T.C.

BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK BİLGİSAYAR ANABİLİM DALI
BİLGİSAYAR AĞLARI VE İNTERNET TEKNOLOJİLERİ BİLİM DALI

METİN MADENCİLİĞİ YÖNTEMİYLE TÜRKÇEDE EN SIK KULLANILAN VE BİRBİRİNİ TAKİP EDEN HARFLERİN ANALİZİ VE BİRLİKTELİK KURALLARI

(Yüksek Lisans Tezi)

Tezi Hazırlayan: Emine Kübra ÇELİKYAY

İSTANBUL, 2010

T.C.

BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK BİLGİSAYAR ANABİLİM DALI
BİLGİSAYAR AĞLARI VE İNTERNET TEKNOLOJİLERİ BİLİM DALI

METİN MADENCİLİĞİ YÖNTEMİYLE TÜRKÇEDE EN SIK KULLANILAN VE BİRBİRİNİ TAKİP EDEN HARFLERİN ANALİZİ VE BİRLİKTELİK KURALLARI

(Yüksek Lisans Tezi**)**

Tezi Hazırlayan:
Emine Kübra ÇELİKYAY
Öğrenci No:
070861004

Danışman:
Yrd. Doç. Dr. Gökhan SİLAHTAROĞLU

İSTANBUL, 2010

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum "Metin madenciliği yöntemiyle Türkçede en sık kullanılan ve birbirini takip eden harflerin analizi ve birliktelik kuralları" başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmanın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 12/08/2010

Aday: Emine Kübra ÇELİKYAY

METİN MADENCİLİĞİ YÖNTEMİYLE TÜRKÇEDE EN SIK KULLANILAN VE BİRBİRİNİ TAKİP EDEN HARFLERİN ANALİZİ VE BİRLİKTELİK KURALLARI

Tezi Hazırlayan: Emine Kübra ÇELİKYAY

Özet

Günümüzde bilgisayar sistemleri her geçen gün ucuzluyor ve güçlerini artırıyor. Bilgisayar teknolojilerinin ilerlemesi ve dijital verilerin artmasıyla birlikte veri madenciliği de büyük önem kazanmıştır. Veri madenciliğinin bir alt türü olan metin madenciliği, metin yığınları arasına gizlenmiş bilgilerin ortaya çıkarılmasıdır. İnternetin yaygınlaşmasıyla birlikte metin sayısının buna paralel olarak artması metin madenciliğinin önemini artırmıştır. Yazılı halde bulunan bilgiler metin madenciliği teknikleri kullanılarak yapısal ve anlamlı hale getirilebilir sonra veri madenciliği analizlerinde kullanılabilir. Metin madenciliğinin temel amacı , veri madenciliğinde kullanılacak anlamlı ve yapısal verilerin elde edilmesidir. Ayrıca metin madenciliği metin yığınlarının özet çıkarımı ve anlamsal çıkarım elde edilebilir.

Anahtar Kelimeler: Veri Madenciliği, Birliktelik Kuralı, Birliktelik Kuralı Algoritmaları, Apriori Algoritması, Metin Madenciliği, Metin Madenciliği Algoritmaları

BY THE METHOD OF TEXT MINING, ANALİZE OF MOST FREQUENTLY USED AND SUCCESSİVE WORDS IN TURKISH AND COOCCURENCE RULES

Presented by: Emine Kübra ÇELİKYAY

Abstract

Today, computer systems are getting cheaper and their power is increasing day by day. Together with the progress of computer technology and digital data increase, the data mining has gained great importance. Text mining which is a sub-type of data mining is disclosure of text information hidden among piles. Proliferation of internet in parallel with the increase in the number of texts has increased the importance of text mining. Using text mining techniques, information in written forms can be made structured and meaningful. And then it becomes available in the data mining analysis. The main goal of text mining is to obtain meaningful and structured data in data mining. Moreover by using text mining, semantic and summative inference of the text stacks can be obtained.

Key words: Data Mining, Association Rules, Association Rule Mining Algorithms, Apriori Algorithm, Text Mining, Text Mining Algorihtms

İÇİNDEKİLER

Sayfa No.

ŞEKİLLER LİSTESİ

Sayfa No.

KISALTMALAR

DBMS : Database Management Systems

FIFO : First In First Out

G/Ç : Giriş/Çıkış

KDD : Knowledge Discovery in Databases

LIFO : Last In First Out

OLAP : On-line Analytical Processing

OLTP : On-line Transaction Processing

RDBMS : Relational Database Management Systems

SQL : Structured Query Language

TID : Transaction ID

WWW : World Wide Web

1. GİRİŞ

Verilerin dijital ortamlarda saklanmaya başlamasıyla birlikte, veri miktarının her yirmi ayda bir iki katına çıktığı varsayılmaktadır. Veritabanı boyutlarının, her gün üretmekte oldukları bilgi miktarındaki büyük artışlar ile terabaytı aştığını söyleyebiliriz. Buna paralel olarak bilgisayar teknolojisinin sürekli gelişiyor ve ucuzluyor olması, bu verilerinin saklanabilmesi imkanını sağlamaktadır. Ancak veriler belirli bir amaç doğrultusunda işlendiklerinde anlam kazanabilmektedirler. Bu büyük miktardaki ham veriden gelecekle ilgili tahmin yapılmasını sağlayan anlamlı bağıntı ve kuralların keşfedilmesi gerekmektedir. Ancak bu büyük miktardaki verilerin analizleri gözle ve elle yapılamayacağı için veri madenciliği ve teknikleri ortaya çıkmıştır. Anlamsız verilerden anlamlı ve kullanılabilir bilgiler elde etmede veri madenciliği önemli bir rol oynamaktadır.

Veri madenciliği üzerindeki eski çalışmalar ilişkisel görev ile ilişkili veriler üzerine yoğunlaşmıştır. Ancak, www dünyasında inanılmaz gelişmeler sonucu gerçekte elde edilebilir bilginin büyük bir çoğunluğu metin veri tabanları üzerinde saklanmaktadır. Bu veri tabanları, makaleler, araştırma yazıları, kitaplar, sayısal kütüphaneler, e-posta mesajları ve web sayfaları gibi çeşitli kaynaklardan, büyük ölçekli doküman koleksiyon oluşmaktadır. Geleneksel bilgi kazanım teknikleri, metin verilerinde bilgi çıkarımında etkisiz kalmış ve bunun sonucu olarak da metin veri madenciliği çalışmaları hızla yayılmıştır.

Bu çalışmada Türkçede en sık kullanılan harf, karakter, ikili ve üçlü olarak birbirini izleyen harflerin metin madenciliği yoluyla belirlenmesine çalışılmıştır.

Tez altı bölümden oluşmaktadır. İlk bölümde veri madenciliği kavramı öncesinde bilinmesi gereken veritabanı veri ambarı ile ilgili temel kavramlar hakkında bilgi verilmiştir.

Tezin ikinci bölümünde veri madenciliği hakkında inceleme yapılmış olup, veri madenciliği ve bilgi keşfi sürecine değinilmiştir.

Üçüncü bölümde birliktelik kuralları madenciliği kavramı ve bu amaçla kullanılan algoritmalar ele alınmıştır.

Dördüncü bölümde metin ve web madenciliği ele alınarak metin madenciliği algoritmaları incelenmiştir.

Beşinci bölümünde en çok bilinen birliktelik kuralı algoritması olan Apriori algoritması baz alınarak metin sınıflandırılması yapılmıştır. Bu bölümde tez çalışmasında kullanılan programlama dilinin uygulama ve çalıştırılması anlatılmıştır.

Altıncı ve son bölüm olan sonuç bölümünde ise Türkçe metinler üzerinde genel bir değerlendirme yapılmış ve bu çalışma yöntemi ile çeşitli önerilerde bulunulmuştur.

2. VERİTABANI ve VERİ AMBARI

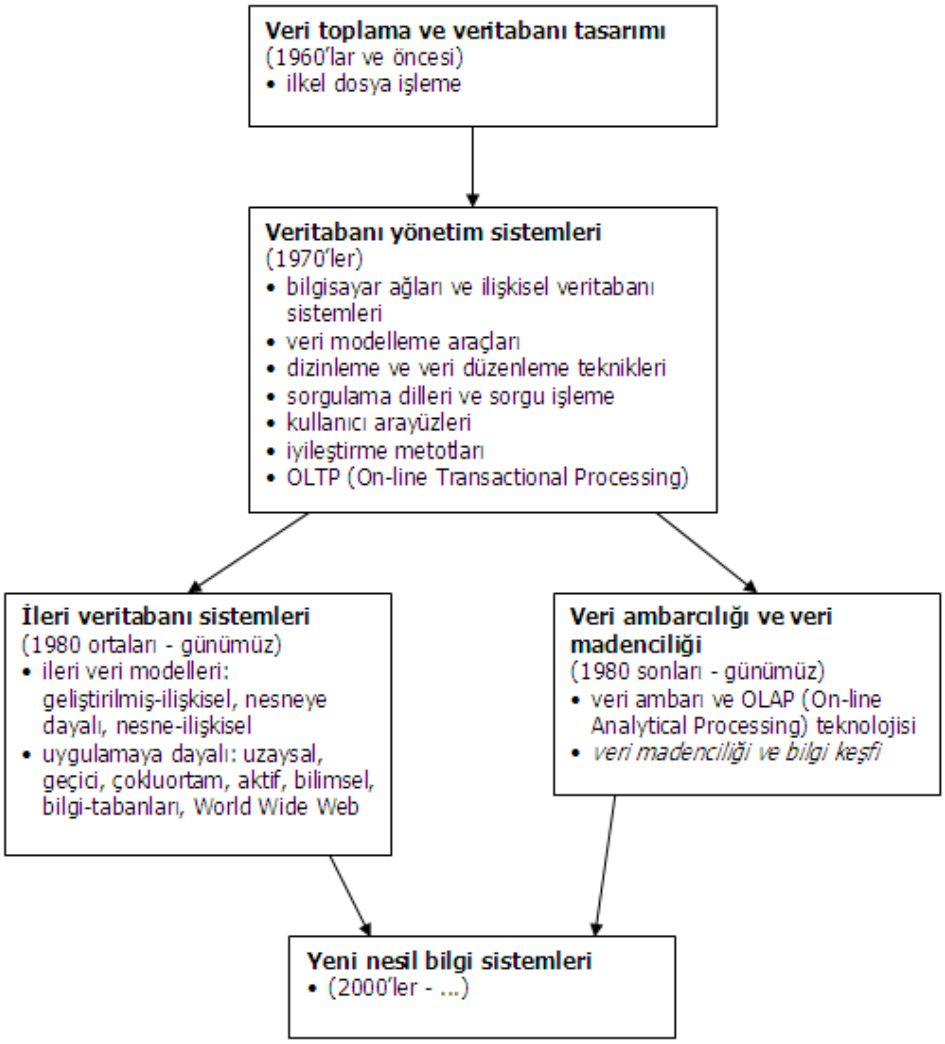
2.1. Veritabanı

Veritabanı bilgi depolayan düzenli bilgiler topluluğudur. Bilgisayar ortamında düzenli bilgilerle sınırlı olmamakla birlikte daha çok bu anlamda kullanılır. Veritabanı organizasyonun her aşamasında veriye ulaşmaktadır. Veritabanları sistematik erişim imkânı olan, hızlı bir şekilde yönetilebilen, sorgularda istenen sonuçlara anında ulaşılabilen, güncellenebilen, taşınabilen, birbirleri arasında tanımlı ilişkiler bulunan veriler kümesidir [1].

Veritabanı, en geniş anlamıyla; birbirleriyle ilişkili verilerin tekrara yer vermeden, çok amaçlı kullanımına olanak sağlayacak şekilde depolanmasıdır.

Bir veritabanını oluşturmak, saklamak, çoğaltmak, güncellemek ve yönetmek için kullanılan programlara Veritabanı Yönetim Sistemleri (DBMS) adı verilir. İlişkisel Veritabanı Yönetim Sistemleri (RDBMS), büyük miktardaki verilerin güvenli bir şekilde tutulduğu, bilgilere hızlı bir şekilde erişim imkânı sağlayan, bilgilerin bütünlük içinde tutulduğu ve birden fazla kullanıcıya aynı anda bilgiye erişim imkânı sağlayan programlardır [1].

Günümüzde veritabanı sistemleri bankacılıktan otomotiv sanayisine, sağlık bilgi sistemlerinden şirket yönetimine, telekomünikasyon sistemlerinden hava taşımacılığına, çok geniş alanlarda kullanılan bilgisayar sistemlerinin alt yapısını oluşturmaktadır.



Şekil.1. Veritabanı Teknolojisinin Gelişimi [2]

2.1.1. **Veri, Metaveri** ve **Bilgi**

Günlük hayatta *veri* (data), *bilgi* (information) ile eş anlamlı olarak kullanılır. Ancak, düzenlenmemiş bir ölçüm olarak nitelendirilebilecek veri düzenlendiğinde bilgiye dönüşmektedir. Veri kendi başına değersizdir, hiçbir anlam ifade etmez. Örneğin veritabanından alınan "345" verisinin müşteri ID'si mi, tutar mı, yoksa ürün numarası mı? olduğu bilinmiyorsa, bu veri bilgi içermez. İsteğimiz, amacımız doğrultusunda bilgidir. Bilgi, bir amaca yönelik işlenmiş veridir. Bir soruya yanıt vermek için veriden çıkarılan olarak tanımlanabilir [3].

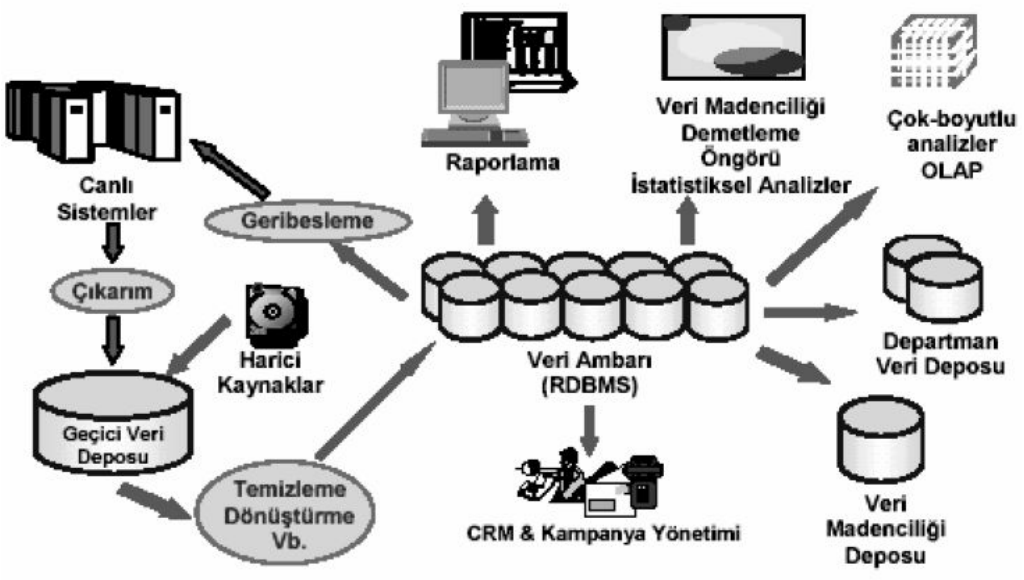
İnformation (enformasyon): Veri kavramının tanımından yola çıkarak adreslemedeki ikinci safhadır. Verilerin ilişkilendirilmiş, düzenlenmiş, anlamlandırılmış, işlenmiş bir halidir. Potansiyel olarak içinde bilgi barından bir veri halindedir.

Bilgi (knowledge) : Bu süreçteki üçüncü aşama, enformasyonun bilgiye dönüşmesi, bireyin onu algılaması, özümsemesi ve sonuç çıkarılmasıyla gerçekleşir. Dolayısıyla bireyin algılama yeteneği, yaratıcılık, deneyim gibi kişisel nitelikleri de bu süreci doğrudan etkilemektedir. Günlük hayatta veri, bilgi ile eşanlamlıdır. Veri düzenlendiğinde bilgiye dönüşmektedir. Veri kendi başına değersizdir, hiçbir anlam ifade etmez. Bilgi bir amaca yönelik işlenmiş veridir.

Veriyi bilgiye çevirmeye **veri analizi** denir. Veriyi oluşturan sayılar, harfler ve onların anlamı, **metaveri** (metadata) olarak bilinir. Metaveri "veri hakkında veri" olarak tanımlanabilir. [4] Metaveri her veri elementinin anlamını, hangi elementlerin hangileriyle nasıl ilişkili olduğunu ve kaynak verisi ile erişilecek veri gibi bilgileri içermektedir.

2.2. Veri Ambarı

Veri ambarı, detay bilgilerinin özet bilgiler haline getirilmiş halidir. Girilmiş ve girilmekte olan detay bilgilerinin belirlenen periyot zamanlarda (saatlik, günlük, haftalık, aylık vb.) çeşitli özet tablolarda birleştirerek toplanmaktadır. Sorgulamalar bu özet tablolar üzerinden yapılmaktadır ve kullanıcı detay bilgiye ulaşmak istediğinde çalışan veritabanına ulaşmaktadır. Böylece sürekli bilgi girişi yapılarak veritabanı sistemi meşgul edilmektedir. Böylece veri ambarları kullanılması doğmuştur.



Şekil.2. Veri Ambarı [5]

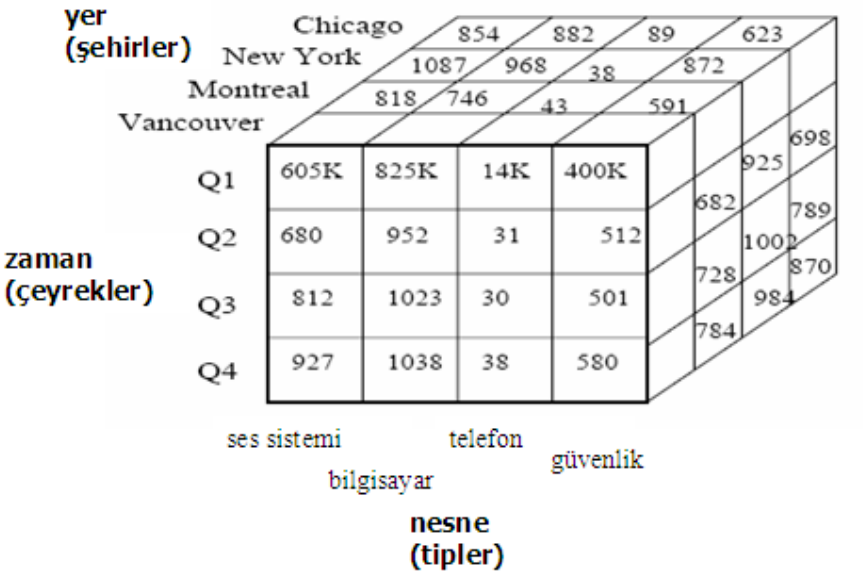
Veri ambarı ilişkili verilerin sorgulandığı ve analizlerinin yapılabildiği bir depodur. Veri ambarı, analiz ve sorgular için kullanılabilir bütünleşmiş bilgi deposudur. Veri ambarı başlangıçta farklı kaynaklardan gelen verinin üzerinden daha etkili ve daha kolay sorguların yapılmasını sağlamaktadır [1].

Veri ambarı ilgili veriyi kolay, hızlı ve doğru biçimde analiz etmek için gerekli işlemleri yerine getirir. Veri ambarı, işlemsel sistemlerdeki veriyi kopyalayıp karar verme işlemi için uygun formda saklar. Veri ambarları metadatalardan oluşur. Tutulan veri hakkındaki ihtiyaç duyulan tüm bilgiler metadata olarak adlandırılır.

Veri madenciliği gibi uzun ve karmaşık süreçler sonucunda analizler yapılabilir. Veri ambarı bir işletmenin değişik birimleri tarafından toplanan bilgileri, gelecekte kullanılabilecek ya da değerlendirilebilecek olanlarının arka planda yığılarak birleştirilmesinden oluşan büyük çaplı bir veri deposudur. Veri ambarı ilgili veriyi kolay, hızlı ve doğru biçimde analiz etmek için veri ambarı, işlemsel veritabanlarındaki veriyi kopyalayıp, karar verme işlemi için uygun formda saklar.

OLAP hızlı gözden geçirim, özetleme ve veri analizi için dizayn edilmiştir. Çok boyutlu veritabanı motorundan verinin çok boyutlu gösterimine olanak sağlar.

Veri ambarlarında saklanan veri analizi için OLAP araçları temin edilmiştir. OLAP kullanıcının sormayı düşündüğü sorgularla sınırlıdır.



Şekil.3. Yer, Zaman, Nesne Boyutlarını (dimension) ve Rakamsal Ölçümleri (measure) içeren 3-boyutlu OLAP Veri Kübü [2]

Bir veri ambarının olması, OLAP'a ihtiyaç olmadığı anlamına gelmez. Veri ambarları ve OLAP birbirlerini tamamlar. Veri ambarı verileri uygun şekilde tutmaya ve kontrol etmeye yarar. OLAP ise, veri ambarı verilerini stratejik bilgilere dönüştürmeye yarar.

Bir şirket yapısı içerisinde, departmanlar bazında incelenecek olursa:

- Pazarlama departmanlarında OLAP'ın en yaygın kullanım alanları, pazar araştırmalarında, satış tahminleri, promosyon ve kampanya analizleri, müşteri analizleri ve pazar/müşteri segmentasyonlarıdır. Veri madenciliği sonuçlarının değerlendirilmesi ve demografikler bazında incelenmesi seviyesinde de olmazsa olmaz araçlardan biri olarak yer almaktadır.

- Üretim ile ilgili uygulamaları ise en yoğun olarak üretim planlama ve hata analizleridir. Özellikle senaryo geliştirmekte ve farklı ürün tipleri ile çalışılan yapılarda, çok boyutlu düşünme imkânı sayesinde maliyetler ve fiyatlamalar kolaylıkla çıkarılabilmektedir.

- Finans departmanları ise OLAP'ı bütçeleme, activity-based costing, finansal performans analizleri ve finansal modelleme amaçları ile kullanabilir. Özellikle birlik konusunda oluşturulacak modeller, çok büyük kolaylıklar sağlamaktadır. Strateji belirleme, satış analizleri ve gelecek tahminleri ise, satış departmanlarındaki OLAP uygulamalarıdır [2].

Veri ambarı müşterilerin gizli kalmış satın alma eğilimlerini tespit eder, satış analizleri ve trendleri üzerine odaklanır, finansal analiz, stratejik analiz yapabilir.

Veri ambarları off-line çalışır. Veri ambarları analiz yapar, uzun ve karmaşık süreçler sonunda veritabanından istenilen sorgulara anında ulaşır. Veri ambarlarında tekrarlanan kişi, yer tek bir alanda toplanmış ve düzenlenmiştir. Aynı olay ve veriler birbirine bağlanmıştır.

Veri ambarındaki veriler, veri ambarı yöneticisinin kullandığı teknik veriler ve veri ambarı kullanıcılarının kullandığı iş verileri olarak ikiye ayrılır:

1. **Teknik veriler**: Operasyonel veritabanı tanımlarını ve veri ambarı tanımlarını içerir.

   Bu iki tanım veya şema veri ambarını çalıştırılabilmesini sağlayan veri taşıma operasyonlarını içerir. Bu bilgiler veri ambarı yöneticisine veri ambarında birbiriyle ilişkili verileri göstererek yardımcı olan bilgilerdir.

2. **İş verileri:** Kullanıcılara yardım eder. Kullanıcıların veritabanı oluşturan veriler dışındaki veri ambarında bulunan bilgilere ulaşmalarına yardımcı olur. Ayrıca veri ambarına verinin ne zaman ve nereden geldiği gibi bilgilere de ulaşılmasını sağlar [1].

