



T.C.

ALTINBAŞ ÜNİVERSİTESİ

Lisansüstü Eğitim Enstitüsü

Elektrik ve Bilgisayar Mühendisliği

**MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE
ÇEVİRİMİÇİ KREDİ KARTI İŞLEMLERİNDE
FRAUD ANALİZİ TAHMİNİ**

Yasin DİKBİYİK

Yüksek Lisans Tezi

Danışman

Prof. Dr. Osman Nuri UÇAN

İstanbul, 2024

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ÇEVİRİMİÇİ KREDİ KARTI İŞLEMLERİNDE FRAUD ANALİZİ TAHMİNİ

Yasin DİKBİYİK

Elektrik ve Bilgisayar Mühendisliği

Yüksek Lisans Tezi

ALTINBAŞ ÜNİVERSİTESİ

2024

“Yasin DİKBİYİK” tarafından hazırlanmış ve 26.04.2024 tarihinde sunulmuş “MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ÇEVİRİMİÇİ KREDİ KARTI İŞLEMLERİNDE FRAUD ANALİZİ TAHMİNİ” başlıklı tez Elektrik - Elektronik Mühendisliği Ana Bilim Dalında Yüksek Lisans Tezi olarak **oy birliği** ile kabul edilmiştir.

Prof. Dr. Osman Nuri UÇAN
Danışman

Tez Savunma Sınavı Jüri Üyeleri:

Prof. Dr. Osman Nuri UÇAN Elektrik Elektronik Mühendisliği
Bölümü,
Altınbaş Üniversitesi

Doç. Dr. Sefer KURNAZ Bilgisayar Mühendisliği
Bölümü,
Altınbaş Üniversitesi

Doç. Dr. Adil Deniz DURU Antrenörlük Eğitimi
Bölümü,
Marmara Üniversitesi

Bu tezin Yüksek Lisans tezi olarak bütün şartları sağladığımı beyan ederim.

Bu tezdeki tüm bilgilerin akademik kurallara ve etik davranışlara uygun olarak edinildiğini ve sunulduğunu beyan ederim. Ayrıca, bu kuralların ve davranışların gerektirdiği şekilde, bu çalışmada, orijinal olmayan tüm materyalleri ve sonuçları tamamen alıntı yaptığımı ve referans gösterdiğimi de beyan ederim.

Yasin DİKBİYİK

İmzası



ÖZET

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ÇEVİRİMİÇİ KREDİ KARTI İŞLEMLERİNDE FRAUD ANALİZİ TAHMİNİ

DİKBIYIK, Yasin

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı, Altınbaş Üniversitesi,

Danışman: Prof. Dr. Osman Nuri UÇAN

Tarih: 04/2024

Sayfa: 82

Kredi kartları, dünya genelinde yaygın kullanımı ve sağlam altyapısı sayesinde hızla insanların günlük hayatlarına entegre olmuş ve güvenle kullanılan ödeme araçlarından biri haline gelmiştir. Ancak, kredi kartı sayılarının artması ve işlem hacminin hızla büyümesi, dolandırıcıları cezbetmiş ve haksız kazanç elde etme amacıyla çeşitli dolandırıcılık yöntemlerini ortaya çıkarmıştır. Günümüzde kredi kartı bilgilerine ulaşmanın kolaylaşması, kredi kartı dolandırıcılarının faaliyetlerini kolaylaştırmaktadır. Gelişen teknoloji ile hesap hareketleri zaman içinde analiz edilebilmekte ve kötü niyetli verilerin kullanımı izlenebilmektedir.

Bu çalışmada, Kaggle veritabanından elde edilen Kredi Kartı Dolandırıcılık Teşhis veri seti kullanılarak bu çalışma, kredi kartı dolandırıcılığı tespiti için topluluk tabanlı XGBoost modeli ile diğer geleneksel makine öğrenmesi modellerini karşılaştırarak önemli bulgular ortaya koymaktadır. XGBoost, Random Forest ve CatBoost modellerinin, kesinlik, geri çağırma gibi performans ölçütleri üzerinde daha iyi performans sergilediği belirtilmektedir. Bu sonuçlar, finansal kurumların günlük operasyonlarında karşılaştıkları dolandırıcılık risklerini azaltma potansiyeline işaret etmektedir.

Çalışmanın dikkate değer bir noktası, XGBoost, Random Forest ve CatBoost'un dengesiz veri setleriyle başa çıkma yeteneğinin vurgulanmasıdır.

Geleneksel modellerin zayıf performans gösterdiği bu durumlarda, XGBoost, RandomForest ve CatBoost'un %99'a kadar tahmin doğruluğu sağladığı ve diğer modellere göre daha iyi bir performans sunduğu gözlemlenmektedir.

Anahtar Kelimeler: Çok Katmanlı Sinir Ağları, Veri Madenciliği, Naive Bayes Yönetmi, Sahtekarlık Tespiti, Performans Ölçütleri.



ABSTRACT

FRAUD ANALYSIS PREDICTION IN ONLINE CREDIT CARD TRANSACTIONS USING MACHINE LEARNING METHODS

DİKBIYIK, Yasin

MSc, Department of Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: 04/2024

Pages: 82

Credit cards have rapidly integrated into people's daily lives and become one of the securely used payment methods worldwide, thanks to their widespread usage and robust infrastructure. However, the increasing number of credit cards and the rapid growth in transaction volume have attracted fraudsters, leading to the emergence of various fraudulent methods aimed at gaining unjust profits. The ease of accessing credit card information today facilitates the activities of credit card fraudsters. With advancing technology, account transactions can be analyzed over time, enabling the tracking of the use of malicious data.

In this study, utilizing the Credit Card Fraud Detection dataset obtained from Kaggle, significant findings are presented by comparing a community-based XGBoost model with other traditional machine learning models for detecting credit card fraud. It is noted that XGBoost, Random Forest, and CatBoost models perform better on performance metrics such as precision and recall. These results indicate the potential of reducing fraud risks encountered in the daily operations of financial institutions.

A noteworthy point of the study is the emphasis on the ability of XGBoost, Random Forest, and CatBoost to handle imbalanced datasets.

In cases where traditional models exhibit poor performance, XGBoost, RandomForest, and CatBoost are reported to achieve up to 99% prediction accuracy and provide better performance compared to other models.

Keywords: Multilayer Neural Networks, Data Mining, Naive Bayes Classifier, Fraud Detection, Performance Metrics.



İÇİNDEKİLER

Sayfa

ÖZET	v
ABSTRACT	xii
ŞEKİL LİSTESİ	xii
ÇİZELGE LİSTESİ	xiv
KISALTMALAR.....	xv
SEMBOL LİSTESİ	xvi
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI.....	2
3. BANKACILIK SİSTEMİNDE DOLANDIRICILIK VE SAHTECİLİK YÖNTEMİ	5
4. FRAUD SİSTEMİ NEDİR?.....	9
5. YAPAY ZEKA.....	12
6. MAKİNE ÖĞRENMESİ ALGORİTMALARI.....	16
6.1 MAKİNE ÖĞRENMESİ SÜREÇLERİ	17
6.1.1 Veri Toplama	17
6.1.2 Veri Ön İşleme	18
6.1.3 Özellik Seçimi.....	18
6.1.4 Model Seçimi	18
6.1.5 Model Eğitimi	18
6.1.6 Model Doğrulama ve Değerlendirme.....	18
6.1.7 Model Ayarlaması ve İyileştirme.....	19
6.1.8 Modelin Dağıtılması ve Uygulaması	19
6.1.9 Modelin Sürekli İzlenmesi	19
6.2 VERİ GEREKSİNİMLERİ VE VERİ İŞLEME.....	19
6.3 MAKİNE ÖĞRENMESİ TAHMİN KULLANIM ÖRNEKLERİ	20
6.3.1 Yeni Ürün Tanıtımı	20

6.3.2 Pazarlama Kampanyaları	20
6.3.3 Tahmine Dayalı Satış Analitiği.....	21
6.3.4 Talep Algılama.....	21
6.4 MAKİNE ÖĞRENİMİ ALGORİTMA TÜRLERİ.....	22
6.4.1 Gözetimli Öğrenme (Supervised Learning)	22
6.4.2 Denetimsiz Öğrenme (Unsupervised Learning)	23
6.4.3 Pekiştirmeli Öğrenme (Reinforcement Learning)	24
6.4.4 Yarı Gözetimli Öğrenme (Semi- Supervised Learning)	24
6.4.5 Takviyeli Öğrenme (Transfer Learning).....	24
6.5 MAKİNE ÖĞRENİMİ MODELLERİ	25
6.5.1 Doğrusal Regresyon	25
6.5.2 Lojistik Regresyon	26
6.5.3 Karar Ağacı.....	26
6.5.4 K En Yakın Komşu Algoritması.....	27
6.5.5 Destek Vektör Makinesi Algoritması	28
6.5.6 Gradient Boosting Algoritması	29
6.5.7 XGBoost Algoritması	30
6.5.7.1 Gradient boosting'in temelleri.....	30
6.5.7.2 Xgboost ve diğer gradient boosting uygulamalarının karşılaştırılması.....	31
6.5.7.3 Xgboost algoritması	31
6.5.7.4 Xgboost'un uygulamaları	32
6.5.8 Random Forest Algoritması	33
6.5.8.1 Random forest nedir?	34
6.5.8.2 Random forest nasıl çalışır?	35
6.5.8.3 Sınıflandırma ve regresyonda rastgele orman	35
6.5.8.4 Rastgele orman modelleri vs. karar ağaçları	36
6.5.8.5 Random forest hiperparametreleri.....	36
6.5.8.6 Öngörü gücünü artırma	36

6.5.8.7 Random forest modelinin hızını artırma	37
6.5.8.8 Rastgele orman modelinin avantaj ve dezavantajları	37
7. YAPAY SİNİR AĞLARI	39
7.1 BİYOLOJİK SİNİR AĞI	39
7.2 YAPAY SİNİR AĞI ÇALIŞMA ŞEKLİ	39
7.3 YAPAY SİNİR AĞI YAPISI	41
7.4 YAPAY SİNİR AĞLARININ SINIFLANDIRILMASI	44
7.4.1 Yapılarına Göre Yapay Sinir Ağları	44
7.4.2 Öğrenme Zamanına Göre Yapay Sinir Ağları	44
7.4.3 Öğrenme Zamanına Göre Yapay Sinir Ağları	45
7.5 YAPAY SİNİR AĞLARININ EĞİTİLMESİ VE ÖĞRENMESİ	47
8. UYGULAMA VE SONUÇLAR	51
8.1 DOLANDIRICILIK İŞLEMLERİ BELİRLİ BİR ZAMAN DİLİMİNDE DAHA SIK MI GERÇEKLEŞİYOR?	52
REFERANSLAR	63

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 6.1: Denetimli Öğrenme [39].	22
Şekil 6.2: Denetimsiz Öğrenme [41].	23
Şekil 6.3: Pekiştirmeli Öğrenme [45].	24
Şekil 6.4: Doğrusal Regresyon [49].	25
Şekil 6.5: Lojistik Regresyon [51].	26
Şekil 6.6: Karar Ağacı [53].	27
Şekil 6.7: K En Yakın Komşu Algoritması [55].	27
Şekil 6.8: Destek Vektör Makinesi Algoritması [57].	28
Şekil 6.9: Karar Ağacı Algoritmalarının Zaman İçindeki Gelişimi [59].	29
Şekil 6.10: Random Forest Algoritmasının Kabaca Kodu [62].	33
Şekil 6.11: Random Forest Algoritması [64].	35
Şekil 7.1: Biyolojik Sinir Hücresi [65].	39
Şekil 7.2: Yapay Sinir Ağı Modelinin Yapay Sinir Ağı Tasarımının Matematiksel Gösterimi [67].	41
Şekil 8.1: Kütüphane Ve Sql Connection Bilgileri.	53
Şekil 8.2: SQL Connection Bilgileri.	53
Şekil 8.3: Veri Kümesi Gösterimi.	54
Şekil 8.4: Veri Kümesine Ait Değerlerinin İstatistiksel Gösterimi.	54
Şekil 8.5: Veri Kümesine Ait Değişkenlerin Türlerinin Gösterimi.	54

Şekil 8.6: Veri Kümesine Ait Değişkenlerin Format Düzenlemesi.....	55
Şekil 8.7: Veri Kümesine Ait Değişken Bilgisi Gösterimi.....	55
Şekil 8.8: Veri Setinin Düzenlenmiş Gösterimi.	55
Şekil 8.9: Veri Kümesine Ait Değişkenlerin İşlem Bazlı Gösterimi.....	56
Şekil 8.10: Veri Kümesine Ait Değişkenlerin İşlem Sınıfı Dağılımı.....	56
Şekil 8.11: Veri Kümesine Ait İşlemlerin Sınıf Bazında Toplamı.....	56
Şekil 8.12: Veri Kümesine Ait Sınıflandırılmış İşlemlerin Gösterimi.	57
Şekil 8.13: Veri Kümesine Ait Toplam Değişkenin Değer Bazında Gösterimi.....	57
Şekil 8.14: Veri Kümesine Ait Zaman Değişkenin Değer Bazında Gösterimi.	57
Şekil 8.15: Veri Kümesine Ait İşlem Bazlı Transaction Gösterimi.	58
Şekil 8.16: Veri Kümesine Ait İşlem Bazlı Transaction Gösterimi.	58
Şekil 8.17: Pyhton IDE'si Üzerinde Veri Kümesinin Matrix Formülü Gösterimi.....	58
Şekil 8.18: Pyhton IDE'si Üzerinde Veri Kümesine Ait Matrix Gösterimi.....	59
Şekil 8.19: Pyhton IDE'si Üzerinde Veri Kümesinin Ölçeklendirilmesi.	60
Şekil 8.20: Veri Kümesine Ait Modelleme Sonucu.	61

ÇİZELGE LİSTESİ

Sayfa

Çizelge 7.1: Dirimsel Sinir Sisteminde Yapay Sinir Ağları Benzetim Karşılıkları.....	42
Çizelge 7.2: Yapay Sinir Ağlarında Kullanılan Eğitim Metotları.	47
Çizelge 8.1: Model Sınıflarının Karşılaştırılması	60



KISALTMALAR

ATM	:	Automated Teller Machine / Bankamatik
DN	:	Doğru Negatif
DP	:	Doğru Pozitif
EDA	:	Exploratory Data Analysis / Keşifsel Veri Analizi
FRAUD	:	Dolandırıcılık / Sahtecilik
KNN	:	K Nearest Neighbors / K-En Yakın Komşu Algoritması
KNIME	:	Konstanz Information Miner / Konstanz Bilgi Madencisinin
MAPE	:	Mean Absolute Percentage Error / Ortalama Mutlak Yüzde Hatası
NB	:	Navie Bayes
RF	:	Random Forest
YN	:	Yanlış Negatif
YP	:	Yanlış Pozitif
YSA	:	Yapay Sinir Ağları
NB	:	Navie Bayes
YN	:	Yanlış Negatif

SEMBOL LİSTESİ

- μ : Elektronun Mobilitesi
 λ : Dalgaboyu
 ω : Açısal Frekans



1. GİRİŞ

Kredi kartları, bankaların sunduğu bir hizmet olarak kullanıcılara sağlanan ve alışveriş noktalarında ödeme yapmak için kullanılan özel kartlardır. Ayrıca, ATM'lerden nakit para çekme imkanı da sunarlar. Kredi kartları, harcamaların taksitlendirilmesi konusunda kullanıcılara kolaylık sağlayarak, tek seferde ödeme yapmak yerine belirli bir süre içinde ödemeyi taksitlere bölmelerine olanak tanır. Dünya genelinde yaygın olarak kullanılan kredi kartları, güçlü altyapısı ve kullanım kolaylığıyla hızla popüler bir ödeme aracı haline gelmiştir.

Ancak, kredi kartlarının yaygın kullanımı beraberinde bazı sorunları da getirmiştir. Kredi kartı bilgilerinin kötü niyetli kişilerin eline geçmesi ve dolandırıcılık amaçlı kullanılması, bu sorunların başında gelmektedir. Kredi kartı dolandırıcılığı, mevcut kart bilgilerinin kopyalanması, e-ticaret sitelerinden çalınarak sahte işlemlerin gerçekleştirilmesi veya kart üzerinden para transferi gibi yöntemlerle gerçekleşebilir. Bu tür dolandırıcılık vakaları, dünya genelinde önemli finansal kayıplara neden olmaktadır.

Bu nedenle, bu çalışmada kredi kartı işlemlerinden elde edilen veriler analiz edilerek, kredi kartı dolandırıcılığını önlemeye yönelik çeşitli stratejiler geliştirilmektedir. Kredi kartı hareketleri üzerinde yapılan detaylı analizler, olası dolandırıcılık durumlarını tespit etme ve önleme amacını taşımaktadır. Bu çabalar, kredi kartı kullanıcılarının güvenliğini artırmayı ve dolandırıcılık riskini minimize etmeyi hedeflemektedir.

2. LİTERATÜR TARAMASI

Sait Gültekin, 2017 yılında hazırlamış olduğu “Yapay Sinir Ağı Ve Doğrusal Regresyon Yöntemleri İle Fiyat Tahmini” Doğrusal Regresyon ve Yapay Sinir Ağı algoritmalarının kullanmıştır. Bu algoritmalar 20 farklı marka için ayrı ayrı değerlendirme yapmak amacıyla kullanılmış. Değerlendirmelerin ortalaması alındığında elde edilen %93 doğruluk oranı, muhtemelen bu iki algoritmanın tahminlerinin gerçek değerlere ne kadar yakın olduğunu göstermektedir. Doğruluk oranı yüksek olduğu için, kullanılan algoritmaların genel olarak iyi bir tahmin performansı sergilediği söylenebilir. Bu sonuçlar, belirli bir ürün veya hizmetin fiyatının tahmin edilmesinde Doğrusal Regresyon ve Yapay Sinir Ağı gibi yöntemlerin etkili olduğunu göstermektedir [1].

Kazım Soylu, 2018 yılında hazırlanmış olduğu “Kredi Kartı Sahte İşlem Tespiti” başlıklı tezinde, hazırlamış olduğu kredi kartı işlem veri kümesinde, %70'i eğitim (train) kümesi olarak kullanılarak model eğitilmiş ve geriye kalan %30'u test kümesi olarak kullanılarak modelin başarısı değerlendirilmiştir. Eğitilen derin öğrenme modelinin sahte işlem tespitindeki başarısı %78,7 olarak belirlenmiştir. Bu sonuç, modelin veri kümesindeki sahte işlemleri doğru tespit etme yeteneğini yansıtmaktadır. Aynı veri kümesi üzerinde Rastgele Orman modeli de kullanılmıştır. Bu modelle elde edilen sonuca göre, doğruluk hata oranı (accuracy error rate) 0,00008 olarak belirlenmiştir. Bu sonuç, Rastgele Orman modelinin sahte işlem tespitindeki başarısını ifade etmektedir. İki farklı modelin kullanıldığı bu çalışma, derin öğrenme modeli ile Rastgele Orman modelinin sahte işlem tespitindeki başarılarını karşılaştırmayı amaçlamış olabilir. Derin öğrenme modeli %78,7'lik bir başarı elde ederken, Rastgele Orman modeli ise çok düşük bir doğruluk hata oranı olan 0,00008 elde etmiştir [2].

Ebru Pekel Özmen ve Tuncay Özcan tarafından 2019 yılında yayınlanan makalede, finans sektöründe dolandırıcılık tespiti konusunda Regresyon Ağacı Yöntemi ve Melez Sınıflandırma kullanılarak bir analiz gerçekleştirilmiştir. Bu çalışmanın sonuçlarına göre, MATLAB platformunun R2014a sürümü kullanılarak yapılan analizde CART ve GA-CART algoritmaları incelenmiştir. CART (Classification and Regression Trees) algoritması %64,28 doğruluk oranı vermiştir.

Bu algoritma, veri setini kullanarak ağaç yapısı oluşturur ve bu yapıyı temel alarak sınıflandırma yapar. Ancak, %64,28 doğruluk oranı düşük bir sonuç olarak değerlendirilebilir.

GA-CART (Genetic Algorithm-CART) algoritması ise %87,95 doğruluk oranı elde etmiştir. Bu algoritma, genetik algoritma prensiplerini kullanarak ağaç yapısını daha iyi optimize eder ve bu sayede daha yüksek doğruluk oranlarına ulaşmayı amaçlar.

Sonuç olarak, GA-CART algoritmasının CART algoritmasına göre daha yüksek bir doğruluk oranı sağladığı görülmektedir. Melez yaklaşımların kullanılmasıyla, finans sektöründe dolandırıcılık tespiti gibi kritik alanlarda daha etkili sonuçlar elde edilebileceği göstermiştir [3].

Safa, M. U. ve R. Ganga yazdıkları bu makalelerinde, üç makine öğrenimi tekniğinin (Naive Bayes, k-en yakın komşu ve lojistik regresyon) yüksek oranda çarpık kredi kartı dolandırıcılığı verileri üzerindeki performansını incelemektedir. Her bir dolandırıcılık işlemi bu modeller kullanılarak değerlendirilmekte ve en iyi yöntem değerlendirme yoluyla belirlenmektedir. Çalışma hem ham hem de önceden işlenmiş verileri kapsamakta ve Python'da uygulanmaktadır.

Performans değerlendirmesi doğruluk, zaman süresi ve dengeli sınıflandırma oranına dayanmaktadır. Sonuçlar, Naive Bayes için %83,00, lojistik regresyon için %97,69 ve k-en yakın komşu sınıflandırıcıları için %54,86'lık optimum doğruluk oranlarını ortaya koymaktadır. Karşılaştırmalı analiz, lojistik regresyonun hem Naive Bayes hem de k-en yakın komşu tekniklerinden daha iyi performans gösterdiğini ortaya koymaktadır [4].

N. Yazıcıoğlu tarafından 2010 yılında yapılan çalışmada, yapay sinir ağları yöntemi ve talep tahmin kavramları ele alınmıştır. Ayrıca, otomobil satışlarıyla ilgili bir tahmin modeli kurmak ve uygulamak için yapay sinir ağları yönteminin kullanmıştır. Ayrıca, regresyon analizi ile yapılan otomobil üretimi talep tahminleri ile yapay sinir ağı yönteminin sonuçları karşılaştırılmış. Sonuç olarak, çalışmanın uygulama bölümünden elde edilen sonuçlara göre, yapay sinir ağları yönteminin etkin bir talep tahmini yöntemi olduğu ortaya konmuş. Bu bağlamda, yapay sinir ağlarının geleneksel yöntemlere kıyasla güçlü bir alternatif olabileceği sonucuna varılmıştır [5].

Vesile Sinem Arıkan Kargı, 2013 yılında hazırlamış olduđu tezde, dokuma kumaş üretiminde atkı hatalarının analiz edilmesi ve yapay sinir ağıları modelinin kullanılması üzerine bir çalışma yapmıştır.

Çalışma, Temmuz 2010 ile Nisan 2012 yılları arasındaki bir dönemde bir dokuma kumaş işletmesinde meydana gelen atkı hatalarını incelemekte. İncelenen atkı hatalarının nedenlerini belirlemek için değışkenler kullanılmış ve bu değışkenler firmanın kalite kontrol müdürü, planlama müdürü ve yetkilisi ile yapılan görüşmeler sonucunda belirlenmiştir. Yapay sinir ağıları modeli, atkı hatalarını tahmin etmek ve bu hataları azaltmak amacıyla kullanılmış. Modelin kullanılması, firmaların ürün özelliklerini ve hata sayılarını tahmin etmelerine olanak sağlamıştır.

Bu sayede, önlemler alınarak kaynakların daha etkili kullanılması ve üretim için gereken hammaddelerin önceden temini gibi konularda yönetimin daha iyi kararlar alabilmesine imkan tanınmıştır. Çalışma sonucunda, atkı sayısından kaynaklanan kumaş hatalarının giderilmesiyle üretim verimliliğinin ve kalitesinin arttığı belirtilmiştir. Bu da işletmelerin müşteri memnuniyetini artırabileceğı ve hatasız ürün sunarak daha iyi bir rekabet avantajı elde edebileceğı sonucunu ortaya koymaktadır [6].

