

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ  
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

MÜŞTERİ İLİŞKİLERİ YÖNETİMİ (MIY) NDE  
VERİ MADENCİLİĞİ METODLARI  
KULLANARAK MÜŞTERİ PROFİLİ  
OLUŞTURMA

140040

Ahmet BAŞGÖZE  
YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ  
ANABİLİM DALI

TEZ DANIŞMANI  
Yrd.Doç.Dr. Mehmet GÖKTÜRK

GEBZE

2003



Ahmet BAŞGÖZE'nin tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 22/07/2003 tarih ve 2003/2.3... sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

### JÜRİ

ÜYE : Yrd. Doç. Dr. Mehmet GÖKTÜRK  
(Tez Danışmanı)

ÜYE : Yrd. Doç. Dr. F. Erdoğan SEVİNGEN

ÜYE : Yrd. Doç. Dr. İlyas KANDEMİR

### ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 30.10.2003 tarih ve 2003/35... sayılı kararı.



Prof. Dr. Adem KILIÇ  
GYTE Mühendislik ve Fen Bil.  
Enstitüsü Müdürü

## ÖZET



Veri madenciliği, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir.

Veri Madenciliği başta elektronik ticaret verileri olmak üzere birçok uygulama alanında kullanılmaktadır. Organizasyonların kullandığı veri miktarı genelde çok büyük boyutlarda olduğundan bu verilerden faydalı bilgiler çıkarmak oldukça zor olmaktadır. Müşterilerinin karlılık oranının belirlenmesi ve ürün promosyonları için en uygun müşterilerin belirlenmesi elektronik ticaret ve büyük miktarda veri kullanan kuruluşlarda önemli problemler arasında gösterilmektedir.

Özellikle pazarlama sektöründe yer alan kuruluşların bu problemlerin üzerine gidebilmeleri için müşterilerini iyi tanımaları gerekmektedir. Günümüzde birçok organizasyon veri madenciliği yöntemlerini kullanarak müşterileri profillerini belirleyebilmekte ve müşterilerini daha iyi tanıma olanağı bulmaktadır.

Bu amaçlar hedef alınarak geliştirilen GYTE-Miner uygulamasında müşteri profili belirlemeye yönelik genel problemlerin üzerine gidilmiştir. Yapılan çalışmada müşterilerin alışveriş bilgilerinde yola çıkarak kişisel profilleri üretilmektedir.

Yapılan çalışmada birbirinden tamamen bağımsız iki yöntem kullanılmıştır. Birinci yöntemde; klasik veri madenciliği metodları kullanarak müşteri profilini ortaya çıkaracak kurallar üretilmekte ve bu kurallardan müşteri profilini elde edecek bilgi ortaya çıkarılmaktadır. İkinci yöntemde ise veritabanındaki müşteri hareketleri tablosu üzerinden kendi geliştirdiğimiz bir algoritma doğrultusunda kompleks sorgular yürütmekte ve gerekli müşteri profilini ortaya çıkarmaktadır.

Elde edilen sonuçlar her iki yöntemden faydalanılarak müşteri profili üretilebileceğini, ikinci yöntemin daha sağlam sonuçlar ürettiğini göstermiştir. Yöntemlerin testi için geliştirilen uygulama programı Oracle 9i veritabanı ve Java programlama dili kullanılarak geliştirilmiştir. Geliştirilen uygulama ileride bu doğrultudaki çalışmalara temel olabilecek derecede esnek ve genişletilebilir bir yapıya sahiptir.

## SUMMARY



Data mining is identifying the data patterns, relations, changes, disorders, rules and structures which are crucial statistically in a semi automated fashion.

Data mining methods have been increasingly popular and widely used in various application areas, especially on electronic commerce data. The amount of data that organizations have accumulated is enormous, and extracting beneficial information is becoming harder. One of the classical problems of e-commerce is to determine the profitability of customers, finding customers who are most likely to purchase the product.

Especially the companies in marketing business, to overcome these problems they need to know their customers well. Today many organisations are able to identify their customer profiles and know their customers well by data mining.

For this purpose, in this study we have developed a system called GYTE-Miner to effectively answer common questions regarding customer profiles. Our method constructs personal profiles based on customer transaction history data.

Two different method have been implemented in application. The first method uses classical data mining algorithms to discover rules which will lead to describe customer profiles. The second method queries the customer's transactional data and directly generates statistical information about customers.

The obtained results proved that using the both conventional data mining techniques. Rules for customers can be developed statistically and customer lists can be sorted out for particular product promotions. It is studied that the second algorithm we have done accomplished better.

The application is developed using Oracle 9i Database and Java programming language. The application developed is flexible and extendible enough to be a base for similar kind of applications.

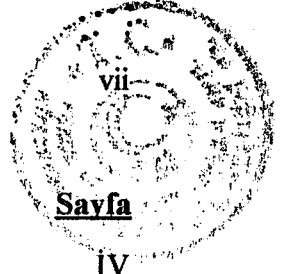


## TEŞEKKÜR

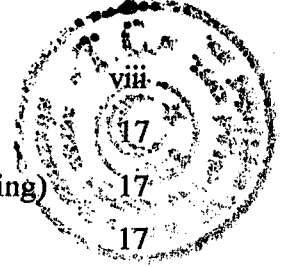
Tez için yaptığım araştırmalarımda desteğini hiçbir zaman esirgemeyen danışmanım Yrd. Doç. Dr. Mehmet GÖKTÜRK' e verdiği okul izinleri ve sağladığı geniş imkanlarından dolayı enstitü müdürümüz Sayın Önder YETİŞ'e, proje yöneticimiz Sayın Çağrı KOÇ GÜNDAR ve grup arkadaşlarıma ve çalışmalarımı bana rahat bir ortam hazırlayan eşim Meltem BAŞGÖZE' ye çok teşekkür ederim.



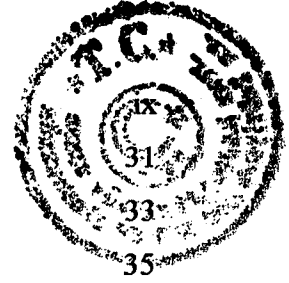
# İÇİNDEKİLER DİZİNİ



|                                                           |     |
|-----------------------------------------------------------|-----|
| ÖZET                                                      | IV  |
| SUMMARY                                                   | V   |
| TEŞEKKÜR                                                  | Vi  |
| İÇİNDEKİLER DİZİNİ                                        | vii |
| SİMGELER VE KISALTMALAR DİZİNİ                            | xi  |
| ŞEKİLLER DİZİNİ                                           | xii |
| TABLolar DİZİNİ                                           | xiv |
| 1. GİRİŞ                                                  | 1   |
| 1.1. Uygulama                                             | 1   |
| 1.2. Veri Madenciliği                                     | 2   |
| 1.3. MIY (Müşteri ilişkileri Yönetimi)                    | 5   |
| 1.4. Veri Madenciliği ve MIY                              | 5   |
| 2. VERİ MADENCİLİĞİ YÖNTEMLERİ                            | 7   |
| 2.1. Güven (Confidence) Parametresi                       | 7   |
| 2.2. Destek (Support) Parametresi                         | 7   |
| 2.3. Veri Madenciliği Modelleri                           | 8   |
| 2.3.1. Sınıflama ve Regresyon Modelleri                   | 9   |
| 2.3.2. Kümeleme Modelleri                                 | 9   |
| 2.3.3. BirlikteKil Kuralları ve Ardışık Zamanlı Örüntüler | 9   |
| 2.3.3.1. Apriori                                          | 11  |
| 2.3.3.2. TBAR (Transaction Based Association Rule)        | 11  |
| 2.3.3.3. Apriori CS                                       | 11  |
| 2.3.3.4. TBAR Cache                                       | 11  |
| 2.3.4. Sınıflandırma Algoritmaları                        | 11  |
| 2.3.4.1. Karar Ağaçları                                   | 12  |
| 2.3.4.2. STAR Metodu                                      | 14  |
| 2.3.4.3. BirlikteKil Kural Tabanlı Sınıflandırma          | 14  |
| 2.4. Veri Tabanlarında Bilgi Keşfi Süreci                 | 15  |
| 2.4.1. Hedefin Tanımlanması                               | 16  |
| 2.4.2. Verilerin Hazırlanması                             | 16  |
| 2.4.2.1. Veri Toplama (Collection)                        | 16  |



|                                                                            |    |
|----------------------------------------------------------------------------|----|
| 2.4.2.2. Değer Biçme (Assessment)                                          | 17 |
| 2.4.2.3. Birleştirme ve Temizleme (Consolidation and Cleaning)             | 17 |
| 2.4.2.4. Seçim (Selection)                                                 | 17 |
| 2.4.2.5. Dönüştürme (Transformation)                                       | 18 |
| 2.4.3. Modelin Kurulması ve Değerlendirilmesi                              | 18 |
| 2.4.4. Modelin Kullanılması                                                | 19 |
| 2.4.5. Modelin İzlenmesi                                                   | 20 |
| 2.5. Kural Keşfi                                                           | 20 |
| 2.5.1. Kural Doğrulama (Rule Validation)                                   | 20 |
| 2.5.1.1. Benzerlik Tabanlı Kural Gruplama (Similarity Based Rule Grouping) | 20 |
| 2.5.1.2. Şablon tabanlı kural filtreleme (Template-based rule filtering)   | 21 |
| 2.5.1.3. Gereksiz kural elemesi (Redundant-rule elimination)               | 21 |
| 2.6. Veri Madenciliğindeki Problemler                                      | 21 |
| 2.6.1. Sınırlı Bilgi                                                       | 21 |
| 2.6.2. Gürültü ve Eksik Değerler                                           | 22 |
| 2.6.3. Belirsizlik                                                         | 22 |
| 2.6.4. Ebat, Güncellemeler ve Konu Dışı Sahalar                            | 22 |
| 3. MÜŞTERİ PROFİLİNİN OLUŞTURULMASI                                        | 23 |
| 3.1. Veri Modeli                                                           | 23 |
| 3.2. Profil Modeli                                                         | 23 |
| 3.3. Hipotez                                                               | 24 |
| 3.4. Yöntem                                                                | 25 |
| 3.4.1. Yöntem 1                                                            | 25 |
| 3.4.2. Yöntem 2                                                            | 26 |
| 3.4.3. Kural Keşfi                                                         | 26 |
| 3.4.4. Alternatif kriterlere Duyulan İhtiyaç                               | 27 |
| 3.4.5. Kural Doğrulama (Rule Validation)                                   | 28 |
| 3.4.5.1. Benzerlik Tabanlı Kural Gruplama (Similarity Based Rule Grouping) | 29 |
| 3.4.5.2. Şablon Tabanlı Kural Filtreleme (Template-Based Rule Filtering)   | 29 |
| 3.4.5.3. Gereksiz kural elemesi (Redundant-rule elimination)               | 30 |



|                                                    |    |
|----------------------------------------------------|----|
| 3.5. Ürün Promosyonu                               |    |
| 3.6. Gruplama Penceresi                            |    |
| 4. UYGULAMA                                        |    |
| 4.1. Uygulamanın Genel Yapısı                      | 35 |
| 4.2. Kural Doğrulama                               | 37 |
| 4.3. Veritabanı                                    | 40 |
| 4.3.1. Veritabanı Mimarisi                         | 40 |
| 4.4. Yazılım Araçları                              | 41 |
| 4.4.1. Oracle 9i                                   | 42 |
| 4.4.2. Java ve Jbuilder8                           | 42 |
| 4.5. Uygulamada Kullanılan Metodlar                | 42 |
| 4.5.1. SQL Tabanlı Kural Üretimi                   | 42 |
| 4.5.2. Birliktelik Kuralları                       | 43 |
| 4.5.2.1. Apriori algoritması                       | 43 |
| 4.5.2.2. Apriori CS algoritması                    | 44 |
| 4.5.2.3. TBAR Algoritması                          | 45 |
| 4.5.2.4. TBAR algoritması (cache li versiyon)      | 46 |
| 4.5.3. Birliktelik Kuralları Kural gösterimi       | 48 |
| 4.5.4. Sınıflandırma Algoritmaları                 | 49 |
| 4.5.4.1. Karar Ağacı                               | 49 |
| 4.5.4.2. Karar ağacından üretilen kurallar         | 50 |
| 4.5.4.3. Karar ağacı parametreleri                 | 51 |
| 4.5.4.4. STAR                                      | 51 |
| 4.5.4.5. ARC                                       | 52 |
| 4.5.4.6. ARC Kural Gösterimi                       | 54 |
| 4.5.5. Gruplama Penceresi                          | 54 |
| 4.5.6. Nitelikleri seçme penceresi                 | 56 |
| 4.6. Uygulamada Kullanılan Diğer Pencereleler      | 58 |
| 4.6.1. JDBC Bağlantısı                             | 58 |
| 4.6.1.1. Oracle veritabanı bağlantı konfigürasyonu | 59 |
| 4.6.1.2. ODBC Köprüsü                              | 59 |
| 4.6.2. Tablo Seçme Penceresi                       | 59 |
| 4.6.3. SQL Sorgulama Penceresi                     | 60 |

4.6.4. Veritabanına Bilgi Alma (Import Data)

4.6.5. Veritabanından dışarıya bilgi transferi (Export Data)

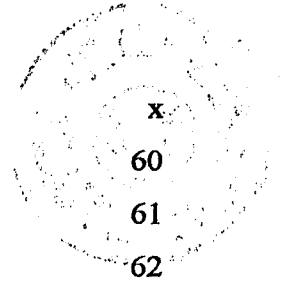
5. SONUÇ

KAYNAKLAR

ÖZGEÇMİŞ

EKLER

EK – 1 UYGULAMANIN KAYNAK KODU

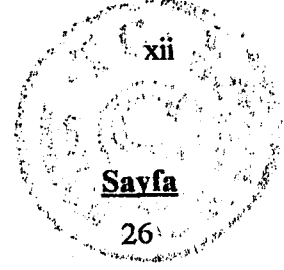


## SİMGELER VE KISALTMALAR DİZİNİ



|         |                                                          |
|---------|----------------------------------------------------------|
| VTBK    | : Veritabanından Bilgi Keşfi                             |
| KDD     | : Knowledge Discovery in Databases                       |
| CRM     | : Customer Relational Management                         |
| MIY     | : Müşteri İlişkileri Yönetimi                            |
| OLAP    | : Online Analytical Processing                           |
| DHP     | : Direct Hashing and Pruning                             |
| TBAR    | : Transaction Based Association Rule                     |
| TBAR CS | : Transaction Based Association Rule Collective Strength |
| ARM     | : Association Rule Mining                                |
| ARC     | : Association Rule Cache                                 |
| ART     | : Association Rule Tree                                  |
| TDIDT   | : Top-Down Induction on Decision Trees                   |
| CLS     | : Concept Learning System                                |
| CART    | : Classification And Regression Trees                    |
| AID     | : Automatic Interaction Detector                         |
| CS      | : Collective Strength                                    |
| URL     | : Uniform Resource Locator                               |
| TCP     | : Transport Control Protocol                             |
| IP      | : Internet Protocol                                      |
| JDBC    | : Java Database Connectivity                             |
| ODBC    | : Open Database Connectivity                             |
| DNS     | : Domain Name Service                                    |
| JVM     | : Java Virtual Machine                                   |
| TCP/IP  | : Transport Control Protocol / Internet Protocol         |
| SQL     | : Structured Query Language                              |
| OLAP    | : Online Analytical Processing                           |

## ŞEKİLLER DİZİNİ



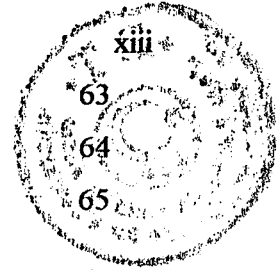
### Sekil

|                                                     |    |
|-----------------------------------------------------|----|
| 3.1. Kural Tabanlı Profil Belirleme prosesi         | 26 |
| 3.2. Sql Sorgulama Tabanlı Profil Belirleme prosesi | 26 |
| 3.3. Üretilen birliktelik kuralları                 | 29 |
| 3.4. Kural doğrulama prosesinin temel algoritması   | 30 |
| 3.5. Ürün Promosyonu İçin Müşteri Listesi           | 32 |
| 3.6. Ürün Promosyonu İçin Müşteri Listesi 2         | 32 |
| 3.7. Gruplama (Domain) Penceresi                    | 34 |
| 4.1. Uygulamanın Genel Yapısı                       | 36 |
| 4.2. Birliktelik Kuralları 1                        | 38 |
| 4.3. Promosyon Penceresi                            | 38 |
| 4.4. Karar Ağacı Penceresi 1                        | 39 |
| 4.5. Veritabanı Tabloları                           | 41 |
| 4.6. Promosyon Penceresi 2                          | 43 |
| 4.7. Apriori çalıştırma penceresi                   | 44 |
| 4.8. Apriori CS Algoritması                         | 45 |
| 4.9. TBAR Algoritması Parametreleri                 | 46 |
| 4.10. TBAR Cache Algoritması Parametreleri          | 47 |
| 4.11. Birliktelik Kuralları Kural Gösterimi         | 48 |
| 4.12. Karar Ağaçları                                | 49 |
| 4.13. Karar Ağacı Parametreleri                     | 51 |
| 4.14. STAR Metodolojisi Penceresi                   | 52 |
| 4.15. ARC Penceresi                                 | 53 |
| 4.16. ARC Algoritmasından Üretilen Kurallar         | 54 |
| 4.17. Gruplama (Domain) Penceresi                   | 55 |
| 4.18. Grup tanıtımı                                 | 56 |
| 4.19. Nitelikleri seçme penceresi                   | 57 |
| 4.20. JDBC Penceresi                                | 58 |
| 4.21. Tablo Seçme Penceresi                         | 59 |
| 4.22. SQL Sorgulama Penceresi                       | 60 |
| 4.23. Veritabanından Bilgi Alma Penceresi           | 61 |

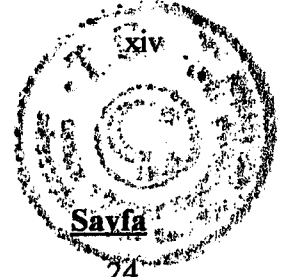
5.1. Karar Ağacından üretilen Promosyon

5.2. SQL Sorgulamasından üretilen Promosyon 1

5.3. SQL Sorgulamasından üretilen Promosyon 2



## TABLÖLAR DİZİNİ



### Tablo

- 3.1. Gerçeğe dayalı (Factual) Veri  
3.2. İşlemsel (Transactional) Veri  
5.1. Veritabanındaki Kayıt Durumu

24

24

62



# 1. GİRİŞ



Hangi sektörden olurlarsa olsunlar, işletmelerin müşterileriyle doğru iletişim sağlayabilmeleri, rekabet arttıkça daha da zorlaşır. Tüketici, her bir yönden gelen “Beni al, onu alma” veya “Benden al, ondan alma” sinyalleriyle çevrelenmiş durumdadır. Tercihini, kendisini en çok mutlu eden, güven veren, en az uğraştıran, en karlı olabilecek ürünlerden veya satış noktalarından kullanabilmek isteyecektir. Dolayısıyla doğru mesajı aldığı an eğilimlerini değiştirebilir. Müşterileri doğru analiz edebilmek, eğilimleri konusunda doğru sonuçlara varmak ve tüm bunları ortaya koyduktan sonra onlara ulaşmaya çalışmak hedeflenmelidir. MIY, bu noktada devreye girerek işletmenin tüm fonksiyonlarını belli bir düzen çerçevesinde birleştirerek müşteriye en iyi hizmeti ve en doyurucu ürünü satabilmeyi sağlayan kalıcı bir yönetim biçimini sağlar.

İyi iş yapabilmek; müşteri isteklerini anlamak ve ihtiyaçlarına cevap vermekle özdeşdir. İnternet, müşteri işlemleri ve davranışları ile ilgili daha fazla veri kaynağı sağlamaktadır. Globalleşme gibi pazar değişimleri ürün ve hizmetlerin sunulabileceği müşteri adayı sayısını artırmaktadır. Birebir pazarlama önemli bir strateji haline gelmiştir, bu da müşteri ve müşteri davranışları ile ilgili daha fazla veri toplanmasını beraberinde getirmektedir. Bu veriyi müşterileri daha iyi tanımak, müşteriler ile daha iyi ilişkiler kurma şirketin karlılığını artırma açısından yeni fırsatlar yakalama amaçlı olarak değerlendirilebilir.

Müşteri Sadakatini sadece %5 artırarak şirket karlılığını %60-100 artırmak mümkündür [5].

## 1.1. Uygulama

Bu tez çalışmasında geliştirilen uygulama (GYTE-Miner) ile MIY sistemlerinde toplanan veri ayrıştırılabilir, raporlama ve gruplama yapılabilir, verilerden ayrıntılı kurallar üretilebilir, çeşitli ürün ve ürün grupları için yapılacak promosyonu ve benzeri kampanyalar için en uygun müşteri listeleri çıkartılabilir. Uygulamanın MIY sistemleri ile entegre bir şekilde kullanımı müşteri ilişkilerinin daha iyi yönetimini ve şirketin karlılığının artırılmasını sağlayacaktır.

Müşterilerini daha iyi tanıyan şirketler onların spesifik ihtiyaçlarına cevap veren ürün ve hizmetler geliştirebilir, uyguladıkları yeni strateji ve programların sonuçlarını daha iyi değerlendirebilir ve başarılarının sürekliliğini sağlayabilirler. Analizlerde hem geçmişteki veriler hem de o anki veriler değerlendirildiğinde değişen durumları daha iyi takip edebilir, müşterilerin davranış değişimlerini izlenebilir.

Sonuç olarak uygulama bünyesindeki analitik gereçlerin kullanımı ve verinin ayrıştırımı ile “Müşteri profilim nedir?”, “Yeni geliştirilecek bir ürün için en iyi müşteri adaylarımız kimler?” gibi sorular cevaplanabilir. Daha iyi kararlar alabilmenin yolları; proseslerin geliştirilmesi, daha iyi veri toplama, daha iyi yöntemler kullanma, eldeki ham veriyi bilgiye dönüştürme olarak özetlenebilir.

Geliştirilen uygulamanın özellikleri hem “ne?” hem de “neden?” sorularına yanıt olması amaçlanmıştır. OLAP ve raporlama gereçleri “Bir müşteri ürünlerimiz için ne kadar harcama yapmış?” ve bunun gibi. ne? sorularına cevap verebilir, ileri analiz teknikler (veri ayrıştırma ile birlikte modelleme ve tahminsel teknikler) ise “Müşterilerimiz profil yapılarına göre bizim müşterimiz olarak kalabilecekleri sürede ne kadar harcama yapabilirler?” gibi daha detaylı ve rekabet avantajı yaratma açısından önemli sorulara cevap verebilir.

Raporlama ve OLAP gereçleri yalnız neler olduğu hakkında bilgi verirken GYTE-Miner ın sunduğu analitik yöntemler ve veri ayrıştırma çözümü ne olacağını şirketin geçmiş ve o anki bilgilerini değerlendirerek geleceğe ait durum tahminleri yapmaya olanak tanır. Şirket bünyesinde geçmiş dönemlerde toplanan verilerden GYTE-Miner bünyesindeki veri ayrıştırma ve/veya diğer analitik yöntemleri kullanılarak müşteri profili belirleme, kampanya yönetimi, müşteri sadakati belirleme, vb. amaçlı modeller geliştirilebilir ve geliştirilen modeller ile istenilen karar destek amaçlı bilgilere çok hızlı bir şekilde ulaşılabilir.

