NİCEL BİRLİKTELİK KURALLARI İÇİN İLGİNÇLİK ÖLÇÜTÜ VE ÇOKLU DESTEK DEĞERİ

Yalçın ATEŞ

Yüksek Lisans Tezi
Yazılım Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Murat KARABATAK
AĞUSTOS -2017

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NİCEL BİRLİKTELİK KURALLARI İÇİN İLGİNÇLİK ÖLÇÜTÜ VE ÇOKLU DESTEK DEĞERİ

YÜKSEK LİSANS TEZİ

Yalçın ATEŞ

(122137104)

Tezin Enstitüye Veriliş Tarihi : 19 Temmuz 2017
Tezin Savunulduğu Tarih : 4 Ağustos 2017

Tez Danışmanı:	**Doç. Dr. Murat KARABATAK (FÜ)**
Diğer Jüri Üyeleri	**Doç. Dr. Cihan VAROL (SHSU)**
	Doç. Dr. Bilal ALATAŞ (FÜ)

AĞUSTOS-2017

ÖNSÖZ

Yüksek Lisans öğrenimim süresince ve özellikle de tez aşamasında bilgi, görüş ve önerilerini benden esirgemeyen başta danışmanım Doç. Dr. Murat KARABATAK olmak üzere Prof. Dr. Engin AVCI, Doç. Dr. Bilal ALATAŞ, Doç. Dr. Erkan TANYILDIZI ve Yrd. Doç. Dr. Mustafa ULAŞ hocalarıma sonsuz teşekkür ederim.

.

Yalçın ATEŞ

İÇİNDEKİLER

Sayfa No

ÖZET

Veri madenciliği, sayısal ortamlarda biriken verilerden anlamlı ve faydalı bilgiler elde etmek için son yıllarda yoğun olarak kullanılan bir çalışma alanıdır. Birliktelik kuralı ise veriler arasındaki ilişkileri ortaya çıkarmak için kullanılan veri madenciliği tekniklerinden biridir. Birliktelik kuralı, ilk olarak 1993 yılında Agrawal ve arkadaşları tarafından ele alınmıştır. Birliktelik kuralı uygulamalarında kullanılan destek ve güven değerleri bu yöntemin en önemli iki parametresidir. Ancak nicel değerlere sahip veri setlerinde destek değerinin belirlenmesi sorun oluşturabilmektedir. Minimum destek değerinin uygun seçilememesi, veri setindeki ilginç ve değerli bazı kuralların ya üretilememesine ya da değerli olmayan çok sayıda gereksiz kural üretilmesine neden olmaktadır. Bu tez çalışmasında, nicel veri setleri üzerinde destek değerinin seçilmesi sorununu giderebilmek için çoklu destek değeri kullanan yeni bir yaklaşım önerilmektedir. Önerilen bu yaklaşım iki farklı örnek veri seti üzerinde uygulanmış ve elde edilen sonuçlar detaylı bir şekilde açıklanmıştır.

Anahtar Kelimeler: Birliktelik Kuralı, İlginçlik Ölçütü, Destek, Çoklu destek

SUMMARY

Interestingness Measure and Multi Support Value for Quantitative Association Rules

Data Mining is a field that has been used extensively in recent years to obtain meaningful and useful information from data accumulated in digital environments. Association Rule is one of the Data Mining techniques that are used to reveal relationships between data. The association rule is first discussed in 1993 by Agrawal et al. The support and confidence values used in association rules are the two most important parameters of this method. However, determining the value of support in quantitative data sets can be problematic. Failure to select the minimum support value causes some interesting and valuable rules in the dataset to either not be produced or to generate a large number of unnecessary rules that are not valuable. In this thesis, a new approach is proposed which uses multiple support values to overcome the problem of selecting support value on quantitative data sets. This proposed approach has been applied on two different sample data sets and the results obtained are explained in detail.

Keywords: Association Rule, Interestingness Measure, Support, Multi Support

ŞEKİLLER LİSTESİ

Sayfa No

TABLOLAR LİSTESİ

Sayfa No

KISALTMALAR LİSTESİ

AIS	Agrawal Imielinski Swami
DHP	Dynamic Hashing and Pruning
DIC	Dynamic Itemset Counting
SETM	Set Oriented Mining
SQL	Structured Query Language
TBA	Temel Bileşen Analizi - *Principal Component Analiysis (PCA)*
VMA	Veri Madenciliği Algoritması
VTBK	Veri Tabanlarında Bilgi Keşfi
YNK	Yoğun Nesne Kümesi
YSA	Yapay Sinir Ağı
D	Veri tabanı
|D|	Veri tabanı boyutu
I	Veri tabanı nitelikleri
T	Veri tabanı işlemleri
C_i	i-elemanlı aday nesne kümeleri
L_i	i-elemanlı yoğun nesne kümeleri
$X \rightarrow Y$	Birliktelik kuralı gösterimi
c	Güven değeri
s	Destek değeri
x	Gerçek giriş uzayındaki gözlemler veya ölçümler
y	Çıkış sınıfları
α	Öğrenme oranı
θ_i	i. İşlem elemanının eşik değeri

1. GİRİŞ

Francis Bacon'nun 1597' de söylediği "Scientia potentia est- Bilgi güçtür" sözü günümüzde önemini daha da artırmıştır. Bilgi ve iletişim çağının hızla geliştiği günümüzde, teknoloji hayatın her alanında kullanılmaya başlanmıştır. Teknolojinin bu kadar yaygın kullanılması çok büyük veri yığınlarını beraberinde getirmektedir. Internet ağının yaygınlaşmasıyla, oluşan bu verilere artık her yerden ulaşılabilmektedir. Ancak bilgiye her yerden ulaşmak kadar ulaşılan bilgilerden maksimum seviyede faydalanmakta önemli hale gelmiştir. Bu nedenle veri tabanı sistemleri hayatın her alanında kullanılmaya başlanmıştır.

Veri tabanı sistemlerinin günümüzde yaygın olarak kullanılması elde edilen verilerde olağanüstü artışlar sağlamıştır. Bu durum, insanları bu verilerden maksimum sonuçlar çıkarmak için yeni yollar keşfetmeye yöneltmiştir. Çünkü veri kendi başına bir şey ifade etmez. Veriyi bilgiye çevirmeye veri analizi denir. Bilgi ise, veri analizi sonucunda elde edilen anlamlı , bir soruya cevap olabilecek nitelikteki ve bir amaca ulaşmak için kullanılan verilerdir. Standart veri işleme araçları büyük veri tabanlarında yetersiz kalmaktadır. Bu nedenle Veri Tabanlarında Bilgi Keşfi (VTBK) adı altında yeni arayışlar ortaya çıkmaktadır. Birçok araştırmacı tarafından VTBK ve veri madenciliği terimleri eş anlamlı olarak kullanılmaktadır [1].

Veri madenciliği günümüzde pek çok alanda kullanılmaktadır. Örneğin bir işletmenin satış verileri incelenerek bir sonraki ayın ürün satış öngörüleri çıkarılabilir, müşteri bilgileri analiz edilerek müşterilere özel kampanya ve satış stratejileri oluşturulabilir. Bu işlemlerin binlerce ürün ve müşteri için uygulanabilirliği maalesef çok zordur. İşte bu aşamada veri madenciliği devreye girmektedir.

Kısaca veri madenciliği; mevcut veri yığınlarından değerli bilgiler elde etme işidir. Bu sayede veriler arasında ilişkiler incelenerek ileriye dönük öngörüler yapılabilir.

Veri madenciliğinde bilgilerin verimli kullanılabilmesi için birçok teknik kullanılmaktadır. Bu tekniklerden biri de birliktelik kuralıdır. Birliktelik kuralı; veri tabanındaki kayıtların analiz edilerek, hangi işlemlerin aynı zamanda meydana gelebileceklerini tespit etmeye yarayan veri madenciliği yöntemlerinden biridir. Birliktelik kuralı, nesnelerin veya niteliklerin bir arada olma durumlarını belirlemekte ve birçok alanda kullanılabilmektedir. Birliktelik kuralı bulma işlemi, yoğun nesne kümesi

hesaplamaya dayalı bir işlem olup büyük veri tabanları üzerinde uygulanması oldukça pahalı bir işlemdir. Bu nedenle daha önceden tespit edilen birliktelik kurallarının korunması da oldukça önemli bir konu olmaktadır.

Genellikle büyük süpermarketlerde oluşan satış verilerine, market sepet verisi adı verilmektedir. Birçok kuruluş market sepet verilerinin önemini kullanarak bu verilerden büyük faydalar sağlamayı amaçlamaktadır. Market sepet verisi üzerinde birliktelik kuralı problemi ilk olarak 1993 yılında ele alınmıştır [2]. Sepet analizinde amaç, nitelikler (ürün satışları) arasındaki ilişkiyi bulmaktır. Bu ilişkilerin bilinmesi şirketin kârını arttırmak için kullanılabilir. X malını alan bir müşterinin Y malını çok yüksek bir ihtimalle aldığı veya bir müşteri X mallarını alır, ancak Y malını almazsa, potansiyel bir Y müşterisidir. Sepet analizi, günlük işlemlerden sonra elde edilen verilerden anlamlı korelasyonlar çıkarmak için kullanılır. "Eğer bir müşteri A malını satın alırsa, % x ihtimalle B malını satın alabilir." şeklindeki bir sonuç bu mağaza için çok kullanışlı bir bilgi olabilir.

1.1. Literatür

Birliktelik kuralı birçok araştırmada ele alınmış ve birliktelik kuralı üreten algoritmalar geliştirilmiştir. Kaynak [3]'te önerilen Apriori algoritması bu algoritmalar arasında en fazla bilinen ve en yaygın kullanıma sahip algoritmalardan biridir. Yine kaynak [3]'te Apriori_TID algoritması da önerilmektedir. Kaynak [4]'te önerilen Apriori_Hybrid algoritması, Apriori ve Apriori_TID algoritmalarının her ikisini beraber kullanan melez bir algoritmadır. Ayrıca kaynak [5]'te çevrimdışı aday nesne kümesi belirleme algoritması, kaynak [6, 7]'de SETM algoritması, kaynak [8]'de DHP algoritması, kaynak [9]'da Partition algoritması, kaynak [10]'da MONET algoritması, kaynak [11]'de Sampling algoritması, kaynak [12]'de DIC algoritması, kaynak [13]'te Eclat, Maxeclat, Clique, Maxclique algoritmaları, kaynak [14]'te Max-Miner algoritması ve kaynak [15]'te Carma algoritması birliktelik kuralı üreten algoritmalar olarak bilinmektedir.

Birliktelik kuralı üreten algoritmalar, sadece ürün birlikteliklerini dikkate almayıp ürünlerin miktar, ağırlık ve hiyerarşik bilgi düzeni gibi özellikleri de göz önüne almaktadır. Birliktelik kuralı üretilecek veri setine ait niteliklerin nicel değerlerden oluşması birliktelik kuralı algoritmaları için problem oluşturmaktadır. Bu problemleri çözmek için literatürde çeşitli çalışmalar bulunmaktadır. Chun ve arkadaşları nitelikler için geçici bulanıklaştırma yöntemi uygulayarak nicel veriler üzerinde birliktelik kuralı uygulaması gerçekleştirmiştir

[16]. Bir diğer çalışmada Gosain ve Maneela, gerçekte birçok verinin nicel değerlerden oluştuğu ve birliktelik kuralı probleminin nicel veriler üzerinde çalışmasına yönelik farklı algoritmalar önermiştir [17]. Ouyang W. ise yaptığı çalışmada geleneksel birliktelik kuralında karşılaşılan problemlere değinmiş ve bu problemin çözümüne yönelik bulanık çoklu destek değer kullanan yeni bir algoritma önermiştir [18]. Yine Lee ve arkadaşları, çalışmasında çoklu minimum destek değeri uygulayan bir çalışma yaparak bu probleme bir çözüm önerisi sunmuşlardır [19]. Ancak yaptıkları çalışmada çoklu destek değerini ürünler bazında gerçekleştirmişlerdir.

Özellik seçimi, sınıflandırma problemlerinde ve birçok öğrenme algoritmalarında önemli bir yer tutmaktadır. Özellik seçiminde amaç, çok sayıda niteliğe sahip olan bir veri tabanında konu ile ilgili olan en önemli niteliklerin seçilmesidir. Bu sayede daha az işlem yükü ile yüksek başarım elde edilebilmektedir. Özellik seçimi yöntemleri ile ilgili literatürde çok sayıda araştırma bulunmakta ve birçok alanda uygulamaları ile karşılaşılmaktadır [20]. Temel bileşen analizi [21, 22] ve doğrusal ayrışım analizi [23], bu alandaki en önemli yöntemlerdendir. Bu yöntemler özellik seçimi yapmaktansa daha çok boyut indirgeme yaptığı için yeterince cazip olamamaktadır. Bu nedenle kaynak [24-34]'te önerilen özellik seçimi yöntemleri ile literatürde yaygın olarak karşılaşmak mümkündür. Ancak veri tabanındaki tüm niteliklerden kaç tanesinin seçilmesinin daha uygun olacağı konusunda henüz tam bir çözüm önerisi getirilememiştir.

Birliktelik kuralı kullanılarak veri tabanları üzerinden bilgi keşfi yapılabildiğine göre, öğrenci verileri üzerinde de birliktelik kuralı kullanımının faydalı sonuçlar vermesi olası bir durumdur. Literatürde bu konuda yapılan çalışmalara da az da olsa rastlanmaktadır. Bu konuda yapılan temel çalışmalar kaynak [35-38]'da belirtilen çalışmalardır.

1.2. Tezin Amacı

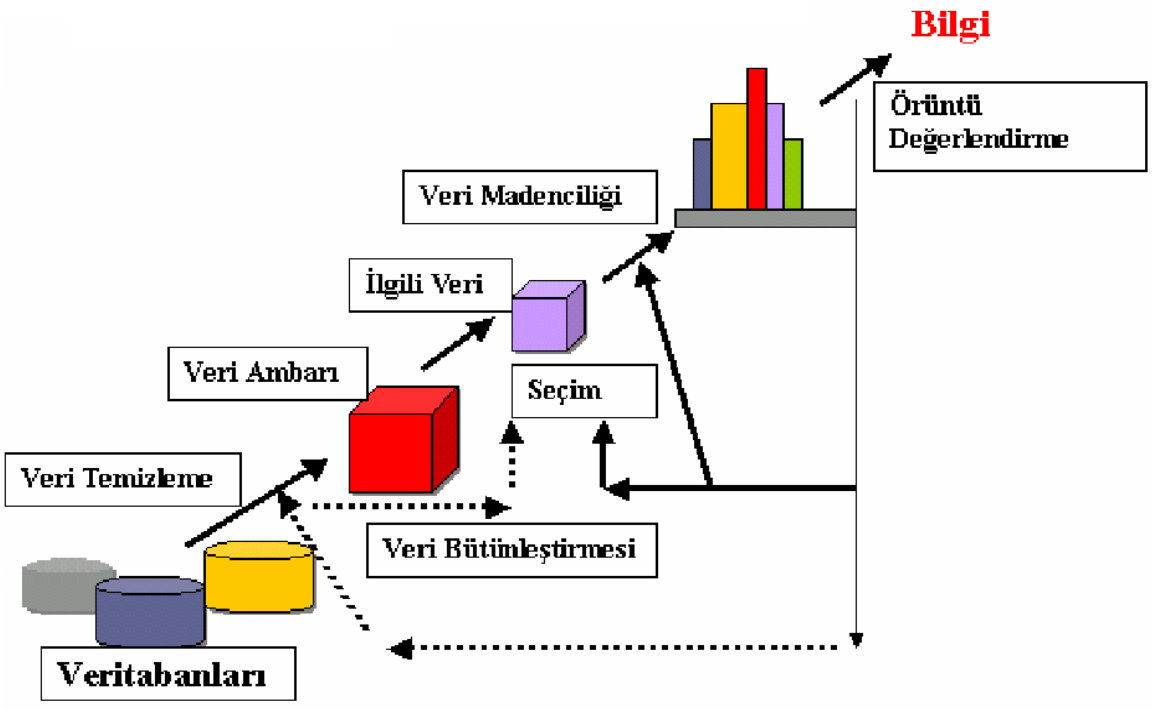
Bu tez çalışmasında, veri madenciliği tekniklerinden biri olan birliktelik kuralı yönteminin detaylı bir şekilde incelenmesi amaçlanmaktadır. Destek ve Güven değerlerinin yanı sıra elde edilen kuralların ilginçliği, birliktelik kurallarında önemli bir parametredir. İlginçlik ölçütü, elde edilen kuralların değerli ve ilginç olması için kullanılan bir parametre olup ilginçliğin literatürde Lift değeri ile değerlendirildiği görülmektedir [39]. Literatürde

bu alanda yapılmış benzer çalışmalar olmasına rağmen, bu tezde ilginçliği yüksek olan kuralların elde edilmesini sağlayacak farklı bir yaklaşım önerilmektedir.

Ayrıca bu tezde, nicel değerler içeren veri setleri üzerinde, destek değerinin seçimi problemine çözüm üretebilecek çoklu minimum destek değeri kullanılması hedeflenmektedir. Destek değerinin, her nitelik için farklı olacak şekilde kullanılması ile ilginç ve değerli kuralların elde edilmesi beklenmektedir. Çoklu destek değerleri kullanılarak, nicel birliktelik kurallarında ilginç kuralların belirlenmesi ile ilgili veri tabanları üzerinde uygulamalar yapılacak ve özellikle farklı ölçek değerine sahip niteliklerde ortaya çıkan destek değeri ile ilgili problemlere çözüm yolu üretilmeye çalışılacaktır. Bu sayede, çoklu destek değeri kullanılarak ilginç kuralların verimli bir şekilde elde edilmesi hedeflenmektedir.

2. VERİ MADENCİLİĞİ

Bilgiye ulaşmanın yanı sıra ulaşılan bu bilgilerden maksimum verim elde etmekte önemli bir hale gelmiştir. Teknolojinin hızla gelişmesiyle oluşan veri yığınlarından yararlı ve gerekli bilgiyi elde etmek için birçok çalışma yapılmaktadır. Veri madenciliği bu safhada göze çarpan bir olgudur. 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.



Şekil 2.1 Veri tabanında bilginin keşfi [1]

Kısaca veri madenciliği; geniş veri tabanlarındaki veriler arasından bilgi çıkarma işlemidir. Kaya ve kum taneleri Veri tabanında bilgi keşfi süreci arasından altın arama işine altın madenciliği denildiği gibi verilerden bilgi keşfetme işine de veri madenciliği

denilmektedir. Veri tabanlarından bilgi keşfi, bilgi çıkarma, veri analizi, veri tarama gibi bazı terimler de veri madenciliği ile benzer anlam taşıyan diğer terimlerdir.

Birçok insan veri madenciliği ile eş anlamlı olarak veri tabanlarında bilgi keşfi terimini kullanmaktadır. VTBK sürecinin nasıl gerçekleştiği Şekil 2.1'de adım adım gösterilmiştir.

