VERİ MADENCİLİĞİ MODELLERİ VE UYGULAMA ALANLARI

Öğr. Gör. Serhat ÖZEKES

Abstract:

The major reason that data mining became one of the hottest current technologies of the information age is the wide availability of huge amounts of data and the need for turning such data into useful information and knowledge. As computer systems getting cheaper and computer power increases, the amount of data available to be collected and processed increases. Therefore using techniques that operates very well with large amounts of data becomes an obvious choice. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration. In the present study, the data mining models like Classification, Regression, Clustering and Association Rules are examined, and their application areas are discussed.

Keywords: Data Mining, Classification, Regression, Clustering, Association Rules

İstanbul Ticaret Üniversitesi, Meslek Yüksek Okulu Öğretim Görevlisi, scthat@iticu.cdu.tr

Özet:

Veri madenciliği, günümüz bilgi çağında en güncel teknolojilerden birisidir. Bilgisayar sistemlerinin her geçen gün hem daha ucuzluyor olması, hem de güçlerinin artıyor olması, bilgisayarlarda daha büyük miktarlarda verinin saklanabilmesine imkan vermektedir. Bu yüzden, büyük miktardaki verileri işleyebilen teknikleri kullanabilmek, büyük önem kazanmaktadır. Veri madenciliği bu gibi durumlarda kullanılan, büyük miktardaki veri setlerinde saklı durumda bulunan örüntü ve eğilimleri keşfetme işlemidir. Bu çalışmada veri madenciliği modelleri işlevlerine göre Sınıflama, Regresyon, Kümeleme ve Birliktelik Kuralları başlıkları altında incelenmekte ve uygulama alanları açıklanmaktadır.

Anahtar Sözcükler: Veri Madenciliği, Sınıflama, Regresyon, Kümeleme, Birliktelik Kuralları

1. GİRİŞ

Verilerin dijital ortamda saklanmaya başlanması ile birlikte, yeryüzündeki bilgi miktarının her geçen gün katlanarak arttığı günümüzde, veri tabanlarının sayısı da benzer, hatta daha yüksek bir oranda artmaktadır. Yüksek kapasiteli işlem yapabilme gücünün ucuzlamasının bir sonucu olarak, veri saklama hem daha kolaylaşmış, hem de verinin kendisi ucuzlamıştır. Veri tabanlarında saklanan veri, bir dağa benzetilirse, bu veri dağı tek başına değersizdir ve kullanıcı için çok fazla bir anlam ifade etmez. Ancak bu veri dağı, belirli bir amaç doğrultusunda sistematik olarak işlenir ve analiz edilirse, değersiz görülen veri yığınında, amaca yönelik sorulara cevap verebilecek çok değerli bilgilere ulaşılabilir.

Veri madenciliği veri tabanı teknolojisi, istatistik, yapay zeka (artificial intelligence), makine öğrenimi (machine learning), örüntü tanımlama (pattem recognition) ve veri görselleştirmesi (data visualization) gibi pek çok teknik alan arasında köprü görevi

gören çok disiplinli bir alandır. Veri madenciliği astronomi, biyoloji, finans, pazarlama, sigorta, tıp gibi bir çok dalda uygulanmaktadır.

Bu çalışmanın amacı, bilişim teknolojileri dünyasındaki önemini her geçen gün daha da arttıran veri madenciliği konusunu ve veri madenciliği modellerini incelemektir.

2. MODELLER VE UYGULAMA ALANLARI

Veri madenciliğinde kullanılan modeller, tahmin edici *(Predictive)* ve tanımlayıcı *(Descriptive)* olmak üzere iki ana başlık altında incelenmektedir [1].

Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçlan bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır [1]. Örneğin, bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır.

Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır [1]. X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu aileler ile, çocuğu olmayan ve geliri X/Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerlik gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir [2].

Veri madenciliği modellerini gördükleri işlevlere göre,

1- Sınıflama *(Classification)* ve Regresyon *(Regression)*
2- Kümeleme *(Clustering)*
3- Birliktelik Kuralları *(Association Rules)*

olmak üzere üç ana başlık altında incelemek mümkündür [2]. Sınıflama ve regresyon modelleri tahmin edici, kümeleme ve birliktelik kuralları modelleri tanımlayıcı modellerdir [2].

