



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



MAKİNE ÖĞRENMESİ TEKNİKLERİYLE KREDİ RİSK ANALİZİ

EMİNE BAHÇE ÇİZER

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik
Mühendisliği Anabilim Dalı
Elektrik-Elektronik
Mühendisliği Programı

DANIŞMAN

Yrd. Doç. Dr. Ayça AK

EŞ-DANIŞMAN

Doç. Dr. Vedat TOPUZ

İSTANBUL, 2018



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



MAKİNE ÖĞRENMESİ TEKNİKLERİYLE KREDİ RİSK ANALİZİ

EMİNE BAHÇE ÇİZER

523113007

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik
Mühendisliği Anabilim Dalı
Elektrik-Elektronik
Mühendisliği Programı

DANIŞMAN

Yrd. Doç. Dr. Ayça AK

EŞ-DANIŞMAN

Doç. Dr. Vedat TOPUZ

İSTANBUL, 2018

MARMARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Emine BAHÇE ÇİZER'in "Makine Öğrenmesi Teknikleriyle Kredi Risk Analizi" başlıklı tez çalışması, 30 Ocak 2018 tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Yrd. Doç. Dr. Ayça Ak (Danışman)

Marmara Üniversitesi(İMZA) 

Doç. Dr. Serhat ÖZEKES (Üye)

Üsküdar Üniversitesi(İMZA) 

Yrd. Doç. Dr. Buket DOĞAN (Üye)

Marmara Üniversitesi(İMZA) 

ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 05.06.2018 tarih ve 2018/06-56 sayılı kararı ile Emine BAHÇE ÇİZER'in Elektrik Elektronik Mühendisliği Anabilim Dalı Elektrik Elektronik Mühendisliği Programında Yüksek Lisans derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Bülent ERGİÇİ



TEŐEKKÜR

Tez konumu belirlememde ve tezimi tamamlamamda benden desteklerini esirgemeyen deęerli hocalarım Yrd. Doę. Dr. Ayęa Ak ve Doę. Dr. Vedat Topuz' a bana ayırdıkları deęerli zamanları ve yardımları için minnettarım.

Tezimin hazırlanması sırasında benden maddi ve manevi desteęini esirgemeyen aileme ve eőime teőekkürü bir borę bilirim.

Aralık 2017

Emine BAHE IZER

ABSTRACT

CREDIT RISK ANALYSIS WITH MACHINE LEARNING TECHNIQUES

In the first stage of this study it was targeted that customers who applied for a loan request should determine the factors affecting the repayment status of the loan requests by making objective decision. Therefore, to be able to make credit risk analysis with machine learning techniques, first decision tree analysis, support vector machine analysis, fuzzy logic and genetic algorithm analysis were applied, then lastly neural network analysis was applied. In the second stage of this study, the results of the analyzes made in the first stage and the estimated percentages were compared. Within the results of the analyzes made, it was observed that the method that classifies the test data at the highest rate with 76% accuracy is the artificial neural network method. With this purpose, data set is used in the analysis downloaded from UCI Machine Learning Repository* open access web site. In these data set 1000 customers have been used. Among the customers who were used, the ones with the good creditability area were good coded as 1 and the ones with the bad creditability area were coded as 0. Analysis performed in this model has one independent variable and 20 dependent variables.

Key words: Machine learning, data mining, decision trees, exchaid analysis, credit analysis, svm analysis, fuzzy logic, genetic algorithm, neural network analysis.

* [https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data))

ÖZET

MAKİNE ÖĞRENMESİ TEKNİKLERİYLE KREDİ RİSK ANALİZİ

Çalışmanın ilk aşamasında kredi talebine başvuran müşterilerin, kredi taleplerinin değerlendirilmesinde geri ödeme durumunu etkileyen faktörlerin objektif kararlar verilerek belirlenmesi hedeflenmiştir. Bu doğrultuda, kredi risk analizinin makine öğrenmesi teknikleriyle yapılabilmesi için, öncelikle karar ağacı analizi, destek vektör makineleri analizi, bulanık mantık ve genetik algoritma analizi uygulanmıştır, son olarak da yapay sinir ağları analizi uygulanmıştır. Çalışmanın ikinci aşamasında ise ilk aşamada yapılmış olan analizlerin sonuçları ve tahmin yüzdeleri karşılaştırılmıştır. Yapılan analizlerin sonuçları içerisinde test verilerini %76 doğruluk oranı ile en yüksek oranda sınıflandıran yöntemin yapay sinir ağları yöntemi olduğu gözlemlenmiştir. Bu amaç doğrultusunda analizde kullanılan veri kümesi UCI Machine Learning Repository*'nin açık erişimde bulunan sitesinden indirilmiştir. Bu data kümesi 1000 adet müşteri verisini içermektedir. Kullanılan müşteriler arasında iyi kredilendirilebilir alanına sahip olanlar 1 olarak kodlanmıştır ve kötü kredilendirilebilir alana sahip olanlar da 0 olarak kodlanmıştır. Analiz yapılan modelde 1 adet bağımsız değişken ve 20 adet bağımlı değişken kullanılmıştır.

Anahtar Kelimeler: Makine öğrenmesi, veri madenciliği, karar ağaçları, exchaid analizi, kredi analizi, destek vektör analizi, bulanık mantık, genetik algoritma, yapay sinir ağları analizi.

* [https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data))

İÇİNDEKİLER

TEŞEKKÜR.....	ii
ABSTRACT.....	iii
CREDIT RISK ANALYSIS WITH MACHINE LEARNING TECHNIQUES.....	iii
ÖZET.....	iv
MAKİNE ÖĞRENMESİ TEKNİKLERİYLE KREDİ RİSK ANALİZİ	iv
SEMBOLLER.....	vii
ŞEKİL LİSTESİ.....	viii
TABLO LİSTESİ.....	ix
KISALTMALAR	x
1. GİRİŞ	1
2. YÖNTEM.....	6
2.1 MAKİNE ÖĞRENMESİ	6
2.1.1 VERİ MADENCİLİĞİ	7
2.2 BULANIK MANTIK.....	17
2.3 GENETİK ALGORİTMA	22
2.4 YAPAY SİNİR AĞLARI	24
3. BULGULAR.....	30
3.1 KULLANILAN VERİ KÜMESİ VE ALANLARI	30
3.2 VERİ KÜMESİNE KARAR AĞACI ANALİZİNİN UYGULANMASI.....	31
3.3 VERİ KÜMESİNE DESTEK VEKTÖR MAKİNELERİ VE TEMEL BİLEŞENLER ANALİZİNİN UYGULANMASI.....	35
3.4 VERİ KÜMESİNE BULANIK MANTIK VE GENETİK ALGORİTMA ANALİZİNİN UYGULANMASI.....	38
3.5 VERİ KÜMESİNE YAPAY SİNİR AĞI ANALİZİNİN UYGULANMASI	40
4. SONUÇ	44

KAYNAKLAR	50
EKLER.....	53
EK 1: Kullanılan Matlab Kodları.....	53
EK 2: Yapay Sinir Ağları Analiz Sonucu Tahmin Yüzdeleri	61
ÖZGEÇMİŞ	67

SEMBOLLER

- C** : Kovaryans matrisi
E : Entropi
g : Grup sayısı
 n_1 : i. gruptaki birim sayısı
 ζ_i : Hata vektörü
 \mathcal{L} : Lagrange deęişkeni
 μ : Veri ortalaması
 λ : Özdeęer
 $\mu_{A\sim}$: Bulanık küme üyelik fonksiyonu
 x : Girdi verisi
 \vec{x} : Vektör

ŞEKİL LİSTESİ

Şekil 2.1 Karar Ağacı örneği	9
Şekil 2.2 Budanmamış karar ağacı ve budanmış karar ağacı örneği	11
Şekil 2.3 SVM gösterimi	14
Şekil 2.4 Klasik küme üyelik fonksiyonu.....	19
Şekil 2.5 Bulanık küme üyelik fonksiyonu.....	19
Şekil 2.6 Bulanık mantık ile kümeleme örneği.....	20
Şekil 2.7 Üçgen üyelik fonksiyonu	20
Şekil 2.8 Yamuk üyelik fonksiyonu	21
Şekil 2.9 Gaussian üyelik fonksiyonu.....	21
Şekil 2.10 Çan üyelik fonksiyonu.....	22
Şekil 2.11 Bulanık mantık sistemi genel yapısı.....	22
Şekil 2.12 Ara katmanlara sahip örnek yapay sinir ağı yapısı.....	25
Şekil 2.13 Basit bir perceptron yapısı	25
Şekil 3.1 Karar ağacı analizi için kurulan modelin genel yapısı.....	33
Şekil 3.2 Analiz sonucu elde edilen ağaç yapısı.....	34
Şekil 3.3 Destek vektör analizi için kurulan modelin genel yapısı.....	36
Şekil 3.4 Temel Bileşenler analizi sonucu indirgenen değişkenler	37
Şekil 3.5 Eğitim verilerinin bulanık mantık ve genetik algoritma ile analizi sonucu.....	39
Şekil 3.6 Test verilerinin bulanık mantık ve genetik algoritma ile analizi sonucu.....	40
Şekil 3.7 Veri kümesinin weka' ya yüklenmesi	41

TABLO LİSTESİ

Tablo 2.1 Bulanık Mantık ve Klasik Mantık karşılaştırması	18
Tablo 3.1 Veri kümesi içerisindeki alan ve açıklamaları	31
Tablo 3.2 EXCHAID analizinin sonuçları	35
Tablo 3.3 Yapılan ilk Destek Vektör Makineleri analizinin sonucu	37
Tablo 3.4 Temel Bileşenler analizi ile indirgenen değişkenlerle yapılan Destek Vektör Makineleri analizi sonucu	38
Tablo 3.5 Eğitim verilerinin Bulanık Mantık ve Genetik Algoritma ile analizi sonucu	39
Tablo 3.6 Test verilerinin Bulanık Mantık ve Genetik Algoritma ile analizi sonucu	40
Tablo 3.7 Tüm veri setinin yapay sinir ağları analizi sonucu.....	42
Tablo 3.8 Bölümlendirilmiş veri setinin yapay sinir ağı analizi sonucu	43
Tablo 4.1 Eğitim verilerinin sınıflandırma sonuçları.	44
Tablo 4.2 Test verilerinin sınıflandırma sonuçları	44

KISALTMALAR

CHAID : Chi-squared Automatic Interaction Detection

EXCHAID : Exhaustive Chi-squared Automatic Interaction Detection

SPSS : Statical Package for Social Sciences

SVM : Support Vector Machines

1. GİRİŞ

Kredi için bankaya başvuran müşterilere kredi verme, bankaların en temel işlevlerinden biri olmakla beraber aynı zamanda bankaların en riskli faaliyet alanlarından biridir. Bankalar yapıları gereği risk alırlar ve aldıkları bu riskleri yönetirler. Bu nedenle, bankaların almış olduğu en önemli risklerden biri olan kredilendirme faaliyetlerini, verimli ve verilen kredilerin geri ödenmesinden doğabilecek zararları minimum düzeyde tutabilecek şekilde yönetmeleri gerekir. Tüm bu sebeplerden ötürü, bankaların son yıllarda kredilendirme faaliyetlerinde hızlı, ve sağlıklı kararlar verebilmesini sağlayan veri madenciliği, ayrıştırma analizi, logit ve probit modelleri, sınıflandırma ve regresyon araçları, bulanık mantık ve yapay sinir ağları gibi teknikleri daha yoğun bir şekilde kullanmaya başladıkları görülmektedir. Banka ve finans sektörlerinde risklerin doğru yönetilmesi, sahip olunan kaynakların etkin ve verimli kullanılması, potansiyel risklerin önceden tahmin edilerek gerekli önlemlerin bugünden alınmasına bağlıdır. Sorunlu kredilerin önceden tahmini bankalarda kararlılık ve kaynak verimliliği açısından büyük önem arz etmektedir.

Veri madenciliği büyük miktarlardaki verilerden anlamlı bilgilerin çıkarılması olarak tanımlanmaktadır. Veri madenciliği birçok uygulamada olduğu gibi risk analizinde de sıklıkla kullanılan bir yöntemdir. Finansal risk analizi kapsamında doğru ve etkin kredi kararı verebilme, geri ödemede sorun çıkaracak müşterileri belirleme, risk derecelendirme, finansal işlemlerde sahtekârlığa yönelik eğilimleri izleme, ekonomik ve finansal yatırımları karşılaştırma, iflas ya da başarısızlık tahmini gibi alanlarda veri madenciliği tekniği ile birlikte istatistiksel uygulama tekniklerine sıklıkla rastlanmaktadır.

İnsanoğlu doğası gereği karar alırken bulanık ve tam anlamı ile kesin olmayan belirsiz dilsel ifadelerden yola çıkarak karar verme işlemini gerçekleştirir. Bulanık mantık ise insanoğlunun kesin olmayan dilsel ifadelerini matematiksel modele ihtiyaç duymadan modellemeyi hedeflemektedir. Oluşturulacak olan modelin sahip olduğu bilgiler ışığında karar verebileceği güvenilir bir yapıyı oluşturmayı hedeflemektedir.

Literatürde bankacılık ve finans alanında veri madenciliği, istatistik teknikleri, bulanık mantık ve yapay sinir ağları kullanılarak yapılmış birçok çalışma bulunmaktadır. Ege Bayrakdaroğlu “İMKB Şirketlerinin Hisse Senedi Getiri Başarılarının Lojistik Regresyon Tekniği ile Analizi” adlı çalışmasında İMKB 30 hisse senetlerinin getiri performansını lojistik regresyon tekniği ile analizini yaparak İMKB de faaliyet gösteren şirketlerin başarı durumlarını belirtmiştir [1]. Girginer ve Cankuş “Tramvay Yolcu Memnuniyetinin Lojistik Regresyon Analiziyle Ölçülmesi: Estram Örneği” adlı çalışmalarında yolcu memnuniyetini Binominal Lojistik Regresyon Analizi ile incelemiştirler [2]. Şerbetçi ve Özçomak, üniversite öğrencilerinin istatistik ve ekonometri derslerine ilişkin başarılarını etkileyen faktörlerin belirlenmesi ve elde edilen sonuca bağlı olarak bu faktörler ile ilgili çalışmalar yaparak öğrencinin akademik eğitim kalitesini iyileştirmeyi hedefleyen bir çalışma yapmışlardır [3]. Özkan ve Doğan’ın çalışmalarında ilköğretim sekizinci sınıf öğrencilerinin cinsiyetleri, evlerinde sahip oldukları kitap sayısı, derslerde okumaya ayırdıkları zaman, zevk için okumaya ayırdıkları zaman, okul dışı (ders amaçlı) okumaya ayırdıkları zaman değişkenlerinin okuma becerisindeki başarılarını ne düzeyde etkilediğinin belirlenmesi amaçlanmıştır [4]. Tatlıdil ve Özel, kredi taleplerinin değerlendirilmesinde uzman sistem tabanlı karar destek sistem tasarımı ve uygulaması üzerine çalışmışlardır [5]. Gümüş, Çelik ve Üçer, bir bankaya başvurarak kredi talep eden bireysel müşterilerin kısa vadeli kredi taleplerinin değerlendirilmesinde, kredi talebinin kabul edilmesi ya da reddedilmesi kararının verilmesine yönelik Analitik Hiyerarşi Prosesi (AHP) yöntemiyle ilişkilendirilen bir model önermektedir [6]. Çelik, Türkiye’de finansal sistemin en önemli unsuru olan bankaların finansal başarısızlıklarının öngörülmesine yönelik erken uyarı modellerinin oluşturulmasında kullanılan Diskriminant Analizi ve Yapay Sinir Ağları modellerinin başarımları karşılaştırılmıştır [7]. Yücel, yüksek lisans çalışmasında bankaların maruz kaldıkları çeşitli risk türlerinin ölçümü ve yönetiminde kullanılan teknikler ve yöntemler incelenmiş ve bir portföyün maruz kaldığı risk değeri Varyans – Kovaryans yöntemi kullanılarak ölçülmüştür [8]. Koyuncugil, borsa şirketlerine yönelik olarak objektif risk tanımı ortaya koyacak bir metodoloji, model ve veri madenciliğine dayalı bir risk bazlı gözetim sistemi sunulmaktadır [9]. Koyuncugil diğer bir çalışmasında borsa şirketlerinin sektörel ayrımında tüm alt gruplarda risk belirmesinin sağlayacak, risk tayininde fikir verebilecek öncü göstergelerin tanımlanmasını

sağlayacak, erken uyarı özellikli, veri madenciliğine dayalı bir risk modeli tanımlamaktadır [10].

Vassiliou, “Fuzzy semi-Markov migration process in credit risk” adlı çalışmasında Bulanık küme teorisinin kullanımını haklı çıkaran sorunlara odaklanarak kredi kuruluşları tarafından kullanılan derecelendirme sistemini keşfetmektedir. Ödenebilir teminatın kredi gelişimini homojen olmayan semi-Markov sürecini bulanık mantık ile modellemişlerdir. Aynı zamanda gerçek olasılık ölçüsünün ön olasılık ölçüsüne dönüştürülmesinin etkilerini incelemişlerdir [11]. Saha, Bose, Mahanti, “A knowledge based scheme for risk assesment in loan processing by banks” adlı çalışmalarında bankaların kredilendirme sürecinde kullanılmak üzere bilgi odaklı bir otomatik uygunluk denetim şeması önermektedir [12]. Malhotra, K. Malhotra, “Evaluating consumer loans using neural networks”, başlıklı çalışmalarında verilecek olan potansiyel kredinin belirlenmesinde çoklu diskriminant analizi ile sinir ağları analizinin performanslarını karşılaştırmaktadır [13]. Zhang, Gao, Shi, “Credit risk evaluation using multi-criteria optimization classifier with kernel, fuzzification and penalty factors” başlıklı bu çalışmalarında çekirdek, bulanıklaştırma ve ceza faktörleri temelli yeni çok ölçütlü optimizasyon sınıflandırıcı yapısı önermektedirler [14]. Wu, Hsu “Credit risk assesment and decision making by a fusion approach”, adlı çalışmalarında sınıflandırıcıların performansını değerlendirebilmek için çok sayıda kredinin atamasını yapmak ve bu atamalara uygun uyarı mekanizmalarını belirleyebilmek için çok kriterli bir karar verme yöntemi sunmaktadırlar [15].

Kredi sınıflandırmasında doğruluk, karmaşıklık ve yorumlanabilirlik çok önemlidir. Ancak çoğu yaklaşımda bu üç bakış açısı başarı ile uygulanamaz. Zhu, Li, Wu, Wang, Liang, “Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-Topsis classification approach” adlı çalışmalarında bu üç bakış açısını dengeleyebilen C-Topsis adlı bir sınıflandırma yaklaşımını ortaya koymuşlardır [16]. Li, Shiue, Huang “The evaluation of consumer loans using support vector machines” adlı çalışmalarında tüketici kredilerinde potansiyel başvuru sahiplerini belirlemek için SVM tekniğini kullanan bir kredi değerlendirme modeli geliştirmişlerdir [17]. Tsai, Lin, Cheng, Lin “The consumer loan default predicting model – An application of DEA-DA and neural network” adlı çalışmalarında Tayvan’daki belirli bir finansal kuruluşun

teminatsız tüketici kredisi alan müşteriler için deneysel analiz yaparak tüketici kredisi için varsayılan tahmini modeli oluşturmuşlardır. Ayrıca borçlunun demografik değişkenlerini ve para tutumunu gerçek zamanlı ayırt edici bilgi olarak kabul etmişlerdir [18]. Danenas, Garsva “Selection of Support Vector Machines based classifiers for credit risk domain” başlıklı çalışmada doğrusal SVM sınıflandırıcısına ek olarak dış değerlendirme ve kayan pencere testleri ile daha büyük very kümeleri üzerinde kredi riski değerlendirmesi için bir yaklaşım açıklamışlardır [19]. Gorzalczany, Rudzinski “A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability” başlıklı çalışmalarında çok nesneli evrimsel optimizasyon algoritmaları kullanan finansal verilerden bulanık kural tabanlı sınıflandırıcıların otomatik olarak tasarlanması için bir yaklaşım önermektedirler [20]. Dahal, Hussain, Hossain “Loan Risk Analyzer based on Fuzzy Logic” çalışmalarında tüm girdilerin geleneksel bulanık mantıksallaştırılması ve aşamalı olarak girdi parametrelerinin önemine göre bulanık mantık yürütülmesi olmak üzere iki ayrı yapı kullanılarak bulanık mantığa dayalı bir kredi risk analiz sistemi geliştirmişlerdir. [21]. Ferreira, Santos, Dias “An AHP-based approach to credit risk evaluation of mortgage loans” adlı çalışmalarında değerlendirme kriterleri arasındaki dengeyi ayarlamak ve karar vericilere daha şeffaf bir ipotek risk değerlendirme sistemi sunmak için tasarlanmış bir metodolojik çerçeve önermektedir [22].

