T.C.
BÜLENT ECEVİT ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
BİYOİSTATİSTİK ANABİLİM DALI

K-EN YAKIN KOMŞULUK, YAPAY SİNİR AĞLARI VE KARAR AĞAÇLARI YÖNTEMLERİNİN SINIFLANDIRMA BAŞARILARININ KARŞILAŞTIRILMASI

Fürüzan KÖKTÜRK

DOKTORA TEZİ

TEZ DANIŞMANI
Prof. Dr. Vildan SÜMBÜLOĞLU

ZONGULDAK
2012

T.C.
BÜLENT ECEVİT ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
BİYOİSTATİSTİK ANABİLİM DALI

K-EN YAKIN KOMŞULUK, YAPAY SİNİR AĞLARI VE KARAR AĞAÇLARI YÖNTEMLERİNİN SINIFLANDIRMA BAŞARILARININ KARŞILAŞTIRILMASI

Fürüzan KÖKTÜRK

DOKTORA TEZİ

TEZ DANIŞMANI
Prof. Dr. Vildan SÜMBÜLOĞLU

ZONGULDAK
2012

KABUL ve ONAY:

'K-EN YAKIN KOMŞULUK, YAPAY SİNİR AĞLARI VE KARAR AĞAÇLARI YÖNTEMLERİNİN SINIFLANDIRMA BAŞARILARININ KARŞILAŞTIRILMASI' başlıklı bu çalışma jürimiz tarafından değerlendirilerek, Biyoistatistik Anabilim Dalı doktora tezi olarak kabul edilmiştir.
27.04.2012

Başkan: Prof. Dr. Vildan SÜMBÜLOĞLU

Üye: Prof. Dr. Orhan BABUCÇU

Üye: Doç. Dr. Çağatay BARUT

Üye: Doç. Dr. Erdem KARABULUT

Üye: Doç. Dr. Pınar ÖZDEMİR

ONAY:
Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.
TARİH:

Doç. Dr. Fermuh Niyazi AYOĞLU
Sağlık Bilimleri Enstitüsü Müdürü

ÖNSÖZ

Her konuda bilgi ve tecrübelerinden faydalandığım, bana her zaman destek olan ve arkamda olduğunu hissettiren Sayın Hocam Prof. Dr. Vildan Sümbüloğlu'na,

Verilerini bizimle paylaşan ve desteklerini esirgemeyen BEÜ Tıp Fakültesi Aile Hekimliği Anabilim Dalı Başkanı Doç. Dr. Nejat Demircan, Kadın Hastalıkları Anabilim Dalı Öğretim Üyesi Doç. Dr. Ülkü Bayar ve Uzman Dr. Evren Kurtul'a,

Tezimle ilgili resmi işlemlerde bana yardımcı olan tüm Sağlık Bilimleri Enstitüsü personeline,

Bugüne kadar hep yanımda olduklarını hissettiren ve beni destekleyen arkadaşlarıma ve aileme sonsuz teşekkürü bir borç bilirim.

Fürüzan KÖKTÜRK
Nisan 2012, ZONGULDAK

ÖZET

Fürüzan Köktürk, K-En Yakın Komşuluk, Yapay Sinir Ağları ve Karar Ağaçları Yöntemlerinin Sınıflandırma Başarılarının Karşılaştırılması. Bülent Ecevit Üniversitesi, Sağlık Bilimleri Enstitüsü, Biyoistatistik Anabilim Dalı, Doktora Tezi, Zonguldak 2012.

Tıp alanında bulunan mevcut veri oldukça fazla ve hayati öneme sahiptir. Veri madenciliği teknikleri ile hayati öneme sahip olan bu verilerden daha fazla yararlanmak mümkündür.

Veri madenciliği son yıllarda oldukça önemli bir konu haline gelmesine ve hemen hemen her alanda uygulama sahası bulmasına rağmen ülkemizde sağlık alanında çok yaygın kullanılmamaktadır.

Bu tez çalışmasında veri madenciliği yöntemlerinden, k-en yakın komşuluk, yapay sinir ağları ve karar ağaçları yöntemlerinin sınıflandırma başarılarının karşılaştırılması amaçlanmıştır. Bu amaçla Bülent Ecevit Üniversitesi Uygulama ve Araştırma Hastanesi Kadın Hastalıkları ve Doğum Polikliniği'ne başvuran erken ve zamanında doğum yapan gebelerden elde edilen veri setine bu üç teknik uygulanarak, sınıflandırma başarıları hesaplanmıştır. Yapılan analizler sonucunda doğru sınıflandırma oranları, k-en yakın komşuluk analizi için % 78.3, yapay sinir ağı tekniği için % 90.8 ve karar ağacı yöntemi için ise % 82.5 olarak bulunmuş ve yapay sinir ağı tekniğinin diğer iki yönteme göre sınıflandırma başarısının daha iyi olduğu görülmüştür.

Anahtar Kelimeler: Veri madenciliği, k-en yakın komşuluk, yapay sinir ağları, karar ağaçları, doğru sınıflandırma oranı

ABSTRACT

Furuzan Kokturk, Comparing Classification Success of K-Nearest Neighbor, Artifical Neural Network and Decision Trees. Bulent Ecevit University, Institute of Health Science, Department of Biostatistics, PhD Thesis, Zonguldak 2012.

The amount of medical data is huge and vital. It is possible to obtain more benefit from these data by data mining techniques.

Although the data mining has been becoming a very important subject and being used in almost all fields in recent years, it has no widely use in the health sector in our country.

In this thesis study, it was aimed to compare of the classification success of the k-nearest neighbor, artifical neural network and the decision trees techniques. For this purpose, these three techniques were applied and the classification success was measured on the pregnants those gave preterm birth and those gave birth in time in Departments of Obstetrics and Gynecology of Bulent Ecevit University. After the analysis of the results, the correct classification ratios found to be 78.3 % for k-nearest neighbor method, 90.8 % for artifical neural network, 82.5 % for decision trees method and it was concluded that the artifical neural network is more successful than the other two methods.

Keywords: Data mining, k-nearest neighbor, artifical neural network, decision trees, correct classification ratio

İÇİNDEKİLER

Sayfa

SİMGELER VE KISALTMALAR

AID : Automatic Interaction Detector
ABD : Amerika Birleşik Devletleri
ART : Adaptive Resonance Theory
BEÜ : Bülent Ecevit Üniversitesi
BKS : Beyaz Kan Hücresi Sayısı
CART : Classification and Regression Trees
CHAID : Chi-squared Automatic Interaction Detector
CT : Classification Trees
EEG : Elektroensefalogram
EKG : Elektrokardiyografi
GRNN : Generalized Regression Neural Network
ID3 : Iterative Dichotomiser 3
LVQ : Learning Vector Quantisation Network
MARS : Multivariate Adaptive Regression Splines
OLAP : Online Analytical Processing
PNN : Probabilistic Neural Network
RBF : Radial Bases Function Network
RT : Regression Trees
SGA : Small for gestational age
SLIQ : Supervised Learning in Quest
SPRINT : Scalable Parallelizable Induction of Decision Trees
VKİ : Vücut Kütle İndeksi
VM : Veri Madenciliği
VTBK : Veri Tabanlarında Bilgi Keşfi
YSA : Yapay Sinir Ağları
QUEST : Quick, Unbiased, Efficient Statistical Tree

ŞEKİL DİZİNİ

Şekil **Sayfa**

TABLO DİZİNİ

Tablo **Sayfa**

1. GİRİŞ VE AMAÇ

Günümüzde bilgi kazanılan en önemli değerlerdendir. Bilginin önemi ona olan ihtiyacı arttırmıştır. Bunun sonucu olarak bilişim sistemlerindeki hızlı gelişme ve otomatik veri depolama araçlarındaki teknolojik gelişmeler yoluyla artık yaptığımız her işlem kayıt altına alınmaya başlanmıştır. Organizasyonlar, firmalar tüm mali ve operasyon bilgilerini veri tabanlarında, veri ambarlarında depolamaktadırlar. Veri tabanlarında depolanan veriler arasındaki ilişkiler, saklı kalmış bilgiler çıkarılmayı beklemektedir. Depolanan veri yığınları arasında saklı kalmış bilgilerin nasıl açığa çıkarılabileceği üzerine yapılan çalışmalar sonucu Veri Tabanlarında Bilgi Keşfi (VTBK) kavramı ortaya çıkmıştır.

VTBK süreci içerisinde, modelin kurulması ve değerlendirilmesi aşamalarından oluşan veri madenciliği (Data Mining) en önemli kesimi oluşturmaktadır. Bu önem, birçok araştırmacı tarafından VTBK ile veri madenciliği terimlerinin eş anlamlı olarak kullanılmasına neden olmaktadır (1).

Veri madenciliği yöntemleri genel olarak kestirim modelleri ve tanımlayıcı modeller olmak üzere iki ana başlık altında incelenebilir. Ancak bazı yöntemler hem tanımlayıcı hem de kestirim özelliğene sahip olabilmektedir.

Kestirim modellerinde, sonuçları bilinen verilerden yola çıkarak bir model kurulması ve kurulan bu modelden sonuçları bilinmeyen verilerin sonuç değerlerinin kestirilmesi amaçlanmaktadır (1).

Tanımlayıcı modellerde, eldeki verilerden strateji geliştirme ve karar verme süreçlerinde kullanılabilecek bilgiler sağlanmaktadır. Tanımlayıcı modeller daha çok veriler arasındaki gizli kalmış ilişkiyi ortaya çıkarırlar (2).

Tanımlayıcı modeller işlevine göre; Kümeleme (Clustering) ve Birliktelik Kuralları (Association Rules) olarak alt bölümlere ayrılmaktadır. Kestirim modellerini ise Sınıflama (Classification) ve Regresyon (Regression) olarak iki alt başlıkta toplamak mümkündür.

Sınıflama, günlük yaşantımızda olduğu gibi bilimsel çalışmalarda da sorunların çözümünde sağladığı fayda nedeniyle oldukça sık başvurulan bir işlemdir. Tıp alanında hastalıkların sınıflandırılması ve bu sınıflandırmaya göre tedavi

yöntemlerinin geliştirilmesi en belirgin örneklerdendir. Tıbbın yanında diğer bilim dallarında da sınıflandırmanın işlerliği görülebilmektedir.

Özellikle tıp ve biyoloji alanında yapılan çalışmalarda veri setleri oldukça karmaşık bir yapı teşkil etmektedir. Veriler üzerinde çalışılmadan önce bu karmaşık yapının düzenlenmesi gerekmektedir; düzenleme, belirlenen amaç doğrultusunda sınıflandırmanın yapılması ile mümkün olmaktadır (3).

Veri madenciliği metotları içerisinde sınıflandırma amacıyla kullanılan pek çok metot mevcuttur. Bu tez çalışmasında veri madenciliği yöntemlerinden sınıflandırma amacıyla kullanılan, k-en yakın komşuluk, yapay sinir ağları ve karar ağaçları yöntemlerinin sınıflandırma başarılarının karşılaştırılması amaçlanmıştır.

2. GENEL BİLGİLER

Bilgisayar teknolojisindeki gelişmelerle birlikte üretilen bilgi miktarının arttığı, veri tabanlarının daha fazla veriyi saklayabilecek boyutlara ulaştığı ve veriye ulaşmanın kolaylaştığı görülmektedir (4). Öyle ki, dünyadaki bilgi miktarının her 12-18 ayda bir ikiye katlandığı tahmin edilmektedir. Veri tabanı sistemlerinin artan kullanımı ve hacimlerindeki bu olağanüstü artış, organizasyonları elde toplanan bu verilerden nasıl faydalanılabileceği problemi ile karşı karşıya bırakmıştır. Geleneksel sorgu veya raporlama araçlarının veri yığınları karşısında yetersiz kalması, **Veri Tabanlarında Bilgi Keşfi-VTBK** *(Knowledge Discovery in Databases)* adı altında, sürekli ve yeni arayışlara neden olmuştur. Büyük miktarlardaki verinin veri tabanlarında tutuldukları bilindiğine göre bu verilerin veri madenciliği teknikleriyle işlenmesine "Veri Tabanında Bilgi Keşfi" denir (1).

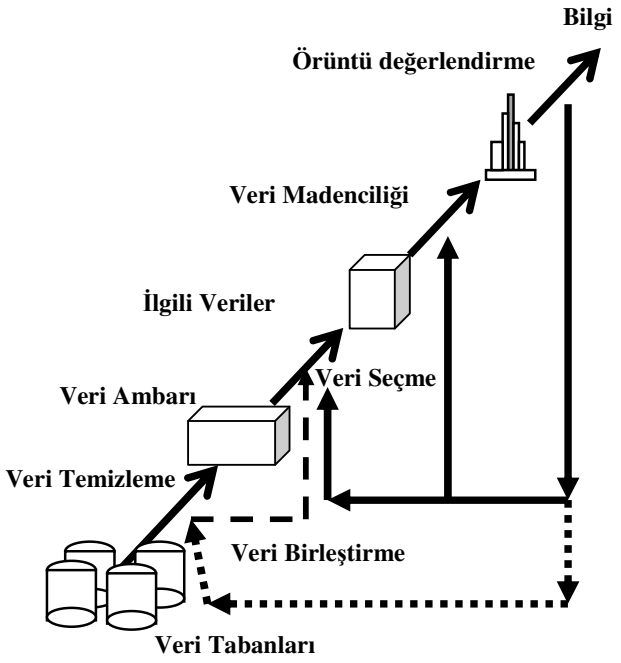
VTBK ve veri madenciliği terimleri farklı süreçleri işaret etseler de çoğunlukla birbirlerinin yerine kullanılırlar. VTBK, veriden faydalı bilgi keşfinin tüm aşamalarını ifade ederken veri madenciliği bu süreçte belirli bir adımdır. Tarihsel olarak veri içerisindeki faydalı yapıların ortaya çıkarılması olayına pek çok isim karşılık gelmiştir. Bunlardan bazıları veri madenciliği, bilgi çıkarımı, bilgi keşfi, bilgi harmanlama, veri arkeolojisi ve veri modelleme sürecidir (5).

Veri Tabanlarında Bilgi Keşfi süreci, birbirini etkileyen ve yinelemeli bir süreç olup aşağıdaki adımlardan oluşmaktadır:

- **Veri temizleme:** Verilen örneklemdeki gürültülü[1] ve gereksiz verinin silinmesi işlemlerini içerir.
- **Veri bütünleştirme:** Bu aşamada çoklu veri kaynaklarından, genellikle heterojen veriler ortak bir kaynakta birleştirilir.
- **Veri seçimi:** Bu evrede analizle ilişkili olabilecek verilere karar verilir ve uygun örneklem kümesi elde edilir.
- **Veri dönüşümü:** Seçilen verinin madencilik süreci için uygun bir şekle dönüştürüldüğü evredir.

[1] Veri girişi veya veri toplanması sırasında oluşan sistem dışı hatalar

- **Veri madenciliği:** Potansiyel faydalı bilgileri çıkarmak için zeki tekniklerin uygulandığı en önemli evredir.
- **Örüntü değerlendirme:** Keşfedilen bilginin geçerlilik, yenilik, yararlılık ve basitlik kıstaslarına göre değerlendirilmesi aşamasıdır.
- **Bilgiyi sunma:** Keşfedilmiş bilginin görsel olarak kullanıcılara sunulduğu final aşamasıdır. Görüntüleme tekniklerinin kullanıldığı bu aşama kullanıcıların veri madenciliği sonuçlarını yorumlama ve anlamasına yardım etmektedir (6) (Şekil 1).



Şekil 1. Veri tabanlarında bilgi keşfi süreci.

2.1. Veri Madenciliği Nedir?

Veri madenciliği, veri ambarlarında yararlı olma potansiyeline sahip, aralarında beklenmedik/bilinmedik ilişkilerin olduğu verilerin keşfedilerek, hem anlaşılır hem de kullanılabilir bir biçime getirilmesine yönelik geliştirilmiş yöntemler topluluğudur (5). Veri madenciliğinin ilk uygulamaları pazarlama alanında olmuştur. Daha sonraki yıllarda karar alma ve bilgi yönetimi süreçlerinde yoğun bir şekilde kullanılmaya başlanmıştır (5, 7).

Veri madenciliğinin görevleri,

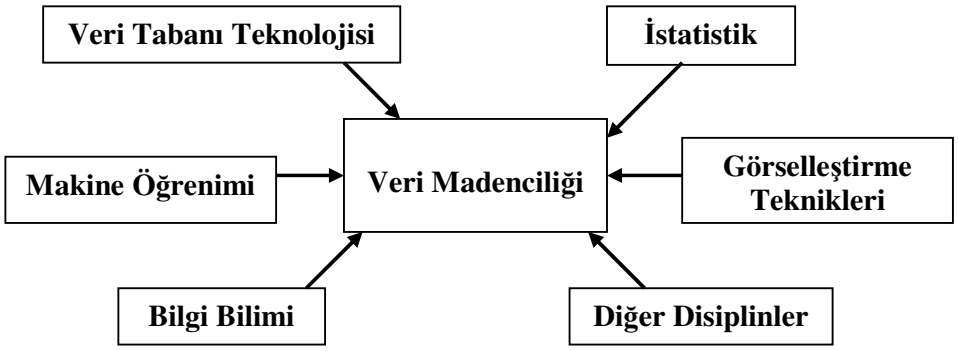
- Sınıflandırma
- Kestirim
- Bölümlendirme (İlişkilendirme ve kümeleme)
- Tanımlama

olarak sınıflandırılabilir (7).

Veri madenciliği (VM), kavramsal olarak 1960'lı yıllarda bilgisayarların veri analiz problemlerini çözmek için kullanılmaya başlamasıyla ortaya çıkmıştır. O dönemlerde bilgisayar yardımıyla yeterince uzun bir tarama yapıldığında istenilen verilere ulaşmanın mümkün olacağı gerçeği kabullenildi. Bu işleme veri madenciliği yerine önceleri veri taraması (data dredging), veri yakalanması (data fishing) gibi isimler verildi. 1990'lı yıllara gelindiğinde veri madenciliği ismi bilgisayar mühendisleri tarafından ortaya atıldı. Bu topluluğun amacı, geleneksel istatistiksel yöntemler yerine, veri analizinin algoritmik bilgisayar modülleri tarafından değerlendirilmesini vurgulamaktı. Bu noktadan sonra bilim adamları veri madenciliğine çeşitli yaklaşımlar getirmeye başladılar (8).

Veri madenciliği büyük miktarda veriyi inceleme amacı üzerine kurulmuş olduğu için veri tabanları ile yakından ilişkilidir. Gerekli verinin hızla ulaşılabilecek şekilde amaca uygun bir şekilde saklanması ve gerektiğinde veriye hızla ulaşılabilmesi gerekir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş ve işlenmeye daha uygun bir özetini saklamayı amaçlar. Günlük veri tabanlarından istenen özet bilgi seçilir ve gerekli ön işlemeden sonra veri ambarında saklanır. Ardından amaç doğrultusunda gerekli veri ambardan alınarak veri madenciliği çalışması için standart bir forma çevrilir. Veri ambarlarının analizi için OLAP (*Online Analytical Processing)* programları kullanılır. OLAP veriye çok boyutlu bakmayı ve incelemeyi sağlar (5).

Veri madenciliği, disiplinler arası doğasından dolayı istatistik, veri tabanları, makine öğrenimi, bilgi toplama, görselleştirme, paralel ve dağıtık hesaplama gibi birçok disiplinden yardım alır (9).



Şekil 2. Veri madenciliğinin diğer disiplinlerle ilişkisi.

2.1.1. Danışmanlı (Supervised) öğrenme

Belirli bir amaca ve sonuca yönelik olarak yapılan veri madenciliği yöntemlerine danışmanlı yöntemler denilebilir. Danışmanlı yöntemler veriden bilgi ve sonuç çıkarmaya yönelik kullanılmaktadır. Örnekten öğrenme olarak da isimlendirilen danışmanlı öğrenimde, ilgili sınıflar önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunması ve bu özelliklerin kural cümleleri ile ifade edilmesidir. Öğrenme süreci tamamlandığında, tanımlanan kural cümleleri verilen yeni örneklere uygulanır ve yeni örneklerin hangi sınıfa ait olduğu kurulan model tarafından belirlenir.

Danışmanlı öğrenimde seçilen algoritmaya uygun olarak ilgili veriler hazırlandıktan sonra ilk aşamada verinin bir kısmı modelin öğrenimi (Learning Dataset), diğer kısmı ise modelin geçerliliğinin test edilmesi (Testing Dataset) için ayrılır. Öğrenim veri seti kullanılarak model kurulduktan sonra test veri seti ile modelin doğruluk derecesi belirlenir. Test sonucunda doğruluk derecesine göre üç durum ortaya çıkabilir. Bunlar, modelin kabul edilmesi, modelin yeniden ele alınarak bir iyileştirme yapılması veya modelin tamamen reddedilmesidir.

Danışmanlı veri madenciliği yöntemlerine örnek olarak, k-en yakın komşuluk, yapay sinir ağları, karar ağaçları, k-ortalamalar kümeleme gibi yöntemler gösterilebilir.

2.1.2. Danışmansız (Unsupervised) öğrenme

Ulaşılmak istenen sonuç için bir tanımlama yapılmamışsa veya bir belirsizlik varsa danışmansız öğrenmeden bahsedilebilir. Danışmansız öğrenme daha çok veriyi anlamaya, tanımaya ve keşfetmeye yönelik olarak kullanılmaktadır.

Danışmansız öğrenmede ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır (10).

Danışmansız bir yöntemle elde edilen bir bilgi veya sonucu, eğer mümkünse denetimli bir yöntemle doğrulamak elde edilen bulguların doğruluğu ve geçerliliği açısından önem taşımaktadır.

Danışmansız veri madenciliği yöntemlerine, aşamalı kümeleme ve kendi kendini düzenleyen haritalar örnek verilebilir.

2.1.3. Veri madenciliği uygulama alanları

Veri madenciliğinin günümüzde karar verme sürecine ihtiyaç duyulan birçok alanda uygulaması mümkün olmakla birlikte belli başlı kullanım alanları olarak;

Pazarlama: Pazar bölümlemesi, müşteri değerleme ve çapraz satış analizleri,

Bankacılık: Risk analizi, usulsüzlük tespiti, müşteri kazanma ve mevcut müşterileri elde tutma analizleri, çapraz satış,

Sigortacılık: Müşteri kaybı sebeplerinin belirlenmesi, usulsüzlüklerin önlenmesi, ana giderlerin azaltılması, poliçe fiyatlarının belirlenmesi,

Perakendecilik: Satış noktası veri analizleri, alış veriş sepeti analizleri, tedarik ve mağaza yerleşim optimizasyonları,

Borsa: Hisse senedi fiyat tahmini, genel piyasa analizleri, alım-satım stratejilerinin optimizasyonu,

Telekomünikasyon: Kalite iyileştirme, hile tespiti, hataların yoğunluk tahmini, müşteri kazanma ve elde tutma analizleri,

İlaç: Test sonuçlarının tahmini, ürün geliştirme,

Sağlık: Tıbbi teşhis, uygun tedavi sürecinin belirlenmesi,

Endüstri: Kalite kontrol, lojistik, üretim süreçlerinin optimizasyonu,

Bilim ve Mühendislik: Ampirik veriler üzerinde modeller kurularak bilimsel ve teknik problemlerin çözümlenmesi sayılabilir.

