T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI

BİRLİKTELİK KURALLARI İLE SEPET ANALİZİ VE UYGULAMASI

Yüksek Lisans Tezi

GÜNEŞ GÜRGEN

İSTANBUL, 2008

T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI

BİRLİKTELİK KURALLARI İLE SEPET ANALİZİ VE UYGULAMASI

Yüksek Lisans Tezi

GÜNEŞ GÜRGEN

Danışman: Doç. Dr. AHMET METE ÇİLİNGİRTÜRK

İSTANBUL, 2008

Marmara Üniversitesi
Sosyal Bilimler Enstitüsü Müdürlüğü

Tez Onay Belgesi

EKONOMETRİ Anabilim Dalı İSTATİSTİK Bilim Dalı Yüksek Lisans öğrencisi GÜNEŞ GÜRGEN nın BİRLİKTELİK KURALLARI İLE SEPET ANALİZİ VE UYGULAMASI adlı tez çalışması ,Enstitümüz Yönetim Kurulunun 19.06.2008 tarih ve 2008-10/25 sayılı kararıyla oluşturulan jüri tarafından oybirliği/oyçokluğu ile Yüksek Lisans Tezi olarak kabul edilmiştir.

	Öğretim Üyesi Adı Soyadı	İmzası
Tez Savunma Tarihi	: 26/06/2008	
1) Tez Danışmanı :	**DOÇ. DR. AHMET METE ÇİLİNGİRTÜRK**	.....
2) Jüri Üyesi :	**DOÇ. DR. DİLEK ALTAŞ**	.....
3) Jüri Üyesi :	**PROF. DR. NURİ ALKAN SOYAK**	.....

ÖZET

Günümüzde sürekli depolanan karmaşık ve büyük veri yığınlarının içerisinde bulunan değerli bilgilerin ortaya çıkarılması için veri madenciliği modelleri sıkça kullanılmaktadır. Bu tez çalışmasında öncelikle veri madenciliği için kullanılan modeller ve teknikler açıklanmıştır. Veri madenciliği modellerinden Birliktelik Kuralları ve Algoritmaları ayrıntılı olarak incelenmiştir. Tezin uygulama kısmında Türkiye'de ki market zincirlerinden birinin 7 günlük fişleri kullanılmıştır. Bu fişlerdeki ürünlerin birbirleri ile olan ilişkileri, Birliktelik Kuralları ile Sepet Analizi uygulaması ve Apriori algoritması ile belirlenmiştir. Ürünler arasında bulunan birliktelikler ürün satışlarının, dolayısı ile gelirin arttırılması için kullanılması amaçlanmıştır.

ABSTRACT

Nowadays, data mining models are being used to obtain valuable information of complex and big data accumulations frequently stored. In this thesis, first of all, models and techniques of data mining have been explained. Especially, Association Rules and algorithms of data mining models have been focused throughout the whole study. Application part of this thesis consists of a research in which the receipt of one of the market chains has been examined through 7 days. The relationship between the items of these receipts has been determined by Basket Analysis with Association Rules and Apriori algorithm. Associations between items have been proposed to be used to increase item sales and income, off course.

ÖNSÖZ

Tez çalışmam süresince kısıtlı zamanını bana ayırarak desteğini hiç eksiltmeyen, değerli bilgileri ile bana yol gösteren ve motivasyonumu her zaman yüksek tutmamı sağlayan tez danışmanım Doç. Dr. Ahmet Mete Çilingirtürk'e teşekkür ediyorum.

Ayrıca mesai saatleri dışındaki dinlenme zamanlarından fedakârlık ederek, tezim için bana yardımcı olan Eşref Mutlu ve Neşe Tanören'e teşekkür ediyorum.

İÇİNDEKİLER

SAYFA

ÖZET…….....i

ABSTRACT……..……ii

ÖNSÖZ…….……....iii

İÇİNDEKİLER…….....iv

TABLO LİSTESİ…….……..viii

ŞEKİLLER LİSTESİ…….....x

KISALTMALAR VE SİMGELER……..…....xi

GİRİŞ……1

BÖLÜM I.

1. VERİ MADENCİLİĞİ……...2

1. VERİ MADENCİLİĞİNE GENEL BAKIŞ…….2

1. Veri Tabanları ve Veri Ambarları……...3

2. Veri Tabanlarında Bilgi Keşfi…….4

3. Veri Madenciliği Süreci……..7

2. VERİ MADENCİLİĞİNDE KULLANILAN MODELLER ve TEKNİKLER……..7

1. Tahmin Edici Modeller Ve Teknikler…….8

1. Sınıflandırma……...9

2. Regresyon Analizi……...9

3. Karar Ağaçları……...10

4. Yapay Sinir Ağları…….12

5. Genetik Algoritmalar ve Evrimsel Programlama ….13

6. Bellek Temelli Nedenleme……13

2. Tanımlayıcı Modeller ve Teknikler……..……...14

1.2.2.1. Kümeleme Analizi……14

1.2.2.2. İlişki Analizi…….15

1.2.2.2.1. Ardışık Zamanlı Örüntüler……...15

1.2.2.2.2. Birliktelik Kuralları……..16

BÖLÜM II

2. BİRLİKTELİK KURALLARI…….18

2.1. BİRLİKTELİK KURALLARININ TANIMI…….18

2.2. BİRLİKTELİK KURALLARININ MATEMATİKSEL MODELİ…...19

2.2.1. Güven Ve Destek Ölçüleri……..21

2.2.2. Birliktelik Kurallarının Kullanım Alanları…….24

2.2.3. Birliktelik Kurallarının Aşamaları……..25

2.3.4. Genelleştirilmiş Birliktelik Kuralları…….….26

2.3.5. Uzaysal Birliktelik Kuralları…….….28

2.3.6. Nicel Birliktelik Kuralları……..….29

2.3.7. Aralık Veri Birliktelik Kuralları…….....33

2.3.8. Çoklu Min-destek Birliktelik kuralları……...34

2.3.9. Mültimedya Birliktelik Kuralları……35

2.3.10. Maksimal Birliktelik Kuralları……..……...35

2.3. BİRLİKTELİK KURALLARI İÇİN KULLANILAN ALGORİTMALAR……...36

2.3.1. Ardışık Algoritmalar

2.3.1.1. AIS Algoritması……...36

2.3.1.2. SETM Algoritması……..38

2.3.1.3. Apriori Algoritması…….……40

2.3.1.4. Apriori-TID Algoritması……46

2.3.1.5. Apriori-Hybrid Algoritması……47

2.3.1.6. OCD Algoritması ……...48

2.3.1.6. Partition Algoritması……...49

2.3.1.7. Sampling Algoritması ……51

2.3.1.8. DIC Algoritması ……51

2.3.1.9. CARMA ……51

2.3.2. Paralel Algoritma……...……52

2.3.2.1. Veri Paralellik Algoritması

2.3.2.1.1. CD Algoritması ……...52

2.3.2.1.2. PDM Algoritması ……...53

2.3.2.2. İş Paralellik Algoritması

2.3.2.2.1. DD Algoritması ……..53

2.3.2.2.2. IDD Algoritması ……54

2.3.2.3. Diğer Paralel Algoritmalar…….……54

2.3.2.3.1. Candidate Distribution Algorithm …….……..54

2.3.2.3.2. SH Algoritması…….……55

2.3.4. Paralel Algoritmaların Geleceği……..56

2.4. ALGORİTMALARIN SINIFLANDIRILMASI……56

2.5. ALGORİTMALARIN KARŞILAŞTIRILMASI……...57

BÖLÜM III

3. TÜKETİM ÜRÜNLERİ PAZAR SEPET ANALİZİ ÖRNEĞİ……...60

3.1. VERİLERİN TOPLANMASI ve DÜZENLENMESİ……..….…….61

3.2. ANAKÜTLE ve ÖRNEKLEM…….…….……61

3.3. YÖNTEM ve ALGORİTMA…….62

3.4. AMAÇ…….……...…64

3.5. ALGORİTMANIN ELLE ÇÖZÜMÜ……64

3.6. BULGULAR……..69

SONUÇ ve ÖNERİLER……...81

KAYNAKÇA……...83

TABLO LİSTESİ

Sayfa No

Tablo 1: Ürün Hareket Tablosu……...23

Tablo 2: Nicel Vasıflar için Örnek "Kişi" Tablosu…….29

Tablo 3: Boolean Birliktelik Kuralları Problemin Kodlanması……..30

Tablo 4: Tablo 3'de Verilen Ürünlerin Bazı Büyük Ürün Kümeleri…….31

Tablo 5: Aralık Veri Seti…….33

Tablo 6: Marketten Yapılan Alışveriş Bilgilerini İçeren D Veritabanı……...42

Tablo 7: SPSS Clementine Programına Ait Bir Arayüz……..63

Tablo 8: Veri Seti Kesiti…….…...……..64

Tablo 9: Sıralanmış Veri Kesiti……...…….65

Tablo 10: C1 aday kümesinden oluşturulan L2 büyük ürün kümesi……..65

Tablo 11: İkili kombinasyonlar için C2 aday kümesi, destek ve güven

Değerleri……66

Tablo 12: Destek eşiğini sağlayan ürün kümeleri……..……...…66

Tablo 13: C3 aday kümesinden oluşan L3 büyük ürün kümeleri……..…67

Tablo 14: C4 aday kümesinden oluşan L4 büyük ürün kümeleri……..……..67

Tablo 15: Veri Kesitinden Elde Edilen Birliktelik Kuralları……..…….68

Tablo 16: Ürün gruplarının hareketlerdeki dağılımı……..…….71

Tablo 17: Ürün grupları için birliktelik kuralları……..……..73

Tablo 18: Kategorileştirilmiş veri seti…….……...74

Tablo 19: Ürün kategorilerinin hareketlerdeki dağılımı…….……75

Tablo 20: Ürün Kategorileri için Birliktelik Kuralları…….……..79

ŞEKİL LİSTESİ

Sayfa No

Şekil 1: Veri Tabanları ve Veri Ambarları Oluşumu……..4

Şekil 2: Veri Madenciliği Süreci…….…6

Şekil 3: Basit Bir Karar Ağacı Yapısı……...11

Şekil 4: Yapay Bir Sinir Hücresi……..12

Şekil 5: Market Sepet Sınıflandırması……..27

Şekil 6: Apriori Algoritmasının Gösterimi……..……43

Şekil 7: Partition Algoritması Kullanarak Büyük Ürün Kümelerinin Bulunması…….……50

Şekil 8: Tam, Düzenli, Ürün Kümesi Algoritmalarının Sınıflandırılması…....56

Şekil 9: Ürün grubu dağılımı……..……..71

Şekil 10: Ürün Kategorilerinin dağılımı……...76

Şekil 11: Web Grafiği ile Kategoriler arasındaki ilişkiler……77

Şekil 12: Sıklığı yüksek olan ilişkiler……...78

KISALTMALAR ve SİMGELER

I : Ürün Kümesi

I_d: Ürün (Nitelik)

m: Ürün Sayısı

D: İşlem Veritabanı

s: Destek

α: Güven

T: D içindeki Hareket

X,Y: Ürün Kümeleri

$X \Rightarrow Y$: Birliktelik Kuralı

L: Büyük Ürün Kümelerinin Kümesi

l: Büyük Ürün Kümesi

L_k: k Değerindeki Büyük Ürün Kümelerinin Kümesi

l_k: k Değerindeki Büyük Ürün Kümesi

C_k: k Değerindeki Aday Kümeler

$\overline{L}_k$: k Değerindeki Büyük Ürün Kümelerinin Kümesi ve Onları Kapsayan TID (bir işlemin tek tanımlayıcısı)

$\overline{C}_k$: k Değerindeki Aday Ürün Kümelerinin Kümesi ve Onları Kapsayan TID (bir işlemin tek tanımlayıcısı)

D^i: D Veritabanı İçin i Kısmı

X^i: D^i Bölümü İçin Ürün Kümesi

L^i: D^i Bölümü İçin Büyük Ürün Kümelerinin Kümesi

C^i: D^i Bölümü İçin Aday Ürün Kümelerinin Kümesi

p: Bölümlerin Sayısı

GİRİŞ

Son yıllarda hızla gelişen teknoloji ile birlikte marketler, işletmeler ve diğer kuruluşlar sahip oldukları verileri depolama imkânı bulmaktadır. Yaptığımız her işlem bu kurumların bünyelerinde saklanarak veri setleri oluşturmaktadır. Her geçen gün büyüyen bu veri setlerinin değerlendirilmesi ve bunlardan yararlı bilgilerin elde edilmesi beklenen bir durum olmuştur. Bu aşamada Veri Madenciliğinde Bilgi Keşfi ve Veri Madenciliği terimleri gündeme gelmektedir.

Bu tezin amacı biriken bu büyük veri setlerinde ürünler arasındaki ilişkiyi bulmaya yarayan bir veri madenciliği modeli olan *"Birliktelik Kuralları"* nın incelenmesi ve bir market için derlenen veri setindeki ürünlerin arasındaki birliktelik kurallarını bulmaktır.

Tezin birinci bölümünde Veri Madenciliği ve Veri Madenciliğinde Bilgi Keşfi süreçleri anlatılmış ve veri madenciliği yöntemlerinden bahsedilmiştir.

Tezin ikinci bölümünde ise veri madenciliği modellerinden Birliktelik Kuralları matematiksel modeli ile birlikte anlatılmıştır. Ayrıca bu kural için geliştirilmiş Algoritmalardan ayrıntılı olarak bahsedilmiş, birbirlerine olan üstünlükleri vurgulanarak uygulama için uygun algoritma seçimi yapılmıştır.

Tezin son bölümümde, Türkiye'de perakende sektöründe faaliyet gösteren bir zincir marketten derlenen veriler ile Birliktelik Kuralı Analizi uygulanmış ve ürünler arasındaki ilişkiler bulunmuştur.

BÖLÜM I.

1. VERİ MADENCİLİĞİ

1. VERİ MADENCİLİĞİNE GENEL BAKIŞ

Günümüzde birçok alanda elde edilen veriler sahip olunan teknolojik olanaklar sayesinde depolanabilmekte ve bunun sonucunda oluşan veri yığınları gün geçtikçe artmaktadır. Sürekli artmakta olan bu veri yığınları beraberinde bazı sorunları da getirmektedir.

Çoğu zaman iyi kullanılmamaları durumunda veri tabanlarında tutulan veri, insanlar için problem haline de gelebilmektedir. Toplanan veri miktarı büyüdükçe ve toplanan verilerdeki karmaşıklık arttıkça, daha iyi çözümleme tekniklerine olan gereksinim de artmaktadır. Bu noktada **Veri Madenciliği** (Data Mining) ve **Veri Tabanlarında Bilgi Keşfi** (Knowledge Discovery in Databases) kavramları ortaya çıkmaktadır.

Veri madenciliğinin birçok farklı tanımı yapılmaktadır. Bunlardan en çok kullanılanlar aşağıda belirtilmektedir.

Veri madenciliği, büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanılarak aranmasıdır.

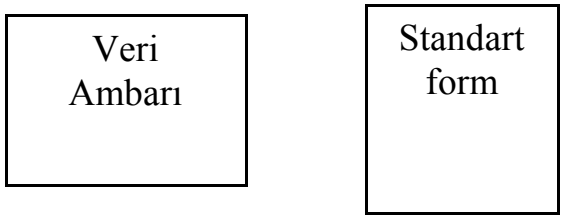
Veri madenciliği, büyük veri tabanlarından, çok net olmayan, üstü kapalı, önceden bilinmeyen ancak potansiyel olarak kullanışlı olabilecek bilginin çıkarılmasıdır.

Veri madenciliği, büyük veri tabanlarında örüntülerin, birlikteliklerin, anormalliklerin ve çeşitli yapıların yarı otomatik bir sistem ile keşfidir.

Veri madenciliği, büyük miktarda ve oldukça hızlı toplanan verilerin, çeşitli analizler sonucunda anlamlı bilgilere dönüştürülmesi noktasında devreye giren bir süreçtir. Veri madenciliği tanımları incelendiğinde; bu tanımlarda ortak olan unsurların ilki çok fazla miktarda verinin veri ambarında tutulması, ikincisi ise bu verilerden anlamlı bilgiler elde edilmesidir.

1. Veri Tabanları ve Veri Ambarları

Veri madenciliği büyük miktarda veri inceleme amacı üzerine kurulmuş olduğu için veri tabanları ile yakından ilişkilidir. Gerekli verinin hızla ulaşılabilecek şekilde amaca uygun bir şekilde saklanması ve gerektiğinde hızla ulaşılabilmesi gerekir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş ve işlemeye daha uygun bir özetini saklamayı amaçlar (Şekil 1).



Günlük
Veri tabanları

Şekil 1 Veri Tabanları ve Veri Ambarları Oluşumu

Günlük veri tabanlarından istenen özet bilgi seçilerek ve gerekli ön işlemeden sonra veri ambarında saklanır. Ardından amaç doğrultusunda gerekli veri ambardan alınarak veri madenciliği çalışması için standart bir forma çevrilir.

2. Veri Tabanlarında Bilgi Keşfi

Veri madenciliğinde bilgi keşfi, verilerden doğru, yeni, faydalı ve anlaşılır modeller ve kalıplar (pattern) elde etmekte kullanılan özel bir süreçtir.

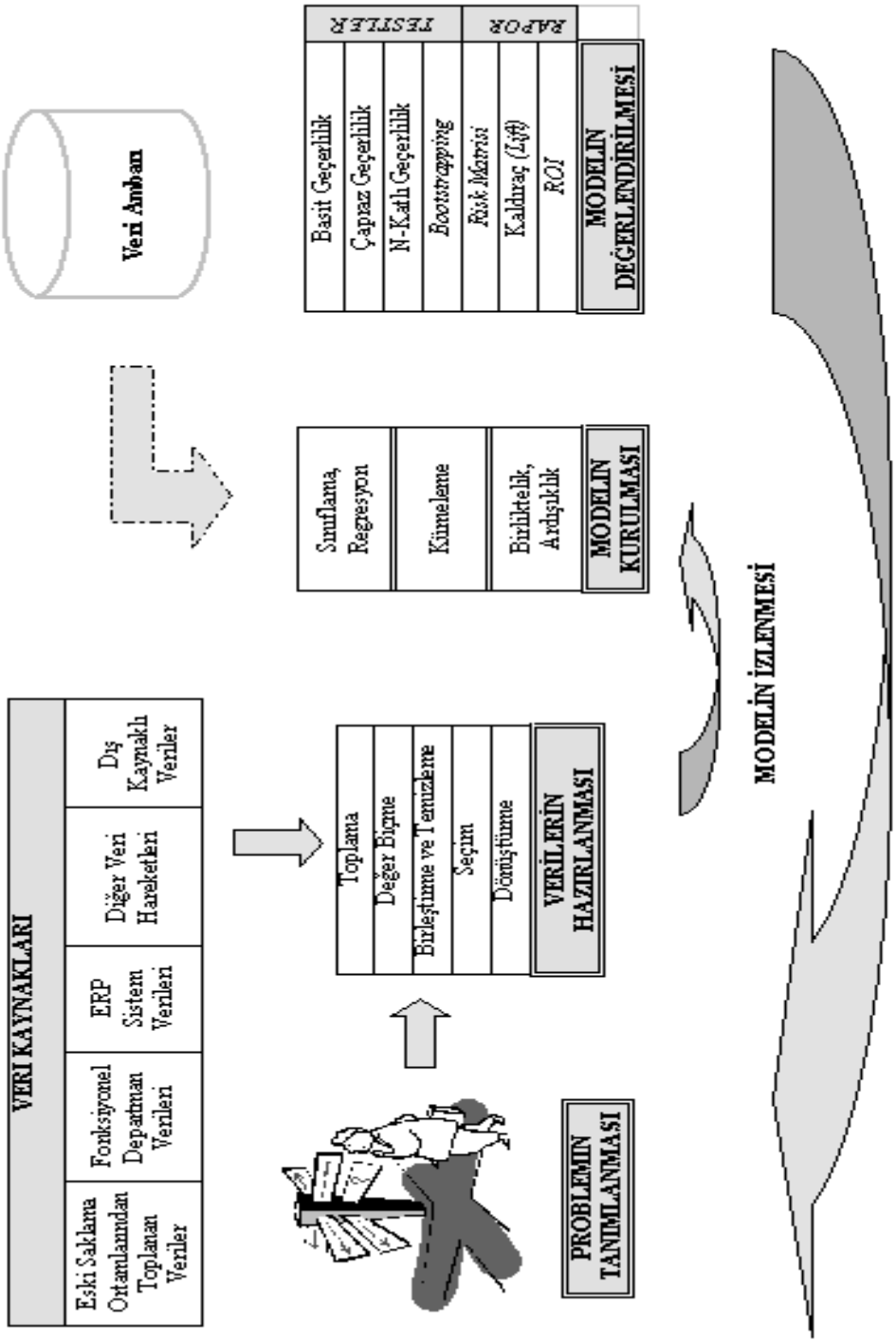
Veri madenciliği terimi ile veri madenciliğinde bilgi keşfi genelde birbirinin yerine kullanılsa da, veri madenciliği bu uzun sürecin sadece bir adımını oluşturmaktadır.

Veri madenciliği, veri tabanlarında bilgi keşfi sürecinin özünü temsil etse de, genellikle bu sürecin küçük bir parçasını kapsar. Bu nedenle, veri madenciliği baştanbaşa bilgi keşfinin sadece bir adımını oluşturur.

Veri madenciliğinde bilgi keşfi sürecinin adımları şu şekildedir;

- Veri Önişlemleri (Data Preprocessing)

- Veri Seçme ve Dönüştürme
    - Veri madenciliği konusu ile ilgili bilgi seçimi.
    - Madencilik yapılacak veri türünün belirlenmesi.
    - Veriler arasında hiyerarşik yapı ve genellemelerin belirlenmesi.
    - Veri madenciliği sonunda bulunacak bilgi için yenilik ve ilginçlik ölçümü yöntemlerinin belirlenmesi.
    - Veri madenciliği sonunda bulunacak veri için sunum ve görselleştirme araçlarının belirlenmesi.
- Veri Madenciliği
- Örüntü Değerlendirme (Pattern Evaluation)
- Bilgi Sunumu (Knowledge Presentation)



Şekil 2 Veri Madenciliği Süreci

KAYNAK: K. Yaralıoğlu, "Veri Madenciliği", Dokuz Eylül Üniversitesi Web Yayın, http://www.deu.edu.tr/userweb/k.yaralioglu/dosyalar/ver_mad.doc, 15 Mayıs 2008

3. Veri Madenciliği Süreci

Şekil 2'de veri tabanlarında bilgi keşfi için veri madenciliği süreci verilmiştir. Şekilden de görüleceği gibi problemin tanımlanmasının ardından farklı veri kaynaklarından elde edilen veriler hazırlanarak ilgili model belirlenmektedir. Model kurulması ve değerlendirilmesi ile elde edilen sonuçlar için rapor hazırlanarak veri madenciliği süreci tamamlanmış olmaktadır. Farklı veri kaynaklarından elde edilen veriler hazırlandıktan sonra hemen kullanılmaya ihtiyaç duyulmayabilirler. Hazırlanan bu veriler veri ambarlarında depolanarak daha sonra tanımlanabilecek bir problemin çözümünde kullanılabilirler. Yine, veri ambarından seçilen ilgili veriler için model kurulması ve değerlendirilmesi aşamalarını tamamladıktan sonra hazırlanan rapor ile veri madenciliği süreci tamamlanmaktadır.

