkarılan kuralın kullanılabilirliğini gösterirken güven değeri kuralın doğruluğunu gösterir (Cios vd., 2007).

Yaygın nesneküme, destek ve güven terimlerinin daha iyi anlaşılabilmesi aşağıdaki örnek incelenebilir.

ÖRNEK: Çizelge 4.1 'de verilen örnek veri tabanımızda beş müşterinin alışveriş kayırları bulunmaktadır.

Çizelge 4.1. Örnek veri tabanı

TID	Nesneler
001	Süt, Çikolata
002	Çikolata, Yumurta
003	Süt, Ekmek
004	Yumurta, Çikolata, Ekmek, Süt
005	Ekmek, Yumurta, Süt

Yaygın nesnekümelerin bulunmasında kullanılan destek sayısı, ürünün işlemler arasında geçişinin sayılması ile bulunur. Süt için veri tabanımızdaki tüm işlemlere bakarsak, sütün dört kez geçtiğini görürüz ve Destek Sayı({Süt}) =4 şeklinde gösterebiliriz. İki, üç ve dört elemanlı nesleler kümesi ve destek değerleri de aynı şekilde hesaplanır. Çizelge 4.2'de, örnek veri tabanından

oluşturulabilecek olası sık geçen nesnekümelerin destek değer sayıları gösterilmiştir.

Çizelge 4.2. Örnek veri tabanı üzerinden 1,2,3 ve 4 elemanlı nesnekümelerin destek değerlerinin hesaplanması

1-Elemental Öğeler Kümesi	Destek Sayıları
{Süt}	4
{Çikolata}	3
{Yumurta}	3
{Ekmek}	3

2-Elemental Öğeler Kümesi	Destek Sayıları
{Süt, Çikolata}	2
{Süt, Yumurta}	2
{Süt, Ekmek}	3
{Çikolata, Yumurta}	2
{Çikolata, Ekmek}	1
{Yumurta, Ekmek}	2

3-Elemental Öğeler Kümesi	Destek Sayıları
{Yumurta, Çikolata, Ekmek}	1
{Yumurta, Çikolata, Süt}	1
{Çikolata, Ekmek, Süt}	1
{Ekmek, Yumurta, Süt}	2

4-Elemental Öğeler Kümesi	Destek Sayıları
{Yumurta, Ekmek, Peynir, Süt}	1

Bu örnekte destek eşik değerleri dikkat alınmadan tüm olası yaygın nesnekümelerin destek değerleri hesaplanmıştır. Normalde yaygın

nesnekümeler bulunurken, işlem başlangıcında kullanıcı tarafından belirlenen minimum eşik destek değerine göre, hesaplama sırasında minimum eşik destek değerinin altında kalan birliktelikler elenerek hesaplamalara devam edilir. Belirlenen eşik güven ve destek değerini sağlayan öğeler sık geçen nesnekümelerine dahil edilir. Bu kümelerden birliktelik kuralları çıkarımı gerçekleştirilir. Destek ve güven değerleri, aslında olasılık ifadeleri olduğunda 0 ve 1 arasında değerler alırlar. Bu iki değer 1'e ne kadar yakın olursa o kadar kuvvetli birliktelikler oluşturulmuş olur. Başlangıçta belirlenen eşik değerleri ne kadar büyük olursa, bu eşik üzerinde kalan öğelerle oluşturulan kurallar o kadar güvenilir ve gerçekçi olur. Güven değerinin 1 olması, o kuralın artık kesin olduğu anlamına gelir. Çizelge 4.1'de verilen veri tabanımızdan destek ve güven değerlerine örnek olarak şunları verebiliriz;

$$\text{Destek}(\{Yumurta\} \Rightarrow \{Ekmek\}) = \frac{\text{Destek Sayısı}(Yumurta, Ekmek)}{\text{Toplam alışveriş sayısı}} = \frac{2}{5} = \%40 \quad (4.3)$$

$$\text{Güven} (\{Ekmek, Süt\} \Rightarrow \{Yumurta\}) = \frac{\text{Destek Sayısı}(Ekmek, Süt, Yumurta)}{\text{Destek Sayısı}(Ekmek, Süt)} = \frac{2}{3} \approx \%66 \quad (4.4)$$

(4.3) eşitliğinde çıkan, %40'lık destek değeri; tüm alışverişlerin %40'ında yumurta ile ekmeğin birlikte satın alındığını ifade eder. (4.4) eşitliğindeki %66'lık güven değeri ise; ekmek ve süt alan müşteriler, %66 olasılıkla yumurta da almıştır demektir.

c

4.2. Birliktelik Kuralı Madenciliği

Veri tabanlarından birliktelik kuralları çıkarımı genelde iki aşamada gerçekleştirilir (Han ve Kamber, 2006):

1. *Sık geçen nesnekümelerin bulunması*: Bu aşamada, önceden belirlenmiş olan minimum destek değerini sağlayan nesnekümeler bulunur. Birliktelik kurallarının performansını belirleyen adım bu adımdır. Bir k-nesnekümenin,

boş olmayan $2^k - 1$ tane alt kümesi vardır. Boş olmayan bu alt kümelerin hepsi potansiyel sık geçen nesne küme olabilir. K değerinin büyük olduğu veri tabanlarında bu işlemin bilgisayarlarla dahi hesaplanıp hafızada tutulması zorlaşır. Bunun için bazı algoritmalar geliştirilmiştir. Bunların en yaygını olan Apriori algoritması, ilerleyen bölümlerde anlatılacaktır.

2.Sık geçen nesnekümelerden güçlü birliktelik kurallarının oluşturulması: İlk adımda bulunan sık geçen nesnekümeler içerisinde minimum güven eşik değerini sağlayan birliktelik kurallarının oluşturulduğu adımdır. Sık geçen nesnekümeler belirlendikten sonra, birliktelik kurallarının oluşturulması zor bir adım değildir.

4.3. Birliktelik Kuralları Madenciliği Algoritmaları

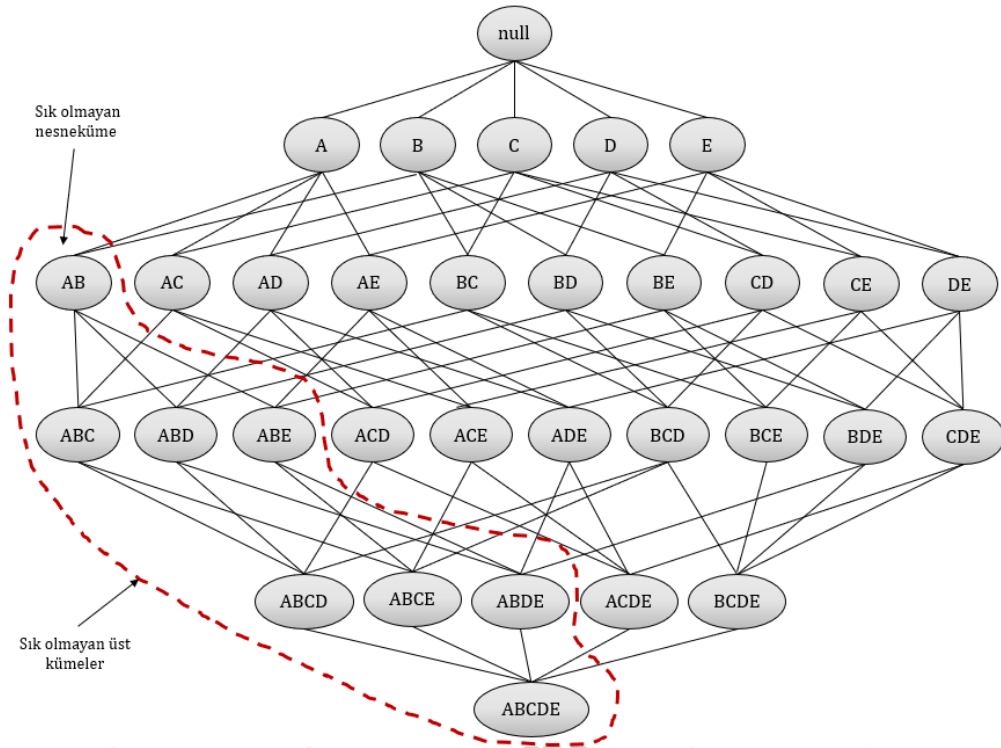
4.3.1. Apriori Algoritması

Birliktelik kuralları madenciliği çalışmaları ilk kez 1993 yılında, Agrawal tarafından başlatılmıştır. Agrawal ve Srikant 1994 yılında, günümüzde en çok bilinen ve kullanılan algoritma olan, Apriori algoritmasını geliştirmişlerdir. Algoritma sık geçen nesnekümelerin bir önceki adımdan tahmin edilmesi üzerine kurulmuştur. Adı da önceki anlamındaki 'prior' kelimesinden gelmektedir (Agrawal vd., 1993).

Apriori algoritması; sık olmayan bir nesnekümenin tüm alt kümelerinin de sık olamayacağını ya da başka bir ifade ile eğer bir nesneküme, minimum eşik destek değerini sağlamıyorsa bu nesnekümenin üst grupları da minimum eşik destek değerini sağlamaz diyor. Bunu şu şekilde kanıtlayabiliriz:

n , veri tabanımızda kayıtlı tüm işlemlerimizin sayısı olsun. A 'nın i işlemlerinin bir altkümesi olduğunu varsayalım. $A' \subset A$ ve $i' \leq i$ olacak şekilde i' işlemler kümesinin bir altkümesi olan A' olduğunu varsayalım. Eğer $i/n < \text{minimum destek değeri}$ ise dolayısıyla $i'/n < \text{min destek değeri}$ olacaktır (Cios vd., 2007).

Şekil 4.1’de, beş elemanlı bir kümenin tüm olası altkümelerini göstermektedir. Şekildeki yaygın öge ağacı üzerinde kırmızı ile çevrili kısımda sık olmayan nesnekümler ayrılmıştır. Burada {A, B}, sık geçen nesneküme değil ise onu içeren tüm üst kümelerin de sık geçen olmayacağı gösterilmiştir. Bunun tam tersi de olabilirdi. Mesela {A, D, E} ‘nin sık geçen nesneküme olduğu varsayılırsa, {A, D, E} ‘nin altkümeleri olan; {A, D}, {A, E}, {D, E}, {A}, {D}, {E}, { } kümeleri de sık geçen nesneküme olacaktı.



Şekil 4.1. Sık olmayan nesnekümenin üst kümelerinin de sık olmayacağıının, 5 elemanlı bir kümenin olası altkümeleri üzerinden gösterimi

Algoritma ilk olarak veri tabanını tarar ve destek değeri olarak adlandırdığımız, öğenin alışverişlerde kaç kez geçtiğine dair olan sayıyı tespit eder. Kullanıcı tarafından başlangıçta belirlenmiş olan minimum eşik destek sayısının altında kalan öğeler kümelere dahil edilmeden ayıklanır.

Apriori özelliği sık öge setlerini bulabilmek için aranması gereken öge sayısını azaltmak için kullanılır. Bu algoritma 1-nesnekümlerden yararlanarak 2-nesnekümleri, 2-nesnekümeden yararlanarak 3-nesneküme bulma işlemlerine iteratif olarak devam eder. Bu da şu şekilde genellenebilir; k

elemanlı yaygın öge adaylarını bulmak için de (k-1) elemanlı sık geçen nesnekümelerden yararlanılır. Apriori algoritmasının işleyişi şu şekildedir (Cios vd., 2007);

-Veri tabanı taranarak ilk olarak 1-nesneküme bulunur.

-Daha sonra, 1-nesnekümeler içinden sık geçen nesnekümeler bulunur ve L_1 ile gösterilir.

-Sonrasında, L_1 'den yararlanılarak 2-nesnekümeler oluşturulur.

-Oluşturulan 2-nesnekümeler arasında sık geçenler belirlenerek L_2 oluşturulur.

-Aynı şekilde L_3 , L_4 , vd. bulunabilmesi için işlemler tekrarlanır.

Apriori özelliğine göre, her yinelemede minimum destek değerini sağlamayan k-nesneküme'ler elenir ve sadece eşik destek değerini sağlayan nesnekümeler (k+1)-nesneküme oluşturmak için kullanılabilir. Bu işlem algoritma sık geçen nesneküme bulamayana kadar devam eder. Bu işlem sayesinde sık geçen nesnekümeler belirlenirken taranması gereken olası sık geçen nesneküme sayısı büyük ölçüde azaltılmış olur. Apriori algoritmasının klasik kod karşılığı Çizelge 4.3 'te, Apriori-gen fonksiyonunun kod karşılığı Çizelge 4.4 ile verilmiştir (Agrawal ve Srikant, 1994).

Çizelge 4.3. Klasik Apriori algoritması özel kodu

```
1)  $L_1 = \{ \text{Sık geçen 1-nesneküme} \};$ 
2) for (k=2;  $L_{k-1} \neq \emptyset$ ; k++) do begin
3)    $C_k = \text{Apriori-gen}(L_{k-1});$ 
4)   forall transactions - t  $\in D$  do begin
5)      $C_t = \text{subset}(C_k, t);$ 
6)     forall candidates c  $\in C_t$  do
7)       c.count++;
8)     end
9)    $L_k = \{ c \in C_k \mid c.\text{count} \geq \text{minsup} \}$ 
10) end
11) Answer =  $\cup_k L_k$  ;
```

Çizelge 4.4. Apriori-gen fonksiyonu kodu

```
insert into Ck  
select p.items1, p.items2, p.items3,....., p.itemsk-1, q.itemsk-1  
from Lk-1 p, Lk-1 q  
where p.item1=q.item1,....., p.itemk-2=q.itemk-2, p.itemk-1 < q.itemk-1 ;  
forall itemsets c ∈ Ck do  
    forall (k-1) - subsets s of c do  
        if (s ∉ Lk-1) then  
            delete c from Ck
```

Çizelge 4.4'de kodları verilen Apriori-gen fonksiyonu ile aday sık geçen nesnekümeler olan C_k'lar oluşturulur. Algoritmadan, k-nesneküme 'lerden C_k oluşturabilmek için L_{k-1} sık geçen öge kümesinin kullanıldığı görülmektedir. Apriori-gen fonksiyonu ile öncelikle L_{k-1} ögesi kendisi ile birleştirme işlemine tabi tutulur. Birleştirme işleminde L_{k-1} aday sık geçen nesnekümesinin her satırında yer alan son öge haricinde diğer öğelerin çapraz olarak benzerliği aranır ve son öge haricinde diğer öğelerle yakalanan benzerliklerden aday sık geçen nesnekümeleri oluşturulur (Özçakır ve Çamurcu ,2007). Oluşturulan aday sık geçen nesnekümeleri budama (prune) işlemine tabi tutulur ve aday sık geçen nesnekümesini geriye döndürmektedir.

