

BİRLİKTELİK KURALI YÖNTEMİ İÇİN BİR VERİ MADENCİLİĞİ YAZILIMI TASARIMI VE UYGULAMASI

Feridun Cemal ÖZÇAKIR*, A. Yılmaz ÇAMURCU**

ÖZET

Bu çalışmada, bir firmanın pastane satış verileri üzerinde veri madenciliği uygulamak için birliktelik kuralları ile bir yazılım tasarlanmıştır. Veritabanlarında bilgi keşfi sürecindeki işlemler gerçekleştirilmiştir. Veri seçme işlemi ile operasyon veritabanından uygulama veritabanına veriler transfer edilmiştir. Veritabanı içindeki veriler üzerinde veri önileme ve veri indirgeme süreçleri uygulanarak veri madenciliğine uygun veri seti elde edilmiştir. Tasarlanan yazılımda, Apriori algoritması kullanılmıştır. Uygulanan Apriori algoritması ile farklı zaman dilimi, farklı satış lokasyonu girdi değerleri doğrultusunda birlikte satın alınan ürünler ile ilgili bağıntılar olduğu gözlemlenmiştir. Genelde aynı ürün grubuna ait ürünlerin, en sık birlikte satın alınan ürünler olduğu görülmüştür. Yazılımın özel tasarımının sağladığı imkan ile yazılımının çalışması esnasında algoritmanın her aşaması izlenebilmiştir.

Anahtar Sözcükler: *Apriori, Birliktelik Kuralı Madenciliği, Veri Madenciliği, Veri Önileme*

A DATA MINING SOFTWARE DESIGN AND APPLICATION FOR ASSOCIATION RULE TECHNIQUE

ABSTRACT

In this study, a software is designed with association rules to apply data mining on patisserie sales data in a company. Processing steps in Knowledge Discovery from Data are implemented. Data is transferred from database of operation location to application location by data selection process. Appropriate data set for data mining has been got using data preprocessing and data reduction processes on data inside database. Apriori algorithm is used in our software. Each step of algorithm is monitored during software run. Each step of algorithm operation can be monitored for that special software.

Keywords: *Apriori, Association Rule Mining, Data Mining, Data Preprocessing*

* *Palmira Turizm Tic. A.Ş. (Divan İşletmeleri), Bilgi Sistem Yöneticisi, İstanbul*

** *Marmara Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, İstanbul*

1. GİRİŞ

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

Her geçen gün işletmeler ve devlet kurumları veri tabanı sistemine daha fazla yatırım yapmakta ve daha fazla oranda veriyi bu sistemlerde depolamaktadır. Fakat, bu boyuttaki veri çoğu işletme veya kurumda anlamlı ve verimli bir şekilde işlenememektedir. Bu büyük boyuttaki verilerden yararlanmak için, bu veriler üzerinde yöntem ve kurallar uygulanarak değerli bilginin keşfedilmesine gerek duyulmuştur. Bu sürece, Veritabanlarında Bilgi Keşfi VTBK (Knowledge Discovery in Databases – KDD) adı verilmektedir (Fayyad vd., 1996). Veri madenciliği, önceden bilinmeyen, veri içinde gizli, anlamlı ve yararlı örüntülerin büyük ölçekli veritabanlarından otomatik biçimde elde edilmesini sağlayan veri tabanlarında bilgi keşfi süreci içinde bir adımdır (Han ve Kamber, 2006). Günümüzde veri madenciliği; satış, pazarlama, bankacılık (Özekes ve Çamurcu, 2003), tıp, biyoloji, meteoroloji (Bilgin ve Çamurcu, 2004) web kullanımı ve risk analizi gibi bir çok alanda uygulanmaktadır (Tan vd., 2006).

Veri madenciliğinde kullanılan ilk tekniklerden birisi de birliktelik kurallarıdır (Agrawal vd., 1993). Birliktelik kuralı, geçmiş verilerin analiz edilerek bu veriler içindeki birliktelik davranışlarının tespiti ile geleceğe yönelik çalışmalar yapılmasını destekleyen bir yaklaşımdır. Birliktelik kuralı madenciliğinin uygulamasına pazar sepeti analizi örnek verilebilir (Frawley vd., 1991). Birliktelik kuralındaki amaç; alışveriş esnasında müşterilerin satın aldıkları ürünler arasındaki birliktelik ilişkisini bulmak, bu ilişki verisi doğrultusunda müşterilerin satın alma alışkanlıklarını tespit etmektir. Satıcılar, keşfedilen bu birliktelik bağıntıları ve alışkanlıklar sayesinde etkili ve kazançlı pazarlama ve satış imkanına sahip olmaktadır. Örneğin, bir marketten müşterilerin süt ve peynir satın almalarının % 70'inde bu ürünler ile birlikte yoğurt da satın alınmıştır. Bu tür birliktelik örüntüsünün tespit edilebilmesi için, örüntü içinde yer alan ürünlerin birden çok satın alma hareketinde birlikte yer alması gerekir. Milyonlarca veri üzerinde veri madenciliği teknikleri uygulandığında, birliktelik sorgusu için kullanılan algoritmalar hızlı olmalıdır (Agrawal ve Srikant, 1995).