2. VERİ MADENCİLİĞİNDE KULLANILAN MODELLER ve TEKNİKLER

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

Bu bilgi keşfi sürecince, kullanılabilecek pek çok veri madenciliği modeli söz konusudur. Uygulanacak olan model, veriye ve araştırmacının ulaşmak istediği sonuca göre değişiklik gösterebilir.

Kullanılan modeller tahmin edici, tanımlayıcı veya her iki birden olabilir. Her tahmin edici model aynı zamanda tanımlayıcı olmak zorunda değildir. Bazı tanımlayıcı modeller de tahmin amaçlı kullanılmamaktadır. Veri madenciliğinde kullanılan modeller;

- Kümeleme Analizi (Tanımlayıcı)

- Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler (Tanımlayıcı)
- Sınıflandırma (Tahmin edici)
- Regresyon Analizi ve Zaman Serileri Analizi (Tahmin edici)

olarak belirtilebilir.

1. Tahmin Edici Modeller Ve Teknikler

Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yaralanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler müşterilerinin özellikleri, bağımlı değişken değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır.

1. *Sınıflandırma*

Veri madenciliğinde sınıflandırma işlemi, eldeki mevcut verileri, önceden belirlenen bir özelliğe göre sınıflara ayırmak ve yeni eklenecek verilerin hangi sınıfa dahil olacağını tayin etme işlemidir. Diğer bir deyişle, yeni karşılaşılan bir girdinin hangi sınıfa dahil olacağına karar verme işlemidir.

Sınıflandırma bir ürünün özellikleri ile müşteri özelliklerinin eşlenmesi için kullanılabilir. Böylece bir müşteri için ideal ürün veya bir ürün için ideal müşteri profili çıkarılabilir. Örneğin;

- Genç bayanlar küçük araba satın alır; yaşlı ve zengin erkekler, büyük ve lüks araba satın alır

Böyle bir sınıflandırmadan faydalanarak firmalar pazarlama stratejilerini daha özele indirgeyerek belirleyebilir ve ilgili kitleye daha rahat ulaşabilir.

2. Regresyon Analizi

Regresyon genellikle geçmişteki değerleri temel alarak gelecekteki değerleri tahmin etmek için kullanılan tahmin edici modeller sınıfında yer alan bir tekniktir.

Başka bir anlatımla, herhangi bir değişkenin (bağımlı değişken), bir veya birden fazla değişkenle (bağımsız-açıklayıcı) arasındaki ilişkinin matematik bir fonksiyon şeklinde yazılması ve bu fonksiyon yardımıyla bağımlı değişkenin ulaşacağı değerin tahmin edilmesidir.

Sınıflandırma ve regresyon, önemli veri sınıflarını ortaya çıkaran veya gelecekteki veri eğilimlerini geçmiş veriler yardımı ile tahmin eden modelleri kuran veri analiz yöntemleridir. Sınıflandırma; belli özelliklere göre gruplanmış değerleri tahmin ederken, regresyon; süreklilik gösteren değerlerin tahmin edilmesinde kullanılır.

Örneğin, sınıflandırma modeli banka kredi uygulamalarının güvenli veya riskli olmalarını sınıflamak amacıyla kullanılırken, regresyon modeli geliri ve mesleği bilinen potansiyel müşterilerin herhangi bir alış veriş esnasında yapacakları harcama davranışlarını tahmin etmek için kullanılabilir. Sınıflandırma ve Regresyon modellerinde kullanılan tekniklerden bazıları;

- Karar ağaçları (Decision trees)
- Yapay sinir ağları (Artificial neural networks)
- Genetik algoritmalar ve Evrimsel Programlama (Genetic algorithms and Evolutionary programming)

- Bellek temelli nedenleme (Memory based reasoning)

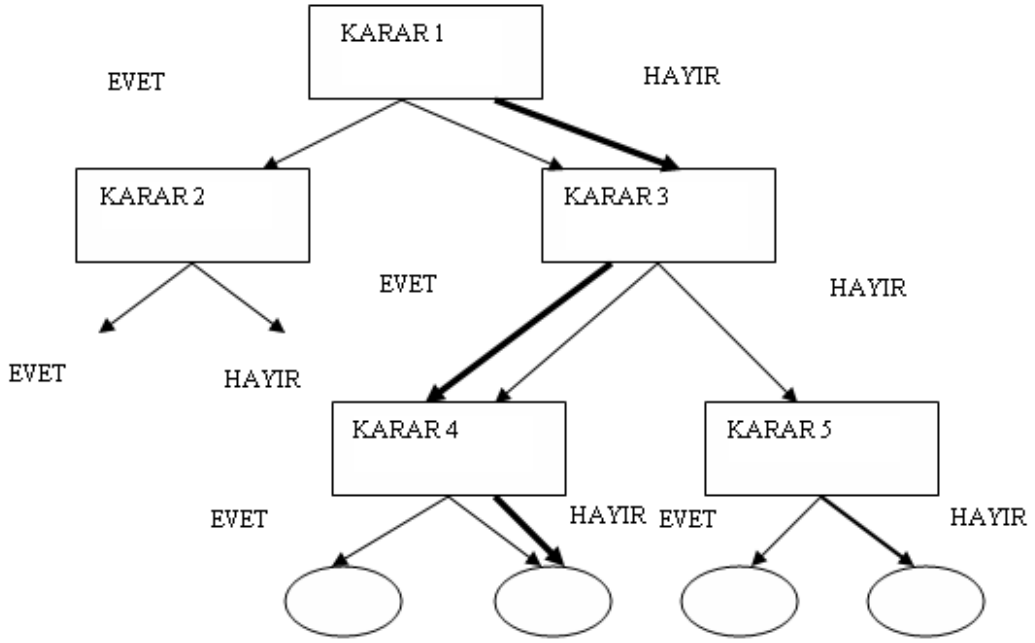
olarak sıralanabilir.

3. Karar Ağaçları

Karar ağaçları, veri madenciliğinde kuruluşlarının ucuz olması, yorumlanmalarının kolay olması, veri tabanı sistemleri ile kolayca entegre edilebilmeleri ve güvenilirliklerinin iyi olması nedenleri ile sınıflama modelleri içerisinde en yaygın kullanıma sahiptir.

Karar ağacı akış diyagramı benzer bir ağaç yapısında olup, her bir dal bir testin sonucu, terminaller düğümleri (yaprak düğümleri) ise sınıfları temsil eder.

Karar ağacı, her bir düğümün, göz önünde bulunduran veri üzerinde, bir test ya da karar belirttiği bir araçtır. Bir veriyi sınıflamak için kök düğümünden başlanır, terminal düğüme (veya yaprağa) ulaşana kadar iddialar takip edilir. Terminal düğüme ulaşıldığı zaman bir karar oluşturulur. Şekil 3'te basit bir karar ağacı yapısı görülmektedir.



Şekil 3 Basit Bir Karar Ağacı Yapısı

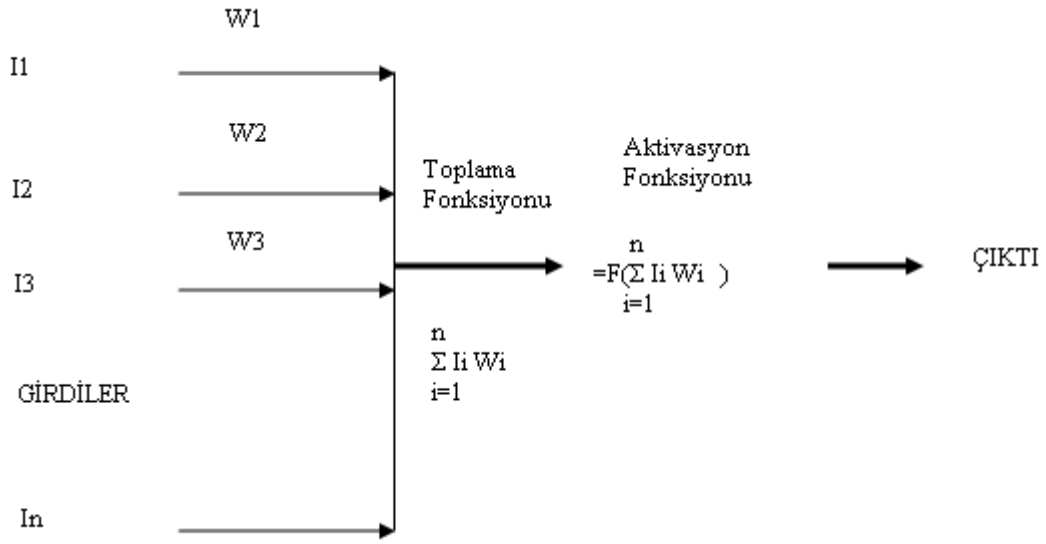
Kaynak: U. Tuğba Şimşek, "Veri Madenciliği ve Müşteri İlişkileri Yönetiminde (CRM) Bir Uygulama", (Basılmamış Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, 2006), s.33

4. Yapay Sinir Ağları

Yapay sinir ağları, insan beyninden esinlenerek tasarlanan, ağırlıklı bağlantılar yardımıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem elemanlarından meydana gelen paralel ve dağıtılmış bilgi işleme yapıları olarak belirtilebilir.

Başka bir anlatımla yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir.

Aşağıda bulunan Şekil 4'de basit bir yapay sinir hücresi gösterilmiştir. N adet girdi değeri (I), kendi ağırlık değeri (W) ile çarpılarak toplanır ve çıktıyı elde etmek için aktivasyon fonksiyonu ile işlem yapılır.



Şekil 4. Yapay Bir Sinir Hücresi

Kaynak: U. Tuğba Şimşek, "Veri Madenciliği ve Müşteri İlişkileri Yönetiminde (CRM) Bir Uygulama", (Basılmamış Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, 2006), s.38

Tüm yapay sinir ağları bu temel yapıdan türetilmiştir.

5. Genetik Algoritmalar ve Evrimsel Programlama

Genetik algoritmalar ve evrimsel programlama, doğal evrimdeki gözlemlenen prensiplerden esinlenilmiş en iyileme stratejileridir.

Bu yöntemde bir problemi çözebilmek amacıyla ilk olarak rasgele başlangıç çözümleri belirlenmektedir. Daha sonra bu çözümler birbirleri ile eşleştirilerek performansı daha yüksek (daha iyi) çözümler meydana getirilmektedir. Bu şekilde sürekli çözümler birleştirilerek yeni çözümler aranmaktadır. Bunu yapmaktaki amaç, organizma popülasyon evriminde olduğu gibi, en iyi çözümlerin bulunduğu çözüm kümesini daha iyi hale getirmektir.

Çözüm uzayından tesadüfî olarak seçilen noktalar üzerinde operatörler yardımı ile daha iyi noktalara ulaşmak amaçtır. Bu arama iyi sonuç elde edilemeyene kadar devam etmektedir. Çözüm uzayı çok büyük problemler için, bu arama sırasında iyi sonuç vermeyecek alanlarda gereksiz arama yapılmaması, hızlı bir şekilde çözüme ulaşmak açısından oldukça iyi bir tekniktir.

6. Bellek Temelli Nedenleme

Bellek temelli nedenleme tekniği, verilen bir problemi doğrudan doğruya geçmiş tecrübeleri ve çözümleri kullanarak çözmeye çalışan bir teknolojidir. Durum, genellikle daha önceden karşılaştırılmış ve çözülmüş özel bir problemdir. Yeni bir problem verildiğinde, bellek temelli nedenleme tekniğinde öncelikle karşılaşılıp saklanmış durumlar incelenir ve benzer olanlar var mı diye bakılır. Eğer benzer durumlar bulunmuşsa, bunların çözümü yeni probleme uygulanır ve problem gelecekte referans olmak üzere saklanır. En yakın k komşu algoritması bu yöntem için önerilebilecek en uygun yöntemdir.

Bu algoritmada tüm örneklemler bir örüntü uzayında saklanır. Algoritma, bilinmeyen bir örneklemin hangi sınıfa dahil olduğunu belirlemek için örüntü

uzayını araştırarak bilinmeyen örnekleme en yakın olan k örneklemi bulur. Yakınlık Öklid uzaklığı ile tanımlanır. Daha sonra, bilinmeyen örneklem, k en yakın komşu içinden en çok benzediği sınıfa atanır.

2. Tanımlayıcı Modeller Ve Teknikler

Tanımlayıcı modellerde amaç; 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.

1.2.2.1. Kümeleme Analizi

Kümeleme analizi, birinci amacı "gözlem birimlerini benzer özelliklerine göre gruplamak" olan çok değişkenli istatistik tekniklerden biridir. Kümeleme analizinde elde edilen bir küme içindeki gözlem birimleri, önceden belirlenmiş bir özellik bakımından birbirine benzemektedir. Dolayısıyla elde edilen kümedeki gözlem birimleri homojendir.

Kümeleme işlemi sınıflandırma işleminin aksine, veri kümesini önceden sınıflara ayırmaz, bunun yerine veriler dağılımlarına göre irdelenerek doğal sınıflandırmalar oluşturur. Kümeleme işleminin sınıflandırma işleminden en önemli farkı önceden belirlenmiş sınıflar ya da sınıf tanımları olmamasıdır. Kümeleme işleminde temel prensip sınıf içi benzerliği maksimum, sınıflar arası benzerliği minimum yapmaktır.

1.2.2.2. İlişki Analizi

Birliktelik kuralları ve ardışık zamanlı örüntüler, pazarlama amaçlı olarak *Market sepet analizi* (*market basket analysis*) tekniği ile veri madenciliğinde yaygın olarak kullanılmaktadır.

1.2.2.2.1. Ardışık Zamanlı Örüntüler

Ardışık zamanlı örüntü, birbirleri ile ilişkisi olan ve birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Belli bir olayın gerçekleşmesinden sonra aynı olayla ilintili bir başka olayın gerçekleşmesi ardışık zamanlı örüntü kavramını oluşturur.

Örneğin;

- X ameliyatı yapıldığında 15 gün içinde %45 olasılıkla Y enfeksiyonu oluşacaktır.
- Çamaşır makinesi alan bir müşteri 6 ay içerisinde, %70 olasılıkla kurutma makinesi de alacaktır. Mağaza %70 olasılığı arttırmak için çamaşır makinesinin alımından 4 ay sonra, kurutma makinelerinde %10 indirim uygulayabilir.

1.2.2.2.2. Birliktelik Kuralları

Birliktelik kuralları satış-pazarlamadan, ürün katalog tasarımına kadar birçok alanda kullanılmaktadır. Örneğin, herhangi bir ürün satın alırken, bu ürünün yanında başka bir ürün ya da ürünlerin satın alınması, bu ürünler arasındaki bağlantıyı ortaya koyar. Bu bağlantıların ortaya çıkarılması ve bunun bir kural olarak ortaya konması ise birliktelik kuralları konusuna girer. Literatürde bu tarz çalışmalara *market sepeti analizi* denilmektedir. Market sepeti analizi müşterilerin alışveriş alışkanlıklarının veritabanındaki bilgiler aracılığıyla ortaya çıkartılması işlemidir. Bu alışveriş

alışkanlıklarının ortaya çıkartılması alış-veriş merkezindeki ürünlerin yerleşmesi, marketin alanının tasarımı ve markette sergilenecek ve satılacak olan ürünlerin belirlenmesinde yardımcı olmaktadır.

Birliktelik kuralları, işlemlerden oluşan ve her bir işlemin de elemanlarının birlikteliğinden oluştuğu düşünülen bir veri tabanında, bütün birliktelikleri tarayarak, sık tekrarlanan birliktelikleri veritabanından ortaya çıkarmaktadır.

Birliktelik kuralı çıkarmada bazı algoritmalar mevcuttur. Bu algoritmalarda minimum güvenirlik ve destek değerlerini sağlayan birliktelik kuralı çıkarımı sağlanmaktadır.

Birliktelik kuralının matematiksel modeli ve ilgili algoritmalar hakkında Bölüm II' de ayrıntılı bilgiler verilecektir.

BÖLÜM II

BİRLİKTELİK KURALLARI

Birliktelik kuralı, geçmiş verilerin analiz edilerek bu veriler içindeki birliktelik davranışlarının tespiti ile geleceğe yönelik çalışmalar yapılmasını destekleyen bir yaklaşımdır. 90'lı yılların başına kadar saklanan satış verilerinde ürün ve müşteri verisi çok nadir yer alırken, genelde mali açıdan önemli olan tutarsal gelir verilerinin depolanması yapılıyordur. 90'lı yılların başından itibaren veri toplama uygulamalarındaki gelişmeler doğrultusunda firmaların satış noktalarında yeni teknoloji otomatik ürün veya müşteri tanıma sistemleri (barkod ve manyetik kart okuyucular) yaygınlaşmaya başlamıştır. Bu tip teknolojik gelişmeler, bir satış hareketine ait verilerin satış esnasında toplanmasına ve elektronik ortamlara aktarılmasına olanak tanımıştır. Günümüzde 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, iskonto vb.) yer alır ve ayrıca hareket numarası tekildir. Bazı kuruluşlar bu tip bilgileri içeren veritabanlarını pazarlama alt yapılarının önemli parçalarından biri olarak görmekte ve bu verileri kullanmak için çaba harcamaktadır.

2.1. BİRLİKTELİK KURALLARININ TANIMI

Birliktelik kuralında, müşterilerin alışveriş esnasında satın aldıkları ürünler arasındaki birliktelik-ilişki bağlarını bularak, müşterilerin satın alma alışkanlıklarının tespit edilmesi amaçlanmaktadır. Keşfedilen bu birliktelik-ilişki bağıntıları sayesinde satıcılar daha etkin ve kazançlı satışlar yapabilme imkanına sahip olmaktadır. Birliktelik kuralı, bir ilişkide bir niteliğin aldığı değerler arasındaki bağımlılıkları, anahtarda yer almayan diğer niteliklere göre gruplama yapılmış verileri kullanarak bulur. Keşfedilen örüntüler veri kümesinde sıklıkla birlikte geçen nitelik değerleri arasındaki ilişkiyi gösterir.

Birliktelik kurallarının kullanıldığı en tipik örnek market sepeti analizidir. 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 etkin satış stratejileri geliştirebilirler.

Örneğin market sepeti analizi ile makarna alan müşterilerin %80 olasılıkla ketçap aldığı şeklinde bir kural bulunabilir. Bunun üzerine market yöneticileri makarna ve ketçap raflarını yan yana koyarak ya da "makarna alanlara ketçap fiyatında indirim yapmak" gibi kampanyalar düzenleyerek satışlarını arttırabilir.

2.2. BİRLİKTELİK KURALLARININ MATEMATİKSEL MODELİ

Birliktelik kuralının matematiksel modeli Arawal, Imienlinski ve Swami tarafından 1993 yılında verilmiştir. Bu modelde; $I = \{I1, I2, ..., Im\}$, m farklı niteliğin bir kümesini; D, her T hareketinin tek bir tanımlayıcısının olduğu bir veri tabanını belirtmektedir ve bir maddeler kümesini kapsamaktadır, öyle ki $T \subseteq I$ olmaktadır. TID her harekete ait tekil (unique) numarayı, k-öğe kümesi k adet ürün

içeren ürün kümesini temsil etmektedir. X bir ürün kümesi olmak üzere T hareketi X ürün kümesini ancak ve ancak $X \subseteq T$ şartını sağlıyorsa içerir. Burada $X \cap Y = \emptyset$; $X,Y \subset I$ ve "ürün kümeleri" olarak adlandırılan ürünlerin oluşturduğu kümelerdir. Bir birliktelik kuralı $X \Rightarrow Y$ şeklinde gösterilir. Bu gösterim ile X, Y'yi belirler (X ürününü içeren hareketler kümesi Y'yi içeren hareketler kümesi içinde kapsanır) veya Y, X'e bağımlıdır denir (Y kümesinin varlığı X kümesinin var olmasına bağlıdır). Hareket numaraları gruplandırılarak elde edilen ürünler arasındaki bağımlılık ilişkisinin yüzde yüz doğru olması beklenemez. Benzer şekilde, çıkarsama yapılan kuralın eldeki hareketler kümesinin önemli bir kısmı tarafından desteklenmesi istenir. Bu nedenlerden dolayı, $X \Rightarrow Y$ eşleştirme kuralı kullanıcı tarafından minimum değeri belirlenmiş güvenilirlik (α: confidence) ve destek (s: support) eşik değerlerini sağlayacak biçimde üretilir. $X \Rightarrow Y$ eşleştirme kuralına, α güvenilirlik, s destek ölçütü iliştirilir ve biçimsel olarak Φ (D) = $< X \Rightarrow Y, \alpha, s>$ ile gösterilir. Burada D örneklemi; $X \Rightarrow Y$ birliktelik ilişki kuralını; α eşik değeri, ilgili kuralın minimum güvenilirliğini (X ürünlerini içeren hareketlerin en az %α 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. Güvenilirlik metriği, birliktelik kuralının doğruluk derecesini, destek metriği ise kuralda yer alan öğelerin (ürünlerin) geçiş sıklığını gösterir. Yüksek güvenilirlik ve destek değerine sahip kurallara güçlü kurallar adı verilir. Birliktelik kuralı çıkarımı büyük veritabanlarından güçlü ilişkilendirme örüntülerinin elde edilmesini gerektirir.

Birliktelik kuralı formülsel olarak şu şekilde tanımlanabilir;

$$X_1, X_2, \ldots, X_m \quad \Rightarrow \quad Y_1, Y_2, \ldots, Y_n \qquad (2.1)$$

Buradaki, X_j ve Y_j yapılan iş veya nesnelerdir. Bu kural genellikle "X_1, X_2, …, X_m meydana geldiğinde, sık olarak Y_1, Y_2, …, Y_n aynı olay veya hareket içinde yer aldığı anlamına gelmektedir.

Örneklendirecek olursak; bir zincir market şubesi için yapılan veri madenciliği sonucu elde edilen kural aşağıdaki gibidir.

"Ürün(X, "yeşillik") ⇒ Ürün(Y, "sebze")"

Burada X bir hareketteki değişkeni simgelemektedir. Bu kuralda, yeşillik alan müşterilerin aynı zamanda sebze almaya yöneldikleri belirtilmektedir.

Başka bir örnek; aşağıdaki kural üç boyutlu bir veri ambarından gelmektedir:

Yaş, Meslek ve Ürün.