## 1.2. Veri Madenciliği

Veri madenciliği, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir. Temel olarak veri madenciliği, veri kümeleri arasındaki desenlerin ya da düzenin, verinin analizi ve yazılım tekniklerinin kullanılması ile

ilgilidir. Veriler arasındaki ilişkiyi, kuralları ve özellikleri belirlemekten bilgisayar sorumludur. Amaç, daha önceden fark edilmemiş veri desenlerini belirleyebilmektir.

Veri madenciliğinde, eldeki verilerden üstü kapalı, çok net olmayan, öncede bilinmeyen ancak potansiyel kullanışlı bilginin ortaya çıkarılması hedef alınır. Bu da kümeleme, veri özetleme, değişikliklerin analizi, sapmaların tespiti gibi belirli sayıda teknik yaklaşımları içerir [15].

Veri madenciliğini istatistiksel bir yöntemler serisi olarak görmek mümkün olabilir. Ancak veri madenciliği, geleneksel istatistikten birkaç yönde farklılık gösterir. Veri madenciliğinde amaç, kolaylıkla mantıksal kurallara ya da görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır. Bu bağlamda, veri madenciliği insan merkezlidir ve insan-bilgisayar arayüzü birleştirilir. Veri madenciliği modellerinde sürecin birçok noktasında interaktif olarak ilgili veri madenciliği parametrelerinin sisteme aktarılması gerekmektedir.

Doksanlı yılların sonlarına doğru veri tabanı sistemlerinin artan kullanımı ve hacimlerindeki bu olağanüstü artış, organizasyonları elde toplanan bu verilerden nasıl faydalanılabileceği problemi ile karşı karşıya bırakmıştır. Geleneksel sorgu (query) veya raporlama araçlarının veri yığınları karşısında yetersiz kalması, Veri Madenciliği (Data Mining) ve Veri tabanlarında Bilgi Keşfi-VTBK (Knowledge Discovery in Databases) alanlarında yeni araştırmalar yapılmasını gerektirmektedir..

Çeşitli veri kaynaklarından verilerin toplanması ile başlayan VTBK süreci, toplanan verilerin analiz için uygun hale getirilmesi aşaması ile devam eder [3]. Ancak veri ambarına (Data Warehouse) sahip olan kuruluşlarda, gerekli verilerin Data Mart olarak isimlendirilen işleve özel veri tabanlarına aktarılması ile doğrudan veri madenciliği işlemlerine başlanabilmesi de mümkündür [16].

Piatetsky-Shapiro veri madenciliğini, verilerden daha önceden bilinmeyen, zımnî, muhtemelen faydalı enformasyonun monoton olmayan bir süreçte çıkartılması işlemi olarak tanımlamaktadır. Bu süreç kümeleme (clustering), veri özetleme (data summarization), sınıflama kurallarının (classification rules) öğrenilmesi, bağımlılık ağlarının (dependency networks) bulunması, değişikliklerin analizi (analysing changes) ve anomali tespiti (detecting anomaly) gibi farklı bir çok teknik yaklaşımı kapsamaktadır [6].

Veri madenciliği astronomi, biyoloji, finans, pazarlama, sigorta, tıp ve bir çok başka dalda uygulanmaktadır. Son 20 yıldır Amerika Birleşik Devletleri'nde çeşitli veri madenciliği algoritmalarının gizli dinlemeden, vergi kaçakçılıklarının ortaya çıkartılmasına kadar çeşitli uygulamalarda kullanıldığı bilinmektedir [6]. Günümüzde veri madenciliği teknikleri özellikle işletmelerde çeşitli alanlarda kullanımı daha da yaygınlaşmıştır. Bunların başlıcaları ilgili alanlara göre aşağıda özetlenmiştir [9].

### **Pazarlama**

- Müşterilerin satın alma örüntülerinin belirlenmesi,
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması,
- Posta kampanyalarında cevap verme oranının artırılması,
- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması,
- Pazar sepeti analizi (*Market Basket Analysis*)
- Müşteri ilişkileri yönetimi (*Customer Relationship Management*)
- Müşteri değerlendirme (*Customer Value Analysis*)
- Satış tahmini (*Sales Forecasting*).

### **Bankacılık**

- Farklı finansal göstergeler arasında gizli korelasyonların bulunması,
- Kredi kartı dolandırıcılıklarının tespiti,
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi,
- Kredi taleplerinin değerlendirilmesi.

### **Sigortacılık**

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

Yapılan çalışmada daha çok satış ve pazarlama alanında hizmet yapan sektörlerin müşteri-ürün analizleri yapılmıştır. Müşterilerini daha iyi tanıyarak kar oranlarının nasıl arttırılacağı problemi üzerine gidilmiştir.



### 1.3. MIY (Müşteri ilişkileri Yönetimi)

Veri madenciliği metodlarının ağırlıklı olarak kullanıldığı alanlardan biri Müşteri ilişkileri yönetimi (MIY) dir [16].

Müşteri ilişkileri yönetimi ingilizce Customer Relationship Management kelimelerinin baş harflerinden oluşturulan CRM kısaltması ile daha sık kullanılmaktadır. MIY müşterinin firmaya olan bağlılığını , tatminini ve firmanın karlılığını arttırmaya yönelik günümüzde elektronik ortamda çalışan bir iş stratejisi ve biçimidir [15].

ABD'de 2000 yılı itibarı ile pazarın boyutu 3.72 milyar\$'dır. Bunun %72'sini yani 2.7 milyar\$'ını oluşturan sektörler;

- Bankacılık ve Finans, Sigorta ( %22 )
- Telekomünikasyon-kablolu,kablosuz- ( %19 )
- Yüksek Teknoloji, elektronik ( %16 )
- Üretim, otomotiv v.b. ( %15 )

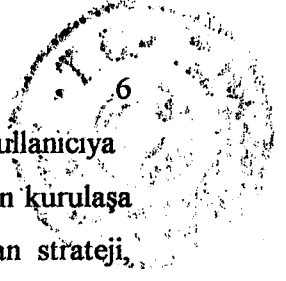
Müşteri ilişkileri yönetimi amaçlı olarak geliştirilmiş birçok Müşteri İlişkileri Yönetimi (MIY) sistemi bulunmaktadır. Bunlardan bazıları satış otomasyon sistemleri, pazarlama kampanyaları yönetim sistemleri, çağrı merkezi (call center) hizmetleri, web dir. Bu sistemler operasyonları daha verimli hale getirmede önemlidirler, ve müşteri satış bilgilerini, müşterilerin pazarlama programlarına verdiği yanıtları, destek amaçlı arama bilgileri ve web kullanımlarına dair bilgileri içeren çok sayıda veri toplanmasını sağlarlar. Bu veriler genelde veri tabanlarında depolanırlar ve veri ambarına gönderilebilirler [3].

Müşteri ile ilgili incelemelerde kullanılan en basit yöntem yönetime sunulan raporlardır, fakat şirketin karını artırmak için müşteri verilerinin daha detaylı incelenmesi ve anlaşılması gerekir. Kullanılan MIY sistemine güçlü analitik tekniklerin entegrasyonu MIY sistemine yapılan yatırımdan en üst düzeyde fayda sağlanması, karlılığın artırılması açısından önemlidir.

### 1.4. Veri Madenciliği ve MIY

Müşteri İlişkileri Yönetimi sistemleri temelde veri madenciliği yöntemlerini

kullanmaktadır. MIY sistemleri kullanan kuruluşlar daha çok sistemin kullanıcıya ve müşteriye bakan tarafları ile ilgilenmektedir. MIY sistemlerinin, kullanan kurulaşa da bağılı olmakla birlikte birçok bileşeni vardır. Bu bileşenler uygulanan strateji, insan kaynakları, arama merkezi, uygulama veya uygulamalar vs. olabilir. Uygulamalar bu bileşenler içerisinde en önemli unsurlardır. Çünkü alt seviyede veri toplama, verilerin analiz edilmesi, kuralların oluşturulması, ilgili kurallardan profillerin oluşturulması, tahmin edici yöntemler ve bunun gibi teknik anlamdaki çoğu iş veri madenciliğinin kapsamına girmektedir. Bu nedenle MIY sürecinin yapıtaşlarını oluşturan uygulamalar veri madenciliği yöntemlerini kullanmak zorundadır.



## 2. VERİ MADENCİLİĞİ YÖNTEMLERİ



Veri madenciliği büyük miktarda veri inceleme amacı üzerine kurulmuş olduğu için veri tabanları ile yakından ilişkilidir. Gerekli verinin hızla ulaşılabilecek şekilde amaca uygun bir şekilde saklanması ve gerektiğinde hızla ulaşılabilmesi gerekir.

Veri ambarında veri oluşturulduktan sonra bu verinin elle veya gözle analizi yapılabilir. Bunun için OLAP (*Online Analytical Processing*) programları kullanılır. Bu programlar veriye her boyutu veride bir alana karşılık gelen çokboyutlu bir küp olarak bakmayı ve incelemeyi sağlar. Böylece boyut bazında gruplama, boyutlar arasındaki korelasyonları inceleme ve sonuçları grafik veya rapor olarak sunma olanağı sağlar.

Veri madenciliği yöntemlerinde sıklıkla kullanılan iki temel parametre *güven* (*confidence*) ve *destek* (*support*) parametreleridir.

### 2.1. Güven (Confidence) Parametresi

Bir veri madenciliği kuralı  $X \Rightarrow Y$  şeklinde bir kuraldır öyleki X ve Y boş küme kesişimli eleman gruplarıdır.  $X \Rightarrow Y$  birliktelik kuralının güvenliği X ve Y yi içeren işlemlerin Y yi içerenlere oranıdır [8].

C güven, X sınıflandırma alanı, Y sınıflandırma alanına karşılık düşen değer olmak üzere;

$$C = ((X \cap Y) / Y) * 100 \quad (2.1)$$

### 2.2. Destek (Support) Parametresi

$X \Rightarrow Y$  şeklinde bir kuralın destek parametresi veritabanında X ve Y yi içeren hareketlerin veritabanındaki tüm işlem sayısına olan oranıdır şeklinde tanımlanabilir [8]. Başka bir deyişle destek parametresi veritabanındaki bütün transactionların bir kesiri (alt kümesi) dir denebilir.

S destek, X sınıflandırma alanı, T toplam transaction sayısı olmak üzere



$$S = ((X \cap Y)/T) * 100 \quad (2.2)$$

### 2.3. Veri Madenciliği Modelleri

Veri madenciliğinde kullanılan modeller, tahmin edici (*predictive*) ve tanımlayıcı (*descriptive*) olmak üzere iki ana başlık altında incelenmektedir. 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. Örneğin bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır [15].

Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılacak mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır. X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu aileler ile, çocuğu olmayan ve geliri X/Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerlik gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir. Veri madenciliği modellerini gördükleri işlevlere göre üç ana başlık altında incelemek mümkündür [7].

- Sınıflama (*Classification*) ve Regresyon (*Regression*),
- Kümeleme (*Clustering*),
- Birliktelik Kuralları (*Association Rules*) ve Ardışık Zamanlı Örüntüler (*Sequential Patterns*),

Sınıflama ve regresyon modelleri tahmin edici, kümeleme, birliktelik kuralları ve ardışık zamanlı örüntü modelleri tanımlayıcı modellerdir [12].

### 2.3.1. Sınıflama ve Regresyon Modelleri

Verileri en az bir veya daha fazla sınıflama kategorisine göre sınıflandırmak ve bu doğrultuda sınıflandırma kuralları üretmek için kullanılır.

Mevcut verilerden hareket ederek geleceğin tahmin edilmesinde faydalanan ve veri madenciliği teknikleri içerisinde en yaygın kullanıma sahip olan sınıflama ve regresyon modelleri arasındaki temel fark, tahmin edilen bağımlı değişkenin kategorik veya süreklilik gösteren bir değere sahip olmasıdır. Ancak çok terimli lojistik regresyon (*multinomial logistic regression*) gibi kategorik değerlerin tahmin edilmesine olanak sağlayan tekniklerle, her iki model giderek birbirine yaklaşmakta ve bunun bir sonucu olarak aynı tekniklerden yararlanılması mümkün olmaktadır. Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler şunlardır; [9].

- Karar Ağaçları (*Decision Trees*),
- Yapay Sinir Ağları (*Artificial Neural Networks*),
- Genetik Algoritmalar (*Genetic Algorithms*),
- K-En Yakın Komşu (*K-Nearest Neighbor*),
- Bellek Temelli Nedenleme (*Memory Based Reasoning*),
- Naive-Bayes,
- Lojistik Regresyon (*Logistic Regression*).

### 2.3.2. Kümeleme Modelleri

Kümeleme modellerinde amaç küme üyelerinin birbirlerine çok benzediği, ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veri tabanındaki kayıtların bu farklı kümelere bölünmesidir. Başlangıç aşamasında veri tabanındaki kayıtların hangi kümelere ayrılacağı veya kümelemenin hangi değişken özelliklerine göre yapılacağı bilinmemekte, konunun uzmanı olan bir kişi tarafından kümelerin neler olacağı tahmin edilmektedir [9].

### 2.3.3. Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler

Bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal

veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi, müşteriye daha fazla ürünün satılmasını sağlama yollarından biridir. Satın alma eğilimlerinin tanımlanmasını sağlayan birliktelik kuralları ve ardışık zamanlı örüntüler, pazarlama amaçlı olarak pazar sepeti analizi (*Market Basket Analysis*) adı altında veri madenciliğinde yaygın olarak kullanılmaktadır [11]. Bununla birlikte bu teknikler, tıp, finans ve farklı olayların birbirleri ile ilişkili olduğunun belirlenmesi sonucunda değerli bilgi kazanımının söz konusu olduğu ortamlarda da önem taşımaktadır.

Birliktelik kuralları aşağıda sunulan örneklerde görüldüğü gibi eş zamanlı olarak gerçekleşen ilişkilerin tanımlanmasında kullanılır.

- *Müşteriler kola satın aldığı anda, % 75 ihtimalle kuruyemiş de alırlar,*
- *Düşük yağlı peynir ve yağsız yoğurt alan müşteriler, %85 ihtimalle diet süt de satın alırlar.*

Ardışık zamanlı örüntüler ise aşağıda sunulan örneklerde görüldüğü gibi birbirleri ile ilişkisi olan ancak birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır.

- *X ameliyatı yapıldığında, 15 gün içinde % 45 ihtimalle Y enfeksiyonu oluşacaktır,*
- *İMKB endeksi düşerken A hisse senedinin değeri % 15'den daha fazla artacak olursa, üç iş günü içerisinde B hisse senedinin değeri % 60 ihtimalle artacaktır,*
- *Çekiç satın alan bir müşteri, ilk üç ay içerisinde % 15, bu dönemi izleyen üç ay içerisinde % 10 ihtimalle çivi satın alacaktır.*

Ticari ve market uygulamalarının kullandığı büyük boyutlardaki işlemsel veritabanları birliktelik kuralları oluşturmak için çok uygundur [15].

Örneğin barkodlar alınan malların veritabanında düzenli olarak saklanmasını kolaylaştırır ve kuralları oluşturacak bilginin veritabanından çekilmesinde büyük avantaj sağlar. Bu tür veritabanlarında bir işlem, tarih ve satın alınan ürünü içerir.

Birliktelik kurallarının formülize edilmesi aşağıda anlatılmıştır.

I bütün elemanların kümesi olsun, TID tekil tanımlayıcı T işlemi ise I'nın bir alt kümesi olsun. İşlem X elemanlarının bir kümesini içerir, öyleki eğer eleman grupları T, X i içerir ise şeklinde olur.

En sık kullanılan birliktelik kuralları algoritmaları şunlardır;

- Apriori
- Apriori CS
- TBAR (Tree Based Association Rule)
- TBAR (cache version)

#### 2.3.3.1. Apriori

Apriori, Agrawal ve Skirant tarafından 1994 yılında geliştirilmiştir. Apriori verilen bir ilişkisel tablodan birliktelik kuralları oluşturmak için alternatiflerden biridir [8].

#### 2.3.3.2. TBAR (Transaction Based Association Rule)

TBAR aynı yıllarda apriori algoritmasının optimize edilmiş bir versiyonudur.

#### 2.3.3.3. Apriori CS

Apriori algoritmasının eleman gruplarında kollektif güç (Collective Strength) parametresinin kullanılmış halidir.

#### 2.3.3.4. TBAR Cache

TBAR Cache I/O yükünden kurtulmak için bütün dataseti belleğe alıp işlemleri burada yapar.

#### 2.3.4. Sınıflandırma Algoritmaları

Öğrenme kümesi (giriş verileri) modellemeye çalıştığımız sınıf örnekleri içerir. Elde ettiğimiz model yeni örnek verileri sınıflandırma veya mevcut veriyi daha iyi yorumlayabilmek için kullanılabilir [13].

En çok kullanılan sınıflandırma modelleri:

- Karar Ağaçları (Decision Trees )
- STAR Metodolojisi (STAR Methodology)
- Birlikte Kural Tabanlı Modeller (Association-Rule Based Models )

Modeller aşağıda açıklanmıştır.

#### 2.3.4.1. Karar Ağaçları

Sınıflandırma veya tanımlama ağaçları olarak da bilinen karar ağaçları en popüler veri madenciliği ve profil belirleme metodudur.

Karar ağaçları verilen bir örneği sınıflandırmak, ağacın başlangıcından başlanarak bir dal üzerinde ilerlemek ve en alt yaprağa erişinceye kadar bu yolu takip etmek sureti ile yapılır [9]. Böylece ilgili sorunun cevabı ağacın her dalından geçtikçe kademeli ve seçenekli olarak verilmiş olur.

Karar ağaçlarının döngüsel olarak yukardan-aşağıya yaklaşımı ile üretilmesine TDIDT (Top-Down Induction on Decision Trees) adı verilmektedir. TDIDT algoritma ailesi CLS (Concept Learning System), CART (Classification And Regression Trees) gibi klasik algoritmaları da içerir [5].

TDIDT algoritmaları genellikle veride herhangi bir gürültü olmadığını varsayar ve verinin en mükemmel tanımını elde etmeye çalışırlar. Bu çoğu zaman gerçek problemlerde, özellikle çok fazla kesinlik istenmeyen durumlarda yada verinin gürültülü olduğu şartlarda sorun oluşturabilir. Bu durum fazlalıkları atma yada budama olarak tabir edilen tekniklerin kullanılma zorunluluğunu ortaya çıkarmıştır. ASSISTANT ve C4.5 algoritmalarında kullanılan bu teknikler taşmayı önlemek amacı ile kullanılmıştır. Budama parametresinden alınan değere göre daha düşük seviyedeki dallar budanır ve kullanıcıya gösterilmez [9].

Tahmin edici ve tanımlayıcı özelliklere sahip olan karar ağaçları, veri madenciliğinde

- Kuruluşlarının ucuz olması,
- Yorumlanmalarının kolay olması,
- Veri tabanı sistemleri ile kolayca entegre edilebilmeleri,
- Güvenilirliklerinin daha iyi olması

nedenleri ile sınıflama modelleri içerisinde en yaygın kullanıma sahiptir. Karar ağacı

temelli analizlerin yaygın olarak kullanıldığı sahalara aşağıda listelenmiştir [13].

- Belirli bir sınıfın muhtemel üyesi olacak elemanların belirlenmesi (*segmentation*),
- Çeşitli vakaların yüksek, orta, düşük risk grupları gibi çeşitli kategorilere ayrılması (*stratification*),
- Gelecekteki olayların tahmin edilebilmesi için kurallar oluşturulması,
- Parametrik modellerin kurulmasında kullanılmak üzere çok miktardaki değişken ve veri kümesinden faydalı olacakların seçilmesi,
- Sadece belirli alt gruplara özgü olan ilişkilerin tanımlanması,
- Kategorilerin birleştirilmesi ve sürekli değişkenlerin kesikliye dönüştürülmesi.

Karar ağacı temelli tipik uygulamalar ise aşağıda verilmiştir [13].

- Hangi demografik grupların mektupla yapılan pazarlama uygulamalarında yüksek cevaplama oranına sahip olduğunun belirlenmesi (*Direct Mail*),
- Bireylerin kredi geçmişlerini kullanarak kredi kararlarının verilmesi (*Credit scoring*),
- Geçmişte işletmeye en faydalı olan bireylerin özelliklerini kullanarak işe alma süreçlerinin belirlenmesi,
- Tıbbi gözlem verilerinden yararlanarak en etkin kararların verilmesi,
- Hangi değişkenlerin satışları etkilediğinin belirlenmesi,
- Üretim verilerini inceleyerek ürün hatalarına yol açan değişkenlerin belirlenmesi.

Sosyal ve ekonomik olayları daha güvenilir bir şekilde gösterebilmek için standart istatistik tekniklerin dışında yeni analiz tekniklerinin geliştirilmesi ile ilgilenen Morgan ve Sonquist tarafından University of Michigan'da 1970'li yılların başlarında kullanıma alınan *Automatic Interaction Detector* – AID karar ağacı temelli ilk algoritma ve yazılımdır [9]. AID tekniği en kuvvetli ve en iyi tahmini gerçekleştirebilmek için bağımlı ve bağımsız değişkenler arasındaki mümkün bütün ilişkilerin incelenmesine dayanmaktadır [9].

#### 2.3.4.2. STAR Metodu

Daha büyük karar ağacı, daha az anlaşılabilirlik. Sınıflandırma problemi çok karmaşık olduğu durumlarda uzman karar ağacının yapısını anlamakta güçlük çekebilir (budama işleminden sonra bile) [4].

Shapiro daha anlaşılır bir model elde etmek amacı ile, kompleks bir karar ağacının daha küçük ağaçlara hiyerarşik olarak parçalanmasını önermiştir (*yapısal indüksiyon*) [9]. Böylece karar ağacının kurallar kümesi olarak ele alınması daha da kolaylaşmaktadır.

Karar ağaçlarına ihtiyaç duymaksızın kuralları direct olarak üreten farklı metodlar da vardır. Bu teknikler daha az etkin olan (*beam search*) gibi sezgisel arama teknikleri kullanır.

Michalski ve takım arkadaşları STAR metodolojisini keşfettiler. STAR metodolojisi lojik işlemlere dayalı bir grup ardışık tümevarımsal öğrenme tekniği kullanır. INDUCE, AQ ve CN2 bu metodolojiyi örnekleyen seçeneklerdir [11].

#### 2.3.4.3. Birliktelik Kural Tabanlı Sınıflandırma

Gerçek hayatta problemler genellikle veriyi tam ve kusursuz olarak sınıflandırmaya uygun olmaz. Parçalı modeller farklı sınıflar içerisinde düzenli kuralları karakterize etmekten daha kullanışlı olabilir. Çok fazla sayıda nitelik varsa komple bir sınıflandırma modeli oluşturmak mümkün olmayabilir. Çünkü niteliklerden bazılarının değeri bilinmiyor olabilir yada çok büyük bir veri kümesinden çekmek zorunda kalabiliriz. Parçalı sınıflandırma ortak kuralları asıl modele bağlı kalmaksızın ayrı sınıflar içerisinde keşfetmeye çalışır [9].

Birliktelik kural madenciliği klasik sınıflandırma tekniklerinin çok efektif olmadığı parçalı sınıflandırma problemlerini çözmek için uygundur. Karar ağaçları eksik verinin üstesinden gelme konusunda kısmen başarılıdır (öğrenme kümesinde bilinmeyen nitelik değerleri) fakat bazı nitelikler başka nitelikler tarafından belirlendiği durumlarda bazı problemler ortaya çıkmaktadır.