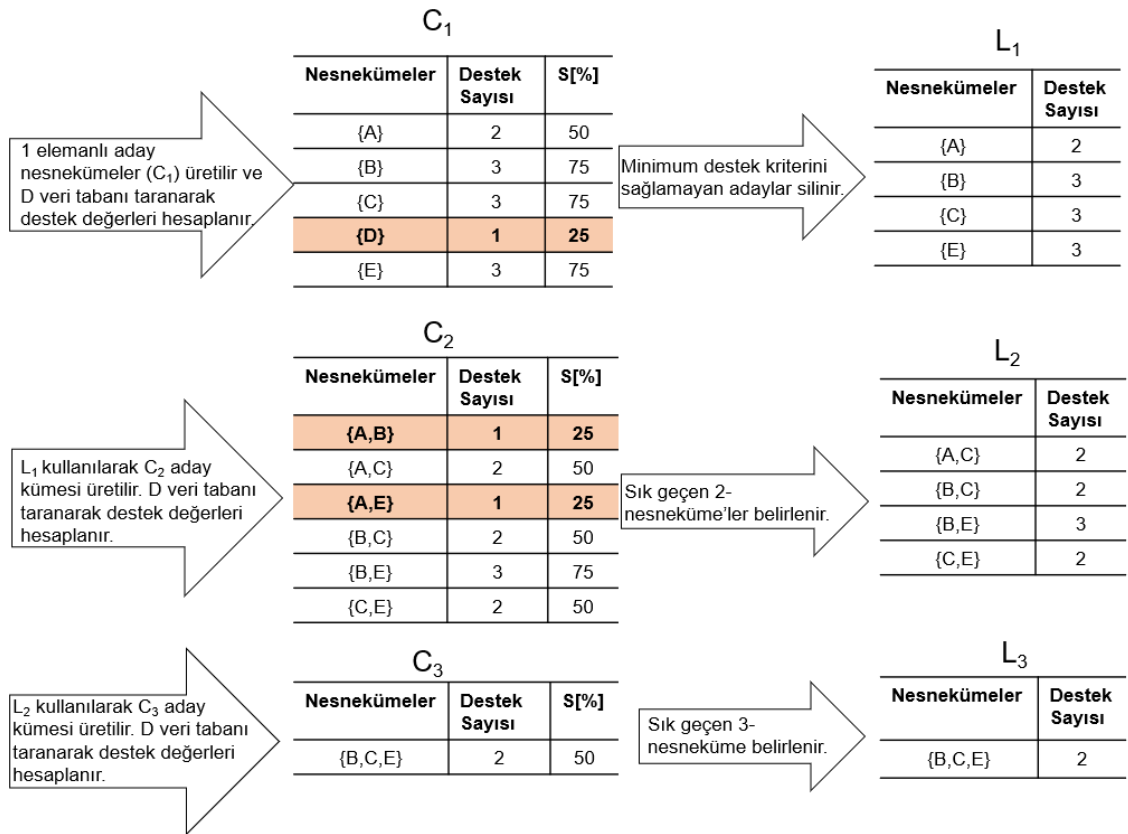
Budama işleminde C_k yaygın öge aday kümesinden eleman silme işlemi uygulanır, aday öge kümesindeki bir ögenin alt kümelerinden biri L_{k-1} yaygın öge kümesinde yer almıyorsa bu öge C_k aday öge kümesinden silinir (Agrawal ve Srikant ,1994).

Apriori algoritmasının işleyişinin daha iyi anlaşılabilmesi için Çizelge 4.5 ile verilen örnek üzerinde algoritmanın uygulamasını gösterelim.

Çizelge 4.5. Apriori algoritması için örnek veri tabanı

TID	İşlemler
1000	A, C, D
2000	B, C, E
3000	A, B, C, E
4000	B, E

Çizelge 4.5'te örnek veri tabanına ait işlemler verilmiştir. Minimum destek=%50, minimum güven değeri=%60 olarak belirlenmiş ve bu şartlar altında Apriori algoritmasının işleyişi Şekil 4.2 'de gösterilmiştir. (Cios vd. ,2007 ; Kantardzic ,2002).



Şekil 4.2. Örnek veri tabanı üzerinde Apriori algoritması adımları

Son olarak, oluşturduğumuz 3 elemanlı {B, C, E} sık geçen nesne kümesinden %50 minimum destek ve %60 minimum güven kriterini sağlayan kurallar üretilir.

{B, C} \Rightarrow {E} [destek = %50, güven=2 / 2= 100%]

{B, E} \Rightarrow {C} [destek = %50, güven=2 / 3= 66,7%]

{C, E} \Rightarrow {B} [destek = %50, güven=2 / 2= 100 %]

{B} \Rightarrow {C, E} [destek = %50, güven=2 / 3= 66,7%]

{C} \Rightarrow {B, E} [destek = %50, güven=2 / 3= 66,7%]

{E} \Rightarrow {B, C} [destek = %50, güven=2 / 3= 66,7%]

Çizelge 4.5 ile verilen örnek veri tabanını, birlikte satın alınan ürünlere ait veri tabanı kayıtları olarak düşünebiliriz. Minimum destek değerimiz %50, veri tabanımız 4 ayrı işlemde olduğu için aynı zamanda 2 değerine karşılık gelmektedir. İlk olarak veri tabanı taranarak 1 elemanlı aday nesnekümeler olan C_1 'ler oluşturulmuştur. C_1 öge kümesinde eşik destek değerini sağlamayan öğeler çıkartılır. Şekle bakıldığında {D} işlemi destek sayısının 1 olması nedeni ile kümeden çıkarılır ve C_1 aday sık geçen nesnekümesinin diğer elemanları ile L_1 oluşturulur. Bir sonraki adımda ise 1 elemanlı sık geçen nesnekümelerden, 2 elemanlı aday sık geçen nesneküme oluşturulmaya çalışılır. Bunun için L_1 sık geçen nesnekümesinin öğeleri kendi içlerinde ikili kombinasyonları oluşturulur ($L_1U_{L_1}$), bu kombinasyonlar C_2 aday nesnekülerin üretilmesinde kullanılır. C_2 kümesi elemanlarının alt kümelerinin L_1 'de olup olmadığı kontrol edilir ve L_1 'de bulunmayan öğeler silinir. Devam eden adımda C_2 'den L_2 oluşturma işlemi için daha önceki adımlarda olduğu gibi C_2 'nin minimum destek kriterini sağlamayan öğeleri silinir. Örneğimizde bu öğeler {A, B} ve {A, E} 'ye karşılık gelmektedir ve C_2 'den bu öğeler çıkarılarak L_2 oluşturulur. Bir sonraki aşamada L_2 kümesinin öğeleri arasındaki 3'lü kombinasyonlara bakılarak C_3 aday sık geçen nesnekümeler oluşturulur. {D} sık geçen nesneküme değildir. Bunun için {B, C, D} aday sık geçen nesnekümelere dahil edilmez. Yine {A, B} sık geçen nesneküme olmadığından, {A, B}'nin üst kümesi olan {A, B, C, E} 'de C_3 e dahil edilmez. C_3 'ün elemanı olabilecek, minimum eşik destek kriterini sağlayan

yalnızca {B, C, E} nesnekümesi vardır. Ve böylece $L_3 = \{B, C, E\}$ olarak belirlenmiş olur. Bu örnek için 4 elemanlı aday sık geçen nesne kümeler bulunamayacağı için işlem burada sona erer fakat büyük veri tabanları için işlem, öge sayısının birerli olarak artması ile devam edecektir. Bu döngü sık geçen nesneküme bulunamayınca kadar devam etmektedir.

4.3.2. Apriori Tid Algoritması

Apriori Tid algoritması, apriori algoritması ile aynı işleve sahip olan bir algoritmadır. Her ikisi de sık geçen nesnekümelerin önceki adımlardan tespit edilmesi üzerine kurulmuştur. Apriori Tid algoritması geçiş işlemlerinden önce apriori gen fonksiyonunu kullanarak sık geçen nesneküme adaylarını belirler. Bu algoritmanın ilginç olan yanı ise ilk geçişten sonraki destek değeri hesaplarında veri tabanını kullanmamasıdır. Sonraki her geçişte bir önceki adımdaki aday sık geçen nesnekümeler kullanılarak destek değerleri hesaplanır. Bu şekilde her geçiş işleminde taranması gereken veri tabanı küçülecek ve böylece zamandan önemli ölçüde tasarruf sağlanmış olacaktır (Agrawal ve Srikant,1994).

Çizelge 4.5 ile verilen örnek veri tabanını kullanarak Apriori Tid algoritmasının işleyişi gösterilmiştir. Veri tabanı taranarak \hat{C}_1 oluşturulmuştur. Oluşturulan \hat{C}_1 kullanılarak destek değerlerini içeren L_1 hesaplanmıştır. Sonrasında apriori-gen fonksiyonu kullanılarak L_1 'den aday sık geçen nesnekümeler olan C_2 oluşturulmuştur. C_2 nesnekümelerinin destek değeri hesabında ilk veri tabanı yerine \hat{C}_1 taranarak \hat{C}_2 oluşturulmuştur. \hat{C}_2 'deki destek değerleri sayılarak L_2 oluşturulur. Aynı işlemlere tekrar edilerek L_2 'den C_3 oluşturulur. C_3 'ün destek değeri hesabı için \hat{C}_2 kullanılarak \hat{C}_3 oluşturulur. C_3 'te 1000 ve 4000 işlemleri yer almadığından \hat{C}_3 'de 1000 ve 4000 işlemleri bulunmamaktadır. C_3 'teki tek aday olan {B, C, E}, L_3 'ü oluşturur ve C_4 boş küme olduğundan algoritma sonlanmış olur.

Şekil 4.3'te, Çizelge 4.5 ile verilen örnek veri tabanı üzerinde Apriori-Tid algoritmasının adım adım işleyişi gösterilmektedir.

TID	İşlemler
1000	A,C,D
2000	B,C,E
3000	A,B,C,E
4000	B,E

TID	Nesnekümeler
1000	{ {A}, {C}, {D} }
2000	{ {B}, {C}, {E} }
3000	{ {A}, {B}, {C}, {E} }
4000	{ {B}, {E} }

Nesnekümeler	Destek
{A}	2
{B}	3
{C}	3
{E}	3

Nesnekümeler	Destek
{A,B}	1
{A,C}	2
{A,E}	1
{B,C}	2
{B,E}	3
{C,E}	2

TID	Nesnekümeler
1000	{ {A,C} }
2000	{ {B,C}, {B,E}, {C,E} }
3000	{ {A,B}, {A,C}, {A,E}, {B,C}, {B,E}, {C,E} }
4000	{ {B,E} }

Nesnekümeler	Destek
{A,C}	2
{B,C}	2
{B,E}	3
{C,E}	2

Nesnekümeler	Destek
{B,C,E}	2

TID	Nesnekümeler
2000	{ {B,C,E} }
3000	{ {B,C,E} }

Nesnekümeler	Destek
{B,C,E}	2

Şekil 4.3. Apriori-Tid algoritması adımları

4.3.3. Apriori-Hybrid Algoritması

Apriori-Hybrid algoritması 1994 yılında Agrawal ve diğerleri tarafından, veri üzerindeki tüm geçişlerde aynı algoritmanın kullanılmasının zorunlu olmadığı fikri üzerinden ortaya çıkarılmıştır. Yapılan çalışmada kullanılan veri tabanında Apriori ve AprioriTid algoritmalarının veri üzerinden farklı geçişlerdeki çalışma sürelerinin farklı olduğu elde edilmiştir. Apriori'nin ilk geçişlerde Apriori-Tid' e göre daha iyi performans sergilediği, sonraki geçişlerde ise Apriori-Tid'in Apriori 'den daha iyi olduğu görülmüştür. Bunun üzerine Apriori-Hybrid algoritması tasarlanmıştır. Bu algortmada ilk geçişlerde Apriori algoritması kullanılırken C_k kümesinin belleğe sığabileceğinin anlaşılması üzerine Apriori-Tid' e geçilir.

Apriori-Hybrid algoritmasının performansı büyük veri setleri üzerinde de denenmiş ve neredeyse her durumda Apriori'den daha iyi performans sergilediği görülmüştür. Ancak Apriori 'den Apriori-Tid'e geçiş maliyetlidir. Bu

maliyete katlanmak, ilk geçişlerde C_k 'nın boyutundaki ani düşüşler olduğu takdirde yani Apriori-Tid 'in uzun süre kullanılacağı durumlarda mantıklıdır. C_k uzun süre büyük olup son geçişlerde düşüş gösterirse Apriori-Tid kısa bir süre kullanılacağından Apriori-Hybrid kullanılması amacına ulaşmaz. Apriori-Hybrid doğru durumlarda kullanılması ile çalışma süresinde büyük ölçüde kısalma sağlanabilir (Agrawal vd. ,1993 ; Srikant ve Agrawal ,1996).

4.3.4. AIS Algoritması

AIS algoritması, Agrawal, Imielinski ve Swami tarafından 1993 yılında geliştirilmiştir. Bu algoritma tüm yaygın nesne kümelerinin bulunması amacıyla geliştirilen ilk algoritmadır (Agrawal vd. ,1993). Algoritmanın hedefi nitelikli olan kuralları bulmaktır. AIS algoritması veri tabanı üzerinde çoklu geçişler yapar ve her geçişte veri tabanını tarar. İlk geçişinde tek nesnelerin desteğini sayarak, bu nesnelerin veri tabanında görülme sıklığını belirler. Her geçişteki yaygın nesnekümeler aday nesnekümeler üretmek için genişletilir. Bir tarama yapıldıktan sonra önceki taramadaki yaygın nesnekümeler ile taranan nesnelerin arasındaki ortak nesnekümeler belirlenir. Bu ortak nesnekümeler yeni aday nesnekümeler üretmek için veri tabanındaki diğer nesneler ile genişletilir. I sık geçen nesne kümesi sadece sık geçen ve alfabetik olarak kendisinden sonra gelen nesne kümelerle genişletilebilir. AIS algoritması bu işlemleri gerçekleştirebilmek için tahmin edici araçları ve budama tekniğini kullanır. Bu teknikler aday gruplarını, gereksiz nesneleri aday listesinden atarak belirler. Belirlenen gruplar arasında minimum destekten büyük veya minimum desteğe eşit olan aday grupları yaygın nesnekümeler olarak seçilir. Yaygın nesnekümeler sıradaki geçişte aday üretmek için genişletilir. Bu işlem sık geçen nesnekümeler bulunamayınca kadar devam eder (Dunham vd. ,2000).

4.3.5. SETM

SETM algoritması, Houtsmal tarafından 1995 yılında, sık geçen nesnekümelerin hesaplanması için önerilmiştir. AIS algoritmasından farklı olarak SETM

algoritması, sık geçen nesneküme adayları oluşturmak için SQL kodlarını kullanmaktadır.

SETM algoritması, AIS algoritması gibi veri tabanını birçok kez tarar. Algoritma, ilk tarama sırasında veri tabanını teker teker sayar ve yaygın nesnelere belirler. Sonraki taramalarda, bir önceki geçişte sık geçen olarak belirlediği nesnekümelere kullanır ve sık geçen nesneküme adaylarını belirler. AIS 'ten farklı olarak, SETM algoritması üzerinde çalışılan işlemin TID bilgisini de tutar. Sonraki aşama aday nesnekümelere nesne ismine göre sıraya dizilerek küçük nesnekümelere silinir. Eğer veri tabanı TID numarasına göre sıralanmışsa bir sonraki tarama esnasında herhangi bir işlemdeki sık geçen nesnekümelere L_k 'nın TID numarasına göre sıralanmasıyla elde edilir. Veri tabanı, bu şekli ile birkaç kez taranır. Başka herhangi bir sık geçen nesneküme bulunmadığında algoritma sonlanır (Silahtaroglu ,2016).

Algoritmanın dezavantajları;

- TID bilgisinin tutulması, yer karmaşıklığını artmasına neden olacaktır.
- Sık geçen nesneküme adaylarının destek sayıları hesaplanırken C_k sıralanmış şekilde olmadığından, nesnekümelere yeniden sıraya dizilmesi gerekecektir. Bu da zaman karmaşıklığını arttıracaktır (Silahtaroglu ,2016).

