T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ İLE BİRLİKTELİK KURALLARININ BULUNMASI

YÜKSEK LİSANS TEZİ

Bil. Müh. Fatih ŞEN

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Tez Danışmanı : Yrd. Doç. Dr. Nilüfer YURTAY

Eylül 2008

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ İLE BİRLİKTELİK KURALLARININ BULUNMASI

YÜKSEK LİSANS TEZİ

Bil.Müh. Fatih ŞEN

Enstitü Anabilim Dalı : **BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**

Tez Danışmanı : **Yrd.Doç. Dr. Nilüfer YURTAY**

Bu tez 05/09/2008 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Prof. Dr. Emin GÜNDOĞAR	Yrd. Doç. Dr. Nilüfer YURTAY	Yrd. Doç. Dr. Kürşat AYAN
.....	.....	.....
Jüri Başkanı	Üye	Üye

TEŞEKKÜR

Bu tez çalışmasında, bana rehberlik eden , kısıtlı zaman içerisinde olumlu yaklaşımları ile sürekli teşvik eden ve desteğini esirgemeyen tez danışmanım Sayın Yrd. Doç. Dr Nilüfer YURTAY'a içtenlikle teşekkür ederim.

Ayrıca çalışmalarımda kullandığım verilere ulaşmamda yardımcı olan GÜN-BAK Yönetim Kurulu üyesi Sayın Kamil Bulut'a ve GÜN-BAK Bilgi İşlem Dairesi çalışanlarına, hayatımın her safhasında desteklerini esirgemeyen aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

VTBK	: Veritabanlarında Bilgi Keşfi
VM	: Veri Madenciliği
L_k	: Sık geçen k adet öğeli veri setleri
C_k	: K adetli sık geçen aday veri setleri
$L_k \infty L_k$	: K öğeli veri setlerinin kombinasyonları
min_sup	: Minimum destek değeri
min_conf	: Minimum güven değeri
$X \Rightarrow Y$	: X ürünün bulunduğu satışlarda Y ürünün de bulunması olayı
PHP	: Personal Home Pages
SQL	: Structured Query Language

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Veri madenciliği, birliktelik kuralları, apriori algoritması

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

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

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

Uygulama bölümünde, gerçek veriler kullanarak Birliktelik Kuralları yöntemi ile Pazar Sepeti Çözümlemesi uygulaması yapılmış ve elde edilen sonuçlar tartışılmıştır. Çalışmanın amacı; Veritabanlarında Bilgi Keşfi, Veri Madenciliği ve Birliktelik Kuralları'nı ayrıntılı olarak incelemek, veri madenciliğinde istatistiksel çözümlemeye ağırlık vererek bir pazar sepeti çözümlemesi uygulaması gerçekleştirip sonuçları değerlendirmektir.

ASSOCIATION RULES FINDING WITH DATA MINING

SUMMARY

Key Words: Data Mining, Association Rules, Apriori Algorithm

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

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

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

In the application, by using real data, market basket analysis application has performed by association rules and the results have been discussed. The aim of the study is to analyze knowledge discovery in databases, data mining and association rules, to carry out a market basket analysis by emphasizing on statistical analysis and to evaluate the results of the application.

BÖLÜM 1. GİRİŞ

Günümüzde işletmelerin yoğun teknoloji ve bilgisayar kullanımlarının artmasıyla birlikte müşteri verileri elektronik ortamda tutulmaya başlanmış, elektronik veri saklama ve analiz araçlarının gelişimiyle de büyük miktarlardaki veriyi işleme yeteneğine sahip teknolojilere gereksinim duyulmuştur.

Bilgisayarlarda, bilgisayar ağlarında çok yüksek boyutlarda verilerin saklandığı günümüzde, kamu kurumları, bilim kuruluşları ve şirketler veri toplama ve saklama işlemleri için oldukça büyük miktarlarda parasal kaynak kullanmaktadırlar. Toplanan verilerin hacimlerinin çok büyük olması ve yapılarının da etkin bir veri analizi yapılmasına uygun olmaması nedeniyle uygulamalarda bu verilerin ancak çok küçük bir kısmının kullanılabilmesine neden olmaktadır.

Rekabetin yoğun yaşandığı iş sektörleri öncelikle sahip oldukları müşterileri rakip firmalara kaçırmamayı, daha sonra da müşteri potansiyellerini arttırmayı amaçlamaktadırlar. Bu sebeple müşterileri mümkün olduğu kadar fazla tanımak amacıyla, müşterilere ait bilgileri elektronik ortamlarda kayıt altına almışlar, bu verilerden anlamlı bilgilere ulaşmayı hedeflemişlerdir. Örneğin eskiden süper marketlerdeki kasalar basit bir toplama makinesinden oluşmaktaydı. Müşterinin o anda satın almış olduğu malların toplamını hesaplamak için kullanılırdı. Günümüzde ise kasa yerine kullanılan satış noktası terminalleri sayesinde yapılan satışın bütün detayları saklanabilmektedir. Saklanan bu binlerce ürün ve müşteri hareket bilgileri sayesinde her malın zaman içindeki satış hareketleri ve eğer müşteriler bir müşteri numarası ile kodlanmışsa herhangi bir müşterinin zaman içindeki verilerine ulaşmak ve analiz etmek mümkün olabilmektedir.

Veri madenciliği; eldeki verilerden üstü kapalı, net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır [1]. Diğer bir deyişle veri madenciliği, büyük veri yığınlarından anlamlı bilgiler elde etmek için, bilgisayar destekli bir bilgi çözümleme işlemidir.

Birliktelik- ilişki kuralları da veritabanındaki fark edilmeyen bilgilerden işe yarar tutarlı bilgiler elde etmeyi sağlayan veri madenciliği modellerinden bir tanesidir. Birliktelik-ilişki kuralları, hareket verileri içinde birlikte hareket eden öğelerin keşfedilmesini, keşfedilen bu bağıntılar ile geleceğe yönelik tahminler üretilmesini sağlamaktadır.

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

Bu çalışmada, veritabanlarında bilgi keşfi süreçleri, veri madenciliği, veri madenciliğinde kullanılan birliktelik-ilişki kuralları ele alınmış ve bu kurallardan Apriori algoritması ile bir uygulama geliştirilmeye çalışılmıştır.

BÖLÜM 2. VERİ MADENCİLİĞİNE GENEL BAKIŞ

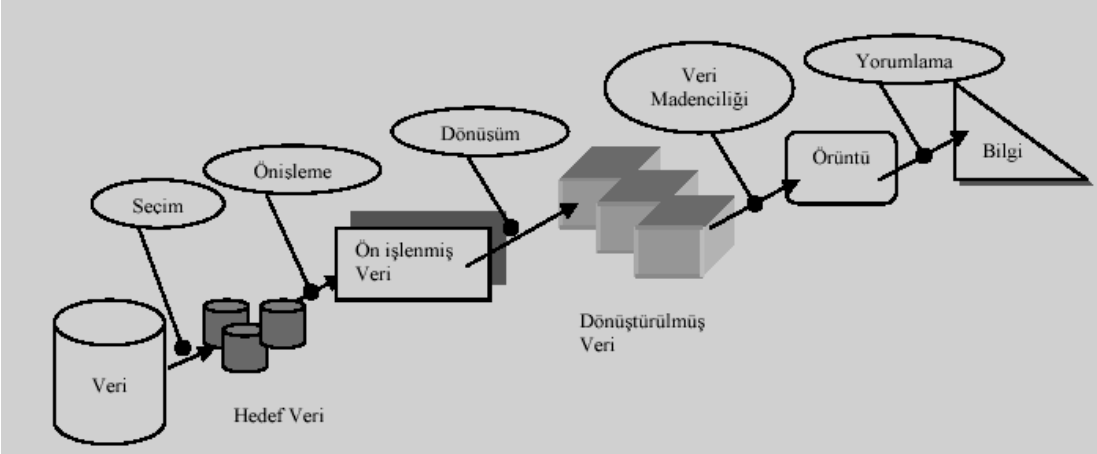
2.1. Veritabanlarında Bilgi Keşfi

Veri kendi başına bir değer ifade etmez, bir gayeye yönelik olarak işlendiğinde bilgi meydana gelir. Veriyi bilgiye çevirme süreci veri analizi olarak nitelendirilir. Yakın geleceğin, günümüzden çok fazla farklı olmayacağı düşünüldüğünde, geçmiş ve günümüzdeki verilerden çıkarılmış olan bilgiler yakın gelecekte de geçerli olacak ve gelecek için doğru tahmin yapmayı sağlayacaktır.

Kayıtlı verilerden anlamlı bilgilere ulaşım sürecine Veritabanlarında Bilgi Keşfi (VTBK) olarak nitelendirilmektedir. Veritabanlarında bilgi keşfi, depolanmış veri içerisindeki geçerli, yeni, faydalı ve sonuç olarak anlaşılabilir örüntülerin çıkarılması sürecidir. Bu sürecin ilk adımı, uygulama alanının öğrenilmesi ile başlar. Veritabanlarında bilgi keşfinin son basamağı ise, elde edilen bilginin görüntüleme ve bilgi gösterimi yöntemleri kullanılarak kullanıcıya sunulması şeklindedir. Bazı araştırmacılar veritabanlarında bilgi keşfi ile Veri Madenciliğini eşanlamlı olarak kabul etmelerine rağmen, genel görüş veri madenciliği VTBK sürecinin bir aşaması şeklindedir.

Veri madenciliği; eldeki verilerden üstü kapalı, net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır [1]. Diğer bir deyişle veri madenciliği, büyük veri yığınlarından anlamlı bilgiler elde etmek için, bilgisayar destekli bir bilgi çözümleme işlemidir.

Şekil 1.1 Veri madenciliğinin veri işleme süreci içerisindeki yeri göstermektedir[2].



Şekil 2.1 Veri Madenciliğinin Veri İşleme Süreci İçerisindeki Yeri

Han ve Kamber'e göre veri madenciliği, büyük veri yığınları içerisinden gelecek ile ilgili tahminde bulunabilmemizi sağlayabilecek bağıntıların bilgisayar programı kullanarak aranmasıdır. Han ve Kamber'e göre de veri madenciliği veritabanlarında bilgi keşfinin bir adımını simgelemektedir.

Veri madenciliği yapılabilmesi için, veritabanlarında bilgi keşfi süreçlerinin veritabanlarında tutulan verilere sıra ile uygulanması gerekmektedir. Her bir süreç tamamlandıktan sonra bir sonraki sürecin başlatılarak veri madenciliği aşamasına ulaşılmalıdır. Veri madenciliği aşamasında veri madenciliği tekniklerinden verilere ve elde edilmek istenen sonuca uygun olan teknik seçilerek uygulanır.

2.1.1. Veritabanlarında bilgi keşfi aşamaları

Veriden bilgiye ulaşım sürecindeki VTBK aşamaları şu şekildedir:

- Veri Seçimi (Data Selection): Bu aşamada birden fazla veri kümesi içerisinden, üzerinde sorgu yapılmasına uygun örnek bir veri kümesi oluşturma aşamasıdır. Veri toplama (data collection) ve farklı kümelerdeki verilerin birleştirilmesi işlemi de bu süreçte yer alır. Toplama, tanımlanan problem için gerekli olduğu düşünülen verilerin ve bu verilerin toplanacağı veri kaynaklarının belirlenmesi adımıdır.

Veri seçimi aşamasında yapılması gerekenler;

1. Farklı ortamlardaki verilerin mevcut yapılarının incelenmesi ve tablo yapılarının ortaya çıkarılması,
2. Veri madenciliği ile hedeflenen sonuca ulaşmak için gerekli verilerin, uygulama için belirlenen veri depolama ortamına aktarılması olarak sıralanabilir.

- Veri Önişleme (Data Preprocessing): Veri seçimi ile elde edilen örnek veri kümesinde yer alan hatalı ve eksik değerlerin düzenlendiği ve çıkarıldığı aşamadır. Veri temizleme (data cleaning) ve veri dönüştürme (data transformation) veri önişleme işlemleridir. Veri temizlemenin amacı gürültülü ve ilgisiz verinin veri setinden çıkarmaktır. Veri dönüşümünün amacı ise, kaynak veri içindeki farklı biçimdeki veri tip ve değerlerini yapılacak veri madenciliği çalışması doğrultusunda değiştirmektir.

Modelde kullanılan veritabanının çok büyük olması durumunda örnekleme yapılması uygun olabilir. Günümüzde hesaplama olanakları ne kadar gelişmiş olursa olsun, çok büyük veritabanları üzerinde çok sayıda modelin denenmesi uzun zaman alması nedeni ile mümkün olamamaktadır. Bu nedenle tüm veritabanını kullanarak bir kaç model denemek yerine, rasgele örneklenmiş bir veritabanı parçası üzerinde birçok modelin denenmesi ve bunlar arasından en güvenilir ve güçlü modelin seçilmesi daha uygun olacaktır.

Veri tipi dönüşümü , basit olarak veri tipi değişimidir. Örnek olarak, integer tipteki bir veriyi boolean tipine dönüştürme işlemi verilebilir. Bu dönüşümün sonucunda, sorgulama yapılacak veri tabanı boyutu azaltılabilir ve sorgularda hız artışı sağlanabilir.

Bazı veritabanlarında bir kolon içinde sürekli tekrarlayan benzer veriler bulunmaktadır. Bu verileri bir kaç grup içine yerleştirme işlemi uygulanarak verinin kalitesi artırılır. Gruplama tekniği ile yorumlamanın daha kolay olması sağlanabilir.

Farklı veritabanlarından gelen veriler tek bir tablo içinde birleştirildiğinde veri alanlarının bazıları boş kalabilir. Bu durumu düzeltmek için, kayıp değerler en çok kullanılan değerler ile doldurabilir, bir kayıtta çok fazla kayıp değer varsa kayıt tamamen silinebilir, en olası ortalama değer ile doldurulabilir.

- Veri İndirgeme (Data Reduction): Seçilen örnek veri kümesindeki ilgisiz nitelikte ve tekrarlı verilerin çıkarıldığı aşamadır. Bu işlem ile verinin boyutu indirgendiğinden veri madenciliği uygulanırken çalıştırılacak sorguların daha hızlı sonuç üretmeleri sağlanır.

- Veri Madenciliği (Data Mining): Bu aşama veri madenciliği yöntemlerinin ve algoritmalarının uygulandığı adımdır. VM; veritabanı sistemleri, verilerin depolanması, istatistik, makine öğrenimi gibi alanların kombinasyonundan oluşan disiplinler arası bir yöntemdir. VM istatistikçiler için yeni bir konu değildir. İstatistik ve VM ortak amaçlara sahiptir, her ikisi de verilerin yapılarının keşfedilmesiyle ilgilidir. Her ne kadar VM istatistiğin bir alt kümesi olarak kabul edilse de VM, veritabanı teknolojisi ve makine öğrenimi gibi diğer alanlara ait fikirleri, araçları ve yöntemleri de kullanır [3].

- Değerlendirme (Evaluation): Bilgi keşfi sürecinde bu aşamadan önceki aşamalar sonucunda elde edilen bilginin geç erlilik, yenilik, yararlılık ve basitlik kıstaslarına göre değerlendirilmesi aşamasıdır [4].

