T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EGİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK KREDİ RİSK ANALİZİ

YÜKSEK LİSANS TEZİ

Ömer Yavuz CAN

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Şubat, 2020

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EGİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK KREDİ RİSK ANALİZİ

YÜKSEK LİSANS TEZİ

Ömer Yavuz CAN
(Y1713.010053)

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Üyesi Ahmet GÜRHANLI

Şubat, 2020

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ

YÜKSEK LİSANS TEZ ONAY FORMU

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı Y1713.010053 numaralı öğrencisi Ömer Yavuz CAN'ın **"Makine Öğrenmesi Teknikleri Kullanılarak Kredi Risk Analizi"** adlı tez çalışması Enstitümüz Yönetim Kurulunun 31.01.2020 tarihli ve 2020/02 sayılı kararıyla oluşturulan jüri tarafından oybirliği/oyçokluğu ile Tezli Yüksek Lisans tezi 21.02.2020 tarihinde kabul edilmiştir.

	Unvan	Adı Soyadı	Üniversite	İmza
		ASIL ÜYELER		
Danışman	Dr. Öğr. Üyesi	Ahmet GÜRHANLI	İstanbul Aydın Üniversitesi	
1. Üye	Dr. Öğr. Üyesi	Adem ÖZYAVAŞ	İstanbul Aydın Üniversitesi	
2. Üye	Dr. Öğr. Üyesi	Ali HAMİTOĞLU	İstanbul Sabahattin Zaim Üniversitesi	
		YEDEK ÜYELER		
1. Üye	Prof. Dr.	Ali GÜNEŞ	İstanbul Aydın Üniversitesi	
2. Üye	Doç. Dr.	Fatih KOÇAN	İstanbul Gelişim Üniversitesi	

ONAY

Prof. Dr. Ragıp Kutay KARACA
Enstitü Müdürü

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum “Makine Öğrenmesi Teknikleri Kullanılarak Kredi Risk Analizi” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (…….)

Ömer Yavuz CAN

ÖNSÖZ

Başta doğduğum bu topraklarda bizim kültürümüzü ve manevi yapımızı oluşturan bu milletin ve devletin bir ferdi olmaktan büyük gurur duymaktayım. Bununla beraber doğumumdan bugüne beni maddi ve manevi destekleyen, benim yapı taşım olan anne ve babama;

Çalıştığım Firma olan OBJEKT Bilişim İnşaat A.Ş.'nin patronu Necdet ÇAKIR' a;

Çalıştığım Firma olan OBJEKT Bilişim İnşaat A.Ş.'nin genel müdürü Tuğba DOĞAN' a;

Çalıştığım Firma olan OBJEKT Bilişim İnşaat A.Ş.'den müdürüm Oğuzhan DOĞAN' a;

Ve bütün OBJEKT Bilişim İnşaat A.Ş. personeline;

Okuduğum okullarda her biri büyük bir katkı sağlayan değerli öğretmenlerime;

Değerli rektörüm Prof. Dr. Mustafa AYDIN' a;

Bölüm başkanım Prof. Dr. Ali GÜNEŞ' e;

İstanbul Aydın Üniversitesi'nde öğrettikleri bilgiler ve verdiği desteklerden dolayı Prof. Dr. Muttalip Kutluk ÖZGÜVEN, Prof. Dr. Haluk GÜMÜŞKAYA, Prof. Dr. Zafer ASLAN, Doç. Dr. Metin ZONTUL, Doç. Dr. Ferdi SÖNMEZ, Doç. Dr. İlham HUSEYINOV, Doç. Dr. Taner ÇEVİK, Dr. Öğr. Üyesi Adem ÖZYAVAŞ, Dr. Öğr. Üyesi Farzad KIANI, Dr. Öğr. Üyesi İlknur DÖNMEZ, Dr. Öğr. Üyesi Ali Alaa HAMEED, Dr. Öğr. Üyesi Mehmet Kamil TULGA'ya;

En son olarak da bu projenin hazırlanmasında bana önderlik ve rehberlik yapan Dr. Öğr. Üyesi Ahmet GÜRHANLI' ya sonsuz şükranlarımı ve teşekkürlerimi belirtip, kendilerinden aldığım bu değerli bayrağı daha üst seviyelere çıkarmak için çalışacağıma bu projeyle söz veririm.

Şubat, 2020 **Ömer Yavuz CAN**

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ

TP **:** True Positive

TN **:** True Negative

FP **:** False Positive

FN **:** False Negative

DVM **:** Destek Vektör Makineleri

XGBoost **:** Extreme Gradient Boosting

ADABoost **:** Adaptive Boosting

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK KREDİ RİSK ANALİZİ

ÖZET

İnsanların son dönemlerde bankalardan kredi talepleri oldukça fazlalaştığı görülmektedir. Bu durum bankalar açısından olumlu bir durum gibi gözükse de aynı zamanda çok fazla risk teşkil etmektedir. Banka ve finans sektörlerinde risk yönetiminin doğru yapılması, mevcut olan kaynakların verimli ve iyi kullanılması, oluşacak riskleri tahmin ederek zamanında önlem alınmasına ile bağlantılıdır. Sorun teşkil eden kredilerin öngörülebilir olması bankalar için kararlılık açısından büyük önem taşımaktadır. Kredi almak için talepte bulunan kişilere, bankaların kredi vermesi, bankaların temel faaliyetlerdendir. Fakat bu temel faaliyet aynı zamanda riskli bir faaliyettir. Bankalar kuruluş amaçları gereği risk almaktan kaçınmazlar ve alınan bu riskleri yönetmektedirler. Bu risk yönetimini yaparken, bankaların verilen kredi tutarlarından oluşabilecek zararları en az seviyede tutabilecek şekilde risk yönetimlerini yapmaları gerekir. Bütün bu sebepler göz önünde bulundurularak, son dönemlerde bankaların kredilendirme işlemlerini hızlandırmak ve olumlu kararlar verebilmek adına veri madenciliği başta olmak üzere, farklı farklı algoritma modelleri, algoritma sınıflandırmaları, yapay sinir ağları gibi makine öğrenmesi tekniklerini kullanmaya başladıkları görülmektedir. Bu çalışmada çeşitli makine öğrenmesi tekniklerinden yararlanılarak kredi talebinde bulunan müşterilerin krediye uygun olup, olmadığının doğruluğu test edilmiştir. Veri seti olarak german credit data UCI' de bulunan erişimi açık veri kümesi kullanılmıştır. Bu çalışmadaki veri kümesinde bulunan 1000 adet müşteri baz alınarak XGBoost sınıflandırıcısında %75,60 başarı oranı yakalanmıştır. Bu başarı oranı daha önce XGBoost sınıflandırıcısı ile yapılan çalışmalar arasında en yüksek başarı oranına sahiptir. Ayrıca yapılan diğer çalışmalarda kullanılan algoritmalar içerisinde de en yüksek başarı oranı sağlanmıştır.

Anahtar Kelimeler: *Kredi Risk Analizi, Makine Öğrenmesi, Veri Madenciliği*

CREDIT RISK ANALYSIS USING MACHINE LEARNING TECHNIQUES

ABSTRACT

It can be easily observed that the general public is putting in more and more loan requests in the banking system recently, which can be regarded as a positive development for the banks, while at the same time presenting a considerable risk. Accurate risk management in the banking and finance sector is related to efficient and optimized use of the current resources, assessment of possible risks and taking timely precautions. It is of utmost importance for the banks to predict the problematic loans in terms of long-term stability. Giving credits to the applicants is one of the fundamental activities of the banks, however; the same activity brings significant risks. As part of their founding purpose, the banks do not avoid taking risks, and they choose to manage them. The banks should perform their risk management in the way to keep the damages resulting from the amount of loans they give to a minimum. Considering the above and in order to speed up the lending procedures in banks while making advantageous decisions, different algorithmic models and classifications, machine learning techniques such as artificial neural networks were started to be used lately, data mining being at the first place. In this study, the accuracy of the applicants' eligibility status for loans was determined by making use of several machine learning techniques. The open-access dataset from the German Credit Data UCI was employed. Based on the 1000 customers in this study's dataset, a 75,60% success rate was achieved in the XGBoost classifier, which has the best success rate among the studies conducted with the XGBoost classifier previously. In addition, the success rate is the highest among the other algorithms used in various studies made.

Keywords: *Credit Risk Analysis, Machine Learning, Data Mining*

1. GİRİŞ

Toplumlarda kredi talep etme ve kullanma oranı son dönemlerde bir artış göstermesinden dolayı, finans merkezli kurum ve kuruluşlar kredi taleplerinin riskli olup-olmadığını analiz etmeye daha fazla yoğunlaşmaya ve önem vermeye başlamışlardır. Bu önem doğrultusunda kredi talebinde bulunan müşterilerin, kredi risk analizini daha iyi ve verimli hale getirmek için istatistiksel yöntemler ve makine öğrenmesi yöntemlerini kullanmayı tercih etmişlerdir. Kredi risk analizi, potansiyel riske sahip olan müşterileri önceden belirleyip, hızlı bir şekilde karar verme aşamasına gelmeyi amaçlar.

Kredi talebinde bulunan müşterilerin risk analizi 2 sınıfa ayrılabilir. Birincisi müşteri başvuru skorlama, diğeri ise müşteri davranış analizidir. Müşteri başvuru skorlama analizi, kredi talebinde bulunan müşterinin, krediye başvururken verdiği bilgiler ya da geçmişte bir kredi talebi veya kredi kullanma durumu var ise geçmiş bilgilerinden yararlanılarak, kredi talebinde bulunan müşterinin, kredi durumunun tahmin edilebilmesi amaçlanır. Diğer analiz yöntemi olan davranış analizin de ise, kredi talebinde bulunan kişinin belli bir zaman aralığındaki davranışları gözlemlenerek kişinin kredi ödemesinde problem yaşayıp-yaşamayacağını tahmin etmeye yarar. İki analiz arasındaki temel fark; birinci analiz yöntemi sabit bilgileri kullanarak tahmin etmeyi, ikinci analiz yöntemi ise belirli bir periyottaki davranışları baz alarak analiz yapmaktadır.

1.1 Literatür Araştırması

Literatür araştırmaları sonucunda kredi risk analizi makine öğrenmesi, istatistik teknikleri, veri madenciliği ve birçok teknik kullanılarak çalışmalar yapılmıştır. Hasan Tahsin Oğuz "Saklı Markov Modeli ile kredi risk analizi" adlı çalışmasında saklı markov modelini kullanarak kredi risk analizinin performansını ölçmek ve sınıflandırmak için çalışmasını yürütmüştür.

Gülnur Derelioğlu "KOBİ kredi risk analizinde modüler yaklaşım" adlı çalışmasıyla esnaf olan kişilerin işlerini büyütmek ve geliştirmek için kullandıkları KOBİ kredisini analiz edip, yorumlamıştır. Gül Efşan ve Bozkurt Gönen'in ortak çalışması olan "Öznitelik seçme ve transfer öğrenme algoritmaları ve kredi risk analizi üzerine uygulamaları" adlı çalışmasında öznitelik belirlemek için probit sınıflandırıcı ve çoklu çekirdek öğrenimini geliştirmişlerdir. Bu geliştirmeler sonucunda kredi risk analizi veri seti üzerinde kullanarak etkinliği verimliliğini ölçmektedir. Erkan Çetiner "Sınıflandırma tekniklerinin kredi risk analizi üzerindeki performansı" adlı çalışmasında kredi risk analizi için kullanılan sınıflandırma ölçütlerini geliştirerek yeni bir sınıflandırma yöntemi oluşturmayı amaçlamıştır. Amir E. Khandani, Adlar J. Kim, Andrew W. Lo ortak çalışması olan "Consumer credit-risk models via machine-learning algorithms" adlı çalışmada müşterilerin kredi taleplerini makine öğrenmesi algoritmaları kullanarak uygun olup olmadığını belirlemeye çalışmışlardır. Lean Yu, Shouyang Wang, Kin Keung Lai kişilerinin "Credit risk assessment with a multistage neural network ensemble learning approach" adlı çalışmasında çok aşamalı bir sinir ağı yapısı kullanarak kredi risk ölçümü yapmayı hedeflemişlerdir. Zhu, Li, Wu, Wang, Liang, "Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-Topsis classification approach" adlı çalışmasında krediyi doğruluk, yorumlanabilirlik ve karmaşıklığını ele alarak yeni bir sınıflandırma ortaya çıkarmışlardır. Li, Shiue, Huang "The evaluation of consumer loans using support vector machines" adlı çalışmasında kredi talebinde bulunan müşterilerin taleplerini değerlendirmek için destek vektör makineleri yöntemini kullanarak bir model oluşturmuşlardır. Huang, Chen, Wang "Credit Scoring with a Data Mining Approach Based on Support Vector Machines" adlı çalışmasında kredi talebinde bulunan kişilerin özniteliklerinden bir kredi puanı oluşturup değerlendirmek için hibrid destek vektör makineleri yöntemi kullanarak kredi puanlama modeli oluşturmuşlardır. Saha, Bose, Mahanti, "A knowledge based scheme for risk assesment in loan processing by banks" adlı çalışmasında kredilendirme aşamalarını denetimini sağlamak için bir önerme gerçekleştirilmiştir. Malhotra, K. Malhotra, "Evaluating consumer loans using neural networks", adlı çalışmasında kredi talebinde bulunan kişileri değerlendirmede çoklu diskriminant analizini ve yapay sinir ağları analizini ele alarak bu iki algoritmaların performanslarını karşılaştırmaktadır. Tsai, Lin, Cheng, Lin "The consumer loan default predicting model – An application of DEA-

DA and neural network" başlıklı çalışmasında tüketici kredisi alan kişilerin analizlerini çıkartarak, tüketici kredisi için başvuran kişiler için bir model oluşturmuşlardır.

1.2 Tezin Amacı

Bu çalışmada içerisinde 1000 veri bulunan German Credit Data UCI veri setinden yararlanılmıştır. Veri seti üzerinde makine öğrenmesi teknikleri kullanılarak kredi çekme talebinde bulunan müşterilerin aslında krediye uygun olup-olmadığını tahmin etmek planlanmıştır. Veri setinde bulunan 1000 kişilik veriden 300 kişisi risk teşkil etmektedir. Kişilerin cinsiyeti, yaşı, meslek grubu, birikim hesabındaki tutarı, kişinin evi olup-olmaması gibi kişilere özgü farklı değerler içermektedir. Bu çalışmada kredi almaya uygun kişiler 1, kredi almaya uygun olmayan kişiler 0 ile gösterilmektedir.

Krediye uygunluk durumunu değerlendirmek için toplam 10 adet öznitelik kullanılmıştır. Bu öznitelikleri toplam da 10 adet makine öğrenmesi yöntemi uygulanmıştır. Uygulanan yöntemlerin sonuçları karşılaştırılıp, en iyi sonucu veren algoritma ile kredi uygunluk tahmini yapılması amaçlanmıştır.

2. YÖNTEM

Bu bölümde, öncelikle makine öğrenmesi hakkında genel bilgilendirme yapılacaktır. Genel bilgilendirmeden sonra çalışmada kullanılan makine öğrenmesi yöntemleri ve makine öğrenmesi sınıflandırıcıları hakkında bilgiler verilecektir.

2.1 Makine Öğrenmesi

Bir bilgisayar programının, insan etkileşimi olmaksızın kendi kendine tecrübe yoluyla, verilen probleme çözüm üretmesini sağlayan veri analizi tekniğine makine öğrenmesi denir. Makine öğrenme algoritmaları, bir model olarak önceden belirlenmiş bir denkleme dayanmaksızın, verileri "öğrenmek" için hesaplama yöntemlerini kullanır. Öğrenme için mevcut örnek sayısı arttıkça performans artar.

Büyük verilerin artmasıyla, makine öğrenimi, birçok alandaki sorunları çözmek için anahtar bir teknik haline gelmiştir. Örneğin, Yüz tanıma, hareket algılama ve nesne algılama için görüntü işleme tekniği ve ses tanıma uygulamaları için doğal dil işleme tekniği kullanılmıştır. Makine öğrenme algoritmaları, girilen verilere uygun kalıplar bulur, daha iyi kararlar ve tahminler yapılmasına yardımcı olur. Tıbbi teşhis, ticari hisse senedi, enerji yükü tahmini ve daha pek çok konuda kritik kararlar almak için her gün kullanılırlar. Örneğin, medya siteleri, size şarkı veya film önermek için milyonlarca seçeneği elden geçirmekte makine öğrenmesine güvenmektedir.
Çok miktarda veri ve birçok değişken içeren karmaşık bir problemde bir formül veya denklem yoksa makine öğrenmesi kullanılabilir. Makine öğrenimi, birçok teknik kullanır. Birisi, önceden tanımlanmış ve gruplanmış girdi ve çıktı verileriyle bir modeli eğiten gözetimli öğrenme tekniğidir. Önceden tanımlanmamış ve gruplanmamış girdileri kullanarak bu verilerinde gizli kalıpları veya içsel yapılarını bulan öğrenme tekniğine denetlenmeyen öğrenme denir.