2.1. Sınıflama ve Regresyon

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen iki veri analiz yöntemidir [3]. Sınıflama kategorik değerleri tahmin ederken, regresyon süreklilik gösteren değerlerin tahmin edilmesinde kullanılır [3]. Örneğin, bir sınıflama modeli banka kredi uygulamalarının güvenli veya riskli olmalarını kategorize etmek amacıyla kurulurken, regresyon modeli geliri ve mesleği verilen potansiyel müşterilerin bilgisayar ürünleri alırken yapacakları harcamaları tahmin etmek için kurulabilir.

Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler şunlardır [2]:

1 - Karar Ağaçları (Decision Trees)
2- Yapay Sinir Ağları (Artificial Neural Networks)
3- Genetik Algoritmalar (Genetic Algorithms)
4- K-En Yakın Komşu (K-Nearest Neighbor)
5- Bellek Temelli Nedenleme (Memory Based Reasoning)
6- Naive-Bayes

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

Karar ağacı, adından da anlaşılacağı gibi bir ağaç görünümünde, tahmin edici bir tekniktir [4]. Ağaç yapısı ile, kolay anlaşılabilen kurallar yaratabilen, bilgi teknolojileri işlemleri ile kolay entegre olabilen en popüler sınıflama tekniğidir.

Karar ağacı karar düğümleri, dallar ve yapraklardan oluşur [3]. Karar düğümü, gerçekleştirilecek testi belirtir. Bu testin sonucu ağacın veri kaybetmeden dallara ayrılmasına neden olur. Her düğümde test ve dallara ayrılma işlemleri ardışık olarak gerçekleşir ve bu ayrılma işlemi üst seviyedeki ayrımlara bağımlıdır. Ağacın her bir dalı sınıflama işlemini tamamlamaya adaydır. Eğer bir dalın ucunda sınıflama işlemi gerçekleşemiyorsa, o daim sonucunda bir karar düğümü oluşur. Ancak daim sonunda belirli bir sınıf oluşuyorsa, o dalın sonunda yaprak vardır. Bu yaprak, veri üzerinde belirlenmek istenen sınıflardan biridir. Karar ağacı işlemi kök düğümünden başlar ve yukarıdan aşağıya doğru yaprağa ulaşana dek ardışık düğümleri takip ederek gerçekleşir.

Karar ağacı tekniğini kullanarak verinin sınıflanması iki basamaklı bir işlemdir [3]. İlk basamak öğrenme basamağıdır. Öğrenme basamağında önceden bilinen bir eğitim verisi, model oluşturmak amacıyla sınıflama algoritması tarafından analiz edilir. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. İkinci basamak ise sınıflama basamağıdır. Sınıflama basamağında test verisi, sınıflama kurallarının veya karar ağacının doğruluğunu belirlemek amacıyla kullanılır. Eğer doğruluk kabul edilebilir oranda ise, kurallar yeni verilerin sınıflanması amacıyla kullanılır.

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

Örneğin, bir eğitim verisi incelenerek kredi duruma sınıfını tahmin edecek bir model oluşturuluyor. Bu modeli oluşturan bir sınıflama kuralı

IF yaş = "41...50" AND gelir = yüksek THEN kredidurumu = mükemmel

şeklindedir. Bu kural gereğince yaşı "41...50" kategorisinde olan (yaşı 41 ile 50 arasında olan) ve gelir düzeyi yüksek bir kişinin kredi durumunun mükemmel olduğu görülür.

Oluşturulan bu modelin doğruluğu, bir test verisi aracılığı ile onaylandıktan sonra model, sınıfı belli olmayan yeni bir veriye uygulanabilir ve sınıflama kuralı gereği yeni verinin sınıfı "mükemmel" olarak belirlenebilir.

Tekrarlamak gerekirse bir karar ağacı, bir alandaki testi belirten *karar düğümlerinden,* testteki değerleri belirten *dallardan* ve sınıfı belirten *yapraklardan* oluşan akış diyagramı şeklindeki ağaç yapısıdır. Ağaç yapısındaki en üstteki düğüm *kök* düğümüdür.

Belirli bir sınıfın muhtemel üyesi olacak elemanların belirlenmesi, çeşitli durumların yüksek, orta, düşük risk grupları gibi çeşitli kategorilere ayrılması, gelecekteki olayların tahmin edilebilmesi için kurallar oluşturulması, sadece belirli alt gruplara özgü olan ilişkilerin tanımlanması, kategorilerin birleştirilmesi gibi alanlarda karar ağaçları kullanılmaktadır [2].