"Yaş(X, "12–17") ^ Meslek(X, "öğrenci") ⇒ Ürün(X, "oyun konsolu-playstation")"

Bu kural ile "12–17" yaşları arasındaki öğrenciler en çok oyun konsolu (playstation) almaktadır" anlamına gelmektedir. Bu örneğimiz çok boyutlu bir birliktelik kuralıdır. Hem "yaş" hem de "meslek" boyutunun ürün boyutu ile ilişkisi incelenmektedir. Bu nedenle farlı boyutları ifade ederken "^" simgesi kullanılmaktadır.

2.2.1. Güven (Confidence) ve Destek (Support) Ölçüleri

Bir birliktelik kuralının desteği (s), $X \cup Y$ 'yi kapsayan hareketlerin sayısının, veritabanındaki toplam hareketlerin sayısına yüzde cinsinden oranı olarak tanımlanır.

$$S = \frac{n(X \cup Y)}{n(D)} \quad (2.2)$$

Bu sebeple, eğer bir kuralın desteğinin %5 olduğunu söylersek, bu, toplam hareketlerin %5'i $X \cup Y$ 'yi kapsıyor demektir. Destek, bir birliktelik kuralının istatistiksel anlamıdır. Eğer markette satılan ürünlere ait işlemlerin %5'inden daha azı fıstık ezmesi ve ekmek kombinasyonundan oluşuyorsa, market müdürü muhtemelen fıstık ezmesi ve ekmek arasında nasıl bir ilişki olduğu konusunda endişe etmeyecektir. Birliktelik kuralları için genellikle yüksek bir destek istenirken, olay her zaman böyle değildir. Örneğin, birliktelik kuralını, bir arızadan önce oluşan olayın temeline dayanarak, telekomünikasyon bağlantılarındaki arızayı önceden tahmin etmek için kullanıyorsak, bu hadiseler çok sık olmasa dahi, aradaki bu ilişkiyi kurması açısından birliktelik kuralları önem teşkil etmektedir.

Hareketlerin verilen belli bir sayısı için güven (α), $X \cup Y$ 'yi kapsayan hareketlerin sayısının, X'i kapsayan hareketlerin sayısına yüzde cinsinden oranı olarak tanımlanır.

$$\alpha = \frac{n(X \cup Y)}{n(X)} \qquad (2.3)$$

Bu sebeple, eğer bir kural %85 güvene sahipse, bu, X'i kapsayan hareketlerin %85'i Y'yi de kapsıyor demektir. Bir kuralın güveni, X ve Y arasındaki veri kümelerinin karşılıklı ilişkisinin derecesini gösterir. Güven, bir kuralın gücünün ölçüsüdür. Birliktelik kuralları için genellikle yüksek güven gerekir. Eğer bir grup olay, bağlantıda arıza olmadan önce zamanın küçük bir yüzdesini oluşturuyorsa veya bir ürün fıstık ezmesiyle beraber çok nadir satın alınıyorsa, aradaki bu ilişki market yönetimi açısından çok fayda sağlamayabilir. Bir veritabanından, birliktelik kurallarının veri madenciği yoluyla çözümlenmesi, kullanıcı tarafından belirlenmiş güven ve desteği karşılayan tüm kuralların bulunmasını içerir.

Bir örnek vermek gerekirse; tablodaki alış hareketlerine bakarak toplam alış hareketlerine göre {Bisküvi, Yoğurt} ile Kola arasındaki ilişki şu şekilde açıklanabilir; (Tablo 1)

Tablo 1 Ürün Hareket Tablosu

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

Kaynak: F. Cemal Özçakır, "Müşteri İşlemlerindeki Birlikteliklerin Belirlenmesinde Veri Madenciliği Uygulaması", (Basılmamış Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, 2006), s.23

$$\text{Destek-support} = \frac{\text{(Bisküvi, Yoğurt, Kola)}}{\text{Toplam Hareket}} = \frac{2}{5} = 0{,}4$$

$$\text{Güven-confidence} = \frac{\text{(Bisküvi, Yoğurt, Kola)}}{\text{(Bisküvi, Yoğurt)}} = \frac{2}{4} = 0{,}5$$

Bu eşitliklerden de anlaşılacağı gibi, {Bisküvi, Yoğurt}⇒ Kola kuralının destek değeri %40, güven değeri ise %50'dir.

Bisküvi ve yoğurdun, kola ile toplam harekette bulunma olasılığı %40 ve bisküvi ve yoğurt alan birinin ayrıca kola da alma olasılığı %50'dir.

2.2.2. Birliktelik Kurallarının Kullanım Alanları

Bir veri madenciliği yöntemi olarak birliktelik kuralları, market satış analizlerinde, ticarette, mühendislikte, tıpta ve finansal konularda kullanılabilir. *Market Sepet Analizi (Market Basket Analysis)* en çok kullanıldığı alanlardan biridir. Müşteri alım alışkanlıklarına ve perakendecilik esaslarına göre kararlar alınmasını sağlar; hangi ürünün indirime konacağı, katalogların nasıl tasarlanacağı, raflarda ürünlerin nasıl dizileceği vb.

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 yazıcı da alma eğilimdeler ise bu ürünlerin yakın yerlere konulması iki ürünün satış oranlarını da arttırabilir. Diğer alternatif bir stratejide, bilgisayar ve yazıcı 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.

Han ve Kamber, birliktelik kuralını aşağıdaki kriterleri taşıyan çok değişik yollar ile sınıflandırmışlardır:

- Kuralda yakalanan değerlerin tiplerine göre: Eğer bir kural nesnelerin varlığı ve yokluğu arasındaki ilişki ile ilgili ise, bu durumda *doğru-yanlış(boolean)birliktelik kuralı'*dır. Bu kurallar sepet analizinden elde edilirler.

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

"Yaş (X, "30..39") ^ Gelir (X, "42K..48K) ⇒ alışlar (X, "Yüksek çözünürlüklü TV")"

- Kuralın içerdiği verinin boyutlarına göre: Bir birliktelik kuralındaki nitelikler ya da nesneler bir boyutu temsil ediyorlarsa, o zaman *tek boyutlu birliktelik kuralı* denir.

"Alışlar(X, "bilgisayar") ⇒ alışlar(X, "finansal_yönetim_yazılımı)"

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

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

"Yaş(X, "30..39") ⇒ alışlar(X, "dizüstü_bilgisayar")"

"Yaş(X "30..39") ⇒ alışlar(X, "bilgisayar")"

Alınan ürünler farklı soyutlama katmanlarında referans edilmişlerdir. (Örneğin bilgisayar, dizüstü bilgisayarın üst-katman soyutlamasıdır.)

2.2.3. Birliktelik Kurallarının Aşamaları

Birliktelik kuralı iki aşamalıdır:

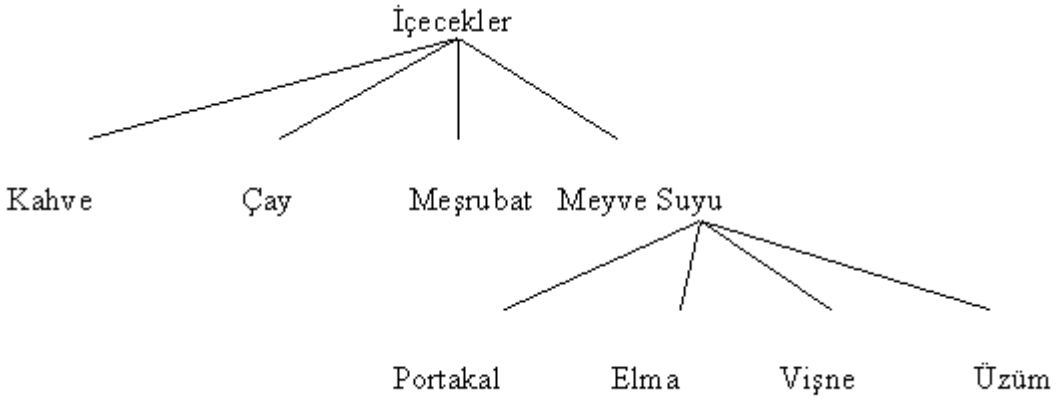
- *Tüm sık geçen ürün kümelerinin bulunması:* Tanıma göre her ürün kümesinin sık geçenler kümesinde yer alabilmesi için, her nesnenin destek değerinin önceden tanımlanmış olan min_destek değerinden büyük olması gerekir.

- *Sık geçen ürün kümelerinden güçlü ilişki kurallarının yaratılması:* Tanıma göre, bu kurallar min_destek ve min_güven durumunu sağlamalıdır.

2.3.4. Genelleştirilmiş Birliktelik Kuralları

Genelleştirilmiş birliktelik kuralları, sınıflandırmada farklı seviyedeki farklı birliktelik kurallarının oluşması için verinin bir hiyerarşik sınıflandırmasının (kavram hiyerarşisi) varlığını kullanır. Şekil 5, market sepet verileri üzerine bir sınıflandırma örneği vermektedir. Burada içecekler, kahve, çay, meşrubat ve meyve suyu olarak gruplara ayrılmıştır. Meyve suyu da portakal, elma, vişne ve üzüm olarak ayrılmıştır. Birliktelik kuralları oluşturulduğunda, onları mevcut hiyerarşik seviyelerden herhangi birinde oluşturabiliriz. Beklendiği gibi, kurallar sıralandırmada yüksek seviyedeki ürünlerden oluşturulduğunda, hem destek hem de güven artacaktır. Verilen bir işlem veritabanında, farklı ürünler için çoklu sınıflandırmalar ve hatta aynı ürün için çoklu sınıflandırmalar olabilir. Bir X ⇒ Y⇒

genelleştirilmiş birliktelik kuralı, Y'deki herhangi bir ürünün X'dekilerin atası olamaması durumu dışında düzenli bir birliktelik kuralının ifadesiyle tanımlanır. Bir ürünün *atası* ifadesi, herhangi bir sıralamada o ürünün üzerinde olacağı anlamına gelmektedir. Bir süpermarket genel olarak meşrubatlarla ilgili birliktelikleri bulmak veya bu meşrubatları türlerine göre ayırarak tanımlamak (kola gibi) isteyebilir. Genelleştirilmiş birliktelik kuralları bunun gerçekleşmesini çok iyi şekilde sağlar ve tüm birliktelik kurallarının (farklı sınıflandırmalardaki farklı seviyedekiler de dahil) bulunmasını temin eder.



Şekil 5 Market Sepet Sınıflandırması

Kaynak: M.H. Dunham, Y. Xiao, L. Gruenwald, Z.Hossain, "A Survey Of Association Rules", Southern Methodist University, University of Oklahoma, 2001, s.41

Genelleştirilmiş birliktelik kuralı problemi tüm sınıflandırmaların tüm seviyeleri için birliktelik kuralları oluşturur. Bunu yapmak için bir yaklaşım, her işlemi almak ve herhangi bir hiyerarşide tüm ürünleri her bir ürünün üzerinde tutmak için her ürünü genişletmektir. Bu pek kullanılmayan yaklaşım çok pahalı olmakla beraber bu yaklaşıma ek olarak daha etkin başka algoritmalar önerilmiştir. Bunlardan biri olan Cumulate (Yığın), her işleme eklenmesi gereken ataların sayısını azaltmak için birkaç optimizasyon stratejisi kullanır. Başka bir yaklaşım olan Stratification (Tabakalaşma), sınıflandırmadaki seviyelerine göre ürün kümelerinin sayımını yapar ve sayılacak ürün sayısını azaltmak için sınıflandırmada ürünler arasındaki ilişkileri kullanır.

2.3.5. Uzaysal Birliktelik Kuralları

Bir özellik kümesinin sahip olduğu başka bir özellik kümesinde ima ettiği özelliklerin tanımını içerir. Örneğin; yapılan bir çalışmada, Dallas' daki bir ailenin ortalama geliri, White Rock gölü yakınında yaşayan aileler için 100,000$ dır.

Uzaysal veritabanları depolanan veriyi ilgilendiren konum bilgilerini içerir. Bu, enlem-boylam çifti, sokak adresi, posta kodu veya diğer coğrafi veriler şeklinde olabilir. Uzaysal veri madenciliği, geleneksel veri madenciliğiyle aynı tarzdaki problemleri incelerken, problemlerin ifadesi ve potansiyel çözümler işin içinde uzaysal veriler olduğu gerçeğine dayanarak değiştirilebilir. Örneğin, uzay işlemleri (içinde, yanında, komşusu, vs.) veritabanındaki hareketler arasındaki ilişkiyi tanımlamakta kullanılabilir. Bir X ⇒ Y⇒ *uzaysal birliktelik* kuralı, aralarından bazıları uzaysal olan X ve Y'nin gösterim kümesi olduğu bir birliktelik kuralıdır. Her iki X ve Y gösterimi T hareketi için doğruysa, veritabanında T için bir uzaysal birliktelik kuralı sağlanmış olur. Güven ve destek ile ilgili tanımlar diğer düzenli birliktelik kurallarındakiler ile aynıdır. Farz edelim ki bir veritabanı belli bir ilçedeki kamu okulları hakkında bilgi içersin. Bu veritabanı kamu tesisleri (parklar, okullar, belediyeye ait binalar, vs.), coğrafik özellikler (nehirler, göller, vs.), özel binalar ve kamuya ait alt yapılar (yollar, köprüler, vs.) hakkında bilgiler içersin. Buna göre bir uzaysal birliktelik kuralı aşağıda olduğu gibidir:

"Okul(T, "İlkokul") ^ İskan gelişimi(T, "Yakın") ⇒ Park(T, "Bitişik")"

Bu kurala göre iskan gelişimine yakın bir ilkokul, aynı zamanda bir parka da bitişiktir. Market sepet verilerinin tersine, bir uzaysal birliktelik kuralının geçerliliğini belirlemek için incelenenin dışındaki başka hareketlere de bakmamız gerekebilir. T hareketinde, bazı niteliklere göre değer alarak gösterileceği üzere kurumun bir ilkokul olup olmadığını görmemiz açısından bakmamız gerekir. Ancak uzaysal gösterimlerin (bu olayda yakınında ve bitişik olan) gerçekliğini yorumlamak

veritabanındaki (ya da diğer veritabanlarındaki) diğer hareketlere bakmamızı gerektirebilir. Bu şekilde, uzaysal gösterimlerin geçekliğini yorumlamak pahalı ve zor bir işlem olabilir. Uzaysal veri madenciliğini geliştiren daha etkin kılan bir yaklaşımsa, ilk aşamasında uzaysal ilişkilerin kabaca yorumlanmasıyla uzaysal gösterimlerin yaklaşık olarak sağlanmasını inceleyen iki aşamalı bir tekniktir.

2.3.6. Nicel Birliktelik Kuralları

Birliktelik kuralları için yapılan çalışmalar önceleri verilerin kategorik olduğunu düşünülerek geliştirilmiştir. Verinin hem nicel hem kategorik olacağı düşüncesinin oluşması ile *Nicel birliktelik kuralları* ortaya çıkmıştır.

Nicel veriler belirli kurallarla aralıklara bölünerek kategoriler oluşturulur ve ardışık tamsayı olarak kodlanır, böylece aralıkların sırası korunur. Veriler bölümlendirilmezse yine her biri ardışık tamsayı numaralarla kodlanır ve sırası korunur.

Tablo 2 Nicel Vasıflar için Örnek "Kişi" Tablosu

Tanım	**Yaş**	**Evlilik**	**Araba Sayısı**
1000	21	Hayır	0
2000	23	Evet	1
3000	24	Hayır	1
4000	26	Evet	2
5000	29	Evet	2

Kaynak: M.H. Dunham, Y. Xiao, L. Gruenwald, Z.Hossain, "A Survey Of Association Rules", Southern Methodist University, University of Oklahoma, 2001, s.44

Tablo 2, örnek "Kişi" tablosunu üç vasıfla göstermektedir. Evlilik nitel bir vasıfken, Yaş ve Araba Sayısı nicel vasıflardır. Eğer nicel kurallar problemi Boolean kuralı problemine kodlanabilirse, Boolean birliktelik kurallarını (veya düzenli) bulmak için kullanılan herhangi bir algoritma, nicel birliktelik kurallarını bulmak için de kullanılabilir.

Tablo 3 Boolean Birliktelik Kuralları Problemin Kodlanması

Tanım	**Yaş: 20–24**	**Yaş: 25–29**	**Evlilik: Evet**	**Evlilik: Hayır**	**A r a b a Sayısı: 0**	**A r a b a Sayısı: 1**	**A r a b a Sayısı: 2**
1000	1	0	0	1	1	0	0
2000	1	0	1	0	0	1	0
3000	1	0	0	1	0	1	0
4000	0	1	1	0	0	0	1
5000	0	1	1	0	0	0	1

Kaynak: M.H. Dunham, Y. Xiao, L. Gruenwald, Z.Hossain, "A Survey Of Association Rules", Southern Methodist University, University of Oklahoma, 2001, s.45

Tablo 3, Tablo 2'de verilen örneğin Boolean kuralları problemine kodlanmasını gösterir. Bu uygulama yaklaşımı beraberinde iki problem getirir. İlk olarak, bir vasıf için aralıkların sayısı çok fazlaysa, herhangi bir aralığın desteği düşük olabilir. Buna "Minsup" (Minimum destek) problemi denir. İkinci olarak, bir vasıf için aralıkların sayısı azsa, bilgi kaybetme olasılığı mevcuttur. Bu da bazı kuralların eşik güveninin olmayabileceğine işarettir. Buna da "Minconf" (Minimum güven) problemi denir. Bu problemlerin üstesinden gelmek için, aralık üzerindeki tüm olası dağılımlar, her aralığın işlemi sırasında birleştirilebilir. Bu da, eşik destek probleminden kaçınmak için bitişik aralıkların birleştirilmesi ve eşik güven probleminden kaçınmak için aralık sayısının arttırılması demektir. Ancak, bu yaklaşım da iki yeni problemi (yüksek işlem zamanı ve çok fazla kural) beraberinde getirir. Yüksek işlem zamanı bir vasfın aralık sayısı arttırıldığında ortaya çıkar. Diğer taraftan, eşik desteğini sağlayan aralıkları içeren herhangi bir dağılımı ele alırsak, sayıca fazlalaşan kurallar ortaya çıkar. "Minsup" probleminden kaçınmak için, nicel vasıfların bitişik değer/aralıkları üzerinden dağılımlar düşünülmüştür. Kullanıcı tarafından belirlenmiş bir "Maksimum destek" parametresi tanımlayarak, bitişik aralıklarının arttırılması kısıtlanmıştır. Bitişik aralıklar, birleşmiş destek maksimum destekten daha az olana kadar birleştirilir. Ancak, desteği maksimum

desteği aşan tek bir aralık hala düşünülebilir. Nicel vasıflı veritabanından sık aralıklı ürün kümeleri bulma problemi üç aşamada çözülmüştür. İlk olarak, her vasfın bölümlendirilip bölümlendirilmeyeceğine karar verilir. Eğer bit vasıf bölümlendirilecekse, bölüm sayısı belirlenmelidir. Daha sonra vasfın değerleri bir ardışık tamsayı kümesine kodlanır. Ardından tüm vasıfların her bir değerinin desteği bulunur. Daha önce de bahsedildiği gibi, "Minsup" probleminden kaçınmak için bitişik değerler, destekleri kullanıcı tarafından belirlenmiş maksimum destekten küçük oluncaya kadar birleştirilir. Minimum desteğe sahip tüm dağılım ve değerler sık aralıklı ürün kümesini oluşturulur. Minimum desteğin %40 olduğu farz edilerek karşılığında destekleri olan bazı büyük ürün kümeleri Tablo 4'de verilmiştir.

Tablo 4 Tablo 3'de Verilen Ürünlerin Bazı Büyük Ürün Kümeleri

Ürün Kümesi	Destek
{<Yaş:20-24>}	%60
{<Yaş:25-29>}	%40
{<Evlilik:Evet>}	%60
{<Evlilik:Hayır>}	%40
{<Araba Sayısı:1>}	%40
{<Araba Sayısı:2>}	%40
{<Yaş:20-24> ve <Evlilik:Hayır>}	%40
{<Yaş:25-29> ve <Evlilik:Evet>}	%40

Kaynak: M.H. Dunham, Y. Xiao, L. Gruenwald, Z.Hossain, "A Survey Of Association Rules", Southern Methodist University, University of Oklahoma, 2001, s.46

Nicel birliktelik kurallarının özel bir çeşidi de *profil birliktelik kuralları*' dır. Burada birliktelik kuralları ifadesinin sağ tarafı müşteri tavrı hakkında bilgi içerirken, sol tarafında müşteri profil verisi verilmiştir. Aşağıda profil birliktelik kuralına bir örnek verilmiştir.

"Gelir > 200,000 YTL $\Rightarrow$ Ev alış fiyatı > 400,000 YTL"

Burada belirtilmek istenen, eğer bir birey 200,000 YTL'nin üzerinde bir gelire sahipse, o zaman 400,000 YTL'nin üzerindeki değerde bir ev satın alacağıdır. Müşteri profilini satın alma bilgisiyle birleştirerek bir şirket, birliktelik kuralının sol tarafında gösterilen profili baz alarak pazarlama kampanyaları hedefleyebilir. Profilin hiyerarşik düzeni, profil kurallarıyla, ürün kümelerinin oluşum giderlerinin azaltılmasında kullanılabilir (gelirin 200,000 YTL ila 300,000 YTL arasında olması gibi).

Başka bir tip nicel kuralsa, *ratio kuralı*dır. Bu kural bir oranı sağlayan vasıflar arasındaki ilişkiyi tanımlamak için kullanılır. Yukarıdaki *profil birliktelik kuralı* bu oran kısıtlamasını sağlamaktadır. Oran kuralları sadece iki değil, çoklu vasıflar arasında da kullanılabilir.

"Gelir: Ev alış fiyatı = 1:2"

Görüldüğü gibi oran 1'e 2'dir. Geliri bir birim olan bir kişi fiyatı iki birim olan bir evi satın alma eğilimindedir.

2.3.7. Aralık Veri Birliktelik Kuralları

Birliktelik kuralı probleminin formülleştirilmesinde, taramanın karmaşıklığının sadece vasıfların sayısına değil, aynı zamanda bir vasfın değerlerinin sayılarına da bağlı olduğu gözlemlenmiştir. Büyük bilgi alanlarına sahip tablolarla ilişkili birliktelik kurallarının madencilik yoluyla çözümlenmesindeki karmaşıklık, verileri bir yerde gruplayarak ve kolektif düşünerek azaltılmıştır. Bir vasıf doğrusal olarak sıralanmışsa, o zaman değerler gruplandırılabilir. Örneğin, bir yaş vasfının tüm değerlerini hesap etmektense, yaş vasfı sınıflandırılabilir (ör: 5 yıllık artışlar halinde). Veri değerlerinin arasındaki ayrımların anlamlı olduğu aralık verisine uygulandığında, az önce önerilen sonuçlar pek de başarı sağlamaz. Tablo 5'de verilen örneği ve 20K'lık bir aralığı ele alalım. Görüyoruz ki herhangi bir değer taşımayan gereksiz bir sürü aralık mevcut. Bu gereksiz aralıklardan kurtulmak için

daha geniş aralıklarla bölümler oluşturulmalıdır. Bu şekilde daha belirgin birliktelik kuralları elde edilebilir.