4.3.6. FP-Growth (Frequent Pattern Growth) Algoritması

FP-Growth algoritması, büyük veri tabanlarında birliktelik kuralı madencilği gerçekleştirmek için etkili bir yöntemdir. FP-Growth algoritması, Apriori algoritmasındaki büyük zaman kaybına yol açan üretme ve test etme adımlarını içermeden sık geçen nesnekümelere belirlenmesi için üretilmiş bir algoritmadır. Bu algoritmada veri tabanı yalnızca iki kez taranır. İlk taramada öğelerin destek değerleri hesaplanır ve sık geçen nesnekümelere FP ağacına sıkıştırılır (Zhang vd. , 2008). Sonrasında sıkıştırılan veri tabanı şartlı veri yapısına bölünür ve her biri sık geçen nesne ile ilişkilendirilmiş veri tabanı ayrı ayrı madenlenir (Döşlü ,2008).

Bu method ile küçük sık geçen nesnekümelere yararlanılarak, büyük sık geçen nesnekümelere belirlenir. Algoritma en az sık geçen olan kümeler son ek

(suffix) olarak kullanarak iyi seçicilik sağlar. Bu method ile arama maliyeti önemli ölçüde azaltılmış olur (Döşlü ,2008) .

FP-Growth algoritmasının temel yürütme süreci şu şekildedir (Zhang vd. , 2008);

1.İlk olarak veri tabanı taranır ve tüm öğelerin destek değerleri hesaplanır. Eşik değerine eşit ve eşik değerinden büyük olan öğeler bulunur.

2. Sık geçen nesnekümlerin destek değerleri büyükten küçüğe olacak şekilde sıralanır.

3.Sadece kökleri olan bir ağaç oluşturulur.

4.Veritabanı tekrar taranır ve her örnek için;

a) Sadece sık öğeleri (2.adımda sıralanan öğeleri) kullanarak numuneden alınan öğeler ağaca eklenir.

b) Tüm numuneler işlenene kadar a adımı tekrarlanır.

FP-Growth algoritmasının işleyişi, aşağıdaki on işlem içeren küçük bir örnek veri tabanı üzerinden anlatılacaktır (Tan vd. , 2005) .

Çizelge 4.6. FP-Growth algoritması için örnek veri tabanı

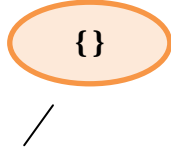
TID	Items
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

Çizelge 3.6'da verilen örnek veri tabanı için FP-Growth algoritmasını uygulayarak FP ağaçlarını oluşturalım. Bu örnek için eşik destek değerini %40 olarak alalım.

1.Adım: Tüm öğelerin destek değerleri hesaplanır {a:8,b:7,c:6,d:5,e:3}.

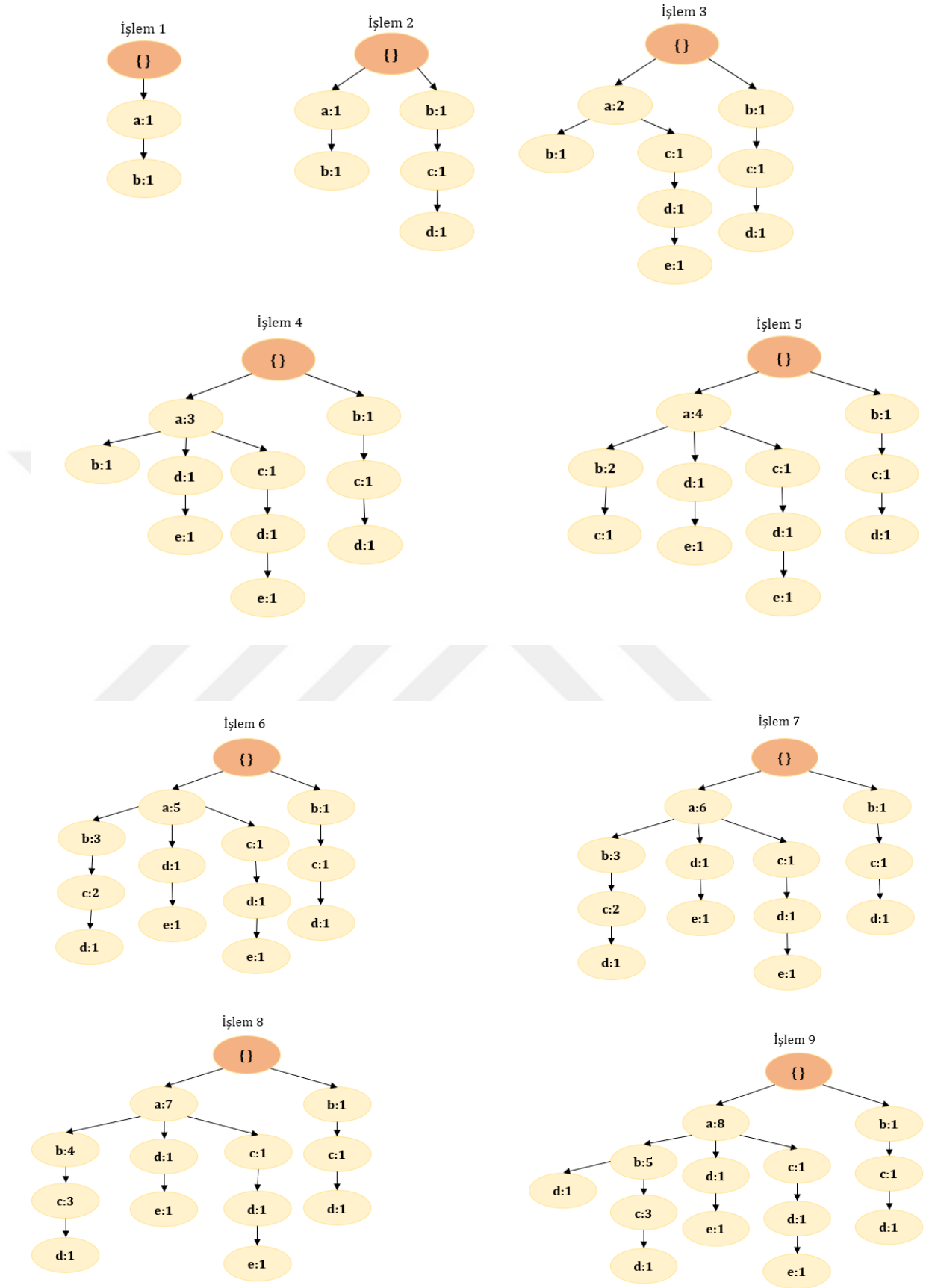
2.Adım: Sık geçen nesnekümler büyükten küçüğe {a:8,b:7,c:6,d:5,e:3} şeklinde sıralanır.

3.Adım:



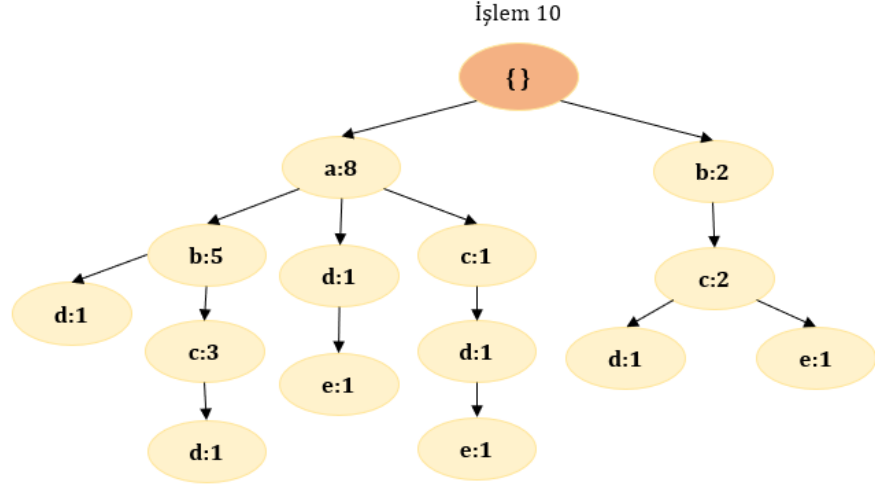
4.Adım: Örnek veri tabanında işlemlere ait ürünler, destek değerleri sıralamasına göre tesadüfî olarak sıralı çıkmıştır. Bu şekilde olmayan durumlarda, her işlem içerisindeki ürün 2. adımdaki ürün sıralamasına göre yeniden sıralanarak uygulamaya devam edilmelidir.

TID	Items
1	{a, b}
2	{b, c,d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}



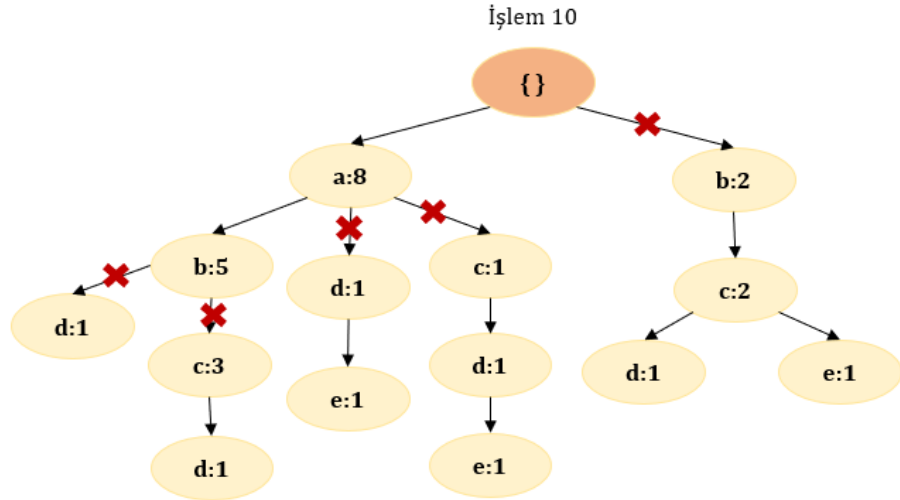
Şekil 4.4. Adım adım FP ağaçlarının oluşturulması

FP ağacının son hali Şekil 4.5 ile gösterilmiştir.



Şekil 4.5. Oluşturulan FP ağacının son hali

Birliktelik kurallarını belirlemek için ağacın her bir düğümündeki nesnelerin destek değerleri kontrol edilir ve destek sayısı eşik destek değerinin altında kalan nesneler, Şekil 4.6'da gösterildiği şekilde kurallara dahil edilmez. Belirlenen dallar budanır.

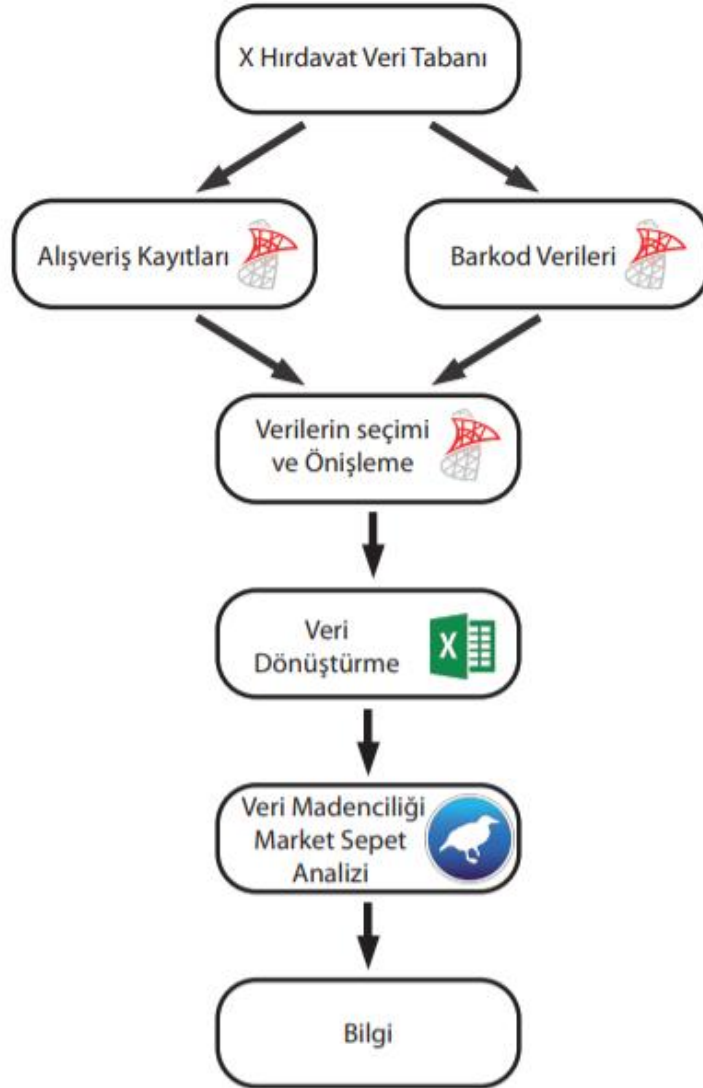


Şekil 4.6. FP ağacı üzerindeki budanacak dalların belirlenmesi

{a, b} : %50 (5/10) şeklinde bir birliktelik kurallı olduğu görülmektedir.

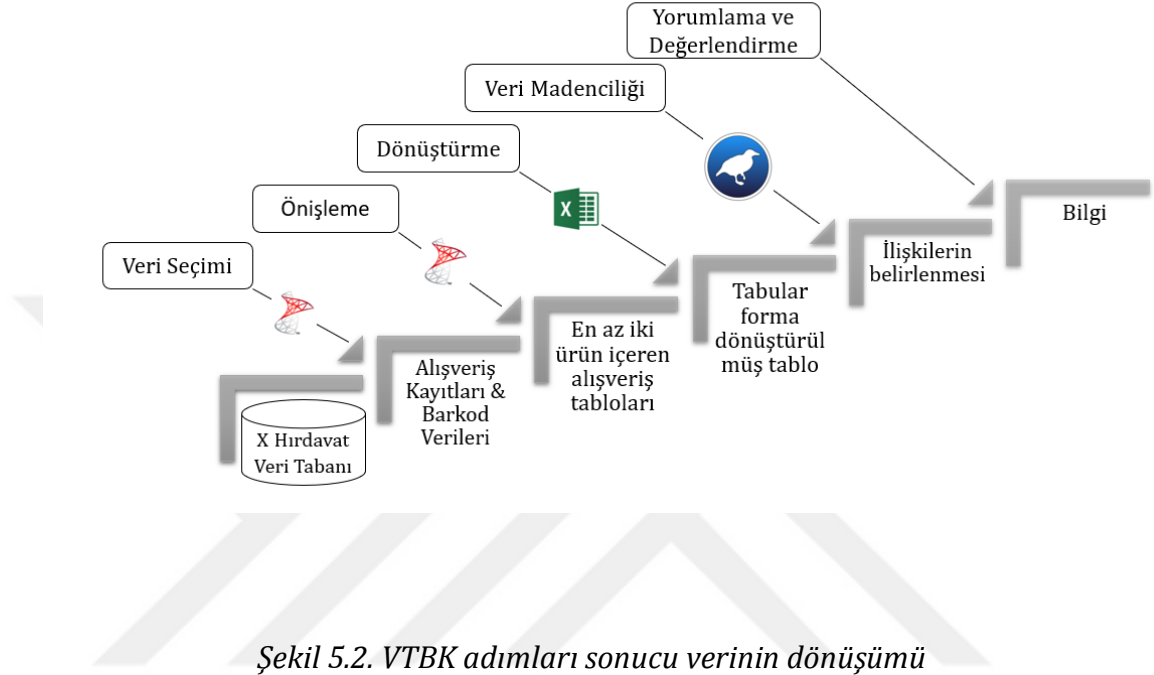
5. METODOLOJİ

Bu çalışmanın amacı, büyük bir hırdavat şirketinin bir şubesine ait alışveriş kayıtlarını kullanarak birlikte satılma eğiliminde olan ürün gruplarını belirlemektir. Çalışmada açık kaynak kodlu bir veri madenciliği yazılımı olan WEKA (Waikato Environment for Knowledge Analysis) programı kullanılmıştır. Programın içerisinde bulunan Apriori ve FP-Growth algoritma özellikleri kullanılarak analiz gerçekleştirilmiştir. Veri tabanı olarak SQL Server 2014 'ten yararlanılmıştır. Şekil 5.1'de çalışmanın kapsamı, akış diyagramı üzerinde gösterilmiştir.