Bu çalışmada¹, veri madenciliği tekniklerinden birliktelik kuralı yöntemlerinden en çok bilinen Apriori algoritmasının örnek bir firmanın pastane satış verileri üzerinde uygulanması için web tabanlı bir uygulama yazılımı geliştirilmiştir (Özçakır, 2006). Geliştirilen uygulama yazılımı ile örnek firmanın yıl içinde farklı dönemlerde ve farklı satış noktalarında gerçekleşen satış verileri üzerinde, birlikte satın alınma tercihi yapılan ürünler tespit edilmektedir.

2. BİRLİKTELİK KURALI ve APRİORİ ALGORİTMASI

2.1. Birliktelik Kuralı

Birliktelik kuralının matematiksel modeli Agrawal, Imielinski ve Swami tarafından 1993 yılında sunulmuştur (Agrawal vd., 1993). Bu modelde, $I = \{i_1, i_2, \dots, i_m\}$ kümesine “ürünler” adı verilmektedir. D , veri bütünlüğündeki tüm hareketleri, T ise ürünlerin her bir hareketini simgeler. TID ise, her harekete ait olan tek belirteçtir.

Birliktelik kuralı şu şekilde tanımlanabilir;

$$A_1, A_2, \dots, A_m \Rightarrow B_1, B_2, \dots, B_n$$

Bu ifadeye yer alan, A_i ve B_j yapılan iş veya nesnelere dir. Bu kural, genellikle “ A_1, A_2, \dots, A_m ” iş veya nesnelere meydana geldiğinde, sık olarak “ B_1, B_2, \dots, B_n ” iş veya nesnelere aynı olay veya hareket içinde yer aldığını belirtir (Zhu, 1998).

Birliktelik kuralı, kullanıcı tarafından minimum değeri belirlenmiş destek ve güvenilirlik eşik değere lerini sağlayacak biçimde üretilir. Bir ürün kümesindeki destek, D ile ifade edilen tüm hareketler içinde ilgili ürün kümesini içeren hareketlerin yüzdesidir. A ve B ürün kümelerinin, birliktelik kuralı “ $A \Rightarrow B$ ” olarak gösterilirse, destek aşağıdaki gibi tanımlanır.

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

$A \Rightarrow B$ birliktelik kuralının güven değeri ise, A ’yı içeren hareketlerin B ’yi de içere me yüzdesidir. Örneğin, bir kural % 85 güvenilirliğe sahip ise, A ’yı içeren ürün kümelerinin % 85’i B ’yi de içermektedir. İşe bağlı veri satırları verilmiş ise, $(A \Rightarrow B)$ güveni aşağıdaki gibi tanımlanır.

¹ Bu çalışma, 2006 yılında Marmara Üniversitesi Fen Bilimleri Enstitüsünde Feridun Cemal ÖZÇAKIR’ın hazırladığı “Müşteri İşlemlerindeki Birlikteliklerin Belirlenmesinde Veri Madenciliği Uygulaması” isimli yüksek lisans tezinden üretilmiştir.

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

Güven değerinin % 100 olması durumunda, kural bütün veri analizlerinde doğrudur ve bu kurallara “kesin” denir.

Birliktelik kuralına ilişkin olarak geliştirilen bazı algoritmalar şunlardır; AIS (Agrawal vd., 1993), SETM (Houtsma ve Swami, 1995), Apriori (Agrawal ve Srikant, 1994), Partition (Savasere vd., 1995), RARM - Rapid Association Rule Mining (Das vd., 2001), CHARM (Zaki ve Hsiao, 2002). Bu algoritmalar içerisinde, ilk olanı AIS, en bilineni ise Apriori algoritmasıdır (Agrawal ve Srikant, 1995).

2.2. Apriori Algoritmasının Yapısı

Apriori Algoritmasının ismi, bilgileri bir önceki adımdan aldığı için “prior” anlamında Apriori’dir (Agrawal ve Srikant, 1994). Bu algoritma temelinde iteratif (tekrarlayan) bir niteliğe sahiptir (Han ve Kamber, 2006) ve hareket bilgileri içeren veritabanlarında sık geçen öğe kümelerinin keşfedilmesinde kullanılır. Apriori Algoritmasına özüne göre, eğer k-öge kümesi (k adet elemana sahip öğe kümesi) minimum destek ölçütünü sağlıyorsa, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar.

Birliktelik kuralı madenciliği, tüm sık geçen öğelerin bulunması ve sık geçen bu öğelerden güçlü birliktelik kurallarının üretilmesi olmak üzere iki aşamalıdır. Birliktelik kuralının ilk aşaması için kullanılan Apriori Algoritması, sık geçen öğeler madenciliğinde kullanılan en popüler ve klasik algoritmadır. Bu algoritmada özellikler ve veri, Boolean ilişki kuralları ile değerlendirilir (Gao, 2004).