Karar ağaçları, hangi demografik grupların mektupla yapılan pazarlama uygulamalarında yüksek cevaplama oranına sahip olduğunun belirlenmesi (Direct Mail), bireylerin kredi geçmişlerini kullanarak kredi kararlarının verilmesi (Credit Scoring), geçmişte işletmeye en faydalı olan bireylerin özelliklerini kullanarak işe alma süreçlerinin belirlenmesi, tıbbi gözlem verilerinden yararlanarak en etkin kararların verilmesi, hangi değişkenlerin satışları etkilediğinin belirlenmesi, üretim verilerini inceleyerek ürün hatalarına yol açan değişkenlerin belirlenmesi gibi uygulamalarda kullanılmaktadır [2].

2.2. Kümeleme

Kümeleme, veriyi sınıflara veya kümelere ayırma işlemidir [5]. Aynı kümedeki elemanlar birbirleriyle benzerlik gösterirlerken, başka kümelerin elemanlarından farklıdırlar. Kümeleme veri madenciliği, istatistik, biyoloji ve makine öğrenimi gibi pek çok alanda kullanılır. Kümeleme modelinde, sınıflama modelinde olan veri sınıfları yoktur [6]. Verilerin herhangi bir sınıfı bulunmamaktadır. Sınıflama modelinde, verilerin sınıfları bilinmekte ve yeni bir veri geldiğinde bu verinin hangi sınıftan olabileceği tahmin edilmektedir. Oysa kümeleme modelinde, sınıfları bulunmayan veriler gruplar halinde kümelere ayrılırlar. Bazı uygulamalarda kümeleme modeli, sınıflama modelinin bir önişlemi gibi görev alabilmektedir [6].

Marketlerde farklı müşteri gruplarının keşfedilmesi ve bu grupların alışveriş örüntülerinin ortaya konması, biyolojide bitki ve hayvan sınıflandırmaları ve işlevlerine göre benzer genlerin sınıflandırılması, şehir planlanmasında evlerin tiplerine, değerlerine ve coğrafik konumlarına göre gruplara aynlması gibi uygulamalar tipik kümeleme uygulamalarıdır. Kümeleme aynı zamanda Web üzerinde bilgi keşfi için dokümanların sınıflanması amacıyla da kullanılabilir [7].

Veri kümeleme güçlü bir gelişme göstermektedir. Veri tabanlarında toplanan veri miktarının artmasıyla orantılı olarak, kümeleme analizi son zamanlarda veri madenciliği araştırmalarında aktif bir konu haline gelmiştir.

Literatürde pek çok kümeleme algoritması bulunmaktadır. Kullanılacak olan kümeleme algoritmasının seçimi, veri tipine ve amaca bağlıdır. Genel olarak başlıca kümeleme yöntemleri şu şekilde sınıflandınlabilir [3]:

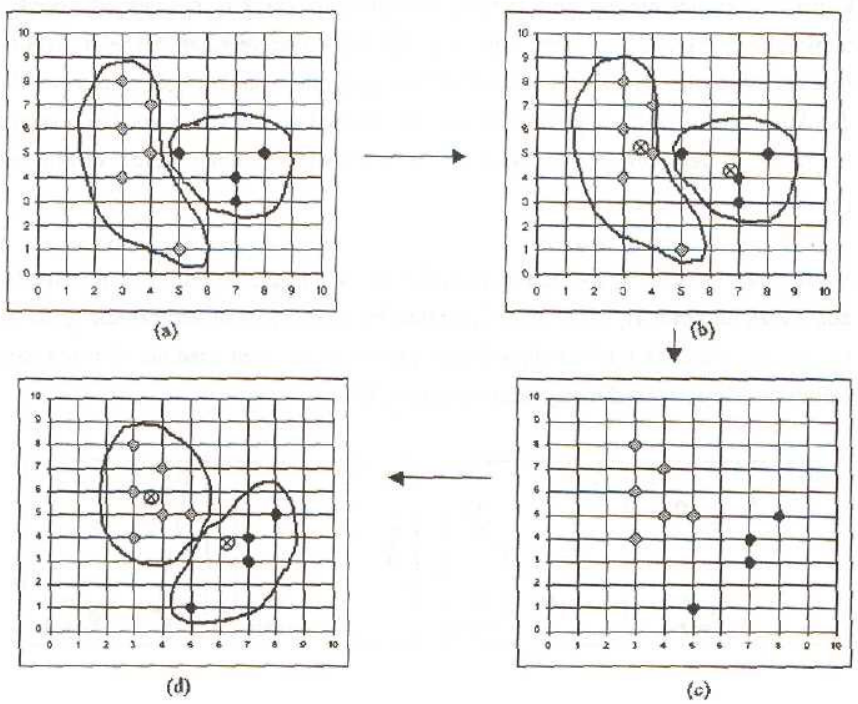
1 - Bölme yöntemleri (Partitioning methods)
2- Hiyerarşik yöntemler (Hierarchical methods)
3- Yoğunluk tabanlı yöntemler (Density-based methods)
4- Izgara tabanlı yöntemler (Grid-based methods)
5- Model tabanlı yöntemler (Model-based methods)