2.1.1 Denetimli Öğrenme

Denetlenen bir öğrenme algoritması, bilinen bir veri girişi kümesini ve verileri (çıktı) bilinen yanıtları alır ve yeni verilere yanıt için makul tahminler üretmek için bir modeli eğitir. Tahmin etmeye çalışılan çıktı için girilen veriler etiketlenmişse genellikle denetimli öğrenme yöntemi kullanılır. Denetlenen öğrenme, tahmini modeller geliştirmek için sınıflandırma ve regresyon tekniklerini kullanır.

Sınıflandırma teknikleri, örneğin, bir e-postanın spam olup olmadığı veya tümörün iyi huylu olup olmadığı gibi farklı konularda tahmin üretmektedir. Sınıflandırma modelleri, girdi verilerini kategorilere ayırır. Örnek olarak; medikal görüntüleme, konuşma-tanıma ve kredi puanlaması verilebilir. Veriler etiketlenebilir, kategorilenebilir veya belirli gruplara ayrılabilirse, sınıflandırma teknikleri kullanılır. Örneğin, el yazısı tanıma uygulamaları, harfleri ve sayıları tanımak için sınıflandırma tekniği kullanır.

Regresyon teknikleri sürekli değişim gösteren durumlarda verilen problem için yanıtları öngörür. Örnek olarak; sıcaklıktaki değişiklikler ve algoritmik ticaret verilebilir. Regresyon tekniklerini kullanabilmek için, ekipmanın arızalanmasına kadar geçen süre gibi bir veri aralığı ile çalışılması gerekmektedir. Konuyu biraz daha açmak için verilen bu örneğe bakılabilir. Klinisyenler, regresyon tekniklerini kullanarak bir kişinin bir yıl içinde kalp krizi geçirip geçirmeyeceğini tahmin edebilirler. Bunun için ellerinde daha önceki hastalarla ilgili yaş, kilo, boy ve kan basıncı gibi veriler olması ve bu hastaların bir yıl içinde kalp krizi geçirip geçirmediğini bilmeleri yeterli olacaktır. Klinisyenler bu mevcut verileri kullanarak regresyon tekniklerini yeni gelen bir kişinin bir yıl içinde kalp krizi geçirip geçirmeyeceğini tahmin edebilecek bir modele geliştirebilirler.

2.1.2 Denetimsiz Öğrenme

Denetimsiz öğrenme, veri içindeki gizli kalıpları veya gruplanmaları bulmak ve keşif amaçlı veri analizi için kullanılır. Etiketli girdiler olmadan sadece verilerinden oluşan kümelerden çıkarımlar yapar. Kümeleme, en yaygın denetlenmeyen öğrenme tekniğidir. Küme analizine örnek olarak; gen dizisi analizi, pazar araştırması ve nesne tanıma verilebilir. Konuyu daha anlaşılır hale getirmek için sıradaki örnek verilmiştir. Bir cep telefonu şirketi, müşterilerine daha iyi bir performans sağlamak için yeni telefon kuleleri inşa etmek istemektedir. Hangi alanlara inşa ederse daha

verimli bir sonuç alacağını öğrenmek istemektedir. Bu nedenle telefon kulelerini kullanma ihtimali olan insanların kümelerinin sayılarını tahmin etmek için makine öğrenimini kullanabilirler. Bir telefon aynı anda sadece bir kule ile bağlantılı olabilir. Bu nedenle eğer kümeleme algoritmaları kullanılırsa, müşteri kümeleri için sinyal alımını optimize edebilir ve hücre kulelerinin en iyi yerleşimini tasarlayıp bu şekilde kuleleri inşa edebilirler. Denetimli ve denetimsiz makine öğrenimi arasında seçim

yapma yönergeleri şunlardır: Eğer sıcaklık veya hisse senedi fiyatı gibi sürekli değişen etiketlenmiş değerler için sınıflandırma yapmak isteniyorsa denetlenmiş öğrenme kullanılır. Örnek olarak, video görüntülerinden otomobil markalarını tanımlamak verilebilir. Öte yandan verileri keşfetmek gerekiyorsa ve verilerin kümelere bölünmesi gibi iyi bir iç temsil bulmak isteniyorsa denetimsiz öğrenme kullanılır.

2.1.3 Yarı Denetimli Öğrenme

Yarı denetimli öğrenme, veri setinde eğitim verisi olarak kullanılacak ve test verisi olarak kullanılacak olan veriler arasında oluşturmak istenilen sınıflandırma modelinin, yeniden eğitime girmesi gibi kısıtlamaları kaldırmak amacı için görüş madenciliğinde oldukça kullanılmaktadır. Görüş sınıflandırmada, yarı denetimli öğrenme yöntemini Aue ve Gamon çalışmalarında temel olarak kullanmıştır.

2.1.4 Takviyeli Öğrenme

Takviyeli öğrenme, bir sistemin hedefe ulaşmasında doğru öğrenme yardımı ile doğru kararlar almasında yardımcı olur. Takviyeli öğrenme, oyun programlama da robotik yazılımlarda, hastalık teşhisi koyma gibi alanlarda yaygın olarak kullanılır.

2.1.5 Regresyon ve Sınıflandırma Yöntemleri

Değişkenlerin bir bağımlı ve bir bağımsız olması durumunda bağımlı değişkenin bağımsız değişken üzerinde fonksiyonu olması durumuna regresyon denir. Regresyon analizi, değişkenler arasında neden-sonuç ilişkisini bulmayı sağlayan analiz yöntemidir.

Sınıflandırma yöntemi, sınıfı belli olmayan verilerin sınıflandırıcı makine öğrenmesi yöntemleri kullanarak sınıflarını tahmin etmeye yarar. Sınıflandırma yöntemi de regresyon analizi gibi danışmanlı öğrenmedir.

2.1.6 Kümeleme Analizi

Kümeleme analizi veri setindeki bilgilerin birbirlerine yakınlık derecesin göre gruplara ayrılması işlemidir. Kümeleme analizinde amaç, henüz sınıflanmamış kümelerin, veri setindeki verilerin anlamlı bir şekilde alt kümelere ayrılmasıdır.

2.1.7 Öznitelik Seçimi / Çıkarımı

Veriler üzerinde yapılacak analizlerin yapılması için kaynak sayısını daha verimli hale getirmek için kullanılır. Veri seti içerisinde oluşturulacak sınıflandırmada

belirleyici olacak özellikler altkümesi olarak belirlenir veya belirleyici olacak özelliklerin birleşiminden yeni bir özellik oluşturulabilir.

2.1.8 Test Aşaması

Bir modelin makine öğrenmesi yardımıyla verilerin öğrenme kısmı bittiğinde öğrenilen model test edilmesi gerekir. Bu aşamanın amacı, öğrenilen modelin veri kümesi üzerindeki başarısını ölçmesidir. Test aşamasında, veri kümesinin eğitime sokulmayan %70'lik kısmı kullanılır. Bu aşamada, öğretilen modelin hiç karşılaşmadığı veriler için doğruluk oranı analiz edilmektedir.

2.1.9 Aşırı Öğrenme

Bazı durumda eğitime sokulan veri ile teste sokulan veriler arasındaki doğruluk oranı arasında çok büyük farklar olabilir. Bu durum eğitime giren verilerin her türlü durumu değerlendirip ezberlemesi ve test aşamasına giren verilerde de eğitime giren verilerin kopyasını aramasından dolayı oluşmaktadır. Bu durumun oluşmasına aşırı öğrenme denir. Aşırı öğrenmenin önüne geçmek için veri kümesi çeşitlendirilebilir.

2.1.10 Çapraz Doğrulama

Çapraz doğrulama yönteminde veri kümesi parçalara ayrılır ve ayrılan veriler farklı eğitim ve farklı test veri kümelerini oluşturur. Bu ayrılan veriler model üzerinde ayrı ayrı doğruluk oranları hesaplanır. Çıkan her sonuç toplanıp, aritmetik ortalaması alınır. Alınan aritmetik ortalama sonucu doğruluk oranı belirlenir. Veriler parçalanıp ayrı ayrı öğrenmeye ve teste girdiği için hem başarı oranı etkili olur hem de aşırı öğrenme yaşanmamasını sağlar.

2.1.11 Performans Değerlendirme

Yapılan modelin performans değerlendirmesi için genellikle karışıklık matrisi kullanılır. Karışıklık matrisine gerek duyulmasının sebebi de yapılan modelin kaç durumu doğru tahmin ettiği veya başarı oranının bilinmesi yeterli değildir.

2.1.11.1 Karışıklık Matrisi

Karışıklık matrisi, yapılan modelin doğruluk ve kesinlik oranını ölçmek için kullanılır. Bu matris veri kümesinde olan durumları ve yapılan modelin doğruluk sayısı ve hata sayısı tahminlerinin sayısını göstermektedir. Gösterilen tahmindeki duruma göre NxN boyutunda matris şeklinde sonuçları çıkartır. Şekil 2.1'de bir karışıklık matrisi örneği verilmiştir. Şekilde TruePositive olan kısım kullanılan

modelin doğru tahmin ettiği pozitif değere(risk durumu 1 olanların) sahip verilerin sayısını gösterir. TrueNegative olan kısım kullanılan modelin doğru tahmin ettiği negatif değere (risk durumu 0 olanların) sahip verilerin sayısını gösterir. FalsePositive olan kısım kullanılan modelin yanlış tahmin ettiği pozitif değere sahip verilerin sayısını gösterir. FalseNegative olan kısım kullanılan mo delin yanlış tahmin ettiği negatif değere sahip verilerin sayısını gösterir.

GERÇEK	ÖNGÖRÜLEN		
		Positive	Negative
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Şekil 2.1 : Karışıklık matrisi gösterimi

Karışıklık matrisindeki bu değerlere göre bazı oranlar çıkartılabilir.

- **Hassasiyet (precision):** Oluşan bütün sınıflarda toplam doğru tahminin ne kadar yapıldığını ölçer. Denklem 2.1'de verilmiştir. Bu denklemde TP truepositive değerleri, TN truenegative değerleri,FP falsepositive değerleri ve FN de falsenegative değerleri kullanarak hesaplanır.

$$\frac{TP}{TP + FP} \tag{2.1}$$

- **İsabet oranı (recall):** Modelde doğru tahmin yapılan toplam pozitif değerlerin ölçümüdür. İsabet oranı mümkün olduğunca yüksek çıkması modelin sonuçlarının iyi olduğunu gösterir. Denklem 2.2'de verilmiştir. Bu denklemde TP truepositive değerleri, TN truenegative değerleri, FP falsepositive değerleri ve FN de falsenegative değerleri kullanarak hesaplanır.

$$\frac{TP}{TP + FN} \tag{2.2}$$

- **Doğruluk oranı:** Kullanılan modelde yapılan doğru tahminin sıklığının ölçümüdür. Denklem 2.3'de verilmiştir. Bu denklemde TP truepositive değerleri, TN truenegative değerleri, FP falsepositive değerleri ve FN de falsenegative değerleri kullanarak hesaplanır.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

- **Yanlış sınıflandırma oranı:** Kullanılan modelde yapılan yanlış tahminin sıklığının ölçümüdür. Denklem 2.4'de verilmiştir. Bu denklemde TP truepositive değerleri, TN truenegative değerleri, FP falsepositive değerleri ve FN de falsenegative değerleri kullanarak hesaplanır.

$$\frac{FP + FN}{TP + TN + FP + FN} \tag{2.4}$$

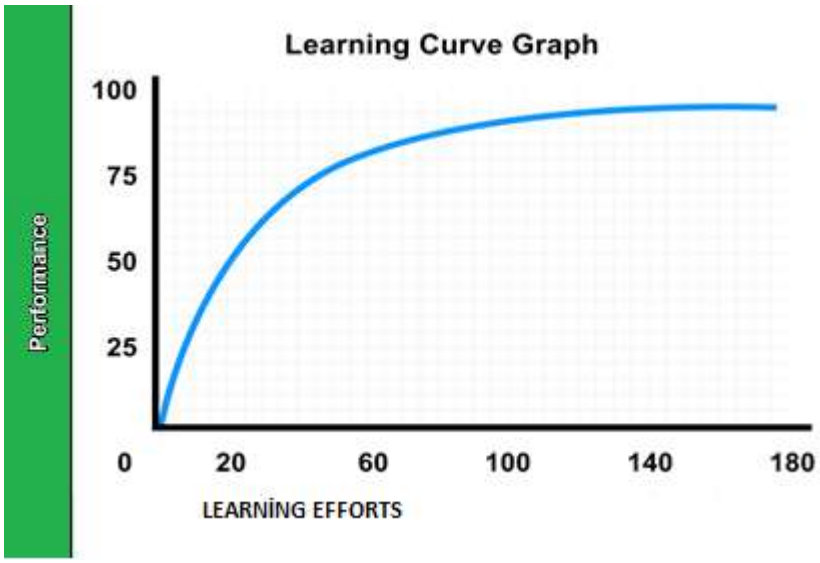
2.1.11.2 F1- skor Ölçümü

F1-skor ölçümü, test aşamasından çıkan verilerin doğruluğunu değerlendirmek için kullanılmaktadır. F-1 skorun hesaplaması denklem 2.5'de verilmiştir. İsabet oranı (recall) ile hassasiyetin(precision) çarpımının 2 katının, hassasiyet oranının (precision) isabet oranı ile toplamına bölümünden çıkartılır.

$$\frac{2 \cdot Hassasiyet \cdot \text{İ}sabet\ Oranı}{Hassasiyet + \text{İ}sabet\ Oranı} \tag{2.5}$$

2.1.11.3 Öğrenme Eğrisi

Öğrenme eğrisi, genellikle büyük veri kümeleri üzerinde hastalık teşhisi olarak kullanılır. Verilerin eğitim aşamasında belirli aralıklarda değer ölçümü yapılarak kullanılan modelin eğitim aşaması ve test aşamalarını grafik olarak gösterir. Şekil 2.2'de örnek olarak bir modelin öğrenme eğrisi verilmiştir.



Şekil 2.2 : Öğrenme eğrisi örneği

Kaynak: (valamis.com, 2020)

2.2 Makine Öğrenmesinin Banka ve Finans Sektöründeki Önemi

Makine Öğrenmesinin banka ve finans sektöründe önemi çok büyüktür. Çünkü geleceğe dair tahminler yapmada, yatırım planlanmasında, enflasyon ve kredi tahminlerinde ve piyasadaki dalgalanmaların önceden belirlenmesine veya tahmin edilmesine, geçmişte kullanılan bilgiler doğrultusunda geleceğe dair bir öngörü sağlar. Özellikle makine öğrenmesi büyük veriler üzerinde çalışıp yüksek sonuçlar alınabilen algoritmalara sahiptir. Banka ve finans sektöründe de elde mevcut halde bulunan verilerin çok büyük olması da makine öğrenmesinin tercih edilmesinde olanak sağlamaktadır. Bankalar açısından sorun teşkil edebilecek, ödeme zorlukları yaşayabilecek müşterilerin belirlenmesinde ve bu konuda sistem üzerinden uyarı yapılmasına olanak sağlayacaktır. Geçmişe dayalı verileri kullanarak sağlıklı sonuçlar alınmasına katkı sağlayacaktır. Aynı zamanda kredi talebinde bulunan müşterilere hızlı bir şekilde geri dönüş yapılacağından dolayı, bankalar ve müşteriler için zamandan da tasarruf sağlanmış olacaktır.

2.3 Kullanılan Makine Öğrenmesi Yöntemleri

Bu bölümde yapılan çalışmada kullanılmış olan makine öğrenmeleri yöntemleri hakkında bilgi verilecektir.

