ORTAÖĞRETİME GEÇİŞ SİSTEMİ (OGES) YERLEŞTİRME PUANLARININ UZMAN SİSTEMLER İLE TAHMİNİ

2013
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

EMİNE UÇAR

ORTAÖĞRETİME GEÇİŞ SİSTEMİ (OGES) YERLEŞTİRME PUANLARININ UZMAN SİSTEMLER İLE TAHMİNİ

Emine UÇAR

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Doktora Tezi
Olarak Hazırlanmıştır

KARABÜK
Ocak 2013

Emine UÇAR tarafından hazırlanan "ORTAÖĞRETİME GEÇİŞ SİSTEMİ (OGES) YERLEŞTİRME PUANLARININ UZMAN SİSTEMLER İLE TAHMİNİ" başlıklı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Baha ŞEN .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 17/01/2013

Unvanı, Adı SOYADI (Kurumu)		İmzası
Başkan :	Prof. Dr. Abdullah ÇAVUŞOĞLU (KBÜ)	.....
Üye :	Prof. Dr. Mehmet AKBABA (KBÜ)	.....
Üye :	Prof. Dr. Fatih Vehbi ÇELEBİ (YBÜ)	.....
Üye :	Doç. Dr. Haldun GÖKTAŞ (YBÜ)	.....
Üye :	Yrd. Doç. Dr.Baha ŞEN (KBÜ)	.....

...../..../2013

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Nizamettin KAHRAMAN .....
Fen Bilimleri Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Emine UÇAR

ÖZET

Doktora Tezi

ORTAÖĞRETİME GEÇİŞ SİSTEMİ (OGES) YERLEŞTİRME PUANLARININ UZMAN SİSTEMLERLE TAHMİNİ

Emine UÇAR

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Yrd. Doç. Dr. Baha ŞEN
Ocak 2013, 127 sayfa

Bilgi miktarının büyük oranlarda arttığı bu bilgi çağında büyük hacimlerdeki verilerden anlamlı bilgilerin elde edilmesi bir süreç gerektirmektedir. Bu sürecin en önemli adımı ise veri madenciliğidir. Veri madenciliği ise önceden bilinmeyen ilişki ve eğilimlerin bulunması için büyük miktarlardaki veriyi analiz eden ve kullanıcılar için anlamsız bilgiyi anlamlı hale dönüştüren bir yöntemdir.

Çalışmada OGES puanlarının tahmini için ilköğretim 8. sınıf öğrencilerinden rastgele seçilen 25000 kayıt kullanılmıştır. Veri madenciliği uygulaması için, OGES yerleştirme puanlarının tahmin edilmesinde kullanılacak uzman sistem tasarlanmıştır. Çalışmada puanların tahmin edilmesi amaçlandığı için sınıflandırma ve öngörü konusunda en çok tercih edilen veri madenciliği tekniklerinden yapay sinir ağları, regresyon analizi, C4.5 karar kuralı türetme algoritması ve destek vektör makineleri

modelleme yöntemi olarak seçilirken, bu dört yöntemin doğruluk oranları ve performansları karşılaştırılarak, en uygun yöntem bulunmaya çalışılmıştır.

Anahtar Sözcükler **:** OGES, uzman sistemler, veri madenciliği.
Bilim Kodu **:** 902.1.014

ABSTRACT

Ph. D. Thesis

PLACEMENT SCORE ESTIMATION OF SECONDARY EDUCATION TRANSITION SYSTEM (SETS) USING EXPERT SYSTEMS

Emine UÇAR

Karabük University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Baha ŞEN
January 2013, 127 pages

Obtaining meaningful information from a large volume of data requires a process in today's information age in which enormous amount of data increases considerably. The most important step of this process is data mining. Data mining is a method that analyses large scale of data to find unknown relationships and trends and converts meaningless data into meaningful information for the users.

In this study, randomly selected 25000 records of elementary school students in the 8^{th} degree have been used to estimate SETS scores. An expert system has been designed which is going to be used to estimate SETS placement scores as a data mining application. As it is aimed to estimate scores in this study, most preferred data mining techniques on classification and prediction such as artificial neural networks, regression analysis, C4.5 decision rule deriving algorithm and support

vector machines have been selected as modelling methods, besides, it is intended to find out the most appropriate method by comparing their accuracy rates and performances of these four methods.

Key Words **:** SETS, expert systems, data mining.
Science Code **:** 902.1.014

TEŞEKKÜR

Doktora öğrenimim süresince bana her türlü desteği veren, yorumları ve yönlendirmeleriyle ufkumu açan, pozitif yapısıyla motivasyonumu üst düzeyde tutan, tez danışmanım, değerli hocam Yrd. Doç. Dr. Baha ŞEN'e,

Tez izleme komitemde yer alan, farklı bakış açıları sunarak hep daha iyiye gitmem için beni teşvik eden, bilgi birikimi ve yardımlarını esirgemeyen değerli hocalarım Prof. Dr. Abdullah ÇAVUŞOĞLU ve Prof. Dr. Fatih Vehbi ÇELEBİ'ye,

Ayrıca bu günlere gelmemde çok büyük emekleri olan, bana maddi manevi her türlü desteği sağlayan anne ve babama, varlıklarıyla hayatıma renk katan kardeşlerime ve son olarak varlığından ve sevgisinden güç aldığım, her konuda bana destek olan sevgili eşim Murat UÇAR' a sonsuz teşekkürler.

Bu çalışmamı, biricik oğlum Ege UÇAR'a atfediyorum.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

OGES : Ortaöğretime Geçiş Sistemi
SBS : Seviye Belirleme Sınavı
YSA : Yapay Sinir Ağları
RTF : Radyal Tabanlı Fonksiyon
DVM : Destek Vektör Makineleri
KA : Karar Ağaçları
LRA : Lojistik Regresyon Analizi
YBP : Yılsonu Başarı Puanı
ÇKA : Çok Katmanlı Algılayıcı
BKD : Bire Karşı Diğerleri
BKB : Bire Karşı Bir

BÖLÜM 1

GİRİŞ

Çalışmamın amacı, Milli Eğitim Bakanlığı, Bilgi İşlem Grup Başkanlığı, e-okul sistemi veri tabanından rastlantısal olarak seçilen 25.000 ilköğretim okulu 8. sınıf öğrencisine ait bilgilerle Ortaöğretime Geçiş Sistemi (OGES) yerleştirme puanlarının veri madenciliği teknikleri kullanılarak tahmin edilmesi, kullanılacak olan bu tekniklerin öngörü modeli olarak başarı performanslarının karşılaştırılması ve elde edilen sonuçlarla tekniklere ilişkin farklılıkları tartışarak, hangi tekniğin hangi koşullarda uygulanmasının daha etkin olacağına yönelik önerilerde bulunmaktır. Birinci bölümde, veri madenciliği kavramı ve gelişimi, veri madenciliği süreci, veri madenciliğinin fonksiyonları ve veri madenciliğinin sınıflandırma fonksiyonu üzerinde durulacak, sınıflandırma tekniklerinden yapay sinir ağları, regresyon analizi, C4.5 karar kuralı türetme algoritması ve destek vektör makineleri teorik olarak incelenecektir.

İkinci bölümde, ilköğretim sekizinci sınıf öğrencilerine ait kişisel bilgiler ve ders notu bilgileri ile oluşturulacak veri tabanı üzerinde veri madenciliğinin sınıflandırma fonksiyonuna ilişkin teknikleri için veri madenciliği sürecinin tüm aşamalarını kapsayan bir uygulama gerçekleştirilmesi planlanmıştır. Çalışmada materyal olarak ilköğretim okulu öğrencilerinin SBS sınavında sorusu bulunan Türkçe, Matematik, Fen ve Teknoloji, Yabancı Dil, Sosyal Bilgiler, Din Kültürü ve Ahlak Bilgisi ve T.C. İnkılâp Tarihi ve Atatürkçülük derslerine ait notları ile SBS puanları kullanılmıştır. Yine öğrencinin sınavdaki başarısına etki edeceği düşünülen okul türü (devlet okulu/özel okul) ile öğrenciye ait bazı kişisel bilgiler (cinsiyet, anne ve babasının medeni durumu, kardeş sayısı, kendi odası olup olmadığı, okul dışında çalışıp çalışmadığı ve burs alıp almadığı) de kullanılmıştır. Yöntem olarak tahmin doğruluklarının karşılaştırılabilmesi için yapay sinir ağı, lojistik regresyon, karar

ağaçları ve destek vektör makineleri modelleri oluşturulmuştur. Bu modeller değerlendirilerek uygulanan teknikler karşılaştırma yoluna gidilmiştir.

Çalışmanın sonunda ise bu uygulama sırasında edinilen deneyimlerin, hem eğitim hedefleri hem de veri madenciliği amacı doğrultusunda elde edilen bulguların paylaşılması amacıyla bir sonuç bölümü yer almaktadır.

Türk eğitim sistemi sınav odaklıdır. Bu nedenle yerleştirme sınavlarında öğrencilerin başarısı veya başarısızlığına yol açan faktörleri incelemek ilgi çekici fakat zorlu bir problemdir. Merkezi yerleştirme sınavları ve akademik başarı birbirleriyle ilişkili kabul edildiğinden beri bu sınavlardaki başarı faktörlerinin analizi akademik başarıyı artırmamıza ve anlamamıza yardımcı olabilir.

Öğrenci velileri bakımından büyük önem taşıyan OGES' de hangi özelliklere sahip olan öğrencilerin başarılı olduğunun ortaya çıkarılmasının öğrenciler, veliler, öğretmenler, yöneticiler ve araştırmacılar bakımından yararlı olacağı düşünüldüğünden çalışmada amaç öğrenci başarısı üzerinde etkili olduğu düşünülen değişkenlerin önem düzeylerinin belirlenerek eğitim hedefleri doğrultusunda bu kişilere yön vermektir.

1.1. LİTERATÜR

Öğrencilerin akademik başarılarını etkileyen etmenleri (örneğin yorumlayıcı değişkenler) anlamak eğitim yapısını anlama ve onu geliştirmede kritik bir girdidir. Önceki çalışmaların çoğu bu olayı her seferinde bir değişkenli olarak analiz etmişlerdir. Tek bir etmen ile onun akademik başarıya etkisi arasındaki ilişkiyi anlamak amacıyla çoğunlukla anket tipi araçlardan veri toplamaya çalışmışlardır. Örneğin, bazı araştırmacılar akademik başarı ile ebeveynlik stilleri arasındaki bağıntıyı incelemişler [1,2], diğerleri sosyoekonomik durumları üzerine odaklanmışlardır [3]. Bazıları farklı okul türlerinin önemine değinirken [4], diğer bazıları da öğretmen yardımının akademik başarıya etkisini araştırmışlardır [5]. Davranışsal tarafta bazı araştırmacılar kişisel kontrol algısını incelemişler [6], diğerleri ise eğitmenlerin verimliliğini [7], cinsiyeti [8] ve kontrol odağını [9]

incelemişlerdir. Bu çalışmaların bazıları çalışmanın temellendirildiği kısıtlı veriler üzerinde bireysel faktörler ve akademik başarı arasında istatistiksel olarak güçlü bağlar bulmuşlardır.

Bazı araştırmacılar merkezi yerleştirme sınavlarındaki başarı seviyesini incelemişlerdir. Bazıları alınan puanlar ve ailenin geliri arasında güçlü bir ilişki bulduklarını iddia etmişlerdir. Buldukları sonuçlar aile gelirinin sınav puanlarına ve ilerideki akademik başarılara pozitif yönde etkiye sahip olduğunu göstermiştir [10]. Diğer taraftan, diğer bazı araştırmacılar gelirin akademik başarıyı doğrudan etkilemediğini fakat aile tutumunu ve inançlarını etkilemek suretiyle nihai sonuca etki ettiğini belirtmişlerdir [11]. Örneğin, yüksek gelire sahip aileler çocuklarını özel okullara gönderebilir ve onları ekstra eğitimlerle destekleyebilir ki sonunda onları daha yüksek bir akademik başarıya taşıyabilirler.

Standart sınavlar birçok kurum için öğrencilerin potansiyel akademik başarılarının değerlendirilmesinde anahtar bileşen olmuştur. Konuyla ilgili geçmiş çalışmalara ait ön incelemeler göstermiştir ki genellikle standart test puanlarının gerçekçi tahmini nispeten keşfedilmemiş bir alandır. Yarı ilişkili çalışmaların çoğu böyle standart sınav puanlarının tahmin değerini içerir; diğer bir deyişle, bu tip çalışmalar genellikle gerçek puanları tahmin etmekte titiz analitik teknikler (örn. Veri madenciliği/ veya istatistiksel analiz) kullanmazlar fakat bunun yerine öğrencilerin beklenen performansını zaten bilinen puanlarına göre ölçme girişiminde bulunurlar [12-14]. Bu tür çalışmalarda zahmetli olmasına rağmen yine de kayda değer olan gerçek şudur ki, tek başına standart sınav son derece tartışmalıdır. SAT (Skolâstik Yetenek Testi) gibi sınavların geçerliliği akademide çekişmenin ana kaynağı olmuştur: önemli mesele testin uygulanabilirliğinin, kapsamının ve adaletinin olmasıdır. Yine, bazıları tahmini içeren birçok analitik çalışma bu nedenle test puanları ile akademik başarı, dereceler, toplumsal istatistikler, sınav ortamı vb. arasındaki ilişkiyi anlamayı araştırır. Yıllarca sadece bu konu üzerine yapılmış olan araştırmalara rağmen, ilgili çalışmaların genel özeti göstermektedir ki cevap hala belirsizliğini korumaktadır.

Veri madenciliği üzerine basılan yayınlara bakılacak olursa; basılan ilk kitap Piatetsky-Shapiro ve Frawley tarafından yazılmış olup, 1989 yılında gerçekleştirilen

bir seminerdeki makalelerin bir araya getirilmesi ile oluşturulmuştur [15]. 1994 yılında yapılan bir seminere ait makalelerden yola çıkarak hazırlanan bir diğer kitap Fayyad vd. tarafından yazılmıştır.

Veri madenciliği ile ilgili olarak yayınlanan sonraki kitaplar, işin teorisinden çok pratiğiyle ilgili ve doğrudan iş odaklı olarak hazırlanmıştır. Bunların arasında Syllogic firmasından Adrians ve Zantige veri madenciliği hakkındaki ilk çalışmalardan birini hazırlamışlardır [16]. IBM firmasında çalışan Cabena vd. tarafından yapılan çalışmada ise gerçek yaşamdaki uygulama örnekleri ile birlikte veri madenciliği süreç ve yöntemleri incelenmiştir [17].

Han ve Kamber, büyük ve birleştirilmiş veri tabanlarında bilgi keşfi konusuna odaklı olarak veri madenciliği konusunu, veri tabanı bakış açısı ile incelemişlerdir. Veri madenciliği konusunda söz sahibi yazarların çalışmalarından disiplinler arası bir kitap oluşturmuşlardır [18].

Mohammadian, yazmış olduğu kitabında, hem internet, hem de veri tabanlarının olağanüstü büyümesi nedeniyle iyice karmaşıklaşan anlamlı bilgilerin elde edilmesi süreci için akıllı sistemlere ihtiyaç duyulduğunu belirtmiştir. Bu amaçla, dünya üzerinde akıllı ajanlar konusunda çalışmalar yapan uluslar arası araştırmacıların çalışmalarını kitabında toplamıştır [19].

Soukup ve Davidson ham verinin, işletmelerin yararlanabileceği veri kümeleri haline dönüştürülmesini ve sonrasında bu veri kümelerinin görsel veri madenciliği yöntemleri kullanılarak analiz edilmesini incelemişlerdir [20].

Keim, hazırlamış olduğu sunumda, özellikle görsel veri keşfi sürecinde kullanılan yöntemleri bir araya getirmiştir. İncelediği görsel veri keşfi teknikleri arasında; geometrik teknikler, ikona tabanlı teknikler, piksel tabanlı teknikler, hiyerarşik teknikler, grafik tabanlı teknikler ve hibrid teknikler vardır [21].

Venkayala Java geliştiricileri dergisinde yayınlanan makalesinde, Java Data Mining (JDM)1.0 standardını açıklamıştır. Kendisi, JSR-73 altında geliştirilen JDM

standardı uzman geliştiricilerindendir. Venkayala, makalesinde, Java yazılım dili ile veri madenciliği yapabilmenin standardı olan JDM'in pratikte nasıl kullanılabileceği konusunu detaylı olarak incelemiştir [22].

Wang, 2001 yılında konusunda uzmanlaşmış kişilere yaptığı çağrısı sonucunda, veri madenciliği ile ilgili yeni teorilerden uygulamalara kadar çok geniş bir yelpazede topladığı makaleler üzerinde yaptığı bir buçuk yıllık titiz bir çalışma sonucunda bir kitap oluşturmuştur [23].

Wang, 2003 yılında hazırlamış olduğu ve çeşitli makaleleri topladığı çalışmasından sonra, 2006 yılında hazırlamış olduğu geniş içerikli kitabında, veri madenciliği ve veri ambarı konusunda uzmanlaşmış, toplam 358 uluslararası araştırmacının makalelerine yer vermiştir [24].

Tang ve MacLennan, SQL sunucu yüklü ortamlarda veri madenciliğinin nasıl yapılabileceği konusunu incelemişlerdir. Microsoft firmasının bir ürünü olan ve dünya üzerinde yoğun olarak kullanılan Microsoft SQL sunucu veri tabanlarında, veri madenciliği tekniklerinden olan Naive Bayes, karar ağaçları, zaman serileri, kümeleme, birliktelik kuralları, yapay sinir ağları ile veri madenciliği yapılması anlatılmıştır [25].

Weiss vd., veri madenciliği üzerine yapılan en önemli çalışma atölyelerinden biri olan K.D.D. 2005'te sunulan çalışmaları, eserlerinde toplamışlardır. Atölyede, hem veri madenciliği, hem de makine öğrenmesi üzerine birçok uygulamalar sunulmuştur [26].

Mattison, son yıllarda gelişen telli iletişim alanında veri ambarlama ve veri madenciliği araştırmaları yapmıştır. Kitabında, veri yöntemlerinden olan yapay sinir ağları ve coğrafi veri madenciliği ile telli iletişim alanında çalışan işletmelerin değer ve bilgi elde etme yollarını çeşitli uygulamalarla açıklamıştır [27].

Keogh vd., değiştirgesiz (parametresiz) veri madenciliği konusunu ortaya atmışlardır. Çoğunluk veri madenciliği algoritmaları, başlangıçta birçok değiştirgenin düzgün bir

şekilde ayarlanmasına ihtiyaç duymaktadır. Keogh vd. makalelerinde değiştirgesiz veri madenciliğinin nasıl yapılabileceğini göstermişlerdir [28].

Mitra ve Acharya, veri madenciliğinin sınıflandırma, kümeleme ve benzer gruplama gibi geleneksel kavram ve fonksiyonlarının yanında, özellikle çoklu ortam (multimedia) ve bilgisayar destekli biyoloji (bioinformatics) alanlarında veri madenciliği yapılması konularına odaklanmışlardır [29].

Berry ve Linoff, ilki 1997 yılında yayımlanan kitaplarının ikinci basımında veri madenciliği konusunu genel olarak üç ayrı kısımda incelemişlerdir. Birinci kısımda; veri madenciliğini tanıtan ve niçin gerekli olduğunu vurgulayan bölümü de içerecek şekilde işletmeler açısından veri madenciliğinin anlamı anlatılmıştır. İkinci bölümde; verinin bilgi haline getirilmesi için hangi durumlarda, hangi veri madenciliği tekniklerinin kullanılması gerektiği detaylandırılmıştır. Üçüncü ve son bölümde ise; veri madenciliği yöntemleri ile ilgili en iyi uygulama alanları, örneklendirilerek anlatılmıştır [30].

Witten ve Frank, Yeni Zelanda'nın Waikato Üniversitesi bilgisayar bilimleri bölümünde çalışan iki öğretim görevlisidir. Yazmış oldukları kitap iki açıdan çok önemlidir. Birincisi, veri madenciliği konusuna uygulanabilirlik açısından baktıklarından bir başucu kitabı olmasıdır. İkincisi ise, bu kitabın yazarlarının, veri madenciliği arenasında ve özellikle akademik ortamlarda çok sık kullanılan WEKA aracını oluşturmuş olmalarıdır [31].

BÖLÜM 2

VERİ MADENCİLİĞİ

Veri madenciliği ile ilgili birçok tanım bulunmaktadır. Bu tanımların tercih edilişleri ve tarihsel sıralamaları açısından farklı kaynaklarda en fazla öne çıkanları aşağıdaki gibidir.

"Veri madenciliği, veri içerisinde gizli kalmış, önceden bilinemeyen ve potansiyel olarak kullanışlı olan anlamlı bilginin çıkarımıdır [15]".

"Veri madenciliği, içerisinde var olan anlamlı örüntü ve kuralları ortaya çıkarmak amacıyla, büyük miktarlardaki verinin otomatik ve yarı otomatik araçlar yardımıyla incelenmesi ve analiz edilmesi sürecidir [30]".

"Veri madenciliği, çeşitli mimarilerde depolanmış olan büyük miktarlardaki verilerden ilgi çekici bilginin keşfedilmesi sürecidir [18]".

"Veri madenciliği, veri ambarında tutulan çok çeşitli verilere dayanarak daha önce keşfedilmemiş bilgileri ortaya çıkarma ve bu bilgileri, karar vermek ve eylem planını gerçekleştirmek için kullanma sürecidir [32]".

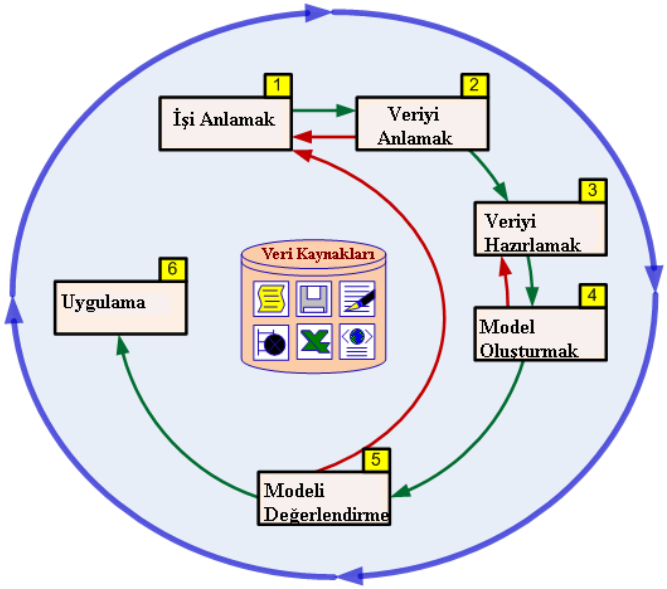
"Veri madenciliği, veriye sahip olan kişi ya da kurum için, kuralları ve ilişkileri keşfederek, önceden bilinemeyen açık ve yararlı sonuçlar elde etmek amacıyla, çok miktardaki verinin seçilmesi, incelenmesi ve modellenmesi sürecidir [33]".

Ancak tüm bu tanımlama çabaları incelendiğinde, "çok fazla miktarda veri" ve "anlamlı bilgi çıkarımı" ifadeleri dikkat çekici bir şekilde ön plana çıkmaktadır. Yine bu tanımlamalardan hareketle kolayca görülmektedir ki, veri madenciliği belirli bir teknik ya da algoritma olmaktan çok bir süreç ifade etmektedir [34].

2.1. VERİ MADENCİLİĞİ SÜRECİ

Veri madenciliği süreci, temel bir takım ortak faaliyetleri kapsamakla birlikte, başlangıçta her kullanıcı tarafından farklı şekilde yürütülmekteydi. Bu nedenle, uygulamada ortaya çıkan farklılıkları en aza indirecek bir standart süreç geliştirilmesi ihtiyacı ortaya çıkmıştır. Bu ihtiyaç doğrultusunda, veri madenciliği uygulamalarının dört önemli lideri ile iki yüzden fazla araştırmacıyı bir araya getiren CRISP–DM konsorsiyumu, düzenlediği çeşitli çalıştaylarla veri madenciliği uygulamalarının olgunluk kazanması ve tüm kullanıcılara yol gösterecek bir standart süreç modelinin oluşturulması için 1996 yılının sonlarında çalışmalarına başlamış ve 2000 yılında bu çalışmayı yayınlamışlardır [35].

CRISP–DM modeli, veri madenciliği sürecini altı temel aşamadan oluşan bir yaşam çemberi olarak ifade etmektedir (Şekil 2.1). Model, her bir aşamanın tamamlanması ile yeni bir aşamaya geçilmesini ya da gerektiğinde önceki aşamalara dönülmesini önermektedir.



Şekil 2.1. CRISP-DM modeli.

2.1.1. İşi Anlamak

Veri madenciliği projesinin belki de en önemli aşaması proje hedeflerini iş perspektifi ile anlamayı, bu bilgiyi veri madenciliği problem tanımına dönüştürmeyi

ve ilgili hedeflere ulaşmak için bir proje planı oluşturmayı kapsayan işi anlama aşamasıdır. İşi anlama, bir başlangıç aşaması olarak, iş perspektifi bakımından projenin hedef ve ihtiyaçlarını anlamaya odaklanmayı ve buradan edinilen bilgi ile veri madenciliğinin problem tanımını oluşturarak, bir önsel plana dönüştürmeyi kapsamaktadır [36].

2.1.2. Veriyi Anlamak

Veriyi anlamak, başlangıç verilerinin elde edilmesini, verileri tanımaya dönük analizleri ve verilerde saklı olabilecek bilgi için ilk izlenimlerin oluşturulmasını kapsayan bir aşamadır. Bu aşama, verilerin elde edilmesinden başlayarak verinin genel yapısı hakkında fikir geliştirmeye, veri kalitesi problemlerini tespit etmeye ve hatta veri içerisinde ilginç olabilecek alt kümeleri tespit etmeye kadar birçok çalışmayı kapsamaktadır.

2.1.3. Veriyi Hazırlamak

Veriyi hazırlamak veri madenciliğinin en önemli aşamalarından biridir ve projenin tablo, değişken ve kayıt seçimi, birleştirme ve temizleme işlemleri gibi faaliyetleri bu aşamada gerçekleştirilmektedir.

Verileri Toplama: Tanımlanan iş için gerekli olduğu düşünülen verilerin ve bu verilerin toplanacağı veri kaynaklarının belirlenmesi adımıdır.

Verileri Birleştirme ve Temizleme: Bu adımda toplanan verilerde bulunan farklılıklar giderilmeye çalışılır. Hatalı veya analizin yanlış yönlenmesine sebep olabilecek verilerin temizlenmesine çalışılır. Eksik veriler tamamlanır ya da silinir.

Dönüştürme: Kullanılacak model ve algoritma çerçevesinde verilerin tanımlama veya gösterim şeklinin de değiştirilmesi gerekebilir. Örneğin öğrenci cinsiyetleri, okul türleri ve bursluluk durumu gibi değişkenlerin kodlanarak gruplanması faydalı olacaktır.

2.1.4. Model Oluşturma

Bu aşamada, modelleme tekniğinin seçilmesi ve bu modellere ilişkin parametrelerin ayarlanması gibi çalışmalar yapılmaktadır. Tanımlanan iş için en uygun modelin bulunabilmesi, çok sayıda modelin kurularak denenmesi ile mümkündür. Bu nedenle veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele varılıncaya kadar yinelenen bir süreçtir.

Model kurma süreci denetimli (supervised) ve denetimsiz (unsupervised) öğrenmenin kullanılmasına göre farklılık göstermektedir. Denetimli öğrenme sürecinde veri kümesinde önceden tanımlanmış sınıflara ilişkin özelliklerin belirlenmesi ve bu özelliklerin kural cümleleri ile ifade edilmesi hedeflenmektedir. Süreç tamamlandığında bu kural cümleleri yeni verilere uygulanmakta ve bu verilerin hangi sınıfa ait olduğu öngörülmektedir. Denetimsiz öğrenmede ise verilerin benzerliklerinden ya da uzaklıklardan hareketle ait oldukları sınıfların üretilmesi amaçlanmaktadır [37].

2.1.5. Modeli Değerlendirme

Elde edilen modelin konuşlandırılması öncesinde model oluşum sürecinin dikkatlice gözden geçirilmesini ve modelin iş hedeflerini başarma konusundaki yeterliliğinin değerlendirilmesini içeren değerlendirme aşaması yer almaktadır. Bu aşamada, veri madenciliği sonuçlarının nasıl kullanılacağına da karar verilmesi gerekmektedir [36].

Bir modelin doğruluğunun test edilmesinde kullanılan en basit yöntem basit geçerlilik (simple validation) testidir. Bu yöntemde tipik olarak verilerin %5 ile %33 arasındaki bir kısmı test verileri olarak ayrılır ve kalan kısım üzerinde modelin öğrenimi gerçekleştirildikten sonra, bu veriler üzerinde test işlemi yapılır. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının, tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile ise doğruluk oranı hesaplanır. (Doğruluk Oranı = 1 - Hata Oranı)

Önemli diğer bir değerlendirme kriteri modelin anlaşılabilirliğidir. Bazı uygulamalarda doğruluk oranlarındaki küçük artışlar çok önemli olsa da, birçok işletme uygulamasında ilgili kararın niçin verildiğinin yorumlanabilmesi çok daha büyük önem taşıyabilir. Çok ender olarak yorumlanamayacak kadar karmaşıklaşsalar da, genel olarak karar ağacı ve kural temelli sistemler model tahmininin altında yatan nedenleri çok iyi ortaya koyabilmektedir [38].

2.1.6. Uygulama

Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama olabileceği gibi, bir başka uygulamanın alt parçası olarak kullanılabilir. Genellikle bir modelin oluşturulması yeterli olmamakta, kullanıcı için organize edilmesi ve sunulması da gerekmektedir. Bu açıdan konuşlandırma aşaması, gerekliliklere bağlı olarak sadece bir raporlama işlemi kadar basit olabileceği gibi canlı sistemler tarafından tekrarlanabilir bir süreç olarak uygulanması kadar karmaşık da olabilir. Ancak bu tercih, veri analizcisinden çok kullanıcının karar vermesini gerektiren bir tercih olup, analizci ile kullanıcının farklı olduğu durumlarda kullanıcının modeli ve modelin ihtiyaçlarını yeterli seviyede kavraması gibi bir görevi de beraberinde getirir [36].

2.2. VERİ MADENCİLİĞİ MODELLERİ

Veri madenciliğinde kullanılan modeller, tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki ana başlık altında incelenmektedir.

Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır.

Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntü1erin tanımlanması sağlanmaktadır [39].

Veri madenciliği modellerini gördükleri işlevlere göre,

- Sınıflama ve regresyon,

- Kümeleme,
- Birliktelik kuralları ve ardışık zamanlı örüntüler, olmak üzere üç ana başlık altında incelemek mümkündür. Sınıflama ve regresyon modelleri tahmin edici, kümeleme, birliktelik kuralları ve ardışık zamanlı örüntü modelleri tanımlayıcı modellerdir.

2.2.1. Sınıflama ve Regresyon Modelleri

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen veri analiz yöntemleridir. Sınıflama kategorik değerleri tahmin ederken, regresyon mevcut değerleri kullanarak diğer değerlerin ne olacağını tahmin etmeye çalışır.

Örneğin, bir sınıflama modeli banka kredi uygulamalarının güvenli veya riskli olmalarını kategorize etmek amacıyla kurulurken, regresyon modeli geliri ve mesleği verilen potansiyel müşterilerin bilgisayar ürünleri alırken yapacakları harcamaları tahmin etmek için kurulabilir [40].

Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler şunlardır [41] :

- Karar ağaçları (Decision Trees),
- Yapay sinir ağları (Artificial Neural Networks),
- Genetik algoritmalar (Genetic Algorithms),
- K-en yakın komşu (K-Nearest Neighbor),
- Lojistik regresyon (Logistic Regression)'dur.

2.2.2. Kümeleme Modelleri

Kümeleme analizi birinci amacı "gözlem birimlerini benzer özelliklerine göre gruplamak" olan çok değişkenli istatistik tekniklerden biridir. Kümeleme analizinde elde edilen bir küme içindeki gözlem birimleri, önceden belirlenmiş bir özellik bakımından birbirine benzemektedir. Dolayısıyla elde edilen kümedeki gözlem birimleri homojendir [42].

Kümeleme modellerinde amaç küme üyelerinin birbirine çok benzediği, ancak özellikleri birbirinden çok farklı olan kümelerin elde edilmesi ve veri tabanındaki kayıtların bu farklı kümelere bölünmesidir [43].

2.2.3. Birliktelik Kuralları ve Ardışık Zaman Örüntüleri

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 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 [44].

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

- Düşük yağlı peynir ve yağsız yoğurt alan müşteriler, %85 ihtimalle 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 kullanılır.
- X ameliyatı yapıldığında, 15 gün içinde % 45 ihtimalle Y enfeksiyonu oluşacaktır,
- IMKB endeksi düşerken A hisse senedinin değeri % 15' den daha fazla artacak olursa, üç iş günü içerisinde B hisse senedinin değeri % 60 ihtimalle artacaktır,
- Çekiç satın alan bir müşteri, ilk üç ay içerisinde % 15, bu dönemi izleyen üç ay içerisinde % 10 ihtimalle çivi satın alacaktır.

BÖLÜM 3

VERİ MADENCİLİĞİ YÖNTEMLERİ

Çalışmada puanların tahmin edilmesi amaçlandığı için sınıflandırma ve öngörü konusunda en çok tercih edilen veri madenciliği tekniklerinden yapay sinir ağları, regresyon analizi, C4.5 karar kuralı türetme algoritması ve destek vektör makineleri modelleme yöntemi olarak seçilmiştir.

3.1. YAPAY SİNİR AĞLARI

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zordur veya mümkün değildir [45]

İlk yapay sinir ağı modeli 1943 yılında, McCulloch ve Pitts tarafından gerçekleştirilmiştir. İnsan beyninin hesaplama yeteneğinden esinlenerek, elektrik devreleriyle basit bir sinir ağını modellemişlerdir. 1949 yılında Hebb öğrenme ile ilgili temel teoriyi ele almıştır. 1957 yılında Frank Rosentblatt'ın Perceptron'u (Perceptron bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanmaktadır) gerçekleştirmesinden sonra yapay sinir ağı alanındaki gelişmeler hızlanmıştır. 1959 yılında, Bernand Widrow ve Marcian Hoff Stanford Üniversitesinde Adaline ve Madaline olarak adlandırdıkları ağ modellerini geliştirmişlerdir. 1969 yılında Minsky ve Papert, Perceptronun yetersizliğini görmüşler ve XOR problemini çözemediğini ispatlamışlardır. Perceptron, doğrusal olmayan yapıları belirleyemez, oysa doğrusal olmayan yapılarla hemen hemen her konuda karşılaşılabilmektedir. Bu durum Exclusive OR problemi olarak tanımlanmaktadır [46].

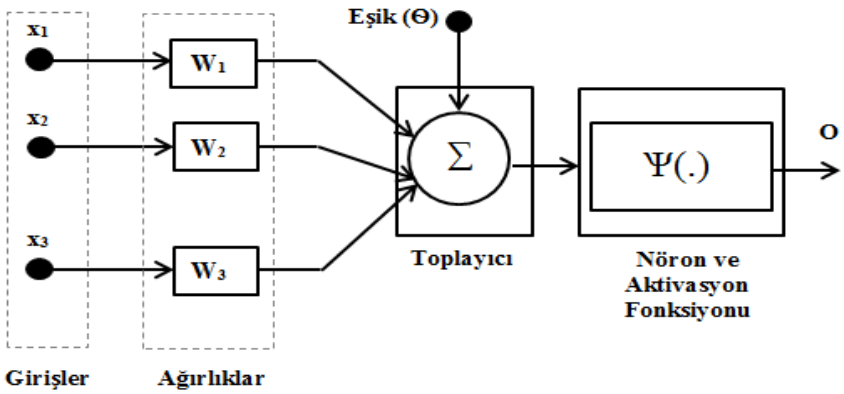
Hopfield 1982 yılında ağların önemli sınıflarının matematik temellerini üretmiş ve Ulusal Bilimler Akademisine bir makale sunmuştur. 1984 yılında Kohonen kendi kendini düzenleyen haritayı (Self-Organizing Maps) tanımlamıştır. 1986 yılında Rumelhart ve McCelland karmaşık ve çok katmanlı ağlar için geriye yayılmalı öğrenme algoritması ortaya koymuştur. 1987 yılında Elektrik Elektronik Mühendisliği Enstitüsü (Institute of Electrical Electronic Engineering-IEEE) tarafından sinir ağlarını konu alan ilk uluslararası konferans 2000'e yakın katılımcıyla San Diego'da gerçekleştirilmiş ve bu konferans yapay sinir ağları disiplininin resmi başlangıcı olarak kabul edilmiştir [47].

3.1.1. Yapay Sinir Hücresi

YSA'nın işleyişi insan sinir hücresine benzemektedir. 1940 yılında McCulloch ve Pitts nöronun, mantık sistemlerinde basit eş değer yapısıyla modellenebileceğini ortaya atmışlardır. Bu amaçla yaptıkları çalışmalar sonunda Şekil 3.1'de görüldüğü gibi bir yapay sinir ağı modeli geliştirmişlerdir [48].

Bir yapay sinir hücresi beş bölümden oluşmaktadır;

- Girdiler
- Ağırlıklar
- Toplama fonksiyonu
- Aktivasyon fonksiyonu
- Çıktılar



Şekil 3.1. Yapay sinir hücresi [48].

Girdiler, nöronlara gelen verilerdir. Girdiler yapay sinir hücresine bir diğer hücreden gelebileceği gibi direk olarak dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir [48, 49].

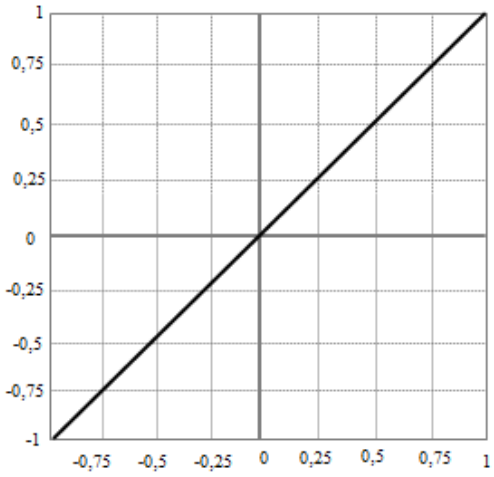
Ağırlıklar, yapay sinir hücresine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan girdilerin çıkış üzerinde herhangi bir etkisi olmamaktadır [49].

Toplama fonksiyonu, Şekil 3.1'de görüldüğü gibi herhangi bir katmandaki i. birime gelen toplam giriş (X_{ij}), ilk katman için girişlerin bağlantılar üzerindeki ağırlıkları (W_{ij}) ile hesaplanmış bir ağırlıklı toplamıdır (NET_j).

$$NET_j = \sum_{i=1}^{N} X_{ij} * W_{ij} \tag{3.1}$$

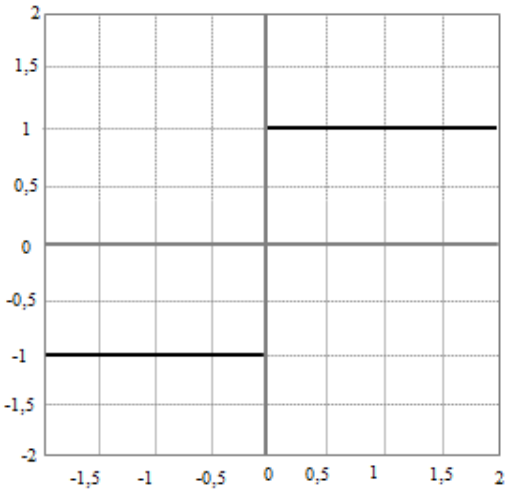
Aktivasyon fonksiyonu, birleştirme (toplama) fonksiyonundan çıkan NET toplam hücrenin çıktısını oluşturmak üzere aktivasyon fonksiyonuna iletilir. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. YSA'nın bir özelliği olan "doğrusal olmama" aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Aktivasyon fonksiyonu seçilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Bir problem için en uygun fonksiyon tasarımcının denemeleri sonucu belirlenebilir. Uygun fonksiyonu gösteren bir formül bulunmuş değildir [45, 49].

Doğrusal aktivasyon fonksiyonu, doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon da seçilebilir. Doğrusal aktivasyon fonksiyonları matematiksel olarak F(x) = A * x olarak genellenebilir. Bu formülde A sabit bir katsayıdır. A değerinin değişimi şekilde gösterilen doğrunun çıkış ekseniyle yaptığı açıyı değiştirmektedir (Şekil 3.2).



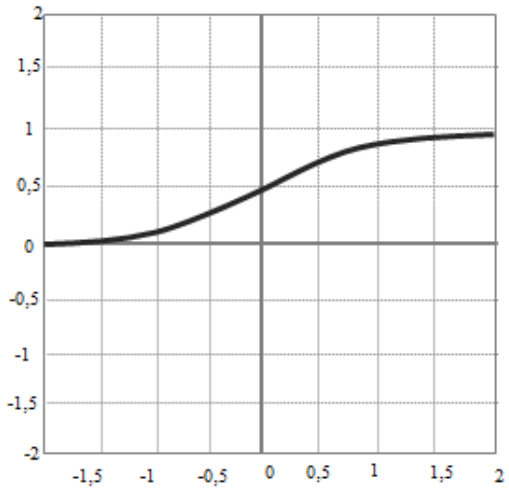
Şekil 3.2. Doğrusal aktivasyon fonksiyonu.