- **Veri Temizleme:** Tutarsız ve gürültülü verilerin veri tabanından silinmesidir. Bu aşama, keşfedilecek bilgilerin kalitesini arttıracaktır.
- **Veri Birleştirme:** Birden fazla veri tabanının bulunduğu durumlarda verilerin birleştirilmesidir.
- **Veri Seçimi:** Bu işlem birden fazla veri tabanından sorgulamaya uygun verilerin seçilmesi işlemidir.
- **Veri Madenciliği:** Veriden örüntüler çıkarmak için kullanılan çeşitli yöntemleri içeren en önemli aşamadır.
- **Örüntü/Model Değerlendirme:** Veri tabanından gerçekte ilginç ve doğru olan verilerin tanımlanmasıdır.
- **Bilgi Sunumu:** Tespit edilen ve elde edilen bilgiler, geçerlik, yenilik, kullanışlılık ve basitlik ölçütlerine göre değerlendirilir ve sunulur.

Çok geniş veri tabanlarında saklı olan bilgileri elde etmek için uzun yıllar yapılan çalışmalar sonucunda birçok yöntem geliştirilmiştir. Veri madenciliği, yıllardır özellikle batı ülkelerinin üzerinde çalıştığı bir konu olmasına rağmen, gerçek hayatta yazılım endüstrisinin son yıllarda üretmiş olduğu ileri teknoloji ürünü yazılımlar ile kullanılmaya başlanmıştır. Ayrıca veri madenciliği, makine öğrenmesi, istatistik ve veri tabanı yönetim sistemleri, veri depolama ve paralel programlama gibi çeşitli disiplinlerde kullanılan yaklaşımları birleştirmektedir. Belirli bir sınırda yer alması gerekliliği, veri madenciliği algoritmalarında paralel programlama kullanılması ihtiyacını da beraberinde getirmiştir.

Veri madenciliğinden, verimli sonuçlar elde edebilmek için bazı noktalara dikkat edilmesi gerekir. Bu hususlar aşağıda kısaca bahsedilmiştir.

Farklı tipteki verileri ele alma: Veri tabanı uygulamalarında birçok veri tipi kullanılmaktadır. Örneğin, çoklu ortam verileri, coğrafi bilgi sistemleri, tam sayı tipleri, köklü sayı tipleri gibi. Veri tipinin fazla olması veri madenciliği algoritmalarının verimli çalışmasını zorlaştırmaktadır. Bu nedenle her veri tipine uygun veri madenciliği algoritmaları geliştirilmektedir.

Veri madenciliği algoritmasının etkinliği ve ölçeklenebilirliği: Çok büyük kapasiteli veri tabanından önemli sonuçlar çıkarabilmek için kullanılacak VMA 'nın etkin ve ölçeklenebilir olması gerekir. VMA büyük veri tabanı üzerinde çalışacağı için işlem süresinin minimum seviyede olması istenir. Bu nedenle VMA da çok terimli veya karmaşık bir yapının kullanılmaması gerekir.

Sonuçların yararlılık, kesinlik ve anlamlılık kıstaslarını sağlaması: Çalışma sonucunda ortaya çıkan bilgiler, veri tabanı ile uyumlu olmalıdır. Sonuçları açığa çıkarırken gürültülü ve aykırı veriler dikkate alınmalıdır. Böylece elde edilen kuralların kalitesi arttırılabilir.

Keşfedilen kuralların çeşitli biçimlerde gösterimi: Bu özellik ortaya çıkan sonuçların kullanıcıya anlaşılır bir şekilde sunulmasını sağlar.

Farklı birkaç soyutlama düzeyi ve etkileşimli veri madenciliği: Büyük veri tabanlarından elde edilecek bilgi ile ilgili öngörü yapılması zordur. Bu nedenle kullanıcı veri tabanı üzerinde esnek sorgulama kabiliyetine sahip olmalıdır.

Farklı ortamlarda yer alan veri üzerinde işlem yapabilme: Kurumlar veri tabanlarını yerel ağ üzerinde veya internet ortamındaki herhangi bir bulut veri tabanında saklayabilirler. Bu nedenle veri madenciliği çalışmamızın farklı ortamlarda da çalışabilmesi gerekmektedir. Bu nedenle veri madenciliğinde paralel ve dağıtık veri madenciliği algoritmaları kullanılmaktadır.

Gizlilik ve veri güvenliğinin sağlanması: Kullanılan veri tabanındaki bilgi ve veri madenciliği sonucunda elde edilen sonuçlar sadece erişim hakkına sahip olan kişiler tarafından takip edilmelidir.

Veri madenciliği astronomiden tıpa, biyolojiden pazarlamaya birçok alanda kullanılmaktadır. Örneğin Amerika'da veri madenciliği çalışmaları sayesinde vergi kaçakçılığıyla mücadele edilmektedir [40].

Bununla birlikte günümüzde veri madenciliği teknikleri özellikle işletmelerde çeşitli alanlarda başarı ile kullanılmaktadır. Örneğin işletmelerde, kârlılığı artırmak, müşteri kaybını azaltmak, müşteri potansiyelini artırmak, pazarlama alanında, müşteri ilişkilerini artırmak, uygun ürünleri potansiyel müşterilerle buluşturmak, satışı artırmak için ek kampanyalar oluşturmak, bankacılık, sigortacılık ve borsa alanında, kredi satışı sağlamak, hesap kontrollerini kolaylaştırmak, yeni müşteriler elde etmek, eski müşterileri elde tutmak, poliçe takibi, müşteriye uygun poliçe düzenlemek, hisse senedi takibi ve fiyat tahmininde bulunmak, eğitim alanında, öğrenci başarısı takibi, kişi-branş uyumunu tahmin

etmek, başarı ve başarısızlıkların nedenlerini tahmin etmek, kütüphanecilik alanında, kişi-kitap arası ilişkiyi belirlemek, tarih ve kitap türü tahmini yapmak, teslim süreçlerine göre dağıtım yapmak, spor alanında, oyuncu performans eşleşmesi sağlamak, oyuncu transfer sürecini değerlendirmek, turizm alanında müşteri-mekan-tarih tahmini yapmak, beldelerin fiyat tahmininde bulunmak, endüstri alanında, ürün kalite arasındaki ilişkinin sebeplerinin araştırmak ve tahmin etmek, ürün süreçlerini takip etmek, web alanında, web sitesi – ziyaretçi arası ilişki tahmini yapmak, reklamların hedef kitleye ulaşmasını sağlamak, ziyaretçi müşteri profillerini incelemek, e-ticaret alanında, pazarlama strateji geliştirmek, ziyaretçi davranışları analizi yapmak, telekomünikasyon alanında, kaçak hatların çözümlenmesi ve bölge tahmini yapmak, müşteriye uygun paketler oluşturmak, kullanıcı davranışlarının analizi ile müşteriye özel kampanyalar oluşturmak, kriminoloji alanında, istihbarat birimlerinde tahminde bulunmak, suç eğilimlerini tahmin etmek, gelecekteki suçları azaltmak, sağlık ve tıp alanında ise hasta tanısı tahmini yapabilmek, hastane ve hasta masraflarını düşürmek, iyileşme sürecini analiz etmek, hastalanma sürecini tahmin etmek ve bölgesel salgınlar hakkında analizler yapmak gibi çalışmalarda kullanılabilir.

2.1. Veri Madenciliğinde Karşılaşılan Problemler

Veri madenciliği uygulamalarında kullanılan esas öğe veri tabanı olduğuna göre, veri tabanından kaynaklanabilecek birçok problemle karşılaşmak mümkündür. En azından veri tabanı, normalizasyon şartlarını taşımalıdır. Özellikle küçük veri tabanlarında çalışan sistemler hızlı ve doğru bir biçimde çalışabilirken, çok büyük veri tabanlarında tamamen farklı davranabilir. Veriye gürültü de eklendiğinde, başarım çok kötü bir biçimde etkilenebilir. Veri madenciliği uygulamalarında genel olarak aşağıdaki problemlerle karşılaşılmaktadır [41]:

- Veri tabanının boyutu
- Gürültülü veri
- Boş değerler
- Eksik veri
- Artık veri
- Dinamik veri

2.1.1. Veri Tabanının Boyutu

Veri tabanı boyutları her geçen gün gün çok hızlı bir şekilde artmaktadır. Genelde, makine öğrenmesi algoritmaları az sayıda kayıttan oluşan küçük veri kümeleri ele alabilecek biçimde geliştirilmiştir. Bu algoritmaların binlerce kat büyük kayıtlardan oluşan veri setleri üzerinde kullanılabilmesi oldukça zordur. Bu nedenle veri tabanının büyüklüğü veri madenciliğinde en çok karşılaşılan sorunlardan biridir. Bu sorunu çözebilmek için veri seti üzerinde çeşitli filtreler uygulanarak veya örneklem yoluyla bazı kayıtlar seçilerek veri setinin boyutu küçültülebilir.

2.1.2. Gürültülü Veri

Bir veri tabanında alanların oluşturulurken belirtilen alanın değerleri yanlış girilmiş olabilir. Yanlış veriler genellikle kullanıcı tarafından yanlış girilen bilgiler veya değerin yanlış ölçülmesiyle gerçekleşir. Yanlış veri girişi veya sistem dışı oluşan hatalara gürültü denir. Veri setimizde gürültü mevcut ise sistem bu gürültüyü tespit ederek sistem dışına almalıdır.

Quinlan [42], gürültünün sistem üzerindeki etkisini araştırmak için çeşitli çalışmalar yapmıştır. Bu çalışmalar sonucunda gürültünün sistemi doğrudan etkileyerek başarının düşmesine sebep olduğunu bildirmiştir. Buna karşın veri kümesindeki %10'luk gürültü belirlenebilmektedir. Chan ve Wong [43], gürültünün sistem üzerindeki etkisini analiz etmek için istatiksel metotlar kullanmışlardır..

2.1.3. Boş Değerler

Veri setimizdeki boş değer, kayıt satırında bazı alanlarının içerisinde verilerin olmamasıdır. Eğer bir kayıt kümesinde bazı niteliklerin değeri boş ise bu değerler algoritma içerisinde kullanılamaz. Bu durum veri tabanı ile ilgili işlem yaparken sıklıkla karşılaşılan durumdur. Bazı durumlarda veri alanları boş olabilir. Bu durumda algoritmada yapılacak kontrollerle bu durumun üstesinden gelinebilir. Bu durumda yaklaşımlar şu şekilde sıralanabilir;

1. Veri kümesini görmezlikten gelmek
2. Boş değerleri elle düzeltmek

3. Ortalama değerlerle veya sıklıkla kullanılan değerler ile değiştirmek
4. Modelleme algoritması kullanarak en yakın değeri aktarmak

2.1.4. Eksik Veri

Veri tabanında istenen özelliklere ait bilgilerin ilgili alanda bulunamamasından kaynaklana bir durumdur. Verileri sorgularken istenen sonuçların elde edilememesi gibi durumlar sonuçların hatalı olarak gösterilmesine neden olur.

Her veri kümesi bütün veri tabanlarında aynı sonucu vermeyebilir. Veri kümeleri oluşturulurken bütün şartlar göz önünde bulundurularak oluşturulmalıdır. Örneğin bir hastalığın tanısını koymak için sadece yaşlı hastalardaki belirtiler göz önüne alınırsa bir çocuğa o hastalığın tanısını koymak pek doğru olmaz. Bu gibi durumlarda bilgi keşif modelimiz öngörüsel kararlar alabilmelidir.

2.1.5. Artık Veri

Veri tabanlarında bulunan bazı veriler ihtiyaca gerek olmayan veya artık niteliği taşıyan veriler olabilir. Bu durum veri tabanında sıklıkla karşılan bir durumdur. Problem ihtiyacının değişmesi ve algoritmanın değişmesi ile veri tabanındaki bazı bilgilere ihtiyaç duyulmayabilir. Artık niteliğindeki verilerin elde edilmesi için geliştirilmiş algoritmalara özellik seçimi adı verilir.

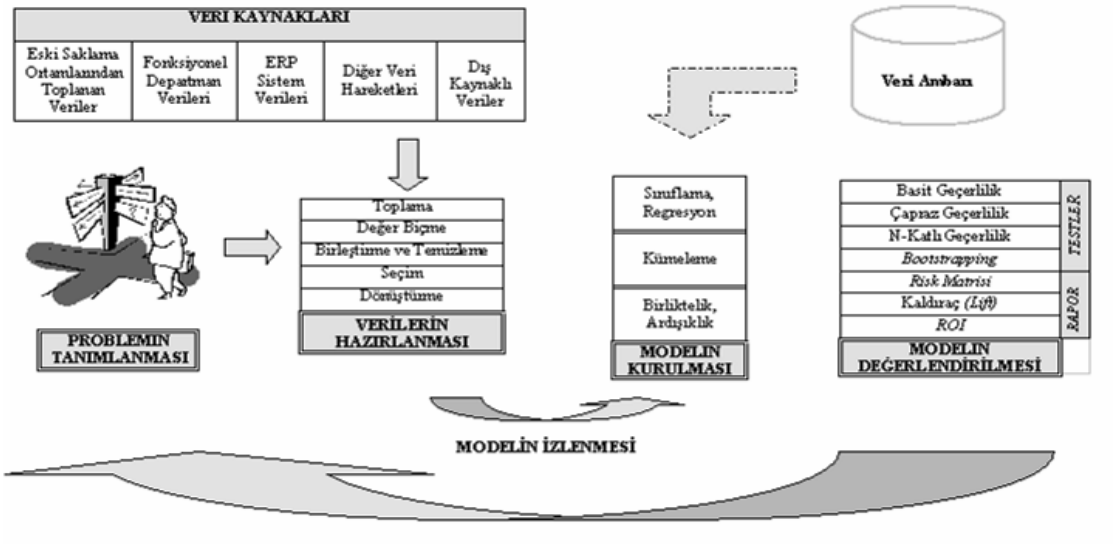
Özellik seçimi veri tabanı işlemleri arama yapılacak olan verilere daha hızlı ulaşmayı ve sınıflandırma niteliklerinin artmasını sağlamaktadır. Özellik seçimi arama kriterlerini küçülterek sınıflama işleminin kalitesinin artırmaktadır.

2.1.6. Dinamik Veri

Aktif bir veri tabanı üzerinde çalışmak çeşit sakıncalar doğurmaktadır. Örneğin çalışan veri tabanı üzerinde VMA uygulanması, kullanılmakta olan uygulamaların performansını düşürebilir. Diğer yandan veri tabanı aktif olarak kullanıldığından veriler değişebilir. Bu durumda elde edilen sonuçların değişen verilerle uyumlu olması gerekir. Bu nedenle VMA sonucunda elde edilen bilgilerin zaman içinde değişen veri tabanına göre kendini güncelleme yeteneğine sahip olması gerekir.

2.2. Veri Tabanlarında Bilgi Keşfi Süreci

Veri madenciliği algoritmasının başarılı olabilmesi için öncelikle yapılan iş ve veri özellikleri hakkında bilgi sahibi olmak gerekir. İş ve veri özellikleri bilinmeden elde edilen sonuçlardan yeterince fayda sağlanması mümkün değildir.



Şekil 2.2. Veri tabanında bilgi keşfi süreci [1]

Şekil 2.2'de ayrıntılı olarak görüldüğü gibi,
Problemin Tanımlanması,
Verilerin Hazırlanması,
Modelin Kurulması ve Değerlendirilmesi,
Modelin Kullanılması,
Modelin İzlenmesi,
veri tabanlarında bilgi keşfi sürecinde izlenmesi gereken temel aşamalardır.

2.2.1. Problemin Tanımlanması

Veri madenciliği çalışmasının başarılı olabilmesi için ilk önce çalışmanın amacı açık bir şekilde belirlenmelidir. Sonrasında çalışma yapılırken bu amaç üzerine yoğunlaşılmalıdır. Çalışma sonunda ortaya çıkan sonuçların ve maliyetlerin işletme üzerindeki etkileri de düşünülmelidir.

2.2.2. Verilerin Hazırlanması

Verilerin hazırlanması, veri madenciliği çalışmasının en önemli bölümlerinden biridir. Modeli oluşturma sürecinde ortaya çıkacak sorunlar bu evrenin sık tekrarlanmasına ve verilerin yeniden düzenlenmesine neden olacaktır. Veri hazırlama, toplama, değer biçme, birleştirme - temizleme, seçme ve dönüştürme aşamalarından oluşur.

2.2.2.1. Toplama

Yapılması gereken çalışma için gerekli kabul edilen veri kaynaklarını ve bu verilerin toplandığı veri kaynaklarını belirleme aşamasıdır. Ayrıca, kuruluşun kendi veri kaynakları dışında verilerin toplanmasında kullanabileceği diğer veritabanlarından da yararlanabilir.

2.2.2.2. Değer Biçme

Veri madenciliğinde, farklı kaynaklardan veri toplama doğal olarak veri uyumsuzluklarına neden olacaktır. Bu tutarsızlıkların başlıca nedeni, farklı zamanlar, kodlama farklılıkları, farklı ölçüm birimleri olabilir. Verilerin hangi koşullarda, ne koşullar altında toplandığı da önemlidir.

Bu sebeplerden dolayı, toplanan verilerin uyumlu olup olmadığı gözden geçirilmeli ve değerlendirilmelidir; çünkü iyi tanımlanmış modeller ancak iyi verilere dayalı olarak oluşturulabilir.

2.2.2.3. Birleştirme ve Temizleme

Bu aşamada, farklı kaynaklardan toplanan verilerde bulunan sorunlar mümkün olduğunca giderilerek tek veri tabanında birleştirilir. Sorun giderme işleminin düzgün yapılmaması daha sonraki aşamalarda daha büyük sorunlar çıkaracağı unutulmamalıdır.

2.2.2.4. Seçim

Bu aşamada, kurulacak modele bağlı olarak veri seçimi yapılır. Modelde, sıra numarası, ID numarası gibi mantıklı olmayan ve diğer değişkenlerin ağırlığının azalmasına neden olan değişkenler modele girilmemelidir. Bazı veri madenciliği algoritmaları, konu ile alâkalı olmayan bu tür değişkenleri otomatik olarak hariç tutmuş olsalar da, pratikte bu işlemin yazılıma bırakılmaması daha mantıklı olur.

Verilerin ve ilişkilerinin görselleştirilmesine izin veren grafiksel araçlar, argümanların seçiminde önemli yararlar sağlayabilir. Yanlış veri girişinden veya belirli bir olayın gerçekleşmesinden kaynaklanan verilerin önemli uyarıcı bilgiler içerdiğini kontrol ettikten sonra veri kümesindeki verileri silmek çoğunlukla tercih edilir.