Shahbudin, Hussain, Hussain, A.Samad, Tahir “Analysis of PCA Based Feature Vectors for SVM Posture Classification” adlı çalışmalarında eğitim sürecinde, iki farklı tanımlayıcı kombinasyonuna dayanan SVM tekniği kullanılarak insan vücudu pozisyonunu analiz etmek ve sınıflandırmak için iki farklı çözücü hesabı kullanarak sınıflandırmışlardır [23]. Martens, Baesens, Gestel, Vanthienen, Leuven “Comprehensible Credit Scoring Models Using Rule Extraction From Support Vector Machines” başlıklı çalışmalarında son zamanlarda sıklıkla kullanılan SVM (destek vektör makineleri) analizleri için kural çıkarma tekniklerine genel bir bakış açısı sunmuşlardır [24]. Huang, Chen, Wang “Credit Scoring with a Data Mining Approach Based on Support Vector Machines” adlı çalışmalarında kredi başvurusunda başvuranın girdi özelliklerinden elde edilen kredi notunu değerlendirmek için hibrid SVM analizi tabanlı kredi puanlama modelleri oluşturmak için üç strateji kullanmışlardır [25]. Ha, Nguyen “Credit scoring with a feature selection approach based deep learning”

çalışmalarında kredi başvurusunda bulunan müşterilerin girdi özelliklerinden elde edilen kredi notunu değerlendirmek için derin öğrenme ve özellik seçimi üzerine kredi puanlama modeli oluşturmuşlardır [26]. Hongjiu, Yanrong, Wuchong “An Application of Support Vector Machine for Evaluating Credit Risk of Bank” adlı çalışmalarında bankalardaki ticari kredilerin kredi riskini değerlendirmek üzere SVM analizinin gelişmiş geribildirim ağı tekniğini kullanmışlardır [27]. Wu, Guo, Zhang, Xia “Study of Personal Credit Risk Assessment Based on Support Vector Machine Ensemble” başlıklı çalışmalarında kredi alacak kişileri iyi ve kötü olacak şekilde sınıflandırmak için bulanık integral üzerine kurulu SVM tabanlı bir yöntem üzerinde çalışmışlardır [28]. Min, Lee “Bankruptcy Prediction Using Support Vector Machine with Optimal Choice of Kernel Function Parameters” adlı çalışmalarında daha açıklayıcı yeni bir model önermek amacıyla destek vektör makinelerini (SVM) iflas öngörme problemine uygulamışlardır. Bu modeli geliştirmek için çalışmalarında SVM’ in çekirdek fonksiyonunun optimum parametre değerlerini bulmak için 5 kat çapraz doğrulama kullanan bir grid arama tekniği kullanmışlardır [29].

Bu çalışmanın birinci bölümünde istatistik ve veri madenciliği tekniklerinden biri olan karar ağaçlarıyla exchaid analizi (Exhaustive Chi-squared Automatic Interaction Detection), sınıflandırma metotlarından SVM (destek vektör makineleri) analizi, bulanık mantık ile genetik algoritma analizi ve son olarak yapay sinir ağları analizi açıklanacaktır. Çalışmanın uygulama bölümünde ise müşterilerin talep ettikleri kredi miktarının geri ödeme durumunu etkileyen faktörleri bulabilmeyi sağlayan en uygun analiz modeli için sırasıyla algoritmalar SPSS yazılımı kullanılarak çalıştırılacaklardır. Ardından Matlab programı kullanılarak bulanık mantık ile genetik algoritma analizi yapılacaktır. Son olarak makine öğrenmesi çalışmalarında sıklıkla kullanılan açık kaynak kodlu olan Weka yazılımı kullanılarak yapay sinir ağları analizi yapılacaktır.

2. YÖNTEM

Bu bölümde hedeflenen model oluşturulurken kullanılan yöntem ve metotlardan sırası ile bahsedilecektir. Bu kapsamda öncelikle sahip olunan ham veriden anlamlı örüntüler çıkarma süreci olan veri madenciliği tekniğinden bahsedilecektir. Ardından veri madenciliğinde kullanılan karar ağaçları, chaid algoritması, destek vektör makineleri ve temel bileşenler analizleri sırasıyla ayrı başlıklar altında ele alınacaktır. Ardından model oluşturulurken kullanılan bulanık mantık, genetik algoritma ve yapay sinir ağları yöntemlerinden ayrı başlıklar altında bahsedilecektir.

2.1 MAKİNE ÖĞRENMESİ

Günümüzde makine öğrenmesi belirli bir problemin çözümünde varolan büyük veri kümelerinden anlamlı örüntüler çıkarılarak tahminlerde bulunmak ya da insan deneyimlerinden yararlanılarak bir işi otomatize edilmesi makine öğrenme teknikleri kullanılarak bilgisayarlara aracılığı ile sıklıkla çözülmeye çalışılan yöntemler haline gelmiştir. Bu kapsamda makine öğrenmesi teknikleri hemen hemen tüm yaşam alanımızda kullanılmaya başlanmıştır. Örneğin internette arama yaparken, bir hastalığın teşhisinde ya da kredi kartlarında dolandırıcılık tespiti gibi bir çok alanda kullanılmaktadır. Makine öğrenmesi ve yapay zeka terimleri günümüzde sıklıkla kullanılmaktadır. Hatta kimi zaman bu iki terimin kavram karmaşıklığı nedeniyle birbirlerinin yerine kullanıldıkları görülmektedir. Ancak yapay zeka tanımlanmış görevleri akıllı olarak tabir edeceğimiz şekilde yapabilen, genel anlamda bir makinenin tanımıdır. Buna karşılık makine öğrenmesi ise yapay zeka olarak tasarlanan uygulama ve makinelerin öğrenmelerine olanak sağlayan yaklaşımı temel alan bir yapay zeka uygulamasıdır.

Makine öğrenimi istatistik teknikleri ve bilgisayar algoritmaları aracılığı ile mevcut verilerden çıkarımlar yaparak bilgisayarların nasıl öğreneceklerini araştıran bir alandır. Makine öğrenmesi alanında çalışma yapmak ve çıkarımda bulunmak için birçok yöntem ve algoritma mevcuttur. Makine öğrenmesi temelde üç ana başlık altında incelenir,

- Eğitilmiş öğrenme (Supervised Learning)

- Eđitmensiz öğrenme (Unsupervised Learning)
- Pekiřtirmeli öğrenme (Reinforcement Learning)

2.1.1 VERİ MADENCİLİĐİ

Veri madenciliđi en genel anlamda çok büyük miktarlardaki verilerin depolanması ve depo edilen bu veri yığını içerisindeki anlamlı örüntüleri alarak bunlardan bir sonuç çıkarma işlemidir. Genel olarak veri madenciliđi tahmin edici ve tanımlayıcı modeller çıkarmasında kullanılması yanı sıra yeni modeller üretme işleminde de kullanılmaktadırlar.

Veri madenciliđi büyük miktarda veri inceleme amacı üzerine kurulmuş olduđu için veritabanları ile yakından ilişkilidir. Gerekli veriye hızla ulařılabilecek biçimde saklanması ve gerektiğinde veriye ulařılabilmesi gerekir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları, günlük kullanılan veri tabanları ile birleřtirilmiş olup işlenmeye daha uygun birer özet veri tabanını saklamayı amaçlar. Günlük veritabanlarından istenen özet bilgi seçilir ve gerekli öniřlemeden sonra veri ambarında saklanır. Ardından istenilen amaç dođrultusunda veri ambarından alınarak veri madenciliđi çalışması için standart bir biçime çevrilir. Veri madenciliđinde amaç, kullanıcının bilgi çıkarma sürecine katkısının olabildiđince az tutulması, işin olabildiđince otomatik olarak yapılabilmesidir [30].

İstatistik literatüründe çokboyutlu analiz (*multivariate analysis*) başlığı altında verinin parametrik bir modelden (çokboyutlu bir Gauss dağılımından) geldiđi varsayılır. Bu varsayım altında sınıflandırma (*classification; discriminant analysis*), regresyon, öbekleme (*clustering*), boyut azaltma (*dimensionality reduction*), hipotez testi, varyans analizi, bağıntı (*association; dependency*) kurma için teknikler istatistikte uzun yıllardır kullanılmaktadır [31].

Veri madenciliđinde kullanılan modeller tahmin edici (*predictive*) ve tanımlayıcı (*descriptive*) olmak üzere iki ana başlık altında incelenmektedir. Tahmin edici modeller, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve geliştirilen bu

modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerinin tahmin edilmesi amaçlanmaktadır. Tahmin edici modeller sınıflama ve regresyon yöntemleridir. Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler aşağıda sıralanmıştır:

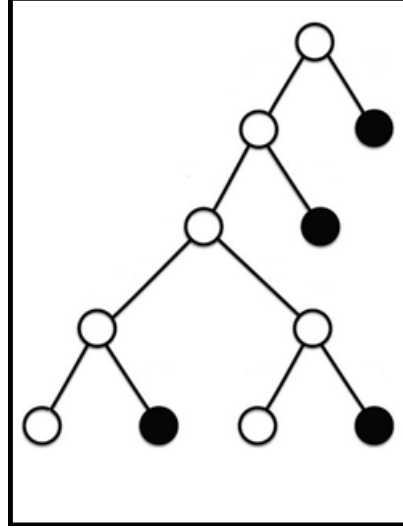
- Karar Ağaçları
- Yapay Sinir Ağları
- Naive Bayes
- Bulanık Mantık

Tanımlayıcı modellerde ise karar verme aşamasında kullanılacak örüntülerin tanımlanması amaçlanır. Tanımlayıcı modeller kümeleme ve sınıflandırma kurallarıdır.

2.1.1.1 KARAR AĞAÇLARI

Karar ağaçları veri madenciliğinde en sık kullanılan tahminleyici modellerin başında gelmektedir. Nedeni ise yorumlanmasının oldukça kolay olması ve veri tabanı sistemleri ile entegre edilebilmesidir. Karar ağaçları düğümler ve dallardan oluşur. Karar ağacında bulunan her bir dalın belirli bir olasılığı mevcuttur. Bu sayede son dallardan kök dala veya istediğimiz dala ulaşmaya kadar olan olasılıkların hesaplanması mümkündür.

İstatistikî yöntemlerde, bulanık mantık veya yapay sinir ağlarında veriden bir model öğrenildikten sonra bu fonksiyonun insanlar tarafından anlaşılabilir bir kural olarak yorumlanması zordur. Karar ağaçları ise ağaç oluşturulduktan sonra, kökten yaprağa doğru inilerek kurallar yazılabilir. Bu şekilde kural çıkarma veri madenciliği çalışmasının sonucunun doğrulanmasını sağlar. Bu kurallar, uygulama konusunda uzman bir karar vericiye gösterilerek sonucun anlamı olup olmadığı denetlenir ve kurallar konusunda karar vericiye bilgi verir. Şekil 2.1 de örnek olarak oluşturulmuş bir karar ağacı yapısı gösterilmektedir.



Şekil 2.1 Karar Ağacı örneği

Bir karar ağacı oluşturulurken, Öncelikle sorun analiz edilir, ardından ilk düğüm karar düğümü olmakla beraber soldan sağa doğru yatay doğrultuda ağaç oluşturulmaya başlanır. Ağacın en üstündeki ilk düğüm olan kök (*root*) düğümün özelliği bilgi kazancı (*information gain*) ile belirlenir. Karar ağaçları algoritmalarında en ayırt edici niteliği belirlemek için her nitelik için bilgi kazancı ölçülür. Bilgi kazancı ölçümünde Entropi ile rastgeleliğin, belirsizliğin ve beklenmeyen durumun ortaya çıkma olasılığı gösterilir. Böylece bilgi kazancı en yüksek olan özellik ağacın en üstünde konumlandırılır.

Entropi tek bileşenli ve iki bileşenli olarak hesaplanabilmektedir. Tek bileşenli entropi formülü ile hesaplama örneği denklem 2.1 de gösterilmiştir.

$$E(X) = - \sum_{i=1}^n P_i \log_2 P_i \quad (2.1)$$

Burada, X rasgelelik içeren bir sistemi temsil etmektedir. X' in gerçekleşebilecek her bir durumunun olasılığı $P_i = P_i(X = x_i)$ ile ifade edilir.

İki bileşenli Entropi formülü ile hesaplama örneği denklem 2.2 de gösterilmiştir.

$$E(X, Y) = - \sum_{i=1}^n \sum_{j=1}^m P_{ij} \log_2 P_{ij} \quad (2.2)$$

Burada, X ve Y rasgelelik içeren iki bileşenli bir sistemi temsil etmektedirler. X ve Y' nin gerçekleşebilecek durumlarının olasılıkları $P(X = x_i, Y = y_j)$ ifade edilmektedir.

Gini katsayısı ise herhangi bir dağılımdaki eşitsizlik miktarını ölçen bir yöntemdir. Genellikle ekonomide ülkelerin gelir dağılımlarının eşitsizliklerini ölçme amacıyla kullanılmaktadır. Denklem 2.3 de tanımlanan hesaplama formülü ile karar ağacı ile oluşturulan her bir düğüm için hesaplanarak en düşük gini değerini veren ayrıma sahip değişken seçilir ve bölünmeye bu özellik üzerinden devam edilir. Karar düğümünün değişkenlerinin alacağı tüm olası değerler düğümden dallara doğru ağaç oluşumuna eklenir. Kök düğümden sonra kalan durumlardaki olayları belirten bir karar düğümü daha eklenir. Bu karar düğümünün altında yer alan diğer karar dallarına kendilerine ait olasılıklarla diğer karar dalları eklenir ve ağacın çizimi bu şekilde dallardan yeni yapraklara doğru, sonuç için kök düğüme ulaşılan kadar devam edilir.

Gini katsayısı hesaplama formülü ise aşağıda denklem 2.3 de gösterilmektedir.

$$GINI(t) = 1 - \sum_i [p(i/t)]^2 \quad (2.3)$$

Burada, p olasılığı ifade etmekle birlikte gini katsayısı hesaplama formülü ile karar ağacı oluşturulurken ağacın bölümlenmesinden sonraki uygun olmayan bölümlenmeleri bulmak için kullanılır.

Entropi hesaplamalarından sonra bilgi kazancı hesaplaması yapılır. Bilgi kazancı genel anlamda bir olayın meydana gelme olasılığının entropi ile hesaplanması sonucunda elde edilen değerdir. Karar ağaçlarında ise bilgi kazancı (information gain) bir veri kümesinin bir özellik üzerinde en iyi parçalanmayı sağlayan değerinin bulunmasıdır. Karar ağacında her düğüm oluşturulurken en fazla bilgi veren özellik kullanılır. Bilgi kazanımı yöntemi ile en fazla bilgi veren özelliğin seçilmesi sağlanır.

Bilgi kazancı hesaplaması ise denklem 2.4 de gösterilmektedir.

$$Gain(X, Y) = Entropy(X) - Entropy(X, Y) \quad (2.4)$$

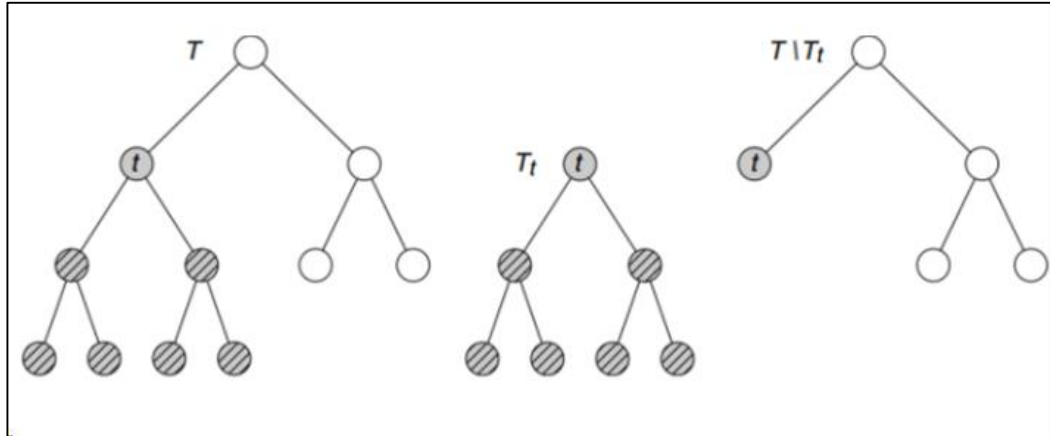
Burada bilgi kazancı hesaplamasının karar ağacının bölümlenmesinden sonra hesaplanan tek bileşenli ve çok bileşenli entropi değerlerinin azaldığı gösterilmektedir.

Karar ağacı algoritmalarında iki temel işlem gerçekleştirilmektedir. Bunlardan ilki bölme (splitting) işlemidir. Bölme işlemi verilerin daha küçük alt kümelere ayrılmasını içeren yinemeli bir süreçtir. Her bölme işleminde değişkenler analiz edilir ve en iyi

bölme seçilir. Diğer işlem ise budamadır (pruning). Bir ağaç oluşturulduktan sonra oluşturulan ağaç içerisinde istenmeyen alt ağaçlar veya düğümler bulunabilir. Budama işlemi ile ağacın derinliği azaltılır. Bu işlem ile daha genel bir model oluşturulması sağlanır. Şekil 2.2 de görüldüğü üzere budama işlemi ile istenmeyen alt ağaçlar ve düğümler ağaç yapısından ayıklanmıştır. Bu ayıklama işleminden sonra elde edilen ağaç ile daha yalın ve okunabilir bir yapı elde edilmiştir. Budama ile ayıklama iki şekilde yapılır:

- Pre-pruning (erken budama), ağaç en büyük şekline ulaşmadan öğrenmenin erken bitirilmesidir.
- Post-pruning (geç budama), ağacın tam büyüklüğe ulaştıktan sonra budanmasıdır.

Erken budama işlemi doğrudan bir yaklaşım gibi görünürken, geç budama işlemi pratikte daha kabul edilir sayılmaktadır [32].



Şekil 2.2 Budanmamış karar ağacı ve budanmış karar ağacı örneği.

2.1.1.2 CHAID ALGORİTMASI

Chaid analizi çok sayıda kategorik değişken içeren veri kümelerinde örüntü aramada kullanılan yararlı yöntemlerden birisidir. Veri kümesi içerisindeki ilişkileri kolayca görselleştirerek verileri özetlemenin kolay yollarından bir tanesidir. Chaid algoritması, kategorik değişkenlere ilişkin veri kümesini ve bağımlı değişkeni en iyi şekilde

açıklayabilecek biçimde ayrıntılı homojen alt kümelere böler. Bu alt kümeler, küçük tahmin edici gruplardan oluşur. En iyi tahmin sonucunu elde etmek için başlangıç değişkenleri bağımsız olarak yeniden kategorileştirilir. Bunun için Ki kare analizi kullanılır. Adımsal olarak uygulanan benzer kategorileri birleştirme işlemi, değişkenler arasında daha fazla birleştirme sağlanamayacağına istatistiksel olarak karar verilmeye kadar devam eder. Değişkenlerin bölünmeye uygun olup olmadığına Benferroni düzeltilmiş “p” değeri kullanılarak karar verilir.

Genel olarak CHAID algoritması, seçilen bağımlı değişkeni en iyi şekilde açıklamak için kullanılır. Açıklayıcı değişkenler sınıflayıcı, sıralayıcı ve aralıklı ölçek ile ölçülmüş olabilir. Analizi yapılacak veri kümesi içerisindeki kayıp verilere yeni bir kategori gibi davranır ve bu kategoriyi “p” değeri hesaplamalarına dahil eder. Hesaplamalara dahil edilen bu “p” değeri Bonferroni düzeltilmiş “p” değeridir ve bu değere bakılarak değişkenlerin bölünmeye uygun olup olmadıklarına karar verilir. Bu değer ile birlikte kategorileri sıralanabilen ya da sıralanamayan, açıklayıcı değişkenlerin yer aldığı veri kümesini, bağımlı değişkene göre detaylı alt kümelere böler ve bu bölünme işlemini gerçekleştirirken açıklayıcı değişkenlere ait kategorileri bağımsız olarak yeniden kategorileştirir. Daha sonraki her bölünmeyi yeniden bağımsız olarak gerçekleştirir.