Adım aktivasyon fonksiyonu, girdilerin sıfırdan büyük olup olmamasına göre -1 veya 1 çıktısı veren fonksiyondur. Sadece iki çeşit çıktı vermektedir (Şekil 3.3).



Şekil 3.3. Adım aktivasyon fonksiyonu.

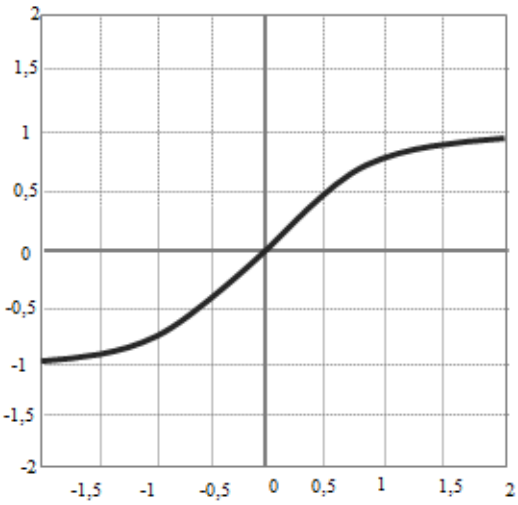
Sigmoid aktivasyon fonksiyonu, sürekli ve türevi alınabilir bir fonksiyondur. Şekil 3.4'de görüldüğü gibi doğrusal olmayışı nedeniyle yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir. Günümüzde en yaygın olarak kullanılan Çok katmanlı algılayıcı modelinde genel olarak sigmoid fonksiyonu kullanılmaktadır. Bu fonksiyon eşitlik 3.2' de gösterilmektedir [45,49].



Şekil 3.4. Sigmoid aktivasyon fonksiyonu.

$$F(NET) = \frac{1}{1+e^{-NET}} \tag{3.2}$$

Tanjant hiperbolik aktivasyon fonksiyonu, Tanjant hiperbolik fonksiyonu, Şekil 3.5' de görüldüğü gibi sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir (Eşitlik 3.3).



Şekil 3.5. Tanjant hiperbolik aktivasyon fonksiyonu.

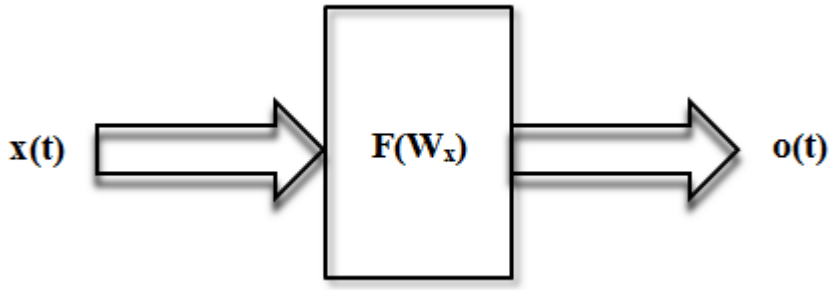
$$F(NET) = \frac{(e^{NET}+e^{-NET})}{(e^{NET}-e^{-NET})} \tag{3.3}$$

Çıktılar, aktivasyon fonksiyonundan çıkan değer nöronun çıktı değeri olmaktadır. Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilir ister tekrardan ağın içinde kullanılabilir. Nöronun bir çıktısı olmasına rağmen bu çıktı istenilen sayıda nörona bağlı olabilir [49].

3.1.2. Yapılarına Göre Yapay Sinir Ağları

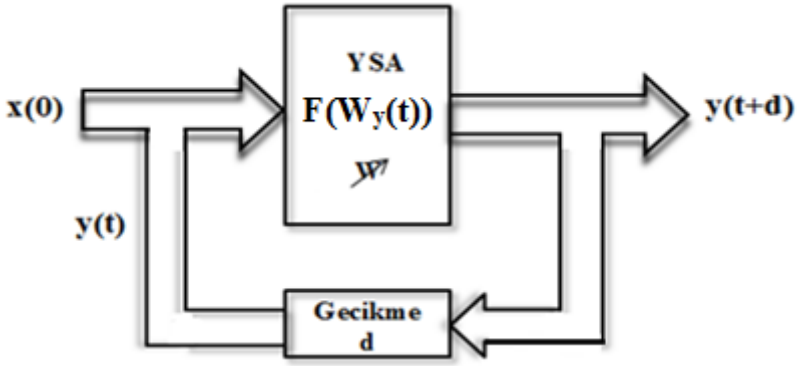
YSA içerdiği nöronların birbirine bağlanış şekline göre ileri ve geri beslemeli olarak ikiye ayrılır.

İleri beslemeli ağlar, ileri beslemeli ağlarda nöronlar girişten çıkışa doğru düzenli katmanlar Şekil 3.6'daki gibidir. 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 [49,50].



Şekil 3.6. İleri beslemeli ağ blok diyagramı.

Geri beslemeli ağlar, geri beslemeli YSA'da 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 YSA doğrusal olmayan dinamik bir davranış göstermektedir. Geri besleme özelliğini kazandıran bağlantıların bağlanış şekline göre geri aynı yapay sinir ağıyla farklı davranışta ve yapıda geri beslemeli YSA elde edilebilir (Şekil 3.7) [49].

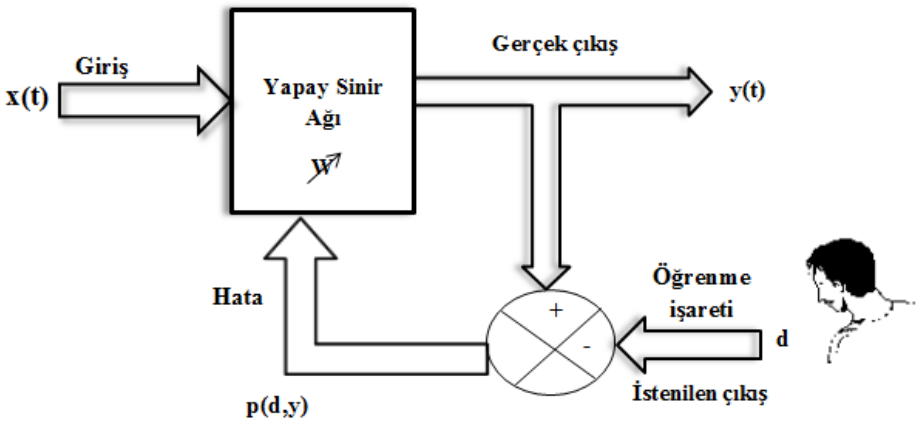


Şekil 3.7. Geri beslemeli ağ blok diyagramı.

3.1.3. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları

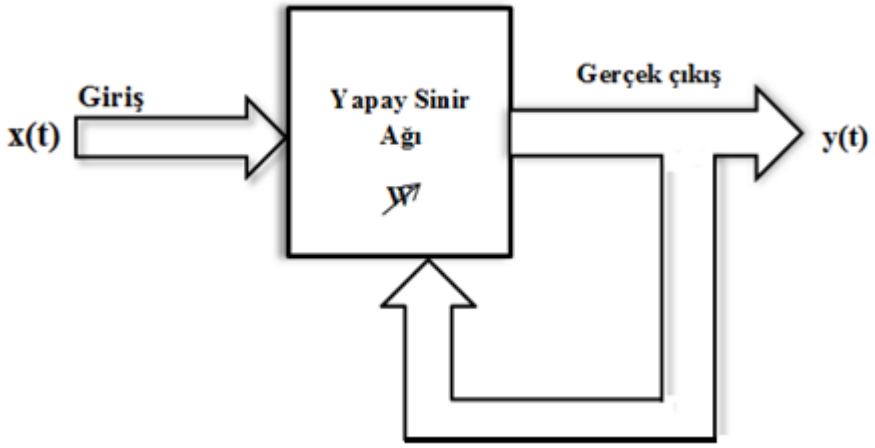
YSA'nın verilen girdilere göre çıktı üretebilmesinin yolu ağın öğrenebilmesidir. Bu öğrenme işleminin de birden fazla yöntemi vardır. YSA öğrenme algoritmalarına göre danışmanlı, danışmansız ve takviyeli öğrenme olarak üçe ayrılır.

Danışmanlı öğrenme, danışmanlı öğrenme sırasında Şekil 3.8'de görüldüğü gibi ağa verilen giriş değerleri için çıktı değerleri de verilir. Ağ verilen girdiler için istenen çıkışları oluşturabilmek için kendi ağırlıklarını günceller. Ağın çıktıları ile beklenen çıktılar arasındaki hata hesaplanarak ağın yeni ağırlıkları bu hata payına göre düzenlenir. Hata payı hesaplanırken ağın bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her nörona düşen hata payı bulunur. Daha sonra her nöron kendine gelen ağırlıkları günceller [45,49,50].



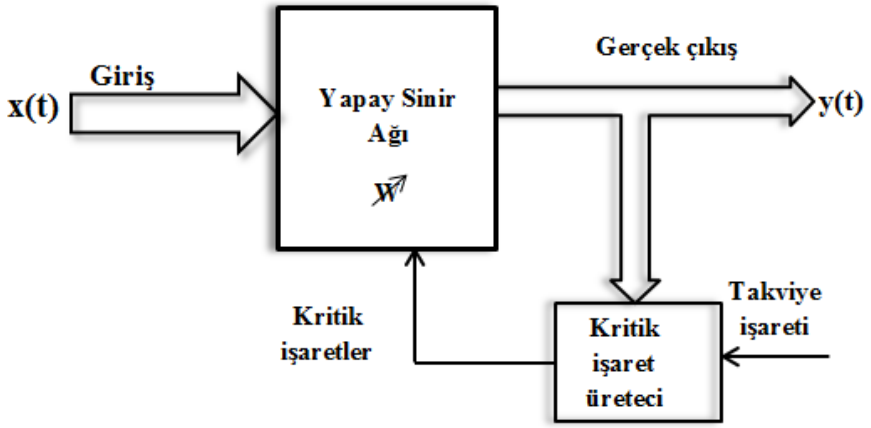
Şekil 3.8. Danışmanlı öğrenme blok diyagramı.

Danışmansız öğrenme, danışmansız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini Şekil 3.9'daki gibi tamamlar [45,49,50].



Şekil 3.9. Danışmansız öğrenme blok diyagramı.

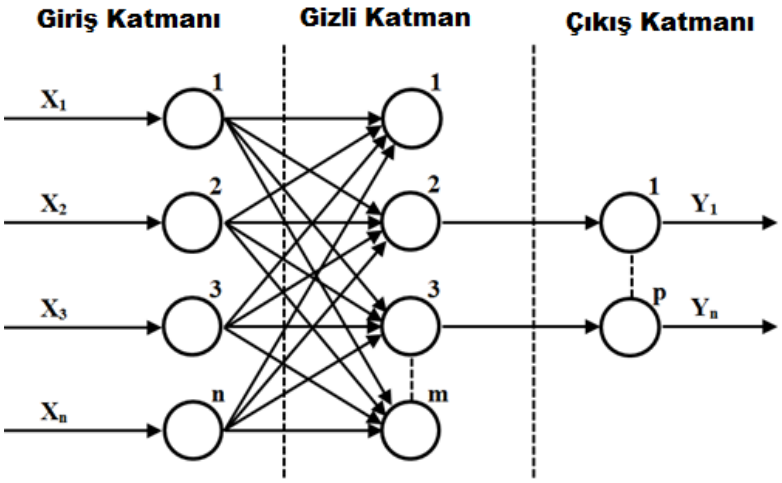
Destekleyici öğrenme, ağın her iterasyon sonunda elde ettiği sonucun iyi veya kötü olup olmadığına dair bir bilgi verilir. Ağ bu bilgilere göre kendini yeniden düzenler. Bu sayede ağ herhangi bir girdi dizisiyle hem öğrenerek hem de sonuç çıkararak Şekil 3.10'daki gibi işlemeye devam eder.



Şekil 3.10. Destekleyici öğrenme blok diyagramı.

3.1.4. Çok Katmanlı Algılayıcı

Bu modelin yapısı Şekil 3.11'de gösterildiği gibidir. Çok katmanlı algılayıcı denetimli öğrenme yöntemini kullanan ve genellikle akademik ya da deneysel sınıflandırma amaçları için kullanılan bir yöntemdir. Birçok eğitim algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir. Birçok katmanlı algılayıcı modeli, bir giriş, bir veya daha fazla ara ve bir de çıkış katmanından oluşur.



Şekil 3.11. Çok katmanlı yapay sinir ağı.

Giriş katmanındaki sinir hücrelerinin sayısı veri kümesindeki değişkenlere bağlı olmaktadır. Çıkış katmanındaki sinir hücrelerinin sayısı ise üzerinde çalışılan sınıflandırma problemine göre değişkenlik göstermektedir. Bununla birlikte, gizli katmanların sayısı ve gizli katmandaki sinir hücrelerinin sayısı kullanıcı tarafından belirlenebilmektedir.

Çok katmanlı algılayıcılarda, bilgi giriş katmanından çıkış katmanına doğru akar. Giriş katmanındaki sinir hücreleri gizli katmandaki hücrelere, gizli katmandaki sinir hücreleri ise çıkış katmanındaki hücrelere bağlanarak öngörüleri üretirler.

Çoğu yapay sinir ağı, tek bir gizli katman içerse de gizli katmanların sayısı birden çok olabilmektedir. Gizli katmandaki hücre sayısının arttırılması ise karmaşık yapıların tanımlanmasında ağın gücünü ve esnekliğini arttırmaktadır. Ancak gizli

katmanın çok fazla sayıda hücre içermesi de aşırı uyum (overfitting) sorununa neden olabilmektedir. Başka bir deyişle, veri kümesinin ezberlenmesi söz konusu olmakta ve model genellenebilirlikten uzaklaşmaktadır [18].

Geri Yayılım Algoritması, çok katmanlı ağları eğitmede en çok kullanılan temel bir öğrenme algoritmasıdır. Eğitme işlemi ve eğitimden sonraki test işlemi bu akışa göre gerçekleştirilir. Geri yayılım algoritması, danışmanlı öğrenme yöntemini kullanır. Örnekler ağa öğretilir ve ağa hedef değeri verilir. Her örnek için ağın çıktı değeri ile hedef değeri karşılaştırılır. Hata değeri, ağa tekrar geri besleme şeklinde verilir. Örnek setindeki hata kareleri toplamını azaltmak için nöronlar arasındaki bağlantı ağırlıkları değiştirilir [51].

Öğrenme algoritması olarak geri yayılım algoritması seçildiğinde öğrenme katsayısı önem kazanmaktadır. Öğrenme katsayısı, ağırlıkların bir sonraki düzeltmede hangi oranda değiştirileceğini göstermektedir. Küçük öğrenme katsayıları, ağın sonuca ulaşmasını yavaşlatır. Büyük öğrenme katsayıları, ağın sonuca daha kısa sürede ulaşmasını sağlar. Bununla birlikte çok yüksek oranlar ağın hesaplamalarında büyük salınımlara neden olur ve ağın dip noktayı bulmasını engelleyebilir. Öğrenme katsayısı için tipik değerler 0,01 ile 0,9 arasında değişir. Karmaşık ve zor çalışmalar için küçük öğrenme katsayıları seçilmesi önerilir [51].

Birçok yapay sinir ağı modeli, öngörü hatasını değerlendirmek üzere,

$$SS_E = \sum_i \sum_k (y_i - \acute{y}_{ik})^2 \tag{3.4}$$

şeklinde ifade edilen hata kareleri toplamını kullanmaktadır. Burada, kayıt sayısı i ve çıktı sayısı k ile gösterilmektedir.

Bağlantı ağırlıklarının en uygun değerinin belirlenmesinde hata kareleri toplamının en küçükleştirilmesi söz konusudur. Ancak sigmoid fonksiyonun doğrusal olmayan yapısı nedeniyle, en küçük kareler kestirimine karşılık gelecek bir fonksiyonel çözüm (closed-form solution) bulunmamaktadır. Bu yüzden, en uygun çözümün sağlanmasından çok, en uygun çözüme en yakın çözümün sağlanması söz konusu

olmaktadır. Bu amaçla kullanılan başlıca yöntem, eğim düşümü (gradient descent) yöntemidir [52].

Bu yöntemde amaç, bağlantı ağırlıklarının hata kareleri toplamını en küçükleştiren değerlerini belirlemektir. Sözgelimi, değeri belirlenmesi gereken m adet ağırlık bulunsun ve bunları bir vektör (w) ile ifade edelim. Eğim düşümü yöntemi, hata karelerinin en küçükleştirilmesi için bu vektörün her bir değerinin hangi yönde değişmesi gerektiğini belirleyecektir. Hata kareleri toplamının ağırlık vektörüne göre gradyantı,

$$\nabla SSE(w) = [\frac{\partial SSE}{\partial w_0}, \frac{\partial SSE}{\partial w_1}, \frac{\partial SSE}{\partial w_2}, \dots\dots\dots, \frac{\partial SSE}{\partial w_m}] \tag{3.5}$$

şeklinde ifade edilir. Bir başka deyişle, hata kareleri toplamının her bir ağırlığa göre kısmi türevlerinin bir vektörüdür [36]

Basit bir yaklaşımla, değeri belirlenecek tek bir bağlantı ağırlığı olduğunu varsayalım. Hata kareleri toplamının bağlantı ağırlığının değerine göre değişimini de bir parabol ile ifade edelim. Eğer söz konusu ağırlığın mevcut değeri en uygun değere göre negatif yönde ise parabolün o noktadaki eğimini gösteren türevi negatif değer alacaktır. Bu sonuç, en uygun değere yaklaşmak için ağırlığın mevcut değerinde pozitif yönde bir ayarlama gerektiği anlamına gelecektir. Aksi durumda ise türev değeri pozitif ve gereken ayarlama negatif yönde olacaktır. Sonuç olarak, ilgili ağırlığın mevcut değerinde gerçekleşmesi gereken ayarlamanın yönü, türevin işareti ile ters yönde olacaktır. Bu yüzden, ağırlığın mevcut değerindeki ayarlama,

$$\Delta w = -(\frac{\partial SSE}{\partial w}) \tag{3.6}$$

şeklinde hesaplanacaktır. Eğim, bağlantı ağırlığının en uygun değerden uzak olduğu noktalarda yüksek, yakın olduğu noktalarda ise düşük gerçekleşecektir. Ancak, burada hesaplanmış bulunan değişim ayarlamanın yönünü göstermekle birlikte, büyüklüğünü belirlemede yeterli değildir. Bu yüzden, öğrenme oranı (η) ile çarpılarak,

$$\Delta w = -\eta\left(\frac{\partial SSE}{\partial w}\right) \tag{3.7}$$

şeklinde ifade edilir. Burada, öğrenme oranı $0 < \eta < 1$ olarak tanımlanır. Öğrenme oranı, algoritmanın başlangıcında küçük bir değere eşit kabul edilirse, ağın yakınsaması kabul edilemez derecede uzun zaman alabilmektedir. Aksi halde ise ağın en uygun sonucu atlaması söz konusu olabilir. Bir çözüm olarak, algoritmanın başlangıcında öğrenme oranı değeri görece yüksek belirlenerek, çözüm yakınsadıkça daha düşük bir değere ayarlanması önerilmektedir [52].

Bu hesaplamaya dâhil edilen bir başka parametre ise momentum terimi (α) olarak bilinmektedir. Momentum teriminin eklenmesi ile geri yayılım algoritması güçlendirilmekte ve hesaplama,

$$\Delta w = -\eta\left(\frac{\partial SSE}{\partial w}\right) + \alpha\Delta w' \tag{3.8}$$

şeklinde olmaktadır. Burada, momentum teriminin değeri $0 \leq \alpha < 1$ ve bağlantı ağırlığının bir önceki ayarlaması $\Delta w'$ olmaktadır. Aslında momentum terimi, ataleti simgelemekte ve aldığı görece büyük değerlerle bağlantı ağırlığındaki mevcut değişimin bir önceki değişimle aynı yönde hareket etmesini sağlamaktadır. Ayrıca geri yayılım algoritmasında momentum terimi, önceki tüm ağırlık değişimlerinin üstel ortalamasının dikkate alınmasına neden olmaktadır. Öyle ki,

$$\Delta w_i = -\eta \sum_{k=0}^{\infty} \alpha^k \frac{\partial SSE}{\partial w_{i-k}} \tag{3.9}$$

ifadesinde yer alan α^k terimi mevcut ayarlamanın hesaplanmasında son dönem ayarlamalarının daha fazla dikkate alınmasını sağlar [36].

Geri yayılmış hata değeri bağlantının ağdaki konumuna bağlı olarak farklı ifadelerle hesaplanmaktadır. Çıkış hücrelerine olan bağlantılarda,

$$\delta_{pj} = (t_{pj} - o_{pj})o_{pj}(1 - o_{pj}) \tag{3.10}$$

ifadesi kullanılmaktadır. Burada, p'inci kayıt için j'inci çıkış hücresinin hedef değeri t_{pj} ile gösterilmiştir. Çıkış hücresine bağlı olmayan bağlantılarda ise,

$$\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{kj} \tag{3.11}$$

ifadesi kullanılmaktadır. Burada, j'inci hücrenin takipçisi konumundaki hücrelerin sayısı k ve k'ıncı hücrenin geri yayılmış hata değeri δ_{pk} şeklinde gösterilmiştir. Yapay sinir ağları, herhangi bir durma kriteri (stopping criterion) sağlanana kadar öğrenmeyi ve öngörülerini iyileştirmeyi sürdürür. Eğer durma kriteri bir zaman problemi ise algoritma kullanıcının tanımlayacağı parametrelere göre sonlandırılır. Alternatif olarak, hata kareleri toplamının bir alt eşik değere ayarlanması da durma kriteri olarak belirlenebilir. Çoğu yapay sinir ağı uygulamaları aşağıdaki çapraz geçerlilik prosedürünü izlemektedir [53].

- Veri kümesinin bir kısmının geçerlilik amacıyla ayrılması,
- Yapay sinir ağının öğrenme kümesi üzerinde yürütülmesi,
- Öğrenme kümesi üzerinde elde edilen bağlantı ağırlıklarının geçerlilik kümesine uygulanması,
- Öğrenme kümesi üzerinde elde edilen "son" ağırlıklarla, hata kareleri toplamını en küçükleştirmiş "en iyi" ağırlıkların geçerlilik kümesi üzerinde izlenmesi,
- "Son" ağırlıkların ürettiği hata kareleri toplamı "en iyi" ağırlıkların hata kareleri toplamından daha büyük olduğunda algoritmanın durdurulması.

Durma kriterinin kullanılması, en uygun sonucun elde edildiği anlamına gelmemektedir. Algoritma, hata kareleri toplamı için yerel en küçük değerde sonlanmış da olabilir. Bu durumda, sonuç en uygun olmasa da "iyi" olacaktır. Yapay sinir ağlarında öngörünün belirlenmesi, bağımlı değişkenin türüne göre farklılık göstermektedir. Sürekli bağımlı değişkenlerin öngörüsünde, modelin [0,1] aralığındaki öngörülerinin gerçek değer aralığına dönüştürülmesi gerekecektir. Çoklu kategorik bağımlı değişkenlerde öngörü, çıkış aktivasyonu en yüksek olan kategori

olmaktadır. İkili kategorik bağımlı değişken için ise çıktı değeri 0,5 etrafında değerlendirilmektedir. Eğer $o < 0{,}5$ ise öngörü 0.0, $o > 0{,}5$ ise öngörü 1.0 olarak belirlenmektedir [36].

Yapay sinir ağlarının en büyük dezavantajı, elde edilen modelin anlamlandırılması ve açıklanabilirliğinde yaşanan güçlük olarak gözükmektedir. Genellikle, süreçte kullanılan bağlantı ağırlıklarının ifade ettiği ilişkisel yapıyı anlamak mümkün değildir. Bu durum, sonuçların nasıl ortaya çıktığını kavramamızı imkânsız hale getirir. Bu yüzden, sonuçların ortaya çıkışındaki fonksiyonel formdan çok, sonuçların kendisinin önemli olduğu durumlarda uygulaması uygun bulunmaktadır. Yapay sinir ağları, bu yapısından dolayı kara kutu (black box) olarak da anılmaktadır. Ancak, birçok sinir hücresi barındırmaları nedeniyle, bozuk veri yapıları üzerinde dahi öğrenme kabiliyetine sahiptirler. Bu açıdan, veri madenciliğinin sınıflandırma fonksiyonu için kullanımı giderek yaygınlaşmıştır [36].

3.1.5. Radyal Tabanlı Fonksiyon

Yapay Sinir Ağlarının bir türü olan Radyal Tabanlı Fonksiyon Ağları (RTF) 80'lerin sonunda ortaya çıkmıştır. Bu ağların kökleri; potansiyel fonksiyonlar, kümeleme (clustering), fonksiyonel yaklaşım gibi eski örüntü tanıma yöntemlerine dayanmaktadır. RTF saklı her ünitesi Radyal fonksiyonları temsil eden iki katmanlı bir yapay sinir ağıdır. Ağın çıkışı saklı ünitelerin çıkışlarının toplamıdır. RTF'nun girişi non-lineer iken çıkışları lineerdir. Bu ağların lineer olmayan yaklaşım fonksiyonlarından dolayı, RTF'lar karmaşık haritalamalarda modelleme için kullanılmaktadır. Bu durumlarda perceptron ağları birçok katmanlar kullanarak ancak bu işlemi yapabilmektedir [53,54].

RTF giriş katmanı, ara katman ve çıkış katmanı olmak üzere üç katmandan oluşan ileri beslemeli bir ağ yapısıdır. Bu tip ağlarda ara katmandaki işlemci elemanı, girişlerin ağırlıklandırılmış şeklini kullanmaz [55]. Sadece ara katman işlemci elemanı çıkışları ağırlıklandırılarak çıkış katmanına iletilir. RTF'nin en önemli özelliği, ara katmanda transfer fonksiyonu olarak doğrusal olmayan radyal tabanlı bir fonksiyon kullanmasıdır. RTF ara katmanındaki işlemci elemanlarının çıkışı, YSA

girişleri ile temel fonksiyonun merkezi arasındaki uzaklığa göre belirlenmektedir [56].

Radyal tabanlı fonksiyon ağı tasarımı ise çok boyutlu uzayda eğri uydurma yaklaşımıdır ve bu nedenle RTF'in eğitimi, çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemine dönüşür. RTF'in genellemesi ise eğitim sırasında bulunan çok boyutlu yüzeyin kullanılmasına eşdeğerdir. Radyal tabanlı fonksiyonlar, sayısal analizde çok değişkenli problemlerin çözümünde kullanılmış ve YSA'nın gelişmesi ile birlikte bu fonksiyonlardan YSA tasarımında yararlanılmıştır. RBF, ileri beslemeli YSA yapılarına benzer şekilde giriş, saklı ve çıkış katmanından oluşur ancak, giriş katmanından saklı katmana dönüşüm, radyal tabanlı aktivasyon fonksiyonları ile doğrusal olmayan sabit bir dönüşümdür. Saklı katmandan çıkış katmanına ise doğrusal bir dönüşüm gerçekleştirilir.

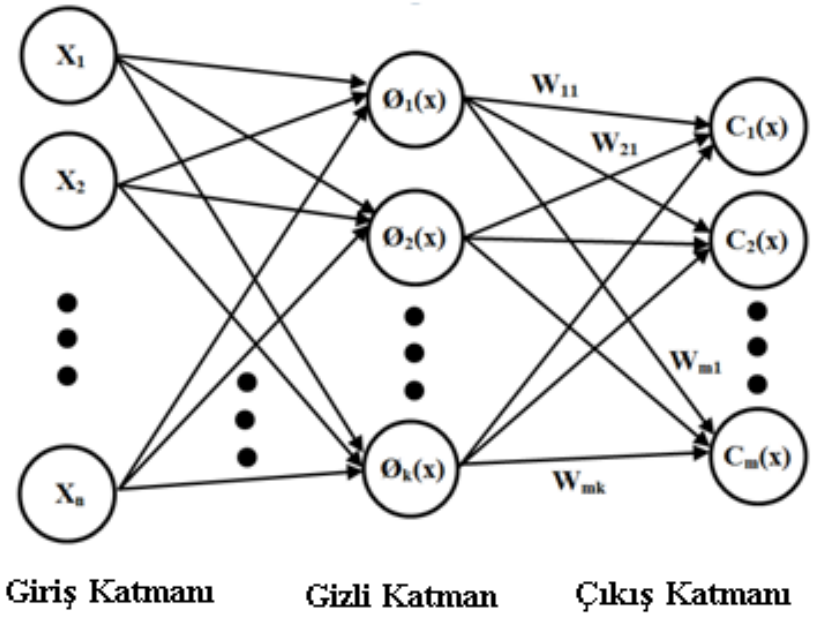
RTF'de uyarlanabilecek serbest parametreler; merkez vektörleri, radyal fonksiyonların genişliği ve çıkış katman ağırlıklarıdır. RTF'in matematiksel ifadesi eşitlik 3.12'deki gibidir [57]:

$$G(x) = \sum_{i=1}^{N} W_{ij}\varphi_i(x) \tag{3.12}$$

Burada W_{ij} gizli katmandaki i. nörondan çıkış katmanındaki j. nöron arasındaki ağırlığı, $\varphi_i(x)$ ise aktivasyon fonksiyonunu gösterir. RTF'de gizli katman aktivasyon fonksiyonu olarak genellikle Gauss Fonksiyonu kullanılır. Gauss Fonksiyonu *x* giriş vektörünü, c_i merkezi, $\|x - c_i\|$ standart Öklid uzaklığını, σ_i de genişliği göstermek üzere eşitlik 3.13'teki gibi ifade edilir.

$$\varphi(r) = \exp(-\frac{\|x-c_i\|^2}{2{\sigma_i}^2}) \tag{3.13}$$

En genel yapısıyla RTF yapısı Şekil 3.12'deki gibidir.



Şekil 3.12. Radyal tabanlı fonksiyon sinir ağı.

3.1.6. RTF Ağlar ve Çok katmanlı Algılayıcıların Karşılaştırılması

RTF Ağlar ve çok katmanlı algılayıcılar doğrusal olmayan katmanlı ileri beslemeli ağların örnekleridir. İkisi de evrensel yakınsayıcılardır. Belirli bir çok katmanlı ağı taklit eden bir RTF ağın veya tam tersinin mevcut olması şaşırtıcı değildir. Ancak bu iki ağ pek çok bakımdan birbirlerinden ayrılırlar:

- Bir RTFA'nın (en çok temel formunda) tek bir saklı katmanı vardır, oysaki çok katmanlı algılayıcı bir veya daha fazla saklı katmana sahip olabilir.

- RTFA'nın saklı katmanı doğrusal değildir, çıkış katmanı ise doğrusaldır. Öte yandan, sınıflayıcı olarak kullanılan çok katmanlı algılayıcının saklı ve çıkış katmanları genellikle doğrusal değildir.

- RTFA'daki her saklı katman birimi aktivasyon fonksiyonunun değişkeni giriş vektörü ve o birimin merkezi arasındaki mesafeyi hesaplar. Çok katmanlı

algılayıcının her saklı birimi giriş vektörü ve o birimin sinaptik ağırlık vektörünün iç çarpımını hesaplar.

- Çok katmanlı algılayıcılar doğrusal olmayan giriş-çıkış haritalamasına global yaklaşmalar oluştururlar. Bu nedenle, küçük veya eğitme verisi olmadığında giriş uzayının bölgelerinde genelleştirme yapabilirler. Öte yandan, üstel olarak azalan lokalleşmiş doğrusal olmayan özellikleri (Gauss) kullanan RTF ağlar, doğrusal olmayan giriş-çıkış haritalamasına yerel yakınsama oluştururlar, bunun sonucu olarak bu ağlar hızlı öğrenme yeteneğine ve eğitme verisinin sunum sırasına azaltılmış hassasiyet oluştururlar [58].

3.2. DESTEK VEKTÖR MAKİNELERİ

Destek vektör makineleri (DVM) son yıllarda özellikle veri madenciliğinde değişkenler arasındaki örüntülerin bilinmediği veri setlerindeki sınıflama problemleri için önerilmiş bir makine öğrenmesidir. Yönteme ilişkin ilk yayın Vapnik tarafından 1964-1965 yıllarında hazırlanmış olmasına karşın, DVM'ler 90'lı yıllara kadar dikkate alınmamıştır. Bunun nedeni ise kullanıcılar pratik uygulamalarda DVM'lerin uygun ve makul olmadığını düşünmekteydiler [59]. Vladimir Vapnik, Bernhard Booser ve Isabella Guyon tarafından 1992 yılında yapılan yayının ardından DVM'ler ilgi odağı haline gelmiştir. Özellikle yüz tanıma, parmak izi tanıma gibi birçok alanda sınıflama amacıyla kullanıldığında çok iyi sonuçlar verdiği fark edildiğinde birçok kullanıcı sınıflama amacıyla DVM'leri tercih etmeye başlamıştır.

3.2.1. Destek Vektör Makinelerinin Tarihçesi

Sınıflama işlemi esas olarak benzer nesneleri sınıf adı verilen, önceden belirlenmiş kategorilere ayırma işlemidir. Tez çalışmasının temelini oluşturan Destek Vektör Makineleri (DVM) de veri madenciliğinde sınıflama yapmak amacıyla kullanılan bir makine öğrenmesi yöntemidir. DVM yaklaşık olarak 50 yıla yakın bir gelişim sürecine sahiptir. Hatta 1936'lara kadar gidilerek Fisher'in Doğrusal Diskriminant yöntemine kadar dayandırılabilinir. Çünkü bugün iki ya da çok sınıflı sınıflama problemlerinin çözümü için kullanılan istatistik, makine öğrenmesi ve örüntü

tanımlama yöntemleri bu temele dayanmaktadır. Fakat gelişme sürecinin asıl başlangıç noktasının 1957'de Frank Rosenblatt tarafından önerilen ilk doğrusal sınıflayıcı makine öğrenmesi modeli "*perceptron*" olduğu söylenebilir. *Perceptron*'un ardından özellikle Vapnik ve diğ. (Vapnik,1982-1995) makine öğrenmesi yöntemleri üzerine detaylı olarak çalışmışlardır. 1963 yılında Vapnik ve Lerner genelleştirilmiş düşey algoritmayı (generalized portrait algorithm) geliştirmişler ve aynı dönemde Vapnik ve Chervonenkis en iyi çoklu düzlem tekniği ile örüntü tanıma algoritmasını sunmuşlardır. Bu çalışmalar DVM'lerin temel yapısının ortaya ilk olarak atıldığı çalışmalardır. En iyi çoklu düzlem kullanılarak örüntü tanımlama yapan bu yöntemin tutarlılığı üzerine çalışarak VC (Vapnik-Chervonenkis) Teorisi olarak adlandırılan teori öne sürülmüştür. VC teorisi öğrenebilirlik için gerekli ve yeterli şartların neler olduğunu ortaya koymaktadır ve ancak 1982 yılında Vapnik tarafından yayınlanan "*Estimation of Dependences Based on Empirical Data"* adlı kitapta özetlenmiştir. En iyi çoklu düzlem görüşünün ardından Cover (1965) en büyük marjinli çoklu düzlem görüşünü ileri sürmüş ve Mangasarian (1965), Duda ve Hart (1973) bu teknikleri örüntü tanıma problemleri için kullanmıştır.

Makine öğrenmesine başlangıçta karşı çıkan istatistikçiler, daha sonra yüksek boyutlu verilerle karşılaştıklarında içinden çıkamadıkları problemlere çözüm getiren bu öğrenme algoritmalarını ve daha sonra Vapnik ve Chervonenkis (1974) tarafından öne sürülen "istatistiksel öğrenme" yöntemlerini benimsemişlerdir. Aslen Rus olan bu iki matematikçinin Rusça yazdığı eserler önce Almanca ardından İngilizceye çevrilince yöntem daha geniş bir alanda ismini duyurmuştur. Busor, Guyon ve Vapnik tarafından 1992 yılında *kernel trick* olarak adlandırılan çekirdek düzenlemesi ile 1995'de Cortes tarafından öne sürülen ihmal edilebilir hataları içeren yumuşak marjin (*soft margin)* yaklaşımı DVM'yi bugün ki formuna taşımıştır.

3.2.2. Destek Vektör Makinelerinde Sınıflandırma

DVM'ler esas olarak iki sınıflı sınıflama problemlerinin çözümü için önerilmiştir. Verilerin iki sınıfa doğrusal olarak ayrılabildiği ve az sayıda noktadan dolayı doğrusal olarak ayrılamadığı durumlar için doğrusal DVM olduğu gibi, daha sonra

doğrusal olarak ayrılması mümkün olmayan veriler için doğrusal olmayan DVM geliştirilmiştir.

Sınıflama için geliştirilmiş bir makine öğrenmesi, eğitim seti için sınıflama performansını maksimum yapmaya çalışmaktadır. Daha sonra test verisine uyguladığında da sınıflama doğruluğunun yüksek olması istenmektedir. Ancak eğer veriden öğrenme sırasında, sınıflayıcı eğitim setine çok uyarsa aşırı uyum sorunu (overfitting) ortaya çıkar. Bu durumda sınıflayıcının bilinmeyen veriye genelleme yeteneği azalır ve bu veri seti için sınıflama performansı düşük olur. Yani genelleme becerisi ile eğitim setine uyum arasında ödünleşim (trade-off) vardır [60]. İki sınıflı sınıflama problemleri için doğrusal olmayan DVM doğrudan karar fonksiyonunun genelleme performansını maksimum yapmak için deneme yapar. n boyutlu girdi uzayı daha yüksek boyutlu bir olay uzayına haritalanır. Daha sonra bu olay uzayında en iyi ayırıcı çoklu düzlem tarafından veriyi iki sınıfa ayıracak kuadratik programlama problemi çözülür.

DVM'lerin sınıflandırma mekanizması, üç ayrı veri durumu için detaylandırılabilir;

- Doğrusal olarak ayrılabilir veriler
- Doğrusal olarak ayrılamaz veriler
- Ayırımı doğrusal olmayan veriler

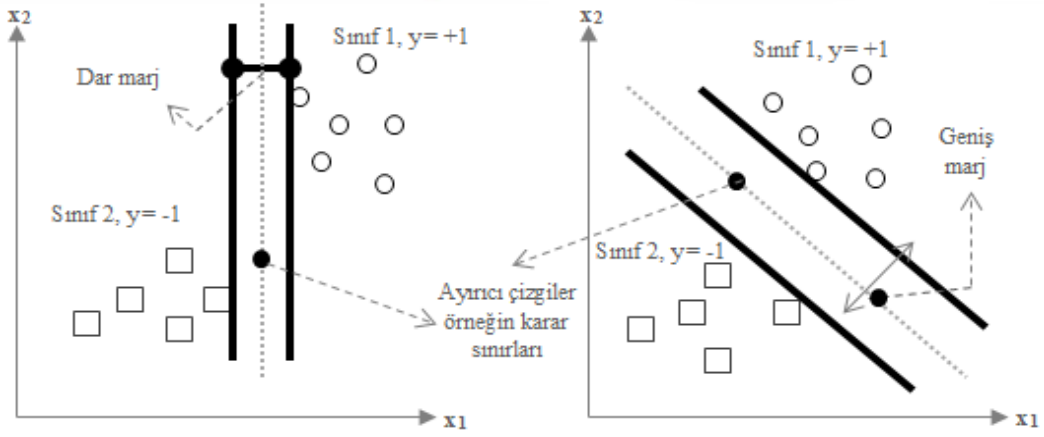
3.2.3. Doğrusal Olarak Ayrılabilir Veriler

Destek vektör makinesinin en basit ve ilk olarak tanıtılan modeli doğrusal sınıflandırıcıdır. Sadece uygun şekilde seçilmiş bir kernel fonksiyonuna bağlı özellik uzayında doğrusal ayrılabilir veriler için çalışmaktadır ve bu nedenle gerçek dünya probleminde kullanılamamaktadır. Destek vektör makinesinin temel yapı taşını oluşturmasının yanı sıra, bu çeşit öğrenen makineyi karakterize eden anahtar özellikleri sergilemektedir. Tanımı çok daha geniş sistemleri anlamak için önemlidir [61].

Eğitim verileri kümesinin iki ayrı sınıftan verildiği düşünülürse; $\{x_i, y_i\},\ i = 1, \dots, l$, $y_i \in \{-1, +1\}$, $x_i \in R^d$ olarak etiketlendirilmiş olsun.Özellik vektörlerinin ayrılabilir

olduğu ve doğrusal bir karar sınırı tarafından ayrılabildiği varsayılmaktadır. Girdi uzayı herhangi bir boyut olabilirken, karar sınırı bir hiperdüzlemdir. Tanımlanan veri kümesi için, iki veri sınıfını ayırabilen, kanonik hiperdüzlemler olarak adlandırılan, hiperdüzlemler kümesi bulunmaktadır [62]. Doğrusal ayrılabilir veriler için, veri kümesini verilen etiketlere göre bir hiper düzlemle ayırıp, aynı sınıfa ait bütün veri noktalarını hiper düzlemin aynı tarafında bırakmak mümkündür [63].

Örneğin iki boyutlu girdi uzayı vakası, $x \in \mathrm{R}^2$ ele alındığında veriler doğrusal ayrılabilirdir ve ayrımı gerçekleştirebilen pek çok hiperdüzlem bulunmaktadır. Amaç doğru genelleyen bir sınıflandırıcı geliştirmektir [64].



Şekil 3.13. Ayırıcı doğrular.

Şekil 3.13'de sağda geniş marjlı iyi bir ayırıcı, solda dar marjlı daha az kabul edilebilir bir ayırıcı gösterilmektedir [65].

