T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE BANKA MÜŞTERİ KAYBI ANALİZİ

Melike PAŞA
2115052003

Tez Danışmanı:
Dr. Öğr. Üyesi Zeynep ÖZER

Bandırma 2024

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE BANKA MÜŞTERİ KAYBI ANALİZİ

Melike PAŞA
2115052003

Tez Danışmanı:

Dr. Öğr. Üyesi Zeynep ÖZER

Bandırma 2024

ETİK BEYAN

TÜRKİYE CUMHURİYETİ
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bu belge ile, bu tezdeki bütün bilgilerin akademik kurallara ve etik ilkelere uygun olarak toplanıp sunulduğunu beyan ederim. Bu kural ve ilkelerin gereği olarak, çalışmada bana ait olmayan tüm veri, düşünce ve sonuçları andığımı ve kaynağını gösterdiğimi ayrıca beyan ederim. (08/02/2024)

Melike PAŞA

ÖZET

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE BANKA MÜŞTERİ KAYBI ANALİZİ

Melike PAŞA

Müşteri kaybı; şirketin ürün ya da hizmetini kullanmayı bırakan ve ilişkisini sonlandırmış müşterilerin yüzdesidir. İşletmeye yeni bir müşteri kazandırmak müşteri kaybetmekten daha zor olabilmektedir. Bu yüzden mevcut müşterinin işletmeyi terk etmesi öngörüldüğünde şirketin zararı minimuma inmektedir. Müşterinin işletmeyi terk edeceğini öngörmek için ise yapay zekâ gibi bilgisayar bilimleri kullanılmaktadır. Makine öğrenmesi yöntemleri ise bu alanlardan biridir. Makine öğrenmesi; genel olarak insanın öğrenme şeklini taklit ederek verilerin ve algoritmaların kullanımına odaklanan bir bilgisayar bilimi ve yapay zekâ dalıdır. Çalışmada Kaggle sitesinden alınan bir bankanın müşteri kaybını tahminlemek için herkese açık olan bir hazır veri seti kullanılmıştır. Veri seti 10.000 adet veri ve 14 kategoriden oluşmaktadır. Çalışmada Fransa, Almanya ve İspanya'nın banka müşterilerinden oluşan veriler bulunmaktadır. Veri setinde; müşteri numarası, soyadı, ülke, cinsiyet, kredi skoru, yaş, müşteri olma süresi, bakiye, ürün sayısı, kredi kart sahibi olup olmadığı, ayrılma durumlarını içeren değişkenlerden oluşmakta olup numpy, pandas, seaborn, matplotlib kütüphaneleri kullanılmıştır. Çalışmada makine öğrenmesi sınıflandırma algoritmalarından olan Lojistik regresyon, K-en yakın komşu, Karar ağacı, Rastgele orman, LightGBM (hafif gradyan arttırma), Catboost (kategorik güçlendirme) sınıflandırma algoritmaları kullanılarak müşteri kaybının nasıl tespit edileceğini ve modellerin performansları karşılaştırılmış sonuçlar ayrıntılı olarak incelemiştir.

Anahtar Kelimeler: Makine Öğrenmesi, Müşteri Kaybı, Sınıflandırma , Veri Madenciliği, Veri Analizi

ABSTRACT

BANK CUSTOMER LOSS ANALYSIS WITH MACHİNE LEARNİNG METHODS

Melike PAŞA

Customer churn is the percentage of customers who stop using the company's product or service and terminate their relationship. It can be more difficult to gain a new customer than to lose one. Therefore, the company's loss is minimized when the existing customer is predicted to leave the business. Computer sciences such as artificial intelligence are used to predict customer defection. Machine learning methods are one of these fields. Machine learning is a branch of computer science and artificial intelligence that focuses on the use of data and algorithms by imitating the way humans learn in general. In this study, a ready-made publicly available data set was used to predict the customer churn of a bank taken from the Kaggle site. The dataset consists of 10,000 data and 14 categories. The data set includes bank customers from France, Germany and Spain. The dataset consists of variables including customer number, surname, country, gender, credit score, age, length of time as a customer, balance, number of products, whether they have a credit card or not, whether they have left or not, and numpy, pandas, seaborn, matplotlib libraries were used. In the study, machine learning classification algorithms such as Logistic regression, K-nearest neighbor, Decision tree, Random forest, LightGBM (light gradient boosting), Catboost (categorical boosting) classification algorithms were used to determine how to detect customer churn and the performances of the models were compared and the results were examined in detail.

Keywords: Machine Learning, Classification, Churn, Data Mining, Data Analysis

ÖNSÖZ

Bu tez çalışmasının gerçekleştirilmesinde değerli katkılarından dolayı danışmanım Sayın Dr. Öğr. Üyesi Zeynep ÖZER' e teşekkürlerimi sunuyorum. Hayatım boyunca benden desteklerini esirgemeyen sevgili aileme sonsuz teşekkür ediyorum.

Melike PAŞA

Bandırma, 2024

İÇİNDEKİLER

TABLOLAR

ŞEKİLLER

KISALTMALAR

CB : Kategorik Güçlendirme

CRM : Müşteri İlişkileri Yönetimi

DT : Karar Ağacı

DNN : Derin Sinir Ağları

EFB : Ayrıcalıklı Öznitelik Destekleme

FP : Yanlış Pozitif

FN : Yanlış Negatif

GBM : Gradyan Arttırma Makinesi

GBDT : Gradyan Destekli Karar Ağacı

GOSS : Gradyan Tabanlı Tek Taraflı Örnekleme

GR : Kazanç Oranı

IML : Bireysel Makine Öğrenimi

K-NN : K- En Yakın Komşu

LGBM : Hafif Gradyan Arttırma Makinesi

ML : Makine Öğrenimi

NB : Naif Bayes

SVM : Destek Vektör Makinesi

RF : Rastgele Orman

TP : Doğru Pozitif

TN : Doğru Negatif

VIF : Varyans Enflasyon Faktörü

XGB : Aşırı Gradyan Arttırma

GİRİŞ

Müşteri kaybı, genel olarak bir firmanın ürünü ya da hizmetini kullanmayı bırakmış kişiler anlamına gelmektedir. Banka, e -ticaret, telekomünikasyon vb. şirketlerin uzun ömürlü olması müşterilere bağlı olmaktadır. Bankacılık sektöründe artan rekabet ile bankalar stratejiler geliştirerek müşterilerin takibini yapmalı ve müşteriyi elinde tutmalıdır. Müşterilerin hizmet sağlayıcılardan ayrılma işlemlerine Churn denir (Mahajan, Mishra, Mahajan, 2015). Müşterilerin daha iyi bir hizmet kalitesi ve fiyat oranı bulması dahilinde bankayı terk etmesi çok kolay olabilmektedir. Şirketlerin karşılaştığı zorluklardan bir diğeri ise müşterilerin değişken davranışları ve istekleridir. Müşteriler, ticari bankalar arasındaki rekabeti daha da arttıran deneyime, kişiselleştirilmiş hizmete, çeşitliliğe ve çevikliğe daha fazla önem veriyor (He vd., 2014). Şirketler için müşterileri elde tutmak, onları ikna etmek oldukça zor olmakla birlikte şirketlerin müşteri kaybetmesi şirketi büyük zarara uğratabilmektedir. Bu zarara uğramamak için şirketlerin müşterinin istek ve ihtiyaçlarını dikkate alması gerekmektedir.

Müşteri kaybı birçok nedenden kaynaklanmaktadır. Örneğin; müşteri belli bir amaç için hesap oluşturmuşsa ve amaç gerçekleştikten sonra bankadan ayrılacaksa ya da bir müşteri banka hizmetinden memnun değilse müşteri kaybı gerçekleşmektedir. Yeni bir müşteri edinme maliyeti, mevcut müşterileri elde tutma maliyetinin yaklaşık 5 katı, memnun olmayan müşteriyi geri kazanmanın maliyeti mevcut müşteriyi elde tutmanın yaklaşık 10 katıdır (Massey ve Mitzi, 2001). Bu nedenle bankaların erken aşamada müşteri kaybını tahmin edebilen bir sistem kullanması oldukça önemlidir. Veri madenciliği ve makine öğrenmesi yöntemleri kullanımı tahminleme ve öngörme işlemleri için uygun olan sistemlerdendir. Bankaların gelecek için yapabileceği en iyi yatırım; müşteri memnuniyetini sağlamak ve müşteriyi elde tutmak olmalıdır. Bankalar ve diğer finansal kuruluşlar müşteri kaybı gerçekleşmeden önce bir müşterinin yaygın uyarı işaretlerini tespit etmek için müşterinin işlemlerini düzenli olarak kontrol eder (Amuda ve Adeyemo, 2019). Banka müşterileri düşük maliyetli olan, etkili reklam faaliyetleri uygulayan ve müşteri odaklı olan diğer firmalara geçmeye eğilimli olabilmektedir. Müşterilerin farklı firmalara geçmesini öngörmek bankaların müşterileri kaybetmeden kişiye özel promosyonlar düzenleyerek müşterilerin geri kazandırılmasını sağlamaktadır.

Müşteri kaybını tahmin etmek ve önlemek için makine öğrenmesi yöntemlerinden yararlanılarak avantaj sağlanabilmektedir. Veri madenciliği, bilgisayar bilimlerinin alt alanıdır. Büyük veri setlerinden anlamlı sonuçlar çıkarmamızı sağlayan veri madenciliği yönteminde çıkarılan sonuçlara göre reaksiyon alındığında sonraki aşama öngörülmekle birlikte işletmelerin bu sonuçlara göre hareket etmesi önem arz etmektedir böylece şirketin zarar etmesi engellenecek ya da zarar minimuma düşürülerek maliyet azalacaktır ve karlılık artacaktır. Veri madenciliği yöntemlerini doğru bir şekilde uygulamak için öncelikle veri ambarı oluşturulması gerekmektedir. Veri ambarları bankaların geçmişten günümüze kadar olan birçok veriyi depolayabilen, temizleme, ayıklama, veri yönetimi, analiz, raporlama gibi birçok yetkinlikten oluşan bir veri yönetimi sistemidir. Veri madenciliği teknikleri, şirketlerin pazarlama yöntemlerinin geliştirilmesi, karlılığının artması, müşteri memnuniyetinin sağlanması açısından önemli bir bilgi teknolojisidir. Veri madenciliği teknikleri; bilişim, işletme, bilim, endüstri, banka, sağlık, sigortacılık, eğitim vb. birçok alanda kullanılmaktadır. Büyük miktardaki verileri hızlı bir şekilde inceleyip tahminlemekte ve işletmenin kâr oranını arttırmaktadır.

Veri madenciliği algoritmalarının birincil görevlerinden biri sınıflandırmadır (Fayyad vd.,1996). Müşteri kaybını tahmin etmek için en yaygın olarak kullanılan teknikler şunlardır; sinir ağları, destek vektör makineleri ve lojistik regresyon modelleri (Zoric, 2016) . Makine öğrenmesi; genel olarak insanın öğrenme şeklini taklit ederek verilerin ve algoritmaların kullanımına odaklanan bir bilgisayar bilimi ve yapay zekâ dalıdır. Makine öğrenimi algoritmaları ile müşterilerin işlemleriyle ilgili verilerin toplanması ve eğitilmesinin ardından yararlı bilgiler keşfedilir ve çözülür. Bireysel müşterilerin ve grupların satın alma kalıpları, müşteri verilerini analiz ederek tanımlanabilir (Wells, Fuerst, Choobineh, 1999). Çeşitli araştırmalarla kanıtlandığı gibi, makine öğrenimi, müşterileri elde tutma oranlarını arttırmak için bankalarda ve diğer sektörlerde Crm tekniklerini uygulamak hayati öneme sahiptir. Görüldüğü gibi, müşteri yıpranmasının azaltılması ticari bankalar için sadece kârın artmasında değil, aynı zamanda temel rekabet gücünün artmasında da önemli bir etkiye sahiptir (He vd., 2014).

Bu çalışmada farklı sınıflandırma algoritmaları kullanarak en uygun teşhis aracı bulunması ve bankadan ayrılma olasılığı yüksek olan müşterilerin erken teşhis edilmesi amaçlanmıştır. Çalışmanın asıl amacı; banka ve müşteri sadakati ile ilgili tahminleme

işlemlerine en uygun ve en doğru makine öğrenmesi yöntemini bulmaktır. Çalışmada banka müşterilerinin bankayı terk etme olasılığı üzerine çeşitli sınıflandırma modelleri kullanılarak müşteri kayıp analizi yapılmıştır. Banka müşterilerinin bankayı terk etme olasılığı hesaplanarak erken aksiyon alınması amaçlanmıştır. Çalışma; veri hazırlık, veri ön işleme, eğitim ve test setlerine bölme, modelleme ve değerlendirme aşamalarından oluşacaktır. Böylece bankalar ve diğer müşteriye bağlı sektörler müşteri verilerini tutarak müşteri ayrılması ve yeni müşteri bulma durumunda oluşan zarardan etkilenmemiş olacaklardır. Bu tez çalışmasında, Kaggle sitesinden alınan banka müşteri kaybı tahmini hazır veri seti kullanılmıştır. 10,000 adet müşteri verisi ile çalışılmıştır. Veri seti ülkelere göre incelendiğinde banka müşteri oranları şu şekildedir; Fransa %50, Almanya %25 ve İspanya %25. Banka müşterilerinin cinsiyet dağılım oranları ise; %54 erkek ve %46 kadındır. Bu çalışmada farklı sınıflandırma algoritmaları kullanılarak banka müşteri kaybı tespitinde en uygun makine öğrenme algoritması tespit edilmeye çalışılmıştır.

Bu tez çalışmasında verilerin analizi ve görselleştirmesi için Python programlama dili kullanılmıştır. Makine öğrenmesi algoritmaları mevcut veri kümesi üzerinde çalışılarak elde edilen sonuçlar ayrıntılı olarak sunulmuştur. Çalışmada gerekli yöntemlerin hesaplanıp analiz edilebilmesi için Spyder programında Numpy, pandas, seaborn, matplotlib kütüphaneleri ve Lojistik regresyon, K-en yakın komşu, Karar ağacı, Rastgele orman, LightGBM (hafif gradyan arttırma), Catboost (kategorik güçlendirme) modelleri kullanılmıştır. Modelin tahminlenmesi için F1 skor, duyarlılık (Recall), kesinlik (precision), doğruluk (accuracy) metrikleri kullanılmıştır. Çalışmada eksik değer bulunmamakla birlikte satır numarası, müşteri ıd, soyad sütunları silinmiştir. Veriler %70 eğitim %30 test ve %80 eğitim %20 test olarak çalıştırılmış olup en iyi sonucu %70 eğitim seti verdiği için çalışmaya onunla devam edilmiştir. Cinsiyet ve ülke sütunlarına tek sıcak kodlama tekniği uygulanmıştır. Kullanılan algoritmalar arasından en yüksek doğruluğu %90,43 ile Catboost (Kategorik Güçlendirme) modeli, 2. yüksek doğruluğu %90,35 ile LightGBM (Hafif Gradyan Arttırma) modeli vermiştir.

Bu çalışma toplamda 3 bölümden oluşmaktadır. Bankacılık sektöründe müşteri kaybı ve makine öğrenmesi bölümünde; bankacılık ve müşteri kaybı alanlarında gerçekleştirilen geçmiş çalışmalar, yazarların kullandığı sınıflandırma algoritmaları ve başarı oranlarından bununla beraber bankacılık sektörü, müşteri kaybı, yapay zekâ

alanları hakkında ayrıntılı olarak bahsedilmiş olup bankacılık sektöründe müşteri kaybının önlenmesi konularından bahsedilmiştir. Makine öğrenmesi yöntemleri ve sınıflandırma algoritmaları bölümünde; mevcut tez çalışması için belirlenen konu hakkında bilgi verilmiş olup veri analizinin gerçekleştirildiği program ve algoritmalar hakkında ayrıntılı olarak bahsedilmiştir. Banka müşteri kaybı analizi ile ilgili uygulama bölümünde; karışıklık matrisi tanımı yapılmış olup, tez çalışması için kullanılan sınıflandırma algoritmalarının başarı hesaplamaları yapılmıştır. Veri setinin ayrıntılı açıklanması, değişkenler arasındaki ilişkilerin veri görselleştirme yöntemiyle yorumlanması, veri hazırlığı, veri önişleme, veri önişleme için kullanılan parametrelerden ayrıntılı olarak bahsedilerek verilerin analizi için kullanılan teknikler ve veri setinin korelasyon analizi yapılarak açıklanmıştır. Özellik ölçeklendirme, eğitim ve test kümesi oluşturma, model oluşturma ve oluşan modellerin değerlendirmesi ayrıntılı bir şekilde sunulmuştur. Son olarak tez çalışması ayrıntılı olarak değerlendirilmiş olup veri analizi için kullanılan algoritmalar ve teknikler açıklanarak en yüksek performans veren algoritma sonucu açıklanarak öneride bulunulmuştur.

BİRİNCİ BÖLÜM

BANKACILIK SEKTÖRÜNDE MÜŞTERİ KAYBI VE MAKİNE ÖĞRENMESİ

Bu bölümde makine öğrenmesi yöntemlerinin bankacılık sektörü ve müşteri kaybına olan etkilerinden bahsedilmiş olup literatür çalışması bölümünde ise bu alanda yapılan çalışmalar, kullanılan algoritmalar ve başarı oranları ele alınmaktadır.

1. MAKİNE ÖĞRENMESİ VE VERİ ANALİZİ KAVRAMLARININ BANKACILIK SEKTÖRÜNE OLAN ETKİLERİ

Bu bölümde makine öğrenmesi, veri analizi ve bankacılık sektörü hakkında bilgi verilecektir.

1.1 Banka Kavramı

Bankalar; para, sermaye, kredi, iskonto ve kambiyo gibi her türlü finansal ve ekonomik işlemleri düzenleyen ve gerçekleştiren ticari kuruluşlardır bununla beraber menkul kıymet alım ve satımı, para havaleleri, faiz, kredi ve banka kartı gibi ödeme hizmetlerini sağlamaktadır. Tarihte bilinen en eski perakende bankası 1472 yılında kurulan "Banca Monte dej Paschj dj Siena" bankasıdır en eski ticaret bankası ise 1590 yılına kurulmuş olan "Berenberg Bank"tır. Banka kelimesinin kökeni "banca" kelimesi olup İtalyancadan dilimize geçmiştir. 1900'lü yılların başında ilk modern bankacılık uygulamalarına başlanmıştır. Bankalar ülkelerin kalkınmasında önemli etkenlerden biridir bununla birlikte bankalar ne kadar güçlü ve gelişmiş olursa ülke ekonomisi de aynı şekilde güçlenmekte ve gelişmektedir. Bankalar, müşterilerin varlıklarını etkin bir şekilde yöneten kuruluşlardır bu bağlamda ülkedeki finansal yapının güçlü olması bankalar sayesinde olmaktadır. Uzun vadeli ekonomik dengeyi korumaya yardımcı olan kuruluşlar, makroekonomik istikrara katkıda bulunurlar. Ekonomik sistemle etkileşimlerini net bir şekilde gösterebilen ve sağlıklı bir yapıda faaliyet gösterebilen bankacılık sektörü, finansal sistem içinde büyük bir öneme sahiptir. Hızlı ekonomik büyüme ve gelişmenin sağlanmasında kilit bir rol oynamaktadır. Bankalar kendilerinin sahip olduğu ve tasarruf sahiplerinden elde ettiği kaynakları etkili bir şekilde işleten, fon akışını sağlayan, kredi işlemlerinde kullanan finansal kurumlardır. Bankalarda gerçekleştirilen işlemler kayıt altında olup ilgili kuruluşlar tarafından düzenli olarak takibi yapılmaktadır. Zamanla gelişen bilişim teknolojileri müşteri odaklı sektörlerden biri olan finans sektöründeki pek çok alanın gelişmesine katkı sağlamıştır

1.2. Müşteri Kaybı

Müşteri kaybı; müşterilerin firmadaki hizmeti kullanmayı bırakması ve rakip firmalara yönelmesi durumudur. Günümüzde bütün firmalar için müşteri kavramı oldukça önem arz etmektedir bu yüzden mevcut müşteriyi elde tutma yöntemleri firmalar için önemli bir stratejidir. CRM (Müşteri İlişkileri Yönetimi), firmaların mevcut müşteriler ile potansiyel müşteriler için etkili bir iletişim kurmayı, müşteriyi elde tutmayı ve müşteri kaybını önleyerek firmanın gelişmesini sağlayan bir yöntemdir. Firmayı terk etmeyen sadık müşteriler firmadaki karlılığı arttırırlar.

Firmalar için yeni müşteri kazanmak oldukça maliyetli bir işlemdir bu yüzden düzenli müşteri takibi yapılması müşterinin firmadan ayrılmadan önce tespit edilerek ayrılma nedeni araştırılmakta ve müşteriyi firmaya geri kazandırmak adına kişisel bazı kampanyalar düzenlenerek müşteri kaybının önüne geçilmektedir. Müşteri kaybı analizi, mevcut müşteriler içerisinde ayrılmayı düşünen potansiyel müşterileri öncesinde doğru bir şekilde tespit etmek, ayrıntılı bir şekilde incelemek ve engellemek için gerekli yöntemler kullanılan süreç olarak da tanımlanabilir. Müşteri kayıp analizi için çoğu zaman yapay zekanın alt dalı olan makine öğrenmesi yöntemleri tercih edilmektedir. Müşteri kaybı analizi yapılmasının ardından ayrılacak olan müşterilerin hangilerinin elde tutulması gerektiğinin bir analizi yapılmalıdır böylece firma gereksiz harcamaları engellemiş olacak ve doğru müşterinin seçilmesine yardımcı olacaktır. Sadık müşterilerin firmayı terk etme oranı azdır ve şirketin karlılığına etkisi fazladır aynı zamanda bu müşteriler firmaya yeni müşteriler kazandırır bunun için müşteriye verilen hizmetin geliştirilmesi sağlanarak müşterinin firmayı terk etmemesi sağlanır. Günümüzde müşteri odaklı pek çok firmanın özellikle bankaların artması pazardaki rekabeti arttırmaktadır. Hızla gelişen bilişim teknolojileri kullanımı ile her geçen gün artmakta olan büyük miktardaki veriler depolanmaktadır bu artan veriler sonucunda büyük veri (Big Data) kavramı ortaya çıkmaktadır. Bu verilerin etkili bir şekilde kullanılması analiz edilmesi ve anlamlı sonuçlar çıkarılması kurumlar için oldukça önemlidir. Veri analizi, müşterilerin sistemdeki hareketlerinden ve gerçekleştirdiği işlemler üzerinden yola çıkarak ham verilerden faydalı bilgileri ortaya koyar. Bankacılık, finans, sağlık gibi pek çok sektör büyük verilere sahiptir.

1.3. Yapay Zekâ, Makine Öğrenmesi ve Veri Madenciliği

Yapay zekâ, makine öğrenmesi ve derin öğrenme yöntemlerini kapsayan büyük bir teknolojidir. Günümüzde bankacılık sektöründe kullanılması kaçınılmaz olan yapay zekâ uygulamaları müşterilerin verilerini toplama, analiz etme ve değerlendirme işlemlerini gerçekleştirir. Makine öğrenimi terimi ilk olarak 1959 yılında yapay zekâ ve bilgisayar oyunları alanında tanınmış olan Arthur Samuel tarafından keşfedilmiştir. Makine öğrenimi gerçek dünya problemleri içerisinde; bankacılık sektörü, görüntü işleme, hava durumu tahmini, sosyal medya uygulamaları gibi pek çok alanda karşımıza çıkmaktadır. Makine öğrenmesi ile veri madenciliği yöntemleri birbirleri ile bağlantılıdır.

Veri madenciliğinde verilerden anlamlı bilgiler ortaya koymak ve bu verileri kullanarak doğru kararlar verebilmek için veri tabanlarındaki verilerin doğru bir şekilde hazırlanması ve işlenmesi gerekmektedir. Veri madenciliği yönteminde büyük veriler içerisinden faydalı bilgiler elde edebilmek için uygulanan tekniklerden olan makine öğrenmesi, mevcut veri seti üzerine gerekli modelleri kullanarak eğitim verilerinden öğrenme gerçekleştirir ve test verilerini değerlendirir, mevcut verilerden firmanın analizini yapmakta ve gelecekte oluşabilecek problemleri tahmin ederek performansı yükseltmeyi amaçlamaktadır. Firmalar gelecekte oluşabilecek problemlerin önüne geçip maliyetlerini düşürerek ve gerekli aksiyonları alarak kârını da arttırabilmektedir bu yüzden de veri madenciliği ve makine öğrenmesi yöntemlerinin kullanımı firmalar için hayati önem taşımaktadır.

Bankalar için yapay zekâ ve makine öğrenmesi teknolojilerinin kullanılması oldukça önem arz etmektedir. Bankalar makine öğrenmesi yöntemleri sayesinde; müşteri verilerinin analiz edilmesi, müşteriler için kişiselleştirilmiş hizmet verilmesi, gerekli yatırım tavsiyelerinde bulunulmasına yardımcı olur. Bankacılık sektörü müşteri temelli bir kuruluş olduğundan ve çok fazla sayıda müşteri bulunduğundan dolayı büyük miktarda veriye sahiptir bu bağlamda bankacılık sektörü için müşteri ilişkileri yönetimi (CRM) uygulaması oldukça önemli bir yöntemdir. Müşteri ilişkileri yönetimi, müşteri ile iletişimi arttırarak müşteri memnuniyetini amaçlar ve firmanın gelirini arttırmasını sağlamaktadır. Veri madenciliği ve makine öğrenmesi yöntemleri ile müşteri ilişkileri yönetimi uygulaması entegre bir şekilde çalışması durumunda müşteri kaybı en aza indirilmektedir.

2. LİTERATÜR TARAMASI

(Wu, 2022) Çalışmasında ticari banka müşterileri ile ilgili müşteri kaybı tahmini kamuya açık bir veri kümesi ile çalışma yürütmüş. Çalışmada; Neural Network (sinir ağı), karar ağacı, lojistik regresyon algoritmaları kullanılmıştır. Çalışma verimliliğini arttırması için HNNSA algoritması kullanılmış olup çalışma sonucunda HNNSA'nın önemli ölçüde bireysel makine öğrenimi (IML) ve derin öğrenmeden daha iyi performans göstermiştir. Algoritmalar arasında yapılan tahminlemede en yüksek doğruluğu ise %93,06 ile Neural Network (sinir ağı) algoritması vermiştir.

(Saini vd., 2017) Telekomünikasyon endüstrisinde karar ağacı algoritması ile müşteri kaybı tahminleme çalışması yapılmış. Çalışmada 33.000 müşteri verisi kullanılmıştır. Çalışma için SPSS programı kullanmış olup çalışma sonunda CHAID (Ki-kare otomatik etkileşim algılama) tekniğinin daha verimli ve daha doğru olduğunu kanıtlamıştır. Elde ettiği sonuca göre en yüksek doğruluğu Karar ağacı algoritması %98,88 ile aynı benzerliği yapay sinir ağı %98,43 doğrulukla vermiştir.

(Sayed vd., 2018) Banka müşterilerinin verileri alınarak müşteri kayıp olasılığını tahmin etmek ve model oluşturmak için ayrıntılı karşılaştırma yapılmış. ML paketinin en iyi sonuca ulaştırdığını söylemiştir. Karar Ağacı, ML, MLlib, algoritmalarını kullanmıştır. Apache Spark ile ML ve MLlib paketleri çalıştırılmış ve ML paketi %79 doğruluk sağlamıştır. Çalışma sonunda MLlib paketinin eğitim süresi için hızlı ve daha doğru olduğu Dada tromes tabanlı API ise test süresi için daha kesin ve doğru sonucuna varılmıştır.

(Çiçek ve Arslan, 2020) Çalışmasında, müşteri kayıp analizi yapılarak sınıflandırma algoritmaları üzerinde çalışmış. Banka ve telefon operatörü olmak üzere iki veri seti ile çalışılmış olup Naif Bayes, K-en yakın komşu, Lojistik Regresyon, Destek Vektör Makinesi, Karar Ağacı ve LDA (Doğrusal Diskriminant Analizi) sınıflandırma modelleri kullanılmıştır. Çalışma sonunda en yüksek doğruluğu %80,41 Karar Ağacı algoritması vermiştir. Tüm sonuçların %70'ten fazla doğruluk verdiği saptanmış.

(Domingos vd., 2021) Bankacılık sektöründe yayık tahmini için DNN'ler kullanılarak farklı hiperparametrelerin etkilerinin deneysel bir analizi çalışılmış. Çalışmada MLP, DNN (Derin Sinir Ağları), AdaGrad, SGD, Rmsprop, Adadelta, Adamax modelleri kullanmış. Çalışma sonunda bankacılık sektöründeki çalkalama tahmininde eğitim ve test aşamalarında farklı parti boyutlarının kullanılmasının DNN'

nin performansı üzerindeki etkisi belirlenmiş. En yüksek doğruluğu %86,45 oranıyla DNN modeli vermiştir.

(Yalçın, 2019) Çalışmasında bankacılık sektörüne ait müşteri verisi üzerinden makine öğrenmesi algoritmalarını kullanarak müşteri kayıp analizi yapılmış. Çalışmada açık kaynak kodlu, görsel veri bilimi geliştirme ortamı olan Knime kullanılmış. Python kodlama dili ve Sklearn kütüphanesi ve AutoML felsefesini benimsemiş olan OptiScorer ürünleri kullanılarak karşılaştırılmış. Çalışmada Karar Ağacı, Naif Bayes, K-en yakın komşu, SVC (Destek Vektör Makinesi), Lojistik regresyon, Rastgele Orman sınıflandırma algoritmaları uygulanmış. Çalışma sonunda Rastgele Orman %85 doğruluk oranıyla başarıya ulaşmış. OptiScorer skorlama sistemiyle doğruluk yüzdesini %90'a çıkartmış. Makalenin bankacılık dışında müşteri kaybının önemli olduğu tüm sektörlere katkı sağlayacağını belirtmektedir.