Bonferroni yaklaşımı ise her grubun ortalama vektörlerinin genel ortalama vektöründen farklarının bulunmasından sonra bulunan farkların sıfır olup olmadığının araştırılmasına dayanmaktadır [33].

$$\vec{x} = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_p \end{bmatrix} \rightarrow_{x_1} \begin{bmatrix} \vec{x}_{11} \\ \vec{x}_{21} \\ \vdots \\ \vec{x}_{p1} \end{bmatrix} \rightarrow_{x_2} \begin{bmatrix} \vec{x}_{12} \\ \vec{x}_{22} \\ \vdots \\ \vec{x}_{p2} \end{bmatrix} \dots \rightarrow_{x_g} \begin{bmatrix} \vec{x}_{1g} \\ \vec{x}_{2g} \\ \vdots \\ \vec{x}_{pg} \end{bmatrix} \quad (2.5)$$

Burada ortalama vektör \vec{x} ve her grubun i. değişkenine göre ortalama vektör ise \vec{x}_g ile gösterilmektedir.

Her matris grubunun ortalama vektörünün, genel ortalama vektöründen farkları değişkenlere göre aşağıdaki gibi hesaplanır [33]:

$$d_1 = \begin{matrix} \rightarrow \\ x_1 \end{matrix} - \begin{matrix} \rightarrow \\ X \end{matrix} = \begin{bmatrix} \rightarrow \\ x_{11} \\ \rightarrow \\ x_{21} \\ \vdots \\ \rightarrow \\ x_{p1} \end{bmatrix} - \begin{bmatrix} \rightarrow \\ x_1 \\ \rightarrow \\ x_2 \\ \vdots \\ \rightarrow \\ x_p \end{bmatrix} \quad d_g = \begin{matrix} \rightarrow \\ x_g \end{matrix} - \begin{matrix} \rightarrow \\ X \end{matrix} = \begin{bmatrix} \rightarrow \\ x_{1g} \\ \rightarrow \\ x_{2g} \\ \vdots \\ \rightarrow \\ x_{pg} \end{bmatrix} - \begin{bmatrix} \rightarrow \\ x_1 \\ \rightarrow \\ x_2 \\ \vdots \\ \rightarrow \\ x_p \end{bmatrix} \quad (2.6)$$

Burada k. grup, l. grup ve i. deęişkenin her birinin kendi ortalamaları ile birbirleri arasındaki ortalamaların farkları arasında $1 - \alpha$ güven aralığı Denklem 2.7 de hesaplanmaktadır [33].

$$(d_{ki} - d_{ii}) = \left(\begin{matrix} \rightarrow \\ x_{ki} \end{matrix} - \begin{matrix} \rightarrow \\ x_{ii} \end{matrix} \right) \mu t \left(\frac{\alpha}{pg(g-1)}, (N - g) \right) \sqrt{\left(\frac{1}{n_k} + \frac{1}{n_i} \right) \frac{w_{ii}}{N-g}} \quad (2.7)$$

Burada, $N = n_1 + n_2 + \dots + n_g$, p deęişken sayısı, g grup sayısı ve w_{ii} de W matrisinin köşegen elemanıdır. W matrisi gruplar içi deęişimi gösterir ve Denklem 2.8 deki gibi hesaplanmaktadır [33].

$$W = \sum_{i=1}^g \sum_{j=1}^{n_i} \left(x_{ij} - \begin{matrix} \rightarrow \\ x_i \end{matrix} \right) \left(x_{ij} - \begin{matrix} \rightarrow \\ x_i \end{matrix} \right)' \quad (2.8)$$

Burada grup sayısı g ile gösterilmektedir. n_1 ise i. gruptaki birim sayısını ifade etmektedir. Her bir deęişken için gruplar ikişerli olarak dikkate alınır ve (2.5) numaralı denklem kullanılarak i. deęişken için elde edilen aralığın sıfır deęerini içerip içermedięi kontrol edilir. Eęer sıfır deęeri kontrol edilen denklem ile belirlenen aralıkta yer alıyorsa, ilgili gruplar arasında anlamlı bir farklılık olmadığı söylenir, ancak sıfır deęeri kontrol edilen denklem ile belirlenen aralıkta bulunuyorsa grupların farklı olduğu söylenir [33].

Exhaustive Chaid algoritması ise Chaid algoritmasında olduğu gibi bir düęümün birden fazla bölünmesine izin verirken her iki algoritma da birleřtirme, bölme ve durdurma olmak üzere üç aşamalı bir yapıdan oluşur. Bir ağaç, kök düęümden başlayarak her düęümde bu üç adımı tekrar tekrar kullanarak büyür. Exhaustive Chaid algoritmasında bölme ve durdurma adımları Chaid algoritmasındaki adımlarla aynıdır. Birleřtirme adımı ise Chaid algoritmasından farklı olarak benzer çiftleri bir çift kalana kadar birleřtirmek için ayrıntılı bir arama yordamı kullanır. Ayrıca Chaid algoritmasındaki

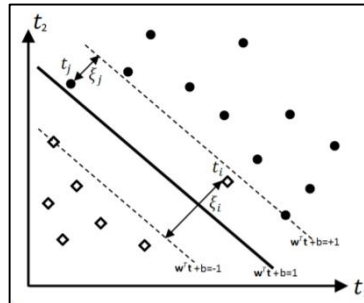
gibi sadece nominal veya sıralı kategorik tahmin edicilere izin verilir, sürekli tahmin ediciler ExChaid algoritmasını kullanmadan önce sıralı kategorik tahmin ediciler haline dönüştürülür.

2.1.1.3 DESTEK VEKTÖR MAKİNELERİ

Sınıflandırmanın temel amacı verileri basitleştirmek ve kullanıcılara daha anlaşılır bilgi sunmaktır. Destek vektör makineleri, verilerin sınıflandırılmasında kullanılan etkin ve basit sınıflandırma yöntemlerinden biridir. Destek vektör makineleri (SVM) 1992 yılında Boser, Guyon ve Vapnik tarafından önerilmiştir. Bir destek vektör makinesinin amacı, eğitim verisinin marjını en yükseğe çıkaran en iyi gruplayan çizgiyi (hiperplane) bulmaktır. Destek vektör makineleri (SVM) algoritmasının eğitim verilerine ihtiyaç duyması durumu destek vektör makinelerini (SVM) hem eğitilmiş öğrenme algoritması hem de sınıflandırma algoritması yapmaktadır. Destek vektör makineleri yapıları gereği hem doğrusal olarak ayırt edilebilir veri kümelerini sınıflandırabilmekte hem de lineer olarak ayırt edilemez veri kümelerini sınıflandırabilmektedir.

Şekil 2.3 de gösterildiği gibi doğru bir dönüşüm ile veriler her zaman destek vektör makineleri (SVM) tarafından çizilen bir çizgi (hiperlane) ile iki sınıfa ayrılabilir. Çizgiye (hiperplane) en yakın öğrenme verilerine destek vektörleri adı verilmektedir.

İki sınıftan oluşan bir sınıflandırma probleminin, esnek kenar boşluklarına sahip destek vektör makineleri için ilk model aşağıdaki denklemler yardımıyla ifade edilmektedir [34]:



Şekil 2.3 SVM gösterimi

$$\min_{w,b} \frac{1}{2} w^t w + C \sum_{i=1}^N \zeta_i \quad (2.9)$$

$$y_i(w^t t_i + b) \geq 1 - \zeta_i, i = 1, \dots, N \quad (2.10)$$

$$\zeta_i \geq 0, i = 1, \dots, N \quad (2.11)$$

Burada, w ve b düzlemdeki noktaları göstermektedir ve bu noktaların hiper düzleme uzaklıkları hesaplanmaktadır. C değişkeni ise Lagrange değişkenin alabileceği en üst sınır değerini ifade eden ceza değişkenidir. Destek vektör makinelerinin iki sınıftan oluşan sınıflandırma probleminin çözümü için sunulan ikili modeli ise aşağıdaki denklemler yardımı ile elde edilmektedir [34]:

$$\min_{\alpha} \mathcal{L}(\alpha) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j t_i^T t_j - \sum_{i=1}^N \alpha_i \quad (2.12)$$

$$\sum_{i=1}^N \alpha_i y_i, i = 1, \dots, N \quad (2.13)$$

$$0 \leq \alpha_i \leq C, i = 1, \dots, N \quad (2.14)$$

Burada, t_i değişkenleri giriş y_i değişkenleri ise çıktı değişkenleridir. α Lagrange değişkenini temsil etmektedir. C değişkeni ise Lagrange değişkenin alabileceği en üst sınır değerini ifade eden ceza değişkenidir. Bunlarla birlikte, destek vektör makineleri yaklaşımını daha etkin olmasını sağlayan, giriş alanını nitelik alanıyla eşleştiren çekirdek işlevlerdir. Destek vektör makinelerinde en sık kullanılan fonksiyonlar ise Gauss, Polynomial, Sigmoid, Linear and Radial Base Core Function (RBF) dir [34].

Destek vektör makineleri yaklaşımının daha etkin olmasını sağlayan çekirdek matrisinin ikili modeli ise aşağıdaki denklemlerle ifade edilmektedir [34]:

$$\min_{\alpha} \mathcal{L}(\alpha) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(t_i, t_j) - \sum_{i=1}^N \alpha_i \quad (2.15)$$

$$\sum_{i=1}^N \alpha_i y_i, i = 1, \dots, N \quad (2.16)$$

$$0 \leq \alpha_i \leq C, i = 1, \dots, N \quad (2.17)$$

Burada, t_i değişkenleri giriş y_i değişkenleri ise çıktı değişkenleridir. α Lagrange değişkenini temsil etmektedir. K ise çekirdek matrisini ifade etmektedir.

İkinci dereceden programlama probleminin çözümlenmesinin bir sonucu olarak ikili uzayda elde edilen sınıflayıcı modeli Denklem 2.18 ile hesaplanmaktadır [34].

$$\hat{y} = \sum_{i \in S}^{\#SV} \alpha_i y_i K(t, t_i), i = 1, \dots, \#SV \quad (2.18)$$

Burada, K_{ij} çekirdek matrisini, S destek vektör makinelerinin kümesini belirtmektedir. $\#SV$ destek vektör makinalarının sayısını belirtmektedir. \hat{y} ise sınıflandırıcının tahminini belirtir. Elde edilen model ile sınıflayıcı modeli ilkel modelden tamamen bağımsız olarak kullanılabilir [34].

2.1.1.4 TEMEL BİLEŞENLER ANALİZİ

Temel bileşenler analizi ile veri kümesinde sınıflandırma yapılırken sınıflandırmanın sonucunu etkileyen özelliklerin etki sıralaması yapılır. Böylece sınıflandırmaya en az etki eden özelliklerin bulunarak veri kümesi içerisinde çıkarılması sağlanır. Temel bileşenler analizi ile hem modelin başarımı hem de modelin çalışma performansı artırılır. Genel olarak temel bileşenler analizi aralarında korelasyona sahip değişken sayıları ile aralarında hiçbir korelasyona sahip olmayan ve orjinal değişken sayısından küçük olan ($k < p$) lineer bileşenlerin k değişkeni ile ifade edilmesi yöntemi olarak açıklanmaktadır. Kovaryans matrisinin ya da korelasyon matrisinin özdeğerleri ve öz vektörleri, veri matrisindeki p değişkeninin doğrusal bileşenlerini bulur. Temel bileşenler analizinin üç ana hedefi vardır [35, 36]:

- Veri miktarını sadeleştirmek,
- Tahmin yapmak,
- Bazı analizler için veri kümesini görüntülemek.

Temel bileşenler analizi sonucunda, p boyutlu alanın gerçek boyutu belirlenir. Elde edilen bu gerçek boyuta temel bileşen analizi denir. Temel bileşenler üç özelliğe sahiptir:

- Temel bileşenler korelasyonsuzdur.
- Birinci temel bileşen toplam değişkenliği en çok açıklayan değişkendir.

- Bir sonraki temel bileşen ise kalan değişkenliği en çok açıklayan değişkendir.

İlk temel bileşen 2.19 numaralı denklem ile ifade edilir,

$$y_1 = a_1^t(-\mu) \quad (2.19)$$

Burada, y_1 birinci temel bileşeni, a_1 ise hesaplanan en büyük özdeğere karşılık gelen birinci en büyük özvektörü temsil etmektedir.

İkinci temel bileşen 2.20 numaralı denklem ile ifade edilir,

$$y_2 = a_2^t(x - \mu) \quad (2.20)$$

Burada, y_2 ikinci temel bileşeni, a_1 ise birinci temel bileşen ile x vektörlerinin korelasyonsuz lineer kombinasyonlarının varyansının maksimumuna karşılık gelen ikinci en büyük özvektörü temsil etmektedir.

Temel bileşen skoru ise 2.21 numaralı denklem ile ifade edilir,

$$y r_j = a_j^t(x r - \mu) \quad (2.21)$$

Burada, temel bileşen skorunu hesaplamak için r . hesaplama durumuna karşılık gelen j . temel bileşenin skorunun hesaplanmasını göstermektedir.

Bileşenin yük vektörleri ise 2.24 numaralı denklem ile ifade edilir,

$$c_j = kök(\lambda_j \cdot a_j) \quad (2.24)$$

Burada, c_j i . eleman ile j . temel bileşen arasındaki kovaryansı göstermektedir.

2.2 BULANIK MANTIK

Günümüzde bulanık mantık verilerin analizi ve sınıflandırılmasında, uzman sistemlerde, otomasyon sistemlerinde makine öğrenmesi içeren tüm çalışma alanlarında sıklıkla kullanılmaktadır. Klasik mantıkta bulunan 1 ve 0 mantıksal durumu insanoğlunun karar verme süreci içerisinde önemini yitirmektedir. Çünkü gerçek hayatta evet ve hayır' a

karşılık dilsel ifade içeren birden fazla durum ve kesinlik içermeyen belirsiz ifadeler söz konusudur. Bulanık mantık vasıtası ile bu belirsiz dilsel ifadeleri matematiksel olarak yorumlayıp işlenebilmektedir. Bulanıklığın matematiksel olarak karşılığı ise bir durumun birden fazla değere sahip olması ile eşdeğerdir. 1965 yılında Lotfi Zadeh Çok Değerli Küme teorisini geliştirerek bulanık mantığın isim babası olmuştur.

Bulanık mantık yaklaşımı ile problem çözülürken, problemi çözülecek işte uzman olan kişinin bilgi ve tecrübeleri alınarak bir kural listesi oluşturulur. Bu kural listesindeki her bir kural matematiksel bir fonksiyona karşılık gelir ve bu kuralların matematiksel olarak işlenmesi ile ideal sonuca varılır.

Zadeh' e göre bulanık mantığın genel özellikleri:

- Kesin değerlere dayanan düşünme yerine, yaklaşık düşünüş kullanır.
- Her şey $[0-1]$ aralığında belirli derece ile gösterilir.
- Bilgi çok, az, biraz, normal gibi dilsel ifadeler şeklinde işlenir.
- Sonuç dilsel ifadeler arasında tanımlanan kurallar ile belirlenir.
- Her mantıksal sistem bulanık olarak ifade edilebilir.
- Bulanık mantık matematiksel modeli çok zor elde edilebilen sistemler için çok uygundur.
- Bulanık mantık olasılık teorisinden farklıdır. Olasılık teorisinde problemin kendisi tanımlıdır.

Klasik mantık ile bulanık mantık arasındaki temel farklar Tablo 2.1' de karşılaştırmalı olarak gösterilmiştir.

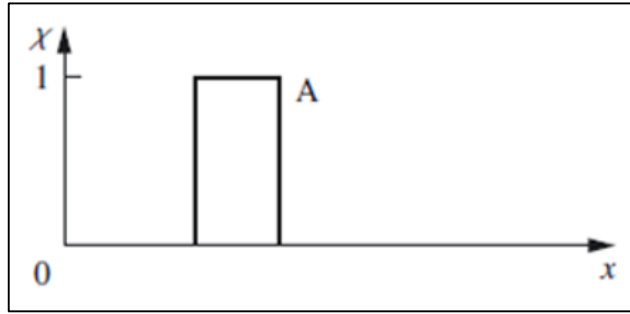
Tablo 2.1 Bulanık Mantık ve Klasik Mantık karşılaştırması

Klasik Mantık	Bulanık Mantık
A veya A Değil	A ve A Değil
Kesin	Kısmi
Hepsi veya Hiçbiri	Belirli Derecelerde
0 veya 1	0 ve 1 Arasında Süreklilik
İkili Birimler	Bulanık Birimler

Genel olarak bulanık mantık geleneksel mantığın ikili kavramları yerine ara kavramlar tanımlar. Tanımlanmış olduğu her bir ara kavrama genel küme içerisinde bir üyelik derecesi vererek gerçek dünya ile daha uyumlu bir mantık geliştirir. Bulanık mantık

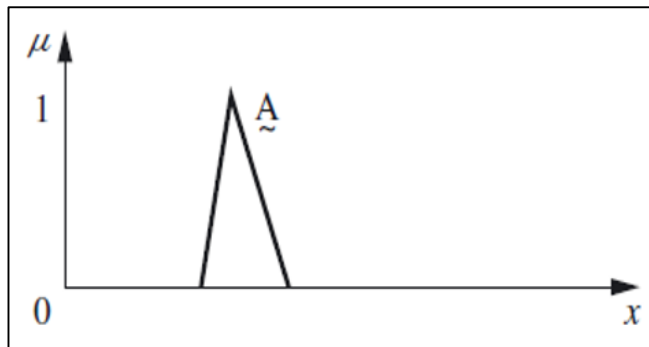
içerisindeki önermeler genellikle dilsel ifadeler şeklinde olduğundan belirsizlik taşırlar, işte bu noktada insanoğlu dilsel ifadelerle karar verirken kesin çıkarım yapmak yerine yaklaşım yaparak karar verir.

Klasik küme teorisine göre bir u elemanı, $\mu_A(u) = 1$, ise u A kümesinin elemanıdır ve $\mu_A(u) = 0$, ise A kümesinin elemanı değildir. Yani bir nesne ya bir grubun üyesidir ya da hiçbir grubun üyesi değildir. Şekil 2.4’ de klasik küme üyelik fonksiyonunun gösterimi bulunmaktadır.



Şekil 2.4 Klasik küme üyelik fonksiyonu.

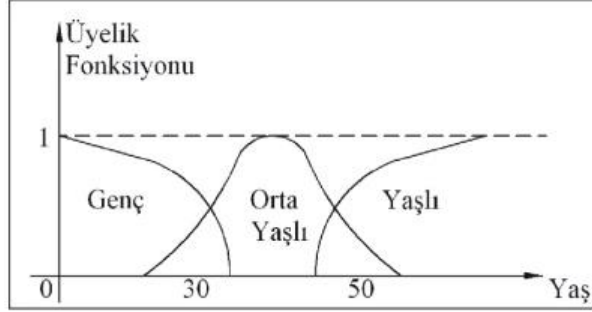
Bulanık küme teorisinde ise bir u elemanının kümeye aitlik derecesi 0 ile 1 arasındadır. $\mu_{A\sim}(x) = 1$, değeri x ’ in $A\sim$ bulanık kümesinin kesin bir elemanı olduğunu, $\mu_{A\sim}(x) = 0$ ise, x ’ in $A\sim$ bulanık kümesinin elemanı olmadığını gösterir ve bu 0 ile 1 arasındaki değerler $\mu_{A\sim}(x) \in [0,1]$ x ’ in bulanık kümesinin belirsiz değerleridir. Şekil 2.5’ te bulanık küme üyelik fonksiyonunun gösterimi bulunmaktadır.



Şekil 2.5 Bulanık küme üyelik fonksiyonu.

Örneğin Şekil 2.6’ da görüldüğü gibi insanoğlu yaşını bilmediği bir kişinin yaşını değerlendirirken genç, ort yaşlı ve yaşlı olarak ayırım yaparken dış görünüşe göre yapmış olduğu değerlendirme ile yaklaşımsal karar vermeye dayalı olarak aynı yaşlarda

bulunan iki kişiden birini belli oranda yaşlı diğerini ise belli oranda orta yaşlı sınıfına koyabilmektedir.

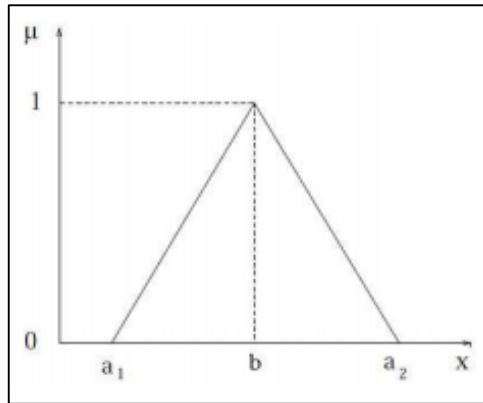


Şekil 2.6 Bulanık mantık ile kümeleme örneği

Üyelik fonksiyonları genel olarak üçgensel üyelik fonksiyonları ve yamuk üyelik fonksiyonları olmak üzere iki başlık altında incelenmekle birlikte Gaussian üyelik fonksiyonu ve çan üyelik fonksiyonları da kullanılmaktadırlar.