k-öge (k adet elemana sahip öğe kümesi) kümesi c ile ifade edilirse, öğeleri (ürünler) $c[1], c[2], c[3], \dots, c[k]$ şeklinde gösterilir ve $c[1] < c[2] < c[3] < \dots < c[k]$ olacak şekilde küçükten büyüğe doğru sıralıdır (Agrawal ve Srikant, 1994). Her öğe kümesine destek ölçütünü tutmak üzere bir sayaç değişkeni eklenmiştir ve sayaç değişkeni öğe kümesi ilk kez oluşturulduğunda sıfırlanır. Geniş (sık geçen) öğe kümeleri L karakteri ile, aday öğe kümeleri ise C karakteri ile gösterilir (Sever ve Oğuz, 2002).

```

1)  $L_1 = \{\text{sık geçen 1-öge kümesi}\};$ 
2) for ( $k=2; L_{k-1} \neq \emptyset; k++$ ) do begin
3)    $C_k = \text{apriori-gen}(L_{k-1});$  // Yeni adaylar
4)   forall transactions-hareketler  $t \in D$  do begin
5)      $C_t = \text{subset}(C_k, t);$  // Adaylar t içindedir
6)     forall candidates – adaylar  $c \in C_t$  do
7)        $c.\text{count}++;$ 
8)   end
9)    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$ 
10) end
11)  $\text{Answer} = \cup_k L_k;$ 

```

Şekil 1. Klasik Apriori Algoritması Özet Kodu

Apriori algoritmasının klasik özet kodu Şekil 1’ de (Agrawal ve Srikant, 1994) görülmektedir. Bu şekilde yer alan *apriori-gen* fonksiyonu (Şekil 2) (Agrawal ve Srikant, 1994), (k-1) adet ögeye sahip L_{k-1} sık geçen öge kümesini kullanarak k adet ögeye sahip aday kümeleri oluşturur. Bu fonksiyon ile, ilk önce, L_{k-1} sık geçen öge kümesine kendisi ile *birleştirme (join)* işlemi uygulanır. Birleştirme işleminde L_{k-1} sık geçen öge kümesinin her satırında yer alan son öge haricinde diğer öğelerin çapraz olarak benzerliği aranır ve son öge haricinde diğer öğelerle yakalanan benzerliklerden yeni aday öge kümeleri oluşturulur. Oluşan kümeler *budama (prune)* adımı ile budanarak fonksiyondan dönlür.

```

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

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

```

Şekil 2. Apriori-Gen Fonksiyonu

Budama işleminde; c aday kümesinin (k-1) ögeye sahip alt kümelerinden L_{k-1} sık geçen öge kümesinde yer almayan tüm alt kümeler silinir. Farklı bir ifade ile budama, C_k aday öge kümesindeki öğelerin alt kümelerinin L_{k-1} sık geçen öge kümesindeki varlığı kontrol edilir, bir ögenin alt kümelerinden biri, L_{k-1} sık geçen öge kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_k aday öge kümesinden silinir (Agrawal ve Srikant, 1994).

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

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

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

3. BİRLİKTELİK KURALLARINA İLİŞKİN YAZILIM TASARIMI

Bu çalışmada, verileri kullanılacak örnek firmanın hizmet noktalarındaki satış verileri üzerinde birliktelik kuralı algoritmalarından Apriori algoritması uygulanmaktadır. Çalışmada, firmanın yedi satış şubesinin 2002 yılı dokuzuncu ayından, 2006 yılı ikinci aya kadarki satış verileri kullanılmıştır.

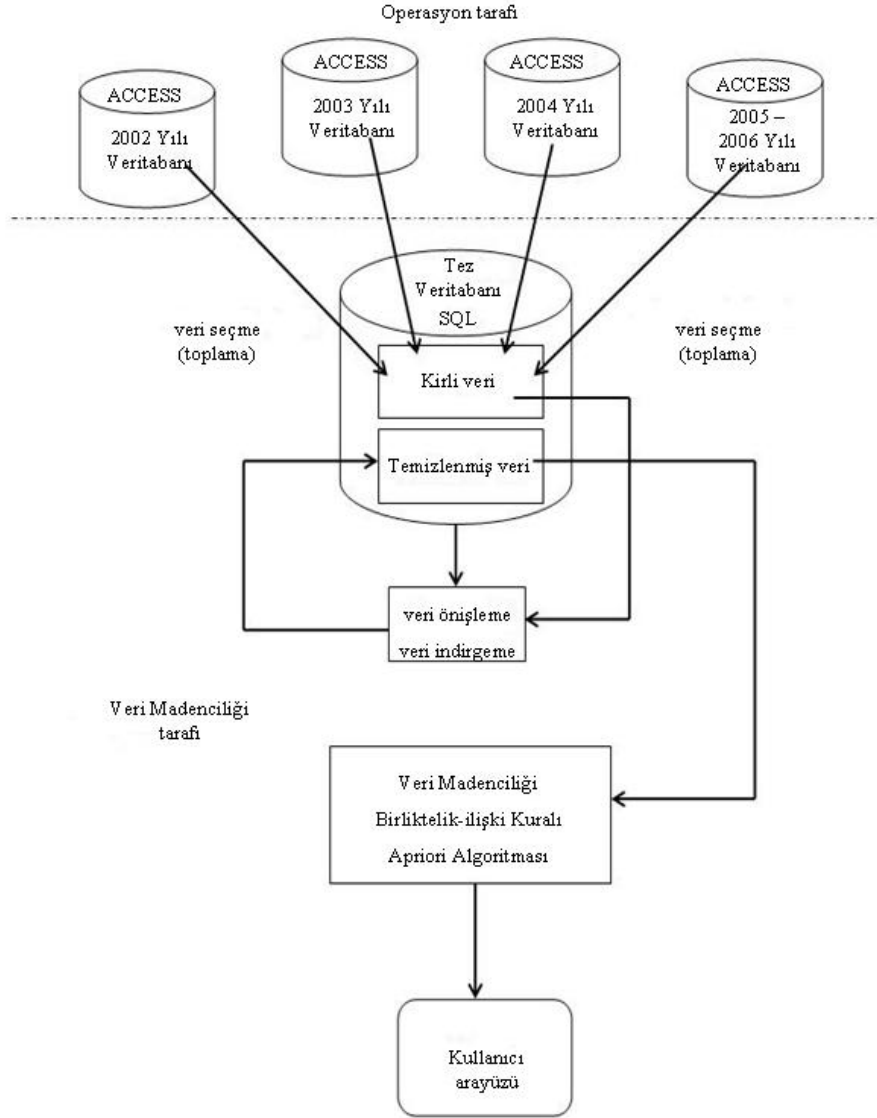
Uygulama yazılımı Microsoft.NET platformunda ASP.NET yapısı üzerinde Microsoft Visual Basic.NET programlama dili kullanılarak web ortamında çalışabilecek şekilde nesne tabanlı olarak geliştirilmiştir. ASP.NET yapısının genel niteliklerinden dolayı uygulama üzerine kurulu olduğu web sunucusu tarafında çalışmaktadır. Kullanıcılar uygulama yazılımına bilgisayarlarında yer alan herhangi bir İnternet tarayıcı program ile web üzerinden erişim sağlamaktadır. Uygulama verileri, SQL 2000 veritabanından okunmakta ve algoritmanın işleyiş adımları doğrultusunda yine SQL 2000 veritabanına geçici tablolar oluşturulmaktadır. Program yazılırken çoklu kullanıcı ortamına uygun programlama teknikleri kullanılmıştır.