(Tekouabou vd., 2022) Çalışmada müşteri ilişkileri yönetimi sistemlerinde sınıflandırma için makine öğrenmesi ve tahmine dayalı analiz yapılmış. Verileri dengelemek için SMOTE tekniği kullanılmış. Yapılan çalışmada Rastgele Orman modeli doğruluk ve f1 puanı açısından en iyi performansı 0,86 olarak vermiş. Çalışma sonunda modelleme ve topluluk yöntemlerini dengelemek için SMOTE kombinasyonunun çapraz doğrulama içeren özlü ve ayrıntılı bir makine öğrenimi modeli oluşturma sürecini önermiştir. Oluşturulan ve optimize edilen modeller 'yaş' özelliğinin en önemli 'HasCrCard' özelliğinin ise daha az önemli olduğu analizinde yorumlamış. Bu çalışmayla hizmet kalitesinin yükseltilmesi, müşteri hizmetlerinin iyileştirilmesi amaçlanmıştır.

(Pamungkas vd., 2020) Kaggle sitesinden aldığı banka veri setini kullanılarak denetimli öğrenme yaklaşımının yöntemlerini analiz edip karşılaştırmış. Çalışmada Python notebook programı kullanılmış. Lojistik Regresyon, K-en yakın komşu, Naif Bayes, SVM (Destek Vektör Makinesi), Rastgele Orman sınıflandırma algoritmaları kullanılmış. Çalışma sonunda en iyi sonucu Rastgele Orman %86 doğruluk değeri vermektedir.

(Dur vd., 2023) Çalışmasında bankacılık sektöründe mobil pazarlama kampanyaları doğrultusunda müşteri kayıp analizi gerçekleştirilmiş. Çalışmada Yapay Sinir Ağı, Lojistik Regresyon ve Destek Vektör Makine algoritmaları kullanılmış. Çalışma sonunda kullanılan yöntemlerin doğruluk, kesinlik, duyarlılık ve f1 performans

metriklerinin değerleri birbirlerine yakın çıkmış olup lojistik regresyon az bir oranla kullanılan yöntemlerden daha iyi sonuç vermiştir.

(Akyiğit ve Taşçı, 2022) Çalışmasında bir sigortacılık şirketine ait veriler ile müşteri kaybı analizi çalışılmıştır. Çalışmada gözetimli öğrenme yöntemlerinden olan; Karar Ağacı, Rastgele Orman, K-en yakın komşu sınıflandırma algoritmaları kullanılmış. Çalışma sonunda en başarılı sonucu %87 oran ile Rastgele Orman vermiştir.

(Kaynar vd., 2017) Çalışmada Telekomünikasyon sektöründe müşteri kaybını tahmin etmek için Destek vektör makinesi, Yapay sinir ağı, Naif bayes sınıflandırma yöntemleri ile analiz gerçekleştirilmiştir. Çalışma sonunda sadık ya da terk eden müşterileri sınıflandırmada Yapay sinir ağı modeli diğer makine öğrenmesi yöntemlerine göre daha başarılı olmuştur.

(Raeisi, 2020) Çalışmasında E-Ticaret sektöründeki müşteri kayıp tahmini üzerine çalışılmış. K-en yakın komşu, Naif Bayes, Gradient Boosted Trees (Gradyan Arttırma Ağaçları), Karar Ağacı, Rastgele Orman sınıflandırma algoritmaları kullanılmıştır. En yüksek doğruluk ise Gradyan Arttırma %86,90 olarak sonuçlanmaktadır.

(Suguna vd., 2022) Finansal firmalardaki kayıp veriler üzerinde ayrıntılı bir çalışma yapılmış. Çalışmada; Lojistik Regresyon, Naif Bayes, Lineer SVM, Rastgele Orman, K-en yakın komşu, Çok Katmanlı Algılayıcı, Gradyan Yükseltme, XGB (Aşırı gradyan Arttırma), LGBM (Hafif gradyan arttırma) sınıflandırıcıları kullanılmıştır. Dengesiz veri setini dengeli hale getirmek için rastgele örnekleme yöntemleri kullanılmış. Çalışma sonucunda çok katmanlı perceptron (Algılayıcı)'un kayıp veriler için iyi performans gösterdiği ortaya konulmuştur.

(Zhang, 2022) Çalışmasında; bankalardaki farklı müşteri kaybı türleri için kurtarma stratejileri üzerinde sınıflandırma çalışması yapılmıştır. Kayıp müşterilerin sınıflandırılması için K-means(K-Ortalamalar) algoritması kullanılmış. Tüm müşteriler beş kümeye ayrılmış ve bu beş kümede bulunan kayıp müşteriler için farklı kümeleme stratejileri yapılmış. XGB (Aşırı gradyan Arttırma) algoritması, müşteri kaybını tahmin etmek için, K-Ortalamalar algoritması ise müşteri kaybını 5 kategoriye ayırmak için kullanılmış, tüm parametreler ayarlandıktan sonra uygulanan sınıflandırma algoritmaları arasından en yüksek doğruluğu %83,65 oranında XGB (Aşırı gradyan Arttırma) sınıflandırıcısı vermektedir.

(Rahman ve Vasimalla, 2020) Çalışmada, veri madenciliği yöntemleri kullanılarak bir bankadaki müşteri kaybını tahminlenmesi amaçlanmıştır. Çalışmada KNN, Karar Ağacı, Rastgele Orman ve Destek Vektör Makinesi algoritmaları kullanılmıştır. KNN, komşu için k değeri 5 olarak ayarlanmıştır. Çalışma sonucunda incelenen sınıflar içerisinde en iyi sonucu %95,74 ile Rastgele Orman algoritması vermiştir.

(Bozyiğit ve Tarhan, 2020) Çalışmada, sosyal medya paylaşımları kullanarak markaların itibarı analiz edilmiş ve tüketicilerin arasındaki ilişkinin yönetilmesi hedeflenmiş. Çalışmada; NB (Naif Bayes), Destek Vektör Makinesi, Yapay Sinir Ağı, Karar Ağacı, Rastgele Orman, K-en yakın komşu sınıflandırma algoritmaları uygulanmıştır. Uygulanan bu sınıflandırma algoritmaları arasından en yüksek doğruluğu %90 oranı, Naif Bayes sınıflandırıcısı ile ulaşılmıştır.

(Mhaske, 2022) Çalışmada, bankacılık sektöründeki müşteri kaybının tahminlenmesi amaçlanmıştır. Çalışmada müşterinin hizmetten ayrılması ve hizmetten ayrılmasına neden olan faktörler incelenmiştir. Çalışmada Kaggle' den alınan hazır veri seti kullanılmıştır. Rastgele Orman, Extra Tree, Sinir Ağı, Gaussian NB, K-en yakın komşu ve Lojistik Regresyon sınıflandırma algoritmaları kullanılmış. Kullanılan modellerin karşılaştırılması sonucu en iyi performansı %86,80 doğruluk oranı ile Rastgele Orman sınıflandırıcısı vermiştir.

(Mahesh, 2019) Çalışmasında, makine öğrenimi algoritmalarının uygulama alanlarından ve gelecekteki beklentilerinden bahsedilmiştir. Çalışmada denetimli, denetimsiz, yarı denetimli ve takviyeli öğrenme çeşitlerini ayrıntılı olarak işlenmiştir. Karar Ağacı, Sinir Ağları, Naif Bayes, Destek Vektör Makinesi, K-en yakın komşu sınıflandırma algoritmaları açıklamalı bir şekilde ifade edilmiş. Çalışma sonunda az ve etiketlenmiş veriler için denetimli öğrenme, büyük veriler için denetimsiz öğrenme, kolayca ulaşılabilen büyük veriler için ise takviyeli öğrenme tekniğinin kullanımını önermiştir.

(Dalmia vd., 2020) Bankacılık sektöründeki müşteri kaybı tahmini yapılması amaçlanmış. Çalışmada denetimli öğrenme yöntemleri kullanılarak bankayı terk etme riski yüksek olan müşteriler hakkında tahmin yapmak ve bankayı bilgilendirme amaçlı bir algoritma oluşturulmuş. K-en yakın komşu ve XGBoost (Aşırı gradyan Arttırma)

sınıflandırma algoritmaları kullanılmış ve sonuçlar karşılaştırıldığında %86,85 doğruluk oranı ile Aşırı gradyan Arttırma en iyi sonucu vermiştir.

(Kunt, 2019) Tez çalışmasında, bir telekomünikasyon şirketinin verileri ile müşteri kaybının tahminlenmesi amaçlanmıştır. SQL script dili ile yapılan bu çalışmada veri madenciliği yöntemlerinden Karar Ağacı, Rastgele Orman, XGBoost (Aşırı gradyan Arttırma), Naif Bayes, Lojistik Regresyon sınıflandırma algoritmaları kullanılmıştır. Karar ağaçları ve Rastgele Orman algoritmalarının işlenmesi ve analiz edilmesi için ise KNİME analytics uygulaması kullanılmıştır.

(Bilgen, 2009) Tez çalışmasında, analitik hiyerarşi yöntemiyle elektronik bankacılık sektöründe müşteri kaybı analizi yapılmıştır. Çalışmada uluslararası bir finansal kuruluşun veri seti örneklendirilmiş. Çalışmada, müşterilerin sistem kullanım bilgileri kullanılarak, müşteri beklentilerini ön görülmesi amaçlanmıştır.

(Kasım, 2022) Tez çalışmasında, yazılım sektöründe hizmet veren bir firmanın veri madenciliği yöntemi ile müşteri kayıp analizi yapılmıştır. Çalışmada Orange yazılımı ve Python programlama dili kullanılmıştır. Sınıflandırma için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, K en yakın komşu, Naif Bayes, Yapay Sinir Ağı algoritmaları kullanılmış. Kullanılan algoritmaların doğruluk oranları karşılaştırılmış ve çapraz doğrulama kullanılarak doğruluk oranları tekrar analiz edilmiş ve iki sonuçta da en yüksek doğruluğu Rastgele Orman algoritması vermiştir.

(Önal, 2017) Çalışmasında, Türkiye'de faaliyet gösteren bir lojistik firmasının 2015 ve 2016 yıllarındaki 5.000 adet müşteri bilgisi incelenmiş ve kaybedilen müşteri davranışları ortaya çıkarılmıştır. Çalışmada, Spyder (Python 3.6) uygulaması kullanılarak makine öğrenmesi algoritmalarından Destek Vektör Makinesi algoritması uygulanmış ve gelecek 3 aydaki müşteri kayıp analizi tahminlemesi yapılmıştır.

İKİNCİ BÖLÜM

MAKİNE ÖĞRENMESİ YÖNTEMLERİ VE SINIFLANDIRMA ALGORİTMALARI

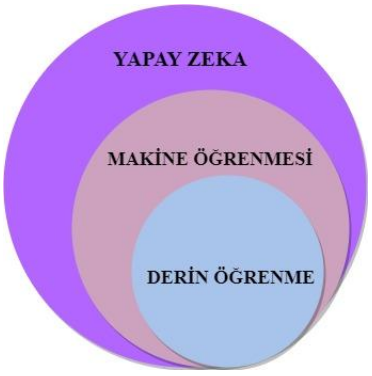
Bu bölümde makine öğrenmesi kavramı, çalışmada kullanılan yöntemler ve sınıflandırma algoritmalarına değinilmektedir.

1.YÖNTEMLER

Bu çalışmada, bankacılık sektöründe aboneliği bulunan müşteri verilerinden yararlanılmış olup bu verilerin makine öğrenmesi yöntemleri ile churn analizi yapılmıştır. Çalışmada banka müşterilerinin verilerinden kayıp tahmini yaparak bankadan ayrılacak müşterileri erken tespit etmek ayrıca müşterilerin aboneliklerini sürdürmesi ve bankaların zarara uğramasının engellenmesi amaçlanmıştır. Gerekli yöntemlerin analiz edilip hesaplanması için Spyder (Python 3.9) uygulaması kullanılmıştır. Çalışmanın analizi için makine öğrenmesi yöntemlerinden olan; Lojistik Regresyon, K-en yakın komşu, Karar Ağacı, Rastgele Orman, LightGBM (Hafif Gradyan Arttırma), Catboost (Kategorik Güçlendirme) sınıflandırma algoritmaları kullanılmıştır.

1.1. Makine Öğrenmesi

Makine öğrenimi terimi ilk olarak 1959 yılında Arthur Samuel tarafından ortaya çıkarılmıştır. Makine öğrenmesi; minimum insan müdahalesiyle insanın öğrenme şeklini taklit ederek geçmiş deneyimlerden ve tahmin yapmak için belirlenen verilerden otomatik olarak öğrenme yeteneği sağlamaktadır ayrıca verilerin ve algoritmaların kullanımına odaklanan bir bilgisayar bilimi ve yapay zekâ dalıdır.



Şekil 2. 1: Makine Öğrenmesi ile Yapay Zekâ ve Derin Öğrenme Kavramlarının Bağlantısı

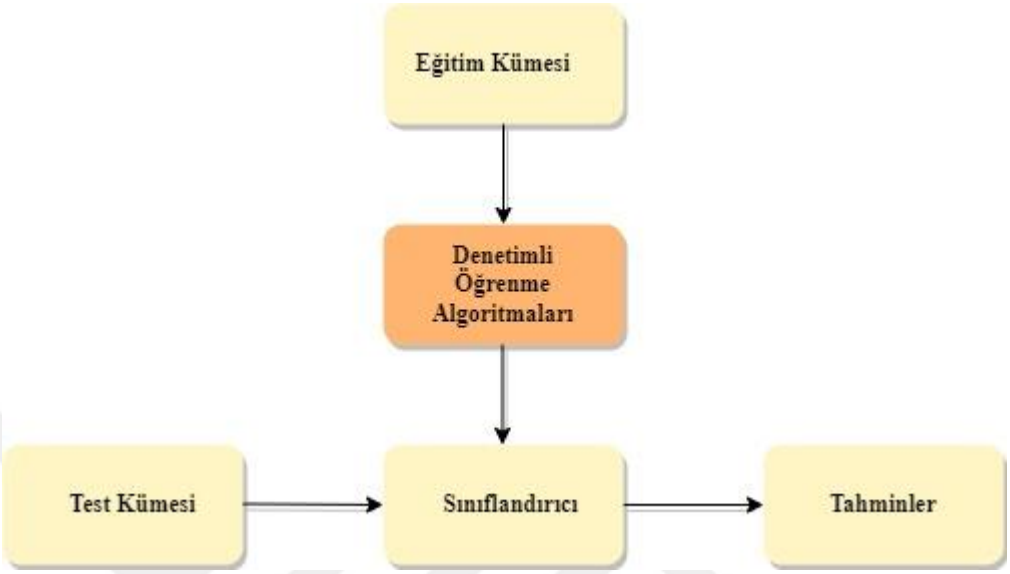
Yapay zekâ, makine öğrenmesi ve derin öğrenmeyi de içine alan bir sistemdir. Arthur Samuel makine öğrenimini açıkça programlanmadığı halde makinelere öğrenme yeteneği kazandıran disiplin olarak tanımlamaktadır (Mahesh, 2018). Makine öğrenimi algoritmaları ile programlamada tahminler ve kararlar vermek için eğitim verileri olarak bilinen örneklere dayalı bir model oluşturmaktadır ardından sonuçlar tahminlenip ilgili verilerle eğitilerek daha doğru sonuçlar vermesine olanak sağlamaktadır. Makine öğrenmesi, verilerdeki yararlı bilgileri keşfetmek ve karmaşık verileri çözmek için olanak sağlamaktadır. Bilgisayar biliminin "sorunlara çözüm getirecek program üretme" güdüsüyle, istatistik biliminin "eldeki verilerden çıkarım yapma" güdüsünü birleştiren makine öğrenmesi, yeni bir disiplin oluşturarak daha etkin sistemler yaratmayı amaçlamaktadır (Mitchell, 2006). Makine öğrenmesi algoritmalarının kullanımı; elde edilen istatistiksel sonuçlar doğrultusunda mevcut problemi en aza indirmeyi sağlamaktadır. Gerekli uygulamalar ile veri setinin analizini yapar ve performans ölçümü ile kullanılan algoritmalar içerisinden en uygun olanının seçilmesini sağlamaktadır.

Makine öğrenimi; Hesaplamalı finans (Kredi puanlama, algoritmik ticaret), Bilgisayar görüşü (Yüz tanıma, hareket izleme, nesne algılama), hesaplamalı biyoloji (Dna dizilimi, beyin tümörü tespiti, ilaç keşfi), otomotiv, havacılık, ses tanıma gibi çok sayıda alandaki sorunları çözmeyi sağlamaktadır. Bir makine öğrenimi modeli; belirli bir veri kümesi kullanılarak oluşturulmuş modelin eğitilmesiyle özgün çıktılar ortaya koyar ve bu çıktılar üzerinden karar almayı sağlar. Makine öğrenim türleri; Gözetimli, Gözetimsiz ve Pekiştirmeli öğrenme olmak üzere üç sınıfa ayrılmaktadır. Gözetimli öğrenme yönteminde verilerin sınıf tahmini yapılabilmesi için öncesinde var olan veriler ile eğitimi gerçekleştirilirken gözetimsiz öğrenme yönteminde ise öncesinde mevcut veri bulunmaksızın sınıflandırma işlemini gerçekleştirilmektedir. Denetimli öğrenme algoritmalarında, meydana gelebilecek tüm durumları içeren eğitim setine sahip olmak oldukça maliyetlidir. Denetimsiz öğrenme algoritmalarında ise veri seti genelde çok geniş ve yorumlanması zordur (Dilki ve Başar, 2020).

1.1.1. Denetimli (Gözetimli) Öğrenme

Verileri sınıflandırmak veya sonuçları doğru tahmin etmek için algoritmaları eğitmek üzere etiketli veri kümelerinin kullanılmasıyla tanımlanmaktadır. Denetimli öğrenme yöntemleri öncesinde mevcut veriler ile eğitilmesinden dolayı denetimsiz

öğrenme yöntemine kıyasla daha başarılıdır. Denetimli öğrenme algoritması şeması Şekil 2.2 ’de gösterilmektedir;



Şekil 2.2: Denetimli Öğrenme Algoritma Modeli

1.1.2. Denetimsiz (Gözetimsiz) Öğrenme

Etiketlenmemiş veri kümelerini analiz etmek ve kümelemek için makine öğrenimi algoritmalarının kullanılmasıdır. Ayrıca denetimsiz öğrenme, verilerdeki gizli yapıların bulunması için oldukça önemli bir makine öğrenimi türüdür. Elde edilmesi hedeflenen sonuç için özel bir tanımlamada bulunulmamışsa veya belirsizlik söz konusu ise denetimsiz terimi kullanılmaktadır (Chapman vd., 2000).

1.1.3. Takviyeli (Pekiştirmeli) Öğrenme

Takviyeli öğrenme (Reinforcement Learning), kendi ortamında algılama ve hareket yapan bağımsız bir etmenin, hedefine ulaşması için en uygun hareketleri nasıl öğrenip uygulayacağını gösteren bir öğrenme türüdür (Karadoğan, 2014). Takviyeli öğrenmenin asıl amacı, mevcut problemler doğrultusunda en iyi sonucu bulmaktır. Sınıflandırma kategorik değerler üzerinde örüntü kurma, sınıflama ve tahmin etme işlerini yaparken regresyon analizi ise süreklilik veya kesikli durum gösteren değişkenlerin birbirleri arasındaki ilişkiyi belirlemeyi sağlar (Çalış vd., 2014).

1.2. Lojistik Regresyon

Denetimli öğrenme türlerinden olan lojistik regresyon ikili ya da çoklu sınıflandırma işlemi yaparak bir olayın meydana gelme olasılığının iki olası sonucunu tahmin etmek veya hesaplamak için kullanılan istatistiksel bir sınıflandırma yöntemidir. Lojistik regresyon ikili (binary) bir bağımlı değişken ile bir dizi bağımsız değişken arasındaki ilişkiyi açıklamaya yönelik tahminleyici bir analizdir (Sevli, 2019). Buna bağlı olarak lojistik regresyon sonuçları 0 ve 1 arasındaki değerlerden oluşmaktadır. Bazen logit model olarak adlandırılan lojistik regresyon, çoklu bağımsız değişkenler ile kategorik bir bağımlı değişken arasındaki ilişkiyi analiz eder ve verileri bir lojistik eğriye uydurarak bir olayın meydana gelme olasılığını tahmin eder (Park, 2013). Lojistik regresyon modelinin kullanımı 1845 yılına kadar gitmektedir. İlk olarak o dönemde nüfus artışına yönelik yapılan matematiksel çalışmalar sırasında ortaya çıkmıştır (Gürcan, 1998).

Lojistik regresyon, 1960'lı yıllarda ikili verilerin analizi için kullanıldığı özellikle tıpta yaygın olduğu istatistik alanında geliştirildi (Cox, 1969). 1970'lerin sonlarında sonlarından başlayarak dilbilimsel varyasyon çalışmasının biçimsel temellerinden biri olarak dilbilimde yaygın olarak kullanılmaya başlandı (Sankoff ve Labou, 1979). Lojistik regresyon, girdiden gerçek değerli özellikleri çıkarak her birini bir ağırlıkla çarpan, toplayan ve bir olasılık oluşturmak için toplamı bir sigmoid işlevinden geçiren denetimli bir makine öğrenimi sınıflandırıcısıdır (Jurafsky ve Martin, 2023). Lojistik regresyon analizi, bağımlı değişkenin ölçüldüğü ölçek türüne göre ve bağımlı değişkenin kategori sayısına bağlı olarak 'Binary lojistik regresyon analizi', 'Multinominal lojistik regresyon analizi', 'sıralı lojistik regresyon analizi' olarak da adlandırılır. Lojistik regresyon modeli oluşturulurken dikkat edilmesi gereken en önemli faktör değişken seçimidir.

Lojistik regresyon 'tek değişkenli lojistik regresyon' ve 'çok değişkenli lojistik regresyon' olarak ikiye ayrılır (Stephenson, 2008, Çokluk, 2010). Lojistik regresyonun kullanım amacı, istatistikteki diğer modeller ile aynıdır. Analiz aşamasında doğru bir model oluşturulabilmesi için; bağımlı değişken ile bağımsız değişkenin birbirleri ile olan ilişkisi minimum değişken bulunmalı ve maksimum uyum sağlaması gerekmektedir. Sınıflandırma problemlerinde kullanılan lojistik regresyon temel makine öğrenmesi

türlerinden biridir. Sınıflandırma ise mevcut verilerin değerlendirilmesi sonucunda 1 veya 0 olarak tahminlenmesi işlemidir.

Lojistik regresyon, veri setindeki tüm değişkenlerin birbirleri arasındaki ilişki analizini yapmayı sağlamaktadır. Lojistik regresyon analizi, son yıllarda sıklıkla kullanılan güçlü analitik araçlardan biridir aynı zamanda kategorik sonuçları tahminlemeyi sağlamaktadır. Lojistik regresyon analizi; makine öğrenimi, ekonomi, sağlık, üretim, pazarlama gibi pek çok alanda yaygın olarak kullanılmaktadır. Lojistik regresyonda tahmin edilen (bağımlı) değişken kategoriktir ve tahmin edilen değer 0 ile 1 arasında yer alır. Lojistik regresyon, - sonsuz ile + sonsuz arasında değerler almaktadır. Logit ismi, odd değerinin doğal logaritmasını ifade etmektedir. Yani π olasılığını göstermek üzere, logit; (Budak, Şen ve Yıldırım, 2013).

$$\log it\big(\pi(x)\big) = \log\left(\pi(x) \div 1 - \pi(x)\right) \quad \textbf{(2.1)}$$

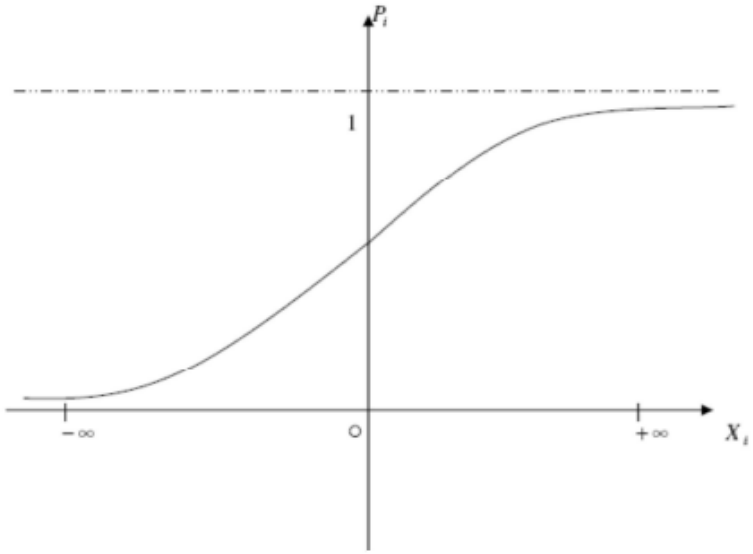
Logit model, bağımsız değişken değeri sonsuza gittiği zaman, bağımlı değişkenin 1'e asimptot olduğu matematiksel bir fonksiyondur (İnal vd.,2006). Lojistik regresyon analizinde, lojistik fonksiyon denklemi aşağıda verilmiştir.

$$\mathbf{P} = \frac{\mathbf{e\beta 0+\beta 1X1+\cdots+\beta kXk}}{\mathbf{1+e\beta 0+\beta 1X1+\cdots\ldots+\beta kXk}} \quad \textbf{(2.2)}$$

P: İncelenen olayın gözlenme olasılığını **β0**: Bağımsız değişkenler sıfır değerini aldığında bağımlı değişkenin değerini başka bir ifadeyle sabiti, **β1 β2..... βk :** Bağımsız değişkenlerin regresyon katsayılarını, **X1 X2.... Xk :** Bağımsız değişkenleri, k: Bağımsız değişken sayısını, **e:** 2.71 sayısını göstermektedir (Özdamar, 2002: 475). Genel lojistik fonksiyon denklemi aşağıdaki gibidir;

$$\boldsymbol{f(x)} = \frac{\mathbf{1}}{\mathbf{1+}\boldsymbol{e}^{\boldsymbol{-x}}} \quad \textbf{(2.3)}$$

Yukarıdaki denklemde x değerine h (0) değeri almaktadır böylece çıkan sonuç 0 ile 1 arasında bir değer olmaktadır. Logit fonksiyon denkleminin görselleştirilmesi sonucunda ortaya çıkan S eğrisi Şekil 2.3'te gösterilmektedir.



Şekil 2.2: Lojistik (Sigmoid) Fonksiyon Grafiği

Kaynak: (Terzi, 2021)

Lojistik regresyon, sigmoid fonksiyonu ile işleme alınmaktadır. Mevcut verilerle yapılan tahminlemede bulunan olasılık değeri 0.5ten küçük olması durumunda modele 0 sonucu atanır aynı şekilde olasılık değeri 0.5ten büyük veya eşit olması durumunda ise model 1 sonucunu almaktadır.

Tablo 2.1: Lojistik Regresyonda Olasılık Değeri Atama

Olasılık Değeri	Sonuç
Bulunan değer < 0.5	0
Bulunan değer>= 0.5	1

Lojistik regresyon; ikili (binary) lojistik regresyon, sıralı (ordinal) lojistik regresyon, multinominal lojistik regresyon olmak üzere üçe ayrılmaktadır. Lojistik regresyon; Odds, Odds Ratio (OR), Lojit model olmak üzere 3 temel özellikten oluşmaktadır.

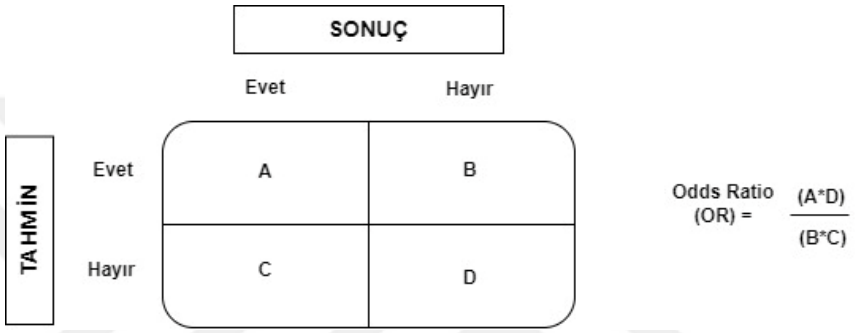
1.2.1. Odds

Bir durumun olma olasılığı ile o durumun olmama olasılığına bölünmesi sonucu ortaya çıkan değerdir.

$$odds = \frac{p(x)}{1-p(x)} \quad (2.4)$$

1.2.2. Odds Ratio (OR)

OR olasılık oranı anlamına gelmektedir. İki ayrı odds değerinin birbirleri ile olan oranıdır ayrıca OR oranı her zaman için pozitif değer almaktadır. OR, belirli bir sonuç için bir risk faktörü olup olmadığını belirlemek ve bunun için farklı risk faktörlerinin büyüklüğünü karşılaştırmak içinde kullanılmaktadır.



Şekil 2.3: Odds Ratio

1.2.3. Lojit

Odds değerinin doğal logaritmasıdır. Logit model; bağımlı değişkenin tahmini değerlerini olasılık olarak hesaplayarak olasılık kurallarına uygun sınıflama yapma imkânı veren, tablolaştırılmış ya da ham veri setlerini analiz eden bir istatistiksel yöntemdir (Budak, Şen ve Yıldırım, 2013). Lojistik regresyon ile doğal logaritma olasılıklarını açıklayıcı değişkenin lineer bir fonksiyonu olan basit lojistik regresyon modellenmesi aşağıdaki denklemde gösterilmektedir.

$$logit\,(y) = \ln(odds) = \ln\left(\frac{p}{1-p}\right) = a + \beta X \quad (2.5)$$

Yukarıdaki denklemde; p: ilgili sonucun olasılığı, a: kesişme parametresi, β: regresyon katsayısı, X: öngörücü anlamlarına gelmektedir.

Lojistik regresyon kullanımından en iyi sonuçları elde etmek için, sürekli bir bağımsız değişkendeki herhangi bir değişikliğin, pozitif bir sonucun logaritmik olasılıkları üzerinde aynı etkiye sahip olması gerekir (Park, 2013). Lojistik regresyon analizi diskriminant analizi ve çok değişkenli regresyon analizinden ayrı olarak bağımsız değişkenin dağılımı ile ilgili varsayımlar yapma zorunluluğunu ortadan kaldırmaktadır.

Lojistik regresyon analizinde, bağımsız değişkenlerin normal dağılıma, doğrusallığa veya varyans-kovaryans matrislerinin eşit olmasına dair şartları sağlama zorunluluğu bulunmamaktadır.

