

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**VERİ MADENCİLİĞİ VE APRIORİ ALGORİTMASI
İLE SÜPERMARKET ANALİZİ**

YÜKSEK LİSANS TEZİ

End. Müh. Derya AY

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Doç. Dr. İbrahim ÇİL

Şubat 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

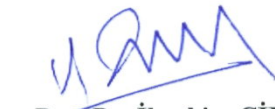
VERİ MADENCİLİĞİ VE APRIORİ ALGORİTMASI
İLE SÜPERMARKET ANALİZİ


YÜKSEK LİSANS TEZİ


End. Müh. Derya AY

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 04 / 02 /2009 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.


Doç. Dr. İbrahim ÇİL
Jüri Başkanı


Yrd. Doç. Dr. Tarık ÇAKAR
Üye


Doç. Dr. Nejat YUMUŞAK
Üye

TEŐEKKÜR

Tez alıőmamda deęerli fikir ve önerileriyle beni yönlendiren danıőman hocam Sayın Do. Dr. İbrahim il' e, alıőmamda uygulama bölümü için gerçek veri saęlayan Migros Türk A.Ő. yetkililerine, yüksek lisans eęitimim süresince burs desteęi saęlayan TÜBİTAK' a ve alıőmam boyunca yardım ve desteęini esirgemeyen niőanlım End. Müh. Emre Yazıcı ve aileme teőekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ	vii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xi
ÖZET.....	xii
SUMMARY	xiii

BÖLÜM 1.

GİRİŞ	1
-------------	---

BÖLÜM 2.

VERİ MADENCİLİĞİ.....	3
2.1. Veritabanlarında Bilgi Keşfi Süreci.....	3
2.2. Veri Madenciliği Tanımı.....	5
2.3. Veri Madenciliğinin Diğer Disiplinlerle Olan İlişkisi	6
2.4. Veri Madenciliğinin Uygulandığı Depolama Birimleri	8
2.4.1. Veri ambarları	9
2.4.2. İlişkisel veri tabanları.....	10
2.4.3. İşlemsel veri tabanları	11
2.4.4. Gelişmiş veri tabanı sistemleri ve uygulamaları	11
2.5. Veri Madenciliği Süreci: CRISP-DM	12
2.6. Veri Madenciliği Uygulama Alanları	13
2.7. Veri Madenciliği Modelleri	15
2.7.1. Sınıflama ve regresyon.....	16
2.7.2. Kümeleme	17

2.7.3. Birliktelik kuralları ve ardışık zamanlı örüntüler.....	18
2.8. Veri Madenciliği Modelleri	18
2.8.1. Karar ağaçları	19
2.8.2. Yapay sinir ağları	19
2.8.3. Genetik algoritmalar	20
2.8.4. K-en yakın komşu	20
2.8.5. Bellek tabanlı yöntemler	20
2.8.6. Naive bayes	21
2.8.7. Regresyon.....	22
2.8.8. K ortalamalar algoritması.....	22
2.9. Veri Madenciliğinde Karşılaşılan Problemler.....	23
2.10. Veri Madenciliği Yazılımları	25
2.10.1. Clementine	25
2.10.2. Enterprise miner	25
2.10.3. Insightful miner.....	26
2.10.4. Darwin.....	26
2.10.5. Statistica data miner	26
BÖLÜM 3.	
BİRLİKTELİK KURALLARI.....	28
3.1. Birliktelik Kuralları Tanımı	28
3.2. Birliktelik Kuralları Matematiksel Modeli Ve Temel Kavramları	29
3.2.1. Birliktelik kuralları matematiksel modeli	29
3.2.2. Destek (support) ve güven (confidence) değeri	30
3.2.3. K-nesneküme (k-itemset).....	31
3.2.4. Sık nesneküme (Frequent itemset).....	31
3.2.5. Minimum destek ve güven değeri.....	31
3.2.6. Güçlü birliktelik kurallar.....	32
3.3. Birliktelik Kuralı Madenciliği.....	32
3.3.1. Sık geçen nesneküme bulunuşu	32
3.3.2. Güçlü birliktelik kurallarının oluşturulması.....	33

3.4. Sık Geçen Nesnekümlerin Temsilleri	34
3.4.1. Maksimal sık nesnekümler	34
3.4.2. Kapalı sık nesnekümler	35
3.5. Sık Geçen Nesnekümleri Oluşturma Metotları Alternatifleri	38
3.5.1. Genişlik ve derinlik öncelikli algoritmalar	39
3.5.2. Genelden-özele ve özelden-genele yaklaşımlı algoritmalar	40
3.6. Sık Geçen Nesnekümler Madenciliği.....	42
3.6.1. Apriori algoritması	42
3.6.1.1. Apriori özelliği.....	43
3.6.1.2. Apriori işleyişi	44
3.6.2. AprioriTid algoritması	49
3.6.3. AprioriHybrid algoritması.....	50
3.6.4. AIS algoritması	52
3.6.5. SETM algoritması	52
3.6.6. FP-Growth algoritması.....	52
3.7. Birliktelik Kuralları Oluşturma.....	56
3.8. Birliktelik Kuralları Madenciliği Çeşitleri	57
3.8.1. Çok seviyeli birliktelik kuralları madenciliği	57
3.8.2. Çok boyutlu birliktelik kuralları madenciliği.....	60
3.8.3. Nicel birliktelik kuralları madenciliği.....	60

BÖLÜM 4.

MAĞAZA YERLEŞİM DÜZENİ	63
4.1. Mağaza İçi Yerleşiminin Önemi	63
4.2. Temel Alan Düzenlemeleri	64
4.2.1. Izgara (Grid) biçimi	64
4.2.2. Yarış alanı biçimi	66
4.2.3. Serbest yerleşim düzeni	66
4.3. Ürünlerin Gruplandırılması.....	66

BÖLÜM 5.	
ÇOK BOYUTLU ÖLÇEKLEME ANALİZİ.....	69
5.1. Çok Boyutlu Ölçekleme Analizi Tanımı	69
5.2. Çok Boyutlu Ölçekleme Analizinde Kullanılan Yöntem	70
BÖLÜM 6.	
UYGULAMA	74
6.1. İş Analizi	74
6.2. Verinin Anlaşılması Ve Modellemeye Hazırlanması	74
6.2.1. Veri tabanı oluşturma.....	76
6.2.2. Clementine programında verinin hazırlanması	81
6.3. Modelleme	87
6.4. Çok Boyutlu Ölçekleme Analizi İle Ürün Haritasının Çıkarılması	91
6.5. Değerlendirme Ve Modelinin Geçerliliği	93
BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER	96
KAYNAKLAR	98
EKLER	104
ÖZGEÇMİŞ	123

SİMGELER VE KISALTMALAR LİSTESİ

BFS	: Genişlik öncelikli arama (Breadth first search)
c	: Güven değeri (confidence)
C	: Kapalı sık nesneküme (Closed frequent itemset)
C_k	: k öğeli aday nesneküme
\acute{C}_k	: k öğeli ve TID içeren sık nesneküme kümesi
CRISP-DM	: Cross Industry Standart Process for Data Mining
ÇBÖ	: Çok Boyutlu Ölçekleme
D	: Veri tabanına ait işlemler kümesi
DFS	: Derinlik öncelikli arama (Depth first search)
d_{ij}	: i. ve j. birimler arasındaki öklid uzaklıkları
I	: Veri tabanındaki nesnelere
KDS	: Karar destek sistemleri
k-nesneküme	: k öğeli nesneküme
L	: Sık nesneküme
L_a	: L' nin boş olmayan alt kümeleri
L_k	: Sık geçen k öğeli nesneküme
m	: öklid uzayı boyut sayısı
MDS	: Multidimensional Scaling
min.	: minimum
n	: nokta/nesne sayısı
no	: Numara
OLAP	: Online Analytical Processing
p_{ij}	: i. ve j. birimler arasındaki gözlenen uzaklıklar
R^2	: Kareli korelasyon katsayısı
s	: Destek değeri (support)
S	: Gerginlik (Stress) ölçüsü

SQL	: Yapılandırılmış sorgu dili (Structured query language)
T	: İşlemler kümesi
TID	: İşlem numarası
VM	: Veri madenciliği
VTBK	: Veri tabanlarında bilgi keşfi
$X \Rightarrow Y$: Birliktelik kuralı
X_i	: Veri tabanındaki i. nesne
Y_i	: Veri tabanındaki j. nesne
YSA	: Yapay sinir ağları

ŞEKİLLER LİSTESİ

Şekil 2.1.	Veri tabanlarında bilgi keşfi sürecinde veri madenciliğinin yeri ..	4
Şekil 2.2.	Veri madenciliğini disiplinler arası bir alandır.....	6
Şekil 2.3.	CRISP-DM metodolojisi	12
Şekil 3.1.	Maksimal sık nesnekümler	35
Şekil 3.2.	Kapalı sık nesnekümler.....	36
Şekil 3.3.	Sık geçen nesnekümlerin temsilleri arasındaki ilişki.....	38
Şekil 3.4.	BFS yaklaşımı	39
Şekil 3.5.	DFS yaklaşımı	39
Şekil 3.6.	DFS yaklaşımını kullanarak nesneküme adayları oluşturma	40
Şekil 3.7.a.	Genelden-özele yaklaşımı	41
Şekil 3.7.b.	Özelden-genele yaklaşımı	41
Şekil 3.8.	İki yönlü arama.....	42
Şekil 3.9.	Apriori algoritmasının birleştirme özelliği.....	43
Şekil 3.10.	Apriori algoritmasının budama özelliği.....	44
Şekil 3.11.	Apriori algoritmasının adımları	49
Şekil 3.12.	AprioriTid algoritması örneği.....	51
Şekil 3.13.	Bir FP-ağacı.....	54
Şekil 3.14.	Şartlı I3 düğümü ile ilişkilendirilmiş şartlı FP-ağacı.....	55
Şekil 3.15.	Örnek veri setine ait ürün hiyerarşisi.....	58
Şekil 3.16.	Tekdüze destek değeri ile çoklu seviyeli madenleme	59
Şekil 3.17.	İndirilmiş destek değeri ile çoklu seviyeli madenleme	60
Şekil 4.1.	Izgara biçimi yerleşim	65
Şekil 5.1.	İki boyutlu düzlemde örnek grafiksel gösterim.....	70
Şekil 5.2.	Dağılım grafiği	72
Şekil 6.1.	Çalışmanın genel kapsamı	75
Şekil 6.2.	Alışveriş kayıtları	76

Şekil 6.3.	Ürün verileri	76
Şekil 6.4.	Ürün hiyerarşisi	77
Şekil 6.5.	SQL veritabanına aktarılmış alışveriş kayıtları tablosu.....	78
Şekil 6.6.	SQL veritabanına aktarılmış ürün verileri tablosu	79
Şekil 6.7.	Müşteri numarası tarih, kasa_no ve fiş_no' dan oluşmaktadır	79
Şekil 6.8.	Ana ürün kategorilerine göre ürün verileri	80
Şekil 6.9.	Özet tablo.....	81
Şekil 6.10.	SQL Veritabanının veri kaynağı yöneticisine eklenmesi	82
Şekil 6.11.	Clementine' da database düğümü ile migros_veritabanına yapılan erişim	82
Şekil 6.12.	Clementine' da database düğümüne table düğümünün eklenmesi	83
Şekil 6.13.	Verilerin Clementine' da görüntülenmesi	83
Şekil 6.14.	Clementine modelleme alanın görüntüsü	85
Şekil 6.15.	Veri kalitesini gösteren (Quality) düğüm çıktısı	85
Şekil 6.16.	Ürün kategorilerin dağılımı	86
Şekil 6.17.	Tanımlayıcı istatistikler	86
Şekil 6.18.	Birliktelik kuralları	88
Şekil 6.19.	Clementine' da kurulan modelin görüntüsü	88
Şekil 6.20.	Tüm ürünler arasındaki birliktelik diyagramı (çizgi değeri \geq %0)	90
Şekil 6.21.	İlişki seviyesi %18 üzerinde olan ürünlerin birliktelik diyagramı	91
Şekil 6.22.	ÇBÖ Analizi ile elde edilen ürün haritası.....	92

TABLolar LİSTESİ

Tablo 2.1.	Veri ambarı ve veri tabanı arasındaki farklar	10
Tablo 2.2.	Satışlar tablosu.....	11
Tablo 3.1.	Birliktelik kurallarının genel gösterimi	30
Tablo 3.2.	Örnek veri tabanı parçası.....	42
Tablo 3.3.	Notasyonlar.....	45
Tablo 3.4.	Apriori algoritmasının kodları	46
Tablo 3.5.	İşlemsel veriler	47
Tablo 3.6.	Örnek veri tabanı parçası (minimum destek değeri=2)	53
Tablo 3.7.	Şartlı örüntü temelleri oluşturarak FP-ağaç madenciliği	55
Tablo 3.8.	FP-Growth algoritması kodları.....	56
Tablo 3.9.	Örnek veri seti	58
Tablo 3.10.	Nicel ve kategorik özelliklerden oluşan örnek veri seti	61
Tablo 3.11.	Nicel birliktelik kuralları örneği (min.s=%40, min.c=%50)	61
Tablo 3.12.	Boolean birliktelik kuralı şeklinde inceleme	62
Tablo 6.1.a.	Tanımlayıcı İstatistikler.....	87
Tablo 6.1.b.	En sık satan 6 ürün	87
Tablo 6.2.	Birliktelik Kuralları	89
Tablo 6.3.	En güçlü birliktelikler	90

VERİ MADENCİLİĞİ VE APRIORİ ALGORİTMASI İLE SÜPERMARKET ANALİZİ

Derya AY

ÖZET

Anahtar Kelimeler: Veri Madenciliği, Birliktelik Kuralları, Market Sepet Analizi, Apriori Algoritması, Yerleşim Düzeni

Günümüzde kuruluşların teknoloji yatırımlarının artması ile çok büyük miktarlardaki veriler toplanıp saklanabilmektedir. Toplanan bu verilerin analizi, yorumlanması ve karar vermede nasıl kullanılabilir hale getirileceği önemli bir konu haline gelmektedir. Teknoloji yatırımlarının artması ile birlikte veri tabanlarında saklı olan bilgilere otomatik bilgi keşfetme teknikleriyle ulaşılmaya çalışılması veri madenciliği kullanımının geniş bir alana yayılmasını sağlamaktadır. Bu bağlamda veri madenciliğinin en çok kullanılan tekniklerinden biri de birliktelik kurallarıdır. Birliktelik kuralları nesnelere arasındaki ilişkileri bulmayı amaçlar ve kullanıldığı en yaygın alanlardan biri ise süpermarket uygulamalarıdır.

Bu çalışmada veri madenciliği hakkında temel bilgiler verilerek, birliktelik kuralları ve birliktelik kurallarının en temel algoritmalarından biri olan Apriori algoritması detaylı olarak incelenmiştir. Bir veri madenciliği programı aracılığıyla, Apriori algoritması kullanılarak Migros Türk A.Ş. verileri ile market sepet analizi yapılmıştır. Yapılan bu çalışma sonunda birlikte satılma eğilimi gösteren ürünler hakkında bilgiler verilerek, yeni bir market yerleşim düzeni önerilmiştir.

DATA MINING AND SUPERMARKET ANALYSIS WITH APRIORI ALGORITHM

Derya AY

SUMMARY

Key Words: Data Mining, Association Rules, Market Basket Analysis, Apriori Algorithm, Store Layout

Today, large amounts of data can be collected and stored with increasing technologic investments of the organizations. Analyzing, expounding and how to use decision making of these collected data are getting important issue. Usage of data mining became prevalent because of using automatic information discovery methods to find information that is hidden in databases with increasing technologic investments. In this context, association rule is one of the most common technique in data mining. Association rules purpose finding relationships among of items and association rule is mostly used for supermarket applications.

This study gives basic information about data mining, association rules and Apriori that is one of the main algorithms of association rule mining are examined in details. Market basket analysis has done with Migros Türk A.Ş. data's by using Apriori algorithm with the data mining software. As a result, this study gives information about products which sales together and then a new market store layout is proposed.

BÖLÜM 1. GİRİŞ

Günümüz teknolojisi hızla ilerlemekte ve bilgisayar sistemlerinin güçleri her geçen gün artmaktadır. Bununla birlikte işletmelerde üretilen ve depolanan veriler de çok büyük hacimlere ulaşmaktadır. Gelişen veri tabanı teknolojisi ve hacimlerindeki bu olağanüstü artış, karar vericilerin toplanan bu verilerden nasıl faydalanacağını ve bu verileri nasıl bilgiye dönüştüreceği sorununu ortaya çıkarmaktadır. Karar vericilerin kendi kanaatleri yerine kanıta dayalı karar vermelerinin önemi giderek artmaktadır. Veri tabanlarındaki bu veriler üzerinde analiz yapmak ve karar destek aşamasında faydalanmak herhangi bir araç kullanmaksızın imkansız hale gelmiştir. Bu noktada çözüm olarak veri madenciliği kavramı ortaya çıkmaktadır.

Veri madenciliği önemli, geçerli, yeni ve kullanışlı bilgiyi büyük veri tabanlarından çıkarma işlemidir. Bu yüzden veri madenciliği, veri tabanından anlamlı örüntüler veya kurallar elde etmek için geniş bir araştırma alanı olarak görülmektedir. Veri madenciliği, veri tabanlarında bilgi keşfi sürecinde anlamlı örüntüleri elde eden keşif algoritmaları ile veri analizini uygulayan bir adımdır. Veri madenciliği, veri tabanlarındaki bilgi keşfi uygulamaları ile birlikte faaliyet alanına yönelik karar destek mekanizmaları için gerekli ön bilgileri temin etmek için kullanılmaktadır [1, 2, 3].

Veri madenciliği ve karar destek sistemleri birbirlerini birçok yönden tamamlamaktadır. Veri madenciliği bir problemi çözmek için bilgiyi veriden seçip çıkartırken, karar destek ise uzmanın elde ettiği bilgileri dikkate alma eğilimindedir. Veri madenciliği ve karar destek entegrasyonu problem çözme metodlarını, süreçlerini ve elde edilen sonuçların kalitesini arttırabilmektedir. Veri madenciliği ve karar destek entegrasyonunun birçok çeşiti bulunmaktadır [4, 5].

Bu çalışmada sadece veri madenciliğinin sonuçları gösterilmekle kalmayıp, elde edilen bu sonuçlar ilgili faaliyet alanında karar verme sürecinde kullanılmıştır. Böylelikle bu çalışmada veri madenciliği karar destek entegrasyonu modellerinden önce veri madenciliği sonra karar destek modeli kullanılmıştır [4, 6].

Çalışmanın ikinci bölümünde veri tabanlarında bilgi keşfi süreci, veri madenciliği tanımı, veri madenciliğinin diğer disiplinlerle olan ilişkisi, CRISP-DM, veri madenciliği uygulama alanları, VM modelleri ve VM teknikleri incelenmiştir. Üçüncü bölümde, birliktelik kuralları, Apriori algoritması, birliktelik kuralları madenciliği çeşitleri detaylı olarak incelenmiştir. Dördüncü bölümde market içi yerleşim düzenin önemi ve beşinci bölümde çok boyutlu karar verme metotlarından çok boyutlu ölçekleme analizi anlatılmıştır.

Uygulama bölümünde ise SPSS Clementine programı kullanılarak birliktelik kurallarının en temel algoritmalarından olan Apriori algoritması ile süpermarket analizi yapılmıştır. Birliktelik kuralları madenciliği sonuçlarına göre birlikte satılma eğilimindeki ürünler hakkında bilgiler verilmiştir. Ayrıca elde edilen sonuçlar çok boyutlu karar verme metotlarından çok boyutlu ölçekleme analizi ile değerlendirilerek iki boyutlu düzlemde ürün haritası elde edilmiştir. Sonuç olarak; birliktelik kuralları ve çok boyutlu ölçekleme analizleri sonuçlarına göre yeni bir market yerleşim düzeni önerilmiştir.

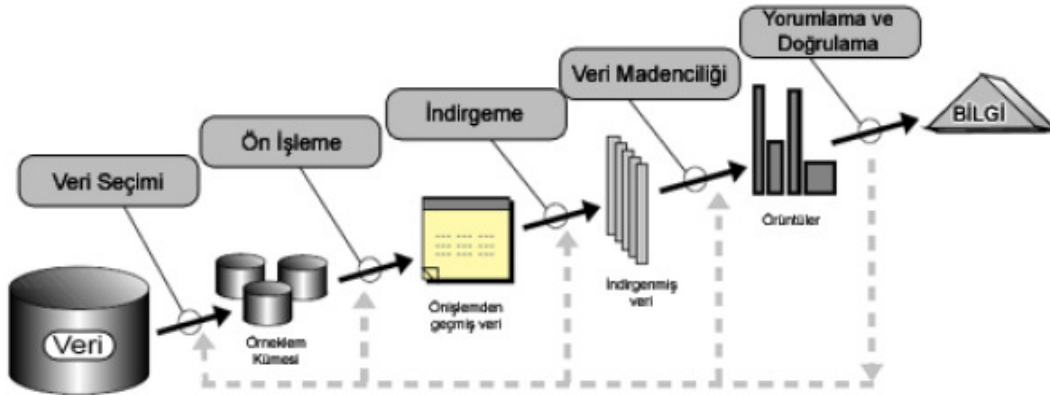
BÖLÜM 2. VERİ MADENCİLİĞİ

2.1. Veri Tabanlarında Bilgi Keşfi Süreci

Günümüzde yapılan her alışverişte, her bankacılık işleminde, her telefon edişte kaydedilen, uzaktan algılayıcılardan, uydulardan toplanan veriler her an inanılmaz boyutlarda artmaktadır.

1995 yılında birincisi düzenlenen Knowledge Discovery in Databases konferansı bildiri kitabı sunuşunda, enformasyon teknolojilerinin oluşturduğu veri dağları şu cümleler ile vurgulanmaktadır. “Dünyadaki enformasyon miktarının her 20 ayda bir ikiye katlandığı tahmin edilmektedir. Bu ham veri seli ile ne yapmamız gerekmektedir? İnsan gözleri bunun ancak çok küçük bir kısmını görebilecektir. Bilgisayarlar bilgelik pınarı olmayı vaat etmekte, ancak veri sellerine neden olmaktadır”.

Veri tabanı sistemlerinin artan kullanımı ve hacimlerindeki bu olağanüstü artış, organizasyonları elde toplanan bu verilerden nasıl faydalanılabileceği problemi ile karşı karşıya bırakmıştır. Geleneksel sorgu veya raporlama araçlarının veri yığınları karşısında yetersiz kalması, Veri Tabanlarında Bilgi Keşfi-VTBK (Knowledge Discovery in Databases) gibi yeni arayışlara neden olmaktadır. Bazı araştırmacıların VTBK ile veri madenciliğini aynı anlamda kullanmasına rağmen, Şekil 2.1’ de gösterildiği gibi veri madenciliği VTBK sürecinin bir aşamasıdır. Veri tabanlarında bilgi keşfi, depolanmış veri içerisindeki geçerli, yeni, faydalı ve sonuç olarak anlaşılabilir örüntülerin çıkarılması sürecidir.



Şekil 2. 1. Veri tabanlarında bilgi keşfi sürecinde veri madenciliğinin yeri

Veri tabanlarındaki bilgi keşfi sürecindeki aşamalar şu şekilde sıralanmaktadır: [2, 7]

Veri Seçimi (Data Selection): Bu adım birkaç veri kümesini birleştirerek, sorguya uygun örneklem kümesini elde etmeyi gerektirir. Bu işlem bir seçme işlemidir fakat seçme işlemi ile bu verilerin bütünleştirilmesi (veri bütünleştirme-data integration) de bu aşama içinde yer alan bir süreç olarak yer alır. Bu aşama veri toplama (data collection) olarak da isimlendirilmektedir.

Veri Ön İşleme (Data Preprocessing): Veri seçimi ile elde edilen veri kümesinde yer alan hatalı tutanakların çıkarıldığı ve eksik nitelik değerlerinin düzenlendiği aşamadır. Bu aşamada da uygulanan işlemler sayesinde, üzerinde veri madenciliği yapılacak veri kaliteli veri niteliğine bürünür. Veri temizleme (data cleaning) ve veri dönüştürme (data transformation) veri ön işleme işlemleridir.

Veri İndirgeme (Data Reduction): Seçilen örneklemden ilgisiz niteliklerin atıldığı ve tekrarlı tutanakların ayıklandığı adımdır. Bu aşama ile verinin boyutu indirgendiğinden seçilen veri madenciliği yönteminin çalışma zamanını iyileştirir.

Veri Madenciliği (Data Mining): Verilen bir veri madenciliği yöntemlerinin (sınıflama, kümeleme, birliktelik-ilişki eşleştirme, vb.) ve algoritmalarının (Apriori, vb.) uygulandığı adımdır.

Değerlendirme (Evaluation): Keşfedilen bilginin geçerlilik, yenilik, yararlılık ve basitlik kıstaslarına göre değerlendirilmesi aşamasıdır.

2.2. Veri Madenciliği Tanımı

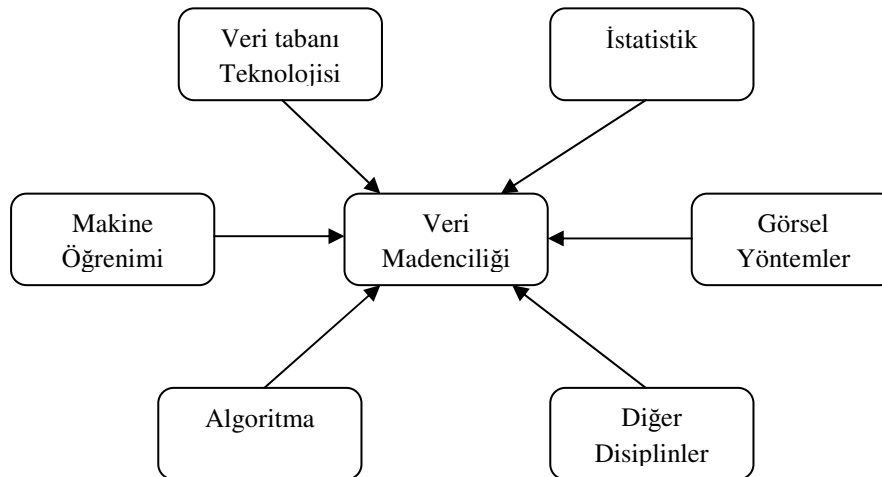
Veri madenciliği büyük miktarda veri içinden gelecekle ilgili tahmin yapılmasını sağlayan bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır. Veri madenciliği, potansiyel olarak faydalı, yeni ve mantıklı bilgi elde etmek için büyük veri tabanları üzerinde birden fazla basamaktan oluşan bir analiz yapmaktır. Veri madenciliğinin amacı, veri tabanlarında saklı olan ilişkileri ortaya koymaktır [8, 9].

Literatürdeki bazı veri madenciliği tanımları aşağıdaki gibidir.

- Han ve Kamber' e göre veri madenciliği, veri tabanları, veri ambarları veya diğer veri havuzlarında depolanmış büyük veri yığınlarından ilginç bilgilerin keşfi sürecidir [10].
- Turban veri madenciliğini şöyle tanımlamıştır: Kullanışlı bilgiyi büyük veri tabanlarından çıkarmak, tanımlamak ve elde etmek için istatistiksel, matematiksel, yapay zeka ve makine öğrenme tekniklerini kullanan bir süreçtir [11].
- Veri madenciliği gizli, bilinmeyen, potansiyel ve değerli, ilginç ya da eğitici bilgiye ulaşmada kullanılan araç ve yaklaşımlar olarak ifade edilir [12].
- Veri madenciliği önemli, geçerli, yeni ve kullanışlı bilgiyi büyük veri tabanlarından çıkarma işlemidir. Bu yüzden veri madenciliği, veri tabanından anlamlı örüntüler veya kurallar elde etmek için geniş bir araştırma alanı olarak görülebilir [1].
- Hand' a göre veri madenciliği, veri tabanı kullanıcıları için veri tabanlarındaki beklenmeyen ilginç, değerli ilişkilerin bulunmasını hedefleyen süreçtir [13].

2.3. Veri Madenciliğinin Diğer Disiplinlerle Olan İlişkisi

Veri madenciliği tanımlarından da anlaşılacağı üzere istatistik, makine öğrenmesi, veri tabanı yönetimi gibi alanlardan faydalanan disiplinler arası bir alandır. İşlenmemiş veriden, son kullanıcının kolayca anlayıp karar alma sürecine dahil edebileceği bilgiyi oluşturana kadar geçen tüm süreci kapsayan bir yöntem olmasından, hipotez doğrulamaya yönelik değil yeni, gizli örüntüler bulmaya yönelik bir alan olmasından ve çok çeşitli teknikleri aynı uygulama içinde kullanabilmeye olanak sağlamasından dolayı veri madenciliği kullanıcılarına kendisini oluşturan makine öğrenmesi, istatistik, matematik gibi yöntemlerden daha farklı bir perspektif sunar [14].



Şekil 2.2. Veri madenciliğini disiplinler arası bir alandır [15]

Veri miktarının büyümesi ile birlikte geleneksel istatistik yaklaşımları analiz ve çözüm için yeterli olmamaktadır. Veri madenciliğinde vurgulanan unsurlar istatistiğin tanımı içinde yer almaktadır. İstatistik verilerin toplanması, sınıflandırılması, özetlenmesi, grafik ve tablolarla sunulması, analiz edilerek ana kütle hakkında anlamlı bilgiler elde edilmesi ve yorumlar yapılmasıdır. Veri madenciliğinde ulaşılmak istenen amaç verilerden bilgiyi keşfetmek ve bunu karar vermede kullanmaktır. Veri madenciliğinde kullanılan temel aracın istatistiksel yöntemler olduğu birçok tanımda ve uygulamada vurgulanmaktadır. Her ikisinde de

temel olan öğeler veri ve bilgidir. Bu nedenle birbiriyle oldukça örtüşen konulardır [16].

Makine öğrenimi yöntemleri, veri madenciliği algoritmalarında kullanılan yöntemlerin çekirdeğini oluşturur. Makine öğreniminde kullanılan karar ağacı, kural tümevarımı pek çok veri madenciliği algoritmasında kullanılmaktadır. Makine öğrenimi önceki örnekleri ve sonuçları inceler ve bu işleri nasıl yeniden yapacağını öğrenir ve yeni durumlar hakkında genellemeler yapar. Genellikle bir makine öğrenim sistemi tek bir gözlemleyici kullanmaz, eğitim kümesi adı verilen bütün bir sistem kullanır. Bu set içinde örnek gözlem kodları bulunan ve makine tarafından okunabilen bazı formlar bulunur. Makine öğrenimi ile veri madenciliği arasında benzerliklerin yanı sıra farklılıklar da göze çarpmaktadır. Öncelikle veri madenciliği algoritmalarında kullanılan örneklem boyutu, makine öğreniminde kullanılan veri boyutuna nazaran çok büyüktür [16, 17].

Veri ambarlama, yapılandırılmış sorgu dili (SQL), geçici sorgular, raporlama, OLAP ve veri görselleştirme gibi yöntemler, veri madenciliği ile bazı ortak amaçlara sahip olmalarına rağmen, veri madenciliği bu yöntemlere alternatif olarak geliştirilmiş bir yöntem değildir. Aksine bu yöntemlerle birlikte kullanıldığında daha hızlı ve sağlıklı sonuçlara ulaşılmaktadır. Veri madenciliğinin en güçlü yönlerinden biri olan belirli bir soruya çözüm aramıyor oluşu, bir yandan OLAP uygulamaları ile farkını oluşturmakta, diğer yandan da, yine OLAP uygulamalarına destek vermesini sağlamaktadır. OLAP uygulamaları, spesifik sorulara cevap bulmayı amaçlar. Öncelikle kullanıcının bir ilişki ile ilgili bir hipotez geliştirmesini gerektirir. Sonrasında da, kullanıcı veri yığını içerisinde gerçekleştirdiği bir dizi sorgu ile bu hipotezin doğruluğunu test eder. Yani OLAP, kullanıcının ilişkileri kendisinin belirlemesini bekler. Buna karşın veri madenciliği uygulamalarının kullanıcıdan beklentisi sadece veri yığını üzerinde uygulanacak algoritmalara karar vermesidir.

Veri madenciliğini SQL ve geçici sorgulardan ve klasik raporlama araçlarından ayıran en önemli nokta, bu araçların oldukça statik bir yapıya sahip olmalarıdır. Veri madenciliği aslında bu klasik sorgu ve raporlama araçlarının evrimleşmiş halidir. Klasik olarak raporlarda kullanılan statik sorgular, veri madenciliği ile daha dinamik

bir yapıya kavuşmuştur. Veri madenciliği, altyapısında sorgulardan çok istatistiksel metotları kullanıyor olmasına rağmen, ortaya çıkardığı sonuçlar sorgulama araçları ile de elde edilebilir. Fakat klasik sorgulamaların yapamadığı, insan müdahalesi olmadan sonuç çıkarma işlemi, ancak veri madenciliği algoritmaları kullanılarak yapılabilmektedir. Klasik sorgulama ve raporlama araçlarında veri yığınının içinde tam olarak ne arandığı kullanıcı tarafından sisteme girilmek zorundadır. Örneğin, “2000 yılında şirketimizin Ankara şubesinde A ürününden kaç adet satılmıştır?”. Bu sorunun cevabı basit bir SQL cümlesi ile veri tabanından sorgulanabilir. Ayrıca bu sorgu, yıl, şube ve ürün çeşidi bilgilerinin kullanıcı tarafından girilip, sonucun SQL sorgusu ile öğrenilip çıktı olarak gösterildiği bir rapor haline getirilebilir. Klasik sorgu ve raporlama araçları, istenilen verileri bulup tek boyutlu olarak bize gösterir ve gücü bu noktada tükenir.

OLAP ise bir adım daha ileri gidip özet, karşılaştırma, analiz ve tahmin yapmak için veri yığınının derinlerine iner. Örneğin, “Yıl ve şube bazında A ürününün satış miktarı ile B ürününün satış miktarını karşılaştır” şeklinde bir isteği OLAP uygulamaları rahatlıkla yerine getirebilir. Fakat A ürünü ile B ürününü karşılaştırmamız gerektiğini nereden biliyoruz? Veri madenciliği kullanmadan bu iki ürün arasındaki ilişkiyi sadece sezgilerimizi kullanarak tahmin ederiz. Örneğin, beklentimiz, hangi ürünlerin aynı anda daha fazla miktarda satıldığını belirlemekse, OLAP ürünü ile tüm olasılıkları olumlu sonuç alana kadar tek tek denemek gerekir. Yani daha önce de değinildiği gibi, kullanıcı önce bir hipotez ortaya atıp, daha sonra OLAP uygulamaları ile bu hipotezi test etmelidir [18].

2.4. Veri Madenciliğinin Uygulandığı Depolama Birimleri

Veri madenciliği depolama birimlerinin her çeşidi üzerinde uygulanabilir. Bunlardan bazıları, veri ambarları (data warehouses), ilişkisel veri tabanları (relational databases), işlemsel veri tabanları (transactional databases), gelişmiş veri tabanları, dosyalar ve web üzerindeki hareketlerdir. Gelişmiş veri tabanı sistemleri arasında, nesneye yönelik, nesne ilişkisel, text veri tabanları, multimedya veri tabanları sayılabilir. Veri madenciliği tekniklerinin avantajları üzerinde uygulandığı depolama sistemlerine göre değişiklik gösterebilir [10].

2.4.1. Veri ambarları

Veri ambarı, bir işletmenin ya da kuruluşun değişik birimleri tarafından canlı sistemler aracılığı ile toplanan bilgilerin, gelecekte kullanılabilir ya da değerlendirilebilecek olanlarının arka planda üst üste yığılarak birleştirilmesinden oluşan büyük çaplı bir veri deposudur. Veri ambarları şirket yönetiminin stratejik karar verme sürecinde verilerin sistematik organizasyonu, anlaşılması ve kullanılması için entegre bir yapı sağlar. Veri ambarının öne çıkan özellikleri zaman dilimli, değişmeyen, bütünlük olmasıdır. Bütünlük özelliği, işletmedeki farklı veri kaynaklarının birleştirilip tek bir noktadan erişilebilir olmasını ifade eder. Zamanla dilimli özelliği, veri ambarındaki her anahtar yapı zamanın bir elemanı olarak hem kesinlik hem de açıklık içerir. Değişmeyen özelliği ise entegrasyonu sağlanan verilerin zamanla değişime uğramayan veriler olduğunu ifade etmektedir. Veriler veri ambarına aktarıldıktan sonra işlem görüp değiştirilmezler [10, 19].

Günümüzde işletmelerin bilgi sistemleri iki başlık altında toplanmaktadır. Bunlar:

1. Canlı Sistemler

Bu sistemlerde güncel veriler bulunmaktadır. Günlük yapılan işleri ve işlemleri gerçekleştirmek, sonuçları saklamak bu sistemlerin görevidir. Bu sistemler, marketlerde ya da mağazalarda stok takibi, üye borçları, satış işlemleri, ödeme kayıtları gibi bilgilerin işlendiği ve tutulduğu bilgi sistemleri olabilir. Bu tür canlı sistemlerde erişilebilirlik ana amaçtır; yani veriye en kısa sürede ulaşmak, işlemleri en kısa sürede sona erdirmek hedeflenir. Bu nedenden dolayı canlı sistemler çevrimiçi çalışma özelliğine sahiptir.