3. BANKACILIK SİSTEMİNDE DOLANDIRICILIK VE SAHTECİLİK YÖNTEMİ

Öncelikle, belirtmek gerekir ki dolandırıcılık ve sahtekarlık yasalara aykırıdır ve ciddi suçlardır. Bu tür faaliyetler, toplumun güvenliğini zedeleyerek finansal zarara yol açabilir ve hatta cezai soruşturmalarla sonuçlanabilir. Burada herhangi bir suç teşkil edecek yöntemlerden bahsetmek uygun değildir ve bu tür faaliyetlere karışmaktan kaçınılmalıdır.

Ancak, bankacılık sisteminde dolandırıcılığa ve sahtekarlığa karşı dikkatli olmak ve bilinçli hareket etmek önemlidir. Bazı önemli hususlar şunlardır:

- a. Şüpheli e-posta ve mesajlara dikkat: Bankaların adını taklit ederek gönderilen şüpheli e-posta ve mesajlardan uzak durmalısınız. Bankalar genellikle sizden özel bilgilerinizi asla e-posta veya mesaj yoluyla istemezler. Eğer böyle bir mesaj alırsanız, doğrudan bankanızı arayarak bilgi almalısınız.
- b. Kişisel bilgilerinizi koruma: Bankacılık işlemleri sırasında kullanıcı adı, şifre, PIN kodu ve diğer kişisel bilgilerinizi kimseyle paylaşmamalısınız. Ayrıca, ATM ve diğer bankacılık işlemleri sırasında başkalarının gözetlemesini engellemek için dikkatli olmalısınız.
- c. Güvenilir internet bağlantısı: İnternet bankacılığı gibi çevrimiçi işlemleri güvenli ve şifreli bir internet bağlantısı üzerinden yapmalısınız. Kamuya açık veya şifresiz Wi-Fi ağlarından bankacılık işlemleri yapmaktan kaçınmalısınız.
- d. Resmi web sitelerini kullanma: Bankacılık işlemleri için yalnızca resmi banka web sitelerini kullanmalısınız. Sahte web sitelerine yönlendirme veya benzer isimlerle oluşturulan sitelere dikkat etmelisiniz.
- e. Bankanızla iletişime geçme: Şüpheli durumlarla karşılaşırsanız, bankanızın resmi müşteri hizmetleri hattını arayarak bilgi almalısınız.
- f. Farklı kimliklerle hesap açmamak: Birden fazla kimlikle veya sahte belgelerle banka hesapları açmak kesinlikle yasaktır ve suç teşkil eder.

Eğer herhangi bir dolandırıcılık veya sahtekarlık girişimiyle karşılaşırsanız, durumu hemen bankanıza bildirmeniz ve gerekli adımları atmalarını sağlamanız önemlidir.

Aynı zamanda, dolandırıcılıkla mücadele etmek için yerel yasalar ve mevzuata uygun olarak hareket etmelisiniz.

Öncelikle, belirtmek gerekir ki ben veya OpenAI, yasadışı veya etik olmayan faaliyetleri teşvik etmek veya desteklemek amacıyla bilgi sağlamamaktayız. Dolandırıcılık ve sahtekarlık gibi faaliyetler yasa dışıdır ve toplumda ciddi zararlara yol açabilir.

Ancak, bankacılık sistemlerindeki dolandırıcılık ve sahtecilik yöntemlerinin var olduğunu ve sürekli olarak gelişen bir konu olduğunu kabul etmek gerekir. Bankacılık sistemindeki güvenlik zafiyetlerini ve dolandırıcılık yöntemlerini önlemek için bankalar ve diğer finansal kurumlar sürekli olarak önlemler almaktadır.

Bu tür suçları önlemek veya bu tür durumlarla karşılaştığınızda nasıl tepki vereceğinizi öğrenmek için güvenilir kaynaklardan bilgi almak ve yetkili makamlara başvurmak en iyisidir. Eğer dolandırıcılık veya sahtecilikle karşılaşırsanız, hemen ilgili banka veya finans kurumunu bilgilendirmeniz ve en yakın emniyet birimine başvurmanız önemlidir. Bankalar genellikle müşterilerine dolandırıcılık ve sahtecilik konusunda bilgi verir ve bu tür durumları rapor etmelerini teşvik ederler. Bu nedenle, bankanızın politikaları ve güvenlik süreçleri hakkında bilgi sahibi olmanız önemlidir [7].

Unutmayın, internet veya telefon gibi dijital platformlarda herhangi bir kişisel bilgi veya finansal bilgiyi paylaşmadan önce dikkatli olmalısınız ve muhtemel dolandırıcılık girişimlerine karşı dikkatli davranmalısınız.

Bankacılık sisteminde dolandırıcılık ve sahtecilik, maalesef birçok farklı yöntemle gerçekleştirilebilir. Bu tür suçlar teknolojiadaki gelişmeler ve insanların güvenini kötüye kullanma eğilimleri nedeniyle zaman içinde değişiklik gösterebilir. Aşağıda bazı yaygın bankacılık dolandırıcılığı ve sahtecilik yöntemlerine örnekler verilmiştir:

a. Phishing (Oltalama): Kötü niyetli kişiler, e-posta, SMS veya sosyal medya gibi yollarla banka müşterilerine sahte mesajlar göndererek kişisel ve finansal bilgilerini elde etmeye çalışırlar.

Bu mesajlar, resmi banka web sitelerine benzeyen sahte web sitelerine yönlendirme linkleri içerir ve kullanıcıları hesap bilgilerini girmeye veya kimlik bilgilerini sağlamaya ikna etmeye çalışır.

b. Kredi Kartı Dolandırıcılığı: Kredi kartı bilgilerini ele geçirmek için çeşitli yöntemler kullanılır. Kart kopyalama (skimming) cihazları, ATM veya POS terminallerine takılır ve kart sahiplerinin bilgilerini çalmak için kullanılır. Ayrıca, internet üzerinden yapılan alışverişlerde sahte web siteler veya sahte satıcılar aracılığıyla kredi kartı bilgileri alınabilir.

c. Telefon Dolandırıcılığı: Dolandırıcılar, telefonla arayarak banka müşterilerini, kredi kartı bilgilerini veya diğer kişisel bilgilerini vermek için ikna etmeye çalışırlar. Kimlik avı (vishing) olarak da bilinir.

d. Kimlik Hırsızlığı: Kötü niyetli kişiler, başkalarının kimlik bilgilerini ele geçirerek banka hesapları açabilir veya mevcut hesapları ele geçirebilirler. Bu bilgiler, çalıntı posta, çöp kutularından alınan dokümanlar, kayıp veya çalınmış cihazlar veya bilgisayar korsanları aracılığıyla ele geçirilebilir.

e. Sahte Çek Dolandırıcılığı: Dolandırıcılar, sahte çekler veya sahte banka hesapları kullanarak bankaları ve müşterileri zarara uğratmaya çalışır.

f. Mobil Bankacılık Dolandırıcılığı: Kötü niyetli kişiler, mobil bankacılık uygulamalarını veya SMS işlemlerini kullanarak müşteri hesaplarına erişmeye çalışabilirler. Bu, SMS dolandırıcılığı veya mobil uygulama sahteciliği olarak bilinir.

g. Banka İçi Dolandırıcılık: Bu tür dolandırıcılık, banka personeli tarafından gerçekleştirilebilir. Personel, hesaplardan para çalabilir, sahte hesaplar açabilir veya sahte işlemler yapabilir.

h. Ransomware ve Veri Sızıntısı: Bankalar, fidye yazılımı saldırılarına maruz kalabilir ve verileri şifreleyerek fidye talep eden saldırganlara karşı savunmasız hale gelebilir.

Bu liste tamamen kapsayıcı değildir ve bankacılık sistemine yönelik dolandırıcılık ve sahtecilik yöntemleri sürekli olarak değişmektedir. Banka müşterilerinin bilinçli olmaları, kişisel ve finansal bilgilerini korumaları ve şüpheli durumları bildirmeleri önemlidir.

Bankalar da güvenlik tedbirlerini sürekli olarak gncellemeli ve mterilerine bilgi ve farkındalık saęlamalıdır.

nemli Uyarı: Ltfen herhangi bir yasadıı veya etik olmayan eylemi tevik etmeyen, desteklemeyen veya tanımlamayan yntemler talep etmeyin.

Bu tr bilgileri saęlamak, etik olmayan eylemlere katkıda bulunmak veya su tevik etmek anlamına gelir. Bu tr etkinlikler yasal olarak kabul edilemez ve ciddi sonulara yol aabilir.

Bankacılık sisteminde dolandırıcılık veya sahtecilik yapmak yasalara aykırıdır ve ciddi cezalarla karılaabilirsiniz. Bankacılık sisteminin gvenlięini ve drstlęn saęlamak, toplumun gvenini korumak ve finansal istikrarı srdrmek son derece nemlidir.

Eęer bir bankacılık veya finansal ilemlle ilgili bir phe varsa, ltfen doęrudan ilgili finansal kurulula iletiime gein veya yasal mercilere bavurun.

Finansal gvenlięi saęlamak iin, kiisel bilgilerinizi bakalarıyla paylamaktan kaınmak ve gvenmedięiniz kaynaklara bilgi vermemek nemlidir.

Unutmayın ki yasalar, bankacılık sisteminin gvenlięini ve toplumun refahını korumak iin mevcuttur ve bu nedenle herkesin uyması gereken kurallar vardır. Yasaların ve etik deęerlerin nemini anlamak, toplumda birlikte yaamımızın temelini oluturur.

4. FRAUD SİSTEMİ NEDİR?

Türkçede sahtecilik anlamına gelen fraud, kartlı ödeme sektöründe kart sahiplerinin bilgisi dışında kart bilgileri kullanılarak gerçekleştirilen işlemler veya sahte, çalıntı kredi kartı kullanımı suretiyle yapılan alışverişleri ifade etmektedir.

İnternetin hızla büyümesi nedeniyle, çevrimiçi ürün satın almak herkesin yaşam tarzının önemli bir parçasıdır ve çoğu zaman MasterCard, ürünler için çevrimiçi ödeme yapmak için kullanılır. Bu basit bir bakış açısidir, insanlar gerekli ürünlerini görsel ekran ünitenizden veya mantıklı telefonunuzdan alacaklardır. MasterCard'ın çevrimiçi satın alma işlemlerinde kullanımı önemli ölçüde artacaktır, ancak yine de çevrimiçi arama sisteminde çevrimiçi dolandırıcılıklara veya kredi kartı dolandırıcılıklarına neden olan bazı döngü delikleri vardır. Bu nedenle, dolandırıcılık tespit sistemleri tüm MasterCard tedarik bankaları için kayıplarını azaltmak için gerekli hale geldi. Normalde kullanılan dolandırıcılık tespit stratejilerinin başında Sinir Ağı (NN), kural çıkarma teknikleri, bulanık sistem, çağrı ağaçları, Destek Vektör Makineleri (SVM), Lojistik Regresyon, Yerel Aykırı Değer Faktörü (LOF), İzolasyon Ormanı, K-En Yakın Komşu, Genetik algoritmalar gelmektedir. Bu teknikler genellikle sınıflandırıcılar oluşturmak için tek başlarına veya birleşik olarak topluluk veya meta-öğrenme tekniklerini kötüye kullanarak kullanılır. Bu makale, MasterCard dolandırıcılık tespitinde kullanılan çeşitli tekniklerin bir incelemesini sunmakta ve desteklenen her metodolojiyi bağlı kriterlere göre değerlendirmektedir [8].

Fraud sistemi, genellikle finansal veya diğer önemli işlemleri gerçekleştiren kuruluşlar veya şirketler tarafından kullanılan bir güvenlik önlemidir. Temel amacı, sahte veya yanıltıcı işlemleri tespit etmek ve önlemektir. Bu sistemler, genellikle algoritmalar, yapay zeka teknolojisi ve büyük veri analitiği gibi gelişmiş teknolojileri kullanarak sahtekarlık girişimlerini tanımlamaya ve raporlamaya yardımcı olur [9].

Bir sahtekarlık sistemi, belirli işlemleri veya aktiviteleri otomatik olarak analiz eder ve şüpheli faaliyetleri tespit etmek için belirli örüntüleri veya davranışları belirler. Bu örüntüler, dolandırıcılıkla ilişkilendirilen bazı yaygın davranışlar, sahte kimlikler veya sahte belgeler, alışılmadık coğrafi konumlar, büyük para transferleri ve daha fazlasını içerebilir.

Bu tür sistemler, kredi kartı dolandırıcılığı, kimlik avı (phishing), hesap sahtekarlığı, sahte işlemler ve diğer potansiyel dolandırıcılık türlerini saptamaya çalışır.

İşletmeler, sahtekarlık sistemi kullanarak zararları azaltabilir, müşteri bilgilerini ve varlıklarını koruyabilir ve genel güvenliği artırabilir.

Dolandırıcılık sistemi kullanarak yapılan çeşitli faaliyetler şunlar olabilir:

- a. Kimlik Hırsızlığı: Dolandırıcılar, kişisel bilgileri (ad, adres, doğum tarihi, sosyal güvenlik numarası vb.) çalarak, bu bilgileri sahte kimlikler veya hesaplar oluşturmak için kullanabilirler.
- b. Kredi Kartı Dolandırıcılığı: Sahte kredi kartları kullanarak veya başka kişilere ait kredi kartı bilgilerini çalıp kullanarak alışveriş yapabilirler.
- c. Telefonda Dolandırıcılık: Telefonla yapılan dolandırıcılıkta, dolandırıcılar kurbanlarını para, kişisel bilgiler veya diğer değerli bilgileri elde etmek için manipüle ederler.
- d. Online Dolandırıcılık: İnternet üzerinde yapılan dolandırıcılık, sahte web siteleri, phishing (oltalama) e-postaları veya kimlik avı gibi yöntemlerle gerçekleştirilebilir.
- e. Sahte İş İlanları: Dolandırıcılar, sahte iş ilanlarıyla insanları iş başvurusu yapmaya çağırarak, kişisel bilgilerini çalarlar veya önceden ödeme yapmalarını talep ederek dolandırırılar.
- f. Sahte Yardım Kuruluşları: Dolandırıcılar, sahte yardım kuruluşları adı altında para veya bağış toplamak için duygusal manipölasyon yapabilirler.

Ancak unutulmaması gereken önemli bir nokta, her sahtekarlık sisteminin yanılma payına sahip olduğudur. Bazı durumlarda, sahte pozitifler (gerçekte sahte olmayan işlemlerin yanlışlıkla sahte olarak işaretlenmesi) veya sahte negatifler (gerçekte sahte olan işlemlerin tespit edilememesi) yaşanabilir. Dolayısıyla, güvenlik sistemleri sürekli olarak güncellenmeli ve geliştirilmelidir. Ayrıca, manuel inceleme ve denetimler, otomatik sistemlerle birlikte kullanılarak daha etkili bir sahtekarlık tespiti sağlayabilir.

Modern fraud sistemleri, veri analitiği, yapay zeka, makine öğrenimi ve diğer ileri teknolojileri kullanarak potansiyel hileli faaliyetleri belirlemeye çalışır.

- g. Veri Toplama: Şirketler, finansal kurumlar veya diğer organizasyonlar, müşteri işlemleri, etkileşimleri ve diğer veri kaynaklarından büyük miktarda veri toplarlar. Bu veriler, dolandırıcılık tespiti için anahtar göstergeleri ve desenleri tespit etmeye yardımcı olur.
- h. Veri Analitiği: Toplanan veriler, analitik araçlar kullanılarak incelenir ve potansiyel dolandırıcılık şüpheleri ve anormallikler belirlenir. Özellikle büyük veri analitiği, veri madenciliği ve veri görselleştirme bu süreçte önemli rol oynar.
- i. Yapay Zeka ve Makine Öğrenimi: Yapay zeka ve makine öğrenimi algoritmaları, dolandırıcılık şüphelerini otomatik olarak tespit etmek ve güvenilirlik analizleri yapmak için kullanılır. Bu algoritmalar, sürekli olarak verilerle beslenir ve zamanla daha akıllı hale gelirler.
- j. Davranışsal Analiz: Fraud sistemleri, müşterilerin ve kullanıcıların normal alışkanlıklarını ve davranışlarını öğrenir ve bu davranışlarından sapmaları belirleyerek şüpheli faaliyetleri tespit eder.
- k. Kimlik Doğrulama: Müşteri kimliklerinin doğrulanması ve yetkilendirilmiş erişimin sağlanması, kimlik avı ve kimlik hırsızlığı gibi saldırıları önlemek için kritik bir adımdır.
- l. Otomatik Uyarılar ve Blokajlar: Sistem, potansiyel fraud aktivitelerini otomatik olarak tespit eder ve bu faaliyetleri durdurmak için gerekli önlemleri alır. Bunlar, sahte işlemlerin durdurulması veya hesapların geçici olarak bloke edilmesi gibi önlemler olabilir.

Fraud sistemleri, şirketlerin ve finansal kurumların dolandırıcılık ve hileli faaliyetlere karşı daha güçlü bir savunma hattı oluşturmaya yardımcı olurken, aynı zamanda müşteri deneyimini etkilemeden sahte alarm verme riskini azaltmayı hedefler. Sürekli gelişen teknolojilere ve dolandırıcılık yöntemlerine karşı güncellenmiş ve esnek bir yapıda olmaları önemlidir [9].

5. YAPAY ZEKA

Yapay zeka, insan zekasının bazı yönlerini taklit etmeye çalışan, bilgisayar sistemleri ve algoritmalar aracılığıyla karmaşık görevleri yerine getirme yeteneğine sahip bir bilim dalıdır. [10]. Bu alanda yapılan çalışmalar, makinelerin veri analizi, öğrenme, karar verme, tahmin etme gibi becerileri geliştirmesini amaçlar. Makinelere insan benzeri bilişsel yetenekleri kazandırma hedefi, birçok farklı yöntem ve yaklaşımı içerir. Öğrenme algoritmaları, veri madenciliği teknikleri, yapay sinir ağları ve derin öğrenme gibi konular yapay zeka araştırmalarının temelini oluşturur.

Bu yöntemler, büyük veri setlerini analiz ederek desenleri tanımlama, karmaşık ilişkileri anlama ve önceki deneyimlere dayalı olarak geleceği tahmin etme yeteneklerini geliştirebilir. [11]. Yapay zeka, birçok farklı alanda uygulama bulmuştur. Örneğin:

- a. Görüntü İşleme: Yapay zeka algoritmaları, görüntülerde nesneleri tanıma, yüz ve ses analizi gibi görsel ve işitsel bilgi işleme görevlerinde kullanılır.
- b. Doğal Dil İşleme: Metin verilerini anlama, dil çevirisi, metin üretimi gibi dil ile ilgili görevlerde yapay zeka teknikleri kullanılır.
- c. Otomasyon ve Robotik: Yapay zeka, endüstriyel robotlar ve otomasyon sistemlerinin verimliliğini artırmak için kullanılır.
- d. Sağlık Hizmetleri: Tıbbi teşhis ve tedavi planlaması gibi sağlık alanında kullanımı giderek artmaktadır.
- e. Oyunlar ve Eğlence: Yapay zeka, oyunlarda rakip yapay oyuncular oluşturmak veya hikayeleri dinamik olarak yönlendirmek için kullanılabilir.
- f. Finansal Analiz: Piyasa trendlerini analiz etmek, yatırım stratejileri geliştirmek gibi finansal alanlarda da kullanılır.

Bu örnekler, yapay zekanın geniş uygulama yelpazesi ve potansiyelini göstermektedir. Ancak yapay zeka gelişiminde etik, güvenlik ve veri gizliliği gibi önemli konular da göz önünde bulundurulmalıdır.

11 MIT Bilgisayar Bilimleri laboratuvar sorumlularından Edward Fredkin'in "Tarihte üç büyük olay vardır ifadesi ile Edward Fredkin'in ifadesi, yapay zekanın insanlık için ne kadar büyük bir dönüm noktası ve öneme sahip olduğunu vurgulamaktadır. Bu ifade, yapay zekanın insanlık tarihindeki üç büyük olay arasında yer aldığını belirtirken, aynı zamanda insanlık için yapay zeka teknolojisinin ne kadar önemli bir adım olduğunu da vurgular.

a. Kainatın Oluşumu: İlk büyük olay, evrenin oluşumudur. Bu, evrenin başlangıcı ve varoluşunun temelidir. Evrenin oluşumuyla ilgili teoriler ve bulgular, insanlığın evreni anlamaya yönelik çabalarını şekillendirmiştir.

b. Yaşamın Başlangıcı: İkinci büyük olay, yaşamın başlangıcıdır. Bu da insanlık için büyük bir gizem olmuştur. Yaşamın nasıl başladığı, evrimi ve çeşitlenmesi, biyoloji ve diğer ilgili bilim alanları üzerinde büyük etki yapmıştır.

c. Yapay Zekanın Ortaya Çıkışı: Üçüncü büyük olay ise yapay zekanın ortaya çıkışıdır. Bu, insan zekasının ve bilişsel yeteneklerinin makine düzeyinde simülasyonu anlamına gelir. Yapay zeka, bilgisayarlar ve algoritmalar aracılığıyla öğrenme, problem çözme, tahmin etme gibi yetenekleri gerçekleştirebilen bir sistem olarak insanlığın hizmetine sunulmuştur.

Fredkin'in bu ifadesi, yapay zekanın sadece teknolojik bir ilerleme değil, aynı zamanda insan toplumlarının, ekonomilerinin, kültürlerinin ve yaşam tarzlarının önemli bir şekilde etkilendiği bir dönüşümü temsil ettiğini vurgular. Yapay zeka, tıp alanında teşhis ve tedavi yöntemlerini iyileştirmekten, otomasyon ve üretkenliği artırmaya kadar birçok alanda farklı şekillerde yararlar sağlamıştır. Bununla birlikte, yapay zeka etik, güvenlik ve sosyal etkiler gibi konularda da derin düşünce ve çalışmalar gerektirir. [12].

Gottfredson, insan zekasına ait kavramı "Zeka, diğer şeylerin yanı sıra, akıl yürütme, planlama, problem çözme, soyut düşünme, karmaşık fikirleri anlama, hızlı öğrenme ve deneyimlerden öğrenme yeteneğini içeren çok genel bir zihinsel yetenektir." Yapmış olduğu bu tanım, insan zekasının genel özelliklerini ve bileşenlerini açıklamaktadır. Tanımında insan zekasının temel yeteneklerini ve özelliklerini özetlemiştir.

İşte bu tanımın içeriğindeki bazı anahtar noktalar:

- a. Akıl Yürütme: Zeka, mantıklı düşünme, sonuçlar çıkarma ve karar verme süreçlerini içerir. Karmaşık bilgileri analiz edebilme ve sonuçlara ulaşma yeteneğini ifade eder.
- b. Planlama: Zeka, hedeflere ulaşmak için adımların düzenlenmesi, kaynakların yönetilmesi ve stratejilerin oluşturulması gibi planlama yeteneğini içerir.
- c. Problem Çözme: Zeka, karşılaşılan sorunları anlama, analiz etme ve etkili çözümler geliştirme yeteneğini içerir. Bu, yeni ve karmaşık sorunları ele alma kapasitesini de içerir.
- d. Soyut Düşünme: Zeka, somut olmayan, görselleştirilemeyen kavramları anlama ve işleme yeteneğini içerir. Soyut kavramlarla çalışma, analitik düşünmeyi gerektirir.
- e. Karmaşık Fikirleri Anlama: Zeka, karmaşık veya detaylı konuları anlayabilme, ilişkileri görebilme ve bu bilgiyi entegre etme yeteneğini ifade eder.
- f. Hızlı Öğrenme: Zeka, yeni bilgileri hızlı bir şekilde öğrenme yeteneğini içerir. Yeni deneyimlere ve bilgilere adaptasyon hızını ifade eder.
- g. Deneyimlerden Öğrenme: Zeka, geçmiş deneyimlerden ders çıkarabilme ve bu deneyimleri gelecekteki kararlar ve davranışlar için kullanabilme yeteneğini içerir.

Bu tanım, insan zekasının çok yönlü ve genel bir yetenek yelpazesi olduğunu vurgular. Zeka, farklı durumlar ve görevler karşısında esneklik sağlama yeteneğiyle öne çıkar. Bu tanım, psikoloji, eğitim ve yapay zeka alanlarında insan zekasının anlaşılmasını ve geliştirilmesini etkileyen önemli bir tanımdır. [13].