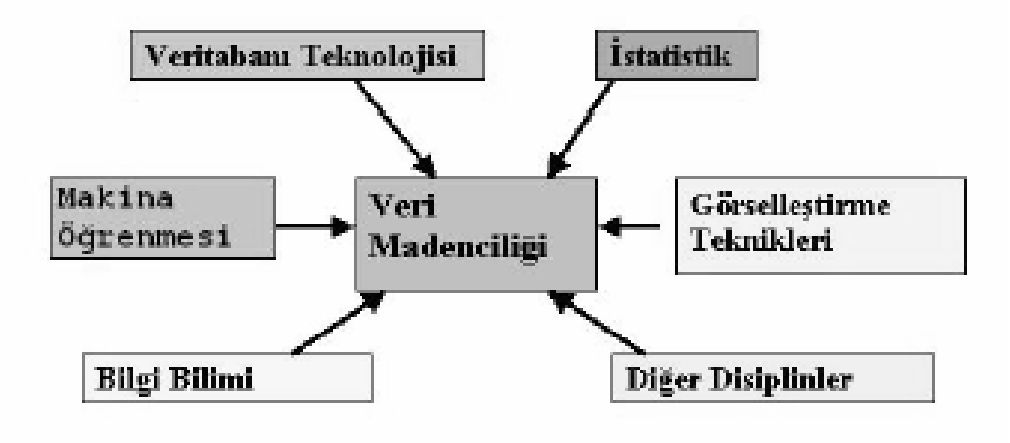
2.2. Veri Madenciliği

Veri madenciliği, önceden bilinmeyen ilişki ve trendlerin bulunması için bugünün endüstrisinde yaratılan büyük miktarlardaki veriyi analiz eden bir yoldur. Yüksek güçlü bilgisayarlara ve gereken yazılımlara kolay ve düşük fiyatlarla ulaşılabilmesi bu teknolojinin işlemesini olanaklı kılmıştır.

Gartner Grup tarafından yapılan tanımda veri madenciliği, istatistik ve matematik tekniklerle birlikte ilişki tanıma teknolojilerini kullanarak, depolama ortamlarında saklanmış bulunan veri yığınlarının elenmesi ile anlamlı yeni ilişki ve eğilimlerin keşfedilmesi sürecidir[5].

VM aracılığıyla, büyük veri kümelerinden oluşan veritabanı sistemleri içerisinde gizli kalmış bilgilerin çekilmesi sağlanır. Bu işlem, istatistik, matematik disiplinleri, modelleme teknikleri, veritabanı teknolojisi ve çeşitli bilgisayar programları kullanılarak yapılır.

Makine öğrenimi, istatistik ve VM arasındaki yakın bir bağ vardır. Bu üç disiplin veri içindeki örüntüleri bulmayı amaçlar. Makine öğrenimi yöntemleri, VM algoritmalarında kullanılan yöntemlerin çekirdeğini oluşturur. Makine öğreniminde kullanılan karar ağacı, kural çıkartımı pek çok VM algoritmasında kullanılmaktadır. Makine öğrenimi ile VM arasında benzerliklerin yanı sıra farklılıklar da göze çarpmaktadır. Öncelikle VM algoritmalarında kullanılan örnekleme boyutu, makine öğreniminde kullanılan veri boyutuna nazaran çok büyüktür.



Şekil 2.2 Veri Madenciliğinin Farklı Disiplinlerle İlişkisi

2.3. Veri Madenciliğinin Kullanım Alanları

Günümüzde VM teknikleri başta isletmeler olmak üzere çeşitli alanlarda basarı ile kullanılmaktadır. Aşağıda veri madenciliği kullanımı yapılabilecek birkaç örnek verilmiştir.

- İşletme kendi müşterisiyken rakibine giden müşterilerle ilgili analizler yaparak rakiplerini tercih eden müşterilerinin özelliklerini elde edebilir ve bundan yola çıkarak gelecek dönemlerde kaybetme olasılığı olan müşterilerin kimler olabileceği yolunda tahminlerde bulunarak onları kaybetmemek, kaybettiklerini geri kazanmak için strateji geliştirebilir.

- Ürün veya hizmette hangi özelliklerin ne derecede müşteri memnuniyetini etkilediği, hangi özelliklerinden dolayı müşterinin bunları tercih ettiği otaya çıkarılabilir.

- Kredi kartı ödemelerini aksatan, gecikmeli olarak yapan veya hiç yapmayanların özelliklerinden yola çıkılarak bundan sonra aynı duruma düşebilecek muhtemel kişiler saptanabilir.

- Bir ürün veya hizmetle ilgili bir kampanya programı oluşturmak için hedef kitlenin seçiminden başlayarak bunun hedef kitleye hangi kanallardan sunulacağı kararına kadar olan süreçte veri madenciliği kullanılabilir.

Veri madenciliğinin uygulama alanları konu başlıkları itibariyle aşağıdaki gibi sınıflandırılabilir[6].

Pazarlama

- Müşterilerin satın alma örüntülerinin belirlenmesi
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması
- Posta kampanyalarında cevap verme oranının artırılması
- Pazar sepeti analizi
- Müşteri ilişkileri yönetimi
- Müşteri değerlendirme
- Satış tahmini
- Müşteri dağılımında
- Çeşitli pazarlama kampanyalarında
- Mevcut müşterilerin elde tutulması için geliştirilecek pazarlama stratejilerinin oluşturulmasında
- Çapraz satış analizleri
- Çeşitli müşteri analizlerinde

Bankacılık

- Farklı finanssal göstergeler arasında gizli korelasyonların bulunması
- Kredi kartı dolandırıcılıklarının tespiti
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi
- Kredi taleplerinin değerlendirilmesi.
- Müşteri dağılımında
- Usulsüzlük tespiti
- Risk analizleri

Sigortacılık

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

Perakendecilik

- Satış noktası veri analizleri
- Alış-veriş sepeti analizleri
- Tedarik ve mağaza yerleşim optimizasyonu
- Hisse senedi fiyat tahmini
- Genel piyasa analizleri
- Alım-satım stratejilerinin optimizasyonu

Telekomünikasyon

- Kalite ve iyileştirme analizleri
- Hisse tespitleri
- Hatların yoğunluk tahminleri

Sağlık ve İlaç

- Test sonuçlarının tahmini
- Ürün geliştirme
- Tıbbi teşhis
- Tedavi sürecinin belirlenmesi
- Semptomlara göre hastalık tespiti,

Endüstri

- Kalite kontrol analizleri
- Lojistik
- Üretim süreçlerinin optimizasyonu

Tablo 2.1'de 2003 yılında veri madenciliğinin sektörler bazında kullanımına ilişkin bir araştırmanın sonuçları yer almaktadır [7].

Tablo 2.1 Veri madenciliğinin uygulandığı alanların dağılımı

Bankacılık (51)	12%
Bioteknoloji / Genetik (11)	3%
Kredi skorlama (35)	8%
CRM (52)	12%
Doğrudan pazarlama (34)	8%
e-Ticaret (11)	3%
Eğlence/ Müzik (4)	1%
Sahtekarlık tespiti (31)	7%
Şans oyunu (2)	0,01 %
Kamu uygulamaları (12)	3%
Sigortacılık (24)	6%
Yatırım / Hisse senedi (5)	1%
Junk email / Anti-spam (5)	1%
Sağlık/ İK (15)	4%
İmalat (19)	5%
Tıp/ Farmakoloji (12)	3%
Perakende (25)	6%
Bilim (17)	4%
Güvenlik / Anti-terörizm(5)	1%
Telekomünikasyon (23)	5%
Seyahat (8)	2%
Web (9)	2%
Diğer (11)	3%

2.4. Veri Madenciliğinde Karşılaşılan Zorluklar

Veri madenciliği girdi olarak kullanılacak ham veriyi veritabanlarından alır. Bu da veritabanlarının dinamik, eksiksiz, geniş ve net veri içermemesi durumunda sorunlar doğurur [8]. Küçük veri kümelerinde hızlı ve doğru bir biçimde çalışan bir sistem, çok büyük veri tabanlarına uygulandığında tamamen farklı davranabilir. Bir VM sistemi tutarlı veri üzerinde mükemmel çalışırken, aynı veriye gürültü eklendiğinde kayda değer bir biçimde kötüleşebilir. Günümüzde VM sistemlerinin karşılaştığı sorunlar şu şekildedir:

2.4.1. Veri tabanı boyutu

Veri tabanı boyutları inanılmaz bir hızla artmaktadır. Pek çok makine öğrenimi algoritması birkaç yüz tutanaklık oldukça küçük örneklemeleri ele alabilecek biçimde geliştirilmiştir. Örneklemenin büyük olması, örüntülerin gerçekten var olduğunu göstermesi açısından bir avantajdır ancak böyle bir örneklemeden elde edilebilecek olası örüntü sayısı da çok büyüktür. Bu yüzden VM sistemlerinin karşı karşıya olduğu en önemli sorunlardan biri veri tabanı boyutunun çok büyük olmasıdır. Dolayısıyla VM yöntemleri ya sezgisel bir yaklaşımla arama uzayını taramalıdır, ya da örneklemeyi yatay/dikey olarak indirgemelidir. Yatayda indirgeme veri alanının örneklenmesi, dikeyde indirgeme ise özelliklerin bulunduğu kolonların azaltılma çalışmasıdır.

2.4.2. Gürültü

Büyük veri tabanlarında pek çok niteliğin değeri yanlış olabilir. Bu hata, veri girişi sırasında yapılan insan hataları veya girilen değerin yanlış ölçülmesinden kaynaklanır. Veri girişi veya veri toplanması sırasında oluşan sistem dışı hatalara gürültü adı verilir. Günümüzde kullanılan ticari ilişkisel veri tabanları, veri girişi sırasında oluşan hataları otomatik biçimde gidermek konusunda az bir destek sağlamaktadır. Hatalı veri gerçek dünya veri tabanlarında ciddi problem oluşturabilir. Bu durum, bir VM yönteminin kullanılan veri kümesinde bulunan gürültülü verilere karşı daha az duyarlı olmasını gerektirir.[9]

2.4.3. Eksik ve artık veriler

Verilen veri kümesi, eldeki probleme uygun olmayan veya artık nitelikler içerebilir. Bir değer bilinmiyor ya da yanlışlıkla girilmemiş olabilir. Veri madenciliğindeki birçok yöntem, her veri nesnesi için sabit bir boyut (özellik sayısı) gerektirdiğinden, eksik veriler sorun yaratır. Artık veri oluşumunu engellemek için özellik seçimi

yapılmalıdır. Özellik seçimi yalnızca arama uzayını küçültmekle kalmayıp, sınıflama işleminin kalitesini de artırır.

2.4.4. Dinamik veri yapısı

Çevrim içi veri tabanları dinamiktir, yani içeriği sürekli olarak değişir. Bu durum, bilgi keşfi metotları için önemli sakıncalar doğurmaktadır. İlk olarak sadece okuma yapan ve uzun süre çalışan bilgi keşfi metodu, bir veri tabanı uygulaması olarak mevcut veri tabanı ile birlikte çalıştırıldığında mevcut uygulamanın da performansı ciddi ölçüde düşer. Diğer bir sakınca ise, veri tabanında bulunan verilerin kalıcı olduğu varsayılıp, çevrim dışı veri üzerinde bilgi keşif metodu çalıştırıldığında, değişen verinin elde edilen örüntülere yansıması gerekmektedir. Burada 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[10].

2.5. Veri Madenciliği Modelleri ve Kullanılan Algoritmalar

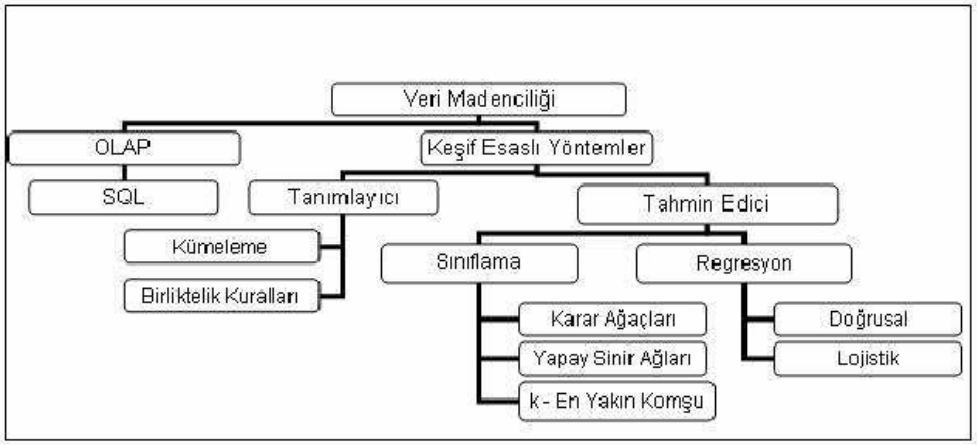
Veri madenciliğinde kullanılan modeller, tahmin edici ve tanımlayıcı olmak üzere iki ana baslık altında incelenmektedir.

Tahmin edici modellerde, keşfe dayalı modellerdir. Sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır [6] .Sınıflama (classification), gerileme (regression) ve sapma (deviation) madenciliği tahmin edici tekniklerden bazılarıdır.

Tanımlayıcı modellerde ise karar vermeye yardım edebilecek, mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır. X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu aileler ile, çocuğu olmayan ve geliri X/Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerlik gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir . Kümeleme (clustering), birliktelik kuralı (association rule) ve ardışık örüntü (sequential pattern) madenciliği tanımlayıcı tekniklerden bazılarıdır.

Veri madenciliği modelleri işlevlerine göre 3 temel grupta toplanır:

1. Sınıflama (Classification) ve Regresyon,
2. Kümeleme (Clustering),
3. Birliktelik kuralları ve sıralı örüntüler (Association rules and sequential patterns).



Şekil 2.3 Veri Madenciliği modelleri

2.5.1. Sınıflama ve regresyon

Dağınık bir yapıda bulunan verilere sınıf niteliğinin uygulanması sürecidir . Sınıflama algoritması, ortak özelliklere sahip kayıtların farklı sınıflar içine

aktarılmasını belirleyen algoritmadır. Sınıf olmak iç in her kaydın sınıf içinde yer alan diğer kayıtlarla belirlenmiş bir ortak özelliği olması gerekir[11]. Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen iki veri analiz yöntemidir. Sınıflama kategorik değerleri tahmin ederken, regresyon süreklilik gösteren değerlerin tahmin edilmesinde kullanılır. Sınıflama, verinin önceden belirlenen çıktılara uygun olarak ayrıştırılmasını sağlayan bir tekniktir. Çıktılar, önceden bilindiği için sınıflama, veri kümesini denetimli olarak öğrenir.

Sınıflama sorgusu kullanılarak bir kaydın daha önceden nitelikleri belirlenmiş bir sınıfa girmesi amaçlanmaktadır Sınıflama algoritması öğrenme verilerini kullanarak hangi sınıfların var olduğu ve bu sınıflara girebilmek için kayıtların hangi özelliklere sahip olması gerektiğini otomatik olarak keşfeder. Sınıflama algoritmaları iki şekilde kullanılır:

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

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

Sınıflama algoritmasının kullanım alanları, banka kredisi onaylama işlemi, kredi kartı sahteciliği tespiti ve sigorta risk analizidir [11].

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

Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler şunlardır [6]:

1 - Karar Ağaçları (Decision Trees)

2- Yapay Sinir Ağları (Artificial Neural Networks)

3- Genetik Algoritmalar (Genetic Algorithms)

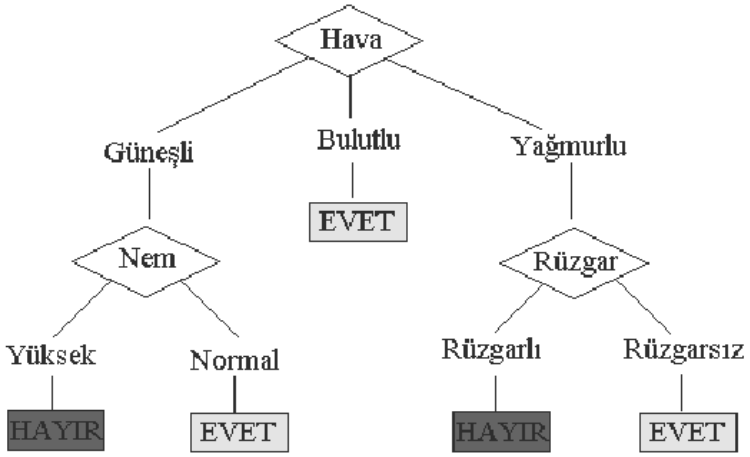
4- K-En Yakın Komşu (K-Nearest Neighbor)

5- Bellek Temelli Nedenleme (Memory Based Reasoning)

6- Naive-Bayes

Karar ağacı, adından da anlaşılacağı gibi bir ağaç görünümünde, tahmin edici bir tekniktir . ağaç yapısı ile, kolay anlaşılabilen kurallar yaratabilen, bilgi teknolojileri işlemleri ile kolay entegre olabilen en popüler sınıflama tekniğidir.