2. Karar Destek Sistemleri

İşletmelerde yer alan ikinci tür bilgi sistemleri ise karar destek sistemleridir. Bu sistemlerde yer alan bilgiler, çeşitli incelemelerden ve araştırmalardan geçerek, işletmelerin ileride karını ya da verimliliğini arttırması, gelecekte izlenecek politikalarının belirlenmesi, benzeri yönetimsel kararların alınmasına yardımcı olur. Bu sistemlerde verilerin erişimi birinci kriter değildir. Herhangi bir veriye herhangi bir zamanda çabuk erişmek gerekmez. Karar destek sistemlerinin önceliği performanstır. Karar destek sistemlerinde veriler, canlı sistemlere oranla çok daha

büyük boyutlardadır. Verilerin büyük boyutundan dolayı, verilerin incelenmesi ve incelemelerden sonuçlar çıkartılması, sistem kaynaklarını aşırı kullanmakta ve uzun süre almaktadır.

Veri ambarı, karar destek sistemi olarak nitelendirilebilir. Veri ambarı esasında günlük işlemlerin gerçekleştiği canlı sistemlerin arka planında bulunmaktadır. Canlı sistemlerde oluşan veriler periyodik olarak veri ambarına aktarılır. Bu periyodun seçimi tamamen veri ambarını kullanan işletmenin ihtiyaçları doğrultusunda belirlenir ve bir gün, bir hafta veya bir ay gibi çok değişken olabilir. Dolayısı ile veri ambarı çevrimdışı olarak çalışmaktadır. Yani veri ambarı içerisindeki kayıtlar güncel olmayabilir, çoğunlukla da güncel değildir. Tablo 2.1' de veri ambarı ile veri tabanı arasındaki farklar özetlenmiştir [20].

Tablo 2.1. Veri ambarı ve veri tabanı arasındaki farklar [21]

Veri Ambarı	Veri tabanı
Meta datalardan oluşur	Verilerden oluşur
Üst yönetime hitap eden KDS' lerdir	Organizasyonun her aşamasında veriye ulaşılır
Son kullanıcı sayısı azdır (<100)	Son kullanıcı sayısı fazladır(>1000)
Çevrimdışı çalışır	Çevrimiçi çalışır
Veri madenciliği gibi uzun ve karmaşık süreçleri sonucunda analizler yapılabilir	Sorgularla istenen sonuçlara anında ulaşılır
Tarihsel verilerden oluşur	Güncel verilerden oluşur

2.4.2. İlişkisel veri tabanları

Veri tabanı yönetim sistemi olarak ta adlandırılan bir veri tabanı sistemi, veri tabanı olarak bilinen birbiriyle ilişkili verilerden oluşmaktadır. Bir ilişkisel veri tabanı tablolardan oluşur. Her tablo tekil bir isim ile gösterilir ve özellik (kolonlar, alanlar) değerinden oluşmaktadır. Genellikle geniş bir satır (kayıtlar, satırlar) kümesi içerir. İlişkisel veri tabanlarındaki her satır, özellik değeri ile tanımlanan bir nesneyi temsil eder. Veri madenciliği, ilişkisel veri tabanlarındaki kayıtlara ait eğilimleri veya örüntüleri analiz etmek için uygulanabilir. Örneğin, bir veri madenciliği sistemi

müşterilerin gelir, yaş ve önceki kredi kartı bilgilerinde onların kredi risk oranını tahmin edebilir. İlişkisel veri tabanları, veri madenciliği uygulamaları için zengin bir bilgi havuzuna sahip en sık kullanılan biçimlerden birisidir [10].

2.4.3. İşlemsel veri tabanları

Genellikle işlemsel veri tabanı, her kaydın bir işlemini temsil ettiği dosyadan oluşur. Bir işlem (transaction) tipik olarak tekil bir işlem tanımlama numarası (TID) ve işlem içerisinde satın alınan nesnelerin listesini içerir. Tablo 2.2' de örnek bir işlemsel veri tabanı tablosu gösterilmiştir [10].

Tablo 2.2. Satışlar tablosu

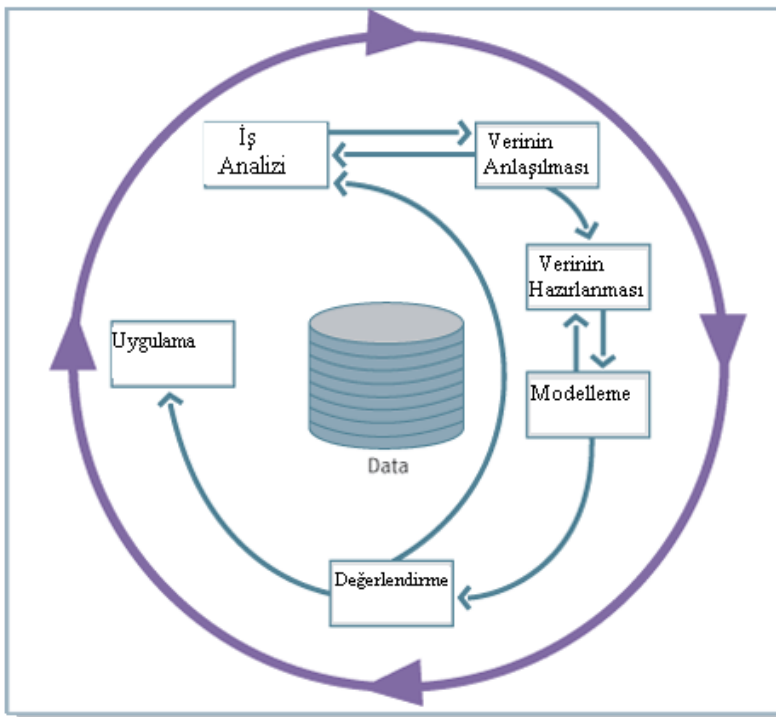
TID	Nesnelerin Listesi
T100	Ekmek, Süt
...	...

2.4.4. Gelişmiş veri tabanı sistemleri ve uygulamaları

Veri tabanı teknolojisinin gelişimi ile birçok yeni kuşak veri tabanı yöntemi uygulamalarda kullanılmaya başlanmıştır. Yeni veri tabanı uygulamaları, uzaysal veri (haritalar vb.), mühendislik tasarım verileri (bina tasarımları, sistem bileşenleri, devreler), multimedya veriler, zaman eksenli veriler ve web verileri gibi veriler üzerinde işlem yapmaktadırlar. Bu tür uygulamalar, karmaşık nesne yapıları, değişken boyutlu kayıt yapıları, metin (text) ve multimedya veriler için verilerdeki dinamik değişimler açısından daha etkin veri yapıları gerektirmektedirler. Bu ihtiyaçlara cevap verebilmek için, gelişmiş veri tabanı sistemleri geliştirilmiştir. Nesne tabanlı veri tabanları, uzaysal veri tabanları, zaman serisi veri tabanları, metin veri tabanları gelişmiş veri tabanı sistemleridir [19].

2.5. Veri Madenciliği Süreci: CRISP-DM

Birçok işletme kendi problemlerine, verilerine göre kendi veri madenciliği sürecini oluşturmaktadır. Ancak veri madenciliği sürecinin oluşturulmasında yapılan yanlışlıklar, sürecin etkinliğine zarar vermektedir. Veri madenciliği sürecinin standartlaştırılması konusunda farklı grup, kurum ve şirketler çeşitli standartlar oluşturmuşlardır. Bunlardan en çok takip edileni Daimler Chrysler ve SPSS tarafından 1996 yılında oluşturulan süreçtir. Şekil 2.3' te gösterilen bu sürece Sektörler Arası Standart Veri Madenciliği Süreci (CRISP-DM) adı verilmiştir [22].



Şekil 2.3. CRISP-DM metodolojisi

CRISP-DM metodolojisine göre veri madenciliği projesin 6 safhadan oluşmaktadır:

- İş Analizi: Bu adımda iş perspektifi ile problemlerin anlaşılması, proje hedeflerinin belirlenmesi gerekir. Belirlenen amaçlar ve gereksinimler doğrultusunda veri madenciliği problem tanımı oluşturulur.

- Verinin Anlaşılması: Verinin anlaşılması aşaması veri kaynaklarına bağlanma, veriyi tanıma, verinin kalitesini anlama ve verinin grafiksel olarak incelenmesi ve veri gruplarını değerlendirme çalışmalarını içerir.
- Verinin Hazırlanması: Veri madenciliği prosesinde kullanılacak olan veri setinin modellemeye hazırlanması aşamasıdır. Hatalı veya analizin yanlış yönlendirilmesine neden olabilecek veriler temizlenir. Eksik veriler önemli ise tahmin edilmeye çalışılır, önem taşıyor ise silinir. Bu aşama en çok iş gücü gerektiren ve toplam süreç içinde en fazla zaman alan aşamadır.
- Modelleme: Bir veri madenciliği problemi için birden fazla teknik kullanılabilir, problem için uygun olan teknik veya tekniklerin bulunabilmesi için birçok teknik oluşturulup bunların içinden en uygun olanlar seçilir. Model oluşturulduktan sonra kullanılan tekniğin gereksinimlerine uygun olarak veri hazırlanması aşamasına tekrar dönülüp gerekli değişiklikler yapılabilir.
- Değerlendirme: Bu aşamada, daha önce oluşturulmuş olan model, uygulamaya koyulmadan önce son kez tüm yönleriyle değerlendirilir, kalitesi ve etkinliği ölçülür. Modelin ilk aşamada oluşturulan proje amacına ulaşmada etkin olup olmadığı ve problemin tüm yönleri için bir çözüm sağlayıp sağlamadığı karara bağlanır. Modelin anlaşılabilirliği ve doğruluk oranı gibi konularda da model amaç için yeterli kaliteyi sağlıyorsa uygulama aşamasına geçilir.
- Uygulama: Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama olabileceği gibi başka bir uygulamanın alt parçası olarak da kullanılabilir. İşlenen veri kullanıcının anlayabileceği, karar alma sürecinde kullanılacak bir şekilde son kullanıcıya verilir [23, 24].

2.6. Veri Madenciliği Uygulama Alanları

Günümüzde veri madenciliği teknikleri işletme ve bilim çevrelerinde yaygın olarak kullanılmaktadır. Veri madenciliğinin kullanıldığı sektörler ve uygulama alanları aşağıda özetlenmiştir [25].

Pazarlama

- Müşterilerin satın alma örüntülerinin belirlenmesi
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması
- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması
- Market sepeti analizi
- Çapraz satış analizi
- Müşteri ilişkileri yönetimi
- Müşteri değerlendirme
- Satış tahmini

Bankacılık

- Farklı finansal göstergeler arasında gizli korelasyonların bulunması
- Kredi kartı dolandırıcılıklarının tespiti
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi
- Kredi taleplerinin değerlendirilmesi
- Usulsüzlük tespiti
- Risk analizleri ve yönetimi

Sigortacılık

- Yeni poliçe talep edecek müşterilerin tahmin edilmesi
- Sigorta dolandırıcılıklarının tespiti
- Riskli müşteri örüntülerinin belirlenmesi

Perakendecilik

- Satış noktası veri analizleri
- Alışveriş sepeti analizleri
- Tedarik ve mağaza yerleşim optimizasyonu

Borsa

- Hisse senedi fiyat tahmini
- Genel piyasa analizleri
- Alım-satım stratejilerinin optimizasyonu

Telekomünikasyon

- Kalite ve iyileştirme analizi
- Hisse tespiti
- Hatların yoğunluk tahmini

Sağlık ve İlaç

- Test sonuçlarının tahmini
- Ürün geliştirme
- Tıbbi teşhis
- Tedavi sürecinin belirlenmesi

Endüstri

- Kalite kontrol analizi
- Lojistik
- Üretim süreçlerinin optimizasyonu

Bilim ve Mühendislik

- Ampirik veriler üzerinde modeller kurarak bilimsel ve teknik problemlerin çözümlenmesi

Eğitim

- Öğrenci davranışlarının öngörülmesi
- Öğrencilerin ders seçme eğilimlerinin belirlenmesi

2.7. Veri Madenciliği Modelleri

Veri madenciliği modelleri genel olarak tahmin edici ve tanımlayıcı olmak üzere iki başlık altında toplanmaktadır.

Tahmin edici modellerde, sonuçları bilinen verilerde hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir.

Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır.

Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılacak mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır. X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu aileler ile çocuğu olmayan ve geliri X/Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerlik gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir.

Veri madenciliği gördükleri işlevlerine göre üç ana başlık altında toplanır: Sınıflama (classification) ve regresyon, kümeleme (clustering), birliktelik kuralları ve ardışık zamanlı örüntüler (association rules and sequential patterns). Sınıflama ve regresyon tahmin edici modeller; kümeleme, birliktelik kuralları ve ardışık zamanlı örüntüler tanımlayıcı modellerdir [7].

2.7.1. Sınıflama ve regresyon

Sınıflama ve regresyon modelleri, mevcut verilerden hareket ederek geleceğin tahmin edilmesinde kullanılır. Veri madenciliği teknikleri içerisinde yaygın bir kullanıma sahiptir. Sınıflama ve regresyon modeli denetimli öğrenme yöntemidir. İstenilen bir değişken bağımlı değişken ve diğerleri tahmin edici (bağımsız) değişkenler olarak adlandırılır. Sınıflama ve regresyon modelleri arasındaki temel fark; bağımlı değişken sayısal değil ise problem sınıflama problemi, eğer bağımlı değişken sayısal ise problem regresyon problemi olarak adlandırılır [7,19].

Regresyon, sürekli sayısal bir değişkenin, aralarında doğrusal ya da doğrusal olmayan bir ilişki bulunduğu varsayılan diğer değişkenler yardımıyla tahmin edilmesi yöntemidir. Regresyon modeli, sayısal değerleri tahmin etmeye yönelik olması dışında sınıflandırma yöntemine benzetilebilir. Çok terimli lojistik regresyon gibi kategorik değerlerin de tahmin edilmesine olanak sağlayan tekniklerin

geliştirilmesi ile sınıflandırma ve regresyon modelleri giderek birbirine yaklaşımakta ve dolayısıyla aynı tekniklerden yararlanılması mümkün olmaktadır [26, 27].

Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler aşağıda verilmiştir.

- Karar Ağaçları (Decision Trees)
- Yapay Sinir Ağları (Artificial Neural Networks)
- Genetik Algoritmalar (Genetic Algorithms)
- K-En Yakın Komşu (K-Nearest Neighbour)
- Bellek Tabanlı Yöntemler (Memory Based Reasoning)
- Naive-Bayes
- Doğrusal ve Lojistik Regresyon

2.7.2. Kümeleme

Kümeleme modellerinde amaç, küme üyelerinin birbirlerine çok benzediği, ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veri tabanındaki kayıtların bu farklı kümelerle bölünmesidir. Başlangıç aşamasında veri tabanındaki kayıtların hangi kümelerle ayrılacağı veya kümelemenin hangi değişken özelliklerine göre yapılacağı bilinmediğinden, kümeleme denetimsiz öğrenme yöntemidir. Sınıflama ile kümelemeyi birbirinden ayıran en önemli fark, kümeleme işleminin sınıflama işleminde olduğu gibi önceden belirlenmiş bir takım sınıflara göre bölme yapmamasıdır. Sınıflamada her bir veri, önceden sınıflandırılmış bir takım sınıflar üzerinde yapılan bir eğitim neticesinde ortaya çıkan bir modele göre önceden belirlenmiş olan bir sınıfa atanmaktadır. Kümeleme işleminde ise önceden tanımlanmış sınıflar ya da örnek sınıflar bulunmamaktadır. Verilerin kümeleneceği işlemi, verilerin birbirlerine olan benzerliklerine göre yapılmaktadır. Oluşan sınıfların hangi anlamları taşıdığına belirlenmesi tamamen çözümlenmeyi yapan kişiye kalmıştır [19, 26].

Kümelemede, genellikle k-ortalama algoritması ya da kohonen şebekesi gibi istatistiksel yöntemler kullanılmaktadır. Bütün yöntemlerde işleyiş aynı şekildedir. Her kayıt var olan kümelerle karşılaştırılır. Bir kayıt kendisine en yakın kümeyle atanır ve bu kümeyi tanımlayan değeri değiştirir. Optimum çözüm bulununcaya

kadar kayıtlar yeniden atanır ve küme merkezleri ayarlanır. En yaygın kullanılan kümeleme algoritması “k ortalamalar algoritması”dır [28].

2.7.3. Birliktelik kuralları ve ardışık zamanlı örüntüler

Birliktelik kuralları, bir veri kümesindeki nesnelere arasındaki ilişkileri bulmayı amaçlayan denetimsiz öğrenen bir veri madenciliği modelidir.

Birliktelik kuralları kullanışlı ve anlaşılması kolay olduğundan finans, telekomünikasyon, pazarlama, perakendecilik ve online ticaret gibi endüstriyel alanlarda geniş bir alana yayılmıştır [29].

Birliktelik kurallarının kullanıldığı en yaygın alanlardan biri süpermarket uygulamalarıdır. Bu uygulamalar literatürde market sepet analizi (market basket analysis) olarak adlandırılmaktadır. Market sepeti analizi hangi nesnelere birlikte satılma eğilimi gösterdiği bilgisini vermektedir.

Birliktelik kuralları ve ardışık zamanlı örüntüleri birbirinden ayıran özellik zaman kavramının uygulamada olmasıdır. Belli bir dönem boyunca nesnelere arasındaki birlikteliklerin incelenmesi “ardışık zamanlı örüntü çözümlemesi” olarak da isimlendirilir. Ardışık analize ait örnekler aşağıda verilmiştir [19, 28].

- “Çadır alan müşterilerin %10’u bir ay içerisinde sırt çantası almaktadır.”
- “A hissesi %15 artarsa üç gün içinde B hissesi %60 olasılıkla artacaktır.”

Birliktelik kuralları bu çalışmanın ana konusunu oluşturduğundan üçüncü bölümde ayrıntılı olarak incelenmiştir.

2.8. Veri Madenciliği Teknikleri

Bu bölümde, veri madenciliği tekniklerinden en bilinenleri hakkında bilgiler verilmiştir.

2.8.1. Karar ağaçları

Karar ağaçları, yaygın olarak kullanılan sınıflama algoritmalarından biridir. İlk basamak öğrenme basamağıdır. Öğrenme basamağında önceden bilinen bir eğitim verisi, model oluşturmak amacı ile sınıflama algoritması tarafından çözümlenir. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. İkinci basamak ise sınıflama basamağıdır. Sınıflama basamağında test verisi, sınıflama kurallarının veya karar ağacının doğruluğunu belirlemek amacıyla kullanılır. Eğer doğruluk kabul edilebilir oranda ise, kurallar yeni verilerin sınıflanması amacıyla kullanılır [8, 19].

En yaygın kullanılan karar ağacı algoritmaları aşağıda verilmiştir.

- CHAID (Chi-Squared Automatic Interaction Detector , Kass 1980),
- C&RT (Classification and Regression Trees, Breiman ve Friedman, 1984),
- ID3 (Induction of Decision Trees, Quinlan, 1986),
- C4.5 (Quinlan, 1993).

2.8.2. Yapay sinir ağları

Yapay sinir ağları, tanımlayıcı ve tahmin edici veri madenciliği algoritmalarındandır. YSA, basit biyolojik sinir sisteminin çalışma şekli simule edilerek tasarlanan programlama yaklaşımıdır. Simule edilen sinir hücreleri (nöronlar) içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA' lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir. Bu teknik, veri tabanındaki örüntüleri, sınıflandırma ve tahminde kullanılmak üzere genelleştirir. Sinir ağları algoritmaları sayısal veriler üzerinde çalışırlar, denetimli öğrenme yöntemidir [28, 30].

2.8.3. Genetik algoritmalar

Genetik algoritmalar, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar. Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir.

Genetik algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme (popülasyon) oluştururlar. Genetik algoritmalar, diğer eniyileme yöntemleri kullanılırken büyük zorluklarla karşılaşılan, oldukça büyük arama uzayına sahip problemlerin çözümünde başarı göstermektedir. Bir problemin bütünsel en iyi çözümünü bulmak için garanti vermezler. Ancak problemlere makul bir süre içinde, kabul edilebilir, iyi çözümler bulurlar [31].

2.8.4. K - en yakın komşu

Veri uzayında birbirine yakın olan aynı tip kayıtlar, birbirlerinin komşusu durumundadırlar. Bu anlayış doğrultusunda, çok kolay fakat güçlü olan k-en yakın komşu algoritması geliştirilmiştir. K-en yakın komşu algoritmasının temel felsefesi komşunun yaptığını yaptırmaktır. Belirli bir bireyin (kayıtın) davranışını (özelliğini) tahmin etmek istenirse, veri uzayında o bireye yakın olan örneğin 10 bireyin davranışına bakılabilir. Bu 10 komşunun davranışının ortalaması hesaplanır ve bu hesaplanan ortalama bireylerin tahmini olur. K-en yakın komşudaki k harfi araştırdığımız komşu sayısıdır. Örneğin, 5-en yakın komşuda 5 komşuya bakılır [19].

2.8.5. Bellek tabanlı yöntemler

İnsanlar kararlarını genellikle daha önce yaşadıkları deneyimlere göre verirler. Örneğin doktorlar bir hastayı incelerken, elde ettiği bulguları daha önce tedavi ettiği

benzer hastalığa yakalanmış hastalar üzerindeki deneyimlerini kullanarak değerlendirirler. Bellek tabanlı yöntemler de benzer şekilde deneyimleri kullanmaktadır. Bu yöntemlerde, bilinen kayıtların bulunduğu bir veri tabanı oluşturulur ve sistem yeni gelen bir kayda komşu olan diğer kayıtları belirler ve bu kayıtları kullanarak tahminde bulunur ya da bir sınıflama işlemi uygular. Bellek tabanlı yöntemlerin en önemli özelliği veriyi olduğu gibi kullanabilme yeteneğidir. Diğer VM yöntemlerinin aksine bellek tabanlı yöntemler, kayıtların şekli (format) yerine sadece iki işlemin varlığı ile ilgilenir. Bu işlemler, iki kayıt arasındaki uzaklığı belirleyen bir uzaklık fonksiyonu ve komşu kayıtları işleyerek bir sonuç üreten kombinasyon fonksiyonudur.

Bellek tabanlı yöntemler sahtekârlık tespiti ve klinik işlemler gibi alanlarda kullanılmaktadır [19].

2.8.6. Naive bayes

Naive Bayes, hedef değişkenle bağımsız değişkenler arasındaki ilişkiyi analiz eden tahminci ve tanımlayıcı bir sınıflama algoritmasıdır.

Naive Bayes, sürekli veri ile çalışmaz. Bu nedenle sürekli değerleri içeren bağımlı ya da bağımsız değişkenler kategorik hale getirilmelidir. Örneğin; bağımsız değişkenlerden biri yaş ise, sürekli değerler “<20”, “21–30”, “31–40” gibi yaş aralıklarına dönüştürülmelidir.

Naive Bayes, modelin öğrenilmesi esnasında, her çıktının öğrenme kümesinde kaç kere meydana geldiğini hesaplar. Bulunan bu değer, öncelikli olasılık olarak adlandırılır. Örneğin; bir banka kredi kartı başvurularını “iyi” ve “kötü” risk sınıflarında gruplandırmak istemektedir. İyi risk çıktısı toplam 5 vaka içinde 2 kere meydana geldiyse iyi risk için öncelikli olasılık 0,4’tür. Bu durum, “Kredi kartı için başvuran biri hakkında hiçbir şey bilinmiyorsa, bu kişi 0,4 olasılıkla iyi risk grubundadır” olarak yorumlanır. Naive Bayes aynı zamanda her bağımsız değişken/bağımlı değişken kombinasyonunun meydana gelme sıklığını bulur. Bu sıklıklar öncelikli olasılıklarla birleştirilmek suretiyle tahminde kullanılır [28].

2.8.7. Regresyon

Regresyon analizi, bir ya da daha fazla bağımsız değişken ile hedef değişken arasındaki ilişkiyi matematiksel olarak modelleyen bir yöntemdir. Veri madenciliğinde yaygın olarak kullanılan regresyon modellerinden doğrusal regresyonda tahmin edilecek olan hedef değişken sürekli değer alırken; lojistik regresyonda hedef değişken kesikli bir değer almaktadır. Doğrusal regresyonda hedef değişkenin değeri; lojistik regresyonda ise hedef değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilmektedir [28].

2.8.8. K ortalamalar algoritması

Diğer kümeleme teknikleri ile karşılaştırıldığında k ortalamalar algoritması (k means, simple k means) büyük veri tabanlarının kümelenmesinde oldukça etkin bir algoritmadır. Yeni bir vaka ortaya çıktığında; algoritma tüm veriyi inceleyerek buna en çok benzeyen vakaların bir alt kümesini oluşturur ve onları çıktığı tahmin etmek için kullanır.

Algoritmanın adımları şu şekildedir:

1. Veri seti rassal olarak k adet başlangıç kümesine ayrılır.
2. Veri setinde yer alan örnekler; merkezi kendisine en yakın olan kümeye atanır.
3. Her atamanın sonunda küme merkezi (ortalama) yeniden hesaplanır.
4. Veri setindeki tüm örneklerin ataması yapılanaya kadar 2. ve 3. adımlar tekrarlanır.

Yeni bir vakanın ait olduğu kümeyi belirlemek için algoritma yeni vakanın öğrenme verisindeki her bir vakadan uzaklığını hesaplar. K değerinin ve uzaklık ölçüsünün modelin kalitesi üzerinde büyük etkisi vardır bu nedenle onları dikkatle seçmek çok önemlidir. K ortalamalar algoritması oldukça etkin bir algoritma olmakla birlikte; sadece nümerik veri ile çalışır fakat veri madenciliği uygulamaları sıklıkla kategorik verileri de içermektedir. K ortalamalar algoritmasının geliştirilmesi ile elde edilen k modlar algoritması ise kategorik veriler üzerinde çalışabilen bir algoritmadır [28].

2.9. Veri Madenciliğinde Karşılaşılan Problemler

Küçük veri kümelerinde hızlı ve doğru bir biçimde çalışan bir sistem, çok büyük veri tabanlarına uygulandığında tamamen farklı davranabilir. Bir veri madenciliği sistemi tutarlı veri üzerinde mükemmel çalışırken, aynı veriye gürültü eklendiğinde kayda değer bir biçimde kötüleşebilir. Veri madenciliğinde karşılaşılan problemler devam eden maddelerde açıklanmıştır [17, 32].

- Veri tabanı boyutu: Veri tabanı boyutları inanılmaz bir hızla artmaktadır. Pek çok makine öğrenimi algoritması birkaç yüz tutanaklık oldukça küçük örneklemi ele alabilecek biçimde geliştirilmiştir. Aynı algoritmaların yüz binlerce kat büyük örneklerde kullanılabilmesi için azami dikkat gerekmektedir. Örneklemin büyük olması, örüntülerin gerçekten var olduğunu göstermesi açısından bir avantajdır ancak böyle bir örneklemden elde edilebilecek olası örüntü sayısı da çok büyüktür. Bu yüzden VM sistemlerinin karşı karşıya olduğu en önemli sorunlardan biri veri tabanı boyutunun çok büyük olmasıdır. Dolayısıyla VM yöntemleri ya sezgisel bir yaklaşımla arama uzayını taramalıdır, ya da örneklemini yatay/dikey olarak indirgemelidir. Yatay indirgeme sürekli değerlerin belirli aralık değerlerine dönüştürülmesi ile ortaya çıkabilecek tekrarlı çoklular tekil hale getirilerek sağlanabilir. Dikey indirgeme, artık niteliklerin indirgenmesi işlemidir ve “artık işleme” maddesi altında verilmiştir.
- Gürültülü veri: Büyük veri tabanlarında pek çok niteliğin değeri yanlış olabilir. Bu hata, veri girişi sırasında yapılan insan hataları veya girilen değerlerin yanlış ölçülmesinden kaynaklanır. Veri girişi veya veri toplanması sırasında oluşan sistem dışı hatalara gürültü adı verilir. Günümüzde kullanılan ticari ilişkisel veri tabanları, veri girişi sırasında oluşan hataları otomatik biçimde gidermek konusunda az bir destek sağlamaktadır. Hatalı veri gerçek dünya veri tabanlarında ciddi problem oluşturabilir. Bu durum, bir VM yönteminin kullanılan veri kümesinde bulunan gürültülü verilere karşı daha az duyarlı olmasını gerektirir. Gürültülü verinin yol açtığı problemler tümevarımsal karar ağaçlarında uygulanan metotlar bağlamında kapsamlı bir biçimde araştırılmıştır. Eğer veri kümesi gürültülü ise sistem bozuk veriyi tanımalı ve ihmal etmelidir.

- Null deęerler: Veri tabanlarında null (boş) deęeri birincil anahtarlar yer almayan herhangi bir nitelięin deęeri olabilir. Eęer bir nitelik deęeri null ise o nitelik bilinmeyen ve uygulanmaz bir deęere sahiptir. Bu durumla iliřkisel veri tabanlarında sıklıkla karřılařılmaktadır. Bir iliřkide yer alan tüm çoklular aynı sayıda nitelięe, nitelięin deęeri null olsa bile sahip olmalıdır. Örneęin, kiřisel bilgisayarların özelliklerini tutan bir iliřkide bazı model bilgisayarlar için ses kartı modeli nitelięinin deęeri null olabilir. Boř deęerli nitelikler veri kümesinde bulunuyorsa, ya bu çoklular tamamıyla ihmal edilmeli ya da bu çoklularda nitelięe olası en yakın deęer atanmalıdır.
- Eksik veri: Evrendeki her nesnenin ayrıntılı bir biçimde tanımlandığı ve bu nesnelerin alabileceęi deęerler kümesinin belirli olduęu varsayılırsın. Verilen bir bağlamda her bir nesnenin tanımı kesin ve yeterli olsa idi sınıflama iřlemi basitçe nesnelerin alt kümelerinden faydalanılarak yapılırdı. Bununla birlikte, veriler kurum ihtiyaçları göz önünde bulundurularak düzenlenip toplandıęından, mevcut veri bilgi keřfi açasından uygun olmayabilir. Örneęin hastalıęın tanısını koymak için kurallar sadece çok yařlı insanların belirtilerinin bulunduęu bir veri kümesi kullanılarak üretilseydi, bu kurallara dayanarak bir çocuęa tanı koymak pek doęru olmazdı. Bu gibi kořullarda bilgi keřfi modeli belirli bir güvenlik (veya doęruluk) derecesinde tahmini kararlar alabilmelidir.
- Artık veri: Verilen veri kümesi, eldeki probleme uygun olmayan veya artık nitelikler içerebilir. Bu durum pek çok iřlem sırasında karřımıza çıkabilir. Örneęin, eldeki problem ile ilgili veriyi elde etmek için iki iliřkiyi ortak nitelikler üzerinden birleřtirirsek, sonuç iliřkide kullanıcının farkında olmadıęı artık nitelikler bulunur. Artık nitelikleri elemek için geliřtirilmiř algoritmalar özellik seçimi olarak adlandırılır.
- Dinamik veri: Kurumsal çevrim içi veri tabanları dinamiktir, yani içerięi sürekli olarak deęiřir. Bu durum, bilgi keřfi metotları için önemli sakıncalar doğurmaktadır. İlk olarak sadece okuma yapan ve uzun süre çalışan bilgi keřfi metodu, bir veri tabanı uygulaması olarak mevcut veri tabanı ile birlikte çalıştırıldıęında mevcut uygulamanın da performansı ciddi ölçüde düşer. Dięer bir sakınca ise, veri tabanında bulunan verilerin kalıcı olduęu varsayıлып, çevrim

dışı veri üzerinde bilgi keşfi metodu çalıştırıldığında, değişen verinin elde edilen örüntülere yansması gerekmektedir. Bu işlem, bilgi keşfi metodunun ürettiği örüntüleri zaman içinde değişen veriye göre sadece ilgili örüntüleri yığılmalı olarak günleme yeteneğine sahip olmasını gerektirir. Aktif veri tabanları tetikleme mekanizmalarına sahiptir ve bu özellik bilgi keşfi metotları ile birlikte kullanılabilir.

2.10. Veri Madenciliği Yazılımları

Bu bölümde, günümüzün en çok tercih edilen veri madenciliği yazılımlarından SPSS Clementine, SAS Enterprise Miner, Insightful Miner, Oracle Darwin, STATISTICA Data Miner hakkında bilgi verilmiştir [19, 32, 33].

2.10.1. Clementine

Clementine SPSS Inc. firmasının veri madenciliği aracıdır. İngiltere’de geliştirilen Clementine, oldukça uzun bir süredir kullanılmakta ve araçları giderek iyileştirilmektedir. Clementine yapay sinir ağları, karar ağaçları, kümeleme, regresyon, birliktelik kuralları algoritmaları mevcuttur. Clementine müşteri ilişkileri yönetimi, kimya sektöründe maddelerin aşındırıcılık tahmininde ve bankacılık alanında kredi kartı dolandırıcılıkları gibi konularda kendine uygulama alanı bulmuştur. Bazı avantajları ve dezavantajları aşağıdaki gibidir.

- Veri madenciliği algoritmaları çok çeşitlidir ve oldukça güçlü, optimal parametreyi araştıran programlar mevcuttur,
- Veri kalitesinin kontrolü için araçlar mevcuttur,
- Ortalama kullanım kolaylığı vardır,
- Grafik özelliği açısından diğer programlara göre zayıftır.

2.10.2. Enterprise miner

SAS firmasının veri madenciliği aracıdır. Enterprise Miner karar ağaçları, yapay sinir ağları, regresyon çözümlemesi, kümeleme, zaman serileri, birliktelik kurallarının bulunması gibi VM sorgularını ele alabilmektedir. Grafikselle arayüzü sayesinde

kullanım kolaylığı sağlar. Diğer VM programlarına oranla istatistik ve regresyon açısından ile birlikte en fazla araca sahiptir. Algoritma derinliği ve görsel arabirim güçlü, zor kullanılır olması zayıf olduğu yönleridir.

2.10.3. Insightful miner

Görel olarak daha basit VM projelerini yapacak olan sıradan çözümleyiciler için mevcut olan belki de en iyi programdır. Statistica Data Miner kadar olmasa da zengin istatistiksel çözümleme algoritmalarına sahiptir. Veri alma/verme, veri inceleme ve veri temizleme ile boyut indirgeme için mükemmel araçlara sahip olması, grafiksel programlama arayüzüne sahip olmamasına rağmen veri madencisi olmayanlar tarafından kolaylıkla kullanılabilmesi ve genel kullanım için mevcut olan programlar arasında görel olarak ucuz olması avantajlarındandır. Diğer programlara göre daha düşük seviyede otomatik olması, karmaşık problemler için yazı (script) arayüzünün olmaması ve dışarıdan model alma özelliğinin olmaması bazı dezavantajlarındandır.

2.10.4. Darwin

Darwin Oracle firmasının VM aracıdır. Darwin regresyon ağaçları, karar ağaçları, kümeleme, yapay sinir ağları, Bayesian öğrenme, k-en yakın komşu gibi birçok algoritmayı destekleyen bir VM aracıdır. Paralel sunucular için geliştirilmiş bir VM sistemidir. Darwin kullanımı kolay bir ara yüze sahiptir. Darwin VM algoritmalarından CART, StarTree, StarNet ve StarMatch'i kullanmaktadır.

2.10.5. Statistica data miner

STATISTICA Data Miner, kendine has kategoride bir programdır. VM projelerindeki tüm görevleri kolaylaştırmadaki başarısı ve birçok işlemi başarıyla gerçekleştirmesi açısından eşsizdir. Diğer programların kullanımı daha kolay olabilir (Örneğin Insightful Miner) ya da daha otomatik olabilirler ancak hiçbir VM programı STATISTICA Data Miner kadar fazla araç sunamamaktadır.

Avantajları;

- VM için parametrik istatistik ve makine öğrenimi kombinasyonu zengin algoritmalar sunar,
- Diğer programlara göre grafiksel programlama arayüzüyle kullanımı kolaydır,
- Tüm ortak VM görevleri için araçlar sunar,
- Model çıktısı için oldukça esnek araçlar mevcuttur,
- Boyut azaltmada kullanılan güçlü araçları mevcuttur,
- Ölçeklenebilirlik özelliği ile boyut ve büyüklük açısından daha geniş veri kümelerini diğer programlardan daha hızlı işleyebilir, Visual Basic dilini temel alan güçlü özelleştirme araçları mevcuttur.

Dezavantajı;

- Sinir ağı modellerinin değerlendirilmesi için kullanılan grafikler kolaylıkla elde edilememektedir.

BÖLÜM 3. BİRLİKTELİK KURALLARI

3.1. Birliktelik Kuralları Tanımı

Birliktelik kuralları veriler arasındaki güçlü birliktelikleri veya korelasyonları elde etmeye yarayan bir veri madenciliği tekniğidir. Çok sayıda verinin depolandığı bir veri tabanı içinde çeşitli nitelikler arasında hemen fark edilmeyen bir takım ilişkilerin ortaya çıkarılması sürecine birliktelik kuralı madenciliği denilmektedir.

Birliktelik kuralları büyük miktarlardaki veriler arasından ilginç birliktelik örüntülerini keşfederek pazarlama, karar verme ve iş yönetimine oldukça fayda sağlamaktadır. Bu yüzden veri tabanlarında bilgi keşfinde yapılan araştırmalarda birliktelik kuralları konusu odak noktası olmaktadır. Birliktelik kuralları kullanışlı ve anlaşılması kolay olduğundan finans, telekomünikasyon, pazarlama, perakendecilik ve online ticaret gibi endüstriyel alanlarda geniş bir alana yayılmıştır [29, 34].