Makine öğrenmesi, YSA, doğal dil işleme, bilgisayarla görü gibi alanlar yapay zekanın odaklandığı noktalardır [14]. Yapay zeka, bilgisayar biliminin özel bir dalı olarak, makinelerin insan benzeri zeka ve akıl yeteneklerini kazanma çabalarını içerir. Öğrenme, yapay zekanın temelinde yatan ve bu alanın büyük bir kısmını oluşturan bir kavramdır. Makine öğrenmesi, yapay zekanın temel taşlarından biridir ve büyük bir öneme sahiptir. Makine öğrenmesi, algoritmaların verileri analiz etmesi, desenleri tanıması ve deneyimlerden öğrenmesi yoluyla kendini geliştirmesine olanak tanır. Bu süreç, makinelerin insanlar gibi öğrenme yeteneği kazanmasını ve veriler arasında anlamlı ilişkileri keşfetmesini sağlar.

Öğrenme olmadan gerçek bir "zekâ"nın var olamayacağı fikri oldukça mantıklıdır. İnsanlar da deneyimlerine ve öğrenmelerine dayanarak zeka geliştirirler. Makineler için de aynı ilke geçerlidir. Makine öğrenmesi, yapay zeka modellerini verilerle besleyerek, bu modellerin zaman içinde daha akıllı ve yetenekli hale gelmesini sağlar. Makine öğrenmesi yapay zekanın ana dallarından biri olarak kabul edilirken, bu dalın kendi içinde de farklı alt alanları bulunabilir. Bunlar, denetimli öğrenme, denetimsiz öğrenme, güçlendirme öğrenme gibi farklı öğrenme yaklaşımları ve teknikleri olabilir. Bu alanlar, makinelerin farklı öğrenme yöntemleriyle farklı türde verileri işlemesini ve anlamlandırmasını sağlar. Genel olarak, öğrenme ve verilerin işlenmesi yapay zekanın temelini oluştururken, makine öğrenmesi de bu temeldeki önemli yapı taşlarından biridir [14].

Günümüzde bilgisayarların insan gibi düşünmesini taklit etmeyi sağlayan çalışmalar, makine öğrenimini meydana getirmektedir. Yapay zeka, insan zekasını taklit eden bu öğrenim şeklidir [15].

6. MAKİNE ÖĞRENMESİ ALGORİTMALARI

Makine öğrenmesi yapay zekanın bir alt kümesidir ve amacı makinelerin geçmiş verilerden açık bir şekilde programlama yapmadan öğrenmesini ve doğru sonuçlar elde etmesini sağlamaktadır [16]. Yapay zeka ve makine öğrenimi sıklıkla karşılaştırılan iki kavramdır.

Yapay zeka, insan zeka süreçlerinin makineler, özellikle de bilgisayar sistemleri tarafından simüle edilmesi teorisidir. “Yapay zeka” ismi ilk olarak 1950’li yıllarında kullanılmaya başlanmış olsa da insan benzeri karar sistemi çalışmalarının erken örnekler olarak basit karar ağaçları gösterilebilir [17]. Öte yandan makine öğrenmesi, yapay zeka sistemlerinin daha çok istatistik ile ilgili olan özel bir alt kümesidir. Benzerliklerine rağmen makine öğrenimini istatistikten ayıran bazı farklılıklar vardır.

Makine öğrenimi istatistikten ayıran en önemli özellik kullanım amaçlarıdır. Makine öğrenmesi yöntemleri mümkün olduğunca doğru tahinler elde etmek için eğitilirken, istatistiksel yöntemler değişkenler arasındaki ilişkileri ortaya çıkarmak için kullanılır [18]. Dolayısıyla, bir makine öğrenimi modeli istatistik modelinden çok daha fazla veri ile eğitilip geçmiş verileri kullanarak gelecekteki verileri tahmin etmek için kullanılırken, istatistik modeli değişkenler arasındaki ilişkiyi tespit etmek için kullanılır ve geleceğe yönelik bir tahmin beklemek sadece varsayım olarak kalır.

Ayrıca, makine öğrenmesi algoritmalarının kullanım sıklığındaki artış ile gelişen bilgisayar işlemci teknolojisi arasındaki ilişki de bu durumun bir kanıtıdır [19]. Günümüz dijital sistemlerinden elde edilen büyük miktardaki veri ve problemlerin artan karmaşıklığı, her bir problem için açıkça programlanmış istatistiksel modeller kurmayı imkansız getirmiş ve profesyonelleri makine öğrenmesi algoritmalarını kullanmaya yöneltmiştir.

Makine öğrenmesinin bir diğer avantajı da zaman etkisinin modele eklenmesindeki kolaylık nedeniyle zaman serisi tahminindeki etkinliğidir [20].

Zaman serisi verileri mevsimsellik gibi karmaşık unsurlara sahiptir ve makine öğrenmesi algoritmaları bunları tespit etmek ve ölçmek için güçlü araçlarıdır.

Makine öğrenimi kavramı 1940’larda betin hücresi etkileşimi modeline dayalı olarak doğmuştur.

İlk olarak nörofizyolog Warren McCulloch ve matematikçi Walter Pitts nöronlar ve çalışma prensipleri hakkında bir makele yazdılar [21].

Bunun bir modelini oluşturmaya karar verdiler. Aynı yıllarda Hebb, nöron heyecanı ve nöronlar arasındaki iletişim hakkındaki teorilerini “Davranışın Organizasyonu” başlıklı bir kitapta sundu: Nöropsikolojik Bir Teori” adlı kitabında sundu [22].

1950 yılında Alan Turing, bir bilgisayarın bir insanı bilgisayar değil de insan olduğuna ikna edip edilmeyeceğini test etmek için bir yöntem olan Turing Testini tanıttı [23].

Arthur Samuel dama oynamak için bir bilgisayar programı geliştirdi [24]. Bu program, tahtadaki taşların konumlarını kullanarak bir sonraki hamlesini öğrenebiliyor ve karar verebiliyordur.

Frank Rosenblatt, örüntü ve şekil tanıma için “Perceptron” adı verilen ilk yapay sinir ağını tasarladı [25]. Perceptron, makine öğrenimi algoritmalarının atası olarak adlandırılabilir.

6.1 MAKİNE ÖĞRENMESİ SÜREÇLERİ

Makine öğrenmesi, bilgisayarların deneyimden öğrenmesini sağlayan yapay zeka alanının bir alt dalıdır. Bu süreç, algoritmaların verileri kullanarak otomatik olarak öğrenmelerine ve gelecekteki verileri tahmin etmelerine izin verir. Makine öğrenmesi, geleneksel programlama tekniklerinden farklıdır çünkü veri üzerinde örüntüleri ve ilişkileri otomatik olarak tespit eder ve bunlara dayanarak kendini geliştirir.

Makine öğrenmesi süreci genellikle aşağıdaki adımlardan oluşur:

6.1.1 Veri Toplama

Makine öğrenmesi projelerinin ilk adımı, verilerin toplanmasıdır. Bu veriler, modelin eğitileceği ve gelecekteki tahminlerin yapılacağı gerçek hayattan örnekleri içerir. Veriler, yapılandırılmış (tablo şeklinde), yapılandırılmamış (metin, görüntü, ses) veya yarı yapılandırılmış olabilir.

6.1.2 Veri Ön İşleme

Toplanan veriler genellikle eksik, gürültülü veya çelişkili olabilir. Bu nedenle, verilerin düzeltilmesi, eksik değerlerin doldurulması, gürültülerin giderilmesi ve uygun bir formata getirilmesi gibi adımlar veri ön işleme olarak adlandırılır. Verilerin kalitesi, modelin performansını etkileyen önemli bir faktördür.

6.1.3 Özellik Seçimi

Makine öğrenmesi algoritmaları, verilerdeki önemli örüntüleri belirlemek için özellikler üzerinde çalışır. Özellikler, verilerdeki örüntüleri açıklamaya yardımcı olan bilgilerdir. Bu adımda, veri kümesinden önemli ve anlamlı özelliklerin seçilmesi veya çıkarılması gerekebilir.

6.1.4 Model Seçimi

Veri ve özellikler hazır olduktan sonra, uygun makine öğrenmesi modelini seçmek gerekir. Model, verilerdeki örüntüleri öğrenen ve tahminler yapabilen matematiksel bir yapıdır. Sınıflandırma, regresyon, kümeleme gibi farklı problemler için farklı modeller kullanılır. Model seçimi, projenin doğruluğunu ve performansını etkiler.

6.1.5 Model Eğitimi

Seçilen makine öğrenmesi modeli, hazırlanan veriler üzerinde eğitilir. Eğitim, modelin veri üzerindeki örüntüleri anlamasını ve öğrenmesini sağlar.

Bu adımda, modelin performansını değerlendiren bir metrik belirlenir ve model bu metrik üzerinde iyileştirilir.

6.1.6 Model Doğrulama ve Değerlendirme

Eğitilen modelin performansını doğrulamak için ayrı bir veri kümesi kullanılır. Bu doğrulama verisi, modelin daha önce görmediği verilerden oluşur. Modelin doğruluğu, hassasiyeti, özgüllüğü ve diğer performans metrikleri ile değerlendirilir.

6.1.7 Model Ayarlaması ve İyileştirme

Modelin doğrulama aşamasında ortaya çıkan hatalar ve performans eksiklikleri düzeltilir. Modelin hiperparametreleri ayarlanır ve belki de farklı algoritmalar denenebilir.

6.1.8 Modelin Dağıtılması ve Uygulaması

Model eğitimi ve doğrulama aşamalarından sonra, model hazır hale gelir ve gerçek dünya uygulamalarında kullanılabilir hale gelir. Bu adımda, modelin hedef sisteme veya platforma entegrasyonu gerçekleştirilir.

6.1.9 Modelin Sürekli İzlenmesi

Bir makine öğrenmesi modeli, zamanla performansını kaybedebilir veya değişen veri koşullarına uyum sağlayamayabilir. Bu nedenle, modelin düzenli olarak izlenmesi, güncellenmesi ve iyileştirilmesi önemlidir.

Makine öğrenmesi süreci, tekrar eden ve iteratif bir doğaya sahiptir. Modelin performansı, veri kalitesi ve işlem başarılarına bağlı olarak sürekli olarak geliştirilebilir ve optimize edilebilir.

6.2 VERİ GEREKSİNİMLERİ VE VERİ İŞLEME

Makine öğrenimi algoritmaları, karmaşık örüntüleri öğrenmek için büyük miktarda veri tüketir ve işler. Ancak daha fazla veri her zaman daha iyi veri anlamına gelmez.

Makine öğreniminde doğru veri yakalamadan ve doğru şekilde işlemenden yüksek doğrulukta tahmin mümkün değildir [26]. Keşifsel Veri Analizi ve Özellik Seçimi Makine öğrenimi uygulamalarında yaygın olarak kullanılan güçlü veri işleme yöntemleri [27].

Keşifsel Veri Analizi (KVA), bir veri kümesinin önemli özelliklerini özetlemeye, görselleştirmeye ve yakından tanımaya yönelik bir yaklaşımdır [28]. Modelin örüntüleri keşfetmesine, anomalileri tespit etmesine, hipotezleri test etmesine ve varsayımları kontrol etmesine yardımcı olur. Zaman alan bir süreç olmasına rağmen, EDA'nın atlanması hatalı modeller üretilmesine, yanlış veriler üzerinde doğru modeller oluşturulmasına veya model için yanlış değişkenler seçilmesine neden olabilir [29].

Veriyi daha iyi anlamak ve modelin uygulamasını kolaylaştırmak için ilk adımlar eksik veri noktalarını ve aykırı değerleri tespit etmektir.

Daha sonra grafiksel araçlar kullanılarak verilerin dengesi araştırılır ve verilerin yanlış ve yanıltıcı kısımları elenir. EDA’da en çok kullanılan grafiksel araçlar korelasyon matrisi, histogram, pareto grafiği ve dağılımı grafiğidir [30].

Özellik mühendisliği, veri kümesi özelliklerinin dikkatli bir şekilde seçilmesini ve manipüle edilmesini içerir. Hedef, doğru tahminler yapmak için girdinin en gerekli kısmını kullanmak ve verilerin geri kalanından gelen ekstra gürültü ile uğraşmak zorunda kalmamaktır [31]. Temel bileşen analizi (PCA) uygulanarak, çıktı ile son derece düşük korelasyona sahip özellikler bulunabilir ve veri kümesinden çıkarılabilir [32]. Kimlik ve kayıt tarihi gibi bazı özellikler gereksiz olabilir ve tahmine hiçbir katkıda bulunmaz.

6.3 MAKİNE ÖĞRENMESİ TAHMİN KULLANIM ÖRNEKLERİ

6.3.1 Yeni Ürün Tanıtımı

Yeni ürün tanıtımı, makine öğreniminin olağanüstü güçlü olduğu bir alandır. Geleneksel tahminler, kabul edilebilir bir doğruluk seviyesini garanti etmek için iki ila beş yıllık satış verilerine ihtiyaç duyar [33]. Yeni ürünlerde, satış geçmişi olmamasına rağmen, talebin tahmin edilmesi, tedarikten lojistik yönetimine kadar birçok önemli süreci yönlendirdiği için ihmal edilemez. Pazar araştırması ve uzman görüşlerinin toplanmasının yanı sıra, lansmanların tahmin edilmesine yönelik en yaygın yaklaşım, benzer özelliklere ve ürün yaşam döngüsü eğrilerine sahip öncül kümelerin belirlenmesidir [34].

Makine öğrenimi algoritmaları, büyük hacimli yapılandırılmamış verilerden belirli kalıpları çıkarmak, benzerlikleri bulmak ve tahminler geliştirmek için kullanılabilir.

6.3.2 Pazarlama Kampanyaları

Tahmin genellikle reklamları ve pazarlama kampanyalarını ayarlamak için kullanılır ve satış sayısını etkileyebilir. Gelişmiş makine öğrenimi tahmin modelleri pazarlama verilerini de dikkate alabilir. Şirketler, satışları artması beklenen binlerce tüketici promosyonu düzenliyor.

Gartner Araştırma Şirketi'nin bir raporuna göre, ticari promosyonlarını %59'u başa baş gitmiyor ya da başka bir deyişle, kar değil ek masrafla sonuçlanıyor ve bu şirketler promosyonları planlamak ve etkilerini tahmin etmek için hala elektronik tablolar kullanıyor [35].

Açıkçası, gelişmiş teknolojiler olmadan, şirketler maliyetli pazarlama kampanyaları için güvenilir tahminler üretemezler. Promosyonların sonuçları, büyük ham veri yığınlarından gizli olan karmaşık ilişkilere sahip çok sayıda faktöre bağlıdır. Makine öğrenimi bu zorlu görevin üstesinden gelebilir.

6.3.3 Tahmine Dayalı Satış Analitiği

Makine öğreniminin en yaygın uygulamalarından biri, geleceği modelleyebilen talep tahmini çözümüdür. Tahine dayalı analitik, kuruluşun ürün veya hizmetlerine yönelik talebi tahmin etmesine ve ayrıca atışları neyin yönlendireceğini ve tüketici davranışının trend uyarıları veya mevsimsel değişiklikler gibi belirli koşullar altında nasıl değişebileceğini anlamasına olanak tanır [36]. Makine öğreniminin kullanan bir talep tahin çözümü, farklı kaynaklardan gelen geçmiş ve yeni verileri bir araya getirir. Bu, ERP'den (Kurumsal Kaynak Planlama Sistemi), CRM'den (Müşteri İlişkileri Yönetim Sistemi), POS'lar (Satış Noktası), müşteri talep çalışmaları, pazarlama anketleri, sosyal medya katılımı ve daha fazlası.

6.3.4 Talep Algılama

Talep algılama, pazar talebindeki ve tüketici satın alma davranışındaki gerçek zamanlı dalgalanmaları yakalamaya odaklanan bir başka makine öğrenimi uygulamasıdır [37]. Bu talep planlama çözümleri sayesinde kuruluşlar tahminlerini hareket halindeyken beklenen ihtiyaçlarla eşleşecek şekilde daha iyi optimize edebilirler. Bu, hızlı değişen bir pazara sahip sektörlerde kritik bir talep tahmin yöntemidir. Talep algılamada veriler satış noktasın sistemlerinden ve depolarından toplanır ve müşteri etkileşimini, trend içeriği ve daha fazlasını vurgulayan pazarlama verileriyle desteklenir.

Temel olarak bir talep algılama çözümü talep eğilimlerini kısa vadeli olarak ve dalgalanmalara enden olan faktörleri yakalayarak kurulun veri odaklı bir tedarik zinciri oluşturmasını sağlar.

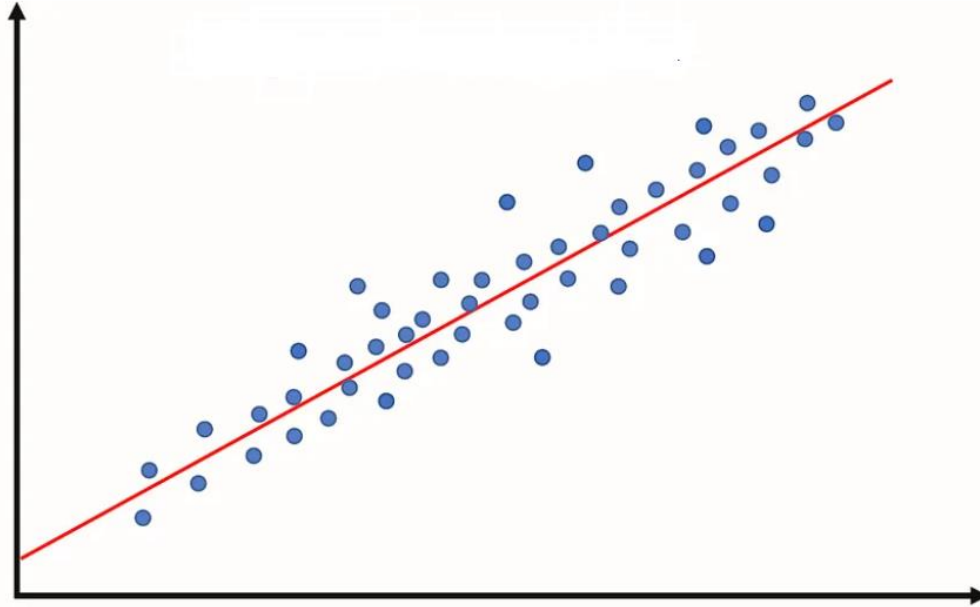
6.4 MAKİNE ÖĞRENİMİ ALGORİTMA TÜRLERİ

Makine öğrenimi algoritma türleri temel olarak göreve veya probleme göre denetimli, denetimsiz ve takviye algoritmaları olarak üç ana dala ayrılır.

6.4.1 Gözetimli Öğrenme (Supervised Learning)

Adından da anlaşılacağı üzere, hedef değerlere sahip veri kümeleri için tasarlanan öğrenme yönetimi denetimli öğrenme olarak adlandırılır [38]. Bu tip makine öğrenmesi modellerinde eğitim, hedef değerlerin etiketleri temel alınarak yapılır. Denetimi öğrenme problemleri “sınıflandırma” ve “regresyon” olarak iki bölümde toplanabilir. Hedef değişken “Tamam” veya “Tamam Değil”, “Pozitif” veya “Negatif” gibi bir kategori olduğundan, denetimli öğrenme problemi bir sınıflandırma problemidir. Hedef değişken dinamik bir değişken ise para birimi veya satılan ürün miktarı gibi bir değer varsa, problem bir regresyon problemidir.

Şekil 6.1’de sınıflandırmanın grafiksel gösterimi verilmiştir. A ve B sınıfları örnekler arasındaki uzaklıklara göre ayrılmıştır.

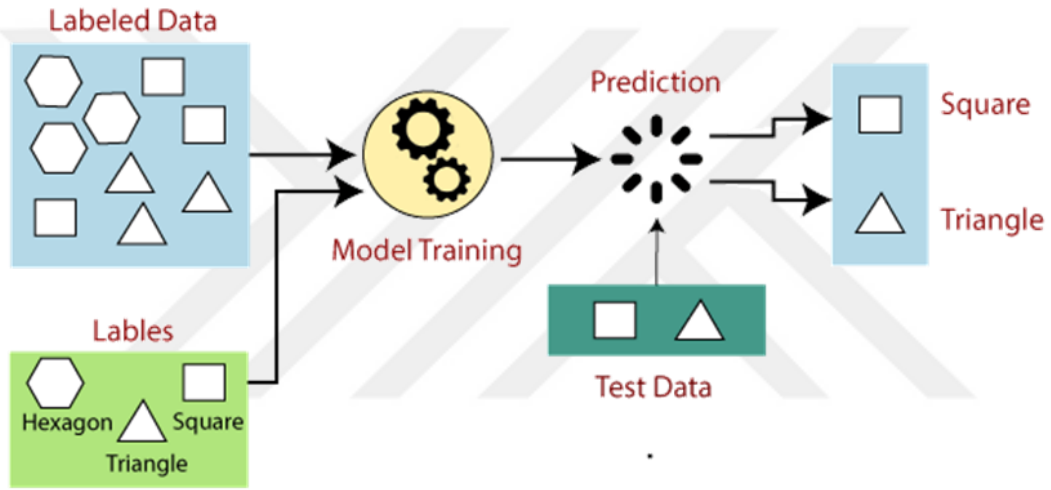


Şekil 6.1: Denetimli Öğrenme [39].

6.4.2 Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenme problemleri, verilerin hedef değerleri içermediği problemlerdir [40]. Bu tür problemlerde veri kümesi sadece özellikle değişkenlerini içerir ve denetimsiz öğrenme problemi verideki yapıyı veya dağılımı ortaya çıkaracak bir modele ihtiyaç duyar.

En yaygın denetimsiz öğrenme problemleri “Kümeleme” problemleridir. Kümeleme problemi, çözümlerinin müşterilerin satın alma davranışları gibi verilerdeki gruplamaları içermesi gereken bir problem türüdür. Şekil 6.2.’de kümelemeyi grafik olarak göstermektedir.



Şekil 6.2: Denetimsiz Öğrenme [41].

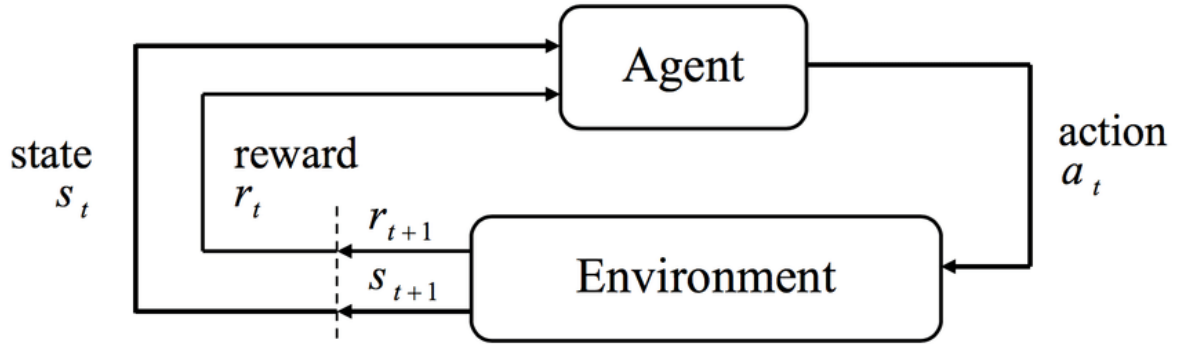
Boyutsallık azaltma ve anomali tespiti diğer denetimsiz öğrenme problemleridir.

Boyutsallık azaltma, bir dizi temel değişken elde ederek dikkate alınan rastgele değişkenlerin sayısını azaltma işlemidir [42].

Boyutsallık azaltma teknikleri özellik eleme işlemi için kullanılır. Anomali tespit teknikleri, veri örneklerinin çoğunu normal olduğu varsayımı altında, veri kümesinin geri kalanına uyan örnekler gözlemleyerek etiketsiz verilerdeki aykırı değerleri tespit eder [43].

