

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

**MÜŞTERİ İŞLEMLERİNDEKİ BİRLİKTELİKLERİN
BELİRLENMESİNDE VERİ MADENCİLİĞİ UYGULAMASI**

**Feridun Cemal ÖZÇAKIR
(Teknik Öğretmen)**

**YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR-KONTROL EĞİTİMİ PROGRAMI**

**DANIŞMAN
Prof.Dr. Yılmaz ÇAMURCU**

İSTANBUL 2006

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

**MÜŞTERİ İŞLEMLERİNDEKİ BİRLİKTELİKLERİN
BELİRLENMESİNDE VERİ MADENCİLİĞİ UYGULAMASI**

**Feridun Cemal ÖZÇAKIR
(Teknik Öğretmen)
(141100419950010)**

**YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR-KONTROL EĞİTİMİ PROGRAMI**

**DANIŞMAN
Prof.Dr. Yılmaz ÇAMURCU**

İSTANBUL 2006

ÖNSÖZ

Bu tez çalışmasında, bana yol gösteren, rehberlik eden, kısıtlı zamanını benden esirgemeyen, olumlu yaklaşımları ile sürekli teşvik eden, her konuda destek veren tez danışmanım Prof.Dr.Yılmaz ÇAMURCU'ya,

Çalışmalarım sırasında beni yalnız bırakmayarak, kendilerine ayıramadığım zamandan fedakarlık eden, anlayış gösteren ve teşvik eden eşime, kızıma ve oğluma,

teşekkürlerimi sunarım.

Temmuz, 2006

Feridun Cemal ÖZÇAKIR

İÇİNDEKİLER

	SAYFA
ÖNSÖZ	I
ÖZET	V
ABSTRACT	VII
YENİLİK BEYANI	IX
SEMBOL LİSTESİ	X
KISALTMALAR	XI
ŞEKİL LİSTESİ	XII
TABLO LİSTESİ	XIV
BÖLÜM I. GİRİŞ ve AMAÇ.....	1
I.1. VERİTABANLARINDA BİLGİ KEŞFİ VE VERİ MADENCİLİĞİ.....	1
I.2. TEZ ÇALIŞMASININ AMACI.....	5
I.3. TEZ BÖLÜMLERİNİN ÖZETİ.....	5
BÖLÜM II. VERİ MADENCİLİĞİ LİTERATÜRÜNE GENEL BAKIŞ ..	7
II.1. VERİTABANLARI VE VERİ AMBARLARI	7
II.1.1. İlişkisel Veritabanları	8
II.1.2. Veri Ambarları	8
II.1.3. İşlemsel Veritabanları.....	9
II.1.4. Çok Boyutlu Veri Modelleri.....	9
II.1.4.1. Çok Boyutlu Veri Modelinde OLAP Operasyonları.....	11
II.2. VERİ SEÇİMİ, ÖNİŞLEME VE İNDİRGEME.....	12
II.2.1. Veri Seçimi.....	12
II.2.2. Veri Önileme.....	13
II.2.2.1. Veri Tipi Dönüşümü	14
II.2.2.2. Gruplama	14
II.2.2.3. Kayıp Değerleri Yönetmek	14

II.2.2.4. Gürültülü Veri.....	14
II.2.3. Veri İndirgeme	15
II.3. VERİ MADENCİLİĞİ.....	15
II.3.1. Sınıflama ve Gerileme-regresyon	16
II.3.1.1. Sınıflama	16
II.3.1.2. Gerileme – regresyon.....	16
II.3.2. Kümeleme.....	17
II.3.3. Zaman Serilerindeki Örüntü.....	18
II.3.4. Ardışıl Örüntü	18
BÖLÜM III. BİRLİKTELİK-İLİŞKİ (EŞLEŞTİRME) KURALI	19
III.1. BİRLİKTELİK-İLİŞKİ KURALININ TANIMI	20
III.2. BİRLİKTELİK-İLİŞKİ KURALININ MATEMATİKSEL MODELİ.....	20
III.2.1. Güven (Confidence) ve Destek (Support) Ölçüleri	22
III.2.2. Birliktelik-İlişki Kuralının Aşamaları	25
III.3. AIS ALGORİTMASI	25
III.4. APRİORİ ALGORİTMASI	26
III.5. DHP ALGORİTMASI -	26
III.6. PARTİON ALGORİTMASI	27
BÖLÜM IV.- BİRLİKTELİK KURALLARINA İLİŞKİN UYGULAMA YAZILIMININ GELİŞTİRİLMESİ	28
IV.1. ARAŞTIRMA YÖNTEMİ	28
IV.2. ARAŞTIRMA ARAÇLARI.....	33
IV.2.1. Donanım	33
IV.2.2. Yazılım	33
IV.3. VERİ MADENCİLİĞİ İŞLEMLERİ	33
IV.3.1. Çalışma Veri Depolama Yapısının Belirlenmesi.....	35
IV.3.2. Veri Depolama Ortamında Tablo Yapılarının ve Tablolar Arasındaki İlişkilerin Oluşturulması	35
IV.3.2. Veri Madenciliği Uygulamasının Çalışacağı Veritabanına Veri Alınması	45
IV.3.2.1. Veri Seçme Süreci.....	45
IV.3.2.2. Veri Önışleme Süreci	45
IV.3.3. Apriori Algoritması ve Veri Madenciliği Uygulama Yazılımı	47
IV.3.3.1. Uygulama Yazılımının Geliştirme Ortamı.....	47
IV.3.3.2. Uygulama Yazılımının Aşamaları.....	48
IV.3.3.2. Apriori Algoritması	50
BÖLÜM V. SONUÇLAR	65
V.1. GİRİŞ	65
V.2. ELDE EDİLEN SONUÇLAR.....	65
BÖLÜM VI. DEĞERLENDİRME VE ÖNERİLER	79
VI.1. DEĞERLENDİRME	79
VI.2. ÖNERİLER.....	80
KAYNAKÇA	82

ÖZGEÇMİŞ	85
-----------------------	-----------

ÖZET

MÜŞTERİ İŞLEMLERİNDEKİ BİRLİKTELİKLERİN BELİRLENMESİNDE VERİ MADENCİLİĞİ UYGULAMASI

Teknolojik gelişmeler ile birlikte günümüzde her alanda sürekli olarak şirketler ve kurumlar özellikle müşteri ve satış verilerini depolamaktadırlar. Bu verilerden veri madenciliği teknikleri uygulanarak önceden bilinmeyen, veri içinde gizli, anlamlı, potansiyel olarak kullanışlı ve değerli bilgiler elde edilmek istenmektedir. Birliktelik-ilişki kuralı da bu tekniklerden biridir. Birliktelik-ilişki kuralı, hareket verileri içinde birlikte hareket eden öğelerin keşfedilmesi, keşfedilen bu bağıntılar ile geleceğe yönelik tahminler üretilmesini sağlar.

Apriori algoritması, veri madenciliğinde sık geçen öğelerin keşfedilmesi için kullanılan en çok bilinen birliktelik-ilişki kuralı algoritmasıdır, temel olarak iteratif bir yapıya sahiptir. Sık geçen öğeleri bulmak için birçok kez veritabanını taramak gerekir, bu taramalar aşamasında Apriori algoritmasının birleştirme, budama işlemleri ve minimum destek ölçütü yardımı ile birliktelik ilişkisi olan öğeler bulunur.

Bu tez kapsamında, veritabanlarında bilgi keşfi süreçleri, veri madenciliği, veri madenciliğinde kullanılan birliktelik-ilişki kuralı ve Apriori algoritması hakkında bilgiler verilmiştir.

Bu tezde ayrıca, bir firmanın pastane satış lokasyonlarındaki satış verileri üzerinde veri madenciliği uygulamak için, çok boyutlu bir veritabanı tasarlanmıştır. Veri seçme işlemi ile operasyon veritabanından uygulama veritabanına veriler transfer edilmiştir. Veritabanı içindeki veriler üzerinde veri ön işleme ve veri indirgeme süreçleri uygulanarak veri madenciliğine uygun veri seti elde edilmiştir.

Apriori algoritmasını uyguyabilmek için .net platformunda Web tabanlı bir uygulama yazılımı geliştirilmiştir. Bu uygulama yazılımı algoritmanın her aşamasında uygulanan işlemleri adım adım görüntülenmesine imkan sağlayacak şekilde tasarlanmıştır.

Uygulama yazılımı ile veritabanındaki veriler üzerinde Apriori algoritması uygulanmıştır, uygulama yazılımının çalışması esnasında algoritmanın her aşaması izlenmiştir. Her aşamadaki girdiler ve sonuçlar net olarak gözlemlenmiştir. Uygulanan Apriori algoritması ile farklı zaman dilimi, farklı satış lokasyonu girdi değerleri doğrultusunda birlikte satın alınan ürünler ile ilgili bağıntılar olduğu gözlemlenmiştir. Genelde aynı ürün grubuna ait ürünlerin en sık birlikte satın alınan ürünler olduğu görülmüştür.

ABSTRACT

DATA MINING APPLICATION IN ASSOCIATION OF CUSTOMER TRANSACTIONS

In this time period, many of companies and corporates specially store customer and sales data in databases together with technological developments. They want to obtain previously unknown, implicit, meaningful, and potentially useful information from data in databases with data mining techniques. Association rule mining is one kind of data mining techniques which discovers strong association or correlation relationships among a large of data items.

The Apriori algorithm is the most popular association rule algorithm which discovers all frequent itemsets in large database of transactions. This algorithm uses iterative approach to count the frequent itemsets. Using this algorithm, candidate patterns which receive sufficient support from the database and the algorithm uses apriori gen actions join and prune to find all frequent itemsets.

In this thesis, processes of knowledge discovery in databases, data mining, association rule and Apriori algorithm are explained.

Furthermore, to apply data mining on real sales data which are several pastry-shops transactions data; designed a multidimensional database, transfered operational data to thesis application database with data selection process, applied data preprocessing and data reduction on application-data and application-data comes ready to apply data mining.

On the .net platform, we generated a web-based application software. This software that represents processes of Apriori algorithm step by step. We applied Apriori algorithm on selected dataset with application software. When the application software is running, observed all processes of Apriori algorithm, also

observed all entries and all results each steps. Algorithm generated some relations between together purchased products with entry values which are different time period and different pastry-shop locations. To observe most frequently purchased products together which are generally in same product group.

July, 2006

Feridun Cemal ÖZÇAKIR

YENİLİK BEYANI

MÜŞTERİ İŞLEMLERİNDEKİ BİRLİKTELİKLERİN BELİRLENMESİNDE VERİ MADENCİLİĞİ UYGULAMASI

Bu tez çalışması, hiçbir lisans ve lisansüstü tez çalışmasının kopyası değildir. Tamamen tarafımızca hazırlanmıştır.

I. ,II. ,III. ve IV. Bölümde yer alan açıklamalar ve değerlendirmelerde gerekli referanslar verilmiştir.

Gerçek bir işletmenin gerçek satış verileri kullanılmıştır.

Bu gerçek veriler üzerinde Apriori algoritmasının uygulanabilmesi için web ortamında çalışabilen bir adet uygulama yazılımı tez öğrencisi Feridun Cemal Özçakır tarafından geliştirilmiştir. Bu yazılım hiçbir yazılımın kopyası değildir. Tamamen tez öğrencisinin yazdığı kodlar ile geliştirilmiştir.

SEMBOL LİSTESİ

k	: küme sayısı
C_k	: aday öge kümesi
L_k	: sık geçen öge kümesi
$\Join$	: apriori-gen işlevinde birleştirme (join) işlemi
$A \Rightarrow B$	: Birliktelik-ilişki kuralı

KISALTMALAR

KDD	: Knowledge Discovery in Databases
VTBK	: Veri Tabanlarında Bilgi Keşfi
CRM	: Customer Relations Management
SQL	: Structured Query Language
OLTP	: Online Transaction Processing
OLAP	: Online Analytical Processing
DMQL	: Data Mining Query Language
I/O	: Input/Output
min.conf	: minumum confidence
min.sup	: minumum support
OLE-DB	: Object Linking and Embedding for Databases
ODBC	: Open Database Connectivity

ŞEKİL LİSTESİ

SAYFA

Şekil I.1 Fayyad'a göre veritabanlarında bilgi keşfi (VTBK) sürecinde yer alan adımlar [3].....	2
Şekil I.2 Han ve Kamber'e göre tipik veri madenciliği sistemi mimarisi [10].....	4
Şekil II.2 Satışlar veri ambarı için <i>yıldız (star)</i> şeması örneği [10].....	10
Şekil II.3 Satışlar veri ambarı için <i>kar tanesi (snowflake)</i> şeması örneği [10].....	11
Şekil IV.1 Yazarkasadan ofis bilgisayarına, ofis bilgisayarından merkez veritabanına transfer işlemi.....	30
Şekil IV.2 Satış verileri (fişli satış verileri) ile ilgili olarak oluşturulan GNDSALES.GTF isimli text dosyanın desen yapısı [27].....	32
Şekil IV.3 Tez çalışmasındaki veri madenciliği aşamaları.....	34
Şekil IV.4 SALES_TRAN_HEADERS tablosunun veri yapısı.....	36
Şekil IV.5 SALES_TRAN_LINES tablosunun veri yapısı	38
Şekil IV.6 DM_STOCK_CARDS tablosunun veri yapısı.....	40
Şekil IV.7 CODE_MAIN_DEPT tablosunun veri yapısı.....	41
Şekil IV.8 CODE_DEPT tablosunun veri yapısı.....	41
Şekil IV.9 CODE_SUB_DEPT tablosunun veri yapısı.....	41
Şekil IV.10 CODE_STOCK_TYPE tablosunun veri yapısı.....	42
Şekil IV.11 CODE_UNIT tablosunun veri yapısı.....	42
Şekil IV.12 BRANCHES tablosunun veri yapısı.....	43
Şekil IV.13 BTIP tablosunun veri yapısı.....	43
Şekil IV.14 HARTIP tablosunun veri yapısı.....	43
Şekil IV.15 Uygulamada kullanılan veritabanının tablo yapıları ve tablolar arasındaki ilişkiler.....	44
Şekil IV.16 Uygulama yazılımına giriş adımları.....	49
Şekil IV.17 Klasik Apriori Algoritması özet kodu [8, 28].....	51
Şekil IV.18 Apriori-gen işlevinin algoritma kesiti [28].....	51

Şekil IV.19 Tez çalışmasındaki yazılımda kullanılan Apriori algoritmasının aşamaları.....	53
Şekil IV.20 Apriori algoritmasında C_1 - L_1 dönüşümü.....	55
Şekil IV.21 Örnek verilerin $L_1 \bowtie L_1$ dönüşümü ile oluşan C_2 aday öge kümesi.....	56
Şekil IV.22 Örnek veriler üzerinde C_2 - L_2 dönüşümünün gerçekleşmesi.....	58
Şekil IV.23 Örnek verilerin $L_2 \bowtie L_2$ dönüşümü ile oluşan C_3 aday öge kümesi.....	59
Şekil IV.24 Apriori algoritmasında C_3 kümesinde budama işlemi.....	61
Şekil IV.25 C_3 - L_3 dönüşümünün gerçekleşmesi.....	63
Şekil IV.26 $L_3 \bowtie L_3$ dönüşümünün gerçekleşmesi.....	63
Şekil IV.27 Apriori algoritmasında C_4 aday öge kümesinde budama işlemi.....	64
Şekil V.1 Tez yazılımının kriter (girdi) girişi ekran görüntüsü.	66
Şekil V.2 Apriori Algoritması uygulanacak verinin görüntülediği ekran.....	67
Şekil V.3 C_1 aday öge kümesi ve minumum destek değeri giriş ekranı.....	68
Şekil V.4 L_1 sık geçen öge kümesi.....	69
Şekil V.5 C_2 aday öge kümesi.....	70
Şekil V.6 L_2 sık geçen öge kümesi.....	71
Şekil V.7 C_3 aday öge kümesi.....	72
Şekil V.8 C_3 aday öge kümesi – budama işlemi sonrası.....	73
Şekil V.9 L_3 sık geçen öge kümesi.....	74
Şekil V.10 C_4 aday öge kümesi.....	75
Şekil V.11 C_4 aday öge kümesi – budama sonrası.....	75
Şekil V.12 C_4 aday öge kümesinin öğelerinin destek değeri < min.destek değeri..	76
Şekil V.13 L_3 sık geçen öge kümesi-01.10-20.10.2005, minumum destek=3.....	77
Şekil V.14 L_3 sık geçen öge kümesi-01.04-30.04.2004, minumum destek=4.....	78

TABLO LİSTESİ

	SAYFA
Tablo II.1 Ürün hareket tablosu [23].....	23
Tablo IV.1 Apriori algoritmasında kullanılan değişkenler [28].....	50

BÖLÜM I

GİRİŞ ve AMAÇ

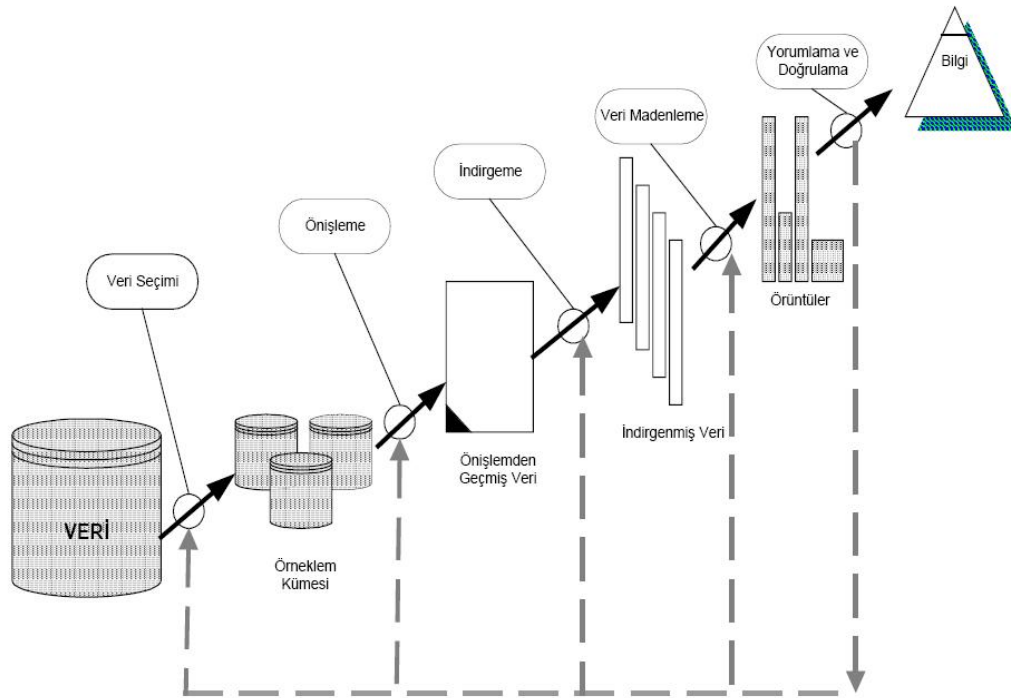
I.1. VERİTABANLARINDA BİLGİ KEŞFİ VE VERİ MADENCİLİĞİ

Yaşamış olduğumuz bu zaman diliminde her alışverişte, her bankacılık işleminde, her türlü kamusal alandaki işlemlerde kayıt edilen veriler, işletmelerin ve devletin kendi bünyelerinde yapılan işlemler sonucunda sakladığı veriler, görüntü ve ses cihazlarından elde edilen çokluortam verileri gibi birçok veri sürekli olarak depolanmakta ve depolanan bu veriler çok hızlı boyutlarda artmaktadır. Fakat bu veriler istenildiği şekilde değerlendirilememekte, hızla büyüyen bilgi yığınları şekline dönüşmektedir

Her geçen gün işletmeler ve devlet kurumları daha fazla veri tabanı sistemine yatırım yapmakta ve daha fazla oranda veriyi bu sistemlerde depolamaktadır, ama bu boyuttaki veri çoğu işletme veya kurumda anlamlı ve verimli bir şekilde işlenememektedir. Bu boyuttaki verilerden yararlanmak için şablonlar ve kurallar uygulanarak değerli bilginin keşfedilmesine ihtiyaç duyulmuştur bu sürece kısaca *Veritabanlarında bilgi keşfi - VTBK (Knowledge Discovery in Databases – KDD)* adı verilmektedir. Bazı araştırmacılar veritabanlarında bilgi keşfi ile veri madenciliğini eş anlamlı olarak kabul etmelerine rağmen, genel görüş veri madenciliği *veritabanlarında bilgi keşfi* sürecinin bir aşaması şeklindedir. *Veri madenciliği*; eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel

olarak kullanışlı bilginin çıkarılmasıdır [1]. Veri madenciliği başka bir ifade ile; anlamlı modeller ve kurallar ortaya çıkarmak amacıyla , otomatik ve yarı-otomatik yöntemler kullanarak büyük miktarlardaki veriyi analiz etme ve inceleme süreci olarak tanımlanabilir [2].

Veritabanlarında bilgi keşfi, depolanmış veri içerisindeki geçerli, yeni, faydalı ve sonuç olarak anlaşılabilir örüntülerin (patterns) çıkarılması sürecidir. Bu sürecin ilk adımı, uygulama alanının öğrenilmesi ile başlar ve uygulamanın amaçları doğrultusunda veri tabanları veya veri ambarlarından örnek veri setleri seçilir. Sonra, gürültülü, hatalı ve tutarsız verilerin çıkarıldığı veri temizleme ve ön işleme aşaması gelir. Gerekli durumlarda verinin, madenciliğe uygun bir biçime dönüştürülmesi için mevcut veri üzerinde indirgeme işlemi uygulanır. Bir sonraki aşama olan *veri madenciliği* ise, akıllı yöntem ve medotlar kullanılarak büyük miktarda veriden anlamlı, faydalı, daha önceden bilinmeyen bilgilerin ortaya çıkarılmasıdır. Daha sonra, veri madenciliği adımıyla ortaya çıkarılan örüntüler, içlerinden yararlı olanların belirlenmesi için değerlendirilir. Veritabanlarında bilgi keşfinin son basamağı ise, elde edilen bilginin görüntüleme ve bilgi gösterimi yöntemleri kullanılarak kullanıcıya sunulması şeklindedir. Şekil 1.1'de *Veritabanlarında bilgi keşfi* sürecinde yer alan adımlar görülmektedir.



Şekil 1.1 Fayyad'a göre veritabanlarında bilgi keşfi (VTBK) sürecinde yer alan adımlar [3].

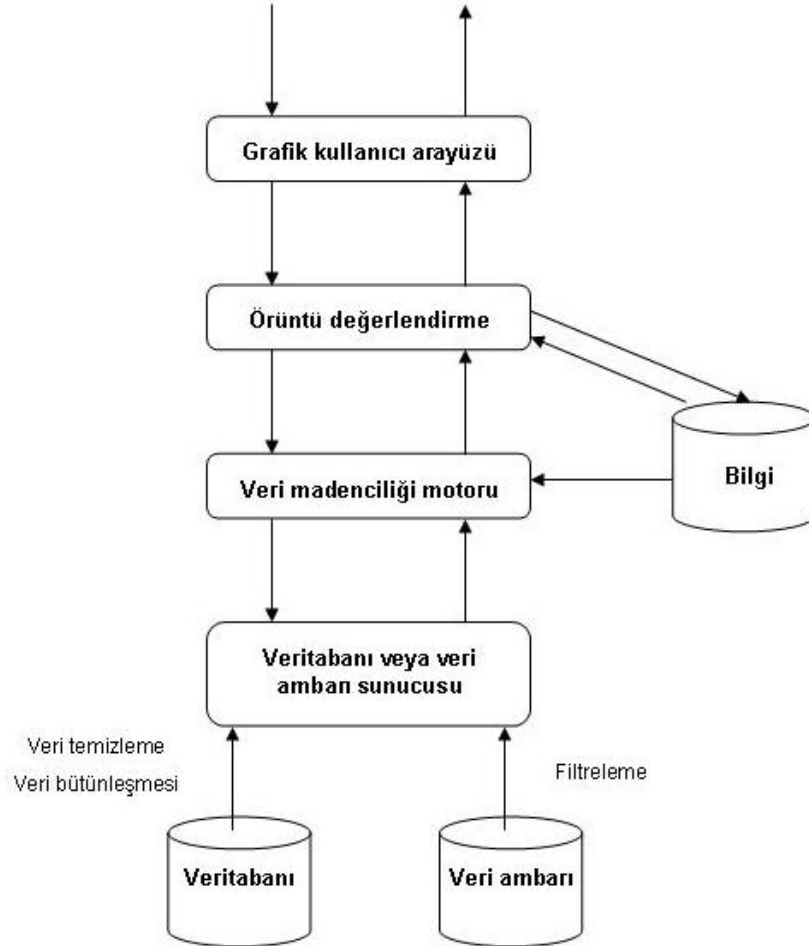
Fayyad' a göre ise *Veri tabanlarında bilgi keşfi (VTBK)* sürecindeki aşamaları özetle şu şekilde sıralanmaktadır [4];

- **Veri Seçimi (Data Selection):** Bu aşamada birden fazla veri kümesi birleştirilerek, üzerinde sorgu yapılmasına uygun örnek veri kümesi elde edilir. 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 Önışleme (Data Preprocessing):** Veri seçimi ile elde edilen örnek veri kümesinde yer alan hatalı ve eksik nitelikteki değerlerin düzenlendiği ve çıkarıldığı aşamadır. Bu aşama da, uygulanan işlemler sayesinde, üzerinde veri madenciliği yapılacak veri kaliteli veri niteliğine bürünür. *Veri temizleme (data cleaning)* ve *veri dönüştürme (data transformation)* veri önışleme işlemleridir [4].
- **Veri İndirgeme (Data Reduction):** Seçilen örnek veri kümesindeki ilgisiz nitelikte ve tekrarlı verilerin çıkarıldığı aşamadır. Bu işlem ile verinin boyutu indirgendiğinden veri madenciliği uygulanırken çalıştırılacak sorguların daha hızlı sonuç üretmeleri sağlanır [4].
- **Veri Madenciliği (Data Mining):** Bu aşama veri madenciliği yöntemlerinin (sınıflama, kümeleme, birliktelik-ilişki-eşleştirme, vb.) ve algoritmalarının (AIS, Apriori, vb.) uygulandığı adımdır [4].
- **Değerlendirme (Evaluation):** Bilgi keşfi sürecinde bu aşamadan önceki aşamalar sonucunda elde edilen bilginin geçerlilik, yenilik, yararlılık ve basitlik kıstaslarına göre değerlendirilmesi aşamasıdır [4].

Veri madenciliği için yapılan diğer tanımlardan bazıları da şunlardır:

- Hand'e göre veri madenciliği, veri tabanı teknolojisi, istatistik, makine öğrenim, örüntü tanımı, yapay sinir ağları, verilerin görselleştirilmesi ve uzaysal veri analizi gibi farklı disiplinlerde yer alan tekniklerin bir birleşimini içerir [5][6].
- Holsheimer tarafından yapılan bir tanıma göre veri madenciliği, büyük veri kümesi içinde saklı olan genel örüntülerin bulunması olarak açıklanmıştır [7].
- Veri madenciliği, önceden bilinmeyen ve potansiyel olarak faydalı olabilecek, veri içindeki gizli bilgilerin çıkarılmasıdır [8] [9].
- Veri madenciliği büyük miktarda veri içinden gelecekle ilgili tahmin yapılmasını sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır [8].

- Han ve Kamber'e göre veri madenciliği, büyük veri yığınları içerisinde gelecek ile ilgili tahminde bulunabilmemizi sağlayabilecek bağıntıların bilgisayar programı kullanarak aranmasıdır. Han ve Kamber'e göre de veri madenciliği veritabanlarında bilgi keşfinin bir adımını simgelemektedir. Şekil 1.2'de Han ve Kamber'e göre tipik veri madenciliği mimarisini görülmektedir [10].



Şekil 1.2 Han ve Kamber'e göre tipik veri madenciliği sistemi mimarisi [10].

Günümüzde veri madenciliği; satış, pazarlama, bankacılık, tıp, biyoloji, web kullanımı ve risk analizi gibi bir çok alanda uygulanmaktadır;

- Pazarlama alanında; müşteri ilişkileri yönetimi, satış tahmini, mevcut müşterilerinin kaybedilmemesi, müşteri satınalma örüntülerinin belirlenmesi gibi işlemlerde,

- Finans alanında; kredi taleplerinin değerlendirilmesi, kredi kartı dolandırıcılıklarında,
- Tıp alanında; SPECT, MRI, PET, gibi görüntüleme cihazları ve EKG, EEG gibi cihazların sinyallerinden elde verilerinin değerlendirilmesinde veri madenciliği teknikleri kullanılmaktadır [11].

I.2. TEZ ÇALIŞMASININ AMACI

Bu tez çalışmasının hazırlanma amacı;

- *Veritabanlarında bilgi keşfi* ve *veri madenciliği* kavramlarının araştırılarak, veritabanlarında bilgi keşfi süreçlerinin ve bu süreçler içinde yer alan veri madenciliği yöntem ve tekniklerinin genel bilgilendirme dökümanının oluşturulması,
- Veri madenciliği tekniklerinden *birlikte-ilişki (eşleştirme)* yönteminin araştırılması ve bu yöntemin en çok bilinen *Apriori* algoritmasının örnek bir firmanın (örnek firma; otel, restaurant, pub ve pastane hizmetleri vermektedir) pastane satış verileri üzerinde uygulanması için web tabanlı uygulama programı yazılması,
- Geliştirilen uygulama programı ile örnek firmanın pastane lokasyonlarında yıl içinde farklı dönemlerde, farklı satış noktalarında gerçekleşen satış verileri üzerinde birlikte satın alınma tercihi yapılan ürünlerin tespit edilmesi,
- Birlikte satın alınması tespit edilen ürün bilgileri ile firmanın üretim, planlama, stoklama, satış ve pazarlama süreçlerinde firma yararı yönünde düzenleme yapılmasına yön verecek sonuçlara ulaşılması

olarak belirlenmiştir.

I.3. TEZ BÖLÜMLERİNİN ÖZETİ

Tez de yer alan bölümlerin kısaca özeti aşağıda görülmektedir:

II.bölümde *veritabanlarında bilgi keşfi* ve *veri madenciliği* ile ilgili olarak yapılan literatür araştırması sonucunda elde edilen ve tez kapsamında yapılacak uygulama çalışmasının temelini ve süreçlerini oluşturacak kavramlar ve veri tabanlarında bilgi keşfi aşamaları bu bölümde açıklanmaktadır. Veri madenciliği için kullanılacak verilerin depolama sistemleri, veri seçme, veri önileme ve veri indirgeme süreçleri detaylı olarak anlatılmaktadır. Bilgi keşfi sürecinde seçilmiş, temizlenmiş,

dönüştürülmüş ve indirgenmiş veriler üzerinde özel teknikler kullanarak daha önceden karşılaşılmamış değerli bilgiye ulaşılan veri madenciliği aşaması, ilgili alanda kabul görmüş teknikler, yöntemler ve algoritmalar bu bölümde yer almaktadır.