Böylece son yıllarda birliktelik kuralları üzerine yapılan akademik çalışmaların da artmasına neden olmuştur. Apriori benzeri algoritmalar gibi algoritmik yenilikler, inter-transaksiyonel birliktelik kuralları, bulanık mantık tabanlı birliktelik kuralları, genetik algoritma tabanlı birliktelik kuralları, sinir ağı tabanlı birliktelik kuralları, niceliksel birliktelik kuralları, çoklu boyutlu ve çoklu seviyeli birliktelik kuralları, geçici birliktelik kuralları, genelleştirilmiş birliktelik kuralları yapılan çalışmaların bazı örneklerindendir [10, 35, 36, 37, 38, 39, 40, 41, 42].

Birliktelik kurallarının kullanıldığı en yaygın alanlardan biri süpermarket uygulamalarıdır. Bu uygulamalar literatürde market sepeti analizi (market basket analysis) olarak adlandırılmaktadır. Geleneksel birliktelik kurallarından market sepeti analizi bir ev alışverişinde sadece satın alınan farklı ürün kategorileri (örneğin ürün kombinasyonları) arasındaki birliktelikleri keşfedebilmektedir. Market sepeti

analizi, hangi ürünlerin birlikte satılma eğiliminde olduğu bilgisini vererek müşteri davranışları, stok kontrol, satış stratejileri ve yerleşim düzeni gibi analizlerde fayda sağlamaktadır [43].

Market sepeti analizi, bir alışveriş boyunca satın alınan ürünlerin birleşimini incelemektedir. Ürünlerin marka veya ürün tipi bazında gruplanıp gruplanmadığı kararını kolaylaştırdığından perakendeciler müşterilerin çapraz-kategorik satın alma davranışları ile ilgili bilgilere oldukça önem vermektedirler. Çünkü bir perakende satış yerinin yerleşim düzeni perakendecinin bilgi işleme, satın alma davranışları gibi toplam performansını önemli ölçüde etkileyen bir faktör olduğu tespit edilmiştir. Sepet analizi, farklı bakış açılarındaki müşteri dağılımını da göstermektedir. Bu dağılım bilgileri planlama, reklam dizaynı, indirim-promosyon, mağaza düzeni ve ürün yatırımı şeklindeki kararlara yardımcı olmaktadır [44, 45].

Market sektörüne yönelik son yıllardaki literatür çalışmalarına bakıldığında Liao ve diğerleri Apriori algoritmasını kullanarak elde ettiği ürün haritalarını yeni ürün geliştirme kaynağı olarak önermiş, Yang ve Lai çevrimiçi alışveriş davranışlarının bilgisine dayanarak ürün promosyonundaki kararların performansını karşılaştırmış, Liao ve Chen birliktelik kurallarını elektronik katalog pazarlama ve indirim yönetiminde kullanmış, Chen ve Lin ise Apriori algoritması ve 0-1 tamsayı algoritmayı kullanarak raf düzeni yönetimi çalışması yapmıştır [43, 44, 46, 47].

3.2. Birliktelik Kuralları Matematiksel Modeli Ve Temel Kavramları

3.2.1. Birliktelik kuralları matematiksel modeli

X_i ve Y_j veri tabanındaki nesnelere bir birliktelik kuralı $\{X_1, X_2, \dots, X_m\} \Rightarrow \{Y_1, Y_2, \dots, Y_n\}$ şeklinde gösterilir. Bu ifade “ X_1, X_2, \dots, X_m nesnelere bulunduğu işlemlerde sık olarak Y_1, Y_2, \dots, Y_n ’ de aynı işlemlerde yer alır” anlamına gelmektedir [48].

Tablo 3.1. Birliktelik kurallarının genel gösterimi

	Öncül ₁	Öncül ₂	...	Öncül _N	Sonuç
Kural ₁
Kural ₂
...
Kural _N

Birliktelik kurallarının genel gösterimi Tablo 3.1' de gösterildiği gibidir. Birliktelik kuralı iki yarıdan oluşur, kuralın sol tarafı öncül (antecedent) ve sağ tarafı sonuç (consequent) olarak adlandırılır.

Bu modelde $I=\{i_1, i_2, \dots, i_m\}$ veri tabanındaki nesnelere ve her i bir nesne olarak adlandırılır. D işlemler kümesi; T bir işlemde yer alan ürünleri ($T \subseteq I$), t her işleme ait tekil (unique) numarayı ifade etmektedir. X ve Y nesnelere kümesi olsun. Bir T işlemler kümesi ancak ve ancak $X \subseteq T$ ise T X ' i kapsıyor denir. Bir birliktelik kuralı $X \Rightarrow Y$ formunda ifade edilir ve X öncül Y sonuç olarak adlandırılır. Burada $X \subseteq I$, $Y \subseteq I$ ve $X \cap Y = \emptyset$ dir [49].

3.2.2. Destek (support) ve güven (confidence) değeri

Bir birliktelik kuralının öncelikle s ile ifade edilen destek değeri tanımlanır. Destek değeri, $X \Rightarrow Y$ kuralının sağ ve sol tarafının aynı anda birlikte sağlanması ihtimalidir. İkinci olarak kuralın c ile ifade edilen güven değeri tanımlanır. Güven, $X \Rightarrow Y$ kuralın sol tarafının sağlanması durumunda sağ tarafının da sağlanması olasılığıdır. Matematiksel olarak destek ve güven değerleri eşitlik 1, 2 ve 3' te gösterildiği gibidir.

$$\text{destek } (X \Rightarrow Y) = P(X \cup Y) \quad (1)$$

$$\text{güven } (X \Rightarrow Y) = P(Y|X) \text{ veya} \quad (2)$$

$$\text{güven } (X \Rightarrow Y) = \text{destek } (X \Rightarrow Y) / \text{destek } (X) \quad (3)$$

Başka bir ifade ile destek ve güven değerleri eşitlik 4, 5 ve 6' da gösterilmiştir.

$$\text{destek}(X) = |X| / |D| \quad (4)$$

$$\text{destek}(X \Rightarrow Y) = |X.Y| / |D| \quad (5)$$

$$\text{güven}(X \Rightarrow Y) = \text{destek}(X \Rightarrow Y) / \text{destek}(X) \quad (6)$$

$|X|$: İncelenen veri tabanında X nesnesini içeren işlemlerin sayısı,

$|X.Y|$: İncelenen veri tabanında X ve Y nesnelere birlikte içeren işlemlerin sayısı,

$|D|$: İncelenen veri tabanındaki bütün işlemlerin sayısıdır.

3.2.3. K-nesneküme (k-itemset)

Nesneküme veri madenciliği literatüründe itemset olarak yer almaktadır. Eğer bir küme k tane nesne içeriyorsa bu küme 'k-nesneküme' olarak ifade edilir. k-nesnekümenin sık geçen kümeleri L_k şeklinde ifade edilmektedir. Örnek olarak; {a, b, c} 3 elemanlı bir nesnekümedir ve 3-nesneküme olarak ifade edilir [10].

3.2.4. Sık nesneküme (Frequent itemset)

Belirlenen minimum destek değerini sağlayan nesneküme sık geçen nesneküme olarak tanımlanır. k-nesnekümenin sık geçen kümeleri L_k şeklinde ifade edilmektedir.

3.2.5. Minimum destek ve güven değeri

Minimum destek ve güven değerleri birliktelik kuralı madenciliğine başlamadan önce kullanıcı tarafından belirlenir. Minimum destek değeri az sıklıkta nesnekümelerini eler, minimum güven değeri ise az sıklıkta görülen (ilginç olmayan) birliktelik kurallarını eler. Kullanıcı birliktelik kuralı madenciliği sonuçlarına göre minimum destek ve güven değerini problem alanına ayarlayabilir. Bu değerler genel olarak 0-1 aralığından çok %0-%100 aralığında ifade edilmektedir [10].

3.2.6. Güçlü birliktelik kurallar

Yüksek güven ve destek değerine sahip kurallara güçlü (strong) kurallar adı verilir. Verilen bir D işlemler kümesinde birliktelik kurallarının amacı, minimum destek ve güven şartını sağlayan $X \Rightarrow Y$ kuralların bulunmasıdır. Örneğin bir X ürünü satın alan müşteriler aynı zamanda Y ürünü de satın alıyorsa, bu kural 7 numaralı eşitlikte gibi gösterilir.

$$X \Rightarrow Y \quad [\text{destek}=\%2, \text{güven}=\%60] \quad (7)$$

Eşitlik 7' deki birliktelik kuralı şu şekilde yorumlanır. İncelenen alışveriş kayıtlarında, X ile Y' nin birlikte görülme olasılığı %2; X ürünü satın alan müşterilerin %60'ı aynı alışverişte Y ürünü de satın aldığı veya X ürünü alan bir müşterinin Y ürünü alma ihtimalinin %60 olduğu söylenir [10, 49].

3.3. Birliktelik Kuralı Madenciliği

Birliktelik kuralı madenciliği genellikle iki adımda incelenir [10].

3.3.1. Sık geçen nesnekümlerin bulunması

Sık geçen nesnekümler madenciliğinin problemi veri setindeki bütün sık geçen nesnekümlerin bulunmasıdır. N tane nesne içeren bir veri setinde $2^N - 1$ tane boş küme olmayan nesneküme oluşturulur ve bunların hepsi sık geçen nesneküme olabilir. Büyük veri tabanlarında özellikle de minimum destek eşik değerinin çok düşük olduğu veri tabanlarında sık geçen nesneküme sayısı çok büyük sayılara ulaşabilir. Bunun nedeni, bir sık geçen nesnekümenin bütün alt kümelerinin de sık geçen nesneküme olmasıdır. Örnek olarak; bir sık geçen nesnekümenin büyüklüğü

100 olsun $(a_1, a_2, \dots, a_{100})$, 1-öğeli sık geçen nesnekümlerin sayısı $\binom{100}{1}$: $(a_1, a_2, a_3,$

$\dots, a_{100})$; 2 öğeli sık geçen nesnekümlerin sayısı $\binom{100}{2}$: $(a_1, a_2), (a_1, a_3), \dots, (a_{99}, a_{100})$

ve bu şekilde k öğeli nesneküme kadar devam edildiğinde toplam sık geçen nesnekümlerin sayısı eşitlik 8' deki gibidir.

$$\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1 \approx 1,27 \times 10^{30} \quad (8)$$

Bu sayı herhangi bir bilgisayarın hesaplaması veya hafızada depolaması için çok yüksektir. Bu zorluğu aşabilmek için “kapalı sık nesneküme” ve “maksimal sık nesneküme” kavramları ortaya çıkarılmıştır. Bu kavramlar devam eden bölümde detaylı olarak açıklanmıştır.

3.3.2. Güçlü birliktelik kurallarının oluşturulması

Kullanıcı tarafından belirlenmiş olan minimum destek değerine göre sık geçen nesneküme oluşturulduktan sonra birliktelik kuralları oluşturulur. Birliktelik kurallarını oluşturmanın zorluğu sık geçen nesneküme sayısının ve büyüklüğünden kaynaklanmaktadır. D tane nesne içeren bir veri setinde mümkün olan birliktelik kurallarının toplam sayısı eşitlik 9’ da verildiği gibidir.

$$3^d - 2^{d+1} + 1 \quad (9)$$

Bu üssel karmaşıklığa rağmen gerçek hayatta sık geçen nesneküme oluşturmak daha maliyetlidir. Çünkü nesneküme sayısının destek değerlerinin hesaplanabilmesi için her bir işlemin üstünden geçmeye ve verileri doğrulamaya ihtiyaç duyulur.

Birliktelik kuralları oluşturmada kullanılan basit bir algoritma şu şekildedir: Her bir sık geçen nesneküme L ve boş olmayan alt kümeleri L_a için minimum güven değerinden büyük ve eşit olan $L_a \Rightarrow L - L_a$ şeklindeki mümkün kurallar oluşturulur. Tanımlamaya göre oluşturulan bütün kurallar minimum destek ve güven değerine uymalıdır [50].

Birliktelik kuralları oluşturmada kullanılan algoritmaların birçoğu aşağıdaki konular üzerinde durur [51].

- Aday nesneküme setini minimize etme
- Veri tabanını en az zamanda tarayarak, aday nesneküme sayısının destek değerlerini hesaplama.

Bu bağlamda birliktelik kuralları algoritmaları genellikle aşağıdaki konular üzerinde farklılık gösterir:

- Adayların oluşturulması
- Bir aday nesnekümenin destek değerinin hesaplanması
- Veri tabanını tarama sayısı
- Kullanılan veri yapıları

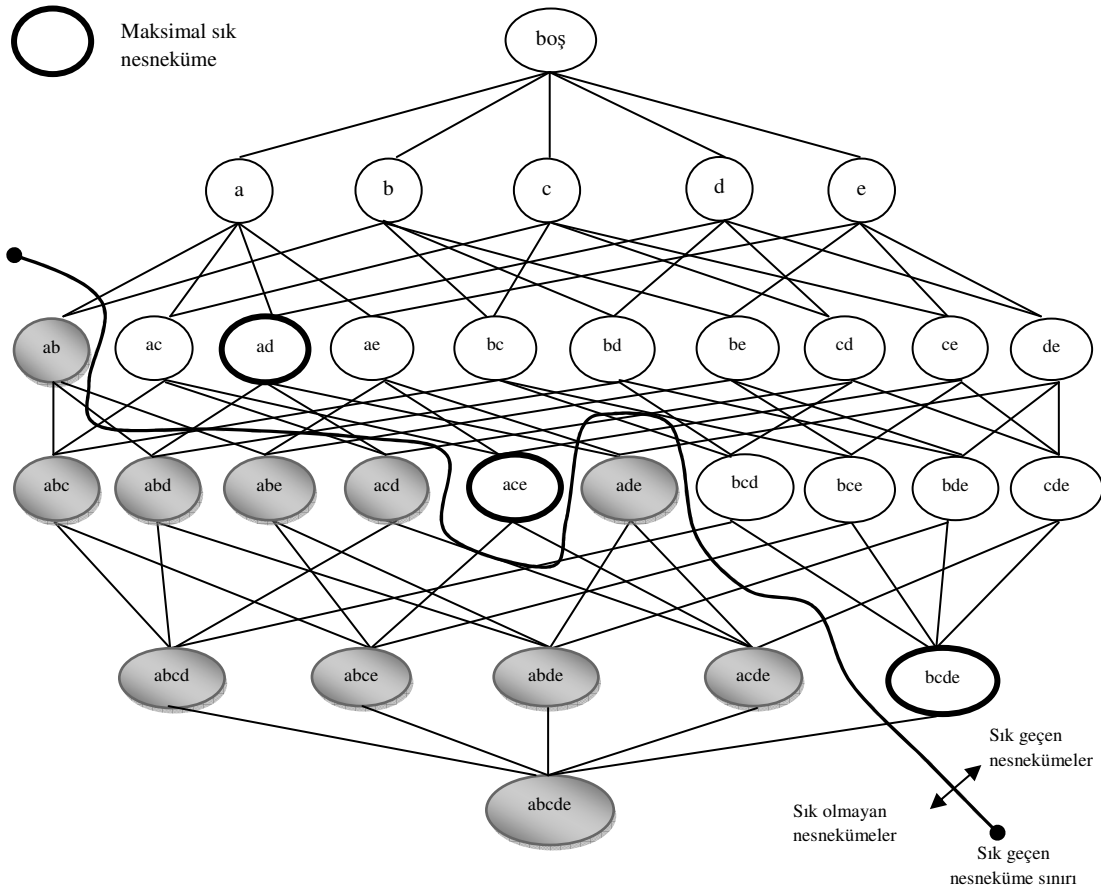
3.4. Sık Geçen Nesnekümenin Temsilleri

Pratikte, bir işlemsel veri setinden çok büyük sayılarda sık geçen nesneküme üretilebilir. Bu yüzden diğer bütün sık geçen nesneküme elde edilen küçük bir örnek grubu tanımlamak elverişli olmaktadır. Bu bağlamda devam eden bölümde maksimal ve kapalı sık geçen nesneküme anlatılmıştır.

3.4.1. Maksimal sık nesneküme

Maksimal sık nesneküme, X sık geçen bir nesneküme olsun, sık geçen X ' in başka bir sık geçen üst kümesi oluşturulamıyorsa bu küme "maksimal sık nesneküme" olarak ifade edilir [10].

Bu kavramı örneklemek için Şekil 3.1' de gösterilen kafes dikkate alınır. Kafesteki nesneküme, sık olan nesneküme ve sık olmayan nesneküme olmak üzere ikiye ayrılır. Sık olan nesneküme sık geçen nesneküme sınırının üzerinde, sık olmayan nesneküme çizginin altında toplanmıştır. Çizgiye yakın olan $\{a, d\}$, $\{a, c, e\}$ ve $\{b, c, d, e\}$ nesnekümenin maksimal sık nesneküme olduğu düşünülür. Çünkü bu kümelerin üst kümeleri sık olmayan nesnekümedir. Örnek olarak $\{a, d\}$ maksimal sık nesnekümedir çünkü bu kümenin üst kümeleri olan $\{a, b, d\}$, $\{a, c, d\}$ ve $\{a, d, e\}$ sık olmayan nesnekümedir. Buna karşılık $\{a, c\}$ maksimal sık olmayan nesnekümedir çünkü $\{a, c\}$ ' nın üst kümelerinden olan $\{a, c, e\}$ sık geçen bir nesnekümedir [52].



Şekil 3.1. Maksimal sık nesneküme

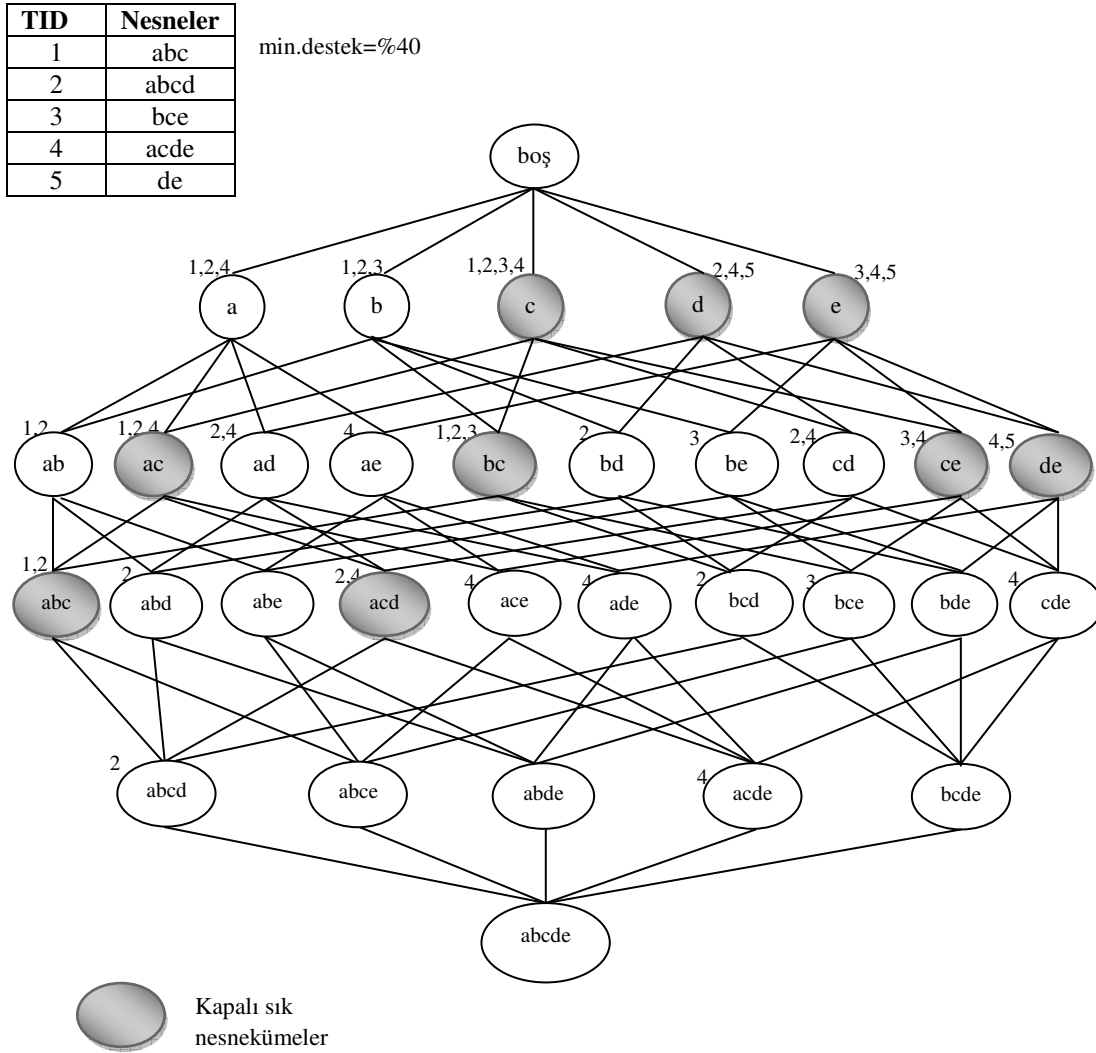
Maksimal sık nesneküme, sık geçen nesneküme çok olduğu veri setlerinde pahalı bir gösterim sunar. Yine de bu yaklaşım maksimal sık nesneküme alt kümelerini birer birer saymadan bulan algoritmalar için elverişlidir.

Maksimal sık nesneküme, alt kümelerinin destek değerleri hakkında herhangi bir bilgi içermezler. Örnek olarak; $\{a, c, e\}$, $\{a, d\}$ ve $\{b, c, d, e\}$ maksimal sık nesneküme'dir ve alt kümelerinin destek değerleri hakkında hiçbir bilgi içermezler.

3.4.2. Kapalı sık nesneküme

Bir S veri setinde, bir kümenin kendisiyle aynı destek sayısına sahip bir üst kümesi mevcut değilse, bu küme kapalı nesneküme olarak ifade edilir. Bu tanımın tersi

olarakta, eğer bir kümenin en yakın üst kümesi bu kümeyle aynı destek sayısına sahip ise bu küme kapalı nesneküme değildir [52].



Şekil 3.2. Kapalı sık nesnekümelere

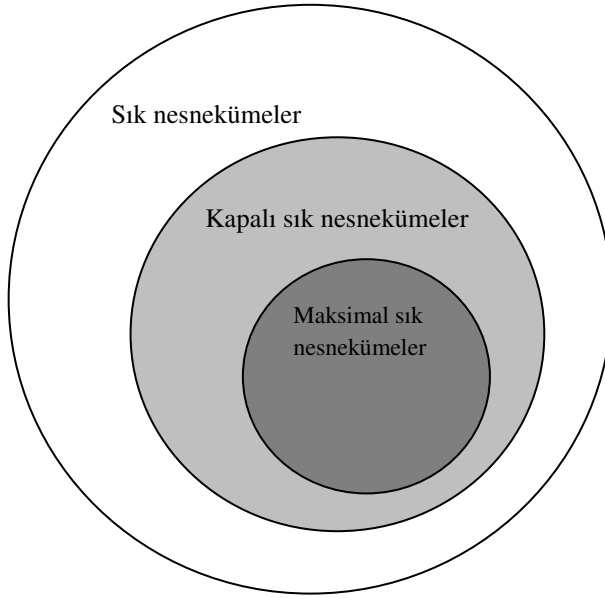
Kapalı nesnekümelere örnek Şekil 3.2' de gösterilmiştir. Nesnekümelere destek sayılarının daha iyi anlaşılması için nesnelere hangi işlemlerde geçtiği de gösterilmiştir. Örneğin; {b, c} düğümü 1, 2, 3 işlemleriyle ilişkilidir, destek sayısı 3' e eşittir. Şekil 3.2' de verilen işlemlere dikkat etmek gerekirse b nesnesini içeren her işlem aynı zamanda c nesnesini de içermektedir. Bu nedenle {b}' nin destek değeri {b, c} ile aynıdır ve {b} kapalı bir küme olarak ifade edilemez. Ancak {b, c} kapalı bir kümedir çünkü {b, c} kümesinin hiçbir üst kümesi aynı destek sayısına sahip değildir. Kapalı sık nesneküme ise destek değeri minimum destek değerinden büyük

ve eşit olan kapalı nesnekümelere denir. Şekil 3.2' de verilen örnekte destek eşik değeri %40 olarak belirlenmiştir. Buna göre {b, c} kapalı sık bir nesnekümedir çünkü destek değeri %60' a eşittir. Diğer kapalı sık nesnekümelere şekilde koyu renkle işaretlenmiştir.

Birliktelik kuralları madenciliği probleminin, sık geçen nesnekümelere bulunması ve birliktelik kurallarının elde edilmesi olduğu geçmiş bölümlerde de ifade edilmiştir. Kapalı sık nesnekümelere madenciliği ile bütün sık geçen nesnekümelere madencilemeye gerek olmadığı ve bütün muhtemel birliktelik kurallarında oluşturulmasına gerek kalmaz. Zaten bu problemi ortadan kaldıran etkin bir algoritma sunmuştur [53].

Kapalı sık nesnekümelere madenciliği gereksiz kuralların atılmasını sağlar. Örneğin, $X \subset X'$ ve $Y \subset Y'$ olsun. $X' \Rightarrow Y'$ kuralının var olduğu durumda $X \Rightarrow Y$ kuralının oluşturulması gereksizdir çünkü iki kuralında destek ve güven değeri aynıdır. Yukarıdaki örneği dikkate alarak, {b} kapalı bir sık olmayan nesneküme iken {b, c} kapalı bir sık nesnekümedir. {b} \Rightarrow {d, e} kuralı gerekli olmayan bir kuraldır çünkü bu kural {b, c} \Rightarrow {d, e} kuralı ile aynı destek ve güven değerine sahiptir. Bu örnekte de olduğu gibi birliktelik kurallarının oluşturulmasında kapalı sık nesnekümelere kullanarak gereksiz kuralların elde edilmesi engellenir [52].

Bir veri setinde belli bir minimum destek değerine göre C kapalı sık nesneküme ve M kümesi maksimal sık nesneküme olsun. Açıkça görülür ki M, C' nin alt kümesi ve C de sık geçen nesnekümelere alt kümesidir. Şekil 3.3' te sık geçen nesneküme, kapalı sık nesnekümelere, maksimal sık nesnekümelere arasındaki ilişki gösterilmiştir [54].



Şekil 3.3. Sık geçen nesnekümelerin temsilleri arasındaki ilişki [52]

3.5. Sık Geçen Nesnekümleri Oluşturma Metotları Alternatifleri

Sık geçen nesnekümleri oluşturma algoritmaları tarama işlemini nasıl yaptığına ve tarama işlemine hangi noktadan başladığına göre ikiye ayrılmaktadır [52]. Algoritmalar kafesi tarama işlemini nasıl yaptığına göre sınıflandırılırsa;

1. Genişlik öncelikli (Breadth first search) arama algoritmaları; kafesi adım adım inceler.
2. Derinlik öncelikli (Depth first search) arama algoritmaları; kafesi bir düğümden gelecek adımdaki bir düğüme ya da mevcut adımdaki gelecek bir düğüme geçerek inceler.

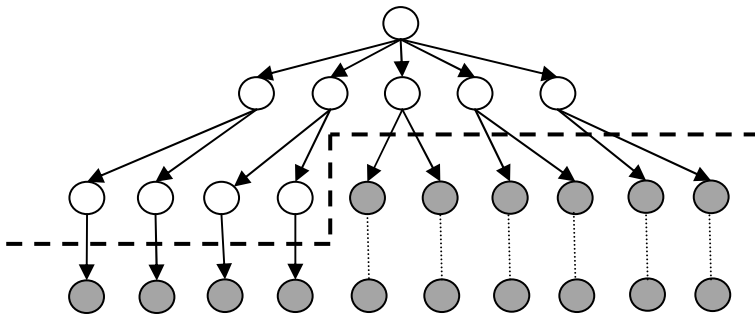
Diğer bir sınıflandırma ise kafesi tarama işlemine hangi noktadan başlanması üzerinde duran algoritmalara göre yapılır;

1. Özelden-genele (Specific to general) arama algoritmaları; kafesi en alt noktasından taramaya başlar.

2. Genelden-özele (General to specific) arama algoritmaları; kafesi en üst noktasından taramaya başlar.

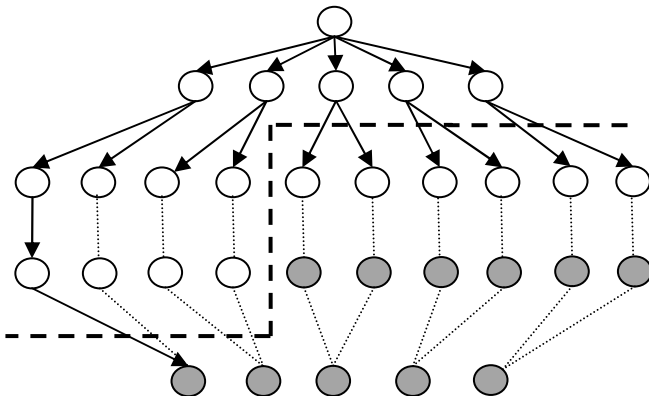
3.5.1. Genişlik ve derinlik öncelikli algoritmalar

BFS ilk olarak sık geçen 1 ögeli nesnekümleri bulur, sonra 2 ögeli nesnekümleri ve bu şekilde oluşturulacak yeni sık geçen nesneküme kalmayana kadar devam eder. Şekil 3.4' te gösterilen BFS yaklaşımına Apriori algoritması tipik bir örnektir.



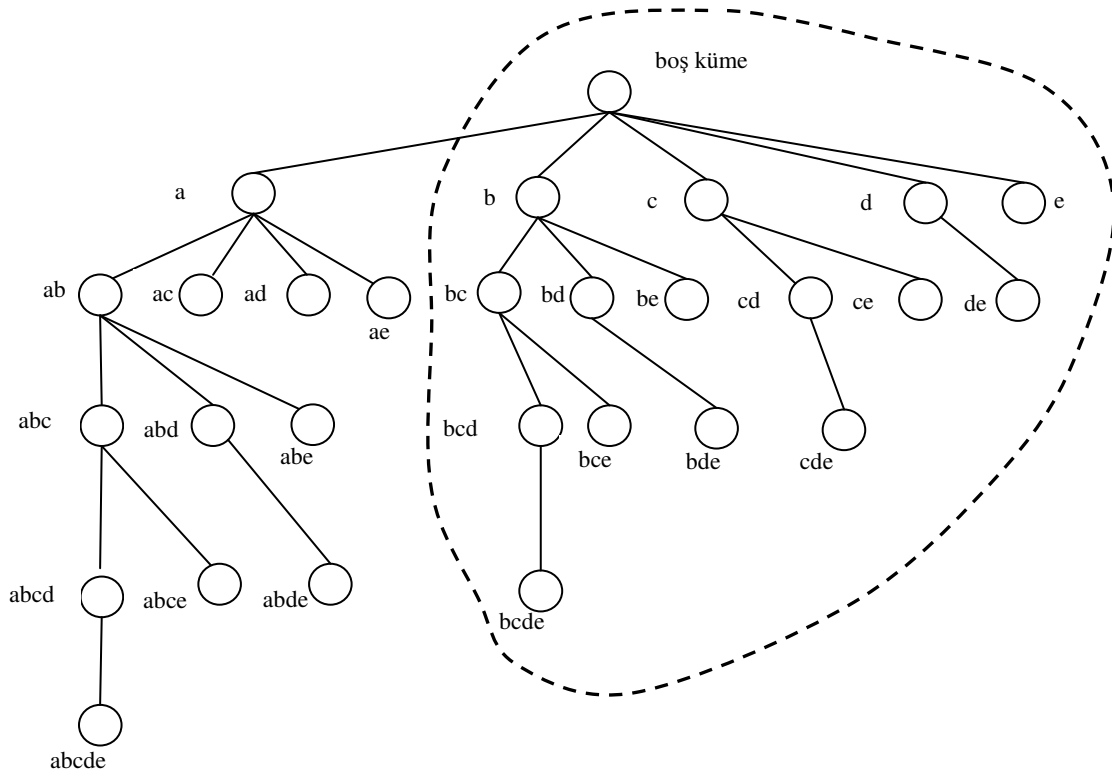
Şekil 3.4. BFS yaklaşımı

DFS yaklaşımı maksimal sık nesnekümleri bulmak için tasarlanmış algoritmalar tarafından sıkça kullanılır. Şekil 3.5' te gösterilen bu yaklaşım sık geçen nesneküme sınırını BFS yaklaşımından daha hızlı saptar. Bir maksimal sık nesneküme bulunduğu zaman, bu kümenin alt kümelerine gerçek budama işlemi gerçekleştirilebilir.



Şekil 3.5. DFS yaklaşımı

Örnek olarak Şekil 3.6' daki bcde düğümü maksimal sık nesneküme ise algoritma bu kümenin bd, bc, c, d, e gibi alt kümelerine uğramak zorunda değildir. Çünkü bu nesnelere herhangi bir maksimal sık nesneküme içermeyecektir. FP-Growth algoritması DFS için tipik bir örnektir. DFS algoritmasına Şekil 3.6 örnek olarak gösterilebilir. DFS a düğümünden başlar, a' nın sık geçen nesneküme olup olmadığını tanımlamak için destek değerini hesaplar. Eğer bu nesne sık geçen nesneküme ise algoritma ab, abc düğümlerinden geçerek sık olmayan bir düğüme (abcd gibi) ulaşıncaya kadar bir sonraki adımdaki düğümlere ilerler. Daha sonra geldiği yoldan geri dönerek başka bir dala geçer (örnek olarak abce) ve arama bu noktadan devam eder.

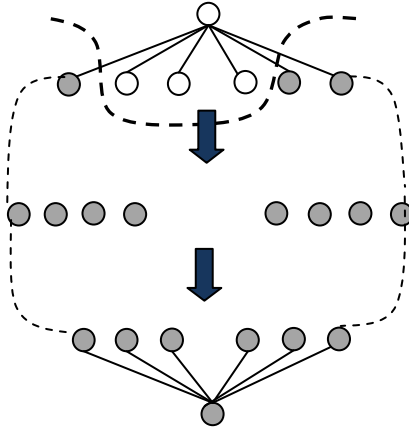


Şekil 3.6. DFS yaklaşımını kullanarak nesneküme adayları oluşturma

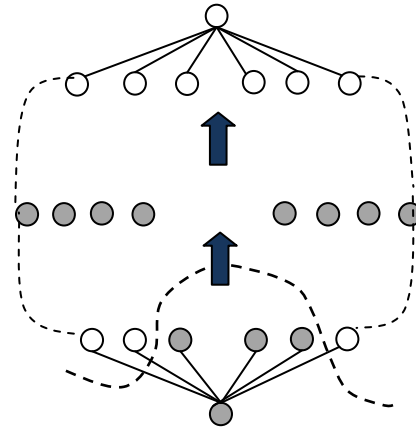
3.5.2. Genelden-özele ve özelden-genele yaklaşımlı algoritmalar

Genelden-özele yaklaşımında $(k-1)$ ögeli sık geçen nesnekümler k ögeli nesnekümler elde etmek için birleştirilirler. Genelden-özele yaklaşımı bir sık nesnekümenin uzunluğunun çok büyük olmadığı durumlarda etkindir.

Özelden-genele yaklaşımı, genel olan sık geçen nesnekümleri bulmadan önce ilk olarak daha spesifik sık geçen nesnekümleri arar. Bu yaklaşım, maksimal sık nesnekümlerini yoğun işlemler arasında bulmak için sık geçen nesneküme sınırının alt noktalara daha yakın olan kafeslerde etkindir [52].



Şekil 3.7.a. Genelden-özele yaklaşımı



Şekil 3.7.b. Özelden-genele yaklaşımı

Sık geçen nesnekümler, kafesin ne alt noktasına ne de üst noktasına yakın olmadığı durumlarda bu iki yaklaşımda etkin değildir. Bu sebeple hibrid bir algoritma önerilmiştir. Bu algoritma iki yönlü arama yapar ve “iki yönlü arama” yaklaşımı olarak adlandırılmıştır. İki yönlü arama algoritmasında tek yönlü arama algoritmalarına göre daha az aday oluşturulur ve daha az geçişler yapılır [55].

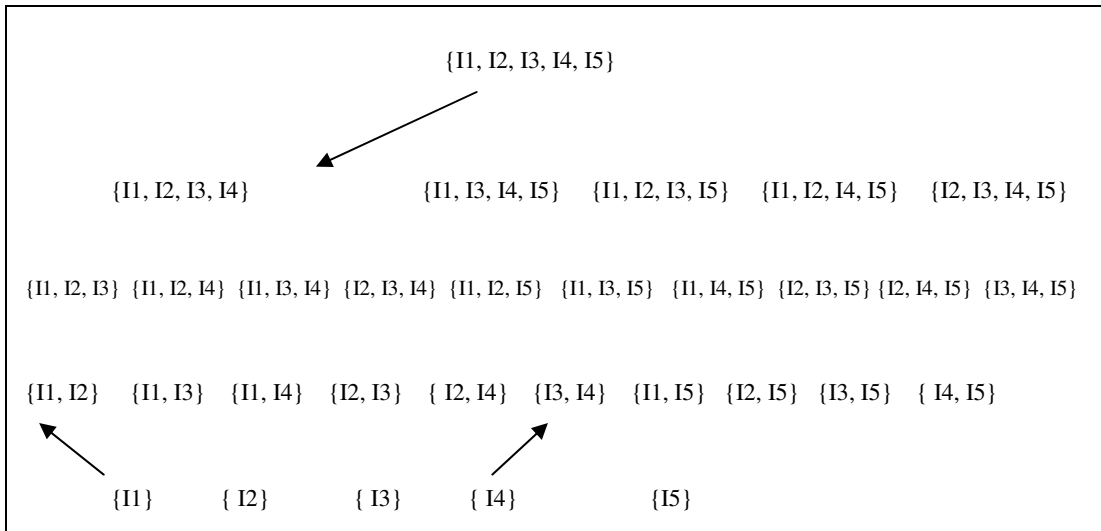
İki yönlü arama algoritması iki özelliğe göre adayları eler. Birinci özelliği, bir nesneküme sık olmayan bir nesneküme ise bu kümenin bütün alt kümeleri de sık olmayan nesnekümlerdir; ikinci özelliği ise bir nesneküme sık geçen bir nesneküme ise bu kümenin bütün alt kümeleri de sık geçen nesnekümedir. İki yönlü arama algoritmasının işleyişi Tablo 3.2’deki örnek veri seti dikkate alınarak aşağıda anlatıldığı gibidir.