6.4.3 Pekiştirmeli Öğrenme (Reinforcement Learning)

Pekiştirmeli öğrenmeyi açıklamadan önce yazılım ajanının ne olduğunu açıklamak faydalı olacaktır. Yazılım ajanı, bir algoritmanın verilen görevi tanımlanmış kurallar ve sınırlar içerisinde yapan bölümüdür. Takviyeli öğrenme, bir yazılım ajanının her hareketin sonunda bir ceza veya ödül verilerek belirli bir harekete veya hareket dizisine koşullandırılabilceği fikrinden doğar [44]. Başlangıçta, herhangi bir durumda yapılabilecek her harekete rastgele bir fayda değeri atanır. Rastgele bir başlangıç durumundan başlayarak, yazılım ajanın hareket etmeye başlar. Bir durumda, bir hamle yapıldığında ödül veya ceza alınır, o hamlenin fayda değeri bir formüle göre artırılır veya azaltılır. Şekil 6.3'te pekiştirmeli öğrenme prensiplerinin bir gösterimi verilmiştir.



Şekil 6.3: Pekiştirmeli Öğrenme [45].

6.4.4 Yarı Gözetimli Öğrenme (Semi- Supervised Larning)

Bu tür algoritmalar hem etiketli hem de etiketlenmemiş verileri kullanarak öğrenme işlemini gerçekleştirirler. Etiketli verilerin sınırlı olduğu durumlarda kullanılabilirler. Verilerin etiketlenme durumuna bağlı olarak bu öğrenme türü, aktif öğrenme, ortak eğitim, kendi kendine eğitim olmak üzere üç gruba ayrılmaktadır [46].

6.4.5 Takviyeli Öğrenme (Transfer Learning)

Bu tür algoritmalar, bir görevde öğrenilen bilgilerin, farklı ancak benzer bir görevde kullanılmasıdır. Önceden öğrenilmiş bir modelin, başka bir görevde hızlı ve daha verimli öğrenmeyi sağlamak için kullanılmasıdır. En yaygın kullanılanı ise Qlearning'dir [47].

6.5 MAKİNE ÖĞRENİMİ MODELLERİ

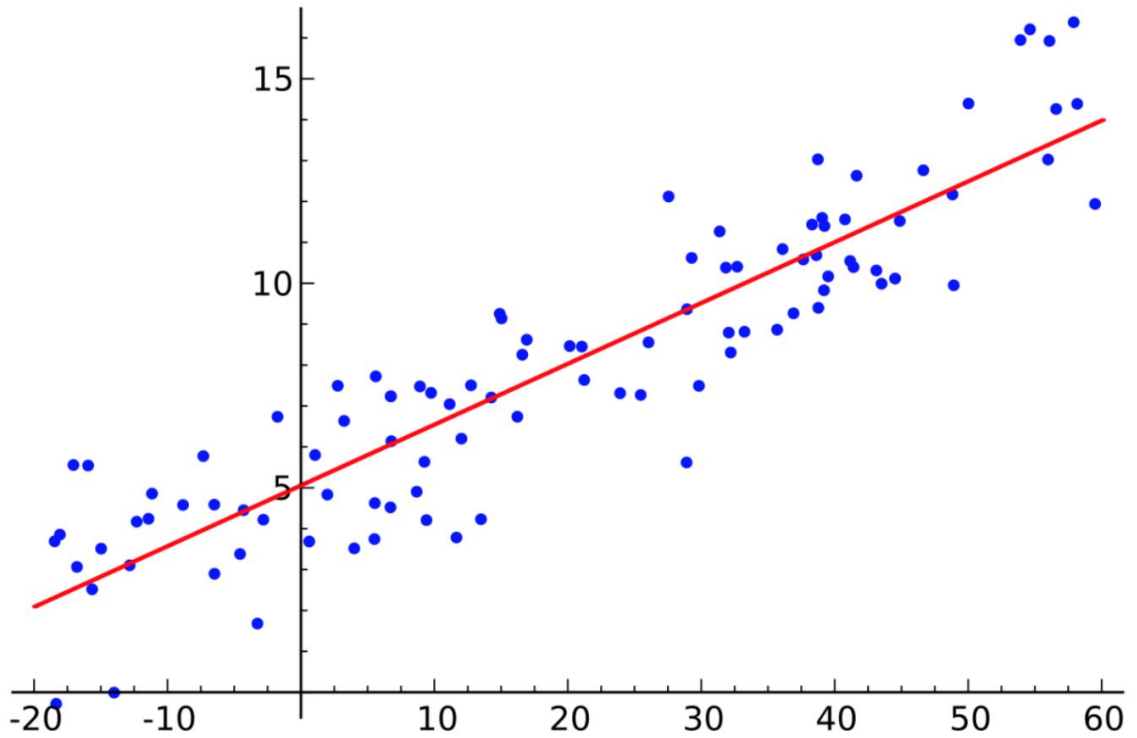
Bu bölümde en sık kullanılan makine öğrenmesi modelleri açıklanacaktır. Kullanım kolaylıkları, farklı veri kümelerine uygunlukları ve çalışma sürelerindeki hızları nedeniyle literatürde en sık kullanılan modellerdir [48].

Makine öğrenmesi algoritmaları, veri üzerinden modeller oluşturarak ve bu modelleri kullanarak çeşitli görevleri matematiksel teknikler ile yerine getirirler.

6.5.1 Doğrusal Regresyon

Doğrusal regresyon, girdi değişkeni (x) ile hedef değişken (y) arasında $y=A + B * x$ olarak adlandırılabilir. Doğrusal bir ilişki olduğunu varsayan doğrusal bir modeldir. [44]. Bu yöntemde, başlangıç A ve B değerleri modele atanır ve geri bildirim incelenir.

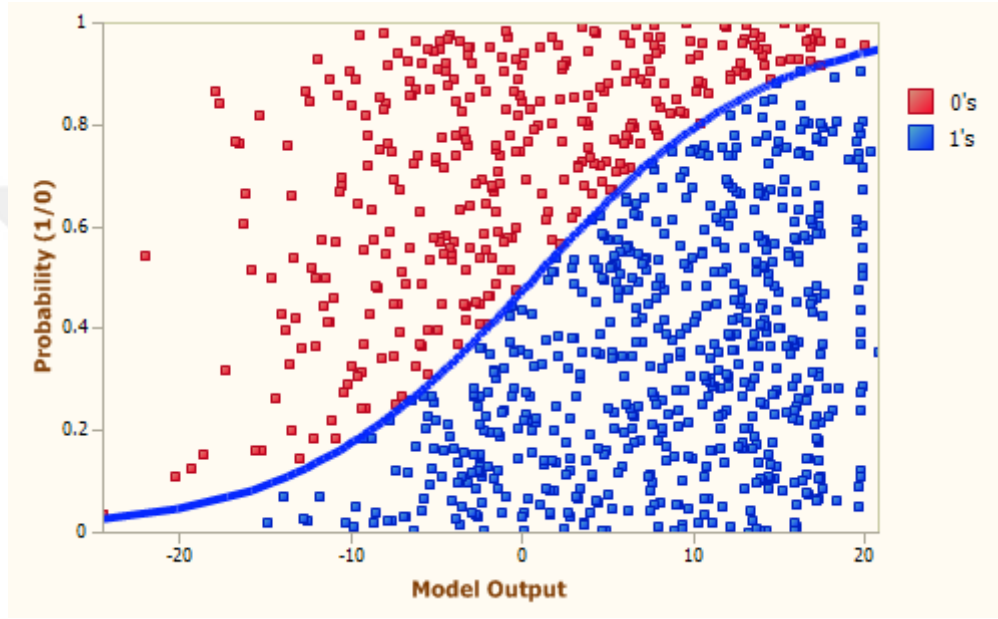
R, RMSE (Root Mean Squared Error), R2 vb. modelin hatasına göre A ve B değerleri güncellenmekte ve optimum iterasyonlarda en iyi doğruluk elde edilmeye çalışılmaktadır. Şekil 6.4'te doğrusal regresyon grafik olarak gösterilmiştir.



Şekil 6.4: Doğrusal Regresyon [49].

6.5.2 Lojistik Regresyon

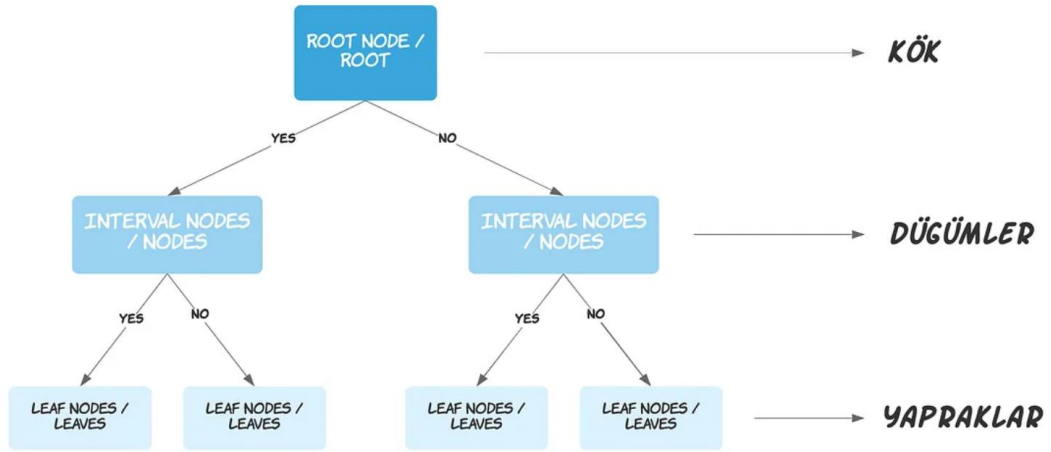
Lojistik regresyon, hedef deęişken iki kategoriye ayrıldığında tahmin yapmak için uygun olan yöntemdir [50]. Lojistik regresyon, verilere bir doğru uydurmak yerine 0'dan 1'e doğru giden "S" şeklinde bir lojistik fonksiyon uydurur. Veri noktaları hangi tarafa daha yakın olduklarını baęlı olarak deęerlendirilir ve tahminler bu perspektifte yapılır. Şekil 6.5'te lojistik regresyonun grafiksel gösterimi verilmiştir.



Şekil 6.5: Lojistik Regresyon [51].

6.5.3 Karar Ağacı

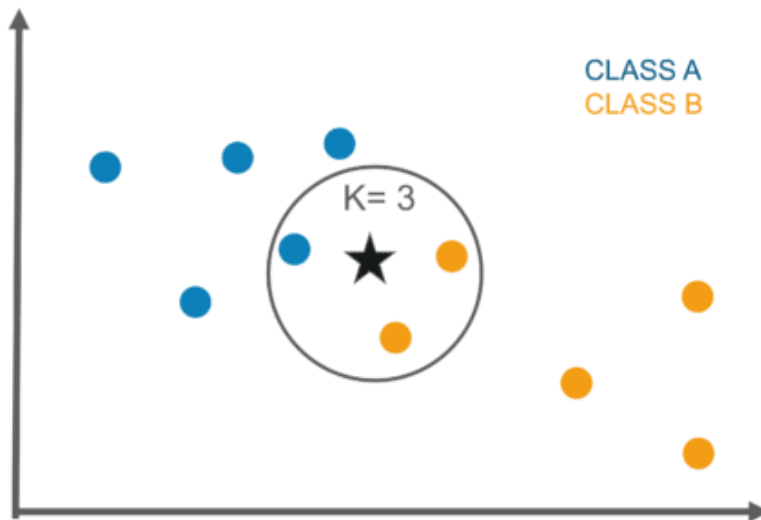
Bir karar ağacı, tüm olası çözümleri grafiksel olarak temsil etmek için ağaç benzeri bir grafik kullanır [52]. Karar ağaçları sadece makine öğrenimi uygulamalarında deęil, yöneylem araştırması ve karar analizinde de sıklıkla kullanılmaktadır. Bu yöntemde başlangıç kuralları ağacın yapraklarını oluşturur ve sırasıyla kurallar takip edilerek karar verilir. Kararların mevcut hatasından geri bildirim alındıktan sonra iterasyonlar yapılır ve kurallar üzerinde güncellemeler yapılır. Şekil 6.6'da örnek bir karar ağacının gösterimi verilmiştir.



Şekil 6.6: Karar Ağacı [53].

6.5.4 K En Yakın Komşu Algoritması

KNN olarak bilinen K En Yakın Komşu algoritması hem regresyon hem de sınıflandırma problemlerinde kullanılabilen bir tür denetimli makine öğrenmesi algoritmasıdır [54]. KNN yönetiminde ilk olarak k değeri bir tamsayı olarak atanır. Bu k değeri, k en yakın komşunun değerlerine bakılarak bilinmeyen veri noktasının sınıfını belirlemek için kullanılır. Bilinmeyen veri noktası, k komşusu arasında en sık görülen hedef değer sınıfı olarak tahmin edilir. K değeri tahmini değiştirebileceğinden, test verisi üzerinde ince ayar yapılmalıdır. Şekil 6.7’de k en yakın komşu algoritması farklı k değerleri ile grafiksel olarak gösterilmektedir.

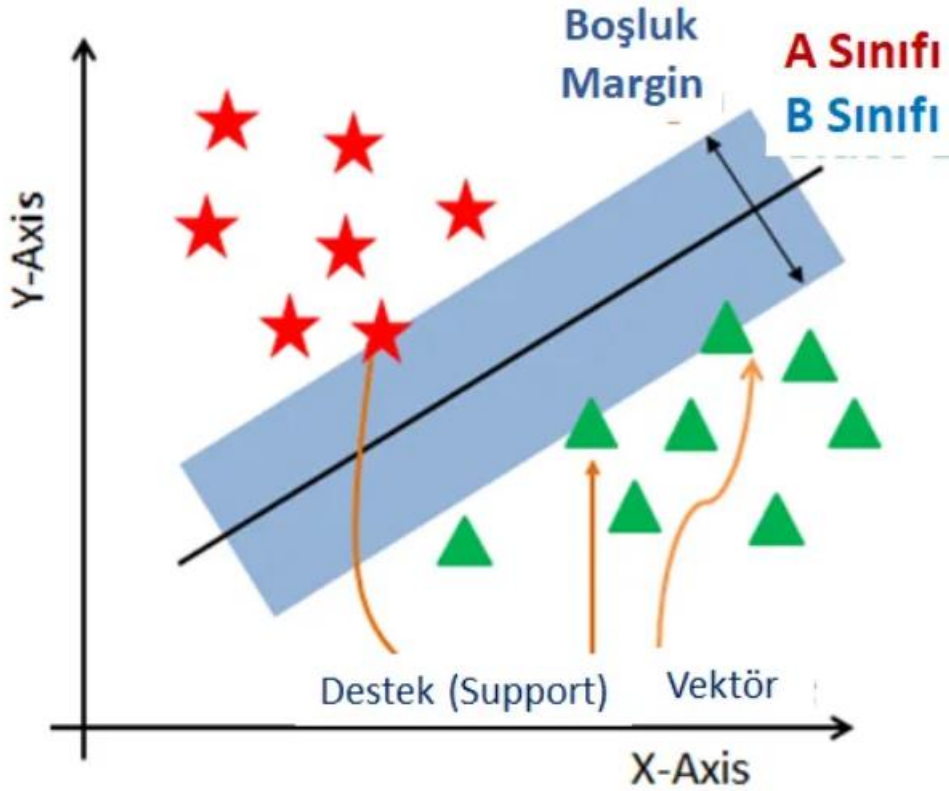


Şekil 6.7: K En Yakın Komşu Algoritması [55].

6.5.5 Destek Vektör Makinesi Algoritması

DVM olarak bilinen Destek Vektör Makinesi algoritmasının amacı, veri noktasını sınıflandırmak için N boyutlu uzayda bir düzlem bulmaktır [56]. Destek Vektör Makinesi yönteminin hem denetimli hem de denetimsiz makine öğrenme türü için problem uygulamaları vardır. Daha açık hale getirmek için Şekil 6.8’de 2 boyutlu denetimsiz öğrenme problemi için uygulama olarak incelenebilir.

Şekil 6.8’ de görüldüğü gibi bu problemin ilk adımda veri kümesi üzerine rastgele bir düzlem, bu durumda bir çizgi atanır. Bu düzlem veri noktalarını iki kategoriye ayırır. Bu veri gruplarının ağırlık merkezi her ikisi için de x ile işaretlenir ve bu iki ağırlık merkezine eşit uzaklıkta olacak şekilde yeni bir düzlem atanır, Şekil 6.8’de de görülebileceği gibi yeni bir düzlem veri noktalarını iki yeni gruba bölecektir ve bu yeni durum için işlem tekrarlanır ve bu da veri noktalarını ve düzlemi Şekil 6.8’deki duruma getirir.



Şekil 6.8: Destek Vektör Makinesi Algoritması [57].

6.5.6 Gradient Boosting Algoritması

Gradient Boosting algoritması, temel olarak karar ağaçlarına dayanan bir öğrenme yöntemidir. Karar ağaçlarının işleyişini anlamak, Gradient Boosting'in nasıl çalıştığını anlamak açısından önemlidir. Karar ağaçları, sınıflandırma ve regresyon tahminleri için yaygın olarak kullanılan bir yöntemdir. Bu yöntem, kolay anlaşılabilir ve yorumlanabilir olması açısından diğer birçok yönteme göre avantaj sağlar. Bir karar ağacı, kök noktasından başlar, verileri değerlendirerek aşağıya doğru dallanmalar oluşturur ve yaprak düğümleriyle sonlanır.

Gradient Boosting, zayıf öğrencileri güçlü öğrencilere dönüştürmek için gradyan artırma yöntemini kullanır [58]. Bu algoritmanın temel prensibi, her yeni oluşturulan karar ağacının, önceki ağaçta yapılan hataları en aza indirmeye yönelik olmasıdır. Algoritmanın başlangıcında bir karar ağacı ile bir tahmin elde edilir. Bu tahminle hedef değer arasındaki fark hesaplanır. Her bir iterasyonda, bu hesaplanan fark ile birlikte yeni bir ağaç oluşturulur. Sonuç olarak, algoritma hedef değer ile tahmin arasındaki farkı sıfırlamaya çalışır.

Gradient Boosting algoritması, belirli iterasyonlarla hata oranını en aza indirerek, yapay sinir ağlarına benzer bir şekilde çalışır.

Gradient Boosting'in yapay sinir ağlarına göre sunduğu önemli avantajlardan biri, açıklanabilirliğinin kolay olmasıdır. Yapay sinir ağları genellikle karmaşık ve anlaşılması zor modeller oluşturabilirken, Gradient Boosting ile oluşturulan modeller daha yorumlanabilir ve anlaşılabilir niteliktedir.



Şekil 6.9: Karar Ağacı Algoritmalarının Zaman İçindeki Gelişimi [59].

6.5.7 XGBoost Algoritması

Makine öğrenmesinde, denetimli öğrenme algoritmalarının sıkça kullanıldığı bir gerçektir. Bu algoritmaların özel bir alt kümesi olan "ensemble" yöntemleri, genellikle tahmin ve sınıflandırma problemlerinde yüksek performans sergiler. Bu yöntemlerden biri olan "Extreme Gradient Boosting" ya da yaygın olarak bilinen adıyla XGBoost, özellikle karar ağaçları üzerine çalışan bir gradient boosting algoritmasıdır ve genellikle yüksek verimlilik ve performansı ile tanınır.

XGBoost, ilk olarak 2016 yılında Tianqi Chen ve Carlos Guestrin tarafından yayınlanan yenilikçi bir makine öğrenmesi algoritmasıdır. Bu makale, veri bilimciler tarafından büyük bir ilgiyle karşılanmış ve Kaggle yarışmalarında kullanımı hızla artmıştır. XGBoost makalesine göre, 2015 yılında 29 Kaggle yarışmasının 17'si XGBoost ile kazanılmıştır (Chen ve Guestrin, 2016). XGBoost'un geliştirilmesi hala devam etmekte olup, Github'ta 500'den fazla katılımcısı ve 5400'den fazla kod eklemesi bulunmaktadır [60].

XGBoost, Gradient Boosting algoritmasının optimize edilmiş bir versiyonudur ve önceki versiyonlara göre sağladığı avantajlar, XGBoost'un popülerliğinin artmasındaki temel nedenlerden biridir. Bu algoritma, ağaç oluştururken maksimum derinlik değerini kullanır. Oluşturulan ağaç aşağı yönde aşırı ilerleme kaydederse, budama işlemi gerçekleştirilerek aşırı öğrenmenin önüne geçilir.

Gradient Boosting algoritması, kayıp fonksiyonun hesaplanmasında birinci dereceden fonksiyonları kullanırken, XGBoost bu hesaplamaları ikinci dereceden fonksiyonlar kullanarak gerçekleştirir.

Paralel çalışma özelliği, diğer algoritmalara kıyasla daha kısa sürede sonuca ulaşmayı sağlar. XGBoost'un sürekli gelişimi, veri bilimcileri tarafından halen büyük bir ilgiyle takip edilmektedir.

6.5.7.1 Gradient boosting'in temelleri

XGBoost'u anlamak için, önce gradient boosting'in temel prensiplerini kavramak önemlidir. Boosting, genellikle zayıf öğreniciler olarak bilinen bir dizi modeli birleştirerek güçlü bir öğrenme algoritması oluşturmayı amaçlayan bir "ensemble" yöntemidir.

Gradient boosting, bu algoritmanın bir uygulaması olup, adından da anlaşılacağı üzere gradyan iniş optimizasyon algoritmasını kullanır.

Bu algoritma iteratif bir şekilde çalışır ve her iterasyonda yeni bir model ekler. Yeni model, önceki modelin hatalarını düzeltmeye çabalar. Temel hedef, tüm modellerin toplamının gerçek çıktıya mümkün olduğunca yaklaşan bir tahmin oluşturmasını sağlamaktır. Bu süreç, loss function'ın gradyanı boyunca adım atarak gerçekleştirilir, bu da gradient boosting'in adını açıklar.

6.5.7.2 Xgboost ve diğer gradient boosting uygulamalarının karşılaştırılması

XGBoost, diğer gradient boosting algoritmalarından farklı olarak birkaç önemli özelliği ile öne çıkar. Bu farklar, genellikle algoritmanın hız ve performans üzerindeki odaklanmasından kaynaklanır.

İşte XGBoost'un bazı önemli özellikleri:

- a. Düzenleştirilmiş Objective Fonksiyonu: XGBoost, sadece modelin hatalarını azaltmaya odaklanan geleneksel gradient boosting'ten farklı olarak, modelin karmaşıklığını kontrol eden düzenleştirme terimlerini içerir. Bu, aşırı uydurmayı önler ve genelleme yeteneğini artırır.
- b. Hız ve Verimlilik: XGBoost, büyük veri setlerinde etkili bir şekilde çalışabilme yeteneği ile bilinir. Bu, kütüphanenin sütun blok yapısına ve önceden sıralama kullanarak belirli özellikleri arama yeteneğine dayanmaktadır.
- c. Esneklik: XGBoost, çeşitli objective fonksiyonları ve özelleştirilmiş değerlendirme kriterlerini belirleme yeteneği ile tanınır. Bu, kullanıcıların belirli bir problem için en uygun modeli oluşturmasına olanak tanır.
- d. Hedeflenen Öğrenme: XGBoost, out-of-core computing yeteneği sayesinde büyük veri setlerini işleme yeteneği sağlar. Bu, disk üzerindeki verileri işleme yeteneği anlamına gelir, bu da RAM sınırlamalarını aşarak çok büyük veri setlerini işleme yeteneği sağlar.

6.5.7.3 Xgboost algoritması

XGBoost'un temel çalışma prensibi, mevcut modelin hatalarını en aza indirgeyecek şekilde yeni bir model eklemektir.

Bu amacı gerçekleştirmek için, kayıp fonksiyonunun ikinci türeviden elde edilen bir adım boyutu hesaplanır. Bu adım, yeni modelin çıktısını, mevcut modelin hatalarını minimize edecek şekilde ayarlamaya yardımcı olur.

Daha teknik bir bakış açısıyla, XGBoost'un optimizasyon hedefi aşağıdaki formüle sahiptir:

$$L(y, \hat{y}) + \Omega(f) \quad (6.1)$$

Bu formülde, $L(y, \hat{y})$ modelin kaybını temsil eder (gerçek çıktı y ve tahmin edilen çıktı $\hat{y}$ arasındaki fark) ve $\Omega(f)$ modelin karmaşıklığını ifade eder. Karmaşıklık terimi genellikle ağacın yapısını (dallanma sayısı) ve yaprak düğümlerindeki çıktıların standart sapmasını içerir.