Şekil 5.1. Akış diyagramı

Şekil 5.2’de ham verinin, veri tabanından bilgi keşfi süreci adımları sonucunda aldığı şekiller ve her adım için kullanılan bilgisayar programaları yer almaktadır.



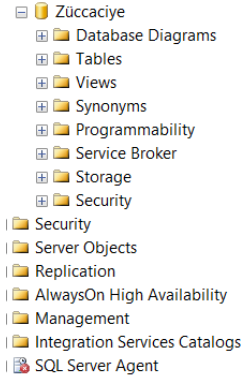
Şekil 5.2. VTBK adımları sonucu verinin dönüşümü

Verilerin Toplanması ve Düzenlenmesi

Bu çalışmada kullanılan veriler, sektörde 50 yılı aşkın tecrübesi ve yirmiye yakın ülke ile aktif iş birliği içerisinde olarak yurtiçi hırdavat piyasasının liderleri arasında yer alan bir şirketin şubesine aittir. Çalışmada şirket gizlilik prensibi nedeni ile şirket ismi gizli tutulacaktır. Gerekli yerlerde şirket ismi, X Hırdavat olarak yer alacaktır.

X Hırdavat, 3081 farklı ürün çeşidine sahiptir. 21.08.2008-20.02.2014 tarihleri arasındaki alışveriş hareketlerinden oluşan veri seti, 81.384 farklı fiş bilgisi içermektedir.

Şekil 5.3 ile SQL veri tabanı üzerinde alışveriş kayıtları ve fiş verilerinin birleştirilmesi ile oluşan, çalışmada kullanılacak ana tablonun bir kesiti gösterilmektedir.



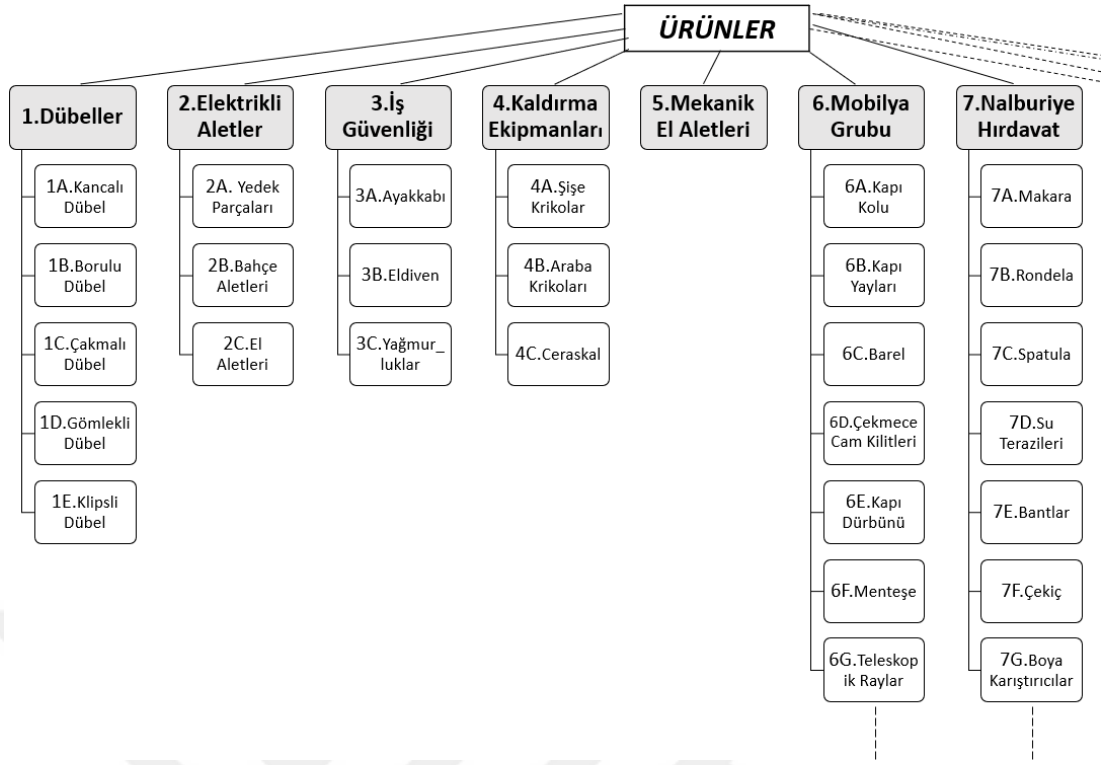
FIS_NO	TARİH	URUN_ADı
220778	162259	2011-06-13 00:00:00.000 TRAPEZ ÇATI VİDASI 5.5 X 50
220779	162484	2011-06-17 00:00:00.000 PİMLİ (KALEM) TEL FIRÇA 24 MM
220780	162765	2011-06-23 00:00:00.000 SDS MATKAP UCU 18 X 300
220781	162833	2011-06-24 00:00:00.000 ELMAS DAIRE TESTERE (TRAPEZ DIŞ) 250 .
220782	162833	2011-06-24 00:00:00.000 BITS UÇ (POZI) PZ 2 X 100 MM
220783	162834	2011-06-24 00:00:00.000 ALÜMİNYUM (MİKNATISLI) SU TERAZİSİ 400 .
220784	162884	2011-06-27 00:00:00.000 SDS MAX 4 ELMAS 20 X 320
220785	162884	2011-06-27 00:00:00.000 SDS MAX KESKİ 400 MM
220786	162892	2011-06-27 00:00:00.000 MATKAP UCU (ELMAS / BETON) 10 X 120
220787	162892	2011-06-27 00:00:00.000 SOMUN ADAPTÖRÜ 10 MM X 65 MM
220788	162896	2011-06-27 00:00:00.000 BITS UÇ (YILDIZ) PH 2 X 50 MM
220789	162937	2011-06-28 00:00:00.000 BITS UÇ (YILDIZ) PH 2 X 50 MM
220790	161805	2011-06-07 00:00:00.000 SDS MATKAP UCU 18 X 300
220791	161805	2011-06-07 00:00:00.000 SDS MATKAP UCU 20 X 300
220792	162650	2011-06-21 00:00:00.000 KONİK VİDA (DALGA) TEL FIRÇA 115 MM
220793	162690	2011-06-21 00:00:00.000 BITS UÇ (TORK) T 30 X 75 MM
220794	162708	2011-06-22 00:00:00.000 SDS MATKAP UCU 11 X 160

Şekil 5.3. X Hırdavat alışveriş kayıtları SQL tablosundan bir kesit

SQL veri tabanında, ürün bilgileri ve alışveriş kayıtları şeklinde ayrı ayrı bulunan tablolar SQL sorguları ile birleştirilerek, belirlenen dönem içerisindeki fişler ve fişler içerisindeki ürünlerin bilgilerini içeren yeni bir tablo oluşturulmuştur.

SQL 'de fiş numarası, alışveriş tarihi, ürün, ürün grubu sütunlar olacak şekilde oluşturulan tablo Excel'e aktarılmıştır. İlk olarak tablo incelendiğinde 293 farklı alt ürün grubu olduğu görülmüştür. Fakat tabloda birçok ürünün grubunun eksik olduğu gözlemlenmiştir.

Veriler ayrıntılı bir şekilde incelenerek içerisindeki eksiklikler tamamlanmış, böylece veriler içerisindeki gürültü temizlenmiştir. Ayrıca ürün grupları 13 temel kategori ve toplam 93 alt kategori olacak şekilde yeniden düzenlenmiştir. Ana kategoriler 1 ile 13 arasındaki rakamlar ile, her ana kategoriye ait alt kategoriler ise A, B, C... şeklinde harfler ile ifade edilmiştir. Düzenlenen tablonun bir kısmı Şekil 5.4 'te yer almaktadır.



Şekil 5.4. Düzenlenen alt ürün grubu tablosundan bir kesit

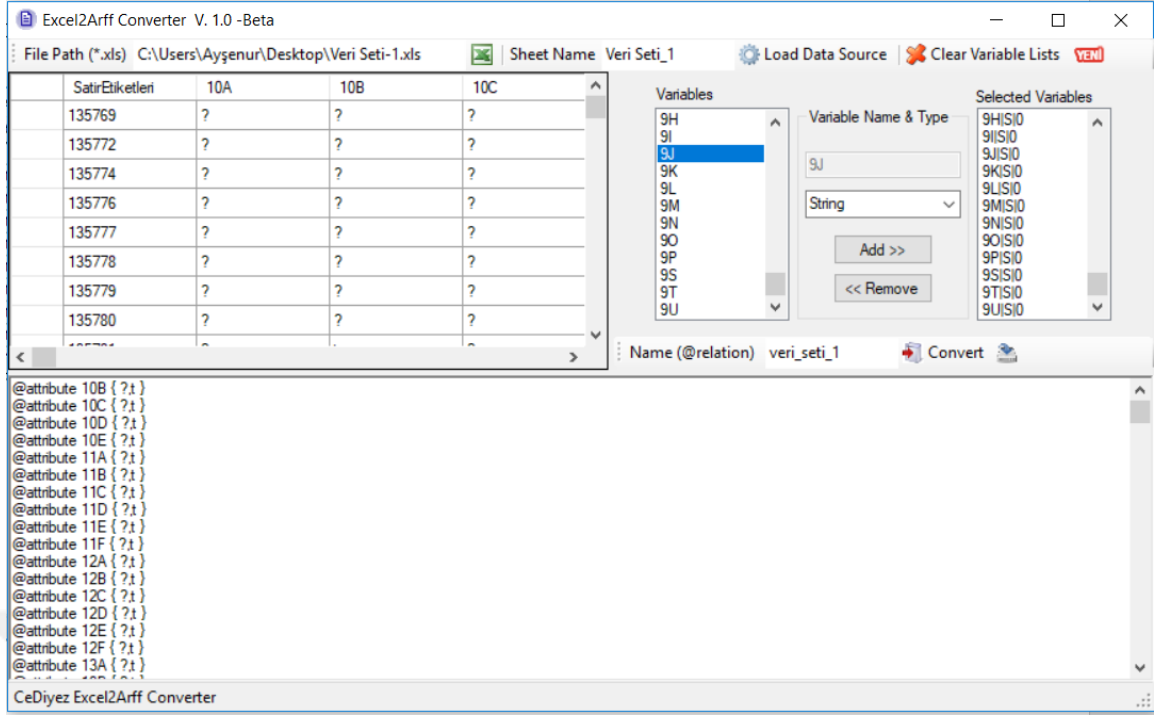
Amacımız fişler içerisinde birlikte satılan ürün gruplarını bulmak olduğundan, satış adedine bakılmaksızın sadece tek bir ürün grubuna ait ürün içeren fişler elenmelidir. Oluşturulan tablo 'SQL Import and Export Wizard' yardımı ile SQL'e aktarılarak gerekli SQL sorguları ile, tablo en az iki farklı ürün grubundan ürün içeren fişler kalacak şekilde yeniden düzenlemiştir. Düzenleme sonunda toplam 54.564 adet en az iki ürün grubu içeren fiş kalmıştır.

Çıkarılan kuralların başarısını bir sonraki dönemler ile kıyaslayabilmek için ardışık yılların bilgilerini içeren veri seti, uygun bir oranla ikiye ayrılmıştır. Veri setinin ilk dönemlerine ait kayıtları içeren veri seti 'Veri Seti -1', son dönemlerine ait kayıtları içeren veri seti, 'Veri Seti-2' olarak adlandırılmıştır. Sırası ile alışveriş tarihi, fiş numarası, ürün adı, ürün grubu sütunlarını içeren veri setinin WEKA 'nın anlayabileceği forma dönüştürülmesi gerekmektedir. Şekil 5.5'te görüldüğü gibi, Excel'in pivot tablo özelliğinden yararlanılarak

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	135789 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?
21	135791 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
22	135796 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
23	135797 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
24	135798 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
25	135812 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
26	135816 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
27	135818 ?	t	?	?	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?
28	135820 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
29	135821 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
30	135822 ?	?	?	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	?
31	135823 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
32	135824 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
33	135826 ?	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	?	?	?
34	135833 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
35	135836 ?	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	?	?	?
36	135837 ?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	?	?	?	?
37	135838 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
38	135839 ?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	?	?	?	?
39	135841 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
40	135843 ?	t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
41	135844 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
42	135846 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
43	135849 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
44	135850 t	t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
45	135851 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
46	135853 ?	?	?	?	?	?	t	?	?	?	t	?	?	?	?	t	?	?	?
47	135855 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
48	135856 ?	t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
49	135858 ?	t	?	?	?	?	?	?	?	t	?	?	?	?	t	?	?	?	?
50	135860 ?	?	?	?	?	?	?	?	?	t	t	?	?	?	?	?	?	?	?
51	135861 ?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
52	135862 t	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	?	?	?

Şekil 5.6. Veri Seti-1'in tabular forma çevrilmiş Excel tablosu

Oluşturulan Excel dosyası 'arff' formatına çevrilerek WEKA 'ya yüklenebilecek biçime dönüştürülmesi gerekmektedir. Dönüştürme işlemi için 'Excel2Arff Converter V.1.0-Beta' programı kullanılmıştır. Program yalnızca '.xls' uzantılı dosyaları kabul ettiğinden, Excel'de oluşturulmuş pivot tablo .xls formatında kaydedilmiştir. Sonrasında Şekil 5.7'de gösterildiği şekilde programa yüklenmiştir.



Şekil 5.7. Excel2Arff Converter programı çalışma sayfası

Arff. formatına dönüştürülecek .xls dosyası 'File path' bölümünden dönüştürücüye yüklenir. 'Sheet Name' kısmına excel çalışma sayfasının ismi yazılmalıdır. 'Load Data Source' butonuna bastığımızda tüm değişkenlerimiz programa yüklenmiş olur. 'Variables' bölümünden değişkenlerin tipini belirleyip 'Add' butonu ile kullanılacak değişkenleri seçtikten sonra, 'name' kısmından çalışma sayfasına isim verilebilir. 'Convert' butonu tıkladığımızda ise alt pencerede .arff formatına dönüştürülmüş verilerimizi görebiliriz.