Bu çalışmada, öncelikle operasyon tarafındaki veri toplama ve depolama süreçleri ile elde edilen veri için çalışmaya uygun bir veri depolama ortamı belirlenmiştir. Veri depolama ortamı içinde verilerin aktarılacağı tablo yapıları, kullanılacak veri madenciliği yöntemi dikkate alınarak oluşturulmuştur. Operasyon tarafında yer alan veriler, Şekil 3’de görüldüğü gibi veritabanlarında bilgi keşfi aşamalarından veri seçme, veri önileme ve veri indirgeme yöntemleri ile uygulama veri depolama yapısına alınmıştır.

Veri seçimi (veri toplama) işlemi, verilerin veri madenciliği uygulanması için mümkün olduğu kadar tek bir veritabanı veya veri ambarı içinde toplanmasıdır. Yıllar bazında ayrı veritabanlarında tutulan satış tablolarının içeriği, ürün bilgilerinin tutulduğu stok kart tablolarının içerikleri ve her yıla ait satış lokasyonu (şube) bilgileri tek bir veritabanına transfer edilerek veri toplama işlemi gerçekleştirilmiştir.

Veri önileme, verilerin veri madenciliği için hazır duruma getirilmesi için veri üzerinde veri tipi dönüşümü, gruplama, kayıp değerleri yönetme ve gürültülü verinin temizlenmesi gibi işlemlerin uygulandığı aşamadır. Gürültülü veri de veri önileme aşamasında veri kümesinden temizlenir. Gürültülü veri, veri kümesi içinde yer alan ama veri madenciliği uygulamasında kullanılmayacak ve bir anlam içermeyen verilerdir (Han ve Kamber, 2006). Örnek firmanın operasyona ait veritabanlarının içindeki tablolarda yer alan bazı veri alanlarında bulunan verilerin gereksiz olduğu tespit edilmiştir. Bu doğrultuda uygulama veritabanındaki tablolar birliktelik kuralına hizmet edecek şekilde daha az veri alanından (kolon) oluşacak şekilde tasarlanmıştır. Bu uygulama ile veri önileme kapsamında gürültülü verinin temizleme işlemi daha tablolar oluşturulurken gerçekleştirilmiştir.

Tüm tablolarda boş değer (NULL) taraması yapılmış, bazı veri alanlarında boş veriler yer aldığı görülmüştür. Sayısal alanlar “0” (sıfır), karakter veri tipindeki alanlar ise Space(0) değerleri ile doldurulmuştur.