6.5.7.4 Xgboost'un uygulamaları

XGBoost, geniş bir uygulama yelpazede kullanılır. En yaygın kullanım alanlarından bazıları şunlardır:

Sınıflandırma: XGBoost, ikili ve çoklu sınıflandırma problemlerinde yaygın olarak kullanılır. Örneğin, hastaların belirli bir hastalığa sahip olup olmadığını belirlemek için kullanılabilir.

Regresyon: XGBoost, bir hedef değişkenin sürekli değerini tahmin etmek için de kullanılır. Örneğin, bir evin fiyatını tahmin etmek için kullanılabilir.

Ranking: XGBoost, öğelerin bir sıralamasını oluşturmak için de kullanılır. Örneğin, bir arama motoru, bir kullanıcının arama sorgusuna en alakalı web sayfalarını belirlemek için XGBoost'u kullanabilir.

XGBoost, yüksek performansı ve esnekliği nedeniyle makine öğrenmesi uygulamalarında sıklıkla tercih edilen bir araçtır. Gradient boosting'in güçlü bir örneği olan XGBoost, hızlı ve verimli hesaplama yeteneği, düzenlenleştirilmiş objective fonksiyonu ve esneklik sayesinde çeşitli pek çok sorunlar üzerinde çalışabilir.

6.5.8 Random Forest Algoritması

Rastgele Orman, genellikle büyük sonuçlar üreten, esnek, kullanımı basit ve verimli bir makine öğrenimi algoritması olan sınıflandırma ve regresyon ağaçlarını temel alan bir algoritmadır. RF algoritması, topluluk öğrenme algoritmaları sınıfına ait olan torbalama yöntemine dayanmaktadır.

Rastgele ormanın en önemli avantajlarından biri, hem sınıflandırma hem de regresyon problemlerinde çözüm sunabilmesidir, bu da diğer makine öğrenimi algoritmalarından ayrılan bir özelliktir. Rastgele orman, paralel olarak oluşturulan bir dizi karar ağacından oluşur, bu ağaçlar sınıflandırma ve regresyon ağaçlarından oluşabilir. Her bir karar ağacındaki düğümler, en uygun çözümü elde etmek için tüm özellikler içinde en iyi öznitelikleri kullanarak bölünür [61].

RF algoritması, büyük veri setleri içindeki gizli bilgileri çözebilmek için yaygın olarak kullanılan bir yöntem haline gelmiştir.

Rastgele bir ormanda, önce önyükleme yöntemiyle eğitim setleri oluşturulur. Ardından her eğitim seti için bir karar ağacı oluşturulur. Bu işlemler sonucunda algoritma tarafından elde edilen sonuç elde edilir. Random Forest algoritmasının temel kodu Şekil 1'de gösterilmiştir.

```
1. Ön koşul: Eğitim seti  $T: (x_1, y_1) \dots (x_n, y_n)$ , özellikler  $F$ , ormandaki ağaç sayısı :  $K$ 
2. fonksiyon Rastgele Orman ( $T, F$ )
3.  $P \leftarrow \emptyset$ 
4. for  $i \in 1, \dots, K$  do
5.  $T^{(i)} \leftarrow A$ ,  $T$ 'den önyükleme yapılır.
6.  $P_i \leftarrow \text{RastgeleAgacOgrenimi}(T^{(i)}, F)$ 
7.  $P \leftarrow P \cup \{P_i\}$ 
8. end for
9. return  $H$ 
10. end fonksiyon
11. fonksiyon RastgeleAgacOgrenimi( $T, F$ )
12. Her düğüm için:
13.  $f \leftarrow F$ 'nin en küçük alt kümesi
14.  $f$  en iyi özelliğe göre bölünür.
15. return öğrenme ağacı
16. end fonksiyon
```

Şekil 6.10: Random Forest Algoritmasının Kabaca Kodu [62].

RF (Rastgele Orman) algoritması çalıştırıldığında, ilk olarak eğitim seti (T), algoritma özellikleri (F), ve ormandaki ağaç sayısı (K) algoritma tarafından okunur. Başlangıçta, olasılık kümesi (P) boş bir küme olarak belirlenir. Ardından, eğitim seti için önyükleme işlemi gerçekleştirilir.

Algoritma döngüsü başladığında, rasgele ağaç öğrenimi için oluşturulan fonksiyon, okunan eğitim seti ve algoritma özelliklerine göre ağacı alt kümelerle bölerek işlemleri gerçekleştirir ve olasılıkları üretmeye başlar. Üretilen olasılıklar, olasılık kümesine aktararak döngü tamamlanır. Daha sonra elde edilen alt kümeler arasında en küçük olanı (f) bulunur. En küçük alt küme, en iyi özelliklere göre bölünür ve algoritmanın sonucu döndürülür.

Random forest, genellikle hiperparametre ayarlamaya gerek duymadan çoğu zaman mükemmel sonuçlar üreten esnek ve kullanımı kolay bir makine öğrenimi algoritmasıdır. Ayrıca, basitliği ve çeşitliliği nedeniyle sıklıkla kullanılan algoritmalar arasında yer almaktadır. (Sınıflandırma ve regresyon görevleri için kullanılabilir özelliği ile dikkat çeker.)

6.5.8.1 Random forest nedir?

Random forest, bir denetimli öğrenme algoritmasıdır. Oluşturduğu "orman", genellikle çantalamayla eğitilen bir dizi karar ağacından oluşan bir ensemble'dir.

Çantalamayla eğitim yönteminin temel fikri, öğrenme modellerinin birleşimini kullanarak genel sonucu artırmaktır.

Daha basit bir ifadeyle; random forest, birden fazla karar ağacı oluşturur ve bunları birleştirerek daha doğru ve istikrarlı tahminler elde eder.

Rastgele ormanlar, bir kombinasyon olarak çalışan ağaç tahmincileridir. Her bir ağaç, rastgele seçilmiş bir alt kümenin değerlerine bağlı olarak bağımsız olarak örneklenmiş vektörleri kullanarak öğrenir. Ormandaki tüm ağaçlar için genelleme hatası, ağaç sayısı arttıkça bir limite yaklaşır.

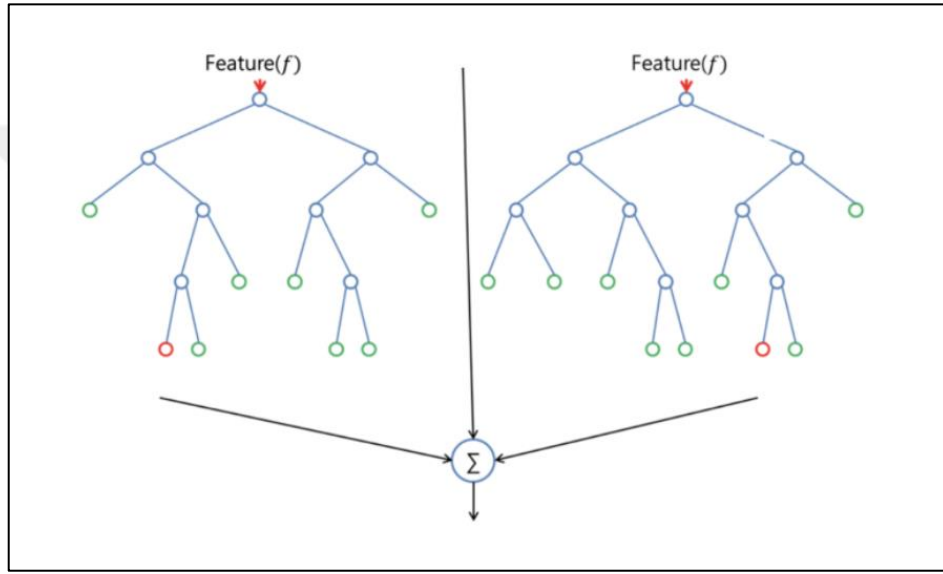
Ağaç sınıflandırıcılardan oluşan bir ormanın genelleme hatası, tek tek ağaçların gücüne, orman içindeki korelasyona ve kullanılan özellik sayısına bağlıdır. Her bir düğümü bölmek için rastgele özellik seçimi yapılır. Hata oranları, Adaboost ile karşılaştırılabilir (Freund ve Schapire [1996]), ancak daha güçlü ve gürültüye daha dirençlidir. Dahili tahminler, hatayı izlemek ve korelasyonu göstermek için kullanılır. Ayrıca, kullanılan özellik sayısını artırmak bölme işleminde de etkilidir. İç tahminler, değişken önemini ölçmek için kullanılır [63].

Bu fikirler aynı zamanda regresyon problemlerine de uygulanabilir.

6.5.8.2 Random forest nasıl çalışır?

Random forest'ün büyük bir avantajı hem sınıflandırma hem de regresyon problemleri için kullanılabilmesidir, ki bu, mevcut makine öğrenimi sistemlerinin büyük bir kısmını oluşturur.

Sınıflandırmada random forest'ü inceleyelim, çünkü sınıflandırma bazen makine öğreniminin temel taşı olarak kabul edilir. Şekil 6.11'de, iki ağaçla nasıl bir random forest modelinin görünebileceğini görebilirsiniz:



Şekil 6.11: Random Forest Algoritması [64].

6.5.8.3 Sınıflandırma ve regresyonda rastgele orman

Random forest, karar ağacı veya çanta sınıflandırıcısı gibi neredeyse aynı hiperparametrelere sahiptir. Neyse ki, bir karar ağacını bir çanta sınıflandırıcı ile birleştirmeye gerek yoktur çünkü random forest'ün sınıflandırıcı sınıfını kolayca kullanabilirsiniz. Random forest ile algoritmanın regresörünü kullanarak regresyon görevleriyle de başa çıkabilirsiniz.

Random forest, ağaçları büyütürken modele ekstra rastgelelik ekler. Bir düğümü bölme sırasında en önemli özelliği aramak yerine, rastgele bir özellik alt kümesi arasında en iyi özelliği arar. Bu genellikle daha iyi bir modelle sonuçlanan geniş bir çeşitlilik yaratır.

Bu nedenle, random forest sınıflandırıcısında bir düğümü bölmek için algoritma tarafından yalnızca özelliklerin rastgele bir alt kümesi dikkate alınır.

Ağaçları daha da rastgele hale getirebilirsiniz, her bir özellik için en iyi olası eşikleri aramak yerine ek olarak rastgele eşikler kullanarak (normal bir karar ağacı gibi).

6.5.8.4 Rastgele orman modelleri vs. karar ağaçları

Random forest modeli bir dizi karar ağacından oluşsada, bazı farklılıklar bulunmaktadır.

Bir karar ağacına özellikleri ve etiketleri içeren bir eğitim veri kümesini girdi olarak vererseniz, ağaç bazı kurallar kümesini formüle eder ve bu kurallar tahmin yapmak için kullanılır.

Örneğin, bir kişinin çevrimiçi reklamlara tıklayıp tıklamayacağını tahmin etmek için, kişi geçmişte tıkladığı reklamları ve kararlarını açıklayan bazı özellikleri toplarsınız. Özellikleri ve etiketleri bir karar ağacına koyarsanız, reklamın tıklanıp tıklanmayacağını tahmin etmeye yardımcı olan bazı kurallar oluşturacaktır. Buna karşılık, random forest algoritması rastgele gözlemleri ve özellikleri seçerek bir dizi karar ağacı oluşturur ve sonuçları ortalamalar.

Bir başka fark da "derin" karar ağaçlarının aşırı uyumdan muzdarip olabileceğidir. Çoğu zaman, random forest bu durumu, özelliklerin rastgele alt kümelerini oluşturarak ve bu alt kümeleri kullanarak daha küçük ağaçlar oluşturarak önler. Daha sonra bu alt ağaçları birleştirir. Bu her zaman işe yaramaz ve ayrıca random forest'un kaç ağaç oluşturduğuna bağlı olarak hesaplamayı daha yavaş hale getirir, bu da önemli bir noktadır.

6.5.8.5 Random forest hiperparametreleri

Random forest'taki hiperparametreler, modelin öngörü gücünü artırmak veya modeli daha hızlı hale getirmek için kullanılır. Sklearn'in yerleşik random forest fonksiyonunun hiperparametrelerine bir göz atalım.

6.5.8.6 Öngörü gücünü artırma

İlk olarak, `n_estimators` hiperparametresi vardır, bu sadece algoritmanın en yüksek oylamayı veya tahminlerin ortalamasını almadan önce oluşturduğu ağaç sayısını belirtir. Genel olarak, daha yüksek bir ağaç sayısı performansı artırır ve tahminleri daha kararlı hale getirir, ancak hesaplamayı da yavaşlatır.

Diğer önemli bir hiperparametre `max_features`'tir, bu random forest'un bir düğümü bölmek için düşündüğü maksimum özellik sayısını belirtir. Sklearn, belgede açıklanan birkaç seçenek sunar.

Son önemli hiperparametre `min_samples_leaf`'tir. Bu, bir iç düğümü bölmek için gereken minimum yaprak sayısını belirler.

6.5.8.7 Random forest modelinin hızını artırma

`N_jobs` hiperparametresi, motora kaç işlemci kullanmasına izin verildiğini söyler. Bir değeri bir ise, yalnızca bir işlemci kullanabilir. Değeri "-1" ise bir sınırlama olmadığı anlamına gelir.

`Random_state` hiperparametresi, modelin çıktısını çoğaltılabilir hale getirir. Model, `random_state`'in belirli bir değerine sahipse ve aynı hiperparametreler ve aynı eğitim verisi verilmişse her zaman aynı sonuçları üretecektir.

Son olarak, `oob_score` (ayrıca `oob` örnekleme olarak da adlandırılır), bir random forest çapraz doğrulama yöntemidir. Bu örneklemede, verilerin yaklaşık üçte biri modeli eğitmek için kullanılmaz ve performansını değerlendirmek için kullanılabilir. Bu örneklemelelere "out-of-bag" örneklemeleleri denir. Bu, `leave-one-out-cross-validation` yöntemine çok benzer, ancak neredeyse hiç ek hesaplama yükü bulunmaz.

6.5.8.8 Rastgele orman modelinin avantaj ve dezavantajları

Random forest'ün avantajları

Random forest'ün en büyük avantajlarından biri çok yönlülüğüdür. Hem regresyon hem de sınıflandırma görevleri için kullanılabilir, ayrıca;

- a. Çok yönlü kullanımlar
- b. Anlaşılması kolay hiperparametreler
- c. Sınıflandırıcı yeterli sayıda ağaçla aşırı uyum sağlamıyor olması da kolaydır.

Random forest ayrıca genellikle iyi bir tahmin sonucu üreten varsayılan hiperparametreleri kullanması nedeniyle çok kullanışlı bir algoritmadır. Hiperparametreleri anlamak oldukça açıktır ve bunların sayısı da çok fazla değildir.

Makine öğrenimindeki en büyük sorunlardan biri aşırı uydurmaktır, ancak çoğu zaman bu durum random forest sınıflandırıcısı sayesinde meydana gelmez. Ormanda yeterince ağaç varsa, sınıflandırıcı modeli aşırı uydurmaz.

Rastgele orman dezavantajları

Random forest'ün başlıca kısıtlamalarından biri, büyük bir ağaç sayısının algoritmayı gerçek zamanlı tahminler için çok yavaş ve etkisiz hale getirebilmesidir. Genel olarak, bu algoritmalar eğitim açısından hızlıdır, ancak eğitildikten sonra tahmin oluşturmak oldukça yavaştır. Daha kesin bir tahmin için daha fazla ağaç gereklidir ve bu da daha yavaş bir modelle sonuçlanır.

Çoğu gerçek dünya uygulamasında, random forest algoritması yeterince hızlıdır, ancak çalışma zamanı performansının önemli olduğu durumlar olabilir ve diğer yaklaşımlar tercih edilebilir.

- a. Artan doğruluk için daha fazla ağaç gerekir
- b. Daha fazla ağaç modeli yavaşlatıyor
- c. Veri içindeki ilişkileri tanımlayamaz

Ve tabii ki, random forest bir tahmin modelleme aracıdır ve bir açıklama arıyorsanız, verilerinizdeki ilişkilerin açıklanması için diğer yaklaşımlar daha iyi olacaktır.

7. YAPAY SİNİR AĞLARI

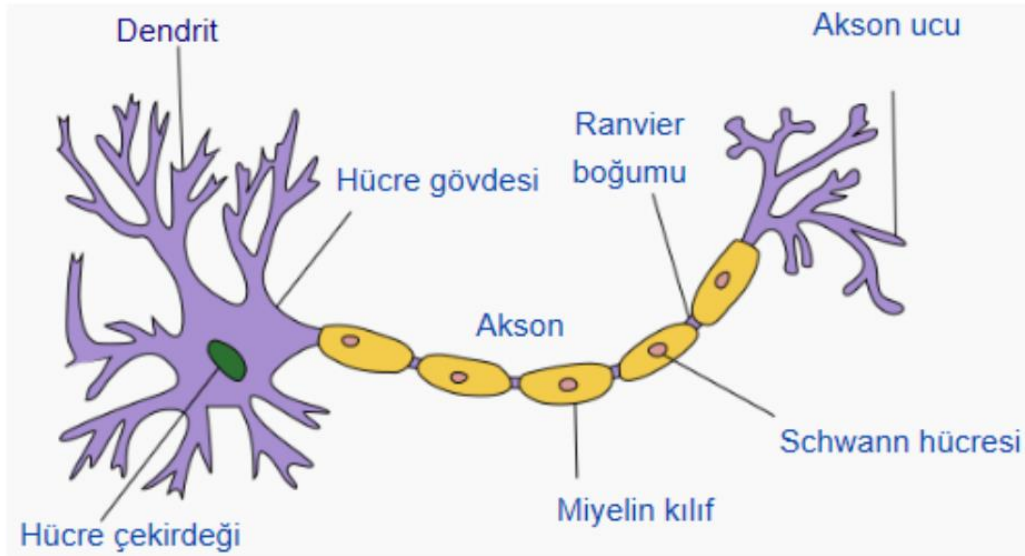
7.1 BİYOLOJİK SİNİR AĞI

Donald Hebb (1949), günümüz sinir ağı teorisinin babası olarak bilinir. Nörolog olan Hebb, beyninin nasıl öğrendiği üzerine çalışmalar yapmıştır. Çalışmalarında beynin temel birimi olan sinir hücrelerinin birbirleriyle nasıl bir korelasyon sergilediğini inceleyerek başlar ve sinir ağı teorisini bu temele yerleştirir. Yapay sinir ağı; sinir hücrelerinin sayısal dünyada gerçekleştirilecek yapının tüm fonksiyonlarıyla paralellik gösterdiği ve insan beyninin modellemesidir.

7.2 YAPAY SİNİR AĞI ÇALIŞMA ŞEKLİ

Yapay Sinir Ağları, insan beyninin çalışma tekniğinden esinlenerek üretilmiş bir kontrol tekniğidir. Yapay Sinir Ağları, biyolojik bir sinir sistemin çalışma şeklinin, matematiksel olarak modellenmesidir. İnsan beynini meydana getiren parçalar, sinir hücreleri olan nöronlardır.

Bu nöronlar farklı şekillerde birbirlerine bağlı olup bir araya geldiklerinde, sinir ağını meydana getirirler.



Şekil 7.1: Biyolojik Sinir Hücresi [65].

Bu sinir ağırları öğrenmeyi, hafızaya almayı ve hafızadaki bilgiler ile mukayese yapmayı gerçekleştirirler. Beyindeki öğrenme nöronlar arasındaki ilişkinin kimyasal bağı ile olmaktadır [66].

Şekil 7.1’de insan beynindeki sinir hücresi nöronun modeli verilmiştir.

Yapay Sinir Ağları, yapılan her çalışmada giriş çıkışları izleyerek, çalışmayla ilgili bilgiler toplar, çalışma hakkında genellemeler yapar. Daha sonra hiç görmediği bir durum ile karşılaşınca öğrendiği bilgileri kullanarak durum hakkında karar verir.

Bu çalışma özelliği, insan beyninin karar verebilme yeteneğini taklit etmektedir. Yapay Sinir Ağlarının kullanıldığı bazı uygulama alanlarına örnekler şunlardır;

- a. Veri madenciliği
- b. Görüntü işlemede karakter tanıma uygulamaları
- c. Pazar performansı tahmini
- d. Bankacılık müşteri analizi, risk analizi ve kredi kartı güvenliği hile kontrolü
- e. Robotlarda ve akıllı kontrollü araçlarda optimum yörünge belirleme
- f. Otomatik pilot uygulamaları
- g. Konuşma, parmak izi ve iris tanıma
- h. İletişimde gürültü filtreleme
- i. Haberleşmede trafik yoğunluğunun kontrol etme ve anahtarlama
- j. Savunma uygulamalarında, hedef seçme ve silah yönlendirme
- k. Hastalık teşhisi ve erken tanı uygulamaları (Röntgen, MR, ultrason)

7.3 YAPAY SİNİR AĞI YAPISI

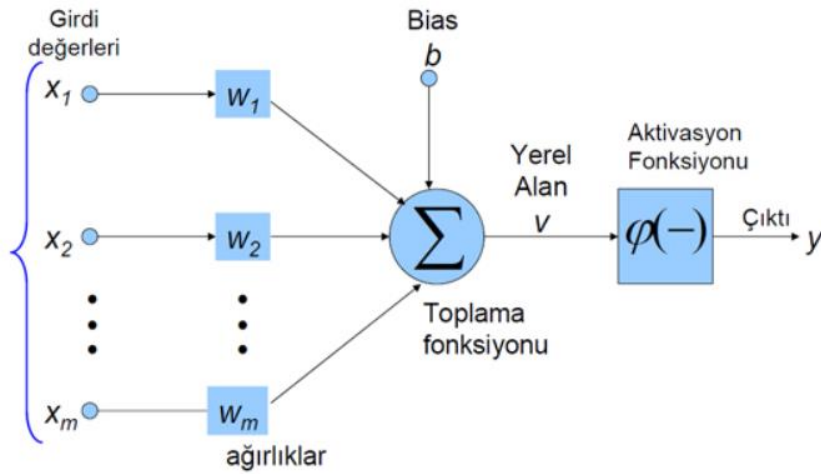
Yapay sinir ağları, beyindeki sinir ağlarına benzetilerek üretilmişlerdir. Tabanda eşsiz yapay sinir ağı hücresi, insan beyindeki sinir hücresine göre hayli zahmetsiz bünye mülkiyetindedir. Temel olarak, bir yapay sinir ağı dokusu mukaddimeler, değerler, total işlevi, aktivasyon (transfer) işlevi ve paylamalardan oluşurlar.

Dirimsel sinir sistemini yapay sinir ağına benzetirken kullanılan ifadeler Çizelge 7.1’de verilmiştir.

Çizelge 7.1: Dirimsel Sinir Sisteminde Yapay Sinir Ağları Benzetim Karşılıkları.

Dirimsel Sinir Sistemi	Yapay Sinir Ağları
Nöron	İşlemci eleman
Dendrit	Toplama fonksiyonu
Hücre Gövdesi	Transfer fonksiyonu
Aksonlar	Yapay nöron çıkışı
Sinaps’lar	Ağırlıklar

Yapay sinir ağlarının belkemiğini meydana getiren eşsiz sinir ağı tasarımı, matematiksel olarak aşağıda Şekil 7.2’de gösterilmiştir.



Şekil 7.2: Yapay Sinir Ağı Modelinin Yapay Sinir Ağı Tasarımının Matematiksel Gösterimi [67].

Şekil 7.2’de teşkili, $x_1, x_2, \dots, x_n$ mukaddime bilgileri; $w_{k1}, w_{k2}, \dots, w_{kn}$ k nöronunun sinaptik değerleridir. u_k mukaddime bilgilerine özgü sahici olarak toplanmış paylamadır.

Yakındaki bias, $(.)$ aktivasyon fonksiyonu ve y_k sinir ağının paylama işaretidir. Yakındaki bias b_k doğrusal toplayıcısı u_k paylama tahmini etki yaratır. Matematiksel olarak eşsiz nöron (k) münhaldeki denkliklerde tasvir edilebilir.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (7.1)$$

$$y_k = \varphi (u_k + b_k) \quad (7.2)$$

$$v_k = u_k + b_k \quad (7.3)$$

$$y_k = \varphi (v_k) \quad (7.4)$$

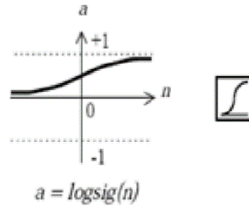
Yapay sinir mensuplarından ele alan mukaddimeler sinir ağı önemindeki hükümünü saptayan kararınca katsayılar, ağırlıklar denir. Enikonu eşsiz girişin kişisel olan ağırlığı vardır. Eşsiz değer miktarının yüksek olması, önemli olması ya da işte mukaddimenin yapay sinire dinçlik sağlaması, önemli olmaması veya küçük olması da zabun olması anlamını gelmektedir.