Arff formatına çevrilmiş Veri Seti-1'in içerisindeki niteliklerin bir kısmı Şekil 5.8 'de gösterilmiştir.

```
VeriSeti_1.arff - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
@attribute 10C { ?,t }
@attribute 10D { ?,t }
@attribute 10E { ?,t }
@attribute 11A { ?,t }
@attribute 11B { ?,t }
@attribute 11C { ?,t }
@attribute 11D { ?,t }
@attribute 11E { ?,t }
@attribute 11F { ?,t }
@attribute 12A { ?,t }
@attribute 12B { ?,t }
@attribute 12C { ?,t }
@attribute 12D { ?,t }
@attribute 12E { ?,t }
@attribute 12F { ?,t }
@attribute 13A { ?,t }
@attribute 13B { ?,t }
@attribute 13C { ?,t }
@attribute 13D { ?,t }
@attribute 13E { ?,t }
@attribute 13F { ?,t }
```

Şekil 5.8. Kategorilerin .arff formatındaki görüntüsünden bir kesit

Veri Seti-1 için verilerin .arff formatında görünüşü Şekil 5.9 ile gösterilmiştir.

```
VeriSeti_1.arff - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
@data
378,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,
135763,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,t,?,
,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
135765,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
,?,?,?,?,?,t,?,?,?,?,?,t,?,?,?,?,?,t,?,?,?,?,?,t,?,?,?,?,?,
135769,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
135772,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
135774,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
135776,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
135777,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
135778,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,
,?,?,?,?,t,t,?,?,?,?,t,t,?,?,?,?,t,t,?,?,?,?,t,t,?,?,?,?,t,t,?,?,?,?,
135779,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,t,?,?,?,?,?
```

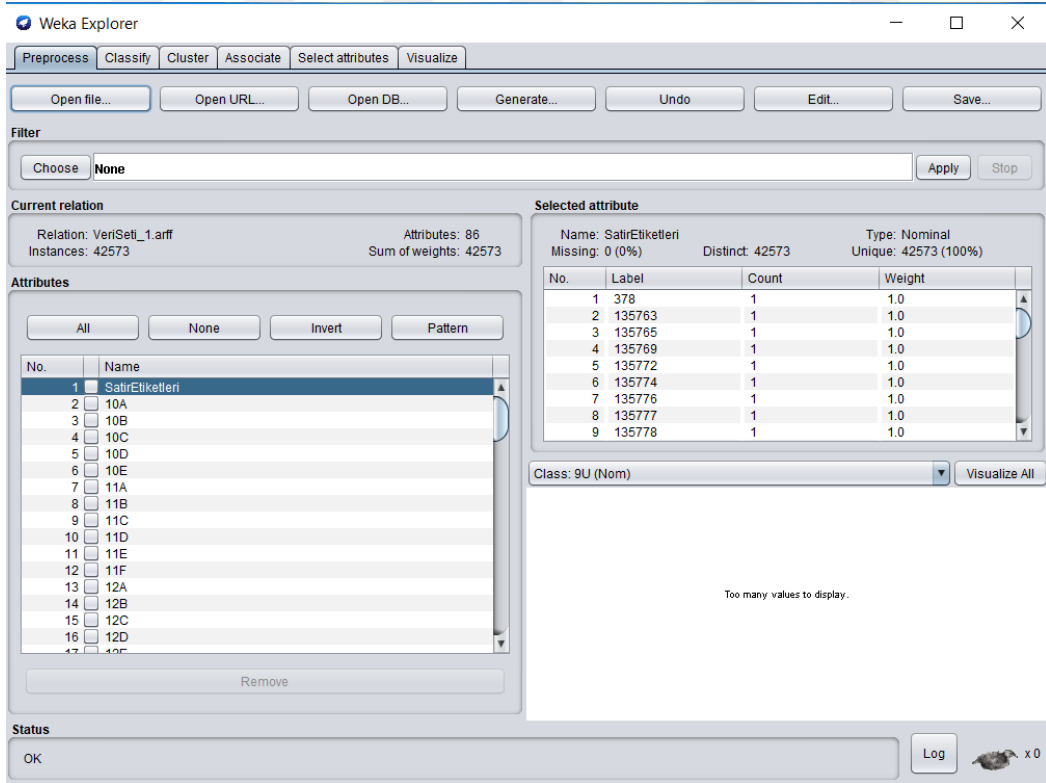
Şekil 5.9. Arff formatına dönüştürülmüş verilerden bir kesit

Böylelikle .arff formatına dönüştürülen veriler WEKA'ya yüklenebilecek hale gelmiştir. Şekil 5.10'da programın giriş ekranı yer almaktadır.



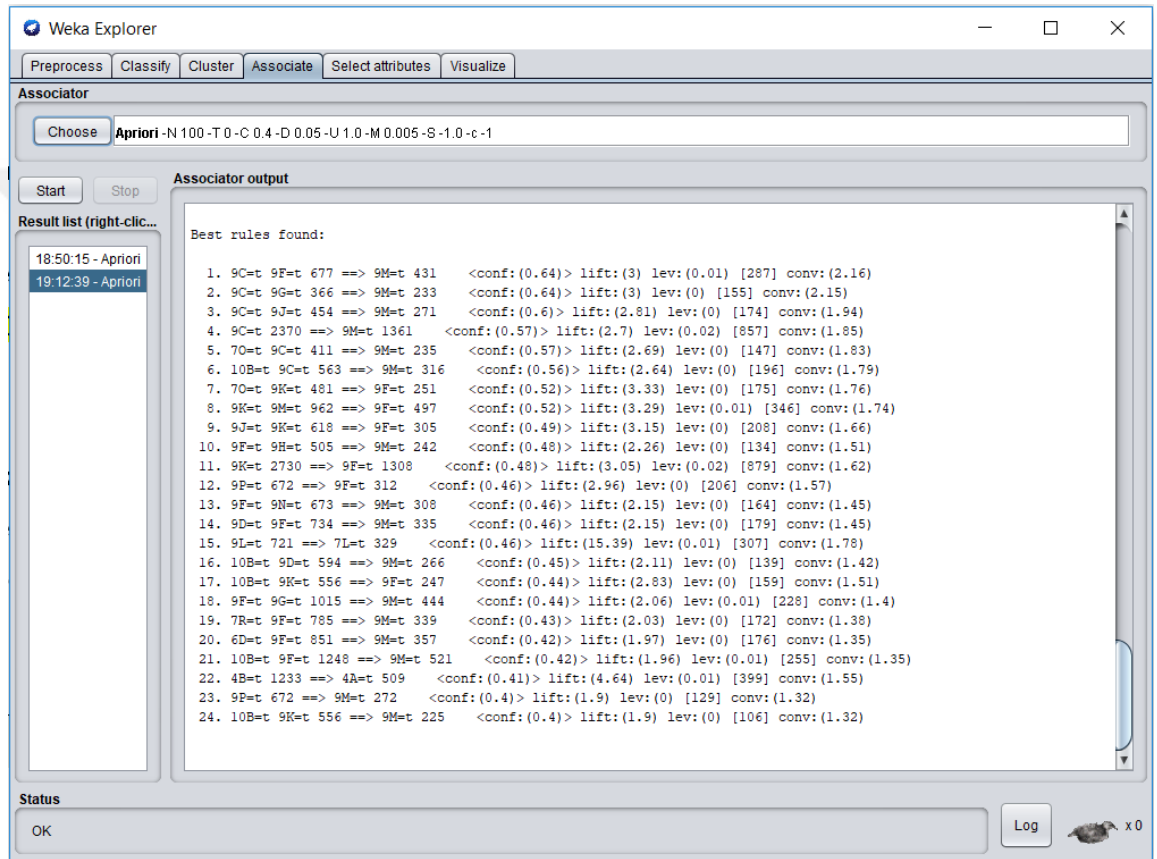
Şekil 5.10. WEKA programı giriş ekranı

Şekil 5.11 'de görüldüğü gibi; programında, 'Preprocess' özelliği altındaki 'Open file' seçeneğinden, üzerinde veri madenciliği gerçekleştirilecek olan .arff formatlı dosya seçilir.



Şekil 5.11. WEKA programı 'Preprocess' ekranının görüntüsü

WEKA programında Associate seçeneği altındaki Apriori ve FP-Growth algoritmaları kullanılarak 'arff' formatına dönüştürülmüş veri seti üzerinden birliktelik kuralları çıkarımı gerçekleştirilmiştir. Bu boyuttaki bir veri seti için, yapılan denemeler sonucunda en iyi kuralların %40 güven ve %0.5 destek değerleri altında çıkarıldığı görülmüştür. Şekil 5.12' de WEKA programında Veri Seti-1 üzerinde Apriori algoritması sonucu çıkarılan kurallar görülmektedir.



Şekil 5.12. Veri Seti -1 için Apriori sonuçları

Veri Seti-1 için Weka programında Apriori algoritması çalıştırılmış ve belirlenen minimum güven ve destek değerlerini sağlayan 24 kural çıkarıldığı görülmüştür.

Veri Seti-1 üzerinde Apriori algoritmasının çalışma süresi 781 sn. olarak ölçülmüştür.

Şekil 5.12'de birliktelik kurallarının kodlarla belirtildiği görülmektedir. Çizelge 5.1, bu kodlara karşılık gelen kategori isimleri ve en iyi on kuralı içermektedir.

Çizelge 5.1. Veri Seti-1 için Apriori Algoritması sonuçları

Öncüller	Sonuç	Güven	Destek	Lift
Murç ve Keskiler, Bits Uçlar Adaptörler (677)	Matkap Uçları (431)	%64	0.01	3
Murç ve Keskiler, Panç Grubu (366)	Matkap Uçları (233)	%64	0.005	3
Murç ve Keskiler, Tel Fırçalar (454)	Matkap Uçları (271)	%60	0.006	2.81
Murç ve Keskiler (2370)	Matkap Uçları (1361)	%57	0.031	2.70
Metreler, Somun Adaptörleri (411)	Matkap Uçları (235)	%57	0.005	2.69
Elmas Testere, Murç ve Keskiler (563)	Matkap Uçları (316)	%56	0.007	2.64
Metreler, Somun Adaptörleri (481)	Bits Uçlar Adaptörler (251)	%52	0.005	3.33
Somun Adaptörleri, Matkap Uçları (962)	Bits Uçlar Adaptörler (497)	%52	0.011	3.29
Tel Fırçalar, Somun Adaptörler (618)	Bits Uçlar Adaptörler (305)	%49	0.007	3.15
Bits Uçlar Adaptörler, Eğeler (505)	Matkap Uçları (242)	%48	0.005	2.26

Bu tablonun ilk kuralını şu şekilde yorumlayabiliriz: {Murç ve Keskiler, Bits Uçlar Adaptörler} grup ürünlerini içeren 677 fişin 431 tanesi {Matkap Uçları} grubundan da ürün içermektedir. Yani 'Murç ve Keski' ve 'Bits Uçlar Adaptörler' ürünü alan müşteriler %64 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin %1'inde bu üç grubun ürünleri birlikte satın alınmaktadır. Lift değerlerini ise kuralların ilginçlik değeri olarak değerlendirebiliriz. Lift değeri ne kadar büyük ise kural o kadar ilginç demektir. Ayrıca lift değeri ile bir ürünün satışının diğer bir ürünün satışına yüzde etkisi de hesaplanabilir. İlk kural için lift değerini aşağıdaki formülü kullanarak şu şekilde yorumlayabiliriz;

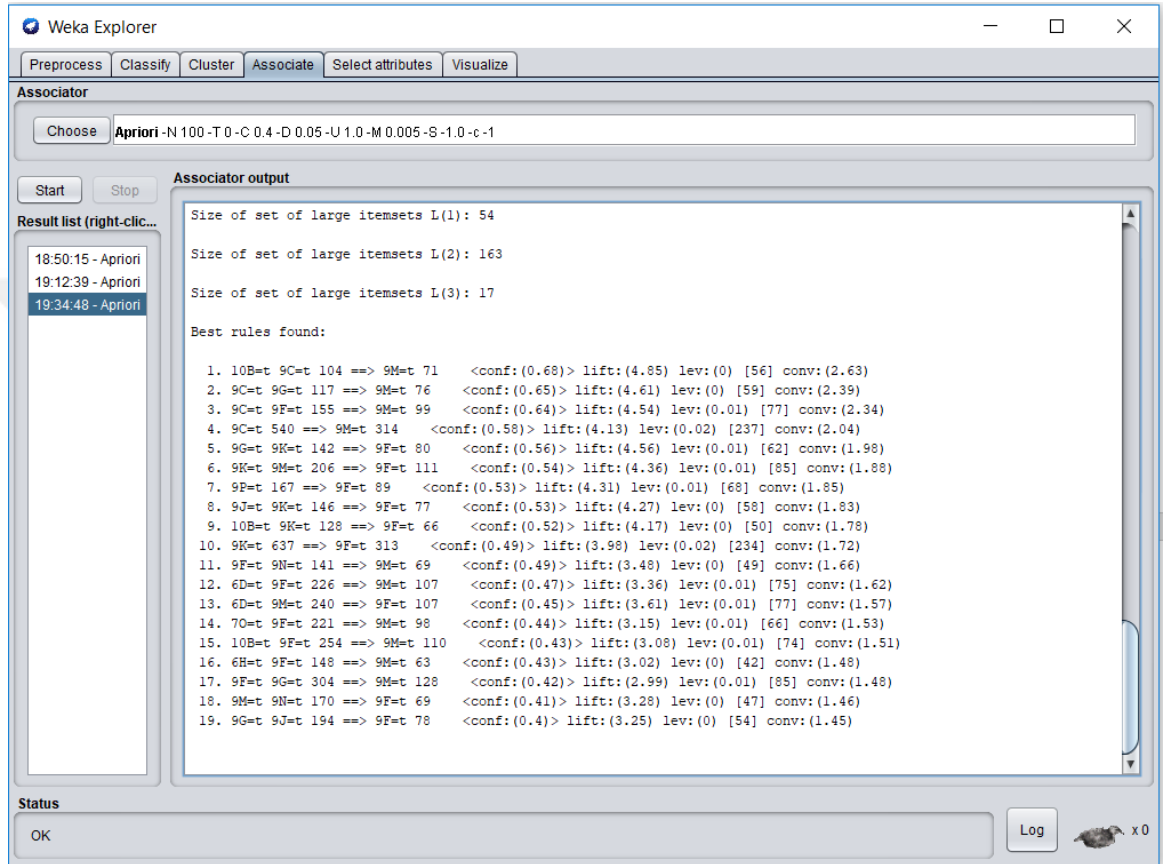
$$\text{Yüzde artış} = (3 - 1) * 100 = \%200$$

Bu da bir fişin 'Murç ve Keski' ve 'Bits Uçlar Adaptörler' grubundan ürün içeriyor olması 'Matkap Uçları' grubundan da ürün içerme olasılığını %200 arttırır şeklinde yorumlanabilir (Anon. 2018) . Sırası ile tablodaki diğer kurallar şu şekilde yorumlanır;

- 'Murç ve Keski' ve 'Panç grubu' ürünü alan müşteriler %64 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin 0.005'inde bu ürünler birlikte görülmektedir.
- 'Murç ve Keski' ve 'Tel Fırça' ürünü alan müşteriler %60 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin 0.006'sında bu ürünler birlikte görülmektedir.
- 'Murç ve Keski' ürünü alan müşteriler %57 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin %0.031'inde bu ürünler birlikte görülmektedir.
- 'Metre' ve 'Somun Adaptörü' ürünü alan müşteriler %57 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin 0.005'inde bu ürünler birlikte görülmektedir.
- 'Elmas Testere' ve 'Murç ve Keski' ürünü alan müşteriler %56 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin 0.007'sinde bu ürünler birlikte görülmektedir.
- 'Metre' ve 'Somun Adaptörü' ürünü alan müşteriler %52 olasılıkla 'Bits Uçlar Adaptörler' da almaktadırlar. Tüm alışverişlerin 0.005'inde bu ürünler birlikte görülmektedir.
- 'Somun Adaptörü ' ve 'Matkap Uçları' ürünü alan müşteriler %52 olasılıkla 'Bits Uç Adaptörler' da almaktadırlar. Tüm alışverişlerin 0.011'inde bu ürünler birlikte görülmektedir.
- 'Tel Fırçalar' ve 'Somun Adaptörü' ürünü alan müşteriler %49 olasılıkla 'Bits Uçlar Adaptörler' da almaktadırlar. Tüm alışverişlerin 0.007'sinde bu ürünler birlikte görülmektedir.
- 'Bits Uç Adaptörler ' ve 'Eğeler' ürünü alan müşteriler %48 olasılıkla 'Matkap Uçları' da almaktadırlar. Tüm alışverişlerin 0.005'inde bu ürünler birlikte görülmektedir.

Sonrasında Veri Seti-2 için aynı minimum güven ve destek değerleri altında Apriori algoritması çalıştırılmıştır. Belirlenen minimum güven ve minimum destek değerlerini sağlayan 19 kural, 41 sn. olarak ölçülen süre içerisinde çıkarıldığı görülmüştür.