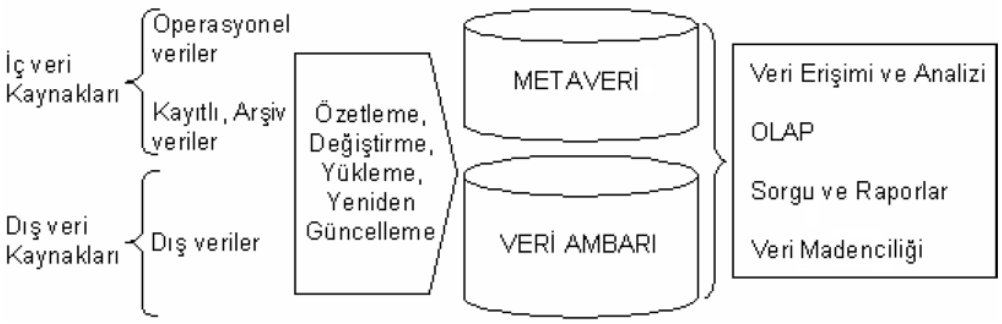
2.3. Datamart

Datamart (veri pazarı) şirketlerde belirli bir bilgi kullanıcısı grubunun ihtiyacına yönelik olarak hazırlanan küçük boyutlu veri ambarı olarak tanımlanabilir. Datamartlar veri ambarının alt kümesi olan 1 - 10 GB'lık bölümsel ambarlardır ve

organizasyonların ve işletmelerin belirli kullanıcıları için ayrılmış ve onlara ait verileri içerirler [1].

2.4. Veri Ambarının Oluşturulması

Veri ambarı, veri madenciliğinde önemli bir aşamayı oluşturmaktadır. Veri ambarını oluşturulması veri madenciliğinin maliyet ve zamanın önemli bir kısmını oluşturmaktadır. Veri madenciliğini yapacağımız veri tek bir yapı içerisinde bulunmayabilir. Bu sebeple bilginin tek bir çatı altında toplanması gerekmektedir. Burada verinin tek bir çatı altında toplanması çözüm değildir. Aynı zamanda veriler içindeki hata ve belirsizliklerin temizlenmesi de gerekmektedir. Bu aşamadaki işlemler veri toplama, uyumlandırma, birleştirme ve temizlenme, seçme ve dönüştürmedir.



Şekil.4. Veri Ambarı Bileşenleri [4]

2.4.1. Toplama

Veriler farklı kaynaklarda olabilir. Verilerin farklı kaynaklardan alınarak bir kayıtta birleştirilmesi işlemine toplama işlemi denir. Bilgilerin farklı kaynaklarda olması durumunda bilgi keşfi için toplama işlemi gerekmektedir.

2.4.2. Uyumlandırma

Veri ambarında veriler farklı kaynaklardan toplandıktan sonra uyumsuz veri tipleri karşımıza çıkacaktır. Veri ambarında başarı veri tiplerinin uyumuna bağlıdır. Değişik veri kaynaklarındaki tutarlılık sağlanır.

2.4.3. Birleştirme ve Temizleme

Uyumlandırma işleminde değişik veri kaynaklarından gelen verilerin birleştirilmesi veya fazlalıklarından temizlenmesi de gerekebilir.

2.4.4. Seçme

Seçme işleminde kuracağımız model için uygun verinin seçilmesi işlemidir. Veritabanlarında her ne kadar işlem hızları artmış olmasına rağmen büyük veritabanlarında birden çok verinin denenmesi oldukça maliyet ve zaman gerekmektedir. Bu sebeple verinin bütününü temsil edecek şekilde bir parça üzerinde işlemler yapılabilir. Ancak seçilecek parçanın verinin tamamını temsil etmesi açısından önemi oldukça büyüktür.

2.4.5. Dönüştürme

Verinin kullanılacak modele göre içeriğini koruyacak şekilde dönüştürülmesi işlemidir. Modele uygun şekilde dönüştürme işlemi yapılmalıdır. Burada verinin gösterilmesinde kullanılacak model ve algoritma önemli bir role sahiptir.

Yukarıda bahsedilen bilgiler dâhilinde veri ambarları ve veritabanları arasındaki farklar göze çarpmaktadır. Veri tabanları içerik olarak bütün detayları kapsamaktadır ancak veri ambarlar özet ve ilgili verileri tutmaktadır. Veri tabanı bir kısım veriler üzerinden yapılmaktayken veri ambarları daha fazla veri üzerinden işlemler yapmaktadır. Veri tabanlarında veriler iki boyutlu tutulurken veri ambarlarında çok boyutlu veri saklama imkânına sahiptir. Bu nedenle verinin analizi kolaylaşmaktadır. Veri tabanları sürekli gücellenmektedir ancak veri ambarları belirli periyotlar ile güncellenmektedir. Veri ambarları analizlerini uzun ve karmaşık süreçler ile yapmaktayken veri tabanlarında istenilen sorgulara anında ulaşılmaktadır. Veri tabanları güncel verilerden oluşurken veri ambarları tarihsel verilerden oluşmaktadır. Veri ambarlarında tekrarlanan kişi, yer tek bir alanda toplanmış ve düzenlenmiştir. Aynı olay ve veriler birbirine bağlanmıştır. Veri ambarındaki bilgiler zaman değişkeniyle ilişkilendirilmiştir, veriler değiştirilemez ve silinemez. Belirli bir döneme aittir, yeni veri giriş çıkışı yapılamaz [6].

Veri ambarı ilişkili verilerin sorgulanabildiği ve analizlerinin yapılabildiği bütünleşmiş bir bilgi deposudur. Veri ambarları farklı kaynaklardan gelen veriler üzerinde daha kolay ve etkili sorgulamaların yapılmasını sağlamaktadır. Veri ambarları geleceğe dönük tahminler yaparak sonuçlar çıkarmak ve işletmelerin yönetim stratejilerini belirlemek için kullanılan bir sistemdir.

3. VERİ MADENCİLİĞİ

3.1. Veri Madenciliğine Genel Bir Bakış

Bilgisayar sistemleri her geçen gün hem daha ucuzlamaktadır, hem de güçleri artmaktadır. İşlemciler gittikçe hızlanıyor, disklerin kapasiteleri artıyor. Artık bilgisayarlar daha büyük miktardaki veriyi saklayabiliyor ve daha kısa sürede işleyebiliyor. Bunun yanında bilgisayar ağlarındaki ilerleme ile bu veriye başka bilgisayarlardan da hızla ulaşabilmek mümkün olabilmektedir. Bilgisayarların ucuzlaması ile sayısal teknoloji daha yaygın olarak kullanılıyor. Veri doğrudan sayısal olarak toplanıyor ve saklanıyor. Bunun sonucu olarak da ayrıntılı ve doğru bilgiye ulaşabiliniyor[3].

Örneğin eskiden süper marketteki kasa basit bir toplama makinesinden ibaretti. Müşterinin o anda satın almış olduğu malların toplamını hesaplamak için kullanılırdı. Günümüzde ise kasa yerine kullanılan satış noktası terminalleri sayesinde bu hareketin bütün detayları saklanabiliyor. Saklanan bu binlerce malın ve binlerce müşterinin hareket bilgileri sayesinde her malın zaman içindeki hareketlerine ve eğer müşteriler bir müşteri numarası ile kodlanmışsa bir müşterinin zaman içindeki verilerine ulaşmak ve analiz etmek mümkün olabilmektedir. Bütün bunlar marketlerde kullanılan barkot, bilgisayar destekli veri toplama ve isleme cihazları sayesinde mümkün olmaktadır[7].

Yukarıdaki market örneğinde olduğu gibi ticari, tıp, askeri, iletişim vb. birçok alanda benzer teknolojilerin kullanılması ile veri hacminin yaklaşık olarak her yirmi ayda iki katına çıktığı tahmin edilmektedir [8].

Veri hacminin hangi boyutlara ulaşabileceği ve bunların islenmesinin ne kadar güç olduğu kolayca anlaşılabilmektedir. Süper market örneği incelendiğinde, veri analizi yaparak her mal için bir sonraki ayın satış tahminleri çıkarılabilir; müşteriler satın aldıkları mallara bağlı olarak gruplanabilir; yeni bir ürün için potansiyel müşteriler belirlenebilir; müşterilerin zaman içindeki hareketleri incelenerek onların davranışları ile ilgili tahminler yapılabilir. Binlerce malın ve müşterinin olabileceği düşünülürse bu analizin gözle ve elle yapılamayacağı,

otomatik olarak yapılmasının gerektiği ortaya çıkar. Veri madenciliği burada devreye girer; veri madenciliği büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların aranmasıdır [7].

Yani, veri madenciliği, büyük miktarlardaki verinin içinden geleceğin tahmin edilmesinde yardımcı olacak anlamlı ve yararlı bağlantı ve kuralların bilgisayar programlarının aracılığıyla aranması ve analizidir. Ayrıca Veri Madenciliği, çok büyük miktardaki verilerin içindeki ilişkileri inceleyerek aralarındaki bağlantıyı bulmaya yardımcı olan veri analizi tekniğidir [9].

Bir başka deyişle, veri madenciliği; verideki eğilimleri, ilişkileri ve profilleri belirlemek için veriyi sınıflandıran bir analitik araç ve bilgisayar yazılım paketidir. Spesifik veri madenciliği yazılımları; kümeleme, doğrusal regresyon, sinir ağları, Bayes ağları, görselleştirme ve ağaç tabanlı modeller gibi pek çok modeli içerir. Veri madenciliği uygulamalarında yıllar boyu istatistiksel yöntemler kullanılmıştır. Bununla birlikte, bugünün veri madenciliği teknolojisinde eski yöntemlerin tersine büyük veri kümelerindeki trend ve ilişkileri kısa zamanda saptayabilmek için yüksek hızlı bilgisayarlar kullanılmaktadır. Böylece veri madenciliği, gizli trendleri minimum çaba ve emekle ortaya çıkarmaktadır [10].

Frawley veri madenciliğini, "*Daha önceden bilinmeyen ve potansiyel olarak yararlı olma durumuna sahip verinin keşfedilmesi*" olarak tanımlamıştır. *Berry ve Linoff* bu kavrama "*Anlamlı kuralların ve örüntülerin bulunması için geniş veri yığınları üzerine yapılan keşif ve analiz işlemleri*" şeklinde bir açıklama getirirken *Sever ve Oğuz* çalışmalarında veri madenciliği hakkında "*Önceden bilinmeyen, veri içinde gizli, anlamlı ve yararlı örüntülerin büyük ölçekli veritabanlarından otomatik biçimde elde edilmesini sağlayan veri tabanlarında bilgi keşfi süreci içerisinde bir adımdır.*" tanımını kullanmışlardır.

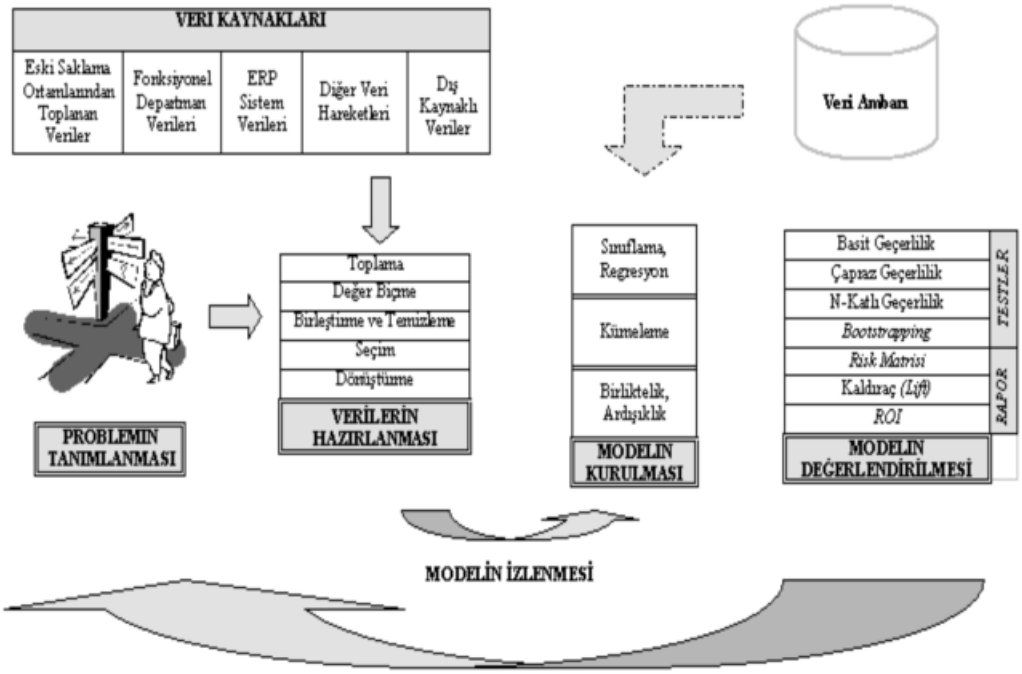
3.2 Veritabanlarında Bilgi Keşfi Süreci

Veri madenciliği, veri ambarlarında tutulan ve ilk bakışta çok net şekilde anlaşılamayan bilgilerin sırlarını ortaya çıkartmak, bir anlamda bilgiyi keşfetmektir. Veri madenciliği matematiksel, istatiksel ve desen tanıma (pattern

recognition) yöntemlerinden herhangi birini veya bir kaçını kullanarak büyük bir veri ambarı içerisindeki desenlerin, benzerliklerin ve korelâsyonların tespit edilmesi ve anlamlandırılması işlemidir.

Veritabanı sistemlerinin artan kullanımı ve hacimlerindeki olağanüstü artış, organizasyonları elde toplanan verilerden nasıl faydalanılabileceği problemi ile karşı karşıya bırakmıştır.

Geleneksel sorgu (query) veya raporlama araçlarının veri yığınları karşısında yetersiz kalması, *Veritabanlarında Bilgi Keşfi - VTBK* (Knowledge Discovery in Databases - KDD) adı altında, sürekli ve yeni arayışlara neden olmaktadır. Bu süreç içerisinde, modelin kurulması ve değerlendirilmesi aşamalarından meydana gelen veri madenciliği en önemli kesimi oluşturmaktadır. Bu önem, bir çok araştırmacı tarafından VTBK ile veri madenciliği terimlerinin eş anlamlı olarak da kullanılmasına neden olmaktadır [4].



Şekil.5. Veritabanlarında bilgi keşfi süreci [9]

VTBK sürecinde izlenmesi gereken temel aşamalar şunlardır:

Problemin tanımlanması, Verilerin hazırlanması,
Modelin kurulması ve değerlendirilmesi, Modelin kullanılması ve
Modelin izlenmesidir [9]

3.3. Veri Madenciliğinin Gelişimini Etkileyen Faktörler

Temel olarak veri madenciliğini 5 ana faktör etkiler [9]:

3.3.1. Veri

Veri madenciliğinin bu kadar gelişmesindeki en önemli faktördür. Son yirmi yılda sayısal verinin hızla artması, veri madenciliğindeki gelişmeleri hızlandırmıştır. Bu kadar fazla veriye bilgisayar ağları üzerinden erişilmektedir. Diğer taraftan bu verilerle uğraşan bilim adamları, mühendisler ve istatistikçilerin sayısı hala aynıdır. O yüzden, verileri analiz etme yöntemleri ve teknikleri geliştirilmektedir

3.3.2. Donanım

Veri madenciliği, sayısal ve istatistiksel olarak büyük veri kümeleri üzerinde yoğun işlemler yapmayı gerektirir. Gelişen bellek ve işlem hızı kapasitesi sayesinde, birkaç yıl önce madencilik yapılamayan veriler üzerinde çalışmayı mümkün hale getirmiştir.

3.3.3. Bilgisayar Ağları

Yeni nesil internet, çok yüksek hızları kullanmamızı sağlamaktadır. Böyle bir bilgisayar ağı ortamı oluştuktan sonra, dağıtık verileri analiz etmek ve farklı algoritmaları kullanmak mümkün olacaktır. Bundan 10 yıl önceki bilgisayar ağları teknolojisinde hayal edilemeyenler artık kullanılabiliyor. Buna bağlı olarak, veri madenciliğine uygun ağların tasarımı da yapılmaktadır.

3.3.4. Bilimsel Hesaplamalar

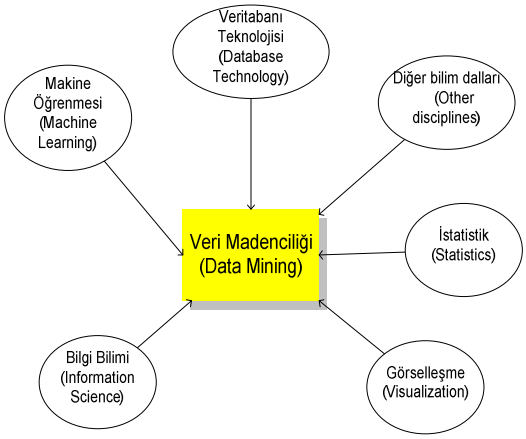
Günümüz bilim adamları ve mühendisleri, simülasyonu, bilimin üçüncü yolu olarak görmekteler. Veri madenciliği ve bilgi keşfi, teori, deney ve simülasyonu birbirine bağlamada önemli bir rol almaktadır.

3.3.5. Ticari Eğilimler

Günümüzde, isletmeler rekabet ortamında varlıklarını koruyabilmek için daha hızlı hareket etmeli, daha yüksek kalitede hizmet sunmalı, bütün bunları yaparken de minimum maliyeti ve en az insan gücünü göz önünde bulundurmalıdır. Bu tip hedef ve kısıtların yer aldığı iş dünyasında veri madenciliği, temel teknolojilerden biri haline gelmiştir. Çünkü veri madenciliği sayesinde müşterilerin ve müşteri faaliyetlerinin yarattığı fırsatlar daha kolay tespit edilebilmekte ve riskler daha açık görülebilmektedir.

3.4. Veri Madenciliği Uygulamaları ve Kullanım Alanları

Veri madenciliğinde araştırma çok farklı disiplinlerin içinde uygulanmaktadır. Veritabanını yöneten araştırmacılar veri madenciliğinin avantajını sorgu işleminden almaktadır. İlgi çeken alanlardan biri de sorgu işlemini genişletmek ve veri madenciliğini kolaylaştırmaktır. Veri depolama başka bir veri yönetim teknolojisidir ve bu ayrı veri kaynaklarını tamamlamakta, veriyi kolay bir şekilde organize edebilmektedir [11].



Şekil.6. Veri Madenciliğin Birçok Disiplinle Bileşimi [2]

Uygulama Alanları	Uygulama	Açıklama
Perakende-cilik	Benzerlik konumu Çapraz satış	Etkin ürünlerin benzerlikleri tespit etmek Müşteriler için birçok ürün bulunması, ürün satışları arasıdaki ilişkiyi tespit etmek
Banka Kredi kart hareketleri	Müşteri ilişki yönetimi (crm) Müşteri kaybı analizi	Müşteri değerlerinin tanımlanması, müşteriler arası benzerliklerin tespit etmek, gelirleri maksimum edici programlar geliştirmek Aldıkları hizmeti iptal etme riski olan müşterileri gösteren raporlar oluşturma.
Sigorta, Bankacılık, Telekomi-nikasyon	Sahtekârlık saptama ve yönetimi	Geçmiş veri kullanılarak sahtekârlık yapanlar için bir model oluşturma ve benzeri davranış gösterenleri belirleme
Hedef Pazar bulma	Pazar analizi	Benzer özellikler gösteren müşterilerin bulunması: Benzer gelir grupları, ilgi alanları, harcama alışkanlıkları. Benzer müşterileri otomatik olarak gruplayarak pazar dilimlerini tanımlayın ve bu dilimleri pazarlama kampanyalarını hazırlarken trend analizi için kullanın
Finans planlaması ve bilânço değerlendir mesi	Risk analizi	Nakit para akışının incelenmesi ve kestirimi, talep incelemesi ve kestirimi, zaman serileri incelemesi

Şekil.7. Veri Madenciliğinin Genel Kullanım Alanları

Veri madenciliğinin asıl amacı, veri yığınlarından anlamlı bilgiler elde etmek ve bunu eyleme dönüştürecek kararlar için kullanmaktır. Örnek birkaç kullanım alanı aşağıdadır:

- İşletme alanındaki uygulamalar:

Müşterilerin rakiplerine gitmelerini önlemek için bir şirket Veri Madenciliğine başvurabilir. Burada Veri Madenciliğinin amacı müşterilerin özelliklerini elde etmek ve rakip şirketlere gidebilecek müşterileri tespit etmek. Sonra da VM sonuçlarından faydalanarak onları kaybetmemek için strateji geliştirilebilir. Ürün ve hizmetlerin tercih edilme sebeplerini ve hangi özelliklerin müşterileri ne kadar etkilediğini belirleme. Banka müşterilerinin kredi durumları ve ödemeleri incelenerek aralarında riskli olan müşterilerin tespit edilmesi ve aynı risk grubuna düşebilecek diğer müşterilerin önceden tahmin edilebilmesi.

- Eğitim alanındaki uygulamalar:

Öğrencileri performanslarına göre branşlara ayırma ve kişi-branş uyumunun arttırılması. Öğrencilerin performans ve memnuniyetlerinin arttırılması.

- Tıp alanındaki uygulamalar:

Bir ilacın hangi yas grup hastalarında nasıl etki yaratacağı konusunda tahminde bulunabilme. Yeni bulunan kanser tedavi yöntemi için en uygun adayı belirleme.

- Spor alanındaki uygulamalar:

Bir basketbol takımına alınacak oyuncunun gelecekteki performansı ve takıma sağlayacağı yararları tahmin edilebilme.

- Kütüphanecilik alanındaki uygulamalar:

Bir müşterinin aldığı kitabı ne zaman getireceği ve bir sonraki gelişinde hangi kitabı seçeceği konusunda tahminler yapılabilme.

- Turizm alanındaki uygulamalar:

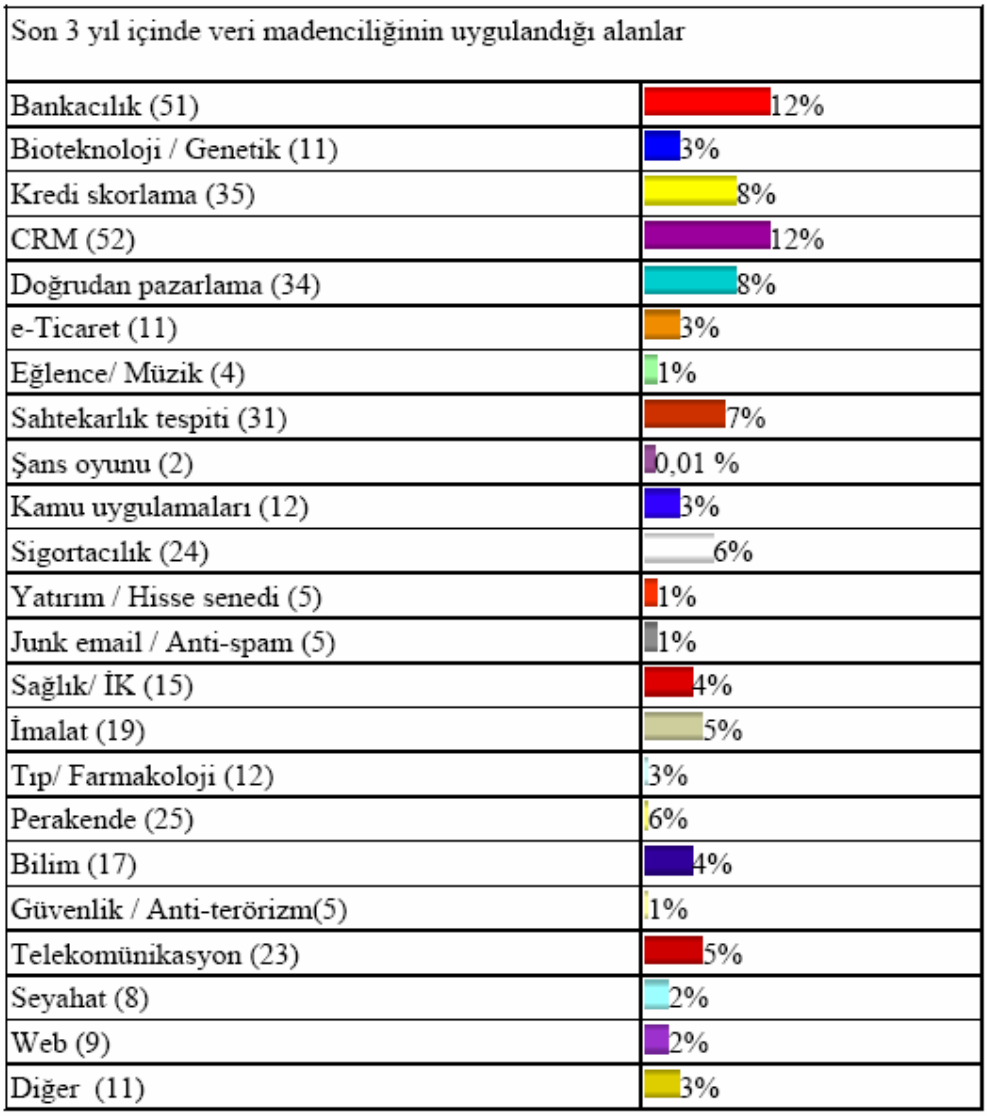
Bir bölgeye turistlerin niçin geldiğini tespit ederek reklâm kampanyası düzenlemek ve bir sonraki sezonda turist sayısını arttırmak.