Modelde kullanılan veri tabanı çok büyükse, tesadüfleri etkilemeyecek bir şekilde örnekleme yapılması uygun olabilir. Günümüzde hesaplama olanakları ne kadar gelişmiş olursa olsun, zaman kısıtlamalarından dolayı çok büyük veritabanlarından çok sayıda model test etmek mümkün değildir. Bu nedenle, tüm veritabanını kullanarak birkaç model denemek yerine, rastgele örneklenen bir veritabanı bölümünde birkaç modelin test edilmesi ve aralarında en güvenilir ve en güçlü modelin seçilmesi daha uygun olacaktır.

2.2.2.5. Dönüştürme

Bir kredi riskini tahmin etmek için geliştirilen bir modelde önceden hesaplanmış bir borç / gelir oranı yerine ayrı borç ve gelir verilerini kullanmak tercih edilebilir. Buna ek olarak, modelde kullanılan algoritma, verilerin görüntülenmesinde önemli bir rol oynayacaktır. Örneğin, bir uygulamada Yapay Sinir Ağı algoritması (YSA) kullanılıyorsa, kategorik değişken değerleri evet / hayır veya karar ağacı algoritması kullanılıyorsa, gelir değişkeni değerleri yüksek / orta / düşük olarak gruplandırılabilir.

2.2.3. Modelin Kurulması ve Değerlendirilmesi

Mümkün olduğunca çok sayıda model deneyerek, amaca en uygun modelin bulunması mümkündür. Bu nedenle, veri hazırlama ve model kurma aşamaları, amaca en uygun bulunana kadar tekrarlanan bir süreçtir.

Model kurulum süreci, denetlenen ve denetlenmeyen öğrenme için kullanılan modellere göre farklılık gösterir.

Ayrıca, önceden belirlenmiş ölçütlere göre sınıflara ayrılan denetimli öğrenmede her sınıf için çeşitli örnekler verilmektedir. Böylelikle verilen örneklere göre her sınıf için belirli özellikler bulunur ve bu özelliklere göre de kural cümleleri oluşturulabilir.

Öğrenme işlemi tamamlandığında tanımlanan kurallar verilen yeni örneklere uygulanır ve yeni örneklerin hangi sınıfa ait olduğu kurulu model tarafından belirlenir.

Denetimsiz öğrenme, kümeleme analizinde olduğu gibi ilgili örnekleri ve bu örneklerin özellikleri arasındaki benzerlikleri gözlemleyerek sınıfları belirlemeyi amaçlamaktadır.

Denetimsiz öğrenmede, kümeleme analizinde olduğu gibi ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır.

Denetlenen öğrenme için seçilen algoritma uyarınca ilgili verileri hazırladıktan sonra, bazı veriler modelin öğrenmesine ayrılırken diğer kısmı ise modelin geçerliliğini test etmek için ayrılır. Eğitim setini kullanarak model öğrenildikten sonra, modelin doğruluk seviyesi test seti ile belirlenir.

Kullanılacak modelin değerlendirilmesindeki bir diğer yöntem ise, modelin uygulaması sonucunda elde edilecek kazanımların, model uygulanırken oluşacak maliyetlere bölünmesiyle oluşacak orandır.

Kullanılan modelin doğruluk derecesi ne kadar yüksek olsa da uygulama esnasında oluşan bazı sorunlar modelin doğruluk derecesini azaltmaktadır. Yapılan çalışmalar sonucunda verimliliğin azalmasındaki başlıca sebepler, model oluşturulurken kabul edilen varsayımlar ve veri analizi için kullanılan verilerin doğru olmamasıdır. Örneğin modelin kurulması sırasında varsayılan faiz oranının zaman içerisinde değişmesi, sistemin çalışmasına olumsuz etki edecektir.

2.2.4. Modelin Kullanılması

Oluşturulan ve geçerliliği onaylanmış olan model, doğrudan bir uygulama veya başka bir uygulamanın alt kümesi olabilir. Belirlenen modeller, risk analizi, kredi değerlendirme, dolandırıcılık tespiti gibi çalışmalarda doğrudan kullanılabilir, promosyon amaçlı bir planlama simülasyonuna entegre edilebilir veya tahmin edilen stok seviyeleri belirlenen değerin altına düştüğünde siparişleri otomatik olarak verecek bir uygulamaya yerleştirilebilir.

2.2.5. Modelin İzlenmesi

Tüm sistemlerin zaman içindeki özelliklerinde meydana gelen değişiklikler ve böylece ürettikleri veriler sürekli izlenmeyi ve gerekirse kurulan modellerin yeniden düzenlenmesini gerektirir. Tahmini ve gözlemlenen değişkenler arasındaki farkı gösteren grafikler, model sonuçlarının izlenmesi için yararlı bir yöntemdir.

2.3. Veri Madenciliği Modelleri

Veri madenciliği modelleri temel olarak iki ana bölümde incelenmektedir. Birincisi bilinmeyen verilerin tahmininde kullanılan öngörme modeli ve diğeri eldeki verilerin tanımlanması için tanımlayıcı modeldir [1].

Öngörü yapan modellerde, sonuçları bilinen veriler kullanılarak bir model geliştirilir. Oluşturulan bu model kullanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerleri ile ilgili öngörü yapılması amaçlanmaktadır. Örneğin bir banka, daha önceki dönemlerde müşterilerine verdiği tüm kredilerle ilgili bilgilere sahiptir. Bu bilgileri kullanarak daha sonraki dönemlerde müşterilere vereceği kredinin geri dönüp dönmeyeceğini müşteri bilgilerini kullanarak öngörü de bulunabilir.

Tanımlayıcı modeller ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanmasını sağlamaktadır. Belirli özelliklere sahip insanların bazı davranışlarının birbirine benzerlik göstermesi tanımlayıcı modele bir örnek olabilir.

Veri madenciliği modellerini gördükleri işlevlere göre ise üç ana başlık altında incelemek mümkündür [44]. Bunlar;

- Sınıflama ve Regresyon,
- Kümeleme,
- Birliktelik kuralları ve ardışık zamanlı örüntülerdir.

2.3.1. Sınıflama ve Regresyon Modelleri

Sınıflama ve regresyon, veri madenciliği tekniklerinde en çok kullanılan yöntemlerden biridir. Mevcut verilerden hareket ederek gelecekteki durumlar ile ilgili öngörü yapılması durumunda faydalanılır ve yeni bir veri elemanını daha önceden belirlenmiş sınıflara atamayı amaçlar. . Sınıflama ve regresyon arasındaki temel fark tahmin edilen bağımlı değişkenin kategorik veya sürekli bir değere sahip olmasıdır. Bununla birlikte, her iki model yavaş yavaş birbirlerine yaklaşır ve sonuç olarak aynı teknikleri kullanmak mümkündür. Sınıflandırma ve regresyon modelinde kullanılan temel teknikler şunlardır;

- Karar Ağaçları
- Yapay Sinir Ağları
- Genetik Algoritmalar
- K-En Yakın Komşu
- Bellek Temelli Sınıflandırma
- Lojistik Regresyon

Karar ağaçları tekniği, çok güçlü bir sınıflandırma ve öngörü aracıdır. Denetimli öğrenmenin kullanıldığı veri madenciliği tekniklerinden biridir. Diğer veri madenciliği tekniklerine nazaran çok daha anlaşılır bir dile sahiptir. Örneğin; kredi kartı başvurusunda bulunan bir müşteri için başvurusunun reddedilmesinin sebebinin *gelir<1 milyar* ve *kredi kartı sayısı<3* olması yeteri kadar açıklayıcı olacaktır. Ayrıca, modelin başarısı kadar, başarılı ve başarısız bir modelin nasıl çalıştığını araştırması da bu tekniği diğer tekniklere göre farklı kılmaktadır.

YSA; tanınması istenen nesnenin öznitelik vektörünü giriş olarak alan ve çıkış ünitelerinin birinde bu nesnenin sınıfını belirleyen bir cevap üreten, pek çok doğrusal olmayan hesaplama elemanlarının paralel işleyişinden meydana gelmiş tümleşik bir yapıdır. YSA'da her çıkış ünitesi gözlenen olayın farklı bir sınıfını belirler. YSA'nın paralel yapıları, bilgisayarları geleneksel yöntemlerden çok daha farklı kullanarak özellikle seri bilgisayarlarda bilinen yöntemlerle yapılması mümkün olmayan ve çok zor olan bir takım işlevleri (ses ve örüntü tanıma gibi) rahatlıkla yapmaları, YSA'yı üstün kılmıştır [45].

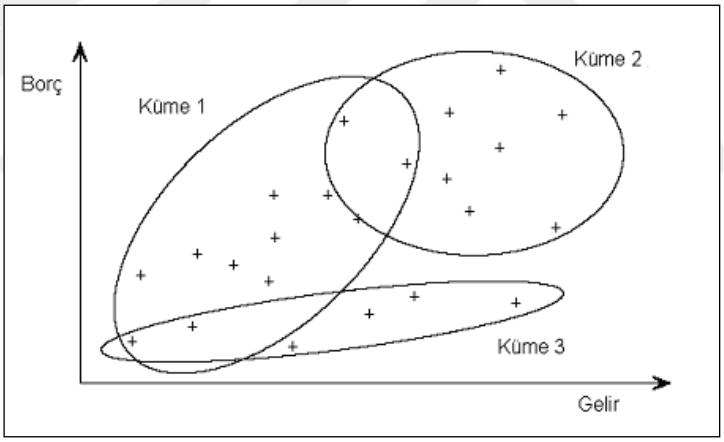
En yakın komşu metodu; Nesne tanımanın en klasik yöntemlerinden biridir ve veritabanındaki en yakın komşu sınıfına dahil edilecek nesnenin vektörünü tanımlar.

Yöntem örnek vektörün istatistiksel dağılımından bağımsızdır ve yalnızca en yakın komşunun sınıfına göre sınıflandırarak tanımayı yapar.

Bellek tabanlı yöntemler; denetimli öğrenmenin kullanıldığı veri madenciliği tekniklerindendir. Bu tekniğin temel özelliği, daha önceki deneyimlerimizden faydalanarak elimizdeki problemlere benzer durumları tanımlayıp geçmiş benzer problemlere getirdiğimiz uygun çözümleri mevcut problemlerimize uygulamaya çalışmaktır.

2.3.2. Kümeleme

Kümeleme, veri tabanındaki verileri alt kümelere ayıran bir yöntemdir. Her bir kümede yer alan elemanlar birbirlerine çok benzemekte, özellikleri farklı olan elemanlar ise farklı kümelerde bulunmaktadır. Başlangıçta, veri tabanında hangi kümelerin sıralanacağı veya hangi kümelenme özelliklerine sahip olacağı bilinmemekte ve bu özellikler konunun uzmanı olan bir kişi tarafından belirlenmektedir.



Şekil 2.3. Kümeleme

Kümeleme analizinde amaç birbirine en çok benzeyen nesneleri aynı grupta toplamaktır. Benzemekten kasıt, geometrik anlamda uzaklık olarak birbirine en yakın nesnelerin seçilmesidir. Bu nedenle nesnelerin sayısal değer olması gerekir. Bu noktada değişkenler dörde ayrılmaktadır [46]. Bunlar;

- Kategorik Değişkenler: Bu değişkenler arasında sadece birbirine benzeme söz konusudur. Sıralama mümkün değildir. (örneğin siyah>beyaz durumu söz konusu değildir.
- Sıralama Değişkenleri: $X>Y$ şeklinde bir sıralama yapmak mümkündür. Ama büyüklüğün ne kadar olduğu belli değildir. (X-Y bulunamaz)

- Aralık Ölçekli Değişkenler: İki nokta arasındaki uzaklık hesaplanır. Fakat bu tür değişkenlerde gerçek "0" değeri yoktur.
- Oran Ölçekli Değişkenler: Anlamlı "0" noktasının bulunduğu, her türlü dört işleme açık değişkenler topluluğudur. (örneğin 20 yaşındaki bir kişi 10 yaşındaki bir kişiden iki katı yaştadır denilebilir.)

Kategorik ve sıralama değişkenlerinin matematiksel hesaplamaların yapılabileceği sayısal değerlere dönüştürülmesi şarttır.

2.3.3. Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler

Birliktelik kuralı, bir veri tabanında bulunan ürün ve işlem birlikteliklerini inceleyerek, sık tekrarlanan ürün birlikteliklerini ortaya çıkarma işlemidir [47]. Bir müşterinin bir alışverişte veya ardışık satın alımlarda satın almak istediği malların veya hizmetlerin tahmin edilmesi, müşteriye daha fazla ürün satma yöntemlerinden biridir. Pazarlama amaçları için pazar sepeti analizi adı altında veri madenciliğinde, satın alma trendlerinin tanımlanmasını sağlayan uyum kuralları ve ardışık düzenler yaygın olarak kullanılmaktadır.

Eş zamanlı olarak ortaya çıkan ilişkileri tanımlamak için birliktelik kuralları kullanılır. Örneğin, bebek bezi satın alan müşterilerin %75 ihtimalle ıslak mendil de almaları veya ekmek ve yağ alanların %90'ının süt de satın almaları birliktelik kuralları kapsamında tespit edilebilir.

Sıralı zaman örüntüleri, birbiriyle ilişkili olan ancak birbirini izleyen dönemlerde ortaya çıkan ilişkileri tanımlamak için kullanılır. A operasyonu yapılan bir hastada, 15 gün içerisinde %45 ihtimalle B enfeksiyonu oluşması, çekiç alan bir müşterinin ilerleyen zamanlarda %15 ihtimalle çivi alması ardışık zamanlı örüntüler olarak tanımlanmaktadır.

3. BİRLİKTELİK KURALI

3.1. Birliktelik Kuralı Tanımı

Günümüzde, birçok alandaki veri bulut sistemleri gibi verinin yoğun olduğu teknolojilerde, server bilgisayarlarda veya kişisel bilgisayarlarda, veri tabanları üzerinde saklanmaktadır. Bu verilerden istenen ve faydalı bilgiye ulaşmak için kullanılan tekniklerden biri de birliktelik kurallarıdır. Birliktelik kuralı, pek çok alanda yaygın şekilde kullanılmaktadır ve nesnelerin veya niteliklerin bir arada bulunma durumlarını belirlemek için kullanılır. Birliktelik kuralı bulma süreci, büyük veritabanlarında uygulanması çok pahalı bir işlem olan YNK hesaplamasına dayanan bir işlemdir. Bu nedenle önceden belirlenmiş birliktelik kurallarını korumak ta çok önemli bir konudur.

Son zamanlarda gelişen teknoloji ile birlikte firmalar satış noktalarında barkod sistemlerini yaygın olarak kullanmaya başladılar. Bu sayede firmaya ait bütün bilgilerin elektronik ortama aktarılması sağlanmıştır. Genellikle büyük süpermarketlerde oluşan bu tür verilere market sepet verisi adı verilmektedir. Birçok kuruluş, market sepet verilerini kullanarak bu verilerden büyük faydalar sağlamayı amaçlamaktadır.

Market sepet verisi üzerinde birliktelik kuralı problemi, ilk olarak 1993 yılında ele alınmıştır. Sepet analizinde amaç, nitelikler (ürün satışları) arasındaki ilişkiyi bulmaktır. Bu ilişkilerin bilinmesi şirketin kârını arttırmak için kullanılabilir. Eğer *X* malını alan müşterilerin *Y* malını da çok yüksek bir olasılıkla aldıkları biliniyorsa veya bir müşteri *X* malını alıyor ama *Y* malını almıyorsa o potansiyel bir *Y* müşterisidir. Sepet analizi günlük işlemler sonucu elde edilen verilerden anlamlı bağıntılar çıkarmada kullanılır. "Eğer *A* malını alıyorlarsa *%x* ihtimalle *B* malını almaya da meyillidirler" şeklinde bir sonuç *A* malını satan bir mağaza için çok faydalı bir bilgi olabilmektedir. Sepet analizi uygulamaları; çapraz satış (cross-selling), mağaza raflarının düzenlenmesi (layout), katalog tasarımı ve fiyatlandırma (pricing) gibi alanlarda kullanılmaktadır.

Birliktelik kuralının matematiksel ifadesi Agrawal ve ark. [2] tarafından tanımlanmıştır. Bu modelde M = {M1,M2,..,M_n} ürün kodlarını; *D*, ürün işlemlerini; *T*, bir işlemdeki ürün kodlarını; ($T \subseteq M$), *tid* her işleme ait birincil anahtarı, *k*-nesne kümesi, *k* adet ürünü içeren kümeyi temsil etmektedir. *X* bir ürün kümesi olmak üzere *T* işlemi *X* ürün kümesini ancak ve ancak $X \subseteq T$ şartını sağlıyorsa içerir. *X* ve *Y*, M ürün kümesinin bir

alt kümesi ve $X \cap Y = \varnothing$ olsun. O zaman, bir birliktelik kuralı $X \rightarrow Y$ şeklinde bir ifade ile gösterilebilir. Bu ifade ile *X, Y*'yi belirler (*X* ürününü içeren işlem kümesi *Y*'yi içeren işlem kümesi içinde kapsanır) veya *Y, X*'e bağımlıdır denir (*Y* kümesinin varlığı *X* kümesinin var olmasına bağlıdır). İşlem numaraları gruplandırılarak elde edilen ürünler arasındaki bağımlılık ilişkisinin %100 doğru olması beklenemez. Benzer şekilde, çıkarılmakta olan kuralın eldeki işlem kümesi tarafından desteklenmesi istenir. Bu nedenlerden dolayı, $X \rightarrow Y$ eşleştirme kuralı kullanıcı tarafından minimum değeri belirlenmiş güvenilirlik (*c:confidence*) ve destek (*s:support*) eşik değerlerini sağlayacak biçimde üretilir. $X \rightarrow Y$ eşleştirme kuralına, *c* güvenilirlik ölçütü ve *s* destek ölçütü iliştirilir ve biçimsel olarak $\Phi(D) = <X \rightarrow Y, c, s>$ ile gösterilir. Burada *D*, örneklemi; $X \rightarrow Y$ ifadesi eşleştirme kuralını; *c* eşik değeri, ilgili kuralın minimum güvenirliğini; *s* eşik değeri ise ilgili kuralın minimum destek değerini göstermektedir. Sepet analizi yapılırken ürünler arasındaki ilişki destek ve güven değerleri ile hesaplanmaktadır. Destek ve güven değerlerinin tanımları aşağıdaki gibidir.

$$\text{Destek :} \quad P(X \text{ ve } Y) = \frac{X \text{ ve } Y \text{ mallarını satın alan müşterilerin sayısı}}{\text{Toplam müşteri sayısı}} \tag{3.1}$$

$$\text{Güven :} \quad P(X/Y) = \frac{P(X \text{ ve } Y)\ \{X \text{ ve } Y \text{ mallarını satın alanların sayısı}\}}{P(Y)\ \{Y \text{ malını satın alanların Sayısı}\}} \tag{3.2}$$

Destek değeri, ürünler arasındaki ilişkinin veride ne sıklıkta olduğunu,güven değeri ise Y ürününü alan bir kişinin X ürününü alma ihtimalini gösterir.