Tablo 5 Aralık Veri Seti

Maaş	Aralık 18K-38K	Aralık 38K-58K	Aralık 58K-78K	Aralık 78K-98K
18K	1	0	0	0
30K	1	0	0	0
31K	1	0	0	0
80K	0	0	0	1
81K	0	0	0	1
82K	0	0	0	1

Kaynak: M.H. Dunham, Y. Xiao, L. Gruenwald, Z.Hossain, "A Survey Of Association Rules", Southern Methodist University, University of Oklahoma, 2001, s.49

2.3.8. Çoklu Min-destek Birliktelik kuralları

İncelenen önceki çalışmalar, tek destekle büyük veritabanlarında birliktelik kurallarının veri madenciliği yoluyla çözümlenmesi üzerine odaklanmıştır. Bazen tek bir min-destek eşiği kullanımı etkin olmayabilir.

Tüm veritabanı için bir eşik desteği kullanıldığından beri, verideki tüm ürünlerin aynı doğadan ve/ya da aynı sıklıkta olduğu farz edilmiştir. Aslında, bazı ürünler nadiren görünürken bazı ürünler gayet sık olabilir. Ancak ikinci olan, ilk olandan daha kapsamlı bilgiye sahip ve daha ilginç olabilir. Örneğin, %8 destekle bir

ekmek ⇒ peynir kuralı bulmanın yanı sıra, %3 destekle bir kepek ekmek ⇒ tulum peyniri kuralı daha çok bilgi içerebilir. Diğer basit bir örnek de bir süpermarkette nadir sıklıkta satılan ancak daha çok kazanç getiren ocak ve tava olabilir. Ocak ⇒ tava gibi kullanışlı bir kural bulmak çok ilginç olabilir.

Eşik desteği çok yüksekse, nadir ürünleri ilgilendiren kurallar bulunamayacaktır. Hem sık hem de nadir ürünleri ilgilendiren kuralları bulmak için, eşik desteği çok düşük olmalıdır. Maalesef bu durum çok fazla kural üreterek kombine bir patlama yaratır çünkü sık ürünler olası her yoldan kural oluşturmaya çalışacaktır ve bu kuralların çoğu anlamsız olacaktır. Bu ikileme "nadir ürün problemi" denir. Bu problemin üstesinden gelmek için şöyle bir strateji izlenebilir: (a) verileri ürünlerin güvenlerine göre birkaç bloğa ayırıp ve her blokta farklı eşik desteğine sahip birliktelik kuralları bulmak, (b) birbiriyle ilişkili nadir ürünleri soyut bir ürün olarak gruplandırmak ve böylece bu soyut ürünün daha sık olmasını sağlamak. Daha sonra sayısal aralık verisindeki birliktelik kurallarını bulmak için algoritmayı uygulamak gerekmektedir.

2.3.9. Mültimedya Birliktelik Kuralları

Mültimedya veritabanları, veritabanı toplulukları arasında en çok umut veren çalışmalardan biri haline gelse de, mültimedya veritabanlarında birliktelik kurallarını bulmak o kadar da dikkat çekici olmamıştır. Birçok ilişkili ve nesne yönlendirmeli veritabanı, geçmiş yıllara göre günümüzde mültimedya nesnelerini daha sık kullanır hale gelmiştir. Bu mültimedya nesneleri fotoğraf, video, müzik, vs. gibi şeyleri içermektedir. Global internetin aşırı kullanımı, mültimedya nesnelerin talebinde artışa neden olmuştur. Bu verileri inceleyip aralarındaki ilişkiyi bulmak gerekli bir hale gelmiştir. Buna örnek bir birliktelik kuralı şu şekildedir:

"Okyanus ile alakalı bir resim, boyutu büyük ⇒ Rengi mavi"

Bu alanda daha fazla araştırma yapılması gerekmektedir.

2.3.10. Maksimal Birliktelik Kuralları

Maksimal birliktelik kuralları, düzenli birliktelik kurallarının kolaylaştırılmasından çok, vasıfların kümeleri arasındaki ilişkinin daha güçlü ifade edilmesine izin vermektedir. Örneğin, Tereyağı ⇒ Gazete şeklinde bir kural, belli bir güven ve desteğe sahip bir işlemde tereyağı olduğu zaman gazete de olacağını ifade eder. Bu ilişki, sadece T_i={Tereyağı, Gazete} 'den oluşan bir işlemde olduğu kadar, T_j={Gazete, Tereyağı, Yumurta}'dan oluşan bir işlem için de doğru olabilir. X ve Y olarak verilen iki ürün ($X \subseteq Y$, X:Y) T_j hareketi ile ilgili olarak ($T_j \cap Y=X$) *maksimal*'dir denir. X ve Y birer maksimal küme iseler X ⇒ Y birliktelik kuralı bir *maksimal birliktelik kuralı*dır. Farz edelim ki bir dükkânda gıda ürünleri ve karışık ürünler olarak iki kategoride ürün satılsın. Tereyağı ve Yumurta gıda ürünleri kategorisindeyken, Gazete karışık ürünler kategorisindedir. Burada Tereyağı ve Gazetenin ikisi de T_i'de maksimal iken, Gazete T_j'de maksimaldir. Böylece Tereyağı ⇒ Gazete T_i'de sağlar ancak T_j'de sağlamaz. Burada destek, düzenli birliktelik kuralında %100 (her iki işlem için sağladığı taktirde), maksimal birliktelik kuralında %50 olmaktadır.

2.3. BİRLİKTELİK KURALLARI İÇİN KULLANILAN ALGORİTMALAR

Büyük veri yığınlarından bilgi sağlamaya yarayan birçok algoritma ardışık ve paralel olarak sınıflandırılabilir. Ardışık algoritmalar ürün kümelerinin oluşturulduğu ve sayıldığı mantıksal ifadeler içerir. Diğer taraftan, paralel algoritmalar ise büyük ürün kümelerinin paralellik sağlayarak oluşturulmasını sağlar.

2.3.1. Ardışık Algoritmalar

2.3.1.1. AIS Algoritması

AIS (Agrawal, Imielinski, Swami) Algoritması, bir işlem veritabanında tüm büyük ürün kümelerini oluşturmak amacıyla geliştirilmiş yayımlanan ilk algoritmadır. 1993 yılında geliştirilmiştir. Veritabanındaki isimlerin, yani, ürün isimlerinin A'dan Z'ye sıralanması kısıdını taşır. Bu algoritma nitel kuralların bulunmasını hedeflemektedir. Bu yöntem, sonuçta tek bir ürünle sınırlandırılmıştır. Yani, birliktelik kuralları $X \Rightarrow I_k \mid \alpha$ şeklindedir. Burada X, ürünlerin kümesi; I_k, I'nın içinde yer alan tek bir üründür ve α ise kuralın güvenidir.

AIS Algoritması tüm veritabanı üzerinden çoklu döngü yapar. İlk döngüde, ayrı ürünlerin desteğinin sayımını yapar ve veritabanında hangilerinin büyük ürün kümesi ve hangilerinin sık tekrarlanan ürün kümesi olduğunu belirler. Her döngüde bulunan büyük ürün kümeleri aday ürün kümeleri oluşturmak amacıyla arttırılır. Bir işlemin taranmasının ardından, bir önceki döngüde bulunan büyük ürün kümeleri ile bu işlemde kullanılan ürünler arasındaki ortak ürün kümeleri belirlenir. Bu ortak ürün kümeleri, yeni aday ürün kümeleri oluşturmak amacıyla işlemde kullanılan diğer ürünlerle arttırılır. Büyük bir ürün kümesi olan *l*, sadece büyük ürün kümelerindeki işlemde kullanılan bu ürünlerle arttırılır ve sözlük sırasına göre *l*'nin içindeki ürünlerin herhangi birinden sonra gelir. Bu işin randımanlı uygulanması için, tahmin ve budama tekniği kullanılır. Tahmin ve budama yöntemleri, aday ürünlerden gereksiz olan ürün kümelerinin çıkartılmasıyla aday ürün kümelerinin belirlenmesini sağlar. Daha sonra, her aday kümenin desteği hesaplanır. Minimum güvene eşit veya daha büyük güvene sahip aday kümeleri büyük ürün kümeleri olarak seçilir. Bu büyük ürün kümeleri bir sonraki döngü için aday kümelerinin oluşturulması amacıyla arttırılır. Daha fazla ürün kümesi kalmayıncaya bu işlem sona erer.

Bir ürün kümesinin tekrarından sakınmak için ürünler sözlük sırasına göre sıralanır. A ürün kümesi, sadece A ürün kümesinin sözlük sırasına göre herhangi bir elemanından daha sonra gelen B (ör: $B = I_1, I_2, ..., I_k$) ürünlerince arttırılmaya

çalışılabilir. Örneğin, I = {p, q, r, s, t, u, v} ve {p, q} bir büyük ürün kümesi olsun. İşlem için T = {p, q, r, s}, aşağıdaki aday ürün kümeleri oluşturulur:

{p, q, r} beklenen büyük ürün kümesi: artmaya devam eder

{p, q, s} beklenen büyük ürün kümesi: daha fazla arttırılamaz

{p, q, r, s} beklenen küçük ürün kümesi: daha fazla arttırılamaz

Büyük bir perakende şirketinden elde edilen satış verilerine uygulanmasıyla, AIS Algoritması'nın etkinliği ölçülmüştür. Veritabanında toplamda 46,873 müşteri işlemi ve 63 departman bulunmaktadır. Algoritma, müşterilerin satın alma davranışında, departmanlar arasında bir ilişki olup olmadığının bulunması maksadıyla kullanılmıştır. AIS Algoritması'ndaki ana problem, sonradan küçük aday kümeleri olarak ortaya çıkan çok sayıda aday kümesinin oluşmasıdır. Kuraldaki bu tek sonucun yanı sıra, AIS Algoritması'ndaki dezavantaj, veri yapılarının büyük ürün kümeleri olarak korunması gerektiği ve aday ürün kümelerinin belirtilmemiş olmasıdır. Bir veritabanının m üründen oluşması ve tüm ürünlerin işlemde kullanılması gibi bir durum söz konusu ise, 2^m kadar potansiyel büyük ürün kümeleri olacaktır. Bu sebeple, bu metot en kötü durumda m cinsinden üstel olan bir karmaşıklığı sergilemektedir.

Birliktelik kuralları için geliştirilmiş bu ilk algoritmanın dezavantajlarını giderebilmek için geliştirilmeye devam edilmiştir.

2.3.1.2. SETM Algoritması

SETM Algoritması'nın bulunması büyük ürün kümelerini hesaplamak için SQL kullanılmasının istenmesiyle yola çıkılmıştır. Bu algoritmada, L_k geniş nesne kümesinin her bir elemanı iki parametreden oluşur; bunların bir tanesi nesnenin ismiyken diğeri bu nesneyi ayırt etmeye yarayan bir özellik numarasıdır. Bu algoritmada sabit büyük ürün kümelerinin her bir elemanı ($\overline{L}_k$), <TID, ürün kümesi> şeklindedir. Burada TID bir işlemin tek tanımlayıcısı(ürünü ayırt etmeye

yarayan bir özellik numarası) anlamındadır. Aynı şekilde, aday ürün kümelerinin her bir elemanı ($\overline{C}_k$), <TID, ürün kümesi> şeklindedir. AIS Algoritması gibi, SETM Algoritması da veritabanı üzerinden çoklu döngüler yapar. İlk döngüde, ayrı ürünlerin desteğinin sayımını yapar ve veritabanında hangilerinin büyük ürün kümesi ve hangilerinin sık tekrarlanan ürün kümesi olduğunu belirler. Daha sonra, bir önceki döngüde bulunan büyük ürün kümelerinin arttırılmasıyla aday ürün kümelerini oluşturur. Ek olarak, SETM, aday ürün kümeleriyle yapılan işlemlerin oluşturulmasının TID'lerini hatırlar. Birleştirme-Katılma işlemindeki ilişki aday ürün kümeleri oluşturmakta kullanılabilir. Aday ürün kümeleri oluştururken, SETM Algoritması, ardışık bir ifadede oluşturulan işlemin TID'si ile birlikte aday ürün kümelerinin bir kopyasını kayıt eder. Daha sonra, aday ürün kümeleri ürün kümelerine göre sınıflandırılır ve küçük ürün kümeleri tek bir fonksiyon kullanılarak silinir. Eğer veritabanı TID'ye göre sınıflandırılarak sıralanmışsa, bir sonraki döngüdeki işlemin kapsadığı büyük ürün kümeleri TID'ye göre $\overline{L}_k$ 'ların sınıflandırılmasıyla elde edilir. Bu yolla, veritabanı üzerinden birçok döngü yapılır. Daha fazla büyük ürün kümesi bulunamadığı zaman algoritma sona erdirilir.

Bu algoritmanın ana dezavantajı, $\overline{C}_k$ aday kümelerinin sayısına bağlıdır. Her bir aday ürün kümesinin bir TID ile ilişkili olmasıyla birlikte, fazla sayıdaki TID'leri kayıt etmek için daha fazla boş yere ihtiyaç duyulmuştur. Ayrıca, aday ürün kümesinin desteği hesaplanırken $\overline{C}_k$ sıralanmış halde değildir, bunun için ürün kümelerinin bir kez daha sıraya dizilmesi gerekecektir. Daha sonra, destek sınırlamasını sağlamayan aday ürün kümeleri çıkarılarak aday ürün kümeleri budanır. Sonuç kümesi ($\overline{L}_k$) için TID'ye göre başka bir sınıflandırma yapılması gerekir. Daha sonra, bir sonraki döngüde aday ürün kümeleri oluşturmak için $\overline{L}_k$ kullanılabilir. SETM Algoritması'nda hafıza yönetim tekniği düşünülmemiştir. $\overline{C}_k$'nın ana belleğe sığacağı farz edilmiştir. Ayrıca, SETM'nin verimli olmadığı ve

bir DBMS (Database Management System) ile karşılaştırılarak çalıştırıldığında rapor edilen hiç bir sonuç çıkmadığı belirtilmiştir.

2.3.1.3. Apriori Algoritması

Apriori ismi, bilgileri bir önceki adımdan aldığı için "prior" anlamında Apriori'dir. Apriori Algoritması, birliktelik kurallarının veri madenciliği yoluyla çözümlenmesinin tarihindeki büyük bir başarıdır. Şu ana kadar bilinen en iyi birliktelik kuralı algoritmasıdır. Bu yöntem, büyük bir ürün kümesinin herhangi bir alt kümesinin de yine büyük bir ürün kümesi olması gerektiği özelliğini kullanır. Aynı zamanda, bir ürün kümesindeki ürünlerin sözlük sırasına göre sıralandığı bir yöntemdir. Bu algoritmanın, AIS ve SETM algoritmalarından temel farklılıkları aday ürün kümelerini oluşturma yolu ve aday ürün kümelerinin sayım için seçilme şeklidir. Daha önce de belirtildiği gibi, AIS ve SETM algoritmalarında, bir önceki döngüde elde edilen büyük ürün kümeleri ile işlemde kullanılan ürünler arasındaki ortak ürün kümeleri elde edilir. Bu ortak ürün kümeleri, aday ürün kümelerini oluşturmak için işlemde kullanılan başka ayrı ürünlerle arttırılır. Ancak, bu ayrı ürünler büyük ürün kümeleri olmayabilir. Bildiğimiz gibi bir büyük ürün kümesinin üst kümesi ve bir küçük ürün kümesi, küçük bir ürün kümesi olarak sonuçlanacaktır. Bu yöntemler, sonradan küçük ürün kümeleri olarak sonuçlanacak çok sayıda aday ürün kümesi oluşturacaktır. Apriori Algoritması bu önemli noktaya hitap etmektedir. Apriori, aday ürün kümelerini, bir önceki döngüde elde edilen büyük ürün kümelerini birleştirerek ve veritabanındaki işlemlere bakmaksızın bir önceki döngüde küçük ürün kümeleri olarak elde edilen alt kümelerin silinmesiyle oluşturur. Sadece bir önceki döngüde elde edilen büyük ürün kümelerini ele alarak, aday ürün kümelerinin sayısı kayda değer bir azalma gösterir. İlk döngüde, tek ürünlü ürün kümelerinin sayımı yapılır. İlk döngüde bulunan büyük ürün kümeleri ile ikinci döngünün aday ürün kümelerini oluşturmak apriori_gen fonksiyonu kullanılır. Aday ürün kümeleri bulunduğunda, veritabanı taranarak ikinci en büyük ürün kümelerini bulmak için aday ürün kümelerinin desteklerinin sayımı yapılır.

Üçüncü döngüde, ikinci döngüde bulunan büyük ürün kümeleri, bu döngünün büyük ürün kümelerini bulmak için aday ürün kümeleri olarak kullanılır. Bu iteratif işlem, daha fazla büyük ürün kümesi bulunamayıncaya kadar sürer ve sonra sona erer. Algoritmanın her *i* döngüsü veritabanını bir kez tarar ve *i* büyüklüğündeki büyük ürün kümelerini belirler. L_i, *i* büyüklüğündeki büyük ürün kümelerini, C_i ise *i* büyüklüğündeki aday ürün kümelerini ifade eder.

Apriori algoritmasına ilişkin bazı varsayımlar şu şekildedir: Bu algoritmada kullanılan market sepeti verisinde her harekette yer alan öğe kodları sayısaldır ve öğe kodları küçükten büyüğe doğru sıralıdır. Öğe kümeleri eleman sayıları ile birlikte anılır ve k adet öğeye sahip bir öğe kümesi ile gösterilir.

Agrawal ve Srikant tarafından geliştirilen Apriori algoritması 1994 yılında 20. VLDB (Very Large Database Endowment) konferansında sunulmuştur. Bu bildiride, Agrawal ve Srikant algoritmanın çalışma ayrıntılarını ve algoritmanın kaba kodunu şu şekilde sunmuştur:

- Verilerin ilk taranması esnasında, geniş ürün kümelerinin tespiti için, tüm ürünler sayılır.
- Bir sonraki tarama, *k* ıncı tarama olsun, iki aşamadan oluşur;
- Apriori-gen fonksiyonu kullanılarak, (k-1)inci taramada elde edilen, L_{k-1} ürün kümeleriyle, C_k aday ürün kümeleri oluşturulur,
- Sonra veritabanı taranarak, C_k daki adayların desteği sayılır.
- Hızlı bir sayım için, verilen bir *L* işlemindeki, C_k yı oluşturan adayların çok iyi belirlenmesi gerekir.

Şekil 6'da, Apriori algoritması bir örnek ile açıklanmaktadır.

Tablo 6 Marketten Yapılan Alışveriş Bilgilerini İçeren D Veritabanı

Fiş No	Ürün
A100	I1,I2,I5
A200	I2,I4
A300	I2,I3
A400	I1,I2,I4
A500	I1,I3
A600	I2,I3
A700	I1,I3
A800	I1,I2,I3,I5
A900	I1,I2,I3

Tablo 6'da bir marketten yapılan alışveriş bilgilerini içeren D veritabanı görülmektedir. Bu veri tabanında yapılan alışverişlerin numaraları Fiş No sütununda görülmektedir. Her alışverişte satın alınan ürünler de Ürün sütununda görülmektedir. Han ve Kamber apriori algoritmasında takip edilen basamakları şu şekilde anlatmaktadır.

Her ürünün sayısı için D'yi tara

C1

Ürün	Destek Sayısı
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Ürün sayılarını minimum destek sayısı ile karşılaştır

L1

Ürün	Destek Sayısı
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L1'den C2 ürünlerini oluştur

C2

Ürün
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I4}
{I3,I5}
{I4,I5}

Her ürünün sayısı için D'yi tara

C2

Ürün	Destek Sayısı
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

Ürün sayılarını minimum destek sayısı ile karşılaştır

L2

Ürün	Destek Sayısı
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

L2'den C3 ürünlerini oluştur

C3

Ürün
{I1,I2,I3}
{I1,I2,I5}

Her ürünün sayısı için D'yi tara

C3

Ürün	Destek Sayısı
{I1,I2,I3}	2
{I1,I2,I5}	2

Ürün sayılarını minimum destek sayısı ile karşılaştır

L3

Ürün	Destek Sayısı
{I1,I2,I3}	2
{I1,I2,I5}	2

Şekil 6 Apriori Algoritmasının Gösterimi

Kaynak: Jiawei Han; Micheline Kamber, "*Data Mining: Consepts and Tecniques*", Morgan Kaufmann Publishers, Inc., San Francisco, California, 2001, s.233

1) Algoritmanın ilk adımında, her ürün tek başına bulunduğu C_1 kümesinin elemanıdır. Algoritma, her ürünün sayısını bulmak için tüm alışverişleri tarar ve elde edilen sonuçlar Şekil 6'da Destek sayısı sütununda görülmektedir.

2) Minimum alışveriş destek sayısının 2 olduğu varsayılırsa, tek başlarına sık tekrarlanan ürünler L_1 kümesinde görülmektedir. C_1 kümesindeki tüm ürünlerin destek sayısı, minimum destek eşik değeri olan 2'den dazla olduğu için C_1 tüm ürünler sık tekrarlanan ürün olarak değerlendirilir ve L_1 kümesinde aktarılır.

3) Hangi ürünlerin ikili olarak sık tekrarlandığını belirlemek için L_1 kümesindeki ürünlerin ikili kombinasyonları bulunarak C_2 kümesi oluşturulur.

4) C_2 kümesindeki ürünlerin destek sayılarını bulmak amacıyla D taranır ve bulunan değerler destek sayısı sütununda belirtilir.

5) C_2 kümesindeki ürünlerin minimum destek eşik değerini aşan ürünler L_2 kümesine aktarılır.

6) Hangi ürünlerin üçlü olarak sık tekrarlandığını belirlemek için L_2 kümesindeki ürünlerin üçlü kombinasyonları bulunarak C_3 kümesi oluşturulur. Bu durumda C_3 = {{I1,I2,I3}, {I1,I2,I5}, {I2,I3,I5}, {I2,I3,I4}, {I2,I3,I5}} olması beklenir. Ancak Apriori algoritmasına göre, sık tekrarlanan öğelerin alt kümeleri de sık tekrarlanan öğe olması

gerekmektedir. Buna göre yukarıdaki C_3 kümesindeki elemanlar sık tekrarlanan olmadığı için, yeni C_3 kümesi C_3 = {{I1,I2,I3}, {I1,I2,I5}} olur.