2.3.1 Lojistik Regresyon

Lojistik Regresyon, genellikle kategori halinde olan verilerin sınıflandırılması için kullanılmaktadır. Lojistik Regresyon yönteminde, bağımsız olan değişkenin veya değişkenlerin, sonuç çıktısı olan değişkenler ile ilişkisini hesaplamak için kullanılır. Eğer sonuç değişkeninin iki olasılıklı sonucu var ise, ikili lojistik regresyon analizi uygulanır. Bu modelde, bağımsız ile bağımlı değişkenler arasındaki ilişkiyi en az değişken kullanarak, değişkenler arası ilişkiyi en iyi duruma gelecek şekilde oluşturulmak ve kabul edilebilir bir model haline getirmek amacıyla tasarlanmıştır. Denklem 2.6'da Lojistik regresyon formülü verilmiştir. Bu formülde s, bağımsız olan x değişkeninin -∞ ile +∞ değerler arasında değer alan doğrusal işlevidir.

$$f(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}}$$

(2.6)

2.3.2 Lineer Diskriminant Analizi

Lineer Diskriminant Analizi, bir bağımlı değişkeni diğer özelliklerin veya ölçümlerin doğrusal bir kombinasyonu olarak ifade etmeye çalışan varyans analizi (ANOVA) ve regresyon analizi ile yakından ilgilidir. LDA, koşullu olasılık yoğunluğunun çalıştığını varsayarak soruna yaklaşır. Lineer diskriminant analizinde amaç verilerin doğru sınıflandırılmasını en az hata ile atamaktır. Denklem 2.7'de Lineer diskriminant analizin formülü verilmiştir.

$$c = \frac{1}{2}(T - \mu_0^T \sum -1\, \mu_0 + \mu_1^T \sum -1 \mu_1$$

(2.7)

2.3.3 K-En Yakın Komşu

Verilerin sınıflandırmasında ve regresyon analizi sırasında oluşan sıkıntılar için kullanılır. En yakın komşu algoritmasında veriler eğitime girmezler. Bundan dolayı büyük veri kümelerinde kullanımı sağlıklı değildir. Veri kümesinde sınıflandırılması planlanan her değer için, o değere en yakın uzaklıktaki x adet örnekler baz alınır. En yakın uzaklıktaki x tane örnek var olan sınıflar içerisinde en fazla bulunuyorsa yeni sınıflandırılacak değer, içerisinde en fazla bulunan sınıfa girer. Denklem 2.8'de en yakın komşu değerini bulmak için kullanılan Öklid mesafe hesaplaması formülü

kullanılır. Bu formülde noktalar arasında değerler ayrı ayrı bulunup karesi alındıktan sonra çıkan sonuçlar toplanır. Toplam sonucun karekökü alınır. Çıkan sonuç yakınlığı gösterir.

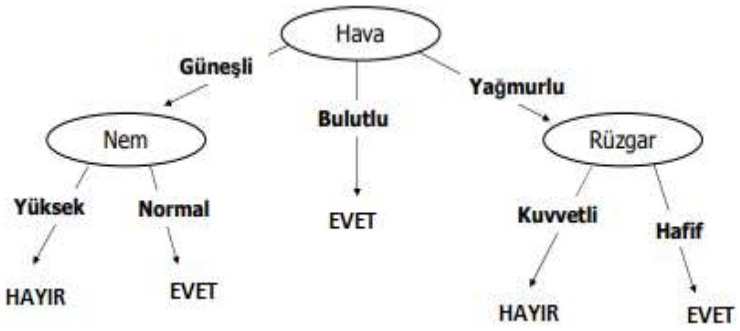
$$d(i,j) = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \cdots + \left|x_{ip} - x_{jp}\right|^2}$$

2.8)

2.3.4 Karar Ağacı

Karar ağacı modeli genellikle veri madenciliği alanında kullanılır. Karar ağacı yönteminin amacı veri kümesinde bulunan verileri sınıflandırma işlemi yapmaktır. Bu amaç doğrultusunda, veri kümesi içerisindeki öznitelikler düğümleri oluşturmaktadır. Bu düğümler, belirlenen kriterlere göre ikiye ayrılır. Ayrılma işleminden sonra, özellik vektörleri değerlendirilir ve en yüksek sonuca ulaştıran düğüm dallanma işlemi yapar. Bu işlem bütün verinin sınıflandırılmasına kadar tekrar eder. Karar ağacının yaprakları da sınıf etiketlerini oluşturmaktadır.

Şekil 2.3'de örnek karar ağacı modeli verilmiştir. Bu modelde hava durumunun nasıl olduğu ile alakalı dallanma gösterilmiştir.



Şekil 2.3 : Karar Ağacı Model Örneği

Karar ağacı modelinde öncelikle entropi hesabı yapılır. Entropi, öngörülmeyen durumların ve belirsiz durumların olma olasılığını hesaba katar. Denklem 2.9'da entropi hesaplaması formülü verilmiştir. Bu denklemde P(x) değişkeni herhangi bir sınıfa ait verilerin yüzdesi, H değişkeni ise entropi hesabını gösterir.

$$H = -\sum P(x) log P(x)$$

(2.9)

Karar ağacı modelinde, çıkan entropi değeri sonucundan sonra bilgi kazancı hesaplanır. Denklem 2.10'da bilgi kazancının formülü verilmiştir. Bu formüle göre seçilen D özelliğinin S orijinal veri seti için bilgi kazancının sonucunu verir. Bilgi kazancının en yüksek çıktığı özellik seçilir ve bu özellik üzerinden dallanma gerçekleştirilir.

$$Gain(S, D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V)$$

(2.10)

2.3.5 Naive Bayes

Naive Bayes algoritması, veri kümesi içerisindeki değerlerin sık kullanımlarını ve oluşabilecek kombinasyonları sayarak olasılık hesaplayan sınıflandırıcıdır. Bu yöntem büyük veriler ile kullanımı etkilidir. Naive Bayes algoritmasında eğitim aşaması yoktur. Verileri sınıflandırmak için bağımlı değişken ile bağımsız değişken arasındaki durumu inceler. Bu yöntem bayes teoremini temel olarak alır. Denklem 2.11'de bayes teoremi formülü gösterilmiştir. Bu formülde C_j veri seti içerisindeki sınıf sayısıdır. x/C_j ifadesi j sınıfında olan durumun x olma olasılığıdır. C_j/x de tam tersi x olan durumun j sınıfında olma olasılığıdır. $P(C_j)$ j sınıfının olasılığıdır. $P(x)$ de örneğin x olma olasılığıdır.

$$P\left(\frac{C_j}{x}\right) = \frac{P\left(\frac{x}{C_j}\right) \cdot P(C_j)}{P(x)}$$

(2.11)

Naive Bayes yöntemi ile sınıflandırma işlemi yapıldığında denklem 2.12'deki eşitsizlik hesaplanır. Oluşan her sınıf için denklem 2.12'deki eşitlik uygulanır ve olasılık hesaplanmış olur. En yüksek sonuca sahip sınıf belirlenmiş olur.

$$\mathrm{Y}' \leftarrow \mathrm{argmax}_{\mathrm{yj}} \frac{P(y_j)\prod_{i=1}^{m} P(x = x_i|y_i)}{\sum_{j=1}^{j} P(y_j)\prod_{i=1}^{m} P(x = x_i|y_i)}$$

(2.12)

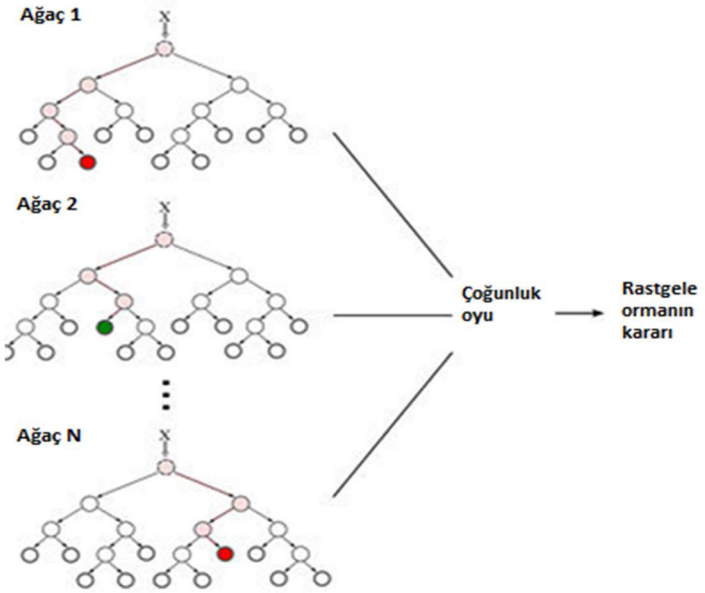
2.3.6 Rastgele Orman

Rastgele orman algoritması, eğitim verisi ve test verisini karar ağacı modeliyle uygulayarak iyi sonuçlar almayı sağlar. Rastgele orman algoritması, sınıflandırma ve regresyon analizi için kullanılabilir. Diğer sınıflandırma ve regresyon modellerine göre daha iyi sonuçlar çıkartabilen bir modeldir. Bunun nedeni ağaç sayısını istenilen şekilde belirlenmesi ve en değerli özniteliği belirleme de kullanılabilir olmasıdır.

Rastgele Orman algoritmasında, değişken sayısını ve ağaç sayısını kullanıcıdan alınır. Parametreler alındıktan sonra veri kümesinin 2/3 lük kısmı eğitim alınır ve öğrenme için kullanılır. Geriye kalan 1/3 lük kesim de test aşaması için kullanılır. Oluşturulan her düğüm için ayrı ayrı m değeri karışık bir şekilde seçilir ve bu seçilen değerler arasında en iyi sonucu veren dal belirlenir. Bu durum için GINI dizini kullanılır. Denklem 2.13'de P_j her veri için kendisinden küçük ve büyük sayıların bölümünün karesini, n ise seçili olan veriyi gösterir.

$$GINI(T)= 1- \sum_{j=1}^{n} (P_j)^2 \qquad \textbf{(2.13)}$$

Şekil 2.4'de Karar ağaçlarından rastgele orman oluşturulmasının şeması verilmiştir.



Şekil 2.4 : Karar Ağaçlarından Rastgele Orman Oluşturulmasının Şeması

2.3.7 Destek Vektör Makineleri

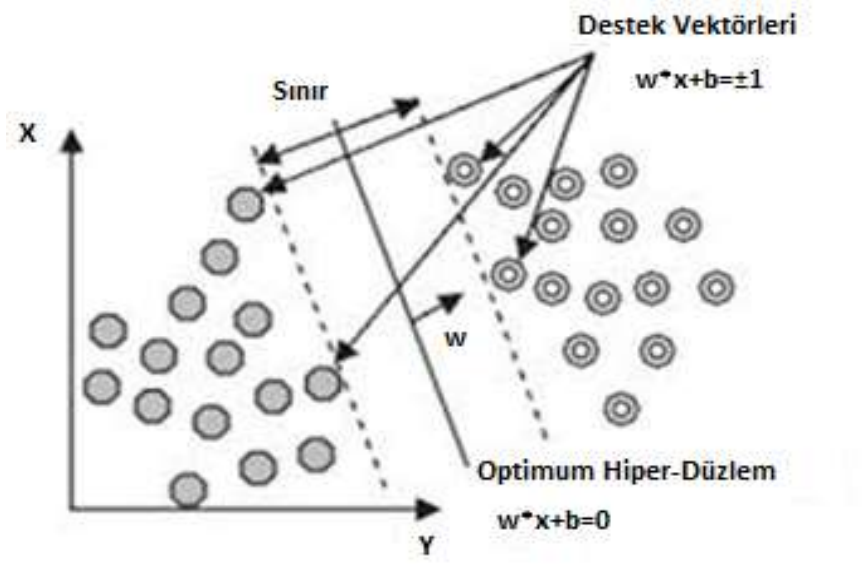
Destek vektör makineleri(DVM), verileri birbirinden ayırmak için kullanılan algoritmadır. Sınıflandırma işlemi yaparak ayırma işlemini gerçekleştirir. Bir düzlemde yer alan örneklerin birbirleri arasına sınır çizilir. Diğer algoritmalardan ayıran özellik, sınıflandırma sırasında oluşan problemi kareli optimizasyona çevirip, problemi çözmesidir. Bu durumdan kaynaklı diğer yöntemlere göre daha hızlı sonuç alınabilir. Bu sebepten dolayı büyük veri kümelerinde kullanımı elverişlidir. Destek vektör makineleri, doğrusal destek vektör makineleri ve doğrusal olmayan destek vektör makineleri olarak 2 gruba ayrılır.

2.3.7.1 Doğrusal destek vektör makineleri

Doğrusal destek vektör makinelerinin denklemi 2.14'de verilmiştir. Bu denklemde n veriden oluşan veri setinin $X = \{x_i, y_i\}, i = 1,2, \dots, n$ olduğu düşünülsün. $y_i \in \{-1,1\}$ etiket değerleri, $x_i \in \Re^d$ özellikler vektörüdür, w ağırlığı, x verileri, b ise eğitim terimi olarak kullanılır. Bu terimlerin değeri hiper-düzlemin konumunu gösterir.

$$F(x) = w^T.x + b = \sum_{i=1}^{n} w_i.x_i + b \qquad \textbf{(2.14)}$$

Şekil 2.5'te doğrusal destek vektör makinelerinin ayrılabilme durumu örnek olarak gösterilmiştir.



Şekil 2.5 : Doğrusal Ayrılabilmede Optimum Hiper-Düzlem ve Destek Vektörleri

2.3.7.2 Doğrusal olmayan destek vektör makineleri

Verilerin çoğu doğrusal şekilde ayrılmaya uygun değildir. Bu durumda doğrusal olmayan destek vektör makineleri kullanılmaktadır. Çekirdek fonksiyonu verileri kendi bünyesinden geçirerek özellik uzayına aktarır. Denklem 2.15, 2.16 ve 2.17'de çekirdek fonksiyonlarının formülleri verilmiştir.

Radyal tabanlı çekirdek fonksiyonu (RBF)

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$

(2.15)

Polinom çekirdek fonksyonu

$$K(x_i, x_j) = (x_i^T . x_j)^d$$

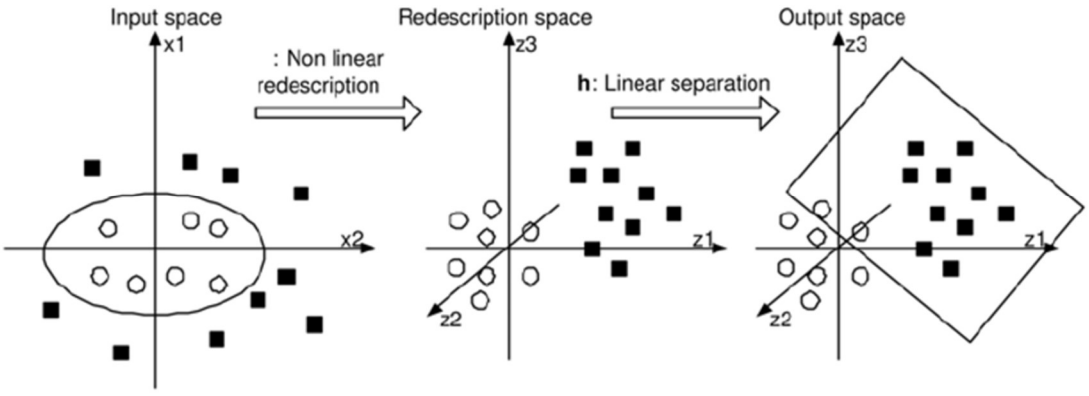
(2.16)

Doğrusal çekirdek fonksiyonu

$$K(x_i, x_j) = (x_i^T . x_j)$$

(2.17)

Şekil 2.6'da lineer olmayan sınıflandırma grafiği ve uzayda lineer düzlem ile ayrılması gösterilmiştir.



Şekil 2.6 : Doğrusal Olmayan Destek Vektör Makineleri Doğrusal Ayırma Gösterimi

2.3.8 Extreme Gradient Boosting(XGBoost)

XGBoost, Tianqi Chen tarafından bir araştırma projesi olarak adı duyulmuştur. İlk olarak bir libsvm yapılandırma dosyası kullanılarak yapılandırılabilen bir terminal uygulaması olarak kullanılmıştır. Higgs Machine Learning Challenge'ın kazanan çözümünde kullanıldıktan sonra ML yarışma çevrelerinde tanınmaya başladı. Kısa bir süre sonra Python ve R paketleri oluşturuldu ve XGBoost şimdi Julia, Scala , Java ve diğer diller için paket uygulamalarına sahiptir.