İlk geçişte bütün 1 ögeli nesnekümler ve 5-nesneküme $\{I1, I2, I3, I4, I5\}$ aday olarak oluşturulur. Destek değeri hesaplama işleminden sonra sık olmayan $\{I5\}$ nesnekümesi bulunur. Özellik 1’ e göre bu kümenin bütün üst kümeleri elenir. İkinci

adımdaki adaylar $\{I1, I2\}$, $\{I1, I3\}$, $\{I1, I4\}$, $\{I2, I3\}$, $\{I2, I4\}$, $\{I3, I4\}$ ve $\{I1, I2, I3, I4\}$ nesnekümeleridir. İkinci destek değeri hesaplama işleminden sonra $\{I1, I2, I3, I4\}$ nesnekümesi sık olan nesneküme olarak bulunur. Özellik 2' ye göre bu kümenin bütün alt kümeleri sık geçen olacağından alt kümeleri incelemeye gerek yoktur. Oluşturulacak başka aday kalmadığından algoritma bu şekilde sonlandırılır.

Tablo 3.2. Örnek veri tabanı parçası

TID	Nesneler
1	I1, I2, I3, I4, I5
2	I1, I3
3	I1, I2
4	I1, I2, I3, I4



Şekil 3.8. İki yönlü arama

3.6. Sık Geçen Nesneküme Madenciliği

3.6.1. Apriori algoritması

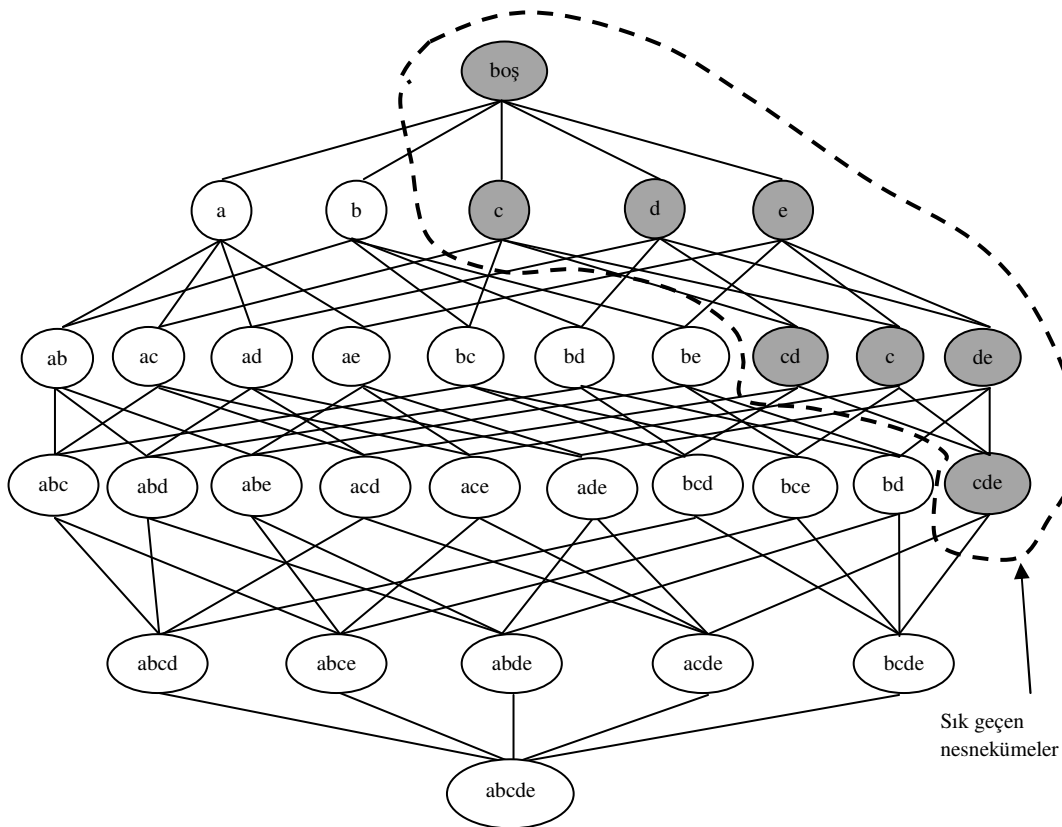
Sık geçen nesneküme madenciliğinin hesap karmaşıklığını azaltmanın birçok yolu bulunmaktadır. Ancak bu alanda en bilinen yöntem Apriori algoritmasıdır. Apriori

algoritması bazı aday nesnekümlerin destek değerlerini saymadan bu adayların elenmesi ile etkili bir çözüm sunar. Algoritmanın ismi, sık geçen nesnekümlerin madenlenmesinde önsel (prior) bilgiyi kullanmasını temel almıştır [49].

3.6.1.1. Apriori özelliği

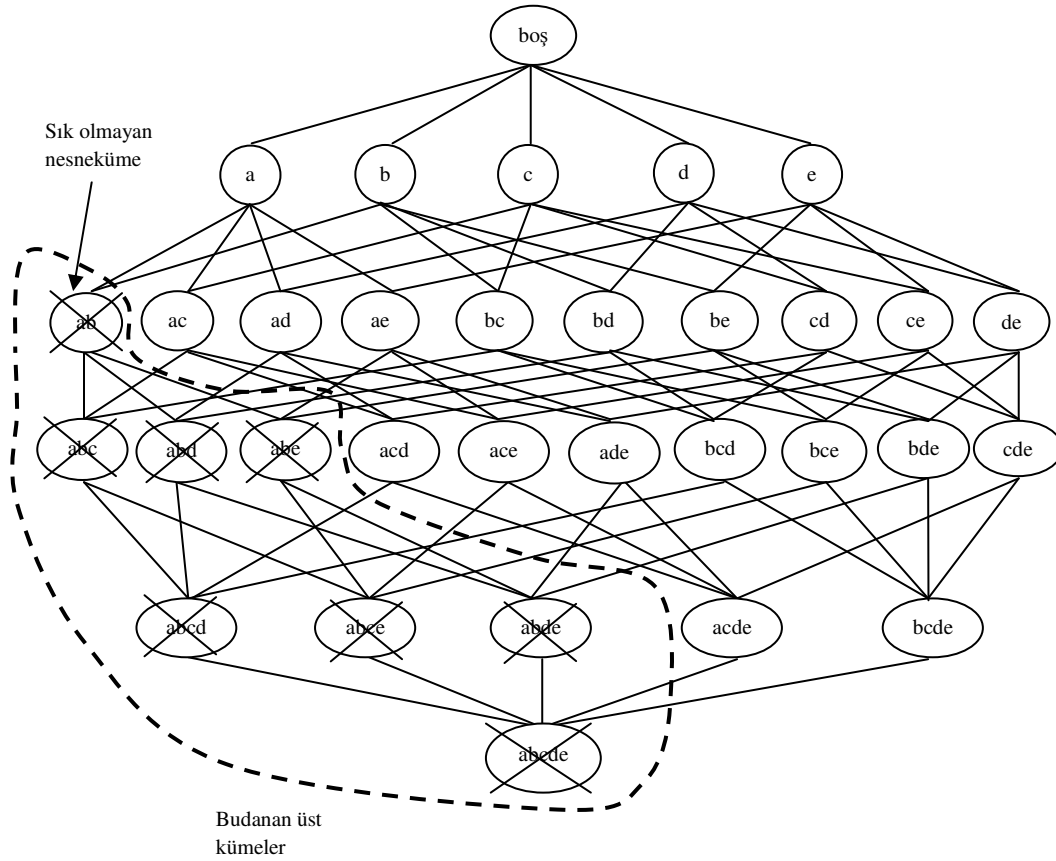
Eğer bir nesneküme sık geçen ise bu kümenin bütün alt kümeleri de sık geçen nesneküme olmalıdır.

Apriori özelliği, Şekil 3.9' daki kafes dikkate alınarak örnekle şu şekilde açıklanır: $\{c, d, e\}$ kümesinin bir sık geçen nesneküme olduğu varsayılır. Açıkça görülür ki $\{c, d, e\}$ kümesini içeren her işlemin bu kümenin alt kümeleri olan $\{c, d\}$, $\{c, e\}$, $\{c\}$, $\{d\}$ ve $\{e\}$ kümelerini de içerir. Sonuç olarak da, $\{c, d, e\}$ sık geçen nesneküme ise bütün alt kümeleri de sık geçen nesneküme olmalıdır.



Şekil 3.9. Apriori algoritmasının birleştirme özelliği

Bunun tersi olarak da, sık olmayan bir nesnekümenin (örneğin {a, b}) bütün üst kümeleri de sık olmayan nesnekümelere. Şekil 3.10’ da gösterildiği gibi {a, b} kümesinin sık olmayan küme olarak bulunduğu anda {a, b} kümesini içeren bütün üst kümeler hemen budanabilir [52].



Şekil 3.10. Apriori algoritmasının budama özelliği

3.6.1.2. Apriori algoritmasının işleyişi

Apriori algoritması seviye mantığı (level-wise) arama olarak bilinen yinemeli bir yaklaşım kullanır. Bu yaklaşımda k öğeli nesnekümelere $(k-1)$ öğeli nesnekümelere birleştirilmesiyle oluşturulur. İlk olarak 1 öğeli nesnekümelere oluşturmak için nesnelere destek değerleri hesaplanır, $(k-1)$ inci geçişte bulunan sık öğeler (L_{k-1}), k geçişte sık geçen nesne aday kümesini (C_k) oluşturmak için kullanılır. Veri tabanı tarama ve C_k ’ ların destek değerleri hesaplanır ve L_k ’ lar oluşturulur. Bu şekilde algoritma sık geçen nesneküme bulamayınca kadar devam eder [49].

Tablo 3.3. Notasyonlar

k-nesneküme	k nesne içeren bir nesneküme
L_k	Sık geçen k öğeli nesnekümeler (minimum destek değerine göre) Bu kümenin her üyesi iki alana sahiptir: i)nesneküme ii)destek hesabı
C_k	k öğeli aday nesnekümeler (potansiyel sık geçen nesnekümeler) Bu kümenin her üyesi iki alana sahiptir: i)nesneküme ii)destek hesabı

L_{k-1} ' in L_k 'nin elde edilmesinde nasıl kullanıldığını anlamak için algoritmanın Birleştirme (Join) ve Budama (Prune) işlemlerini kapsayan adımlar aşağıda anlatıldığı gibidir [10].

1. Birleştirme Adımı: L_k ' yı bulmak için k öğeli aday nesnekümeler (C_k), L_{k-1} 'in kendi arasında birleştirilmesiyle oluşturulur. l_1 ve l_2 , L_{k-1} ' de bulunan nesnekümeler olsun. $l_i[j]$ gösterimi, l_i ' deki j. nesneyi temsil eder (örneğin, $l_1[k-2]$, l_1 ' deki sondan 2.nesneyi temsil eder). Apriori, bir işlemdeki nesnelere veya nesnekümelerin alfabetik sıraya göre sıralandığını varsayar. k-1 öğeli nesnekümeler için nesnelere $l_i[1] < l_i[2] < \dots < l_i[k-1]$ şeklinde sıralanmıştır. L_{k-1} ' in ilk (k-2) nesnelere ortak ise $L_{k-1} \cap L_{k-1}$ birleşmesi gerçekleştirilebilir. Yani, L_{k-1} ' in l_1 ve l_2 nesnekümleri $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] = l_2[k-1])$ şeklinde ise birleştirilir. $l_1[k-1] < l_2[k-1]$ şartlı aynı adayların oluşturulmamasını sağlar. l_1 ve l_2 ' nin birleştirilmesiyle oluşan nesnekümenin sonuçları $l_1[1], l_1[2], \dots, l_1[k-2], l_1[k-1], l_2[k-2]$.
2. Budama Adımı: C_k ' nin elemanları sık geçen olabilir veya olmayabilir fakat bütün sık geçen nesnekümleri içerir ve L_k ' nin üst kümesidir. C_k ' daki her bir adayın destek değerinin hesaplanması için yapılan bir taramanın sonuçları L_k ' yı belirler (örneğin, minimum destek değerinden büyük ve eşit değere sahip bütün adaylar sık geçen nesneküme olarak tanımlanır ve L_k ' da yer alır). C_k çok büyük

boyutta olabilir ve çok ağır hesaplamalar gerektirebilir. C_k 'nin boyutunu azaltmak için Apriori özelliği kullanılır. Herhangi bir sık olmayan (k-1)-nesneküme sık geçen k-nesnekümenin alt kümesi olamaz. Bundan dolayı, aday k-nesnekümenin büyük (k-1) ögeli alt kümeleri L_{k-1} 'de yoksa bu aday sık geçen nesneküme olamaz ve C_k 'dan çıkarılır.

Tablo. 3.4. Apriori algoritmasının kodları

Algoritma: Apriori. Sık geçen nesneküme tekrarlı seviye mantığı yaklaşımı kullanarak aday oluşturma

Girdi: Bir D veri tabanı, minimum destek eşik değeri

Output: L, sık nesneküme

Metod:

- 1) $L_1 = \text{bul} \{ \text{large 1-nesneküme} \};$
- 2) **for** { $k=2; L_{k-1} \neq \emptyset; k++$ } **do begin**
- 3) $C_k = \text{apriori-gen} (L_{k-1});$ // yeni adaylar
- 4) **forall** işlemler $t \in D$ **do begin** // D destek sayıları için taranır
- 5) $C_t = \text{alt küme} (C_k, t);$ // Aday olan t'lerin alt kümeleri elde edilir
- 6) **forall** adaylar $c \in C_t$ **do**
- 7) $c.\text{sayısı} ++;$
- 8) **end**
- 9) $L_k = \{ c \in C_k \mid c.\text{sayısı} \geq \text{min_sup} \}$
- 10) **end**
- 11) **return** = $\cup_k L_k;$

Apriori-gen prosedürü (L_{k-1} : sık geçen (k-1)-nesneküme)

- 1) **forall** nesneküme $l_1 \in L_{k-1}$
- 2) **forall** nesneküme $l_2 \in L_{k-1}$
- 3) **if** ($l_1[1] = l_2[2] \wedge l_1[2] = l_2[2] \wedge \dots \wedge l_1[k-2] = l_2[k-2] \wedge l_1[k-1] < l_2[k-1]$) **then** {
- 4) $c = l_1 \cup l_2;$ // birleştirme adımı: aday oluşturma
- 5) **if** sık olmayan alt kümeler(c, L_{k-1}) varsa **then**
- 6) **sil** c; // budama adımı: gereksiz adaylar çıkarılır
- 7) **else** c'yi C_k 'ya ekle;
- 8) }
- 9) **return** $C_k;$

Sık olmayan alt kümeler prosedürü (c: aday k-nesneküme;

L_{k-1} : sık geçen (k-1)-nesneküme); // prior bilgisini kullan

- 1) **forall** (k-1)-alt küme c'nin s' i
- 2) **if** $s \notin L_{k-1}$ **then**
- 3) **return** TRUE;
- 4) **return** FALSE;

Apriori algoritması ile sık geçen nesnekümelerin elde edilmesi yukarıdaki maddelerde anlatılmıştır. Tablo 3.4' te kodları verilen birleştirme ve budama adımları aşağıdaki Tablo 3.5' te verilen örnek veri tabanına göre maddeler halinde anlatılmış ve Şekil 3.11' de gösterilerek özetlenmiştir.

Tablo 3.5. İşlemsel veriler [52]

TID	Nesneler
1	Ekmek, Süt
2	Ekmek, Çocuk bezi, Bira, Yumurta
3	Süt, Çocuk bezi, Bira, Kola
4	Ekmek, Süt, Çocuk bezi, Bira
5	Ekmek, Süt, Çocuk bezi, Kola

Tablo 3.5' te bir D veri tabanına ait işlemler verilmiştir. Bu veri tabanında 5 işlem bulunmaktadır, $|D|=5$ şeklinde gösterilir.

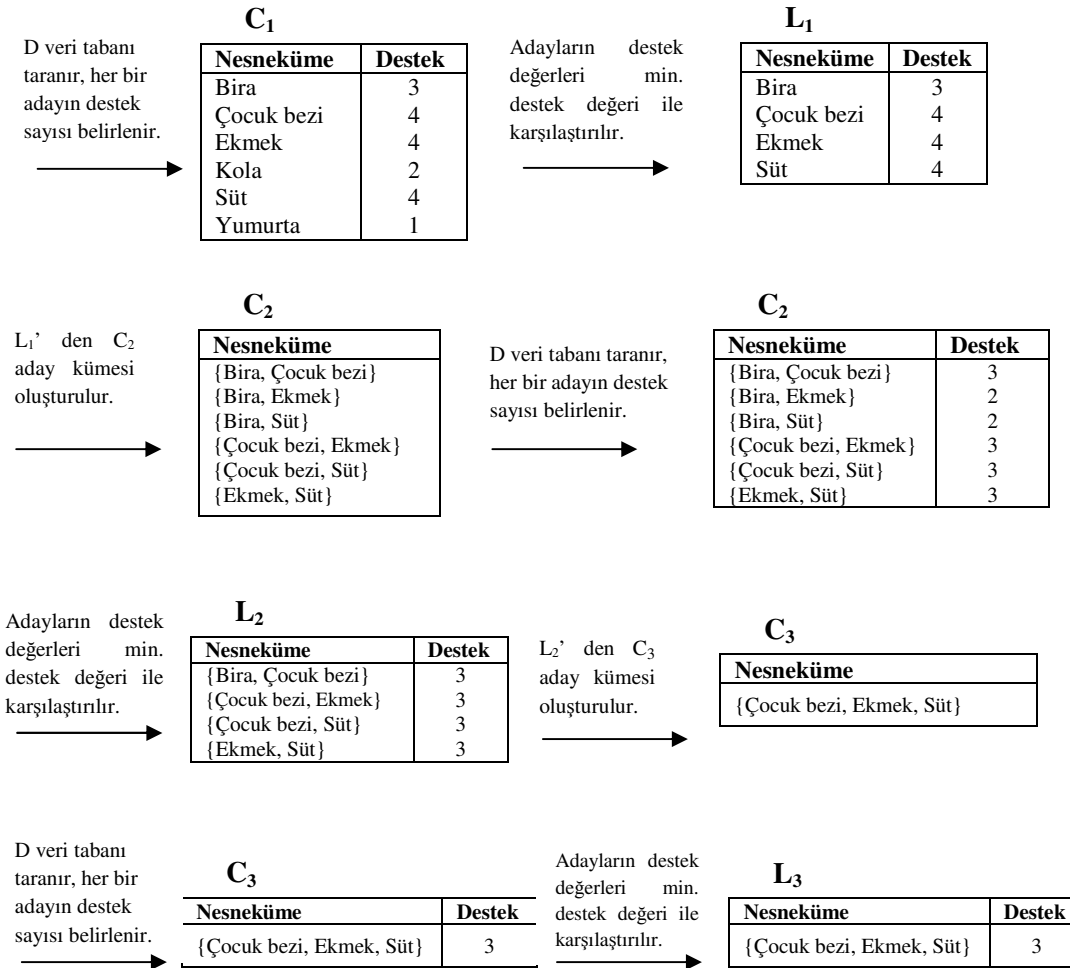
- 1) Algoritmanın ilk iterasyonunda her bir 1-nesnekümeler C_1 aday kümesinin elemanıdır. Algoritma, her ürünün destek değerini hesaplamak için bütün işlemleri basitçe tarar.
- 2) Destek eşik değeri 3 olarak belirlenmiştir ($3/5=60\%$). 1 öğeli nesnekümeler ile sık geçen L_1 kümesi oluşturulur. Bu örneğe göre C_1 kümesindeki 4 adayın destek değerleri minimum destek değerinin üzerindedir.
- 3) 2 öğeli sık geçen nesnekümelere (L_2) algoritma $L_1 \times L_1$ birleşmesini kullanır ve 2 öğeli nesnekümelere (C_2) oluşturur. C_2 , L_1 ' in ikili kombinasyonlarından oluşur.
- 4) D veri tabanındaki işlemler taranır ve C_2 kümesindeki nesnelerin destek değerleri belirlenir.
- 5) C_2 kümesindeki minimum destek değerine sahip olan 2 öğeli nesnekümelere L_2 ' yi oluşturur.

- 6) 3 ögeli adaylar kümesini (C_3) oluşturmak için $L_2 \times L_2$ birleşimi kullanılır. Detayları aşağıda anlatılan 3 ögeli adaylar kümesi $C_3 = L_2 \times L_2 = \{\{\text{Çocuk bezi, Ekmek, Süt}\}\}$ şeklinde elde edilir.
- a) Birleşme: $C_3 = L_2 \times L_2 = \{\{\text{Bira, Çocuk bezi}\}, \{\text{Çocuk bezi, Ekmek}\}, \{\text{Çocuk bezi, Süt}\}, \{\text{Ekmek, Süt}\}\} \times \{\{\text{Bira, Çocuk bezi}\}, \{\text{Çocuk bezi, Ekmek}\}, \{\text{Çocuk bezi, Süt}\}, \{\text{Ekmek, Süt}\}\} = \{\{\text{Bira, Çocuk bezi, Ekmek}\}, \{\text{Bira, Çocuk bezi, Süt}\}, \{\text{Çocuk bezi, Ekmek, Süt}\}\}$.
- b) Budama, Apriori özelliğini kullanarak:
- $\{\text{Bira, Çocuk bezi, Ekmek}\}$ ' in 2 ögeli alt kümeleri $\{\text{Bira, Çocuk bezi}\}$, $\{\text{Bira, Ekmek}\}$ ve $\{\text{Çocuk bezi, Ekmek}\}$ ' tür. $\{\text{Bira, Ekmek}\}$ L_2 ' nin üyesi olmadığından sık geçen nesneküme değildir. Bu yüzden C_3 ' ten çıkarılır.
 - $\{\text{Bira, Çocuk bezi, Süt}\}$ ' in 2 ögeli alt kümeleri $\{\text{Bira, Çocuk bezi}\}$, $\{\text{Bira, Süt}\}$ ve $\{\text{Çocuk bezi, Süt}\}$ ' tür. $\{\text{Bira, Süt}\}$ L_2 ' nin üyesi olmadığından sık geçen nesneküme değildir. Bu yüzden C_3 ' ten çıkarılır.
 - $\{\text{Çocuk bezi, Ekmek, Süt}\}$ ' in 2 ögeli alt kümeleri $\{\text{Çocuk bezi, Ekmek}\}$, $\{\text{Çocuk bezi, Süt}\}$ ve $\{\text{Ekmek, Süt}\}$ ' tür. Bütün 2 ögeli nesnekümler L_2 ' nin üyesidir ve C_3 ' te saklanır.
- 7) D veri tabanı taranır ve C_3 kümesindeki nesnelere destek değerleri belirlenir.
- 8) 4 ögeli adaylar kümesini (C_4) oluşturmak için $L_3 \times L_3$ birleşimi kullanılır. L_3 kümesinde tek bir nesneküme olduğundan dolayı C_4 boş kümedir ve algoritma sık geçen bütün nesnekümleri bulduğundan sonlandırılır.

Apriori algoritmasının budama stratejisinin etkinliği basit bir hesapla görülebilir. Yukarıdaki örnekte; normalde 1 ögeliden 3 ögeli nesneye kadar oluşturulabilecek toplam aday sayısı eşitlik 10' da, Apriori algoritması ile oluşturulan toplam aday sayısı eşitlik 11' de verilmiştir. Sonuç olarak, Apriori algoritması ile oluşturulan aday sayısında %61 azalma olduğu görülmektedir [10].

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 6 + 15 + 20 = 41 \quad (10)$$

$$\binom{6}{1} + \binom{4}{2} + \binom{4}{3} = 6 + 6 + 4 = 16 \quad (11)$$



Şekil 3.11. Apriori algoritmasının adımları

3.6.2. AprioriTid algoritması

AprioriTid algoritması geçiş işlemleri başlamadan önce aday nesnekümelere tanımlamak için apriori-gen fonksiyonunu kullanır. Bu algoritmanın ilginç olan özelliği; veri tabanı ilk geçişten sonra destek değeri hesaplamaları için kullanılmamasıdır. İlk geçişten sonra veri tabanı yerine \hat{C}_1 kümesi kullanılarak sık

geçen nesnekümler bulunmaktadır. AprioriTid algoritmasının işleyişi Şekil 3.12' de gösterilen veri tabanı dikkate alınarak aşağıda anlatıldığı gibidir [49].

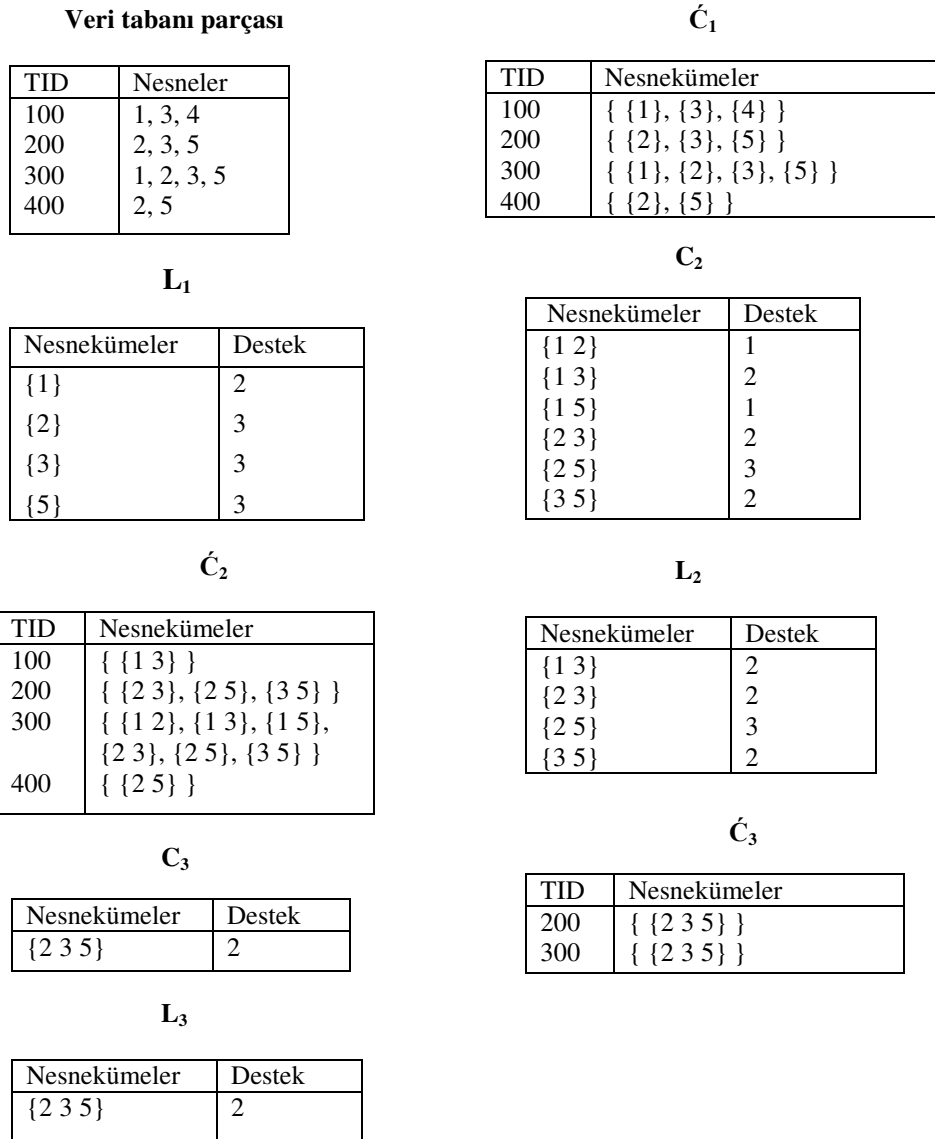
Veri tabanındaki alışveriş kayıtlarına göre minimum destek değeri 2 olarak belirlenmiştir. İlk olarak tüm veri tabanı taranır ve \hat{C}_1 nesnekümleri elde edilir. \hat{C}_1 ' deki kayıtlar sayılarak L_1 hesaplanır. Apriori-gen fonksiyonu kullanılarak L_1 ' den C_2 aday kümesi oluşturulur. C_2 ' deki nesnekümlerin destek değerleri hesaplanırken tüm veri tabanı yerine \hat{C}_1 taranır ve \hat{C}_2 oluşturulur daha sonra \hat{C}_2 ' deki destek değerleri sayılarak da L_2 elde edilir. Yine apriori-gen fonksiyonu kullanılarak L_2 ' den C_3 oluşturulur. Verinin üzerinden \hat{C}_2 ile geçilir ve C_3 ' ten \hat{C}_3 oluşturulur. \hat{C}_3 kümesinde 100 ve 400. işlemlere giriş yapılmadığı dikkat çekmektedir. Çünkü 100 ve 400. işlemlerdeki nesnelere C_3 aday kümesindeki nesnekümlerde yer almamaktadır. C_3 kümesindeki tek aday $\{2\ 3\ 5\}$ nesnekümesi minimum destek değerini sağlayarak L_3 kümesini oluşturur. C_4 kümesi boş küme olduğundan algoritma sonlandırılır.

3.6.3. AprioriHybrid algoritması

Veri üzerindeki tüm geçişlerde aynı algoritmayı kullanma zorunluluğu yoktur. Yapılan bir çalışmada kullanılan bir veri tabanında Apriori ve AprioriTid algoritmalarının verinin üzerinden farklı geçişlerdeki çalışma süreleri farklı elde edilmiştir. İlk geçişlerde Apriori algoritmasının daha iyi olmasına rağmen son geçişlerde AprioriTid algoritması daha iyi performans göstermiştir. Apriori ve AprioriTid aynı aday oluşturma prosedürü kullanır ve aynı nesnekümleri sayar. Sonraki geçişlerde nesnekümlerin sayısının azalmasına rağmen Apriori veri tabanındaki tüm işlemleri kontrol etmeye devam eder. Diğer taraftan AprioriTid destek sayılarını elde etmek için veri tabanından daha küçük boyuttaki C_k ' yi tarar. Bu incelemeler üzerine AprioriHybrid algoritması tasarlanmıştır. Bu algoritmada ilk geçişlerde Apriori algoritması kullanılırken C_k kümesinin belleğe sığabileceğinin anlaşılması üzerine AprioriTid' e geçilir.

AprioriHybrid neredeyse bütün durumlarda Apriori' den daha iyi performans gösterir. Ancak Apriori' den AprioriTid' e geçiş maliyetlidir. Genellikle

AprioriHybrid' in Apriori' ye göre avantajı sonraki geçişlerde C_k kümesinin boyutunun azalmasına bağlıdır. Eğer C_k uzun süre büyük kalıp sonlara doğru ani düşüş gösterirse AprioriTid kısa bir süre kullanıldığından AprioriHybrid' in kullanılması amacına ulaşmaz. Diğer yönden C_k ' nın boyutundan büyük ölçüde azalma olursa AprioriTid' in kullanılmasıyla çalışma süresinde önemli bir gelişme sağlanabilir [49].



Şekil 3.12. AprioriTid algoritması örneği

3.6.4. AIS algoritması

AIS algoritması sık geçen nesnekümleri keşfetmek için ortaya çıkarılmıştır. AIS, aday nesnekümleri veri tabanı taranırken anında sayar ve oluşturur. Her bir geçiş boyunca bütün veri tabanı taranır. Bir işlem okunduktan sonra önceki geçişte sık bulunan nesnekümlerin bu işlemde hangilerinin içerildiği tanımlanır. Bu sık geçen nesnekümler genişletilerek yeni aday nesnekümler oluşturulur. Bir f sık geçen nesneküme sadece sık geçen nesnekümlerle genişletilir, f kümesindeki her bir nesne alfabetik sırada dizilir [49].

3.6.5. SETM algoritması

Setm algoritması sık geçen nesnekümleri bulmak için SQL kullanır. AIS gibi SETM algoritması da veri tabanı taranırken adayları oluşturur. Ancak, aday oluşturmak için standart SQL komutlarını kullanır. Bu iki algoritma da gereksiz birçok aday oluşturup saydığından etkin değildir [49].

3.6.6. FP-Growth algoritması

Apriori algoritmasının dezavantajlarından biri aday oluşturmadır. Uzunluğu 100 olan $\{a_1, a_2, \dots, a_{100}\}$ şeklindeki bir sık örüntüyü oluşturmak için toplam aday sayısı yaklaşık $1,27 \times 10^{30}$ olacaktır. Bu adayları oluşturabilmek için de veri tabanının yüzlerce kez taranması gerekir. Bu hesaplamanın karmaşıklığı üssel olarak artmaktadır [56].

Frequent-Pattern Growth ya da kısaca FP-Growth metodu büyük veri tabanlarında sık geçen nesnekümler madenciliği için etkili bir yöntemdir. FP-Growth metodu genişlik öncelikli yaklaşıma tipik bir örnektir FP-Growth, fazla zaman alan aday oluşturma prosesi olmaksızın sık geçen nesnekümleri oluşturur. Esasen bu algoritma, arama maliyetlerini düşürür.

FP-Growth metodu en az sık geçen nesnekümleri sonek olarak kullanılır. Algoritma ilk olarak sık geçen nesnekümleri bir FP-ağacına sıkıştırır daha sonra şartlı veri

tabanlarını oluşturarak her bir veri tabanını ayrı ayrı madenler. FP-Growth metodunun işleyişi Tablo 3.6’ da verilen örnek veri tabanına göre anlatılmıştır [10].

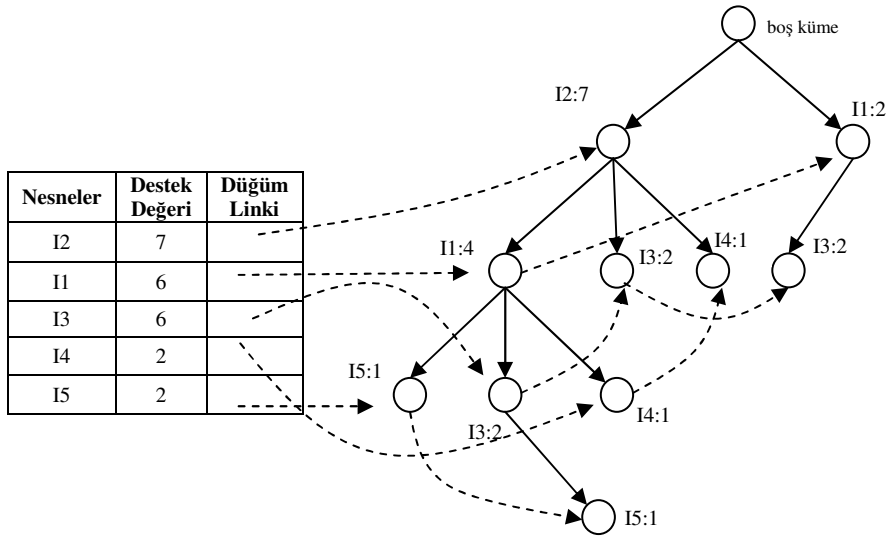
Veri tabanını ilk tarama işlemi Apriori’ de olduğu gibidir. 1 öğeli nesnekümeler oluşturulur ve destek değerleri hesaplanır. Sık geçen nesnekümeler destek değerlerine göre azalan sırada sıralanır, sonuçlar L listesinde kümelenir.

$$L = \{ (I2:7), (I1:6), (I3:6), (I4:2), (I5:2) \}$$

Tablo 3.6. Örnek veri tabanı parçası (minimum destek değeri=2)

TID	Nesneler
T100	I1, I2, I5
T200	I2,I4
T300	I2,I3
T400	I1, I2, I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

L listesi oluşturulduktan sonra FP-ağacı inşa edilir. İlk önce, “null” olarak ifade edilen ağacın kökü oluşturulur. Veri tabanı ikinci kez taramır. Her işlemdeki nesneler L listesindeki sıraya göre işleme alınır ve her bir işlem için bir dal oluşturulur. Örnek olarak; T100: I1,I2,I5 işlemi L listesindeki göre (I2, I1, I5) ağacın ilk dalının üç düğümünü {(I2:1), (I1:1) ve (I5:1)} şeklinde oluşturur. İkinci işlem, T200, L listesine göre I2 ve I4 nesnelerini içerir. I4, köke bağlı olan I2 düğümüne eklenir. Bu şekilde kullanılan her ortak önek değeri 1 arttırılır, yeni eklenen düğümün değeri de 1 olur. Bütün işlemler tarandıktan sonra elde edilen ağaç Şekil 3.13’ te gösterilmiştir. Böylelikle veri tabanlarında sık örüntüler oluşturma problemi FP-ağacı madenlenmesine dönüştürülmüştür.



Şekil 3.13. Bir FP-ağacı

FP-ağaç madenciliği uzunluğu 1 olan bir sık örüntüden başlar ve şartlı örüntü temeli oluşturulduktan sonra da şartlı FP-ağacı oluşturulur. Şekil 3.13' te özetlenmiş olan FP-ağacı ayrıntıları aşağıda anlatıldığı gibidir.