1.3. K-en Yakın Komşu

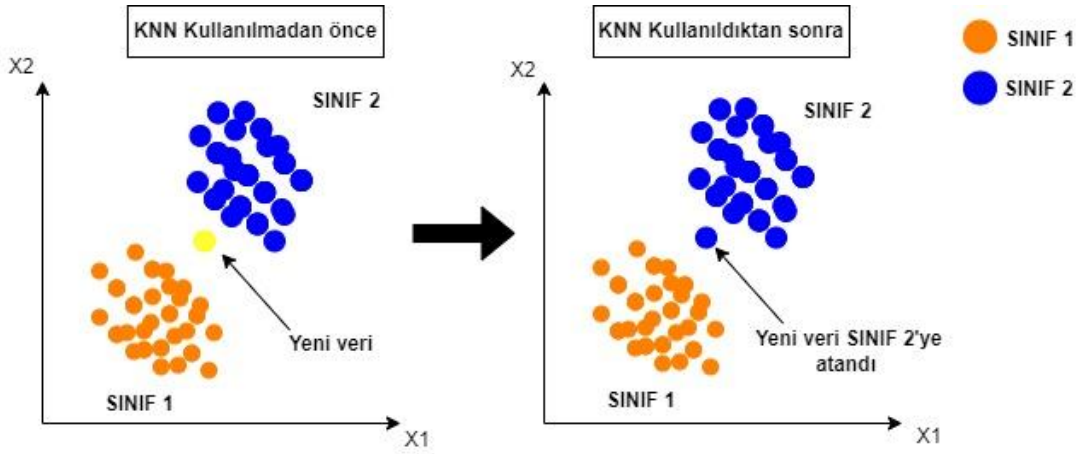
K- en yakın komşu (K- Nearest Neighbors/ KNN) denetimli bir makine öğrenmesi algoritmasıdır. Sınıflandırma ve regresyon problemleri için kullanılan Knn, büyük verilerle kullanılabilen, basit yapılı bir algoritmadır. İstatistik alanında 1951 yılında Joseph Hodges ve Evelyn Fix tarafından önerilmiş daha sonra ise Thomas Cover tarafından geliştirilmiştir. KNN algoritması, belirli sınıflara sahip bir örneklem setindeki gözlem değerlerinden, yeni bir gözlemin hangi sınıfa dahil edileceğini belirlemek için kullanılmaktadır (Arya vd., 1998). K-en yakın komşu algoritması; veri madenciliği, yapay zekâ, görüntü işleme, istatistik, örüntü tanıma, bilişsel psikoloji, biyoinformatik ve tıp gibi çeşitli alanlarda yaygın olarak kullanılmakla birlikte özellikle tıpta teşhis süreçlerinde karar vermeye odaklanan birçok uygulama bulunmaktadır. K en yakın komşu algoritması; sağlık, siber güvenlik ve veri madenciliği gibi pek çok alanda kullanılmaktadır. Bununla beraber büyük verilerle çalışarak sınıflandırma ve tahminleme işlemlerini gerçekleştirmektedir. Knn algoritması, sınıflandırma görevlerini yerine getirmek için uzaklık hesaplamalarını kullanır. Algoritmanın temel amacı, bireyleri veya nesneleri, bu özellikler temelinde önceden belirlenmiş sınıflara veya gruplara en doğru şekilde atamaktır. Ayrıca, algoritma yeni bir gözlemle karşılaştığında da sınıflandırma işlemini gerçekleştirebilmektedir. Sınıflandırılması istenen gözlem, öğrenme veri setindeki en yakın k komşusundan en çok benzer olanlarla aynı veri setinde sınıflandırılmaktadır.

K en yakın komşu algoritması; k değerine göre belirlenen komşular arasında yeni örneğin mevcut veri setindeki diğer örneklere olan yakınlığının ölçülerek seçilen k değeri kadar komşu örnekleri arasında en yakın olan sınıf belirlenir. Bu şekildeki sınıf belirlemeye ise ağırlıklı oylama adı verilir (Larose, 2005).

K en yakın komşu algoritmasında uygulanan adımlar;

1. İlk olarak k değeri belirlenir.

2. Mevcut veri setine eklenecek olan örneğin veri tipi ve probleme göre uzaklık ölçüsü belirlenir.
3. Yeni örneğin mevcut veri seti içindeki diğer örneklere uzaklığı/benzerliği belirlenen uzaklık ölçüsüne göre hesaplanır.
4. Bulunan uzaklıklar içerisinde belirlenen k değerine uygun en yakın olan komşular seçilir.
5. Son olarak yeni gelen örneğin sınıflandırılması yapılır.



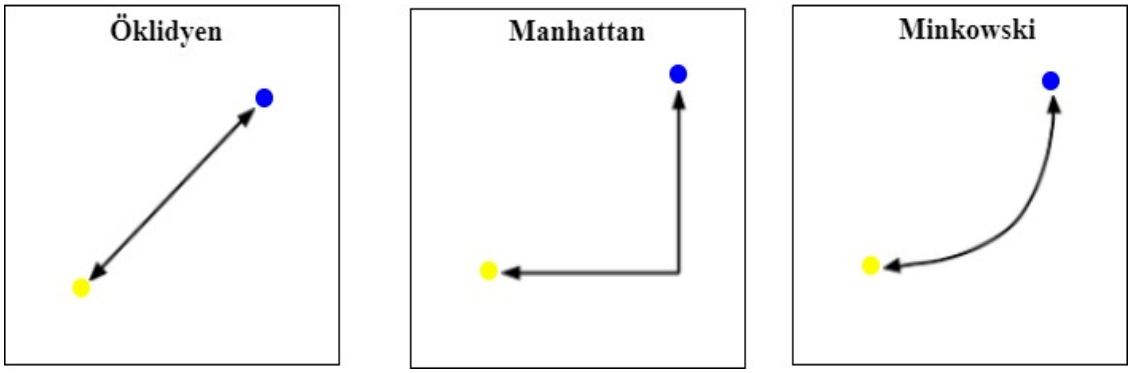
Şekil 2.4: K-En Yakın Komşular Algoritmasının Örnek Şeması

En yakın komşunun bulunması için k değerine tek sayılar verilmektedir. KNN ile ilgili gerçekleştirilen çalışmalarda k değerine 1,3,5 değerlerinin verildiğinde yüksek performans gösterdiği görülmüştür. Algoritma için k değerinin belirlenmesi önem taşıması ile beraber değerinin büyük seçilmesi birbirine benzemeyen örneklerin bir araya getirilmesine, k değerinin küçük seçilmesi ise birbirine benzeyen örneklerin dahi ayrı sınıflara dâhil edilmesine neden olabilmektedir (Silahtaroğlu, 2008).

Örneğin, k parametresine 3 değeri verilmesi durumunda k kendine en yakın bulduğu üç komşu seçiliyor. Sonrasında eğitim kümesindeki üç komşunun bulunduğu sınıflar hesaplanarak test kümesinde sınıflandırma işlemi yapılır. Sınıflandırma işlemi sonrasında ise k değerine en yakın olan sınıf seçilmesinin ardından veri etiketleme işlemi gerçekleştiriliyor. K-NN algoritmasının üç anahtar elementi vardır: sınıfı bilinen örnekler, örnekler arası mesafenin hesaplanması için uzaklık metriği ve en yakın komşu sayısını ifade eden k değeri $D = (x, y)$ kümesi sınıfı bilinen eğitim örnekleri, $z = (x', y')$ ise sınıfı belirlenmek istenen test örneği (x veri, y sınıf etiketi) olmak üzere k-NN

algoritması benzerliği (uzaklığı) hesaplamak için örnek uzayda z noktasının D kümesindeki tüm eğitim veri noktalarına olan uzaklıklarını hesaplar (Gök, 2017). Bu yöntemde öncelikle sınıflandırılacak test verilerinin eğitim verileri ile benzerliği hesaplanır. En yakın görünen k verisinin ortalaması ile belirlenen eşik değerine göre sınıflandırma yapılır. Yöntemin performansı, öğrenme kümesindeki en yakın komşu sayısı, eşik değeri, benzerlik ölçümü ve yeterli sayıda normal davranıştan etkilenir (Çalışkan ve Soğukpınar, 2008).

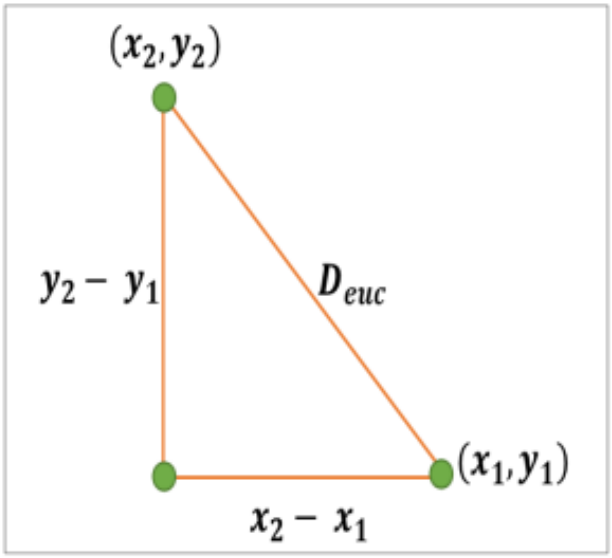
K-en yakın komşu algoritması için en önemli etkenlerden bir diğeri hangi uzaklık ölçüsünün kullanılacağıdır. Çünkü; algoritmanın başarısının bir ölçüsü olan doğruluk, farklı örnekler arasındaki uzaklıkların hesabı için kullanılan uzaklık ölçülerine de bağlıdır (Weinberger ve Saul, 2009). K-NN'de mesafe hesabı için kullanılan algoritmanın belirlenmesi ve komşu sayısı kritik optimizasyon noktalarıdır (Köklü, Sabancı ve Unlersen, 2016). Knn algoritmasında sınıflandırma performansının yüksek olabilmesi için k değerine uygun değer verilmesi ve doğru uzaklık hesaplamalarının seçilmesi önem arz etmektedir. K en yakın komşu algoritmasında uzaklık hesaplanması için kullanılan; Öklidyen, Manhattan, Chebyshev, Hamming, Minkowski, Jacckard uzaklık ölçülerinin sadece birkaçıdır. K en yakın komşular algoritmasının uzaklık ölçütlerinden bazılarının örnek gösterimi Şekil 2.6'da verilmektedir;



Şekil 2.5: K-En Yakın Komşu Uzaklık Metriklerinin Görselleri

Knn algoritmasında en yaygın kullanılan uzaklık hesaplama yöntemleri; Öklidyen, Manhattan ve Minkowski ölçütleridir. Bu uzaklık ölçütleri sürekli değişken olması durumunda kullanılmaktadır. Öklid uzaklık fonksiyonu daha çok veri setinde sürekli değişkenlerin bulunması durumunda, Manhattan uzaklık fonksiyonu ise daha çok veri setinde kategorik değişkenlerin bulunması durumunda kullanılmaktadır. L2 uzaklığı

olarak da adlandırılan Öklid uzaklığı, iki nokta arasındaki doğrusal uzaklığın bulunması için kullanılan temel uzaklık ölçütüdür. Sınıflandırma ve kümeleme yöntemlerinde sıklıkla kullanılmaktadır. Öklid uzaklığı Şekil 2.7'de verilmiştir.



Şekil 2.6: Öklid Uzaklığı

Kaynak: (Yousefi, Odabaş; Oktaş, 2020)

Matemetikte pisagor bağlantısı kullanılarak bulunan iki nokta arasındaki ölçüm birimidir. Buna göre iki boyutlu düzlemde iki nokta arasındaki mesafe basitçe iki noktanın x ve y koordinatlarının ayrı ayrı farklarının hipotenüs'üne eşittir (Atcılı, 2022). Öklid uzaklığının matematiksel gösterimi aşağıdaki gibidir;

$$L(x,y) = \sqrt{(\mathrm{x_1} - \mathrm{y_1})^2 + (\mathrm{x_2} - \mathrm{y_2})^2 + \cdots + (\mathrm{x_k} - \mathrm{y_k})^2} \quad \textbf{(2.6)}$$

$$\textbf{Öklid Fonksiyonu} = \sqrt{\mathbf{\Sigma_i}^{\mathrm{k}}{}_{=1}(\mathbf{x_i} - \mathbf{y_i})^2} \quad \textbf{(2.7)}$$

xi ve yi noktalarının aralarındaki mesafe hesaplanır, x için sınıflayıcı etiket görevini görmektedir. Veri setinde bulunan değişkenlerin sayısının ikiden fazla olduğu durumlarda Standardize Edilmiş Öklid Uzaklık Fonksiyonu kullanılmaktadır. Her bir değişken kendi içerisinde z dönüşümü uygulanarak standardize edilip eşitlik formüle yerleştirilmektedir. Böylelikle değişkenler arasındaki ölçüm farklılıkları ortadan kalkmış olmaktadır (Akyiğit ve Taşçı, 2022). Standardize edilmiş Öklid uzaklığı aşağıdaki gibidir;

$$\mathbf{d^2}_{\mathrm{st}} = (\mathbf{x_s} - \mathbf{y_t})\mathbf{V}^{-1}(\mathbf{x_s} - \mathbf{y_t})' \quad \textbf{(2.8)}$$

Manhattan (City block) uzaklığı; n boyutlu iki nokta arasındaki mesafe farkının mutlak değerlerinin toplamı ile bulunmaktadır. Manhattan uzaklığı; iki vektör arasındaki mesafelerin mutlak farklarından ortaya çıkan değerlerin toplanması ile hesaplanmaktadır. Herhangi iki nokta, P ve Q arasındaki Manhattan uzaklığı P= (x1, x2, …, xn) ve Q= (y1, y2, …, yn) olmak üzere (Taşçı ve Onan, 2016).

$$\sum_{i=1}^{k} |x_i - y_i| \qquad \textbf{(2.9)}$$

Minkowski uzaklığı; Öklid ve Manhattan uzaklık ölçütlerinin genelleştirilmiş halidir. Minkowski uzaklığına genelleştirilmiş mesafe hesaplama yöntemi denilmektedir. P-norm mesafesi olarak da bilinen Minkowski mesafesi, Öklid uzayında bir mesafe metriğinin genel biçimi olarak ifade edilir (Lu vd. 2016).

Minkowski mesafesinde p=1 değerinin verilmesiyle Manhattan uzaklığı elde edilir, p=2 değeri ile Öklid uzaklığı elde edilir, p = ∞ değerinin verilmesiyle ise Chebyshec mesafesi elde edilmektedir.

$$\textbf{Minkowski Fonksiyonu} = \sum_{i=1}^{k} (|x_i - y_i|^q)^{1/q} \qquad \textbf{(2.10)}$$

Örneğin Shahabi ve ark. (2002) ve Shahid ve ark. (2009), yaklaşık yol mesafelerini ve TT'leri tahmin etmek için Minkowski mesafesini kullanmışlardır (Lu vd. 2016). Minkowski mesafesinde, Mesafe metriğinin karşılaması gereken birkaç koşul vardır: (Sözkesen, 2021).

Negatif olmama: d (x, y)>= 0 Kimlik: d (x, y) = 0 ancak ve ancak x == y ise Simetri: d (x, y) = d (y, x) Üçgen Eşitsizliği: d (x, y) + d (y, z)>= d (x, z) ($\sum |xi - yi \mid n\ p$ i=1) 1/p (9)

1.4. Karar Ağacı

Karar ağacı; sınıflandırma ve regresyon yöntemlerinde problem çözümü için kullanılan denetimli (Supervised) bir makine öğrenmesi algoritmasıdır. Sınıflandırma ve regresyon ağaçları (CART) olarak da bilinen karar ağaçları, ağaç tabanlı algoritmaların en temelidir. İlk olarak 1960'li yılların başında Morgan ve Sonquist tarafından ortaya atılan karar ağacı yöntemi; Breiman ve ark. tarafından 1984 yılında "Classification and Regression Tree" adlı kitabında C&RT algoritmasından bahsetmişlerdir. 1986 yılında J.R Quinlen tarafından ID3 algoritması oluşturulmuş daha sonra ise ID3 algoritmasının

devamı olarak 1993 yılında C4.5 algoritmasını öne sürmüştür. DT, karakteristik uzayı özyinelemeli bir şekilde alt uzaylara bölerek ifade eden ve tahminin temelini oluşturan bir tahmin modelidir (Alfadhili, 2020).

Karar ağaçlarının indüksiyonu (TIDIDT), eğitim veri setini yinelemeli olarak bölümleyerek sıralı hiyerarşik bir yapı ile sınıflandırma kurallarının bir temsilini oluşturan denetimli bir öğrenme tekniğidir (Suyanto, 2018).

Kök düğüm, dallar ve yapraklar karar ağacı grafiğini oluşturur. Karar ağacı algoritmasında mevcut problem ağaç dallarına ayrılarak çözülmektedir. Sınıflandırma yapraklarda gerçekleşirken, her oluşumun sonuçları dallarda depolanır. Kategorizasyon kuralları oluşturulurken kök düğümden yaprak düğümlere giden yollar dikkate alınır (Nahzat ve Yağanoğlu, 2021). Karar ağacı algoritmaları gerçek hayatta; tıp, endüstri, görüntü işleme gibi alanlarda uygulanmaktadır. Karar ağacı algoritmaları ile çalışılırken aşağıdaki durumlara dikkat edilmelidir (Larose, 2005):

- Model oluşturmak için kullanılacak eğitim veri setinde sınıf etiketleri belirli olmalıdır.

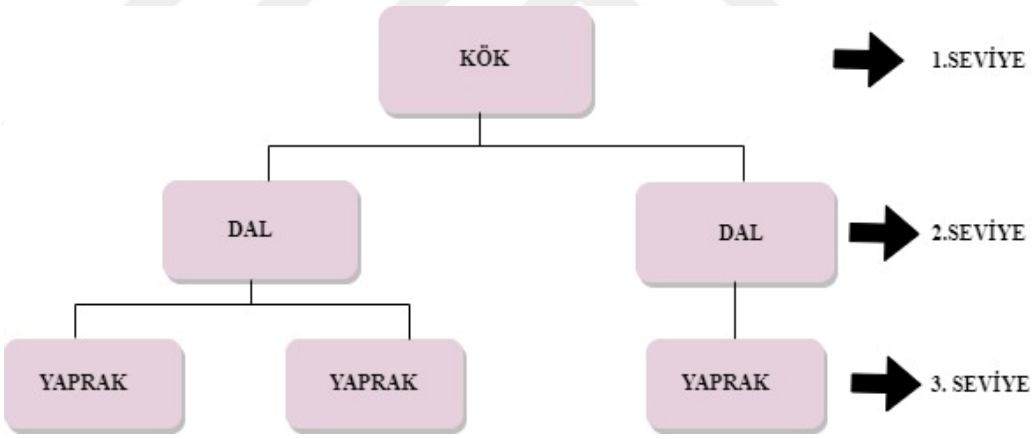
-Model kurulurken kullanılacak veri seti, tüm sınıfların iyi bir şekilde tanımlanabilmesi ve kuralların çıkarılabilmesi için yeterli olmalıdır.

- Sınıf etiketleri ayrık olmalı yani belirli bir sınıfa ait olup olmadığını belirtecek şekilde sınırlandırılmış değerlere sahip olmalıdır.

Veri setindeki değişkenleri kullanarak hedef değeri tahmin edebilecek bir ağaç modeli oluşturmayı ve görselleştirmeyi sağlar. Akış şemasını andıran bir yapıda olan karar ağacı, oluşturulan ağaç modeli ile kolaylıkla anlaşılabilen ve yorumlanabilen bir yapıdadır. Karar ağacı algoritmasında mevcut ağaç yapısına bakıldığında ağacın yaprakları sınıf etiketlerini, dalları ise özellikler üzerinde gerçekleştirilen işlemleri göstermektedir. Karar ağacı; kök düğüm (root node), dal (sub-ree) ve yaprak/terminal düğümü (leaf/terminal node) olmak üzere 3 bölümden oluşmaktadır. Yapılan işlemler kök düğümden yaprak düğüme doğru aşağı inilerek gerçekleşir ve IF-THEN yapısı kullanılmaktadır. Algoritma, veri kümesindeki her öznitelik için ayrı düğüm oluşturur ve en iyi sonucu veren öznitelik kök düğüm yerine geçer. En iyi sonucu veren özniteliği bulmak için ise eğitim veri kümesi içerisindeki verilerin istatistiksel hesaplamalarının

yapılması sonucu problemimize en faydalı olan öznitelik hangisi ise o seçilir. Kök düğüm; ilk düğümdür, bağımlı hedef değişkeni temsil eder. Yapraklar; çıktıyı dallar ise düğüm ile yaprak arasındaki akışı sağlayan alt ağaçtan oluşan yapılardır. Her bir dal test sonucunu vermektedir. Yapraklarda sınıf etiketi bulunur ayrıca bölünme işlemi gerçekleşmez bununla beraber yaprak düğümlerde sınıflandırma ve karar verme işlemi gerçekleşmekte ayrıca sınıf etiketi sonucu vermektedir.

Karar ağacı; bütünden parçaya doğru ilerleyen bir yapıdadır ve iki veya daha çok yaprak düğümden oluşabilmektedir. Sınıflandırma ve tahmin etmede karar ağaçlarının kullanılması, eğitim verisinden karar ağacı modelinin oluşturulması, bu modelin, test verisi kullanılarak uygun sınama ölçütleri aracılığıyla değerlendirilmesi ve ilgili modelin gelecekteki değerleri tahmin edilmesinde kullanılması şeklinde işlemektedir (Laencina vd., 2010; Birant, 2011; Onan, 2015). Basit bir karar ağacı yapısı Şekil 2.8'de verilmektedir.



Şekil 2.7: Karar Ağacı Akış Şeması

Karar ağacı ilk olarak veri setindeki en uygun öznitelik için düğüm oluşturur ve özniteliği kök düğüme yerleştirir. Kök düğüm seçilirken hangi öznitelik seçileceği konusu oldukça önem arz etmektedir, bu seçim için karar ağacı algoritması; bilgi kazancı, entropi, gini indekslerini kullanarak en uygun özniteliği kök düğüme yerleştirmektedir ardından mevcut problemi değerlendirmek için, koşullara uyan düğümler takip edilerek ilerlenir son olarak ise yaprak düğümünde hedef değişken uygun sınıfa atanmaktadır. Karar ağaçları algoritmaları arasında ID3, CART, C4.5 ve C5.0 sayılabilir. Bu algoritmalardan ID3 bilgi kazancına göre C4.5 ve C5.0 ise kazanım oranına göre dallanma gerçekleştirmektedir. C4.5, ID3 algoritmasının eksiklerini gidermek üzere

geliştirilmiş bir algoritma olup hem nümerik hem de kategorik değişkenler ile çalışmakta, kayıp değerler bu algoritma için sorun teşkil etmemektedir. C5.0 ise C4.5 algoritmasının ticari bir devamı niteliğinde olup bilgi www.rulequest.com/ adresinde yer almaktadır (Han vd., 2012). Karar ağaçlarında kök düğümün belirlenerek alt dallanmaların nasıl oluşturulacağı çeşitli hesaplamalara dayanarak yapılmaktadır. Bu yöntemlerden bazıları entropi, gini endeksi, sınıflandırma hatası endeksi, en küçük kareler sapması şeklinde sayılabilir (Akpınar, 2014). Öznitelik seçerken; Gini İndeksi, Kazanım Oranı (Gain Ratio) ve Bilgi Kazanımı (İnformation Gain) ölçütleri kullanılmaktadır. Scikit-learn kitaplığında karar ağaçları için bölme kriteri Gini Dizini'dir.

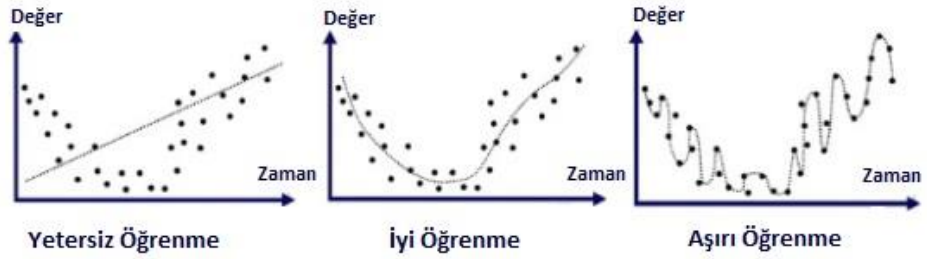
Testlerde Scikit-learn kütüphanesinin sağladığı karar ağacı yöntemi kullanılmıştır. Scikit-learn kitaplığı, ikili ağaçlar oluşturan CART algoritmasını kullanır (Karakaya ve Kazan, 2021).

Karar ağacı algoritmasında değişken seçimi aşamasında elamanların bulunduğu düğümde tek bir eleman kalmayana kadar atama işlemi devam etmektedir. Atama yapılacak değer kalmadığı zaman modeldeki döngü durur ve karar ağacı modeli oluşur. Karar ağacı modellemesi yapılırken büyük verilerle işlenmesi durumunda aşırı öğrenme gerçekleşebilmektedir. İki olası sonuç olması gerekirken bazen bir eksik veri olduğunda karar ağacı onu ayrı bir sonuç olarak görebilir ve üçüncü olası sonuç olarak karşımıza çıkartabilmektedir. Bu durumda aşırı öğrenmeyi (overfitting) engellemek için budama (Pruning) işlemi yapılmaktadır. Bu işlem karar düğümündeki alt düğümün kaldırılmasına denmektedir.

1.4.1. Aşırı Öğrenme (Overfitting)

Aşırı öğrenme durumu; modelin eğitim ve test olarak ayrılması ve çalıştırılmasının ardından eğitim kümesindeki verilerde ezberleme gerçekleşmesi durumudur. Aşırı öğrenme durumu karmaşıklığa neden olarak modelin yanlış tahminlenmesine neden olur. Modelde aşırı öğrenme olduğunu anlamanın yolu ise öncelikle test ve eğitim oranlarını uygun vermektir. Örneğin %70 eğitim %30 test ya da %80 eğitim %20 test gibi orantılı değerler verilmelidir. Ardından eğitim ve test kümeleri ayrı ayrı çalıştırılarak doğruluk oranları kontrol edilmelidir. Eğer eğitim kümesindeki doğruluk oranı ile test veri kümesinin doğruluk oranları arasında fazlasıyla fark olması durumunda aşırı öğrenme gerçekleştiği anlamına gelmektedir. Ayrıca veri setinde bulunan gereksiz değişkenlerde

aşırı öğrenme gerçekleşmesine neden olmaktadır. İstatistiksel modellemede veri setinin tahminleme aşamasında 3 durum oluşabilmektedir bunlar; Underfitted (Yetersiz öğrenme), Overfitted (Aşırı öğrenme), Good Fit/Robust (İyi öğrenme)' dir. Aşağıdaki Şekil 2.9 'da öğrenme türleri verilmiştir.



Şekil 2.8: İstatistiksel Veri Modellemede Öğrenme Türleri

Aşırı uyum durumunda, bir model eğitim verilerinin gürültüsünü ezberler ve temel kalıpları yakalayamaz (Kumar, 2021). Aşırı öğrenmeyi engellemek için birkaç yöntem bulunmaktadır. Bu yöntemlerden en fazla kullanılan ise budama yöntemidir. Budama yöntemi; karar ağacındaki gereksiz büyüme, dallanmayı engellemek ve olası hatalı sonuçları engellemek için uygulanan bir tekniktir. Budama yöntemi; Ön budama (Pre-pruning) ve son budama (Post-pruning) olmak üzere ikiye ayrılmaktadır. Ön budama; karar ağacının aşırı öğrenmesini erken durdurmayı sağlamaktadır bunun için scikitlearn kütüphanesinde max_depth (maksimum derinlik), min_samples_leaf (minimum yapraktaki örnek) gibi etkili parametreler kullanılmaktadır. Karar ağacındaki maksimum derinlik değeri 5 olmalıdır daha fazla olması durumunda modelde aşırı öğrenme gerçekleşmektedir. Son budama ise; karar ağacının büyümesine ve dallanmasını sağlar ardından modeldeki aşırı öğrenmeyi engellemek için budama işlemini sonradan gerçekleştirir. Son budama işleminde hata tahmini şu şekilde hesaplanmaktadır;

$$e = \left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Big/ \left(1 + \frac{z^2}{N} \right) \tag{2.11}$$

f: Eğitim verilerinde meydana gelen hata

N: Yaprağın kapsadığı örneklerin sayısı

z: Normal dağılım

Karar ağacı algoritmasının avantajları;

- Veri hazırlama süresi azdır

- Görselleştirilmesi, yorumlanması ve okunması oldukça kolaydır

- Güvenilir bir makine öğrenmesi algoritmasıdır

- Sayısal ve kategorik değişkenleri işleyebilir

Karar ağacı algoritmasının dezavantajları;

- Eğitim verisinin az olması durumunda hatalı sonuç verebilir

- Ezbere öğrenme gerçekleşebilir bu durumda ise budama yöntemi ile sorun çözülmektedir ancak budama algoritmalarında ise maliyet artışı gerçekleşir

Karar ağaçları kategorik ve sayısal/sürekli verilerle işlem yapmaktadır. Kategorik veriler işlenmesi durumunda sınıflama ağacı (classification tree), sayısal/sürekli verilerin işlenmesi sonucu ise regresyon ağacı (regression tree) kullanılmalıdır. Sürekli öznitelikler için en küçük kareler yöntemi, kategorik değişkenler için ise entropi, gini, sınıflandırma hatası yöntemleri kullanılmaktadır. Entropi (ID3); 0 ile 1 arasında sonuç vermektedir. Entropi değeri 0'a yaklaştığında belirsizlik azalırken 1'e yaklaştığında belirsizlik artmaktadır. Eğer entropi değeri 0 ise o dal yaprak düğümüdür ancak entropi değeri 0'dan büyük ise dalın bölünmesi gerekmektedir. Entropi hesaplaması en çok CART, ID3, C4.5 algoritmalarında kullanılmaktadır. Karar ağacı algoritmalarının birçok çeşidi bulunmaktadır. Başlıca kullanılan karar ağaçları ise; CART, CHAİD, ID3, C4.5 ve C5.0 algoritmalarıdır.