Karar ağacı yapılarında, her düğüm bir nitelik üzerinde gerçekleştirilen testi, her dal bu testin çıktısını, her yaprak düğüm ise sınıfları temsil eder. En üstteki düğüm kök düğüm olarak adlandırılır. Karar ağaçları, kök düğümden yaprak düğüme doğru çalışır [12]. Şekil 2.4 de hava durumuna göre tenis oynayıp oynamama kararını veren karar ağacı gösterilmiştir.



Şekil 2.4 Örnek bir karar ağacı

Karar ağacından şu kurallar türetilebilir:

1. Eğer Hava = Güneşli ve Nem = Yüksek ise o zaman Tenis= Oynama.
2. Eğer Hava = Güneşli ve Nem = Normal ise o zaman Tenis= Oyna.
3. Eğer Hava = Bulutlu ise o zaman Tenis= Oyna.
4. Eğer Hava = Yağmurlu ve Rüzgar = Rüzgarlı ise o zaman Tenis= Oynama.
5. Eğer Hava = Yağmurlu ve Rüzgar = Rüzgarsız ise o zaman Tenis= Oyna.

Karar ağacı tekniğini kullanarak verinin sınıflanması iki basamaklı bir işlemdir. İlk basamak öğrenme basamağıdır. öğrenme basamağında önceden bilinen bir eğitim verisi, model oluşturmak amacıyla sınıflama algoritması tarafından analiz edilir. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. İkinci basamak ise sınıflama basamağıdır. Sınıflama basamağında test verisi, sınıflama kurallarının veya karar ağacının doğruluğunu belirlemek amacıyla kullanılır. Eğer doğruluk kabul edilebilir oranda ise, kurallar yeni verilerin sınıflanması amacıyla kullanılır.

Test verisine uygulanan bir modelin doğruluğu, yaptığı doğru sınıflamanın test verisindeki tüm sınıflara oranıdır. Her test örneğinde bilinen sınıf, model tarafından tahmin edilen sınıf ile karsılaştırılır. Eğer modelin doğruluğu kabul edilebilir bir değer ise model, sınıfı bilinmeyen yeni verileri sınıflama amacıyla kullanılabilir.

Karar ağaçları, 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 isletmeye en faydalı olan bireylerin özelliklerini kullanarak ise 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 gibi uygulamalarda kullanılmaktadır [6].

Sinir ağları, tanımlayıcı ve tahminci veri madenciliği algoritmalarındandır. İnsan beyninin fizyolojisini taklit ederler. Komplike ve belirsiz veriden bilgi üretirler. Keşfettikleri örüntü ve trendler, insanlar yada bilgisayarlarca kolay keşfedilemez. Bu tür karmaşık problemlerde birbirleriyle etkileşimli yüzlerce değişken bulunur [13]. Bu teknik, veritabanındaki örüntüleri, sınıflandırma ve tahminde kullanılmak üzere genelleştirir. Sinir ağları algoritmaları sayısal veriler üzerinde çalışırlar.

Genetik algoritma, Darvin tarafından geliştirilen evrim teorisine dayalıdır. Algoritma ilk olarak populasyon adı verilen bir çözüm kümesi (öğrenme veri kümesi) ile başlatılır. Bir populasyondan alınan sonuçlar bir öncekinden daha iyi olacağı beklenen yeni bir populasyon oluşturmak için kullanılır. Evrim süreci (yeni populasyonlar yaratma iterasyonu) tamamlandığında bağımlılık kuralları veya sınıf modelleri ortaya konmuş olur [14].

Veri uzayında birbirine yakın olan aynı tip kayıtlar, birbirlerinin komşusu durumundadırlar. Bu anlayış doğrultusunda, çok kolay fakat güçlü olan k – en yakın komşu algoritması geliştirilmiştir. k - en yakın komşu algoritmasının temel felsefesi komşunun yaptığını yaptır. Belirli bir bireyin (kayıtın) davranışı (özelliğini) tahmin etmek istenirse, veri uzayında o bireye yakın olan örneğin 10 bireyin davranışına bakılabilir. Bu 10 komşunun davranışının ortalaması hesaplanır ve bu hesaplanan ortalama bireylerin tahmini olur. k - en yakın komşudaki k harfi araştırdığımız komşu sayısıdır. Örneğin, 5 - en yakın komşuda 5 komşuya bakılır[15].

İnsanlar kararlarını genellikle daha önce yasadıkları deneyimlere göre verirler. Örneğin doktorlar bir hastayı incelerken, elde ettiği bulguları daha önce tedavi ettiği benzer hastalığa yakalanmış hastalar üzerindeki deneyimlerini kullanarak değerlendirirler. Bellek tabanlı yöntemler de benzer şekilde deneyimleri kullanmaktadır. Bu yöntemlerde, bilinen kayıtların bulunduğu bir veritabanı

oluşturulur ve sistem yeni gelen bir kayda komşu olan diğer kayıtları belirler ve bu kayıtları kullanarak tahminde bulunur ya da bir sınıflama işlemi uygular. Bellek tabanlı yöntemlerin en önemli özelliği veriyi olduğu gibi kullanabilme yeteneğidir. Diğer VM yöntemlerinin aksine bellek tabanlı yöntemler, kayıtların sekli (format) yerine sadece iki işlemin varlığı ile ilgilenir. Bu işlemler, iki kayıt arasındaki uzaklığı belirleyen bir uzaklık fonksiyonu ve komşu kayıtları isleyerek bir sonuç üreten kombinasyon fonksiyonudur [16].

Bellek tabanlı yöntemler sahtekarlık tespiti ve klinik işlemler gibi alanlarda kullanılmaktadır.

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

2.5.2. Kümeleme

Kümeleme işlemi, heterojen yapıya sahip bir kitleyi daha homojen birkaç alt gruba ya da kümeye bölme işlemidir. Kümeleme analizi, nesnelerin altdizinlere gruplanmasını yapan bir işlemdir. Böylece nesneler, örneklenen kitle özelliklerini iyi yansıtan etkili bir temsil gücüne sahip olmuş olur. Sınıflamanın aksine, yeniden tanımlanmış sınıflara dayalı değildir. Kümeleme, bir denetimsiz öğrenme (unsupervised learning) yöntemidir.

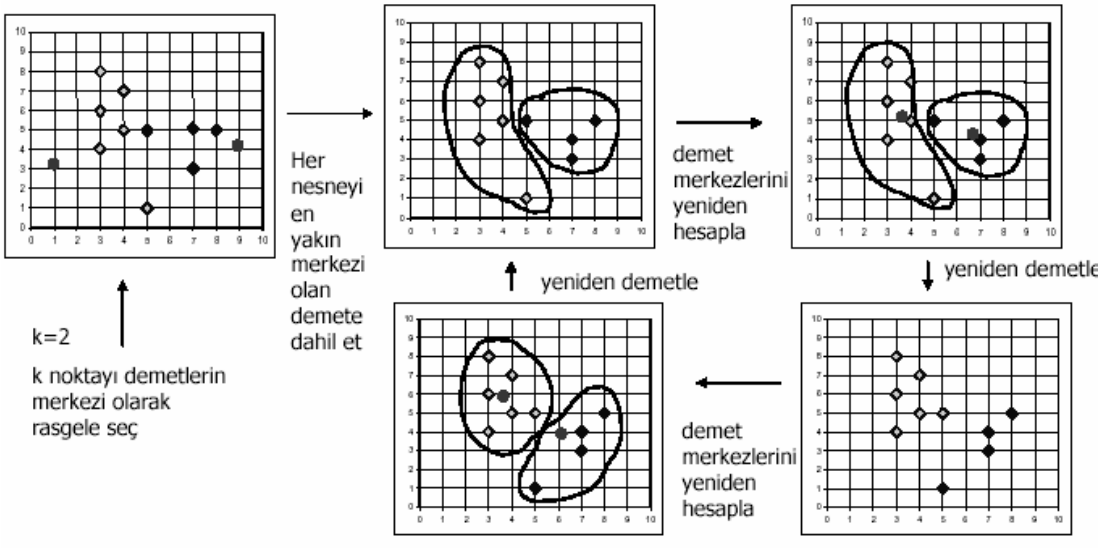
Sınıflama ile kümelemeyi birbirinden ayıran en önemli fark, kümeleme işleminin sınıflama işleminde olduğu gibi önceden belirlenmiş bir takım sınıflara göre bölme yapmamasıdır. Sınıflamada her bir veri, önceden sınıflandırılmış bir takım sınıflar üzerinde yapılan bir eğitim neticesinde ortaya çıkan bir modele göre önceden belirlenmiş olan bir sınıfa atanmaktadır. Kümeleme işleminde ise önceden tanımlanmış sınıflar ya da örnek sınıflar bulunmamaktadır. Verilerin kümelenmesi işlemi, verilerin birbirlerine olan benzerliklerine göre yapılmaktadır. Oluşan sınıfların hangi anlamları taşıdığının belirlenmesi tamamen çözümlemeyi yapan kişiye kalmıştır.

Kümeleme modelinde, sınıflama modelinde olan veri sınıfları yoktur [17]. Verilerin herhangi bir sınıfı bulunmamaktadır. Sınıflama modelinde, verilerin sınıfları belirlenmiştir ve yeni bir veri geldiğinde bu verinin hangi sınıftan olabileceği tahmin edilmektedir. Oysa kümeleme modelinde ise, sınıfları bulunmayan veriler belirlenen benzerlik-yakınlık kriterlerine göre gruplar halinde kümelere ayrılırlar. Küme içindeki elemanların benzerliği olmalı, kümeler arasında ise benzerliğin az olması gerekir.

En yaygın kullanılan kümeleme algoritması k ortalamalar algoritmasıdır. Diğer kümeleme teknikleri ile karşılaştırıldığında k ortalamalar algoritması (k means) büyük veritabanlarının kümelenmesinde oldukça etkin bir algoritmadır. Yeni bir vaka ortaya çıktığında; algoritma tüm veriyi inceleyerek buna en çok benzeyen vakaların bir altkümesini oluşturur ve onları çıktıyı tahmin etmek için kullanır [18].

k-means yöntemi, ilk önce n adet nesneden rasgele k adet nesne seçer ve bu nesnelerin her biri, bir kümenin merkezini veya orta noktasını temsil eder. Geriye kalan nesnelerden her biri kendisine en yakın olan küme merkezine göre kümelere dağılırlar. Yani bir nesne hangi kümenin merkezine daha yakın ise o kümeye yerleşir. Ardından her küme için ortalama hesaplanır ve hesaplanan bu değer o kümenin yeni merkezi olur. Bu işlem tüm nesneler kümelere yerleşinceye kadar

devam eder . Şekil 2.5 de örnek bir veri setinin K Means algoritması ile kümelenmesi görülmektedir.



Şekil 2.5 Veri setinin K Means Algoritması ile kümelenmesi

2.5.3. Birliktelik kuralları ve sıralı örüntüler

Birliktelik Kuralları büyük veri kümeleri arasındaki ilginç ilişkileri veya korelasyonları bulmak için kullanılır. Birliktelik Kuralları, verilen veri kümesi içindeki sıkça görülen özellik değer durumlarını tespit eder. Birliktelik kurallarının tipik ve genel kullanılan alanı market sepet analizidir[19]

Birliktelik kurallarına ait örnekler aşağıda yer almaktadır:

1. “Müşteriler bira satın aldıklarında %75 olasılıkla çocuk bezi de satın alırlar.”
2. “Düşük yağlı peynir ve yağsız süt alan müşteriler %85 olasılıkla diyet süt alırlar.”

Ardışık analiz ise birbiriyle ilişkisi olan ancak birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Aşağıda ardışık analize ait örnekler yer almaktadır.

1. "Çadır alan müşterilerin %10'u bir ay içerisinde sırt çantası almaktadır."
2. "A hissesi %15 artarsa üç gün içinde B hissesi %60 olasılıkla artacaktır."
3. "X ameliyatı yapıldığında, 15 gün içinde % 45 ihtimalle Y enfeksiyonu oluşacaktır."
4. "Çekiç satın alan bir müşteri, ilk üç ay içerisinde % 15, bu dönemi izleyen üç ay içerisinde % 10 ihtimalle çivi satın alacaktır."

Tanımlayıcı tekniklerden olan Birliktelik-İlişki Kuralları takip eden bölümde geniş olarak verilmektedir.

BÖLÜM 3. BİRLİKTELİK KURALI

Birliktelik kuralı, geçmiş verilerin analiz edilerek bu veriler içindeki birliktelik davranışlarının tespiti ile geleceğe yönelik çalışmalar yapılmasını destekleyen bir yaklaşımdır. 90 yılların başına kadar saklanan satış verilerinde ürün ve müşteri verisi çok nadir yer alırken, genelde mali açıdan önemli olan tutarsal gelir verilerinin depolaması yapılıyordu. 90 yılların başından itibaren veri toplama uygulamalarındaki gelişmeler doğrultusunda firmaların satış noktalarında yeni teknoloji otomatik ürün veya müşteri tanıma sistemleri (bar kod ve manyetik kart okuyucular) yaygınlaşmaya başlamıştır. Bu tip teknolojik gelişmeler, bir satış hareketine ait verilerin satış esnasında toplanmasına ve elektronik ortamlara aktarılmasına olanak tanımıştır. Günümüzde süper marketlerde, orta ve büyük ölçekli alışveriş mağazalarındaki satış noktalarında akıllı satış sistemlerinin kullanımı oldukça yaygındır. Bu satışlardan elde edilen verilerde, işlem tarihi, satın alınan ürünlere ait bilgiler (ürün kodu, miktar, fiyat, ıskonto vb.) yer alır ve ayrıca hareket numarası tekildir. Bazı kuruluşlar bu tip bilgileri içeren veritabanlarını pazarlama alt yapılarının önemli parçalarından biri olarak görmekte ve bu verileri kullanmak iç in çaba harcamaktadırlar [20].

Birliktelik kuralında, müşterilerin alışveriş esnasında satın aldıkları ürünler arasındaki birliktelik-ilişki bağlarını bularak, müşterilerin satın alma alışkanlıklarının tespit edilmesi amaçlanmaktadır. Keşfedilen bu birliktelik-ilişki bağıntıları sayesi ile satıcılar daha etkin ve kazançlı satışlar yapabilme imkanına sahip olmaktadırlar. Süper market alışverişi esnasında müşteriler patates cipsi aldıktan sonra genelde aynı alışverişte kola da satın alıyorlarsa, bu iki ürün arasında kuvvetli bir birliktelik-ilişki kuralı var anlamı yakalanır. Bu elde edilen veri sayesi ile bu ürünlere ek ürün satışı yapmak için düzenlemeler yapılabilir.

Birliktelik kurallarının kullanıldığı en tipik örnek market sepeti uygulamasıdır. Bu işlem, müşterilerin yaptıkları alışverişlerdeki ürünler arasındaki birliktelikleri bularak müşterilerin satın alma alışkanlıklarını analiz eder. Bu tip birlikteliklerin keşfedilmesi, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve market yöneticileri de bu bilgi ışığında daha etkili satış stratejileri geliştirebilirler.

Örneğin, bir süper markette ekmek ve peynir satın alınan satış hareketlerinin %75'inde zeytin de satın alınmıştır. Bu tür birliktelik-ilişki örüntüleri ancak, örüntüde yer alan öğelerin birden fazla harekette tekrarlandığında potansiyel olarak mevcut olabilirler.

Markette bulunabilecek tüm ürünlerin kümesini evren olarak düşünecek olursak, her ürünün varlığını veya yokluğunu gösteren boolean bir değişkeni olacaktır. Böylece her bir sepeti bu boolean değerlerden oluşan bir vektör olarak tasvir edebiliriz. Bu vektörlerden alınan numuneler hangi ürünlerin beraber satıldığını ortaya koyabilir. Bu numuneler ilişkisel kurallar formunda tasvir edilebilir[21].