Destek kriteri, veri içerisinde ürünler arasındaki bağıntının ne kadar sık olduğunu, güven kriteri ise *Y* ürününü almış olan bir kişinin hangi olasılıkla *X* ürününü alacağını gösterir. İki ürün arasında ilişkinin önemli olması için hem destek hem de güven ölçütleri mümkün olduğunca yüksek olmalıdır.

Sepet analizi yapılırken bazı problemlerle karşılaşmak mümkündür. Bunlardan en önemlisi, bu tekniğin analiz sonunda bulunan bağıntının rastlantısal olup olmadığını anlayamamasıdır. Bu duruma bir örnek verecek olursak, bir markette tuz alan müşterilerin %40'ının şeker de almakta olduğunu düşünelim. Bu bilgi ilk bakışta ilginç görünse de çok anlamlı olmayabilir. Çünkü marketten alışveriş yapan insanların belki de %40'ı zaten şeker alıyordur, dolayısı ile tuz ve şeker arasındaki bağıntı sadece rastlantısaldır. Alışveriş

yapanların sadece %15'i şeker alırken, tuz alanların % 40'ının şeker almasının bilgisi analizin sonucunu çok daha farklı etkileyecektir. Bu duruma ürünlerin birbirini çekmesi denilmektedir. Başka bir durumda şeker alanlar tüm alışveriş verisinin %65'ini oluştururken tuz alanların sadece %40'ının şeker almasının bilgisi bu ürünlerin birbirini ittiğini gösterir. Karşılaşılan problemlerin çoğu sepet analizi tekniğinin bağıntılar arasındaki rastlantısallığının derecesini anlamıyor olmasından kaynaklanmaktadır. Özetle, sepet analizi tekniğinin güçlü ve zayıf yönlerini sıralamak gerekirse;

Güçlü yönleri;

- Denetimsiz veri madenciliği yöntemini başarı ile uygular
- Anlaşılabilir sonuçlar verir

Zayıf yönleri;

- İleri derecede bilgisayar performansına bağımlıdır
- Arasında bağıntı oluşturacak ürünlerin doğru olarak seçilmesi, analiz süreci için oldukça önemlidir.

Birliktelik kuralı çıkarmada kullanılan bazı algoritmalar mevcuttur. Minimum güvenirlik ve destek değerlerini sağlayan birliktelik kuralı çıkarma işlemi iki adıma ayrılmıştır [2, 3].

İlk adım, kullanıcı tarafından belirlenmiş minimum destek değerini sağlayan ürünlerin bulunmasıdır. Bulunan bu ürünlere sık geçen nesne kümesi veya yoğun nesne kümesi adı verilmektedir. Verilen örneklemde N adet ürün (nesne) var ise, potansiyel olarak 2^N adet yoğun nesne kümesi olabilir. Bu adımda üstel arama uzayını etkili biçimde tarayarak yoğun nesne kümelerini bulan etkili yöntemler kullanılmalıdır.

İkinci adım ise yoğun nesne kümelerinin incelenerek minimum güven değerini sağlayan birliktelik kurallarının bulunmasıdır. Bu işlem oldukça basittir ve şu şekilde yapılmaktadır. Yoğun olan her l nesne kümesi için, boş olmayan l'nin tüm alt kümeleri üretilir. l'in boş olmayan alt kümeleri a ile gösterilsin. Her a kümesi için $a \Rightarrow (l - a)$ gerektirmesi, l kümesinin destek ölçütünün a kümesinin destek ölçütüne oranı minimum güvenirlik eşiği ölçütünü sağlıyorsa $a \Rightarrow (l - a)$ birliktelik kuralı olarak üretilir. Böylece minimum destek eşiğine göre üretilen çözüm uzayında, minimum güvenirlik eşiğine göre taranarak bulunan birliktelikler kullanıcının ilgilendiği ve potansiyel olarak önemli bilgi içeren birlikteliklerdir.

Birliktelik kuralı işleminin başarısını birinci adım belirler. YNK belirlendikten sonra, birliktelik kurallarının bulunması ise basit bir işlemdir [48].

3.2. Apriori Algoritması

Yoğun nesne kümelerini bulmak için birçok kez veri tabanını taramak gerekir. İlk taramada bir elemanlı minimum destek değerini sağlayan yoğun nesne kümeleri bulunur. İzleyen taramalarda bir önceki taramada bulunan yoğun nesne kümeleri, aday kümeler adı verilen yeni potansiyel yoğun nesne kümelerini üretmek için kullanılır. Aday kümelerin destek değerleri tarama sırasında hesaplanır ve aday kümelerinden minimum destek değerini sağlayan kümeler o geçişte üretilen yoğun nesne kümeleri olur. Yoğun nesne kümeleri bir sonraki geçiş için aday küme olurlar. Bu süreç yeni bir YNK bulunmayana kadar devam eder.

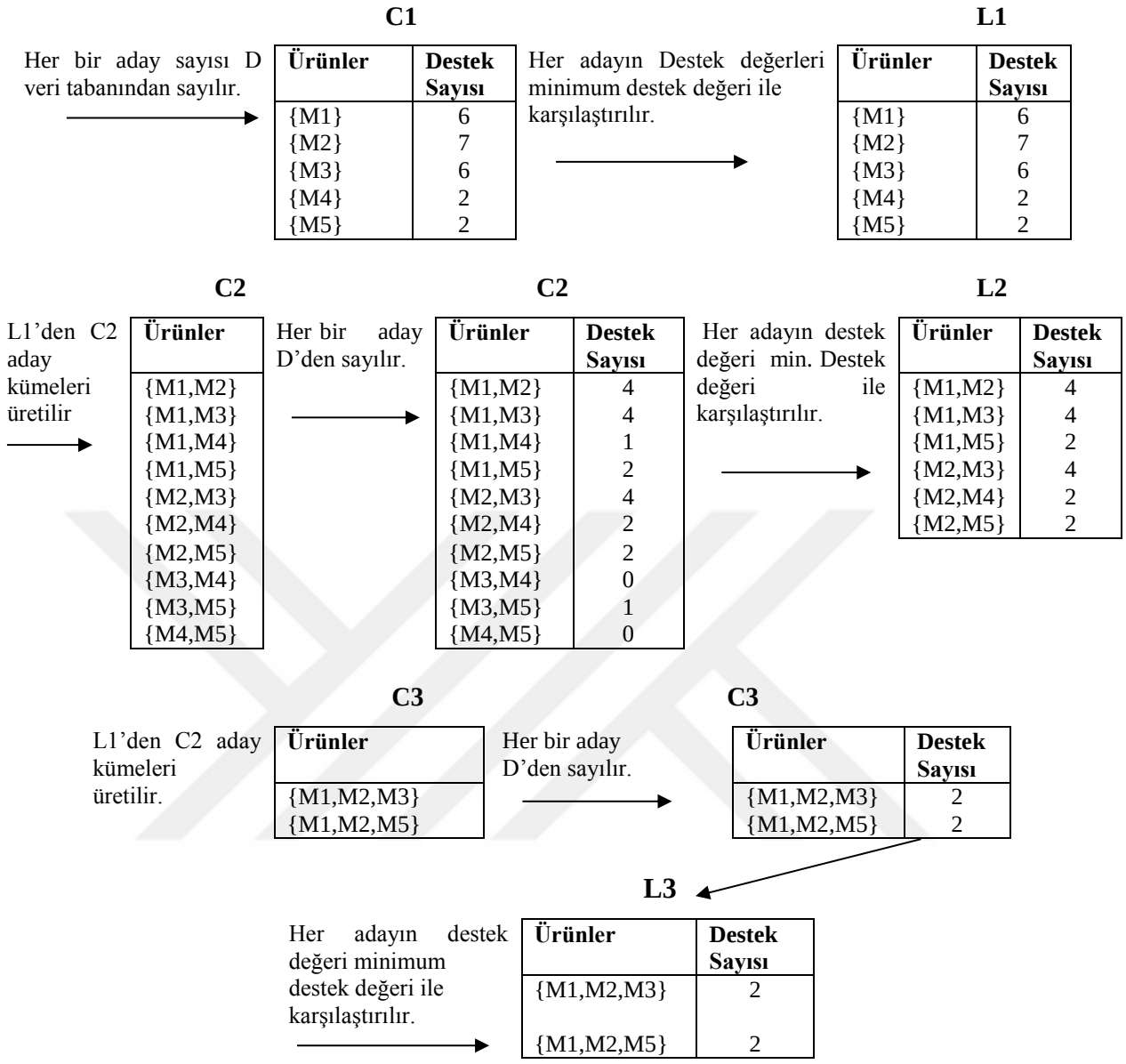
Bu algoritmada temel yaklaşım eğer k-nesne kümesi minimum destek değerini sağlıyorsa bu kümenin alt kümeleri de minimum destek değerini sağlar şeklindedir.

Apriori algoritmasında önce aday nesne kümeleri oluşturulur. Bu kümeler potansiyel olarak yoğun nesne kümeleridir. *C* ile gösterilir ve *C1*, *C2*, *C3*, …..*Ck* olarak k-nesne kümesini oluştururlar. Her *Ck* nesne kümesi *C(k-1)* nesne kümesini içerir ve *C1* < *C2* < *C3*....< *Ck* şeklinde sıralıdır. Yoğun olan k-nesne kümeleri ise *L* ile gösterilir ve minimum destek kıstaslarını sağlarlar.

Tablo 3.1. Bir marketten satın alınan ürünler listesi

İşlemler	**Satın Alınan Ürün Listesi**
T1	M1, M2, M5
T2	M2, M4
T3	M2, M3
T4	M1, M2, M4
T5	M1, M3
T6	M2, M3
T7	M1, M3
T8	M1, M2, M3, M5
T9	M1, M2, M3

Örnek: Tablo 3.1'de bir firmada satın alınmış ürünlerle ilgili bir veri tabanı görülmektedir. Bu veri tabanını *D* olarak adlandırılsın. Veri tabanında 9 adet işlem görüldüğüne göre |*D*| =*9* olmaktadır. Bu durumda *D* veri tabanı üzerinde Apriori algoritması kullanılarak yoğun nesnelerin nasıl bulunabileceği Şekil 3.1'de görülmektedir.



Şekil 3.1. Apriori algoritması ile nesne kümelerini belirlenmesi (Minimum destek değeri 2).

1) Algoritmanın ilk iterasyonunda her nesne, yani bir elemanlı nesneler kümesi C1 olarak elde edilir. Algoritma, her nesnenin sayısını bulmak için veri tabanını basitçe taramaktadır.

2) Bu veri tabanı için minimum destek sayısı 2 olarak belirlenmiştir. Minimum destek değeri 2/9 = %22'dir. Buradan bir elemanlı nesneler kümesi olan L1 elde edilebilir. C1 deki tüm nesneler minimum destek değerini sağlamakta ve L1 elde edilmektedir.

3) L2'yi bulamak için önce iki elemanlı aday nesne kümesini elde etmek gerekir ve L1xL1 den C2 elde edilir. L1'in ikili kombinasyonları C2'nin eleman sayısını vermektedir.

4) Daha sonraki işlem D'yi taramak ve C2 deki her aday nesnenin minimum destek değerini bulmaktır.

5) C2'de minimum destek değerine sahip olmayan aday nesnelerin çıkarılması ile L2 nesne kümesi elde edilir.

6) 3 elamanlı aday nesne kümesinin elde edilmesi daha sonraki bölümde detaylı olarak anlatılacaktır. Normalde C3=L2xL2 ={{M1,M2,M3}, {M1,M2,M5}, {M1,M3,M5}, {M2,M3,M4}, {M2,M3,M5}, {M2,M4,M5}}olarak elde edilir. Bunların hepsi sık tekrarlanıyor görülebilir ancak Apriori algoritması özelliğine göre son dört tanesi sık tekrarlanmıyor olabilir. Bu nedenle son dördünü C3'ten çıkarabiliriz. Böylece D'yi taramak ve L3'ü belirlemek için bazı işlemler boşu boşuna yapılmamış olacaktır.

7) C3 üzerinde minimum destek değerine sahip olmayan aday nesnelerin çıkarılması ile L3 nesne kümesi elde edilir.

8) Algoritmada L3xL3'ü kullanılarak 4 elemanlı aday nesne kümesi elde edilir. Bunun sonucunda {M1,M2,M3,M5} elde edilmesine rağmen bu kümenin alt kümesi olan {M2,M3,M5} L3'te bulunmamaktadır. Bu nedenle C4=0'dır ve tüm sık kullanılan birliktelikler elde edilerek algoritma sona ermektedir.

3.2.1. Apriori ile L2 Kullanılarak C3 Oluşturulması

Apriori algoritmasında L(k-1)'ler kullanılarak Ck elde edilmektedir. k>2 için Ck'ların elde edilmesi, k=3 için aşağıda detaylı olarak anlatılmıştır [49].

1) C3=L2xL2 ={{M1,M2}, {M1,M3}, {M1,M5}, {M2,M3}, {M2,M4}, {M2,M5}} X {{M1,M2}, {M1,M3}, {M1,M5}, {M2,M3}, {M2,M4}, {M2,M5}} = {{M1,M2,M3}, {M1,M2,M5}, {M1,M3,M5}, {M2,M3,M4}, {M2,M3,M5}, {M2,M4,M5}}.

2) Elde edilen aday nesne kümelerinden bazıları kullanılmayabilir. Çünkü bunlar Apriori'ye göre yoğun nesneler değildir. Bunu elde etmek için bir eleme işlemi gerçekleştirilir.

3) {M1,M2,M3} kümesinin iki elemanlı alt kümeleri {M1,M2}, {M1,M3} ve {M2,M3} tür. Bu iki elamanlı alt kümelerin tümü L2'nin bir nesnesi olduğuna göre {M1,M2,M3} C3'ün bir aday nesnesi olabilir.

4) {M1,M2,M5} kümesinin iki elemanlı alt kümeleri {M1,M2}, {M1,M5} ve {M2,M5} tir. Bu iki elamanlı alt kümelerin tümü L2'nin bir nesnesi olduğuna göre {M1,M2,M5} C3'ün bir aday nesnesi olabilir.

5) {M1,M3,M5} kümesinin iki elemanlı alt kümeleri {M1,M3}, {M1,M5} ve {M3,M5} tir. Bu iki elamanlı alt kümelerden {M3,M5}, L2'nin bir nesnesi değildir ve bu nedenle sık tekrarlı değildir. Bundan dolayı {M1,M3,M5} C3'ün aday nesneliğinden çıkarılmalıdır.

6) {M2,M3,M4} kümesinin ün iki elemanlı alt kümeleri {M2,M3}, {M2,M4} ve {M3,M4} tür. Bu iki elamanlı alt kümelerden {M3,M4}, L2'nin bir nesnesi değildir ve bu nedenle sık tekrarlı değildir. Bundan dolayı {M1,M3,M5} C3'ün aday nesneliğinden çıkarılmalıdır.

7) {M2,M3,M5} kümesinin iki elemanlı alt kümeleri {M2,M3}, {M2,M5} ve {M3,M5} tir. Bu iki elamanlı alt kümelerden {M3,M5}, L2'nin bir nesnesi değildir ve bu nedenle sık tekrarlı değildir. Bundan dolayı {M1,M3,M5} C3'ün aday nesneliğinden çıkarılmalıdır.

8) {M2,M4,M5} kümesinin iki elemanlı alt kümeleri {M2,M4}, {M2,M5} ve {M4,M5} tir. Bu iki elamanlı alt kümelerden {M4,M5}, L2'nin bir nesnesi değildir ve bu nedenle sık tekrarlı değildir. Bundan dolayı {M1,M3,M5} C3'ün aday nesneliğinden çıkarılmalıdır.

9) Sonuç olarak eleme sonucunda C3= {{M1,M2,M3}, {M1,M2,M5}} elde edilir.

3.2.2. Algoritma

Giriş: Database (*D*), minimum destek *min_sup*.

Çıkış: *L*, D de yoğun olan nesne kümeleri

```
(1)     L1=find_frequent_1-itemsets(D)
(2)     for (k=2;L(k-1)≠0;k++) {
(3)         Ck=apriori_gen(L(k-1), min_sup)
(4)         for each transaction t∈D{   //sayım için D yi tara
(5)             Ct=subset(Ck,t);
(6)             for each candidate c∈Ct
(7)                 c.count++;
(8)         }
(9)         Lk={ c∈Ck | c.count ≥min_sup }
(10)    }
(11)    return L=Uk Lk;
```

```
procedure apriori_gen (L(k-1);min_sup)
(1)             for each itemset l1∈L(k-1)
(2)         for each itemset l2∈L(k-1)
(3)             if (l1[1]= l2[1] )∧(l1[2]=l2[2]) ∧…∧ (l1[k-1]=l2[k-1]) then {
(4)                 c=l1x l2;  // aday nesnelerin üretilmesi
(5)                 if has_infrequent_subset(c,L(k-1)) then
(6)                         delete c; // gereksiz adayların silinmesi
(7)                 else add c to Ck;
(8)             }
(9)             return Ck;

procedure has_infrequent_subset(c,L(k-1));
(1)             for each (k-1)-subset s of c
(2)         if s∉L(k-1) then
(3)             return TRUE
(4)             return FALSE
```

Yukarıda Apriori algoritması ve bu algoritmaya ait alt programlar görülmektedir. Algoritmanın 1. satırında 1-elemanlı aday nesneler yani *L1* bulunmaktadır. 2 ile 10. satırlar arasında ise Lk elde etmek amacı ile *L(k-1)* kullanılarak *Ck*'lar elde edilmektedir. Bu satırlarda bulunan **apriori_gen** alt programı (3. satır) ise aday nesneleri bulmakta ve bazı yoğun olmayan nesne kümelerini tespit ederek eleme yapmaktadır. 4. satırda ilk önce üretilen adaylar veri tabanından taranmakta ve 5. satırda ise aday nesnelerin tüm alt kümeleri bulunması için **subset** fonksiyonu kullanılmaktadır. 6 ve 7. satırlarda da bu adayların her birinin sayısı toplanmaktadır. Sonuç olarak, minimum destek değerine sahip tüm yoğun nesne kümeleri (*L*) elde edilmektedir.

Apriori_gen alt programı 2 çeşit görev yerine getirmektedir. Bunlardan birincisi aday nesne oluşturma ve ikincisi ise eleme işlemleridir. Aday oluşturmada *L(k-1)* kendisi ile birleştirilerek aday nesneler oluşturulur. Eleme işleminde ise Apriori özelliğine göre yoğun olmayan aday nesne kümelerinin elenmesi işlemi gerçekleştirilmektedir. Yoğun olmayan nesne kümelerinin test edilmesi işlemi de **has_infrequent_subset** alt programı ile gerçekleştirilmektedir.