En iyi ayırıcı, hiperdüzlem payı enbüyüklenerek verilir. Çünkü her iki sınıf verilerinden de mümkün olduğunca uzakta olan karar sınırı, en iyi ayırıcıdır. Büyük pay, kestirimin eğitim setinde güvenilir olmasını ve görünmeyen örnekler üzerinde kestirimin başarımının iyi olmasını sağlar. Karar sınırı problemi aşağıdaki kısıtlamalı eniyileme problemi ile ifade edilir.

$$\text{Minimum} \quad \frac{1}{2}\|w\|^2 \tag{3.14}$$

$$\text{Kısıtlama} \quad y_i(w^T x_i + b) \geq 1 \ \ \forall i \tag{3.15}$$

Bu kısıtlamalı eniyileme problemi (3.12), (3.13), Lagrange çarpanları ($\alpha_i > 0$) ve bir Lagrangian ele alınır;

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i (y_i (w^T x_i + b) - 1) \qquad \alpha_i \geq 0 \quad \forall i \tag{3.16}$$

Bu Lagrangian, w ve b değişkenlerine göre enküçüklenir, α çarpanlarına göre enbüyüklenir. Problem;

$$\max_\alpha \min_{w,b} L(w, b, \alpha) \tag{3.17}$$

olarak yazılır. Eniyileme problemleri ikincil biçimlerine dönüştürülebilirler. Bu Lagrangian'ın özgün değişkenlerine göre kısmi türevleri alınıp çözülmesi ve sonuçların Lagrangian'da yerlerine konarak elenmesi şeklinde yapılır. Sonuç sadece Lagrange çarpanlarında enbüyüklenecek bir bağıntıdır. Özgün değişkenlerdeki eşitsizlik kısıtlamaları da artık çarpanlardaki eşitlik kısıtlamalarına dönüşmüştür. İkincil eniyileme problemi,

$$\max_\alpha W(\alpha), \qquad \alpha \geq 0 \tag{3.18}$$

biçiminde olacaktır. Birincil Lagrangian'ın w ve b'ye göre kısmi türevlerinden

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i \; y_i \; x_i$$
$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^n \alpha_i \; y_i \tag{3.19}$$

(3.16), (3.17), (3.18), (3.19) dan yararlanılarak ikincil problem;

$$\max_\alpha W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.20}$$

ve kısıtlamaları

$$\begin{cases} \alpha_i \geq 0, & i = 1 \dots n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{3.21}$$

olarak yazılır. Bu bir karesel programlama (QP) problemidir ve α_i'nin bir evrensel enbüyüğü her zaman bulunabilir. Problem, gradient ve Newton gibi yöntemlerle çözülebilir. İkincil Lagrangian ile; daha basit kısıtlamalar gelir ve problem, semer noktasından basit bir enbüyüklemeye çevrilir. Fakat ikincil Lagrangian kullanmanın esas sebebi, problemi "Kernel Hilesi"nin kullanılmasına izin veren bir biçime çevirmesidir.

(3.19) bağıntısının ilk terimi, çözüm vektörünün, eğitim örüntülerinin (pattern) bir alt kümesinin terimlerinde açılıma sahip olduğunu göstermektedir. Başka bir deyişle bu örüntüler, Lagrange çarpanları sıfır olmayan aktif kümelerle bulunurlar. Bu, dikkat edilmesi gereken bir noktadır, α_i'lerden pek çoğu sıfırdır. Optimal ayırıcı, aktif kısıtlamalar yani sıfırdan farklı α_i'lerle bulunur. W, az sayıdaki veri noktalarının doğrusal bir birleşimidir. α_i'nin sıfır olmadığı x_i'lere "destek vektörleri" denir. Ve karar sınırı sadece destek vektörleri ile belirlenir. Eğer veri, doğrusal olarak ayrılabiliyor ise tüm destek vektörleri, Karush-Kuhn-Tucker (KKT) tanımlayıcı koşulundan,

$$\alpha_i(y_i(w^T x_i + b) - 1) = 0, \qquad i = 1, \dots, n \tag{3.22}$$

bağıntısını sağlayan yardımcı hiperdüzlemler üzerinde bulunurlar. Bu nedenle destek vektör sayısı çok küçük olabilir. Sonuçta, en iyi hiperdüzlem eğitim kümesinin bir alt kümesi tarafından belirlenir, diğer noktalar eğitim kümesinden atılabilir. Bu matematiksel olarak ifade edilirse $t_j (j = 1, \dots, s)$ ler s tane destek vektörünün indisleri olarak alındığında eşitlik 3.23'deki gibi olur.

$$w = \sum_{j=1}^{s} \alpha_{t_j} \, y_{t_j} \, x_{t_j} \tag{3.23}$$

Yeni bir z datasını test etmek için

$$w^T z + b = \sum_{j=1}^{s} \alpha_{t_j}\ y_{t_j}\ (x_{t_j}^T z) + b \tag{3.24}$$

hesaplanır ve z, bu toplam pozitif ise sınıf 1'e, diğer hallerde sınıf 2'ye aittir denir, yani karar fonksiyonu aşağıdaki gibi yazılabilir;

$$f(z) = sign(w^T z + b) = sign(\sum_{j=1}^{s} \alpha_{t_j}\ y_{t_j}\ (x_{t_j}^T z) + b) \tag{3.25}$$

b parametresi de, (3.20) bağıntısından çekilerek hesaplanabilir [66].

3.2.4. Doğrusal Olarak Ayrılamayan Veriler

Verilerin doğrusal bir düzlemle ayrılamama durumunda negatif olmayan ve hataları ifade eden ξ_i gevşek değişkenlerinin optimizasyon modeline eklenmesi sağlanarak soruna çözüm aranır.
Gevşek değişkenler yardımıyla, $y_i(w^T x + b) \geq 1,\ \forall i$ yerine,

$$y_i(w^T x_i + b)\ \geq 1 - \xi_i\ \ (i = 1,2,\dots,n), \xi_i \geq 0 \tag{3.26}$$

veya

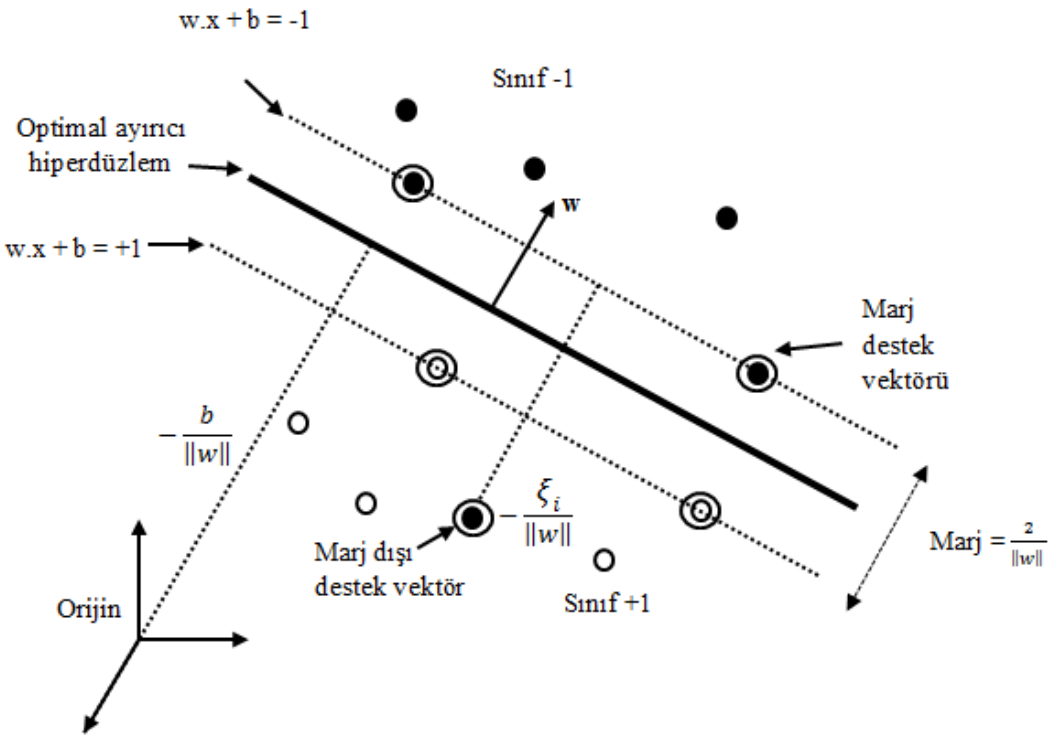
$$(w^T x_i + b)\ \geq 1 - \xi_i\ ,\quad y_i = +1 \text{ için}$$
$$(w^T x_i + b)\ \leq -1 - \xi_i\ ,\quad y_i = -1 \text{ için} \tag{3.27}$$

yazılabilir. Burada $\xi_i > 1$ olan veriler hiperdüzlemin diğer tarafında kalan, yani doğrusal olarak ayrılmayı önleyen bölgedeki gözlem değerleridir. $0 < \xi_i < 1$ ise, hiperdüzlemin doğru yanında yer alan, ancak en büyük alan margin bölgesi içinde kalan gözlem değerlerini ifade eder.

Bu tür bir genelleştirilmiş optimal hiper düzlem için maksimize edilecek fonksiyon, doğrusal ayırmayı engelleyen bu tür durumlar için bir ilave terime sahip olacaktır. C ceza parametresi olmak üzere amaç fonksiyonu eşitlik 3.26'daki gibi ifade edilir:

$$L(w,\xi) = \frac{1}{2}w^T w + C(\sum_{i=1}^{n} \xi_i)^k \tag{3.28}$$

Burada C>0 bir sabittir ve kullanıcı tarafından seçilir. Eğer C küçük ise ideal pozisyonda olmayan birçok gözleme izin verilir. Aksi taktirde, ideal pozisyonda olmayan çok az sayıda gözleme sahip olunmak istenir. Formülde k =1 seçildiğinde konveks programlama problemi haline dönüşür.



Şekil 3.14. Doğrusal ayrılabilir olmayan durumda optimal ayırıcı hiperdüzlem.

Bir doğrusal ayırma problemi için (3.28) quadratik programlama probleminin 3.26 kısıtları altında çözülmesidir. $L(w,b,\xi,\alpha,\beta)$ primal Lagrange fonksiyonu k =1 için eşitlik 3.29'deki gibidir.

$$L(w,b,\xi,\alpha,\beta) =$$
$$\frac{1}{2}w^T w + C(\sum_{i=1}^{n} \xi_i) - \sum_{i=1}^{n} \alpha_i \{ y_i[w^T x_i + b] - 1 + \xi_i \} - \sum_{i=1}^{n} \beta_i \xi_i \tag{3.29}$$

Burada α_i ve β_i Lagrange çarpanlarıdır. Bu problem primal ya da dual olarak çözülebilir.

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i \; y_i \; x_i$$
$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^{n} \alpha_i \; y_i$$
$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow \quad \alpha_i + \beta_i = 0 \tag{3.30}$$

Bunun dışında KKT koşulu ilave edilebilir:

$$\alpha_i \{ y_i [w^T x_i + b] - 1 + \xi_i \} = 0, \;\; i = 1,2, \dots, n \tag{3.31}$$

Dual Lagrange fonksiyonu eşitlik 3.32'deki gibidir.

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \tag{3.32}$$

Optimal hiperdüzlemi bulmak için $L(\alpha)$ dual Lagrange fonksiyonu pozitif α_i ler için; yani C $\geq \alpha_i \geq 0$ i= 1,2,…,n koşulları altında maksimize edilmelidir. Bu durumda, doğrusal ayrılabilen durum için elde edilen quadratik programlama problemi ile aynı sonuç elde edilmiştir. Buradaki tek fark, α_i Lagrange çarpanları için bir C üst sınır getirilmesidir. Eğer C = ∞ olarak kabul edilirse, verilerin tümüyle doğrusal olarak ayrılabildiği durum elde edilir. Sonuç olarak, verilerin doğrusal olarak ayrılamadığı durum için quadratik programlama modeli eşitlik 3.33'deki gibidir.

Maksimum: $L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$
Kısıtlama: $\sum_{i=1}^{n} \alpha_i y_i = 0, \;\; i = 1,2, \dots, n$
C $\geq \; \alpha_i \; \geq 0, \;\; i = 1,2, \dots, n$ (3.33)

Yukarıdaki bağıntılar matrislerle eşitlik 3.34'deki gibi ifade edilir [67]:

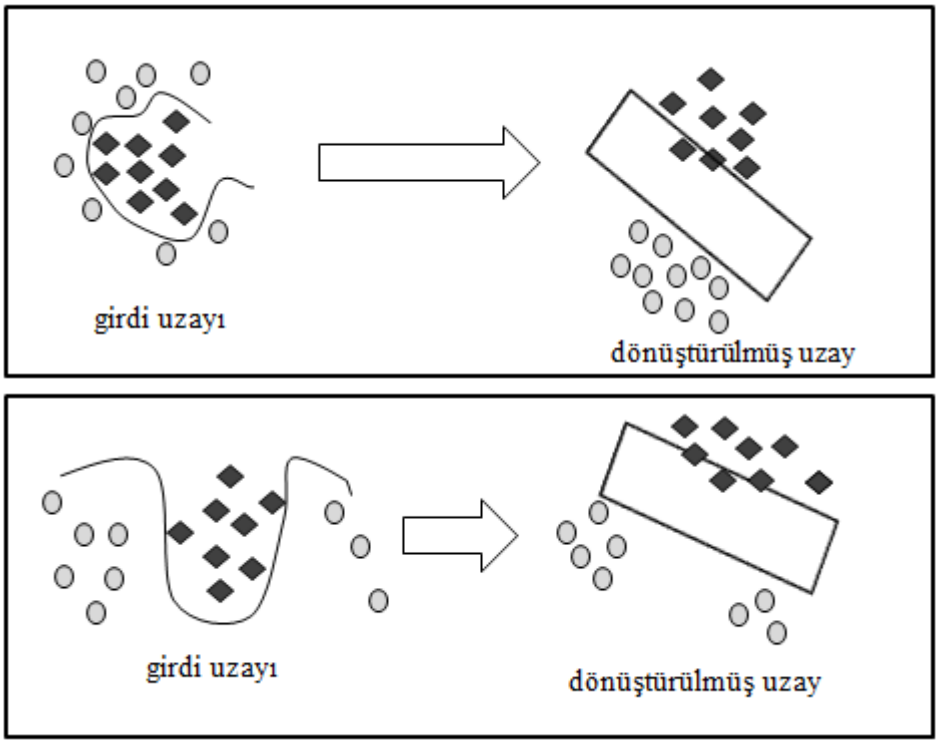
Maksimum: $\alpha^T I - \frac{1}{2}\alpha^T H\alpha$

Kısıtlama: $y^T\alpha = 0$

C $\geq \alpha \geq 0$ (3.34)

3.2.5. Doğrusal Olmayan Veriler

Uygulamada her zaman yukarıda anlatıldığı gibi verilerin doğrusal olarak ayrılabildiği durumlarla karşılaşılmamaktadır. Şekil 3.15'de olduğu gibi iki sınıf iç içe geçmiş gibi ya da veri grupları arasında kalmış gibi bir yapı gösterebilir.



Şekil 3.15. Doğrusal ayrılamayan verilerin farklı boyutlardaki uzaylara aktarılması.

DVM'ler böyle durumlarla karşılaştığında doğrusal olmayan haritalama (mapping) yaparak, verileri n boyutlu orjinal girdi uzayından daha yüksek boyuta sahip olay (feature) uzayına taşır [68].

$x \in R^n \rightarrow \Phi(x) \in R^f$ (3.35)

Doğrusal olmayan DVM, verilerin taşındığı bu yeni boyutta doğrusal DVM gibi çalışarak verileri ayıracak optimum çoklu düzlem arar.

Dönüştürme işlemi için kullanılacak olan fonksiyon $\Phi(x)$ olarak belirlensin. Bu durumda doğrusal DVM'den tek farkı x yerine $\Phi(x)$ kullanılması olacaktır. Buradan hareketle dönüştürülmüş uzayda kullanılacak karar fonksiyonu:

$$< w, \Phi(x) > + b = 0 \tag{3.36}$$

şeklinde olacaktır. Destek vektörlerinin üzerinde yer aldığı ve ayırıcı çoklu düzleme paralel doğruların ayırdığı veriler aşağıdaki şekilde sınıflanır:

$$< w, \Phi(x) > + b \geq +1 \tag{3.37}$$
$$< w, \Phi(x) > + b \leq -1 \tag{3.38}$$

Aynı şekilde nesne fonksiyonu ve buna ilişkin formül (3.37) ile (3.38)'in birleşiminden oluşan kısıt aşağıdaki gibidir:

$$\min_{w,b} \tau(w) = \frac{1}{2}\|w\|^2 \tag{3.39}$$
$$y_i(< \Phi(x).w > + b) - 1 \geq 0, \qquad \forall i \text{ için} \tag{3.40}$$

Burada iki sorun ortaya çıkmaktadır. İlk olarak dönüştürülmüş uzayda oluşturulacak doğrusal karar sınırı ile ilgili nasıl bir haritalama fonksiyonu kullanılacağı açık değildir. İkinci sorun ise uygulanan haritalama fonksiyonu biliniyorsa, kurulan optimizasyon probleminin yüksek boyutlu olay uzayında çözümü zor ve karmaşık hesaplamalar gerektirecektir.

w ve b parametrelerini hesaplamak için:

$$w = \sum \alpha_i y_i \Phi(x_i) \tag{3.41}$$
$$f(x) = (\sum \alpha_i y_i \Phi(x_i).\Phi(x)) + b = 0 \tag{3.42}$$

yukarıdaki denklemler dönüştürülmüş uzaydaki iki vektörün iç çarpımını içermektedir. Boyut sorunundan (curse of dimensionality) dolayı bu iç çarpımların hesaplanması zordur. Bu sorunu önlemek amacıyla çekirdek düzenlemesi olarak adlandırdığımız "Kernel Trick" yöntemi önerilmiştir.

3.2.6. Çekirdek Düzenlemesi ve Çekirdek Fonksiyonları

Çekirdek düzenlemesi yapılarak dönüştürülmüş uzaydaki Φ(x) vektörü yerine girdi uzayındaki verilerden oluşan bir çekirdek fonksiyon oluşturularak işlemler gerçekleştirilir.

Çekirdek düzenlemesinin iki önemli faydası bulunmaktadır:

- Direk girdi uzayındaki veriler kullanılacağı için Φ haritalama fonksiyonun kesin olarak ne olduğunu bilmeye gerek yoktur.

- Çekirdek fonksiyon kullanarak iç çarpım hesaplamak, dönüştürülmüş nitelik seti Φ(x) kullanarak hesaplamaya kıyasla daha kolaydır ve maliyeti düşüktür.

İç çarpımlar genellikle iki girdi vektörü arasındaki benzerliğin bir ölçüsü olduğu için, çekirdek düzenlemesi, orijinal veriyi kullanarak dönüştürülmüş uzayda bir benzerlik hesaplaması yapar [1]. Dönüştürülmüş uzaydaki iki girdi vektörü u ve v için iç çarpımlar:

$$\Phi(\mathrm{u})\Phi(\mathrm{v}) = \left(\mathrm{u}_1^2, \mathrm{u}_2^2, \sqrt{2}\mathrm{u}_1, \sqrt{2}\mathrm{u}_2\,, 1\right)\left(\mathrm{v}_1^2, \mathrm{v}_2^2, \sqrt{2}\mathrm{v}_1, \sqrt{2}\mathrm{v}_2\,, 1\right) = (\mathrm{uv}+1)^2 \qquad (3.43)$$

Dönüştürülmüş uzaydaki iç çarpımlar orijinal nitelik verisinden hesaplandığı için bu benzerlik fonksiyonu K ile gösterilen "çekirdek fonksiyon" olarak adlandırılır (3.44).

$$K(u, v) = \Phi(\mathrm{u})\Phi(\mathrm{v}) = (\mathrm{uv}+1)^2 \qquad (3.44)$$

Doğrusal olmayan DVM'de kullanılan çekirdek fonksiyonları Mercer Teoremi (Vapnik,1995) olarak bilinen matematiksel bir kurala uymak zorundadırlar [69]. Bu

kural yüksek boyutta çalışılırken çekirdek fonksiyonların her zaman iki girdi vektörünün iç çarpımı şeklinde ifade edilmesini sağlamaktadır.

Bu tez çalışmasında kullanımı yaygın olan dört çekirdek fonksiyona yer verilmiştir. Bu fonksiyonlar:

- Doğrusal Fonksiyon
- Polinomial Fonksiyon
- Sigmoid Fonksiyon
- Radyal Tabanlı Fonksiyon

Doğrusal Fonksiyon: Girdi uzayında veriler doğrusal olarak ayrılabiliyor ise veriyi yüksek boyuta taşımaksızın doğrusal çekirdek fonksiyonu yardımıyla sınıflama işlemi yapılır. Bu fonksiyon herhangi bir boyut değeri ya da katsayı içermemektedir.

$$K(x_i, x_j) = x_i^T x_j \tag{3.45}$$

Polinomial Fonksiyon: Polinomial çekirdek fonksiyon, d gibi belirli bir derecede girdi vektörlerinin iç çarpımından oluşmaktadır. Fonksiyonun matematiksel gösterimi eşitlik 3.46'daki gibidir.

$$K(x_i, x_j) = (x_i, x_j)^d \tag{3.46}$$

d = 1 olduğu durumlarda polinomial fonksiyon doğrusal fonksiyona dönüşmektedir.

Sigmoid Fonksiyon: Sigmoid fonksiyon k ve δ gibi iki parametre içermektedir. Kaynaklar belirli parametreler için sigmoid fonksiyonun radyal tabanlı fonksiyon şeklinde çalıştığını göstermektedir [68].

$$K(x_i, x_j) = tanh(kx_i, x_j - \delta) \tag{3.47}$$

Radyal Tabanlı Fonksiyon: Radyal tabanlı fonksiyon çekirdek fonksiyonlar arasında kullanımı en yaygın çekirdek fonksiyondur. R programında sistem standart çıktılarını

radyal tabanlı fonksiyona göre vermektedir. γ yarıçap kontrolünü sağlayan parametredir [60].

$$K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \quad \gamma > 0 \tag{3.48}$$

DVM, temelde iki sınıflı problemlerin çözümü için kullanılan bir yöntem olmasına rağmen, son yıllarda geliştirilen yaklaşımlarla ikiden fazla sınıfa ait veri kümesini sınıflandırabilen ve çok sınıflı DVM olarak isimlendirilen yöntemler önerilmiştir [70,71]. Bu yaklaşımlardan en önemlileri, Bire Karşı-Diğerleri (BKD) ve Bire Karşı-Bir (BKB) yöntemleridir. Temelde bu yaklaşımlar da, sınıf sayısına bağlı olarak ikili sınıflandırmalar yaparak çok sınıflı problemleri çözmektedir.

Çok sınıflı DVM sınıflandırıcılar için BKD prensibi: Vapnik tarafından 1998 de önerilen bir yöntemdir. Bu metottaki ana fikir, BKD yönteminde her örnek kümesi, geri kalan tüm örnekler bir kümeye aitmiş gibi kabul edilerek eğitilir [72]. Yani k farklı sınıf olması durumunda, k tane eğitim işlemi yapılır. Veri kümesinin büyük olduğu durumlarda her sınıfın ayrımında ayrıca eğitim yapıldığından fazla eğitim zamanı tüketebilir.

Çok sınıflı DVM sınıflandırıcılar için BKB prensibi: Bu teknikte k farklı sınıf etiketi bulunan sınıflandırma uygulaması için *k*(k-1)/2* sayıda destek vektör makinesinin eğitilmesi gerekir [72]. Her eğitim aşamasında sadece iki farklı sınıf verisinin alınması yeterlidir. Sınıf etiketlerinin sayısına bağlı olarak kullanılacak destek vektör makinesi sayısı değişir. Örneğin 3 sınıflı bir sınıflandırma problemi için birinci sınıflandırıcı, bir ve iki etiketli sınıfları birbirinden ayırır. İkinci sınıflandırıcı bir ve üç etiketli sınıfları birbirinden ayırır. Üçüncü sınıflandırıcı ise, iki ve üç etiketli sınıfları birbirinden ayırır.

3.3. KARAR AĞAÇLARI

Karar ağaçları (KA) son yıllarda literatürde yaygın kullanımı olan bir sınıflandırma ve örüntü tanımlama algoritmasıdır. Bu yöntemin yaygın olarak kullanımının en

önemli nedeni ağaç yapılarının oluşturulmasında kullanılan kuralların anlaşılabilir ve sade olmasıdır. KA sınıflandırma işleminin gerçekleştirilmesinde çok aşamalı veya ardışık bir yaklaşım kullanmaktadır [73].

Tahmin edici ve tanımlayıcı özelliklere sahip olan karar ağaçları, veri madenciliğinde hem kuruluşlarının hem de yorumlanmalarının kolay olması, veri tabanı sistemlerine kolayca entegre edilebilmeleri, güvenilirliklerinin daha iyi olması nedenleri ile sınıflama modelleri içerisinde en yaygın kullanıma sahip olan bir tekniktir [74]. Büyük veri tabanlarının kullanıldığı pek çok sınıflama probleminde ve karmaşık ya da hata içeren bilgilerde karar ağaçları yararlı bir çözüm olmaktadır [75]. Morgan ve Sonquist (1963) Regresyon ağaçlarını geliştirmişlerdir [76]. Karar ağaçları öğrenmesi, bir kesikli hedef fonksiyonu yaklaştırması yöntemidir. Burada sözü edilen yöntem karar ağacı ile gösterilen bir öğrenme fonksiyonunu ortaya koyar. Öğrenme sürecini tamamlamış ağaçlar konuşma diline yakın "if-then" kuralları biçiminde de ifade edilebilir. Tümevarımcı çıkarım yöntemleri arasında yer alan bu öğrenme yöntemleri tıptan kredi kartı risk analizlerine kadar pek çok uygulamada başarıyla uygulanmıştır [77].

3.3.1. Karar Ağaçlarının Elde Edilmesi Süreci

Karar ağaçları, sınıfları bilinen örnek veriden tümevarım yöntemiyle öğrenilen ağaç şekilli bir karar yapısı çeşididir [78]. Bir karar ağacı, basit karar verme adımları uygulanarak, büyük miktarlardaki kayıtları, çok küçük kayıt gruplarına bölerek kullanılan bir yapıdır. Her başarılı bölme işlemiyle, sonuç gruplarının üyeleri bir diğeriyle çok daha benzer hale gelmektedir [79]. Karar ağaçları Bierman ve Friedman tarafından 1973 yılında önerilmiş olup değişkenleri parçalayarak bir ağaç oluşturmaya dayanmaktadır [80]. Karar ağacında, tanımlanan sorunun cevabı gruplara ayrılmaktadır. Soruya verilecek bir ölçüt belirlendikten sonra kümeler arasındaki riski maksimize edecek şekilde cevaplar bölünmektedir. En iyi bölünmeyi bulmak için her soruda bu işlem tekrar edilmektedir. Bir soru için grup oluşturulduktan ve gruplar arasındaki risk maksimize edildikten sonra oluşan iki grup için aynı işlemler devam ettirilmektedir. Bu işlemler istatistik olarak anlamlı bir fark bulunana kadar devam ettirilip istatistik olarak anlamlı bir fark bulunmadığı

durumlarda ise son verilmektedir. Ayrıştırma işlemi tamamlandıktan sonra ise o grup içerisinde yer alan gözlemlerin oranına göre grup değerlendirilmektedir [81].

Büyük hacimli verileri kullanarak ona uygun bir karar ağacının elde edilmesi amacıyla birçok yöntem geliştirilmiştir. Regresyon ağaçları yöntemi de bunlardan biridir. Bağımsız değişken/değişkenler ile bağımlı değişken arasındaki ilişkiyi inceleyen regresyon analizi (basit ve çoklu), bir takım varsayımların (doğrusallık, normallik, homojenlik, toplanabilirlik gibi) yerine getirilmesinden sonra uygulanabilen bir test istatistiğidir. Varsayımların yerine gelmemesi durumunda bilinen bir takım dönüştürme (transformation) işlemlerine tabi tutulacak olan veri kümesi, uygun hale getirilmeye çalışılmaktadır. Bu da veri setindeki orijinal değerlerin ya logaritmik dönüşümleri, ya da karekök gibi dönüşüm yöntemleri ile yapılabilmektedir. Parametrik yöntemlerde her ne kadar dönüştürme metotları kullanılarak ön koşul varsayımlar yerine getirilmeye çalışılıyorsa da, yapılan analiz dâhilinde veri setine ilişkin yanlı (biased) sonuçlar elde edilmesi söz konusu olabilmektedir. Bu nedenle alternatifin olmadığı durumlarda başvurulması verileri dönüştürme yoluna gidilmesi istatistikî açıdan daha doğrudur. Bahsedilen avantajlı yönleriyle çoklu regresyon analizine alternatif sayılabilecek ve çoklu regresyon analizinin gerektirdiği bir takım varsayımları taşımayan Sınıflandırma ve Regresyon Ağacı'nın (Classification and Regression Tree; CART) kullanılması uygun olur. Sınıflama ve regresyon ağacı, sadece bağımlı değişken ile bağımsız değişken arasındaki ilişkinin yapısını araştırmakla kalmaz; aynı zamanda bağımsız değişkenlerin birbirleri ile olan etkileşimlerini de ortaya koymaya çalışır. Sınıflama ve regresyon ağacı analizinin kullandığı güçlü algoritma, bağımsız değişkenlerin bağımlı değişkenle ilişkisini değerlendirmede ve model içindeki etkileşim yapısını çözümlemede önemli avantajları mevcuttur. Sınıflama ve regresyon ağacının sahip olduğu algoritma, benzerlik gösteren (similarity) değişkenlerin aynı ağaç düğümünde toplanmasına dayalı olup, bütün oluşturduğu alt dalları (sub-branches) bağımlı değişken olan kök düğüme bağlamayla son bulmaktadır. Kullanılan veri çeşidine göre parametrik veya parametrik olmayan yöntemler grubunda yer alabilen regresyon ağacı yönteminde, bağımsız değişkenlerin kesikli veya sürekli olması önem arz etmediği gibi, parametrik testlerin önemsediği normallik, homojenlik ve doğrusallık gibi varsayımları da dikkate almamaktadır. Araştırmacılara sağladığı bu avantajlarla

birlikte elde ettiği sonuçların diyagram biçimindeki sunumu, tabloyla gösterimin ötesinde diğer araştırmacılar için açıklayıcı olmaktadır [82].

3.3.1.1. Verinin Kullanımı

Sınıflandırma ağaçlarının oluşturulabilmesi için öncelikle girdi verisine, yani örneklere gereksinim vardır. Örnekler sayısal ya da kategorik niteliklerden oluşabilir. Bunun yanı sıra bir adet hedef niteliğe gereksinim vardır. Hedef nitelik çıktı değerlerini içerir. Sınıflandırma sürecinde sınıflandırma ağacının eğitiminde kullanılacak verinin seçimi gerekir. Temizlenerek kullanılmaya hazır hale getirilmiş tüm veri ağacın eğitilmesinde kullanılmaz. Aksi takdirde elde edilecek ağacın doğruluğunu test etme olanağı yoktur. Dolayısıyla, ilgili veri kümesi ikiye bölünür. Bu kümelerden biriyle (eğitim kümesi) ağaç oluşturulurken, diğeri (test kümesi) ağacın test edilmesi işleminde kullanılır. Böylelikle bu karar mekanizmasına ne kadar güvenileceği bilgisi elde edilmiş olur [83].

3.3.1.2. Öğrenme Süreci

Eğitim için ayrılan veri kümesi ile önce sınıflandırma ağacı ve onun bir sonucu olarak karar kuralları türetilir. Karar ağacı eğitim verisinden hareketle bir öğrenme olayı sonucunda oluşturulur. Karar ağaçlarının oluşturulması için çok sayıda veri madenciliği algoritması (C5.0, CHAID, ID.3, C4.5, QUEST, vb.) geliştirilmiştir.

Karar ağaçları oluşturulurken kullanılan algoritmanın ne olduğu önemlidir. Kullanılan algoritmaya göre ağacın şekli değişebilir. Değişik ağaç yapıları da farklı sınıflandırma sonuçları verecektir. Kök denilen ilk düğümü oluşturan A_i'nin farklı olması, en uçtaki yaprağa ulaşılırken izlenecek yolu ve dolayısıyla sınıflandırmayı da değiştirecektir. Gerek kök düğümün gerekse de bundan sonraki her bir düğümün belirlenmesinde en büyük kriter o noktadan dallara ayrıldığında veritabanının geri kalan kısmı belli eşit parçalara bölünmüş olmasıdır. Örneğin veritabanında bulunan cevap evet/hayır gibiyse iki eşit parçaya, evet/hayır/belki gibi üç değişkenliyse

mümkün olduğunca üç eşit parçaya bölünmesi istenmektedir. Burada amaç, en kısa yoldan istenen yanıta ya da sınıfa ulaşmaktır [83].

Karar kuralları, karar ağacının kökten başlayarak yapraklara doğru yorumu yapılarak elde edilir. Elde edilen karar kuralları karar verme sürecinde kullanılır. Bir veri kümesine sınıflandırma algoritması uygulanarak karar ağacı elde edilmektedir. Eğitilmiş olan bu karar ağacı kullanılarak yeni bir örnek hakkında karar verilebilir.

3.3.1.3. Karar Ağaçlarının Elde Edilmesi

Ağaç yapısı benzeri akış şemaları olarak bilinen karar ağaçları (decision trees), kök düğüm (root node) ile başlamakta, dallarla (branches) birbirlerine bağlanan iç düğümleri (internal nodes) takip eden yaprak düğümleri (leaf nodes) ile son bulmaktadır. Her bir düğüm, bir değişken üzerinde gerçekleştirilen sınamayı, her bir dal ilgili değişken üzerinde gerçekleştirilen bölümlemeleri ve her bir yaprak da öngörünün sonucunu ifade etmektedir [84].

Bir başka deyişle karar ağaçları, bir kategori ya da değerle sonuçlanan kural serilerini betimleme yöntemidir. Kategorik bağımlı değişkenin kullanıldığı karar ağaçlarına sınıflandırma ağaçları (classification trees), sürekli bağımlı değişken kullanılanlara ise regresyon ağaçları (regression trees) adı verilmektedir [85].

Karar ağaçları ile sınıflandırma süreci kök düğümden başlar. Ağacın her bir düğümü bir niteliği tanımlar. Düğümlerden niteliğin her bir değerine göre dallar ayrılır. Bu dallanma süreci yapraklar elde edilinceye dek devam eder. Karar ağacı bilginin en iyi gösterim biçimlerinden birisi olmasına karşın, bazı durumlarda karmaşık ve yorumlaması oldukça zor olan bir yapıya sahip olabilir [86]. Bir karar ağacında sadece bir kök düğüm olabilir. Diğer simgeler için bir sayı sınırlaması yoktur.

3.3.1.4. Karar Ağaçlarının Avantajları Ve Dezavantajları

Her tekniğin kendine göre avantaj ve dezavantajları vardır. Karar ağaçlarını, diğer yöntemlere göre daha avantajlı kılan bazı özellikler şu şekilde sıralanabilir:

- Karar ağaçları çok sayıda nitelik sayısına ve çok büyük hacimli veri kümelerine kolayca uygulanabilir. Bunun sonucunda daha hızlı uygulama geliştirme olanağı elde edilmiş olur. Elde edilen sonuçların yorumu, diğer yöntemlere göre daha kolay ve kullanışlıdır [76].
- Parametrik olmayan yöntemler arasında olması nedeniyle diğer çok değişkenli tekniklerin gerektirdiği istatistiksel varsayımlar söz konusu değildir.
- İlişkilerin yönünü ve önem sırasını görsel olarak ortaya koyması bir diğer avantajdır [87].
- Hatalı verilere sahip veri kümelerinin yanı sıra eksik değerler içeren veri kümeleri ile çalışma olanağı sağlar [88].
- Sürekli değere sahip niteliklerin yanı sıra kategorik niteliklerin de kullanılması mümkündür. Karmaşık ilişkiler ağaç yapısı ile daha anlaşılır biçimde ortaya konulmaktadır [89].

Karar ağaçlarının dezavantajlarını da göz önüne almak gerekir. Bunlar aşağıdaki gibi sıralanabilir.

- Büyük hacimli verilerin kullanıldığı ve çok sayıda sayısal değişkenin yer aldığı uygulamalarda, karar ağaçlarının elde edilmesi diğer sınıflandırma yöntemlerine göre daha fazla hesaplama maliyeti gerektirebilir [89].

- Karar ağaçları için geliştirilen bazı algoritmalar (ID3 gibi) sadece kategorik niteliklerle çalışır [88]. Bu husus, söz konusu algoritmanın bir sınırlaması olarak değerlendirilebilir.

3.3.2. Karar Kurallarının Belirlenmesi

Karar ağaçları bilgi keşfi sırasında pek çok test gerçekleştirerek, hedefi tahmin etmede en iyi sırayı bulmaya çalışırlar. Her bir test karar ağacındaki dalları oluşturur ve bu dallar da diğer testlerin gerçekleşmesine neden olur. Bu durum, test işleminin bir yaprak düğümünde (leaf node) sonlanmasına kadar devam eder.

Kökten hedef yaprağa kadar olan yol, hedefi sınıflandıran "kural" olarak adlandırılır. Kurallar "eğer-sonra" (if-then) yapısındadır [90]. Bu yapı, elde edilen kuralların yeni veriye uygulanmasında kolaylık sağlar.

3.3.3. Kuralların Geçerliliğini Doğrulama

Karar ağacının eğitilmesi, yani öğrenme işlemi eğitim kümesi ile yapılır. Eğer elde edilen karar kuralları test kümesi için geçerli değilse, bu kuralların kestirim içinde kullanılması söz konusu olamaz. Kestirim yapabilmek için belirlenen karar ağacının ve buna bağlı karar kurallarının başarılı olduğunun kanıtlanması gerekir [91].

3.3.3.1. Sınıflandırıcı Doğruluğu

Eğitim kümesi verisi ile eğitilen ağacın geçerliliği test edildikten sonra, sonuçlar olumlu ise bir sonraki adımda karar ağacı yeni değerlerin kestirilmesinde kullanılır.

Sürekli değerlerin kestirimini yapmak için istatistiksel yöntemlerden geniş ölçüde yararlanılır. Tek değişkenli veya çok değişkenli regresyon modelleri bu tür kestirim modelleri oluşturmak için kullanılabilir. Örneğin, yeni çıkacak bir ürüne olan talebi tahmin etmek için geçmiş yılların ücret seviyesi bilgilerinden yararlanılarak bir regresyon modeli oluşturulabilir. Bu model yardımıyla, bu ürün için belirlenen bir fiyatın satışlar üzerindeki etkisi ortaya konulabilir.

Sınıflandırma işlemlerinde, eğitim verisi üzerinde bir sınıflandırma algoritması uygulanarak sınıflandırıcıya ulaşılır. Ancak doğal olarak sınıflandırıcı doğruluğunun belirlenmesi önem taşımaktadır. Sınıflandırma işlemi sınıflandırıcı yardımıyla

yapılır. Veri kümesini kullanarak bir sınıflandırıcı elde edildikten sonra bu sınıflandırıcının doğruluğunun sınanması gerekir. Bu amaç için birçok yöntem geliştirilmiştir. Holdout, Çapraz-doğrulama ve Bootstrap yaygın biçimde kullanılan doğrulama yöntemleridir [84].

3.3.3.2. Doğruluk Değeri

Bir sınıflandırma modelinin doğruluğunun belirlenmesi süreçteki en önemli adımlardan birisidir. Model, eğitim verisine dayalı olarak oluşturulduktan sonra sınıfları veya test veritabanı değerlerini tahmin etmek için kullanılır. Test verisi üzerinde modelin çalıştırılmasından sonra elde edilen sonuçlar değeri bilinen gerçek verilerle karşılaştırılır. Yanlış tahmin edilen veya sınıflanan örnek sayısının toplam test örnek sayısına oranı modelin hata oranını verir. Benzer olarak doğru sınıflanan veya tahmin edilen verinin toplam test örnek sayısına oranı da doğruluk oranını verir. Bir başka şekilde ifade edilirse, "doğruluk oranı = 1- (hata oranı)" şeklinde yazılabilir [92].

3.3.4. Kuralların Uygulanması ve Tahmin

Kuralların doğruluğu test edildikten sonra bu kurallar kestirim amacıyla kullanıma açılır. Sisteme daha önce eğitim ve test aşamasında verilmemiş değerlerle tahmin yürütmesi sağlanır.

3.3.5. Karar Ağaçlarında Entropiye Dayalı Bölünme

Sınıflandırma ağaçlarının oluşturulması esnasında hangi nitelikten başlayarak bölünmenin gerçekleştirileceği en önemli noktadır. Bölünmenin başlayacağı niteliğin seçilmesi amacıyla birçok yöntem geliştirilmiştir. Bunlar arasında yer alan kazanç ölçütü ve kazanç oranı, entropi tabanlı bölünme konusunda sıkça kullanılan ölçütlerdir [91].

3.3.5.1. Entropi

Felsefe, teoloji ve bilimde de yer alan entropi sözcüğü "enerji" gibi korunabilme özelliği olmayan bir sistemdeki rastgelelik, belirsizlik ve düzensizliğin ölçüsü olarak tanımlanabilir. Kısaca entropi bir sistemdeki belirsizliğin ölçüsüdür. S bir kaynak olsun. Bu kaynağın $\{m_1, m_2, ..., m_n\}$ olmak üzere n mesaj üretilebildiğini varsayalım. Tüm mesajlar birbirinden bağımsız olarak üretilmektedir ve m_i mesajlarının üretilme olasılıkları p_i'dir. $P = \{p_1, p_2, ..., p_n\}$ olasılık dağılımına sahip mesajları üreten S kaynağının entropisi;

$$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{3.49}$$

olarak ifade edilir.