3.1. Birliktelik Kuralının Matematiksel Gösterimi

Birliktelik kuralının matematiksel modeli 1993 yılında Agrawal, Imielinski ve Swami tarafından ifade edilmiştir. Bu modele göre; I=$(i_1, i_2, i_3 ..... i_m)$ nesnelerin kümesi ve D işlemler kümesi olarak ifade edilir. Her i, bir nesne (ürün) olarak adlandırılır. D veritabanında her işlem T, T $\subseteq$ I olacak şekilde tanımlanan nesnelerin kümesi olsun. Her işlem bir tanımlayıcı alan olan TID ile temsil edilir. A ve B nesnelerin kümeleri olsun. Bir T işlemler kümesi ancak ve ancak A $\subseteq$ T ise yani A, T'nin alt kümesi ise A'yı kapsıyor denir. Bir birliktelik kuralı A $\Rightarrow$ B formunda ifade edilir. A önce ve B sonuç olarak adlandırılır. Burada, A $\subseteq$ I, B $\subseteq$ I ve A $\cap$ B = $\varnothing$ dır.

Hareket numaraları gruplandırılarak elde edilen ürünler arasındaki bağımlılık ilişkisinin yüzde yüz doğru olması beklenemez. Benzer şekilde, çıkarsama yapılan kuralın eldeki hareketler kümesinin önemli bir kısmı tarafından desteklenmesi istenir. Bu nedenlerden dolayı, $X \Rightarrow Y$ eşleştirme kuralı kullanıcı tarafından minimum değeri belirlenmiş güvenirlik (c:confidence) ve destek (s:support) eşik değerlerini sağlayacak biçimde üretilir. $X \Rightarrow Y$ eşleştirme kuralına, c güvenirlik ölçütü ve s destek ölçütü iliştirilir ve biçimsel olarak $\theta(D)=(X \Rightarrow Y,c,s)$ ile gösterilir. Burada D örneklemi; $X \Rightarrow Y$ birliktelik-ilişki kuralını; c eşik değeri, ilgili kuralın minimum güvenirliğini (X ürünlerini içeren hareketlerin en az %c oranında Y içeren hareketler kümesinde yer aldığını); s ilgili kuralın, minimum desteğini (X ve Y ürünlerini içeren hareket tutanaklarının toplam hareket tutanakları içinde en az %s oranında var olduğunu) gösterir[22].

Ürünler kümesi ailesini $\Im(I)$ ile gösterelim ve X ve Y'nin her ikisi de $\Im$ (I) üzerinde değişebilen iki rasgele değişken olsun. Pr(X), X kümesi içinde yer alan tüm ürünlerin herhangi bir sepet varlığında bulunma olasılığını; $Pr(X \cap Y)$, X ve Y rasgele değişkenlerince paylaşılan ortak ürünlerin herhangi bir sepet varlığında bulunma olasılığını; ve $Pr(X \cup Y)$, X ve Y rasgele değişkenlerinin birleşiminde yer alan ürünlerin herhangi bir sepet varlığında bulunma olasılığını göstersin. O zaman, güvenirlik eşiği $Pr(Y/X)=Pr(X \cap Y)/Pr(X)$ ile, destek eşiği ise $Pr(X \cup Y)$ ile ifade edilir. Güvenirlik metriği, eşleştirme kuralının doğruluk derecesini, destek metriği ise kuralda yer alan öğelerin (ürünlerin) geçiş sıklığını gösterir. Yüksek güvenirlik ve destek değerine sahip kurallara güçlü kurallar adı verilir[22].

Birliktelik-ilişki kuralı formülsel olarak şu şekilde tanımlanabilir;

$$A1,A2,.....An \Rightarrow B1,B2,.....Bm \qquad (3.1)$$

Buradaki, Ai ve Bj yapılan iş veya nesnelerdir. Bu kural genellikle A1, A2,...,An meydana geldiğinde, sık olarak B1, B2, ..., Bm aynı olay veya hareket içinde yer aldığı anlamına gelmektedir [23].

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

Ürün(X,"dijital fotoğraf makinesi") $\Rightarrow$ Ürün(X,"bellek kartı")

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

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

Yaş (X, "12 – 17"), Meslek(X, "öğrenci") $\Rightarrow$ Ürün(X,"oyun konsolu - playstation")

Bu kural ile, "12-17 yaşları arasındaki öğrenci en çok "oyun konsolu (playstation) almaktadır" anlamı elde edilmektedir [23].

Yaş(X, "30...39") $\wedge$ gelir(X, "60K...69K") $\Rightarrow$ alış(X, "Plazma TV")

Meslek(X, "öğrenci") $\wedge$ yaş (X, "15...20") $\Rightarrow$ alış(X, "Oyun Konsolu")

Yukarıdaki ilk kuralda, otuzlu yaşlarındaki, yıllık gelirleri 60K-69K arasında olan müşterilerin Plazma TV satın almış olduğunu gösterir . Bir sonraki kural ise, yirmi yaş altı öğrenci olan müşterilerin oyun konsolu satın almış olduğunu ifade etmektedir.

3.1.1. Güven (confidence) ve destek (support) kavramları

Kuralın destek ve güven değerleri, kuralın ilginçliğini ve ilgililiğini ifade eden iki ölçüdür. Bu değerler sırasıyla keşfedilen kuralların yararlılığını (kullanışlılığını) ve kesinliğini (doğruluğunu) ifade eder.

Güven ve destek değerlerinin örnek bir formülü şu şekildedir:

$A \Rightarrow B$ [destek = % 2, güven = % 60] (3.2)

$(A \Rightarrow B)$ güveni aşağıdaki gibi hesaplanır:

güven $(A \Rightarrow B)$ = (A ve B'nin bulunduğu satır sayısı) / (A'nın bulunduğu satır sayısı) (3.3)

Güven değerinin %60 olduğu (3.2) den çıkan sonuç ; A ürünü satın alanların %60'ı B ürününü de almışlardır. Güven değerinin %100 olması demek A ürünün alan her kişi B ürünün de almıştır anlamına gelir ve böyle kurallara kesin kural adı verilir.

$(A \Rightarrow B)$ desteği ise şu şekildedir:

destek $(A \Rightarrow B)$= (A ve B'nin bulunduğu satır sayısı) / (toplam satır sayısı) (3.4)

Destek değeri %2 olan (3.2) den çıkan sonuç; Satılan tüm satışların %2'sinde A ve B birlikte bulunmaktadır.

Tablo 3.1 Ürün satış tablosu

TID	ÜRÜNLER
1	Su , Ekmek, Kek, Süt
2	Su, Kek, Ekmek, Balık
3	Bira, Ekmek, Kek, Süt
4	Ekmek, Kek, Süt
5	Su, Bira, Kek, Süt

Tablo 3.1 den yola çıkarak toplam alış hareketlerine göre {Kek, Süt} ile Su arasındaki ilişki şu şekilde açıklanabilir:

$$\text{Destek-support} = \frac{(\text{Kek, Süt, Su})}{\text{Toplam hareket}} = \frac{2}{5} = 0.4$$

$$\text{Güven-confidence} = \frac{(\text{Kek, Süt, Su})}{(\text{Kek ,Süt})} = \frac{2}{4} = 0.5$$

Bu eşitliklerden de anlaşılacağı gibi, {Kek, Süt} $\Rightarrow$ Su kuralı %40 destek, %50 güven ölçülerine sahiptir

Birliktelik kuralının kullanım alanları, market satış analizlerinde, ticarette, mühendislikte, tıp ve finans şeklinde sıralanabilir. Sepet analizi (Market basket analysis) en çok kullanıldığı alanlardan biridir. Müşteri alım alışkanlıklarına ve perakendecilik esaslarına göre kararlar alınmasını sağlar; hangi ürün indirime konacağı, katalogların nasıl tasarlanacağı, raflarda ürünleri nasıl dizileceği vb [22].

Örnek olarak sepet analizi yöntemi farklı raf dizimlerinin olabilmesine olanak tanır. Bir stratejide, birlikte sık olarak alınan ürünler raflarda yakın yerlere dizilebilirler. Bilgisayar alan müşterilerin çoğunluğu yazılım da alma eğilimdeler ise

bu ürünlerin yakın yerlere konulması iki ürünün satış oranlarını da artırabilir. Diğer alternatif bir stratejide, bilgisayar ve yazılım ürünlerini markete ait bir rafın başlangıcına ve sonuna koymak, müşteriyi kandırma metotlarından birisi olabilir. Çünkü müşteri raf boyunca başka ürünlere bakarak ilerler ve bunları satın alma olasılığı doğar [19].

Birliktelik-ilişki kuralı madenciliği 2 aşamalıdır:

1. Tüm sık geçen nesne kümelerinin bulunması: Tanıma göre her nesne kümesinin sık geçenler kümesinde yer alabilmesi için, her nesnenin destek değerinin önceden tanımlanmış olan min_destek değerinden büyük olması gerekir.

2. Sık geçen nesne kümelerinden güçlü ilişki kurallarının yaratılması: Tanıma göre, bu kurallar min_destek ve min_güven durumunu sağlamalıdır.

Birliktelik kuralı algoritmalarının performansını belirleyen adım birinci adımdır. Sık geç en öğe kümeleri belirlendikten sonra, eşleştirme kurallarının bulunması düz bir adımdır.

Birliktelik kuralı çıkarmak için en çok kullanılan algoritma Apriori algoritmasıdır.

3.2. Apriori Algoritması

Apriori, boolean ilişki kuralları için geçerli bir veri madenciliği algoritmasıdır. Algoritmanın ismi, bilgileri bir önceki adımdan aldığı için "prior" anlamında Apriori'dir. Bu algoritma özünde iteratif (tekrarlayan) bir niteliğe sahiptir [19] ve

hareket bilgileri içeren veritabanlarında sık geçen öğe kümelerinin keşfedilmesinde kullanılır.

Sık geçen öğe kümelerini bulmak için birçok kez veritabanını taramak gerekir. İlk taramada bir elemanlı minimum destek ölçütünü sağlayan sık geçen öğe kümeleri bulunur. İzleyen taramalarda bir önceki taramada bulunan sık geçen öğe kümeleri aday kümeler adı verilen yeni potansiyel sık geçen öğe kümelerini üretmek için kullanılır. Aday kümelerin destek değerleri tarama sırasında hesaplanır ve aday kümelerinden minimum destek ölçütü sağlayan kümeler o geçişte üretilen sık geçen öğe kümeleri olur. Sık geçen öğe kümeleri bir sonraki geçiş için aday küme olurlar. Bu süreç yeni bir sık geçen öğe kümesi bulunamayana kadar devam eder [20].

Bu algoritmada temel yaklaşım eğer k-öğe kümesi minimum destek ölçütünü sağlıyorsa, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar. Bir öğeler kümesinin destek değeri altkümesinin destek değerinden büyük olamaz. Yani Y kümesi X kümesinin alt kümesi ise:

$$(X \subseteq Y) \Rightarrow s(X) \geq s(Y) \qquad (3.5)$$

şeklinde olmalıdır.

Bir sık geçen nesne kümesinin bütün boş olmayan altkümeleri de sık geçmektedir. Bu özellik su gözleme dayanmaktadır. Eğer bir nesne küme I, minimum destek eşik değeri olan min_des değerini sağlayamıyor ise, o zaman I sık geçen değildir denir. Bu durum P(I) < min_des seklinde ifade edilir. Eğer bir A nesnesi I nesne kümesine eklenir ise, kümenin son hali $I \cup A$, I kümesinden daha fazla sık geçmez, yani $I \cup A$ da sık geçen değildir[19].

Kullanılan pazar sepeti verisinde her harekette yer alan ürün kodları sayısaldır ve ürün kodları küçükten büyüğe doğru sıralıdır. Öğe kümeleri eleman sayıları ile birlikte anılır ve k adet ürüne sahip bir öğe kümesi, k-öğe kümesi diye isimlendirilir. k-öğe kümesi c ifadesi ile gösterilirse, öğeleri (ürünler) c[1], c[2], c[3],...,c[k] şeklinde gösterilir ve c[1] < c[2] < c[3]< ... <c[k] olacak şekilde küçükten büyüğe doğru sıralıdır [23]. Her öğe kümesine destek metriğini tutmak üzere bir sayaç değişkeni iliştirilmiştir ve sayaç değişkeni öğe kümesi ilk kez yaratıldığında sıfırlanır. Aday öğe kümeleri C karakteri ile gösterilir.

Tablo 3.2 Apriori Algoritmasında kullanılan değişkenler

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

Apriori algoritmasının klasik özet kodu Şekil 3.1 de görülmektedir. Bu şekilde yer alan apriori-gen işlevi, (k-1) adet öğeye sahip $L_{(k-1)}$ öğeler kümesini kullanarak k adet öğeye sahip aday kümeleri üretir. Bu işlev şu biçimde çalışır. İlk önce, $L_{(k-1)}$ ile $L_{(k-1)}$ birleştirme (join) işlemine tabi tutulur.

Birleştirme işleminde $L_{(k-1)}$ öğe kümesinin her satırında yer alan son öğe haricinde diğer öğelerin çapraz olarak benzerliği aranır ve son öğe haricinde diğer öğelerle yakalan benzerliklerden yeni aday öğe kümeleri oluşturulur. Oluşan kümeler budama (prune) adımı ile budanarak işlevden dönülür. Budama işlemi şu şekilde

yapılır; c aday kümesinin (k-1) öğeye sahip alt kümelerinden 'de yer almayan kümeler silinir. Apriori-gen işlevinin algoritma

kesiti, Şekil 3.2'de verilmiştir [23].

```
1)  L1 = { sık geçen 1-öğe kümesi };
2)  for ( k = 2; Lk-1 ≠ Ø; k++ ) do begin
3)      Ck = apriori-gen(Lk-1); // Yeni adaylar
4)      forall transactions-hareketler t ∈ D do begin
5)          Ct = subset(Ck, t); // Adaylar t içindedir
6)          forall candidates – adaylar c ∈ Ct do
7)              c.count++;
8)      end
9)      Lk = {c ∈ Ck | c.count ≥ minsup }
10) end
11) Answer = ∪k Lk;
```

Şekil 3.1 Apriori Algoritması özet kodu[23]

Budama aşamasında, tüm öğe kümeleri c $\in$ C_k şeklindeki öğe kümeler, bunların bazıları c kümesinin (k-1) öğeye sahip içinde $L_{(k-1)}$ barındırmayan tüm alt kümeleri silinir [23]. Farklı bir ifade ile budama, C_k aday öğe kümesindeki öğelerin alt kümelerinin $L_{(k-1)}$ kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, $L_{(k-1)}$ kümesinde yer almıyorsa ilgili öğe değerlendirme dışı kalır ve C_k aday öğe kümesinden silinir.

```
insert into C_k
select p.items_1, p.items_2, ...., p.items_k-1, q.item_k-1
from L_k-1 p, L_k-1 q
where p.item_1 = q.item_1, ...., p.item_k-2 = q.item_k-2 ,
      p.item_k-1 < q.item_k-1;

forall itemsets c ∈ C_k do
      forall (k-1)-subsets s of c do
            if (s ∉ L_k-1) then
                  delete c from C_k
```

.