Üçgensel üyelik fonksiyonu 2.25 numaralı denklemde paylaşılan formül ile tanımlanmakla birlikte Şekil 2.7' de gösterilmektedir.

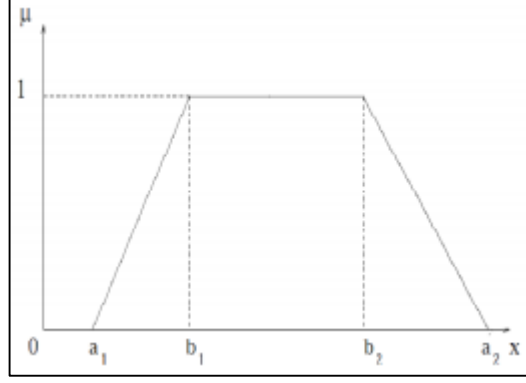
$$\mu(x; a_1, b, a_2) = \begin{cases} a_1 \leq x \leq b & \text{ise } (x - a_1)/(b - a_1) \\ b \leq x \leq a_2 & \text{ise } (a_2 - x)/(a_2 - b) \\ x > a_2 \text{ veya } x < a_1 & \text{ise } 0 \end{cases} \quad (2.25)$$



Şekil 2.7 Üçgen üyelik fonksiyonu.

Yamuk üyelik fonksiyonu 2.26 numaralı denklemde paylaşılan formül ile tanımlanmakla birlikte Şekil 2.8' de gösterilmektedir.

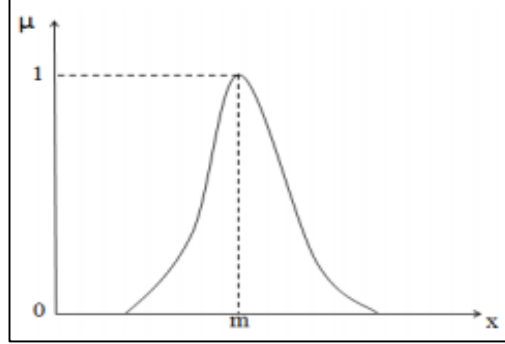
$$\mu(x; a_1, b_1, b_2, a_2) = \begin{cases} a_1 \leq x \leq b_1 & \text{ise } (x - a_1)/(b_1 - a_1) \\ b_1 \leq x \leq b_2 & \text{ise } 1 \\ b_2 \leq x \leq a_2 & \text{ise } (x - a_2)/(b_2 - a_2) \\ x > a_2 \text{ veya } x < a_1 & \text{ise } 0 \end{cases} \quad (2.26)$$



Şekil 2.8 Yamuk üyelik fonksiyonu.

Gaussian üyelik fonksiyonu 2.27 numaralı denklemde paylaşılan formül ile tanımlanmakla birlikte Şekil 2.9' te gösterilmektedir.

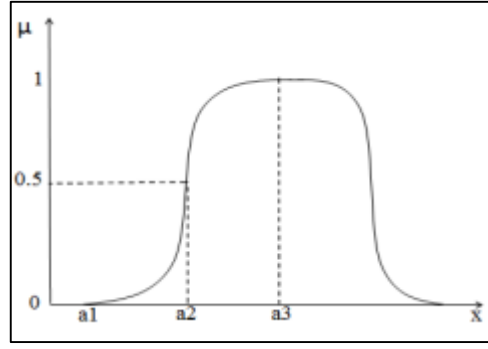
$$\mu(x; m, s) = \exp\left\{\frac{-(-x-m)^2}{2s^2}\right\} \quad (2.27)$$



Şekil 2.9 Gaussian üyelik fonksiyonu.

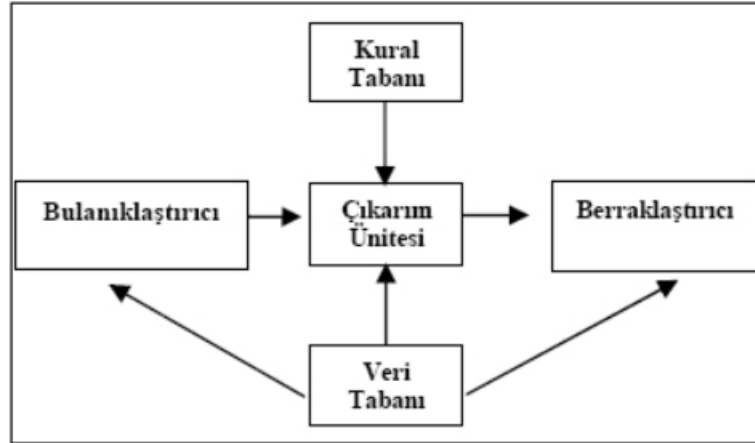
Çan üyelik fonksiyonu 2.28 numaralı denklemde paylaşılan formül ile tanımlanmakla birlikte Şekil 2.10' te gösterilmektedir.

$$\mu(x; a_1, a_2, a_3) = \left\{ \frac{1}{1 + \left| \frac{x-a_3}{a_1} \right|^{a_2}} \right\} \quad (2.28)$$



Şekil 2.10 Çan üyelik fonksiyonu.

Bir bulanık mantık sistemi genel olarak bulanıklaştırıcı (Fuzzifier), kural tabanı (Rule-Base), çıkarım mekanizması (Inference Mechanism) ve berraklaştırıcı (Defuzzifier) olmak üzere dört bölümden oluşur. Şekil 2.11’ de bulanık mantık sisteminin genel yapısı gösterilmektedir.



Şekil 2.11 Bulanık mantık sistemi genel yapısı.

2.3 GENETİK ALGORİTMA

Genetik algoritma kavramı Charles Darwin’ in doğal seleksiyon kuramından esinlenilerek oluşturulmuş, optimizasyon problemlerini çözmeye dayalı sezgisel aramaya dayalı stokastik optimizasyon algoritmasıdır. İlk kez 1975 yılında John Holland tarafından kullanılmıştır. Günümüzde genel olarak genetik algoritmalar büyük ve karmaşık veri kümelerindeki arama problemlerinin çözümünde sıklıkla kullanılmaktadırlar. Genetik algoritmanın işleyiş süreci evrim sürecini temel olarak

oluşturulduğundan genetik algoritma ile bir problemi çözebilmek için doğal seçim, mutasyon ve rekombinasyon gibi evrimsel süreçler baz alınarak algoritmalar oluşturulmaktadır.

Genetik algoritma kullanılarak bir probleme çözüm sağlanırken olası çözümler popülasyonu oluşturulur. Bu olası çözümler daha sonra doğal evrimsel süreçlerde olduğu gibi rekombinasyona ve mutasyona tabi tutulur. Bunun sonucunda evrimde olduğu gibi yeni çocuklar üretilir ve ilgili süreç oluşturulan yeni çocuklarla yeni nesil üzerinden tekrarlanır. Her bir bireye ya da her bir olası çözüme bir uygunluk değeri atanır ve uygun olan bireylere ya da uygun olan olası çözümlere daha fazla çiftleşme şansı verilerek daha fazla uyum sağlayan bireyler ya da daha fazla uyum sağlayan çözümler elde edilmeye çalışılır. Bu durum Darwin'in en iyinin hayatta kalması teoremine dayanır.

Genetik algoritmanın günümüz problemlerinde popüler olarak kullanılmasının avantajları şöyle sıralanabilir:

- Herhangi bir ikincil bilgi gerektirmez.
- Geleneksel arama metotları ile karşılaştırıldıklarında daha hızlı ve daha verimlidirler.
- Sürekli ve ayrık fonksiyonları optimize ettikleri gibi çok objektifli problemleri de optimize ederler.
- Çözülen probleme karşı sadece tek bir çözüm sunmak yerine olası iyi çözümlerin listesini sunar.
- Her zaman probleme karşı bir çözüm sunulur ve sunulan bu çözüm zamanla iyileşir.
- Arama uzayı büyüktür ve çok sayıda parametre içeren büyük veri kümelerinde daha kullanışlıdır.

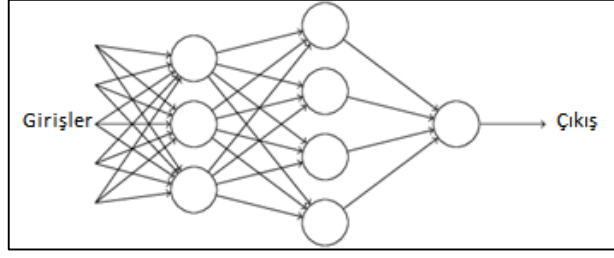
Genetik algoritmanın tüm bu avantajlarının yanı sıra her problemde özellikle basit problemlerde kullanılması uygun olmayabilir ve algoritma sonucu uygun sonuçlar vermeyebilir. Aynı zamanda her bir uygunluk (fitness) fonksiyonu değeri tek tek hesaplandığı için hesaplama süresi açısından uygun olmayabilir. Genetik algoritmalar problemlerin çözümlerine stokastik bir yaklaşım ile çözüm sunmalarından ötürü

sundukları çözümlerin kalitesi üzerinde garanti vermezler ve genetik algoritmalar doğru şekilde uygulanmazlarsa en uygun çözümleri sunamazlar.

2.4 YAPAY SİNİR AĞLARI

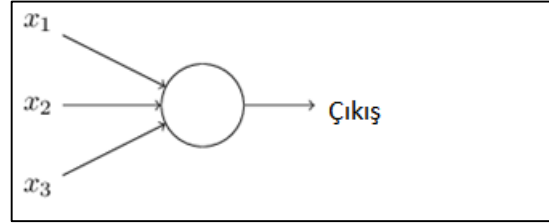
İlerleyen teknolojinin etkisi ve belirli bir probleme karşılık ona özgü uzman sistem oluşturmanın yaygınlaşmasıyla günümüz problemlerini makine öğrenmesi algoritmaları kullanarak çözümlenebilmek hem insan iş yükünü azaltmakta hem de insanların istemeden de olsa yanlış kararlar almasını önlemede önemli bir adım olarak görülmektedir. İnsanoğlu bir problem ile karşılaştığında o problemin çözümü için geçmiş deneyimlerini gözden geçirir ve mevcut sorunun çözümü için geçmiş bilgilerinden çözüm bulmaya çalışır. Makine öğrenmesi algoritmaları kullanarak oluşturulan uzman sistemlerde tıpkı insanoğlu gibi önceki bilgilerden öğrenir ve bu bilgileri kullanarak bir çıkarımda bulunur. Yapay sinir ağları, insanoğlunun öğrenme aşamalarından kimyasal öğrenme olmaksızın diğer tüm öğrenme yöntemlerini kullanarak insan beyni dışında öğrenebilmeyi ve karar verebilmeyi sağlayabilmek için biyolojik sinir ağlarını model almaktadırlar. İnsanların deneyerek, yaparak, keşfederek öğrendikleri bilgileri yapay sinir ağlarına öğretmek makinelerin insanlar gibi öğrenmesi, düşünmesi ve karar vermesi hedeflenmektedir. Yapay sinir ağları sinirlerden, sinirleri birbirine bağlayan düğüm noktalarından ya da sinirlerin birbirleri arasındaki bağlantılardan oluşmaktadırlar.

Basit bir şekilde bir yapay sinir ağı üç parçadan oluşur. Bunlar, girdi katmanı, gizli katmanlar (ara katmanlar) ve çıktı katmanıdır. Girdi katmanı girilen bilgileri toplamaktan sorumlu katmandır bu katmanda işlem gerçekleştirilmez sadece girilen bilgiler toplanarak bir sonraki katmana iletilir. Ara katmanlar ya da gizli katmanlar girdi katmanından aldıkları bilgileri işleyerek çıktı katmanına göndermekten sorumlu nöronlardan oluşurlar. Bu katman eğer çok katmanlı bir ağ ise içerisinde birden fazla gizli katman barındırıyor demektir. Ara katmanlara sahip örnek bir yapay sinir ağı yapısı Şekil 2.12' de gösterilmektedir. Çıktı katmanı ise bir önceki katmanın işlediği bilgileri alarak girdi katmanından alınan bilgilere karşılık bir çıktı üretmekten sorumludur.



Şekil 2.12 Ara katmanlara sahip örnek yapay sinir ağı yapısı.

Frank Rosenblatt ilk perceptronu nörolog Warren McCulloch ve matematikçi Walter Pitts' in sinir ağları üzerine yapmış oldukları çalışmalardan esinlenerek oluşturmuştur. Perceptronlar yapay sinir ağlarının temelini oluşturan en eski ancak en basit sinir ağı yapısıdır. Temel olarak perceptronlar birden fazla giriş değerini alıp işleyerek çıktı üretmekten sorumludurlar. Yapısı gereği bir perceptron doğrusal olarak sınıflara ayrılabilen AND, OR ya da NOT işlemleri gibi problemlerin çözümünde kullanılmaktadırlar.



Şekil 2.13 Basit bir perceptron yapısı.

Şekil 2.13' te gösterilen basit bir perceptron yapısında x_1, x_2, x_3 perceptronun giriş verilerini göstermektedir. Perceptronun her bir giriş verisinin w_1, w_2, w_3 ile temsil edildiği ağırlık değerleri bulunmaktadır. Bu ağırlık değerleri her bir girdinin çıkışa etki eden önemlerini sayısal olarak ifade etmektedirler. Bu basit perceptronun sahip olduğu nöronun çıktısının 0 veya 1 olması durumu her bir giriş değerlerinin ağırlıkları ile çarpımları toplamlarının eşik değerinden küçük eşit veya büyük olması ile belirlenir. Matematiksel olarak ifadesi ise şu şekildedir:

$$çıkı\tau = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{eşik deęer} \\ 1 & \text{if } \sum_j w_j x_j > \text{eşik deęer} \end{cases} \quad (2.29)$$

Perceptronlar doğrusal olarak sınıflara ayıramayan XOR gibi problemlerin çözümünde başarısızdırlar. Bu tarz problemleri çözebilmek için çok katmanlı ağ modeli geliştirilmiştir. Çok katmanlı algılayıcı modeli (multi layer perceptron) sıklıkla kullanılan en popüler sinir ağlarından biridir. Çok katmanlı algılayıcı modeli girdiler ve buna karşılık gelen çıkış değeri arasında önceden bilgi sahibi olunan eğitici öğrenme durumunda kullanılan bir modeldir. Bu modelde ağın öğrenbilmesi için ağa giriş değerleri ve bu giriş değerlerine karşılık gelen daha önceden elde edilmiş çıkış değerlerinin verilmesi gerekmektedir (eğitim kümesi). Böylece oluşturulan modelin, kendisine benzer örnekler verilerek daha önceden öğrenmiş olduğu örneklerden çıkarımlar yaparak uygun çözümler sunması sağlanır. Çok katmanlı algılayıcı modelinden ağ çıktısını hesaplayan ve ağın ağırlıklarını değiştiren iki adet hesaplama yöntemi bulunmaktadır. Bunlar:

- İleriye doğru hesaplama (Feed Forward)
- Geriye doğru hesaplama (Back Propagation)

İleri doğru hesaplama yönteminde perceptronlar giriş katmanı, orta ya da gizli katmanlar ve çıkış katmanı olmak üzere, katmanlar halinde ileri yönlü olarak düzenlenirler. Ağın işletimi sırasında ileri yönlü bir işlem olmasından ötürü ağa girdi olarak verilen verilerin ağırlıkları ağdan çıkış elde edilinceye kadar sabit kalır değişmez. Giriş katmanından alınan bilgileri sürekli ileri doğru olacak şekilde işlenip çıkışa iletilirler ancak aynı katmanlardaki perceptronlar arasında bağlantı bulunmamaktadır. İleri yönlü ağlarda ağın öğrenmesi ağa verilen girdi ve çıktı değerleri arasındaki ileri yönlü ilişkiye bağlı olduğu için eğitimi öğrenme olarak adlandırılmaktadır. Bu ağın giriş katmanındaki k . işlem elemanının çıktısı şu şekilde hesaplanır [37]:

$$C_k^i = G_k \quad (2.30)$$

Ara ya da gizli katmandaki her bir işlem elemanına gelen net girdi değeri ise şu şekilde hesaplanmaktadır:

$$NET_j^k = \sum_{k=1}^n A_{kj} C_k^i \quad (2.31)$$

Bu formüldeki k. giriş elemanının j. ara katman elemanına bağlayan bağlantının ağırlık değeri A_{kj} ile temsil edilmektedir. j. ara katman elemanının çıktısı ise yukarıda hesaplanan NET girdinin aktivasyon fonksiyonlarından birinin kullanılmasıyla hesaplanması sonucu elde edilir. j. ara katman elemanının çıktısını hesaplamak için kullanılacak olan aktivasyon fonksiyonunun Sigmoid olması halinde şu şekilde hesaplama yapılır:

$$\zeta = \frac{1}{1 + e^{-(NET_j^k + \beta_j^a)}} \quad (2.32)$$

Hesaplama kullanılan bu formülde β_j^a ara ya da gizli katmanda bulunan j. elemana bağlanan eşik değere sahip elemanın ağırlığını göstermektedir. Eşik değeri her zaman sabit olup 1' e eşit olmakla beraber bu değer eğitim sırasında ağ tarafından belirlenmektedir. Kullanılan sigmoid aktivasyon işleminin ardından ileri doğru hesaplama işlemi tamamlanmış olur.

Geriye doğru hesaplama yönteminde ise yapay sinir ağına girdi olarak verilen veriler sonucunda yapay sinir ağının ürettiği değer ile yapay sinir ağından beklenen değer arasında oluşan farkın hata olarak kabul edilir. Bu hatayı azaltmak için ağırlıkların her biri yeniden hesaplanır. Hatanın ağda geriye doğru yayılımı sırasında ağın girdi ağırlıkları yeniden düzenlenir ve böylece ağın çıktısının beklenen çıktıya yakınsaması amaçlanır.

Ağın çıkış katmanından m. işlem için elde edilen ağın hatası E_m ile ifade edecek olursak ağın tek bir işlemi için elde edilen hatası şöyle bulunabilir [37]:

$$E_m = B_m - \zeta_m \quad (2.33)$$

Çıkış katmanında oluşacak olan toplam hata (TH) aşağıdaki matematiksel ifade kullanılarak hesaplanabilir [37]:

$$TH = \sqrt{\sum_m E_m^2} \quad (2.34)$$

Hatanın ağda geriye doğru yayılımı esnasında ağırlıklarının değişim işlemi ya ara katman ile çıkış katmanı arasındaki ağırlıkların değiştirilmesi ile ya da ara katmanlar arası veya ara katman ile giriş katmanı arasındaki ağırlıkların değiştirilmesi ile olur.

Ara katmandaki j . işlem elemanının çıkış katmanındaki m . işlem elemanı arasındaki bağlantının değişim miktarına ΔA^a denirse herhangi bir t anında hatanın geriye doğru yayılımı ile ağdaki değişim miktarı şöyle hesaplanır [37]:

$$\Delta A_{jm}^a(t) = \lambda \delta_m \zeta_j^a + \alpha \Delta A_{jm}^a(t-1) \quad (2.35)$$

Yukarıdaki denklemde λ öğrenme katsayısını, α momentum katsayısını göstermektedir. Momentum katsayısı ağırlık öğrenmesi esnasında öğrenmenin yerel bir optimum noktaya takılıp kalmaması için ağırlık değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlar.

m . çıkışın hatası şöyle gösterilmektedir [37]:

$$\delta_m = f'(NET) E_m \quad (2.36)$$

Yukarıdaki matematiksel notasyonda gösterilen $f'(NET)$ aktivasyon fonksiyonunun türevi olmakla beraber eğer aktivasyon fonksiyonu olarak sigmoid fonksiyon kullanılırsa bu durumda m . çıkışın hatası şöyle gösterilir [37]:

$$\delta_m = \zeta_m(1 - \zeta_m)E_m \quad (2.37)$$

Ağırlıkların t . iterasyondaki değişim miktarına bağlı yeni değerlerinin hesaplanması şöyle yapılır [37]:

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t) \quad (2.38)$$

Çıkış katmanında bulunan işlem elemanlarının eşik değer aralıkları β^c ile gösterilir ve çıkış katmanının çıktısı değerinin sabit olması nedeniyle eşik değerlerinin değişim miktarı şöyle hesaplanır [37]:

$$\Delta \beta_m^c(t) = \lambda \delta_m + \alpha \Delta \beta_m^c(t-1) \quad (2.39)$$

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta \beta_m^c(t) \quad (2.40)$$

Giriş katmanı ile ara katman arasındaki ağırlıkların değişimi ΔA^i ile gösterildiğinde değişim miktarı formülü [37]:

$$\Delta A_{kj}^i(t) = \lambda \delta_j^a \zeta_k^i + \alpha \Delta A_{kj}^i(t-1) \quad (2.41)$$

Aktivasyon fonksiyonunun sigmoid seçilmesi durumunda çıkış hatası şu şekilde hesaplanır[37]:

$$\delta_j^a = \zeta_j^a (1 - \zeta_j^a) \sum_m \delta_m A_{jm}^a \quad (2.42)$$

Ağırlıkların yeni değeri ise şöyle hesaplanmaktadır [37]:

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t) \quad (2.43)$$

Ara katman eşik değer ağırlıkları β^a ile gösterilir ve eşik değer ağırlıklarının değişim miktarı şöyle hesaplanır [37]:

$$\Delta B_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1) \quad (2.44)$$

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta \beta_j^a(t) \quad (2.45)$$

Tüm bu hesaplamalarla birlikte çok katmanlı algılayıcılarda geriye doğru hesaplama yöntemi kullanılarak ağırlıklarının tümünün değiştirilmesi sağlanmış olur [37].

