ÖNSÖZ

Bu çalışmanın her aşamasında bana yardımcı olan ve beni destekleyip motive eden Sayın Yard. Doç. Dr. Ali ERCENGİZ'e, veri madenciliği konusuyla ilgilenmeme aracı olan Sayın Prof. Dr. Gazanfer ÜNAL'a ve her zaman yanımda yer alan ailem ve arkadaşlarıma teşekkürlerimi sunarım.

Aralık, 2004 Ayşen BÜYÜKAKIN

İÇİNDEKİLER

KISALTMALAR

ASP	: Active Server Pages
BFB	: Bulanık Fonksiyonel Bağlılık
BVTYS	: Bulanık Veri tabanı Yönetim Sistemi
CART	: Classification and Regression Trees
CHAID	: Chi Square Automatic Iteraction Detector
DFB	: Dereceli Fonksiyonel Bağlılık
Dipnot	: Diploma Notu
DVD	: Digital Video Disc
EDA	: Exploratory Data Analysis
FB	: Fonksiyonel Bağlılıklar
GIGO	: Garbage in Garbage out
I/O	: Input/Output
IIS	: Internet Information Services
İMKB	: İstanbul Menkul Kıymetler Borsası
KDD	: Knowledge Discovery in Databases
k-NN	: K-En Yakın Komşuluk
Matagr	: Matematik Ağırlıklı Notu
MS	: Microsoft
OLAP	: Online Analytical Processing
ÖSS	: Öğrenci Seçme Sınavı
SQL	: Structured Query Language
TC	: Türkiye Cumhuriyeti
VT	: Veri tabanı
VTYS	: Veri tabanı Yönetim Sistemi
www	: World Wide Web

TABLO LİSTESİ

ŞEKİL LİSTESİ

SEMBOL LİSTESİ

τ	: Bir kuralın doğruluk değeri
$\approx$	: Bulanık benzerlik operatörü
α	: Kelime
A_j	: Veri tabanındaki alanlar
G_i	: Dereceli ifade
G_i'	: Dereceli ifadenin tersi
$Q_i . A_j$	: Q_i nesnesinin A_j alanındaki değeri
Q_i	: Veri tabanındaki herhangi bir nesne
Q	: Niteleyici
S	: Özetleyici
$\mu(A)$	: Abulanık kümesinin üyelik derecesi
$\mu_{yüksek}(A)$	: Abulanık kümesinde üyelik derecesi yüksek olanlar

BULANIK MANTIK İLE VERİ MADENCİLİĞİ

ÖZET

Günümüzde, insanların gerçekleştirdiği hemen hemen tüm faaliyetler kayıt altına alınmaktadır. Örneğin bir marketten alışveriş yaparken, bir arkadaşınıza para havale ederken, üretimde kullanılacak malzemelerin depoya girişini kontrol ederken, işletmeler arasındaki günlük rutin ilişkileri gerçekleştirirken faaliyetler, veri tabanlarında kaydedilmektedir. Bu tür verilerin boyutları her geçen gün hızla artmaktadır. Saklanması gereken verilerin bu kadar hızlı çoğalması, hedef bilgiye ulaşmada kullanılan geleneksel sorgulama ve raporlama tekniklerinin yetersiz kalmasına neden olmaktadır. İşte veri madenciliğinden bu büyük veri yığınları arasından bilgilerin elde edilmesinde yararlanılmaktadır.

Veri madenciliğinde istenilen bilgiye ulaşmak için birçok farklı algoritma ve teknik kullanılmaktadır. İnsanın düşünme ve düşündüklerini ifade etme şekline uyan yöntemlerden biri bulanık mantıktır. İşte bu özelliklerinden dolayı bu tez çalışmasında birçok teknik arasından bulanık mantık tercih edilmiştir.

Çalışmada veri madenciliği ve bulanık mantık hakkında bilgi verildikten sonra veri madenciliği ve bulanık mantığın kesişiminden söz edilmekte ve son olarak Windows ortamında ASP ve MS Access kullanılarak geliştirilen; bulanık sorgulama, dilsel özetleme yapılabilen ve bulanık fonksiyonel bağlılık ile dereceli fonksiyonel bağlılık hesaplamalarının da gerçekleştirildiği uygulama hakkında bilgi verilmektedir.

DATA MINING WITH FUZZY LOGIC

SUMMARY

In our time, nearly all activities performed by people are recorded. For instance, shopping in a market, transferring money to a friend, checking of materials into warehouse that will be used in production, routine daily operations between the organizations are recorded in the databases. The dimensions of these data are rapidly increasing day by day. Fast increase in data that must be stored is leading to insufficient traditional queries and reporting techniques. Thus, data mining is useful in gathering data from huge data mountains.

Miscellaneous algorithms and techniques are in use in data mining to retrieve the data needed. One of these methods that suits the human thinking and his way of expressing thoughts is fuzzy logic. Therefore, because of these features, among many other techniques, fuzzy logic has been chosen in this thesis study.

This study contains information about how data mining and fuzzy logic intersect. Finally it explains the application, which is generated by in Windows platform by using ASP and MS Access. In this application a user can fuzzy questioning, linguistic summary, and examine fuzzy functional dependency and gradual functional dependency.

1. GİRİŞ

Veri madenciliği, veri tabanlarında bilgi keşfi (*Knowledge Discovery in Databases – KDD*) konsepti içinde yer alan, bilgi çıkarımı, bilgi hasatı, veri arkeolojisi gibi terimlerden biridir. Veri tabanlarında bilgi keşfi, aynı zamanda veri madenciliğinin arkasında yatan mantık şu şekilde tanımlanabilir:

Veri tabanlarında bilgi keşfi, veri içerisinden geçerli, yeni, potansiyel olarak kullanışlı ve sonuçta anlaşılabilir örüntüler (pattern) tanımlama sürecidir [10].

Bu ifadedeki veri, gerçekler kümesidir. Örüntüler ise, verinin bir alt kümesinin ya da bu alt kümeye uygulanabilecek modellerin tanımını gösteren herhangi bir dildeki ifadedir. Örüntülerin çıkarılması aynı zamanda veriye bir modelin uydurulmasını, verilerden bir yapı bulunmasını veya genel olarak bir veri setinin yüksek seviyeden tarifini de belirtmektedir. Süreç kelimesi ise veri tabanında bilgi keşfedilirken bir çok adımın tekrarlanarak uygulandığını ifade etmektedir. Bulunan örüntü yeni bir veri üzerinde belirli bir kesinlikle geçerli olmalıdır. Örüntülerin en azından sistem ya da kullanıcı için yeni olması ve aynı zamanda potansiyel yararlar içermeleri, örneğin kullanıcıya ya da bir işe fayda sağlaması, istenmektedir. Son olarak örüntüler anında olmasa da sonunda anlaşılır olmalıdır. Bütün koşullar sağlanınca örüntüler bilgiye dönüşecektir.

Veri tabanlarında bilgi keşfi, 1989 yılında veriden bilgiyi arama uğraşında sınırsız ve 'yüksek seviyede' olma genel konseptini ifade etmek için geliştirilmiştir. Veri madenciliği terimi ise yüksek seviyedeki uygulama teknikleri/araçları anlamına gelmektedir ve karar vericilere veri sunmakta, verileri analiz etmekte kullanılmaktadır. Veri madenciliği terimi daha çok istatistikçiler, veri analistleri ve yönetim bilişim sistemleri (management information systems) topluluğu tarafından kullanılırken, yapay

zeka ve makine öğrenmesini (machine learning) araştıran kişiler veri tabanlarında bilgi keşfi tanımını kullanmaktadır [6].

Başka bir deyişle, veri madenciliği, verilerin içerisindeki örüntülerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir [16].

Veri madenciliğinde, istatistik, yapay zeka, makine bilgisi, veri tabanı ve yüksek performanslı işlemler kullanıldığından aynı zamanda disiplinler arasıdır.

Gartner Group veri madenciliğini "depolarda tutulan verilerin gözden geçirilmesi, örüntü farketme teknolojilerinin ve istatistiki, matematiksel tekniklerin kullanılmasıyla yeni, anlamlı korelasyonların, örüntülerin ve trendlerin bulunma süreci" olarak tanımlamaktadır.

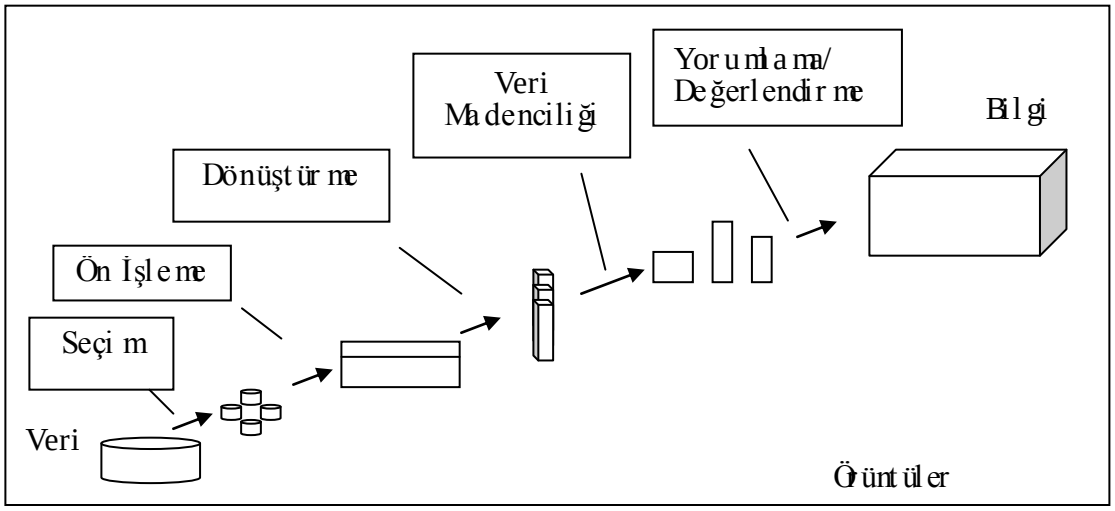
Son olarak diyebiliriz ki, veri madenciliği büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır [2].

2. VERİ MADENCİLİĞİ VE İLİŞKİLİ KAVRAMLAR

Bu bölümde veri madenciliği, ilişkili olduğu kavramlar, kullanılan teknikler, uygulama alanları gibi çok farklı boyutlarda ele alınarak detaylı incelenecektir.

2.1. Veri Tabanında Bilgi Keşfi Sürecinde Veri Madenciliği

Veri tabanlarında bilgi keşfi süreci Şekil 2.1'de gösterildiği gibi şu adımlardan tekrar tekrar yinelenmesiyle gerçekleşmektedir [9]:



Şekil 2.1 Veri Tabanlarında Bilgi Keşfi Süreci

- Uygulama alanı, daha önceki bilgiler ve son kullanıcının hedeflerinin anlaşılması.
- Hedef veri grubunun oluşturulması: verinin seçilmesi veya keşfin gerçekleşeceği değişkenlerin bir alt kümesine ya da veri örneklemlerine odaklanılması.
- Verilerin temizlenmesi ve ön işleme: gürültünün (noise) ve istisna, dışa düşenlerin uzaklaştırılması, gürültünün raporlanması veya modellenmesi için gerekli bilginin

toplanması, kayıp veri alanları ile başa çıkma stratejilerinin geliştirilmesi, bilinen değişikliklerle ve zaman sırasına göre bilginin raporlanması.

- Veri çıkarma ve gösterme: görevin hedeflerine göre verinin sunulmasında kullanışlı olan yeni özellikler bulmak, boyutsal çıkarma veya dönüştürme yöntemlerini kullanarak göz önünde bulundurulan değişken sayısını azaltmak veya veri için değişmez gösterimler bulmak.

- Veri madenciliği görevini seçmek: veri tabanında bilgi keşfi sürecinin hedefinin ne olduğuna (sınıflandırma, regresyon, kümeleme vs) olduğuna karar vermek.

- Veri madenciliği algoritmasına karar vermek: veri içerisinde bulunan örüntüleri araştırmada hangi yöntemlerin kullanılacağını seçmek; hangi modellerin veya parametrelerin daha uygun olduğuna karar vermek; veri tabanında bilgi keşfi sürecinin genel kriterine uygun belirli bir veri madenciliği yöntemi bulmak.

- Veri madenciliği: belirli bir gösterim formunda veya bu gösterimlerin bir kümesinde; sınıflandırma kuralları, ağaçlar, regresyon, kümeleme vs ile istenilen örüntülerin araştırılması.

- Madencilik yapılarak bulunmuş örüntülerin yorumlanması.

- Keşfedilen bilginin birleştirilmesi [10].

Veri madenciliği bu süreçte, verilerden örüntülerin (veya modellerin) belirli bir sırasını yaratan sayısal tekniklerin belirli bir etkinlik sınırı içinde uygulanması olarak ifade edilebilir. Veri madenciliği daha çok verilerden örüntülerin çıkarıldığı ve sıralandığı algoritmalar olarak düşünülmektedir.

2.2. Neden Veri Madenciliği

Bilimsel veri toplamadaki (uzaktan algılayıcılar/ uydular), barkod işlemesindeki ve hükümet işlemlerindeki gelişmeler, verilerin hacmini büyütmüştür. Gelişmiş veri depolama teknolojileriyle de birleşince, veri tabanı yönetimi ve veri ambarı

teknolojilerinin geniş çapta kullanılması, verinin büyüklüğünü oldukça artırmaktadır. Genetik kod projeleri ve astronomi araştırmaları terabayt düzeyinde veri üretmektedir. Uydular ve uzaktan algılayıcılar saatte 50 gigabayt veri üretmektedirler. Yeryüzünde her 20 ayda bir bilgi kendini ikiye katlamaktadır.

Organizasyonlar artan veri ile ne yapacakları problemi ile karşı karşıya kalmışlardır. Geleneksel sorgulama ve raporlama araçlarının büyük veri yığınları karşısında etkisiz kalması veri tabanında bilgi keşfi adı altında faaliyetler yapılmasına ve dolayısıyla veri madenciliğinin ortaya çıkmasına neden olmuştur.

Veri madenciliğinin özellikle işletmelerde karar vermede kullanılmasının birçok nedeni vardır:

- büyük veri tabanlarında kullanılmayan değerler,
- veri tabanı kayıtlarının tek müşteri görüntüsüne doğru birleştirilmesi,
- veri tabanlarının birleştirilmesinden doğan bilgi veya veri ambarı kavramı,
- veri depolama ve işlemede kullanılan donanım sistemlerinin maliyet/performans oranlarındaki inanılmaz düşüş. Beş yıl önce terabayt düzeyinde verinin saklanması 10 milyon dolar civarındayken, bugün bu tutar 1 milyon doların altına düşmüştür.
- doyuma ulaşan pazarlardaki yoğun rekabet,
- imalatı ve pazarı özelleştirebilme ve küçük pazar segmentlerine yönelik reklam yapabilme,
- veri madenciliği ürünleri için pazarın 1994'ün başlarında 500 milyon dolar olarak tahmin edilmesi.

2.3. Veri Madenciliğinin Gelişimi

Etkin kararların mevcut doğru verilere dayanan bilgilerle alındığı çok uzun zamandan beri bilinmektedir. Karar vermede doğru verilerin bulunmasında değerlendirme ve

geliştirme 30 yıl önceden başlamıştır ve gelişimi çeşitli aşamalarla devam etmiştir. Tablo 2.1'de bu süreç gösterilmektedir [13].

Tablo 2.1. Veri Madenciliğinin Gelişimi

Aşama	İşletme Sorusu	Olanaklı Kılan Teknolojiler	Üreticiler	Nitelikleri
Veri Toplama	Geçtiğimiz beş yıl içinde kazancımız ne kadar oldu?	Bilgisayarlar, kasetler, diskler	IBM, CDC	Geçmişle ilgili statik veri dağıtımı
Veri Erişimi	Geçen Mart Marmara'daki birim satışlar ne kadardı?	İlişkisel veri tabanı yönetim sistemleri (RDBMS), yapısal sorgulama dili (SQL), Açık veri tabanı bağlantısı (ODBC)	Oracle, Sybase, Informix, IBM, Microsoft	Geçmişle ilgili, kayıt seviyesinde dinamik veri dağıtımı
Veri Sorgulama	Geçen Mart Marmara'daki birim satışlar ne kadardı? İstanbul'u yakından göster (drill down).	On-line Analytical Processing (OLAP), çok boyutlu veri tabanları, veri ambarları	Pilot, IRI, Arbor, Redbrick, Evolutionary Technologies	Geçmişle ilgili, birçok seviyede veri dinamik veri dağıtımı
Veri Madenciliği	İstanbul'da birim satışlar ne şekilde gelişecek? Neden?	Gelişmiş algoritmalar, çok işlemcili bilgisayarlar,	Lockleed, IBM, SGI, sayısız yeni açılışlar	Umulan, proaktif bilgi dağıtımı

Veri madenciliğinin gelişim aşamaları şu şekildedir:

- Veri Toplama: 1960'larda, önceden biçimlendirilmiş bilgilerden oluşan raporlar, veri tabanlarında bulunan verilerden yararlanılarak oluşturulmaktaydı. Bir başka ifadeyle belirli karar verme gereksinimlerini karşılamak için yapısal raporlardan yararlanılmaktaydı. İşte veri tabanları verileri saklarken uygulamalar, bu raporları hazırlamak için verileri düzeltip yönetiyorlardı.

- Veri Erişimi: 1980'lerde ise, kullanıcılar bilgiye daha sık ulaşmayı ve aynı zamanda bilginin daha kişisel olmasını istemeye başladılar. Sonuçta veri tabanlarında sorgulamalar yapmaya, bilgiye yönelik taleplerde bulunmaya başladılar. Bunlardan genellikle yapısal raporlardan çok, düşük seviyede detay içeren tek kullanımlık

bilginin elde edilmesinde yararlanıyorlardı. Sistem geliştiriciler genellikle sorgulamaları tanımlıyor ve sistem içerisinde inşa ediyordu.

- Veri Sorgulama: Sonraları 1990'larda, kullanıcılar daha detaylı bilgilere anında erişme gereksinimi duydular. Bir başka deyişle, "uçan" soruların cevaplarını aradılar. Bilgiyi, ürün ve karar verme süreçleriyle ilişkilendirebilmek için tam zamanında olmasını istediler. Bu, bütün kullanıcıların bilgi gereksinimlerinin sistemde, daha önceden programlanmış olarak bulunamayacağı anlamına geliyordu. Bu aşamada kullanıcılar kendi sorgulamalarını yazmaya ve veri tabanından ihtiyaç duydukları bilgiyi çıkarmaya başladılar.

- Veri Madenciliği: Son yıllarda kullanıcılar elde ettikleri bilginin, uygulamaları için daha anlamlı olmasını sağlamak amacıyla, veriler arasındakileri ilişkileri belirlemeye ve bulmaya yarayan daha fazla araca, tekniğe ihtiyaçları olduğunu fark ettiler. Bununla birlikte şirketlerde çok geniş hacimlerde veri biriktirdiklerinin farkına vardılar ve sonuç olarak bu verileri düzenleyip bilgi gereksinimlerini karşılayacak araçlara ihtiyaç duydular. Bu tür araçlar son kullanıcının doğrudan müdahalesi olmadan sistemin, veriler içindeki saklı ilişkilerin araştırılmasına olanak sağlar. Veri madenciliği araçları ilk olarak bilim adamlarına geleneksel yollarla yapılması çok zaman ve kaynak alan, büyük veriler arasındaki ilişkilerin veya örüntülerin bulunmasında yardımcı olmak için geliştirilmiştir.

2.4. Veri Madenciliği ve İstatistik

Veri madenciliği geleneksel istatistik tekniklerini yerini almamıştır. Daha çok istatistikçiler topluluğunda meydana gelen değişikliğin bir sonucu olan istatistiksel yöntemlerin bir genişlemesidir [3]. Veri madenciliği ve istatistiğin bir çok ortak yönü olmasına rağmen, bir o kadar da farklı yönleri bulunmaktadır. İkisinin de verilerin yapısını incelemesi gibi örtüşen konuların varlığı nedeniyle, insanlar veri madenciliğini istatistiğin bir alt kolu olarak görmektedirler; ancak bu pek de doğru değildir. Çünkü veri madenciliği istatistikten farklı olarak diğer bilimsel alanlarla beraber çalışarak yeni fikirler, araçlar ve yöntemler geliştirmektedir.

İstatistik ve veri madenciliği arasındaki farkların belli başlılarını şu şekilde gözler önüne serebiliriz:

- İstatistik belli tutucu tarafları olan bir bilimdir. Temelinde matematik yattığından bir yöntemin uygulanabilmesi için öncelikle ispatlanması gerekir. Oysa veri madenciliği bilgisayarlardan yararlandığından ispat ön koşulu yoktur ve daha çok deneysel bir yaklaşım olarak nitelendirilebilir.

- İstatistikte betimleyici yöntemler bulunmasına rağmen, istatistiğin genel olarak çıkarımlarla ilgilendiğini belirtirsek çok daha yanlış yapmış sayılmayız. Genel olarak istatistik bilimi, bir örneklemden yola çıkarak bütün hakkında fikir sahibi olmaya çalışır. Veri madenciliğinde de çıkarım vardır ancak aralarındaki fark kullandıkları verilerin büyüklüğündedir.

- Verilerin bu kadar geniş hacimde olması, istatistikçilerin "elle" gerçekleştirdikleri işlemlerin yetersiz kalacağını ve veri madenciliğinde kullanılan bilgisayarların gerekli olduğunu işaret etmektedir.

- İstatistiğin veri madenciliği ile bir bakıma örtüştüğü bir diğer konu da modellerdir. İstatistikte modelleri, teorik ve teorik olmayan olmak üzere iki farklı grupta toplayabiliriz. Teorik modeller, genellikle gözlemlenen veri üzerindeki değişkenlerin analizi teorisine dayanırken, teorik olmayan modeller olası açıklayıcı değişkenleri tekrar tekrar kullanılarak tahmin edici gücü yüksek modeller oluşturulmaya çalışmaktadır. Veri madenciliğinde ikinci tür model tanımı kabul görmektedir.

- İstatistikte model her şeyin özünü teşkil ederken, hesaplama, model seçim kriteri gibi faktörler ikinci plandadır. Ancak istatistikte de doğrusal olmayan çok boyutlu analiz (nonlinear multivariate analysis) denilen bir yöntemde istenirse modelden istenirse teknikten başlanabilmektedir. Veri madenciliğinde merkezde algoritmalar bulunmaktadır. Algoritmaların temeli oluşturmasının bir nedeni, veri madenciliğinin bilgisayar ve benzer alanlara olan ilişkisidir.