Bölme yöntemlerinde, *n* veri tabanındaki nesne sayısı ve *k* oluşturulacak küme sayısı olarak kabul edilir. Bölme algoritması *n* adet nesneyi, *k* adet kümeye böler *($k \leq n$).* Kümeler tarafsız bölme kriteri olarak nitelendirilen bir kritere uygun oluşturulduğu için aynı kümedeki nesneler birbirlerine benzerken, farklı kümedeki nesnelerden farklıdırlar [3].

En iyi bilinen ve en çok kullanılan bölme yöntemleri k-means yöntemi, k-medoids yöntemi ve bunların varyasyonlarıdır [8].

k-means yöntemi, ilk önce *n* adet nesneden rasgele *k* adet nesne seçer ve bu nesnelerin her biri, bir kümenin merkezini veya orta noktasını temsil eder. Geriye kalan nesnelerden her biri kendisine en yakın olan küme merkezine göre kümelere dağılırlar. Yani bir nesne hangi kümenin merkezine daha yakın ise o kümeye yerleşir. Ardından her küme için ortalama hesaplanır ve hesaplanan bu değer o kümenin yeni merkezi olur. Bu işlem tüm nesneler kümelere yerleşinceye kadar devam eder [3].

Bir nesne grubunun, Şekil 2.1'de görüldüğü gibi uzayda konumlanmış olduğu varsayılsın. Kullanıcının bu nesneleri iki kümeye ayırmak istediği varsayılırsa, *k=2* olur [3]. Şekil 2.1 (ayda ilk önce rasgele iki nesne, iki kümenin merkezi olarak seçilmiş ve diğer nesneler de bu merkezlere olan yakınlıklarına göre iki kümeye ayrılmışlardır. Bu ayrıma göre her iki kümenin nesnelerinin yeni ortalaması alınmış ve bu değer kümelerin yeni merkezleri olmuştur. Bu yeni merkezler Şekil 2.1(b)'de üstünde çarpı işareti bulunan noktalarla gösterilmektedir. Bu yeni çarpı işaretli merkezlere göre, her iki kümede de birer nesne diğer kümenin merkezine daha yakm duruma gelmişlerdir. Bu durum Şekil 2.1(c)'de görülmektedir. (5,1) koordinatındaki nesne ile (5,5) koordinatındaki nesne küme değiştirmişlerdir. Her iki kümedeki bu yeni katılımlar ile kümelerdeki nesnelerin ortalama değerleri ve dolayısıyla merkezleri değişmiştir [3]. Yeni hesaplanan merkezler Şekil 2.1(d)'de üstünde çarpı işareti bulunan noktalarla gösterilmektedir. Artık açıkta bir nesne kalmadığı ve her nesne içinde bulunduğu kümenin merkezine en yakm durumda bulunduğu için k-means yöntemi ile kümelere bölünme işlemi Şekil 2.1(d)'de görüldüğü gibi sonlanmıştır [3].



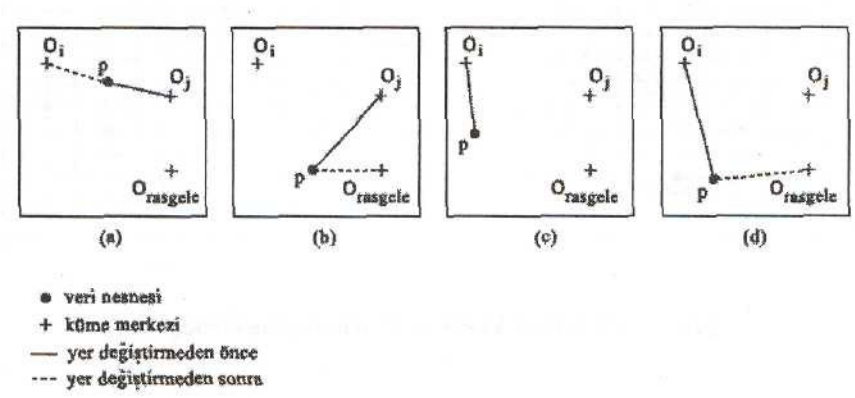
Şekil 2.1. k-means Yöntemiyle Kümeleme Örneği