7) C_3 kümesindeki ürünlerin destek sayılarını bulmak amacıyla D taranır ve bulunan değerler destek sayısı sütununda belirtilir.

8) C_3 kümesindeki ürünlerden minimum destek eşik değerini aşan ürünler L_3 kümesine aktarılır.

9) Hangi ürünlerin dörtlü olarak sık tekrarlandığını belirtmek için L_3 kümesindeki ürünlerin dörtlü tek kombinasyonu {I1,I2,I3,I5} olarak belirlenir. Ancak bu kümenin tamamı sık tekrarlanan öğe olmadığı için C_4 kümesi boş küme olur ve Apriori tüm sık tekrarlanan öğeleri bularak sonlanmış olur.

Apriori her zaman AIS ve SETM algoritmalarını geride bırakmaktadır. Tablo 2'de verilen örneği tekrar inceleyelim. Apriori dördüncü aşamada sadece bir aday küme ({Elma, Simit, Tavuk, Diyet Kola}) oluşturmaktadır. Diğer taraftan AIS ve SETM şu şekilde beş adet aday küme oluşturacaktır: {Elma, Simit, Tavuk, Diyet Kola}, {Elma, Simit, Tavuk, Yumurta}, {Elma, Simit, Diyet Kola, Yumurta}, {Elma, Tavuk, Diyet Kola, Yumurta} ve {Simit, Tavuk, Diyet Kola, Yumurta}. Bildiğimiz gibi, büyük ürün kümeleri, aday kümelerin desteklerinin sayımı yapıldıktan sonra bulunmaktadır. Bu yüzden, eğer AIS ya da SETM yollarından biri izlenirse gereksiz yere destek sayımı yapılmış olacaktır.

Apriori, k döngüsünün aday oluşturma fazında depolanması gereken bütün L_{k-1} büyük ürün kümeleri ve C_k aday ürün kümelerinin belleğe sığmayabileceği gerçeğinin üstesinden gelmek amacıyla hafıza yönetimi içerir. Buna benzer bir problem C_k'nın depolandığı yer olan sayım zarfında ortaya çıkabilmektedir ve veritabanı işlemlerine hafıza oluşturmak için en az bir sayfaya ihtiyaç duyulmaktadır.

Bu tür durumların üstesinden gelmek için iki tane yaklaşım düşünülmüştür. İlk olarak L_{k-1}'in belleğe sığıp, C_k'nın sığmayacağı bir durumda, programcılar bu problemi apriori_gen fonksiyonunu yeniden modifiye ederek çözmektedir. Bu sayede belli sayıda C_k' aday kümeleri oluşturularak belleğe sığması sağlanır. C_k' den çıkarılan L_k büyük ürün kümeleri diske yazılır ve küçük ürün kümeleri silinir. Bu işlem tüm C_k ölçümleri yapılana kadar devam eder. İkinci olarak ise L_{k-1}'in belleğe sığmayacağı bir durum söz konusu olabilmektedir. Bu problem de L_{k-1}'in harici olarak sınıflandırılmasıyla çözümlenir. Bir parça L_{k-1} içinde ilk (k-2) ürünlerin aynı olduğu belleğe getirilir. L_{k-1} parçası okunur ve bellek dolana kadar aday kümeler oluşturulur. Bu işlem tüm C_k'nın sayımı yapılana kadar devam eder. Apriori'nin performansı büyük ürün kümelerini bulmak için yapılan birçok deneyde değerlendirmiştir. Yapılan deneylerin sonuçlarında Apriori Algoritması'nın her zaman AIS ve SETM Algoritması'nı geride bıraktığı görülmektedir.

2.3.1.4. Apriori-TID Algoritması

Daha önce de bahsedildiği gibi Apriori, desteğin sayımını yapmak için her döngünün her aşamasında tüm veritabanını taramaktadır. Tüm veritabanını taramak, döngünün her aşaması için gerekli olmayabilir. Bu temele dayanarak, Apriori-TID adıyla yeni bir algoritma geliştirilmiştir. Apriori'ye benzer bir şekilde Apriori-TID de, döngüye başlamadan önce aday ürün kümlerini belirlemek için yine Apriori'nin aday kümeleri oluşturma fonksiyonunu kullanır. Apriori'den ana farkı, ilk döngüden sonra desteğin sayımı için veritabanını kullanmıyor olmasıdır. Onun yerine, $\overline{C}_k$ olarak gösterilen bir önceki döngüde kullanılan aday ürün kümelerinin bir kodlamasını kullanır.

İlk başta tüm veritabanı taranarak ürün kümeleri terimlerinden $\overline{C}_1$ elde edilir. Bu, $\overline{C}_1$ için girilen her değerin TID ile beraber tüm ürünlere sahip olduğu anlamına

gelir.L_1 dediğimiz 1-ürünlü büyük ürün kümeleri, C_1'e girilen değerlerin sayımının yapılmasıyla hesaplanır. Daha sonra C_2'yi elde etmek için apriori_gen kullanılır. Bir T işlemine karşılık gelen, $\overline{C}_2$ için girilen değerler, T'de mevcut olan C_2'nin elemanları göz önünde bulundurularak elde edilir. Bu görevi gerçekleştirmek için, tarama işlemi tüm veritabanı yerine $\overline{C}_1$'e yapılır. Daha sonra $\overline{C}_2$'nin desteğinin sayımı yapılarak L_2 elde edilir. Bu işlem aday ürün kümeleri boş küme olarak bulununcaya kadar devam ettirilir. Bu kodlama işlevinin kullanılmasının avantajı, sonraki döngülerde kodlama işlevinin büyüklüğünün veritabanından daha küçük olmasıdır. Böylece veri okuma işlemi daha az çabayla gerçekleşir ve daha kısa sürer. Apriori-TID de AIS ve SETM algoritmalarına baskın bir yöntemdir. L_3'ün {{Elma, Simit, Tavuk}, {Elma, Simit, DiyetKola}, {Elma, Tavuk, DiyetKola}, {Elma, Tavuk, Yumurta}, {Simit, Tavuk, DiyetKola}} olarak bulunduğu Tablo 2'te verilen örneğe bakacak olursak, Apriori gibi, Apriori-TID yönteminin de sadece bir aday ürün kümesi {Elma, Simit, Tavuk, DiyetKola}oluşturduğunu görürüz. Daha önce de bahsedildiği gibi AIS de, SETM de 5 adet aday ürün kümesi oluşturmaktadır. Bunlar {Elma, Simit, Tavuk, DiyetKola}, {Elma, Simit, Tavuk, Yumurta}, {Elma, Simit, DiyetKola, Yumurta}, {Elma, Tavuk, DiyetKola, Yumurta} ve {Simit, Tavuk, DiyetKola, Yumurta}'dır.

Apriori-TID yönteminin, hafızaya sığması ve uzun bir kuyruğa sahip olan büyük ürün kümelerinin dağılımı açısından az sayıda $\overline{C}_k$ kümesi olduğu durumlarda Apriori'ye üstünlük sağladığı da bilinmektedir. Bu, büyük ürün kümelerinde girilen verilerin dağılımının ilk aşamalarda yüksek olduğu anlamına gelmektedir. Dağılım derhal küçülür ve ardından doruk noktasına ulaşarak uzun bir süre devam eder. Büyük veri kümeleri için Apriori'nin performansı, Apriori-TID'in performansından daha iyi olduğu bildirilmiştir. Diğer taraftan, $\overline{C}_k$ kümeleri nispi

olarak küçük oldukları (belleğe sığdıkları) zaman Apriori-TID, Apriori'ye üstünlük sağlar. Bu yüzden, bir hibrit yöntemi olan "Apriori-Hybrid" çıkarılmıştır.

2.3.1.5. Apriori-Hybrid Algoritması

Apriori-Hybrid algoritması Apriori ve Apriori-TID algoritmasının karışımıdır. Bu algoritma, veri üzerinden her döngüde aynı algoritma kullanılmasının şart olmaması fikri üzerine kurulmuştur. Belirtildiği gibi, ilk döngülerde Apriori yöntemi, sonraki döngülerde Apriori-TID yöntemi daha iyi sonuçlar vermiştir. Deneysel gözlemlere dayanarak, Apriori-Hybrid yöntemi ilk döngülerde Apriori kullanıp, daha sonra $\overline{C}_k$ kümesinin döngünün sonunda belleğe sığacağı umularak Apriori-TID yöntemine geçiş yapmasıyla geliştirilmiş bir yöntemdir. Bu sebeple, her döngünün sonunda $\overline{C}_k$'nın tahmini gereklidir. Aynı zamanda, Apriori'den Apriori-TID'e geçmenin bir maliyet ilişkisi vardır. Bu yöntemin performansı büyük veri kümeleri için deneylerin yönetilmesiyle de değerlendirilmiştir. Apriori-Hybrid yönteminin en son döngüdeki geçiş aşaması dışında Apriori yönteminden daha iyi çalıştığı gözlemlenmiştir.

2.3.1.6. OCD Algoritması (Off-line Candidate Determination/Off-Line Ürün Kümesi Tespiti)

OCD yöntemi, geniş ürün kümelerini belirlemek için veritabanından alınmış küçük örneklerin çok iyi sonuçlar verebileceği fikrine dayanan bir algoritmadır. OCD yöntemi, gereksiz aday kümelerini yok etmek için önceki döngülerden elde edilen bilgilerin birleşik analiz sonuçlarını kullanır. Bir $Y \subseteq I$ altkümesinin seyrek olup olmadığını bilmemiz için, işlemlerin en az (1-s) kadarının taranması gerekmektedir. Burada s destek eşiğidir. Bu yüzden s'in düşük değerleri için nerdeyse bütün ilişkili veriler okunmalıdır. Açıkça görülüyor ki eğer veritabanı çok büyükse, veri üzerinden mümkün olduğunca az döngü yapılması önem taşımaktadır.

OCD aday kümeleri belirlemede AIS'ten farklı bir yaklaşım sergiler. OCD döngüyü olabildiğince basit tutar ve aday kümeleri bulmak için önceki döngülerde elde edilen tüm sonuçları kullanır. Bu basit çözüm aksine çok kapsamlı bir işlemdir. Burada k+1 büyüklüğünün tüm alt kümeleri incelenir ve bu gereksiz aday kümelerinin sayısını çok arttırır.

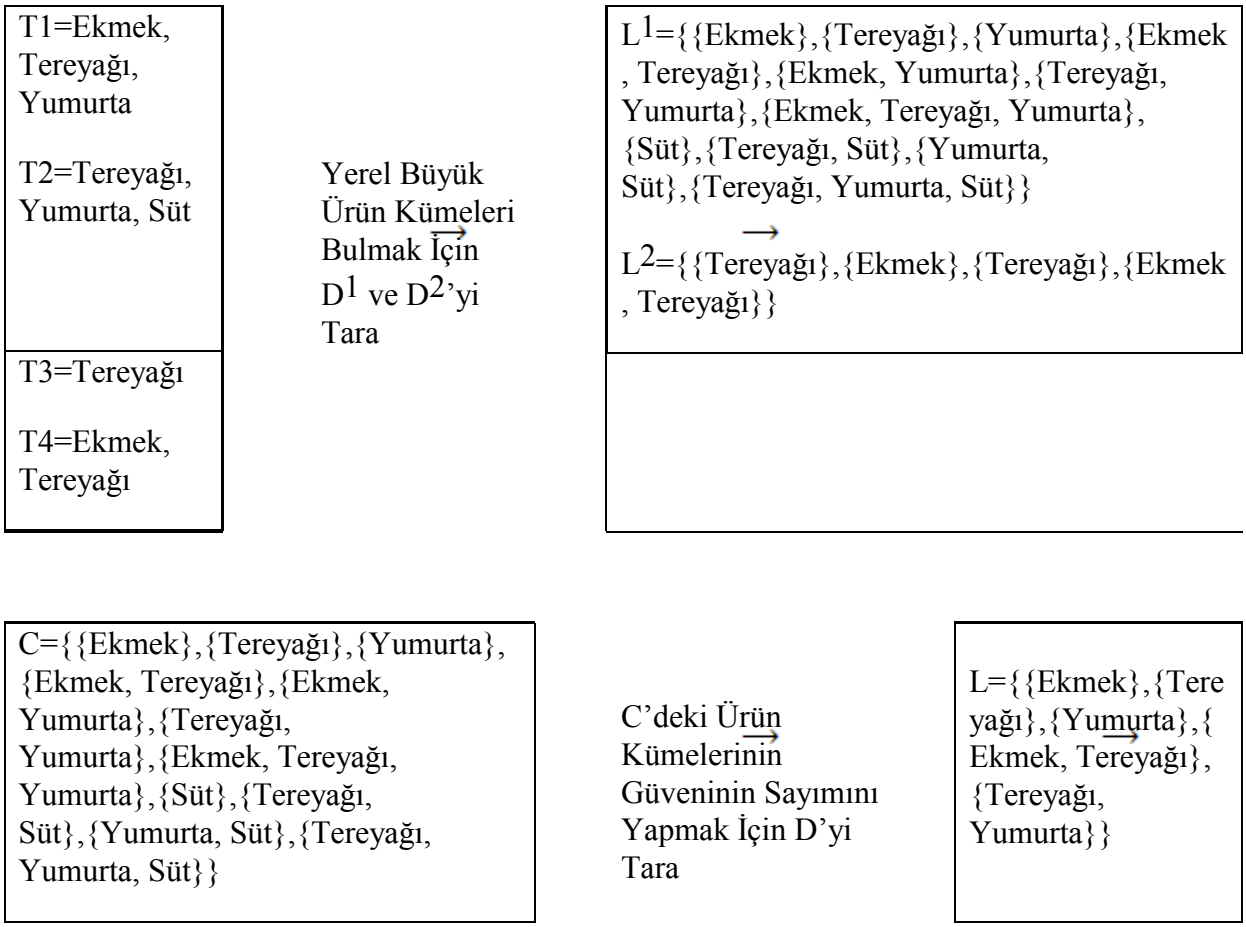
Bu algoritmanın performansı iki veri kümesi kullanılarak değerlendirilmiştir. Bunlardan biri 4734 öğrencinin bir kursa kayıt olmasıdır. İkincisi ise bir telefon şirketinin 30,000 adet kayıtlı ağ sorununu içeren hata yönetim veritabanıdır. Deneysel bulgular, OCD için gerekli zaman yüzdesinin AIS'ninkinin %10–20 kadarı olduğunu belirtir. OCD'nin avantajı düşük destek eşiği ile yükselir. AIS'de oluşturulmuş aday kümeleri, OCD'de oluşturulanlardan kayda değer bir şekilde yüksektir. AIS döngü esnasında aday kümelerini bazen iki kez oluştururken, OCD her aday kümesinden tek bir tane oluşturur ve veritabanıyla karşılaştırmadan önce büyük alt kümelerinin olup olmadığını kontrol eder.

2.3.1.6. Partition Algoritması (Bölümleme Algortiması)

PARTITION veritabanı taramalarının sayısını ikiye düşürür. Veritabanını küçük bölümlere ayırarak, her bir bölümün ana bellekte ele alınmasını sağlar. Bu bölme işlemi ile bellekte işgal edilen yeri azaltıp daha hızlı sonuca ulaşımı sağlamaktadır. Veritabanının bölümlerine D^1, D^2, ..., D^p diyelim. İlk taramada her D^i ($1 \leq i \leq p$) bölümünde yerel büyük ürün kümeleri bulur, ör:$\{X \mid X.count \geq s \times |D^i|\}$. Yerel büyük ürün kümeleri (L^i), Apriori gibi yüksek seviye bir algoritma kullanılarak bulunabilir. İkinci taramada, tüm veritabanı içerisindeki büyük bir ürün kümesinin, veritabanının en az bir bölümünde yerel olarak büyük ürün kümesi olması gerektiği özelliğini kullanır. Daha sonra her bölümde bulunan yerel büyük ürün kümelerinin birleşimi aday ürün kümeleri gibi kullanılır ve bütün büyük ürün kümelerini bulmak için tüm veritabanı içerisinde sayımları yapılır.

Şekil 7, Tablo 2'deki veriler ile PARTITION Algoritmasının nasıl kullanıldığını göstermektedir. Veritabanı ilk iki işlemi içeren ilk bölüm ve kalan iki

işlemi içeren ikinci bölüm olarak iki bölüme ayrılmıştır. Minimum destek %40 olduğundan ve her bölümde sadece iki işlem olmasından dolayı, sadece bir kez oluşan ürün kümesi büyük ürün kümesidir. Daha sonra iki bölümün içindeki yerel büyük ürün kümeleri işlemin tüm alt kümelerini oluşturur. Bunların birleşimi ikinci taramada aday ürün kümelerinin kümesi olur.



Şekil 7 Partition Algoritması Kullanarak Büyük Ürün Kümelerinin Bulunması

M.H. Dunham, Y. Xiao, L. Gruenwald, Z.Hossain, "*A Survey Of Association Rules*", Southern Methodist University, University of Oklahoma, 2001, s.19

PARTITION, homojen veri dağılımını sağlar. Eğer bir ürün kümesinin sayımı her bölüme düzgün bir şekilde dağılırsa, ikinci taramada sayılacak olan birçok ürün kümesi büyük ürün kümesi olacaktır. Ancak, çarpık bir veri dağılımı için, ikinci taramadaki birçok ürün kümesi, işlemin sonunda karşımıza küçük ürün kümesi olarak çıkabilir. Bu da yanlış ürün kümelerinin bulunmasına devam edilmesiyle işlemcinin gereksiz yere zaman kaybetmesine neden olur.

2.3.1.7. Örnekleme Algoritması (Sampling Algoritması)

Sampling, en iyi durumda tarama sayısını bire düşürür ve en kötü durumda da bu sayı iki olur. Ana belleğe sığacak bir örnek öncelikle veritabanından alınır. Örnekteki büyük ürün kümelerinin kümesi, Apriori gibi yüksek seviye bir algoritma metodu kullanılarak bu örnekten bulunur.

2.3.1.8. DIC Algoritması (Dynamic Itemset Counting/Dinamik Ürün Kümesi Sayımı)

DIC Algoritması ürün kümelerinin önceden sayılmasına ve oluşturulmasına çalışır, böylece veritabanı taramalarının sayısını azaltır. Veritabanı, işlemlerin aralıkları gibi görünür ve bu aralıklar ardışık olarak taranır. Veritabanının sonuna ulaşıldığında, veritabanın en başa geri alır ve tam olarak sayılmayan ürün kümelerini sayar. Veritabanı taramalarının gerçek sayısı aralığın büyüklüğüne bağlı olarak değişir. Aralık yeteri kadar küçükse, tüm ürün kümeleri ilk taramada oluşturulacaktır ve ikinci taramada tam olarak sayımı yapılacaktır. Aynı zamanda PARTITION Algoritması'nda olduğu gibi homojen bir dağılım elde etmemizi sağlar.

2.3.1.9. CARMA (Continuous Association Rule Mining/Sürekli Birliktelik Kuralı Madenciliği Algoritması)

CARMA Algoritması, büyük ürün kümelerini “online” olarak hesaplamamızı sağlar. “Online” olmakla CARMA, kullanıcıya mevcut birliktelik kurallarını gösterir; veritabanındaki ilk tarama boyunca herhangi bir işlemdeki parametrelerin, minimum güvenin, minimum desteğin değiştirilmesine izin verir. En çok 2 veritabanı taramasına ihtiyaç duyar. DIC gibi, CARMA da ürün kümelerini oluşturma işlemini ilk taramada ve tüm ürün kümelerinin sayılması işlemini ikinci taramada yapar. Daha sonra, eğer o anda oluşan ürün kümelerinin alt kümeleri minimum desteğe ve okunan veritabanın bölümlerine bağlı kalarak o an itibariyle potansiyel birer büyük ürün kümesi iseler, işlemden yeni ürün kümeleri oluşturur. Bir ürün kümesinin potansiyel olarak büyük bir ürün kümesi olup olmadığının daha iyi tahmin etmek için, ürün oluşturulmadan önce mevcut sayımların ve oluşumların tahmini sayılarının toplamı olan ürün kümelerinin sayımları için bir üst sınır hesaplanır. Oluşumların tahmini sayıları (maksimum eksikler), ürün kümesi ilk oluşturulduğunda hesaplanır.

2.3.2. Paralel Algoritmalar

Günümüzdeki paralel algoritmalar bir seri algoritma olan Apriori üzerine kurulmuştur. Paralel algoritmalar birden fazla hafıza bölümü bulunan algoritmalardır. Paralel algoritmalarda “veri paralelliği” ve “iş paralelliği” olmak üzere iki diziden bahsedebiliriz. Bu iki dizi, aday ürün kümelerinin tüm işlemcilere dağılımının yapılıp yapılmadığı açısından farklılık gösterir. Veri paralelliğinde her düğüm aynı aday ürün kümelerinin kümelerini sayarken, iş paralelliğinde aday ürün kümeleri bölümlere ayrılır ve işlemcilere dağılımı yapılır ve her düğüm farklı aday ürün kümelerinin kümelerini sayar.

2.3.2.1. Veri Paralellik Algoritması

2.3.2.1.1. CD Algoritması (Count Distribution / Sayım Dağılımı)

CD Algoritması'ndan D veritabanı $\{D^1, D^2,\ldots D^p\}$ olarak bölümlere ayrılmış ve n sayıda işlemciye paylaştırılmıştır. Temel olarak üç aşama mevcuttur. 1. aşamada, yerel veritabanı bölüm D^i 'deki C_k aday ürün kümelerinin yerel destek sayımları bulunur. 2. aşamada, her işlemci, tüm aday ürün kümelerinin genel desteğini bulmak için, birbirine tüm aday ürün kümelerinin yerel destek sayımlarını verir. Üçüncü aşamada ise genel olarak büyük ürün kümeleri olan L_k 'lar tanımlanır ve her işlemcide birbirinden bağımsız olarak L_k'ya apriori_gen uygulanarak k+1 kadar aday ürün kümesi oluşturulur. CD Algoritması 1 – 3 aşamalarını daha fazla aday ürün kümesi bulunamayıncaya kadar sürdürür.

2.3.2.1.2. PDM Algoritması (Parallel Data Mining / Paralel Veri Madenciliği)

PDM Algoritması, CD Algoritması'nın direkt karar tekniği içererek değişikliğe uğramış halidir. Karar tekniği ileriki döngülerde bazı aday ürün kümelerini budamak için kullanılır. Bu, özellikle Apriori Algoritması'nın L_1'den C_2 oluşturması esnasında budama yapmadığı ikinci döngü için kullanışlıdır.

DMA (Distributed Mining Algorithm/ Dağılım Madenciliği Algoritması) ve CCPD Algoritması (Common Candidate Partitioned Database / Ortak Aday Ürün Kümelerine Bölünmüş Veritabanı) diğer veri paralellik algoritmalarındandır.