3. BULGULAR

Bu bölümde hedeflenen modelin oluşturulması aşamalarından sırası ile detaylı bir şekilde bahsedilecektir. Bu bağlamda öncelikle kullanılan veri kümesi ve veri kümesi içerisindeki alanlar gösterilecektir. Ardından sırasıyla veri kümesi üzerine karar ağacı analizi, destek vektör ve temel bileşenler analizi, bulanık mantık ve genetik algoritma analizi ve son olarak yapay sinir ağı analizi uygulanacaktır. Uygulanan her bir analiz ayrı başlıklar altında detaylı bir şekilde açıklanacaktır.

3.1 KULLANILAN VERİ KÜMESİ VE ALANLARI

Kredi geri ödeme durumunun analizini yapmadan önce oluşturulacak olan modelde kullanılacak veri kümesinin seçimi hesaplamaların sonuçlarını değiştireceği için çok dikkatlice yapılmalıdır. Kullanılacak olan veri kümelerinin hedefleri en az değişken ile en uygun modeli oluşturmak olmalıdır. Bu doğrultuda veri kümesinin içerisinde mümkün olduğunca eksik verilerin olmaması, analizi yapmaya yetecek derecede anlamlı ve yeterli sayıda veri içermesi gerekmektedir. Aynı zamanda analiz modeli oluşturulurken modelde yer alacak bağımsız değişkenler arasında yüksek dereceli korelasyon varlığı da incelenmelidir.

Bu bölümde hedeflenen modelin oluşturulmasında ilk aşama olarak veri kümesine karar ağacı analizi uygulanacaktır. Analizlerde kullanılacak veri kümesi UCI Machine Learning Repository' nin açık erişimde bulunan sitesinden indirilmiştir. İndirilen veri kümesi içerisinde 21 adet alan bulunmaktadır ve bu alanların tanımları Tablo 3.1' de yapılmıştır.

Tablo 3.1 Veri kümesi içerisindeki alan ve açıklamaları.

Veri Kümesi İçerisindeki Alan	Açıklaması / Tanımı
Account Balance	Hesap Bakiyesi
Duration of Credit (month)	Kredi Süresi (Ay)
Payment Status of Previous Credit	Kredi Geçmişi
Purpose	Kredi Alış Amacı
Credit Amount	Kredi Miktarı
Value Savings/Stocks	Tasarruf Hesapları
Length of current employment	Çalışma yılı
Instalment per cent	Mevcut çalışma süresi
Sex & Marital Status	Cinsiyet ve Medeni Hali
Guarantors	Garantörler
Duration in Current address	Mevcut Adresinde Bulunma Süresi
Most valuable available asset	Varlıklar
Age (years)	Yaş
Concurrent Credits	Eşzamanlı Krediler / Diğer Krediler
Type of apartment	Ev Durumu (Kiracı / Ev Sahibi/ Diğer)
No of Credits at this Bank	Bu Bankada Varolan Önceki Kredileri
Occupation	Mesleği
No of dependents	Bakmakla Yükümlü Olduğu Kişi Sayısı
Telephone	Telefon (Kendi Adına Kayıtlı / Yok)
Creditability	Kredi Verilebilirlik
Foreign Worker	Yabancı Çalışan

3.2 VERİ KÜMESİNE KARAR AĞACI ANALİZİNİN UYGULANMASI

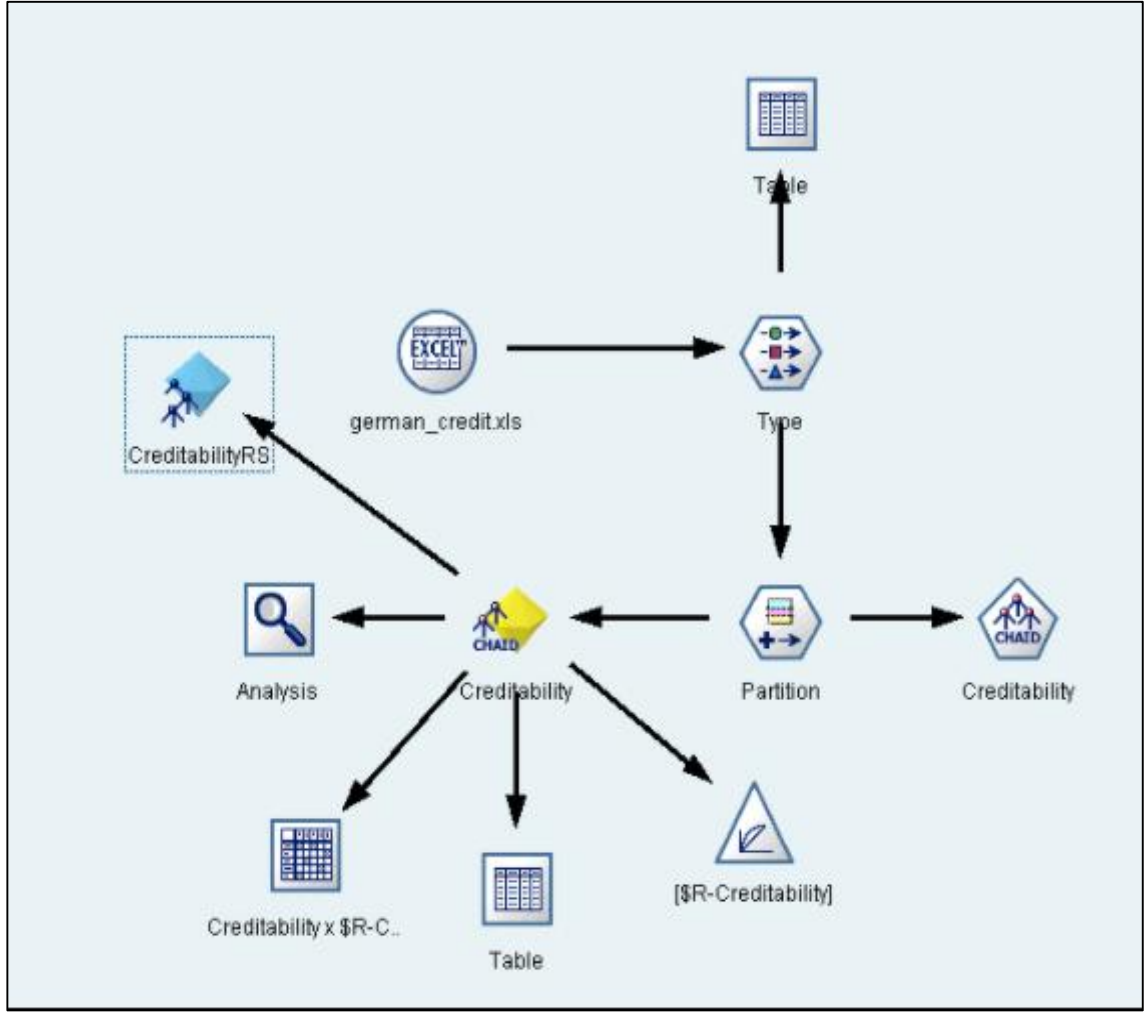
Kullanılan veri kümesi ile karar ağacı analizi yapabilmesi için bağımlı ve bağımsız değişkenlere ihtiyaç duyulmaktadır. Model oluştururken veri kümesi içerisinde seçilen bağımlı değişkene etki edebilecek faktörlerin iyi belirlenmesi gerekmektedir. Yukarıda tablo içerisinde açıklanmış olan alanlar içerisinde kredi verilebilirlik (Creditability) alanı sınıflandırma algoritması olarak kullanılan CHAID algoritması tarafından bağımlı değişken (dependent variable) olarak kullanılacaktır. Veri kümesi içerisindeki geriye kalan diğer tüm alanlar ise algoritma tarafından bağımsız değişkenler (independent variables) olarak kullanılacaktır. Bu şekilde belirlenmiş olan bu 20 adet bağımsız değişkenin kredi başvurusunda bulunan müşterilerin kredi verilebilirliği üzerinde belirleyici etkilerinin olduğu varsayılacaktır. Böylelikle yapılacak olan sınıflandırma analizinde bir adet bağımsız değişken ile birlikte yirmi adet bağımlı değişken kullanılacaktır. Yüksek başarımlı bir model oluşturabilmek için daha öncede değinildiği

gibi modele daha fazla deęişken ve veri eklemek yerine mümkün olduęunca az deęişken ve yeterli sayıda veri ile model oluřturmaya alıřmak en doęrusu olacaktır.

Karar aęacı ğrenme algoritmalarının veri kümeleri üzerinde analizlerinin yapılması ve analiz sonuçlarının yorumlanması kolay ve anlaşılır olduęundan en ok kullanılan sınıflandırma uygulamalarından biridir. Genel olarak bir karar aęacı algoritması en tepedeki düęümünden en ařaęıdaki düęüme kadar yinelemeli olarak bölümlenme işleme dayanır. Eęer kök düęüme gelene kadar tüm deęişkenler kullanılmışsa ve kullanılacak başka deęişken kalmamışsa bu bölümlenme işlemi durdurulur ya da kök düęüme gelene kadar tüm örnekler aynı sınıfa dahil edilmişse bölümlenme işlemi durdurulur ve böylelikle analiz sonucunda oluřturulan aęaç yapısının en tepedeki düęümünden kök düęümüne kadar kural tablosı çıkarılır. Her bir kural “Eęer - ise” (If - then - else) yapısı ile oluşur. Son adım olarak en dıřta bulunan kök düęümlerde oluřturulacak modelin sınıflandırma tahminlerini içerir.

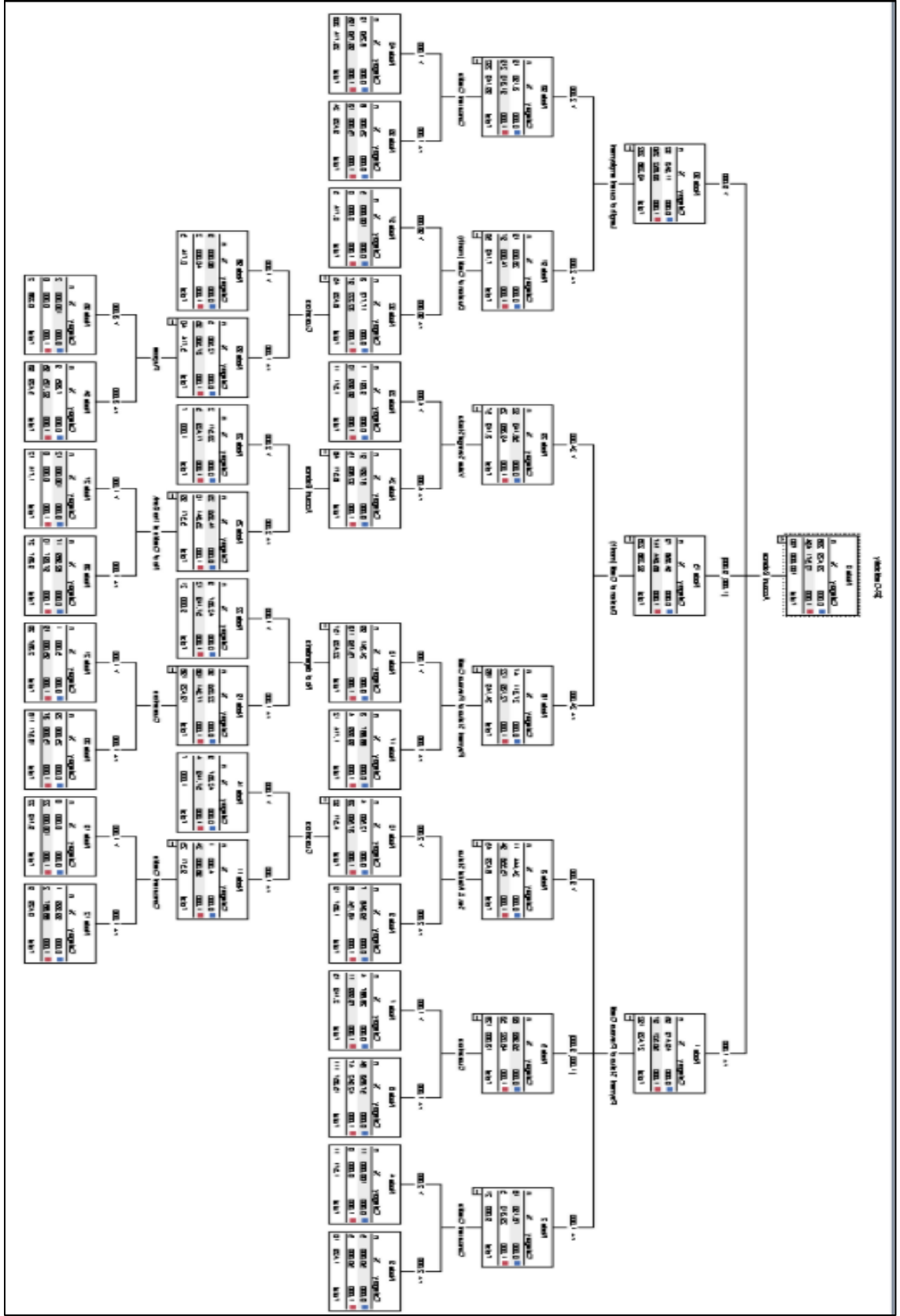
Veri kümesinin karar aęacı algoritmasına göre sınıflandırma analizini yapabilmek için öncelikle SPSS Clementine programına veri kümesinin yüklenmesi gerekmektedir. Yüklenen veri kümesi analize başlamadan önce baęımlı deęişken alanına kredi verilebilirlik (Creditibility) alanını verip kalan dięer deęişkenlerde baęımsız deęişkenler alanına eklendi. Ardından veri kümesinin %70’ i eğitim (train) ve %30’ u ise test olarak bölümlendirildi.

řimdiye kadar SPSS yazılımı ile oluřturulan sınıflandırma modelinin genel yapısı ise řekil 3.1 de gösterilmektedir.



Şekil 3.1 Karar ağacı analizi için kurulan modelin genel yapısı.

Analiz çalıştırdıktan sonra oluşturulan karar ağacı yapısı ise Şekil 3.2 de gösterilmektedir.



Şekil 3.2 Analiz sonucu elde edilen ağaç yapısı.

Yukarıda oluşturulan modelin analizi sonucunda kullanılan yazılım ile 22 adet kural kümesi elde edilmiştir. Kural kümesi içerisinde kredi verilebilirlik (Creditability) alanının 0 olduğu duruma karşılık 10 adet kural kümesi elde edilmekle birlikte kredi verilebilirlik (Creditability) alanının 1 olduğu duruma karşılık ise 12 adet kural kümesi elde edildi.

Karar ağacı algoritmalarından biri olan CHAID analizi yönteminin EXCHAID yöntemi kullanarak yapılan analiz sonucunda elde edilen model veri kümesinden eğitim (training) olarak ayrılan kısmın %79,14' ünü doğru tahmin ederek doğru sınıfa dahil etmiştir. Buna karşılık bu model eğitim (training) olarak ayrılan veri kümesinin %20,86 ' sını ise yanlış tahminleme yaparak yanlış sınıflandırmıştır. Aynı şekilde oluşturulan model test kümesine uygulandığında test verilerinin %72' sini doğru tahminleyerek doğru sınıfa dahil ettiği görülmüştür. Buna karşılık model test verilerinin %28 ' ini ise yanlış tahminleyerek yanlış sınıflandırmıştır. Modelin analizinin sonuçları Tablo 3.2 de gösterilmektedir.

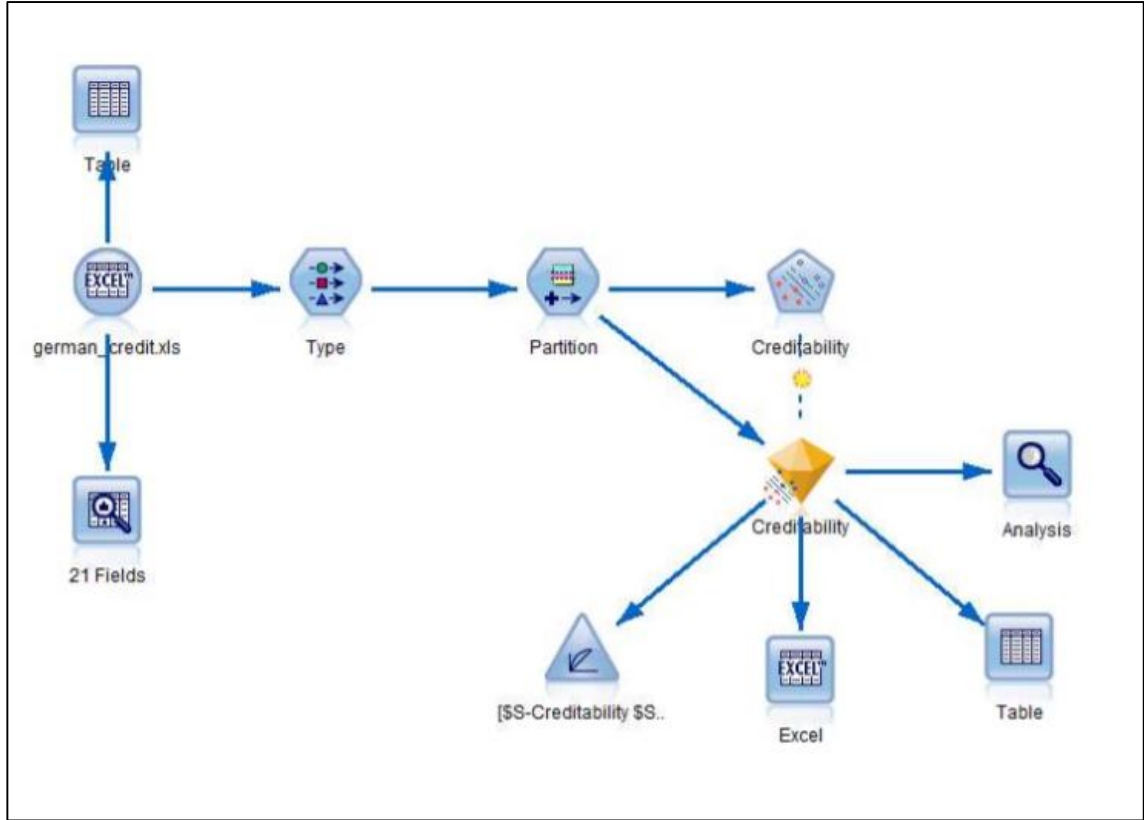
Tablo 3.2 EXCHAID analizinin sonuçları.

Bölümlendirme	Doğruluk (%)	Duyarlılık (%)
Eğitim Verileri	%79.14	%83
Test Verileri	%72	%75

3.3 VERİ KÜMESİNE DESTEK VEKTÖR MAKİNELERİ VE TEMEL BİLEŞENLER ANALİZİNİN UYGULANMASI

Bu bölümün ikinci aşamasında, karar ağacı algoritmasında kullanılan veri kümesine sınıflandırma algoritmalarından biri olan destek vektör makineleri algoritmaları analizi uygulanmıştır. Günümüzde destek vektör makineleri yöntemi birçok gerçek hayat problemlerinin çözümüne uygulanmaktadır. Nedeni ise gerçek yaşam problemlerinin birçok farklı bileşenden oluşması ve değişkenlik gösteren bu bileşenlerin bazen doğrusal olarak sınıflandırılmamasıdır. Bu durumun çözümü olarak destek vektör makinelerinin sınıflandırılmaz bileşenlerin sınıflandırma özelliğinden yararlanır.

