T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yüksek Lisans Tezi

TELEKOMÜNİKASYON SEKTÖRÜNDE SINIFLANDIRMA ALGORİTMALARI ile MÜŞTERİ KAYIP ANALİZİ

Ezgi USTA

Enformatik Anabilim Dalı

Enformatik Programı

DANIŞMAN
Doç. Dr. Çiğdem EROL

Mayıs, 2021

İSTANBUL

Bu çalışma, 20.05.2021 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı, Enformatik Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Çiğdem EROL(Danışman)
İstanbul Üniversitesi
Fakülte

Prof. Dr. Sevinç GÜLSEÇEN
İstanbul Üniversitesi
Enformatik Bölümü

Doç. Dr. Tuncay ÖZCAN
İstanbul Teknik Üniversitesi
İşletme Mühendisliği

İntihal Programı Beyanı

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Proje Destekleri

Tezden Üretilmiş Yayınların Künye Bilgileri

ÖNSÖZ

Bilgi ve tecrübesi ile bana yol gösteren, yüksek lisans eğitimim boyunca sabır ve desteğini hiç eksik etmeyen çok değerli danışman hocam Doç. Dr. Çiğdem EROL'a sonsuz teşekkürlerimi sunarım.

Annem Elife USTA, babam Hayrullah USTA ve kardeşim Önder USTA'ya sevgileri, üzerimdeki emekleri ve destekleri için teşekkür ederim.

Yüksek Lisans eğitimim boyunca her daim yardım ve desteğini esirgemeyen, kendisinden çok şey öğrendiğim sevgili arkadaşım Burak ÖZDEMİR'e teşekkür ederim.

Mesleki deneyimi ve bilgisi ile gösterdiği destek ve emek için sevgili Süheyla ŞENGÜL VARDARLI'ya teşekkür ederim.

Mayıs 2021 Ezgi USTA

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

λ : Lambda

$\pi(x)$: Lojistik regresyon fonksiyonu

$p(x)$: Olasılık Fonksiyonu

P : Olasılık Değeri

Kısaltmalar **Açıklama**

API : Uygulama Programşama Ara Yüzü (*Application Programming Interface)*

AUC : Eğri Altında Kalan Alan (*Area Under Roc*)

CART : Sınıflandırma ve Regresyon Ağaçları (*Classification and Regression Trees*)

CLOB : Karakter Büyük Obje (*Character Large Object*)

CRM : Müşteri İlişkileri Yönetimi (*Customer Relationship Management*)

CSV : Virgülle Ayrılmış Değerler (*Comma-seperated Values*)

CPU : Merkezi İşlem Birimi (*Central Process Unit*)

DN : Doğru Negatif (*True Negative*)

DP : Doğru Pozitif (*True Positive*)

DAG : Yönlendirilmiş Döngüsüz Grafik (*Directed Acyclic Graph*)

ETL : Çıkar, Dönüştür ve Yükle (*Extract, Transform and Load*)

ELT : Çıkar,Yükle ve Dönüştür (*Extract, Load, Transform*)

FFTH : Eve Kadar Fiber (*Fiber to the home*)

FTTB : Binaya Kadar Fiber (*Fiber to the building*)

GB : Gigabyte

G/Ç : Giriş/Çıkış

GBM : Gradyan Artırma Makineleri (*Gradient Boosting Machines*)

FPR : Yanlış Pozitif Oranı (*False Positive Rate*)

HDFS : Hadoop Dağıtık Dosya Sistemi (*Hadoop Distributed File System*)

IEEE : Elektrik ve Elektronik Mühendisleri Enstitüsü (*The Institute of Electrical and Electronics Engineers*)

IPTV : İnternet Protokolü Televizyonu (*Internet Protocol Television*)

JAR : Java Arşivi (*Java Archieve*)

JSON : Java Script Nesne Notasyonu (*JavaScript Object Notation*)

Mbps : Saniyede Aktarılan Veri Sayısı (*Mega Bits Per Second*)

ML : : Makine Öğrenmesi (*Machine Learning*)

MLLIB **:** Makine Öğrenmesi Kütüphanesi (*Machine Learning Library*)

NoSQL : SQL ve Daha Fazlası (*Not Only SQL*)

PB : Petabyte

RDMBS : İlişkisel Veri Tabanı Yönetim Sistemi (*Relational Database Management System)*

RDDs : *Resilient Distributed Datasets*

RFID **:** Radyo Frekansı ile Tanımlama (*Radio-Frequency Idefication*)

ROC : Alıcı İşlem Karekteristiği (*Receiver Operating Characteristic*)

SCI : *Science Citation Index*

SMOTE : *Synthetic Minority Over-Sampling Technique*

SQL : Yapılandırılmış Sorgu Dili (*Structured Query Language*)

TB : Terabyte

TNR: : Gerçek Negatif Oranı (*True Negative Rate*)

TPR : Doğru Pozitif Oranı (*True Posivitve Rate*)

XGBOOST : Aşırı Gradyan Artırma (*Extreme Gradient Boosting)*

XML : Genişletilebilir İşaretleme Dili (*Extensible Markup Language*)

YARN : *Yet Anathor Resource Negotiator*

YN : Yanlış Negatif (*False Negative*)

YP : Yanlış Pozitif (*False Positive*)

ÖZET

YÜKSEK LİSANS TEZİ

TELEKOMÜNİKASYON SEKTÖRÜNDE SINIFLANDIRMA ALGORİTMALARI ile MÜŞTERİ KAYIP ANALİZİ

Ezgi USTA

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman : Doç. Dr. Çiğdem EROL

Dijital evren dünya çapında kullanıcılar tarafından üretilen büyük miktarda veri ile doludur. Çeşitli kaynaklardan gelen ve hacmi hızlı bir şekilde artan veri için farklı analiz ve depolama yöntemlerine ihtiyacı duyulmuştur ve bu noktada karşımıza 'Büyük Veri' kavramı çıkmaktadır. Şirketler veya kurumlar süreçlerine bağlı olarak gittikçe daha fazla veri depolamakta ve bu veriyi analiz ederek değer çıkarmaya çalışmaktadır. Günümüz iş dünyası gittikçe rekabetçi olmakta ve müşteri davranışlarını anlamak sürdürebilir rekabette önemli bir avantaj sağlamaktadır. Şirketlerin başarısı büyük ölçüde mevcut müşteri verisini ne kadar iyi analiz edebileceklerine ve anlamlı bilgiler çıkarabileceklerine bağlıdır. Bu kapsamda müşteri ayrılma eğilimini tahmin etmek ve bu eğilime neden olan durumları ortaya çıkarmak için 'Müşteri Kayıp Analizi' çalışmaları yapılmaktadır. Bu tez çalışmasının amacı büyük veri analizi teknolojisi ile telekomünikasyon sektöründe ayrılma eğilimi olan müşterileri tahmin eden modeli oluşturmaktır. Bu kapsamda bir telekomünikasyon şirketinden temin edilen veri kümesine Apache Spark'ın makine öğrenmesi kütüphanesi (MLlib) ile çeşitli sınıflandırma

yöntemleri uygulanmıştır. İlk olarak veri üzerinde ön hazırlık işlemleri yapılmış, sonrasında veri setine Lojistic Regresyon, Karar Ağaçları, Rastgele Orman, Gradyan Artırma Makineleri yöntemleri uygulanmıştır. Çalışma sonuçları incelendiğinde müşteri kaybı tahmininde en başarılı sonuçları %86,34 doğruluk %90,82 AUC ve %63,45 F-Ölçütü oranları ile Gradyan Artırma Makineleri yönteminin verdiği ve churn davranışını belirleyen en önemli niteliğin kontrat bitimine kalan süre olduğu görülmüştür.

Mayıs 2021, 87. sayfa.

Anahtar kelimeler: Büyük Veri, Apache Spark, Müşteri Kayıp Analizi, Makine Öğrenmesi

SUMMARY

M.Sc. THESIS

CUSTOMER CHURN ANALYSIS WITH CLASSIFICATION ALGORITHMS IN TELECOMMUNICATION SECTOR

Ezgi USTA

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor : Assist. Prof. Dr. Çiğdem EROL

The digital universe is filled with massive amounts of data generated by users around the world. Different analysis and storage methods are needed for data coming from various sources and whose volume is increasing rapidly, and at this point, the concept of 'Big Data' emerges. Companies or institutions are storing more and more data depending on their processes and trying to extract value by analyzing this data. Today's business world is increasingly competitive and understanding customer behavior provides a significant advantage in sustainable competition. The success of companies largely depends on how well they can analyze existing customer data and extract meaningful information. In this context, "Customer Churn Analysis" studies are carried out in order to predict the customer leaving tendency and to reveal the situations that cause this trend. The aim of this thesis is to create a model that predicts customers who tend to diverge in the telecommunications industry with big data analysis technology. In this context, Apache Spark's machine learning library (MLlib) and various classification methods were applied to the data set obtained from a telecommunication

company. Firstly, preliminary preparations were made on the data, and then Logistic Regression, Decision Trees, Random Forest, Gradient Boosting Machines methods were applied to the data set. When the results of the study were examined, it was seen that Gradient Boosting Machines method gave the most successful results in churn prediction with 86,34% accuracy, 90,82% AUC and 63,45% F-Score rates. It has been observed that the most important factor causing customer churn is remaining time to the end of the contract.

May 2021, .87. pages.

Keywords: Big data, Apache Spark, Churn Prediction, Machine Learning

1. GİRİŞ

Günümüzün iş ve teknoloji dünyasında veri vazgeçilmezdir. Veri kurumlar için her geçen gün daha değerli bir varlık haline gelirken veri analizinin etkin kullanımı da kurumların hayatta kalması ve sürdürülebilir rekabette avantajını korumalarını sağlamaktadır. Büyük veri faydalı hale getirildiğinde bir markanın veya işletmenin pazar koşullarını anlamasına, yeni müşteri kazanımına, müşterileri hakkında değerli bilgiler edinmesine ve bunun sonucu olarak müşteri ile etkileşimin arttırılmasına, dolayısıyla pazarlama stratejileri konusunda dönüşüm yapılmasına yardımcı olur.

Bir işletmenin varlığını devam ettirebilmesi yeni pazar fırsatlarını bulması ve müşterileri ile aralarındaki ilişkiye bağlıdır. Buna göre; müşteri davranışlarının takip edilmesi ve bu davranışlardaki değişikliklere göre hareket edilmesi oldukça önemlidir. Bu kapsamda var olan müşterilerin elde tutulması amacıyla "Müşteri Kayıp Analizi" çalışmaları yapılmaktadır. Müşteri kayıp analizi çalışmaları ile beraber aboneliklerini sonlandırma eğilimi gösteren müşteriler ve bu eğilime neden olan durumlar ortaya çıkartılmaya çalışılmaktadır (Koçoğlu, 2017).

İnternet ve internet teknolojilerinin hayatımızın her alanına girmesi ile birlikte insanlar günlük aktivitelerini gerçekleştirirken bile veri üretir duruma gelmiştir. Üretilen veri türü, sadece geleneksel veri tabanlarında tutulan veri değil, sosyal medya içerikleri, e-postalar, görüntü, video, ses gibi farklı türleri içeren veridir. Arama motorlarında yapılan aramalar, sosyal medyada paylaşılan gönderiler, e-ticaret siteleri üzerinden satın alınan ürünler, akıllı telefonlar aracılığıyla yapılan işlemler ile kullanıcılar arkalarında veri bırakmaktadırlar. Veri hacmindeki artış ve dijital ekonominin hızla yükselişi veri depolama ve veri analizi ihtiyacı arttırmıştır. Klasik veri tabanı sistemleri ile verinin saklanması ve veri analizi yöntemleri hızla artan veri hacmi karşısında yetersiz kalmıştır ve bu noktada 'Büyük Veri' kavramı doğmuştur.

Büyük Veri sadece internet ve internet teknolojilerinde değil, gen dizilişlerinin analizi, hava durumu sensörlerinden gelen veri başta olmak üzere verileme işlemlerinin yapıldığı birçok alanda büyük veri bir ihtiyaç olarak karşımıza çıkmaktadır.

Büyük veriyi avantaja dönüştürme aşamasında karşımıza büyük veri analizi çıkar. Büyük veri analizindeki temel yaklaşım, üretilen verinin herhangi bir veri depolama sisteminde saklanıp sonrasında bu veriden çıkarımlar yapılarak değer elde edilmesi yönündedir. Bu işlem için birtakım teknolojiler sunulmuştur. Apache Spark bu teknolojilerden biridir. Hızı, ölçeklenebilirliği, ileri analitik işlemleri ve kullanım kolaylığı sebebi ile veriden bilgi ve ön görü çıkarmayı hedefleyen veri bilimi alanında gittikçe popüler olmuştur. Java, Scala, R, Python gibi programlama dilleri ile uygulama geliştirme fırsatı sunmasının yanı sıra, çok amaçlı bir dizi kütüphane sunmaktadır. Makine öğrenmesi için MLlib, akış işleme için Spark Streaming, grafik işleme için GraphX gibi ileri analitik ihtiyacını karşılayacak ve birbiri ile yakın çalışabilecek şekilde tasarlanmış bir dizi bileşen içerir. Bu tasarımın en önemli avantajı, farklı işleme ve analiz yöntemleri gerektiren uygulamaları sorunsuz bir şekilde birleştirebilme yeteneğidir. Örneğin sosyal medyadan gerçek zamanlı olarak akan veriye makine öğrenmesi kütüphanesi ile sınıflandırma yöntemleri uygulanabilir.

Bu tezde büyük veri platformlarından biri olan Apache Spark ile telekomünikasyonda müşteri kayıp analizi gerçekleştirilmiştir. Analiz aracı olarak Apache Spark'ın seçilmesini sebebi, daha önce Apache Spark ile yapılmış müşteri kayıp analizi tez çalışması yapılmamış olmasıdır. Bu tez çalışmasının GENEL KISIMLAR ana başlığı altında müşteri kayıp analizi tanımı ve detayları belirtilmiş, büyük veri tanımı, tarihçe ve büyük veri işleme araçlarına yer verilmiş, seçilen teknoloji olarak Apache Spark tanıtılmış ve literatürde Apache Spark ile yapılmış büyük veri çalışmaları incelenmiştir. Ayrıca çalışmada kullanılan algoritmaların detaylarına yer verilmiştir.

MALZEME ve YÖNTEM bölümünde tez çalışmasında kullanılan veri setinin detaylarına, kullanılan programlama araçlarına, veri seti üzerinde yapılan işlemlere ve çalışmayı gerçekleştirmek için veri setine uygulanan yöntemlere yer verilmiştir.

BULGULAR bölümünde geliştirilen modellerden elde edilen sonuçlar ele alınmış, ardından çeşitli performans değerlendirme ölçütlerine göre yöntemlerin performans analizlerine yer verilmiştir ve en başarılı model belirlenmiştir.

TARTIŞMA ve SONUÇ bölümünde tez çalışmasının diğer çalışmalardan farkı, bulguların yorumları, elde edilen sonuçların literatüre katkısı, gelecek çalışmaların neler olabileceği konularına yer verilmiştir.

2. GENEL KISIMLAR

2.1. MÜŞTERİ KAYIP ANALİZİ

Yeni pazarlar geliştikçe, şirketler arasındaki rekabet keskin bir şekilde artmaktadır. Rekabet zorlaştığı ve telekomünikasyon satış ürünü haline geldiği için şirketler maliyetleri en aza indirmek, hizmetlerine değer katmak ve farklılığı garanti etmek için yarışırlar. Artık müşteriler hizmet sağlayıcılarını seçebiliyor durumda olduğundan şirketler pazardaki durumlarını korumak için müşteri ilişkilerine dikkat etmektedir. Zorlu rekabet koşulları altında şirketler, müşterilerin davranışlarına odaklanmaya çalışırlar (Kisioglu ve Topcu, 2011). Müşterilerin davranışları ve aldıkları aksiyonlar organizasyonlar için önemli ipuçlarıdır. Yeni müşteri kazanımının yanında şirketler, var olan müşterileri de kaybetmemek durumundadır.

Müşteri elde tutma, yeni ve potansiyel olarak riskli müşterileri arama ihtiyacını azaltarak, mevcut müşterilerin taleplerine odaklanmaya olanak tanıması ve uzun vadeli müşterilerin daha fazla satın alma eğiliminde olması sebebi ile ekonomik değere sahiptir. Müşteri memnuniyeti yüksek müşterilerden gelen olumlu sözler, yeni müşterilerin kazanılması için iyi bir yoldur ve uzun vadeli müşterilerin taleplerine ilişkin daha geniş bir veri tabanı olması nedeniyle mevcut müşterilere hizmet vermek daha az maliyetlidir (Van den Poel ve Lariviere 2004).

Müşteri ilişkileri yönetimi (CRM) araçları, müşteri kazanımını ve müşteri elde tutmayı geliştirmek, karlılığı artırmak ve tahmine dayalı modelleme ve sınıflandırma gibi önemli analitik görevleri desteklemek için geliştirilmiştir. Tipik bir CRM uygulaması her bir müşteriyle ilgili çok sayıda bilgi içerir. Bu bilgi, müşterilerin şirketteki faaliyetlerinden, kayıt sürecinde müşteri tarafından girilen veriden, çağrı merkezine yapılan aramalardan vb. elde edilir. Bu verinin doğru analizi, pazarlama amaçları için değil, aynı zamanda sözleşmelerini feshetme olasılığı olan müşterileri belirlemek için de dikkate değer sonuçlar sağlayabilir (Lazarov ve Capota, 2007). Bu amaç doğrultusunda veri madenciliğinin uygulama alanlarından biri olan Müşteri Kayıp Analizi çalışmaları karşımıza çıkmaktadır. Müşteri kayıp analizi çeşitli sektörlerde yapılmaktadır. Müşteri kaybını tahmin etmede birçok veri madenciliği teknikleri kullanılmaktadır. Makine öğrenmesi, büyük veri analizi teknikleri, meta-sezgisel teknikler ve hibrit modeller müşteri kaybını tahmin etmede yaygın olarak kullanılan yöntemler arasındadır.

"*Churn*", İngilizce değişim anlamına gelen *change* ve dönüş anlamına gelen *turn* kelimelerinden türetilen bir kelimedir. Bir sözleşmenin sona ermesi anlamına gelir (Lazarov ve Capota, 2007).

Müşteriler aldıkları hizmeti kullanmayı bırakıp başka bir hizmet sağlayıcıya geçmesi durumunda "*churner*" olarak adlandırılır. Bu durum, rakiplere kolayca geçiş yapılabilen ve birçok müşterisi olan bankalar, sigorta ve telekomünikasyon şirketleri gibi şirketler için büyük bir endişe kaynağıdır (Richeldi ve Perrucci, 2002).

Lazarov ve Capota (2007) müşteri kaybını 3 sınıfa ayırmıştır. Aktif müşteri kaybı, müşterinin mevcut sözleşmesini iptal edip başka bir hizmet sağlayıcıya geçmesini; dönüşümlü müşteri kaybı, müşterinin rakip sağlayıcıya geçme amacı olmadan sözleşmesini iptal etmesini; pasif müşteri kaybı ise hizmet sağlayıcı tarafından sözleşmenin iptal edilmesini ifade eder. En zoru gönüllü (aktif) müşteri kaybını tahmin etmektir. Aktif müşteri kaybını önlemek için, düşük hata olasılığı ile ayrılma eğiliminde olan müşterilerin hangi müşteriler olduğunun tahmin edilmesi ve bu müşterilerin neden bir rakibin yararına şirketten ayrılmaya karar verdiğinin bilinmesi gerekir (Lazarov ve Copata, 2007).

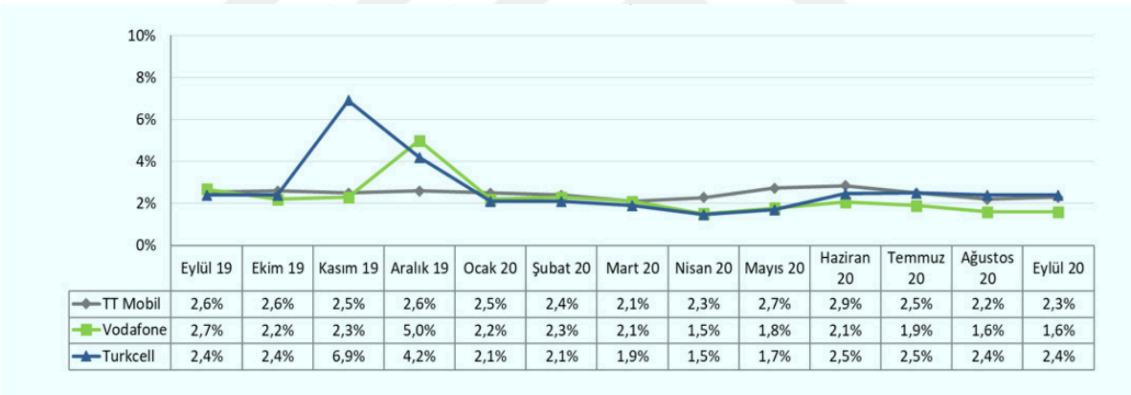
2.1.1. Telekomünikasyonda Müşteri Kayıp Analizi ve Önemi

Berson ve diğerleri (2000) müşteri kaybını, kablosuz telekomünikasyon hizmet sektöründe, bir sağlayıcıdan diğerine müşteri hareketini belirtmek için kullanılan bir terim olduğunu ve "müşteri kaybı yönetimini", bir operatörün kârlı müşterileri elde tutma sürecini tanımlayan bir terim olduğunu belirtmiştir. Benzer şekilde, Kentrias (2001), telekomünikasyon hizmetleri endüstrisindeki müşteri kaybı yönetimi teriminin bir şirket için en önemli müşterileri güvence altına alma prosedürünü tanımlamak için kullanıldığını düşünmektedir. Temel olarak, uygun bir müşteri yönetimi, müşterinin bir hizmet sağlayıcıdan diğerine geçme kararını, 'müşteri karlılığının' bir ölçümünü ve hareketi azaltmak için bir dizi stratejik elde tutma önlemini tahmin etme yeteneğini varsayar (Hung ve diğ., 2006).

Telekom sektöründe hem ses hem de veri hizmeti için müşteriler servis sağlayıcı olarak çeşitli şirketlerden hizmet alabilir ve hizmetlerini bir sağlayıcıdan diğerine taşıyabilir. Sektörde rekabetin yüksek olması ve sürekli yenilenen ürünler nedeniyle, müşteriler özel ürünleri ve daha iyi hizmeti çok daha düşük fiyatlarla talep etmektedirler. Birçok telekomünikasyon şirketi, müşterileri daha uzun süre hizmette tutmak için elde tutma stratejileri uygulamaktadır. Müşteri

kaybının azaltılması şirketler için bir numaralı iş hedefi haline gelmiştir. Telekomünikasyon gibi abone yönetimi yapan ve gelirlerini abonelerinden sağlayan sektörlerde abone sayısını arttırmak için ana strateji yeni abone kazanımını arttırmak ile başlar. Ancak kazanılmış abonelerin kontrolsüz bir şekilde ayrılması hem şirketin uzun ve kısa vadede elde edeceği gelir planlamasını yapamamasına sebep olur, hem de abone kazanmak için katlanmış olduğu iletişim ve altyapı maliyetleri için bir tehdit oluşturur. Finansal sonuçlarını etkileyen abone hareketlerinde önemli olan nokta, ne kadar satış yapılmış olduğu ve ne kadar net abone sahibi olduğundan geçmektedir. Net abone kazanımını maksimize edebilmek için etkin bir müşteri elde tutma yönetimi vazgeçilmez bir stratejidir. Aynı zamanda uzun süre abonelik hayatına devam eden müşteri kitlesi, şirketin faaliyetinde bulunan farklı ürünlerin çapraz satışları için de firmaya fırsat ve zaman sağlamaktadır.

Şekil 2.1'de Türkiye'deki mobil operatörlerin Eylül 2019 – Eylül 2020 dönemlerini kapsayan müşteri kayıp oranı verilmiştir.

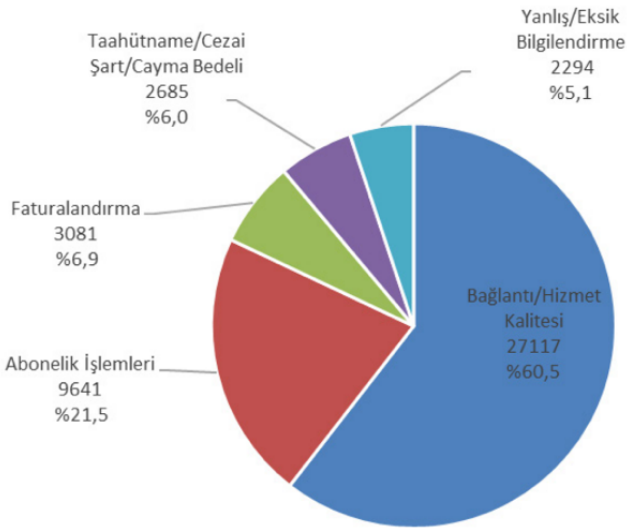


	Eylül 19	Ekim 19	Kasım 19	Aralık 19	Ocak 20	Şubat 20	Mart 20	Nisan 20	Mayıs 20	Haziran 20	Temmuz 20	Ağustos 20	Eylül 20
TT Mobil	2,6%	2,6%	2,5%	2,6%	2,5%	2,4%	2,1%	2,3%	2,7%	2,9%	2,5%	2,2%	2,3%
Vodafone	2,7%	2,2%	2,3%	5,0%	2,2%	2,3%	2,1%	1,5%	1,8%	2,1%	1,9%	1,6%	1,6%
Turkcell	2,4%	2,4%	6,9%	4,2%	2,1%	2,1%	1,9%	1,5%	1,7%	2,5%	2,5%	2,4%	2,4%

Şekil 2.1: Mobil işletmecilerin müşteri kayıp oranı (BTK, 2020).