Veri analizi ve veri madenciliği uygulamaları, günümüzde başta ABD ve Avrupa Birliği ülkeleri olmak üzere pek çok ülkede üniversite başvurularının değerlendirilmesi ve burs verilmesi, işsizlik sigortası değerlendirmeleri, vergi iadelerindeki usulsüzlük tespiti, sosyal sigortalardaki hileli kullanım tespiti, vergi sistemindeki değişikliklerin bütçeye etkisininin öngörülmesi, bütçe likidite yönetimi, vergi usulsüzlüklerinin tespiti, güvenlik ve savunma alanındaki uygulamalara kadar pek çok alanda yaygın biçimde kullanılmaktadır (11).

2.1.4. Tıpta veri madenciliği

Yukarıda belirtilen alanlar dışında veri birikmesinin en yoğun yaşandığı alanlardan birisi de tıp sektörüdür. Özellikle günümüzde artık neredeyse tüm tıbbi cihazların dijital hale gelmesi bu sonucu doğal hale getirmiştir. Tıp alanında bulunan mevcut veri hem oldukça fazla hem de hayati öneme sahiptir. Hastane bilgi sistemleri sayesinde bu veriler düzenli olarak tutulmaktadır. Hayati öneme sahip olan bu verilerden daha fazla yararlanmak mümkündür. Bu aşamada yardıma veri madenciliği teknikleri yetişmektedir. Bu sayede aşırı miktardaki verinin zeki olarak işlenip yorumlanması mümkün hale gelmiştir. Böylece klasik yöntemlerle bulunması çok zor veya imkansız olan bazı ilişkilerin de bu sayede ortaya çıkartılması imkanı oluşmuştur.

Tıpta veri madenciliği; tıp alanındaki uzmanlar, veri madenciliği uzmanları ve veri işleyicilerinin sıkı bir şekilde birleştiği bir olgudur. Hastane bilgi sistemlerinden veya diğer tıbbi veri toplayan sistemlerden alınan veriler üzerinde yapılan veri madenciliği çalışmaları hem uzmanlar, hem hastane yönetimi için hem de hastaların daha kaliteli bir hizmet almalarında etkin rol alabilir (9).

Kağıt üzerinde veri toplanan klasik hastane bilgi sistemlerinden farklı olarak buradaki verilerden yararlanmak çok daha kolay gibi görünsede aslında diğer

alanlardaki veriler gibi bunların da bireysel çalışmalarla işlenmesi ve yorumlanması kolay değildir. Tıbbi veri tabanlarında veri madenciliği ve bilginin bulunması da diğer türdeki veri tabanlarındakinden çok farklı değildir. Ancak, tıbbi veride diğer veri türlerinde olmayan bazı özellikler vardır. Bu özellikler aşağıdaki gibi özetlenebilir:

- Çok sayıda yordam görüntülemeyi bir tanı aracı olarak kullanmaktadır. Bu nedenle, görüntülerden oluşan veri tabanlarında etkin bir veri madenciliği gerçekleştirebilmek için yöntemler geliştirmek gerekmektedir. Bu da sayısal veri tabanlarındaki veri madenciliğinden hem daha farklı hem de daha zordur.
- Tıbbi veri tabanları her zaman heterojendir. Örneğin bir organa ait görüntü her zaman hekimin yorumu (klinik izlenim, tanı) gibi başka klinik bilgilerle bir aradadır. Bu ise, bu tür verilerin çözümlemesi için yeni araçlar ve yüksek kapasiteli veri depolama aygıtları gerektirir.
- Hekimler görüntüler, sinyaller ya da diğer klinik bilgilerle ilgili yorumlarını, standartlaştırılması çok güç olan serbest metinler olarak yazmaktadırlar. Örneğin aynı hastalık açıklanırken bile farklı adlar kullanılabilmektedir. Tıbbi kavramlar arasındaki ilişkileri açıklamak için de farklı dilbilgisi yapıları kullanılmaktadır.
- Gizlilik, güvenlik ve hasta mahremiyeti gibi konular veriye erişimde kısıtlama getirmektedir. Veri internet üzerinden elektronik olarak aktarıldığından güvenli değildir. Bu nedenle veri bir kurum içinde bir birimden diğerine aktarılacak olsa da dikkatli bir biçimde şifrelenmelidir.
- Tıp alanındaki veri genellikle farklı kaynaklarda toplanmaktadır. Örneğin hastanın laboratuvar ile ilgili verileri ile hastanın teşhis bilgileri farklı kaynaklarda ve farklı şekillerde tutulmaktadır.
- Tıptaki temel veri yapıları birçok alanla karşılaştırıldığında matematiksel olarak karakterize edilmeye pek uygun değildir. Veri madencisinin bilgiyi düzenleyebileceği, kümeleme, gerileme modelleri ya da dizi çözümlemeleri gibi karşılaştırılabilir yapılar yoktur (12).

Tıp alanında veri madenciliği uygulamalarına örnek olarak;

- Belirli bir hastalığa sahip hastaların ortak özelliklerinin tahmin edilmesi,
- Tıbbi tedaviden sonra hastaların durumlarının tahmin edilmesi,
- Hastane maliyetlerinin tahmin edilmesi,
- Ölüm oranları ve salgın hastalıkların tahmin edilmesi,
- Genetik bozuklukların tespiti,
- İlaç yan etkilerinin tanımlanması gibi çeşitli çalışmaları sayabiliriz (13).

2.1.5. Veri madenciliğinde kullanılan teknikler

Veri madenciliğinde kullanılan teknikler hem özel türde veri yapıları hem de belirli algoritmik yaklaşımlar gerektirir. Bunlar önce parametrik modeller ve parametrik olmayan modeller olmak üzere iki genel gruba ayrılabilir:

1. **Parametrik modeller:** Girdi ile çıktı arasındaki ilişkiyi bazı değişkenlerin belirlenmediği cebirsel eşitlikleri kullanarak açıklar. Bu belirlenmemiş değişkenler girdi örnekleri sağlanarak belirlenir. Parametrik modelleme bazen kullanılsa da veri hakkında çok fazla bilgi gerektirdiği için gerçek yaşamla ilgili sorunlarda kullanışlı olmayabilir.
2. **Parametrik olmayan modeller:** Bu modeller veri madenciliği için daha uygundur, çünkü bu modeller veriyi temel alır. Burada modeli belirlemek için hiçbir eşitlik kullanılmaz. Bu, modelleme işleminin eldeki veriye uyumlu hale getirilebileceği anlamına gelir. Parametrik olmayan yöntemlerde veriye göre bir model oluşturulur.

Parametrik ve parametrik olmayan modeller karşılaştırıldığında;

- Parametrik modellemede başlangıçta belirli bir model varsayılır. Parametrik olmayan modellemede ise girdiye göre bir model oluşturulur.
- Parametrik modellemede, modelleme işleminden önce veri hakkında çok fazla bilgi gerekir; parametrik olmayan modellemede ise modelleme işleminin kendisi için girdi olarak çok miktarda veri gerekir (12).

2.1.6. Veri madenciliği modelleri

Veri madenciliğinde kullanılan modeller kestirim modelleri ve tanımlayıcı modeller olmak üzere iki ana başlık altında incelenmektedir.

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

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

Veri madenciliği modellerini gördükleri işlevlere göre,

- Kümeleme *(Clustering),*
- Birliktelik Kuralları *(Association Rules)* ve Ardışık Zamanlı Örüntüler *(Sequential Patterns),*
- Sınıflama *(Classification*) ve Regresyon *(Regression)*

olmak üzere üç ana başlık altında incelemek mümkündür. Sınıflama ve regresyon modelleri kestirim; kümeleme, birliktelik kuralları ve ardışık zamanlı örüntü modelleri ise tanımlayıcı modellerdir (1).

2.1.6.1. Kümeleme modelleri

Kümeleme modellerinde amaç, küme üyelerinin birbirlerine çok benzediği ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veri tabanındaki kayıtların bu farklı kümelere bölünmesidir.

Bu modellerde başlangıçta sınıflama modelinde olan veri sınıfları yoktur. Verilerin herhangi bir sınıfı bulunmamaktadır. Sınıflama modelinde verilerin sınıfları bilinmekte ve yeni bir veri geldiğinde bu verinin hangi sınıftan olabileceği tahmin edilmektedir. Oysa kümeleme modelinde sınıfları bulunmayan veriler gruplar halinde kümelere ayrılırlar. Bazı uygulamalarda kümeleme modeli sınıflama modelinin bir ön işlemi gibi görev alabilmektedir (14).

Veri tabanlarında toplanan veri miktarının artmasıyla orantılı olarak kümeleme analizi son zamanlarda veri madenciliği araştırmalarında aktif bir konu haline gelmiştir. Kümeleme analizleri veri madenciliği, istatistik, biyoloji ve makine öğrenimi gibi pek çok alanda kullanılmaktadır.

Literatürde pek çok kümeleme algoritması bulunmaktadır. Kullanılacak olan kümeleme algoritmasının seçimi veri tipine ve amaca bağlıdır. Genel olarak başlıca kümeleme yöntemleri su şekilde sınıflandırılabilir (15);

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

2.1.6.2. Birliktelik kuralları ve ardışık zamanlı örüntüler

Bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi müşteriye daha fazla ürünün satılmasını sağlama yollarından biridir. Satın alma eğilimlerinin tanımlanmasını sağlayan birliktelik kuralları ve ardışık zamanlı örüntüler pazarlama amaçlı olarak pazar sepeti analizi *(Market Basket Analysis)* adı altında veri madenciliğinde yaygın olarak kullanılmaktadır. Bununla birlikte bu teknikler tıp, finans ve farklı olayların birbirleri ile ilişkili olduğunun belirlenmesi sonucunda değerli bilgi kazanımının söz konusu olduğu ortamlarda da önem taşımaktadır.

Birliktelik kuralları aşağıda sunulan örneklerde görüldüğü gibi eş zamanlı olarak gerçekleşen ilişkilerin tanımlanmasında kullanılır.

- Müşteriler bira satın aldığında, % 75 olasılıkla patates cipsi de alırlar,
- Düşük yağlı peynir ve yağsız yoğurt alan müşteriler, % 85 olasılıkla diyet süt de satın alırlar.

Ardışık zamanlı örüntüler ise aşağıda sunulan örneklerde görüldüğü gibi birbirleri ile ilişkisi olan ancak birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılmaktadır.

- X ameliyatı yapıldığında, 10 gün içinde % 45 olasılıkla Y enfeksiyonu oluşacaktır,
- İMKB endeksi düşerken A hisse senedinin değeri % 15'den daha fazla artacak olursa üç iş günü içerisinde B hisse senedinin değeri % 60 olasılıkla artacaktır,
- Çekiç satın alan bir müşteri ilk üç ay içerisinde % 15, bu dönemi izleyen üç ay içerisinde % 10 olasılıkla çivi satın alacaktır.

2.1.6.3. Sınıflama ve regresyon modelleri

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen iki veri analiz yöntemidir. Sınıflama kategorik değerleri tahmin ederken, regresyon sürekli değerlerin tahmin edilmesinde kullanılır (15).

Mevcut verilerden hareket ederek geleceğin tahmin edilmesinde faydalanılan ve veri madenciliği teknikleri içerisinde en yaygın kullanıma sahip olan sınıflama ve regresyon modelleri arasındaki temel fark, tahmin edilen bağımlı değişkenin kategorik veya sayısal bir değere sahip olmasıdır. Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler:

- Karar ağaçları,
- Yapay sinir ağları,
- Genetik algoritmalar,
- K-en yakın komşu,
- Bellek temelli nedenleme,

- Naïve-Bayes,
- Lojistik regresyon

olarak sıralanabilir.

2.1.7. Makine öğrenimi (Machine learning)

Makine öğrenimi metotları geçmişteki verileri kullanarak veriye en uygun modeli bulmaya çalışan metotlardır. Bu metotlar yeni gelen verileri de bu modele göre analiz ederler ve sonuç üretirler. Makine öğreniminin ilgilendiği konular aşağıdaki gibi sınıflandırılabilir (16):

- Sınıflandırma
- Kümeleme
- Regresyon
- Özellik seçimi/çıkarımı

Veri madenciliği; makine öğrenimi, istatistik, veri tabanı yönetim sistemleri, veri ambarlama, koşut programlama gibi farklı disiplinlerde kullanılan yaklaşımları birleştirmektedir. Makine öğrenimi, istatistik ve VM arasında yakın bir bağ bulunmaktadır. Bu üç disiplin veri içindeki ilginç düzenlilikleri ve örüntüleri bulmayı amaçlar. Makine öğrenimi yöntemleri, VM algoritmalarında kullanılan yöntemlerin çekirdeğini oluşturur. Makine öğreniminde kullanılan karar ağacı, kural tümevarımı pek çok VM algoritmasında kullanılmaktadır. Makine öğrenimi ile VM arasında benzerliklerin yanı sıra farklılıklar da göze çarpmaktadır. Öncelikle VM algoritmalarında kullanılan örneklem genişliği, makine öğreniminde kullanılan veri boyutuna nazaran çok büyüktür. Genellikle makine öğreniminde kullanılan örneklem genişliği 100 ile 1000 arasında değişirken, VM algoritmaları milyonlarca gerçek dünya nesneleri üzerinde uğraşmaktadır ki, bunların karakteristiği boş, artık, eksik, gürültülü değerler olarak belirlenebilir. Aynı zamanda VM algoritmaları bilgi keşfetmeye uygun nesne niteliklerinin elde edilme sürecindeki karmaşıklıkla baş etmek zorundadır (17).

Popüler makine öğrenimi metotları aşağıdaki gibidir:

1. Lojistik regresyon
2. Linear diskriminant analizi ve Fisher kriteri
3. K-en yakın komşuluk tekniği
4. Sınıflama ve regresyon ağaçları
5. Bagging
6. Boosting
7. Random Forest
8. Destek vektör sistemleri
9. Yapay sinir ağları
10. Nearest shrunken centroids
11. Naïve Bayes

2.2. K-En Yakın Komşuluk Yöntemi

Sınıflandırma tekniklerinden olan k-en yakın komşu yöntemi mesafeye dayalı olarak sınıflandırma yapan çok bilinen bir algoritmadır. Parametrik olmayan bu yöntem en basit ve yorumlanması kolay denetimli makine öğrenimi algoritmalarından biridir (18).

K-en yakın komşuluk yöntemi, n boyutlu özellik uzayında nesneleri sınıflandırmak ya da tahmin etmek için en yakın komşu örneklerini kullanır. K-en yakın komşu yönteminde sınıflandırma yapabilmek için kaç adet en yakın komşu sayısının katılacağı, k gibi bir pozitif tam sayı ile belirtilir. Eğer k=1 ise sınıflandırmaya çalıştığımız nesne en yakın komşusunun bulunduğu sınıfa dahil olacaktır. Bu yöntem kesitim için de kullanılmaktadır.

En yakın komşuların belirlenmesinde seçilen örnek ile eğitim kümesindeki örnekler arasındaki uzaklık ölçümü yapılır. Uzaklık mesafeleri en kısadan en uzağa doğru sıralanır, bu sıralama aynı zamanda seçilen örneğe en yakın komşudan en uzak komşuya olan sıralamayı da gösterir.

Yöntemde, sınıflandırma yapılacak verilerin öğrenme kümesindeki normal davranış verilerine benzerlikleri hesaplanarak; en yakın olduğu düşünülen k verinin ortalamasıyla, belirlenen eşik değere göre sınıflara atamaları yapılır. Yani

sınıflandırmada kullanılan bu algoritmaya göre sınıflandırma sırasında çıkarılan özelliklerden (feature extraction), sınıflandırılmak istenen yeni bireyin daha önceki bireylerden k tanesine yakınlığına bakılmaktadır. Önemli olan, her bir sınıfın özelliklerinin önceden net bir şekilde belirlenmiş olmasıdır. Yöntemin performansını;

- K en yakın komşu sayısı,
- Eşik değer,
- Benzerlik ölçütü etkilemektedir (19).

Buradaki k, 1'den büyük ve genelde tek sayı olarak seçilen bir tam sayıdır. K-en yakın komşuluk yönteminde sadece bir tane en büyük benzeme değerine değil, k tane en büyük benzeme değerine bakılarak sonuca ulaşılır. Seçilmiş olan k değerinin küçük olması durumunda birbirine benzerlikleri yüksek olan kayıtlar bir sınıfa sokulurken, k değerinin büyük seçilmesi birbirine benzemeyen kayıtların aynı sınıfa sokulması hatasını ortaya çıkarabilir.

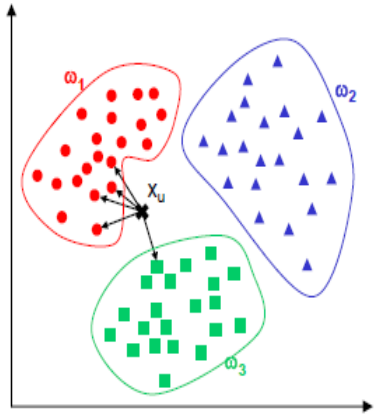
Bu yöntemin avantajları arasında;

- K-en yakın komşunun ortalaması alındığı için gürültülü veriden az etkilenmesi,
- Uygulaması ve anlaşılmasının kolay olması,

Dezavantajları olarak;

- En yakın komşuların sayısı olan k parametresinden,
- Seçilen uzaklık ölçütünden oldukça etkilenmesi,
- Eğitim verisinin büyük olduğu durumlarda verimli olması sayılabilir.

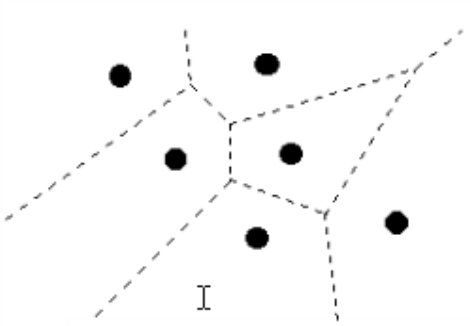
Örneğin, w_1, w_2 ve w_3 olmak üzere üç adet sınıf olduğu varsayılsın ve bilinmeyen bir x_u örneği sınıflandırılmaya çalışılsın. k=5 değeri için 5 en yakın komşu incelendiğinde 4 adet komşunun w_1 sınıfında olduğu, 1 adet komşunun ise w_3 sınıfında olduğu görülmektedir. Baskın olan taraf w_1 olduğu için x_u örneği w_1 sınıfına dahil olur (Şekil 3).



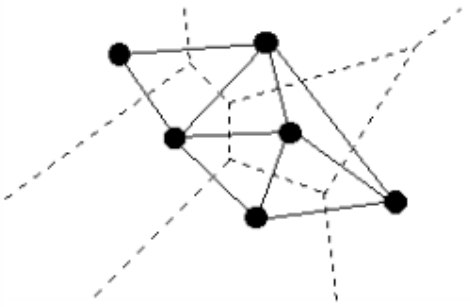
Şekil 3. K-en yakın sınıflandırma örneğine ait grafik (20).

2.2.1. Voronoi çizeneği yaklaşımı

Bir noktaya en yakın bölgelere bölümlenmesi ile elde edilen çizeneğe Voroni çizeneği denmektedir. Noktaları birleştiren doğru parçalarının ortalarından geçen dikmeler yardımıyla bu çizenek elde edilebilir (Şekil 4) (Şekil 5).

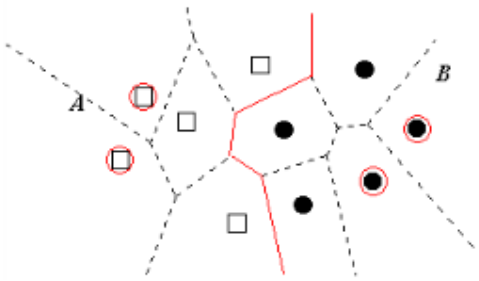


Şekil 4. Voronoi çizeneği (21).



Şekil 5. Voronoi çizeneği (21).

En yakın komşu kuralı ile sınıfı belirsiz bir örüntü çizenekteki bölgesine göre sınıflandırılabilir. Sınıflandırma sırasında değişik sınıflara ait bölgeler arası sınırlar önem kazanır (Şekil 6) (21).



Şekil 6. Voronoi çizeneği ile en yakın komşu ilişkisi (21).

2.2.2. Benzerlik ve uzaklık ölçütleri

Benzerlik, iki özellik veya nesne arasındaki ilişkinin gücünü yansıtan sayısal bir büyüklüktür. Benzerliğin ölçümü oldukça zordur. Bu büyüklük genelde -1, +1 aralığındadır ve normalize edilerek 0, +1 aralığına çekilir (22).

Benzerlik ölçülebilirse:

- Bir nesne diğerinden ayırtedilebilir,
- Benzerlik veya benzemezlik durumlarına göre nesneler gruplandırılabilir,
- Nesneleri grupladıktan sonra her grubun sahip olduğu özellikler daha iyi anlaşılabilir,
- Grupların davranışları açıklanabilir,
- Gruplama aynı zamanda verinin daha iyi organize edilebilmesini ve depolandığı yerden daha kolay çekilmesini sağlar,
- Yeni bir nesneyi oluşturulan gruplara yerleştirmek kolay olur,
- Yeni nesnenin davranışını tahmin etmek kolaylaşır,
- Eldeki veriler arasındaki ilişkiler oluşturulabilir.

Uzaklık ise benzemezliği ölçer. Benzemezlik ayrıca iki nesne arasındaki uyumsuzluğun bir ölçüsü olarak da düşünülebilir. Bu özellikler nesne için özellikler uzayında koordinat değerleri olarak da kullanılabilir. $d(i, j)$, i ve j noktaları

arasındaki uzaklığı göstermek üzere uzaklık aşağıdaki ilk üç durumu sağlayan sayısal bir değişkendir:

1. $d(i, j) \geq 0$; uzaklık her zaman pozitif veya sıfırdır,
2. $d(i, j) = 0$; uzaklık sadece $i = j$ olduğunda sıfırdır,
3. $d(i, j) = d(j, i)$; uzaklık simetriktir,
4. $d(i, k) \geq d(i, j) + d(j, k)$; uzaklık üçgen eşitsizliğini sağlar (18).

Uzaklık yukarıdaki dört koşulu da sağlarsa metrik olarak adlandırılır. Bu yüzden her metrik uzaklıktır ama her uzaklık metrik değildir.