- İstatistik daha çok doğrulayıcı analizle ilgilenmektedir. Doğrulayıcı analizde, bir modelin uyumu dikkate alınır. Diğer bir ifadeyle önerilen modelin gözlemlenen verilere iyi bir açıklama getirip getirmediğiyle ilgilenmektedir. Aksine veri madenciliği daha çok betimleyici bir süreçtir. Beklenilmeyen ancak değerli bilgilerin keşfidir. Betimleyici veri analizleri istatistikçiler için yeni değildir. Belki de bu yüzden istatistikçiler veri madenciliğini başlattıklarını düşünmektedir. Ancak veri madenciliğinde kullanılan verilerin büyüklüğü burada da temel farklılık olarak göze çarpmaktadır.

- İstatistikte sunulan bir probleme uygun olarak verilerin toplanması da önem taşırken, örneğin deneysel yöntemle mi yoksa anket yöntemiyle mi toplandığı, veri madenciliğinde ise verinin toplanmış olduğu kabul edilmekte ve esas olarak veriler arasındaki sırların ortaya çıkarılması üzerinde yoğunlaşılmaktadır.

- İstatistikte veriler uzun bir aradan sonra kullanılmaktadır. Oysa veri madenciliğinde gerçek zamanlı verilerden yararlanılmaktadır.

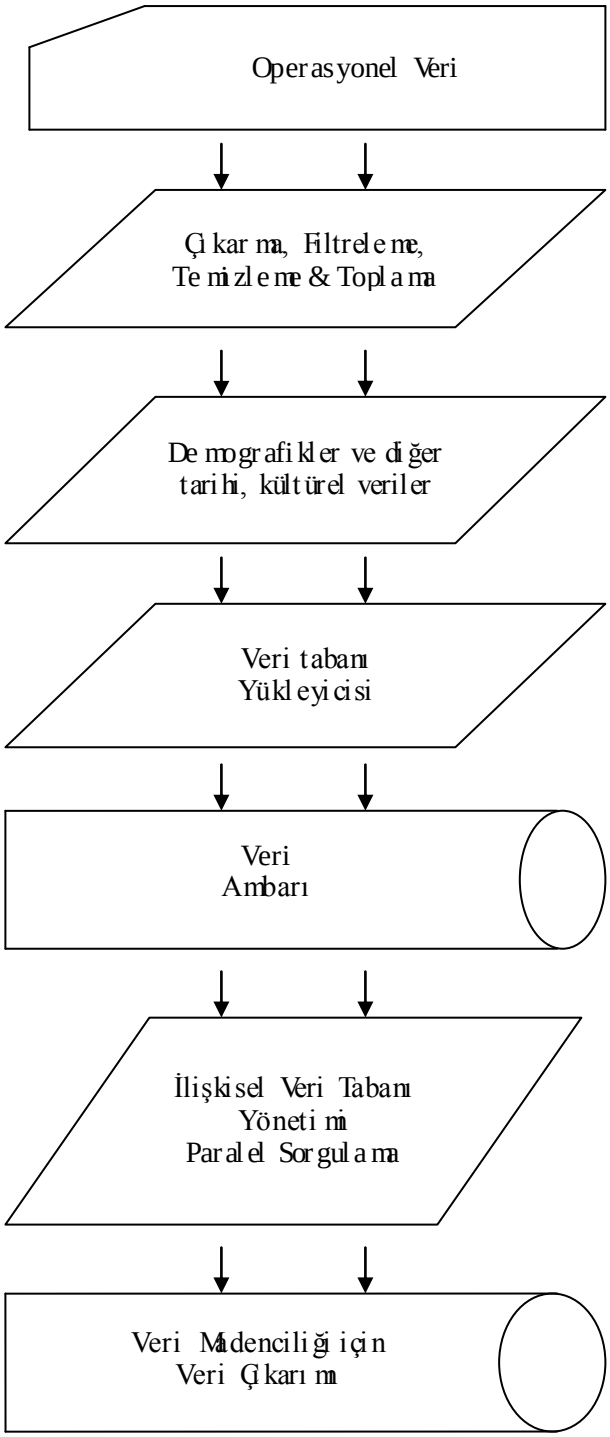
- İstatistikte sayısal verilerle ilgilenilirken, veri madenciliğinde veri resim, yazının bir parçası gibi diğer biçimlerde de olabilmektedir.

- Verileri istatistik kullanarak ele alan kişilerin uzman olması gerekmektedir. Oysa veri madenciliğinde böyle bir gereksinim yoktur [8, 18].

2.5. Veri Madenciliği ve Veri Ambarı

Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş ve işlemeye daha uygun bir özetini saklamayı amaçlar

Veri madenciliği sürecinin en önemli adımı, çok geniş hacimdeki verilerin, son kullanıcılar tarafından düzeltmenin, yorumlamanın ve sınıflandırmanın kolaylıkla gerçekleştirildiği kategori formlarına dönüştürülmesidir. Verinin 'madencilik' için toplanması bile kendi başına zor bir süreçtir. Veri, çıkarımı için çok da uygun olmayan arşiv formunda saklanmaktadır.



Şekil 2.2 Veri Ambarı ve Veri Madenciliği Süreci

Veri ambarı araçları iki çeşittir: veri dönüştürme, temizleme veya iptal etme ile son kullanıcı veri erişim araçları. Bu araçlar veri ambarının veri bütünlüğüne, zaman

çizgileri arasında kararlılığa, yüksek etkinliğe ve düşük işletme maliyetlerine sahip olmasını garanti eder. Veri ambarının en önemli elemanı, verinin hızlı erişime olanak tanıyan farklı özet seviyelerinde saklanıyor olmasıdır. Bu noktadan sonra veri, veri madenciliği için çıkarılabilir.

Hızlı yükleme ve paralellik için önceden gerekli olan sistem altyapısı yüksek I/O bant genişliğidir. Şekil 2.2, veri ambarlarına yön veren veri kaynaklarını ve süreçlerini göstermektedir.

Paralel akış maliyet etkin, ölçülebilir bir paralelliğin veri ambarlarında kritik teknoloji olduğunu göstermektedir. Veri çıkarım süreci madencilik için kullanışlı veri alt kümelerini çıkarır. Belirleyicileri örnekleme ve seçmek çıkarılan verinin boyutlarını sınırlandırabilirken, toplama (bir araya getirme) ilgili verileri özetlemektedir. Veri temizlemesi verinin geçerliliğini garanti eder ve verilerin gereğinden fazla tutulmasını (data redundancy) minimize eder. Normalleştirme gereksiz yere tutulan verilerin miktarını azaltmak amaçlı kullanılabilmektedir ama bazen veri erişiminin hızlandırmak için diğer tarihi veya kültürel özellikleri kullanmak gerekmektedir, örneğin demografikler – özellikle pazar araştırmalarında.

Veri ambarcılığındaki en büyük problem verinin kalitesidir. GIGO (garbage in garbage out) prensibinden kaçınmak için, verinin çok az değer kaybetmesi gerekmektedir; çünkü bu veri madenciliğinin sonuçlarını etkileyecektir. Burada anahtar nokta, verinin veri ambarına eklendiği andan itibaren sürekli olarak izlenmesi ve veri bütünlüğünü garanti altına almak için veri madenciliğinin ön hazırlık safhalarında verinin biçimsel olarak sınanmasıdır.

2.6. Veri Madenciliği ve OLAP

Veri ambarında veri oluşturulduktan sonra bu verinin elle veya gözle analizi yapılabilir. Bunun için OLAP (*Online Analytical Processing*) programları kullanılır. Bu programlar veriye her boyutu veride bir alana karşı gelen çok boyutlu bir küp olarak bakmayı ve incelemeyi sağlar. Böylece boyut bazında gruplama, boyutlar arasındaki korelasyonları

inceleme ve sonuçları grafik veya rapor olarak sunma olanağı sağlar. Kullanıcılarına, belirli bir bölgedeki geçmiş yıllara ait veriler üzerinde, gerçekleşen ve planlanan satışların karşılaştırması gibi karmaşık sorgulama yapma olanağı tanır.

Veri işleme uzmanlarının sordukları en yaygın sorulardan biri de veri madenciliği ve OLAP arasındaki farktır. Bunlar birbirlerini tamamlayan farklı araçlardır.

Karar destek araçları yelpazesinde yer alan OLAP, geleneksel sorgulama ve raporlama araçlarından farklı olarak veri tabanlarında ne olduğunu değil de daha ileri giderek, neden bazı şeylerin doğru olduğunu cevaplamaktadır. Kullanıcı bir ilişki hakkında bir hipotez oluşturur ve veriye uyguladığı bir sorgu serisiyle hipotezini doğrular. Örneğin, bir analizci, kredi verme faktörlerini belirlemek isteyebilir. Öncelikle düşük gelirli insanların kötü kredi riskleri olduğunu varsayabilir ve veri tabanını OLAP ile bu varsayımı doğrulamak veya yanlış olduğunu kanıtlamak için analiz edebilir. Eğer bu hipotez veri tarafından çürütülmediyse analizci, o zaman risk belirleyicisi olarak yüksek borca bakabilir. Veri bu tahmini desteklemediyse, kötü kredinin risklerinin en iyi tahmincileri olan borç ve geliri aynı anda deneyebilir.

Diğer bir deyişle, OLAP analizcisi hipotezsel örüntüler ve ilişkiler serisi yaratır ve veri tabanına karşı sorguları, onları doğrulamak veya yanlış olduklarını kanıtlamak için kullanır. OLAP analizi aslında tümdengelimli bir süreçtir. Peki ya analiz edilen değişkenlerin sayısı düzineler hatta yüzlerce ise ne olacak? İyi bir hipotez bulmak ve veri tabanını OLAP ile doğrulamak veya yanlış olduklarını kanıtlamak için analiz etmek daha zor ve zaman alıcı olacaktır [3].

Bununla beraber, OLAP kullanılarak ulaşılan sonuçlar ve değerler mevcut verilerin bir çıkarımı veya bütünüdür. Oysa veri madenciliği, belirli algoritmaları ve arama motorlarını kullanarak, veri içerisindeki görülmesi zor olan örüntüleri ve trendleri keşfeder ve bu örüntülerden kurallar çıkarır. Bu kurallar veya fonksiyonlarla, kullanıcı iş ya da bilimsel alanda aldığı kararları destekleme, gözden geçirme ve sınama imkanı bulur.

Veri madenciliğinde amaç, kullanıcının bilgi çıkarma sürecinde katkısının olabildiğince az tutulması, işin olabildiğince otomatik olarak yapılabilmesidir. Bununla beraber OLAP programlarının insanlar tarafından yönlendirilmesi gerekir. Araştırma, boyut hiyerarşisinde bir seviyeyi belirleyen kullanıcı tarafından gerçekleştirilen sorgulamalarla sürmektedir. Çıkarım ya da modelleme tamamen analiste bırakılmıştır. Analistten verilerin az boyutlu izdüşümlerinden ya da özetlerinden görüntüleme yoluyla ilgi çekecek örüntüler bulması beklenmektedir. OLAP programlarını kullanırken bulunabilecek sonuçlar kullanıcının sormayı düşündüğü sorgularla sınırlıdır. Ama veri içinde kullanıcının hiç aklına gelmeyecek bilgiler de olabilir. Zaten veri madenciliğinde esas amaç bu tip bilgileri bulabilmektir[2].

2.7. Veri Madenciliğinin Kullanım Alanları

Rakipleriyle etkin bir şekilde rekabet edebilmek için işletmeler, veri kaynaklarını çok iyi anlamak zorundadır. Örüntüleri anlamak ve zamanında karar vermek işletmelere rekabette ilerleme sağlar. Veri madenciliği işletmelerin, başta operasyonel veriler olmak üzere tüm verilerini kendi çıkarları için kullanmasında çok önemli bir araç haline gelmiştir. Veri madenciliği bugün, üretim maliyetlerinin nasıl en aza indirileceği sorusuna cevap almakta, envanter yönetiminde ve perakendecilik, pazarlama, bankacılık, finans, üretim, sağlık, sigortacılık, telekomünikasyon gibi sektörlerde yeni iş fikirleri üretilmesinde kullanılmaktadır. Petrol endüstrisinde, bilimde, orman yangınlarının önlenmesinde, kimyasal yapıların tanımlanmasında, suçun ortaya çıkarılmasında ve tıbbi tanılarda da veri madenciliğinden yararlanılmaktadır.

2.7.1. Pazarlama Uygulamaları

Veri madenciliği en çok müşterilerin satın alma örüntülerinin belirlenmesinde kullanılmaktadır. Ayrıca, müşterilerin demografik özellikleri arasında ilişkiler bulunmasında, posta kampanyalarına cevap verme oranının arttırılmasında, mevcut müşterilerin elde tutulması ve yeni müşterilerin kazanılmasında, pazar sepeti analizinde (market basket analysis), müşteri ilişkileri yönetiminde (customer relationship

management), müşteri değerlemede (customer value analysis) ve satış tahminlerinde (sales forcasting) yine veri madenciliğinden yararlanılmaktadır.

2.7.2. Bankacılık Uygulamaları

Veri madenciliği bankacılık sektöründe, farklı finans göstergeleri arasında gizli korelasyonların bulunmasında, kredi kartı dolandırıcılıklarının bulunmasında, kredi kartı harcamalarına göre müşteri gruplarının bulunmasında ve kredi taleplerinin değerlendirilmesinde kullanılmaktadır.

2.7.3. Sigortacılık Uygulamaları

Yeni poliçe talep edecek müşterilerin tahmin edilmesi, sigorta dolandırıcılıklarının tespiti ve riskli müşteri örüntülerinin belirlenmesi, veri madenciliğinin sigortacılık alanındaki uygulamalarına örnek olarak gösterilebilir [1, 10].

2.7.4. İnternet Uygulamaları

Veri madenciliğinin bir diğer uygulama alanı da web içerikleri veya web bağlantı yapılarıdır. Aynı zamanda kullanıcıların interneti kullanım verileri üzerinde de veri madenciliği çalışmaları gerçekleştirilebilmektedir. Böylelikle, kullanıcıların web loglarından olası örüntüler tanımlanmaya çalışılır. Çalışmaların genel amacı ise elektronik ticaret için olası müşterilerin belirlenmesi ve son kullanıcılara sunulan hizmetlerin kalitesinin arttırılmasıdır [14].

2.7.5. Diğer Uygulamalar

Veri madenciliği aynı zamanda çeşitli hastalıklar için en iyi tedavi yönteminin bulunmasında, genetik sıralarla ilgili verilerin analizinde kullanılmaktadır. Kalite kontrolünde, hatalı malların önceden tespitinde, fiyat simulasyonlarında ve hileleri keşfetmede de veri madenciliği kullanılmaktadır.

2.8. Veri Madenciliği Sistemlerinin Sınıflandırılması

Bugün birçok veri madenciliği sistemi mevcuttur ve yeni sistemler de geliştirilmektedir. Bazıları, verilen bir veri kaynağına adanmış veya sınırlı veri madenciliği fonksiyonlarına sahip özel olarak geliştirilmiş sistemlerdir. Bazılarıysa çok yönlü ve ayrıntılıdır. Veri madenciliği sistemleri çok çeşitli kriterlere göre sınıflandırılabilir:

- Madenciliğin yapıldığı veri kaynağına göre sınıflandırma: bu sınıflandırma veri madenciliği sistemlerini mekansal (spatial), multimedya, zaman serisi, metin, www verisi gibi, ilgilenilen verilerin türüne göre sınıflandırmaktadır.

- Kullanılan veri modeline göre sınıflandırma: bu durumda veri madenciliği sistemleri, dayandıkları veri modellerine göre sınıflandırılmaktadır. Veri modelleri ilişkisel veri tabanı, nesneye dayalı veri tabanı, veri ambarı, işlem vs. olabilir.

- Keşfedilen bilgiye göre sınıflandırma: bu sınıflandırma veri madenciliği sistemlerini keşfedilen bilgiye veya kullanılan veri madenciliği işlevlerine göre kategorize eder. Örneğin, nitelendirme, ayrım, birliktelik, sınıflandırma, kümeleme gibi. Bazı sistemler bu işlevleri beraber sunacak kadar kapsamlıdır.

- Kullanılan veri madenciliği tekniklerine göre sınıflandırma: veri madenciliği sistemleri birçok farklı tekniği kullanmaktadır. Bu sınıflandırma veri madenciliği sistemlerini; kullanılan veri analizi yaklaşımlarına göre, makine öğrenmesi, yapay sinir ağları, genetik algoritmalar, istatistik, görüntüleme, veri tabanına dayalı veya veri ambarına dayalı gibi, kategorize etmektedir. Sınıflandırma ayrıca veri madenciliği sürecine dahil olan kullanıcılarla etkileşimi de dikkate almaktadır. Örneğin, sorgu yönlendirmeli sistemler, etkileşimli keşif sistemleri, özerk sistemler. Kapsamlı bir sistem farklı durumlara ve seçeneklere uyum sağlayacak, farklı derecelerde kullanıcı etkileşimi sağlayan çeşitli veri madenciliği teknikleri sunmalıdır.

2.9. Veri Madenciliğinin İşlevleri

Veri madenciliğinin görevleri, tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki ana başlık altında toplanabilir.

Tahmin edici modellerde, sonuçları bilinen verilerden hareket ederek bilinmeyen veya ileride oluşabilecek sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımsız değişken değeri ise kredinin geri dönüp dönmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılır.

Tanımlayıcı modellerde ise kullanıcıların yorumlarında kullanılacak veri ve sonradan verinin sunumunu tanımlayan örüntülerin bulunması amaçlanmaktadır. Örneğin, X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu ailelerle, geliri X/Y aralığından düşük, arabası ve çocuğu olmayan ailelerin satın alma örüntülerinin benzerlik gösterdiğinin bulunması tanımlayıcı modellerle gerçekleştirilebilir.

2.10. Veri Madenciliği Algoritmaları

Çok farklı problemleri çözmeye veya hedeflere ulaşmaya yaran birçok veri madenciliği algoritması bulunmaktadır; ancak en çok kullanılanları birliktelikler (associations), sınıflandırma (classification), ardışık zamanlı örüntüler (sequential patterns) dir. Birliktelikler in temel dayanak noktası, bir maddeler kümesinin bulunmasının diğer maddeleri de içermesi örneğinde olduğu gibi bütün birliktelikleri bulmaktır. Sınıflandırma ya da profil üretme, farklı gruplar için profil üretir. Ardışık zamanlı örüntüler, kullanıcı tarafından belirlenen minimum maliyette ardışık örüntüleri belirler. Kümeleme, bir veri tabanını alt gruplara ve kümelere bölmektedir.

2.10.1. Birliktelikler (Associations)

Alışveriş sırasında bir müşterinin hangi mal ve hizmetleri satın almaya eğilimli olduğunun bilinmesi, müşteriye daha fazla ürün satılabilmesinin yollarından biridir. Birliktelik algoritmasının süper marketler, envanter planlama, raf planlama, doğrudan pazarlamada kullanılan posta iliştirme (attached mailling), promosyon satışlarını planlama örneklerinde olduğu gibi başta pazarlama olmak üzere finans, tıp gibi çok çeşitli uygulama alanları vardır. Örneğin birliktelik kuralları, ürünlerde bulunan barkod okuyucuları sayesinde, işlemlerin tutulduğu bir veri tabanından veri madenciliği aracılığıyla 'alışveriş sepeti'ni veya bir müşterinin dükkanı tek bir ziyaretinde satın aldığı ürünlerin listesini çıkarabilir.

Birliktelik kuralı eş zamanlı olarak gerçekleşen ilişkilerin tanımlanmasında kullanılır:

"Kola alan müşterilerin %75'i aynı zamanda cips de satın almaktadır."

"Düşük yağlı peynir alan müşterilerin %80'i aynı zamanda yağsız yoğurt satın almaktadır."

%75, kuralın tahmin etme gücü ölçüsü olan güven faktörüdür. Sol elde kola, sağ elde cips bulunmaktadır. Algoritma bu kurallardan oldukça fazla üretir. Kuralların daha yüksek güven seviyesine sahip olan bir alt grubunu, listelerin yüzdelerini veya bu kuralı takip eden 'alışveriş sepeti'ni seçmek kullanıcıya bağlıdır.

Aynı zamanda çoklu birliktelikler de yer alabilir:

"Kola ve cips alan müşterilerin %65'i aynı zamanda sos da almaktadır."

"Düşük yağlı peynir ve yağsız yoğurt alan müşteriler, %85 ihtimalle diyet süt de satın alırlar."

Şansa bağlı bir korelasyon mu (kola ve cips satıştaydı) yoksa bilinmeyen ama önemli bir korelasyon mu (aynı zamanda sos da alındı) olduğunu bilmek kullanıcı için çok önemlidir.

Benzerlik algoritması çok satılan ürünlerin, raflara ya da kataloglara yerleştirilmesinde, çok satılacak ürünlerin birlikte gözükecek şekilde uygun olarak düzenlenmesinde kullanılmaktadır. İlişkili ürünlerin envanteri birbirini yakından takip etmelidir. Çapraz satış fırsatlarının belirlenmesi, hizmetlerde ve ürünlerde satış arttırıcı paketlerin, gruplandırmaların ve promosyonların yapılmasında da kullanılmaktadır. Örneğin, süper marketin sos satışları nasıl patlatılabilir, Pepsi promosyonu olsa ne olur, sorularının cevapları birliktelik algoritması sayesinde bulunabilir [1].

2.10.2. Sınıflandırma (Classification)

En çok kullanılan veri madenciliği algoritmalarından biri sınıflandırmadır; çünkü insan düşünce yapısına çok yakındır. İnsanoğlu dünya üzerindeki maddeleri daha iyi anlamak, başkalarına anlatmak için hemen hemen her şeyi sürekli sınıflandırmakta, kategorilere ayırmakta ve derecelendirmektedir. Maddeleri elementlere, köpekleri türlere, ülkeleri şehirlere, şehirleri semtlere vb. kategorize etmektedir.