- Web alanındaki uygulamalar:

Kullanıcıların profilini belirleyip onlara uygun ürünlerin reklâm kampanyalarını sayfalara koyma. Geçmişteki ve su anki veriler analiz edilerek geleceğe yönelik tahminlerde bulunulabilir. Bu durum özellikle karlılık, ciro, pazar payı, hava durumu gibi analizlerde çok rahat kullanılabilir.

Şekil 8'de 2003 yılında veri madenciliğinin sektörler bazında kullanımına ilişkin bir araştırmanın sonuçları yer almaktadır. Bu çizelgede araştırmaya katılan

toplam 421 şirketin 51 adedinin bankacılık alanında veri madenciliğini kullandığı görülmektedir [10].



Şekil.8. Veri madenciliği uygulama alanları[10]

3.5. Veri Madenciliği Uygulamalarında Karşılaşılan Problemler

VM büyük hacimli gerçek dünya verileriyle uğraştığı için, bu büyük hacimli veriler VM'de büyük sorunlar oluşturur. Bundan dolayı mesela küçük veri setleriyle ve yapay hazırlanmış verilerle doğru çalışan sistemler büyük hacimli, eksik, gürültülü, NULL değerli, artık, dinamik verilerle yanlış çalışabilir. Bundan dolayı bu sorunların asılması gerekmektedir. Veri madenciliği girdi olarak kullanılacak ham

veriyi veritabanlarından alır. Bu da veritabanlarının dinamik, eksiksiz, geniş ve net veri içermemesi durumunda sorunlar doğurur [7].

Diğer sorunlar da verinin konu ile uyumsuzluğundan doğabilir. Sınıflandırmak gerekirse başlıca sorunlar aşağıdaki gibidir:

1. Sınırlı bilgi: Veritabanları genel olarak veri madenciliği dışındaki amaçlar için tasarlanmışlardır. Bu yüzden, öğrenme görevini kolaylaştıracak bazı özellikler bulunmayabilir.

2. Gürültü ve kayıp değerler: Veri girişi veya veri toplanması esnasında oluşan sistem dışı hatalara gürültü denir. Veri toplanması esnasında oluşan hatalara ölçümden kaynaklanan hatalar da dâhil olmaktadır. Bu hataların sonucu olarak VM'de birçok niteliğin değeri yanlış olabilir.

3. Belirsizlik: Yanlışlıkların şiddeti ve verideki gürültünün derecesi ile ilgilidir. Veri tahmini, bir kesif sisteminde önemli bir husustur.

4. Ebat, güncellemeler ve konu dışı sahalar: Veri tabanlarındaki bilgiler, veri eklendikçe ya da silindikçe değişebilir. Veri madenciliği perspektifinden bakıldığında, kuralların hala aynı kalıp kalmadığı ve istikrarlılığı problemi ortaya çıkar. Öğrenme sistemi, kimi verilerin zamanla değişmesine ve kesif sisteminin verinin zamansızlığına karsın zaman duyarlı olmalıdır.

5. Artık veri: Artık veri, problemde istenilen sonucu elde etmek için kullanılan örneklem kümesindeki gereksiz niteliklerdir. Artık nitelikleri elemek için geliştirilmiş algoritmalar, özellik seçimi olarak adlandırılır. Özellik seçimi arama uzayını küçültür ve sınıflama işleminin kalitesini de artırır.

3.6. Veri Madenciliği Teknikleri

Veri madenciliği teknikleri işlevlerine göre 3 temel grupta toplanır [10]:

- Sınıflama,
- Kümeleme,
- Birliktelik kuralları ve sıralı örüntüler.

3.6.1. Sınıflama

Sınıflama, verinin önceden belirlenen çıktılara uygun olarak ayrıştırılmasını sağlayan bir tekniktir. Çıktılar, önceden bilindiği için sınıflama, veri kümesini denetimli olarak öğrenir [10].

Örneğin; A finans hizmetleri şirketi; müşterilerinin yeni bir yatırım fırsatıyla ilgilenip ilgilenmediğini öğrenmek istemektedir. Daha önceden benzer bir ürün satmıştır ve geçmiş veriler hangi müşterilerin önceki teklife cevap verdiğini göstermektedir. Amaç; bu teklife cevap veren müşterilerin özelliklerini belirlemek ve böylece pazarlama ve satış çalışmalarını daha etkin yürütmektir.

Müşteri kayıtlarında müşterinin önceki teklife cevap verip vermediğini gösteren "evet"/ "hayır" seklinde bir alan bulunur. Bu alan "hedef " ya da "bağımlı" değişken olarak adlandırılır. Amaç, müşterilerin diğer niteliklerinin (gelir düzeyi, is türü, yas, medeni durum, kaç yıldır müşteri olduğu, satın aldığı diğer ürün ve yatırım türleri) hedef değişken üzerindeki etkilerini analiz etmektir. Analizde yer alan diğer nitelikler "bağımsız" ya da "tahminci" değişken adını alır.

Temel sınıflama algoritmaları aşağıdadır:

- Diskriminant analizi,
- Naive Bayes,
- Karar ağaçları,
- Sinir ağları,
- Kaba kümeler,
- Genetik algoritma,
- Regresyon analizi.

3.6.1.1. Diskriminant Analizi

Diskriminant analizi, bir dizi gözlemi önceden tanımlanmış sınıflara atayan bir tekniktir. Model, ait oldukları sınıf bilinen gözlem kümesi üzerine kurulur. Bu küme, öğrenme kümesi olarak da adlandırılır. Öğrenme kümesine dayalı olarak,

diskriminant fonksiyonu olarak bilinen doğrusal fonksiyonların bir kümesi oluşturulur. Diskriminant fonksiyonu, yeni gözlemlerin ait olduğu sınıfı belirlemek için kullanılır. Yeni bir gözlem söz konusu olduğu için tüm diskriminant fonksiyonları hesaplanır ve yeni gözlem diskriminant fonksiyonunun değerinin en yüksek olduğu sınıfa atanır [15].

3.6.1.2. Naive Bayes

Naive Bayes, hedef değişkenle bağımsız değişkenler arasındaki ilişkiyi analiz eden tahminci ve tanımlayıcı bir sınıflama algoritmasıdır[16].

Naive Bayes, sürekli veri ile çalışmaz. Bu nedenle sürekli değerleri içeren bağımlı ya da bağımsız değişkenler kategorik hale getirilmelidir. Örneğin; bağımsız değişkenlerden biri yas ise, sürekli değerler "<20" "21–30", "31–40" gibi yas aralıklarına dönüştürülmelidir.

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

3.6.1.3. Karar Ağaçları

Karar ağaçları, yaygın olarak kullanılan sınıflama algoritmalarından birisidir. Karar ağacı yapılarında, her düğüm bir nitelik üzerinde gerçekleştirilen testi, her dal bu testin çıktısını, her yaprak düğüm ise sınıfları temsil eder. En üstteki düğüm, kök düğüm olarak adlandırılır. Karar ağaçları, kök düğümden yaprak düğüme doğru çalışır [17].

En yaygın kullanılan karar ağacı algoritmaları;

• CHAID (Chi-Squared Automatic Interaction Detector, Kass 1980),
• C&RT (Classification and Regression Trees, Breiman ve Friedman, 1984),
• ID3 (Induction of Decision Trees, Quinlan, 1986),
• C4.5 (Quinlan, 1993).

3.6.1.4. Sinir Ağları

Sinir ağları, tanımlayıcı ve tahmin edici veri madenciliği algoritmalarındandır. İnsan beyninin fizyolojisini taklit ederler. Komplike ve belirsiz veriden bilgi üretirler. Keşfettikleri örüntü ve trendler, insanlar ya da bilgisayarlarca kolay keşfedilemez. Bu tür karmaşık problemlerde birbirleriyle etkileşimli yüzlerce değişken bulunur[18].

Bu teknik, veritabanındaki örüntüleri, sınıflandırma ve tahminde kullanılmak üzere genelleştirir. Sinir ağları algoritmaları sadece sayısal veriler üzerinde çalışırlar.

3.6.1.5. Kaba Kümeler

Kaba küme teorisi 1970'li yıllarda Pawlak tarafından geliştirilmiştir. Kaba küme teorisinde bir yaklaştırma uzayı ve bir kümenin alt ve üst yaklaştırmaları vardır. Yaklaştırma uzayı, ilgilenilen alanı ayrı kategorilerde sınıflandırır. Alt yakınlaştırma belirli bir altkümeye ait olduğu kesin olarak bilinen nesnelerin tanımıdır. Üst yakınlaştırma ise alt kümeye ait olması olası nesnelerin tanımıdır. Alt ve üst sınırlar arasında tanımlanan herhangi bir nesne ise "kaba küme" olarak adlandırılır [19].

3.6.1.6. Genetik Algoritma

Algoritma ilk olarak popülâsyon adı verilen bir çözüm kümesi (öğrenme veri kümesi) ile başlatılır. Bir popülâsyondan alınan sonuçlar bir öncekinden daha iyi olacağı beklenen yeni bir popülâsyon oluşturmak için kullanılır. Evrim süreci (yeni popülâsyonlar yapma iterasyonu) tamamlandığında bağımlılık kuralları veya sınıf modelleri ortaya konmuş olur [20].

3.6.1.7. Regresyon Analizi

Regresyon analizi, bir ya da daha fazla bağımsız değişken ile hedef değişken arasındaki ilişkiyi matematiksel olarak modelleyen bir yöntemdir. Veri madenciliğinde yaygın olarak kullanılan regresyon modellerinden doğrusal regresyonda tahmin edilecek olan hedef değişken sürekli değer alırken; lojistik regresyonda hedef değişken kesikli bir değer almaktadır. Doğrusal regresyonda hedef değişkenin değeri; lojistik regresyonda ise hedef değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilmektedir [21].

3.6.2. Kümeleme

Kümeleme, verideki benzer kayıtların gruplandırılmasını sağlayan bir tekniktir. Kümelemede, genellikle k-ortalamalar algoritması ya da Kohonen şebekesi gibi istatistiksel yöntemler kullanılmaktadır. Hangi yöntem kullanılırsa kullanılsın süreç aynı şekilde isler. Her kayıt var olan kümelerle karşılaştırılır. Bir kayıt kendisine en yakın kümeye atanır ve bu kümeyi tanımlayan değeri değiştirir. Optimum çözüm bulununcaya kadar kayıtlar yeniden atanır ve küme merkezleri ayarlanır [21].

3.6.3. Birliktelik Kuralları ve Ardışık Örüntüler

Birliktelik analizi, bir veri kümesindeki kayıtlar arasındaki bağlantıları arayan denetimsiz veri madenciliği şeklidir. Birliktelik analizi çoğu zaman perakende sektöründe süpermarket müşterilerinin satın alma davranışlarını ortaya koymak için kullanıldığından "pazar sepeti analizi" olarak da adlandırılır [21].

Birliktelik kurallarına ait bir örnek: "Düşük yağlı peynir ve yağsız süt alan müşteriler %85 olasılıkla diyet süt alırlar."

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

• “Çadır alan müşterilerin %10’u bir ay içerisinde sırt çantası almaktadır.”
• “A hissesi %15 artarsa üç gün içinde B hissesi %60 olasılıkla artacaktır.”

3.7. Veri Madenciliği Süreci

Ne kadar etkin olursa olsun hiç bir veri madenciliği algoritmasının, üzerinde inceleme yapılan isin ve verilerin özelliklerinin bilinmemesi durumunda fayda sağlaması mümkün değildir. Bu nedenle yukarıda tanımlanan tüm aşamalardan önce, iş ve veri özelliklerinin öğrenilmesi-anlaşılması basarının ilk şartı olacaktır. Başarılı bir veri madenciliği projesinde izlenmesi gereken adımlar aşağıdadır [25]:

1. Problemin tanımlanması,
2. Verilerin hazırlanması,
3. Modelin kurulması ve değerlendirilmesi,
4. Modelin kullanılması,
5. Modelin izlenmesi.

3.7.1. Problemin Tanımlanması

Veri madenciliği çalışmalarında başarılı olmanın en önemli şartı, projenin hangi işletme amacı için yapılacağının ve elde edilecek sonuçların basarı düzeylerinin nasıl ölçüleceğinin tanımlanmasıdır. Ayrıca yanlış tahminlerde katlanılacak olan maliyetlere ve doğru tahminlerde kazanılacak faydalara ilişkin tahminlere de bu aşamada yer verilmelidir. Bu aşamada mevcut is probleminin nasıl bir sonuç üretilmesi durumunda çözüleceğinin, üretilecek olan sonucun fayda-maliyet analizinin başka bir ifadeyle üretilen bilginin isletme için değerinin doğru analiz edilmesi gerekmektedir [10].

Analistin, işletmede üretilen sayısal verilerin boyutlarını, proje için yeterlilik düzeyini ve is süreçlerini iyi analiz etmesi gerekmektedir.

3.7.2. Verilerin Hazırlanması (Veri Önişleme)

Veri madenciliğinin en önemli aşamalarından biri olan verinin hazırlanması aşaması, analistin toplam zaman ve enerjisinin %50-%85’ini harcamasına neden

olmaktadır. Veri kalitesi, veri madenciliğinde anahtar bir konudur. Veri madenciliğinde güvenilirliğin artırılması için, veri ön isleme yapılmalıdır. Aksi halde hatalı girdi verileri kullanıcıyı hatalı çıktıya götürecektir. Veri ön isleme, çoğu durumlarda yarı otomatik olan ve yukarıda da belirtildiği gibi zaman isteyen bir veri madenciliği aşamasıdır. Verilerin sayısındaki artış ve buna bağlı olarak çok büyük sayıda verilerin ön işlemden geçirilmesinin gerekliliği, otomatik veri ön isleme için etkin teknikleri önemli hale getirmiştir [23].

Veri ön isleme teknikleri şu şekilde sıralanabilir:

1. Veri Temizleme
2. Veri Birleştirme
3. Veri Dönüştürme
4. Veri İndirgeme

Veri önişleme tekniklerinden kısaca bahsetmek gerekirse;

3.7.2.1. Veri Temizleme

Veri temizleme, eksik verilerin tamamlanması, aykırı değerlerin tespit edilmesi amacıyla gürültünün düzeltilmesi ve verilerdeki tutarsızlıkların giderilmesi gibi işlemleri gerektirmektedir.

3.7.2.2. Veri Birleştirme

Veri madenciliğinde bazen farklı veri tabanlarındaki verilerin birleştirilmesi gerekebilir. Farklı veri tabanlarındaki verilerin tek bir veri tabanında birleştirilmesiyle sema birleştirme hataları oluşur. Örneğin, bir veri tabanında girişler "tüketici-ID" seklinde yapılmışken, bir diğerinde "tüketici-numarası" seklinde olabilir. Bu tip sema birleştirme hatalarından kaçınmak için meta veriler kullanılır.

3.7.2.3. Veri Dönüştürme

Veri dönüştürme ile veriler, veri madenciliği için uygun formlara dönüştürülürler. Veri dönüştürme; düzeltme, birleştirme, genelleştirme ve

normalleştirme gibi değişik işlemlerden biri veya bir kaçını içerebilir. Veri normalleştirme en sık kullanılan veri dönüştürme işlemlerinden birisidir.

3.7.2.4. Veri İndirgeme

Veri indirgeme teknikleri, daha küçük hacimli olarak ve veri kümesinin indirgenmiş bir örneğinin elde edilmesi amacıyla uygulanır. Bu sayede elde edilen indirgenmiş veri kümesine veri madenciliği teknikleri uygulanarak daha etkin sonuçlar elde edilebilir.

Veri indirgeme yöntemleri ise aşağıdaki biçimde özetlenebilir:

1. Veri Birleştirme veya Veri Küpü
2. Boyut indirgeme
3. Veri Sıkıştırma
4. Kesikli hale getirme

3.8. Modelin Kurulması ve Değerlendirilmesi

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

Model kuruluş süreci denetimli ve denetimsiz öğrenimin kullanıldığı modellere göre farklılık göstermektedir [24].

Örnekten öğrenme olarak da isimlendirilen denetimli öğrenimde, bir denetçi tarafından ilgili sınıflar önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunmasıdır.

Öğrenme süreci tamamlandığında, tanımlanan kurallar verilen yeni örneklere uygulanır ve yeni örneklerin hangi sınıfa ait olduğu kurulan model tarafından belirlenir. Denetimsiz öğrenmede, kümeleme analizinde olduğu gibi ilgili örneklerin

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

Denetimli öğrenimde seçilen algoritmaya uygun olarak ilgili veriler hazırlandıktan sonra, ilk aşamada verinin bir kısmı modelin öğrenimi, diğer kısmı ise modelin geçerliliğinin test edilmesi için ayrılır. Modelin öğrenimi, öğrenim kümesi kullanılarak gerçekleştirildikten sonra, test kümesi ile modelin doğruluk derecesi belirlenir [10].

Bir modelin doğruluğunun test edilmesinde kullanılan en basit yöntem basit geçerlilik testidir. Bu yöntemde tipik olarak verilerin %5 ile %33 arasındaki bir kısmı test verileri olarak ayrılır ve kalan kısım üzerinde modelin öğrenimi gerçekleştirildikten sonra, bu veriler üzerinde test işlemi yapılır. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının, tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile ise doğruluk oranı hesaplanır [9].

Değerlendirme kriterlerinden en önemlisi, modelin anlaşılabilirliğidir. Bazı uygulamalarda doğruluk oranlarındaki küçük artışlar çok önemli olsa da, birçok işletme uygulamasında ilgili kararın niçin verildiğinin yorumlanabilmesi çok daha büyük önem taşıyabilir. Çok ender olarak yorumlanamayacak kadar karmaşıklaşsalar da, genel olarak karar ağacı ve kural temelli sistemler model tahmininin altında yatan nedenleri çok iyi ortaya koyabilmektedir. Kurulan modelin doğruluk derecesi ne denli yüksek olursa olsun, gerçek dünyayı tam anlamı ile modellediğini garanti edebilmek mümkün değildir. Yapılan testler sonucunda geçerli bir modelin doğru olmamasındaki baslıca nedenler, model kuruluşunda kabul edilen varsayımlar ve modelde kullanılan verilerin doğru olmamasıdır.

3.8.1. Modelin Kullanılması

Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama olabileceği gibi, bir başka uygulamanın alt parçası olarak kullanılabilir. Kurulan modeller risk analizi, kredi değerlendirme, dolandırıcılık tespiti gibi isletme uygulamalarında doğrudan kullanılabileceği gibi, promosyon planlaması simülasyonuna entegre

edilebilir veya tahmini envanter düzeyleri yeniden sipariş noktasının altına düştüğünde, otomatik olarak sipariş verilmesini sağlayacak bir uygulamanın içine gömülebilir [25].

3.8.2. Modelin İzlenmesi

Zaman içerisinde bütün sistemlerin özelliklerinde ve dolayısıyla ürettikleri verilerde ortaya çıkan değişiklikler, kurulan modellerin sürekli olarak izlenmesini ve gerekiyorsa yeniden düzenlenmesini gerektirecektir. Tahmin edilen ve gözlenen değişkenler arasındaki farklılığı gösteren grafikler model sonuçlarının izlenmesinde kullanılan yararlı bir yöntemdir [25].

3.8.3. Problemin Tanımlanması

İşletmelerde maksimum verimlilik sağlanması, ürün kalitesinin artırılması ve çalışma zaman kayıplarının önlenmesini sağlamak için atılması gereken ilk adım işletmelerdeki süreçlerin gerçek zamanlı olarak sürekli izlenmesi, ikinci adım ise bu süreçlerin izlenmesi ile elde edilen verilerin, veri madenciliği gibi tahmin ve sınıflama kabiliyetine sahip bir araç ile analiz edilmesidir. Bu bize üretimi durdurmadan, belirli aralıklarla isletme duruşlarına gidilerek makine ve ekipmanlardan verileri alarak, oluşturulan veri madenciliği sistemi ile ortaya çıkabilecek olan hataların önceden tespit edilmesini sağlayacaktır.

4. MARKET SEPET ANALİZİ ve BİRLİKTELİK KURALLARI

4.1. Market Sepet Analizi

Geçmiş tarihli hareketleri çözümlemek, karar destek sistemlerinde verilen kararın kalitesini artırmak için izlenen bir yaklaşımdır. 90'lı yılların başına değin teknik yetersizlikten dolayı, kurumlara veya müşterilere satış yapıldığı anda değil, belirli bir zaman aralığında (günlük, haftalık, aylık, yıllık) gerçekleşen satış hareketlerinin tamamına ilişkin genel veriler elektronik ortamda tutulmaktaydı. Barkod uygulamalarındaki gelişme ile, bir harekete ait verilerin satış hareketi oluştuğu anda toplanması ve elektronik ortama aktarılması olanaklı hale gelmiştir. Genellikle süpermarketlerin satış noktalarında bu tür veriler toplandığından, toplanan bu veriye market sepeti verisi adı verilmiştir. Market sepeti verisinde yer alan bir kayıtta, tekil olan hareket numarası, hareket tarihi ve satın alınan ürünlere ilişkin ürün kodu, miktarı, fiyatı gibi bilgiler yer almaktadır [2].

*Market sepet analizi*nde (market basket analysis) amaç, satışlar arasındaki ilişkileri bulmak ve buna bağlı kuralları çıkarmaktır. Bu ilişkilerin bilinmesi, şirketin kârını arttırmak için kullanılabilir. Eğer X ürününü alanların Y ürününü de çok yüksek olasılıkla aldıkları biliniyorsa ve eğer bir müşteri X ürününü alıyor ama Y ürününü almıyorsa, o potansiyel bir Y müşterisidir denilebilir [4].

Buna benzer veri analizleri yaparak her ürün için bir sonraki ayın satış tahminleri çıkarılabilir, birlikte satın alınan ürünler için promosyon uygulaması ve reyon dizilişleri yapılabilir, müşteriler satın aldıkları ürünlere göre gruplandırılabilir, yeni bir ürün için potansiyel müşteriler belirlenebilir [3].

4.2. Birliktelik Kuralı

Veri madenciliğinde kullanılan ilk tekniklerden birisi de birliktelik kurallarıdır. [26] Birliktelik kuralı, geçmiş verilerin analiz edilerek bu veriler

içindeki birliktelik davranışlarının tespiti ile geleceğe yönelik çalışmalar yapılmasını destekleyen bir yaklaşımdır. Birliktelik kuralı madenciliğinin uygulamasına pazar sepeti analizi örnek verilebilir [27].

Birliktelik kuralındaki amaç; alışveriş esnasında müşterilerin satın aldıkları ürünler arasındaki birliktelik ilişkisini bulmak, bu ilişki verisi doğrultusunda müşterilerin satın alma alışkanlıklarını tespit etmektir. Satıcılar, keşfedilen bu birliktelik bağıntıları ve alışkanlıklar sayesi ile etkili ve kazançlı pazarlama ve satış imkânına sahip olmaktadırlar. Örneğin, bir marketten müşterilerin süt ve peynir satın alımlarının % 70'inde bu ürünler ile birlikte yoğurt da satın alınmıştır. Bu tür birliktelik örüntüsünün tespit edilebilmesi için, örüntü içinde yer alan ürünlerin birden çok satın alma hareketinde birlikte yer alması gerekir.
Milyonlarca veri üzerinde veri madenciliği teknikleri uygulandığında, birliktelik sorgusu için kullanılan algoritmalar hızlı olmalıdır [28].

Birliktelik kurallarındaki bir nesnenin ve bir işlemin tanımı uygulamaya bağlıdır. Market sepeti analizinde; nesneler, müşterilerin aldığı ürünler ve işlem, beraber alınan bütün nesnelerin kümesidir. Birliktelik kurallarında sıklıkla kullanılan birkaç önemli terim vardır. Bunlar; kuralın sol tarafını ifade eden önce (antecedent), kuralın sağ tarafını ifade eden sonuç (consequent), destek değeri, güven değeri, min_destek olarak gösterilen minimum destek değeri, min_güven olarak gösterilen minimum güven değeri, nesneküme, yaygın nesne kümesive aday nesnekümesidir [29].