Telekomünikasyon sektörü, artan abone sayısı, hızla yenilenen ürünler ve artan büyüklükte ses ve data kullanımı nedeniyle büyük miktarda veri üretmektedir. Veri madenciliği için büyük veri analizi teknolojisinin kullanılması, müşteri kaybını tahmin etmek ve müşteri kaybını azaltmak için telekomünikasyon sektöründe araştırmaların odak noktası haline gelmiştir. Sektörde işletmeciler arasındaki rekabetin büyük olması sebebi ile verinin zamanında ve hızlı şekilde analiz edilmesi çok önemlidir. Bu nedenle müşteri ayrılma analizi ve ayrılmaya sebep olan durumların tespit edilerek müşteri deneyimini iyileştirme çalışmaları yaygın olarak yapılmaktadır. Telekomünikasyon şirketlerinin müşteri kaybını azaltmaları için yüksek riskli müşterileri belirlemesi gerekir. Buna ek olarak belirlenen müşteri kitlesinin ihtiyaçlarının

karşılanabilmesi ve müşterilerin operatörde kalmaya ikna edilebilmesi için müşterilerin ne zaman ayrılacağının da tahmin edilmesi önem taşır. Şekil 2.2'de internet servis sağlayıcıları için konularına göre abone şikayetleri arasında en yüksek paya bağlantı/hizmet kalitesi ve abonelik işlemlerinin sahip olduğu görülmektedir (BTK, 2020). En çok abone şikayetine konu hususlar; bağlantı/hizmet kalitesi (n=27117, %60,5), abonelik işlemleri (n=9641, %21,5), faturalandırma (n=3081, %6,9), taahhütname/cezai şart/cayma bedeli (n=2685, %6,0), yanlış/eksik bilgilendirmedir (n=2294, %5,1). Müşterinin aldığı hizmetten memnun olmaması ayrılma eğilimine neden olan etkenlerdendir. Bu tür abonelerin tespit edilip hizmet kalitesinin arttırılması oldukça önemlidir. Müşteri kayıp analizi çalışmaları ile ayrılma eğilimine neden olan durumların ortaya çıkarılması, müşteri elde tutmayı sağlamasının yanında yeni abone kazanımı da arttırabilir. İnsanlar olumlu hizmet deneyiminden daha çok, olumsuz hizmet deneyimini paylaşma eğilimindedir ve bu da şirketin gelecekteki olası müşterileri arasında olumsuz bir imaja sahip olmasına neden olur. Memnuniyeti yüksek olan müşterilerden gelen olumlu sözler yeni müşterilerin kazanılması için iyi bir yöntem olabilir.



Şekil 2.2: İnternet servis sağlayıcıları için müşteri şikayetlerinin konularına göre dağılımı (BTK, 2020).

Yeni müşteri kazanımı, mevcut müşterileri elde tutmaktan çok daha pahalıya mal olmaktadır. Genel olarak şirket müşteri kazanımı maliyetlerini tamamen telafi etmeden önce yeni müşteri ayrılmış olacaktır, bu nedenle mevcut müşterileri elde tutmaya yönelik harcamalar yapmak, yeni müşteriler kazanmaktan daha verimlidir. Sonuç olarak, müşteri kayıp yönetimi çok önemli bir rekabet silahı ve müşteri odaklı pazarlama çabalarının tamamı için bir temel olarak ortaya

çıkmıştır. Etkin bir müşteri elde tutma yönetimi ile bir şirket ne tür müşterilerin ayrılma olasılığının ve hangi müşterilerin sadık olma olasılığının yüksek olduğunu belirleyebilir. Sürecin bir kısmı müşteri değerini belirlemektir, bazen karlı olmayan veya çok az karlı olan müşterilerin serbest bırakılması istenebilir. Bu tür bilgiler şirket için mevcut olduğunda, pazarlama yöneticileri ayrılan müşterileri en aza indirmek, değerli olanları geri kazanmak ve gelecekte ayrılma ihtimali en az olanlar da dahil olmak üzere doğru müşterileri daha uygun maliyetle çekmek için bilinçli ve stratejik önlemler alabilirler (Richeldi ve Perrucci, 2002).

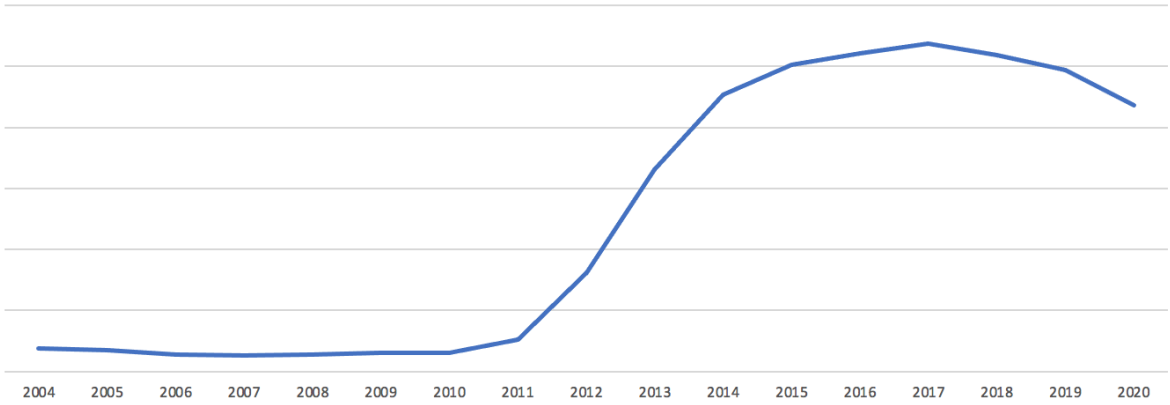
Genel olarak, telekomünikasyon endüstrisinde müşteri kaybını tahmin etmek için kullanılan özellikler arasında müşteri demografik bilgileri, sözleşmeye bağlı veri, müşteri hizmetleri günlükleri, müşteri hesap bilgileri, çağrı ayrıntıları, şikayet kayıtları, fatura ve ödeme bilgileri, müşterilerin sahip olduğu faydalar, arama detayları, çağrı alma detayları bulunur (Huang ve diğ., 2012).

2.2. BÜYÜK VERİ VE ANALİZİ

2.2.1. Büyük Veri

Büyük veri kavramı ilk kez 1998 yılında Silicon Graphics International çalışanı John Mashey tarafından "*Big Data and the NextWare of InfraStress*" başlıklı sunumda kullanılmıştır (Diebold, 2012). Büyük veri kavramının bir Veri Madenciliği kitabında yer alması ise ilk kez Weiss ve Indrukya'nın "*Predictive Data Mining: A Practical Guide*" kitabı ile gerçekleşmiştir (Weiss ve Indurkhya, 1998). İlk akademik makalede kullanılması 2000 yılında Francis X. Diebold'un " "*Big Data" Dynamic Factor Models for Macroeconomic Measurement and Forecasting"* isimli makalesi ile olmuştur (Diebold, 2000). 9

90'lı yılların ortalarına yapılan referanslara rağmen Şekil 2.3'de görüldüğü gibi, büyük veriye olan ilgi 2011 yılında artmaya başlamıştır.



Şekil 2.3: Büyük veriye olan ilginin yıllara göre değişimi (Google Trends-Big Data, 2020).

2010 yılında Apache Hadoop büyük veriyi; kabul edilebilir bir kapsamda genel bilgisayarlar tarafından yakalanamayan, yönetilemeyen ve işlenemeyen veri kümeleri olarak tanımlamıştır (Chen ve diğ., 2014). Manyika ve diğerleri (2011), büyük veriyi; boyutu, tipik veri tabanı yazılım araçlarının yakalama, saklama, yönetme ve analiz etme yeteneğinin ötesinde olan veri kümesi olarak tanımlar. IDC, 2011 yılında yayınladığı raporunda büyük veriyi diğerlerinden daha geniş kapsamlı olarak ele alır ve büyük veriyi yüksek hızda yakalama, keşif ve analiz sağlayarak, çok çeşitli veriden çok büyük miktarlarda ekonomik olarak değer elde etmek için tasarlanmış yeni nesil teknoloji ve mimariler olarak tanımlar (Gantz ve Reinsel, 2011).

Büyük Veri Bileşenleri:

Veri Büyüklüğü, büyük veri nedir sorusu göz önüne alındığında akla gelen ilk özelliktir. Ancak, çok çeşitli kaynaklardan gelen veri düşünüldüğünde veri miktarının yanında başka özellikler de ortaya çıkmıştır.

Laney (2001), bilgi teknolojileri organizasyonlarının veri yönetimine geleneksel yaklaşımlarla bakmamaları gerektiğini belirtmiş ve 3-V olarak bilinen Hacim (*Volume*), Hız (*Velocity*) ve Çeşitlilik (*Variety*) bileşenlerinin veri yönetiminde karşılaşılan zorluklar olduğunu belirtmiştir. 3-V, büyük veriyi tanımlamak için ortak bir çerçevedir (Chen ve diğ., 2012; Kwon ve diğ., 2014; Gandomi ve Haider ,2015). 3-V büyük veriyi tanımlamak ve diğer büyük ölçekli veri kümelerinden ayırmada kullanılsa dahi, büyük veriyi tanımlayan iki önemli bileşen daha vardır:

değer (*Value*) ve doğruluk (*Veracity*) (Lycett, 2013, Ebner ve diğ., 2014; Erevelles ve diğ., 2016). Bu özellikler Şekil 2.4'de görüldüğü üzere büyük verinin 5V'si olarak anılmaktadır.

Bazı çalışmalarda ise değişkenlik (*Variability*) ve karmaşıklık (*Complexity*) büyük verinin diğer boyutları olarak incelenmiştir (Gandomi ve Haider, 2015).

Hacim: Veri hacmi büyük verinin birincil niteliğidir. Büyük verinin üretilmesi ve toplanmasıyla, veri ölçeğinin giderek daha büyük hale gelmesini ifade eder. Büyük veri hacimlerinin tanımları görecelidir, zaman ve veri türü gibi faktörlere göre değişir. Günümüzde büyük veri olarak kabul edilebilecek veri miktarı, gelecekte daha fazla veri kümesinin yakalanmasına izin verecek şekilde depolama kapasitelerinin artması ile eşiği karşılamayabilir. Ayrıca, çeşitlilik altında tartışılan veri türü, 'büyük veri' ile neyin kastedildiğini tanımlar. Aynı boyuttaki iki veri kümesi, türlerine bağlı olarak farklı veri yönetimi teknolojileri gerektirebilir. Örneğin: Tablo ve video verisi. Dolayısıyla, büyük verinin tanımları da sektöre bağlıdır. Bu nedenle, bu hususlar büyük veri hacimleri için belirli bir eşik tanımlamayı imkânsız kılmaktadır (Gandomi ve Haider, 2015).

Hız: Veri üretilme hızını temsil eder. Büyük veri yüksek hızlı olarak nitelendirilir (Elragal, 2014). Veriden anlamlı bilgi çıkarmak için verinin ne kadar hızlı analiz edilmesi gerektiğinin bir göstergesidir (López ve diğ.,2014). Mobil cihazlardan yayılan ve mobil uygulamalardan akan veri, günlük müşteriler için gerçek zamanlı, kişiselleştirilmiş teklifler oluşturmak için kullanılabilecek meta veri üretir. Bu veri, gerçek müşteri değeri oluşturmak için gerçek zamanlı olarak analiz edilebilen coğrafi konum, demografi ve geçmiş satın alma modelleri gibi müşteriler hakkında sağlam bilgiler sağlar (Gandomi ve Haider, 2015).

Çeşitlilik: Veri kümesinde ki yapısal çeşitliliği ifade eder. Veri farklı türlerde yapısal, yarı yapısal ve yapısal olmayan veri olabilir (Gandomi ve Haider, 2015).

- Yapısal veri, verinin tanımlanmış iş kurallarına göre tablolar halinde düzenlendiği geleneksel veri tabanlarında (SQL veya diğerleri) bulunur. Yapılandırılmış veri genellikle çalışılması en kolay veri tipidir, çünkü veri tanımlanmış ve dizinlenmiştir, erişimi ve filtrelenmesi kolaydır (Ohlhorst, 2013).
- Yapısal olmayan veri tablolar halinde düzenlenmemiş ve uygulamalar tarafından yerel olarak kullanılmamış veya bir veri tabanı tarafından yorumlanmamış veridir. (Ohlhorst,

2013). Bu türdeki verinin sunulmasının zor olması, veri işleme süreçlerinde NoSQL (Not only SQL) gibi ilişkisel olmayan ve yatay mimariye sahip yeni teknolojilerin ortaya çıkmasına neden olmuştur. Radar ve sensör verisi, görüntü, ses, video verisi, e-posta ve metinler yapısal olmayan veriye örnektir.

- Yarı yapısal veri, yapılandırılmamış ve yapılandırılmış verisi arasındadır. Yarı yapılandırılmış verinin, tablo ve ilişkileri olan bir veri tabanı gibi resmi bir yapısı yoktur. Ancak, yapılandırılmamış verinin aksine, yarı yapılandırılmış veri, anlamsal öğeleri ayırmak ve içindeki kayıt ve alan hiyerarşilerini sağlamak için etiketler veya başka işaretler içeren veri biçimidir (Ohlhorst, 2013). Bu veri türüne JSON ve XML türündeki veri örnek verilebilir.

Değer: Sürekli artan veri miktarı değer sorununa yol açmaktadır. Önemsiz ve alakasız veriyi ortadan kaldırmak, iç görü elde edilecek veriyi ortaya çıkarmaktadır. Buradaki sorun hangi verinin alakalı olduğunu tanımlamak ve daha sonra bu veriyi zamanında analiz için hızlı bir şekilde ayıklamaktır (Erevelles ve diğ., 2016).

Doğruluk: IBM, Doğruluğu bazı veri kaynaklarında bulunan güvensizliği temsil eden dördüncü V olarak önermiştir (Gandomi ve Haider, 2015). Veri belirsizliği ve belirli veri türleriyle ilişkili güvenilirlik düzeyini ifade eder. Yüksek veri kalitesine ulaşmak için çabalamak önemli bir büyük veri gereksinimi ve zorluktur, ancak en iyi veri temizleme yöntemleri bile hava durumu, ekonomi veya bir müşterinin gelecekteki fiili satın alma kararları gibi bazı veri kümelerinin doğal öngörülemezliğini ortadan kaldıramaz. Belirsizliği kabul etme ve planlama ihtiyacı büyük verinin bir başka boyutudur. Belirsizliği ele almanın ve yönetmenin önemini vurgular (Miele ve Shockley, 2013).

Şekil 2.4: Büyük verinin 5-V'si (Demchenko ve diğerleri, 2013).

Değişkenlik: Değişkenlik, veri akış hızındaki değişimi ifade eder ve SAS (Statistical Analysis System) tarafından ortaya atılmıştır. Artan veri hızlarına ve çeşitliliğine ek olarak, veri akışları öngörülemez şekilde sık sık değişmektedir ve büyük ölçüde başkalaşıma uğramaktadır. Bu büyük veriyi yönetmek açısından zorlayıcı bir durumdur, ancak işletmelerin sosyal medyada ne zaman trend olduklarını ve günlük, sezonluk ve olaylarla tetiklenen en yüksek veri yüklerini nasıl yöneteceklerini bilmeleri gerekir (SAS, 2020).

Karmaşıklık: Karmaşıklık, büyük verinin sayısız kaynak aracılığıyla üretildiği gerçeğini ifade eder. Farklı kaynaklardan alınan veriyi bağlama, eşleştirme, temizleme ve dönüştürme ihtiyacı gibi kritik zorlukları ortaya çıkarır (Gandomi ve Haider, 2015).

Büyük Verinin Sınıflandırılması:

Büyük veri birçok şekilde depolanabilir, elde edilebilir, işlenebilir ve analiz edilebilir. Her büyük veri kaynağının, verinin frekansı, hacmi, hızı, türü ve doğruluğu gibi farklı özellikleri vardır. Büyük veri işlendiğinde ve depolandığında, yönetişim, güvenlik ve yönetim politikaları gibi ek boyutlar devreye girer. Uygun bir büyük veri mimarisini seçmek ve uygun çözümü oluşturmak, pek çok faktörün dikkate alınması gerektiği için oldukça zordur. Bu nedenle büyük veri, özelliklerinin daha iyi anlaşılabilmesi için farklı kategoriler halinde sınıflandırılabilir. Büyük veriyi türe göre kategorize etmek, her tür verinin özelliklerini görmeyi kolaylaştırır. Bu

özellikler, verinin nasıl elde edildiğini, uygun formatta nasıl işlendiğini ve yeni verinin ne sıklıkla kullanılabilir hale geldiğini anlamamıza yardımcı olur. Farklı kaynaklardan gelen verinin farklı özellikleri vardır; örneğin, sosyal medya verisi sürekli olarak gelen video, resim ve blog gönderileri gibi yapılandırılmamış metinlere sahip olabilir. Verinin nasıl toplandığı, analiz edildiği ve işlendiği gibi, büyük verinin özelliklerine belirli hatlardan bakmak yararlıdır. Veri sınıflandırıldıktan sonra uygun büyük veri modeliyle eşleştirilebilir (Mysore ve diğ., 2013). Tablo 2.1'de belirtildiği üzere on iki farklı kategoriye ayrılır (Mysore ve diğ., 2013; Hashem ve diğ., 2015; Terzi ve diğ., 2015; Eyüpoğlu, 2018). Bu kategorilerin her birinin kendine has özellikleri vardır.

Tablo 2.1: Büyük verinin sınıflandırılması (Mysore ve diğ., 2013; Hashem ve diğ., 2015; Terzi ve diğ., 2015).

Kategori	Açıklama
Veri Tipi	Meta veri, ana veri, tarihsel veri, işlem verisi
İçerik Formatı	Yapısal, yarı yapısal, yapısal olmayan
Veri Kaynağı	Web ve Sosyal medya, IoT, iç veri kaynakları, işlem verisi, insan
Veri kullanımı	Sanayi, akademi, araştırma merkezleri, devlet
Veri Tüketicisi	Veri Depoları, insani, iş süreci, kurumsal uygulamalar
Analiz Tipi	Gerçek zamanlı, yığın, etkileşimli, hibrit
İşleme Metodolojisi	Tahmini analiz, analitik, modelleme, sorgu ve raporlama
İşleme Yöntemi	Yüksek performanslı hesaplama, dağıtık, paralel, küme, grid
Veri deposu	İlişkisel, graf, belge odaklı, sütun odaklı, anahtar değeri
Veri Sıklığı	İsteğe bağlı, sürekli, gerçek zamanlı, zaman serisi
Veri Hazırlama	Temizleme, normalizasyon, dönüşüm
Donanım	Ticari donanım, son teknoloji donanım

1. **Veri Tipi:** İşlenecek verinin türüdür. Veri türünü bilmek, depodaki veriyi ayırmaya yardımcı olur. İşlemsek, geçmiş, ana veri ve diğerleri.
2. **İçerik Formatı:** Gelen veriyi nasıl işlenmesi gerektiğini belirler, araç ve tekniklerin seçilmesi ve bir çözümün işletme perspektifinden tanımlanması için kilit önem taşır. Yapılandırılmış (örneğin RDMBS), yapılandırılmamış (örneğin ses, video ve görüntüler) veya yarı yapılandırılmış biçimde olabilir
3. **Veri Kaynağı:** Veri kaynağı veya verinin üretildiği yer. Tüm veri kaynaklarını belirlemek, kapsamı bir işletme perspektifinden belirlemeye yardımcı olur. Veri

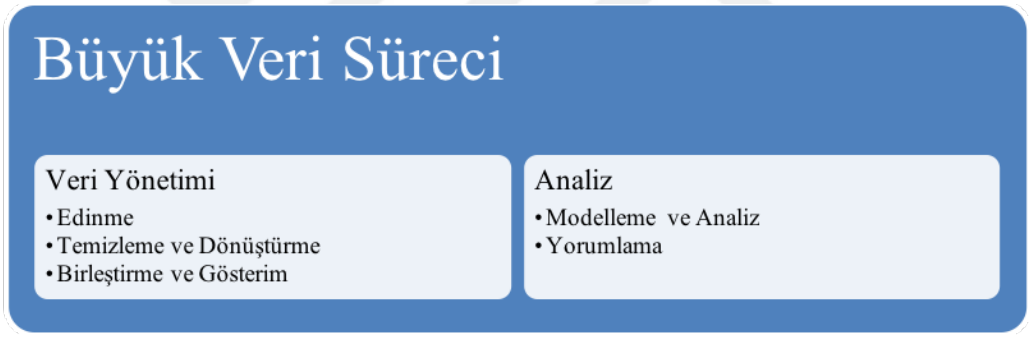
kaynakları Web ve Sosyal Medya (Facebook, Twitter, Bloglar) siteleri, makine ve insan tarafından üretilen veri, sensörler, işlem verisi, IoT olabilir.

4. **Veri Kullanımı:** Veriyi kullanacak olan kişiler veya kurumlar. Sanayi, Akademi, Devlet Kurumları, araştırma merkezleri, sivil toplum kuruluşları
5. **Veri Tüketicisi:** İşlenmiş verinin olası tüketicileri. İnsan, iş süreçleri, kurumsal uygulamalar, veri depoları.
6. **Analiz Tipi:** Veri gerçek zamanlı, yığın, etkileşimli veya hibrit olarak analiz edilebilir. Analiz tipi; ürünler, araçlar, donanım, veri kaynakları ve beklenen veri frekansı hakkındaki diğer bazı kararları etkilediğinden dikkatli seçilmelidir. Dolandırıcılık tespiti için kullanılacak analiz tipi gerçek zamanlı veya gerçek zamanlıya yakın olmalıdır. Stratejik iş kararları için yapılan trend analizi ise yığın olarak yapılabilir.
7. **İşleme Metodolojisi:** Verinin işlenmesi için kullanılacak teknik türü. İş gereksinimleri uygun işleme metodolojisini belirler. Tekniklerin bir kombinasyonu kullanılabilir. İşleme metodolojisi seçimi, büyük veri çözümünüzde kullanılacak uygun araç ve tekniklerin belirlenmesine yardımcı olur. Ön görücü analiz, analitik (metin analizi, konuşma analizi vb.), modelleme, raporlama.
8. **İşleme Yöntemi:** İşleme yöntemi olarak yüksek performanslı hesaplama, dağıtık, paralel, küme ve grid yapısı kullanılabilir. Kullanılacak büyük veri işleme aracı işleme yöntemine göre belirlenir.
9. **Veri Deposu:** Verinin bulunduğu depolar. İlişkisel, graf, belge odaklı, sütun odaklı, anahtar değeri (Anahtar değeri, çok büyük bir boyuta ölçeklenmek üzere tasarlanmış veriyi depolayan ve bunlara erişen alternatif bir ilişkisel veri tabanı sistemidir)
10. **Veri Sıklığı:** Ne kadar veri ne sıklıkta gelir? Veri geliş sıklığı ve boyutunu bilmek, depolama mekanizmasını, depolama biçimini ve gerekli önişleme araçlarını belirlemeye yardımcı olur. Veri sıklığı ve boyutu veri kaynaklarına bağlıdır: Sosyal Medya verisi isteğe bağlı, hava durumu ve işlem verisi sürekli veya gerçek zamanlı, zamana dayalı verisi zaman serileri olarak gelir.
11. **Veri Hazırlama:** Veriye yapılacak olan temizleme, entegrasyon, dönüşüm, normalizasyon, veri azaltma, ayrıklaştırma gibi veri ön işleme adımlarını ifade eder.
12. **Donanım:** Büyük veri çözümünün uygulanacağı donanım türü. Donanımın sınırlamalarını anlamak, büyük veri çözümü seçimine yardımcı olur.

2.2.2. Büyük Veri Analizi

Veri kümelerinin boyutları sürekli olarak artmaktadır, şu anda tek bir veri setinde hacim olarak terabayttan (TB) petabayt (PB) seviyesine kadar veri bulunmaktadır. Büyük veri analizi, büyük veri kümelerine gelişmiş analiz tekniklerin uygulandığı kısımdır, ancak veri seti ne kadar büyük olursa, analizi ve yönetimi o kadar zorlaşır.

Büyük veriden iç görü çıkarmanın genel süreci, Şekil 2.5'te gösterilen beş aşamaya (Labrinidis ve Jagadish, 2012) ayrılabilir. Bu beş aşama, iki ana alt süreci oluşturur: veri yönetimi ve veri analizi. Veri yönetimi, veri elde etmek ve depolamak, analiz için hazırlamak ve almak için süreçleri ve destekleyici teknolojileri içerir. Öte yandan veri analizi, büyük veriyi analiz etmek ve iç görü elde etmek için kullanılan teknikleri ifade eder. Bu nedenle, büyük veri analitiği, büyük veriden genel olarak 'içgörü çıkarma' sürecinde bir alt süreç olarak görülebilir (Gandomi ve Haider, 2015).



Şekil 2.5: Büyük Veri Süreci (Gandomi ve Haider, 2015).

Geleneksel veri yönetiminde yapılandırılmış veri depolama ve erişim yöntemleri arasında ilişkisel veri tabanları ve veri ambarları bulunur. Veri dış kaynaklardan çıkartan, operasyonel ihtiyaçlara göre dönüştüren ve son olarak veri tabanına veya veri ambarına yükleyen; Çıkart (*Extract*), Dönüştür (*Transform*), Yükle (*Load*) veya Çıkart, Yükle, Dönüştür (ELT) araçları kullanılarak yüklenir. Böylece, veri madenciliği ve çevrimiçi analitik işlevler için kullanılabilir hale getirilmeden önce veri temizlenir, dönüştürülür, birleştirilir ve kataloglanır (Elgendy ve Elragal, 2014; Bakshi, 2015).

İlişkisel veri tabanları gittikçe daha pahalı olan donanımlar kullanmaktadır. Görünüşe göre geleneksel ilişkisel veri tabanları büyük verinin büyük hacmini ve heterojenliğini kaldıramaz. Araştırma toplulukları farklı açılardan bazı çözümler önermiştir (Chen ve diğ., 2014). Örneğin

bulut bilişim, maliyet verimliliği, esneklik ve sorunsuz sürüm yükseltme ve sürüm azaltma gibi büyük veri kümeleri için gerekli olan altyapı gereksinimlerini karşılamak için kullanılır. Büyük ölçekli ve düzensiz veri kümelerinin kalıcı olarak depolanması ve yönetilmesi için dağıtılmış dosya sistemleri (Howard ve diğ.,1988) ve NoSQL (Cattel, 2011) veri tabanları uygun seçeneklerdir.