Birliktelik kurallarını kullanarak parçalı sınıflandırma problemini aşağıdaki 2 yaklaşımla ele alabiliriz.



### 1. Öğrenme kümesini bölmek

Her bir C sınıfının bütün öğrenme kümeleri için dağıtık A eleman kümeleri bulunur. Sonuç desenler bulunan diğer desenler ile karşılaştırılır. Sonra, dağıtık desenler istenilen desenler olarak elde edilmiş olur. (chi-squared test yöntemi) [13].

$$r(A \Rightarrow C) = P(C|A) / P(C|\neg A) \quad (2.3)$$

Bağıl risk (Relative risk) olarak adlandırılan bu değer yüksek olduğunda  $A \Rightarrow C$  kuralının doğru (ilgilenilen kural) olma ihtimali de doğru orantılı olarak yüksek olmaktadır.

### 2. Sınıfın farklı bir nitelik gibi değerlendirilmesi

Birliktelik kural madenciliği bütün verikümesine uygulanır. Böylece Apriori ve DHP gibi klasik birliktelik kuralları algoritmaları kullanılabilir. Birliktelik kurallarının sonuç kümesi elde edilmiş olacaktır. Elde edilen sonuç kümesi muhtemelen çok büyük olacaktır [9].

$A \Rightarrow C$  birliktelik kuralının *bağıl riski* aşağıda eşitlikte verilmiştir.

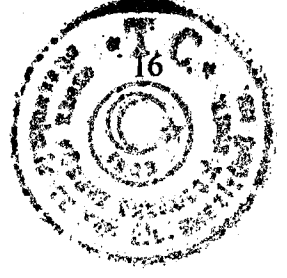
$$\begin{aligned} r(A \Rightarrow C) &= P(C|A) / P(C|\neg A) \\ &= [ [\text{Support}(C \cup A) / \text{Support}(A)] / \\ &\quad [ [\text{Support}(C) - \text{Support}(C \cup A)] / 1 - \text{Support}(A)] ] \end{aligned} \quad (2.4)$$

Gerekli potansiyel bilgiyi kaybetmemek için minimum destek eşik değeri düşük olmalıdır. Bazı sektörlerde bu tür bir bilgi kaybına tahammül edilememektedir ve bütün dağıtık eleman kümelerinin tam olarak hesaplanması gerekir [13].

Her iki strateji de benzer sonuçlar elde edilir, fakat ilki daha çok kullanılmaktadır.

## 2.4. Veri Tabanlarında Bilgi Keşfi Süreci

Veri tabanlarında bilgi keşfi sürecinde izlenmesi gereken temel aşamalar şunlardır.



- Hedefin Tanımlanması,
- Verilerin Hazırlanması,
- Modelin Kurulması ve Değerlendirilmesi,
- Modelin Kullanılması,
- Modelin İzlenmesi

Ne kadar etkin olursa olsun hiç bir veri madenciliği algoritmasının, üzerinde inceleme yapılan işin ve verilerin özelliklerinin bilinmemesi durumunda fayda sağlaması mümkün değildir. Bu nedenle yukarda tanımlanan tüm aşamalardan önce, iş ve veri özelliklerinin öğrenilmesi / anlaşılması başarının ilk şartı olacaktır [16].

#### **2.4.1. Hedefin Tanımlanması**

Veri madenciliği çalışmalarında başarılı olmanın ilk şartı, uygulamanın hangi işletme amacı için yapılacağını açık bir şekilde tanımlanmasıdır. İlgili işletme amacı işletme problemi üzerine odaklanmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği tanımlanmalıdır. Ayrıca yanlış tahminlerde katlanılacak olan maliyetlere ve doğru tahminlerde kazanılacak faydalara ilişkin tahminlere de bu aşamada yer verilmelidir [16].

#### **2.4.2. Verilerin Hazırlanması**

Modelin kurulması aşamasında ortaya çıkacak sorunlar, bu aşamaya sık sık geri dönülmesine ve verilerin yeniden düzenlenmesine neden olacaktır. Bu durum verilerin hazırlanması ve modelin kurulması aşamaları için, bir analistin veri keşfi sürecinin toplamı içerisinde enerji ve zamanının % 50 - % 85'ini harcamasına neden olmaktadır [14]. Verilerin hazırlanması aşaması kendi içerisinde toplama, değer biçme, birleştirme ve temizleme, seçme ve dönüştürme adımlarından meydana gelmektedir.

##### **2.4.2.1. Veri Toplama (Collection)**

Tanımlanan problem için gerekli olduğu düşünülen verilerin ve bu verilerin toplanacağı veri kaynaklarının belirlenmesi adımdır. Verilerin toplanmasında kuruluşun kendi veri kaynaklarının dışında, nüfus sayımı, hava durumu, merkez

bankası kara listesi gibi veri tabanlarından veya veri pazarlayan kuruluşların veri tabanlarından faydalanılabilir [3].



#### 2.4.2.2. Değer Bıçme (Assessment)

Veri madenciliğinde kullanılacak verilerin farklı kaynaklardan toplanması, doğal olarak veri uyumsuzluklarına neden olacaktır. Bu uyumsuzlukların başlıcaları farklı zamanlara ait olmaları, kodlama farklılıkları (örneğin bir veri tabanında cinsiyet özelliğinin e/k, diğer bir veri tabanında 0/1 olarak kodlanması), farklı ölçü birimleridir. Ayrıca verilerin nasıl, nerede ve hangi koşullar altında toplandığı da önem taşımaktadır. Bu nedenlerle, iyi sonuç alınacak modeller ancak iyi verilerin üzerine kurulabileceği için, toplanan verilerin ne ölçüde uyumlu oldukları bu adımda incelenerek değerlendirilmelidir [2].

#### 2.4.2.3. Birleştirme ve Temizleme (Consolidation and Cleaning)

Bu adımda farklı kaynaklardan toplanan verilerde bulunan ve bir önceki adımda belirlenen sorunlar mümkün olduğu ölçüde giderilerek veriler tek bir veri tabanında toplanır. Ancak basit yöntemlerle ve baştan savma olarak yapılacak sorun giderme işlemlerinin, ileriki aşamalarda daha büyük sorunların kaynağı olacağı unutulmamalıdır [4].

#### 2.4.2.4. Seçim (Selection)

Bu adımda kurulacak modele bağlı olarak veri seçimi yapılır. Örneğin tahmin edici bir model için, bu adım bağımlı ve bağımsız değişkenlerin ve modelin eğitiminde kullanılacak veri kümesinin seçilmesi anlamını taşımaktadır. Sıra numarası, kimlik numarası gibi anlamlı olmayan ve diğer değişkenlerin modeldeki ağırlığının azalmasına da neden olabilecek değişkenlerin modele girmemesi gerekmektedir. Bazı veri madenciliği algoritmaları konu ile ilgisi olmayan bu tip değişkenleri otomatik olarak elese de, pratikte bu işlemin kullanılan yazılıma bırakılmaması daha akılcı olacaktır [7].

Verilerin görselleştirilmesine olanak sağlayan grafik araçlar ve bunların sunduğu ilişkiler, bağımsız değişkenlerin seçilmesinde önemli yararlar sağlayabilir [14]. Genellikle yanlış veri girişinden veya bir kereye özgü bir olayın gerçekleşmesinden kaynaklanan verilerin (*Outlier*), önemli bir uyarıcı bilgi içerip

içermediği kontrol edildikten sonra veri kümesinden atılması tercih edilir. Modelde kullanılan veri tabanının çok büyük olması durumunda tesadüflüğü bozmayacak şekilde örnekleme yapılması uygun olabilir. Hesaplama olanakları ne kadar gelişmiş olursa olsun, çok büyük veri tabanları üzerinde çok sayıda modelin denenmesi zaman kısıtı nedeni ile mümkün olamamaktadır. Bu nedenle tüm veri tabanını kullanarak bir kaç model denemek yerine, tesadüfi olarak örneklenmiş bir veri tabanı parçası üzerinde bir çok modelin denenmesi ve bunlar arasından en güvenilir ve güçlü modelin seçilmesi daha uygun olacaktır [15].

#### 2.4.2.5. Dönüştürme (Transformation)

Kredi riskinin tahmini için geliştirilen bir modelde, borç/gelir gibi önceden hesaplanmış bir oran yerine, ayrı ayrı borç ve gelir verilerinin kullanılması tercih edilebilir. Ayrıca modelde kullanılan algoritma, verilerin gösteriminde önemli rol oynayacaktır. Örneğin bir uygulamada bir yapay sinir ağı algoritmasının kullanılması durumunda kategorik değişken değerlerinin evet/hayır olması; bir karar ağacı algoritmasının kullanılması durumunda ise örneğin gelir değişken değerlerinin yüksek/orta/düşük olarak gruplanmış olması modelin etkinliğini artıracaktır [3].

#### 2.4.3. Modelin Kurulması ve Değerlendirilmesi

Veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele varılncaya kadar yinelenen bir süreçtir. Tanımlanan problem için en uygun modelin bulunabilmesi, olabildiğince çok sayıda modelin kurularak denenmesi ile mümkündür.

Model kuruluş süreci denetimli (*Supervised*) ve denetimsiz (*Unsupervised*) öğrenimin kullanıldığı modellere göre farklılık göstermektedir. Örnekten öğrenme olarak da isimlendirilen denetimli öğrenimde, bir denetçi tarafından ilgili sınıflar önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunması ve bu özelliklerin kural cümleleri ile ifade edilmesidir. Öğrenme süreci tamamlandığında, tanımlanan kural cümleleri verilen yeni örneklerle uygulanır ve yeni örneklerin hangi sınıfa ait olduğu kurulan model tarafından belirlenir [14].

Denetimsiz öğrenmede, kümeleme analizinde olduğu gibi ilgili örneklerin

gözenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır.

Denetimli öğrenimde seçilen algoritmaya uygun olarak ilgili veriler hazırlandıktan sonra, ilk aşamada verinin bir kısmı modelin öğrenimi, diğer kısmı ise modelin geçerliliğinin test edilmesi için ayrılır. Modelin öğrenimi öğrenim kümesi kullanılarak gerçekleştirildikten sonra, test kümesi ile modelin doğruluk derecesi (*Accuracy*) belirlenir.

Bir modelin doğruluğunun test edilmesinde kullanılan en basit yöntem basit geçerlilik (*Simple Validation*) testidir. Bu yöntemde tipik olarak verilerin % 5 ile % 33 arasındaki bir kısmı test verileri olarak ayrılır ve kalan kısım üzerinde modelin öğrenimi gerçekleştirildikten sonra, bu veriler üzerinde test işlemi yapılır [14]. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının, tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile ise doğruluk oranı hesaplanır.

$$(\text{Doğruluk Oranı} = 1 - \text{Hata Oranı}) \quad (2.5)$$

Sınırlı miktarda veriye sahip olunulması durumunda, kullanılabilecek diğer bir yöntem çapraz geçerlilik (*Cross Validation*) testidir. Bu yöntemde veri kümesi tesadüfi olarak iki eşit parçaya ayrılır. İlk aşamada a parçası üzerinde model eğitimi ve b parçası üzerinde test işlemi; ikinci aşamada ise b parçası üzerinde model eğitimi ve a parçası üzerinde test işlemi yapılarak elde edilen hata oranlarının ortalaması kullanılır. Bir kaç bin veya daha az satırdan meydana gelen küçük veri tabanlarında, verilerin n gruba ayrıldığı n katlı çapraz geçerlilik (*N-Fold Cross Validation*) testi tercih edilebilir [1].

#### 2.4.4. Modelin Kullanılması

Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama olabileceği gibi, bir başka uygulamanın alt parçası olarak kullanılabilir. Kurulan modeller risk analizi, kredi değerlendirme, dolandırıcılık tespiti gibi işletme uygulamalarında doğrudan kullanılabileceği gibi, promosyon planlaması simülasyonuna entegre edilebilir veya tahmin edilen envanter düzeyleri yeniden sipariş noktasının altına düştüğünde, otomatik olarak sipariş verilmesini sağlayacak bir uygulamanın içine

gömülebilir.

#### 2.4.5. Modelin İzlenmesi

Zaman içerisinde bütün sistemlerin özelliklerinde ve dolayısıyla ürettikleri verilerde ortaya çıkan değişiklikler, kurulan modellerin sürekli olarak izlenmesini ve gerekiyorsa yeniden düzenlenmesini gerektirecektir. Tahmin edilen ve gözlenen değişkenler arasındaki farklılığı gösteren grafikler model sonuçlarının izlenmesinde kullanılan yararlı bir yöntemdir.

### 2.5. Kural Keşfi

Alışkanlıkları belirlemenin sezgisel ve betimsel yolları olmasının yanı sıra, sıklıkla veri madenciliğinde, uzman sistemlerde, lojik programlamada ve başka birçok alanda kullanılan iyi tasarlanmış bağlayıcı kurallar da bize yardımcı olabilir. Bununla birlikte önceki bölümlerde anlatılan başta birliktelik ve sınıflandırma kuralları olmak üzere birçok kural keşfetme algoritması kural keşfinde kullanılmaktadır [10].

#### 2.5.1. Kural Doğrulama (Rule Validation)

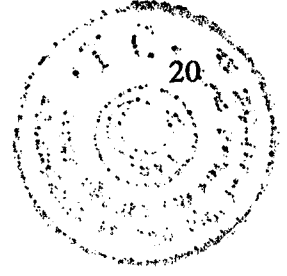
Veri madenciliği metodları çoğu zaman çok fazla sayıda kural üretirler. Bu kuralları çoğu istatistiksel olarak kabul edilebilir olsa da birçoğu gereksiz, sıradan, sahte veya uygulama için yararlı olmayan bilgiler içerir. Bu sebeple Kural doğrulama önemli bir gerekliliktir [10].

##### 2.5.1.1. Benzerlik Tabanlı Kural Gruplama (Similarity Based Rule Grouping)

Bu uzmanın belirlediği benzerlik kriterlerine göre benzer kuralları bir gruba koyar. Sonuç olarak uzman kuralları gruplar bazında değerlendirme şansına sahip olmuş olur [1].

Örneğin, eleman yapısı benzerlik durumuna göre, aynı yapıya sahip olan bütün kurallar (eleman değerleri ve istatistiksel parametreler gözardı edildiğinde) benzerdir.

Özellik yapısı itibari ile gruplama, benzerlik tabanlı kural gruplama





yöntemlerinin sadece biridir. Buna benzer birçok grupta yöntemleri uygulanabilir.

#### **2.5.1.2. Şablon tabanlı kural filtreleme (Template-based rule filtering)**

Bu operatör kuralları ekspertin belirlediği kural şablonuna göre filtreler. Ekspert kabul edilecek ve reddedilecek kural şablonunu önceden belirler. Doğal olarak üretilen kurallar içerisinde kabul durumuna karşılık düşen kurallar kabul edilir, red durumuna karşılık düşenler ise reddedilir. Herhangi bir duruma karşılık düşmeyen kurallar ise geçersiz (unvalidated) olarak işaretlendikten sonra veritabanına yazılır [4,7].

#### **2.5.1.3. Gereksiz kural elemesi (Redundant-rule elimination)**

Bu operatör diğer kurallardan üretilebilecek kuralları eler. Başka bir deyişle müşterinin davranış, alışkanlık ve profili konusunda yeni bir bilgi taşımayan, başka kuralların farklı anlatımı niteliğindeki kuralları devre dışı bırakmaya yöneliktir. Operatör, tekrar eden kuralların önüne geçmek için içerisinde buluna bir algoritmayı döngüsel olarak çalıştırır [6].

### **2.6. Veri Madenciliğindeki Problemler**

Veri madenciliğindeki sorunlar daha çok eldeki ham veri den kaynaklanmaktadır. Diğer sorunlar verinin konu ile uyumsuzluğundan doğabilir [5].

Sınıflandırmak gerekirse başlıca sorunlar şunlardır :

#### **2.6.1. Sınırlı Bilgi**

Veri tabanları genel olarak veri madenciliği dışındaki amaçlar için tasarlanmışlardır. Bu yüzden, öğrenme görevini kolaylaştıracak bazı özellikler bulunmayabilir.

Başlangıçta bir satış departmanı için geliştirilmiş veritabanı genelde, alınan malın adedi, özellikleri, malı alana ait bilgileri içerecek şekilde tasarlanır. Fakat veri madenciliği metodlarını ilgili veriler üzerinde çalıştırabilmek için üretilen kuralların yazılacağı kurallar tablosu, müşteri profillerinin yazılacağı profiller tablosu ve buna gibi tablolara algoritmanın çalışabilmesi için ihtiyaç duyulmaktadır. Bu tabloların



veri madenciliği sürecine başlanmadan önce tanımlanması gerekmektedir.

### **2.6.2. Gürültü ve Eksik Değerler**

Veri özellikleri ya da sınıflarındaki hatalara gürültü adı verilir. Veri tabanlarındaki eksik bilgi ve bu yanlışlardan dolayı veri madenciliği amacına tam olarak ulaşmayabilir. Bu bilgi yanlışlıkları, ölçüm hatalarından, ya da öznel yaklaşımdan kaynaklanabilir.

### **2.6.3. Belirsizlik**

Yanlışlıkların şiddeti ve verideki gürültünün derecesi ile ilgilidir. Veri tahmini bir keşif sisteminde önemli bir husustur.

### **2.6.4. Ebat, Güncellemeler ve Konu Dışı Sahalar**

Veri tabanlarındaki bilgiler, veri eklendikçe ya da silindikçe değişebilir. Veri madenciliği perspektifinden bakıldığında, kuralların hala aynı kalıp kalmadığı ve istikrarlılığı problemi ortaya çıkar. Öğrenme sistemi, kimi verilerin zamanla değişmesine ve keşif sisteminin verinin zamansızlığına karşın zaman duyarlı olmalıdır [16].

### 3. MÜŞTERİ PROFİLİNİN OLUŞTURULMASI

Kişiselleştirme tüccar veya yayıncının bir web sitesi, reklam veya ürün promosyonunu sayesinde, kişinin geçmiş hareketleri veya benzer kişilerin hareketlerinden yola çıkarak belirleme çalışmasıdır. Profil belirleme prosesinin iki temel aşaması kural keşfi (rule discovery) ve doğrulama (validation) dır.

Sonraki aşamada ise elde edilen kurallardan müşteri profilleri elde edilmektedir.

#### 3.1. Veri Modeli

Uygulamalar birçok müşteri hakkında çok çeşitli veri kullanırlar. Birçok uygulama bu verileri iki temel sınıfa ayırır; [7].

Tanımsal (*factual*) : Müşterinin kim olduğuna dair bilgiler içerir.

İşlemsel (*transactional*) : Müşterinin hareketleri ile ilgili bilgiler içerir.

Örneğin müşterilerin alışveriş bilgilerini içeren bir market uygulamasında, tanımsal veri isim, adres, telefon, adres, maaş gibi tanımlama bilgilerini içerir. İşlemsel veri ise hangi tarihte hangi ürünün satın alındığını, alışveriş tutarı vs. gibi bilgileri içerir.

#### 3.2. Profil Modeli

Bütün bir müşteri profili iki parçadan oluşur: tanımsal (*factual*) ve alışkanlıklara dayanan (*behavioral*). Tanımsal profil isim, adres, doğum tarihi gibi bilgiler içerir ve bu bilgiler kişiselleştirme prosesini yapan sistem tarafından müşterinin tanımsal bilgisinden elde edilir [7]. Tanımsal profil aynı zamanda işlemsel veriden üretilen bilgiler de içerebilir. Örneğin “müşterinin en sevdiği içecek Fanta dır” veya “müşterinin geçen ay yaptığı en yüksek tutarlı alışverişin tutarı 200.000.000 TL dir” gibi.

Alışkanlığa dayanan profil modeli müşterinin hareketlerini ifade eder ve genellikle işlemsel veriden elde edilir. Alışkanlıklara örnek olarak “Hasan kırtasiyeden alışveriş yaptığında genellikle roman satın alır” ve “Haftasonları Hasan

kırtasiyelerde 300.000.000 TL den daha fazla para harcar” cümleleri verilebilir



Tablo 3.1 Gerçeğe dayalı (Factual) Veri

| MüşteriNo | Adı    | Soyadı  | Doğum tar. | Medeni Hal | Cinsiyet |
|-----------|--------|---------|------------|------------|----------|
| 0521426   | Hasan  | Kaya    | 02.10.1975 | Evli       | Erkek    |
| 0521168   | Ahmet  | Başgöze | 05.04.1977 | Evli       | Erkek    |
| 0530021   | Serpil | Çiftci  | 10.12.1980 | Bekar      | Bayan    |

Tablo 3.2 İşlemsel(Transactional) Veri

| MüşteriNo | Tarih      | Zaman   | Firma      | Ürün       | Kupon |
|-----------|------------|---------|------------|------------|-------|
| 0521426   | 07/10/2002 | 10:18am | Migros     | Ekmekek    | Evet  |
| 0521426   | 07/09/2002 | 10:18am | Migros     | Meyva suyu | Hayır |
| 0521168   | 07/10/2002 | 10:29am | Carrefourr | Dondurma   | Hayır |
| 0521426   | 07/10/2002 | 07:02pm | Şok Market | Parfüm     | Hayır |
| 0530021   | 07/10/2002 | 08:34pm | Carrefourr | Müzik seti | Hayır |
| 0530021   | 07/10/2002 | 08:34pm | Carrefourr | Meyva suyu | Hayır |
| 0521168   | 07/12/2002 | 01:13pm | Migros     | Biberon    | Evet  |
| 0530021   | 07/12/2002 | 01:13pm | Migros     | Ekmekek    | Hayır |

### 3.3. Hipotez

Her müşteri aynı zamanda potansiyel bir müşteri adayıdır. Günümüzün yoğun rekabet ortamında şirketlerin başarıya ulaşabilmesi ancak müşterilerini iyi tanıması ile mümkün olabilir. Bunun için var olan müşterilerine yeni satışlar yapabilmesi ve onları memnun edebilmesi, yaptığı promosyonlardan hangilerinin hangi özellikli müşterileri tarafından ilgi göreceğini önceden bilmesi, şirkete kalıcı müşteriler kazandıracak promosyon paketlerinin hangileri olduğunu belirleyebilmesi gerekir. Şirket bünyesinde geçmiş dönemlerde toplanan verilerden veri ayrıştırma yöntemleri kullanılarak müşteri profili belirleme, kampanya yönetimi, müşteri sadakati belirleme, vb. amaçlı modeller geliştirilebilir ve geliştirilen modeller ile istenilen karar destek amaçlı bilgilere çok hızlı bir şekilde ulaşılabilir. Perekende sektöründe hangi müşterilerin hangi ürün kombinasyonlarını satın aldıklarının belirlenmesi (basket analizi) oldukça önemli bir veri ayrıştırma uygulamasıdır ve elde edilen

sonular promosyonlarda hedef kitlenin daha doęru belirlenmesi ve rnlerin yerleřtirimi ile ilgili kararlarda nemli karar desteęi saęlar.

Uygulamada sadece zel bir rn veya rnler iin deęil bir tre ait btn rnler iin (rn btn kozmetik rnleri) veya trne bakılmaksızın grupta (domain) penceresi ile oluřturulan gruplar iin kural retilabilir ve bu gruplar iin mřteri listeleri oluřturulabilir.

Uygulamada sonuların istatistiksel grafiklerle grsel bir arabirim aracılıęı ile kullanıcıya gsterilmesi anlařılabilirlięi kolaylařtıracadı gibi kullanıcıya programı kullanma isteęi vereceęinden nem tařımaktadır.