k-means yöntemi, sadece kümenin ortalaması tanımlanabildiği durumlarda kullanılabilir [9], Kullanıcıların k değerini, yani oluşacak küme sayısını belirtme gerekliliği bir dezavantaj olarak görülebilir. Esas önemli olan dezavantaj ise dışarıda kalanlar (outliers) olarak adlandırılan nesnelere karşı olan duyarlılıktır [3]. Değeri çok büyük olan bir nesne, dahil olacağı kümenin ortalamasını ve merkez noktasını büyük bir derecede değiştirebilir. Bu değişiklik kümenin hassasiyetini bozabilir.

Bu sorunu gidermek için kümedeki nesnelerin ortalamasını almak yerine, kümede ortaya en yakın noktada konumlanmış olan nesne anlamındaki medoid kullanılabilir. Bu işlem k-medoids yöntemi ile gerçekleştirilir.

k-medoids kümeleme yönteminin temel stratejisi ilk olarak n adet nesnede, merkezi temsili bir medoid olan k adet küme bulmaktır [3]. Geriye kalan nesneler, kendilerine en yakın olan medoide göre *k* adet kümeye yerleşirler. Bu bölünmelerin ardından kümenin ortasına en yakın olan nesneyi bulmak için medoid, medoid olmayan her nesne ile yer değiştirir. Bu işlem en verimli medoid bulunana kadar devam eder [3].

Şekil 2.2'de O_i ve O_j iki ayrı kümenin medoidlerini, $O_{rasgele}$ rasgele seçilen ve medoid adayı olan bir nesneyi, p ise medoid olmayan bir nesneyi temsil etmektedir. Şekil 2.2 $O_{rasgele}$'nin, şu anda medoid olan O_j'nin yerine geçip, yeni medoid olup olamayacağını belirleyen dört durumu göstermektedir [3].



Şekil 2.2. k-medoids Yöntemiyle Kümeleme Örneği

(a): p nesnesi şu anda O_j medoidine bağlıdır (Oj medoidinin bulunduğu kümededir). Eğer O_j , $O_{rasgele}$ ile yer değiştirir ve p O_i 'ye en yakınsa, p nesnesi O_i 'ye geçer.

(b): p nesnesi şu anda O_j medoidine bağlıdır. Eğer O_j , $O_{rasgele}$ ile yer değiştirir ve p Or_{asgele}'ye en yakınsa, p nesnesi $O_{rasge!e}$'ye geçer.

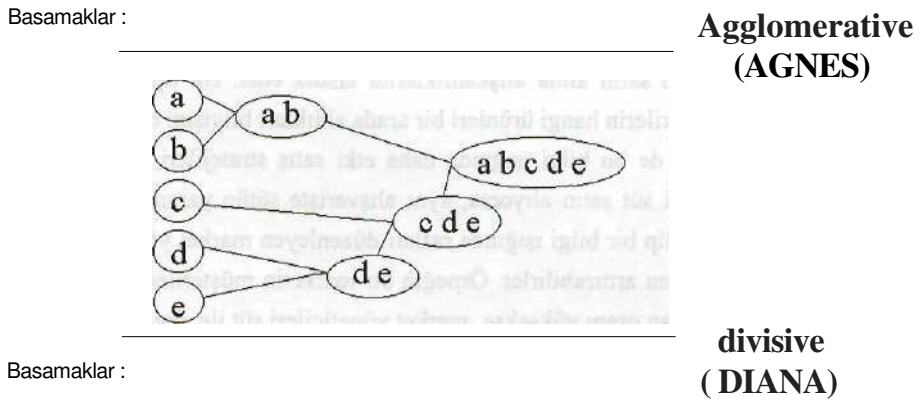
(c): p nesnesi şu anda 0_i medoidine bağlıdır. Eğer O_j , $O_{rasgele}$ ile yer değiştirir ve p hala $O_{rasgele}$ 'ye en yakınsa, p nesnesi yine O_i 'ye bağlı kalır.

(d): p nesnesi şu anda O_i medoidine bağlıdır. Eğer O_j, Orasgele ile yer değiştirir ve P $O_{rasgele}$ 'ye en yakınsa, p nesnesi $O_{rasgele}$ 'ye geçer.