III.bölümde bu tez çalışmasının uygulama aşamasında kullanılacak Birliktelik-ilişki kuralı veri madenciliği yöntemi detaylı olarak açıklanmakta, birliktelik-ilişki kuralı algoritmaları kısaca anlatılmaktadır.

IV.bölümde tez uygulama yazılımının çalışacağı gerçek veri setlerinin operasyon tarafında araştırılmasının aşamaları ve sonuçları, araştırma araçları hakkında bilgilendirme, operasyonel verinin veri seçme süreci ile uygulama veritabanına alınması aşaması, veritabanına alınan verilere veri ön işleme ve veri indirgeme işlemlerinin uygulanması süreçleri ve birliktelik-ilişki kuralının en çok bilinen algoritması Apriori hakkında detaylı bilgilendirme yer almaktadır. Apriori Algoritmasının çalışma prensipleri doğrultusunda uygulama yazılımının işleyişi aşamalı ve örnek kod yapıları, örnek veri modeli ile birlikte detaylı olarak açıklanmaktadır.

V.bölümde Apriori algoritmasının uygulama yazılımı ile gerçek veri seti üzerinde uygulanması aşamaları ve ekran görüntüleri yer almaktadır. Apriori algoritmasının gerçek veriseti üzerinde uygulanması ile uygulama yazılımı tarafından ilgili satış verilerinden keşfedilen birliktelik bağıntıları açıklanmaktadır.

VI.bölümde tez çalışmasının genel bir değerlendirmesi yapılmakta, elde edilen sonuçlara ve tez kapsamına göre önerilerde bulunmaktadır.

BÖLÜM II

VERİ MADENCİLİĞİ LİTERATÜRÜNE GENEL BAKIŞ

II.1. VERİTABANLARI VE VERİ AMBARLARI

İşletmelerde ve devlet kurumlarında 90 lı yılların başından itibaren bilgisayar sistemlerinin yaygınlaşması ile her türlü veri farklı depolama alanları içinde hızla büyüyen boyutlarda saklanmaya başlamıştır. Zamanla kurum ve işletmelerin mevcut veriler üzerinde yaptıkları çalışmalar ile elde edilen sonuçlar geleceğe yönelik planlamada kullanılarak kazanç elde etme çalışmaları artmıştır. Daha fazla değerli veriyi toplama çalışmaları önde gelen amaçlardan biri olmuştur. Örneğin daha önceden perakende satış yapılan noktalarda kullanılan satış cihazları standart, sadece satış fişi vermeye yönelik yazarkasalar iken, günümüzde çoğu satış noktasında satılan malların bilgilerini toplamak üzere kullanılan akıllı terminaller haline gelmişlerdir. Bu terminallerden elde edilen satış verileri sayesinde satış yapılan malların ve müşterilerin bilgileri veri depolama sistemlerinde toplanmakta ve toplanan bu veriler üzerinde kurum için yararlı olacak ilişki bilgileri ortaya çıkmaktadır.

Veri madenciliği birçok depolama birimi üzerinde uygulanabilir. Bunlar, ilişkisel veritabanları (relational databases), veri ambarları (data warehouses), işlemsel veri tabanları (transactional databases), gelişmiş veri tabanları, dosyalar ve web üzerindeki hareketler olabilir. 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].

II.1.1. İlişkisel Veritabanları

İlişkisel veri tabanları, tablolardan oluşmaktadır. Her tablonun;

- tekil (unique) bir adı vardır,
- veri alanlarından (kolonlar) oluşmaktadır,
- genelde geniş bir satır kümesi içerirler (kayıtlar).

İlişkisel veri tabanlarındaki her satır, veri alan değerleri ile tanımlanan bir nesneyi temsil eder [10].

İlişkisel veri, SQL (Structured Query Language) gibi yapısal sorgu dilleri ile yazılan sorgular ile veya grafik kullanıcı arayüzleri ile erişilebilen verilerdir. Kullanılan sorgu dili veya kullanıcı arayüzünün sağladığı olanaklar çerçevesinde, bu veritabanlarından istenilen veriler alınarak veri raporlaması, veri eklemesi, veri güncellemesi, veri silmesi işlemleri gerçekleştirilebilir [10].

II.1.2. Veri Ambarları

Kendi başına değerli olmayan ve büyük miktardaki verilerin saklanması için kullanılan veri ambarları, geleneksel sorgulama ve raporlama yöntemlerinin yanısıra, veri madenciliği ve çok boyutlu çözümleme (*multi-dimensional analysis*) gibi güçlü veri çözümleme teknikleri için altyapı sağlar. Bu tekniklerin kullanılması sayesinde, karar verme sürecinde gerekli olan bilgiye daha kolay erişim sağlanabilmektedir [8, 12].

Veri tabanlarında bilgi keşfinin ilk adımı olarak kabul edilen *veri toplama* işlemi için bir depolama alanı gereklidir, bu depolama alanları veri tabanları veya veri ambarlarıdır. *Veri ambarı*, bir işletmenin veya kamusal bir kurumun değişik bölümleri tarafından toplanan bilgilerin, gelecekte değerlendirilmek üzere arka plandaki sistemlerde birleştirilmesinden oluşan geniş ölçekli veri deposudur. W.H. Inmon'a göre ise, "*veri ambarı, özneye dayalı, bütünleşmiş, zaman dilimli ve yöneticinin karar verme işleminde yardımcı olacak biçimde toplanmış değişmeyen veriler topluluğudur*" [10].

Veri ambarları müşteri, tedarikçi, ürün bilgisi, stok, alış ve satış verisi gibi önemli özneler üzerine kurulur ve veriler veri ambarlarına tarihi bir bakış açısından bilgi sağlamak için depolanır. Veri ambarları genellikle ilişkisel veri tabanları, farklı

türden kayıt dosyaları ve sürecin işlemsel kayıtları ile bütünleştirilerek oluşturulur. Veri ambarları hareket işleme, veriyi geri çekme, ilişkisel kontrol mekanizmaları gerektirmez, veri ambarlarında sadece verinin ilk yüklenmesi ve daha sonradan veriye erişim mekanizmaları olması yeterlidir.

II.1.3. İşlemsel Veritabanları

İşlemsel veritabanları sistemleri çevrim içi hareket işlemeyi ve sorgulamayı gerçekleştirmektedirler. Bu sistemlere *çevrim içi hareket işleme süreci (OLTP - Online Transaction Processing)* ismi verilir. Bu sistemler bir işletmeye veya kuruma ait alım-satım, üretim, finansal kayıt gibi günlük kayıtlarını tutmaktadır [10].

Veri ambarları ise veri ambarını kullananlara veri analizi ve karar verme rolü içerisinde hizmet vermektedir, bu sistemler veriyi değişik biçimlerde gösterebilir ve organize edebilirler. Bu sistemlere ise *çevrim içi analitik işlem süreci (OLAP - Online Analytical Processing)* ismi verilir [10].

OLTP ile OLAP arasındaki farklar;

- OLTP müşteri merkezlidir. Normal kullanıcılar, tedarikçiler ve müşteriler tarafından işlem hareketleri ve sorgular için kullanılır. OLAP ise pazar merkezlidir ve bilgi çalışanları, yöneticiler, uygulayıcılar ve analistler tarafından veri analizi için kullanılır [10].
- OLTP sistemi, karar vermede kolaylık sağlaması için fazla detaylı olan güncel veriyi yönetir. Bir OLAP sistemi ise, büyük miktarlarda tarihi veriyi yönetir, analiz esnasında stratejik veriye hızlı erişim ve özetlemede kolaylıklar sağlar ve öge boyutunun farklı seviyelerindeki bilgiyi saklar ve yönetir [10].

II.1.4. Çok Boyutlu Veri Modelleri

Han ve Kamber'e [10] göre veri ambarları ve OLAP teknolojisi çok boyutlu veri modeli üzerine temellenmiştir. İşlenmemiş veriye sahip büyük veritabanlarında OLAP teknikleri kullanılarak her boyutunda çoklu soyut seviyesi olan çoklu boyutlar düzenlenir. Çok boyutlu veri modeli, veriyi veri küpü formunda incelemektedir [10].

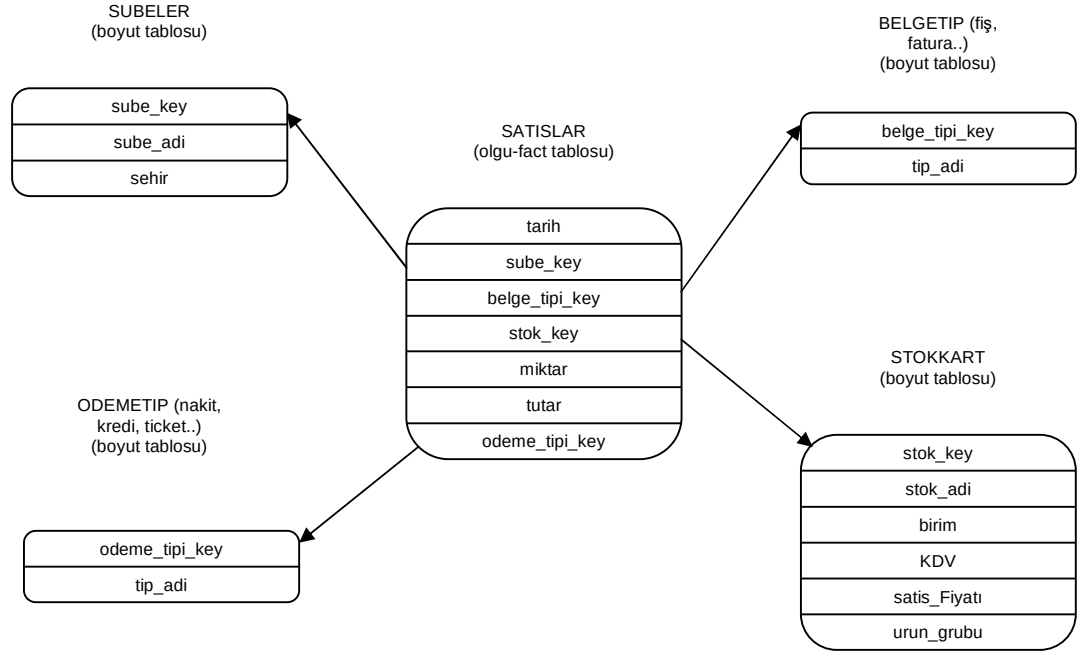
Veri küpü, verinin çoklu boyutta modellenmesini ve incelenmesini sağlar, boyutlar ve olgular ile tanımlanır. Genel koşullar altında boyutlar, organizasyonun kayıtlarını tutmak istediği perspektifler veya varlıklar ile ilgilidir. Çok boyutlu bir veri modeli tipik olarak, merkezi bir tema (örn. satış verisi) etrafında organize edilebilir. Tema bir olgu (fact) tablosu ile temsil edilir. Olgular, rakamsal ölçülerdir [10].

Veri ambarı için en popüler veri modeli, çok boyutlu modeldir. Böyle bir model yıldız, kar tanesi ve olgu takımı yıldız şeması biçiminde olabilir [10].

Yıldız şema (Star schema): En çok bilinen şema şeklidir. Şema yapısında;

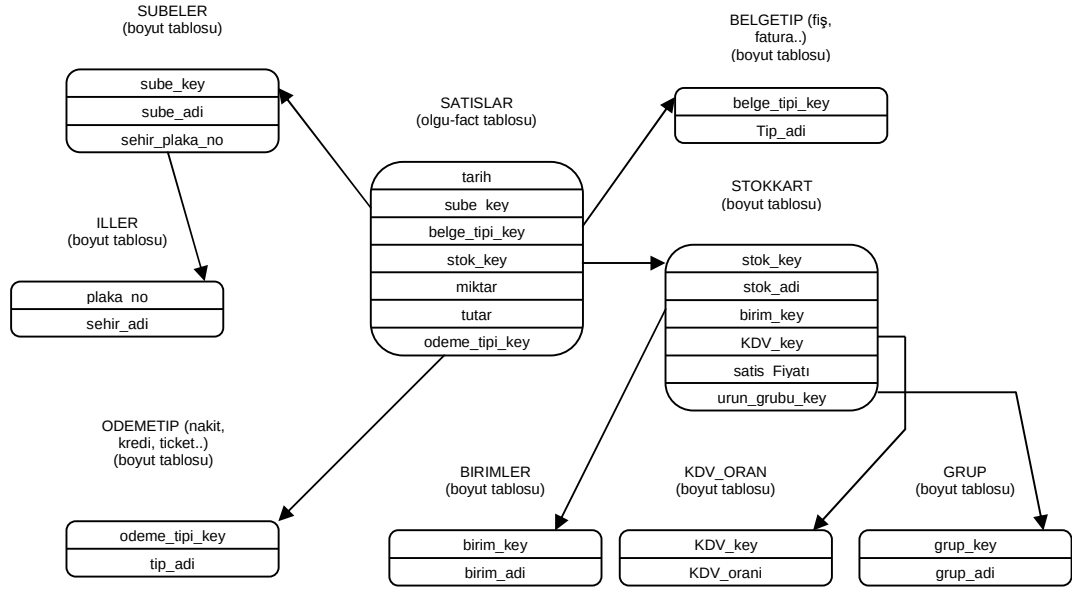
- en önemlinin içinde tutulduğu büyük bir merkezi tema tablosu (olgu tablosu)
- her biri tek boyut için kullanılan küçük yardımcı tablolar kümesi (boyut tabloları) yer almaktadır [10].

Şekil II.2’de satışlar veri ambarı için hazırlanmış bir yıldız şema örneği yer almaktadır.



Şekil II.2 Satışlar veri ambarı için *yıldız (star)* şeması örneği [10].

Kar tanesi şema (Snowflake schema): Kar tanesi şeması, bazı boyut tablolarının normalize edildiği, bundan dolayı verinin ek tablolara doğru bölündüğü, yıldız şema modelinin değişik bir biçimidir [10]. Şekil II.3’de satışlar veri ambarı için hazırlanmış bir kar tanesi şeması örneği yer almaktadır.



Şekil II.3 Satışlar veri ambarı için *kar tanesi (snowflake)* şeması örneği [10].

II.1.4.1. Çok Boyutlu Veri Modelinde OLAP Operasyonları

Han ve Kamber'e göre [10], çok boyutlu modelde veriler çoklu boyutlara organize edilmiştir ve her boyut, kavram hiyerarşisi tarafından tanımlanan çok boyutlu soyutlamalar içermektedir. Bu organizasyon kullanıcılara, veriyi farklı perspektiflerden inceleme esnekliği sağlar. Belirli sayıda OLAP veri küpü işlemleri, bu farklı incelemeleri gerçekleştirmek için, eldeki verinin etkileşimli sorgusu ve analizine imkan veren biçimde mevcuttur. Bundan dolayı OLAP, etkileşimli veri analizi için kullanımı rahat bir ortam sunmaktadır [10].

OLAP basit olarak 3 aşamadan oluşur: (1) veri ambarından veri seçimi, (2) veri küpü kurma ve, (3) veri küpü üzerinde çevrim içi analizi uygulama. Çok boyutlu veri küpleri üzerinde bir çok OLAP algoritması beceri ile uygulanmaktadır; detaylandırma (roll-up), detay azaltma (drill-down), dilimleme ve parçalama (slice and dice), çevirim (pivot- rotate), vb [13].

OLAP, veri analizine yardımcı olan bir veri toplama biçimidir. Veri madenciliği ise büyük ve ham olan verilerin için değerli olan bilgiyi bulmak, yeni bilgileri keşfetme biçimidir. OLAP araçları daha çok veri analizi için uygundur ve bu amaca hizmet için tasarlanmışlardır. Veri madenciliği OLAP teknolojisinden çok daha ileri bir kavramdır. Çünkü OLAP da olmayan birliktelik (association), sınıflama (classification), tahmin (prediction), kümeleme (clustering) ve zaman serili analizler (time-series analysis) sadece veri madenciliğinde yer alan kavramlardır. Veri madenciliği veri ambarları içinde yer alan verilerden analiz amacıyla faydalanır.

Ayrıca çok boyutlu analizlerde, çoklu ortam uygulamalarında kullanılan veriler üzerinde, text tabanlı işlem yapma kabiliyeti vardır [10].

II.2. VERİ SEÇİMİ, ÖNİŞLEME VE İNDİRGEME

Veri madenciliği yapılabilmesi için, birinci bölümünde tanımlanan *veritabanlarında bilgi keşfi* süreçlerinin veritabanlarında veya veri ambarlarında tutulan verilere sıra ile uygulanması gerekmektedir. Her bir süreç tamamlandıktan sonra bir sonraki sürecin başlatılarak veri madenciliği aşamasına ulaşılmalıdır. Veri madenciliği aşamasında veri madenciliği tekniklerinden verilere ve elde edilmek istenen sonuca uygun olan teknik seçilerek uygulanır. Veri madenciliği uygulaması sonucunda ortaya çıkan örüntüler değerlendirilerek, en son aşamada kullanıcıya grafik ortamlarda gösterimi yapılır.

II.2.1. Veri Seçimi

Veri seçimi (veri toplama) aşaması veri madenciliğinin ilk aşamasıdır. Bu aşamada yapılan işlem, Bölüm II.1’de detaylı olarak anlatılan veri depolama ortamlarındaki (veritabanları – veri ambarları) verilerin veri madenciliği uygulanması için mümkün olduğu kadar tek bir veritabanı veya veri ambarı içinde toplanmasıdır.

Bu işlem bir veri seçme işlemi olmasına rağmen toplanan verinin tek bir ortamda birleştirilmesinden dolayı veri bütünleştirme de bu aşamanın bir sürecidir. Veri madenciliğinde birden fazla veri depolama alanındaki dataları (farklı kaynaklı veriler) uygun bir veritabanında birleştirmek için sıklıkla veri bütünleştirme işlemine gereksinim duyulur [10]

Veri seçimi aşamasında yapılması gerekenler;

- farklı ortamlardaki verilerin mevcut yapılarının incelenmesi ve tablo yapılarının ortaya çıkarılması,
- veri madenciliği çalışmasında hedeflenen sonuca ulaşmak için gerekli verilerin, veri madenciliği uygulamak için belirlenen veri depolama ortamına, veri transfer teknikleri kullanılarak transfer edilmesidir.

Veri transfer yöntemleri veri kaynağı olarak belirlenen veri depolama ortamlarına göre değişkenlik gösterir. Her veri depolama ortamındaki veriye erişim için farklı erişim araçları olabilir veya aynı erişim aracı (Örn :ODBC – Open

Database Connectivity – Açık Veritabanı Bağlantısı) birden fazla veri ortamına erişim imkanı tanıyabilir.

II.2.2. Veri Önişleme

Veri madenciliğinin uygulanabilmesi için bir bütün halinde verilerin bir depolama ortamında bulunması ön koşuldur. Veri madenciliği farklı formatlarda çok sayıda depolama ortamında bütün halindeki veriler içinden önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılması için araçtır. Günümüzdeki istatistiksel raporlamada yaşanan zorluklar içinde eldeki verinin çokluğu da yer almaktadır, veri madenciliği yöntemleri bu tarz raporlamalarda işletme ve kurumları oldukça rahatlatmaktadır. Elektronik ortamlardaki veri depolama ve bilgisayarlardaki veri işleme hızlarındaki artış veri madenciliğini cazip duruma getirmiştir. Ama bu kadar büyük veriyi işlemek yine de zaman almaktadır, bu nedenle veri madenciliğinin uygulanabilmesi verilerin önişlemeden geçirilmesi, farklı bir ifade ile verilerin veri madenciliği için hazır duruma getirilmesi veri madenciliğinin en önemli aşamalarındandır.

Büyük veri grupları çoğunlukla eksik, kirli ve hatalı veri kayıtları içerir. Bu tip problemleri veriye sahip olmayan veri grubu ile çok nadir karşılaşılmaktadır [6].

Veri seçme ve veri önişleme aşamaları veri madenciliğinde en fazla zamanın harcandığı aşamalardır. Veri madenciliği projelerinde veri temizleme, bütünleştirme ve dönüşümü yoğun kaynak kullanımının yapıldığı adımdır [14]. Veri önişleme; veri madenciliği için verilerin daha yararlı bir hale getirilmesi amacıyla verilerin işlenmesidir [6].

Veri madenciliğinde üzerinde çalışma yapılan verinin kalitesi çok önemlidir. Veri hatalı, tutarsız, eksik niteliklere sahipse, veri madenciliğinin güvenilirliğin artırılması için, veri üzerinde veri temizleme ve önişleme yapılmalıdır. Eğer ki veri temizleme ve önişleme yapılmazsa veya çok yüzeysel yapılırsa hatalı girdi verileri ile veri madenciliği işlemi ile elde etmek istediğiniz sonuçlar da hatalı olacaktır. Veri önişleme aşaması, genellikle yarı otomatik yapıya sahip zaman kullanımının fazla olduğu bir veri madenciliği aşamasıdır [6].

Veri temizleme ve *veri dönüşümü* işlemleri veri önişleme aşamasını oluşturmaktadır. Veri temizlemenin amacı gürültülü ve ilgisiz verinin veri setinden çıkarmaktır. Veri dönüşümünün amacı ise, kaynak veri içindeki farklı biçimdeki veri tip ve değerlerini yapılacak veri madenciliği çalışması doğrultusunda değiştirmektir. Veri önişlemede uygulanan çeşitli teknikler vardır [14].

II.2.2.1. Veri Tipi Dönüşümü

Veri tipi dönüşümü, basit olarak veri tipi değişimidir. Örnek olarak, boolean tipdeki bir veriyi integer tipine dönüştürme işlemi verilebilir. Bu dönüşümün sonucunda, bazı veri tabanı algoritmalarında boolean veri yerine integer veri kullanılması daha iyi verim alınmasını sağlayabilir.

II.2.2.2. Gruplama

Bazen bir kolon içinde oldukça fazla oran da tekrarlayan benzer veriler olur. Bu verileri bir kaç grup içine indirgeme işlemi uygulanarak verinin kalitesi artırılır. Gruplama tekniği ile yorumlamanın daha kolay olması sağlanabilir.

II.2.2.3. Kayıp Değerleri Yönetmek

Birden fazla veritabanından gelen veriler tek bir tablo içinde birleştirildiğinde veri alanlarının bazılarında veri olmayabilir. Operasyondan gelen verinin ilgili veri alanı boş bırakıldığından veya farklı veritabanından gelen veride bu tip bir veri alanı bulunmadığından bazı kayıtlarda bu alan boş kalabilir. Bu durumu düzeltmek için,

- kayıp değerler en çok kullanılan değerler ile doldurabilir,
- bir kayıta çok oranda kayıp değer varsa kayıtın tamamı silinebilir,
- kayıp değerlerin içi el ile doldurulabilir,
- en olası ortalama değer ile doldurulabilir [10].

II.2.2.4. Gürültülü Veri

Gürültülü veri, veri kümesi içinde yer alan ama veri madenciliği uygulamasında kullanılmayacak bir anlam içermeyen verilerdir. Gürültülü veriden kurtulmak için veri temizleme işlemi uygulanır. Gürültülü verilerin teşhis edilmesi amacıyla histogram, kümeleme analizi ve regresyon gibi teknikler kullanılabilir [6].

Uygunsuz değerler kümeler ile belirlenebilir. Benzer değerler aynı grup veya küme içinde yer alırken, uygunsuz değerler kümelerin dışında yer alacaktır, kümelerin dışında kalan bu değerler veri madenciliği uygulanacak veri bütünlüğünden çıkarılır [6].

II.2.3. Veri İndirgeme

Veri indirgeme fazla olan bazı değişkenlerin atılması ve birleştirilmesi veya kümeleme yolu ile veri büyüklüğünün azaltılması amaçlanır [6]. Veri indirgeme yöntemi ile çok büyük miktarda verinin analiz edilmesi uzun süreceğinden, veri bütünlüğünü bozmadan verinin tamamını temsil edecek az miktarda veri alınır ve analiz edilir. Bu tekniğin içeriğinde veri birleştirme (data aggregation), veri küpü (data cube), boyut indirgeme (dimension reduction), veri sıkıştırma (data compression), çokluk azaltımı (numerosity reduction), ayrıklaştırma (discretization) ve kavram hiyerarşisi üretme (concept hierarchy generation) gibi yöntemler yer almaktadır [10].

II.3. VERİ MADENCİLİĞİ

Veri madenciliği; önceden bilinmeyen, veri içinde gizli, anlamlı ve yararlı örüntülerin büyük ölçekli veritabanlarından otomatik biçimde elde edilmesini sağlayan veri tabanlarında bilgi keşfi süreci içinde bir adımdır [4].

Veriyi analiz eden kişi için farklı nesnelere uygun olacak çeşitli veri madenciliği görevleri (*tasks*) bulunmaktadır. Bu görevler *tahmin edici (predictive)* ve *tanımlayıcı (descriptive)* olmak üzere iki ana başlık altında modellenmektedir [15].

Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır [16]. *Sınıflama (classification)*, *gerileme (regression)* ve *sapma (deviation)* madenciliği tahmin edici tekniklerden bazılarıdır [17].

Tanımlayıcı modellerde ise karar vermeye yardım edebilecek, 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 [16]. *Kümeleme (clustering)*, *birliktelik kuralı (association rule)* ve *ardışıl örüntü (sequential pattern)* madenciliği tanımlayıcı tekniklerden bazılarıdır [17].

II.3.1. Sınıflama ve Gerileme-regresyon

II.3.1.1. Sınıflama

Dağınık bir yapıda bulunan verilere sınıf niteliğinin uygulanması sürecidir [17]. Sınıflama algoritması, ortak özelliklere sahip kayıtların farklı sınıflar içine aktarılmasını belirleyen algoritmadır. Sınıf olmak için her kaydın sınıf içinde yer alan diğer kayıtlarla belirlenmiş bir ortak özelliği olması gerekir.

Sınıflama sorgusu kullanılarak bir kaydın daha önceden nitelikleri belirlenmiş bir sınıfa girmesi amaçlanmaktadır [18]. Sınıflama algoritması öğrenme verilerini kullanarak hangi sınıfların var olduğu ve bu sınıflara girebilmek için kayıtların hangi özelliklere sahip olması gerektiğini otomatik olarak keşfeder [8, 17].

Sınıflama algoritmaları iki şekilde kullanılır.

Karar Değişkeni ile Sınıflama: Seçilen bir niteliğin (bu niteliğe karar değişkeni adı verilir) aldığı değerlere göre sınıflama işlemi yapılır. Veritabanındaki kayıtlar karar değişkeni olarak belirlenen nitelik değerlerine göre sınıflara ayrılır [3].

Örnek ile Sınıflama: Bu sınıflama biçiminde veritabanındaki veriler iki kümeye ayrılır, kümelerden biri pozitif, diğeri negatif verileri içerir [3].

Sınıflama algoritmasının kullanım alanları, banka kredisi onaylama işlemi, kredi kartı sahteciliği tespiti ve sigorta risk analizidir [3].

II.3.1.2. Gerileme – regresyon

Gerileme genellikle geçmişteki değerleri temel alarak gelecekteki değerleri tahmin etmek için kullanılan tahmin edici modeller sınıfında yer alan bir tekniktir. *Doğrusal gerileme* tekniğinde, girdi verisi ile çıktı verisi arasında doğrusal bir ilişki olduğu varsayılır [19].

Sınıflama ve regresyon, önemli veri sınıflarını ortaya çıkaran veya gelecekteki veri eğilimlerini geçmiş veriler yardımı ile tahmin eden modelleri kuran veri analiz yöntemleridir [10]. Sınıflama; belli özelliklere göre gruplanmış değerleri tahmin ederken, regresyon; süreklilik gösteren değerlerin tahmin edilmesinde kullanılır [10]. Örneğin, sınıflama modeli banka kredi uygulamalarının güvenli veya riskli olmalarını sınıflamak amacıyla kullanılırken, regresyon modeli geliri ve mesleği bilinen potansiyel müşterilerin herhangi bir alışveriş esnasında yapacakları harcama davranışlarını tahmin etmek için kullanılabilir. Sınıflama ve regresyon modellerinde kullanılan tekniklerin bazıları [16, 20];

- 1 - Karar ağaçları (Decision trees)
- 2- Yapay sinir ağları (Artificial neural networks)
- 3- Genetik algoritmalar (Genetic algorithms)
- 4- K-en yakın komşu (K-nearest neighbor)
- 5- Bellek temelli nedenleme (Memory based reasoning)
- 6- Naive-bayesian

II.3.2. Kümeleme

Küme veya sınıflar içindeki verinin belirlenmiş benzerlik kriterleri doğrultusunda gruplanmasıdır. Amaç, verileri alt kümelere ayırmaktır. Sınıflama algoritmasında olduğu gibi ortak özellikleri olan veriler bir kümeye girer. Alt kümelere ayırmak için keşfedilen kurallar ile bir verinin hangi alt kümeye girdiği kümeleme algoritmaları ile bulunur [8].

Kümeleme bir çok alanda kullanılmaktadır; veri madenciliği, istatistik, biyoloji ve cihaz öğrenimi, nüfus bilimi, bankacılık uygulamaları vb. Genel olarak 2 tip kümeleme vardır:

1. *Geleneksel kümeleme*; nesnelerin geometrik yapısını baz alarak kümeleme yapar.
2. *Kavramsal kümeleme*; iki parçadan oluşur:
 - birincisi; nesneleri farklılıklarına göre gruplar,
 - ikincisi; sınıflamada olduğu gibi o her sınıf için bir değişken (ifade –tanım) oluşturup, nesneleri bu değişkene göre sınıflar [10].

Kümeleme modelinde, sınıflama modelinde olan veri sınıfları yoktur [21]. Verilerin herhangi bir sınıfı bulunmamaktadır. Sınıflama modelinde, verilerin sınıfları belirlenmiştir ve yeni bir veri geldiğinde bu verinin hangi sınıftan olabileceği tahmin edilmektedir. Oysa kümeleme modelinde ise, sınıfları bulunmayan veriler belirlenen benzerlik-yakınlık kriterlerine göre gruplar halinde kümelere ayrılırlar. Küme içindeki elemanların benzerliği olmalı, kümeler arasında ise benzerliğin az olması gerekir [21].

- 1- Bölme yöntemleri,
- 2- Hiyerarşik yöntemler,
- 3- Yoğunluk tabanlı yöntemler,
- 4- Izgara tabanlı yöntemler,
- 5- Model tabanlı yöntemler.