Yapılan alıřmada mřteri-rn analizi yapılarak bu analiz sonucu doęru malı doęru zamanda doęru mřteriye sunabilmek mmkn olabilir. Bu řekilde řirket aısından en karlı olabilecek mřteriler nceden belirlenmiř olacaktır.

### 3.4. Yntem

Bu alıřmada hipotezi doęrulamak iin bir uygulama geliřtirilmiřtir. GYTE-Miner adındaki uygulamada mřteri profillerini belirlemek iin eřitli yntem ve algoritmalar gereklenmiřtir.

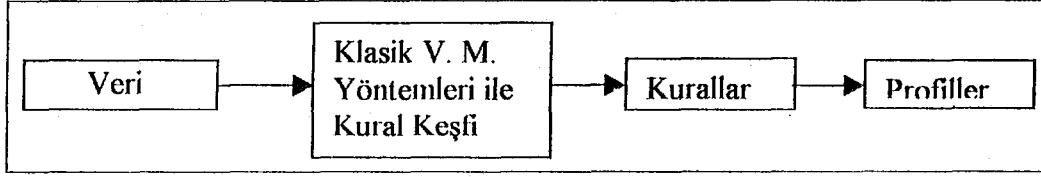
#### 3.4.1. Yntem 1

GYTE-Miner uygulamasında ncelikle nceęi blmlerde anlatılan klasik veri madencilięi yntemleri kullanılarak mřterilere ynelik kurallar elde edilmiřtir. Elde edilen bu kurallar belirli algoritmalara gre analiz edilerek gerekli olan mřteri-rn bilgisi elde edilmek istenmiřtir.

Profil belirleme prosesinin iki temel ařaması kural keřfi (rule discovery) ve doęrulama (validation) dır.

Kural keřfi nceęi blmlerde anlatılan veri madencilięi yntemlerini kullanarak yapılmakta ve kurallar oluřturulmaktadır. Oluřturulan kurallardan ikinci adıma geilir. İlgili kurallar analiz edilerek veri kmesindeki herhangi bir rn iin yapılacak bir promosyonda en uygun n adet mřteri listesi oluřturulmaktadır.

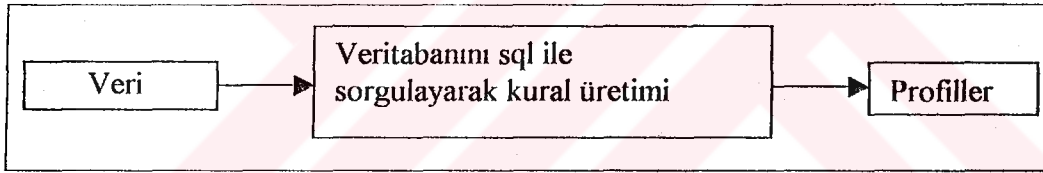
Oluşturulan bu listenin amacı yapılacak olan promosyondan maksimum oranda müşteri geri dönüşü (feedback) ve kar sağlamaktır. Şekil 3.1. yöntemin temel çalışma şeklini göstermektedir.



Şekil 3.1. Kural Tabanlı Profil Belirleme prosesi

### 3.4.2. Yöntem 2

Bu yöntemde veritabanındaki müşteri hareketleri tablosu üzerinden kendi geliştirdiğimiz bir algoritma doğrultusunda kompleks sorgular yürütmekte ve gerekli müşteri profilini ortaya çıkarmaktadır.



Şekil 3.2. Sql Sorgulama Tabanlı Profil Belirleme prosesi

### 3.4.3. Kural Keşfi

Birliktelik kuralları ve sınıflandırma yöntemlerini kullanarak herbir müşteri için ayrı ve bağlayıcı kurallar üretmek mümkündür.

Kişiselleştirme adına biz yaptığımız uygulamada kural keşif metodlarını her müşteri verisine ayrı olarak uygulamak mümkündür. Bu kuralları müşteri veya ürün girdi olmak koşulu ile belirli grup ve domainlere ayırarak ve bu gruplara çeşitli isimler vererek kural üretmek mümkündür.

Yapılan çalışmada farklı türde veri madenciliği algoritması denendi. Uygulamada alışkanlıkları ortaya çıkaracak ve kişiselleştirme prosesinin yapıtaşı

olacak kuralları üretmek için kullanılan algoritmalar aşağıda listelenmiştir.

Kullanılan birliktelik kuralları ;

- Apriori
- Apriori CS
- TBAR
- TBAR cache

Uygulamada adından sıkça bahsedilen Apriori algoritması 1994 yılında Agrawal ve Skirant tarafından geliştirilmiştir. Apriori verilen bir ilişkisel tablodan birliktelik kuralları oluşturmak için alternatiflerden biridir.

Kullanılan sınıflandırma algoritmaları

- Karar Ağaçları (Decision Trees)
- Yıldız Metodolojisi (Star Methodology)
- Birliktelik Kuralları tabanlı modeller (Association-Rule Based Models)

olarak uygulamada implemente edilmiştir.

#### 3.4.4. Alternatif kriterlere duyulan ihtiyaç

Herhangi bir birliktelik kuralları algoritması kullanarak veri madenciliği yaparken, veritabanındaki eleman gruplarının frekansının (dağılım) farklılık göstermesi sahte eleman grupları oluşmasına veya asıl elde edilmek istenilen kuralların gözden kaçmasına sebep olabilir.

Örneğin, 5000 kolej öğrencisi için üniforma bilgisini veritabanından topladığımızı varsayalım. Öğrencilerden 3000 tanesi düzenli olarak spor yapıyor, 3750 tanesi arkadaşları ile dışarı çıkınca alkol alıyor ve 2000 tanesi ise bunların her ikisini de yapıyor. Klasik algoritma, insanların 75% i alkollü içecek içerken "sportmen=>alkol" birlikteliği 66.6% güven oranı ile elde eder. Bu kural yanıltıcıdır, çünkü alkol sportmenler içerisinde alkol kullananların sayısı bütün popülasyondakilere göre daha azdır. Aslında, düzenli spor yapmak alkol tüketimini azaltmaktadır. Böylece {sportmen,!alkol} eleman grubunun {sportmen,alkol} eleman grubundan daha az dağılıma sahip olmasına rağmen, "sportmen=>!alkol"

kuralı 33.3% güven oranı ile daha net sonuç verecektir.

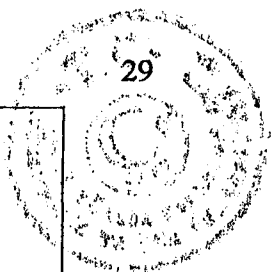
Eğer destek (support) ve güven (confidence) eşik değerleri yeterince düşük olursa, çelişkili birliktelik kuralları elde edilebilir. (yukardaki örnekte olduğu gibi, “sportmen=>alkol” ve “sportmen=>!alkol”). Ayrıca bu eşik değerlerinin aşırı düşük değerlerinden muazzam büyüklükte ve yönetilmesi zor birliktelik kuralları takımı elde edilebilir. Yukardaki örnekte eşik parametrelerini daha yüksek değerlere set ettiğimiz zaman, bizim için daha az anlamlı olan “sportmen=>alkol” kuralını elde ettik. Başka bir deyişle, doğru birliktelik kuralları elde etmek için eşik değerlerinin hangi durumda nasıl set edileceğine dair bir kombinezon bulunmamaktadır. (yukardaki örnek için “sportmen=>!alkol” ).

Yukardaki örneğe benzer durumlar gözlemlendikten sonra Aggarwal ve Yu, ilgilenilen eleman grupları elde etmek için bir alternatif pay kriteri kullanma fikrini öne sürdüler. Bu alternatif pay kriteri kollektif güç (collective strength) olarak adlandırıldı [8].

### 3.4.5. Kural Doğrulama (Rule Validation)

Keşfedilen kuralları doğrulamanın bir yolu, operatörün üretilen kuralları kontrol edip mevcut müşterinin güncel davranışlarına uyup uymadığına karar vermesidir. Operatör bazı kuralları kabul eder ve bazılarını reddeder. Kabul edilen kurallar müşterinin alışkanlıklarını ve profilini bir ölçüde çizmiş olur.

Kural doğrulamadaki önemli bir unsur ise esnekliktir. Çoğu kişiselleştirme uygulamasında müşteri sayısı aşırı fazladır. Örneğin kredi kartı uygulamasındaki müşteri sayısı milyonlara ulaşabilir. Eğer herbir müşteri için ortalama 100 kural ürettiğimizi kabul edersek oluşan toplam kural sayısı yüz milyonlar mertebesinde olacaktır. Bu kadar kuralın bir ekspert tarafından teker teker ele alınması mümkün değildir. Geliştirilen uygulamada bu problem giderilmek için hem oluşturulacak kural sayısı, hem güvenilirlik parametresi ve birkaç ekstra parametre ile bu tür durumlar kısıtlanmıştır. Kurallar üretilmeden önce bu parametreler set edilir ve kural üretme işine başlanır. Set edilen bu parametreler aşağıda belirtilen yöntemleri kullanarak kural doğrulama işini gerçekleştirir.

- 
- (1)  $Urun = MeyveSuyu \Rightarrow Kurum = \text{Şok Market} (2.4\%, 95\%)$   
 (2)  $Urun = Ekmek \Rightarrow Kurum = \text{Migros} (3\%, 88\%)$   
 (3)  $Urun = MeyveSuyu \Rightarrow Kupon = \text{EVET} (2\%, 60\%)$   
 (4)  $Zaman = \text{Sabah} \Rightarrow Gun = \text{Cumartesi} (4\%, 77\%)$   
 (5)  $Zaman = \text{HaftaSonu} \& Urun = \text{PortakalSuyu} \Rightarrow Adet = \text{Büyük boy} (2\%, 75\%)$   
 (6)  $Urun = \text{BebekSandalyesi} \Rightarrow Gun = \text{Pazar} (0.8\%, 61\%)$   
 (7)  $Urun = \text{BebekSandalyesi} \& Kupon = \text{EVET} \Rightarrow Adet = \text{Büyük boy} (2.5\%, 67\%)$

Şekil 3.3 Üretilen birliktelik kuralları

### 3.4.5.1. Benzerlik Tabanlı Kural Gruplama (Similarity Based Rule Grouping)

Şekil 3.2 deki kuralları düşünecek olursak; Eleman yapısı benzerliğine göre, gruplama operatörü 1. ve 2. kuralı aynı gruba yerleştirecektir. Çünkü her iki kuralın eleman yapısı  $Urun \Rightarrow Kurum$  şeklindedir. Dolayısı ile bu tür yapıya sahip olan bütün kuralla 1 ve 2 kuralı ile aynı gruba konulacaktır. Kural 3 bu gruba dahil edilemez, çünkü  $Urun \Rightarrow Kupon$  şeklinde olup farklı bir yapıya sahiptir.

### 3.4.5.2. Şablon tabanlı kural filtreleme (Template-based rule filtering)

Şöyle bir kural şablonunun tanımlandığını düşünelim:

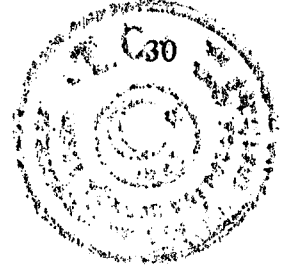
*REJECT HEAD* = {*Kurum* = Şok Market}.

Bu şablon, “başlığında *Kurum* = Şok Market olan bütün kuralları reddet” anlamını taşır. Şekil 3 teki yedi kuraldan sadece kural 1 bu şablonla eşleşiyor, sonuç olarak kural 1 reddedilecek.

Daha karmaşık bir şablon olarak

*ACCEPT BODY* . {*Urun*} AND *HEAD* {*Gun*, *Adet*}.

Bu kural, gövdesinde *Urun* alanına sahip olan (muhtemelen diğer elemanlar içerisinde) ve başlığına *Gun* veya *Adet* geçen kuralları kabul et anlamındadır. Şekil 3.2 de 3. ve 5. kuralların bu kriteri uyduğu görülmektedir.



### 3.4.5.3. Gereksiz kural elemesi (Redundant-rule elimination)

Şu birliktelik kuralını düşünelim ;

$Urun = MeyveSuyu \Rightarrow Kurum = Migros (2\%, 100\%),$

Bu kural Hasanın sadece özel bir hareketini vurgulamaktadır. Kurala göre “O sadece Migrostan Meyve Suyu alır” gibi bir anlam çıkmaktadır. Dolayısıyla bu kuralı olduğu gibi kişinin profiline dahil etmek çok mantıklı olmayabilir. Bu müşterinin daha önce Migros tan yoğun olarak alışveriş yaptığını saptamış olduğunu varsayarsak, elmasıyu kuralı bu kuralın özel bir hali durumunda olur. Bu yüzden “Müşteri sadece Migrostan alır” kuralını Hasan ın profiline koymak elmasıyu kuralını profile eklemek gerekliliğini ortadan kaldırmış olur.

Yukardaki ağaç operatörlerine ek olarak uygulamada diğer kural doğrulama metodları da uygulanmıştır. Bu metodlar görsel, grafiksel ve istatistiksel olarak gerçekleştirilmiştir. Karar ağaçları ve kuralların grafiksel olarak gösterimi yapılmıştır.

Şekil 3.3 de kural doğrulama prosesi görülmektedir. Ekspert in belirlediği *TerminateValidationProcess* durumuna ulaşana kadar yukarda anlatılan metodlardan bir veya birkaçı kurallara uygulanır.

Girdi: Kurallar listesi  $R_{all}$ .

Çıktı: Müşteri-Ürün ilişkili kurallar listesi  $R_{acc}$ ,  $R_{rej}$ ,  $R_{unv}$ , such that  $R_{all} = R_{acc} \cup R_{rej} \cup R_{unv}$ .

(1)  $R_{unv} := R_{all}$ ,  $R_{acc} := \emptyset$ ,  $R_{rej} := \emptyset$ .

(2) **while** (*not TerminateValidationProcess*()) **begin**

(3) Sistem bir kural doğrulama operatörü seçer ( $O$ )

(4)  $O$ ,  $R_{unv}$  e uygulanır. Sonuçlar  $O_{acc}$  ve  $O_{rej}$  in farkundan elde edilir.

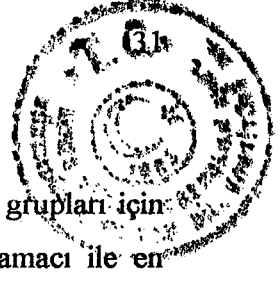
(5)  $R_{unv} := R_{unv} - O_{acc} - O_{rej}$ ,  $R_{acc} := R_{acc} \cup O_{acc}$ ,  $R_{rej} := R_{rej} \cup O_{rej}$ .

(6) **end**

Şekil 3.4 Kural doğrulama prosesinin temel algoritması

Bu aşamadan sonra ekspertin belirleyeceği sayıda müşteri  $R_{acc}$  içerisinde sıralı bir şekilde seçilerek ürün promosyonu gibi amaçlar için kullanılabilir.

### 3.5. Ürün Promosyonu



Hipotez kısmında da bahsedildiği gibi, herhangi bir ürün veya ürün grupları için yapılabilecek promosyon, kampanya vs. gibi amaçlara hizmet etmek amacı ile en uygun müşteri listesinin elde edilebilmesi ürün promosyonu ile mümkün olmaktadır. Bu şekilde şirketin karlılığını arttırmak mümkün olabilmektedir.

GYTE-Miner uygulamasında sözü edilen yöntem ve algoritmalar kullanılarak ürün promosyonu yapılabilmektedir. Uygulamada ürün promosyonu iki farklı yöntem kullanılarak elde edilebilmektedir. Birinci yöntemde daha önce de bahsedildiği gibi, klasik veri madenciliği kullanılarak elde edilen kurallar değerlendirilerek müşteri-ürün ilişkisi ortaya çıkarılmaktadır.

İkinci yöntem daha çok OLAP (Online Analytical Processing) türü bir sistem gibi çalışmaktadır. Operatörün başlangıçta verdiği parametreler doğrultusunda veritabanına bir dizi karmaşık sorgular göndererek müşteri-ürün bilgisini kullanıcıya göstermektedir.

Şekillerde her iki yöntemin aynı ürün için ürettiği sonuçlar gösterilmektedir. Şekil 3.5 deki yüzdeler müşteri için yüzde kaç uygun olduğunu göstermektedir. Şekil 3.6 da sonuç kümesinin daha fazla olduğu görülmektedir. Nedenle grafikte isimler üstüste gelmiştir.

**Ürün Promosyonu - TİBİT**

Seçenekler Yardım

ÜRÜN: **YUMURTA**

En az % **50** katsayılı **100** müşteri

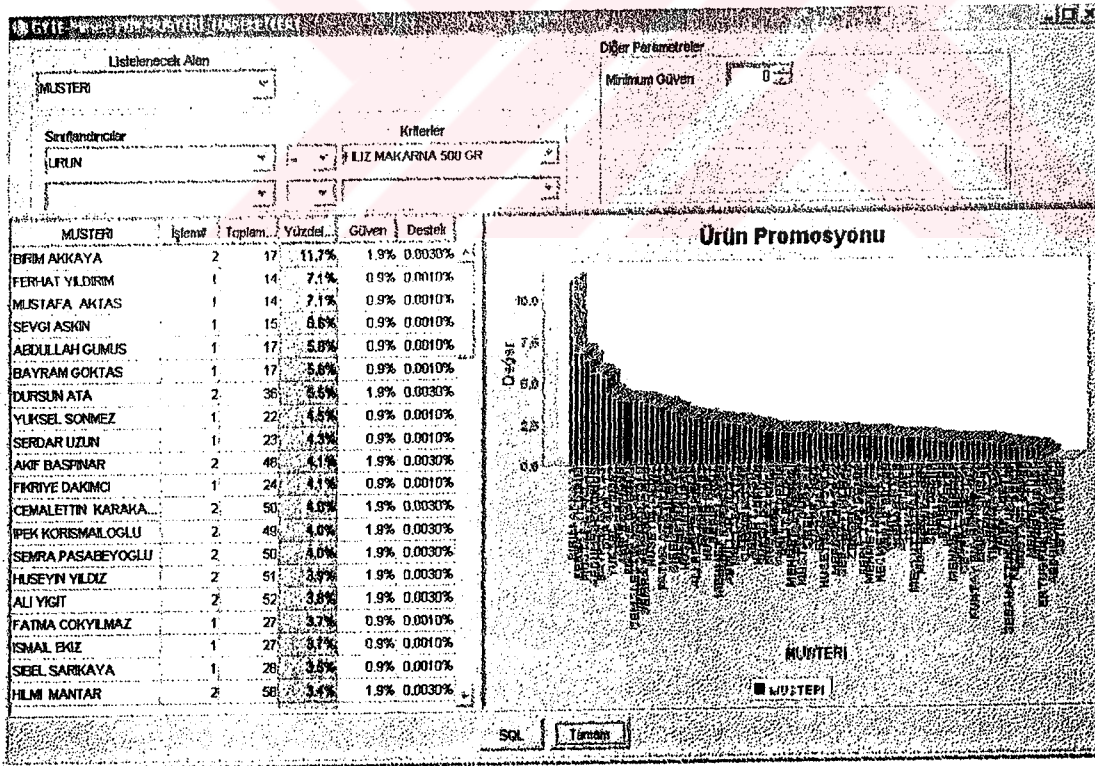
|                  |     |
|------------------|-----|
| MEHMET AKTAS     | 94% |
| NEVZAT YAZANOGLU | 94% |
| HILMI MANTAR     | 93% |
| ZULFU GUL        | 93% |
| DAVUT BASER      | 92% |
| ONDER DANACI     | 92% |
| HASAN ANUL       | 90% |
| HASAN ATASEVEN   | 90% |
| ABDULLAH GUMUS   | 88% |
| BAYRAM KARATAS   | 88% |
| SADIK AKDUMAN    | 87% |
| RAMAZAN ASUK     | 76% |
| IBRAHIM KILINC   | 75% |

İretim: **TOPLAM = 1461 kayıt**

Geri Cıkış

Yardım

Şekil 3.5 Ürün Promosyonu İçin Müşteri Listesi 1



Şekil 3.6 Ürün Promosyonu İçin Müşteri Listesi 2

### 3.6. Gruplama Penceresi



Gruplama penceresi yöntem 1 de elde edilen sonuçlar tatmin edici bulunmamış ve bu amaçla yöntemi sağlamlaştırmak ve daha iyi test edebilmek için oluşturulmuştur. Bu pencere sayesinde sadece özel bir ürün veya ürünler için değil bir türe ait bütün ürünler için (örn bütün kozmetik ürünleri) veya türüne bakılmaksızın gruplama (domain) penceresi ile oluşturulan gruplar için kural üretilebilir ve bu gruplar için müşteri listeleri oluşturulabilir.

Gruplama (domain) penceresi analiz edilmek istenilen nitelikleri gruplamaya, sadece seçilen nitelikler ile alakalı kural üretmeye yönelik hazırlanmış kullanıcının işini kolaylaştıran bir penceredir. Şekilde de görüldüğü gibi kozmetik ürünlerinin hepsine kozmetik diye bir etiket verilmiştir. Sonraki aşamalarda program oluşturulan kurallarda, müşteri listelerinde ve analizlerde ürünlerin adı yerine kozmetik etiketini kullanacaktır. Bu tür bir domain tanımı ürünün ayrıntısından ziyade ürünün türü ile karşılaştırma ve madencilik yapılmak istendiği durumlarda çok faydalı olmaktadır. Ayrıca bütün nitelikler için kural üretmek istemediğimiz durumlarda sadece ilgilenilen nitelikler seçilerek işleme devam edilebilir. Böylece ilgilenilmeyen niteliklerin fazladan kalabalığı engellenmiş olur.

DM KİŞİLERİ HAREKETLERİ Domain Ekranı

Kolon: URUN

Sembolik simge

| Değerler                  | Domainler |
|---------------------------|-----------|
| BASAK PRINC UNU 500 GR    | kozmetik  |
| CARMINA DEO YESİL         |           |
| CARMINA FLU 102           |           |
| CEK-BIRAK OTO             |           |
| DELTA PEN SET 5500        |           |
| FIDAN MAYONEZ 8/1         |           |
| JAGUAR KUMANDALI          |           |
| KENT IRMIK 500 GR         |           |
| KUTU AHSAP PUZZLE 1000 PR |           |
| OZEL 1                    |           |
| SILEN NAT.TUV.KAG.10LU    |           |

Yeni Ekle Sil ID OTOMATİK

Tamam Yardım

Şekil 3.7 Gruplama (Domain) Penceresi



## 4. UYGULAMA

### 4.1. Uygulamanın Genel Yapısı

GYTE-Miner müşteri profili belirlemeye yönelik kurallar oluşturur, bu kuralların görsel olarak gösterimini sağlar. Sistem veritabanında kayıtlı olan müşteri bilgisi ve müşteri hareketlerinden yola çıkarak her bir müşterinin profilini belirlemede kullanılacak geçerli kurallar üretir ve üretilen kurallardan müşterileri daha iyi tanımayı kolaylaştırır. Önceki bölümlerde de bahsedildiği gibi uygulamada bu iş iki farklı metod izlenerek yapılmıştır.

Şekil 4.1 de uygulamanın genel mimarisi gösterilmiştir.

Metod 1 teknik açıdan 3 aşamadan oluşmaktadır.