2.3.2.2. İş Paralellik Algoritması

2.3.2.2.1. DD Algoritması (Data Distribution / Veri Dağılımı)

DD Algoritması'nda, aday ürün kümeleri tüm işlemcilere dairesel döngü şeklinde bölünmüş ve paylaştırılmıştır. Üç aşaması vardır. Birinci aşamada, her işlemci, kendilerine dağılımı yapılmış aday ürün kümelerinin yerel sayımlarını elde etmek için yerel veritabanının bölümlerinin tarar. İkinci aşamada, her işlemci kendi

veritabanı bölümlerini diğer işlemcilere dağıtır ve diğer işlemcilerden onların veri tabanı bölümlerini alır. Daha sonra tüm veritabanında global destek sayımları elde etmek için, alınan veri tabanı bölümlerini tarar. Son aşamada, her işlemci aday ürün kümesi bölümündeki büyük ürün kümelerini hesaplar, bunları tüm büyük ürün kümelerini elde etmek için diğer işlemcilerle paylaşır ve sonrasında aday ürün kümeleri oluşturur, bunları bölümlere ayırır ve bu aday ürün kümelerini diğer işlemcilere dağıtır. Bu aşamalar daha fazla aday ürün kümesi bulunamayıncaya kadar devam eder.

2.3.2.2.2. IDD Algoritması (Intelligent Data Distribution / Akıllı Veri Dağılımı)

IDD Algoritması, DD Algoritması'nın bir geliştirilmiş halidir. Aday ürün kümelerinin, aday ürün kümelerindeki ilk ürüne dayanarak işlemciler arasında bölümlere ayırır. Bu sayede aynı ilk ürüne sahip aday ürün kümeleri aynı bölümde bulunurlar. Bu yüzden, her işlemcinin sadece işlemciye atanmış ürünlerden biriyle başlayan alt kümeleri kontrol etmesi gerekir. Bu da DD Algoritması'ndaki gereğinden fazla olan hesaplamaların azalmasını sağlar.

HPA (Hash-based Paralel Mining of Association Rules/ Bağlantı Kurallarının Çırpı Temelli Paralel Madenciliği) ve PAR Paralel Association Rules/Paralel Bağlantı Kuralları) gibi iş paralellik algoritmaları da mevcuttur.

2.3.2.3. Diğer Paralel Algoritmalar

Paralel algoritmalar için bu iki dizide gruplandırılamayan başka algoritmalar mevcuttur. Bu algoritmalar dizilerle aynı yapıda olsalar da farklı özelliklere sahiptirler. Bu paralel algoritmalar Candidate Distribution, SH ve HD Algoritmaları'nı içerir.

2.3.2.3.1. Candidate Distribution Algorithm (Aday Küme Dağılım Algoritması)

Aday Küme Dağılım Algoritmasına, CD ve DD Algoritması'ndaki eşleme ve iletişim giderlerini azaltmak için başvurulmuştur. Bazı *l* döngülerinde, bir işlemcinin diğer işlemcilerden bağımsız olarak aday ürün kümelerinden tek bir küme oluşturabileceği şekilde işlemciler arasında L_{l-1} büyük ürün kümelerini bölüştürür. Bu aday ürün kümelerini bölümlere ayırma işleminden sonra her işlemci yerel veritabanı bölümünü kullanmak kaydıyla kendi aday ürün kümeleri bölümünü sayarak bağımsız çalışmış olur. Aday ürün kümesinin bölümlere ayırma işleminin öncesinden beri, CD ve DD Algoritması kullanılabilirdi. Aday Küme Dağılım Algoritması da bu iki dizinin bir çeşit hibritidir.

2.3.2.3.2. SH Algoritması

SH Algoritması'nda aday ürün kümeleri, bir seri algoritma olan Apriori'den farklı olarak, önceki büyük ürün kümelerinden evvel oluşturulmaz. Bunun yerine aday ürün kümeleri, veritabanının bölümleri taranırken, her işlemci tarafından bağımsız olarak oluşturulur. k iterasyonunda, her işlemci kendi veritabanı bölümündeki işlemlerden k-ürün kümelerini oluşturur ve sayımını yapar. SH Algoritmasının, Apriori'den faklı bir algoritma üzerine kurulduğu görülmektedir ancak aslında Apriori'ye çok yakın bir algoritmadır. Öncelikle, Apriori gibi iteratiftir, yani, sadece bir iterasyonun en sonunda büyüklükleri artmış yeni aday ürün kümeleri oluşturulur. Apriori'den farkı, aday ürün kümelerinin oluşturulduğu esnada ortaya çıkar; öyle ki, Apriori her iterasyonun sonunda önce aday ürün kümelerini oluştururken, SH algoritması bunu işlem esnasında yapmaktadır. İkinci olarak, eğer veritabanı düzgün bir biçimde sınıflandırılırsa, SH Algoritması tarafından oluşturulan aday ürün kümeleri, Apriori tarafından oluşturulanlarla tamamen aynı olacaktır. Sadece veritabanı aşırı derecede çarpıksa aday ürün kümelerinde farklılıklar olacaktır.

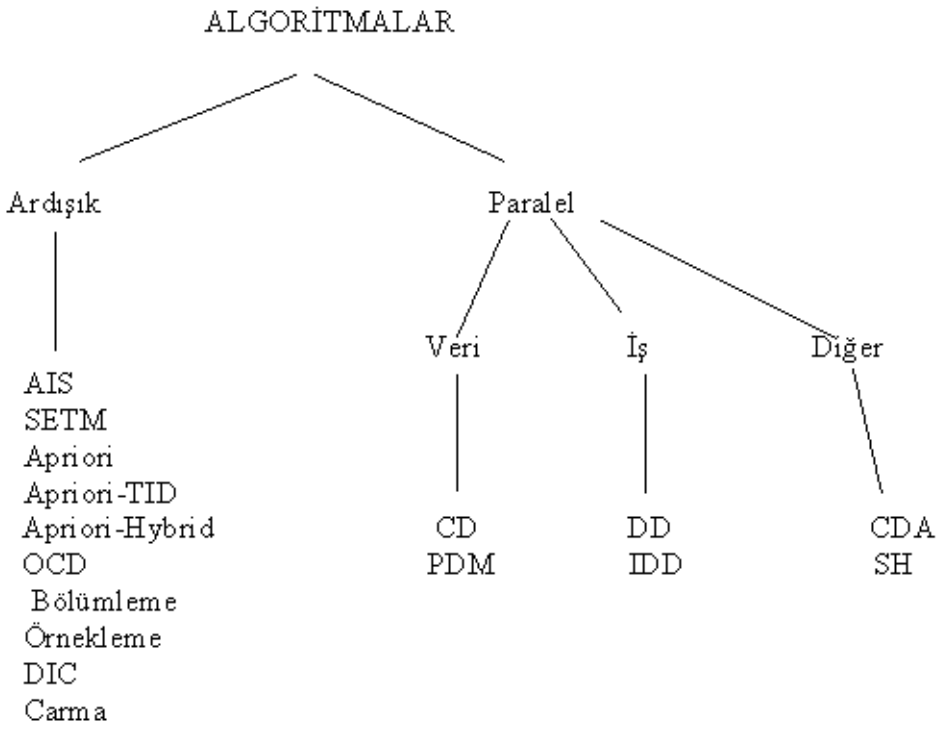
HD Algoritması (Hybrid Distribution / Hibrit Dağılımı)'sı da diğer paralel algoritmalardandır.

2.3.4. Paralel Algoritmaların Geleceği

Birliktelik kurallarının veri madenciliği yoluyla çözümlenmesi için kullanılan tüm paralel algoritmalar bir seri algoritma olan Apriori Algoritması temelleri üzerine kurulmuştur. Apriori Algoritması, özellikle veritabanı taramalarının sayısını azaltması açısından diğer birçok algoritmanın yardımıyla geliştirilmiştir. Geliştirilmiş algoritmaların paralelleştirilmesinin daha iyi çözümler getireceği umulmaktadır.

2.4. ALGORİTMALARIN SINIFLANDIRILMASI

Çok sayıdaki algoritmayı ayırt edebilmek için yaklaşım tarzlarının bir sınıflandırma düzeni Şekil 12'de gösterilmektedir.



Şekil 8 Tam, Düzenli, Ürün Kümesi Algoritmalarının Sınıflandırılması

Şekil 8'de gösterilen algoritmalar sadece büyük ürün kümesi oluşturan, tam ve düzenli olanlardır. Tüm sınıflandırma özelliklerinin yine tüm algoritmalara uygulanabilir olmasına rağmen, olaya daha basit bakmak açısından, bu şekil belli başlı olan algoritmaları ayıt etmek için gerekli ana özellikleri göstermektedir.

2.5. ALGORİTMALARIN KARŞILAŞTIRILMASI

Bu bölümde kıstaslarına göre çeşitli algoritmaları karşılaştırılacaktır. Veritabanı taraması esnasında sayımı yapılan aday ürün kümelerinin maksimum sayısına bakarak boşluk gereksinimleri tahmini yapılabilir. Gerekli veritabanı taramalarının ve karşılaştırma işlemlerinin maksimum sayısını sayarak zaman gereksinimlerini tahmin edebiliriz. Bu nedenle çalışmalar, tüm veritabanı üzerindeki taramaların sayısı üzerine odaklanmaktadır. En kötü durum veritabanındaki her işlem, bütün ürünlere sahip olduğunda açığa çıkar. m, her işlemdeki ürün sayısı ve L_k da D veritabanında k-ürünlerinden oluşan büyük ürün kümeleri olsun. Görülüyor ki büyük ürün kümelerinin sayısı 2^m'dir. Yüksek seviye tekniklerinde (ör: AIS, SETM, Apriori), L_1'deki bütün büyük ürün kümeleri veritabanındaki ilk tarama esnasında elde edilmiştir. Aynı şekilde L_2'deki bütün büyük ürün kümeleri ikinci tarama esnasında elde edilmiştir ve bu, bu şekilde devam eder. L_m'deki tek ürün kümesi m'inci tarama esnasında elde edilmiştir. Tüm algoritmalar, büyük ürün kümelerinde daha fazla veri elde edilemeyince sonlandırılır. Bu durumda ekstra bir tarama gerekli olur. Bu sebepten, tüm veritabanı m+1 kez taranır. Hatırlanacağı gibi Apriori-TID algoritması ilk döngüde tüm veritabanını tarar. Daha sonra (k+1)inci döngüde tüm veritabanı yerine $\overline{C}_k$ 'yı kullanır. Ancak en kötü durumda bile bunun bir yararı olmayacaktır. Bunun sebebi C_k'nın tüm işlem boyunca ürünleriyle birlikte bütün işlemleri içeriyor olmasıdır. Diğer taraftan, OCD tekniği L_1'de büyük ürün kümeleri elde etmek için algoritmanın başlangıcında tüm veritabanını sadece bir kez tarar. Dahası, OCD ve Sampling Algoritmaları, $1 \leq k \leq m$ olduğu durumda L_k'nın aday ürün kümelerini bulmak için ilk döngüde elde edilen bilgilerin ve tüm veritabanının sadece bir parçasını kullanırlar. Bu yüzden, yeterli ana belleğin verildiği en kötü durumda 2 tarama olacaktır. PARTITION tekniği, veritabanı tarama sayısını 2'ye azaltır. Aynı şekilde CARMA da en fazla 2 tarama yapar.

Bir algoritmanın iyiliği oluşturduğu "gerçek" aday ürün kümelerinin sayısının doğruluğuna bağlıdır. Daha önce de bahsettiğimiz gibi, tüm algoritmalar aday ürün kümeleri oluşturmak için önceki döngülerin büyük ürün kümelerini kullanırlar. Aday ürün kümeleri de destek sayımlarının yapılabilmesi için ana bellekte bulunmak durumundadır. Yeterli bellek mevcut olmadığından beri, farklı algoritmalar farklı çeşitlerdeki hafıza yönetimi ve depolama yapılarını önerirler. AIS, L_{k-1}'ın gerektiğinde diskte yer edebileceğini öne sürer. SETM, $\overline{C}_k$ ana belleğe sığmayacak kadar büyük olduğunda FIFO ifadesiyle diske yazılabileceğini önerir. Apriori ailesi L_{k-1}'in diskte tutulmasını ve C_k'nın bulunması için ana belleğe her seferinde bir blok getirilmesini tavsiye eder. Ancak, Apriori-TID ve Apriori-Hybrid yöntemlerinde destek sayımının elde edilebilmesi için $\overline{C}_k$ 'nın ana bellekte olması gerekir. Diğer taraftan, diğer tüm teknikler bu problemi yeterli bellek olduğunu farz ederek çözümlerler. Diğer tüm ardışık teknikler (PARTITION, Sampling, DIC, CARMA) tüm veritabanının ana belleğe sığacak uygun bir kısmını değerlendirir. Apriori ailesi, aday ürün kümelerinde olduğu kadar büyük ürün kümeleri için de uygun veri yapıları (karar ağacı, dizi) önermiştir. Buna karşılık, ne AIS ne de SETM herhangi bir depolama yapısı önermemektedir.

Birliktelik kurallarını oluşturmayı amaçlayan ticari niteliğe sahip birçok kullanılabilir uygulama, Apriori tekniğinin kullanımı üzerine bel bağlamıştır.

Bazı algoritmalar belli başlı durumlarda kullanım açısından daha elverişlidir. AIS, veritabanındaki ürün sayısı çok fazla olduğunda iyi sonuç vermez. Bu yüzden AIS, düşük önemlilikteki işlem veritabanlarında kullanılması daha uygundur. Daha önce de bahsedildiği gibi, Apriori önceki döngülerde, Apriori-TID'e göre daha az işlem zamanına ihtiyaç duyar. Diğer taraftan Apriori-TID ileriki döngülerde Apriori'ye üstünlük sağlar. Bu sebepten, Apriori-Hybrid yerine göre değiştirmeleriyle mükemmel bir performans sağlar. Ancak, Apriori'den

Apriori-TID'e yapılan bu değişiklik çok kritik ve pahalıdır. OCD yaklaşık bir teknik olmasına rağmen, düşük eşik desteğiyle sık aralıklı ürün kümelerinin bulması açısından çok etkilidir. CARMA ise işlem sıralarının bir ağdan okunduğu online kullanıcı interaktif geri besleme sistemi ile yönlendirilmiş bir teknik olarak en uygunudur.

BÖLÜM III

3. TÜKETİM ÜRÜNLERİ PAZAR SEPET ANALİZİ ÖRNEĞİ

Veri madenciliği literatüründe birliktelik kuralları ile çeşitli uygulamalar mevcuttur.

"Pazar Sepet Analizinde Ürün İlişkilerinin Bulunması" konulu bir yüksek lisans tez çalışmasında, bir marketteki alış-veriş hareketlerinden oluşan veri topluluğuna birliktelik kuralı uygulaması yapılarak birbirleri ile ilişkide bulunan ürünler tespit edilmeye çalışılmıştır. Apriori algoritmasından yola çıkarak geliştirilen bir algoritma ile verilerin analizi yapılmış ve ürünler arası birliktelik kuralları bulunmuştur.

“Yüksek Öğrenimde Öğrenci Performansının Veri Madenciliği Teknikleri ile Belirlenmesi” konulu bir tez çalışmasında, birliktelik kuralları, karar ağaçları ve istatistiksel analize dayalı veri madenciliği teknikleri, Borgelt tarafından geliştirilen iki farklı yazılım kullanılarak incelenmiştir. Kullanılan veri setinin Boğaziçi Üniversitesi Kayıt Bürosu’ndan elde edildiği bu tezin amacı, öğrenci ders performansları ve ders seçimi ile ilgili olarak bilgi tabanlı bir sistem geliştirilmesi ve ham veri şeklindeki mevcut öğrenci kayıtlarının, hem öğrenciye hem de Üniversite Yönetimine yaralı bilgilere dönüştürülmesidir.

“Bir Alışveriş Merkezinde Yapılan Satışlar İçin Sepet Analizi” uygulamasında bir marketten alınan 8 günlük hareketlerin sonucunda 8687 fiş hareketi toplanmıştır. Bu hareketler SPSS Clementine paket programına aktarılmış, apriori algoritması kullanılarak ürünler arasındaki ilişki bulunmaya çalışılmıştır.

“Apriori Algoritması ile Öğrenci Başarısı Analizi” uygulamasında Fırat Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Eğitimi bölümü öğrencilerinin notları kullanılarak öğrenci başarısı analizi gerçekleştirilmiştir. Bu bağlamda 250 öğrencinin 12 dersten aldıkları notlar ortalamaları, hazırlanmış, Apriori algoritmasını kullanan bir MATLAP programında analiz edilmiştir. Böylece birliktelik kuralı algoritmalarından Apriori’nin eğitim alanında nasıl kullanılabileceğine dair bir örnek çalışma yapılmıştır.

3.1. VERİLERİN TOPLANMASI VE VERİLERİN DÜZENLENMESİ

Bu uygulamada kullanılan veri seti Türkiye’de perakende sektöründe faaliyet gösteren market zincirlerinden bir tanesinin, 2008 Mayıs ayı içerisindeki 7 günlük zaman diliminde toplanan alışveriş hareketlerinden oluşmaktadır. Gizlilik prensipleri nedeni ile market ismi veya ürün markaları çalışmada belirtilmeyecektir.

Toplanan fişler SPSS 13’e her fişin satırına o fişte bulunan ürünler gelecek şekilde girilmiştir. Fişte yer alan ürüne “1”, fişte yer almayan ürüne “0” değeri

verilmiştir. Daha sonra bu ürünler kendi içinde kategorilerde toplanmış, yeni bir SPSS sayfası oluşturulmuş ve burada da ürünlerin fişte yer alıp almaması durumuna göre 1 veya 0 değerleri verilerek kodlanmıştır.

3.2. ANAKÜTLE ve ÖRNEKLEM

Belirli bir zaman dilimde yapılan bir gözlem söz konusu olduğundan anakütlemiz sonsuz sayıdadır. Bu süre içerisinde toplanan 2010 fiş içerisinden uygun elemeler yapılarak fiş sayısı 603'e düşürülmüştür. 603 fişte gözlemlenen 123 ürün grubu bulunmaktadır. Daha sonra bu ürün grupları birleştirilerek, birliktelikleri daha iyi tanımlayabilmek için 18 ürün kategorisi oluşturulmuştur.

Araştırmanın sonuçları incelenirken şu kısıtlar göz önünde bulundurulmalıdır. Bu çalışma araştırmanın yapıldığı markette yer alan ürünler ile gerçekleştirilmiştir. Belirli bir zaman aralığında toplanan veriler farklı dönemlerde farklı sonuçlar verecektir. Fişlerin toplandığı dönem içerisinde ürünlerin sahip oldukları kampanya ve promosyonlar alışları etkileyecektir. Çalışmanın gerçekleştirildiği marketin bulunduğu bölge ve hitap ettiği hedef kitle nedeni ile alışveriş alışkanlıkları farklılık gösterecektir.

3.3. YÖNTEM ve ALGORİTMA

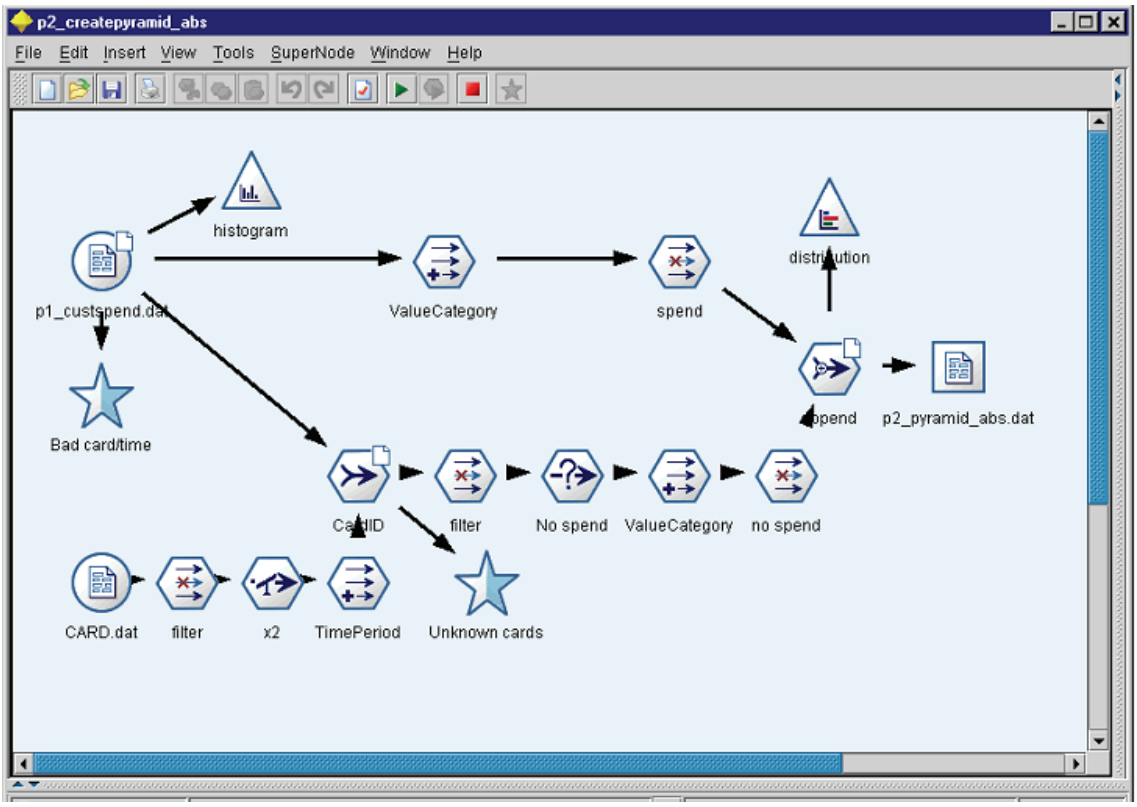
Veri setini oluşturan ürünler arasındaki ilişkinin bulunmasında bir birliktelik kuralı tekniği olan "Pazar Sepet Analizi" kullanılacaktır. Bu teknik için apriori algoritması, bir önceki bölümde ayrıntılı olarak anlatıldığı üzere, diğer algoritmalara olan üstünlüğü ve yaygın olarak kullanılması nedeni ile tercih edilmiştir.

Analizde SPSS Clementine 11.1 programı kullanılacaktır. Clementine, bir altın madencisinin kızı için yazdığı ünlü bir şarkıdan esinlenerek veri madenciliği araçlarından birine verilmiş olan isimdir. Program İngiliz menşeli bir firma olan ISL (Integral solutions Limited) tarafından geliştirilmiş ve piyasaya sunulmuştur.