3.2.3. Yoğun Nesne Kümelerinden Birliktelik Kurallarının Çıkarılması

Veri tabanı üzerinden yoğun nesne kümeleri bulunduktan sonra bu nesne kümeleri kullanılarak çok güçlü kurallar üretilebilmektedir. Bu kuralar minimum destek ve güven değerlerinin her ikisini de sağlamaktadır. Bu kurallar aşağıdaki güven denklemleri kullanılarak elde edilebilir [49].

$$\text{Güven } (A \Rightarrow B) : P(B \backslash A) = \frac{\textit{Destek değeri (A U B)}}{\textit{Destek değeri (A)}} \qquad (3.3)$$

(A U B) Destek değeri, hem *A* hem de *B*'nin birlikte bulunduğu nesne sayısıdır. *(A) Destek Değeri,* sadece *A*'yı içeren nesne sayısıdır. Bu denklemden, aşağıdaki gibi birliktelik kuralları elde edilebilir.

- *l* nesnesini tara ve *l*'nin boş olmayan alt kümelerini bul.
- Her boş olmayan altküme nesnesi için elde edilen kural;

$$\text{“s} \Rightarrow (l\text{-s)” if } \frac{destek(l)}{destek(s)} \geq min\_guv \qquad (3.4)$$

Denklem 3.4'teki *min_guv* değeri, minimum güven değeridir. Kurallar, yoğun nesne kümelerinden üretildiği için her bir kural otomatik olarak minimum destek değerini sağlamaktadır.

Örnek: Kuralların çıkarılmasını Şekil 3.1'de gösterilen veri tabanı (*D*) üzerinde bir örnekle göstermek mümkündür. Bu veri tabanındaki yoğun nesne kümelerinden biri *l*= {M1, M2, M5} tir. Bu *l* yoğun nesne kümesi üzerinden hangi kuralların üretilebileceği şu şekilde hesaplanır: *l*'nin tüm boş olmayan alt kümeleri {M1, M2}, {M1, M5}, {M2, M5}, {M1}, {M2} ve {M5} tir. Buna göre çıkarılabilecek birliktelik kuralı sonuçları aşağıdaki listede verilmektedir.

(1) $M1 \wedge M2 \Rightarrow M5$, güven = 2 / 4 = % 50

(2) $M1 \wedge M5 \Rightarrow M2$, güven = 2 / 2 = % 100

(3) $M2 \wedge M5 \Rightarrow M1$, güven = 2 / 2 = % 100

(4) $M1 \Rightarrow M2 \wedge M5$, güven = 2 / 6 = % 33

(5) $M2 \Rightarrow M1 \wedge M5$, güven = 2 / 7 = % 29

(6) $M5 \Rightarrow M1 \wedge M2$, güven = 2 / 2 = % 100

Yukarıdaki listeye göre eğer minimum güven değeri %70 ise sadece 2. 3. ve 6. kurallar istenilen sonuçlardır.

3.3. Diğer Birliktelik Kuralı Algoritmaları

Apriori algoritması dışında birliktelik kuralı üreten diğer bazı algoritmalar Tablo 3.2'de karşılaştırmalı olarak verilmektedir.

Tablo 3.2. Birliktelik kuralı algoritmaları ve özellikleri

Algoritma	**Veri Tabanını Tarama Sayısı**	**Algoritmanın Özelliği**
AIS	N	Adaylar, veri tabanı taranarak elde edilir. Küçük ölçekli veritabanlarında uygundur.
Apriori	N	Adaylar, *L(k-1)* ile *L(k-1)'*in birleştirilmesiyle elde edilir ve veri tabanı taranarak sayılır. Orta ölçekli veritabanlarında uygundur. AIS in dezavantajlarından arındırılmıştır.
Apriori_TID	N	Adaylar, *L(k-1)* ile *L(k-1)'*in birleştirilmesinden elde edilir. $Ck_{çok}$ ise yavaş aksi halde Apriori'den daha performanslıdır.
Apriori_Hybrid	N	Apriori ve Apriori_TID'e göre daha iyi çalışır. Her iki algoritmayı da kullanan melez bir yapıdır.
OCD	N	Adaylar, veri tabanı taranarak elde edilir. Büyük veritabanlarında ve küçük destek değerlerinde uygulanabilir.
SETM	N	SQL komutları uyumluluğu mevcuttur.
DHP	N	Adaylar, *L(k-1)* ile *L(k-1)'*in birleştirilmesi ile elde edilir ve veri tabanı taranarak sayılır. Hash tablosu kullanır.
Partition	2	Geniş veritabanlarında uygulanabilir. Homojen veriler üzerinde etkilidir.
MONET	N	Adaylar, *L(k-1*) ile *L(k-1)'*in birleştirilmesiyle elde edilir ve kolonların kesiştirilmesi ile sayılır.
Sampling	≤ 2	Adaylar, *L(k-1)* ile *L(k-1)'*in birleştirilmesi ile elde edilir ve veri tabanı taranarak sayılır. Geniş veritabanlarında ve küçük destek değerlerinde uygulanabilir.
DIC	$\leq$ N (genelde 2)	Adaylar, veri tabanı taranarak sayılır. Yoğun olabilecek tüm nesne kümeleri kontrol edilir.
MaxClique	1	Olası yoğun nesne kümeleri incelenir. *tidlist* liste yapısı kullanır ve kesiştirme işlemi ile adaylar sayılır.
Max-Miner	N	Adaylar, *L(k-1)* ile *L(k-1)'*in birleştirilmesiyle elde edilir ve veri tabanı taranarak sayılır.
Carma	2	Veri tabanı verileri ağ üzerinden elde edildiğinde uygulanabilir. Destek ve güven değerleri çevrimiçi olarak değiştirilebilmektedir.

3.4. Birliktelik Kuralı Türleri

Bilindiği gibi birliktelik kuralları algoritmaları, veri tabanını tarayarak nitelikler arasındaki ilişkileri ortaya çıkarmakta ve bir niteliğin varlığına veya yokluğuna bağlı olarak ilişkiler ve kurallar üretmektedir. Diğer özellikler olan miktar, ağırlık ve hiyerarşik bilgi düzeni gibi özellikler göz önüne alınmamaktadır. Bu bölümde birliktelik kurallarının nesne kümesi üretimine bağlı olarak çeşitleri hakkında kısaca bilgi verilecektir.

3.4.1. Hiyerarşik Birliktelik Kuralları

Birçok durumda veri tabanı nitelikleri belirli bir hiyerarşi yapısında olabilmektedir. Örneğin bir market veri tabanı için, içecekler→ alkolsüz içecekler→ meyve suları→ portakal suyu gibi bir hiyerarşi söz konusu olabilir. Bu tür durumlarda Genelleştirilmiş (çok seviyeli) birliktelik kuralları [50], aynı seviyede olan nitelikler arasındaki kurallarda olduğu gibi hiyerarşinin farklı seviyelerindeki birliktelik kurallarını bulmayı amaçlamaktadır. Genelleştirilmiş bir birliktelik kuralının örneği "*alkolsüz içecekler → meyve*" şeklindedir. Orijinal veri tabanında olmayan hiyerarşi seviyeleri için yeni bir kolon üretilmesi bu durumlar için yetersiz bir çözüm olabilmektedir. [50-51]'de Hiyerarşik bilgiyi algoritmayla birleştiren yeterli ve etkili algoritmalar önerilmektedir. Nesne merkezli bir metot [52]'de ve [53]'da çok seviyeli birliktelik kurallarını bulmak için SQL sorguları kullanılmıştır. Son olarak [54]'de esnek çok seviyeli birliktelik kuralları tartışılmaktadır.

3.4.2. Sınırlandırılmış Birliktelik Kuralları

Gerçek hayatta, kullanıcılar genellikle veri tabanından çıkarılmış birliktelik kurallarının küçük bir kısmı ile ilgilenmektedirler. Örneğin bir kullanıcı sadece bazı nitelikler arasındaki birlikteliklerin varlığı veya yokluğu üzerine yoğunlaşmış olabilmektedir. Bu konuda birçok algoritma olmakla beraber [56]'de sınırlandırılmış birliktelik kuralı için bir algoritma önerilmektedir. Algoritmada kullanılan destek ve güven değerleri de elde edilen birliktelik kurallarını sınırlandırmaktadır. Ayrıca, son zamanlarda elde edilen kurallar için ilginçlik ölçütü de kural sınırlandırmada kullanılmaktadır [55, 57, 59]. Bir niteliğin değeri beklenenden oldukça yüksek bir değere sahip olduğunda bu nitelik ile elde edilen kurallardan birçoğu önemsiz olabilmektedir. Bir diğer ölçüt [60]'da önerilen

ve ortalama karesel hataya dayalı metottur. [61]'de de ki-kare korelasyon testi birliktelik kurallarında ölçüt olarak kullanılmıştır.

3.4.3. Sıralı Örüntüler

Birliktelik kurallarının amacı, belli bir zamanda meydana gelen oluşumları keşfetmektir. Ancak, verilerin uzun bir zaman süreci ile depolanması ile beraber sıralı örüntülerin keşfi önemli bir konu haline gelmiştir. Sıralı örüntüler için bir müşterinin "Yıldız Savaşları" filmini kiralaması, daha sonra "İmparator Geri Dönüyor" filmi ve daha sonra da "Jedi'nin Dönüşü" adlı filmleri kiralaması örnek olarak verilebilir [62]. Sıralı örüntülerde nesnelerin sıralı olması gerekmez ancak oluşumları sıralı olmalıdır. Kaynak [62]'de sıralı örüntü keşfi için üç ayrı algoritma önerilmektedir. Sıralı örüntü keşfi, kaynak [63]'te ise hiyerarşik bilgi ve kayan pencereler kullanılarak geliştirilmiştir. Ayrıca kaynak [64 - 66]'da ise sıralı örüntü keşfi için etkili algoritmalar önerilmektedir.

3.4.4. Periyodik Kurallar

Birliktelik kuralları, sıralı yapıda olan bir veri üzerindeki periyodik özellikleri ortaya çıkarabilmektedir. Elde edilen bazı kurallar, belirli bir zaman periyodu için yeterli destek değerine sahip olabilmektedir. Ancak bu kurallar tüm veri tabanı ele alındığında bazen minimum destek değerini sağlamayabilir. Özden ve arkadaşları tarafından [49] belirli bir zaman periyodunda minimum destek ve güven değerini sağlayan kurallar için Periyodik Birliktelik Kuralları kavramı öne sürülmüştür. Örneğin "insanlar her Pazar süt ile beraber gazete de alırlar" gibi bir kural her zaman için geçerli olmayıp sadece periyodik olarak tekrar eden bir kuraldır. Kaynak [49]'da her *t* zaman süresince tekrarlayan durumlar için iki algoritma önerilmektedir. Kaynak [69]'da ise takvimsel birliktelik kurallarını tespit edilmeye çalışılmıştır. Burada kullanıcının tarafından belirli bir takvim dönem bilgisi izlenmektedir. Bu çalışmalar özellikle tam periyodik bir yapıya sahiplerdir. Elde edilen kurallar, periyodik bir yapıdaki her zaman noktası için geçerli olmak zorundadır. Han ve arkadaşları [71] ise bu kısıtlamayı ortadan kaldırmak için kısmen periyodik örüntüler üzerine çalışmışlardır.

3.4.5. Ağırlıklandırılmış Birliktelik Kuralları

Birliktelik Kuralı algoritmalarında genellikle veri tabanı üzerindeki tüm nitelikler aynı önemi taşımaktadır. Cai ve arkadaşları [66], çalışmalarında her bir niteliğin kendi önemini yansıtmasını amaçlamıştır. Bu ağırlıklar, bazı ürünler üzerindeki promosyonlar veya ürünlerin fiyatları olabilmektedir. Bu durumda bazı nitelikler için ağırlıklandırılmış destek değeri kullanılması gerekir. Daha önceki birliktelik kuralı algoritmalarına sadece hesaplama yaparak ağırlıklı destek değerinin uygulanabilmesi mümkün olmamaktadır. Bu nedenle[66] yeni bir algoritma önerilmiştir.

3.4.6. Negatif Birliktelik Kuralları

Savasere ve arkadaşları [69] tarafından yapılan çalışmada, nitelikler arasındaki olumlu birliktelik kuralları yerine olumsuz birliktelik kuralları araştırılmıştır. $A \rightarrow B$ şeklinde gösterilen birliktelik kuralı $A \rightarrow \neg B$ şeklinde negatif birliktelik kuralı olarak gösterilmektedir. Yani A ürününü alan müşterilerin B ürününü almaması şeklindedir. Örneğin, "Asitli içecek alan insanların çoğu meyve almaz" şeklinde bir kural negatif birlikteli kuralı olarak ele alınmaktadır. Negatif birlikteliklerin bulunması için minimum destek ve güven değerlerinin mümkün olduğunca azaltılması gerekir. Bu durumda $A \rightarrow B$ birliktelik kuralına ait destek ve güven değerleri kullanılarak $A \rightarrow \neg B$ şeklindeki negatif birliktelik kuralına ait destek ve güven değerini hesaplamak mümkündür [67]. Fakat bunun sonucunda çok miktarda ilginç olmayan kurallar da üretilmektedir. Kaynak [68]'de bu durum için etkili ve yeterli bir algoritma bulunmaktadır.

3.4.7. Nicel Birliktelik Kuralları

Orijinal birliktelik kuralı problemi, niteliklerin var/yok (*boolean*) değeri ile ilgilenmektedir. Ancak niteliklerin aldığı değerler birçok problemde ikiden fazla olabilmektedir. Bu durumlarda, niteliklerin sayısal veya kategorik değerleri ile kural üretmek için [28]'de etkili bir algoritma önerilmektedir. Örneğin, Yaş:30..39 ve Evli: Evet → Arabaların Sayısı: 2 (%40, 90%) şeklinde verilen bir kural nitel birliktelik kuralı olarak değerlendirilmektedir. Bu kural, yaşı 30 ile 39 arasında olan ve evli olan insanların %90'ının iki araba sahibi olduğu anlamına gelmekte ve veri tabanı üzerindeki destek değeri %40'tır.

Nicel birliktelik kurallarının hesaplanmasındaki önemli nokta, nitel verilerin değerlerinin nasıl bir şekilde bölümlere ayrılacağıdır. Fukuda ve arkadaşları [72] optimize edilmiş birliktelik kurallarını geliştirerek destek veya güven değerini maksimuma çıkaracak şekilde nümerik verilen değerlerin bölümlerini bulmaya çalışmışlardır. Nümerik veya kategorik veriler için aynı kavram [30]'da da araştırılmıştır. Ayrıca Wang ve arkadaşları [73] bir kuralın ilginçliğini maksimuma çıkarmak için farklı aralıkları kullanarak ilginçliğe dayalı bir yöntem önermiştir. Bir başka çalışmada, nümerik değerler için bulanık birliktelik kuralları önerilmektedir [73]. Bulanık birliktelik kuralları, $X = A \rightarrow Y = B$ formatında olup buradaki X ve Y nesne kümelerini, A ve B ise sırasıyla X ve Y'yi tanımlayan bulanık değerleri göstermektedir.

Tablo 3.3. Nicel değerler içeren örnek veri seti

ID	Cinsiyet	Gelir	Yaş
1	Bayan	Düşük	yaş <18
2	Erkek	Orta	18<yaş<25
3	Erkek	Orta	25<yaş<35
4	Erkek	Düşük	Yaş>35
5	Erkek	Yüksek	Yaş>35

4. İLGİNÇLİK ÖLÇÜTÜ

Diğer verilerden dikkat çekici şekilde ayrılan ve göreceli olarak az sayıda bulunan örüntüler olağandışı veri olarak nitelenmekte ve bu tip veriler ilginçlik ölçümü için en çarpıcı adaylar arasında yer almaktadır. Bu verilere ilişkin kuralların ve bu verileri oluşturan düzeneklerin keşfi, daha gelişmiş karar destek sistemlerinin oluşması için anlamlı bir katkı olacaktır.

4.1. Nicel Birliktelik Kurallarında İlginçlik Ölçütü

En ilginç veri ve örüntülerin bulunabilmesi için olağandışılık kavramını temel alan pek çok ilginçlik ölçütü ortaya atılmış, ancak bu ölçütler gereksinimin ancak kısıtlı bir bölümünü karşılamıştır. Ortaya çıkan şablonun sürükleyicilik ölçümü, veri madenciliği araştırmasında etken ve önemli bir alandır. Bu alanda çok çalışma yapılmasına rağmen şimdiye kadar konuyla ilgili sürükleyiciliğin resmi bir tanımlamasında genel bir kanı yoktur. Şimdiye kadar yapılan tanımlamaların çeşitliliğine dayanarak ilginçlik ölçütü; özlülük, kapsama, gerçeklik, özellik, çeşitlilik, yenilik, şaşırtıcılık, yararlılık ve hareketliliği vurgulayan geniş bir kavram olarak alınabilir. Bu spesifik dokuz kriter örneklerin farklı olup olmadığını belirlemek için kullanılır [75]. Bunların tanımlamaları aşağıdadır:

Özlülük, Kısa bir çift ya da çift serisini anlamak uzun çift serilerine göre daha kolaydır. Bu nedenle minimum çiftlerin güvenirliliğini ölçmek için birçok çalışma yapılmıştır.

Yaygınlık/ Kapsamlılık, Eğer bir örüntü, örnek bilgi serisinin geniş alt kümesini içeriyorsa yaygındır. Yaygınlık (ya da kapsamlılık) modelin kapsamlılığını ölçer ki o modelle örtüşen veri setindeki bütün kayıtların parçasıdır. Eğer bir örüntü veri serisinde daha fazla bilgiyi simgelerse, örüntü daha ilginç olmaya meyillidir. Yaygın nitelik kümelerini bulmak için kullanılan en iyi bilinen algoritma Apriori algoritmasıdır. Bazı yaygın ölçümler kısaltma stratejilerinin temelini oluşturabilirler, mesela destek ölçütü Apriori algoritmada nitelik kümesinin özeti gibi kullanılır. Yaygınlık, çoğunlukla özlülüke denk gelir. Çünkü kısa örüntüler daha büyük kapsama olmaya eğilimlidir.

Güvenilirlilik, Bir örüntü tarafından tanımlanan ilişki uygulanabilir derecede yüksek yüzdeye sahipse bu örüntü güvenilirdir. Mesela bir sınıflama kuralı eğer tahmini

büyük ölçüde doğruysa ve birliktelik kuralı çok güven veriyorsa güvenilirdir. Olasılık, istatistik ve bilgiden gelen ölçümlerin çoğu birliktelik kurallarının güvenilirliğini ölçmek için kullanıldığı ileri sürülmektedir.

Özgünlük, Eğer bir örüntü, bazı uzaklık ölçümlerine göre bulunan diğer örüntülerden uzaksa özeldir. Özel örüntüler özel verilerden oluşurlar ki bunlar sayı olarak nispeten az ve geri kalan verilerden önemli ölçüde farklıdırlar. Özel örüntüler kullanıcı tarafından bilinmeyebilir ve bu yüzden farklıdırlar.

Çeşitlilik: Bir örüntü, bir örüntü gurubundaki her bir örüntüden önemli ölçüde farklıysa, örüntünün elementleri birbirinden farklıdır. Çeşitlilik özetin ilginçliğini ölçmede kullanılan yaygın bir faktördür. Eğer muhtemel dağılımı tek bir örneğinin dağılımından uzaksa farklıdır.