1.4.2. CART (Classification and Regression Trees) Algoritması

Makine öğrenme yöntemlerinden olan karar ağacı algoritmasının bir çeşidi ise sınıflandırma ve regresyon ağacı tahminleme algoritmasıdır. CART analizi, ikili özyinelemeli bölümlemenin bir biçimidir (Lewis, 2000). CART algoritması; veri kümesinde bulunan değişkenlerin sınıflandırılmasında hangi özelliğe göre dallanacağını belirlemektedir. Veri kümesindeki hedef değişkenin kategorik olması durumunda sınıflandırma yöntemi, sürekli olması durumunda ise regresyon yöntemi

kullanılmaktadır. CART algoritması ilk olarak Leo Breiman, Jenome Friedman, Richard A. Olshen ve Charles J. Stone tarafından 1984 yılında ortaya atılmıştır.

Avantajları;

- Aykırı değerlerden etkilenmez
- Sayısal ve kategorik değişkenleri işler
- Ortaya çıkan modellerin anlaşılması ve yorumlanması diğer sınıflandırma yöntemlerine göre oldukça basittir
- Denetim oldukça azdır
- Eksik değerlerin işlenmesi kolaydır

Dezavantajları;

- Kararsız ve karmaşık bir yapıya sahiptir bu yüzden hatalı sonuçlar verebilir.
- Aşırı uyum gösterebilir.
- Çarpık yapıdadır

1.4.3. CHAİD Algoritması

CHAİD (ki-kare otomatik etkileşim dedektörü) algoritması ilk olarak 1980 yılında G. V. Kass tarafından ortaya atılmıştır. Kass bu yöntem ile karar ağacı yönteminde en iyi bölmeyi bulmayı amaçlamıştır. CHAİD (Ki-kare otomatik etkileşim dedektörü), sürekli ve kategorik değişkenleri işleyebilmekte ve değişkenlerin aralarındaki ilişkiyi gösterebilmektedir. Bölünen dal sayısı tahmin edicinin kategori sayısına göre değişiklik göstermektedir. CHAİD algoritması kullanımında bölme işlemleri ki-kare testi ile sağlanmaktadır. CHAİD ile diğer yöntemler arasındaki en önemli farklılıklarından birisi, ağaç türetimidir.

ID3 (Tekrarlı İkilikçi Ağaç), C4.5 ve CART ikili ağaçlar türetirken, CHAİD ikili olmayan çoklu ağaçlar türetir (Berson vd.,1999). Diğer algoritmalarda uygun bölmeleri seçmek için Gini, Entopi yöntemleri kullanılırken CHAİD algoritmasında ki-kare testi kullanılmaktadır.

1.4.4. ID3 Algoritması

(Tekrarlı İkilikçi Ağaç 3) algoritması 1986 yılında J. Ross Quinlan tarafından geliştirilen basit yapılı bir karar ağacı algoritmasıdır. Özelliklerin değerlerini test ederek

nesnelerin sınıflandırılmasını belirler. Belirli bir veri için, bir dizi nesneden ve bir özellik belirtiminden başlayarak yukarıdan aşağıya doğru bir karar ağacı oluşturur.

Ağacın her düğümünde, bir özellik şuna dayalı olarak test edilir: bilgi kazancını en üst düzeye çıkarmak ve entropiyi en aza indirmek ve sonuçlar nesne kümesini bölmek için kullanılır. Bu işlem, belirli bir alt ağaçtaki küme homojen olana (yani, aynı kategoriye ait nesneleri içeren) kadar yinelemeli olarak yapılır (Singh ve Gupta, 2014). ID3 algoritmasında entropi ve bilgi kazancı hesaplamasına göre yukarıdan aşağı doğru dallar oluşmaktadır.

1.4.5. C4.5 ve C5.0 Algoritmaları

C4.5, Ross Quinlan tarafından geliştirilmiş bir karar ağacı oluşturmak için kullanılan bir algoritmadır (Quinlan, 1993). C4.5 ve C5.0 algoritmalarında kazanım oranı hesaplamasına göre dallanma gerçekleşmektedir. C4.5 algoritması, ID3 algoritmasının iyileştirilmiş hali olmakla birlikte sürekli ve kategorik değişkenleri işleyebilmektedir. C4.5'in ID3 algoritmasında yaptığı iyileştirmelerden bir diğeri ise hem sürekli hem de ayrık öznitelikleri işleme sürekli öznitelikleri işlemek için C4.5 bir eşik oluşturur ve ardından listeyi öznitelik değeri eşiğin üzerinde olanlara ve eşikten küçük veya ona eşit olanlara böler (Quinlan, 1996). C4.5 algoritması eksik değerler bulunan veri setlerinde çalışabilmektedir.

C5.0 algoritması ise C4.5 algoritmasının iyileştirilmiş hali olup hız, az bellek kullanımı ve karar ağacında budama yaparak daha başarılı sonuçlar vermektedir. C5.0, kategorik değişkenlerin tahminlenmesinde kullanılan, öğrenme süresi kısa ve kolay yorumlanabilir bir algoritmadır. Karar ağacı algoritmasında uygun düğümün seçilmesini belirlemek için; Gini index, Entropi, İnformation gain, Gain ratio, ki-kare istatistiği ölçütleri kullanılmaktadır. Bu ölçütlerin ayrıntılı açıklamaları aşağıda verilmiştir.

1.4.6. Gini İndeksi

Gini indeksi; karar ağacı algoritmalarının kök düğümünden itibaren bölünmesi için kullanılan bir hesaplama yöntemidir. Kategorik değişken bulunan veri kümelerinin hesaplanmasında kullanılmaktadır.

CART (sınıflandırma ve regresyon) algoritması öznitelik seçimi hesaplamasında ikili bölmeler oluşturulması için kullanılmaktadır. Gini indeksi; karar ağacındaki

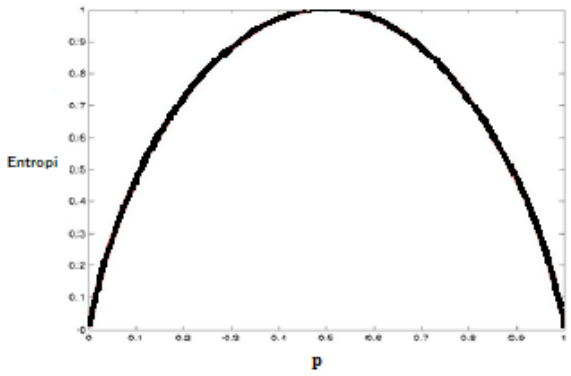
safsızlığı ölçmeyi sağlamaktadır. Gini değeri 1'e ne kadar yakın bir değer ise saf olma durumu aynı oranda yüksektir bununla birlikte homojenliği de yüksek olmaktadır. Gini indeksinin formülü aşağıdaki gibidir;

$$\mathbf{Gini} = \mathbf{1} - \sum_{i=1}^{C} (\mathbf{p}_i)^2 \qquad \textbf{(2.12)}$$

Gini indeks formülünde bulunan c; sınıf sayısını, pi; sınıfların arasındaki olasılığı temsil etmektedir. Gini değeri; veri kümesindeki her bir sınıfın başarılı ve başarısızlık olasılık değerlerinin kareleri alınarak birbirlerinden çıkarılmasıyla hesaplanmaktadır (p^2+q^2).

1.4.7. Entropi

Karar ağaçlarında kullanılan entropi; mevcut veri kümesindeki safsızlığı ölçmektedir. Entropi ölçütü veri kümesi kategorik olan değişkenlerin bulunması durumunda kullanılmaktadır. Safsızlık; belirsizlik anlamına gelmektedir, belirsizlik arttığında entropi değeri de aynı oranda artmaktadır belirsizliğin azalması durumunda ise entropi değeri de düşmektedir. Entropi ölçütü; ID3 ile C4.5 algoritmaları tarafından kullanılmaktadır.



Şekil 2.9: Entropi Grafiği

Entropi Formülü: $\boldsymbol{E}(\mathbf{S}) = \sum_{i=1}^{c} - \mathbf{p}_i \log_2 \mathbf{p}_i$ **(2.13)**

Formüldeki c: veri kümesinde bulunan sınıf sayısını, pi ise sınıfların aralarındaki olasılığı temsil etmektedir. Entropi değerindeki azalmalara bilgi kazancı adı verilmektedir. Bilgi kazancı, veri kümesindeki öznitelik değerlerinin entropi değeri ile bölümü sonucunda ortaya çıkan entropinin aralarındaki farkının hesaplanması sonucunda ortaya çıkmaktadır. Karar ağaçlarında Entropi hesaplanırken 0 ile 1 arasındaki değerleri

almaktadır. Entropi değeri 0 olan dala yaprak düğümü denir. Eğer entropi 0'dan büyük ise tekrar bölünmesi gerekmektedir.

Entropi değerinin safsızlık oranının yüksek değerden düşük değere geçmesi sonucu aradaki değere bilgi kazancı adı verilmektedir bu durumda karar ağacı veri kümesindeki değişkenlerin alacakları değerler daha kolay tahmin edilebilmektedir. Karar ağacı algoritmalarında bilgi kazanımı ölçütünü hesaplamak için entropi kullanılmaktadır. Veri kümesindeki özelliklerin ayrı ayrı entropisi hesaplanır ve bu değerlerdeki azalma hesaplanır ve ortaya çıkan değerler ile bilgi kazanımı hesaplanmaktadır.

1.4.8. Bilgi kazancı (İnformation Gain)

Bilgi kazanımı (IG); karar ağacında düğümün bölünmesini sağlayarak veri kümesinde bulunan özniteliklerin içerdiği bilgiyi tahminlemesini sağlayan ölçümdür. ID3 karar ağacı algoritmasında kullanılmaktadır. Veri kümesindeki değişkenlerin değerlerine bölünmeden önceki entropi ile bölünmesinden sonraki entropi arasındaki farka bilgi kazancı denmektedir. Entropinin yüksek değerden düşük değere geçmesi durumudur.

Entropi kullanılarak hesaplanan bilgi kazanımı hangi öznitelikte dallanma olacağını hesaplamaktadır. Bilgi kazancı, aşağıdaki formül ile hesaplanmaktadır;

$$\mathbf{Gain(T, X) = Entropy(T) - Entropy(T, X)} \quad \mathbf{(2.14)}$$

T: Bağımlı (Hedef) değişken

X: Bağımsız değişken

Hesaplamanın ardından en yüksek bilgi kazanımı (IG) değeri bulunup bu değere göre dallanma oluşmaktadır. Bilgi kazanımı 4 adımdan oluşmaktadır;

1. Hedef değişkenin entropisinin hesaplanır

2. Veri setinin farklı özelliklere bölünür

3. Her dal için ayrı bir entropi hesaplanır

4. En son hesaplanan entropi değeri ile ilk entropi değeri birbirinden çıkarılarak bilgi kazanımı değeri hesaplanmış olur ve en büyük bilgi kazanımı değeri karar düğümü olarak seçilir.

1.4.9. Kazanç Oranı (Gain Ratio/ GR)

Bilgi kazanımı ile benzer yapıda olan kazanç oranı; karar ağacı algoritmasında bölünme sonucu en yüksek sonucu veren değer öznitelik olarak seçilmektedir.C4.5 karar ağacı algoritmasında kullanılmaktadır. Kazanç oranı (GR) aşağıdaki gibi hesaplanmaktadır;

$$\textbf{Gain Ratio} = \frac{\textbf{Information Gain}}{\textbf{SplitInfo}} = \frac{\textbf{Entropi}(\text{önce}) - \sum_{j=1}^{K} \textbf{Entropi}(j,\text{sonra})}{\sum_{j=1}^{K} wj \log_2 wj} \quad \textbf{(2.15)}$$

Kazanç oranı= Bilgi kazanımı/ Bölünme bilgisi

GR hesaplamasında; önce, veri kümesinde bölünme gerçekleşmeden önceki entropi, sonra; bölünme gerçekleştikten sonraki entropi değeridir. Kazanç oranı; bilgi kazancının splitinfo değerine bölünmesiyle hesaplanmaktadır. SplitInfo, ağırlıkların logaritmasıyla çarpılan ağırlıkların toplamı olarak tanımlanır; burada ağırlıklar, geçerli alt kümedeki veri noktası sayısının üst veri kümesindeki veri noktası sayısına göre oranıdır (Silipo, R. 2019).

Kazanç oranının hesaplanma adımları;

1. İlk olarak bağımlı değişkenin entropisi hesaplanmaktadır

2. Bilgi kazancı değeri hesaplanır

3. Bölünme bilgisi hesaplanır

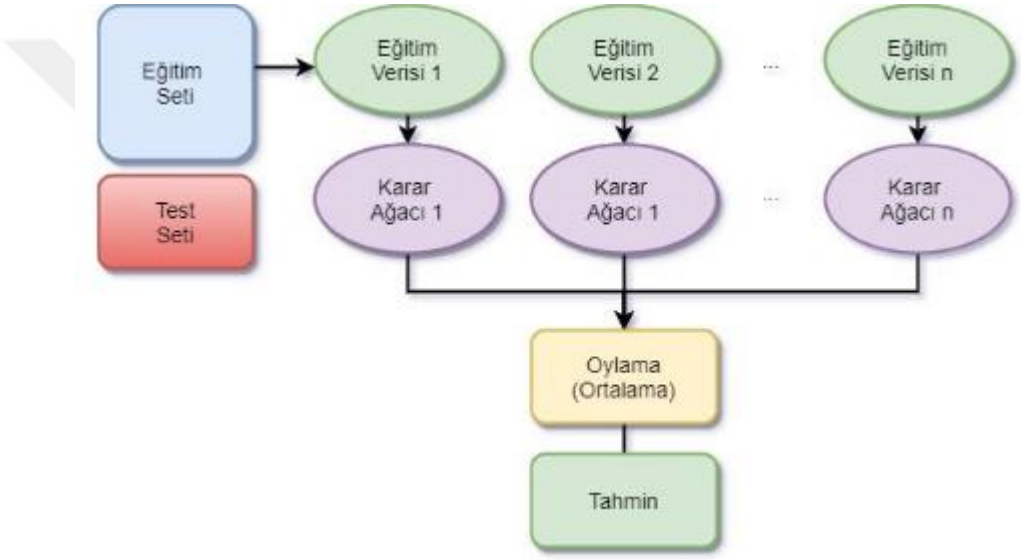
4. Kazanç oranı formülü uygulanır

5. Veri kümesindeki değişkenlerin ayrı ayrı kazanç oranları karşılaştırılarak en yüksek orana sahip olan değişken seçilir.

1.5. Rastgele Orman Algoritması

Rastgele orman veya Rastgele karar ormanları algoritması, 2001 yılında Leo Breiman tarafından geliştirilmiş denetimli bir makine öğrenmesi sınıflandırma algoritmasıdır. Regresyon ve sınıflandırma yöntemlerinde kullanılan rastgele orman algoritması temelinde karar ağaçları bulunan bir sınıflandırıcıdır. Bu yöntemde, Bagging yöntemi Leo Breiman tarafından 1996 yılında geliştirilen ve önerilen rastgele alt grupları

seçmek için kullanılan Rastgele Altuzay tekniği 1998 yılında Tin Kam Ho tarafından birleştirildi (Şahin ve İçen, 2021). Rastgele orman yönteminde {f (x,) j=1, ...} şeklinde ağaç tipi sınıflandırıcılar kullanır. Burada x giriş verisidir; rastgele vektörü temsil eder (Breiman, 2001). Rastgele orman, sınıflandırma algoritmaları içerisinde en popüler olan ve yüksek performans gösteren algoritmalardandır. Kategorik ve sürekli veri setleri için kullanılabilen rastgele orman algoritması büyük verileri kolaylıkla işleyebilmektedir. Rastgele orman bir topluluk öğrenme yöntemi olmakla birlikte, birden çok karar ağacının tahminlenmesi sonucunda oluşmaktadır.



Şekil 2.10: Rastgele Orman Algoritması

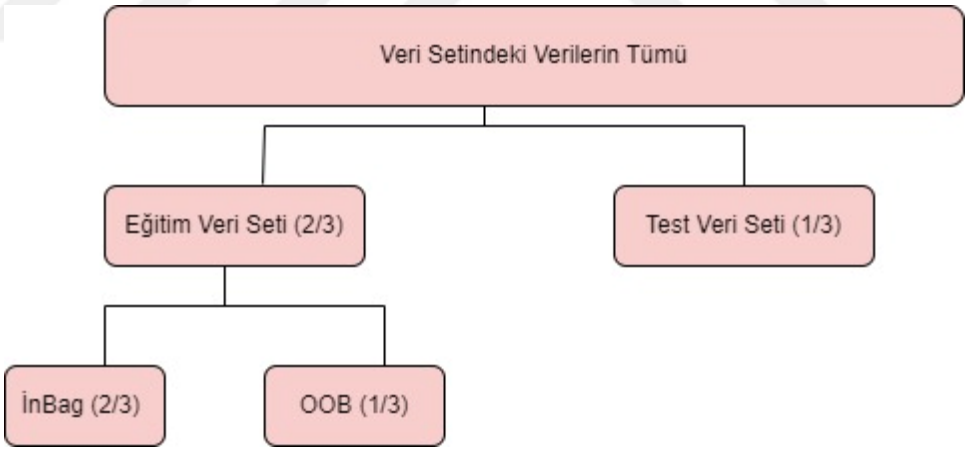
Rastgele orman yönteminde eğitim verileri için genellikle bagging (torbalama) ve bootstrap (önyükleme) teknikleri uygulanmaktadır. Bagging yönteminde karar ağaçları veri setinden bootstrap tekniği ile örneklem seçilerek birbirinden bağımsız olarak oluşturulmaktadır. Random Subspace yöntemi ise, karar ağacının her düğümünde en iyi dallara ayırıcı değişkenin, tüm değişkenler arasından rastgele (random) seçilen az sayıdaki değişken içinden seçilmesidir (Akman, 2010).

RF algoritması, veri kümesinden farklı alt kümeler seçer ve farklı karar ağaçları oluşturur ve her karar ağacıyla ayrı bir tahmin veya sınıflandırma yapar. RF algoritması kullanılarak bireyleri sınıflandırmak amaçlanıyorsa, en çok oy alan birey tahminler arasından seçim yapmalı, amaç tahmin yapmak olduğunda ise karar ağacı tahminlerinin ortalaması kullanılmalıdır. RF algoritması, tüm değişkenler arasından en iyi dalı seçip

her düğümü dallara bölmek yerine, her düğümde rastgele seçilen değişkenler arasından en iyisini kullanarak düğümleri dallandırır (Breiman, 2001).

Rastgele orman modelinin çalışma aşamaları;

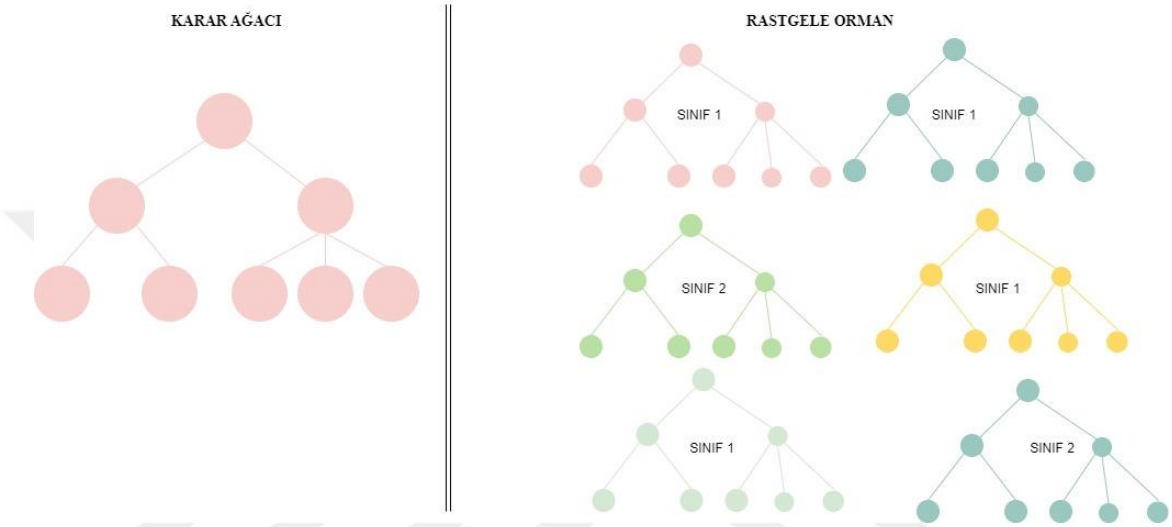
1) Rastgele ormanın oluşması için 2 parametre kullanılmalıdır bunlar; değişken sayısı (m) ve ağaç sayısı (N) dır.
2) Mevcut veri setindeki eğitim veri setinin 2/3'ü öğrenme verisi (InBag) ağaç oluşturmak için ayrılırken, 1/3'ü ise OOB (test verisi) ağaçların performansının test edilmesi için ayrılmaktadır.
3) Parametreler oluşturulup eğitim ve test verileri ayrılmasının ardından oluşturulan ağaçlar için n adet önyüklemeli (bootstrap) örneklemler oluşturulur.
4) Budama işlemi olmadan ağaç, ek düğümlere ayrılarak gelişmeye başlar
5) Ağaçlarda bulunan düğümlerdeki değişkenlerden m adet rastgele örnek seçilir ardından n adet ağacın tahminlerine bakılarak bir tahmin yapılır ve oylamada en yüksek oyu veren sınıf belirlenir.



Şekil 2.11: Rastgele Orman Çalışma Modeli

Rastgele orman algoritması, büyük veri tabanlarında yüksek başarı ve verimli bir şekilde çalışabilmektedir. Rastgele orman algoritması, karar ağaçlarına kıyasla hata oranını daha doğru bir şekilde tahmin eder. Daha spesifik olarak, hata oranının ağaç sayısı arttıkça her zaman yakınsadığı matematiksel olarak kanıtlanmıştır (Breiman 2001). Rastgele orman, Random Subspace (Rastgele Altuzay) ve Torbalama (Bagging) yöntemi methodunun bir araya getirilmesiyle oluşturulmuş bir algoritmadır. Karar ağaçlarında budama işlemi yapılırken rastgele ormanda budama yapılmamaktadır. Ağaç tabanlı

sınıflandırıcılar için budama yöntemi seçimi sınıflandırma algoritmalarının performansını etkilemektedir. Karar ağaçları sonuç olarak ağaç çıktısı verirken rastgele ormanda ağaç çıktısı verilmemektedir. Karar ağacına göre daha iyi performans gösteren rastgele orman, karar ağaçlarındaki fazla öğrenme olasılığını düzeltir. Tek tek ağaçların gücünü arttırmak, orman hata oranını azaltır.



Şekil 2.12: Karar Ağacı ve Rastgele Orman Algoritmaları

Her karar ağacını oluşturmak için, herhangi bir değişiklik olmaksızın (normalde 2/3) rastgele seçilmiş N gözlem alt kümesi kullanılır. Daha istikrarlı modeller ve değişken önem tahminlerinin doğru üretimi için çok sayıda ağaç (ntree) önerilir. Ancak, çok sayıda ağaç (ntree) daha fazla belleğe ve daha uzun işlem sürelerine yol açar. Küçük veri kümeleri için 50 ağaç yeterli olabilirken, daha büyük veri kümeleri için 500 veya daha fazlasını gerektirebilir. Bu nedenle, sahada deneme yanılma yoluyla parametre ayarı yapılabilir (Cutler vd., 2007). Eğitimde, Rastgele Orman algoritması birden çok CART benzeri ağaçlar oluşturur (Breiman vd., 1984). Rastgele orman algoritmasında düğümlerde bulunan değişkenler arasından en iyi sonucu veren dalı bulmak için CART yöntemi kullanılmaktadır. Cart yöntemi ile düğümde bulunan dallar gini katsayısı kullanılarak iki gruba ayrılır.

GINI katsayısının küçük değere sahip olması durumunda sınıfın homojen, büyük bir katsayıya sahip olması durumunda ise sınıfın heterojen olduğunu göstermektedir. Gini katsayısı sıfır olması durumunda ise bölünme sona ermektedir. Gini katsayısı aşağıdaki denklem ile hesaplanmaktadır.

$$\Sigma\,\Sigma\,j \neq i(f(Ci,T)/T)\ (f(Ci,T)/T) \qquad (2.16)$$

Bu denklemde; T: eğitim veri setini, Ci: Ait olunan sınıfı, (Ci/T)T ise seçilen örneğin Ci sınıfına ait olma olasılığını vermektedir.

Rastgele orman algoritması, topluluktaki ağaçlar içinde daha az korelasyon elde edildiği için oluşan modelin doğruluğu daha yüksektir (Suchetana vd., 2017). Rastgele orman algoritması, gerçek dünya problemlerinde, bankacılık, sağlık, e-ticaret başta olmak üzere pek çok sektörde müşteri davranışını tahminleme ve hastalıkların tahminlenmesi gibi çeşitli alanlarda kullanılmaktadır.

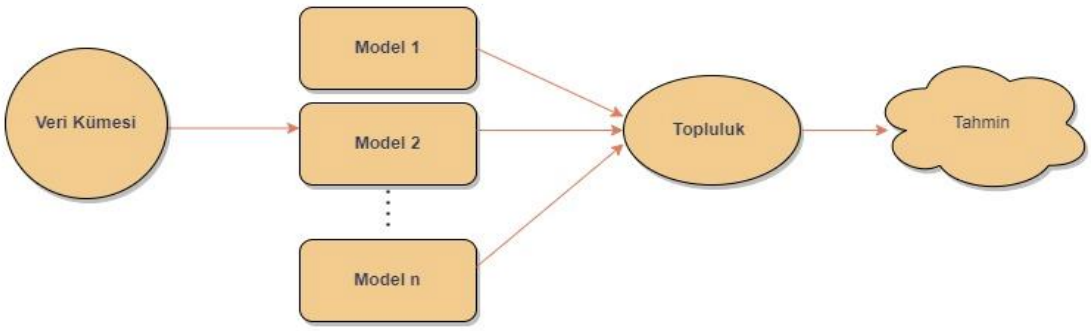
Rastgele Orman algoritmasının avantajları:

- Küçük ve büyük bütün veri tabanları ile etkili bir şekilde çalışabilir
- Aşırı öğrenme problemi yoktur
- Budama işlemi gerçekleştirmez
- Verileri hızlı bir şekilde işleyebilir.
- Eksik değerler ile çalışabilir

Rastgele Orman algoritmasının dezavantajları:

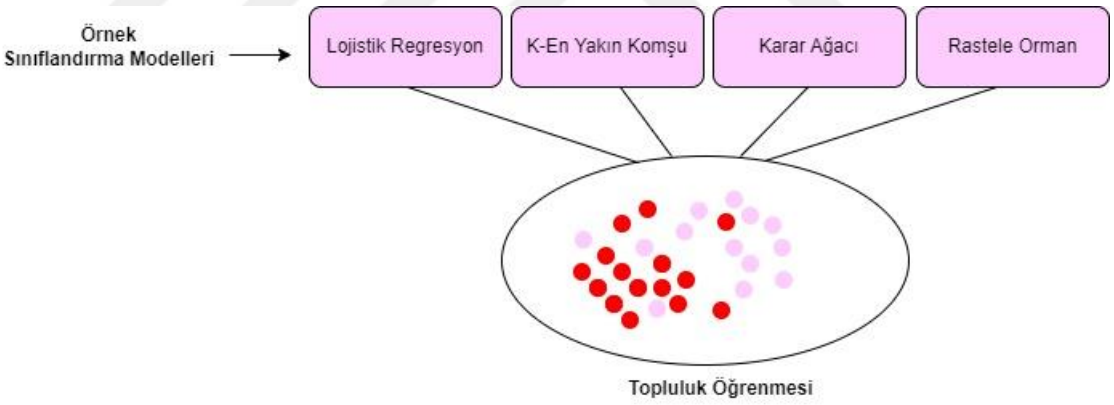
- Çok fazla sayıda ağaç kullanıldığı için tahminlemede yavaştır
- Yorumlanması zordur
- Karmaşık bir yapıya sahiptir

Rastgele Orman algoritması, topluluk öğrenimi (Ensemble Learning) yöntemini kullanmaktadır. En sık kullanılan topluluk öğrenimi yöntemleri; Torbalama ve Arttırma yöntemleridir. Topluluk öğrenmesi, birden çok tahmine dayalı modelin sonucunu birleştirip tek bir tahmin elde etmeyi sağlayan yöntemdir. Topluluk öğrenmesi yönteminde kullanılan tahmincilere ise "Ensemble (Topluluk öğrenmesi)" adı verilmektedir. Topluluk öğrenmesi yönteminin basit gösterimi Şekil 2.14 'te verilmiştir.



Şekil 2. 13: Topluluk Öğrenme (Ensemble Learning) Yapısı

Topluluk öğrenme yönteminin ortaya çıkmasındaki amaç, makine öğrenme algoritmalarının performanslarının iyileştirilmesi ve geliştirilmesidir. Topluluk öğrenimi, birden çok öğrencinin aynı sorunu çözmek için eğitildiği bir makine öğrenimi paradigmasıdır. Eğitim verilerinden bir hipotez öğrenmeye çalışan sıradan makine öğrenimi yaklaşımlarının aksine, topluluk yöntemleri bir dizi hipotez oluşturmaya ve bunları kullanmak için birleştirmeye çalışır (Zhou, 2021).



Şekil 2.14: Topluluk Öğrenmesi Yapısı

Makine öğrenmesi sınıflandırma modelleri tek başına çalıştırıldığında iyi performans gösterebilir ancak topluluk öğrenmesinde, veri setine uygun birden çok sınıflandırma modeli toplanıp bir modelleme yaptığında ortaya daha iyi bir performans çıktığı görülmektedir bununla birlikle topluluk öğrenmesi her zaman en yüksek sonucu vermeyebilmektedir bunda yüksek etken olan durum ise veriler için uygun sınıflandırma modellerinin kullanılmaması olmaktadır.

Topluluk öğrenmesi, birden fazla eğitilmiş olan modellerin bir araya getirilerek tahmin sonuçlarının birleştirilmesini ve tek bir tahmin sonucu üretmesini sağlamaktadır.