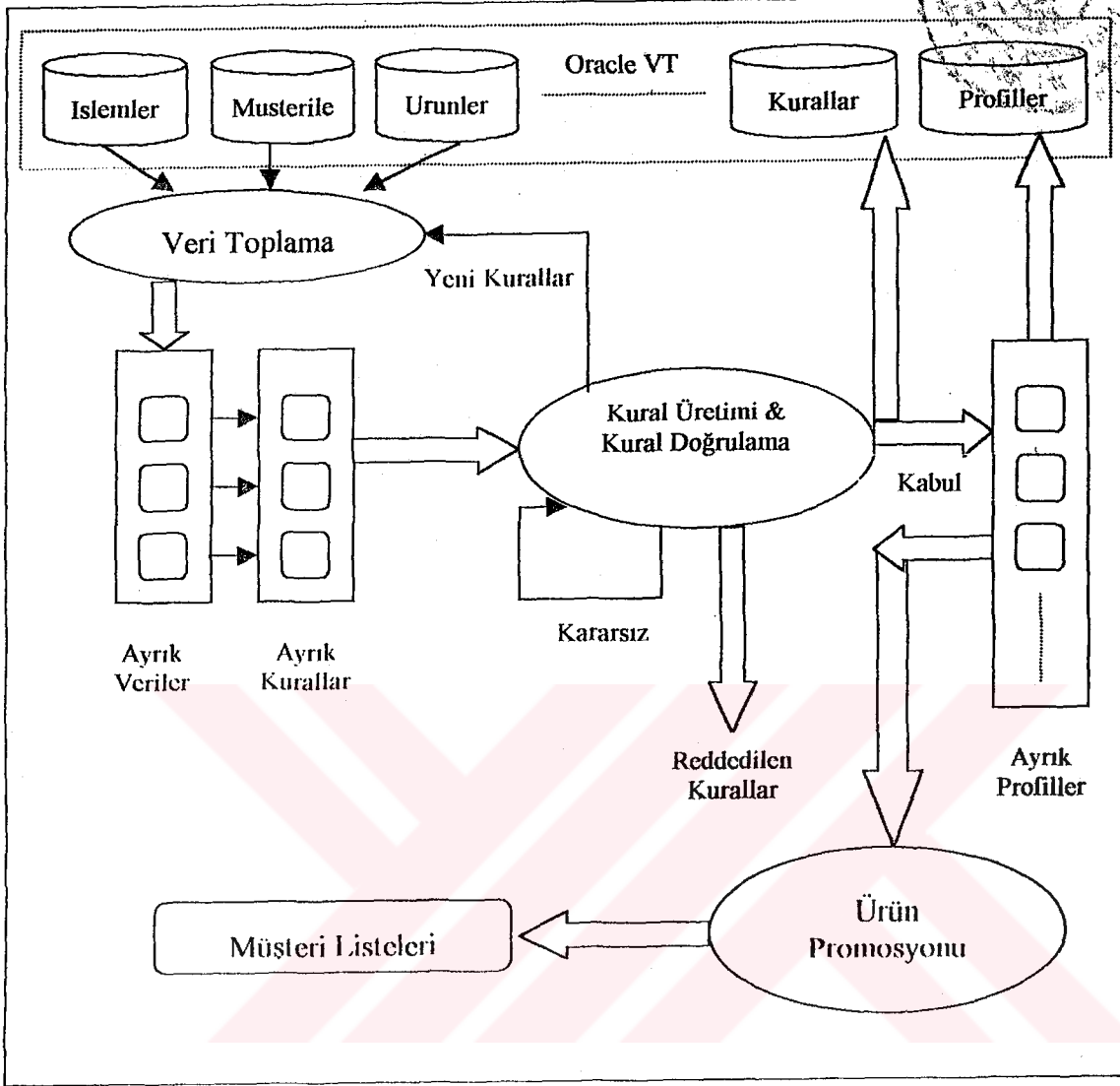
**Birinci aşama :** Çeşitli veri madenciliği yöntemleri kullanılarak kurallar oluşturulur.

**İkinci aşama :** Üretilen bu kurallar kural doğrulama sürecinden geçirilerek daha geçerli ve net kurallar elde edilir. Bu aşamada müşteriye ait kurallar *geçerli kurallar* tablosuna yazılır.

**Üçüncü aşama :** İstenirse eldeki geçerli kurallar kullanılarak istenilen ürün, ürünler veya ürün grupları için yapılabilecek promosyon türü kampanyalarda kullanılmak üzere müşteri listeleri elde edilmektedir.

Bu aşamanın başlangıcında ikinci aşamadan alınan (müşteri-ürün bilgisi içeren) bütün kurallar sistem tarafından üretilmiş geçerli kurallar içerisinde seçilir ve kuralın hangi müşteriye ait olduğunu belirten bir ID verir. Başlangıçta bu kuralların sayısı aşırı fazladır. Bu süreç sonunda hangi müşterinin ilgili ürün promosyonu için yüzde kaç uygun olduğu tespit edilmiş olur.

Kuralları üretmek için kullanılan algoritmalar çeşitli parametre ve eşik değerleri ile (kullanıcı tarafından sisteme girilebilmektedir) ürün promosyonu yada benzer bir amaç için uygun şekilde girilir. Elde edilen kurallar yine kullanıcı tarafından sınıflandırılabilir ve herhangi bir ürün promosyonu için istenilen sayıda uygun müşteri listesi çıkartılabilir.



Şekil 4.1 Uygulamanın Genel Yapısı

Birinci yöntemde kural üretimi ve doğrulama klasik veri madenciliği yöntemlerinden elde edilmektedir. Daha çok karar ağaçlarından üretilen kurallar ele alınıp analiz edilmiştir. İkinci yöntemde ise kural üretimi ve doğrulama işlemi OLAP sistemlerinde olduğu gibi sql sorgulamaları yürütülerek elde edilmektedir.

Her iki yöntemde de operatör tarafından girilen eşik değerlerine (*TerminateValidationProcess*) ulaşıncaya kadar işlem devam etmektedir. *TerminateValidationProcess* yüzdelik bir değer olabileceği gibi müşteri sayısını belirten bir tamsayı da olabilir. Böylece "Bu ürün için en uygun 300 müşteriye

listele” gibi bir istekte bulunulduğunda sistem ilgili ürün promosyonundan en fazla getiri sağlayabilecek 300 müşteri ismini kar oranına göre sıralı olarak getirecektir. Aynı işlemi yüzdelik ifade ile de elde edebiliriz, “%70 e kadar uygun olanları listele” komutu verildiğinde sistem %70 den yukarı olan müşterilerin tümünü listeleyecektir.

Uygulama yapısal olarak üç ana bölümden oluşmaktadır. Kural üretimi, oluşan kurallardan profilleri belirleme ve uygulamadaki verilerin saklanması için kullanılan ilişkisel veritabanı sistemi.

## 4.2. Kural Doğrulama

Müşteri-ürün ilişkisi tarafından ele alınacak olursa herbir müşterinin yaptığı alışveriş başlangıçta ayırık kurallar olarak değerlendirilmiştir. Şekil 4.1. de de görüldüğü gibi bu ayırık kurallardan daha genel kurallar üretilmektedir.

Uygulamada kullanılan her iki yöntemde de üretilen kurallar benzer şekilde doğrulanmaktadır. Yöntem 1 de kurallar karar ağaçları (TDIDIT) ve ilişkisel kural üretimi (ARM) algoritmalarından elde edilmektedir. Yöntem 2 de ise kurallar sql sorgulamaları ile üretilmektedir. Sonraki aşamada üretilen kurallar listesi aşağıdaki aşamalardan geçmektedir

- filtreleme,
- parametrelerin set edilmesi,
- sıralama

filtreleme işlemi sırasında bölüm 3.5 deki yöntemler kullanılarak kural doğrulaması yapılmaktadır.

Parametrelerin set edilmesi kısmında operatörün girdiği değişkenlere de bağlı olan bazı parametreler set edilmektedir. Bu parametrelerden en önemli ikisi güven (confidence) ve destek (support) parametreleridir. Bu parametrelerin tanımları ve formülleri bölüm 2 de anlatılmıştır.

T.C. İÇİŞİLERİ BAKANLIĞI  
MİLLÎ GÜVENLİK BAKANLIĞI  
ULUSAL GÜVENLİK KURULU

Kuralların Düzenlenmesi DÜŞÜNCE

Beğeniler Yardım Ürün Promosyonu

Minimum güven 50 %

Kural 8 / 84

IF  
URUN: 'DOKME BAYRAM SEKERI'  
THEN  
MUSTERI: 'ALİ GULBURUN'  
WITH  
GUVEN=66.6% [2/3]  
AND  
DESTEK=2.0% [2/100]

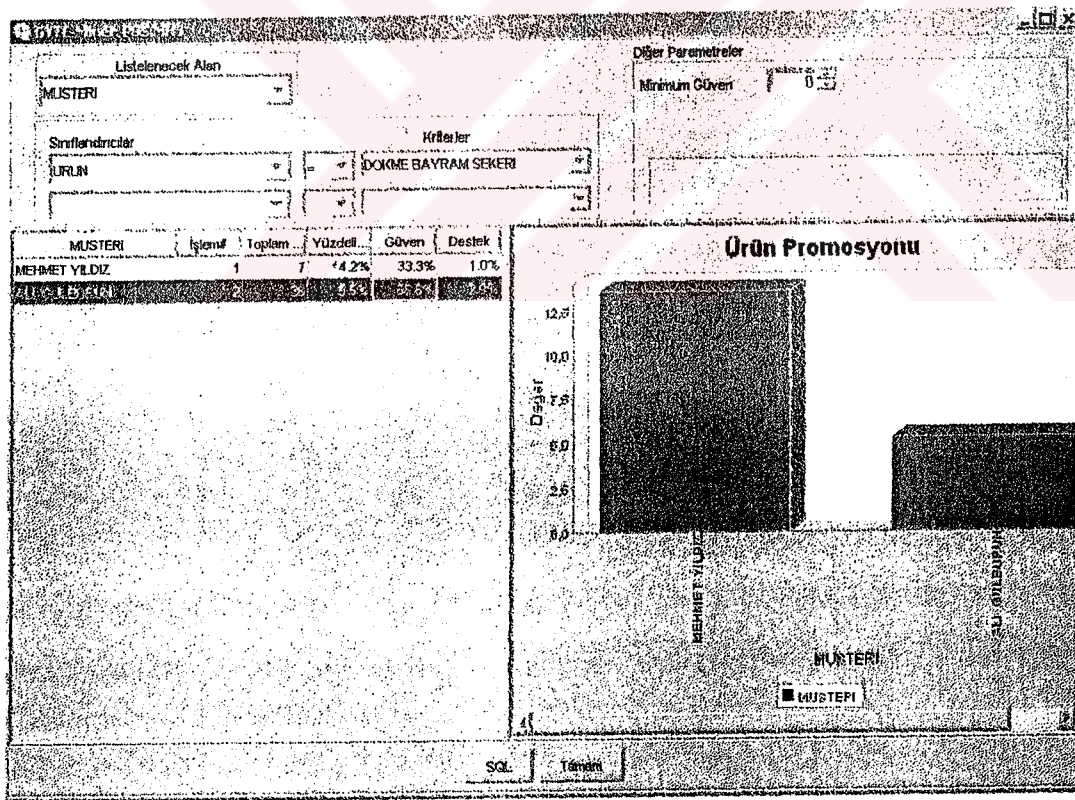
Destek

Güven

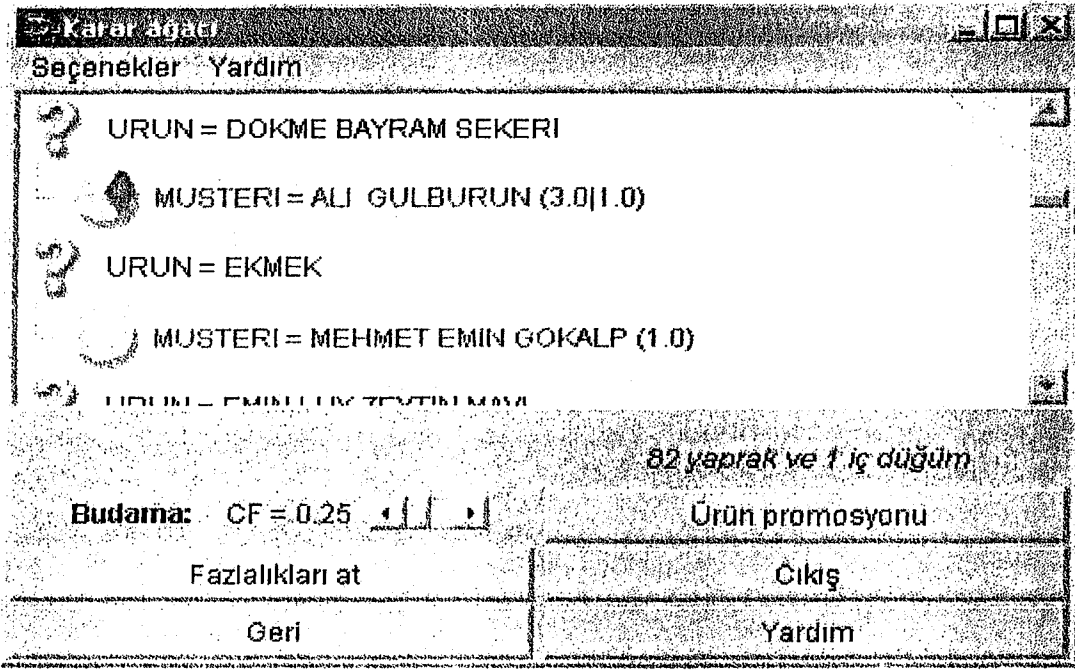
İlk önceki Sonraki Son

Eleman gruplarını tekrar hesapla Kuralları tekrar hesapla Çıkış Yardım

Şekil 4.2 Birliktelik Kuralları 1



Şekil 4.3 Promosyon Penceresi 1



Şekil 4.4 Karar Ağacı Penceresi 1

Şekil 4.2 , Şekil 4.3 ve Şekil 4.4 aynı kurala ait ekran çıktıları bulunmaktadır. Uygulamada daha çok karar ağaçları ve sql sorgularından elde edilen kurallar değerlendirilmiştir. Yukardaki şekillere göre;

Urun = “DOKME BAYRAM ŞEKERİ” ise Müşteri = “ALI GÜLBURUN” kuralı verilmiş ve bu kurala ilişkin sonuçlar gösterilmiştir. Bu kurala göre;

Urun = “DOKME BAYRAM ŞEKERİ” ne X

Müşteri = “ALI GÜLBURUN” = Y diyelim

X ve Y içeren işlem sayısı  $X \cap Y = 2$  (Ali Gülburun’un aldığı şeker sayısı)

X içeren işlem sayısı = 3 (Toplam kaç adet şeker satılmış)

2.1. Formülünden Güven =  $\frac{2}{3} * 100 = 66\%$  bulunmaktadır.

Tablodaki toplam işlen sayısına T dersek,  $T = 100$

2.2. Formülünden Destek  $=2/100*100 = 2\%$  olarak bulunmaktadır.

Şekillerden de görüldüğü gibi birliktelik kuralları ve sql sorgulama metodlarından üretilen bir kuralın güven ve destek parametreleri birbirlerinin aynısıdır. Buradan ikinci metod ile geliştirilen algoritmanın doğruluğu ispatlanmış bulunmaktadır..

Geliştirilen ikinci yöntemde sistemin daha mantıklı kararlar verebilmesi hedeflenmiştir. Bu amaçla güven ve destek parametrelerinden yola çıkılarak elde edilen yüzdelik değerin hesaplanmasında yapay zeka yöntemlerinden faydalanılmıştır. Öyle ki toplam 4 alışverişinin 2 sinde elma satın alan birinci müşteri ile 100 alışverişinden 50 sinde elma satın almış olan ikinci müşteriye ele alırsak, yüzdelik değerleri aynı olmasına rağmen ikinci müşteri promosyon listesinin daha üst seviyelerinde olacaktır.

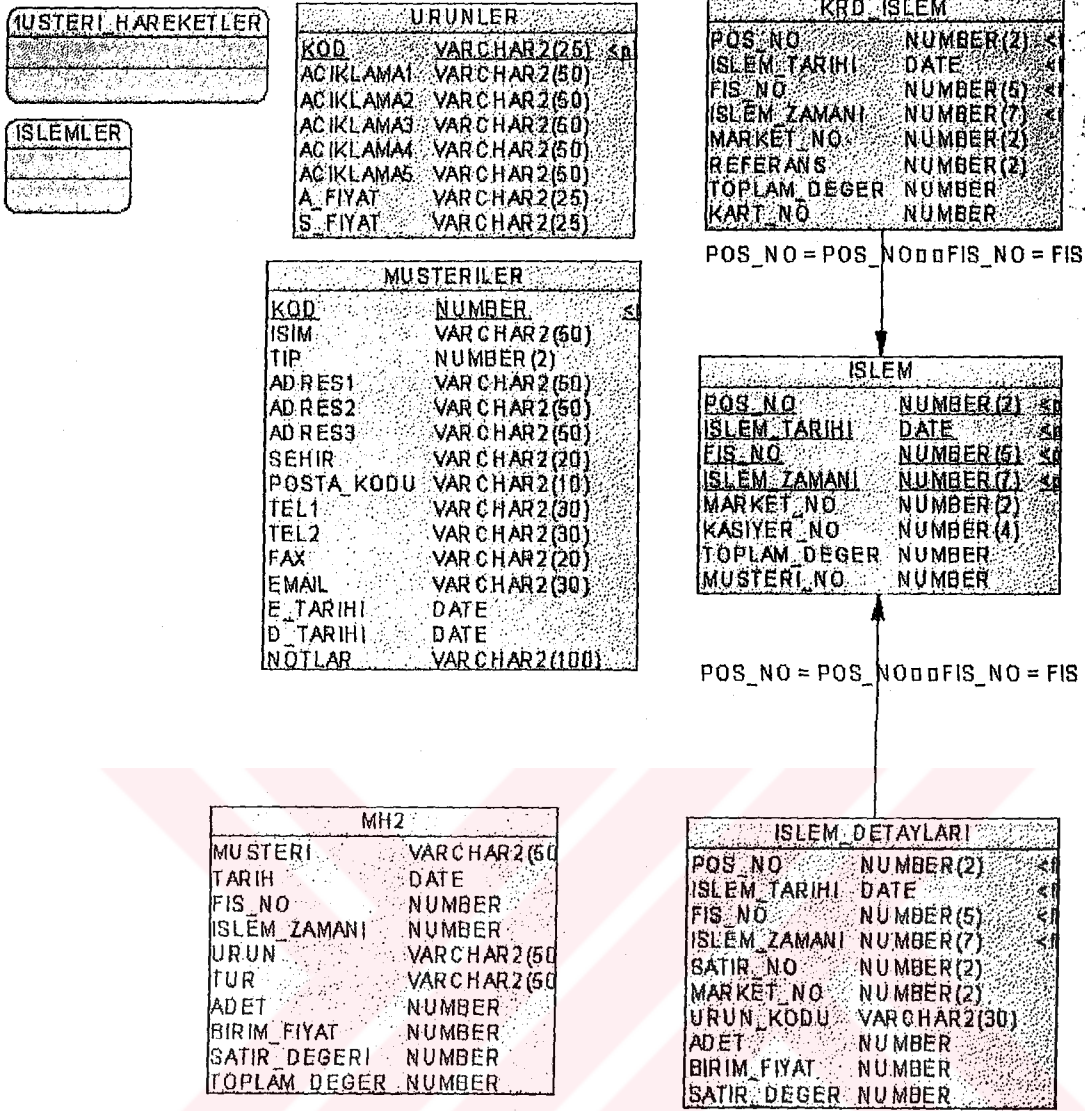
### 4.3. Veritabanı

Uygulamada Oracle 9i kullanılmıştır. Çok büyük miktarda veri depolayabilme, yönetebilme ve güvenilirlik kriterleri göz önüne alınarak bu seçim yapılmıştır. Ayrıca Oracle 9i ile gelen Oracle 9i Veri Madenciliği paketinin kullanılabileceği düşüncesi başlangıçta bu veritabanını seçmeyi cazip kılmış, fakat bu paket uygulamada kullanılmamıştır.

Her ne kadar bu çalışmada veriler Oracle Veritabanında tutulsa da uygulama başka veritabanı sistemleri ile kolayca entegre edilebilir. ODBC konfigürasyonu ile kolay bir şekilde herhangi bir veritabanına bağlanabilmek mümkündür.

#### 4.3.1. Veritabanı Mimarisi

Veritabanında kullanılan en önemli aşağıda gösterilmiştir. İsimlerinden de anlaşılacağı gibi; *ürünler* tablosunda ürün detayları, *Müşteriler* tablosunda müşteri bilgileri bulunmaktadır. *İslem* tablosu herbir müşterinin *t* zamanında yaptığı alışverişin ID ve müşteri numarası arasındaki ilişkiyi belirleyen tablodur. *İslem\_Detayları* tablosu tüm işlemlerin detaylarını içermektedir.



Şekil 4.5 Veritabanı Tabloları

Bu uygulamada kullanılan asıl tablo *Musteri\_Hareketleri* tablosudur. Aslında bu tablo iki tablonun birleşimi olan bir görüntü (view) dır. Hangi müşteri hangi ürünü aldığını bu görüntü sayesinde anlamaktayız. Bu görüntü uygulamanın veritabanı bağımlılığı açısından oldukça esneklik sağlamaktadır. Çünkü herhangi bir veritabanında benzer yapıda bir görüntü oluşturarak uygulamayı bu veriler üzerinde çalıştırmak mümkün olmaktadır.

#### 4.4. Yazılım Araçları

Uygulamada üretilen kurallar, tablolar, profiller birer nesne olarak ele alındığı için bu şartları sağlayacak, gerektiğinde esnek olabilecek, her ortamda çalışabilecek



bir dilde yazılması gerekmektedir. Çünkü tam bir farklı standartlarla da gerektiğinde çalışabilmesi için esnek yapıya sahip olmalıdır. Bu özellikleri genel kapsayan dil Java olduğundan bu dil kullanıldı.

Veri madenciliği kullanmaya ihtiyaç duyan kuruluşlar genelde çok büyük miktarda veri depolamaktadır. Bu nedenle kullanılan veritabanının çok büyük kapasitedeki verilere hakim olması temel bir ihtiyaçtır. Bu nedenle veritabanı olarak Oracle seçilmiştir.

Kullanılan yazılım araçları aşağıda açıklanmıştır.

#### **4.4.1. Oracle 9i**

Veritabanı olarak “Oracle9i Enterprise Edition Release 9.2.0.1.0 – Production With the Partitioning, OLAP and Oracle Data Mining options “ kullanılmaktadır.

#### **4.4.2. Java ve Jbuilder8**

Yazılım dili olarak “Java™ 2 SDK, Standard Edition Version 1.4” ve de yazılım geliştirme editörü olarak “Borland Jbuilder8 “ kullanılmıştır.