Şekil 3. Gerçekleştirilen Veri Madenciliđi Aşamaları

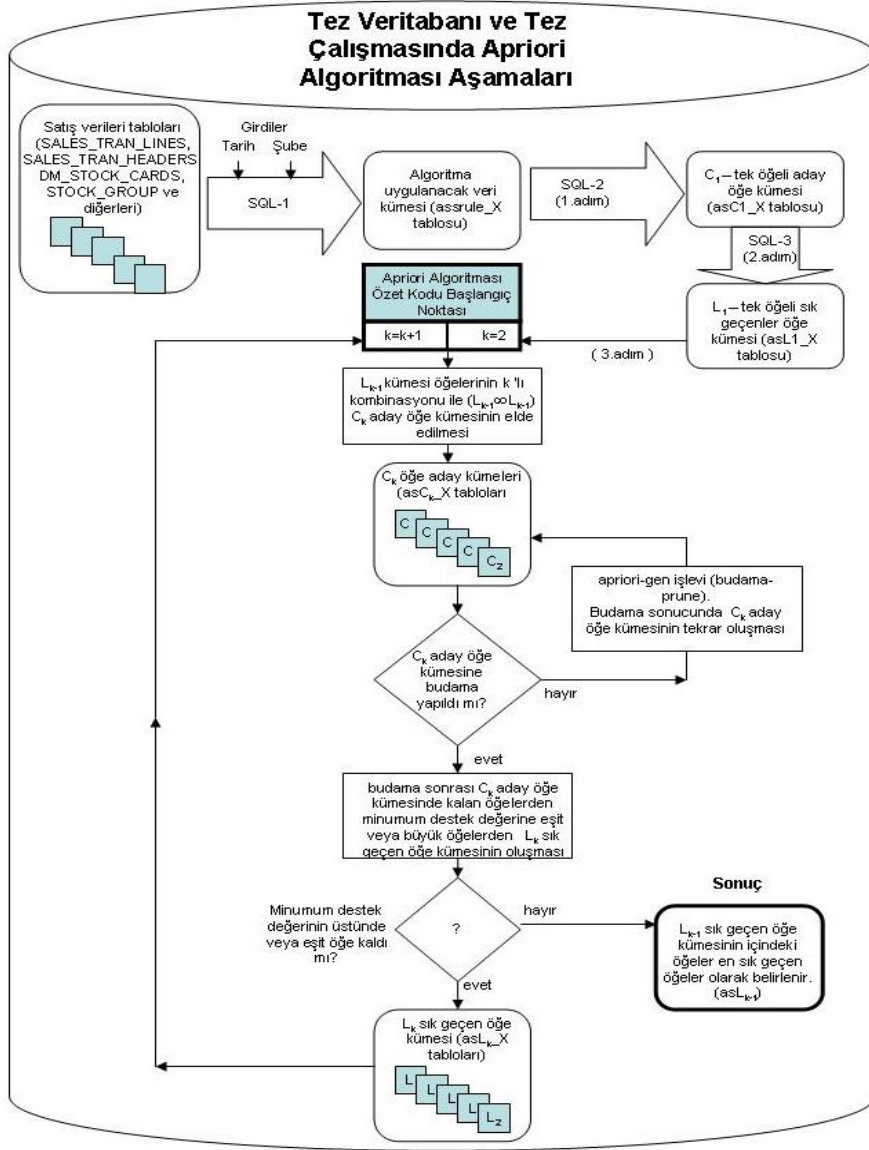
Veri önışleme ve veri indirgeme sürecinden geçirilen veri üzerinde geliştirilen uygulama yazılımı ile Apriori Algoritması uygulanmıştır.

Bu çalışmada Apriori algoritması ile veri madenciliği işlemini gerçekleştirmek için tarafımızca geliştirilen programın çalışma aşamaları Şekil 4’de görülmektedir. Uygulama yazılımında ilk olarak kullanıcı adı ve şifresinin sorulduğu giriş ekranı gelir. Uygulama Microsoft Active Directory yapısı içinde çalışacak ise bu ekrana girilen kullanıcı adı ve şifresi etki alanı (domain) sunucularına LDAP sorguları ile gönderilir, sorgu sonucu olumlu gelirse kullanıcı programa giriş hakkı kazanır. Eğer ki uygulama etki alanı yapısına sahip olmayan bir ortamda kullanılacak ise bu giriş ekranı kullanıcı adı ve şifrelerinin tutulduğu bir veritabanı tablosu içine sorgu gönderir, sorgunun sonucuna göre kullanıcıya giriş hakkı verilir. Giriş hakkı alan kullanıcı “işlem menü” sayfasına ulaşır ve bu sayfadan “Birliktelik-ilişki kuralı” seçeneğini seçerek, çalışma yapılacak yıl bilgisinin belirlendiği ekrana ulaşır. Uygulama 2002 - 2006 yılları arasında çalışmaktadır, her yılın stok kart yapısı ve satış şube bilgileri farklı olduğundan tarih bazlı bu tip gruplama yapılmıştır. Çalışma yapılacak yıl bilgisi seçildikten sonra algoritmanın çalıştırılacağı kriter seçim sayfasına ulaşılır. Kriter sayfasında algoritmanın hangi şubeler için hangi zaman diliminde, hangi tip satışlar üzerinde çalışacağı belirlenir ve algoritmanın çalışması için gerekli tuşa basılır. Bu aşamadan sonra, kriter sayfasında belirlenen kriterler doğrultusunda veritabanından belli satış şubeleri için belli zaman dilimindeki kayıtlardan seçilen veri bütünlüğünde, Şekil 1’deki Apriori algoritmasının özet kodu (Agrawal ve Srikant, 1994) dikkate alınmıştır.

Bu aşamalarda gerçekleştirilen işlemlerin açıklaması aşağıda verilmektedir.

Belirtilen kriterler doğrultusunda ilgili satış kayıtları tarih, satış şubesi, fiş numarası, ürün kodu, birimi, satış miktarı ve dahil olduğu satış grup bilgisi sıralanışı ile oluşturularak geçici bir tabloya aktarılması için SQL sorgusu (Şekil 4’deki SQL-1) kullanılır.