Topluluk öğrenme modelleri; Bagging (Torbalama), Boosting (Arttırma), Stacking (Yığma) ve Blending (Harmanlama) olmak üzere dörde ayrılmaktadır. Torbalama yöntemi, tekli sınıflandırıcıları eğitmek için seçilen örneklerin eğitim verilerinin Bootstrap ile yinelenmesidir. Bu, her örneğin her eğitim veri setine dahil edilme şansının eşit olduğu anlamına gelir. Arttırma yönteminde ise, her son sınıflandırıcı daha önce üretilmiş sınıflandırıcılar tarafından yanlış sınıflandırılmış örneklere odaklanır.

Bu nedenle doğru sınıflandırmaya daha büyük ağırlık değeri atanırken, iyi sınıflandırılmamış olana daha düşük ağırlık değeri atanır (Schapire, 1990, Karaoğlu ve Hayrettin, 2020). Bu tez çalışmasında banka müşteri kaybı veri setinin tahminlenmesi için Boosting algoritmalarından; LightGBM (Hafif gradyan Arttırma) ve CatBoost (Kategorik güçlendirme) modelleri ile çalışılmıştır.

1.6. Boosting (Arttırma) Algoritması

Boosting algoritması; Kearns ve Valiant (1988, 1989)'ın sorduğu "Bir grup zayıf öğrenici tek bir güçlü öğrenici yaratabilir mi?" sorusundan yola çıkarak Robert Schapire tarafından 1989 yılında geliştirilmiştir. Boosting kelimesi "Arttırma/Yükseltme" anlamlarına gelmektedir. Sınıflandırma aynı zamanda regresyon yöntemleri için de kullanılabilmektedir. Boosting; zayıf öğrenicilerden güçlü öğrenici oluşturmayı sağlayan bir topluluk öğrenmesi (Ensemble Learning) yöntemi olan makine öğrenmesi algoritmasıdır. Arttırma algoritmasının matematiksel formülü aşağıdaki gibidir;

$$\mathbf{F(x)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \mathbf{w_i h_i(x)} \qquad \textbf{(2.17)}$$

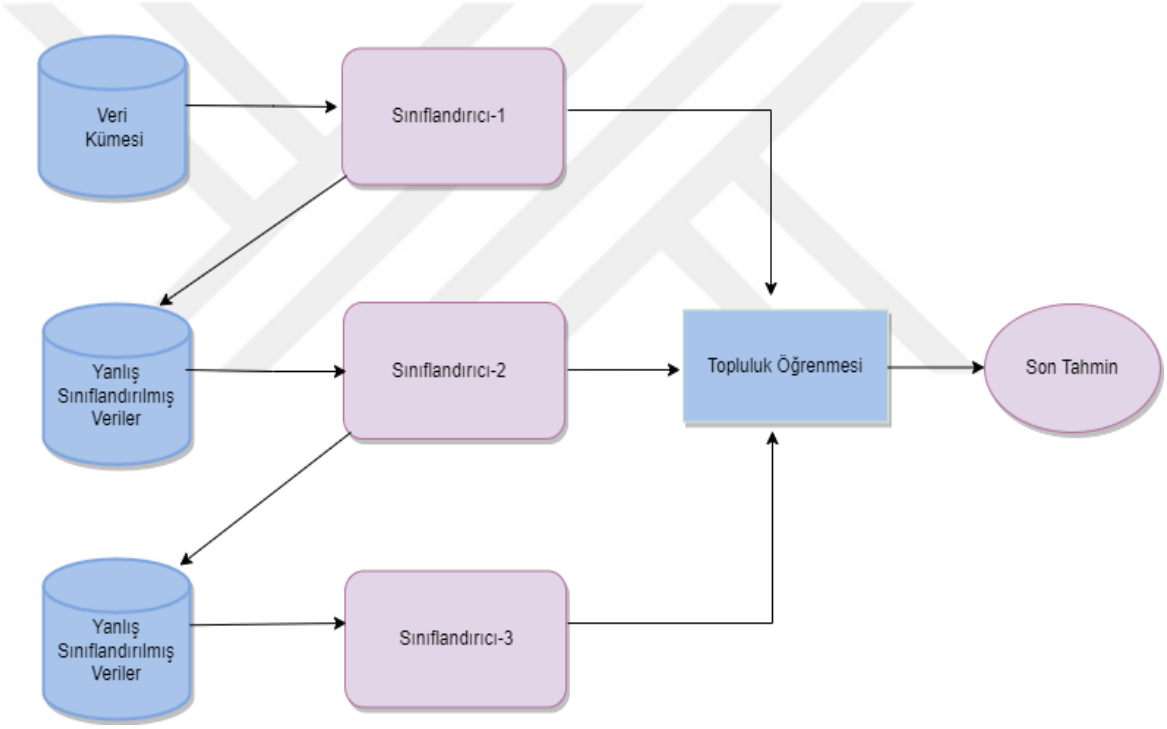
Burada, $h_i(x)$, i = 1, 2..., n birkaç zayıf öğrenen, n zayıf öğrenenlerin sayısı ve w_i, i = 1, 2, ..., n zayıfların ağırlıkları anlamına gelmektedir.

Boosting algoritmasının temelinde tek başına zayıf ve başarısız performansa sahip bir kestirim kuralının (örneğin küçük bir sınıflandırma ya da regresyon ağacının) birleştirilerek daha az hatalı kestirim kuralının elde edilmesi bulunmaktadır (Schapire ve Freund, 2012). Arttırma algoritmalarında ağaç sayısı, ağaçtaki mevcut düğüm sayısı ve öğrenme oranı olmak üzere üç parametreden oluşmaktadır.

Arttırma algoritmalarında çoğunlukla karar ağacı methodu kullanılmaktadır ve algoritma ağırlıklandırma yöntemi ile çalışmaktadır ve genel olarak ağaç tabanlı yöntemlerle birlikte kullanılmaktadır. Algoritma öncelikle veri setindeki eğitim kümesi

ile rastgele sınıflandırılır. Her bir sınıflandırıcının ağırlığı başlangıçta eşittir sınıflandırma işlemi yapılırken hata oranı fazla olan sınıflandırıcıların ağırlığı artar, kısmen daha az hata oranı olan sınıflandırıcıların ağırlıkları ise azalmaktadır. Tahminleme işlemleri sırayla yapılmaktadır. Sondaki tahminci ağaç kendinden önceki tahminci ağacın hatalı verilerini alır ve yeniden işler bunun amacı hatayı en aza indirmek ve güçlü bir öğrenici ortaya çıkartmaktır.

Boosting yöntemi, düşük doğruluk ve performansa sahip sınıflandırıcıları güçlendirerek yüksek doğruluk sağlayarak hataların en aza indirgenmesini sağlamaktadır. Arttırma algoritmasının örnek görseli Şekil 2.16' da verilmiştir.



Şekil 2.15: Arttırma Algoritması Çalışma Yapısı

Arttırma algoritması kullanımının avantajları ve dezavantajları aşağıdaki gibidir;

Avantaj:

- Eksik veri işleme kabiliyeti
- Yüksek performans
- Hata oranını azaltma
- Yüksek doğruluk
- Güçlü tahminleme

Dezavantaj:

- Karmaşık bir yapıya sahip olma
- Eğitim verilerinde gerçekleşen fazla uyum
- Ağaç sayısının, parametrelerin uygun ayarlanmaması durumunda aşırı öğrenme gerçekleşmesi
- İşlenmesi açısından yoğun ve karmaşıktır

Güçlendirme algoritmalarının birçok çeşidi bulunmaktadır. Güçlendirme algoritmaları; AdaBoost, Gradient Boosting (Gradyan Arttırma), CatBoost (Kategorik Güçlendirme), LightGBM (Hafif Gradyan Arttırma) ve XGBoost (Aşırı Gradyan Arttırma) modellerinden oluşmaktadır. Bu algoritmalar arasından en popüler olanı Schapire tarafından 1995 yılında geliştirilen AdaBoost (Uyarlanabilir Arttırma) algoritmasıdır. Bu tez çalışmasında güçlendirme algoritmaları arasından Kategorik Güçlendirme ve Hafif Gradyan Arttırma algoritmaları kullanılmıştır.

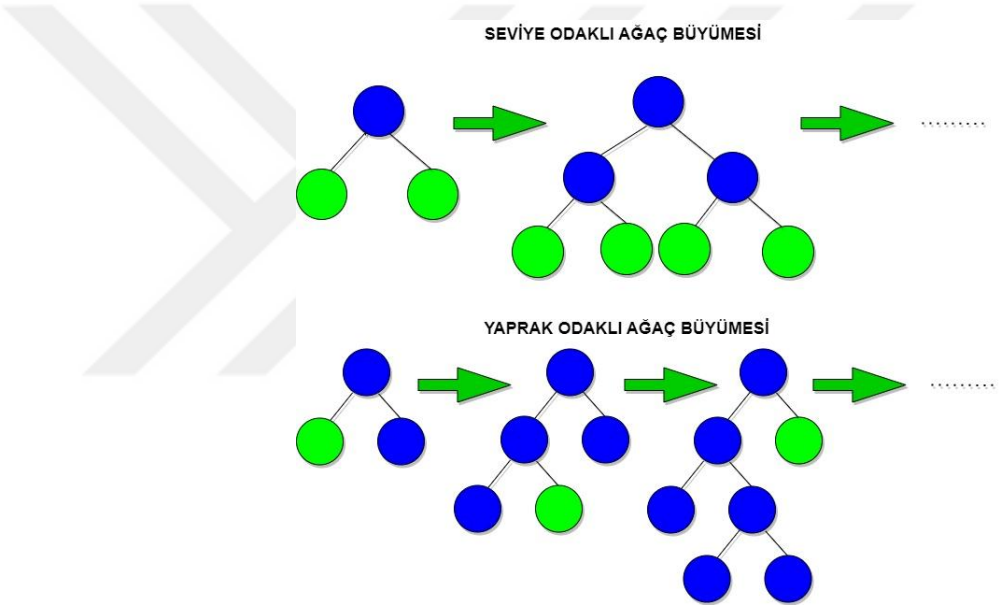
1.7. LightGBM

LightGBM (Hafif Gradyan Arttırma Makinesi / LGBM) algoritması, Microsoft tarafından 2017 yılında geliştirilen karar ağacı tabanlı gradyan arttırma yöntemi olan topluluk öğrenme algoritmalarındandır. Büyük verileri işleme, az RAM kullanımı, yüksek tahminleme oranı Hafif Gradyan Arttırma'nın avantajlı özelliklerindendir. Hafif Gradyan Arttırma, histogram tabanlı çalışan modelin verimliliğini arttırarak bellek kullanımını azaltmayı sağlayan bir algoritmadır bununla birlikte karar ağacındaki, ağaç derinliği, yaprak sayısı gibi özelliklerin optimizasyonunu yaparak aşırı öğrenmenin önüne geçmeyi sağlar. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree" makalesinde belirtildiği üzere, yapılan çalışmalarda LightGBM modelinin diğer modellere göre 20 kat daha hızlı olduğu sonucuna ulaşılmıştır (Ke vd., 2017). Hafif Gradyan Arttırma algoritması; veri seti içerisindeki kaybın en aza indirilmesini amaçlamaktadır. Bunun için kullanılan kayıp fonksiyonu aşağıdaki denklemde verilmektedir.

$$\mathbf{L}(\boldsymbol{\theta}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{l}\left(\boldsymbol{y}_{\mathrm{i}}, \boldsymbol{f}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{\theta})\right) + \boldsymbol{\Omega}(\boldsymbol{\theta}) \qquad \textbf{(2.18)}$$

Burada, L (θ), θ model parametrelerine göre en aza indirmek istediğimiz nesnel fonksiyondur. Objektif fonksiyon iki terimden oluşur: ilk terim, her eğitim örneği i için

öngörülen çıktı f (xi; θ) ile gerçek çıktı yi arasındaki farkı ölçen ampirik risktir. l (yi, f (xi; θ)) fonksiyonu, her eğitim numunesi için öngörülen ve gerçek çıktılar arasındaki farkı ölçen kayıp fonksiyonudur. İkinci terim olan Ω (θ), eklenen düzenlileştirme terimidir aşırı öğrenmeyi önlemek için objektif fonksiyona hafif gradyan arttırma kullanılacak düzenlileştirme türünü belirtmek için seçenekler sunar. 'Lambda' parametresi L2 düzenlileştirme gücünü kontrol ederken, 'alfa' parametresi L1 düzenlileştirme gücünü kontrol eder (Omotehinwa, Oyewola ve Dada, 2023). Karar ağaçları öğreniminde iki farklı yöntem kullanılır, bunlar; Seviye odaklı (Level-wise) ve Yaprak odaklı (Leaf-wise) yöntemlerdir.



Şekil 2.16 :Seviye Odaklı ve Yaprak Odaklı Ağaç Büyümesi Karşılaştırması

Veri analiz işlem sürecinde çoğu karar ağacı dengenin korunduğu seviye odaklı strateji ile bölünürken Hafif Gradyan Arttırma algoritmasında kaybı azaltma amacıyla yapılan yaprak odaklı bölünme işlemi gerçekleşir bu algoritmada karar ağacı delta değerinin en yüksek olduğu yaprağı seçerek dallanmaya devam etmektedir. Hafif Gradyan Arttırma, Aşırı Gradyan Arttırma algoritmasındaki eksiklerin tamamlanması amacıyla oluşturulmuş bir algoritmadır ayrıca Gradyan Arttırma (GBM) ve Aşırı Gradyan Arttırma (XGBoost) algoritmalarına göre daha başarılı tahminleme yapmaktadır.

1.7.1. Gradient Boosting (GBM) Algoritması

Gradyan arttırma algoritması ilk olarak Leo Breiman tarafından ortaya atılmış ardından 1999 yılında Friedman tarafından sınıflandırma ve regresyon için bir topluluk

yaklaşımı olarak tanıtılmıştır. Gradyan arttırma, zayıf öğrenicilerden daha güçlü öğreniciler elde etmeyi amaçlayan, her aşamada hatayı azaltmayı hedefleyen tahmine dayalı bir arttırma algoritmasıdır. Gradyan artırma fikri, Leo Breiman'ın artırmanın uygun bir maliyet fonksiyonu üzerinde bir optimizasyon algoritması olarak yorumlanabileceği gözleminden kaynaklanmıştır. (Breiman, 1997). Modelde tahminleme için karar ağaçları oluşturulmaktadır. Gradyan arttırma algoritması; karar ağacı ve arttırma (Boosting) yöntemlerinden oluşmaktadır. Gradyan destekli karar ağacı (GBDT) modelinde ise hataların en aza indirgenmesi amaçlanır, karar ağaçlarından oluşur ve arttırma (boosting) algoritmasının bir topluluk modeli olarak tanımlanabilir. Hafif gradyan arttırma algoritması; gradyan arttırma (GBM), GOSS ve EFB yöntemlerinden oluşmaktadır. Hafif gradyan arttırma algoritmasında performans arttıran iki teknik kullanmaktadır bunlar;

-Gradyan Tabanlı Tek Taraflı Örnekleme (Gradientbased One-Side Sampling/GOSS)

-Ayrıcalıklı Öznitelik Destekleme (Exclusive Feature Bundling/EFB)

1.7.1.1. -Gradyan Tabanlı Tek Taraflı Örnekleme (GOSS)

Hafif gradyan arttırma'nın performansının artmasında temel etkenlerden biri olan Goss algoritması, karar ağaçlarındaki daha az önemli olan verilerin sayısını bilgi kazanımı tahminin doğruluğunu bozmadan azaltmayı sağlar. Goss algoritması adından da anlaşılacağı üzere gradyana dayalı tek taraflı bir örnekleme yöntemidir ve örnekleme için yanlızca gerekli veriler kullanılır. Makine öğrenmesinde modelin öğrenimi için en fazla artış gerçekleşen yönü belirleyen vektöre 'gradyan' adı verilir. Burada Goss algoritması, mevcut verilerden bir karar ağacı oluşturur işlem sonucunda küçük gradyanlı örnekler az miktarda eğitim hatasına sahip örneklerdir, büyük gradyanlı örnekler ise eğitiminin yeterli olmadığı örneklerdir. Sonuç olarak diğerlerine kıyasla büyük gradyana sahip olan örnek seçilmektedir böylece veriler azaltılarak sonuca daha odaklanılmış olup daha verimli sonuçlar elde edilmektedir.

1.7.1.2. Ayrıcalıklı Öznitelik Destekleme (EFB)

Hafif gradyan arttırmanın performansını arttıran bir diğer yöntem ise EFB algoritmasıdır. Veri setinde bulunan değişkenlerden içerik bakımından az olanların

yapılarını bozmadan birleştirir bu yöntem sayesinde modeldeki işlem karmaşıklığının azalması sağlanarak daha verimli ve hızlı bir eğitim süreci oluşur. GOSS ve EFB algoritmaları gradyan destekli karar ağaçlarına göre daha iyi performans gösterir. Bu algoritmaların amacı ise, modelin daha hızlı ve verimli çalışmasını sağlayarak işlem karmaşıklığının azalmasını sağlamaktır.

Hafif gradyan arttırma algoritmasının avantajları;

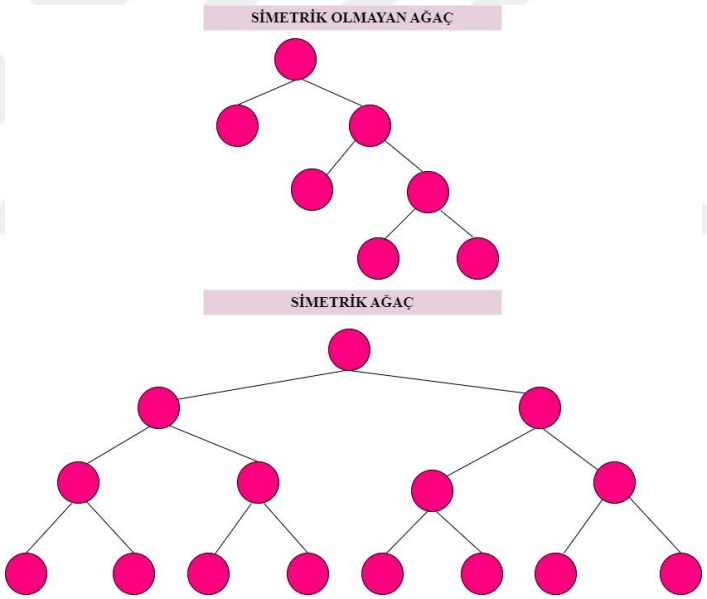
- Hızlı işlem gerçekleştirme
- Normale göre daha az bellek (RAM) kullanımı
- Yüksek doğruluğa sahip olma
- Büyük ölçekli verileri işleme
- Daha az hata oranına sahip olma

Hafif gradyan arttırma dezavantajı ise; küçük ölçekli verileri işlenmesi sonucu aşırı öğrenmeye yol açabilmesidir.

1.8. CatBoost

Xgboost (Aşırı Gradyan Arttırma) ve LightGbm (Hafif Gradyan Arttırma)'den sonra 2017 yılında Catboost algoritması oluşturulmuştur. Catboost algoritması, Yandex mühendisleri tarafından geliştirilmiş Xgboost ve Lightgbm temelli, açık kaynak kodlu bir gradyan arttırma karar ağacı (GBDT) algoritmasıdır. Catboost algoritması adını "Category" ve "Boosting" kelimelerinin kısaltılmasından almıştır ve "Kategorik Güçlendirme" anlamını taşımaktadır. Diğer makine öğrenme algoritmaları kategorik değişkenleri direkt işleyemez bunun için değişkenleri sayısal değerlere dönüştürülmesi gereklidir ancak kategorik güçlendirme algoritmasında veri kümesindeki kategorik öznitelikleri ayrıca herhangi bir kodlama tekniği (Tek sıcak kodlama, Etiket kodlama) kullanmadan ağaç bölümünden önce otomatik olarak sayısal özniteliklere dönüştürmeyi sağlar böylece veri hazırlığı aşamasında gecen süreyi kısaltır ayrıca regresyon ve sınıflandırma problemlerinde kullanılabilmektedir. Simetrik karar ağaçlarından oluşan Catboost algoritması modeldeki aşırı uyum sorununu çözmeyi sağlar ve küçük veri kümelerinde de yüksek hız ve doğrulukla çalışmaktadır.

Catboost algoritması, Xgboost ve Lightgbm algoritmalarına göre daha yüksek performans göstermektedir. Kategorik, sayısal ve boş veriler ile kolayca işlem yapması, yüksek tahminleme özelliği, aşırı öğrenmenin önüne geçmesi avantajlı özelliklerinden birkaçıdır. Catboost, veri seti içerisindeki eksik değerleri işlemesi için SWQS algoritmasını kullanarak aşırı öğrenmeyi azaltır bununla beraber veri setinin performansının artmasını sağlamaktadır. Görüntü, metin, ses içeren kategorik verilerle rahatlıkla çalışabilmektedir. Catboost tahmin kayması durumunun önüne geçmek için Sıralı Arttırma (Ordered Boosting) yöntemini kullanmaktadır. Sıralı Arttırma yöntemi, hedef değişkenin tahminlenmesinde oluşabilecek hataları ve veri kayıplarının önlenmesini sağlamaktadır.



Şekil 2.17 :Simetrik Ağaç ile Simetrik Olmayan Ağaçların Karşılaştırması

Catboost, ilk olarak veri setindeki eğitim verilerinin rastgele permütasyonunu oluşturur, Rastgele bir permütasyonu örnekleyerek ve temelinde gradyanlar elde ederek algoritmanın sağlamlığını artırmak için çoklu permütasyonlar kullanılır (Huang vd.., 2019). Modelin eğitim sürecinde performansı yükseltmek için CPU ve GPU uygulamaları kullanılmaktadır GPU uygulaması modelin hızlı bir şekilde eğitilmesini sağlar böylece eğitim süresini kısaltır. Catboost çalışmadaki hatayı azaltmak için her seferinde ikili karar ağacı oluşturur. CatBoost, uygun grafik kartı kullanıldığı sürece GPU

üzerinde çalışır ve bilgisayarın ani kapanması gibi beklenmeyen durumlarda hafıza kopyaları alarak öğrenme süresi uzun olan problemlerde avantaj sağlar (Hancock ve Khoshgoftaar, 2020). Catboost modelinin denklemi aşağıda verilmiştir;

$$\mathbf{Z = H(x_i) = \Sigma J_J = 1c_J 1\{x \epsilon R_J\}} \quad \textbf{(2.19)}$$

Burada, $H(x_i)$ karar ağacının açıklayıcı değişkeni iken x_i ve Rj ağaç yapraklarına karşılık gelen ayrık bölgeleri temsil etmektedir (Tekin ve Sarı, 2022). Catboost, kategorik değişken işleme, gradyan arttırma tekniklerinin yanı sıra sıralı yükseltme yöntemini kullanmaktadır. Sıralı yükseltme yöntemi, modeldeki aşırı öğrenmeyi azaltırken hedef sızıntı sorununu çözmeyi, tahminlemede güçlendirmeyi ve tahmin kaymasını önlemeyi sağlamaktadır. Tahmini kaydırmayı genel olarak hedef değişkenin tahminlenmesinde doğruluktan uzaklaşılması olarak ifade edebiliriz. Gradyan güçlendirme tekniğinin kullanılması veri kümesine karşı aşırı uyum (overfitting) göstermesi problemine neden olabilmektedir. Bu sorun CatBoost algoritması tarafından ele alınmış ve aşırı uyum gösterme probleminin önüne geçilmesi amacıyla gradyan güçlendirme tekniğinin geliştirilmiş bir versiyonu kullanılmıştır (Dorogush, Ershov ve Gulin, 2018).

Kategorik güçlendirme algoritması veri setinde bulunan eksik veriler için minimum ve/veya maksimum olmak üzere farklı değerler vererek XGB ve LGBM algoritmalarına göre farklılık göstermektedir. Kategorik güçlendirme algoritmasının avantajları aşağıdaki gibidir;

- Parametreye ihtiyaç duymadan kategorik verileri işleme
- Aşırı öğrenmeyi azaltma
- Veri kümesindeki eksik değerleri işleme
- Sayısal, kategorik ve metinsel verilerle çalışabilme
- Hızlı tahminleme ve yüksek performans
- Yüksek doğruluk

CatBoost'taki farklı parametrelerin modelin tahmin sonuçları üzerinde belirli bir etkisi vardır, bu nedenle daha iyi tahmin sonuçları elde etmek için parametreleri optimize edebiliriz. Parametrelerin seçimi, daha hızlı eğitim hızı, daha yüksek tahmin doğruluğu ve aşırı uyumun önlenmesi olmak üzere üç ilkeye dayanmaktadır (Wei vd., 2023).

Catboost algoritması diğer makine öğrenmesi algoritmalarına göre çok fazla parametre ayarına ihtiyaç duymamakla birlikte aşırı öğrenmeyi engellemek amacıyla;

max ağaç uzunluğu, oluşturulacak ağaç sayısı, öğrenme oranı ve kategorik özellikler parametrelerini kullanmaktadır.

Karar ağacı bölümlemelerini yeni bir teknikle yapmakta olan Catboost, bu tekniğe en az varyans örneklemesi, “Minimal Variance Sampling” (MVS) adını vermiştir (Ibragimov ve Gusev, 2019: 1).

ÜÇÜNCÜ BÖLÜM

BANKA MÜŞTERİ KAYBI ANALİZİ İLE İLGİLİ UYGULAMA

Bu bölümde karışıklık matrisi, uygulama çalışması, veri setinde bulunan değişkenler arası ilişkiler, veri önişleme parametreleri, korelasyon analizi ve çalışmada kullanılan modellerin performanslarına değinilmiştir.

1. VERİ MADENCİLİĞİ UYGULAMASI

1.1. Karışıklık Matrisi (Confisuon Matrix)

Hata matrisi olarak da bilinen karışıklık matrisi, makine öğrenmesi yöntemlerinde kullanılan sınıflandırma algoritmalarında her sınıfın ayrı ayrı başarı hesaplanmasını, doğru ve yanlış tahmin oranlarını gösteren tablodur. Karışıklık matrisinin bir ekseni modelin gerçek etiketi diğer ekseni ise tahminlenen etiketten oluşmaktadır.

GERÇEK

TAHMİN		Pozitif	Negatif
	Pozitif	Doğru Pozitif (TP)	Yanlış Pozitif(FP)
	Negatif	Yanlış Negatif (FN)	Doğru Negatif(TN)

Şekil 3. 1:Karışıklık Matrisi ve Açıklaması

TP (Doğru Pozitif); Gerçekte doğru olan bir değerin tahminleme sonrası sonucun pozitif olduğunu gösterir.

FP (Yanlış Pozitif); Gerçekte yanlış olan bir değerin tahminleme sonrası sonucun pozitif olduğunu gösterir.

FN (Yanlış Negatif); Gerçekte yanlış olan bir değerin tahminlenme sonrası sonucun negatif olduğunu gösterir.

TN (Doğru Negatif); Gerçekte doğru olan bir değerin tahminleme sonrası sonucun negatif olduğunu gösterir.

1.1.1. Tahmin Hatası

Karışıklık matrisinde kullanılan veriler arasından hatalı tahmin edilen verileri gösterir. Matrisin performansını ölçmek için kullanılmaktadır.

$$\frac{\mathbf{FP+FN}}{\mathbf{TP+TN+FP+FN}} \quad \mathbf{(3.1)}$$

1.1.2. Doğruluk Oranı (Accuracy)

Veri seti üzerinden sınıflandırılmış veri örneklerini temsil eden değerdir. Doğruluk oranı ile sınıflandırıcının tahminleme sıklığı ölçülür.

$$\frac{\mathbf{TN+TP}}{\mathbf{TN+FP+TP+FN}} \quad \mathbf{(3.2)}$$

1.1.3. Kesinlik (Precision)

Yapılan tahminlerin kaç tanesinin doğru sonuç verdiğini gösteren ölçüdür. Bir sınıflandırıcının iyi olması için kesinlik değeri en ideal 1 olmalıdır ve kesinlik yalnızca pay ve payda eşit olduğunda 1 olmaktadır. Bu durum ise FP'nin sıfır olması gerektiğini gösterir.

$$\frac{\mathbf{TP}}{\mathbf{TP+FP}} \quad \mathbf{(3.3)}$$

1.1.4. Duyarlılık (Recall)

Sınıflandırıcının verilerdeki tüm pozitif sınıf örnekleri üzerinden kaç tane pozitif sınıfı doğru tahmin ettiğini gösteren bir ölçüdür.

$$\frac{\mathbf{TP}}{\mathbf{TP+FN}} \quad \mathbf{(3.4)}$$

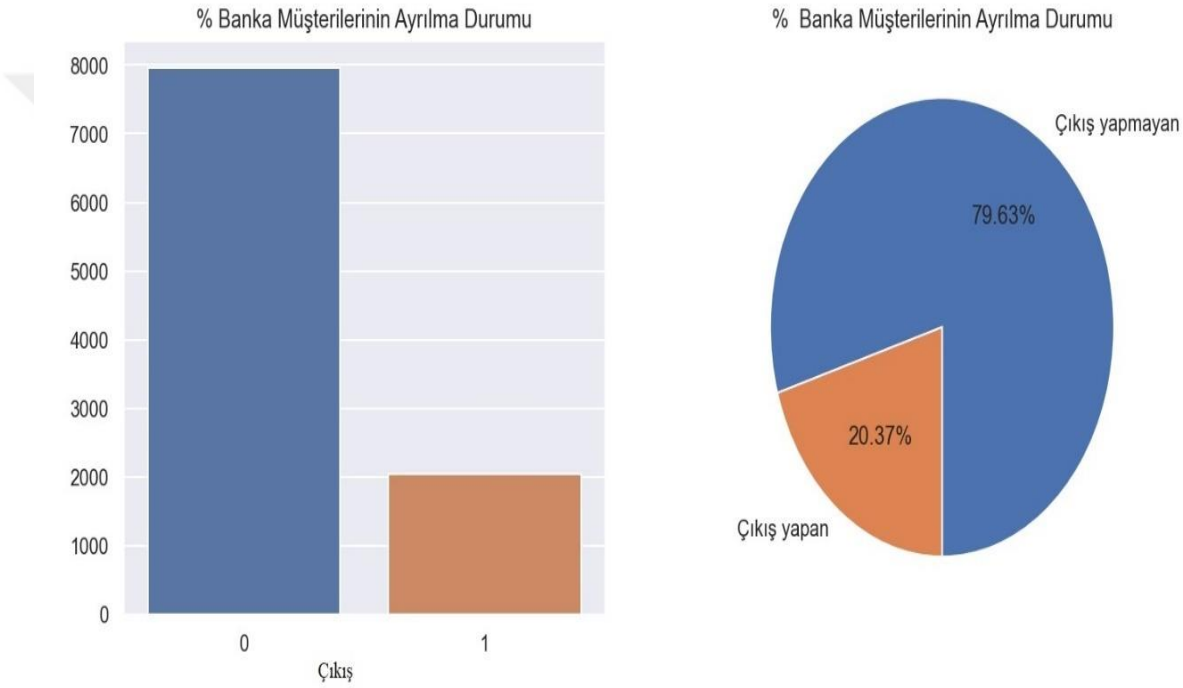
1.1.5. F1 Skor

Modelin doğruluğunu ölçen makine öğrenimi değerlendirme metriğidir. Genellikle Kesinlik (Precision) ve Duyarlılık (Recall) değerlerini birleştirerek bu değerlerin harmonik ortalamasının alınmasıyla oluşan ölçüdür.

$$\mathbf{2} * \frac{\mathbf{Kesinlik} * \mathbf{Duyarlılık}}{\mathbf{Kesinlik+Duyarlılık}} \quad \mathbf{(3.5)}$$

1.2. Uygulama

Bu tez çalışmasında 3 ülkenin banka müşterilerine ait bilgiler makine öğrenmesi yöntemleri ile analiz edilerek banka müşterilerinin bankayı terk edip etmeme olasılığı üzerine çalışılmıştır. Çalışmadaki modellerin eğitilmesinde Spyder programı ve Python programlama dili kullanılmıştır. Veri setinde 10.000 adet veri içermekte olup 14 adet değişken bulunmaktadır. Şekil 3.2' de Banka müşteri kaybı veri setinde bulundan mevcut müşterilerin bankadan ayrılma oranları gösterilmektedir.