L listesine göre sık geçen nesnekümlerinin tamamı 5 alt kümeye (bu örneğe göre) ayrılabilir. Listenin sonundan başlayarak;

1. I5' i içeren,
2. I4'ü içeren ancak I5' i içermeyen,
3. I3' ü ancak I4 ve I5' i içermeyen,
4. I1'i ancak I3, I4 ve I5'i içermeyen,
5. I2' yi ancak I1, I3, I4 ve I5' i içermeyen sık geçen nesnekümler.

I5, FP-ağacının iki dalında bulunur. Bu dallar (I2, I1, I5: 1) ve (I2, I1, I3, I5: 1)' dir. I5 bir son ek olarak dikkate alındığından dolayı bu nesnenin önek yolları (I2, I1: 1) ve (I2, I1, I3: 1) olup şartlı örüntü temeli formunu oluşturur. Bu nesnenin FP-ağacı

Tablo 3.8. FP-Growth algoritmasının kodları

<p>Algoritma: FP Growth. Sık geçen nesnekümelerin bir FP-Ağacı kullanılarak madenlenmesi</p> <p>Girdi: İşlemsel bir DB veri tabanı, minimum destek eşik değeri</p> <p>Çıktı: Sık örüntü setinin tamamı</p> <p>Metot:</p> <ol style="list-style-type: none"> 1. FP Ağacı aşağıdaki adımlara göre oluşturulur: <ol style="list-style-type: none"> a) ilk önce DB veri tabanı taranır. Sık geçen nesneküme F ve destek değerleri belirlenir. F kümesi destek değerine göre azalan değerde sıralanır, sık geçen nesneküme L listesi oluşturulur. b) FP Ağacının kökü oluşturulur ve “null” olarak adlandırılır. Her işlem için DB veri tabanında devam eden Trans işlemleri gerçekleştirilir. L listesine göre Trans işlemindeki sık geçen nesneküme seçilir ve sıralanır. Sıralanmış sık nesnelere Trans'ta listelenir $[p P]$, p ilk eleman ve P ise geriye kalan liste. Insert_tree ($[p P]$, T) olarak isimlendirilir. Insert_tree($[p P]$, T) fonksiyonu şu şekilde devam eder: Eğer T' nin N gibi bir yavrusu varsa, N.nesne-ismi=p.nesne-ismi, sonra N' nin sayısı 1 arttırılır; diğer durumda N gibi yeni bir düğüm oluştur, düğümün sayısını 1 yap ve düğümün ana linkini T' ye bağla. Ayrıca aynı nesne-ismi yolu ile diğer düğümlere de bağlanır. Eğer P boş küme değil ise insert_tree (P,N)' i çağır. 2. FP-Ağacı, aşağıdaki gibi yürütülen FP growth(FP ağacı, null) ile madenlenir: <p>Prosedür Fp growth (ağaç, α)</p> <ol style="list-style-type: none"> (1) If ağaç P gibi tek bir yoldan oluşursa (2) P yolundaki düğümlerin(β ile gösterilen) her bir kombinasyon için (3) β daki düğümlerin destek=minimum destek sayıları ile $\beta \cup \alpha$ örüntülerini oluştur; (4) Else ağacın baş kısmındaki her a_i için { (5) $\beta = a_i \cup \alpha$ örüntüsü destek=a_i.destek sayıları ile oluştur; (6) β' nin şartlı örüntü temeli yapısı kurulur ve Then β' nin şartlı FP-Ağacı $Tree_{\beta}$; (7) If $Tree_{\beta} \neq \emptyset$ (8) Then, FP- growth ($Tree_{\beta}, \beta$) çağır }
--

3.7. Birliktelik Kuralları Oluşturma

Bir D veri tabanındaki işlemlerden elde edilen sık geçen nesnekümeleden minimum destek ve güven değeri altında güçlü birliktelik kuralları oluşturulur. Birliktelik kuralları aşağıdaki adımlarda anlatıldığı gibi elde edilir [10].

- Her bir L sık nesneküme için boş olmayan alt kümeleri oluşturulur.
- Her bir boş olmayan L' nin alt kümesi için minimum güven eşik değeri altında oluşturulacak kural “ $L_a \Rightarrow (L-L_a)$ ” dir.

Apriori algoritması ile elde edilen sık geçen nesnekümenin (Bkz. Tablo 3.4) alt kümeleri {Çocuk bezi, Ekmek}, {Çocuk bezi, Süt}, {Ekmek, Süt}, {Çocuk bezi}, {Ekmek}, {Süt}' tür. Bu sık kümeden oluşturulabilecek birliktelik kuralları aşağıdaki gibidir.

Çocuk bezi \wedge Ekmek \Rightarrow Süt,	güven = $3/3 = \%100$
Çocuk bezi \wedge Süt \Rightarrow Ekmek,	güven = $3/3 = \%100$
Ekmek \wedge Süt \Rightarrow Çocuk bezi,	güven = $3/3 = \%100$
Çocuk bezi \Rightarrow Ekmek \wedge Süt,	güven = $3/4 = \%0,75$
Ekmek \Rightarrow Çocuk bezi \wedge Süt,	güven = $3/4 = \%0,75$
Süt \Rightarrow Çocuk bezi \wedge Ekmek,	güven = $3/4 = \%0,75$

Minimum güven eşik değeri $\%80$ olarak belirlenirse birinci, ikinci ve üçüncü kurallar güçlü birliktelik kuralları olarak belirlenir.

3.8. Birliktelik Kuralları Madenciliği Çeşitleri

Apriori algoritması gibi bir kısım birliktelik kuralları algoritmaları, tek seviyeli birliktelik kuralları olarakta bilinen genel birliktelik kuralları üzerinde durur. Aslında bir çok uygulamada güçlü ve ilginç birliktelik kurallarının bulunması zordur. Başka bir yönden ise güçlü ve ilginç birliktelik kuralları tek boyutlu veri tabanı tablosu ile sınırlandırılmaz, bu kurallar çoklu veri tabanı tablolarından da elde edilebilir. Bu yüzden farklı senaryolardan birçok birliktelik kuralları tipleri oluşturulabilir [48].

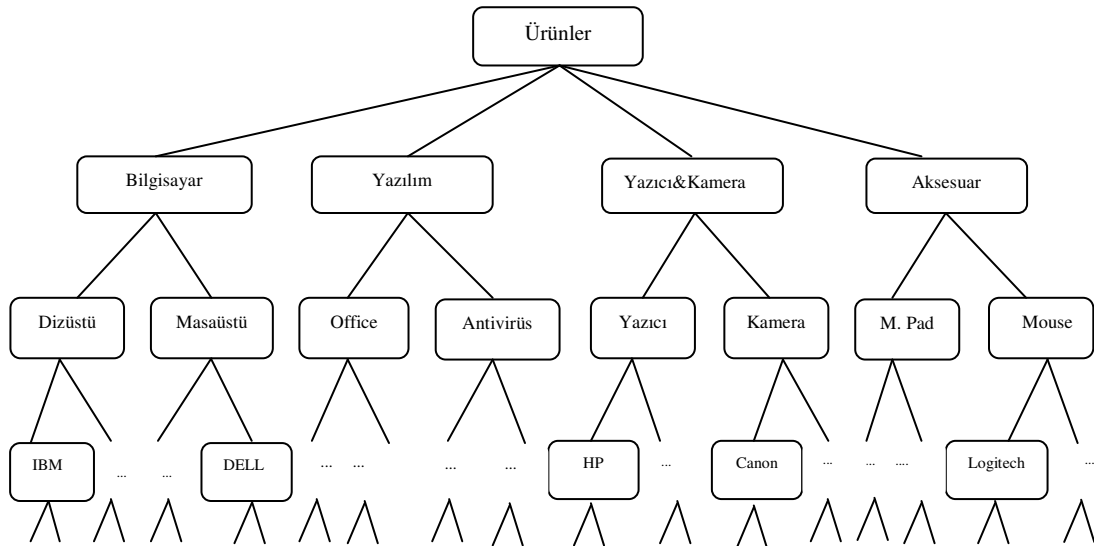
3.8.1. Çok seviyeli birliktelik kuralları madenciliği

Birliktelik kurallarının çoklu seviyelerdeki verilerin madenlenerek oluşturulmasına çok seviyeli birliktelik kuralları madenciliği denir. Çok seviyeli birliktelik kuralları, bir destek-güven çatısı altında hiyerarşi kavramını etkin bir şekilde kullanarak madenlenebilir. Genellikle bir genelden-özele yaklaşımı işletilir. Hiyerarşide 1.seviyeden başlayarak aşağı doğru sık geçen nesneküme bulunamayınca kadar daha spesifik seviyelere inilir. Her seviyede sık geçen nesnekümelere bulmak için Apriori veya onun versiyonlarından herhangi bir algoritma kullanılabilir.

Çok seviyeli birliktelik kuralı madenciliğini bir örnekle açıklamak gerekirse Tablo 3.9' daki bir elektronik mağazasına ait satışlar dikkate alınmalıdır. Tablodaki nesnelere ait hiyerarşi Şekil 3.16' da gösterilmektedir. Bu hiyerarşi, alt seviyelerden üst seviyelere bir sıra dizisini tanımlar. Şekil 3.15' deki hiyerarşi 5 seviyeden oluşmaktadır; 0. seviyeden başlar ve 4. seviyede kadar iner. 4. seviye hiyerarşinin daha spesifik elemanlarından oluşur.

Tablo 3.9. Örnek veri seti

TID	Satın alınan Ürünler
T100	IBM-ThinkPad-T40/2373, HP-Photosmart-7660
T200	Microsoft-Office-Professional-2003, Microsoft Plus!-Digital-Media
T300	Logitech-MX700-Cordless-Mouse, Fellowes-Wirst-Rest
T400	Dell-Dimension-XPS, Canon-PowerShot-S400
T500	IBM-ThinkPad-R40/P4M, Symantec-Norton-Antivirus-2003
...	...



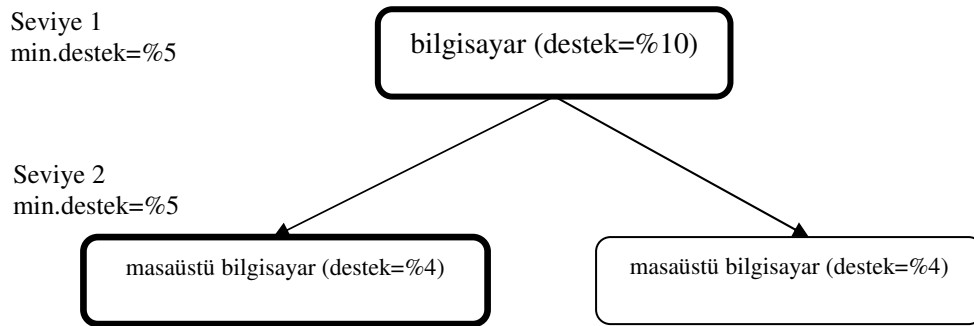
Şekil 3.15. Örnek veri setine ait ürün hiyerarşisi

Çoklu seviyeli birliktelik kuralı madenciliği yaklaşımının iki varyasyonu vardır ve aşağıda tanımlandığı gibidir. Her bir varyasyon biraz farklı yöntemle destek eşik değeri ile oynar. Şekil 3.16 ve Şekil 3.17' de gösterildiği gibi düğümler bir nesne

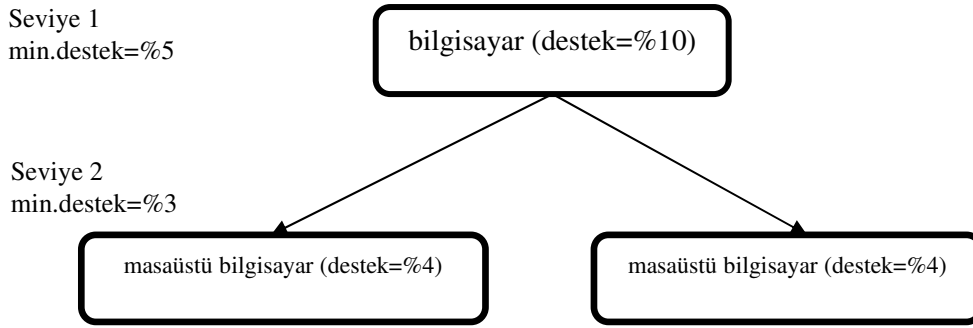
veya nesnekümeyi göstermektedir. Kalın kenarlıkla gösterilen düğümler sık geçen nesneküme olarak bulunmuştur [10].

Bütün seviyelerde tekdüze minimum destek değerini kullanma: Bütün madenleme seviyelerinde aynı eşik destek değeri kullanılır. Örneğin Şekil 3.16’ da bütün seviyelerde aynı eşik destek değeri kullanılmıştır. “bilgisayar” ve “dizüstü bilgisayar” sık geçen nesneküme bulunurken, “masaüstü bilgisayar” sık olmayan nesneküme olarak elde edilmiştir. Bu yaklaşımın dezavantajı; eğer minimum destek eşik değeri çok yüksek ayarlanırsa yöntem alt seviyelerde bazı anlamlı birliktelikleri kaçırabilir. Destek eşik değeri çok düşük ayarlanırsa üst seviyelerde birçok gereksiz kural üretilebilir.

Alt seviyelerde indirilmiş minimum destek değerini kullanma: Her seviyenin kendine ait minimum destek değeri mevcuttur. Şekil 3.17’ de görüldüğü gibi seviye 1 ve 2 için minimum destek değerleri sırasıyla %5 ve %3’ tür. Böylece “bilgisayar, “dizüstü bilgisayar” ve “masaüstü bilgisayar” sık geçen nesneküme olarak belirlenir.



Şekil 3.16. Tekdüze destek değeri ile çoklu seviyeli madenleme



Şekil 3.17. İndirilmiş destek değeri ile çoklu seviyeli madenleme

3.8.2. Çok boyutlu birliktelik kuralları madenciliği

Bundan önceki konularda işlemsel veri tabanlarındaki birliktelik kuralları üzerinde durulmuştur. Bu tip kuralları genellikle tek boyutlu birliktelik kurallarını kapsamaktadır. Tek boyutlu birliktelik kuralları; tek bir fiilden oluşturulan kurallardır. Tablo 3.9’ daki veri setine göre oluşturulabilecek bir Boolean kural aşağıdaki gibidir.

satın alma (X, “dijital kamera”) \Rightarrow satın alma (X, “HP Printer”)

İlişkisel bir veri tabanında veya bir veri ambarında bir nesneyi satın alan müşteriye ait meslek, yaş, gelir, adres gibi veriler de kayıt altına alınmaktadır. Veri tabanlarındaki bu nitelikler, çok boyutlu bir birliktelik kuralının bir boyutunu oluşturur.

yaş (X, “20-29”) \wedge meslek (X, “öğrenci”) \Rightarrow satın alma (X, “laptop”)

Çok boyutlu birliktelik kuralı madenciliği için Han ve Kamber bir veri kübü modeli önermiştir [10].

3.8.3. Nicel birliktelik kuralları madenciliği

Birliktelik kurallarını oluşturmanın problemlerinden önceki konularda bahsedilmiştir. Bu problem, bütün özelliklerin boolean olan bir ilişkisel tabloda “1” değerleri arasından birliktelikleri bulma olarak da düşünülebilir. Bu tabloda her nesneye uyan bir özellik ve her işleme uyan bir kayıt mevcuttur. Nesneye uyan özellik işlemde mevcut ise verilen bir kayıt için özellik değeri “1” dir, başka bir kayda uyuyorsa “0” dir. Bu durum Boolean birliktelik kuralları olarak tanımlanmıştır. Boolean birliktelik

kuralları herhangi bir algoritma ile bulunabilir ve boolean birliktelik kuralları da nicel birliktelik kurallarını elde etmek için kullanılabilir.

İlişkisel tablolarda birçok özellik tipi bulunmaktadır. Bu özellikler kantitatif (yaş, gelir vb.) ve kategorik (posta kodu, araba markası vb.) olabilir. Srikant ve Agrawal tarafından kantitatif ve kategorik özellikleri içeren ilişkisel tablolarda bu tür kuralları elde edebilecek teknikler sunulmuştur. Bu tür madenlemeye de nicel birliktelik kuralları adı verilmiştir [57].

Örnek olarak 3 anahtar olmayan özellikli bir veri tabanı parçası Tablo 3.10' da gösterilmiştir. Yaş ve araba sayısı nicel özellik, medeni hal ise kategorik özelliktir [57].

Tablo 3.10. Nicel ve kategorik özelliklerden oluşan örnek veri seti

Kayıt Numarası	Yaş	Medeni Hali	Araba sayısı
100	23	Bekar	1
200	25	Evli	1
300	29	Bekar	0
400	34	Evli	2
500	38	Evli	2

Tablo 3.11. Nicel birliktelik kuralları örneği (min.s=%40, min.c=%50)

Örnek Kurallar	Destek	Güven
< Yaş: 30-39 > ve < Medeni Hali: Evli > \Rightarrow < Araba sayısı: 2	%40	%100
< Araba sayısı: 0-1 > \Rightarrow < Medeni Hali: Bekar >	%40	%66.6

Tablo 3.11' e göre, bir nicel birliktelik kuralı aşağıdaki gibi verilebilir:

< Yaş: 30-39 > ve < Medeni hali: Evli > \Rightarrow < Araba sayısı: 2 >

Tablo 3.12' de anahtar olmayan özelliklerin incelenmesi gösterilmiştir. Yaş 20-29 ve 30-39 gibi iki aralığa ayrılmıştır. Kategorik özellik olan medeni hali ise iki boolean

özelliğe ayrılmıştır: “Medeni Hali: Evli” ve “Medeni Hali: Bekar”. Araba sayısı değerleri küçük olduğundan aralıklara bölmek yerine boolean olarak incelenmiştir.

Tablo 3.12. Boolean birliktelik kuralı şeklinde inceleme

Kayıt No	Yaş: 20-29	Yaş: 30-39	M.H. : Evli	M.H. : Bekar	A.S. :0	A.S. : 1	A.S. :2
100	1	0	0	1	0	1	0
200	1	0	1	0	0	1	0
300	1	0	0	1	1	0	0
400	0	1	1	0	0	0	1
500	0	1	1	0	0	0	1

Örneğin 100. kayıta yaş 20-29, medeni hali bekar ve araba sayısı 1’ dir. Burada yaş, Medeni hali ve araba sayısı anahtar olmayan özelliklerdir ve farklı kayıtlar için aynı değerleri alabilir ancak kayıt numarası anahtar bir özelliktir ve sadece o kayda ait ayırt edici bir özelliktir.

Nicel birliktelik kuralları madenciliği bazı problemler içerir. İlk problem bölümler arasında keskin sınırların bulunmasıdır. Algoritmalar bölümler arasındaki sınırlardaki elemanların etkisini ya zayıflatır ya da aşırı değerli hale getirir. Örnek olarak aralık metodunun kullanıldığı bir sınıflandırmada 35 yaşın iki aralığı birbirinden ayırdığı düşünülürse; yapılan sınıflandırmaya göre 35’ ten küçük olanlar genç, yaşı 35’ ten büyük olanlar ise yaşlı olarak nitelendirilmiş olur. Buradan da görüldüğü gibi aslında günlük hayatta bu kadar keskin sınırlar geçerli değildir. Bu tür problemler, yapay zeka teknikleriyle çözülebilmektedir [58].

BÖLÜM 4. MAĞAZA İÇİ YERLEŞİM DÜZENİ

4.1. Mağaza İçi Yerleşiminin Önemi

Mağaza şekli tüketici davranışlarını etkileyen önemli bir faktördür ve mağaza yerleşim dizaynı da yerleşim şekli oluşturulmasında kritik karar tayin edici ve tanımlayıcı bir etmendir. Lewison' a göre; satış yerleri düzeni son derece önemlidir çünkü mevcut alışveriş zamanı ve sıklığını, alışveriş atmosferini, alışveriş davranışını ve operasyonel verimliliği tamamen etkilemektedir. Hatta müşterilerin mağaza içinde gezinerek, aslında satın almayı düşünmediği ürünleri de satın almasına sebep olur. Benzer şekilde Merrilees ve Miller raporunda, mağaza yerleşim düzeni dizaynı satış yeri bağımlılığına karar verici ve tanımlayıcı en önemli etmenlerden biri olduğunu ve Simonson, mağaza düzeni sadece müşteri ihtiyaçlarını karşılamada tatmin edici değil ayrıca müşterilerin isteklerini ve tercihlerini etkileyen anahtar bir rol oynadığını belirtmektedir [59].

Mağaza içi yerleşim düzeninin nasıl olması gerektiğine karar verilirken, etkinlik ön planda tutulmalıdır. Etkin bir mağaza içi yerleşim düzeni müşterileri satın almaya teşvik eder. Hatta müşterilerin mağaza içinde gezinerek aslında satın almayı düşünmediği ürünleri de satın almasına sebep olur. Ayrıca, mağaza alanının en verimli şekilde kullanılmasını sağlar.

Mağaza içi yerleşim düzeninin temel amacı, müşteri trafiği oluşmasına teşvik ederek müşterilerin perakendeciye ait tüm ürünleri görmesine müsaade etmektir. Ayrıca müşterinin, normal şartlarda ziyaret etmeyeceği bölgelere onu yönlendirmektir [60].

4.2. Temel Alan Düzenlemeleri

Doğru bir yerleşim planı, müşterilerin izleyeceği yolu bulmalarına, mağaza içi trafik akışının düzenlenmesine ve mağazanın etkin bir şekilde kullanılmasına yardımcı olur. Etkin bir yerleşim planı müşterileri satın almaya teşvik eder. Mağaza içi yerleşim planı müşterilerin perakendeciye ait tüm ürünleri görmesine izin vermelidir. Müşterilerin ziyaret etmeyeceği bölgelere onu yönlendirmelidir.

Perakendecilerin yerleşim planı konusunda tercih edebileceği birçok seçenek bulunmaktadır. Uygun seçenek, mağaza konumlandırmasına, sergileme türüne, mağaza büyüklüğüne, hedef kitleye, uygulanacak satış yöntemine, fiyat seviyesine ve güvenlik olanaklarına bağlı olarak seçilmelidir. Yerleşim düzeninde kullanılacak en uygun plan/planların neler olduğuna market içerisinde müşteri davranışlarını inceleyen araştırmalar sonucu karar verilmiştir [61].

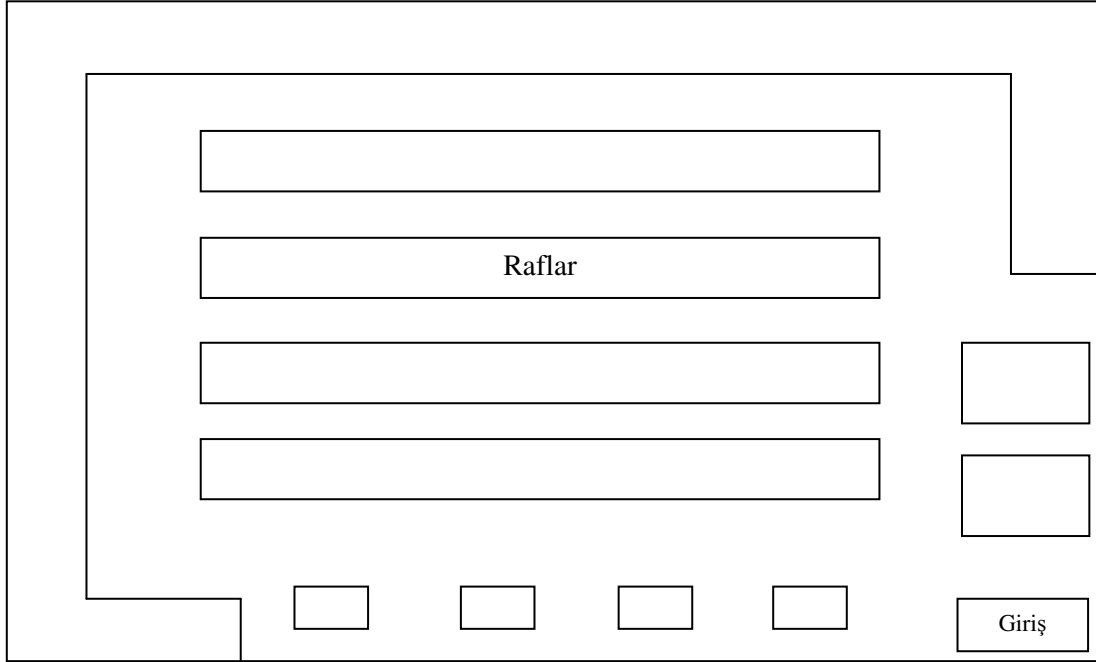
Literatürdeki geleneksel perakendecilik mağaza yerleşim teorisine göre 3 tip yerleşim düzeni mevcuttur. Bunlar; ızgara (grid), serbest düzen (freeform) ve yarış pisti (racetrack) yerleşim türleridir. Gıda perakendecileri tarafından tercih edilen yerleşim türü grid yerleşimdir. Diğer yerleşim türleri mağazalar ve butikler tarafından tercih edilmektedir [59].

4.2.1. Izgara (Grid) biçimi

Bu yerleşim biçimi gıda perakendecileri tarafından daha çok kullanılmaktadır. Market içi genellikle birbirine paralel, aralarında uzun geçitlerin olduğu dikdörtgen düzen görünümündedir, kasalar genellikle market giriş ve çıkışa yakın bölgelerde yer almaktadır. Izgara biçimi yerleşim düzenine Şekil 4.1 örnek olarak gösterilebilir. Izgara biçimi sunum açısından o kadar da çarpıcı olmamasıyla beraber, müşterinin market içerisinde aradığı ürüne hemen ulaşmasını sağladığı için tercih edilmektedir [62].

Izgara yerleşim düzeninde müşteriler istedikleri ürünlere ulaşırken hiyerarşik bir yapı içinde dolaşmaktadır (Örneğin; ürün kategorileri $\leftarrow \rightarrow$ alt ürün kategorileri $\leftarrow \rightarrow$ son

ürün). Örneğin, haftalık alışveriş için markete gelen müşteriler her ürünün nerede olduğunu iyi bilir ve her seferinde arama işlemine ek zaman ayırmaz. Ayrıca bu plan maliyet bakımından en elverişlisidir [59, 62].



Şekil 4.1. Izgara biçimi yerleşim

Izgara biçiminde minimum yer kaybı olur. Diğerlerinden farklı olarak bu biçimde raflar arasında sadece müşteriler ve onların kullandığı alışveriş arabalarına yer bırakılır. Diğer yerleşim türleri mağazalar ve butikler tarafından tercih edilmektedir. Müşteriler alışveriş planlarını önceden belirlemiş olarak markete gittiklerinden grid yerleşim market sektöründe avantaj sağlamaktadır [59, 62].

Izgara yerleşimin diğer avantajları:

- Yerleşim türleri arasında mağaza kullanım alanında en verimli olanıdır.
- Açık ve belirgin koridorlar sayesinde alışverişini kolaylaştırır.
- Verimli bir mağaza atmosferi yaratır.
- Rutin ve planlanmış alışveriş davranışlarını kolaylaştırdığı gözlenmiştir.

4.2.2. Yarış alanı biçimi

Bu planın en büyük dezavantajı tüketicilere tüm sunumu görme olanağı sağlamamasıdır. Bu nedenle gıda perakendeciliğinde istifadesine ender rastlanmaktadır. Izgara biçimi plan müşterilerin tek seferde birkaç bölüme girmesini teşvik etmektedir. Bu düzende bölümler arası geçitler büyük önem arz etmektedir. Anlık satın alma kararlarını tetikleyici unsurları barındıran bu biçim sayesinde müşterilerin dikkati uygun bir sırayla dizilmiş bölümlere çekiliyor. Yarış alanı biçimi yerleşim planında birkaç giriş kullanılmaktadır [62].

4.2.3. Serbest yerleşim planı

Butik biçimi olarak da adlandırılan bu düzende askı ve demirbaşlar asimetrik şekilde dizilir. Küçük özellikli mağazalar ve departmanlı mağazaların bazı katlarında başarıyla uygulanıyor. Bu biçim sayesinde müşterilerine kendini evindeymiş gibi hissetmesi sağlanıyor. Böylece onlar ürünleri daha rahat inceliyor ve alışveriş verimli geçiyor. Bu düzende müşterilerin mağaza içerisindeki trafiğini yönlendirmek mümkün olmadığından daha fazla müşteri temsilcisi bulundurmamak gerekiyor. Ayrıca bu biçimde daha fazla alana gerek oluyor. Buna rağmen, kaliteli düzenleme satışı ve karı artıracak için serbest biçimin etkili kullanımı mümkündür [62].

4.3. Ürünlerin Gruplandırılması

Mağaza içinde ürünlerin gruplandırılma biçimleri ürün satışını doğrudan etkiler. Ürünler türlerine göre sınıflandırılıp, mağazada uygun şekilde konumlandırılırsa müşterilerin seçim yapması kolaylaşır ve satışlar teşvik edilmiş olur [61].

Mağazada tüm alanlar aynı değere sahip değildirler. Bazı bölgelerin satış potansiyeli daha fazla iken, bazı bölgelerde satış yapmak daha zordur. Örneğin, mağazanın ön kısımlarında yer alan ürünlerin satılma ihtimali, arkalarda yer alan ürünlere göre daha fazladır. Diğer bir ifade ile mağaza içinde ürünlerin sergilendiği bölgeler satışlar üzerinde etkilidir.

Mağazanın ön kısımlarında yer alan ürünlerin tüketiciler tarafından görülme ve incelenme ihtimali, müşteri trafiğinin yoğun olması sebebiyle daha fazladır. Bu sebeple mağazada giriş en pahalı alan iken girişe yakın alanlar ikinci değerde, arkaya doğru ise daha az değerdedir.

Müşteriler mağazaya girdiklerinde genellikle sağ taraftan ortaya doğru yöneldikleri için, satılması arzu edilen ürünler bu bölgelerde sergilenmelidir. Bu bölgeler aynı zamanda anlık ve kolayda ürünler için de uygun yerlerdir. Sebze, meyve, konserve, çorba gibi her gün satın alınan ürünler mağaza girişine yakın yerlere konmalıdır. Ayrıca sebze, meyve bolluk ve tazeliği temsil ettiğinden müşteriye mağazaya çekmektedir. Böylece müşteri girdiği anda satın almaya başlar.

Kolayda mallar, müşterilerin en az güç ve zaman harcayarak satın almak istediği ürünlerdir. Örneğin, çiklet bisküvi vb. Bu tip ürünler tüketicinin fazla ödeme yapmadığı, mağazada fazla satış gücü harcamadığı ürünlerdir. Mağaza içinde tüketici trafiğinin en yoğun olduğu ve en fazla görünen yerlerde teşhir edilmelidirler. Ancak sürekli satın alınan kolayda ürünler (şeker, makarna, pirinç, un, vb.) nasıl olsa tüketici tarafından satın alındıklarından, arkalara veya yan duvarların bulunduğu bölgelere yerleştirilebilir [60].

Plansız satın alınan ürünler, müşterilerin fazla ödeme yapmadığı ürünlerdir. Satın alınmasına anlık karar verilen bu ürünler mağazada müşteri akımının yoğun olduğu yerlerde konumlandırılmalıdır. Örneğin; dergiler, şeker, sakız ve sigaralar. Daha çok alışveriş çıkışına yakın, kasa yakınında bulunmaları tavsiye edilir [61].

Beğenmeli mallar, tüketicilerin zaman ve emek harcayarak birçok marka arasından, fiyat ve kalite karşılaştırması yaparak satın aldıkları ürünlerdir. Televizyon, beyaz eşya vb. ürünler bu gruba örnek gösterilebilir. Bu tip ürünler hacimli ürünler olduklarından mağazada dolaşma alanının sonuna doğru yerleştirilmelidir. Çünkü müşteri alışverişin en başında bu ürünleri sepetine koyarak başlarsa diğer ürünlere yer kalmaz.

Son olarak da tamamlayıcı ürünler birbirlerine yakın yerleştirilmelidir. Birbiri ile birlikte kullanılan, tamamlayıcı ürünler bir arada sergilenmesi hem müşteriye kolaylık sağlar hem de müşterinin aklında olmayan bir alışveriş yapmasına sebep olur. Örneğin; erkek ayakkabıları ve çorapları, saç bakım ürünleri ve saç aksesuarları, çocuk giysileri ve ayakkabıları tamamlayıcı ürünler olduklarından müşterinin hepsini bir arada bulması bu ürünlerin satışını arttırmaktadır. Böylece çapraz satışı teşvik etmek amacıyla birçok mağazada ilgili ürünler bir arada veya yan yana bölümlerde sergilenmektedir. Ayrıca ürünlerin yerleri de sık sık değiştirilmemelidir. Müşteriler mağazaya girince ürünleri alıştıkları yerde görmek isterler. Mağazada ürünler yerleştirilirken tüm bu unsurlara dikkat edilmeli ve uygun bir düzen oluşturulmalıdır [60, 61].

BÖLÜM 5. ÇOK BOYUTLU ÖLÇEKLEME ANALİZİ

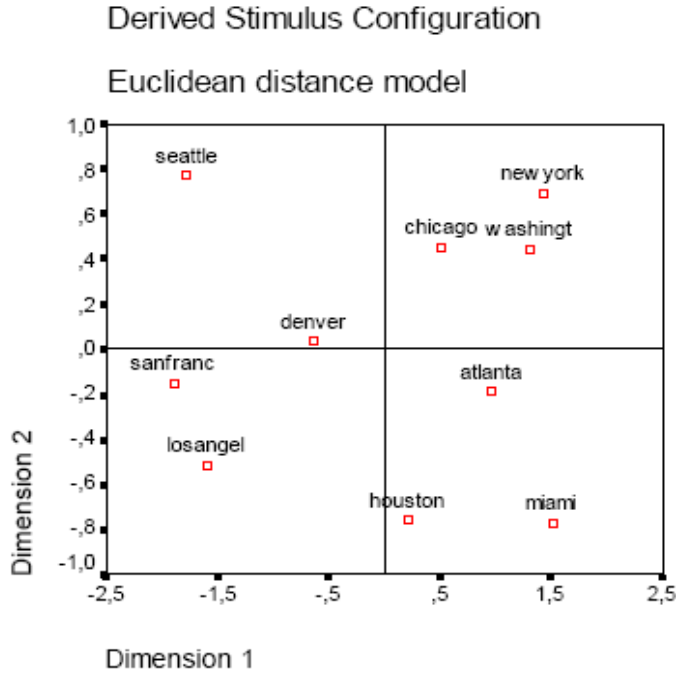
5.1. Çok Boyutlu Ölçekleme Analizi Tanımı

Çok boyutlu ölçekleme analizi (Multidimensional Scaling-MDS), nesne ya da birimler arasında gözlemlenen benzerlikler ya da farklılıklardan oluşan uzaklık değerlerine dayalı olarak bu nesnelerin tek ya da çok boyutlu uzaydaki gösterimini elde etmeyi amaçlayan, böylece nesnelere arasındaki ilişkilerin belirlenmesini sağlayan çok değişkenli bir istatistiksel analiz yöntemidir [63, 64].

Çok boyutlu ölçeklemenin ilk olarak ortaya çıkışı 1938 yılında Young ve Householder'ın yaptığı çalışmalara kadar götürülebilmektedir. 1958'e kadar üzerinde fazla durulmayan bu yöntem Torgerson ve arkadaşları tarafından daha da geliştirilmiştir. Pazarlama araştırmalarında ilk uygulanması da psikologlar tarafından yapılmıştır. Ancak çok boyutlu ölçekleme asıl gelişmesini pazarlama disiplini içinde sağlamıştır [65, 66].

Pazarlamacılar çok boyutlu ölçekleme metodlarını, varlıklar üzerindeki (markalar, reyon mağazaları, başkan adayları vs.) müşteri algı ve önceliklerinin öklid uzaydaki bir haritada gösterilmesinde kullanmışlardır. ÇBÖ analizinin temel sonucu bir uzaysal haritadır. Veri uzaklık matrisinden elde edilen birim ya da nesnelere arasındaki uzaklıkların daha az boyutlu bir grafiksel olarak gösterilmesine grafiksel gösterim adı verilir. Grafiksel gösterimi elde etmek için veri koordinatlarının en az hata ile grafiksel gösterim koordinatlarına dönüştürülmesi gerekmektedir. Şekil 5.1' de örnek harita gösterilmektedir. Bu haritada nesnelere noktalar olarak gösterilmektedir. Analizde, nesnelere benzerlikleriyle nesnelere arasındaki uzaklıklar karşılaştırılmaktadır. Noktalarla temsil edilen benzer nesnelere birbirine daha yakın, farklı nesnelere ise birbirine daha uzaktır. ÇBÖ analizinin genel amacı, az boyutla,

nesnelerin yapısını (uzaklık değerlerini kullanarak) orijinal şekle yakın bir biçimde ortaya koymaktır [63, 67].



Şekil 5.1. İki boyutlu düzlemde örnek grafiksel gösterim

5.2. Çok Boyutlu Ölçkleme Analizinde Kullanılan Yöntem

ÇBÖ literatüründe temel ayrım, metrik ve metrik olmayan modeller arasındaki farktır. Metrik ve metrik olmayan modeller, ÇBÖ modeli tarafından tahmin edilebilen koordinatlarla hesaplanan, veri ve uzaklıklar arasındaki ilişkiler hakkında farklı varsayımlara sahiptir. Metrik modeller ilişkinin en az aralık ölçeği ile ölçülmüş niteliklere sahip olduğunu varsayarken, metrik olmayan modeller sadece ordinal ölçekle ölçülmüş nitelikleri gerektirmektedir [63].