İlk olarak istatistiksel fizikte kullanılan bu kavram, bilişim teorisine, bu teorinin yaratıcısı kabul edilen Claude Elwood Shannon tarafından uyarlanmıştır. Öz bilgi (self information) olarak adlandırdığı bir nicelikten bahseden Shannon'a göre bir A mesajının taşıdığı bilgi, aşağıdaki bağıntıda ifade edildiği gibi, onun gerçekleşme olasılığına bağlıdır ve bu olasılığın x tabanına göre eksi işaretli logaritması ile ifade edilir [93].

$$I(A) = \log_x \frac{1}{P(A)} = -\log_x P(A) \tag{3.50}$$

Bilginin birimi logaritmanın tabanına bağlıdır. Eğer taban 2 ise birim bit'tir, e ise birim nat'tır, 10 ise birim hartley'dir. Bilgisayar dünyasında 0 ve 1'lerle yani bit'lerle çalışıldığı için, logaritmanın tabanı 2 kabul edilir: Bu bağıntıya göre, iletilen mesajın olasılığı 1/8 ise, mesajın ilettiği bilgi $-\log_2\left(\frac{1}{8}\right) = 3$ bit olarak ifade edilir.

Bu formülde verilen öz bilgi, o bilgiyi ifade edebilmek için kaç bit kullanılması gerektiğini gösterir. Bu eşitlik, yüksek olasılığa sahip mesajların düşük bilgi içerdiğini, düşük olasılığa sahip mesajların ise yüksek bilgi içerdiğini göstermektedir.

Entropi tanımının ortaya koyduğu gibi,

- Örnekler aynı sınıfa aitse entropi = 0
- Örnekler sınıflar arasında eşit dağılmışsa entropi = 1
- Örnekler sınıflar arasında rastgele dağılmışsa 0 < entropi < 1

olur.

3.3.5.2. Kazanç Ölçütü veya ID3 Algoritması

Bölünmenin nereden başlayacağı konusunda kullanılan yöntemler arasında ID3 ve onun daha gelişmiş biçimi olan C4.5 sayılabilir [94].

ID3 algoritması, öğrenme süreci sonunda bir karar ağacı üretir. Karar ağacı şeklinde kural üreten algoritmaların en iyilerinden biri olarak kabul edilmektedir. Örnek kümesinde geçen karakteristiklerin bilgi kazancını hesaplayarak verilen örnekler kümesini alt kümelere ayırır ve karar ağacını bu şekilde oluşturur (Bilgi kazancının hesaplanması sırasında bağıl frekans dikkate alınır). Bu algoritma verilen bir örneği sınıflandırmak için gerekli olan kural veya kuralları minimum sayıda test yaparak oluşturmayı amaçlamaktadır. ID3 algoritması, bilgi kazancı en fazla olan karakteristiği seçerek bu karakteristiği ilk aşamada ağacın kökü, diğer aşamalarda ise bir düğüm (alt set için kök) olarak alır ve buna göre karar ağacını dallandırır.

Quinlan entropy kurallarını içeren bilgi teorisini kullanmıştır. Shannon ve Weaver'ın Bilgi Teorisinde temel olarak kaynak, mesaj ve alıcı vardır. Bu sistemde bilgi, mesaja bakılarak değil de, alıcıya bakılarak elde edilir. Alıcı mümkün olan mesaj uzayı bilgisine ve bu mesajların olasılıklarına sahiptir. Ağaçlardaki bazı düğümler ve bu düğümlerdeki kararlar anlamsız ve gereksiz olabilmektedir. Ancak bu tip düğümler de negatif–pozitif olay balanslarına sahiplerdir. İşte bu şekilde sınıflandırma yapılabilmektedir [95].

Örnek olarak X düğümünde 5 pozitif ve 3 negatif olay var. Bu noktada yapılacak bir sınıflandırmanın pozitif olasılığı 5/8'dir, negatif olasılığı 3/8'dir. İşte bu olasılıksal

sınıflandırmayı türetme yeteneğinin anlamı şudur: Doğru olarak sınıflandırılmış bir örneğin söylediği mesajın bilgi içeriği artık hesaplanabilir. Öyle ki bir tablonun sonuçları mesaj olsun ve mesajlar iki değere sahip olsunlar. Bu değerlerle birlikte p bilgisi pozitif olasılığını, q bilgisi negatif olasılığını gösterir. Bu iki değerin toplamı zaten 1 (p+q) olmak zorundadır. Doğru sınıflandırma veren bir mesajın bilgi içeriği

$$I(p,n) = -p\log_2 p - q\log_2 q \tag{3.51}$$

şeklinde hesaplanır.

Bu formül genel bilgi içerik formülünün özel bir durumudur. Çünkü özel olarak iki olasılık mevcuttur: pozitif ve negatif.

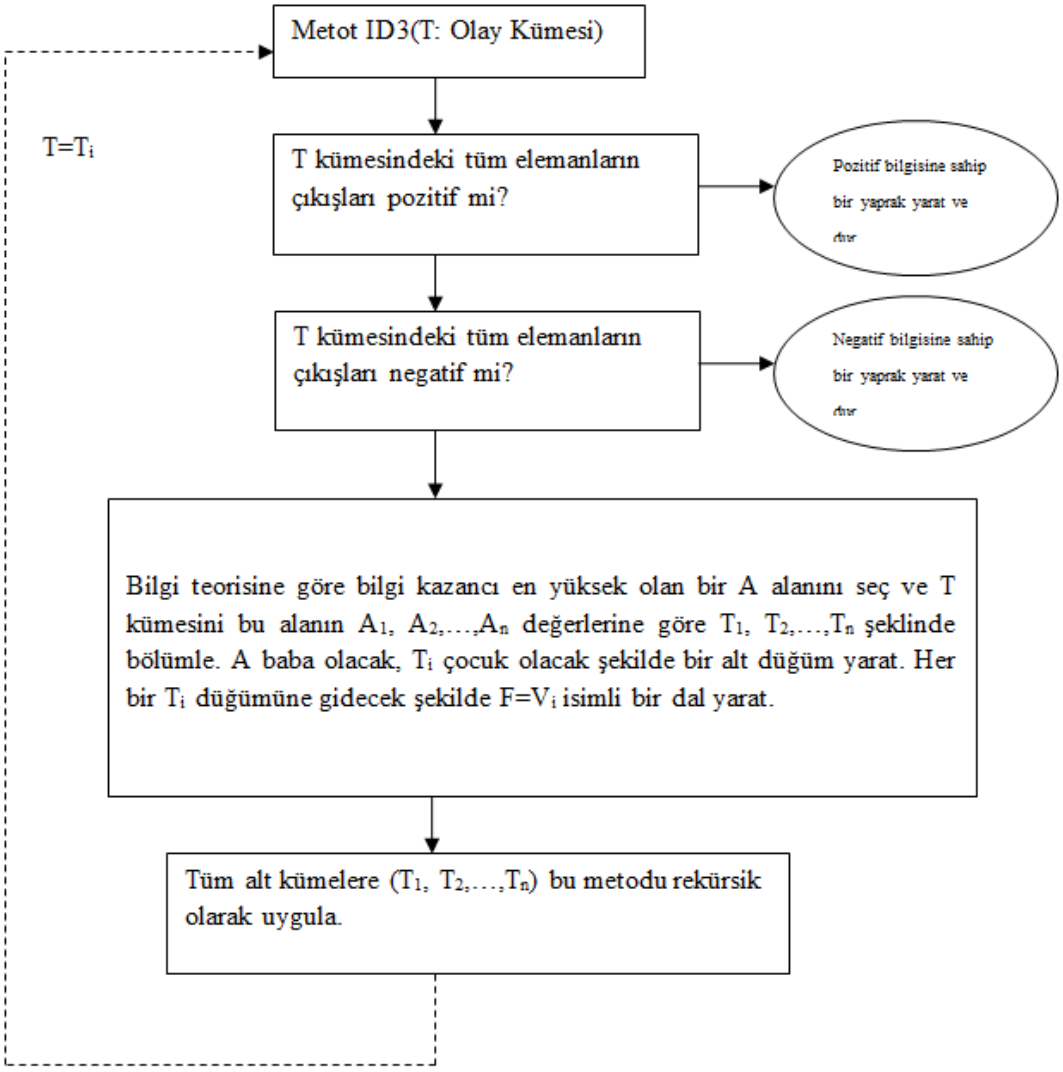
$\{A_1,A_2,...,A_n\}$ değerlerine sahip A özelliği ağacın bölünmesi için kullandığında,T kümesi $\{T_1,T_2,...,T_n\}$ şeklinde bölünecektir. Bu bölümleme de T kümesindeki A özelliğinin A_i olduğu bölgelere T_i densin. Bu kümedeki pozitif olayların sayısını p_i temsil etsin, negatif olayların sayısını n_i temsil etsin. Bu durumda T_i alt ağacı için beklenen bilgi gereksinimi ise $I(p_i,n_i)$ olur. T ağacı için beklenen bilgi gereksinimi tüm T_i ağaçlarının beklenen bilgi gereksinmelerinin ağırlıklı ortalamalarının toplamı olur ve

$$E(A) = \sum_{i=1}^{n} \frac{p_i+n_i}{p+n}\, I(p_i n_i) \tag{3.52}$$

şeklinde hesaplanır. Dolayısı ile A özelliği üzerinden sağlanan bilgi kazancı

$$Bilgi\ kazancı\ (A) = I(p,n) - E(A) \tag{3.53}$$

şeklinde ifade edilir. Şekil 3.16 ID3 algoritmasının aşamalarını ifade etmektedir.



Şekil 3.16. ID3 algoritması akış diyagramı.

Bilgi gereksinimi ve bilgi kazancı ID3 algoritmaları için iki önemli kavramdır. Belirleyici bir sınıflandırma için bilgi ihtiyacı aslında doğru sınıflandırmayı sağlayan mesajın bilgi içeriğinden başka bir şey değildir. Buna yönelik olarak yaratmak istenilen karar ağaçlarının amacı doğru soruları sormasıdır. Ve sonunda öyle bir noktaya ulaşılmalı ki, bu noktanın karar için bilgi gereksinimi 0 olsun. İşte bu noktada ID3 algoritmasının yaptığı şey, ağacı doğru kurmaktır. Kurulu karar ağacının her seviyesinde geriye kalan bilgi gereksinimi (remaining information required) minimize edilir.

Bu bilgiler ışığında Çizelge 3.1'deki örnek ele alınsın.

Çizelge 3.1. Örnek bir olay kümesi.

	Büyüklük	**Renk**	**Biçim**	**Sonuç**
1	Orta	Mavi	Tuğla	Evet
2	Küçük	Kırmızı	Kama	Hayır
3	Küçük	Kırmızı	Küre	Evet
4	Geniş	Kırmızı	Kama	Hayır
5	Geniş	Yeşil	Sütun	Evet
6	Geniş	Kırmızı	Sütun	Hayır
7	Geniş	Yeşil	Küre	Evet

Bu olayların hepsi birden *evet* ya da hepsi birden *hayır* olamadıklarından bilgi kazancı en yüksek olan özellikten başlayarak bölümleme işlemi gerçekleştirilir. Örnek uzayda 4 adet pozitif olay olduğunda bir olayın pozitif gelme olasılığı 4/7=0,57'dir. Negatif gelme olasılığı 3/7=0,43'dür. Bundan dolayı doğru bir sınıflandırma için gerekecek bilgi kazancı

$$-(0{,}57x\ \log_2 0{,}57) - \ (0{,}43x \log_2 0{,}43) = 0{,}99$$

olur.

Şimdi her bir özellik için bilgi gereksinmeleri hesaplansın. Büyüklük özelliği için küçük, orta ve büyük olmak üzere üç tip değer vardır. *Büyük* değeri kümenin {4,5,6,7} elemanlarını kapsamaktadır. Bu küme içerisinde 2 *evet* ve 2 *hayır* sınıfı bulunduğundan ve p,n = 2/4=0,5 olduğundan gereken bilgi kazancı

$$-(0{,}5\ x\ \log_2 0{,}5) - \ (0{,}5\ x \log_2 0{,}5) = 1$$

olur. Aynı işlem *küçük* değeri {2,3} elemanlarını içermektedir. Bu kümede 1 *evet* ve 1 *hayır* sınıfı vardır. Bu durumda gereken bilgi kazancı

$-(0{,}5\ x\ \log_2 0{,}5) -\ (0{,}5\ x \log_2 0{,}5) = 1$

olur. *Orta* değeri için hesap edildiğinde ilgili kümede sınıflardan sadece bir tanesi olduğundan sonuç 0 çıkar.

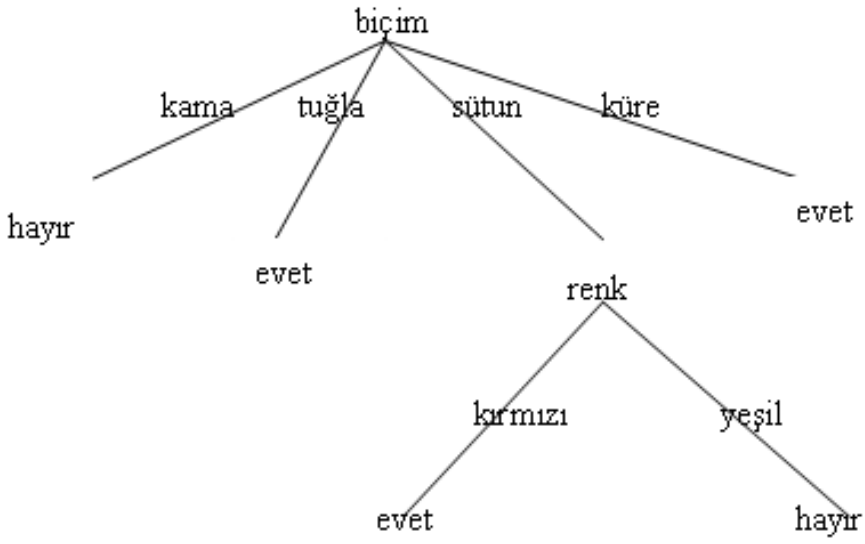
Şu anda büyüklük özelliği için beklenen bilgi gereksinimini hesap edilebilir. Bütün bilgi gereksinim sonuçları ilgili özellik değerlerinin orantısıyla çarpılarak toplanır ve

(1 x 4/7) + (1 x 2/7) +(0 x 1/7) = 0,86

olur. Büyüklük özelliği için beklenen bilgi kazancı mevcut bilgi ihtiyacından beklenen bilgi ihtiyacı çıkarılarak hesaplanır;

0,99 – 0,86 = 0,13

Aynı işlemi renk ve biçim için yapılarak işlem tamamlanır. Bilgi kazancı *renk* için 0,52 ve *biçim* için 0,7 bulunur. Bu koşullar altında *biçim* özelliği en yüksek bilgi kazancına sahip özellik olur. Buna göre ağaç tekrar oluşturulduğunda Şekil 3.17'deki gibi daha sade bir şekil alır.



Şekil 3.17. Çizelge 3.1'in ID3 ile oluşturulmuş karar ağacı.

3.3.5.3. Kazanç Oranı veya C4.5 Algoritması

C4.5 öğrenme algoritması Quinlan tarafından geliştirilmiştir. Bu algoritma da tıpkı ID3 algoritması gibi yukarıdan aşağıya doğru karar ağacı üretir. Günümüzde en çok bilinen ve en çok kullanılan algoritma olarak kabul edilmektedir. C4.5 Algoritması ID3 algoritmasının bütün özelliklerini taşıyan bir algoritmadır. Ayrıca ID3 için yapılan açıklamalarda yer alan kavramlara yeni kavramlar eklenmiştir. Bölünme-Dağılma Bilgisi (Split-Info), özelliklerin kayıp değerleriyle baş edilmesi, sayısal özellik değerlerinin hesaplara katılması bu kavramlardan en önemlileridir [96].

ID3 algoritması karar ağacının her düğümü için bilgi kazancı değerlerini hesaplar. C4.5 ise bilgi kazancı ile beraber alt sette yer alan karakteristiklerin bilgi kazanç oranlarını da hesaplayarak bilgi kazanç oranı en yüksek olan karakteristiği düğüm noktası olarak seçer. Karar ağacının her dalı sadece bir tek sınıfa karşılık gelinceye kadar işlemleri sürdürür. C4.5 (Kazanç Oranı Tabanlı) algoritması karar ağacı oluşturma prosedürü, düğüm noktalarının belirlenmesi dışında ID3 (Bilgi Kazancı Tabanlı) ile aynıdır. Daha sonra karar ağacını kural setine dönüştürür. Bir kategorik özelliğin olası değer çeşitliliği ne kadar yüksek olursa o özelliğin bilgi kazancı gereksiz bir şekilde yüksek çıkar ve bu durum ağacın doğruluğunu kötü bir şekilde etkiler. Bu tip özellikler işe yaramadıkları gibi bilgi kazancı yüksek özelliklerin de önüne geçip veride gizlenmiş kuralların çıkarılmasına engel teşkil ederler. Karar ağaçlarının oluşturulması esnasında bölünmenin nereden başlayacağı konusunda kazanç miktarı kavramı kullanılabilir. Daha duyarlı sonuçlar elde etmek için bu ifade yerine, "Kazanç oranı" adı verilen bir ifade tercih edilebilir. Bunun için "bölünme bilgisi" (split information) adı verilen kavramdan yararlanılır. Bölünme bilgisi ile ilgili açıklamalar aşağıda verilmiştir. A bir özellik, A_i bu özelliğin değerleri; T_i, A_i özelliğinin bu veride kaç kez tekrarlandığı, T ise ele alınan olay sayısını temsil etsin. Bu durumda bölünme bilgisi (3.54) ile ifade edilir [77].

$$split\ info(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \log_2\left(\frac{|T_i|}{T}\right) \tag{3.54}$$

Bu bölünme bilgisinin tüm özelliklerin bilgi kazanç formülüne bölen olarak eklenmesiyle elde edilen oran kazanç oranı olarak ifade edilir. Bu durumda A özelliğinin kazanç oranı; Kazanç oranı = bilgi kazancı (A) / bölünme bilgisi (A) şeklinde hesap edilir [77]. Kazanç oranı (3.54) ile bağlantılı olarak aşağıdaki eşitlikle açıklanır.

$$gain\ ratio(X) = \frac{gain(X)}{split\ info(X)} \tag{3.55}$$

3.3.6. Karar Ağacının Budanması

Bir karar ağacı oluşturulduğunda, birçok dalda, öğrenme verisindeki gürültü ve kayıplardan dolayı aykırılık oluşacaktır. Ağacın budanma metodu bu sorunu ortadan kaldırmaya yardımcı olabilir. Bu metot, tipik olarak en az güvenilir olan dalı istatistiksel olarak hesaplayıp kaldırmaktan ibarettir ve daha hızlı ve güvenilir bir sınıflandırma ile sonuçlanır. İki adet budama yöntemi vardır.

Bunlardan birincisi, önceden budama yöntemidir. Bu yöntemde öğrenme verisi sınıflandırılırken ağacın o dalının ileriye yönelik devam edip etmeyeceğine önceden karar verilir ve gerekiyorsa, geri kalan bölünmeden sonra geriye kalan verinin sınıflandırılması durdurularak, en fazla hedef değeri taşıyan değer yaprak yapılır. Bu yöntemde, önceden bir eşik değeri belirlenir. Bu eşik değerini aşmayan bilgi kazançlarına sahip olan nitelikler gruplandırılır. Program devam ederken bu bilgi kazancının düştüğü noktada ağacın büyümesine izin verilmeden diğer dala geçilir.

Her iki koşulda da, bu eşik değerini belirlemek işin en zor kısmıdır. Çünkü eşik değeri çok yüksek tutulursa ortaya çıkan ağaç çok fazla basit ve genel kurallardan oluşan bir ağaç olur. Eşik değeri çok düşük tutulursa, ağacın sınıflandırması çok özele inebilir ve test verisi üzerinde doğru sonuçlar ortaya çıkmayabilir.

İkinci yöntem, sonradan budama yöntemidir. Tamamen büyümüş bir ağaç üzeride uygulanır. Tüm dalların çıkardığı kurallar denenerek, bunlardan en fazla hata oranını oluşturan dal budanır. Böylece ortaya daha basit bir ağaç yapısı çıkartılabilir [84].

C4.5 karar ağaçlarında kullanılan ön budama yöntemi, daha az hesaplama içermesi, veri setinin ayrılması için en iyi yolu araştırması ve bilgi kazancının değerlendirilmesi yönlerinden önemli avantajlara sahiptir. Bu değerlendirme belirli bir eşik değerinin altına düştüğünde bölünme kabul edilmez ve veri için en uygun yaprak olduğuna karar verilir [80,97]. Eğer tek bir yaprağı olan alt ağacın veya bu ağacın en çok kullanılan dalının budanması beklenen hata oranını düşürecekse ağaç budanır. Alt dallardaki hata oranı azaldığından tüm ağaç için hata oranı azalacaktır. Budama işlemi sonunda hata oranının minimum hale getirildiği bir ağaç elde edilir.

3.4. LOJİSTİK REGRESYON ANALİZİ

İstatistiksel uygulamalarda, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi tanımlayabilmek amacıyla geliştirilen, yaygın olarak kullanılan alternatif tahminleme yöntemlerden birisi de lojistik regresyon analizidir. Son zamanlarda lojistik regresyon analizi kullanım kolaylığının yanında, sayısal olarak rahat yorumlanabilmesiyle ön plana çıkmış ve birçok uygulamada sıklıkla kullanılan bir yöntem haline gelmiştir.

Lojistik Regresyon Analizinin yaygın bir şekilde kullanılmaya başlanması ile katsayı tahmin yöntemleri daha fazla geliştirilmiş ve lojistik regresyon modelleri daha detaylı bir şekilde incelenmeye başlanmıştır. Cornfield (1962) lojistik regresyondaki katsayı tahmin işlemlerinde diskriminant fonksiyonu yaklaşımını ilk kez kullanarak popüler hale getirmiştir [98]. Lee (1984) basit dönüşümlü (cross-over) deneme planları için lineer lojistik modeller üzerinde durmuştur [99]. Bonney (1987) lojistik regresyon modelinin kullanımı ve geliştirilmesi üzerinde çalışmış olup Roberts ve diğerleri (1987) ise lojistik regresyonda standart ki-kare, olabilirlik oranı (G2), en çok olabilirlik tahminleri, uyum iyiliği ve hipotez testleri üzerinde çalışma yapmışlardır [100, 101]. Duffy (1990) lojistik regresyonda hata terimlerinin dağılışı ve parametre değerlerinin gerçek değerlere yaklaşımını incelemiştir [102].

Çok değişkenli istatistiksel verilerin sınıflandırılmasında kullanılan çok değişkenli istatistiksel yöntemlerden biri olan lojistik regresyon analizinde verilerin yapısındaki

grup sayısı bilinmekte ve bu verilerden hareketle bir ayrımsama modeli oluşturulmaktadır [103].

Lojistik regresyon analizinin temel amacı diğer regresyon yöntemlerinde olduğu gibi bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi incelemektir. Başka bir deyişle amaç minimum uygun sayıda değişken ile sonuç değişkeni ve açıklayıcı değişkenler arasındaki ilişkiyi tanımlayan kabul edilebilir modeli kurmaktır. Lojistik regresyon yönteminde bağımlı değişkenin sürekli olması gibi bir varsayım yoktur, özellikle bağımlı değişkenin iki veya daha çok değer aldığı durumlarda kullanılır [103].

Lojistik regresyon analizi, bağımlı değişkenin türüne göre 3 farklı şekilde kullanılabilir:

- İkili (Binary) Lojistik Regresyon,
- Sıralı (Ordinal) Lojistik Regresyon,
- İsimsel (Nomial ve Multinomial) Lojistik Regresyon.

İkili lojistik regresyon yönteminde sınıflayıcı değişken iki sonuçludur. Bu değişken sayısal veya kısa alfanümerik bir değişken olabilir. Analizde sınıflayıcı değişken bağımlı değişken olarak referans kabul edilir ve bağımsız değişkenlerle olan ilişkisi incelenerek sınıflandırmada kullanılacak tahmini regresyon denklemi kurulur. Kurulan denklem yardımıyla sınıfların tahminine çalışılır. Sıralı lojistik regresyon bağımlı değişkenin üç veya daha fazla cevaplı olması durumunda uygulanan bir yöntemdir. Ayrıca cevaplar arasında sıralı (ordinal) bir ilişki de olması gerekir. İsimsel lojistik regresyon yöntemi ise Sıralı lojistik regresyona benzer ancak burada bağımlı değişkenin aldığı cevapların sıralı olması şartı aranmamaktadır. Elde edilen gözlem değerlerine lojistik regresyon analizi uygulanacağına karar verildikten sonra katsayıların tahmini, yorumlanması, katsayılara ilişkin hipotez testlerinin yapılması ve modelin başarısının değerlendirilmesi gerekmektedir.

Lojistik regresyon analizinde bağımlı değişkene bağlı olarak bağımsız değişkenlerin dağılımının doğrusal olmadığı durumlarda kullanılabilir olduğu görülmektedir [104].

Bağımsız değişkenlerin doğrusal olmaması sebebi ile eşitliğin ifadesi çoklu regresyon modellerinden biraz daha karmaşık olabilmektedir. Lojit modeller;

$$ln\ \frac{P}{1-P} = A + \sum B_j\ X_{ij} \tag{3.56}$$

eşitliği ile ifade edilmektedir.

3.4.1. Lojistik Regresyon Analizinde Değişken Seçimi

Lojistik Regresyon Analizi; sürekli, kesikli, ikili ya da bunların bir karışımı olan veri setlerinden kategorik bir sonucu tahmin etmeye olanak sağlar. Lojistik regresyon modellerinde kategorik bağımsız değişken/değişkenler, sadece sürekli bağımsız değişken/değişkenler veya hem kategorik hem de sürekli bağımsız değişkenler kullanılabilir [105].

Bağımlı değişkendeki değişimi açıklayabilmek için kurulan bir regresyon eşitliğine girecek değişken sayısı ne kadar çok olursa, eşitlik o kadar küçük hata taşımaktadır. Ancak, gerek bağımsız değişkenlerin birisiyle gözlem elde etmenin getireceği yük, gerekse bu gözlemleri belirli bir zaman aralığında yapma mecburiyetinin getireceği zorluklar ve olası hatalar bağımsız değişken sayısını azaltmayı zorunlu kılabilir. Bu nedenle, sınıflandırma tahmininin doğruluğu mümkün olduğunca yüksek tutulmalı; ayrıca ekonomik yük ve zorlukların yanı sıra, fazla değişkenle ilgili veri elde etmenin getirebileceği sistematik hataları mümkün olduğunca azaltabilecek sayıda bağımsız değişkenle çalışılması araştırıcılar açısından önemli bulunmaktadır [106].

Lojistik regresyon analizinde değişken seçimi analize bağımsız değişkenin nasıl dâhil edileceği ile ilgilidir. Farklı yöntemler kullanılarak değişkenlerin seçimi yapılabilmektedir. Diğer çok değişkenli yöntemlerde olduğu gibi adımsal seçim modellerinde bir sonraki aşamada hangi değişkenin modele dâhil edilebileceğine karar verilmektedir. İstatistiksel olarak algoritmalardan hiçbirisi en iyi modeli sağlamayı garanti edememektedir. Bu aşamada farklı modellerin denenip bu

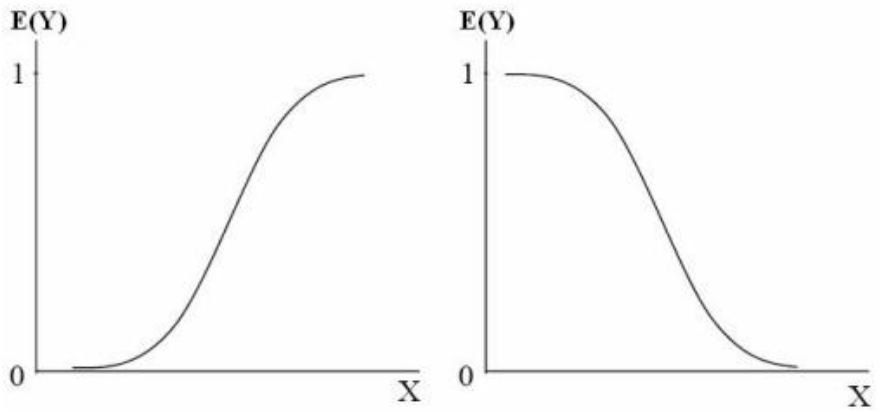
modeller arasından sınıflandırma başarısına göre seçim yapmak en iyi yaklaşım olmaktadır [107].

Kullanılan yaklaşımlar ise şunlardır:

- Tüm Değişkenlerin Modele Dâhil Edildiği Yaklaşım: Bütün değişkenler bir blok olarak tek aşamada modele dâhil edilir.
- İleri Seçim (Koşullu): İleriye doğru adımsal bir yöntemdir. Değişkenler modele teker teker alınarak kriterleri sağlamayanlar modelde tutulmaz. Değişkenler modele alınırken skor istatistiğinin önemine, çıkarılırken de koşullu parametre tahminlerine dayanan olabilirlik oranına göre karar verilir.
- İleri Seçim (Olabilirlik Oranı): İleriye doğru adımsal bir yöntemdir. Değişkenler modele alınırken skor istatistiğinin önemine, çıkarılırken de maksimum kısmi olabilirlik tahminlerine dayanan olabilirlik oranına göre karar verilir.
- İleri Seçim (Wald): İleriye doğru adımsal bir yöntemdir. Değişkenler modele alınırken skor istatistiğinin önemine, çıkarılırken de Wald istatistiğine göre karar verilir.
- Geriye Eleme (Şartlı): Geriye doğru adımsal bir seçim yöntemidir. Önce tüm değişkenler modele alınır daha sonra birer birer kriterleri sağlamayan değişkenler modelden çıkartılır. Tüm geriye doğru yöntemlerde önce tüm değişkenler alınıp sonra teker teker çıkarma yaklaşımı geçerlidir. Değişkenler modelden çıkarılırken koşullu parametre tahminlerine dayanan olabilirlik oranına göre karar verilir.
- Geriye Eleme (Olabilirlik Oranı): Geriye doğru adımsal seçim yöntemidir. Değişkenler modelden çıkarılırken kısmi olabilirlik tahminlerine dayanan olabilirlik oranına göre karar verilir.
- Geriye Eleme (Wald): Geriye doğru adımsal seçim yöntemidir. Değişkenler modelden çıkarılırken Wald istatistiğine göre karar verilir (SPSS Regression Models 16.0)

3.4.2. Lojistik Regresyon Modeli

Lojistik Regresyon Modeli (LRM), genel doğrusal modellerin binom dağılımlı bağımlı değişkenler için elde edilmiş olan özel bir biçimidir. Hem teorik hem de deneysel incelemeler bağımlı değişken iki sonuçlu iken cevap fonksiyonunun p/(1-p) şeklinin S veya ters S şeklinde olacağını göstermiştir. Şekil 3.18'de görüldüğü üzere bağımlı değişken bitiş noktaları dışında yaklaşık olarak doğrusaldır yani cevap fonksiyonları 0 ile 1 değerlerinde X ve Y eksenlerine asimptottur [108].



Şekil 3.18. İkili bağımlı değişkenin S ve ters S olasılık fonksiyonu grafikleri.

$$\pi(x)=\frac{e^{\beta_0+\beta_1 X_1+ \ldots .. +\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+ \ldots .. +\beta_p X_p}} \qquad (3.57)$$

Lojistik regresyon modeli Eşitlik 3.57'de ifade edildiği gibidir ve bu eşitlikteki [109];

$\pi(x)$:İncelenen olayın gözlenme olasılığını,

β_0 :Bağımsız değişkenler sıfır değerini aldığında bağımlı değişkenin değerini başka bir ifadeyle sabiti,

$\beta_1\beta_2 \ldots \beta_p$:Bağımsız değişkenlerin regresyon katsayılarını,

$X_1X_2 \ldots X_p$:Bağımsız değişkenleri,

p :Bağımsız değişken sayısını,

e :2,718 sayısını göstermektedir.

Lojistik regresyon modelinde bağımlı (sonuç) değişken 0 ve 1 gibi ikili değişken yapısında olup; risk belirten durum 1, diğer durum 0 ile gösterilir. Regresyon problemlerinde temel nokta verilen bir bağımsız değişkenin değerine bağlı olarak bağımlı (sonuç) değişkenin ortalama değerini bulmaktır. Bu değer koşullu ortalama olarak adlandırılır ve E(Y/x) ile gösterilir. Burada Y bağımlı değişkeni, x ise bağımsız değişkeni göstermektedir. Doğrusal regresyon analizinde koşullu ortalamanın x'in doğrusal bir denklemi olduğu varsayılır ve Eşitlik 3.58'deki gibi ifade edilir.

$$E(Y/x)=\beta_0+\beta_1 x \tag{3.58}$$

Bu eşitlik x'in aralığının $-\infty$ ve $+\infty$ arasında değişmesinden dolayı E(Y/x) 'in mümkün olan her değeri alabileceğini göstermektedir. Lojistik regresyon analizinde ise koşullu ortalama, Eşitlik 3.59'da ifade edildiği gibi 0'dan büyük yada eşit, 1'den küçük yada 1'e eşit olmak zorundadır.

$$0\leq E(Y/x)\leq 1 \tag{3.59}$$

Lojistik regresyon modelinde $E(Y/x)=\beta_0+\beta_1 x$ eşitliği 0-1 arasında sınırlı olasılık değerleri aldığı ve bu değerler sonsuz değerler alabilen açıklayıcı değişkenlerle ilişkilendirildiği için söz konusu eşitlik her zaman sağlanamamaktadır. Böylesi bir durumla karşılaşılmaması için en iyi çözüm sonuç değişkeni olarak ifade edilen olasılık değerinin çeşitli dönüşümlerle $-\infty$ ve $+\infty$ arasında tanımlı hale getirilmesidir. Gösterimi kolaylaştırmak için bu çalışmada lojistik dağılım kullanıldığında x bilindiğinde Y'nin koşullu ortalamasını göstermek için $\pi(x)=E(Y/x)$ dönüşümü yapılmıştır [110].

$$\pi(x)=\frac{1}{1+e^{-(\beta_0+\sum \beta_i X_i)}} \tag{3.60}$$

3.4.3. Modelin Parametre Tahmin Yöntemleri

Modelin katsayılarının tahmininde En Çok Olabilirlik Yöntemi, Yeniden Ağırlıklandırılmış İteratif En Küçük Kareler Yöntemi, Minimum Lojit Ki-Kare Yöntemi kullanılmaktadır. Açıklayıcı değişkenlerin hepsi sürekli ise minimum lojit ki-kare yöntemi, değişkenlerin hepsi kesikli ise en çok olabilirlik yöntemi, hem sürekli hem de kesikli ise ağırlıklandırılmış iteratif en küçük kareler yöntemi kullanılmaktadır [111].

3.4.3.1. En Çok Olabilirlik Yöntemi

Lojistik regresyon modelinde parametrelerin tahmin edilmesinde en çok kullanılan yöntem "En Çok Olabilirlik Yöntemi" dir. En çok olabilirlik yöntemi, doğrusal regresyon analizindeki en küçük kareler yöntemine benzerlik göstermektedir [112].

En çok olabilirlik yöntemi, istatistiksel modellerin tüm çeşitleri için yaygın olarak kullanılan genel bir tahmin yöntemidir. Popüler olmasının iki nedeni vardır: Birincisi, en çok olabilirlik kestiricilerinin, büyük örneklemlerde istenen bazı iyi özelliklere sahip olduğu bilinir. Genel olarak bu kestiriciler tutarlı, asimptotik olarak etkin ve normal dağılımlıdır. İkinci neden ise; çözüm için başka olasılıklar bulunmadığında, en çok olabilirlik kestiricilerinin türetilmesi gerekmektedir. En çok olabilirlik yöntemi, kategorik bağımlı değişkenler söz konusu olduğunda iyi sonuçlar vermektedir [113].

Eğer y, 0 ya da 1 olarak gösterilirse, $\pi(x)$ ifadesi, verilen x değeri için y'nin 1'e eşit olması koşullu olasılığını verir. Bu da $P(y = 1 \mid x)$ olarak gösterilir. ($\beta' = [\beta_0, \beta_1]$ rastgele değeri, parametrelerin vektörüdür.) $1 - \pi(x)$ niceliği, verilen x için y'nin 0'a eşit olması koşullu olasılığını verir ve bu da $P(y = 0 \mid x)$ olarak gösterilir. Dolayısıyla (x_i , y_i) ikilileri için $\pi(x_i)$, hesaplanmış x_i için $\pi(x)$ değerini göstermek üzere;

$y_i = 1$ olan ikililerin olabilirlik fonksiyonuna katkısı $\pi(x_i)$ "dir.

$y_i = 0$ olan ikililerin olabilirlik fonksiyonuna katkısı $1 - \pi(x_i)$ "dir.

Buna göre *(xi , yi)* ikililerinin olabilirlik fonksiyonuna katkılarını göstermek için şöyle bir yol izlenir:

$$\zeta(x_i) = \pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i} \tag{3.61}$$

Gözlemlerin bağımsız olduğu varsayımından dolayı, yukarıdaki (3.61) ifadesinde verilen terimlerin bir çarpımı olarak "olabilirlik fonksiyonu" şu şekilde elde edilir:

$$l(ß) = \prod_{i=1}^{n} \zeta(x_i) \tag{3.62}$$

En çok olabilirlik ilkesi, (3.62) eşitliğini maksimum yapan ß tahmininin kullanılmasını öngörmektedir. Ancak matematiksel olarak, bu eşitliğin logaritması ile çalışmak daha kolaydır. Buna göre "log olabilirlik" kavramı şöyle tanımlanır:

$$L(ß) = \ln[l(ß)] = \sum_{i=1}^{n}\{y_i \ln[\pi(x_i)] + (1 - y_i)\ln[1 - \pi(x_i)]\} \tag{3.63}$$

L(ß)'yi maksimum yapan ß değerini bulmak için, L(ß) değeri, β_0 ve β_1'e göre türevi alınıp 0'a eşitlenir. Böylece "olabilirlik eşitlikleri" diye adlandırılan aşağıdaki eşitlikler elde edilir;

$$\sum_{i=1}^{n} y_i - \pi(x_i) = 0 \tag{3.64}$$

$$\sum_{i=1}^{n} x_i[y_i - \pi(x_i)] = 0 \tag{3.65}$$

Doğrusal regresyonda, karelerin sapmaları toplamının ß'ya göre türevi alınarak elde edilen olabilirlik eşitlikleri bilinmeyen parametrelerde doğrusaldırlar ve çözümleri kolaydır. Ancak lojistik regresyon için (3.64) ve (3.65) eşitliklerindeki olabilirlik eşitlikleri, β_0 ve β_1'de doğrusal değildirler ve dolayısıyla çözümleri için özel yöntemlere ihtiyaç duyulur. Bu yöntemler doğaları gereği iteratif yani tekrarlıdırlar [110]. Tekrarlanan bu tahmin süreci, test edilmesi ve yeniden tahmin edilmesi iterasyon olarak adlandırılır. Olabilirlik fonksiyonundaki değişim sonraki aşamalarda ihmal edilebilir duruma gelinceye kadar çözüme devam edilir. Olabilirlik

fonksiyonunu maksimize edecek en iyi parametre kümesini oluşturmak için yapılan tüm bu işlemler için bilgisayar uygulamalı algoritmalar tasarlanmıştır [114].

(3.64) ve (3.65) eşitliklerinden elde edilen ß değeri "en çok olabilirlik tahmini" diye adlandırılır ve ß ile gösterilir. Genel olarak "^" sembolü bu niceliğin en çok olabilirlik tahminini gösterir. Örneğin, $\hat{\pi}(x_i)$, *π(xi)*'nin en çok olabilirlik tahminidir. Bu nicelik, *x = xi* olarak verildiği zaman, *y*'nin 1 olma koşullu olasılığı için bir tahmin verir. Bu da lojistik regresyon modelinin uyarlanan ve tahmin edilen değerini gösterir. Bu durumda (3.64) eşitliği aşağıdaki gibi olur;

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{\pi}(\mathrm{x_i}) \tag{3.66}$$

Verilen (3.66) eşitliği, (3.64) ifadesinden elde edilen eşitliktir ve *y*'nin gözlenen değerlerinin toplamının, beklenen (tahmin edilen) değerlerinin toplamına eşit olduğunu gösterir [110].

3.4.3.2. Yeniden Ağırlıklandırılmış İteratif En Küçük Kareler Yöntemi

Doğrusal regresyonda bilinmeyen parametreleri bulmak için sıklıkla kullanılan yöntem en küçük kareler yöntemidir. Bu yöntemle modele göre tahmin edilen *y* değerlerinin gözlemlenen değerlerden sapmalarının karesini minimize edecek β_0 ve β_1 elde edilir. En küçük kareler yöntemi bilinen varsayımlar altında oldukça iyi sonuçlar verir. Ancak iş bağımlı değişkenin kesikli olması durumuna gelince en küçük kareler aynı varsayımları sağlamaktan uzak kalır [114].

Gruplandırılmış verilerde *J* grubun her birinde n_j denemeden *r.* başarı elde edildiğinde başarı oranı $P_j = r_j / n$ olarak tanımlanabilir. $Var\ (r_j / n_j) = P_j\ (1- P_j) / n_j$ olduğundan, her binom dağılımlı gözlem için varyans değişmektedir.