2.2.2.1. Büyük Veri Analizi Teknikleri

Büyük Veri analizi teknikleri, istatistiksel analiz, veri madenciliği, makine öğrenmesi, yapay sinir ağları, sosyal ağ analizi, sinyal işleme, örüntü tanıma, optimizasyon yöntemleri ve veri görselleştirme yaklaşımları gibi çeşitli yöntemleri içerir. Birçok çalışma, kullanılan teknikleri geliştirerek, yenilerini önererek veya çeşitli algoritma ve teknolojilerin birleşimini test ederek büyük veriyi ele alır. Bu tez çalışmasında makine öğrenmesi yöntemleri kullanılmıştır.

Makine öğrenmesi veriden öğrenerek örüntüyü çıkarmakla ilgilidir. İstatistik, yapay zeka ve bilgisayar biliminin kesişme noktasındaki bir araştırma alanıdır ve aynı zamanda tahmine dayalı analitik veya istatistiksel öğrenme olarak da bilinir (Müller ve Guido, 2016). Makine öğrenmesinin en belirgin özelliği veriden öğrenmektir. Büyük Veri ile başa çıkabilmek için hem denetimli öğrenim hem de denetimsiz öğrenmeyi kapsayan makine öğrenimi algoritmalarını büyütmemiz gerekir (Chen ve Zhang, 2014).

Büyük veri analizinde kullanılabilecek diğer yöntemler; Tahmin Analitiği, Veri Madenciliği, Veri Görselleştirme, Sosyal Ağ Analizi'dir (Chen ve Zhang, 2014; Elgendy, 2014; Elragal, 2014; Ali ve diğ., 2016).

Tahmin Analitiği: Öngörücü analitik geçmiş deneyimlere dayanarak (toplanan veriler için) gelecekteki bazı olayları veya davranışları (veri madenciliği ve makine öğrenimi tekniklerini kullanarak) tahmin ederek rekabet avantajı sağlamayı amaçlayan bir teknolojiyi ifade eder. Öngörücü analitik, veri bilimi, makine öğrenimi, öngörülü ve istatistiksel modelleme içerir ve verilen girdi ampirik verilere dayalı ampirik tahminler çıkarır. Temel dayanak, geleceğin geçmiş deneyimler temelinde tahmin edilebileceğidir (Ali ve diğ., 2016).

Veri Madenciliği: Veri madenciliği, kümeleme analizi, sınıflandırma, regresyon ve ilişkilendirme kuralı öğrenimi de dahil olmak üzere veriden değerli bilgileri çıkarmak için kullanılan bir dizi tekniktir. Makine öğrenimi ve istatistik yöntemleri içerir. Büyük veri

madenciliği, geleneksel veri madenciliği algoritmalarına kıyasla daha zordur. Çoğu uzantı genellikle belirli miktarda büyük veri örneğini analiz etmeye dayanır ve örnek tabanlı sonuçlar genel verilere uygulanır (Chen ve Zhang, 2014; Emani ve diğ., 2014).

Veri Görselleştirme: Görselleştirme Yaklaşımları, verileri anlamak için tablolar, görüntüler, diyagramlar ve diğer sezgisel görüntüleme yollarını oluşturmak için kullanılan tekniklerdir (Chen ve Zhang, 2014).

Sosyal Ağ Analizi: Sosyal Ağ Analizi sosyal varlıklar arasındaki ilişkilerin yanı sıra bu ilişkilerin örüntüleri ve sonuçlarına odaklanmaktadır. Sosyal ağ analizi, kimin kim olduğunu ve kimin hangi bilgi veya bilgiyi kiminle paylaştığını ve neyi kullandığını, etkileşimde bulunan taraflar arasında bilgi akışını kolaylaştıran şeyleri kavramak için hem resmi hem de resmi olmayan ilişkileri planlar ve ölçer (Elgendy, 2014; Elragal, 2014).

Büyük Veri analizi çalışmaları çeşitli alanlara bölünebilir; yapısal veri analizi, metin veri analizi, multimedya veri analizi, sosyal medya analizi ve mobil veri analizi gibi çeşitli alanlara bölünebilir (Tablo 2.2). Böyle bir sınıflandırma veri özelliklerini vurgulamayı amaçlar, bazı alanlarda benzer teknolojiler kullanılabilir.

Tablo 2.2: Büyük veri analizi alanları (Chen ve Zhang, 2014, Gandomi ve Haider, 2015).

Kategori	Açıklama
Yapısal Veri Analizi	İlişkisel veri tabanlarına dayanan yapısal veriyi analiz etmeyi hedefler. İş uygulamaları ve bilimsel araştırmalar çok büyük yapılandırılmış veri oluşturabilir ve bu veriyi analiz etme ihtiyacı oluşabilir.
Metin	Bilgi toplamak için büyük metin kümelerini (e-posta, web sayfaları vb.) analiz etmeyi amaçlar. Konu modelleme, bilgi çıkarma, metin özetleme, soru cevaplama, duygu analizi için kullanılır. Doğal Dil İşleme yöntemlerine dayanır.
Multimedya	Görüntü, ses, video gibi multimedya içeriklerini analiz etmeyi içerir. Multimedya verileri yapısal olmayan verilerdir. Multimedya analizine örnek olarak konuşma analizi, video özetleme, görüntü tanımlama, multimedya öneri sistemleri verilebilir.
Sosyal Medya Analizi	Sosyal medya analitiği, sosyal medya kanallarından gelen yapılandırılmış ve yapılandırılmamış verilerin analizini ifade eder. Sosyal medya, kullanıcıların içerik oluşturmasına ve değiş tokuş etmesine olanak tanıyan çeşitli çevrimiçi platformları kapsayan geniş bir terimdir. Sosyal medya analitiği üzerine yapılan araştırmalar, psikoloji, sosyoloji, antropoloji, bilgisayar bilimi, matematik, fizik ve ekonomi dahil olmak üzere çeşitli disiplinleri kapsamaktadır. Pazarlama alanı son yıllarda sosyal medya analitiğinin birincil uygulaması olmuştur

Tablo 2.2: Büyük veri analizi alanları (Chen ve Zhang, 2014, Gandomi ve Haider, 2015) (devam).

Kategori	Açıklama
Mobil Trafik Analizi	Mobil kullanıcı sayısının artması ile birlikte, cep telefonları artık farklı kültürlere, ilgi alanlarına ve coğrafi konumlara ait toplulukları birleştirmede etkin rol oynamaktadır. Bireyler mobil ağlar aracılığıyla aynı hobilere sahip insanlarla bir araya gelebilir ve ortak hedefler oluşturabilir. RFID etiketleri fiziksel objeleri takip edebilir, wireless sensörler kişinin sağlık durumunu gözlemleyebilir. Mobil cihazlar, sensörler ve wireless vb gibi cihazlardan üretilen veriyi analiz etmek önemli bir ihtiyaçtır.

2.2.2.2. Büyük Veride Sınıflandırma Algoritmaları

Kullanılan makine öğrenmesi sınıflandırma algoritmaları aşağıdaki gibidir.

1. **Lojistik Regresyon:** Lojistik regresyon, bağımlı değişkenin sabit sayıda değerden birini alabildiği bir regresyon modelidir. Örnek sınıfı ile girdiden çıkarılan özellikler arasındaki ilişkiyi ölçmek için bir lojistik fonksiyonu kullanır. İkili sınıflandırma için yaygın olarak kullanılmasına rağmen, çok sınıflı sınıflandırma problemlerini çözmek için genişletilebilir.

   Lojistik regresyon modelinin spesifik formu (Hosmer & Lemeshow, 1989):

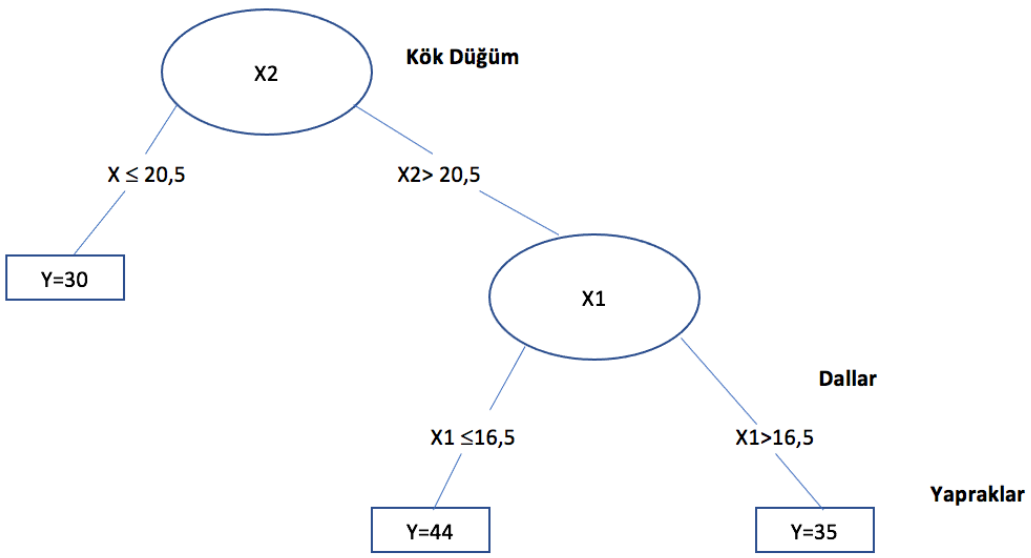
$$\pi(x) = \frac{e^{\beta_0+\beta_1\chi}}{1+e^{\beta_0+\beta_1\chi}} \tag{2.1}$$

   $\pi(x)$ lojistik fonksiyonunun dönüşümü, logit dönüşüm olarak bilinir:

$$g(x) = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_0 + \beta_1\chi \tag{2.2}$$

   Doğrusal regresyon modeli altında en küçük kareler fonksiyonuna giden geleneksel tahmin yöntemine, bir lojistik regresyon modelinin parametrelerini tahmin etmek için temel sağlayan maksimum olabilirlik denir (Nie ve diğ,2011). $\beta_0, \beta_1, \ldots . \beta_k$ regresyon katsayıları maksimum olabilirlik yöntemi ile tahmin edilir. $x_1, x_2, \ldots x_k$ değerleri bağımsız değişkenlerdir. Lojistik regresyon fonksiyonu, π(x), bir olayın gerçekleşip gerçekleşmediğine bağlı olarak 1 veya 0 olarak hesaplanır.

2. **Karar Ağaçları:** Sınıflandırma yöntemlerinden bir diğeri karar ağaçlarıdır (*decision trees*)**.** Bir karar ağacı, böl ve yönet stratejisini uygulayan hiyerarşik bir veri yapısıdır (Alpaydın, 2020). Ağaç yapısı, dallar ve yapraklardan oluşur. Şekil 2.6'da bir karar ağacının yapısı görülmektedir. En üstte bulunan yapı kök düğüm, en sonda bulunan yapı ise yapraklardır ve bunların arasında kalan yapılar ise dal olarak adlandırılır (Quinlan, 1993). Karar ağacı, her dahili düğümün bir özniteliği ifade ettiği, her dalın bir sonucu temsil ettiği ve her yaprak düğümün bir sınıf etiketi tuttuğu akış şemasına benzer yapılardır (Han ve diğ., 2012). Uygulama kolaylığı ve diğer sınıflandırma algoritmalarına kıyasla daha kolay anlaşılması nedeniyle en yaygın kullanılan algoritmadır (Yadav ve diğ., 2012).



Şekil 2.6: X1 ve X2 özniteliklerini içeren bir karar ağacı (Özkan, 2020).

Sınıflandırma ağaçlarında en önemli sorunlardan birisi hangi kökten itibaren bölümlemenin veya başka bir ifadeyle dallanmanın yapılacağıdır. Sınırlı sayıda veriden oluşan bir eğitim kümesinden yararlanarak mümkün olan tüm ağaç yapılarını ortaya çıkarmak ve içlerinden en uygun olan öznitelikten itibaren dallanmayı başlatmak kolay bir işlem olmamasından dolayı, karar ağacı algoritmalarının çoğu daha başlangıçta birtakım değerleri hesaplayarak ona göre ağaç oluşturmaya başlarlar (Özkan, 2020). Bu hesaplamalar belli bir kriterlere göre yapılır.

Aşağıda entropi, bilgi kazancı, kazanım oranı kavramları hesaplama formülleri ile birlikte açıklanmıştır (Quinlan, 1996; Han ve diğ., 2012, Akpınar, 2014, Özkan, 2020).

- **Bilgi Kazancı (*Information Gain*):** Claude Shannon'ın (Shannon, 1948) mesajların değerini veya *bilgi içeriğini* inceleyen bilgi teorisi üzerine yaptığı öncü çalışmasına dayanmaktadır.

  Öncelikle denklem (2.3) ile bilgi miktarı veya entropi hesaplanır.

$$E(T) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{2.3}$$

  Bir X özniteliği tarafından, T sınıf niteliğine katkıda bulunan bilgi miktarı denklem (2.4) ile hesaplanır. Alt kümelerin entropilerinin ağırlıklı ortalamasıdır.

$$Entropi_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} E(T_i) \tag{2.4}$$

  İki denklemden elde edilen bilgi kazancı denklem (2.) ile hesaplanır.

$$Bilgi\ kazancı(X) = E(T) - E(X,T) \tag{2.5}$$

  En yüksek bilgi kazancını sağlayan öznitelikten itibaren dallanma yapılır.

Quinlan (1993) kazanç oranının hesaplanmasında bilgi bölünmesi (*split information*) kavramını ortaya çıkarmıştır. Bir özniteliğin çok sayıda değer içeriyor olması durumunda ağırlıklı ortalamaların değeri sıfır veya sıfıra yakın bir değer alacak ve bilgi kazancı en büyük değere ulaşıp yanıltıcı sonuçlar oluşturabilecektir (Özkan, 2020). Bu durumun engellenmesi için kazanç ölçütü yerine kazanç oranı kullanılır. Kazanç oranı denklem (2.7) ile hesaplanır.

Bu değer X niteliğinin k farklı değeri için, T kümesi üzerindeki k farklı sonuçlarına karşılık gelen potansiyel bilgiyi temsil eder. Bilgi bölünmesi olarak bilinir ve denklem (2.7) ile hesaplanır.

$$SplitInfo_X(T) = -\sum_{i=1}^{k} \frac{|T_i|}{|T|} \log_2\left(\frac{|T_i|}{|T|}\right) \tag{2.6}$$

$$Gain\ Ratio = Kazanç\ oranı(X) = \frac{Kazanç(X)}{SplitInfo(X)} \tag{2.7}$$

Karar ağaçları algoritmalarına ID3, CART (*Classification and Regression Trees*), C4.5 ve C5.0 algoritmaları örnek verilebilir. Bu algoritmalardan ID3 bilgi kazancına göre, onun gelişmiş versiyonu olan ve sayısal öznitelikler ile karar ağaçları oluşturulmasına izin veren C4.5 ve C5.0 ise kazanım oranına göre dallanma yapmaktadır. ID3 ve C4.5 algoritmaları Quinlan tarafından, Sınıflandırma ve Regresyon Ağaçları (CART) ise Leo Breiman, Jerome Friedman, Richard Olshen ve Charles J. Stone'dan oluşan bir gup istatistikçi tarafından geliştirilmiştir (Brieman ve diğ., 1984; Han ve diğ., 2012).

3. **Rastgele Orman (*Random Forests*):** Rastgele Orman topluluk öğrenme (*ensemble learning*) yöntemlerinden biridir. Rastgele orman, her ağacın ormandaki tüm ağaçlar için aynı dağılıma sahip bağımsız olarak örneklenen rastgele bir vektörün değerlerine bağlı olacak şekilde ağaç tahmin modellerinin bir kombinasyonudur (Brieman, 2001).

   Rastgele Orman algoritmasının detayına inmeden önce torbalama (bagging) yöntemini açıklamak gerekmektedir. Torbalama yöntemi, bir tahmincinin birden çok sürümünü oluşturarak ve bunları toplu bir tahminci elde etmek için kullanan bir yöntemdir. Her tahmincinin ürettiği sonuçların ortalaması alınır. Sınıf tahmini yapılırken oylama yapılarak nihai sonuç, topluluğun sonucu olarak ilan edilir (Breiman,1996).

   Rastgele karar orman için ilk algoritma Ho (Ho,1998) tarafından *random subspace* yöntemi kullanılarak oluşturulmuştur. Algoritmanın bir uzantısı Leo Brieman (Brieman, 2001) tarafından geliştirilmiştir. Bu algoritma torbalama yöntemi (Breiman, 1996) ile random subspace (Ho, 1998) yöntemlerini birleştirir. Algoritma bootstrap yöntemi ile eğitim veri setinden rastgele örnekler seçerek bir dizi ağaç oluşturur. Ağaçlar oluşturulduktan sonra her bir düğümünde en iyi dallara ayırıcı değişken, tüm değişkenler arasından rastgele seçilen daha az sayıdaki değişken arasından seçilir. Bu seçim işlemi random subspace uygulamasıdır (Ho,1998). Bu iki mekanizma, rastgele ormandaki tüm ağaçların farklı olmasını sağlar. Karar ormanını oluşturacak ağaçlar, ormanı oluşturacak nitelikler ve değişkenler seçildikten sonra CART algoritması kullanılarak büyütülür (Brieman, 2001).

   Rastgele Orman kullanarak tahmin yapmak için algoritma önce ormandaki her ağaç için bir tahmin yapar. Bu her ağacın, olası her çıktı etiketi için bir olasılık sağlayan bir tahmin yaptığı anlamına gelir. Sınıflandırma için oylama stratejisi kullanılır, her ağacın ürettiği

tahmin değeri oylanır ve en fazla oy alan tahmini değeri nihai tahmin değeri olarak sunulur. Tüm ağaçların tahmin ettiği olasılıkların ortalaması alınır ve en yüksek olasılığa sahip sınıf tahmin sınıfı olarak sunulur (Brieman, 2001).

Karar Ağaçlarının temel bir dezavantajı eğitim verisine aşırı uyum gösterme eğiliminde olmalarıdır. Rastgele Orman algoritması bu sorunu çözmenin bir yolu olarak ortaya çıkmıştır. Rastgele orman, esasen her ağacın diğerlerinden biraz farklı olduğu bir karar ağaçları topluluğudur. Rastgele Orman'ın arkasındaki fikir, her ağacın göreceli olarak iyi bir tahmin işi yapabileceği, ancak muhtemelen bir kısmının eğitim verisine aşırı uyum durumu göstereceğidir. Hepsi iyi tahmin yapan ve farklı şekillerde veri setine aşırı uyum gösteren birçok ağaç inşa edilerek ve sonuçlarının ortalaması alınarak aşırım uyum miktarı azaltılır (Müller ve Guido, 2016).

4. **Gradyan Artırma Makineleri (*Gradient Boosting Machines – GBM*):** Gradient Boosting daha güçlü bir model oluşturmak için birden çok karar ağacını birleştiren başka bir topluluk öğrenme yöntemidir. Boosting yöntemi topluluk içerisindeki zayıf sınıflandırıcıları bir araya getirerek güçlü bir sınıflandırıcı oluşturmayı amaçlar. Jerome H. Friedman tarafından geliştirilmiştir (Friedman, 1999a) (Friedman, 1999b).

   Spesifik olarak, her yinelemede, eğitim verisinin bir alt örneği, tam eğitim veri setinden rastgele ve değiştirilmeden alınır. Bu rastgele seçilen alt örnek daha sonra temel öğreniciye uyması ve mevcut yineleme için model güncellemesini hesaplaması için tam örneklem yerine kullanılır (Friedman, 1999b). Rastgele orman yaklaşımının aksine Gradient Boosting, ağaçları seri bir şekilde inşa ederek çalışır, burada her ağaç bir öncekinin hatalarını düzeltmeye çalışır. Gradient Boosting algoritmasının arkasındaki temel yaklaşım zayıf öğrenen ağaçları birleştirmektir. GBM genellikle çok sığ ağaçlar kullanır, bu da modeli bellek açısından daha küçük hale getirir ve tahminleri daha hızlı yapar. Her ağaç yalnızca verinin bir kısmı hakkında iyi tahminler sağlar ve bu nedenle performansı yinelemeli olarak iyileştirmek için daha fazla ağaç eklenir (Müller ve Guido, 2016).

   İlk olarak her bir gözlemin eşit ağırlıkta olduğu bir karar ağacı eğiterek başlanır. İlk ağacı değerlendirdikten sonra sınıflandırması zor olan gözlemlerin ağırlıklarını arttırır, sınıflandırması kolay olanların ağırlıklarını düşürür. İkinci ağaç bu ağırlıklı veri

üzerinden oluşturulur. Buradaki amaç ilk ağacın öngörülerini geliştirmektir. Dolayısıyla yeni modelimiz Ağaç 1 + Ağaç 2'dir. Daha sonra bu yeni 2 ağaçlı topluluk modelinden sınıflandırma hatası hesaplanır ve revize edilmiş kalıntıları (*residuals*) tahmin etmek için üçüncü bir ağaç eklenir. Bu işlem belirli sayıda yineleme için tekrarlanır. Sonraki ağaçlar, önceki ağaçlar tarafından iyi sınıflandırılmamış gözlemleri sınıflandırmaya yardımcı olur. Nihai topluluk modelinin tahminleri, önceki ağaç modelleri tarafından yapılan tahminlerin ağırlıklı toplamıdır. Gradient Boosting algoritması zayıf öğrenicilerin eksikliklerini kayıp fonksiyonundaki gradyanları kullanarak belirler. Kayıp fonksiyonu, modelin katsayılarının temeldeki veriye uymada ne kadar iyi olduğunu gösteren bir ölçüdür (Friedman,1999b).

2.2.2.3. Büyük Veri Analizi Araçları

Geleneksel araçlar ve teknolojiler büyük miktarda veri depolamak, işlemek ve en önemli olan bu veriden değer çıkarmak için yeterli değildir. Büyük verinin hızlı büyümesi, üretilen verinin özellikleri bu verinin işlenmesinde yeni teknolojilere ihtiyacı duyulmuştur. Büyük veri kavramının özellikleri göz önüne alınarak bu ihtiyaca cevap olacak teknolojiler geliştirilmiştir.

Apache Hadoop:

Apache Hadoop, büyük veriyle yakından ilişkili bir teknolojidir. Apache'nin açık kaynaklı bir projesi olan Nutch tarafından geliştirildi ve ilk tasarımı Doug Cutting ve Mike Cafarella tarafından tamamlandı (Zikopoulos ve Eaton, 2011). 2006 yılında Hadoop, Yahoo, Facebook ve diğer internet şirketleri tarafından yaygın olarak kullanılan bağımsız bir açık kaynaklı Apache projesi oldu.

Apache Hadoop yazılım kütüphanesi, büyük veri kümelerinin basit programlama modelleri kullanarak bilgisayar kümeleri arasında dağıtılmış olarak işlenmesine izin veren bir çerçevedir (Apache Hadoop, 2020). Üzerinde çalışan uygulamaları ve *Hadoop Distributed File System* (HDFS) olarak adlandırılan dağıtık dosya sistemi ile MapReduce programlama modelini bir araya getirmektedir. Java ile geliştirilmiş açık kaynaklı bir yazılım kütüphanesidir. Hadoop'un HDFS ve MapReduce programlama çerçevesi olmak üzere iki ana bileşeni vardır. Her biri yerel hesaplama ve depolama sunan tek sunucudan binlerce makineye ölçeklendirilmek üzere tasarlanmıştır.

Hadoop aşağıdaki modülleri içerir (Apache Hadoop, 2020).

- **Hadoop Common:** Diğer Hadoop modüllerinin ihtiyaç duyduğu kütüphaneleri ve yardımcı programları içerir.

- **Hadoop Distributed File System (HDFS):** Uygulama verisine yüksek verimli erişim sağlayan dağıtık dosya sistemi.

- **Hadoop YARN:** Hadoop küme kaynak yönetim sistemidir.

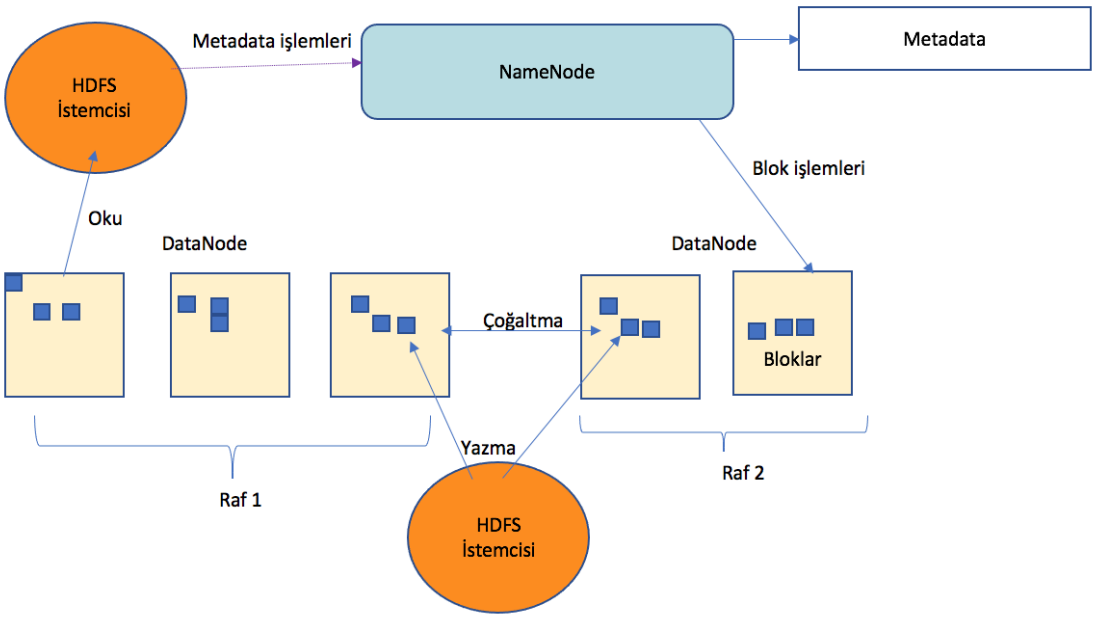
- **Hadoop MapReduce:** Büyük ölçekli veri kümelerinin işlenmesi için MapReduce programlama modelinin bir uygulaması.