Birliktelik kuralı madenciliği iki aşamalıdır:

- **Tüm yaygın nesne kümelerinin bulunması**:
  Her nesnekümesinin yaygın nesne kümesi olarak yer alabilmesi için, her nesnenin destek değerinin önceden tanımlanmış olan min_destek değerinden büyük olması gerekir.
- **Yaygın nesne kümelerinden güçlü birliktelik kurallarının elde edilmesi**:
  Bu kurallar min_destek ve min_güven durumunu sağlamalıdır.

Birliktelik kuralı algoritmalarının performansını belirleyen adım birinci adımdır. Yaygın nesnekümeleri belirlendikten sonra, birliktelik kurallarının bulunması sıradan bir adımdır.

Minimum güven ve destek değerlerini sağlayan birliktelik kuralları çıkarım problemi iki adıma bölünmüştür [30]:

1. Yaygın geçen nesnekümeler bulunur: Kullanıcı tarafından belirlenmiş olan minimum destek eşik değerini sağlayan nesnekümelere yaygın nesneküme adı verilmektedir. Bu adımda yaygın nesne kümeleri bulan etkili yöntemler kullanılmalıdır.
2. Yaygın nesne kümelerden güçlü birliktelik kuralları oluşturulur: yaygın nesnekümeleri kullanarak minimum güven eşik değerini sağlayan birliktelik kurallarının bulunmasıdır. Bu adımdaki işlem oldukça basittir. Minimum güven eşik değerine göre taranarak bulunan birliktelik kuralları kullanıcının ilgilendiği ve potansiyel olarak önemli bilgiyi içeren kurallardır. Birliktelik kuralı algoritmasının performansını belirleyen adım birinci adımdır. Yaygın nesnekümeleri belirlendikten sonra, birliktelik kurallarının bulunması kolay bir adımdır [2].

Sepet analizinin başarılı olduğu noktalar;

- Kolay ve anlaşılır sonuçlar üretir,
- Değişik boyutlardaki veriler üzerinde çalışabilir,

Her ne kadar kayıtların sayısı ve kombinasyon seçimine göre işlem adedi artsa da sepet analizi için her adımda gerekli olan hesaplamalar diğer yöntemlere göre (genetik algoritmalar, yapay sinir ağları vb.) çok daha basittir.

Sepet analizinin başarısız olduğu noktalar;

- Sorunun boyutu büyüdükçe, gerekli hesaplamalar üstel olarak artmaktadır,
- Kayıtlarda çok az rastlanan ürünleri yok sayar. Sepet çözümlemesi yönteminin en doğru sonucu, tüm ürünlerin kayıtlar içinde yaklaşık aynı

frekansta göründüğü durumlarda üretmektedir,

- Destek ve güven eşik değerleri üretilen kural sayısında sınırlama getirirler fakat eşik değerlerinin çok düşük belirlendiği durumda kullanıcı gerçekten ilgilendiği kuralları kaybetme tehlikesi ile karşı karşıya kalır.
- Birliktelik kurallarının keşfi katalog tasarımı, müşterilerin satın alma alışkanlıklarının belirlenmesi ve sınıflandırılması, mağaza ürün yerleşim planı gibi birçok uygulama alanında kullanılabilir. Sepet analizi yöntemi, birliktelik kurallarının uygulama alanındaki türlerinden birisidir.

4.2.2. Birliktelik Kuralları Çeşitleri

Birliktelik kurallarının birçok türü vardır. Birliktelik kuralları aşağıdaki kriterleri taşıyan çok değişik yollar ile sınıflandırılabilir.

Kuralda kullanılan değerlerin tiplerine göre: Eğer bir kural nesnelerin varlığı ve yokluğu arasındaki birliktelikler ile ilgili ise, bu duruma mantıksal birliktelik kuralıdır. Bu kurallar sepet analizinden elde edilirler. Kurallar nicel nesneler ya da özellikler arasındaki birliktelikleri tanımlıyor ise nicel birliktelik kuralıdır. Bu kurallarda, nesneler için nicel değerler ya da özellikler aralıklara bölünmüştür.

Kuralın içerdiği verinin boyutlarına göre: Bir birliktelik kuralındaki özellikler (attribute) ya da nesneler sadece bir boyutu temsil ediyorlarsa, o zaman kurala tek boyutlu birliktelik kuralıdır denir.

Birliktelik kurallarının çeşitli boyutlarına göre: Birliktelik kuralları çözümlemesi, korelâsyon çözümlemesinin genişletilmişi olabilir. Aynı zamanda,"maxpattern" ve "frequent closed itemset" çözümlemelerinin genişletilmişi de olabilir. Bu iki yöntem, çözümleme sırasında oluşturulan yaygın nesnekümelerin sayısını azaltmak için kullanılmaktadır[29].

4.3. Birliktelik Kurallarının Belirlenmesinde Kullanılan Temel Algoritmalar

Yaygın nesnekümelerini (large itemsets) belirlerken birliktelik kuralları oluşturmak için kullanılan algoritmalar, sıralı (sequential) ve paralel olarak sınıflandırılabilir. Birçok durumda, nesne adına bağlı olarak nesnekümelerinin sözlük (lexicographic) sıralı olarak tanımlandığı ve depolandığı varsayılır. Bu sıralama, nesnekümelerinin üretilmesi ve sayılması sırasında kolaylık sağlayan bir tarz oluşturmaktadır ve sıralı algoritmalarda rastlanan olağan bir yaklaşımdır. Diğer yandan, paralel algoritmalar yaygın nesnekümelerinin bulunması işleminin paralelleştirilmesi üzerinde odaklanır [4].

4.3.1. Sıralı Algoritmalar

4.3.1.1. AIS Algoritması

AIS algoritması, Agrawal, Imielinski ve Swami tarafından 1993 yılında veritabanındaki tüm yaygın nesne kümelerini oluşturmak için geliştirilmiş ve yayınlanmış ilk algoritmadır. Karar destek sorgulamaları yapmak için veri tabanlarının işlevlerini artırmaya odaklanmıştır. Bu algoritmanın hedefi nitelikli kuralları bulmaktır. Bu teknik, sonuçta sadece bir nesne ile kısıtlanmıştır. Birliktelik kuralları biçiminde $X \Rightarrow I_j \mid$ biçiminde tanımlanır. Burada X nesneler kümesi, , I alanındaki tek bir nesne ve de kuralın güven değeridir.

AIS algoritması, veritabanının üzerinden çoklu geçişler yapar. Her geçiş esnasında, veritabanını tarar. İlk geçişinde tek nesnelerin desteğini sayar ve veri tabanında bunların hangi sıklıkta olduğunu belirler. Her bir geçişin yaygın nesne kümeleri aday nesne kümeleri üretmek için genişletilir. Bir tarama yapıldıktan sonra önceki taramadaki yaygın nesnekümeleri ile taranan nesnelerin asındaki ortak nesnekümeleri belirlenir. Bu ortak nesnekümeleri yeni aday nesnekümeleri üetmek için veritabanındaki diğer nesneler ile genişletilir. Yaygın bir nesne kümesi olan *l*, yaygın ve sözlük sıralı *l*'deki her bir nesneden sonra gelen nesneler ile genişletilir. Bu işlemleri verimlice yapmak için AIS algoritması tahmin edici araçlar ve budama tekniği kullanılır. Bu teknikler aday gruplarını, gereksiz nesneleri aday listesinden

atarak belirlerler. Sonra her bir adayın grubu hesaplanır. Minimum destekten büyük veya minimum desteğe eşit olan aday grupları yaygın nesnekümeleri olarak seçilir. Yaygın nesne kümeleri sıradaki geçişte aday üretmek için genişletilirler. Bu işlemler yaygın nesnekümeleri bulunamayana kadar devam eder [5].

4.3.1.2. SETM Algoritması

SETM algoritması, 1995 yılında Houtsmal tarafından önerilmiş olup yaygın nesne kümelerinin hesaplanmasında SQL kullanılmasını baz almaktadır. Bu algoritmada yaygın nesne kümelerinin her bir üyesi, $\overline{L}_k$, TID birincil anahtar olmak üzere <TID, nesnekümesi> biçimindedir. Benzer şekilde aday kümelerinin her bir üyesi, $\overline{C}_k$ da <TID, nesnekümesi> biçimindedir.

AIS algoritmasına benzer olarak, SETM algoritması da veritabanı üzerinden çoklu geçişler yapar. İlk geçişte, ayrı her bir nesnenin destek sayısını sayar ve veritabanında hangilerinin büyük veya sık olduğunu bulur. Daha sonra, bir önceki geçişte oluşan yaygın nesne kümelerini genişleterek aday nesnekümelerini oluşturur. Ek olarak, SETM aday nesne kümelerinin TID'lerini de bilir. Aday nesne kümelerini oluştururken ilişkisel birleştirme işlemleri kullanılabilir.[31] Aday nesne kümelerini oluştururken aday nesnekümelerinin TID'leriyle birlikte bir kopyasını ardışık biçimde saklar. Sonra, aday kümeleri nesne kümelerini üzerinde sıralanır ve bir kabul fonksiyonuyla küçük nesne kümeleri silinir. Eğer veritabanı TID sıralı ise bir işlemde yer alan yaygın nesne kümeleri b bir sonraki geçişte $\overline{L}_k$ 'yı TID'ye göre sıralanarak elde edilir. Bu yolla veritabanı üzerinde birçok geçiş yapılır. Algoritma daha fazla nesne kümesi bulamadığında sonlanır [30].

4.3.1.3. Apriori Algoritması

Apriori algoritmasının ismi, bilgileri bir önceki adımdan aldığı için "pior" anlamında Apriori'dir.[30]. Bu algoritma temelinde iteratif (tekrarlayan) bir niteliğe sahiptir [32] ve hareket bilgileri içeren veritabanlarında sık geçen öğe kümelerinin keşfedilmesi kullanılır. Apriori algoritmasına özüne göre, eğer k- öğe kümesi (k adet elemana sahip öğe kümesi) minimum destek ölçütüne sağlıyorsa, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar.

Birliktelik kuralı madenciliği, tüm sık geçen öğelerin bulunması ve sık geçen bu öğelerden güçlü birliktelik kurallarının üretilmesi olmak üzere iki aşamalıdır. Birliktelik kuralının ilk aşaması için kullanılan en popüler ve klasik algoritmadır. Bu algoritmada özellikler ve veri, Boolean ilişki kuralları ile değerlendirilir [33].

k- öğe (k adet elemana sahip öğe kümesi) kümesi c ile ifade edilirse, öğeleri(ürünler) c[1], c[2], c[3],….c[k] şeklinde gösterilir ve c[1]<c[2]<c[3]….<c[k] olacak şeklinde küçükten büyüğe doğru sıradadır [30]. Her öğe kümesine destek ölçütünü tutmak üzere bir sayaç değişkeni eklenmiştir ve sayaç değişkeni öğe kümesi ilk kez oluşturulduğundan sıfırlanır. Geniş (sık geçen) öğe kümeleri L karakteri ile aday öğe kümeleri ise C karakteri ile gösterilir [34].

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

Şekil.9. Klasik Apriori Algoritması Özet Kodu[30]

Apriori algoritmasının klasik özet kodu Şekil 9'de [30] görülmektedir. Bu şekilde yer alan apriori-gen fonksiyonu (Şekil 10) [30], (k-1) adet öğeye sahip L_{k-1} sık geçen öğe kümesini kullanarak k adet öğeye sahip aday kümeleri oluşturur. Bu fonksiyon ile ilk önce, L_{k-1} sık geçen öğe kümesine kendisi ile *birleştirme (join)* işlemi uygulanır. Birleştirme işleminde L_{k-1} sık geçen öğe kümesinin her satırında yer alan son öğe haricinde diğer öğelerin çapraz olarak benzerliği aranır ve son öğe haricinde diğer öğelerle yakalanan benzerliklerden yeni aday öğe

kümeleri oluşturulur. Oluşan kümeler *budama (prune)* adımı ile budanarak fonksiyondan dönülür.

1) insert into C_k
2) select p.items$_1$, p.items$_2$,.....,p.items$_{k-1}$, q.item$_{k-1}$
3) from L_{k-1} p, L_{k-1} q
4) where p.item$_1$=q.item$_1$, ..., p.item$_{k-2}$=q.item$_{k-2}$,
5) p.item$_{k-1}$ < q.item$_{k-1}$;
6) forall itemsets $c \in C_k$ do
7) forall (k-1)-subsets s of c do
8) if ($s \notin L_{k-1}$) then
9) delete c from C_k

Şekil.10. Apriori-Gen Fonksiyonu[30]

Budama işleminde; c aday kümesinin (k-1) öğeye sahip alt kümelerinden L_{k-1} sık geçen öğe kümesinden yer almayan tüm alt kümler silinir. Farklı bir ifade ile budama, C_k aday öğe kümelerindeki öğelerin alt kümelerinin L_{k-1} sık geçen öğe kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_{k-1} sık geçen öğe kümesinde yer almıyorsa ilgili öğe değerlendirme dışı kalır ve C_k aday öğe kümesinden silinir [30].

Bu algoritma AIS ve SETM algoritmalarına göre, aday nesne kümelerinin üretilme şeklinde ve sayılacak aday nesne kümelerinin seçilmesinde farklılık arz eder. Daha önce de değinildiği üzere, hem AIS hem de SETM algoritmalarında bir önceki geçişteki yaygın nesne kümeleri arasındaki ortak nesne kümeleri ve bir hareketteki nesneler elde edilir. Bu ortak nesnekümeleri hareket içindeki diğer bağımsız nesnelerle genişletilerek aday nesne kümeleri oluşturulur. Buna rağmen bu bağımsız nesneler yaygın olamazlar. Bilindiği gibi yaygın bir nesnekümesinin süperkümesi ile küçük bir nesnekümesi, küçük bir nesne kümesi sonucunu verir. Bu teknikler küçük olarak belirlenecek birçok aday nesne kümesi üretir. Apriori algoritması bu önemli nokta üzerinde odaklanır. Apriori, önceki geçişte oluşan yaygın nesnekümelerini birleştirerek aday nesne kümelerini

oluşturur ve veritabanındaki hareketlerle ilgilenmeden, önceki geçişte oluşan altkümelerden küçük olanlarını siler. Önceki geçişte oluşan yaygın nesnekümelerinin dikkate alınmasıyla yaygın aday nesne kümelerinin sayısında anlamlı bir azalma olur.

İlk geçişte sadece bir nesne içeren nesne kümeleri sayılır. İlk geçişte bulunan yaygın nesne kümelerinden *apriori_gen()* fonksiyonu kullanılarak, ikinci geçiş için aday kümeleri oluşturulur. Aday nesne kümeleri bulunduğunda veritabanı taranarak iki uzunluklu yaygın nesnekümelerini bulmak için destek değerleri sayılır. İ1k geçişteki yaygın nesnekümeleri, üçüncü geçişteki yaygın nesnekümelerini oluşturacak aday kümeler olarak ele alınırlar.

Apriori, seviye mantığı (level_wise) arama olarak bilinen yinelemeli bir yaklaşım kullanılır. Bu yaklaşmada k-nesne kümeler, (k+1) nesne kümelerin araştırılması için kullanılır. İlk olarak, yaygın 1- nesne kümelerinin kümesi bulunur. Bulunan bu küme L_1 olarak adlandırılır. L_1, L_2'nin (yaygın 2-nesnekümelerin kümesi) bulunmasında kullanılır. L_2, L_3'ün bulunmasında kullanılır ve algoritma şekilde daha fazla yaygın k-nesnekümeler bulamayıncaya kadar yinelemeli bir şekilde devam eder. Her L_k'nın bulunması bütün veritabanının taranması anlamına gelmektedir [2].

apriori_gen() fonksiyonunun iki adımı vardır[30]. İlk adım esnasında L_{k-1} kendisiyle birleştirilerek C_k elde edilir. İkinci adımda, birleşme sonucunda oluşmuş ve (k-1) altkümelerinden bir veya birkaçı L_{k-1} 'de olmayan tüm nesneler silinir. Kalan yaygın k-nesne kümeleri sonuç olarak döndürülür. Subset () fonksiyonu, bir harekette görülen aday kümelerin altkümelerini döndürür. Adayların destek değerlerinin sayılması, algoritmada zaman alan bir adımdır[4].

```
1)  L_1 = find_frequent_1-itemsets(D);
2)  for (k = 2; L_{k-1} ≠ φ; k++) {
3)      C_k = apriori_gen(L_{k-1}, min_sup);
4)      for each transaction t ∈ D { // scan D for counts
5)          C_t = subset(C_k, t); // get the subsets of t that are candidates
6)          for each candidate c ∈ C_t
7)              c.count++;
8)      }
9)      L_k = {c ∈ C_k | c.count ≥ min_sup}
10) }
11) return L = ∪_k L_k;

procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets; min_sup: minimum support)
1)  for each itemset l_1 ∈ L_{k-1}
2)      for each itemset l_2 ∈ L_{k-1}
3)          if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2]) ∧ ... ∧ (l_1[k-2] = l_2[k-2]) ∧ (l_1[k-1] < l_2[k-1]) then {
4)              c = l_1 ⋈ l_2; // join step: generate candidates
5)              if has_infrequent_subset(c, L_{k-1}) then
6)                  delete c; // prune step: remove unfruitful candidate
7)              else add c to C_k;
8)          }
9)  return C_k;

procedure has_infrequent_subset(c: candidate k-itemset; L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
1)  for each (k-1)-subset s of c
2)      if s ∉ L_{k-1} then
3)          return TRUE;
4)  return FALSE;
```

Şekil.11. Apriori Algoritması [2]

4.3.1.4. Apriori-TID Algoritması

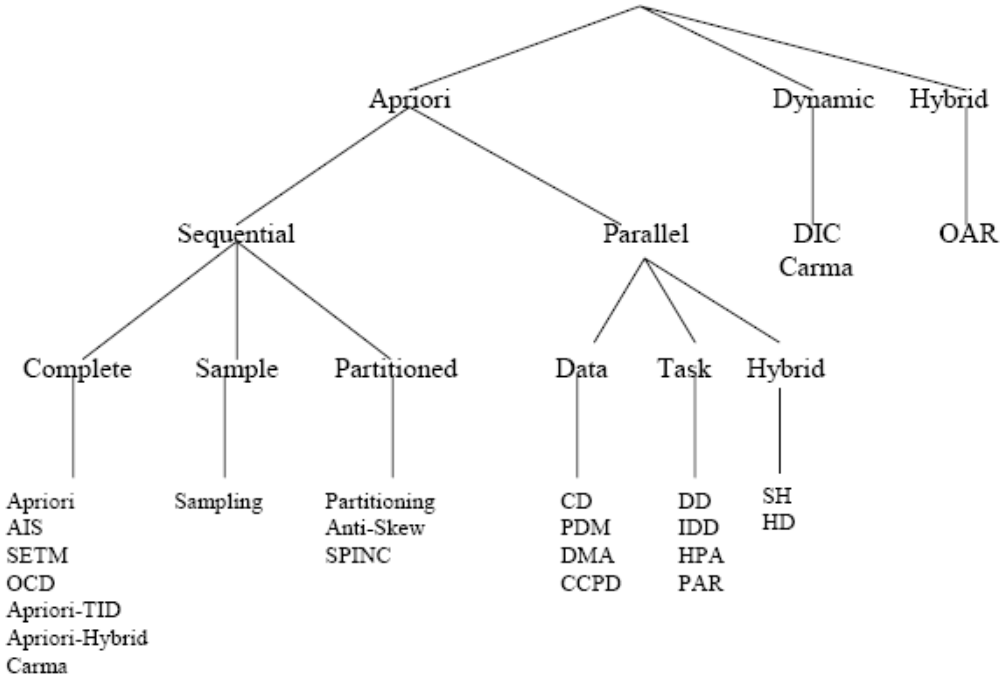
Daha önce de belirtildiği üzere, Apriori destek değerlerini saymak için her geçişte tüm veritabanını tarar. Ancak her geçişte tüm veitabanını taramak gereksizdir. Bu varsayıma dayanarak 1994 yılında Agrawal ve diğerleri, Apriori-TID ismiyle bir algoritma önermişlerdir. Apriori'ye benzer olarak, bir geçişin başında aday nesnekümelerini belirlemek için Apriori-TID de Apriori'nin aday üretim fonksiyonunu kullanır. Apriori'ye göre ana fakı ise ilk geçişten sonra destek değerlerini saymak için veritabanını kullanmaz. Bundan ziyade, $\overline{C}_k$ ile gösterilen, önceki geçişte kullanılan aday kümelerinin şifrelenmiş şeklini kullanır. SETM'deki gibi $\overline{C}_k$ 'nın her bir üyesi <TID, X_k> biçimindedir, TID tanımlayıcı, X_k ise potansiyel yaygın k-nesnekümesidir.

İlk geçişte, elde edilen C_1 veritabanına benzer. Buna rağmen, her nesne nesnekümesi tarafından yer değiştirilir. Diğer geçişlerde, T hareketinin yerini tutan

$\overline{C}_k$'nın bi üyesi <TID, c> olarak ifade edilir. c, T'de bulunan C_k'ya bağlı bir adaydır. Bundan dolayı, C_k'nın büyüklüğü veritabanındaki hareket sayısından daha küçük olur. Ayrıca $\overline{C}_k$'daki he bir kayıt, büyük k değerlerindeki yerini tutan hareketlerden daha küçüktür. Bunun nedeni çok az adayın harekette içerilmesidir. Başka şekilde ifade etmek gerekirse, $\overline{C}_k$'daki her bir kayıt küçük k değerlerindeki ilgili hareketten daha büyüktür [31].

4.3.1.5. Apriori-Hybrid Algoritması

Bu algoritma, veri üzerindeki tüm geçişlerde aynı algoritmanın kullanılmasının zorunlu olmadığı fikrine dayanır. 1994 yılında Agrawal ve diğerleri tarafından değinildiği üzere, Apriori ilk geçişlerde daha iyi performansa sahiptir ve sonraki geçişlerde Apriori-TID, Apriori'den daha iyidir. Deneysel gözlemlere dayanarak Apriori-Hybrid tekniği, ilk geçişlerde Apriori'yi kullanacak şekilde tasarlanmıştır ve geçişlerin sonundaki $\overline{C}_k$ kümesinin bellekte karşılanacağını umarak Apriori-TID'e geçiş yapar. Bundan dolayı, her bir geçiş sonunda $\overline{C}_k$'nın tahmin edilmesi gerekmektedir. Ayrıca, Apriori'den Apriori-TID'e geçiş masraflıdır. Bu tekniğin performansı büyük verisetleri üzerindeki tecrübelere dayanarak da hesaplanmıştır. Geçişlerin sonlarına doğru değişimin olması durumu dışında, Apriori-Hybrid'in Apriori'den daha iyi performans gösterdiği belirlenmiştir [31].



Şekil.12. Birliktelik algoritmalarının sınıflandırılması [35]

4.4. Birliktelik Kuralı Algoritmalarının Karşılaştırılması

Veritabanının herhangi bir taranmasında sayılan adayların azami sayısına bakılarak alan ihtiyacı tahmin edilebilir. Zaman ihtiyacını belirlemek için gereken veritabanı taramalarının azami sayısının (G/Ç tahmini) ve karşılaştırma işlemlerinin azami sayısının (işlemci tahmini) sayılması gerekir. Birçok hareketsel veritabanlarının ikincil disklerde saklanmasından ve G/Ç ek yüklerinin işlemci ek yüklerinden daha önemli olmasından bu yana, veritabanının bütününde yapılan tarama sayılarına odaklanılmıştır. Açıkça, veritabanındaki her bir hareketi tüm nesneleri içermesi en kötü durumdur. Her bir hareketteki nesne sayısı m ve D veritabanındaki k-nesnelerini içeren yaygın nesnekümeleri L_k olarak ifade edilmek üzere, yaygın nesnekümeleri sayısı 2^m kadardır. Seviye-mantığı (level-wise) tekniklerde (AIS, SETM, Apriori, … vb.), veritabanının ilk taranması sırasında tüm L_1 yaygın nesnekümeleri elde edilmiş olur. Benzer şekilde, tüm L_2 yaygın nesnekümeleri ikinci tarama sırasında elde edilir ve bu işlem böyle devam eder. *m.*

taramada L_m nesne kümesi elde edilir. Yaygın nesnekümelerine ek girişler olmadığında ek taramaya gerek kalmayacağı için tüm algoritmalar sonlanır.