XGBoost, kaynakları doğru ve verimli bir şekilde kullanmak ve önceki gradient boosting kısıtlamalarından kurtulmak için oluşturulmuştur. Regresyon, Sınıflandırma ve Sıralama gibi denetimli öğrenme görevleri için kullanılabilir. XGBoost, Gradient Boosting konseptinin uygulamalarından biridir, ancak XGBoost'u benzersiz kılan şey, algoritmanın yazarı Tianqi Chen'e göre " aşırı uydurmayı kontrol etmek için daha düzenli bir model formalizasyonu " kullanmasıdır. XGBoost kütüphanesi R kullanıcıları arasında oldukça popüler kullanılmaktadır. Diğer algoritmalara kıyasla çok daha iyi tahmin performansı üretmekte ve aynı zamanda görevleri hızlı bir şekilde tamamlamaktadır. Xgboost paketinin yazarlarına göre (Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang) mevcut gradyan artırma paketlerinden 10 kat daha hızlı olabilen tek bir makinede otomatik olarak paralel hesaplama yapar.

XGBoost algoritmasında, karar ağaçları sıralı olarak oluşturulur. XGBoost'ta ağırlıklar önemli bir rol oynar. Ağırlıklar, tüm bağımsız değişkenlere atanır ve daha sonra sonuçları tahmin eden karar ağacına beslenir. Ağaç tarafından yanlış tahmin edilen değişkenlerin ağırlığı artırılır ve bu değişkenler daha sonra ikinci karar ağacına beslenir. Bu bireysel sınıflandırıcılar daha sonra güçlü ve daha kesin bir model vermek için toplanır.

2.3.9 Gradient Boosting

Gradient Boosting algoritması, regresyon ve sınıflandırma problemleri için, genellikle karar ağaçları olan zayıf tahmin modelleri topluluğu şeklinde bir tahmin modeli üreten bir makine öğrenme tekniğidir. Model, diğer arttırıcı yöntemlerin yaptığı gibi aşamalı olarak inşa eder ve keyfi farklılaşabilir bir kayıp fonksiyonunun optimizasyonuna izin vererek onları genelleştirir. Gradyan yükseltme fikri, Leo Breiman tarafından artırmanın uygun bir maliyet fonksiyonu

üzerinde bir optimizasyon algoritması olarak yorumlanabileceği gözlemi sonucu ortaya çıkmıştır. Yani, negatif gradyan yönünü gösteren bir işlevi (zayıf hipotez) yinelemeli olarak seçerek maliyet fonksiyonunu işlev alanı üzerinde optimize eden algoritmalardır. Güçlendirmenin bu işlevsel gradyan görünümü, makine öğrenimi ve istatistiklerin pek çok alanında regresyon ve sınıflamanın ötesinde güçlendirme algoritmalarının geliştirilmesine yol açmıştır.

2.3.10 Adaptive Boosting (ADA Boosting)

Adaptive Boosting(ADABoost), Performansı artırmak için diğer birçok öğrenme algoritmasıyla birlikte kullanılabilir. Diğer öğrenme algoritmalarının çıktısı, güçlendirilmiş sınıflandırıcının nihai çıktısını temsil eden ağırlıklı bir toplamda birleştirilir. AdaBoost, sınıflandırıcılar tarafından yanlış sınıflandırılan örnekler lehine ayarlanması anlamında uyarlanabilir. AdaBoost aykırı değerlere duyarlıdır. Bazı problemlerde, aşırı öğrenme roblemine karşı diğer öğrenme algoritmalarına göre daha az duyarlı olabilir. Her öğrenme algoritması bazı sorun türlerine diğerlerinden daha iyi uyma eğilimindedir ve genellikle bir veri kümesinde en iyi performansı elde etmeden önce ayarlamak için birçok farklı parametre ve yapılandırmaya sahiptir. AdaBoost genellikle en iyi çıkış olarak adlandırılır. AdaBoost algoritmasının her aşamasında, her eğitim örneğinin göreceli sertliği hakkında toplanan bilgiler, daha sonraki ağaçlar daha sert odaklanma eğilimi gösterecek şekilde ağaç yetiştirme algoritmasına beslenir. Örnekleri sınıflandırır.

3. BULGULAR

Bu bölümde, yapılan çalışma kapsamında elde edilen analiz ve sonuçlar hakkında detaylı bir bilgi verilecektir. Öncelikle kullanılan veri seti ve veri seti içindeki öznitelik alanları gösterilecektir. Veri seti içerisindeki verilerin birbirleri ile ilişkileri karşılaştırılacaktır. Ardından veri setine sırasıyla algoritmalar uygulanacaktır. Lojistik regresyon, lineer diskriminant analizi, k-en yakın komşu analizi, karar ağacı analizi, naive bayes, rastgele orman, destek vektör makineleri, extreme gradient boosting, gradient boosting ve adaptive boosting uygulanacaktır.

3.1 Veri Seti

Bu çalışmada, 1000 kişinin çeşitli özelliklerini barındıran german credit data UCI veri seti kullanılmıştır. Çalışmada kullanılan grafikler için Numpy, Pandas, Matplotlib, Seaborn kütüphaneleri, çalışmada kullanılan algoritmalar için ise Scikit-Learn kütüphanesi kullanılmıştır. 1000 adet verinin 3/4(%75) eğitime girmiştir, kalan 1/4(%25) veri ise test için kullanılmıştır. Tablo 3.1'de veri seti içerisinde bulunan alanların ve bu alanların açıklamaları verilmiştir.

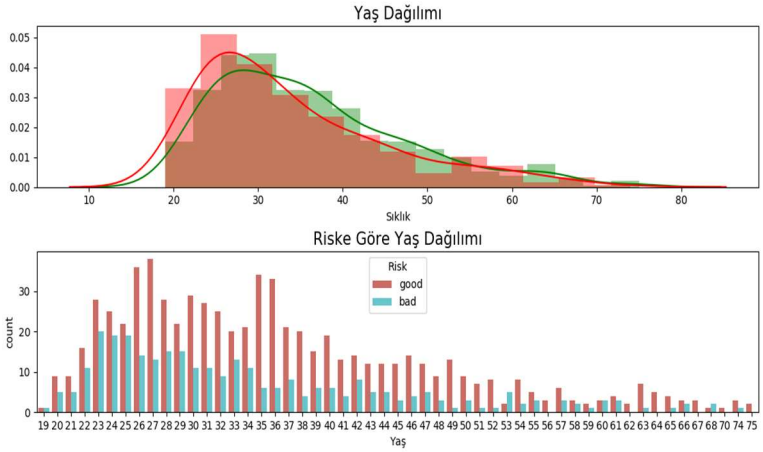
Çizelge 3.1 : Veri seti içerisindeki alanlar ve açıklamaları

Veri Seti İçerisindeki Alan	**Açıklaması**
Age	Yaş
Sex	Cinsiyet
Job	Meslek Grubu
Housing	Ev Durumu(Kendinin, Kira)
Saving Accounts	Birikim Hesapları
Checking Accounts	Vadesiz Hesaplar
Credit Amount	Kredi Miktarı(Dolar)
Duration	Vade(Ay)
Purpose	Kredi amacı
Risk	Risk durumu(1 yada 0)

3.2 Veri Seti İçerisindeki Alanların Karşılaştırılması

Veri seti içerisinde yer alan alanlar, doğrudan risk durumunun analiz edilmesinde önemli rol oynar. Kredi talebinde bulunan kişilerin yaşı, meslek grubu, evi olup-olmaması gibi faktörler kişilerin kredilerini geri ödemelerinde zorluk yaşayıp-

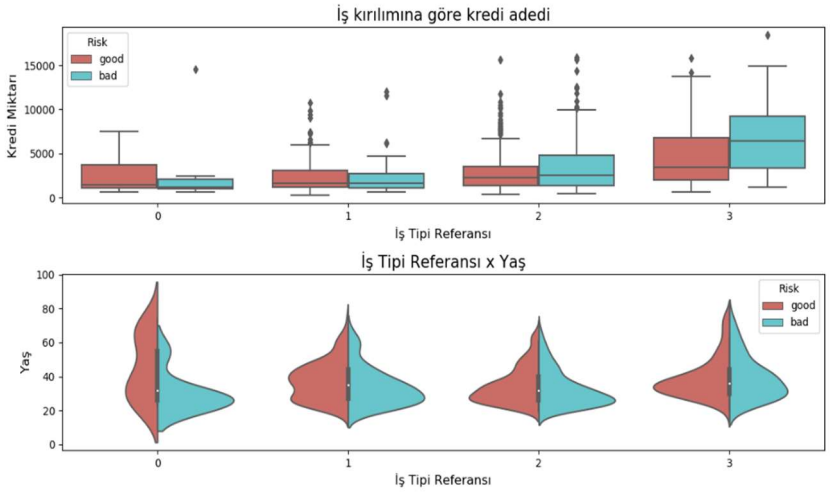
yaşamayacağının tahmin edilmesinde yardımcı olur. Şekil 3.1'de ilk tabloda veri seti içerisinde yer alan kişilerin yaşları ile frekans analizi yapılmıştır. İkinci tablo da ise risk durumu iyi yada kötü olan kişilerin yaşları ile ilişkisi gösterilmiştir. Risk durumu kötü olan yaş olarak en yüksek sonuç 23 yaş olduğu gözlemlenmiştir. Risk durumu iyi olan yaş olarak en yüksek sonuç ise 27 olarak görülmüştür.



Şekil 3.1 : Veriler içerisinde yaş dağılımı ve riske göre yaş risk dağılımı

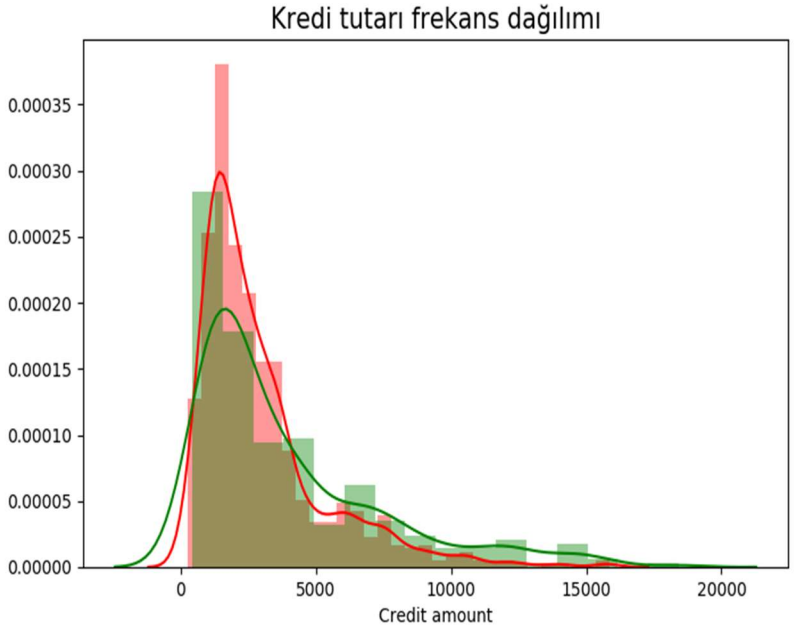
Şekil 3.2'deki ilk tabloda kredi talebinde bulunan kişilerin meslek grubu ile talep ettiği kredi miktarı arasındaki ilişki verilmiştir. Meslek grubunda "0" olarak ifade edilen grup, deneyimsiz ve geçici bir işte çalışan kişilerin oluşturduğu gruptur. "1" ile gösterilen grup ise deneyimsiz ama kalıcı bir işte çalışan kişilerin oluşturduğu gruptur. "2" olarak gösterilen meslek grubu deneyimli kişileri, "3" olarak gösterilen kişiler ise yüksek deneyimi olan kişileri göstermektedir. Bu durumlar göz önüne alındığında, birinci tabloda "0" meslek grubunda yer alan kişiler az miktarda kredi talebinde bulunmuşlardır. Bu kişilere de bakıldığında risk durumu genellikle iyi kişilerdir. Meslek grubu "1" olan kişiler risk durumu dengeli olarak gözükmektedir. Yani kredi talebinde bulunan kişiler, kredilerini yüzde elli şekilde ödeme ihtimali vardır. Meslek grubu "2" olan kişiler gözlemlendiğinde, risk durumu kötü olan kişiler talepte bulunduğu miktara oranla risk durumu iyi olan kişilerden fazladır. Bu da meslek grubu "2" olan kişilerin kredi talepleri bir daha değerlendirilmesi gerektiğini göstermektedir. Son olarak meslek grubu "3" olan kişilerin de risk durumu kötü olan kişiler talepte bulunduğu miktara oranla risk durumu iyi olan kişilerden fazladır. Aynı zamanda meslek grubu"3" olan kişilerin talepte bulunduğu kredi miktarları da diğer meslek gruplarına göre oldukça fazladır.

İkinci tabloya bakıldığında kişilerin meslek grupları ile yaşları arasındaki ilişki gösterilmiştir. Bu tabloya bakıldığında meslek grubu "0" olan kişilerin 20 ile 30 yaş aralığında oldukça risk teşkil ettiği görülmüştür. Aynı zamanda yaş grubu 60 ile 80 arasındaki kişilerin ise hiç sorun teşkil etmeyeceği gözlenmiştir. Yaş aralığı 40 ile 60 arasında olan kişilerin ise ortalama risk teşkil ettiği görülmüştür. Meslek grubu "1", yaş aralığı 20 ile 40 olan kişilerin hem yüksek şekilde risk teşkil ettiği hem de bir o kadar da sorun teşkil etmediği gözlemlenmiştir. Yaş aralığı 40 ile 60 arasında olan kişiler çoğunlukla iyi durumda ve 60 ile 80 yaş aralığında olan kişiler ise risk olarak ortalamanın üzerinde iyi durumdadır. Meslek grubu "2", yaş aralığı 20 ile 40 arasında olan kişilerin hem yüksek şekilde risk teşkil ettiği hem de bir o kadar da sorun teşkil etmediği gözlemlenmiştir. Yaş aralığı 40 ile 60 arasında olan kişiler çoğunlukla iyi durumda ve 60 ile 80 yaş aralığında olan kişiler ise risk olarak ortalamanın üzerinde iyi durumdadır. Meslek grubu "3", yaş aralığı 20 ile 40 arasında olan kişilerin risk durumu ortalamanın üzerinde iyi durumdadır. Yaş aralığı 40 ile 60 arasında olan kişiler risk teşkil etmekte olduğu görülmüştür. Aynı zamanda 60 ile 80 yaş aralığında olan kişiler ise risk olarak ortalamanın üzerinde kötü durumdadır.



Şekil 3.2 : Veriler içerisinde meslek grubu ile kredi miktarı ve yaş risk dağılımı

Şekil 3.3'de Veri seti içerisinde yer alan kredi miktarlarının frekans dağılımı gösterilmiştir. Frekans dağılımı, verilerin tekrar sayılarını gösterir. Frekans gösteriminde en çok tekrar eden tutar aralığı 0 ile 5000 arası olarak gözükmektedir. Buda demektir ki en fazla talepte bulunulan kredi tutarı 0 ile 5000 arasındadır.