Şekil 3. 2: Banka Müşterilerinin Bankadan Ayrılma Oranı

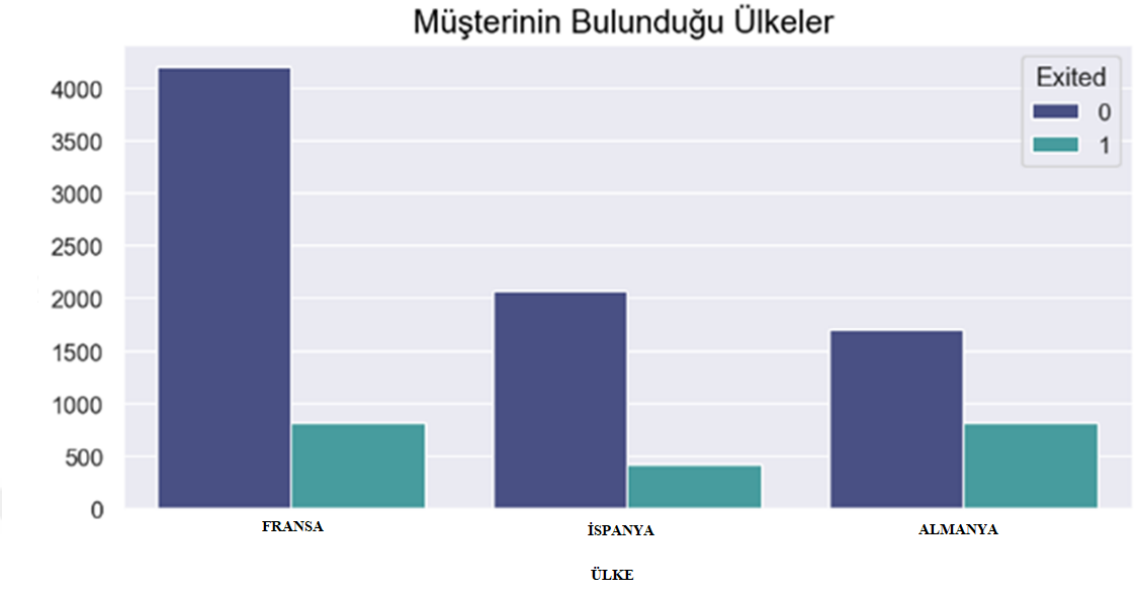
Bankadan ayrılan müşteri oranı %20,37 olup bankada müşteri olarak kalmaya devam eden kişi oranı ise %79,6 olarak hesaplanmıştır. Bankadan çıkış yapmayan müşteri sayısı 7963 kişi, çıkış yapan müşteri sayısı ise 2037 olarak bulunmuştur. Tez çalışmasının gerçekleştirildiği veri seti değişkenleri ve açıklamaları Tablo 3.2 'de verilmektedir. Tablo 3.3'te ise veri setindeki değişkenlere ait sayısal veriler yer almaktadır.

Tablo 3.2: Veri Seti Değişkenleri ve Açıklaması

Değişkenler	Açıklama
RowNumber	Satır numaraları 1'den 10.000'e kadardır.
CustomerId	Müşteri numarası
Surname	Müşterinin soyadı
Credit score	Müşterinin kredi notu
Geography	Müşterinin ülkesi
Gender	Müşterinin cinsiyeti
Age	Müşterinin yaşı
Tenure	Müşterinin bankada bulunduğu süre
Balance	Müşterini bankadaki bakiyesi
NumOfProducts	Müşterinin kullandığı banka ürünlerinin sayısı (mobil bankacılık, internet bankacılığı vb.)
HasCrCard	Müşterinin kredi kartı olup olmadığı
IsActiveMember	Müşterinin bankadaki aktiflik durumu
EstimatedSalary	Müşterinin tahminlenen maaşı
Exited	Müşterinin bankadan ayrılıp ayrılmama durumu (1: evet / 0: hayır)

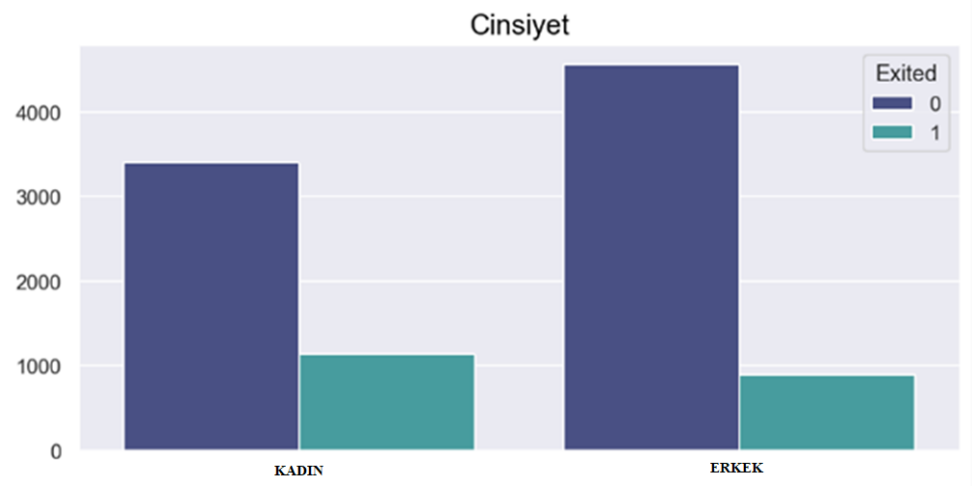
Tablo 3.3: DataFrame'de Bulunan Sayısal Verilere Ait Açıklama Tablosu

	Kredi Notu	Yaş	Süre	Bakiye	Ürün sayısı	Kredi kartı	Aktiflik durumu	Tahmini maaş	Çıkış
Sayım	10000	10000	10000	10000	10000	10000	10000	10000	10000
Maks	850	92	10	25098	4	1	1	199992	1
75%	718	44	7	127644	2	1	1	149388	0
50%	652	37	5	97198.5	1	1	1	100194	0
Ort	650.529	38.9218	5.0128	76485.9	1.5302	0.7055	0.5151	100090	0.2037
25%	584	32	3	0	1	0	0	51002.1	0
Min	350	18	0	0	1	0	0	11.58	0
Std	96.6533	10.4878	2.89217	62397.4	0.581654	0.45584	0.499797	57510.5	0.70279



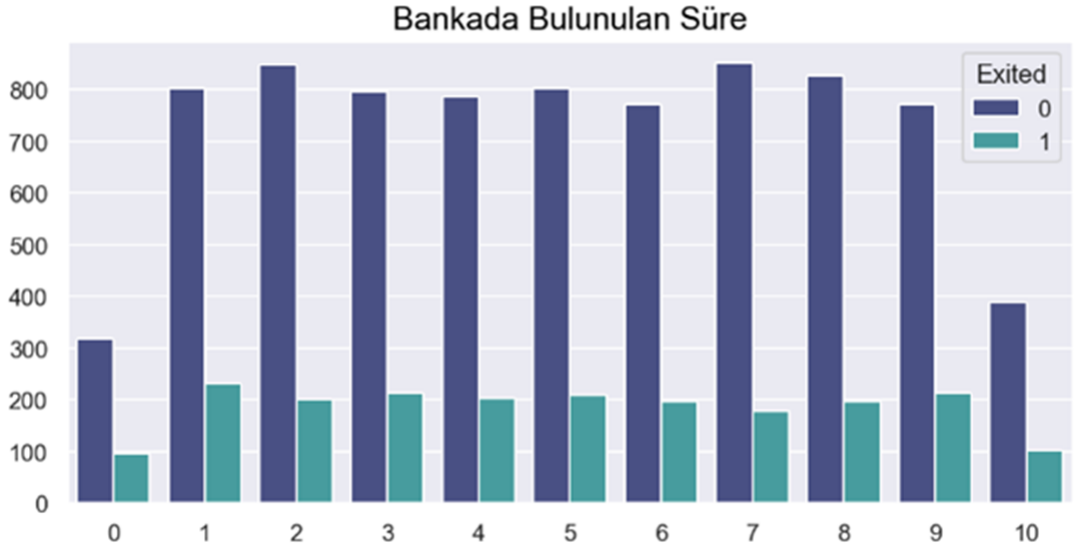
Şekil 3. 3: Müşterinin Bankadan Ayrılması ile Bulundukları Ülkeler Arasındaki İlişki Grafiği

Çalışmada, Şekil 3.3'te verilen grafiklerden ilk olarak müşterilerin ülkeler arasındaki dağılımı görülmektedir. Banka veri seti içerisinde en fazla müşterinin bulunduğu ülke Fransa ikinci Olarak İspanya ve en az müşterinin bulunduğu ülke ise Almanya'dır. Bankayı terk eden müşterilerin ülkelerine bakıldığında Almanya ve Fransa aynı oranda müşteri kaybederken ispanyada bankayı terk eden müşteri sayısı en az olarak görülmekle birlikte Fransa, müşterilerin bankada kalma oranı en yüksek olan ülke olarak görülmektedir.



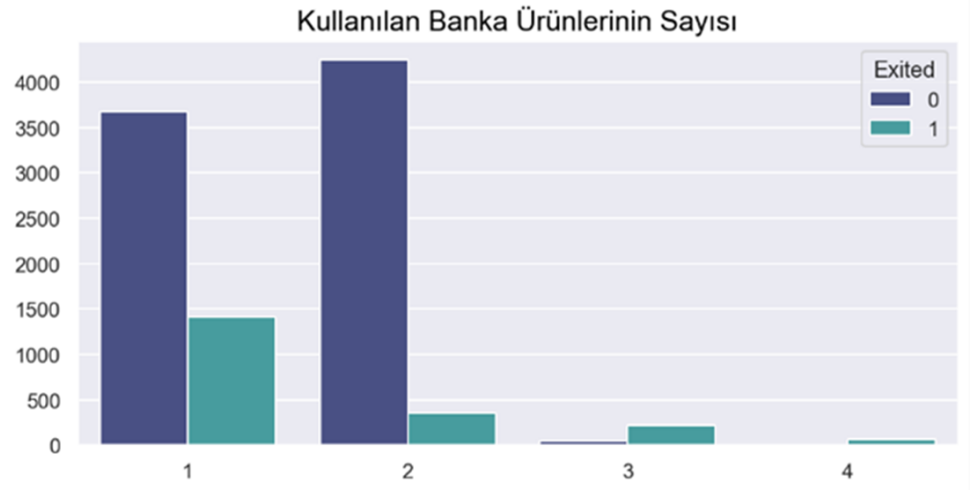
Şekil 3. 4 : Bankada Kayıtlı Müşterilerin Cinsiyet Dağılım Grafiği

Bankadaki müşterilerin cinsiyet dağılımlarına bakıldığında erkek müşteriler kadın müşterilere oranla daha fazla olduğu, müşterilerin cinsiyetlerine göre bankayı terk etme durumlarına bakıldığında ise kadın müşterilerin erkek müşterilere göre bankayı terk etme oranı daha yüksek olduğu görülmektedir.



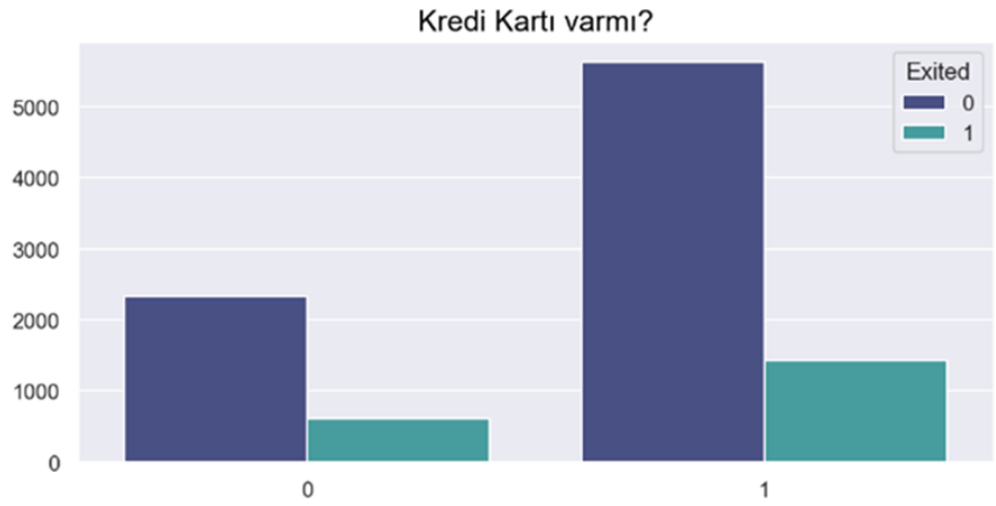
Şekil 3. 5: Müşterilerin Bankada Bulunduğu Süre

Şekil 3.5'te müşterinin bankada bulunduğu süre/yıl ile bankayı terk etme ilişkisine bakıldığında müşterinin bankaya girişinin ilk ayları ve 10. yılında bankayı terk etme oranı diğer yıllara göre daha düşük olduğu görülmektedir.



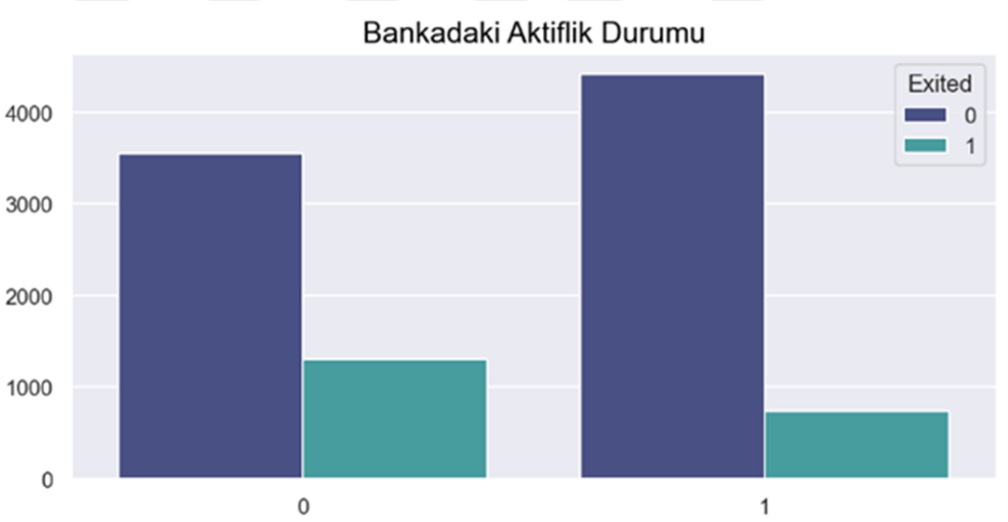
Şekil 3. 6: Müşterilerin Banka Aracılığıyla Aldıkları Ürün Sayısı Grafiği

Şekil 3.6'daki grafikte müşterinin banka aracılığıyla aldığı ürün sayısı arttıkça bankayı terk etme oranı gözle görülür bir şekilde azaldığı görülmektedir.



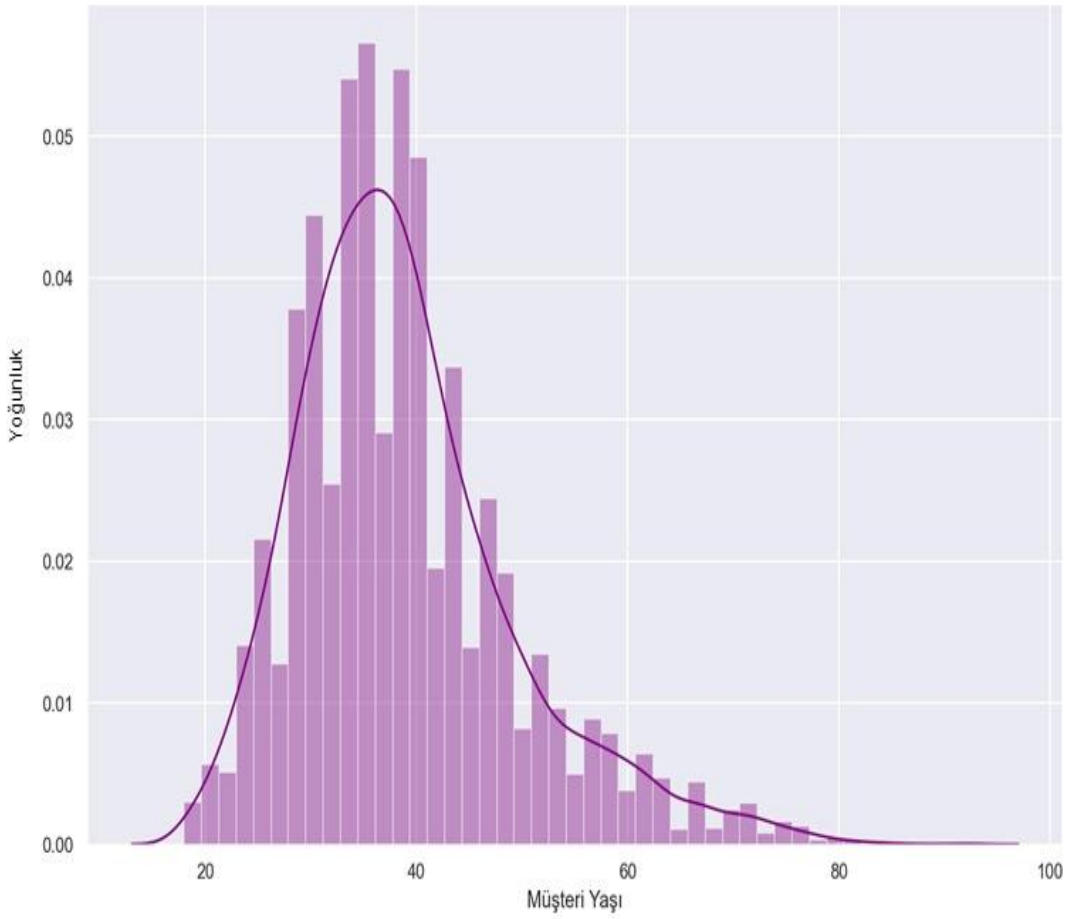
Şekil 3. 7: Müşterilerin Kredi Kartı Olup Olmaması ile Bankadan Ayrılma İlişkisi

Şekil 3.7'de banka müşterilerinin verilerine göre kredi kartı olan ve olmayan müşterilerin her ikisinde de ayrılma gerçekleşse de bankaya kayıtlı kartı olan müşterilerin daha fazla olduğu ve bankadan ayrılma oranının daha yüksek olduğu görülmektedir.



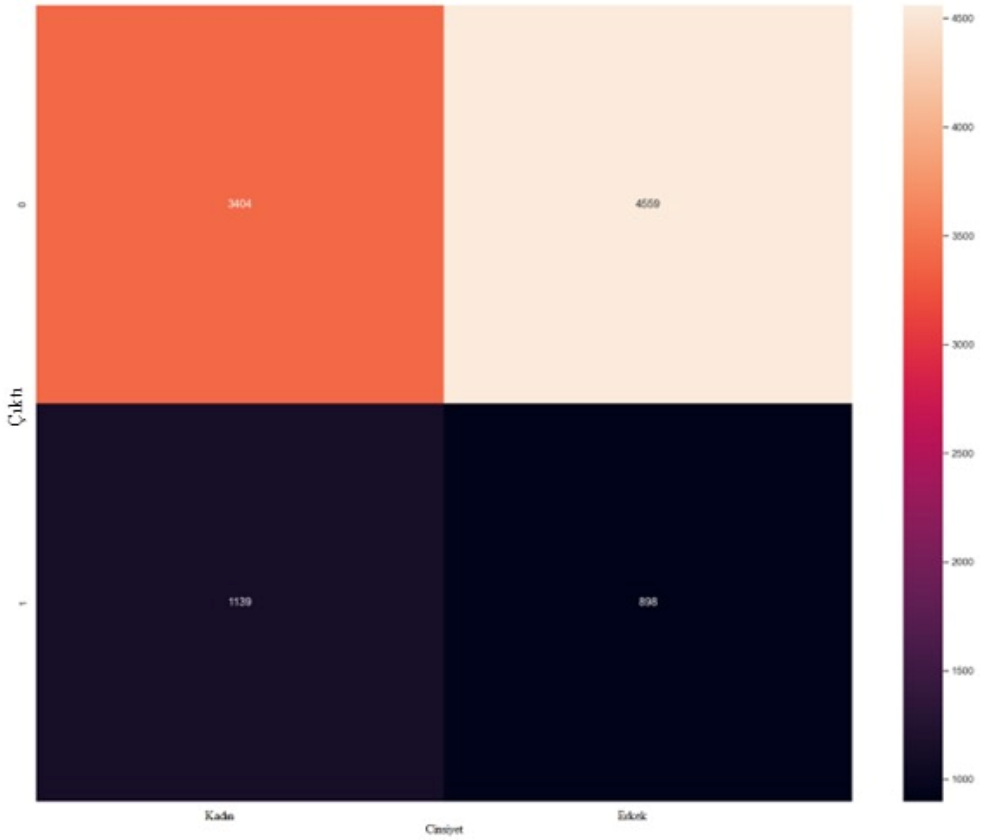
Şekil 3. 8: Müşterilerin Bankadaki Aktiflik Durumu ile Bankadan Ayrılma Durumu Arasındaki İlişki Grafiği

Şekil 3.8'de bankada aktif olmayan müşterilerin bankayı terk etme oranı daha yüksek olduğu bununla birlikte banka hesabını aktif bir şekilde kullanan müşterilerin çoğunluğunun bankayı terk etmediği görülmektedir.



Şekil 3. 9: Banka Müşterilerinin Yaşı ile Bankayı Terk Etme Durumları Arasındaki İlişki

Yukarıdaki grafikte banka müşterilerinin yaş verilerinin yoğunluğu histogram grafiği ile görselleştirilmiştir. Grafikte görüldüğü üzere veri setinde bulunan müşterilerin yaşlarına bakıldığında en fazla yoğunluk 30 ila 50 yaş aralığındaki müşteriler arasında bulunmaktadır. Şekil 3.10'da ısı haritası kullanılarak banka müşterilerinin cinsiyetleri ile bankayı terk etme durumu arasındaki ilişki gösterilmektedir.



Şekil 3. 10: Müşterilerinin Cinsiyetlere Göre Bankayı Terk Etmesi Arasındaki İlişkinin Isı Grafiği ile Gösterimi

Yukarıda verilen ısı haritası görselinde banka müşterileri arasında erkek müşterilerin bankayı terk etme oranının en düşük olduğu, kadın müşterilerin erkek müşterilere oranla bankayı terk etme oranının daha yüksek olduğu gözlemlenmiştir. Veri madenciliği uygulaması 6 adımdan oluşmaktadır bunlar;

1. Veri Hazırlama
2. Veri temizliği/veri önişleme
3. Veri dönüştürme
4. Özellik ölçeklendirme
5. Veri madenciliği algoritmalarının uygulanması
6. Sonuçların değerlendirmesi

1.2.1. Veri Hazırlığı

Bu tez çalışmasında veri analizine hazırlanmak için öncelikle seaborn, numpy, pandas matplotlib kütüphaneleri yüklenmiştir. Kategorik değişkenlerin işlenmesi için

etiket kodlama (Label encoding), kategorik değişkenlerin ikili vektör olarak işlenmesi için ise one hot encoding (Tek sıcak kodlama) işlemleri yüklenmiştir. Veri hazırlığında kullanılan bir diğer parametre ise standard scalerdir. Dizi ya da matrisleri rastgele test ve eğitim seti olarak ayırması için train-test split parametresi yüklenmiştir.

Veri biliminde oldukça sık kullanılan yöntemlerden biri olan Smote aşırı öğrenme sorununu çözmek için veri hazırlığı aşamasında kurulumu gerçekleştirilmiştir. Son olarak f1 skor, kesinlik, duyarlılık ölçüm araçlarının kurulumu yapılmıştır.

1.2.2. Veri Önişleme (Data Pre-processing)

Veri önişleme; veri madenciliğindeki önemli adımlardan biridir. Veri ön işlemenin amacı verilerin kalitesini yükseltmek, veri analizinde performansı arttırmak ve veri madenciliği uygulaması için uygun duruma getirmektir. Sınıflandırma algoritmalarının uygulanmasında, kullanılacak veri setinin yapısı oldukça önemli olmaktadır. Bu sebeple, söz konusu algoritmaların uygulanmasına geçilmeden önce genellikle veri seti üzerinde bazı ön işlemler gerçekleştirilmekte olup, söz konusu ön işlemler ile veri setinin hazırlanması makine öğrenmesi sürecinde çok fazla zaman gerektiren aşamalardan biridir (Miksovsky vd., 2002; Görmez, 2021:30). Veri önişleme, verilerin düzenlenip dönüştürülmesini ve veri analizi için uygun bir duruma getirilmesi için yapılan ön hazırlık işlemidir. Veri ön işleme adımlarından bazıları aşağıdaki gibidir;

- Veri temizleme
- Veri dönüştürme
- Veri standardizasyon ve normalizasyonu
- Veri entegrasyonu
- Veri azaltma

Veri ön işleme temel olarak veri temizleme, veri entegrasyonu, dönüştürme ve azaltma gibi çeşitli teknikleri içerisinde barındırmaktadır (Alasadi, Bhaya, 2017). Bu tez çalışmasında veri analizine katkıda bulunmayacak olan “Satır Numarası”, “Müşteri Numarası”, “Soyad” değişkenleri veri setinden çıkarılmıştır. Veri setindeki “Cinsiyet”, “Ülke” değişkenlerine tek sıcak kodlama tekniği uygulanmıştır. Çalışmada kullanılan veri setine ait veri önişleme çalışması yapılmış olup gereksiz sütunlar silinmiştir. Cinsiyet ve

ülke isimleri için sıcak kodlama tekniği kullanılmıştır. Yapılan önişleme sonrası veri setine ait ilk 14 veri Şekil 3.11'de verilmiştir.

Index	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Male	Germany	Spain
0	619	42	2	0	1	1	1	101349	1	0	0	0
1	608	41	1	83807.9	1	0	1	112543	0	0	0	1
2	502	42	8	159661	3	1	0	113932	1	0	0	0
3	699	39	1	0	2	0	0	93826.6	0	0	0	0
4	850	43	2	125511	1	1	1	79084.1	0	0	0	1
5	645	44	8	113756	2	1	0	149757	1	1	0	1
6	822	50	7	0	2	1	1	10062.8	0	1	0	0
7	376	29	4	115047	4	1	0	119347	1	0	1	0
8	501	44	4	142051	2	0	1	74940.5	0	1	0	0
9	684	27	2	134604	1	1	1	71725.7	0	1	0	0
10	528	31	6	102017	2	0	0	80181.1	0	1	0	0
11	497	24	3	0	2	1	0	76390	0	1	0	1
12	476	34	10	0	2	1	0	26261	0	0	0	0
13	549	25	5	0	2	0	0	190858	0	0	0	0
14	635	35	7	0	2	1	1	65951.6	0	0	0	1

Şekil 3. 11: Veri Setinde Veri Önişleme Sonrası İlk 14 Verinin Görüntüsü

1.2.3. Veri Önişleme İçin Kullanılan Bazı Parametreler

Bu bölümde veri analizi için gerekli aşamalardan biri olan veri önişleme işleminde kullanılan bazı parametrelerden bahsedilmiştir.

1.2.3.1. Eğitim- Test Kümeleme (Train- Test Split)

Eğitim ve test bölümünde, veri setinde bulunun veriler eğitim kümesinde modelin eğitimini gerçekleştirirken test kümesinde ise doğruluk ve performans ölçümü yapılmasını sağlar. Scikit Learn kütüphanesinde kullanılan Train_test_split metodu, veri setinde tahminleme oluşturmak için hedef değişken ile bağımsız değişkenlerin bulunduğu verileri rastgele karıştırarak eğitim ve test kümelerini oluşturur.

```
X = df.drop(columns=['Exited'])
y = df['Exited']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)
```

Şekil 3. 12: Veri Setindeki Verilerin Gruplara Ayrılması

Eğitim ve test kümeleme işleminde gerekli kütüphanelerin yüklenmesi ve veri setinin yüklenmesinin ardından öncelikle veri setindeki veriler x ve y adlarıyla gruplara

ayrılır burada x kümesinde bağımsız değişkenler, y kümesine ise hedef değişken olacak şekilde düzenlenir.

Veriler, X_train, X_test, y_train, y_test işlemi ile gruplara ayrılır. Veri setindeki test kümesi için ayrılması gereken oran ise test_size işlemi ile hesaplanmaktadır. Bu çalışmada test_size: 0,3 olarak seçilmiş olup veri setindeki verilerin %30'u test için ayrılmıştır.