Şekil 7.3’de toplama fonksiyonu ile enikonu eşsiz değer in kişisel olan mukaddimelerle çarpımının totallerini (aktivasyon potansiyeli), ön değer ile birleşerek aktivasyon fonksiyonuna iletilir.

Aktivasyon fonksiyonu, aktivasyon gücünü paylama belirtecini arzulanan çizgide çizecek biçimde tabiileştiren fonksiyondur.

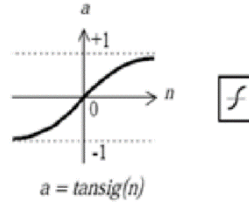
Böylelikle, süre önemli olduğunda aktivasyon gücünü paylaması değişmesini sağlamaktadır.

Genel olarak iyelik aktivasyon fonksiyonları, logaritmik sigmoid, tanjant sigmoid ve esas fonksiyonlardır. İşte fonksiyonların resimleri Şekil 7.5, Şekil 7.6 ve Şekil 7.7’de, matematiksel denklemleri izleyen denklemlerde verilmiştir.



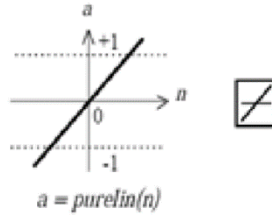
Şekil 7.3: Yapay Sinir Ağı Aktivasyon Fonksiyonu(logsig(n)).

$$a = \text{logsig}(n) = \frac{1}{1 + e^{-n}} \quad (7.5)$$



Şekil 7.4: Yapay Sinir Ağı Aktivasyon Fonksiyonu(tansig(n)).

$$a = \text{tansig}(n) = \frac{1 - e^{-2n}}{1 + e^{2n}} \quad (7.6)$$



Şekil 7.5: Yapay Sinir Ağı Aktivasyon Fonksiyonu(purelin(n)).

$$a = \text{purelin } n = n \quad (7.7)$$

Sinir hücresi, aktivasyon fonksiyonunun başlangıç aşamasının sınırı geçemediğinde paylama üretmez. Başkaca, sinir ağına eşsiz ön değer denenerek aktivasyon fonksiyonu ilave edilebilir. [68].

Sinir Ağı'nın çıkışları, aktivasyon fonksiyonunun sonucunun diğer sinirlere ve dış dünyaya gönderildiği bölümdür. Eşsiz sinirin eşsiz paylaması mevcuttur. Sinirin belirtilen paylaması, kişinin üzerine gelen gelişigüzel numaradaki öbür sinirlere mukaddimede de olabilir.

7.4 YAPAY SİNİR AĞLARININ SINIFLANDIRILMASI

7.4.1 Yapılarına Göre Yapay Sinir Ağları

Yapay sinir ağları koruduğu nöronların karşılıklı ilinti modellerine özgü ileri ve geri büyütme olmak üzere iki başlığa ayrılır.

- a. İleri Büyütmeli Ağlar: İleri büyütmeli ağlarda nöronlar mukaddimeden paylamaya yönelik tertipli katmanlar formundadır.
- b. Eşsiz katmandan tekçe ondan sonra gelen katman ile bağlantılıdır. Yapay sinir ağına uğrayan haberler mukaddime katmanına akabinde düzenle geçit katmanından ve mukaddime katmanında işlem görerek ilerler akabinde takiben hariç cihana iletilir.
- c. Geri Büyütmeli Ağlar: Geri büyütmeli yapay sinir ağlarında ileri büyütmeli bütün aksine tek dokunun sonucu arkasızca ondan sonraki yaklaşan dokunun katmanına gider olmak üzere iletilemez.

Kendinden önceki katman veya kendi katmanında varlığı süren gelişigüzel tek dokuya da gider olmak üzere birleşebilmektedir. Bu yapıyla ileri büyütmeli yapay sinir ağlarında uzay modelinde bulunmayan canlı modelde tek eyleme nezaret etmektedir.

Geri büyütme özgünlüğünü sağlayan bağlantıların bağlanma biçimine göre geri arka özdeş yapay sinir ağıyla özdeş olmayan tutumda ve yapıda geri büyütmeli yapay sinir ağları edinilebilir [69].

7.4.2 Öğrenme Zamanına Göre Yapay Sinir Ağları

Yapay sinir ağları bilgi edinme süresi gereğince durağan bir de canlı bilgi edinmeyi gerçekleştirmek çift alt başlığa bölünür [70].

- a. Durağan Bilgi Edinme: Durağan bilgi edinme prensibiyle emek veren yapay sinir ağlarından faydalanılmadan önce tahsil uygulaması ile karşılaşırlar.

Tahsilini bitirmesinin akabinninde ağ arzulanan modelde kullanıma tabi olabilirler. Lakin iyelik esnasında ağın yüzeyinde aksi bir düzenle karşılaşılmamaktadır.

- b. Canlı Bilgi Edinme: Canlı bilgi edinme prensini ise yapay sinir ağlarının çaba harcadığı zaman içinde bilgi edinmesi öngörülerek modellenmiştir.

Yapay sinir eğitimi etabını başardıktan biraz sonra ardından gelecek iyeliklerinden paylanan onaylamanın sonucunda kütlelerini ayırarak çaba harcamayı sürdürmektedir [71].

7.4.3 Öğrenme Zamanına Göre Yapay Sinir Ağları

Yapay sinir ağlarının elde edilen çıktı karşıtlarına göre sonuç temin edebilmenin usulü ağın bilgi edinmesidir.

Bilgi edinme uygulamasının tam yöntemi bulunmaktadır. Yapay sinir ağları bilgi edinme algoritmalarına göre bilirkişili, bilirkişisiz ve dayanaklı bilgi edinme olmak koşuluyla 3 gruba paylaştırılır [72].

Bilirkişili Bilgi Edinme: Bilirkişili bilgi edinmede bir muallime gereksinim duyulur. Muallim, tek bilgi talim ekibi veya ağın başarını oranlayan bir takipçi durumu olasıdır. Bilirkişiki bilgi edinmede, arzulanan ağ paylamasının üretilebilmesi için, paylama yanlışlığının düşürülmesinde kütlelerin uyması için çalışma gerekebilir.

Buna göre her mukaddime pahası için arzulanan paylama düzenine sokulur ve yapay sinir ağının mukaddime-paylama ilintisini meydana getirene denk gittikçe düzenlenir.

Bilirkişili bilgi edinmeye çok katmanlı perceptron, back-propagation, delta usulü, Widrow-Hoff yahut küçücük dörtgenlerin avarajı ve değiştirilebilir uzaysal görevli manaya gelen Adaline misal örnek gösterilebilir [73].

Bilirkişili bilgi edinme sırasında ağa verilen mukaddime verileri için paylama verileri de verilir. Ağ iletilen mukaddimeler için arzulanan paylamaları ortaya çıkarmak için kendi kütlelerini update eder.

Ağın paylamalarıyla beraber istenen paylamalar ortasındaki yanlışlığı oranlama yaparak taze kütleleri paylamaya hazır hale getirirler.

Yanılğı hissesi oranlanırken ağın istenen paylamaları ile ağın tamamı olan paylamalar ortasındaki ayırtı oranlanır ve bu başkalığa müsait olan her bir dokuya denk gelen yanlışlığı hissesi oranlanır. Akabininde her bir doku kendine gelen kütleleri update eder [74].

Bilirkişisiz Bilgi Edinme: Bilirkişisiz bilgi edinmede muallim yahut bilirkişi, mukaddime ağın hangi yönde makbul akıbet vereceğini yahut bilgi bölümünü topluluğa katıldığını belirtmez.

Burada dışarıdan bir dış güç olamaz. Ağ veriyi müdavimleri benzer olan yığınlara ışık vermeden seçer. Bilirkişisiz bilgi edinme bilirkişili bilgi edinmeye oranla çok ivediklidir.

Ayrıca matematik algoritmaları da çok basittir. Bilirkişisiz bilgi edinmeye oyuncu bilgi edinme, Hebbian bilgi edinme, Kohonen'in Grossberg bilgi edinme, öz düzenlemeli krokiler vb. gibi bilgi edinme prensipleri misal olarak verilebilir [75].

Bilirkişisiz bilgi edinmede ağa öğrenme boyunca ancak misal mukaddimeler gösterilebilir. İstenen paylama datası gösterilemez. Mukaddimede verilen datalara göre ağ birden fazla misali sınıflarında sınıflayacak modelde kendi prensiplerini ortaya çıkarır. Ağ ilintili kütlelerini aynı özellikte olan hücrelere parçalayacak tasarıda sıraya koyan öğrenme metodunu yerine getirir.

Dayanklı Bilgi Edinme: Bu bilgi edinme tutumunda ağın her epok akıbetinde ortaya çıkardığı akıbeti faydalı yahut faydasız olup olmadıklarına dair aktarımlarda bulunur.

Ağ bu aktarımlar için kendini üsteleyerek sıraya koyar. Sonuçta ağ gelişigüzel biçimde mukaddime edilmiş sırada akıbet edinerek hem de öğrenerek kazanç sağlamayı sürdürür.

Yapay Sinir Ağlarında kullanılan eğitim metotları Çizelge 7.2'de gösterilmektedir.

Çizelge 7.2: Yapay Sinir Ağlarında Kullanılan Eğitim Metotları.

Fonksiyon	Algortıma
trainlm	Levenberg-Marquardt
trainbr	Bayesian Regularization
trainbfg	BFGS Quasi-Newton
trainrp	Resilient Backpropagation
trainseg	Scaled Conjugate Gradient
traincgb	Conjugate Gradient with Powell / Beale Restarts

Çizelge 7.2: Yapay Sinir Ağlarında Kullanılan Eğitim Metotları (Devamı).

traincgf	Fletcher-Powell Conjugate Gradient
traincgp	Polak-Ribière Conjugate Gradient
trainoss	One Step Secant
traingdx	Variable Learning Rate Gradient Descent
traingdm	Gradient Descent with Momentum
traingd	Gradient Descent

7.5 YAPAY SİNİR AĞLARININ EĞİTİLMESİ VE ÖĞRENMESİ

İnsanoğlunun hayatağacı, dünyaya gelişinin akabinde yetiştiği süre içinde konumundan dolayı duyum uzuvlarıyla anladığı eylemleri tefsir eder ve bu bilinçleri başka eylemlerinde de uygulamaya çalışır. Hayatı boyunca hayatağacı yükselir ve değişikliğe uğrar.

Bundan dolayı eylemlerde ne karşılık işleyeceği genellikle bu sürede bilinmez. Halbuki hiç rastlamadığı bir eylem durumunda yine tecrübesiz eylemler sergileyebilir.

Yapay sinir ağlarının bilgi edinme süre içinde de hariç toplumlardan mukaddimeler ele alınır, aktivasyon metotlarından geçilerek bir değişikliğe paylama oluşturabilir. Bu paylama tecrübesiyle verilen paylama ile karşı karşıya kaldığında yanılğı sonucuna varılır [76].

Değişik bilgi edinme algoritmalarıyla yanılğıya uğrayan gerçek paylamalara yaklaştırmaya çaba harcar.

Bu uygulama vaktinde tazelenen yapay sinir ağının kütleleridir. Kütleler her bir çevrimde (cycle) tazeleyerek gayeye varmaya çaba harcanır. Gayeye varmanın yahut yavaşmanın oranı da gene dıştan verilecek meziyettir.

Lakin yapay sinir ağı mukaddime-paylama çiftiyle gayeye varmanın ise kütle meziyet durumudur [77].

Belirli bir uygulamaya dönük konfigüre edilen bir sinir ağı çalışılmadan önce eğitilmelidir.

Bu dönemde, deneyim sayesinde bilgi edinme vasfı için eleştirel meziyeti durumunu göstermektedir. Tümelde, açılış kütleleri gelişigüzel belirlenir ve eğitime ya da öbür bir deyimle öğrenme metodu bu metotlarının akabinde çalışılır [78].

Yapay sinir ağlarında uygulama parçalarının ilintilerinin kütle oranlarının saptanması uygulamasına ağın eğitilmesi denmektedir. Başlayışta kütle oranının rassal olarak belirtilir. Yapay sinir ağları kendilerine benzerleri atadıkça bu kütle oranları değişime uğrar. Gaye ağa verilen benzerler için asıl mukaddimleri meydana getirecek kütle oranlarını oranlarlar. Benzerleri ağa süreli olarak atandıkça en orijinal kütle oranlarını bulmak hedef gösterilir. Ağın asıl kütle oranlarına ulaşması asıllarını damgaladığı eyleme dair tamim yapabilme yeteneğine erişmesini gösterir. Bu tamim yeteneğine erişme uygulamasına ağın bilgi edinmesi denmektedir. Kütlelerin oranlarının farklılaşması belirli kurallara göre kullanılır. Bu prensiplere bilgi edinme prensipleri denmektedir [79].

Kütleleri sürekli tazeleyerek talep eden sonuçlara ulaşılan kadar geçen zamana öğrenme adı verilir. Yapay sinir ağı öğrendiğinin hemen akabinde öncelikle belirtilmeyen mukaddimleri işaret edip, sinir ağı paylamasıyla asıl paylama tutumunu sergiler.

Şayet taze gösterilen benzerlere de asıl olarak yanaşıyorsa sinir ağı bu sorunu öğrenmiş ifadesine gelir. Sinir ağına gösterilen benzerlerin içinden kararınca olan oranlardan daha fazlası ise sinir ağı bu sorunun öğrenmemiş, gelişigüzel işliyor ifadesine benzetilir. Tümelce var olan benzerlerinin %80'i ağa iletilip ağ eğitilir. Akabinde arda kalan %20'lik parça ağa belirtilerek ağın vaziyeti incelenir. Bu iyilikte ağın incelemesi uygulanmış olmaktadır [80].

a. Örneklerin Toplanması: Örneklerin Toplanması: Ağın öğrenmesini arzu eden eylemler için başlangıçta kazanılan benzerlerin aranması sürecidir.

Ağın eğitilmesi için benzerlerin ele alındığı gibi (eğitim bölümü) ağın uygulama yapılması içinde benzerlerinin (test bölümü) ele alınması elzemdir. Eğitim bölümündeki benzerler sırasıyla ağa iletilerek ağın eylemi öğrenmesi amaçlanır. Ağ eylemi öğrendiğinin hemen akabinde test bölümündeki benzerler ağa iletilerek ağın başarısı denir. Şimdiye kadar hiç ağa iletilmeyen benzerler karşılığında başarının ağın başarılı biçimde öğrenip öğrenmediği sunulur.

- b. Ağın Topolojik Yapısının Belirlenmesi: Öğrenilmesi arzulanan eylemler için meydana getirilecek olan ağın topolojik durumu saptanır. Kaç parça mukaddime bölümü, kaç parça orta katman, her orta katmanda kaç adet uygulama ögesi, kaç adet paylama ögesinin gerçekleşmesinin istenmesi bu aşamada saptanır.
- c. Öğrenme Parametrelerinin Belirlenmesi: Ağın öğrenme miyarı, uygulama ögesinin aktivasyon ve tutar fonksiyonları, momentum miyarı gibi değerler bu aşamada saptanır.
- d. Ağırlıkların Başlangıç Değerlerinin Atanması: Uygulama ögelerini ilintileyen kütle oranlarının ve başlangıç ağırlık bölümünün kütlelerinin başlangıç kütlelerinin seçilmesi meydana getirilir. Başlayışta tümel olan rassal oranlarını tayin edilir. Akabinde ağ müsait oranları öğrenme sırasında saptar.
- e. Öğrenme Setinden Örneklerin Seçilmesi ve Ağa Gösterilmesi: Ağın öğrenme başlangıcında ve öğrenme prensip uygunluğu olarak kütleleri farklılaştırması için ağa varolan yapıya göre iletilir.
- f. Öğrenme Sırasında İleri Hesaplamaların Yapılması: İletilen mukaddimeler ağın paylama oranları olarak oranlanır.
- g. Gerçekleşen Çıktının Beklenen Çıktı İle Karşılaştırılması: Ağın meydana getirdiği yanılğı oranı bu adımda oranlanır.
- h. Ağırlıkların Değiştirilmesi: Geçmiş oranlama metotları göz önüne alınarak yanılğının indirilmesi için kütleler farklılaştırılabilir.
- i. Öğrenmenin Tamamlanması İleri büyütmeli sinir ağı öğrenmeyi tüketinceye denk, meydana gelen paylamalar ile arzulanan paylamalar boşluğundaki yanılğılar müsait kademelere getirilinceye kadar uygulama sürmektedir [81].

Yapay sinir ağları elastik ve parametrik olmayan bir tasarlama ögesidir [82]. Yapay sinir ağları, hayatağacının bilişsel bilgi edinme zamanının benzeri ile yol alan bir metottur.

Karma sorunlarda oldukça başarılı olduğundan öngörme, bölümlendirme, gruplama gibi birden fazla soruna pratik sonuçlar verebilir. Sinir ağlarının değerli ayrımı, karma düzenlere ait geride kalan derslerden yol alarak, var olanları önemseyerek öğrenme yordamıyla sorunun halletme kabiliyetidir [83].

Geleneksel metodlar, hatalı çıktıların meydana gelmesi zararı gerekçesiyle noksan yahut ekstrem durumlarda bilgiler için hesaplı değildir [84]. Yapay sinir ağları tutumu ise, bilgilere tutunmadan, noksan, eksiği ola ya da ekstrem durumlarda bilgileri bile göz önüne alabilir. Ve de karma birleşimleri öğrenebilir, tamimleyebilir ve bu sayede hiç tanımadığı çıkmazlara, uygunlukta bir yanılıyla yanıtlayabilirler [85].

Bu seçkin sıfatlardan dolayı yaşadığımız çağda epey mühendislik icraatlarında ve istek öngörülerinde yapay sinir ağları tasarımları kapsamlı olarak kullanılırlar.



8. UYGULAMA VE SONUÇLAR

Tahmin modelinin doğruluğu, doğru algoritmanın seçilmesine ve verilerin araştırılması ve anlaşılmasına bağlıdır. Bu iki süreç birbirine bağlıdır. Belirli veri yapıları ile iyi çalışan belirli algoritmalar bulunmaktadır. Ayrıca, her makine öğrenimi algoritması belirli veri yapılarına ve veri hazırlama süreçlerine ihtiyaç duyar.

a. Problem Açıklaması:

Kredi Kartı Dolandırıcılığı Tespit Problemi, geçmiş kredi kartı işlemlerinin, dolandırıcılık olduğu ortaya çıkanların bilgisi ile modellenmesini içerir. Bu model daha sonra yeni bir işlemin hileli olup olmadığını belirlemek için kullanılır. Buradaki amacımız, yanlış dolandırıcılık sınıflandırmalarını en aza indirirken dolandırıcılık işlemlerinin %100'ünü tespit etmektir.

b. Sınıf Dengesizliği Model Performansını Neden Etkiler?

Genel olarak, FPR'yi (Yanlış Pozitif Oranı) sınırlarken geri çağırma en üst düzeye çıkarmak isteriz, ancak çok sayıda yükü yanlış sınıflandırabilir ve yine de düşük bir FPR'yi koruyabilirsiniz çünkü çok sayıda gerçek negatifiniz vardır.

Bu, nispeten düşük bir eşik seçmeye elverişlidir, bu da yüksek geri çağırma ancak son derece düşük hassasiyetle sonuçlanır.

c. Sorun Nedir?

Bir modeli dengeli bir veri kümesi üzerinde eğitmek, doğrulama verileri üzerindeki performansı optimize eder. Ancak amaç, dengesiz üretim veri kümesi üzerindeki performansı optimize etmektir. Nihayetinde üretimde en iyi şekilde çalışan bir denge bulmanız gerekir.

Bu sorunun bir çözümü şudur: Tüm hileli işlemleri kullanın, ancak hedef oranımıza ulaşmak için hileli olmayan işlemleri gerektiği kadar alt örnekleyin.

8.1 DOLANDIRICILIK İŞLEMLERİ BELİRLİ BİR ZAMAN DİLİMİNDE DAHA SIK MI GERÇEKLEŞİYOR?

Dağılımları görmek, bu özelliklerin ne kadar çarpık olduğu konusunda bir fikir edinmemize yardımcı olabilir, ayrıca diğer özelliklerin de daha fazla dağılımını görebiliriz. Bu dağılımları daha az çarpık hale getirebilecek tekniklerin gelecekte bu defterde uygulanacağı görünüyor. Yukarıdaki gözleme göre, işlemin gerçekten zamanın önemli olmadığı görünmüyor. Şimdi modelleme ve tahmin için veri kümesinin bir örneğini alalım.

Bu korelasyonlar yüksek olsa da, çoklu doğrusallık riskini beklemiyorum.

Korelasyon matrisi ayrıca V1'den V28'e PCA bileşenlerinin birbirleriyle hiçbir korelasyonun olmadığını gösteriyor, ancak Class'ın V bileşenleri ile bazı pozitif ve negatif korelasyonlar olduğunu gözlemleyebiliriz, ancak Zaman ve Miktar ile hiçbir korelasyonu yok.

- a. Sahtekarlık işlem ağırlığı: 0.0017994745785028623
- b. Sahtekarlık olmayan işlem ağırlığı: 0.9982005254214972
- c. EĞİTİM: X_train: (159491, 30), y_train: (159491,)
- d. DOĞRULAMA: X_validate: (39873, 30), y_validate: (39873,)
- e. TEST: X_test: (85443, 30), y_test: (85443,)

a. Gözlemler

Çok az sayıda işlem gerçekten hilelidir (%1'den az). Veri seti oldukça çarpıktır ve toplam 284.807 gözlemde 492 dolandırıcılıktan oluşmaktadır.

Bu da sadece %0,172 dolandırıcılık vakası ile sonuçlanmıştır. Bu çarpık set, dolandırıcılık işlemlerinin sayısının düşük olmasıyla doğrulanmaktadır.

b. Veri Seti

Veri kümesi, V1'den V28'e kadar olan 28 'Temel Bileşen Analizi (PCA)' dönüştürülmüş özelliğin sayısal değerlerinden oluşmaktadır. Ayrıca, orijinal özellikler hakkında herhangi bir meta veri sağlanmadığından ön analiz veya özellik çalışması yapılamamıştır.

'Zaman' ve 'Miktar' özellikleri dönüştürülmüş veriler değildir.

Veri kümesinde kayıp değer bulunmamaktadır.

```
# Gerekli kütüphanelerin yüklenmesi
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
import matplotlib.pyplot as plt
import pyodbc
import pandas as pd
import numpy as np
import seaborn as sns
sns.set()

# scaler = StandardScaler()
# x_scaled = scaler.fit_transform(X)
def sql(server,database,query):
    conn = pyodbc.connect(f'DRIVER=SQL Server;'
                        f'SERVER={server};'
                        f'DATABASE={database};'
                        f'Trusted_Connection=yes')
    return pd.read_sql(query, conn, coerce_float=True)

# Numpy float format
np.set_printoptions(precision=4, suppress=True, formatter={'float_kind': '{:0.2e}'})

# Pandas float format
pd.set_option('display.float_format', '{:0.2f}'.format)
```

Şekil 8.1: Kütüphane Ve Sql Connection Bilgileri.

Uygulamanın sonuçlanması ve verilerin okunabilirliği açısından kullandığım Python IDE'si; web uygulamaları, yazılım geliştirme, veri bilimi ve makine öğreniminde (ML) yaygın olarak kullanılan bir programlama dilidir. Geliştiriciler, etkili ve öğrenmesi kolay olduğu ve birçok farklı platformda çalıştırılabildiği için Python'ı kullanır. Program üzerinden yapmış olduğum uygulama için ihtiyaç duyulan kütüphaneler eklendikten sonra kısa yolu tanıtılır. Daha sonra hazırlanmış olan veri seti My Sql üzerinden ETL yöntemi ile yüklenir. Yüklenen veriler için kullanılan veritabanı bilgileri kısa yol olarak tanıtılır. Veriler üzerinde bulunan değerlerin formatıda belirlenir.

```
df = sql('S0134K0SEX01','BI','SELECT * FROM [dbo].[DummyData_06022024]')
✓ 74s
c:\ProgramData\Anaconda3\lib\site-packages\pandas\io\sql.py:762: UserWarning: pandas only support SQLAlchemy connectable(engine/connection) or database string URI or sqlite3 DBAPI2 conn
warnings.warn(
```

Şekil 8.2: SQL Connection Bilgileri.