Şekil 3.3 : Veriler içerisinde kredi miktarının frekans dağılımı

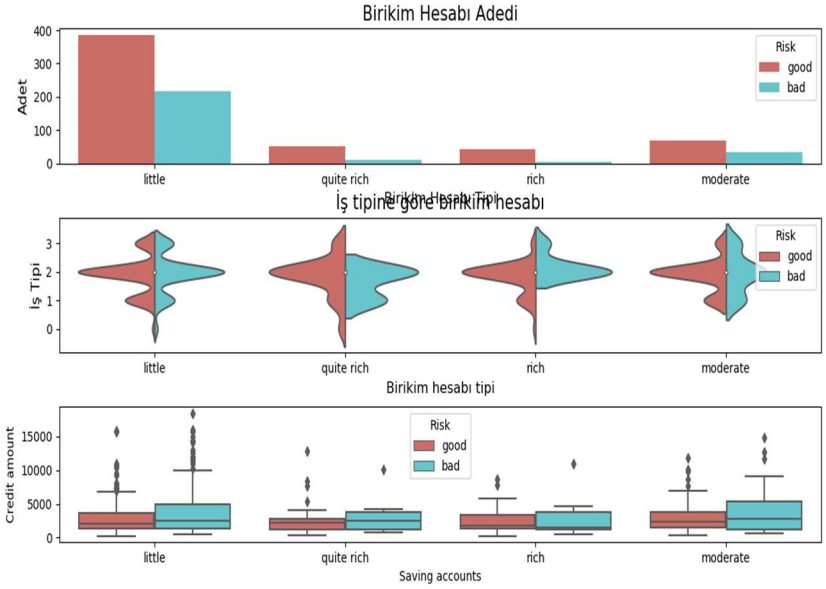
Şekil 3.4'de birikim hesabının meslek grubu ve kredi tutarı dağılımı verilmiştir. Birinci tabloya bakıldığında birikim hesabı olan kişilerin hesaplarındaki tutar ile sınıflandırılması gösterilmiştir. Birikim hesabı 4 nitelik ile belirtilmiştir. Bunlar hesaptaki tutarın, az olması, oldukça zengin olması, zengin olması ve ortalama olması olarak belirlenmiştir. Birikim hesabındaki tutar az olan kişilerin risk durumu ortalamanın üzerinde iyi durumdadır. Ama sorun teşkil edebilecek kişiler de azımsanmayacak kadar fazladır. Oldukça zengin olan kişilerin risk durumu iyi olarak gözükmektedir. Aynı şekilde zengin olan kişilerin de risk durumu iyi olarak gözükmektedir. Ortalama durumu olan kişilerin risk durumu iyi olmakla birlikte, risk durumu kötü olan kişilerde iyi olan kişilerin yarısı kadardır.

İkinci tabloda meslek grubu ile birikim hesabı durumu değerlendirilmiştir. Meslek grubu "0" olan kişiler ile birikim hesabında az tutar olan kişilerin risk durumu kötü gözükmektedir. Meslek grubu "1" olan kişiler ile birikim hesabında az tutar olan kişilerin risk durumu değerlendirildiğinde iyi olarak belirlenmiştir. Meslek grubu "2" olan kişiler ile birikim hesabında az tutar olan kişilerin sayısı oldukça fazladır. İyi durumda olanların sayısı fazladır ama kötü durumda olanlar da iyi olan kişilere yakındır. Meslek grubu "3" olan kişiler ile birikim hesabında az tutar olan kişiler arasındaki ilişki bakıldığında da risk durumu olarak kötü durum, iyi durumdan fazladır. Meslek grubu "0" olan kişiler ile birikim hesabına göre oldukça zengin olan

kişiler arasındaki risk durumu iyi gözükmektedir. Meslek grubu “1” olan kişiler ile birikim hesabına göre oldukça zengin olan kişiler değerlendirildiğinde risk durumu kötü olarak belirlenmiştir. Meslek grubu “2” olan kişiler ile birikim hesabına göre oldukça zengin olan kişiler arasındaki ilişkiye bakıldığında risk durumu iyi gözükmektedir. Ama risk teşkil eden kişilerin sayısı da azımsanmayacak şekilde fazladır. Meslek grubu “3” olan kişiler ile birikim hesabına göre oldukça zengin olan kişiler arasındaki ilişki bakıldığında da risk durumu oldukça iyi durumdadır. Meslek grubu “0” olan kişiler ile birikim hesabına göre zengin olan kişiler arasındaki risk durumu oldukça iyi gözükmektedir. Meslek grubu “1” olan kişiler ile birikim hesabına göre zengin olan kişiler değerlendirildiğinde de risk durumu oldukça iyi olarak belirlenmiştir. Meslek grubu “2” olan kişiler ile birikim hesabına göre zengin olan kişiler arasındaki ilişkiye bakıldığında risk durumu ortalama olarak gözükmektedir. Hem risk teşkil eden hem de iyi durumda olan kişilerin sayısı fazladır. Meslek grubu “3” olan kişiler ile birikim hesabına göre zengin olan kişiler arasındaki ilişki bakıldığında da risk durumu çok kötü durumdadır. Meslek grubu “0” olan kişiler ile birikim hesabına göre orta derecede olan kişiler arasındaki risk durumu incelendiğinde risk durumu kötü sonuç vermiştir. Meslek grubu “1” olan kişiler ile birikim hesabına göre orta derecede olan kişiler değerlendirildiğinde risk durumu ortalama olarak hem iyi hem de kötü sonuç olarak belirlenmiştir. Meslek grubu “2” olan kişiler ile birikim hesabına göre orta derecede olan kişiler arasındaki ilişkiye bakıldığında risk durumu ortalamanın üzerinde iyi olarak gözükmektedir. Hem risk teşkil eden hem de iyi durumda olan kişilerin sayısı fazladır. Meslek grubu “3” olan kişiler ile birikim hesabına göre orta derecede olan kişiler arasındaki ilişki bakıldığında da risk durumu kötü durumdadır. Risk durumu kötü olan kişilerin sayısı, risk durumu iyi olan kişilerin sayısında fazladır.

Üçüncü tabloya bakıldığında ise kişilerin birikim hesabı ile kredi tutarları arasındaki grafik gösterilmiştir. Bu grafiğe göre birikim hesabında az tutar olan kişilerin talepte bulunduğu kredi tutarı baz alındığında risk teşkil etmektedirler. Ortalama olarak 5000 doların altında kredi talepleri olmuştur. Fakat 5000 doların üstünde olacak talepleri de risk durumu olarak kötü sonuçlanması olanaklıdır. Birikim hesabına göre oldukça zengin olan kişilerin talepte bulunduğu kredi tutarı baz alındığında yine risk teşkil etmiştir. Bu kişiler de kredi talepleri ortalama olarak 5000 dolardır. Birikim hesabına göre zengin olan kişilerin talepte bulunduğu kredi tutarı baz alındığında risk

durumu kötü olanların iyi olanlara göre fazla olduğu görülmüştür. Ama ortalama olarak birbirine en yakın sonuç veren durumdur. Birikim hesabına göre orta derecede olan kişilerin talepte bulunduğu kredi tutarı baz alındığında bütün sonuçlar gibi riskli gözükmektedir.



Şekil 3.4 : Birikim hesabının meslek grubu ve kredi tutarı risk dağılımı

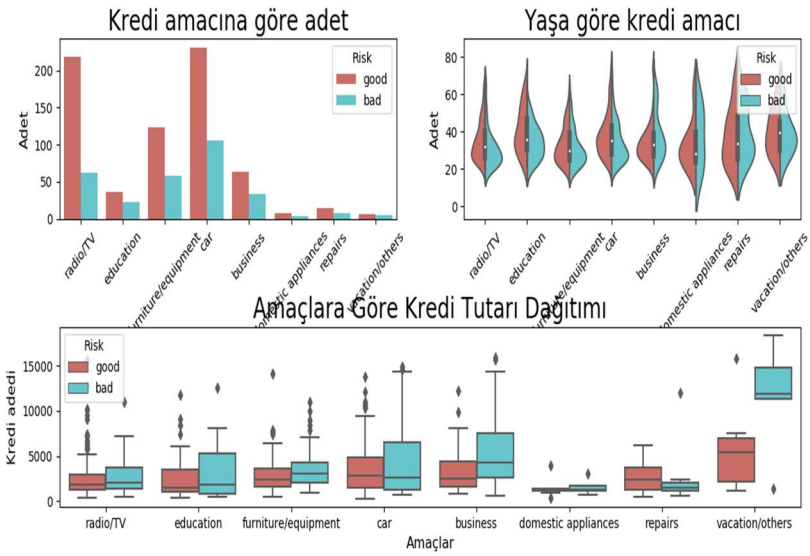
Şekil 3.5'de üç tane tablo verilmiştir. Bu tablolar sayı olarak kaç kişinin hangi kredi amacını seçtiğini, yaş ile kredi amacı arasındaki ilişkiyi ve kredi miktarı ile kredi amacı arasındaki ilişkileri göstermektedir. Veri seti içerisinde kredi amacı alanında radyo ve televizyon, eğitim, mobilya/ekipman, araba, iş, ev aletleri, tamir ve tatil/diğer olarak sınıflandırılmıştır. Birinci tabloya bakıldığında radyo ve televizyon için kredi çeken kişilerin sayısı toplam 250'den fazladır. Bu kişilerin 200'den biraz fazlasının risk durumu iyi 50'den biraz fazla olan kişiler ise risk durumu kötü olan kişilerdir. Eğitim amaçlı kredi çeken kişilerin toplam sayısı yaklaşık 100 kişidir. Ortalama olarak 45 kişinin risk durumu iyi, 35 kişinin risk durumu kötü olarak gözükmektedir. Mobilya/ekipman almak amacıyla kredi çeken kişilerin toplam sayısı ortalama olarak 170 kişidir. Bu kişilerden 120 tanesinin risk durumu iyi, 50 tanesinin risk durumu kötüdür. Araba almak amacıyla kredi çeken kişilerin toplam sayısı 300'den fazladır. Bu kişilerden 200'den fazlasının risk durumu iyi, 100 tanesinin risk durumu kötüdür. İş amacıyla kredi çeken kişilerin toplam sayısı yaklaşık olarak 100 işidir. Bu kişilerden yaklaşık 60 kişinin risk durumu iyi, 30 tanesinin risk durumu

kötüdür. Ev aletleri almak amacıyla kredi çeken kişilerin toplam sayısı yaklaşık olarak 30 kişidir. Bu kişilerden yaklaşık 20 kişinin risk durumu iyi, 10 tanesinin risk durumu kötüdür. Tamirat amacıyla kredi çeken kişilerin toplam sayısı yaklaşık olarak 40 kişidir. Bu kişilerden yaklaşık 25 kişinin risk durumu iyi, 15 tanesinin risk durumu kötüdür. Tatile çıkmak ya da diğer amacıyla kredi çeken kişilerin toplam sayısı yaklaşık olarak 20 kişidir. Bu kişilerden yaklaşık 11 kişinin risk durumu iyi, 9 tanesinin risk durumu kötüdür.

İkinci tabloya bakıldığında yaş ile kredi amacının arasındaki ilişki grafiği çıkartılmıştır. Radyo ve televizyon almak amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu kötü olanların iyi olanlara göre fazla olduğu saptanmıştır. Radyo ve televizyon almak amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu oldukça iyi gözükmektedir. Radyo ve televizyon almak amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu da oldukça iyi gözükmektedir. Eğitim amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu kötü olanların iyi olanlara göre fazla olduğu saptanmıştır. Eğitim amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu iyi olanların kötü olanlara göre fazla olduğu saptanmıştır. Eğitim amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu oldukça iyi olduğu gözlemlenmiştir. Mobilya/ekipman almak amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu ortalama olarak görülmektedir. Yaklaşık olarak risk durumu iyi olan ile risk durumu kötü olan kişi dağılımı eşittir. Mobilya/ekipman almak amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu ortalama olarak görülmektedir. Yaklaşık olarak risk durumu iyi olan ile risk durumu kötü olan kişi dağılımı eşittir. Mobilya/ekipman almak amacıyla başvuran kişilerin 60 ile 80 yaş aralığında talep bulunmamaktadır. Araba almak amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu iyi olanların kötü olanlara göre fazla olduğu saptanmıştır. Araba almak amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu iyi olanlar ile kötü olanlar yaklaşık olarak birbirine yakındır. Araba almak amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu iyi olanlar ile kötü olanlar yaklaşık olarak birbirine yakındır. İş amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu iyi olanların kötü olanlara göre fazla olduğu saptanmıştır. İş amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu kötü olarak gözlemlenmektedir. İş amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu kötü olarak gözlemlenmektedir. Ev aletleri almak amacıyla başvuran

kişilerin 20 ile 40 yaş aralığında risk durumu iyi olanların kötü olanlara göre fazla olduğu görülmektedir. Ev aletleri almak amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Ev aletleri almak amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Tamirat amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu ortalama olarak dengeli gözükmektedir. Tamirat amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu ortalama olarak dengeli gözükmektedir. Tamirat amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu iyi gözükmektedir. Tatil veya diğer amacıyla başvuran kişilerin 20 ile 40 yaş aralığında risk durumu kötüye yakın görülmektedir. Tatil veya diğer amacıyla başvuran kişilerin 40 ile 60 yaş aralığında risk durumu ortalama dengeli olarak görülmektedir. Tatil veya diğer amacıyla başvuran kişilerin 60 ile 80 yaş aralığında risk durumu kötüye yakın görülmektedir.

Üçüncü tabloda ise kredi tutarları ile kredi amaçları arasındaki ilişkileri gösteren grafik verilmiştir. Radyo ve televizyon amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Eğitim amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Mobilya/ekipman almak amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Araba almak amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun kötü olanların iyi olanlara göre fazla olduğu görülmektedir. İş amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Ev aletleri almak amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Tamirat amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun iyi olanların kötü olanlara göre fazla olduğu görülmektedir. Tatil veya diğer amacıyla kredi talebinde bulunan kişilerin kredi tutar talepleri doğrultusunda risk durumunun iyi olanların kötü olanlara göre fazla olduğu görülmektedir.

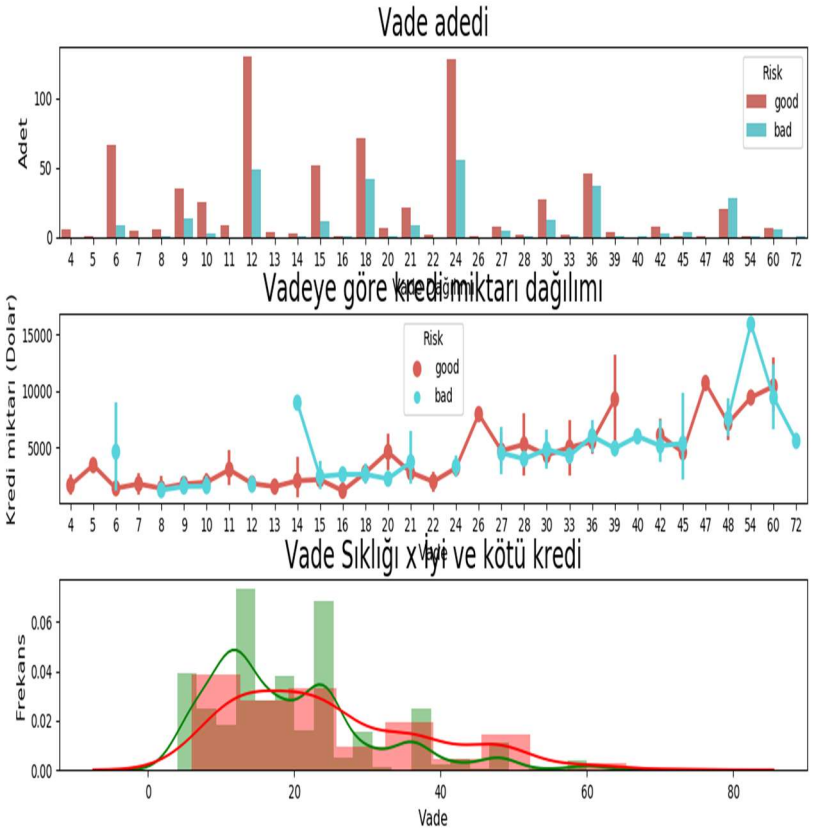


Şekil 3.5 : Kredi amacının yaşa göre ve kredi miktarına göre risk dağılımı

Şekil 3.6'da üç tane tablo verilmiştir. Bu tablolardan birincisi kredi çekme talebinde bulunan kişilerin seçtikleri vade sayıları(ay olarak) göstermektedir. İkinci tabloda çekilen kredi miktarı ile vade sayısının risk durumunda etkisi grafik olarak gösterilmiştir. Üçüncü tabloda ise Vade sayısı ile risk durumu (iyi yada kötü) olanların frekans dağılımları gösterilmiştir. Bu grafikte yer alan yeşil çizgiler risk durumu iyi olanları, kırmızı olan çizgi ise risk durumu kötü olanları göstermektedir. Birinci tablo ele alındığında risk durumu iyi olma olasılığı vade olarak 12 ay ve 24 ay olarak gözükmektedir. Bu vade sayılarına göre hareket eden kişi sayıları toplam 200 kişiden fazladır. Risk durumu kötü olma olasılığı vade olarak yine 12 ay ve 24 ay olarak gözükmektedir. Yüzdelik olarak bakıldığında ise 4 ay, 8 ay, 11 ay, 13 ay, 20 ay ve 39 ay vade sayısı olarak risk durumu iyi gözükmektedir. 27 ay, 36 ay,45 ay ve 48 ay vade sayısı olarak risk durumu kötü gözükmektedir. İkinci tablo ele alındığında kredi tutarı ve vade sayısı karşılaştırılmıştır. 54 ay vadede çekilen 15000 dolarlık tutar risk durumu en kötü olan durumdur. Aynı zamanda 14 ay ve 6 ay vadede çekilen sırasıyla 10000 ve 5000 dolarlık tutarlık yüzdelik olarak risk teşkil eden durumlardır. Risk durumu en iyi olan 47 ay vade ile çekilen 10000 dolarlık durumdur. Ayrıca 26 ay vade ile çekilen 7500 dolar, 22 ay vade ile çekilen 2000 dolar, 14 ay vade ile çekilen 2000 dolar, 13 ay vade ile çekilen 1500 dolar,11 ay vade ile çekilen 3000 dolar, 7 ay vade ile çekilen 1500 dolar, 5 ay vade ile çekilen 4000

dolar, 4 ay vade ile çekilen 1500 dolar risk durumu olarak iyi durumu göstermektedir.