Bir algoritmanın iyiliği, geliştirdiği "doğru" adayların sayısına bağlıdır. Daha önce de belirtildiği gibi, tüm algoitmalar aday kümeleri oluşturmak için önceki geçişlerde oluşan yaygın nesnekümelerini kullanır. Aday nesnekümelerini oluşturmak için bir önceki nesnekümelerinden oluşturulmuş yaygın nesnekümeleri ana belleğe alınır. Aday nesnekümeleri, destek değerlerinin sayılması için ana bellekte olmalıdır. Belleğin yetersiz gelmesi durumuna binaen farklı çeşitlilikte tampon (buffer) yönetimi ve depolama mimarilerine imkân veren çeşitli algoritmalar vardır. AIS, L_{k-1}'in ihtiyaç durumunda diskte kalmasını önermiştir. SETM, eğer $\overline{C}_k$ ana bellekte yer alamayacak kadar büyükse, FIFO (first in first out) tarzında belleğe yazılmasını önermiştir. Apriori ailesi, L_{k-1}'in diskte saklanmasını ve C_k'yı bulmak için bir bloğun ana belleğe alınmasına değinmiştir. Buna rağmen hem Apriori-TID hem de Apriori-Hybrid'de C_k, destek değerlerinin hesaplanması için ana bellekte olmalıdır. Diğer yandan, tüm diğer teknikler bu problemlerle baş etmek için yeterli belleğin olduğunu varsayarlar. Diğer tüm sıralı teknikler (PARTITION, Sampling, DIC ve CARMA) ana bellekte yer kaplayabilecek kadar tüm veritabanının elverişli bir bölümünün bulunmasına odaklanır. Apriori ailesi, yaygın nesnekümeleri için hash ağacı veya dizisi (hash tree or array) gibi uygun veri yapılarını önermiştir. Buna rağmen AIS ve SETM, herhangi bir depolama mimarisi önermemiştir.

Birçok ticari şekilde uygulanabilir araçlar, birliktelik kuralları oluşturmak için Apriori tekniğini baz alırlar. Bazı algoritmalar belirli durumlar altında kullanmak için son derece uygundur. AIS, veritabanındaki nesnelerin çok büyük sayıda olması durumda iyi çalışmaz. Bundan dolayı AIS, küçük işlemsel veritabanları için uygundur. Daha önce de bahsedildiği üzere, Apriori ilk geçişlerde Apriori-TID' e oranla küçük çalışma sürelerine ihtiyaç duyar. Diğer yandan, Apriori-TID sonraki geçişlerde Apriori'ye göre daha iyidir. Bundan dolayı uygun geçişle, Apriori-Hybrid en iyi performansı göstermektedir. Buna rağmen, Apriori'den Apriori-TID'e geçiş çok önemli v ek yük getiren bir iştir. OCD, tahmini teknik olmasına rağmen, düşük destek eşik değerlerindeki yaygın nesne kümelerini

bulmakta çok etkindir. CARMA kullanıcı etkileşimli, geri dönüşüm tabanlı online bir tekniktir ve hareket dizilerinin bir ağdan okunması sırasında etkilidir.

Aşağıdaki tablo, bahsedilen çeşitli algoritmaların kısaca karşılaştırılmasına ışık tutmakta ve özetlemektedir. Bu tablo azami tarama sayısını, önerilen veri yapılarını ve bazı açıklamaları içermektedir.

Algoritma	Tarama Sayısı	Veri Yapısı	Açıklamalar
AIS	m+1	Belirtilmemiş	Belli başlı, az veri içeren seyrek hareketsel veritabanları için uygundur.
SETM	m+1	Belirtilmemiş	SQL uyumlu.
Apriori	m+1	L_{k-1}: Hash tablosu C_k: Hash ağacı	Belli başlı, orta düzey veri içeren hareketsel veritabanları için uygundur. AIS ve SETM'e oranla daha iyidir. Paralel algoritmaların baz aldığı algoritmadır.
Apriori-TID	m+1	L_{k-1}: Hash tablosu C_k: TID ile dizinlenmiş dizi C_k: Sıralı mimari ID: bitmap	C_k'nın büyük olması durumunda çok yavaştır. Küçük boyutlu C_k ile Apriori'den daha iyidir.
Apriori-Hybrid	m+1	L_{k-1}: Hash tablosu Đlk safhada: C_k: Hash ağacı Đkinci safhada: C_k: TID ile dizinlenmiş dizi C_k: Sıralı mimari ID: bitmap	Apriori'den daha iyidir. Buna rağmen, Apriori'den Apriori-TID'e geçiş ek yük gerektirir. Geçiş noktasını saptamak çok önemlidir.
OCD	2	Belirtilmemiş	Düşük destek eşik değerleri ve çok büyük veritabanları için uygundur.
PARTITION	2	Hash tablosu	Büyük veritabanları için uygundur. Homojen veri dağılımını destekler.
Sampling	2	Belirtilmemiş	Düşük destek eşik değerleri ve çok büyük veritabanları için uygundur.
DIC	Aralık boyutuna bağlıdır	Ağaç	Veritabanı, hareketsel aralıklar şeklindedir. Yüksek boyutlu adaylar, bir aralığın sonunda oluşturulur.
CARMA	2	Hash tablosu	Hareket dizilerinin bir ağdan okunmasına uygundur. Online. Kullanıcılardan sürekli geri-besleme gelir ve destek ve/veya güven değerlerini işlem sırasında herhangi bir anda değiştirebilirler.
FP-Growth	2	FP-Tree	Aday üretimsiz bir tekniktir. Apriori ve türevlerinden daha iyi performansa sahiptir.
CD	m+1	Hash tablosu ve ağacı	Veri paralelleştirme.
PDM	m+1	Hash tablosu ve ağacı	Veri paralelleştirme, erken aday budaması.
DMA	m+1	Hash tablosu ve ağacı	Veri paralelleştirme, aday budaması.
CCPD	m+1	Hash tablosu ve ağacı	Veri paralelleştirme, paylaşımlı-bellek içeren makinelerde kullanılır.
DD	m+1	Hash tablosu ve ağacı	Görev paralelleştirme, round-robin bölümlendirmesi.
IDD	m+1	Hash tablosu ve ağacı	Görev paralelleştirme, ilk nesnelerle bölümlendirme.
HPA	m+1	Hash tablosu ve ağacı	Görev paralelleştirme, hash fonksiyonu ile bölümlendirme.
SH	m+1	Hash tablosu ve ağacı	Veri paralelleştirme, adaylar her bir işlemcide bağımsız olarak oluşturulur.

Şekil.13. Algoritmaların karşılaştırılması [35]

5. METİN MADENCİLİĞİ

5.1. Veri, Metin ve Web Madenciliği

Yapısal veri, bir yapı içerisinde organize edilebilen ve bundan dolayı tanımlanabilen veri için kullanılan bir terimdir. En yaygın kullanılan yapısal veri kaynakları SQL (Structured Query Language) ve Access gibi veri kaynaklarıdır. Örneğin SQL, kolon (değişken) ve satır (kayıt) bazlı bilginin seçimine imkân vermektedir. Yapısal veri, içerikteki veri tipine göre organize edilebilen ve arama yapılabilen veridir. Buna karşın yapısal olmayan verinin tanımlanabilir bir yapısı yoktur. En çok bilinen yapısal olmayan veri türleri; resim dosyaları, pdf, word ve text gibi metin dosyaları, web üzerinde tutulan log dosyaları ve e-postalardır. E-postalar veritabanlarında Microsoft Outlook gibi araçlar ile organize edilebilmesine rağmen bu tür veriler herhangi bir yapısal veri türü ile eşleşmediklerinden ham veri olarak düşünülür. Excel gibi hücre yapısına sahip veri türleri yapısal olmasına rağmen halen yapısal olma ve olmama konusundaki yeri tartışılmaktadır.

Birçok kurumun verisinin çoğu yapısal olmayan veri olarak veritabanlarında tutulmaktadır. Merrill Lynch, potansiyel olarak kullanılan bütün verilerin yaklaşık %80'inin yapısal olmayan türde olduğunu ifade etmiştir. [36,37,38].

Veri madenciliği büyük veri yığınlarında gizli olan örüntüleri ve ilişkileri ortaya çıkarmak için istatistik ve yapay zekâ kökenli çok sayıda ileri veri çözümleme yönteminin tercihen görsel bir programlama ara yüzü üzerinden kullanıldığı bir süreçtir. Veri madenciliği algoritmaları; istatistiksel algoritmalar, matematiksel algoritmalar ve yapay zekâ algoritmalarını (sinir ağları, karar ağaçları, kohonen ağlar, birliktelik kuralları vb.) bir arada içerir [39].

Veri madenciliği çözümleri ve algoritmaları metin veya web verisindeki kalıpları bulmadan veya model oluşturmadan önce metin veya web verisinin yapısal olması gerekmektedir. Metin ve Web madenciliği işlemleri, veri madenciliğinde kullanılacak yapısal veriye ulaşmak için kullanılan araçlar olarak tanımlanabilir.

Metin ve web madenciliği son yıllarda oldukça fazla çalışılan birbiri ile ilişkili alanlardır. Metin madenciliği, çok büyük belgelerin analizi ve metin tabanlı verinin içerisindeki gizli kalıpların elde edilmesidir. Web madenciliği ise, web içerikleri, sayfa yapıları ve web bağlantı istatistiklerinin de içinde olduğu web ile ilişkili olan verinin analizini içermektedir [37].

5.1.1. Metin Madenciliği

Veri farklı şekillerde bulunabilir. Bazıları otomatik veri analizi için üstesinden gelinebilir ve uygun iken bazıların analizi çok daha zordur. Klasik veri analiz yöntemleri verinin değişken ve kayıt bazlı düzenlendiği varsayımı ile işlem yapmaktadır. Buradaki soru, eğer veri metin formatında yani kayıtların ve değişkenlerin olmadığı bir yapıda ise ne yapmamız gerektiğidir. Metin verisindeki anlamın ortaya çıkarılabilmesi için kullanılan yöntem metin madenciliğidir.

Metin yazımında standart kurallar olmadığından dolayı bilgisayar bunları anlayamamaktadır. Her bir metnin dili ve içerdiği anlam amaca bağlı olarak çeşitlilik göstermektedir. Yapısal olmayan bilgiden içerik çıkarmak için kullanılan geleneksel yöntemler; anahtar kelimeler veya mantıksal aramalar, istatistiksel veya olasılıksal algoritmalar, sinir ağları ve kalıp keşfedici sistemler gibi dilbilimsel olmayan yöntemlerdir.

Bu yöntemler, hem sorgudaki hem de metindeki kelimelerin karakterlerini karşılaştıran bir temele dayanır. Bundan dolayı içeriği açıklayıcı sonuçlar elde edemez. Dili anlamının temeli dilbilimsel yollara dayanır ve bu çoğunlukla Natural Language Processing (NLP) olarak ifade edilir. NLP'yi içeren bir sistemde, karmaşık yapıların bulunduğu ifadeler (örneğin; duştan akan soğuk su ile içilen soğuk su arasındaki fark gibi) akıllı olarak çıkarabilmekte ve terimleri sınıflayarak; ürünler, organizasyonlar veya kişiler gibi sınıflara atamaktadır.

Metin madenciliği doğal dil metinlerinden bilgi ve nitelikli bilgi elde edilmesi sürecidir.

iki aşamada gerçekleşir.

- Anahtar içerik/ifadeler metinden elde edilir,
- Elde edilen içerik/ifadeler, yüksek dereceden ilişkili olduğu kategorilere atanır. Bu aşamaları basit bir örnek üzerinden açıklamak gerekirse;

1. Aşama: “CPU” ve “CD-ROM” ifadeleri metinden elde edilir,

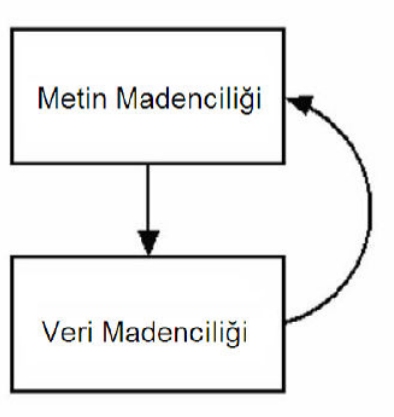
2. Aşama: Bu iki ifade, otomatik olarak “Bilgisayar Donanımı” etiketli kategoriye aktarılır. Metin madenciliği uygulamaları iki ana sınıfta ayrılabilir:

- Metnin anlaşılması/özetlenmesi: Metin madenciliğinin amaçlarından bir tanesi metinden anlamlı nitelikli bilginin çıkarılmasıdır. Böylece metnin içerdiği anahtar içerik anlaşılabilecektir. Örneğin, yavaş tamir veya sipariş gibi sorunlar yüzünden $ikayet eden müşterilerin oranını öğrenmek isteyebiliriz.

- Metin ile modelleme: Daha yaygın olarak, metin madenciliği terk etme veya ürün alma gibi müşteri davranışlarının tahmin edildiği bir modelin geliştirilmesi aşamasının bir bölümünü oluşturmaktadır. Metinden elde edilen içerik girdi değişkeni olarak kullanılır ve diğer bilgiler ile beraber öngörüsel model geliştirilir.

Veri madenciliği girdi olarak sadece yapısal veriyi kullandığından dolayı veri madenciliği çözümleri ve algoritmaları kullanılarak metin verisinden kalıplar bulunup, modeller kurulmadan önce metinden elde edilecek bilginin yapısal hale dönüştürülmesi zorunludur. Metin madenciliği sonucunda, kategorilerin olu$turulması ile yapısal olmayan veri yapısal hale dönüşmektedir[40,41,42].

Metin ve veri madenciliği arasındaki ilişki şekilde tanımlanmıştır.



Şekil.14. Süreçler arasındaki ilişki[40]

Şekil 14 de görüldüğü gibi, metin ve veri madenciliği arasında interaktif bir ilişki vardır. Metin madenciliği sonucunda elde edilene yapısal veri, veri madenciliği modellerinde kullanılmakta ve elde edilen sonuçlar daha sonra metnin yapısının incelenmesinde kullanılmaktadır.

Metin madenciliği uygulama alanlarından bazıları;

A) Finans Alanı

- Borsadaki haberlerin anlık olarak işlenmesi ve finansal çıkarım
- Spekülatif haberlerin ve satın almaların tespiti

B) Pazarlama - CRM Alanı

- Teknolojideki ve insanların tüketim alışkanlıklarındaki yeni trendlerin tespiti
- Anlık kişi, profil, içerik analizinin yapılması ve kişiye özel reklam sistemlerinin oluşturulması
- Müşterilerin, internette firmalar ve ürünleri hakkında paylaştığı görüşlerinin tespiti
- Müşteri hizmetlerine yapılan aramaların veya yazılı şikâyetlerin otomatik olarak gruplanması ve konunun tespit edilerek ilgili birimlere otomatik yönlendirilmesi

- Bütün müşterilerin e- mail, işlem, çağrı merkezi ve anket gibi erişim noktalarından elde edilen metin bilgilerinden nitelikli bilgi çıkarılır. Bu nitelikli bilgi müşterinin terk etme ve çapraz satışlarını tahmin etmek üzere kullanılır.

C) Sağlık ve Biyoloji Alanı

- Sağlık ve biyoloji alanındaki en son yenilikler ve yazılı makaleler ile ilgili bilgilerin derlenmesi, özet haline getirilerek hastane içi birimlere özel gruplandırılması, bu yolla her türlü yeniliğin takibi
- Hasta bilgi kaydı ve raporlarının analizi ve bu yolla belirli bir hastalığı tetikleyen bilinmeyen etmenlerin veya olası genetik eğilimlerin tespiti
- Hasta raporları, makale başlıkları, yayınlanmış araştırma sonuçları ve diğer yayınlar gibi metin materyallerinden çıkarım yapılır.

D) Eğitim Alanı

- Akademik bir çalışmanın çalıntı olup olmadığının tespiti,
- İsimsiz bir metnin yazarının tespiti.

E) Kamuya Özgü Genel ve İstihbarat Amaçlı Uygulamalar

- Geçmiş patentlerin analizi sonucu, yeni patent başvuruların olası benzerliklerinin tespiti-önlenmesi,
- Polis vaka kayıtlarının analizi ve yeni vakalar ile eskilerinin benzerliklerinin tespiti-ilişkilendirilmesi,
- MSN ve e-mail üzerinden dönen bilgi trafiğinin zeki arabirimlerce izlenmesi, erken uyarı sistemi olarak kullanılması,
- Şifreli yazışmaların dilin temel yapısına uygun olarak çözümlenmesine yönelik uygulamalar,
- Kara para aklama ve hesap hareketlerinin, şirketler arası yazışmaların incelenmesi sonucu, link analizi için gerekli enformasyonun çıkarımı ve sonucu tüm şebeke ve üyelerinin ortaya çıkarılması,

- Hukuki davaların sonuçları ile vaka özetlerinin ilişkilendirilmesi ve hâkimlerin karar vermesini kolaylaştırıcı yönde benzer diğer dava sonuçlarının otomatik tespiti.
- Bilirkişi raporlarının semantik olarak indekslenebilmesi ve metin tabanlı örnek bilirkişi raporu aratabilme imkânı,

F) İnternet - Yazılım Alanı

- Sitelerdeki illegal içeriğin otomatik tespiti,
- Spam Maillerin zeki arabirimlerce ayıklanması,
- Yapılması planlanan bir yazılım projesinin özelliklerinden hareketle gerekli teknik ihtiyaçların otomatik çıkarımı,
- Bir yazılımın planlama safhasındaki dokümantasyonlarına bakılarak yazılımın kaynak kodu ile örtüşüp örtüşmediğinin bulunması,
- Çok daha sağlıklı işleyen arama sonuçlarının ve arama motorlarının kurgulanması
- Bir metnin hangi dilde yazıldığının otomatik tespiti,
- Şirketler bünyesindeki büyük veri kümlerinin(dataset) gruplanması ve veri madenciliğine uygun hale getirilmesi.

G) Diğer Ticari Uygulamalar

- Verilen bir metinden veya haberden özet çıkarımı,
- Farklı kaynaklardan gelen ancak aynı konu ile ilgili haberlerin otomatik tespiti,
- Düzensiz veri kümelerinin düzenli hale getirilmesi, veri madenciliği içinde kullanılabilecek hale getirme,
- Bir metnin farklı bir dile otomatik çevrimi,
- Sahtekârlık (Fraud) tespiti: Sağlık, sigorta ve hükümet tarafında toplanan büyük çaptaki metin verilerinde kalıplar ve anormallikler aranarak sahtekârlıklar tespit edilir,
- Güvenlik/istihbarat: Organizasyonlar ve bireyler arasındaki kalıplar ve bağlantılar, terörist tehlikeleri ve kriminal davranışları tahmin etmek ve engelleyebilmek için büyük çaptaki metin içerisinde aranır,

- Pazar araştırması: Yayınlanmış belgeler, basın bültenleri ve web sayfaları pazar etkisinin ölçülmesi için aranır ve izlenir. Metin madenciliği kantitatif yöntemler ile açık uçlu anket soruları ve mülakatların değerlendirilmesinde kullanılabilmektedir [40, 42].

5.1.2. Metin Veri Madenciliğinin Önemi ve Sorunları

Özellikle internet ve kişisel bilgisayarların yaygınlaşmasına bağlı olarak, gittikçe büyüyen hacme sahip doküman yığınları oluşmaktadır. Bu yığınlar içinde önemli bilgiler kaybolup giderken, değerli bilgilere ulaşmak için dokümanların içeriğinin belirlenmesi ve buna uygun sorgulanabilmesi ihtiyacı kendini hissettirmektedir.

Bilgisayarlı sistemlerden önce, dokümanlar içindeki bilgiye erişim için elle indeksleme sistemleri kullanılıyordu. 1996-2001 yılları arasında yayınlanan 164.000 periyodik yayın olduğu düşünülürse, elle indekslemenin yavaş, zor ve yetersiz olduğunu anlaşılabilir. Şu an internette 2 milyardan fazla web sayfası bulunmaktadır. Bu kadar çok dokümanın elle indekslenmesinin ve bu indekslerin güncellenmesinin zorluğunun yanı sıra, elle yapılan indeksleme uzmanların sübjektif yorumlarını da içerdiğinden, aranılan bilgilere ulaşmayı kolaylaştıramamanın ötesinde, yanıltıcı sonuçlara da neden olabilmektedir.Bu problemleri aşmak için otomatik bilgiye ulaşma yöntemleri geliştirilmiştir. Bu yöntemler yarı-yapısal verilerin madenciliği ya da doküman veri madenciliği adı altında incelenmektedir.

Geleneksel bilgiye ulaşma yöntemleri, doküman yığınları içinden ana konu başlıklarına yönelik aramaları başarılı bir şekilde karşılayabilmektedir. Örneğin Google arama motoru, en başarılı araçlardan biri olarak, "müşteri ilişkileri yönetimi" gibi bir ana konuyla ilişkili dokümanları sorunsuz bir şekilde sorgu sonucu olarak getirmektedir. Ancak, arama yapılan doküman sayısı artıkça, sorgu neticesinde gelen doküman sayısı binlerle ifade edilir hale gelebilmekte, bunun sonucu olarak da daha özel ihtiyaçları karşılayabilecek çözümlere ihtiyaç duyulmaktadır. Örneğin "müşteri ilişkileri yönetiminin işletme verimliliğine etkisi" gibi bir konunun sorgulanması ihtiyacı doğabilmektedir.

Google gibi sistemler, dokümanları, içindeki kelimelerle; dokümanların arasındaki ilişkileri de bu kelimelerin tüm doküman yığını içindeki istatistikî kullanım örüntüleri ile ifade etmektedirler Kelimeleri baz alan bu sistemler otomatik indeksleme yapabildiklerinden, dokümanların madenciliği açısından büyük kolaylıklar sağlamaktadır. Ancak bu sistemler belirli kısıtlara da sahip olmuşlardır.

İlk kısıt yazarların dokümanları oluştururken kullandığı kelimelerle, bu dokümanlar içinde arama yapmak isteyen kişilerin sorgu ifadelerinde kullandıkları kelimelerin birbirinden farklı olabilmesidir. Bunun nedeni kelimelerin çokanlamlı ve/veya eşanlamlı olabilmesidir. Benzer edecek şekilde kullanılması durumu da bu kısıtın başka bir boyutudur. Bu nedenle, kelimeler her zaman aynı anlamı taşımayabilir. İlerde anlatılacak Gizli Anlambilimsel Dizinleme (GAD) yöntemi bu kısıtı ortadan kaldıran bir çözümdür.

İkinci kısıt kelimelerin tek başlarına taşıdığı anlamın dışında yan yana kullanımları ile bambaşka anlamlar ifade edebilmesi durumudur. Bu çalışma kapsamında geliştirilen n-gram kelimelerle GAD yöntemi ile bu kısıt ortadan kaldırılarak, daha detay konularda arama yapmayı sağlamak mümkün hale getirilmiştir.

Üçüncü kısıt mevcut sistemlerin kelimelerin anlama kattığı değeri hesaplamak için sadece ne sıklıkla kullanıldıklarına bakmasıdır, oysa kelimelerin dokümanların anlamını ifade etmede diğerlerinden daha etkili olduğu farklı durumlar da vardır. Örneğin Türkçe cümlelerde fiile yakın kelimeler anlam açısından daha önemlidir. n-gram kelimelerle GAD yönteminin kullandığı kelime grubu oluşturma yaklaşımı ile bu kısıt için de bir çözüm oluşturmaktadır.

Doküman madenciliği için kullanılan teknikler iki ana dalda incelenebilir.

- eğitmenli sistemler
- eğitmensiz sistemler

Eğitmenli sistemler, önce bir deney seti ile girdi ile çıktı arasındaki ilişkiyi öğrenen ve daha sonra kendisine verilen girdi için uygun çıktıyı üreten sistemlerdir.

Bu gibi sistemlerin başarısı, öğrenme süreçlerinin doğruluğuna bağlıdır ve sürekli genişleyerek büyüyen doküman yığınlarına uygulanmaları zordur.

Eğitmensiz sistemler ise, eğitmenli sistemlerdeki, sistemin eğitilmesi için yapılan işlemleri içermez. Bu sistemler doğrudan doküman seti üzerinde çalışır. Eğitmensiz sistemlerin her türlü doküman yığını üzerinde hiçbir ön işlem yapmadan çalıştırılabilmesi, eğitmenli sistemlere göre büyük esneklik sağlar [45,46].