Farklı veri tiplerine göre değişik uzaklık ölçütleri tanımlanmıştır. Sayısal değişkenler için uzaklık ölçütleri genel olarak d boyutlu örüntüler için kullanılan genel bir metrik sınıfı olan Minkowski metriğinden türemişlerdir. Minkowski uzaklığının genel formu aşağıda verilmiştir:

$$d(i, j) = \left[\left| x_{i1} - x_{j1} \right|^m + \left| x_{i2} - x_{j2} \right|^m + \cdots + \left| x_{ip} - x_{jp} \right|^m \right]^{/m} \quad (1)$$

Minkowski uzaklığında m değeri 2 alınırsa formül Öklid uzaklığı formuna dönüşür. Öklid uzaklığı en sık kullanılan uzaklık ölçüsüdür. Öklid uzaklığı aşağıdaki gibi tanımlanır:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \quad (2)$$

Bir diğer uzaklık ölçüsü olan Manhattan (City Blok) uzaklığı yine Minkowski uzaklığından türetilmiştir. Minkowski uzaklıklığı formülünde $m = 1$ alındığında uzaklık, Manhattan uzaklık ölçüsüne dönüşür:

$$d(i, j) = \left(\left| x_{i1} - x_{j1} \right| + \left| x_{i2} - x_{j2} \right| + \cdots + \left| x_{ip} - x_{jp} \right| \right) \quad (3)$$

Chebyshev uzaklığı ise iki nesne koordinatları arasındaki farkın mutlak büyüklüğünü dikkate alır. Bu uzaklık ölçüsüne maksimum değer uzaklığı da denir. Bu uzaklık metriği hem sıralı hem de niceliksel değişkenler için kullanılabilir. Formülü aşağıdaki gibi tanımlanır:

$$d(i, j) = \max_m \left| x_{1m} - x_{2m} \right| \quad (4)$$

İki nesnenin uzaklığını hesaplamada kullanılan matematiksel yöntem nesneler sadece sayısal niteliklerden oluşuyorsa gayet açıktır: İki nesnenin sahip olduğu değerlerin farkı alınarak nesnelerin birbirlerinden uzaklığı bulunabilir. Ancak nitel özellikler mevcutsa bu nitelikler arasında da bir uzaklık kavramı oluşturmak gerekir. Kırmızı, yeşil ve mavi gibi değerleri barındıran bir nitelik için uzaklık nasıl hesaplanabilir? Bunun için en bilinen yaklaşım nitelik değeri aynı ise 0 farklı ise 1 değeri vermektir. Bazı durumlarda 0 ve 1 değerleri yeterli olmayabilir. Kırmızı ve sarı renkleri arasındaki uzaklığın, beyaz ve siyah renkleri arasındaki uzaklıktan daha küçük olması beklenebilir. Bu tip nitelikler için kendine özgü uzaklık metriği düşünmek gerekir (23). Bu ölçülerden en bilineni Hamming uzaklık ölçüsüdür. Metin sınıflandırma, örtüşme metriği gibi durumlarda bu ölçü kullanılır.

2.2.3. K-en yakın komşu sınıflandırması örneği

İki adet bağımsız değişken ve bu değişkenlerin sonuçlarına bağlı olarak yanlış/doğru değerlerini alan bir bağımlı değişken olduğu varsayılsın. Sınıflandırılmak istenen yeni değerler (3, 7) olarak seçilsin.

Tablo 1. Sınıflandırmada Kullanılacak Bağımlı ve Bağımsız Değişkenlere Ait Değerler.

X	Y	Sınıflandırma
3	3	Yanlış
7	4	Yanlış
3	4	Doğru
1	4	Doğru

Adım 1: k değerimiz 3 olarak seçilsin.

Adım 2: Eklenecek noktaya ait uzaklıklar hesaplanır (Öklid uzaklık ölçüsü kullanılmıştır).

Tablo 2. Sınıflandırılacak Yeni Değere Ait Öklid Uzaklıkları.

X	Y	(3, 7) koordinatları için Öklid uzaklıkları
3	3	$(3-3)^2+(7-3)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

Adım 3: Uzaklıklar sıralanır ve en küçük uzaklığa bağlı olarak en yakın komşular bulunur.

Tablo 3. En Küçük Uzaklığa Bağlı Olarak Bulunan En Yakın Komşular.

X	Y	(3, 7) koordinatları için Öklid uzaklıkları	Sıralama	3 en yakın komşu
3	3	$(3-3)^2+(3-7)^2=16$	3	Evet
7	4	$(7-3)^2+(4-7)^2=25$	4	Hayır
3	4	$(3-3)^2+(4-7)^2=9$	1	Evet
1	4	$(1-3)^2+(4-7)^2=13$	2	Evet

Adım 4: En yakın komşuların kategorileri belirlenir.

Tablo 4. En Küçük Uzaklığa Bağlı Olarak Bulunan En Yakın Komşuların Sınıfı.

X	Y	(3, 7) koordinatları için Öklid uzaklıkları	Sıralama	3 en yakın komşu	Sınıflama
3	3	$(3-3)^2+(7-3)^2=16$	3	Evet	Yanlış
7	4	$(7-3)^2+(4-7)^2=25$	4	Hayır	-
3	4	$(3-3)^2+(4-7)^2=9$	1	Evet	Doğru
1	4	$(1-3)^2+(4-7)^2=13$	2	Evet	Doğru

Adım 5: Yeni sınıflanacak olan noktanın sınıfı belirlenir. Yukarıdaki duruma göre 2 adet doğru kategorisi varken 1 adet yanlış kategorisi vardır. Bu durumda (3, 7) noktasının sınıfı doğru olarak belirlenir.

2.3. Yapay Sinir Ağları Yöntemi

Yapay sinir ağları (YSA), ilk olarak 1943 yılında McCulloch ve Pitts tarafından ortaya atılan, yapısal olarak biyolojik sinir sistemlerini temel alan mantıksal bir model ile gündeme gelmiştir. Ardından 1969 yılında Minsky ve Papert tarafından yayımlanan "Perceptrons" adlı kitapla olgunlaşan yapay sinir ağları kavramı, Teuvo Kohonen, Stephen Grossberg, James Anderson ve Kunihiko Fukushima gibi bilim adamlarının çalışmalarıyla daha da gelişmiştir. Seksenli yılların başlarında gelişen donanım teknolojilerine bağlı olarak ilk meyvelerini vermeye başlayan yapay sinir ağları, bugün birçok üniversitenin psikoloji, fizik, bilgisayar bilimleri, biyoloji gibi bölümlerinde yapılan araştırmalarda kullanılan çok önemli bir öğrenme algoritması olarak kullanılmaktadır (24).

Yapay sinir ağı, öğrenme, veriyi sınıflandırarak bilgiye çevirme ve eş zamanlı birden çok işlem yapabilme kabiliyetine sahip matematiksel bir modeldir. YSA beynin çalışma ilkelerinin bilgisayarlar üzerinde taklit edilmesi fikri ile ortaya çıkmış ve ilk çalışmalar beyni oluşturan biyolojik hücrelerin ya da nöronların matematiksel olarak modellenmesi üzerinde yoğunlaşmıştır. Ancak kendisinden esinlenilen biyolojik sinir ağlarının çalışma yönteminin mükemmelliğine, karmaşıklığına ve verimliliğine erişebilmesi mümkün olmamıştır. Yapay sinir ağlarının, karar hızı açısından insan beyni ile yarışabilecek aşamayı henüz katetmemiş olmalarına rağmen gün geçtikçe hızları artmaktadır (25).

YSA'lar bugün birçok alanda başarılı şekilde kullanılmaktadırlar. Uygulama alanları için bir sınır yoktur fakat kestirim, modelleme ve sınıflandırma gibi bazı alanlarda ağırlıklı olarak kullanılmaktadır. YSA'lar 1950'li yıllarda ortaya çıkmalarına rağmen, ancak 1980'li yılların ortalarında genel amaçlı kullanım için yeterli seviyeye gelmişlerdir (26). YSA'ların gerçek hayattaki yaygın uygulama alanlarına şu örnekler verilebilir:

Otomotiv sektörü: Yol izleme, rehberlik, yol koşullarına göre sürüş analizi,

Bankacılık: Kredi uygulamalarının geliştirilmesi, kredi kartı suçlarının tespiti, imza tanıma,

Uzay sanayisi: Uçuş simülasyonları, otomatik pilot uygulamaları, uzay mekiğinde manevra denetimi, uçaklarda titreşim seviyeleri ve sesin görüntülenerek motor sorunlarının erken uyarı sistemi,

Elektrik: Çiplerin bozulma analizi, non-lineer modellemeler,

Finans: Döviz kuru tahminleri, makro ekonomik tahminler,

Sağlık: Kanserin erken teşhis ve tedavisi, EEG, EKG analizleri, kan analizi, ilaç etkileri analizi, kalite artırımı, hastalıkların resimlerden tanınması, kanserin izlenmesi, olası kazalarda sakatlıklardan korunma, solunum hastalıklarının teşhisi,

Askeriye: Askeri uçaklarda uçuş yönlerinin belirlenmesi, silahların doğru yönlendirilmesi, mayın arama aletlerinde kullanım,

Endüstri: Ürünlerin tasarımı, ürünlerin kalite kontrolü, müşteri tahmini analizleri, konuşmayı yazıya çevirme.

Bu alanlar dışında sigortacılık, eğlence, üretim, petrokimya, robotik uygulamalarında da yapay sinir ağları kullanılmaktadır (27).

2.3.1. Yapay sinir ağı modelinin temel özellikleri

Yapay sinir ağları insanın idrak etmesi ve biyolojik nöron yapısının matematiksel modelinin aşağıdaki kurallar varsayılarak genelleştirilmesi sonucunda oluşturulmuştur:

- Bilgi işleme nöron (işlem elemanı) adı verilen birimlerde gerçekleşir,
- Sinyaller, bir nörondan diğerine bağlantılar aracılığıyla iletilir,
- Her bir bağlantının gönderilen sinyal ile çarpılan bir ağırlık değeri vardır,
- Her bir nöron, bir çıktı sinyali elde etmek için ağ girdisine (ağırlıklarla çarpılmış sinyallerin toplamı) bir aktivasyon fonksiyonu uygular. Bu, genellikle doğrusal olmayan bir fonksiyondur.

Herhangi bir yapay sinir ağı;

- Nöronlar arasındaki bağıntının bir modeli yani mimarisi ile,
- Bağlantılardaki ağırlıkların hesaplanması (bu hesaplama, eğitim kuralı ya da öğrenme algoritması olarak da adlandırılır) ile,
- Aktivasyon fonksiyonu ile tanımlanabilir (28).

Bir yapay sinir ağı, nöron, birim, hücre ya da düğüm olarak adlandırılan çok sayıdaki basit işlem birimlerinden oluşur. Her bir nöron, diğer bir nörona belli bir ağırlık değerine sahip olan haberleşme bağlantılarıyla bağlanır. Ağırlıklar, yapay sinir ağının bir problemi çözmesi için gerekli olan bilgiyi hazırlamaktadır (29). Yapay sinir ağlarında, bir girdi dizisi ağa girdikten sonra, birinci katmandaki her bir nöron girdi dizisinin bir elemanını alır. Her nöron, analiz etmek istedigimiz ya da bir tahmin edici olarak kullanmak istedigimiz bir numunenin bir özelliğine ya da bir karakteristiğine kodlanır veya karşılık gelir. Birimler katmanlar biçiminde organize olurlar. Katmandaki diğer nöronlarla paralel bir şekilde ağırlıklarla çarpılan girdiler aktivasyon fonksiyonunda işlenir ve son katmandaki nöronlara bir tek çıktı olarak iletilir. Elde edilen sonuç, girdinin niteliklerini temsil eden bir çıktı dizisidir. Zamanla girdiler ve uyarlanabilir ağırlıklar değiştiği için ağ buna adapte olur ve öğrenir.

Herhangi bir yapay sinir ağının üç temel özelliği nöronlar, ağ mimarisi ve öğrenme algoritması ya da eğitim kurallarıdır (30).

Yapay sinir ağları kullanılarak yapılan modeller, biyolojik sinir ağlarının çalışma biçimlerinden esinlenerek oluşturuldukları için biyolojik sinir ağlarının üstünlüklerine sahiptir. Bu üstünlüklerin bir kısmı yapay sinir ağları fikrinin de ortaya çıkış sebepleridir ve şu şekilde özetlenebilir:

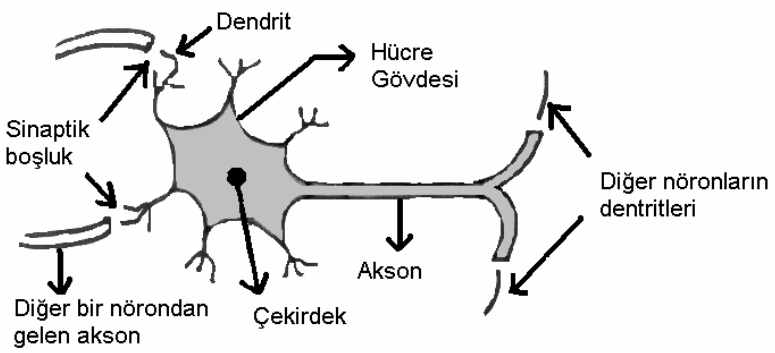
- Doğrusal olmayan yapı
- Paralellik
- Öğrenme
- Hafıza
- İlişkilendirme
- Sınıflandırma
- Genelleme
- Tahmin
- Özellik belirleme
- Hata toleransı
- Uyarlanabilirlik
- Eksik verilerle çalışma
- Sınırsız sayıda değişken ve parametre kullanma

Yapay sinir ağlarının yukarıda belirtilen avantajları geniş uygulama alanları bulmasını sağlamaktadır. Ancak yapay sinir ağlarının göz önünde bulundurulması gereken bazı dezavantajları da bulunmaktadır. Bunlar arasında en önemlisi, geniş veri seti gereksinimidir. Yapay sinir ağlarının eğitilmesine ve test edilmesine yetecek genişlikte veri setine ihtiyaç duyulur. Bununla birlikte, yeterli veri seti genişliği için kesin bir kriter yoktur, bu kriter uygulamaya bağlı olarak değişir. Dezavantaj sayılabilecek diğer bir nokta ise basit olarak görülebilecek modelleme yapılarına rağmen uygulamanın zor ve karmaşık olabilmesidir.

2.3.2. Nöronun biyolojik yapısı ve nöron modeli

Biyolojik sinir sistemi, bilgiyi alan, yorumlayan ve uygun kararı üreten bir merkez ve bu merkezin kontrolünde bulunan alıcı ve tepki sinirlerinden oluşmaktadır. Sinir hücrelerine tıp terminolojisinde nöron denmektedir. Nöron, sinir sisteminin temel birimidir ve gövde, gövdeye giren sinyal alıcılar (dentrit), gövdeden çıkan sinyal ileticiler (akson) olmak üzere başlıca üç kısımdan oluşmaktadır.

Dendritler üzerinden alınan girişler soma tarafından işlenir. Nörondaki sinyalleri taşıyan uzun bir sinirsel bağlantı halindeki akson işlenen girişleri çıkışa aktarır. Akson dendrit bağlantısı ise sinaps olarak adlandırılır. Sinaps, nöronlar arasındaki elektrokimyasal bağlantıyı sağlamaktadır. Şekil 7'de basit bir nöron hücresi görülmektedir (25).



Şekil 7. Biyolojik nöronun yapısı (29).

2.3.3. Biyolojik sinir hücresinin ve yapay sinir hücresinin bağdaştırılması

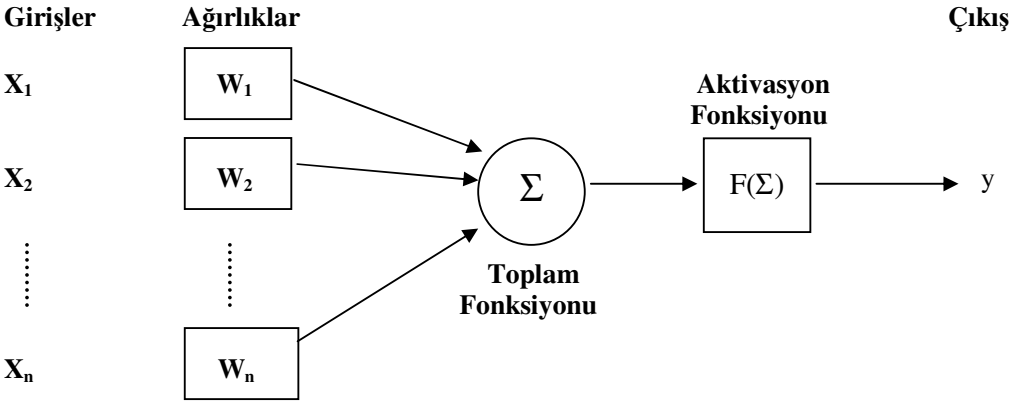
Yapay sinir ağları, birbirine bağlı doğrusal ve/veya doğrusal olmayan birçok elemandan oluşmaktadır. Biyolojik sinir sistemi ile yapay sinir sistemi arasındaki benzerlikler aşağıdadır (Tablo 5) (25).

Tablo 5. Biyolojik Sinir Sistemi ile Yapay Sinir Sistemi Arasındaki Benzerlikler (31).

Biyolojik Sinir Sistemi	**Yapay Sinir Ağı**
Nöron	Algılayıcı (Perceptron)
Dentrit	Toplama işlevi
Hücre gövdesi	Aktivasyon işlevi
Aksonlar	Algılayıcı çıkışı
Sinapslar	Ağırlıklar

Yapay sinir hücresi gerçek biyolojik hücreyle aynı ilkelere dayandırılmaya çalışılmıştır (28).

Şekil 8'de görülen yapay sinir hücresinin dendritleri x_n ve her bir dendritin ağırlık katsayısı (önemlilik derecesi) w_n ile belirtilmiştir. Böylece x_n girdi sinyallerini, w_n ise o sinyallerin ağırlık katsayılarının değerlerini taşımaktadır. Çekirdek yani toplam fonksiyonu ise tüm girdi sinyallerinin ağırlıklı toplamlarını elde etmektedir. Tüm bu toplam sinyal $F(\sum)$ ile gösterilmiş ve sinapsise yani aktivasyon fonksiyonuna girdi olarak yönlendirilmiştir. Sinapsis üzerindeki aktivasyon fonksiyonundan çıkan sonuç sinyali y ile belirtilmiş ve diğer hücreye beslenmek üzere yönlendirilmiştir.



Şekil 8. Yapay sinir hücre yapısı.

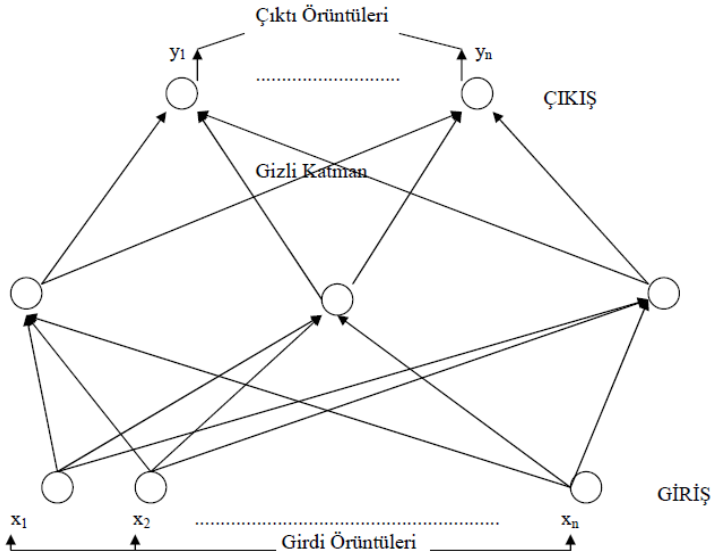
Yapay sinir hücresinin görevi kısaca, x_n girdi örüntüsüne karşılık y çıktısı sinyalini oluşturmak ve bu sinyali diğer hücrelere iletmektir. Her x_n ile y arasındaki korelasyonu temsil eden w_n ağırlıkları, her yeni girdi örüntüsü ve çıktı sinyaline göre tekrar ayarlanır. Bu ayarlama süreci öğrenme olarak adlandırılır. Öğrenmenin tamamlandığının belirtilebilmesi için girdi örüntüleri, w_n ağırlıklarındaki değişim durağan olana dek sistemi beslemektedir. Durağanlık sağlandığı zaman hücre öğrenmesini tamamlamıştır.

Yapay sinir ağları, görevi yukarıdaki biçimde belirtilen yapay sinir hücrelerinin birleşiminden oluşan katmanlı yapının tümü olarak nitelendirilir. Böylece *"n"* adet yapay sinir hücresinin katmanlı yapısıyla yapay sinir ağı modeli kurulmuş olunmaktadır (32).

2.3.4. Yapay sinir ağı ve yapı taşları

Nöronların katmanlar içindeki yerleşimleri ve diğer katmanlardaki nöronlarla olan bağlanma şekilleri ağ mimarisi olarak adlandırılmaktadır. Ağ yapısı katmanlar şeklinde yerleşmiş nöronlardan oluşmaktadır. Nöronlar, girdileri diğer işlem elemanlarından gelen heyecanlandırıcı ya da engelleyici ağırlıkları bağlantılar yoluyla alırlar. Heyecanlandırıcı ağırlıklar genellikle pozitif, engelleyici ağırlıklar ise negatif değere sahiptir. Bütün sistemin davranışını belirleyen temel faktörler aktivasyon fonksiyonları ve sinyallerin gönderildiği bağlantılar üzerindeki ağırlıklardır.

Genel olarak bir sinir ağında üç ayrı katman bulunmaktadır. Bu katmanlar giriş katmanı, gizli katman ve çıkış katmanıdır. Yapay sinir ağında ilk katman girdi katmanıdır ve dışarıdan gelen verilerin yapay sinir ağına alınmasını sağlar. Son katman ise bilgilerin dışarıya iletildiği çıktı katmanıdır. Girdi ve çıktı katmanlarının arasında bulunabilecek bir ya da daha fazla sayıdaki katmana ise gizli katman adı verilir. Gizli katmanlar giriş uzayını keyfi bölgelere ayırarak karmaşık problemlerin çözümünde gerekli gücü sağlar. Bir yapay sinir ağında gizli katman olması gerekmediği gibi birden fazla gizli katman da bulunabilir. Aşağıdaki şekilde bir yapay sinir ağının yapısı görülmektedir (Şekil 9).



Şekil 9. Sinir ağının genel görünümü (25).

YSA'ların temel yapı taşı canlılardaki sinir sisteminde olduğu gibi sinir hücreleridir (nöron). YSA'nın yapı taşı olan yapay sinir hücrelerinin beş temel bölümü vardır:

1. Girdiler
2. Ağırlıklar
3. Toplama fonksiyonu
4. Aktivasyon fonksiyonu
5. Çıktılar

Girdiler: Yapay sinir ağına dışarıdan verilen bilgilerdir.