Üçüncü tablo ele alındığında vade sayılarının iyi ve kötü risk ile frekansını göstermektedir. 0 ile 20 ay vade aralığı hem iyi hem de kötü risk kredi frekansı olarak gözükmektedir.

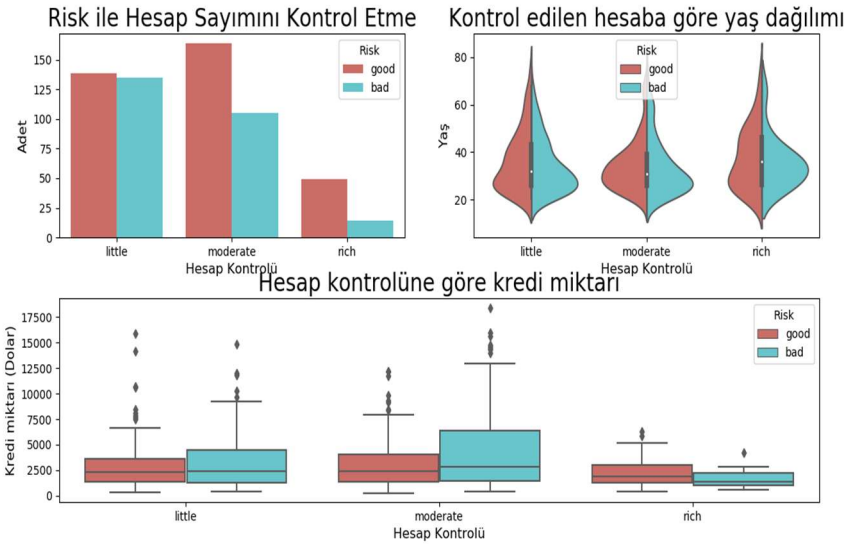


Şekil 3.6 : Vade sayısının kredi miktarı ve risk durumuna göre dağılımı

Şekil 3.7'de üç tane tablo verilmiştir. Bu tablolardan birincisi kişilerin vadesiz hesaplarındaki tutarlara göre az, orta ve zengin olma durumunda olan kişilerin sayılarını göstermektedir. İkinci tabloda vadesiz hesap ile yaş dağılımı gösterilmiştir. Üçüncü tabloda ise vadesiz hesap durumu ile kredi miktarı karşılaştırılmıştır. Birinci tablo ele alındığında, vadesiz hesabındaki tutar az olan kişilerin sayı olarak risk durumu iyi olanların kötü olanlara göre fazla olduğu gözükmektedir. Aynı şekilde vadesiz hesabındaki tutar orta olan kişilerin sayı olarak risk durumu iyi olanların kötü olanlara göre fazla olduğu gözükmektedir. Son olarak vadesiz hesabındaki tutar zengin olarak adlandırılan kişilerin sayı olarak risk durumu iyi olanların kötü olanlara göre fazla olduğu gözükmektedir.

İkinci tablo ele alındığında, vadesiz hesabındaki tutar az olan kişilerin, 20 ile 40 yaş aralığında olması durumunda risk durumu kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Vadesiz hesabındaki tutar az olan kişilerin, 40 ile 60 yaş aralığında olması durumunda risk durumu neredeyse birbirine eşittir. Vadesiz hesabındaki tutar az olan kişilerin, 60 ile 80 yaş aralığında olması durumunda risk durumu iyi olanların kötü olanlara göre fazla olduğu görülmektedir. Vadesiz hesabındaki tutar orta olan kişilerin, 20 ile 40 yaş aralığında olması durumunda risk durumu iyi olanların kötü olanlara göre fazla olduğu görülmektedir. Vadesiz hesabındaki tutar orta olan kişilerin, 40 ile 60 yaş aralığında olması durumunda risk durumu kötü olanların iyi olanlara göre fazla olduğu görülmektedir. Vadesiz hesabındaki tutar orta olan kişilerin, 60 ile 80 yaş aralığında olması durumunda risk durumu kötü olanların iyi olanlara göre fazla olduğu görülmektedir.

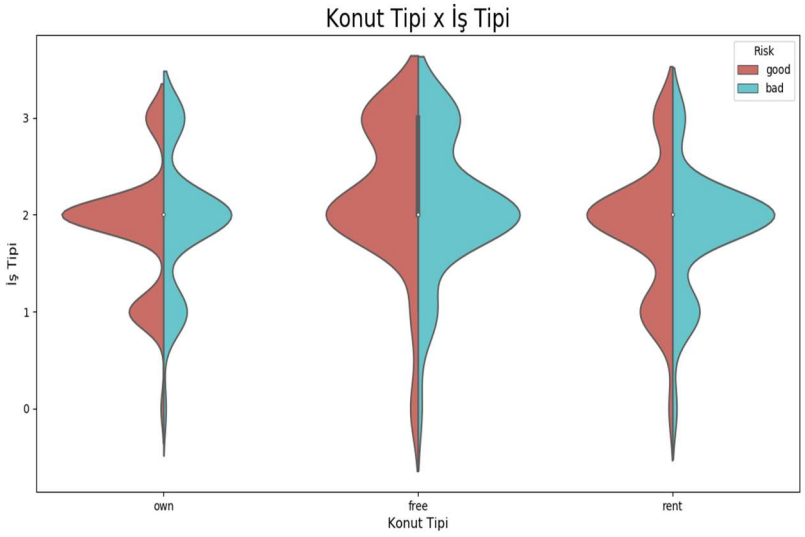
Üçüncü tablo ele alındığında, vadesiz hesap durumu az olan kişilerin çektiği kredi miktarı ile karşılaştırıldığında risk durumu kötü olanların iyi olanlara göre fazla olduğu gözlemlenmiştir. Risk durumu iyi olan kişilerin çektiği ortalama tutar 2500 dolar, risk durumu kötü olanların çektiği ortalama tutar 2500 dolardan biraz fazladır. Vadesiz hesap durumu orta olan kişilerin çektiği kredi miktarı ile karşılaştırıldığında risk durumu kötü olanların iyi olanlara göre fazla olduğu gözlemlenmiştir. Risk durumu iyi olan kişilerin çektiği ortalama tutar 2500 dolardan biraz fazla, risk durumu kötü olanların çektiği ortalama tutar 3500 dolar civarındadır. Vadesiz hesap durumu zengin olan kişilerin çektiği kredi miktarı ile karşılaştırıldığında risk durumu iyi olanların kötü olanlara göre fazla olduğu gözlemlenmiştir. Risk durumu iyi olan kişilerin çektiği ortalama tutar 2000 dolar civarında, risk durumu kötü olanların çektiği ortalama tutar 1000 dolar civarındadır.



Şekil 3.7 : Hesap durumu ile yaş ve kredi miktarı risk dağılımı

Şekil 3.8'de kredi çekme talebinde bulunan kişilerin konut durumu ve meslek grubu ile risk dağılımı verilmiştir. Tabloya göre; meslek grubu "0" olan kişilerin, kendine ait evi olması ile ilişkisi bakıldığında risk durumu birbirine yakın durumdadır. Kötü risk teşkil edenler, iyi risk durumu olanlara göre biraz daha fazladır. Meslek grubu "1" olan kişilerin, kendine ait evi olması ile ilişkisi bakıldığında risk durumu iyi olanların kötü olanlara göre oranı daha fazladır. Meslek grubu "2" olan kişilerin, kendine ait evi olması ile ilişkisi bakıldığında risk durumu iyi olanların kötü olanlara göre oranı oldukça daha fazladır. Meslek grubu "3 olan kişilerin, kendine ait evi olması ile ilişkisi bakıldığında risk durumu kötü olanların iyi olanlara göre oranı daha fazladır. Meslek grubu "0" olan kişilerin, ücretsiz evlerde kalması ile ilişkisi bakıldığında risk durumu iyi olanların kötü olanlara göre oranı daha fazladır. Meslek grubu "1" olan kişilerin, ücretsiz evlerde kalması ile ilişkisi bakıldığında risk durumu kötü olanların iyi olanlara göre oranı daha fazladır. Meslek grubu "2" olan kişilerin, ücretsiz evlerde kalması ile ilişkisi bakıldığında risk durumu kötü olanların iyi olanlara göre oranı daha fazladır. Meslek grubu "3" olan kişilerin, ücretsiz evlerde kalması ile ilişkisi bakıldığında risk durumu iyi olanların kötü olanlara göre oranı daha fazladır. Meslek grubu "0" olan kişilerin, kiralık evlerde kalması ile ilişkisi bakıldığında risk durumu kötü olanların iyi olanlara göre oranı daha fazladır. Meslek grubu "1" olan kişilerin, kiralık evlerde kalması ile ilişkisi bakıldığında risk durumu birbirine yakındır. İyi risk durumu olanların sayısı kötü risk durumunda olanlara göre

biraz daha fazladır. Meslek grubu “2” olan kişilerin, kiralık evlerde kalması ile ilişkisi bakıldığında risk durumu birbirine yakındır. Kötü risk durumu olanların sayısı iyi risk durumunda olanlara göre biraz daha fazladır. Meslek grubu “3” olan kişilerin, kiralık evlerde kalması ile ilişkisi bakıldığında risk durumu birbirine yakındır. İyi risk durumu olanların sayısı kötü risk durumunda olanlara göre biraz daha fazladır.



Şekil 3.8 : Konut durumu ile meslek grubu risk dağılımı

3.3 Verilere Lojistik Regresyon Uygulanması

Veri seti içerisindeki alanların birbirleri ile ilişkileri doğrultusunda analizler çıkartılmıştır. Bu analizler sonucunda ilk olarak lojistik regresyon analizi uygulanmıştır. Lojistik Regresyon uygulanırken ilk önce lojistik regresyon sınıflandırıcısı tanımlanmıştır. Şekil 3.9’da gösterilmiştir.

```
from sklearn.linear_model import LogisticRegression
```

Şekil 3.9 : Sklearn kütüphanesinden lojistik regresyon tanımlama

Sklearn kütüphanesinden lojistik regresyon tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Şekil 3.10’da kod kısmı gösterilmiştir.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=42)
```

Şekil 3.10 : Eğitim ve test verilerinin seçilmesi

Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sonra doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.11'de bu bölümlerin kod kısmı verilmiştir.

```
model = LogisticRegression()
LR = LogisticRegression(class_weight=None, dual=False, fit_intercept=True,
                        intercept_scaling=1, max_iter=100, multi_class='warn', n_jobs=None,
                        penalty='l2', random_state=None, solver='warn',tol=0.0001, verbose=0, warm_start=False)
LR.fit(X_train, y_train)
y_pred = LR.predict(X_test)
print('LogisticRegression Sonucu: ')
x=accuracy_score(y_test,y_pred)
print(x)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
plt.show(x)
```

Şekil 3.11 : Lojistik regresyon için modelin eğitilmesi ve sonuçların alınması

Bu modelde kullanılan parametreler; class_weight,dual,fit_intercept, intercept_scaling, max_iter,multi_class,n_jobs,penalty, random_state, solver,tol, verbose,warm_start'tır.

- **Class_weight:** Formdaki sınıflarla ilişkili ağırlıklardır. Eğer herhangi bir değer verilmezse, tüm sınıfların bir tane ağırlığa sahip olması gerekir.
- **Dual:** Çift veya primal formülasyon olarak kullanılır. Çift formülasyon sadece liblinear çözücü ile l2 ceza için uygulanır.
- **Fit_intercept:** Karar işlevine sabit eklenip eklenmeyeceğini belirtir.
- **İntercept_scaling:** Yalnızca 'liblinear' çözücüsü kullanıldığında ve self.fit_intercept değeri True olarak ayarlandığında kullanışlıdır. Bu durumda, x [x, self.intercept_scaling] olur, yani örnek vektörüne intercept_scaling'e eşit sabit değere sahip bir "sentetik" özellik eklenir.
- **Max_iter:** Çözücülerin yakınsaması için alınan maksimum yineleme sayısıdır.
- **Multi_class:** Seçilen seçenek 'ovr' ise, her bir etiket için bir ikili problem uygundur. 'Multinomial' için en aza indirilen kayıp , *veriler ikili olsa bile* , tüm olasılık dağılımı boyunca multinomiyal kayıp uyumudur . çözücü = 'liblinear' olduğunda 'multinomial' kullanılamaz. 'auto' veri ikiliyse veya

- çözücü = 'liblinear' ise 'ovr' seçeneğini, aksi takdirde 'multinomial' seçeneğini seçer.
- **N_jobs:** Multi_class = 'ovr' ise sınıflar üzerinden paralelleştirilirken kullanılan CPU çekirdeği sayısı solver.multi_class belirtilmiş olsun ya da olmasın, 'liblinear' olarak ayarlandığında bu parametre yok sayılır. Bu bağlamda none olmadığı sürece 1 anlamına gelir.
- **Penalty:** Cezalandırmada kullanılan normu belirtmek için kullanılır. 'Newton-cg', 'sag' ve 'lbfgs' çözücüleri sadece l2 cezalarını destekler.
- **Random_state:** Verileri karıştırırken kullanılacak sahte rasgele sayı üreticisidir.
- **Solver:** Optimizasyon probleminde çözüm için kullanılır.
- **Tol:** Durdurma kriterleri toleransıdır.
- **Verbose:** Liblinear ve lbfgs çözücüler için ayrıntılı olarak verbosity için herhangi bir pozitif sayıya ayarlanır.
- **Warm_start:** True olarak ayarlandığında, önceki çağrının çözümünü başlatma olarak yeniden kullanılır.

Lojistik regresyon analizinin sonucunda %74,80 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.2'de verilmiştir. Ayrıca tablo 3.3'de lojistik regresyonun performans değerlendirme şeması verilmiştir.

Çizelge 3.2 : Lojistik regresyon sonucu karışıklık matrisi

Lojistik Regresyon	
159	19
44	28

Çizelge 3.3 : Lojistik regresyon performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.78	0.89	0.83	178
1	0.60	0.39	0.47	72

3.4 Verilere Lineer Diskriminant Analizi Uygulanması

Lojistik regresyon analizinden sonra, lineer diskriminant analizi modele uygulanmıştır. Lineer diskriminant analizi uygulanırken ilk önce Lineer diskriminant analizi sınıflandırıcısı tanımlanmıştır. Şekil 3.12'de gösterilmiştir.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
```

Şekil 3.12 : Sklearn kütüphanesinden lineer diskriminant analizi tanımlama

Sklearn kütüphanesinden lineer diskriminant analizi tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.13'de bu bölümlerin kod kısmı verilmiştir.

```
model = LinearDiscriminantAnalysis()
LDA = LinearDiscriminantAnalysis(solver='lsqr',shrinkage='auto',n_components=None,tol=1.0e-4)
LDA.fit(X_train, y_train)
y_pred = LDA.predict(X_test)
print('LinearDiscriminantAnalysis Sonucu: ')
y=accuracy_score(y_test,y_pred)
print(y)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.13 : Lineer diskriminant için modelin eğitilmesi ve sonuçların alınması

Bu modelde kullanılan parametreler; solver,shrinkage,n_components ve tol parametreleridir.

Solver: svd': Tekil değer ayrışımı (varsayılan). Kovaryans matrisini hesaplamaz, bu nedenle bu çözücü çok sayıda özelliğe sahip veriler için önerilir.'lsqr': En küçük kareler çözeltisi, büzülme ile kombine edilebilir.'eigen': Özdeğer ayrışması, büzülme ile birleştirilebilir.

- **Shrinkage:** Büzülmenin sadece 'lsqr' ve 'eigen' çözücülerle çalışmaktadır. None: büzülme yok (varsayılan). Auto: Ledoit-Wolf lemması kullanılarak otomatik büzülme. Float between 0 and 1: sabit büzülme parametresi.

- **N_components:** Boyut azalması için bileşen sayısıdır (<= min (n_classes - 1, n_features)). None ise min olarak ayarlanır (n_classes - 1, n_features).
- **Tol:** SVD çözücüsünde sıra tahmini için kullanılan eşiktir.

Lineer diskriminant analizinin sonucunda %73,60 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.4'de verilmiştir. Ayrıca tablo 3.5'de Lineer diskriminant analizinin performans değerlendirme şeması verilmiştir.

Çizelge 3.4 : Lineer diskriminant analizi sonucu karışıklık matrisi

Lineer diskriminant analizi	
156	22
44	28

Çizelge 3.5 : Lineer diskriminant analizi performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.78	0.88	0.83	178
1	0.56	0.39	0.46	72

3.5 Verilere En Yakın Komşu Uygulanması

Lineer diskriminant analizinden sonra, en yakın komşu algoritması modele uygulanmıştır. En yakın komşu algoritması uygulanırken ilk önce En yakın komşu algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.14'de gösterilmiştir.