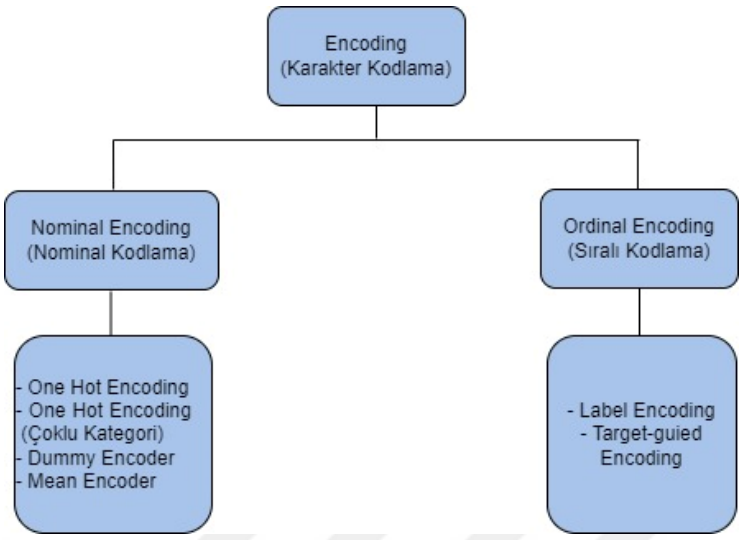
1.2.3.2. Numaralandırma (Enumerate)

Python'un işlevlerinden biri olan Enumerate methodu, veri setinde liste içerisinde bulunan elemanları numaralandırmayı sağlar. Enumerate, nesnelere ayrı ayrı index numarası atar, herhangi bir numaralandırma parametresi kullanılmaması durumunda index numarası 0'dan başlamaktadır. Numaralandırma methodu, tekrarlamaların/ döngülerin sayısını hesaplamak için kullanılmaktadır. Veri setinde bulunan nesnelerin index numaralarıyla beraber listelenerek tekrar eden nesnelerin takibini kolaylaştırmaktadır.

1.2.3.3. Karakter kodlama (Encoding)

Makine öğrenmesi ve veri analizi modelleme işlemlerinde değişken türleri oldukça önemlidir çoğu makine öğrenme sürecinde sayısal değişkenler ile işlem yapılmaktadır bu yüzden çalışılacak olan veri setindeki değişkenler kategorik ise sayısala dönüştürülmelidir. Karakter kodlama (Encoding), veri önişleme sürecinin önemli bir parçasıdır. Karakter kodlama, nominal kategorik değişken ve sıralı kategorik değişken olarak ikiye ayrılmaktadır. Kategorik değişkenlerin makine diline çevrilmesi için kullanılan teknikler;

- One Hot Encoding (Tek sıcak kodlama)
- Dummy Encoding (Kukla kodlama)
- Ordinal Encoding (Sıralı kodlama)
- Binary Encoding (İkili karakter kodlama)
- Target Encoding (Hedef kodlama)
- Mean Encoding (Ortalama kodlama)

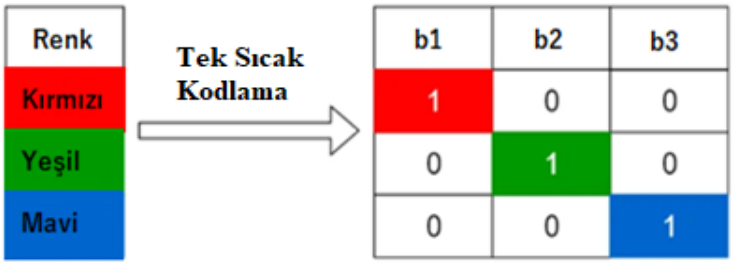


Şekil 3. 13: Karakter Kodlama (Encoding)

Encoding yöntemleri ile veri analizi sürecinde kolaylık sağlayarak daha güvenilir sonuçların alınmasını sağlar.

1.2.3.4. Tek Sıcak Kodlama (One Hot Encoding)

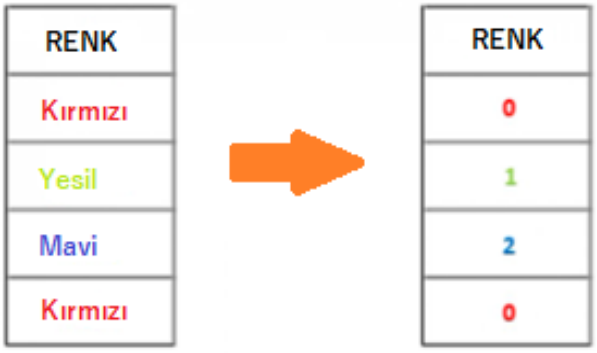
Python'un işlevlerinden bir diğeri ise 'Tek Sıcak Kodlama' methodudur. Tek Sıcak Kodlama, kategorik değişkenlerin sayısal değerlere dönüştürülmesinde kullanılmaktadır. Sayısal değişkenlere dönüştürülen tamsayı değerleri ise ikili vektör olarak göstermektedir. Listede bulunan her bir değişken için yeni bir sütun oluşturur ve kodlama işlemini gerçekleştirir. Tek Sıcak Kodlama methodunun etiket kodlama methodundan farkı; Etiket kodlama, kategorik değişkenlere direkt sayı ataması yaparken tek sıcak kodlama ise değişkenleri ayrı ayrı ele alarak ayrıntılı bir dönüştürme işlemi sağlar.



Şekil 3. 14 Tek Sıcak Kodlama

1.2.3.5. Etiket kodlama (Label Encoding)

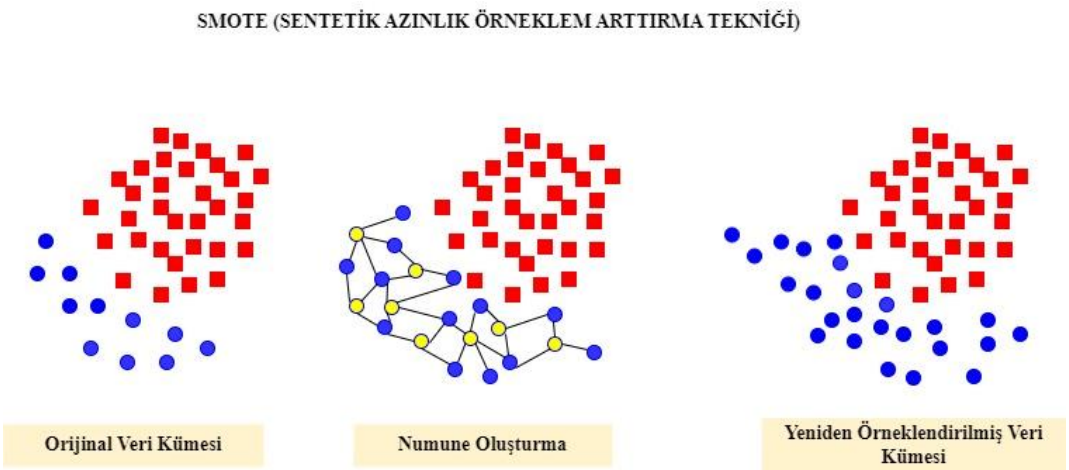
Etiket kodlama; veri setindeki kategorik değişkenleri sayısal değerlere dönüştürmek için kullanılan kodlama yöntemidir. Bu yöntem ile veri setinde bulunan her bir kategorik değişken için ayrı bir sayısal değer atama işlemi gerçekleştirir.



Şekil 3. 15: Etiket Kodlama

1.2.4. Smote

Sentetik azınlık yüksek örnekleme tekniği anlamına gelen Smote, makine öğrenmesi ve veri bilimi alanında sıklıkla kullanılan bu teknik, veri setinin çalışıldığı modelde oluşabilecek aşırı öğrenmeyi engelleyerek dengesiz sınıf dağılımının önüne geçmektedir. Smote, sınıflandırma aşamasında veri setindeki sayıca az olan sınıf için sentetik veriler üretir ve sınıfların dengelenmesini sağlayarak doğruluğu arttırmayı sağlamaktadır.



Şekil 3. 16: Sentetik Azınlık Örneklem Arttırma Tekniği

1.2.5. Constant

Constans, programlama süresince değiştirilemeyen değerler için kullanılan sabit bir değişken türüdür. Constans değişkeni oluşturulurken rakamla başlamamalı ve sözel karakter kullanmamaya özen gösterilmelidir ayrıca constans değişkeni kullanılırken tanımlanacak olan sabit büyük harflerle yazılmaktadır. Constants, kod içerisinde hata oluşumun önüne geçmektedir.

1.2.6. Eksik Veri (Missing Values)

Eksik veri; veri setinin içerisinde bulunan boş değerlerdir. Veri setinde eksik veri bulunması makine öğrenmesinde uygulanan algoritmalarının sonuçlarını değiştirebilir. Yüksek doğruluklu bir analiz için eksik veriler tespit edilmeli, silme ya da atama yöntemi ile düzeltilmelidir.

1.2.7. Çoklu Bağlantı Durumu

Çoklu bağlantı sorunu, iki veya daha fazla bağımsız değişken arasında doğrusal güçlü bir ilişki/ korelasyon oluşturan durumdur. Çoklu bağlantı durumu; bağımsız değişkenlerin birbiri arasında çoklu bağlantı durumu kontrol edilmediği taktirde tahminlerin katsayısını yükselterek analizden doğru sonuçlar elde etmeyi zorlaştırır. Çoğunlukla çoklu bağlantı sorununun çözülmesi için korelasyon katsayısı ve VIF (Variance İnflation Factor/ Varyans Enflasyon Faktörü) yöntemleri kullanılmaktadır.

1.2.8. Varyans Enflasyon Faktörü (VIF)

VIF (Variance İnflation Factor/ Varyans enflasyon faktörü); Makine öğrenmesinde ve regresyon analizinde bağımsız değişkenler arasındaki çoklu bağlantıyı ölçmek için en sık kullanılan yöntemlerden biridir. Bağımsız değişkenler arasındaki ilişkinin artması durumunda varyans enflasyon faktörünün de değeri artmaktadır bununla birlikte değişkenler arasındaki ilişkinin azalması durumunda ise varyans enflasyon faktörünün değeri azalmaktadır. VIF değeri 1'e eşit olması durumunda değişkenlerin birbirleri ile arasında hiçbir ilişki olmadığı anlamına gelmektedir. VIF değerinin 1 ile 5 arasında olması değişkenler arasında orta dereceli bir ilişkinin olduğu anlamına gelmektedir. Eğer VIF değeri 5 ile 10 arasında veya 10'dan daha fazla bir değere sahip ise değişkenler arasındaki ilişkinin oldukça yüksek olduğu anlamına gelmektedir.

Regresyon analizinde değişkenler arasındaki ilişkinin yüksek olması çoklu bağlantı sorununa yol açmaktadır bu da zararlı ve istenmeyen bir durumdur.

Bu sorun için ise modeldeki VIF değeri hesaplanmasına yüksek olan değişkenlerin çıkarılması daha iyi sonuçlar vermektedir. VIF değerinin hesaplanması için kullanılan formül aşağıda verilmiştir;

$$\boldsymbol{VIF}_{\mathrm{i}} = \frac{\mathbf{1}}{\mathbf{1}-\mathbf{R}_{\mathrm{i}}{}^{2}} \qquad \textbf{(3.6)}$$

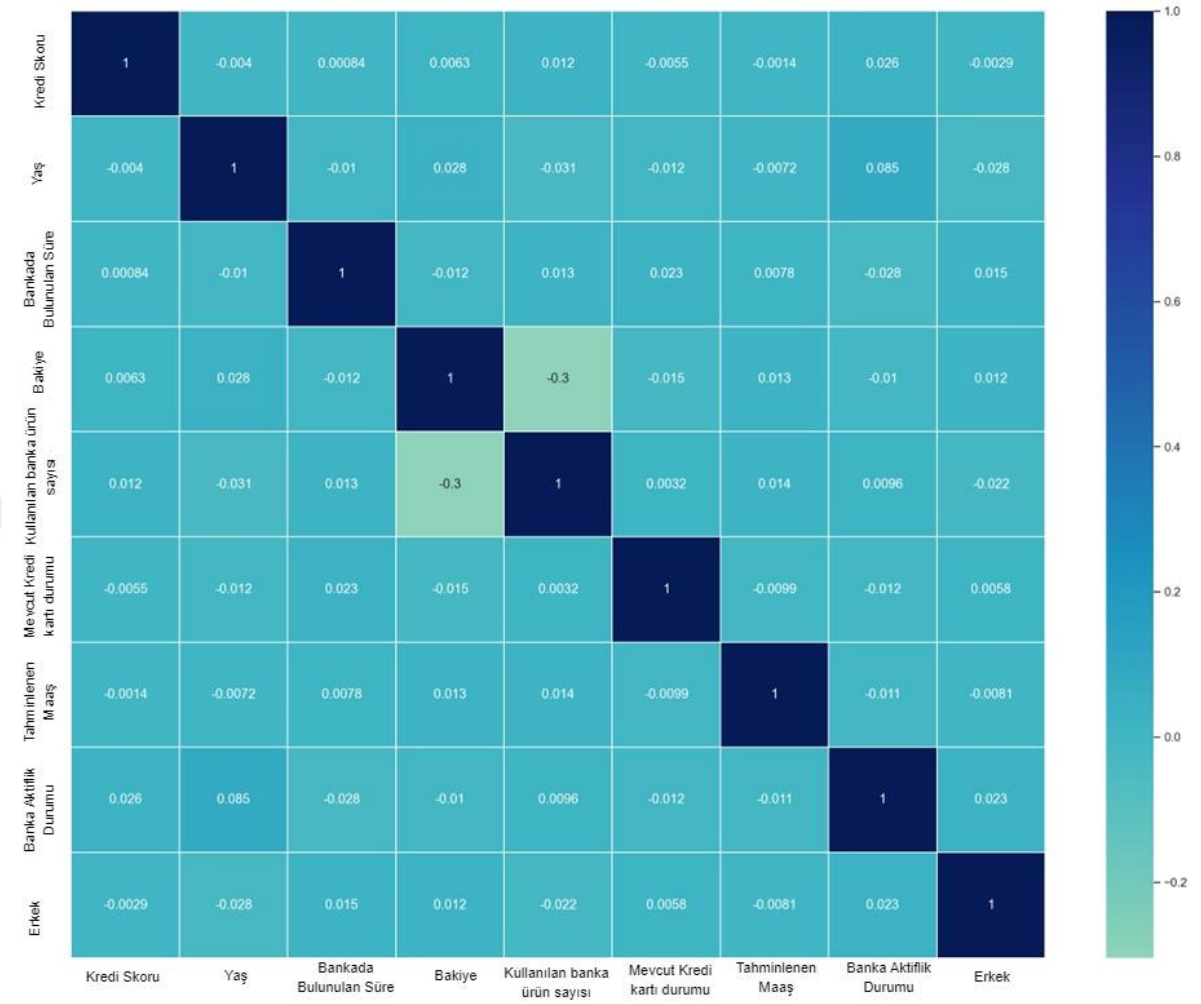
Burada; i, bağımsız değişken, Ri^2 ise bağımsız değişkenin diğer değişkenler üzerindeki regresyon için ayarlanmış olan belirleme katsayısıdır.

1.2.9. Korelasyon Analizi

İki ya da daha çok bağımlı ve bağımsız değişken arasındaki ilişkiyi bulan ve bu ilişkinin büyüklüğünü hesaplayan istatistiksel yönteme korelasyon analizi denmektedir. Korelasyon katsayısında değişkenler $-1 \leq r \leq 1$ aralığında bulunmaktadır. Korelasyon analizinde katsayı eğer 0 ile 1 arasında ise bağımlı değişken, -1 ile 0 arasında ise bağımsız değişken olarak tanımlanmaktadır. Eğer değer 0 ile 1 arasında ise değişkenler arasındaki ilişkinin pozitif yönlü olduğunu, -1 ile 0 arasında olması ise ilişkinin negatif yönlü olduğunu gösterir. Korelasyon analizi, çalışmalardaki anlamayı ve yorumlamayı kolaylaştırmaktadır. Korelasyon matrisinde; -1'e yakın değerler negatif yönlü, +1'e yakın değerler ise pozitif yönlü korelasyonlardır eğer değer 0'a yakınsa değişkenler arasında bağlantının olmadığı anlamına gelmektedir. Korelasyon aralığı ile ilişki düzeyi ayrıntılı bir şekilde Tablo 4'te verilmektedir;

Tablo 3.4: Korelasyon Aralığı Tablosu

Korelasyon Aralığı	İlişki Düzeyi
(-0,25) - 0 ve 0 - (0,25)	Çok Zayıf
(-0,49) – (-0,26) ve (0,26) - (0,49)	Zayıf
(-0,69)- (-0,50) ve (0,50) – (0,69)	Orta
(-0,89) - (-0,70) ve (0,70) – (0,89)	Yüksek
(-1) - (-0,90) ve (0,90) – 1	Çok yüksek



Şekil 3. 17: Veri Seti Korelasyon İlişkisi

Yukarıdaki korelasyon matrisine bakıldığında müşterinin kullandığı banka ürünlerinin sayısı (internet/mobil bankacılık vb.) ile müşterinin bankada bulunan bakiyesi değişkenleri arasındaki ilişkide bir değerin artarken diğerinin azaldığı görülmektedir Buna bağlı olarak negatif yönlü bir ilişkinin oluştuğu görülmektedir. Bununla birlikte banka aktiflik durumu değişkeni ile yaş değişkeni arasında pozitif yönlü bir ilişkinin olduğu görülmektedir.

1.3. Özellik Ölçeklendirme

Bir veri kümesindeki özelliklerin değerlerini, mesafe hesaplamasına orantılı olarak katkıda bulunacak şekilde ölçeklendirme işlemidir. En yaygın olarak kullanılan özellik ölçekleme tekniği standardizasyon (veya Z-skoru Normalleştirme) ve min-maks ölçeklendirmedir (Akyiğit ve Taşçı. 2022). Ölçeklendirmenin temel amacı; değişkenler

arasındaki ölçüm farklılığını düzeltmektir. Standardizasyon ve normalizasyon yöntemleri sınıflandırma algoritmalarının daha doğru bir şekilde eğitilmesini sağlamaktadır.

1.3.1. Standardizasyon (Z-Puanı Normalleştirme)

Veri setindeki özniteliklerin ortalamasının alınması ve standart sapmaya bölünmesiyle elde edilmektedir. Bu işlem veri kümesi içerisindeki öznitelikleri aynı ölçeğe getirdiği için eğitim süresinde bir iyileştirme sağlayabilmektedir (Ahmad ve Aziz, 2019). Standardizasyon işlemi hesaplamasında özniteliklerin ortalama değeri 0 ve standart sapma sapmaya bölünmesinde ise 1 değeri olmalıdır bu değerlere ise "standart puan/ Z-puanı" adı verilmektedir. Standardizasyon (Z-Puanı) formülü aşağıdaki gibidir;

$$\boldsymbol{Z = \frac{X-\mu}{\sigma}} \qquad \textbf{(3.7)}$$

Formüldeki; μ: ortalama değeri, σ: standart sapma değerini, x: değişkenin değerini vermektedir

1.3.2. Normalizasyon

Min-maks normalizasyonu olarak da adlandırılmaktadır. Veri setindeki bir veya birden çok sayısal özniteliğin minimum 0 değeri ile maksimum 1 değeri aralığında normalize ederek yeniden ölçeklendirmektedir. Normalizasyon, değişken değerlerini belirli bir aralıkta tutmayı ve aykırı değer oluşumunu önlemektedir. Min-max normalizasyonu formülü;

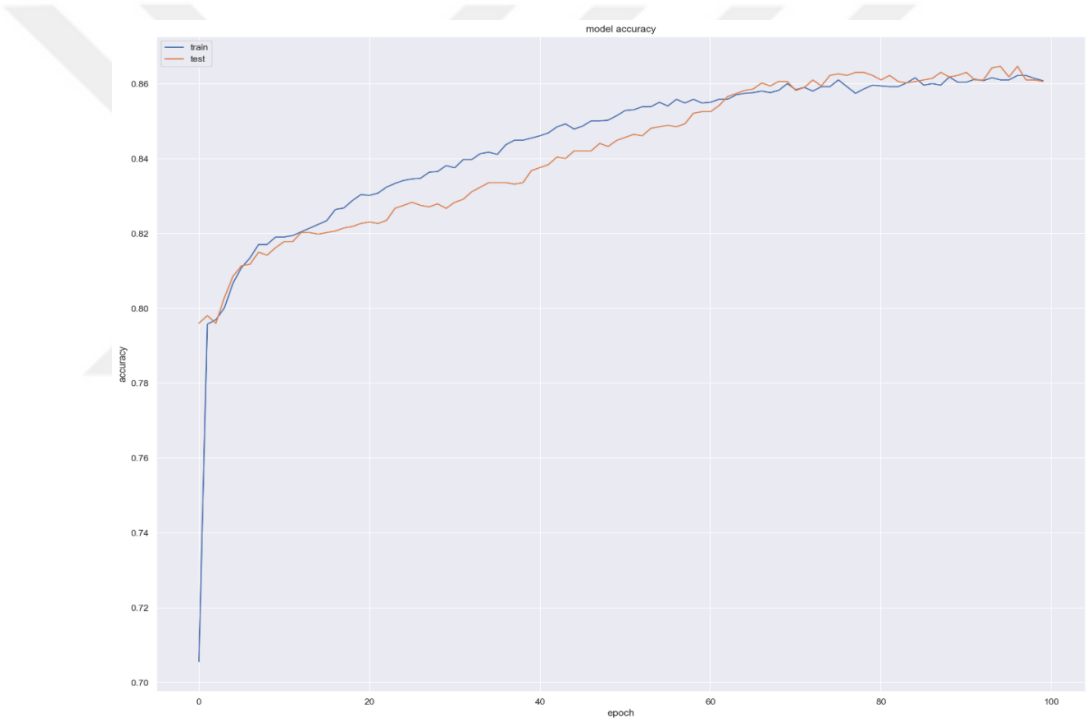
$$\boldsymbol{X' = \frac{X-\min(X)}{\text{maks}(X)-\min(X)}} \qquad \textbf{(3.8)}$$

Bu denklemde, min (X) ve max (X) her bir özniteliğin minimum ve maksimum değeridir. X ise, asıl değer anlamına gelmektedir.

1.4. Eğitim ve Test Kümesi Oluşturma

Veri setindeki gerekli işlemlerin (veri önişleme, veri temizliği vb.) yapılmasının ardından eğitim ve test kümesi olmak üzere rastgele bir şekilde ikiye ayrılmaktadır. Veri setinin ikiye ayrılmasının ardından eğitim veri kümesinde bulunan veriler modeli eğitmek için kullanılırken test verileri ise modelin doğruluğunu ve performansını ölçmek için kullanılan veri kümeleridir. Verilerin eğitim ve test kümelerine ayrılmasının amacı; makine öğrenimindeki modellerin performansını veriler üzerinde tahminlemektir. Python sklearn kütüphanesinde bulunan ve eğitim ve test kümelenmesinde kullanılan train-test

split methodu; Veri setinin eğitim ve test olarak rastgele ayrılmasını sağlar, makine öğrenmesi algoritmalarının performansını değerlendirmeye yönelik bir işlemdir. Uygulamada, veri önişleme aşaması tamamlanan veri setinden rastgele seçilen %30'luk dilimde test veri seti, %70'lik dilim ise veri eğitim veri seti olarak seçilmiştir. Veri setinde toplamda 10.000 adet veri bulunmakla birlikte eğitim veri kümesine 7.000 adet ve test veri kümesine 3.000 adet veri ayrılmıştır. Epoch grafiğinin oluşturulması için tensorflow ve keras kütüphaneleri kullanılmıştır. Veri setindeki eğitim ve test verilerinin Epoch grafiği ile gösterimi aşağıdaki gibidir. Modelde bankacılık sektörü müşteri kaybı veri seti kullanılmış olup epoch=6, bach_size =10 değerleri verilmiştir.



Şekil 3. 18: Eğitim ve Test Verilerinin Epoch Grafiği

1.5. Model Oluşturma

Çalışmada mevcut veri setini test ve eğitim olarak oluşturulmasının ardından model oluşturma aşamasına geçilmiştir. 10.000 banka müşterisinden alınan veriler 2 bölüme ayrılmıştır. Verilerin %70'i eğitim, %30 ise test etmek için kullanılmıştır. Çalışmada; Lojistik Regresyon, Karar Ağacı, Rastgele Orman, K-en yakın komşu, Hafif Gradyan Arttırma, Kategorik güçlendirme algoritmaları kullanılmıştır. Çalışmada modellerin tahminleme başarısının hesaplanması için; Accuracy (Doğruluk oranı),

Precision (Kesinlik), Recall (Duyarlılık) ve F1 skor metrikleri kullanılarak veri setinin modellemesi yapılmıştır.

1.6. Oluşan Modellerin Değerlendirilmesi

Bu uygulama, banka müşterilerinin bankadan ayrılıp ayrılmayacağını önceden tahminlemek ve müşteri kaybını önlemek amacıyla yapılmıştır. Bu bölümde oluşturulan modellerin performanslarının değerlendirilmesine yer verilmiştir. Çalışmada önerilen tahmin modelini bulmak için 6 farklı makine öğrenmesi sınıflandırma algoritması kullanılmıştır. Bu algoritmalar sırasıyla aşağıdaki gibi listelenmiştir;

- Lojistik Regresyon
- Karar Ağacı
- Rastgele orman
- K-en yakın komşu
- LightGBM (Hafif Gradyan Arttırma)
- Catboost (Kategorik Güçlendirme)

Çalışılan modeller arasında en iyi modeli bulmak amacıyla F1 skor, duyarlılık, kesinlik, doğruluk oranı olmak üzere 4 adet performans değerlendirme ölçütü kullanılmıştır. Çalışmada seaborn, numpy, pandas, matplotlib, scikit-learn kütüphaneleri kullanılmıştır.

Matplotlib; veri görselleştirmesi ve grafik çizimi için kullanılan bir Python kütüphanesidir.

Seaborn; verileri anlamlandırmayı sağlayan veri görselleştirmesi ve grafik çizimi için kullanılan matplotlib tabanlı bir Python kütüphanesidir.

Numpy; büyük boyutlu dizi ve matrislerle çalışmayı sağlayan ve sayısal hesaplamalar yapan bir Python kütüphanesidir.

Pandas; veri işlemesi, veri analizi gibi alanlarda kullanılan bir Python kütüphanesidir.

Scikit-Learn; makine öğrenimi için kullanılan bir Python kütüphanesidir. Çizim için matplotlib dizi vektörleştirme için numpy kitaplıkları ile bağlantılı çalışmaktadır.

Veri setinin analizinde kullanılan modellerin performanslarını ölçmek için ilk olarak StandartScaler parametresi kullanılmıştır. Bu performans ölçüm sonucu aşağıdaki gibidir;

Tablo 3.5: Uygulanan Makine Öğrenimi Modellerinin Performans Sonuçları

Makine Öğrenimi Modelleri	Doğruluk Oranı	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	0.8110	0.581818	0.237037	0.336842
KNN	0.8270	0.606498	0.414815	0.492669
Karar Ağacı	0.8055	0.507830	0.560494	0.532864
Rastgele Orman	**0.8695**	**0.749117**	**0.523457**	**0.616279**
LGBM	0.8595	0.697452	0.540741	0.609179
CatBoost	0.8635	0.724490	0.525926	0.609442

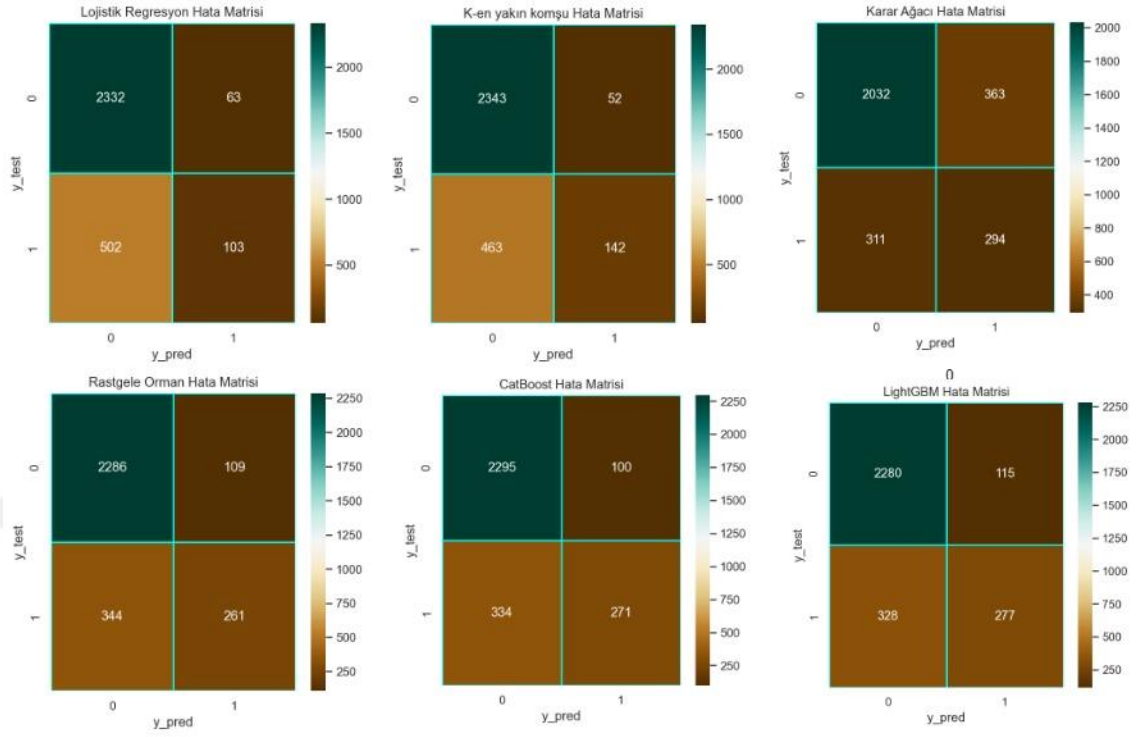
Bu çalışmada modellerin performansı ölçülmeden önce banka müşteri kaybı analizi veri kümesine ait veriler alınarak veri ön işleme işlemi gerçekleştirilmiştir. Verilerin %70 eğitim %30 test olmak üzere bölünerek eğitilmeye hazır hale getirilmiştir.