II.3.3. Zaman Serilerindeki Örüntü

Zaman serilerindeki örüntü belli periyotta, belli sıklıkta gerçekleşen olaylardır. Belli sıklıkla tekrarlanan bu olaylar zaman serileriyle yapılan veri madenciliği algoritmaları sayesinde keşfedilir. Örneğin, her yıl mayıs ayının ortalarından eylül ayına kadar ki zaman diliminde güneş kremi ve güneş yağı alımlarında artış olması her yıl belli zaman aralığında tekrarlanan bir örüntüyü temsil etmektedir.

II.3.4. Ardışıl Örüntü

Ardışıl örüntü ise birbirleri ile ilişkisi olan ve birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Belli bir olayın gerçekleşmesinden sonra aynı olayla ilintili bir başka olayın gerçekleşmesi ardışıl örüntü kavramını oluşturur. Örneğin; otomobil alan bir kişinin kısa vadede otomobili için aksesuar alması ardışıl bir örüntüdür [8].

Tanımlayıcı tekniklerden olan *Birliktelik-İlişki Kuralları* Bölüm III'de geniş olarak verilmektedir.

BÖLÜM III

BİRLİKTELİK-İLİŞKİ (EŞLEŞTİRME) KURALI

Birliktelik kuralı, geçmiş verilerin analiz edilerek bu veriler içindeki birliktelik davranışlarının tespiti ile geleceğe yönelik çalışmalar yapılmasını destekleyen bir yaklaşımdır. 90 yılların başına kadar saklanan satış verilerinde ürün ve müşteri verisi çok nadir yer alırken, genelde mali açıdan önemli olan tutarsal gelir verilerinin depolaması yapılıyordu. 90 yılların başından itibaren veri toplama uygulamalarındaki gelişmeler doğrultusunda firmaların satış noktalarında yeni teknoloji otomatik ürün veya müşteri tanıma sistemleri (barkod ve manyetik kart okuyucular) yaygınlaşmaya başlamıştır. Bu tip teknolojik gelişmeler, bir satış hareketine ait verilerin satış esnasında toplanmasına ve elektronik ortamlara aktarılmasına olanak tanımıştır. Günümüzde süpermarketlerde, orta ve büyük ölçekli alışveriş mağazalarındaki satış noktalarında akıllı satış sistemlerinin kullanımı oldukça yaygındır. Bu satışlardan elde edilen verilerde, işlem tarihi, satın alınan ürünlere ait bilgiler (ürün kodu, miktar, fiyat, iskonto vb.) yer alır ve ayrıca hareket numarası tekildir. Bazı kuruluşlar bu tip bilgileri içeren veritabanlarını pazarlama alt yapılarının önemli parçalarından biri olarak görmekte ve bu verileri kullanmak için çaba harcamaktadırlar [3].

III.1. BİRLİKTELİK-İLİŞKİ KURALININ TANIMI

Birliktelik kuralında, müşterilerin alışveriş esnasında satın aldıkları ürünler arasındaki birliktelik-ilişki bağlarını bularak, müşterilerin satın alma alışkanlıklarının tespit edilmesi amaçlanmaktadır. Keşfedilen bu birliktelik-ilişki bağıntıları sayesinde satıcılar daha etkin ve kazançlı satışlar yapabilme imkanına sahip olmaktadır. Birliktelik kurallarının kullanıldığı en tipik örnek pazar sepeti uygulamasıdır. Süpermarket alışverişi esnasında müşteriler patates cipsi aldıktan sonra genelde aynı alışverişte kola da satın alıyorsa, bu iki ürün arasında kuvvetli bir birliktelik-ilişki kuralı var anlamı yakalanır. Bu elde edilen veri sayesinde bu ürünlere ek ürün satışı yapmak için düzenlemeler yapılabilir [20].

Birliktelik kuralı, bir ilişkide bir niteliğin aldığı değerler arasındaki bağımlılıkları, anahtarda yer almayan diğer niteliklere göre gruplama yapılmış verileri kullanarak bulur. Keşfedilen örüntüler veri kümesinde sıklıkla birlikte geçen nitelik değerleri arasındaki ilişkiyi gösterir [3]. Örneğin, bir süpermarket te ekmek ve peynir satın alınan satış hareketlerinin %75'inde zeytin de satın alınmıştır. Bu tür birliktelik-ilişki örüntüleri ancak, örüntüde yer alan öğelerin birden fazla harekette tekrarlandığında potansiyel olarak mevcut olabilirler. Gerçek hayattaki uygulamalarda veri madenciliği teknikleri milyonlarca veri üzerinde uygulandığından birliktelik-ilişki sorgusu sırasında kullanılan algoritmalar hızlı olmalıdır (Srikant ve Agrawal, 1995) [5]. Diğer veri madenciliği tekniklerinde olduğu gibi, birliktelik-ilişki sorguları etkinlik, ölçeklenebilirlik, kullanılabilirlik ve anlaşılabilirlik gibi önemli ölçütleri karşılamalıdır.

III.2. BİRLİKTELİK-İLİŞKİ KURALININ MATEMATİKSEL MODELİ

Birliktelik-ilişki kuralının matematiksel modeli Agrawal, Imielinski ve Swami tarafından 1993 yılında verilmiştir [3, 22]. Bu modelde $I = \{i_1, i_2, \dots, i_m\}$ verilen bağlamda yer alan ürün kodlarını; D bağlamda yer alan hareketleri; T bir harekette yer alan ürün kodlarını ($T \subseteq I$) (bir harekette yer alan her ürün miktar bilgisi ile değil, o harekette satın alınıp alınmadığını gösteren (alındı/alınmadı) mantıksal bir değişkenle ifade önemlidir.), tid her harekete ait tekil (unique) numarayı, k -öge kümesi k adet ürün içeren ürün kümesini temsil etmektedir. X bir ürün kümesi olmak

üzere T hareketi X ürün kümesini ancak ve ancak $X \dot{I} T$ şartını sağlıyorsa içerir. Bir birliktelik-ilişki kuralı X ve Y, I ürün kümesinin bir alt kümesi ve $X \cap Y = \emptyset$ olmak üzere $X \Rightarrow Y$ biçiminde bir bağımlılık ifadesidir. Bu ifade ile X, Y'yi belirler (X ürününü içeren hareketler kümesi Y'yi içeren hareketler kümesi içinde kapsanır) veya Y, X'e bağımlıdır denir (Y kümesinin varlığı X kümesinin var olmasına bağlıdır). Hareket numaraları gruplandırılarak elde edilen ürünler arasındaki bağımlılık ilişkisinin yüzde yüz doğru olması beklenemez. Benzer şekilde, çıkarsama yapılan kuralın eldeki hareketler kümesinin önemli bir kısmı tarafından desteklenmesi istenir. Bu nedenlerden dolayı, $X \Rightarrow Y$ eşleştirme kuralı kullanıcı tarafından minimum değeri belirlenmiş güvenilirlik (c:confidence) ve destek (s:support) eşik değerlerini sağlayacak biçimde üretilir. $X \Rightarrow Y$ eşleştirme kuralına, c güvenilirlik ölçütü ve s destek ölçütü iliştilir ve biçimsel olarak $\Phi(D) = \langle X \Rightarrow Y, c, s \rangle$ ile gösterilir. Burada D örnekleme; $X \Rightarrow Y$ birliktelik-ilişki kuralını; c eşik değeri, ilgili kuralın minimum güvenilirliğini (X ürünlerini içeren hareketlerin en az %c oranında Y içeren hareketler kümesinde yer aldığını); s ilgili kuralın, minimum desteğini (X ve Y ürünlerini içeren hareket tutanaklarının toplam hareket tutanakları içinde en az %s oranında var olduğunu) gösterir [3, 22].

Ürünler kümesi ailesini $\mathcal{I}(I)$ ile gösterelim ve, X ve Y'nin her ikisinde $\mathcal{I}(I)$ üzerinde değişebilen iki rasgele değişken olsun. $Pr(X)$, $Pr(X \cap Y)$ ve $Pr(X \cup Y)$ marjinal olasılık değerleri, sırasıyla, X kümesi içinde yer alan tüm ürünlerin herhangi bir sepet varlığında bulunma olasılığını, X ve Y rasgele değişkenlerince paylaşılan ortak ürünlerin herhangi bir sepet varlığında bulunma olasılığını, ve X ve Y rasgele değişkenlerinin birleşiminde yer alan ürünlerin herhangi bir sepet varlığında bulunma olasılığını gösterebilir. O zaman, güvenilirlik eşiği, $Pr(Y/X) = Pr(X \cap Y) / Pr(X)$ ile ve destek eşiği $Pr(X \cap Y)$ ile ifade edilir. Güvenirlik metriği, birliktelik-ilişki kuralının doğruluk derecesini, destek metriği ise kuralda yer alan öğelerin (ürünlerin) geçiş sıklığını gösterir. Yüksek güvenilirlik ve destek değerine sahip kurallara güçlü kurallar adı verilir. Birliktelik-ilişki kuralı çıkarımı büyük veritabanlarından güçlü ilişkilendirme (eşleştirme) örüntülerinin elde edilmesini gerektirir [3, 22].

Birliktelik-ilişki kuralı formüsel olarak şu şekilde tanımlanabilir;

$$A_1, A_2, \dots, A_m \rightarrow B_1, B_2, \dots, B_n \quad (III.1)$$

Buradaki, A_i ve B_j yapılan iş veya nesnelerdir. Bu kural genellikle “ $A_1, A_2, \dots, A_m$ meydana geldiğinde, sık olarak $B_1, B_2, \dots, B_n$ aynı olay veya hareket içinde yer aldığını anlamına gelmektedir [13].

Örneklendirmek gerekirse; aşağıdaki kural bir digital ürün satış mağazasının satış hareketlerinden gelmektedir.

$$\text{Ürün}(X, \text{"digital fotoğraf makinası"}) \rightarrow \text{Ürün}(X, \text{"bellek kartı"})$$

Burada X bir hareketteki değişkeni simgelemektedir. Bu kural da, digital fotoğraf makinası alan müşterinin aynı zamanda ayrıca ek bellek kartı almaya yöneldiği anlamı çıkarılmaktadır.

Başka bir örnek; aşağıdaki kural üç boyutlu bir veri ambarından gelmektedir: Yaş, Meslek ve Ürün.

$$\text{Yaş}(X, \text{"12 - 17"}), \text{Meslek}(X, \text{"öğrenci"}) \rightarrow \text{Ürün}(X, \text{"oyun konsolu - playstation"})$$

Bu kural ile, "12-17 yaşları arasındaki öğrenci en çok "oyun konsolu (playstation) almaktadır" anlamına elde edilmektedir [13].

Başka bir örnek; bir elektronik ve digital cihaz satışı yapan firmanın müşterilerinin alışveriş alışkanlıkları üzerine birliktelik kuralının formu;

$$P(X:\text{müşteri}, W) \wedge Q(X, Y) \Rightarrow \text{alış}(X, Z)$$

X, müşteri ilişki anahtarı; P ve Q, işe bağlı verinin bir parçası olan bağıl nitelik veya boyutlardan o an yüklenen değişkenler ve W, Y ve Z, müşteri X'in o ana kadar ki işlemlerinin nesne değişkenleridir.

Nitelik kuralları için arama yukarıdaki temel kurala uygun olmalıdır

$$\text{Yaş}(X, \text{"30...39"}) \wedge \text{gelir}(X, \text{"60K...69K"}) \Rightarrow \text{alış}(X, \text{"Plazma TV"}) \quad [\%4.5, \%20]$$

$$\text{Meslek}(X, \text{"öğrenci"}) \wedge \text{yaş}(X, \text{"15...20"}) \Rightarrow \text{alış}(X, \text{"Oyun Konsolu"}) \quad [\%5.2, \%59]$$

Yukarıdaki ilk kuralda, otuzlu yaşlarındaki, yıllık gelirleri 60K-69K arasında olan müşterilerin %20'ı Plazma TV satın almış ve tüm satışların %4.5 'sini temsil eder. Bu açıklamadaki %20 ifadesi *güven-confidence*, %4.5 ifadesine ise *destek-support* olarak tanımlanır. Bir sonraki kuralda, yirmi yaş altı öğrenci olan müşterilerin %59'i oyun konsolu satın almış ve tüm satışların %5.2 'ünü temsil eder. Bu açıklamaya göre ise %59 *güven-confidence*, %5.2 ise *destek-support* olarak tanımlanır.

III.2.1. Güven (Confidence) ve Destek (Support) Ölçüleri

Tüm keşfedilen örüntüler, doğruluk (destek) veya güvenilirlik kesinlik ölçülerine sahip olmalıdırlar. A ve B'nin ürün kümeleri ise, A'nın ilgili kural için kesinlik ölçütü

gösterimi $(A \Rightarrow B)$ *güven(confidence)* dir. İşe bağlı veri satırları verilmiş olsun, $(A \Rightarrow B)$ güveni aşağıdaki gibi tanımlanır.

$$\text{güven-confidence}(A \Rightarrow B) = (A \text{ ve } B\text{'nin bulunduğu satır sayısı}) / (A\text{'nın bulunduğu satır sayısı}) \quad (\text{III.2})$$

İşe bağlı veri kümemizin, bir yapı-marketin küçük ev aletleri bölümünde satılan ürünler olduğunu varsayalım. Birliktelik kuralı için *güven-confidence* %73; sarjlı matkap alan müşterilerin %73'ü tornavida takım seti de satın aldığı anlamına gelir.

$$\text{alış}(X, \text{"sarjlı matkap"}) \Rightarrow \text{alış}(X, \text{"tornavida seti"})$$

Güven değeri %100 olması demek kural bütün veri analizlerinde doğrudur ve bu kurallara *"kesin(exact)"* denir.

Örüntünün potansiyel yararlılığı örüntünün ilgililiğini tanımlamada bir etmendir. İlişkilendirelen örüntünün kullanılarak bulunan iş ile ilgili veri satırlarının yüzdesi, hangi örüntünün doğru olduğunu gösterir. A ve B ürün kümeleri, ilişki kuralları $A \Rightarrow B$ olarak gösterilsin,

$$\text{destek-support}(A \Rightarrow B) = (A \text{ ve } B\text{'nin bulunduğu satır sayısı}) / (\text{toplam satır sayısı}) \quad (\text{III.3})$$

İşe bağlı verimizin elektronik firmasının bilgisayar bölümü olduğunu varsayalım. İlgili kural için *destek-support* %5 demek, bütün müşterilerin %5'i hem sarjlı matkap hem de tornavida seti almış anlamına gelir [10].

Güven-confidence ve destek-support ile ilgili başka bir örnek; tablodaki alış hareketlerine bakarak toplam alış hareketlerine göre {Bisküvi, Yoğurt} ile Kola arasındaki ilişki şu şekilde açıklanabilir; (Tablo II.1)

Tablo II.1 Ürün hareket tablosu [23].

ID	Ürünler
1	Kola, Ekmek, Bisküvi, Yoğurt
2	Kola, Bisküvi, Ekmek, Balık
3	Meyve suyu, Ekmek, Bisküvi, Yoğurt
4	Ekmek, Bisküvi, Yoğurt
5	Kola, Meyve suyu, Bisküvi, Yoğurt

$$\begin{aligned} \text{Destek-support} &= \frac{(\text{Bisküvi, Yoğurt, Kola})}{\text{Toplam hareket}} = \frac{2}{5} = 0.4 \\ \text{Güven-confidence} &= \frac{(\text{Bisküvi, Yoğurt, Kola})}{(\text{Bisküvi, Yoğurt})} = \frac{2}{4} = 0.5 \end{aligned}$$

Bu eşitlikliklerden de anlaşılacağı gibi, {Bisküvi, Yoğurt} $\Rightarrow$ Kola kuralı %40 destek-support, %50 güven-confidence ölçülerine sahiptir [23].

İlişkili kurallar, kullanıcının tanımladığı minimum *güven-confidence* ve *destek-support* eşik değerlerini sağlarlarsa *güçlü birliktelik-ilişkili kuralları (strong association rule)* olarak tanımlanır. Az destekli değerlerde kurallar *gürültü (noise)* olarak kabul edilir [10].

Birliktelik kuralının kullanım alanları, market satış analizlerinde, ticarete, mühendislikte, tıpta ve finans şeklinde sıralanabilir. *Sepet analizi (Market basket analysis)* en çok kullanıldığı alanlardan biridir. Müşteri alım alışkanlıklarına ve perakendecilik esaslarına göre kararlar alınmasını sağlar; hangi ürün indirimde konacağı, katalogların nasıl tasarlanacağı, raflarda ürünleri nasıl dizileceği vb [13]. Örnek olarak sepet analizi yöntemi farklı raf dizimlerinin olabilmesine olanak tanır. Bir stratejide, birlikte sık olarak alınan ürünler raflarda yakın yerlere dizilebilirler. Bilgisayar alan müşterilerin yoğunluğu yazılım da alma eğilimindedir ise bu ürünlerin yakın yerlere konulması iki ürünün satış oranlarını da artırabilir. Diğer alternatif bir stratejide, bilgisayar ve yazılım ürünlerini markete ait bir rafın başlangıcına ve sonuna koymak, müşteriyi kandırma metotlarından birisi olabilir. Çünkü müşteri raf boyunca başka ürünlere bakarak ilerler ve bunları satın alma olasılığı doğar [10].

Sepet analizi yöntemi, birliktelik-ilişki kurallarının uygulandığı tek alan değildir. Jan ve Kamber'e [10] göre birliktelik-ilişki kuralı aşağıdaki kriterleri taşıyan çok değişik yollar ile sınıflandırılabilir:

- Kuralda yakalanan değerlerin tiplerine göre: Eğer bir kural nesnelerin varlığı ve yokluğu arasındaki ilişki ile ilgili ise, bu durumda *mantıksal(boolean) birleştirme kuralıdır* denir. Bu kurallar sepet analizinden elde edilirler [10].

Kurallar nicel nesneler ya da attribute değerleri arasındaki ilişkiyi tanımlıyor ise *nicel birleştirme kuralıdır*. Bu kurallarda, nesneler için nicel değerler ya da attributeler, aralıklara bölünmüşlerdir. Aşağıdaki kural böyle bir kuraldır (X değişkeni bir müşteriyi temsil etmektedir) [10]:

$\text{yaş}(X, "30..39") \wedge \text{gelir}(X, "42K..48K") \Rightarrow \text{alışlar}(X, \text{Yüksek çözünürlüklü TV})$

- Kuralın içerdiği verinin boyutlarına göre: Bir birliktelik-ilişki kuralındaki nitelikler ya da nesneler bir boyutu temsil ediyorsa, o zaman kural *tek boyutlu birleştirme kuralıdır* denir [10].

$\text{alışlar}(X, "bilgisayar") \rightarrow \text{alışlar}(X, "finansal_yönetim_yazılımı")$

Kural tek boyutludur. Eğer kural iki ya da daha çok boyuta referans ediyor ise, örneğin alışlar, alışveriş_zamanı, müşteri_kategorisi boyutları gibi, o zaman kural *çok boyutludur*. Üst taraftaki 1. kural çok boyutlu birleştirme kuralıdır. 3 boyut içerir: yaş, gelir, alışlar [10].

- Kural kümesindeki soyutlama katmanlarına göre: Birliktelik-ilişki kuralları analizi için bazı metotlar farklı soyutlama katmanlarında kurallar bulabilir. Örneğin, bir kural kümesi aşağıdaki kurallardan oluşuyor olsun [10]:

yaş(X, "30....39") $\hat{a}$ alışlar(X, "laptop_bilgisayar")

yaş(X, "30....39") $\hat{a}$ alışlar(X, "bilgisayar")

alınan ürünler farklı soyutlama katmanlarında referans edilmişlerdir (Örneğin bilgisayar, laptop bilgisayarın üst-katman soyutlamasıdır.) [10].

III.2.2. Birliktelik-İlişki Kuralının Aşamaları

Birliktelik-ilişki kuralı madenciliği 2 aşamalıdır:

- *Tüm sık geçen nesne kümelerinin bulunması*: Tanıma göre her nesne kümesinin sık geçenler kümesinde yer alabilmesi için, her nesnenin destek değerinin önceden tanımlanmış olan min_destek değerinden büyük olması gerekir.
- *Sık geçen nesne kümelerinden güçlü ilişki kurallarının yaratılması*: Tanıma göre, bu kurallar min_destek ve min_güven durumunu sağlamalıdır [10].

Birliktelik-ilişki sorgusu algoritmalarının performansını belirleyen adım birinci adımdır. Sık geçen öge kümeleri belirlendikten sonra, eşleştirme kurallarının bulunması düz bir adımdır.

Birliktelik-ilişki kuralınının uygulandığı algoritmalarını bazılar şunlardır; AIS, Apriori, DHP, Partition.

Bu algoritmalarından en ilkeli AIS (Agrawal, Imielinski ve Swami) [22], en bilineni Apriori (Srikant ve Agrawal) algoritmasıdır [3, 24].

III.3. AIS ALGORİTMASI

AIS (Agrawal, Imielinski ve Swami) algoritmasında üretilen eşleştirme kurallarının sağ kesiminde sadece bir elemanlı ürünler kümesi yer almabilmektedir ($X \Rightarrow I_k$ biçimindedir) [22]. AIS algoritmasının tersine diğer algoritmalar birden fazla

elemana sahip kurallar üretebilmektedir. Apriori algoritmasında k öğeli sık geçen öge küme adayları, $(k-1)$ öğeli sık geçen öge kümelerinden faydalanılarak bulunur. Ancak bu algoritma veritabanının pek çok kere taranmasını gerektirmektedir.

III.4. APRIORI ALGORİTMASI

Apriori, boolean ilişki kuralları için geçerli bir veri madenciliği algoritmasıdır. Algoritmanın ismi, bilgileri bir önceki adımdan aldığı için “prior” anlamında Apriori’dir. Bu algoritma özünde iteratif (tekrarlayan) bir niteliğe sahiptir [10] ve hareket bilgileri içeren veritabanlarında sık geçen öge kümelerinin keşfedilmesinde kullanılır

Sık geçen öge kümelerini bulmak için birçok kez veritabanını taramak gerekir. İlk taramada bir elemanlı minimum destek ölçütünü sağlayan sık geçen öge kümeleri bulunur. İzleyen taramalarda bir önceki taramada bulunan sık geçen öge kümeleri aday kümeler adı verilen yeni potansiyel sık geçen öge kümelerini üretmek için kullanılır. Aday kümelerin destek değerleri tarama sırasında hesaplanır ve aday kümelerinden minimum destek ölçütü sağlayan kümeler o geçişte üretilen sık geçen öge kümeleri olur. Sık geçen öge kümeleri bir sonraki geçiş için aday küme olurlar. Bu süreç yeni bir sık geçen öge kümesi bulunamayana kadar devam eder [3].

Bu algortmada temel yaklaşım eğer k -öge kümesi minimum destek ölçütünü sağlıyorsa, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar.

Bu tez çalışmasında geliştirilen veri madenciliği uygulama yazılımında Apriori algoritması kullanılmıştır. Bölüm IV.3.3.2’deki tez çalışmasının içinde Apriori algoritması daha geniş olarak verilmektedir.

III.5. DHP ALGORİTMASI

DHP (Direct Hashing and Pruning) algoritması da k -öge kümesi adaylarını $k-1$ elemanlı sık geçen öge kümelerinden elde eder, Apriori algoritmasından farklı olarak sık geçen küme adaylarını (arama uzayını) azaltır. DHP algoritması da veritabanının bir çok kere taranmasını gerektirir [25].

III.6. PARTITION ALGORİTMASI

Partition algoritması giriş/çıkış işlemlerini, veritabanını sadece iki kez okuyarak en aza indirger. Bu algoritma veritabanını bellekte ele alınabilecek küçük parçalara böler. İlk geçişte potansiyel olarak sık geçen öge kümelerini bulur, ikinci geçişte ise öge kümelerinin destek değerleri hesaplanır [26].

BÖLÜM IV

BİRLİKTELİK KURALLARINA İLİŞKİN UYGULAMA YAZILIMININ GELİŞTİRİLMESİ

Hizmet ve ürün satışı sağlayan firmalar (tez süresince verileri incelenecek örnek firma; otel, restaurant, pub ve pastane hizmetleri vermektedir) farklı cihaz ve yöntemler ile satış noktalarında hizmet ve ürün satın alan müşterilerinin işlem ve alışveriş verilerini veri ambarları içine kayıt etmektedirler. Burada geliştirilen uygulama yazılımının amacı; veri madenciliği uygulayarak, veri ambarı içinde tutulan verilerden anlamlı, faydalı, daha önceden bilinmeyen bilgilerin ortaya çıkarılması ve ortaya çıkan bu bilgilerden faydalanılarak firmalar yararına müşteri odaklı yeni süreçlerin başlatılmasıdır.

Bu geliştirilen yazılım, ülkemizde hizmet ve ürün satışı sektöründe yer alan şirketlerin toplamış oldukları ham müşteri işlem ve alışveriş verilerini, veri madenciliği modelleri kullanılarak farklı ve yararlı bilgi bütünlükleri şekline dönüştürme ve sektöre farklı bir bakış açısı ile yeni bir görüş getirilmesini sağlar.

IV.1. ARAŞTIRMA YÖNTEMİ

Bu tez kapsamında verileri kullanılacak örnek firmanın ilk aşamada pastane lokasyonlarındaki satış verileri üzerinde veri madenciliği yapılmasının uygun olduğuna karar verilmiştir. Bu doğrultu da pastane lokasyonlarında yıl içinde farklı

dönemlerde gerçekleşen satış verileri üzerinde birlikte satın alınma tercihi yapılan ürünlerin tespit edilmesi, tespit edilen bu ürün bilgileri ile;

- ilgili satış dönemi öncesinde verimli bir üretim planlaması yapmak,
- birbiri ile ilişki halindeki bu ürünler için özel satış kampanyaları düzenlemek,
- daha fazla satış için birlikte satın alınan bu ürünlerin dahil olduğu ürün gruplarında ürün çeşitliliği artırmak,
- satış lokasyonu (pastane) içindeki reyon düzenlemelerinin elde edilen veriler doğrultusunda düzenlemek,
- satış personelinin daha fazla ürün satışı gerçekleştirebilmesi için bu ilişki bilgileri ile desteklemek gibi amaçlara ulaşmak hedeflenmiştir.

Bu amaçlara ulaşmak için ilgili satış verileri üzerinde veri madenciliği kullanılarak Birliktelik-ilişki (eşleştirme) kuralı algoritmalarından Apriori algoritmasının uygulanmasına karar verildi. Bu algoritmanın uygulanması için, ön hazırlık olarak;

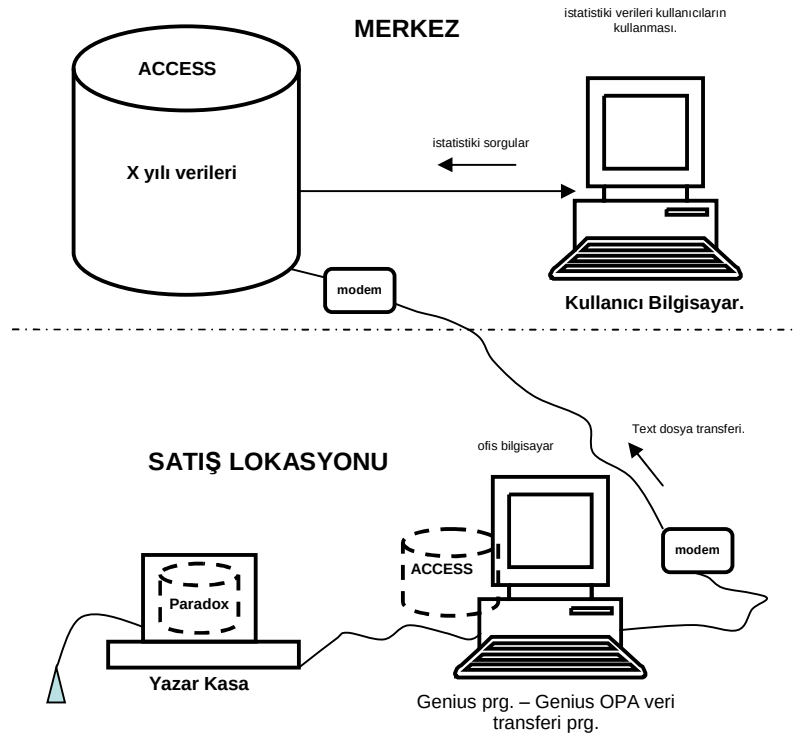
- satış verileri ile çalışma yapılacak toplam satış noktası lokasyon sayısı belirlendi,
- veri kaynağı tespiti yapıldı,
- bu kaynaklardan, verinin tutulduğu ortam tespit edildi,
- ilgili veri kaynaklarında yer alan satış verisinin merkez ofise nasıl alındığı araştırıldı,
- merkeze alınan verilerin merkezde bulunduğu veri tabanı ortamı incelendi,
- merkez veritabanında yer alan tablo ve kayıt yapıları incelendi,
- merkez veritabanındaki tablolar arasındaki ilişkiler tespit edildi.

Yapılan ön hazırlık araştırması sonucunda; ilgili firmanın 7 pastanesinin satış verileri üzerinde çalışma yapıldı. Bu satış lokasyonlarının 2002 yılı 9.ayından, 2006 yılı 2.ayına kadar ki satış verilerine ulaşıldı. Bu satış noktalardan bazılarının bu süreç içinde kapandığı, bazılarının ise 2002 yılında satışta olmadığı daha sonraki zaman dilimi içinde açılarak satışa başladığı tespit edildi. Bu lokasyonlardan 5 adedi İstanbul, 2 adedi Ankara da faaliyet göstermektedir.

Lokasyonda satış verileri akıllı yazarkasalar tarafından üretilmektedir. Bu yazarkasalar ağ ortamında ağ iletişim cihazlarını kullanarak çalışma alanı (domain) veya çalışma grubu (workgroup) içinde bir öge olarak yer alabilir, diğer grup veya alan üyeleri ile veri alış verişinde bulunabilirler.

Bu satış terminaline bağlı giriş/çıkış üniteleri (klavye, barkod okuyucu) ile satış verisi (fiş veya fatura dökümü tamamlandıktan sonra) cihaz üzerinde yüklü olan

Paradox veritabanına aktarılmaktadır. Normalde bu satış terminali ethernet portu ile ağ ortamında yer alan bir kişisel bilgisayara bağlıdır. Bilgisayar üzerinde çalışan Genius ismi verilen (Microsoft Visual Basic 6.0 da geliştirilmiş) bir arayüz programı ile satış terminalinin Paradox veritabanında yer alan veriler otomatik olarak bilgisayar üzerinde yüklü olan Microsoft Access veritabanlarına aktarılmaktadır. Eğer ki yazarkasa ağ yapısından koparsa üzerinde yer alan veritabanı (Paradox veritabanı) sayesinde uzun süre çevrim dışı olarak çalışabilir. Şekil IV.1’de yazarkasadan ofis bilgisayarına, ofis bilgisayarından merkez veritabanına transfer görülmektedir.