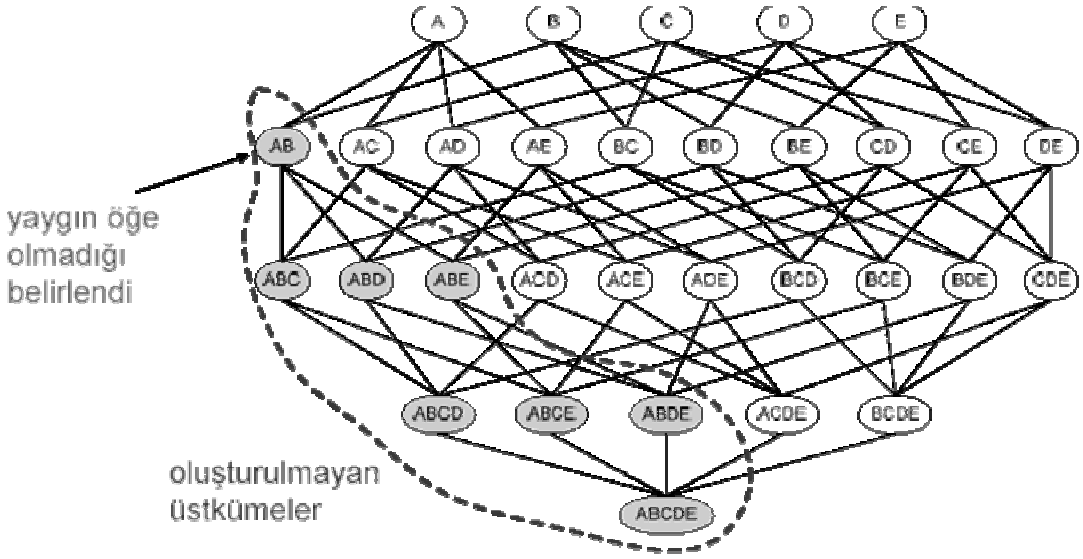
Şekil 3.2 Apriori-gen işleminin özet kodu

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

Kod içinde kurulan döngü yapısı ile ilk aşamada L_1 sık geçen öğe kümesinin öğelerinin ikili kombinasyonuna benzer bir şekilde ($L_1 \infty L_1$) yeni bir küme oluşur, bu işleme birleştirme (join) adı verilir, bu tarz oluşan kümelere de aday öğe kümeler adı verilir ve C harfi ile simgelenir. Oluşan bu aday öğe kümesinin her elemanı iki adet öğeden 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 öğe kümesinde olup olmadığına bakılır, alt kümelerden herhangi birisi L_1 içinde yer almayan küme elemanları C_2 aday kümesinden silinir. Apriori algoritması uygulanan veri topluluğu tekrar taranarak budama işleminden geçen

C_2 aday kümesi elemanlarının kaç adet hareket kaydı içinden geçtiği (destek sayacı) bulunur ve bulunan destek sayaç bilgileri doğrultusunda C_2 aday kümesi elemanlarının destek sayacı minimum destek değerine eşit veya büyük destek değerine sahip olan elemanları L_2 sık geçen öğe kümesini oluşturur. Diğer elemanlar ise silinir. Şekil 3.3 de budama işleminin grafiksel gösterimi verilmiştir.

Döngü bir sonraki aşamada L_2 kümesi öğelerinin üçlü kombinasyonu ile oluşturulan yeni bir aday öğe 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 öğe kümesi oluşturulur. Bu döngü her dönüşünde öğe sayısını artırarak devam eder. Bu süreç yeni bir sık geç en öğe kümesi bulunamayana kadar devam eder.



Şekil 3.3 Apriori budama işleminin grafiksel gösterimi.

Örnek bir veri seti Tablo 3.3 de verilmiştir. Bu örnek tablo üzerinde Apriori algoritmasını çalıştırarak en çok sık geçen ürünleri bulmaya çalışalım.

Tablo 3.3 Hareketler ve ürünler tablosu.

Müşteri Numarası	**Aldığı Ürünler**
101	Elma , Şeker , Çay , Domates
102	Ekmek , Domates , Un , Şeker
103	Elma , Domates , Ekmek
104	Şeker , Çay , Domates ,Peynir,
105	Elma , Domates , Un ,Çay
106	Makarna , Domates , Çay
107	Elma , Zeytin ,Domates
108	Un , Üzüm , Çay
109	Üzüm , Şeker ,Çay
110	Çay , Makarna , Elma , Un , Domates

Birliktelik kuralları , item-setler arasındaki eğilimi ya da ilişkiyi bulur. Item set , itemlerın kümesini oluşturur. Her bir hareket , item set olarak adlandırılır.Örneğin , Tablo 3.3'deki örnekte 108 numaralı müşterinin yapmış olduğu alışverişteki "Un-Üzüm-Çay" bir item settir.

Adım 1: Minimum destek sayısı (min.support) ve minimum güven değerinin (min.confidence) belirlenmesi

Minimum Destek: 2

Minimum Güven : %70 olarak seçilmiştir.

Adım 2 : İtem setler içerisindeki her bir itemin destek değerinin bulunması(Her ürünün hareket listesindeki geçiş sayısı C_1 tablosu).

Tablo 3.4 Tekli birlikteliklerin destek değerleri.

Ürün Seti	Destek Değeri
Elma	5
Şeker	4
Çay	7
Un	4
Ekmek	2
Domates	8
Peynir	1
Makarna	2
Üzüm	2
Zeytin	1

Adım 3 : Minimum destek değerinden daha düşük desteğe sahip olan itemlerın devre dışı bırakılması(Destek değeri 2 den küçük olan ürünlerin çıkarılması L_1 tablosu)

Tablo 3.5 Minimum destek değerini sağlayan ürünler

Ürün Seti	Destek Değeri
Elma	5
Şeker	4
Çay	7
Un	4
Domates	8

Adım 4 : Elde edilen tekli birliktelikler dikkate alınarak ikili birlikteliklerin oluşturulması ($L_1 \infty L_1$ yani C_2 tablosu)

Tablo 3.6 İkili birliktelikler ve destek değerleri

Ürün Seti	***Destek Değeri***
Elma , Şeker	*1*
Elma , Çay	*3*
Elma , Un	*2*
Elma , Domates	*5*
Şeker , Elma	*1*
Şeker , Çay	*3*
Şeker , Un	*2*
Şeker , Domates	*3*
Çay , Elma	*2*
Çay , Şeker	*3*
Çay , Un	*3*
Çay , Domates	*5*
Un , Elma	*2*
Un , Şeker	*2*
Un , Çay	*3*
Un, domates	*3*
Domates , Elma	*5*
Domates , Çay	*5*
Domates , Un	*3*
Domates , Şeker	*3*

Bu aşamaların her birinde , oluşturulan birlikteliklerin support değerleri göz önüne alınarak min.support değeri belirlenir. Burada AB ile $B \Rightarrow A$ ikililerinin biri dikkate alınmamaktadır.

Adım 5 : Minimum destek 3 olarak seçilirse ve bu değerden düşük olan ürün setleri çıkartılırsa liste Tablo 3.7 deki gibi olur. (L_2 tablosu)

Tablo 3.7 İkili birlikteliklerden destek değerini sağlayan setler

Ürün seti	***Destek Değeri***
Elma , Çay	*3*
Elma , Domates	*5*
Şeker , Çay	*3*
Şeker , Domates	*3*
Çay , Un	*3*
Çay , Domates	*5*

Adım 6 : Üçlü birlikteliklerin oluşturulması (C_3 tablosu). Genelde ikili birliktelikler göz önüne alınsa da veritabanındaki itemlerin birbirleri ile olan ilişkileri dikkate alınarak üçlü ve dörtlü veya daha fazla birliktelikler oluşturulabilir. Burada ele alınan market-basket verisine göre üçlü birliktelikler oluşturulabilir.

Tablo 3.8 Üçlü birliktelikler ve destek değerleri

Ürün Seti	***Destek Değeri***
Elma , Çay , Şeker	*1*
Elma , Çay , Domates	*3*
Elma , Domates ,Şeker	*2*
Elma , Domates ,Çay	*3*
Elma , Domates ,ekmek	*1*
Elma , Domates ,Un	*2*
Elma , Domates ,Makarna	*1*
Şeker , Çay ,elma	*1*
Şeker , Çay ,Domates	*2*
Şeker , Çay ,Peynir	*1*
Şeker , Çay , üzüm	*1*
Şeker , Domates , Elma	*1*
Şeker , Domates ,Çay	*2*
Şeker , Domates ,un	*1*
Şeker , Domates ,ekmek	*1*
Şeker , Domates ,Peynir	*1*
Çay , Un , Domates	*2*
Çay , Un ,Elma	*2*
Çay , Un , Üzüm	*1*
Çay , Un ,Makarna	*1*
Çay , Domates , Şeker	*2*
Çay , Domates , Elma	*3*
Çay , Domates , Peynir	*1*
Çay , Domates , Un	*2*
Çay , Domates ,Makarna	*2*

Adım 7 : Üçlü birlikteliklerden minimum destek değeri olan 3 değerini geçenlerin dışındakilerin çıkarılması (L_3 Tablosu)

Tablo 3.9 Üçlü birlikteliklerden destek değerini aşan ürün setleri

Ürün Seti	***Destek Değeri***
Elma , Çay , Domates	*3*

Tabloda oluşan üçlü ürün setinin ikili alt kümelerinden herhangi birisi Adım 5 teki L_2 tablosunda yer almasaydı bu ürün seti de silinmiş olacaktı. Fakat (Elma,Çay) , (Elma,Domates) ve (Çay, Domates) ürün setlerinin her biri L_2 sık geçen öğeler tablosunda yer aldığı için (Elma, Çay, Domates) ürün seti kabul edilir.

Adım 8 : Üçlü birlikteliklerden birliktelik kurallarının çıkarılması

Tablo 3.10 Üçlü birlikteliklerden çıkan birliktelik kuralları

Birliktelik	Açıklama	Güven
Elma & Çay $\Rightarrow$ Domates	Elma ve Çayın bulunduğu sette domatesin olma olasılığı	3/3=%100
Elma & Domates $\Rightarrow$ Çay	Elma ve domatesin bulunduğu sette çayın olma olasılığı	3/5=%60
Çay & Domates $\Rightarrow$ Elma	Çay ve Domatesin bulunduğu sette elmanın olma olasılığı	3/5=%60
Çay $\Rightarrow$ Elma & Domates	Çayın bulunduğu sette elma ve domatesin olma olasılığı	3/7 = %42
Domates $\Rightarrow$ Elma & Çay	Domatesin bulunduğu sette çay ve elmanın olma olasılığı	3/8 = %38
Elma $\Rightarrow$ Çay &Domates	Elmanın bulunduğu sette çay ve domatesin olma olasılığı	3/5 = %60

Tablo 3.10 dan çıkan sonuçlara göre minimum güvenilirlik değeri olan %70 barajını geçen (Elma & Çay $\Rightarrow$ Domates) kesin kural olarak çıkmıştır. Burada dikkat edilecek husus her (Elma & Çay) grubunu alan kişilerin Domates de aldığı fakat her (Elma & Domates) grubunu alan kişilerin Çay almadığı sonucudur.

BÖLÜM 4. UYGULAMA

Çalışmada İstanbul Güngören'deki Gün-Bak toptan satış firmasının 5 ile 21 Nisan 2008 tarihleri arasındaki veritabanı kayıtları kullanılmıştır. Firma genel itibariyle Güngören ilçesindeki bakkal, market ve büfelere toptan tekel ürünleri dağıtmaktadır. Uygulama ile amaç en sık birlikte geçen tekel ürünlerini bularak, elde edilen birliktelik-ilişki verileri ile stok planlama aşamalarını tekrardan yapılandırabilmek, satış alanında reyon düzenlenmesini ürün birlikteliklerini dikkate alarak satış rakamlarını destekleyecek şekilde değiştirebilmektir.

4.1. Uygulamada Kullanılan Teknolojiler

Uygulamada web programlama dillerinden sunucu tabanlı PHP (Personal Home Pages) dili kullanılmıştır. PHP dili script bir dildir yani kodları düz yazı dosyaları halinde kaydedilir ve kullanılacağı ortamda bir yorumlayıcı tarafından yorumlanmaktadır. PHP de yazılan program çalıştırılabilir dosya (EXE) haline getirilmesine gerek yoktur. Fakat dosyaların bulunduğu web sunucu PHP anlar hale getirilmelidir.

PHP kodlarını çalıştırmak üzere Apache web sunucu teknolojisi kullanılmıştır. PHP kodlarının Apache üzerinde çalıştırılması için Apache sunucusunun konfigürasyon dosyalarında değişiklik yapmak gerekir. Bunun için Apache sunucunun bulunduğu dizinde yer alan conf klasöründeki httpd.conf dosyası açılarak dosyanın herhangi bir yerine şu kodlar eklenmelidir:

```
ScriptAlias /php/ "c:/php/"
```

AddType application/x-httpd-php .php

Action application/x-httpd-php "/php/php.exe"

Kodlar eklendikten sonra konfigürasyon dosyası kaydedilir. Konfigürasyon dosyasında yapılması gereken bir değişiklik daha bulunmaktadır. Uygulamada yoğun veritabanı işlemleri yapıldığı için sunucunun cevap verme süre değeri değiştirilmelidir. Apache sunucuda bu değer default olarak 300 saniyedir. Bu değeri veritabanının genişliğine göre artırmak gereklidir.

#

Timeout: The number of seconds before receives and sends time out.

#

Timeout 3000

Bu şekilde Apache ayarları tamamlanmış olur. Aynı şekilde PHP scriptlerinin bir ömür süresi bulunmaktadır. Bunun sebebi kaynak kullanımını en aza indirmektir. Bu değer PHP de 30 saniyedir. Bu ayar Windows klasöründe yer alan php.ini dosyasındadır. Uygulama için bu değer sonsuz değer olan sıfır değerine atanmıştır.

; Resource Limits ;

max_execution_time = 0 ; Maximum execution time of each script, in seconds

Bu ayarlardan sora Apache sunucu ve PHP uygulamayı çalıştıracak hale gelmektedir.

Uygulamada yer alan veritabanı işlemleri için MySQL sunucu programı kullanılmıştır. MySQL teknik itibariyle sunucuda daimi olarak çalışır. MySQL arzu eden programa bildireceği veritabanı dosyasından veri çekerek sunar.

4.2. Uygulamada Veri Madenciliği Süreçleri

Bu kısımda uygulamanın hazırlanması esnasında geçen veri madenciliği süreçleri anlatılmaktadır.

4.2.1. Veri seçimi, ön işleme ve indirgeme

Uygulamada esas alınacak veritabanın elde edildiği aşamadır. Öncelikle firmanın kullandığı veritabanları incelenmiştir. Birçok tablo içerisinden satış verilerinin olduğu tablolardan fatura bilgileri ve birim fiyatlar, ürün tablolarından ürün kod, ürün açıklamaları gibi ürünle ilgili bilgiler, müşteri kodları ve tarih bilgileri çekilmiştir. Oluşan veri öbeği yılbaşından itibaren yapılan satış verilerini içermektedir. Günde ortalama 200 fatura ve her faturada ortalama 13 kalem satış meydana geldiğinden veritabanı oldukça fazla kayıt içermektedir. Bu yüzden uygulama için 15 günlük bir örnekleme yapılmıştır.