Yenilik: Eğer bir kişi herhangi bir örüntüyü daha önceden bilmiyorsa ve diğer bilinen örüntülerden dikkat çekici farklılıkları bulunuyorsa yeni bir örüntüdür. Veri madenciliği sisteminde kullanıcının bildiği her şeyi gösteren hiç veri yoktur ve bu nedenle yenilik kullanıcının bilgisiyle kesin olarak ölçülemez. Bunun gibi bilinen hiçbir veri madenciliği sistemi kullanıcının neyi bilmediğini göstermez ve bu yüzden yenilik kullanıcının bilgisizliğiyle de kesin olarak ölçülemez. Bunun yerine yenilik, kullanıcının örüntüyü kesinlikle yeni olarak tanımlaması ya da örüntüyü daha önceki örüntülerden çıkarım yapamaması ve onlarla çelişmesi durumuyla çıkartılır.

Şaşırtıcılık: Bir örüntü eğer kişinin önceki bilgileriyle ya da beklentileriyle çelişiyorsa şaşırtıcıdır. Bir örüntü önceden bulunan yaygın örüntülere göre daha genel olan sıra dışı bir örüntü de şaşırtıcı olarak düşünülebilir. Şaşırtıcı örüntüler farklıdır çünkü onlar daha önceki bilgilerdeki eksikliği belirler ve sonraki ihtiyaç duyulan veri çalışmalar için öneri sunabilir. Şaşırtıcılık ve yenilik arasında ki fark şaşırtıcı örüntü kullanıcının önceki bilgi ve beklentilerine ters düşerken yenilikçi bir örüntü yenidir ve önce bilinen örüntülerle çelişmez.

Yararlılık: Bir örüntü eğer bir kişi tarafından kullanımı bir amaca ulaşılmasına katkı sağlıyorsa yararlıdır. Farklı insanların veri kümesinden çıkmış bilgiyle ilişkili farklı amaçları olabilir. Mesela bir kişi toptan satışlarda büyük artışlı bütün işlemleri bulmakla ilgilenirken başka biri işlem bilgi kümesindeki yüksek karlı bütün satışları bulmayla ilgilenebilir. Bu tür farklılıklar işlenmemiş verilere ek olarak kullanıcı tanımlı yararlılık fonksiyonları temellidir.

Hareketlilik/ Uygulanabilirlik: Eğer bir örüntü, bir alanda gelecekteki çalışmalar için karar verme imkânı sağlıyorsa hareketlidir. Hareketlilik bazen örüntü seçim stratejisiyle bağdaştırılabilir. Şimdiye kadar hareketliliği ölçecek hiçbir genel metot bulunamamıştır. Var olan ölçümler uygulamalara bağlıdır. Mesela, müşterinin bugünkü durumundaki değişiklik hedefle uyuştuğu için hesaplanabilirliği ölçmüştür, oysaki hesaplanabilirliği birleşme kuralının getirebileceği kar olarak ölçmüştür.

Yukarıda bahsedilen ilginçlik ölçütleri bazen birbiriyle bağımsızlıktan ziyade birbiriyle ilişkilidir. Mesela uygulanabilirliğin ilginçlik için iyi bir tahmin olacağını ya da tam tersi olacağını savunur: Daha önce belirtildiği gibi özlülük, sıklıkla yaygınlıkla ve yaygınlık gerçekçiliğin bir şekli olan gürültüye azaltılmış duyarlılıkla uyumludur. Aynı zamanda sonradan bahsedilen yenilikle uyumluyken yaygınlık özgünlülükle çelişir.

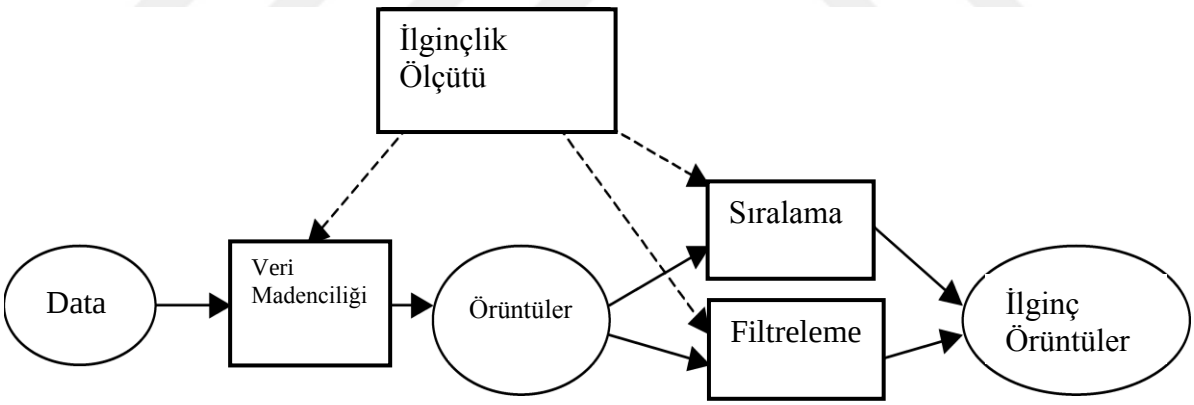
Bu dokuz kriter daha sonra üç gurupta sınıflandırılabilirler: nesnel, öznel ve anlamsal merkezli. Nesnel bir ölçme sadece işlenmemiş veri tabanıdır. Kullanıcıyla ya da uygulamayla ilgili hiçbir bilgi gerekli değildir. Nesnel ölçümlerin çoğunluğu olasılık, istatistik ya da bilgi teorilerine dayalıdır. Özlülük, yaygınlık, güvenilirlik, özgünlük ve çeşitlilik sadece veri ve örüntülere dayalıdır ve bu nedenle nesnel düşünülebilir. Öznel bir ölçüt hem veriyi hem de bu verileri kullanan kullanıcıları hesaba katar. Öznel ölçümü tanımlamak için kullanıcının alan bilgisi ya da veri ile ilgili geçmiş bilgisine ulaşmak gereklidir. Bu kullanım veri madenciliği süreci boyunca kullanıcı ile etkileştirilerek ya da kesin olarak kullanıcının bilgi ya da beklentilerini göstererek elde edilebilir. Sonraki durumda anahtar konu veri madenciliğindeki çeşitli sistem çalışmaları ve prosedürleri tarafından gönderilen kullanıcı bilgisidir. Yenilik ve şaşırtıcılık veri ve örüntünün kullanımı kadar kullanıcılara da bağlıdır ve bu nedenle öznel olarak düşünülebilir.

Anlamsal bir ölçüm örüntülerin anlamlarını ve açıklamalarını dikkate alır. Çünkü bu ölçümler kullanıcının alan bilgisini içerir, bazı araştırmacılar onları öznel ölçünün özel bir çeşidi olarak düşünür. Faydacılık ve hareketlilik verilerin anlamlarına dayalıdır ve bu nedenle anlamsal olarak düşünülebilir. Faydacılık temelli ölçümler en yaygın anlamsal ölçüm şekilleridir. Faydacılık temelli yaklaşımda kullanıcı alanla ilgili ilave bilgiyi açıkça belirtmelidir. Mesela bir mağaza müdürü yüksek karlı niteliklerle ilgili birleşme kurallarını istatistiksel olarak daha yüksek olana tercih edebilir.

Bir örüntünün farklı olup olmadığını belirlemek için dokuz kriter düşünüldüğünde ilginçlik tespiti dediğimiz bu tespiti göstermek için üç metodu düşünelim. Öncelikle biz her örüntüyü farklı ya da farksız olarak sınıflandırabiliriz. Mesela ilginçlik ve sıradanlık

arasındaki farkı tespit etmek için iki kare testi kullanırız. İkinci olarak; bir örüntünün diğerinden daha farklı olduğunu gösteren tercihi ilişkilendirmeyle belirlenebilir. Bu metot kısmı sıralamayı ortaya çıkarır. Üçüncü olarak; örüntüleri sıraya koyabiliriz. İkinci ve üçüncü yaklaşımda daha önce de bahsedilen dokuz kriter temelli bir ilginçlik ölçümünü tanımlayabiliriz ve bu ölçümü ilk yaklaşımdaki farklı ve farklı olmayan örüntüleri ayırmak için ya da üçüncü yaklaşımdaki örüntüleri sıralamak için kullanırız. Böylece farklı olmayan ölçümleri kullanma genel ve pratik yaklaşımı otomatik tanımlanan farklı örüntüler için kolaylık sağlar.

Veri madenciliği süreci boyunca ilginçlik ölçümleri ilginçlik ölçümlerinin rolü dediğimiz üç yolda kullanılabilir. Şekil 4.1. bu kuralları göstermektedir. İlk olarak ölçümler araştırma alanını daraltmak ve böylece veri madenciliğinin etkinliğini geliştirmek için veri madenciliği süreci boyunca farklı olmayan örüntüleri kısaltmak için kullanılabilir. Örneğin, minimum destek değeri veri madenciliği işlemi boyunca düşük destekli örüntüleri ayırmak için kullanılabilir ve böylece etkinliliği geliştirir. Benzer olarak bazı faydacılık temelli ölçümler için yararlılık eşiği tanımlaması yapılabilir ve düşük faydalı değerleri olan örüntüleri özetlemede kullanılır.



Şekil 4.1. Veri madenciliği sürecinde ilginçlik ölçümlerinin rolü

İkinci olarak; ölçümler ilginçlik skorlarına göre sınıflandırılabilir. Üçüncü olarak; sonraki işlemler boyunca farklı örüntüleri seçmek için kullanılabilir. Mesela veri madenciliği işleminden sonra önemli ilişkileri olan kuralları seçmek için iki kare testi kullanabiliriz. Araştırmacılar farklı türdeki örüntülerin ilginçlik ölçümlerini önerdiler, teorik özellikleri incelediler, deneysel olarak değerlendirdiler ve belirli alanlar ve ihtiyaçlar

için uygun ölçümü seçme stratejileri tavsiye ettiler. İlginçlik ölçümlerince değerlendirilen en yaygın örüntüler birliktelik kuralları, sınıflandırma kuralları ve bazı kısaltmaları içerir.

Örneğin bir veri tabanındaki kişilerin tamamı erkek ve bu kişilerin yarısı derslerinde başarısız olması durumunda, güven değeri %100 olan *Başarısız→Erkek* şeklinde bir kural çıkmaktadır. Bu kurala göre başarısız olanların tamamı erkektir ancak bu kural ilginç bir kural değildir. Literatürde bu gibi durumlarda oluşturulan kuralların ilginçliğini ölçmek için çeşitli çalışmalar yapılmıştır. Bu çalışmalardan biri de ilginçlik değeri olan "Lift" değeridir.

Tablo 4.1. İlginçlik ölçütü örnek veri seti

	Basketbol Oynayanlar	**Basketbol Oynamayanlar**	**Toplam**	
Cips Yiyenler	2000	1750	3750	%75
Cips Yemeyenler	1000	250	1250	%25
Toplam	3000	2000	5000	
	%60	%40		

Tablo 4.1'e göre basketbol oynayanların % 66.7'sinin cips yediği görülmektedir. Ancak tabloya göre öğrencilerin zaten %75'i cips yediği için bu kuralın ilginç olduğu söylenemez. Diğer taraftan basketbol oynayıp cips yemeyenlerin güven değeri % 33.6 olmasına rağmen tüm öğrenciler içinde %25 oranında cips yemeyen olduğu düşünüldüğünde bu kural daha ilginç bir kural olabilmektedir. Lift değeri, oluşturulan kurallara bir ilginçlik katsayısı vermektedir. Bu katsayı eğer 1'in üzerinde ise kural pozitif anlamda ilginçtir, 1'in altında ise nesneler arasında negatif anlamda bir ilginçlik vardır denir. Buna göre ilginçlik katsayı olan Lift değeri Denklem 4.1'e göre hesaplanmaktadır.

$$Lift(A \rightarrow B) = \frac{sup(A \wedge B)}{sup(A).sup(B)} = \frac{P(B \backslash A)}{P(B)} \qquad (4.1)$$

Yukarıdaki örneğe göre ilginçlik katsayıları hesaplandığında;

Basketbol Oynayıp→Cips Yiyenler $= \frac{\frac{2000}{5000}}{\frac{3000}{5000} * \frac{3750}{5000}} = 0.89$

Basketbol Oynayıp→Cips Yemeyenler=$\frac{\frac{1000}{5000}}{\frac{3000}{5000}*\frac{1250}{5000}} = 1.33$ olarak elde edilir.

5. NİCEL BİRLİKTELİK KURALLARI İÇİN ÇOKLU DESTEK DEĞERİ

Geleneksel birliktelik kuralı algoritmaları, niteliklerin varlığı veya yokluğu üzerine kurallar üretmektedir. Ancak veri tabanlarında niteliklerin aldığı değerler sadece varlık veya yokluk değeri almayabilirler. Bir niteliğin birden fazla değer aldığı durumlar olabilir. Bu durumlarda kural üretmek için niteliklerin sayısal veya kategorik değerlerini işleyen algoritmalar kullanılmalıdır. Örneğin veri setinde medeni hal, öğrenim durumu, cinsiyet, doğum yeri gibi değerler olabilir. Her biri farklı ölçekte nitelik değerlerine sahip olan bu verilerden birliktelik kurallarının doğru bir şekilde elde edilebilmesi gerekmektedir. Bu nedenle Bölüm 3.2'de önerilen birliktelik kuralı algoritması bu problem için yeterli olmamaktadır.

5.1. Problemin Tanımı

Birliktelik kuralı, minimum destek değerini parametre olarak kullanmakta ve veri setini tarayarak bu değeri sağlamayan nitelikleri her adımda elemektedir. Veri setine ait tüm niteliklerin aldığı değerler eşit aralıkta olduğunda sabit bir minimum destek değeri kullanılması sorun teşkil etmemektedir. Ancak nicel veya nitel değerler alan ve niteliklerin aldığı değerlerin farklılıklar oluşturduğu veri setlerinde sabit minimum destek değeri uygun sonuçlar üretmemektedir. Veri setlerindeki farklı niteliklerin aynı öneme, sıklığa veya yapıya sahip olmamaları veya bazı niteliklerin sık olmasa da daha büyük öneme sahip olmaları çeşitli sorunlar ortaya çıkarabilmektedir [19].

- Minimum destek değeri yüksek olduğunda, bazı ilginç ve değerli kurallar sık tekrarlanmadığı için elde edilememekte,

- Minimum destek değeri düşük olduğunda ise çok sayıda kural elde edilmekle birlikte ilginç ve değerli birliktelik kurallarının bulunması da oldukça zorlaşmaktadır.

Örneğin bir veri tabanında cinsiyet niteliği sadece iki değer (bay, bayan) alabilirken, öğrenim durumu niteliği yedi farklı değer (ilk, orta, lise, ön lisans, lisans, yüksek lisans, doktora) alabilmektedir. Böyle durumlara sahip bir veri seti üzerinde kullanılacak olan sabit bir minimum destek değeri, elde edilen kuralların ilginçliğini ve değerini düşürecektir. Destek değerinin yüksek seçilmesi, öğrenim durumu ile ilgili hiçbir kuralın üretilmemesine, destek değerinin düşük verilmesi ise cinsiyet niteliğinin elde edilen tüm kurallar içerisinde yer almasına, anlamlı veya anlamsız birçok kural oluşmasına sebep olacaktır.

5.2. Yöntem

Yukarıda verilen problemin çözümü için, veri setindeki her nitelik için ayrı ayrı minimum destek değeri kullanan yeni bir yaklaşım önerilmiştir. Tablo 2'de, bir okuldaki öğrencilerle ait bir veri seti örneği görülmektedir. Bu tabloya göre birliktelik kuralları oluşturulmaya çalışıldığında, minimum destek değerinin seçimi nitelikler arasındaki ölçek farkından dolayı sorun oluşturabilmektedir. Seçilecek minimum destek değerinin yüksek olması durumunda, "gelir durumu" ile ilgili hiçbir kuralın elde edilememesi sorunu ortaya çıkabilmektedir. Minimum destek değerinin düşük olduğu durumlarda ise "cinsiyet" ve "internet kullanımı" gibi sadece iki farklı değer alan niteliklerin her kuralda yer alması ve ilginç olmayan değersiz kuralların elde edilmesi durumu ortaya çıkmaktadır. Çünkü veri setinde "internet", "cinsiyet" ve "başarı durumu" nitelikleri sadece iki farklı değer alırken "yaşam" niteliği dört, "gelir durumu" niteliği ise üç farklı değer almaktadır. Nitelikler arasındaki bu ölçek farklılığı, nicel veya nitel birliktelik kuralları için problem oluşturmaktadır.

Tablo 5.1: Öğrenci yaşam ve başarı durumu

ID	Cinsiyet	Gelir	İnternet	Yaşam	Başarı
1	Bayan	Düşük	Yok	Yurt	Başarısız
2	Erkek	Orta	Var	Akraba	Başarılı
3	Erkek	Orta	Var	Arkadaş	Başarısız
4	Erkek	Düşük	Var	Yurt	Başarısız
5	Erkek	Yüksek	Yok	Arkadaş	Başarılı
6	Bayan	Yüksek	Yok	Akraba	Başarısız
7	Bayan	Orta	Yok	Aile	Başarılı
8	Bayan	Düşük	Var	Yurt	Başarısız
9	Erkek	Düşük	Var	Yurt	Başarılı
10	Bayan	Yüksek	Var	Arkadaş	Başarısız

Bu problemin çözümü için, algoritmaya girilen minimum destek değeri parametresini her nitelik üzerinde farklı olacak şekilde kullanan çoklu destek değeri

kullanılması önerilmektedir. I = {I1, I2, … In} veri setine ait nitelikler olmak üzere her bir niteliğe ait minimum destek sayısı Denklem 5.1'deki formül ile hesaplanması önerilmektedir.

$$Minsup(i) = \frac{|T|.s.2}{100.\varphi(i)} \qquad (5.1)$$

formülde,

Minsup(i) = *i* niteliği için minimum destek sayısı,
|T| = Kayıt sayısı,
s = % olarak destek değeri,
φ(i) = *i* niteliğin aldığı farklı değer sayısı olarak verilmektedir.

Örneğin Tablo 5.1'de verilen veri seti için "yaşam" niteliğine ait destek sayısı;

$$Minsup(yaşam) = \frac{10.40.2}{100.4} = 2$$

olarak hesaplanmaktadır.

Tablo 5.2'de her bir niteliğin aldığı değerler ve her bir nitelik için kullanılacak minimum destek sayısı verilmektedir. Ayrıca veri setine dönüşüm uygulanarak, Apriori algoritması uygulanabilecek mantıksal formata dönüştürülmüştür. Bu şekilde her bir niteliğin aldığı farklı değerler o niteliğin bir alt niteliğini oluşturmakta ve alt niteliklere de o niteliğe ait minimum destek değeri uygulanmaktadır.

Tablo 5.2. Nitelikler ve %40 destek değeri için her bir niteliğin destek sayıları (d).