Program veri girişi yapılmasına, farklı veri setlerini birleştirmeye, modellemeye ve analiz etmeye olanak sağlamaktadır

SPSS Clementine uygulamacıya, kolay kullanabileceği bir arayüz ve kısa sürede sonuca ulaşabileceği görsel bir çalışma ortamı sunmaktadır. Araştırmacının kullanmak isteyeceği tüm fonksiyonlar arayüzün alt tarafında bulunan bir araç çubuğunda ikonlar halinde yer almaktadır. Yapılması gereken, gerekli ikonların analiz sayfasına taşınarak birbirlerine bağlanması ve çalıştırılmasıdır. Tablo 3'de SPSS Clementine programına ait bir arayüz görülmektedir.

Tablo 7 SPSS Clementine Programına Ait Bir Arayüz



Veri madenciliği ve bilgi keşfi konusunda yayın ve araştırma yapan KDnuggets dergisi, “En çok tercih edilen veri madenciliği çözümü” konulu bir araştırma yapmış, üçüncüsü düzenlenen uluslar arası geleneksel anket çalışmalarının tümünde uzman sektörsel kullanıcılar, kendilerine yöneltilen sorulara verdikleri yanıtlarda SPSS Clementine’i birinci sıraya taşımışlardır.

Yaygın kullanılması ve tercih edilen bir program olması, kullanım kolaylığı, farklı kaynaklardan elde edilen veri setlerinin kolay entegre edilebilmesi özelliklerinden dolayı bu tezin uygulama bölümünde analizlerin SPSS Clementine programı ile gerçekleştirilmesine karar verilmiştir.

3.4.AMAÇ

Veri topluluğu içerisinde birbiri ile ilgili olan ürün kümelerini bularak, satış işleminde bu ürünlere uygulanacak kampanya ve promosyonlarda bu ilişkilerden yararlanmak, raf düzenini ürünlerin ilişkisine göre kurmak, dolayısıyla sayış miktarı ve satış gelirini arttırmaya çalışmaktır.

Bu amaç doğrultusunda hipotezimiz şu şekildedir;

"Veri madenciliği yöntemlerinden birliktelik kuralları kullanılarak pazar sepeti içerisinde, alınmış olan farklı ürünlerin arasındaki ilişkiler bulunabilir"

3.5. ALGORİTMANIN ELLE ÇÖZÜMÜ

Bu başlık altında veri setimizin bir kesiti kullanılarak apriori algoritmasının çözümü gösterilecek ve bu kesit için birliktelik kuralları bulunacaktır.

Tablo 8: Veri Seti Kesiti

ID	Nonfood	Meyve	Sebze	Sıvıyağ-margarin	Unlu ürünler
1	1	0	1	1	1
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	0	0	0
5	1	0	1	0	0
Toplam	2	2	3	1	3

Yukarıda ki veri setimizde, I_K= {Nonfood, Meyve, Sebze, Sıvı-yağ, Unlu ürünler} ürün setimizi, D=5 örneklemimizi temsil etmektedir. İlk olarak ürünleri alfabetik sıraya dizilir.

Tablo 9: Sıralanmış Veri Kesiti

ID	**Meyve**	**Nonfood**	**Sebze**	**Sıvıyağ-margarin**	**Unlu ürünler**
1	0	1	1	1	1
2	0	0	1	0	1
3	1	0	0	0	1
4	1	0	0	0	0
5	0	1	1	0	0
Toplam	2	2	3	1	3

Şimdi ürünlerin destekleri bulunarak sayımı yapılacaktır. Meyve için destek 2/5=0,4 olarak bulunacaktır. Aynı işlem diğer ürünler için de yapılarak tüm destek değerleri bulunur.

Tablo 10 C1 aday kümesinden oluşturulan L1 büyük ürün kümesi

C1

Ürün Kümesi	Destek
Meyve	0,4
Nonfood	0,4
Sebze	0,6
Sıvıyağ-margarin	0,2
Unlu ürünler	0,6

L1

Ürün	Destek
Meyve	0,4
Nonfood	0,4
Sebze	0,6
Sıvıyağ-margarin	0,2
Unlu ürünler	0,6

Minimum destek eşiğimiz 0,20 olsun. C1'de minimum destek eşiğini sağlayamayan ürünümüz yoktur. L1 bu şekilde oluşturulur. L1'deki ürünlerin ikili kombinasyonlarının destekleri hesaplanarak C2 oluşturulur. {Meyve, Nonfood} için destek değeri 0/5=0'dır. {Meyve, Nonfood} için güven değeri 0/2=0'dır. Diğer kombinasyonların destek ve güven değerleri de bu şekilde bulunduktan sonra C2 tablosu aşağıdaki gibi oluşmuş olur.

Tablo 11: İkili kombinasyonlar için C2 aday kümesi, destek ve güven değerleri C2

Ürün Kümesi	Destek	Güven %
Meyve, Nonfood	0,00	-
Meyve, Sebze	0,00	-
Meyve, Sıvıyağ Margarin	0,00	-
Meyve, Unlu ürünler	0,20	50
Nonfood, Sebze	0,40	100
Nonfood, Sıvıyağ Margarin	0,20	50
Nonfood, Unlu ürünler	0,20	50
Sebze, Sıvıyağ Margarin	0,20	33,33
Sebze, Unlu ürünler	0,40	66,66
Sıvıyağ-Margarin, Unlu ürünler	0,20	100

C2 aday kümesinden destek eşiğini sağlamayan ikililer çıkarılarak L2 büyük ürün kümesi oluşturulur.

Tablo 12: Destek eşiğini sağlayan ürün kümeleri

L2

Ürün Kümesi	Destek	Güven %
Meyve, Unlu ürünler	0,20	50
Nonfood, Sebze	0,40	100
Nonfood, Sıvıyağ Margarin	0,20	50
Nonfood, Unlu ürünler	0,20	50
Sebze, Sıvıyağ Margarin	0,20	33,33
Sebze, Unlu ürünler	0,40	66,66
Sıvıyağ-Margarin, Unlu ürünler	0,20	100

L2'den üçlü kombinasyonlar elde etmek için yeni destek değerleri berlilenerek C3 oluşturulur.

Tablo 13: C3 aday kümesinden oluşan L3 büyük ürün kümeleri

C3

Ürün Kümesi	Destek	Güven %
Nonfood, Sebze, Sıvıyağ-margarin	0,2	50

Nonfood, Sebze, Unlu ürünler	0,2	50
Nonfood, Sıvıyağ-Margarin , Unlu ürünler	0,2	50
Sebze, Sıvıyağ-Margarin , Unlu ürünler	0,2	33,33

L3

Ürün Kümesi	Destek	Güven %
Nonfood, Sebze, Sıvıyağ-margarin	0,2	50
Nonfood, Sebze, Unlu ürünler	0,2	50
Nonfood, Sıvıyağ-Margarin , Unlu ürünler	0,2	50
Sebze, Sıvıyağ-Margarin , Unlu ürünler	0,2	33,33

C3 aday kümesi içinde tüm ürün kümeleri destek eşik değerini sağlamaktadır. L3 büyük ürün kümesi aynen oluşturulmuştur.

Tablo 14: C4 aday kümesinden oluşan L4 büyük ürün kümeleri

C4

Ürün Kümesi	Destek	Güven %
Nonfood, Sebze, Sıvıyağ-margarin, Unlu ürünler	0,2	50

L4

Ürün Kümesi	Destek	Güven %

Nonfood, Sebze, Sıvıyağ-margarin, Unlu ürünler	0,2	50

Son olarak oluşan C4 ve L4 sonucunda oluşturulacak olan birliktelik kuralları ortaya çıkarılmıştır. Aşağıda bu kuralların tamamı bir tabloda verilmiştir.

Tablo 15: Veri Kesitinden Elde Edilen Birliktelik Kuralları

Kural	**Önce gelen (Antecedent)**	**Takip eden (Consequent)**	**Destek (Support) %**	**Güven (Confidence) %**

1	Meyve	Unlu ürünler	20	50
2	Nonfood	Sebze	40	100
3	Nonfood	Sıvıyağ-Margarin	20	50
4	Nonfood	Unlu ürünler	20	50
5	Sebze	Sıvıyağ-Margarin	20	33,33
6	Sebze	Unlu ürünler	40	66,66
7	Sıvıyağ-Margarin	Unlu ürünler	20	100
8	Nonfood	Sebze, Sıvıyağ-margarin	20	50
9	Nonfood	Sebze, Unlu ürünler	20	50
10	Nonfood	Sıvıyağ-Margarin , Unlu ürünler	20	50
11	Sebze	Sıvıyağ-Margarin , Unlu ürünler	20	33,33
12	Nonfood	Sebze, Sıvıyağ-margarin , Unlu ürünler	20	50

Veri kesitimize uyguladığımız Apriori algoritması sonucunda elde edilen birliktelik kuralları Tablo 11'de verilmiştir.

Meyve ⇒ Unlu ürünler kuralı için;

Meyve ve Unlu ürünlerin toplam fiş hareketlerinde birlikte görülme olasılıkları %20'dir. Meyve alan müşterilerin %50 olasılıkla Unlu ürünler de aldığı söylenebilir.

Nonfood ⇒ Sebze kuralı için;

Nonfood ve Sebzenin toplam fiş hareketlerinde birlikte görülme olasılıkları %40'dir. Nonfood alan müşterilerin %100 olasılıkla Sebze de aldığı söylenebilir.

Nonfood ⇒ Sebze, Sıvıyağ-margarin, Unlu ürünler kuralı için;

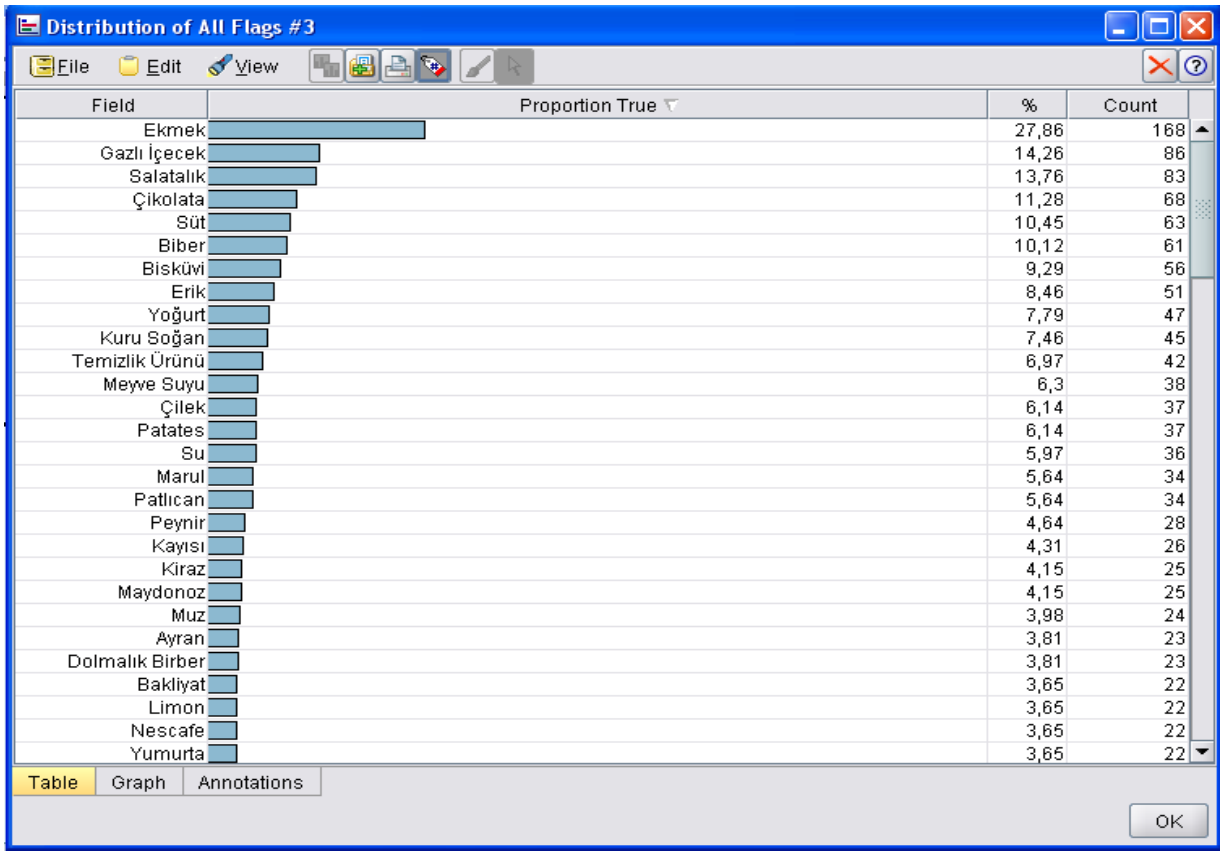
Nonfood ve (Sebze, Sıvıyağ-margarin, Unlu ürünler) ürün kümesinin toplam fiş hareketlerinde birlikte görülme olasılıkları %20'dir. Nonfood alan müşterilerin %50 olasılıkla Sebze, Sıvıyağ-margarin, Unlu ürünleri de aldığı söylenebilir.

Kurallar bu şekilde tanımlanmakta ve ifade edilmektedir.

3.6. BULGULAR

Öncelikle hareketlerde bulunan 123 ürün grubunun dağılımı saptanmış ve her bir ürün grubunun satın alınma oranlarını ve satın alınma frekanslarını gösteren dağılım tablosu elde edilmiştir. Tablo 16 bu dağılımı göstermektedir.

Tablo 16 Ürün gruplarının hareketlerdeki dağılımı



Distribution of All Flags #3

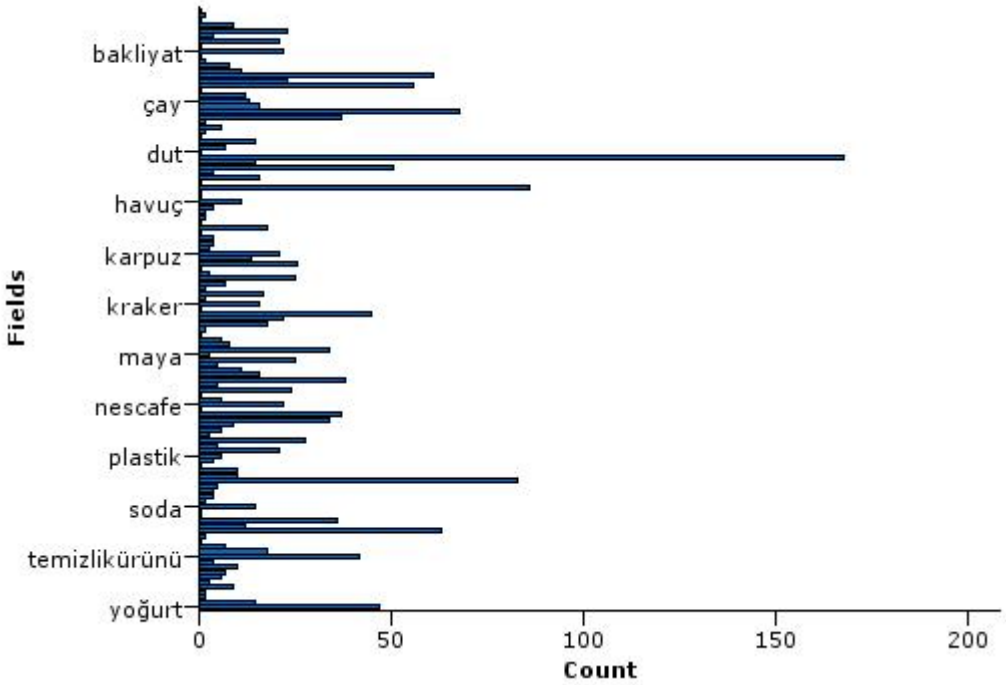
File Edit View

Field	Proportion True	%	Count
Ekmek		27,86	168
Gazlı İçecek		14,26	86
Salatalık		13,76	83
Çikolata		11,28	68
Süt		10,45	63
Biber		10,12	61
Bisküvi		9,29	56
Erik		8,46	51
Yoğurt		7,79	47
Kuru Soğan		7,46	45
Temizlik Ürünü		6,97	42
Meyve Suyu		6,3	38
Çilek		6,14	37
Patates		6,14	37
Su		5,97	36
Marul		5,64	34
Patlıcan		5,64	34
Peynir		4,64	28
Kayısı		4,31	26
Kiraz		4,15	25
Maydonoz		4,15	25
Muz		3,98	24
Ayran		3,81	23
Dolmalık Biber		3,81	23
Bakliyat		3,65	22
Limon		3,65	22
Nescafe		3,65	22
Yumurta		3,65	22

Table Graph Annotations

OK

Tablodan da görüldüğü gibi fişlerde en sık tekrarlanan ürünler ekmek (%27,86), gazlı içecekler (%14,26), salatalık (%13,76), çikolata (%11,28), süt (%10,45) ve biber (%10,12) olarak sıralanabilir. Tüm ürün gruplarının dağılımı ayrı bir şekil üzerinde aşağıda gösterilmektedir.



Şekil 9 Ürün grubu dağılımı

123 ürün grubu için birliktelik kuralları Apriori algoritması kullanılarak aşağıdaki gibi elde edilmiştir. Burada olabildiğince fazla kural görebilmek için minimum destek değer %3 ve minimum güven değeri %25 olarak alınmıştır. Sadece müşterilerin satın alınan ürünler arasındaki birliktelik kuralı incelenmiş satın alınıp-alınmayanlar arasındaki ilişki analizinin yapılmasına tezin amacı doğrultusunda gerek duyulmamıştır. Bu kısıtlar içerisinde yapılan analiz sonucunda,

veri seti içerisinde 34 birliktelik kuralı elde edilmiş ve bu kurallar Tablo 13'te verilmiştir.

Tablo 17 Ürün grupları için birliktelik kuralları

Kural	**Takip eden (Consequent)**	**Önce gelen (Antecedent)**	**Destek (Support) %**	**Güven (Confidence) %**
1	Ekmek	Yumurta	3,648	68,182
2	Ekmek	Peynir	4,643	42,857
3	Ekmek	Biber ve salatalık	3,151	42,105
4	Ekmek	Biber	10,116	40,984
5	Ekmek	Patates	6,136	40,541
6	Ekmek	Yoğurt	7,794	40,426
7	Biber	Patlıcan	5,638	38,235
8	Ekmek	Meyve suyu	6,302	36,842
9	Ekmek	Bakliyat	3,648	36,364
10	Kuru soğan	Patates	6,136	35,135
11	Biber	Salatalık ve ekmek	3,814	34,783
12	Ekmek	Marul	5,638	32,353
13	Erik	Kiraz	4,146	32,000
14	Salatalık	Biber ve ekmek	4,146	32,000
15	Salatalık	Biber	10,116	31,148
16	Ekmek	Temizlik ürünü	6,965	30,952
17	Kuru soğan	Biber dolmalık	3,814	30,435
18	Salatalık	Patlıcan	5,638	29,412
19	Patates	Kuru soğan	7,463	28,889
20	Biber	Maydanoz	4,146	28,000
21	Kayısı	Kiraz	4,146	28,000
22	Ekmek	Salatalık	13,765	27,711
23	Gazlı içecek	Neskafe	3,648	27,273
24	Kiraz	Kayısı	4,312	26,923

25	Salatalık	Kuru soğan	7,463	26,667
26	Patlıcan	Biber ve salatalık	3,151	26,316
27	Kuru soğan	Biber ve salatalık	3,151	26,316
28	Ekmek	Ayran	3,814	26,087
29	Kabak	Biber dolmalık	3,814	26,087
30	Biber	Biber dolmalık	3,814	26,087
31	Salatalık	Biber dolmalık	3,814	26,087
32	Ekmek	Süt	10,448	25,397
33	Erik	Muz	3,980	25,000
34	Çikolata	Bisküvi	9,287	25,000

Ürünler için oluşan bazı birliktelik kuralları aşağıdaki gibidir;

Ekmek ⇒ Yumurta kuralı için;

Ekmek ve Yumurtanın toplam fiş hareketlerinde birlikte görülme olasılıkları %3,65'dir. Ekmek alan müşterilerin %68,18 olasılıkla Yumurta da aldığı söylenebilir.

Ekmek ⇒ Peynir kuralı için;

Ekmek ve Peynirin toplam fiş hareketlerinde birlikte görülme olasılıkları %4,64'tür. Ekmek alan müşterilerin %42,85 olasılıkla Peynir de aldığı söylenebilir.

Patates ⇒ Kuru soğan kuralı için;

Patates ve Kuru soğanın toplam fiş hareketlerinde birlikte görülme olasılıkları %7,46'dır. Patates alan müşterilerin %28,88 olasılıkla Kuru soğan da aldığı söylenebilir.

Kabak ⇒ Biber dolmalık kuralı için;

Kabak ve Biber dolmalık toplam fiş hareketlerinde birlikte görülme olasılıkları %3,81'dir. Kabak alan müşterilerin %26,08 olasılıkla Biber dolmalık da aldığı söylenebilir.

Ürün bazında yapılan bu analiz yerine oluşturulan kategoriler ile yapılacak bir analiz daha belirgin birliktelik kuralları elde etmemizi sağlayabilir. Bu nedenle 123 ürün grubu 18 kategoride toplanmış ve tekrar birliktelik analizine tabi tutulmuştur.

Tablo 18 Kategorileştirilmiş veri seti

Table (19 fields, 603 records) #3

File Edit Generate

	Hareket...	NonFood	Meyve	Sebze	Sıvıyağ_...	Yoğurtv...	Bakliyat	Şekerli_...	Tuzlu_ü...	Çay_şe...
1	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
2	2.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
3	3.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
4	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
6	6.00	1.00	0.00	1.00	1.00	0.00	1.00	0.00	0.00	1.00
7	7.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00
8	8.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	9.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
11	11.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
12	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	13.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
14	14.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
15	15.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
16	16.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	17.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00
18	18.00	0.00	1.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00
19	19.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
20	20.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00

Table Annotations

OK

Ürün grupları birleştirilerek “0” ve “1” değerleri verilmek sureti ile yeni veri seti oluşturulmuştur. Veri seti Tablo 18’de görülmektedir.

Yeni ürün kategorilerinin satın alınma oranları ve satın alınma frekansları Tablo 19'da verilmektedir.