Kümeleme yöntemlerinden biri olan hiyerarşık yöntemler, veri nesnelerini kümeler ağacı şeklinde gruplara ayırma esasına dayanır [9]. Hiyerarşık kümeleme yöntemleri, hiyerarşık ayrışmanın aşağıdan yukarıya veya yukarıdan aşağıya doğru olmasına göre *agglomerative* ve *divisive hiyerarşik* kümeleme olarak sınıflandırılabilir [9].

Agglomerative hiyerarşik kümelemede, Şekil 2.3'de görüldüğü üzere hiyerarşik ayrışma aşağıdan yukarıya doğru olur [9]. İlk olarak her nesne kendi kümesini oluşturur ve ardından bu atomik kümeler birleşerek, tüm nesneler bir kümede toplanıncaya dek daha büyük kümeler oluştururlar.

Divise hiyerarşik kümelemede, Şekil 2.3'de görüldüğü üzere hiyerarşik ayrışma yukarıdan aşağıya doğru olur [9]. İlk olarak tüm nesneler bir kümededir ve her nesne tek başına bir küme oluşturana dek, kümeler daha küçük parçalara bölünürler.



Şekil 2.3. Hiyerarşik Kümeleme Örneği

Şekil 2.3, bir agglomerative hiyerarşik kümeleme yöntemi olan AGNES (AGlomerative NESting) ve bir divise hiyerarşik kümeleme yöntemi olan DIANA (Dlvise ANAlysis) uygulaması göstermektedir [9]. Bu yöntemler beş nesneli (a,b,c,d,e) bir veri setine uygulanmaktadır. Başlangıçta AGNES her nesneyi bir kümeye yerleştirir. Kümeler, bazı kriterlere göre basamak-basamak birleşirler. Örneğin C_1 ve C_2 kümeleri, eğer C_1 kümesindeki bir nesne ve C_2 kümesindeki bir nesne ile, diğer kümelerdeki herhangi iki nesne arasında belirlenen uzaklık mesafesini karşılayacak bir mesafe varsa birleşebilirler. Bu birleşme işlemi tüm nesneler bir kümede toplanıncaya kadar devam eder [3]. DIANA'da ise tüm nesnelerin içinde toplandığı küme, her küme bir nesne içerecek duruma gelene kadar bölünür [9].

2.3. Birliktelik Kuralları

Birliktelik kuralları, büyük veri kümeleri arasında birliktelik ilişkileri bulurlar [10]. Toplanan ve depolanan verinin her geçen gün gittikçe büyümesi yüzünden, şirketler veritabanlarındaki birliktelik kurallarını ortaya çıkarmak istemektedirler. Büyük miktardaki mesleki işlem kayıtlarından ilginç birliktelik ilişkilerini keşfetmek, şirketlerin karar alma işlemlerini daha verimli hale getirmektedir.

Birliktelik kurallarının kullanıldığı en tipik örnek market sepeti uygulamasıdır. Bu işlem, müşterilerin yaptıkları alışverişlerdeki ürünler arasındaki birliktelikleri bularak müşterilerin satın alma alışkanlıklarını analiz eder. Bu tip birlikteliklerin keşfedilmesi, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve market yöneticileri de bu bilgi ışığında daha etki satış stratejileri geliştirebilirler. Örneğin bir müşteri süt satın alıyorsa, aynı alışverişte sütün yanında ekmek alma olasılığı nedir? Bu tip bir bilgi ışığında rafları düzenleyen market yöneticileri ürünlerindeki satış oranını arttırabilirler. Örneğin bir marketin müşterilerinin süt ile birlikte ekmek satın alan oranı yüksekse, market yöneticileri süt ile ekmek raflarını yan yana koyarak ekmek satışlarını arttırabilirler.

Örneğin bir A ürününü satın alan müşteriler aynı zamanda B ürününü da satın alıyorlarsa, bu durum (2.1)'deki Birliktelik Kuralı ile gösterilir [11]:

$$A \Rightarrow B \quad [destek = \%2,\ güven = \%60] \tag{2.1}$$

Buradaki destek ve güven ifadeleri, kuralın ilginçlik ölçüleridir. Sırasıyla, keşfedilen kuralın kullanışlığını ve doğruluğunu gösterirler. (2. l)'deki Birliktelik Kuralı için 2% oranındaki bir destek değeri, analiz edilen tüm alışverişlerden %2'sinde A ile B ürünlerinin birlikte satıldığını belirtir. %60 oranındaki güven değeri ise A ürününü satın alan müşterilerinin %60'ının aynı alışverişte B ürününü de satın aldığını ortaya koyar [11]. Kullanıcı tarafından minimum destek eşik değeri ve minimum güven eşik değeri belirlenir ve bu değerleri aşan birliktelik kuralları dikkate alınır.