MDS' de n birim ya da nesneden oluşan bir veri setinden elde edilen uzaklık değerleriyle, n noktanın tek ya da m ($m < n$) boyutlu uzayda, genellikle öklid uzayında grafiksel gösterimini elde etmek amaçlanmaktadır. M boyutlu bir öklid uzayında, i . ve j . noktalar arasındaki uzaklık eşitlik 12' deki gibi elde edilmektedir [68].

Öklid uzaklığı, her bir boyuttaki (değişken) farkın karesinin ve bu kareli farkların toplamının karekökünün alınmasıyla hesaplanan bir geometrik uzaklık ölçüsüdür. Öklid uzaklığı, Pisagor teoreminden çıkmakta ve hipotenüsün uzunluğunun hipotetik dik üçgenin iki noktası arası olduğunu ifade etmektedir. Öklid uzaklığının formülü eşitlik 12' de verilmiştir [63, 64].

$$d_{ij} = \sqrt{\sum_{\alpha=1}^{mL} (X_{i\alpha} - X_{j\alpha})^2} \quad (12)$$

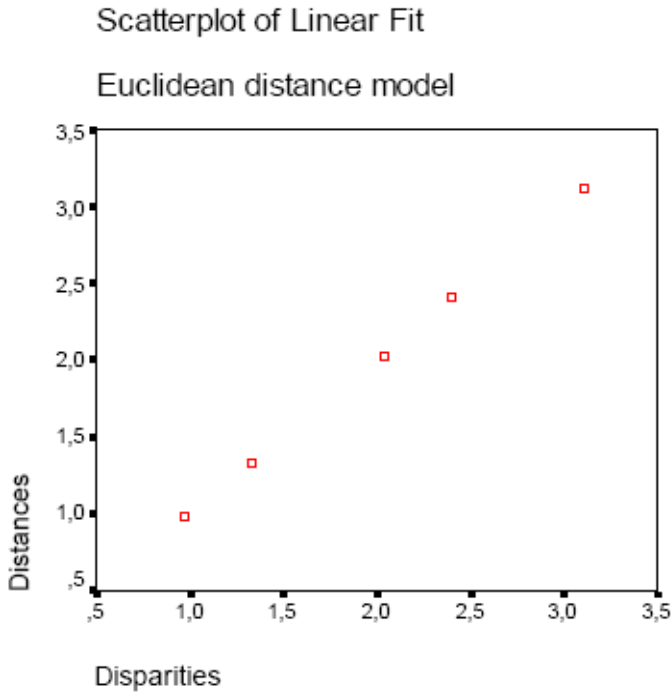
Veri setinde bulunan i. ve j. birimler arasındaki uzaklık ölçümü p_{ij} ile gösterilmek üzere, MDS bu uzaklıkların bir geometrik uzayda, örneğin öklid uzayında gösterimini sağlar. Öklid uzaklıkları (d_{ij}) ile gözlenen uzaklıklar (p_{ij}) arasındaki ilişki, uygun bir $d_{ij}=f(p_{ij})$ dönüşümü yardımıyla ortaya konulur. Örneğin bu ilişki $f(p_{ij})=a(d_{ij})+b$ fonksiyonu ile gösterilebilecek doğrusal bir ilişki olabilir. Uzaklıklara dayalı bu yöntemde genel olarak öklit uzaklıklarının kullanılmasına karşın, asimetrik uzaklıkların bulunması durumunda öklid yerine diğer uzaklık ölçütlerinden de yararlanılmaktadır [68].

Uyumun veya uyum iyiliğinin bir ölçüsü olan gerginlik (stress) ölçüsü, ÇBÖ analizinde geniş bir kullanıma sahiptir ve analiz sonucu elde edilen grafiksel düzenlemede kullanılan boyut sayısının uygun olup olmadığının belirlenmesinde önemli bir ölçüt olarak karşımıza çıkmaktadır. Burada n değişkenli n boyutlu veri matrisine sahip olan n birey ya da birimin kaç boyutlu bir uzayda gösterilebileceğine karar vermede, istenilen boyut için elde edilen konfigürasyon uzaklıkları ile orijinal veriden elde edilen uzaklıklar arasındaki uygunluk dikkate alınmaktadır. Gerginlik ölçüsü;

$$s = \sqrt{\frac{\sum (d_{ij} - p_{ij})^2}{\sum p_{ij}^2}} \quad (13)$$

Stres deęerinin 0' a eřit olması mükemmel uyumu gösterirken, 1' e eřit olması tam uyumsuzluęu göstermektedir. Kruskal 0,05 stres deęerinin iyi, 0,20 stres deęerinin zayıf bir uyumu gösterdięini ileri sürmüřtür. Ancak bu deęiřebilen bir kavram olup, birim ya da nesne sayısı arttıka ve boyut sayısı azaldıkça stres deęerinin arttıęı bilinmektedir [63, 64, 68, 69].

MDS' de gözlemsel uzaklıklar ile konfigürasyon uzaklıklarının uyumluluęunu Shepard diyagramı adı verilen bir grafik ile deęerlendirilebilir. Shepard diyagramında, gözlenen uzaklıklar Y ekseninde ve farklar (disparity) deęerleri X ekseninde yer alacak biçimde bir daęılım (scatterplot) grafięi oluřturulur. Gözlenen uzaklıklar ile konfigürasyon uzaklıklarının daęılımını gösteren shepard grafięi Őekil 5.2' de gösterilmiřtir. Buna göre iki farklı uzaklık deęerleri arasında doęrusal bir iliřki bulunduęu görölmektedir [67, 68].



Őekil 5.2. Daęılım grafięi

Bu çalıřmada çok boyutlu ölçekleme analizi için SPSS paket programı kullanılmıřtır. SPSS' te ÇBÖ analizi yapılırken benzemezlik/uzaklık matrisi kullanmak gerekmektedir. Kruskal ve Wish' e göre eęer toplanan veri benzerlik matrisi ise bunu

benzemezlik matrisine çevirilmesi gerekmektedir. Benzerlik matrisinde değerler büyüdükçe benzerlik artmaktadır, benzemezlik matrisinde ise değerler büyüdükçe benzerlik azalmaktadır. Benzerlik matrisindeki en büyük sabit bir değerden orijinal veriler çıkarılarak benzemezlik matrisine dönüştürülebilir [64, 66, 70, 71].

BÖLÜM 6. UYGULAMA

6.1. İş Analizi

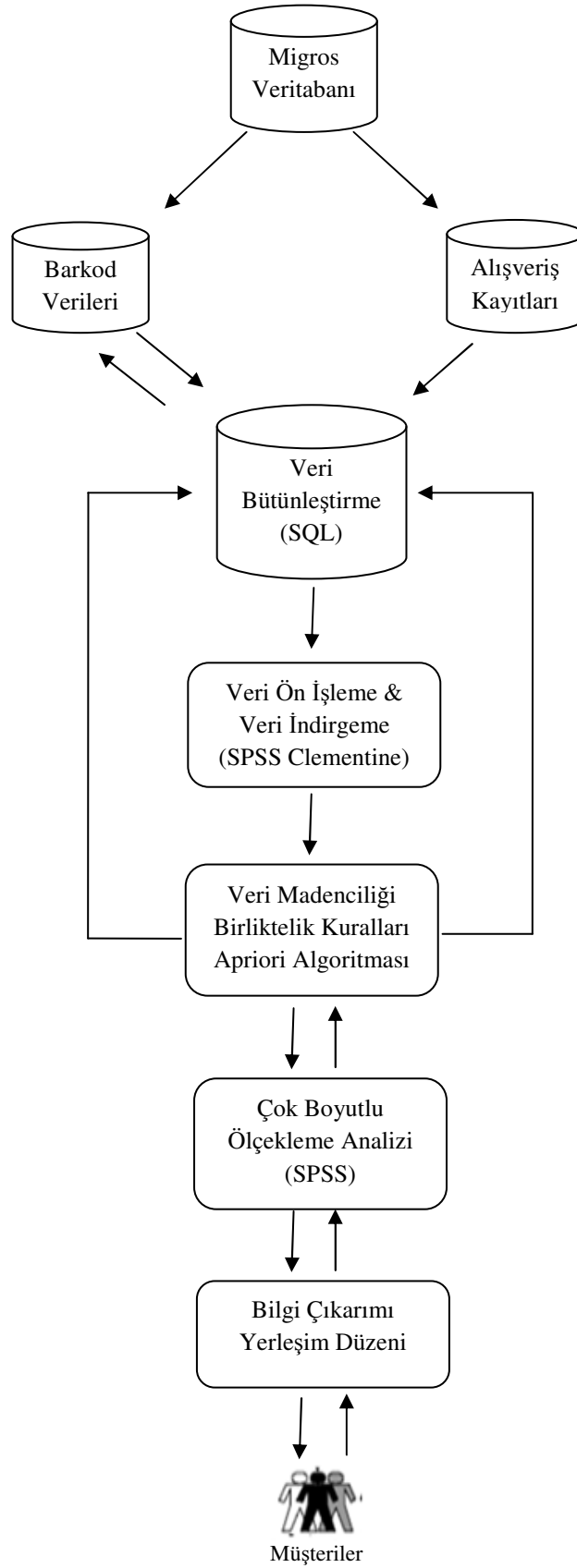
Bu çalışmanın amacı, bir süpermarkete ait alışveriş kayıtlarını kullanarak hangi ürünlerin birlikte satılma eğiliminde olduğu bilgisini vermek ve bu bilgiler doğrultusunda yeni bir yerleşim düzeni önerisi geliştirmektir. Şekil 6.1' de çalışmanın genel kapsamı gösterilmiştir.

Çalışmada üniversitemizin bölümüne ait lisanlı program olan SPSS Inc. şirketinin veri madenciliği için geliştirdiği SPSS Clementine programının Apriori algoritması, veri tabanı olarakta SQL Server 2005 kullanılmıştır.

6.2. Verinin Anlaşılması Ve Modellemeye Hazırlanması

Bu aşamada veriyi tanıma, veri kaynağına bağlanma, veri kalitesini anlama, verinin grafiksel olarak incelenmesi ve analizin yanlış yönlendirilmesine neden olabilecek verilerin temizlenmesi işlemleri yapılmıştır.

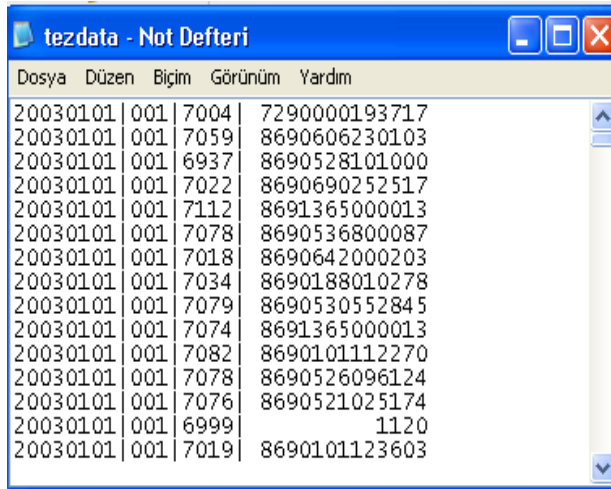
Bu çalışmada, Migros Türk A.Ş. MMM büyüklüğündeki bir mağazaya ait 01.01.2003 ve 31.05.2003 tarihleri arasındaki alışveriş kayıtları kullanılmıştır.



Şekil 6.1. Çalışmanın genel kapsamı

6.2.1. Veri tabanı oluşturma

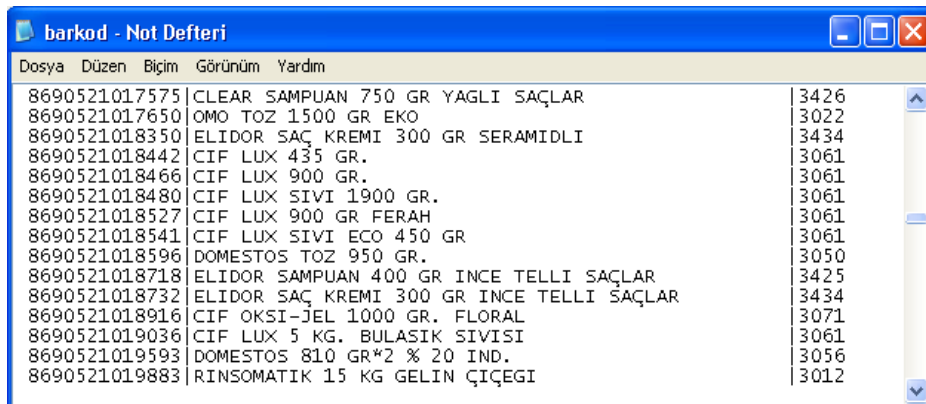
Migros firmasından alınan veriler iki metin dosyası şeklindedir. Birinci dosyada Şekil 6.2’ de gösterilen alışveriş kayıtları, diğerinde ise Şekil 6.3’ te gösterilen ürün verileri bulunmaktadır. Ürün kodları incelendiğinde toplam 12077 ürün ve 558 ürün grubu olduğu gözlenmiştir.



Dosya	Düzen	Biçim	Görünüm	Yardım
20030101	001	7004	7290000193717	
20030101	001	7059	8690606230103	
20030101	001	6937	8690528101000	
20030101	001	7022	8690690252517	
20030101	001	7112	8691365000013	
20030101	001	7078	8690536800087	
20030101	001	7018	8690642000203	
20030101	001	7034	8690188010278	
20030101	001	7079	8690530552845	
20030101	001	7074	8691365000013	
20030101	001	7082	8690101112270	
20030101	001	7078	8690526096124	
20030101	001	7076	8690521025174	
20030101	001	6999	1120	
20030101	001	7019	8690101123603	

Şekil 6.2. Alışveriş kayıtları

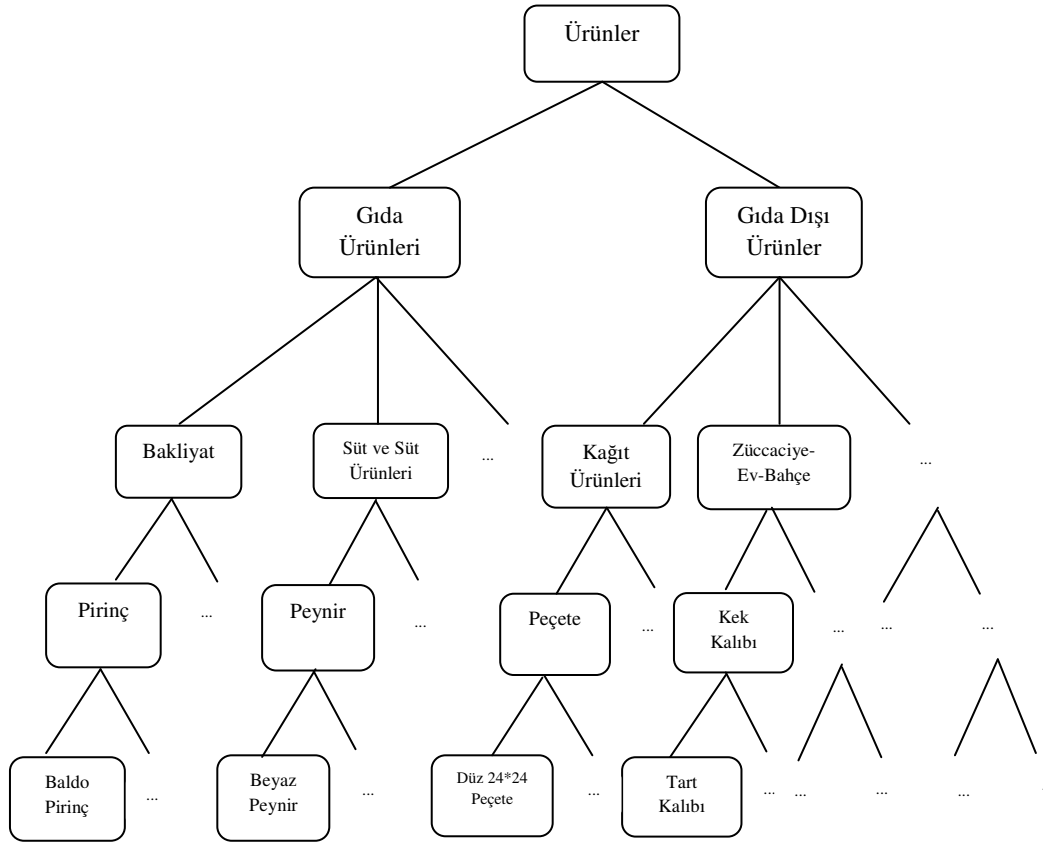
Şekil 6.2’ de gösterilen alışveriş kayıtların metin dosyasındaki alanlar sırasıyla; alışveriş tarihi, kasa no, fiş no ve barkod kodu ifade etmektedir. Şekil 6.3’ te gösterilen ürün verileri metin dosyasındaki alanlar sırasıyla; barkod kod, ürün adı ve grup kodu ifade etmektedir.



Dosya	Düzen	Biçim	Görünüm	Yardım
8690521017575			CLEAR SAMPUAN 750 GR YAGLI SAÇLAR	3426
8690521017650			OMO TOZ 1500 GR EKO	3022
8690521018350			ELIDOR SAÇ KREMI 300 GR SERAMIDLİ	3434
8690521018442			CIF LUX 435 GR.	3061
8690521018466			CIF LUX 900 GR.	3061
8690521018480			CIF LUX SIVI 1900 GR.	3061
8690521018527			CIF LUX 900 GR FERAH	3061
8690521018541			CIF LUX SIVI ECO 450 GR	3061
8690521018596			DOMESTOS TOZ 950 GR.	3050
8690521018718			ELIDOR SAMPUAN 400 GR INCE TELLI SAÇLAR	3425
8690521018732			ELIDOR SAÇ KREMI 300 GR INCE TELLI SAÇLAR	3434
8690521018916			CIF OKSI-JEL 1000 GR. FLORAL	3071
8690521019036			CIF LUX 5 KG. BULASIK SIVISI	3061
8690521019593			DOMESTOS 810 GR*2 % 20 IND.	3056
8690521019883			RINSOMATİK 15 KG GELİN ÇIÇEĞİ	3012

Şekil 6.3. Ürün verileri

Bu çalışmada birliktelik kurallarından elde edilecek sonuçlar ile geleneksel perakendecilik teorisine göre marketlerde kullanılması etkin olan grid tip market düzeni önerisi geliştirilmiştir. Grid yerleşim düzeninde, daha önceki bölümlerde de bahsedildiği üzere müşteriler alışverişlerini hiyerarşik bir seviyede yapmaktadırlar. Migros firmasından alınan verilerde mevcut olan hiyerarşik yapı son ürün ve alt ürün kategorileri şeklindedir. Veriler detaylı incelenerek toplamda 35 adet ana ürün kategorisi elde edilmiştir. Böylece Şekil 6.4 'te de gösterildiği gibi ana ürün kategorileri-alt ürün kategorileri-son ürün hiyerarşi oluşturulmuştur.



Şekil 6.4. Ürün Hiyerarşisi

Veritabanı yönetim sistemi olarak SQL Server 2005 kullanılmıştır. Veriler öncelikle SQL Server' a aktarılarak tablolar oluşturulmuştur. Oluşturulan tablolar aşağıdaki Şekil 6.5 ve Şekil 6.6' da görüldüğü gibi alışveriş kayıtları, ürün verileri tablolarıdır.

Veri setindeki fiş numarası kasa numarasına bağlı olarak mükerrer olduğundan, bir müşteri numarası tarih, kasa no ve fiş no' dan oluşmaktadır. Yine SQL kodları kullanılarak aynı tablo üzerinde yeni bir alan eklenerek müşteri numaraları oluşturulur. Toplamda 186.883 adet müşteri kaydı elde edilmiştir.

Tarih	Kasa_no	Fis_no	Barkod_kod	Mid
20030102	005	7749	8691330580458	200301020057749
20030102	005	7781	86912440	200301020057781
20030102	005	7834	8690624200300	200301020057834
20030102	005	7889	410188000004	200301020057889
20030102	005	7941	8697409882652	200301020057941
20030102	005	7819	8696368163482	200301020057819
20030102	005	7906	482	200301020057906
20030102	005	7689	4027800019502	200301020057689
20030102	005	7861	40329598	200301020057861
20030102	005	7630	1830	200301020057630
20030102	005	7823	8690451065677	200301020057823
20030102	005	7772	5000267014005	200301020057772
20030102	005	7946	1199	200301020057946
20030102	005	7918	88629317969	200301020057918
20030102	005	7854	8691365000013	200301020057854
20030102	005	7735	410186000006	200301020057735
20030102	005	7779	8692095140000	200301020057779
20030102	005	7921	2800176000006	200301020057921
20030102	005	7804	1427	200301020057804

Şekil 6.5. SQL veritabanına aktarılmış alışveriş kayıtları tablosu

Table - dbo.barkod_data		Summary	
	Barkod_kod	Urun_adi	Grup_kod
▶	31	ARMUT DEVEC...	2702
	48	ARMUT SAN. ...	2702
	86	AYVA (A) KG.	2703
	116	CEVİZ HİNDİS...	0601
	123	ÇİLEK SERA KG.	2708
	147	HURMA KG.	2717
	154	ÇAĞLA BADEM...	2707
	178	ERİK CAN KG.	2711
	185	ERİK PAPAZ KG.	2711
	192	ERİK İTALYAN ...	2711
	208	GREYFURT FIL...	2716
	239	ERİK MALTA KG.	2712
	246	ELMA AMASYA...	2713
	253	ELMA GRANNY...	2713
	277	ELMA ARAP KI...	2713
	284	ELMA STARKİN...	2713
	291	ELMA GOLDEN ...	2713
	314	FISTIK ANTEP ...	2715

Şekil 6.6. SQL veritabanına aktarılmış ürün verileri tablosu

Table - dbo.alisveris_data		Summary			
	Tarih	Kasa_no	Fis_no	Barkod_kod	Mid
▶	20030102	005	7749	8691330580458	200301020057749
	20030102	005	7781	86912440	200301020057781
	20030102	005	7834	8690624200300	200301020057834
	20030102	005	7889	410188000004	200301020057889
	20030102	005	7941	8697409882652	200301020057941
	20030102	005	7819	8696368163482	200301020057819
	20030102	005	7906	482	200301020057906
	20030102	005	7689	4027800019502	200301020057689
	20030102	005	7861	40329598	200301020057861
	20030102	005	7630	1830	200301020057630
	20030102	005	7823	8690451065677	200301020057823
	20030102	005	7772	5000267014005	200301020057772
	20030102	005	7946	1199	200301020057946
	20030102	005	7918	88629317969	200301020057918
	20030102	005	7854	8691365000013	200301020057854
	20030102	005	7735	410186000006	200301020057735

Şekil 6.7. Müşteri numarası tarih, kasa_no ve fiş_no' dan oluşmaktadır

Table - dbo.urun_data		Summary		
	Barkod_kod	Urun_adi	Grup_kod	Anagrup_kod
	86900218	DİMES KUŞBURN...	0805	11
	86900225	DİMES PORTAKAL...	0805	11
	86900232	DİMES ELMA SUY...	0805	11
	86900362	DARDANEL TON 8...	0905	14
	86900454	ETİ MOZAIK DİLİ...	0509	19
	86900478	ETİ CİN 9314	0701	19
	86900591	VERNEL 2 KG. AR...	3058	23
	86900621	VERNEL 2 KG. TA...	3058	23
	86900638	VERNEL 6 LT GÜL...	3058	23
	86900669	PRIL 1900 GR.	3061	23
	86900690	DIXI 900 GR. BIO...	3071	23
	86900720	YAYLA MARGARİ...	0420	35
	86900881	ÜLKER BİTTER BA...	0703	19
	86902304	KARPER ÜÇGEN P...	1025	5
	86902328	KARPER ÜÇGEN P...	1025	5
	86902335	KARPER ÜÇGEN P...	1025	5
	86902359	KARPER ÜÇGEN P...	1025	5
	86902397	KARS KUTU ÜÇGE...	1025	5
	86902588	DANONE DOĞAL ...	1101	5
	86902595	DANONE MUZLU ...	1101	5
	86902618	DANONE ÇİLEKLİ ...	1101	5
	86902625	DANONE KAKAOL...	1101	5

Şekil 6.8. Ana ürün kategorilerine göre ürün verileri

SPSS Clementine’ da Apriori algoritmasını çalıştırabilmek için verilerin Şekil 6.9’ da görüldüğü gibi tabular formata getirilmesi gerekmektedir. Bu tablo ürün tablosu ile alışveriş kayıtları tablosundan elde edilecektir. Son olarak SQL kodları kullanılarak satırlarda müşteri numaralarının sütunlarda ise ana ürün kategorileri kodlarının bulunduğu SPSS Clementine’ da kullanılacak olan tabular formattaki kayıtlar oluşturulur.

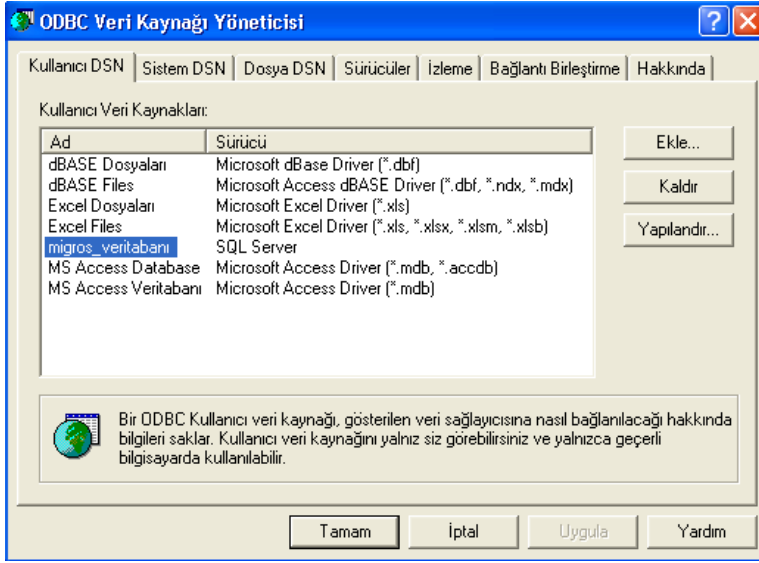
Table - dbo.tabular_data		Summary				
Mid	14	13	32	31	30	
200301010016953	NULL	NULL	NULL	NULL	NULL	
200301010016954	NULL	NULL	NULL	NULL	NULL	
200301010016955	NULL	NULL	NULL	NULL	NULL	
200301010016958	NULL	NULL	NULL	NULL	NULL	
200301010016959	NULL	NULL	1	NULL	NULL	
200301010016960	1	NULL	NULL	NULL	NULL	
200301010016963	NULL	NULL	NULL	NULL	NULL	
200301010016964	NULL	NULL	NULL	NULL	1	
200301010016969	NULL	NULL	NULL	NULL	1	
200301010016972	NULL	NULL	NULL	NULL	NULL	
200301010016973	NULL	NULL	NULL	NULL	NULL	
200301010016974	NULL	NULL	NULL	1	NULL	
200301010016975	NULL	NULL	NULL	2	NULL	
200301010016978	NULL	NULL	NULL	NULL	NULL	
200301010016979	NULL	NULL	NULL	NULL	NULL	
200301010016981	NULL	NULL	NULL	NULL	NULL	
200301010016983	NULL	NULL	NULL	1	NULL	
200301010016984	NULL	NULL	NULL	NULL	NULL	
200301010016987	NULL	NULL	NULL	NULL	NULL	
200301010016989	NULL	NULL	NULL	NULL	NULL	
200301010016990	NULL	NULL	NULL	NULL	NULL	

Şekil 6.9. Özet tablo

Veriler SQL’ de özet tablo şekline dönüştürüldükten sonra SPSS Clementine’ da SQL veri tabanına bağlanma işlemleri yapılarak veri kaynağına erişim sağlanır ve veri temizleme işlemleri yapılır.

6.2.2. Clementine programında verinin hazırlanması

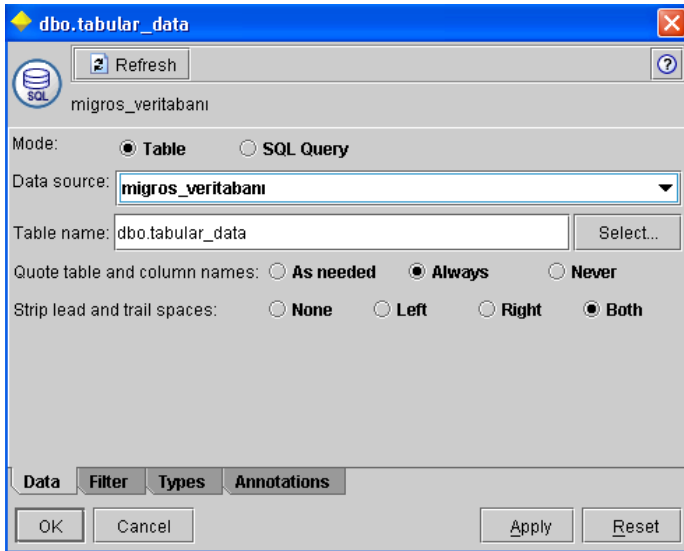
Clementine’ da veri kaynağına erişim sağlanması için öncelikle sırasıyla denetim masası-yönetimsel araçlar-veri kaynakları (ODBC) menüsü takip edilerek SQL’ de kullanılan veritabanının eklenmesi gerekmektedir. Şekil 6.10’ da migros veritabanının eklenmiş hali gösterilmiştir.



Şekil 6.10. SQL Veritabanının veri kaynağı yöneticisine eklenmesi

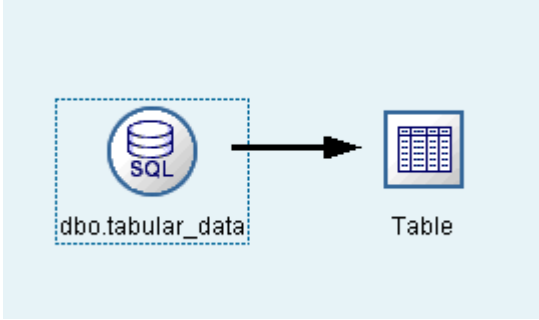
Bu tanımlama yapıldıktan sonra Clementine’ da migros_veritabanına erişim sağlanabilir ve verinin modellemeye hazırlanması işlemlerine başlanabilir.

Clementine’ da SQL veritabanına database düğümü kullanılarak erişilmektedir. Modelleme alanına bir database düğümü yerleştirildikten sonra düğüm sağ tıklanıp, Şekil 6.11’ de gösterildiği gibi veri kaynağına migros_veritabanı ve tablo olarak da dbo.tabular_data tablosu eklenir.



Şekil 6.11. Clementine’ da database düğümü ile migros_veritabanına yapılan erişim

Veri kaynağına yapılan erişimin kontrolü için Şekil 6.12’ de görüldüğü gibi database düğümüne bir table düğümü eklenerek çalıştırılır. Şekil 6.13’ teki gibi veriler görüntülenebiliyorsa, veri kaynağına erişim sağlanmış olur.



Şekil 6.12. Clementine’ da database düğümüne table düğümünün eklenmesi

The screenshot shows the Clementine software interface displaying a table with 36 fields and 186,883 records. The table is displayed in a grid format with columns for 'Mid' and various numerical fields. The 'Mid' column contains values ranging from 200301010016931 to 200301010016972. The numerical fields contain values such as 14, 13, 32, 31, 30, 6, 15, 29, 3, 28, 7, 18, and 32. The table is displayed in a grid format with columns for 'Mid' and various numerical fields. The 'Mid' column contains values ranging from 200301010016931 to 200301010016972. The numerical fields contain values such as 14, 13, 32, 31, 30, 6, 15, 29, 3, 28, 7, 18, and 32.

	Mid	14	13	32	31	30	6	15	29	3	28	7	18	32
1	200301010016931	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
2	200301010016932	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
3	200301010016938	\$...	\$...	\$...	1	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
4	200301010016940	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
5	200301010016944	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
6	200301010016947	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
7	200301010016948	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
8	200301010016949	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
9	200301010016950	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
10	200301010016952	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
11	200301010016953	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
12	200301010016954	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
13	200301010016955	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
14	200301010016958	\$...	\$...	\$...	\$...	\$...	\$...	\$...	4	\$...	\$...	1	1	\$...
15	200301010016959	\$...	\$...	1	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
16	200301010016960	1	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
17	200301010016963	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
18	200301010016964	\$...	\$...	\$...	\$...	1	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
19	200301010016969	\$...	\$...	\$...	\$...	1	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...
20	200301010016972	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...	\$...

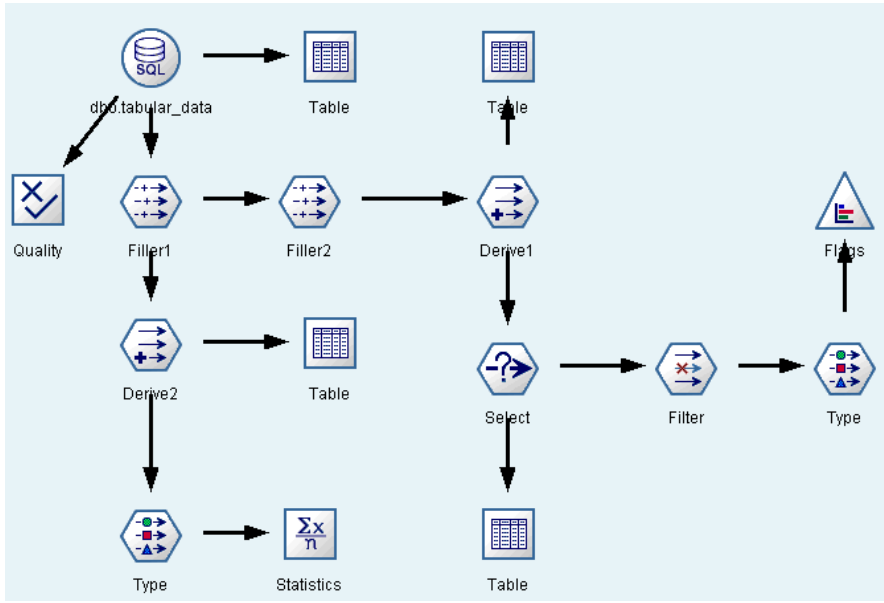
Şekil 6.13. Verilerin Clementine’ da görüntülenmesi

SQL’ de oluşturulan özet tabloda, müşterilerin almadığı ürünlerin değerlerinin “null” olduğu görülmektedir (Bkz. Şekil 6.9). Null değerlerin sıfır olması gerektiğinden Clementine modelleme alanında filler işlemcisi ile null değerler sıfıra dönüştürülür.

Ayrıca sadece bir tane ürün alan müşteriler birliktelik analizinde herhangi bir fayda sağlamayacağından hatta kayıt sayısını arttırarak sık geçen nesnekümelere destek değerini düşüreceğinden veri setinden çıkarılır. Son olarak verilerin boolean tipte olması gerektiğinden bir müşterinin bir ürünü satın aldığı durumlar “1” olarak değiştirilmiştir.

Clementine’ da veri kaynağına erişim ve veri temizleme işlemlerinin yapılmasının sonucunda modelleme alanında elde edilen görüntü Şekil 6.14’ te görüldüğü gibidir. Yapılan işlemler aşağıdaki maddelerde özetlenmiştir.

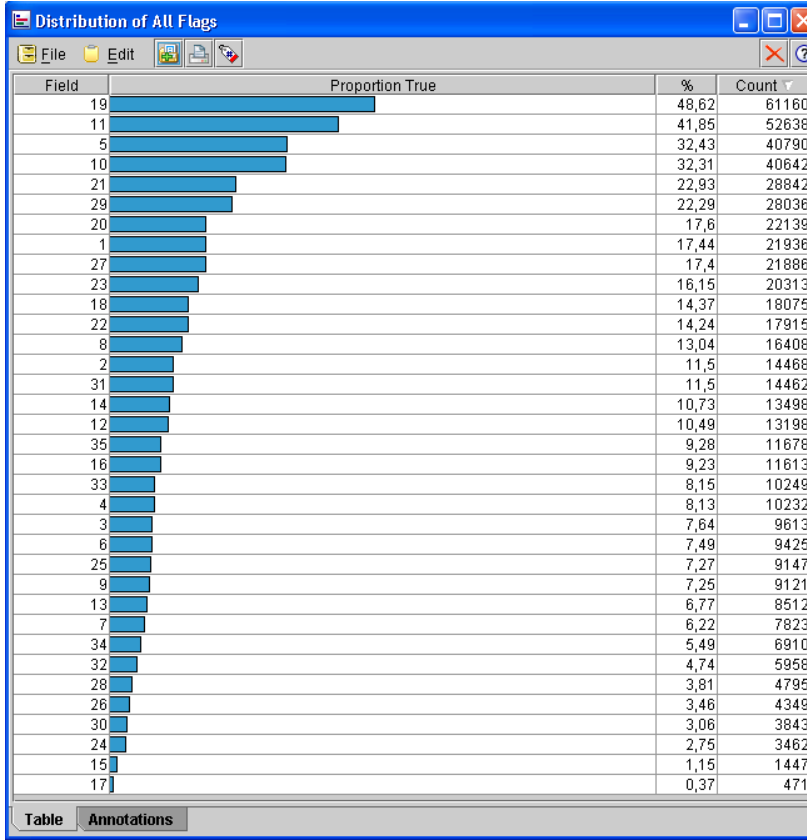
- Quality düğümü verinin kalitesi hakkında bilgi vermektedir. Şekil 6.15’ te gösterildiği gibidir. Örneğin, mid alanında hiç boş/null değer yoktur, tamamlanma oranı %100’ dür.
- Filler1 düğümü ile “nul” değerler sıfıra dönüştürülmüştür.
- Filler2 düğümü ile 1’ den büyük değerler 1 olarak değiştirilmiştir.
- Derive1, select ve filter düğümü ile sadece bir tane ürünü alan müşteriler çıkarılmıştır.
- Type düğümü ile verilerin tip ve yönleri belirlenmiştir.
- Flags düğümü sadece bir ürün alan müşteriler elendikten sonra ürün hakkında bilgi veren grafik, Şekil 6.16’ da gösterilmiştir.,
- Derive2 düğümü bir müşterinin alışveriş toplamını göstermektedir.
- Statistics düğümü, Şekil 6.17’ de gösterilen tanımlayıcı istatistikleri elde etmeyi sağlamaktadır.