5.1.3. Web Madenciliği

Web madenciliği işlemleri kullanılarak yapısal olmayan web verileri yapısal veriye dönüştürülür. Web madenciliği uygulamaları temel olarak üç alt başlık altında toplanabilir;

- Web yapı madenciliği: Web yapı madenciliği ile internetin temel yapısını olu$turan web siteleri, web sayfaları arası ya da web sayfasındaki bağlantılar arasındaki ilişkiler incelenir.

- Web içerik madenciliği: Web içerik madenciliği ile web sayfalarının içerikleri incelenir ve kullanışlı bilgi çıkarımı sağlanır. Web içerik madenciliği kullanarak web sayfalarının başlıkları, içerisinde geçen kelimeler, resimler veya müzik dosyaları incelenir. Bulunan içeriklere göre web siteleri belirli sınıflara veya kümelere ayrılabilir.

- Web kullanım madenciliği: Web kullanım madenciliği ile web sunucularında tutulan kullanıcı erişim kayıtları incelenerek anlamlı ve faydalı kalıplar bulunabilir. Web kullanım madenciliği yöntemleri uygulanarak web sitelerini ziyaret eden kişilerin davranış ve tutumları belirlenebilir.

Web madenciliğinin günümüzde birçok alanda kullanılmasının en önemli sebebi; kişilerin web sayfalarında göstermiş oldukları davranışların, hareketlerin ve yapmış oldukları işlem bilgilerinin var olan iş süreçlerine entegrasyonunu sağlayarak müşterinin en iyi şekilde anlaşılmasını sağlayan müşteri odaklı bir sistem oluşturmasıdır.

Web madenciliği kullanım alanları aşağıdaki gibidir;

- Web üzerinden ürün satışı gerçekleştiren şirketler web verilerini analiz ederek müşteri profili ve kümeleri oluşturmaktadırlar.

- Google vb. arama motorları web içerik madenciliği uygulayarak aranan anahtar kelimeyi içeren web sitelerini belirlemektedirler.

- Web madenciliği uygulanarak web sitelerinin iyileştirilmesi ve güncel kalması sağlanmaktadır[43,44].

5.1.4. Metin Verilerinin İncelenmesi ve Enformasyonun Çıkartılması

5.1.4.1. Metin Verilerinin Çözümlenmesi ve Bilgi Çıkarımı

"Bilgi çıkarımı nedir?". Bilgi kazanımı, yıllardır veri tabanı sistemleri ile paralel olarak geliştirilmektedir. Yapısal veriler üzerinde sorgu ve işlembilgi işleme üzerine odaklanan veri tabanı sistemlerinin aksine bilgi çıkarımı organizasyon ile ilgili olup, metin tabanlı dokümanlardan bilginin çıkarılmasıdır. Tipik bir bilgi çıkartımı problemi anahtar kelimeler veya örnek dokumanlar vb. kullanıcı girişlerine bağlı olarak ilişkili dokümanların bulunmasıdır. Tipik bilgi çıkartım sistemleri, çevrim içi kütüphane katalog sistemleri ve çevrim içi doküman yönetim sistemlerini içerir.

Madem bilgi çıkarımı ve veri tabanı sistemlerinin her biri farklı tipte veriyi işlemektedirler; uyumluluk kontrolü, geri kazanım, işlembilgi yönetimi ve güncelleme gibi bazı veri tabanı sistemi problemleri, genellikle bilgi çıkartımı sistemlerinde bulunmazlar. Ayrıca yapısal olmayan dokümanlar, anahtar kelimelere bağlı olarak yaklaşıklık taraması ve anlamlılık vb. gibi bazı ortak bilgi çıkartımı problemlerine genellikle, geleneksel veri tabanı sistemlerinde rastlanmaz [47].

5.1.5. Metin Çıkartımı İçin Temel Ölçümler

"Varsayalım ki bir metin çıkartımı sistemi sorgu formundaki bir girişimize bağlı olarak birçok doküman getirmiş olsun. Peki, sistemin doğru çalışıp çalışmadığını nasıl değerlendireceğiz?" Sorgu ile ilişkili doküman kümesini [Relevant] olarak ve sonuçta elde edilen dokümanları ise [Retrieved] olarak adlandıralım. Hem ilişkili hem de elde edilen dokümanları Venn şemasında görüldüğü gibi [Relevant] ∩ [Retrieved] olarak adlandıralım. Burada metin çıkarımının kalitesini değerlendirmek için iki temel ölçümümüz bulunmaktadır[48].

Hassasiyet: Sorgu ile ilişkili elde edilen dokümanların, elde edilen dokümanlara olan oranının yüzdesidir (örn. "doğru sonuçlar").

Çağırma: Sorgu ile ilişkili elde edilen dokümanların, ilişkili olan dokümanlara olan oranının yüzdelik ifadesidir.

5.1.6. Anahtar Kelime ve Benzerlik Tabanlı Bilgi Çıkartımı

"Bilgi çıkarımı için hangi metotlar bulunmaktadır?" Tüm bilgi çıkarım sistemleri anahtar kelime tabanlı ve/veya benzerlik tabanlı çıkarımı destekler. Anahtar kelime tabanlı bilgi çıkarımında, bir doküman anahtar kelimelerden oluşan bir dizgi ile temsil edilir. Kullanıcı anahtar bir kelime veya "araç ve tamirhaneler", "çay ve kahve", "Oracle'ın haricindeki veri tabanı sistemleri" gibi anahtar kelimelerden oluşan bir küme ifadesi sağlar. İyi bir bilgi çıkarım sistemi bu tür sorgularda eş anlamlı sözcükleri de dikkate almalıdır. Örneğin, "araba" kelimesi girildiğinde eş anlamlıları olan "araç" ve "otomobil" gibi kelimeleri de dikkate almalıdır. Anahtar kelime tabanlı sistem iki önemli zorlukla karşı karşıya gelen basit bir sistem modelidir. Bunların ilki eş anlam problemidir: örneğin "yazılım ürünü" gibi anahtar bir kelime doküman gerçekten bir yazılım ürünü ile ilişkili olsa dokümanın her hangi bir bölümünde bulunmayabilir. İkicisi ise, çokanlamlılıktır; aynı kelime içerik olarak farklı anlamlarda kullanılmış olabilir.

Benzerlik tabanlı çıkarım sistemleri ortak anahtar kelimeler kümesini temel alarak benzer dokümanları bulmaktadır. Bu tür bir çıkarımın çıktısı kelimelere

yakınlığı, kelimelerin bağıl frekanslarını temel alan bir ölçüm ile belirlenen ilişki derecesini temel almaktadır. Çoğu durumda, anahtar kelime kümeleri arasındaki ilişkinin derecesinin hassasiyet ölçümünü belirlemek zor olmaktadır.

"Anahtar kelime ve benzerlik tabanlı bilgi çıkarım sistemleri nasıl çalışmaktadır?". Bir metin çıkarım sistemi bir "dur listesi" ile bir doküman kümesini ilişkilendirir. Bir "dur listesi" bir kelime kümesini "konu ile ilişkisi olmayan" olarak addeder. Örneğin "bir", "nin", "için", "ile" gibi kelimeler sıklıkla karşılaşılmalarına rağmen "dur" kelimeleridir. Doküman kümeleri değiştikçe "dur" listeleri de değişmektedir. Örneğin veri tabanı sistemleri bir gazete içerisinde önemli bir kelime olabilir. Bununla beraber, veri tabanı sistemleri konferansında yayınlanan makaleler kümesi içerisinde bir "dur" kelimesi olarak değerlendirilebilir.

Farklı kelimelerden oluşan bir grup, aynı kelime gövdesini paylaşabilir. Bir metin çıkarım sistemi, bir grup içerisindeki kelimelerin diğer kelimelere olan küçük söz dizimsel değişimlerinden oluşan kelimeleri tanımlama ihtiyacı duyar ve her grup için ortak kelime gövdesini derler. Örneğin, "drug", "drugged" ve "drugs" kelime grubu, aynı "drug" kelime gövdesini paylaşmakta ve aynı kelimenin farklı bulunma durumlarını gösterebilmektedir.

"Bilgi çıkarımını gerçekleştirmek için bir dokümanı nasıl modelleyebiliriz?" Bir "d" doküman kümesi ve "t" terim kümesi ile başlayarak, her dokümanı "t" boyutlu "R^t" uzayında "v" vektörü ile modelleyebiliriz. "v" vektörünün "j." koordinatı verilen dokuman için j. terimin ilişkisini ölçen bir sayıdır: bu değer eğer dokuman terimi içermiyorsa genellikle 0, içeriyorsa sıfırdan farklıdır. Bu vektörde sıfırdan farlı girişler için terim ağırlıklandırma tanımlamanın farklı yolları bulunur. Örneğin, eğer j. terime dokuman içerisinde rastlanmış ise $v_j = 1$ olarak tanımlanır veya t_i teriminin dokuman içerisinde karşılaşılma sayısı direk olarak kullanılarak v_j terim frekansı, terimin karşılaşılma sayısının toplam terimlere oranı kullanılarak göreli frekans değeri olarak kullanılabilir.

Veri Madenciliği veya Veri tabanlarında Bilgi Keşfi, verilerdeki yeni ve anlaşılabilir biçimlerin tanımlanması işlemidir[47]. Veri inceleme yalnızca

enformasyon veya kullanıcının hâlihazırda sormayı bildiği sorulara yanıtlar aramakla kalmaz aynı zamanda veriler içerisine gömülmüş olan derin bilgileri de keşfeder. Bunu yapmak için veri inceleme işleminde hesaplama teknikleri kullanılır, bunlar genellikle bir öğrenme algoritması biçimindedir ve verideki potansiyel olarak yararlı biçimlerin bulunması amacını taşır. Mevcut veri inceleme yaklaşımlarının büyük bölümü verilerin ilişkisel bir tablosu içerisindeki biçimleri arar.

Metin inceleme veya metin verisi inceleme, yararlı veya ilginç biçimlerin, modellerin, yönlerin, eğilimlerin veya kuralların yapılandırılmamış metinden bulunması işlemi, veri inceleme tekniklerinin metinden bilginin otomatik olarak bulunması amaçlı veri inceleme tekniklerinin uygulanmasının açıklanması amacıyla kullanılır. Genellikle metin inceleme işlemine, veri incelemenin doğal bir uzantısı olarak bakılır [48]. Bu durum, metin incelemenin bulunmasının, büyük ölçüde veri incelemenin filizlendiği alanı temel alır.

Bununla birlikte, ya ilişkisel veri tabanlarında ya da veri depolarında mevcut olan iyi yapılandırılmış koleksiyonlar üzerinde odaklanan veri incelemeden farklı olarak, metin inceleme çok daha az yapılandırılmış olan verileri açığa çıkartır. Bugünün elektronik verilerinin büyük bölümü geleneksel ilişkisel veritabanlarında bulunmaz, bunlar Web'de ve doğal dilli dokümanlarda "gizlenmiştir". Bu çalışmada geleneksel veri inceleme ve Enformasyon Çıkartılmasının (IE) entegrasyonunu temel alan metin incelemesi için yapılan çalışmalardan söz edilecektir.

Bir IE sisteminin amacı doğal dilli metinler içerisindeki özel verilerin bulunmasıdır. Çıkartılacak olan veriler tipik olarak, dokümandan alınan alt dizilerle doldurulacak olan bir yuvalar listesi belirleyen bir şablonla verilir. IE bir dizi uygulama için yararlıdır, özellikle de son zamanlarda Internet'in ve web dokümanlarının çoğalması göz önüne alındığında. Yakın zamandaki uygulamalar kurs ve araştırma projesi ana sayfalarını, seminer duyurularını, daire kiralama ilanlarını, iş ilanlarını, coğrafi web dokümanlarını, hükümet raporlarını ve tıbbi özetleri kapsamaktadır [48].

Geleneksel veri inceleme işleminde "incelenecek" olan enformasyonun hâlihazırda bir ilişkisel veri tabanı biçiminde olduğu varsayılır. Ne yazık ki birçok uygulama için elektronik enformasyon yapılandırılmış veri tabanlarından çok, yalnızca yapılandırılmamış doğal dilli dokümanlar halindedir. IE metinsel dokümanların bir külliyatının daha yapılandırılmış bir veritabanına dönüştürülmesi sorununu hedef alır ve böylece standart VTBK yöntemleri ile birleştirildiğinde metin incelemesinde oynanabilecek olan açık bir rol ortaya koyar. Bu çalışmada, bir IE modülünün ham metin içerisindeki özel veri bölümlerinin konumlandırılması ve sonuçta ortaya çıkan veritabanının kural incelemesi için VTBK modülüne sağlanması amacıyla kullanımı anlatılmaktadır.

5.1.7. Metin Verilerinin Heterojenliği

İlişkisel veri tabanları ile karşılaştırıldığında, internet üzerinde mevcut olan doğal dilli çoğunlukla heterojen ve gürültülüdür. Birçok metinsel veri tabanı alanına yapılan girişler inceleme algoritmalarının önemli düzenlilikleri keşfetmesine engel olabilecek küçük farklılıklar gösterebilir. Farklılıklar tipografik hatalardan, yanlış yazımlardan, kısaltmalardan ve diğer kaynaklardan kaynaklanabilir.

Farklılıklar özellikle yapılandırılmamış veya yarı-yapılandırılmış dokümanlardan veya web sayfalardan otomatik olarak çıkartılan verilerde ifade edilir. Örneğin, haber grubu postalarından otomatik olarak çıkardığımız yerel iş olanakları konusundaki verilerde, Windows işletim sistemi değişik şekillerde "Microsoft Windows", "MS Windows", "Windows 95/98/ME" vb. şekillerde adlandırılmaktadır[48].

Daha önce yapılmış olan işlerin bir bölümü benzer veya çoğaltılmış kayıtların tanımlanması sorununu hedef almıştır, bu işlem kayıtların bağlantılandırılması, birleştirme/ayırma sorunu, çoğaltma algılaması yumuşak veri tabanlarının sertleştirilmesi ve referans uyumlandırması olarak adlandırılmıştır. Tipik olarak, sabit bir metinsel benzerlik ölçümü, iki değerin veya kaydın kopya olmak için yeterince benzer olup olmadığının belirlenmesinde kullanılmıştır. Bu yaklaşımda, "Microsoft Windows", "MS Windows" ve "Windows 95/98/ME" işlem öncesi bir basamak olarak tek bir terim içerisine haritalandırılmıştır [48].

Ataları ve ardılları veritabanı girişlerine yeterli benzerlik temelinde değerlendirilen "yumuşak uyumlandırma" kurallarının keşfedilmesi yoluyla "kirli" verilerin direkt olarak bulunması biçimindeki alternatif yöntemlerden ilerleyen sayfalarda anlatılacaktır. Metnin benzerliği standart "kelimeler çantası" ölçümleri kullanılarak veya düzenleme-mesafe ölçümleri kullanılarak ölçülebilir; diğer standart benzerlik ölçümleri nümerik ve ek veri türleri için kullanılabilir. Örneğin, "Windows bir iş için gerekli becerilerin listesiyse, o zaman bu iş için IIS bilgisi de gereklidir" gibi yumuşak uyumlandırma kuralları bir dizi iş ilanından keşfedilir. Bu durumda, "Windows" ve "IIS", sırasıyla "MS Windows" veya "IIS Hizmetleri" gibi benzer dizilere uyumlandırılabilir.

5.2. Metin Sınıflandırma

Sınıf olmak için her kaydın belli ortak özellikleri olması gerekir. Ortak özelliklere sahip olan kayıtların hangi özellikleriyle bu sınıfa girdiğini belirleyen algoritma, sınıflama algoritmasıdır. Sınıflama algoritması, denetimli öğrenme kategorisine giren bir öğrenme biçimidir. Denetimli öğrenme, öğrenme ve test verilerinin hem girdi hem de çıktıyı içerecek şekilde olan verileri kullanmasıdır.

Sınıflama sorgusuyla, bir kaydın önceden belirlenmiş bir sınıfa girmesi amaçlanmaktadır (Bolat 2003). Bir kaydın önceden belirlenmiş bir gruba girebilmesi için sınıflama algoritması ile öğrenme verileri kullanılarak hangi sınıfların var olduğu ve bu sınıflara girmek için bir kaydın hangi özelliklere sahip olması gerektiği otomatik olarak keşfedilir. Test verileriyle de bu öğrenmenin testi yapılarak ortaya çıkan kurallar optimum sayısına getirilir.

Sınıflama algoritmasının kullanım alanları sigorta risk analizi, banka kredi kartı sınıflaması, sahtecilik tespiti, vb. alanlardır.

Metin Sınıflandırma, eldeki sınıflardan birine ait olduğu bilinen bir dokümanın, hangi sınıfa girdiğinin bulunması işlemidir. Günlük hayatta bir gazete ya da bir kitap okunduğunda, bu metinlerde geçen olaylar daha önceden bilinen birtakım olaylara bağlanır. Bir konunun nasıl anlaşıldığı da bu bilgilerin kendi aralarında nasıl bağlandığına ve her konunun içine konduğu sınıflara bağlıdır.

Otomatik metin sınıflandırma işlemi de günlük hayattaki bu uygulamanın bilgisayar dünyasındaki karşılığıdır[49].

Metin süzme (MS), dokümanların sisteme girmesiyle birlikte denetlenmesi ve kullanıcı sorgusuna uygun olanların seçilmesi işlemidir. MS uygun/uygun olmayan şeklinde karar verirken aslında dokümanları belli sınıflara ayırır. Bu yüzden MS bir sınıflandırma işlemi olarak da görülebilir[34]. Bu bakımdan ele alındığında dosyaların veya elektronik mektupların konularına göre önceden belirlenmiş klasörlere taşınmasında, belirli bir konuya özgün çalışmalarda, konunun belirlenmesinde ve yapısal aramalarda da kullanılabilir.

Birçok alanda yeni metinlerin sınıflandırılmasında profesyonel insanlar rol alır. Metin sınıflandırma çok zaman ve paraya mal olan bir işlemdir. Bundan dolayı otomatik metin süzme ve sınıflandırma işlemlerinde hızla gelişen teknolojiye ve uygulamalara bir ilgi vardır. Bağlanım modelleri, en yakın komşu sınıflandırıcıları, karar ağaçları, Bayesian sınıflandırıcıları, destek yöney makineleri, kural öğrenme algoritmaları, ilgililik geri besleme ve yapay sinir ağları gibi pek çok istatistiksel, matematiksel ve otomatik öğrenme teknikleri bu ilgiden kaynaklanmıştır.

5.2.1. Metin Madenciliğinin Ön Aşamaları ve Sınıflama

İster Metin Madenciliği, ister metin erişimi olsun, tüm bu konulara ait tekniklerin kullandıkları ortak yöntemler vardır. Bu bölümde bu yöntemlerden bahsedilecektir.

5.2.1.1. Ayrıştırma

Metin veri madenciliğinde yapılan ilk işlem, karakter dizileri olan metinlerin öğrenme algoritmaları ve sınıflandırma işlemleri için uygun bir hale getirilmesidir. Bunun için ilk önce metindeki XML (EXtensible Markup Language) ve HTML (Hyper Text Markup Language) gibi her türlü etiket kelimesinin çıkarılması gerekir. Ardından harf olmayan karakterler boşluklarla yer değiştirir. Tek harfli sözcükler silinir. Bütün karakterler küçük harflere çevrilir [50].

5.2.1.2. Durdurma Kelimelerinin Çıkarılması

Önişlemle, kullanılacak sözcüklerin ortaya çıkmasından sonra, dokümanın içerisinde çokça geçen fakat kendi başlarına bir anlamları olmayan ve dokümanlara fazla anlam katmayan (ve, sonra, ile... gibi) durdurma kelimeleri çıkarılır. Durdurma kelimelerinin bilgi erişim sistemlerinde gerekli olmadığı, bu sistemlerle ilgili çalışmalarının ilk günlerinden beri bilinmektedir. Bu kelimelerle yapılacak herhangi bir sorgunun, eldeki veri kümesinin her elemanını sonuç olarak döndüreceğinden, bu kelimelerin ayrım yaptırma güçleri zayıftır. Ayrıca durdurma kelimeleri, dokümanlarda çok fazla yer tutarak sistemin hantallaşmasına neden olur. Bu kelimeler, her doküman kümesinde istatistiksel yöntemlerle bulunabilse de, genelde tek bir durdurma kelimesi listesi kullanılır. Bu liste bir adres hesaplama tablosunda (hash table) da tutulabilir.

5.2.1.3. Gövdeleme

Durdurma kelimelerinin çıkarılmasının ardından, her kelimenin eklerinin çıkarılmasıyla kelime kökleri bulunur. Kelime köklerinin bulunması, kelimelerin biçimsel benzerlerinin bulunması anlamına gelir. Böylece, koşucular, koşucu, koşmak, koş, koşuyorum gibi aynı anlam grubundaki kelimeler bir araya getirilmiş olur. Kök bulmada karşılaşılabilecek iki sorun vardır: Birincisi, bu işlemde çok ileri giderek birbirinden anlamca çok farklı kelimelerin aynı anlam grubuna bağlanmasıdır. Bu durumda sistem, konuya uygun olmayan dokümanları da konuyla ilgili şeklinde yorumlayabilir. Diğer bir sorun da, kelimelerin köklerine ulaşılmaya çalışılırken çok az ekin çıkarılması işlemidir. Bu durumda da sistem konuya uygun dokümanları, "uygun olmayan" dokümanlar olarak algılayabilir.

Gövdelemeye yarayan pek çok farklı algoritma vardır. Bu yöntemlerden biri tüm dizin sözcüklerinin ve köklerinin Tablo 3.1.' deki gibi bir tabloda tutulmasıdır.

Gizlemek	Gizle
Gizlenmek	Gizle
Gizle	Gizle

Şekil.15. Kelimelerin ve Köklerinin Bir Tabloda Tutulması[49]

Bu yöntemin dezavantajı, çok fazla saklama alanına gereksinim duyması ve böyle bir tablonun yaratılmasının zor olmasıdır.

Diğer bir yöntem de, eldeki dokümanlardan oluşturulan bir sözlüğün içindeki her kelimenin, her harfinin tek tek ele alınarak ardıl farklılıklarının incelenmesiyle yapılır. Kökü bulunacak kelimenin sözlük içinde farklı bir kelime olarak bulunabilen ilk n harfi, kelimenin kökü olarak alınır. Mesela sözlüğün içerisinde koş ve koşucu kelimeleri olsun. Koşucu kelimesinin kökünü bulmak için, k, ko, koş kelimelerine ulaşılır. Koş sözcüğünün sözlükte bir kelime olarak görülmesiyle kelimenin kökü bulunmuş olur.

Yukarıdaki yöntemler her dil için geçerli olan yöntemlerdir. Veri kümesi İngilizce metinlerden oluşan çalışmalarda, Porter Stemmer algoritması, daha basit ve hızlı olmasına rağmen diğerleriyle performans bakımından farkı olmaması nedeniyle, bu konu için en çok kullanılan algoritmadır[49].

5.2.1.4. Metin Gösterimi

Metinler sayısal ortamlarda saklanırken, en çok, doğal yazının sayısal ortamdaki şekli halinde bulunur. Fakat metin halinde depolanan dokümanların üzerinde hesaplamaya dayanan işlemler yapmak zor olduğu için, dokümanlar farklı gösterim şekillerine dönüştürülür. Aşağıda bu gösterim şekillerinden birisi olan vektör uzayı modeli açıklanmıştır.

5.2.1.5. Vektör Uzayı Modeli

Bu konudaki en çok bilinen yöntem vektör uzayı modelidir. Bu modele sahip bir dokümanlar kümesinde, her doküman *MxN* kelime vektörleriyle ifade edilir. *M* tüm dokümanlardaki her bir farklı kelime ve *N* de elde bulunan tüm

dokümanların sayısıdır. Bu vektördeki her girdi, bir kelimenin o dokümandaki kullanılma sıklığını ifade eder.

5.2.1.6. Boyut Küçültme

Her kelime, her dokümanda geçmediği için, yukarıda *A* ile gösterilen matris genellikle seyrek matristir. Matristeki satır sayısı *M*, sözlükteki kelime sayısına eşit olduğu için *M* çok büyük bir sayı olabilir. Bu da matrisin büyümesine ve işlemler sırasında gereksiz zaman ve iş kaybı anlamına gelir. Bu problemi aşmak için farklı algoritmalar uygulanabilir [49].