Büyük veri tabanlarında birliktelik kuralları bulunurken, şu iki işlem basamağı takip edilir [11]:

1- Sık tekrarlanan öğeler bulunur: Bu öğelerin her biri en az, önceden belirlenen minimum destek sayısı kadar sık tekrarlanırlar.
2- Sık tekrarlanan Öğelerden güçlü birliktelik kuralları oluşturulur: Bu kurallar minimum destek ve minimum güven değerlerini karşılamalıdır.

Sık tekrarlanan öğeleri bulmak için kullanılan en temel yöntem Apriori Algoritmasıdır [21,25,26,27]. Aşağıda Apriori algoritması bir örnekle anlatılmaktadır.

Tablo 2.1. Marketten Yapılan Alışveriş Bilgilerini İçeren *D* Veritabanı [3]

ANO	Ürün NO
A100	I1 , 12,15
A200	I2,14
A300	I2,13
A400	I1,12,14
A500	I1,13
A600	I2,13
A700	I1,13
A800	I1,12,13,15
A900	I1,12,13

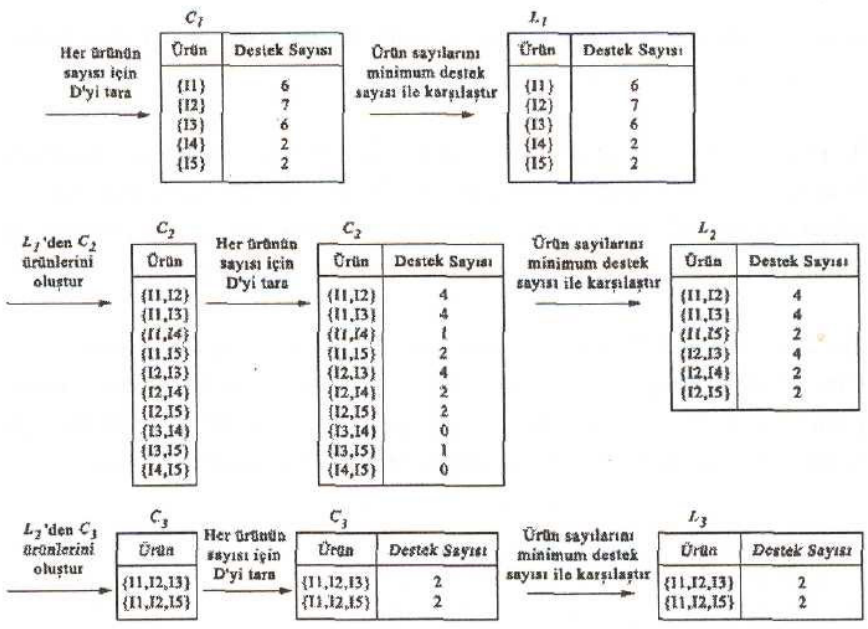
Tablo 2.1'de bir marketten yapılan alışverişlerin bilgilerini içeren *D* veritabanı görülmektedir. Bu veritabanında yapılan alışverişlerin numaraları ANO sütununda görülmektedir. Her alışverişte satın alınan ürünler de Ürün No sütununda görülmektedir. Apriori algoritmasında takip edilen basamaklar Şekil 2.4'de gösterilmektedir. [3]

1- Algoritmanın ilk adımında, her ürün tek başına bulunduğu C_1 kümesinin ele manıdır. Algoritma, her ürünün sayısını bulmak için tüm alışverişleri tarar ve elde edilen sonuçlar Şekil 2.4'de Destek Sayısı sütununda görülmektedir. Tablo 2.1'de görülebileceği gibi *D'*de I1 ürününden 6 adet, 12 ürününden 7 adet, 13 ürününden 6 adet, 14 ürününden 2 adet ve 15 ürününden de 2 adet satıldığı görülmektedir.

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

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

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



Şekil 2.4. Apriori Algoritmasının Gösterimi

5- C_2 kümesindeki ürünlerden minimum destek eşik değerini aşan ürünler L_2 küme sine aktarılır.

6- Hangi ürünlerin üçlü olarak sık tekrarlandığını belirlemek için L_2 kümesindeki ürünlerin üçlü kombinasyonları bulunarak C_3 kümesi oluşturulur. Bu durumda C_3 = {{I1,I2,I3}, {I1,I2,I5}, {I1,I3,I5}, {I2,I3,I4}, {I2,I3,I5}} olması beklenir.