Destek vektör makineleri analizine başlamadan önce veri kümesi içerisindeki 21 adet değişkenden kredi verilebilirlik (creditability) alanını bağımlı değişken olarak belirlenmiştir. Ardından geri kalan 20 adet alanı ise bağımsız değişken olarak belirlenmiştir. Şekil 3.3 de destek vektör makineleri analizi için oluşturulan modelin genel yapısı gösterilmektedir.



Şekil 3.3 Destek vektör analizi için kurulan modelin genel yapısı.

İlk aşama olarak destek vektör makineleri analizinde 1 adet bağımlı ve 20 adet bağımsız değişken kullanıldı. 1000 kişinin verisini içeren veri kümesinin %70' i eğitim verisi ve %30' u test verisi olarak ayırdıktan sonra veri kümesine destek vektör analizi uygulandı. Destek vektör analizi sonucunda destek vektör makineleri yöntemi eğitim verilerinin %97.29' unu doğru bir şekilde sınıflandırdı. Test verilerinin ise %69.67' sini doğru bir şekilde sınıflandırdı. Tablo 3.3 de ilk aşamada yapılan destek vektör makineleri analizinin sonucu gösterilmektedir.

Tablo 3.3 Yapılan ilk Destek Vektör Makineleri analizinin sonucu.

Bölümlendirme	Doğruluk (%)	Duyarlılık (%)
Eğitim Verileri	%97.29	%89
Test Verileri	%69.67	%70

Destek vektör makineleri analizinin ikinci aşamasında ise ilk aşamada kullanılan veri kümesi üzerinde temel bileşenler analizi uygulandı. Temel bileşenler analizi sonucunda veri kümesi 21 adet değişkenden 8 adet değişkene indirildi. Elde edilen bu değişkenlerle yeniden destek vektör makineleri analizini uygulandı. Bir önceki analizde olduğu gibi veri kümesi aynı şekilde %70' i eğitim verisi %30' u ise test verisi olacak şekilde bölümlendirildi. Şekil 3.4' de temel bileşenler analizi sonucu veri kümesini 8 adet değişkene indirgeme oranları gösterilmektedir.

Component	Total Variance Explained								
	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	2.536	12.680	12.680	2.536	12.680	12.680	2.167	10.836	10.836
2	1.969	9.843	22.522	1.969	9.843	22.522	1.528	7.640	18.475
3	1.415	7.075	29.598	1.415	7.075	29.598	1.514	7.570	26.045
4	1.320	6.601	36.199	1.320	6.601	36.199	1.456	7.279	33.324
5	1.213	6.064	42.263	1.213	6.064	42.263	1.338	6.691	40.015
6	1.169	5.846	48.109	1.169	5.846	48.109	1.310	6.549	46.564
7	1.138	5.691	53.801	1.138	5.691	53.801	1.293	6.466	53.029
8	1.088	5.442	59.242	1.088	5.442	59.242	1.243	6.213	59.242
9	.981	4.907	64.149						
10	.889	4.447	68.597						
11	.873	4.363	72.960						
12	.809	4.043	77.003						
13	.776	3.881	80.883						
14	.758	3.791	84.675						
15	.684	3.420	88.094						
16	.633	3.165	91.259						
17	.531	2.655	93.913						
18	.493	2.467	96.381						
19	.464	2.319	98.700						
20	.260	1.300	100.000						

Extraction Method: Principal Component Analysis.

Şekil 3.4 Temel bileşenler analizi sonucu indirgenen değişkenler.

Temel bileşenler analizi yöntemi kullanarak yapılan destek vektör makineleri analizi eğitim verilerinin %90' ının doğru bir şekilde sınıflandırımıştır. Test verilerinin ise %73' ünün doğru bir şekilde sınıflandırımıştır. Tablo 3.4 de temel bileşenler analizi sonucu elde edilen indirgenmiş değişkenlerle yapılan destek vektör makineleri analizinin sonucu gösterilmektedir.

Tablo 3.4 Temel Bileşenler Analizi ile indirgenen değişkenlerle yapılan Destek Vektör Makineleri analizi sonucu.

Bölümlendirme	Doğruluk (%)	Duyarlılık (%)
Eğitim Verileri	%90	%95
Test Verileri	%73	%74

3.4 VERİ KÜMESİNE BULANIK MANTIK VE GENETİK ALGORİTMA ANALİZİNİN UYGULANMASI

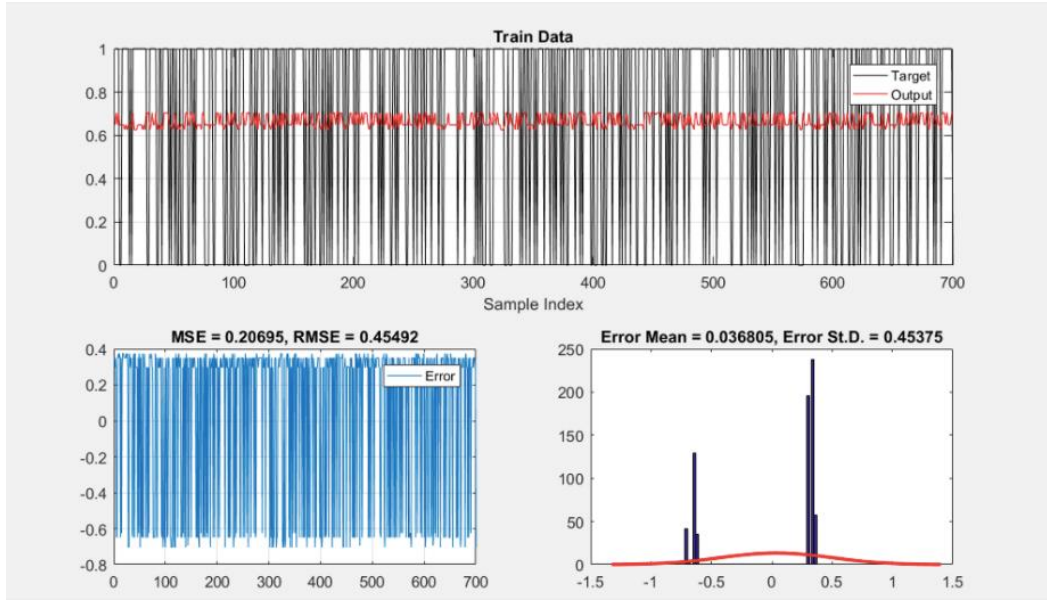
Veri kümesi üzerine destek vektör makineleri analizi uyguladıktan sonra bu aşamada önceki analizlerde kullanılan veri kümesine Matlab yazılımı kullanılarak bulanık mantık ile genetik algoritma analizi uygulandı. Önceki analizlerde yapıldığı gibi veri kümesi eğitim verileri ve test verileri olacak şekilde bölümlendirildi. Bu analizde de giriş parametreleri olarak 1 adet bağımlı değişken ile 20 adet bağımsız değişken kullanılarak analiz gerçekleştirildi.

Matlab ile, bulanık mantık ile genetik algoritma analizini yapabilmek için kullanılan kodlar yarpiz* sitesinde akademik ve profesyonel alanda bilimsel çalışmalar yapmak için açık kaynak kodlu olarak paylaşılan matlab ile genetik algoritma analizi kodlarından yararlanıldı. Kullanılan matlab kodları Ek 1' de paylaşılmıştır.

Bulanık mantık ile genetik algoritma analizinin sonucunda analizi yapılan eğitim ve test verilerinin standart sapması 0.45 olarak çıkarken, eğitim ve test verilerinin Ortalama kareler hatası (Mean Squared Error) 0.20 olarak çıkmıştır. Ortalama kareler hatası istatistikte tahmin edilen modelin hassaslığını, doğruluğunu ve performansını ölçmek için kullanılan önemli bir kriterdir. Ortalama kareler hatası tahmin yapan modelin regresyon çizgisinin ne kadar yakınına geldiğini gösterir ve bu durumu regresyon çizgisine kadar noktalar arasındaki mesafelerin karelerini alarak yapar. Kare alma işlemi negatif uzaklıkları pozitif çevirmek için gerekli bir işlemdir. Ortalama kareler hatası ne kadar küçük ise regresyon çizgisine o kadar yakınız demektir. En basit anlatımla ortalama kareler hatası tahmin edilen parametre ile gözlenen parametre arasındaki karesel farkın ortalaması olarak ifade edilebilir. Şekil 3.5 ve Tablo 3.5 de eğitim verilerinin bulanık mantık ile genetik algoritma analizinin sonucu gösterilmektedir.

* <http://yarpiz.com/>

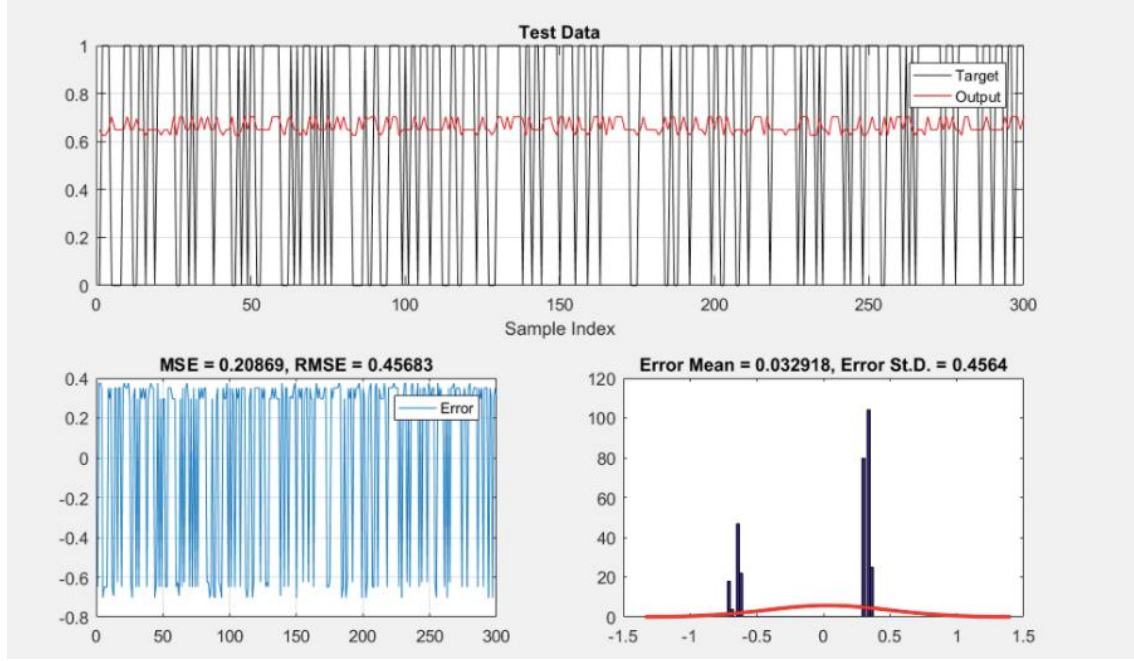
Benzer olarak Şekil 3.6 ve Tablo 3.6 da test verilerinin bulanık mantık ile genetik algoritma analizinin sonucu gösterilmektedir.



Şekil 3.5 Eğitim verilerinin bulanık mantık ve genetik algoritma ile analizi sonucu.

Tablo 3.5 Eğitim verilerinin bulanık mantık ve genetik algoritma ile analizi sonucu.

Bölümlendirme	Adet	Ortalama Kareler Hatası (MSE)	Standart Sapma (Error Std. D.)
Eğitim Verileri	700	0.20	0.45



Şekil 3.6 Test verilerinin bulanık mantık ve genetik algoritma ile analizi sonucu.

Tablo 3.6 Test verilerinin bulanık mantık ve genetik algoritma ile analizi sonucu.

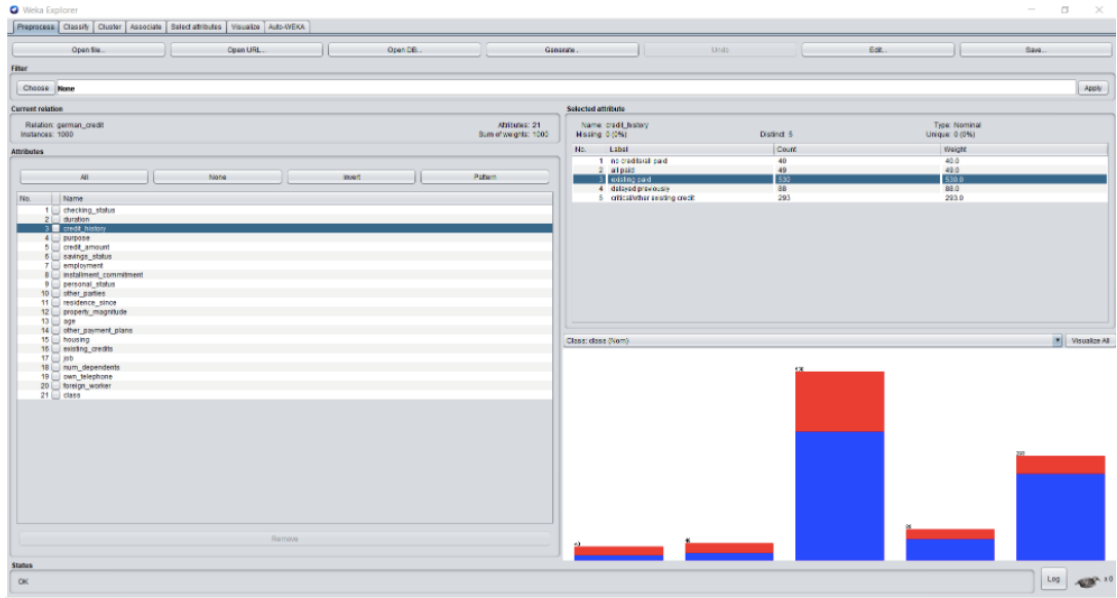
Bölümlendirme	Adet	Ortalama Kareler Hatası (MSE)	Standart Sapma (Error Std. D.)
Eğitim Verileri	300	0.20	0.45

3.5 VERİ KÜMESİNE YAPAY SINIR AĞI ANALİZİNİN UYGULANMASI

Veri kümesi üzerine yukarıda yapılan karar ağaçları, destek vektör makineleri ve bulanık mantık ile genetik algoritmaları analizleri uyguladıktan sonra son aşama olarak kullanılan veri kümesine yapay sinir ağları analizi Weka* yazılımı kullanılarak uygulandı. Daha önceden yapılan analizlerde de olduğu gibi veri kümesi eğitim ve test verileri olmak üzere bölümlendirildi. Ancak yapay sinir ağları analizini bölümlendirilen veriler üzerinde yapmadan önce yapay sinir ağı analizinin performansını ölçebilmek için öncelikle veri kümesinin tamamını yapay sinir ağı analizi yapıldı ardından veri kümesi bölümlendirip bölümlendirilen kısmın analizi yapıldı.

* <http://www.cs.waikato.ac.nz/ml/weka/>

Weka programı Waikato üniversitesi tarafından geliştirilmiş veri madenciliği uygulamaları için machine learning algoritmalarını çalıştırmasını sağlayan java programlama dili ile yazılmış açık kaynak kodlu bir yazılımdır. Weka içerisinde bulunan sınıflandırma, regresyon, kümeleme, data ön işleme ve birliktelik kuralları gibi algoritmalar kullanılarak analiz yapılabileceği gibi java programlama dili kullanılarak yeni bir makine öğrenmesi algoritması yazılabilir. Weka ile yapay sinir ağı analizini gerçekleştirmeden önce veri kümesinin Weka uygulamasına yüklenmesi gerekmektedir. Şekil 3.7 de kullanılan veri kümesinin Weka yazılımı içerisinde yüklenmesi gösterilmektedir.



Şekil 3.7 Veri kümesinin weka' ya yüklenmesi.

Kullanılan veri kümesinin weka' nın anlayacağı arff tipine çevrilip weka' ya yüklenmesinden sonra ilk aşama olarak tüm veriler ile geriye doğru hata yayılım yöntemi (backpropagation) ile çok katmanlı algılayıcılar (multilayerperceptron) analizi algoritması çalıştırıldı. Yapay sinir ağları analizi yaparken dikkat edilecek en önemli noktalardan bir tanesi oluşturulacak olan modelin giriş parametrelerinin en uygun şekilde ayarlanmasıdır. En uygun model oluşturulmadan önce bir kaç deneme yapıp en uygun analiz sonucunu yakalamak gerekir. Bu aşamada en uygun parametre değerlerinin bulunarak modele uygulanabilmesi için bir kaç kez modeli oluşturmayı sıladıktan sonra en uygun parametre değerleri tüm veri kümesi için ilk aşamada şu

şekilde ayarlanmıştır. Öğrenme oranı (learning rate) 0.1 momentum 0.3 yığın boyutu (batchSize) 100 eğitim zamanı (training time) 5000 vererek gizli katman (hidden layer) sayısını ise J. Heaton' ın "Introduction to Neural Networks with Java" adlı kitabında da bahsettiği gibi girişteki tüm parametrelerin örneklemlerinin (instance) toplam sayılarının 2/3 katı artı çıktı katmanı kadar olacak şekilde ayarlanmıştır.

Belirlenen bu parametre değerleri ile tüm veri kümesinin yapay sinir ağı analizi sonucunda %99.4 doğruluk oranında doğru bir şekilde sınıflandırdı. Bu analiz ile oluşturulacak olan modelin ilk aşaması gerçekleştirildi. Tüm veri kümesinin yapay sinir ağı ile analizinin detaylı sonuçları Tablo 3.7' de gösterilmektedir.

Tablo 3.7 Tüm veri kümesinin yapay sinir ağları analizi sonucu.

Doğru Sınıflandırılmış Örnekler (Correctly Classified Instances)	994	99.4 %
Yanlış Sınıflandırılmış Örnekler (Incorrectly Classified Instances)	6	0.6 %
Kappa İstatistiği (Kappa statistic)	0.9857	-
Ortalama Kareler Hatası (Mean absolute error)	0.0079	-
Ortalama Karekök Hatası (Root mean squared error)	0.0775	-
Mutlak Bağlı Hatası (Relative absolute error)	1.884 %	-
Mutlak Bağlı Karekök Hatası (Root relative squared error)	16.9162 %	-
Toplam Örneklem Sayısı (Total Number of Instances)	1000	-
Duyarlılık (Sensitivity)	%99	-

İkinci aşama olarak veri kümesi eğitim verileri ve test verileri olmak üzere bölümlendirildi. Giriş parametrelerinden öğrenme oranı (learning rate) 0,1 olarak ayarlandı. Momentum' u 0,3 olarak ayarlandı. Yığın boyutu (batchSize) 100 olarak ayarlandı. Eğitim zamanı (training time) 50000 olarak ayarlandı. Gizli katman (hidden layer) sayısı ise bir önceki yapay sinir ağları analizinde de olduğu gibi girişteki tüm parametrelerin örneklemlerinin (instance) toplam sayılarının 2/3 katı artı çıktı katmanı kadar olacak şekilde ayarlandı. Bu yöntem diğer analizlerde de kullanılan veri kümesine uygulandı. 43 adet hidden layer olması gerektiği kanısına bir kaç adet analiz yapıldıktan

sonra varılmıştır. Gizli katman (hidden layer) sayıları 43 adetten az olacak şekilde ya da 43 adetten fazla olacak şekilde ayarlanıp model analizi yapıldıktan sonra sonuçların hedeflenenden çok daha düşük çıktığı gözlemlenmiştir. Girilen bu parametre değerleri ile bölümlendirilen veri kümesinin yapay sinir ağı analizi sonucunda modele eğitim verileri dışında kullanılan test verileri verilip daha önceden öğrenilen model kullanılarak analiz gerçekleştirilmiştir. Bu analiz ile test verilerinin %76' sının doğru bir şekilde sınıflandırıldığı gözlemlendi. %24' ünün ise yanlış sınıflandırıldığı gözlemlendi. Sınıflandırılan 200 adet test verisinin tahmin yüzdeleri ise Ek 2' de paylaşılmıştır. Bölümlendirilmiş veri kümesinin yapay sinir ağı ile analizinin detaylı sonuçları Tablo 3.8 de gösterilmektedir.