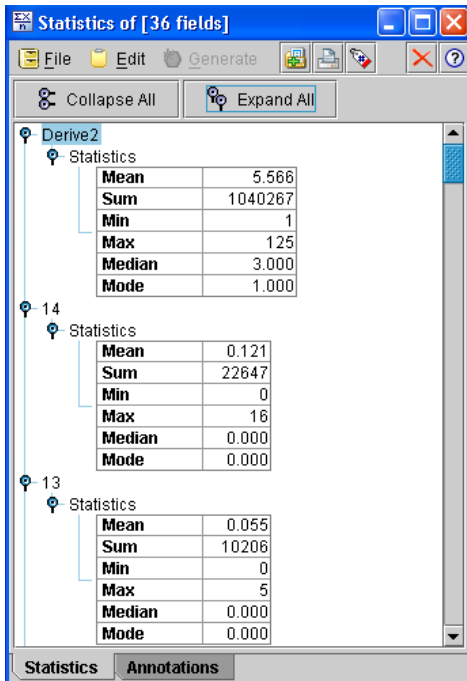
Şekil 6.14. Clementine modelleme alanının görüntüsü

Field	% Complete	Valid Records
Mid	100	186883
9	5,04	9417
8	8,96	16742
7	4,25	7946
6	5,23	9772
5	22,78	42567
4	5,54	10349
35	6,36	11888
34	4,6	8590
33	6,6	12343
32	3,72	6943
31	8,57	16019
30	2,56	4784
3	5,31	9918
29	16,56	30939
28	2,84	5314
27	12,21	22823
26	2,63	4922
25	4,96	9274
24	1,88	3514
23	11,58	21641
22	10,13	18924
21	16,36	30578
20	12,69	23724
2	11,73	21918
19	38,13	71253
18	10,03	18743
17	1,13	2104
16	6,45	12055
15	0,79	1482
14	7,44	13899
13	4,8	8967
12	7,48	13983
11	33,4	62416
10	21,75	40642
1	15,92	29756

Şekil 6.15. Veri kalitesini gösteren (Quality) düğüm çıktısı



Şekil 6.16. Ürün kategorilerin dağılımı



Şekil 6.17. Tanımlayıcı istatistikler

Veri seti hakkında, statistics düğümü ile elde edilen temel istatistik bilgileri Tablo 6.1.a ve Tablo 6.1.b' de özetlendiği gibidir.

Tablo 6.1.a. Tanımlayıcı istatistikler

	Fişlerdeki Hareket Sayıları
Toplam	1.047.267
Ortalama	6
Medyan	3
Mod	1
Maksimum	125
Minimum	1

Tablo 6.1.b. En sık satan 6 ürün

Ana Ürün Kategorisi	Satış miktarı (kere)	%
19	152.122	14,62
11	100.461	9,66
21	85.153	8,19
5	77.702	7,42
29	60.574	5,82
10	48.960	4,71

6.3. Modelleme

Bu çalışmada birliktelik kuralları algoritmalarından bilinen en temel algoritma olan Apriori algoritmasından yararlanılmıştır. Yapılan çalışmalar sonucunda Apriori algoritması, minimum destek değeri %0.5, güven değeri %20 ve öncül sayısı 1 belirlenerek çalıştırılmıştır. Öncül sayısının 1 olmasının amacı sadece ikili ürünler arasındaki ilişkileri görebilmektir. Sonuç olarak Şekil 6.18' de gösterildiği gibi toplamda 518 tane kural oluşmuştur (Bkz. EK 1). Şekil 6.18' deki consequent kuralın sonucunu, antecedent kuralın öncülünü, confidence kuralın güven değerini, rule support ise kuralın destek değerini ifade etmektedir. Şekil 6.19' de ise Clementine' da kurulan modelin son aşamasındaki görüntüsü verilmiştir. Modelin çalıştırılması sonucunda elde edilen kurallardan bazıları Tablo 6.2' de verilmiştir, devam eden maddelerde de kuralların anlamları açıklanmıştır.

35 fields

File Generate

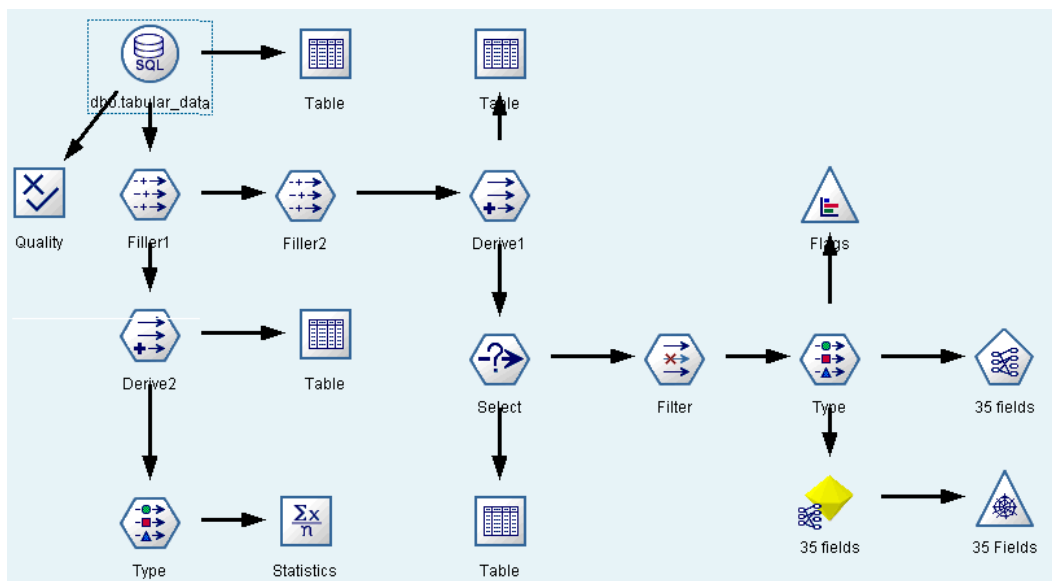
Sort by: Confidence % 518 of 518

Consequent	Antecedent	Confidence %	Rule Support %
5	25	76.703	5.578
5	8	70.612	9.211
5	35	70.577	6.552
5	24	70.566	1.942
19	9	68.918	4.997
5	15	68.417	0.787
5	9	67.986	4.930
5	18	67.386	9.683
5	4	67.064	5.455
21	7	65.985	4.104
5	16	65.168	6.016
5	3	63.820	4.877
5	7	63.799	3.968
19	20	63.517	11.179
5	14	63.283	6.791
19	24	63.229	1.740
19	4	62.627	5.094
21	22	62.529	8.905
5	22	61.596	8.773
19	15	61.507	0.708
19	35	61.346	5.695
19	8	61.330	8.000
19	11	61.199	25.610

Model Settings Summary Annotations

OK Cancel Apply Reset

Şekil 6.18. Birliktelik kuralları



Şekil 6.19. Clementine' da kurulan modelin görüntüsü

Tablo 6.2. Birliktelik Kuralları

Öncül	Sonuç	Güven%	Destek%
25	5	77	5,58
9	19	69	5,00
8	19	61	8,00
16	21	56	5,20
26	11	56	1,94

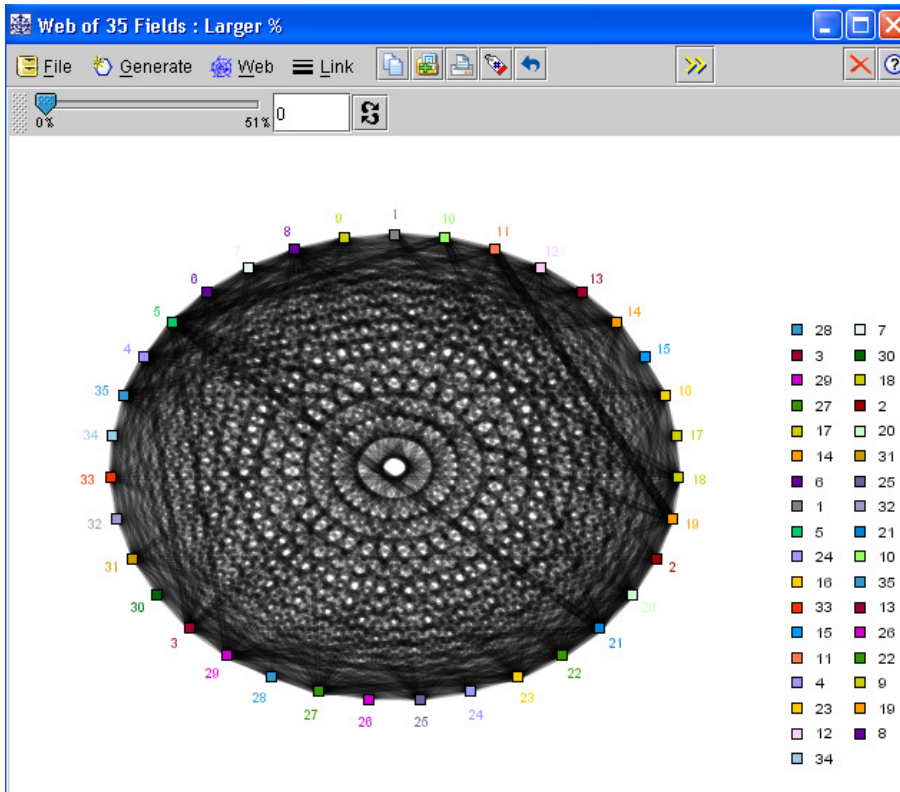
- 25 ürününü alan müşteriler %77 olasılıkla 5 ürününü de almaktadır. Bu ürünlerin veri setindeki işlemlerde birlikte bulunma olasılığı %5,58' tir.
- 9 ürününü alan müşteriler %69 olasılıkla 19 ürününü de almaktadır. Bu ürünlerin veri setindeki işlemlerde birlikte bulunma olasılığı %5,00' tir.
- 8 ürününü alan müşteriler %61 olasılıkla 19 ürününü de almaktadır. Bu ürünlerin veri setindeki işlemlerde birlikte bulunma olasılığı %8,00' tir.
- 16 ürününü alan müşteriler %56 olasılıkla 21 ürününü de almaktadır. Bu ürünlerin veri setindeki işlemlerde birlikte bulunma olasılığı %5,20' tir.
- 26 ürününü alan müşteriler %56 olasılıkla 11 ürününü de almaktadır. Bu ürünlerin veri setindeki işlemlerde birlikte bulunma olasılığı %1,94' tir.

Birliktelik kurallarının nesnelere arasındaki ilişkileri gösterdiği daha önceki bölümlerde ifade edilmişti. Ürünler arasındaki ilişkileri görebilmek için Clementine' da web grafiğinden yararlanılmıştır. Web grafiğindeki çizgi kalınlığı ilişkinin baskınlığını göstermektedir. Çizgi kalınlığı arttıkça ilişkinin kuvvetliliği de artmaktadır. Şekil 6.20' de çizgi değeri %0 olarak ayarlanmıştır. Bu sayede tüm ürünler arasındaki ilişki görülebilmektedir. Şekil 6.21' de gösterilen grafikte ise çizgi değeri %18 olarak ayarlanmıştır. Bu grafikte aralarındaki ilişki %18' den büyük ve eşit olan ürün kategorileri yer almaktadır. Tablo 6.3' te en güçlü birlikteliklere sahip

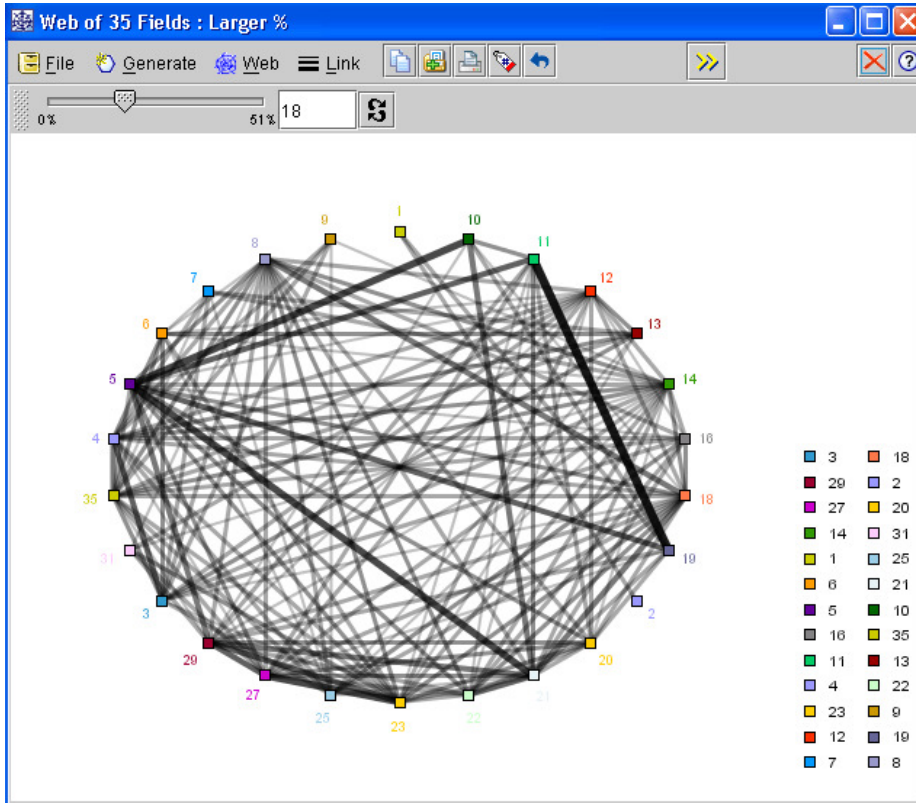
ürünler gösterilmiştir. Örneğin; 11 ve 19 numaralı ürün kategorileri arasındaki ilişki %52,67 ile en güçlü olmaktadır. Ürün kodlarının açıklamaları Ek 2’ de verilmiştir.

Tablo 6.3. En güçlü birliktelikler

Ürün1	Ürün2	Çizgiler %
11	19	52,67
5	21	41,50
23	27	41,35
5	10	40,70
21	22	38,84
27	29	38,66
23	29	38,58
5	11	37,59
5	19	36,00
3	4	35,41



Şekil 6.20. Tüm ürünler arasındaki birliktelik diyagramı (çizgi değeri>=%0)



Şekil 6.21. İlişki seviyesi %18 üzerinde olan ürünlerin birliktelik diyagramı

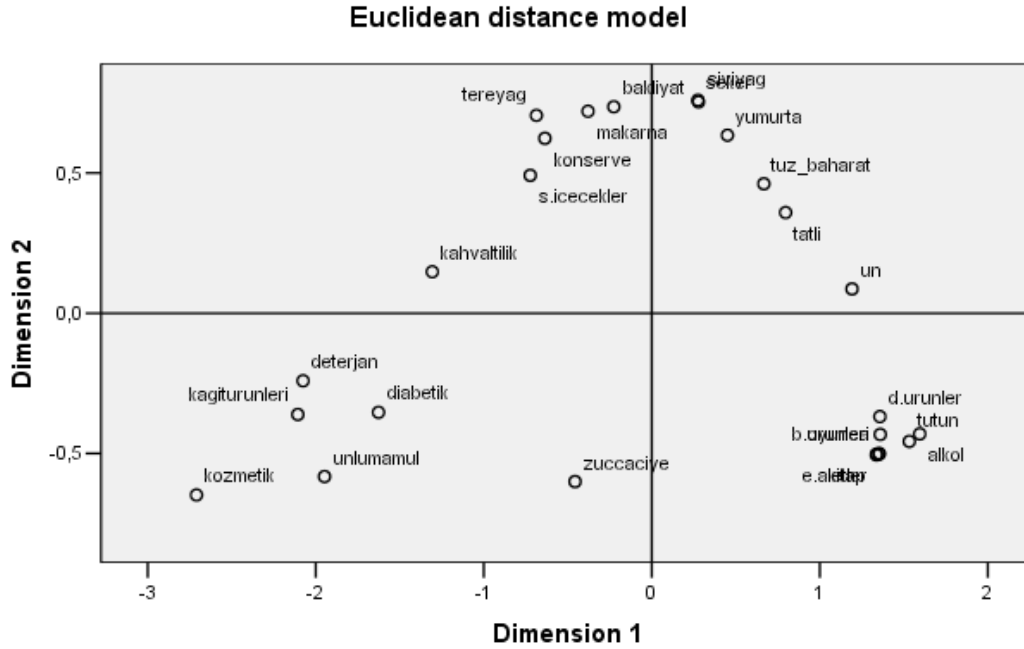
6.4. Çok Boyutlu Ölçkleme Analizi İle Ürün Haritasının Çıkarılması

Birliktelik kuralları sonucunda ürünler arasındaki ilişkiler hakkında bilgiler verilmiştir. Bu bilgilerin nasıl kullanılacağı konusunun önem arz ettiği, market sepet analizi sonuçlarının kampanya, yerleşim düzeni gibi alanlarda kullanıldığından önceki bölümlerde bahsedilmiştir. Bu tez çalışmasının amacı doğrultusunda birliktelik kuralları sonuçları çok boyutlu karar verme metotlarından çok boyutlu ölçkleme analizi kullanılarak yeni bir yerleşim düzeni geliştirilmeye çalışılmıştır. Bu kapsamda, birliktelik kurallarından elde edilen güven değeri ile ürünler arasındaki yakınlık matrisi oluşturulmuştur. Oluşturulan ilişki matrisi (Bkz. EK 3) ile SPSS programında MDS analizi yapılarak ürünlerin iki boyutlu düzlemde grafiksel gösterimleri elde edilmiştir.

MDS analizi sonuçları ve ürün haritası Şekil 6.22' de gösterildiği gibidir. Sebze-meyve, süt ve süt ürünleri, çerez, içecek ürün kategorileri aralarındaki çok güçlü

birliktelikler diğer ürünler arasındaki kümeleşmeyi görmeyi güçleştirdiğinden bu ürün kategorileri analize dahil edilmemiştir.

Derived Stimulus Configuration



Şekil 6.22. ÇBÖ Analizi ile elde edilen ürün haritası

Çok boyutlu ölçekleme analiz sonucuna göre stres değeri 0,093 ve kareli korelasyon katsayısının (R^2) değeri 0,96 olarak elde edilmiştir. Sonuçlar anlamlı ve kabul edilebilirdir. Şekil 6.22’ de görüldüğü gibi belirgin olarak dört küme oluşmuştur. Birinci küme; bakliyat, makarna, konserve, sıcak içecekler, kahvaltılık ürünlerinden oluşmaktadır. Birinci kümeye en yakın olan ikinci kümede temel gıda ürünleri olan sıvıyağ, şeker, yumurta, tuz-baharat-harç, un ve tatlı/hamur işleri ürün kategorileri bulunmaktadır. Üçüncü küme, kâğıt ürünleri, deterjan ve temizlik ürünleri, kozmetik ve cilt bakım ürünleri, unlu mamüller ve diyabetik ürünlerinden oluşmaktadır. Dördüncü kümede ise alkollü içecekler, tütün, kitap-dergi-kırtasiye, oyuncak-pet ürünleri-hobi, tekstil ve bebek ürünleri yer almaktadır.

6.5. Değerlendirme Ve Modelinin Geçerliliği

Birliktelik kuralları madenciliği sonucunda ürün kategorileri arasındaki ilişki diyagramı (Bkz. Şekil 6.20) ve en baskın birlikteliklere sahip ürünler (Bkz. Tablo 6.3) tablosuna bakıldığında içecekler, çerez, süt ürünleri, sebze, unlu mamüller, deterjan ve temizlik ürünleri, kağıt temizlik ürünleri, kozmetik ürün kategorileri arasındaki ilişkilerin en güçlü birliktelikler olduğu görülmektedir. Satış istatistiklerine bakıldığında (Bkz. Tablo 6.1.b) ise bu ürünlerin en çok satan ürünler listesi başında geldiği görülmektedir.

Migros mağazasının mevcut yerleşim düzeni incelendiğinde en sık satılan ürünlerin birbirine yakın yerlerde olduğu gözlenmiştir. Ek 4' te verilen mevcut yerleşim düzeninde, kırmızı çizgili alan genel olarak müşterilerin alışveriş yaptığı bölgeyi ifade etmektedir. Müşterinin satınalma davranışlarını etkilemeyi, alışveriş zamanını ve satınalmaları arttırmayı hedefleyen yeni yerleşim düzeni de Ek 5' te gösterilmiştir. Mevcut yerleşim düzenindeki hatalar ve yapılan değişiklikler devam eden maddelerde anlatıldığı gibidir.

- Öncelikle mağazanın kasalarının yerleri değiştirilmiştir. Mevcut düzende kasalar mağazanın uzun kenarına paralel şekilde yerleştirilmiştir. Bu yerleşim mağazanın orta alanının kullanımını kısıtladığından (reyonlar sadece birbirine paralel iki sıra şeklinde), kasalar mağazanın kısa kenarına paralel olacak şekilde yerleştirilmiştir. Böylelikle mağazanın arka duvarına doğru paralel reyon sayısı üçe çıkarılarak ürün kümeleri arasındaki ilişkilerin daha iyi gösterilmesi sağlanmıştır.
- Mağaza içi yerleşim düzeni kurallarına göre; market girişinin sağ tarafına sebze-meyve reyonları yer almalıdır. Bu yüzden mağaza yerleşimine ilk olarak sebze-meyve reyonu ile başlanmıştır.
- Sebze ürün kategorisi ile arasında en güçlü birlikteliğe sahip ürün grubu olan süt ve süt ürünleri; meyve-sebze reyonunun yanında yer almaktadır.

- Süt ve süt ürünleri dolapta teşhir edilen ürün grubu olduğundan devamında yine dolapta teşhir edilmesi gereken tereyağ-margarin, et/piliç, şarküteri ürün grubu yerleştirilmiştir.
- Süt ve süt ürünlerinden sonra aralarında güçlü ilişki bulunan unlu mamüller, içecek ve çerez ürün grupları yerleştirilmiştir. Bu ürünlerinin satış sıklığı fazla olduğundan müşterilerin bu ürünleri market içerisinde arama ihtimali yüksektir. Bu yüzden bu ürünler mağazanın arka taraflarına yerleştirilmiştir. Böylece müşterinin mağaza içerisinde dolaşması sağlanarak alışveriş zamanı arttırılmaya çalışılmıştır. Mevcut yerleşimde içecekler ve çerez ürün grubu hemen mağaza girişinde bulunmaktaydı. İçecekler ve çerez ürünü grubu arasındaki birliktelik yaklaşık %53 ile en güçlü olan ilişkidir. Bu ürün gruplarının mağaza girişinde kasa yakınında ve yan yana olmaları müşterilerin başka ürün almaya yönlendirmeden alışverişlerini sonlandırmasına neden olabileceğinden bu ürünlerin marketin arka taraflarında ve süt ürünlerinden sonra teşhir edilmesi fayda sağlayacaktır.
- İçecekler, çerez, süt ve süt ürünlerinin arasına üçüncü kümedeki ürünler yerleştirilmiştir. Daha önceden de ifade edildiği gibi, bu ürün kategorileri arasındaki birliktelikler en güçlü birlikteliklerdir. Mevcut yerleşimde bu ürün grupları birbirlerine ters yöndedirler.
- Marketin orta sıradaki reyonlarına birbirleri arasındaki ilişkilere göre birinci ve ikinci kümedeki ürünler yerleştirilmiştir.
- Mağaza yerleşim düzeni kurallarına göre hacimli ürünlerin kasa çıkışına yakın yerde olması gerektiğinden elektrikli ve elektronik ev aletleri, züccaciye-ev-bahçe ürünleri marketin en sonuna yerleştirilmiştir. Mevcut yerleşim düzeninde bu ürün grubu dağınık parçalar halinde mağazanın çeşitli kısımlarında yer almaktaydı.
- Son olarak da dördüncü kümeyi oluşturan ürünler mağaza girişinin önündeki reyonlara yerleştirilmiştir. Bilindiği gibi mağaza girişinde bulunan ürünlerin arka

taraftaki ürünlere göre satılma ihtimali daha yüksektir. Mağazanın arka tarafına yerleştirilen ürünler temel ihtiyaç ürünleri ve satış sıklığı yüksek olan ürünlerdir olduğundan ön tarafa yerleştirilen gıda dışı bu ürünlerin değerinin artması beklenmektedir.

Sektördeki önde gelen diğer firmaların yerleşim düzeni örnekleri incelendiğinde, genellikle mağaza girişinin sağ tarafından meyve-sebze ürünleri ile yerleşime başlanıp devamında dolapta teşhir edilen süt ve süt ürünleri, et/piliç, şarküteri ürünlerinin yer aldığı, içecekler ve hacimli/kırılabilir ürünler olan elektronik-züccaciye ürünleri ile mağazanın duvar kenarındaki yerleşimin tamamlandığı görülmüştür. Mağazanın orta alanlarında ise temel besin gıdaları ve temizlik/cilt bakım ürünleri; ön taraflarda ise ürün değeri arttırılmak istenen gıda dışı ürünlerin teşhir edildiği yapılan gözlemlerde elde edilen diğer sonuçlardır. Böylelikle yapılan analizlerin sonuçları değerlendirilerek uzman görüşüne dayalı önerilen yerleşim düzeninin gerçeğe uygunluğu da kanıtlanmıştır.

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Gelişen teknoloji ile birlikte kullanım alanları artmakta olan veritabanlarında çok büyük miktarlarda veriler toplanmaktadır. Bu verilerin analizi, yorumlanması, karar vermede kullanılması veri madenciliği kullanımının geniş bir alana yayılmasını sağlamaktadır.

Veri madenciliğindeki amaç çok büyük miktarlardaki veriden değerli bilginin elde edilmesidir. Değerli bilginin keşfi için birçok veri madenciliği tekniği bulunmaktadır. Bu çalışmanın araştırma konusu olan birliktelik kuralları tekniği en yaygın veri madenciliği tekniklerinden biridir.

Birliktelik kuralları, büyük miktarlardaki nesne verileri arasından birliktelikleri keşfetmeyi amaçlar. Başka bir ifade ile birliktelik kuralları, aynı işlem içinde birlikte sık görülen nesnelere içeren kurallardır.

Birliktelik kuralları kullanışlı ve anlaşılması kolay olduğundan perakendecilik, finans, pazarlama, online ticaret gibi endüstriyel alanlarda geniş bir kullanım alanına sahiptir. Bu nedenle birliktelik kuralları son yılların en çok araştırılan konularından biri olmuştur.

Birliktelik kuralları konusundaki bilinen en temel ve birçok algoritmanın temelini teşkil eden algoritma Apriori algoritmasıdır. Birliktelik kurallarının kullanıldığı en yaygın alanlardan biri market sepet analizi çalışmalarıdır. Market sepeti analizi, bir alışveriş boyunca satın alınan ürünlerin birleşimini incelemektedir. Market sepet analizi, market yerleşim düzeni kararına yardımcı olmaktadır. Perakendeciler karmaşık müşteri taleplerini karşılayabilmek ve müşterilerin satın alma davranışlarını etkileyebilme çabasıdadırlar. Bundan dolayı market düzeni yerleşimi perakende sektöründeki önemli konular arasındadır.

Bu çalışmada öncelikle veri madenciliği ve ilgili kavramları hakkında temel bilgiler verilmiştir. Bu çalışmanın temelini oluşturan birliktelik kuralları, birliktelik kurallarının temel kavramları, birliktelik kuralları madenciliğinin adımları ve sık geçen nesnekümelerin bulunmasını sağlayan Apriori algoritması detaylı olarak anlatılmıştır. Aynı zamanda sık geçen nesnekümelerin bulunmasını sağlayan diğer bazı algoritmalar hakkında da bilgiler verilmiştir.

Çalışmanın uygulama bölümünde, SPSS Clementine veri madenciliği programının Apriori algoritması aracı ile gerçek veriler üzerinde birliktelik kuralları madenciliği yapılmıştır. Birliktelik kuralları madenciliğinden elde edilen kurallar sonucunda birlikte satılma eğilimi gösteren ürünler ve ürünler arasındaki ilişkiler hakkında bilgiler verilmiştir. Ürünler arasındaki ilişkiler, çok boyutlu karar verme metotlarından çok boyutlu ölçekleme analizi ile değerlendirilerek iki boyutlu düzlemde ürün haritası elde edilmiştir. Birliktelik kuralları ve çok boyutlu ölçekleme analizi sonuçları ışığında firma için yeni bir yerleşim düzeni önerilmiştir. Böylece amaçlandığı üzere önerilen yerleşim düzeni ile müşteri davranışlarının etkilenmesi sağlanacaktır.

Elde edilen deneyimler ışığında bu çalışmayı daha ileriye götürecek olan araştırma önerileri aşağıdaki maddelerde listelenmiştir.

- Kullanılan verilerin zaman periyodunun genişletilmesi ile mevsimsel analiz yapılarak kurallar oluşturulabilir.
- Mevsimsel kuralların sonuçlarına göre market için ayrı ayrı yerleşim düzeni önerileri geliştirilebilir.
- Müşteri bilgileri ve müşterilerin alışveriş zamanlarının da bulunduğu veriler ile zamana bağlı birliktelik kuralları analizi yapılarak, müşteri bazında kampanyalar yapılabilir.
- Veri madenciliği sürecince uygulanan bu adımlar yapay zeka teknikleri ile bütünleştirilerek, uzman ihtiyacı duymaksızın işlemlerin otomatik olarak yapılması sağlanabilir.

KAYNAKLAR

- [1] SRINIVASA, K.G., VENUGOPAL, K.R., PATNAIK, L.M., A Self-Adaptive Migration Model Genetic Algorithm For Data Mining Applications, Information Sciences, sayı 177, syf. 4295-4313
- [2] FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., From Data Mining To Knowledge Discovery In Databases, AI Magazine, sayı 17, syf. 37-54, 1996
- [3] <http://mail.baskent.edu.tr/~20394676/0302/bil483/HW2.pdf>, 24.04.2008
- [4] www.informatik.uni-freiburg.de/~ml/ecmlpkdd/WS-Proceedings/w04/paper5.pdf, 30.10.2008
- [5] ÇİL, İ., Bilgi Tabanlı İmalat Karar Destek Sistemleri Ve Bir Uygulama, End. Müh. Dergisi, sayı 1, syf. 13, 2002
- [6] ÇİL, İ., ALPTÜRK, O., YAZGAN, H.R., A New Collaborative System Framework Based On A Multiple Perspective Approach: InteliTeam, Decision Support Systems, sayı 39, syf. 619-641, 2005
- [7] AKPINAR, H., Veri Tabanlarında Bilgi Keşfi Ve Veri Madenciliği, İ.Ü. Dergisi, sayı 29, syf. 1-22, 2000
- [8] ALPAYDIN, E., Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, Bilişim 2000 Eğitim Semineri, Boğaziçi Üniversitesi, Bilgisayar Müh. Böl., syf. 1-3, 2000
- [9] <http://web.sakarya.edu.tr/~ademiriz/SakaryaWebSite/dms2006.htm>, 11.04.2008
- [10] HAN, J., KAMBER, M., Data mining: Concepts and Techniques, Morgan Kaufmann Publishers, syf. 105-259, 2006
- [11] NGAI, E.W.T., XIU, L., CHAU, D.C.K., Application Of Data Mining Techniques In Customer Relationship Management: A Literature Review And Classification, Expert Systems with Applications, sayı 2, cilt 36, syf. 2592-2602, 2008

- [12] LU, C., CHEN, C., A Study Of Applying Data Mining Approach To The Information Disclosure For Taiwan's Stock Market Investors, Expert Systems with Applications, sayı 2, cilt 36, syf. 3536-3542, 2008
- [13] HAND, D.J., Data Mining: Statistics and More, The American Statistician, sayı 2, cilt 52, syf. 112-118, 1998
- [14] FEELDERS, A., DANIELS, H., HOLSHEIMER, M., Information and Management, sayı 5, cilt 37, syf. 271-281, 2000
- [15] www.cs.sjsu.edu/faculty/lee/cs157/Archana_Data_Mining.ppt, 16.04.2008
- [16] KEKEÇ, G., Veri Madenciliği Ve Karınca Koloni Optimizasyonu İle Kümeleme, Yüksek Lisans Tezi, SAÜ Fen Bilimleri Enstitüsü, 2005
- [17] <http://eprints.rclis.org/5910/1/173-204.pdf>, 21.04.2007
- [18] KOLAY, G., İşletmelerde Bilgi Sistemleri Verimliliğini Arttırmada Veri Madenciliği Yöntemi: Bir Simülasyon Çalışması, Yüksek Lisans Tezi, ZKÜ, Sosyal Bilimler Enstitüsü, 2006
- [19] DOLGUN, M.Ö., Büyük Alışveriş Merkezleri İçin Veri Madenciliği Uygulamaları, Yüksek Lisans Tezi, HÜ Fen Bilimleri Enstitüsü, 2006
- [20] KOYUNCU, F., Veri Madenciliğinde Apriori Temelli İlişkilendirme Kuralının Uygulama Ve Karşılaştırılması, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, 2004
- [21] <http://web.sakarya.edu.tr/~cagil/e-isletme/VeriAmbarlari.ppt>, 09.04.2008
- [22] <http://www.crisp-dm.org/Overview/index.htm>, 02.01.2008
- [23] Clementine giriş eğitim notu, SPSS Türkiye, 2004
- [24] <http://www.crisp-dm.org/CRISPWP-0800.pdf>, 02.01.2008
- [25] http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=538, 25.02.2008
- [26] DÖŞLÜ, A., Veri Madenciliğinde Market Sepet Analizi Ve Birliktelik Kurallarının Belirlenmesi, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, 2008
- [27] ÖZÇINAR, H., KPSS Sonuçlarının Veri Madenciliği Yöntemleriyle Tahmin Edilmesi, Yüksek Lisans Tezi, PÜ Fen Bilimleri Enstitüsü, 2006
- [28] AKBULUT, S., Veri Madenciliği Teknikleri İle Bir Kozmetik Markanın Ayrılan Müşteri Analizi Ve Müşteri Segmentasyonu, Yüksek Lisans Tezi, GÜ Fen Bilimleri Enstitüsü, 2006

- [29] CHEN, Y-L., CHEN, J-M., TUNG, C-W., A Data Mining Approach For Retail Knowledge Discovery With Consideration Of The Effect Of Shelf-Space Adjacency On Sales, Decisions Support Systems, sayı 3, cilt 42, syf. 1503- 1520, 2006
- [30] http://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1, 31.03.2008
- [31] http://tr.wikipedia.org/wiki/Genetik_algoritmalar, 31.03.2008
- [32] YURTSEVER, U., Veri Madenciliği Ve Uygulaması, Yüksek Lisans Tezi, SAÜ Sosyal Bilimler Enstitüsü, 2002
- [33] AYDOĞAN, F., E-ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı Ve Gerçekleştirimi, Yüksek Lisans Tezi, HÜ Fen Bilimleri Enstitüsü, 2003
- [34] HAN, J., FU, Y., Discovery Of Multiple-Level Association Rules From Large Databases, Proceedings of the 21st VLDB Conference, Zurich, syf. 420-431, 1995
- [35] LAZCORRETA, E., BOTELLA, F., CABALLERO, A., Towards Personalized Recommendation By Two-Step Modified Apriori Data Mining Algorithm, Expert Systems With Applications, sayı 3, cilt 35, syf. 1422-1429, 2008
- [36] LEE, A.J.T., WANG, C-S., WENG, W-Y., CHEN, Y-A., WU, H-W., An Efficient Algorithm For Mining Closed Inter-Transaction Itemsets, Data&Knowledge Engineering, sayı 1, cilt 66, syf. 68-91, 2008
- [37] LEE, Y-C., HONG, T-P., WANG, T-C., Multi-level Fuzzy Mining With Multiple Minimum Supports, Expert Systems with Applications, sayı 1, cilt 34, syf. 459-468, 2008
- [38] SOHN, S.Y., KIM, Y., Searching Customer Patterns Of Mobile Service Using Clustering And Quantitative Association Rule, Expert Systems with Applications, sayı 2, cilt 34, syf. 1070-1077, 2008
- [39] YAN, X., ZHANG, C., ZHANG, S., Genetic Algorithm-Based Strategy For Identifying Association Rules Without Specifying Actual Minimum Support, Expert Systems with Applications, sayı 2, cilt 36, syf. 3066-3076, 2008
- [40] KARABATAK, M., İNCE, M.C., An Expert System For Detection Of Breast Cancer Based On Association Rules And Neural Network, Expert Systems with Applications, sayı 2, cilt 36, syf. 3465-3469, 2009
- [41] LI, Y., NING, P., WANG, X.S., JAJODIA, S., Discovering Calender-Based Temporal Association Rules, Data&Knowledge Engineering, sayı 2, cilt 44, syf. 193-218, 2003