Ancak Apriori algoritmasına göre, sık tekrarlanan öğelerin alt kümeleri de sık tekrarlanan öğe olması gerekmektedir. Buna göre yukarıdaki C_3 kümesindeki elemanlar sık tekrarlanan olmadığı için, yeni C_3 kümesi C_3 = {{I1,I2,I3}}, {I1,I2,I5}} olur.

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

8- C_3 kümesindeki ürünlerden minimum destek eşik değerini aşan ürünler L_3 küme sine aktarılır.

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

Sık tekrarlanan öğeleri bulduktan sonra l = {I1, I2, I5} urallarını oluşturmaya gelir. Örneğin sık tekrarlanan bir öğe olaNiçin, boş olmayan tüm alt kümeler şunlardır [11]: {I1, I2}, {I2, I5}, {I2, I5}, {I1}, {I2},{I5}. Bu durumda Tablo 2.1'deki veritabanına bakarak şu birliktelik kuralları çıkartılabilir [3].

1- I1 Λ I2 ⇒ I5,	güven = 2 / 4 = %50
2- I1 Λ I5 ⇒ I2,	güven = 2 / 2 = %100
3- I2 Λ I5 ⇒ I1,	güven = 2 / 2 = %100
4- I1 ⇒ I2 Λ I5,	güven = 2 / 6 = %33
5- I2 ⇒ I1 Λ I5,	güven = 2 / 7 = %29
6- I5 ⇒ I1 Λ I2,	güven = 2 / 2 = %100

Eğer minimum güven eşik değeri %70 olarak belirlenmişse, ikinci, üçüncü ve altıncı kurallar dikkate alınır çünkü diğer kurallar eşik değerini aşamamış olurlar [3].

3. SONUÇ

Bu çalışmada veri madenciliği modelleri Sınıflama ve Regresyon, Kümeleme ve Birliktelik Kuralları başlıkları altında incelenmiş ve kullanım alanları örnekler verilerek açıklanmıştır. Bir veri madenciliği uygulaması gerçekleştirileceği zaman, eldeki verinin ve çözülmesi gereken mesleki problemin çok iyi bir şekilde analiz edilmesi ve anlaşılması gerekir. Bu iki unsur, veri madenciliği uygulamasının başarısını önemli bir oranda etkileyecektir. Eldeki veri ve problem anlaşıldıktan sonra, veri madenciliği modellerinden ve tekniklerinden amaca en uygun olanı seçilmelidir. Uygun olmayan bir teknik seçilerek yapılan veri madenciliği uygulamasının başarılı olması düşünülemez.

KAYNAKLAR

[1] Methodologies for Knowledge Discovery and Data Mining : Third Pacific-Asia Conference, Pakdd-99, Beijing, China, April 26-28, 1999 : Proceedings, Zhong, N. - Zhou, L., *Springer Verlag,* 1999.

[2] Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, Akpınar, H., *İstanbul Üniv. İşletme Fakültesi Dergisi,* C:29 S: 1 Nisan 2000

[3] Data Mining Concepts and Techniques, Han, J.-Kamber, M., *Morgan Kaufmann Publishers,* 1st Ed., San Francisco, USA, 2000.

[4] Mastering Data Mining: The Art and Science of Customer Relationship Management, Berry, M.J.A. - Linoff, G.S., *John Wiley & Sons,* 1st Ed., 1999.

[5] Chameleon: Hierarchical Clustering Using Dynamic Modeling, Karypis G. - Han E.-Kumar V, *IEEE Computer,* 1999 : 68-75

[6] Clustering Data Without Distance Functions, Ramkumar G.D. - Swami A., *IEEE Bulletin ofthe Technical Committee on Data Engineering,* Vol.21 No.l, March 1998 : 9-14.

[7] Data Mining with Microsoft SQL Server 2000, Seidman, C, *Microsoft Press,* 1 st Ed.; Washington, USA, 2001.

[8] Mining Databases: Towards Algorithms for Knowledge Discovery, Fayyad U., *IEEE Bulletin of the Technical Committee on Data Engineering,* Vol.21 No1, March 1998:41-48.

[9] Cluster Analysis, Han, J., *http://www-sal.es.uiuc.edu/~hanj/bk/8clst.ppt.*

[10] Mining Multiple-Level Association Rules in Large Databases, Han J. - Fu Y., *IEEE Transactions on Knowledge and Data Engineering,* vol 11 no.5, 1999

[11] Parallel and Dİstributed Association Mining: A Survey, Zaki, M. J., *IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining,* Vol. 7, No. 5, December 1999 : 14-25