Şekil IV.1 Yazarkasadan ofis bilgisayarına, ofis bilgisayarından merkez veritabanına transfer işlemi.

Bilgisayar üzerinde yer alan Genius arayüz programı;

- lokasyondaki satış ile ilgili raporların (ürün satış raporları, mali raporlar, stok raporları) alınabilmesine,
- yeni satış stok kartı açılabilmesine,
- fiyat değişikliği yapılabilmesine imkan tanımaktadır.

Bu program ayrıca ek bir modülü ile (Genius OPA) Access veritabanı içindeki stok kart verilerini, fişli ve faturalı satış verilerini sabit bir desen yapısı içinde text format ta dosya şeklinde oluşturabilmektedir. Bu şekilde dışarıya veri aktarabildiği

gibi aynı şekilde mevcut Access veritabanına da veri alabilmektedir (stok kartları, müşteri ve cari kart bilgileri, fiyat değişiklikleri, reyon, departman, grup bilgileri, vb.).

Örnek niteliğinde satış verileri (fişli satış verileri) ile ilgili olarak oluşturulan GNDSALES.GTF isimli text dosyanın desen yapısına ilişkin bir örnek kısaca Şekil IV.2’de yer almaktadır [27].

Satış lokasyonu tarafında ilgili arayüz programı ile oluşturulan bu dosyalar, ilgili lokasyon VPN (Virtual Private Network – Özel Sanal Ağ) içinde ise standart dosya kopyalama komutlarını içinde barındıran görevler ile, eğerki lokasyona çevirmeli ağ ile ulaşıyorsa uzaktan erişim (örn: pcanywhere) programları ile merkez ofiste yer alan bir server üzerine lokasyon ayrımlı (her bir lokasyona ait text dosyalar lokasyon ismi ile aynı dizin içinde tutulmaktadır) olarak transfer edilmektedir. Kurum içinde Microsoft Visual Basic 6.0 da geliştirilen bir arayüz programı ile ilgili text dosya okunmakta ve veriler Microsoft Access 97 veritabanı içindeki ilgili tablolara aktarılmaktadır. Farklı yıllara ait veriler ayrı Access veritabanlarında tutulmaktadır.

Access veritabanlarında yer alan tablolar,

- stok kartları, (stok kodu, adı, fiyatı, bolum_id, ..)
- stok kartların bağlı olduğu üretim bölüm bilgileri, (Çikolata imalat, Pasta imalat, vb.)
- stok kartların bağlı olduğu üretim alt bölüm bilgileri, (Madlen, Pralin, Tek Pasta, Tart, Donmuş Pasta, vb.)
- stok kartların bağlı olduğu grup bilgileri, (Pasta-Turta, Çikolata-Şokola, Bisküvi, Tuzlu, ..)
- stok kartlarda kullanılan birim bilgileri, (Adet, Kg., Lt., vb.)
- şube bilgileri, (kodu, adı, şehri, ..)
- kasa satışı hariç lokasyondaki hareket belgeleri başlıkları, (mal alım irsaliyesi, iade irsaliyesi, başka şubeye transfer fişi, sayım belgeleri vb.. belgelerin başlık bilgileri; tarih, lokasyon bilgisi, belge cinsi, iskonto oranı, vb.) –
- kasa satışı hariç lokasyondaki hareket belgeleri detayları, (belge içindeki ürün bilgileri, miktarları, kalem iskontoları, vb.)
- kasa satışı belgeleri başlıkları, (fiş, fatura)
- kasa satışı belgeleri detayları, (belge içindeki ürün bilgileri, miktarları, kalem iskontoları, vb.)
- kasa satışı belgeleri ödeme detayları, (nakit, kredi, ticket ödeme bilgileri)

Dosya ismi : GNDSALES.GTF

Dosya tipi : Text

1. Satır işareti: 03 Başlangıç kolonu:12 "Para üstü,ödeme v.s." 0:Paranın müşteriden alındığını 1:Para üstü verildiğini gösterir.
- 2:Döviz ile ödeme yapıldığını gösterir.

Versiyon bilgisi	V2.10.5			
Satır İşareti	Açıklaması	Türü	Başlangıç Kolonu	Uzunluk
01	Hareketin yapıldığı POS	Integer	4	6
01	Hareket tarihi (kasa tarihi convert edilir)	Date	10	14
01	Kasanın sürekli artırıp gün içinde resetlediği no	Long	24	12
01	011054 (saat 01:10:54 da yapılan bir hareket)	Long	36	12
01	Mağaza no	Integer	48	6
01	İşlemi yapan kasiyer	Text	54	8
01	Belge tipi, 0: Fiş 1: Fatura 2: İade 3: diplomatik 4: irsaliye	Byte	62	1
01	0 : Geçerli belge 1 : Kullanıcı tarafından iptal edildi. 2 : Uygulama tarafından iptal edildi (belge gecislerinde) 3 : Askıya alındığından dolayı iptal edildi 4: Program acilirken yarım kalan belgeden dolayı iptal edildi (power-on)	Byte	63	1
01	Fatura no (Seri no) kasiyer girecek	Text	64	10
01	Hareket başlangıç zamanı (kasiyer performans ölçümleri için)	Integer	74	6
01	Hareket bitiş zamanı (kasiyer performans ölçümleri için)	Integer	80	6
01	Toplam satır sayısı	Integer	86	6
01	KDV dahil genel toplam	Currency	92	15
01	Belgenin KDV toplamı	Currency	107	15
				
01	Müşteri numarası	Text	212	24
02	Satış hareketinin, belge içindeki sıra numarası (1'den başlar)	Integer	4	6
02	Stoğun kodu	Text	10	24
02	Hareket tip 0: Normal satır 1: iptal satırı	Byte	34	1
02	KDV referansı (0-9) (10 adet fiyata KDV dahil olup,olmadığı)	Byte	41	2
02	Uygulanmış KDV yüzdesi	Byte	43	3
02	Miktar	Currency	46	15
02	Birim 0:Adet 1:Kilogram 2:Metre 3:Litre	Byte	61	1
02	Birim fiyat	Currency	62	15
02	Toplam fiyatı	Currency	77	15
02	Toplam KDV	Currency	92	15
				
03	Ödeme hareketinin, belge içindeki sıra numarası (1'den başlar)	Integer	4	6
03	Ödeme tipi 0 Ödeme 1 Para üstü 2 Döviz ile ödendi	Byte	12	1
04	Peşinat	Currency	4	15
				
04	Taksit Sayısı	Byte	33	3
04	Taksit Şablon Kodu	Integer	36	6
05	Kayıt tipi 0:İndirim satırı 1:Bilgi satırı	Integer	4	1
05	Promosyon kodu	Long	5	6
05	İndirim tutarı	Currency	11	15

Şekil IV.2 Satış verileri (fişli satış verileri) ile ilgili olarak oluşturulan GNDSALES.GTF isimli text dosyanın desen yapısı [27].

IV.2. ARAŞTIRMA ARAÇLARI

V.2.1. Donanım

- Bilgisayar, (Notebook, Intel 1.6 GHz. Centrino, 1GB Ram, 40GB HDD), - uygulama geliştirme kapsamında kullanıldı.
- Web Server, (Intel XEON İşlemci, 3GB Ram, 36GB HDD) – uygulamanın web ortamında test edilmesi için kullanıldı.
- Yazar kasa (IBM 4694), ilgili cihaz sadece araştırıldı. Veri yapısı incelendi.

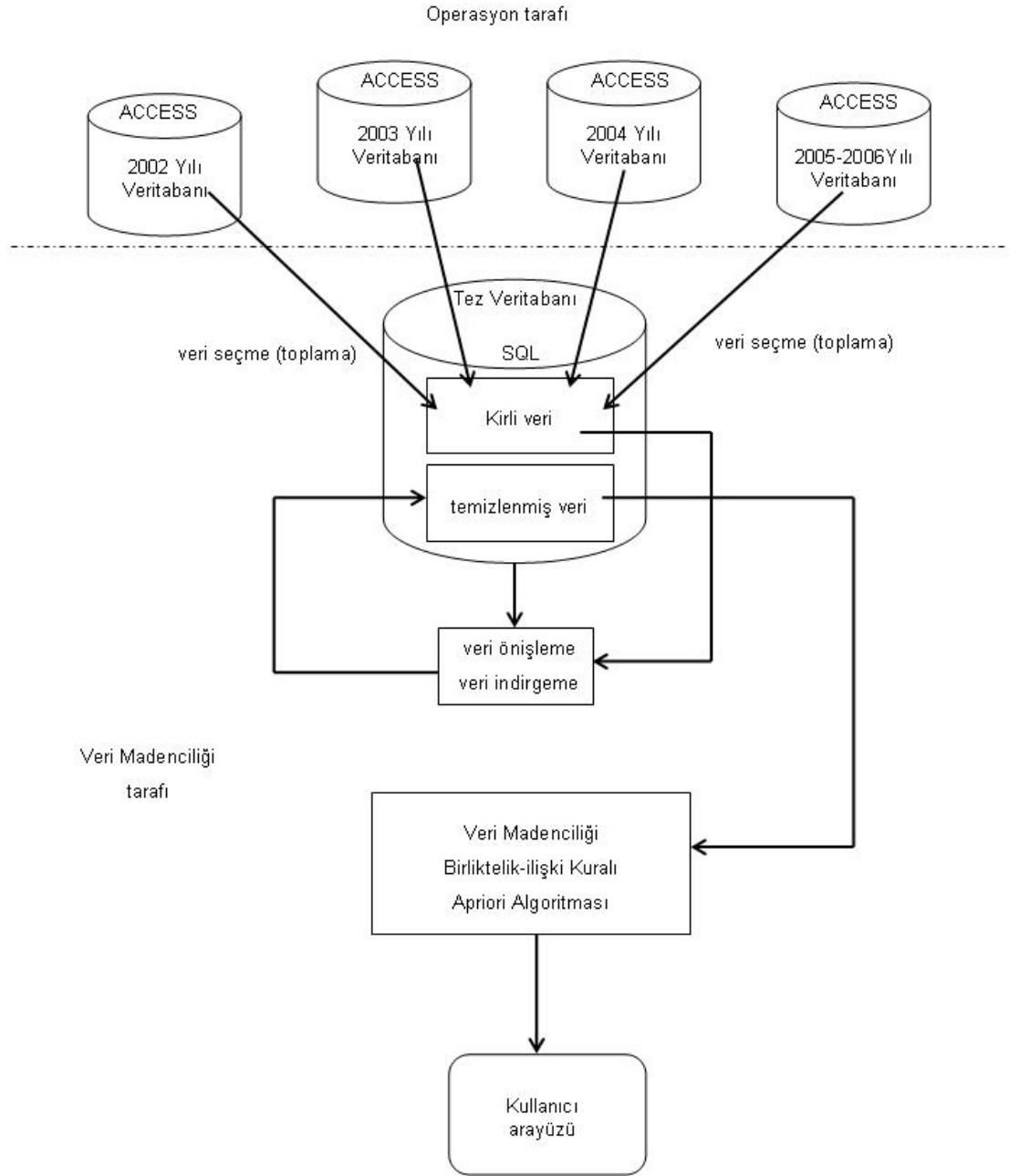
IV.2.2. Yazılım

- Yazar kasa ön ofis programı - Genius, ilgili yazım sadece incelendi.
- Satış noktasında yer alan kişisel bilgisayar ve merkezdeki server üzerinde yer alan Microsoft Access veritabanı programı, ilgili veritabanından uygulama veritabanına veri transferi yapıldı.
- Merkez deki server üzerinde yer alan Microsoft SQL 2000 veritabanı programı, uygulama veritabanının bulunduğu server.
- Merkez server üzerinde yer alan Microsoft Visual Basic 6.0 da kurum içi (in house) geliştirilen veri transfer programı.
- Apriori algoritması ile veri madenciliği uygulanabilmesi için oluşturulmuş veri üzerinde çalıştırılan web tabanlı programın yazımında kullanılan; Microsoft IIS, Microsoft .NET platformu, Microsoft ASP.NET ve Microsoft VB.NET programlama dilleri.

IV.3. VERİ MADENCİLİĞİ İŞLEMLERİ

Bu tez çalışmasında Bölüm IV.1’de detaylı olarak açıklaması yapılan operasyon tarafındaki veri toplama ve depolama süreçleri ile elde edilen veri üzerinde veri madenciliği işlemi gerçekleştirilmesi için; (Şekil IV.3’de tez kapsamında izlenen veri madenciliği işlemleri yer almaktadır.)

- Çalışmaya uygun bir veri depolama ortamı belirlendi,
- Veri depolama ortamı içinde verilerin aktarılacağı tablo yapıları, kullanılacak veri madenciliği yöntemi dikkate alınarak oluşturuldu,



Şekil IV.3 Tez çalışmasındaki veri madenciliği aşamaları.

- Operasyon tarafında yer alan veriler, veritabanlarında bilgi keşfi aşamalarından veri seçme, veri önışleme ve veri indirgeme aşamalarındaki yöntemler uygulanarak uygulama veri depolama yapısına alındı,

- Üzerinde veri madenciliği yapılacak kaliteye ulaştırılan veriler üzerinde Birliktelik-ilişki kuralı algoritmalarından Apriori algoritması için web ortamında çalışma kabiliyetine sahip bir uygulama yazılımı geliştirildi.
- Geliştirilen uygulama yazılımının çalıştırılması ile üzerinde Apriori algoritması uygulanan verilerden farklı zaman dilimi ve farklı lokasyonlarda birlikte satın alınma ilişkisine sahip ürün ve ürün gruplarını bulunabilmesi sağlandı.

IV.3.1. Çalışma Veri Depolama Yapısının Belirlenmesi,

Veri madenciliği uygulamalarında verinin azlığı genelde hedeflenen sonucun gerçekçilikten uzak olmasına neden olmaktadır. Verinin çokluğunda ise veri madenciliği uygulaması ile hedeflenen sonuca ulaşılması çok zaman alır, bu performans sorununun çözülmesi için daha üstün özellikli donanım ve yazılım kaynaklarının kullanılmasına ihtiyaç duyulur. Veri madenciliği uygulamasında kullanılacak verinin depolandığı ortam da performansı etkileyen faktörlerden biridir.

Bu çalışmada kullanılan veri 2002 yılının 9.ayından 2006 yılının 2.ayına kadarlık bir zaman dilimindeki gerçek operasyondan gelmektedir. Seçilecek veri depolama ortamının bu büyüklükteki veriyi depolayacak ve uygulanacak sorgulara hızlı cevap verecek nitelikte olması gerekmektedir. Bu nedenle Microsoft SQL 2000 veritabanı programı tez kapsamında kullanıldı.

IV.3.2. Veri Depolama Ortamında Tablo Yapılarının ve Tablolar Arasındaki İlişkilerin Oluşturulması

Operasyon tarafındaki veriler Bölüm IV.1’de açıklandığı gibi her yıla ait veriler ayrı birer Microsoft Access 97 veritabanı içinde tablolar halinde tutulmaktadır. Uygulama yazılımındaki sorguların anlamlı sonuç üretebilmeleri için her yıla ait verilerin tablolar halinde tek bir veritabanı içinde birleştirilmesinin yararlı olacağına karar verildi. Bu doğrultu da tez çalışması için kullanılan veritabanında operasyonel veritabanlarındaki tablo isim ve yapılarına benzer tablo yapıları oluşturuldu.

Özellikle hareket ve stok kartı tabloları oluşturulurken her tabloda kesinlikle bir veri alanı birincil anahtar ve tekil olacak şekilde tanımlanmıştır.

Operasyon tarafında her yıla ait veritabanlarında yer alan yazar kasa satış hareketlerinin başlık ve detay bilgilerini içlerinde tutan Sales_Tran_Headers (kasa satışı belge başlıkları) ve Sales_Tran_Lines (kasa satışı belge detayları) tablolarının yapısına benzer iki adet tablo oluşturuldu. Bu tabloların veritabanındaki konumu merkez olgu (fact) tablosu niteliğindedir, bu tablolar;

- kasa satış hareketi belgeleri (fiş, fatura) başlık bilgilerini içinde tutan SALES_TRAN_HEADERS ve
- kasa satış hareketi belgeleri (fiş, fatura) detay bilgilerini içinde tutan SALES_TRAN_LINES tablolarıdır.

SALES_TRAN_HEADERS tablosunda satış belgelerinin başlık bilgileri aşağıda belirtilen veri alanlarında (data fields) tutulmaktadır. Şekil IV.4'de SALES_TRAN_HEADERS tablosunun veri yapısı yer almaktadır.

- WYEAR – çalışma yılı,
- TRH_ID – her yıla ait kasa satış belgeleri başlık bilgisine ait ilgili yıl için tekil (unique) nitelikteki tanımlama veri alanı (ID),
- TRH_OzKd – Kasa satış belgeleri başlık bilgilerinin hangi lokasyondaki, hangi tarihdeki, hangi saatteki ve hangi tip satışa ait olduğu bilgilerinin tamamını içinde tutan, tekil (unique) nitelikteki özel bir veri alanıdır,
- TRH_Tarih – satışın gerçekleştiği tarih,
- TRH_GfsNo – satış fişinin numarası,
- TRH_Saat – satışın yazarkasa da gerçekleştiği saat,
- TRH_MgzNo – satışın gerçekleştiği lokasyonun kodu,
- TRH_Ksyr – satışı gerçekleştiren kasiyer no,
- TRH_BlgTp – satış belgesinin tipi (fiş, fatura),
- TRH_Iptal – satış işleminin iptal olup olmadığı bilgisi,
- TRH_FtNo – satış da kullanılan belge fatura ise fatura numarası,

	Column Name	Data Type	Length	Allow Nulls
🔑	ID	int	4	
	WYEAR	int	4	✓
	TRH_ID	int	4	
▶	TRH_OzKd	nvarchar	22	✓
	TRH_Tarih	smalldatetime	4	✓
	TRH_GfsNo	smallint	2	✓
	TRH_Saat	nvarchar	4	✓
	TRH_MgzNo	smallint	2	✓
	TRH_Ksyr	nvarchar	8	✓
	TRH_BlgTp	tinyint	1	✓
	TRH_Iptal	tinyint	1	✓
	TRH_FtNo	nvarchar	10	✓
	TRH_Line	smallint	2	✓
	TRH_GnlTpl	float	8	✓
	TRH_BlgKDv	float	8	✓
	TRH_IndTpl	float	8	✓
	TRH_GIndTpl	float	8	✓
	CD	tinyint	1	✓

Şekil IV.4 SALES_TRAN_HEADERS tablosunun veri yapısı.

- TRH_Line – satışın kaç satır olduğu bilgisi (satılan kalem sayısı),
- TRH_GnlTpl – satış belgesinin toplam bedeli,
- TRH_BlgKDV – satış belgesi KDV toplam bedeli,
- TRH_IndTpl – satış belgesinin dibinde yapılan indirim tutarı,
- TRH_GindTpl – satış belgesi dibinde ve satırlarda yapılan indirimlerin toplam tutarı.

SALES_TRAN_LINES tablosundaki kasa satış belgelerinin detay (satır) bilgileri aşağıda belirtilen veri alanlarında (data fields) tutulmaktadır:

- WYEAR – çalışma yılı,
- TRL_ID – her yıla ait kasa satış belgeleri detay bilgisine ait ilgili yıl için tekil nitelikteki tanımlama veri alanı (ID),
- TRL_HeadID – İlgili kayıttan SALES_TRAN_HEADERS tablosundaki hangi kayıta bağlı olduğunu bildiren veri alanı (SALES_TRAN_HEADERS tablosundaki TRH_ID veri alanı),
- TRL_Tarih – satışın gerçekleştiği tarih,
- TRH_MgzNo – satışın gerçekleştiği lokasyonun kodu,
- TRL_GfsNo – satış fişinin numarası,
- TRL_BlgTp – satış belgesinin tipi (fiş, fatura),
- TRL_FtNo – satış da kullanılan belge fatura ise fatura numarası,
- TRL_Line – satışın satır numarası,
- TRL_Stk_Kd – satışı yapılan ürünün stok kodu,
- TRL_HTip – hareket tipi (normal satır, iptal),
- TRL_KDV_Ref – ürünün KDV oran tablosundaki KDV kodu,
- TRL_KDV_Orn – ürünün KDV oranı,
- TRL_Miktar – satışı yapılan ürünün miktar bilgisi,
- TRL_Brm – satılan ürünün birim kodu,
- TRL_Brm_Fiy – satışı yapılan ürünün birim fiyatı,
- TRL_LineTpl – ilgili satış satırının toplam tutar bilgisi,
- TRL_KDVTpl - ilgili satış satırının toplam KDV tutar bilgisi,
- TRL_IndTpl - ilgili satış satırında yapılan indirim tutarı,
- TRL_LSonTpl - ilgili satış satırının indirim sonrası kalan toplam tutar bilgisi.

Şekil IV.5'de SALES_TRAN_LINES tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
PK	ID	int	4	
	WYEAR	int	4	✓
	TRL_ID	int	4	
	TRL_HeadID	int	4	✓
	TRL_Tarih	smalldatetime	4	✓
	TRL_MgzNo	smallint	2	✓
	TRL_GFsNo	smallint	2	✓
	TRL_BlgTp	tinyint	1	✓
	TRL_FtNo	nvarchar	10	✓
	TRL_Line	smallint	2	✓
	TRL_Stk_Kd	nvarchar	24	✓
	TRL_HTip	tinyint	1	✓
	TRL_KDV_Ref	tinyint	1	✓
	TRL_KDV_Orn	tinyint	1	✓
	TRL_Miktar	float	8	✓
	TRL_Brm	tinyint	1	✓
	TRL_Brm_Fiy	float	8	✓
	TRL_LineTpl	float	8	✓
	TRL_KDVTpl	float	8	✓
	TRL_IndTpl	float	8	✓
	TRL_LSonTpl	float	8	✓
	CD	tinyint	1	✓

Şekil IV.5 SALES_TRAN_LINES tablosunun veri yapısı

Operasyon tarafında yer alan satış bilgileri her yıla ait ayrı veritabanlarında tutulmaktadır, bu verilerin tek bir veritabanı içinde yıl ayrımı yapılmaksızın tek tablolar halinde depolanması planlandığından, uygulama veritabanındaki satış tablolarına WYEAR isimli çalışma yılı bilgisini içerecek veri alanı eklenmiştir.

Ürün satış verilerinde en önemli tablolardan biride ürün bilgilerinin yer aldığı boyut tablosu niteliğindeki **DM_STOCK_CARDS** tablosudur. Oluşturulan bu tablonunda veri alanları,

- WID – ilgili tablonun tekil nitelikteki ID numarasını içeren veri alanı, - primary key
- WYEAR – çalışma yılı,
- STORE – stok kodunun hangi lokasyona ait olduğu bilgisi,
- ID – birden fazla stok kart tablosunun birleşiminden bu tablo oluşturulduğundan ilgili yıllara ait tablolardaki stok kart ID numaraları,
- CODE – stok kodu,
- EXPLAIN – stok adı,
- BARCODE_TYPE – barkod tipi,
- MEDIUM_EXPLAIN – fatura üzerinde çıkacak stok adı,
- RECEIPT_EXPLAIN – ilgili ürünün üretim programındaki karşılık kodu,
- SHELF_EXPLAIN – ilgili ürün için raf etiketi açıklaması,
- SCALE_EXPLAIN – ilgili ürünün terazi cihazındaki stok adı,

- UNIT1 – birim 1 (ürünle ilgili ilk tercih edilen birimleme),
- UNIT2 – birim 2 (ürünle ilgili ikinci tercih edilen birimleme),
- UNIT3 – birim 3 (ürünle ilgili üçüncü tercih edilen birimleme),
- UNIT2_COEFF – birim 2 nin birim 1 ile olan çarpan veya bölen ilişkisi,
- UNIT3_COEFF – birim 3 ün birim 1 ile olan çarpan veya bölen ilişkisi,
- MAIN_DEPT – ana departman kodu (Üretim, Üretim Dışı, Ambalaj, Hediye),
- DEPT – ürünün üretim departman kodu,
- SUB_DEPT – ürünün üretim alt departman kodu,
- TYPE – ürünün satış grubu tip kodu,
- SELL_PRICE1 – ürünün 1.satış fiyatı,
- SELL_PRICE2 – ürünün 2.satış fiyatı,
- SELL_PRICE3 – ürünün 3.satış fiyatı,
- BUY_PRICE1 – ürünün 1.alış fiyatı,
- BUY_PRICE2 – ürünün 2.alış fiyatı,
- BUY_PRICE3 – ürünün 3.alış fiyatı,
- VAT_INCLUDED_FLAG – KDV dahil – hariç bayrağı,
- SELL_VAT_GRP – satış KDV grubu,
- BUY_VAT_GRP – alış KDV grubu,
- ACTIVE – stok kodunun aktiflik durumu,
- SHELF_LIFE – raf ömrü,
- SCALE_MODE – terazi modu,
- NOTES – özel not alanı,
- AUDIT_WHO – kodu ilk oluşturan kullanıcı kodu,
- AUDIT_WHEN – kodun ilk oluşum zamanı.

Şekil IV.6'de DM_STOCK_CARDS tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
▶	WID	int	4	
	WYEAR	int	4	✓
	STORE	int	4	✓
	ID	int	4	
	CODE	nvarchar	20	
	EXPLAIN	nvarchar	30	
	EXPLAIN_CONT	nvarchar	30	✓
	BARCODE_TYPE	tinyint	1	
	MEDIUM_EXPLAIN	nvarchar	20	✓
	RECEIPT_EXPLAIN	nvarchar	13	✓
	SHELF_EXPLAIN	nvarchar	20	✓
	SCALE_EXPLAIN	nvarchar	16	✓
	UNIT1	nvarchar	8	✓
	UNIT2	nvarchar	8	✓
	UNIT3	nvarchar	8	✓
	UNIT2_COEFF	float	8	
	UNIT3_COEFF	float	8	
	UNIT2_MULTIPLIES	bit	1	
	UNIT3_MULTIPLIES	bit	1	
	MAIN_DEPT	nvarchar	8	✓
	DEPT	int	4	✓
	SUB_DEPT	int	4	✓
	TYPE	nvarchar	8	✓
	SELL_PRICE1	money	8	✓
	SELL_PRICE2	money	8	✓
	SELL_PRICE3	money	8	✓
	BUY_PRICE1	money	8	✓
	BUY_PRICE2	money	8	✓
	BUY_PRICE3	money	8	✓
	VAT_INCLUDED_FLAG	tinyint	1	✓
	SELL_VAT_GRP	tinyint	1	✓
	BUY_VAT_GRP	tinyint	1	✓
	ACTIVE	tinyint	1	✓
	SHELF_LIFE	smallint	2	✓
	SCALE_MODE	tinyint	1	✓
	NOTES	nvarchar	255	✓
	AUDIT_WHO	nvarchar	8	✓
	AUDIT_WHEN	smalldatetime	4	✓
	SENT	tinyint	1	✓
	TS	varbinary	8	✓

Şekil IV.6 DM_STOCK_CARDS tablosunun veri yapısı.

Diğer boyut tabloları ve yapılarını;

CODE_MAIN_DEPT – Ana bölüm kod, açıklama bilgilerini içeren boyut tablosudur.

(Üretim, Üretim Dışı, Ambalaj, Hediyeelik, vb.)

- CODE – tekil nitelikteki ana bölüm kodu, - primary key
- EXPLAIN – ana bölüm adı,
- SENT – aktiflik bilgisi.

Şekil IV.7’de CODE_MAIN_DEPT tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
►	CODE	nvarchar	8	
	EXPLAIN	nvarchar	20	
	SENT	tinyint	1	✓
	TS	varbinary	8	✓

Şekil IV.7 CODE_MAIN_DEPT tablosunun veri yapısı.

CODE_DEPT – Üretim bölümlerinin kod, açıklama bilgilerini içeren boyut tablosudur. (Şokola İmalat, Turta İmalat, vb.)

- ID – tekil nitelikteki ID numarası
- CODE – tekil nitelikteki bölüm kodu, - primary key
- EXPLAIN – bölüm adı,
- PARENT_ID - bağlı olduğu CODE_MAIN_DEPT kodu,
- SENT – aktiflik bilgisi.

Şekil IV.8’de CODE_DEPT tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
►	ID	int	4	
🔑	CODE	nvarchar	8	
	PARENT_ID	nvarchar	8	✓
	EXPLAIN	nvarchar	20	
	SENT	tinyint	1	
	TS	varbinary	8	

Şekil IV.8 CODE_DEPT tablosunun veri yapısı.

CODE_SUB_DEPT – Üretim alt departmanlarının kod, açıklama bilgilerini içeren boyut tablosudur. (Madlen, Pralin, Tek Pasta, Tart, Donmuş Pasta, vb.)

- ID – tekil nitelikteki ID numarası
- CODE – tekil nitelikteki alt bölüm kodu, - primary key
- EXPLAIN – alt bölüm adı,
- PARENT_ID - bağlı olduğu CODE_DEPT kodu,
- SENT – aktiflik bilgisi.

Şekil IV.9’de CODE_SUB_DEPT tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
►	ID	int	4	
🔑	CODE	nvarchar	8	
	PARENT_ID	int	4	✓
	EXPLAIN	nvarchar	20	
	SENT	tinyint	1	
	TS	varbinary	8	

Şekil IV.9 CODE_SUB_DEPT tablosunun veri yapısı.

CODE_STOCK_TYPE – Ürünlerin ürün satış grubu kod, açıklama bilgilerini içeren boyut tablosudur. (Şokola satış hesabı, Turta satış hesabı, vb.)

- CODE – tekil nitelikteki satış grup kodu, - primary key
- EXPLAIN – satış grup adı,
- SENT – aktiflik bilgisi.

Şekil IV.10’de CODE_STOCK_TYPE tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
▶	CODE	nvarchar	8	
	EXPLAIN	nvarchar	20	
	SENT	tinyint	1	
	TS	varbinary	8	

Şekil IV.10 CODE_STOCK_TYPE tablosunun veri yapısı.

CODE_UNIT – Ürünlerin birim bilgilerini içeren boyut tablosudur. (KG., ADET, vb.)

- CODE – birim kodu
- SHORT_EXPLAIN – kısa birim açıklaması,
- EXPLAIN –birim açıklaması,
- WEIGHT_GR – her birimin gram cinsinden karşılığı,
- SENT – aktiflik bilgisi.