1. Algoritmanın ilk adımında, algoritma basit olarak, tüm tabloyu tarar ve her nesnenin kaç işlemde geçtiği bilgisine erişilir, elde edilen bu veriler veritabanında geçici bir tablo içinde tutulur. Bu geçici tablodaki veriler Apriori algoritmasının C_1 aday öge kümesini temsil eder. Bu veriler ekranda stok kodu, stok adı, işlem sayısı ve dahil olduğu satış grup bilgisi şeklinde görüntülenir. İlgili geçici tablonun oluşturulmasında SQL sorgusu (Şekil 4’deki SQL_2) kullanılır. Bu adıma ilişkin ekran görüntüsü Şekil 5’de görülmektedir. Belirlediğimiz kriterler doğrultusunda işlem yapmak amacı ile veri setinde hareketli öge sayısının 195 olduğu görülmektedir.



Şekil 4. Yazılımda Kullanılan Apriori Algoritmasının Aşamaları

2. İkinci adımda, minimum destek değeri belirlenir. Şekil 5’de görüldüğü gibi minimum destek değeri olarak 4 girilmiştir. Bu minimum destek değeri yardımı ile Apriori algoritmasının en sık geçen tek öğeleri içeren L_1 öge kümesi elde edilir. Diğer bir ifade ile, minimum destek değerine eşit veya yüksek destek değerine sahip olan öğeler L_1 kümesini oluştururlar ve ekranda görüntülenirler. Elde edilen bu veriler veritabanında geçici bir tablo içinde tutulur, tablonun oluşturulmasında SQL sorgusu (Şekil 4’deki SQL_3) kullanılır.

3. Uygulamanın üçüncü adımından itibaren Apriori algoritmasının özet kodunda tanımlanan döngü ve özel işlevler çalıştırılmaya başlar. Üçüncü adımda, L_2 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L_1 kümesinin öğelerinin ikili kombinasyonundan ($L_1 \times L_1$ - birleştirme işlemi) yeni bir aday öge kümesi olan C_2 oluşturulur. Bu işleme *birleştirme işlemi* denir.

4. Apriori algoritmasının özet koduna göre C_2 aday öge kümesi oluştuktan sonra budamanın yapılması gerekir. *Budama* işlemi kısaca şöyle gerçekleşir; C_2 aday öge kümesindeki öğelerin alt kümelerinin L_1 kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_1 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_2 aday öge kümesinden silinir. Fakat özet kodda C_2 için bu işlem yapılması gerekiyor olmasına rağmen C_2 için bu işlem gerçekleştirilmez, çünkü C_2 kümesinin tüm öğelerinin alt kümeleri tek öğeden oluşmaktadır ve L_1 kümesinden birleştirme işlemine uğrayarak C_2 aday öge kümesine gelmişlerdir. Bu nedenden dolayı, uygulamanın dördüncü adımında, C_2 aday kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L_2 sık geçen öge kümesi elde edilir ve ekranda görüntülenir.

5. Uygulamanın beşinci adımında, L_3 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L_2 kümesinin öğelerine birleştirme işlemi uygulandığında ($L_2 \times L_2$ - birleştirme) yeni bir aday öge kümesi olan C_3 oluşturulur. C_3 kümesi elde edilirken veritabanında birden fazla geçici tablo oluşturulur.

6. Altıncı adımda, C_3 aday kümesi üzerinde *apriori-gen* işlemlerinden *budama* işlemi uygulanır. C_3 aday öge kümesindeki öğelerin alt kümelerinin L_2 kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_2 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_3 aday öge kümesinden silinir. Bu aşamada aday kümedeki tüm öğeler budama sonucunda değerlendirme dışı kalabilir ve algoritma sonlanabilir. C_k aday öge kümesinde yapılan budama işleminde algoritma sonlandığında L_{k-1} sık geçen öğeler kümesindeki öğeler en çok birlikte satın alınan ürünler olarak kabul edilir veya tekrardan algoritmayı farklı bir minimum destek değeri belirleyerek çalıştırmak için 1. adıma geri dönüş yapılması gerekmektedir.

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

Min.Destek(Support) Değeri:

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

[Geri](#)

1140.1150.1160, kodlu mağazalar seçili durumdadır, yeni mağaza seçimi için Geri linkini tıklayınız

C1 aday öge kümesi- her nesnenim işlemlerden geçiş sayısını simgeler. C1 kümesinde 195 adet nesne mevcuttur.

SKOD	STKAD	SUPPORT	GRUP
20000	SABLE	335	BİSKÜVİ SATIŞ HESABI
25000	PEYNİRLİ PIZZA	221	TUZLU SATIŞ HESABI
32001	TEK PASTA ŞOKOLALI EKLER	168	TURTA SATIŞ HESABI
11000	PRALİN AMANDA	137	SOKOLA SATIŞ HESABI
30209	TURTA ŞOKOLA KROKANLI (P)	118	TURTA SATIŞ HESABI
30207	TURTA DİVAN SPECIAL (P)	117	TURTA SATIŞ HESABI
69053	SU	113	YIYECEK İÇECEK HES.