Ağırlıklar: Hücreler arasındaki bağlantıların sayısal değeridir. Bir hücrenin üzerine gelen bilginin değerini ve hücre üzerindeki etkisini gösterir.

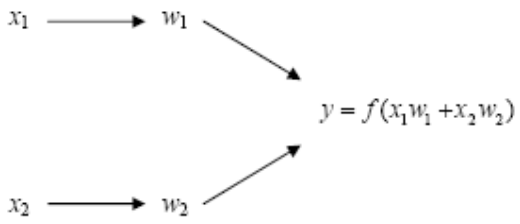
Toplama fonksiyonu: Hücreye gelen net girdinin hesaplanmasını sağlayan fonksiyondur. En yaygın kullanım şekli her girdi değerinin kendi ağırlığıyla çarpılarak toplanmasıdır.

Aktivasyon fonksiyonu: Bu fonksiyon hücreye gelen net girdinin işlenmesiyle hücrenin bu girdiye karşılık üreteceği çıktıyı belirlemesini sağlar. En yaygın olarak sigmoid fonksiyonu kullanılmaktadır.

Çıktılar: Aktivasyon fonksiyonundan çıkan değer nöronun çıktı değeri olmaktadır. Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilir ister yeniden ağın

içinde kullanılabilir. Nöronun bir çıktısı olmasına rağmen bu çıktı istenilen sayıda nörona bağlı olabilir.

Her bir katman özellikli düğümlerden oluşmaktadır. Bu özellikli düğümlere algılayıcı denir. Algılayıcı, biyolojik bir sinir ağındaki sinir hücrelerine karşılık gelmektedir. Bir algılayıcı yapay sinir ağının en küçük birimidir. Algılayıcının genel işleyişi Şekil 10'da gösterilmektedir.



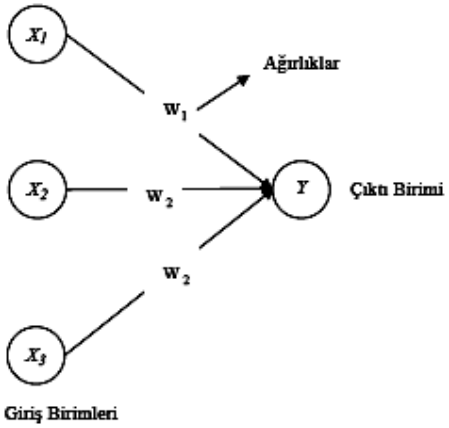
Şekil 10. Algılayıcının genel işleyişi (25).

x_1 ve x_2 değerleri giriş değerleri ve sırasıyla bu giriş değerlerinin ağırlıkları olan w_1 ve w_2 değerleri birbirleriyle çarpılmaktadır. Bu değerler toplanarak *"etkinleştirme işlevi"* (activation function) diye de anılan bir işlevden geçirilerek çıkış değeri elde edilmektedir. Bu işlevin *"etkinleştirme işlevi"* diye anılmasının sebebi, biyolojik sinir hücresinin etkin olabilmesi (iletim yapabilmesi) için gerekli olan eşik değerini aşabilecek değeri üreten bir işlev gibi düşünülmesidir.

Her nöronun bir iç durumu vardır ve bu iç durum aktivasyon ya da aktivasyon düzeyi olarak adlandırılır. Bu düzey, alınan giriş değerlerinin bir fonksiyonudur. Herhangi bir nöron, kendi aktivasyonunu genelde sinyal şeklinde diğer nöronlara gönderir. Bu sinyal birden fazla nörona aynı anda gönderilebilir.

Örnek olarak Şekil 11'de gösterilen bir Y nöronu düşünülsün. Bu nöron X_1, X_2 ve X_3 nöronlarından giriş sinyallerini alır. Bu nöronların aktivasyonları yani çıkış sinyalleri, sırasıyla x_1, x_2 ve x_3'tür. Bağlantılar üzerindeki ağırlıklar X_1, X_2 ve X_3 nöronlarından Y nöronuna doğru sırasıyla w_1, w_2 ve w_3'tür. Ağ girişi olan y_in değeri X_1, X_2 ve X_3'den Y'ye giden ağırlıklı sinyallerin toplamıdır. y_in değeri 5. eşitlikteki gibi hesaplanır.

$$y\_in = w_1 x_1 + w_2 x_2 + w_3 x_3 \quad (5)$$



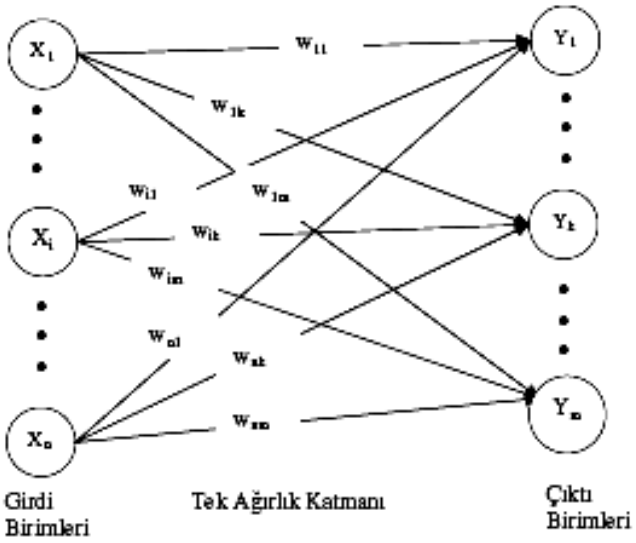
Şekil 11. Basit bir yapay nöron (29).

Y nöronunun aktivasyonu *y*, ağa giriş değerlerinin bir fonksiyonu olarak tanımlanır (29).

$$y = f(y\_in) \quad (6)$$

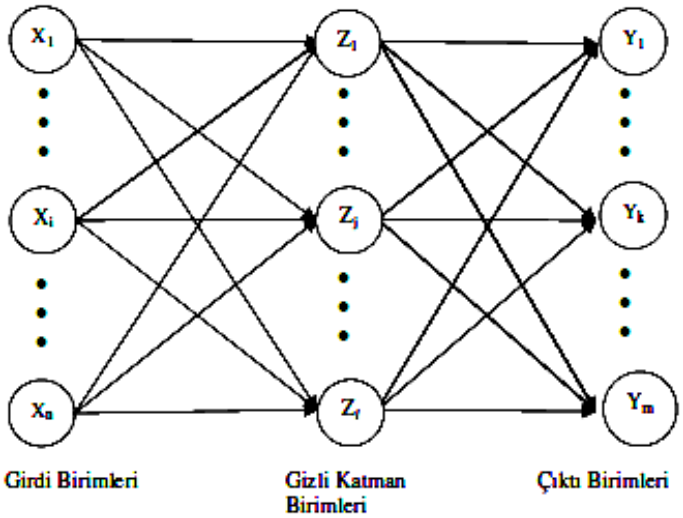
Sinir ağları tek katmanlı ya da çok katmanlı olarak sınıflandırılırlar. Katman sayısını belirlerken, girdi birimi bir katman olarak sayılmaz; çünkü bunlar üzerinde hiçbir hesaplama işlemi yapılmaz. Bir ağ içindeki katman sayısı, nöronları bağlayan ağırlıklı bağlantı sayısına eşittir (25).

Tek katmanlı yapay sinir ağlarında bir tane ağırlıklı bağlantı katmanı bulunur. Çoğu kez birimler, sinyalleri alan girdi birimi ve ağın cevabının alınacağı çıktı birimi olmak üzere ikiye ayrılmaktadır. Tipik bir tek katmanlı YSA Şekil 12'de verilmiştir.



Şekil 12. Tek katmanlı bir yapay sinir ağı modeli (30).

Çok katmanlı yapay sinir ağlarında ise girdi birimleri ile gizli birimlerin arasında bir ya da birden fazla katman bulunmaktadır. Genellikle gizli ve çıktı birimleri arasında ağırlıklı bağlantı katmanı bulunan bu tip ağ mimarileri tek katmanlı ağ mimarilerine göre daha karmaşık problemleri çözebilir. Şekil 13'te iki katmanlı bir yapay sinir ağı görülmektedir.



Şekil 13. Çok katmanlı bir yapay sinir ağı modeli (30).

2.3.5. Yapay sinir ağlarının sınıflandırılması

Yapay sinir ağları işleyiş olarak benzer olmalarına rağmen herhangi bir tasarım ve işleyiş standardı bulunmamaktadır. Nöron dizilimlerine, nöronların ağırlıklarının düzenlenmesi için yapılan hesaplamaların türüne ve zamanına göre yapay sinir ağları üç ayrı dalda incelenebilir:

Yapay sinir ağları yapılarına göre;

1. **İleri beslemeli yapay sinir ağları:** İleri beslemeli yapay sinir ağlarında nöronlar arasında hiyerarşik bir yapı vardır ve bir katmandaki nöronlar sadece kendinden sonraki katmana veri iletir. Bu ağlarda nöronlar girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katmanlara bağ bulunmaktadır. Yapay sinir ağına gelen bilgiler giriş katmanına daha sonra sırasıyla ara katmanlardan ve çıkış katmanından işlenerek geçer ve daha sonra dış dünyaya çıkar.
2. **Geri beslemeli yapay sinir ağları:** Bu YSA modelinde ise bir nöron kendinden sonraki katmana veri ilettiği gibi kendinden önceki katmana veya kendi katmanına da veri iletebilir. Geri beslemeli yapay sinir ağlarında ileri beslemeli olanların aksine bir nöronun çıktısı sadece kendinden sonra gelen nöron katmanına girdi olarak verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir nörona girdi olarak bağlanabilir. Bu yapısı ile geri beslemeli yapay sinir ağları doğrusal olmayan dinamik bir davranış göstermektedir. Geri besleme özelliğini kazandıran bağlantıların bağlanış şekline göre aynı yapay sinir ağıyla farklı davranışta ve yapıda geri beslemeli yapay sinir ağları elde edilebilir.

Öğrenme algoritmalarına göre;

1. **Danışmanlı öğrenme:** Bu tip öğrenmede, yapay sinir ağlarına örnek olarak bir doğru çıktı verilir. İstenilen ve gerçek çıktı arasındaki farka (hataya) göre nöronlar arasındaki bağlantıların ağırlıkları, en uygun çıktıyı elde etmek için sonradan düzenlenebilir. Bu sebeple danışmanlı öğrenme algoritmasının bir "danışmana" ihtiyacı vardır. Widrow-Hoff tarafından geliştirilen delta kuralı, Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı ve geriye beslemeli öğrenme algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir (33).

2. **Danışmansız öğrenme:** Danışmansız öğrenmede ağ, girdi olarak verilen örnekten elde edilen çıktı bilgisine göre sınıflandırmayı kendi kendine geliştirmektedir. Bu öğrenme algoritmalarında hiçbir hedef vektörü verilmez. Öğrenme sürecinde sadece giriş bilgileri verilir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler oluşturmak üzere ayarlar. Kohonen'in kendi kendini düzenleyen haritalarını (Kohonen's self-organizing maps) ve adaptif rezonans teorisi (adaptive resonance theory) danışmansız öğrenmeye örnek olarak verilebilir (34).
3. **Destekleyici öğrenme:** Bu öğrenmede YSA'ya giriş verilerinin yanı sıra her veri setine ait çıkışa bir puan verilir. YSA, katsayılarını en yüksek puanı alacak şekilde düzenler. Bu öğrenme yaklaşımında ağın her tekrarlanması sonucunda elde ettiği sonucun iyi veya kötü olup olmadığına dair bir bilgi verilir. Ağ bu bilgilere göre kendini yeniden düzenler. Bu sayede ağ herhangi bir girdi dizisiyle hem öğrenerek hem de sonuç çıkararak işlemeye devam eder. Örneğin satranç oynayan bir yapay sinir ağı yaptığı hamlenin iyi veya kötü olduğunu anlık olarak ayırt edememesine rağmen yine de hamleyi yapar. Eğer oyunun sonuna gelindiğinde program oyunu kazandıysa yaptığı hamlelerin iyi olduğunu varsayacaktır ve bundan sonraki oyunlarında benzer hamleleri iyi olarak değerlendirerek oynayacaktır (35).

Öğrenme zamanına göre;

1. **Dinamik:** Dinamik öğrenme kuralı yapay sinir ağlarının çalıştığı süre boyunca öğrenmesini öngörerek tasarlanmıştır. Yapay sinir, eğitim aşaması bittikten sonra da daha sonraki kullanımlarında çıkışların onaylanmasına göre ağırlıklarını değiştirerek çalışmaya devam eder.
2. **Statik:** Statik öğrenme kuralıyla çalışan yapay sinir ağları kullanılmadan önce eğitilmektedir. Eğitim tamamlandıktan sonra ağ istenilen şekilde kullanılabilir. Ancak bu kullanım sırasında ağın üzerindeki ağırlıklarda herhangi bir değişiklik olmaz.

2.3.6. Aktivasyon fonksiyonları

Transfer fonksiyonu ya da etkinleştirme fonksiyonu olarak da geçen aktivasyon fonksiyonu, toplama fonksiyonundan elde edilen net girdiyi bir işlemden geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Hücre modellerinde hücrenin gerçekleştireceği işleve göre çeşitli tipte aktivasyon fonksiyonları kullanılabilir. Bu fonksiyonlar içinde en çok kullanılanı sigmoid fonksiyonlarıdır. Örneğin, eğer ağın bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, ortalamadan sapmanın öğrenilmesi isteniyorsa hiperbolik tanjant fonksiyon kullanılması önerilmektedir. Aktivasyon fonksiyonları bir YSA'da nöronun çıkış genliğini, istenilen değerler arasında sınırlar. Bu değerler genellikle [0,1] veya [-1,1] arasındadır. YSA'da kullanılacak aktivasyon fonksiyonlarının türevinin alınabilir olması ve süreklilik arz etmesi gereklidir. Lineer veya doğrusal olmayan aktivasyon fonksiyonlarının kullanılması YSA'ların karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır.

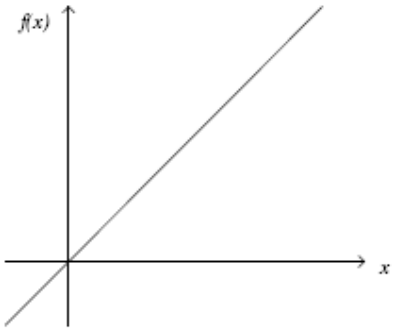
Toplama fonksiyonunun çıktısı aktivasyon formülünde girdi olarak kullanılır. Ancak bu girdileri belirli bir seviyenin üstünde tutmak için bir eşik değeri seçilmelidir. Toplama fonksiyonundan gelen değerler bu eşik değerinden yukarıda ise işleme tabi tutulur. Aktivasyon fonksiyonu girdileri algoritma ile gerçek bir çıktıya dönüştürür. Aktivasyon fonksiyonunda genel olarak türevi alınabilen fonksiyonlar kullanılır (36).

Bir yapay sinir ağında yapılması gereken en temel işlemler, ağırlıklı girdi değerlerinin toplanması ve bir çıktı ya da aktivasyon fonksiyonu uygulamasıdır. Aktivasyon fonksiyonunun doğru seçilmesi, ağın performansını önemli derecede etkiler. Aktivasyon fonksiyonlarının çeşitleri aşağıdaki gibi verilebilir.

2.3.6.1. Doğrusal aktivasyon fonksiyonu

Doğrusal bir problemi çözmek amacıyla kullanılan doğrusal hücre ve YSA'da ya da genellikle katmanlı YSA'nın çıkış katmanında kullanılan doğrusal aktivasyon fonksiyonu, hücrenin net girdisini doğrudan hücre çıkışı olarak verir (Şekil 14) (37).

Genellikle girdi değerleri için kullanılan aktivasyon fonksiyonu tüm x'ler için $f(x) = x$ olan özdeşlik fonksiyonudur.



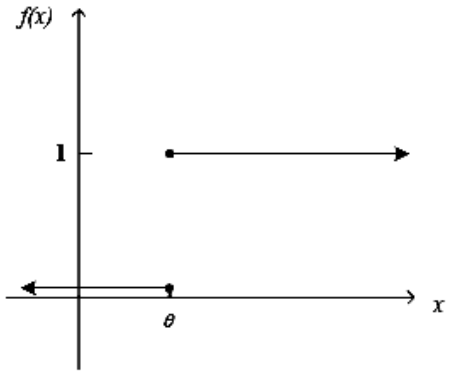
Şekil 14. Doğrusal aktivasyon fonksiyonu (29).

2.3.6.2. Eşik (Basamak) aktivasyon fonksiyonu

İkili adım fonksiyonu ya da Heaviside fonksiyonu olarak da bilinmektedir. Bu fonksiyon aşağıdaki formülle verilebilir:

$$f(x) = \begin{cases} 0, & x < \theta \; ise \\ 1, & x \geq \theta \; ise \end{cases} \tag{7}$$

Aşağıdaki eşik fonksiyonunun grafiği görülmektedir (Şekil 15).



Şekil 15. Eşik aktivasyon fonksiyonu (29).

2.3.6.3. Sigmoid aktivasyon fonksiyonu

Sigmoid fonksiyonları oldukça kullanışlı aktivasyon fonksiyonlarıdır. Sigmoid fonksiyonları içinde lojistik ve hiperbolik tanjant fonksiyonları en yaygın olarak kullanılanlarıdır. Özellikle yapay sinir ağı modellerinden biri olan ve uygulamada sıkça kullanılan geriye beslemeli ögrenme algoritmalarında bu fonksiyonların kullanımı diğerlerine göre daha avantajlıdır. Çünkü fonksiyonun belirli bir noktadaki değeri ile onun türevinin değeri arasındaki ilişki öğrenme zamanındaki hesap yükünü azaltmaktadır.

a) Lojistik fonksiyon

Lojistik fonksiyon, değerleri 0 ile 1 arasında değişen bir sigmoid fonksiyondur ve yapay sinir ağları için aktivasyon fonksiyonu olarak sıkça kullanılmaktadır. Fonksiyonun aralık değerini vurgulamak için bu fonksiyona ikili sigmoid adı verilmekle birlikte lojistik sigmoid adı da verilmektedir (28).

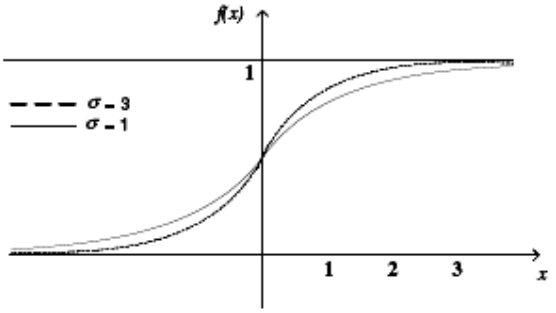
Şekil 16'da fonksiyonun adım parametresi olan α'nın farklı değerleri için ikili sigmoid eğrileri görülmektedir. Lojistik fonksiyon,

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \tag{8}$$

veya türevi olan,

$$f'(x) = \sigma f(x)\left[1 - f(x)\right] \tag{9}$$

formülleri ile hesaplanmaktadır.



Şekil 16. Lojistik sigmoid fonksiyonu (29).

Lojistik sigmoid fonksiyonu istenen değer aralığına göre ölçeklenebilir ve böylece probleme uygun bir fonksiyon haline gelebilir. En yaygın kullanılan aralık -1 ile 1 aralığıdır. Bu sigmoid fonksiyonu, iki kutuplu sigmoid olarak adlandırılır.

b) Hiperbolik tanjant fonksiyonu

Hiperbolik tanjant fonksiyonu, sigmoid fonksiyonunun biraz farklı şeklidir. Hiperbolik tanjant fonksiyonu istenilen çıkış aralığı -1 ile 1 arasında ise aktivasyon fonksiyonu olarak kullanılmaktadır. Bu fonksiyon,

$$h(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{10}$$

veya türevi olan,

$$h'(x) = \left[1 + h(x)\right]\left[1 - h(x)\right] \tag{11}$$

formülleri ile hesaplanmaktadır (38).

2.3.7. Tek katmanlı yapay sinir ağı modelleri

Tek katmanlı yapay sinir ağları, yapay sinir ağlarının önemli bir sınıfını oluştururken, aynı zamanda daha karmaşık yapıda bulunan çok katmanlı yapay sinir ağlarına da ışık tutarlar (38).

Tek katmanlı YSA; örnek sınıflandırma, tanıma, örnek ilişkilendirme ve bunun gibi diğer problemlerin çözümlenmesinde kullanılabilir.

Yapay sinir ağlarında bilgi, ağdaki bağlantıların ağırlıklarında depolanır. Öğrenme, sistemin bir bütün olarak istenilen işlevi yerine getirecek şekilde ağırlıklarının ayarlanması sürecidir. YSA'da toplam ağ hatası istenilen düzeye erişinceye kadar eğitim devam eder (38).

Tek katmanlı sinir ağlarının eğitilmesinde üç önemli yöntem aşağıdaki gibidir:

- Hebb kuralı
- Perseptron öğrenme kuralı
- Delta kuralı

Gerçek dünyada karşılaşılan birçok problem daha karmaşık mimarileri ve karmaşık eğitim kurallarını gerektirir ve genel olarak tek katmanlı yapay sinir ağları bu tip problemleri çözmede yeterli değildir. Ancak şartlar bu ağları kullanmak için elverişli ise doğru sonuçlar alınabilmesi mümkündür (39).

2.3.7.1. Hebb kuralı

Hebb kuralı, bir yapay sinir ağı için en eski ve en basit öğrenme kuralı olarak bilinir. Hebb, öğrenmenin sinaps uzunluklarını (ağırlıkları) değiştirerek meydana geleceğini önermiştir. Hebb'e göre eğer birbiri ile bağlı iki nöronun her ikisi de aynı zamanda "aktif" ise, bu nöronlara uygun ağırlıkların artırılması gerekmektedir. Benzer olarak, eğer her iki nöron aynı zamanda "pasif" ise ağırlıkların artırılması gerekir. Hebb eğitiminde iki nöron arasındaki bağlantı, bu nöronların öğrenmeleri sırasındaki aktivasyon değerlerinin arasındaki korelasyon miktarıyla orantılıdır (40).

2.3.7.2. Perseptron öğrenme kuralı

Perseptronlar, YSA'nın öğrenebilir niteliğini taşıyan ilk modelidir. Hebb kuralından daha yetenekli bir öğrenme kuralıdır. Perseptron tekrarlı öğrenme algoritmasıdır ve çözümün varlığı durumunda yakınsama niteliğine sahiptir. Bu, perseptron modelinin en önemli niteliklerinden biridir.

Rosenblatt (1962) ve Minsky-Papert (1969, 1988) tarafından çeşitli perseptron modelleri tanımlanmıştır. Orijinal perseptronlar duyumsal birimler, birleştirici birimler ve cevap birimleri olmak üzere nöronların üç durumuna sahiptirler. Örneğin, bir basit perseptron duyumsal ve birleştirici birimler için ikili aktivasyon, cevap birimi için ise +1, 0, veya -1 değerlerini üreten aktivasyon uygulayabilir.