Örnekleme sonucunda 37218 adet kayıt içeren ve fatura_id, tarih, firma_kodu, urun_kodu, urun_adi, miktar, birim_fiyat alanlarına sahip bir tablo oluşmuştur. Yapılan uygulamada birim fiyat , firma kodu gibi alanlar algoritma için gerekli olmadığından , bu alanlar veri setinden çıkarılmıştır. Alan isimleri de:

fatura_id → fat_no

urun_kodu → urunkod

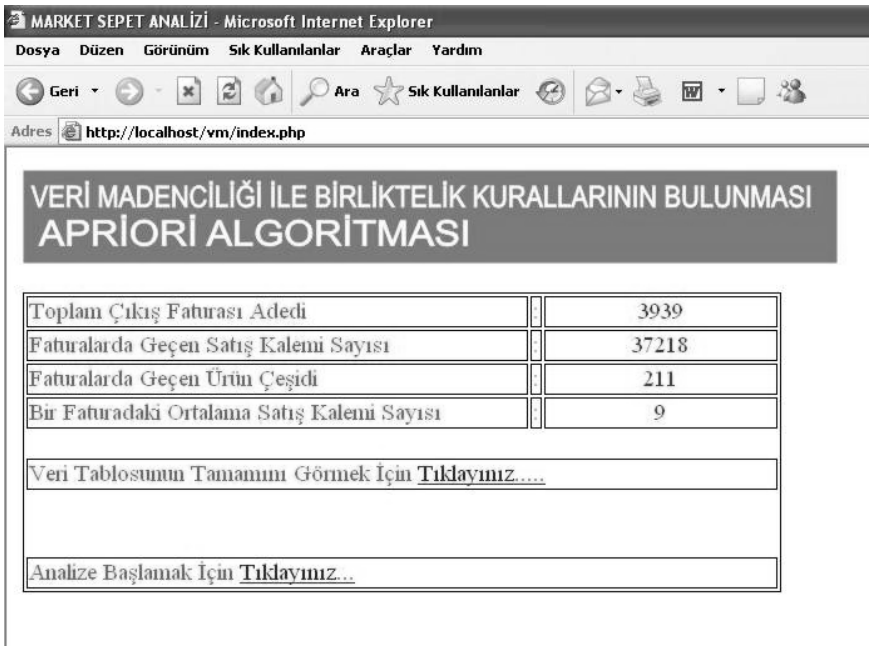
urun_adi → aciklama

miktar → adet

olarak değiştirilmiştir. MySQL sunucuda vm isimli bir veritabanı yaratılıp tablo data ismiyle kayıt edilmiştir. Tablo alanlarının veri tipleri korunmuştur.

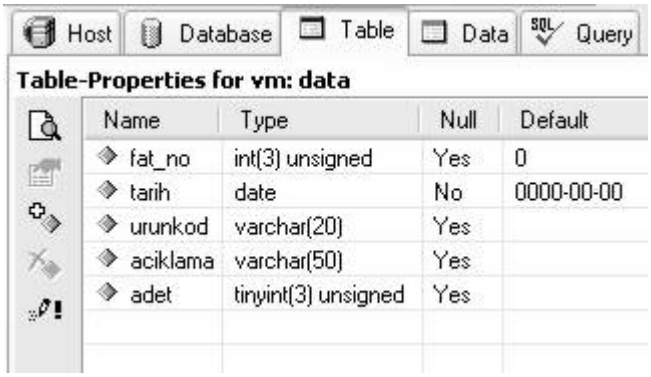
4.2.2 Uygulama ile veri madenciliği

Apriori algoritmasının uygulanması için toplamda 22 PHP dosyası oluşturulmuştur. Bunlar genel hatlarıyla, veritabanındaki tüm verileri gösteren giriş sayfaları, tarih aralığının seçildiği sayfa, algoritmanın uygulandığı adım sayfaları ve en son algoritma sonlandıktan sonra güven değerlerinin hesaplandığı güven sayfaları olarak sıralanabilir. Veritabanı işlemleri için ise başlangıçta ana verinin tutulduğu data tablosu bulunmaktadır. Bu data tablosu firmadan alınan bilgilerin ön işlemden geçmiş şeklidir. Daha sonra algoritmanın adımları takip edilirken birçok geçici tablo oluşturulmaktadır.



Şekil 4.1 Uygulama giriş ekranı

Şekil 4.1 de görüldüğü gibi firmadan alınan veritabanı ile ilgili genel bilgiler bulunmaktadır. Bu bilgilere firmaya ait farklı veritabanlarından ön işleme ile alınmış olan vm veritabanı içerisindeki data tablosundan ulaşılmaktadır. Şekil 4.2 de data tablosunun yapısı görülmektedir.



Şekil 4.2 Veritabanı içerisindeki ana data tablosunun yapısı

Data tablosundan verileri almak için Html içerisinde gömülü olan PHP kodları kullanılır. Öncelikle veritabanına bağlantı kurulmalıdır. Bunun için uygulamada sqlconn.php dosyası kullanılmaktadır. Sqlconn.php dosyasının içeriği şöyledir:

```
<?
$yol = mysql_connect("localhost","root","saw") or die("mySQL Sunucusuna Bağlanılamadı");
$vt = mysql_select_db("vm",$yol) or die("Veritabanı Bulunamadı ");
?>
```

Bu kod kümesindeki “mysql_connect("localhost","root","saw")” ile yerel sunucu olan Apache üzerinde root kullanıcı hesabıyla gerekli olan şifre girilerek bağlantı yolu kurulmaktadır. İkinci satırdaki “mysql_select_db("vm",$yol) or die("Veritabanı

Bulunamadı ")" kodu yol bağlantısı ile uygulamada kullanılan vm veritabanına bağlantıyı gerçekleştirmektedir. Sqlconn.php dosyası bir fonksiyon gibi düşünülebilir. Veri tabanına bağlanma ihtiyacı duyulan yerde :

```
<?
include("./sqlconn.php") ;
?>
```

kodu yazılarak sqlconn.php dosyası çağırılır ve veritabanına bağlantı gerçekleştirilmiş olur.

Şekil 4.1 deki verileri elde etmek için gerekli php kodu:

```
<?
include("./sqlconn.php") ;
@$sorgu =mysql_query("select count(distinct fat_no) from data ");
@$fatura_sayisi=mysql_result ($sorgu,0,0);  //Toplam fatura sayısı
@$sorgu =mysql_query("select count(adet) from data ");
@$kalem_sayisi=mysql_result ($sorgu,0,0);
@$sorgu =mysql_query("select count(distinct urunkod) from data ");
@$urun_cesidi=mysql_result ($sorgu,0,0);
?>
```

şeklindedir. Burada mysql_query() fonksiyonu ile SQL sorgusu yapılmaktadır ve $sorgu isimli değişkene aktarılmaktadır. $sorgu bir tablo olarak döner. $fatura_sayisi

değişkenine, mysql_result ($sorgu,0,0) fonksiyonundan gelen sonuç atanmaktadır. ($sorgu,0,0) ise, $sorgu değişkeni bir tabloyu temsil ettiği için, bu tablonun 0. satırının 0. elemanı anlamına gelmektedir. Elde edilen değişkenler html içerisinde gerekli yerlere gömülerek çıkış görüntüsü elde edilmektedir. Bunun için gerekli kod şu şekildedir:

```
<tr class="a">
<td width="400">Faturalarda Geçen Ürün Çeşidi  </td>
 <td width="5">:</td>
<td class="b"><? echo @$urun_cesidi;?> </td>
 </tr>
```

Buradaki kod ekrana basılan tablonun bir satırını temsil etmektedir. Kodda geçen <? echo @$urun_cesidi;?> kod parçacığı urun çeşidi değişkenini tablonun hücresi içerisinde ekrana yazdırmaktadır.

Giriş ekranından veritabanının tamamını görebilmek mümkündür. Şekil 4.3 de data tablosundan bir kesit görülmektedir.

Host | Database | Table | Data | Query

vm / data: 37218 Records (37218 retrieved)

fat_no	tarih	urunkod	adet
11632	2008-04-17	8691693006015	1
11632	2008-04-17	8691693000044	1
11632	2008-04-17	8698562404323	1
11632	2008-04-17	8691693001010	1
11632	2008-04-17	4032900005447	1
11636	2008-04-17	8691693002048	1
11636	2008-04-17	8691693006015	1
11636	2008-04-17	8691693000075	1
11636	2008-04-17	8690101111426	1
11641	2008-04-17	4031800001771	1
11641	2008-04-17	8690101112287	1

Şekil 4.3 Data tablosundan bir kesit

Şekil 4.3 den görüleceği üzere veriler faturalarda geçen kalemler şeklinde depolanmıştır. Her satır, bir fatura içerisindeki bir satış kalemini temsil etmektedir. Bu da veri tabanındaki veri satırlarının fazla olmasına neden olmaktadır. Şekil 4.1 den görüleceği üzere veritabanında 3939 adet fatura olmasına karşın data tablosu satış kalemlerinden oluşması itibarıyla 37218 kayıttan oluşmaktadır. Apriori algoritması veritabanının bir çok sefer taranmasını gerektirdiği için data tablosu indirgenmelidir. Veritabanı genel sorguları yine data tablosundan yapılırken, birliktelikler belirleneceği zaman, her satırında fatura ve o faturada geçen satış kalemlerinin oluşturacağı bir veri tablosuna ihtiyaç duyulmuştur. Burada 2 sorun oluşmaktadır. Birincisi her faturada geçen satış kalemi sayısı değişkenlik gösterdiğinden oluşturulacak tablonun kolon sayısı bilinmemektedir. Bunun için data tablosundan gerekli veritabanı sorguları yapılarak en çok satış kalemi olan fatura belirlenmiştir ve satış kalem sayısı olan 52 sayısı oluşturulacak tabloda kolon sayısı olarak kullanılmıştır. İkinci sorun da burada çıkmaktadır. Data tablosunda sadece bir satış kalemli faturalar da bulunmaktadır. Bu da yeni oluşturulan veri isimli tabloda bu fatura için 51 tane boş alan oluşacağı anlamına gelmektedir. Birliktelikler ve destek değerleri hesaplanırken kullanılan veri isimli tablodan bir kesit Şekil 4.4 de görülmektedir.

vm / veri: 3939 Records (3939 retrieved)

fat_no	tarih	urun1	urun2	urun3
11593	2008-04-17	8691693008033	4032900005447	4032900005805
11595	2008-04-17	8691693002048	8691693006015	4032900005447
11598	2008-04-17	8691693008019	8691693003014	8691693002017
11599	2008-04-17	8691693002017	8691693006015	8691693001010
11586	2008-04-16	8697530200011	8697530200226	8697530200028
11604	2008-04-17	8691693010876	4031800001771	8690101111426
11607	2008-04-17	8691693000075	8691693000044	8691693006015
11608	2008-04-17	8691693000013	8691693000044	8691693000075
11609	2008-04-17	8691693000044	8691693000075	8691693008033
11610	2008-04-17	4032900005447	4032900005805	4032900005041
11611	2008-04-17	8690101112287	4031800001832	8690101110078
11612	2008-04-17	8698562417729		

Şekil 4.4 Veri isimli tablodan bir kesit

Giriş sayfasından analize başlamak için tıklandığında step1.php çalıştırılmakta ve karşımıza analizin yapılacağı tarih aralığını seçmemizi isteyen ekran gelmektedir. Firmadan alınan bilgilerin geçtiği aralık 15 günü kapsamaktadır. Bu aşamada bu 15 günlük zaman aralığı içerisinden istenirse tamamı istenirse de bir bölümü içerisinden analiz yapmak mümkün kılınmıştır. Buradaki amaç bütün zaman aralığından elde edilen bir analiz sonucunun, daha ufak zaman aralıklarında da geçerli olup olmadığını deneme imkanı sağlamaktır.

Şekil 4.5 de görülen başlangıç ve bitiş tarihleri SQL komutları kullanılarak veritabanından alınmaktadır. SQL sorguları şu şekildedir:

```
@$sorgu =mysql_query("SELECT min(tarih) FROM data ");
@$baslangic=mysql_result ($sorgu,0,0); // analiz başlangıç tarihi
@$sorgu =mysql_query("SELECT max(tarih) FROM data ");
@$bitis=mysql_result ($sorgu,0,0); // analiz bitiş tarihi
```

Şekil 4.5 Tarih aralığı seçim ekranı

Alınan bu veriler analiz edilmek istenen tarih aralığı seçilecek olan form alanlarına gün ay ve yıl olarak yerleşmektedir. Bu kısımdan 1-15 gün arasında seçim yapılabilmesi sağlanmaktadır.

Ara komutuyla beraber Apriori algoritmasına geçilmektedir. Algoritmanın birinci aşaması olarak satış faturalarındaki ürünler belirlenmekte ve faturalarda geçiş sayıları hesaplanmaktadır. Hesaplama yapılırken step1 isimli geçici tablo oluşturulur. Faturalarda geçiş sayıları yani destekleri hesaplanan ürünler step1 tablosuna girilmektedir. Bunun için gerekli PHP kodu şöyledir:

```
@$soru ="select distinct urunkod from data
        where tarih between  '$tarih1' and '$tarih2'  ";
@$snc = mysql_query($soru);
@$sayi=mysql_num_rows($snc);
```

Bu kısma kadar data tablosundan belirlenen tarih aralığında satılan ürünlerin kodları seçilmekte ve ürün adedi belirlenmektedir.

```
@$tablo=mysql_query("CREATE TABLE `step1`
(`id` INT (3) UNSIGNED DEFAULT '0' NOT NULL AUTO_INCREMENT,
`urunkod` VARCHAR (13) DEFAULT '0',
`destek` INT (3) UNSIGNED DEFAULT '0',
 UNIQUE(`id`))");
```

Bu komutla step1 isimli geçici tablo alanları ve alanlardaki veri tipleri ile beraber oluşturulmaktadır. Bundan sonraki kısım desteklerin hesaplanarak tabloya kayıt edilmesi ve bu esnada kullanıcı için ekrana basılmasıdır. Bu bir döngü ile

sağlanmaktadır. Döngünün çevrim sayısı ürün adetini tutan $sayi değişkeniyle belirlenmektedir.

```
for (@$i=0; $i<=($sayi);$i=$i+2)
        {

        @$urun=mysql_result($snc,$i,0);
        @$urun2=mysql_result($snc,$i+1,0);

        @$adet="SELECT adet FROM data
        WHERE urunkod='$urun'
        and
        tarih between  '$tarih1' and '$tarih2'";

        @$adet2="SELECT adet FROM data
        WHERE urunkod='$urun2'
        and
        tarih between  '$tarih1' and '$tarih2' ";

        @$gelen=mysql_query($adet);
        @$gelen2=mysql_query($adet2);
        @$destek=mysql_num_rows($gelen);
        @$destek2=mysql_num_rows($gelen2);
        $toplam=$toplam+$destek+$destek2;
        @$no=$i+1;
        @$no2=$i+2;

        if($destek>0)
                {
                        echo  "

                                <tr>
                        <td>$no</td>
                        <td width='110' class='f' >$urun</td>
                        <td  class='b'>$destek</td>
                        <td width='50'></td>";

                        @$tabloyaekle=mysql_query("INSERT INTO step1
                        (id, urunkod, destek)
                        VALUES (NULL, '$urun', '$destek')");

                }

        if($destek2>0)
                {
                        echo "
                        <td>$no2</td>
```

```
                    <td class='f'  >$urun2</td>
                    <td class='b' >$destek2</td>
                            </tr> ";

                    @$tabloyaekle2=mysql_query("INSERT INTO step1
                    (id, urunkod, destek)
                    VALUES (NULL, '$urun2', '$destek2')");
                }

}
```

Kodun çalışmasıyla ekrana Şekil 4.6 deki gibi bir ekran görüntüsü gelmektedir.

No	Ürün Kodu	Destek		No	Ürün Kodu	Destek
1	8691693008033	282		2	8691693001010	160
3	8691693006022	38		4	8698562417729	9
5	8690101111150	20		6	4033100001093	87
⋮			⋮		⋮	
113	7312040090754	1		114	7312040050758	1
115	7312040017072	1		116	4030600001257	1
117	8698562404828	1		118	8691693010692	1

Ortalama 32

Destek Değerini Belirleyiniz...

Gönder

Şekil 4.6 Ürünler ve destek değerleri

Bu ekranda destek değeri belirlenmekte ve Gönder düğmesiyle desteği geçemeyen ürünlerin belirlenerek silinmesi için step1-1.php dosyası çalıştırılmaktadır. Şekil 4.7 de destek değeri 220 olarak girilmiş ekran çıktısı verilmiştir. Desteği geçen kayıtlar destek değerleri ile listelenmiştir. Bu esnada step1b geçici tablosu oluşturularak desteği geçen ürünler kayıt edilmektedir. Kullanıcı isterse bu aşamada destek

değerini değiştirebilir ve ya ikili birliktelikleri belirlemek için Devam düğmesiyle step2.php dosyasını çalıştırabilir.