Tablo 19 Ürün kategorilerinin hareketlerdeki dağılımı

Distribution of All Flags #4

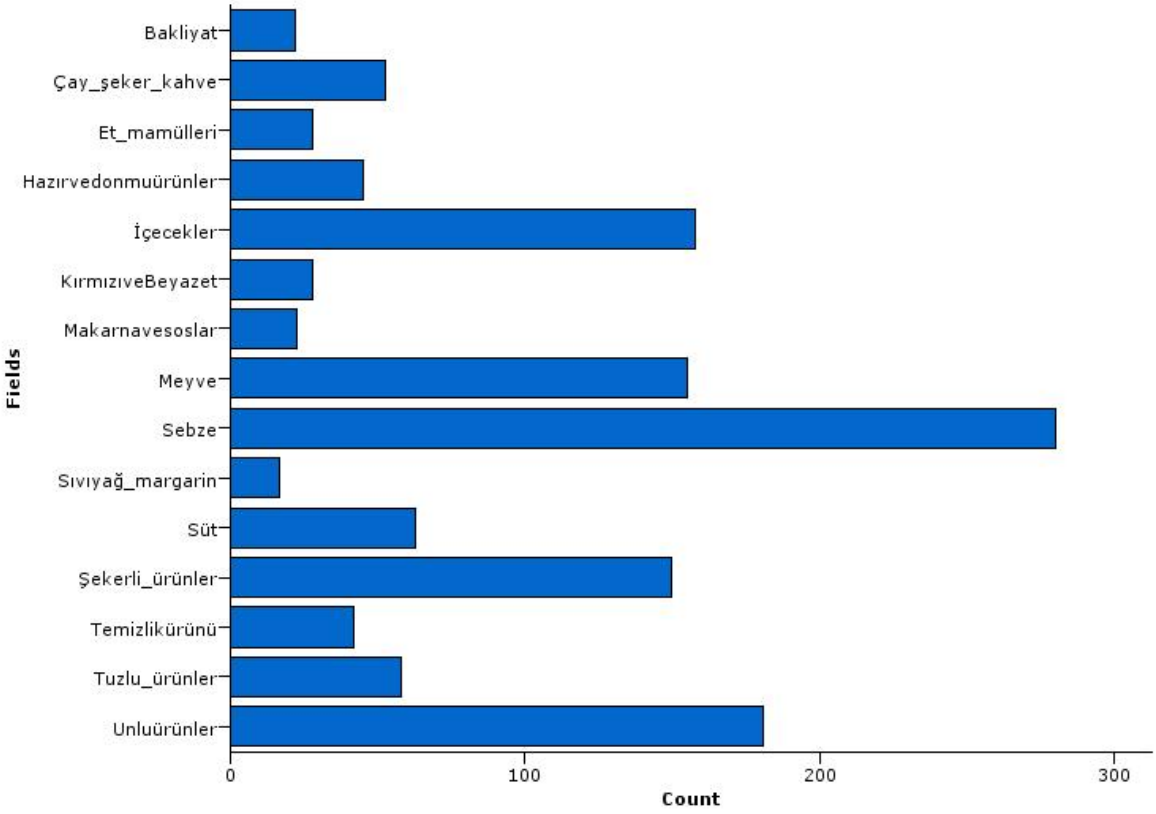
File Edit View

Field	Proportion True	%	Count
Bakliyat		3,65	22
Çay-Şeker-Kahve		8,79	53
Et mamülleri		4,64	28
Hazır ve donmuş ürünler		7,46	45
İçecekler		26,2	158
Kırmızı ve Beyaz et		4,64	28
Makarna ve soslar		3,81	23
Meyve		25,7	155
Sebze		46,43	280
Sıvıyağ ve margarin		2,82	17
Süt		10,45	63
Şekerli ürünler		24,88	150
Temizlik ürünleri		6,97	42
Tuzlu ürünler		9,62	58
Unlu ürünler		30,02	181

Table Graph Annotations

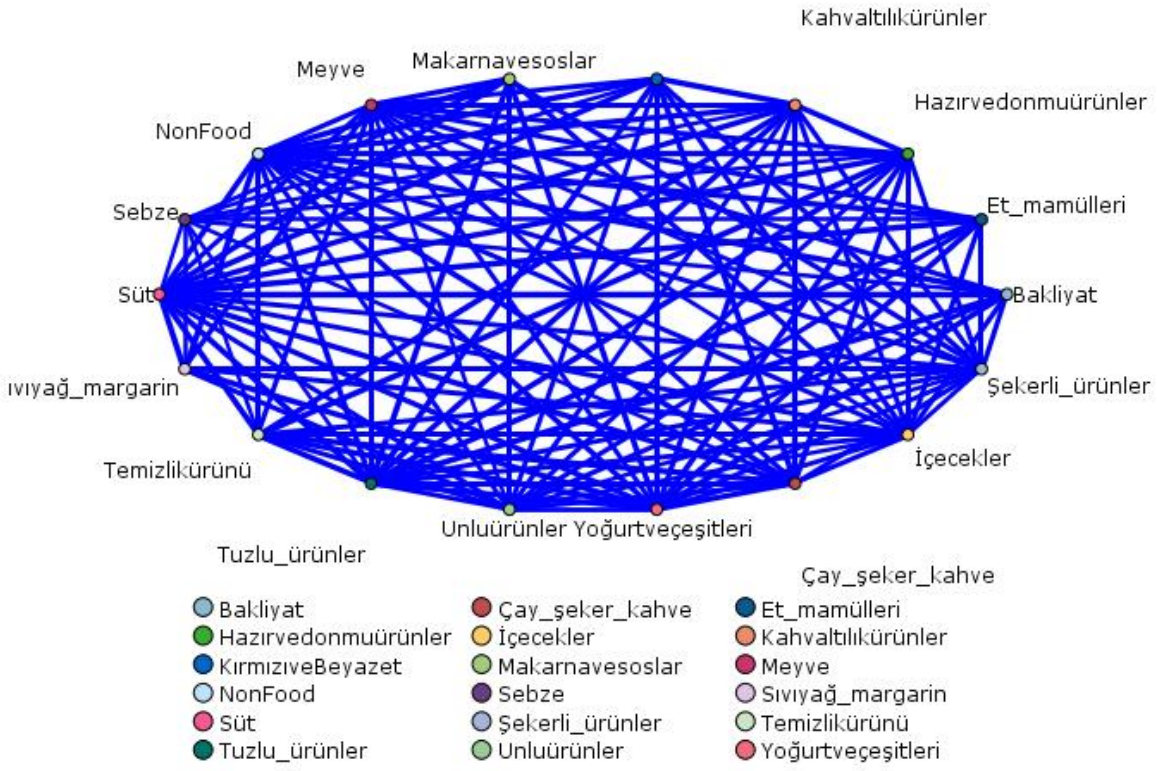
OK

Tablo 19 incelendiğinde en çok %46,43 oranı ile Sebze satın alındığı görülmektedir. Daha sonra %30,02 ile Unlu Ürünler, %26,2 ile İçecekler ve %24,88 ile Şekerli Ürünler gelmektedir.



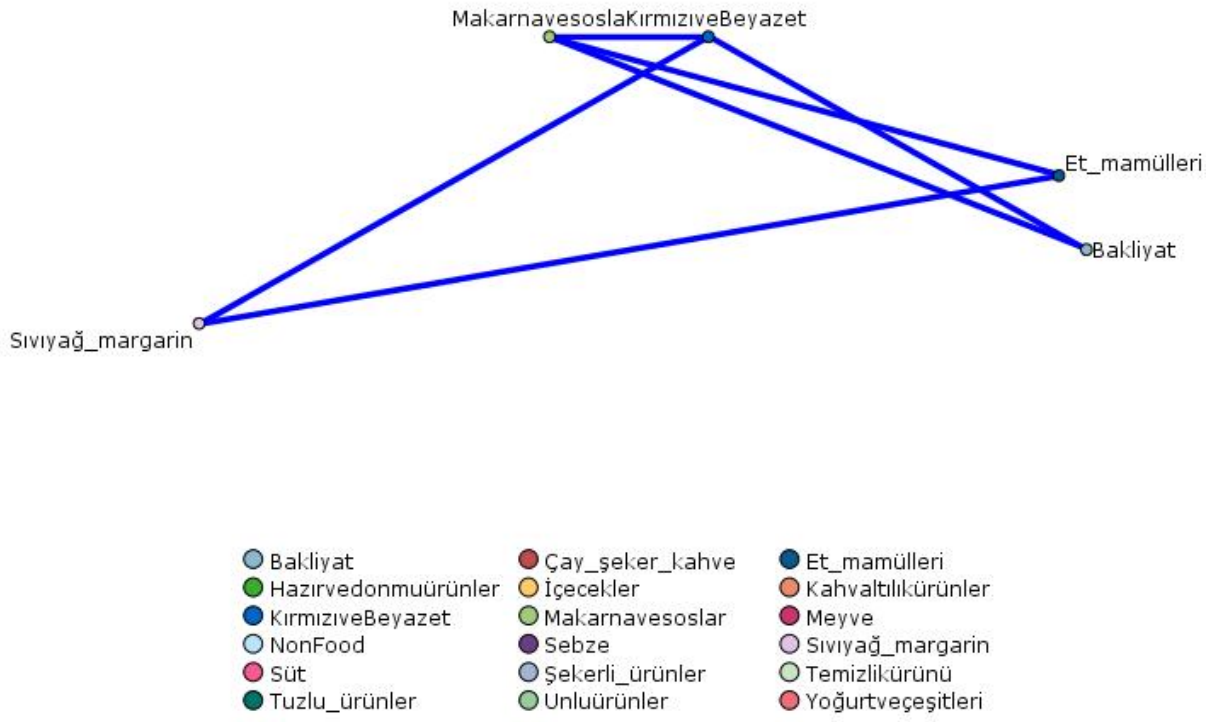
Şekil 10 Ürün Kategorilerinin dağılımı

Birliktelik kurallarının belirlenmesi için ilk olarak Web Grafiği ile kategoriler arası ilişkiler incelenmiştir. Ürün grupları için incelenen Web Grafiği anlamlı bir sonuç vermediği için yukarıda bu bölme yer verilmemiştir.



Şekil 11 Web Grafiği ile Kategoriler arasındaki ilişkiler

Kategorilerin birbirleri ile olan ilişkileri bu ağda çok yoğun olarak görülmektedir. Web grafiğinin yoğunluk değeri [0,34;0,58] ‘dir. Bu yoğunluk değeri değiştirildikçe web grafiğinde gösterilen sıklıklarda değişecektir. Bir sonraki aşamada sıklığı daha yüksek olan ilişkilerin görüntülenmesi sağlanmıştır ve Şekil 11’de gösterilmiştir.



Şekil 12 Sıklığı yüksek olan ilişkiler

Web grafiğinin yoğunluğu, [0,60;0,61] ‘dir

Kırmızı ve Beyaz et ile Sıvıyağ-Margarin ve Bakliyat birlikte sık alınmaktadır. Ayrıca Makarna ve soslar ile Kırmızı ve Beyaz et, Et Mamulleri ve Bakliyat alım sıklığı yüksektir.

Kategoriler için elde edilen birliktelik kuralları ise Tablo 20'de görülmektedir. Ürün kategorileri için yine minimum destek değeri %3 ve minimum güven değeri %25 olarak belirlenmiş analiz bu kısıtlar içerisinde gerçekleştirilmiştir.

Tablo 20 Ürün Kategorileri için Birliktelik Kuralları

Kural	**Takip eden (Consequent)**	**Önce gelen (Antecedent)**	**Destek (Support) %**	**Güven (Confidence) %**
1	Sebze	Meyve ve Unlu ürünler	5,804	60,000
2	Sebze	Unlu ürünler	30,017	53,591
3	Sebze	Meyve	25,705	50,968
4	Sebze	Süt	10,448	46,032
5	Sebze	Çay_şeker_kahve	8,789	45,283
6	Sebze	Hazır ve donmuş ürünler	7,463	42,222
7	Sebze	İçecekler ve Unlu ürünler	5,970	38,889
8	Sebze	Tuzlu_ürünler	9,619	37,931
9	Unlu ürünler	Sebze	46,434	34,643
10	İçecekler	Şekerli_ürünler	24,876	34,000
11	Unlu ürünler	Temizlik ürünü	6,965	33,333
12	Şekerli_ürünler	İçecekler	26,202	32,278
13	Unlu ürünler	Çay_şeker_kahve	8,789	32,075
14	Meyve	Hazır ve donmuş ürünler	7,463	31,111
15	Unlu ürünler	Hazır ve donmuş ürünler	7,463	31,111
16	Şekerli_ürünler	Tuzlu_ürünler	9,619	31,034
17	İçecekler	Tuzlu_ürünler	9,619	31,034
18	Şekerli_ürünler	Süt	10,448	30,159
19	Unlu ürünler	Süt	10,448	30,159
20	Unlu ürünler	Tuzlu_ürünler	9,619	29,310
21	Şekerli_ürünler	Hazır ve donmuş ürünler	7,463	28,889
22	Sebze	Temizlik ürünü	6,965	28,571
23	Meyve	Sebze	46,434	28,214
24	Meyve	Tuzlu_ürünler	9,619	27,586
25	İçecekler	Süt	10,448	26,984
26	Meyve	Süt	10,448	26,984

27	Unlu ürünler	Meyve ve Sebze	13,101	26,582
28	Şekerli_ürünler	Temizlik ürünü	6,965	26,190
29	Meyve	Temizlik ürünü	6,965	26,190

Bazı kurallar aşağıda belirtilmiştir.

Sebze ⇒ Meyve ve Unlu Ürünler kuralı için;

Sebze ve Meyve ve Unlu Ürünlerin toplam fiş hareketlerinde birlikte görülme olasılıkları %5,80'dir. Sebze alan müşterilerin %60 olasılıkla Meyve ve Unlu Ürünlerini de aldığı söylenebilir.

Sebze ⇒ Unlu Ürünler kuralı için;

Sebze ve Unlu Ürünlerin toplam fiş hareketlerinde birlikte görülme olasılıkları %30,01'dir. Sebze alan müşterilerin %53,59 olasılıkla Unlu Ürünlerini de aldığı söylenebilir.

İçecekler ⇒ Şekerli ürünler kuralı için;

İçecekler ve Şekerli ürünler toplam fiş hareketlerinde birlikte görülme olasılıkları %24,87'dir. İçecekler alan müşterilerin %34 olasılıkla Şekerli ürünlerini de aldığı söylenebilir.

İçecekler ⇒ Tuzlu ürünler kuralı için;

İçecekler ve Tuzlu ürünler toplam fiş hareketlerinde birlikte görülme olasılıkları %9,61'dir. İçecekler alan müşterilerin %31,03 olasılıkla Tuzlu ürünlerini de aldığı söylenebilir.

SONUÇ ve ÖNERİLER

Bu tez çalışmasında günümüzde sıkça kullanılan veri madenciliği tekniklerinden ve bir veri madenciliği modeli olan birliktelik kurallarından bahsedilmiştir. Uygulamanın temelini oluşturan birliktelik kuralı ve birliktelik kuralı için kullanılan algoritmalara ayrıntılı olarak yer verilmiştir.

Birliktelik kuralı, Türkiye'de perakende sektöründe faaliyet gösteren zincir marketlerden birinden toplanan fişler üzerinde uygulanmıştır. Tezin amacı birliktelik kurallarını kullanarak ürünler arasındaki anlamlı ilişkileri bulmak ve bu ilişkilerden faydalanarak çeşitli kampanya ve promosyonların, raf düzenlerinin ve reklam kampanyalarının belirlenmesini sağlamak dolayısı ile satış miktarını ve satış gelirini arttırmaya çalışmaktır.

Marketten 7 gün boyunca toplanan fişler birliktelik kuralına uygun olarak bilgisayar ortamına aktarılmış ve analizler için kullanımı çok yaygın olan ve çoğu sektör tarafından veri madenciliği programları arasında ilk sırada gösterilen SPSS Clementine paket programı tercih edilmiştir.

Ürünlerin dağılımları incelendikten sonra apriori algoritması ile veri seti taranarak birliktelik kuralları bulunmuştur. Önce ürün grupları daha sonra ise ürün

kategorileri için aynı işlem gerçekleştirilmiş ve elde edilen kurallar tablolar halinde sunularak ilgili açıklamalar yapılmıştır.

Çıkan sonuçlar incelendiğinde ekmek ve gazlı içecek alımlarının diğer ürünlere göre daha fazla olduğu görülmektedir. Tabi burada araştırmamızın kısıtları göz önünde bulundurulmalıdır. Bunlar şu şekilde sıralanabilir; Çalışma sadece araştırmanın yapıldığı markette yer alan ürünler ile gerçekleştirilmiştir, belirli bir zaman aralığında toplanan veriler farklı dönemlerde farklı sonuçlar verecektir, fişlerin toplandığı dönem içerisinde ürünlerin sahip oldukları kampanya ve promosyonlar alışları etkileyecektir ve çalışmanın gerçekleştirildiği marketin bulunduğu bölge ve hitap ettiği hedef kitle nedeni ile alışveriş alışkanlıkları farklılık gösterecektir.

En yüksek güvenilirlikle elde edilen ekmek ve yumurta, ekmek ve peynir, ekmek ve biber ve salatalık arasındaki ilişkiler bize market çevresinde yaşayan kitlenin gelir seviyesinin düşük olduğu ve beslenme alışkanlıklarının bu doğrultuda oluştuğunu gösterebilir.

Çalışma sonucunda elde edilen bilgiler, daha etkin bir raf düzenlemesi veya farklı ürün kampanyaları düzenlemede kullanılabilir. İlişki kurulan ürünlerin daha çok temel ürünler olması nedeni ile market yetkilileri bu ürünlerin stoklarını daha sık kontrol edebilir ve uygun düzeyde tutabilir.

Bundan sonraki çalışmalarda, ürün gruplarının yanı sıra müşteri profilini de içeren bir veri seti üzerinde çalışılabilir. Ürünler için elde edilen birliktelik kuralları müşteri profiline göre gruplandırılabilir ve farklı kampanyalar geliştirilerek müşterilerin farklı ürünleri de almaları sağlanabilir. Ya da farklı müşteri grupları hedef alınarak yürütülecek bir reklam kampanyası sayesinde yeni müşteri grupları oluşturulabilir.

Diğer yandan zaman serilerini de içinde barındıran bir birliktelik kuralı analizi gerçekleştirilebilir. Böylece haftanın hangi günlerinde hangi ürünlerin daha ilişkili oldukları veya hangi aylarda hangi ürünlerin daha çok birlikte alındıkları üzerine bir

çalışma yapılarak gerekli kampanyalar ve düzenlemeler aylara veya günlere göre düzenlenebilir.

KAYNAKÇA

1) Kiremitçi, Barış, "Veri Ambarlarında Veri Madenciliği ve Ulaştırma-Lojistik Sektöründe Bir Uygulama", (Basılmamış Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, 2005)

2) Sarıkan, Burçin, "Veri Madenciliği ve İstatistik", (Basılmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, 2003)

3) Öztemel, Ercan, "Yapay Sinir Ağları", Pataya Yayıncılık Eğitim, İstanbul, 2003

4) Alpaydın, Ethem, "Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri", Bilişim 2000 Semineri, http://www.cmpe.boun.edu.tr/~ethem/files/papers/veri-maden_2k-notlar.doc, (11 Nisan 2008)

5) Aydoğan, Fatih, "E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı ve Gerçekleştirimi", (Basılmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, 2003)

6) Türkan, Fatma, "Uzay Veri Madenciliği", Gebze İleri teknoloji Enstitüsü, 2007

7) Özçakır, F. Cemal, "Müşteri İşlemlerindeki Birlikteliklerin Belirlenmesinde Veri Madenciliği Uygulaması", (Basılmamış Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, 2006)

8) Silahtaroğlu, Gökhan, "Kavram ve Algoritmalarıyla Temel Veri Madenciliği", 1. Basım, İstanbul, Papatya Yayıncılık Eğitim, 2008

9) Eker, Hakan, "Veri Madenciliği veya Bilgi Keşfi", http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=538, 10 Haziran 2008

10) Kona, H. Vali, "Association Rule Mining Over Multiple Databases: Partitioned and Incremental Approaches", The University of Texas , Arlington, 2003

11) Akpınar, Haldun, "Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği", İstanbul Üniversitesi İşletme Fakültesi Dergisi, C.29, no.1, 2000

12)Jiawei Han; Micheline Kamber, "Data Mining: Consepts and Tecniques", Morgan Kaufmann Publishers, Inc., San Francisco, California, 2001

13) Yaralıoğlu, K., "Veri Madenciliği", Dokuz Eylül Üniversitesi Web Yayın, http://www.deu.edu.tr/userweb/k.yaralioglu/dosyalar/ver_mad.doc, 15 Mayıs 2008

14) Dunham, M.H., Xiao, Y., Gruenwald, L., Hossain, Z., "A Survey Of Association Rules", Southern Methodist University, University of Oklahoma, 2001

15) Goebel, Michael, Gruenwald, Le, "A Survey of Data Mining and Knowledge Discovery Software Tools", ACM SIGKDD Explorations Newsletter, Vol.1, no.1, 1999

16) Dolgun, M. Özgür, Zor, İbrahim, "Bir Alışveriş Merkezinden Yapılan Satışlar İçin Sepet Analizi", Hacettepe Üniversitesi, SPSS Yazılım Eğitim Danışmanlık Ticaret Ltd. Şti., 2006

17) Karabatak, Murat, İnce, M. Cevdet, "Apriori Algoritması ile Öğrenci Başarısı Analizi", Fırat Üniversitesi, 2004

18) Tan, Pang-Ning, Steinbach, Michael ve Kumar, Vipin "Association Analysis: Basic Concepts and Algorithms", Hidayet Takçı (çev.), http://www.bilmuh.gyte.edu.tr/BIL454/birliktelik_kurallari.pdf, 5 Haziran 2008

19) Toktaş, Peral ve Demirhan, M. Başak, "Risk Analizinde Veri Madenciliği Uygulamaları", yöneylem araştırması/Endüstri Mühendisliği XXIV Ulusal Kongresi, 2004

20) Agrawal, Rakesh, Shafer, J. Christopher, "Paralel Mining of Association Rules" IEE Transactions on Knowledge and Data Engineering, Vol.8, No.6, s.1

21) Agwaral, Rakesh, Srikant, Ramakrishnan, "Fast Algorithms for Mining Association Rules", IBM Research Report RJ9839, 1994

22) Agrawal, Rakesh, Imielinski, Tomasz, Swami, Arun, "Mining Associations Between Sets of Items in Large Databases", ACM SIGMOD Conf. Management of Data, 1993

23) Özekes, Serkan, "Veri Madenciliği Modelleri ve Uygulama Alanları" İstanbul Ticaret Üniversitesi Dergisi, http://www.iticu.edu.tr/kutuphane/dergi/d3/M00041.pdf, (21 Mayıs 2008)

24) Özmen, Şule, "İş Hayatı Veri Madenciliği ile İstatistik Uygulamalarını Yeniden Keşfediyor", Marmara Üniversitesi, İ.İ.B.F., http://idari.cu.edu.tr/sempozyum/bil38.htm, (12 Haziran 2008)

25) Bilgin, T. Tugay, "Veri Madenciliğinde Kümeleme Analizi Yönetimi Uygulaması", (Basılmamış Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, 2003)

26) Şimşek, U. Tuğba, "Veri Madenciliği ve Müşteri İlişkileri Yönetiminde (CRM) Bir Uygulama", (Basılmamış Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, 2006)

27) Kaya, Yalçın, "Pazar Sepet Analizinde İlişkilerin Bulunması", (Basılmamış Yüksek Lisans Tezi, Gebze İleri teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü, 2003)