Bu durumda lojit (r_j / n_j)'nin açıklayıcı değişkenler üzerinde $w_j = n_j / P_j\ (1- P_j)$ ağırlığı ile ağırlıklandırılmış regresyonu uygulanmalıdır. Ancak w_j ağırlık değerleri de P_j'nin bir fonksiyonu olduğu için en küçük kareler yöntemi iteratif olarak

uygulanacak ve ağırlık değerleri her adımda (kestirim değerlerine bağlı olarak) yeniden elde edilecektir [115].

3.4.3.3. Minimum Lojit Ki-Kare Yöntemi

Ağırlıklı en küçük kareler tahmin yönteminin özel bir biçimi olan ve Berkson tarafından geliştirilen bu yöntemde, 2*J çapraz tablolarındaki beklenen ve gözlenen lojit değerleri arasındaki farktan yararlanılmaktadır. Yöntem tekrarlı veriler olması durumlarında kullanılmaktadır. Bir önceki yöntemde verilen *Pj* olasılığı üzerinden yapılan lojit dönüşümü, bu yöntemde sonuç değişkenini oluşturmaktadır.

Tahminde kullanılan ağırlık değerleri *nj . Pj(1-Pj)* olarak elde edilmektedir. Bu bilgiler ışığında yöntem, lojit değeri olarak tanımlanan sonuç değişkeninin, açıklayıcı değişkenler ile (tanımlanan ağırlık değerleri ile ağırlıklandırılmış) regresyonundan en küçük kareler kestirimlerini elde etmeye dayanmaktadır. Buradan tek adımda bulunan ağırlıklı en küçük kareler kestirimleri minimum lojit ki-kare kestirimleri adını almaktadır [115].

Kısaca anlatılan bu üç tahmin yöntemi dışında kullanılan başka yöntemler de bulunmaktadır. Ancak çok özel durumlarda kullanılmaları nedeniyle, bu çalışmada değinilmemiştir. Lojistik modellerde parametrelerin tahmin edilmesinden sonra modeldeki katsayıların anlamlılığının ölçülmesi işlemine geçilir.

3.4.4. Modeldeki Katsayıların Anlamlılık Testi ve Yorumlanması

Modelin verilere uyumunun belirlenmesindeki önemli adımlardan biri, uyumun iyiliği diğer bir deyişle, modelin gözlenen verileri ne kadar iyi tanımlanabildiğinin incelenmesidir [110]. Bağımsız değişkenlerin modele eklenmesi veya çıkarılması ile ilgili olarak yapılan analitik çalışma burada ele alınacaktır. Bu analiz ile modelde kullanılacak katsayıların önem kontrolü yapılmış olacaktır.

3.4.4.1. Olabilirlik Oran Testi

Doğrusal regresyonda anlamlılık ölçülürken *SSR* değerinin büyüklüğü üzerinde durulur. Yani; büyük değerler, bağımsız değişkenin önemli olduğunu, küçük değerler ise bu bağımsız değişkenin cevabı tahmin etmek için önemli olmadığını gösterir. Lojistik regresyonda da temel prensip aynıdır. Bağımsız değişkenin dâhil olduğu ve olmadığı modellerdeki gözlenen değerler, tahmin edilen değerlerle karşılaştırılır. Bu karşılaştırmanın yapılması log-olabilirlik fonksiyonuna dayanmaktadır ve bunun için aşağıda (3.67) eşitliğinde verilen olabilirlik fonksiyonu tanımlanır:

$$D = -2\ln\left[\frac{\text{(Modelin olabilirliği)}}{\text{(Doymuş modelin olabilirliği)}}\right] \tag{3.67}$$

Doymuş bir model, değişken sayısı kadar parametre içeren modeldir. Yukarıdaki (3.67) eşitliğinde parantez içinde yer alan ifade, olabilirlik oranı (likelihood ratio) olarak adlandırılır. Eşitlikteki "-2*ln*" değeri matematikseldir ve hipotez testinde kullanılmak üzere dağılımı bilinen bir nicelik elde edilmesi için gereklidir. *D* (deviance) istatistiği, doğrusal regresyondaki hata kareleri toplamı ile aynı rolü oynar, bu test olabilirlik oranı testi olarak adlandırılır ve test için kullanılan istatistik (3.68) eşitliğindeki gibidir:

$$D = -2\sum_{i=1}^{n}\left[y_i \ln\left(\frac{\hat{\pi}_i}{y_i}\right) + (1-y_i)\ln\left(\frac{1-\hat{\pi}_i}{1-y_i}\right)\right] \tag{3.68}$$

Bir değişkenin modeldeki etkisini ölçmek için değişken modelde yer alırken ve modelden çıkartıldığında elde edilen D değerleri arasındaki farka bakılır.

G =D (Değişken İçermeyen Model) - D (Değişken İçeren Model) =-2ln (Değişken İçermeyen Modelin Olabilirliği / Değişken İçeren Modelin Olabilirliği) (3.69)

şeklinde bulunur. Burada bulunan G değeri Ki-kare dağılımına uymaktadır [116].

3.4.4.2. Wald ve Score Testi

Wald testi, β_1 eğim parametresinin en çok olabilirlik kestirimlerinin karşılaştırılması sonucu elde edilir ve Eşitlik 3.70'de ifade edildiği gibi β_1 parametresi ile standart hatasının oranından oluşur.

$$W = \frac{\hat{\beta}_1}{\widehat{SE}(\hat{\beta}_1)} \tag{3.70}$$

Wald istatistiğinin dezavantajı büyük β_1 değerleri için standart hataların tahminin arttırılmasıdır. Bu durum, H_0 hipotezi yanlış iken reddedilmesi konusunda yanılgı oluşmasına neden olur. LR ve Wald testlerinin her ikisi de β_1 için maksimum olabilirlik kestiriminin hesaplanmasına gerek duymaktadır [110].

Wald testi sonucu bulunan *W* değeri *z* dağılımı göstermekte ve standart normal dağılıma ait tablo değeri ile karşılaştırılmaktadır. Wald test istatistiği değerleri kullanılarak modeldeki bağımsız değişkenlerin ayrı ayrı anlamlı olup olmadıkları belirlenmektedir. Büyük 1 β değerleri için standart hataların tahmininin arttırılması Wald istatistiğinin dezavantajıdır. Bu durum 0 *H* hipotezi yanlış iken reddedilmesi konusunda yanılgıya neden olmaktadır. Hem olabilirlik oran testi *G*, hem de Wald testi *W*, 1 β için en çok olabilirlik tahmininin hesaplanmasına gerek duymaktadır.

Score testi ise, bu şekilde hesaplamalar gerektirmeyen bir yöntemdir. Ancak bu testin hazır paket programlarda fazla bulunmaması kullanılma sıklığını kısıtlamaktadır. Log-olabilirliklerinin türevlerinin dağılım teorilerine dayanan Score testi genel olarak, matris hesaplamalarını gerektiren çok değişkenli bir testtir [110].

Score test için test istatistiği (*ST*):

$$ST = \frac{\sum_{i=1}^{n} x_i(y_i - \bar{y})}{\sqrt{\bar{y}(1-\bar{y})\sum_{i=1}^{n}(x_i - \bar{x})^2}} \tag{3.71}$$

ile ifade edilmektedir.

3.4.4.3. Pearson Ki-Kare Testi

Karl Pearson tarafından 1900 yılında bulunan ve değişik kullanım amaçları olmasına karşılık, var olan veya olması gereken frekanslar arasındaki farklılıkların anlamlılığının test edilmesi temeline dayanan bir başka testte Pearson ki-kare testidir [117].

$$r_i = \frac{y_i - \pi(x_i)}{\sqrt{\pi(x_i)(1-\pi(x_i))}} \tag{3.72}$$

formülü ile hesaplanan bu istatistiğinin değerinin büyük olması yani anlamlı çıkmaması modelin verilere uyumunun başarısız olduğunu göstermektedir. Ki-kare dağılımına uyduğu ifade edilse de bazı şartlar sağlanmadıkça tam bir uyum ölçütü olarak bu istatistiğin kullanılamayacağı düşünülmektedir [118]. İstatistiğin ki-kare dağılımına uyması için Koehler ve Larntz'ın

- Toplam gözlem sayısının $n \geq 10$
- Sınıf sayısının $c \geq 3$
- Beklenen değerlerin hepsinin $E \geq 0{,}25$

şeklinde önerdiği koşulların sağlanması veya bir başka yol olarak çok sayıda küçük beklenen değerlerin bir araya getirilmesi gerektiğini belirtmiştir.

3.4.5. Multinomial Lojit Model

Multinomial lojit modeller, ikiden fazla kategorili bir bağımlı değişken ile bağımsız değişkenler arasındaki ilişkiyi göstermek amacıyla kullanılır. Bu model, ikili lojit modelin; bağımlı değişken ikiden fazla düzeyli kategoriden oluştuğu duruma genişletilmiş halidir.

Multinomial lojit modellerin pek çok alanda uygulaması vardır. Pryanishnikov'un çalışmasında Avusturya'da işçi sınıfının endüstride hangi branşta çalışmayı tercih ettiğini araştırmak için multinomial lojit model kullanılmıştır [119]. Yine Lo ve

Lam'ın çalışmasında sürücülerin güzergâh tercihi multinomial lojit modeller ile araştırılmıştır [120]. Medina ve Ward'ın çalışmasında ise tüketicilerin et alırken hangi perakende pazarı tercih ettikleri multinomial lojit modeller ile belirlenmiştir [121]. Yine ekonomide yapılan başka bir uygulama ise Mesak ve Means'in reklem ve rekabet için denge çözümlemesinde multinomial lojit pazar payı modellerinin uygulanabilirliğini inceledikleri çalışmadır [122]. Abe ve Boztuğ, çalışmalarında marka tercihini etkileyen faktörleri incelemede multinomial lojit modeli kullanmışlar ve bu modeli başka yöntemlerle karşılaştırmışlardır [123]. Fujimoto, Japon işçi kadınları ile ilgili bir çalışmada oransal *odds* modeli, oransa olmayan *odds* modeli ve multinomial lojit modeli karşılaştırmıştır [124].

Sosyal sağlık alanında yapılan uygulamalar da bulunmaktadır. Van Campen ve Woittiez, çalışmalarında bireylerin çeşitli özelliklerine göre sosyal sağlık desteğine ihtiyaç duymaları durumunu multinomial lojit modeller ile incelemiştir [125]. Bazen bir uygulamayı Porell ve Miltiades değişik bölgelerde yaşayan insanların, vücutlarındaki hareket kısıtlılığı açısından farklı olup olmadıklarını değerlendirmek amacıyla yapmışlardır [126]. Bir başka çalışmayı da Zweig ve Lindberg; ergenlik çağındaki çocukların risk profillerini inceleyerek gerçekleştirmişlerdir [127]. Liu ve arkadaşları ise körfez savaşı sırasında orduda görev yapan hastaların gösterdikleri fiziksel semptomlar ile sosyodemografik özellikler arasındaki ilişkiyi açıklamak için multinomial lojit modelleri kullanmışlardır.

3.4.5.1. Multinomial Lojit Model Çözümlemesi

Y bağımlı değişkeninin J kategorili bir nominal değişken olduğu varsayılsın (sıralama rastgele olarak). $\{ \pi_1, \dots, \pi_j \}$ise yanıt olasılıklarını göstersin. ($\sum_j \pi_j = 1$ olmak üzere). Bu olasılıklara bağlı olarak n tane bağımsız gözlem alındığında, J kategorinin hepsinde ortaya çıkan gözlem sayısı multinomial dağılım gösterir. Bir multinomial lojit model J-1 lojiti aynı anda oluşturur ve bağımlı değişkenin J-1 kategorisini, referans (baseline) kategorisi ile karşılaştırır. Bu kategorinin belirlenmesinde herhangi bir kısıt yoktur. Kullanıcı referans kategorisini kendi amaçları doğrultusunda belirleyebilir. Doğal bir referans kategorisi olmadığında;

yanıt kategorileri içinde prevalansı en yüksek kategoriyi referans kategori olarak belirlemek uygun olur. İstatistik paket programları genellikle son kategoriyi referans kategorisi olarak kullanırlar. Buna göre; eğer Y, J kategorili bir yanıt değişkeni ise, lojit oluşturabilecek;

$$\binom{J}{2} = \frac{J(J-1)}{2} \tag{3.73}$$

tane yanıt çifti vardır. Bunların J-1 tanesi dışında kalanlar gereksiz çiftlerdir. Multinomial lojit modeller tüm kategori çiftlerini temsil eder ve bir kategorideki cevaba göre diğerinin oddsunu tanımlar.

Yanıt olasılıklarının terimleri ile genel lojit model;

$$\log\left(\frac{\pi_{ij}}{\pi_{i1}}\right) = x_i\beta_j \qquad \begin{matrix} \text{j: 2,…,} \\ \text{i: 1,…,N} \end{matrix} \tag{3.74}$$

şeklinde tanımlanır. Burada $\pi_{ij}; P(Y = j|x)$ olması olasılığıdır. Yani bu olasılık;

$$\pi_{ij} = \frac{\exp\{x_i\beta_j\}}{\sum_{j=1}^{J}\exp\{x_i\beta_j\}} \tag{3.75}$$

şeklindedir. Bu modelde,

$$P_i(Y = 0) = \frac{1}{1+\sum_{j=1}^{J}\exp\{x_i\beta_j\}} \tag{3.76}$$

$$P_i(Y = j) = \frac{\exp\{x_i\beta_j\}}{1+\sum_{j=1}^{J}\exp\{x_i\beta_j\}} \tag{3.77}$$

$$P_i(Y = \text{J}) = \frac{\exp\{x_i\beta_j\}}{1+\sum_{j=1}^{J}\exp\{x_i\beta_j\}} \tag{3.78}$$

olur. Burada j:2,…,J-1 ve i:1,…,N'dir [9].

Lojitleri oluşturabilmek için bir yol her bir yanıt kategorisi ile referans kategorisini eşleştirmektir. Son kategori referans kategorisi olduğunda, iki açıklayıcı değişkeni A ve B olan bir model için j inci lojit şu şekilde yazılabilir:

$$\log\left(\frac{\pi_{j|h_i}}{\pi_{J|h_i}}\right) = \log\left(\frac{m_{hij}}{m_{hiJ}}\right) = \alpha_j + \beta_{hj}^{A} + \beta_{ij}^{B} \quad (j= 1,2,\ldots, J-1) \tag{3.79}$$

Burada; $\pi_{j|h_i}$, A'nın h'inci ve B'nin i'inci düzeyinde j yanıtının ortaya çıkma olasılığını, α_j sabit terimi ve β_j ise regresyon katsayılarını göstermektedir.

Multinomial lojit modeller ortak değişken kombinasyonlarının her birinde yanıt sayılarının multinomial olduğunu ve farklı ortak değişken kombinasyonlarındaki multinomial sayıların bağımsız olduğunu varsayar.

Multinomial lojit model kullanmanın bir faydası; her bir kategorinin oddsunu ortak değişkenlerin bir fonksiyonu olarak; bir referans kategori ile ilişkilendirerek modellemesidir. Multinomial lojit modeller, katsayıların eşitliğini karıştırıcı etkileri farklı olsa bile test edebilir [124].

Lojit daha önce de belirtildiği gibi; bağımlı değişkenin bir kategorisinin diğerine göre oddsunun logaritması demektir. Yanıt değişkeni lojit olduğu için, lojit model parametrelerinin yorumlanışı da farklıdır. İki bağımsız değişkeni x_1 ve x_2 olan, yanıt değişkeni ise y ile gösterilen bir lojit model şu şekildedir:

$$logO_2^y = \alpha + \beta_i x_1 + \beta_k x_2 + \beta_{ik} x_1 x_2 \tag{3.80}$$

Burada α (kesişim değeri); kestiricilerin bütün *ik* değerlerindeki lojitin ortalama değeridir. β parametreleri ise, değişkenin ya da değişken kombinasyonlarının bağımlı değişkenin log oddsu üzerindeki etkisini gösterir. Ana etkiler tek bir değişken indisi ile ($\beta_i x_1$ gibi) , birinci derece etkileşimler ise ikili indis ile ($\beta_{ik} x_1 x_2$ gibi) gösterilir.

Standart bir lojit modelde lojitlerin kestiriminde kullanılan bir ortak değişken kümesi vardır. Burada π, değişkenin belirli bir değeri alma olasılığıdır.

$$\ln\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots \quad (3.81)$$

olarak gösterilebilir. β_1, x_1 'deki bir birimlik değişime karşı, π/(1-π) 'in lojitindeki değişimi gösterir.

Multinomial lojit modelde ise $\ln(\pi_2/\pi_1)$, $\ln(\pi_3/\pi_1)$,…'leri eşzamanlı hesaplayan bir ortak değişken kümesi vardır. π_1, π_2, π_3 bütün olası sonuç kategorileri göstermektedir. π_1 ise referans kategorisidir. Bu durum; eşzamanlı uyum yaklaşımı olarak adlandırılır [128].

Ana etkilerin ve etkileşimlerin etkisinin değeri, bağımlı değişkenin lojitinin artması ya da azalması şeklinde yorumlanır. Diğer bir deyişle; tahmin edici ile (ya da etkileşim durumunda tahmin ediciler kombinasyonu) bağımlı değişkenin lojiti arasında ilişki olmaması durumunun (bu durumda söz konusu değer 0 olur) altında ya da üstünde olmasına göre yorumlanır.

Değişkenlerin herhangi bir düzeyindeki odds değerini bulmak için, denklemde buna ilişkin değerler yerine konarak işlem yapılır ve çıkan değerin üsteli alınır. Bu değer tahmin edilen odds değeri olur. Değişkenin iki düzeyi arasındaki odds oranı için ise bu iki değere ilişkin odds değeri birbirine bölünür.

BÖLÜM 4

UYGULAMA

Bu çalışmada veriler M.E.B. Bilgi İşlem Grup Başkanlığı, e-okul sistemi veritabanından alınmıştır. İlköğretim 8.sınıf öğrencilerinden rastgele seçilen 2008 yılına ait 25000 kayıt üzerinde işlem yapılmıştır. Öğrencilerin SBS sınavında sorusu bulunan Türkçe, Matematik, Fen ve Teknoloji, Yabancı Dil, Sosyal Bilgiler, Din Kültürü ve Ahlak Bilgisi ve T.C. İnkılâp Tarihi ve Atatürkçülük derslerine ait notları ile SBS sınav puanları kullanılmıştır. Yine öğrencinin sınavdaki başarısına etki edeceği düşünülen okul türü (devlet okulu/özel okul) ile öğrenciye ait bazı kişisel bilgiler (cinsiyet, anne ve babasının medeni durumu, eğitim durumu, kardeş sayısı, kendi odası olup olmadığı, okul dışında çalışıp çalışmadığı, burs alıp almadığı ve özel ders alıp almadığı) de kullanılmıştır.

4.1. VERİ

Veriler tek bir kaynaktan elde edildiği için birleştirme aşamasına gerek duyulmamıştır. Veri temizleme aşamasında içerdiği veri miktarı açısından %1'lik bir doluluk oranına bile sahip olmayan öğrencilerin özel durum bilgilerinin tutulduğu değişken uygulamadan çıkarılmıştır. Öğrenciye ait cinsiyet, bursluluk, okul türü, yılsonu başarı puanları, özel ders alıp almadığı veya özel dershaneye gidip gitmediği bilgileri ile sınav puanları alanlarında herhangi bir eksik veri gözlemlenmemiştir. Bunların dışında kalan kayıtlarda veri eksiklikleri gözlenmiş fakat bu eksikliklerin toplam kayıta oranının %1 veya %2 olması nedeniyle eksiklik bulunan kayıtların silinmesi uygun görülmüştür. Toplamda 3217 kayıt silinerek 21783 kayıt üzerinde uygulama çalıştırılmıştır. Öğrencilere ait ders notları veri madenciliği uygulamasında kullanılırken 100'lük sistemden 5'lik sisteme dönüştürülmüştür (Çizelge 4.1).

Çizelge 4.1. Ders notları puanlama sistemi.

Puan Sistemi	Kullanılan Değer
85-100	5
70-84	4
55-69	3
45-54	2
0-44	1

Çıkış değişkeni olarak kullanılacak OGES puanları incelendiğinde öğrencilerin almış oldukları puanların 120 ile 497 arasında olduğu görülmüştür ve bu puanlar Çizelge 4.2'deki hale dönüştürülmüştür.

Çizelge 4.2. OGES puanları.

OGES puanları	Kullanılan Değer
450-497	ÇOK İYİ
400-449	İYİ
300-399	ORTA
200-299	KÖTÜ
120-199	ÇOK KÖTÜ

Öğrencilerin kardeş sayıları 1 ile 23 arasında değişmektedir. Bu nedenle uygulamada Çizelge 4.3'de gösterildiği gibi kullanılmaktadır.

Çizelge 4.3. Kardeş sayıları.

Kardeş Sayıları	Kullanılan Değer
1	1
2	2
3	3
4	4
5	5
[6, 23]	6 ve daha fazla

Uygulamada kullanılan değişkenler ve bu değişkenlere ait açıklamalar Çizelge 4.4'de görüldüğü gibidir.

Çizelge 4.4. Uygulamada kullanılan bağımsız değişkenlerin listesi.

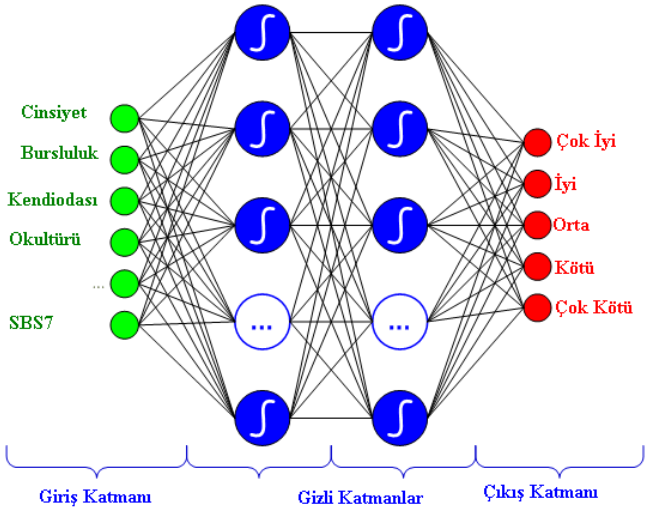
ALAN ADI	**TİPİ**	**AÇIKLAMASI**
CINSIYET	Metin	Öğrencilerin cinsiyeti
ANNEDURUM	Metin	Annesinin yaşayıp yaşamadığı
BABADURUM	Metin	Babasının yaşayıp yaşamadığı
BIRLIKTEMI	Metin	Anne ve babasının birlikte olup olmadığı
CALISIYORMU	Metin	Öğrencilerin okul dışında çalışıp çalışmadığı
ANNEMESLEK	Metin	Annesinin mesleği
BABAMESLEK	Metin	Babasının mesleği
ANNEEGITIM	Metin	Annesinin eğitim durumu
BABAEGITIM	Metin	Babasının eğitim durumu
KARDESSAYISI	Sayı	Kardeş sayısı
KENDIODASI	Metin	Kendi odası olup olmadığı
OZELDERS	Metin	Özel ders alıp almadığı
OZELDERSHANE	Metin	Özel dershaneye gidip gitmediği
BURSLUMU	Metin	Burs alıp almadığı
KURUMTURU	Metin	Okul türü
OZELEGITIM	Metin	Özel eğitim ihtiyacı olup olmadığı
YSB6	Sayı	6. sınıfa ait yılsonu başarı puanı
YSB7	Sayı	7. sınıfa ait yılsonu başarı puanı
YSB8	Sayı	8. sınıfa ait yılsonu başarı puanı
TURKORT	Sayı	6,7 ve 8. sınıfa ait Türkçe dersi ortalamaları
MATORT	Sayı	6,7 ve 8. sınıfa ait Matematik dersi ortalamaları
FENORT	Sayı	6,7 ve 8. sınıfa ait Fen ve Teknoloji dersi ortalamaları
YADORT	Sayı	6,7 ve 8. sınıfa ait Yabancı Dil dersi ortalamaları
DINORT	Sayı	6,7 ve 8. sınıfa ait Din dersi ortalamaları
SOSORT	Sayı	6,7 ve 8. sınıfa ait Sosyal Bilgiler dersi ortalamaları
SBS6	Sayı	6. sınıf SBS puanı
SBS7	Sayı	7. sınıf SBS puanı

4.2. YAPAY SİNİR AĞI UYGULAMASI

Uygulamada amaç ileri dönük OGES puanlarının tahmini olduğu için çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağları kullanılmıştır. Bu nedenle ileri beslemeli ağ modeli kullanılmıştır. Ağa eğitim sırasında hem girdiler hem de üretmesi gereken sonuçlar gösterilmiştir. (Danışmanlı öğrenme).

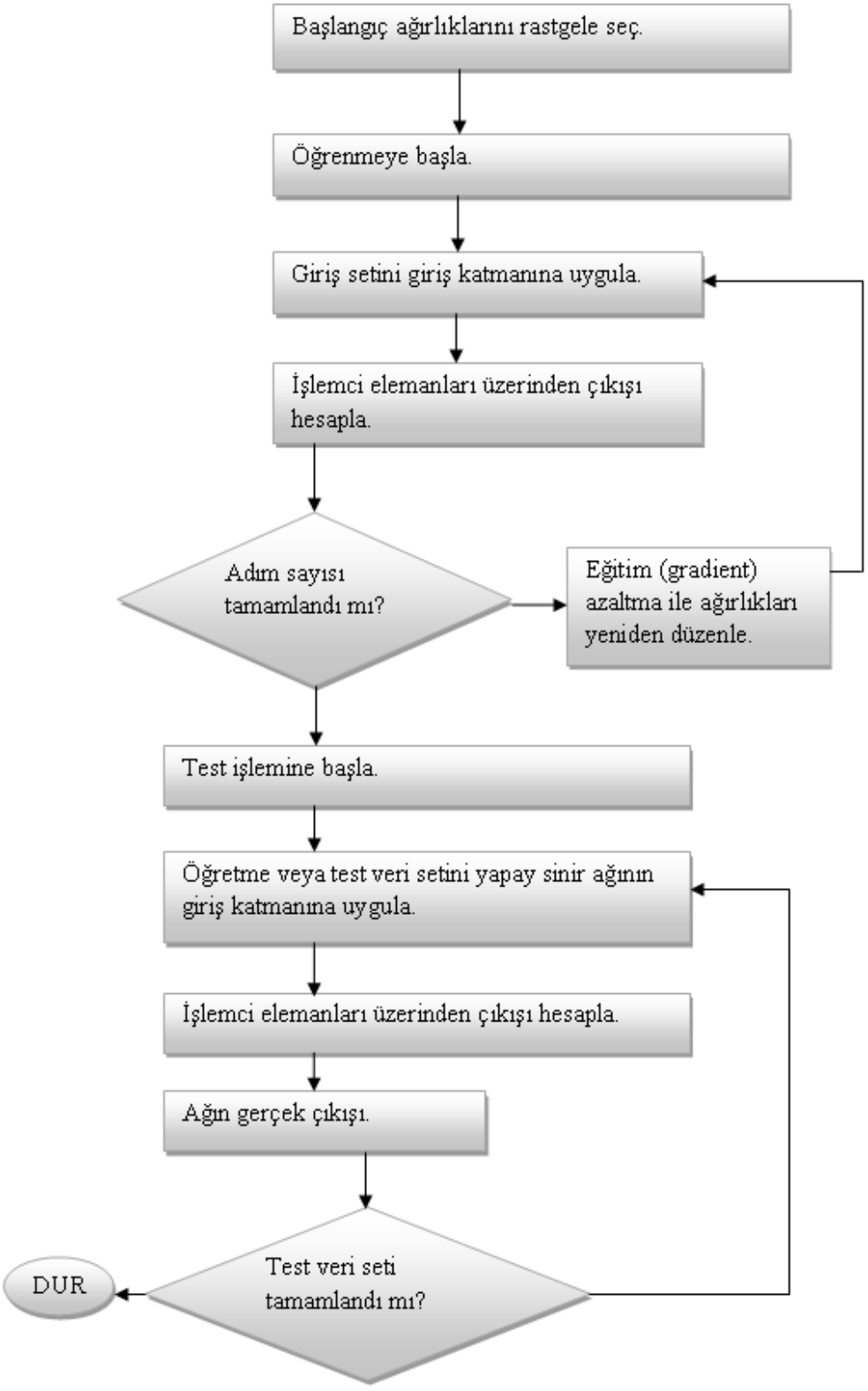
4.2.1. Çok Katmanlı Algılayıcı

Çok katmanlı algılayıcı ile yapılan eğitim denemelerinde hem sigmoid hem de tanjant hiperbolik aktivasyon fonksiyonları kullanılmış ve en iyi sonuç veren fonksiyon eğitimde kullanılmıştır. Eğitim tamamlandıktan sonra test setinde olan veya olmayan tüm sorgulamalarda ağ üzerindeki ağırlıklarda herhangi bir değişiklik olmaz. Bu model, öğrenme ve aktivasyon fonksiyonlarına ait detaylı bilgiler Bölüm 3'te verilmiştir. Çok katmanlı algılayıcıda ağın ilk ağırlık değerleri (w), deneme yolu ile [-1,1] aralığında rastgele verilmiştir. Ağırlığın rastgele verilmesi nedeniyle aynı özelliklere sahip yapay sinir ağlarının farklı ama yakın hata sonuçları ürettiği gözlemlenmiştir. Uygulamada çok katmanlı algılayıcının giriş katmanında 27 adet giriş sütunu olması nedeniyle 27 nöron bulunmaktadır. Gizli katman nöron sayısı 20-30 arasında seçilmiştir. 5 adet çıkış sütunu olması nedeniyle çıkış katmanında 5 nöron bulunmaktadır (Şekil 4.1).



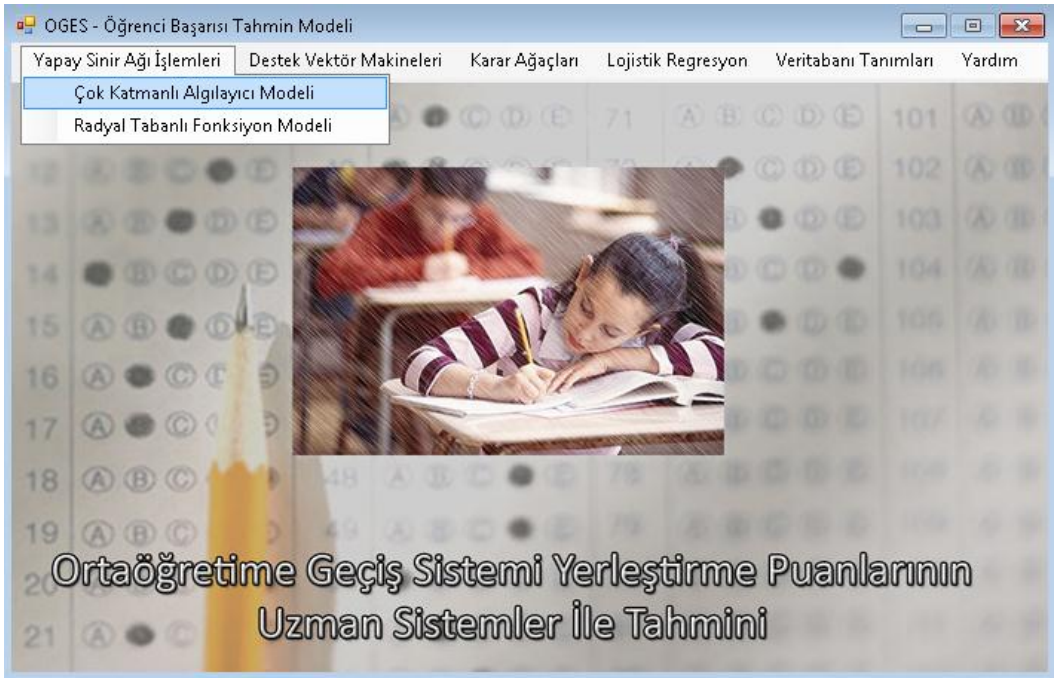
Şekil 4.1. Çok katmanlı algılayıcı yapay sinir ağı mimarisi.

Uygulamada kullanılan ileri beslemeli geri yayılım algoritmasına ait akış diyagramı Şekil 4.2'deki gibidir.



Şekil 4.2. Geri yayılım algoritması akış diyagramı.

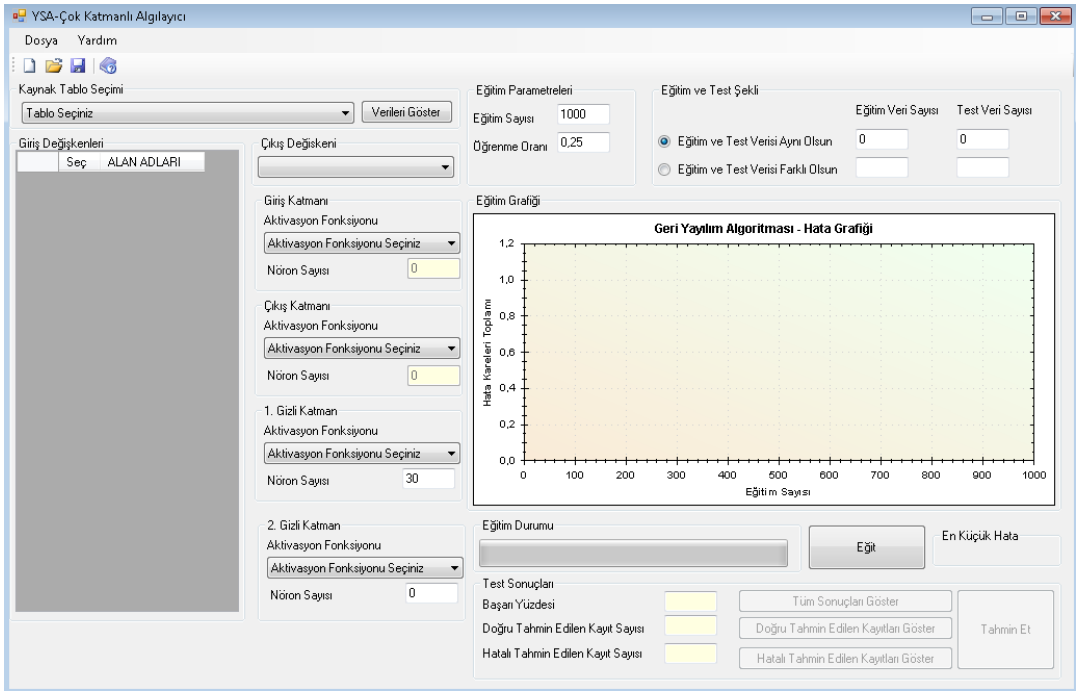
Çalışmada kullanılan yazılım Visual Studio 2010 platformunda C# dilinde hazırlanmıştır. Eğitimde ve test aşamasında kullanılacak veriler ve eğitim sonuçları Oracle veritabanında tutulmaktadır. Oluşturulan yazılımın giriş ekranı Şekil 4.3'deki gibidir. Burada Yapay Sinir Ağı işlemleri menüsü altında bulunan Çok katmanlı Algılayıcı Modeli seçeneği tıklanarak ağ oluşturma ekranına geçilir.



Şekil 4.3. OGES- Öğrenci başarısı tahmin modeli giriş ekranı.

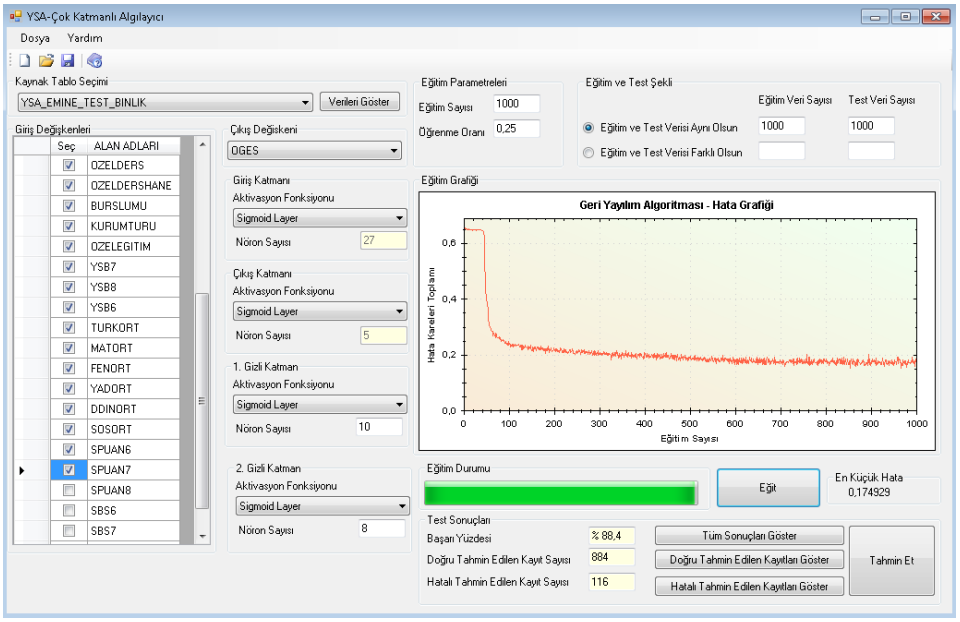
Gelen ekranda Oracle veritabanında bulunan tablolar Kaynak Tablo Seçimi alanında listelenir. Açılır listeden uygulamada kullanılacak verilerin saklandığı tablo adı seçildikten sonra giriş değişkenleri ve çıkış değişkeni başlığının altındaki listeler otomatik olarak dolar. Daha sonra bu ekranda gelen giriş değişkenleri ve çıkış değişkeni kullanıcı tarafından seçilir, giriş değişkenleri birden fazla olabileceği için alanların sol tarafında bulunan kutucuklar işaretlenerek birden çok seçim yapılabilir. Seçilen tablo içerisindeki veriler Verileri Görüntüle butonu ile görüntülenebilir. Ekranda bulunan aktivasyon fonksiyonlarının da seçilmesi gerekmektedir. Aktivasyon fonksiyonları ilgili grup içerisindeki açılır kutulardan seçilerek belirlenir. Giriş ve çıkış katmanına ait nöron sayıları seçilen değişkenlere göre program tarafından otomatik olarak atandığı için değiştirilememektedir. Fakat gizli katmanlara ait nöron sayıları kullanıcı tarafından değiştirilebilir. Ayrıca modelde

kullanılacak olan gizli katman sayısı da kullanıcının seçimine bağlıdır. İstenirse iki gizli katmanda kullanılabilir. Ekranda bulunan eğitim sayısı, öğrenme oranı eğitim ve test şeklinin nasıl olacağına dair bilgilerin seçilebildiği alanlarda kullanıcı tarafından değiştirilebilecek alanlardır (Şekil 4.4).



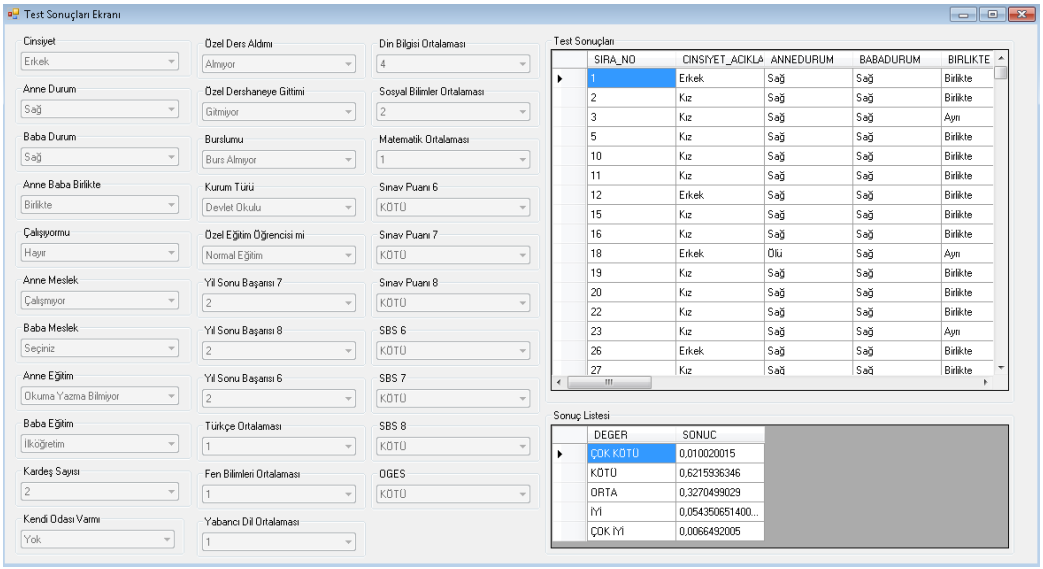
Şekil 4.4. Çok katmanlı algılayıcı yapay sinir ağı oluşturma ekran görüntüsü.

Kullanıcı tarafından istenilen parametrelerde değişiklikler gerçekleştirildikten sonra "Eğit" butonuna tıklanarak yapay sinir ağının eğitilmesi işlemine geçilir. Eğitim işleminin süresi kullanılan verinin boyutuna ve eğitim sayısına göre değişmektedir. Örneğin 5000 kayıtlık bir verinin, 1000 adımda eğitim süresi yaklaşık 10 dakika sürerken; 25000 kayıtlık bir verinin 1000 adımda eğitim süresi 45 dakika kadar sürebilmektedir. Test aşaması ise yine teste tabi tutulacak kayıt sayısı ile orantılı olarak değişmektedir. Örneğin 5000 kayıtlık bir verinin test süresi 10 dakika sürerken; 25000 kayıtlık bir verinin test süresi 50 dakika kadar sürebilmektedir. Eğitim ve test aşamaları bittikten sonra Test sonuçları başlığı altında bulunan alanlar dolmakta ve modele ait hata grafiği çizdirilmektedir. Örnek bir yapay sinir ağı modeli ve sonuçları Şekil 4.5'deki gibidir.