- [42] TSENG, M-C., LIN, W-Y., Efficient Mining Of Generalized Association Rules With Non-Uniform Minimum Support, Data&Knowledge Engineering, sayı 1, cilt 62, syf. 68-91, 2007
- [43] CHEN, M-C., LIN, C-P., A Data Mining Approach To Product Assortment And Shelf Space Allocation, Expert Systems with Applications, sayı 4, cilt 32, syf. 976-986, 2007
- [44] YANG, T-C., LAI, H., Comparison Of Product Bundling Strategies On Different Online Shopping Behaviors, Electronic Commerce Research and Applications, sayı 4, cilt 5, syf. 295-304, 2006
- [45] GRIFFITH, D.A., An Examination Of The Influences Of Store Layout In Online Retailing, Journal of Business Research, sayı 10, cilt 58, syf. 1391-1396, 2005
- [46] LIAO, S-H., HSIEH, C-L., HUANG, S-P., Mining Product Maps For New Product Development, Expert Systems with Applications, sayı 1, cilt 34, syf. 50-62, 2008
- [47] LIAO, S-H., CHEN, Y-J., Mining Customer Knowledge For Electronic Catalog Marketing, Expert Systems with Applications, sayı 4, cilt 27, syf. 521-532, 2004
- [48] ZHOU, Z., Maintaining Incremental Data Mining Association Rules, Yüksek Lisans Tezi, University of Windsor , Canada, 2000
- [49] AGRAWAL, R., SRIKANT, R., Fast Algorithms For Mining Association Rules, Proceedings of the 20th VLDB Conference, Santiago, syf. 487-499, 1994
- [50] GALA, D., Analyzing Association Rules Produced By Applying The Apriori Algorithm To Structured Data, Yüksek Lisans Tezi, Universtiy of Nevada, Las Vegas, 2006
- [51] BONDUGULA, P. Implementation And Analysis of Apriori Algorithm for Data, Yüksek Lisans Tezi, Universtiy of Nevada, Las Vegas, 2006
- [52] TAN, P.N., STEINBACH M., KUMAR, V., Introduction To Data Mining, Pearson Addison Wesley, syf. 359-369, 2005
- [53] ZAKI, M., HSIO, C-J., CHARM: An Effient Algorithm For Closed Association Rule Mining, Computer Science Department Rensselaer Polytechnic Institute In Technical Report 99-10, syf. 1-20, 1999
- [54] MARGAHNY, M.H., MITWALY, A.A., Fast Algorithm For Mining Association Rules, AIML 05 conference, Egypt, syf. 36-40, 2005

- [55] LIN, D-I., Fast Algorithm For Discoverin The Maximum Frequent Set, Doktora Tezi, Department of Computer Science New York University, NY, 1998
- [56] KANTARDZIC, M., Data mining: Concepts, Models, Methods And Algorithms, IEEE Press, syf. 167-173, 2003
- [57] SRIKANT, R., AGRAWAL, R., Mining Quantitative Association Rules In Large Relational Tables, ACM SIGMOD International Conference on Management of Data, syf. 1-12, 1996
- [58] GÜNDOĞAN K.K., Birliktelik Kurallarının Bulanık Mantık Ve Genetik Algoritma İle Otomatik Keşfi, Yüksek Lisans Tezi, FÜ Fen Bilimleri Enstitüsü, 2003
- [59] VRECHOPOULOS, A.P., O'KEEFE, R.M., DOUKIDIS, G.I., SIOMKOS, G.J., Virtual Store Layout: An Experimental Comparison In The Context Of Grocery Retail, Journal of Retailing, sayı 1, cilt 80, syf. 13-22, 2004
- [60] ARSLAN, F.M., BAYÇU S., Mağaza Atmosferi, Anadolu Üniversitesi Web-Ofset, syf. 237-254, Eskişehir, 2006
- [61] AKAYDIN, H., Perakende Mağaza Atmosferinin Müşterilerin Satın Alma Kararı Üzerindeki Rolü: Eskişehir İlindeki Alışveriş Merkezi Müşterileri İle Bir Araştırma, Yüksek Lisans Tezi, AÜ Sosyal Bilimler Enstitüsü, 2007
- [62] GAMBAROV, V., Perakendeci Mağazalarda Atmosfer: Azerbaycan' da Tüketicilerin Mağaza Seçimini Etkileyen Faktörler Üzerine Bir Araştırma, Yüksek Lisans Tezi, DÜ Sosyal Bilimler Enstitüsü, 2007
- [63] YENİDOĞAN T.G., Pazarlama Araştırmalarında Çok Boyutlu Ölçekleme Analizi: Üniversite Öğrencilerinin Marka Algısı Üzerine Bir Araştırma, Akdeniz İ.İ.B.F Dergisi, sayı 15, syf. 138-169, 2008
- [64] KRUSKAL, J.B., WISH, M., Multidimensional Scaling, Sage Publications, syf. 7-45, 1978
- [65] USTA AHMETOĞLU E., Marka Konumlandırma Çok Boyutlu Ölçekleme Tekniğinin Uygulanması, Yüksek Lisans Tezi, KÜ Sosyal Bilimler Enstitüsü, 2005
- [66] GIGUERE, G., Collecting And Analyzing Data In Multidimensional Scaling Experiments: A Guide For Psychologists Using Spss, Tutorials in Quantitative Methods for Psychology, sayı 1, cilt 2, syf. 26-37, 2006
- [67] ÖZDAMAR, K., Paket Programlar İle İstatistiksel Veri Analizi-2 (Çok Değişkenli Analizler), Kaan kitabevi, syf. 501-518, 2005

- [68] SİĞİRLİ, D., EDİZ, B., CANGÜR, Ş., ERCAN, İ., KAN, İ., Türkiye Ve Avrupa Birliği' ne Üye Ülkelerin Sağlık Düzeyi Ölçütlerinin Çok Boyutlu Ölçekleme Analizi İle İncelenmesi, İnönü Üniversitesi Tıp Fakültesi Dergisi, sayı 2, cilt 13, syf. 81-85, 2006
- [69] <http://psiexp.ss.uci.edu/research/papers/MultidimensionalScaling.pdf>, 11.10.2008
- [70] <http://faculty.chass.ncsu.edu/garson/PA765/mds.htm>, 11.10.2008
- [71] http://www.stat.psu.edu/~chiaro/BioinfoII/mds_sph.pdf, 11.10.2008

EKLER

EK 1. Birliktelik Kuralları

Tablo 1. Çalışma sonucunda elde edilen birliktelik kuralları

Kural No	Sonuç	Öncül	Güven	Destek
248	5	25	76,71	5,58
418	5	8	70,61	9,21
361	5	35	70,58	6,55
84	5	24	70,58	1,94
203	19	9	68,92	5,00
19	5	15	68,44	0,79
201	5	9	67,99	4,93
450	5	18	67,39	9,68
296	5	4	67,07	5,46
152	21	7	65,99	4,10
317	5	16	65,17	6,02
272	5	3	63,82	4,88
153	5	7	63,80	3,97
467	19	20	63,52	11,18
399	5	14	63,29	6,79
86	19	24	63,24	1,74
298	19	4	62,63	5,10
433	21	22	62,53	8,91
435	5	22	61,60	8,77
21	19	15	61,53	0,71
364	19	35	61,35	5,70
421	19	8	61,33	8,00
517	19	11	61,20	25,61
62	19	26	60,44	2,09
383	19	12	59,95	6,29
179	5	13	59,71	4,04
274	19	3	59,35	4,54

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
250	19	25	59,19	4,31
402	19	14	59,05	6,34
224	5	6	58,75	4,40
507	5	21	58,70	13,46
247	21	25	58,31	4,24
181	19	13	58,30	3,95
20	11	15	58,29	0,67
323	19	1	57,66	10,06
381	5	12	56,77	5,96
47	19	28	56,53	2,16
315	21	16	56,28	5,20
60	5	26	56,23	1,95
61	11	26	56,00	1,94
453	19	18	55,73	8,01
363	11	35	55,60	5,16
155	19	7	55,53	3,45
202	11	9	55,38	4,02
111	19	34	55,35	3,04
249	11	25	55,23	4,02
226	19	6	55,15	4,13
297	11	4	55,13	4,48
438	19	22	55,12	7,85
319	19	16	55,04	5,08
85	11	24	54,78	1,51
154	11	7	54,53	3,39
515	19	5	53,98	17,51
401	11	14	53,79	5,77
489	19	27	53,65	9,34
180	11	13	53,37	3,61
273	11	3	53,30	4,07
471	29	23	53,25	8,60
17	10	15	53,25	0,61
437	11	22	53,06	7,56
18	21	15	53,04	0,61
497	19	29	52,96	11,80
225	11	6	52,91	3,97
518	11	19	52,67	25,61
420	11	8	52,65	6,87

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
476	5	23	52,58	8,49
318	11	16	52,57	4,85
485	5	27	52,51	9,14
479	19	23	51,53	8,32
45	5	28	51,44	1,96
452	11	18	51,39	7,39
121	19	33	51,22	4,17
465	11	20	51,17	9,01
382	11	12	50,96	5,35
98	19	32	50,80	2,41
343	19	31	50,40	5,80
34	19	30	50,23	1,54
487	11	27	50,11	8,72
480	29	27	49,53	8,62
397	21	14	49,36	5,30
342	11	31	48,95	5,63
448	21	18	48,75	7,01
81	29	24	48,60	1,34
513	11	5	48,51	15,73
493	5	29	48,05	10,71
511	19	21	47,72	10,94
293	29	4	47,71	3,88
478	11	23	47,60	7,69
109	5	34	47,56	2,61
83	21	24	47,44	1,31
246	10	25	47,33	3,44
271	21	3	47,29	3,61
110	11	34	46,81	2,57
295	21	4	46,62	3,79
463	5	20	46,58	8,20
509	11	21	46,46	10,65
495	11	29	46,45	10,35
12	18	15	46,34	0,53
269	29	3	46,08	3,52
176	29	13	46,02	3,12
46	11	28	45,83	1,75
78	20	24	45,54	1,25
218	23	6	45,51	3,41

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
5	16	15	45,37	0,52
167	12	13	45,32	3,07
175	27	13	45,31	3,07
221	29	6	45,13	3,38
79	23	24	45,11	1,24
129	19	2	44,81	5,16
358	29	35	44,64	4,15
469	27	23	44,55	7,20
314	10	16	44,41	4,10
341	5	31	44,40	5,11
240	18	25	44,35	3,23
32	5	30	44,33	1,36
80	27	24	44,10	1,21
97	11	32	44,07	2,09
72	35	24	43,86	1,21
33	11	30	43,78	1,34
59	21	26	43,63	1,51
174	23	13	43,53	2,95
500	10	21	43,05	9,87
416	21	8	42,86	5,59
360	21	35	42,70	3,97
447	10	18	42,63	6,13
378	29	12	42,43	4,45
308	18	16	42,34	3,91
96	5	32	42,32	2,01
292	27	4	42,08	3,42
150	29	7	42,05	2,62
120	11	33	42,04	3,43
42	29	28	41,87	1,60
337	29	31	41,60	4,78
290	23	4	41,51	3,38
508	21	5	41,50	13,46
267	23	3	41,45	3,17
470	23	27	41,35	7,20
198	29	9	41,16	2,99
501	5	10	40,94	13,23
396	10	14	40,93	4,39
415	10	8	40,90	5,34

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
106	29	34	40,88	2,25
245	29	25	40,82	2,97
268	27	3	40,82	3,12
502	10	5	40,79	13,23
29	29	30	40,74	1,25
220	27	6	40,35	3,02
16	29	15	40,19	0,46
413	29	8	40,00	5,22
322	11	1	39,96	6,97
119	5	33	39,93	3,25
68	3	24	39,85	1,10
116	29	33	39,84	3,25
151	10	7	39,81	2,48
223	21	6	39,79	2,98
354	23	35	39,71	3,69
395	29	14	39,60	4,25
356	27	35	39,38	3,66
128	11	2	39,13	4,50
434	22	21	38,84	8,91
200	21	9	38,79	2,81
294	10	4	38,73	3,15
82	10	24	38,70	1,07
195	20	9	38,67	2,80
481	27	29	38,66	8,62
472	23	29	38,59	8,60
359	10	35	38,37	3,56
41	27	28	38,34	1,46
351	18	35	38,34	3,56
432	10	22	38,23	5,45
333	23	31	38,22	4,39
146	18	7	38,09	2,37
376	27	12	37,78	3,96
287	18	4	37,71	3,07
251	4	3	37,70	2,88
514	5	11	37,59	15,73
348	8	35	37,50	3,48
178	21	13	37,40	2,53
40	23	28	37,28	1,42

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
505	19	10	37,20	12,02
9	14	15	37,16	0,43
239	22	25	37,10	2,70
335	27	31	37,07	4,26
69	4	24	36,85	1,01
374	23	12	36,83	3,86
380	21	12	36,69	3,85
148	23	7	36,62	2,28
149	27	7	36,53	2,27
313	29	16	36,47	3,37
75	8	24	36,47	1,00
57	29	26	36,44	1,26
237	8	25	36,39	2,65
64	13	24	36,39	1,00
270	10	3	36,36	2,78
93	29	32	36,33	1,72
458	29	20	36,22	6,38
265	18	3	36,19	2,77
77	18	24	36,13	1,00
284	8	4	36,07	2,93
372	20	12	36,00	3,78
516	5	19	36,00	17,51
58	10	26	35,98	1,24
503	11	10	35,90	11,60
445	29	18	35,84	5,15
73	12	24	35,81	0,99
282	14	4	35,79	2,91
388	18	14	35,58	3,82
252	3	4	35,42	2,88
474	21	23	35,06	5,66
405	18	8	34,99	4,57
145	22	7	34,89	2,17
244	27	25	34,79	2,53
74	14	24	34,71	0,96
278	35	4	34,70	2,82
108	21	34	34,64	1,90
142	14	7	34,64	2,15
262	8	3	34,63	2,65

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
199	10	9	34,40	2,50
430	29	22	34,28	4,88
483	21	27	34,26	5,96
173	20	13	34,08	2,31
197	27	9	34,06	2,47
306	22	16	33,89	3,13
170	8	13	33,87	2,29
353	20	35	33,83	3,14
11	22	15	33,77	0,39
14	23	15	33,63	0,39
95	21	32	33,48	1,59
256	35	3	33,40	2,55
393	27	14	33,34	3,58
243	23	25	33,34	2,43
196	23	9	33,30	2,42
289	20	4	33,29	2,71
340	21	31	33,06	3,80
482	10	27	33,06	5,75
411	27	8	33,06	4,31
494	29	5	33,03	10,71
66	6	24	33,01	0,91
266	20	3	32,96	2,52
10	8	15	32,87	0,38
260	14	3	32,87	2,51
231	16	25	32,79	2,39
384	8	14	32,68	3,51
379	10	12	32,62	3,42
122	1	2	32,47	3,74
391	23	14	32,40	3,48
147	20	7	32,39	2,01
15	27	15	32,32	0,37
53	18	26	32,00	1,11
177	10	13	31,95	2,16
339	10	31	31,89	3,67
214	8	6	31,87	2,39
144	8	7	31,85	1,98
194	18	9	31,85	2,31
27	23	30	31,84	0,97

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
13	20	15	31,77	0,37
406	8	18	31,77	4,57
28	27	30	31,69	0,97
210	35	6	31,66	2,37
312	27	16	31,61	2,92
192	8	9	31,59	2,29
409	23	8	31,55	4,12
491	21	29	31,52	7,02
165	35	13	31,51	2,13
206	4	6	31,50	2,36
159	3	13	31,49	2,13
216	18	6	31,41	2,35
216	18	6	31,41	2,35
233	35	25	31,23	2,27
242	20	25	31,02	2,26
390	20	14	30,83	3,31
461	21	20	30,81	5,42
222	10	6	30,77	2,31
103	20	34	30,68	1,69
492	29	21	30,64	7,02
31	21	30	30,62	0,94
172	18	13	30,57	2,07
204	3	6	30,56	2,29
449	18	21	30,55	7,01
499	21	10	30,55	9,87
279	4	35	30,41	2,82
156	6	13	30,40	2,06
30	10	30	30,39	0,93
134	3	7	30,27	1,88
311	23	16	30,27	2,79
407	20	8	30,26	3,95
54	20	26	30,21	1,05
161	4	13	30,19	2,04
443	27	18	30,16	4,33
92	27	32	30,14	1,43
367	8	12	30,07	3,16
304	8	16	30,05	2,77
105	27	34	30,01	1,65

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
205	6	3	29,97	2,29
386	22	14	29,95	3,21
52	22	26	29,93	1,04
451	18	5	29,86	9,68
117	10	33	29,85	2,43
76	22	24	29,74	0,82
346	14	35	29,73	2,76
460	10	20	29,61	5,21
107	10	34	29,49	1,62
258	12	3	29,46	2,25
473	10	23	29,44	4,76
422	18	22	29,31	4,17
168	13	12	29,23	3,07
118	21	33	29,10	2,37
423	22	18	29,05	4,17
207	6	4	29,01	2,36
115	27	33	29,00	2,36
280	12	4	28,86	2,35
140	35	7	28,81	1,79
455	20	23	28,79	4,65
104	23	34	28,74	1,58
65	9	24	28,70	0,79
403	22	8	28,66	3,74
441	23	18	28,65	4,12
459	20	29	28,61	6,38
457	20	27	28,57	4,97
236	14	25	28,52	2,07
302	14	16	28,51	2,63
490	10	29	28,48	6,35
217	20	6	28,42	2,13
419	8	5	28,41	9,21
50	14	26	28,39	0,98
127	5	2	28,39	3,27
55	23	26	28,35	0,98
67	25	24	28,33	0,78
136	4	7	28,26	1,76
51	8	26	28,25	0,98
456	27	20	28,25	4,97

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
193	22	9	28,20	2,05
486	27	5	28,17	9,14
344	12	35	28,13	2,61
428	27	22	28,05	4,00
310	20	16	28,04	2,59
94	10	32	27,99	1,33
212	12	6	27,89	2,09
160	13	3	27,89	2,13
286	22	4	27,76	2,26
91	23	32	27,72	1,31
504	10	11	27,72	11,60
114	23	33	27,70	2,26
257	3	35	27,49	2,55
157	13	6	27,46	2,06
370	18	12	27,43	2,88
56	27	26	27,43	0,95
191	14	9	27,42	1,99
90	20	32	27,32	1,29
264	22	3	27,25	2,08
334	31	23	27,21	4,39
309	16	18	27,20	3,91
321	5	1	27,17	4,74
283	4	14	27,13	2,91
188	35	9	27,08	1,96
436	22	5	27,06	8,77
213	14	6	27,04	2,03
63	7	24	27,00	0,74
141	12	7	26,99	1,68
385	14	8	26,88	3,51
113	20	33	26,82	2,19
424	20	22	26,74	3,81
439	20	18	26,70	3,84
349	35	8	26,69	3,48
332	20	31	26,68	3,07
26	20	30	26,67	0,82
389	14	18	26,57	3,82
169	14	13	26,47	1,79
454	23	20	26,42	4,65

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
404	8	22	26,25	3,74
350	22	35	26,24	2,44
477	23	5	26,19	8,49
426	23	22	26,14	3,72
275	16	4	26,05	2,12
484	27	21	26,00	5,96
2	25	15	25,97	0,30
232	25	16	25,83	2,39
39	20	28	25,81	0,98
138	16	7	25,79	1,60
71	31	24	25,73	0,71
101	22	34	25,73	1,41
347	35	14	25,72	2,76
44	21	28	25,56	0,98
211	6	35	25,55	2,37
442	18	23	25,50	4,12
410	8	23	25,49	4,12
510	21	11	25,46	10,65
369	22	12	25,35	2,66
464	20	5	25,28	8,20
162	13	4	25,12	2,04
4	4	15	24,93	0,29
7	35	15	24,93	0,29
444	18	27	24,91	4,33
345	35	12	24,89	2,61
171	22	13	24,88	1,68
412	8	27	24,78	4,31
352	35	18	24,77	3,56
496	29	11	24,74	10,35
506	10	19	24,72	12,02
475	23	21	24,70	5,66
253	16	3	24,68	1,89
190	12	9	24,68	1,79
125	10	2	24,66	2,84
22	31	30	24,64	0,75
135	7	3	24,63	1,88
70	16	24	24,60	0,68
229	4	25	24,56	1,79

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
303	16	14	24,53	2,63
336	31	27	24,50	4,26
365	14	12	24,46	2,57
235	12	25	24,46	1,78
234	25	35	24,46	2,27
417	8	21	24,38	5,59
498	29	19	24,28	11,80
368	12	8	24,19	3,16
375	12	23	23,93	3,86
366	12	14	23,92	2,57
215	22	6	23,84	1,79
132	6	7	23,75	1,48
462	20	21	23,65	5,42
48	16	26	23,63	0,82
299	35	16	23,62	2,18
139	31	7	23,52	1,46
300	16	35	23,49	2,18
414	8	29	23,41	5,22
261	3	14	23,41	2,51
164	31	13	23,40	1,58
227	3	25	23,16	1,69
446	18	29	23,11	5,15
398	14	21	23,10	5,30
102	18	34	23,09	1,27
427	22	23	23,06	3,72
88	22	32	23,02	1,09
184	4	9	23,02	1,67
277	31	4	23,01	1,87
468	20	19	22,99	11,18
166	13	35	22,96	2,13
429	22	27	22,96	4,00
276	4	16	22,96	2,12
112	31	33	22,87	1,86
24	22	30	22,87	0,70
355	35	23	22,83	3,69
377	12	27	22,78	3,96
87	31	32	22,72	1,08
316	16	21	22,66	5,20

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
8	12	15	22,65	0,26
6	31	15	22,65	0,26
387	14	22	22,57	3,21
512	21	19	22,50	10,94
285	4	8	22,49	2,93
241	25	18	22,44	3,23
408	8	20	22,43	3,95
133	25	7	22,38	1,39
281	4	12	22,37	2,35
43	10	28	22,33	0,85
89	18	32	22,32	1,06
100	8	34	22,30	1,23
35	9	28	22,21	0,85
36	8	28	22,19	0,85
99	31	34	22,07	1,21
228	25	3	22,04	1,69
255	31	3	22,04	1,69
327	31	12	22,02	2,31
307	16	22	21,97	3,13
230	25	4	21,96	1,79
130	13	7	21,95	1,37
431	22	29	21,90	4,88
440	18	20	21,80	3,84
38	18	28	21,79	0,83
25	18	30	21,72	0,66
425	22	20	21,64	3,81
182	3	9	21,63	1,57
331	18	31	21,62	2,49
137	7	4	21,61	1,76
23	8	30	21,54	0,66
392	14	23	21,53	3,48
466	20	11	21,52	9,01
325	31	35	21,49	2,00
373	12	20	21,46	3,78
338	31	29	21,46	4,78
259	3	12	21,46	2,25
123	2	1	21,42	3,74
330	22	31	21,39	2,46

Tablo 1 (Devam)

Kural No	Sonuç	Öncül	Güven	Destek
288	4	18	21,35	3,07
320	21	1	21,30	3,72
305	16	8	21,27	2,77
49	35	26	21,22	0,73
189	9	35	21,15	1,96
3	3	15	21,13	0,24
219	6	23	21,12	3,41
37	22	28	21,08	0,80
301	12	16	21,04	1,94
329	8	31	21,03	2,42
208	16	6	21,03	1,58
357	35	27	21,01	3,66
400	14	5	20,94	6,79
291	4	23	20,91	3,38
328	31	14	20,90	2,24
209	31	6	20,87	1,56
488	27	11	20,83	8,72
187	31	9	20,80	1,51
124	29	2	20,72	2,38
163	16	13	20,69	1,40
324	1	19	20,68	10,06
394	14	27	20,57	3,58
126	21	2	20,53	2,36
185	9	4	20,52	1,67
183	9	3	20,52	1,57
254	3	16	20,43	1,89
158	25	13	20,38	1,38
186	16	9	20,37	1,48
1	26	15	20,30	0,23
238	25	8	20,29	2,65
263	3	8	20,29	2,65
362	35	5	20,21	6,55
131	7	13	20,17	1,37
326	12	31	20,10	2,31
143	7	14	20,08	2,15
371	12	18	20,03	2,88

EK 2. Ürün Kategori Numaraları

Tablo 2. Ürün kategori numaraları açıklamaları

Ürün kategori no	Ürün kategori adı
1	Alkollü içecekler
2	Tütün
3	Bakliyat
4	Makarna
5	Süt ve süt ürünleri
6	Sıvı yağlar
7	Tuz-baharat-harç
8	Kahvaltılıklar
9	Tatlılar ve hamur işleri
10	Unlu mamüller
11	İçecekler
12	Sıcak içecekler
13	Şeker
14	Konserve
15	Kırmızı et ürünleri
16	Tavuk ürünleri
17	Diğer ürünler
18	Et şarküteri
19	Çerezler
20	Şekerli ve diabetik ürünler
21	Sebze
22	Meyve
23	Deterjan ve temizlik ürünleri
24	Un
25	Yumurta
26	Dondurulmuş ürünler
27	Kağıt ürünleri
28	Bebek ürünleri
29	Kozmetik
30	Elektrikli ve elektronik ev aletleri
31	Züccaciye-ev-bahçe ürünleri
32	Oyuncak-Pet ürünleri-Hobi
33	Tekstil ve ayakkabı
34	Kitap-Dergi-Kırtasiye
35	Tereyağ-margarin

EK 3. İlişki Matrisi

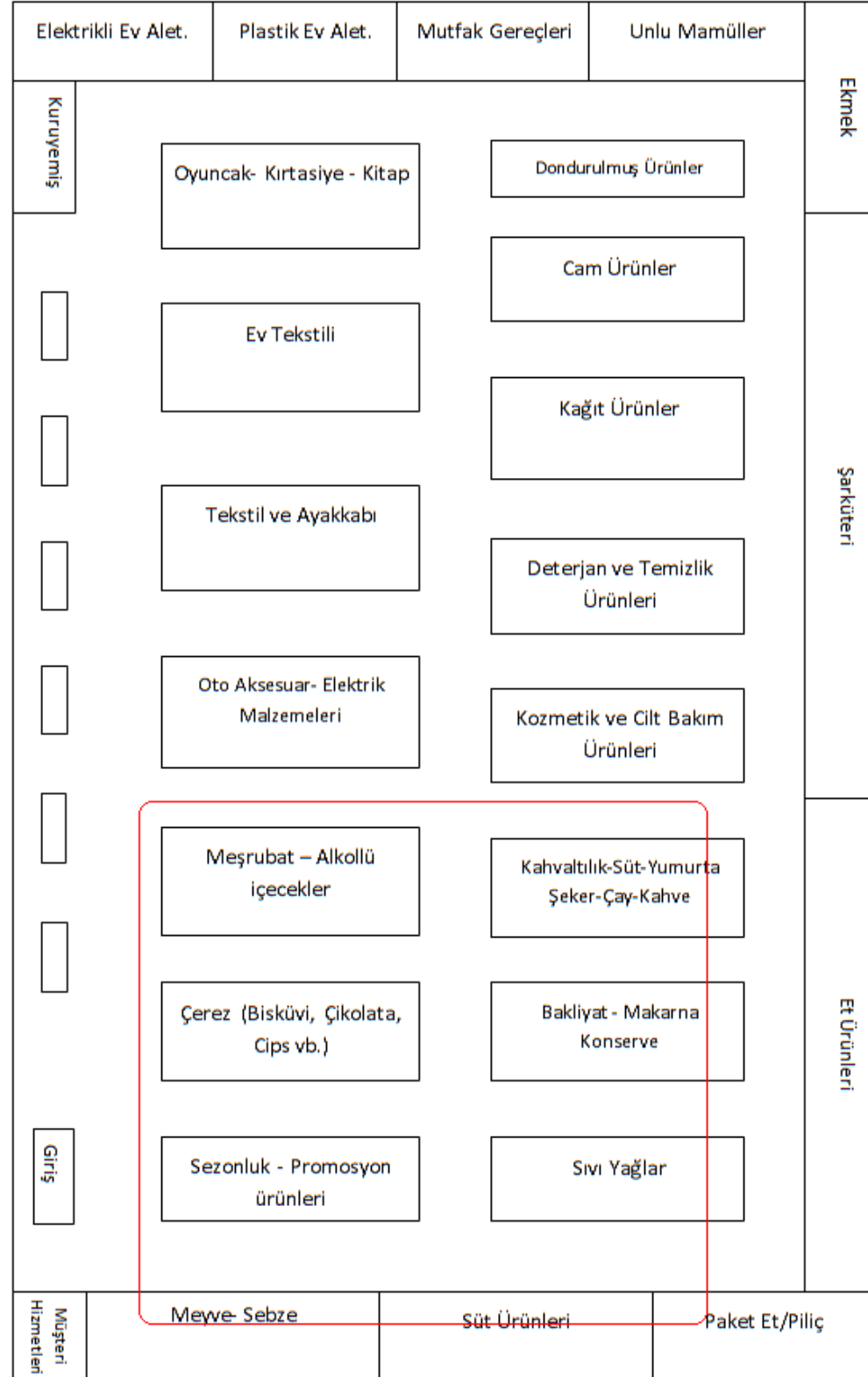
Tablo 3. Birliktelik kurallarından oluşturulan ilişki matrisi

Ü.K.	1	3	28	23	26	30	27	8	34	14	29	4	32	12	6
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	41	0	0	41	35	0	33	46	38	0	29	30
28	0	0	0	37	0	0	38	22	0	0	42	0	0	0	0
23	0	0	0	0	0	0	45	25	0	22	53	21	0	24	21
26	0	0	0	28	0	0	27	28	0	28	36	0	0	0	0
30	0	0	0	32	0	0	32	22	0	0	41	0	0	0	0
27	0	0	0	41	0	0	0	25	0	21	50	0	0	23	0
8	0	20	0	32	0	0	33	0	0	27	40	22	0	24	0
34	0	0	0	29	0	0	30	22	0	0	41	0	0	0	0
14	0	23	0	32	0	0	33	33	0	0	40	27	0	24	0
29	0	0	0	39	0	0	39	23	0	0	0	0	0	0	0
4	0	35	0	42	0	0	42	36	0	36	48	0	0	29	29
32	0	0	0	28	0	0	30	0	0	0	36	0	0	0	0
12	0	21	0	37	0	0	38	30	0	24	42	22	0	0	0
6	0	31	0	46	0	0	40	32	0	27	45	31	0	28	0
13	0	31	0	44	0	0	45	34	0	26	46	30	0	45	30
20	0	0	0	26	0	0	28	22	0	0	36	0	0	21	0
9	0	22	0	33	0	0	34	32	0	27	41	23	0	25	0
33	0	0	0	28	0	0	29	0	0	0	40	0	0	0	0
35	0	27	0	40	0	0	39	38	0	30	45	30	0	28	26
7	0	30	0	37	0	0	37	32	0	35	42	28	0	27	24
2	32	0	0	0	0	0	0	0	0	0	21	0	0	0	0
24	0	40	0	45	0	0	44	36	0	35	49	37	0	36	33
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	23	0	33	0	0	35	36	0	29	41	25	0	24	0
31	0	0	0	38	0	0	37	21	0	0	42	0	0	20	0

Tablo 3 (Devam)

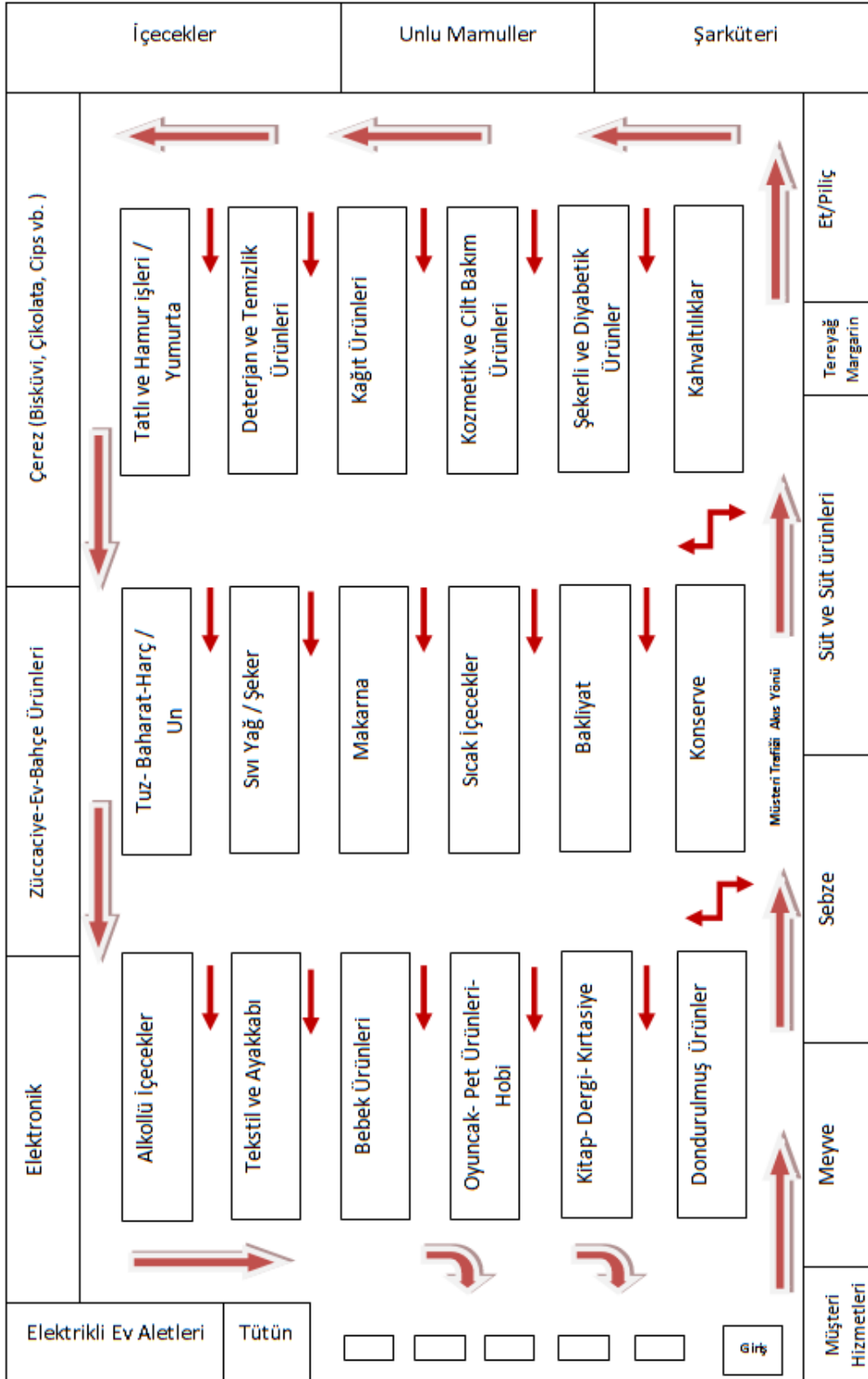
Ü.K.	13	20	9	33	35	7	2	24	10	25	31
1	0	0	0	0	0	0	21	0	0	0	0
3	28	33	21	0	33	25	0	0	36	22	22
28	0	26	22	0	0	0	0	0	22	0	0
23	0	29	0	0	23	0	0	0	29	0	27
26	0	30	0	0	21	0	0	0	36	0	0
30	0	27	0	0	0	0	0	0	30	0	25
27	0	29	0	0	21	0	0	0	33	0	24
8	0	30	0	0	27	0	0	0	41	20	0
34	0	31	0	0	0	0	0	0	29	0	22
14	0	31	0	0	26	20	0	0	41	0	21
29	0	29	0	0	0	0	0	0	28	0	21
4	25	33	21	0	35	22	0	0	39	22	23
32	0	27	0	0	0	0	0	0	28	0	23
12	29	36	0	0	25	0	0	0	33	0	22
6	27	28	0	0	32	0	0	0	31	0	21
13	0	34	0	0	32	20	0	0	32	20	23
20	0	0	0	0	0	0	0	0	30	0	0
9	0	39	0	0	27	0	0	0	34	0	21
33	0	27	0	0	0	0	0	0	30	0	23
35	23	34	21	0	0	0	0	0	38	24	21
7	22	32	0	0	29	0	0	0	40	22	24
2	0	0	0	0	0	0	0	0	25	0	0
24	36	46	29	0	44	27	0	0	39	28	26
10	0	0	0	0	0	0	0	0	0	0	0
25	0	31	0	0	31	0	0	0	47	0	0
31	0	27	0	0	0	0	0	0	32	0	0

EK 4. Mevcut Yerleşim Düzeni



Şekil 3. Mevcut yerleşim düzeni

EK 5. Önerilen Yerleşim Düzeni



Şekil 4. Önerilen yerleşim düzeni

ÖZGEÇMİŞ

Derya Ay, 22.09.1984 de İstanbul' da doğdu. İlk ve orta eğitimini Gaziosmanpaşa ve lise eğitimini Eminönü' nde tamamladı. 2002 yılında Çemberlitaş Kız Lisesi' nden mezun oldu. 2002 yılında başladığı Sakarya Üniversitesi Endüstri Mühendisliği Bölümün' den 2006 yılında mezun oldu. 2006-2007 yılları arasında sırasıyla Arma Form Endüstriyel İzolasyon San. ve Tic. A.Ş. ve Santana Deri ve Saraciye Ürünleri Paz. San. ve Tic. Ltd. Şti.' nde endüstri mühendisi olarak çalıştı. Şu anda Işık Sigorta A.Ş. AR-GE&Reasürans departmanında endüstri mühendisi olarak görev yapmaktadır.