Uygulama üzerinde verilerin çağırılacağı değer tanıtılır. Uygulama üzerinde kullanılacak veri seti artık “df” olarak çağırılacaktır.

```
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	154301	2.05	-0.10	-1.44	0.17	0.17	-0.78	0.10	-0.19	0.43	...	0.28	0.91	-0.04	-0.37	0.25	-0.10	-0.02	-0.07	0.74	False
1	154301	1.82	-0.17	-1.83	1.28	0.50	-0.62	0.62	-0.25	0.04	...	0.13	0.33	-0.13	-0.40	0.42	-0.49	-0.04	-0.06	104.00	False
2	154302	1.92	0.12	-1.56	1.32	0.46	-0.85	0.62	-0.31	-0.14	...	0.13	0.53	-0.04	0.04	0.42	-0.52	-0.02	-0.06	39.97	False
3	154303	1.93	-0.41	-0.89	0.48	-0.46	-0.15	-0.82	0.26	1.20	...	0.16	0.52	0.16	0.67	-0.31	0.56	-0.03	-0.03	17.00	False
4	154303	-0.03	-0.65	-2.16	-2.48	0.60	-1.51	0.32	-0.06	-3.08	...	0.53	1.66	0.12	0.83	-1.21	-0.13	0.18	0.17	49.00	False

5 rows x 31 columns

Şekil 8.3: Veri Kümesi Gösterimi.

Pandas kütüphanesinin bir fonksiyonu olan “head” fonksiyonu veri setinin ilk n satırını getirir. Bu n değeri varsayılan olarak 5’tir. Yukarıdaki şekilde veri setine ait Time değişkenini, Amount, Class ve V1-V28 PCA dönüştürülmüş sayısal değerleri değerlerini gösterir.

```
df.describe()
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V20	V21	V22	V23	V24	V25	V26
count	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00	...	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00	284807.00
mean	0.00	-0.00	-0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00	...	0.00	-0.00	-0.00	-0.00	0.00	-0.00	-0.00
std	1.96	1.65	1.52	1.42	1.38	1.33	1.24	1.19	1.10	1.09	...	0.77	0.73	0.73	0.62	0.61	0.52	0.48
min	-56.41	-72.72	-48.33	-5.68	-113.74	-26.16	-43.56	-73.22	-13.43	-24.59	...	-54.50	-34.83	-10.93	-44.81	-2.84	-10.30	-2.60
25%	-0.92	-0.60	-0.89	-0.85	-0.69	-0.77	-0.55	-0.21	-0.64	-0.54	...	-0.21	-0.23	-0.54	-0.16	-0.35	-0.32	-0.33
50%	0.02	0.07	0.18	-0.02	-0.05	-0.27	0.04	0.02	-0.05	-0.09	...	-0.06	-0.03	0.01	-0.01	0.04	0.02	-0.05
75%	1.32	0.80	1.03	0.74	0.61	0.40	0.57	0.33	0.60	0.45	...	0.13	0.19	0.53	0.15	0.44	0.35	0.24
max	2.45	22.06	9.38	16.88	34.80	73.30	120.59	20.01	15.59	23.75	...	39.42	27.20	10.50	22.53	4.58	7.52	3.52

8 rows x 29 columns

Şekil 8.4: Veri Kümesine Ait Değerlerinin İstatistiksel Gösterimi.

Pandas kütüphanesinin bir fonksiyonu olan “describe” fonksiyonu veri setinde sayısal değerleri sahip olan değişkenlere ait count, mean, max, min gibi temel istatistiksel değerleri gösterir.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Time        284807 non-null object
 1   V1          284807 non-null float64
 2   V2          284807 non-null float64
 3   V3          284807 non-null float64
 4   V4          284807 non-null float64
 5   V5          284807 non-null float64
 6   V6          284807 non-null float64
 7   V7          284807 non-null float64
 8   V8          284807 non-null float64
 9   V9          284807 non-null float64
10  V10         284807 non-null float64
11  V11         284807 non-null float64
12  V12         284807 non-null float64
13  V13         284807 non-null float64
14  V14         284807 non-null float64
15  V15         284807 non-null float64
16  V16         284807 non-null float64
17  V17         284807 non-null float64
18  V18         284807 non-null float64
19  V19         284807 non-null float64
...
29  Amount     284807 non-null float64
30  Class      284807 non-null bool
dtypes: bool(1), float64(29), object(1)
memory usage: 65.5+ MB
```

Şekil 8.5: Veri Kümesine Ait Değişkenlerin Türlerinin Gösterimi.

Pandas kütüphanesinin bir fonksiyonu olan “info” fonksiyonu veri setinde toplam sütun ve satır bilgisini, değişken türleri (Veri setine ait float64, Boolean ve object değişken türleri bulunur.) gösterir.

```
pd.set_option("display.float", "{:.3f}".format)
df.describe()
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V20	V21	V22	V23	V24
count	284807.000	284807.000	284807.000	284807.000	284807.000	284807.000	284807.000	284807.000	284807.000	284807.000	...	284807.000	284807.000	284807.000	284807.000	284807.000
mean	0.000	-0.000	-0.000	0.000	0.000	0.000	-0.000	0.000	-0.000	0.000	...	0.000	-0.000	-0.000	-0.000	0.000
std	1.959	1.651	1.516	1.416	1.380	1.332	1.237	1.194	1.099	1.089	...	0.771	0.735	0.726	0.624	0.606
min	-56.408	-72.716	-48.326	-5.683	-113.743	-26.161	-43.557	-73.217	-13.434	-24.588	...	-54.498	-34.830	-10.933	-44.808	-2.837
25%	-0.920	-0.599	-0.890	-0.849	-0.692	-0.768	-0.554	-0.209	-0.643	-0.535	...	-0.212	-0.228	-0.542	-0.162	-0.355
50%	0.018	0.065	0.180	-0.020	-0.054	-0.274	0.040	0.022	-0.051	-0.093	...	-0.062	-0.029	0.007	-0.011	0.041
75%	1.316	0.804	1.027	0.743	0.612	0.399	0.570	0.327	0.597	0.454	...	0.133	0.186	0.529	0.148	0.440
max	2.455	22.058	9.383	16.875	34.802	73.302	120.589	20.007	15.595	23.745	...	39.421	27.203	10.503	22.528	4.585

8 rows x 29 columns

Şekil 8.6: Veri Kümesine Ait Değişkenlerin Format Düzenlemesi.

Veri setinde bulunan değişkenlerin “pd.set_option” fonksiyonu gösterim format düzenlenir.

```
[52]: df.columns
      ✓ 0ds
      Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
            'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
            'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
            'Class'],
            dtype='object')
```

Şekil 8.7: Veri Kümesine Ait Değişken Bilgisi Gösterimi

Veri setinde bulunan kolonların değişken bilgisi “df. columns ()” fonksiyonu ile gösterilir. Veri setinde V1'den V28'e kadar olan 28 'Temel Bileşen Analizi (PCA)' dönüştürülmüş özelliğin sayısal değer, Zaman kavramı, toplan değer ve sınıf değişkeni kolonları bulunur.

```
df.head(15)
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	154301	2.047	-0.098	-1.437	0.172	0.170	-0.784	0.101	-0.186	0.430	...	0.275	0.913	-0.040	-0.371	0.251	-0.101	-0.024	-0.074	0.740	False
1	154301	1.824	-0.172	-1.825	1.281	0.502	-0.616	0.624	-0.245	0.036	...	0.132	0.331	-0.132	-0.400	0.418	-0.495	-0.041	-0.060	104.000	False
2	154302	1.920	0.119	-1.559	1.316	0.457	-0.850	0.619	-0.315	-0.144	...	0.132	0.527	-0.038	0.039	0.422	-0.517	-0.021	-0.065	39.970	False
3	154303	1.932	-0.415	-0.890	0.475	-0.465	-0.151	-0.820	0.261	1.202	...	0.161	0.521	0.158	0.675	-0.306	0.564	-0.031	-0.028	17.000	False
4	154303	-0.028	-0.648	-2.163	-2.482	0.604	-1.505	0.317	-0.064	-3.083	...	0.533	1.664	0.121	0.834	-1.206	-0.128	0.180	0.171	49.000	False
5	154303	-0.198	1.240	-1.210	-1.441	1.579	-0.879	1.611	-0.297	-0.423	...	0.201	0.842	-0.385	-0.983	-0.149	0.136	0.498	0.308	1.540	False
6	154304	0.054	0.879	0.239	-0.622	0.506	-1.040	1.019	-0.211	-0.069	...	-0.268	-0.579	0.046	-0.111	-0.462	0.147	0.249	0.096	1.980	False
7	154304	1.858	-0.090	-1.807	0.606	0.112	-1.400	0.408	-0.411	0.539	...	0.264	0.734	-0.104	-0.065	0.211	-0.110	-0.016	-0.014	102.000	False
8	154304	1.636	-0.758	-2.360	0.413	0.313	-0.386	0.244	-0.098	1.256	...	-0.182	-0.867	-0.040	-0.134	-0.099	-0.095	-0.060	0.017	234.560	False
9	154305	1.978	-1.059	-1.870	-1.729	1.477	3.662	-1.332	0.998	1.621	...	0.223	0.750	0.143	0.796	-0.214	0.814	-0.015	-0.056	30.000	False
10	154305	1.946	-0.502	-0.475	0.136	-0.541	-0.118	-0.659	0.188	0.924	...	-0.221	-0.690	0.474	0.720	-0.707	0.179	-0.058	-0.047	19.890	False
11	154305	2.084	-0.026	-1.223	0.308	0.137	-0.884	0.199	-0.315	0.482	...	-0.304	-0.703	0.291	-0.403	-0.240	0.224	-0.064	-0.064	0.890	False
12	154305	0.105	0.553	0.625	-0.156	-0.331	0.635	-1.329	-2.524	0.590	...	-1.128	0.968	-0.236	-0.760	0.769	-0.023	0.159	0.267	0.010	False
13	154306	0.134	1.105	0.399	-0.566	1.056	-0.700	1.056	-0.154	-0.233	...	-0.322	-0.718	0.093	0.532	-0.389	0.115	0.232	0.086	6.990	False
14	154307	1.986	-0.772	-1.726	-0.903	1.659	3.848	-1.291	1.044	1.363	...	0.175	0.653	0.178	0.745	-0.131	0.146	0.036	-0.049	5.000	False

15 rows x 31 columns

Şekil 8.8: Veri Setinin Düzenlenmiş Gösterimi.

Veri setinin düzenlenmiş ve seçilmiş 15 adet değişken bilgisi için “df. head ()” fonksiyonu ile gösterilir.

```

LABELS = ["Normal", "Fraud"]

count_classes = pd.value_counts(df['class'], sort = True)
count_classes.plot(kind = 'bar', rot=0)
plt.title("Transaction Class Distribution")
plt.xticks(range(2), LABELS)
plt.xlabel("Class")
plt.ylabel("Frequency")

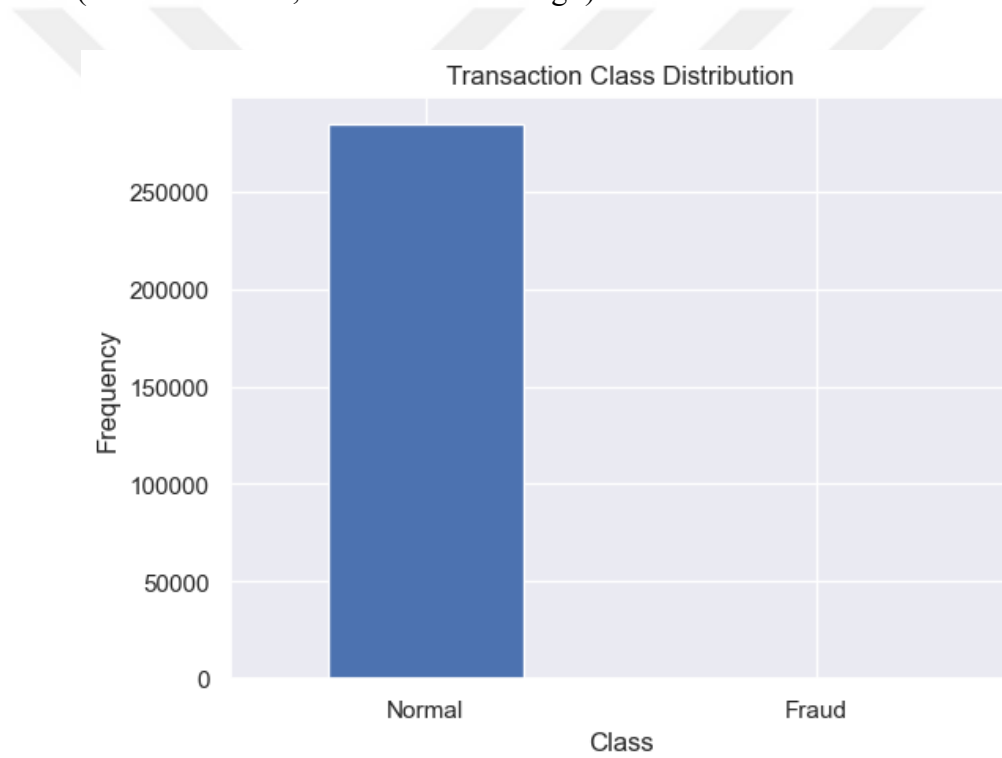
[34] ✓ 0.5s
... Text(0, 0.5, 'Frequency')

```

Şekil 8.9: Veri Kümesine Ait Değişkenlerin İşlem Bazlı Gösterimi.

Uygulamada kullanılacak modeller için dönüştürülmemiş değişkenler;

- Zaman
- Miktar
- Sınıf (1: dolandırıcılık, 0: dolandırıcılık değil)



Şekil 8.10: Veri Kümesine Ait Değişkenlerin İşlem Sınıfı Dağılımı.

```

df.class.value_counts()

[35] ✓ 0.0s
...
False    284315
True       492
Name: class, dtype: int64

```

Şekil 8.11: Veri Kümesine Ait İşlemlerin Sınıf Bazında Toplamı.

Veri setinde bulunan dolandırıcılık işlemi be dolandırıcılık işlemi olmayan değerler toplamı için kullanılan Sınıf (Class) değişkeni için Doğru (True) – Yanlış (False) değerlerine ait değerler toplamı gösterilmiştir. Veri kümesindeki 492 adet işlemin dolandırıcılık işlemi olduğu gözlemlenmektedir.

Orijinal veri setimizin ne kadar dengesiz olduğuna bir kez daha dikkat çekebiliriz. İşlemlerin çoğu dolandırıcılık işlemi değildir. Bu veri çerçevesini tahmin modellerimiz ve analizlerimiz için temel olarak kullanırsak çok fazla hata alabilir ve kullanılan algoritmaları üzerinde muhtemelen çoğu işlemin dolandırıcılık olmadığını "varsayacağı" için aşırı uyum sağlayacaktır. Ancak modelimizin varsayımda bulunmasını istemiyoruz, modelimizin dolandırıcılık işaretleri veren kalıpları tespit etmesini amaçlıyoruz.

```
fraud = df[df['Class']==1]
normal = df[df['Class']==0]

print(f"Shape of Fraudulent transactions: {fraud.shape}")
print(f"Shape of Non-Fraudulent transactions: {normal.shape}")
```

[17] ✓ 0.0s Python

... Shape of Fraudulent transactions: (492, 31)
Shape of Non-Fraudulent transactions: (284315, 31)

Şekil 8.12: Veri Kümesine Ait Sınıflandırılmış İşlemlerin Gösterimi.

Sahte işlemlerin adedi: (492, 31)

Sahte Olmayan işlemlerin şekli: (284315, 31)

```
pd.concat([fraud.Amount.describe(), normal.Amount.describe()], axis=1)
```

[18] ✓ 0.0s Python

...

	Amount	Amount
count	492.000	284315.000
mean	122.211	88.291
std	256.683	250.105
min	0.000	0.000
25%	1.000	5.650
50%	9.250	22.000
75%	105.890	77.050
max	2125.870	25691.160

Şekil 8.13: Veri Kümesine Ait Toplam Değişkenin Değer Bazında Gösterimi.

Veri kümesi üzerinde fraud olmayan değerlerin “Toplam” değeri üzerinden istatistiksel gösterimdir.

```
pd.concat([fraud.Time.describe(), normal.Time.describe()], axis=1)
```

[19] ✓ 0.2s Python

...

	Time	Time
count	492	284315
unique	468	124479
top	60207	163152
freq	6	36

Şekil 8.14: Veri Kümesine Ait Zaman Değişkenin Değer Bazında Gösterimi.

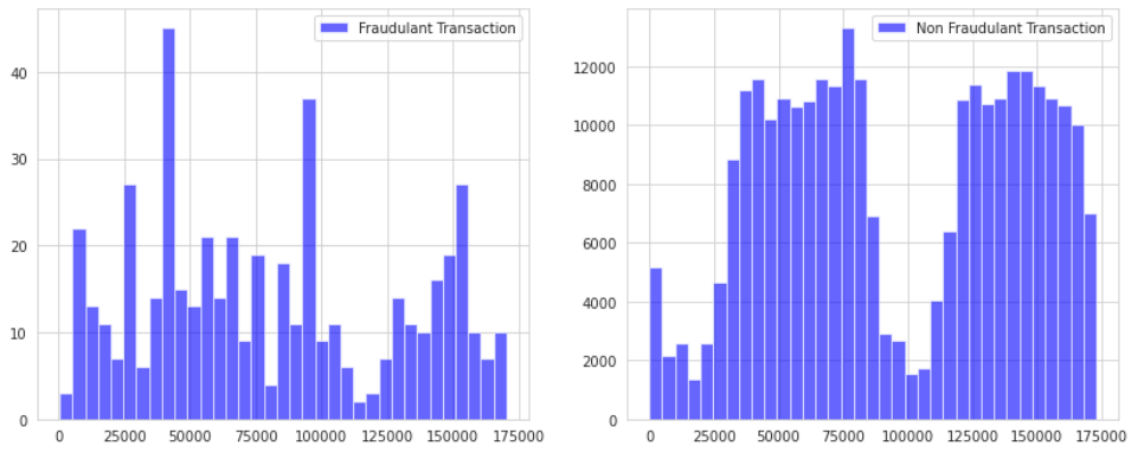
Veri kümesi üzerinde fraud olmayan değerlerin “Zaman” değeri üzerinden istatistiksel gösterimidir.

```
# data[data.Class == 0].Time.hist(bins=35, color='blue', alpha=0.6)
plt.figure(figsize=(14, 12))

plt.subplot(2, 2, 1)
df[df.Class == '1'].Time.hist(bins=35, color='blue', alpha=0.6, label="Fraudulent Transaction")
plt.legend()

plt.subplot(2, 2, 2)
df[df.Class == '0'].Time.hist(bins=35, color='blue', alpha=0.6, label="Non Fraudulent Transaction")
plt.legend()
```

Şekil 8.15: Veri Kümesine Ait İşlem Bazlı Transaction Gösterimi.

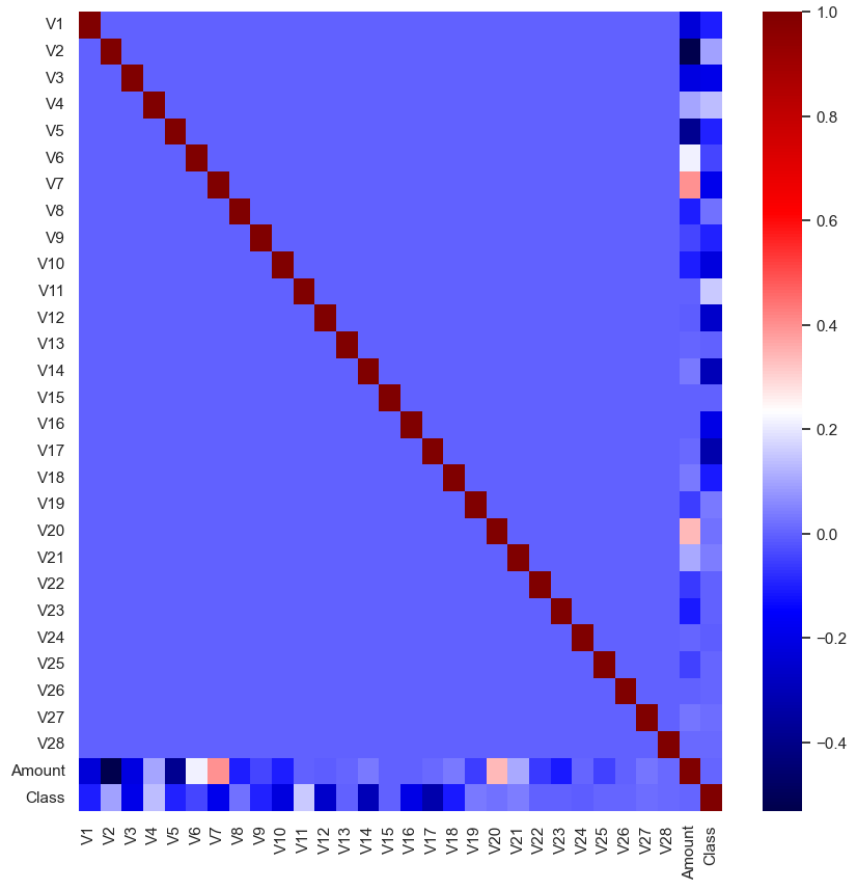


Şekil 8.16: Veri Kümesine Ait İşlem Bazlı Transaction Gösterimi.

Şekil 8.16’da gösterilen dolandırıcılık ve dolandırıcılık olmayan işlemlerin dağılım grafiğine bakacak olursak, veri setinde bulunan bu özelliklerin ne kadar çarpık olduğu hakkında fikir sahibi olabiliriz. Veri setinde bulunan diğer değişken özelliklerinin dağılımlarını görmek mümkün.

```
plt.figure(figsize=(10,10))
sns.heatmap(data=df.corr(), cmap="seismic")
sns.boxplot(data=df.corr(), cmap="seismic")
plt.show()
```

Şekil 8.17: Pyhton IDE’si Üzerinde Veri Kümesinin Matrix Formülü Gösterimi.



Şekil 8.18: Pyhton IDE'si Üzerinde Veri Kümesine Ait Matrix Gösterimi.

Yukarıdaki şekilde Seaborn kütüphanesine ait kolerasyon matrisi üzerinde veri kümesinde bulunan değişkenlerin özelliklerini görselleştirerek anlama yorumlamızı kolaylaştıran bir matrix elde ettik.

Bu matrixe bakarak en yüksek korelasyonlar;

- a. Zaman & V3 (-0,42)
- b. Miktar & V2 (-0,53)
- c. Miktar & V4 (0,4)

Bu korelasyonlar yüksek olsa da, çoklu doğrusallık riski taşımamasını beklenemez. Korelasyon matrisi ayrıca V1 ila V28 PCA bileşenlerinin hiçbirinin birbiriyle herhangi bir korelasyona sahip olmadığını göstermektedir.

Ancak Sınıf'ın V bileşenleriyle pozitif ve negatif korelasyonlara sahip olduğunu, ancak Zaman ve Miktar ile hiçbir korelasyona sahip olmadığını gözlemlenmektedir.