5.2.1.7. Yeniden Değiştirgeleme

Yeniden değiştirgeleme, eldeki özelliklerin yeniden yapılandırılması veya birleştirilmesiyle yeni özellikler yaratılmasına dayanır. Bu yöntemde, kelimelerin arasında gizli bir ilişki olduğu kabul edilir ve bu ilişkiyi ortaya çıkarmak için, Gizli Anlambilimsel Dizinleme (Latent Semantic Indexing) gibi birtakım istatistiksel yöntemler kullanılır.

5.3. Ağırlıklandırma

Yukarıda belirtilen *A* matrisinin taşıdığı ağırlık değerlerinin belirlenmesinde pek çok yöntem kullanılır.Fakat bu yöntemlerin hemen hemen hepsi iki önemli noktaya dayanır;

Bir sözcük, bir dokümanın içinde ne kadar çok sayıda geçerse, o dokümanın bir kategoriye atanmasında o kadar etkili olur. Bir sözcük ne kadar çok farklı dokümanda bulunursa, o sözcüğün ayırt edici özelliği o kadar azdır.

5.4. Metin Madenciliği Algoritmaları

Metinler vektör uzayına geçirilip gerekli ağırlık değerleri değişikliklerinin yapılmasının ardından, artık üzerlerinde Metin Madenciliği algoritmaları kullanılabilir hale gelirler. Bu aşamada daha önce kullanılmış birkaç algoritma açıklanacaktır.

5.4.1. Rocchio Algoritması

Rocchio yönteminde, her kategori için, o kategoriye ait eğitim örneklerinin ortalaması alınarak prototip bir doküman vektörü oluşturulur. Hangi kategoriye ait olduğu bulunmaya çalışılan dokümanın, oluşturulan prototipe olan mesafesine bakılarak süzme işlemi gerçekleştirilir. Bu oldukça hızlı bir şekilde eğitilebilen ve pek çok türevi olan bir yöntemdir. Bu tez çalışmasında kullanılan EHİ algoritması da, Rocchio algoritmasının türevlerinden birisidir.

5.4.2. Naive Bayes

Naive Bayes yöntemi, bir dokümanın içindeki özellikleri birbirinden bağımsız düşünerek çalışır. Yani bir dokümanın sözcüklerinin birbirleriyle olan kombinasyonları, Naive Bayes yönteminde önemli değildir. Bu bağımsızlık fikri her ne kadar doğru değilmiş gibi görünse de, Naive Bayes büyük bir doğruluk oranı gösterir. Naive Bayes, eğitim kümesi verilerinin ve yeni girilen dokümanın verilerinin her birini tek tek kullanarak, yeni dokümandaki her sözcüğün kategoriyi etkileme ihtimallerini hesaplayarak tahminde bulunmaya çalışır. Naive Bayes formülü olasılıklara dayanır[49].

5.4.3. Karar Ağacı

Bu yöntemde doküman vektörü *d,* eğitim kümesi dokümanlarıyla oluşturulan bir karar ağacıyla karşılaştırılarak, kullanıcı için uygun ya da uygun olmadığı anlaşılır. Bu karar ağacının oluşturulmasında farklı algoritmalar kullanılsa da, bu ağacın her yaprağı farklı bir kategoriyi temsil eder. Kullanılan her algoritmanın amacı, yeni bir dokümanı en doğru biçimde bir kategoriye atayabilecek karar ağacını oluşturmaktır[51]. Aşağıda bu yöntemlerin en popülerlerinden birisi olan, CART yöntemi açıklanacaktır.

5.4.3.1. Ağacı Oluşturma (CART)

CART, ikili karar ağaçları oluştururken eğitimde kullanılan her bir vektörü, içindeki elemanlarından birini kullanarak, bir fonksiyon yardımıyla, ikiye ayırır. Bu yüzden, ilk karar verilmesi gereken, hangi elemanın en iyi ayrıştırıcı

olduğunun saptanmasıdır. En iyi ayrıştırıcı, kümeyi en türdeş biçimde ayırabilen ayrıştırıcıdır. Dolayısıyla eğitim kümesindeki çeşitlemeyi en aza indirebilen ayrıştırıcı, en iyi ayrıştırıcıdır.

5.4.3.2. Ağacın Budanması

Eğitim kümesini kullanırken hata oranı en aza indirilmiş olsa da, yeni gelen verilerin kategorilere atanmasında en iyi sonucu vermeyebilir. Ağaç tamamen eğitim kümesi elemanlarıyla örtüştüğü için, ağacın yeni verilere uygun olması budanma ile sağlanır.

5.4.4. Destek Yöney Makineleri

Destek yöney makineleri, metin sınıflamada olduğu kadar diğer pek çok geniş alanda da başarı göstermiştir. Vladimir Vapnik'in verilerin dağılımıyla ilgili olan yapısal risk enküçültmesi teorisine dayanır. Destek yöney makineleri yöntemi sadece ikili sınıflamalar yapabildiği için bütün sistem çok sayıdaki ikili kararların birleşmesinden oluşur[49].

5.4.5. Bayesian Ağları

Bayesian Ağları, pek çok değişken ve çok sayıdaki olasılığın geçerli olduğu bir uzayın yoğunlaştırılmasıyla ilgilidir. Yönlendirilmiş çevrimsiz çizge (directed acyclic graph) (DAG) ile ilişkiler tanımlanır. Her X_i ,ağın içerisinde bir boğum olarak gösterilir. Her boğum özellik yay ise iki özelliğin birbirleriyle olasılıksal bağımlılıklarını gösterir. Yani iki boğum arasında bir yay olmaması, bu iki boğumun birbirinden bağımsız olduğu anlamına gelir. Boğumlar, sadece alt boğumlarıyla ve bir yayla bağlandıkları boğumlarla ilişkilendirilebilir. Her boğum kendisinin üstündeki (X_i) boğumda kendisi için saklanan olasılık değerlerini alır. Üst boğuma sahip olmayan boğumlar, sadece altlarındaki Xi boğumları için önsel olasılık dağılımlarına sahiptir.

Bayes ağları, dokümanlar için düşünülecek olursa, dokümandaki her terim için bir ikili değer verilerek dokümanın içinde hangi kelimelerin geçtiği ve

hangilerinin geçmediği hakkında bir bilgi tutulabilir. Diğer bir deyişle; Bayes ağının içindeki tüm boğumlar bir vektörün içinde toplanmış olur. 6 terime sahip bir dokümanın Bayes ağ yapısı örneği gösterilmiştir. Bu örnekte, görülmesi beklenen kelimeler arasındaki bağımlılık olasılıkları da belirtilmiştir. Eğer bu 6 boğumun aralarındaki ilişkiler ele alınmazsa elimizde 2^6 =64-1=63 tane kelimeler arası bağıntı olasılığı olur.

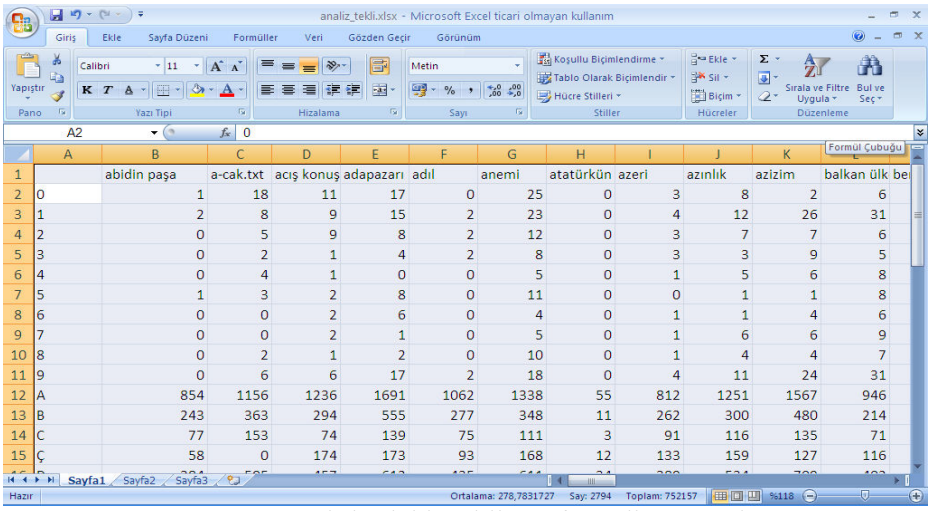
6. UYGULAMA

6.1. Uygulama

Metin madenciliği, Türkçe metinlerde en çok kullanılan harf ve birbirini izleyen harflerin ortaya çıkarılması için de kullanılır. Bu uygulamada Türkçe metinlerde en sık kullanılan harfler, iki harfin birbirini takip etme sıklığı ve üç harfin ardı ardına gelme sıklıkları belirlenmiştir. Ayrıca Türkçede kullanılan harflerin yanı sıra X, W, Q gibi harfler ve sayılar da kullanılmıştır. Türkçe konulardaki metinlerin içerdiği harf sıklıklarının değişip değişmediği saptamak için internet üzerinden Türk Dil Kurumu sitesinden yararlanılarak MS Office veritabanı ile MS Visual Studio ortamında C # .Net platformu ile apriori algoritması yapılmıştır. Her bir dosya kendi içinde incelenip çalışma sonucunda karşılaştırma yapılarak tesadüfî hata en aza indirilmiştir. Apriori algoritması orta düzey veri içeren hareketsel veritabanları için uygundur. AIS ve SETM algoritmasına oranla performans açısından daha iyi olduğu için apriori algoritması kullanılmıştır.

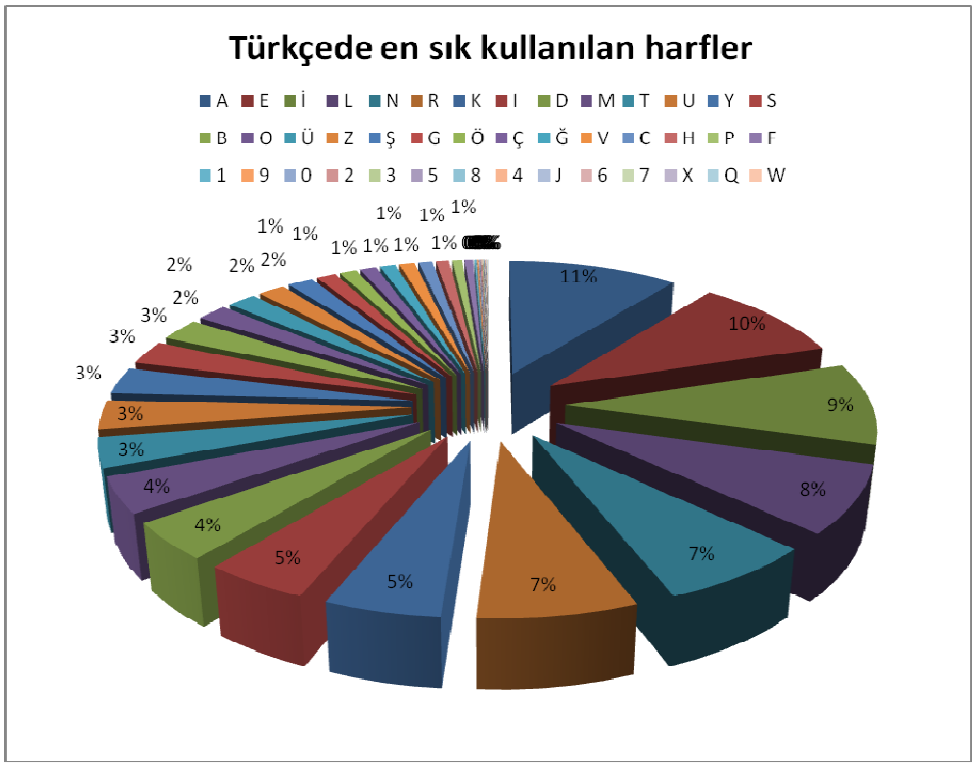
6.2. Türkçede En Sık Kullanılan Harfler

Microsoft Excel formatında 26 KB büyüklüğündeki veride 118.553 karakterden oluşan metinlerin taranmasıyla yapılan çalışmada Türkçe metinlerde en sık rastlanan harfler aşağıdaki gösterilmiştir.

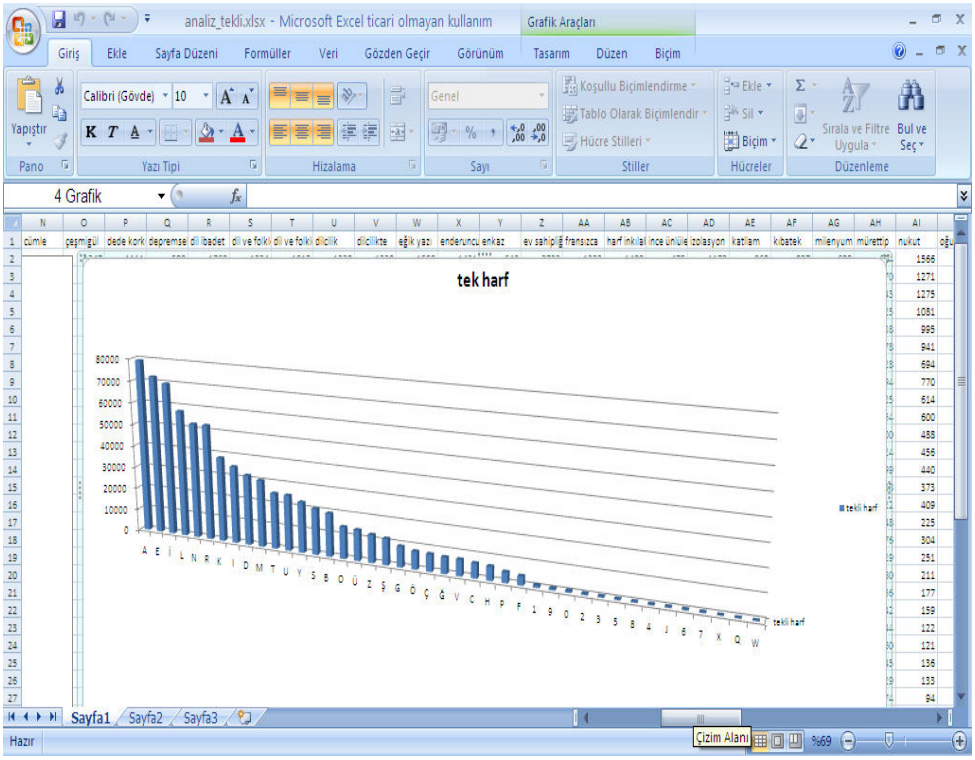


	A	B	C	D	E	F	G	H	I	J	K	L
1		abidin paşa	a-cak.txt	acış konuş	adapazarı	adıl	anemi	atatürkün	azeri	azınlık	azizim	balkan ülk
2	0	1	18	11	17	0	25	0	3	8	2	6
3	1	2	8	9	15	2	23	0	4	12	26	31
4	2	0	5	9	8	2	12	0	3	7	7	6
5	3	0	2	1	4	2	8	0	3	3	9	5
6	4	0	4	1	0	0	5	0	1	5	6	8
7	5	1	3	2	8	0	11	0	0	1	1	8
8	6	0	0	2	6	0	4	0	1	1	4	6
9	7	0	0	2	1	0	5	0	1	6	6	9
10	8	0	2	1	2	0	10	0	1	4	4	7
11	9	0	6	6	17	2	18	0	4	11	24	31
12	A	854	1156	1236	1691	1062	1338	55	812	1251	1567	946
13	B	243	363	294	555	277	348	11	262	300	480	214
14	C	77	153	74	139	75	111	3	91	116	135	71
15	Ç	58	0	174	173	93	168	12	133	159	127	116

Şekil.16. Metinlerdeki Tekli Harf Analiz Sonuçları



Şekil.17. Türkçe Metinlerde En Sık Kullanılan Harfler

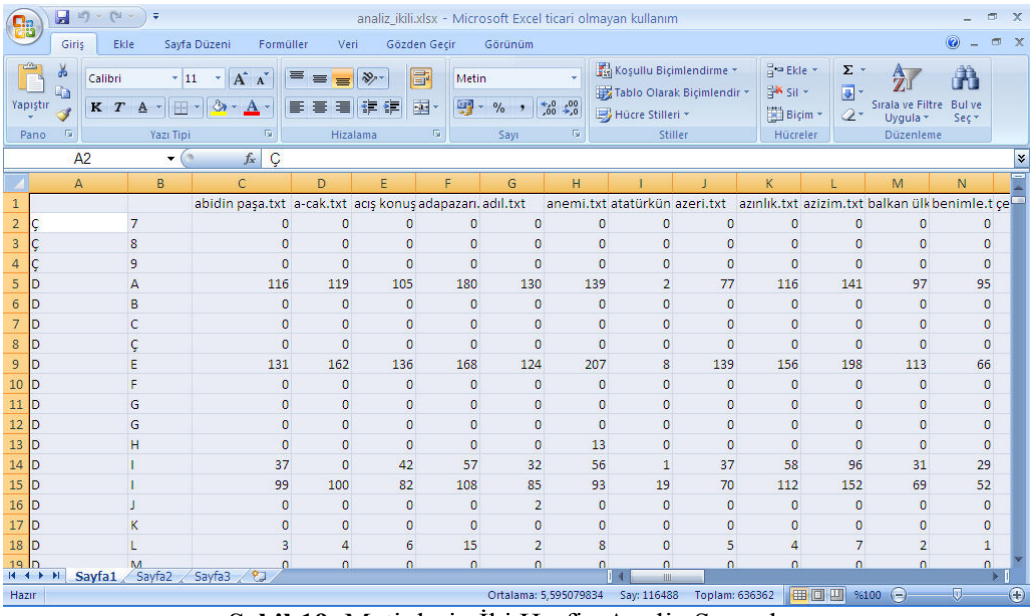


Şekil.18. Türkçe Metinlerde En Sık Kullanılan Harfler

Türkçede en sık rastlanan 6 harf A, E, İ, L, N ve R harfi olmuştur. R ve N harfleri yüzdeliklerine bakıldığında hemen hemen aynı durumdadır. Türkçe metinlerde en az kullanılan harf J olmuştur. J harfini sırasıyla J, F, P, Ç harfleri izlemektedir.

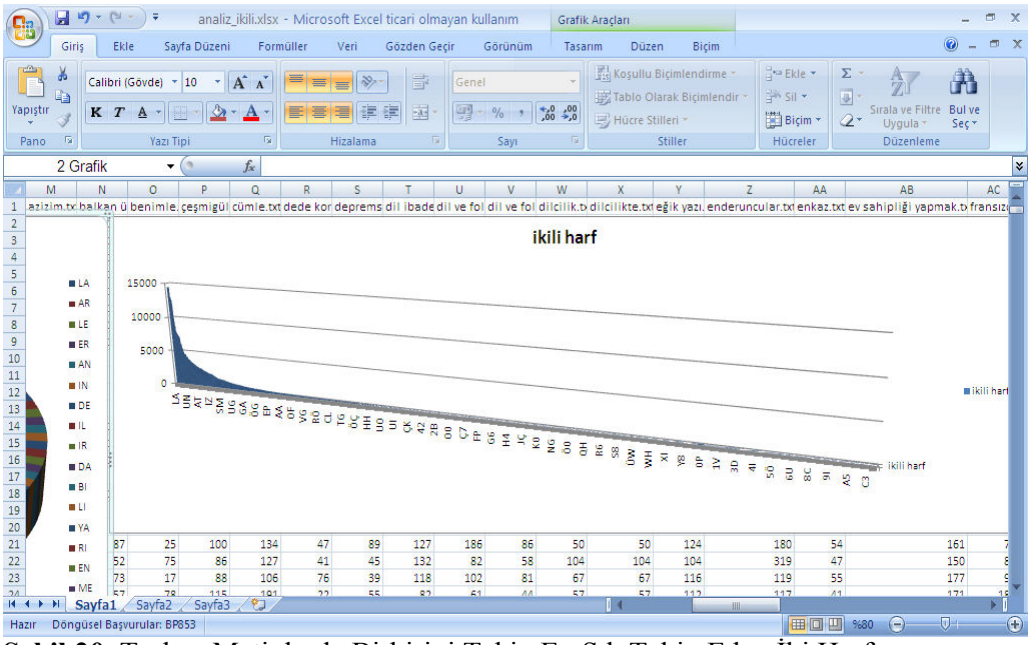
6.3. İki Harfin Birbirini Takip Etme Sıklığı

Harflerin tek tek sayılmasından sonra ikinci aşamada Türkçede hangi iki harfin birbirini en çok takip ettiğinin belirlenmesi çalışması yapılmıştır. MS Excel formatında gösterilen 476 KB büyüklüğündeki veride 118.553 karakterden oluşan metinlerin taranmasıyla yapılan çalışma aşağıda gösterilmiştir.