ID	Cinsiyet d=4		Gelir d=2			İnternet d=4		Yaşam d=2				Başarı d=4	
	Bayan	Erkek	Düşük	Orta	Yüksek	Yok	Var	Aile	Arkadaş	Akraba	Yurt	Başarısız	Başarılı
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	0	1	0	0	1	0	0	0	0	1	1	0
2	0	1	0	1	0	0	1	0	0	1	0	0	1
3	0	1	0	1	0	0	1	0	1	0	0	1	0
4	0	1	1	0	0	0	1	0	0	0	1	1	0
5	0	1	0	0	1	1	0	0	1	0	0	0	1
6	1	0	0	0	1	1	0	0	0	1	0	1	0
7	1	0	0	1	0	1	0	1	0	0	0	0	1
8	1	0	1	0	0	0	1	0	0	0	1	1	0
9	0	1	1	0	0	0	1	0	0	0	1	0	1
10	1	0	0	0	1	0	1	0	1	0	0	1	0

5.3. Kuralların Oluşturulması

Tablo 5.2'de elde edilen veri setine Apriori algoritması, her nitelik değeri için ayrı destek değeri uygulanacak şekilde yürütülmektedir. Tablo 5.3'te her bir alt niteliğe ait destek değerleri hesaplanmış ve minimum destek değeri ile karşılaştırılmıştır. Minimum destek değerinin altında destek değere sahip alt nitelikler aday kümeden çıkarılarak apriori algoritması sürdürülmektedir. Burada H alt niteliği aday nesne kümesinden elenmektedir.

Tablo 5.3. C1-Bir Elemanlı Aday Nesne kümesi

Nitelik	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Destek Sayısı	5	5	4	3	3	4	6	1	3	2	4	6	4
Minimum Destek	4	4	2	2	2	4	4	2	2	2	2	4	4

Apriori algoritmasının 2. Adımında H alt niteliği elendikten sonra kalan alt niteliklerin tüm ikili birliktelikleri üretilmektedir. Daha sonra, oluşturulan bu ikili birlikteliklerin destek değerleri hesaplanmaktadır. Bu destek değerleri daha sonra minimum destek değerleri karşılaştırıldığında, minimum destek değerini sağlamayan ikili birliktelikler aday kümeden çıkarılır. İkili ve daha sonra elde edilecek birliktelikler minimum destek değeri ile karşılaştırılırken alt niteliklere ait birden fazla minimum destek değeri elde edildiği için bu destek değerlerinden küçük olan değer minimum destek değeri olarak kabul edilmektedir. Örneğin ikili birliktelikte yer alan (A-E) ikili birlikteliği için, "A" alt niteliğe ait minimum destek değeri 4 ve "E" alt niteliğine ait minimum destek değeri 2 olduğu görülmektedir. Bu iki niteliğin birlikteliğinde minimum destek değeri olarak "E" niteliğin destek değeri daha küçük olduğu için 2 kabul edilerek algoritmaya yürütülmektedir. Minimum destek değerinin altında kalan alt nitelikler elenerek bu şekilde algoritma sürdürülmektedir.

Tablo 5.4. C2-İki elemanlı Aday Nesne Kümesi

Nitelik	**Destek Sayısı**	**Min. Destek**
A-B	*0*	*4*
A-C	**2**	**2**
A-D	1	2

Nitelik	**Destek Sayısı**	**Min. Destek**
C-E	0	2
C-F	1	2
C-G	**3**	**2**

Nitelik	**Destek Sayısı**	**Min. Destek**
E-M	1	2
F-H	0	4
F-I	1	2

A-E	**2**	**2**
A-F	3	4
A-G	2	4
A-I	1	2
A-J	1	2
A-K	**2**	**2**
A-L	**4**	**4**
A-M	1	4
B-C	**2**	**2**
B-D	**2**	**2**
B-E	1	2
B-E	1	4
B-G	**4**	**4**
B-I	**2**	**2**
B-J	1	2
B-K	**2**	**2**
B-L	2	4
B-M	3	4
C-D	0	2

C-I	0	2
C-J	0	2
C-K	**4**	**2**
C-L	**3**	**2**
C-M	1	2
D-E	0	2
D-F	1	2
D-G	**2**	**2**
D-I	1	2
D-J	1	2
D-K	0	2
D-L	1	2
D-M	**2**	**2**
E-F	**2**	**2**
E-G	1	2
E-I	**2**	**2**
E-J	1	2
E-K	0	2
E-L	**2**	**2**

F-J	1	2
F-K	1	2
F-L	2	4
F-M	2	4
G-I	**2**	**2**
G-J	1	2
G-K	3	2
G-L	**4**	**4**
G-M	2	4
I-J	0	2
I-K	0	2
I-L	**2**	**2**
I-M	1	2
J-K	0	2
J-L	1	2
J-M	1	2
K-L	**3**	**2**
K-M	1	2
M-L	0	4

Apriori algoritması bu şekilde yürütüldüğünde Tablo 5.5'te görülen 4 elemanlı yoğun nesne kümeleri birliktelik kuralı üretmek için elde edilen son yoğun küme olarak elde edilmektedir. Tabloda görüldüğü üzere 4 elemanlı 3 farlı yoğun nesne kümesi elde edilebilmiştir. Bu durum daha fazla sayıda güven değeri yüksek ve lift değeri uygun kurallar üretilebilmesine olanak sağlamaktadır.

Tablo 5.5. L4 Dört Elemanlı Yoğun Nesne Kümesi

Nitelik	**Destek Sayısı**	**Min. Destek**
A-C-K-L	2	2
B-C-G-K	2	2
C-G-K-L	2	2

Tablo 5.6'da tekli destek değeri ile çoklu destek değeri kullanımı sonucunda elde edilen yoğun nesne kümeleri karşılaştırmalı olarak verilmektedir. Algoritma %40 minimum destek değeri ile işletildiğinde, önerilen yöntem Denklem 5.1'e göre her bir niteliğe ait destek sayısını ayrı ayrı hesaplamakta ve algoritma yürütülmektedir. Tablo 5.6'da harf kodları yerine veri setine ait gerçek değerler verilmektedir. Tablodan görüldüğü üzere destek değeri %40 olmasına rağmen destek sayısı 2 olan bazı nitelik değerleri de yoğun nesne kümelerine dahil edilmiştir.

Tablo 5.6. Tekli ve Çok Minimum Destek Değerli Yoğun Nesne Kümelerinin Karşılaştırılması

Tekli Destek (%40)	**Çoklu Destek (%40)**

Bayan-Başarısız	: Destek=4	Bayan-Gelir Düşük-Yurt-Başarısız	: Destek=2
Erkek-İnt. Var	: Destek=4	Erkek-Gelir Düşük-İnt. Var-Yurt	: Destek=2
Gelir Düşük-Yurt	: Destek=4	Gelir Düşük-İnt. Var-Yurt-Başarısız	: Destek=2
İnt. Var-Başarısız	: Destek=4		

Tablo 5.6'dan görüldüğü üzere çoklu destek değeri ile önerilen algoritma, verilen veri seti üzerinde 4 elemanlı yoğun nesne kümeleri üretebilirken standart birliktelik kuralı algoritması en fazla 2 elemanlı yoğun nesne kümeleri üretebilmiştir. Bu durum, nicel birliktelik kurallarında çoklu destek değeri kullanılarak ilginç ve değerli kuralların üretilebilmesinin mümkün olduğunu göstermektedir. Tablo 5.6'da verilen yoğun nesne kümeleri için %80 güven değeri ile kurallar elde edildiğinde her iki yöntem için de elde edilen kurallar Tablo 5.7'de görülmektedir. Çoklu destek değeri ile elde edilen kural sayısı fazla olup bu kurallardan sadece 4 tanesi Tablo 5.7'de verilmektedir.

Tablo 5.7. Tekli ve Çok Minimum Destek Değerli ile Elde Edilen Kurallar

Tekli Destek (%40) **Tüm Kurallar**		**Çoklu Destek (%40)** **Örnek Kurallar**	
Bayan→Başarısız	:Güven= %80	Düşük-Başarısız→ Yurt	: Güven=%100 lift= 2.50
Erkek→İnt. Var	: Güven= %80	Bayan-Düşük-Yurt → Başarısız	: Güven=%100 lift= 2.00
Gelir Düşük→Yurt	: Güven=%100	Erkek-Düşük-Yurt→ Int. Var	: Güven=%100 lift= 1.67
Yurt→Gelir Düşük	: Güven=%100	Bayan-Yurt→Düşük-Başarısız	: Güven=%100 lift= 3.33

Tablo 5.7 incelendiğinde, tekli destek değeri ile minimum güven değerini sağlayan sadece 4 kuralın elde edildiği görülmektedir. Oysa çoklu destek değeri kullanıldığında hem güven değeri hem de ilginçliği yüksek çok güçlü kuralların elde edildiği görülmektedir. Bu kuralların klasik birliktelik kurallarından elde edilemediği tabloda açıkça görülmektedir. Bu kuralların elde edilebilmesi, klasik birliktelik kuralında destek değerinin %40 yerine %20 seçilmesi ile ancak mümkün olabilecektir. Bu durumda, destek değeri düşük olduğu için çok sayıda kural üretilecek ve bu kuralların ilginçlik katsayıları yetersiz olacaktır.

6. UYGULAMA

Tezin bu bölümünde, önerilen çoklu destek değeri yaklaşımı ile ilgili bir uygulama gerçekleştirilmiştir. Uygulama olarak literatürde yaygın olarak kullanılan "Oyun" veri seti örnek olarak alınmıştır. Bu veri seti Tablo 6.1'de görülmektedir. Bu veri seti üzerinden hem tekli destek değeri kullanan geleneksel birliktelik kuralı hem de bu tezde önerilen çoklu destek değeri kullanan birliktelik kuralı uygulanarak elde edilen kurallar karşılaştırmalı olarak verilmektedir.

Tablo 6.1. Oyun Veri Seti

HAVA	ISI	NEM	RÜZGÂR	OYUN
Güneşli	Sıcak	Yüksek	Hafif	Hayır
Güneşli	Sıcak	Yüksek	Kuvvetli	Hayır
Bulutlu	Sıcak	Yüksek	Hafif	Evet
Yağmurlu	Ilık	Yüksek	Hafif	Evet
Yağmurlu	Soğuk	Normal	Hafif	Evet
Yağmurlu	Soğuk	Normal	Kuvvetli	Hayır
Bulutlu	Soğuk	Normal	Kuvvetli	Evet
Güneşli	Ilık	Yüksek	Hafif	Hayır
Güneşli	Soğuk	Normal	Hafif	Evet
Yağmurlu	Ilık	Normal	Hafif	Evet
Güneşli	Ilık	Normal	Kuvvetli	Evet
Bulutlu	Ilık	Yüksek	Kuvvetli	Evet
Bulutlu	Sıcak	Normal	Hafif	Evet
Yağmurlu	Ilık	Yüksek	Kuvvetli	Hayır

6.1. Tekli Destek Değeri ile Uygulama

Bu aşamada Tablo 6.1'de verilen veri setine tekli destek değeri kullanan birliktelik kuralı algoritması uygulanmıştır. Birliktelik kuralı uygularken kullanılacak minimum destek ve güven değerleri için sırası ile %40 ve %75 değerleri seçilmiştir. Tablo 6.2'de her bir nitelik ve aldığı değerlere ait destek sayıları verilmektedir. %40 minimum destek değeri 14 kayıt için uygulandığında 14*0.4= 5.6 olduğundan minimum destek sayısı 6 olarak hesaplanmaktadır. Bu nedenle algoritmanın 1. adımında tüm nitelik değerlerinin tekrar sayıları veri tabanı taranarak hesaplanmakta ve destek sayısı 6'nın altında kalan nitelikler elenmektedir. Tablo 6.2'de bir elemanlı aday nesne kümeleri ile destek sayıları, Tablo 6.3'te ise bir elemanlı yoğun nesne kümeleri verilmektedir.

Tablo 6.2.Nitelik destek değerleri

HAVA	Destek Değeri	ISI	Destek Değeri	NEM	Destek Değeri	RÜZGÂR	Destek Değeri	OYUN	Destek Değeri
Güneşli	5	Sıcak	4	Yüksek	7	Hafif	8	Hayır	5
Bulutlu	4	Ilık	6	Normal	7	Kuvvetli	6	Evet	9
Yağmurlu	5	Soğuk	4						

%40 destek değeri için; Minimum destek sayısı = 0.40 * 14 = 5.6 = 6

Tablo 6.3.Destek değerini sağlayan aday kümeler

ISI	Destek Değeri	NEM	Destek Değeri	RÜZGÂR	Destek Değeri	OYUN	Destek Değeri
Ilık	6	Yüksek	7	Hafif	8	Evet	9
		Normal	7	kuvvetli	6		

Algoritmanın 2. adımında tüm aday nesnelerin ikili kombinasyonları alınmakta ve bu ikililerin destek değerleri hesaplanarak 2 elemanlı aday nesne kümeleri elde edilmektedir (Tablo 6.4). Daha sonra aday nesne kümesinden minimum destek değerini sağlamayanlar elenmekte ve yoğun nesne kümeleri elde edilmektedir. Tek destek değerli klasik birliktelik kuralı Bölüm 3'te anlatıldığı şekilde yeni aday nesne kümeleri üretilemeyene kadar sürdürülmektedir.

Tablo 6.4. İki elemanlı aday kümeler

Kurallar	Destek Değeri

Isı Ilık –Nem Yüksek	4
Isı Ilık- Nem Normal	2
Isı Ilık- Rüzgar Hafif	3
Isı Ilık –Rüzgar Kuvvetli	3
Isı Ilık –Oyun Evet	4
Nem Yüksek-Rüzgar Hafif	3
Nem Yüksek-Rüzgar Kuvvetli	2
Nem Yüksek-Oyun Evet	3
Nem Normal-Rüzgar Hafif	4
Nem Normal –Rüzgar Kuvvetli	3
Nem Normal-Oyun Evet	6
Rüzgar Hafif- Oyun Evet	6
Rüzgar Kuvvetli-Oyun Evet	3

Tablo 6.5. İki elemanlı aday kümelerden destek değerini sağlayan elemanlar

Kurallar	**Destek Değeri**
Nem Normal-Oyun Evet	6
Rüzgar Hafif- Oyun Evet	6

Seçilen minimum destek değerine göre bu veri seti üzerinde 3 elemanlı aday ve yoğun nesne kümeleri üretilemediğinden algoritma bu adımda sonlanmaktadır. Elde edilen yoğun nesne kümelerinden elde edilen kurallar ise Tablo 6.6'da verilmektedir.

Tablo 6.6. Elde edilen Birliktelik Kuralları

No	**Kural**	**Destek**	**Güven**	**Lift**
1	**Nem Normal → Oyun Evet**	**%42**	**%85.7**	**1.33**
~~2~~	~~Oyun Evet → Nem Normal~~	~~%42~~	~~%66.7~~	~~1.33~~
~~3~~	~~Rüzgâr Hafif → Oyun Evet~~	~~%42~~	~~%75.0~~	~~1.17~~
~~4~~	~~Oyun Evet → Rüzgâr Hafif~~	~~%42~~	~~%66.7~~	~~1.17~~

Tekli destek değeri ile sürdürülen algoritmada güven değerini sağlamayanlar kurallar çıkarıldığında geriye sadece bir kural kalmaktadır. Güven değeri dikkate alınmasa bile Tablo 6.6'dan görüldüğü üzere Hava ve Isı değerleri hiçbir kuralda yer almamıştır. Çünkü Hava değeri için 3 farklı değer (Bulutlu, Yağmurlu, Güneşli), Isı değeri için de 3 farklı değer (Sıcak, Soğuk, Ilık) bulunmaktadır. Kurallarda yer alan Nem, Rüzgâr ve Oyun değerlerinin 2 farklı değeri olduğu ve bu nedenle minimum destek değerin sağlayarak kurallara dahil olduğu görülmektedir.

6.2. Çoklu Destek Değeri ile Uygulama

Tezin bu aşamasında ise Tablo 6.1'de verilen veri setine, bu tezde önerilen çoklu destek değeri kullanan birliktelik kuralı algoritması uygulanmıştır. Birliktelik kuralı

uygularken kullanılacak minimum destek ve güven değerleri için yine sırası ile %40 ve %80 değerleri seçilmiştir. Ancak önerilen yöntemde tüm niteliklere 14*0.4= 5.6 destek sayısı uygulanmak yerine Bölüm 5'te önerilen formül ile her nitelik için ayrı ayrı destek sayıları hesaplanarak uygulanmaktadır. Bu hesaplamalar yapıldığında destek sayıları;

Hava :
$Minsup(i) = \frac{|T|.s.2}{100.\varphi(i)} = \frac{14.40.2}{100.3} = 3.7$ (4)

Isı :
$Minsup(i) = \frac{|T|.s.2}{100.\varphi(i)} = \frac{14.40.2}{100.3} = 3.7$ (4)

Nem :
$Minsup(i) = \frac{|T|.s.2}{100.\varphi(i)} = \frac{14.40.2}{100.2} = 6$

Rüzgar :
$Minsup(i) = \frac{|T|.s.2}{100.\varphi(i)} = \frac{14.40.2}{100.2} = 6$

Oyun :
$Minsup(i) = \frac{|T|.s.2}{100.\varphi(i)} = \frac{14.40.2}{100.2} = 6$

olarak hesaplanmaktadır. Destek sayısı tam sayı olmayan değerler bir üst tamsayıya yuvarlanarak alınmaktadır. Tablo 6.7'de her bir nitelik değerlerine ait destek sayıları ve minimum destek değeri verilmektedir. Tablodan görüldüğü üzere algoritmanın 1. adımında sadece Oyun-Hayır niteliği minimum destek değerini sağlamamakta ve 1 elemanlı aday nesne kümesinden çıkarılmaktadır.

Tablo 6.7. Bir elemanlı Aday Nesne Kümeleri

HAVA Min destek=4		**ISI Min destek=4**		**NEM Min destek=6**		**RÜZGÂR Min destek=4**		**OYUN Min destek=4**	
Güneşli	5	Sıcak	4	Yüksek	7	Hafif	8	*Hayır*	*5*
Bulutlu	4	Ilık	6	Normal	7	Kuvvetli	6	Evet	9
Yağmurlu	5	Soğuk	4						

Algoritma, Bölüm 5'te anlatıldığı şekilde devam ettirildiğinde en fazla 4 elemanlı yoğun nesne kümeleri elde edilebilmekte ve bu kümelerden elde edilen birliktelik kuralları Tablo 6.8'de görülmektedir. Tekli destek değerli klasik birliktelik kuralında sadece 1 kural üretilirken, önerilen yöntem ile güven değeri %100 olan ve lift değeri yüksek olan 5 yeni kuralın da üretilebilmesi sağlanmıştır.