Veri madenciliğinde geçerli olan sınıflandırma algoritmasında amaç, yeni karşılaşılan bir girdinin özelliklerinin incelenip, bu girdinin daha önceden tanımlanmış olan sınıflardan hangisine atanacağına karar vermektir. Algoritma şu şekilde işlemektedir: Öz niteliklerleriyle (attribute) verilen kayıt kümesi, yani kayıtların sınıflarını belirten etiketler ve belirli bir kayıta hangi etiketin atandığı verildiğinde, sınıflandırma fonksiyonu bu etiketleri araştırır ve her sınıf için kayıtların nitelik tanımlarını üretir. Örneğin kredi analizinde kredi kartı dağıtan bir firma, tanımlayıcılar içeren çok sayıda müşteri kaydı tutmaktadır. Kredi geçmişi bilinen bir müşteri için müşteri kayıt etiketi 'çok iyi', 'iyi', 'orta' veya 'zayıf' olabilir. Sınıflama kuralı da şu şekilde olabilir:

"Çok iyi kredi geçmişine sahip olan müşterilerin, %10'dan daha az borç/varlık oranı vardır."

Bu kural daha sonra yeni veri kümelerinin sınıflandırmasında kullanılabilir.

2.10.3. Ardışık Örüntüler (Sequential Patterns)

Bu teknik, zaman içerisinde ardışık olarak meydana gelen satın almalara veya olaylara bakar. Örneğin, bir perakendeci televizyon satın alan müşterilerinin %60'ının ileride 8mm kamera alacağını keşfedebilir. Benzer bir kural şu şekilde olabilir:

"X ameliyatı yapıldığında, 15 gün içinde %45 Y enfeksiyonu oluşacaktır."

"İMKB endeksi düşerken A hissesinin değeri %15'den daha fazla artacak olursa, üç iş günü içerisinde B hisse senedinin değeri %60 ihtimalle artacaktır."

"Çekiç satın alan bir müşteri, ilk üç ay içerisinde %15, bu dönemi izleyen üç ay içerisinde %10 çivi satın alacaktır[1]."

Bu algoritma en çok dükkanların düzeninde ve satış promosyon çabaları için hedef müşterilerin belirlenmesinde yararlı olmaktadır. Aynı zamanda, katalog firmaları ve finansal araçların fiyatlarını etkileyen ardışık olayları analiz edebilen finansal yatırım firmaları için faydalıdır.

2.10.4. Kümeleme (Clustering)

Kümeleme, veri tabanını birkaç alt gruba veya kümeye bölecektir. Bu istatistiksel veya yapay ya da sembolik denetimsiz çıkarım metotlarıyla gerçekleştirilebilir. Veriler içerisinde küme oluşturulurken dikkat edilen unsur, seçilen her noktanın küme içinde çok yakın (benzer) olmasıdır. Bu benzerlik kullanıcı veya uzman tarafından belirlenen uzaklık fonksiyonu ile tanımlanabilir.

Kümeleme ile sınıflandırmayı birbirinden ayıran en önemli fark, kümelemede sınıflandırmada olduğu gibi önceden belirlenmiş bir takım sınıflara göre bölümleme yapılmamasıdır. Kümelemede, önceden tanımlanmış sınıflar ya da örnek sınıflar bulunmamaktadır. Kayıtların kümelenmesi işlemi, kayıtların birbirlerine olan benzerliklerine göre yapılmaktadır. Oluşan sınıfların hangi anlamları taşıdığının belirlenmesi tamamen analizi yapana kalmıştır. Örneğin hastaların kayıtlarından oluşan

verilerin kümelenmesi sonucunda semptomlardan oluşan kümeler, değişik hastalıklara işaret edebilir.

Kümeleme işlemi çoğunlukla bir başka veri madenciliği işlemi için bir ilk işlem olarak kullanılır. Örneğin, kümeleme işlemi bir pazar payı araştırması için bir ilk işlem olarak uygulanabilir. "Ne tip promosyonlar müşteriler tarafından rağbet görür?" sorusunun cevabını bulmayı kolaylaştırmak için herkese tek bir model uygulamaktan vazgeçip, müşteriler alışveriş alışkanlıklarına göre kümelendirilirse, her küme için "Bu kümedeki müşteriler hangi tip promosyonlara rağbet eder?" sorusunun cevabı çok daha kolay verilir [14].

Çok boyut olduğu zaman bazı kümeleme algoritmaları kullanışsız olur. Bunu engellemek için kümeleme algoritmalarında öncelikle mantıklı boyutlar seçilmelidir. Amaç, verilerin birbirleriyle ilişkilerinin değişebilir olduğu özel boyutları bulmaktır. Bu süreç veri gürültüsünü azaltır ancak aynı zamanda önemli bilgilerin kaybedilmesine neden olabilir.

2.10.5. Niteleme (Characterization)

Veri niteleme hedef sınıftaki nesnelerin genel özelliklerinin özetlenmesidir ve özellik kurallarını üretir. Kullanıcı tarafından belirlenmiş sınıfla ilgili olan veri, normalde bir veri tabanı sorgusu ve çeşitli soyutlama düzeylerindeki verinin özünü seçip çıkartmak için bir özetleme biriminni gözden geçirmeyle elde edilir. Örneğin DVD kiralayan bir dükkan müşterileri arasından yılda 30 film kiralayanları belirlemek isteyebilir. Hedef sınıfını tanımlayan öz niteliklerdeki kavram hiyerarşileriyle, mesela veri özetlemeyi gerçekleştirmek için öz niteliğe dayalı çıkarım yöntemi kullanılabilir.

2.10.6. Veri Görüntüleme

Veri görüntülemede kullanıcılara resimler sunularak analistlerin verileri daha derin bir şekilde anlamaları sağlanmaktadır; çünkü dikkatlerini diğer yöntemler tarafından bulunmuş bir takım örüntülere odaklamaktadır. Örneğin dört değişkenin bulunduğu grafik şekil özlü bir şekilde çok geniş bilgi sunmaktadır. Çeşitli renklerin, boyutların ve

derinliklerin kullanılması ile yeni birliktelikllerin bulunması ve aralarındaki farklılıkların geliştirilmesi mümkündür.

Verilerin görüntülenmesi örüntülerin, ilişkilerin, kayıp ve istisna değerlerin teşhisinde çok faydalı bir tekniktir. Ancak en büyük kısıt görüntülemenin bir çok farklı boyutu iki - üç boyutlu ekrana aktarılmasıdır. Ayrıca veri görüntüleme için geliştirilen araçların kullanımı genellikle iyi bir eğitim gerektirmektedir ve renk körlüğü olan ya da uzaysal analizlerde zorluk yaşayan kişiler için uygun değildir[14].

2.10.7. Ayrım (Discrimination)

Veri ayrımı, fark kuralları üretir ve temel olarak hedef sınıf ve çelişen sınıf olarak bahsedilen iki sınıf arasındaki nesnelerin genel özelliklerinin karşılaştırılmasıdır. Örneğin, bir dükkandan geçen yıl 30 film kiralayan müşterilerin özellikleriyle, 5 yıldan kısa süredir film kiralayanların özellikleri karşılaştırmak istenebilir. Veri ayrımında kullanılan teknikler veri nitelemede kullanılan tekniklere çok benzerdir, ancak veri ayrımı sonuçları karşılaştırmalı ölçümler içerir.

2.10.8. Tahmin Etme (Prediction)

Tahmin etme, özellikle işletmelerde kullanılan kestirim araçlarının yaygın kullanılmasında önem kazanmaktadır. Başlıca iki çeşit tahmin etme türü vardır: ya var olmayan veri değerleri veya kararlaştırılmayan trendler tahmin edilmeye çalışılır ya da bazı verilerin sınıf etiketleri tahmin edilir. İkinci kısım sınıflandırmayla ilişkilidir. Eğitim kümesine bağlı olarak bir sınıflandırma modeli kurulduğunda, bir nesnenin sınıf etiketi, nesnenin öz nitelik değerlerine ve sınıfların öz nitelik değerlerine bağlı olarak öngörülebilir. Tahmin etme, daha çok eksik sayısal değerlerin veya zamanla ilişkili verilerde artan/azalan trendlerin kestirilmesinde kullanılmaktadır. Ana fikir, çok büyük sayıda geçmiş dönem verileri kullanarak gelecekteki olası değerleri hesaba katmaktır.

2.10.9. Aykırı Değer Analizi (Outlier Analysis)

Aykırı değerler, verilen bir grup veya küme içinde gruplanamayan elemanlardır. İstisnalar (dışa düşenler) veya sürprizler olarak da bilinirler; tanımlanmaları sıklıkla

önemtaşır. Aykırı değerler bazı uygulamalarda gürültü ve atılmış (discarded) olarak dikkate alınırken, diğer etki alanlarından önemli bilgileri açığa çıkarabilirler. Bu yüzden kendileri çok önemli, analizleri de çok değerli olabilir.

2.10.10. Evrim ve Sapma Analizi (Evolution and Deviation Analysis)

Evrim ve sapma analizi; zamanla değişen, zamanla ilgili verilerin incelenmesine mahsustur. Evrim analizi, verideki evrimsel trendleri modeller. Bu trendler nitelemeye, karşılaştırmaya, sınıflandırmaya ya da zamanla ilgili verilerin kümelenmesine izin verir. Diğer yandan sapma analizi ise, ölçülmüş değerler ile beklenen değerler arasındaki sapmayı dikkate alır ve sapmaların nedenini beklenen değerlerden bulmaya çalışır [18].

2.11. Veri Madenciliği Teknikleri

Veri madenciliğinde çok çeşitli teknikler kullanılmaktadır. İstatistiksel teknikler, yapay sinir ağları, karar ağaçları, genetik algoritmalar bunlardan birkaçıdır.

2.11.1. İstatistiksel Teknikler

Veri madenciliği istatistikten farklı bir olgu olsa da istatistiksel tekniklerden yararlanmaktadır. Bu teknikler istatistik literatüründe çok boyutlu analiz (multivariate analysis) başlığı altında toplanır ve genelde verinin parametrik bir modelden (çoğunlukla çok boyutlu bir Gauss dağılımından) geldiğini varsayar. Bu varsayım altında sınıflandırma, regresyon, kümeleme, boyut azaltma (dimensionality reduction), hipotez testi, varyans analizi, bağıntı (associations; dependency) kurma için teknikler istatistikte uzun yıllardır kullanılmaktadır [4].

Geleneksel teknikler olarak nitelendirebileceğimiz bu tekniklerin bazılarını şunlardır:

2.11.1.1 Doğrusal ve Lojistik Regresyon

Tahmin edilen alan nümerik bir değişken olduğunda tahmin modeli regresyon olarak isimlendirilmektedir. İstatistikte çok çeşitli regresyon türleri bulunmaktadır. Ancak hepsinin arkasında yatan temel düşünce, tahmin edicilerin değerlerinden yola çıkarak

tahmin sırasında en az hataya neden olacak bir model tasarlamaktır. Regresyonun en basit şekli doğrusal regresyondur. Doğrusal regresyon bir tahmin edici ve bir tahmin içerir. Bu ikisi arasındaki ilişki iki boyutlu bir uzayda haritalandırılabilir ve kayıtlar tahminler için Y ekseninde, tahmin ediciler için de X ekseninde çizilir. Bundan sonra basit bir doğrusal regresyon modeli, gerçek tahmin değeri ile kendi üzerinde bulunan noktalar arasında hata değerini en aza indirgeyen doğru olarak ifade edilebilir [19].

Lojistik regresyon ise doğrusal regresyonun genelleştirilmiş halidir. Genellikle ikili değişkenlerin ve daha seyrek olarak çok sınıflı değişkenlerin tahmin edilmesinde kullanılır. Lojistik regresyon modelleri, ayrık değişkenlerden oluşan olayların olasılıklarının logaritmalarını tahmin eder. Lojistik regresyonun temel kabullenmesi ayrık değişkenlerin katsayılarının logaritmalarının doğrusal olasılıkları olacağıdır. Bu tekniği kullanan analistler, doğru değişkenleri, sonuç değerleri ile fonksiyonel ilişkilerini ve olası etkileşimleri seçebilmek için yeterli deneyime ve beceriye sahip olmalıdır [14].

2.11.1.2 Zaman Serisi Tahmini

Zaman serisi tahminleri, "zamanla değişen tahmin edici serilerin bilinmeyen gelecek değerlerini öngörmekte" kullanılır. Zaman serisi veri tabanları, sıralı değer serilerini ve zaman içinde değişen olayları içermektedir. Bu değerlerin trendleri Y=f(t) şeklinde fonksiyonların kurulmasında kullanılabilir. Böylelikle nitelikler zaman veya diğer süreç değerleri baz alınarak tahmin edilebilir. Çalışmama zamanı kurulumun bir fonksiyonu olarak tahmin edilebilir. Bu bilgi ile önleyici bakım programları uyarlanabilir, çizelgelenebilir ve gerçek zamana ayarlanabilir. Bu teknikte dönemlerin, mevsimselliklerin, takvim etkilerinin ve tarih aritmetiklerinin hiyerarşisi gibi önemli faktörler sonuçları etkileyebilir. Bu nedenle zaman serisi tahminlerinde bu faktörler hesaplanmalıdır [14].

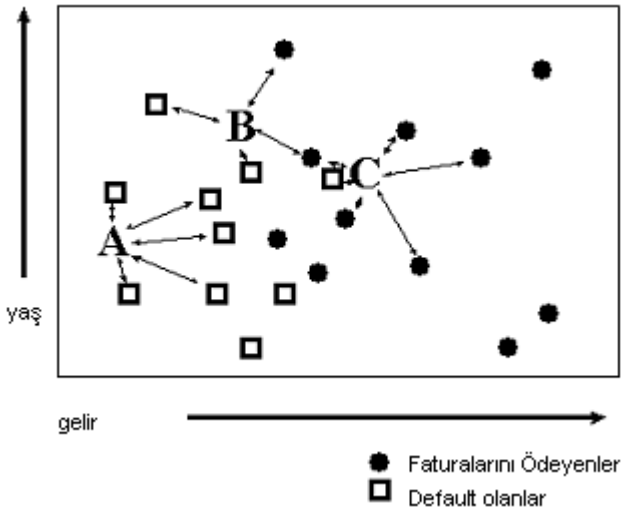
2.11.2 Bellek Tabanlı Yöntemler

Bellek tabanlı veya örnek tabanlı bu yöntemler (memory-based, instance-based methods; case-based reasoning) istatistikte 1950'li yıllarda önerilmiş olmasına rağmen o yıllarda gerektirdiği hesaplama ve bellek yüzünden kullanılamamış; ancak günümüzde

bilgisayarların ucuzlaması ve kapasitelerinin artmasıyla, özellikle de çok işlemcili sistemlerin yaygınlaşmasıyla, kullanılabilir olmuştur. Bu yönteme en iyi örnek en yakın *k* komşu algoritmasıdır (k-nearest neighbor, k-NN) [2].

k-NN veriler rakamsal olduğunda ilişkileri ve dizileri tespit etmek için kullanılan klasik bir tekniktir. Bu teknik, bir objeyi niteliklerini inceleyerek bir sınıf veya gruba yerleştirdikten sonra ona en yakın niteliklere sahip olan objeleri de yine aynı gruba dahil etmektedir. Rakamsal olmayan niteliklerde veya değişkenlerde bu tekniği uygulamak zordur. Çünkü rakamsal olmayan değerlerin arasındaki mesafeyi ölçmek için kullanılan bir metriği tanımlamak zordur. Örneğin mavi ve yeşilin uzaklığı nedir? Objeler arasındaki mesafe ölçüldükten sonra komşuluğun ne kadar geniş olacağına, komşuların ne şekilde ağırlıklandırılacağına ve sonuçta yeni objenin hani sınıfa dahil edileceğine karar verilir.

k-NN bilgisayarda çok geniş hesaplama gerektirir; çünkü hesaplama zamanı, mevcut noktaların faktoriyeli olarak artmaktadır. k-NN'de her yeni vaka için yeni bir hesaplama gerekir. k-NN'i hızlandırmak için bütün veri bellekte tutulur. k-NN modelleri, çok az tahmin edici değişken olduğunda oldukça anlaşılırdır. Uygun metrik bulunduğunda, metin gibi standart olmayan veri tiplerini içeren modeller kurmak mümkündür [3].



Şekil 2.3 Kredi riskleri. Bir k-NN'in Örneği

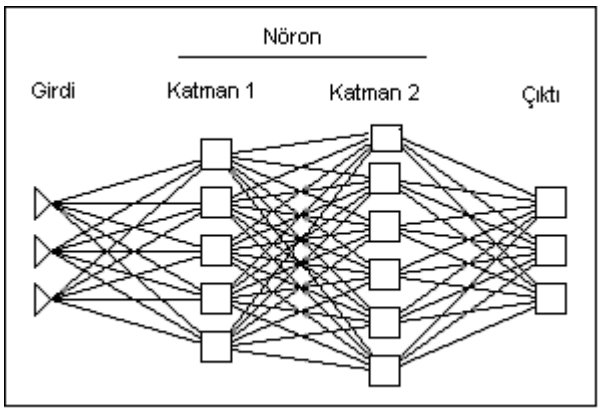
Şekil 2.3'de insanlar kredi risklerine göre gruplandırılmaktadır. Default olarak nitelendirilenler, kredi riski yüksek olanları göstermektedir. Görüldüğü gibi C örneğinin en yakını bir default değerdir ve C'nin çevresi neredeyse kredi kayıtları iyi olan kişilerle doludur. Bu durumda C'nin en yakın komşusu büyük bir olasılıkla bir istisna olacaktır. Bunun sonucunda da veri hatalı olacaktır.

Bu gibi durumlarda tek bir en yakın komşuluğun dikkate alınmasındansa 9 veya 15 en yakın komşuluğun önerilmesi sistemi için daha doğru bir tahmin olanağı sağlayacaktır. Genellikle bu, tahminlerin çoğunun k en yakın komşuluktan alınması ile başarılır [19].

2.11.3. Yapay Sinir Ağları

1980'lerden sonra yaygınlaşan yapay sinir ağlarında (artificial neural networks) amaç fonksiyon birbirine bağlı basit işlemci ünitelerinden oluşan bir ağ üzerine dağıtılmıştır. Yapay sinir ağlarında kullanılan öğrenme algoritmaları veriden üniteler arasındaki bağlantı ağırlıklarını hesaplar. Yapay sinir ağları istatistiksel yöntemler gibi veri hakkında parametrik bir model varsaymaz yani uygulama alanı daha geniştir ve bellek tabanlı yöntemler kadar yüksek işlem ve bellek gerektirmez [2].

Doğrusal olmayan tahmin edilebilir modeller arasında yer alan yapay sinir ağlarında, belirli bir profille uyuşmanın sağlanması için kalıp düzenler kontrol edilmektedir. Bu süreçte belirli bir öğrenme faaliyeti gerçekleştirilerek sistem geliştirilmektedir.



Şekil 2.4 Yapay Sinir Ağları

Yapay sinir ağlarında başlıca üç çeşit katman bulunmaktadır: girdi, gizli ve çıktı katmanları. Bu katmanlar, bir çok düğümden oluşmaktadır. Girdi düğümlerinde, örneğin bir kredi riski sürecinde, gelir, borç, yaş gibi faktörler olacak; çıktı düğümleri ise iyi veya kötü kredi riski sonuçları olacaktır. Her düğüm arasındaki bağlantı ağırlandırılmakta ve girdi değerleri bu ağırlıkla çarpılıp toplanmakta ve bir sonraki katmana iletilmektedir. Bu yolda ilerleyerek çıktı katmanındaki değerler hesaplanır [19].

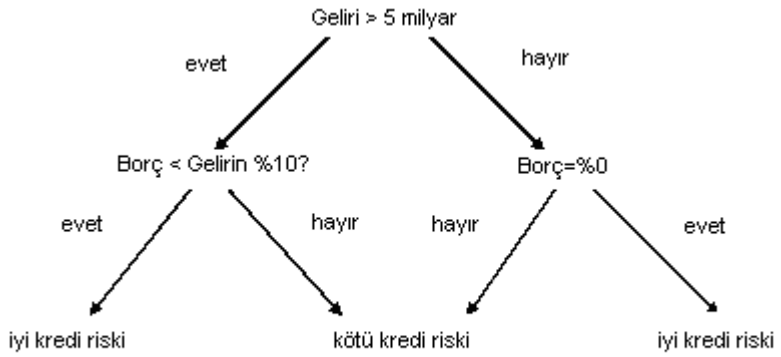
Yapay sınır ağları, eğitim verileri ile ağı eğitir ve sonra tahminlerde kullanır. Yapay sinir ağları, genellikle geniş veri tabanlarında eğitilemez ama uygun örnekleme yöntemleriyle ağ, küçük ve orta boy veri tabanlarında anlamlı doğruluk gösterebilmektedir. Sinir ağlarındaki temel problem sonuçlarla ilgili herhangi bir açıklamanın sunulmayışıdır (kara kutu işlemleri). Bu durum sonuçlara duyulan güveni, sonuçların kabul edilmesini ve uygulanmasını engellemektedir. Bununla beraber yapay ağı, anlaşılabilir kurallar kümesine dönüştüren bazı yapay ağ ürünleri bulunmaktadır. Bu uygulama daha çok el yazılarında örüntü fark etmede (pattern recognition) ve elektrokardiyogramların yorumlanmasında kullanılmaktadır [14].

Yapay sinir ağlarının bir avantajı, birçok paralel bilgisayarda aynı anda çalışacak şekilde kullanılmasıdır. Bu durumda her düğüm kendi hesaplamasını eş zamanlı olarak gerçekleştirmektedir. Yapay sinir ağları çok geniş çeşitlilikteki sorunların çözülmesinde kullanılabilir ve karmaşık durumlarda iyi sonuçlar üretir.

Bununla beraber, yapay sinir ağları kolaylıkla yorumlanmaz. Sinir ağlarının kararlara veya tahminlere yönelik açık bir mantık sunmazlar. İkinci olarak, yapay sinir ağları problem küçük değilse, olukça uzun eğitim süreleri gerektirmektedir. Ancak bir kere eğitildi mi, tahminleri oldukça hızlı gerçekleştirmektedir. Üçüncü olarak, diğer yöntemler gibi oldukça fazla veri hazırlığı gerektirmektedir. Örneğin, bütün değişkenlerin sayısal olması gerekir. Bu nedenle kategorik verilerin sayısal değerlere dönüştürülmesi gerekmektedir. Son olarak, yapay sinir ağları, veri set çok büyük olduğunda ve gürültü sinyali oranı (signal to noise ratio) yüksek olduğunda çok iyi sonuçlar vermektedir. Çok esnek oldukları ndan düşük gürültü sinyali oranında birçok yanlış örüntü bulacaktır [3].

2.11.4. Karar Ağaçları

En çok kullanılan veri madenciliği tekniklerinden biridir. Ağacın her dalı bir sınıflandırma sorusudur ve yaprakları ise sınıflandırmaları ile birlikte veri setinin bir parçasıdır.