Şekil 4.7 Destek değerini aşan ürünler ve değerleri

Devam düğmesiyle birlikte step2.php dosyası çalışır ve desteği geçen ürünlerin kayıtlı olduğu step1b tablosundaki ürünlerin ikili kombinasyonları alınır ve step2 adında bir tablo oluşturularak bu ikili kombinasyonlar tabloya eklenir. Bu iç içe iki tane for döngüsü ile sağlanır. PHP kodu şu şekildedir:

```
@$snc = mysql_query("select urunkod from step1b");

@$ss= mysql_query("select count(id) from step1b");

@$adet=mysql_result($ss,0);
```

Bu kısma kadar ürünlerin sayısı ve ürünlerin kodları alınmaktadır. Daha sonra step2 geçici tablosu oluşturularak döngülere girilmektedir.

```
@$tablo=mysql_query("CREATE TABLE `step2`
(`id` INT (3) UNSIGNED DEFAULT '0' AUTO_INCREMENT,
`urun1` VARCHAR (13) DEFAULT '0',
`urun2` VARCHAR (13) DEFAULT '0',
`destek` INT (4) UNSIGNED DEFAULT '0', UNIQUE(`id`)) "); // tablo oluştu

for (@$i=0; $i<=($adet-2);$i=$i+1)
        {
        @$urun1=mysql_result($snc,$i);
                for (@$k=$i+1; $k<=($adet-1);$k=$k+1)
                        {
                        @$urun2=mysql_result($snc,$k);
                        @$tabloyaekle=mysql_query("INSERT INTO step2
                                (id, urun1,urun2, destek)
                                VALUES (NULL, '$urun1','$urun2', 0)");
                        }
        }
```

Bu kısımdan sonra ise ikili destek değerlerinin belirlenmesi için veri tablosunu tarayan kod gelmekte ve ardından hesaplanan desteklerle beraber ekrana bastırılmaktadır.

```
@$snc = mysql_query("select id,urun1,urun2 from step2");
@$ss= mysql_query("select count(id) from step2");
```

```
@$sayac=mysql_result($ss,0);

// buraya kadar step2 den veriler alındı ve step 2 deki kayıt sayısı belirlendi.
for (@$t=0; $t<=($sayac-1);$t=$t+1)
        {

        @$id=mysql_result($snc,$t,0);
        @$urun1=mysql_result($snc,$t,1);
        @$urun2=mysql_result($snc,$t,2);

        @$sor = mysql_query("select fat_no from data
                where
                urunkod='$urun2'
                and
                 tarih between  '$tarih1' and '$tarih2'");

        @$say=mysql_num_rows($sor);
        @$destek_kayit=0;

        for (@$m=0; $m<$say;$m=$m+1)
                {

                @$fatura=mysql_result($sor,$m,0);
                @$sor4 = mysql_query("select * from veri
                                        where fat_no='$fatura'");

                for (@$a=2; $a<52;$a=$a+1)
                        {

                        @$kontrol=mysql_result($sor4,0,$a);

                        if($kontrol==$urun1)

                                {

                                @$destek_kayit=$destek_kayit+1;
                                break;

                                }
                        }
                }

        @$tabloyuduzelt=mysql_query("UPDATE step2
                                        SET destek='$destek_kayit'
                                        WHERE id='$id'");


        }
```

Destekler de hesaplandıktan sonra ikili birliktelikler ve destek değerleri ekrana basılmakta ve Şekil 4.8 deki gibi bir ekran görüntüsü elde edilmektedir.

Birli Ürün Destek Degeri : 220

No	Ürün 1	Ürün 2	Destek
1	8691693006015	8691693008033	223
2	8691693006015	8691693002048	200
3	8691693008033	8691693002048	193
4	8691693006015	8690101111426	188
5	8691693008033	8690101111426	180
6	8690101111426	8691693002048	146

Ortalama 188

Ikili Ürün Destek Degerini Belirleyiniz... Gönder

Şekil 4.8 İkili birliktelikler ve destek değerleri

Bu aşamada istenirse destek değeri birli ürün destek değeriyle aynı girilebileceği gibi tablodaki destek değerleri göz önünde bulundurularak değiştirilebilir. Değer belirlenip gönder düğmesiyle beraber desteği aşan ürünleri belirlemek için step2-1.php dosyası çalıştırılmaktadır.

Bu kısımda kodlar step1-1.php ile benzerlik göstermektedir. Desteği aşan ikili birliktelikler ve destek değerleri step2b isimli geçici tabloya girilmektedir. Üçlü birliktelikler oluşturmak için ikili birlikteliklerde geçen ürünler step3temp isimli geçici tabloya aktarılmaktadır. Üçlü birliktelikler için kombinasyon oluşturulurken bu tablodan faydalanılır. Step2-1.php dosyasının çalıştırılmasıyla ekrana Şekil 4.9 daki gibi bir görüntü ekrana gelir.

Ikili Ürün Destek Degeri : 300
Destegi Geçen Kayit Sayisi : 3 (% 14)
Destegi Geçemeyen Kayit Sayisi : 18 (% 86)

No	Ürün 1	Ürün 2	Destek
1	8691693008033	8691693006015	338
2	8691693006015	8691693002048	327
3	8691693008033	8691693002048	318

Ortalama 327
Üçlü Birliktelikleri Belirle ------> Devam

Şekil 4.9 Destek değerinin aşan ikili birliktelikler

Devam düğmesiyle birlikte step3.php çalıştırılır ve üçlü birliktelikler belirlenmeye başlanmaktadır. Üçlü birlikteliklerin oluşturulması için step3temp tablosundaki ürünlerden faydalanılır. Burada bulunan ürünlerin üçlü kombinasyonları alınır. Oluşturulan üçlü kombinasyonların destekleri veri tablosundan taranarak hesaplanmaktadır. Oluşan kombinasyonlar step3 geçici tablosuna eklenmektedir. Destekler hesaplandıktan sonra sonuçlar Şekil 4.10 daki gibi ekrana bastırılmaktadır.

Ikili Ürün Destek Degeri : 185

No	Ürün 1	Ürün 2	Ürün 3	Destek
1	8691693006015	8691693008033	8691693002048	168
2	8691693006015	8691693008033	8690101111426	153
3	8691693006015	8691693002048	8690101111426	139
4	8691693008033	8691693002048	8690101111426	135

Ortalama 148
Üçlü Ürün Destek Degerini Belirleyiniz... Gönder

Şekil 4.10 Üçlü birliktelikler ve destek değerleri

Şekil 4.10 daki ekrandan yeni destek değeri belirlenebilir ve ya ikili destek değeri aynen kullanılabilir. Gönder düğmesiyle beraber step3-1.php çalıştırılmakta desteği geçen ürünler belirlenerek step3b geçici tablosuna eklenmektedir. Ayrıca dörtlü birliktelikler oluşturabilmek için step3 de kayıtlı üçlü kombinasyonlarda yer alan ürünler step4temp isimli geçici tabloya aktarılır. Eğer step4temp tablosunda dörtten az ürün oluştuysa dörtlü birliktelik oluşturacak ürün olmadığından step4.php dosyası çalıştırılmaz . Oluşan üçlü birliktelikler en sık geçen öğe gruplarını oluşturmaktadır.

Bu aşamadan sonra güven değerlerinin hesaplanmasına geçilmektedir. Eğer dört veya daha fazla ürün oluşmuşsa step4.php çalıştırılarak dörtlü kombinasyonlar oluşturulur. Oluşturulan kombinasyonlar veri tablosundan taranarak destek değerleri belirlenmektedir. Oluşan kombinasyonlar step4 geçici tablosuna kayıt edilmektedir. Şekil 4.11 de bu aşamanın ekran görüntüsü verilmiştir.

Üçlü Ürün Destek Degeri : 120

No	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Destek
1	8691693006015	8691693008033	8691693002048	8690101111426	121
2	8691693006015	8691693008033	8691693002048	8691693000044	117
3	8691693006015	8691693008033	8690101111426	8691693000044	85
4	8691693006015	8691693002048	8690101111426	8691693000044	77
5	8691693008033	8691693002048	8690101111426	8691693000044	73

Ortalama 94

Dörtlü Ürün Destek Degerini Belirleyiniz... Gönder

Şekil 4.11 Dörtlü birliktelikler ve destek değerleri

Destek değeri belirlenip Gönder düğmesine basıldığında desteği geçen değerler step4b geçici tablosuna kayıt edilmektedir. Step4b tablosundaki dörtlü

kombinasyonlarda geçen ürünler beşli birlikteliklerde kullanılmak üzere step5temp geçici tablosuna eklenmektedir. Eğer step5temp tablosunda beş elemandan daha az bir ürün adedi oluşmuşsa algoritma sonlanmaktadır ve güven değerleri hesaplanmaktadır. Şekil 4.12 böyle bir durumu göstermektedir.

Dörtlü Ürün Destek Degeri : 190
Destegi Geçen Kayit Sayisi : 1 (% 20)
Destegi Geçemeyen Kayit Sayisi : 4 (% 80)

No	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Destek
1	8691693006015	8691693008033	8691693002048	8691693000044	193

Destegi Geçen Kayitlardaki Ürün Sayisi "4" Oldugundan Besli Birliktelik Olusturulamaz
---
Güven Degerlerini Belirle ------> Devam

Şekil 4.12 Algoritmanın sonlanması

Devam düğmesine basılmasıyla güven değerlerinin hesaplanmasına geçilmektedir. güven değerlerinin hesaplanması için guven.php dosyası çalıştırılmaktadır. Dosyanın çalıştırılmasıyla oluşturulan birliktelik setleri ilgili tablodan okunmaktadır ve ürünler destek değerleri üzerinden işlem yapılarak güven değerleri hesaplanmaktadır. Aşağıda dörtlü birliktelikler için güven değeri hesaplayan PHP kodu görülmektedir.

```
<?
include("./sqlconn.php") ;

@$snc = mysql_query("select * from step4b order by destek desc ");

@$sayi=mysql_num_rows($snc);
```

```
for (@$i=0; $i<=($sayi-1);$i=$i+1)
        {
        @$urun1=mysql_result($snc,$i,1);
        @$urun2=mysql_result($snc,$i,2);
        @$urun3=mysql_result($snc,$i,3);
        $urun4=mysql_result($snc,$i,4);



        @$d1 = mysql_query("select destek from step1 where urunkod='$urun1'");

        @$deger1=mysql_result($d1,0,0);

        @$d2 = mysql_query("select destek from step2
                                where urun1='$urun1'  and urun2='$urun2'");

        @$deger2=mysql_result($d2,0,0);

        @$guven1=($deger2*100)/$deger1;
                $kac=strpos($guven1,".");//virgülü atmak için
                                if($kac!=0)
                                {
                                        $guven1=substr($guven1,0,$kac);
                                }

        @$d3=mysql_query("select destek from step3
                                where urun1='$urun1'  and urun2='$urun2'
                                and urun3='$urun3'");

        @$deger3=mysql_result($d3,0,0);

        @$guven2=($deger3*100)/$deger2;
                                $kac=strpos($guven2,".");//virgülü atmak için
                                if($kac!=0)
                                {
                                        $guven2=substr($guven2,0,$kac);
                                }

        @$d4=mysql_query("select destek from step4
                                where urun1='$urun1'  and urun2='$urun2'
                                and urun3='$urun3' and urun4='$urun4'");

        @$deger4=mysql_result($d4,0,0);

        @$guven3=($deger4*100)/$deger3;

                        $kac=strpos($guven3,".");//virgülü atmak için
```

```
                    if($kac!=0)
                {
                    $guven3=substr($guven3,0,$kac);
                }

echo

?>
<table width="847" border="1" bordercolor="#000000" align="center" >
 <tr>
  <td width="120" class="k">Veri Seti <?echo $i+1;?></td>
  <td colspan="8"> </td>
 </tr>
 <tr>
  <td colspan="9"> </td>
 </tr>
 <tr>
  <td class="f"><? echo $urun1; ?></td>
  <td width="46" align="center"><img src="imge/ok.jpg"></td>
  <td width="30" class="f"><? echo $urun2; ?></td>
  <td width="76"> </td>
  <td width="34"> </td>
  <td width="69"> </td>
  <td width="65" class="b"> </td>
  <td width="59" class="b"> </td>
  <td width="330" class="b">Güven % <? echo $guven1;?></td>
 </tr>
 <tr>
  <td class="f"><? echo $urun1; ?></td>
  <td align="center"><img src="imge/arti.jpg"></td>
  <td class="f"><? echo $urun2; ?></td>
  <td align="center"><img src="imge/ok.jpg"></td>
  <td class="f"><? echo $urun3; ?></td>
  <td> </td>
  <td class="b"> </td>
  <td class="b"> </td>
  <td class="b">Güven % <? echo $guven2;?></td>
 </tr>
  <tr>
  <td class="f"><? echo $urun1; ?></td>
  <td align="center"><img src="imge/arti.jpg"></td>
  <td class="f"><? echo $urun2; ?></td>
  <td align="center"><img src="imge/arti.jpg"></td>
  <td class="f"><? echo $urun3; ?></td>
  <td><img src="imge/ok.jpg"></td>
  <td class="f"><? echo $urun4; ?></td>
  <td class="b"> </td>
  <td class="b">Güven % <? echo $guven3;?></td>
 </tr>
```

```
 <tr>
  <td class="a"><? echo $aciklama1; ?></td>
  <td align="center"><img src="imge/arti.jpg"></td>
  <td class="a"><? echo $aciklama2; ?></td>
  <td align="center"><img src="imge/arti.jpg"></td>
  <td class="a"><? echo $aciklama3; ?></td>
  <td><img src="imge/ok.jpg"></td>
  <td class="a"><? echo $aciklama4; ?></td>
  <td class="b"> </td>
  <td class="a">Güven % <? echo $guven3;?></td>
 </tr>
  <tr>
  <td colspan="9"  height="15" class="f"> </td>
 </tr>

</table>
<?
;
}
?>
```

Çalıştırılan guven.php sonrası oluşan ekran görüntüsü Şekil 4.13 de verilmiştir.

Veri Seti 1								
8691693006015	→	8691693008033						Güven % 78
8691693006015	+	8691693008033	→	8691693002048				Güven % 75
8691693006015	+	8691693008033	+	8691693002048	→	8690101111426		Güven % 72
Veri Seti 2								
8691693006015	→	8691693008033						Güven % 78
8691693006015	+	8691693008033	→	8691693002048				Güven % 75
8691693006015	+	8691693008033	+	8691693002048	→	8691693000044		Güven % 69
Veri Seti 3								
8691693006015	→	8691693008033						Güven % 78
8691693006015	+	8691693008033	→	8690101111426				Güven % 68
8691693006015	+	8691693008033	+	8690101111426	→	8691693000044		Güven % 55

Şekil 4.13 Güven değerleri

Uygulamada data tablosundan yola çıkılarak destek değeri 650 olarak belirlenmiştir. Tarih aralığı ise verilerin tamamını kapsayacak şekilde 5-21 Nisan 2008 olarak girilmiştir.

Birinci adımda 211 adet ürünün destek değerleri hesaplanmıştır. En yüksek destek değerinin 2542 olarak "8691693006015" kodlu ürüne ait olduğu belirlenmiştir. Destek değeri 650 olarak girildiğinde 211 adet üründen 18 tanesinin desteği geçtiği görülmüştür. İkinci adıma geçilerek ikili kombinasyonlar oluşturulmuştur.