Tablo 3.8 Bölümlendirilmiş veri kümesinin yapay sinir ağı analizi sonucu

Doğru Sınıflandırılmış Örnekler (Correctly Classified Instances)	152	76 %
Yanlış Sınıflandırılmış Örnekler (Incorrectly Classified Instances)	48	24 %
Kappa İstatistiği (Kappa statistic)	0.3919	-
Ortalama Kareler Hatası (Mean absolute error)	0.2701	-
Ortalama Karekök Hatası (Root mean squared error)	0.4762	-
Mutlak Bağlı Hatası (Relative absolute error)	66.2521 %	-
Mutlak Bağlı Karekök Hatası (Root relative squared error)	108.3357 %	-
Toplam Örneklem Sayısı (Total Number of Instances)	200	-
Duyarlılık (Sensitivity)	%80	-

4. SONUÇ

Çalışmanın bulgular kısmında kredi almak için bankaya başvuran müşterilerin önceki kredi bilgilerinden yola çıkılarak istenen kredinin geri ödeme durumlarını etkileyen faktörleri belirleyebilmek ve bu faktörlerin en iyi şekilde belirlenmesine yardım edecek en uygun analiz yöntemini bulmak hedeflenmiştir. İstatistik ve veri madenciliği tekniklerinden biri olan karar ağaçları algoritmalarının exhaustif analiz (Exhaustive Chi-squared Automatic Interaction Detection), sınıflandırma metodlarından SVM (destek vektör makineleri) analizi, bulanık mantık ile genetik algoritma analizi ve son olarak yapay sinir ağları analizi uygulanmıştır. Yapılan bu dört analiz yönteminin, verileri doğru sınıflandırma oranlarının sonuçları Tablo 4.1 ve Tablo 4.2 de detaylı bir şekilde paylaşılmıştır.

Eğitim verileri ile yapılan analizlerin sınıflandırma sonuçları aşağıda bulunan Tablo 4.1’ de gösterilmektedir.

Tablo 4.1 Eğitim verilerinin sınıflandırma sonuçları.

Analizler	Doğru	Yanlış
Karar Ağacı (Exchaid)	%79.14	%20.86
SVM (Destek Vektör Makineleri)	%90	%10
Bulanık Mantık Genetik Algoritma	%55	%45
Yapay Sinir Ağları	%99.4	%0.6

Test verileri ile yapılan analizlerin sınıflandırma sonuçları aşağıda bulunan Tablo 4.2’ de gösterilmektedir.

Tablo 4.2 Test verilerinin sınıflandırma sonuçları

Analizler	Doğru	Yanlış
Karar Ağacı (Exchaid)	%72	%28
SVM (Destek Vektör Makineleri)	%73	%27
Bulanık Mantık Genetik Algoritma	%55	%45
Yapay Sinir Ağları	%76	%24

Genel olarak tüm modeller ayrı ayrı incelendiklerinde başvurdukları kredinin geri ödeme durumlarını etkileyen faktörler arasında müşterilerin mevcut hesap bakiyeleri (Account Balance), istemiş oldukları kredi miktarı (Credit Amount), başvurdukları

kredinin taksitlerinin sayısı (Duration of Credit), geçmiş kredilerinin ödeme durumu (Payment Status of Previous Credit), başvuru kredinin taksit miktarı (Installment), müşterinin mevcut hesabında bulunan varlıklarının durumu ve miktarı (Value Savings/ Stocks), kredinin alınış amacı (Purpose) ve mevcut çalışma süresi (Length of Current Employment)' nin kredi geri ödeme durumunu etkileyen önemli faktörler oldukları gözlemlenmiştir.

Yapılan dört farklı analiz özelinde, her bir analizin veri kümesi ile ilk aşamada yapılan analizlerle mevcut verilerle oluşturulan modeller eğitilmiştir. Ardından eğitilen modeller ile ikinci bir analiz gerçekleştirilmiştir. Buradaki amaç eğitilen modellere eğitim verisi olarak kullanılan veriler dışında verilerin verilmesi ve bu durumda verileri hangi doğrulukta sınıflandıracaklarının gözlemlenerek modellerin doğruluğunun sınanmasıdır. Bu analizleri gerçekleştirebilmek amacıyla her bir analiz için veri kümeleri, eğitim ve test verileri olacak şekilde bölümlendirilerek kullanıldı. Böylelikle tüm analizlerde hem eğitimli öğrenme yönteminin kullanılarak analizleri gerçekleştirildi hem de modellerin doğruluk oranları test verileri kullanılarak sılandı. Yapılan karar ağacı analizi, destek vektör makineleri analizi, bulanık mantık ile genetik algoritma analizi ve yapay sinir ağları analizi arasında modellerin kullanmış oldukları verileri doğru sınıflandırma yüzdeleri arasında yapay sinir ağları modelinin en yüksek doğruluk oranına sahip olduğu gözlemlenmiştir.

İlk aşamada yapılan karar ağacı analizi yöntemi ile veri kümesi içerisinde bulunan müşterilerin önceki kredi ödeme durumlarını içeren kredi verilebilirlik (creditability) alanı kullanılarak müşteriler bu model yardımı ile sınıflandırılmıştır. Karar ağacı analizinin, sınıflandırma algoritmaları içerisinde uygulanmasının kolay olması ve anlaşılabilir yorumlanmasının basitliği, bu yöntemi en sık kullanılan yöntemlerden biri haline getirmiştir. Karar ağacı analizinin bu özelliklerinin yanı sıra bir çok dezavantajı da bulunmaktadır. Giriş verileri içerisinde bir verinin değişmesi veya önemsiz bir değişkenin değişmesi bile oluşturulan ağaçta büyük değişikliklerin yapılmasına neden olarak ağaç yapısının yeniden çizilmesini gerektirmektedir. Bu durum zaman almasının yanı sıra kaynakların yoğun kullanılmasına da neden olmaktadır. Karar ağacı analizinin bir diğer önemli dezavantajı ise kullanılan veri kümesinin büyüklüğü ile analizle oluşturulan ağaç yapısının giderek karmaşıklaşmasıdır. Burada kullanılan veri kümesi

1000 kişinin verisini içermektedir. Kullanılan veri kümesi bankacılık alanında yapılan istatistiki analiz çalışmalarına nazaran çok büyük sayılmadığından karar ağacı analizi ile oluşturulan ağaç yapısı çok karmaşık değildir. Ancak hangi analiz modeli kullanılacak olursa olsun kurulan modelden iyi sonuçlar almak veri kümesi içerisinde mümkün olduğunca çok örneklemin bulunması ve mümkün olduğunca az değişken ile analizin yapılmasına bağlıdır. Burada 1000 kişilik veri içeren küme ile yapılan analizle modelin eğitim verileri dışında kullanılan verileri sınıflandırma yüzdesi olarak %72 oranını elde edildi. İlerleyen çalışmalarda karar ağacı analizi modelinin tahmin yüzdesini arttırmak için içerisinde daha fazla verinin bulunduğu bir veri kümesi kullanılması gerekir. Ancak bu durumda da oluşturulacak olan ağaç yapısının karmaşıklaşmasından ötürü analizin yorumlanması zorlaşacaktır.

İkinci analiz olan destek vektör makineleri analizinde bir önceki analizde de olduğu gibi veri kümesi içerisindeki müşterilerin önceki kredi ödeme durumlarını içeren kredi verilebilirlik (creditability) alanını kullanarak müşteriler bu model ile sınıflandırıldı. Destek vektör makineleri eğitilmiş öğrenme algoritma tekniği popüler olarak kullanılan bir diğer sınıflandırma yöntemidir. Destek vektör makineleri ile analiz yapmanın popüler olmasının nedenleri arasında yapay sinir ağı algoritmalarının eğitilmesinden daha kolay olarak eğitilmeleri ve yapay sinir ağı algoritmalarının sahip oldukları yerel optimum (local optimum) değerlerine sahip olmamalarıdır. Destek vektör makineleri algoritmasının yapay sinir ağı algoritmalarına karşın sahip oldukları bu üstün özelliklerinin yanı sıra en büyük dezavantajları arasında destek vektör makinelerinin en uygun ayırıcı çizgisi olan süper düzlem'i (hiperlane) bulabilmek için en uygun parametrelerinin seçilmesindeki zorluk olmakla beraber destek vektör makineleri analizi yaparken seçilecek olan çekirdek fonksiyonunun (kernel function) hangisinin seçilmesi gerektiğinin belli ve kesin bir kuralının olmamasıdır. Destek vektör makinelerinin sahip olduğu tüm bu avantaj ve dezavantajları ile birlikte yapılan analizde çekirdek fonksiyonu olarak bir çok çalışmada sıklıkla kullanılan Radyal Taban Fonksiyonlu çekirdek fonksiyonunu kullanılarak analiz yapıldı. Destek vektör makineleri analizinin ilk aşamasında veri kümesi içerisindeki 21 adet değişken kullanılarak analiz yapıldı. Ancak destek vektör analizi algoritmasının test verilerini doğru sınıflandırma yüzdesini arttırmak için veri kümesine öncelikle temel bileşenler analizi uygulandı. Ardından destek vektör makineleri analizinde kullanılacak değişken sayısını azaltılarak mümkün

olan en az deęişken ile en uygun sonucun bulunması hedeflendi. Temel bileşenler analizi sonucunda elde edilen deęişkenler ile yapılan destek vektör makineleri analizi sonucunda eęitilen modelin eęitim verileri olarak kullanılmayan test verilerini bir önceki destek vektör makineleri analizine göre %4 arttırarak %73 oranında doęru sınıflandırdığı gözlemlendi.

Bulanık mantık ile genetik algoritma analizinde bulanık mantık algoritmasının dięer yöntemlere göre daha az veriye ihtiyaç duyduğundan ve analizi yapılacak modelin eęitilmesine olanak vermesinden ötürü bulanık mantık yönteminin kullanılması kararlaştırılmıştır. Bulanık mantık algoritmasının bu avantajının yanı sıra karar ağacı algoritmalarında olduğu gibi yorumlanmaları ve anlaşılabilirlikleri kolaydır. Kısaca bulanık mantık ile analiz modeli oluşturmak hem doęal dil kurallarına yakın bir model oluşturulmasını sağlar hem de model sonucunda oluşturulan kural kümeleri ile gerçek dünyada doęal dil kuralları ile elde edilen dilsel sonuçlara yakın sonuçlar elde edilmesini sağlar. Genetik algoritma yöntemi bir çok stokastik optimizasyon problemlerinin çözümünde kullanılan bir metod olmakla beraber genetik algoritma ile oluşturulan model içerisinde algoritmanın sahip olduğu yerel minimum (local minimum) ve yerel maksimum (local maximum) 'a sahip olduğu durumlarda bile iyi sonuçlar verebilme özelliğine sahiptir.

Genetik algoritmaların dięer stokastik analiz yöntemlerine göre en önemli avantajlarından bir tanesi genetik algoritma analizinin paralel olarak çalıştırılabilmesidir. Bu durum dięer stokastik yöntemlerle yapılan analiz süresini azaltmakla beraber işlemci maliyetini azaltmaktadır. Tüm bu avantajlarının yanı sıra genetik algoritmanın sahip olduğu yerel maksimum ve minimum noktalarına sahip olmasından ötürü genel maksimum noktasını bulmayı garanti edemez. Bir dięer önemli husus ise genetik algoritma ile analiz gerçekleştirirken genetik algoritma parametrelerinin seçiminin yanı sıra hangi genetik kodlamanın seçileceğine dair kesin bir kural olmamasından ötürü deneme yanılma yöntemi ile uygun modelin kurulması zaman alan bir durumdur. Üçüncü analiz olan bulanık mantık ile genetik algoritma analizinde yapılan dięer analizlerde de kullanılan veri kümesi eęitim ve test verileri olmak üzere bölümlendirdikten sonra Matlab yazılımı kullanılarak analiz modeli oluşturulmuştur. Yapılan bulanık mantık ile genetik algoritma analizi sonucunda eęitim

ve test verilerinin standart sapması 0.45 olarak çıkarken eğitim ve test verilerinin ortalama kareler hatası (Mean Squared Error) 0.20 olarak çıkmıştır.

Veri kümesi ile yapılan son analiz olan yapay sinir ağları analizinin geriye doğru hata yayılım yöntemi (backpropagation) ile çok katmanlı algılayıcılar (multilayerperceptron) analizini gerçekleştirilmiştir. Yapay sinir ağları analizini kullanma avantajının en başında günümüzde sıklıkla analizleri yapılan görüntü, ses, metin tanıma ve zaman serilerinin gelmesiyle birlikte bir çok problemin çözümünde popüler olarak kullanılmalarından ötürü geniş bilgiye sahip olması ve bir çok alanda yapılmış çalışmaya kolaylıkla erişilebilinmesidir. Popüler olarak kullanılmasının en önemli sebeplerinden biri yapay sinir ağlarının karmaşık verilerden anlamlı sonuçlar çıkarma kabiliyetinin güçlü olmasıdır. Yapay sinir ağları analizinin bir çok alana uygulanıyor olması büyük bir avantaj olmakla beraber her problemin çözümü için uygun çıktılar üretememesinden ötürü yapay sinir ağları analizinin kullanımını dezavantaja çevirmektedir.

Yapay sinir ağı analizini yapmaya başlarken öncelikle kullanılan veri kümesi içerisinde müşterilerin önceki kredi durumlarını gösteren alanın olmasından ötürü yapılacak analizin eğitimli öğrenme (supervised learning) yöntemini kullanmasının uygun olacağı düşünülerek bu yöntem kullanılmıştır. Ardından ilk aşamada yapay sinir ağı varolan veriler ile eğitilerek ikinci aşamada yapay sinir ağına eğitim verileri dışında kullanılan test verileri verilerek yapay sinir ağının doğru sınıflandırma oranı sınamaya hazır hale getirilmiştir. İlk aşamada eğitilen yapay sinir ağı ile uzman bir sistem elde edilmiştir.

Hazırlanan bu uzman sistemin ikinci aşamasında bölümlendirilen veri kümesi üzerinde kullanılarak farklı verilerin “eğer olursa” tarzı sorunlarını çözümlenmesinde kullanıldı. Gerçekleştirilen yapay sinir ağı analizinin yapılan diğer analiz yöntemleri ile karşılaştırıldığında eğitim verilerinin %99.4’ ünü doğru sınıflandırırken ikinci aşamada yapılan analiz ile test verilerinin ise %76’ sını doğru bir şekilde sınıflandırması ile analizler içerisinde en yüksek doğru sınıflandırma oranına sahip olduğu gözlemlendi. Yapay sinir ağları analizinin verileri bu denli yüksek doğruluk oranında sınıflandırmasının yanı sıra analizi yaparken geriye doğru hata yayılım yöntemi (backpropagation) seçildiği için, bu yöntem analiz zamanında artışa neden olmakla

beraber kaynak tüketimi açısından bir dezavantaj sergilemiştir. Tüm bunlara ek olarak yapay sinir ağları analiz yönteminin büyük veri kümeleri üzerinde uygulanması ile daha anlamlı sonuçlar elde edebileceğinden kullanılan veri kümesinin sadece 1000 adet kişi verisini içerdiği için elde edilen doğru sınıflandırma yüzdelerinin ilerleyen çalışmalarda daha büyük veri kümeleri kullanarak arttırılabileceği düşünülmektedir.

KAYNAKLAR

- [1] Bayrakdarođlu, E., (2009), “İMKB Şirketlerinin Hisse Senedi Getiri Başarılarının Lojistik Regresyon Tekniđi” İle Analizi, ZKÜ Sosyal Bilimler Dergisi, Cilt 5, Sayı 10, 139-158.
- [2] Girginer, N, Cankuş, B (2008) “Tramvay Yolcu Memnuniyetinin Lojistik Regresyon Analiziyle Ölçülmesi: Estram Örneđi”, Yönetim ve Ekonomi Dergisi, Cilt 15, Sayı 1, 181-193
- [3] Şerbetçi, A, Özçomak, M, (2013), “Sıralı Lojistik Regrsyon Analizi İle İstatistik ve Ekonometri Derslerinde Başarıyı Etkileyen Faktörlerin Belirlenmesi: Atatürk Üniversitesi İktisadi ve İdari Bilimler Fakültesi Öğrencileri Üzerine Bir Uygulama”, Yüksek Lisans Tezi, Atatürk Üniversitesi, Sosyal Bilimler Enstitüsü.
- [4] Özkan, Y, Dođan, B, (2013) “İlköğretim 8. Sınıf Öğrencilerinin Okuma Becerilerinin Kestirilmesinde Etkili Olan Deđişkenlerin Belirlenmesi”, International Journal of Social Science, Cilt 6, Sayı 4, 667-680
- [5] Tatlıdil, H, Özel, M, (2005) “Firma Derecelendirme Çalışmaları Konusunda Çok Deđişkenli İstatistiksel Analize Dayalı Karar Destek Sistemlerinin Kullanımı”, Bankacılar Dergisi, Sayı 54.
- [6] Gümüş, T, Çelik, G, Üçer, M (2011) “Kredi Risk Yönetimi Ve Analitik Hiyerarşi Prosesi Yöntemiyle Bir Bireysel Kredi Talep Deđerlendirme Modeli”, XI Üretim Araştırmaları Sempozyumu
- [7] Çelik, M (2010) “Bankaların Finansal Başarısızlıklarının Geleneksel ve Yeni Yöntemlerle Öngörüsü”, Yönetim ve Ekonomi Dergisi, Celal Bayar Üniversitesi.
- [8] Yücel, A (2003) “Bankacılık Sektöründe Risk Ölçümü ve Yönetimi”, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Endüstri Mühendisliđi
- [9] Koyuncugil, A (2008) “Borsa Şirketlerinin Risk Bazlı Gözetimine Yönelik Veri Madenciliđine Dayalı Metodoloji ve Sistem Önerisi”, Sermaye Piyasası Kurulu Araştırma Raporu.
- [10] Koyuncugil, A (2007) “Borsa Şirketlerinin Sektörel Risk Profillerinin Veri Madenciliđiyle Belirlenmesi”, Sermaye Piyasası Kurulu Araştırma Raporu.
- [11] Vassiliou, P. –C.G. (2013) “Fuzzy semi-Markov migration process in credit risk”, Fuzzy Sets and Systems, Sayı 223, 39-58.
- [12] Saha, P, Bose, I, Mahanti, A., (2016), “A knowledge based scheme for risk assessment in loan processing by banks”, Decision Support System, Sayı 84, 78-88.
- [13] Malhotra, R, Malhotra, D.K. (2003), “Evaluating consumer loans using neural networks”, The International Journal of Management Science, Sayı 31, 83-96.

- [14] Zang, Z, Gao, G, Shi, Y, (2014), “*Credit risk evaluation using multi-criteria optimization classifier with kernel, fuzzification and penalty factors*”, European Journal of Operational Research, Sayı 237, 335-348.
- [15] Wu, T, Hsu, M, (2012), “*Credit risk assessment and decision making by a fusion approach*”, Knowledge Based Systems, Sayı 35, 102-110.
- [16] Zhu, X, Li, J, Wu, D, Wang, H, Liang, C, (2013), “*Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-TOPSIS classification approach*”, Knowledge Based Systems, Sayı 52, 258–267.
- [17] Li, S, Shiue, W, Huang, M, (2006), “*The evaluation of consumer loans using support vector machines*”, Experts Systems with Applications, Sayı 30, 772–782.
- [18] Tsai, M, Lin, S, Cheng, C, Lin, Y, (2009), “*The consumer loan default predicting model – An application of DEA–DA and neural network*”, Expert Systems with Applications, Sayı 36, 11682–11690.
- [19] Danenas, P, Garsva, G, (2012), “*Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach*”, Procedia Computer Science, Sayı 9, 1324 – 1333,
- [20] Gorzalczany, M, Rudzinski, F, (2016), “*A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability*”, Applied Soft Computing, Sayı 40, 206–220.
- [21] Dahal, K, Hussain, Z, Hossain, A, “*Loan Risk Analyzer based on Fuzzy Logic*”, Member IEEE School of Informatics, University of Bradford.
- [22] Ferreira, F, Santos, S, Dias, V, (2014), “*An AHP-based approach to credit risk evaluation of mortgage loans*”, International Journal of Strategic Property Management, Sayı 18, 38-55.
- [23] Shahbudin, S, Hussain, A, Hussain, H, Samad, A, Tahir, N, (2010), “*Analysis of PCA Based Feature Vectors for SVM Posture Classification*”, 6th International Colloquium on Signal Processing and Its Applications (CSPA).
- [24] Martens, D, Baesens, B, Gestel, T, Vanthienen, J, Leuven, “*Comprehensible Credit Scoring Models Using Rule Extraction from Support Vector Machines*” K, Dept. of Decision Sciences and Information Management, Naamsestraat 69, B-3000 Leuven, Belgium.
- [25] Huang, C, Chen, M, Wang, C, (2007), “*Credit scoring with a data mining approach based on support vector machines*”, Expert Systems with Applications, Sayı 33, 847-856.
- [26] Ha, V, Nguyen, H, (2016), “*Credit scoring with a feature selection approach based deep learning*”, MATEC Web of Conferences 54.