Etkili sınıflandırma yöntemleri ve tahminleme ölçütleri kullanılarak en yüksek sonucu veren sınıflandırma modeli en iyi sonuç olarak kabul edilmiştir. Modellerin tahminlenmesi için Doğruluk oranı, Kesinlik, Duyarlılık ve F1 Skor olmak üzere 4 adet performans ölçütü kullanılmıştır. Doğruluk oranı ölçütü, bankadan ayrılan ve ayrılmayan müşterilerin sınıflarını ayırt etme başarısını göstermektedir. Bu çalışmada, K-en yakın komşu, Karar ağacı, Lojistik regresyon, LightGBM (hafif gradyan arttırma), Rastgele orman, CatBoost (kategorik güçlendirme) sınıflandırma yöntemlerinden elde edilen performans sonuçları Tablo 3.6'da gösterilmiştir. Optimum düzeyde sonuç elde edebilmek için aşırı öğrenmeyi engelleyen SMOTE parametresi kullanılmıştır. Bu parametrenin kullanımından sonraki modellerin performans çıktısı aşağıdaki gibidir;

Tablo 3.6: Uygulanan Makine Öğrenimi Modellerinin Performans Sonuçları

Makine Öğrenimi Modelleri	Doğruluk oranı	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	0.715281	0.718935	0.710526	0.714706
K-NN	0.833822	0.780112	0.930660	0.848762
Karar ağacı	0.826915	0.815546	0.845865	0.830429
Rastgele orman	0.888866	0.882232	0.898079	0.890085
LGBM	0.903516	0.923718	0.880117	0.901390
CatBoost	**0.904353**	**0.931019**	**0.873851**	**0.901530**

Tüm sonuçlar incelendiğinde 6 model arasından en yüksek doğruluğu veren sınıflandırma modeli %90,43 oranı ile Catboost algoritması olarak bulunmuştur. En düşük performans ise %71,81 oranı ile Lojistik regresyon modeli olmuştur. Tablo 3.5'te verilen performans sonuçlarında en yüksek performansı Rastgele orman vermiştir. RO algoritmasının sınıflandırma algoritmaları arasında yüksek performans göstermesi diğer algoritmalar arasında daha yüksek tahminleme özellikleri mevcuttur ancak karmaşık bir yapıya sahip olmakla birlikte içerisinde çok sayıda karar ağacı kullanıldığı için tahminlemede yavaş ve yorumlanması zor bir algoritmadır. Tablo 3.6'da ise performans sonuçlarının dengelenmesi için SMOTE parametresi kullanımı ile birlikte kategorik güçlendirme (Catboost) algoritması yüksek çıktığı görülmüştür. Catboost algoritması aşırı öğrenmenin önüne geçerek veri setinin performansını arttırmayı bununla birlikte veri hazırlığı süresinin kısaltılmasını sağlamaktadır. Catboost algoritmasını %0.9035 oranı ile LightGBM algoritması takip etmiştir. Çalışma sonucunda, modellerin performans sonuçları karşılaştırmalı olarak ısı haritası ile ayrıntılı bir şekilde aşağıda sunulmuştur.



Şekil 3. 19: Veri Setine Uygulanan Sınıflandırma Algoritmalarının Performans Sonuçlarının Isı Grafiği ile Gösterimi

Yukarıdaki görselde Heatmap tekniği ile, veri setine uygulanan modellerin performans sonuçlarının karşılaştırılmalı olarak ısı haritası ile veri görselleştirme çalışması yapılmıştır. Çalışmada; Hata matrisi (Confusion matrix) kullanılarak her bir sınıflandırma algoritmasının performansı değerlendirilmiştir. Karışıklık matrisi mevcut veri setindeki gerçek veriler ile tahminlenmiş olan verileri karşılaştırarak doğruluk oranları ve hata oranlarını göstermektedir. Sınıflandırma algoritmalarının veri seti üzerindeki performansının analiz edilebilmesi karışıklık matrisi ile ölçülebilmektedir. Çalışmada kullanılan ısı haritası, karışıklık matrisinin verdiği sonuçları renk ölçeği ile veri görselleştirmesini yaparak sonuçların kolay yorumlanmasını sağlamaktadır. Şekil 3.19'da verilen performans sonuçlarına bakıldığında çalışmada kullanılan sınıflandırma algoritmaları arasında başarısızlık oranının en düşük olduğu sınıflandırıcı CatBoost algoritması olmakla birlikte Karar Ağacı algoritması ise en yüksek başarısızlık oranına sahiptir.

SONUÇ

Yapay zekanın alt alanı olan makine öğrenmesi, bankacılık sektörü ve diğer sektörlerdeki müşteri kaybının önlenmesi için önemli bir alandır. Bankacılık sektörü müşteri ilişkilerine dayalı bir sektördür Yeni bir müşteri edinme maliyeti, mevcut müşterileri elde tutma maliyetinin yaklaşık 5 katı, memnun olmayan müşteriyi geri kazanmanın maliyeti mevcut müşteriyi elde tutmanın yaklaşık 10 katıdır (Massey ve Mitzi, 2001). Bunun için müşterilerin bankadan ayrılmadan önce takibi yapılmalı ve gerekli önlemleri alarak her müşteri için kişiye özel hizmetler sunularak müşterilerin elde tutulmasını sağlamalı ve şirketin zarara uğramasının önüne geçilmelidir. Çalışmanın asıl amacı; banka ve müşteri sadakati ile ilgili tahminleme işlemlerine en uygun ve en doğru makine öğrenmesi yöntemini bulmaktır. Çalışmada banka müşterilerinin bankayı terk etme olasılığı üzerine çeşitli sınıflandırma modelleri kullanılarak müşteri kayıp analizi yapılmıştır. Banka müşterilerinin bankayı terk etme olasılığı hesaplanarak müşterinin bankadan ayrılacağını önceden tespit edilmesi ve erken aksiyon alınması amaçlanmıştır. Çalışma; veri ön işleme, veri hazırlık, eğitim ve test setlerine bölme, modelleme ve değerlendirme aşamalarından oluşmaktadır.

Bu tez çalışmasında, Kaggle sitesinden alınan banka müşteri kaybı tahmini hazır veri seti kullanılmıştır. Veri seti; 3 ülkenin banka müşteri verilerinden ve 14 adet öznitelikten oluşmaktadır bununla birlikte 10,000 adet müşteri verisi ile çalışılmıştır. Bu çalışmada farklı sınıflandırma algoritmaları kullanılarak banka müşteri kaybı tespitinde en uygun makine öğrenmesi yöntemi tespit edilmeye çalışılmıştır. Çalışmada gerekli yöntemlerin hesaplanıp analiz edilebilmesi, görselleştirilmesi için Spyder programında Python programlama dili kullanılmıştır. Kullanılan veri setinin analizi için veriler %70 eğitim %30 test ve %80 eğitim %20 test olarak değerlendirilmiş olup ve en iyi sonucu %70 eğitim %30 test seti vermiş ve bu oranlarla çalışmaya devam edilmiştir. Çalışmada bankacılık sektöründeki müşteri sadakatini ölçmek için makine öğrenmesi algoritmalarından; Lojistik regresyon, K-en yakın komşu, Karar Ağacı, Rastgele Orman, LightGBM (Hafif gradyan arttırma), CatBoost (kategorik güçlendirme) modelleri ve Numpy, pandas, seaborn, matplotlib kütüphaneleri kullanılmıştır. Bu çalışmada farklı sınıflandırma algoritmaları kullanarak en uygun teşhis aracı bulunması ve bankadan ayrılma olasılığı yüksek olan müşterilerin erken teşhis edilmesi amaçlanmıştır. Modelin

tahminlenmesi ve performans ölçümü için F1 skor, kesinlik, duyarlılık, doğruluk oranı metrikleri kullanılmıştır.

Çalışmada eksik değer bulunmamakla birlikte satır numarası, müşteri numarası, soyad sütunları silinmiştir. Veri setinde bulunan Cinsiyet ve ülke sütunlarına kategorik değişken oldukları için tek sıcak kodlama tekniği uygulanmıştır. Çalışmada sağlıklı düzeyde sonuç elde edebilmek için sınıf dengesini sağlayan Smote tekniği kullanılmıştır. Kullanılan algoritmaların performans sonuçları değerlendirildiğinde en yüksek doğruluğu %90,43 oranı ile CatBoost modeli vermiştir. Bu çalışma, bankacılık sektörü gibi müşteri odaklı firmalar için makine öğrenmesi yöntemlerini kullanarak müşteri verilerinden anlamlı sonuçlar çıkarılması ve gelecekte yaşanabilecek kayıpları ön görerek gerekli aksiyonların alınmasını bununla beraber rekabet gücünün arttırılmasını sağlamak ve müşteri kaybının önlenerek firmaların zarara uğramasını engellemek amacıyla gerçekleştirilmiştir. Bu veri analiz çalışmasında Smote parametresinin kullanılması modelin performansını arttırdığı ve daha yüksek sonuçlar elde edildiği görülmüştür. Gelecekte CatBoost algoritmasının müşteri sadakatini ve müşteri kaybını tespit etmesindeki çalışmaları incelenebilir.

KAYNAKÇA

AHMAD, T.; M. N. AZİZ (2019) "Data Preprocessing and Feature Selection For Machine Learning İntrusion Detection Systems", **ICIC Express Lett**, 13(2), 93-101.

AKPINAR, H. (2014) "**Data-Veri Analizi, Veri Madenciliği**", Papatya Yayıncılık, İstanbul, ISBN: 978-605-4220-816.

AKYİĞİT, H.E.; T. TAŞÇI (2022) "Sigortacılık Sektöründe Makine Öğrenmesi ile Müşteri Kaybı Analizi", **Tasarım Mimarlık ve Mühendislik Dergisi** 2, sy 1: 66-79.

ALASADI, S. A.; W. S. BHAYA (2017) " Review of data preprocessing techniques in data mining", **Journal of Engineering and Applied Sciences**, 12(16), 4102-4107.

ALFADHİLİ, H. L. M. (2020) " A new security system in android os combining power spectral density and waevelet transform based on KNN", (Master's thesis, **Altınbaş Üniversitesi**/Lisansüstü Eğitim Enstitüsü).

AMUDA, K. A.; A. B. ADEYEMO (2019) "Customers churn prediction in financial institution using artificial neural network", **arxiv preprint arxiv**:1912.11346.

ARYA, S.; ve diğerleri (1998) "An Optimal Algorithm For Approximate Nearest Neighbor Searching Fixed Dimensions", **Journal of the ACM 45**, sy 6: 891-923. https://doi.org/10.1145/293347.293348.

ATCILI, A. (2022), "**Knn- (K-En Yakın Komşu)**" Web: https://medium.com/machine-learning-t%C3%BCrkiye/knn-k-en-yak%C4%B1n-kom%C5%9Fu-7a037f056116

AYTUĞ, O. (2015) "Şirket İflaslarının Tahminlenmesinde Karar Ağacı Algoritmalarının Karşılaştırmalı Başarım Analizi", **Bilişim Teknolojileri Dergisi**, 8(1).

BERSON, A.; K. THEARLİNG (1999) "CRM İçin Veri Madenciliği Uygulamaları Oluşturma", **McGraw-Hill**, Inc.

BİLGEN, O. (2009) "Churn Management of Electronic Banking Customers", (**Doctoral dissertation, Fen Bilimleri Enstitüsü**).

BİRANT, D. (2011) " Comparison of Decision Tree Algorithms for Predicting Potential Air Pollutant Emissions with Data Mining Models", **Journal of Environmental Informatics**, 17(1).

BOZYİĞİT, B.Ş.; Ç. TARHAN (2020) "Makine Öğrenmesi Yöntemleri Kullanılarak Sosyal Medyada Marka İtibar Analizi", **Yönetim Bilişim Sistemleri Dergisi**, 6(2), 57-76.

BREİMAN, L., J. FRİEDMAN, R. OLSHEN; C. STONE (1984) "**Cart. Classification and Regression Ttrees**".

BREİMAN, L. (1997) "Arcing the Edge ", (pp. 1-14) Technical Report 486, Statistics Department, **University of California at Berkeley.**

BREİMAN, L. (2001) "**Random Forest", Machine Learning, 45, 5-32**. "Random Forests". Erişim20/05/2023.

BUDAK, İ., B. ŞEN; M. Z. YILDIRIM (2013) "Lojistik Regresyon ile Bilgisayar Ağlarında Anomali Tespiti", **Akademik Bilişim.**

"Churn Modelling Kaggle", Erişim 16 Mayıs 2023 "Çevrimiçi" https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling/code.

CUTLER, D. R.; ve diğerleri (2007) "Random Forests for Classification in Ecology", **Ecology**, 88(11), 2783-2792.

CHAPMAN, P.; ve diğerleri (2000), CRISP-DM 1.0: **Step-by-step data mining guide**. SPSS inc, 9(13), 1-73.

COX, D. (1969) "**Analysis of Binary Data**", Chapman and Hall, London.

ÇALIŞ, A., S. KAYAPINAR; T. ÇETİNYOKUŞ (2014) "**Veri Madenciliğinde Karar Ağacı Algoritmaları ile Bilgisayar ve İnternet Güvenliği Üzerine Bir Uygulama**", Endüstri Mühendisliği, 25(3), 2-19.

ÇALIŞKAN, S. K.; İ. SOĞUKPINAR (2008) "**Kxknn: K-Means ve K En Yakin Komşu Yöntemleri ile Ağlarda Nüfuz Tespiti**", EMO Yayınları, 120, 24.

ÇİÇEK, A., & ARSLAN, Y. (2020). Müşteri Kayıp Analizi İçin Sınıflandırma Algoritmalarının Karşılaştırılması. **İleri Mühendislik Çalışmaları ve Teknolojileri Dergisi**, 1(1), 13-19.

ÇOKLUK, Ö. (2010) "**Lojistik Regresyon Analizi: Kavram ve Uygulama**", Kuram ve uygulamada eğitim bilimleri, 10(3), 1357-1407.

DALMİA, H., C.V.S.S. NİKİL; S. KUMAR (2020) "Churning of Bank Customers Using Supervised Learning", https://doi.org/10.1007/978-981-15-3172-9_64. **A visual comparison between the complexity of decision trees and random forests**, 2020.

DİLKİ, G.; Ö. D. BAŞAR (2020) "İşletmelerin İflas Tahmininde K-en Yakın Komşu Algoritması Üzerinden Uzaklık Ölçütlerinin Karşılaştırılması", **İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi**, 19(38), 224-233.

DOMİNGO, R.T. (2003) "**Applying Data Mining to Banking**". Erişim 19 Mayıs 2023. https://www.rtdonline.com/BMA/BSM/4.html.

DOMİNGOS E., B. OJEME; O. DARAMOLA (2021) "**Experimental Analysis of Hyperparameters for Deep Learning-Based Churn Prediction In the Banking Sector**", Computation, 9(3), 34.

DOROGUSH, A. V., V. ERSHOV; A. GULİN (2018) "**CatBoost: Gradient Boosting with Categorical Features Support**", arXiv, ss. 1–7. Erişim Adresi: http://arxiv.org/abs/1810.11363.

DUR, R., S. KOÇER; Ö. DÜNDAR (2022) " Evaluation of Customer Loss Analysis for Marketing Campaigns in the Banking Sector", **Politeknik Dergisi**, 1-1.

ELASAN S. (2019) "Classification of Variables Affecting Birth Weight by Decision Trees and K-Nearest Neighbor Methods", https://www.academia.edu/69933353/Classification_of_Variables_Affecting_B ,**International Journal of Scientific and Technological Research.**

FAYYAD, U., G. PIATETSKY-SHAPİRO; P. SMYTH (1996) "From Data Mining to Knowledge Discovery in Databases". **AI Magazine** 17, sy 3: 37-37. https://doi.org/10.1609/aimag.v17i3.1230.

GÖK M. (2017) "Makine Öğrenmesi Yöntemleri İle Akademik Başarının Tahmin Edilmesi", **Gazi University Journal Of Science Part C: Design And Technology**, 5(3), 139-148.

GÖRMEZ, B. (2021) "Adli Bilişimde Makine Öğrenmesi: Makine Öğrenmesi Algoritmaları ile Terör Olaylarının Tahmin Edilmesi Çalışması", Yüksek lisans tezi, **Ankara üniversitesi**, Ankara.

GUPTA, H.; ve diğerleri (2021) "Meme Kanseri Tahmini İçin Kategori Artırıcı Makine Öğrenme Algoritması", **Reve Roumaine Des SciencesTechniques—Seire Électrotecnique et Éneretique** , 66 (3), 201-206.

GÜRCAN, M. (1998) "Lojistik Regresyon Analizi ve Bir Uygulama", Yayınlanmamış yüksek lisans tezi, **Ondokuz Mayıs Üniversitesi**, Fen Bilimleri Enstitüsü, Samsun.

HAN, J., J. PEİ, M. KAMBER (2012) "**Data Mining: Concepts and Techniques, Elsevier**", USA, ISBN: 978-0-12-381479-1.

HANCOCK, J. T.; T. M. KHOSHGOFTAAR (2020) "CatBoost for Big Data: An İnterdisciplinary Review", **Journal of big data**, 7(1), 1-45.

HE, B.; ve diğerleri (2014) "Prediction of Customer Attrition of Commercial Banks Based on SVM Model", **Procedia computer science**, 31, 423-430. https://doi.org/10.1016/j.procs.2014.05.286.

HUANG, G.; ve diğerleri (2019) "Evaluation of CatBoost Method for Prediction of Reference Evapotranspiration in Humid Regions", **Journal of Hydrology**, 574, 1029-1041.

IBRAGIMOV, B.; G. GUSEV (2019) "Minimal Variance Sampling in Stochastic Gradient Boosting", **Advances in Neural Information Processing Systems**, 32.

İNAL, M. E., D. TOPUZ; O. UÇAN (2006) "Doğrusal Olasılık ve Logit Modelleri ile Parametre Tahmini", **Sosyoekonomi**, 3(3).

JURAFSKY, D., J. MARTİN (2023) "Logistic Regression", Erişim Tarihi 03 Haziran 2023. "Çevrimiçi" https://web.stanford.edu/~jurafsky/slp3/5.pdf. **Speech and Language Processing**, 11-17.

KARADOĞAN, A. (2014) " Takviyeli Öğrenme için Yapay Atom Algoritması (A3) Kullanımı.", Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı, **İnönü Üniversitesi**, Malatya.

KARAKAYA, R.; S. KAZAN (2021) "Handwritten Digit Recognition Using Machine Learning", **Sakarya university journal of science**, 25(1), 65-71.

KARAOĞLU, T. T.; H. Okut (2020) "Classification of the Placement Success in the Undergraduate Placement Examination According to Decision Trees with Bagging and Boosting Methods", **Cumhuriyet Science Journal**, 41(1), 93-105.

KASIM, S. (2022) "Veri Madenciliği Yöntemleriyle Müşteri Kaybı Analizi: Yazılım Sektörü= Customer Churn Analysıs wıth Data Mınıng Methods: Software Industry".

KE, G.; ve diğerleri (2017) "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree", **Advances in neural information processing systems**, 30. k-nearest neighbor algorithm, http://en.wikipedia.org/wiki/K-earest_neighbor_algorithm.

KOKLU, M., K. SABANCI; M.F. UNLERSEN (2016) "Classification of Heuristic Information by Using Machine Learning Algorithms", **International Journal of Intelligent Systems and Applications in Engineering**, Special Issue , 252-254 . Retrieved from https://dergipark.org.tr/tr/pub/ijisae/issue/25999/281903

KUMAR, S. (2021) "**3 Techniques to Avoid Overfitting of Decision Tree**", Web: https://towardsdatascience.com/3-techniques-to-avoid-overfitting-of-decision-trees-1e7d3d985a09

KUNT M. (2019) "Telekomünikasyan Sektöründe Müşteri Kaybı Analizi", Yüksek lisans tezi, **Ankara üniversitesi**, Ankara.

LAROSE, D.T.; C.D. LAROSE (2014) "Discovering Knowledge in Data: An Introduction to Data Mining", **John Wiley & Sons**.

LEWİS, R. J. (2000) "An Introduction to Classification and Regression Tree (CART) Analysis", **In Annual meeting of the society for academic emergency medicine in San Francisco, California (Vol. 14)**. San Francisco, CA, USA: Department of Emergency Medicine Harbor-UCLA Medical Center Torrance.

LU, B., M. CHARLTON, C. BRUNSDON; P. HARRİS (2016) "The Minkowski Approach for Choosing the Distance Metric in Geographically Weighted Regression", **International Journal of Geographical Information Science**, 30(2), 351-368.

MAHAJAN, V., R. MİSRA; R. MAHAJAN (2015) "Review of Data Mining Techniques for Churn Prediction in Telecom", **Journal of Information and Organizational Sciences** 39, sy 2 : 183-97.

MASSE, A.P., M.M. MONTOYA-WEISS; K. HOLCOM (2001) "Re-Engineering the Customer Relationship: Leveraging Knowledge Assets at IBM", Decision Support Systems, **Decision Support Issues in Customer Relationship**

Management and Interactive Marketing for E-Commerce, 32, sy 2: 155-70. https://doi.org/10.1016/S0167-9236(01)00108-7.

MAHESH, B. (2018) "**Machine Learning Algorithms**", Web: https://www.researchgate.net/profile/BattaMahesh/publication/344717762_Machine_Learning_Algorithms_A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf?eid=5082902844932096, DOI: 10.21275/ART20203995

MAHESH B. (2019) "**Machine Learning Algorithms -A Review**", DOI: 10.21275/ART20203995

MHASKE, S. (2022) "Use of Machine Learning for Customer Churn Analysis in Banking", **International Research Journal of Modernization in Engineering Technology and Science.**

MİNKOVSKY, P., K. MATOUSEK; Z. KOUBA (2002) "Data Pre-Processing Support for Data Mining", **In IEEE International Conference on Systems**, Man and Cybernetics (Vol. 5, pp. 4-pp). IEEE.

MİTCHELL, T. (2006) "The Discipline of Machine Learning ", (**Technical Report** CMUML-06-108).

NAHZAT, S.; M. YAĞANOĞLU (2021) "Classification of Epileptic Seizure Dataset Using Different Machine Learning Algorithms and PCA Feature Reduction Technique", **Journal of Investigations on Engineering and Technology**, 4(2), 47-60.

OMOTEHINWA, T.O., D. O. OYEWOLA; E. G. DADA (2023) "A Light Gradient-Boosting Machine algorithm with Tree-Structured Parzen Estimator for breast cancer diagnosis", **Healthcare Analytics**, 100218.

ÖNAL, B. (2017) "Makine Öğrenmesi Yöntemleri ile Müşteri Kaybı Analizi", Yüksek lisans tezi, **İstanbul aydın üniversitesi**, İstanbul.

ÖZDAMAR, L.; E. EKİNCİ (2002) "A Logistics Planning Decision Support System for Deployment of Military Forces in Military Crises and Natural Disasters", **Savunma Bilimleri Dergisi**, 1(1), 1-28.doi: 10.17134/sbd.32054.

PAMUNGKAS, F. S., B. D. PRASETYA; I. KHARİSUDİN (2020) "Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers

Menggunakan Python", **In PRISMA, Prosiding Seminar Nasional Matematika** (Vol. 3, pp. 692-697).

PARK, H. A. (2013) "An Introduction to Logistic Regression: From Basic Concepts to İnterpretation with Particular Attention to Nursing Domain", **Journal of Korean Academy of Nursing**, 43(2), 154-164.https://doi.org/10.4040/jkan.2013.43.2.154.

RAEİSİ, S.; H. SAJEDİ (2020) " E-Commerce Customer Churn Prediction By Gradient Boosted Trees", In 2020 10th **International Conference on Computer and Knowledge Engineering** (ICCKE) (pp. 055-059). IEEE.

RACHKA, S. (2020) "**Introduction to Deep Learnin and Generative Models**", Web: https://github.com/rasbt/stat453-deep-learning-ss20/blob/master/L01-intro/L01-intro_slides.pdf

RAHMAN, M.; V. KUMAR (2020) "Machine learning based customer churn prediction in banking", In 2020 4th **international conference on electronics, communication and aerospace technology** (ICECA) (pp. 1196-1201). IEEE.

SAİNİ, N., G. K. MONİKA; K. GARG (2017) "Churn Prediction in Telecommunication Industry Using Decision Tree", **International Journal of Engineering Research and Technology**, 6, 439-443.

SANKOFF, D.; W. LABOV (1979) "On the Uses of Variable Rules", **Language in society**, 8(2-3):189–222.

SAYED, H., M. A. ABDEL-FATTAH; S. KHOLİEF (2018) "Predicting Potential Banking Customer Churn Using Apache Spark ML and MLlib Packages: a Comparative Study", **International Journal of Advanced Computer Science and Applications**, 9(11).

SCHAPİRE, R. E. (1990) "The Strength of Weak Learnability", **Machine learning**, 5, 197-227.

SCHAPİRE, R. E.; Y. Freund (2012) "Boosting: Foundations and Algorithms", 1621 **Cambridge**, MA.

SEVLİ, O. (2019) "Performance Comparison of Different Machine Learning".

SEVLİ, O. (2019) "Göğüs Kanseri Teşhisinde Farklı Makine Öğrenmesi Tekniklerinin Performans Karşılaştırması", **Avrupa Bilim ve Teknoloji Dergisi**, (16), 176-185.

SİLAHTAROĞLU, G. (2008) "**Kavram ve Algoritmalarıyla Temel Veri Madenciliği**", Papatya Yayıncılık Eğitim, İstanbul, ISBN:978-975-6797-81-5.

SINGH, S.; P. GUPTA (2014) "Comparative Study ID3, Cart and C4. 5 Decision Tree Algorithm: A Survey", **International Journal of Advanced Information Science and Technology** (IJAIST), 27(27), 97-103.

STEPHENSON, B. (2008) "**Binary Response and Logistic Regression Analysis**", Web: www.public.iastate.edu / ~stat415/stephenson/stat415_chapter3.pdf

SUCHETANA, B., B. RAJAGOPALAN; J. SİLVERSTEİN (2017) "Bir Regresyon Ağacı Modeli Kullanılarak Atık Su Arıtma Tesisi Uygunluğunun Azalan Amonyak Deşarj Limitlerine Uygunluğunun Değerlendirilmesi", **Toplam çevre bilimi** , 598 , 249-257.

SİLİPO, R. (2019) "**Tek Karar Ağacından Rastgele Ormana**", Web: https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147

SÖZKESEN, O. (2021) "Sefalometrik Röntgen Görüntülerinde Hog ve Knn Kullanarak Önemli Noktaların Yer Tespiti", Yüksek lisan tezi, **İstanbul Aydın Üniversitesi**.

SUGUNA, R.; ve diğerleri (2022) "An Optimal Classifier Discovery for Diagnosing the Account Health in Financial Firms and A Study of Classifier Performance on İmbalanced Data", **Research Square,** https://doi.org/10.21203/rs.3.rs-1242944/v1.

SUYANTO, M. L. T. D. (2018) Machine Learning, Lanjut. Bandung: Informatika.

ŞAHİN, H.; D. İÇEN (2021) "Application of Random Forest Algorithm for the Prediction of Online Food Delivery Service Delay", **Turkish Journal of Forecasting** , 05 (1) , 1-11 . DOI: 10.34110/forecasting.842180

TAŞÇI E.; A. ONAN (2016) "K-en Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi", **Akademik Bilişim**, 1(1), 4-18.

TERZİ, Y. (2021) "**Lojistik Regresyon Analizi**", Web: https://avys.omu.edu.tr/storage/app/public/yukselt/118305/Lojistik%20regresyon%20analizi.pdf

TEKİN, A. T.; C. SARI (2022) "Mağaza Tabanlı Talep Tahmini: Topluluk Öğrenmesi Yaklaşımları ile Zaman Serisi Yaklaşımı Karşılaştırılması", **International Engineering and Technology Management Summit.**

TEKOUABOU, S. C.; ve diğerleri (2022) "Towards Explainable Machine Learning for Bank Churn Prediction Using Data Balancing and Ensemble-Based Methods", **Mathematics**, 10(14), 2379.https://doi.org/10.56726/IRJMETS29877.

YALÇIN, C. (2019) "Müşteri Kayıp Analizi (Customer Churn Analysis)". **YBS Ansiklopedi** Cilt 7, Sayı 1, 2019

YOUSEFİ, T., M. S. ODABAŞ; R. OKTAŞ (2020) "Kümeleme Algoritmalarında Kullanılan Farklı Yöntemlere Genel Bakış", **Black Sea Journal of Engineering and Science**, 3 (4), 173-189. DOI: 10.34248/bsengineering.698741

ZHANG, T. (2022) "Prediction and Clustering of Bank Customer Churn Based on XGBoost and K-means", **BCP Business & Management** 23: 360-66. https://doi.org/10.54691/bcpbm.v23i.1373.

ZHOU, Z.H. (2021) " Ensemble learning ", ss: 181-210. **Springer Singapore**.

ZORİC A.B. (2016) "Predicting Customer Churn İn Banking İndustry Using Neural Networks", **Interdisciplinary Description of Complex Systems: INDECS**, 14(2), 116-124.

WEI X.; ve diğerleri (2023) "Risk Assessment of Cardiovascular Disease Based on SOLSSA-CatBoost Model", **Expert Systems with Applications**, 219, 119648.

WEİNBERGER, K.Q.; L.K. SAUL (2009) "Distance Metric Learning for Large Margin Nearest Neighbor Classification", **Journal of Machine Learning Research**, 10 (Feb), 207-244.

WELLS, J. D., W. L FUERST; J. CHOOBİNEH (1999) "Managing Information Technology (IT) for One-to-one Customer Interaction", **Information & Management**, 35(1), 53-62.https://doi.org/10.1016/S0378-7206(98)00076-7.

WU, H. (2022) "A High-Performance Customer Churn Prediction System based on Self-Attention", **arxiv preprint arxiv**:2206.01523.

QUİNLAN, J. R. (2014) "C4. 5: Programs for Machine Learning", **Elsevier**.

QUİNLAN, J. R. (1996) "Improved use of continuous attributes in C4. 5", **Journal of artificial intelligence research**, 4, 77-90.