Şekil 5. C₁ Aday Öge Kümesi ve Minimum Destek Değeri Giriş Ekranı

7. Yedinci adımda, C₃ aday öge kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L₃ sık geçen öge kümesi elde edilir ve ekranda görüntülenir. C_k aday kümelerinde L_k sık geçen öge kümelerinin elde edilmesi işleminde, C_k aday öge kümesindeki öğelerin tamamı minimum destek değerinin altında kalabilir. Böyle bir durum algoritmanın sonlandığını gösterir ve belirlenen minimum destek değerine göre L_{k-1} kümesinin öğeleri en sık birlikte satın alınan ürünler olarak belirlenir. Şekil 6'da L₃ sık geçen öge kümesinin öge sayısının 12 olduğu görülmektedir.

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

Min.Destek(Support) Değeri:

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

[Geri](#)

1140.1150.1160, kodlu mağazalar seçili durumdadır, yeni mağaza seçimi için Geri linkini tıklayınız

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

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

Şekil 6. L₃ Sık Geçen Öge Kümesi

8. Sekizinci adımda, L_4 sık geçenler öge kümesini belirlemek üzere algoritma doğrultusunda L_3 kümesinin öğelerinin birleştirme işleminden ($L_3 \infty L_3$ - birleştirme) yeni bir aday öge kümesi olan C_4 oluşturulur. Şekil 7’de birleştirme işlemi sonucunda oluşan C_4 kümesinin 1 öğeden oluştuğu görülmektedir.

Başlangıç: Gün: 01, Ay: 10, Yıl: 2005; Bitiş: Gün: 20, Ay: 10, Yıl: 2005
 Rapor tipi: Tüm belge satışları (Fiş, Fatura) Fişli satışlar Faturalı satışlar
 Min.Destek(Support) Değeri: 4

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

[Geri](#)

1140,1150,1160, kodlu mağazalar seçili durumdadır, yeni mağaza seçimi için Geri linkini tıklayınız

C4 aday öge kümesi- L3XL3 birleştirme işlemi kullanarak L3'deki elemanların dörtlü kombinasyonları ile oluşturuldu. C4 kümesinde 1 adet nesne mevcuttur.

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

Şekil 7. C_4 Aday Öge Kümesi

9. Dokuzuncu adım da, C_4 aday kümesi üzerinde budama işlemi uygulanır. C_4 aday öge kümesindeki öğelerin alt kümelerinin L_3 kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_3 kümesinde yer almıyorsa ilgili öge değerlendirme dışı kalır ve C_4 aday öge kümesinden silinir. Budama işlemi sonucunda elenenler de ekran üzerinde kullanıcıya gösterilmektedir. Şekil 8’de budama işlemi sonucunda oluşan C_4 kümesinin bir öğesinin budanmadığı, C_4 aday öge kümesinde 1 öğenin bulunduğu görülmektedir.

Rapor tipi: Tüm belge satışları (Fiş, Fatura) Fişli satışlar Faturalı satışlar
 Min.Destek(Support) Değeri: 4

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

[Geri](#)

1140,1150,1160, kodlu mağazalar seçili durumdadır, yeni mağaza seçimi için Geri linkini tıklayınız

C4 aday öge kümesi- budama(prune) işlemi sonucu kalanlar. C4 kümesinde 1 adet nesne mevcuttur. Sağ daki grid : budama sonucu kümeden çıkarılanlar.

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

Şekil 8. C_4 Aday Öge Kümesi – Budama Sonrası

10. Onuncu adımda, C_4 aday kümesine bakılır ve minimum destek değerine eşit veya yüksek olan değerlerden L_4 sık geçen öge kümesi elde edilir ve ekranda görüntülenir. C_k aday kümelerinde L_k sık geçen öge kümelerinin elde edilmesi esnasında C_k aday kümesindeki hiç bir öge minimum destek değerini geçemeyebilir, bu durumda algoritma sonlanmıştır ve belirlenen minimum destek değerine L_{k-1} kümesinin öğeleri en sık birliktelik satın alınan ürünler olarak belirlenir. Mevcut C_4 aday öge kümesi ile onuncu adım çalışığında Şekil 9'daki sonuç alındı.

Şekil 9. C_4 Aday Öge Kümesinin Öğelerinin Destek Değeri < Min. Destek Değeri

Bu sonuca göre C_4 aday öge kümesinin ögesinin destek değeri minimum destek değerinden küçüktür ve yedinci adımda elde edilen L_3 sık geçen öge kümesi en sık olarak kullanılan nesnelere olarak belirlenir. L_3 sık geçen öge kümesinin 12 elemanı birlikte en sık satın alınan üçlü ürün setleridir.

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

4. SONUÇLAR VE DEĞERLENDİRME

Apriori algoritması uygulama yazılımı, firmanın veri seti üzerinde uygulandığında farklı zaman dilimi, farklı satış şubelerine göre ürünler arasında birlikte satın alınma bağıntıları bulunduğu gözlemlenmiştir. Genelde aynı ürün grubu içinde yer alan ürünlerin birlikte satın alınma davranışlarına sahip oldukları görülmüştür. Bu bağıntılardan bazıları;

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

Uygulama yazılımı web ortamında çalışabilecek şekilde nesne tabanlı olarak geliştirilmiştir. Uygulama aktif olarak kullanıma açıldığında uygulamayı kullanan kullanıcıların bilgisayarlarının İnternet ortamında ve bilgisayarlarında herhangi bir İnternet tarayıcı programı olması yeterlidir. Herhangi bir ek yazılıma veya donanıma ihtiyaç duyulmamaktadır. Yazılımın geliştirilmesi yapılırken çoklu kullanıcı ortamına uygun programlama teknikleri kullanılmıştır, bu doğrultuda aynı anda birden fazla kullanıcı programı web ortamından kullanabilmektedir. Web sunucusu tarafında çalıştırılan uygulamanın sonuçları kullanıcı bilgisayarına bir HTML sayfası olarak gönderilmektedir, bu teknik yapının sağladığı imkan ile kullanıcı bilgisayarının performansı da etkilenmemektedir.