Şekil 2.5 Karar ağacı Örneği

Karar ağacı, ağaca benzer biçime sahip bir yapıdır. Görsel olarak bir takım kurallar, şartlar tanımlayarak kararın verilmesine imkan sağlamaktadır. Karar ağaçları ile bir verinin sınıflandırılması için otomatik kurallar üretilebilir. Bu metodoloji verileri sınıflandırmak için if-then ifadelerinden oluşan hiyerarşiyi kullanmaktadır.

Karar ağaçları, olayları, kök düğümden yaprak düğümlere doğru sıralayarak sınıflandıran bir modeldir. Ağaçtaki her düğüm olaya ilişkin bir özelliğin bir değerini göstermektedir. Ayrıca bu düğümden sonra gelen her dal bu özelliğin mümkün olan değerlerlerinden birine uymaktadır. Her olay, karar ağacının kök düğümünden başlanıp bu düğümden ifade edilen özellik test edilerek sınıflandırılır. Daha sonra özelliğin değerlerine uygun olarak ağacın dallarından aşağı doğru gidilir. Bu işlemler bulunduğumuz daldaki düğümde tekrar edilir ve bir yaprak düğüme ulaşana kadar devamedilir. Mümkün olan tüm ağaçları yapmaktansa her birinin büyüklüğü ölçülür ve bunlardan en küçük olan seçilir.

Karar ağacı tekniği hipotez üretme sürecini ve daha sonra geçerliliğini diğer veri madenciliği tekniklerinden daha tam ve daha entegre bir şekilde gerçekleştirmektedir. Ham veri çok az ya da hiç ön işleme olmadan ustaca ele alınmaktadır. Karar ağaçları, kredi kartı uygulamalarından, farklı kurlar arasındaki kambiyo oranın zaman serisi tahminlerine kadar çok çeşitli işletme problemlerinde kullanılmaktadır.

Karar ağaçlarının bazı çeşitleri lojistik regresyon gibi standart istatistiksel teknikler (regresyon ağaçları) için verinin arındırılması ve önişlemesi amacıyla geliştirilmiş keşif araçları olsalar da, tahmin etme amacıyla kullanılmaktadırlar.

Karar ağacı teknolojisi veri kümelerinin ve işletme problemlerinin keşfinde kullanılabilir. Bu genellikle ağacın her bir kesiti için seçilen tahmin ediciye ve değerlere bakılarak gerçekleştirilmektedir. Tahmin ediciler genellikle faydalı kavrayış sunarlar ya da cevaplanması gereken soruları belirtirler.

Karar ağaçlarının bir diğer kullanılma şekli de diğer tahmin teknikleri için verinin ön işlenmesidir. Tahmin edicinin tipi sayı veya kategorik gibi sınırlı olduğundan ve çok hızlı oldukları dan karar ağaçları, veri madenciliğinin ilk uygulandığında yararlı tahmin edici alt kümeler üretilmesinde ve daha sonra yapay sinir ağlarını, en yakın komşuluk veya normal istatistiksel tekniklerini beslemede kullanılır.

Karar ağaçlarının en büyük avantajı, yapay sinir ağlarından daha hızlı ve anlaşılır olmasıdır. Bununla beraber en büyük dezavantajı ise verinin aralık ya da kategorik olma zorunluluğudur. Sürekli veriler bu veri tiplerinde kaydedilir; ancak bu veri içerisinde önemli kırılma noktalarının saklanması olasılığını beraberinde getirmektedir. If-then ifadeleri özellikle koşul listesi uzun olduğunda oldukça karmaşık olabilmektedir.

Karar ağacı tekniklerinden en çok kullanılan iki tanesi CART ve CHAID'dir.

2.11.4.1 CART

CART (Classification and Regression Trees – Sınıflandırma ve Regresyon Ağaçları) Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone tarafından 1984 yılında geliştirilen keşif ve tahmin aracıdır.

CART ağacı oluşturulurken her tahmin edici kayıtları farklı tahminlerle ne kadar iyi parçaladığına göre seçilir. Örneğin, tahmin ediciyi verecek bir noktanın diğerinden daha iyi olduğunu anlamak için kullanılabilen bir ölçü entropi metriğidir.

CART'ın en önemli avantajlarından biri modelin ve algoritmanın içinde kurulan genel optimal modelin geçerli oluşudur. CART bunu çok karmaşık ağaçlar geliştirdikten sonra çapraz geçerliliğin veya test kümesinin geçerlilik sonuçlarından yararlanarak oluşturur. Daha sonra ağacı, genel optimal ağaç şeklini alacak şekilde budar. Ağaç, test kümesi verilerindeki budanmış ağaçların performansı dikkate alınarak tekrar budanır. Çapraz geçerlilik kullanılarak oluşturulan ağaç, yeni ve görülmemiş verilerin seçilmesinde en iyi sonucu vermektedir.

CHART algoritması eksik veriler konusunda da oldukça güçlüdür. Belirli bir kayıttaki bir tahmin edicinin değeri eksikse bu kayıt ağaç oluşturulurken en iyi yarığın belirlenmesinde kullanılmayacaktır. CART en iyi olası yarığın seçiminde çok fazla bilgiden yararlanmaktadır.

CART yeni bir verinin tahmin edilmesinde kullanılırken eksik veriler "vekilleri" aracılığıyla idare edilebilir. Vekiller, ağaçtaki gerçek yarıklar gibi davranan ve tercih edilen tahmin edicinin eksik olması durumunda kullanılabilen yarık değerleri ve tahmin edicilerdir. Örneğin ayakkabı numarası boy için iyi bir tahmin edici değildir; ancak CART ile tahmin edilen belirli bir kayıttaki boyla ilgili veri kaybolduğunda vekil olarak kullanılabilir [19].

2.11.4.2 CHAID

Bir diğer popüler karar ağacı tekniği de CHAID (Chi-Square Automatic Interaction Detector - Ki Kare Otomatik Etkileşim Kontrolü) dir. CHAID, karar ağacı oluşturma konusunda CART'a benzemektedir; ancak yarıkları seçme konusunda CART'tan faklılıklar göstermektedir. En uygun yarığın seçiminde entropi metriği yerine bağımlı değişkeni en fazla etkileyen bağımsız değişken, bağımlı değişkenin sürekli olması durumunda F testi, kategorik olması durumunda ise Ki Kare testi kullanılarak belirlenir.

Kategorik ve sürekli değişkenler üzerinde çalışabilmesi ve ağaçta her düğümü ikiden fazla alt gruba ayırabilmesi gibi nedenlerle tercih edilen bir algoritmadır [19].

2.11.5. Kural Çıkarımı

Kural çıkarımı teknikleri veriler arasındaki birliktelikleri ortaya çıkarmaya çalışır. Mevcut kayıtlar içerisinde benzerlikleri araştırarak bu ilişkileri ifade eden kuralları çıkarır. Verideki olayların belirli bir şekilde meydana geliyor olması kuraldaki güven faktörünün kurulmasında kullanılır.

Daha sonra yeni verilerin değerlerini tahmin etmede kullanılacak olan hiyerarşik olmayan koşul kümeleri oluşturulur. Belirli yazılım uygulamaları, çakışan kurallardan kaçınmak için en iyi tahminleri veren kuralları seçerek, kural kümesini değerlendirme ve ayıklama eğilimindedir. Tahmin sürecinde kullanılan kurallar karar ağaçlarından daha genel ve güçlüdür. Kısmi karar ağaçlarının bulunduğu çok faklı değer alabilen tahmin ormanları şeklinde değerlendirilebilir. Bu tahmin edici modeller tamamıyla saydamdır ve tahminler için bütün açıklamayı sağlarlar.

Örneğin bir kredi kartı firması, betimleyici veya niteleyici müşteri kayıtlarını tutuyor olabilir. Bilinen bir kredi geçmişiyle, bu kayıtlar iyi, orta ve zayıf olmak üzere etiketlenebilir/sınıflandırılabilir. Çıkarım tekniği "kart sahibi 5 milyardan fazla kazanıyorsa, 45-55 yaşları arasındaysa, iyi bir bölgede yaşıyorsa iyi kredi riskine sahiptir" şeklinde bir kural oluşturan sembolik bir sınıflandırma modeli üretebilir.

2.11.6. Kural Çıkarımı ve Karar Ağaçları Arasındaki Farklar

Karar ağaçları da kural üretmektedir; ancak bu kurallar kural çıkarımdan oldukça farklı bir yolla gerçekleştirilmektedir. Karar ağaçları ve kural çıkarımı arasındaki temel farklılıklar şu şekilde listelenebilir:

Karar ağaçları birlikte kullanılamayan (mutually exclusive) kurallar üretirler ve genel olarak bu kurallar veri tabanının eğitilmesi sırasında ayrıntılıdırlar. Oysa kural çıkarımı sistemleri aynı anda kullanılan ve ayrıntılı olabilen kurallar üretmektedir.

Diğer bir deyişle karar ağaçlarında verilen bir kayıt için onu kapsayan bir kural ve kurallar için de sadece bir tek kural olacaktır. Bununla beraber kural çıkarım sistemlerinde verilen bir kayıta uyan çok fazla sayıda kural olabileceği gibi birçok sistem için, çoğu sistem genel önceden belirlenmiş kurallar oluştursa da, rastlanabilecek olası kayıtların her biri için bir kural garantisi vermemektedir.

Bu farklılığın bir nedeni iki algoritmanın farklı işlemesidir. Kural çıkarımı en alttan yola çıkarak ilginç bütün olası örüntüleri toplar ve sonra bu örüntüleri tahmin hedefinde kullanır. Diğer taraftan karar ağaçları "greedy" arama olarak bilinen yöntem gibi tahmin hedefinden başlayarak aşağı iner. Greedy algoritmaları ağacın üst seviyelerinde seçim yapabilir. Kural çıkarım sistemlerinde bütün olası örüntüler kullanılmasalar bile tutulmaktadır.

Örneğin birbiriyle çok fazla ilişkisi bulunan iki kolondan oluşmuş bir veri olsa ve bu kural çıkarımında iki farklı kural olarak sonuçlansa da karar ağaçlarında tek bir tahmin edici seçilecektir; çünkü ikincisi gereksizdir ve tekrar seçilmemektedir. Bir örnek yıllık ve aylık ücretler olabilir. Ücret miktarı tahmin ediliyorsa karar ağacı bu iki tahmin ediciden birini seçecek ve ağacın bir yerinde yarık noktası olarak kullanacaktır. Karar ağacı tahmin değerini, tahmin ediciden yola çıkarak etkin bir şekilde "sıkıştırmış" ve sonra diğerine geçmiştir. Kural çıkarım sistemi ise iki kural üretecektir.

Bu örnekteki tahmin ediciler uç vakaları ifade etmektedir; ancak daha farklı vakalar da mevcuttur. Örneğin, karar ağacında ayakkabı numarası yerine boy kullanılırken kural çıkarımı sistemlerinde ikisi de kural olarak gösterilecektir.

2.11.7. Genetik Algoritmalar

Genetik algoritmalar, optimal sonucun bulunmasında olası girdi parametrelerinin üretilmesi ve test edilmesi olarak ifade edilebilir. Genetik algoritmalar doğal evrim sürecine uyan genetik kombinasyon, mutasyon ve doğal seçim gibi kavramları kullanmaktadır.

Birbirleriyle yarışan potansiyel problemlerin çözümlerinin bir arada bulunduğu bir koleksiyondan en iyi çözüm seçilerek birbirlerine bağlanır. Böylelikle organizmalardaki populasyonun evrim sürecine benzer bir şekilde genel çözüm kümesinin daha da iyileşmesi beklenmektedir. Bir başka ifadeyle; populasyon içerisinde bulunan her birey, çözülmesi gereken problemin potansiyel bir çözümünü temsil etmektedir. Genetik algoritmalarda amaç, çözüm uzayının stokastik global araştırılması ile en sağlıklı bireyin bulunmasıdır.

Genetik algoritmalar diğer veri madenciliği algoritmalarını geliştirmek için kullanılan optimizasyon teknikleridir. Böylece bir veri grubu için en iyi modeli oluşturmak mümkün olmaktadır. Sonuç, model veriye uygulanarak gizli kalmış kalıpları ortaya çıkarmakta ve bu sayede tahminler yapılabilmektedir.

Genetik algoritmalar veri madenciliğinde değişkenler arasındaki bağlılıklarla ilgili hipotezler üretmektedir. Genetik algoritmalar, en iyi segmentasyon/kümeleme uygulamalarında kullanılmaktadır. Aynı zamanda öğrenme içeren durumlar için de kullanılabilmektedir. Geçtiğimiz yıllar boyunca genetik algoritmalar yapay sinir ağlarının eğitilmesi, bellek tabanlı yöntemlerde kombinasyon fonksiyonunun oluşturulması gibi işlerde kullanılmıştır. Ancak karmaşık durumların genetik kodlamasının oldukça zor olduğunu ve optimal sonucun üretilmesini garanti altına almadığını ve genetik algoritmaların çalıştırılması sırasında çok ağır bir işlem yükü getireceğini belirtmek gerekir [7].

2.11.8. Bulanık Mantık

Bulanık mantık belirsizliğin gösterilmesinde ve işlenmesinde anahtar metodolojidir. Bugünün veri tabanlarında belirsizlik bir çok farklı şekilde ortaya çıkmaktadır: kesin olmayış, muğlaklık, tutarsızlık. Bulanık mantık belirsizliği sistemin karmaşıklığını yönetilebilir kılmada kullanır. Kesin ve sert sınırlardan çok bulanık nosyonları ele almaktadır. Örneğin 1 ve 0'ların yanı sıra 0.87, 0.34 gibi değerleri de alabilmektedir.

Bulanık kümeler, sadece tam olmayan, gürültülü veya kesin olmayan verilerle ilgilenmek için değil, aynı zamanda geleneksel sistemlere kıyasla performansları daha

çabuk ve düzgün olan belirsiz veri modellerinin geliştirilmesinde de kullanılmaktadır. Bulanık sistemler belirsizliğe müsamaha gösterebildiklerinden ve dilden kaynaklanan muğlaklıkları da düzgün veri gecikmelerinde kullanabildiklerinden, kesin girdilere ulaşılamadığı veya girdinin çok pahalı olduğu durumlarda güçlü, gürültüye tahammüllü modeller veya tahminler sunmaktadır.

2.11.9. Araştırmacı Veri Analizi (EDA – Exploratory Data Analysis)

Araştırmacı veri analizi veri kümesinin önceden tasarlanmış kabullenmelere ve modellere çok bağlı kalmadan etkileşimli bir şekilde araştırılması ve sonucunda ilgi çekici örüntülerin bulunmasıdır. Verinin grafik gösterimleri, gözlerin gücünden ve insanın sezgisinden yararlanmak amacıyla kullanılmaktadır [7].

2.11.10. Veri Madenciliği Tekniği Seçim Önerileri

Yukarıda bahsedildiği gibi, veri madenciliği teknikleri bir çok farklı veri madenciliği işlevini yerine getirmek üzere kullanılmaktadır. Hangi tekniğin ne zaman kullanılacağına dair bir öneri aşağıdaki tabloda verilmektedir [17]:

Tablo 2.2 Veri Madenciliği Tekniklerinin Seçimi

Sınıflandırma	Kural çıkarım yöntemleri, Karar Ağaçları, Yapay Ağlar, En yakın komşuluk,
Tahmin	Regresyon analizi, Regresyon ağaçları, Yapay Ağlar, En yakın komşuluk
Birliktelik/Bağımlılık	Korelasyon analizi, Regresyon Analizi, Birliktelik kuralları
Veri tanımlama ve özetleme	İstatistiksel teknikler, OLAP
Segmentasyon/Kümeleme	Kümeleme teknikleri, Yapay Ağlar, Görüntüleme yöntemleri

3. BULANIK MANTIK

Mantık, sistem, küme vb. için bulanıklık, belirsizliğin bir ifadesi olarak karşımıza çıkmaktadır. Geçmişte belirsizliklerin işlenmesi ve anlamlı sonuçlara varılabilmesi için olasılık teorisi kullanılmıştır. Matematik ve mühendislikte bu teori, belirsizlik durumlarında istatistiki yöntemlerle beraber kullanılır. Bu nedenle de, bütün belirsizliklerin rastgele karakterde olduğu kavramı yaygınlaşmıştır; ancak bilinen belirsizliklerin hepsi rastgele karakterde değildir. Örneğin sözel belirsizlikler durumunda inceleme ve sonuç çıkarma işlemlerinde olasılık hesabı ve istatistik gibi sayısal belirsizlikleri gerektiren metodolojiler kullanılamaz [15].

Günlük hayatımızda karşılaştığımız bir çok konuda ve sorunda sayısal bilgiler sunmaktan çok kendi yorumlarımızı, göreceli olarak nitelendirebileceğimiz dilsel ifadelerle aktarmaktayız. Örneğin, birçok kişi "çalışkan öğrenci" ifadesine göreceli olarak farklı anlamlar yükleyecektir. Kimine göre çalışkan öğrenci not ortalaması 4,5'un üstünde olan öğrencileri kapsarken kimine göre de 3'ün üstünde not alan öğrenci çalışkan olarak atfedilebilecektir. Sonuç olarak, "çalışkan" kelimesinin betimlediği sayısal anlayışlarda bir belirsizlik bulunmaktadır. Buradaki belirsizlik rast gelelik değildir. İşte kelimelerin içerdiği bu tür belirsizliklere bulanıklık (fuzzy) denmektedir.

Bulanık mantık yaklaşımı, makineler ve insanların özel verilerini işleyebilme ve onların deneyimlerinden ve önsezilerinden yararlanarak çalışabilme yeteneği verir. Bu yeteneği kazandırırken sayısal ifadeler yerine sembolik ifadeler kullanır. Sembolik ifadelerin makinelere aktarılması matematiksel bir temele dayanır. Bu matematiksel temel bulanık mantık küme kuramı ve buna dayanan bulanık mantıktır [5].

Bulanık mantık yaklaşımıyla ilgili ilk ilkeler 1965 yılında Azerbaycan asıllı Lütfü Askerzade Zadeh tarafından ortaya atılmıştır. Zadeh bu çalışmasında insan düşüncesinin

büyük çoğunluğunun bulanık olduğunu, kesin olmadığını belirtmiştir. Zadeh tarafından bulanık mantığın genel özellikleri şu şekilde ifade edilmiştir:

1. Bulanık mantıkta, kesin değerlere dayanan düşünme yerine, yaklaşık düşünme kullanılır.

2. Bulanık mantıkta her şey [0,1] aralığında belirli bir derece ile gösterilir.

3. Bulanık mantıkta bilgi büyük, küçük, çok az gibi dilsel ifadeler şeklindedir.

4. Bulanık çıkarımsel dilsel ifadeler arasında tanımlanan kurallar ile yapılır.

5. Her mantıksal işlem bulanık olarak ifade edilebilir.

6. Bulanık mantık matematiksel modeli çok zor elde edilen sistemler için çok uygundur.

3.1. Klasik ve Bulanık Kümeler

Klasik anlamda küme, belirli bir özelliğe veya birden çok özelliğe sahip ve birbirinden farklı olan elemanların toplamına denir. Örneğin bir zar atıldığında gelebilecek sayılar kümesini

Z={1, 2, 3, 4, 5, 6}

ile gösterebiliriz. Z kümesi ayrık ve sonlu sayıda elemana sahip bir kümedir. Verilen Z küme ifadesinde, 5 kümenin elemanıdır, 7 kümenin elemanı değildir. Görüldüğü gibi klasik küme tanımında belirli bir kümeye ait olmak veya olmamak üzere ayrım yapılmaktadır. Klasik bir kümenin elemanlarını karakteristik fonksiyon kullanarak şu şekilde gösterebiliriz:

$$A(x) = \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases} \qquad (3.1)$$

Bir anlamda karakteristik fonksiyonda yer alan 1 ve 0, x değişkeninin üyelik derecelerini göstermektedir. Görüldüğü gibi klasik kümelerde üyelik dereceleri arasında keskin geçişler bulunmaktadır. Klasik kümeler ve bulanık kümeler arasındaki en önemli fark işte bu üyelik derecelerindedir.

Bulanık kümeleri bir örnek yardımıyla tanımlamaya çalışalım. Aile bireylerinin yaşlarından oluşmuş bir küme verilmiş olsun.

A ={16, 18, 20, 22, 29, 30, 36, 45, 52}

Bu kümenin bir alt kümesi olan orta yaşlılar kümesi de şu şekilde verilsin:

O = {30, 36, 45}

Gençler kümesi ise aşağıdaki şekilde tanımlansın:

G = {16, 18, 20, 22, 29}

29 elemanını ele alalım. 29 gençler kümesinde yer almasına rağmen orta yaşlılar kümesinin alt sınırı olan 30 yaşına, diğer genç kümesi elemanlarına kıyasla daha yakındır. Bir diğer ifade ile orta yaşlılar kümesinin elemanlarıyla yapılan herhangi bir incelemede 29 yaş ihmal edilecektir. Aslında günlük hayatta sınıra yakın değerlerin hangi aralığa düşeceği oldukça bulanıktır. Bu tür önemli olabilecek ihmalleri ortadan kaldırmak amacıyla bulanık küme yaklaşımı kullanılmaktadır.

Bulanık kümelerde üyelik derecelerinin geçişi yumuşak olmaktadır. Bulanık bir küme şu şekilde ifade edilebilir: X boş olmayan bir küme olsun. X altındaki herhangi bir A bulanık kümesi her $x \in X$ için $A: X \rightarrow [0,1]$ ile gösterilebilir. Bir başka ifadeyle her x elemanının [0,1] aralığında bir üyelik derecesi vardır ve A bulanık kümesinin her elemanı üyelik derecesi ile birlikte gösterilmektedir. Bir elemanın üyelik derecesi $\mu(a)$ ile gösterilir. Bundan sonra karışıklığa yer vermemek için bulanık kümeler bir alt çizgi ile gösterilecektir, $\underline{A}$ gibi.