Sınıflandırma problemlerinde Eşitlik 12'de verilen eşik değerli aktivasyon fonksiyonu kullanılır:

$$f(y\_in) = \begin{cases} -1, & y\_in < -\theta \; ise \\ 0, & -\theta \le y\_in \le \theta \; ise \\ 1, & y\_in > \theta \; ise \end{cases} \qquad (12)$$

Çıktı biriminin aktivasyonu $y = f(y\_in)$ şeklinde hesaplanır.

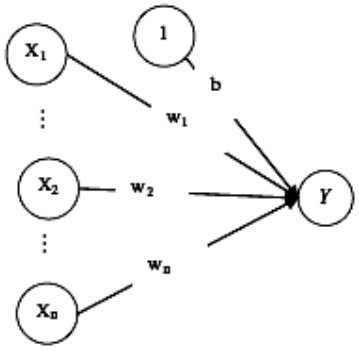
Birleştirici birimden cevap birimine giden bağlantıların ağırlıkları perseptron öğrenme kuralı ile ayarlanır. Her eğitim girişi için sinir ağı çıkış biriminin cevabını hesaplar. Daha sonra sinir ağı bu örnek için çıkış değeri ile hedeflenen çıkış arasındaki farkı karşılaştırarak bir hata oluşup oluşmadığını tespit eder. Yapay sinir ağı, hesaplanmış çıkış değeri "0" ve hedef değeri "-1" olan örnek için hatayı ayırt edemez, buna karşıt olarak hesaplanmış çıkış değeri "+1" ve hedef değeri "-1" olan örnek için hatayı ayırt edebilir. Bu durumlarda, hedef verinin işareti yönünde ağırlıkların işareti değiştirilmelidir. Bununla birlikte çıkış birimine "0" olmayan sinyaller gönderen bağlantıların ağırlıkları ayarlanmalıdır. Eğer belirli bir eğitim giriş örneğinde hata oluşuyorsa, ağırlıklar Eşitlik 13'teki gibi değiştirilmelidir.

$$w_i(yeni) = w_i(eski) + \alpha t x_i \tag{13}$$

Burada hedef değeri *t*, "+1" ya da "-1"dir ve α öğrenme oranı katsayısıdır. Eğer hata oluşmadıysa ağırlıklar değiştirilmemelidir. Eğitim işlemi hata oluşmayıncaya kadar devam etmelidir. Bu kuralın amacı, ağın tam olarak doğru cevap veremediği eğitim örnekleri için ağırlıkları ayarlamaktır. Ayrıca, eğitim sonunda bu ağ sınırsız sayıdaki eğitim adımları için ağırlıkların değerlerini bulmalıdır (28).

Sınıflandırma problemlerinde sinir ağının görevi tüm giriş örneklerinin belirli bir sınıfa ait olup olmadığını belirlemektir. Sınıfa ait olma çıkışın "+1" değerine, ait olmama ise çıkışın "-1" değerine uygun olmasıyla belirlenir. Sınıflandırma işlemi yapılabilmesi için ağ tekrarlı bir teknik ile eğitilir. Girdi ve hedefler ikili veya iki kutuplu olabilir. θ eşik değeri tüm birimler için değişmezdir. Sapma ve eşik değerinin her ikisinin aynı zamanda kullanılmasına ihtiyaç duyulmaktadır.

Şekil 17'de perseptronun mimarisi gösterilmiştir. Burada $X_1,\ldots,X_n$ girdi birimleri, *Y* çıktı birimi ve *l* sapma sinyalidir. *b* ise sapma ağırlığı olup w_i $(i = 1,\ldots,n)$ ağırlıklardır.



Şekil 17. Basit bir perseptron mimarisi (29).

2.3.7.3. Delta öğrenme kuralı

Widrow ve Hoff kuralı olarak da bilinen delta kuralı, Widrow ve Hoff tarafından 1960 yılında ortaya atılmış yinelemeli bir öğrenme sürecidir. Delta kuralında, tüm girdi numuneleri için çıktı ve hedef farkları karelerinin toplamının, başka bir ifadeyle, toplam hatanın küçültülmesi hedeflenmiştir. Amaç, tüm eğitim numunelerinin hatalarını en aza indirgemektir. Ağırlık düzeltmeleri, çok sayıdaki eğitim numunesi ile beraber biriktirilebilir ve bu yığın güncelleştirilmesi olarak adlandırılır (40).

2.3.8. Çok katmanlı yapay sinir ağları

Tek katmanlı ağların doğrusal olarak ayrılamayan problemlerin çözümünde başarısız oldukları görüldüğünde bilim adamları çok katmanlı YSA modellerini incelemişlerdir. Burada önemli aşamalardan biri bu tip ağlar için akıllı bir eğitim algoritması geliştirmekti. 1986 yılında Rumelhart, Hinton ve Williams tarafından bu gerçekleştirildi (41). Standart geriye yayılım (back-propagation) olarak adlandırılan bu eğitim metodu hata kareler toplamının geriye yayılım yöntemiyle küçültülmesi fikrine dayanır ve genelleştirilmiş delta kuralını kullanır. Dolayısıyla bu yöntem her adımda hatanın küçültülmesi için Widrow-Hoff eğitiminde olduğu gibi gradient azalış yöntemini kullanır. Bu durumda gizli katmanda doğrusal olmayan aktivasyon fonksiyonları, örneğin lojistik sigmoid fonksiyonu ve ona uygun olarak genelleştirilmiş delta kuralı uygulanmaktadır. Bu yöntem daha iyi tahmin yapmak, sınıflandırmak ve öngörü problemleri için büyük imkanlar sağlamaktadır (42).

2.3.8.1. Standart geriye yayılım ağ mimarisi

Geri yayılım ağlarında giriş, çıkış ve en az bir tane gizli katman yer almaktadır. Gizli katmandaki düğüm sayısı değişebilir ve düğüm sayısı artarsa ağın hatırlama yeteneği de artar. Ancak düğüm sayısının artması öğrenme süresini uzatır. Bunun tam tersi olarak düğüm sayısının azalmasıyla öğrenme süresi kısalır ancak hatırlama yeteneği azalır. Bir katmandaki düğümlerin her biri kendinden sonraki katmanda yer alan düğümlerin her birine bağlıdır. Ancak aynı katman içindeki düğümlerin hiçbiri bir diğerine bağlı değildir (43).

Geri yayılım ağında hatalar ileri beslemeli ağlar için kullanılan bağlantılar yardımıyla geriye doğru aktarılırlar. Bu yöntemde ağırlık ayarlamaları geriye doğru yapıldığı için geri yayılım olarak isimlendirilmektedirler.

Bu algoritmalarda ağ ağırlıkları öncelikle rastgele seçilir. Seçilen bu ağırlıklar yeni girişlerin çıkışlarının hesaplanmasında kullanılır. Bu ağda iki türlü bağlantı söz konusudur. İlk olarak ileri besleme işlemi ile ağırlıklar kullanılarak girişlere karşılık gelen çıkışlar elde edilir. İkinci olarak geri besleme ile elde edilen çıkışlarla beklenen çıkışlar arasındaki hata değerinin geri yayılması sağlanır. Bu geri yayılma işlemi en iyi çıkışı verecek en uygun ağırlığı belirleyebilmek için ağdaki tüm katmanlar için gerçekleştirilir. Bu işlem toplam hata minimuma düşürülünceye kadar devam ettirilir.

Standart geriye yayılım yöntemi üç aşamadan oluşmaktadır: ileri besleme, hatanın hesaplanması ve geriye yayılması, ağırlıkların güncellenmesi.

Geri yayılım ağ mimarisinde çeşitli algoritmalar kullanılmaktadır. Kullanılan algoritma çeşitleri aşağıdaki gibidir:

a) Momentum

Momentum geriye yayılımda ağırlık değişiminin yönü o anki eğimle bir önceki eğimin kombinasyonu şeklindedir. Bu eğim azaltma yönteminin değiştirilmiş bir şeklidir ve bazı eğitim verileri eğitim verilerinin büyük bir çoğunluğundan farklılık gösteriyorsa bu değişim iyi bir avantaj sağlar. Eğer hiç alışılmamış bazı veriler kullanılacaksa bu değişikliği küçük bir öğrenme oranı ile kullanmak iyi

olacaktır. Bununla birlikte eğitim verileri benzer olsa da bu değişiklik kullanılarak yaklaşmanın hızı arttırılabilir (44).

b) Eşlenik eğim algoritması

Standart geriye yayılım algoritması ağırlıkları eğimin ters yönünde adımsal olarak ayarlamaktadır. Bu yön, ağ performans fonksiyonunun en hızlı azaldığı yöndür. Fonksiyon eğimin ters yönü boyunca en hızlı şekilde azalsada bu önemli sayılacak hızlı bir yakınsama üretmemektedir. Eşlenik eğim algoritmalarında genellikle adımsal azalma yönlerinden daha hızlı yakınsamayı sağlayan eşlenik yönler boyunca bir arama yapılmaktadır (30).

c) Adapte olabilen öğrenme oranları

Standart geriye yayılım algoritması, o anki ağırlıklar için olan hata yüzeyinde hatanın en hızlı azaldığı yönde ağırlıkları değiştirir. Ağırlık ayarlamasının yönünün değiştirilmesi hakkında çeşitli yöntemler önerilmiştir ve bu konuda çalışılmıştır. Bunlardan biri de geriye yayılım ağının öğrenme oranının arttırılması için öğrenme oranının eğitim sırasında değiştirilmesidir. Başlangıç ağ çıktısı ve hatası bulunur. Her döngüde ağırlıklar ve sapma o an ki öğrenme oranı kullanılarak hesaplanır ve yeni ağırlıklar ve hata bulunur. Yeni hata eski hatayı geçerse yeni ağırlıklar ve sapma uygulanmaz ve buna ek olarak öğrenme oranı azaltılır (genelde 0,7 ile çarpılır), aksi halde yeni ağırlıklar ve sapma uygulanır ve öğrenme oranı arttırılır (genelde 1,05 ile çarpılır) (30).

d) Delta-bar-delta

Bu yaklaşımın temelinde her ağırlığın kendi öğrenme oranının olması vardır. Öğrenme oranları eğitim sırasında değiştirilir. Eğer ağırlık değişimi (artma veya azalma) birkaç adım için aynı yönde ise ağırlıklar için öğrenme oranı arttırılmalıdır. Bir ağırlık için oluşan hatanın kısmi türevinin işareti birkaç adım için aynı olursa ağırlık değişimi aynı yönde değiştirilmelidir, tam tersi olursa azaltılmalıdır (30).

2.3.8.3. Öğrenme algoritmalarının uygulanış biçimleri

Yapay sinir ağı hem gerçek zamanlı (online) hem de toplu işlem yolu ile (batch learning) eğitebilir. Gerçek zamanlı yöntemde, her bir giriş için bir sonraki giriş gelmeden ağırlıklar güncellenir. İkinci yöntemde ise daha önceden elde edilmiş verilere göre ağırlıklar güncellenir. Bu yöntemde, bütün veri düzenleri toplanır ve ağırlıklar minimize edilmiş toplam hata fonksiyonuna göre güncellenir. Online veya batch learning algoritması yönteminin uygulaması problemin yapısına bağlıdır. Bazı sistemlerde ise toplu öğrenme, tüm girdiler bitmeden, birkaç girdide bir ağırlık güncellemesi yapılarak uygulanır. Bu yöntem her iki yontemin bir nevi birleşimi gibidir ve "mini-batch" olarak anılmaktadır. Gerçek zamanda ve toplu işlem yolu ile öğrenme yöntemleri arasındaki bazı farklılıklar şu şekilde özetlenebilir:

1. Yapay sinir ağının eğitimi başlamadan önce gerekli veriler mevcut değilse ya da eğitim örnekleri çok geniş ise gerçek zamanlı öğrenme yöntemi tercih edilir.

2. Yüksek doğruluk gerektiren durumlarda toplu işlem yolu ile öğrenme yönteminin kullanılması daha uygun olur.

Ağ mimarisinin belirlenmesi işlemi yapay sinir ağları kullanımında önemli bir problemdir. YSA yapıları arasında performans ve karakteristik özellikler bakımından farklar vardır. YSA yapıları özellikle ağın modelleme yeteneğini belirledikleri için oldukça önemlidirler. Yapay sinir ağının tasarımı aşamasında bu ağ yapıları arasından uygulamaya en elverişli olanı seçilir. Yapay sinir ağlarında, Radyal Tabanlı Fonksiyon Ağları (Radial Bases Function Network, RBF), Vektör Kuantalamalı Öğrenme (Learning Vector Quantisation Networks, LVQ), Olasılık Sinir Ağları (Probabilistic Neural Networks, PNN), Genelleştirilmiş Regresyon Ağları (Generalized Regression Neural Network, GRNN), Hopfield Ağı, Elman ve Jordan Ağları, Kohonen Ağı ve Adaptif Rezonans Teorisi (Adaptive Resonance Theory, ART) gibi çok çeşitli ağ yapıları ve modelleri vardır (45).

2.4. Karar Ağaçları Yöntemi

Karar ağaçları, yorumlanmalarının kolay olması, veri tabanı sistemleri ile kolayca entegre edilebilmeleri, güvenilirliklerinin iyi olması nedenlerinden dolayı sınıflama modelleri içerisinde en yaygın kullanıma sahip olan yöntemlerdir. Bu yöntemler tahmin edici ve tanımlayıcı özelliklere sahiptir.

Karar ağacı düğüm, dal ve yaprak olarak adlandırılan üç temel kısımdan oluşan, anlaşılması oldukça kolay olan bir tekniktir (46). Bu ağaç yapısında her bir değişken bir düğüm tarafından temsil edilir. Dallar ve yapraklar ağaç yapısının diğer elemanlarıdır. Ağaçta en son kısım yaprak en üst kısım ise kök olarak adlandırılır. Kök ve yapraklar arasında kalan kısımlar ise dal olarak ifade edilir. Başka bir ifadeyle bir ağaç yapısı; verileri içeren bir kök düğümü, iç düğümler (dallar) ve uç düğümlerden (yapraklar) oluşur. Eğitim verilerine ait değişken bilgilerinden yararlanılarak bir karar ağacı yapısı oluşturulmasında temel prensip verilere ilişkin bir dizi sorular sorulması ve elde edilen cevaplar doğrultusunda hareket edilerek en kısa sürede sonuca gidilmesi olarak ifade edilebilir. Bu şekilde karar ağacı sorulara aldığı cevapları toplayarak karar kuralları oluşturur. Ağacın ilk düğümü olan kök düğümünde verilerin sınıflandırılması ve ağaç yapısının oluşturulması için sorular sorulmaya başlanır ve dalları olmayan düğümler ya da yapraklar bulunana kadar bu işlem devam eder.

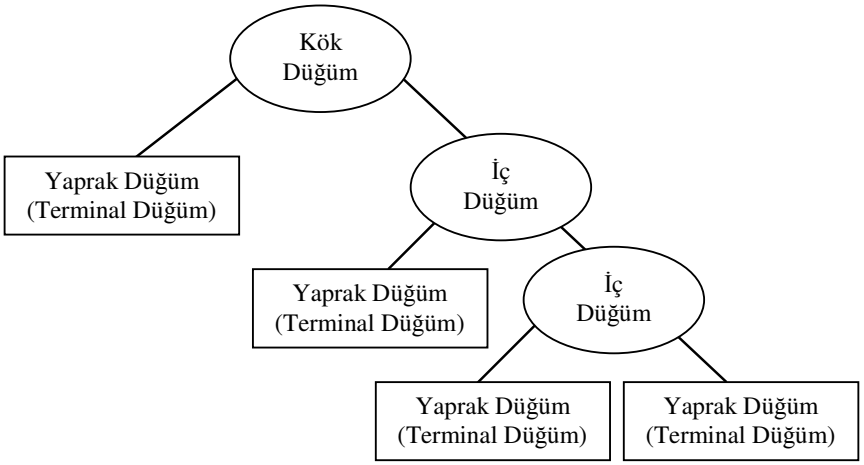
Karar ağaçlarının oluşturulmasındaki en önemli adım ağaçtaki dallanmanın hangi kritere veya kıstasa göre yapılacağı ya da hangi değişken değerlerine göre ağaç yapısının oluşturulacağıdır. Literatürde bu problemin çözümü için geliştirilmiş çeşitli yaklaşımlar vardır. Bunlardan en önemlileri bilgi kazancı ve bilgi kazanç oranı, Gini indeksi, Twoing kuralı ve Ki–Kare olasılık tablo istatistiği yaklaşımlarıdır.

Karar ağacında bulunan her bir dalın belirli bir olasılığı mevcuttur. Bu sayede son dallardan köke veya istediğimiz yere ulaşana dek olasılıkları hesaplamamız mümkündür. Karar düğümü, gerçekleştirilecek testi belirtir. Bu testin sonucu ağacın veri kaybetmeden dallara ayrılmasına neden olur. Her düğümde test ve dallara ayrılma işlemleri ardışık olarak gerçekleşir ve bu ayrılma işlemi üst seviyedeki ayrımlara bağımlıdır. Ağacın her bir dalı sınıflama işlemini tamamlamaya adaydır. Eğer bir dalın ucunda sınıflama işlemi gerçekleşemiyorsa, o dalın sonucunda bir karar düğümü oluşur. Ancak dalın sonunda belirli bir sınıf oluşuyorsa, o dalın

sonunda yaprak vardır. Bu yaprak, veri üzerinde belirlenmek istenen sınıflardan biridir. Karar ağacı işlemi kök düğümünden başlar ve yukarıdan aşağıya doğru yaprağa ulaşana dek ardışık düğümleri takip ederek gerçekleşir (47).

Karar ağacı tekniğini kullanarak verinin sınıflanması iki basamaklı bir işlemdir (46). İlk basamak öğrenme basamağıdır. Öğrenme basamağında önceden bilinen bir eğitim verisi, model oluşturmak amacıyla sınıflama algoritması tarafından analiz edilir. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. İkinci basamak ise sınıflama basamağıdır. Sınıflama basamağında test verisi, sınıflama kurallarının veya karar ağacının doğruluğunu belirlemek amacıyla kullanılır. Eğer doğruluk kabul edilebilir oranda ise kurallar yeni verilerin sınıflanması amacıyla kullanılır (47).

Karar ağaçları geçmiş veriye dayanarak yeni verilerin hangi sınıfa ait olduğuna, kurallar çıkartarak karar vermektedir. Karar ağacı, sorulan sorular ve alınan cevaplar doğrultusunda hareket eder ve sorulan sorulara alınan cevapları birleştirerek kurallar oluşturur. Oluşan ağaç birçok "eğer-ise"(if-then)'den oluşan kurallar bütünüdür de diyebiliriz. Soru sormaya verideki hangi değişkenden başlanacağına karar verildiğinde ilgili değişken ağacın kök düğümünü oluşturmuş olur. Kök düğümden başlayarak, cevabı veritabanında bulunan sorular sorulup alınan cevaplara göre yeni düğümler oluşturulmaktadır. Her düğüm kendinden sonra iki dala veya ikiden fazla dala ayrılmaktadır. Oluşan düğümden sonra yeni soru sorulamıyorsa dallanma bitmiştir ve bir sınıfı temsil eden yaprağa ulaşılmıştır (10). Şekil 18'de karar ağacını oluşturan kök düğüm, iç düğüm ve terminal düğümler gösterilmiştir.



Şekil 18. Karar ağacı örneği.

2.4.1. Karar ağaçlarının kullanım alanları

Karar ağaçları sınıflama, karar teorisi, kümeleme ve tahminsel fonksiyonlarda kullanılmaktadır. Karar ağaçlarını oluşturacak verideki değişkenler (nitelik/özellik) kategorik veya sürekli olabilirler. Eğer bağımlı değişken sürekli ise karar ağaçları regresyon ağaçları olarak adlandırılır. Eğer bağımlı değişkenler kategorik ise buna sınıflama ağacı denilmektedir. Bu farklılığa rağmen karar ağaçları benzer biçimde kurulmaktadırlar. Karar ağaçları tıp alanında teşhis için, botanikte sınıflama için, felsefede karar teorisi için, ekonomide ise yatırım alternatiflerini belirlemek için sıklıkla kullanılır. Karar ağaçları kurulma biçimlerine göre birbirlerinden ayrılmaktadırlar. Bazı durumlarda yukarıdan aşağı doğru kurulurken bazı durumlarda soldan sağa doğru kurulabilirler (48).

Karar ağacı temelli analizlerin yaygın olarak kullanıldığı sahalar;

- Belirli bir sınıfın üyesi olacak elemanların belirlenmesi,
- Çeşitli vakaların yüksek, orta, düşük risk grupları biçiminde kategorilere ayrılması,
- Gelecekte gerçekleşebilecek olayların tahmin edilebilmesi için kurallar oluşturulması,
- Parametrik modellerin kurulmasında kullanılacak çok sayıdaki değişken ve veri kümesinden önemli olanlarının seçilmesi,

- Yalnızca belirli alt gruplara özgü ilişkilerin tanımlanması,
- Sürekli değişkenlerin kategorik değişkenlere dönüştürülmesi ve kategorilerin birleştirilmesi olarak sayılabilir.

Karar ağacının kullanıldığı uygulamalardan bazıları ise aşağıdaki biçimde sıralanabilir:

- Hangi demografik grupların mektup aracılığıyla yapılan pazarlama uygulamalarında yüksek cevaplama oranına sahip olduğunun belirlenmesi,
- Kredi geçmişlerinin kullanılmasıyla bireylere ilişkin kredi kararlarının verilmesi,
- İşletmeye en faydalı olan bireylerin özelliklerinin kullanılmasıyla işe alma süreçlerinin belirlenmesi,
- Tıp ile ilgili gözlem verilerinden hareketle en etkin kararların verilmesi,
- Satışları hangi değişkenlerin etkilediğinin belirlenmesi,
- Ürün hatalarına yol açan değişkenlerin belirlenmesi (49).

2.4.2. Karar ağaçlarının avantajları ve dezavantajları

Karar ağaçları sınıflandırma ve tahmin için güçlü ve popüler araçlardır. Karar ağaçlarının çekici olan yönü bir takım kuralları temsil etmesidir.

Karar ağaçlarının güçlü yönleri aşağıdaki gibi özetlenebilir:

- Karar ağaçları anlaşılabilir kurallar üretirler,
- Sınıflandırma yaparken çok küçük bir hesaplama gerektirir,
- Sınıflama ve kestirim için hangi değişkenlerin daha önemli olduklarını açık belirtiler ile gösterir,
- Parametrik olmayan bir model olduğu için varsayımları çok kısıtlıdır,
- Bağımlı ve bağımsız değişkenler arasındaki ilişki görsel sunuma sahip olduğundan ağaç şeklindeki model sonuçları kolaylıkla yorumlanabilir,
- Hem bağımlı hem bağımsız değişkenler için kayıp veya eksik değerler ile aşırı uç değerlerden etkilenmeyen bir metottur (49).