```
from sklearn.neighbors import KNeighborsClassifier
```

Şekil 3.14 : Sklearn kütüphanesinden en yakın komşu tanımlama

Sklearn kütüphanesinden En yakın komşu algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve

sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.15'de bu bölümlerin kod kısmı verilmiştir.

```
model = KNeighborsClassifier()
KNN = KNeighborsClassifier(weights='distance',algorithm='auto',leaf_size=30,n_jobs=None)
KNN.fit(X_train, y_train)
y_pred = KNN.predict(X_test)
print('KNeighborsClassifier Sonucu: ')
z=accuracy_score(y_test,y_pred)
print(z)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.15 : En yakın komşu ile modelin eğitilmesi ve sonuçların alınması

Bu modelde kullanılan parametreler; weights,algorithm,leaf_size,n_jobs parametreleridir.

- **Weights:** tahminde kullanılan ağırlık fonksiyonudur.
- **Algorithm:** En yakın komşuları hesaplamak için kullanılan algoritmadır.
- **Leaf_size:** Yaprak boyutudur. Optimal değer sorunun doğasına bağlıdır.
- **N_jobs:** Komşu arama için çalıştırılacak paralel iş sayısıdır.

En yakın komşu algoritması sonucunda %67,20 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.6'da verilmiştir. Ayrıca tablo 3.7'de En yakın komşu algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.6 : En yakın komşu algoritması sonucu karışıklık matrisi

En yakın komşu algoritması	
150	28
54	18

Çizelge 3.7 : En yakın komşu algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.74	0.84	0.79	178
1	0.39	0.25	0.31	72

3.6 Verilere Karar Ağacı Uygulanması

En yakın komşu algoritmasından sonra, karar ağacı algoritması modele uygulanmıştır. Karar ağacı algoritması uygulanırken ilk önce Karar ağacı algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.16'da gösterilmiştir.

```
from sklearn.tree import DecisionTreeClassifier
```

Şekil 3.16 : Sklearn kütüphanesinden karar ağacı tanımlama

Sklearn kütüphanesinden Karar ağacı algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.17'de bu bölümlerin kod kısmı verilmiştir.

```
model = DecisionTreeClassifier()
CART = DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,max_features=None,
                              max_leaf_nodes=None,min_impurity_decrease=0.0, min_impurity_split=None,min_samples_leaf=5,
                              min_samples_split=10,min_weight_fraction_leaf=0.0, presort=False, random_state=None,splitter='best')
CART.fit(X_train, y_train)
y_pred = CART.predict(X_test)
print('DecisionTree Sonucu: ')
w=accuracy_score(y_test,y_pred)
print(w)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.17 : Karar ağacı ile modelin eğitilmesi ve sonuçların alınması

Bu modelde kullanılan parametreler; class_weight, criterion, max_depth, max_features, max_leaf_nodes, min_impurity_decrease, min_impurity_split, min_samples_leaf, min_samples_split, min_weight_fraction_leaf, presort, random_state,splitter parametreleridir.

- **Class_weight:** Formdaki sınıflarla ilişkili ağırlıklardır. Eğer verilmezse, tüm sınıfların bir tane ağırlığa sahip olması gerekir.
- **Criterion:** Bölmenin kalitesini ölçme işlevi için kullanılır.

- **Max_depth:** Ağacın maksimum derinliğidir.
- **Max_features:** En iyi ayrımı ararken göz önünde bulundurulması gereken özellik sayısıdır.
- **Max_leaf_nodes:** En iyi düğümler safsızlıkta göreceli azalma olarak tanımlanır. None ise o zaman sınırsız sayıda yaprak düğümü oluşur.
- **Min_impurity_decrease:** Eğer bu ayrılma safsızlığın bu değere eşit veya bu değere eşit bir azalmasına neden olursa bir düğüm bölünecektir.
- **Min_impurity_split:** Ağaç büyümesinde erken durma eşiğidir. Bir düğüm, safsızlığı eşiğin üzerinde olduğunda bölünür, aksi takdirde bir yapraktır.
- **Min_samples_leaf:** Bir yaprak düğümünde olması gereken minimum örnek sayısıdır.
- **Min_samples_split:** Bir iç düğümü ayırmak için gereken minimum örnek sayısıdır.
- **Min_weight_fraction_leaf:** Bir yaprak düğümünde olması gereken toplam ağırlıkların toplamının (tüm girdi örneklerinin) minimum ağırlıklı fraksiyonudur.
- **Random_state:** rastgele sayı üretecidir.
- **Splitter:** Her düğümdeki bölünmeyi seçmek için kullanılan stratejidir.

Karar ağacı algoritması sonucunda %71,60 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.8'de verilmiştir. Ayrıca tablo 3.9'da Karar ağacı algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.8 : Karar ağacı algoritması sonucu karışıklık matrisi

Karar ağacı algoritması	
147	31
40	32

Çizelge 3.9 : Karar ağacı algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.79	0.83	0.81	178
1	0.51	0.44	0.47	72

3.7 Verilere Naive Bayes Uygulanması

Karar ağacı algoritmasından sonra, naive bayes algoritması modele uygulanmıştır. Naive bayes algoritması uygulanırken ilk önce Naive bayes algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.18'de gösterilmiştir.

```
from sklearn.naive_bayes import GaussianNB
```

Şekil 3.18 : Sklearn kütüphanesinden naive bayes tanımlama

Sklearn kütüphanesinden naive bayes algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.19'da bu bölümlerin kod kısmı verilmiştir.

```
model = GaussianNB()
NB = GaussianNB(var_smoothing=1e-9)
NB.fit(X_train, y_train)
y_pred = NB.predict(X_test)
print('GaussianNB Sonucu: ')
q=accuracy_score(y_test,y_pred)
print(q)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.19 : Naive Bayes ile modelin eğitilmesi ve sonuçların alınması

Bu modelde kullanılan parametreler; var_smoothing'dir. Bu sınıflandırıcının parametresi toplam 2 adettir. Diğeri de priors parametresidir.

- **Var_smoothing:** Hesaplama kararlılığı için varyanslara eklenen tüm özelliklerin en büyük varyansının kısmıdır.
- **Priors:** Sınıfların önceki olasılıklarıdır.

Naive bayes algoritması sonucunda %64,80 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.10'da verilmiştir. Ayrıca tablo 3.11'de Naive bayes algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.10: Naive bayes algoritması sonucu karışıklık matrisi

Naive bayes algoritması	
124	54
34	38

Çizelge 3.11: Naive bayes algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.78	0.70	0.74	178
1	0.41	0.53	0.46	72

3.8 Verilere Rastgele Orman Uygulanması

Naive bayes algoritmasından sonra, rastgele orman algoritması modele uygulanmıştır. Rastgele orman algoritması uygulanırken ilk önce Rastgele orman algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.20'de gösterilmiştir.

```
from sklearn.ensemble import RandomForestClassifier
```

Şekil 3.20 : Sklearn kütüphanesinden rastgele orman algoritması tanımlama

Sklearn kütüphanesinden rastgele orman algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.21'de bu bölümlerin kod kısmı verilmiştir.

```
model = RandomForestClassifier()
RF = RandomForestClassifier(max_depth=None, max_features='auto', n_estimators=100, random_state=1)
RF.fit(X_train, y_train)
y_pred = RF.predict(X_test)
print('Random Forest Sonucu: ')
r=accuracy_score(y_test,y_pred)
print(r)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.21 : Rastgele orman ile modelin eğitilmesi ve sonuçların alınması

Bu modelde kullanılan parametreler; max_depth, max_features, n_estimators, random_state parametreleridir.

- **Max_depth:** Ağacın maksimum derinliğidir.
- **Max_features:** En iyi ayrımı ararken göz önünde bulundurulması gereken özellik sayısıdır.
- **N_estimators:** Ormandaki ağaç sayısıdır.
- **Random_state:** Hem ağaçlar oluştururken kullanılan örneklerin önyüklemesinin rastgeleliğini hem de her bir düğümde en iyi ayrımı ararken dikkate alınması gereken özelliklerin örneklenmesini kontrol eder.

Rastgele orman algoritması sonucunda %73,20 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.12'de verilmiştir. Ayrıca tablo 3.13'de Rastgele orman algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.12 : Rastgele orman algoritması sonucu karışıklık matrisi

Rastgele orman algoritması	
160	18
49	23

Çizelge 3.13 : Rastgele orman algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.77	0.90	0.83	178
1	0.56	0.32	0.41	72

3.9 Verilere Destek Vektör Makineleri Uygulanması

Rastgele orman algoritmasından sonra, destek vektör makineleri modele uygulanmıştır. Destek vektör makineleri algoritması uygulanırken ilk önce destek vektör makineleri algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.22'de gösterilmiştir.

```
from sklearn.svm import SVC
```

Şekil 3.22 : Sklearn kütüphanesinden destek vektör makineleri algoritması tanımlama

Sklearn kütüphanesinden destek vektör makineleri algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır.

Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.23'de bu bölümlerin kod kısmı verilmiştir.

```
model = SVC()
SVM = SVC(C=0.95, cache_size=200, class_weight=None, coef0=0.0,decision_function_shape='ovr',
          degree=3, gamma='auto_deprecated', kernel='rbf', max_iter=-1, probability=False,
          random_state=42,shrinking=True, tol=0.001, verbose=False)
SVM.fit(X_train, y_train)
y_pred = SVM.predict(X_test)
print('SVM Sonucu: ')
u=accuracy_score(y_test,y_pred)
print(u)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.23 : DVM ile modelin eğitilmesi ve sonuçları alınması

Bu modelde kullanılan parametreler; c, cache_size, class_weight, coef0, decision_function_shape, degree, gamma, kernel, max_iter, probability, random_state, shrinking, tol ve verbose parametreleridir.

- **C:** Düzenleme parametresidir.
- **Cache_size:** Çekirdek önbelleğinin boyutunu belirtmek için kullanılır.

- **Class_weight:** Eğer değer verilmezse, tüm sınıfların bir tane ağırlığa sahip olması gerekir.
- **Coef0:** Çekirdek fonksiyonunda bağımsız terimdir.
- **Decision_function_shape:** karar fonksiyonunun döndürülüp döndürülmeyeceği belirler(n_samples , n_classes * (n_classes - 1) / 2).
- **Degree:** Polinom çekirdek fonksiyonunun derecesidir.
- **Gamma:** çekirdek katsayısıdır.
- **Kernel:** Algoritmada kullanılacak çekirdek türünü belirtir.
- **Max_iter:** Çözücü içindeki yinelemelerde sabit sınır veya sınırsız -1 olarak kullanılır.
- **Probability:** Olasılık tahminlerinin etkinleştirilip etkinleştirilmeyeceği belirler.
- **Random_state:** Kullanılan sözde rastgele sayı üretecisidir.
- **Shrinking:** Küçülen buluşsal yöntemlerin kullanılıp kullanılmayacağı belirler.
- Tol: Kriteri durdurma toleransıdır.
- **Verbose:** Ayrıntılı çıktıyı etkinleştirmek için kullanılır.

Destek vektör makineleri algoritması sonucunda %71,60 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.14'de verilmiştir. Ayrıca tablo 3.15'te destek vektör makineleri algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.14 : Destek vektör makineleri algoritması sonucu karışıklık matrisi

Destek vektör makineleri algoritması	
172	6
65	7

Çizelge 3.15 : Destek vektör makineleri algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.73	0.97	0.83	178
1	0.54	0.10	0.16	72

3.10 Verilere XGBoost Algoritması Uygulanması

Destek vektör makineleri algoritmasından sonra, XGBoost modele uygulanmıştır. XGBoost algoritması uygulanırken ilk önce XGBoost algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.24'de gösterilmiştir.

```
from xgboost import XGBClassifier
```

Şekil 3.24 : Sklearn kütüphanesinden xgboost algoritması tanımlama

Sklearn kütüphanesinden XGBoost algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.25'te bu bölümlerin kod kısmı verilmiştir.

```
model = XGBClassifier()
XGB = XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,colsample_bytree=1,
                    gamma=0, learning_rate=0.1, max_delta_step=0,max_depth=3, min_child_weight=1,
                    missing=None, n_estimators=70,n_jobs=1, nthread=None)
XGB.fit(X_train, y_train)
y_pred = XGB.predict(X_test)
print('XGB Sonucu: ')
k=accuracy_score(y_test,y_pred)
print(k)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.25 : XGBOOST ile modelin eğitilmesi ve sonuçları alınması

Bu modelde kullanılan parametreler; base_score, booster, colsample_bylevel, colsample_bytree, gamma, learning_rate, max_delta_step, max_depth, min_child_weight, missing, n_estimators,n_jobs ve nthread parametreleridir.

- **Base_score:** Tüm örneklerin başlangıç tahmin puanıdır.
- **Booster:** Hangi güçlendirici kullanılacağını belirler.
- **Colsample_bylevel:** her seviye için sütunların alt örnek oranıdır.
- **Colsample_bytree:** Sütunların alt örneklemesi için bir parametre ailesidir.
- **Gamma:** Ağacın yaprak düğümünde daha fazla bölüm oluşturmak için gereken minimum kayıp azaltımıdır.

- **Max_delta_step:** Her bir yaprak çıkışına izin verilen maksimum delta adımıdır.
- **Max_depth:** Bir ağacın maksimum derinliğidir.
- **Min_child_weight:** minimum örnek ağırlığıdır.
- **N_estimators:** Ağaç sayısıdır.
- **N_jobs:** Paralel iş sayısıdır.
- **Nthread:** XGBoost'u çalıştırmak için kullanılan paralel iş parçacığı sayısıdır.

XGBoost algoritması sonucunda %75,60 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.16'da verilmiştir. Ayrıca tablo 3.17'de aşırı gradyan yükseltme algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.16 : XGBoost algoritması sonucu karışıklık matrisi

XGBoost algoritması	
166	12
49	23

Çizelge 3.17 : XGBoost algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.77	0.93	0.84	178
1	0.66	0.32	0.43	72

3.11 Verilere Gradient Boosting Algoritması Uygulanması

XGBoost algoritmasından sonra, Gradient Boosting algoritması modele uygulanmıştır. Gradient Boosting algoritması uygulanırken ilk önce gradient boosting algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.26'da gösterilmiştir.

```
from sklearn.ensemble import GradientBoostingClassifier
```

Şekil 3.26 : Sklearn kütüphanesinden gradient boosting algoritması tanımlama

Sklearn kütüphanesinden Gradient Boosting algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak

için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.27'de bu bölümlerin kod kısmı verilmiştir.

```
model = GradientBoostingClassifier()
Gradient = GradientBoostingClassifier(criterion='friedman_mse',
                                      init=None,learning_rate=0.1, loss='deviance', max_depth=3,max_features=None,
                                      max_leaf_nodes=None,min_impurity_decrease=0.0, min_impurity_split=None,min_samples_leaf=1,
                                      min_samples_split=2,min_weight_fraction_leaf=0.0, n_estimators=100,n_iter_no_change=None,
                                      presort='auto', random_state=None,subsample=1.0, tol=0.0001, validation_fraction=0.1,verbose=0,
                                      warm_start=False)
Gradient.fit(X_train, y_train)
y_pred = Gradient.predict(X_test)
print('GRAB Sonucu: ')
n=accuracy_score(y_test,y_pred)
print(n)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.27 : Gradient Boosting ile modelin eğitilmesi ve sonuçları alınması

Bu modelde kullanılan parametreler; criterion, init, loss, max_depth, max_features, max_leaf_nodes, min_impurity_decrease, min_impurity_split, min_samples_leaf, min_samples_split, min_weight_fraction_leaf, n_estimators, n_iter_no_change, presort, random_state, subsample, tol, validation_fraction, verbose, warm_start parametreleridir.