Klasik bir kümenin elemanları

$A = \{a_1, a_2, a_3 \ldots\}$

biçiminde gösterilirken, bulanık küme elemanları

$$\underline{A} = \{\mu(a_1)/a_1 + \mu(a_2)/a_2 + \mu(a_3)/a_3 + \ldots\} = \{\sum_i \mu(a_i)/a_i\} \quad (3.2)$$

şeklinde gösterilir. Bulanık küme sürekli ise

$$\underline{A} = \{\int \mu(a_i)/a_i\} \quad (3.3)$$

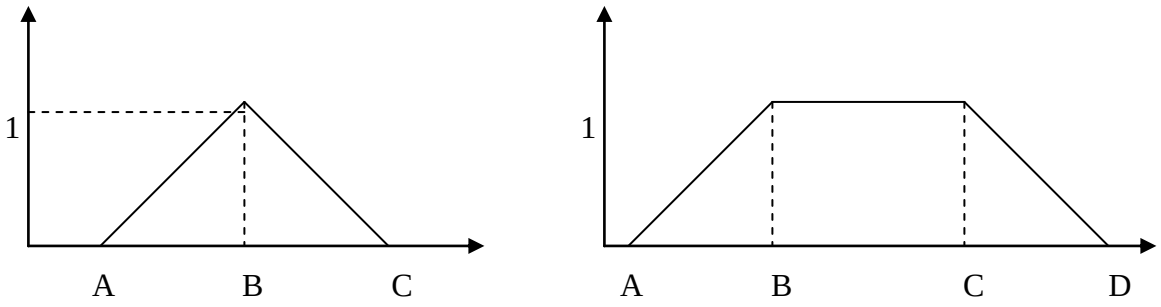
şeklinde gösterilir.

Bu gösterimlerdeki bölme ve toplama işaretleri bilinen bölme ve toplama anlamında olmayıp, üyelik derecesinin hangi küme üyesine ait olduğunu ve elemanlar arasındaki ilişkiyi göstermek amaçlı kullanılmaktadır.

3.1.1. Üyelik Fonksiyonları

Bulanık bir kümenin elemanlarına üyelik dereceleri atayan fonksiyonlara üyelik fonksiyonları denilmektedir. Üyelik fonksiyonlarının en bilinenleri üçgen, yamuk, Sigmoid ve π fonksiyonlarıdır. Kullanılan üyelik fonksiyonuna bağlı olarak küme elemanının alacağı üyelik derecesinin değişebileceği kolaylıkla görülmektedir.

Aşağıda belirli üyelik fonksiyonlarının tanımları ve şekilleri verilmektedir.

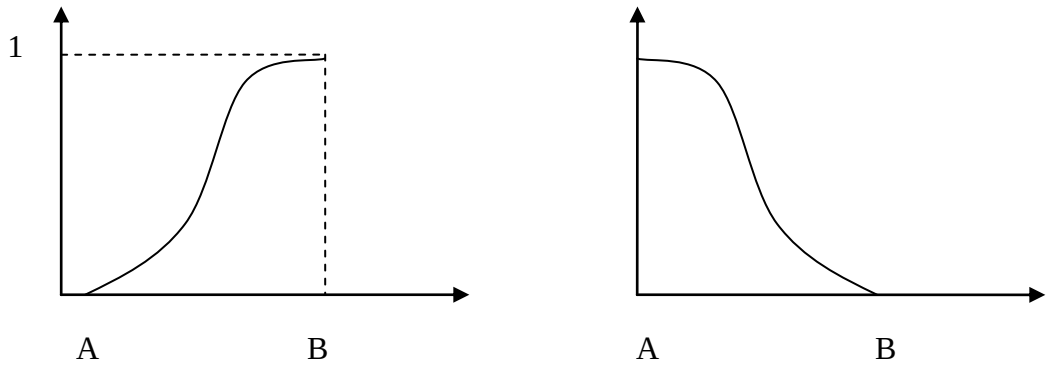


Şekil 3.1 Üçgen ve Yamuk Üyelik Fonksiyonları

$$\text{Üçgen}(x; A, B, C) = \begin{cases} 0 & ; \quad x < A \\ \dfrac{x-A}{B-A} & ; \quad A \le x < B \\ 1 & ; \quad x = B \\ \dfrac{C-x}{C-B} & ; \quad B < x \le C \\ 0 & ; \quad x > C \end{cases} \tag{3.4}$$

$$\text{Yamuk}(x; A, B, C, D) = \begin{cases} 0 & ; \quad x < A \\ \dfrac{x-A}{B-A} & ; \quad A \le x < B \\ 1 & ; \quad B \le x \le C \\ \dfrac{D-x}{D-C} & ; \quad C < x \le D \\ 0 & ; \quad x > D \end{cases} \tag{3.5}$$

Çok kullanılan sigmoid fonksiyonundan elde edilen S ve Z yapıların şekilleri ve formülleri aşağıda verilmektedir.

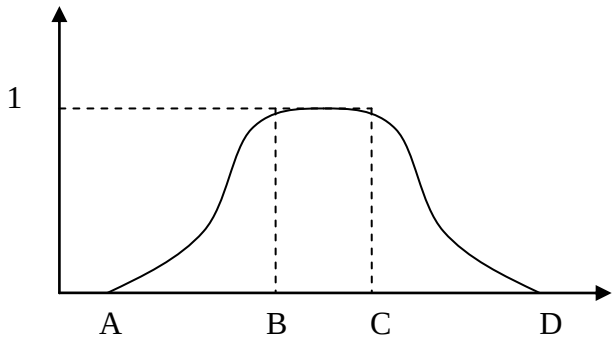


Şekil 3.2 S ve Z Yapısındaki Üyelik Fonksiyonları

$$S(x; A, B) = \begin{cases} 0 & ; \quad x \le A \\ 2 * \left(\dfrac{x-A}{B-A} \right)^2 & ; \quad A < x \le (A+B)/2 \\ 1 - 2 * \left(\dfrac{x-B}{B-A} \right)^2 & ; \quad (A+B)/2 < x \le B \\ 1 & ; \quad x > B \end{cases} \tag{3.6}$$

$$Z(x;\ A,\ B) = \begin{cases} 1 & ;\quad x \le A \\ 1 - 2*(\dfrac{x-A}{B-A})^2 & ;\quad A < x \le (A+B)/2 \\ 2*(\dfrac{x-A}{B-A})^2 & ;\quad (A+B)/2 < x \le B \\ 0 & ;\quad x > B \end{cases} \qquad (3.7)$$

Son olarak π fonksiyonunun şekli ve formülü aşağıdaki gibidir:



Şekil 3.3 Pi Üyelik Fonksiyonu

$$\pi(x;\ A,\ B,\ C,\ D) = \begin{cases} S(x;A,B,C) & ;\quad x < B \\ 1 & ;\quad B \le x \le C \\ Z(x;A,B,C) & ;\quad x > C \end{cases} \qquad (3.8)$$

3.1.2. Küme İşlemleri

Bulanık kümelerde de klasik kümelerde olduğu gibi birleşim, kesişim kümeleri ile bir kümenin tümleyeni tanımlanabilmektedir.

3.1.2.1 Birleşim Küme

Klasik kümelerde A ve B iki küme olmak üzere A birleşim B ($A \cup B$) kümesinin elemanları A veya B kümesine dahil olan elemanlardan oluşur.

$$A \cup B = \{x \mid x \in A \text{ veya } x \in B\} \qquad (3.9)$$

Bulanık kümelerde ise $\underline{A}$ ve $\underline{B}$ bulanık küme olmak üzere, $\underline{A} \vee \underline{B}$ birleşim kümesinin elemanlarına karşı gelen üyelik dereceleri, her bir kümeye ait olan $\underline{A}$ ve $\underline{B}$' deki üyelik derecelerinin en büyüğünün alınmasıyla bulunur.

$$\underline{A} \vee \underline{B} = \max(\mu(\underline{A}), \mu(\underline{B})) \qquad (3.10)$$

3.1.2.2 Kesişim Kümesi

Klasik kümelerde A ve B iki küme olmak üzere A kesişim B ($A \cap B$) kümesinin elemanları hem A hem de B kümesine dahil olan elemanlardan oluşur.

$$A \cap B = \{x \mid x \in A \text{ ve } x \in B\} \tag{3.11}$$

Bulanık kümelerde ise $\underline{A}$ ve $\underline{B}$ bulanık küme olmak üzere, $\underline{A} \wedge \underline{B}$ birleşim kümesinin elemanlarına karşı gelen üyelik dereceleri, her bir kümeye ait olan $\underline{A}$ ve $\underline{B}$' deki üyelik derecelerinin en küçüğünün alınmasıyla bulunur.

$$\underline{A} \wedge \underline{B} = \min(\mu(\underline{A}), \mu(\underline{B})) \tag{3.12}$$

3.1.2.3 Tümleyen Küme

Evrensel kümesi içinde yer alan bir A kümesinin tümleyeni, A kümesinde bulunmayan evrensel kümenin diğer elemanlarının tümü olarak tanımlanır ve $\overline{A}$ ile gösterilir.

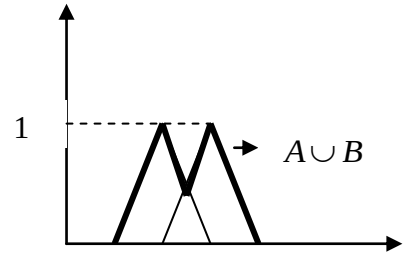
Genel olarak, bulanık bir $\underline{A}$ alt kümesinin tümleyenini $\underline{\overline{A}}$ bulmak için $\underline{A}$ kümesinin elemanlarının üyelik dereceleri 1'den çıkarılmalıdır.

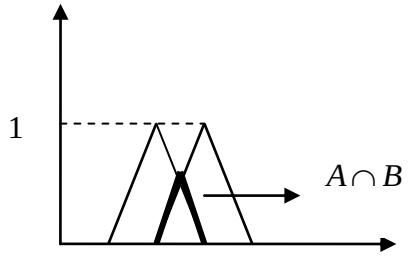
$$\mu(\underline{\overline{A}}) = 1 - \mu(\underline{A}) \tag{3.13}$$

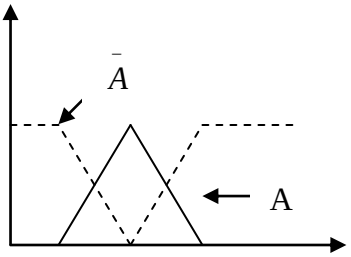
yazılabilir.

$$\overline{\overline{A}} = A \text{ veya } \underline{\underline{A}} = A \tag{3.14}$$

ifadeleri geçerlidir.







Şekil 3.4 Bulanık Küme İşlemleri (Birleşim, Kesişim, Değil)

Bulanık kümelerde birleşim, çarpım ve tümleyen kavramlarını bir örnekle pekiştirelim.

$\underline{A} = \{0.1/a + 0.4/b + 0.7/c + 1.0/d\}$

$\underline{B} = \{0.2/x + 0.3/b + 0.6/z + 0.9/d\}$

bulanık kümeleri verilmiş olsun. Bu durumda

$$\underline{A} \vee \underline{B} = \max(\mu(\underline{A}), \mu(\underline{B})) \tag{3.15}$$

$$= \{0.1/a + 0.4/b + 0.7/c + 1.0/d + 0.2/x + 0.6/z\}$$

$$\underline{A} \wedge \underline{B} = \min(\mu(\underline{A}), \mu(\underline{B})) \tag{3.16}$$

$$= \{0.3/b + 0.9/d\}$$

$$\bar{A} = \{0.9/a + 0.6/b + 0.3/c + 0/d\} \text{ ve } \bar{B} = \{0.8/x + 0.7/b + 0.4/z + 0.1/d\} \tag{3.17}$$

4. VERİ MADENCİLİĞİNDE BULANIK MANTIK

Bulanık mantık, sözel ifadelerin bilgisayar ortamına aktarılmasında kullanılan matematiksel temel olarak adlandırılabilir. Günümüzde büyük veri yığınları arasından işe yarayan bilgileri ortaya çıkarmak oldukça önemlidir. Veri madenciliği olarak adlandırdığımız bu işlemde bulanık mantıktan yararlanılması, veri sorgulamalarında günlük hayatta kullanılan ifadelere daha uygun ifadeler kullanılmasına olanak sağlar ve bu nedenle olası sonuçların göz ardı edilmesini engellemede büyük bir role sahiptir.

Bu tez çalışmasında Dan Rasmussen ve Ronald R. Yager'in birlikte yazmış oldukları "SummarySQL – A Fuzzy Tool For Data Mining" ve "Finding fuzzy and gradual functional dependencies with SummarySQL" başlıklı makaleleri temel alınmıştır. Bu bölümde öncelikle bulanık sorgulama ve dilsel özetler, ardından veri tabanından bulanık kural çıkarımı üzerinde durulacak ve son olarak bulanık ve dereceli fonksiyonel bağlılıklar hakkında bilgi verilecektir.

4.1. Bulanık Sorgulama

Veri tabanlarından bilgi elde etmek amacıyla çeşitli sorgu cümlecikleri kullanılır. Örneğin, "Doğum yeri İstanbul olan kişileri göster" veya "Nüfus yoğunluğu 2 milyondan büyük olan şehirleri göster". Görüldüğü gibi klasik bir sorgulamada seçilen kısım mantıksal bir ifadedir. Ancak bulanık sorgulamada seçilen kısım bulanık bir ifadedir ve bulanık sorgulamalarda uzun, kısa, çok, az gibi dilsel kavramlar da kullanılabilmektedir. Örneğin lisede diploma notu yüksek olan öğrencileri görmek istersek oluşturmamız gereken sorgu "Diploma notu yüksek olan öğrencileri göster" şeklinde olacaktır.

VT bir veri tabanı ve o_i bu veri tabanı içinde bir nesne olsun $o_i \in VT$. $A = \{ a_1, a_2, a_3 ..\}$ ise bu veri tabanındaki alanların değerini göstersin. Örneğin nesne öğrenciler olurken, alanlar matematik puanı, okul türü, ÖSS puanı olabilir. Bu durumda $o_i.a_j$ notasyonu, o_i nesnesinin a_j alanındaki değerini ifade edecektir. Bununla birlikte her bir alan için üyelik fonksiyonları yardımıyla çeşitli bulanık kavramlar da atayabiliriz. Örneğin $\mu_{yüksek}$(diploma notu) üyelik fonksiyonu, yüksek kavramını diploma notu alanı üzerinde tanımlamaktadır.

Bulanık sorgulamanın sonucu, bulanık veri tabanındaki elemanlardan oluşan bir alt kümedir ve bu elemanların üyelik dereceleri sorgunun geçerlilik derecesini göstermektedir. Örnek olarak Tablo 4.1'de verilen veri tabanından yararlanalım.

Tablo 4.1. Veri Tabanının Bir Alt Kümesi

Ad	Diploma Notu (dipnot)	Matematik Ağırlıklı Puanı (Matagr)
Şennur	4.43	502.994
Okan	3.00	542.845
Metin	2.77	532.942
Hakan	4.62	584.755

Veri tabanı için tablonun yanı sıra,

VT = {(Şennur, 4.43, 502.994), (Okan, 3.00, 542.845), (Metin, 2.77, 532.942), (Hakan, 4.62, 584.755) }

şeklinde bir gösterimde kullanılabilir.

"Diploma notu yüksek öğrencileri göster" sorgusu şu şekilde gösterilir.

$$Q_{dipnot=yüksek}(VT) = \{ o_i \in VT \mid \mu_{yüksek}(o_i.Dipnot) \} \qquad (4.1)$$

Üyelik fonksiyonlarından S kullanılarak her bir nesneye üyelik derecesi atandıktan sonra sorgunun cevabı;

$Q_{dipnot=yüksek}(VT)$ = {1/(Hakan, 4.62, 584.755), 0.76/(Şennur, 4.43, 502.994), 0.13/(Ckan, 3.00, 542.845), 0.04/(Metin, 2.77, 532.942)}

Tablo 4.2 Diploma Notu Yüksek Olan İnsanların Üyelik Derecesi

Ad	Diploma Notu (dipnot)	Matematik Ağırlıklı Puanı (Matagr)	Üyelik Derecesi
Hakan	4.62	584.755	1
Şennur	4.43	502.994	0.76
Ckan	3.00	542.845	0.13
Metin	2.77	532.942	0.04

Normal veri tabanı sorgulamalarında olduğu gibi bulanık sorgulamalarda da "ve" ya da "veya" operatörlerini kullanabiliriz. Ancak burada bulanık mantık bölümünde gösterildiği gibi "ve" operatörü için üyelik derecelerinin minimumu, "veya" operatörü içinise üyelik derecelerinin maksimumu alınmalıdır. Örneğin "diploma notu yüksek ve matematik ağırlıklı puanı yüksek olan kişileri göster" sorgusu için bir önceki örnekte gösterilen adımları yinelersek;

$$Q_{dipnot=yüksek\ VE\ matagr=yüksek}(VT) = \{ o_i \in VT \mid \mu_{yüksek}(o_i.Dipnot) \wedge \mu_{yüksek}(o_i.Matagr)\}$$

$$= \{o_i \in VT \mid \min(\mu_{yüksek}(o_i.Dipnot), \mu_{yüksek}(o_i.Matagr))\}$$

$Q_{matagr=yüksek}(VT)$ = {1/(Hakan, 4.62, 584.755), 0.88/(Ckan, 3.00, 542.845), 0.80/(Metin, 2.77, 532.942), 0.42/(Şennur, 4.43, 502.994)}

$Q_{dipnot=yüksek\ VE\ matagr=yüksek}(VT)$ = {min (1/(Hakan, 4.62, 584.755), 0.76/(Şennur, 4.43, 502.994), 0.13/(Ckan, 3.00, 542.845), 0.04/(Metin, 2.77, 532.942)), (1/(Hakan, 4.62, 584.755), 0.88/(Ckan, 3.00, 542.845), 0.80/(Metin, 2.77, 532.942), 0.42/(Şennur, 4.43, 502.994))}

$Q_{dipnot=yüksek VE matagr=yüksek}(VT) = \{1/(Hakan, 4.62, 584.755), 0.42/(Şennur, 4.43, 502.994), 0.13/(Okan, 3.00, 542.845), 0.04/(Metin, 2.77, 532.942)\}$

Tablo 4.3. Diploma Notu Ve Matematik Puanı Yüksek Olanların Üyelik Derecesi

Ad	Diploma Notu (dipnot)	Matematik Ağırlıklı Puanı (Matagr)	Üyelik Dereci
Hakan	4.62	584.755	1
Şennur	4.43	502.994	0.42
Okan	3.00	542.845	0.13
Metin	2.77	532.942	0.04

4.2. Dilsel Eşikler

Genel olarak, kelimelerin başlarına ilave edilen önsıfatlarla onların anlamları biraz daha daraltılır veya genişletilir. Bu ifadeler arasında çok, oldukça, biraz, aşağı yukarı gibi birçok kelime bulunmaktadır. Dilsel eşik işlemleri, verilmiş olan bir küme için yapılır. Bu nedenle bulanık kümenin elemanlarında bir değişiklik olmaz; sadece bu elemanların üyelik dereceleri değişir. Zaten dilsel eşiklerin amacı yalnızca üyelik derecelerini değiştirmektir.

Genel olarak α şeklindeki bir kelimenin eşikleri aşağıdaki ifadelerden biri olur. α kelimesinin üyelik fonksiyonu $\mu_\alpha(x)/x$ olarak verilirse, eşiklerin matematiksel ifadeleri aşağıdaki gibi olacaktır.

"çok" $\alpha = \alpha^2 = \mu^2{}_\alpha(x)/x$; "çok, çok" $\alpha = \alpha^4$ "artı" $\alpha = \alpha^{1.25}$;

"eksi" $\alpha = \alpha^{0.75}$ "oldukça" $\alpha = \alpha^{0.5}$

Eşikler kullanılarak üyelik fonksiyonlarının değerleri daraltılabilir, genişletilebilir veya yoğunlaştırılabilir. Daraltma işlemi sonucunda bulanık kümede bulunan bütün elemanların üyelik derecelerinde azalma olur. Buradan α'nın 1'den büyük üslerinin alınması durumunda daraltma işleminin gerçekleştiğini söyleyebiliriz. Genişletme

işleminde ise α'nın 1'den küçük üsleri alınır – oldukça ve eksi örneklerinde olduğu gibi. Böylelikle verilen bulanık kümenin her elemanının üyelik derecesinde artma meydana gelir. Yoğunlaştırma işleminde ise bazı elemanların üyelik dereceleri düşerken, bazılarının üyelik dereceleri yükselmektedir. Genel olarak üyelik dereceleri 0.5'den büyük olanların üyelik dereceleri genişleyerek artarken, 0.5'den küçük olanların üyelik dereceleri daralarak azalmaktadır.

Bunların dışında kelimelere mantıksal bağlaçlar ve dilsel eşikler katarak daha karmaşık durumlar da elde edilebilir; ancak burada sıranın önemi vardır. Bir kümedeki elemanlar için uzun saçlı ve zeki kelimelerine ait üyelik dereceleri verilmiş olsun. O zaman uzun saçlı değil ve çok zeki ifadesini elde etmek için önce değil ve çok işlemleri yapılır ve sonra "ve" bağlacı ile elde edilen yeni üyelik fonksiyonları birleştirilir [15].

4.3. Dilsel Özet

Dilsel özetler, veri tabanındaki ilişkisel bilgileri ifade etmek için kullanılan verilerin bir tasviri olarak tanımlanabilir. Dilsel bir özete örnek olarak "Veri tabanındaki çoğu öğrencinin diploma notu yüksektir." veya "Veri tabanındaki çok az kişi üstün zekalıdır" verilebilir.

Genel olarak dilsel özetler "**VT**'deki **Q** nesneleri **S**'dir" formundadır. Buradaki S'ye özetleyici denilmektedir ve veri tabanındaki bazı alanlar üzerinde tanımlanmış olan bulanık bir ifade veya dilsel bir kavram olabilir. Q ise bağdaşabilme miktarıdır ve Zadeh'in dilsel niteleyici olarak tanımladığı sınıf bağlamına girmektedir. Çoğu, çok az veya biraz kelimeleri bu tür nesnelere örnek olarak verilebilir. Dilsel özet ile bağlantılı olan bir diğer kavram da özetin geçerlilik ölçüsü olarak isimlendirilen dilsel özetin doğruluk değeridir $\tau \in [0,1]$. Bir diğer ifadeyle τ, dilsel özetin veri tabanı ile ne kadar uyumlu olduğunu göstermektedir.

Dilsel özetin τ geçerliliğini şu şekilde hesaplayabiliriz:

$VT = \{ o_1, ..., o_n \}$ veri kayıtlarının kümesini; S, $o_i \in VT$ nesnelerinin A alanlarında tanımlanmış olan bulanık ifadeyi; Q ise bulanık niteleyiciyi göstersin.

1. Her $o_i \in$ VT için o_i'lerin S özetleyicisini sağlama derecesi $S(o_i)$ hesaplanır.

2. VT'de S'yi sağlayan nesnelerin oranı $r = \frac{1}{n}\sum_{i=1}^{n} S(o_i)$ ile bulunur.

3. Bulunan r değeri $\tau = Q(r)$ ifadesinde yerleştirilir yani r'nin önerilen miktardaki üyelik derecesi hesaplanır [12].