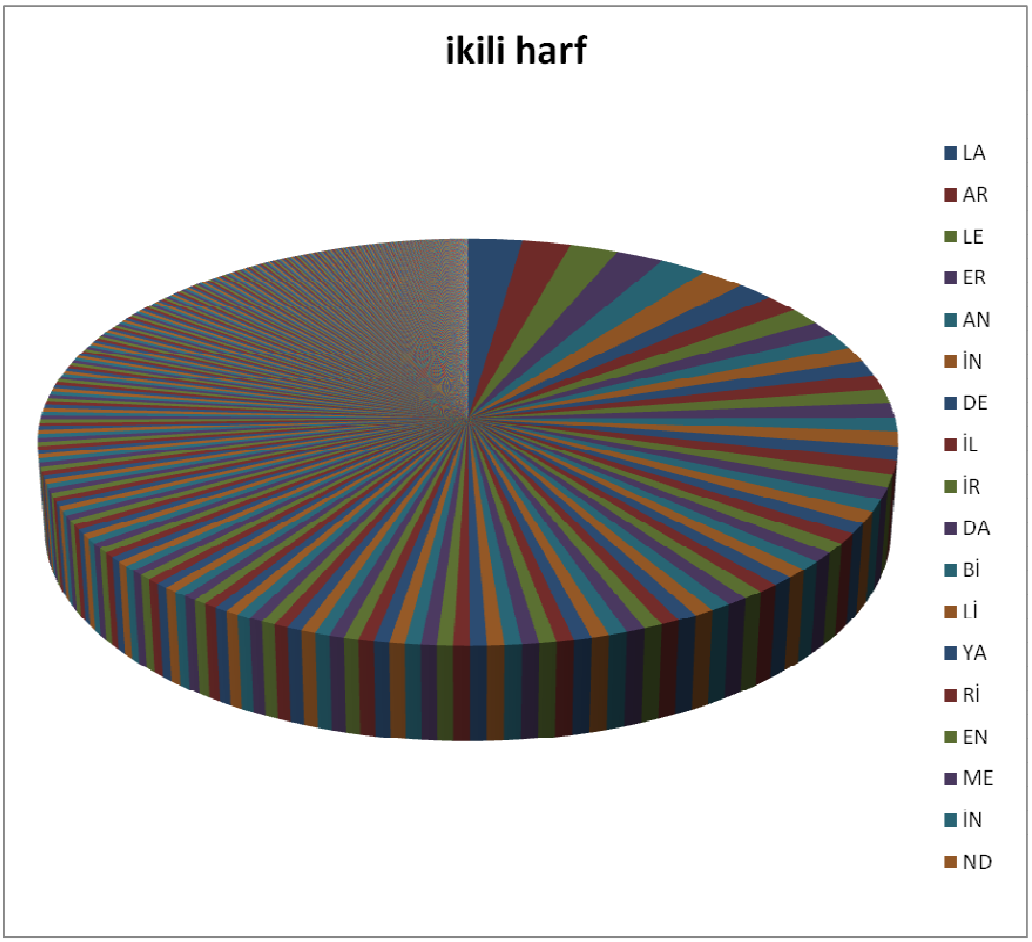
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1			abidin paşa.txt	a-cak.txt	acış konuş	adapazarı.	adıl.txt	anemi.txt	atatürkün	azeri.txt	azınlık.txt	azizim.txt	balkan ülk	benimle.t
2	Ç	7	0	0	0	0	0	0	0	0	0	0	0	0
3	Ç	8	0	0	0	0	0	0	0	0	0	0	0	0
4	Ç	9	0	0	0	0	0	0	0	0	0	0	0	0
5	D	A	116	119	105	180	130	139	2	77	116	141	97	95
6	D	B	0	0	0	0	0	0	0	0	0	0	0	0
7	D	C	0	0	0	0	0	0	0	0	0	0	0	0
8	D	Ç	0	0	0	0	0	0	0	0	0	0	0	0
9	D	E	131	162	136	168	124	207	8	139	156	198	113	66
10	D	F	0	0	0	0	0	0	0	0	0	0	0	0
11	D	G	0	0	0	0	0	0	0	0	0	0	0	0
12	D	G	0	0	0	0	0	0	0	0	0	0	0	0
13	D	H	0	0	0	0	0	13	0	0	0	0	0	0
14	D	I	37	0	42	57	32	56	1	37	58	96	31	29
15	D	I	99	100	82	108	85	93	19	70	112	152	69	52
16	D	J	0	0	0	0	2	0	0	0	0	0	0	0
17	D	K	0	0	0	0	0	0	0	0	0	0	0	0
18	D	L	3	4	6	15	2	8	0	5	4	7	2	1
19	D	M	0	0	0	0	0	0	0	0	0	0	0	0

Şekil.19. Metinlerin İki Harfin Analiz Sonuçları



Şekil.20. Türkçe Metinlerde Birbirini Takip En Sık Takip Eden İki Harf

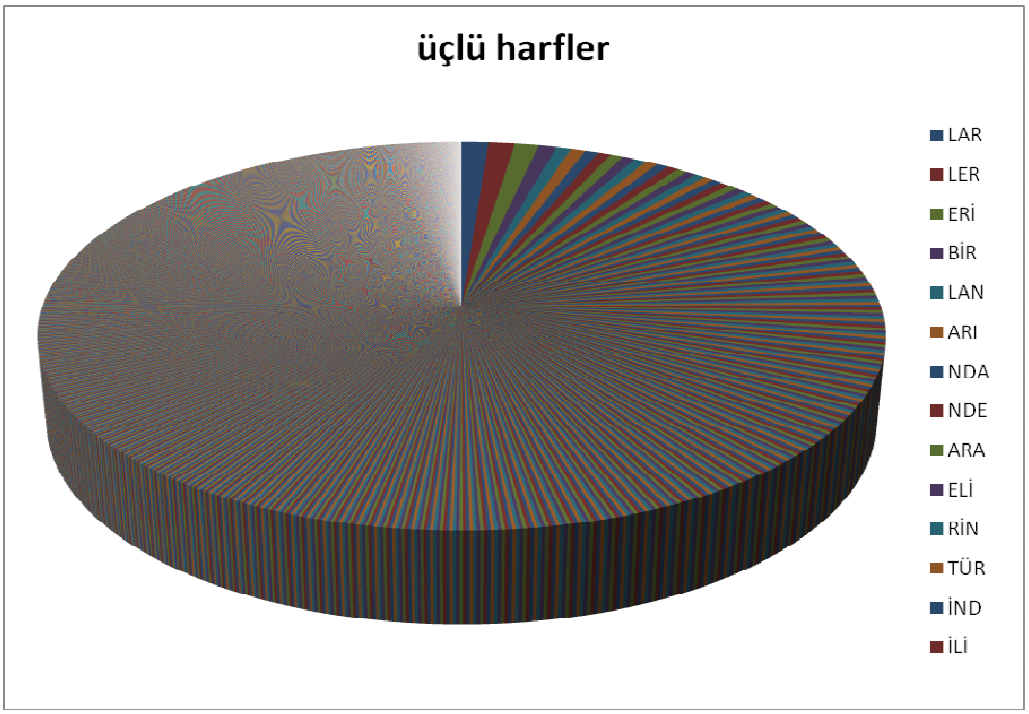
Türkçe metinlerde birbirini en sık takip eden ikili gruplar yukarıdaki şekilde gösterilmiştir. Uygulama sonucu en sık takip eden ikili harfler, ‘L-A, A-R, L-E, E-R, A-N’ dır.



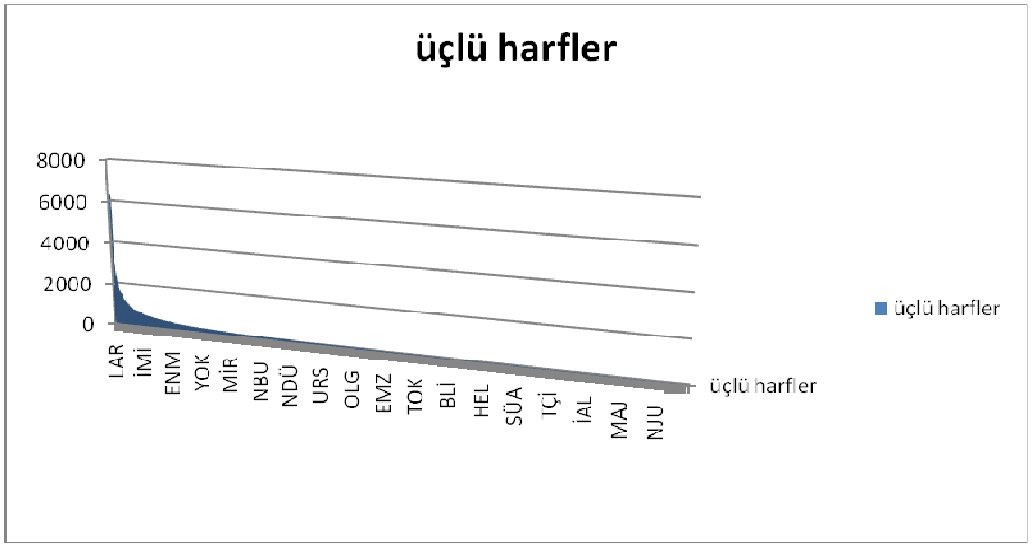
Şekil.21. Türkçe Metinlerde İki Harfin Birbirini Takip Etme Sıklığı

6.4. Türkçe Üç harfin Birbirini Takip Etme Sıklığı

Türkçe metinlerdeki üç harfin birbirini takip etme sıklıkları L-A-R, L-E-R, E-R-İ, B-İ-R,L-A-N'dır. Bu bilgiler aşağıdaki şekilde mevcuttur.



Şekil.22. Türkçe Metinlerde Üç Harfin Birbirini Takip Etme Sıklığı



Şekil.23. Türkçe Metinlerde En Sık Yan Yana Gelen Üç Harf

6.5 Uygulamanın Çalıştırılması

Girilen metin tek harf, iki harf ve üç harf üzerinden olasılıkları hesaplanarak uygulanır.

Boşluk karakterleri ayıklama işlemi aşağıdaki gösterilmiştir.

```
text = text.Trim();
```

Türkçe karakter için yer değiştirme (replace) fonksiyonu kullanılarak büyük harfe(upper) dönüştürme işlemi aşağıda gösterilmiştir.

```
text = text.Replace("ı", "I");
text = text.Replace("i", "İ");
text = text.Replace("ğ", "Ğ");
text = text.Replace("ü", "Ü");
text = text.Replace("ö", "Ö");
text = text.Replace("ş", "Ş");
```

Büyük harflere çevirme işlemini ise ;

```
text = text.ToUpper();
```

Türkçe harfler, sayılar ve X, Q ve W gibi harflerin eklendiği kısım;

```
string gecerli = "ABCÇDEFGĞHIİJKLMNOÖPQRSŞTUÜVWXYZ0123456789";
for (int i = 0; i < text.Length; i++)
{
   if (gecerli.IndexOf(text[i]) == -1)
   {
      text = text.Replace(text[i].ToString(), " ");
   }
}
```

Her üç olasılık hesaplama için öncelikle veritabanındaki olasılıklar sıfırlanıp işlemlerimizi gerçekleştiriyoruz.

6.6. Türkçede En Sık Kullanılan Harflerin Program Tanımı

Girilen metin sonuna kadar döngüde kalır. Sıradaki her harf için Microsoft Access veritabanında TEKLI tablosunda olasılık kolonu bir arttırılarak kaydedilir. Fonksiyon:

```
private void tekliHarf(string girdi)
    {
```

Fonksiyon girdiğimiz metin "ab" olsun. "a"yı ve "b"yi işleyeceğiz. Döngü sayısı uzunluk (Lenght) olur.

…

Veritabanı kodları…

Veritabanı komutu için "cmdText" değişkeni kullanılıyor.

```
string cmdText = "";
```

Olasılık için "ihtimal" değişkeni kullanılıyor.

```
int ihtimal = 0;
```

,

Döngü buradan başlar.

```
for (int i = 0; i < girdi.Length; i++)
{
```

…

Tekli tablosundan döngü sırasındaki harfin olasılığı sorgulanıp çalıştırılır.

```
cmdText = "select OLASILIK from TEKLI WHERE HARF = '" + girdi[i] + "'";
```

…

Sorgu sonucu dönen değeri tamsayıya çevirip ihtimal değişkenine atılıp, değişken bir artırılır.

```
ihtimal = Int32.Parse(dt.Rows[0][0].ToString());
            ihtimal++;
```

…

```
cmdText = "UPDATE TEKLI SET OLASILIK = " + ihtimal + " WHERE HARF='"
+ girdi[i] + "'";
```

…

Olasılığı bir arttırılan harf için Tekli tablosunda güncelleme komutu yazılıp çalıştırılıyor.

6.7. Türkçede En Sık Kullanılan İki Harfin Birbirini Takip Etme Program Tanımı

Girilen metin sonuna kadar döngüde kalır. Sıradaki her ikili harf için Microsoft Access veritabanında IKILI tablosunda olasılık kolonu bir arttırılarak kaydedilir. Fonksiyon:

```
private void ikiliHarf(string girdi)
    {
```

Fonksiyon Girdiğimiz metin “abc” olsun. “ab”yı ve “bc”yi işleyeceğiz. Döngü sayısı uzunluğun bir eksiği (Lenght-1) olur.

…

Veritabanı kodları…

Veritabanı komutu için “cmdText” değişkeni kullanıyoruz.

```
string cmdText = "";
```

Olasılık için “ihtimal” değişkeni kullanıyoruz.

```
int ihtimal = 0;
```

,

Döngü buradan başlıyor.

```
for (int i = 0; i < girdi.Length-1; i++)
{
```

…

İkili tablosundan döngü sırasındaki ikili harfin olasılığını sorgulayıp çalıştırılır.

```
cmdText = "select OLASILIK from IKILI WHERE HARF1 = '" + girdi[i] + "' AND HARF2 = '" + girdi[i + 1] + "'";
```

…

Sorgu sonucu dönen değeri tamsayıya çevirip (Int32.Parse) ihtimal değişkenine atılıyor. Değişken aşağıdaki gibi bir artılıyor.

```
ihtimal = Int32.Parse(dt.Rows[0][0].ToString());
            ihtimal++;
cmdText = "UPDATE IKILI SET OLASILIK = "+ ihtimal +" WHERE HARF1='" +
girdi[i] + "' AND HARF2='" + girdi[i + 1] + "'"
```

Olasılığı bir arttırılan harf için İkili tablosunda güncelleme komutu yazılıp çalıştırılır.

6.8. Türkçede En Sık Kullanılan Üçlü Harfin Program Tanımı

Girilen metin sonuna kadar döngüde kalır. Sıradaki her harf için Microsoft Access veritabanında UCLU tablosunda olasılık kolonu bir arttırılarak kaydedilir. Fonksiyon:

```
private void ucluHarf(string girdi)
    {
```

Fonksiyon Girdiğimiz metin “abcd” olsun. “abc”yı ve “bcd”yi işleyeceğiz. Döngü sayısı uzunluğun iki eksiği (Lenght-2) olur.

…

Veritabanı kodları…

Veritabanı komutu için “cmdText” değişkeni yine kullanılıyor..

```
        string cmdText = "";
```

Olasılık için “ihtimal” değişkeni kullanılıyor.

```
        int ihtimal = 0;
```

,

Döngü buradan başlıyor.

```
        for (int i = 0; i < girdi.Length-2; i++)
        {
```

…

Üçlü tablosundan döngü sırasındaki üç harfin olasılığını sorgulayıp çalıştırılıyor.

```
cmdText = "select OLASILIK from UCLU WHERE HARF1 = '" + girdi[i] + "' AND
HARF2 = '" + girdi[i + 1] + "' AND HARF3 = '" + girdi[i + 2] + "'";
```

…

Sorgu sonucu dönen değeri tamsayıya çevirip ihtimal değişkenine atıp, değişkeni bir arttırıyor.

```
ihtimal = Int32.Parse(dt.Rows[0][0].ToString());
                ihtimal++;
…
cmdText = "UPDATE IKILI SET OLASILIK = "+ ihtimal +" WHERE HARF1='" + girdi[i] + "' AND HARF2='" + girdi[i + 1] + "'";
…
```

Olasılığı bir arttırılan harf için Üçlü tablosunda güncelleme komutu yazılıp çalıştırıyor.

7. SONUÇ VE ÖNERİLER

7.1. GENEL SONUÇLAR

Bu bölümde araştırma kapsamında gerçekleştirilen uygulamanın istatistiksel analizi ile elde edilen sonuçlar özetlenmekte ve aşağıda belirtilen araştırma soruları cevaplanmakta, ayrıca uygulamaya yönelik öneriler ve gelecek araştırmalar için de öneriler değerlendirilmektedir.

Bu tez çalışmasında kullandığımız metinlerin tamamı internetten elde edilmiştir. Bunun sebebi ise doğal bir şekilde yazılmış metinlerin internet kaynaklarından daha kolay ulaşılabilecek başka bir kaynağın olmamasıdır. İlk olarak Türkçe metinlerde en sık kullanılan harfler incelenmiştir. İkinci olarak Türkçe metinlerde iki harfin birbirini takip etme sıklığı incelenmiştir. En son olarak Türkçe metinlerde üç harfin birbirini takip etme sıklığı incelenmiştir. Bu tezde kısaca aşağıdaki sorulara cevap aranmaktadır:

1. Türkçe metinlerde en sık kullanılan harfler ve en az kullanılan harf nedir?

2. Türkçe metinlerde iki harfin birbirini takip etme sıklığı nedir?

3. Türkçe metinlerde üç harfin birbirini takip etme sıklığı nedir?

Bu tez çalışmasında toplam 118.553 karakterden oluşan Türkçe metin taramasında en sık rastlanan harfin A harfi olduğu görülmüştür. Türkçe metinlerde en sık kullanılan harfler A, E, İ, L, N ve R 'dir. Toplamda en az kullanılan harf J harfidir.

Türkçe metinlerde birbirini en sık takip eden ikili gruplar, 'L-A, A-R, L-E, E-R, A-N' dır. Birbirini en sık takip eden iki harf L ve A harfidir. Birbirini en çok takip eden harfler Türkçe de en çok kullanılan ilk 6 harf içinden çıkmıştır ve bu da doğaldır. Türkçe metinlerde ikili kullanım açısından bir ünlü bir ünsüzü ya da bir ünsüz bir ünlü takip etmektedir. İki ünsüzün birbirini takip etme sıklığı çok azdır.

Türkçe metinlerde birbirini en sık takip eden üçlü gruplar, ‘L-A-N, A-R-İ, N-D-A, N-D-E, A-R-A, E-L-İ’ dır. Birbirini en çok takip eden harfler yine Türkçe de en çok kullanılan 6 harften çıkmıştır.

Bu tez çalışması Türkçe metinlerdeki harfler hakkında istatistiki bilgi sağlayacaktır. Bu çalışma sadece Türkçe metinlere özgü bir metot değildir. Bu çalışma başka uygulama alanlarında metin türlerini gruplayarak (hukuk, tarih, ekonomi, teknoloji, spor vs) kullanmamızda ümit verici olabilir. Bu alandaki çalışmalar hem teorik hem yönetsel anlamda önemli katkılar sağlayacaktır.

Kelimelerdeki harf analizlerinin arama motorlarında kullanılan yöntemleri iyileştirmek için kullanılması öngörülebilir. Bu çalışma ile Türk kullanıcısına en uygun klavye seçimi de yapılabilir. En çok kullanılan harflerin işlevsel parmakların en kolay ulaşılabileceği noktalar yerleştirilmesi dışında birbirini en çok izleyen harflerinde belirlenip klavyenin buna göre oluşturulmasında faydalı olabilir.

Bu uygulama farklı çalışma koşulları ve yöntemler içeren algoritmaların, farklı veri yapıları üzerinde çalıştırılarak performans, ölçümleri yapılabilir.

KAYNAKLAR

[1] http://mail.baskent.edu.tr/~20394676/0302/bil483/HW2.pdf

[2] Han, J. ve Kamber, M., (2000), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, Burnaby.

[3] Alpaydın, E., (2000), "Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri", Bilişim 2000 Eğitim Semineri, 1-10.

[4] Abdullah Baykal,"Veri Madenciliğinde Market Sepet Analizi ve Birliktelik Kurallarının Belirlenmesi" Yıldız Teknik Üniversitesi Yüksek Lisans Tezi

[5] Tantuğ, A. C., (2002), Veri Madenciliği ve Demetleme, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[6] "Tıpta Veri Ambarları Oluşturma ve Veri Madenciliği Uygulamaları", Selçuk Üniversitesi

[7] Aydogan, F., 2003. E-Ticarette Veri Madenciligi Yaklasımlarıyla Müsteriye Hizmet Sunan Akıllı Modüllerin Tasarımı ve Gerçeklestirimi. H.Ü. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 179s, Ankara.

[8] Frawley, W. J., Piatetsky-Shapiro, G., Matheus, C. J., 1991. Knowledge Discovery Databases: An Overview, in Knowledge Discovery in Databases. Cambridge, MA: AAAI/M1T, 1-27.

[9] Akpınar, H. , (2000), "Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği", İstanbul Üniversitesi İşletme Fakültesi Dergisi, Nisan 2000 Sayısı, C: 29.

[10] Akbulut, S., 2006. Veri Madenciligi Teknikleri ile Bir Kozmetik Markanın Ayrılan Müsteri Analizi ve Müsteri Segmentasyonu. G.Ü. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 91s, Ankara.

[11] Kalıkov, A., 2006. Veri Madenciligi ve Bir E-Ticaret Uygulaması. G.Ü. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 94s, Ankara.

[12] David L. Olson Dursun Delen, Advanced Data Mining Techniques p-11- 53

[13] http://www.microsoft.com.tr/download/sql/datasheets/SQLDataMiningDatasheet.pdf

[14] www3.itu.edu.tr/~sgunduz/courses/verimaden/

[15] Huberty, C., 1994. Applied Discriminant Analysis. John Wiley & Sons Inc., Canada, 25-35.

[16] Hudairy, H., 2004. Data Mining and Decision Making Support in The Governmental Sector. M. Sc. Thesis, Faculty of Graduate School of The University of Louisville, 100p, Kentucky.

[17] Wei, C., Chiu T., 2002. Turning Telecommunications Call Details to ChurnPrediction: A Data Mining Approach. Expert Systems with Applications, 23,93-102.

[18] Aryeetey, K., 2003. Data Analysis and Predictive Modelling Using The Variable Precision Rough Set Approach. M. Sc. Thesis, Faculty of Graduate Study and Research of University of Regina, 87p, Canada.

[19] Pawlak, Z., 2002. Rough Sets, Decision Algorithms and Bayes Theorem. Europen Journal of Operation Research, 136, 181-189.

[20] Shah. S., Kursak. A., 2004. Data Mining and Genetic Algorithms Based Gene / SNP Selection. Artificial Intelligence in Medicine, 31, 183 -196.

[21] Hui, S., Jha, G., 2000. Application Data Mining for Customer Service Support. Information and Management, 38, 1-13.

[22] Bland, C., 2002. The Discovery of Multiple Level Profile Association Rules. Ph. D. Thesis, Graduate School of Computer and Information Science of University of Missisipi, 123p, Missisipi.

[23] Oğuzlar, A., 2003. Veri Önisleme. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21, 67-76.

[24] Witten I., Frank E., 2000. Data Mining: Practical Machine Learning Tools and Techniques With Java Implementations. Morgan Kaufmann Publishers, San Fransisco, 120-129, 267-277.

[25] Shearer, C., 2000. The Crisp-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing, 5 (4), 13-23.

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

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

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

[29] Dolgun, M. Ö., (2006), Büyük Alışveriş Merkezleri için Veri Madenciliği Uygulamaları, Yüksek Lisans Tezi, Hacettepe Üniversitesi İstatistik Anabilim Dalı, Ankara.

[30] Agrawal, R. ve Srikant, R., (1994), Fast Algorithms for Mining Association Rules, VLDB Conference, Santiago, Chile.

[31] Srikant, R. ve Agrawal R., (1996), Mining Quantitive Association Rules in Large Relational Tables, ACM SIGMOD'96 International Conference of Management of Data, Montreal, 1-12.

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

[33] Gao, W., (2004), "A Hierarchical Document Clustering Algoritm", MSc Thesis, Dalhousie University, Halifax, Nova Scotia.

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

[35] Dunham, M. H., Xiao, Y., Gruenwald L. ve Hossain, Z., (2000), "A Survey of Association Rules", Teknik Rapor, Southern Methodist University, Department of Computer Science, TR00-CSE-8.

[36] Hearst, M. (2009), What is text mining, *http://www.sims.berkeley.edu/~hearst/textmining.html*.

[37] Tan, A.H., Yu, P.S. (2004), Guest Editorial: Text and Web Mining, *Applied Intelligence 18, 239-241, Kluwer Academic Publisher*.

[38] Unstructured data (2009), *http://en.wikipedia.org/wiki/Unstructured_data*.

[39] Özdemir Güzel, T., Dolgun, M.Ö., Patır, U., Deliloğlu, S., Korkmaz, H.E. (2007), 2005 Yılı Öğrenci Seçme Sınavı (ÖSS) Verileri Kullanılarak Öğrenci Profilinin Belirlenmesi, *5. +statistik Kongresi*, Antalya.

[40] Introduction to Text Mining (2008), *SPSS Inc*.

[41] Sholom M.W., Indurkhya N., Zhang T., Damerau F. (2004), Text Mining: Predictive Methods for Analyzing Unstructured Information, *Springer*.

[42] W. Fan, L. Wallace, S. Rich, Z. Zhang. (2006), Tapping into the power of text mining, *Communications of ACM*, 49(9), 76-82.

[43] Chakrabarti, S. (2003), Mining the Web: Discovering Knowledge from Hypertext Data, *Morgan Kaufmann Publishers*, San Francisco.

[44] Liu, B. (2007), Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, *Springer*.

[45] A. Güven, Ö. Özgür Bozkurt, O. Kalıpsız 'Veri Madenciliği Geleceği',Yıldız Teknik Üniversitesi

[46] A. Adsız, 'Metin Madenciliği',Ahmet Yesevi Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans Programı Dönem Projesi

[47] Berry, M. Survey of Text Mining : Clustering, Classification, and Retrieval (Hardcover) Springer; 1 edition, 2003

[48] Nahm, U. Text Mining With Information Extraction 2004, http://www.cs.utexas.edu/ftp/pub/AI-Lab/tech- reports/UT-AI-TR-04-311.pdf

[49] Bolat, M., Metin Filtrelemde En Hızlı İniş Metoduyla Optimal Sorgunun Bulunması, Başkent Üniversitesi Fen Bilimleri Enstitüsü,Yüksek Lisans Tezi, 2003

[50] Tonta, Y.. Bitirim, Y , Sever, H. Turkce Arama Motorlarinda Performans Degerlendirme, 2002, http://www.baskent.edu.tr/~sever/

[51] Sebastian F. Machine Learning in Automated Text Categorized .2005 http://www.math.unipd.it/~fabseb60/Publications/ACMCS02.pdf

ÖZGEÇMİŞ

04 Mart 1984 tarihi, Malatya doğumluyum. İlköğretim eğitimimi Malatya'da tamamladıktan sonra, Lise eğitimimi İstanbul'da tamamladım. 2001 yılında Girne Amerikan Üniversitesi, Mimarlık Mühendislik Fakültesi, Bilgisayar Mühendisliğine kaydoldum. Bu bölümden 2007 yılında mezun olduktan, aynı sene içerisinde Beykent Üniversitesi, Matematik Bilgisayar Anabilim Dalı Bilgisayar Ağları ve İnternet Teknolojileri bilim dalında yüksek lisans eğitimine başladım. 2008 yılından beri, kamu kuruluşunda yazılım mühendisliği görevini sürdürmekteyim.

Yabancı dilim İngilizcedir.

Emine Kübra ÇELİKYAY