Profesyonel veri madenciliği paket yazılımları yerine firmanın satış verileri üzerinde Apriori algoritmasının bu yazılım ile uygulanması maliyet açısından kazanım sağlamıştır. Ayrıca uygulama yazılımının kodunda yapılan özel tasarımın sağladığı imkan ile algoritmanın her aşaması gözlemlenebilmektedir. Yazılımın sağladığı bu farklı gözlemeleme imkanı algoritmanın işleyişinin anlaşılmasını kolaylaştırmaktadır.

Uygulama yazılımı mevcut durumu ile belirlenen veri üzerinde sadece Apriori algoritmasının uygulanmasını sağlamaktadır. Fakat yazılım geliştirmeye açıktır, yazılım koduna eklenecek modüller ile diğer veri madenciliği algoritmaları da uygulamanın kapsamına alınabilir.

Bu çalışma için gerçek verilerden hazırlanan veritabanındaki satış verileri farklı veri madenciliği algoritmalarının uygulanmasına imkan tanıyacak yapıdadır.

5. KAYNAKÇA

Agrawal, R., Imielinski, T. ve Swami, A., (1993), “Mining Association Rules Between Sets of Items in Large Databases”, In Proceedings of the ACM SIGMOD International Conference on Management of Data (ACMSIGMOD '93), 207-216, Washington, USA, 207-216.

Agrawal, R. ve Srikant, R., (1994), “Fast Algorithms for Mining Association Rules”, In Proceedings of the 20th International Conference on Very Large Databases (VLDB '94), Santiago, Chile, 487-489.

Agrawal R. ve Srikant R., (1995), “Mining Sequential Patterns”, 11th International Conference on Data Engineering, Taipei, Taiwan, 3-14.

Bilgin, T. ve Çamurcu, A. Y., (2004), “A Data Mining Application on Air Temperature Database”, Lecture Notes in Computer Science, 3261, Springer Verlag, ADVIS, 68-76.

Das A., Ng, W. K. ve Woon Y. K., (2001), “Rapid Association Rule Mining”, In Proceedings of the Tenth International Conference on Information and Knowledge Management, ACM Press, Atlanta, GA, USA, 487-499.

Fayyad, U., Piatetsky-Shapiro, G. ve Smyth, P., (1996), “The KDD Process for Extracting Useful Knowledge From Volumes of Data”, Communications of ACM, 39, 11, 27-34.

Frawley, W. J., Piatetsky-Shapiro, G. ve Matheus, C. J., (1991), “Knowledge Discovery Databases: An Overview, in Knowledge Discovery in Databases”, AAAI 58 AI Magazine, Cambridge, 1-27.

Gao, W., (2004), “A Hierarchical Document Clustering Algorithm”, MSc Thesis, Dalhousie University, Halifax, Nova Scotia.

Han, J. ve Kamber, M., (2006), Data Mining Concepts and Techniques, Morgan Kauffmann Publishers Inc., 1-35.

Houtsma, M. ve Swami, A, (1995), “Set-Oriented Mining for Association Rules in Relational Databases”, Proceedings of the 11th IEEE International Conference on Data Engineering, Taipei, Taiwan, 25-34.

Özçakır, F. C., (2006), “Müşteri İşlemlerindeki Birlikteliklerin Belirlenmesinde Veri Madenciliği Uygulaması”, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Özekes S. ve Çamurcu Y., (2003), “Veri Madenciliğinde Karar Ağaçları Yöntemi Uygulaması”, Bilgi Teknolojileri Kongresi II, Pamukkale Üniversitesi, Denizli.

Tan, P. N., Steinbach, M. ve Kumar, V., (2006), *Introduction to Data Mining*, Addison Wesley.

Savesere, A., Omiecinski, E. ve Navathe, S., (1995), “An Efficient Algorithm for Mining Association Rules in Large Databases”, In *Proceedings of 20th International Conference on VLDB*, 432-444.

Sever, H. ve Oğuz, B., (2002), “Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım, Kısım 1: Eşleştirme Sorguları ve Algoritmalar”, *Bilgi Dünyası*, 3, 2, Ekim, 173-204.

Zaki, M. J. ve Hsiao, C. J., (2002), “CHARM: An Efficient Algorithm for Closed Itemset Mining”, In *2nd SIAM International Conference on Data Mining (SDM'02)*, Eds. Grossman, R. L., Han, J., Kumar, V., Mannila, H. ve Motwani, R., Siam, Arlington, VA, USA, 457-473.

Zhu, H., (1998), “On-Line Analytical Mining of Association Rules”, MSc. Thesis, Simon Fraser University, Ottawa, Canada.