Şekil 4.5. Eğitim sonucunda oluşan ekran görüntüsü (Çok katmanlı algılayıcı).

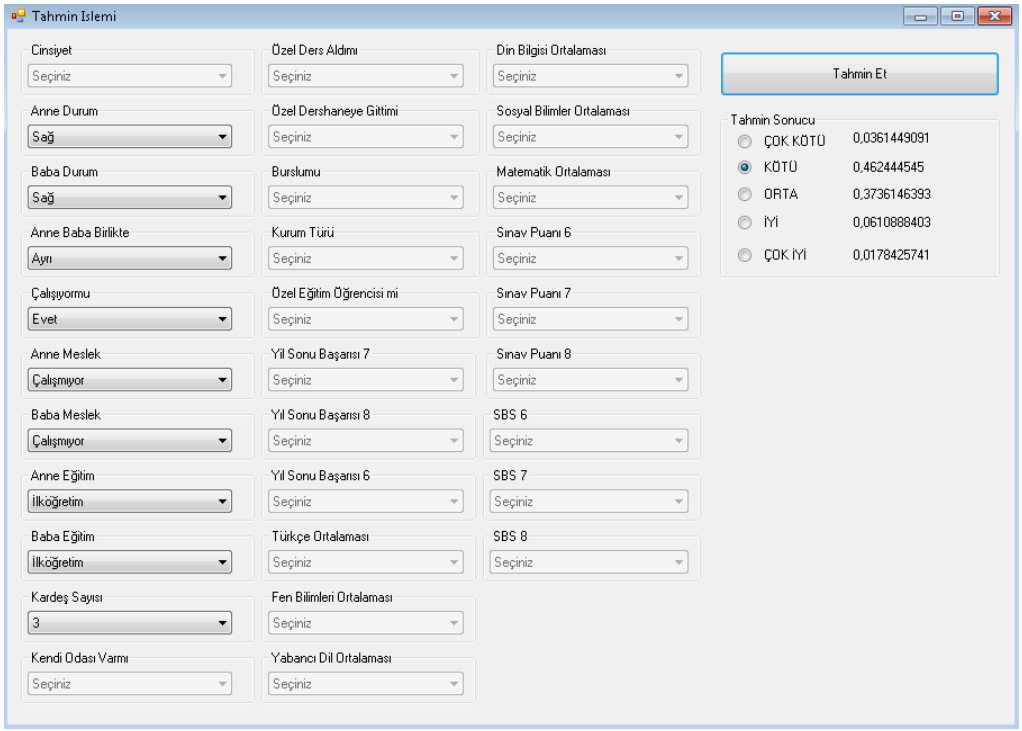
Burada Başarı yüzdesi oluşturulan yapay sinir ağı modelinin test aşamasındaki başarısının yüzdelik olarak karşılığıdır. Doğru tahmin edilen kayıt sayısı modelin doğru tahmin ettiği kayıt sayısını göstermektedir. Yanlış tahmin edilen kayıt sayısı da modelin yanlış tahmin ettiği kayıt sayısını göstermektedir. Burada "Doğru Tahmin Edilen Kayıtları Göster" butonuna tıklandığında modelin doğru tahmin ettiği kayıtların detaylı bir görüntüsü karşımıza çıkmaktadır (Şekil 4.6).



Şekil 4.6. Doğru tahmin edilen kayıtların detaylı görüntüsü.

Şekil 4.6'da doğru tahmin edilen kayıtlar için giriş değişkenlerinin ve çıkış değişkenin değerleri teker teker görüntülenebilir. "Yanlış Tahmin Edilen Kayıtları Göster" ve "Tüm Sonuçları Göster" butonlarına tıklandığında da yine aynı şekilde detaylı bilgilerin bulunduğu ekranlar gelmektedir.

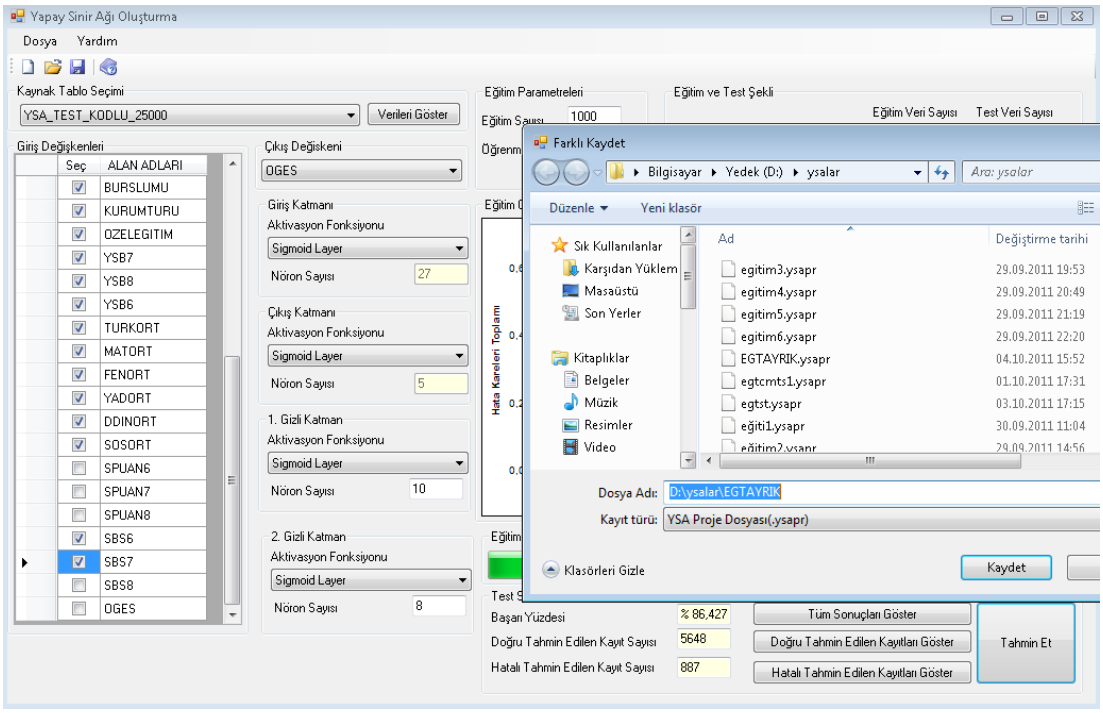
Yapay sinir ağı oluşturma ekranında "Tahmin Et" butonuna tıklandığında Şekil 4.7'deki gibi bir ekran karşımıza gelmektedir. Bu ekranda eğitim yapılan modelde kullanılan giriş değişkenleri aktif diğer alanlar ise pasif olarak gelmektedir. Bu ekranın amacı örneğin bu yıl sınava girecek bir 8. Sınıf öğrencisinin bilgilerinin girilerek OGES başarısının yaklaşık bir tahminini verebilmektir. Bu ekranda gerekli alanlar seçilerek "Tahmin Et" butonuna basıldığında Tahmin sonuçları bölümünde YSA tarafından hesaplanan çıkış nöron değerleri gösterilmekte ve en yüksek değerli kayıt seçilerek kullanıcıya sonuç olarak gösterilmektedir.



Şekil 4.7. Tahmin ekranı.

Yapay sinir ağı oluşturma ekranında yapabileceğimiz işlemlerden bir diğeri de ekranda bulunan butonu ile eğittiğimiz bir yapay sinir ağını kaydedebilmemizdir. Böylelikle saatler süren bir eğitim aşamasını tekrar tekrar yapmak zorunda kalmamış

oluruz ve oluşturduğumuz modele göre tahmin etme işlemini daha hızlı gerçekleştirebiliriz (Şekil 4.8). Yine ekranda bulunan butonu ile de daha önceden kaydedilmiş olan bir yapay sinir ağı modeli açılarak üzerinde işlem yapılabilir.



Şekil 4.8. Kaydetme ekranı.

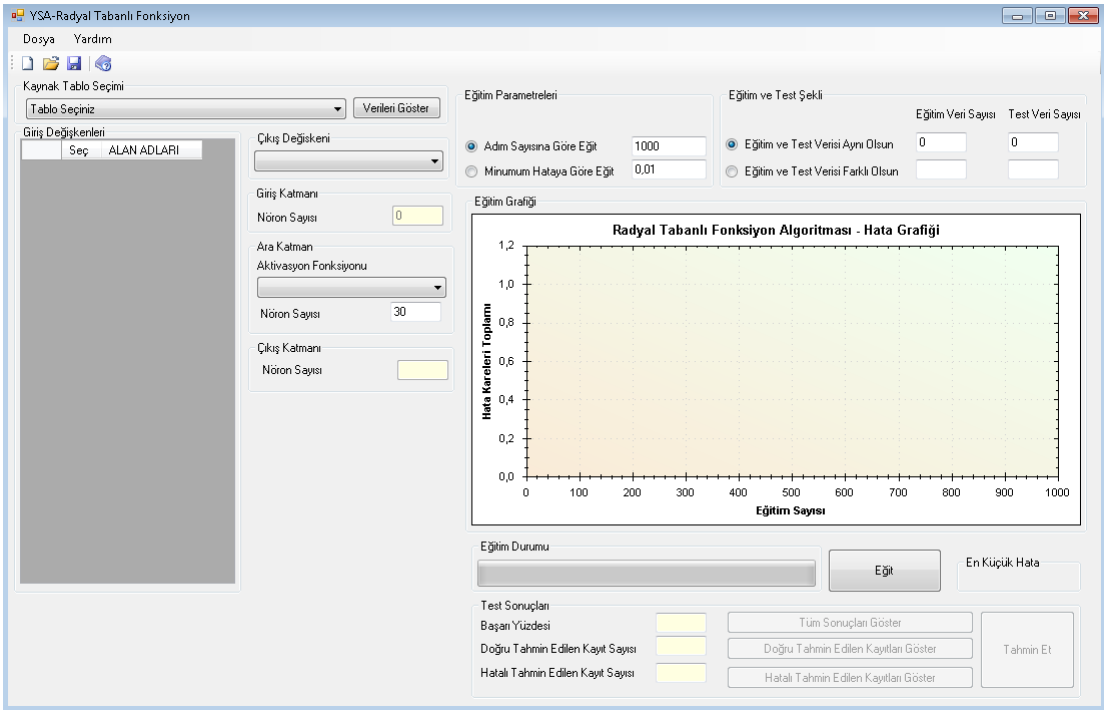
4.2.2. Radyal Tabanlı Fonksiyon

Radyal tabanlı fonksiyon ağlarında girdi katmanı sadece modele dış dünyadan veri alınmasını sağlar. Bu bağlamda girdi, hiç bir şekilde işlenmeden doğrudan girdi katmanı aracılığı ile gizli katman nöronlarına iletilir. Diğer bir ifade ile girdi katmanını gizli katmana bağlayan tüm ağırlık değerlerinin "1" olduğu ve çözüm süresince değişmediği varsayılır. Bu özelliği ile öğrenme aşamasında değeri değiştirilecek parametre sayısında önemli bir azalma gerçekleşir ve dolayısıyla öğrenme hızlanır. Bu bakımdan RTFA'nın çok katmanlı YSA'ya göre daha kullanışlı olduğu söylenebilir. Bunun nedeni, eldeki problemin çözümüne uygun RTFA'nın oluşturulması aşamasında ağ mimarisine ilişkin verilecek tek kararın, gizli katmanda bulunacak nöron sayısının belirlenmesi olmasıdır.

Herhangi bir radyal tabanlı fonksiyon, merkez (c) ve yarıçap (r) olmak üzere iki parametre ile belirlenir. Bu parametrelerden c, fonksiyonun en büyük ya da en küçük değerini aldığı noktayı gösterirken, r bu noktaya olan uzaklıkları ölçeklendiren parametredir. Literatürde yer almış birçok radyal tabanlı fonksiyon olmakla beraber bunlardan başlıcaları Gauss, Cauchy, Çoklu-Kuadratik ve Ters Çoklu-Kuadratik fonksiyonlardır. Gauss ve Çoklu-Kuadratik fonksiyonlar merkez değere yaklaştıkça en büyük değerlerine yaklaşırken, Cauchy ve Ters Çoklu-Kuadratik türü fonksiyonlarda en küçük değer, merkez noktasında alınır ve merkezden uzaklaştıkça fonksiyon daha büyük değerler alır. Gizli katman nöron sayısı RTF ağlarının mimarisinde karar verilmesi gereken en önemli parametrelerden biridir. Aşırı uyum ve yetersiz öğrenme durumlarına düşmemek gerekir.

Uygulamada radyal tabanlı fonksiyonun giriş katmanında 27 adet giriş sütunu olması nedeniyle 27 nöron bulunmaktadır. Gizli katman nöron sayısı 30-200 arasında deneme yanılma yoluyla seçilmiştir. 5 adet çıkış sütunu olması nedeniyle çıkış katmanında 5 nöron bulunmaktadır. Aktivasyon fonksiyonu olarak Gauss kullanılmıştır.

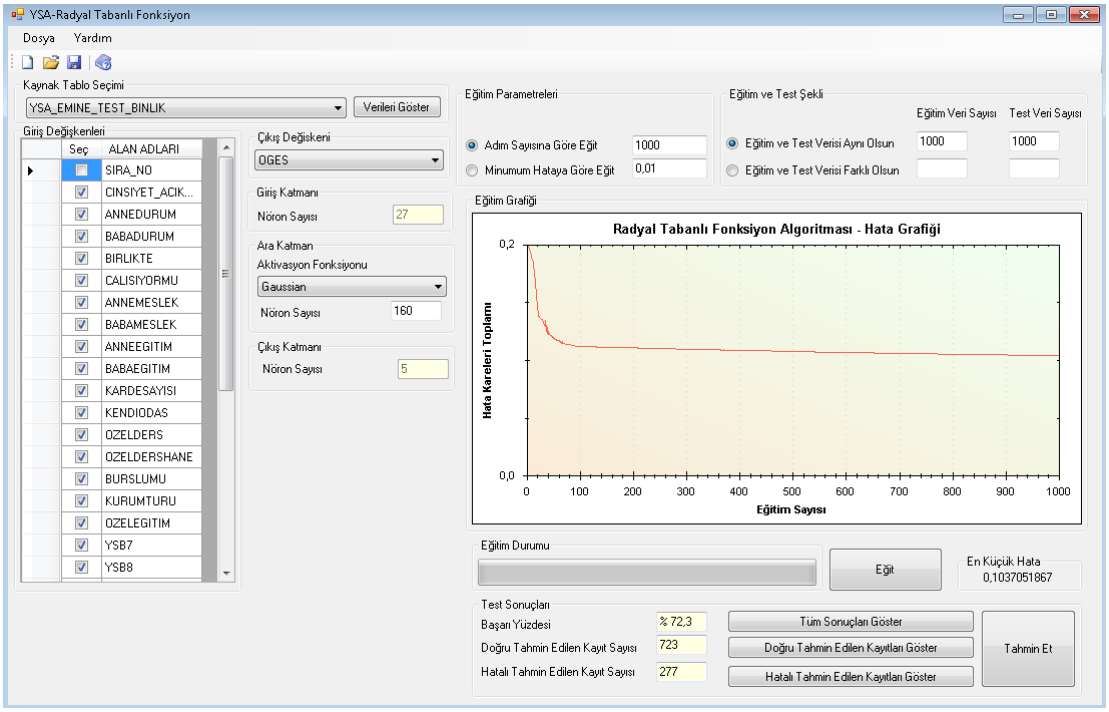
Radyal tabanlı fonksiyon ağını oluşturmak için Yapay Sinir Ağı işlemleri menüsü altında bulunan Radyal Tabanlı Fonksiyon Modeli seçeneği tıklanarak ağ oluşturma ekranına geçilir. Gelen ekranda Oracle veritabanında bulunan tablolar Kaynak Tablo Seçimi alanında listelenir. Açılır listeden uygulamada kullanılacak verilerin bulunduğu tablo adı seçilir. Daha sonra bu ekranda gelen giriş değişkenleri ve çıkış değişkeni kullanıcı tarafından seçilir. Seçilen tablo içerisindeki veriler Verileri Görüntüle butonu ile görüntülenebilir. Ekranda bulunan ara katmana ait aktivasyon fonksiyonlarının da seçilmesi gerekmektedir. Aktivasyon fonksiyonları ilgili grup içerisindeki açılır kutulardan seçilerek belirlenir. Giriş ve çıkış katmanına ait nöron sayıları seçilen değişkenlere göre program tarafından otomatik olarak atandığı için değiştirilememektedir. Fakat gizli katmana ait nöron sayıları kullanıcı tarafından değiştirilebilir. Ekranda bulunan eğitim sayısı, eğitim ve test şeklinin nasıl olacağına dair bilgilerin seçilebildiği alanlarda kullanıcı tarafından değiştirilebilecek alanlardır (Şekil 4.9).



Şekil 4.9. Radyal tabanlı fonksiyon ağı oluşturma ekranı.

Kullanıcı tarafından istenilen parametrelerde değişiklikler gerçekleştirildikten sonra "Eğit" butonuna tıklanarak yapay sinir ağının eğitilmesi işlemine geçilir. Eğitim işleminin süresi kullanılan verinin boyutuna ve eğitim sayısına göre değişmektedir. Örneğin 5000 kayıtlık bir verinin, 1000 adımda eğitim süresi yaklaşık 1-2 dakika sürerken; 25000 kayıtlık bir verinin 1000 adımda eğitim süresi 10 dakika kadar sürebilmektedir. Test aşaması ise yine teste tabi tutulacak kayıt sayısı ile orantılı olarak değişmektedir. Örneğin 5000 kayıtlık bir verinin test süresi 1-2 dakika sürerken; 25000 kayıtlık bir verinin test süresi 10-12 dakika kadar sürebilmektedir. Eğitim ve test aşamaları bittikten sonra Test sonuçları başlığı altında bulunan alanlar dolmakta ve modele ait hata grafiği çizdirilmektedir. Örnek bir yapay sinir ağı modeli ve sonuçları Şekil 4.10'daki gibidir.

Diğer tüm özellikler çok katmanlı algılayıcı modelinde olduğu gibi radyal tabanlı fonksiyon ağı için de bulunmaktadır.

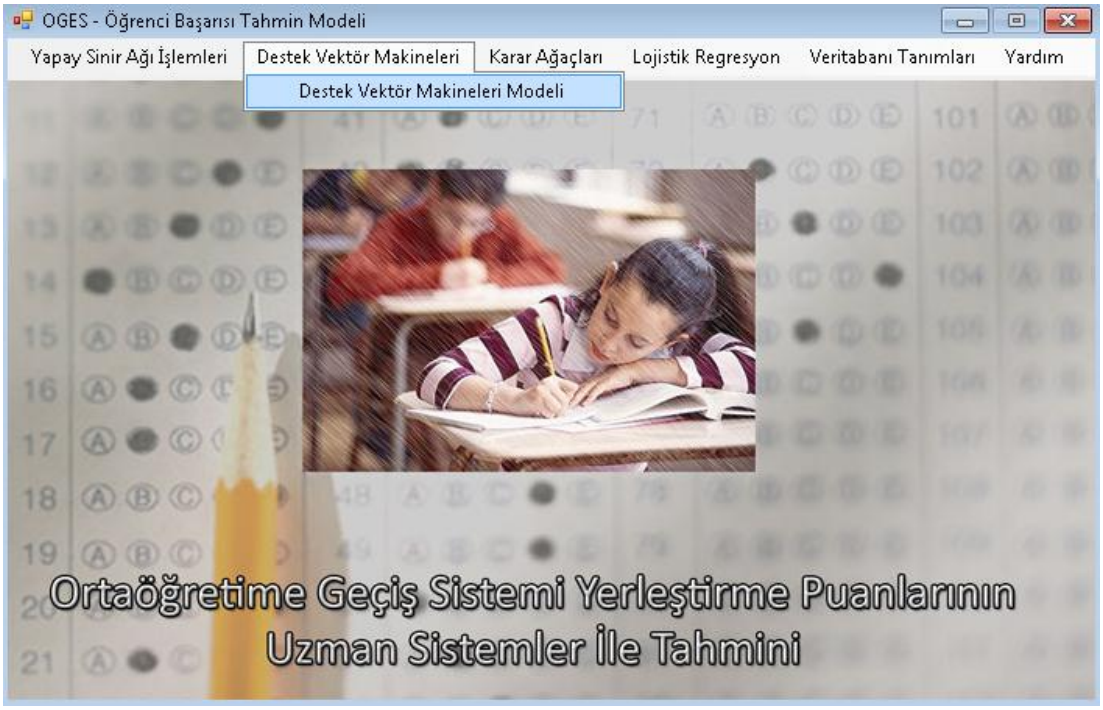


Şekil 4.10. Eğitim ve test işlemi sonucunda oluşan ekran görüntüsü (RTF).

4.3. DESTEK VEKTÖR MAKİNESİ UYGULAMASI

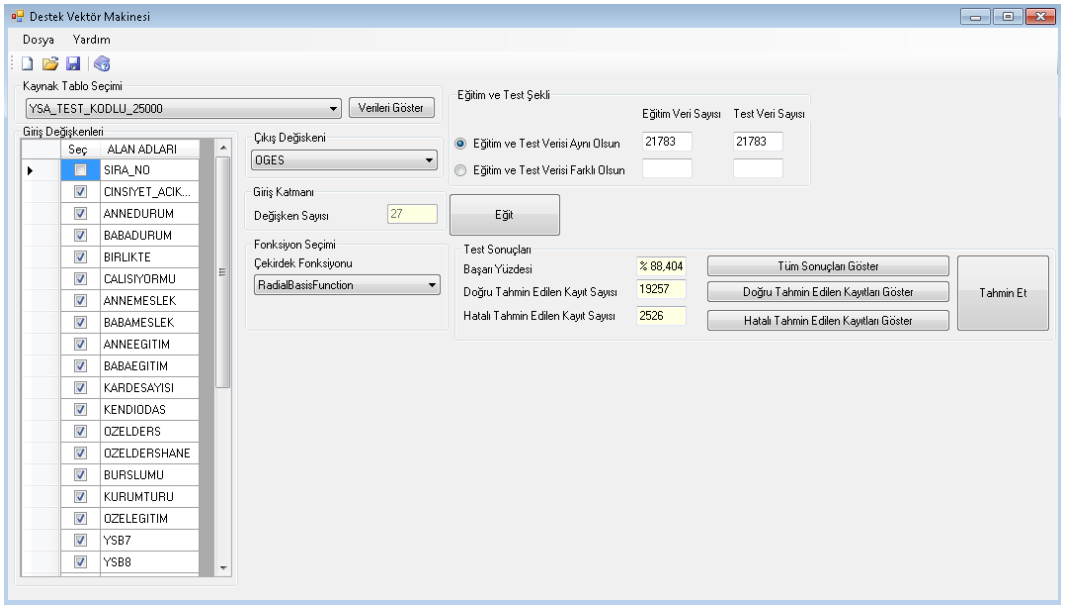
DVM sınıflandırıcılar iki sınıf etiketli durumlar düşünülerek eğitilirler. Çünkü DVM iki sınıfı birbirinden doğrudan ayırabilen bir ayırıcı fonksiyonun araştırılması üzerine kurulmuş olan bir yöntemdir. Bu çalışmada da çıkış iki sınıflı olmadığı için çoklu DVM çözümlerinden yararlanılmıştır. Literatürde iki farklı yaklaşım bulunmaktadır. Bunlardan birincisi bire bir diğeri ise bire karşı hepsi mantığıyla işlem yapmaktadır. Uygulamada bire bir DVM çözümü kullanılmıştır. Veri setinin çıkışı 5 sınıflı olduğu için k*(k-1)/2 formülünden 10 farklı DVM eğitilmiştir. Her eğitim aşamasında sadece iki farklı sınıf verisi alınmıştır. Çekirdek fonksiyon polinomial fonksiyon ve hata 0.001 alındığında en iyi sonuç elde edilmiştir.

Destek Vektör Makineleri ile sınıflandırma uygulamasını başlatmak için Destek Vektör Makineleri menüsü altında bulunan Destek Vektör Makineleri Modeli seçeneği tıklanır (Şekil 4.11).



Şekil 4.11. Destek vektör makineleri modeli giriş ekranı.

Gelen ekranda Oracle veritabanında bulunan tablolar Kaynak Tablo Seçimi alanında listelenir. Açılır listeden uygulamada kullanılacak verilerin bulunduğu tablo adı seçilir. Daha sonra bu ekranda gelen giriş değişkenleri ve çıkış değişkeni kullanıcı tarafından seçilir. Seçilen tablo içerisindeki veriler Verileri Görüntüle butonu ile görüntülenebilir. Ekranda bulunan ara katmana ait aktivasyon fonksiyonlarının da seçilmesi gerekmektedir. Çekirdek fonksiyonu ilgili grup içerisindeki açılır kutulardan seçilerek belirlenir. Giriş ve çıkış katmanına ait eleman sayıları seçilen değişkenlere göre program tarafından otomatik olarak atandığı için değiştirilememektedir. Ekranda bulunan test şeklinin nasıl olacağına dair bilgiler kullanıcı tarafından değiştirilebilecek alanlardır (Şekil 4.12).



Şekil 4.12. Eğitim ve test sonucunda oluşan ekran görüntüsü (DVM).

Diğer tüm özellikler yapay sinir ağı modellerinde olduğu gibi destek vektör makineleri modeli için de bulunmaktadır.

4.4. KARAR AĞAÇLARI UYGULAMASI

Karar ağaçları uygulamasında Quinlan tarafından geliştirilen ve günümüzde en çok kullanılan C4.5 algoritması kullanılmıştır. C4.5 Algoritması ID3 algoritmasının bütün özelliklerini taşıyan, daha gelişmiş bir algoritmadır. ID3 algoritması karar ağacının her düğümü için bilgi kazancı değerlerini hesaplar. C4.5 ise bilgi kazancı ile beraber alt sette yer alan karakteristiklerin bilgi kazanç oranlarını da hesaplayarak bilgi kazanç oranı en yüksek olan karakteristiği düğüm noktası olarak seçer. Karar ağacının her dalı sadece bir tek sınıfa karşılık gelinceye kadar işlemleri sürdürür. Daha sonra karar ağacını kural setine dönüştürür.

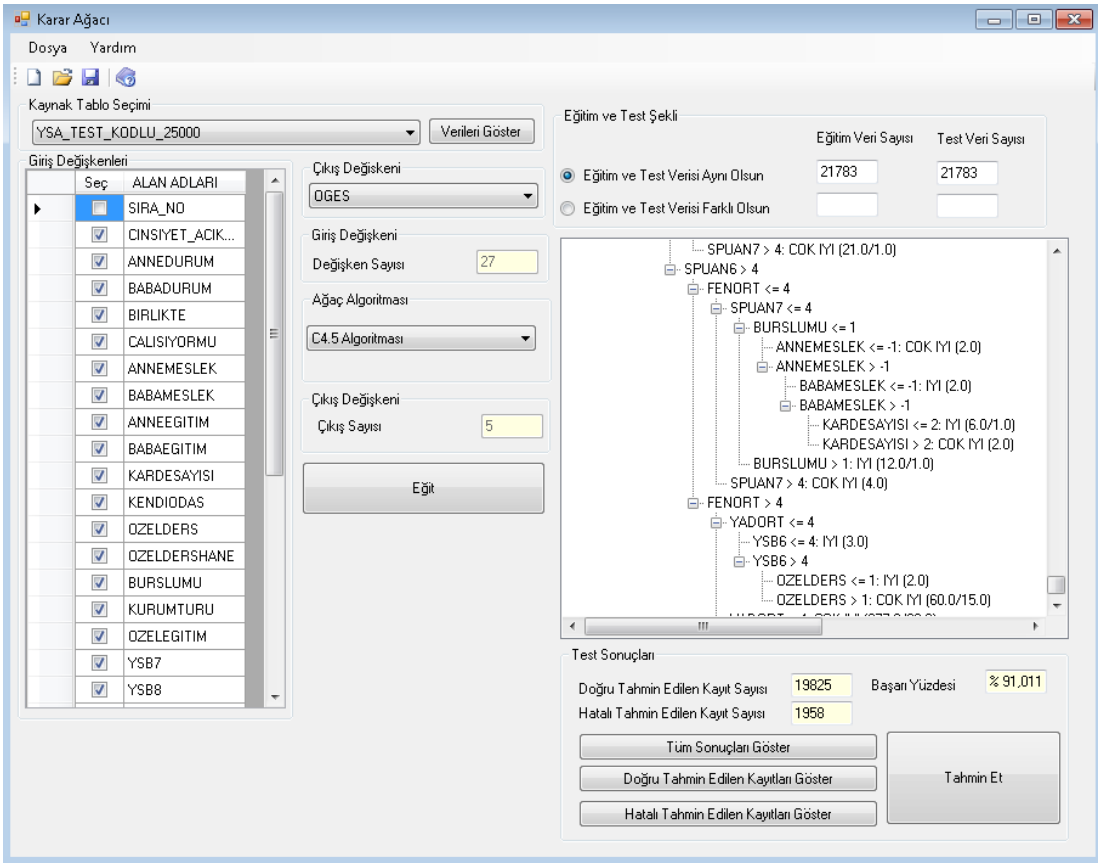
Bir karar ağacı oluşturulduğunda, birçok dalda, öğrenme verisindeki gürültü ve kayıplardan dolayı aykırılık oluşacaktır. Ağacın budanma metodu bu sorunu ortadan kaldırmaya yardımcı olabilir. Bu metot, tipik olarak en az güvenilir olan dalı istatistiksel olarak hesaplayıp kaldırmaktan ibarettir ve daha hızlı ve güvenilir bir sınıflandırma ile sonuçlanır. İki adet budama yöntemi vardır.

Bunlardan birincisi, önceden budama yöntemidir. C4.5 karar ağaçlarında kullanılan ön budama yöntemi, daha az hesaplama içermesi, veri setinin ayrılması için en iyi yolu araştırması ve bilgi kazancının değerlendirilmesi yönlerinden önemli avantajlara sahiptir. Bu yöntemde öğrenme verisi sınıflandırılırken ağacın o dalının ileriye yönelik devam edip etmeyeceğine önceden karar verilir ve gerekiyorsa, geri kalan bölünmeden sonra geriye kalan verinin sınıflandırılması durdurularak, en fazla hedef değeri taşıyan değer yaprak yapılır. Bu yöntemde, önceden bir eşik değeri belirlenir. Bu eşik değerini aşmayan bilgi kazançlarına sahip olan nitelikler gruplandırılır. Program devam ederken bu bilgi kazancının düştüğü noktada ağacın büyümesine izin verilmeden diğer dala geçilir. İkinci yöntem, sonradan budama yöntemidir. Tamamen büyümüş bir ağaç üzeride uygulanır. Tüm dalların çıkardığı kurallar denenerek, bunlardan en fazla hata oranını oluşturan dal budanır. Böylece ortaya daha basit bir ağaç yapısı çıkartılabilir.

Karar ağaçları ile sınıflandırma uygulamasını başlatmak için Karar Ağaçları menüsü altında bulunan Karar Ağaçları Modeli seçeneği tıklanır (Şekil 4.13).

Şekil 4.13. Karar ağaçları modeli giriş ekranı.

Gelen ekranda Oracle veritabanında bulunan tablolar Kaynak Tablo Seçimi alanında listelenir. Açılır listeden uygulamada kullanılacak verilerin bulunduğu tablo adı seçilir. Daha sonra bu ekranda gelen giriş değişkenleri ve çıkış değişkeni kullanıcı tarafından seçilir. Seçilen tablo içerisindeki veriler Verileri Görüntüle butonu ile görüntülenebilir. Verilere uygulanmak istenen ağaç algoritması seçilir. Giriş ve çıkış katmanına ait eleman sayıları seçilen değişkenlere göre program tarafından otomatik olarak atandığı için değiştirilememektedir. Ekranda bulunan test şeklinin nasıl olacağına dair bilgiler kullanıcı tarafından değiştirilebilecek alanlardır. Eğitim ve test sonucunda oluşan karar ağacı görüntüsü de bu ekranda test sonuçları üzerinde görüntülenmektedir (Şekil 4.14).



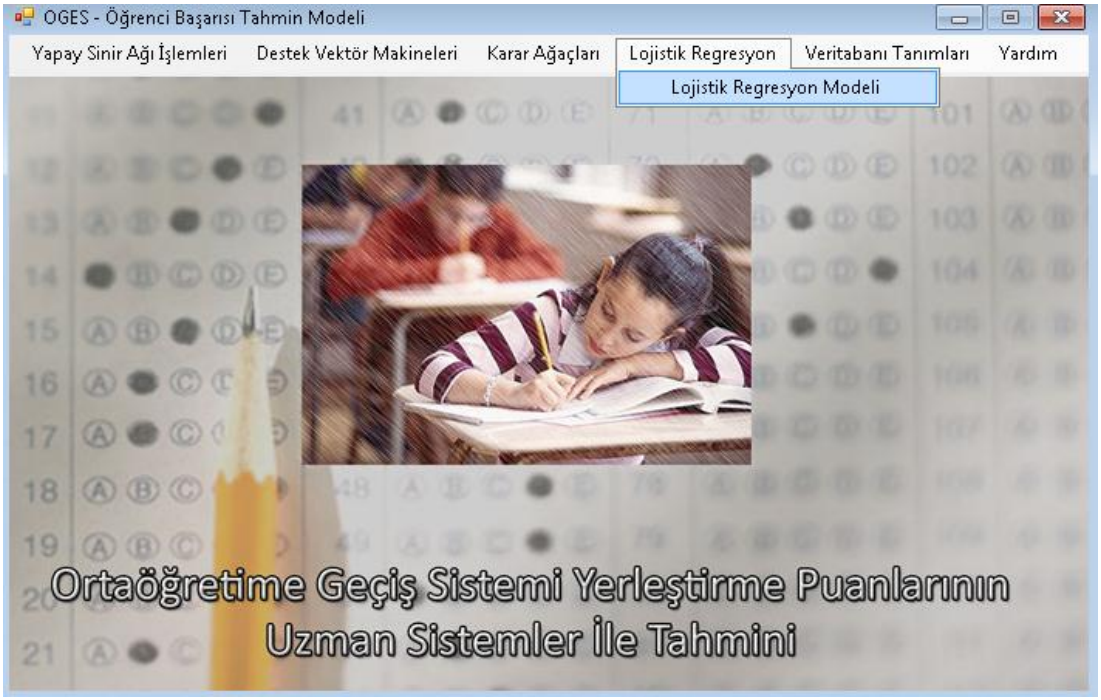
Şekil 4.14. Eğitim ve test sonucunda oluşan ekran görüntüsü (KA).

Diğer tüm özellikler yapay sinir ağı modellerinde olduğu gibi karar ağaçları modeli için de bulunmaktadır.

4.5. LOJİSTİK REGRESYON ANALİZİ UYGULAMASI

Lojistik regresyon analizinin temel amacı diğer regresyon yöntemlerinde olduğu gibi bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi incelemektir. Uygulamada bağımlı değişkenin ikiden fazla düzeyli kategoriden oluşması sebebiyle Multinomial lojit model kullanılmıştır. Multinomial lojistik regresyon, ikili lojit modelin; bağımlı değişken ikiden fazla düzeyli kategoriden oluştuğu duruma genişletilmiş halidir.

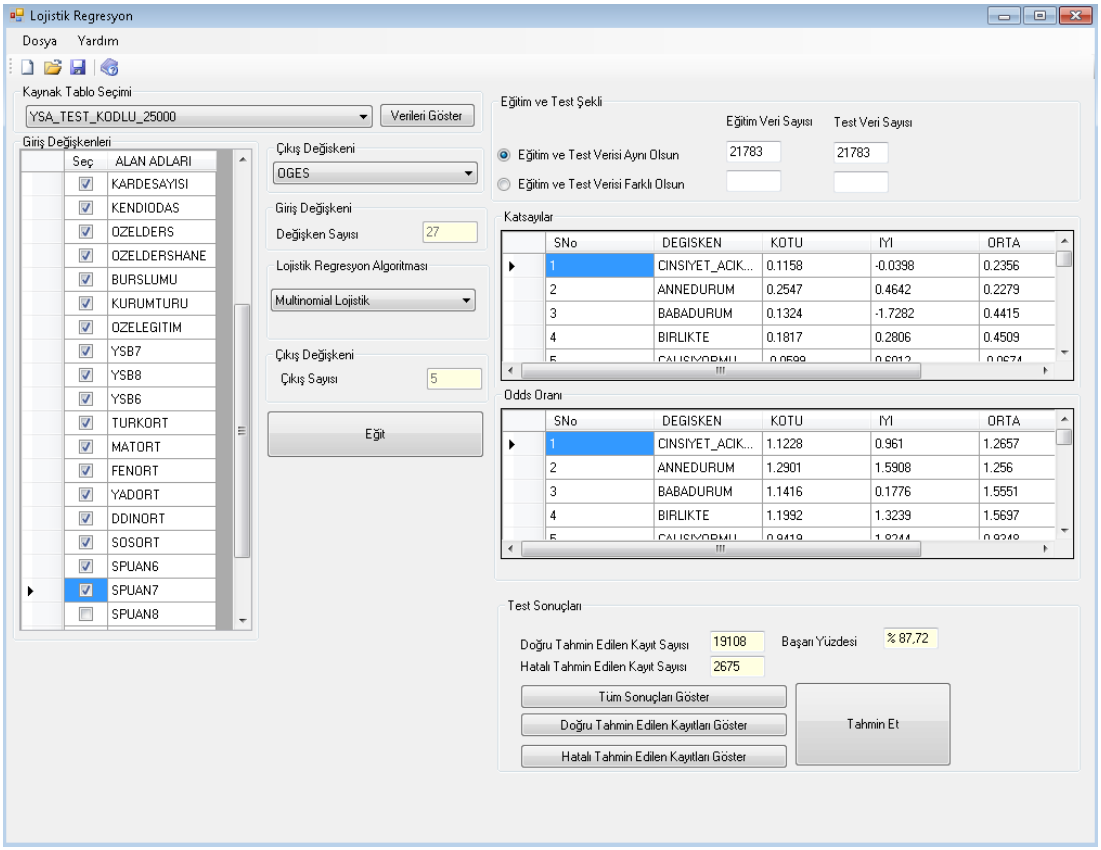
Lojistik Regresyon analizi ile sınıflandırma uygulamasını başlatmak için Lojistik Regresyon menüsü altında bulunan Lojistik Regresyon Modeli seçeneği tıklanır (Şekil 4.15).



Şekil 4.15. Lojistik regresyon analizi modeli giriş ekranı.

Gelen ekranda Oracle veritabanında bulunan tablolar Kaynak Tablo Seçimi alanında listelenir. Açılır listeden uygulamada kullanılacak verilerin bulunduğu tablo adı seçilir. Daha sonra bu ekranda gelen giriş değişkenleri ve çıkış değişkeni kullanıcı tarafından seçilir. Seçilen tablo içerisindeki veriler Verileri Görüntüle butonu ile görüntülenebilir. Verilere uygulanmak istenen logistik regresyon analizi seçilir.

Giriş ve çıkış katmanına ait eleman sayıları seçilen değişkenlere göre program tarafından otomatik olarak atandığı için değiştirilememektedir. Ekranda bulunan test şeklinin nasıl olacağına dair bilgiler kullanıcı tarafından değiştirilebilecek alanlardır. Eğitim ve test sonucunda oluşan analiz sonuçları bu ekranda test sonuçları üzerinde görüntülenmektedir (Şekil 4.16).



Şekil 4.16. Eğitim ve test sonucunda oluşan ekran görüntüsü (LRA).

Diğer tüm özellikler yapay sinir ağı modellerinde olduğu gibi lojistik regresyon modeli için de bulunmaktadır.

BÖLÜM 5

SONUÇLAR ve DEĞERLENDİRME

Çalışmada ilk olarak yapay sinir ağları modeli, daha sonra da sırasıyla destek vektör makineleri, karar ağaçları ve lojistik regresyon analizi modelleri oluşturulmuştur. Yapay sinir ağları modelinde çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağları olmak üzere iki farklı mimari kullanılmıştır.

Çok katmanlı algılayıcı ile yapılan eğitimde 1.gizli katmanda 10, ikinci gizli katmanda 8 ve çıkış katmanında 5 nörondan oluşan bir ağ yapısı tasarlanmıştır. Momentum ve öğrenme oranı ayarlanabilen bir geri yayılım algoritması kullanılmıştır. Eğitimde öğrenme oranı 0.25 ve momentum 0.07 seçildiğinde en iyi sonuçlar alınmıştır. Eğitim işlemi 1000 adımda tamamlanmıştır. Modelde tüm katmanlar için aktivasyon fonksiyonu sigmoid olarak seçilmiştir. Eğitim aşaması yaklaşık 45 dakika, test aşaması ise 50 dakika sürmüştür. Çizelge 5.1'de değiştirilen parametrelere göre ÇKA modelinin hata değeri ve başarı yüzdesi verilmiştir.

Çizelge 5.1. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (ÇKA).