Zayıf yönleri ise,

- Sürekli değişken değerlerini tahmin etmekte çok başarılı değildir,
- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı değildir,
- Zaman ve yer karmaşıklığı öğrenme kümesi örnekleri sayısına, değişken sayısına ve oluşan ağacın yapısına bağlıdır,
- Hem ağaç oluşturma karmaşıklığı hem de ağaç budama karmaşıklığı fazladır.

Karar ağacı oluşturulduktan sonra bir test verisini sınıflandırmak oldukça kolaydır. Kök düğümden başlayarak kayda test koşulu uygulanır ve her sonuç için ona ait uygun dal takip edilir. Buradan ya yeni test koşulunun uygulanacağı başka bir iç düğüme, ya da bir yaprak düğüme ulaşılır. Böylece test verisinin hangi sınıfa ait olduğu hangi yaprakta sonlandığına göre belirlenmiş olur (10).

Karar ağaçları oluşturulurken kullanılan algoritmanın ne olduğu önemlidir. Çünkü kullanılan algoritmaya göre oluşturulan ağacın şekli değişebilir. Değişik ağaç yapıları farklı sınıflandırma sonuçları verir. Kök düğümü oluşturan ilk düğümün farklı olması, en uçtaki yaprağa ulaşırken izlenecek yolu dolayısıyla sınıflamayı değiştirecektir. Gerek kök düğümün gerekse de sonraki her bir düğümün belirlenmesinde en önemli kriter, o noktadan dallara ayrıldığında veritabanının geri kalan kısmının benzer büyüklükte parçalara ayrılıp ayrılmadığıdır. Örneğin veri tabanında bulunan cevap evet/hayır gibiyse iki eşit parçaya, evet/hayır/belki gibi üç kategorili ise mümkün olduğunca üç eşit parçaya bölünmesi istenmektedir. Burada amaç en kısa yoldan istenilen yanıta veya sınıfa ulaşmaktır (2).

Ağaç tabanlı yöntemlerin temelini oluşturan karar ağaçları modellerinin ilk uygulamaları AID (Automatic Interaction Detector) algoritması ile yapılmıştır ve çeşitli algoritmalar ile sürdürülmüştür. Geliştirilen bu algoritmalar içerisinde CHAID (Chi-Squared Automatic Interaction Detector), CART (Classification and Regression Trees), ID3 (Iterative Dichotomiser 3), Exhaustive CHAID, C4.5, MARS (Multivariate Adaptive Regression Splines), QUEST (Quick, Unbiased, Efficient Statistical Tree), C5.0, SLIQ (Supervised Learning in Quest), SPRINT (Scalable Parallelizable Induction of Decision Trees) başlıcalarıdır (50).

Burada sayılan algoritmalardan başka çok çeşitli ağaç tabanlı algoritmalar da geliştirilmiştir. Son yıllarda birden çok sınıflandırıcının bir araya getirilmesi ile oluşan ve topluluk yöntemler ya da komiteler olarak adlandırılan algoritmalar önem kazanmıştır.

2.4.3. En sık kullanılan karar ağacı algoritmaları

2.4.3.1. CART (Classification and Regression Trees) algoritması

Ele alınan bağımlı değişken (hedef değişken, target variable) kategorik yapıda ise yöntem sınıflama ağaçları (Classification Trees, CT), sürekli yapıda ise regresyon ağaçları (Regression Trees, RT) adını almaktadır. 1984 yılında Breiman, Freidman, Olhsen ve Stone tarafından geliştirilen algoritma ikili karar ağaçları oluşturur. CART, iki çocuk düğümü oluşturup bütün açıklayıcı değişkenleri kullanarak veriyi alt gruplara ayırmak üzerine kurulmuştur. En iyi açıklayıcı değişken safsızlık ve değişim ölçülerindeki değişkenliği kullanarak seçilir. Burada amaç, hedef değişkene ilişkin mümkün olabilen veri alt gruplarını oluşturmaktır. Bağımlı değişken tektir ve sınıflı, sıralı ve sürekli tipte; bağımsız değişken ise bir ya da daha fazla olabilir ve yine sınıflı, sıralı ve sürekli tiptedir. CART algoritmasının limitleri ağacın ikili ayırmalarla sınırlandırılması ve değişken seçerken maliyet matrisinin dikkate alınması şeklinde özetlenebilir.

CART algoritmasının adımları şu şekildedir:

1. İlk düğümden başlayarak bölünmeyi sağlayacak tüm mümkün adayların içinden bir tane ayıraç seçilir,
2. Hedef değişkenin tipine göre safsızlık ölçütü hesaplanır,
3. Açıklayıcı değişkenler safsızlık ölçütlerine göre karşılaştırılır,
4. Safsızlık ölçütünü maksimum yapan değişkene göre ayrıştırma yapılır,
5. Ayıraç belirleme sürecine diğer açıklayıcı değişkenler için de devam edilir,
6. Herhangi bir durdurma kuralına rastlayana kadar ağaç büyütülür (50).

Sınıflama ağacı, kök düğümden başlayarak devam eden ve her düğümde o düğüme ait deney ünitelerine uygulanan basit sorulardan alınan evet/hayır

cevaplarına göre oluşan yollar içerir. Her düğümde uygulanan bu sorulara "ayıraç" denir. Bu işlem ayırma olarak adlandırılır.

Deney ünitelerinin birden fazla bağımsız değişken içermesi durumunda değişen tek şey ayıraçların tüm değişken ve değişken kombinasyonlarını tek tek ele almasıdır. Bu durumda, deney ünitelerinin içerdiği bağımsız değişkenler ve bu değişkenlerin birbirleri ile kombinasyonlarının tanımlı bulunduğu aralıklardaki tüm olası değerler birer ayıraç olarak düşünülüp, mümkün olan tüm olası ayırmalar belirlenir. Oluşan ağaçlarda homojen olmayan düğümlere "çocuk düğümü", homojen düğümlere ise "terminal düğüm" adı verilir.

Herhangi bir düğümün heterojenlik değeri safsızlık (impurity) ölçüsü olarak adlandırılır ve bu değer safsızlık fonksiyonu kullanılarak hesaplanır. Safsızlık ölçüsü sıfır değerini alıyorsa düğüm tamamen homojendir.

Sınıflama ağaçlarında kullanılabilecek birçok alternatif safsızlık ölçüsü (Gini, Twoing, Chi-square, G-square) vardır. Ayırma fonksiyonundan anlaşılacağı gibi, kullanılan safsızlık ölçüsü herhangi bir t düğümü için en iyi ayırmanın seçimini önemli bir şekilde etkilemektedir. Bu nedenle safsızlık ölçüleri literatürde en iyi ayırma kriterleri (ya da ayırma kuralları) olarak da bilinirler. En yaygın olarak kullanılan ayırma kriterleri Gini Diversity Index (Gini) ve Twoing Kuralı'dır. Ayırım işlemi kategorik bağımlı değişkenler için *gini, twoing,* sürekli değişkenler için *en küçük kareler sapması (Least-Squared Deviation)* indeks hesaplamalarına göre yapılmaktadır (1).

CT metodunda bir sınıflama ağacı oluşturulurken ön olasılıklar (prior probabilities) kullanılır. Ön olasılıklar deney ünitelerinin ait olacağı sınıfın belirlenmesini etkiler. j sınıfı için ön olasılık değeri (π_j) ile gösterilir ve bu değerler ya veri setinden hesaplanır ya da araştırmacı tarafından bildirilir. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile ise doğruluk oranı hesaplanır (Doğruluk Oranı = 1 - Hata Oranı). Verilerin sınıflandırılması için oluşturulan modellerin hata oranlarına karar vermek için risk matrisi kullanılmaktadır. Ayırma sonucunda ortaya çıkan herhangi bir düğüme atanacak olan en uygun sınıf aşağıdaki gibi tahmin edilir;

$C(j/i)$: i sınıfını j sınıfı gibi sınıflamanın maliyeti (risk matrisi katsayıları),

π_i : i sınıfının önceki olasılığı,

N_i : Öğrenme setinde i sınıfında bulunan deney ünitelerinin sayısı,

N_i^t : t düğümünde i sınıfında bulunan deney ünitelerinin sayısı olmak üzere;

$\frac{C(j/i)\pi_i N_i^t}{C(j/i)\pi_i N_j^t} > \frac{N_i}{N_j}$ eşitsizliği j'nin bütün değerleri ($j = 1, 2, \ldots, k$ ve $j \neq i$) için sağlanıyorsa t düğümüne en uygun olarak i sınıfı atanır.

Düğümün yapısına göre bazı durumlarda birden fazla sınıf yukarıda belirtilen eşitsizliği sağlayarak en uygun sınıf konumuna girer ya da hiçbir sınıf bu eşitsizliği sağlayamaz. Böyle bir durumda en uygun sınıfın belirlenmesi için çoğulluk ve minimum risk olmak üzere iki alternatif kural mevcuttur. Çoğulluk kuralı hatalı sınıflama maliyetini göz önüne almaksızın (eşit varsayarak) düğüm içerisinde en büyük orana sahip olan sınıfı en uygun sınıf olarak atar. Minimum risk kuralı ise düğüm içerisinde deney ünitelerinin sınıflara dağılımını göz önüne almaksızın (eşit varsayarak) düğüm içerisinde hatalı sınıflama maliyetini minimum yapan sınıfı en uygun sınıf olarak belirler. CT modellerinde tekrarlı ikili bölünmelerle homojen alt gruplar elde edilir ve ağaç bu şekilde büyümeye devam eder.

CT' de meydana gelen büyüme;

1. Her çocuk düğümündeki gözlem sayısı sadece bir veya on gözlem ise,
2. Her düğümde grup içi homojenlik söz konusu ise,
3. Ağacın düzey sayısında analizi yürüten kişi tarafından bir sınırlama yapıldıysa,
4. Yeni oluşacak düğümlerde fazla bir değişiklik yaratmıyorsa durur.

Ağaç inşası sonunda elde edilen ağaç büyük (maksimum) ağaç olarak adlandırılır ve öğrenme setindeki deney ünitelerine en uygun ağaçtır. Ancak maksimum ağaç pratikte iki dezavantaja sahiptir:

1. Maksimum ağaç öğrenme setini kusursuz biçimde tanımlar, çünkü eklenen her bağımsız değişken hatalı sınıflama oranını düşürür. Bu durumda, maksimum ağaç öğrenme seti için olması gerekenden daha iyi bir tahmin

modeli (overfitting) sunar. Ancak, öğrenme setine aşırı uyumlu maksimum ağaçlar farklı bir veri seti (örneğin test seti) söz konusu olduğunda iyi bir tahmin sağlayamazlar.

2. Bir sınıflama ağacının karmaşıklık ölçüsü o ağacın terminal düğüm sayısına eşittir. Terminal düğüm sayıları ve dolayısıyla karmaşıklığı yüksek olan maksimum ağacın anlaşılması ve yorumlanması güçtür.

Maksimum ağacın pratikte ortaya çıkardığı bu sorunların çözümü için maksimum ağacın budanması yani maksimum ağaçtan oluşturulan daha küçük bir ağacın seçilmesi gereklidir. Maksimum ağacın budanması daha küçük ağaçlar dizisi oluşturur ve oluşturulan bu dizi içerisinden optimum ağaç seçilir. Optimum ağaç maksimum ağaçtan daha az karmaşıklığa sahiptir ancak, öğrenme setine maksimum ağaçtan daha az uyumludur ve hatalı sınıflama oranı daha yüksektir. Maliyet-karmaşıklık budama metoduna göre maksimum ağaç, maliyet-karmaşıklık ölçüsü minimum değerine ulaşıncaya kadar budanır ve optimum ağaç elde edilir. Sınıflama ağaçlarında üç alternatif doğruluk tahmin yöntemi vardır. Bunlar yeniden yerine koyma tahmini, test örneği tahmini ve çapraz geçerlilik testidir. Model geçerliliğinin sınanmasında sık kullanılan çapraz geçerlilik yönteminde veri seti rastgele olarak k gruba ayrılır. k değeri genellikle 10 olarak seçilir. Bu gruplardan biri dışarıda bırakılarak geriye kalan kısım ile model oluşturulur. Oluşturulan modelin sınıflama performansı dışarıda bırakılan veriler üzerinde test edilir. Sırasıyla bu işlem dönüşümlü olarak diğer k grup için de tekrarlanır. Son olarak elde edilen k tane doğru sınıflama oranının ortalaması alınarak modelin genel performansı değerlendirilir.

CART, ele alınan veri kümesi eksik değerler içerdiğinde kullanışlı bir analizdir. Eksik değerler çok fazla olduğunda bu değerler bir vekil değişken olarak ağaç yapısında yer alırlar.

2.4.3.2. CHAID (Chi-squared Automatic Interaction Detector) algoritması

CART algoritmasının dışında en çok kullanılan karar ağacı algoritmalarından biri de CHAID algoritmasıdır. CHAID metodu 1980'de Kaas tarafından en iyi bölmeyi hesaplamak için istatistik olarak anlamlı bir farklılığın olmadığı, hedef değişkene uyan çiftlerde tahmin değişkeninin olası kategori çiftini birleştirmesiyle

oluşturulmuştur (51). En uygun bölümleri seçmek için kullanılan entropy veya gini metrikleri yerine Ki-kare testi kullanılmaktadır. Bu analizde amaç veriyi daha homojen alt gruplara bölmektir. Kullanılan istatistiksel test, hedef değişkenin ölçüm düzeyine bağlıdır. Eğer hedef değişken sürekli bir değişken ise F testi, kategorik ise Ki-kare testi kullanılmaktadır.

CHAID ile diğer yöntemler arasındaki en önemli farklılıklardan birisi, ID3, C4.5, ve CART ikili ağaçlar türetirken, CHAID çoklu ağaçlar türetmektedir (52). CHAID sürekli ve kategorik tüm değişken tipleriyle çalışabilmektedir. Bununla beraber, sürekli tahmin edici değişkenler otomatik olarak analizin amacına uygun şekilde kategorize edilmektedir. CHAID, Ki-Kare metriği vasıtasıyla, ilişki düzeyine göre farklılık rastlanan grupları ayrı ayrı sınıflamaktadır. Dolayısıyla, ağacın yaprakları, ikili değil, verideki farklı yapı sayısı kadar dallanmaktadır (53). CHAID analizi bağımlı değişkendeki varyasyonu bölümler içi minimum, bölümler arası maksimum olacak şekilde farklı alt gruplara veya bölümlere tekrarlı olarak parçalayan bir tekniktir. CHAID orijinal olarak değişkenlerdeki etkileşim veya kombinasyonları bulan bir teknik olarak geliştirilmiştir (54).

Yöntem, karmaşık bir veri setindeki yapıyı aramak için kullanılan bir yöntem olarak belirli avantajlara sahiptir. Bu avantajlar;

- Bağımlı ve bağımsız değişkenler için ölçü tipi sınıflı, sıralı veya süreklidir,
- Bağımsız değişkenlerin tamamının aynı düzeyde ölçülmesine gerek yoktur,
- Bağımsız değişkenlerdeki kayıp değerler sabit olmayan kategori (floating category) olarak muhafaza edilebilir,
- Uygun bir istatistiksel kriter kullanılırsa, sonuçlandırılan modelden şansa bağlı olmaksızın çok güçlü sonuçlar elde edilmesini sağlar.

Genel olarak CHAID yönteminin algoritması şu şekildedir;

Bağımlı değişken $d \geq 2$ kategoriye, analizde kullanılan belirli bir bağımsız değişkenin de $c \geq 2$ kategoriye sahip olduğunu varsayalım. Analizdeki bir alt problem, bağımsız değişkenin müsaade edilen kategorileri birleştirilerek verilen cxd boyutlu olumsallık tablosunun en anlamlı jxd boyutlu tablo durumuna indirgenebilme

problemi olsun. Kavramsal olarak ilk önce $T_j^{(i)}$ istatistiği hesaplanır. Bu, cxd tablosu için ($j = 2, 3, 4, \ldots, c$) bilinen χ^2 istatistiğidir. Eğer $T_j^* = \max T_j^{(i)}$ ise en iyi jxd tablosu için χ^2 değeri elde edilmiş demektir. Bu durumda T_j^* en anlamlı olarak seçilir.

Algoritmanın adımları şu şekildedir;

Adım 1. Her bir bağımsız değişken için, bağımlı değişkenin kategorileri ile bağımsız değişkenin kategorileri arasında çapraz tablo oluşturulur.

Adım 2. 2xd alt tablosunda bağımsız değişkene ait anlamlılığı en düşük olan kategori çiftleri bulunur. Birleşmeleri anlamlı bulunan iki kategori birleştirilir. Bu birleşme bir bileşik kategori olarak düşünülür ve bu adım bağımsız değişkenin kendi içindeki birleşmeleri anlamsız oluncaya kadar devam eder.

Adım 3. Üç ya da daha çok sayıda orijinal kategori içeren bileşik kategorilerin her biri için birleşmenin tekrar çözümlendiği en önemli iki bölünme bulunur. Eğer anlamlılık bir kritik değerin altındaysa bölünme tamamlanarak ikinci adıma dönülür.

Adım 4. Optimum düzeyde birleştirilen bağımsız değişkenlerin her birinin anlamlılığı hesaplanır, en çok anlamlı olan ayrılır. Eğer bu anlamlılık kritik bir değerden büyükse seçilen bağımsız değişkenin birleştirilen kategorilerine göre veri alt gruplara bölünür.

Adım 5. Henüz analiz edilmemiş veri için birinci adıma gidilir.

Her bir bağımsız değişken için kendi içinde kategorileri en anlamlı bir şekilde birleştirilip en iyi bölünme bulunduktan sonra, bağımlı değişkene göre olumsallık tablosu oluşturulur. Daha sonra χ^2 ve Bonferroni *p* değeri hesaplanır. Bağımsız değişkenler birbiri ile karşılaştırılıp en küçük *p* değerine sahip olan bağımsız değişkenin kategorilerine göre veriler alt gruplara ayrılır (54).

Ayrıntılı CHAID (Exhaustive CHAID), CHAID'in modifiye edilmiş şeklidir. CHAID yönteminin zayıf kalan yönlerini gidermek amacıyla geliştirilmiştir. Ayrıntılı CHAID, kullandığı istatistiksel testler ve kayıp değerleri değerlendirmesi açısından CHAID analizine benzerdir fakat hesaplanması uzun zaman almaktadır. Verilere

bağlı olsa da CHAID ile Ayrıntılı CHAID sonuçları arasında farklılık bulunmamaktadır (49).

2.4.3.3. QUEST (Quick Unbiased Efficent Statistical Tree) algoritması

QUEST yöntemi hızlı, yansız istatistiksel ağaç olarak bilinir. 1997 yılında Loh ve Shih tarafından geliştirilmiştir. CART algoritmasında olduğu gibi ikili karar ağaçları oluşturmak üzerine kurulu bir yapısı vardır. Fakat CHAID ve CART yöntemlerinden farklı olarak değişken seçimi ve ayırma noktası seçimi işlemlerini ayrı ayrı ele almaktadır. Bu yöntemde bağımlı değişken tek ve sınıflı, bağımsız değişken bir ya da daha fazla sayıda; sıralı, sınıflı ve sürekli yapıdadır. QUEST, bağımsız değişken kategorik olduğunda, yansız ağacın önemli olduğunda, büyük ve karmaşık bir veri setinin olduğu ve ağaç ikili bölünme ile sınırlandırıldığı durumlarda tercih edilir.

3. GEREÇ VE YÖNTEM

Bu çalışmanın uygulama bölümü, Uzman Dr. Evren KURTUL'a ait 16.04.2009 tarihinde (Toplantı no: 2009/5) etik kurul onayı alınmış tez çalışmasında kullanılan veri seti temel alınarak gerçekleştirilmiştir.

Çalışmada kullanılan veriler 01.09.2008–01.05.2009 tarihleri arasında BEÜ Uygulama ve Araştırma Hastanesi Kadın Hastalıkları ve Doğum Servisi'nde yatışı yapılan, yaşları 17 ile 46 arasında değişen gebelerden elde edilmiştir. Erken doğum yapan gebeler vaka grubunu, zamanında doğum yapanlar ise kontrol grubunu oluşturmuştur. Her iki gruptan da 120'şer gebe çalışmaya alınmıştır. Erken doğum risk faktörleri olarak düşünülen sorulardan bir anket formu oluşturularak gebelere uygulanmış, gebelerin doğuma girmeden önceki kiloları, boyları ve tansiyonları, hemogram, açlık kan şekeri, platelet değerleri ölçülmüştür. Yine doğum sonrası 1. gün kan lipid değerleri ölçülerek kaydedilmiştir.

Bu doğrultuda birimlerin sınıflandırılmasında kullanılabilecek olan değişkenler kadın hastalıkları ve doğum uzmanı ile birlikte değerlendirilerek belirlenmiştir. Modele alınan bağımsız değişkenler aşağıda sunulmuştur:

- Yaş (yıl)
- Anne eğitimi (1: İlkokul, 2: Ortaokul, 3: Lise, 4: Üniversite)
- Gelir düzeyi (1: 1000 TL'den az, 2: 1000-2000 TL arası, 3: 2000 TL'den fazla)
- Kilo alımı (1: 9 kilodan az, 2: 9-13 kilo arası, 3: 13 kilodan fazla)
- Erken doğum hikayesi (1: Hayır, 2: Evet)
- Sga bebek hikayesi (1: Hayır, 2: Evet)
- Kronik hastalık (1: Yok, 2: Var)
- Sigara (1: İçmiyor, 2: İçiyor)
- Takiplere geliş (1: Rutin takip, 2: Takipsiz)
- Anemi (1: Yok, 2: Var)
- Vajinal enfeksiyon (1: Yok, 2: Var)
- Preeklampsi hikayesi (1: Yok, 2: Var)
- Vajinal kanama (1: Yok, 2: Var)

- Cinsel ilişki (1: Yok, 2: Var)
- Egzersiz (1: Yok, 2: Var)
- Annenin çalışma durumu (1: Çalışıyor, 2: Ev hanımı)
- Beyaz kan hücresi sayısı (µl)
- Vücut kütle indeksi (VKİ)

Analizlerde kullanılan bağımlı değişken ise;

- Grup (1: Erken doğum yapanlar, 2: Zamanında doğum yapanlar)

Üç yöntemin performansını değerlendirmek üzere veri setine yapay sinir ağları, k-en yakın komşuluk ve karar ağaçları yöntemleri uygulanmıştır.

K-en yakın komşuluk analizinde minimum hata oranına sahip k değerine ulaşılmaya çalışılmıştır. Modelin geçerlilik sınaması için 10 katlı çapraz geçerlilik testinden faydalanılmıştır.