Şekil 5.13, WEKA programının Apriori algoritması sonuçlarını içermektedir.



Şekil 5.13. Veri Seti -2 için Apriori algoritması sonuçları

Çıkarılan kurallara bakıldığında ilk kural şu şekilde yorumlanabilir; 'Elmas Testere' kategorisi ve 'Murç ve Keskiler' kategorisinden ürün alan 104 müşterinin 71 tanesi 'Matkap Uçları' kategorisinden de ürün almıştır. Yani 'Elmas Testere kategorisi' ve 'Murç ve Keskiler' kategorisinden ürün alanlar %68 olasılıkla 'Matkap Uçları' kategorisinden de ürün alırlar. Tüm alışverişlerin 0.005'inde bu ürünler birlikte görülmektedir. 4.85 olan lift değeri ise; fişin 'Elmas Testere' ve 'Murç ve Keskiler' kategorisinden ürün içeriyor olması

'Matkap Uçları' kategorisinden de ürün içeriyor olma olasılığını %385 arttırır şeklinde yorumlanabilir. Diğer kurallar da bu doğrultuda, benzer şekilde yorumlanabilir.

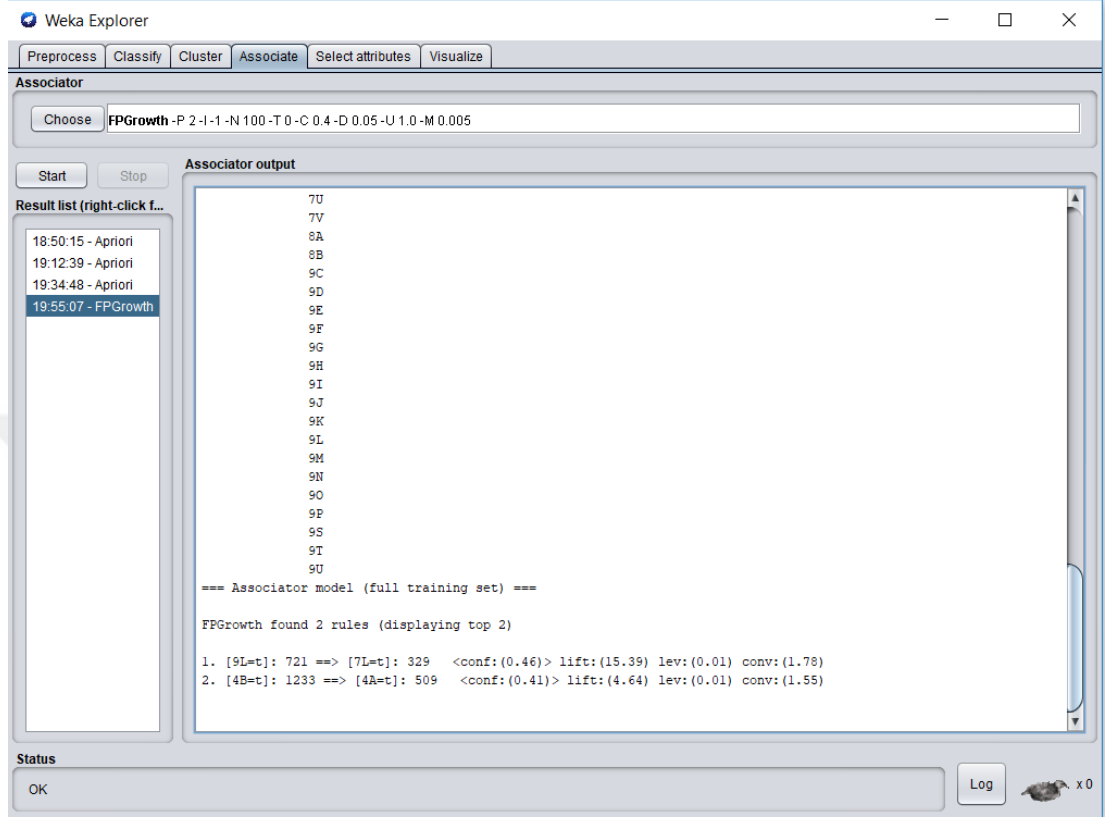
Çizelge 5.2 , kategoriler arasındaki ilişkiyi açıklayacak kuralların en iyi on tanesini göstermektedir.

Çizelge 5.2. Veri Seti-2 için Apriori Algoritması Sonuçları

<i>Öncüller</i>	<i>Sonuç</i>	<i>Güven</i>	<i>Destek</i>	<i>Lift</i>
Elmas Testere, Murç ve Keskiler (104)	Matkap Uçları (71)	%68	0.005	4.85
Murç ve Keskiler, Panç Grubu (117)	Matkap Uçları (76)	%65	0.006	4.61
Murç ve Keskiler, Bits Uçlar Adaptörler (155)	Matkap Uçları (99)	%64	0.008	4.54
Murç ve Keskiler (540)	Matkap Uçları (314)	%58	0.026	4.13
Panç Grubu, Somun Adaptörleri (142)	Bits Uçlar Adaptörler (80)	%56	0.006	4.56
Somun Adaptörleri, Matkap Uçları (206)	Bits Uçlar Adaptörler (111)	%54	0.009	4.36
SDS Plus Adaptörleri (167)	Bits Uçlar Adaptörler (89)	%53	0.007	4.31
Tel Fırçalar, Somun Adaptörleri (146)	Bits Uçlar Adaptörler (77)	%53	0.006	4.27
Elmas Testere, Somun Adaptörleri (128)	Bits Uçlar Adaptörler (66)	%52	0.005	4.17
Somun Adaptörleri (637)	Bits Uçlar Adaptörler (313)	%49	0.026	3.98

Weka programında birliktelik kurallarının belirlenmesi için kullanılan diğer bir algoritma da FP-Growth algoritmasıdır. Aynı şekilde Veri Seti-1 ve Veri Seti-2 veri setleri üzerinde bu algoritma ayrı ayrı çalıştırılmış, Şekil 5.14 ve Şekil 5.15 ile gösterilen kurallar çıkarılmıştır.

Şekil 5.14'te görüldüğü gibi Veri Seti-1 üzerinde FP-Growth Algoritması uygulanması sonucunda belirlenen eşik destek ve güven kriterlerini sağlayan iki adet kural çıkarılmıştır.



Şekil 5.14. Veri Seti-1 için FP-Growth Algoritması sonuçları

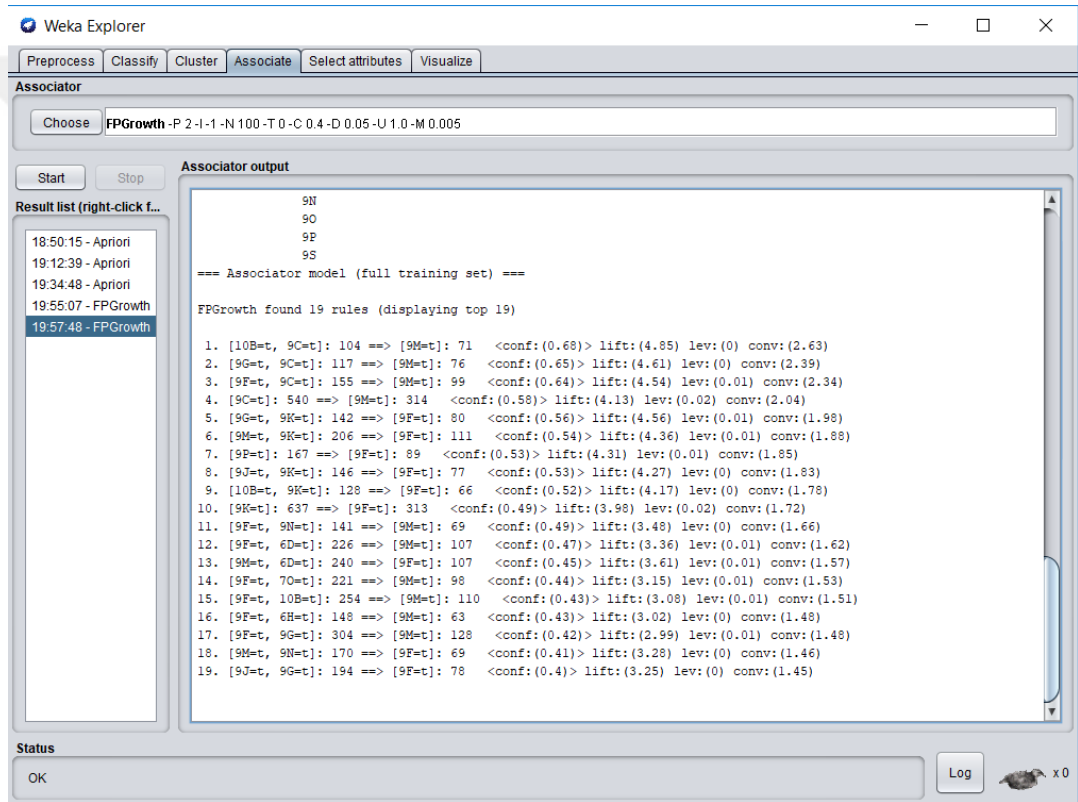
Şekil 5.14'te kodlarla gösterilen birliktelik kuralları, Çizelge 5.3 ile kodlara karşılık gelen kategoriler belirtilmiş, çıkarılan kurallar gösterilmiştir. FP-Growth algoritması Veri Seti-1 üzerinde 1sn. içerisinde sonuç vermiştir.

Çizelge 5.3. Veri Seti-1 için FP-Growth algoritması sonuçları

Öncüller	Sonuç	Güven	Destek	Lift
Kaplin ve Soketler (721)	Spiral Hava Hortumları (329)	%46	0.007	15.39
Araba Krikoları (1233)	Şişe Krikolar (509)	%41	0.011	4.64

Weka'da FP-Growth algoritması ile Veri Seti-1 üzerinde iki uygun kuralın çıkarıldığı görülmüştür. İlk kural; 'Kaplin ve Soket' grubu ürün bulunan 721, fişin 329 tanesinde 'Spiral Hava Hortumları' grubundan da ürün bulunduğunu göstermektedir. Çıkan kurallar Apriori algoritması sonucu çıkarılan kurallar ile karşılaştırıldığında, FP-Growth algoritmasının yüksek güven değerine sahip kuralları atladığını göstermektedir.

Şekil 5.15'te görüldüğü gibi Veri Seti-2 üzerinde FP-Growth Algoritması uygulanması sonucunda belirlenen eşik destek ve güven kriterlerini sağlayan 19 adet kural çıkarılmıştır.



Şekil 5.15. Veri Seti-2 için FP-Growth Algoritması sonuçları

Şekil 5.15'te kodlarla gösterilen birliktelik kuralları, Çizelge 5.4 ile kodlara karşılık gelen kategoriler belirtilmiş, çıkarılan kurallar gösterilmiştir. FP-Growth algoritması Veri Seti-2 üzerinde 1sn.'den az bir sürede sonuç vermiştir.

Çizelge 5.4. Veri Seti-2 için FP-Growth Algoritması sonuçları

Öncüller	Sonuç	Güven	Destek	Lift
Elmas Testere, Murç ve Keskiler (104)	Matkap Uçları (71)	%68	0.005	4.85
Panç Grubu, Murç ve Keskiler (117)	Matkap Uçları (76)	%65	0.006	4.61
Bits Uçlar Adaptörleri, Murç ve Keskiler (155)	Matkap Uçları (99)	%64	0.008	4.54
Murç ve Keskiler (540)	Matkap Uçları (314)	%58	0.026	4.13
Panç Grubu, Somun Adaptörleri (142)	Bits Uçlar Adaptörler (80)	%56	0.006	4.56
Matkap Uçları, Somun Adaptörleri (206)	Bits Uçlar Adaptörler (111)	%54	0.009	4.36
SDS Plus Adaptörleri (167)	Bits Uçlar Adaptörler (89)	%53	0.007	4.31
Tel Fırçalar, Somun Adaptörleri (146)	Bits Uçlar Adaptörler (77)	%53	0.006	4.27
Elmas Testere, Somun Adaptörleri (128)	Bits Uçlar Adaptörler (66)	%52	0.005	4.17
Somun Adaptörleri (637)	Bits Uçlar Adaptörler (313)	%49	0.026	3.98

Veri Seti-2 üzerinde FP-Growth algoritması çalıştırıldığında ise ilk on kuralın Apriori algoritması ile aynı olduğu görülmektedir. Ancak iki algoritmanın çalışma zamanına bakıldığında FP-Growth algoritması'nın Apriori algoritmasından çok daha önce sonuç verdiği görülmüştür.

6. SONUÇ VE ÖNERİLER

Teknolojinin gelişmesi ile birlikte şirket veri tabanlarındaki veri hacimleri her geçen gün artmaktadır. Şirketlerin, rekabetin giderek arttığı günümüzde, ayakta kalabilmeleri için ellerindeki veriyi işleyerek anlamlı hale dönüştürmeleri gerekir. Veri madenciliği sayesinde büyük veri yığınları arasından anlamlı bilgiler çıkarılır. Veriler üzerinde, çeşitli analizler ve raporlama işlemleri yapılarak şirketler için özet bilgiler elde edilir. Gelecek ile ilgili kararlar, geçmiş yıllardaki benzer sonuçları bilindiğinden daha güvenilir bir şekilde verilebilir. Veri madenciliği, müşteri satın alma alışkanlıklarının belirlenmesi, satış tahmini, ürünlerin birlikte satılma eğilimlerinin belirlenmesi, kaybedilen müşteri analizi ve bunun gibi birçok alanda kullanılmaktadır.

Bu çalışmada, WEKA 3.8 veri madenciliği programının Apriori ve FP-Growth algoritmaları kullanılarak gerçek veriler üzerinde birliktelik kuralları madenciliği gerçekleştirilmiştir. Böylelikle birlikte satılma eğiliminde olan ürün gruplarını belirlemek amaçlanmıştır. Yapılan analiz sonucunda şirketin analiz dönemleri içerisinde en güçlü birlikteliklerin; {Murç ve Keskiler}, {Bits Uçlar}, {Matkap Uçları}, {Panç Grubu}, {Metreler}, {Somun Adaptörleri} ve {Eğeler} arasında olduğu görülmüştür. X Hırdavat'ın ardışık beş buçuk yıllık satış bilgilerinden oluşan veri seti, uygun bir oranda Veri Seti-1 ve Veri Seti-2 ismiyle ikiye ayrılmıştır.

İlk olarak Veri Seti-1 üzerinde Apriori ve FP-Growth algoritmaları ayrı ayrı uygulanmıştır. Algoritmalar performans açısından karşılaştırıldığında, bu boyuttaki bir veri setinde FP-Growth algoritmasının Aprioriye göre 781 kat daha hızlı sonuç verdiği görülmüştür. Ancak çıkarılan kurallara bakıldığında FP-Growth 'un yüksek güven değerine sahip ilk 14 kuralı bulamadığı görülmektedir. Bu durumda Veri Seti-1 için Apriori algoritması ile çalışmak, FP-Growth algoritması ile çalışmaktan daha güvenli ve avantajlı olacaktır. Sonrasında Veri Seti-2 için Apriori ve FP-Growth algoritmaları uygulanmış, yine FP-Growth un 1saniyeden daha az bir sürede sonuç verdiği Apriori'nin ise 41saniyede sonuç verdiği görülmüştür. Bununla birlikte Veri Seti-2 üzerinden

çıkarılan kural sayılarının ve sıralamalarının iki algoritmada da aynı olduğu görülmüştür. Çıkarılan kurallar ve algoritmaların çalışma süreleri göz önüne alındığında Veri Seti-2 için FP-Growth algoritmasının daha kullanışlı olacağı söylenebilir.