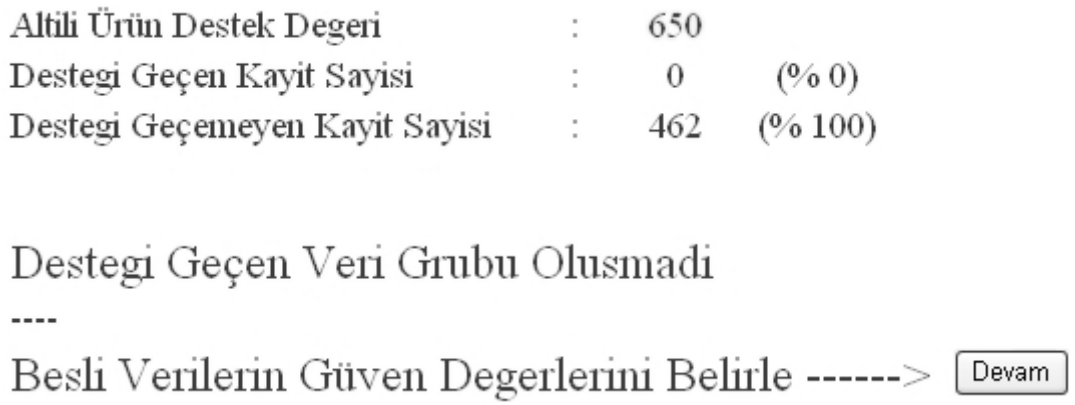
İkinci adımda 153 adet ikili birliktelik belirlenmiş ve destek değerleri hesaplanmıştır. En yüksek destek değerinin 2010 olarak "8691693006015--- 8691693008033" ikili birlikteliğine ait olduğu belirlenmiştir. Destek değeri değiştirilmeden 650 olarak girildiğinde 153 birliktelikten 75 tanesinin desteği geçtiği belirlenmiştir. Üçüncü adıma geçilerek üçlü kombinasyonlar oluşturulmuştur.

Üçüncü adımda 560 adet üçlü birliktelik belirlenmiş ve destek değerleri hesaplanmıştır. En yüksek destek değerinin 1602 olarak "8691693006015---8691693008033---8691693002048" üçlü birlikteliğine ait olduğu belirlenmiştir. Destek değeri değiştirilmeden 650 olarak girildiğinde 560 birliktelikten 99 tanesinin desteği geçtiği belirlenmiştir. Dördüncü adıma geçilerek dörtlü kombinasyonlar oluşturulmuştur.

Dördüncü adımda 1001 adet dörtlü birliktelik belirlenmiş ve destek değerleri hesaplanmıştır. En yüksek destek değerinin 1109 olarak "8691693006015---8691693008033---8691693002048---8690101111426" dörtlü birlikteliğine ait olduğu belirlenmiştir. Destek değeri değiştirilmeden 650 olarak girildiğinde 1001 birliktelikten 58 tanesinin desteği geçtiği belirlenmiştir. Beşinci adıma geçilerek beşli kombinasyonlar oluşturulmuştur.

Beşinci adımda 792 adet beşli birliktelik belirlenmiş ve destek değerleri hesaplanmıştır. En yüksek destek değerinin 774 olarak **"**8691693006015---8691693008033---8691693002048---8691693000044---8691693001010" beşli birlikteliğine ait olduğu belirlenmiştir. Destek değeri değiştirilmeden 650 olarak girildiğinde 792 birliktelikten 8 tanesinin desteği geçtiği belirlenmiştir. Altıncı adıma geçilerek altılı kombinasyonlar oluşturulmuştur.

Altıncı adımda 462 adet altılı birliktelik belirlenmiş ve destek değerleri hesaplanmıştır. En yüksek destek değerinin 599 olarak "8691693006015-8691693008033--8691693002048--8691693000044—8691693001010--8691693000013" altılı birlikteliğine ait olduğu belirlenmiştir. Destek değeri değiştirilmeden 650 olarak girildiğinde desteği geçen ürün grubu oluşmadığından Şekil 4.14 deki görüntü elde edilmiştir.



Şekil 4.14 Altıncı adımın sonlanması

Altılı birlikteliklerden destek değeri olan 650 barajını aşan olmadığından sık geçen birliktelikler kümesini step5b tablosundaki beşli birliktelikler oluşturmuştur. Şekil 4.15 de destek değerini aşan ve en sık geçen birliktelikler olarak belirlenen veri setlerini tutan step5b tablosu görülmektedir.

vm / step5b: 8 Records (8 retrieved)

id	urun1	urun2	urun3	urun4	urun5	destek
1	8691693006015	8691693008033	8691693002048	8691693000044	8691693001010	774
2	8691693006015	8691693008033	8691693002048	8691693000044	8691693000075	730
3	8691693006015	8691693008033	8691693002048	8691693000044	8691693000013	710
4	8691693006015	8691693008033	8691693002048	4032900005447	4032900005805	674
5	8691693006015	8691693008033	8691693002048	8690101111426	8691693000044	668
6	8691693006015	8691693008033	8691693002048	8691693000044	8691693001034	668
7	8691693006015	8691693008033	8691693002048	8691693001010	8691693000013	659
8	8691693006015	8691693008033	8691693000044	8691693001010	8691693000013	651

Şekil 4.15 Sık geçen birliktelikler tablosu

Şekil 4.14 de devam düğmesine basılarak güven değerlerinin hesaplanmasına geçilmiştir. Şekil 4.15 de görüleceği gibi 8 adet beşli birliktelik elde edilmiştir. Bu beşli birliktelikler ve hesaplanan güven değerleri Şekil 4.16 de verilmiştir.

Burada kullanıcının belirleyeceği güven değerini aşan birliktelikler en sık geçen birliktelikleri oluşturmaktadır. Güven değeri % 70 olarak belirlenmiştir. Güven değerini aşan birliktelikler şu şekildedir:

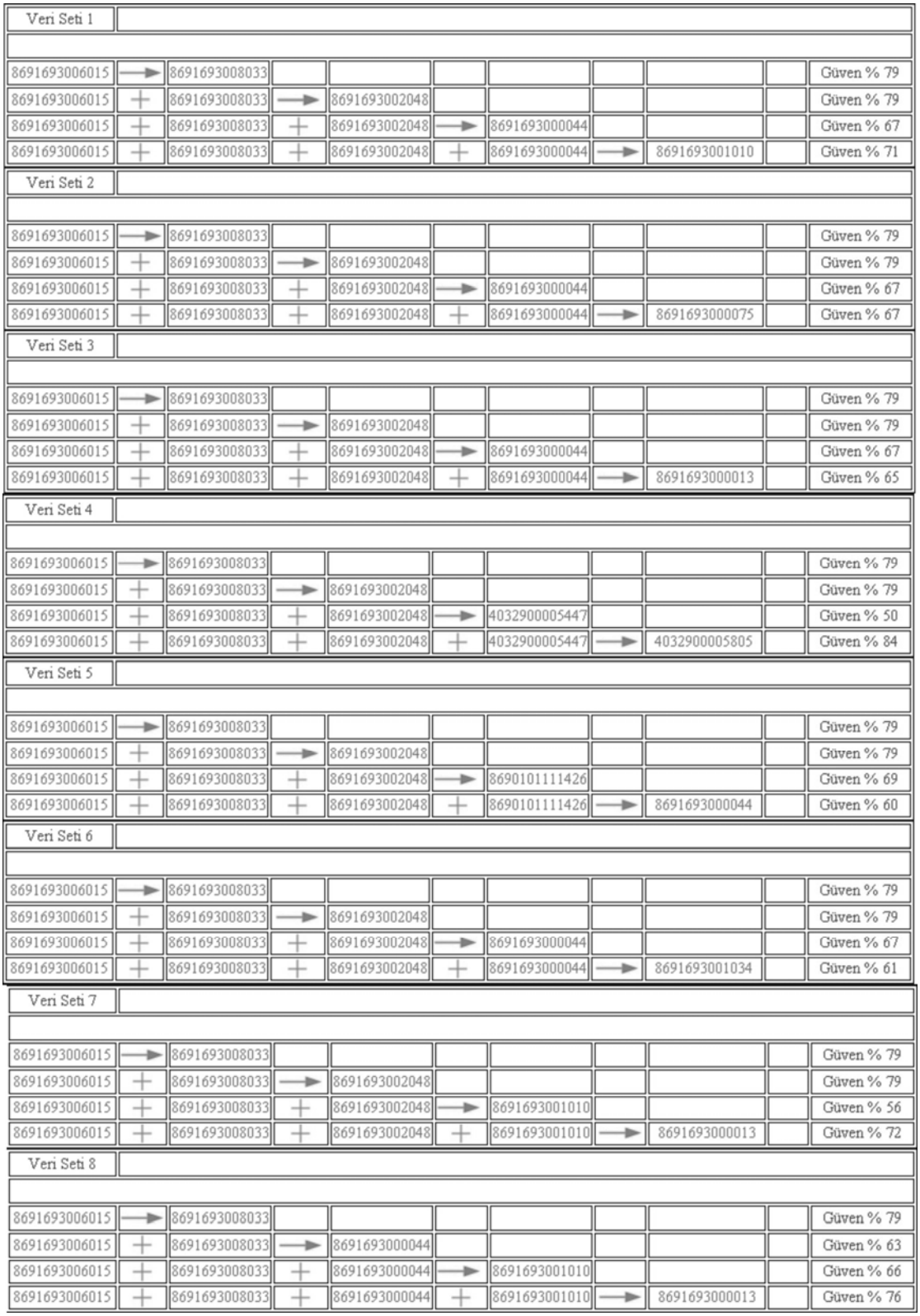
Veri Seti 1:

8691693006015 → 8691693008033 (%79)

8691693006015 ürününü alan müşterilerin %79'u 8691693008033 ürününü de almıştır.

8691693006015 + 8691693008033 →8691693002048 (%79)

8691693006015 ve 8691693008033 ürününü birlikte alan müşterilerin %79'u 8691693002048 ürününü de almıştır.



Şekil 4.16 Sonuç birliktelikleri ve güven değerleri

8691693006015 + 8691693008033 + 8691693002048 + 8691693000044 → 8691693001010 (%71)

8691693006015, 8691693008033, 8691693002048 ve 8691693000044 ürünlerini birlikte alan müşterilerin %71'i 8691693001010 ürününü de almıştır.

Veri Seti 4 :

8691693006015 + 8691693008033 + 8691693002048 + 4032900005447 → 4032900005805 (%84)

8691693006015, 8691693008033, 8691693002048 ve 4032900005447 ürünlerini birlikte alan müşterilerin %84'ü 4032900005805 ürününü de almıştır.

Veri Seti 7 :

8691693006015 + 8691693008033 + 8691693002048 + 8691693001010 → 8691693000013 (%72)

8691693006015, 8691693008033, 8691693002048 ve 8691693001010 ürünlerini birlikte alan müşterilerin %72'si 8691693000013 ürününü de almıştır.

Veri Seti 8 :

8691693006015 + 8691693008033 + 8691693000044 + 8691693001010 → 8691693000013 (%76)

8691693006015, 8691693008033, 8691693000044 ve 8691693001010 ürünlerini birlikte alan müşterilerin %76'sı 8691693000013 ürününü de almıştır.

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Sektör çalışanları elde edilen bu birliktelik-ilişki verileri ile üretim planlama aşamalarını tekrardan yapılandırabilir, satışta yer alan ürün gamını düzenleyebilir, satış yerindeki reyon düzenlemesini ürün birlikteliklerini dikkate alarak satış rakamlarını destekleyecek şekilde değiştirebilir, dönemsel satışlardan elde ettikleri sonuçlar doğrultusunda dönemsel satış kampanyaları düzenleyebilirler.

Bu çalışmada Apriori algoritması bir firmanın merkez biriminden müşterilere yapılan satış verileri üzerinde kullanıldı. Algoritmanın veriler üzerinde çalışması için bu çalışma kapsamında yerel ağ ortamında çalışan bir uygulama yazılımı geliştirilmeye çalışıldı. Yazılım algoritma süreçlerinin aşamalı olarak çalıştırılmasına imkan tanıyacak şekilde gerçekleştirilmiştir. Bu özelliğin sağladığı ortamla, algoritmanın her bir aşamasında gerçekleşen işlemlerin daha rahat anlaşılabilmesine ve her bir sürece ait sonuçların izlenebilmesine imkan tanınması hedeflenmiştir..

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

KAYNAKLAR

[1] ALATAŞ, B., AKIN, E., "Veri Madenciliğinde Yeni Yaklaşımlar", YA/EM'2004 - Yöneylem Araştırması /Endüstri Mühendisliği - XXIV Ulusal Kongresi, Gaziantep – Adana, 15-18 Haziran 2004.

[2] HUDAIRY H., "Data mining and decision making support in the governmental sector", Master Thesis, Faculty of Graduate School of The University of Louisville, Kentucky, 2004; 1-5.

[3] YALÇINTAŞ, G., "Veri Madenciliği", Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2003.

[4] FAYYAD, U., PIATETSKY-SHAPIRO, G., SMITH, P., "The KDD process for extracting useful knowledge from volumes of data.", Communications of ACM, 39(11), 1996; 27-34.

[5] LARSE, D.T., Discovering Knowledge in Data: An Introduction at Data Mining, Jhn Wiley & Sns Inc., 2005; 42-70

[6] AKPINAR, H., "Veritabanlarında bilgi keşfi ve veri madenciliği", İstanbul Üniversitesi İşletme Fakültesi Dergisi, 2000; 29: 1-22

[7] Current data mining applications / percentage in different industries, http://www.kdnuggets.com/polls/2003/data_mining_applications_industries.html, 2005.

[8] AYDOĞAN F., "E-ticarette veri madenciliği yaklaşımlarıyla müşteriye hizmet sunan akıllı modüllerin tasarımı ve gerçekleştirimi", Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2003; 12-16.

[9] QUINLAN, J. R. The effect of noise on concept learning, San Mateo, CA: Morgan Kauffmann Inc., 1986; 2: 149-166.

[10] HULTEN, G., SPENCER, L., DOMINGOS, P., "Mining time-changing data streams", 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Fransisco, CA: ACM Pres, 2001; 97-106.

[11] KONA, H. V., "Association Rule Mining Over Multiple Databases: Partioned and Incremental Approaches", The University of Texas, Arlington , 2003.

[12] WEI C., CHIU T., "Turning telecommunications call details to churn prediction: a data mining approach,", Expert Systems with Applications, 2002; 23:103-102.

[13] ARYEETEY, K., "Data analysis and predictive modelling using the variable precision rough set approach", Master Thesis, Faculty of Graduate Study and Research of University of Regina, Canada, 2003; 28-33.

[14] HUI, S., JHA G., "Application data mining for customer service support", Information and Management, 2000; 38: 1-13.

[15] ADRIANS, P., ZANTINGE, D., "Data Mining", Addison Wesley Longman Limited, Harlow, 1996; 159.

[16] TANTUĞ, A.C., "Veri Madenciliği ve Demetleme", Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2002.

[17] RAMKUMAR, G. D., SWAMI, A., "Clustering Data Without Distance Functions", IEEE Bulletin of the Technical Committee on Data Engineering, March 1998; 21: 9-14.

[18] ANAND, A., "A study and comparison of data clustering techniques", Master Thesis, Faculty of The Graduate School of The University of Texas, El Paso, 2003; 21-22 .

[19] HAN, J., KAMBER, M., "Data Mining: Concepts and Techniques" Morgan Kaufmann, 2001; 45-53.

[20] SEVER, H., OĞUZ, B., "Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım", Bilgi Dünyası, Ekim 2002 ; 3:173-204.

[21] ERKAN, Ü., EYÜP, S., EMRE, Ç., HARUN, U., AHMET, A., "Web Basın Verilerine Apriori Algoritması Uygulanarak Düzenli Birliktelik Kurallarının Bulunması", 4rd International Advanced Technologies Symposium, September 2005.

[22] AGRAWAL, R., IMIELINSKI, T., SWAMI, A., "Mining association rules between sets of items in large databases", ACM SIGMOD Conference on Management of Data, Washington, 1993.

[22] ZHU, H., "On-Line Analytical Mining Of Association Rules", Master Thesis, Simon Fraser University, Canada, 1998.

[23] AGRAWAL, R., SRIKANT, R., “Fast Algorithms for Mining Association Rules”, Proceedings of the VLDB, Santiago de Chile, Chile, 1994.

ÖZGEÇMİŞ

Fatih Şen, 04.12.1983 de Sakarya' da doğdu. İlk, orta ve lise eğitimini İstanbul'da tamamladı. 2001 yılında Esenler İbrahim Turhan Lisesi'nden mezun oldu. 2001 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümünü 2005 yılında bitirdi. Aynı yıl Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Bilişim Mühendisliği bölümüne giriş yaptı ve hala devam etmektedir.