- **Hadoop Ozone:** Hadoop'un nesne deposu.

MapReduce, büyük veriyi büyük kümelerde işlemek üzere geliştirilmiş bir dağıtık programlama modelidir. Genel yapısı itibariyle MapReduce, map ve reduce olmak üzere iki ana fonksiyondan oluşur. Map (Haritala veya Eşle), bir yığının tüm üyelerini sahip olduğu fonksiyon ile işleyip bir sonuç listesi döndürür. Reduce (İndirge) ise paralel bir şekilde çalışan iki veya daha fazla map fonksiyonundan dönen sonuçları özetler. Map ve Reduce fonksiyonları paralel bir şekilde çalışırlar, aynı anda aynı sistemde olmaları gerekmez. Birçok programlama dili kullanarak (Java, C++, Python, Perl, Ruby, C vs.) MapReduce programlama modeli geliştirilebilir.

Hadoop Dağıtılmış Dosya Sistemi bir veri depolama sistemidir. Bir kümede yüzlerce düğümü destekler ve uygun maliyetli, güvenilir bir depolama sistemi sağlar. Büyük hacimli yapılandırılmış ve yapılandırılmamış veriyi işleyebilir ve saklayabilir. HDFS yüksek derecede hataya dayanıklıdır ve her biri yerel olarak hesaplama ve depolama imkânı sunan tek bir sunucudan binlerce sunucuya ölçeklenebilir. Şekil 2.7'de Hadoop Dağıtılmış Dosya Sisteminin master/slave mimarisi gösterilmektedir (Apache Hadoop HDFS Architecture, 2020). Büyük veriyi küme boyunca dağıtır. Kümede dosya sistemi işlemlerini yöneten bir ana (*NameNode*) ve tek tek hesaplama düğümlerindeki veri depolamasını yöneten ve koordine eden birçok *slave* (*DataNodes*) bulunur. *NameNode*, ana sunucu görevi görür ve dosya sistemi ad alanını yönetme, istemcinin dosyalara erişimini düzenleme, dosyaları yeniden adlandırma, kapatma ve açma gibi dosya sistemi işlemlerini yürütme görevini yerine getirir. *DataNode*, *NameNode* için köle görevi görür. İstemcinin isteğine göre dosya sistemlerinde okuma-yazma işlemleri

gerçekleştirir. Genellikle kümedeki düğüm başına bir tane olmak üzere bir dizi *DataNode* vardır. HDFS, bir dosya sistemi ad alanını ortaya çıkarır ve kullanıcı verisinin dosyalarda saklanmasına izin verir. Dahili olarak, bir dosya bir veya daha fazla bloğa bölünür ve bu bloklar bir *DataNode* setinde saklanır. *DataNode*, *NameNode* talimatlarına göre blok oluşturma, silme ve çoğaltma gibi işlemleri de gerçekleştirirler. *NameNode*, blokların *DataNode*'lara eşlenmesini belirler. *DataNode'lar*, ana sunucudan gelen okuma ve yazma isteklerinin sunulmasından sorumludur. Hadoop veri kullanılabilirliği sağlamak için veri çoğaltması yapar. Genel olarak, kullanıcı verisi HDFS dosyalarında saklanır. Bir dosya sistemindeki dosya bir veya daha fazla bölüme ayrılır ve/veya tek tek veri düğümlerinde saklanır. Bu dosya segmentlerine blok denir. Başka bir deyişle, HDFS'nin okuyabileceği veya yazabileceği minimum veri miktarına Blok adı verilir.



Şekil 2.7: HDFS Mimarisi (Apache Hadoop HDFS Architecture, 2020).

Çeşitli Bilgi Teknolojisi şirketleri ve toplulukları, Hadoop altyapısını, araçlarını ve hizmetlerini geliştirmek ve zenginleştirmek için çalışmıştır. Cloudera Hortonworks Data Platform[1], Amazon

[1] https://www.cloudera.com/products/hdp.html

Elastic MapReduce[2], MapR[3], IBM InfoSphere BigInsights[4], GreenPlum Pivotal HD[5], Oracle Big Data[6], Windows Azure HDInsight[7] Hadoop'un ticari dağıtımlarıdır.

2.3. APACHE SPARK

Spark ilk olarak 2009 yılında Berkeley Üniversitesi AMP laboratuvarında Matei Zaharia tarafından başlatılmış ve 2010 yılında açık kaynaklı hale getirilmiş, 2013 yılında proje Apache Yazılım Vakfı'na bağışlanmıştır (Zaharia ve diğ.,2016).

Piyasaya sürülmesinden bu yana, birleşik analitik motoru olan Apache Spark, çok çeşitli sektörlerdeki işletmeler tarafından hızlı bir şekilde benimsenmiştir. Netflix, Yahoo ve eBay gibi internet teknolojileri şirketleri, Spark'ı kendi sistemlerine uygulamış ve toplu olarak 8.000 düğümü aşan kümelerde birden fazla petabayt seviyesinde veri işlemiştir. 250'den fazla kuruluşun katkısıyla hızla büyük veri için en büyük açık kaynak topluluğu haline gelmiştir (Databricks, 2020).

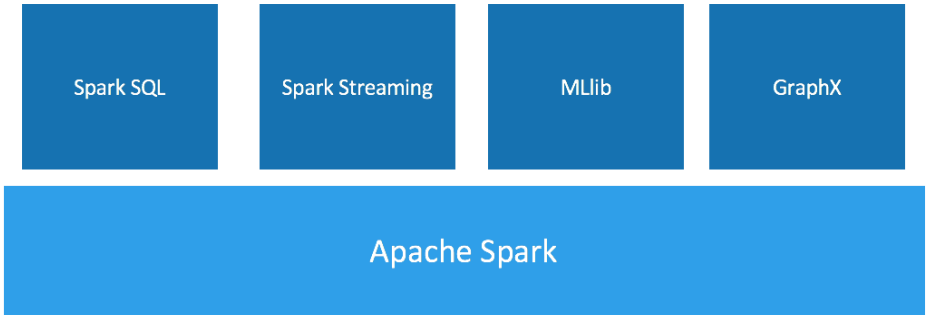
Apache Spark büyük ölçekli veri işleme için birleşik bir analiz motorudur (Apache Spark, 2020). Bir uygulamanın, farklı türlerdeki büyük veri kümelerini işleyebilmesi için CPU, bellek ve depolama kaynaklarını bir sunucu kümesinde kolayca kullanılmasını sağlayan basit bir programlama arabirimi sağlar. Gerçek zamanlı veri işleme, makine öğrenmesi ve graf (çizge) hesaplama için kullanılabilir.

Apache Spark, Scala programlama dili ile geliştirilmiştir ve uygulamanın varsayılan programlama dilidir, ancak Java, Python ve R gibi birden çok dilde bir uygulama programlama arabirimi (*API*) sağlar. Bu dillerden herhangi birinde bir Spark uygulaması geliştirilebilir.

Apache Spark projesi, birbirine çok yakın entegre bileşenler içermektedir. Özünde birçok makinede çalışan ve birden fazla hesaplama görevinden oluşan uygulamaları planlamak, dağıtmak ve izlemekle sorumlu olan bir hesaplama aracıdır. Spark hem hızlı hem de genel amaçlı olduğundan, SQL ve DataFrames, makine öğrenmesi için MLlib, GraphX ve Spark

[2] https://aws.amazon.com/emr/
[3] https://www.hpe.com/us/en/software/data-fabric.html
[4] https://www.ibm.com/support/knowledgecenter/SSPT3X_3.0.0/
[5] https://greenplum.org/
[6] https://www.oracle.com/tr/big-data/
[7] https://azure.microsoft.com/tr-tr/services/hdinsight/

Streaming dahil olmak üzere birçok kütüphaneyi içerir (Şekil 2.8). Bu kütüphaneler aynı uygulamada sorunsuz bir şekilde çalışacak şekilde tasarlanmıştır.

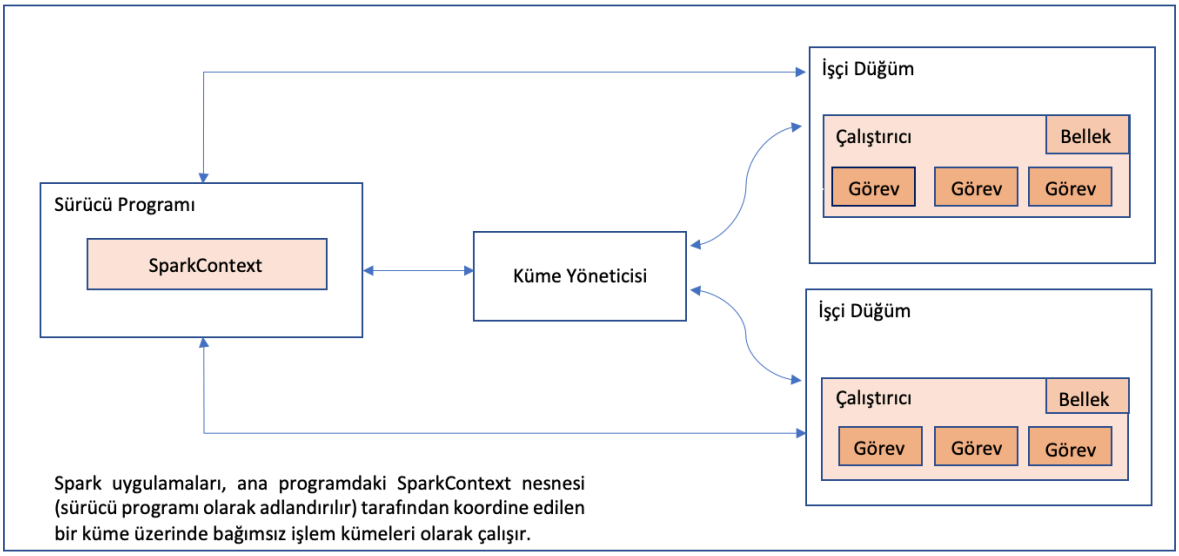


Şekil 2.8: Apache Spark Kütüphaneleri (Apache Spark, 2020).

Spark genel amaçlı bir yazılım çerçevesidir ve çeşitli büyük veri uygulamaları için kullanılabilir. Ancak hızın çok önemli olduğu büyük veri uygulamaları için idealdir. Bu tür uygulamalar, etkileşimli analiz ve yinelemeli veri işleme algoritmalarını kullanan uygulamalardır (Zaharia ve diğ.,2010).

- **Yinelemeli (*Iterative Jobs*):** Yinelemeli algoritmalar aynı veri üzerinde birden çok kez yinelenen veri işleme algoritmalarıdır. Pek çok yaygın makine öğrenimi algoritması, bir parametreyi optimize etmek için aynı veri kümesine tekrar tekrar bir işlev uygular.
- **Etkileşimli Analiz (*Interactive Analysis*):** Etkileşimli veri analizi bir veri kümesini etkileşimli olarak keşfetmeyi içerir. Spark'ın etkileşimli analiz için ideal olmasının nedeni yine bellek içi bilgi işlem yetenekleridir.

Bir Spark uygulaması Şekil 2.9'da görüldüğü gibi beş temel varlık içerir: Sürücü programı (*driver program*), kaynak yöneticisi (*cluster manager*), işçi düğüm (*worker node*), çalıştırıcı düğüm (*executor*) ve görev (*task*) (Apache Spark Overview, 2020).

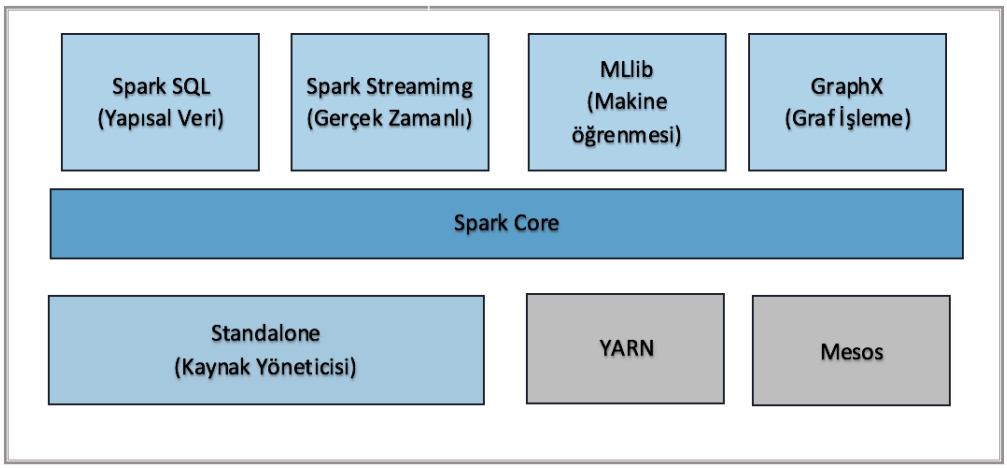


Şekil 2.9: Apache Spark çalışma prensibi (Apache Spark Cluster Mode Overview, 2020).

İşçi düğüm Spark uygulamasına CPU, bellek ve depolama kaynakları sağlar. Bir düğüm kümesine dağıtılmış olan bir Spark uygulamasını çalıştırır. Spark, bir işi yürütmek üzere küme kaynaklarını elde etmek için bir küme kaynak yöneticisi kullanır. Kaynak yöneticisi, adından da anlaşılacağı gibi bilgi işlem kaynaklarını, çalışan düğümler kümesi üzerinde yönetir. Uygulamalar arasında küme kaynaklarının düşük düzeyde zamanlanmasını sağlar. Birden çok uygulamanın küme kaynaklarını paylaşmasını ve aynı işçi düğümlerinde çalışmasını sağlar. Spark şu anda üç küme kaynak yöneticisini desteklemektedir: Apache Spark'ın kendine ait Spark Standalone adı verilen kaynak yöneticisi, Mesos ve YARN. Mesos ve YARN, Spark ve Hadoop uygulamalarını aynı işçi düğümlerde aynı anda çalıştırmanıza izin verir. Ana düğümdeki sürücü programı SparkContext'i oluşturur ve işçi düğümlere veri işleme kodunu sağlar. Ana düğüm, bir Spark kümesinde bir veya daha fazla işi başlatabilir. Çalıştırıcı düğüm, Spark'ın bir uygulama için her işçi düğüm üzerinde oluşturduğu bir java sanal makinesi işlemidir. Her uygulamanı kendi çalıştırıcısı vardır. Uygulama kodunu aynı anda birden çok iş parçacığında (*task*) yürütür. Ayrıca bellek veya diskteki veriyi önbelleğe alabilir. Bir çalıştırıcı düğüm, oluşturulduğu uygulama ile aynı ömre sahiptir. Bir Spark uygulaması sona erdiğinde, bunun için oluşturulan tüm çalıştırıcılar da sona erer. Görev (*task*), Spark'ın bir ana düğüme gönderdiği en küçük iş birimidir. Alt düğümdeki bir çalıştırıcı tarafından yürütülür. Her görev, bir sonucu bir ana düğüm programına döndürmek veya çıktısını diğer düğüm kümelerine dağıtmak için bölümlemek için bazı hesaplamalar yapar. Spark, her veri bölümü için bir görev oluşturur. Bir çalıştırıcı düğüm aynı anda bir veya daha fazla görevi yürütür. Her iş, birbirine

bağlı aşama (*Stage*) adı verilen daha küçük görev gruplarına bölünür. Paralellik miktarı aşama sayısı ile belirlenir. Daha fazla aşama, veriyi paralel olarak işleyen daha fazla görev anlamına gelir.

Bir Spark uygulaması çalıştırıldığında bir küme kaynak yöneticisine bağlanır ve küme içindeki işçi düğümlerdeki çalıştırıcıları alır, sonrasında uygulama kodunu (SparkContext'e iletilen JAR veya Python dosyaları) çalıştırıcılara gönderir. Son olarak, SparkContext çalıştırıcılara çalıştırmaları için görevler gönderir (Apache Spark Overview, 2020). Spark, bir işi yönlendirilmiş döngüsüz grafiğe (*directed acyclic graph*) böler (Salloum ve diğ., 2016). Daha sonra, bir küme kaynak yöneticisi tarafından sağlanan düşük düzeyli bir zamanlayıcı kullanarak bu aşamaların çalıştırıcı düğümler üzerinde yürütülmesini zamanlar ve çalıştırıcı düğümler, Spark tarafından verilen görevleri paralel olarak yürütür (Guller, 2015).



Şekil 2.10: Apache Spark Mimarisi (Karau ve diğ.,2015).

Şekil 2.10'da gösterilen Spark kütüphanelerinin her birini kısaca tanıtılacaktır.

2.3.1. Spark Core ve Resilient Distributed Datasets (RDDs)

Spark Core kütüphanesi Apache Spark projesinin temeli olarak düşünülebilir. Spark Core, görev zamanlama, bellek yönetimi, hata giderme, depolama sistemleriyle etkileşim gibi Spark'ın temel işlevlerini içerir.

Spark Core, Spark'ın ana programlama modelini tanımlayan RDDs'ye ev sahipliği yapar. RDDs'ler, bellek içerisinde tutulan, paralel olarak işlenebilen ve birçok hesaplama düğümüne

dağıtılabilen bir öğe koleksiyonunu temsil eder. RDDs'ler, Hadoop ya da bir başka dosya sistemindeki bir veriden veya kullanıcılar tarafından Spark programında üretilen veriden de oluşturulabilir. Üretilen her RDDs, bölümlenerek dağıtık sistemde yer alan bilgisayarlarda konumlandırılan çalıştırıcı düğüm'ler (*Executor*) ile işlenir. RDDs'ler iki tür işlemi destekler: var olan bir veri kümesinden yeni bir veri kümesi oluşturan *Transformations* ve veri kümesinde bir hesaplama yaptıktan sonra ana düğüm programına bir değer döndüren *Actions* (Apache Spark, 2020).

2.3.2. Spark SQL

Spark SQL, yapılandırılmış veri işleme için bir tasarlanan Spark modülüdür. Apache Spark'ın 1.0 versiyonu ile ortaya çıkmış bir kütüphanedir (Apache Spark, 2020). Yapılandırılmamış veri Dataframe adı verilen veri yapılarına dönüştürülerek SparkSQL tablolarında tutulur. Bu veri yapısı SQL üzerinden sorgulama yapılmasına izin verir. DataFrame, adlandırılmış sütunlar halinde düzenlenmiş bir veri kümesidir. Kavramsal olarak ilişkisel veri tabanlarındaki bir tabloya veya R/Python'daki bir Data Frame'e eşdeğerdir.

2.3.3. Spark Streaming

Spark Streaming, akan verinin ölçeklenebilir, yüksek verimli, hataya dayanıklı olarak işlenmesini sağlayan temel Spark bir uzantısıdır. Veri Kafka, Flume, Kinesis veya TCP soketleri gibi birçok kaynaktan alınabilir ve üst düzey fonksiyonlar kullanılarak işlenebilir. İşlenen veri dosya sistemlerine, veri tabanlarına ve canlı gösterim tablolarına aktarılabilir. (Apache Spark Streaming Programming Guide, 2020). Apache Spark Streaming teknolojisi kullanılarak makina öğrenmesi ve grafik hesaplama algoritmalarının akan veriye uyarlanması mümkün hale getirilmiştir.

2.3.4. MLlib

MLlib, Spark'ın makine öğrenimi kütüphanesidir. Amacı, makine öğrenimini ölçeklenebilir ve kolay hale getirmektir (Apache Spark MLlib Guide, 2020). Algoritmaların paralel kümeler üzerinde çalışmasına olanak sağlamaktadır.

Paralelliğinin önemli olduğu büyük ölçekli makine öğrenmesi algoritmalarının geliştirilmesini sağlar. Bu yinelemeli algoritmalar, yinelemeli hesaplamalar için tasarlanmış Spark çekirdeği

ile uyumlu bir şekilde çalışır. Makine öğrenmesi algoritmaları, boru hatları, öznitelik çıkarımı ve öznitelik dönüşümleri, model eğitme, model değerlendirme ve model optimizasyonu gibi görevleri yerine getirir. Bu bağlamda MLlib bu tür algoritmaların ve boru hatlarının tasarımını ve uygulamasını basitleştirmek için dağıtılmış bir makine öğrenimi kütüphanesi olarak tasarlanmıştır.

RDDs tabanlı (MLlib) ve DataFrame tabanlı (Spark ML) olmak üzere iki farklı paketi vardır. MLlib Apache Spark'ın ilk makine öğrenmesi kütüphanesidir. Spark 2.0'dan itibaren, spark.mllib paketindeki RDD tabanlı API'ler bakım moduna girmiştir. Spark için birincil makine öğrenmesi kütüphanesi artık dataframe tabanlı spark.ml paketidir. (Apache Spark MLlib Guide, 2020). Bu tez çalışmasında spark.ml kullanılmıştır.

Her iki paket de öznitelik çıkarımı, dönüşümler, model eğitimi, model değerlendirme ve model optimizasyonu gibi çeşitli ortak makine öğrenmesi görevlerini destekler (Apache Spark MLlib Guide, 2020).

- Sınıflandırma, regresyon, kümeleme ve iş birliğine dayalı filtreleme gibi sık kullanılan makine öğrenmesi algoritmalarını destekler.
- Özellik oluşturmak için öznitelik çıkarma, dönüştürme, boyut azaltma ve seçim gibi işlemleri destekler.
- Makine öğrenmesi boru hatlarının (*pipeline*) oluşturulması, değerlendirilmesi ve ayarlanması sağlar.
- Algoritmalar ve boru hatlarını kaydedilebilir ve yüklenebilir.
- Lineer cebir, istatistik, veri işleme vb. yardımcı araçları içerir.

Boru hattı, spark.ml, makine öğrenimi iş akışlarının oluşturulmasını, ayarlanmasını ve denetlenmesini kolaylaştırmak için Spark 1.2 versiyonu ile projeye dahil olmuştur. Bir makine öğrenimi iş akışı, belirli bir sırayla çalıştırılacak bir dizi algoritmadan oluşan bir boru hattı (pipeline) olarak temsil edilir. Bu aşamaların her biri bir Transformer veya bir Estimator olabilir (Salloum ve diğ., 2016). Transformer, mevcut bir DataFrame'den yeni bir DataFrame oluşturur. Bir DataFrame'i girdi olarak alan ve ilk DataFrame'e bir veya daha fazla yeni sütun ekleyerek yeni bir DataFrame döndüren *transform* adında bir yöntem uygular. Bir Estimator, bir eğitim veri kümesinde makine öğrenme modelini eğitir veya modeli bir veri setine uygular. Bir makine

öğrenme algoritmasını temsil eder. DataFrame'i bağımsız değişken olarak alan ve bir makine öğrenme modeli döndüren *fit()* adlı bir yöntem uygular (Guller, 2015).

2.3.5. Apache Spark'ın Avantajları

Spark, büyük veri dünyasındaki en aktif açık kaynak projesidir. Son yıllarda Hadoop'tan daha fazla konuşulur hale gelmiştir ve Hadoop MapReduce'un varisi olarak kabul edilmektedir (Databricks, 2020). Spark'ın şirketler ve araştırma enstitüleri tarafından hızla benimsenmesi nedeniyle birçok kuruluş MapReduce yerine Apache Spark'ı kullanmaya başlamıştır (Databricks, 2020).

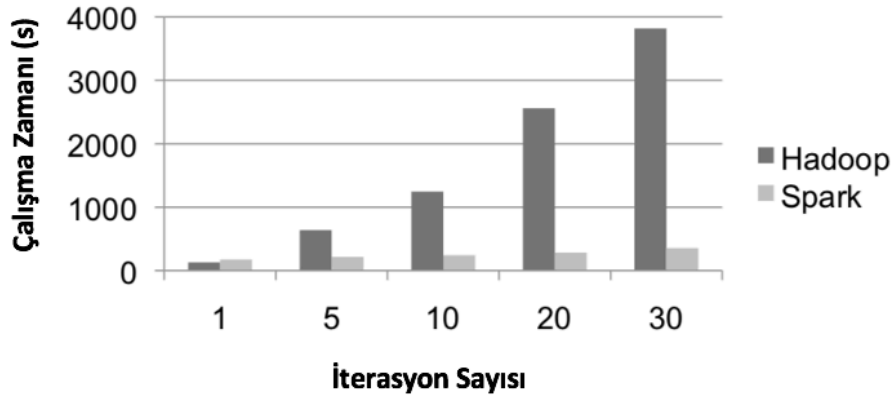
Kavramsal olarak Apache Spark Hadoop MapReduce'a benzemektedir; her ikisi de büyük veriyi işlemek için tasarlanmıştır. Her ikisi de düşük maliyetli veri işlemeyi mümkün kılmaktadır ancak Spark, Hadoop MapReduce'a göre birçok avantaj sunmaktadır.

Apache Spark'ın temel özellikleri şunlardır: (Guller,2015; Zaharia ve diğ.,2016; Zaharia ve diğ.,2010; Apache Spark 2020).

- Kullanım kolaylığı
- Hız
- Genel amaçlılık
- Ölçeklenebilirlik
- Hata dayanıklılığı

Apache Spark Hadoop'un MapReduce programlama modeline benzer bir programlama modeli sunar fakat veri RDDs adı verilen programlama modeli ile işlenir (Zaharia ve diğ., 2010). Bu model, MapReduce tarafından sağlanan programlama modelinden daha basit bir yaklaşım sağlar. Spark ile dağıtılmış bir veri işleme uygulaması geliştirmek, MapReduce ile aynı uygulamayı geliştirmekten çok daha kolaydır. Buna ek olarak, Spark, çok sayıda basma kalıp kod gerektiren Hadoop MapReduce ile karşılaştırıldığında daha kısa kod yazmanıza olanak sağlar. Hadoop MapReduce içinde 50 satır kod gerektiren bir veri işleme algoritması, Spark'ta 10 satırdan az kodda uygulanabilir (Guller, 2015).