Şekil IV.11’de CODE_UNIT tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
▶	CODE	nvarchar	8	
	SHORT_EXPLAIN	nvarchar	4	
	EXPLAIN	nvarchar	20	
	WEIGHT_GR	money	8	
	SENT	tinyint	1	
	TS	varbinary	8	

Şekil IV.11 CODE_UNIT tablosunun veri yapısı.

BRANCHES – Satış lokasyonu (şube) bilgilerini içeren boyut tablosudur.

- ID – şube tablosu için tekil nitelikteki ID numarası – primary key
- WYEAR – çalışma yılı,
- CODE – şube kodu,
- AREA_CODE – alan kodu,
- CITY_CODE – şehir kodu,
- EXPLANATION – şube adı,
- MODEM_PHONE –şubelere bağlanmak için kullanılan modem numaraları,
- DOWNLOAD_PATH – şubelerden gelen text dosyaların kopyalandığı konum,

- ACT_EXP – şubenin aktiflik durum bilgisi.

Şekil IV.12’de BRANCHES tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
PK	ID	int	4	
	WYEAR	int	4	✓
	CODE	smallint	2	
	AREA_CODE	tinyint	1	
	CITY_CODE	smallint	2	
	EXPLANATION	nvarchar	30	✓
	MODEM_PHONE	nvarchar	20	
	DOWNLOAD_PATH	nvarchar	50	
	ACT_EXP	varchar	20	✓
	TS	varbinary	8	

Şekil IV.12 BRANCHES tablosunun veri yapısı.

BTIP – Satış belgesinin kod ve açıklama bilgilerini içeren boyut tablosudur (fiş, fatura, vb.).

- B_ID – belge için tekil nitelikteki ID numarası ve belge kodu – primary key
- B_EXP – belge açıklaması.

Şekil IV.13’de BTIP tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
PK	B_ID	tinyint	1	
	B_Exp	nvarchar	50	✓

Şekil IV.13 BTIP tablosunun veri yapısı.

HARTIP – Satış belgesindeki satırın normal veya iptal satırı bilgisini içeren boyut tablosudur.

- HID – belge için tekil nitelikteki ID numarası ve hareket tip kodu – primary key
- HEXP – belge açıklaması.

Şekil IV.14’de HARTIP tablosunun veri yapısı yer almaktadır.

	Column Name	Data Type	Length	Allow Nulls
PK	HID	tinyint	1	
	HExp	nvarchar	50	✓

Şekil IV.14 HARTIP tablosunun veri yapısı.

Şekil IV.15’de bu tablo yapıları arasındaki ilişkiler gösterilmektedir. Bu tablolar ile oluşan veritabanının veri modeli, çok boyutludur. Veri modeli kar tanesi şeması biçimindedir, kar tanesi şema yapısında,

- en önemli verinin içinde tutulduğu büyük bir merkez tablosu (olgu – fact tablosu) yer alır,
- her biri tek boyut için kullanılan küçük yardımcı tablolar kümesi (boyut tabloları) bulunur,
- ve bazı boyut tablolarının ek tabloları yer alır [10].

Kullanılan veritabanındaki SALES_TRAN_HEADERS ve SALES_TRAN_LINES tabloları *olgu tablolarıdır*.

DM_STOCK_CARDS, BRANCHES, BTIP, HARTIP tabloları *boyut tablosudur*.

CODE_UNIT, CODE_MAIN_DEPT, CODE_DEPT, CODE_SUB_DEPT; CODE_STOCK_TYPE tabloları da *boyut ek tablosudur*.

IV.3.2. Veri Madenciliği Uygulamasının Çalışacağı Veritabanına Veri Alınması

Verinin operasyonel veritabanlarından alınıp veri madenciliği uygulamasının çalışacağı uygulama veritabanına transfer edilmesi işleminde, veri tabanlarında bilgi keşfi süreçlerinden veri seçme, veri önileme ve veri indirgeme aşamaları kullanıldı. Bu aşamalar ile veri madenciliği yapılmasına uygun bir veri bütünlüğü oluşturuldu.

IV.3.2.1. Veri Seçme Süreci

Veri seçimi (veri toplama) işlemi, verilerin veri madenciliği uygulanması için mümkün olduğu kadar tek bir veritabanı veya veri ambarı içinde toplanmasıdır.

Operasyon tarafında yıllar bazında tutulan veriler veri madenciliği için hazırlanan veritabanına ODBC bağlantı yapısı kurularak (Access–SQL arasında), ekleme sorguları (append query) ile transfer edildi .

Operasyon tarafında yıllar bazında ayrı veritabanlarında tutulan satış tablolarının içeriği veri madenciliği için oluşturulan veri tabanındaki satış tablolarının (Bölüm IV.3.1’de detaylı olarak anlatılan SALES_TRAN_HEADERS ve SALES_TRAN_LINES) içine transfer işlemi esnasında verisi transfer edilen yıl bilgisi, ilgili tabloların WYEAR isimli veri alanlarına aktarıldı.

Operasyon tarafında yer alan ürün bilgilerinin tutulduğu stok kart tablolarının içindeki yer alan veri uygulama veritabanına transfer edilirken yıl bilgisi uygulama veritabanındaki stok kart (DM_STOCK_CARDS) tablosunun WYEAR veri alanına aktarıldı. Operasyon tarafında yer alan stok kart yapısı 2002, 2003 ve 2004 yıllarında merkezi olarak yönetilmiş ve tüm lokasyonlarda aynı stok kart yapısı kullanılmıştır. Fakat 2005 ve 2006 yıllarında merkezi yapıdan vazgeçilerek stok kartı

açma işlemi lokasyonların inisiyatifine bırakılmış, bu nedenle 2002-2004 yılları arasında tekil yapıda olan stok kart tablosu, 2005-2006 yıllarında bu özelliğini kaybetmiştir. Tez kapsamında tekil bir yapı kullanabilmek için lokasyonlardan 2005 ve 2006 yıllarında kullanmış oldukları stok kart bilgileri alındı, tekilleştirme sürecinden geçirilerek kullanılan uygulama veritabanındaki stok kart tablosuna aktarıldı.

Stok kart tablosunun ek ilişki tabloları (CODE_UNIT, CODE_MAIN_DEPT, CODE_DEPT, CODE_SUB_DEPT; CODE_STOCK_TYPE) ile hareket tablolarının diğer boyut tabloları (BTIP, HARTIP) uygulama veritabanında yer alan benzer veri alanları sahip tablolara sorgular ile aktarıldı.

Her yıla ait satış lokasyonu (şube) bilgileri uygulama veritabanında benzer alanlar içeren tabloya (BRANCHES) sorgular ile transfer edildi. Verisi aktarılan yıl bilgiside bu tabloda yer alan WYEAR veri alanına dolduruldu.

IV.3.2.2. Veri Önileme Süreci

Veri önileme, verilerin veri madenciliği için hazır duruma getirilmesi için veri üzerinde veri tipi dönüşümü, gruplama, kayıp değerleri yönetme ve gürültülü verinin temizlemesi gibi işlemlerin uygulandığı aşamadır.

Gürültülü veri de veri önileme aşamasında veri kümesinden temizlenir. Gürültülü veri, veri kümesi içinde yer alan ama veri madenciliği uygulamasında kullanılmayacak bir anlam içermeyen verilerdir [6]. Tez için hazırlanacak uygulamada Birliktelik-ilişki kuralının kullanılacağı düşünülerek Bölüm IV.1’de yapısı incelenen operasyona ait veritabanlarının içindeki tablolarda yer alan bazı veri alanlarında bulunan verilerin gereksiz olduğu tespit edilmiş ve bu doğrultuda uygulama veritabanındaki tablolar birliktelik-ilişki kuralına hizmet edecek şekilde daha az veri alanından (kolon) oluşacak şekilde tasarlanmıştır. Bu uygulama ile veri önileme kapsamında gürültülü verinin temizleme işlemi daha tablolar oluşturulurken gerçekleştirilmiştir.

Bölüm IV.3.2.1’de veri seçme sürecinde anlatılan 2005-2006 yılların da operasyonun işleyişi esnasında merkezi tekil yapı yerine lokasyonların inisiyatifine bırakılan stok kart bilgilerinin, ilgili veri üzerinde veri madenciliği uygulanabilmesi için lokasyonlardan ayrı ayrı toplanması ve tekilleştirilerek ortak stok kart tablosuna (DM_STOCK_CARDS) aktarılması, veri önileme aşaması içinde uygulanan *gruplama* işlemine örnek teşkil etmektedir.

Access veritabanı içinde yer alan tabloların bazı veri alanlarında boş veriler yer almaktadır, bu boş veriler Access veritabanını kullanan kurum içi geliştirilmiş

yazılımda problem çıkarmazken, SQL veri tabanına bu veriler aktarılınca sorguların çalışmasında ve performansında problem yaratmıştır. Bu nedenle SQL veritabanına aktarılan tüm tablolarda boş değer (NULL) taraması yapılmış, ilgili veri alanının veri tipi niteliğine göre veri madenciliği uygulamasını engellemeyecek ve sorun çıkarmayacak değerler ile doldurulmuştur. Sayısal alanlar genelde “0” (sıfır), karakter veri tipindeki alanlar ise Space(0) değerleri ile doldurulmuştur. Bu gerçekleştirilen işlem, veri önışleme aşaması içinde uygulanan *kayıp değerlerin yönetimi* işlemine örnek teşkil etmektedir.

IV.3.3. Apriori Algoritması ve Veri Madenciliği Uygulama Yazılımı

Veri depolama ortamının belirlenmesi, belirlenen veri depolama ortamı içindeki tablo yapıları ve tablolar arasındaki ilişkilerin tanımlanması, veritabanlarında bilgi keşfi süreçlerinde veri seçme-toplama aşaması ile operasyon tarafındaki verilerin uygulama veritabanına aktarılması, veritabanı içinde yer alan bu verilerin toplama esnasında veya veritabanı içinde iken veri önışleme sürecinden geçirilmesi ile veri madenciliğinin uygulanabilmesi için gerekli tüm süreçler tamamlanmıştır. Bu aşamada üzerinde veri madenciliği yapılacak kaliteye ulaştırılan veriler üzerinde Birliktelik-ilişki kuralı algoritmalarından Apriori algoritması için web ortamında çalışma kabiliyetine sahip bir uygulama yazılımı geliştirilmiştir.

IV.3.3.1. Uygulama Yazılımının Geliştirme Ortamı

Uygulama yazılımı Microsoft .NET platformunda ASP.NET yapısı üzerinde Microsoft Visual Basic.NET programlama dili kullanılarak web ortamında çalışabilecek şekilde nesne tabanlı olarak geliştirilmiştir.

Kullanıcıların programı çalıştırabilmesi için bilgisayarlarının İnternet ortamında ve bilgisayarlarında herhangi bir internet browser (İnternet Explorer, Navigator, vb.) programı olması yeterlidir.

ASP.NET yapısının genel niteliklerinden dolayı uygulama üzerine kurulu olduğu web sunucusu tarafında çalışmaktadır. Web sunucusu tarafında çalıştırılan uygulamanın sonuçları kullanıcı bilgisayarına bir HTML sayfası olarak gönderilmektedir, bu nedenle kullanıcı bilgisayarının performansı etkilenmemekte, herhangi ek yazılım veya donanıma ihtiyaç duyulmamaktadır.

Uygulama verileri SQL 2000 veritabanından okumakta ve algoritmanın işleyiş adımları doğrultusunda yine SQL 2000 veritabanına geçici tablolar oluşturmaktadır.

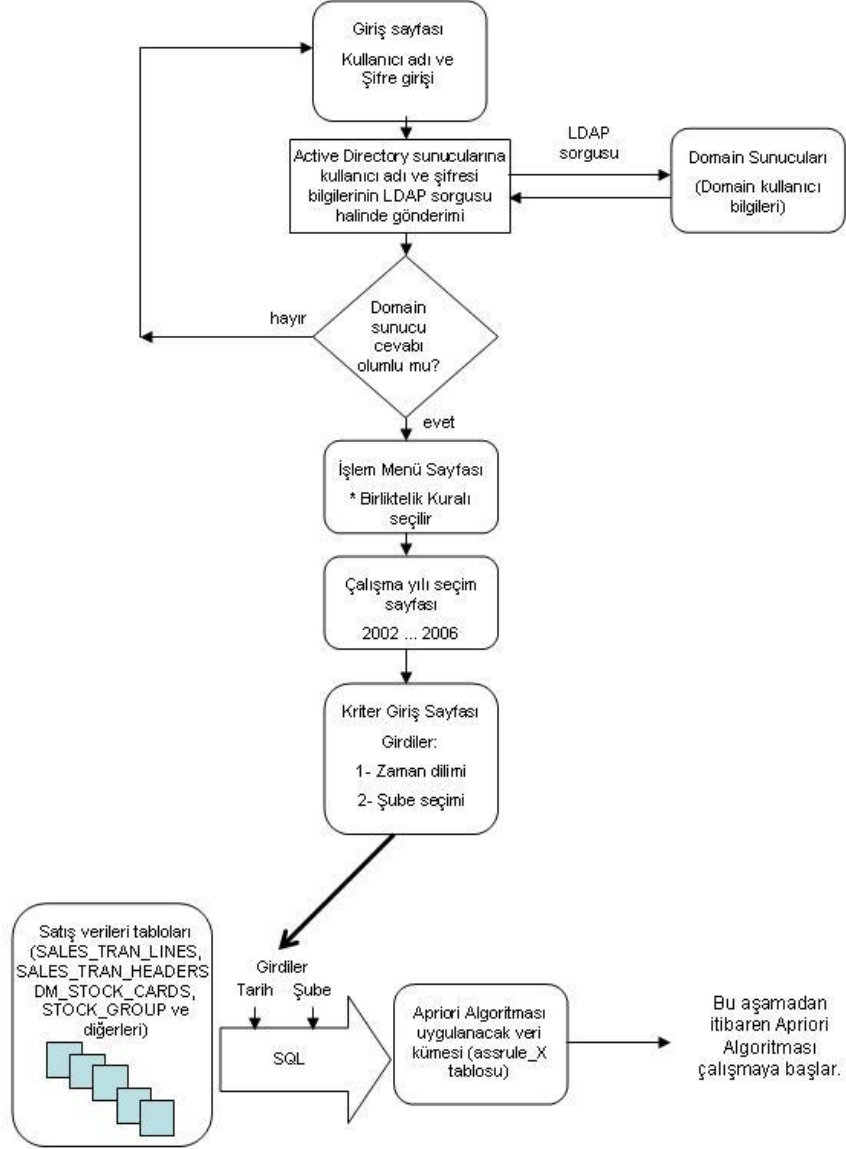
Program yazılırken çoklu kullanıcı ortamına uygun programlama teknikleri kullanılmıştır. Örneğin birden fazla kullanıcı aynı anda aynı veritabanı tablosu üzerinde algoritmayı çalıştırırken kullanıcılar için oluşturulan geçici tablolar kesinlikle çakışmamaktadır. Çünkü oluşturulan geçici tablolar isimleri kullanıcı adı ile ilişkilendirilmektedir. Eğer ki ortak nitelikteki bir kullanıcı adı (testuser, genel, vb.) ile programa girilirse, bu durumda geçici tabloların isimleri kullanıcı adı ile değil kullanıcının IP numarası ile ilişkilendirilmektedir.

IV.3.3.2. Uygulama Yazılımının Aşamaları

Uygulama yazılımı işleyişi şu şekildedir;

- İlk olarak kullanıcı adı ve şifresinin sorulduğu giriş ekranı gelir. Uygulama Microsoft Active Directory yapısı içinde çalışacak ise bu ekrana girilen kullanıcı adı ve şifresi active directory domain sunucularına LDAP sorguları ile gönderilir, sorgu sonucu olumlu gelirse kullanıcı programa giriş hakkı kazanır. Eğer ki uygulama active directory yapısına sahip olmayan bir ortamda kullanılacak ise bu giriş ekranı kullanıcı adı ve şifrelerinin tutulduğu bir veritabanı tablosu içine sorgu gönderir, sorgunun sonucuna göre kullanıcıya giriş hakkı verilir.
- Giriş hakkı alan kullanıcı işlem menü sayfasına ulaşır ve bu sayfadan Birliktelik-ilişki kuralı seçeneğini seçerek, çalışma yapılacak yıl bilgisinin belirlendiği ekrana ulaşır.
- Uygulama 2002 - 2006 yılları arasında çalışmaktadır, her yılın stok kart yapısı ve satış lokasyonu bilgileri farklı olduğundan tarih bazlı bu tip gruplama yapılmıştır. Çalışma yapılacak yıl bilgisi seçildikten sonra algoritmanın çalıştırılacağı kriter seçim sayfasına ulaşır.
- Kriter sayfasında algoritmanın hangi lokasyonlar için hangi zaman diliminde, hangi tip satışlar üzerinde çalışacağı belirlenir ve algoritmanın çalışması için "Çalıştır" tuşuna basılır.
- Belirtilen kriterler doğrultusunda ilgili satış kayıtları tarih, satış lokasyonu, fiş numarası, ürün kodu, birimi, satış miktarı ve dahil olduğu satış grup bilgisi sıralanışı ile data grid içinde ekranda belirir. Programın arka planında ekrana gelen bu veriler kullanıcının adı ile ilişkilendirilmiş bir geçici tabloya (assrule_<kullanıcı adı>) aktarılır. Kullanıcı aynı aşamayı bir defa daha çalıştırana kadar veya tekrar programa girene kadar bu tip tüm geçici tablolar veritabanında saklı kalır. Bu aşamadan itibaren Apriori algoritması çalışmaya başlar. Şekil IV.16'de uygulama yazılımına giriş adımları gösterilmektedir.

Uygulama Yazılımına Giriş Adımları



Şekil IV.16 Uygulama yazılımına giriş adımları.

IV.3.3.2. Apriori Algoritması

Apriori Algoritmasındaki temel yaklaşım eğer k-öge kümesi minimum destek ölçütünü sağlıyorsa, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar.

Birliktelik kuralı madenciliği iki aşamalıdır; tüm sık geçen öğelerin bulunması ve sık geçen bu öğelerden güçlü birliktelik kurallarının üretilmesidir. Birliktelik kuralını ilk aşaması için Apriori, sık geçen öğeler madenciliğinde kullanılan en popüler ve klasik algoritmadır. Bu algoritmada özellikler ve veri boolean ilişki kuralları ile değerlendirilir [29].

Kullanılan pazar sepeti verisinde her harekette yer alan ürün kodları sayısaldır ve ürün kodları küçükten büyüğe doğru sıralıdır. Öge kümeleri eleman sayıları ile birlikte anılır ve k adet ürüne sahip bir öge kümesi, k-öge kümesi diye isimlendirilir. k-öge kümesi c ifadesi ile gösterilirse, öğeleri (ürünler) $c[1], c[2], c[3], \dots, c[k]$ şeklinde gösterilir ve $c[1] < c[2] < c[3] < \dots < c[k]$ olacak şekilde küçükten büyüğe doğru sıralıdır [28]. Her öge kümesine destek metriğini tutmak üzere bir sayaç değişkeni iliştilmiştir ve sayaç değişkeni öge kümesi ilk kez yaratıldığında sıfırlanır. Aday öge kümeleri C_k karakteri ile gösterilir [3].

Algoritmada kullanılan değişkenler Tablo IV.1'de özetlenmiştir.

Tablo IV.1 Apriori algoritmasında kullanılan değişkenler [28].

k-öge kümesi (k-itemset)	k adet öge içeren öge kümesi.
L_k	Geniş (sık geçen) k-öge kümeleri setleri (bu kümeler minimum destek şartını sağlar). Bu setlerin her bir elemanının iki alanı vardır: i) öge kümesi ve ii) destek sayacı.
C_k	Aday k-öge kümeler setleri (potansiyel olarak geniş öge kümeleridir). Bu setlerin her bir elemanının iki alanı vardır: i) öge kümesi ve ii) destek sayacı.

Apriori algoritmasının klasik özet kodu Şekil IV.17 görülmektedir. Bu şekilde yer alan *apriori-gen* işlevi, (k-1) adet öğeye sahip L_{k-1} öğeler kümesini kullanarak k adet öğeye sahip aday kümeleri üretir. Bu işlev şu biçimde çalışır. İlk önce, L_{k-1} ile L_{k-1} *birleştirme (join)* işlemine tabi tutulur. Birleştirme işleminde L_{k-1} öge kümesinin her satırında yer alan son öge haricinde diğer öğelerin çapraz olarak benzerliği

```

1)  $L_1 = \{ \text{sık geçen 1-öğeler kümesi} \};$ 
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 transactions-hareketler  $t \in D$  do begin
5)      $C_t = \text{subset}(C_k, t);$  // Adaylar  $t$  içindedir
6)     forall candidates – adaylar  $c \in C_t$  do
7)        $c.\text{count}++;$ 
8)   end
9)    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$ 
10) end
11)  $\text{Answer} = \bigcup_k L_k;$ 

```

Şekil IV.17 Klasik Apriori Algoritması özet kodu [8, 28].

aranır ve son öğe haricinde diğer öğelerle yakalan benzerliklerden yeni aday öğe kümeleri oluşturulur. Oluşan kümeler *budama (prune)* adımı ile budanarak işlevden dönlür. Budama işlemi şu şekilde yapılır; c aday kümesinin $(k-1)$ öğeye sahip alt kümelerinden L_{k-1} 'de yer almayan kümeler silinir. *Apriori-gen* işlevinin algoritma kesiti, Şekil IV.18'da verilmiştir [3, 28].

```

insert into  $C_k$ 
select  $p.\text{item}_1, p.\text{item}_2, \dots, p.\text{item}_{k-1}, q.\text{item}_{k-1}$ 
from  $L_{k-1} p, L_{k-1} q$ 
where  $p.\text{item}_1 = q.\text{item}_1, \dots, p.\text{item}_{k-2} = q.\text{item}_{k-2},$ 
       $p.\text{item}_{k-1} < q.\text{item}_{k-1};$ 

forall itemsets  $c \in C_k$  do
  forall  $(k-1)$ -subsets  $s$  of  $c$  do
    if ( $s \notin L_{k-1}$ ) then
      delete  $c$  from  $C_k$ 

```

Şekil IV.18 Apriori-gen işlevinin algoritma kesiti [3, 28].

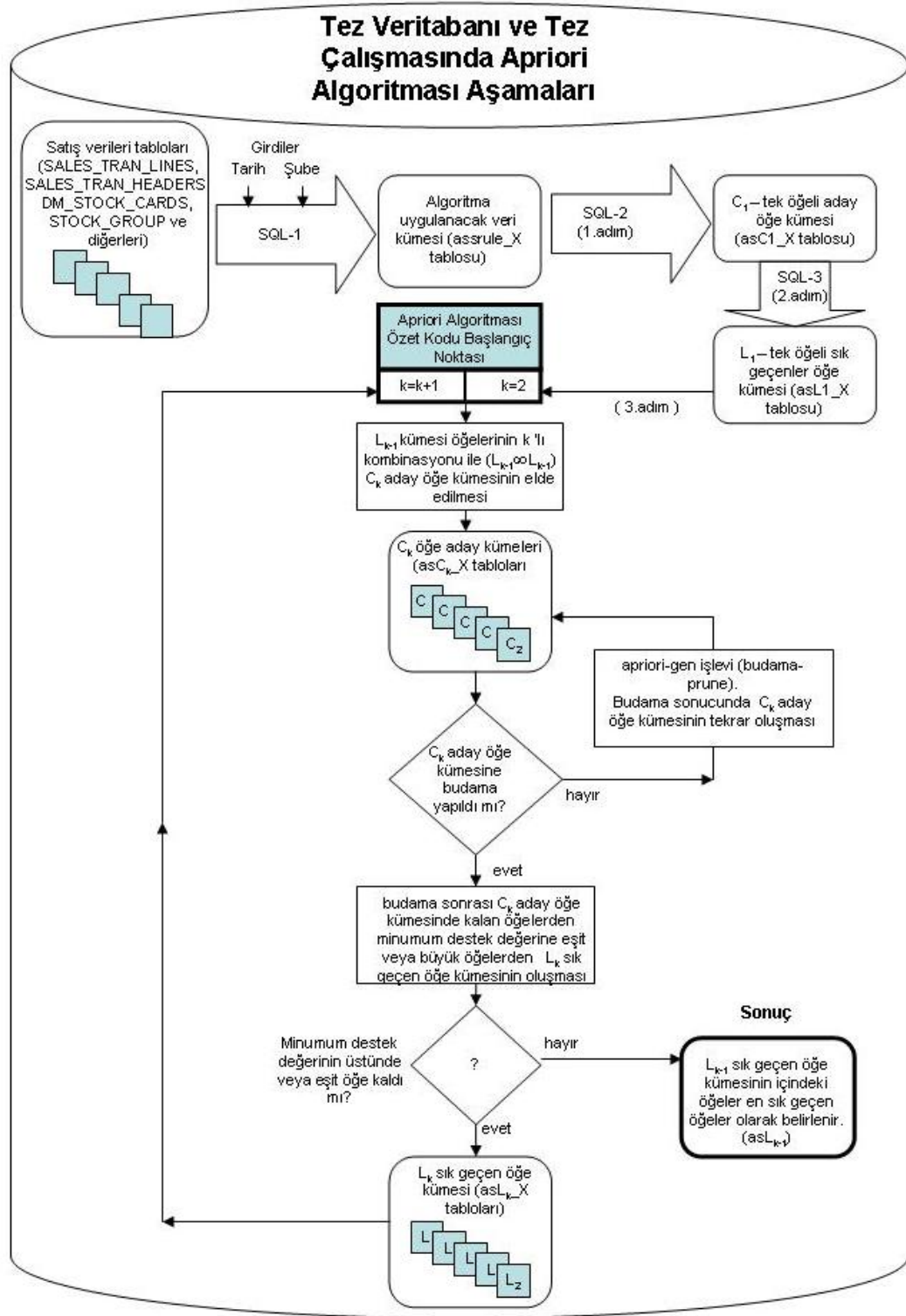
Budama aşamasında, tüm öğekümeleri $c \in C_k$ şeklindeki öğekümeler, bunların bazıları c kümesinin $(k-1)$ öğeye sahip içinde L_{k-1} barındırmayan tüm alt kümeleri silinir [28]. Farklı bir ifade ile budama, C_k aday öğe kümesindeki öğelerin alt kümelerinin L_{k-1} kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_{k-1} kümesinde yer almıyorsa ilgili öğe değerlendirme dışı kalır ve C_k aday öğe kümesinden silinir. [28]

Apriori algoritması özet kodu incelendiğinde sık geçen öge kümelerini bulmak için bir çok kez veritabanının tarandığı görülmektedir. İlk aşamadan önce, veri madenciliği uygulanacak veri topluluğunun taranarak ögelerin kaç adet hareket kayıtları içinde yer aldığı tespit edildiği (her öge için tespit edilen bu değere destek sayacı adı verilir) ve destek sayacı minimum destek değerine eşit veya büyük olan ögelerin L_1 sık geçen 1-öge kümesi olarak belirlendiği varsayılarak işleme başlanır.

Kod içinde kurulan döngü yapısı ile ilk aşamada L_1 sık geçen öge kümesinin ögelerinin ikili kombinasyonuna benzer bir şekilde ($L_1 \bowtie L_1$) yeni bir küme oluşur, bu işleme *birleştirme (join)* adı verilir, bu tarz oluşan kümelere de aday öge kümeleri adı verilir ve C harfi ile simgelenir. Oluşan bu aday öge kümesinin her elemanı iki adet ögeden oluştuğu için C_2 ifadesi ile isimlendirilir. Bu aday küme apriori-gen işlevi ile budama işlemine tabi tutulur ve C_2 kümesinin elemanlarına ait alt kümelerinin L_1 öge kümesinde olup olmadığına bakılır, alt kümelerden herhangi birisi L_1 içinde yer almayan küme elemanları C_2 aday kümesinden silinir. Apriori algoritması uygulanan veri topluluğu tekrar taranarak budama işleminden geçen C_2 aday kümesi elemanlarının kaç adet hareket kayıtları içinden geçtiği (destek sayacı) bulunur ve bulunan destek sayacı bilgileri doğrultusunda C_2 aday kümesi elemanlarının destek sayacı minimum destek değerine eşit veya büyük destek değerine sahip olan elemanları L_2 sık geçen öge kümesini oluşturur.

Döngü bir sonraki aşamada L_2 kümesi ögelerinin üçlü kombinasyonu ile oluşturulan yeni bir aday öge kümesi oluşturur ve bu küme C_3 ifadesi ile simgelenir. İlk aşamada olduğu gibi bu kümede budama işleminden geçer ve budama işleminden sonra minimum destek seviyesinin üstünde kalan elemanları ile L_3 sık geçen öge kümesi oluşturulur. Bu döngü her dönüşünde öge sayısını artırarak devam eder. Bu süreç yeni bir sık geçen öge kümesi bulunamayana kadar devam eder.

Bu çalışmada Apriori algoritması ile veri madenciliği işlemini gerçekleştirmek için tarafımızca geliştirilen programın çalışma aşamaları Şekil IV.19'da görülmektedir. Bu aşamalarda tez için hazırlanan veritabanından belli satış lokasyonları için belli zaman dilimindeki kayıtlardan seçilen veri bütünlüğünde, Agrawal ve Srikant'ın 1994 yılında tanımladığı Apriori algoritmasının özet kodu [28] dikkate alınmıştır.



Şekil IV.19 Tez çalışmasındaki yazılımda kullanılan Apriori algoritmasının aşamaları.

Bu aşamalarda gerçekleştirilen işlemlerin açıklaması aşağıda verilmektedir.

1) Belirtilen kriterler doğrultusunda ilgili satış kayıtları tarih, satış lokasyonu, fiş numarası, ürün kodu, birimi, satış miktarı ve dahil olduğu satış grup bilgisi sıralanışı ile oluşturularak geçici bir tabloya (assrule_<kullanıcı adı>) aktarılması için SQL sorgusu (Şekil IV.19'daki SQL-1) kullanılır.

```
INSERT INTO <assrule_kullanıcı adı> ( TARİH, MGZKOD, FISNO, SKOD, STSMIK,
KHSTSTUT)
SELECT TRL_Tarih, TRL_MgzNo, TRL_GFsNo, TRL_Stk_Kd,
SUM(TRL_Miktar) AS MIKTAR, SUM(TRL_LineTpl-TRL_KDVTpl) AS TUTAR
From SALES_TRAN_LINES
WHERE (((TRL_Tarih) Between <Dönem Başlangıç Tarihi> AND <Dönem Bitiş Tarihi>)
AND (<Seçilen Şubeler>)
AND ((WYEAR) =<Çalışma Yılı>))
GROUP BY TRL_Tarih, TRL_MgzNo, TRL_GFsNo, TRL_Stk_Kd
```

Sorgu sonucunda elde edilen tablo algoritmanın işleyişi esnasında algoritmanın uygulanacağı veri kümesini temsil etmektedir. Bu tablonun içeriği ekran üzerinde bir data grid içinde kullanıcıya gösterilir.