Tablo 4.1.'deki veriler için "Veri tabanındaki çoğu öğrencinin diploma notu yüksektir" dilsel özetinde bu metodolojiyi uygulayalım

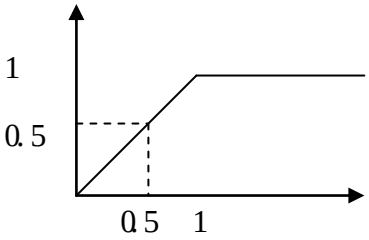
Adım 1: Diploma notu yüksek olan kişiler aranarak bulunur ve sonucu Tablo 4.2'de gösterilmektedir.

Adım 2: $r = \frac{1}{4}\sum_{i=1}^{4} \mu_{dipnot=yüksek}(o_i)$ (4.2)

$$r = \frac{(1 + 0.76 + 0.13 + 0.04)}{4} \Rightarrow r = 0.48$$

Adım 3: Bulunan r = 0.48 değeri $\tau = \mu_{çoğo}(r)$ ifadesinde yerine konur.

$\mu_{çoğo}$ bulanık fonksiyonunun yapısı aşağıdaki gibidir:



Şekil 4.1 Çoğu Bulanık Fonksiyonunun Yapısı

$\tau = \mu_{çoğo}(0.48) = 0.48$

Verilen dilsel ifadenin bu veri tabanındaki geçerlilik ölçüsü veya diğer bir ifadeyle doğruluk değeri 0.48'dir.

Bir başka örnek olarak "Veri tabanındaki çoğu matematik ağırlıklı puanı yüksek olan öğrencinin diploma notu da yüksektir." dilsel özetini alalım. Bu dilsel özette bir önceki örnekten farklı olarak veri tabanının tümü için değil, sadece belirli bir kısmı için (matematik ağırlıklı puanı yüksek olan öğrenciler) fikir ileri sürülmektedir ve genelde veri tabanından bulanık kural çıkarımında bu tür dilsel özetler daha kullanışlıdır. Bu türden dilsel özetlerin genel yapısı "**VT**'deki **QR** nesneleri **S**'dir"şeklindedir.

τ'nun hesaplamasında yukarıdaki yöntem 2. adım hariç geçerlidir. Bu kez r'yi S'yi sağlayan veri tabanındaki **R** nesnelerinin bir oranı olarak hesaplamalıyız.

$$r = \frac{\sum_{i=1}^{n} T(S(o_i), R(o_i))}{\sum_{i=1}^{n} R(o_i)} \qquad (4.3)$$

Burada R ve S özette kullanılan kavramları temsil eden bulanık alt kümelerdir. T ise minimum ve çarpım operatörlerinin dahil olduğu t-normdur ve minimum operatörü kullanılmaktadır [12].

Son dilsel özeti biraz değiştirerek yukarıdaki yöntemi uygulayalım. "Veri tabanındaki çoğu matematik ağırlıklı puanı yüksek olan öğrencinin diploma notu çok yüksektir."

Bu durumda öncelikli olarak diploma notu çok yüksek için yeni üyelik değerlerini bulmalıyız.

$\mu_{çok}(x) = \mu^2(x)$ bilindiğine göre

$\mu_{çokyüksekdipnot}(\text{Hakan}) = \mu^2{}_{yüksekdipnot}(\text{Hakan}) = 1,\ \mu_{çokyüksekdipnot}(\text{Şennur}) = (0.76)^2 = 0.57$

$\mu_{çokyüksekdipnot}(\text{Ckan}) = (0.13)^2 = 0.0169,\ \mu_{çokyüksekdipnot}(\text{Metin}) = (0.04)^2 = 0.0016$

$$r = \frac{\min(1,1) + \min(0.88,0.169) + \min(0.80,0.0016) + \min(0.42,0.57)}{1 + 0.88 + 0.80 + 0.42} = 0.46$$

$\tau = \mu_{çoğo}(0.46) = 0.46$

4.4. Bulanık Kurallar

"Veri tabanındaki matematik ağırlıklı puanı yüksek nesnelerin diploma notu yüksektir" şeklinde bulanık kuralının doğrulanması, "Veri tabanındaki çoğu matematik ağırlıklı puanı yüksek olan nesnelerin diploma notu yüksektir." dilsel özetinin gerçeklenmesine bağlıdır. Bu dilsel özet ise "Matematik ağırlıklı puanı yüksek olan öğrenciler içindeki çoğu nesnenin diploma notu yüksektir." ile aynı formülasyona sahiptir ve burada matematik puanı yüksek öğrenciler, orijinal veri tabanının yapısında bir bulanık veri tabanı oluştururlar; ancak tek farklılıkları matematik ağırlıklı puanı yüksek öğrencilerdeki her nesnenin bir üyelik derecesi vardır.

matagr yüksek öğrenciler$(o_i) = \mu_{matagr=yüksek}(o_i.Matagr)$.

Bulanık veri tabanında matematik ağırlıklı puanı yüksek öğrenciler nesnesi ve yüksek matematik ağırlıklı puanı aynı üyelik derecesine sahiptir. Bu eşitliği kullanarak, öncelikle veri tabanından matematik puanı yüksek öğrenciler bulanık kümesi seçilir (matagr yüksek öğrenciler = Q_{Matagr}(VT)) ve ardından bu yeni veri tabanına dilsel özet uygulanır.

$$\sum_{Çoğo} (o_i \in \text{matagr yüksek öğrenciler} \mid o_i.\text{Dipnot}=\text{yüksek})\ (\text{matagr yüksek öğrenciler})$$

$$= \sum_{Çoğo} (o_i \in \{o_j \in VT \mid \mu_{matagr}(o_j.Matagr)\} \mid \mu_{dipnot}(o_i.Dipnot)).$$

4.5. Bulanık ve Dereceli Fonksiyonel Bağlılıklar

Fonksiyonel bağlılık (FB) bir veri tabanının alanları üzerinde tanımlanan bir fonksiyondur ve

$$A \rightarrow B \Leftrightarrow \forall (o_i, o_j) \in VT,\ o_i.A = o_j.A \Rightarrow o_i.B = o_j.B \qquad (4.4)$$

şeklinde tanımlanabilir. Bir başka ifadeyle, B kümesindeki nesnelerin değerleri A kümesindeki nesneler tarafından tek bir şekilde belirlenmektedir. Örneğin TC kimlik numarası tek bir ismi belirlemektedir, TC Kimlik Numarası $\rightarrow$ İsim

İlişkisel veri tabanlarının 1970'lerde ortaya çıkmasından sonra fonksiyonlar bağlılıklar, hem gereksiz verilerin azaltılmasına yardımcı olan bir veri tabanı tasarım aracı olarak hem de veri tabanı yönetim sisteminin (VTYS) bütünlük kısıtı olarak önemli roller üstlenmiştir. 1980'lerin başından itibaren VTYS içerisine bulanık mantığı da katarak bulanık veri tabanı yönetim sistemleri (BVTYS) elde edilmiştir. BVTYS de normal VTYS gibidir; ancak kullanıcılarına çok, yaşlı, sıcak gibi bulanık kavramlar içeren sorgular sunmaktadır. BVTYS ile çalışmaya başlamanın bir sonucu olarak da bulanık fonksiyonel bağlılıklar (BFB) ortaya çıkmıştır. BFB gereksiz verilerin yok edilmesinden çok "Benzer müzik tarzını dinleyen kişiler benzer filmleri seyrederler" şeklindeki modelleme özelliklerine daha yakındır. Diğer bir ifadeyle BFB'lerin araştırılması, veri tabanı tasarımından çok veri tabanından bilgi çıkarılmasına daha yakındır.

Bir diğer fonksiyonel bağlılık çeşidi de dereceli fonksiyonel bağlılık (DFB) olarak adlandırılır. DFB de BFB gibi mevcut veri kümesinin bulanık bir özelliğidir. DFB'ye bir örnek "İnsanlar uykusuz kaldıkça algılama hızları yavaşlar" olabilir.

4.5.1. Bulanık Fonksiyonel Bağlılıklar

$A \rightarrow_F B$ ile gösterilen bulanık fonksiyonel bağlılık, veri tabanı üzerinde nesneler kümesi üzerinde belirli bir $\tau \in [0,1]$ derecesini almaktadır. Fonksiyonel bağlılıktan farklı olarak $o_i.A = o_j.A$ ifadesi yerine $o_i.A \approx o_j.A$ ifadesi kullanılır ve bu ifade [0,1] aralığında değer döndürür. Burada 1'in anlamı benzerdir. $o_i.A \approx o_j.A$ ilişkisinin değerlendirilmesi $o_i.A$ ve $o_j.A$ alanlarının benzerliklerin birleşimini gösterir ve $Agg(a_{i1} \approx a_{j1}, \ldots, a_{in} \approx a_{jn})$ fonksiyonu ile tanımlanmaktadır. Agg birleşim fonksiyonudur ve örneğin minimum operatörü olarak alınabilir.

BFB'de amaç, A ve B alanları arasında bulanık fonksiyonel bağlılıklar tanımlayabilmektir. Öncelikli olarak VT'deki her o_k nesnesine bakılarak o nesne için

BFB ne kadar gerçeklendiğini gösteren derece hesaplanmalıdır. Her o_k nesnesi aşağıdaki gibi bir bulanık kural (R_k) tanımlıyor olsun:

her $o_j \in$ VT için eğer $o_k.A \approx o_j.A$ ise $o_k.B \approx o_j.B$ (4.5)

dir.

Bu durumda her bulanık kural veri tabanına göre belirli bir doğruluk değerine sahip olacaktır ve doğruluk değeri de BFB'ın o_k nesnesi tarafından ne kadar sağlandığını göstermektedir. Bulanık kuralların her birinin doğruluk değerini hesaplamak için şu özeti kullanabiliriz: $\tau_k = \sum_{id}(o_i \in \{o_j \in VT | o_k.A \approx o_j.A\} | o_k.B \approx o_i.B)$. İfadenin ilk kısmında yer alan $\{o_j \in VT | o_k.A \approx o_j.A\}$ A'ya göre o_j ve o_k nesneleri benzer olan bulanık bir alt kümedir ve id bulanık niteleyicisi birim niteleyicidir: id(r)=r.

Her R_k bulanık kuralı için τ_k doğruluk değerleri bulunduktan sonra τ BFB değeri hesaplanır $A \rightarrow_F B$: "Eğer A için veri tabanındaki herhangi iki nesne benzer değerlere sahip ise, bu iki nesne B için de benzer değerlere sahip olacaktır." ifadesi bütün veri tabanında geçerlidir. Bu da her bir yerel bulanık kural için τ_k doğruluk değerlerinin özetlenmesi ile gerçekleştirilir ve sonuç olarak BFB için şu tanım elde edilir [11]:

$$\tau = \sum_{id}(o_k \in VT | \tau_k),$$

$$\tau = \sum_{id}(o_k \in VT | \sum_{id}(o_i \in \{o_j \in VT | o_k.A \approx o_j.A\} | o_k.B \approx o_i.B)). \quad (4.6)$$

Bu metodolojiyi kullanarak Tablo 4.1'de verilen örnek veri tabanımızda bulanık fonksiyonel bağlılıkları araştıralım. Bir diğer ifadeyle "Benzer diploma notuna sahip kişiler benzer matematik ağırlıklı puanına sahiptir" bulanık kuralının bulanık fonksiyonel bağımlılığını inceleyelim: {Diploma Notu} $\rightarrow_F$ {Matematik Ağırlıklı Puanı}

Hesaplamalara geçmeden önce $\approx$ fonksiyonunu ve T-normu tanımlamamız gerekmektedir. $A \rightarrow_F B$'ye her bir A ve B kümesinde, birden fazla alan varsa Agg birleşim fonksiyonunu tanımlamak gerekir. T-normu minimum olarak alalım: $a \approx b$ ise

diploma notu alanı için Max ((2-|a-b|)/2,0) şeklinde, matematik ağırlıklı puanı için ise Max((200-|a-b|)/200,0) olarak tanımlansın[11]. Burada 2 ve 200 bulanıklık operatörüdür ve iki sayının benzerliğini incelerken kullanılacak aralık adım büyüklüğünü belirler. Örneğin 1 ve 2 sayılarının benzerlikleri bulanıklık operatörü 2 iken 0.5, 200 iken 0.995 olacaktır. Bu nedenle bulanıklık operatörünün seçimi gerçeği yansıtan sonuçların elde edilmesi açısından oldukça önemlidir.

Öncelikle bulanık kurallar oluşturulmalıdır.

R_1 = Eğer 4.43 ≈ o_1.Dipnot ise 502.994 ≈ o_1.Matagr

R_2 = Eğer 3.00 ≈ o_1.Dipnot ise 542.845 ≈ o_1.Matagr

R_3 = Eğer 2.77 ≈ o_1.Dipnot ise 532.942 ≈ o_1.Matagr

R_4 = Eğer 4.62 ≈ o_1.Dipnot ise 585.755 ≈ o_1.Matagr

Her kural için doğruluk değeri aşağıdaki gibi hesaplanır.

4.43 ≈ o_1.Dipnot için 4.43 ≈ 4.43 = Max ((2 - |4.43 - 4.43|) / 2, 0) = 1

4.43 ≈ 3.00 = Max ((2- |4.43 - 3.00|) / 2, 0) = 0.2

4.43 ≈ 2.77 = Max ((2- |4.43 - 2.77|) / 2, 0) = 0.17

4.43 ≈ 4.62 = Max ((2- |4.43 - 4.62|) / 2, 0) = 0.90

Benzer şekilde 3.00 ≈ o_1.Dipnot, 2.77 ≈ o_1.Dipnot ve 4.62 ≈ o_1.Dipnot için de hesaplamalar yapılır.

502.994 ≈ o_1.Matagr için 502.994≈502.994= Max((200-|502.994–502.994|)/200,0) = 1

502.994≈542.845= Max((200-|502.994–542.845|)/200,0)=0.80

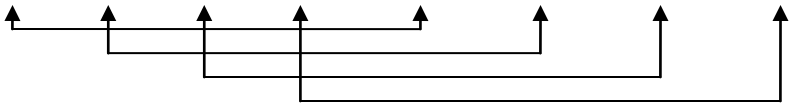
502.994≈532.942= Max((200-|502.994–532.942|)/200,0)=0.85

502.994≈584.755= Max((200-|502.994–584.755|)/200,0)=0.58

Tablo 4.4.'de bulunan bütün sonuçlar gösterilmektedir.

Tablo 4.4 Bulanık Fonksiyonel Bağlılık Ara Tablosu

Nesne	Diploma not	Matagr	4.43 ≈ o. Dip	3.00 ≈ o. Dip	2.77 ≈ o. Dip	4.62 ≈ o. Dip	502.994 ≈ o. Matagr	542.845 ≈ o. Matagr	532.942 ≈ o. Matagr	585.755 ≈ o. Matagr
Şennur	4.43	502.994	1	0.28	0.17	0.90	1	0.80	0.85	0.58
Okan	3.00	542.845	0.28	1	0.88	0.19	0.80	1	0.95	0.79
Metin	2.77	532.942	0.17	0.88	1	0.075	0.85	0.95	1	0.74
Hakan	4.62	585.755	0.90	0.19	0.075	1	0.58	0.79	0.74	1



$$\tau_1 = \frac{\min(1,1) + \min(0.28,0.80) + \min(0.17,0.85) + \min(0.90 + 0.58)}{1 + 0.28 + 0.17 + 0.90} = 0.86$$

$$\tau_2 = \frac{\min(0.28,0.80) + \min(1,1) + \min(0.88,0.95) + \min(0.19 + 0.79)}{0.28 + 1 + 0.88 + 0.19} = 1$$

$$\tau_3 = \frac{\min(0.17,0.85) + \min(0.88,0.95) + \min(1,1) + \min(0.075 + 0.74)}{0.17 + 0.88 + 1 + 0.075} = 1$$

$$\tau_4 = \frac{\min(0.90,0.58) + \min(0.19,0.79) + \min(0.075,0.74) + \min(1+1)}{0.90 + 0.19 + 0.075 + 1} = 0.85$$

$$\tau = \frac{\tau_1 + \tau_2 + \tau_3 + \tau_4}{4} = 0.92$$

Böylelikle $\{\text{Diploma Notu}\} \rightarrow_F \{\text{Matematik Ağırlıklı Puanı}\}$=0.92 olarak bulunmuş olur.

4.5.2. Dereceli Fonksiyonel Bağlılık

Dereceli fonksiyonel bağlılık veri tabanındaki nesneler arasında ilişkileri ve veri deki monotonluğu ifade etmektedir. FFD' ye benzer şekilde DFB de bir [0,1] aralığında τ derecesi almaktadır.

DFB' ye örnek olarak;

"Bir öğrenci daha yüksek diploma notuna sahip ise matematik ağırlıklı puanı da daha yüksektir."

verilebilir.

DFB' deki monotonluk dereceli terimlerle ortaya çıkmaktadır. "Daha yüksek, daha çalışkan, daha başarılı" gibi artan dereceli terimler kullanılabileceği gibi "daha az, daha başarısız gibi azalan dereceli terimler de kullanılabilir. Genel DFB yapısı şu şekildedir:

"Veri tabanındaki nesneler ne kadar G_1 ise o kadar G_2 'dir."

Burada G_1 ve G_2 daha çalışkan, daha başarılı gibi dereceli ifadelerdir.

o_k nesnesi için, veri tabanında yer alan ve kendisinden büyük değerlerin o_j nesnesinde de büyük değerlere sahip olması, kendisinden küçük olanların o_j nesnesinin de kendisinden küçük olduğu koşuluna uyan τ_k değerleri bulunur. DFB' nin τ_k doğruluk derecesinin nasıl ölçüleceğini yukarıdaki örnekten faydalanarak açıklayalım. Lise notu kendisinden büyük olan lise notlarına karşı gelen matematik ağırlıklı puanlarının kendisininkinden büyük olduğu ve kendisinden küçük lise notlarının matematik ağırlıklı puanının kendisininkinden küçük olduğu koşulunu sağlayan τ_k değerleri bulunur. Bütün nesneler için bu τ_k değerleri hesaplandıktan sonra DFB' nin geçerliliğini belirtmek için τ hesaplanır [11].

Her bir τ_k' şu özet kullanılarak bulunabilir:

$$\tau_k = \sum_{id} (o_i \in VT \mid G_1(o_k, o_j) \wedge G_2(o_k, o_j)) \vee (G_1^{'}(o_k, o_j), G_2^{'}(o_k, o_j)) \qquad (4.7a)$$

Buradaki $G_1(o_k, q_j)$ ve $G'_1(o_k, q_j)$ dereceli ifadeler olan G_1 ve G'_1'nin dönüşümleridir. Böylelikle bu terimler veri tabanındaki nesnelerle ilişkilendirilebilmektedir. Örneğin,

G_1 = "daha yüksek diploma notu" ifadesini,

$G_1(o_k, q_j)$ = 'o_k.dipnot < q_j.dipnot' ve $G'_1(o_k, q_j)$ = 'o_k.dipnot > q_j.dipnot'

ile gösterebiliriz. Bu dönüşüm kullanılarak

$$\tau_k = \sum_{id} o_i \in VT \mid (o_k.dipnot < o_j.dipnot \wedge o_k.matagr < o_j.matagr)$$

$$\vee (o_k.dipnot > o_j.dipnot \wedge o_k.matagr > o_j.matagr) \qquad (4.7b)$$

elde edilebilir.

Dereceli terimin azalması durumunda bu ifadelerdeki küçüktür işaretini büyüktür şeklinde değiştirmemiz gerekmektedir. Yani "daha düşük diploma notu" ifadesi şu şekilde gösterilecektir.

$G_1(o_k, q_j)$ = 'o_k.dipnot > q_j.dipnot' ve $G'_1(o_k, q_j)$ = 'o_k.dipnot < q_j.dipnot'

Eğer G_i ifadesinde birden fazla dereceli terim bulunuyorsa örneğin "daha yüksek diploma notu ve daha düşük matematik puanına sahip" benzer mantıkla,

'o_k.dipnot < q_j.dipnot ve o_k.matagr > q_j.matagr'

şeklinde ifade edilmelidir.

İlişkiler yani G'ler binary yani sadece bir (doğru) ve sıfır (yanlış) değerlerini almaktadır. Büyüktür veya benzerdir "$>\approx$" ve küçüktür veya benzerdir "$<\approx$" binary bulanık ilişkileri kullanılarak DFB genelleştirilebilir ve birim aralıkta bir değer döndürür. Bu ilişkileri, büyüktür ">" ve küçüktür "<" ilişkileri ile benzerlik fonksiyonu "$\approx$" olarak iki parçada ele alabiliriz. "$x <\approx y$" ilişkisi örneğin Max(Max((2-|x-y|)/2, 0), x<y) şeklinde tanımlanabilir. Maksimum fonksiyonundaki 2 bulanık fonksiyonel bağlılıkta olduğu gibi bulanıklık operatörüdür.

$$\tau_k = \sum_{id}(o_i \in VT \mid G_1(o_k, o_j) \wedge G_2(o_k, o_j)) \vee (G_1^{'}(o_k, o_j), G_2^{'}(o_k, o_j)) \quad (4.8a)$$

ve 4.7b'ye benzer şekilde aşağıdaki denklemelde edilir:

$$\tau_k = \sum_{id} o_i \in VT \mid (o_k.dipnot < o_j.dipnot \wedge o_k.matagr <\approx o_j.matagr)$$

$$\vee (o_k.dipnot > o_j.dipnot \wedge o_k.matagr >\approx o_j.matagr) \quad (4.8b)$$

τ_k'ların hesaplanmasında ifadenin ilk kısmında birden fazla belirleyicinin olduğu durum önemlidir. Örneğin "Diploma notu daha yüksekse ve matematik ağırlıklı puanı daha yüksekse, kazanılan bölümün tercih sırası daha düşüktür." kuralını ele alalım.

İlk kısmında yer alan iki nesne için dört tane alternatif bulunmaktadır.