Tablo 6.8. Çoklu Destek Değeri ile elde edilen Birliktelik Kuralları

No	**Kural**	**Güven**	**Lift**
1	Nem Normal → Oyun Evet	%85.70	1.33
2	Isı_Soğuk→Nem_Normal	%100.00	2.00
3	Hava_Bulutlu→Oyun_Evet	%100.00	1.55

4	Hava_Yağmurlu-Rüzgar_Hafif→Oyun_Evet	%100.00	1.55
5	Hava_Yağmurlu-Oyun_Evet→Rüzgar_Hafif	%100.00	1.75
6	Isı_Soğuk-Oyun_Evet→Nem_Normal	%100.00	2.00

Tablo 6.8'de görülen kuralların tekli destek değeri kullanan geleneksel yöntemle de elde edilmesi mümkündür. Ancak bu kuralları elde edebilmek için %40'dan daha küçük bir destek değerinin seçilmesi gerekecektir. Bu da algoritmayı yavaşlatacak ve çok daha fazla ve ilginçlik değeri düşük kurallar üretilmesine neden olacaktır. Tablo 6.9'da hem tekli destek değeri ile hem çoklu destek değeri ile oluşturulan kurallar görülmektedir.

Tablo 6.9. Tekli Destek Değeri ve Çoklu Destek Değeri ile elde edilen Birliktelik Kuralları

Tekli Destek=%40, Güven %80 **Tüm Kurallar**	**Çoklu Destek=%40, Güven %80** **Tüm Kurallar**
Nem_Normal → Oyun_Evet : Güven= % 85.71	Nem_Normal→Oyun_Evet : Güven= % 85.71, lift= 1.33
	Isı_Soğuk→Nem_Normal : Güven= % 100.00, lift= 2.00
	Hava_Bulutlu→Oyun_Evet : Güven= % 100.00, lift= 1.55
	Hava_Yağmurlu-Rüzgar_Hafif→Oyun_Evet : Güven= % 100.00, lift= 1.55
	Hava_Yağmurlu-Oyun_Evet→Rüzgar_Hafif : Güven= % 100.00, lift= 1.75
	Isı_Soğuk-Oyun_Evet→Nem_Normal : Güven= % 100.00, lift= 2.00

7. SONUÇLAR

Bu tez çalışmasında, günümüzde neredeyse her alanda kullanılan ve bilhassa ticari hayata yeni uygulamalar kazandıran bir disiplin olarak ortaya çıkmış Veri Madenciliği (VM) anlatılmış ve VM'de çok kullanılan yöntemlerden biri olan birliktelik kuralları çıkarımı konularına ilişkin kavramlar incelenmiştir. Birliktelik kuralı türlerinden biri olan Nicel Birliktelik Kuralları ve bu kurallardaki ilginçlik ölçütü detaylı olarak incelenmiş ayrıca destek değerinin seçimi sorunu çoklu minimum destek değeri ile çözülmeye çalışılmıştır.

Çoklu destek değeri önerilen bu çalışmada, tek destek değeri sorununa çözüm üretmek amacı ile her bir niteliğin aldığı değerlere bağlı olarak farklı minimum destek değerlerinin uygulanması ile bu sorunun çözüldüğü görülmektedir. Böylece birliktelik kuralı algoritmasına giriş olarak tek bir minimum destek değeri girilmekte ancak algoritma her bir niteliğe bu değeri farklı uygulayarak yoğun nesne kümelerini hesaplamaktadır. Bu sayede ilginçliği yüksek kuralların elde edilmesi mümkün kılınmaktadır. Ayrıca tekli destek değeri kullanan geleneksel birliktelik kuralında çok sayıda ilginç kuralları elde etmek için destek değerinin düşük tutulması sorunu ortadan kalkmaktadır.

Önerilen yöntem örnek bir veri seti üzerinde detaylı olarak anlatılmış ve ayrıca literatürde yaygın olarak kullanılan "Oyun" veri seti üzerinde de test edilmiştir. Elde edilen sonuçlar, önerilen yöntemin nicel birliktelik kurallarında rahatlıkla kullanılabileceğini göstermektedir. Bu sayede nitelikler arası ölçek farklılıklarının nicel birliktelik kurallarında oluşturduğu sorun giderilmiştir. Ayrıca, önerilen yöntemin daha büyük veri setlerinde test edilmesi ve elde edilen kuralların daha detaylı olarak karşılaştırılıp verimliliğinin test edilmesi, gelecekte yapılabilecek çalışmalar olarak önerilmektedir.

KAYNAKLAR

[1] **Akpınar, H., 2000**, "Veri tabanlarında bilgi keşfi ve veri madenciliği", İ.Ü. İşletme Fakültesi Dergisi, 29, 1-22.

[2] **Agrawal, R., Imielinski, T., Swami, A**., 1993, "Mining association rules between sets of items in large databases", In ACM SIGMOD Conf. Management of Data.

[3] **Agrawal, R. and Srikant, R**., 1994, Fast algorithms for mining association rules, Proceeding of the 20th Int'l Conference on Very Large Databases, Santiago, Chile.

[4] **Agraval, R., Manilla, H., Srikant, R., Toivonen, H., Verkamo, A.I**., 1996, Fast discovery of association rules, Advanced in Knowledge Discovery and Data Mining, 307-328.

[5] **Manila, H., Toivonen, H., Verkamo**, A.I., 1994, Efficient algorithms for discovering association rules. In Proceedings of AAAI'94 Workshop on Knowledge Discovery in Databases (KDD'94), 181-192, Seattle, Washington, USA.

[6] **Houtsma, M., Swami, A.**, 1995, Set-Oriented mining for association rules in relational databases, Proceedings of the 11th IEEE International Conference on DataEngineering, 25-34, Taipei, Taiwan,

[7] **Srikant, R.**, 1996, Fast algorithms for mining association rules and sequential patterns, Wisconsin Üniversitesi Doktora Tezi, Madison.

[8] **Park, J.S., Chen, M.S., Philip, S.Y**., 1995, An effective hash based algorithm for mining association rules, Proceeding of ACM SIGMOID, s175-186, San Jose, California, USA.

[9] **Savasere, A., Omiecinski, E., Navathe, S.B.**, 1995, An efficient algorithm for mining association rules in large databases, 21. International Conference on Very Large Databases, 432-444, Zurih, İsviçre

[10] **Holsheimer, M., Kertsen, M., Manila, H., Toivonen, H.**, 1995, A perspective on databases and data mining, Proceeding of 1st International Conference on Knowledge and Data Mining, 150-155, Montreal, Kanada

[11] **Toivonen, H.,** 1996, Sampling large databases for association rules, 22. International Conference on Very Large Databases, 134-145, Mumbai, Hindistan

[12] **Brin, S., Motwani, R., Ullman, J.D., Tsur, S**., 1997, Dynamic Itemset Counting and Implication Rules for Market Basket Data, Proceedings of the ACM SIGMOD Conference, 255-264.

[13] **Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W**., 1997 New algorithms for fast discovery of association rules, 3. International. Conference. on Knowledge Discovery and Data Mining.

[14] **Bayardo, R.J.,** 1998, Efficiently mining long patterns from databases, Proceeding of ACM SIGMOID International Conference 85-93 Seattle, Washington, USA.

[15] **Hidber, C**., 1999, Online association rule mining, proceedings ACM SIGMOD International Conference on Management of Data, 145-156, Philadephia, Pennsylvania,

[16] **Chun Hao Chen,Guo Cheng Lan,Tzung Pei Hong,Shih Bin Lin** April 2016, Mining fuzzy temporal association rules by item lifespans☆

[17] **Gosain, A., Maneela B**., 2013, A comprehensive survey of association rules on quantitative data in data mining." Information & Communication Technologies (ICT), 2013 IEEE Conference on.

[18] **Ouyang, W**., 2012, Mining positive and negative fuzzy association rules with multiple minimum supports. In Systems and Informatics (ICSAI), 2012 International Conference on (pp. 2242-2246). IEEE.

[19] **Lee, Y. C., Hong**, T. P., & Lin, W. Y., 2005. Mining association rules with multiple minimum supports using maximum constraints. International Journal of Approximate Reasoning, 40(1), 44-54.

[20] **Zongker, D., Jain, A**., 1996, Algorithms for feature selection: An evaluation, Proceedings of ICPR 96.

[21] **Chatterjee, C., Roychowdhury, V.P., Chong, E.K.P**., 1998, On relative convergence properties of principal component analysis algorithms, Neural Networks, IEEE Transactions on, 9 (2), 319- 329.

[22] **Karabatak, M., İnce, M.C., Avcı, E**., 2008, Göğüs kanseri için temel bileşen analizi yöntemi tabanlı uzman teşhis sistemi, IEEE 16. Sinyal İşleme ve İletişim Uygulamaları Kutultayı, Didim-Aydın.

[23] **Duda R.**, 2001, Pattern Classification, John Wiley, 117-124,

[24] **Kitler, J.,** 1978, Feature set search algorithm, Pattern Recognition and Signal Processing, C. H. Chen, Ed., 41-60, Hollanda

[25] **Hyvarinen, A., Oja, E.,** 1999, Independent component analysis: Algorithms and applications, Neural Networks, 13, 411-430.

[26] **Hyvärinen, A.**, 1999, Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks 10(3), 626-634.

[27] **Duda, R., Hart, P., Stork, D**., 2001, Pattern classification, 2nd ed. New York: John Wiley and Sons.

[28] **Genç, H.M., Çataltepe Z., Pearson, T.,** 2007, Yeni bir temel/bağımsız bileşen analizi (TBA/BBA) tabanlı öznitelik seçme yöntemi, IEEE 15. Sinyal İşleme ve İletişim Uygulamaları Kutultayı, Eskişehir.

[29] **Stearns, S. D.**, 1976, On selecting features for pattern classifiers, Third International Conference on Pattern Recognition, 71-75.

[30] **Pudil, P., Ferri, F.J., Novovicova, J., Kittler, J**., 1994, Floating search methods for feature selection with nonmonotonic criterion functions. IEEE 12th International Conference on Pattern Recognition, 2, 279–283.

[31] **Narendra, P.M., Fukunaga, K.,** 1977, A Branch and Bound algorithm for feature subset selection, IEEE Transaction on Computers, 26, no:9, 917-922.

[32] **Yu, B., Yuan B.,** 1993, A more efficient branch and bound algorithm for feature subset selection. Pattern Recognition, 26(6), 883-889.

[33] **Vriesenga, M.R.,** 1995, Genetic selection and neural modeling for designing pattern classifier. Doctora Tthesis, University of California-Irvine

[34] **Siedlecki, W., Sklansky, J.,** 1989, Anote on genetic algorithms for large-scale feature selection. Pattern Recognition Letters, 10, 335-347.

[35] **Arivazhagan, S. Ganesan, L.,** 2003, Texture classification using wavelet transform, Pattern Recog. Lett. 24, 1513-1521.

[36] **Weszka, J.S., Dyer, C.R., Rosenfeld, A.,** 1976, A comparative study of texture measures for terrain classification, IEEE Trans. Syst. Man. Cybernet. 6, 269-285,

[37] **Muneeswaran, K., Ganesan, L., Arumugam, S. Soundar, K.,** 2005, Texture classification with combined rotation and scale invariant wavelet features, Pattern Recognition, 38, 10, 1495-1506.

[38] **Li, S. Taylor, J.S.,** 2005, Comparison and fusion of multiresolution features for texture classification, Pattern Recognition Letters, 26, 5, 633-638.

[39] **Zhao, Y., Sourav S. B.,** 2015, Association Rule Mining with R. A Survey Nanyang Technological University, Singapore (2015).

[40] **Dilly, Rulh**. 1995, Data Mining: An Introduction www-pcc.qub.ac.uk/tec/ courses/datamining/stu_notes/dm_book_1.html (10/06/1999).

[41] **Alataş, B**., 2002, Veri madenciliği, Fırat Ünv. Fen Bilimleri Enstitüsü Yüksek Lisans Semineri, Elazığ.

[42] **Quinlan, J. R**., 1986, "The effect of noise on concept learning. In Machine Learning: In An Artificial Intelligence Aproach", R. Michalski, J. Carbonell, and T. Mitchell, (eds.), 2, 149-166, SanMateo, CA: Morgan Kauffmann Inc.

[43] **Chan K.C.C., Wong, A.K.C.**, 1991, A statistical technique for extracting classificatory knowledge from databases, Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. J. Frawley, eds.), 107-123, Cambridge, MA: AAAI/MIT.

[44] **Özkan, Y.,** 2008, Veri Madenciliği Yöntemleri, Papatya Yayıncılık, İstanbul

[45] **Türkoğlu İ.,** 1996, Yapay sinir ağları ile nesne tanıma, F.Ü. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Elazığ, 112s.

[46] **Biçer, P.,** 2002, Veri madenciliği: Sınıflandırma ve tahmin yöntemlerini kullanarak bir uygulama" Yıldız Tek. Ünv. Sosyal Bil. Ens. Yüksek Lisans Tezi, İstanbul.

[47] **Agrawal, R. and Shafer, J.C.,** 1996, "Parallel mining of association rules: Design, Implementation and Experience", IBM Research Report RJ 10004.

[48] **Han, J., Kamber, M.,** 2001, Data Mining: Concepts and Techniques, Morgan Kaufman Publishers, 225

[49] **Özden, B., Ramaswamy, S., Silberschatz, A.,** 1998, Cyclic association rules, In Proceedings of the 16th British National Conference on Database, 49-63, Cardiff, Wales.

[50] **Srikant, R., Agrawal, R.**, 1995, Mining generalized association rules, proceedings of the 21nd International Conference on Very Large Databases, 407-419, Zurich, İsviçre

[51] **Han, J., Fu, Y.,** 1995, Discovery of multiple-level association rules from large Databases, Proceedings of the 21nd International Conference on Very Large Databases, 420-431, Zurich, İsviçre.

[52] **Fortin, S., Liu, L.**, 1996, An object-oriented approach to multilevel association rule mining. In Proceeding of 5th International Conference on Information and Knowledge Management, 56-72, Rockville, Maryland, USA.

[53] **Thomas, S., Sarawagi, S**., 1998, Mining generalized association rules and sequential patterns Using SQL Queries, In Proceeding of the 4th International Conference on Knowledge Discovery on Data Mining, 344-348, New York City, USA

[54] **Shen, L., Shen, H.**, Mining flexible multiple-level association rules in all concept hierarchies, In Proceeding of 9th International Conference on Database and Expert Sysyem Applications, 786-795, Vienna, Avusturya.

[55] **Dunham, M.H., Xiao, Y., Gruenwald, L., Hossain, Z**., 2000. A survey of association rules, ACM Survey Journal.

[56] **Srikant, R., Vu, Q., Agraval, R**., 1997, Mining association rules with item constraints, In Proceeding of 3rd International Conference on Knowledge Discovery and Data Mining, California, USA,

[57] **Aggarwal, C.C., Sun, Z., Yu P.S**., 1998, Online algorithms for finding profile association rules, Proceedings of the ACM CIKM Conference, 86- 95.

[58] **Tsur, D., Ullman, J.D., ağabeyteboul, S., Clifton, C., Motwani, R., Nestorov, S., Rosenthal, A**., 1998, Query flocks: a generalization of association rule mining, Proceedings ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA.

[59] **Srikant, R., Vu, Q., Agrawal, R.,** 1997, Mining association rules with item constraints, American Association for Artificial Intelligence,

[60] **Korn, F., Labrinidis, A., Kotidis, Y., Faloutsos, C**., 1998, Ratio rules: A new paradigm for fast quantifiable data mining, Proceedings of the 24th VLDB Conference.

[61] **Brin, S., Motwani, R., Silverstein, C.**, Beyond market baskets: Generalizing association rules to correlations, Proceedings of the ACM SIGMOD Conference, 265-276.

[62] **Agrawal, R., Srikant, R**., 1995, Mining sequential patterns, In Proceedings of the 11th IEEE International Conference on Data Engineering, Taipei, Taiwan, IEEE Computer Society Press.

[63] **Srikant, R., Agraval, R**., 1996, Mining sequential patterns: Generalizations and performance improvements, In Proceedings of the 5th International Conference on Exten ding Database Technology, 3-17, Avignon, Fransa

[64] **Guralnik, V., Wijesekera, D., Srivastava, J**., 1998, Pattern directed mining of sequence data. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, 51-57, New York, USA.

[65] **Das, G., Lin, K., Manila, H., Renganathan, G., Smyth, P.**, 1998, Rule discovery from time series. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, 16-22, New York, USA

[66] **Cai, C.H., Fu, A.W.C., Cheng, C.H., Kwong, W.W**., 1998, Mining association rules with weighted items, In Proceedings of 1998 International Database Engineering and Applications Symposium, 68-77, Cardiff, Wales.

[67] **Savasere, A., Omiecinski, E., Navathe, S.B.,** 1998, Mining for strong negative associations in a large database of customer transaction, In Proceedings of the 14nd International Conference on Data Engineering, 494-502, Orlando, Florida, USA.

[68] **Yuan, X., Buckles, B., Yuan, Z., Zhang, J**., 2002, Mining negative association rules, Seventh International Symposium on Computers and Communications, 623-629, Taormina–Giardini Naxos, İtalya,

[69] **Savasere, A., Omiecinski, E., Navathe, S.B.,** 1995, An efficient algorithm for mining association rules in large databases, In Proceedings of the 21nd International Conference on Very Large Databases, 432-444, Zurich, İsviçre

[70] **Ramaswamy, S., Mahajan, S., Silberschatz, A**., On the discovery of interesting patterns in association rules, 1998, In Proceedings of the 24th International Conference on Very Large Datanasaes, 368-379, New York, USA

[71] **Han, J., Dong, G., Yin, Y**., 1999, Efficient mining of partial periodic patterns in time series databases, In Proceedings of the 15th International Conference on Data Engineering, 1206-115, Sydney, Avustralya.

[72] **Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T**., 1996, Mining optimized association rules for numeric attributes, PODS96, 182-191, Montreal, Quebec, Kanada.

[73] **Wang, K., Tay, S.H.W., Liu, B**., 1998, Interestingness based interval merger for numeric association rules, In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, 121-128, New York, USA

[74] **Rastogi, R, Shim, K.**, 1998, Mining optimized association rules with categorical and numerical attributes, In Proceedings of 14th International Conference on Data Engineering. 503-512, Orlando, Florida, USA.

[75] **Jiawei Han and Micheline Kamber**, "Data Mining:Concepts and Techniques", 2000

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı: Yalçın ATEŞ

Doğum Yeri ve Tarihi: Adana-1977

Eğitim Durumu

Lisans Öğrenimi: Fırat Üniversitesi Teknik Eğitim Fak. Bilgisayar Öğretmenliği

Deneyim

Çalıştığı Kurumlar ve Yıl:

- Adana-Karataş ÇPL- 2 Yıl
- Kuleli Askeri Lisesi-1 Yıl
- Kahramanmaraş Altınşehir Mes. Tek. And. Lisesi 14 Yıl (Devam)

İletişim

E-posta Adresi: y_ates@hotmail.com

Gsm: 0 533 233 33 14