- [27] Hongjiu, L, Yanrong, H, Wuchong, “*An Application of Support Vector Machine for Evaluating Credit Risk of Bank*”, Proceedings of the 7th International Conference on Innovation & Management.
- [28] Wu, C, Guo, Y, Zhang, X, Xia, H, (2010), “*Study of Personal Credit Risk Assessment Based on Support Vector Machine Ensemble*”, International Journal of Innovative Computing, Information and Control, Volume 6, Number 5.
- [29] Min, J, Lee, Y, (2005), “*Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters*”, Expert Systems with Applications, Sayı 28, 603-614.
- [30] Alpaydın, E., “*Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşım Yöntemleri*”, Bilişim 2000 Eğitim Semineri.
- [31] Rencher, A.C., (1995), “*Methods of Multivariate Analysis*”, Second Edition, A John Wiley and Sons, Inc. Publication.
- [32] Bounsaythip, C., Esa, R., (2001) “*Overview of Data Mining for Customer Behavior Modeling*”, VTT Information Technology Research Report, Version:1, Sayfa. 21.
- [33] Altan, Ş., Atan, M., Kızılkaya, S., (2015), “*Genel Sağlık Durumunu Etkileyen Faktörlerin Chaid Analizi Yöntemi ile İncelenmesi, Odtü Örneği*”, Dergipark, Cilt 10, Sayı 3, 92-106.
- [34] Karagül, K., (2014), “*The Classification Of The Firms Traded In Istanbul Stock Exchange By Using Support Vector Machines*”, Pamukkale University Journal of Engineering Sciences, Volume 20, Number 5, 174-178.
- [35] İstatistik | Sınıf, (Web, Erişim Tarihi, 28.05.2017), <https://goo.gl/hZ4KUR>.
- [36] Atatürk Üniversitesi, (Web, Erişim Tarihi, 30.05.2017), <https://goo.gl/WpNB0>
- [37] Doç. Dr. Ersan Kabalcı Multilayerperceptron ders notları, (Web, Erişim Tarihi, 02.06.2017), <https://ekblc.files.wordpress.com/2013/09/mlp.pdf>

EKLER

EK 1: Kullanılan Matlab Kodları

```
%% Load Data
data=LoadData();
%% Generate Basic FIS
fis=CreateInitialFIS(data,10);
%% Train Using PSO
Options = {'Genetic Algorithm', 'Particle Swarm Optimization'};
[Selection, Ok] = listdlg('PromptString', 'Select training method for ANFIS:', ...
    'SelectionMode', 'single', ...
    'ListString', Options);
pause(0.01);
if Ok==0
    return;
end
switch Selection
    case 1, fis=TrainAnfisUsingGA(fis,data);
    case 2, fis=TrainAnfisUsingPSO(fis,data);
end
%% Results
% Train Data
TrainOutputs=evalfis(data.TrainInputs,fis);
PlotResults(data.TrainTargets,TrainOutputs,'Train Data');
% Test Data
TestOutputs=evalfis(data.TestInputs,fis);
PlotResults(data.TestTargets,TestOutputs,'Test Data');
```

LoadData.m

```
function data=LoadData()

    data=load('eminedata');
    Inputs=data.Inputs';
    Targets=data.Targets';
    Targets=Targets(:,1);

    nSample=size(Inputs,1);

    % Shuffle Data
    S=randperm(nSample);
    Inputs=Inputs(S,:);
    Targets=Targets(S,:);

    % Train Data
    pTrain=0.7;
    nTrain=round(pTrain*nSample);
```

```

TrainInputs=Inputs(1:nTrain,:);
TrainTargets=Targets(1:nTrain,:);

% Test Data
TestInputs=Inputs(nTrain+1:end,:);
TestTargets=Targets(nTrain+1:end,:);

% Export
data.TrainInputs=TrainInputs;
data.TrainTargets=TrainTargets;
data.TestInputs=TestInputs;
data.TestTargets=TestTargets;

end

CreateInitialFis.m
function fis=CreateInitialFIS(data,nCluster)

if ~exist('nCluster','var')
    nCluster='auto';
end

x=data.TrainInputs;
t=data.TrainTargets;

fcm_U=5;
fcm_MaxIter=100;
fcm_MinImp=1e-5;
fcm_Display=true;
fcm_options=[fcm_U fcm_MaxIter fcm_MinImp fcm_Display];
fis=genfis3(x,t,'mamdani',nCluster,fcm_options);

end

```

SetFisParams.m

```

function fis=SetFISParams(fis,p)

nInput=numel(fis.input);
for i=1:nInput
    nMF=numel(fis.input(i).mf);
    for j=1:nMF
        k=numel(fis.input(i).mf(j).params);
        fis.input(i).mf(j).params=p(1:k);
        p(1:k)=[];
    end
end
end

```

```

nOutput=numel(fis.output);
for i=1:nOutput
    nMF=numel(fis.output(i).mf);
    for j=1:nMF
        k=numel(fis.output(i).mf(j).params);
        fis.output(i).mf(j).params=p(1:k);
        p(1:k)=[];
    end
end
end
end

```

GetFisParams.m

```

function p=GetFISParams(fis)

p=[];

nInput=numel(fis.input);
for i=1:nInput
    nMF=numel(fis.input(i).mf);
    for j=1:nMF
        p=[p fis.input(i).mf(j).params];
    end
end

nOutput=numel(fis.output);
for i=1:nOutput
    nMF=numel(fis.output(i).mf);
    for j=1:nMF
        p=[p fis.output(i).mf(j).params];
    end
end

end
end

```

PlotResults.m

```

function PlotResults(Targets, Outputs, Name)

figure;

Errors=Targets-Outputs;

MSE=mean(Errors.^2);
RMSE=sqrt(MSE);

error_mean=mean(Errors);
error_std=std(Errors);

```

```

subplot(2,2,[1 2]);
plot(Targets,'k');
hold on;
plot(Outputs,'r');
legend('Target','Output');
title(Name);
xlabel('Sample Index');
grid on;

subplot(2,2,3);
plot(Errors);
legend('Error');
title(['MSE = ' num2str(MSE) ', RMSE = ' num2str(RMSE)]);
grid on;

subplot(2,2,4);
histfit(Errors, 50);
title(['Error Mean = ' num2str(error_mean) ', Error St.D. = ' num2str(error_std)]);

end

```

TrainAnfisUsingGA.m

```

function bestfis=TrainAnfisUsingGA(fis,data)

%% Problem Definition

p0=GetFISParams(fis);

Problem.CostFunction=@\(x\) TrainFISCost\(x,fis,data\);

Problem.nVar=numel(p0);

% Problem.VarMin=-25;
% Problem.VarMax=25;
Problem.VarMin=1;
Problem.VarMax=80;

%% GA Params
Params.MaxIt=100;
Params.nPop=50;

%% Run GA
results=RunGA(Problem,Params);

%% Get Results

p=results.BestSol.Position.*p0;
bestfis=SetFISParams(fis,p);

```

```

end

function results=RunGA(Problem,Params)

    disp('Starting GA ...');

    %% Problem Definition

    CostFunction=Problem.CostFunction;    % Cost Function

    nVar=Problem.nVar;    % Number of Decision Variables

    VarSize=[1 nVar];    % Size of Decision Variables Matrix

    VarMin=Problem.VarMin;    % Lower Bound of Variables
    VarMax=Problem.VarMax;    % Upper Bound of Variables

    %% GA Parameters

    MaxIt=Params.MaxIt;    % Maximum Number of Iterations

    nPop=Params.nPop;    % Population Size

    pc=0.4;    % Crossover Percentage
    nc=2*round(pc*nPop/2); % Number of Offsprings (Parnets)

    pm=0.7;    % Mutation Percentage
    nm=round(pm*nPop);    % Number of Mutants

    gamma=0.7;

    mu=0.15;    % Mutation Rate

    beta=8;    % Selection Pressure

    %% Initialization

    empty_individual.Position=[];
    empty_individual.Cost=[];

    pop= repmat(empty_individual,nPop,1);

    for i=1:nPop

        % Initialize Position
        if i>1
            pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
        else
            pop(i).Position=ones(VarSize);
        end
    end
end

```

```

end

% Evaluation
pop(i).Cost=CostFunction(pop(i).Position);

end

% Sort Population
Costs=[pop.Cost];
[Costs, SortOrder]=sort(Costs);
pop=pop(SortOrder);

% Store Best Solution
BestSol=pop(1);

% Array to Hold Best Cost Values
BestCost=zeros(MaxIt,1);

% Store Cost
WorstCost=pop(end).Cost;

%% Main Loop

for it=1:MaxIt

    P=exp(-beta*Costs/WorstCost);
    P=P/sum(P);

    % Crossover
    popc= repmat(empty_individual,nc/2,2);
    for k=1:nc/2

        % Select Parents Indices
        i1=RouletteWheelSelection(P);
        i2=RouletteWheelSelection(P);

        % Select Parents
        p1=pop(i1);
        p2=pop(i2);

        % Apply Crossover
        [popc(k,1).Position, popc(k,2).Position]=...
            Crossover(p1.Position,p2.Position,gamma,VarMin,VarMax);

        % Evaluate Offsprings
        popc(k,1).Cost=CostFunction(popc(k,1).Position);
        popc(k,2).Cost=CostFunction(popc(k,2).Position);

    end
end

```

```

popc=popc(:);

% Mutation
popm= repmat(empty_individual,nm,1);
for k=1:nm

    % Select Parent
    i=randi([1 nPop]);
    p=pop(i);

    % Apply Mutation
    popm(k).Position=Mutate(p.Position,mu,VarMin,VarMax);

    % Evaluate Mutant
    popm(k).Cost=CostFunction(popm(k).Position);

end

% Create Merged Population
pop=[pop
     popc
     popm]; %#ok

% Sort Population
Costs=[pop.Cost];
[Costs, SortOrder]=sort(Costs);
pop=pop(SortOrder);

% Update Worst Cost
WorstCost=max(WorstCost,pop(end).Cost);

% Truncation
pop=pop(1:nPop);
Costs=Costs(1:nPop);

% Store Best Solution Ever Found
BestSol=pop(1);

% Store Best Cost Ever Found
BestCost(it)=BestSol.Cost;

% Show Iteration Information
disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);

end

disp('End of GA. ');
disp(' ');

```

```

%% Results

results.BestSol=BestSol;
results.BestCost=BestCost;

end

function [y1, y2]=Crossover(x1,x2,gamma,VarMin,VarMax)

    alpha=unifrnd(-gamma,1+gamma,size(x1));

    y1=alpha.*x1+(1-alpha).*x2;
    y2=alpha.*x2+(1-alpha).*x1;

    y1=max(y1,VarMin);
    y1=min(y1,VarMax);

    y2=max(y2,VarMin);
    y2=min(y2,VarMax);

end

function y=Mutate(x,mu,VarMin,VarMax)

    nVar=numel(x);

    nmu=ceil(mu*nVar);

    j=randsample(nVar,nmu)';

    sigma=0.1*(VarMax-VarMin);

    y=x;
    y(j)=x(j)+sigma*randn(size(j));

    y=max(y,VarMin);
    y=min(y,VarMax);

end

TrainFisCost.m

function [z, out]=TrainFISCost(x,fis,data)

    MinAbs=1e-5;
    if any(abs(x)<MinAbs)
        S=(abs(x)<MinAbs);
        x(S)=MinAbs.*sign(x(S));
    end

```

```

p0=GetFISParams(fis);

p=x.*p0;

fis=SetFISParams(fis,p);

x=data.TrainInputs;
t=data.TrainTargets;
y=evalfis(x,fis);

e=t-y;

MSE=mean(e(:).^2);
RMSE=sqrt(MSE);

z=RMSE;

out.fis=fis;
%out.y=y;
%out.e=e;
out.MSE=MSE;
out.RMSE=RMSE;

end

```

EK 2: Yapay Sinir Ağları Analiz Sonucu Tahmin Yüzdeleri

Örneklem	Gerçek Değer	Tahmin Edilen Değer	Tahmin Hatası
1	1:iyi	1:iyi	0.96
2	1:iyi	1:iyi	1
3	1:iyi	1:iyi	1
4	1:iyi	1:iyi	1
5	1:iyi	1:iyi	0.863
6	2:kötü	2:kötü	0.997
7	2:kötü	2:kötü	1
8	2:kötü	2:kötü	0.946
9	1:iyi	1:iyi	1
10	2:kötü	1:iyi	+ 0.896
11	2:kötü	1:iyi	+ 0.897
12	1:iyi	1:iyi	1
13	1:iyi	1:iyi	0.62
14	2:kötü	1:iyi	1
15	2:kötü	2:kötü	0.993
16	1:iyi	1:iyi	0.959

17	1:iyi	1:iyi	0.771
18	1:iyi	1:iyi	0.986
19	1:iyi	1:iyi	1
20	1:iyi	1:iyi	1
21	1:iyi	1:iyi	1
22	1:iyi	1:iyi	0.99
23	1:iyi	1:iyi	1
24	1:iyi	2:kötü	+ 0.944
25	2:kötü	2:kötü	0.989
26	1:iyi	1:iyi	0.994
27	1:iyi	1:iyi	0.606
28	1:iyi	2:kötü	+ 0.913
29	2:kötü	1:iyi	1
30	1:iyi	2:kötü	+ 0.999
31	2:kötü	1:iyi	1
32	1:iyi	1:iyi	1
33	1:iyi	2:kötü	+ 0.683
34	1:iyi	1:iyi	0.687
35	2:kötü	2:kötü	1
36	1:iyi	1:iyi	0.94
37	2:kötü	1:iyi	+ 0.999
38	1:iyi	1:iyi	1
39	1:iyi	1:iyi	1
40	1:iyi	1:iyi	1
41	2:kötü	1:iyi	+ 0.997
42	2:kötü	1:iyi	+ 0.91
43	1:iyi	1:iyi	0.605
44	1:iyi	1:iyi	1
45	1:iyi	1:iyi	1
46	1:iyi	1:iyi	1
47	1:iyi	1:iyi	1
48	1:iyi	1:iyi	1
49	1:iyi	1:iyi	0.918
50	1:iyi	2:kötü	+ 0.85
51	2:kötü	2:kötü	0.689
52	2:kötü	2:kötü	0.995
53	1:iyi	1:iyi	1
54	2:kötü	1:iyi	1
55	1:iyi	1:iyi	0.892
56	1:iyi	1:iyi	1
57	1:iyi	1:iyi	0.853

58	2:kötü	1:iyi	+ 0.982
59	1:iyi	1:iyi	1
60	2:kötü	1:iyi	1
61	1:iyi	1:iyi	0.997
62	1:iyi	1:iyi	0.81
63	1:iyi	1:iyi	1
64	1:iyi	1:iyi	0.815
65	1:iyi	1:iyi	1
66	2:kötü	2:kötü	0.679
67	1:iyi	2:kötü	+ 0.981
68	1:iyi	1:iyi	1
69	1:iyi	1:iyi	1
70	1:iyi	2:kötü	+ 0.999
71	1:iyi	1:iyi	1
72	1:iyi	2:kötü	1
73	1:iyi	1:iyi	1
74	2:kötü	2:kötü	0.997
75	1:iyi	1:iyi	1
76	1:iyi	1:iyi	1
77	1:iyi	1:iyi	0.985
78	2:kötü	2:kötü	0.503
79	1:iyi	1:iyi	0.999
80	2:kötü	1:iyi	+ 0.842
81	1:iyi	1:iyi	1
82	1:iyi	1:iyi	1
83	1:iyi	1:iyi	1
84	1:iyi	2:kötü	+ 0.992
85	1:iyi	1:iyi	1
86	1:iyi	2:kötü	+ 0.687
87	1:iyi	1:iyi	0.982
88	1:iyi	2:kötü	1
89	1:iyi	1:iyi	1
90	1:iyi	1:iyi	1
91	1:iyi	1:iyi	1
92	1:iyi	2:kötü	+ 0.924
93	1:iyi	1:iyi	1
94	1:iyi	1:iyi	0.951
95	1:iyi	1:iyi	1
96	1:iyi	2:kötü	+ 0.826
97	1:iyi	1:iyi	1
98	2:kötü	2:kötü	0.842

99	1:iyi	1:iyi	1
100	2:kötü	1:iyi	+ 0.987
101	1:iyi	2:kötü	+ 0.916
102	1:iyi	1:iyi	0.751
103	1:iyi	1:iyi	1
104	1:iyi	2:kötü	+ 0.941
105	1:iyi	2:kötü	1
106	1:iyi	1:iyi	0.897
107	1:iyi	1:iyi	1
108	1:iyi	2:kötü	+ 0.983
109	1:iyi	1:iyi	1
110	1:iyi	1:iyi	1
111	1:iyi	1:iyi	0.729
112	2:kötü	2:kötü	1
113	1:iyi	1:iyi	1
114	1:iyi	1:iyi	0.998
115	1:iyi	1:iyi	0.974
116	2:kötü	2:kötü	0.566
117	1:iyi	1:iyi	0.762
118	1:iyi	1:iyi	1
119	1:iyi	2:kötü	+ 0.989
120	1:iyi	1:iyi	0.999
121	1:iyi	1:iyi	0.993
122	1:iyi	1:iyi	0.584
123	1:iyi	1:iyi	0.956
124	2:kötü	2:kötü	1
125	1:iyi	1:iyi	1
126	1:iyi	1:iyi	1
127	1:iyi	1:iyi	1
128	1:iyi	1:iyi	1
129	2:kötü	1:iyi	+ 0.824
130	1:iyi	1:iyi	1
131	1:iyi	1:iyi	1
132	2:kötü	2:kötü	0.997
133	1:iyi	1:iyi	1
134	2:kötü	1:iyi	1
135	1:iyi	1:iyi	0.957
136	1:iyi	1:iyi	1
137	1:iyi	1:iyi	0.96
138	1:iyi	2:kötü	+ 0.908
139	1:iyi	1:iyi	1

140	1:iyi	1:iyi	0.998
141	1:iyi	1:iyi	1
142	1:iyi	2:kötü	1
143	1:iyi	1:iyi	0.98
144	1:iyi	1:iyi	1
145	2:kötü	2:kötü	0.988
146	1:iyi	2:kötü	+ 0.918
147	1:iyi	1:iyi	0.998
148	1:iyi	1:iyi	0.996
149	1:iyi	1:iyi	0.838
150	1:iyi	2:kötü	1
151	2:kötü	2:kötü	0.994
152	2:kötü	2:kötü	1
153	1:iyi	1:iyi	1
154	1:iyi	1:iyi	1
155	1:iyi	1:iyi	0.754
156	1:iyi	1:iyi	1
157	1:iyi	1:iyi	0.612
158	2:kötü	1:iyi	+ 0.999
159	1:iyi	1:iyi	0.996
160	2:kötü	2:kötü	0.675
161	1:iyi	1:iyi	0.992
162	2:kötü	2:kötü	0.666
163	2:kötü	2:kötü	1
164	1:iyi	2:kötü	1
165	1:iyi	1:iyi	1
166	2:kötü	1:iyi	+ 0.611
167	2:kötü	2:kötü	0.996
168	1:iyi	1:iyi	0.931
169	2:kötü	1:iyi	+ 0.75
170	1:iyi	1:iyi	1
171	2:kötü	2:kötü	1
172	1:iyi	1:iyi	1
173	1:iyi	1:iyi	1
174	2:kötü	2:kötü	0.903
175	1:iyi	1:iyi	1
176	2:kötü	2:kötü	0.985
177	1:iyi	1:iyi	0.508
178	2:kötü	2:kötü	1
179	1:iyi	1:iyi	0.995
180	1:iyi	1:iyi	1

181	1:iyi	2:kötü	+ 0.942
182	2:kötü	1:iyi	1
183	2:kötü	2:kötü	1
184	1:iyi	2:kötü	1
185	1:iyi	2:kötü	+ 0.948
186	1:iyi	1:iyi	1
187	2:kötü	2:kötü	0.822
188	2:kötü	1:iyi	1
189	1:iyi	1:iyi	0.943
190	1:iyi	1:iyi	1
191	1:iyi	1:iyi	1
192	1:iyi	1:iyi	1
193	2:kötü	1:iyi	+ 0.994
194	1:iyi	1:iyi	1
195	1:iyi	1:iyi	0.612
196	2:kötü	2:kötü	0.889
197	1:iyi	1:iyi	0.999
198	1:iyi	2:kötü	+ 0.997
199	1:iyi	1:iyi	0.994
200	1:iyi	1:iyi	0.999

ÖZGEÇMİŞ

9 Nisan 1988 tarihinde Diyarbakır'da doğdu. 2012 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar ve Kontrol Öğretmenliği bölümünde lisans eğitimini tamamladı. Lisans Eğitimi sonrası Marmara Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği bölümünde yüksek lisans eğitimine başladı. Şuan da İş Bankasının iştirak şirketlerinden biri olan Softtech firmasında Yazılım Analisti olarak çalışmaktadır.