1. o_k.dipnot > o_q.dipnot ve o_k.matagr > o_q.matagr
2. o_k.dipnot < o_q.dipnot ve o_k.matagr < o_q.matagr
3. o_k.dipnot > o_q.dipnot ve o_k.matagr < o_q.matagr
4. o_k.dipnot < o_q.dipnot ve o_k.matagr > o_q.matagr

Bu durumda sadece 1 ve 2 olasılıkları kuralın doğruluğunu ispatlamakta kullanılmaktadır; 3 ve 4 ise kuralla ilişkili değildir. τ değeri veri tabanındaki bütün nesneler üzerinden hesaplandığından $G_1(o_k, o_q)$ veya $G'_1(o_k, o_q)$ durumlarını sağlamayan nesneleri hesaba katmamalıyız. Sadece $G_1(o_k, o_q)$ veya $G'_1(o_k, o_q)$'i sağlayan nesneler için τ_k ise şu şekilde hesaplanmalıdır:

$$\tau_k = \sum_{id}(o_j \in \{o_i \in VT \mid G_1(o_k, o_j) \wedge G_1^{'}(o_k, o_j)\} \mid (G_1(o_k, o_j), G_2(o_k, o_j))$$

$$\vee (G_1^{'}(o_k, o_j), G_2^{'}(o_k, o_j)) \quad (4.9)$$

Veri tabanındaki her bir o_k için τ_k bulunduktan sonra DFB'nin geçerliliğini hesaplamada (τ) kullanabiliriz:

$$\tau = \sum_{id}(o_k \in VT | \tau_k)$$

$$\tau = \sum_{id}(o_k \in VT \mid \sum_{id}(o_j \in \{o_i \in VT \mid G_1(o_k, o_j) \wedge G_1^{'}(o_k, o_j)\}|$$

$$(G_1(o_k, o_j), G_2(o_k, o_j)) \vee (G_1^{'}(o_k, o_j), G_2^{'}(o_k, o_j)) \qquad (4.10)$$

Bu durumda daha çalışkan olanların matematik ağırlıklı puanı daha yüksektir kuralı için doğruluk değeri

$$\tau = \sum_{id}(o_k \in VT \mid \sum_{id} o_i \in \{o_i \in VT \mid o_k.dipnot < o_j.dipnot \vee o_k.dipnot > o_j.dipnot \}|$$

$$(o_k.dipnot < o_j.dipnot \wedge o_k.matagr <\approx o_j.matagr)$$

$$\vee (o_k.dipnot > o_j.dipnot \wedge o_k.matagr >\approx o_j.matagr) \qquad (4.11)$$

Tablo 4.1'de verilen veri tabanı yardımıyla (4.11) denkleminin sayısal değerini hesaplayabiliriz.

Öncelikle bulanık ilişleri hesaplamamız gerekmektedir.

x<≈y ifadesini hesaplamak için Max(Max(200-|x-y|)/200, 0), x<y);

x>≈y ifadesini hesaplamak için ise Max(Max(200-|x-y|)/200, 0), x>y) ifadesinden yararlanabiliriz.

502.994 <≈ o9.Matagr için

502.994<≈502.994=Max(Max((200-|502.994–502.994|)/200, 0), 502.994<502.994) = 1

502.994<≈542.845=Max(Max((200-|502.994–542.845|)/200, 0), 502.994<542.8450) = 1

502.994<≈532.942=Max(Max((200-|502.994–532.942|)/200, 0), 502.994<532.942) = 1

502.994<≈584.755=Max(Max((200-|502.994–584.755|)/200, 0), 502.994<584.755) = 1

Benzer şekilde 502.994 >≈ ç. Mütagr için

502.994>≈502.994=Max(Max((200-|502.994–502.994|)/200,0),502.994>502.994) = 1

502.994>≈542.845=Max(Max((200-|502.994–542.845|)/200,0),502.994>542.8450) =.8

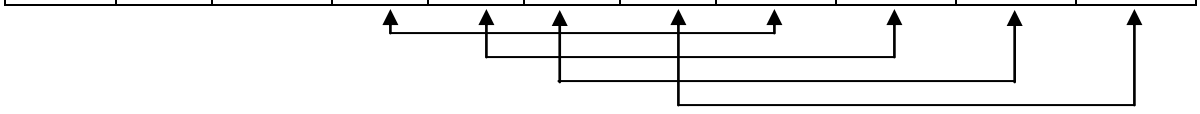
502.994>≈532.942=Max(Max((200-|502.994–532.942|)/200,0),502.994>532.942) =.85

502.994>≈584.755=Max(Max((200-|502.994–584.755|)/200,0),502.994>584.755) =.58

Diğerleri için de aynı değerler hesaplandıktan sonraki değerler aşağıdaki tabloda gösterilmektedir.

Tablo 4.5. Dereceli Fonksiyonel Bağlılıklar

Nesne	Dip.not	Mütagr	4.43 < ç. Dip veya 4.43> ç. Dip	3.00 < ç. Dip veya 3.00> ç. Dip	2.77 < ç. Dip veya 2.77> ç. Dip	4.62 < ç. Dip veya 462> ç. Dip	(4.43< ç. Dip ve 502.994 <≈ ç. Mütagr) veya (4.43>ç. Dip ve 502.994 >≈ ç. Mütagr)	(3.00< ç. Dip ve 542.845 <≈ ç. Mütagr) veya (3.00>ç. Dip ve 542.845 >≈ ç. Mütagr)	(2.77< ç. Dip ve 532.994 <≈ ç. Mütagr) veya (2.77>ç. Dip ve 532.994 >≈ ç. Mütagr)	(4.62 ç. Dip ve 585.755 <≈ ç. Mütagr) veya (4.62>ç. Dip ve 585.755 >≈ ç. Mütagr)
Şennur	4.43	502.994	0	1	1	1	0	0.80	0.85	1
Okan	3.00	542.845	1	0	1	1	0.80	0	1	1
Metin	2.77	532.942	1	1	0	1	0.85	1	0	1
Hakan	4.62	585.755	1	1	1	0	1	1	1	0



$\tau_1 = [\min(0,0)+\min(1,0.80)+\min(1,0.85)+\min(1,1)]/(0+1+1+1) = 0.88$

$\tau_2 = [\min(1,0)+\min(0,0)+\min(1,1)+\min(1,1)]/(1+0+1+1) = 1$

$\tau_3 = [\min(1, 0.85) + \min(1, 1) + \min(0, 0) + \min(1, 1)]/(1+1+0+1) = 0.95$

$\tau_4 = [\min(1, 1) + \min(1, 1) + \min(0, 1) + \min(0, 0)]/(1+1+1+0) = 1$

$\tau = (0.88+1+0.95+1)/4$

$\tau = 0.95$

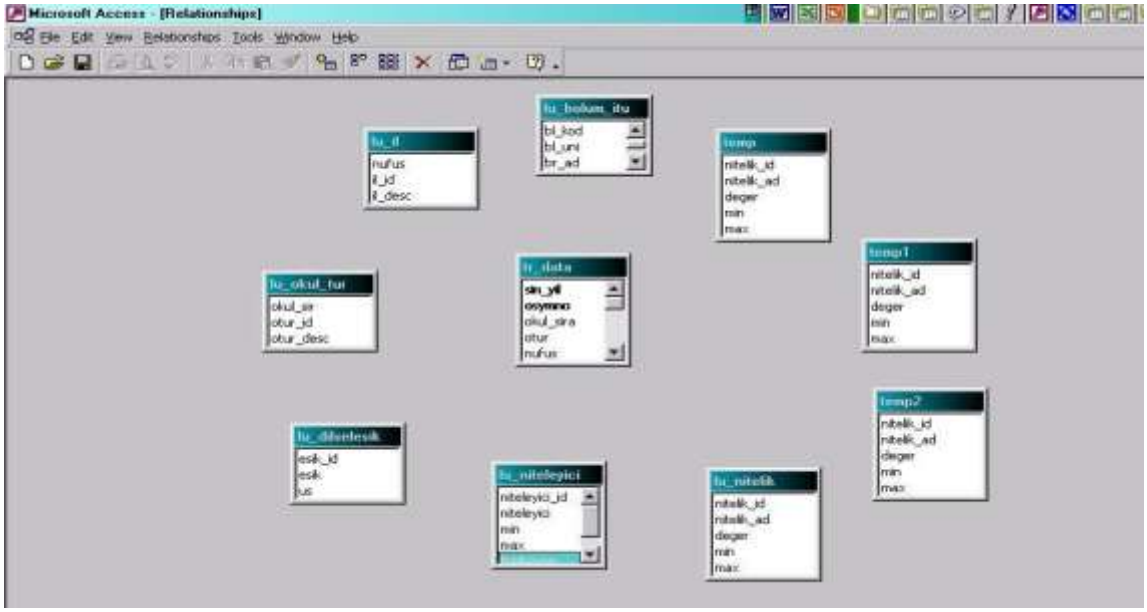
Daha yüksek diploma notuna sahip olan öğrencilerin daha yüksek matematik ağırlıklı puanına sahip olacağı yönündeki dereceli fonksiyonel bağlılığın doğruluk değeri 0.95 olarak bulunur.

5. GELİŞTİRİLEN UYGULAMANIN YAPISI

Uygulama, Windows platformunda MS Access ve genel olarak ASP kullanılarak geliştirilmiştir. Programlama dili olarak ASP' nin tercih edilme nedeni, uygulamanın web ortamından erişimine olanak sağlayarak, İstanbul Teknik Üniversitesine kayıt yapmaya hak kazanan kişilerin profillerini merak edenlere kolaylık sağlamaktır.

5.1. Veri tabanı

Veri kaynağı olarak 1995-1997 yılları arasında İstanbul Teknik Üniversitesinde okumaya hak kazanan 6942 kişinin bilgilerinin tutulduğu veri tabanından yararlanılmıştır. Uygulamaya hazırlık olarak öncelikle veri temizliği yapılmıştır. Bulanık mantık kullanımını kolaylaştırmak için yeni tablolar oluşturulmuştur. Veri tabanında bulunan tablolar Şekil 5.1' de gösterilmektedir.



Şekil 5.1 Veri Tabanında Bulunan Tablolar

Tablo 5.1, kullanılan tabloların bulanık mantıkta yararlanılan kısımlarını göstermektedir. Bazı tablolara veri hazırlık aşamasında farklı alanlar eklenmiş olmasına rağmen bu alanlar gelecekte olabilecek değişik isteklerin gerçekleştirilmesinde kullanılmak üzere saklanmıştır. Ana tabloların dışında temp, temp1 ve temp2 gibi hesaplamalar sırasında geçici olarak kullanılan tablolar da bulunmaktadır. İçerikleri yapılan hesaplamaya göre değişiklik göstermektedir.

Tablo 5.1 Veri Tabanında Bulunan Tablolar ve İçerikleri

tr_data	Ana kayıt tablosu, sorgulama yapılabilecek tüm alanlar için kişi bazında veriler
lu_nitelik	Sorgulama yapılabilecek tüm alanların isimleri, kodu, alabilecekleri en az ve en çok değerler ile sözel değerler
lu_niteleyici	Dilsel özetlemede kullanılan niteleyiciler, kodları ve alabilecekleri en az ve en çok değerler
lu_dilselesik	Dilsel eşik ismi, kodu ve üs değeri
lu_bolum_itu	Kazanılan bölümün adı, kodu ve bölüm sıralaması
lu_okul_tur	Kişinin bitirdiği okul ismi, kodu ve okul sıralaması
lu_il	Kişinin sınava girerken kayıtlı olduğu ilin ismi, kodu ve nüfus bilgisi

5.2 Kullanıcı Ara Yüzü

Geliştirilen uygulama internet ortamından erişilebilecek şekilde tasarlanmasına rağmen, herhangi bir alan satın alınmadığı için, kurulu olan IIS' ten lokal olarak faydalanılmaktadır. http://localhost/tez/default.asp adresinden uygulamaya giriş yapabilirler.

Kullanıcının karşısına çıkan ilk sayfa ana sayfadır ve Şekil 5.2' de gösterilmektedir. Ana sayfada kullanıcının gerçekleştirebileceği tüm uygulamaların birer bağlantısı

bulunmaktadır. İstenilen uygulamanın bağlantısı üzerine tıklamak suretiyle ilgili sayfalar arasında geçiş yapılabilir.

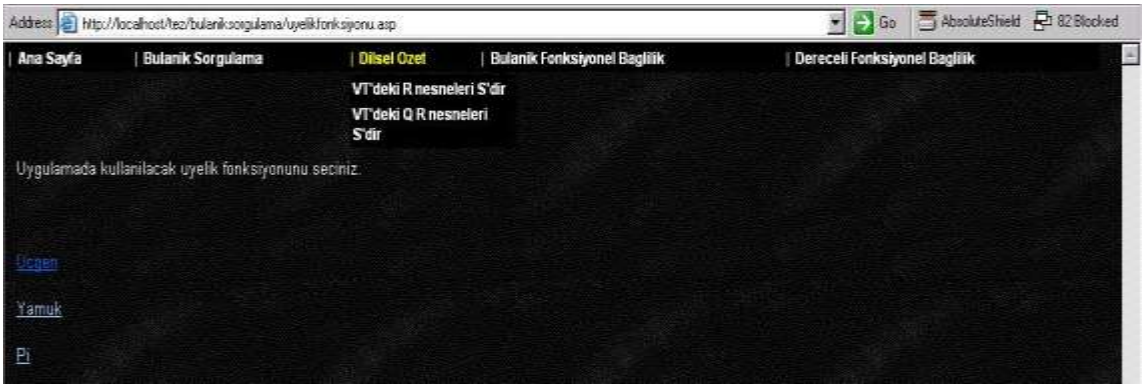
Kullanıcı ara yüzünün tam olarak anlaşılması için her uygulama sırasıyla incelenecektir.

Şekil 5.2 Ana Sayfa

Ana sayfa dışında tüm sayfaların üst kısımlarında bir menü bulunmaktadır. Menüde ana sayfa, tüm uygulamalar ve bu uygulamaların alt başlıkları yer almaktadır. Her uygulamanın alt başlığı, imleç üstüne getirildiğinde gözükecek şekilde, diğer bir ifadeyle dinamik olarak tasarlanmıştır. Dinamik Menüde Javascript dilinden yararlanılmıştır. Kullanıcı, internet gezgininin ileri, geri düğmelerini kullanarak uygulamalar arasında ve uygulama içinde geçişler yapabileceğinden aynı işlevde yeni düğmeler yaratılmamıştır.

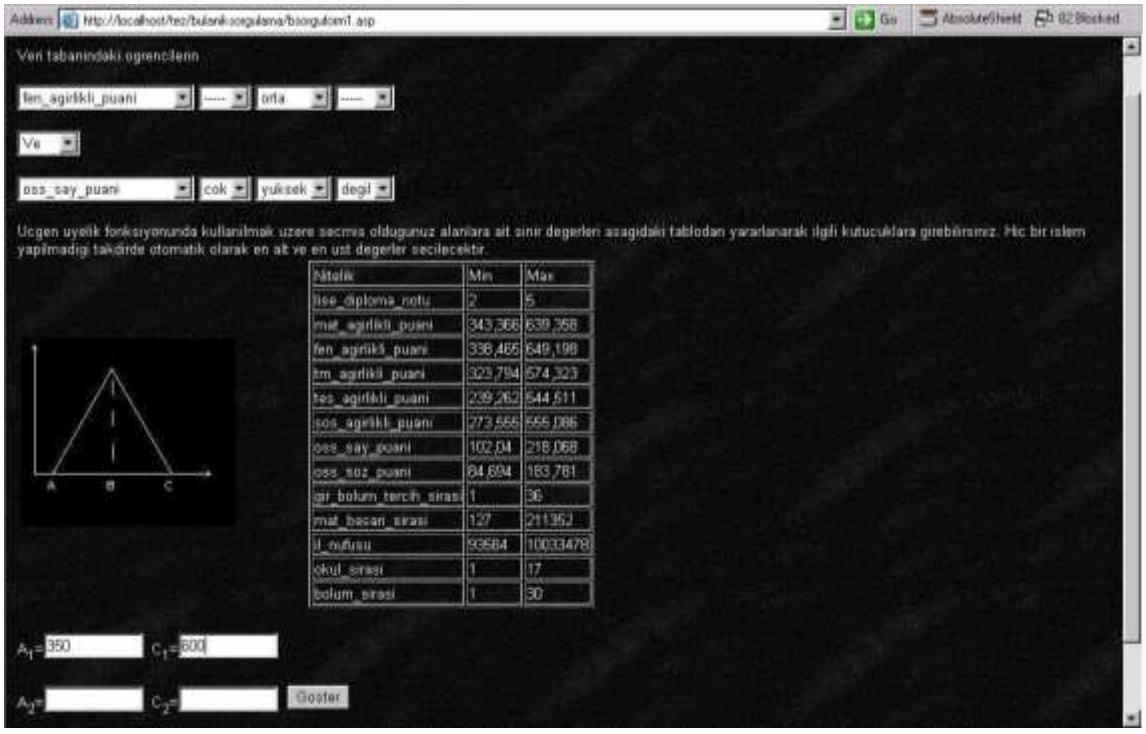
5.2.1. Bulanık Sorgulama

Bulanık sorgulama yapmak isteyen kullanıcının karşısına ilk olarak, uygulamada hangi üyelik fonksiyonunu kullanmak istediğini soran bir sayfa gelmektedir.



Şekil 5.3 Üyelik Fonksiyonu Seçim Sayfası

Üçgen, yamuk veya pi üyelik fonksiyonlarından istediğini seçen kullanıcı, sorgulamanın gerçekleştirileceği sayfaya yönlendirilmektedir.



Nitelik	Min	Max
lise_diploma_notu	2	5
mat_agirlikli_puani	343,366	639,358
fen_agirlikli_puani	336,465	649,198
tm_agirlikli_puani	323,794	674,323
tes_agirlikli_puani	239,262	644,611
sos_agirlikli_puani	273,555	555,086
oss_say_puani	102,04	218,068
oss_soz_puani	84,694	183,761
gr_bolum_tercih_sirasi	1	36
mat_basari_sirasi	127	211352
il_nufusu	93684	10033478
okul_sirasi	1	17
bolum_sirasi	1	30

Şekil 5.4 Bulanık Sorgulama Temel Sayfası

Bulanık sorgu cümleleri, Tablo 5.1'de tanımlanan alanlar üzerinde gerçekleştirilebilmektedir. Kullanıcı sorgulama yapmak istediği alanları, aşağı ok düğmesine basınca genişleyen bir listeden seçmektedir. Seçilen alanlar için çok,

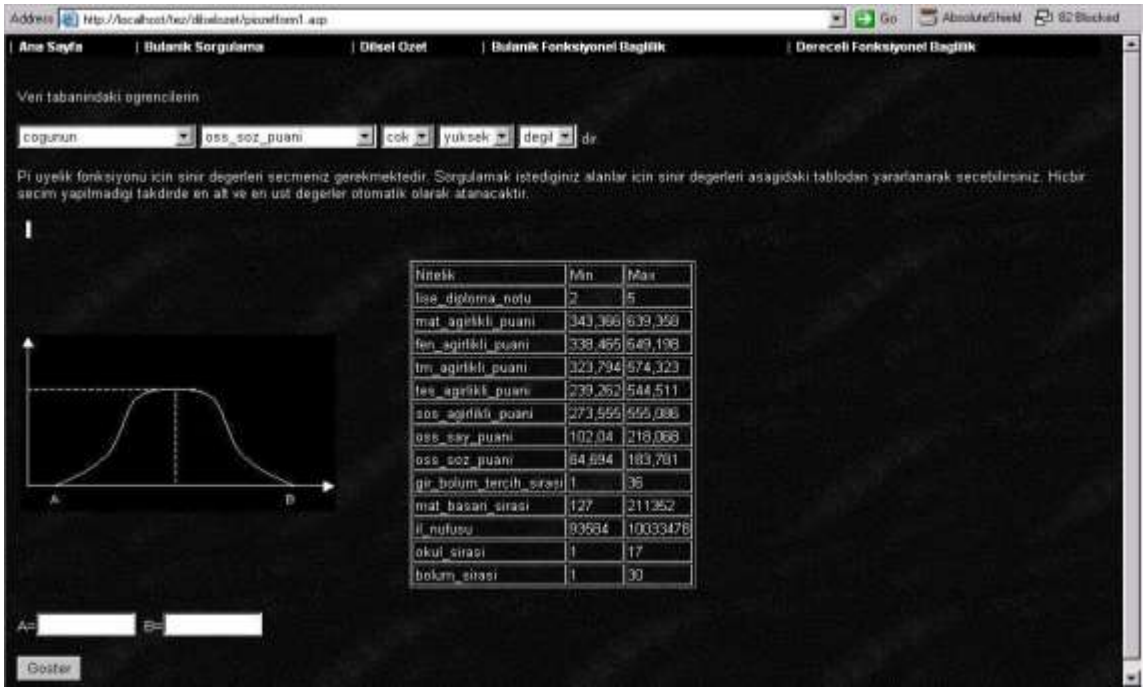
oldukça şeklinde dilsel eşikler kullanılabileceği gibi sorgunun olumsuzu da oluşturulabilmektedir. İstenirse tek bir sorgu cümlesi kurulabilir; ancak istendiğinde sorgu cümleleri "ve" ya da "veya" bağlaçları ile birleştirilebilmektedir. Gerek seçilen üyelik fonksiyonuna, gerekse kullanılan bağlacın çeşidine göre elde edilen sonuçların farklılık göstereceği unutulmamalıdır.

Daha önceden belirtildiği gibi, sorgulama yapılabilen alanların en az ve en çok değerleri veri tabanında saklanmaktadır. Bu değerler uygulama sayfasında tablo halinde kullanıcıya gösterilmektedir. Bu verilerin kullanıcıya gösterilmesindeki amaç, seçilen üyelik fonksiyonunda mevcut sınır değerleri kullanmak yerine kullanıcının istediği sınır değerleri seçme hakkını tanımaktır. Hiçbir seçim yapılmazsa sınır değerler otomatik olarak seçilmiş kabul edilerek hesaplamalar gerçekleştirilmektedir. Kullanıcının verileri elle girdiği her ortamda hatanın olabileceği göz önünde bulundurularak yanlış veri girişleri koyulan kriterlerle engellenmeye çalışılmaktadır.

Yapılan bulanık sorgulamanın sonucu bir sonraki sayfada tablo halinde gösterilmektedir.