Sırasıyla, Veri Seti-1 ve Veri Seti-2'nin Apriori algoritması sonucu çıkarılan kurallarını içeren Çizelge 5.1 ile Çizelge 5.2 karşılaştırılmıştır. Çizelge 5.1'deki önemli güven değerine sahip kuralların Çizelge 5.2'de de yer aldığı görülmektedir. Örneğin; {Murç ve Keskiler, Bits Uçlar Adaptörler} → {Matkap Uçları} kuralı, %64 güven değeri ile ilk dönem verilerinin bulunduğu Veri Seti-1 içerisinde çıkarılan bir kuraldır. Bu kuralın ardışık dönem verilerinin bulunduğu, Veri Seti-2 içerisinde de %64 güven değeri ile görülmesi, Veri Seti-1 için çıkarılan kuralların gerçek hayatta da büyük ölçüde geçerli olduğu şeklinde yorumlanabilir.

Çıkarılan kuralların geneline bakıldığında matkap uçları ve bits uç adaptörler kategorilerinin, diğer kategoriler ile güçlü bir bağlantı içerisinde olduğu bilgisine ulaşılmıştır.

İlk dönemlerde çıkarılan kurallardan işletme habersizdir. Buna rağmen ardışık dönemde çıkarılan kurallardan, satışların paralel olduğu görülmüştür. Eğer işletme bu birlikteliklerin farkında olsaydı, bu kuralları katalog düzenlemesi, mağaza içi yerleşim düzeni, depo tasarımı, indirim- promosyon düzenlemeleri ve karar destek sistem tasarımları gibi konularda kullanarak satışlarında, buna bağlı olarak ta gelirlerinde önemli ölçüde artış sağlayabilirdi.

Çalışma, ilerleyen zamanlarda şu şekilde geliştirilebilir:

- Mevsimin hırdavat sektörü ile ilişkisi üzerinde çalışarak, mevsimsel birliktelik kuralları çıkarılabilir.
- Yıllara göre firmanın satışları analiz edilerek, trend analizleri gerçekleştirilebilir.
- Çıkarılan birliktelik kuralları doğrultusunda yeni bir mağaza yerleşim düzeni öneri çalışması gerçekleştirilebilir.

- Yine mevsimler birliktelik kuralları araştırılıp, inşaat sektöründeki mevsimsel hareketlenmenin hırdavat sektörü ile olan ilişkisi araştırılabilir.
- Ürünler arasındaki ilişki bilgisi kullanılarak, yeni bir depo tasarım çalışması yapılabilir.
- Verilerin daha kapsamlı hale getirilmesi ile örneğin müşteri bilgilerini de içeren alışveriş kayıtlarını içermesi ile, müşteriler kendi içlerinde sınıflandırılabilir. Böylece işletme değerli müşterilerinin farkına vararak, onlara daha kaliteli hizmet verebilir.



KAYNAKLAR

- Agrawal R., Imielinski T., Swami A., 1993. Mining Association in Large Databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data - SIGMOD '93* 207,13(2-3),161-180.
- Agrawal R., Srikant R., 1994. Fast Algorithms for Mining Association Rules.,94 *Proceedings of the 20th International Conference on Very Large Data Bases* 1215,487-499.
- Akpınar H., 2014. *DATA Veri Madenciliği Veri Analizi*. Papatya Yayıncılık Eğitim.
- Akpınar H., 2000. "Veri Tabnlarında Bilgi Keşfi ve Veri Madenciliği." *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 1,1-22.
- Alpaydın E., 2000. Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, Pp. 1-10, *Bilişim 2000 Eğitim Semineri*, İstanbul, Boğaziçi Üniversitesi.
- Anon., 2018. Ver Tabanı ve Veri Ambarı Arasındaki Farklar, EsDifferent. Erişim Tarihi: 02.02.2018.
<https://tr.esdifferent.com/difference-between-database-and-data-warehouse>
- Anon., 2018. "Association Analysis Simplified." *Analytics Tarinings*. Erişim tarihi:10.12.2017.<http://analyticstrainings.com/?p=151>
- Arslan, Halil. 2008. Sakarya Üniversitesi Web Sitesi Erişim Kayıtlarının Web Madenciliği İle Analizi, Sakarya Üniversitesi.
- Beyer K., Goldstein J., Ramakrishnan R., Shaft U., 1999. When Is 'Nearest Neighbor' Meaningful? ,Pp. 217-35 , *International Conference of Database Theory*, Madison.
- Bircan H., 2004.Lojistik Regresyon Analizi : Tıp Verileri Üzerinde Bir Uygulama., *Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 2,185-208.
- Cabena, P., Hadjinian P., Stadler R., Verhees J., Zanasi A., 1998. *Discovering Data Mining. From Concept to Implemantation*, New Jersey, Prentice Hall.
- Ceylan Z., 2014. Shelf Arrangement Via Data Mining Techniques and A Case Study Shelf Arrangement Via Data Mining, Marmara University.
- Chen Y. L., Tang K.,Shen R. J., Hu Y. H., 2005. Market Basket Analysis in a Multiple Store Environment ,*Decision Support Systems* 40(2),339-354.
- Chiang W. Y., 2011. To Mine Association Rules of Customer Values via a Data Mining Procedure with Improved Model: An Empirical Case Study, *Expert Systems with Applications* 38(3),1716-1722.

- Cios K. J., Pedrycz W., Swiniarski R. W., Kurgan L. A., 2007. *Data Mining: A Knowledge Discovery Approach*, (1st ed.) ,New York, Springer Science.
- Çakmak, Z., Uzgören N., Keçek G., 2005. Kümeleme Analizi Teknikleri İle İllerin Kültürel Yapılarına Göre Sınıflandırılması ve Değişimlerinin İncelenmesi, *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, (12),15–36.
- Çil İ.,2012. Consumption Universes Based Supermarket Layout through Association Rule Mining and Multidimensional Scaling, *Expert Systems with Applications*,39(10),8611–8625. Elsevier .
- Dhanabhakya M., Punithavalli M., 2013. An Efficient Market Basket Analysis Based on Adaptive Association Rule Mining with Faster Rule Generation Algorithm, *The SIJ Transactions on Computer Science Engineering & Its Applications (CSEA)*, 1(3),105–10.
- Doğan B. , Erol B., Buldu A., 2014. Sigortacılık Sektöründe Müşteri İlişkileri Yönetimi İçin Birliktelik Kuralı Kullanılması, *Marmara Fen Bilimleri Dergisi* 26(3),105–14.
- Döşlü A., 2008. Veri Madenciliğinde Market Sepet Analizi ve Birliktelik Kurallarının Belirlenmesi, Yıldız Teknik Üniversitesi, İstanbul.
- Dumhan M. H., 2003. *Data Mining: Introductory and Advanced Topics*, New Jersey, Pearson.
- Dunham M.H., Xiao Y., Gruenwald L., Hossain Z., 2000. *A Survey of Association Rules.*, Texas.
- Durdu M., 2012. Application of Data Mining in Customer Relationship Management Market Basket Analysis in a Retailer Store, Dokuz Eylül University Graduate.
- Emel G. G., Taşkın Ç., 2002. Genetik Algoritmalar ve Uygulama Alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Dergisi* ,21(1),129–152.
- Erdem S., Özdağoğlu G., 2008. Ege Bölgesi'deki Bir Uygulama ve Araştırma Hastanesinin Acil Hasta Verilerinin Veri Madenciliği İle Analiz Edilmesi, *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi* ,9(7),261–270.
- Ergün E. 2008. Ürün Kategorileri Arasındaki Satış İlişkisinin Birliktelik Kuralları ve Kümeleme Analizi İle Belirlenmesi ve Perakende Sektöründe Bir Uygulama, Afyon Kocatepe Üniversitesi.
- Erpolat S. 2012. Otomobil Yetkili Servisinde Birliktelik Kurallarının Belirlenmesinde Apriori ve FP-Growth Algoritmalarının Karşılaştırılması, *Anadolu Üniversitesi Sosyal Bilimler Dergisi* ,12(2),137–46.

- Fayyad U., Piatetsky-Shapiro G., Smyth P., 1996. From Data Mining to Knowledge Discovery in Databases, *AI Magazine* ,17(3),37.
- Frawley W. J., Piatetsky-shapiro G., Matheus C. J., 1992. Knowledge Discovery in Databases : An Overview *AI Magazine* ,13(3),57–70.
- Gözcü M. K., 2015. Hastane Bilgi Yönetim Sistemlerinde İş Zekası Uygulaması Başkent Üniversitesi.
- Guidici P., 2004. *Applied Data Mining: Statistical Methods for Business and Industry*, (2nd ed.) Hoboken: John Wiley and Sons Ltd.
- Güngör E., Yalçın N., Yurtay N., 2008. Apriori Algoritması İle Teknik Seçmeli Ders Seçim Analizi, *Endüstri Mühendisliği Dergisi* ,21(2),14–29.
- Gürsakal N., 2001. *Sosyal Bilimlerde Araştırma Yöntemleri*, Bursa, VİPAŞ.
- Han J., Kamber M., 2006. *Data Mining: Concepts and Techniques*. (edited by 2nd.) USA, Morgan Kaufmann Publishers, Elsevier.
- Hanguang L.Yu N., 2012. Intrusion Detection Technology Research Based on Apriori Algorithm, Pp. 1615–1620 in *2012 International Conference on Applied Physics and Industrial Engineering*, (24), Nanjing, Elsevier.
- Hastie T., Tibshirani R., Friedman J. ,2004. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, (edited by Second.), USA, Springer.
- Hoche S., Geist I., Castillo L.P., Schulz N. , 2006. Databases, Knowledge Discovery, Information Retrieval and Web Mining, *Copyright American Society of Agricultural Engineers* ,(6),168–84.
- Imhoff C., Galemno N., Geiger J.G., 2003. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*.
- İnan, Osman. 2018. Genetik Algoritma, Erişim tarihi:20.12.2017. <https://www.slideshare.net/osmaninan42/genetik-algoritma>
- Jacobs P., 1999. Data Mining: What General Managers Need to Know, *Harvard Management Update* ,4(10),8–10.
- Kalogirou S., 2000. Applications of Artificial Neural-Networks for Energy Systems, *Applied Energy* ,(67),17–35.
- Kamley S., Jaloree S., Thakur R. S., 2014. An Association Rule Mining Model for Finding the Interesting Patterns in Stock Market Dataset, *International Journal of Computer Applications*, 93(9),975–8887.
- Kantardzic M., 2002. *Data Mining Concepts, Models, Methods, and Algorithms*, USA, WILEY INTERSCIENCE, IEEE.

- Kaur M., Kang S., 2016. Market Basket Analysis: Identify the Changing Trends of Market Data Using Association Rule Mining, Pp. 78–85 in *International Conference on Computational Modeling and Security (CMS 2016)*, (85) India, Procedia Computer Science, ELSEVIER.
- Kaya Y., Yeşilova A., 2011. İki Durumlu Karışımli Lojistik Regresyona İlişkin Bir Uygulama, *Bilişim Teknolojileri Dergisi*, 4(3),53–58.
- Kittler R., Wang W., 1999. The Emerging Role for Data Mining, *Solid State Technology*, 42,45–58.
- Kocabaş F. M., 2010. Veri Madenciliği Süreci ve Gerçek Bir Veri Seti Üzerinde Uygulanması, Hacettepe Üniversitesi.
- Koyuncugil A. S., Özgülbaş N., 2009. Veri Madenciliği :Tıp ve Sağlık Hizmetlerinde Kullanımı ve Uygulamaları, *Bilişim Teknolojileri Dergisi*, 2(2),21–32.
- Köktürk F., Ankaralı H., Sümbüloğlu V., 2009. Veri Madenciliği Yöntemlerine Genel Bakış, *Türkiye Klinikleri*, 1(1),20–25.
- Kuo, R. J., Lin S. Y., Shih C. W., 2007. Mining Association Rules through Integration of Clustering Analysis and Ant Colony System for Health Insurance Database in Taiwan, *Expert Systems with Applications*, 3(33),794–808.
- Li T., Li X., 2010. Novel Alarm Correlation Analysis System Based on Association Rules Mining in Telecommunication Networks, *Information Sciences*, 180(16),2960–2978.
- Mansur A., Kuncoro T., 2012. Product Inventory Predictions at Small Medium Enterprise Using Market Basket Analysis Approach-Neural Networks, *Procedia Economics and Finance*, (4),312–320.
- Nahar J., Imam T., Tickle K. S., Chen Y. P. P., 2013. Association Rule Mining to Detect Factors Which Contribute to Heart Disease in Males and Females, *Expert Systems with Applications*, 40(4),1086–1093.
- Nahar J., Tickle K. S., Ali A. B.M.S., Chen Y.P.P., 2011. Significant Cancer Prevention Factor Extraction: An Association Rule Discovery Approach, *Journal of Medical Systems*, 3(35),353–367.
- Oğuzlar A., 2004. *Veri Madenciliğine Giriş*, Bursa: Ekin Kitabevi.
- Öğüdücü Ş., 2018. Veri Madenciliği İlişkilendirme Kuralları. Erişim tarihi:15.03.2018
<http://web.itu.edu.tr/~sgunduz/courses/verimaden/slides/d0.pdf>

- Özçakır F.C., Çamurcu A.Y., 2007. Birliktelik Kuralı Yöntemi İçin Bir Veri Madenciliği Yazılımı Tasarımı ve Uygulaması, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 2(12),21–37.
- Özçalıcı M., 2017. Veri Madenciliğinde Birliktelik Kuralları ve İkinci El Otomobil Piyasası Üzerine Bir Uygulama, *Ordu Üniversitesi Sosyal Bilimler Araştırmaları Dergisi*, 7(1),45–58.
- Özekes S. , Çamurcu Y., 2003. Veri Madenciliğinde Karar Ağaçları Yöntemi Uygulaması, *Bilgi Teknolojileri Kongresi II*, Denizli, Pamukkale Üniversitesi.
- Özkan Y., 2016. *Veri Madenciliği Yöntemleri*, (3rd ed.) İstanbul, Papatya Bilim.
- Preeti P.J., Deshpande U., 2013. A Stock Market Portfolio Recommender System Based on Association Rule Mining, *Applied Soft Computing Journal* 13(2),1055–1063.
- Van Den Poel D., De Schamphelaere J., Wets G., 2004. Direct and Indirect Effects of Retail Promotions on Sales and Profits in the Do-It-Yourself Market, *Expert Systems with Applications*, 1(27),53–62.
- Rao, B. S. S., Shield B., 1998. Diaper-Beer Syndrome, *Forbes LLC*, 161(7),128–130.
- Savaş S., Topaloğlu N., Yılmaz M., 2012. Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 21,1–23.
- Silahtaroglu, G., 2016. *Veri Madenciliği: Kavram ve Algoritmaları*. (3rd ed.) İstanbul, Papatya Bilim.
- Solnet D., Boztug Y., Dolnicar S., 2016. An Untapped Gold Mine? Exploring the Potential of Market Basket Analysis to Grow Hotel Revenue, *International Journal of Hospitality Management*, 56,119–125.
- Srikant R., Agrawal R., 1996. Mining Quantitative Association Rules in Large Relational Tables, Pp. 1–12 in *ACM SIGMOD'96 International Conference of Management of Data*, Montreal.
- Şeker Ş. E., 2013. *İş Zekası ve Veri Madenciliği*, İstanbul, Cinius Yayınları.
- Şekeroğlu S., 2010. Hizmet Sektöründe Bir Veri Madenciliği Uygulaması, İstanbul Teknik Üniversitesi.
- Şentürk A. ,2006. *Veri Madenciliği Kavram ve Teknikleri*, Ekin Yayınevi.
- Tan P. N., Steinbach M., Kumar V., 2005. *Introduction To Data Mining*, USA, Pearson Addison Wesley.