```
>>> from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

scalar = StandardScaler()

X = df.drop('Class', axis=1)
y = df.Class

X_train_v, X_test, y_train_v, y_test = train_test_split(X, y,
                                                         test_size=0.3, random_state=42)
X_train, X_validate, y_train, y_validate = train_test_split(X_train_v, y_train_v,
                                                            test_size=0.2, random_state=42)

X_train = scalar.fit_transform(X_train)
X_validate = scalar.transform(X_validate)
X_test = scalar.transform(X_test)

w_p = y_train.value_counts()[0] / len(y_train)
w_n = y_train.value_counts()[1] / len(y_train)

print("Fraudulent transaction weight: (w.p)")
print("Non-Fraudulent transaction weight: (w.n)")

[66] ✓ 25s Python
... Fraudulent transaction weight: 0.0018380243098356647
Non-Fraudulent transaction weight: 0.9981619756901644
```

Şekil 8.19: Python IDE'si Üzerinde Veri Kümesinin Ölçeklendirilmesi.

Veri Ön işleme yapılarak “Zaman” ve “Miktar” değişkenleri diğer sütunlar gibi ölçeklendirilmelidir.

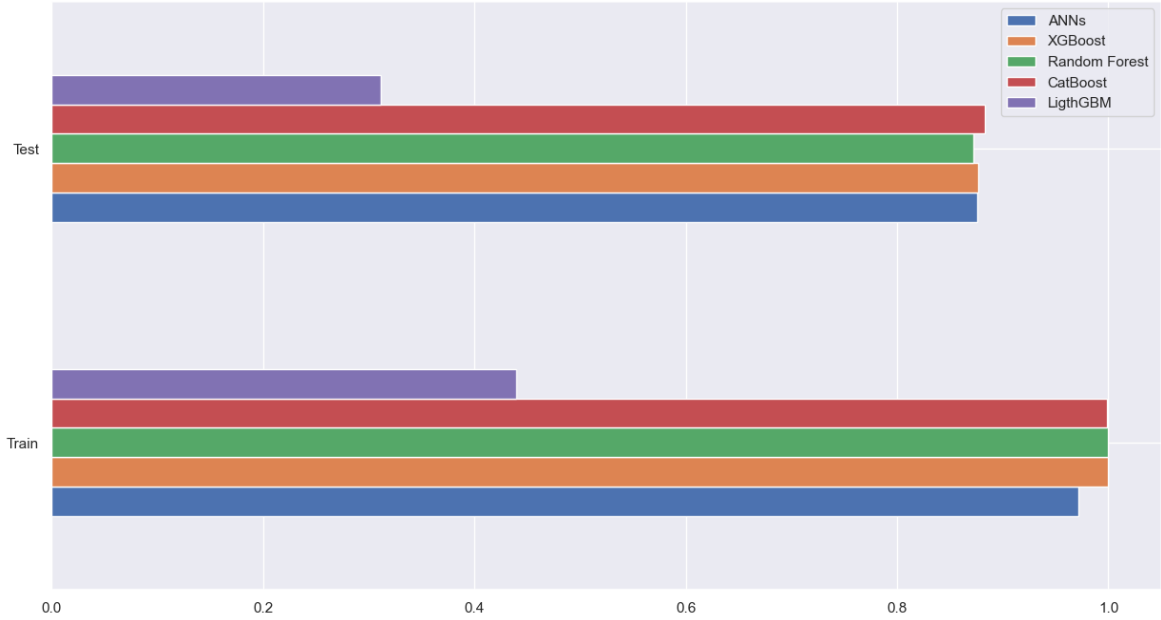
Çizelge 7.1’de, sınıflandırma görevinin deneysel sonuçlarını göstermektedir. Bu deneysel sonuçlar, kredi kartı işlem veri seti üzerinde farklı makine öğrenimi modellerinin performansını değerlendirmek için elde edilmiştir. Veri seti öncelikle temizlenmiş, ardından normalizasyon ve veri standardizasyon süreçleri uygulanmıştır. Daha sonra, veri kümesi eğitim (%70) ve test (%30) veri kümeleri olarak ayrılmıştır. Eğitim veri seti, çeşitli modellerin oluşturulması için kullanılmıştır ve test veri seti bu modellerin performansının değerlendirilmesinde kullanılmıştır.

Aşağıda, Çizelge 8.1’de gösterilen sınıflandırma görevinin deneysel sonuçlarını açıklayan bir örnek bulunmaktadır:

Çizelge 8.1: Model Sınıflarının Karşılaştırılması.

	Accuracy	Precision	Recall	F1-Score
ANNs	0.9999	0.998	0.974	0.986
XGBoost	0.9996	0.974	0.974	0.986
Random Forest	0.9996	0.979	0.901	0.936
CatBoost	0.9996	1.00	0.997	0.998
LightGBM	0.9996	0.997	0.997	0.997

Bu tablo, farklı makine öğrenimi modellerinin performansını göstermektedir. Örneğin, XGBoost modelinin %99 doğruluk, %97 hassasiyet ve %98 F1 skoru elde ettiği görülmektedir, bu da veri集中的 sınıfları ne kadar iyi ayırdığını gösterir. Diğer modellerin performansı da benzer şekilde değerlendirilmiştir.



Şekil 8.20: Veri Kumesine Ait Modelleme Sonucu.

Farklı modellerin performansı doğruluk, hassasiyet, geri çağırma ve F1 puanı gibi farklı ölçümler kullanılarak değerlendirilir.

Çeşitli makine öğrenimi yöntemlerinin performansı, topluluk öğrenimi tabanlı XGBoost modeliyle karşılaştırılmıştır. Analizin XGBoost, Random Forest ve CatBoost modellerinin doğruluğu diğer geleneksel modellere göre daha yüksektir.

Bu, XGBoost, Random Forest ve CatBoost gibi topluluk öğrenmeye dayalı yaklaşımın diğer geleneksel yaklaşımlara göre daha iyi tahmin gerçekleştirdiğini göstermektedir. Şekil 8.19’da bu modellerin karşılaştırmalı grafik sonuçlarını temsil eder.

Bu çalışma, kredi kartı dolandırıcılığı tespiti için topluluk tabanlı XGBoost modeli ile diğer geleneksel makine öğrenmesi modellerini karşılaştırarak önemli bulgular ortaya koymaktadır. XGBoost, Random Forest ve CatBoost modellerinin, kesinlik, geri çağırma gibi performans ölçütleri üzerinde daha iyi performans sergilediği belirtilmektedir.

Bu sonuçlar, finansal kurumların günlük operasyonlarında karşılaştıkları dolandırıcılık risklerini azaltma potansiyeline işaret etmektedir.

Çalışmanın dikkate değer bir noktası, XGBoost, Random Forest ve CatBoost'un dengesiz veri setleriyle başa çıkma yeteneğinin vurgulanmasıdır. Geleneksel modellerin zayıf performans gösterdiği bu durumlarda, XGBoost, RandomForest ve CatBoost'un %99'a kadar tahmin doğruluğu sağladığı ve diğer modellere göre daha iyi bir performans sunduğu gözlemlenmektedir.

Ayrıca, çalışmanın e-ticaret operasyonlarındaki büyüme ve internetten alışverişin artmasıyla ilgili bir bağlamda ele alınması önemlidir. Covid-19 salgını gibi faktörler, tüketicileri çevrimiçi işlemlere yönlendirerek dolandırıcılık risklerini artırabilir. Dolayısıyla, finansal kurumların etkili dolandırıcılık tespit sistemlerine sahip olmaları hayati önem taşımaktadır.

Önerilen gelecek çalışmalar arasında, hibrit modellerin oluşturulması ve tespit algoritmalarının çevrimiçi uygulanması bulunmaktadır. Bu yaklaşımların kullanılmasıyla, kredi kartı dolandırıcılığı tahminlerinde daha doğru ve etkili modeller geliştirilebilir. Bu çalışma, finansal kurumların dolandırıcılık risklerini azaltmak için makine öğrenimi ve topluluk tekniklerinin nasıl kullanılabileceğine dair önemli bir katkı sunmaktadır.

REFERANSLAR

- [1] Sait Uğur Gültekin, “Yapay Sinir Ağı ve Doğrusal Regresyon Yöntemleri İle Fiyat Tahmini” (Master’ thesis, Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü), 2017.
- [2] Kazım Soylu, “Kredi Kartı Sahte İşlem Tespiti”, 2018.
- [3] Ebru Pekel Özmen, Tuncay Özcan, “Finans Sektöründe Dolandırıcılık Tespiti Üzerine Melez Sınıflandırma ve Regresyon Ağacı Uygulaması”, Cilt 5, Sayı 2, Sayfa 12-20, 2019.
- [4] Safa, M. U. and R. Ganga, "Credit Card Fraud Detection Using Machine Learning.", 2019.
- [5] Yazıcıoğlu, N., “Yapay Zekâ İle Talep Tahmini”, (Yüksek Lisans Tezi), Uludağ Üniversitesi Fen Bilimler Enstitüsü, Bursa, 2010.
- [6] Kargı, V. S., Yapay Sinir Ağ Modelleri ve Bir Tekstil Firmasında Uygulama. Bursa: Ekin Basın Yayın Dağıtım, 23-80, 2015.
- [7] Ebru Pekel Özmen., Tuncay Özcan, “Finans Sektöründe Dolandırıcılık Tespiti Üzerine Melez Sınıflandırma ve Regresyon Ağacı Uygulaması” Cilt5, Sayı2, Sayfa12-20, 2019.
- [8] P. Kumar ve F. Iqbal, "Makine Öğrenimi Yaklaşımlarını Kullanarak Kredi Kartı Dolandırıcılığının Belirlenmesi", 2019 1. Uluslararası Bilgi ve İletişim Teknolojilerinde Yenilikler Konferansı (ICICT), Chennai, Hindistan, 2019.
- [9] Gölyeri, M., Çelik, S., Bozyiğit, F., & Kılınç. D. “Fraud detection on e-commerce transactions using machine learning techniques.” Artificial Intelligence Theory and Applications, 3(1), 45-50., 2023.
- [10] Umut Kaya, Atınç Yılmaz, Yalım Dikmen., “Sağlık Alanında Kullanılan Derin Öğrenme Yöntemleri”. European Journal of Science and Technology, sayı 16: 792–808. <https://doi.org/10.31590/ejosat.573248>., 2019.
- [11] Ölçer, İlayda, Ve Atınç Yılmaz., “Yapay Zekanın Cerrahi Uygulamalara Entegrasyonu”. Beykent Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi 13/2: 21–27. <https://doi.org/10.20854/Bujse.873770>., 2021.
- [12] Pirim, Harun., “YAPAY ZEKA”. Journal of Yaşar University, 81–93., 2006.
- [13] Gottfredson, Linda S., “Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography”. Intelligence 24 (1): 13–23. [https://doi.org/10.1016/s0160-2896\(97\)90011-8](https://doi.org/10.1016/s0160-2896(97)90011-8)., 1997.

- [14] Ölçer, İlayda, Ve Atınç Yılmaz., “Yapay Zekanın Cerrahi Uygulamalara Entegrasyonu”. *Beykent Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi* 13/2: 24–26. <https://doi.org/10.20854/Bujse.873770>., 2021.
- [15] Kononenko, Igor., “Machine learning for medical diagnosis: History, state of the art and perspective”. *Artificial Intelligence in Medicine* 23 (1): 89–109. ([https://doi.org/10.1016/S0933-3657\(01\)00077-X](https://doi.org/10.1016/S0933-3657(01)00077-X)), 2001.
- [16] Michalski, R. S., Carbonell, J. G., Mitchell, T. M., *Machine learning: an artificial intelligence approach*. Palo Alto, CA: Tioga Pub. Co., 1983.
- [17] Nilsson, N. J., *The quest for artificial intelligence: a history of ideas and achievement*. Cambridge: Cambridge University Press, 2010.
- [18] Paluszek, M., & Thomas, S., *An Overview of Machine Learning. MATLAB Machine Learning*, Vol. 1, pp. 3–15, 2016.
- [19] Arel, I., Rose, D. C., Karnowski, T. P., *Deep Machine Learning - A New Frontier in Artificial Intelligence Research*, *IEEE Computational Intelligence Magazine*, Vol. 5 No. 4, pp. 13-18., 2010.
- [20] Längkvist, M., Karlsson, L., Loutfi, A., *A review of unsupervised feature learning and deep learning for time-series modeling*. *Pattern Recognition Letters*, Vol. 42, pp. 11-24, 2014.
- [21] McCulloch, W. S., Pitts, W., *How we know universals: the perception of auditory and visual forms*, *Bulletin of mathematical Biophysics*, Vol. 9(3), pp. 127-147, 1947.
- [22] Hebb, D. O., *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall., 1949.
- [23] Turing, A., *Computing machinery and intelligence-AM Turing*, *Mind*, Vol. 59(236), pp. 433., 1950.
- [24] Samuel, A. L., *Some studies in machine learning using the game of checkers*, *IBM Journal of research and development*, Vol. 3(3), pp. 210-229., 1959.
- [25] Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain*, *psychological review*, Vol. 65(6), pp. 386., 1958.
- [26] Domingos, P., *A few useful things to know about machine learning*, *Communications of the ACM*, Vol. 55(10), pp. 78-87., 2012.
- [27] Friedman, J. H., Tukey, J. W., *A projection pursuit algorithm for exploratory data analysis*, *IEEE Transactions on computers*, Vol. 100(9), pp. 881-890., 1974.

- [28] Behrens, J. T., Principles and procedures of exploratory data analysis, Psychological Methods, Vol. 2(2), pp. 131., 1997.
- [29] Martinez, W. L., Martinez, A. R., Solka, J., Martinez, A., Exploratory data analysis with MATLAB. Crc Press., 2010.
- [30] Behrens, J. T., DiCerbo, K. E., Yel, N., Levy, R., Exploratory data analysis. Handbook of Psychology, Second Edition, 2, 2012.
- [31] Xue, B., Zhang, M., Browne, W. N., Yao, X., A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation, Vol. 20(4), pp. 606-626., 2015.
- [32] Abdi, H., Williams, L. J., Principal component analysis. Wiley interdisciplinary reviews: computational statistics, Vol. 2(4), pp. 433-459., 2010.
- [33] Ramanathan, U., Supply chain collaboration for improved forecast accuracy of promotional sales. International Journal of Operations & Production Management., 2012.
- [34] Christensen, K., Nørskov, S., Frederiksen, L., & Scholderer, J., In search of new product ideas: Identifying ideas in online communities by machine learning and text mining. Creativity and Innovation Management, Vol 26(1), pp. 17-30., 2017.
- [35] Gartner Research, Market Guide for Trade Promotion Management and Optimization for the Consumer Goods Industry, 2017.
- [36] Rod Koch CMA, P. M. P., From business intelligence to predictive analytics. Strategic Finance, Vol. 96(7), pp. 56, 2015.
- [37] Folinas, D., & Rabi, S., Estimating benefits of Demand Sensing for consumer goods organisations. Journal of Database Marketing & Customer Strategy Management, Vol. 19(4), pp. 245-261., 2012.
- [38] Caruana, R., Niculescu-Mizil, A., An empirical comparison of supervised learning algorithms, 23rd International Conference on Machine learning, pp. 161-168., 2006.

- [39] Özgür Doğan, Makine Öğrenmesi Nedir?, (2020). Makine Öğrenmesi Algoritmaları. [Online]. Url (<https://teknoloji.org/makine-ogrenmesi-nedir-makine-ogrenmesi-algoritmaları/>).
- [40] Ghahramani, Z., Unsupervised learning, Summer School on Machine Learning, Springer, pp. 72-112., 2003.
- [41] Hatice Candan, (2022). Adım Adım Makine Öğrenmesi Bölüm 4: Denetimli Öğrenme ve Denetimsiz Öğrenme Arasındaki Fark, [Online]. Url (<https://medium.com/machine-learning-t%C3%BCrkiye/ad%C4%B1m-ad%C4%B1m-makine-%C3%B6%C4%9Frenmesi-b%C3%B6l%C3%BCm-4-denetimli-%C3%B6%C4%9Frenme-ve-denetimsiz-%C3%B6%C4%9Frenme-aras%C4%B1ndaki-fark-4aa174983380>).
- [42] Kumar, V., Kalitin, D., Tiwari, P., Unsupervised learning dimensionality reduction algorithm PCA for face recognition, 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 32-37., 2017.
- [43] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S., A geometric framework for unsupervised anomaly detection, Applications of data mining in computer security, pp. 77-101., 2002.
- [44] Kaelbling, L. P., Littman, M. L., Moore, A. W., Reinforcement learning: A survey. Journal of artificial intelligence research, Vol. 4, pp. 237-28, 1996.
- [45] Özkan Doğan, Pekiştirmeli Öğrenmeye Giriş, (2019). [Online]. Url (<https://medium.com/deep-learning-turkiye/peki%C5%9Ftirmeli%C3%B6%C4%9Frenmeye-giri%C5%9F-c7c2a8cce50b>).
- [46] Li, D., Zhang, B., Li, C., A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems, IEEE Internet of Things Journal, Vol. 3(4), pp. 590-597. Tsai, C. F. (2009). Feature selection in bankruptcy prediction, Knowledge-Based Systems, Vol. 22(2), pp. 120-127., 2015.

- [47] Matsunaga, A., & Fortes, J. A., On the use of machine learning to predict the time and resources consumed by applications, 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 495-504., 2010.
- [48] Seber, G. A., & Lee, A. J., Linear regression analysis, Vol. 329. John Wiley & Sons., 2012.
- [49] Cem İstanbullu, Python ile doğrusal regresyon uygulaması, (2022). [Online]. Url (<https://miuul.com/not-defteri/python-ile-dogrusal-regresyon-uygulamasi>)
- [50] Mohammed, M., Bashier, E., Machine Learning: Algorithms and Applications. Crc Press., 2016.
- [51] Rümeyza Nazlı, R İle Lojistik Regresyon Analizi, (2021). [Online]. Url: [8https://medium.com/@rmys.nzl/r-i-CC%87le-lojistik-regresyon-analizi-600e188d2f1e](https://medium.com/@rmys.nzl/r-i-CC%87le-lojistik-regresyon-analizi-600e188d2f1e)).
- [52] Freund, Y., Mason, L., The alternating decision tree learning algorithm, ICML, Vol. 99, pp. 124-133, 1999.
- [53] Mehmet Fatih Akca, Karar Ağaçları (Makine Öğrenmesi Serisi-3), (2020). [Online]. Url (<https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5>).
- [54] Kuang, Q., Zhao, L., A practical GPU based kNN algorithm, International Symposium on Computer Science and Computational Technology (ISCSCI 2009), pp. 151., 2009.
- [55] Özgür Doğan, Makine Öğrenmesi Nedir?, (2020). Makine Öğrenmesi Algoritmaları. [Online]. Url (<https://teknoloji.org/makine-ogrenmesi-nedir-makine-ogrenmesi-algoritmaları/>)
- [56] Kotsiantis, S., Supervised Machine Learning: A Review of Classification Techniques, Emerging artificial intelligence applications in computer engineering, Vol. 160, pp. 3-24., 2007.

- [57] Yiğit Şener, Destek Vektör Makineleri (Support Vector Machine | SVM) Çalışma Mantığı ve Python Uygulaması, 2021. [Online]. Url (<https://yigitsener.medium.com/destek-vekt%C3%B6r-makineleri-support-vector-machine-svm-%C3%A7al%C4%B1%C5%9Fma-mant%C4%B1%C4%9F%C4%B1-ve-python-uygulamas%C4%B1-992163ff3eec>).
- [58] Feng Z., Xu C., Tao D., Historical Gradient Boosting Machine. GCAI-2018: 4th Global Conference on Artificial Intelligence, Luxembourg City / Luxembourg, September 18-21, 2018.
- [59] Dhivya Ravindran, (2021). Tree-Based Machine Learning Algorithms Explained. [Online]. Url(<https://medium.com/analytics-vidhya/tree-based-machine-learning-algorithms-explained-b50937d3cf8e>).
- [60] Ali Cihat Kelle, Hüseyin Yüce., MQTT Trafığında DoS Saldırılarının Makine Öğrenmesi ile Sınıflandırılması ve Modelin SHAP ile Yorumlanması, 2022.
- [61] Cem Deniz Kumral, Ali Topal, Mevlüt Ersoy, Recep Çolak, Tuncay Yiğit, Random Forest Algoritmasının FPGA Üzerinde Gerçekleştirilerek Performans Analizinin Yapılması, 2022.
- [62] Sruthi E R., Understand Random Forest Algorithms with Examples, 2024. [Online]. Url (<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>)
- [63] Leo Breiman, RANDOM FORESTS, 2001.
- [64] Niklas Donges, Random Forest: A Complete Guide for Machine Learning, (2023). [Online]. Url(<https://builtin.com/data-science/random-forest-algorithm>).
- [65] Tanju Doğan, Yapay Sinir Ağları ve Tek Katmanlı Ağlarda Öğrenme, (2018). [Online]. Url:(<https://tr.linkedin.com/pulse/yapay-sinir-a%C4%9Flar%C4%B1-ve-tek-katmanl%C4%B1-a%C4%9Flarda-%C3%B6%C4%9Frenme-tanju-do%C4%9Fan>).
- [66] Başoğlu, B., Bulut, M., “Kısa Dönem Elektrik Talep Tahminleri İçin Yapay Sinir Ağları ve Uzman Sistemler Tabanlı Hibrid Tahmin Sistemi Geliştirilmesi”, Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, Ankara, 575-583, 2017.

- [67] Eren Öztürk, Yapay Sinir Ağları (YS), (2022). [Online].
Url (<https://academy.patika.dev/blogs/detail/yapay-sinir-aglariys>).
- [68] Sağıroğlu, Ş., Beşdok, E. ve Erler, M., “Mühendislikte Yapay Zeka Uygulamaları-1: Yapay Sinir Ağları”, Ufuk Kitap Kırtasiye Yayıncılık, Kayseri, 2003.
- [69] Çayıroğlu, İ., “İleri Algoritma Analizi-5 Yapay Sinir Ağları”, 2013.
- [70] Körpü E. Yavuz., “Yapay Sinir Ağları”, 2020.
- [71] Başkurt, M., Durmuş, M. C. ve Gümüşay, M. Ü., “Yapay Sinir Ağları Yardımıyla Plaka Tanıma Sisteminin Oluşturulması ve CBS ile Bütünleştirilmesi”, 2011.
- [72] Hu, C., “Advanced Tourism Demand Forecasting: ANN and Box-Jenkins Modelling”, Doktora Tezi, Purdue University, MI, USA, 2002.
- [73] Haykin, S., “Neural Networks a Comprehensive Foundation”, (2 nd Edition). New Jersey: Prentice Hall, 1999.
- [74] Elmaz, Ç., “Yapay Sinir Ağları: (Kuram, Mimari, Eğiti Uygulama)”, Seçkin Yayıncılık, p.16., 2003.
- [75] Elmas, Ç., “Yapay Sinir Ağları:(Kuram, Mimari, Eğitim, Uygulama)”, Seçkin Yayıncılık, p. 18., 2003.
- [76] Özalp, A. ve Anagün, A.S., “Yapay Sinir Ağı Performansına Etki Eden Faktörlerin Analizinde Taguchi Yöntemi: Hisse Senedi Fiyat Tahmini Uygulaması”, İstatistik Araştırma Dergisi. 2(1): 29-40, 2003.
- [77] Keleşoğlu, Ö., Ekinci, C. E. ve Fırat, A. “The Using of Artificial Neural Networks in Insulation Computations, Sigma”, 3, 58-66, 2005.
- [78] Yılmaz, İ., “Osmanlı Dönemi Mimarlık Eserleri Restorasyon İnşaat Maliyetlerinin Yapay Zeka Yöntemleri ile Tahmini”, İstanbul Kültür Üniversitesi, 2012.
- [79] Öztemel, E., “Yapay Sinir Ağları”, 2005.

- [80] Özalp, A. ve Anagün, A.S., “Yapay Sinir Ağı Performansına Etki Eden Faktörlerin Analizinde Taguchi Yöntemi: Hisse Senedi Fiyat Tahmini Uygulaması”, İstatistik Araştırma Dergisi. 2(1): 29-45, 2003.
- [81] Başkurt, M., Durmuş, M. C. ve Gümüşay, M. Ü., “Yapay Sinir Ağları Yardımıyla Plaka Tanıma Sisteminin Oluşturulması ve CBS ile Bütünleştirilmesi”, 2011.
- [82] Tang, T. C. ve Chi, L.C., “Neural Networks Analysis in Business Failure Prediction of Chinese Importers: A Between-Countries Approach”, Expert Systems with Applications, (29): 244–255, 2005.
- [83] Hu, C., “Advanced Tourism Demand Forecasting: ANN and Box-Jenkins Modelling”, Doktora Tezi, Purdue University, MI, USA, 2002.
- [84] Özalp, A. ve Anagün, A.S., “Yapay Sinir Ağı Performansına Etki Eden Faktörlerin Analizinde Taguchi Yöntemi: Hisse Senedi Fiyat Tahmini Uygulaması”, İstatistik Araştırma Dergisi. 2(1): 29-45, 2003.