Apache Spark, Hadoop MapReduce'dan daha hızlıdır (Şekil 2.11). Özellikle büyük veri kümelerini işlerken hız önemlidir. Bir veri işleme işi günler veya saatler alırsa, karar vermeyi yavaşlatır ve verinin değerini azaltır. Spark'ın bellek içi hesaplama yeteneği performans artışı sağlar. Bir sabit diskten veri okumak ile karşılaştırıldığında bellekten veri okunurken sıralı okuma hacmi 100 kat daha fazladır. Başka bir deyişle veri bellekten, diske göre 100 kat daha hızlı okunur. Disk ve bellek arasındaki okuma hızı farkı bir uygulama küçük bir veri kümesini okuyup işlerken fark edilmeyebilir. Ancak terabayt seviyesinde veri okunduğunda ve işlendiğinde, Giriş/Çıkış (G/Ç) gecikmesi (diskten belleğe veri yükleme süresi) genel iş yürütme süresine önemli bir katkıda bulunur.



Şekil 2.11: Hadoop ve Spark'ın Lojistik Regresyon modeli performans karşılaştırması (Zaharia ve diğ., 2010).

Spark, bir uygulamanın bellekteki veriyi işlenmek üzere önbelleğe almasına izin verir. Bu, uygulamanın disk G/Ç'sini en aza indirmesini sağlar. MapReduce tabanlı veri işleme, her işin diskten veri okuduğu, işlediği ve sonuçları diske yazdığı bir dizi işten oluşur. Dolayısıyla, MapReduce ile uygulanan karmaşık bir veri işleme uygulaması, veriyi birkaç kez okur ve diske yazar. Spark, verinin bellekten önbelleğe alınmasına izin verdiğinden, Spark ile çalıştırılan aynı uygulama veriyi diskten yalnızca bir kez okur. Veri bellekte önbelleğe alındıktan sonra sonraki her işlem doğrudan önbelleğe alınan veri üzerinde gerçekleştirilir. Böylece Spark, bir uygulamanın daha önce belirtildiği gibi genel iş yürütme süresine önemli bir katkıda bulunabilecek G/Ç gecikmesini en aza indirir.

Spark'ın Hadoop MapReduce'dan daha hızlı olmasının ikinci nedeni ise gelişmiş bir iş yürütme modeline sahip olmasıdır. Hem Spark hem de MapReduce bir işi aşamaları yönlendirilmiş döngüsüz grafiğe dönüştürür. Hadoop MapReduce, her iş için tam olarak tanımlanmış iki aşama olan map ve reduce özelliğine sahip bir graf oluşturur. MapReduce ile uygulanan karmaşık bir veri işleme algoritmasının sırayla yürütülen birden fazla işe bölünmesi gerekir. Bu tasarım Hadoop MapReduce'un herhangi bir optimizasyon yapmasını önler. Spark bir geliştiriciyi karmaşık bir veri işleme algoritmasını birden fazla işe ayırmaya zorlamaz. Spark'daki bir graf herhangi bir sayıda aşama içerebilir. Basit bir işin sadece bir aşaması olabilir, oysa karmaşık bir iş birkaç aşamadan oluşabilir. Bu, Spark'ın MapReduce ile mümkün olmayan optimizasyonları yapmasına izin verir (Guller,2015).

2.4. APACHE SPARK İLE MÜŞTERİ KAYIP ANALİZİ

Literatür incelemesi amacıyla Google Akademik (*Google Scholar*) arama motoru kullanılarak "apache spark churn prediction" anahtar kelimeleri ile arama yapılmış ve 12 makaleye erişilmiştir. Sonuçlar aşağıda özetlenmektedir.

Huang ve diğerleri (2015), Apache Spark ile telekomünikasyon veri seti kullanarak Rasgele Orman, Gradyan Artırma Makineleri, Faktorizasyon Makineleri ve Lojistik Regresyon algoritmaları ile müşteri kayıp analizi yapmıştır. Çalışma sonucunda Rasgele Orman algoritmasının diğer modellere oranla daha başarılı sonuçlar verdiğini gözlemlemiştir.

Zhang ve diğerleri (2016), sigorta şirketi veri setini kullanarak SPSS ve Spark üzerinde müşteri kayıp analizi yapmış, her yöntemi yürütme akışı, çalışma süresi, model değerlendirme ve model kesinlik bakımından karşılaşmıştır. Deneysel sonuçlar, Spark ML'in kullanımının kolay olduğunu ve büyük veri sorunlarıyla baş edebileceğini göstermiştir.

Spanoudes ve Nguyen (2017) derin öğrenme yöntemlerini kullanarak müşteri kayıp analizi yapmıştır. Veri seti erişimi için Hadoop Dağıtık Dosya Sistemi (HDFS), hesaplama aracı olarak Apache Spark kullanılmıştır.

Meher ve diğerleri (2017), ön ödemeli mobil hatlar için anahtar performans göstergelerini dikkate alarak boyut azaltma yöntemlerini kullanarak müşteri kayıp yönetimi çerçevesi önermiştir. Model eğitim ve tahmin için Apache Spark MLLib kütüphanesi kullanılmıştır.

Etaiwi ve diğerler (2017), Apache Spark makine öğrenimi kütüphanesi (MLlib) içerisinde yer alan Naif Bayes ve Destek Vektör Makinesi sınıflandırma algoritmaları arasında karşılaştırmalı bir çalışma sunmuştur. Banka müşterilerine ait 14 milyon kayıt içeren veri seti kullanılmıştır ve abonelerin satın alma davranışları tahmin edilmeye çalışılmıştır. Deneysel sonuçlara göre Naif Bayes modelinin duyarlılık (%49), kesinlik (%4) ve F-ölçütü (%7,3) performans değerlendirme ölçütlerine daha başarılı sonuçlar verdiği gözlemlenmiştir.

Chamberlain ve diğerleri (2017), küresel bir e-ticaret sitesi için müşteri yaşam süresi değeri (*Customer LifeTime Value*) tahmin sistemi önermiştir. Önerilen model, abonelerin satın alma geçmişinden oluşan veri seti ile, Apache Spark makine öğrenmesi boru hattı kullanarak Rasgele Orman regresyon algoritmasını eğiterek müşterilerin önümüzdeki 12 aydaki net harcamalarını tahmin etmektedir.

Olasehinde ve diğerleri (2018), müşteri kaybı tahmininde R paketlerinden biri olan Caret (*Classification And REgression Training)* ile Sparklyr'ı verimlilik bakımından karşılaştırmıştır. Veri seti olarak 21 nitelik alanı ve 7043 kayıttan oluşan telekom veri seti kullanmıştır. Rasgele Orman algoritması her iki yaklaşım için aynı parametreler ile çalıştırılmış, sonuç olarak Apache Spark'ın çalışma süresinin daha kısa olduğu tespit edilmiştir.

Sayed ve diğerleri (2018), Spark ML ve Spark MLlib paketleri arasında doğruluk oranı, model eğitimi ve model değerlendirme açısından karşılaştırmalı bir analiz yapmıştır. Karşılaştırmayı banka müşterileri işlem veri seti üzerinde gerçekleştirmiş ve müşterilerin ayrılma olasılığını tahmin etmek için karar ağacı algoritmasını kullanmışlardır. RDD tabanlı MLlib paketi model eğitim süresi, DataFrame tabanlı Spark ML paketi test süresi ve doğruluk oranı açısından daha iyi sonuçlar verdiğini tespit etmiştir.

Ahmad ve diğerleri (2019), Apache Spark ile telekomünikasyon veri setine Karar Ağaçları, Rasgele Orman, Gradyan Artırma Makineleri ve Aşırı Gradyan Artırma makineleri sınıflandırma algoritmalarını uygulayarak müşteri kayıp analizi yapmıştır. Sosyal ağ analizi kullanarak yeni nitelik alanları oluşturulmuş, en başarılı sınıflandırma algoritması %93,3 AUC skoru ile Aşırı Gradyan Artırma Makineleri olduğu tespit edilmiştir.

Nonghuloo ve diğerleri (2019), MongoDB ve Apache Spark kullanarak gerçek veri setine dayalı olarak müşteri kaybı tahminine yönelik deneysel bir araştırma yapmayı amaçlamıştır. Veri seti kontrat, tarife planı, kullanım, ücretlendirme ve abonelik süresi gibi bilgileri içeren 19

nitelik alanı ve 3.333 satır içermektedir. MongoDB ve Apache Spark arasında kurulan bağlantı ile veri seti veri tabanından alınmış ve Boosting algoritması uygulanmıştır. Model performansını değerlendirmek için alıcı işletim karakteristiği (ROC) kullanılmış, %74,1 AUC skoru elde etmiştir.

Deligiannis ve Argyriou (2020) mevcut bir müşterinin alım-satım sektöründeki aboneliğini sonlandırma olasılığını sürekli güncellenen bir algoritma ile tahmin eden bir prototip önermiştir. Algoritmanın sahip olduğu formüle göre aboneler risksiz, düşük riskli, orta riskli, yüksek riskli ve kritik riskli olarak sınıflandırılmaktadır. Veri seti olarak abonelik işlem geçmişi kullanılmış, algoritma paralel olarak olarak Apache Spark ile işlenmiştir.

Wassouf ve diğerleri (2020), telekomünikasyon şirketlerinin farklı değere sahip müşterileri için uygun teklifler ve hizmetler sunabilmesi için bir metodoloji önermiştir. Önerilen metodolojiye göre aboneler ilk olarak yeni yaklaşıma göre segmentlere ayrılmıştır ve her segment veya grup için sadakat seviyesi tanımlanmıştır. Demografik bilgileri ile müşteriler için en iyi davranış özellikleri seçilmiş ve Rasgele Orman, Karar Ağaçları, Gradyan Artırma Ağacı, Çok Katlamlı Algılayıcı sınıflandırma algoritmaları uygulanmış ve doğruluk açısından en başarılı sınıflandırma algoritması seçilmiştir. Hedef aboneler, uygun teklifler ve hizmetler ile optimize edilmiş ve yeni müşteriler sadakate göre sınıflandırılmıştır. Veri depolama için Hadoop Dağıtık Dosya Sistemi, veri işleme için Apache Spark kullanılmıştır.

Yüksek Öğretim Kurulu Başkanlığı Ulusal Tez Merkezinde yapılan inceleme sonucunda tez çalışmalarına yer verilmiştir. Tez taraması ‘büyük veri’ anahtar kelimesi ile tez adından yapıldığında 2003-1 Nisan 2021 yılları arasında yapılmış 155 adet, detaylı tarama alanında tez özetinden yapıldığında ise 852 adet tez kaydı gelmektedir. Tez taraması ‘apache spark’ anahtar kelimesi ile tez adından yapıldığında 2016-1 Nisan 2021 tarihleri arasında 9 adet tez, detaylı tarama alanında tez özetinden arama yapıldığında 2015-1 Nisan 2021 yılları arasında 36 adet tez kaydı gelmektedir. Ek olarak “churn” anahtar kelimesi ile tez adından arama yapıldığında 2001- 1 Nisan 2021 tarihleri arasında müşteri kayıp analizi ile ilgili yapılmış 41 adet, detaylı tarama alanında tez özetinden yapıldığında 109 adet yüksek lisans ve doktora tezi gelmektedir. Bu tez çalışmalarının hiçbiri bu tezde analiz aracı olarak kullanılan Apache Spark ile yapılmamıştır. Bu bağlamda bu tez çalışması ilk olma özelliği taşımaktadır.

Hallaç (2014) Apache Hadoop ve Apache Spark kullanarak büyük veride paralel algoritmaların çalıştırılması ve dağıtık makine öğrenmesi algoritmalarının büyük veriye uyarlanması için çeşitli uygulamalar geliştirmiştir.

Akgün (2016) Apache Spark'ın Spark Streaming kütüphanesini kullanarak akan veriye veri madenciliği yöntemleri uygulamıştır. Çalışmanın amacı Spark Streaming teknolojisini kullanarak akan veri için Destek Vektör Makineleri sınıflandırma yöntemini geliştirmek ve alınan sonuçları hali hazırda var olan Lojistik Regresyon yöntemi ile karşılaştırarak, hangi sınıflandırma yönteminin akan büyük veri sınıflandırmada daha başarılı olduğunu tespit etmektir. Çalışma sonucunda Destek Vektör Makinaları algoritmasının daha başarılı sonuçlar verdiğini gözlemlemiştir. Sınıflandırma algoritmaları Apache Spark MLlib kütüphanesi kullanılarak uygulanmıştır.

Erdem (2017) Apache Spark'ın MLlib makine öğrenmesi kütüphanesi içerisinde yer alan sınıflandırma ve kümeleme yöntemlerini büyük veri kümesine uygulamış ve Naif Bayes, K-Means ve Gaussian Mixture yöntemlerinin çalışma sürelerini farklı veri boyutları kullanarak tespit etmiştir.

Özdeş (2017) denetimli makine öğrenmesi yöntemlerinden Naif Bayes, Lojistik Regresyon ve Karar Ağaçları kullanarak ingilizce şarkı sözleri üzerinde duygu analizi gerçekleştirmiştir, Algoritmanın çalıştırılması sonucu elde edilen başarım oranları ve algoritmanın işlenmesi için geçen süre Apache Spark ve RStudio arasında karşılaştırma yapmış ve karşılaştırma sonucunda Spark'ın çok daha hızlı olduğu sonucuna varmıştır.

Gebreyesus (2018) Spark ML kütüphanesini kullanarak Twitter verisi üzerinde duygu analizi yapmış ve yüksek hacimli akan verinin sınıflandırma algoritmalarının performansı üzerindeki etkisini araştırmıştır.

Keskin (2018) bulut servisi üzerinde büyük veri araçları kullanılarak büyük veride keşifçi veri analizi, büyük veri görselleştirmesi ve büyük veride makine öğrenmesi uygulamaları gerçekleştirmiştir. SparkR kullanmıştır. Karar ağaçları, Lojistik Regresyon, Naif Bayes, Gradyan Artırma Makineleri, Rasgele Orman algoritmaları ile analizler yapmıştır.

Oğur (2018) Apache Kafka, MongoDB, Apache Spark Streaming teknolojilerini ve MLlib kütüphanesinde bulunan Lojistik Regresyon algoritmasını kullanarak, EKG verisinden hastalık

tespit eden gerçek zamanlı bir veri analitik sistemi önermiştir. Oluşturulan modellerin %100 doğruluk oranı ile çalıştığını gözlemlemiştir.

Çetin (2019) Apache Spark'ı kullanarak MovieLens veri seti üzerinde hibrit bir öneri sistemi geliştirmiştir. Hibritleştirmek için kullanılan yöntemler ise Değişken En Küçük Kareler yöntemi ile Negatif Benzerlikli İşbirlikçi Filtreleme yöntemidir. Yinelenen algoritmaların işlenmesinde Apache Spark'ın, MapReduce'a göre daha hızlı çalıştığını gözlemlemiştir.

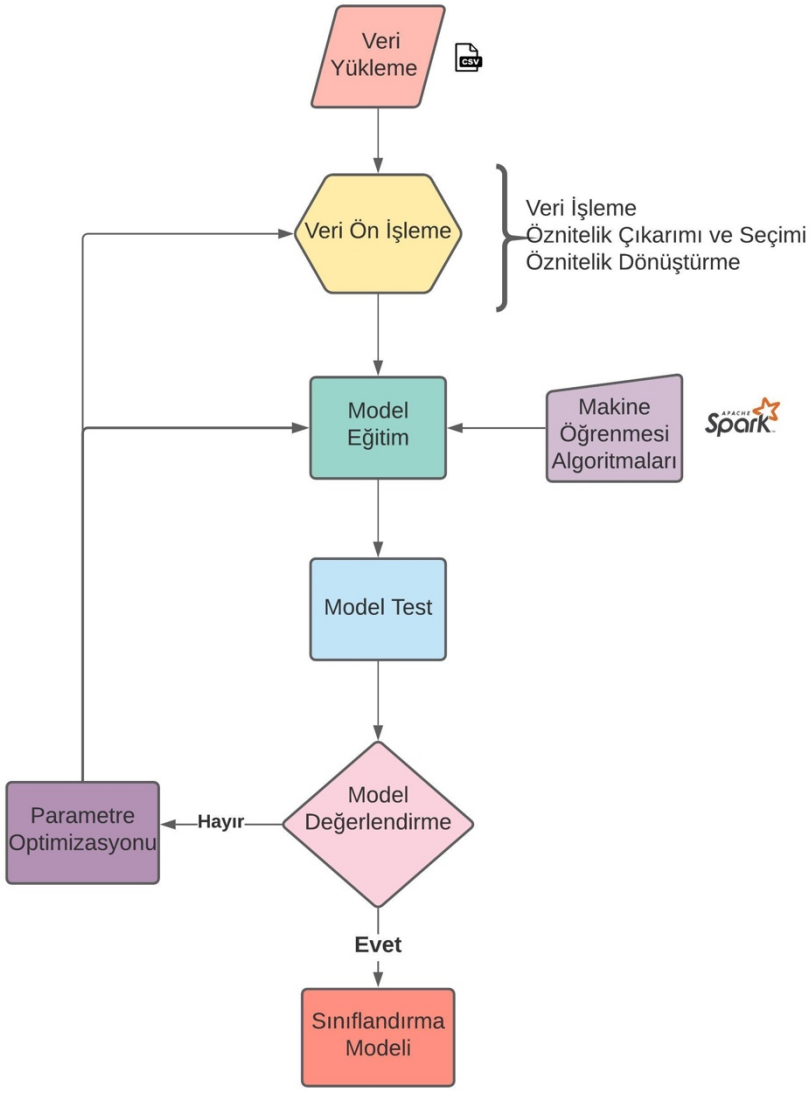
Karabay (2019) uluslararası haber ajanslarından ve kaynaklarından elde edilen ve dünyada yaşanan terör olaylarını içeren veri kümesine Apache Spark'ın makine öğrenmesi kütüphanesinde bulunan Destek Vektör Makineleri, Rastgele Orman, Karar Ağacları, Naif Bayes ve Lojistik Regresyon sınıflandırma yöntemlerini uygulamış ve yaşanan terör olayının hangi örgüt tarafından yapıldığını tespit etmeyi amaçlamıştır. En yüksek doğruluk oranını %98,2 ile K-En Yakın Komşu algoritması ile elde etmiştir.

Kurt (2019) Apache Spark'ın makine öğrenmesi kütüphanesini kullanarak ağ saldırısı tespit sistemlerine referans kaynak oluşturmak amacıyla KDD Cup'99 veri setine makine öğrenmesi algoritmalarından Lojistik Regresyon, Destek Vektör Makineleri, Naif Bayes ve Rastgele Orman yöntemlerini uygulamış, performans sonuçlarını karşılaştırmalı olarak analiz etmiştir. Çalışma sonucunda en yüksek doğruluk değerini (%99,1) Lojistik Regresyon modeli ile elde etmiş, Apache Spark'ın ağ saldırılarını tespit etmede giderek etkin hale geldiğini gözlemlemiştir.

Dolapcı (2020), Apache Spark'ın MLlib makine öğrenmesi kütüphanesini kullanarak Kaggle platformundan elde edilen gemi görüntülerine sınıflandırma işlemi yapmıştır. Sınıflandırma için kullanılan yöntemler Naif Bayes, Karar Ağaçları ve Rastgele Orman yöntemleridir. En yüksek başarımı (%99,62) Rastgele Orman modeli ile elde etmiştir

3. MALZEME VE YÖNTEM

Bu bölümde tez çalışmasında kullanılan veri setinin detayları, analiz ve algoritma geliştirmek için kullanılan programlama dili ve araçlarına, veri seti üzerinde yapılan işlemlere ve çalışmayı gerçekleştirmek için veri setine uygulanan makine öğrenmesi algoritmalarına yer verilmiştir (Şekil 3.1).



Şekil 3.1: Analiz sürecinin akış diyagramı.

Bu çalışmada algoritmalar pyspark kullanılarak geliştirilmiştir. PySpark, Apache Spark ve Python'un iş birliğini desteklemek için piyasaya sürülmüştür (Pyspark, 2020). Spark için

kullanılan bir Python uygulama programlama arayüzüdür. Veri hazırlığı için SparkSQL ve Pandas kullanılmıştır. Algoritmalar ise Spark MLlib kütüphanesi (DataFrame tabanlı Spark ML) kullanılarak çalıştırılmıştır.

Apache Spark'ı kullanmak için çeşitli yöntemler mevcuttur. Spark hem Windows hem de UNIX benzeri sistemlerde (örneğin Linux, Mac OS) çalışır ve desteklenen bir Java sürümünü çalıştıran herhangi bir platformda çalışmalıdır. Bu çalışmada çalışma ortamı olarak local bilgisayar kullanılmıştır. Modellerin eğitildiği local bilgisayarın özellikleri Tablo 3.1'de verilmiştir. Diğer bir yöntem ise Databricks platformudur. Databricks, Spark'ın bir tarayıcıyla kullanmanın bir yolu olan ve Apache Spark'ın orijinal yaratıcıları tarafından oluşturulan bir platformdur. Databricks kurulum yükünü ortadan kaldırır ve bilgi işlem gücünü doğrudan tarayıcınıza getirir. Ticari bir platform olmasına rağmen ücretsiz bir topluluk sürümü mevcuttur.[8]

Tablo 3.1: Sistem ve yazılım özellikleri.

Sistem ve Yazılım Özellikleri	
İşletim Sistemi	macOS High Sierra 10.13.6
İşlemci	2,8 GHz Intel Core i7
Bellek	8 GB 1600 MHz DDR3
Java	1.8.0_231
Scala	2.13.1
Sbt	1.3.9
Python	3.7.6
Apache Spark	Apache Spark 3.0.1
MLlib	3.0

3.1. VERİ SETİ

Bu çalışmada bir telekomünikasyon firmasından alınan 2019 yılı Mart dönemine ait müşteri verisi kullanılmıştır. Alınan örneklem verisi tamamen anonim olup abonelerin bilgilerine erişilebilir durumda değildir. Örneklem verisi kendi isteği (gönüllü) ile iptal işlemi yapmış veya başka bir servis sağlayıcıya geçmiş olan sabit geniş bant abone kümesini kapsamaktadır. Veri setindeki tüm aboneler taahhütlü aboneler olup, taahhüt bitimlerine 3 aydan az kalmış

[8] https://community.cloud.databricks.com/

abonelerden seçilmiştir. Kullanılan veri seti toplamda 50 bin abone içermektedir ve 13MB boyutundadır. Bu tezde kullanılan veri birden çok kaynak ve veri tabanından toplanmıştır. Her kaynak, veriyi yapılandırılmış, yarı yapılandırılmış (XML-JSON) veya yapılandırılmamış (CSV-Metin) olarak farklı türde dosyalarda tutar. Örnek olarak kullanım verisi bir XML dosyası içerisinde, abonelerinde çağrı merkezi aracılığı yapmış olduğu görüşmeler ise Oracle veri tabanlarında yer alan CLOB adı verilen veri tipi içerisinde tutulmaktadır.

Operatörden alınan müşteri verisi aşağıdaki bilgileri içermektedir:

1. **Abonenin demografik bilgileri:** Cinsiyet, yaş, medeni durum, eğitim durumu.
2. **Abonenin kullandığı paket bilgileri:** paket hızı, adil kullanım kotası, kota türü (Limitli ve Limitsiz), erişim tipi (yalın internet veya telefon hattı), abonelik süresi, erişim türü (fiber, ultranet vs.)
3. **Abonenin sözleşme bilgileri:** Taahhüt başlangıç ve bitiş tarihi, taahhüt kanalı, kampanya türü (ikna, mevcut müşteri, yeni kazanım vs).
4. **Abonenin kullanım bilgileri:** download ve upload bilgileri, varsa ek adil kullanım noktası, ek kota, upsell ve downsell adet bilgisi.
5. **Abonenin fatura ve ödeme bilgileri:** fatura tutarı, borçtan hat kapama adedi, otomatik ödeme talimatı, e-fatura durumu.
6. **Abonenin operatörden aldığı diğer ürün bilgileri:** Iptv, telefon, modem, uydu, cep telefonu.
7. **Abonenin müşteri deneyimi bilgileri:** fatura şikayet sayısı, internet arızası, nakil durumu, nakil süresi.
8. **Çağrı merkezi görüşme bilgileri:** arama adet, kabul ve ret adet, ulaşılamadı adet, aynı gün içinde çağrı merkezi arama adet
9. **Abonenin operatör ile etkileşim bilgileri:** İç ve dış arama adedi, yapılan aramanın ret ve kabul adedi, abonenin daha önce yaptığı iptal başvurusu adedi, ceza sorgulama durumu, abonenin ikna edilme sayısı.

Yukarıda verilen müşteri bilgileri dikkate alındığında veri setinde ilgili dönem için 50.000 satır ve 88 nitelik alanı bulunmaktadır. Niteliklerin çoğu sayısal değerlerden oluşturmaktadır ancak kategorik ve tarih veri tipinde değerler de mevcuttur (Tablo 3.2).

Tablo 3.2: Veri setindeki nitelik alanları ve veri tipleri.