- Timor M., Şimşek U.T., 2008. Veri Madenciliğinde Sepet Analizi İle Tüketici Davranışı Modellemesi, *Yönetim* ,59,3-10.
- Yang T.C., Lai H., 2006. Comparison of Product Bundling Strategies on Different Online Shopping Behaviors, *Electronic Commerce Research and Applications* 5,295-304.
- Yarımağan Ü. 2010. *Veri Tabanı Sistemleri*, Ankara, Akademi Yayıncılık.
- Yazgan H.R., Yener F., Cömert S.E., Kır S., Kaya Y., 2016. Birliktelik Kuralları ve Genetik Algoritma İle Sipariş Yığınlama Probleminin Çözümü: Bir Ecza Deposunda Uygulama, *Journal of Transportation and Logistics* ,1(1),129-142.
- Yıldırım P., Birant D., Alpyıldız T., 2017. "Discovering the Relationships between Yarn and Fabric Properties Using Association Rule Mining, *Turkish Journal of Electrical Engineering & Computer Sciences* ,25,4788-4804.
- Yun, C. H., Chuang K.T., Chen M.S., 2006. Adherence Clustering: An Efficient Method for Mining Market-Basket Clusters, *Information Systems* ,31(3),170-186.
- Zhang W., Liao H., Zhao N., 2008. Research on the FP Growth Algorithm about Association Rule Mining, Pp. 315-318 in *2008 International Seminar on Business and Information Management, ISBIM 2008*, 1, Kunming: IEEE.

EKLER

EK A. Çizelgeler

Çizelge A.1. Ana Kategoriler ve Kategori Kodları

Ana Kategori Kodu	Kategori Adı
1	DÜBELLER
2	ELEKTRİKLİ ALETLER
3	YAPIŞTIRICILAR VE KİMYASALLAR
4	TEKNİK HIRDAVAT
5	İŞ GÜVENLİĞİ
6	KALDIRMA EKİPMANLARI
7	MEKANİK EL ALETLERİ
8	MOBİLYA GRUBU
9	NALBURİYE VE HIRDAVAT
10	OTO AKSESUARLARI
11	TESTERELER
12	VİDALAR
13	ZIMPARALAR

Çizelge A.2. Alt Kategoriler ve Kategori Kodları

Alt Kategori Kodu	Alt Kategori Adı
1A	KANCALI DÜBEL
1B	BORULU DÜBEL
1C	ÇAKMALI DÜBEL
1D	GÖMLEKLİ DÜBEL
1E	KLİPSLİ DÜBEL
2A	ELEKTRİKLİ ALET YEDEK PARÇALARI
2B	BAHÇE ALETLERİ
2C	EL ALETLERİ
3A	AYAKKABI
3B	ELDİVENLER
3C	YAĞMURLUKLAR
4A	ŞİŞE KRİKOLAR
4B	ARABA KRİKOLARI
4C	CERASKAL

5A	MEKANİK EL ALETLERİ
6A	KAPI KOLU
6B	KAPI YAYLARI
6C	BAREL

Çizelge A.2 devamı

6D	ÇEKMECE CAM KİLİTLERİ
6E	KAPI DÜRBÜNLERİ
6F	MENTEŞELER
6G	TELESKOPIK RAYLAR
6H	FREZE BIÇAKLARI
6I	AHŞAP JALUZİ
7A	MAKARA
7B	RONDELA
7C	SPATULA
7D	SU TERAZİLERİ
7E	BANTLAR
7F	ÇEKİÇLER
7G	BOYA KARIŞTIRICILARI
7H	CIRTLI ZIMPARA TABANLARI
7I	KERPETEN, PENSE, YANKEKSİ, KARGABURUN
7J	MAKET BIÇAĞI
7K	ASMA KİLİTLER
7L	SPIRAL HAVA HORTUMLARI
7M	BALTA
7N	KESER
7O	METRELER
7P	SIVA FİLESİ
7R	TABANCALAR
7S	TOKMAK
7T	SİNEK TELİ
7U	ÇEKMECE CAM KİLİTLERİ
7V	PLANYA BIÇAĞI
8A	AKSESUARLAR
8B	OTO YEDEK PARÇALARI
9A	ENJEKSİYON KALIP ELEMANLARI
9B	YEDEK PARÇALAR
9C	MURÇ VE KESKİLER
9D	MANDRENLER
9E	ANAHTARLAR
9F	BITS UÇLAR ADAPTÖTLER
9G	PANÇ GRUBU
9H	EGELER
9I	MANYETİK BITS ADAPTÖRÜ
9J	TEL FIRÇALAR
9K	SOMUN ADAPTÖRLERİ
9L	KAPLİN VE SOKETLER
9M	MATKAP UÇLARI
9N	TORNAVİDALAR
9O	MASKE CAMLARI

9P	SDS PLUS ADAPTÖTLERİ
9R	ELEKTRİKLİ ÜRÜNLER
9S	RULMANLAR
9T	PİM

Çizelge A.2 devamı

9V	YALITIM MALZEMELELERİ
9Y	KOMPRATÖRLER
10A	TESTERE AĞZI
10B	ELMAS TESTERE
10C	DEMİR TESTERE KOLU
10D	DAİRE TESTERELER
10E	TESTERE BIÇAĞI
10F	MDF TESTERE
11A	ÖZEL VİDALAR
11B	BULDESK VİDALAR
11C	ALÇIPAN VİDALAR
11D	SUNTALEM VİDALAR
11E	MATKAP UÇLU VİDALAR
11F	ÇATI TRAPEZ VİDALARI
12A	TİNER
12B	HIZLI YAPIŞTIRICILAR
12C	ÇIVATA SABİTLEYİCİLER
12D	KİT HIZLI YAPIŞTIRICILAR
12E	MUM ÇUBUKLARI
12F	MDF KİT
13A	BANT ZİMPARALARI
13B	KESİCİ DİSK
13C	SU ZİMPARASI
13D	FLAP DİSKLER
13E	CIRTLI ZİMPARALAR
13F	RULO ZİMPARALAR
13G	ZİMPARA YEDEK PARÇALARI
13H	DİĞER ZİMPARALAR

Çizelge A.3. Veri Seti-1 için WEKA programı Apriori Algoritması Sonuçları

1. 9C=t 9F=t 677 ==> 9M=t 431	<conf:(0.64)> lift:(3) lev:(0.01) [287] conv:(2.16)
2. 9C=t 9G=t 366 ==> 9M=t 233	<conf:(0.64)> lift:(3) lev:(0) [155] conv:(2.15)
3. 9C=t 9J=t 454 ==> 9M=t 271	<conf:(0.6)> lift:(2.81) lev:(0) [174] conv:(1.94)
4. 9C=t 2370 ==> 9M=t 1361	<conf:(0.57)> lift:(2.7) lev:(0.02) [857] conv:(1.85)
5. 7O=t 9C=t 411 ==> 9M=t 235	<conf:(0.57)> lift:(2.69) lev:(0) [147] conv:(1.83)
6. 10B=t 9C=t 563 ==> 9M=t 316	<conf:(0.56)> lift:(2.64) lev:(0) [196] conv:(1.79)
7. 7O=t 9K=t 481 ==> 9F=t 251	<conf:(0.52)> lift:(3.33) lev:(0) [175] conv:(1.76)
8. 9K=t 9M=t 962 ==> 9F=t 497	<conf:(0.52)> lift:(3.29) lev:(0.01) [346] conv:(1.74)
9. 9J=t 9K=t 618 ==> 9F=t 305	<conf:(0.49)> lift:(3.15) lev:(0) [208] conv:(1.66)
10. 9F=t 9H=t 505 ==> 9M=t 242	<conf:(0.48)> lift:(2.26) lev:(0) [134] conv:(1.51)
11. 9K=t 2730 ==> 9F=t 1308	<conf:(0.48)> lift:(3.05) lev:(0.02) [879] conv:(1.62)
12. 9P=t 672 ==> 9F=t 312	<conf:(0.46)> lift:(2.96) lev:(0) [206] conv:(1.57)
13. 9F=t 9N=t 673 ==> 9M=t 308	<conf:(0.46)> lift:(2.15) lev:(0) [164] conv:(1.45)
14. 9D=t 9F=t 734 ==> 9M=t 335	<conf:(0.46)> lift:(2.15) lev:(0) [179] conv:(1.45)
15. 9L=t 721 ==> 7L=t 329	<conf:(0.46)> lift:(15.39) lev:(0.01) [307] conv:(1.78)
16. 10B=t 9D=t 594 ==> 9M=t 266	<conf:(0.45)> lift:(2.11) lev:(0) [139] conv:(1.42)
17. 10B=t 9K=t 556 ==> 9F=t 247	<conf:(0.44)> lift:(2.83) lev:(0) [159] conv:(1.51)
18. 9F=t 9G=t 1015 ==> 9M=t 444	<conf:(0.44)> lift:(2.06) lev:(0.01) [228] conv:(1.4)
19. 7R=t 9F=t 785 ==> 9M=t 339	<conf:(0.43)> lift:(2.03) lev:(0) [172] conv:(1.38)
20. 6D=t 9F=t 851 ==> 9M=t 357	<conf:(0.42)> lift:(1.97) lev:(0) [176] conv:(1.35)
21. 10B=t 9F=t 1248 ==> 9M=t 521	<conf:(0.42)> lift:(1.96) lev:(0.01) [255] conv:(1.35)
22. 4B=t 1233 ==> 4A=t 509	<conf:(0.41)> lift:(4.64) lev:(0.01) [399] conv:(1.55)
23. 9P=t 672 ==> 9M=t 272	<conf:(0.4)> lift:(1.9) lev:(0) [129] conv:(1.32)
24. 10B=t 9K=t 556 ==> 9M=t 225	<conf:(0.4)> lift:(1.9) lev:(0) [106] conv:(1.32)

Çizelge A.4. Veri Seti-1 için WEKA programı FP-Growth Algoritması Sonuçları

1. [9L=t]: 721 ==> [7L=t]: 329	<conf:(0.46)> lift:(15.39) lev:(0.01) conv:(1.78)
2. [4B=t]: 1233 ==> [4A=t]: 509	<conf:(0.41)> lift:(4.64) lev:(0.01) conv:(1.55)

Çizelge A.5. Veri Seti-2 İçin WEKA programı Apriori Algoritması Sonuçları

1. 10B=t 9C=t 104 ==> 9M=t 71	<conf:(0.68)> lift:(4.85) lev:(0) [56] conv:(2.63)
2. 9C=t 9G=t 117 ==> 9M=t 76	<conf:(0.65)> lift:(4.61) lev:(0) [59] conv:(2.39)
3. 9C=t 9F=t 155 ==> 9M=t 99	<conf:(0.64)> lift:(4.54) lev:(0.01) [77] conv:(2.34)
4. 9C=t 540 ==> 9M=t 314	<conf:(0.58)> lift:(4.13) lev:(0.02) [237] conv:(2.04)
5. 9G=t 9K=t 142 ==> 9F=t 80	<conf:(0.56)> lift:(4.56) lev:(0.01) [62] conv:(1.98)
6. 9K=t 9M=t 206 ==> 9F=t 111	<conf:(0.54)> lift:(4.36) lev:(0.01) [85] conv:(1.88)
7. 9P=t 167 ==> 9F=t 89	<conf:(0.53)> lift:(4.31) lev:(0.01) [68] conv:(1.85)
8. 9J=t 9K=t 146 ==> 9F=t 77	<conf:(0.53)> lift:(4.27) lev:(0) [58] conv:(1.83)
9. 10B=t 9K=t 128 ==> 9F=t 66	<conf:(0.52)> lift:(4.17) lev:(0) [50] conv:(1.78)
10. 9K=t 637 ==> 9F=t 313	<conf:(0.49)> lift:(3.98) lev:(0.02) [234] conv:(1.72)
11. 9F=t 9N=t 141 ==> 9M=t 69	<conf:(0.49)> lift:(3.48) lev:(0) [49] conv:(1.66)
12. 6D=t 9F=t 226 ==> 9M=t 107	<conf:(0.47)> lift:(3.36) lev:(0.01) [75] conv:(1.62)
13. 6D=t 9M=t 240 ==> 9F=t 107	<conf:(0.45)> lift:(3.61) lev:(0.01) [77] conv:(1.57)
14. 7O=t 9F=t 221 ==> 9M=t 98	<conf:(0.44)> lift:(3.15) lev:(0.01) [66] conv:(1.53)
15. 10B=t 9F=t 254 ==> 9M=t 110	<conf:(0.43)> lift:(3.08) lev:(0.01) [74] conv:(1.51)
16. 6H=t 9F=t 148 ==> 9M=t 63	<conf:(0.43)> lift:(3.02) lev:(0) [42] conv:(1.48)
17. 9F=t 9G=t 304 ==> 9M=t 128	<conf:(0.42)> lift:(2.99) lev:(0.01) [85] conv:(1.48)
18. 9M=t 9N=t 170 ==> 9F=t 69	<conf:(0.41)> lift:(3.28) lev:(0) [47] conv:(1.46)
19. 9G=t 9J=t 194 ==> 9F=t 78	<conf:(0.4)> lift:(3.25) lev:(0) [54] conv:(1.45)

Çizelge A.6. Veri Seti-2 için WEKA programı FP-Growth Algoritması Sonuçları

1. [10B=t, 9C=t]: 104 ==> [9M=t]: 71	<conf:(0.68)> lift:(4.85) lev:(0) conv:(2.63)
2. [9G=t, 9C=t]: 117 ==> [9M=t]: 76	<conf:(0.65)> lift:(4.61) lev:(0) conv:(2.39)
3. [9F=t, 9C=t]: 155 ==> [9M=t]: 99	<conf:(0.64)> lift:(4.54) lev:(0.01) conv:(2.34)
4. [9C=t]: 540 ==> [9M=t]: 314	<conf:(0.58)> lift:(4.13) lev:(0.02) conv:(2.04)
5. [9G=t, 9K=t]: 142 ==> [9F=t]: 80	<conf:(0.56)> lift:(4.56) lev:(0.01) conv:(1.98)
6. [9M=t, 9K=t]: 206 ==> [9F=t]: 111	<conf:(0.54)> lift:(4.36) lev:(0.01) conv:(1.88)
7. [9P=t]: 167 ==> [9F=t]: 89	<conf:(0.53)> lift:(4.31) lev:(0.01) conv:(1.85)
8. [9J=t, 9K=t]: 146 ==> [9F=t]: 77	<conf:(0.53)> lift:(4.27) lev:(0) conv:(1.83)
9. [10B=t, 9K=t]: 128 ==> [9F=t]: 66	<conf:(0.52)> lift:(4.17) lev:(0) conv:(1.78)
10. [9K=t]: 637 ==> [9F=t]: 313	<conf:(0.49)> lift:(3.98) lev:(0.02) conv:(1.72)

ÖZGEÇMİŞ

Adı Soyadı : Ayşe Nur SAĞIN
Doğum Yeri ve Yılı : İstanbul, 21/02/1994
Medeni Hali : Bekar
Yabancı Dili : İngilizce
E-posta : anur.sagin@istanbulticaret.edu.tr



Eğitim Durumu

Lise : Özel Gaziosmanpaşa Şefkat Koleji, 2012
Lisans : Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Edebiyat
Fakültesi, Matematik Bölümü
Anadolu Üniversitesi, İşletme Fakültesi, İşletme Bölümü

Yayınları

Sağın, A. N., Ayvaz, B.,2018. Determination of Association Rules with Market Basket Analysis: An Application in the Retail Sector, Southeast Europe Journal of Soft Computing, (7)1,10-19.