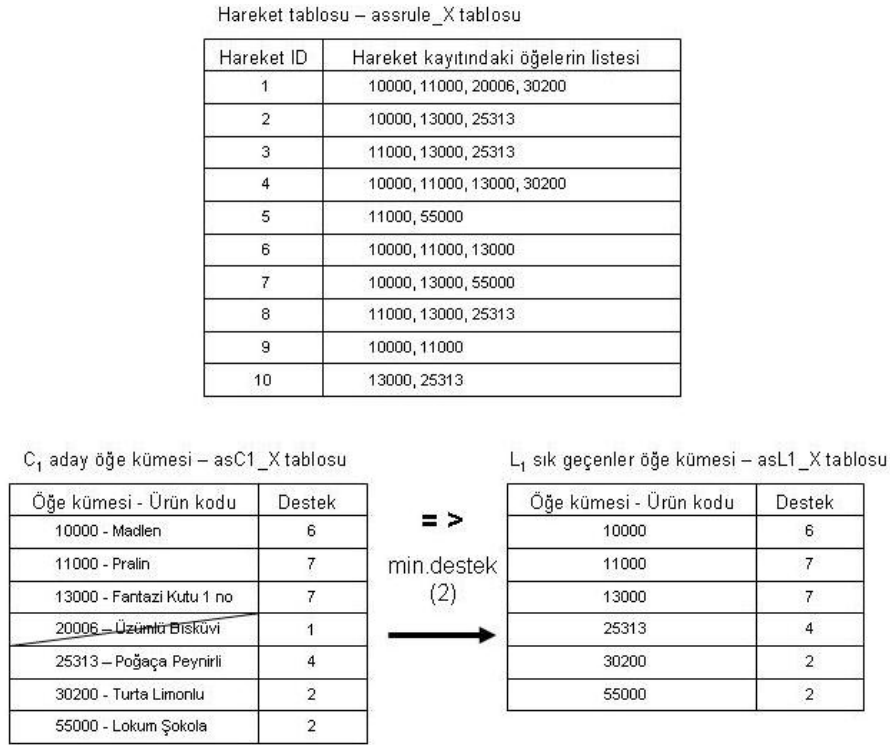
2) Tez uygulamasındaki ifade ile algoritmanın ilk adımında, algoritma basit olarak, tüm tabloyu tarar ve her nesnenin kaç işlemde geçtiğinin bilgisine erişilir, elde edilen bu veriler veritabanında geçici bir tablo (asC1_<kullanıcı adı>) içinde tutulur. Bu geçici tablodaki veriler apriori algoritmasının C_1 aday öge kümesini temsil eder. Bu veriler ekranda stok kodu, stok adı, işlem sayısı ve dahil olduğu satış grup bilgisi şeklinde görüntülenir. İlgili geçici tablonun oluşturulmasında SQL sorgusu (Şekil IV.19'daki SQL_2) kullanılır.

```
SELECT SKOD, STKAD, COUNT(SKOD) AS SUPPORT, STYPAD AS GRUP
INTO <asC1_kullanıcı adı>
From <assrule_kullanıcı adı>
GROUP BY SKOD, STKAD, STYPAD
ORDER BY SUPPORT DESC
```

3) İkinci adım da, minimum destek (support) değeri belirlenir. Bu min. destek değeri yardımı ile Apriori algoritmasının en sık geçen tek öğeleri içeren L_1 öge kümesi elde edilir, başka bir ifade ile; min destek (support) değerine eşit veya yüksek destek değerine sahip olan öğeler L_1 kümesini oluştururlar ve ekranda görüntülenirler. Elde edilen bu veriler veritabanında geçici bir tablo (asL1_<kullanıcı adı>) içinde tutulur, tablonun oluşturulmasında SQL sorgusu (Şekil IV.19'daki SQL-3) kullanılır.

```
SELECT *
INTO <asL1_kullanıcı adı>
FROM <asC1_kullanıcı adı>
WHERE SUPPORT < min.destek değeri
```

L_1 sık geçenler öge kümesinin oluşumu örnek veriler ile Şekil IV.20'de görülmektedir [10].



Şekil IV.20 Apriori algoritmasında C_1 - L_1 dönüşümü.

4) Uygulamanın üçüncü adımından itibaren Apriori algoritmasının özet kodunda tanımlanan döngü ve özel işlevler çalıştırılmaya başlar. 3.adım da, L_2 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L_1 kümesinin öğelerinin ikili kombinasyonundan ($L_1 \bowtie L_1$ - *birleştirme işlemi*) yeni bir aday öge kümesi olan C_2 oluşturulur. Bu işleme *birleştirme (join) işlemi* denir, birleştirme *apriori-gen* işlevinin işlemlerinden biridir. Oluşan bu aday küme verileri veritabanında geçici bir tablo (asC2_<kullanıcı adı>) içine aktarılır. Yazılım tarafında bu işlemin gerçekleşmesi için dizi, döngü yapıları ve SQL sorguları kullanılır.

C_2 aday öge kümesinin oluşturulması için kullanılan Visual Basic.NET kodu;

```
'C2 aday öge kümesi tablosunun yapısının oluşturulması,
Dim cmd As New SqlCommand
cmd.Connection = cnn
cmd.CommandType = CommandType.Text
cmd.CommandText = "Create table <asC2_kullanıcı adı>" _
    & "(" _
    & "SKOD_21 char(15), " _
    & "SKOD_22 char(15), " _
    & "L1XL1 char(30) " _
    & ")"
cmd.ExecuteNonQuery()

'L1 sık geçen öge kümesinin bir benzerinin JOIN işlemi için oluşturulması, mtL1K tablosu
cmd.CommandText = "SELECT * INTO mtL1K " _
```

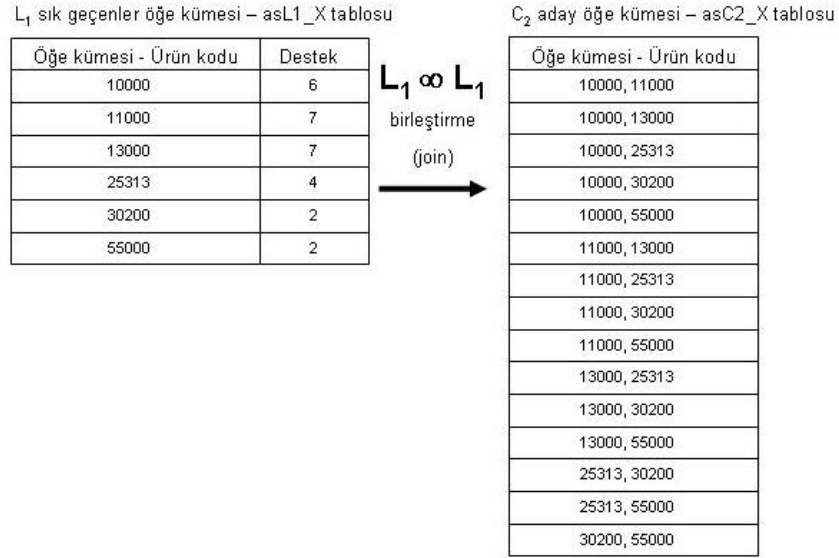
```

& "From <asL1_kullanıcı adı> "
cmd.ExecuteScalar()

'C2 aday öge kümesi kümesinin JOIN işlemi uygulanarak oluşturulması
cmd.CommandText = "insert into <asC2_kullanıcı adı> (SKOD_21, SKOD_22, " _
& "L1XL1) " _
& "SELECT <asL1_kullanıcı adı>.SKOD, mtL1K.SKOD, RTRIM(" _
<asL1_kullanıcı adı>.SKOD) " _
& "+ ' X ' + RTRIM(mtL1K.SKOD) AS L1XL1 " _
& "FROM <asL1_kullanıcı adı> INNER JOIN " _
& "mtL1K ON <asL1_kullanıcı adı>.SKOD < mtL1K.SKOD " _
& "ORDER BY <asL1_kullanıcı adı>.SKOD, mtL1K.SKOD"
cmd.ExecuteScalar()

```

C₂ aday öge kümesinin oluşumu örnek veriler ile Şekil IV.21'de görülmektedir.



Şekil IV.21 Örnek verilerin L₁ $\bowtie$ L₁ dönüşümü ile oluşan C₂ aday öge kümesi.

5) Apriori algoritmasının özet koduna göre C₂ aday öge kümesi oluştuktan sonra *apriori-gen* işlevinin işlemlerinden budamanın yapılması gerekir. *Budama* işlemi kısaca, C₂ aday öge kümesindeki ögelerin alt kümelerinin L₁ kümesindeki varlığı kontrol edilir, bir ögenin alt kümelerinden biri, L₁ kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C₂ aday öge kümesinden silinir. Fakat özet kod da C₂ için bu işlem yapılması gerekiyor olmasına rağmen C₂ için bu işlem gerçekleştirilmez, çünkü C₂ kümesinin tüm ögelerinin alt kümeleri tek öğeden oluşmakta ve L₁ kümesinden birleştirme işlemine uğrayarak C₂ aday öge kümesine gelmişlerdir.

Bu nedenden dolayı, uygulamanın dördüncü adımında, C₂ aday kümesine bakılır ve minumum destek değerine eşit veya yüksek olan değerlerden L₂ sık geçen öge kümesi elde edilir ve ekranda görüntülenir. L₂ kümesi elde edilirken

veritabanında birden fazla geçici tablo oluşturulur ve en son aşamada L₂ kümesi bir geçici tablo (asL2_<kullanıcı adı>) içinde saklanır.

L₂ sık geçen öge kümesinin oluşturulması için kullanılan Visual Basic.NET kodu;

```
'asC2_X tablosundaki kayıt sayısı hesabı
cmd.CommandText = "select count(*) from <asC2_kullanıcı adı>"
c = cmd.ExecuteScalar

Dim dizi(c, 2) As String
'asC2_X tablosundaki kayıtlar datareader içine alınıyor.
cmd.CommandText = "SELECT * FROM <asC2_kullanıcı adı>"
Dim dtr As SqlDataReader
dtr = cmd.ExecuteReader
Dim tk As Integer = 1

'2 boyutlu dizinin 1. ve 2.kolonu C2 tablosunun kayıtları datareader sayısı ile dolduruluyor
Do While dtr.Read = True
    dizi(tk, 1) = dtr(0)
    dizi(tk, 2) = dtr(1)
    tk = tk + 1
Loop
dtr.Close()

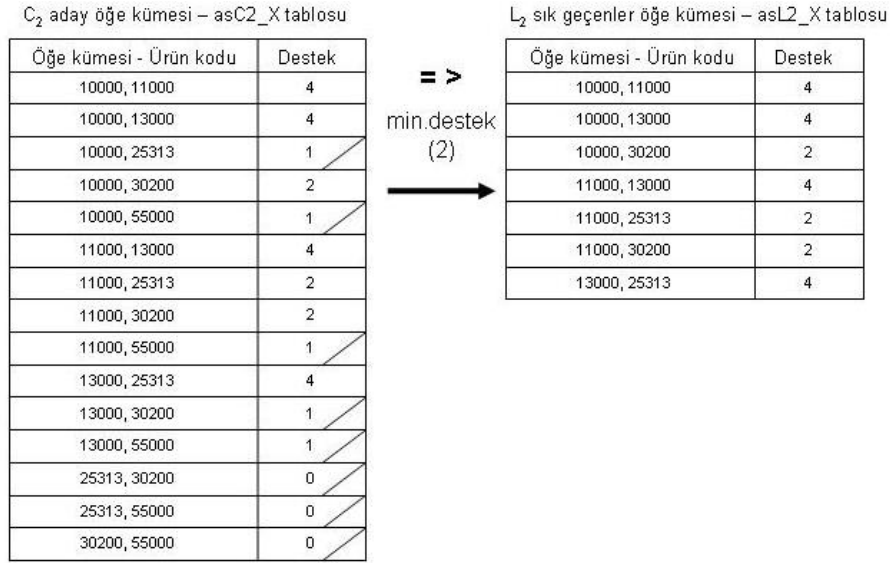
Dim m As Integer = 1
'2 boyutlu dizinin elemanlarına göre hareketlerin mtL2_1 ve mtL2_2 geçici tablolarına aktarılması
Dim p As Integer
For p = 1 To c
    'dizinin 1.kolon elemanlarına göre satış kayıtlarının mtL2_1 tablosuna aktarımı
    cmd.CommandText = "INSERT INTO mtL2_1 ( SKOD_L2, BELGE ) " _
    & "SELECT '" & RTrim(dizi(p, 1)) & " X " & RTrim(dizi(p, 2)) & "', " _
    & "CAST(TARİH AS char(8))+ '-' + MGZKOD + '-' + " _
    & "CAST(FISNO AS char(5)) as BELGE " _
    & "From <assrule_kullanıcı adı> " _
    & "WHERE SKOD = '" & dizi(p, 1) & "'"
    cmd.ExecuteNonQuery()

    'dizinin 2.kolon elemanlarına göre satış kayıtlarının mtL2_2 tablosuna aktarımı
    cmd.CommandText = "INSERT INTO mtL2_2 ( SKOD_L2, BELGE ) " _
    & "SELECT '" & RTrim(dizi(p, 1)) & " X " & RTrim(dizi(p, 2)) & "', " _
    & "CAST(TARİH AS char(8))+ '-' + MGZKOD + '-' + " _
    & "CAST(FISNO AS char(5)) as BELGE " _
    & "From <assrule_kullanıcı adı> " _
    & "WHERE SKOD = '" & dizi(p, 2) & "'"
    cmd.ExecuteNonQuery()
Next p

'mtL2_1 ile mtL2_2 tablolarının joinlenerek DESTEK seviyelerinin asL2 tablosuna aktarımı
cmd.CommandText = "INSERT INTO <asL2_kullanıcı adı> ( SKOD_L2, SUPPORT ) " _
& "SELECT mtL2_1.SKOD_L2, COUNT(mtL2_1.BELGE) AS SUPPORT " _
& "FROM mtL2_1 INNER JOIN " _
& "mtL2_2 ON mtL2_1.SKOD_L2 = mtL2_2.SKOD_L2 AND " _
& "mtL2_1.BELGE = mtL2_2.BELGE " _
& "GROUP BY mtL2_1.SKOD_L2 " _
& "ORDER BY SUPPORT DESC"
cmd.ExecuteNonQuery()

'Min.Destek seviyesinin altındaki değerler silinir
cmd.CommandText = "DELETE FROM <asL2_kullanıcı adı> " _
& "WHERE SUPPORT < min.destek değeri "
cmd.ExecuteNonQuery()
```

L₂ sık geçenler öge kümesinin oluşumu örnek veriler ile Şekil IV.22'de görülmektedir.



Şekil IV.22 Örnek veriler üzerinde C₂-L₂ dönüşümünün gerçekleşmesi.

6) Uygulamanın beşinci adımında, L₃ sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L₂ kümesinin öğelerine birleştirme işlemi uygulandığında (L₂ ∩ L₂ - birleştirme) yeni bir aday öge kümesi olan C₃ oluşturulur. C₃ kümesi elde edilirken veritabanında birden fazla geçici tablo oluşturulur ve en son aşamada C₃ kümesi geçici bir tablo (asC3_<kullanıcı adı>) içine aktarılır. Yazılım tarafında bu işlemin gerçekleşmesi için dizi, döngü yapıları ve SQL sorguları kullanılır.

C₃ aday öge kümesinin oluşturulması için kullanılan Visual Basic.NET kodu;

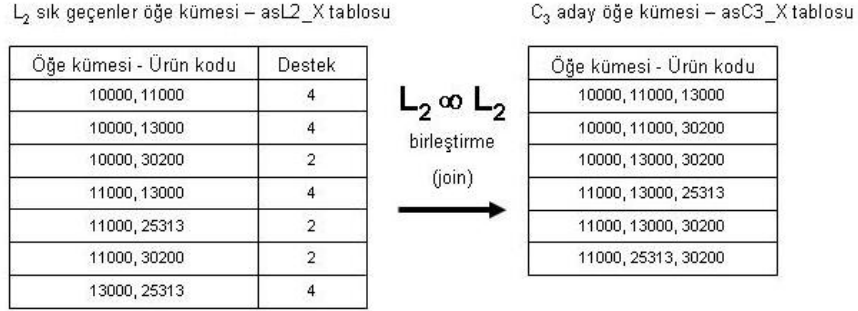
```
'C3 aday öge kümesinin tablo yapısını oluşturma,
Dim cmd As New SqlCommand
cmd.Connection = cnn
cmd.CommandType = CommandType.Text
cmd.CommandText = "Create table <asC3_kullanıcı adı> " _
& "(" _
& "SKOD_C31 char(20), " _
& "SKOD_C32 char(20), " _
& "SKOD_C33 char(20), " _
& "L2XL2 char(30) " _
& ")"
cmd.ExecuteNonQuery()

'L2 sık geçen öge kümesinin benzerinin JOIN işlemi için oluşturulması, mtL2K
cmd.CommandText = "SELECT * INTO mtL2K " _
& "From <asL2_kullanıcı adı> "
cmd.ExecuteNonQuery()

'C3 aday öge kümesinin JOIN işlemi ile oluşturulması,
cmd.CommandText = "insert into <asC3_kullanıcı adı> (SKOD_C31, SKOD_C32, SKOD_C33, " _
& "L2XL2) " _
& "SELECT <asL2_kullanıcı adı>.SKOD_21, <asL2_kullanıcı adı>.SKOD_22, " _
& "mtL2K.SKOD_22, RTRIM(<asL2_kullanıcı adı>.SKOD_21) " _
& "+ ' X ' + RTRIM(<asL2_kullanıcı adı>.SKOD_22) + ' X ' + RTRIM(" _
& "mtL2K.SKOD_22)AS L2XL2 " _
& "FROM <asL2_kullanıcı adı> INNER JOIN " _
& "mtL2K ON <asL2_kullanıcı adı>.SKOD_21 = mtL2K.SKOD_21 AND " _
& "<asL2_kullanıcı adı>.SKOD_22 < mtL2K.SKOD_22 " _
& "ORDER BY <asL2_kullanıcı adı>.SKOD_21, <asL2_kullanıcı adı>.SKOD_22, " _
```

```
& "mtL2K.SKOD_22"
cmd.ExecuteScalar()
```

C_3 aday öge kümesinin oluşumu örnek veriler ile Şekil IV.23'de görülmektedir.



Şekil IV.23 Örnek verilerin $L_2 \bowtie L_2$ dönüşümü ile oluşan C_3 aday öge kümesi.

7) Altıncı adım da, C_3 aday kümesi üzerinde *apriori-gen* işlemlerinden *budama* (prune) işlemi uygulanır. C_3 aday öge kümesindeki ögelerin alt kümelerinin L_2 kümesindeki varlığı kontrol edilir, bir ögenin alt kümelerinden biri, L_2 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_3 aday öge kümesinden silinir. Budama işlemi sonucunda elenenler ve C_3 aday öge kümesinin geçici tablosundan (asC3_<kullanıcı adı>) silinen ögeler, veritabanında farklı bir geçici tabloya aktarılır.

Bu aşamada aday kümedeki tüm ögeler budama sonucunda değerlendirme dışı kalabilir ve algoritma sonlanabilir. C_k aday öge kümesinde yapılan budama işleminde algoritma sonlandığında L_{k-1} sık geçen ögeler kümesindeki ögeler en çok birlikte satın alınan ürünler olarak kabul edilir veya tekrar dan algoritmayı farklı bir minimum destek değeri belirleyerek çalıştırmak için 1.adıma geri dönüş yapılması gerekmektedir.

C_3 aday öge kümesi üzerinde budama (prune) işleminin yapılması için kullanılan Visual Basic.NET kodu;

```
c = 3 ' C3 ün bir kayıttaki eleman sayısı

'Kombinasyon hesabı (n,2) = n!/(2!X(n-2)!) - başı
Dim kombi As New Integer, nkom1, nkom2

'nkombi=n! hesabı
nkom1 = 1
For i = 1 To c
    nkom1 = nkom1 * i
Next i

'nkombi=(n-2)! hesabı
nkom2 = 1
For i = 1 To c - 2
    nkom2 = nkom2 * i
Next i

kombi = nkom1 / (2 * 1 * nkom2)
'Kombinasyon hesabı - sonu

Dim indis(kombi, 2) As String
Dim malzeme(c) As String
```

```

Dim mm As Integer, cntind, icin, anas
Dim srctxt As String

cmd.CommandText = "SELECT count(*) FROM <asC3_kullanıcı adı>"
countd = cmd.ExecuteScalar
'C3 yapısında bir dizi tanımlanır
Dim soudata(countd, c) As String

cmd.CommandText = "SELECT * FROM <asC3_kullanıcı adı>"
Dim dtr As SqlDataReader
dtr = cmd.ExecuteReader
cntind = 1
'C3 kümesi 3 boyutlu dizi içine doldurulur
Do While dtr.Read = True
    For icin = 1 To c
        soudata(cntind, icin) = dtr(icin - 1)
    Next icin
    cntind = cntind + 1
Loop
dtr.Close()

anas = 1
'C3 kümesi nin her satırı için yapılan işlemler
For anas = 1 To countd

    srctxt = ""
    'C3 kümesi nin her satırındaki değerler tek boyutlu bir dizinin içine aktarılıyor ayrıca her
satırdaki öğeler arama text i şekline dönüştürülüyor
    For mm = 1 To c Step 1
        malzeme(mm) = soudata(anas, mm)
        If mm = 1 Then
            srctxt = RTrim(soudata(anas, mm))
        Else
            srctxt = srctxt & " X " & RTrim(soudata(anas, mm))
        End If
    Next mm

    Dim tk As New Integer, t, k, z, yy, m, p, s, dongu
    t = 1
    tk = 1
    yy = 0
    m = 1
'2 boyutlu dizinin 1 ve 2.kolunu içi tek boyutlu indisten gelen üç öge nin (C3 ün her satırında yer alan
öğeler) ikili kombinasyonları ile dolduruluyor
    For dongu = 1 To c Step 1
        If c - t <> 0 Then
            '2 boyutlu dizinin 1.kolunu dolduruluyor
            For k = 1 To c - t
                indis(tk, 1) = malzeme(dongu)
                tk = tk + 1
            Next k

            End If
            t = t + 1
        Next dongu
        m = 1
        '2 boyutlu dizinin 2.kolunu içi dolduruluyor
        For p = 1 To kombi
            For m = 1 To c
                If malzeme(m) = indis(p, 1) Then GoTo mFor
            Next m
        mFor:
            For s = m + 1 To c
                indis(p, 2) = malzeme(s)
                p = p + 1
            Next s
            p = p - 1
        Next p
'2 boyutlu dizinin her satırındaki değerlerin (bunlar C3'ün 2 alt kümeleridir) L2 tablosundaki varlıkları
döngü ile kontrol edilir.
        Dim dbb As New Integer, src
        For dbb = 1 To kombi
            cmd.CommandText = "select count(*) from <asL2_kullanıcı adı> " _
                & "WHERE SKOD_L2 = '" & RTrim(indis(dbb, 1)) & " X " _
                & RTrim(indis(dbb, 2)) & "'"
            src = cmd.ExecuteScalar
        Next dbb
    Next anas

```

'Eğer ki C_3 ün her bir satırındaki üç öğenin ikili alt kümelerinden en az biri L_2 de bulunamazsa C_3 deki ilgili öğe bir geçici tabloya aktarılıp C_3 den silinir

```

If src = 0 Then
    cmd.CommandText = "insert into mtC3_4 (L2XL2) values " _
        &"('" & srctxt & "'"")"
    cmd.ExecuteNonQuery()

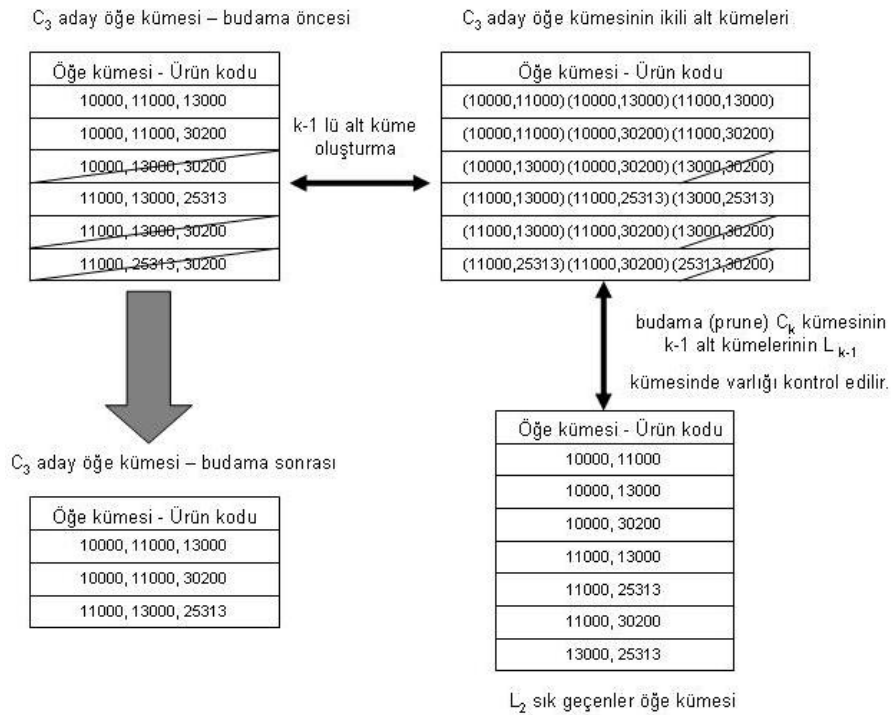
    cmd.CommandText = "delete from <asC3_kullanıcı adı> " _
        &"WHERE L2XL2 = '" & srctxt & "'"
    cmd.ExecuteNonQuery()

Exit For
End If
Next dbb

Next anas

```

C_3 aday öğe kümesi üzerinde budama işlemi yapılması örnek veriler ile Şekil IV.24'de görülmektedir.



Şekil IV.24 Apriori algoritmasında C_3 kümesinde budama işlemi.

8) Yedinci adım da, C_3 aday öğe kümesine bakılır ve minumum destek değerine eşit veya yüksek olan değerlerden L_3 sık geçen öğe kümesi elde edilir ve ekranda görüntülenir. L_3 kümesi elde edilirken veritabanında birden fazla geçici tablo oluşturulur ve en son aşamada L_3 kümesi bir geçici tablo (asL3_<kullanıcı adı>) içinde saklanır. C_k aday kümelerinde L_k sık geçen öğe kümelerini elde edilmesi esnasında C_k aday kümesindeki hiç bir öğe minumum destek değerini geçemeyebilir, bu durumda algoritma sonlanmıştır ve belirlenen minumum destek değerine L_{k-1} kümesinin öğeleri içinde en sık birlikte satın alınan ürünler olarak belirlenir.

L₃ sık geçen öge kümesinin oluşturulması için kullanılan Visual Basic.NET kodu;

```
'asC3_X tablosundaki kayıt sayısı hesabı
cmd.CommandText = "select count(*) from <asC3_kullanıcı adı>"
c = cmd.ExecuteScalar

Dim indis(c, 3) As String

cmd.CommandText = "SELECT * FROM <asC3_kullanıcı adı>"
Dim dtr As SqlDataReader
dtr = cmd.ExecuteReader

Dim tk As Integer = 1

'3 boyutlu dizinin 1. 2. ve 3. kolonu C3 tablosu ile dolduruluyor
Do While dtr.Read = True
    indis(tk, 1) = dtr(0)
    indis(tk, 2) = dtr(1)
    indis(tk, 3) = dtr(2)
    tk = tk + 1
Loop
dtr.Close()

Dim m As Integer = 1
'3 boyutlu dizinin elemanlarına göre hareketlerin mtl3_1, mtl3_2 ve mtl3_3 geçici tablolarına
aktarılması
Dim p As Integer
For p = 1 To c
    'dizinin 1.kolon elemanlarına göre satış kayıtlarının mtl3_1 tablosuna aktarımı
    cmd.CommandText = "INSERT INTO mtl3_1 ( SKOD_L3, BELGE ) " _
        & "SELECT '" & RTrim(indis(p, 1)) & " X " & RTrim(indis(p, 2)) & " X " _
        & RTrim(indis(p, 3)) & "', CAST(TARİH AS char(8))+ '-' + MGZKOD + " _
        & "'-' + CAST(FISNO AS char(5)) as BELGE " _
        & "From <assrule_kullanıcı adı> " _
        & "WHERE SKOD = '" & indis(p, 1) & "'"
    cmd.ExecuteScalar()

    'dizinin 2.kolon elemanlarına göre satış kayıtlarının mtl3_2 tablosuna aktarımı
    cmd.CommandText = "INSERT INTO mtl3_2 ( SKOD_L3, BELGE ) " _
        & "SELECT '" & RTrim(indis(p, 1)) & " X " & RTrim(indis(p, 2)) _
        & " X " & RTrim(indis(p, 3)) & "', CAST(TARİH AS char(8))+ '-' " _
        & "+ MGZKOD + '-' + CAST(FISNO AS char(5)) as BELGE " _
        & "From <assrule_kullanıcı adı> " _
        & "WHERE SKOD = '" & indis(p, 2) & "'"
    cmd.ExecuteScalar()

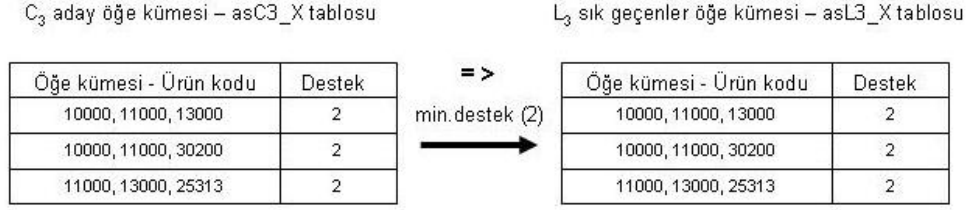
    'dizinin 3.kolon elemanlarına göre satış kayıtlarının mtl3_3 tablosuna aktarımı
    cmd.CommandText = "INSERT INTO mtl3_3 ( SKOD_L3, BELGE ) " _
        & "SELECT '" & RTrim(indis(p, 1)) & " X " & RTrim(indis(p, 2)) & " X " _
        & RTrim(indis(p, 3)) & "', CAST(TARİH AS char(8))+ '-' + MGZKOD + " _
        & "'-' + CAST(FISNO AS char(5)) as BELGE " _
        & "From <assrule_kullanıcı adı> " _
        & "WHERE SKOD = '" & indis(p, 3) & "'"
    cmd.ExecuteScalar()

Next p

'mtl3_1, mtl3_2 ile mtl3_3 tablolarının joinlenerek DESTEK seviyelerinin asL3 tablosuna aktarımı
cmd.CommandText = "INSERT INTO <asL3_kullanıcı adı> ( SKOD_L3, SUPPORT ) " _
    & "SELECT mtl3_1.SKOD_L3, COUNT(mtl3_1.BELGE) AS SUPPORT " _
    & "FROM mtl3_1 INNER JOIN " _
    & "mtl3_2 ON mtl3_1.SKOD_L3 = mtl3_2.SKOD_L3 AND " _
    & "mtl3_1.BELGE = mtl3_2.BELGE INNER JOIN " _
    & "mtl3_3 ON mtl3_1.SKOD_L3 = mtl3_3.SKOD_L3 AND " _
    & "mtl3_1.BELGE = mtl3_3.BELGE " _
    & "GROUP BY mtl3_1.SKOD_L3 " _
    & "ORDER BY SUPPORT DESC"
cmd.ExecuteScalar()

'Min.Destek seviyesinin altındaki değerler silinir
cmd.CommandText = "DELETE FROM <asL3_kullanıcı adı> " _
    & "WHERE SUPPORT < min.destek değeri "
cmd.ExecuteScalar()
```

L_3 sık geçenler öge kümesinin oluşumu örnek veriler ile Şekil IV.25’de görülmektedir.