	Nitelik Adı	Veri Tipi		Nitelik Adı	Veri Tipi
1	Veri Seti Dönemi	Nümerik	39	Desteklenen Hız	Kategorik
2	Dummy ID	Nümerik	40	Güncel Hız	Kategorik
3	Paket ID	Nümerik	41	Mesafe	Kategorik
4	Paket Hızı	Nümerik	42	Çağrı merkezi aracılığı ile taahhüt	Nümerik
5	Abonelik Statü	Nümerik	43	Dış Arama aracılığı ile taahhüt	Nümerik
6	Paket AKN	Nümerik	44	Borçtan kapama Sayısı	Nümerik
7	Paket AKN - 1	Kategorik	45	Borçtan Açma Sayısı	Nümerik
8	Paket Kota	Kategorik	46	Churn Başvuru Sayısı	Nümerik
9	Erişim Tipi	Kategorik	47	İşlemsiz İkna Sayısı	Nümerik
10	Abonelik Yaşı	Nümerik	48	İkna Sayısı	Nümerik
11	Erişim Türü	Kategorik	49	Hat Dondurma Sayısı	Nümerik
12	Eğitim Durumu	Kategorik	50	Yazılı Churn Başvuru Sayısı	Nümerik
13	Cinsiyet	Kategorik	51	Churn Başvuru Tarihi	Tarih
14	Medeni Durum	Kategorik	52	Ek AKN Sayısı	Nümerik
15	Abone Yaş	Nümerik	53	Aktif Ek AKN	Kategorik
16	Taahhüt Tipi	Nümerik	54	EK AKN Sayısı	Nümerik
17	Taahhüt Başlangıç Tarihi	Tarih	55	Ödenmemiş Fatura Sayısı	Nümerik
18	Taahhüt Bitiş Tarihi (AA:GG:YYYY SS:SS)	Tarih	56	Aktif Uydu	Kategorik
19	Ayrılma Tarihi	Tarih	57	Aktif Uydu Taahhüt	Kategorik
20	Taahhüt Kanalı	Kategorik	58	Uydu Taahhüt Bitiş Tarihi	Tarih
21	Taahhüt Kampanya ID	Nümerik	59	IPTV Taahhüt Bitiş Tarihi	Tarih
22	İkna Kampanyası Flag	Nümerik	60	Aktif IPTV	Kategorik
23	Taahhüt Bitiş Tarihi (AA:GG:YYYY)	Tarih	61	Aktif IPTV Taahhüt	Kategorik
24	Taahhüt Bitiş Dönem	Nümerik	62	Ceza Sorma Sayısı	Nümerik
25	Son 10 Gün Kullanım (Gün)	Nümerik	63	AKN Aşım Durumu	Nümerik
26	Son 10 Gün Kullanım (GB)	Nümerik	64	AKN Aşım Durumu 2 Ay	Nümerik
27	Upload 10 (GB)	Nümerik	65	AKN Aşım Durumu 3 Ay	Nümerik
28	Kullanım 3 Ay (GB)	Nümerik	66	Müşteri Tipi	Kategorik
29	Kullanım 2 Ay (GB)	Nümerik	67	Müşteri Tipi (ID)	Nümerik
30	Kullanım 1 Ay (GB)	Nümerik	68	Arıza Sayısı 1 Ay	Nümerik
31	Upload 1 Ay (GB)	Nümerik	69	ÇM Tarafından Yapılan Arama Sayısı 3 Ay	Nümerik
32	Fatura Tutarı 3 Ay	Nümerik	70	Ulaşılamama Sayısı 3 Ay	Nümerik
33	Fatura Tutarı 2 Ay	Nümerik	71	Arama Red Sayısı 3 Ay	Nümerik
34	Fatura Tutarı 1 Ay	Nümerik	72	Arama Kabul Sayısı 3 Ay	Nümerik
35	AKN Upsell 3 Ay	Kategorik	73	ÇM Tarafından Yapılan Arama Sayısı 6 Ay	Nümerik
36	Hız Upsell 3 Ay	Kategorik	74	Ulaşılamama Sayısı 6 Ay	Nümerik
37	Downsell 3 Ay	Kategorik	75	Arama Red Sayısı 6 Ay	Nümerik
38	Abone Şikayet Sayısı 1 Ay	Nümerik	76	Arama Kabul Sayısı 6 Ay	Nümerik

Tablo 3.2: Veri setindeki nitelik alanları ve veri tipleri (devam).

	Nitelik Adı	Veri Tipi		Nitelik Adı	Veri Tipi
77	Arıza Şikayet Sayısı 1 Ay	Nümerik	83	Fatura Talimat Durunu	Kategorik
78	Fatura Şikayet Sayısı 1 Ay	Nümerik	84	E-Fatura Durumu	Kategorik
79	Aynı Gün ÇM Arama Sayısı	Nümerik	85	IPTV Taahhüt Durum	Kategorik
80	Nakil Başvurusu Sayısı	Nümerik	86	THK Taahhüt Durum	Kategorik
81	Tamamlanan Nakil Başvuru Sayısı	Nümerik	87	Modem Taahhüt Durum	Kategorik
82	Maksimum Nakil Süresi	Nümerik	88	Churn Durum	Kategorik

Veri setindeki müşteri ayrılma oranı %20'dir. Veri setinden de anlaşılacağı üzere telekomünikasyon abonesinin ayrılma davranışı az görülen bir davranıştır. Veri seti, dengesiz sınıf dağılımı açısından incelendiğinde dengesiz bir dağılım olduğu gözlenmektedir.

3.2. VERİ ÖN İŞLEME

Veri setinin analizlere uygun hale gelebilmesi için bazı ön işlemlerden geçmesi gerekmektedir. Veri seti operatörden temin edilirken eksik veri barındırmamasına dikkat edilmiştir.

- Taahhüt bitiş tarihi (GG.AA.YYYY SS.SS) ve taahhüt bitiş tarihi (GG.AA.YYYY) nitelikleri alanları mükerrer bilgi içerdiğinden veri setinden çıkarılmıştır. Onun yerine "Taahhüt Bitiş Dönem" kullanılmıştır.
- Fatura Tutarı 3 Ay, Fatura Tutarı 2 Ay, Fatura Tutarı 2 ay sütunları birleştirilip "fat_net" kolonu eklenmiştir.
- Taahhüt Başlangıç Tarihi ve Churn Tarihi nitelik alanları ihtiyaç olmaması sebebi ile veri setinden çıkarılmıştır.
- Veri Seti Dönemi, Taahhüt Tipi, Dummy ID, Paket ID, Taahhüt Kampanya ID, Müşteri Tipi (ID), Abone Statu alanları abonelerin kullandıkları paketleri tanımlayan alanlar olması ve tahmin modeline etkileri olmaması sebebi ile veri setinden çıkarılmıştır.
- Veri setinde bulunan ve müşteri beyanına dayanan eğitim durumu niteliği birçok müşteri için boş değerler içerdiğinden modellemenin daha isabetli olması adına çıkarılmıştır.
- Paket AKN - 1 sütununda paket hızı bilgisi kategorik, Paket AKN sütunu ise paket hızı bilgisini nümerik olarak tutmaktadır. Bu iki alan aynı bilgiyi farklı formatta tuttuğundan Paket AKN - 1 veri setinden çıkarılmıştır.

- Churn Başvuru Tarih, Uydu Taahhüt Bitiş Tarihi, Iptv Taahhüt Bitiş Tarihi nitelik alanlarındaki tarih formatı “yıl-ay” olacak şekilde nümerik değer yapılmıştır.

Nümerik değer alan nitelik alanlarının özeti Tablo 3.3 verilmiştir.

Tablo 3.3: Veri seti özeti.

summary	count	mean	stddev	min	0,250	0,500	0,750	max
paket_hızı	50000	18479,145	9014,922	4096	16384	16384	24576	102400
paket_akn	50000	1,495	6,449	0	0	0	0	40
abonelik_yaşı	50000	1618,388	1226,643	275	691	1215	2188	5785
abone_yaş	50000	43,604	13,507	19	33	42	53	96
taah_bitiş	50000	201904,733	0,936	201903	201904	201905	201906	201906
kullanım_10_gün	50000	9,054	2,545	0	10	10	10	10
kullanım_10_gün_gb	50000	32,931	34,116	0	10,881	23,947	44,323	873,084
upload_10_gün	50000	0,078	0,519	0	0	0	0	40,521
kullanım_3_ay	50000	103,210	98,157	0	37,868	79,397	140,166	2775,796
Kullanım_2_ay	50000	115,042	107,881	0	41,381	86,428	158,577	1494,986
kullanım_1_ay	50000	101,805	96,360	0	37,424	77,057	138,349	2376,270
upload_gb1ay	50000	0,237	1,349	0	0	0	0	73,307
fat_net	50000	72,753	40,864	0	58,420	66,330	76,580	1858,410
abone_şikayet1ay	50000	0,005	0,080	0	0	0	0	3
teknik_şikayet_1ay	50000	0,052	0,301	0	0	0	0	8
fatura_şikayet1ay	50000	0,002	0,051	0	0	0	0	2
çm_arama_sayısı	50000	0,075	0,329	0	0	0	0	8
nakil_başvuru_sayısı	50000	0,025	0,192	0	0	0	0	12
tamamlanan_nakil_sayısı	50000	0,020	0,145	0	0	0	0	4
max_süre_nakil	50000	0,090	1,232	0	0	0	0	75
borckapama_sayısı	50000	0,058	0,256	0	0	0	0	4
borcacma_sayısı	50000	0,065	0,260	0	0	0	0	2
churn_başvuru_sayısı	50000	0,043	0,238	0	0	0	0	6
işlemsiz_ikna_sayısı	50000	0,033	0,204	0	0	0	0	6
ikna_sayısı	50000	0,038	0,222	0	0	0	0	6
hatdondurma_sayısı	50000	0,000	0,010	0	0	0	0	1
yazılı_başvuru_sayısı	50000	0,007	0,088	0	0	0	0	3
churn_başvuru_tarih	50000	7397,438	37931,695	0	0	0	0	201903
ek_akn_sayısı	50000	0,009	0,098	0	0	0	0	4
ödenmemiş_fatura_sayısı	50000	0,253	0,461	0	0	0	0	15
uydu_taahhüt_bitiştarihi	50000	10231,392	44291,192	0	0	0	0	202105
iptv_taahhüt_bitiştarihi	50000	13300,404	50091,532	0	0	0	0	202103
ceza_sorma_sayısı	50000	0,014	0,118	0	0	0	0	1
ariza_sayısı1ay	50000	0,043	0,250	0	0	0	0	6
arama_sayısı3ay	50000	0,800	0,832	0	0	1	1	5

Tablo 3.3: Veri seti özeti (devam).

summary	count	mean	stddev	min	25%	50%	75%	max
ulaşılamama_sayısı3ay	50000	0,278	0,543	0	0	0	0	4
ret_sayısı3ay	50000	0,420	0,672	0	0	0	1	4
kabul_sayısı3ay	50000	0,011	0,103	0	0	0	0	3
arama_sayısı6ay	50000	1,659	1,503	0	0	2	3	7
ulaşılamama_sayısı6ay	50000	0,616	0,998	0	0	0	1	6
ret_sayısı6ay	50000	0,841	1,147	0	0	0	2	6
kabul_sayısı6ay	50000	0,027	0,167	0	0	0	0	3

Kategorik sütunlar için StringIndexer ve OneHotEncoder fonksiyonları ile değişken dönüşüm işlemleri gerçekleştirilmiştir.

StringIndexer fonksiyonunun arkasındaki konsept basitçe her kategorik değişkenin bir nümerik değer ile değiştirilmesidir. Sonrasında dönüştürülmüş değerler makine öğrenmesi modellerinde girdi olarak kullanılmıştır.

OneHotEncoder, kategorik bir değişkeni tüm özellik değerleri kümesi arasından belirli bir özellik değerinin varlığını gösteren en fazla tek bir değerle, bir ikili vektöre eşler. Bu dönüşüm işlemi Lojistik Regresyon gibi sürekli değişkenlerin kullanılmasını bekleyen algoritmalarda kategorik değişkenlerin kullanılmasına izin verir. String tipi giriş verisi için önce StringIndexer kullanılarak dönüşüm işlemi yapılmıştır.

Apache Spark'ın makine öğrenmesi kütüphanesinde modeller çalıştırılmaya başlanmadan önce VectorAssembler fonksiyonu ile birlikte birleştirme ve dönüştürme işlemi yapılır. VectorAssembler, belirli bir sütun listesini tek bir vektör sütununda birleştiren bir dönüştürücüdür. Çeşitli makine öğrenmesi modellerini eğitmek için, farklı özellik dönüştürücüleri tarafından üretilen ham nitelik alanlarını, tek bir nitelik vektöründe birleştirmek için kullanılır. VectorAssembler sayısal veri tiplerini, mantıksal ve vektör tipindeki veri tiplerini kabul eder. Her satırda, giriş sütunlarının değerleri belirtilen sırada bir vektör olarak birleştirilmiştir.

3.3. MODELLEME

Tez çalışması kapsamında gerçekleştirilen analizlerde veri setine Lojistik Regresyon, Karar Ağaçları, Rastgele Orman ve Gradyan Artırma Makineleri algoritmaları uygulanmıştır.

Modeller oluşturulurken farklı doğrulama yöntemleri kullanılmıştır. Sınama Seti (*Holdout*) yaklaşımında eğitim ve test seti olarak ayırma işlemi %70-%30 ve %80-%20 oranları ile k-katlı çapraz doğrulama yöntemi (*K-fold Cross Validation*) ise 5 ve 10 değerleri ile gerçekleştirilmiştir. Çapraz doğrulama yöntemi kullanıldığında performans değerlendirme ölçüleri, tüm katlardan elde edilen değerlerin ortalaması alınarak elde edilmiştir.

Kullanılan algoritmalar çeşitli parametre değerlerine ihtiyaç duymaktadır. Parametrelerde meydana gelecek değişiklikler analiz sonuçlarını da etkileyebileceğinden, her bir model için parametre değerlerinin belirlenmesi gerekmektedir.

Parametre ayarlama işleminden önce, algoritmaların varsayılan parametre değerleri ile oluşturulan modellere ait performans değerlendirme sonuçları Tablo 3.4'de verilmiştir.

Tablo 3.4: Varsayılan parametre değerleri ile elde edilen sonuçlar.

Performans Değerlendirme Yöntemi	Model	%70 Eğitim %30 Test	%80 Eğitim %20 Test	Çapraz Doğrulama (5 Kat)	Çapraz Doğrulama (10 Kat)
Doğruluk	Karar Ağaçları	0,8350	0,8415	0,8357	0,8304
	Lojistik Regresyon	0,7456	0,6904	0,1979	0,7995
	Random Forests	0,7324	0,6527	0,7376	0,7785
	Gradyan Artırma Makineleri	0,8495	0,8473	0,8562	0,8585
Hata	Karar Ağaçları	0,1650	0,1585	0,1438	0,1415
	Lojistik Regresyon	0,2544	0,3096	0,8562	1,0000
	Random Forests	0,2676	0,3473	0,1438	0,0000
	Gradyan Artırma Makineleri	0,1505	0,1527	0,8562	1,0000

Tablo 3.4: Varsayılan parametre değerleri ile elde edilen sonuçlar (devam).

Performans Değerlendirme Yöntemi	Model	%70 Eğitim %30 Test	%80 Eğitim %20 Test	Çapraz Doğrulama (5 Kat)	Çapraz Doğrulama (10 Kat)
F - Ölçütü	Karar Ağaçları	0,5714	0,5788	0,5584	0,5299
	Lojistik Regresyon	0,4245	0,4234	0,3304	0,0000
	Random Forests	0,4628	0,4665	0,4782	0,5920
	Gradyan Artırma Makineleri	0,6129	0,6033	0,6102	0,6156
AUC	Karar Ağaçları	0,7667	0,7572	0,7736	0,7782
	Lojistik Regresyon	0,7057	0,7026	0,7036	0,0000
	Random Forests	0,7625	0,7643	0,7647	0,7864
	Gradyan Artırma Makineleri	0,8936	0,8901	0,8966	0,8974
Duyarlılık	Karar Ağaçları	0,5487	0,5539	0,5202	0,4752
	Lojistik Regresyon	0,4681	0,5780	1,0000	0,0000
	Random Forests	0,5752	0,7724	0,5989	0,7895
	Gradyan Artırma Makineleri	0,5915	0,5740	0,5624	0,5775
Kesinlik	Karar Ağaçları	0,5960	0,6061	0,6026	0,5990
	Lojistik Regresyon	0,3883	0,3340	0,1979	0,0000
	Random Forests	0,3872	0,3342	0,3980	0,4735
	Gradyan Artırma Makineleri	0,6359	0,6358	0,6669	0,6591
Özgünlük	Karar Ağaçları	0,9067	0,9119	0,9144	0,9199
	Lojistik Regresyon	0,8152	0,7179	0,0000	0,0000
	Random Forests	0,7718	0,6234	0,7724	0,7757
	Gradyan Artırma Makineleri	0,9146	0,9166	0,9297	0,9271

Kullanılan ilk makine öğrenmesi algoritması lojistik regresyondur. Lojistik regresyonda en önemli parametre değeri model karmaşıklığını belirleyen parametredir. Bu parametre MLlib'de regParam parametresidir. Bu parametre λ (*lambda*) değerine karşılık gelir. Yüksek regParam değerleri daha az genellenebilir olmaya karşılık gelir. Başka bir deyişle, regParam parametresi için yüksek bir değer kullandığımızda, eğitim setine mümkün olan en iyi şekilde uymaya çalışır ve overfittinge sebep olur bu da yeni bir veri setinde modelin genelleme yeteneğinin düşük olmasına sebep olur. regParam parametresinin düşük değerler alması durumunda ise modeller sıfıra yakın bir katsayı vektörü bulmaya daha fazla önem verir ve underfittinge neden olur. Underfitting durumunda model ne mevcut eğitim setine ne de yeni bir veri setine genellebilir durumdadır. Aynı anda hem overfitting hem de underfittingden kaçınmak iyi regParam değerine iyi bir değer atanmalıdır. Varsayılan değeri 0'dır. regParam için şu değerler denenmiştir: {0.01, 0.01, 0.1,1, 10 }, en iyi sonuç 1 değerinde alınmıştır.

Karar Ağaçları algoritmasını için parametre ayarı yapmak için birçok yöntem vardır, Spark uygulamasının desteklediği hiper parametrelerden en önemlileri maksimum derinlik ve minimum örnek sayısı parametreleridir. Maksimum derinlik (*maxDepth*) parametresi için daha derin ağaçlar daha ifade edicidir ve daha yüksek doğruluğa izin verir, ancak aynı zamanda eğitilmeleri daha çok zaman alır ve overfitting ihtimali daha olasıdır. Ağacın derinliğini sınırlamak overfittingi azaltır. Varsayılan değeri 5'tir, en iyi sonuç 7 değeri ile alınmıştır. Bir yaprak düğümünde olması gereken minimum örnek sayısı (*minInstancesPerNode*) ise bir yaprak düğümünü sonlandırmak için gereken minimum örnek sayısını ifade eder. Overfittingi kontrol etmek için kullanılır. Varsayılan değeri 1'dir, 1'den büyük herhangi bir değer alabilir. En en iyi sonucun varsayılan değer ile alınmıştır.

Rastgele Orman ve Gradyan Arttırma Makineleri algoritmaları genellikle Karar Ağacları algoritmasının sahip olduğu hiper parametrelere sahiptir. İki algoritmanın kendilerine ait hiper parametleri de mevcuttur. Rastgele Orman çok güçlüdür, genellikle yoğun parametre ayarı gerekmeden iyi çalışır ve verinin ölçeklendirilmesini gerektirmez. Gradyan Artırma Makineleri genelde parametre ayarlarına Rastgele Ormandan biraz daha duyarlıdır, parametreler doğru ayarlanmışsa daha iyi doğruluk sağlayabilir.

Rastgele Orman algoritması ile model oluşturmak için, inşa edilecek ağaç sayısına karar verilmesi gerekir. numTrees parametresi ile eğitilecek ağaç sayısı belirlenir, varsayılan değeri

20’dir. En iyi sonuç 50 değeri ile alınmıştır, bu değerden sonra birbirine yakın değerler alınmakta fakat modelin çalışma süresi uzamaktadır. Minimum örnek sayısı (*minInstancesPerNode*) için 3 ve maksimum derinlik (*maxDepth)* için 20 değeri ile en başarılı sonuçlar elde edilmiştir.

Gradyan Artırma Makineleri algoritması için optimize edilmesi gereken parametre öğrenme oranı (*stepSize*) ve toplam yineleme sayısı (*maxIter*) parametreleridir. Öğrenme oranı her ağacın topluluk tahminine yaptığı katkı miktarını kontrol eder, 0 ile 1 arasında değerler alır. Öğrenme oranı için 0,1 değeri kullanılmıştır. Toplam yineleme sayısının varsayılan değeri 20’dir, en başarılı sonuç 200 değeri ile elde edilmiştir. Minimum örnek sayısı (*minInstancesPerNode*) için ise 9 değeri kullanılmıştır.

Parametre ayarlama işleminde kullanılan parametreler ve değerleri Tablo 3.5’de verilmiştir.

Tablo 3.5: Parametre ayarlama işleminde kullanılan parametreler.

Sınıflandırma Algoritması	Parametre	Değer
Lojistik Regresyon	RegParam	1
Karar Ağaçları	maxDepth	7
	minInstancesPerNode	1
Rasgele Orman	numTrees	50
	minInstancesPerNode	3
	maxDepth	20
Gradyan Artırma Makineleri	stepSize	0,1
	maxIter	200
	maxDepth	5
	minInstancesPerNode	9

İlk denemelerde Lojistik Regresyon ve Rastgele Orman algoritmalarının oldukça başarısız sonuçlar verdiği gözlenmiştir. Lojistik regresyon algoritmasının doğrusal olmayan veri setlerinde zayıf performans gösterdiği bilinmektedir. Bunun sebebi lojistik regresyonun sahip olduğu, bağımlı hedef değişken ile bağımsız değişkenler arasında doğrusallık olduğu varsayımıdır. Gerçek dünyada veri nadiren doğrusaldır ve churn tahmini de doğrusal olmayan problemler arasındadır. Diğer bir sebep ise veri setinin sahip olduğu dengesiz sınıf dağılımıdır. Bu problemin üstesinden gelmek ve modelleri çalışır hale getirebilmek için sınıf ağırlıklarını ayarlama yoluna gidilmiştir. Sınıflara ağırlık vererek sınıf dengesizliği giderebilir. Varsayılan olarak bir makine algoritması, her sınıfa eşit olarak yaklaşır. Sınıflara farklı ağırlıklar vererek

onları farklı şekilde ele almasını sağlayabilir ve dengesizliği gidermek için bu ağırlıkları kullanılabilir. Sınıf ağırlığını ayarlama özelliği Apache Spark 1.6 sürümünden itibaren sadece Lojistik Regresyon için mevcuttu, ancak Spark 3.0'dan itibaren bu özellik tüm sınıflandırma modellerinde kullanılabilir hale getirilmiştir. Dolayısıyla 3.0 sürümünden itibaren, bu özelliği dengesizliği gidermek için herhangi bir sınıflandırma algoritmasıyla birlikte kullanabilir. Kullanılan veri seti için, Lojistik Regresyon ve Rastgele Orman algoritmalarının churn=1 sınıfına 4 kat oranda ağırlık vermesi sağlanmıştır.

3.4. PERFORMANS DEĞERLENDİRME

Modellerin tahmin başarısının değerlendirilmesi için Doğruluk Oranı, Hata Oranı, Kesinlik, Duyarlılık, Özgünlük, F Ölçütü ve AUC skoru kullanılmıştır.

Karmaşıklık matrisi üzerinde çeşitli hesaplamalar yapılarak çeşitli ölçütlerle modellerin tahmin performansı hesaplanabilir. Tablo 3.6'da karmaşıklık matrisi verilmiştir.

Karmaşıklık matrisinin doğru pozitif, yanlış pozitif, yanlış negatif, yanlış pozitif ve doğru negatif olmak üzere 4 adet çıktısı mevcuttur.

Tablo 3.6: Karmaşıklık Matrisi.

		Tahmin Edilen Sınıflar	
		Pozitif	Negatif
Gerçek Sınıflar	Pozitif	Doğru Pozitif *(DP)*	Yanlış Negatif *(YN)*
	Negatif	Yanlış Pozitif *(YP)*	Doğru Negatif *(DN)*

Doğru Pozitif (*True positive - TP*): Gerçekte pozitif sınıfına ait bir örnek, sınıflandırıcı tarafından pozitif olarak tahmin etmiştir.

Doğru Negatif (*True negative - TN*): Gerçekte negatif sınıfına ait bir örnek, sınıflandırıcı tarafından negatif olarak tahmin edilmiştir.

Yanlış Pozitif (False positive - FP): Gerçekte negatif sınıfına ait bir örnek sınıflandırıcı tarafından pozitif olarak tahmin edilmiştir.

Yanlış Negatif (*False negative - FN*): Gerçekte pozitif sınıfına ait bir örnek sınıflandırıcı tarafından negatif olarak tahmin edilmiştir.

Model performansını değerlendirmek için yaygın olarak kullanılan yöntemlere aşağıda yer verilmiştir (Özkan, 2020).

Doğruluk Oranı (*Accuracy*): Belirli bir örnek kümesi üzerinde sınıflandırıcı tarafından yapılan doğru sınıflandırmaların sıklığıdır. Sınıflandırma doğruluğu, doğru sınıflandırma sayısının, toplam örnek sayısına bölünmesiyle hesaplanır.

$$Doğruluk = \frac{DP+DN}{DP+YP+YN+DN} \tag{3.1}$$

Hata Oranı (*Error Rate*): Hata oranı, belirli bir örnek kümesi üzerinde sınıflandırıcı tarafından yapılan hataların sıklığıdır. Yanlış tahmin edilen örnek sayısının tüm örnek sayısına oranlanması ile elde edilir. Doğruluk Oranının 1'den çıkarılması ile elde edilir.

$$Hata\ Oranı = \frac{YP+YN}{DP+YP+YN+DN} = 1 - Doğruluk \tag{3.2}$$

F Ölçütü: Duyarlılık ve kesinlik ölçütlerini birleştirir, iki ölçütün harmonik ortalaması F ölçütüdür. F ölçütü modelin önemli sayıda örneği kaçırmadan, kaç örneği doğru şekilde tanımladığını belirler.

$$F = (2)\frac{(Duyarlılık).(Kesinlik)}{Duyarlılık+Kesinlik} \tag{3.3}$$

F ölçütünü hesaplamak için gerekli olan Duyarlılık ve Kesinlik ölçütleri aşağıdaki gibi hesaplanır.