	Öğr. Oranı	Giriş Kat. Nör.	Giriş Kat. Aktiv. Fonk.	1. Gizli Kat. Nör.	1.Gizli Kat. Aktiv. Fonk.	2. Gizli Kat. Nör.	2. Gizli Kat. Aktiv. Fonk.	Çıkış Kat. Nör.	Çıkış Kat. Aktiv. Fonk.	Hata	Başarı Oranı (%)
Çok Katmanlı Algılayıcı	0,25	27	Sigmoid	30	Sigmoid	-	---	5	Sigmoid	0,16	88,09
	0,25	27	Sigmoid	10	Sigmoid	8	Sigmoid	5	Sigmoid	0,15	89,09
	0,3	27	Sigmoid	10	Sigmoid	8	Sigmoid	5	Sigmoid	0,18	86,38
	0,4	27	Sigmoid	10	Sigmoid	8	Sigmoid	5	Sigmoid	0,19	87,78
	0,5	27	Sigmoid	10	Sigmoid	8	Sigmoid	5	Sigmoid	0,21	83,38
	0,3	27	Sigmoid	30	Sigmoid	-	---	5	Sigmoid	0,15	88,08
	0,4	27	Sigmoid	30	Sigmoid	-	---	5	Sigmoid	0,18	86,38
	0,5	27	Sigmoid	30	Sigmoid	-	---	5	Sigmoid	0,19	85,68
	0,25	27	Tanjant	30	Tanjant	-	---	5	Tanjant	0,48	66,36

Radyal tabanlı fonksiyon ile yapılan eğitimde gizli katmanda sırasıyla 30, 50, 75, 100, 125, 150, 175 nöron kullanılmıştır. Bant genişliği [0,2] arasında kullanılmıştır. Aktivasyon fonksiyonu olarak gauss kullanılmıştır. Eğitim işlemi 1000 adımda tamamlanmıştır. Eğitim aşaması yaklaşık 10 dakika test aşaması ise 15 dakika sürmüştür. Yapılan eğitimler sonucu elde edilen hata değeri ve başarı yüzdesi Çizelge 5.2'de gösterildiği gibidir.

Çizelge 5.2. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (RTF).

	Giriş Katmanı Nöron	Ara Katman Aktivasyon Fonksiyonu	Ara Katman Nöron Sayısı	Çıkış Katmanı Nöron	Hata	Başarı Oranı (%)
	27	Gauss	30	5	0,153	49,34
	27	Gauss	50	5	0,135	54,63
	27	Gauss	75	5	0,123	56,86
Radyal	27	Gauss	100	5	0,097	68,75
Tabanlı	27	Gauss	125	5	0,096	69,5
Fonksiyon	27	Gauss	150	5	0,09	70,43
	27	Gauss	160	5	0,081	72,35
	27	Gauss	165	5	0,098	67,1
	27	Gauss	175	5	0,099	66,45
	27	Gauss	200	5	0,102	60,85

Destek vektör makineleri ile yapılan eğitimde veri setimizin çıkışı 5 sınıflı olduğu için 10 DVM eğitilmiştir. Her bir eğitim aşamasında sadece iki farklı sınıf verisi alınmıştır. Çekirdek fonksiyonu radyal tabanlı fonksiyon olarak seçilmiştir. Eğitim ve test aşamaları birlikte 5 dakika sürmüştür. Yapılan eğitimler sonucu elde edilen başarı yüzdesi ve parametreler Çizelge 5.3'te gösterildiği gibidir.

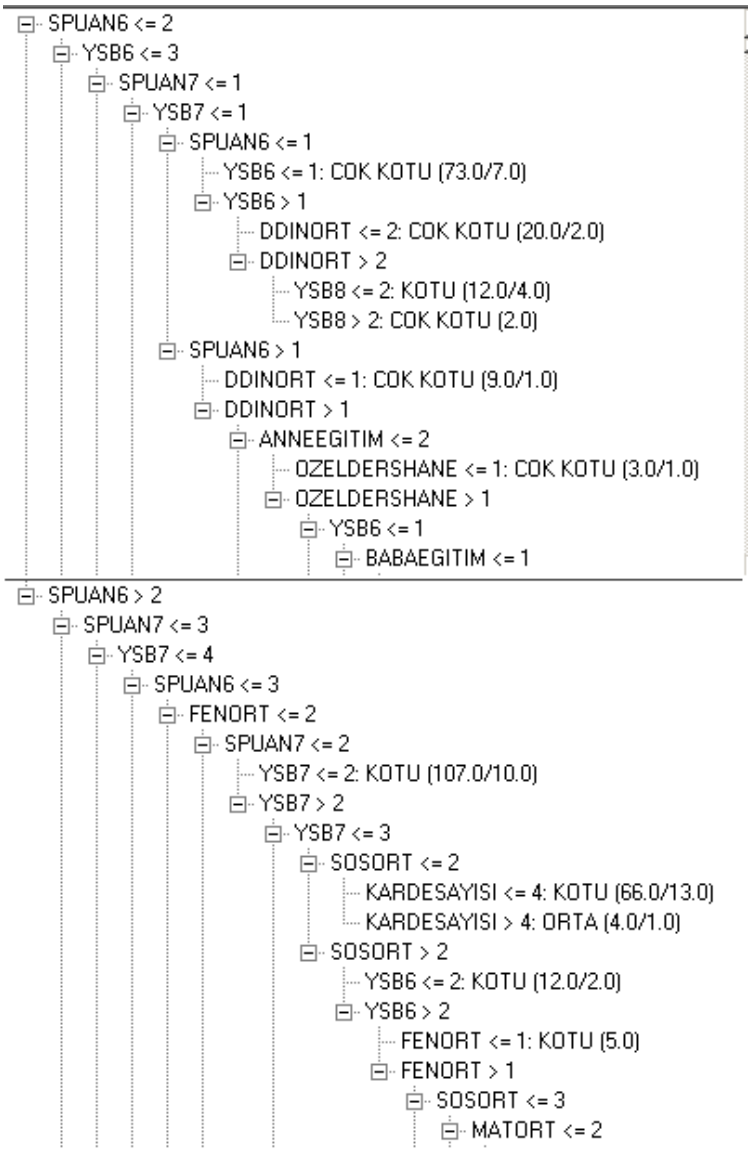
Çizelge 5.3. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (DVM).

	Giriş Katmanı Değişken Sayısı	Çekirdek Fonksiyonu	Çoklu DVM Çözümü	Hata	Başarı Oranı (%)
Destek	27	Polinomial	Bire Karşı Bir	0,001	71,474
Vektör	27	Sigmoid	Bire Karşı Bir	0,001	15,01
Makineleri	27	Linear	Bire Karşı Bir	0,001	76,413
	27	Radyal Tabanlı Fonksiyon	Bire Karşı Bir	0,001	88,404

Karar ağaçları kullanılarak yapılan eğitimde C4.5 karar ağacı türetme algoritması kullanılmış ve eğitim ve test işlemleri yapılmıştır. Eğitim ve test aşamaları birlikte 12 dakika sürmüştür. Yapılan eğitimler sonucunda elde edilen başarı yüzdesi ve parametreler Çizelge 5.4'te, karar ağacı ise Şekil 5.1'de gösterilmiştir.

Çizelge 5.4. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (KA).

	Giriş Katmanı Değişken Sayısı	Kullanılan Algoritma	Hata	Başarı Oranı (%)
Karar Ağaçları	27	C4.5	0,001	91,584



Şekil 5.1. Yapılan eğitimler sonucunda elde edilen karar ağacı.

Karar ağacı algoritmasında kullanılan değişkenlerin bilgi kazancı ve bilgi kazancı oranları Çizelge 5.5'te gösterildiği gibidir.

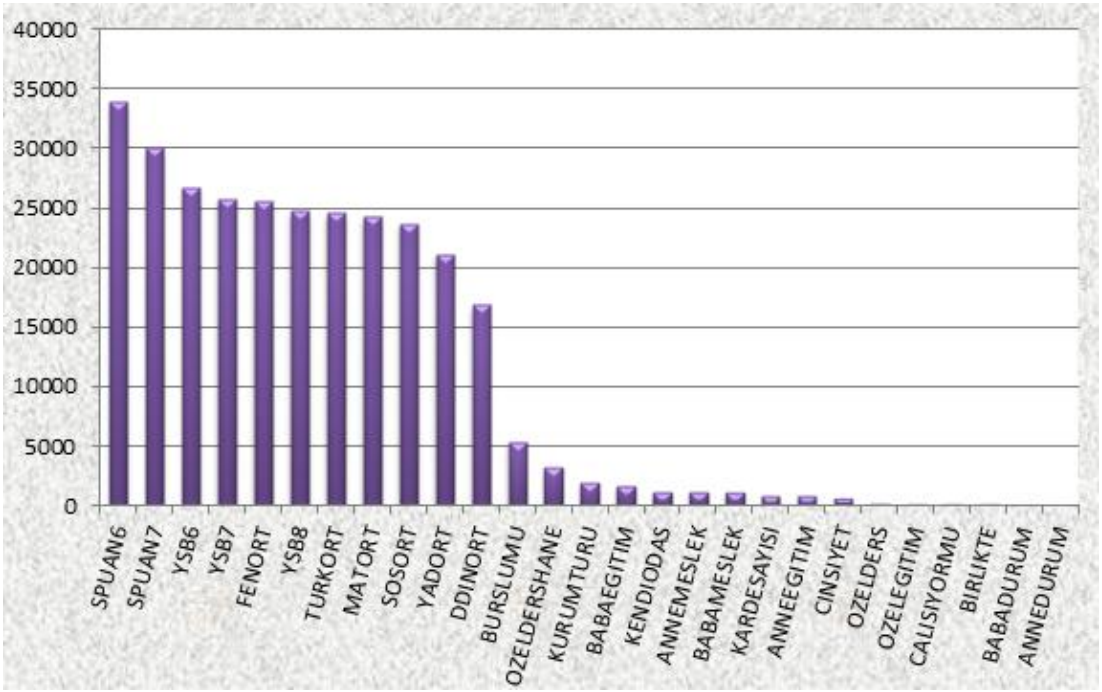
Çizelge 5.5. Değişkenlere ait bilgi kazancı ve bilgi kazancı oranı değerleri.

Sıra No	Alan Adı	Bilgi Kazancı	Bilgi Kazancı Oranı
1	CINSIYET	0.01947	0.01948
2	ANNEDURUM	0	0
3	BABADURUM	0	0
4	BIRLIKTEMI	0.000879	0.00236
5	CALISIYORMU	0.001169	0.00982
6	ANNEMESLEK	0.030636	0.04699
7	BABAMESLEK	0.036426	0.01844
8	ANNEEGITIM	0.026015	0.02321
9	BABAEGITIM	0.049199	0.03213
10	KARDESSAYISI	0.030123	0.01650
11	KENDIODASI	0.039067	0.03909
12	OZELDERS	0.006396	0.04120
13	OZELDERSHANE	0.104062	0.12827
14	BURSLUMU	0.093560	0.39665
15	KURUMTURU	0.040536	0.20153
16	OZELEGITIM	0.002473	0.06850
17	YSB6	0.788751	0.38809
18	YSB7	0.765074	0.37175
19	YSB8	0.740134	0.36516
20	TURKORT	0.754887	0.34828
21	MATORT	0.748831	0.32936
22	FENORT	0.789135	0.35688
23	YADORT	0.661830	0.29513
24	DINORT	0.543328	0.30121
25	SOSORT	0.735688	0.34105
26	SBS6	0.875317	0.49256
27	SBS7	0.744582	0.43165

Lojistik regresyon kullanılarak yapılan eğitimde multinomial lojistik regresyon analizi kullanılmış ve eğitim ve test işlemleri yapılmıştır. Eğitim ve test aşamaları birlikte 7 dakika sürmüştür. En iyi sonuç veren analizi bulabilmek amacıyla her seferinde bir yanıt değişkeni referans olarak kabul edilmiştir. Yapılan eğitimler sonucunda elde edilen başarı yüzdesi ve parametreler Çizelge 5.6'da gösterildiği gibidir. Parametrelerin etki sırasına göre dağılımı Şekil 5.2'de gösterilmiştir.

Çizelge 5.6. Değiştirilen parametrelere göre başarı yüzdesi (LR).

	Giriş Katmanı Değişken Sayısı	Kullanılan Lojistik Regresyon Analizi	Referans	Başarı Oranı (%)
	27	Multinomial	ÇOK KÖTÜ	87,145
Lojistik	27	Multinomial	KOTU	87,091
Regresyon	27	Multinomial	ORTA	87,294
Analizi	27	Multinomial	IYI	87,280
	27	Multinomial	ÇOK IYI	87,091



Şekil 5.2. Parametrelerin etki dağılımı.

Çizelge 5.7. Multinomial lojistik regresyon analizi ki kare dağılımları.

Sıra No	Alan Adı	Ki Kare Dağılımı
1	CINSIYET	574,10
2	ANNEDURUM	0,00
3	BABADURUM	0,00
4	BIRLIKTEMI	27,06
5	CALISIYORMU	36,02
6	ANNEMESLEK	1138,66
7	BABAMESLEK	1067,48
8	ANNEEGITIM	832,68
9	BABAEGITIM	1679,76
10	KARDESSAYISI	868,86
11	KENDIODASI	1142,32
12	OZELDERS	194,17
13	OZELDERSHANE	3142,36
14	BURSLUMU	5352,55
15	KURUMTURU	1917,98
16	OZELEGITIM	66,27
17	YSB6	26774,27
18	YSB7	25765,82
19	YSB8	24741,78
20	TURKORT	24595,94
21	MATORT	24221,96
22	FENORT	25512,47
23	YADORT	21089,04
24	DINORT	16874,18
25	SOSORT	23687,52
26	SBS6	33867,65
27	SBS7	30056,04

Çizelge 5.8. Multinomial lojistik regresyon analizi katsayı değerleri (REF = ORTA).

Sıra No	Alan Adı	ÇOK KÖTÜ	KÖTÜ	İYİ	ÇOK İYİ
1	CINSIYET	-0.2493	-0.1292	-0.2688	-0.4076
2	ANNEDURUM	-0.2958	-0.0209	0.2479	-0.2186
3	BABADURUM	-0.5578	-0.4185	-2.1307	-44.2402
4	BIRLIKTEMI	-0.5188	-0.2987	-0.1287	0.3165
5	CALISIYORMU	0.0324	-0.024	0.6822	1.6636
6	ANNEMESLEK	-0.1185	0.0091	-0.0707	0.0216
7	BABAMESLEK	-0.0176	-0.0159	-0.034	-0.0614
8	ANNEEGITIM	0.1595	-0.0329	0.0442	-0.0484
9	BABAEGITIM	0.0211	-0.0718	0.038	-0.0158
10	KARDESSAYISI	0.2964	0.1606	-0.1441	-0.2207
11	KENDIODASI	-0.3155	-0.1461	0.0012	0.0893
12	OZELDERS	-0.6674	-0.1653	-0.1444	-0.0284
13	OZELDERSHANE	0.7234	0.4184	-0.4633	-0.5263
14	BURSLUMU	-0.6628	-0.3583	0.876	1.2978
15	KURUMTURU	11.72	0.0279	-0.1979	-0.5938
16	OZELEGITIM	3.6299	1.1825	-2.0605	153.9544
17	YSB6	-1.8763	-0.9435	1.3084	2.8113
18	YSB7	-0.4565	-0.1091	-0.0331	0.0525
19	YSB8	-0.917	-0.2947	0.204	0.922
20	TURKORT	-1.261	-0.857	0.361	0.8908
21	MATORT	-0.5057	-0.3549	0.9945	2.7709
22	FENORT	-0.6666	-0.6488	0.8198	2.0506
23	YADORT	-0.3927	-0.3465	0.2224	1.3656
24	DINORT	-0.9722	-0.4286	0.38	0.3994
25	SOSORT	-0.8325	-0.6425	0.3813	0.733
26	SBS6	-5.2133	-2.8265	2.5889	4.4198
27	SBS7	-4.771	-2.7186	4.4347	8.3437
	INTERCEPT	9.2714	25.5865	-36.5438	-367.8072

Çizelge 5.9. Multinomial lojistik regresyon analizi odds oranları (REF = ORTA).

Sıra No	Alan Adı	ÇOK KÖTÜ	KÖTÜ	İYİ	ÇOK İYİ
1	CINSIYET	0.7794	0.8788	0.7643	0.6653
2	ANNEDURUM	0.744	0.9793	1.2814	0.8036
3	BABADURUM	0.5725	0.6581	0.1188	0
4	BIRLIKTEMI	0.5952	0.7418	0.8792	1.3723
5	CALISIYORMU	1.033	0.9763	1.9783	5.2781
6	ANNEMESLEK	0.8883	1.0091	0.9317	1.0218
7	BABAMESLEK	0.9826	0.9842	0.9666	0.9404
8	ANNEEGITIM	1.1729	0.9676	1.0452	0.9528
9	BABAEGITIM	1.0213	0.9307	1.0387	0.9843
10	KARDESSAYISI	1.3451	1.1742	0.8658	0.802
11	KENDIODASI	0.7295	0.8641	1.0012	1.0935
12	OZELDERS	0.5131	0.8476	0.8655	0.972
13	OZELDERSHANE	2.0614	1.5196	0.6292	0.5908
14	BURSLUMU	0.5154	0.6989	2.4012	3.6612
15	KURUMTURU	123002.4	1.0283	0.8205	0.5522
16	OZELEGITIM	37.7096	3.2626	0.1274	7.270E66
17	YSB6	0.1531	0.3892	3.7001	16.6316
18	YSB7	0.6335	0.8967	0.9675	1.0539
19	YSB8	0.3997	0.7448	1.2263	2.5143
20	TURKORT	0.2834	0.4244	1.4348	2.437
21	MATORT	0.6031	0.7013	2.7035	15.9732
22	FENORT	0.5135	0.5227	2.2701	7.7723
23	YADORT	0.6752	0.7072	1.249	3.918
24	DINORT	0.3782	0.6514	1.4622	1.4909
25	SOSORT	0.435	0.526	1.4641	2.0814
26	SBS6	0.0054	0.0592	13.3152	83.0772
27	SBS7	0.0085	0.066	84.3254	4203.7961

Tezde kullanılan yöntemlerin sınıflandırma başarılarına bakıldığında karar ağaçları %91,584 lük bir doğruluk ile birinci sırada yer almaktadır. Karar ağaçlarını %89,09 ile çok katmanlı yapay sinir ağları, %88,404 ile destek vektör makineleri ve %87,294 ile lojistik regresyon analizi takip etmektedir. Radyal tabanlı fonksiyon ise %72,35 ile sonuncu sırada yer almaktadır.

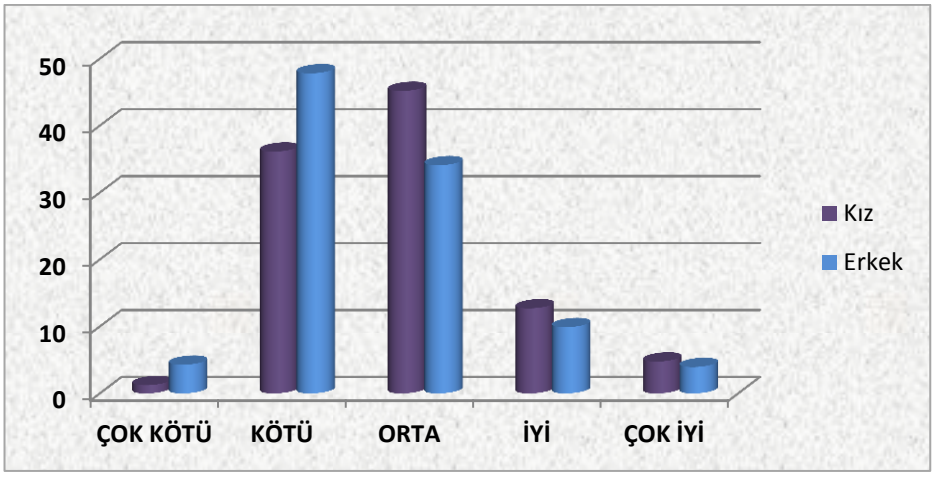
Tez çalışması süresince öncelikle rastgele seçilmiş 5000 kayıt üzerinde SPPS programı kullanılarak yapılan çalışmada çapraz doğrulama (10 fold cross validation) yöntemi ile sonuçlar elde edilmiştir. Bu çalışmada sonuca en çok etki eden parametrelerin belirlenmesi için duyarlılık analizi kullanılmıştır [129]. Daha sonra çok katmanlı yapay sinir ağı mimarisine ek olarak, radyal tabanlı fonksiyon çalışmaya dâhil edilmiş ve çok katmanlı yapay sinir ağı mimarisinin tahmin ve doğruluk olarak daha sonuç verdiği görülmüştür [130].

Değişkenler tek tek incelendiğinde cinsiyet değişkenin Ortaöğretime Geçiş Sistemi (OGES) yerleştirme puanına göre dağılımı Çizelge 5.10'da görüldüğü gibidir. Cinsiyete göre OGES puan dağılımı Şekil 5.3'de gösterilmiştir.

Çizelge 5.10. OGES puanlarının cinsiyete göre dağılımı.

		ÇOK KÖTÜ	**KÖTÜ**	**ORTA**	**İYİ**	**ÇOK İYİ**
Kız	**Değer**	130	3817	4774	1342	502
	%	**1,23**	**36,12**	**45,18**	**12,71**	**4,76**
Erkek	**Değer**	481	5363	3823	1109	442
	%	**4,28**	**47,81**	**34,08**	**9,89**	**3,94**

Çizelge ve grafikten anlaşılacağı üzere kız öğrenciler erkek öğrencilere göre daha başarılıdır.

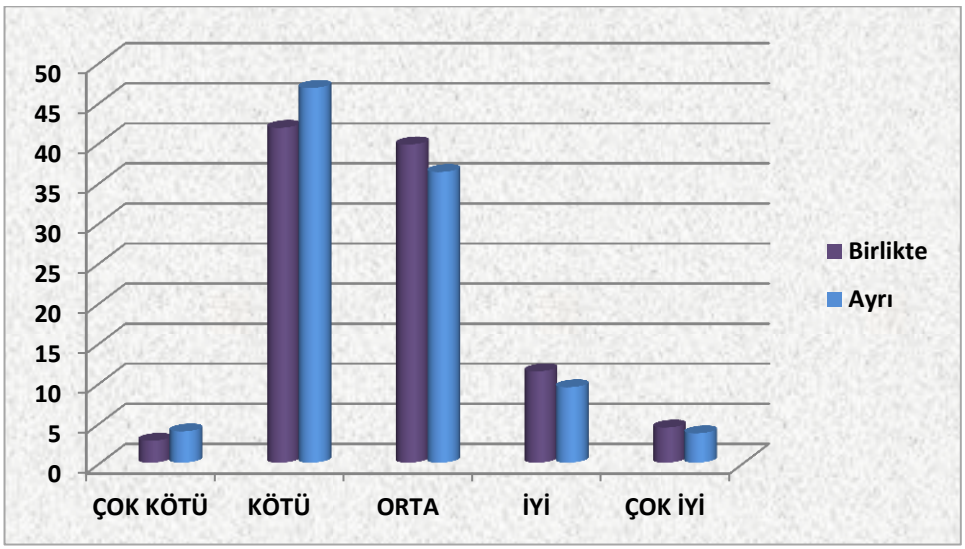


Şekil 5.3. OGES puanlarının cinsiyete göre dağılım grafiği.

Anne ve babasının birlikte veya boşanmış olup olmamalarının öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.11'de görüldüğü gibidir. Birlikte değişkenine göre OGES puan dağılımı Şekil 5.4'de gösterilmiştir.

Çizelge 5.11. OGES puanlarının anne babanın birlikte olmasına göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Birlikte	**Değer**	550	8449	8030	2304	887
	%	**2,73**	**41,78**	**39,71**	**11,4**	**4,38**
Ayrı	**Değer**	61	731	567	147	57
	%	**3,9**	**46,76**	**36,27**	**9,41**	**3,66**



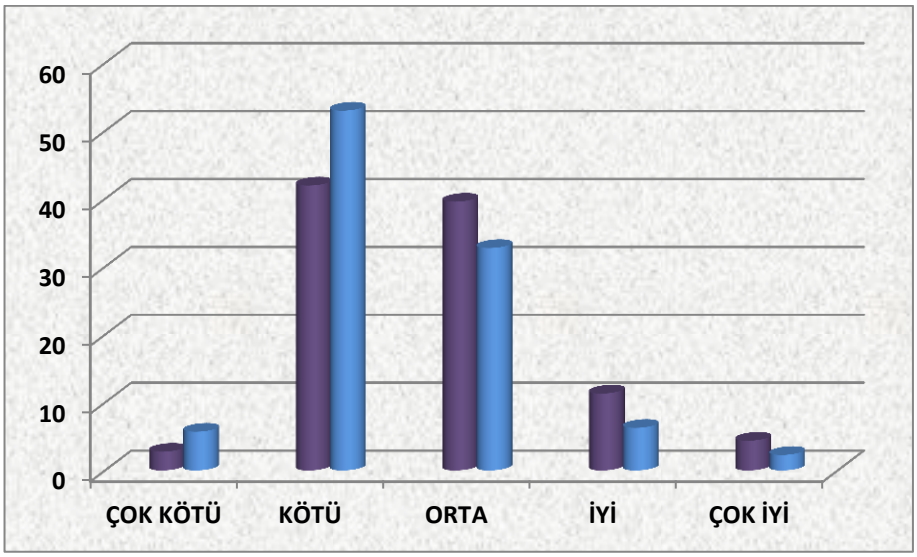
Şekil 5.4. OGES puanlarının anne babanın birlikte olmasına göre dağılım grafiği.

Çizelgeden ve grafikten anlaşılacağı üzere öğrencinin anne babasının evli veya boşanmış olmasının öğrenci başarısı üzerinde çok etkili olmadığı görülmüştür.

Öğrencinin çalışıp çalışmamasının öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.12'de görüldüğü gibidir. Çalışıp çalışmama değişkenine göre OGES puan dağılımı Şekil 5.5'de gösterilmiştir.

Çizelge 5.12. OGES puanlarının öğrencinin çalışıp çalışmamasına göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Çalışıyor	**Değer**	20	186	115	22	8
	%	**5,7**	**52,99**	**32,76**	**6,27**	**2,28**
Çalışmıyor	**Değer**	591	8994	8482	2429	936
	%	**2,75**	**41,96**	**39,58**	**11,33**	**4,38**



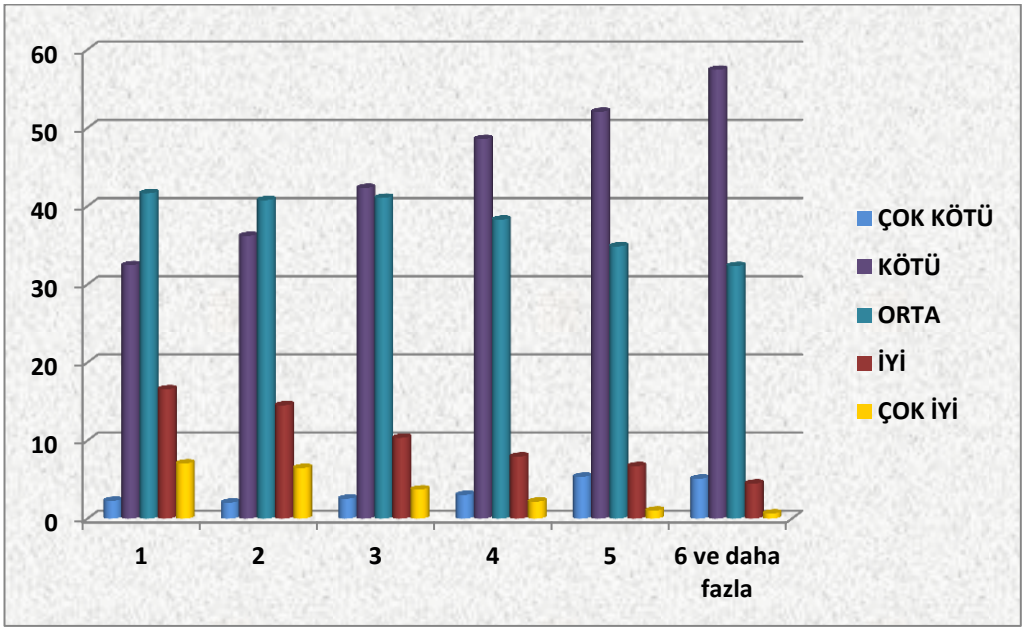
Şekil 5.5. OGES puanlarının öğrencinin çalışıp çalışmamasına göre dağılım grafiği.

Çizelgeden ve grafikten anlaşılacağı üzere çalışan öğrencilerin sınav puanlarının çalışmayanlara göre daha düşük olduğu görülmüştür.

Öğrencilerin kardeş sayısının öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.13'de görüldüğü gibidir. Kardeş sayısı değişkenine göre OGES puan dağılımı Şekil 5.6'da gösterilmiştir.

Çizelge 5.13. OGES puanlarının öğrencinin kardeş sayısına göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
1	**Değer**	31	441	566	225	96
	%	**2,28**	**32,45**	**41,65**	**16,56**	**7,06**
2	**Değer**	168	2978	3355	1195	533
	%	**2,04**	**36,19**	**40,77**	**14,52**	**6,48**
3	**Değer**	157	2605	2527	637	229
	%	**2,55**	**42,32**	**41,06**	**10,35**	**3,72**
4	**Değer**	84	1340	1056	219	60
	%	**3,04**	**48,57**	**38,27**	**7,94**	**2,18**
5	**Değer**	68	660	442	85	13
	%	**5,36**	**52,05**	**34,86**	**6,71**	**1,02**
6 ve daha fazla	**Değer**	103	1156	651	90	13
	%	**5,11**	**57,42**	**32,34**	**4,48**	**0,65**



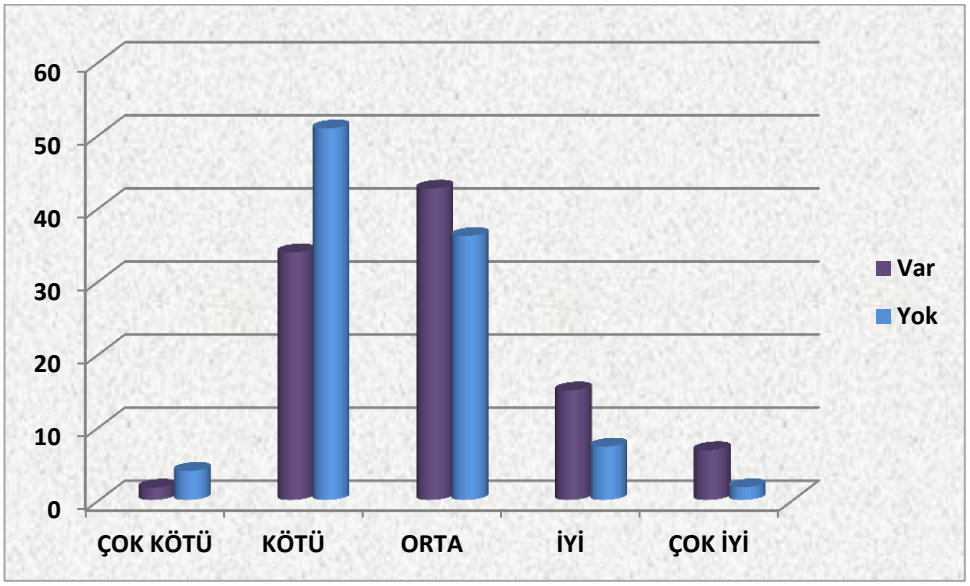
Şekil 5.6. OGES puanlarının kardeş sayısına göre dağılım grafiği.

Çizelgeden ve grafikten anlaşılacağı üzere kardeş sayısı arttıkça başarı oranı da düşmektedir.

Öğrencilerin kendine ait bir odasının olup olmamasının öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.14'de görüldüğü gibidir. Kendi odası değişkenine göre OGES puan dağılımı Şekil 5.7'de gösterilmiştir.

Çizelge 5.14. OGES puanlarının öğrencinin kendi odası olmasına göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Var	**Değer**	189	3792	4770	1678	761
	%	**1,69**	**33,89**	**42,62**	**14,99**	**6,81**
Yok	**Değer**	422	5388	3827	773	183
	%	**3,98**	**50,86**	**36,13**	**7,3**	**1,73**



Şekil 5.7. OGES puanlarının kendi odası olup olmamasına göre dağılım grafiği.

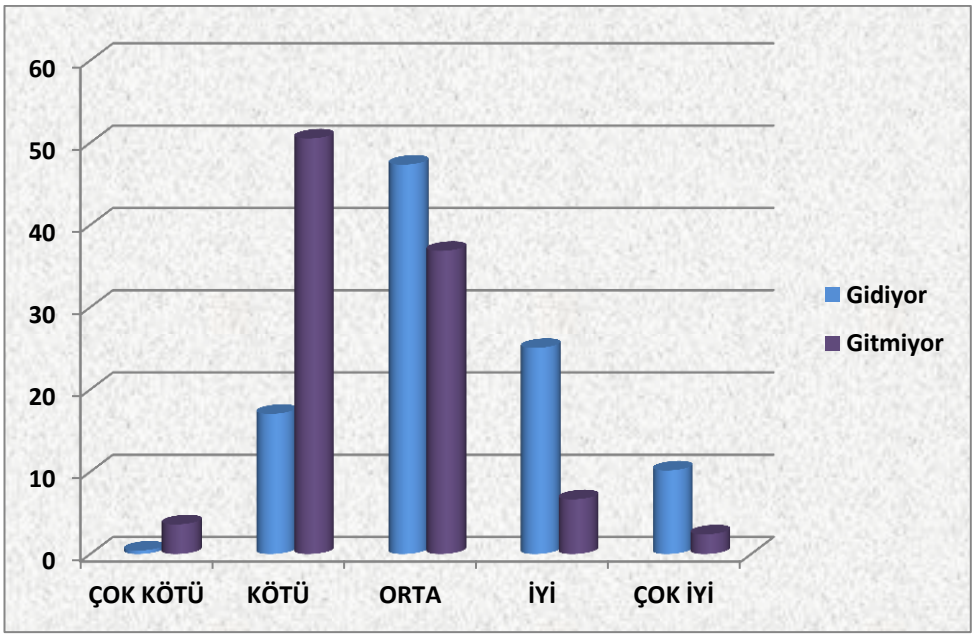
Çizelgeden ve grafikten görüleceği üzere kendine ait bir çalışma odası olan öğrencilerin başarı oranı daha yüksektir.

Öğrencilerin dershaneye gidip gitmemesinin öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.15'de görüldüğü gibidir. Özel dershane değişkenine göre OGES puan dağılımı Şekil 5.8'de gösterilmiştir.

Çizelge 5.15. OGES puanlarının öğrencinin dershaneye gitmesine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Gidiyor	**Değer**	23	928	2578	1364	553
	%	**0,42**	**17,04**	**47,33**	**25,05**	**10,16**
Gitmiyor	**Değer**	588	8252	6019	1087	391
	%	**3,6**	**50,51**	**36,85**	**6,65**	**2,39**

Çizelgeden ve grafikten görüleceği üzere dershaneye giden öğrencilerin başarı oranı daha yüksektir.

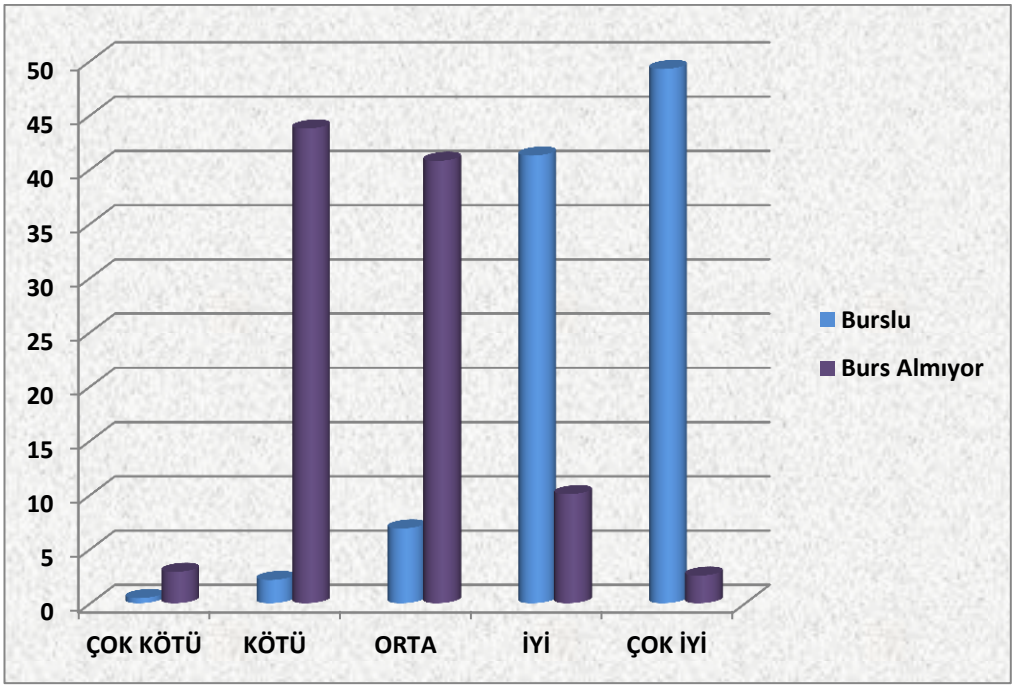


Şekil 5.8. OGES puanlarının öğrencinin dershaneye gitmesine göre dağılım grafiği.

Öğrencilerin burs alıp almamasının öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.16'da görüldüğü gibidir. Burs değişkenine göre OGES puan dağılımı Şekil 5.9'da gösterilmiştir.

Çizelge 5.16. OGES puanlarının öğrencinin burslu olup olmamasına göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Burs Almıyor	**Değer**	607	9162	8539	2104	530
	%	**2,89**	**43,75**	**40,77**	**10,05**	**2,54**
Burslu	**Değer**	4	18	58	347	414
	%	**0,47**	**2,14**	**6,89**	**41,26**	**49,24**



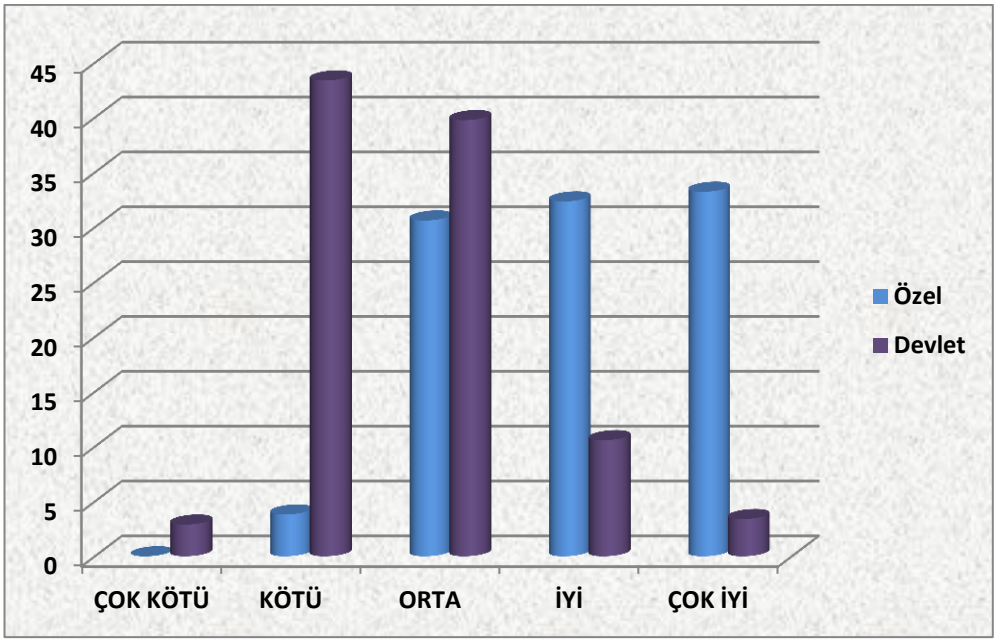
Şekil 5.9. OGES puanlarının öğrencinin burs alıp almamasına göre dağılım grafiği.

Çizelgeden ve grafikten görüleceği üzere burslu öğrencilerin başarı oranı, burs almayanlara göre daha yüksektir.

Öğrencilerin özel okulda veya devlet okulunda öğrenim görmesinin öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.17'de görüldüğü gibidir. Kurum türü değişkenine göre OGES puan dağılımı Şekil 5.10'da gösterilmiştir.

Çizelge 5.17. OGES puanlarının öğrencinin öğrenim gördüğü kuruma göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Özel	**Değer**	0	26	209	221	227
	%	**0**	**3,81**	**30,61**	**32,35**	**33,23**
Devlet	**Değer**	611	9154	8388	2230	717
	%	**2,89**	**43,38**	**39,76**	**10,58**	**3,39**



Şekil 5.10. OGES puanlarının öğrenim gördüğü kuruma göre dağılım grafiği.

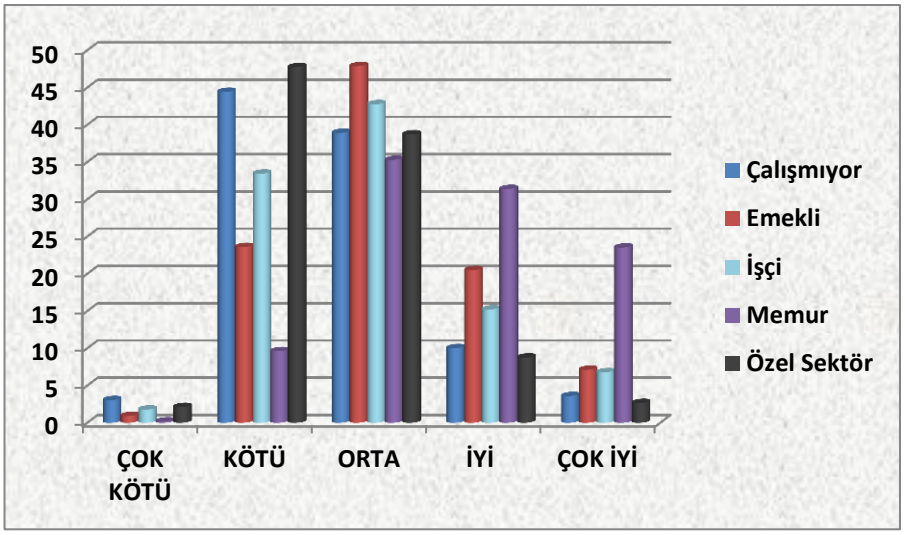
Çizelgeden ve grafikten görüleceği üzere özel okulda öğrenim gören öğrencilerin başarı oranı, devlet okulunda öğrenim görenlere göre daha yüksektir.