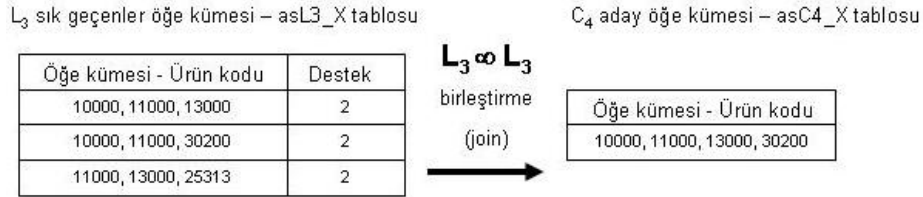


Şekil IV.25 C_3 - L_3 dönüşümünün gerçekleşmesi.

9) Sekizinci adım da, L_4 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L_3 kümesinin öğelerinin birleştirme işleminden ($L_3 \bowtie L_3$ - birleştirme) yeni bir aday öge kümesi olan C_4 oluşturulur. C_4 kümesi elde edilirken veritabanında birden fazla geçici tablo oluşturulur ve en son aşamada C_4 kümesi geçici bir tablo (asC4_<kullanıcı adı>) içine aktarılır.

C_4 aday öge kümesi oluşturulması işleminin Visual Basic.NET kodu C_3 aday öge kümesi Visual Basic.NET koduna benzeridir, sadece öge sayısı ile bağlantılı olarak kod da oluşturulan geçici tablolarında veri alanı sayısı +1 fazladır.

C_4 aday öge kümesinin oluşumu örnek veriler ile Şekil IV.26’da görülmektedir.

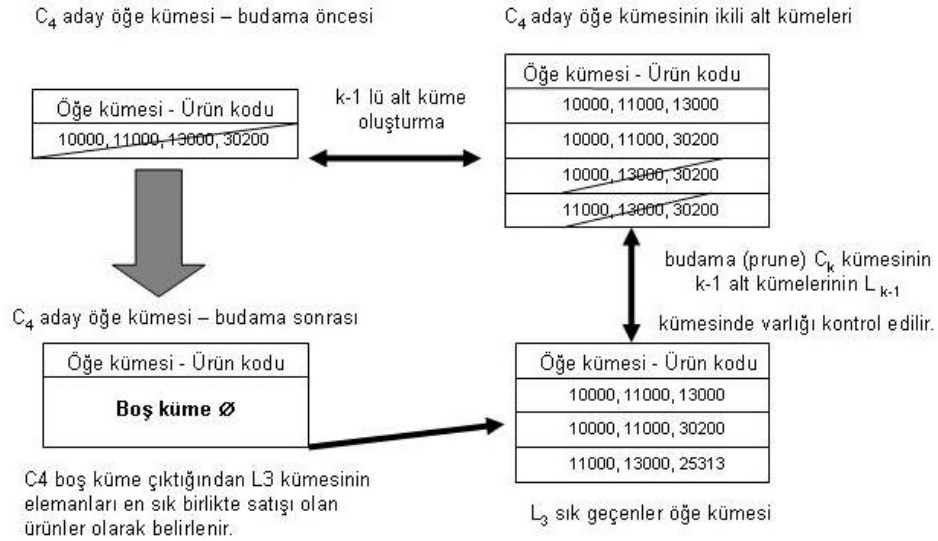


Şekil IV.26 $L_3 \bowtie L_3$ dönüşümünün gerçekleşmesi.

10) Dokuzuncu adım da, C_4 aday kümesi üzerinde yani budama işlemi uygulanır. C_4 aday öge kümesindeki öğelerin alt kümelerinin L_3 kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_3 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_4 aday öge kümesinden silinir. Budama işlemi sonucunda elenenler ve C_4 aday öge kümesinin geçici tablosundan (asC4_<kullanıcı adı>) silinen öğeler, veritabanında farklı bir geçici tabloya aktarılır.

C_4 aday öge kümesi üzerinde budama işleminin Visual Basic.NET kodu C_3 aday öge kümesi üzerinde budama işleminin Visual Basic.NET koduna benzeridir, sadece bu aşamada öge alt kümelerinin değerlerinin aktarıldığı dizi değişkeni +1 boyut fazladır.

C_4 aday öge kümesi üzerinde budama işlemi yapılması örnek veriler ile Şekil IV.27'de görülmektedir.



Şekil IV.27 Apriori algoritmasında C_4 aday öge kümesinde budama işlemi.

11) Onuncu adım da, C_4 aday kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L_4 sık geçen öge kümesi elde edilir ve ekranda görüntülenir. L_4 kümesi elde edilirken veritabanında birden fazla geçici tablo oluşturulur ve en son aşamada L_4 kümesi bir geçici tablo (asL4_<kullanıcı adı>) içinde saklanır. C_k aday kümelerinde L_k sık geçen öge kümelerini elde edilmesi esnasında C_k aday kümesindeki hiç bir öge minimum destek değerini geçemeyebilir, bu durumda algoritma sonlanmıştır ve belirlenen minimum destek değerine L_{k-1} kümesinin öğeleri en sık birliktelik satın alınan ürünler olarak belirlenir.

Bu adımlar veri kümesini durumuna göre iteratif (tekrarlayan) bir şekilde devam edebilir veya bu adımlara ulaşmadan da algoritma kesilebilir.

BÖLÜM V

SONUÇLAR

V.1. GİRİŞ

Bu bölümün amacı, birliktelik-ilişki kuralı algoritmalarından Apriori algoritmasının gerçek veri seti üzerinde uygulanarak elde edilen sonuçları değerlendirmektir.

Uygulama yazılımın üzerinde Apriori algoritmasını uygulayacağı veri bütünlüğü IV.bölüm’de belirtildiği gibi, örnek bir firmaya ait yedi adet pastane satış lokasyonunun 2002 – 2006 yılları arasındaki satış verilerinden oluşmaktadır. Veriler SQL 2000 veritabanında bütün halinde bulunmaktadır.

Verilerin boyutu oldukça büyüktür. Veri setimiz yaklaşık 1.350.000 satış hareketinden, yıl bazlı verilerin ayrıldığı dikkate alınırsa toplam dört yılın 13.000 stok kart verisinden oluşmaktadır.

V.2. ELDE EDİLEN SONUÇLAR

Geliştirilen yazılım iki farklı girdi değeri alarak çalışmaya başlamaktadır. Bunlar, algoritmanın çalıştırılacağı zaman diliminin belirlenmesi, diğeri ise satış lokasyonu seçimi yapılabilmesidir. Ayrıca satış lokasyonlarındaki satışın tipine (fişli satış, faturalı satış) göre de algoritmanın çalışmasına imkan tanımaktadır.

Yazılıma girdi değerleri olarak, 01.10.2005 – 20.10.2005 tarihleri arasında, 1140, 1150 ve 1180 şube kodlu satış lokasyonları seçildi. Satış tipi olarak ise fişli satışlar tercih edildi. Bu girdiler doğrultusunda uygulama çalıştırıldı. Şekil V.1’de kriter girişi ekran görüntüsü yer almaktadır.

Yüksek Lisans Tez Çalışması - Veri Madenciliği Uygulaması 05.07.2006 18:24:20

:: Birliktelik-ilişki (eşleştirme) kuralı ::
 :: Association Rule - Apriori algoritması ::

Terh:

Rapor tipi:
☐ Tüm belge satışları (Fiş, Fatura)
☒ Fişli satışlar
☐ Faturalı satışlar

Kod	KODU	ADI	GİRİŞ_Mİ
☐	1070	İSTİTLER PASTANESİ	Active
☐	1090	AMKARA PASTANESİ	Active
☒	1140	BEYLİKHÜZÜ PASTANESİ	Active
☒	1150	MASLAK PASTANESİ	Active
☒	1180	İSTİNYE PASTANESİ	Active

[Geri](#)

Şekil V.1 Tez yazılımının kriter (girdi) girişi ekran görüntüsü.

Belirtilen kriterler doğrultusunda veri setinden veri madenciliği uygulanacak veri çalışma ortamına çekilir ve Şekil V.2'de görülen ekran görüntüsü ile kullanıcıya inceleme amaçla sunulur. Bu aşamadan itibaren Apriori algoritması süreçleri çalışmaya başlar.

Yüksek Lisans Tez Çalışması Veri Madenciliği Uygulaması 05.07.2006 18:36:47

... Birliklilik-iliski (eşleştirme) kuralı ::
 ... Association Rule - Apriori algoritması ::

Tarih:

Rapor Tipi: ☐ Tüm belge satışları (Fis, Fatura) ☒ Fişli satışlar ☐ Faturalı satışlar

1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım

Algoritma 1. adımda, basit olarak, tüm tabloyu tarar ve her nesnenin kaç işlemde geçtiğinin bilgisine erişilir. Bu veriler apriori algoritmasının C1 aday öge kümesini temsil eder.

[Gör](#)

1348,1258,1356. kodlu mağazalar aktif durumdadır, yeni mağaza seçimi için Geniştirilmesini bekleyiniz

TARİH	MGZKOD	MGZADI	FISNO	SKOD	STKAD	BRM	MIKTAR	GRUP
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	4	25314	POĞAÇA KIYMAU (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	5	25313	POĞAÇA PEYNİRLİ (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	5	25314	POĞAÇA KIYMAU (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	5	25317	POĞAÇA SADE (NORM)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	6	25316	POĞAÇA PATATESLİ (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	7	25316	POĞAÇA PATATESLİ (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	8	32001	TEK PASTA, ŞOKOLALI EKLER	ADET	2	TURTA SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	9	22541	KEK ŞOKOLALI (S)	ADET	1	TURTA SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	10	25313	POĞAÇA PEYNİRLİ (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	11	25313	POĞAÇA PEYNİRLİ (KLASİK)	ADET	1	TUZLU SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	12	30229	TURTA GANAJ FRAMBUAZ (P)	ADET	6	TURTA SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	13	11000	PRALİN AMANDA	KG	0,445	ŞOKOLA SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	13	15120	KAHVE TOP DRAJE	KG	0,18	ŞOKOLA SATIS HESABI
01.10.2005 00:00:00	1140	BEYÜKDÜZÜ PASTANESİ	13	21300197	FİSTİK KAPLI GÜLLÜ DÜDÜRT	ADET	1	HEDİYELİK PSYA HES

Şekil V.2 Apriori Algoritması uygulanacak verinin görüntüldüğü ekran.

Apriori algoritması belirlenen veri seti üzerinde ilk aşamada, tüm veri setini tarayarak, her ögenin kaç işlem den geçtiği bilgisini tespit eder ve C_1 aday öge kümesi içinde bu veriyi tutar. Her ürünün kaç adet hareket kayıtlı içinde geçtiği bilgisi ürünün destek (support) değeri olarak belirlenir. Şekil V.3'de belirlediğimiz kriterler doğrultusunda işlem yapmak amacı ile belirlenen veri setinde hareket gören öge sayısının 195 olduğu görülmektedir. Bu aşamada minumun destek değeri belirlenir. Minumum destek değeri olarak 4 girildi.

Rapor tip: ☐ Tüm belge satışları (Fiy, Fatura) ☒ Fiyli satışlar ☐ Faturalı satışlar

Min Destek(Support) Değeri:

1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım 11. adım

2. adım da, minimum destek (support) değeri belirlenir. Bu min destek değeri yardımı ile Apriori algoritmasının en sık geçen tek öğeleri içeren L_1 öge kümesi elde edilir, başka bir ifade ile; min destek (support) değerine eşit veya yüksek destek değerine sahip olan öğeler L_1 kümesini oluştururlar.

[Geri](#)

1140,1150,1168, kodlu mağazalar seçili durumda, yeni mağaza seçimi için [Geri linkini](#) tıklayınız

C_1 aday öge kümesi- her nesnenin işlem den geçiş sayısını simgeler. C_1 kümesinde 195 adet nesne mevcuttur.

SKOD	STKAD	SUPPORT	GRUP
20000	SABLE	335	BİSKÜVİ SATIŞ HESABI
25000	PEYNİRLİ PIZZA	221	TUZLU SATIŞ HESABI
32001	TEK PASTA ŞOKOLALI EKLER	168	TURTA SATIŞ HESABI
11000	PRALİN AMANDA	137	SOKOLA SATIŞ HESABI
30209	TURTA ŞOKOLA KROKANLI (P)	118	TURTA SATIŞ HESABI
30207	TURTA DİVAN SPECIAL (P)	117	TURTA SATIŞ HESABI
69053	SU	113	YIYECEK İÇECEK HES.
80000	ÇAY	112	YIYECEK İÇECEK HES.
30213	TURTA FORENUAR (P)	111	TURTA SATIŞ HESABI
25310	POĞAÇA PEYNİRLİ (NORM)	109	TUZLU SATIŞ HESABI
12000	ŞOKOLA PORTAKAL	106	SOKOLA SATIŞ HESABI
32033	TEK PASTA ELENORE	98	TURTA SATIŞ HESABI
65303	GÜLLAÇ	95	ÜRETİM DIŞI VS. HES.
32017	TEK PASTA MARIGNY	88	TURTA SATIŞ HESABI
32034	TEK PASTA AJDA	85	TURTA SATIŞ HESABI
30208	TURTA ŞOKOLA FISTIKLI (P)	84	TURTA SATIŞ HESABI
32007	TEK PASTA GANAJ PRAMİT	83	TURTA SATIŞ HESABI
32014	TEK PASTA GANAJ FRAMBUAZ	80	TURTA SATIŞ HESABI
25317	POĞAÇA SADE (NORM)	79	TUZLU SATIŞ HESABI
30272	TURTA ELENORE (P)	73	TURTA SATIŞ HESABI
10500	MADLEN SÜTLÜ YUVARLAK	73	SOKOLA SATIŞ HESABI
25315	POĞAÇA PATATESLİ (NORM)	69	TUZLU SATIŞ HESABI
30012	TURTA POTİFUR	67	TURTA SATIŞ HESABI
25330	CEVİZLİ AY DÖREĞİ	62	TUZLU SATIŞ HESABI

Şekil V.3 C_1 aday öge kümesi ve minumum destek değeri giriş ekranı.

Destek değerleri minumum destek değerine eşit veya büyük C_1 aday öge kümesinin öğeleri L_1 sık geçen öge kümesini oluşturur. Şekil V.4'de girilen minumum destek değerine eşit veya büyük öğelerin oluşturduğu L_1 sık geçen öge kümesi 110 adet öğeden oluştuğu görülmektedir. Minumum destek değerinden küçük 85 adet ürün değerlendirme dışı kalmıştır.

Rapor tipi: ☐ Tüm belge satışları (Fiy, Fatura) ☒ Fiili satışlar ☐ Faturalı satışlar

Min Destek(Support) Değeri:

1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım 11. adım

3. adım da, L2 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L1 kümesinin öğelerinin ikili kombinasyonundan (L1-L1 birleştirme işlemi) yeni bir aday öge kümesi olan C2 oluşturulur.

[Geri](#)

1140,1150,1160 kodlu mağazalar seçili durumdadır. yeni mağaza seçimi için [Geri linkini tıklayınız](#)

L1 sık geçen öge kümesi- Min destek değerine eşit yada yüksek destek değerine sahip olan öğelerin kümesidir. L1 kümesinde 110 adet nesne mevcuttur.

SKOD	STKAD	SUPPORT	GRUP
20000	SABLE	335	BISKÜVİ SATIS HESABI
25000	PEYNIRLI PIZZA	221	TUZLU SATIS HESABI
32001	TEK PASTA ŞOKOLALI EKLER	168	TURTA SATIS HESABI
11000	PRALIN AMANDA	137	SOKOLA SATIS HESABI
30209	TURTA ŞOKOLA KROKANLI (P)	118	TURTA SATIS HESABI
30207	TURTA DİVAN SPECIAL (P)	117	TURTA SATIS HESABI
69063	SU	113	YIYECEK İÇECEK HES.
80000	ÇAY	112	YIYECEK İÇECEK HES.
30213	TURTA FORENUAR (P)	111	TURTA SATIS HESABI
25310	POĞAÇA PEYNIRLI (NORM)	108	TUZLU SATIS HESABI
12000	ŞOKOLA PORTAKAL	106	SOKOLA SATIS HESABI
32033	TEK PASTA ELENORE	98	TURTA SATIS HESABI
66303	GÜLLAÇ	95	ÜRETİM DİSİ VS. HES.
32017	TEK PASTA MARIGNY	89	TURTA SATIS HESABI
32034	TEK PASTA AIDA	85	TURTA SATIS HESABI
30208	TURTA ŞOKOLA FISTIKLI (P)	84	TURTA SATIS HESABI
32007	TEK PASTA GANAJ PRAMIT	83	TURTA SATIS HESABI
32014	TEK PASTA GANAJ FRAMBUAZ	80	TURTA SATIS HESABI
25317	POĞAÇA SADE (NORM)	79	TUZLU SATIS HESABI
30272	TURTA ELENORE (P)	73	TURTA SATIS HESABI
10500	MADLEN SÜTLÜ YUVARLAK	73	SOKOLA SATIS HESABI
25316	POĞAÇA PATATESLİ (NORM)	68	TUZLU SATIS HESABI
30012	TURTA POTIFUR	67	TURTA SATIS HESABI
25330	CEVİZLİ AY CÖREĞİ	62	TUZLU SATIS HESABI

Şekil V.4 L_1 sık geçen öge kümesi

Üçüncü adım da, L_1 kümesinin öğeleri üzerinde *apriori-gen* aşamalarından *birleştirme (join) işlemi* uygulandığında ($L_1 \bowtie L_1$) C_2 aday öge kümesi oluşur. Şekil V.5'de birleştirme işlemi sonucunda oluşan C_2 kümesinin 5995 öğeden oluştuğu görülmektedir.

Rapor tipi: ☐ Tüm belge satışları (Fiş, Fatura) ☒ Fişli satışlar ☐ Faturalı satışlar

Min.Destek(Support) Değeri:

1. adım 2. adım 3. adım **4. adım** 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım

4.adım da, C_2 aday kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L_2 sık geçen öge kümesi elde edilir.

Geri

1140,1150,1168, kodlu mağazalar seçili durumda, yeni mağaza seçimi için Geri linkini tıklayınız

C_2 aday öge kümesi- L_1 'deki elemanların ikili kombinasyonlarından ($L_1 \times L_1$) oluşmuştur. C_2 kümesinde 5995 adet nesne mevcuttur.

L1XL1	SKOD_21	SKOD_22
10500 X 11000	10500	11000
10500 X 11200	10500	11200
10500 X 11500	10500	11500
10500 X 11700	10500	11700
10500 X 11702	10500	11702
10500 X 12000	10500	12000
10500 X 12100	10500	12100
10500 X 12102	10500	12102
10500 X 12204	10500	12204
10500 X 12389	10500	12389
10500 X 13000	10500	13000
10500 X 13001	10500	13001
10500 X 13002	10500	13002
10500 X 13003	10500	13003
10500 X 13006	10500	13006
10500 X 13007	10500	13007
10500 X 13013	10500	13013
10500 X 13029	10500	13029
10500 X 13034	10500	13034
10500 X 13037	10500	13037
10500 X 13057	10500	13057
10500 X 13058	10500	13058
10500 X 13065	10500	13065
10500 X 13066	10500	13066
10500 X 13067	10500	13067

Şekil V.5 C_2 aday öge kümesi

Dördüncü adım da, C_2 aday kümesi içinde yer alan ikili öge setlerinin veri tabanı üzerinde taraması yapılarak ikili öğelerin birlikte yer aldıkları kayıt sayısı destek değeri olacak şekilde tespit edilir ve minimum destek değerine eşit veya büyük destek değerine sahip olan C_2 aday kümesi öğeleri L_2 sık geçen öge kümesini oluşturur. Yazılımın en çok zaman harcadığı aşama bu aşamadır. Çünkü C_2 aday kümesindeki ikili öğelerin sayısı ikili kombinasyon işlemine dolayı çok fazladır. Şekil V.6'da 4.adımın sonucunda ortaya çıkan L_2 sık geçen öge kümesinin 113 adet öğeden oluştuğu görülmektedir.

Rapor tipi: ☐ Tüm belge satışları (Fis, Fatura) ☒ Fişli satışlar ☐ Faturalı satışlar

Min.Destek(Support) Değeri:

1. adım 2. adım 3. adım 4. adım **5. adım** 6. adım 7. adım 8. adım 9. adım 10. adım 11. adım

5.adım da, L_3 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L_2 kümesinin öğelerinin üçlü kombinasyonundan ($L_2 \times L_2$ - birleştirme) yeni bir aday öge kümesi olan C_3 oluşturulur.

Geri

1148,150,1160 kodlu mağazalar seçili durumdadır, yeni mağaza seçimi için [Geri](#) linkini tıklayınız.

L_2 sık geçen öge kümesi- C_2 aday öge kümesinin minimum destek değerine eşit veya yüksek destek değerine sahip olan öğelerin kümesi, L_2 kümesinde 113 adet nesne mevcuttur.

SKOD_L2	SUPPORT
20000 X 25000	87
20000 X 20403	27
32014 X 32017	27
32033 X 32034	27
32014 X 32033	26
20000 X 20400	25
32001 X 32007	25
11000 X 12000	23
32014 X 32034	20
25310 X 25315	18
20000 X 20300	17
25000 X 30012	17
32017 X 32034	17
12000 X 50000	16
11000 X 11702	15
25313 X 25316	14
32017 X 32033	14
25310 X 80000	13
20300 X 25000	12
25311 X 25315	12
25311 X 80000	12
25315 X 80000	12
25000 X 80000	11
12000 X 20000	10
20000 X 32033	10

SKOD_21	SADI_21	SKOD_22	SADI_22
20000	SABLE	25000	PEYNIRLI PIZZA
20000	SABLE	20403	BUYUK ŞOKOLALI BREÇEL
32014	TEK PASTA GANAJ FRAMBUAZ	32017	TEK PASTA MARIGNY
32033	TEK PASTA ELENÖRE	32034	TEK PASTA AIDA
32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENÖRE
20000	SABLE	20400	ŞOKOLA BREÇEL BISKÜM
32001	TEK PASTA ŞOKOLALI EKLER	32007	TEK PASTA GANAJ PRAMIT
11000	PRALIN AMANDA	12000	ŞOKOLA PORTAKAL
32014	TEK PASTA GANAJ FRAMBUAZ	32034	TEK PASTA AIDA
25310	POĞAÇA PEYNIRLI (NORM)	25315	POĞAÇA PATATESLI (NORM)
20000	SABLE	20300	VIŞNELİ MEKİK
25000	PEYNIRLI PIZZA	30012	TURTA POTIFUR
32017	TEK PASTA MARIGNY	32034	TEK PASTA AIDA
12000	ŞOKOLA PORTAKAL	50000	LOKUM ŞOKOLALI
11000	PRALIN AMANDA	11702	TRUFFE SÜTLÜ
25313	POĞAÇA PEYNIRLI (KLASİK)	25316	POĞAÇA PATATESLI (KLASİK)
32017	TEK PASTA MARIGNY	32033	TEK PASTA ELENÖRE
25310	POĞAÇA PEYNIRLI (NORM)	80000	ÇAY
20300	VIŞNELİ MEKİK	25000	PEYNIRLI PIZZA
25311	POĞAÇA KIYMALI (NORM)	25315	POĞAÇA PATATESLI (NORM)
25311	POĞAÇA KIYMALI (NORM)	80000	ÇAY
25315	POĞAÇA PATATESLI (NORM)	80000	ÇAY
25000	PEYNIRLI PIZZA	80000	ÇAY
12000	ŞOKOLA PORTAKAL	20000	SABLE
20000	SABLE	32033	TEK PASTA ELENÖRE

Şekil V.6 L_2 sık geçen öge kümesi

Altıncı adım da, C_3 aday kümesi üzerinde *apriori-gen* aşamalarından *budama* (*prune*) işlemi uygulanır. C_3 aday öge kümesindeki öğelerin alt kümelerinin L_2 kümesindeki varlığı kontrol edilir, bir ögenin alt kümelerinden biri, L_2 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_3 aday öge kümesinden silinir. Budama işlemi sonucunda elenenler de ekran üzerinde kullanıcıya gösterilmektedir. Şekil V.8’de budama işlemi sonucunda oluşan C_3 kümesinin öge sayısının 439’dan 109 ögeye düştüğü görülmektedir.

Rapor tipi: ☐ Tüm belge satışları (Fig. Fatura) ☒ Fiili satışlar ☐ Faturalı satışlar

Min.Destek(Support) Değeri:

1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım 11. adım

7.adım da, C_3 aday öge kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L_3 sık geçen öge kümesi elde edilir.

[Geri](#)

1146,1150,1160, kodlu mağazalar seçili durumdadır, yeni mağaza seçimi için [Geri](#) linkini tıklayınız

C_3 aday öge kümesi- budama(prune) işlemi sonucu kalanlar. C_3 kümesinde 109 adet nesne mevcuttur. Sağdaki grid : budama sonucu kumeden çıkarılanlar.

L2X12	SKOD_C31	SKOD_C32	SKOD_C33	ÇIKARILANLAR
10500 X 11000 X 15120	10500	11000	15120	10500 X 15120 X 20000
10500 X 11000 X 20000	10500	11000	20000	11000 X 11700 X 11702
11000 X 11700 X 12000	11000	11700	12000	11000 X 11700 X 12100
11000 X 12000 X 12100	11000	12000	12100	11000 X 11700 X 15120
11000 X 12000 X 15120	11000	12000	15120	11000 X 11700 X 20000
11000 X 12000 X 20000	11000	12000	20000	11000 X 11700 X 20403
11000 X 12000 X 20403	11000	12000	20403	11000 X 11700 X 50000
11000 X 12000 X 50000	11000	12000	50000	11000 X 11702 X 12000
11000 X 20000 X 20403	11000	20000	20403	11000 X 11702 X 12100
12000 X 20000 X 20403	12000	20000	20403	11000 X 11702 X 15120
12000 X 20000 X 25000	12000	20000	25000	11000 X 11702 X 20000
20000 X 20300 X 25000	20000	20300	25000	11000 X 11702 X 20403
20000 X 20300 X 30012	20000	20300	30012	11000 X 11702 X 50000
20000 X 20300 X 65303	20000	20300	65303	11000 X 12100 X 15120
20000 X 20400 X 25000	20000	20400	25000	11000 X 12100 X 20000
20000 X 25000 X 25330	20000	25000	25330	11000 X 12100 X 20403
20000 X 25000 X 30012	20000	25000	30012	11000 X 12100 X 50000
20000 X 25000 X 30213	20000	25000	30213	11000 X 15120 X 20000
20000 X 25000 X 32001	20000	25000	32001	11000 X 15120 X 20403
20000 X 25000 X 32007	20000	25000	32007	11000 X 15120 X 50000
20000 X 25000 X 32034	20000	25000	32034	11000 X 20000 X 50000
20000 X 25000 X 40500	20000	25000	40500	11000 X 20403 X 50000
20000 X 25000 X 65303	20000	25000	65303	12000 X 12100 X 15120
20000 X 25000 X 80000	20000	25000	80000	12000 X 12100 X 20000
20000 X 25000 X 80001	20000	25000	80001	12000 X 12100 X 20403

Şekil V.8 C_3 aday öge kümesi – budama işlemi sonrası

Yedinci adım da, C_3 aday kümesine bakılır destek seviyeleri minimum destek seviyesine eşit veya yüksek olan C_3 aday öge kümesinin öğelerinden L_3 sık geçen öge kümesi oluşur. L_2 kümesinin oluşmasında olduğu gibi bu süreç uzun sürmez çünkü, budama işlemi ile öge sayısı budanmıştır. Şekil V.9'de L_3 sık geçen öge kümesinin öge sayısının 12 olduğu görülmektedir.

Rapor tipi: ☐ Tüm belge satışları (Fiy, Fatura) ☒ Fiili satışlar ☐ Faturalı satışlar

Min Destek(Support) Değeri: 1

1. Adım 2. Adım 3. Adım 4. Adım 5. Adım 6. Adım 7. Adım 8. Adım 9. Adım 10. Adım 11. Adım

8.adım da, L4 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L4 kümesinin öğelerinin dörtli kombinasyonundan (L3xL3 - birleştirme) yeni bir aday öge kümesi olan C4 oluşturulur.

Çıktı

ITEMID,LIKÖ kodlu mağazalar seçili durumda, yeni mağaza seçimi için geri tıklayınız

L3 sık geçen öge kümesi- C3 aday öge kümesinin minimum destek değerinden yüksek destek değerine sahip olan öğelerinin kümesi. L3 kümesinde 12 adet nesne mevcuttur.