### **4.5. Uygulamada Kullanılan Metotlar**

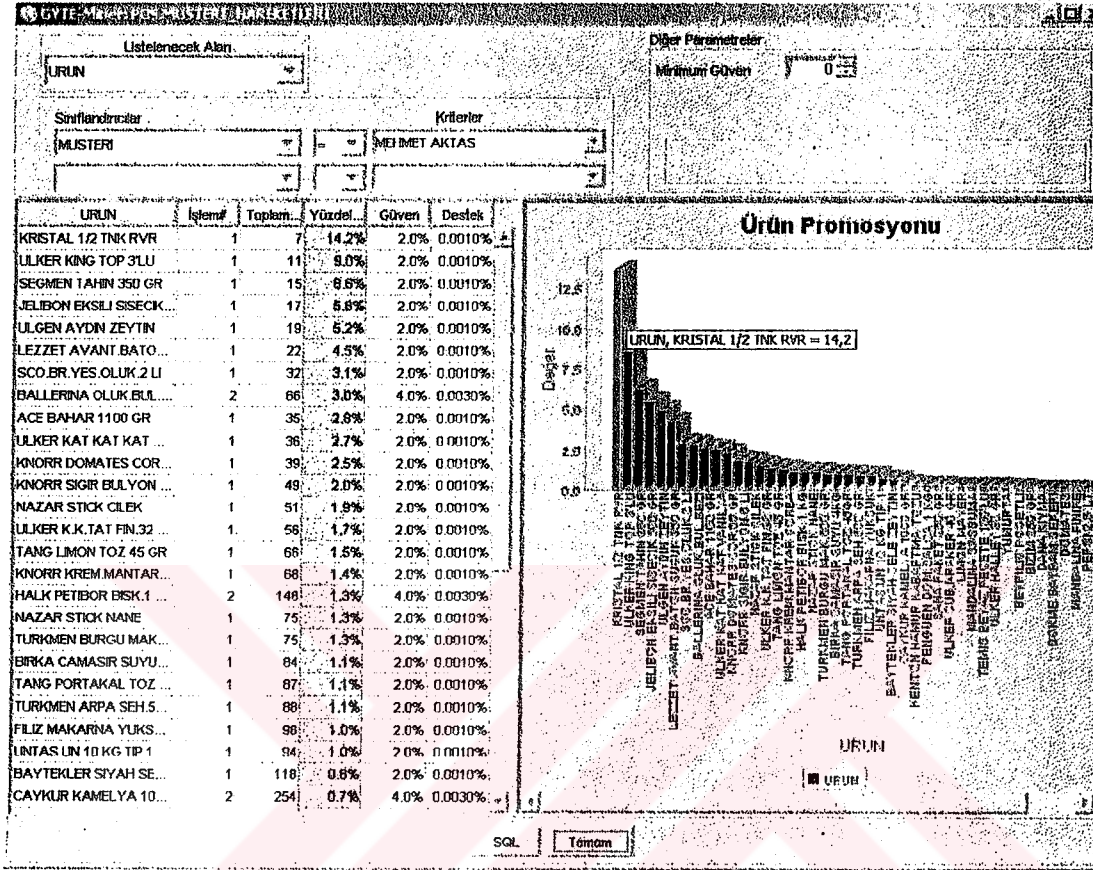
#### **4.5.1. SQL Tabanlı Kural Üretimi**

Çalışma şekli önceki bölümlerde anlatılan sql sorgulama penceresi aşağıdaki şekilde gösterilmiştir. Oldukça görsel bir arabirime sahip olan pencerede kullanıcı promosyon yapılacak alanı ve maksimum iki adet sınıflandırıcı alan seçebilmektedir.

Bu pencere kullanılarak müşteri-ürün kuralı düşünüldüğünde kullanıcı şu tür işlemler yapabilir.

- X ürünü için en uygun müşteriler listesi
- X müşterisi için en uygun ürünler listesi
- En çok/en az alışveriş yapan müşteriler listesi

- En çok/en az satılan ürünler listesi



Şekil 4.6. Promosyon Penceresi 2

#### 4.5.2. Birliktelik Kuralları

GYTE-Miner modülünde kullanılan birliktelik kuralları algoritmaları şunlardır;

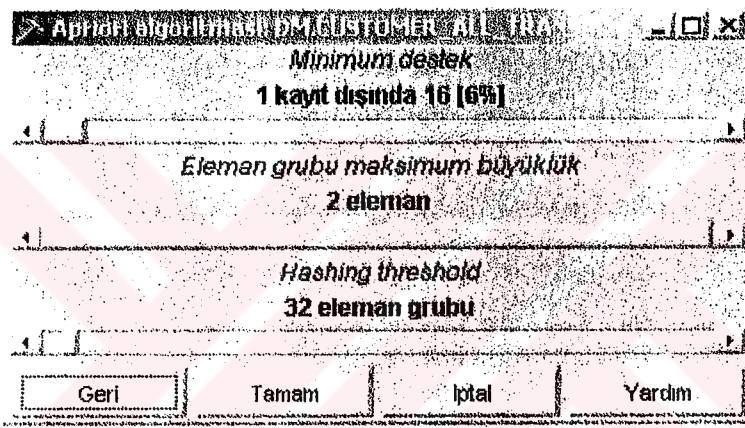
- Apriori
- Apriori CS
- TBAR
- TBAR (cache version)

##### 4.5.2.1. Apriori algoritması

Apriori algoritmasının implementasyonunda aşağıdaki parametreler

kullanılmıştır.

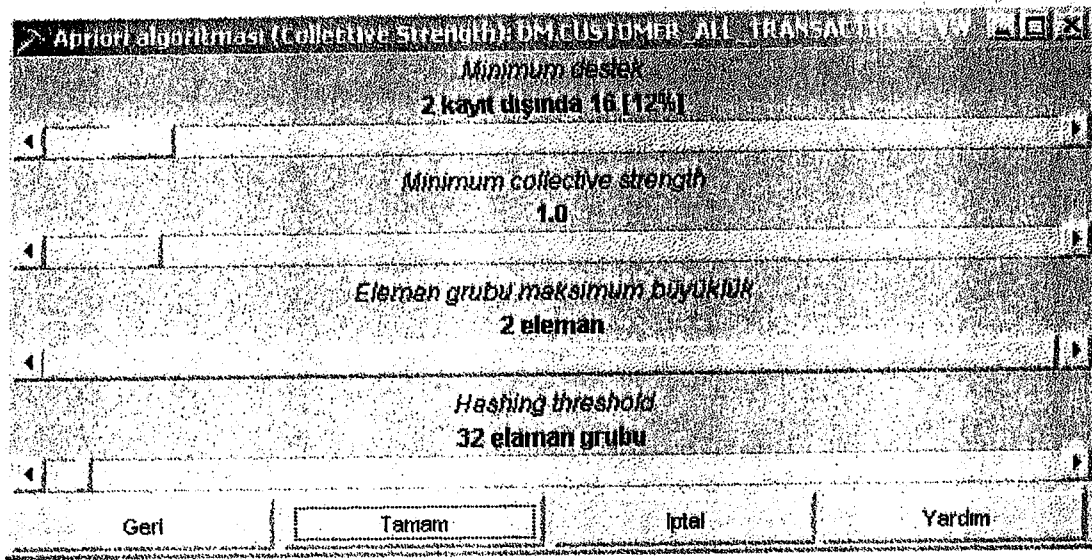
- Minimum destek (MinSupport); verilen bir eleman kümesinde bulunması zorunlu olan minimum sayıdaki kayıt sayısı.
- Eleman kümesi maksimum büyüklük (itemset maximum size) : Eleman kümesinde maksimum kaç eleman bulunacağını belirten bir üst limit. Bu parametre bazı durumlarda oluşabilecek eksponansiyel büyümelerin önüne geçebilmek için konulmuş bir kısıtlamadır.
- Hash algoritma eşiği (Hashing threshold) :Apriori algoritmasının oluşturduğu çok boyutlu bir hash tablosunda her bir düğüme yerleştirilecek maksimum eleman grubu sayısını ifade eder.



Şekil 4.7 Apriori çalıştırma penceresi

#### 4.5.2.2. Apriori CS algoritması

GYTE-Miner aynı zamanda Apriori algoritmasının COLLECTIVE STRENGTH (kollektif güç) parametresine sahip versiyonunun implemantasyonunu da sunmaktadır. Bu parametre minimum destek parametresinin tamamlayıcısı gibidir.



Şekil 4.8 Apriori CS Algoritması

Kullanılan parametreler;

- Minimum destek (MinSupport); verilen bir eleman kümesinde bulunması zorunlu olan minimum sayıdaki kayıt sayısı.
- Eleman kümesi maksimum büyüklük (itemset maximum size) : Eleman kümesinde maksimum kaç eleman bulunacağını belirten bir üst limit. Bu parametre bazı durumlarda oluşabilecek eksponansiyel büyümelerin önüne geçebilmek için konulmuş bir kısıtlamadır.
- Hash algoritma eşiği (Hashing threshold) :Apriori algoritmasının oluşturduğu çok boyutlu bir hash tablosunda herbir düğüme yerleştirilecek maksimum eleman grubu sayısını ifade eder.
- Eleman gruplarının minimum kollektif gücü (Minimum Collective Strength) : Bu parametre 0 a eşit yada daha büyük bir reel sayıdır. 0 değeri mükemmel bir negatif korelasyone işaret eder (eleman grubu içerisinde bir elemanın varlığı, başka elemanları devre dışı bırakması durumu) . 1'e yakın bir değer ise, eleman grubunun varlığı, onun ayırık elemanlarının olasılıklarından elde edilmesi beklenildiği durumlarda verilir.

Butonların işlevleri Apriori penceresindekilerle aynıdır.

#### 4.5.2.3. TBAR Algoritması

GYTE-Miner TBAR diye adlandırdığımız bir algoritmayı da bünyesinde

bulundurmaktadır. Bu algoritma veritabanından birliktelik kuralları elde etmek için kullanılır. Apriori baz alınarak geliştirilmiştir fakat farklı bir veri yapısı kullanır.

Bu algoritmanın implementasyonunda aşağıdaki parametreler kullanılmıştır.

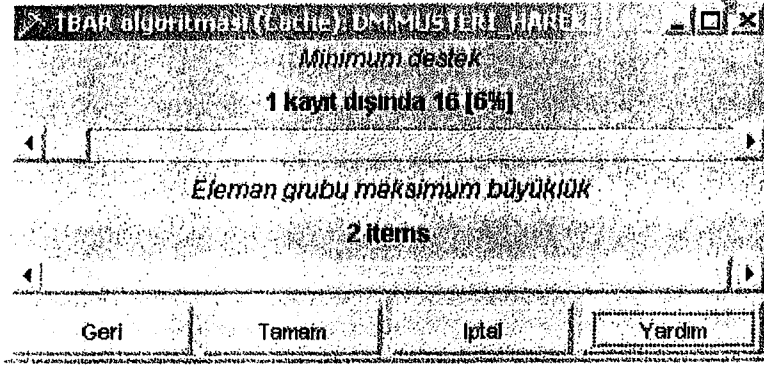
Şekil 4.9 TBAR Algoritması Parametreleri

- Minimum destek (MinSupport); verilen bir eleman kümesinde bulunması zorunlu olan minimum sayıdaki kayıt sayısı.
- Eleman kümesi maksimum büyüklük (itemset maximum size) : Eleman kümesinde maksimum kaç eleman bulunacağını belirten bir üst limit. Bu parametre bazı durumlarda oluşabilecek eksponansiyel büyümelerin önüne geçebilmek için konulmuş bir kısıtlamadır.
- Hash algoritma eşiği (Hashing threshold) :TBAR algoritmasının oluşturduğu çok boyutlu bir hash tablosunda her bir düğüme yerleştirilecek maksimum eleman grubu sayısını ifade eder.
- DHP hash tablo büyüklüğü : DHP (Direct Hashing and Prunning) tarafından kullanılacak olan hash tablosunun büyüklüğünü set etmek için kullanılmaktadır. 0 değeri DHP kullanılmayacağı anlamını taşımaktadır.

#### 4.5.2.4. TBAR algoritması (cache li versiyon)

TBAR algoritmasının cache li versiyonu bütün veri kümesinin belleğe alarak

burada işlem yapar. Böylece tablo boyunun aşırı büyük olmadığı durumlarda mükemmel performans elde edilmiş olur.

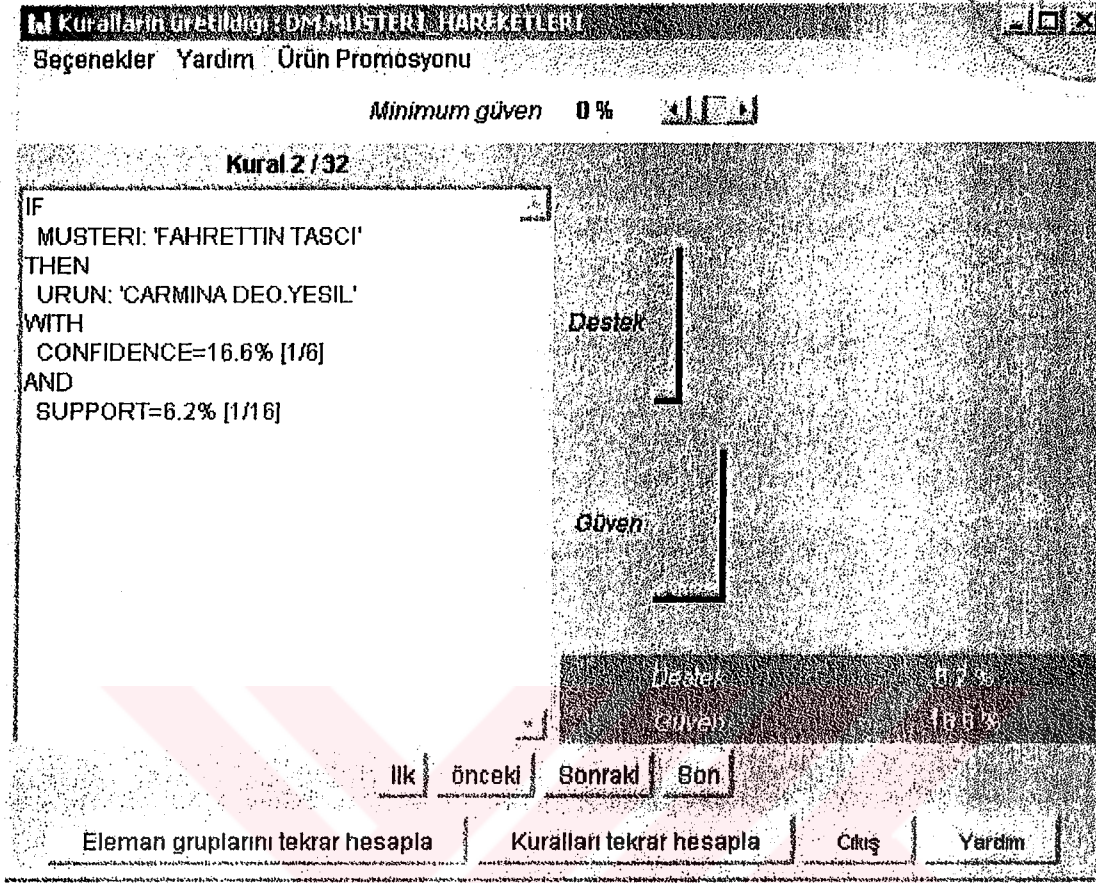


Şekil 4.10 TBAR Cache Algoritması Parametreleri

TBAR ın bu versiyonunun implementasyonunda aşağıdaki parametreler kullanılmıştır:

- Minimum destek (MinSupport); verilen bir eleman kümesinde bulunması zorunlu olan minimum sayıdaki kayıt sayısı.
- Eleman kümesi maksimum büyüklük (itemset maximum size) : Eleman kümesinde maksimum kaç eleman bulunacağını belirten bir üst limit. Bu parametre bazı durumlarda oluşabilecek eksponansiyel büyümelerin önüne geçebilmek için konulmuş bir kısıtlamadır.

#### 4.5.3. Birliktelik Kuralları Kural gösterimi



Şekil 4.11 Birliktelik Kuralları Kural Gösterimi

GYTE-Miner da üretilen her birliktelik kuralı aynı zamanda grafiksel olarak da gösterilmektedir. Grafik kuralın destek ve güven parametrelerini göstermektedir.

Penceredeki *ilk*, *son*, *önceki* ve *sonraki* butonları üretilen mevcut kurallar üzerinde hareket etme imkanı sunmaktadır.

Minimum destek parametresi kullanıcı tarafında interaktif olarak değiştirilebilmekte ve parametrenin yeni değerine göre tekrar kural üretimi yapılabilmektedir. Minimum destek parametresi değiştirilince *kuralları tekrar hesapla* butonu aktif hale gelecektir.

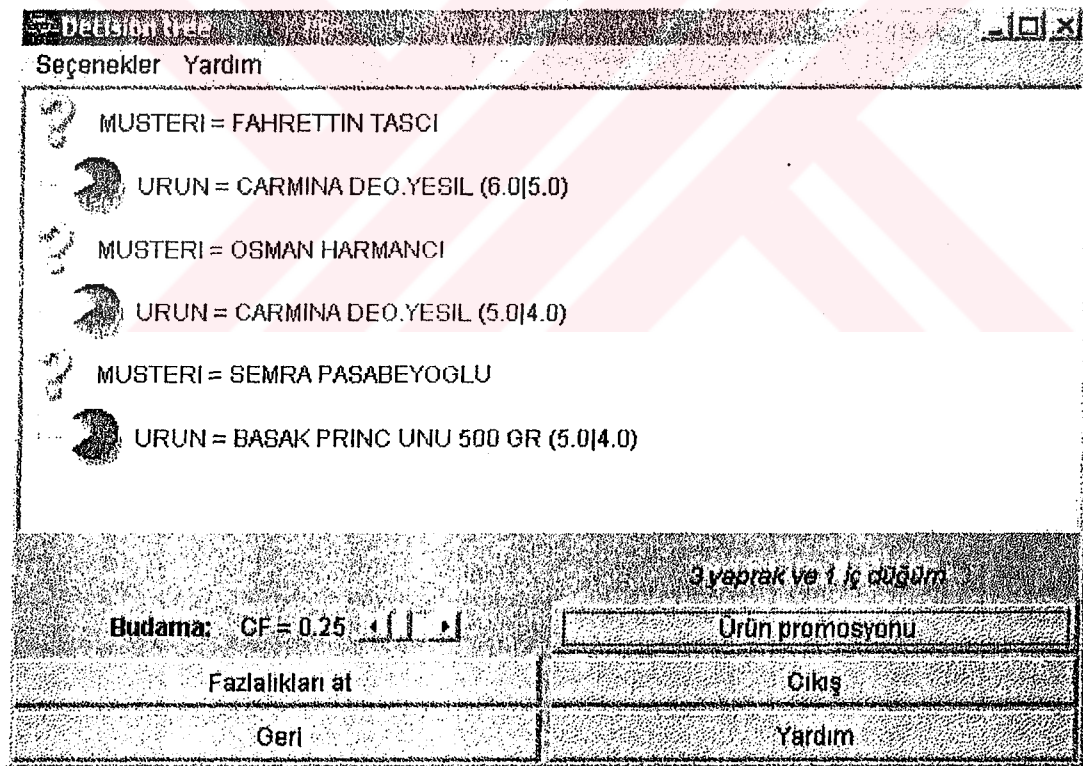
Bununla birlikte *eleman gruplarını tekrar hesapla* butonuna basarak algoritmanın kullandığı diğer parametreleri değiştirip tekrar kural üretilmektedir.

#### 4.5.4. Sınıflandırma Algoritmaları

##### 4.5.4.1. Karar Ağacı

GYTE-Miner karar ağaçlarını grafiksel olarak göstermektedir. Herbir düğümün yanındaki ikona tıklayarak karar ağacının dalları açılmaktadır. Karar ağacının alt seviyesindeki düğümler daha genel sınıfları ifade etmektedir. Pencerede kaç düğüm gösterildiği, kaç iç düğüm olduğu, hatalı sınıflandırmaların sayısı gibi bilgiler gösterilmiştir. Her bir ana düğümün sol tarafındaki küçük sektör diyagramı ilgili ifadenin yüzde oranını göstermektedir. Mavi etiketli kısım doğruluk oranı, kırmızı etiketli kısım ise olumsuzluk oranını ifade etmektedir.

Öğrenme süreci esnasında elde edilen bilgi bir ağaç yapısı şeklinde gösterilmiştir. Her düğüm başlangıcında bir soru işareti ve dairesel bir simge içerisinde muhtemel olasılıklar görsel olarak sunulmuştur.



Şekil 4.12 Karar Ağaçları

Uygulamada karar ağaçları döngüsel olarak yukardan-aşağıya yaklaşımı ile üretilmiştir (genel concept e bağımlı kalınarak). Kullanılan bu yaklaşım TDIDT

(Top-Down Induction on Decision Trees) olarak adlandırılmaktadır.

- Karar ağacını oluşturmak için kullanılan parametrelere *geri* butonu veya *karar ağacı parametreleri* menüsü ile erişilebilir.
- *Fazlalıkları atma* menüsü veya butonu ile budama algoritmasını çalıştırılabilir. GYTE-Miner C4.5 kuralları için J.R.Quinlan tarafından geliştirilen post-prunning algoritmasını keşfetmiştir. Bu budama işlemini yapmak için CF parametresi ayarlanmalıdır.
- Karar ağacından koşul-sonuç (IF-THEN) şeklindeki kurallara geçmek için *Karar ağacından üretilen kurallar* menüsüne yada *Kurallar* butonuna basılmalıdır.

Üretilen kurallardan rapor oluşturulup html olarak saklamak mümkündür.

#### 4.5.4.2. Karar ağacından üretilen kurallar

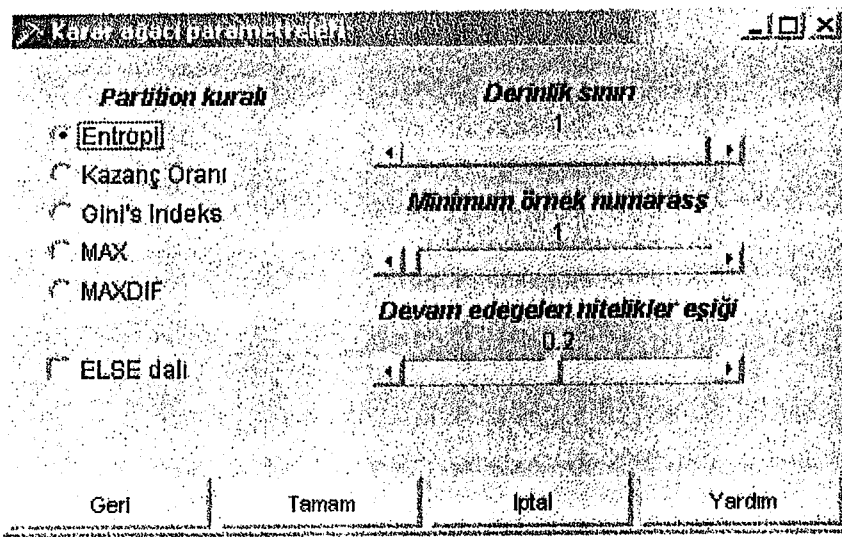
Verilen bir karar ağacından Koşul-Sonuç (IF-THEN) kuralları üretmek çok kolaydır. Pencerenin en üstündeki seçimlik listeden görülmesi istenilen kuralların ait olduğu sınıf seçilebilir.

Karar ağacı etiketli buton ile karar ağacı penceresine geri dönülür. Geri butonu da aynı işi yapmaktadır.

GYTE-Miner aynı zamanda üretilen bu kuralları genelleştirebilir. Kurallar C4.5 kuralları olarak genelleştirilmektedir. Dolayısı ile güvenlik faktörü parametresi olan CF ye burada ihtiyaç duyulmaktadır.

Üretilen kurallardan rapor oluşturulup html olarak saklamak mümkündür.