Karar ağacı uygulamasında ise CHAID algoritması kullanılmıştır. CHAID algoritmasının seçilme nedenlerinin başında sürekli ve kategorik tüm değişken tipleriyle çalışabilmesi gelmektedir. Bununla beraber, sürekli bağımlı degişkenler otomatik olarak analizin amacına uygun olarak kategorize edilmektedir. CHAID, ki-kare metriği vasıtasıyla ilişki düzeyine göre farklılık rastlanan grupları ayrı ayrı sınıflamakta ve ağacın yaprakları ikili değil, verideki farklı yapı sayısı kadar dallanmaktadır. Geçerlilik sınaması olarak yine 10 katlı çapraz geçerlilik testi kullanılmıştır. En uygun ağaç yapısı belirlenirken yerine koyma (resubstitution) ve çapraz geçerlilik (cross-validation) hata oranlarını minimum yapan ağaç yapısı belirlenmeye çalışılmıştır.

YSA analizinde sınıflandırma başarısını maksimum yapan ağ mimarisini belirlemek üzere çok sayıda deneme yapılmıştır. Eğitim yöntemi olarak Batch yöntemi tercih seçilmiştir. Bu yöntemin tercih edilme sebepleri, küçük veri setleri için en kullanışlı yöntem olması ve toplam hatayı direkt olarak en aza indirgemesidir.

Analizler için SPSS 18.0 paket programından faydalanılmıştır.

Model başarısı değerlendirilirken doğruluk oranı, duyarlılık ve seçicilik ölçütleri kullanılmıştır. Bir sınıflayıcı tarafından yapılan tahmin edilmiş sınıflamamalar ve gerçek durum hakkındaki bilgi karmaşıklık matrisi ile verilir. Tablo 6'da bir karmaşıklık matrisini gösterilmektedir. DP ve DN doğru sınıflandırılmış örnek sayısıdır. YP, aslında negatif sınıfındayken pozitif olarak tahminlenmiş örneklerin sayısıdır. YN ise aslında pozitif sınıfındayken negatif olarak tahminlenmiş örneklerin sayısını ifade eder. Genel olarak bir karışıklık matrisinde ana köşegen doğru tahminlenmiş örnek sayılarını; ana köşegen dışında kalan matris elemanları ise hatalı sonuçları ifade etmektedir.

Tablo 6. Karmaşıklık Matrisi.

		Gerçek durum	
		Pozitif	Negatif
Test Sonucu	Pozitif	DP	YP
	Negatif	YN	DN

Model başarımının ölçülmesinde sıklıkla kullanılan doğruluk oranı doğru sınıflandırılmış örnek sayısının toplam örnek sayısına oranıdır.

$$\text{Doğruluk oranı} = \frac{DP+DN}{DP+DN+YP+YN} \tag{14}$$

Duyarlılık ise doğru sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranı olarak tanımlanır.

$$\text{Duyarlılık} = \frac{DP}{DP+YN} \tag{15}$$

Model başarısını değerlendirmede kullanılan diğer bir ölçüt olan seçicilik doğru sınıflandırılmış negatif örnek sayısının toplam negatif örnek sayısına oranıdır.

$$\text{Seçicilik} = \frac{DN}{DN+YP} \tag{16}$$

4. BULGULAR

Analize dahil edilen 240 gebenin 120'si erken doğum yapmış olup (% 50), 120'si ise zamanında doğum yapmıştır (% 50). 240 olgudan elde edilen verilere ait tanımlayıcı istatistikler Tablo 7'de verilmiştir.

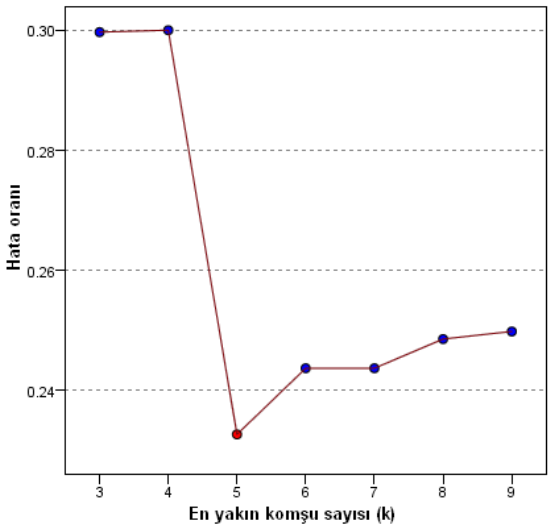
Tablo 7. Gruplara Ait Tanımlayıcı Özellikler.

		Hasta n= 120		**Kontrol n= 120**	
Anne yaşı		28.6 ± 5.6		27.3 ± 4.9	
Vücut kütle indeksi		29.3 ± 5.1		29.7 ± 5.0	
Beyaz kan hücresi sayısı		12415 ± 3751		10482 ± 2400	
		Sayı	**%**	**Sayı**	**%**
Annenin eğitim düzeyi	İlkokul	73	67.0	36	33.0
	Ortaokul	18	52.9	16	47.1
	Lise	17	30.4	39	69.6
	Üniversite	12	29.3	29	70.7
Gelir düzeyi	1000 TL'den az	71	84.5	13	15.5
	1000-2000 TL	35	35.4	64	64.6
	2000 TL'den fazla	14	24.6	43	75.4
Kilo alımı	9 kilodan az	50	90.9	5	9.1
	9–13 kilo arası	31	37.3	52	62.7
	13 kilodan fazla	39	38.2	63	61.8
Erken doğum hikayesi	Var	15	78.9	4	21.1
	Yok	105	47.5	116	52.5
Sga bebek hikayesi	Var	11	84.6	2	15.4
	Yok	109	48.0	118	52.0
Kronik hastalık	Var	38	71.7	15	28.3
	Yok	82	43.9	105	56.1
Sigara	İçiyor	56	65.1	30	34.9
	İçmiyor	64	41.6	90	58.4
Takiplere geliş	Takipsiz	26	92.9	2	7.1
	Rutin takip	94	44.3	118	55.7
Anemi	Var	78	61.9	48	38.1
	Yok	42	36.8	72	63.2
Vajinal enfeksiyon	Var	40	81.6	9	18.4
	Yok	80	41.9	111	58.1
Preeklampsi hikayesi	Var	31	93.9	2	6.1
	Yok	89	43.0	118	57.0
Vajinal kanama	Var	33	63.5	19	36.5
	Yok	87	46.3	101	53.7
Cinsel ilişki	Var	73	60.8	47	39.2
	Yok	47	39.2	73	60.8
Egzersiz	Yapan	58	43.9	74	56.1
	Yapmayan	62	57.4	46	42.6
Annenin çalışma durumu	Çalışıyor	15	27.8	39	72.2
	Çalışmıyor	105	56.5	81	43.5

4.1. K-En Yakın Komşuluk Analizi Sonuçları

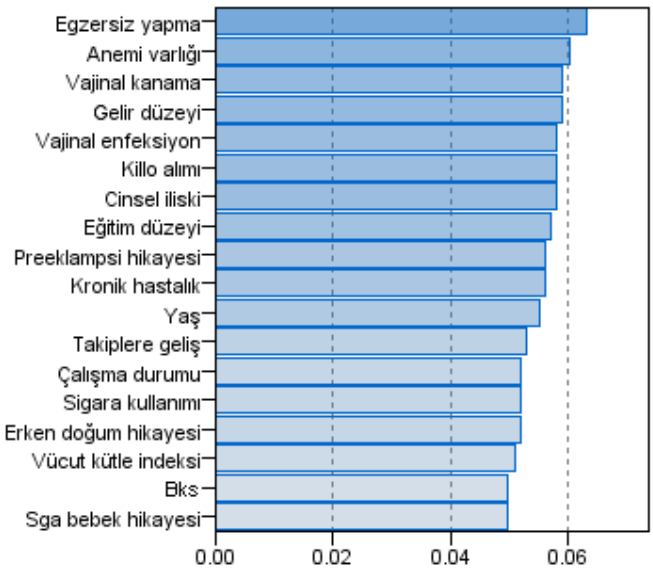
K-en yakın komşuluk analizinde bağımlı değişken olarak, erken doğum yapanlardan oluşan hasta grubu ve zamanında doğum yapanlardan oluşan kontrol grubu grup değişkeni; yaş, vücut kütle indeksi, annenin eğitimi düzeyi, gelir düzeyi, hamilelik boyunca kilo alımı, erken doğum hikayesi, sga bebek hikayesi, kronik hastalık varlığı, sigara, takiplere geliş, anemi varlığı, vajinal enfeksiyon, preeklampsi hikayesi, vajinal kanama, cinsel ilişki, egzersiz, annenin çalışma durumu ve beyaz kan hücresi sayısı (BKS) bağımsız değişken olarak alınmıştır.

Analizde verilerin % 70'i eğitim seti, % 30'u ise geçerlilik seti olarak ayrılmıştır. Uzaklık ölçütü olarak doğruluk oranını artırmasından dolayı Manhattan (City Blok) uzaklık ölçüsü seçilmiştir. Geçerlilik sınaması için 10 katlı çapraz geçerlilik testi sonucunda en düşük hata oranını veren k değeri 5 olarak hesaplanmıştır. Seçilen çeşitli k değerlerine ait hata grafiği aşağıda verilmiştir (Şekil 19).



Şekil 19. Seçilen k değerlerine ait hata oranları grafiği.

Analiz sonucunda egzersiz yapma değişkeni sonuca katkısı bulunan en önemli değişken, sga bebek hikayesi en az katkıda bulunan değişken olarak saptanmıştır (Şekil 20).



Şekil 20. Değişkenlerin önem sıralamasını gösteren grafik.

Yapılan sınıflandırma işlemi sonucunda testin doğruluk oranı % 78.3, duyarlılığı % 65, seçiciliği ise % 91.7 olarak hesaplanmıştır. Elde edilen sonuçlar Tablo 8'de verilmiştir.

Tablo 8. K-En Yakın Komşuluk Analizine Ait Sınıflandırma Sonuçları.

		Gerçek durum		**Toplam**
		Vaka	Kontrol	
Test Sonucu	Vaka	78	10	88
	Kontrol	42	110	152
Toplam		120	120	240

4.2. Karar Ağacı Analizi Sonuçları

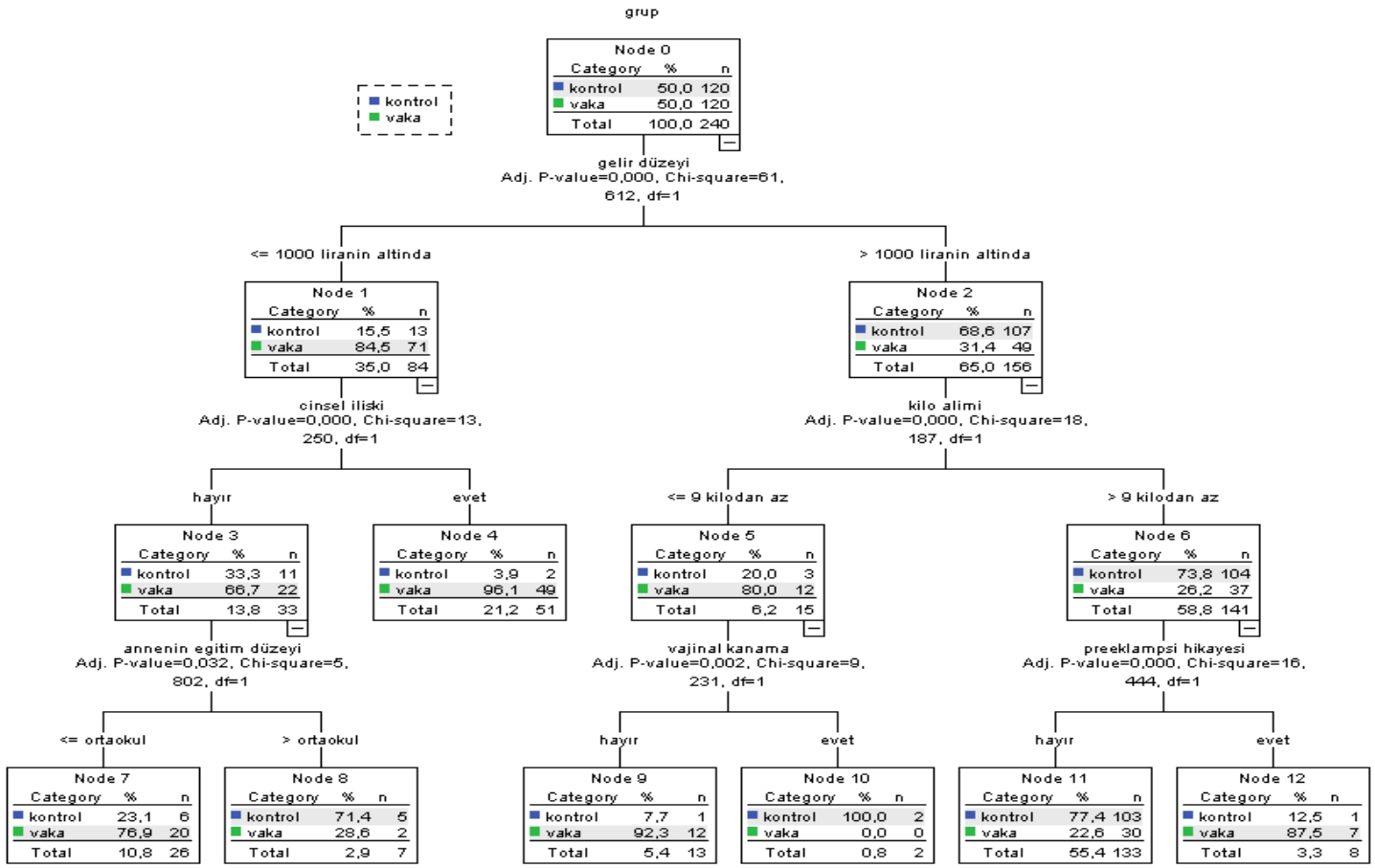
Karar ağaçları uygulamasında, k-en yakın komşuluk analizine dahil edilen bağımlı ve bağımsız değişkenlerin tümü analize katılmıştır. Bu çalışmada karar ağacı algoritmalarından sıklıkla kullanılan CHAID algoritması kullanılmıştır. Geçerlilik sınaması olarak yine 10 katlı çapraz geçerlilik testi kullanılmıştır. En uygun ağaç yapısı belirlenirken yerine koyma (resubstitution) ve çapraz geçerlilik (cross-validation) hata oranlarını minimum, sınıflandırma başarısını maksimum yapan ağaç

yapısını belirlemek üzere terminal düğüm sayısı 5, çocuk düğüm sayısı 2 olarak belirlenmiştir. Yerine koyma ve çapraz geçerlilik sınamalarına ait risk ve standart hata değerleri aşağıdaki tabloda gösterilmiştir (Tablo 9).

Tablo 9. CHAID Analizi Sonucu Yerine Koyma ve Çapraz Geçerlilik Sınamalarına Ait Risk ve Hata Değerleri.

	Risk	**Standart hata**
Yerine koyma	0.175	0.025
Çapraz geçerlilik	0.254	0.028

Analiz sonucunda bağımlı değişken olan grup değişkenini etkileyen en önemli değişkenin gelir düzeyi değişkeni olduğu görülmektedir ($p<0.001$). Daha sonra yer alan değişkenler ise cinsel ilişki, kilo alımı, annenin eğitim düzeyi, vajinal kanama ve preeklampsi hikayesi olmuştur. Analize ait karar ağacı aşağıda gösterilmiştir (Şekil 21).



Şekil 21. CHAID analizine ait karar ağacı.

Sınıflandırma başarısı olarak değerlendirildiğinde yapılan CHAID analizinin doğruluk oranı % 82.5 olarak bulunurken, duyarlılığı % 73.3, seçiciliği ise % 91.7 olarak bulunmuştur. Elde edilen sonuçlar aşağıdaki tabloda gösterilmiştir (Tablo 10).

Tablo 10. CHAID Analizine Ait Sınıflandırma Sonuçları.

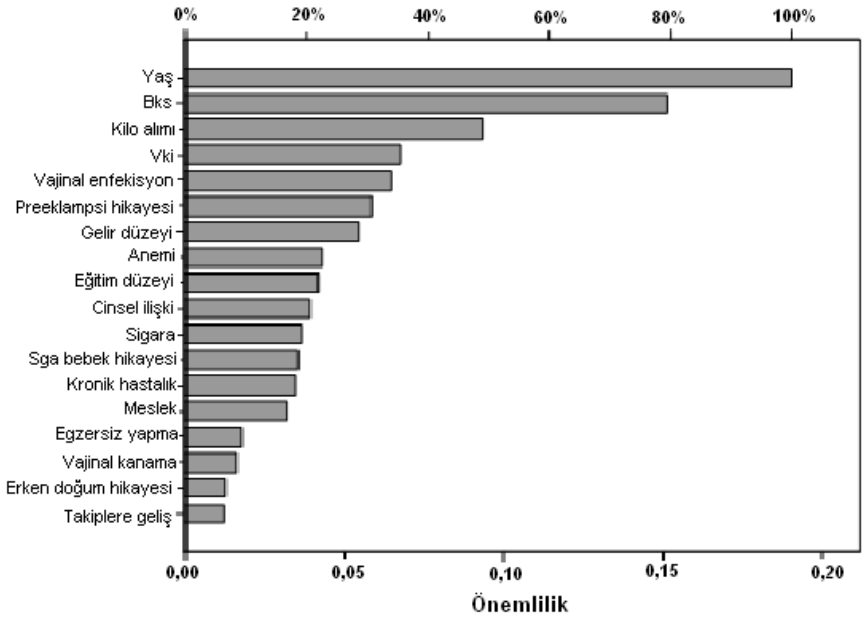
		Gerçek durum		**Toplam**
		Vaka	Kontrol	
Test Sonucu	Vaka	88	10	98
	Kontrol	32	110	142
Toplam		120	120	240

4.3. Yapay Sinir Ağı Analizi Sonuçları

YSA uygulamasında çok katmanlı yapay sinir ağı (multi perception neural network) modeli kullanılmıştır. Sayısal olan yaş, vücut kütle indeksi ve bks değişkenleri kovaryans olarak; kategorik olan egzersiz yapma, anemi varlığı, gelir düzeyi, vajinal kanama, vajinal enfeksiyon, kilo alımı cinsel ilişki, annenin eğitim düzeyi, preeklampsi hikayesi, kronik hastalık varlığı, takiplere geliş, çalışma durumu, sigara kullanımı, erken doğum hikayesi ve sga bebek hikayesi değişkenleri ise faktör olarak analize katılmıştır.

Analiz sırasında verilerin 164'ü eğitim seti, 76'sı test seti olarak ayrılmıştır. Yapay sinir ağının mimarisinde en yüksek sınıflandırma performasını sağlamasından ötürü aktivasyon fonksiyonu olarak gizli katmanda Hiperbolik tanjant, çıktı katmanında Sigmoid fonksiyon kullanılmış ve gizli katman sayısı 1 olarak belirlenmiştir. Girdi katmanında 18 birim, gizli katmanda 4 birim, çıktı katmanında ise 1 birim bulunmaktadır. Eğitim yöntemi olarak ise Batch yöntemi tercih seçilmiştir.

Analiz sonucunda analize en önemli katkıyı yapan değişken yaş olarak bulunurken en az katkıyı yapan ise takiplere geliş olarak bulunmuştur. Değişkenlerin önemlilik sırasını gösteren grafik aşağıda verilmiştir (Şekil 22).



Şekil 22. YSA analizi sonucu değişkenlerin önemlilik oranları.

YSA analizi için geçerlilik sınaması yapılamadığından modelin performansı test seti üzerinden değerlendirilmiştir. Yapılan sınıflandırma işlemi sonucunda eğitim verisine ait doğruluk oranı % 82.9, test verisine ait doğruluk oranı ise % 90.8 olarak bulunmuştur. Bulunan sınıflandırma sonuçları diğer performans ölçütleri olan duyarlılık ve seçicilik bakımından değerlendirildiğinde eğitim setinin duyarlılığı % 78.5, seçiciliği % 87.1; test setinin duyarlılığı % 90.2, seçiciliği ise % 91.4 olarak hesaplanmıştır. Elde edilen sonuçlar aşağıdaki tabloda gösterilmiştir (Tablo 11).

Tablo 11. YSA Analizine Ait Eğitim ve Test Verilerinin Sınıflandırma Sonuçları.

Gerçek durum		**Test sonucu**		**Toplam**
		Kontrol	Vaka	
Eğitim	Kontrol	74	11	% 87.1
	Vaka	17	62	% 78.5
	Toplam	% 55.5	% 44.5	% 82.9
Test	Kontrol	32	3	% 91.4
	Vaka	4	37	% 90.2
	Toplam	% 47.4	% 52.6	% 90.8

5. TARTIŞMA

Bu tez çalışması, veri madenciliğinde kullanılan üç sınıflandırma tekniğinin sağlık alanından elde edilmiş gerçek bir veri seti üzerindeki performanslarının değerlendirilmesi amacıyla yapılmıştır.

Günümüzde veri miktarının artması ile orantılı olarak veri madenciliği kavramı da büyük önem kazanmış, pek çok alanda kullanımı yaygınlaşmıştır. Literatürde, çeşitli alanlarda veri madenciliği yöntemleri ile elde edilmiş sonuçlara dayanan çalışmaların yanı sıra, kullanılan tekniklerin tahmin başarılarını karşılaştırmak amacıyla yapılmış pek çok çalışma da mevcuttur. Yapılan bu karşılaştırma çalışmalarının kimi aynı amaçla kullanılan veri madenciliği tekniklerinin karşılaştırılması üzerine kurgulanmışken kimisi de veri madenciliği yöntemlerinin klasik istatistiksel yöntemlerle olan karşılaştırmalarını kapsamaktadır.

Eski ve yeni veri madenciliği yöntemlerinin karşılaştırılması amacıyla yapılmış çalışmalardan biri, Baumgartner ve arkadaşlarının (55) yenidoğanlardaki metabolik hastalıkların sınıflandırılması ile ilgili yaptıkları çalışmadır. Çalışmada, diskriminant analizi, lojistik regresyon, karar ağaçları, k-en yakın komşuluk, yapay sinir ağları ve destek vektör makineleri yöntemleri iki farklı veri setinde karşılaştırılmış ve ilk veri setinde % 99.9, diğer veri setinde ise % 99.3 doğruluk oranı ile YSA analizinin karar ağaçları ve k-en yakın komşuluk analizine göre daha iyi performans gösterdiği saptanmıştır.

Bir diğer araştırmada, Zhang ve Zhang (56), serebral palsili hastalarda yürüme sorunları ile ilgili çalışmalarında Kernel Fisher diskriminant analizi, Fisher lineer diskriminant analizi, karar ağaçları, k-en yakın komşuluk, çok katmanlı yapay sinir ağları ve destek vektör makinaları yöntemlerini uygulamışlar, YSA analizi için % 95.6, karar ağaçları için % 91.5 ve k-en yakın komşuluk analizi için ise % 67.7 doğruluk oranına ulaşmışlardır.