Duyarlılık (*Recall, Sensivity*): Gerçekte doğru şekilde sınıflandırılan pozitif örnek oranını belirtir. Doğru Pozitif Oranı (TPR) olarak da bilinir. Tüm pozitif örneklerin hangi oranda sınıflandırıcı tarafından doğru bir şekilde pozitif olarak tahmin edildiğini belirtir.

$$Kesinlik = \frac{DP}{DP+YN} \tag{3.4}$$

Kesinlik (*Precision*): Pozitif olarak tahmin edilen sınıfın, gerçekte hangi oranda pozitif olduğunu belirtir.

$$Duyarlılık = \frac{DP}{DP+YP} \quad (3.5)$$

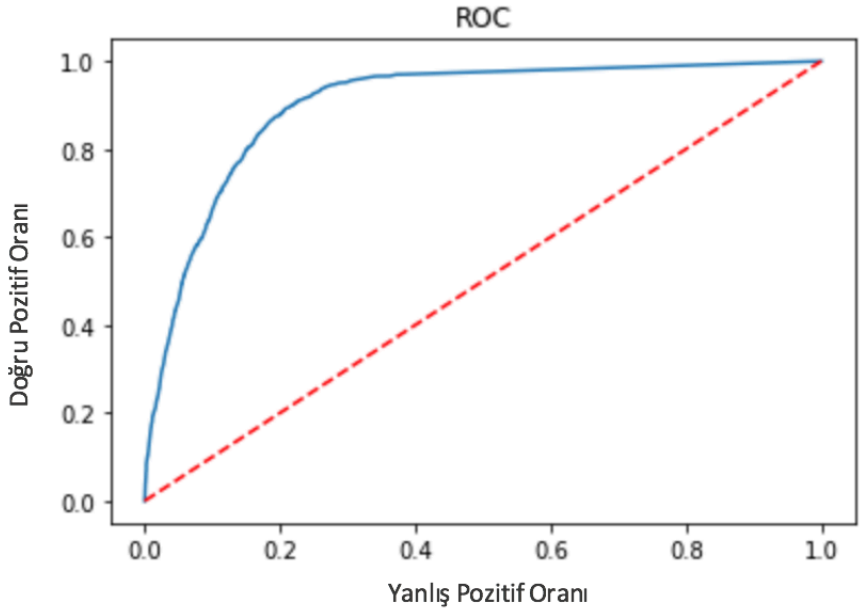
Özgünlük (*Specifity*): Tüm negatif örneklerin hangi oranda sınıflandırıcı tarafından doğru şekilde negatif olarak tahmin edildiğini belirtir. Aynı zamanda Gerçek Negatif Oranı (TNR) olarak da bilinir.

$$\text{Özgünlük} = \frac{DN}{DN+YP} \quad (3.6)$$

Spark ML kütüphanesi içerisinde bulunan İkili Sınıflandırma ve Çok Sınıflı Sınıflandırma değerlendirme metrikleri, F ölçütünü hesaplarken ağırlıklı duyarlılık ve kesinlik değerlerini kullanır (Apache Spark MLLib Guide, 2020). Bu formüle göre hesaplanan F ölçütü, denklem (3.3)'de ile hesaplanan F ölçütü değerinden farklı sonuç vermektedir. F ölçütünde standardı sağlamak için denklem (3.3) kullanılmıştır.

$$F(\beta) = (1+\beta^2).\left(\frac{(Duyarlılık).(Kesinlik)}{\beta^2(Duyarlılık)+(Kesinlik)}\right) \quad (3.7)$$

Alıcı İşlem Karekteristiği (*Receiver Operating Characteristic - Area Under the Receiver Operating Characteristic Curve- AUC/ROC*): ROC eğrisi sınıflandırma problemlerinde kullanılan bir performans ölçütüdür. Doğru pozitif oranı (*TPR*) ve yanlış pozitif oranı (*FPR*) göz önünde bulundurularak X ve Y ekenlerinde bir eğri oluşturulur ve bu eğriye ROC eğrisi denir. ROC bir olasılık eğrisidir ve eğrinin altında kalan alana *Area Under Curve* denir, bir modelin sınıf niteliklerini ne kadar iyi ayırabildiğinin derecesini veya ölçüsünü temsil eder. Bu alanın büyük olması oluşturulan makine öğrenmesi modelinin başarılı olduğu anlamına gelir. AUC ne kadar yüksek olursa, model 0 sınıfını 0 olarak ve 1 sınıfını 1 olarak tahmin etmede o kadar iyidir demek olur.



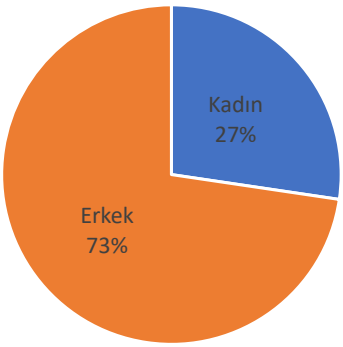
Şekil 3.2: ROC eğrisi.

AUC skorunu elde etmek için MLlib kütüphanesi içersinde yer alan ve modelleri değerlendirmek için varsayılan ölçütü *areaUnderROC* olan *BinaryClassificationEvaluator* kullanılmıştır. Bu fonksiyondan elde edilen ROC eğrisi Şekil 3.1'de verilmiştir. Kırmızı kesik çizgi, rastgele bir tahmin için ROC eğrisini temsil eder ve her bir durum için pozitif veya negatif rasgele tahmin değeri için TPR ve FPR eğrisini gösterir ve bu çizgi için eğri altındaki alan 0,5'tir.

4. BULGULAR

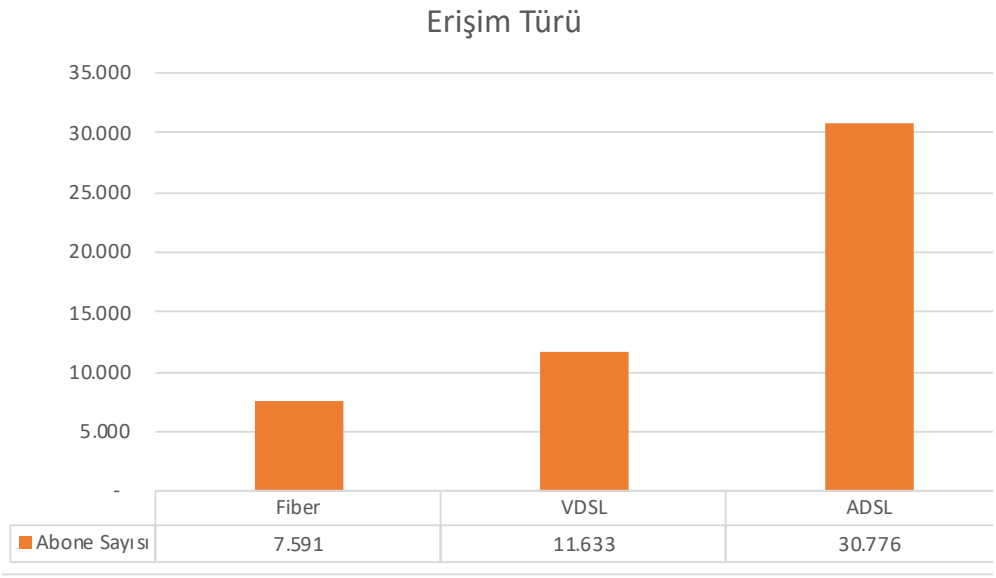
Bu bölümde veri seti ile ilgili birtakım istatistiklere ve oluşturulan modellere ait analizlerin belirtilen performans değerlendirme ölçütlerine göre karşılaştırmalı sonuçlarına yer verilmiştir.

Veri setindeki cinsiyet dağılımına bakıldığında abonelerin %73'ünün erkek (n=36330), %27'sinin kadın (n= 36330) olduğu görülmektedir (Şekil 4.1).



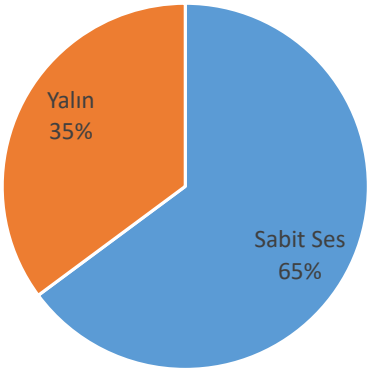
Şekil 4.1: Veri setindeki abonelerin cinsiyet dağılımı.

Abonelerin kullandıkları internet paketlerinin erişim türü Şekil 4.2'de verilmiştir. 30.776 abonenin (%61,55) ADSL altyapıda, 11.633 abonenin (%23,27) VDSL ve 7591 abonenin (%15,18) ise fiber alt yapıda olduğu görülmektedir. Fiber altyapıdaki aboneler eve kadar fiber (FTTH) veya binaya kadar (FTTB) fiber alt yapıya sahiptir.



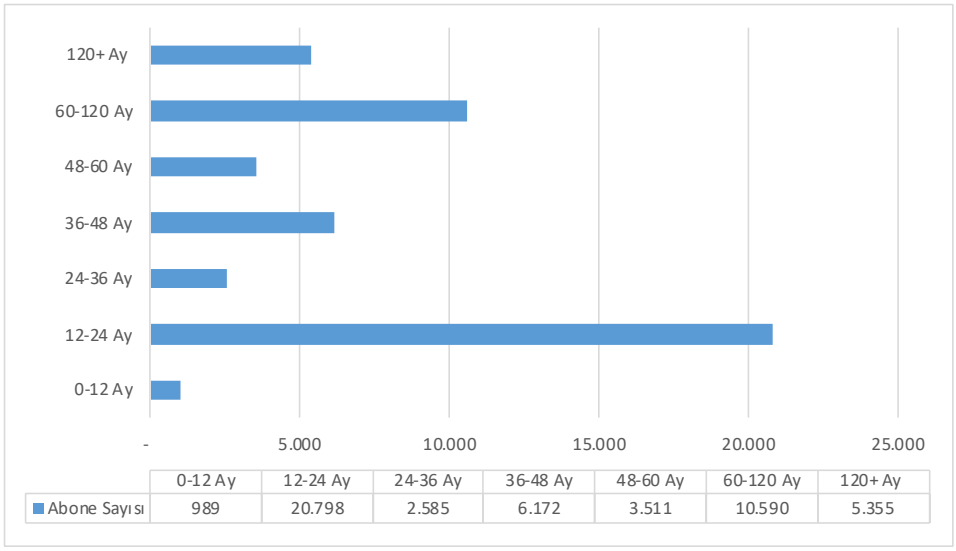
Şekil 4.2: Veri setindeki abonelerin paket erişim türü.

Abonelerden %35'i yalın internet (n=17.575), %65'i ise bir sabit ses (n=32.425) hizmetine bağlı olarak internet kullanmaktadır (Şekil 4.3).



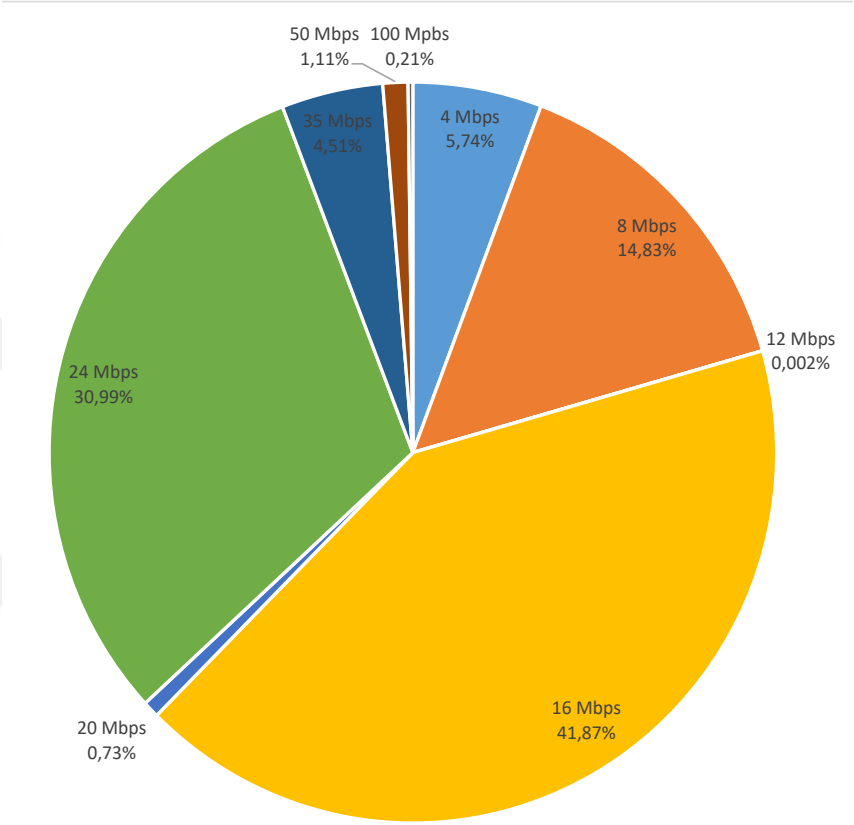
Şekil 4.3: Erişim tipi dağılımı.

Veri setindeki abonelerin abonelik yaşları Şekil 4.4'de verilmiştir. Abonelik yaşına göre abonelerin dağılımı incelendiğinde en çok abonenin (%41,60) 12 ile 24 ayı kapsayan süredir operatörden hizmet aldığı görülmektedir. Diğer abonelerin, abonelik yaşlarına göre dağılımı; %21,18'inin 60 ile 120 ay, %12,34'ünün 36 ile 48 ay, %7,02'sinin 48 ile 60 ay, %5,17'sinin 24 ile 36 ay olmak üzere operatörden hizmet aldığı görülmektedir. 10 yıl ve üzeri süredir hizmet alan 5.355 abone (%10,71) vardır. Veri setindeki abonelerin %1,98'inin abonelik yaşı 1 yıldan azdır.



Şekil 4.4: Abonelik yaşları.

Paket hızlarının dağılımı Şekil 4.5'de verilmiştir. Buna göre aboneler tarafından en çok tercih edilen ilk 3 paket hızının %41,87 ile 16 Mbps (n=20.937), %30,99 ile 24 Mbps (n=15.497) ve %14,83 ile 8 Mbps (n=7.416) olduğu görülmektedir. Diğer paketler hızlarının ise %5,74 ile 4 Mbps (n=2.868), %4,51 ile 35 Mbps (n=2256), %1,11 ile 50 Mbps (n=557), %0,73 ile 20 Mbps (n=363), %0,21 ile 100 Mbps (n=105) ve %0,021 ile 12 Mbps (n=1) olduğu görülmektedir.



Şekil 4.5: Paket hızlarının oransal dağılımı.

Oluşturulan modeller arasında müşteri ayrılma eğilimini en iyi tahmin eden sınıflandırma algoritmasının Gradyan Artırma Makinaları olduğu görülmüştür. Sınama seti (%80 eğitim-%20 test ve %70 eğitim-%30 test) ve k-katlı çapraz doğrulama (k=5 ve k=10) yöntemlerinin ikisinde de doğruluk, F ölçüsü, AUC skoru performans değerlendirme ölçütlerine göre en başarılı sonuçları Gradyan Artırma Makineleri algoritması vermiştir. Bu model için veri seti %70 eğitim ve %30 test verisi olarak bölündüğünde doğruluk oranı %85,70, F ölçüsü %61,57, AUC skoru, %90,57 olarak elde edilmiştir. Veri seti %80 eğitim ve %20 test verisi olarak bölündüğünde ise doğruluk oranı %86,16, F ölçüsü %62,07, AUC skoru %90,75 olarak elde edilmiştir. K katlı çapraz doğrulama yöntemi kullanıldığında ise k=5 olarak seçildiğinde en yüksek doğruluk

değeri %86,34, en yüksek F ölçüsü %63,45, en yüksek AUC skoru %90,82 olarak elde edilmiştir. K değerinin 10 olarak seçilmesi durumunda ise doğruluk %85,49, F ölçüsü %63,45, AUC skoru %90,74 olarak elde edilmiştir.

GBM modeli için duyarlılık ve kesinlik değerleri incelediğinde, duyarlılık değeri için gerçekte churn eğiliminde olan müşteriler içerisinde müşterilerin bu durumunu doğru tahmin etme oranı %60,04 olarak en iyi şekilde k- katlı çapraz doğrulama yöntemi ile k=10 seçilerek elde edilmiştir. Kesinlik ölçüsü incelendiğinde ise ayrılma eğilimli olarak tahmin edilen müşterilerin %68,22'sinin gerçekte de ayrılma eğilimli olduğu tahmin oranı ile k-katlı çapraz doğrulama yöntemi ile k=5 seçilerek elde edilmiştir.

Modellere ait doğruluk, hata, F ölçüsü, duyarlılık, kesinlik ve özgünlük değerleri her modele için oluşturulan karmaşıklık matrisinden hesaplanmıştır. Karmaşıklık matrisi (*confusion matrix*) veri setinin %70 eğitim, %30 test verisi ve %80 eğitim, %20 test verisi şeklinde bölünmesi ile oluşturulmuştur. Çapraz doğrulama yöntemi kullanıldığında performans değerleri her kattan gelen değerlerin ortalaması alınarak hesaplandığından, karmaşıklık matrisi oluşturulamamıştır.

Karar Ağaçları modeli için veri setinin %70 eğitim, %30 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.1'de verilmiştir.

Tablo 4.1: Karar ağaçları modeli %70 eğitim, %30 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1604	1423
	Negatif	970	11105

Karar Ağaçları modeli için veri setinin %80 eğitim, %20 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.2'de verilmiştir.

Tablo 4.2: Karar ağaçları modeli %80 eğitim, %20 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1145	841
	Negatif	763	7351

Lojistik regresyon modeli için veri setinin %70 eğitim, %30 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.3'te verilmiştir.

Tablo 4.3: Lojistik regresyon modeli %70 eğitim, %30 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1179	1848
	Negatif	1652	10423

Lojistik regresyon modeli için veri setinin %80 eğitim, %20 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.4'te verilmiştir.

Tablo 4.4: Lojistik regresyon modeli %80 eğitim, %20 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1150	836
	Negatif	2325	5799

Rasgele Orman için veri setinin %70 eğitim, %30 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.5'de verilmiştir.

Tablo 4.5: Rasgele Orman modeli %70 eğitim, %30 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	2386	641
	Negatif	2668	9407

Rasgele Orman için veri setinin %80 eğitim, %20 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.6'da verilmiştir.

Tablo 4.6: Rasgele Orman modeli %80 eğitim, %20 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1659	327
	Negatif	2013	6101

Gradyan Artırma Makineleri modeli için veri setinin %70 eğitim, %30 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.7'de verilmiştir.

Tablo 4.7: GBM modeli %70 eğitim, %30 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1730	1297
	Negatif	863	11212

Gradyan Artırma Makineleri modeli için veri setinin %80 eğitim, %20 test olarak bölünmesi sonucu oluşturulan karmaşıklık matrisi Tablo 4.8'de verilmiştir.

Tablo 4.8: GBM modeli %80 eğitim, %20 test verisi ile oluşan karmaşıklık matrisi.

Karmaşıklık Matrisi		Tahmin Sınıfı	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	1144	842
	Negatif	556	7558

Karar Ağaçları, Lojistik Regresyon, Rasgele Orman ve Gradyan Artırma Makineleri modellerine ait analizlerin karşılaştırmalı sonuçları Tablo 4.9'da verilmiştir.

Tablo 4.9: Parametre ayarlama sonrası modellerin performans değerlendirme ölçütlerine göre performansları.

Performans Değerlendirme Yöntemi	Model	%70 Eğitim %30 Test	%80 Eğitim %20 Test	Çapraz Doğrulama (5 Kat)	Çapraz Doğrulama (10 Kat)
Doğruluk	**Karar Ağaçları**	0,8415	0,8412	0,8359	0,8389
	Lojistik Regresyon	0,7682	0,6880	0,1980	0,1932
	Rasgele Orman	0,7809	0,7683	0,7837	0,7788
	Gradyan Artırma Makineleri	**0,8570**	**0,8616**	**0,8634**	**0,8549**
Hata	**Karar Ağaçları**	0,1585	0,1588	0,1641	0,1588
	Lojistik Regresyon	0,2318	0,3120	0,6880	0,3120

Tablo 4.9: Parametre ayarlama sonrası modellerin performans değerlendirme ölçütlerine göre performansları (devam).

Performans Değerlendirme Yöntemi	Model	%70 Eğitim %30 Test	%80 Eğitim %20 Test	Çapraz Doğrulama (5 Kat)	Çapraz Doğrulama (10 Kat)
Hata	**Rasgele Orman**	0,2191	0,2317	0,2163	0,2212
	Gradyan Artırma Makineleri	**0,1430**	**0,1384**	**0,1366**	**0,1451**
F-Ölçütü	**Karar Ağaçları**	0,5728	0,5881	0,5475	0,5642
	Lojistik Regresyon	0,4025	0,4219	0,3305	0,3238
	Rasgele Orman	0,5905	0,5864	0,6032	0,6120
	Gradyan Artırma Makineleri	**0,6157**	**0,6207**	**0,6345**	**0,6325**
AUC	**Karar Ağaçları**	0,7799	0,8104	0,7732	0,7816
	Lojistik Regresyon	0,7057	0,7026	0,7030	0,7041
	Rasgele Orman	0,8588	0,8610	0,8670	0,8727
	Gradyan Artırma Makineleri	**0,9057**	**0,9075**	**0,9082**	**0,9074**
Duyarlılık	**Karar Ağaçları**	0,5299	0,5765	0,4987	0,5084
	Lojistik Regresyon	0,3895	0,5791	**1,0000**	**1,0000**
	Rasgele Orman	**0,7882**	**0,8353**	0,8251	0,8349
	Gradyan Artırma Makineleri	0,5715	0,5760	0,5930	0,6004
Kesinlik	**Karar Ağaçları**	0,6232	0,6001	0,6068	0,6338
	Lojistik Regresyon	0,4165	0,3319	0,1980	0,1932
	Rasgele Orman	0,4721	0,4518	0,4753	0,4830
	Gradyan Artırma Makineleri	**0,6672**	**0,6729**	**0,6822**	**0,6681**
Özgünlük	**Karar Ağaçları**	0,9197	0,9060	0,9197	0,9242
	Lojistik Regresyon	0,8632	0,7147	0,0000	0,0000
	Rasgele Orman	0,7790	0,7519	0,7735	0,7640
	Gradyan Artırma Makineleri	**0,9285**	**0,9315**	**0,9310**	**0,9217**

Tablo 4.10'da veri setinde bulunan en önemli 10 nitelik ve önem dereceleri verilmiştir. Taahhüt bitiş tarihi, abonelik yaşı ve fatura tutarı nitelikleri ilk sıralarda yer almaktadır.

Tablo 4.10: GBM modeline niteliklerin önem derecesi.

Nitelik Adı	Önem Derecesi
Taahhüt Bitiş	0,327114
Abonelik Yaşı	0,069136
Fatura Tutarı	0,042022
Paket Hızı	0,040708
Arama Red Sayısı (6 Ay)	0,033316
Ceza Sorma Sayısı	0,028098
Son 10 Gün Kullanım (GB)	0,023705
Son 10 Gün Kullanım (Gün)	0,020813
Abone Yaş	0,020438
Erişim Türü	0,019839

5. TARTIŞMA VE SONUÇ

Gelişen teknolojiden ve aynı zamanda farklı ürün ve hizmetlerin önemli ölçüde bulunabilirliğinden kaynaklanan tüketici tercih ve beklentilerindeki değişim çeşitli sektörlerde rekabetçi bir ortam yaratmıştır. Bu rekabetçi ortamla başa çıkabilmek için şirketler çeşitli stratejiler geliştirmeli ve pazarlama faaliyetlerini değişen müşteri davranışlarına göre ayarlamalıdır. Pazarlama stratejileri belirlenirken benimsenen veri odaklı yaklaşım şirketlerin başarısını önemli ölçüde etkilemektedir. Şirket ve kuruluşlar daha çeşitli ve daha kullanıcı odaklı ürünler ve hizmetler oluşturdukça, kişiselleştirmeler, öneriler ve tahmine dayalı ürün geliştirmek için veri madenciliği uygulamalarına yönelik artan bir ihtiyaç vardır. Veri madenciliği için büyük veri analizi teknolojisinin kullanılması, müşteri kaybını tahmin etmek ve müşteri kaybını azaltmak, telekomünikasyon sektöründeki araştırmaların odak noktası haline gelmiştir. Son zamanlarda, büyük veri teknolojileri daha popüler hale gelmiş ve pek çok alanda kullanılmaya başlamıştır. Çoğu şirket ve kurum için büyük miktarda veriyi işlemek için en uygun veri analitiği çözümlerini bulmak kritik bir konudur. Birçok kaynaktan gelen yüksek hacimdeki veriyle yüksek performans ile verimli bir şekilde başa çıkmak konusunda gerçek bir rekabet vardır. En yaygın ihtiyaçlardan biri, veriye ve faaliyetlerine bağlı olarak müşteri kaybını tahmin etmektir. Var olan müşterilerin, hizmeti kullanmayı terk etmesi genel olarak birçok sektörde büyük bir sorundur ve önemli endişe kaynaklarından biridir. Özellikle yeni müşteri çekmenin zor olduğu ve başka bir servis sağlayıcıya geçmenin nispeten kolay olduğu telekomünikasyon sektöründe, müşteri kaybı ve yeni müşteri kazanımının şirket geliri üzerinde doğrudan etkisi olmasından dolayı şirketler potansiyel müşteri kaybını tahmin etmek için çeşitli araçlar geliştirmeye çalışmaktadır. Bu kapsamda müşteri ayrılma eğilimini tahmin etmek ve bu eğilime neden olan durumları ortaya çıkarmak için ‘Müşteri Kayıp Analizi’ çalışmaları yapılmaktadır.

Bu tez çalışmasının amacı büyük veri analizi teknolojisi ile telekomünikasyon sektöründe ayrılma eğilimi olan müşterileri tahmin eden modeli Apache Spark ile oluşturmaktır. Analiz aracı olarak Apache Spark’ın seçilmesinin sebebi müşteri kayıp analizi çalışmalarında yaygın olarak kullanılmamasıdır, bu nedenle bu tez diğer çalışmalardan ayrışmaktadır.

Bu çalışma kapsamında veri seti kullanılan nitelik alanları bakımından literatür (Huang ve diğ., 2015, Nonghuloo ve diğ., 2020) ile uyum göstermektedir. Huang ve diğerleri (2015)

çalışmalarında 160 nitelik alanı ve 12 aylık bir dönemi kapsayan aylık 2 milyon aboneyi içeren veri seti kullanmıştır. Nitelik alanları paket bilgileri, kullanım bilgileri, fatura kayıtları (ödeme ve borç bilgileri), demografik bilgileri, abonelik yaşam döngüsü, satın alma bilgileri, sözleşmeleri bilgilerinin yanı sıra paket hızı, arıza ve kesinti gibi operasyon ve network katmanından gelen nitelik alanlarını da içermektedir. Nonghuloo ve diğerleri (2020) çalışmalarında kontrat, tarife planı, kullanım, ücretlendirme ve abonelik süresi gibi bilgileri içeren 19 nitelik alanı ve 3.333 satır içeren veri seti kullanmıştır. Bu çalışmada kullanılan veri seti 88 nitelik alanı 50.000 satırdan oluşmaktadır. Nitelik alanları abonenin demografik bilgileri, paket ve ücret bilgileri gibi yapılandırılmış verinin yanı sıra çeşitli sistemlerden gelen yarı yapılandırılmış veri (XML dosyası içinde tutulan kullanım bilgileri) ve yapılandırılmamış veriye (Çağrı merkezi kayıtları) sahiptir. Veri boyutu açısından bakıldığında bu tez sonuçları 2 milyon abone içeren veri seti ile benzer özellikler taşımaktadır. Veri boyutu istenilen miktarda olmasa da, benzer sonuçlar elde edilmiştir.