#### 4.5.4.3. Karar ağacı parametreleri



Şekil 4.13 Karar Ağacı Parametreleri

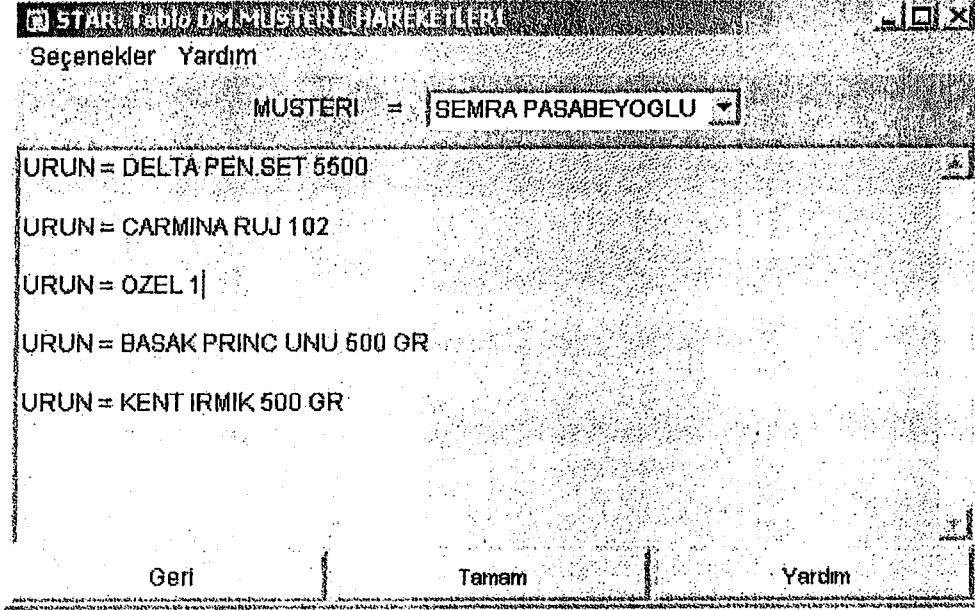
TDIDT algoritmasının ince ayarını yapmak için bu pencere kullanılmaktadır. Penceredeki parametreler aşağıda açıklanmıştır;

- Bölünme kuralı (partition rule) kullanılmıştır.
- ELSE dalının olup olmayacağı.
- Öğrenme kümesindeki minimum örnek sayısı
- Derinlik sınırı : Ön budama parametresi gibi düşünülebilir. Karar ağacının ulaşabileceği maksimum seviye (alt düğümler) sayısını sınırlamak için kullanılmıştır.
- Her yaprakta minimum sayıda bulunması gereken kuralları ifade eder. (örneğin en az 2 yaprakta yoksa buda gibi). Bu parametre de ön budama algoritmasıdır.
- Sayısal nitelik eşiği : Sayısal niteliklerin kullanılması durumunda ikili ağaçta ortaya çıkabilecek dengesiz ağaç oluşumunu engellemek amacı ile kullanılmıştır. (C4.5 kurallarında J.R.Quinlan tarafından geliştirilmiştir.)

#### 4.5.4.4. STAR

GYTE-Miner da STAR metodolojini kullanarak sınıflandırma yapan iki algoritma

gerçeklendi. AQ ve CN2.



Şekil 4.14 STAR Metodolojisi Penceresi

*Star maximum size* parametresi set edilerek *beam search* arama tekniği ile bulunan listenin maksimum büyüklüğü sınırlandırılmaktadır.

#### 4.5.4.5. ARC

GYTE-Miner, sınıflandırma problemlerini çözmede kullanılan TBAR birliktelik kural madenciği algoritmasını içermektedir.(bkz 4.9 Birliktelik Kural Tabanlı Sınıflandırma)

ARC: Tablo D4 MÜSTERİ HAREKETLERİ

Minimum destek  
1 kayıt Dışında 16 [6%]

Eleman grubu maksimum büyüklük  
2 items

Hash sınırı  
14 itemsets

DHP hash tablo büyüklüğü  
BK Girdiler

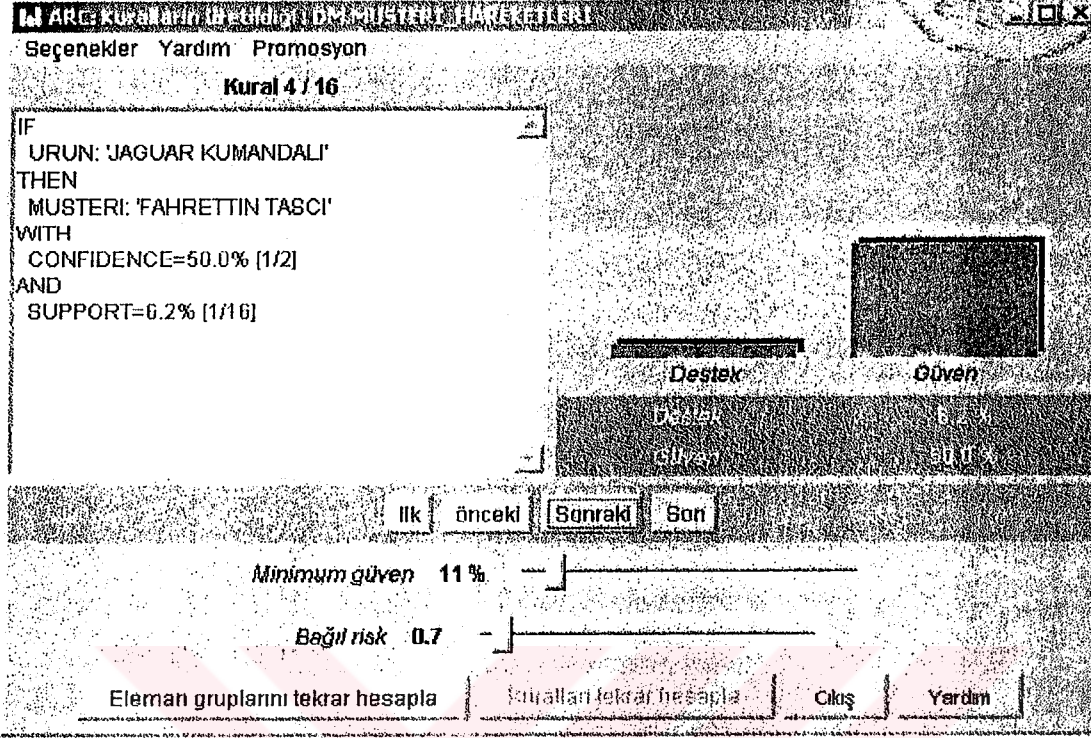
Geri Tamam İptal Yardım

Şekil 4.15 ARC Penceresi

Penceredeki parametreler ;

- Minimum destek (MinSupport); verilen bir eleman kümesinde bulunması zorunlu olan minimum sayıdaki kayıt sayısı.
- Eleman kümesi maksimum büyüklük (itemset maximum size) : Eleman kümesinde maksimum kaç eleman bulunacağını belirten bir üst limit. Bu parametre bazı durumlarda oluşabilecek eksponansiyel büyümelerin önüne geçebilmek için konulmuş bir kısıtlamadır.
- Hash algoritma eşiği (Hashing threshold) :TBAR algoritmasının oluşturduğu çok boyutlu bir hash tablosunda herbir düğüme yerleştirilecek maksimum eleman grubu sayısını ifade eder.

#### 4.5.4.6. ARC Kural Gösterimi



Şekil 4.16 ARC Algoritmasından Üretilen Kurallar

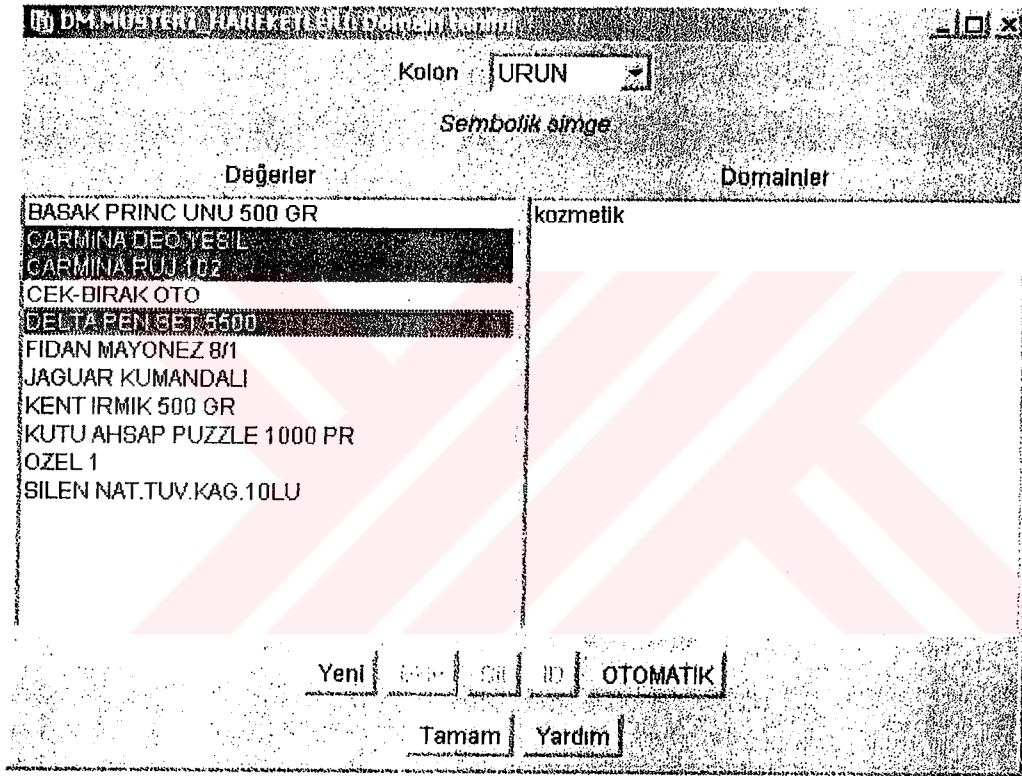
GYTE-Miner da üretilen her birliktelik kuralı aynı zamanda grafiksel olarak da gösterilmektedir. Grafik kuralın destek ve güven parametrelerini göstermektedir. Penceredeki *ilk*, *son*, *önceki* ve *sonraki* butonları üretilen mevcut kurallar üzerinde hareket etme imkanı sunmaktadır.

Minimum destek ve bağıl risk parametreleri kullanıcı tarafında interaktif olarak değiştirilebilmekte ve parametrenin yeni değerine göre tekrar kural üretimi yapılabilmektedir. Parametreler değiştirilince *kuralları tekrar hesapla* butonu aktif hale gelecektir. Bununla birlikte *eleman gruplarını tekrar hesapla* butonuna basarak algoritmanın kullandığı diğer parametreleri değiştirip tekrar kural üretilmektedir.

#### 4.5.5. Gruplama Penceresi

Gruplama (domain) penceresi analiz edilmek istenilen nitelikleri gruplamaya, sadece seçilen nitelikler ile alakalı kural üretmeye yönelik hazırlanmış kullanıcının işini kolaylaştıran bir penceredir. Şekilde de görüldüğü gibi kozmetik ürünlerinin

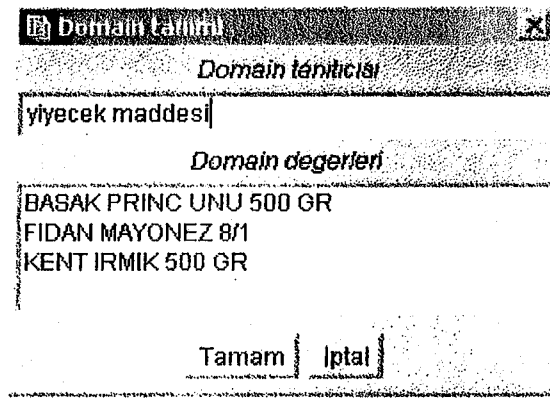
hepsine kozmetik diye bir etiket verilmiştir. Sonraki aşamalarda program oluşturulan kurallarda, müşteri listelerinde ve analizlerde ürünlerin adı yerine kozmetik etiketini kullanacaktır. Bu tür bir domain tanımı ürünün ayrıntısından ziyade ürünün türü ile karşılaştırma ve madencilik yapılmak istendiği durumlarda çok faydalı olmaktadır. Ayrıca bütün nitelikler için kural üretmek istemediğimiz durumlarda sadece ilgilenilen nitelikler seçilerek işleme devam edilebilir. Böylece ilgilenilmeyen niteliklerin fazladan kalabalığı engellenmiş olur.



Şekil 4.17 Gruplama (Domain) Penceresi

Butonlar ;

Yeni : Sol taraftaki listede seçili nitelik yada nitelikler için yeni bir domain tanımlar. Eğer seçili nitelikler birden fazla ise bu seçilen grup için bir grup etiketi girilmesi istenecek ve şekil 4.4 deki pencere ekrana gelecektir. Seçilen nitelik bir tane ise pencere açılmayacak ve program otomatik olarak ilgili niteliğin değerini domain etiketi olarak atayacaktır.



Şekil 4.18 Grup tanıtımı

**Ekle :** Daha önceden tanımlanmış bir gruba yeni şekil 4.5 deki pencere yardımı ile eleman yada elemanlar ekler.

**Sil :** Tanımlanmış domainleri silmek için kullanılır.

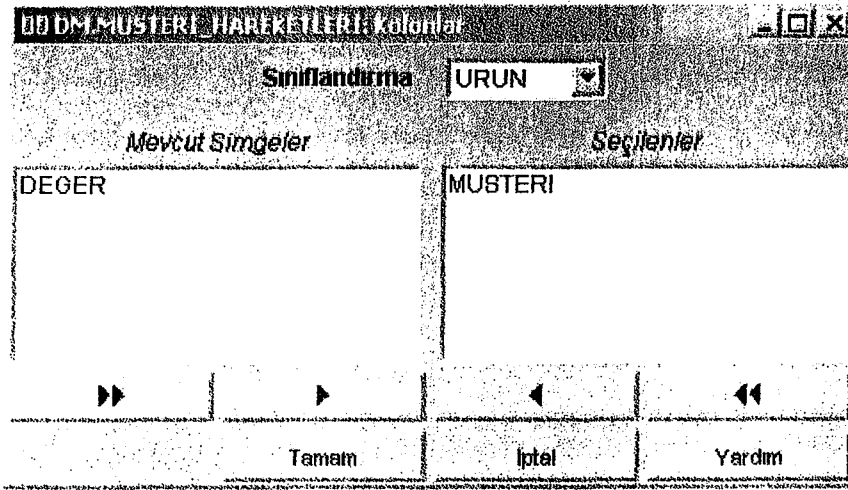
**ID :** Şekil 4.5 deki pencereyi gösterir ve ilgili domain in adını değiştirme, elman ekleme çıkarma vs. gibi işlemler yapmaya olanak sunar.

**AUTO :** Bütün nitelikler için otomatik olarak domain üretir. Eğer daha önce tanımlı domainleriniz varsa üzerine yazabilmek için kullanıcı onayı soracaktır.

Domain tanımlama yada kontrol etme işlemi bittikten sonra tamam butonu ile işleme devam edilir.

#### 4.5.6. Nitelikleri seçme penceresi

Çalışacak olan veri madenciliği algoritmalarının hangi kolonları analiz edeceği bu pencereden seçilir. Şekil 4.6 da görüldüğü gibi soldaki alanda seçilebilecek nitelikler bulunuyor. Seçilenler ise sağda belirmektedir. Algoritmanın çalışabilmesi için en az iki alan seçilmiş olmalıdır. Seçilen niteliklerin sayısı arttıkça algoritmanın çalıştığı zaman artacaktır. Bununla birlikte elde edilen sonuçlar daha ayrıntılı ve karmaşık ve anlaşılabilirliği az olmaktadır. Bu sebeple uygulamanın ürün promosyonu kısmı ikiden fazla nitelik seçilirse çalışmayacaktır.



Şekil 4.19 Nitelikleri seçme penceresi

Buttonlar;

- ▶▶ Mevcut olan bütün nitelikleri seçer. Bunun anlamı bütün nitelikler ilgili algoritma tarafından kullanılacaktır.
- ▶ Sol tarafta seçilmiş olan nitelikleri sağa taşır.
- ◀ Sağ tarafta seçilmiş olan nitelikleri sola taşır. Dolayısı ile bu nitelikler ilgili algoritma tarafından artık kullanılmayacaktır.
- ▶▶ Sağdaki bütün nitelikleri sola taşır.

Eğer bir sınıflandırma algoritması çalıştırılacak ise nitelik seçme penceresinde şekil 4.7 de görüldüğü gibi sınıflandırıcı bir nitelik seçilmelidir. Seçilen niteliğin değeri sınıfı ifade etmektedir. Bu nitelik pencerenin üst tarafındaki seçimlik listeden seçilir.

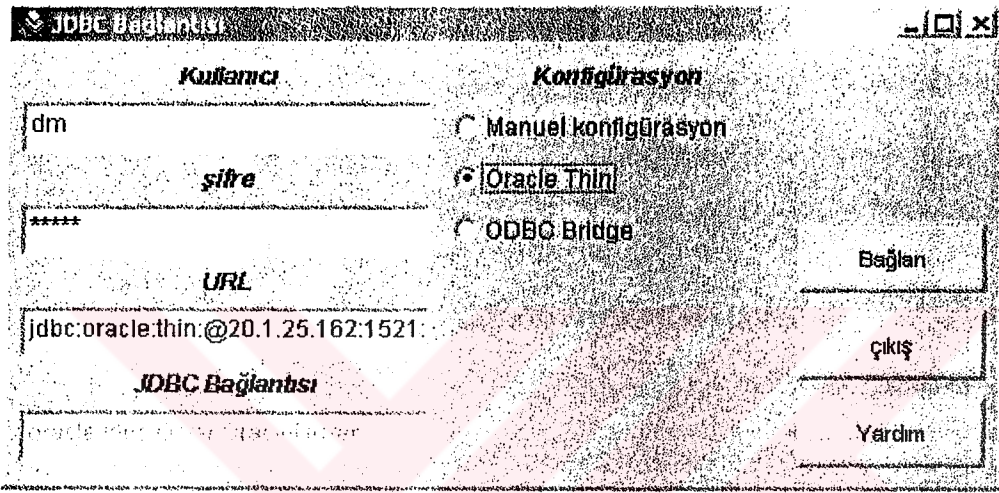
Eğer birliktelik kuralları üretilmek isteniyorsa yada birliktelik kuralı madenciliği kullanan karma bir model kullanılacaksa (ARC veya ART gibi), GYTE-Miner nitelikleri gruplamaya ve domainler oluşturmaya izin verecektir. Bu durumda domain butonu pencerenin sol alt tarafında belirmiş olacaktır. Bu buton ile domain penceresine girip seçili niteliklerin değerlerini gruplayabilirsiniz.

İstenilen nitelikler seçildikten ve gruplandıktan sonra (gruplama isteğe bağlıdır) tamam butonuna basarak algoritma başlatılır.

## 4.6. Uygulamada Kullanılan Diğer Pencereler

### 4.6.1. JDBC Bağlantısı

Bir veritabanı sunucusuna bağlanmak için aşağıdaki parametrelerin eksiksiz girilmesi gerekir;



Şekil 4.20 JDBC Penceresi

- Kullanıcı : Veritabanında tanımlı olan bir kullanıcı ismi.
- Şifre : İlgili kullanıcının şifresi.
- URL : Haberleşme protokolünü içerisinde barındırır. IP adresi (eğer DNS sunucu kullanılıyor ise), bağlantının kullandığı TCP portu (eğer bağlantı soketler ile sağlanıyor ise) ve bağlanılacak olan veritabanı adı gibi bilgiler içerir.
- JDBC sürücüsü : Genellikle veritabanı tarafından desteklenir durumdadır.

GYTE-Miner farklı veritabanlarına bağlanabilme opsiyonu sunmaktadır. Temel olarak oracle veritabanına bağlantı esas alınmıştır. Bunun yanısıra ODBC köprüsü yardımı ile de istenilen veritabanına bağlantı yapabilme şansı sunmaktadır.

Veritabanına bağlanabilmek için yukarda açıklanan parametreleri eksiksiz ve doğru olarak girmek gerekmektedir. Aşağıda aynı zamanda uygulamanın da kullandığı oracle veritabanına bağlantı örnek olarak açıklanmaktadır.

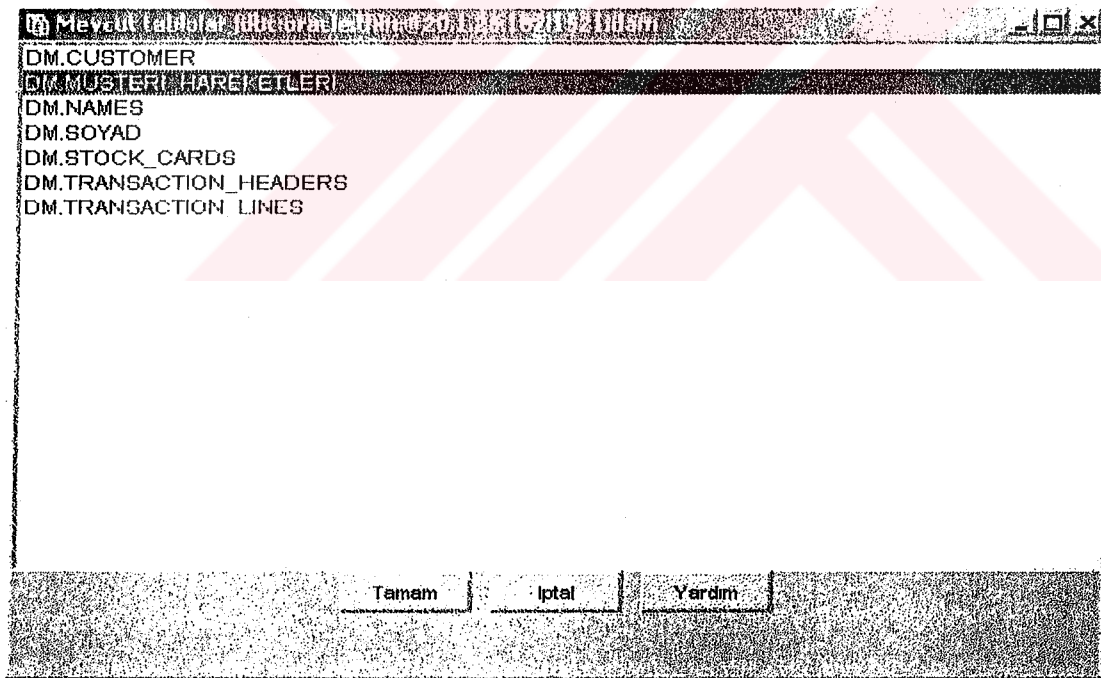
#### 4.6.1.1. Oracle veritabanı bağlantı konfigürasyonu

Oracle veritabanı sunucusunun 20.1.30.100 ip numaralı makinada çalıştığını, veritabanı adının (Oracle SID) DM olduğunu, veritabanı dinleyicisi 1521 numaralı porttan dinlediğini düşünürsek; URL "jdbc:oracle:thin:@20.1.30.100:1521:DM" şeklinde olacaktır. Verilerin bulunduğu kullanıcı adı ve parolası yazıldığı takdirde bağlantı başarılı bir şekilde sağlanmış olacaktır.