Cong ve arkadaşları (57) ilaç tasarımında TNF-α dönüştürücü enzim inhibitörlerinin belirlenmesinin tahmini ile ilgili yaptıkları araştırmada destek vektör makineleri, karar ağaçları, geri yayılımlı yapay sinir ağı ve k-en yakın komşuluk yöntemlerinin tahmin doğruluklarını karşılaştırdıklarında, YSA analizi ile karar ağaçları

yöntemlerinin doğruluk oranlarının % 97.5 iken k-en yakın komşuluk yönteminin % 96.6 olduğunu saptamışlardır.

Yedi farklı algoritmanın karşılaştırıldığı bir çalışmada, Acharya ve arkadaşları (58), ultrason kullanılarak tiroid lezyonlarını sınıflandırılmışlardır. Söz konusu çalışmada araştırmacılar k-en yakın komşu, karar ağaçları, destek vektör makinaları, Gauss karışım modelleri, radyal tabanlı olasılıksal yapay sinir ağları, Sugeno bulanık modelleri ve Naive-Bayes sınıflayıcılarını kullanmışlar ve bu sınıflandırma işleminde destek vektör makinası sınıflayıcısının % 100 doğruluk oranı ile en iyi sınıflamayı yaptığını görmüşlerdir.

Metin madenciliği alanında yapılan bir çalışmada ise, Hmeidi ve arkadaşları (59) Arapça metinlerin sınıflandırılması amacıyla k-en yakın komşu ve destek vektör makineleri tekniklerini karşılaştırmış, her iki tekniğinde çok yüksek performans gösterdiğini saptamışlardır.

Biyoteknoloji alanında, proteinlerin hücresel yerleşimlerinin tahmin edilmesi ile ilgili olarak Cai ve Chou (60) tarafından yapılan çalışmada, k-en yakın komşu analizi kullanılmış, uygulama sonucunda geliştirilen modelin başarı oranının çok yüksek olduğu gözlenmiştir.

Bilinen veri madenciliği tekniklerinin yanı sıra bazı tekniklerin birleştirilmesi sonucu ortaya çıkmış olan hibrit veri madenciliği modelleri de mevcuttur. Saeedmanesh ve arkadaşları (61) geliştirdikleri hibrit veri madenciliği tekniği ile karar ağaçları, yapay sinir ağları ve k-en yakın komşuluk yöntemini zaman serisi verisi üzerinde karşılaştırmış, önerilen modelin diğer üç modele göre tahmin doğruluğunu en az % 34 arttırdığını saptamışlardır. Çalışmada birleştirilmiş modellerin tek modellere göre bazı durumlarda daha güçlü sonuçlar verebileceği belirtilmiştir.

İşletme alanında Albayrak ve Yılmaz (62) tarafından yapılan bir çalışmada, İstanbul Menkul Kıymetler Borsası Ulusal 100 endeksi sanayi ve hizmet sektörlerinde faaliyet gösteren 173 şirket çalışma kapsamına alınmış ve şirketlere ait finansal bilgiler kullanılarak elde edilen verilere karar ağacı algoritmalarından CHAID algoritması uygulanmıştır. Yapılan analiz sonucunda karar ağaçları tekniğiyle işletmelerin birbirlerine göre konumları ortaya konmuş ve sektör değişkenini etkileyen en önemli değişkenler saptanmıştır.

Görüldüğü üzere, veri madenciliği algoritmaları ile yapılan çalışmaların sonuçları birbirlerinden farklılık göstermektedir. Bu algoritmaların karşılaştırılması yolu ile yapılan deneysel çalışmalar bilim dünyasında keskin eleştirilere maruz kalmaktadır. Hand (63), veri madenciliği algoritmalarının karşılaştırılması hakkında karşılaştırma sonuçlarının doğru olmayacağını, literatürde yer alan makalelerdeki çalışmaların aslında bir illüzyon yarattığını, deneysel çalışmaların ortaya koyduğu sonuçların gerçekle bağdaşmayacağını belirtmiştir. Doğası gereği veri madenciliği model başarımlarının veriye bağlı olduğunu, veri üzerinde yapılan önişleme işlemlerinin ve kullanılan algoritma parametrelerinin oluşan sonuç üzerinde farklı etkileri olacağını, kullanıcıya bağlı olarak aynı modelle farklı sonuçlar elde edilebileceğini belirtmiştir. Literatürde yer alan ve yeni geliştirilmiş olan bir algoritmanın eski bir algoritmayla karşılaştırılması yapılarak yeni algoritmanın daha başarılı olduğunun ispatlanmaya çalışıldığı makalelerde, yukarıda belirtilen sebeplerden ötürü ve geliştiricinin isteyerek ya da istemeden sergileyebileceği yanlı yaklaşımların etkili olacağı belirtilmektedir. Literatürdeki diğer karşılaştırma çalışmalarında sonucun kullanıcının yatkın olduğu modele bağlı olduğu, bu yüzden farklı makalelerde farklı sonuçlara ulaşılacağı belirtilmiştir. Bunun ötesinde bazı çalışmalarda kompleks algoritmaların klasik algoritmalara karşı daha başarılı olduğu şeklindeki iddiaların da aslında illüzyondan ibaret olduğu ifade edilmektedir (23).

Yapılan bir başka eleştiride ise akademik literatürde yapılmış olan karşılaştırma çalışmalarının çoğunda gerçek veriler kullanılmadığı, bu nedenle yapılan değerlendirmelerin doğru sonuç üretmemiş olduğu yönündedir (64). Tüm bu eleştirilere rağmen algoritmaların karşılaştırılması gerekliliği ortak bir görüş olarak kabul edilmiş, gerek uygulama gerekse geliştirme anlamında yapılan akademik çalışmalarda ve güncel uygulamalarda yer edinmiştir (23).

Bununla birlikte, Michie ve Spiegelhalter (65) araştırma sonuçlarını yayınladıkları kitaplarında benzer veri setlerinde belli algoritmaların daha başarılı olduğunu belirtmişlerdir. Bu bağlamda, hangi algoritmanın daha başarılı bir model ürettiğinin araştırıldığı bir çalışmada farklı veri kaynakları üzerinde daha çok sayıda algoritmanın kullanılarak karşılaştırılma yapılması ve farklı veri kaynaklarındaki karşılaştırılmaların sınıflandırılması gerekecektir.

6. SONUÇLAR

Sağlık alanındaki kullanımı gün geçtikçe yaygınlaşan veri madenciliğinde çeşitli amaçlar için farklı yöntemler kullanılmaktadır. Bu yöntemlere ait pek çok algoritma geliştirilmiştir. Bu algoritmalardan hangisinin daha üstün olduğu üzerine pek çok çalışma yapılmış, yapılan bu çalışmalarda farklı sonuçlar elde edilmiştir. Bunun başlıca sebepleri; yapılan işlemin performansının, kullanılan veri kaynağına, veri üzerinde yapılan ön işleme, algoritma parametrelerinin seçimine bağlı olmasıdır. Farklı kişiler tarafından, farklı veri kaynakları üzerinde, farklı parametrelerle yapılan çalışmalarda farklı sonuçlar oluşması doğaldır.

Sınıflandırma amacıyla kullanılan üç veri madenciliği tekniğinin karşılaştırıldığı bu çalışmada, yapay sinir ağları tekniği en iyi performansı gösteren teknik olmuştur.

Deneysel çalışmalar üzerine yapılan eleştiriler ışığında yapılan bir karşılaştırma işlemine dayanarak bir algoritmanın diğer bir algoritmaya kesin bir üstünlüğünden söz etmek doğru olmayacaktır. Yinede model başarımı karşılaştırmalarının, bir veri madenciliği çalışmasında önemli katkıları olacağı açıktır. Bir problem üzerinde yapılacak model oluşturma işleminde farklı algoritmaların karşılaştırılarak en başarılı olanın bulunmasının sonuçlara katkısı büyük olacaktır.

7. KAYNAKLAR

1. Akpınar H. Veri tabanlarında bilgi keşfi ve veri madenciliği. İ.Ü. İşletme Fakültesi Dergisi 29(1):1-22, 2000.
2. Silahtaroğlu G. Veri madenciliği. 1. Basım, s. 30, Papatya Yayınevi, İstanbul, 2008.
3. Ercan İ, Ediz B, Hacımustafaoğlu M, Kan İ, Bostan Ö. Kümeleme çözümlemesinin yeni doğan sarılıklı olgulara uygulanması. U. Ü. Tıp Fakültesi Dergisi 24 (1-2-3):17-22, 1997.
4. Raghavan VV, Deogun JS, Sever H. Data mining: Trends and issues. Journal of The American Society For Information Science 49(5):397-402, 1998.
5. Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to discovery knowledge in databases. AI Magazine 3(17):37-54, 1996.
6. Han J, Fu Y, Koperski K, Melli G, Wang W, Zaiane OR. Knowledge mining in databases: An integration of machine learning methodologies with database technologies. Canadian AI Magazine 38:4-8, 1995.
7. Luan J. Data mining and its applications in higher education. In Knowledge Management: Building a Competitive Advantage in Higher Education (Edited by A. Serban and J. Luan), pp. 17-36, Jossey Bass, San Francisco, USA, 2002.
8. Glymour C, Madigan D, Pregibon D, Smyth P. Statistical themes and lessons for data mining. Data Mining and Knowledge Discovery 1(1):11-28, 1997.
9. Stühlinger W, Hogl O, Stoyan H, Müller M. Intelligent data mining for medical quality management. Fifth Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2000), Workshop Notes of the 14th European Conference on Artificial Intelligence (ECAI-2000), pp. 55-67, Berlin, Germany, August 20-25, 2000.
10. Akman M. Veri madenciliğine genel bakış ve Random Forest yönteminin incelenmesi: Sağlık alanında bir uygulama. A. Ü. Sağlık Bilimleri Enstitüsü, Biyoistatistik Anabilim Dalı Yüksek Lisans Tezi, Ankara, 2010.
11. Alkan A, Falay E. Kamu uygulamalarında çözüm veri madenciliğinde. Strateji Geliştirme Başkanlığı Strateji Bülteni, 5:7-8, Eylül-Ekim 2007.

12. Baykal N (2003), Veri tabanı ve veri madenciliği, Tıp Bilişimi Güz Okulu Ders Notları, Orta Doğu Teknik Üniversitesi, Erişim adresi: www.turkmia.org/eski/file/231verimadenciligi_baykal.ppt. Erişim Tarihi: 05.08.2011.
13. Yıldırım P, Uludağ M, Görür A. Hastane veri sistemlerinde veri madenciliği. Akademik Bilişim Kongresi Bildiriler Kitabı, s. 106. Çanakkale, 30 Ocak-1 Şubat 2008.
14. Ramkumar GD, Swami A. Clustering data without distance functions, IEEE Bulletin of the Technical Committee on Data Engineering 21(1):9-14, 1998.
15. Özekes S. Veri madenciliği modelleri ve uygulama alanları. İstanbul Ticaret Üniversitesi Dergisi 2(3):65-82, 2003.
16. Amasyalı MF. Yeni makine öğrenmesi metotları ve ilaç tasarımında uygulamaları. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Bilimleri Anabilim Dalı Doktora Tezi, 2008.
17. Sever H, Oğuz B. Veri tabanlarında bilgi keşfine formel bir yaklaşım: Kısım 1-Eşleştirme sorguları ve algoritmalar. Bilgi Dünyası 3(2):173-204, 2002.
18. Cunningham P, Delany SJ. K-neighbour classifiers. Technical Report UCD-CSI-2007-4, School of Computer Science and Informatics, University College Dublin, Ireland, 2007.
19. Kırmızıgül Çalışkan S, Soğukpınar İ. KxKNN: K means ve k en yakın komşu yöntemleri ile ağlarda nüfuz tespiti, 2. Ağ ve Bilgi Güvenliği Sempozyumu, Girne, 16-18 Mayıs 2008.
20. Gutierrez-Osuna R. Introduction to pattern analysis. Course Notes, Department of Computer Science, A&M University, Texas, 2005.
21. Karabağ R. Kullanıcı davranış analizi ile nüfuz tespiti. Gebze Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi, Gebze, 2006.
22. Teknomo K (2006). K-nearest neighbor. Erişim adresi: http://people.revoledu.com/kardi/tutorial/Similarity/index.html, Erişim tarihi: 20.08.2011.
23. Coşkun C, Baykal A. Veri madenciliğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması. 13. Akademik Bilişim Konferansı, 2 Şubat-4 Mayıs, Malatya, 2011.
24. Kröse BJA, Van Der Smagt PP. An introduction to neural networks. 7th. Edition, pp:1-64, University of Amsterdam, The Netherlands, 1994.

25. Karakaya B. Yapay sinir ağlarının incelenmesi ve sırt ağrısı olan bireyler üzerinde bir uygulaması, A. Ü. Sağlık Bilimleri Enstitüsü, Biyoistatistik Anabilim Dalı Yüksek Lisans Tezi, Ankara, 2007.
26. Yurtoğlu H. Yapay sinir ağları metodolojisi ile öngörü modellemesi: Bazı makroekonomik değişkenler için Türkiye örneği. Devlet Planlama Teşkilatı Uzmanlık Tezi, Ankara, 2005.
27. İÜBK elektronik dergisi (2010). Erişim adresi: http://www.bilisimdergi.com/ YSA-Temel-Yapi-ve-ozellikleri-11-3.html, Erişim tarihi: 15.07.2011.
28. Fausett L. Fundamentals of neural networks. pp. 32-36, Prentice Hall, USA, 1994.
29. Ocakoğlu G. Lojistik regresyon analizi ve yapay sinir ağları tekniklerinin sınıflama özelliklerinin karşılaştırılması ve bir uygulama. Uludağ Üniversitesi Sağlık Bilimleri Enstitüsü, Biyoistatistik Anabilim Dalı Yüksek Lisans Tezi, Bursa, 2006.
30. Sığırlı D. Sınıflandırma probleminin çözümlenmesinde yapay sinir ağları ile diskriminant analizinin karşılaştırılması ve bir uygulama. Uludağ Üniversitesi Sağlık Bilimleri Enstitüsü, Biyoistatistik Anabilim Dalı Yüksek Lisans Tezi, Bursa, 2006.
31. Beşdok E, Erler M, Sağıroğlu S. Mühendislikte yapay zeka uygulamaları 1: Yapay sinir ağları. s. 160-165, Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd. Sti. Kayseri, 2003.
32. Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü (2011). Yapay sinir ağları, Erişim adresi:http://ube.ege.edu.tr/~cinsdiki/UBI521/Chapter-1/cinsdikici-neural-net-giris.pdf, Erişim tarihi: 20.07.2011.
33. Gallant SI. Neural network learning and expert systems. pp. 1-14, MIT Press, London, UK, 1993.
34. Haykin S. Learning processes; single-layer perceptrons; multilayer perceptrons. Neural Networks A Comprehensive Foundation. 2nd Edition, pp.14-68, Prentice Hall, USA, 1999.
35. Kakıcı A (2012). Yapay sinir ağları, Erişim adresi: http://www.ahmetkakici.com/ yapay-sinir-aglari/yapay-sinir-aglarinin siniflandirilmasi, Erişim tarihi: 03.01.2012.

36. Özdemir R. Elektrodepolama yöntemi ile elde edilen ZnFe ince filmlerinin elektriksel özdirenç özelliklerinin sezgisel yöntemler yardımıyla incelenmesi. Kilis 7 Aralık Üniversitesi Fen Bilimleri Enstitüsü, Fizik Anabilim Dalı Yüksek Lisans Tezi, Kilis, 2010.
37. Ham MF, Kostanic I. Principles of neurocomputing for science and engineering. pp. 136-140, Mcgraw-Hill Companies, New York, USA, 2001.
38. Bishop M. Neural networks for pattern recognition. pp. 77-145, Oxford University Press, USA, 1995.
39. Öztemel E. Yapay sinir ağları. s. 23-47, Papatya Yayıncılık, İstanbul, 2003.
40. Abdi H, Valentin D, Edelman B. Neural Networks. pp. 1-9, Sage Publications, California, USA, 1999.
41. Rumelhart DE, Hinton GE, Williams R. Learning internal representations by error propagation, parallel distributed processing. Volume 1, pp. 318-362, MIT Press, Cambridge, UK, 1986.
42. Wang S. An adaptive approach to market development forecasting. Neural Computing & Applications 8:3-8, 1999.
43. Kartalopoulos SV. Understanding neural network and fuzzy logic. pp. 75-76, IEEE Press, New York, USA, 1996.
44. Hertz J, Krogh A, Palmer RG. Introduction to the theory of neural computation. pp. 119-141, Addison-Wesley Publishing Co., New York, USA, 1993.
45. Schalkoff JR. Artificial Neural Network. pp. 142-149, McGraw-Hill International Editions, New York, USA, 1997.
46. Han J, Kamber M. Data mining: Concepts and techniques (Morgan Kaufmann Publishers), 1st Edition., pp. 332-336, San Francisco, USA, 2000.
47. Özekes S. Veri madenciliği modelleri ve uygulama alanları. İstanbul Ticaret Üniversitesi Dergisi 2(3):65-82, 2003.
48. Omitaomu OA. Decision Trees. In Michael W. Berry and Murray Browne (Eds.), Lecture Notes in Data Mining, pp. 39-51, World Scientific Publishing of Hackensack, New Jersey, USA, 2006.
49. Oğuzlar A. CART analizi ile hanehalkı işgücü anketi sonuçlarının özetlenmesi. Atatürk Üniversitesi İİBF Dergisi 18(3-4):79-90, 2004.
50. Pehlivan G. CHAID analizi ve bir uygulama, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstatistik Anabilim Dalı Yüksek Lisans Tezi, İstanbul, 2006.

51. Ziarko W. The Discovery, analysis, and representation of data dependencies in databases. In Piatetsky-Shapiro G. and Frawley WJ. (Eds.), Knowledge Discovery in Databases, pp. 195-209, AAAI/MIT Press, Cambridge, UK, 1991.
52. Koltan Yılmaz Ş. Veri madenciliği: İstanbul Menkul Kıymetler Borsası (İMKB) örneği, Zonguldak Karaelmas Üniversitesi Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı Yüksek Lisans Tezi, Zonguldak, 2008.
53. Koyuncugil AS. Borsa şirketlerinin sektörel risk profillerinin veri madenciliğiyle belirlenmesi. Sermaye Piyasası Kurulu Araştırma Raporu, Araştırma Dairesi, Ankara, 2007.
54. Doğan N, Özdamar K. Chaid analizi ve aile planlaması ile ilgili bir uygulama. Türkiye Klinikleri Tıp Bilimleri Dergisi 23(5):392-397, 2003.
55. Baumgartner C, Bohm C, Baumgartner D, Marini G, Weinberger K, Olgemoller B. Supervised machine learning techniques for the classification of metabolic disorders in newborns. Bioinformatics 20(17):2985-2996, 2004.
56. Zhang B, Zhang Y. Classification of cerebral palsy gait by Kernel Fisher Discriminant Analysis. International Journal of Hybrid Intelligent Systems 5(4):209-218, 2008.
57. Cong Y, Yang XG, Lv W, Xue Y. Prediction of novel and selective TNF-alpha converting enzyme (TACE) inhibitors and characterization of correlative molecular descriptors by machine learning approaches. Journal of Molecular Graphics and Modelling 28(3):236-244, 2009.
58. Acharya UR, Vinitha Sree S, Krishnan MM, Molinari F, Garberoglio R, Suri JS. Non-invasive automated 3D thyroid lesion classification in ultrasound: a class of ThyroScan™ systems, Ultrasonics 52(4):508-20, 2012.
59. Hmeidi I, Hawashin B, El-Qawasmeh E. Performance of KNN and SVM classifiers on full word Arabic articles. Advanced Engineering Informatics 22(1):106-111, 2008.
60. Cai YD, Chou KC. Nearest neighbour algorithm for predicting protein subcellular location by combining functional domain composition and pseudo-amino acid composition. Biochemical and Biophysical Research Communications 305(2):407-411, 2003.
61. Saeedmanesh M, Izadi T, Ahvar E. HDM: A Hybrid data mining technique for stock exchange prediction. In: International MultiConference of Engineers and Computer Scientists (IMECS), March 17-19, Hong Kong, 2010.

62. Koltan Yılmaz Ş, Albayrak AS. Veri madenciliği: Karar ağacı algortimaları ve İMKB verileri üzerine bir uygulama. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi 14(1):31-52, 2009.
63. Hand DJ. Classifier technology and the illusion of progress. Statistical Science 21(1):1-15, 2006.
64. Salzberg L. Methodological Note On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach; Data Mining and Knowledge Discovery 1. pp. 317-328, Kluwer Academic Publishers, Boston, UK, 1997.
65. Michie D, Spiegelhalter DJ, Taylor CC. Machine Learning, Neural and Statistical Classification. pp. 131-174, Prentice Hall, New York, USA, 1994.

8. EKLER

Ek 1: Etik Kurul Onayı

T.C.
ZONGULDAK KARAELMASÜNİVERSİTESİ
Klinik Araştırmalar Etik Kurul Başkanlığı

TOPLANTI TARİHİ : 06/03/2012
TOPLANTI NO : 2012/05

KARARLAR :

2- ZKÜ Tıp Fakültesi Biyoistatistik Anabilim Dalı öğretim üyesi Prof. Dr. Vildan SÜMBÜLOĞLU'nun sorumluluğunda yapılacak olan 2012-10-21/02 Protokol no'lu "K-En Yakın Komşuluk, Yapay Sinir Ağları ve Karar Ağaçları Yöntemlerinin Sınıflandırma Başarılarının Karşılaştırılması" konulu çalışmasının; Etik Kurallara uygunluğuna,

Oy birliği ile karar verilmiştir.

A S L I G İ B İ D İ R

Doç.Dr. Banu DOĞAN GÜN
Z.K.Ü. Klinik Araştırmalar Etik Kurul Başkanı

Z.K.Ü. Klinik Araştırmalar Etik Kurulu, 67600 KOZLU/ ZONGULDAK, Tel:0 372 261 32 60 Fax: 0 372 261 02 65

ÖZGEÇMİŞ

1973 yılında Zonguldak Çaycuma'da doğdu. İlk, orta ve lise öğrenimini Çaycuma'da tamamladıktan sonra 1992 yılında girdiği Hacettepe Üniversitesi Fen Fakültesi İstatistik Bölümü'nden 1996 yılında mezun oldu. 1997-2002 yılları arasında özel sektörde, 2006-2007 yılları arasında TÜİK'de çalıştı. 2004-2007 yılları arasında Ankara Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı'nda yüksek lisans öğrenimini tamamladı. 2007 yılında Karaelmas Üniversitesi Tıp Fakültesi Biyoistatistik Anabilim Dalı'nda öğretim görevlisi olarak çalışmaya başladı. Halen aynı üniversitede öğretim görevlisi olarak çalışmaya devam etmektedir.