- **Criterion:** Bölmenin kalitesini ölçme işlevidir.
- **İnit:** İlk tahminleri hesaplamak için kullanılan bir tahminci nesnesidir.
- **Loss:** optimize edilecek kayıp fonksiyonudur.
- **Max_depth:** Bir ağacın maksimum derinliğidir.
- **Max_features:** En iyi ayrımı ararken göz önünde bulundurulması gereken özellik sayısıdır.
- **Max_leaf_nodes:** En iyi düğümler safsızlıkta göreceli azalma olarak tanımlanır. None ise o zaman sınırsız sayıda yaprak düğümü oluşur.
- **Min_impurity_decrease:** Eğer bu ayrılma safsızlığın bu değere eşit veya bu değere eşit bir azalmasına neden olursa bir düğüm bölünecektir.
- **Min_impurity_split:** Ağaç büyümesinde erken durma eşiği.
- **Min_samples_leaf:** Bir yaprak düğümünde olması gereken minimum örnek sayısıdır.
- **Min_samples_split:** Bir iç düğümü ayırmak için gereken minimum örnek sayısıdır.

- **Min_weight_fraction_leaf:** Bir yaprak düğümünde olması gereken toplam ağırlıkların toplamının (tüm girdi örneklerinin) minimum ağırlıklı fraksiyonudur.
- **N_estimators:** Ağaç sayısıdır.
- **N_iter_no_change:** Geçerlilik puanı iyileşmediğinde eğitimi sonlandırmak için erken durdurmanın kullanıp kullanmayacağına karar vermek için kullanılır.
- **Random_state:** Verileri karıştırırken kullanılacak sahte rasgele sayı üreticisidir.
- **Subsample:** Bireysel temel öğrencilerin uyumu için kullanılacak örneklerin oranıdır.
- **Tol:** Durdurma kriterleri toleransıdır.
- **Validation_fraction:** Erken durma için doğrulama seti olarak ayrılacak eğitim verilerinin oranıdır.
- **Verbose:** Liblinear ve lbfgs çözücüler için ayrıntılı olarak verbosity için herhangi bir pozitif sayıya ayarlanır.

Gradient Boosting algoritması sonucunda %70,80 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.18'de verilmiştir. Ayrıca tablo 3.19'da gradyan yükseltme algoritmasının performans değerlendirme şeması verilmiştir.

Çizelge 3.18 : Gradient Boosting algoritması sonucu karışıklık matrisi

Gradient Boosting algoritması	
160	18
55	17

Çizelge 3.19 : Gradient Boosting algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.74	0.90	0.81	178
1	0.49	0.24	0.32	72

3.12 ADABoost Algoritması Uygulanması

Gradient Boosting algoritmasından sonra, ADABoost algoritması modele uygulanmıştır. ADABoost algoritması uygulanırken ilk önce uyarlamalı güçlendirme algoritması sınıflandırıcısı tanımlanmıştır. Şekil 3.28'de gösterilmiştir.

```
from sklearn.ensemble import AdaBoostClassifier
```

Şekil 3.28 : Sklearn kütüphanesinden Ada Boost algoritması tanımlama

Sklearn kütüphanesinden ADABoost algoritması tanımlama işleminden sonra eğitim ve test verilerinin tanımlaması işlemi yapılmıştır. Eğitim ve test verilerinin seçilmesinden sonra öğrenme verisi olarak seçilen veriler "Fit" fonksiyonu ile oluşturulan örnek sayesinde modelin öğrenme işlemi tamamlanmıştır. Model, eğitim işleminden sonra veri seti içerisindeki her veri için tahmin edilen bilgileri sağlamak için test verisi olarak seçilen veriler için "predict" fonksiyonu kullanılmıştır. Test işlemi de tamamlandıktan sora ekrana doğruluk oranı, karışıklık matrisi ve sınıflandırma raporu ekrana yazdırılmıştır. Şekil 3.29'da bu bölümlerin kod kısmı verilmiştir.

```
model = AdaBoostClassifier()
ADA = AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,learning_rate=1.0, n_estimators=50, random_state=None)
ADA.fit(X_train, y_train)
y_pred = ADA.predict(X_test)
print('ADAB Sonucu: ')
m=accuracy_score(y_test,y_pred)
print(m)
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

Şekil 3.29 : ADABoost ile modelin eğitilmesi ve sonuçları alınması

Bu modelde kullanılan parametreler; algorithm, base_estimator, learning_rate, n_estimators, random_state parametreleridir.

- **Algorithm:** Algoritma belirlemeye yarar.
- **Base_estimator:** Güçlendirilmiş topluluğun inşa edildiği temel tahmincidir.
- **Learning_rate:** Öğrenme oranıdır.
- **N_estimators:** Ormandaki ağaç sayısıdır.

- **Random_state:** Verileri karıştırırken kullanılacak sahte rasgele sayı üreticisidir.

ADABoost algoritması sonucunda %70 oranında başarı sağlanmıştır. Bu analiz sonucunun karışıklık matrisi tablo 3.20'de verilmiştir. Ayrıca tablo 3.21'de uyarlamalı güçlendirme algoritmasının performans değerlendirme şeması verilmiştir.

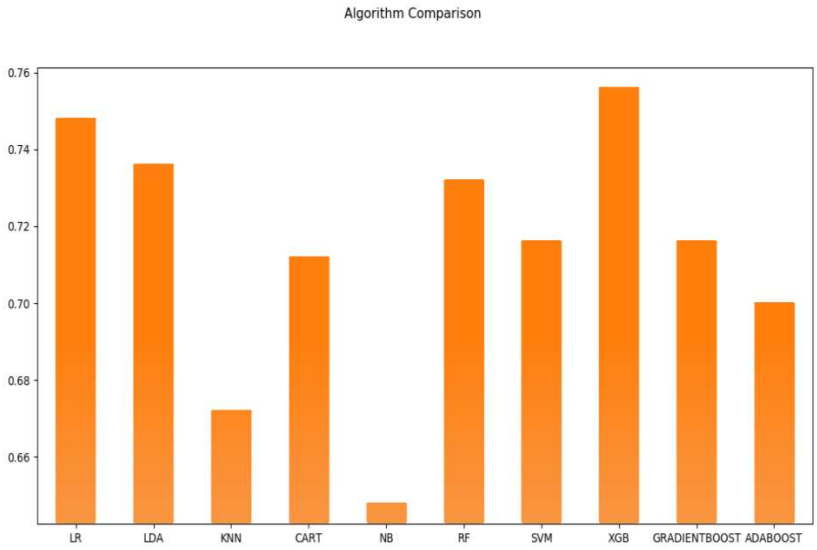
Çizelge 3.20 : ADABoost algoritması sonucu karışıklık matrisi

ADABoost algoritması	
158	20
55	17

Çizelge 3.21 : ADABoost algoritması performans değerlendirme

	Precision	Recall	F1-score	Support
0	0.74	0.89	0.81	178
1	0.46	0.24	0.31	72

Bu algoritmaları sonuçlarını ele alırsak, Lojistik Regresyon yöntemi ile veri kümesi işlem gördüğünde çıkan doğruluk oranı %74,80 olarak saptanmıştır. Lineer Diskriminant analizi yöntemi ile %73,60 başarı oranı yakalanmıştır. En yakın komşu yöntemi ile %67,20 ile çalışmanın en düşük 2. Başarı oranı sağlanmıştır. Bir diğer algoritma olan Karar Ağacı Algoritmasında testler sonucunda maksimum dallanma sayısı 5 ve rastgelelik durumu none olduğu takdirde elde edilen sonuç ise %71,20 olarak saptanmıştır. Naive Bayes yöntemi kullanılarak yapılan çalışmada %64,80 ile çalışmanın en düşük sonucu alınmıştır. Rastgele orman algoritması ile %73,20 oran yakalanmıştır. Destek vektör makineleri yöntemiyle de %71,60 oran yakalanmıştır. XGBoost Modeli ile çalışmanın en yüksek doğruluk oranı olan %75,60 değeri alınmıştır. Gradient Boosting sınıflandırıcısı %71,60 başarılı olmuştur. Son olarak ADA Boosting sınıflandırıcısında %70 başarı oranı gerçekleşmiştir. Bu 10 algoritmanın da parametreleri ve metotları üzerinde çalışılmış olup, Şekil 3.30'da görüldüğü gibi en yüksek sonuç XGBoost Sınıflandırıcısında görülmüştür.



Şekil 3.30: Algoritma sonuçlarının karşılaştırılması

4.SONUÇ VE ÖNERİLER

Kredi talebinde bulunan kişilerin risk teşkil edip etmediği önceden belirlenmesi banka ve finans sektörü için büyük önem arz etmektedir. Bu çalışma 1000 kişi arasında 300 kişinin risk teşkil ettiği, 10 tane kişisel alandan oluşan veri kümesindeki kişilerin krediye uygunluk durumunun tahmin edilmesi konusunda şimdiye kadar yapılan çalışmalar arasında en yüksek başarı oranında tespit edilebildiği görülmüştür. Veri setini eğitilmesi için en uygun ve yüksek oranı veren algoritma XGBoost algoritması olmuştur.

Bu konuda yapılan çalışmalarda en yüksek oran rastgele orman algoritması ile %73,60 iken bu çalışmada bulunan oran %75,60 olarak XGBoost algoritmasının müşterilerin krediye uygunluk durumunu tahmin etmek amacıyla şimdiye kadar yapılan çalışmalarda başarı oranı en yüksek uygulama olduğu görülmüştür.

Ayrıca kaynak olarak kullanılan çalışmadaki algoritmalarında doğruluk oranı artırılmıştır. Ancak XGBoost sınıflandırıcısının başarı oranı en yüksek olduğu için bu algoritma üzerinde durulmuştur. Eğitime entegre edilen öznitelikler ile birlikte işlenen algoritmadaki en yüksek doğruluk oranı bulunmuş ve krediye uygunluk durumu tahmin edildiği gözlemlenmiştir.

KAYNAKLAR

Aue A, Gamon M. (2005). "Customizing sentiment classifiers to new domains:A case study". International Conference on Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria, 21-23 September.

Brownlee J. (2019). A Gentle Introduction to Learning Curves for Diagnosing Machine Learning Model Performance.

Cetiner, Erkan. (2008)" Classifier performances for credit risk analysis.

Chen, Tianqi, (2015). "Xgboost: extreme gradient boosting." R package version 0. 4 -2: 1-4.

Cover,Thomas,andPeterHart.(1967)."Nearestneighborpattern classification. "IEEE transactions on information theory 13.1: 21-27.

Derelioğlu, Gülnur, Fikret Gürgen, and Nesrin Okay. (2009). "A neural approach for SME's credit risk analysis in Turkey." *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, Berlin, Heidelberg,

Edwards, Ward, and Detloff von Winterfeldt. (1986). "Decision analysis and beha vioral research." Cambridge University Press 604: 6-8.

Efşan, Gül, And Bozkurt Gönen. (2005). " Feature Selectıon And Transfer Learnin g Algorıthms Wıth Applıcatıons On Credıt Rısk Analysıs.

Fisher, Ronald A. (1936). "The use of multiple measurements in taxonomic problem s." Annals of eugenics 7.2: 179-188.

Friedman, Jerome H. (2001) "Greedy function approximation: a gradient boosting machine." Annals of statistics: 1189-1232.

Hameed, A. A., Karlik, B., & Salman, M. S. (2016). Back-propagation algorithm with variable adaptive momentum. Knowledge-Based Systems, 114, 79-87.

Ho, Tin Kam. (1995)."Random decision forests." Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE.

Huang, C.L., Chen, M.C. ve Wang, C.J. (2007). Credit scoring with a data mining approach based on support vector machines, Expert systems with applications, 33(4), 847–856.

Huang, C, Chen, M, Wang, C, (2007), *"Credit scoring with a data mining approac h based on support vector machines"*, Expert Systems with Applications, Sayı 33, 847-856.

Kalaycı, Sacide, Mustafa Kamasak, and Seçil Arslan. (2018). "Credit risk analysis using machine learning algorithms." *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE,

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, *34*(11), 2767 - 2787.

Li, S, Shiue, W, Huang, M, (2006), *"The evaluation of consumer loans using suppo rt vector machines"*, Experts Systems with Applications, Sayı 30, 772–782.

Lin C. Probabilistic Topic Models for Sentiment Analysis on the Web. PhD Thesis, University of Exeter, Exeter, UK, 2011

Margineantu, Dragos D., and Thomas G. Dietterich. (1997). "Pruning adaptive bo osting." ICML. Vol. 97.

Malhotra, R, Malhotra, D.K. (2003), *"Evaluating consumer loans using neural net works"*, The International Journal of Management Science, Sayı 31, 83-96.

McCallum, Andrew, and Kamal Nigam. (1998). "A comparison of event models f or naive bayes text classification." AAAI-98 workshop on learning for text categorization. Vol. 752. No. 1.

Murat, G. Ö. K. (2017). "Makine Öğrenmesi Yöntemleri İle Akademik Başarının T ahmin Edilmesi." Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım Ve Teknoloji 5.3: 139-148.

Oguz, Hasan Tahsin, and Fikret S. Gurgen. (2008). "Credit risk analysis using hid den markov model." *2008 23rd International Symposium on Computer and Information Sciences*. IEEE.

Saha, P, Bose, I, Mahanti, A., (2016), *"A knowledge based scheme for risk assessm ent in loan processing by banks"*, Decision Support System, Sayı 84, 78-88.

Sarıman, Güncel. (2011). "Veri madenciliğinde kümeleme teknikleri üzerine bir çal ışma: k-means ve k-medoids kümeleme algoritmalarının karşılaştırılması." Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi 15.3 192-202.

Thomas, L.C., Ho, J. ve Scherer, W.T. (2001). Time will tell: behavioural scoring and the dynamics of consumer credit assessment, IMA Journal of Management Mathematics, 12(1), 89–103.

Tsai, M, Lin, S, Cheng, C, Lin, Y, (2009), *"The consumer loan default predicting*

model – An application of DEA–DA and neural network", Expert Systems with Applications, Sayı 36, 11682–11690.

Vapnik, Vladimir N., and Aleksei Yakovlevich Chervonenkis. (1968)." The unifo rm convergence of frequencies of the appearance of events to their probabilities." Doklady Akademii Nauk. Vol. 181. No. 4. Russian Academy of Sciences,

Verhulst, Pierre-François. (1838). "Notice sur la loi que la population suit dans son accroissement." Corresp. Math. Phys. 10: 113-126.

Yu, Lean, Shouyang Wang, and Kin Keung Lai. (2008). "Credit risk assessment w ith a multistage neural network ensemble learning approach." *Expert systems with applications* 34.2: 1434-1444.

Zhu, X, Li, J, Wu, D, Wang, H, Liang, C, (2013), *"Balancing accuracy, complexit y and interpretability in consumer credit decision making: A C-TOPSIS classification approach"*, Knowledge Based Systems, Sayı 52, 258–267.

İnternet Kaynakları

Url-1 < www.analyticsvidhya.com >, alındığı tarih: 10.12.2019.

Url-2 < machinelearningmastery.com >, alındığı tarih: 17.12.2019.

ÖZGEÇMİŞ

Adı Soyadı : Ömer Yavuz CAN

Doğum Tarihi ve Yeri : 14.04.1995

E-posta : yavuz_can95@hotmail.com

ÖĞRENİM DURUMU:

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Yüksek Lisans	Bilgisayar Mühendisliği	İstanbul Aydın Üniversitesi	
Lisans	Elektrik-Elektronik Mühendisliği	İstanbul Yeni Yüzyıl Üniversitesi	2017
Lise	Bilişim Teknolojileri	Bahçelievler Türk Telekom Teknik ve E.M.L.	2013

MESLEKİ DENEYİM:

Yıl	Firma/Kurum	Görevi
2019	OBJEKT Bilişim İnşaat Müzayede	Müzayede Elemanı
2017-2019	Zeytinburnu Muhsin Ertuğrul Mesleki Eğitim Merkezi	Bilişim Teknolojileri ve Elektrik-Elektronik Öğretmeni