Öğrencilerin annesinin mesleğinin öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.18'de görüldüğü gibidir. Anne mesleği değişkenine göre OGES puan dağılımı Şekil 5.11'de gösterildiği gibidir.

Çizelge 5.18. OGES puanlarının öğrencilerin annesinin mesleğine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Çalışmıyor	**Değer**	580	8459	7413	1908	684
	%	**3,05**	**44,42**	**38,92**	**10,02**	**3,59**
Emekli	**Değer**	12	305	619	265	92
	%	**0,92**	**23,59**	**47,87**	**20,5**	**7,12**
İşçi	**Değer**	10	187	239	85	38
	%	**1,78**	**33,45**	**42,76**	**15,22**	**6,79**
Memur	**Değer**	1	49	180	160	120
	%	**0,19**	**9,61**	**35,29**	**31,38**	**23,53**
Özel Sektör	**Değer**	8	180	146	33	10
	%	**2,12**	**47,74**	**38,72**	**8,76**	**2,66**

Çizelgeden ve grafikten görüleceği üzere, annesi memur olan öğrencilerin başarı oranı en yüksektir. Bu oranı sırasıyla emekli ve işçi olanların başarısı takip etmektedir. Annesi çalışmayan veya özel sektörde olan öğrencilerin başarı oranı ise en düşüktür.

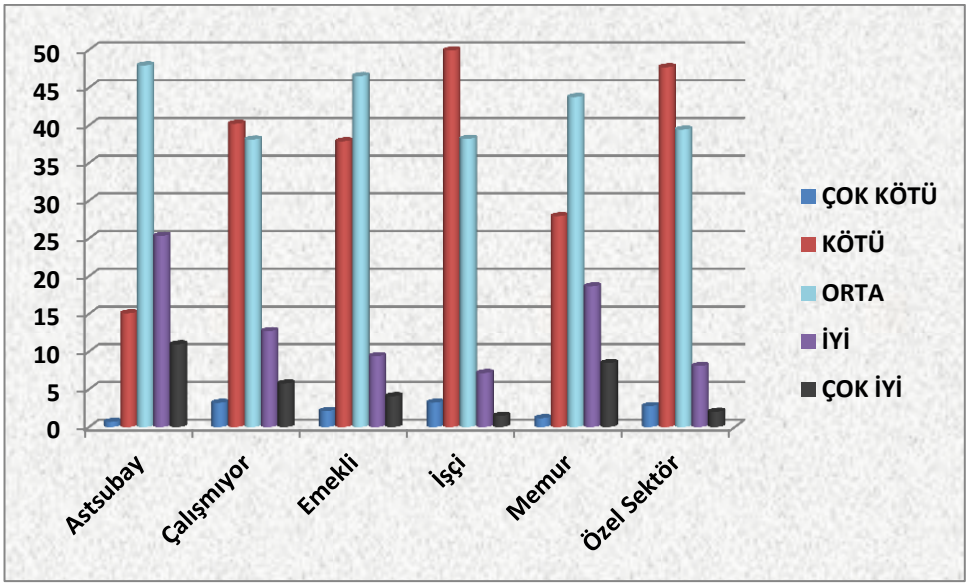


Şekil 5.11. OGES puanlarının öğrencinin annesinin mesleğine göre dağılım grafiği.

Öğrencilerin babasının mesleğinin öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.19'da görüldüğü gibidir. Baba mesleği değişkenine göre OGES puan dağılımı Şekil 5.12'de gösterildiği gibidir.

Çizelge 5.19. OGES puanlarının öğrencilerin babasının mesleğine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Astsubay	**Değer**	1	22	70	37	16
	%	**0,68**	**15,07**	**47,94**	**25,34**	**10,97**
Çalışmıyor	**Değer**	257	3226	3059	1021	463
	%	**3,21**	**40,19**	**38,11**	**12,72**	**5,77**
Emekli	**Değer**	14	250	307	62	27
	%	**2,12**	**37,88**	**46,52**	**9,39**	**4,09**
İşçi	**Değer**	237	3666	2806	525	109
	%	**3,23**	**49,92**	**38,21**	**7,15**	**1,49**
Memur	**Değer**	39	932	1459	622	283
	%	**1,17**	**27,95**	**43,75**	**18,65**	**8,48**
Özel Sektör	**Değer**	63	1084	896	184	46
	%	**2,77**	**47,69**	**39,42**	**8,09**	**2,03**



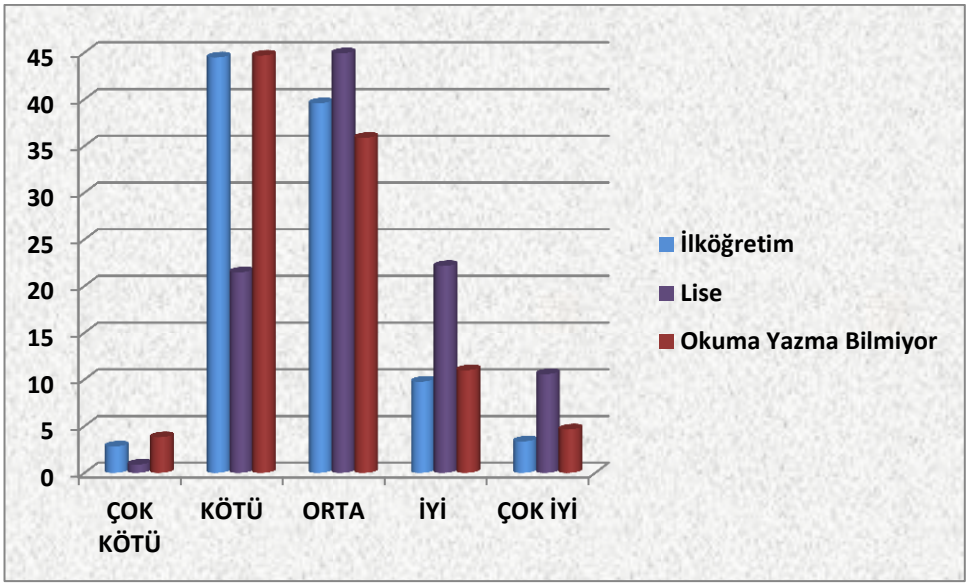
Şekil 5.12. OGES puanlarının öğrencinin babasının mesleğine göre dağılım grafiği.

Çizelgeden ve grafikten görüleceği üzere babası astsubay olan öğrencilerin başarı oranı en yüksektir. Bu oranı sırasıyla memur ve çalışmayanların başarısı takip etmektedir. Babası emekli, işçi ve özel sektörde olan öğrencilerin başarı oranı ise en düşüktür.

Öğrencilerin annesinin eğitim düzeyinin öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.20'de görüldüğü gibidir. Anne eğitim düzeyi değişkenine göre OGES puan dağılımı Şekil 5.13'de gösterildiği gibidir.

Çizelge 5.20. OGES puanlarının öğrencinin annesinin eğitim düzeyine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
İlköğretim	**Değer**	444	6978	6208	1531	528
	%	**2,83**	**44,47**	**39,57**	**9,76**	**3,37**
Lise	**Değer**	20	482	1008	497	237
	%	**0,89**	**21,47**	**44,92**	**22,15**	**10,57**
Okuma Yazma Bilmiyor	**Değer**	147	1720	1381	423	179
	%	**3,82**	**44,67**	**35,87**	**10,99**	**4,65**



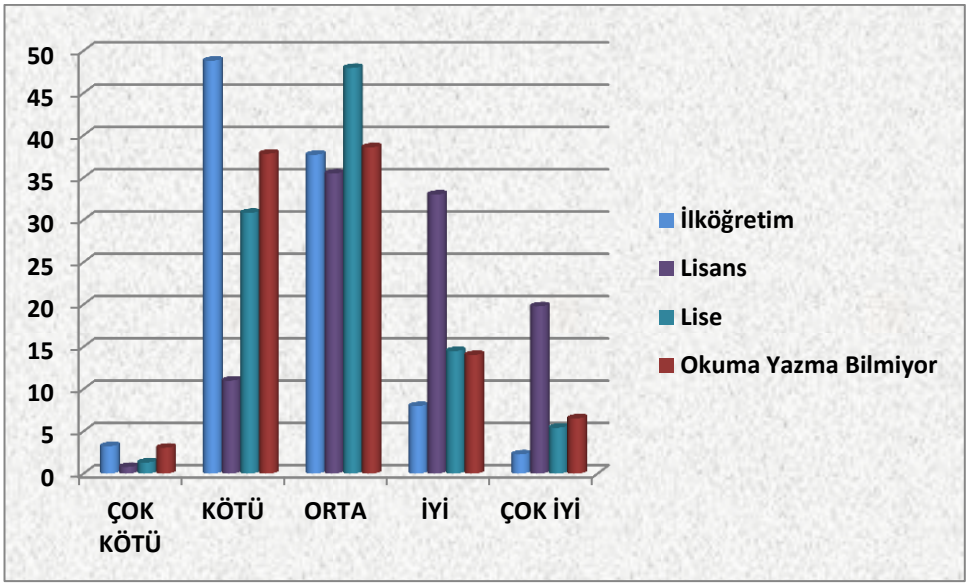
Şekil 5.13. OGES puanlarının öğrencinin annesinin eğitimine göre dağılım grafiği.

Çizelgeden ve grafikten görüleceği üzere annesi lise mezunu olan öğrencilerin başarı oranı, ilköğretim mezunu veya okuma yazma bilmeyenlere göre daha yüksektir.

Öğrencilerin babasının eğitim düzeyinin öğrenci başarısı üzerinde etkisinin dağılımı Çizelge 5.21'de görüldüğü gibidir. Baba eğitim düzeyi değişkenine göre OGES puan dağılımı Şekil 5.14'de gösterildiği gibidir.

Çizelge 5.21. OGES puanlarının öğrencinin babasının eğitim düzeyine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
İlköğretim	**Değer**	411	6182	4773	1011	287
	%	**3,24**	**48,82**	**37,69**	**7,98**	**2,27**
Lisans	**Değer**	6	84	271	252	151
	%	**0,78**	**10,99**	**35,47**	**32,99**	**19,77**
Lise	**Değer**	46	1082	1684	509	190
	%	**1,31**	**30,82**	**47,96**	**14,49**	**5,42**
Okuma Yazma Bilmiyor	**Değer**	148	1832	1869	679	316
	%	**3,05**	**37,82**	**38,58**	**14,02**	**6,53**



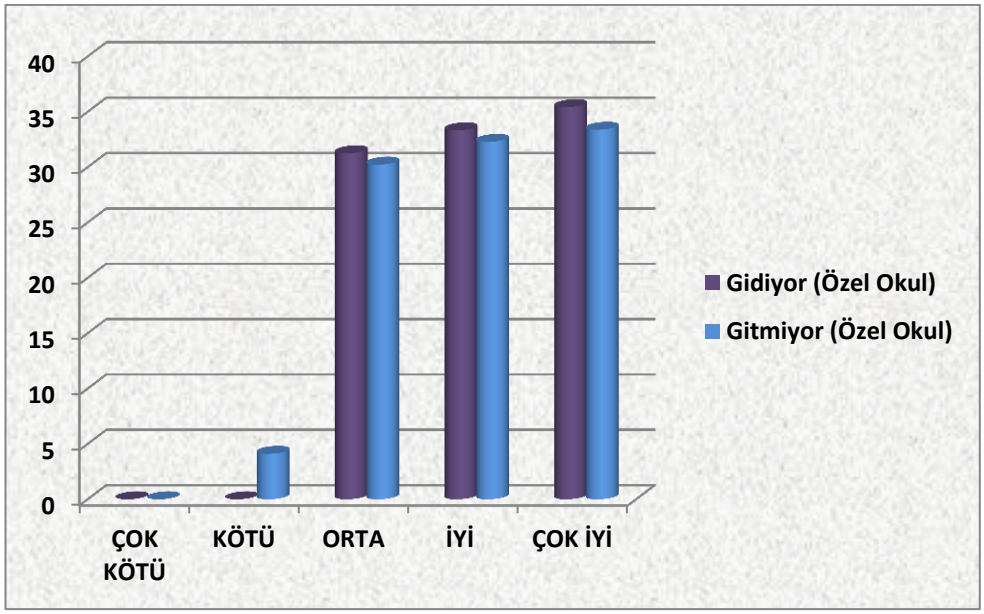
Şekil 5.14. OGES puanlarının öğrencinin babasının eğitimine göre dağılım grafiği.

Çizelgeden ve grafikten görüleceği üzere babası lisans mezunu olan öğrencilerin başarı oranı en yüksektir. Bu oranı babası lise mezunu öğrencilerin başarısı takip etmektedir. Babası ilköğretim mezunu veya okuma yazma bilmeyenler ise en düşük başarı oranına sahiptir.

Kurum türü ve özel okul değişkeni birlikte incelendiğinde öğrenci başarısı üzerindeki etkisinin dağılımı Çizelge 5.22, 5.23'de görüldüğü gibidir. Bu iki değişkene göre OGES puan dağılımı Şekil 5.15, 5.16'da gösterildiği gibidir.

Çizelge 5.22. OGES puanlarının özel okula giden öğrencinin dershaneye gitmesine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Gidiyor (Özel Okul)	**Değer**	0	0	15	16	17
	%	**0**	**0**	**31,25**	**33,33**	**35,42**
Gitmiyor (Özel Okul)	**Değer**	0	26	192	205	212
	%	**0**	**4,11**	**30,23**	**32,28**	**33,38**

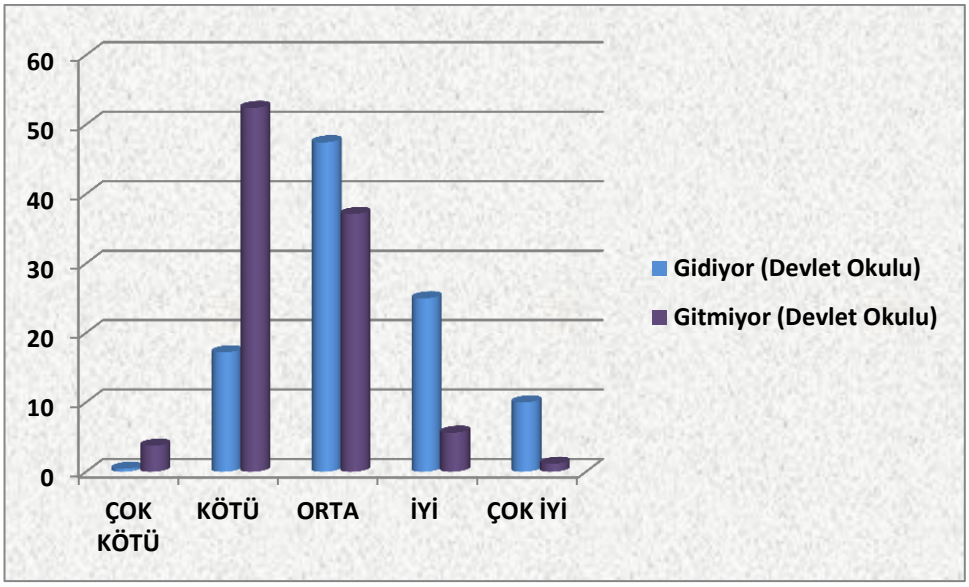


Şekil 5.15. OGES puanlarının özel okulda öğrenim gören öğrencinin dershaneye gidip gitmemesi ile birlikte dağılım grafiği.

Çizelgeden ve grafikten görüleceği üzere özel okulda okuyan öğrenciler eğer özel bir dershaneden de faydalanıyorsa daha başarılılardır. Fakat bu başarı oranı dershaneye gitmeyenlere göre çok büyük bir fark değildir.

Çizelge 5.23. OGES puanlarının devlet okuluna giden öğrencinin dershaneye gidip gitmemesine göre dağılımı.

		ÇOK KÖTÜ	KÖTÜ	ORTA	İYİ	ÇOK İYİ
Gidiyor (Devlet Okulu)	**Değer**	23	928	2561	1348	538
	%	**0,42**	**17,19**	**47,44**	**24,97**	**9,98**
Gitmiyor (Devlet Okulu)	**Değer**	588	8226	5827	882	179
	%	**3,74**	**52,39**	**37,11**	**5,62**	**1,14**



Şekil 5.16. OGES puanlarının özel okulda öğrenim gören öğrencinin dershaneye gidip gitmemesi ile birlikte dağılım grafiği.

Çizelgeden ve grafikten görüleceği üzere devlet okulunda okuyan öğrenciler eğer özel bir dershaneden de faydalanıyorsa, dershaneye gitmeyenlere göre daha başarılıdır.

Diğer değişkenler incelendiğinde öğrenci başarısının sınavda sorusu bulunan derslerle ve sınav puanları ile doğru orantılı olduğu görülmüştür. Ders notları ve sınav puanları yüksek olan öğrencinin OGES puanı da yüksektir.

Bu çalışma, öğrenci velileri açısından büyük önem taşıyan OGES'te hangi özelliklere sahip öğrencilerin başarılı olduğunun ortaya çıkarılmasının öğrenciler, veliler, öğretmenler, yöneticiler ve araştırmacılar için yararlı olacağı düşünülmektedir. Ayrıca uygulamada kullanılan verilerin Türkiye genelinden rastgele seçilmesi ile de belli bir bölgeye ait özelliklerin değil ülke genelinde tüm öğrencilerin karşılaştırma işlemi yapılmıştır. Çalışma bu açıdan da önemli görülmektedir.

KAYNAKLAR

1. Attaway, N.M. and Bry, B.H., "Parenting style and black adolescents' academic achievement", ***Journal of Black Psycholog***, 30: 229-247 (2004).

2. Steinberg, L., Lamborn, S.D., Darling, N., Mounts, N.S. and Dornbusch, S.M., "Overtime changes in adjustment and competence among adolescents from authoritative, authoritarian, indulgent, and neglectful families", ***Child Development***, 63: 754–770 (1994).

3. Goddard, R.D., Sweetland, S.R. and Hoy, W.K., "Academic emphasis of urban elementary schools and student achievement in reading and mathematics: A multilevel analysis", ***Educational Administration Quarterly***, 36 (5): 683–702 (2000).

4. Carpenter, P., "Type of school and academic achievement", ***Journal of Sociology***, 21 (2): 219–236 (1985).

5. Gerber, S.B. and Fin, J.D., "Teacher aides and students' academic achievement", ***Educational Evaluation and Policy Analysis***, 23 (2): 123–143 (2001).

6. Stipek, D.J., "Perceived personal control and academic achievement", ***Review of Educational Research,*** 51 (1): 101–137 (1981).

7. Gentilucci, J.L., "Principals' influence on academic achievement: the student perspective", ***NASSP Bulletin,*** 91 (3): 219–236 (2007).

8. Kelly, K., "The relation of gender and academic achievement to career self-efficacy and interests", ***Gifted Child Quarterly***, 37 (2): 59–64 (1993).

9. Bain, H.C., Boersma, F.J. and Chapman, J.W., "Academic achievement and locus of control in father-absent elementary school children", ***School Psychology International***, 4 (2): 69–78 (1983).

10. Yenilmez, K. and Duman, A., "Interviewing with students about the factors that affect the achievement of mathematic in primary school", ***Sosyal Bilimler Dergisi***, 19: 251–268 (2008).

11. Davis-Kean, P.E., "The influence of parent education and family income on child achievement: the indirect role of parental expectations and the home environment" ***Journal of Family Psychology***, 19 (2): 294–304 (2005).

12. Minaei-Bidgoli, B., Kashy, D.A., Kortmeyer, G. and Punch, W.F., "Predicting student performance: an application of data mining methods with an educational web-based system", ***In the Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference***, Boulder, CO, 13-18 (2003).

13. Boaler, J., "High-stakes testing: when learning no longer matters: standardized testing and the creation of inequality", ***Phi Delta Kappan***, 84 (7): 502-506 (2003).

14. Suttle, B., "Toward an integral approach to standardized testing: using the integral model to improve test performance and evaluate current testing methodologies", ***Journal of Integral Theory & Practice*** 5 (2): 31-53 (2010).

15. Flawley W.J., Piatetsky-Shapiro G. and Matheus C.J., "Knowledge discovery in databases : an overview", ***AI Magazine***, 13 (3): 57-70 (1992).

16. Adrians, P. and Zantige, D., "Data Mining", ***Addison Wesley Publishing***, 1-9 (1996).

17. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. and Zanasi, A., "Discovering Data Mining: From concept to implementation", ***Prentice Hall Publishing***, 195 (1998).

18. Han, J. and Kamber M.,"Data Mining: Concepts and Techniques", ***Morgan Kaufmann Publishing***, 5-9 (2001).

19. Mohammadian, M., "Intelligent Agents for Data Mining and Information Retrieval", ***Idea Group Publishing***, 15-30 (2004).

20. Soukup, T. and Davidson, I., "Visual Data Mining: Techniques and Tools for Data Visualization and Mining", ***Wiley Computer Publishing***, 25-27 (2006).

21. Keim, D.A., "Visual Techniques for Exploring Databases", ***University of Halle Wittenberg***, 2-9 (2004).

22. Venkayala, S., Using java data mining to develop advanced analytics applications, ***Java Developers Journal***, 1-3 (2005).

23. Wang, J., "Data Mining: Opportunities and Challenges", ***Idea Group Publishing***, 174-189 (2003).

24. Wang, J., "Encyclopedia of Data Warehousing and Mining", ***Idea Group Publishing***, 1171-1174 (2006).

25. Tang, Z.and MacLennan, J., "Data Mining with SQL Server 2005", **Wiley Computer Publishing**, 2-6 (2005).

26. Weiss, G., Saar-Tsechansky and M., Zadrozny, B., "First International Workshop on Utility-Based Data Mining", ***Association for Computing Machinery Publishing***, 69-77 (2005).

27. Mattison, R, "Data Warehousing and Data Mining for Telecommunication", ***Artech House Publishing***, 18-20 (1997).

28. Keogh, E., Lonardi, S. and Ratanamahatana, C.A., "Towards Parameter-Free Data Mining", ***ACM Publishing***, 206-215 (2004).

29. Mitra, S., and Acharya T., "Data Mining: Multimedia, Soft Computing and Bioinformatics", ***Wiley Computer Publishing***, 1-5 (2003).

30. Berry, M.J.A. and Linoff, G.S., "Data Mining Techniques For Marketing, Sales and Customer Relationship Management 2nd ed", ***Wiley Publishing***, 1 (2004).

31. Witten, I.H. and Frank, E., "Data Mining, Practical Machine Learning Tools and Techniques 2nd ed", ***Elsevier Press***, 9 (2005).

32. Swift R., "Accelerating Customer Relationship", ***Prentice Hall PTR***, 93 (2001).

33. Giudici P., "Applied Data Mining: Statistical Methods for Business and Industry", ***John Wiley & Sons***, 2 (2003).

34. Özmen Ş., "İş hayatı veri madenciliği ile istatistik uygulamalarını yeniden keşfediyor", ***V. Ulusal Ekonometri ve İstatistik Sempozyumu***, Adana, 1-4 (2001).

35. Shearer C., "The CRISP-DM model: the new blueprint for data mining", ***Journal of Data Warehousing***, 5 (4): 13-22, (2000).

36. Çakır Ö., " Veri madenciliğinde sınıflandırma yöntemlerinin karşılaştırılması – bankacılık müşteri veritabanı üzerinde bir uygulama", Doktora Tezi, ***Marmara Üniversitesi, Sosyal Bilimler Enstitüsü***, İstanbul, 17-19 (2008).

37. Hegland M., "Data Mining Techniques", ***Acta Numerica***, 313-355, (2001).

38. Tiryaki S., "Lojistik alanında bir veri madenciliği uygulaması", Yüksek Lisans Tezi, ***İstanbul Teknik Üniversitesi, Fen Bilmleri Enstitüsü***, İstanbul, 25 (2006).

39. Zahn, E., "Informations technologie und Informations management", ***München,*** 300-357 (1997).

40. Berson, A., Smith, S. and Thearling, K., "Building Data Mining Applications for CRM", ***McGraw-Hill Professional Publishing***, New York, USA, 33-36 (2000).

41. Eker, H., "Veri Madenciliği Veya Bilgi Keşfi", **http://www.bilgiyonetimi.org/cm/** (2010).

42. Hair J.F., Anderson R.E., Tatham R.L. and Black W.C., "Multivariate Data Analysis", ***Prentice Hall***, U.S.A., 473 (1998).

43. Akpınar H., "Veri tabanlarında bilgi keşfi ve veri madenciliği", **İstanbul Üniversitesi İşletme Fakültesi Dergisi**, 1: 6 (2000).

44. Araya, S., Silva, M. and Weber, R., "A methodology for web usage mining and ıts application to target group identification", ***Elsevier B.V.***, Fuzzy Sets and System, 103-107 (2004).

45. Öztemel, E., "Yapay Sinir Ağları", ***Papatya Yayıncılık***, İstanbul, 13-113 (2003).

46. Hasan Y., "Yapay sinir ağları metodolojisi ile öngörü modellemesi: bazı makroekonomik değişkenler için Türkiye örneği", **Devlet Planlama Teşkilatı Uzmanlık Tezi**, 102, (2005).

47. Elmas, Ç., "Yapay Sinir Ağları"**,** Ankara, ***SeçkinYayınevi***, 27-29 (2003).

48. McCullogh, WS., and Pitts, W., "A logical calculus of ideas imminent in nervous activity", ***Bull. Math. Biophysics***, 5: 115-33 (1943).

49. Saraç, T., "Yapay sinir ağları", ***Seminer Projesi***, Ankara, 22-75 (2004).

50. Internet: Kakıcı A., Yapay Zeka "Yapay Sinir Ağları", **http://www.ahmetkakici.com/yapay-sinir-aglari/yapay-sinir-aglarinin-mimarisi-ve-yapi-elemanlari/,** (2012).

51. Aynekin G., "İnternet içerik madenciliğinde yapay sinir ağları ve bir uygulama", Yüksek Lisans Tezi, ***Uludağ Üniversitesi, Fen Bilimleri Enstitüsü***, Bursa, 63 (2006).

52. Larose D.T., "Discovering Knowledge in Data: An Introduction to Data Mining", ***John Wiley & Sons***, 41-66 (2005).

53. Haykin S., "Neural Networks: A Comprehensive Foundation", ***Prentice Hall***, 138, (1994).

54. Tou, Julius T. and Rafael C. Gonzalez., "Pattern Recognition Principles", ***Addison-Wesley Publishing Co***, 173-181 (1974).

55. Sağıroğlu, S. Beşdok, E. ve Erler, M., "Mühendislikte Yapay Zekâ Uygulamaları-1: Yapay Sinir Ağları", ***Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd. Sti***, 55-58 (2003).

56. Demuth, H. Beale, M. and Hagan, M., "Neural Network Toolbox User's Guide for Use with MATLAB", ***The MathWorks Inc***., 2-10 (2006).

57. Şenol C. Yıldırım T., "Standart ve hibrid yapılar kullanarak yapay sinir ağları ile imza tanıma", ***Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu-ELECO***, Bursa, 261-265 (2004).

58. Haykin, S., "Neural Networks", ***Prentice Hall***, New Jersey 25-32 (1999).

59. Wang, L., "Support Vector Machines: Theory and Application, Studies in Fuzziness and Soft Computing", **Springer**, 177: (2005).

60. Abe, S. "Support Vector Machine for Pattern Classification", ***Springer***, 343 (2005).

61. Walgampaya C. K., "Cost-benefit analysis in multiple time series prediction". Master of Science Thesis. ***Department of Computer Engineering and Computer Science, University of Lousville***, Kentucky, 36 (2006).

62. Sravanthi A.K., "Edited- bootstrapped support vector machines for one-class data classification", ***Bachelor of Technology in Electronics and Communication Engineering, Nagarjuna University***, Guntur, India, 16-23 (2003).

63. Sezer, O.G., Erçil A, and Keskinöz M., "Independent component based 3d object recognition using support vector machines", ***Proceeding of the IEEE 13th Signal Processing and Communications Applications Conference***, IEEE, 99-102 (2005).

64. Christopher J.C. Burges, "A tutorial on support vector machines for pattern recognition, data mining and knowledge discovery", ***Kluwer Academic Publishers***, 2 (2) : 121-167 (1998).

65. Kecman V., "Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models". ***MIT Press***, Cambridge, Massachusetts, England, 148-176 (2001).

66. Kabaoğlu O., R., "Destek vektör makineleri tabanlı hata bulma, tanıma ve hata toleranslı kontrol yöntemleri", Doktora Tezi, ***İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü***, 63 (2010).

67. Özkan Y., "Veri Madenciliği Yöntemleri", ***Papatya Yayıncılık Eğitim A.Ş***. 195-199 (2008).

68. Aydoğan Ü., "Destek vektör makinelerinde kullanılan çekirdek fonksiyonların sınıflama performanslarının karşılaştırılması", Yüksek Lisans Tezi, ***Hacettepe Üniversitesi, Sağlık Bilimleri Enstitüsü***, 37 (2010).

69. Tan P. N., Steinbach M. and Kumar V., "Introduction to Data Mining", ***Pearson Education***, 146 (2006).

70. Weston J. and Watkins, C., "Support vector machines for multi-class pattern recognition", ***Proceedings of the Seventh European Symposium on Artificial Neural Networks***, Bruges, 219-224 (1999).

71. Cristianini, N. and Taylor J.S., "An Introduction to support vector machines and other kernel-based learning methods", ***Cambridge University Press***, New York, 93-112 (2000).

72. Vapnik, V.N., "Statistical Learning Theory", ***John Wiley & Sons***, New York, 16-20 (1998).

73. Safavian S.R. and Landgrebe D., "A survey of decision tree classifier methodology", ***IEEE Transactions on Systems Man and Cybernetics***, 21: 660-674 (1991).

74. Olafsson, S, Xiaonan L. and Shuning W., "Operations research and data mining," ***European Journal of Operational Research***, 187 (3): 1429- 1448 (2008).

75. Aitkenhead, M. J. "A co-evolving decision tree classification," ***Expert Systems with Applications***, 34 (1): 18-25 (2008).

76. Morgan, J., N. and Sonquist J. A., "Problems in the analysis of survey data, and a proposal", ***J.Amer. Statist. Assoc***., 58: 415-434 (1963).

77. Mitchell, T.M., "Machine Learning", ***McGraw-Hill***, 52-55 (1997).

78. Sun, Jie and Hui Li, "Data mining method for listed companies, financial distress prediction", ***Knowledge-Based Systems***, 21 (1): 1-5 (2008).

79. Berry., M. and Linoff., G., "Data Mining Techniques: For Marketing, Sales, And Customer Relationship Management", ***Wiley***, 237-240 (1997).

80. Breiman, L., Freidman, J.H., Olshen, R.A. and Stone, C.J., "Classification And Regression Trees", ***Chapman&Hall***, 18-55 (1998).

81. Thomas, Lyn. C., "A survey of credit and behavioral scoring: forecasting financial risk of lending to consumer", ***International Journal of Forecasting***, 16 (2): 149-172 (2000).

82. Kayri, M. and Boysan, M., "Assesment of relation between cognitive vulnerability and depression's level by using classification and regression tree analysis", ***Hacettepe University Journal of Education***, 34: 168-177 (2008).

83. Kuzey C., "Veri madenciliğinde destek vektör makineleri ve karar ağaçları yöntemlerini kullanarak bilgi çalışanlarının kurum performansı üzerine etkisinin ölçülmesi ve bir uygulama", Doktora Tezi, ***İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü,*** 51 (2012).

84. Han J. and Kamber M., "Data Mining : Concepts and Techniques", ***Academic Press***, 285-294 (2001).

85. Two Crows Co., "Introduction to Data Mining and Knowledge Discovery 3rd Edition" ***Two Crows Corporation***, USA, 19-23 (2005).

86. Koza, J., "Data mining using grammar based geneticprogramming and applications", ***Kluwer Academic Publishers***, 9-16 (2002).

87. Albayrak, A.S. ve Yılmaz, K., "Veri madenciliği: karar ağacı algoritmaları ve imkb verileri üzerine bir uygulama", ***Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi***, 14 (1): 31-52 (2009).

88. Maimon O. and Rokach, L., "Data Mining And Knowledge Discovery Handbook", ***Springer***, 165-185 (2005).

89. Myatt, G.J. and Johnson W.,P., "Making Sense of Data II, A Practical Guide to Data Visualization, Advanced Data Mining Methods and Applications", ***Wiley***, 271-291 (2007).

90. Bounsaythip, C. ve Esa, R. R., "Overview of data mining for customer behavior modeling", ***VTT Information Technology Research Report,*** 1: 1-53 (2001).

91. Dondurmacı G., "Veri madenciliğinde regresyon ağaçları ile sınıflandırma ve bir uygulama", Doktora Tezi, ***Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Bilimleri Enstitüsü***, İstanbul 67 (2011).

92. Vercellis, C., "Business intelligence: Data Mining and Optimization for Decision Making", ***Wiley***, 23-36(2009).

93. Mackay, D.J.C., "Information Theory, Inference, and Learning Algorithms", ***Cambridge University Press***, 15-27 (2003).

94. Duda, R.O., Hart, P.E. and Stork, D.G., "Pattern Classification", ***Wiley***, 395 (2000).

95. Kocabaş, Ş., "A review of learning", The Knowledge Engineering Review, ***Cambridge University Press***, 6 (3): 195-222.

96. Quinlan, J.R., "C 4.5: Programs for Machine Learning"***, Los Altos, Morgan Kaufmann***, 71-80 (1993).

97. Pal M., Mather P.M., "An assessment of the effectiveness of decision tree methods for land cover classification", ***Remote Sensing of Environment***, 86, 554-565 (2003).

98. Cornfield, J., "Joint dependence of the risk of coronary heart disease on serum cholesterol and systolic blood pressure: a discriminate function analysis", ***Federation Proceedings***, 21, 58-61, (1962).

99. Lee, C.T., "Logistic models for cross-over designs", ***Biometrika***, 71, 216-217, (1984).

100. Bonney, G.E., "Logistic regression for dependent binary observations", ***Biometrics***, 43, 951-973, (1987).

101. Roberts, G., Rao, J.N.K. and Kumar, S., "Logistic regression analysis of sample survey data", ***Biometrika***, 74 (1): 1-12, (1987).

102. Duffy, D., "On continuity corrected residuals in logistic regression", ***Biometrika***, 77, 287-293 (1990).

103. Ulupınar, S.D., "2001 krizi öncesi ve sonrası türk ticari bankalarının karlılıklarının lojistik regresyon analizi ile incelenmesi", Yüksek Lisans Tezi, ***Marmara Üniversitesi, Sosyal Bilimler Enstitüsü***, İstanbul, 39 (2007).

104. Tabachnick, B.G. ve Fidell, L.S., "Using Multivariate Statistics Fourth Edition", ***Pearson Education Company***, USA, 131-138 (2001).

105. Kaşko, Y., "Çoklu bağlantı durumunda ikili (binary) lojistik regresyon modelinde gerçekleşen 1. tip hata ve testin gücü", Yüksek Lisans Tezi, ***Ankara Üniversitesi, Fen Bilimleri Enstitüsü***, 19 (2007).

106. Düzgüneş, O., Kesici, T., Kavuncu, O. ve Gürbüz, F., "Araştırma ve deneme metotları", ***Ankara Üniversitesi Ziraat Fakültesi Yayınları***, Ankara, 344 (1987).

107. Albayrak, A.S., "Uygulamalı Çok Değişkenli İstatistik Teknikleri", ***Asil Yayın Dağıtım A.Ş.***, Ankara, 273-299 (2006).

108. Bircan, H., "Lojistik regresyon analizi: tıp verileri üzerine bir uygulama", ***Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi***, 2: 186, 187, 189, 197 (2004).

109. Özdamar, K., "Paket Programlar İle İstatistiksel Veri Analizi", ***Kaan Kitabevi***, Eskişehir, 625 - 627 (2002).

110. Hosmer, D.W. and Lemeshow, S., "Applied logistic regression", ***Wiley Series In Probability And Statisytics***, Canada, 110-111 (2000).

111. Başarır, G., "Çok değişkenli verilerde ayrımsama sorunu ve lojistik regresyon analizi", Doktora Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 17 (1990).

112. Ryan, T.P. "Modern Regression Analysis", ***John Wiley&Sons Inc***, USA, 19 (1997).

113. Allison, P. D., "Logistic Regression Using the SAS System: Theory and Application", ***SAS Institute Inc***, USA, 11-15 (2001).

114. Menard, S., "Applied Logistic Regression Analysis 2nd Edition", ***Sage Publications***, USA, 41-56 (2002).

115. Ürük, E., "İstatistiksel uygulamalarda lojistik regresyon analizi", Yüksek Lisans Tezi, ***Marmara Üniversitesi, Fen Bilimleri Enstitüsü***, İstanbul, 11 (2007).

116. Tatlıdil, H., "Uygulamalı Çok Değişkenli İstatistik Teknikleri", ***Cem Ofset Ltd.Şti.***, Ankara, 424 (1996).

117. Canyüker, E. ve Aşan, Z., "Parametrik Olmayan İstatistik Teknikler", ***Anadolu Üniversitesi Yayınları***, Eskişehir, 1-3 (2001).

118. Conover, W.J., "Practical Nonparametric Statistics", ***Wiley Publication***, 239 (1999).

119. Pryanishnikov, I. and Zigova, K., "Multinomial logit models for the Austrian labor market", ***Austrian Journal of the Statistics***, 32: 267-282. (2003).

120. Internet: Lo, H. and Lam, W.S.P., "A Modified Logit Model of Route Choice for Drivers Using the Transportation Information System" **http://www.iasi.cnr.it/ewgt/13conference/54_lo.pdf.** (2012).

121. Medina, S. and Ward, R.W., "A multinomial logit model of retail outlet selection for beef" ***International Food and Agribusiness Management Review***, 2 (2): 195–219 (1999).

122. Mesak, H. and Means, T.L., "On the appropriateness of multinomial logit market share models for equilibrium analyses of advertising competition" ***Proceedings of the Annual Meeting of the Decision Sciences Institute***, 409-411 (1998).

123. Internet: Abe, M., Boztug, Y. and Hildebrant, L. 2003. "Investigating the Competitive Assumption of Multinomial Logit Models of Brand Choice by Nonparametric Modeling" **http://www.cirje.e.utokyo.ac.jp/research/dp/2003/2003cf193.pdf,** (2003).

124. Fujimoto, K., "Application of multinomial and ordinal regression to the data of japanese female labor market", ***University of Pittsburgh***, 20- 23 (2003).

125. Van Campen, C. and Woittiez, I.B., "Client demands and the allocation of home care in the netherlands a multinomial logit model of client types, care needs and referrals", ***Health Policy***, 64 (2): 229-241 (2003).

126. Porell, F.W. and Miltiades, H.B., "Regional differences in functional status among the aged", ***Social Sciences&Medicine***, 54: 1181-1198 (2002).

127. Zweig, J.M. and Lindberg, L.D., "Predicting adolescent profiles of risk: looking beyond demographics", ***Journal of the Adolescent Health***, 31: 343-353 (2002).

128. Agresti, A., "Categorical Data Analysis", ***John Wiley&Sons***, USA, 117-139 (1990).

129. Şen B, Uçar E. and Delen D. "Predicting and analyzing secondary education placement-test scores: A data mining approach", ***Expert Systems With Applications,*** 39 (10): 9468-9476 (2012).

130. Uçar E., Şen B. and Bayır Ş. "Placement score estimation of secondary education transition system (SETS) using artificial neural networks", ***Energy Education Science and Technology Part A: Energy Science and Research***, 30 (2): 749-758 (2013).

ÖZGEÇMİŞ

Emine UÇAR 1982 yılında Ankara'da doğdu; ilk ve orta öğrenimini aynı şehirde tamamladı. Gazi Anadolu Teknik, Teknik ve Endüstri Meslek Lisesi Bilgisayar Bölümü'nden mezun oldu. 2000 yılında Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik Bilgisayar Eğitimi Bölümü, Bilgisayar Sistemleri Öğretmenliği Anabilimdalı'nda öğrenime başlayıp 2004 yılında iyi derece ile mezun oldu ve aynı yıl Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik-Bilgisayar Eğitimi Bölümü'nde yüksek lisans programına başladı. 2008-2009 eğitim öğretim yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilimdalı'nda başlamış olduğu doktora programını 2013 yılında tamamladı.

2004 yılında Kastamonu Tosya Atabinen Kız Meslek Lisesi'nde öğretmen olarak göreve başladı. 2006 yılında Milli Eğitim Bakanlığı Eğitim Teknolojileri Genel Müdürlüğü Bilişim Hizmetleri Dairesi'nde çalışmaya başladı. 2011 yılında M.E.B. Bilgi İşlem Grup Başkanlığı Yönetim Bilgi Sistemleri Şubesi'ne görevlendirildi ve halen bu göreve devam etmektedir.

ADRES BİLGİLERİ

Adres : Milli Eğitim Bakanlığı

Bilgi İşlem Grup Başkanlığı

Zemin Kat B. Blok Bakanlıklar /ANKARA

Tel : (505) 396 33 93

E-posta : emineucar@meb.gov.tr