Büyük veri setleri üzerinde makine öğrenmesi çalışması yapılırken veri setinin sahip olduğu sınıf dağılımlarının oldukça önemli olduğu görülmüştür (Ahmad ve diğ., 2019). Tez kapsamında sınıf dengesizliğinden dolayı bazı modellerin diğer modellere göre başarısız sonuçlar verdiği görülmüştür, bu kapsamda veri setindeki sınıf dağılımı ile model performans ilişkisini değerlendirmesi açısından bu bulgunun literatür ile uyumlu olduğu düşünülmektedir.

Huang ve diğerleri (2015) çalışmalarında Rasgele Orman, Gradyan Artırma Makineleri, Faktorizasyon Makineleri ve Lojistik Regresyon algoritmalarını kullanmış, Rasgele Orman algoritmasının diğer modellere oranla daha başarılı sonuçlar verdiğini gözlemlemiştir. Ahmed ve diğerleri (2019) Karar Ağaçları, Rastgele Orman, Gradyan Artırma Makineleri ve Aşırı Gradyan Artırma Makineleri algoritmalarını kullanmış, en başarılı sonuçları %93,30 AUC skoru ile Aşırı Gradyan Artırma Makineleri algoritması ile elde etmiştir.

Bu tez çalışması kapsamında bir telekomünikasyon şirketinden temin edilen veri setine Apache Spark MLlib kütüphanesi içinde yer alan Karar Ağaçları, Lojistik Regresyon, Rasgele Orman ve Gradyan Artırma Makineleri olmak üzere 6 farklı sınıflandırma algoritması uygulanmış ve modellerin performansı Sınama Seti ve Çapraz Doğrulama yöntemleri kullanılarak değerlendirilmiştir. Modellerin performansı Doğruluk Oranı, F- Ölçütü. Hassasiyet, Kesinlik, Özgünlük ve AUC skoru ölçütlerine bakılarak incelenmiş ve ölçütlere göre karşılaştırma yapılarak müşteri ayrılma eğilimini en iyi tahmin eden sınıflandırma algoritmasının Gradyan

Artırma Makineleri modeli (%86,34 doğruluk, %63,45 F- ölçüsü ve %90,82 AUC) olduğu belirlenmiştir (Tablo 4.9).

Ahmed ve diğerleri (2019) nitelik seçme ve nitelik alanı oluşturmak için istatistiksel niteliklerinden yanında, sosyal ağ analizinden elde edilen nitelik alanları da kullanmıştır. Müşteri ayrılma eğilimini belirleyen en önemli nitelik alanlarını diğer operatörün müşteriyle olan benzerliği, en son işlem yapılan gün, toplam bakiye, radyo erişim türü ortalaması (2G veya 3G), müşterilerin birbiri ile olan yakınlığı, diğer operatörler ile yapılan işlemlerin yüzdesi, abone yaşı, sosyal güç faktörü, sinyal sorunları, abonelik yaşı olarak tahmin etmiştir. Olasehinde ve diğerleri (2018) müşteri ayrılma eğilimini tanımlayan en önemli faktörleri abonelik yaşı, kontrat, internet servisi (fiber optik) ve ödeme tipi olarak bulmuştur.

Kurulan modeller içerisinde en başarılı performans gösteren GBM modeline göre müşteri ayrılma eğilimini belirleyen en önemli niteliğin taahhüt bitimine kalan süre olduğu görülmüştür. Operatörler, abonelerine sağladığı faydalar karşılığında şirket gelirini güvence altına almak adına aboneler ile sözleşme yapmaktadır. Sözleşmede belirtilen taahhüt bitiş tarihinden önce aboneliğin sonlandırılması durumunda yapılan sözleşme kapsamında cayma bedeli adı verilen tutar yansıtılmaktadır. Bu nedenle taahhüt bitiş tarihine uzun süre kalan aboneler, aboneliklerini sonlandırmaları durumunda taahhüt bitiş tarihine kısa süre kalan abonelere göre daha yüksek cayma bedeli ödeyecekleri için daha az ayrılma eğiliminde olmaktadırlar. Bu tez çalışması kapsamında operatörden temin edilen veri setindeki tüm aboneler için taahhüt bitiş tarihlerine 3 aydan az kalan abonelerden seçilmiştir. Bu yaklaşımın temel amacı taahhüt cayma bedelinin dışında churn davranışına sebep olan diğer faktörleri tespit edebilmektir. Diğer nitelikler ise abonelik yaşı, fatura tutarı, paket hızı, son 6 ay içerisinde operatör tarafından yapılan aramalara ret adedi ve ceza sorma adedidir. Abonelik yaşı fazla olan aboneler daha az ayrılma eğiliminde olmasının sebebi, uzun süredir abone olan müşteriler rakiplerin pazarlama faaliyetlerine daha az duyarlı olmasıdır. Bilgi Teknolojileri Kurumu'nun yayınladığı 3. Çeyrek Pazar Verileri Raporuna (BTK, 2020) göre Türkiye'de 15,9 milyon sabit geniş bant abonesi bulunmaktadır ve bu abonelerin yaklaşık %34,7'sinin 10-16 Mbps hızda bağlantı sunan paketleri kullandıkları görülmektedir. Bu hız bandındaki paketlerin yoğun olarak kullanılmasının sebebi mevcuttaki paket ücretleri ve hizmetin verildiği alt yapının desteklediği hızdır. Yüksek paket hızları uygun fiyata sunulduğu takdirde aboneler başka bir sağlayıcı lehine daha fazla ayrılma eğiliminde olmaktadır.

Bu çalışma ile birlikte Apache Spark'ın makine öğrenmesi kütüphanesinin müşteri kayıp analizi çalışmalarında kullanım kolaylığı ve hızı nedeniyle etkin olarak kullanılabileceği görülmüştür. Spark MLlib'in sahip olduğu tüm fonksiyonlar pyspark ile kullanılabilmektedir. Gerekli olduğu durumlar Python fonksiyonlarının da kullanılabiliyor olması kullanıcıya esneklik sağlamaktadır. Spark bellek içi hesaplama yazılım çerçevesi olmasından dolayı yoğun bir bellek kullanımı vardır, bu sebeple local bilgisayar kullanıldığında sistemin çalışmasında duraklamaya ve ısınma sorunlarına sebep olmaması için kullanılacak bellek kapasitesi önem taşımaktadır. Bu nedenle tek düğüm kümesi için en az 8-16 GB bellek tavsiye edilmektedir. Veri hacmi arttıkça bellek ihtiyacı da artmaktadır.

Gelecek çalışmalarda en iyi performans gösteren GBM modelinden elde edilen niteliklerin önem dereceleri göz önüne alınarak ayrılma eğilimini belirlemede önemli olduğu tespit edilen nitelikler (taahhüt bitiş, abonelik yaşı, fatura tutarı, paket hızı, arama ret sayısı (6 ay), ceza sorma sayısı, son 10 gün kullanım (gb), son 10 gün kullanım (gün), abone yaş, erişim türü) ile yeni bir veri seti oluşturulabilir veya sadece bu nitelikler kullanılarak veri seti üzerinde analizler yapılabilir. Böylelikle modellerin çalışma süresinin kısalacağı ve model karmaşıklığının azalmasından dolayı modellerin çıktılarının yorumlanmasının daha kolay olacağı düşünülmektedir. Ek olarak bir web arayüzü oluşturulup bir sistem kurulabilir ve belirtilen nitelikler üzerinden churn tahmini yapılması için sistem herkese açılabilir. Gelecek çalışmalarda hacim ve çeşit olarak daha zengin veri setleri ile çalışmanın denenmesi, model performansını arttırılmasına katkı sağlayabilir.

Çalışmada sınıf dengesizliğini aşmak ve diğer modellere göre başarısız sonuç veren Lojistik Regresyon ve Rastgele Orman modellerini iyileştirebilmek için Apache Spark 3.0 ile birlikte gelen örnek ağırlık desteği kullanılmıştır. Sonraki çalışmalarda veri setinin sahip olduğu dengesiz dağılım, bu çalışmadan kullanılan yöntemden daha farklı yöntemler kullanılarak giderilebilir. Bu noktada özellikle F ölçütü ve doğruluk oranında artış gözlemlenebilir.

Gelecek çalışmalarda Apache Spark ile diğer veri analizi araçları ile performans açısından karşılaştırılabilir. Spark ML mevcutta doğrusal yöntemler, ağaçlar ve topluluk öğrenme yöntemlerini içeren paralel olarak çalışabilen sınıflandırma ve regresyon algoritmalarını desteklemektedir. Kütüphaneye yeni bir sınıflandırma algoritması eklenmesi durumunda, veri seti üzerinde yeni modeller çalıştırılabilir ve GBM'den daha iyi performans gösteren başka bir

model tespit edilebilir. Çalışma Spark tarafından desteklenen Scala ve R programlama dilleri ile tekrarlanabilir ve bu dillerin sahip olduğu esneklikler ve kısıtlar karşılaştırılabilir.

Sonuç olarak, tez çalışması veri analizi alanında büyük veri yaklaşımı ile makine öğrenmesi konusunda hazırlanmıştır. Veri analizi aracı olarak Apache Spark kullanılmış, büyük veri yaklaşımı ile müşteri kayıp analizi ele alınmış ve Apache Spark'ın etkin olarak kullanılabileceğine yönelik bir çalışma ortaya konmuştur. Bu tez çalışmasının çeşitli yönleri ve gelecek çalışmalara yönelik sunduğu öneriler bakımından literatüre katkısı olacağı düşünülmektedir.

KAYNAKLAR

Ahmad, A. K., Jafar, A., Aljoumaa, K., 2019, Customer churn prediction in telecom using machine learning in big data platform, *Journal of Big Data,* 6(1), 1-24.

Akgün, B., 2016, *Apache Spark Tabanlı Destek Vektör Makineleri ile Akan Büyük Veri Sınıflandırma*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Akpınar, H., 2014, *Data-Veri Analizi, Veri Madenciliği*, Papatya Yayıncılık, İstanbul, ISBN: 978-605-4220-816.

Ali, A., Qadir, J., ur Rasool, R., Sathiaseelan, A., Zwitter, A., and Crowcroft, J. ,2016, Big data for development: applications and techniques, *Big Data Analytics*, 1(1), 2.

Alpaydin, E., 2020, *Introduction to machine learning*, MIT press, Cambridge Massachuttes, ISBN: 978-0262-043793.

Apache Hadoop, 2020, *Apache Hadoop HDFS Architeture*, https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html , [Ziyaret Tarihi: 10 Aralık 2020]

Apache Hadoop, 2020, *Apache Hadoop*, https://hadoop.apache.org/ [Ziyaret tarihi: 8 Şubat 2021].

Apache Spark, 2020, Apache Spark Cluster Mode Overview, 2020, https://spark.apache.org/docs/latest/cluster-overview.html , [Ziyaret tarihi: 17 Kasım 2020].

Apache Spark, 2020, *Apache Spark MLLib Guide*, https://spark.apache.org/docs/latest/ml-guide.html , [Ziyaret tarihi: 17 Kasım 2020].

Apache Spark, *Spark Streaming Programming Guide*, 2020, https://spark.apache.org/docs/latest/streaming-programming-guide.html , [Ziyaret tarihi: 17 Mayıs 2020].

Bakshi, K., 2012, *Considerations for big data: Architecture and approach*, 2012 IEEE Aerospace Conference, 3-10 March 2012 Big Sky MT, USA, IEEE, ISBN: 978-1-4577-0557-1, 1-7.

Berson, A., Smith, S., Thearling, K, 2000, *Building data mining applications for crm*. New York, NY: McGraw-Hill.

Bilgi Teknolojileri Kurumu, *3. Çeyrek Pazar Verileri Raporu*, 2020, https://www.btk.gov.tr/uploads/pages/pazar-verileri/uc-aylik-pazar-verileri-2020-3-kurumdisi.pdf, [Ziyaret Tarihi 31 Ocak 2021)

Breiman L., 1996, Bagging Predictors, *Machine Learning*, 24(2), 123-140.

Breiman L., 2001, Random Forests, *Machine Learning*, 45(1), 5–32.

Breiman, L., Friedman, J., Stone, C., Olshen, R. A.,1984, *Classification and regression trees*, Wadsworth

Cattell, R., 2011, Scalable SQL and NoSQL data stores, *ACM Sigmod Record*, 39 (4), 12-27.

Chamberlain B. P., Cardoso A., Liu C. B., Pagliari R., Deisenroth M. P., 2017, Customer lifetime value prediction using embeddings, *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , August 2017 Halifax Canada, New York, Association for Computing Machinery, 1753-1762,

Chen, C.P., Zhang, C.Y., 2014, Data-intensive applications, challenges, techniques and technologies: A survey on big data, *Information Sciences*, 275, 314-347.

Chen, H., Chiang, R.H.L., Storey, V.C., 2012, Business intelligence and analytics: from big data to big impact, *MIS Quarterly*, 36 (4), 1165-1188.

Chen, M., Mao, S., Zhang, Y., and Leung, V. C., 2014, *Big data: related technologies, challenges and future prospects*, Springer, London, ISBN 978-3-319-06245-7

Çetin, M.F., 2019, *Building a hybrid recommender system using apache spark*, Yüksek Lisans Tezi, Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü.

Databricks, 2020, *Databricks*, https://databricks.com/spark/about , [Ziyaret tarihi: 24 Nisan 2020].

Deligiannis A., Argyriou C., 2020, Designing a real-time data-driven customer churn risk ındicator for subscription commerce, *International Journal of Information Engineering & Electronic Business*, 12(4), 1-14.

Demchenko, Y., Grosso, P., De Laat, C. and Membrey, P., 2013, Addressing Big Data Issues in Scientific Data Infrastructure, *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 20-24 May 2013, San Diego, CA, USA, 48-55.

Diebold, F., 2000, ”*Big Data” dynamic factor models for macroeconomic measurement and forecasting*, Discussion Read to the Eighth World Congress of the Econometric Society.

Diebold, F., 2012, *On the origin(s) and development of the term”big data”*, Pier working paper archive, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania.

Dolapcı B., 2020, *Apache spark kullanılarak büyük boyutlu görüntülerin analizi*, Yüksek Lisans Tezi, Karabük Üniversitesi Lisansüstü Eğitim Enstitüsü.

Ebner, K., Bühnen, T., Urbach, N., 2014, Think big with big data: ıdentifying suitable big data strategies in corporate environments. *47th Hawaii International conference on system sciences,* 6-9 Jan 2014 Waikoloa HI*, USA,* IEEE, 3748-3757.

Elgendy, N., Elragal, A., 2014, *Big data analytics: a literature review paper*, Lecture Notes in Computer Science, 214-227.

Elragal, A., 2014, ERP and big data: The inept couple, *Procedia Technology*, 16, 242-249.

Emani, C.K., Cullot, N., Nicolle, C., 2015, Understandable big data: a survey, *Computer science review*, 17, 70-81.

Erdem, Y., 2017, *Büyük verinin makine öğrenmesi yöntemleri ile apache spark teknolojisi kullanılarak sınıflandırılması*, Yüksek Lisans Tezi, Karabük Üniversitesi Fen Bilimleri Enstitüsü.

Erevelles S., Fukawa N., Swayne L., 2016, Big data consumer analytics and the transformation of marketing, *Journal of business research*, 69(2), 897-904.

Etaiwi W., Biltawi M., Naymat G., 2017, Evaluation of classification algorithms for banking customer's behavior under Apache Spark Data Processing System, *Procedia computer science*, 113, 559-564.

Eyüpoğlu C., 2018, *Büyük veride etkin gizlilik koruması için yazılım tasarımı, Doktora Tezi*, İstanbul Üniversitesi Fen Bilimleri Enstitüsü.

Friedman, J.H., 1999a, Greedy function approximation: a gradient boosting machine, https://statweb.stanford.edu/~jhf/ftp/trebst.pdf [Ziyaret Tarihi: 28 Kasım 2020].

Friedman,J.H., 1999b, Stochastic Gradient Boosting https://statweb.stanford.edu/~jhf/ftp/stobst.pdf, [Ziyaret Tarihi: 28 Kasım 2020].

Gandomi, A., Haider, M., 2015, Beyond the hype: Big data concepts, methods, and analytics, *International Journal of Information Management*, 35 (2), 137-144.

Gantz J., Reinsel D., 2011, Extracting value from chaos*, IDC iview,* 1142(2011), 1-12.

Gebreyesus, T.Y., 2018, *Sentiment analysis in social media: a comparative study*, Yüksek Lisans Tezi, Atılım Üniversitesi Fen Bilimleri Enstitüsü.

Guller, M., 2015, *Big data analytics with Spark: A practitioner's guide to using Spark for large scale data analysis*, Apress.

Hallaç, İ.R., 2014, *Büyük veri analizinde dağıtık makine öğrenmesi algoritmalarının kullanılması*, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

Han, J., Kamber, M. ve Pei, J., 2012, *Data mining: concepts and techniques*, 3. baskı, Morgan Kaufman Publishers, ABD, ISBN: 978-0-12-381479-1.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U., 2015, The rise of "big data" on cloud computing: Review and open research issues, *Information Systems*, 47, 98-115.

Ho, T. K.,1998, The random subspace method for constructing decision forests, *IEEE transactions on pattern analysis and machine intelligence*, 20(8), 832-844.

Hosmer, D. W., Lemeshow, S., 1989, *Applied logistic regression*, John Wiley & Sons, New Jersey, ISBN: 978-0-470-58247-3.

Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R. N., West, M.J., 1988, Scale and performance in a distributed file system, *ACM Transactions on Computer Systems*, 6 (1), 51-81.

Huang, B., Kechadi, M. T., Buckley, B., 2012, Customer churn prediction in telecommunications*, Expert Systems with Applications*, 39(1), 1414-1425.

Huang, Y., Zhu, F., Yuan, M., Deng, K., Li, Y., Ni, B., Zeng, J., 2015, Telco churn prediction with big data, *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, May 2015 Melbourne, Australia, New York Association for Computing Machinery, 607-618.

Hung, S. Y., Yen, D. C., Wang, H. Y., 2006, Applying data mining to telecom churn management, *Expert Systems with Applications*, 31(3), 515-524.

Karabay, B.,2019, *Yaşanan terör olaylarını içeren büyük verinin makine öğrenmesi teknikleri ile analizi*, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

Karau, H., Konwinski, A., Wendell, P., Zaharia, M.,2015, *Learning Spark: Lightning-Fast Big Data Analytics*, O'Reilly Media Inc., Sebastopol, ISBN: 978-1-449-359862-4.

Keskin, V.M., 2018, *Büyük veride makine öğrenmesi uygulaması*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Kisioglu, P., Topcu, Y. I., 2011, Applying bayesian belief network approach to customer churn analysis: A case study on the telecom industry of Turkey, *Expert Systems with Applications*, 38(6), 7151–7157.

Koçoğlu F.Ö, 2017, *Müşteri kayıp analizi probleminin çözümünde analitik yaklaşımlar*, Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü.

Kurt, M.E. ,2019, *Apache spark ve makine öğrenmesi algoritmaları ile ağ saldırısı tespiti*, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü.

Kwon, O., Lee, N., Shin, B., 2014, Data quality management, data usage experience and acquisition intention of big data analytics, *International Journal of Information Management*, 34 (3), 387-394.

Labrinidis, A. and Jagadish, H.V., 2012, Challenges and opportunities with big data, *VLDB Endowment*, 5 (12), 2032-2033.

Laney, D., 2001, 3D data management: controlling data volume, velocity, and variety, *Application Delivery Strategies by META Group Inc*., https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf , [Ziyaret tarihi: 24 Şubat 2020]

Lazarov, V., Capota, M., 2007, *Churn prediction*, Business Analytics Course, TUM. Computer Science, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.462.7201&rep=rep1&type=pdf , [Ziyaret Tarihi: 11 Kasım 2020].

López, V., del Río, S., Benítez, J. M., Herrera, F., 2015, Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data, *Fuzzy Sets and Systems*, 258, 5-38.

Lycett, M., 2013, 'Datafication': making sense of (big) data in a complex World, *European Journal of Information Systems*, 22(4), 381–386.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A.H., 2011, *Big Data: The Next Frontier for Innovation, Competition, and Productivity,* McKinsey Global Institute, McKinsey&Company, New York, NY, USA, https://bigdatawg.nist.gov/pdf/MGI_big_data_full_report.pdf , [Ziyaret tarihi: 23 Eylül 2020].

Meher A. K., Wilson J., Prashanth R, 2017, Towards a large scale practical churn model for prepaid mobile markets, *Advances in Data Mining Applications and Theoretical Aspects*, *ICDM 2017*, 12-13 July 2017 New York, USA, Springer Cham, 93-106.

Miele S., Shockly R., 2013, *Analytics: The real-world use of big data,* https://www.informationweek.com/pdf_whitepapers/approved/1372892704_analytics_the_real_world_use_of_big_data.pdf , [Ziyaret Tarihi: 23 Eylül 2020].

Müller, A, Guido, S., 2016, *Introduction to machine learning with python*, O'Reilly Media Inc., Sebastopol, ISBN: 978-1-449-36941-5.

Mysore, D., Khupat, S. and Jain, S., 2013, *Big data architecture and patterns, Part 1: Introductionto big data classification and architecture, How to classify big data into categories*, IBM developerWorks, https://www.ibm.com/developerworks/analytics/library/bd-archpatterns1/index.html, [Ziyaret tarihi: 23 Eylül 2020].

Nonghuloo, M. S., Reddy, R. A., Manideep, G., SarathVamsi, M. R., Lavanya, K. ,2020, Call churn prediction with pyspark, *Soft Computing for Problem Solving*, Springer, Singapore 783-793.

Oğur, N.B., 2018, *EKG verileri için gerçek zamanlı veri analitiği mimarisi*, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

Ohlhorst, F., 2013, *Big data analytics turning big data into big Money*, John Wiley & Sons, New Jersey, ISBN 978-1-118-22582-0.

Olasehinde O., Johnson O., Fakoya, J., 2018, Computational efficiency analysis of customer churn prediction using spark and caret random forest classifier, *Information and Knowledge Management*, 8(2), 8-16.

Özdeş, M., 2017, *Büyük veri araçlarını kullanarak duygu analizi gerçekleştirimi*, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

Özkan, Y., 2020, *Veri Madenciliği Yöntemleri*, 4. baskı, Papatya Yayıncılık Eğitim, ISBN: 978-975-6797-82-2.

Pyspark, 2020, *PySpark Documentation*, https://spark.apache.org/docs/latest/api/python/ , [Ziyaret Tarihi: 3 Mart 2021].

Quinlan, J.R., 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN: 978-1-55860-238-0.

Quinlan, J.R., 1996, Learning Decision Tree Classifiers, *ACM Computing Surveys (CSUR),* 28 (1), 71–72.

Richeldi, M., Perrucci, A., 2002, *Churn analysis case study*, https://www-ai.cs.tu-dortmund.de/PublicPublicationFiles/richeldi_perrucci_2002b.pdf , [Ziyaret Tarihi: 7 Temmuz 2020].

Salloum, S., Dautov, R., Chen, X., Peng, P. X., & Huang, J. Z., 2016, Big data analytics on Apache Spark, *International Journal of Data Science and Analytics*, 1(3-4), 145-164.

SAS, 2020, *Big data: what it is and why it is matter*, https://www.sas.com/tr_tr/insights/big-data/what-is-big-data.html ,[Ziyaret tarihi: 8 Aralık 2020].

Sayed, H., Abdel-Fattah, M. A., Kholief, S., 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, 674-677.

Spanoudes P., Nguyen T.,2017, Deep learning in customer churn prediction: unsupervised feature learning on abstract company independent feature vectors, *arXiv preprint,* arXiv:1703.03869.

Terzi, D.S., Terzi, R., Sagiroglu, S., 2015, A survey on Security and Privacy Issues in Big Data, *10th International Conference for Internet Technology and Secured Transactions (ICITST)*, 14-16 December 2015, London, UK, 202-207.

Van den Poel, D., Lariviere, B., 2004, Customer attrition analysis for financial services using proportional hazard models, *European journal of operational research*, 157(1), 196-217.

Wassouf W. N., Alkhatib R., Salloum K., Balloul S., 2020, Predictive analytics using big data for increased customer loyalty: Syriatel Telecom Company case study, *Journal of Big Data*, 7, 1-24.

Weiss, S. M., Indurkhya, N., 1998, *Predictive data mining: a practical guide*, Morgan Kaufmann.

Yadav, S. K., ve Pal, S., 2012, Data mining: A prediction for performance improvement of engineering students using classification, *arXiv preprint arXiv:1203.3832*.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I., 2010, Spark: cluster computing with working sets, *HotCloud*, 10(10-10), 95.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Ghodsi, A. ,2016, Apache spark: a unified engine for big data processing, *Communications of the ACM*, 59(11), 56-65.

Zhang, W., Mo, T., Li, W., Huang, H., Tian, X, 2016, The comparison of decision tree based insurance churn prediction between spark ML and SPSS, *9th International Conference on Service Science (ICSS)*, 15-16 October 2016 Chongqing, IEEE, 134-139.

Zikopoulos, P., Eaton, C., 2011, *Understanding big data: analytics for enterprise class hadoop and streaming data*, McGraw-Hill Osborne Media.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Ezgi Usta
Doğum Yeri	-
Doğum Tarihi	-
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	-
E-Posta Adresi	-
Web Adresi	

Eğitim Bilgileri	
Lisans	
Üniversite	İstanbul Ticaret Üniversitesi
Fakülte	Mühendislik ve Tasarım
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	Tarih girmek için tıklayın veya dokunun.

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Enformatik
Programı	Enformatik