#### 4.6.1.2. ODBC Köprüsü

Sistemde JDBC sürücülerinin olmaması durumunda JDBC-ODBC köprüsü yardımı ile yine oracle veritabanına bağlanılabilir. Bu durumda URL ; "sun.jdbc.odbc.JdbcOdbcDriver" yada "jdbc:odbc:dsn" şeklinde olacaktır. ODBC köprüsü, Access, paradox, Sybase vs. gibi başka veritabanlarına bağlanarak da işlem yapabilme imkanı sunmaktadır.

#### 4.6.2. Tablo Seçme Penceresi

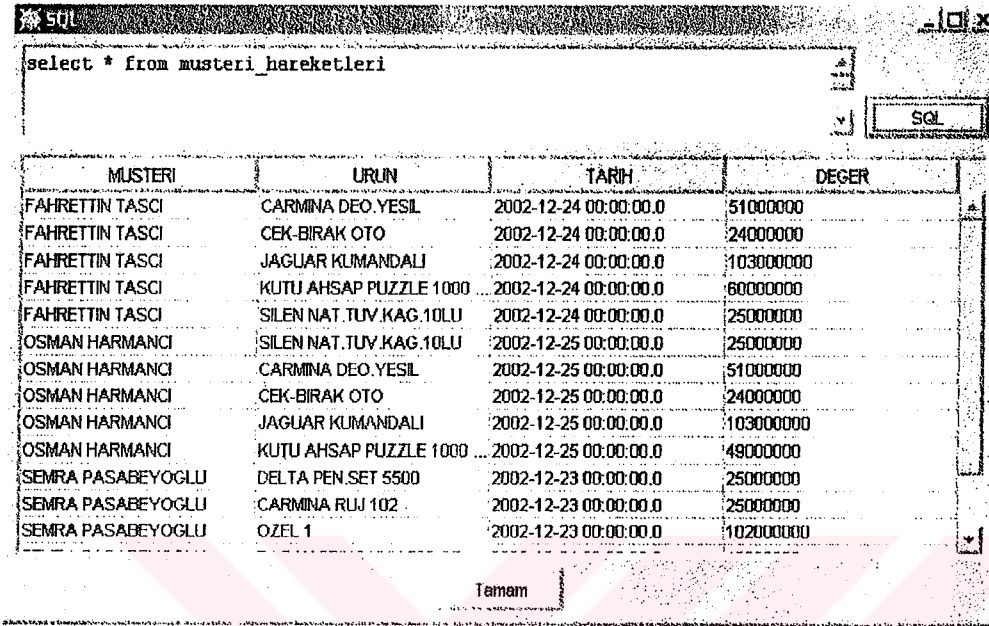


Şekil 4.21 Tablo Seçme Penceresi

Başarılı bir JDBC bağlantısından sonar ekrana mevcut şemadaki tablo ve görüntü listesi gelir. Bu listeden algoritmaların yürüteceği ve analiz yapılacak olan tablo veya görüntü seçilmelidir.

#### 4.6.3. SQL Sorgulama Penceresi

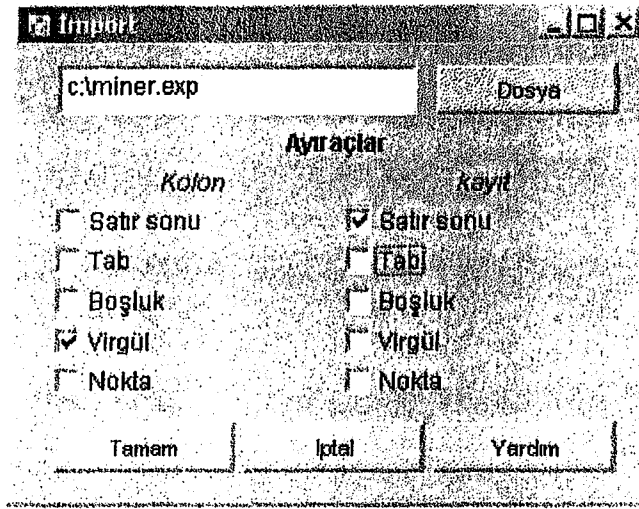
Kullanıcının ihtiyacını karşılayabilecek özel sql sorgulamaları bu pencereden yapılabilmektedir.



Şekil 4.22 SQL Sorgulama Penceresi

#### 4.6.4. Veritabanına Bilgi Alma (Import Data)

Bu özellik üzerinde çalışılan veritabanına bir ASCII dosyadan veri transfer edebilmek için eklenmiştir. Veriyi import edebilmek için ayraçlar, satır sonu vs. gibi parametreler uygun olarak set edilmelidir. Ayrıca ilgili dosyadaki verinin veritabanı sosyasına doğru şekilde alınabilmesi için dosyadaki format tablonun formatına uygun olmalıdır.



Şekil 4.23 Veritabanından Bilgi Alma Penceresi

#### 4.6.5. Veritabanından dışarıya bilgi transferi (Export Data)

Import işleminin tersi olarak veritabanındaki bilginin bir dosyaya aktarılması işlemidir. Import penceresindeki ayrıntılara burada da dikkat edilmelidir.

## 5. SONUÇ

Yapılan bu çalışma sonucunda herhangi bir veritabanındaki bilgiler kullanılarak birliktelik ve sınıflandırma kurallarının başarılı bir şekilde üretilbildiği görüldü. Üretilen müşteri-ürün kurallarından faydalanılarak ürün promosyonu için en uygun müşteri listesinin elde edildiği gözlemlendi. Oluşturulan bu müşteri listelerinden faydalanarak ilgili kişilere mail, adres, telefon veya daha farklı kanallardan ulaşmak mümkün olabilir.

Geliştirilen sistemi test etmek için bir süpermarketin gerçek verileri sistemin veritabanına aktarıldı. Aktarılan veriler üzerinde değişiklik yapılarak sisteme uygun hale getirildi. Verilerin gerçek dünyadan alınmış olması nedeniyle oldukça homojen bir yapıya sahip veritabanı üzerinde çalışıldı. Veritabanındaki sistemi en çok ilgilendiren değerlerin aşağıdaki şekilde olduğu gözlemlenmiştir;

Tablo 5.1. Veritabanındaki Kayıt Durumu

|                                               |        |
|-----------------------------------------------|--------|
| Müşteri tablosundaki müşteri sayısı           | 4752   |
| Ürünler tablosundaki ürün sayısı              | 29.778 |
| Müşteri Hareketleri tablosundaki toplam işlem | 57.775 |

Veriler bir süpermarketin 25/11/2002 ve 23/12/2002 tarihleri arasındaki kayıtlarıdır. Bu nedenle sistemde üretilen sonuçlar bu bir ay için değerlendirilmelidir. Farklı aylar için farklı tablolar oluşturulabilir (özellikle büyük veritabanlarında) veya farklı aylara ait işlem kayıtları birleştirilerek bütün işlemler için veri madenciliği yapılabilir. Bu çalışmada da bellek kısıtlaması ve zamandan kazanmak için müşteri hareketleri tablosu bazı testler için birkaç alt parçaya bölünmüştür.

Oluşturulan bu veriler üzerinde uygulama çalıştırıldı. Elde edilen sonuçlar analiz edildi ve başlangıçta referans alınan bazı hareketlerin üretilen kurallardan okunduğu gözlemlendi. Buradan yola çıkarak üretilen kuralların doğru olduğunu söylemek mümkündür.

Bundan sonra üretilen kurallardan ürün promosyonları için müşteri listeleri oluşturulmuştur. Burada elde edilen listeden yine başlangıçta referans alınan bazı müşterilerin bulunduğu gözlemlenmiştir. Veriler üretilirken belli kişilerin belli ürünleri

sıklıkla aldığı yaklaşık %80 gibi bir doğruluk oranı ile müşteri listelerinden gözlenmiştir.

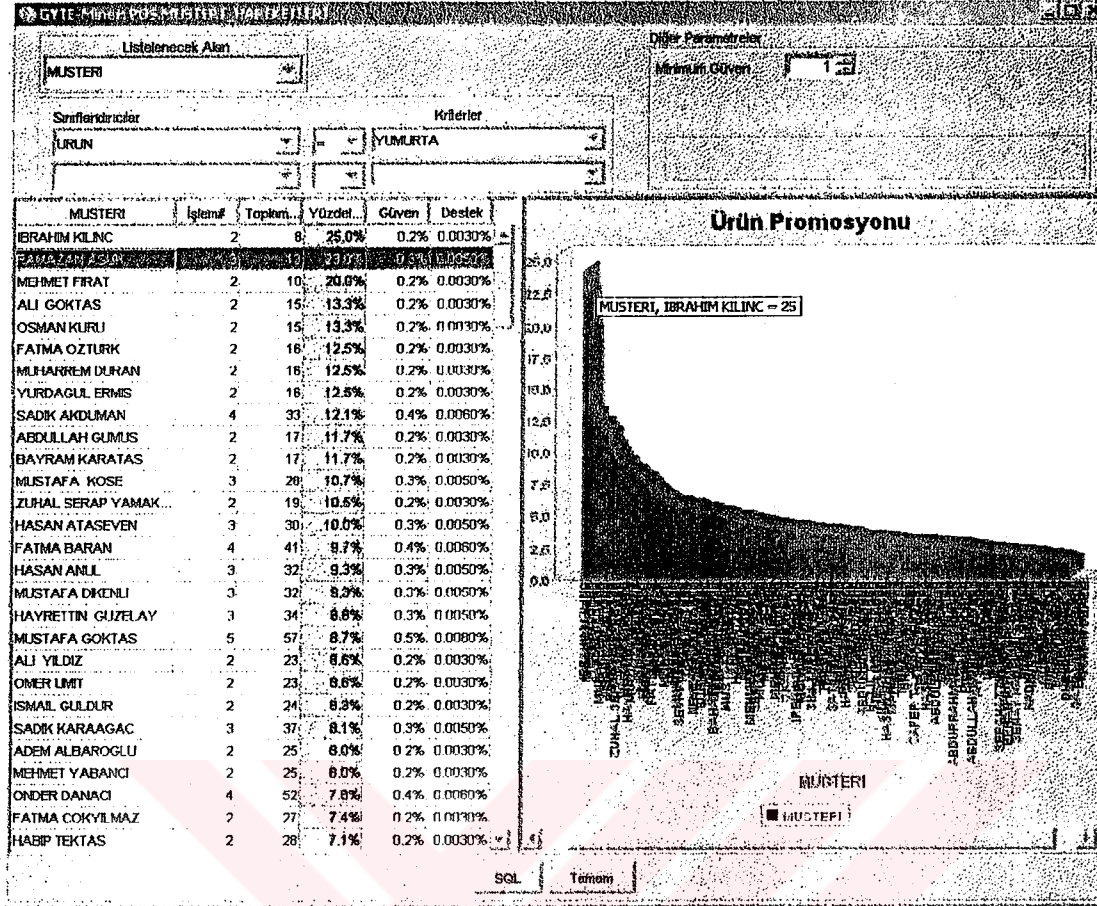
Aşağıdaki şekilde karar ağcından elde edilen kurallar değerlendirilerek oluşturulmuş bir müşteri-ürün listesi bulunmaktadır. Şekil 5.2. de ise aynı ürüne ait promosyonun ikinci yöntem ile elde edilmiş sonuçları görülmektedir. Şekillerden de anlaşılacağı üzere yöntem 2 nin daha ayrıntılı bir sonuç verdiği açıkça ortadadır.

Bu sonuçlara göre karar ağacından üretilen kuralların ürün promosyonu için çok elverişli olmadığını söyleyebiliriz. Karar ağacı kuralları daha çok bir durum makinası gibi çalışırlar ve bir durum için bütün olasılıkları değil sadece bir dala saparlar. Bu özelliği sebebi ile sonuç kümesi yöntem 2 ye göre oldukça küçüktür.

Ayrıca karar ağaçlarının algoritması gereği fazla bellek tükettiği gözlemlendi ve büyük tablolarda çalıştırıldığında bellek yetersizliğinden algoritmanın çalışmadığı gözlemlendi. Yöntem 2 de ise böyle bir durum ile karşılaşılması.

| Seçenekler Yardım |                          |
|-------------------|--------------------------|
| ÜRÜN              | KUMURTA                  |
| En az %           | 50 katsayılı 100 müşteri |
| MEHMET AKTAS      | 94%                      |
| NEVZAT YAZANOGLU  | 94%                      |
| HILMI MANTAR      | 93%                      |
| ZULFU GUL         | 93%                      |
| DAVUT BASER       | 92%                      |
| ONDER DANACI      | 92%                      |
| HASAN ANUL        | 90%                      |
| HASAN ATASEVEN    | 90%                      |
| ABDULLAH GUMUS    | 88%                      |
| BAYRAM KARATAS    | 88%                      |
| SADIK AKDUMAN     | 87%                      |
| RAMAZAN ASUK      | 76%                      |
| IBRAHIM KILINC    | 75%                      |
| Üretim            | TOPLAM = 1467 kayıt      |
| Geri              | Çıkış                    |
|                   | Yardım                   |

Şekil 5.1. Karar Ağacından üretilen Promosyon



Şekil 5.2. SQL Sorgulamasından üretilen Promosyon 1

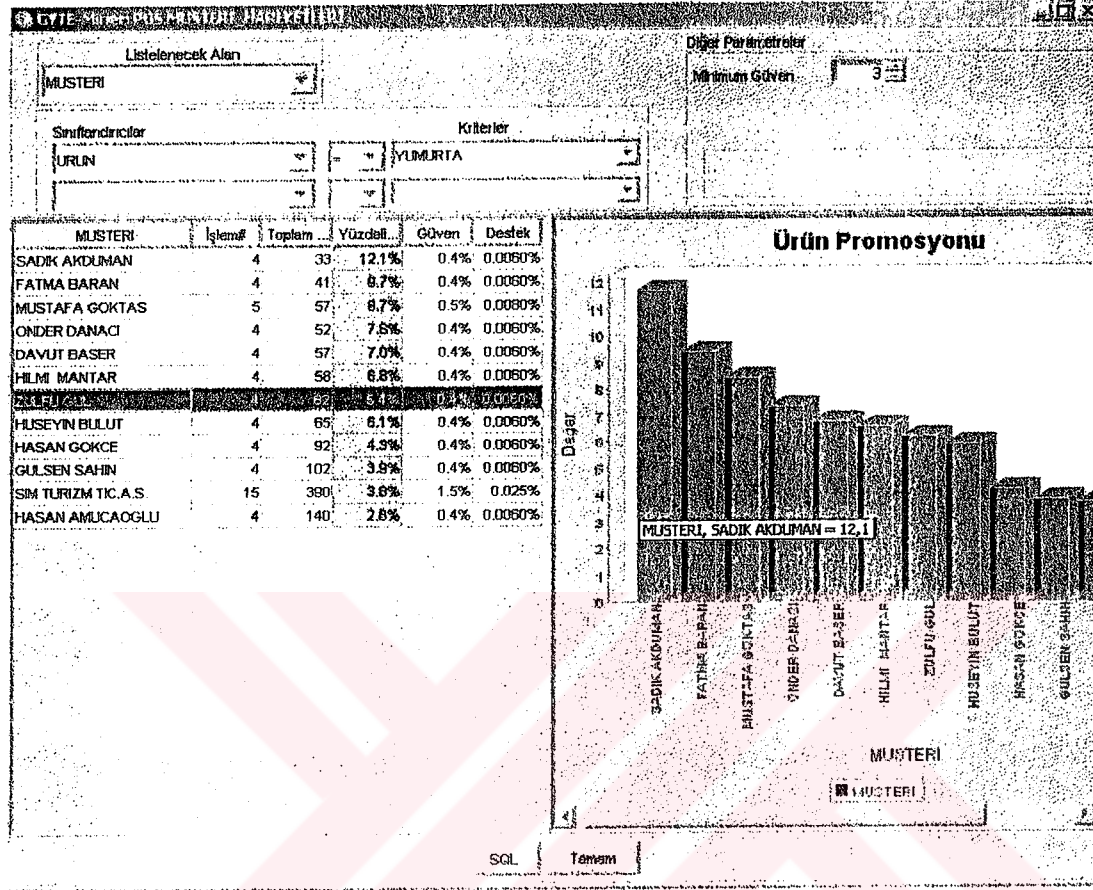
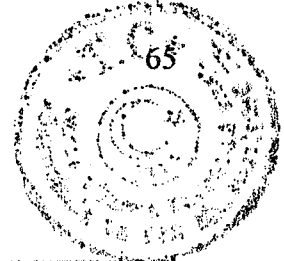
SQL sorgulama penceresi ile kural üretmek ve promosyon hesaplamak diğer yöntemlere göre daha az karmaşık olduğu fakat daha iyi sonuçlar verdiği gözlemlendi. Minimum güven parametresi artırıldığında yukardaki grafiğin daha kesik kesik (geçişler daha seyrek) olduğu gözlemlendi. Şekil 5.3 de bu durum gösterilmiştir.

Buradan belli kişileri belli ürünleri almaktan vazgeçmediği sonucu çıkarılabilir.

Daha başka parametreler eklenerek sistemin çok daha karmaşık sorgulara cevap verebilmesi sağlanabilir.

Uygulamada hem kural üretimi sırasında hem de müşteri listelerinin oluşturulması ve değerlendirilmesi sırasında sistem son kullanıcı ile interaktif bir şekilde çalışmaktadır. Ayrıca hem üretilen kurallarını hem de müşteri listelerini rapor

olarak kaydedilebilir ve bu şekilde eski kuralların arşivleri tutulabilir



Şekil 5.3. SQL Sorgulamasından üretilen Promosyon 2

GYTE-Miner veri madenciliği ve müşteri profili belirleme uygulaması ilerideki çalışmalara iyi bir temel olabilecek nitelikte bir çalışmadır. Uygulama gerek veritabanı açısından gerekse kullanılan yazılım dili ve gerçekleştirme tekniği bakımından oldukça esnektir.

Uygulamada yapılabilecek işlemler;

- Birlikte Kuralları üretme ve grafiksel gösterimi
- Sınıflandırma Kuralları üretme ve grafiksel gösterimi
- Sınıflandırma Kurallarından Karar Ağaçları oluşturma ve grafiksel gösterimi
- Üretilen kurallar içerisinde müşteri-ürün ilişkisine sahip olanları ayırt edilmesi

- Müşteri-ürün ilişkili kurallardan faydalanarak istenilen miktarda uygun müşteri listesi üretme
- Niteliklerin (bu çalışmada daha çok ürünler olarak ele alındı) çeşitli kriterlere göre gruplandırılması ve bu gruplar bazında sınıflandırma ve promosyon yapabilme
- Veritabanından çok çeşitli sayıda istatistik toplayabilme ve grafiklendirme.

Veri madenciliği metodları herbir müşteri için ayrı kurallar üretir, bu metodlar işlem sayısının fazla olduğu uygulamalarda (kredi kartı hareketleri, market otomasyon uygulamaları vs.) gayet iyi sonuç verir. Fakat işlem sayısının çok az olduğu araç satışı, ev, arsa alım-satım işlemlerinde bu metodlar çok iyi sonuç veremez. Çünkü bu tür uygulamalar diğerlerine göre daha az sayıda hareket (transaction) üretirler. GYTE-Miner ilerde bu tür bir veriyapısına sahip organizasyonların da faydalanabileceği şekilde genişletilebilir. Başka bir dezavantaj ise son geliştirilen yöntemde tahmini yöntemler kullanılmamaktadır.

Yaptığımız çalışmanın ilerde veri madenciliği ve müşteri profillendirme projelerinde aktif olarak kullanılacağına inanıyoruz.

## KAYNAKLAR



1. P. Resnick et al., GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Computer-Supported Cooperative Work Conf.*, ACM Press, New York, 1994, pp. 175-186.
2. *Comm. ACM*, Special Issue on Recommender Systems, vol. 40, no. 3, 1997.
3. P. Hagen, Smart Personalization, *The Forrester Report*, Forrester Research, Cambridge, Mass., July 1999.
4. U. Shardanand and P. Maes, Social Information Filtering: Algorithms for Automating 'Word of Mouth,' ACM Press, New York, 1995, pp. 210-217. 5.
5. W. Hill et al., Recommending and Evaluating Choices in a Virtual Community of Use, ACM Press, New York, 1995, pp. 194-201. 6.
6. M. Pazzani, A Framework for Collaborative, Content Based and Demographic Filtering, *Artificial Intelligence Review*, Dec. 1999, pp. 393-408.
7. G. Adomavicius and A. Tuzhilin, Expert-Driven Validation of Rule-Based User Models in Personalization Applications, *J. Data Mining and Knowledge Discovery*, Jan. 2001, pp. 33-58.
8. R. Agrawal et al., Fast Discovery of Association Rules, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, Calif., 1996, chap. 12.
9. L. Breiman et al., *Classification and Regression Trees*, Wadsworth, Belmont, Calif., 1984.
10. G. Adomavicius and A. Tuzhilin, Using Data Mining Methods To Build Customer Profiles, *IEEE Computer* February 2001.
11. G. Piatetsky-Shapiro and C.J. Matheus, The Interestingness of Deviations, AAAI Press, Menlo Park, Calif., 1994, pp. 25-36.
12. A. Silberschatz and A. Tuzhilin, What Makes Patterns Interesting in Knowledge Discovery Systems, *IEEE Trans. Knowledge and Data Engineering*, Dec. 1996, pp. 970-974.
13. R. Srikant, Q. Vu, and R. Agrawal, Mining Association Rules with Item Constraints, AAAI Press, Menlo Park, Calif., 1997, pp. 67-73.
14. Visualizing Data Mining Models, Kurt Thearling et al,  
<http://www3.shore.net/~kht/text/dmviz/modelviz.htm>
15. Data Mining - Finding Business value in Data, Iain McLaren,

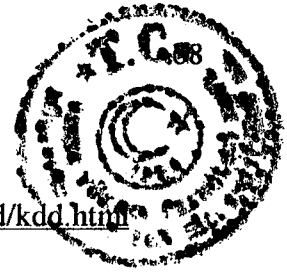
<http://home.clara.net/imclaren/dmpaper.html>

**16. Data Mining and Knowledge Discovery in Databases,**

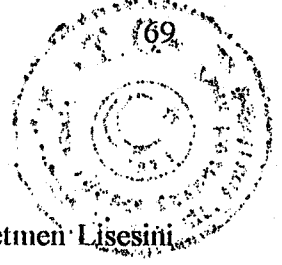
<http://www.cs.sfu.ca/research/groups/DB/sections/publication/kdd/kdd.html>

**17. Java grafik komponentleri**

<http://www.jfree.org>



## ÖZGEÇMİŞ



1977 yılında Erzincan'da doğdu. Erzincan Milliyet Anadolu Öğretmen Lisesini 1995'de bitirdi ve aynı sene Karadeniz Teknik Üniversitesi Müh-Mim Fakültesi Bilgisayar Mühendisliği bölümüne girdi. 1999 yılında lisans eğitimini tamamladı; 2000 yılında Gebze Yüksek Teknoloji Enstitüsünde Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans öğrenimine başladı ve halen devam etmektedir.

Eylül 1999 ve Nisan 2000 tarihleri arasında özel bir yazılım firmasında uygulama geliştirici ve veritabanı analisti olarak çalıştı. 2000'den beri TÜBİTAK Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsünde araştırmacı olarak görevine devam etmektedir.



## EKLER



### EK – 1 Uygulamanın Kaynak Kodu

Uygulamanın kaynak kodu [www.bilmuh.gyte.edu.tr/datamining](http://www.bilmuh.gyte.edu.tr/datamining) adresinden elde edilebilir.