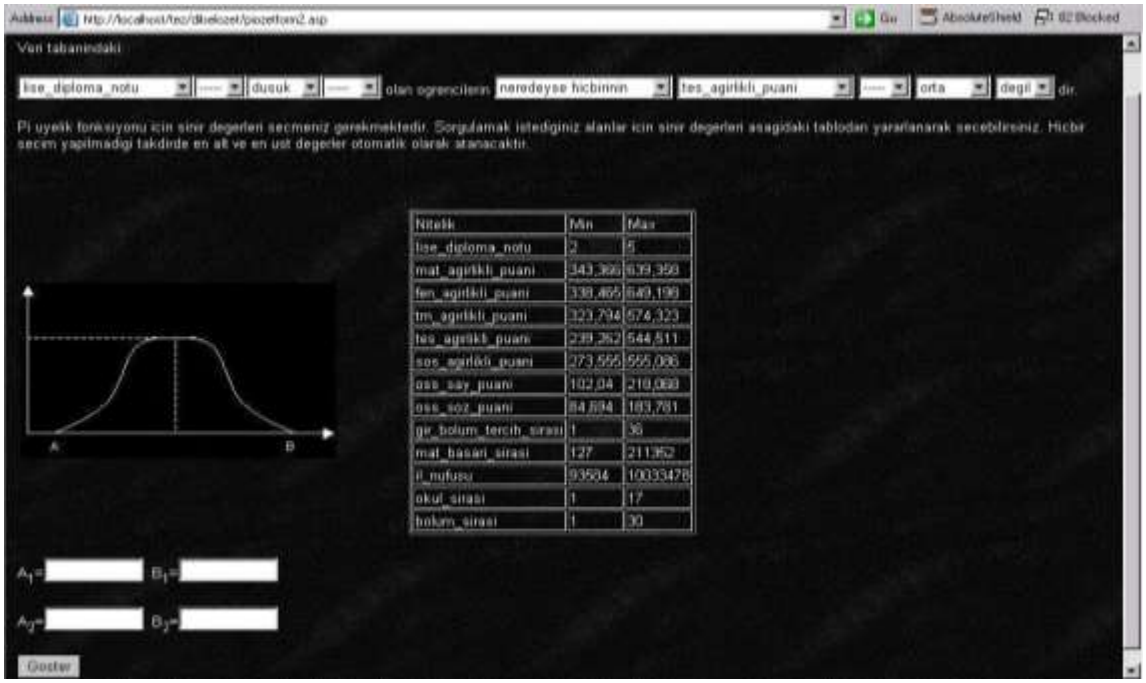
5.2.2. Dilsel Özet

Bir başka uygulama konusu da dilsel özettir. Dilsel özet uygulaması Bölüm 4.3'te anlatıldığı gibi iki ana başlık altında gerçekleşmektedir: "VT' deki **Q** nesneleri **S**' dir" ve "VT' deki **Q R** nesneleri **S**' dir". Şekil 5.5'te gösterilen "Veri tabanındaki öğrencilerin hemen hemen hepsinin ÖSS sözel puanı çok yüksek değildir" dilsel özeti, "VT' deki **Q** nesneleri **S**' dir" yapısındadır. Burada S özetleyicisi yüksek, kullanılan dilsel eşik çok ve niteleyici, çoğunundur.



Şekil 5.5 VT' deki Q Nesneleri S' dir Şeklindeki Dilsel Özet Sayfası

Bulanık sorgulamada olduğu gibi kullanıcıya üyelik fonksiyonu ve sınır değerler seçtirilmektedir. Şekil 5.6 ise "VT' deki **Q R** Nesneleri **S**' dir" formundaki dilsel özete örnek oluşturmaktadır. Bu örnekte kullanıcı, lise notu düşük olan öğrencilerden neredeyse hiçbirinin eşit ağırlıklı puanın orta olmadığı görüşünü sınamaktadır. Dikkat edilecek olursa kullanıcı öncelikle veri tabanından lise notu düşük olan öğrencilerden oluşan yeni bir alt küme oluşturmakta, ardından bir önceki örnektekine benzer şekilde özeti gerçekleştirmektedir.

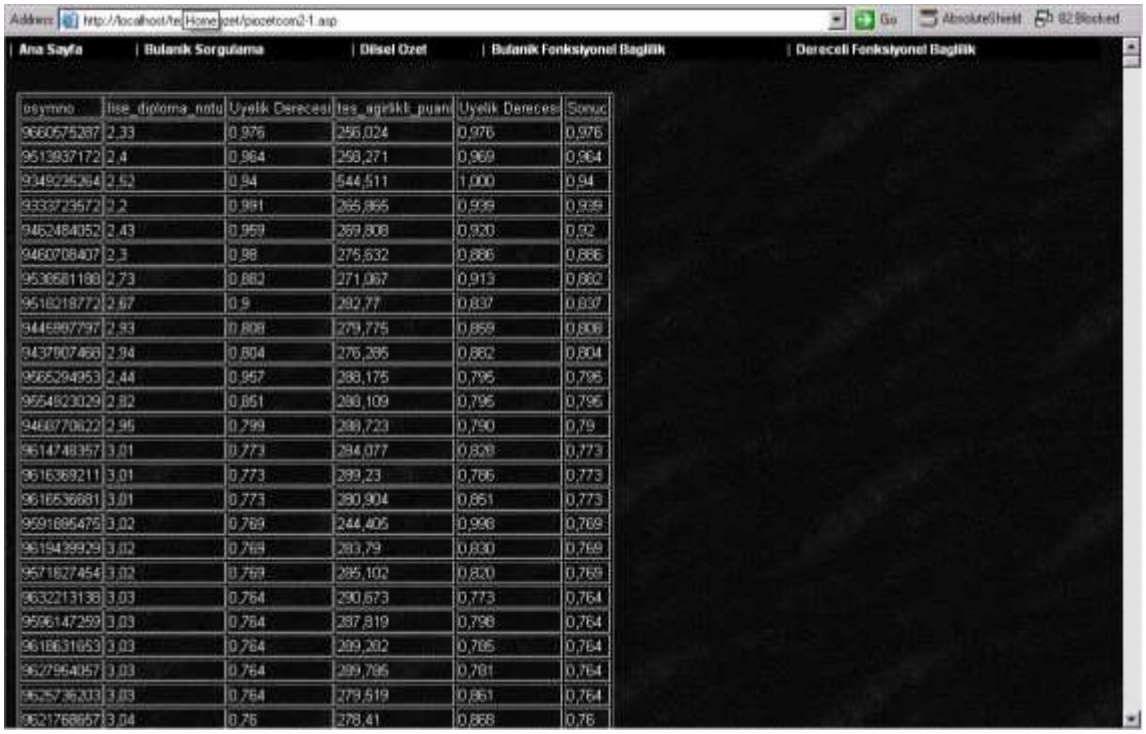


Şekil 5.6 VT' deki QR Nesneleri S' dir Şeklindeki Dilsel Özet Sayfası

Sınır değerlerin atanmasının ardından özetin geçerlilik ölçüsü olan doğruluk değeri hesaplanmaktadır. Uygulamada doğruluk değeri hesaplanıp yeni bir sayfada gösterilmekte ve bu sayfada kullanıcıya ayrıntıları görme hakkı tanınmaktadır.

Şekil 5.7 Dilsel Özetin Doğruluk Değeri

Ayrıntı görmek isteyen kullanıcıya Şekil 5.8'de görüldüğü gibi her bir alanın aldığı üyelik derecesi ve sonuç tablo halinde sunulmaktadır.



osymno	lise_diploma_notu	Uyelik Derecesi	tes_agirlikli_puani	Uyelik Derecesi	Sonuc
9660575287	2,33	0,976	256,024	0,976	0,976
9513937172	2,4	0,964	259,271	0,969	0,964
9349236264	2,52	0,94	544,511	1,000	0,94
9333723572	2,2	0,991	265,865	0,939	0,939
9462484052	2,43	0,959	269,808	0,920	0,92
9460708407	2,3	0,98	275,632	0,886	0,886
9539581188	2,73	0,882	271,067	0,913	0,882
9518018772	2,67	0,9	282,77	0,837	0,837
9445897797	2,93	0,808	279,775	0,859	0,808
9437907468	2,94	0,804	276,395	0,882	0,804
9565294953	2,44	0,957	288,175	0,795	0,795
9554823029	2,82	0,851	288,109	0,795	0,795
9468770822	2,95	0,799	288,723	0,790	0,79
9614748357	3,01	0,773	284,077	0,828	0,773
9616369211	3,01	0,773	289,23	0,786	0,773
9616536881	3,01	0,773	280,904	0,851	0,773
9591885475	3,02	0,769	244,405	0,996	0,769
9619439929	3,02	0,769	283,79	0,830	0,769
9571827454	3,02	0,769	285,102	0,820	0,769
9632213138	3,03	0,764	290,573	0,773	0,764
9596147259	3,03	0,764	287,819	0,798	0,764
9618631953	3,03	0,764	289,282	0,785	0,764
9627964057	3,03	0,764	289,795	0,781	0,764
9625736203	3,03	0,764	279,519	0,861	0,764
9621768857	3,04	0,76	278,41	0,868	0,76

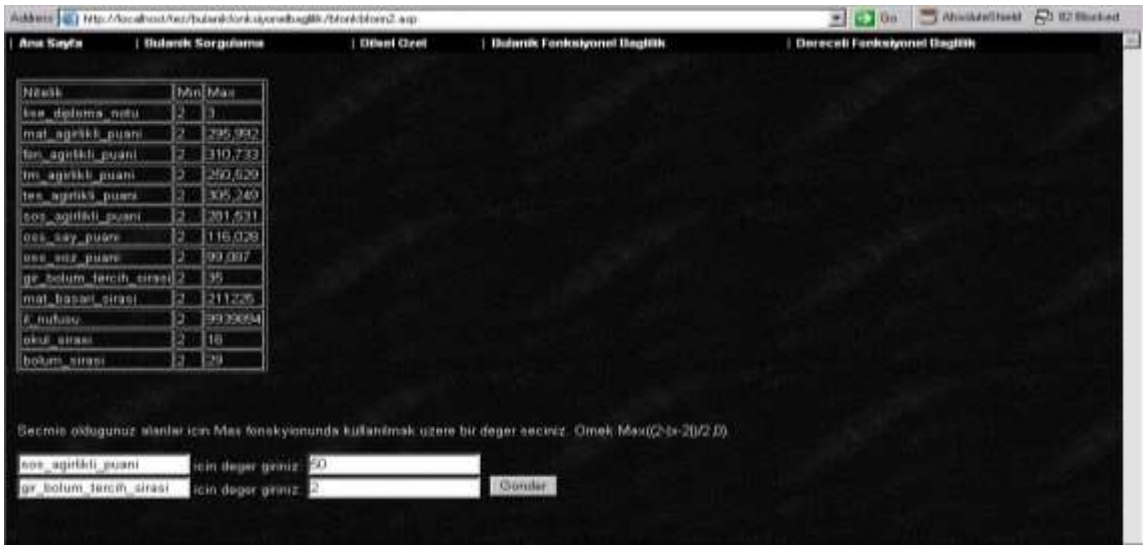
Şekil 5.8 Dilsel Özet Ayrıntı Tablosu

5.2.3. Bulanık Fonksiyonel Bağlılık

Geliştirilen uygulamalardan biri de bulanık fonksiyonel bağlılıktır. Kullanıcı belirli bir alan üzerinde benzer değerlere sahip olan verilerin, seçtiği diğer alan üzerinde de benzer değerlere sahip olup olmadığını sınamaktadır. Şekil 5.9'da yer alan "sosyal ağırlıklı puanı benzer olan öğrencilerin girdiği bölümü tercih etme sırası da benzerdir" ifadesi bulanık fonksiyonel bağlılığa bir örnek göstermektedir.

Şekil 5.9 Bulanık Fonksiyonel Bağlılık Başlangıç Sayfası

Bölüm 4.5.1'de anlatıldığı gibi öncelikle ilk alan olan sosyal ağırlıklı puan kendi kendisiyle kıyaslanarak benzer değerler tespit edilir. Daha sonra maksimum fonksiyonu aracılığıyla iki alan arasındaki benzerliği tanımlamaya yarayacak kurallar oluşturulur ve son olarak tüm kurallar yardımıyla bulanık fonksiyonel bağlılığın doğruluk değeri hesaplanır. Maksimum fonksiyonu içinde kullanılan bulanıklık operatörü, kullanıcının benzerlik görmek istediği alana göre değişeceğinden, kullanıcı bu değeri elle girmelidir. Dilsel özette olduğu gibi kullanıcının girebileceği miktarların alt ve üst sınır değerleri bir tablo ile gösterilmektedir. Kullanıcının seçtiği niteliğe ait olan sınırlar içerisinde kalması denetlenmektedir. Unutulmamalıdır ki, maksimum fonksiyonu içerisinde yer alması istenen değer ne kadar yüksek seçilirse, benzerlik iddiasının doğruluğu o kadar yüksek çıkacaktır.



Şekil 5.10 Bulanık Fonksiyonel Bağlılıktaki Max Fonksiyon Değerinin Atanması

Doğruluk değeri yeni bir sayfada gösterilmekte ve kullanıcıya dilsel özette olduğu gibi ayrıntıları görme seçeneği sunulmaktadır.

5.2.4. Dereceli Fonksiyonel Bağlılık

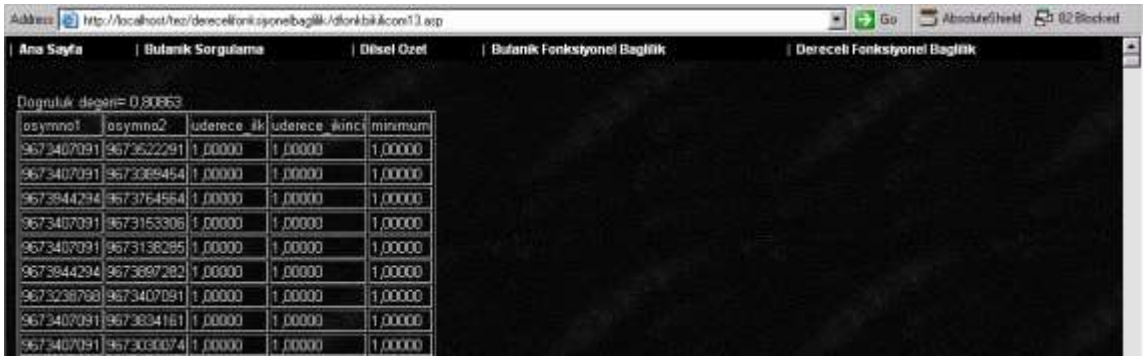
Dereceli fonksiyonel bağlılık uygulaması iki başlık altında geliştirilmiştir. İlkinde "veri tabanındaki öğrencilerin ÖSS sayısal puanı daha düşük olanların matematik başarı sırası daha yüksektir" biçiminde tek bir koşul ve bir öner meden oluşan dereceli fonksiyonel

bağlılıklar incelenirken; ikincisinde iki koşul ve bir öner meden oluşan dereceli fonksiyonel bağlılık incelenmektedir.

Şekil 5.11 Dereceli Fonksiyonel Bağlılık

Şekil 5.11'de görülen iki koşullu dereceli fonksiyonel bağlılık örneğinde koşullar daha yüksek fen ağırlıklı puanı ve daha düşük nüfuslu ildir. Buna karşın getirilen öner me ise daha yüksek türkçe matematik puanıdır.

Doğruluk değerinin hesaplanması için öncelikle dereceli fonksiyonel bağlılığın koşul kısmı için [0,1] aralığında yer alan değerler hesaplanır. Kullanıcı bulanık fonksiyonel bağlılık uygulamasında olduğu gibi maksimum fonksiyonunda yer alacak, sunduğu öner meye ait sınır değerler içinde kalan bir bulanıklık operatörü belirlemelidir. Maksimum fonksiyonunun kullanıldığı öner me kısmı için de değerler hesaplandıktan sonra bulanık bir kural oluşturulmuş olur. Her bulanık kuralın değeri hesaplanır ve bulanık kural değerlerinin toplanıp mevcut veri sayısına bölünmesi ile dereceli fonksiyonel bağlılığın doğruluk derecesi hesaplanır.



Şekil 5.12 Dereceli Fonksiyonel Bağlılık Sonuç Sayfası

Bulanık ve dereceli fonksiyonel bağlılık uygulamalarına diğer uygulamalardan farklı olarak fazladan bir kısıt eklenmiştir; çünkü veri tabanının MS Access'te tutulmuş olması ve web ortamı performansı olumsuz yönde etkilemektedir. Bu iki uygulama da birbiri içine geçmiş işlemlerden oluştuğundan tüm veri tabanı üzerinde uygulanması oldukça uzun zaman almaktadır. Bu nedenle veri tabanının bir alt kümesi keyfi olarak seçilmiş ve bu iki uygulama bu kümeler üzerinde gerçekleştirilmiştir.

6. SONUÇ

Veri madenciliği değişeniş yaşamı ve gelişen teknoloji sayesinde gün geçtikçe önemini arttırmaktadır. Hedeflenen bilgiye ulaşmak için hangi veri madenciliği yönteminin kullanılacağı değişmektedir. Önemli olan en etkin ve verimli bir şekilde bilgiye ulaşma, bilgiyi değerlendirme olanağı veren yöntemin seçilmesidir.

Bulanık mantık gerek insanın düşünce yapısına uyan bir yöntem olması nedeniyle kolay anlaşılabilir; gerekse matematiksel olarak modellerinin çıkarılması zor olan problemlerde de kullanılabilir olması nedeniyle bir çok alanda yaygın bir şekilde kullanılmaya başlanmıştır. Kullanım alanlarından biri de bu tez çalışmasının konusu olan veri tabanları ve dolayısıyla veri madenciliğidir.

Veri madenciliğinde bulanık mantık sayesinde günlük hayatta kullanılan ifadelerle sorgulamalar yapılabilmekte, dilsel belirsizliklerin veri tabanı sorgulamalarında ve özetlemelerinde kullanılmasına olanak sağlanmaktadır. Bulanık fonksiyonel bağlılıklar ve dereceli fonksiyonel bağlılıklar ile veriler arasındaki ilişkiler daha geniş kapsamda incelenebilmektedir. Daha önceden de belirtildiği gibi bu iki uygulamada dikkat edilmesi gereken nokta benzerliği tanımlarken kullanılan bulanıklık operatörünün değeridir. Çünkü kuralların doğruluk değerlerini doğrudan etkilemektedir. Sonuçların değerlendirilmesinde, kullanıcının karar vereceği alt sınırlara göre, bulunan doğruluk değerleri kabul görmektedir.

Tez çalışmasının uygulaması, farklı veri tabanı yönetim sistemi ve veri tabanı üzerinde çalışacak şekilde geliştirilmiştir. Uygulamada kullanılan veri tabanı yönetim sistemi gelişmiş bir tanesi ile değiştirildiğinde performansın daha da artacağına inanmaktayım. Uygulama web ortamında kullanıcılara sunulmaktadır. İstenirse görsel açıdan sayfalar geliştirilebilir.

Bulanık mantık ile geliştirilmiş ara yüzler sayesinde bilgisayar veya veri tabanı bilgisi çok gelişmiş olmayan kişilerin de çok etkin bir şekilde bilgiye ulaşacaklarına ve bu tarz ürünlerin piyasada bulunan birçok klasik veri tabanı uygulamasından daha fazla rağbet göreceğine inanmaktayım. Bu amaçla bulanık mantığın veri madenciliğinde kullanımı çalışmalarına büyük bir ivmeyle hız verilmesi gerektiğini savunmaktayım.

7. KAYNAKLAR

[1] **Akpınar, H,** 1998. Veri tabanlarında bilgi keşfi ve veri madenciliği. *www.google.com/veri_madenciligi*, İ. Ü İşletme Fakültesi, İstanbul.

[2] **Alpaydın, E.,** 2000. Zeki veri madenciliği: Ham veriden altın bilgiye ulaşma yöntemleri. *Bilişim2000 Semineri*, İstanbul.

[3] **Corporation.,** 1999. Introduction to data mining and knowledge discovery, third edition. *www.twocrows.com*, Two Crows Corporation. Potomac, MD.

[4] **Editör,** 2002. Bilinmeyen yönleriyle veri madenciliği. *Infomag*, Aralık, İstanbul.

[5] **Elmas, Ç.,** 2003. Bulanık Mantık Denetleyiciler (Kuram, Uygulama, Sinirsel Bulanık Mantık). Seçkin Yayıncılık, Ankara.

[6] **Fayyad, U and Stolorz P.,** 1997. Data mining: promise and challenges. *Futur Generation Computer Systems*. **13**, 99-115.

[7] **Goebel M and Gruenwald L.,** 1999. A survey of data mining and knowledge discovery software tools. SIGKDD Explorations, **1**, 20-33.

[8] **Hand, D J.,** 1999. Statistics and data mining: Intersecting disciplines. SIGKDD Explorations, **1**, 16-19.

[9] **Levin N and Zahavi J.,** 1999. Data mining. *www.elsevier/data_mining/data_mining.pdf*

[10] **Oberlé, V.,** 2000. Data mining: eine Einführung. *www.elsevier/data_mining/vincent_oberle.pdf*.

[11] **Rasmussen, D and Yager R R,** 1999. Finding fuzzy and gradual functional dependencies with SummarySQL, *Fuzzy Sets and Systems*, **106**, 131-142.

[12] **Rasmussen, D and Yager R R,** 1997. SummarySQL – A Fuzzy Tool For Data Mining, *Intelligent Data Analysis*, **1**, 39-52.

[13] **Rygielski, C, Wang, L C and Yen, D C,** 2002. Data mining techniques for customer relationship management. www.elsevier.com/locate/techsoc.

[14] **Solarte, J.,** 2002. A proposed data mining methodology and its application to industrial dynamics. *Master Thesis*. University of Tennessee, Knoxville.

[15] **Şen, Z.,** 2001. Bulanık Mantık ve Modelleme İlkeleri. Bilge Kültür Sanat, İstanbul.

[16] **Vahaplar, A ve İnceoğlu M M,** 2000. Veri madenciliği ve elektronik ticaret. http://www.bayar.edu.tr/bid/dokumanlar/inceoglu.doc, Ege Üniversitesi, İzmir.

[17] http://dms.irb.hr/tutorial/

[18] www.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/

[19] www.thearling.com/text/dmtechniques/dmtechniques.htm

ÖZGEÇMİŞ

Bu tezi hazırlamış olan Ayşen BÜYÜKAKIN, 30.09.1978 İstanbul doğumludur. İlkokulu Kazım Karabekir İlkokulu'nda okumuş, ortaokul ve liseyi Kartal Anadolu Lisesi'nde bitirmiştir. 1996 yılında girdiği İstanbul Teknik Üniversitesi Fen-Edebiyat Fakültesi Matematik Mühendisliği Bölümü'nden 2001 yılında mezun olmuştur. Aynı yıl İstanbul Teknik Üniversitesi Sosyal Bilimler Enstitüsü İşletme Bölümü'nde ve Fen Bilimleri Enstitüsü Sistem Analizi Bölümü'nde yüksek lisans eğitimlerine başlamış, 2003 yılında İşletme yüksek lisans eğitim programından mezun olmuştur.