SKOD_L3	SUPPORT	SKOD_31	SADI_31	SKOD_32	SADI_32	SKOD_33	SADI_33
32014 X 32033 X 32034	9	32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENORE	32034	TEK PASTA AIDA
32014 X 32017 X 32033	8	32014	TEK PASTA GANAJ FRAMBUAZ	32017	TEK PASTA MARIGNY	32033	TEK PASTA ELENORE
32014 X 32017 X 32034	8	32014	TEK PASTA GANAJ FRAMBUAZ	32017	TEK PASTA MARIGNY	32034	TEK PASTA AIDA
32017 X 32033 X 32034	8	32017	TEK PASTA MARIGNY	32033	TEK PASTA ELENORE	32034	TEK PASTA AIDA
20000 X 32014 X 32033	6	20000	SABLE	32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENORE
11000 X 12000 X 50000	5	11000	PRALIN AMANDA	12000	ŞOKOLA PORTAKAL	50000	LOKUM ŞOKOLALI
12000 X 20000 X 20403	5	12000	ŞOKOLA PORTAKAL	20000	SABLE	20403	BÜYÜK ŞOKOLALI BREÇEL
20000 X 20400 X 25000	5	20000	SABLE	20400	ŞOKOLA BREÇEL BISKÜVİ	25000	PEYNİRLİ PIZZA
25311 X 25315 X 80000	4	25311	POĞAÇA KIYMALI (NÖRM)	25315	POĞAÇA PATATESLİ (NÖRM)	80000	ÇAY
32001 X 32014 X 32033	4	32001	TEK PASTA ŞOKOLALI EKLER	32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENORE
32007 X 32014 X 32034	4	32007	TEK PASTA GANAJ PRAMİT	32014	TEK PASTA GANAJ FRAMBUAZ	32034	TEK PASTA AIDA
32007 X 32033 X 32034	4	32007	TEK PASTA GANAJ PRAMİT	32033	TEK PASTA ELENORE	32034	TEK PASTA AIDA

Şekil V.9 L_3 sık geçen öge kümesi

Sekizinci adım da, L_3 kümesinin öğeleri üzerinde birleştirme (join) işlemi uygulandığında ($L_3 \bowtie L_3$) C_4 aday öge kümesi oluşur. Şekil V.10'da birleştirme işlemi sonucunda oluşan C_4 kümesinin 1 öğeden oluştuğu görülmektedir.

İzlemler: 1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım 11. adım

9.adım da, C_4 aday kümesi üzerinde apriori-gen işlemi yani budama işlemi uygulanır. C_4 aday öge kümesindeki öğelerin alt kümelerinin L_3 kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_3 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_4 aday öge kümesinden silinir.

[Geri](#)

1140,1150,1160, kodlu mağazalar seçili durumda, yeni mağaza seçimi için [Geri](#) linkini tıklayınız

C_4 aday öge kümesi- $L_3 \times L_3$ birleştirme işlemi kullanarak L_3 'deki elemanların dörtlü kombinasyonları ile oluşturuldu. C_4 kümesinde 1 adet nesne mevcuttur.

SKOD_C41	SKOD_C42	SKOD_C43	SKOD_C44	L3XL3
32014	32017	32033	32034	32014 X 32017 X 32033 X 32034

Şekil V.10 C_4 aday öge kümesi

Dokuzuncu adım da, C_4 aday kümesi üzerinde budama (prune) işlemi uygulanır. C_4 aday öge kümesindeki öğelerin alt kümelerinin L_3 kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_3 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_4 aday öge kümesinden silinir. Budama işlemi sonucunda elenenler de ekran üzerinde kullanıcıya gösterilmektedir. Şekil V.11'de budama işlemi sonucunda oluşan C_4 kümesinin bir öğesinin budanmadığı, C_4 aday öge kümesinde 1 öğenin bulunduğu görülmektedir.

İzlemler: 1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım 11. adım

10.adım da, C_4 aday kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L_4 sıklık geçen öge kümesi elde edilir.

[Geri](#)

1140,1150,1160, kodlu mağazalar seçili durumda, yeni mağaza seçimi için [Geri](#) linkini tıklayınız

C_4 aday öge kümesi- budama(prune) işlemi sonucu kalanlar. C_4 kümesinde 1 adet nesne mevcuttur. Sağdaki grid : budama sonucu kümeden çıkarılanlar.

L3XL3	SKOD_C41	SKOD_C42	SKOD_C43	SKOD_C44	ÇIKARILANLAR
32014 X 32017 X 32033 X 32034	32014	32017	32033	32034	

Şekil V.11 C_4 aday öge kümesi – budama sonrası

Onuncu adım da, C_4 aday kümesine bakılır destek seviyeleri minimum destek seviyesine eşit veya yüksek olan C_4 aday öge kümesinin öğelerinden L_4 sık geçen öge kümesi oluşur. Mevcut C_4 aday öge kümemizle 10.adım çalışıldığında Şekil V.12'deki sonuç alındı.

Yüksek Lisans Tez Çalışması - Feriðun C. Özçakır - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites

Address: http://localhost/DMining/AssRule.aspx

Yüksek Lisans Tez Çalışması **Veri Madenciliği Uygulaması** 05.07.2006 18:52:09

...:: Birliklilik-iliski (eşleştirme) kuralı ::...

...:: Association Rule - Apriori algoritması ::...

Apriori algoritmasının uygulanma ile ilgili örnek satırların dokümanını görmek için tıklayınız.

Tarih: Başlangıç: Gün Ay Yıl: 01 10 2005 Bitiş: Gün Ay Yıl: 20 10 2005

Rapor tipi: ☐ Tüm belge satışları (Fiş, Fatura) ☒ Fişli satışlar ☐ Faturalı satışlar

Min.Destek(Support) Değeri: 5

1. adım 2. adım 3. adım 4. adım 5. adım 6. adım 7. adım 8. adım 9. adım 10. adım

Minimum destek(support) seviyesinin üzerinde kayıt kalmamıştır. destek seviyesini tekrar belirleyerek algoritmayı çalıştırmak için Geri tuşunu kullanarak 1.adıma dönerek süreci tekrar başlatınız. Böylece birlikte en sık olarak kullanılan nesneler 7.adımda belirtilen L_3 kümesidir.

[Geri](#)

1140,1150,1168, kodlu mağazalar seçili durumdadır. yeni mağaza seçimi için Geri linkini tıklayınız

Şekil V.12 C_4 aday öge kümesinin öğelerinin destek değeri < min.destek değeri,

Bu sonuca göre C_4 aday öge kümesinin öğesinin destek değeri minimum destek değerinden küçüktür ve 7 adımda elde edilen L_3 sık geçen öge kümesi en sık olarak kullanılan nesneler olarak belirlenir. L_3 sık geçen öge kümesinin 12 elemanı birlikte en sık satın alınan üçlü ürün setleridir. Bu ürünler Şekil V.9'da yer almaktadır.

Bu ürünler incelendiğinde genelde aynı ürün grubuna ait ürünlerin en sık birlikte satın alınma özelliğine sahip olduğu gözlenmektedir. Özellikle tek pasta ürünlerinin yine tek pasta ürünleri ile birlikte satın alındığı, tuzlu ürünlerin (tuzlu kuru pasta ürünleri) tatlı bisküvileri ile birlikte satın alındığı ve çikolatalı ürünlerin (şokola) yine çikolata içeren farklı gruba ait ürünler ile (şokolalı lokum, şokolalı bisküvi, vb.) birlikte satın alındığı görülmektedir.

Aynı zaman dilimi ve aynı lokasyonlar için program tekrar çalıştırıldığında ve minimum destek seviyesi 3 olarak belirlendiğinde yine algoritma L₄ seviyesine ulaşmadan kesilmiştir ve Şekil V.13'deki ekran görüntüsünde yer alan 31 adet üçlü ürün setleri en sık birlikte satın alınan ürünlerdir.

1140,128,110, kodlu mağazalar seçili durumdadır, veri mağaza seçimi için Gösterilecek Mağazalar

L3 sık geçen öge kümesi- C3 aday öge kümesinin minimum destek değerinden yüksek destek değerine sahip olan öğelerinin kümesi L3 kümesinde 31 adet nesne mevcuttur.

SKOD_L3	SUPPORT	SKOD_31	SADI_31	SKOD_32	SADI_32	SKOD_33	SADI_33
32014 X 32033 X 30034 9		32014	TEK PASTA GANAJ FRAMBUAZ	30033	TEK PASTA ELENORE	30034	TEK PASTA AIDA
32014 X 32017 X 30033 6		32014	TEK PASTA GANAJ FRAMBUAZ	32017	TEK PASTA MARIGNY	30033	TEK PASTA ELENORE
32014 X 32007 X 30034 8		32014	TEK PASTA GANAJ FRAMBUAZ	32007	TEK PASTA MARIGNY	30034	TEK PASTA AIDA
32017 X 32033 X 30034 6		32017	TEK PASTA MARIGNY	30033	TEK PASTA ELENORE	30034	TEK PASTA AIDA
20000 X 32014 X 30033 6		20000	SABLE	32014	TEK PASTA GANAJ FRAMBUAZ	30033	TEK PASTA ELENORE
11000 X 12000 X 50000 5		11000	PRALIN AMANDA	12000	ŞOKOLA PORTAKAL	50000	LOKUM ŞOKOLALI
12000 X 20000 X 20403 5		12000	ŞOKOLA PORTAKAL	20000	SABLE	20403	BÜYÜK ŞOKOLALI BREÇEL
20000 X 20400 X 25000 5		20000	SABLE	20400	ŞOKOLA BREÇEL BISKÜVİ	25000	PEYNİRLİ PIZZA
25311 X 25315 X 80000 4		25311	POĞAÇA KIYMALI (NORM)	25315	POĞAÇA PATATESLİ (NORM)	80000	ÇAY
32001 X 32014 X 30033 4		32001	TEK PASTA ŞOKOLALI EKLER	32014	TEK PASTA GANAJ FRAMBUAZ	30033	TEK PASTA ELENORE
32007 X 32014 X 30034 4		32007	TEK PASTA GANAJ PRAMİT	32014	TEK PASTA GANAJ FRAMBUAZ	30034	TEK PASTA AIDA
32007 X 32033 X 30034 4		32007	TEK PASTA GANAJ PRAMİT	30033	TEK PASTA ELENORE	30034	TEK PASTA AIDA
11000 X 11700 X 12000 3		11000	PRALIN AMANDA	11700	TRUFFE HINDISTAN CEMİZLİ	12000	ŞOKOLA PORTAKAL
11000 X 12000 X 20403 3		11000	PRALIN AMANDA	12000	ŞOKOLA PORTAKAL	20403	BÜYÜK ŞOKOLALI BREÇEL
12000 X 20000 X 25000 3		12000	ŞOKOLA PORTAKAL	20000	SABLE	25000	PEYNİRLİ PIZZA
20000 X 20300 X 20400 3		20000	SABLE	20300	VIŞNELİ MEKİK	20400	ŞOKOLA BREÇEL BISKÜVİ
20000 X 20300 X 25000 3		20000	SABLE	20300	VIŞNELİ MEKİK	25000	PEYNİRLİ PIZZA
20000 X 20300 X 25330 3		20000	SABLE	20300	VIŞNELİ MEKİK	25330	CEMİZLİ AY ÇÖREĞİ
20000 X 20300 X 32007 3		20000	SABLE	20300	VIŞNELİ MEKİK	32007	TEK PASTA GANAJ PRAMİT
20000 X 20403 X 25000 3		20000	SABLE	20403	BÜYÜK ŞOKOLALI BREÇEL	25000	PEYNİRLİ PIZZA
20000 X 25000 X 30012 3		20000	SABLE	25000	PEYNİRLİ PIZZA	30012	TURTA POTİFUR
20000 X 32001 X 32007 3		20000	SABLE	32001	TEK PASTA ŞOKOLALI EKLER	32007	TEK PASTA GANAJ PRAMİT
20000 X 32007 X 30033 3		20000	SABLE	32007	TEK PASTA GANAJ PRAMİT	30033	TEK PASTA ELENORE
20000 X 32007 X 30034 3		20000	SABLE	32007	TEK PASTA GANAJ PRAMİT	30034	TEK PASTA AIDA
20000 X 32014 X 30034 3		20000	SABLE	32014	TEK PASTA GANAJ FRAMBUAZ	30034	TEK PASTA AIDA
20000 X 32033 X 30034 3		20000	SABLE	32033	TEK PASTA ELENORE	30034	TEK PASTA AIDA
20300 X 25000 X 30012 3		20300	VIŞNELİ MEKİK	25000	PEYNİRLİ PIZZA	30012	TURTA POTİFUR
25311 X 25315 X 25317 3		25311	POĞAÇA KIYMALI (NORM)	25315	POĞAÇA PATATESLİ (NORM)	25317	POĞAÇA SADE (NORM)
32001 X 32007 X 32014 3		32001	TEK PASTA ŞOKOLALI EKLER	32007	TEK PASTA GANAJ PRAMİT	32014	TEK PASTA GANAJ FRAMBUAZ
32001 X 32007 X 30034 3		32001	TEK PASTA ŞOKOLALI EKLER	32007	TEK PASTA GANAJ PRAMİT	30034	TEK PASTA AIDA
32001 X 32014 X 30034 3		32001	TEK PASTA ŞOKOLALI EKLER	32014	TEK PASTA GANAJ FRAMBUAZ	30034	TEK PASTA AIDA

Şekil V.13 L₃ sık geçen öge kümesi-01.10-20.10.2005 , minumum destek=3

01.04 – 30.04.2004 tarihleri arasında 1080, 1150 ve 1180 kodlu satış lokasyonları için minumum destek seviyesi 4 olarak belirlenerek çalıştırılan programın çıktısı Şekil V.14'de görülmektedir.

SKOD_L3	SUPPORT	SKOD_31	SADI_31	SKOD_32	SADI_32	SKOD_33	SADI_33
25310 X 25315 X 25318	13	25310	POĞAÇA PEYNİRLİ (NORM)	25315	POĞAÇA PATATESLİ (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)
25310 X 25311 X 25318	12	25310	POĞAÇA PEYNİRLİ (NORM)	25311	POĞAÇA KIYMALI (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)
25311 X 25315 X 25318	12	25311	POĞAÇA KIYMALI (NORM)	25315	POĞAÇA PATATESLİ (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)
20000 X 20300 X 25000	11	20000	SABLE	20300	VIŞNELİ MEKİK	25000	PEYNİRLİ PIZZA
25310 X 25317 X 25318	11	25310	POĞAÇA PEYNİRLİ (NORM)	25317	POĞAÇA SADE (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)
25313 X 25314 X 25318	11	25313	POĞAÇA PEYNİRLİ (KLASİK)	25314	POĞAÇA KIYMALI (KLASİK)	25318	POĞAÇA PATATESLİ (KLASİK)
25310 X 25311 X 25315	10	25310	POĞAÇA PEYNİRLİ (NORM)	25311	POĞAÇA KIYMALI (NORM)	25315	POĞAÇA PATATESLİ (NORM)
32014 X 32033 X 32034	10	32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENORE	32034	TEK PASTA AIDA
25311 X 25315 X 25317	9	25311	POĞAÇA KIYMALI (NORM)	25315	POĞAÇA PATATESLİ (NORM)	25317	POĞAÇA SADE (NORM)
25311 X 25317 X 25318	8	25311	POĞAÇA KIYMALI (NORM)	25317	POĞAÇA SADE (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)
25310 X 25311 X 25317	7	25310	POĞAÇA PEYNİRLİ (NORM)	25311	POĞAÇA KIYMALI (NORM)	25317	POĞAÇA SADE (NORM)
32001 X 32014 X 32033	7	32001	TEK PASTA ŞOKOLALI EKLER	32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENORE
32001 X 32033 X 32039	7	32001	TEK PASTA ŞOKOLALI EKLER	32033	TEK PASTA ELENORE	32039	TEK PASTA CHARME
32014 X 32033 X 32039	7	32014	TEK PASTA GANAJ FRAMBUAZ	32033	TEK PASTA ELENORE	32039	TEK PASTA CHARME
20000 X 20400 X 25000	6	20000	SABLE	20400	ŞOKOLA BREÇEL BISKÜVİ	25000	PEYNİRLİ PIZZA
25310 X 25315 X 25317	6	25310	POĞAÇA PEYNİRLİ (NORM)	25315	POĞAÇA PATATESLİ (NORM)	25317	POĞAÇA SADE (NORM)
25310 X 25318 X 60000	6	25310	POĞAÇA PEYNİRLİ (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)	60000	ÇAY
25315 X 25317 X 25318	6	25315	POĞAÇA PATATESLİ (NORM)	25317	POĞAÇA SADE (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)
32001 X 32033 X 32034	6	32001	TEK PASTA ŞOKOLALI EKLER	32033	TEK PASTA ELENORE	32034	TEK PASTA AIDA
32014 X 32017 X 32033	6	32014	TEK PASTA GANAJ FRAMBUAZ	32017	TEK PASTA MARIGNY	32033	TEK PASTA ELENORE
32033 X 32034 X 32039	6	32033	TEK PASTA ELENORE	32034	TEK PASTA AIDA	32039	TEK PASTA CHARME
20000 X 49001 X 49021	5	20000	SABLE	49001	DİVAN ÇİKOLATALI GOFRET - SÜTLÜ	49021	DİVANUTİ
25313 X 25316 X 25330	5	25313	POĞAÇA PEYNİRLİ (KLASİK)	25316	POĞAÇA PATATESLİ (KLASİK)	25330	CEVİZLİ AY ÇÖREĞİ
32014 X 32017 X 32034	5	32014	TEK PASTA GANAJ FRAMBUAZ	32017	TEK PASTA MARIGNY	32034	TEK PASTA AIDA
32014 X 32034 X 32039	5	32014	TEK PASTA GANAJ FRAMBUAZ	32034	TEK PASTA AIDA	32039	TEK PASTA CHARME
32017 X 32033 X 32039	5	32017	TEK PASTA MARIGNY	32033	TEK PASTA ELENORE	32039	TEK PASTA CHARME
32017 X 32034 X 32039	5	32017	TEK PASTA MARIGNY	32034	TEK PASTA AIDA	32039	TEK PASTA CHARME
49001 X 49021 X 45122	5	49001	DİVAN ÇİKOLATALI GOFRET - SÜTLÜ	49021	DİVANUTİ	45122	MAJESTİK-386
10500 X 11000 X 15120	4	10500	MADLEN SÜTLÜ YUVARLAK	11000	PRALİN AMANDA	15120	KAHVE TOP DRAJE
11000 X 20000 X 20300	4	11000	PRALİN AMANDA	20000	SABLE	20300	VIŞNELİ MEKİK
15120 X 49001 X 49021	4	15120	KAHVE TOP DRAJE	49001	DİVAN ÇİKOLATALI GOFRET - SÜTLÜ	49021	DİVANUTİ
20000 X 20300 X 32033	4	20000	SABLE	20300	VIŞNELİ MEKİK	32033	TEK PASTA ELENORE
20000 X 20400 X 32001	4	20000	SABLE	20400	ŞOKOLA BREÇEL BISKÜVİ	32001	TEK PASTA ŞOKOLALI EKLE
20000 X 32034 X 32039	4	20000	SABLE	32034	TEK PASTA AIDA	32039	TEK PASTA CHARME
20300 X 20400 X 25000	4	20300	VIŞNELİ MEKİK	20400	ŞOKOLA BREÇEL BISKÜVİ	25000	PEYNİRLİ PIZZA
25000 X 25310 X 25318	4	25000	PEYNİRLİ PIZZA	25310	POĞAÇA PEYNİRLİ (NORM)	25318	POĞAÇA ZEYTİNLİ (NORM)

Şekil V.14 L3 sık geçen öge kümesi-01.04-30.04.2004 , minumum destek=4

BÖLÜM VI

DEĞERLENDİRME VE ÖNERİLER

VI.1. DEĞERLENDİRME

Veri madenciliği tekniklerinden Birliktelik-ilişki kuralı, önceden elde edilmiş verilerin üzerinde analiz işlemleri yapılarak verilerdeki birliktelik ilişkilerinin keşdedilmesini ve elde edilen örüntüler doğrultusunda geleceğe yönelik çalışmalar yapılmasına imkan tanır.

Apriori algoritması, veri madenciliğinde sık geçen öğelerin keşfedilmesi için kullanılan en çok bilinen birliktelik-ilişki kuralı algoritmasıdır. Bu algoritmanın genel mantığı belirlenen bir veri setinin tekrar tekrar taranması ile veri seti içinde birlikte sık geçen verilerin bulunması üzerine kurulmuştur. Bu algoritma özellikle pazar sepeti analizinde yoğun olarak kullanılmaktadır.

Bu çalışmada Apriori algoritması bir firmanın Pastane birimlerindeki müşterilere yapılan satış verileri üzerinde kullanıldı. Algoritmanın veriler üzerinde çalışması için bu çalışma kapsamında web ortamında çalışan bir uygulama yazılımı geliştirildi. Yazılım algoritma süreçlerinin aşamalı olarak çalıştırılmasına imkan tanıyacak şekilde gerçekleştirildi. Bu özellikliğin sağladığı ortamla algoritmanın her bir aşamasında kullanılan tekniklerin daha rahat anlaşılabilmesine ve her bir sürece ait sonuçların izlenebilmesine imkan tanındı.

Apriori algoritması uygulama yazılımı ile birlikte pastane veri seti üzerinde uygulandığında farklı zaman dilimi, farklı satış lokasyonuna göre ürünler arasında birlikte satın alınma bağıntıları bulunduğu gözlemlendi. Genelde aynı ürün grubu

içinde yer alan ürünlerin birlikte satın alınma davranışlarına sahip oldukları görüldü.

Bu bağıntılardan bazıları;

- Tuzlu grubu ürünleri olarak nitelendirilen tuzlu kuru pastalar (küçük peynirli pizza, chester, vb.) bisküvi ürünü grubu tatlı bisküvilerle (sable, şokolalı bisküvi, vb.) birlikte sıklıkla satın alınmaktadırlar.
- Poğaçı, açma, simit tarzı ürünler ise genelde aynı ürün grubu içindeki diğer ürünler ile birlikte satın alındığı gözlenmektedir.
- Bisküvi grubuna ait ürünler ise tüm ürün grupları ile birlikte satın alınma özelliğine sahip görünmektedirler.
- Pasta ürün grubunda yer alan ürünler, genelde kendi ürün grubundaki ürünler ile birlikte satın alınmaktadırlar.
- Çikolatalı ürünler ise yine içeriğinde çikolata içeren ürünler ile birlikte müşteri tarafından tercih edilmiş durumdadırlar.
- Ayrıca tuzlu ürünleri (poğaçı, açma, küçük peynirli pizza, vb.) ile birlikte sıcak içecek satışları da dikkat çekmektedir.

Farklı zaman dilimi ve farklı lokasyon kriterleri ile belirlenen veri üzerinden elde edilen birliktelik ilişki bilgilerinin doğru bir şekilde değerlendirilmesini ancak sektörün üretim ve pazarlama çalışanları yapacaktır. Sektör çalışanları elde edilen bu birliktelik-ilişki verileri ile üretim planlama aşamalarını tekrardan yapılandırabilir, satışta yer alan ürün gamını düzenleyebilir, satış lokasyon içindeki reyon düzenlemesini ürün birlikteliklerini dikkate alarak satış rakamlarını destekleyecek şekilde değiştirebilir, dönemsel satışlardan elde ettikleri sonuçlar doğrultusunda dönemsel satış kampanyaları düzenleyebilirler.

Çalışma kapsamınsa kullanılan veri boyutunun büyük olmasında dolayı, algoritma aşamalarında çalıştırılan sorguların sonuç üretmelerinin zaman aldığı gözlemlenmiştir.

VI.2. ÖNERİLER

Çalışma kapsamında geliştirilen yazılım geliştirmeye açıktır. Gerçek verilerden bu çalışma için hazırlanan veritabanı satış verileri farklı veri madenciliği algoritmalarının uygulanmasına imkan tanıyacak yapıdadır. Ürün gruplarının sınıflandırılması için Sınıflama ve kümeleme algoritmaları bu veri seti üzerinde rahatlıkla uygulanabilir.

Veri boyutunun büyük olması sorgu zamanlarının uzamasına neden olmuştur, daha yüksek teknik özellikler sahip bilgisayar donanımları ile veri madenciliği işleminde daha hızlı bir şekilde tutarlı örüntüler elde edilecektir.

KAYNAKÇA

- [1] Alataş, B.; Akın, E.: “Veri Madenciliğinde Yeni Yaklaşımlar”, *YA/EM'2004 - Yöneylem Araştırması/Endüstri Mühendisliği - XXIV Ulusal Kongresi*, Gaziantep – Adana, (15-18 Haziran **2004**).
- [2] Berry, M.J.A.; Linoff, G.S.: “Data Mining Techniques for Marketing, Sales and Customer Support”, *John Wiley & Sons*, (**1997**).
- [3] Sever, H.; Oğuz, B.: “Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım, Kısım1:Eşleştirme Sorguları ve Algoritmalar”, *Bilgi Dünyası*, 3(2), (Ekim **2002**) 173-204.
<http://www.baskent.edu.tr/~sever/bilgi-p01-03.pdf> (Erişim Tarihi : Nisan **2006**).
- [4] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: “The KDD process for extracting useful knowledge from volumes of data.”, *Communications of ACM*, 39(11), (**1996**) 27-34.
- [5] Agrawal, R.; Srikant, R.: “Mining Sequential Patterns.”, *11th International Conference on Data Engineering*, Taipei, Taiwan, (**1995**).
- [6] Oğuzlar, A.: “Veri Ön İşleme”, *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, Sayı: 21, (Temmuz-Aralık **2003**).
- [7] Holsheimer, M.; Siebes, A.P J. M.: “Data mining: the search for knowledge in databases.”, *Technical Report CS-R9406*, Amsterdam, The Netherlands, (**1994**).
- [8] Aydoğan, F.: “E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı ve Gerçekleştirimi”, *Yüksek Lisans Tezi*, Hacettepe Üniversitesi, Ankara, (**2003**).
- [9] Frawley, W. J.; Piatetsky-Shapiro, G.; Matheus, C. J.: “Knowledge discovery databases: An overview, In Knowledge Discovery In Databases”, *AAAI 58 AI Magazine* ,Cambridge, (**1991**).
- [10] Han, J.; Kamber, M.: “Data Mining: Concepts and Techniques”, *Morgan Kaufmann Publishers, Inc.*, San Francisco, California, (**2001**).
- [11] Bulun, M.; Tuğ, E.; Şakiroğlu, M.: “Tıbbi Veri Tabanlarında Gizli Bilgilerin Keşfedilmesi”, *Türkiye Bilişim Derneği 20. Bilişim Kurultayı*, (**2003**).

- [12] Cambazoğlu, T.: “Veri ambarı [Data Warehousing) temelleri”, (2000) www.bilisimrehber.com.tr/arastirma/tr_arastirma_veriambari_temelleri.phtml (Erişim Tarihi : Mart 2006).
- [13] Zhu, H.; “On-Line Analytical Mining Of Association Rules”, *MSc Thesis*, Simon Fraser University, Ottawa, Canada, (1998).
- [14] Tang, Z.; MacLennan, J.: “Data Mining with SQL Server 2005”, *Wiley Publishing, Inc.*, Indianapolis, Indiana, USA, (2005).
- [15] Hand, D.; Mannila, H.; Smyth, P.: “Principles of Data Mining (Adaptive Computation and Machine Learning)”, *The MIT Press*, (August 2001)
- [16] Akpınar, H.: “Veri tabanlarında bilgi keşfi ve veri madenciliği”, *İ.Ü. İşletme Fakültesi Dergisi, Nisan 2000 Sayısı*, C:29, (Nisan 2000) 1.
- [17] Kona, H. V.: “Association Rule Mining Over Multiple Databases: Partioned and Incremental Approaches”, *The University of Texas, Arlington*, (December 2003)
- [18] Weiss, S. M.; Kulikowski, C. A.: “Computer systems that learn: classification and prediction methods from statistics, Neural Nets, Machine Learning, and Expert Systems”, *Morgan Kaufman*, (1991)
- [19] Baykal, N.: “Veritabanı ve Veri Madenciliği”, *Tıp Bilişimi Güz Okulu Ders Notları*, Orta Doğu Teknik Üniversitesi, , Ankara, (Ekim 2003). www.turkmia.org/file/231verimadenciligi_baykal.ppt (Erişim Tarihi : Mart 2006).
- [20] ÖZEKES, S.: “Veri Madenciliği Modelleri ve Uygulama Alanları”, *İstanbul Ticaret Üniversitesi Dergisi*, No.3, (2003) 65-82.
- [21] Ramkumar, G.D.; Swami, A.: “Clustering Data Without Distance Functions”, *IEEE Bulletin of the Technical Committee on Data Engineering*, Vol.21 No.1, (March 1998) 9-14.
- [22] Agrawal, R.; Imielinski, T.; Swami, A.: “Mining association rules between sets of items in large databases.”, *ACM SIGMOD Conference on Management of Data*, Washington, (1993).
- [23] Ding, Q.: “Association Rule Mining on Remotely Sensed Imagery Using P-Trees”, *PhD Thesis*, Fargo, North Dakota, (April 2002)
- [24] Agrawal, R.; Srikant, R.: “Mining Sequential Patterns.” *11th International Conference on Data Engineering*, Taipei, Taiwan. (1995)
- [25] Zaki, M.J.; Ogihara, M.: “Theoretical foundations of association rules.”, *3rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, Seattle, WA, (June 1998).
- [26] Chen, M.S.; Han, J.; Yu, P. S.: “Data Mining: An Overview from a Database Perspective.”, *IEEE Transactions on Knowledge and Data Engineering*, 8(6), (1996) 866-883.
- [27] Genius Yazılımının Text Transfer Dosya Desenleri, POS A.Ş Web sitesi, <http://www.pos.com.tr/destek/genius/GeniusOPAInterface.pdf> (Erişim Tarihi: Kasım 2005)

- [28] Agrawal, R.; Srikant, R.: “Fast Algorithms for Mining Association Rules”, *Proceedings of the VLDB*, Santiago de Chile, Chile, (September **1994**)
- [29] Gao, W.: “A Hierarchical Document Clustering Algorithm”, *MSc Thesis*, Dalhousie University, Halifax, Nova Scotia, (April **2004**)

ÖZGEÇMİŞ

FERİDUN CEMAL ÖZÇAKIR

02 Temmuz 1971 yılında Sakarya'da doğdu. İlk öğrenimini İstanbul İçerenköy Yahya Kemal Beyatlı İlkokulunda, Orta öğrenimini İçerenköy İlköğretim Okulunda ve Lise öğrenimini Haydarpaşa Teknik Lisesinde tamamladı.

1989 yılında girdiği Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Teknolojisi Eğitimi Öğretmenliği (ing.) bölümünden 1994 yılında iyi dereceyle mezun oldu.

1995 yılında Marmara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar-Kontrol Eğitimi programında yüksek lisansa başladı. 1999 ve 2002 yıllarında başarısızlıktan dolayı kayıtlı silindi. 2005 yılında 5316 sayılı kanundan yararlanarak yeniden yüksek lisans eğitimine başladı.

1994 – 1996 yılları arasında Kocaeli Hereke Nuh Çimento Endüstri Meslek ve Teknik Lisesinde Bilgisayar öğretmeni olarak görev yaptı. 1997 yılında özel bir şirkette Bilgi Sistemleri Yöneticisi olarak göreve başladı. Halen aynı şirkette Bilgi Sistemleri Yöneticisi olarak görevine devam etmektedir.

Evli ve iki çocuk babasıdır.