T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BÜYÜK VERİ ARAÇLARINI KULLANARAK DUYGU ANALİZİ GERÇEKLEŞTİRİMİ

YÜKSEK LİSANS TEZİ

MERVE ÖZDEŞ

DENİZLİ, NİSAN - 2017

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BÜYÜK VERİ ARAÇLARINI KULLANARAK DUYGU ANALİZİ GERÇEKLEŞTİRİMİ

YÜKSEK LİSANS TEZİ

MERVE ÖZDEŞ

DENİZLİ, NİSAN - 2017

KABUL VE ONAY SAYFASI

Merve ÖZDEŞ tarafından hazırlanan "Büyük Veri Araçlarını Kullanarak Duygu Analizi Gerçekleştirimi" adlı tez çalışmasının savunma sınavı 22.05.2017 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / ~~oy çokluğu~~ ile Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Danışman Prof. Dr. Sezai TOKAT	.....
Üye Yrd. Doç. Dr. Hasan BULUT	.....
Üye Yrd. Doç. Dr. Meriç ÇETİN	.....

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 21/06/2017 tarih ve ..24/18.... sayılı kararıyla onaylanmıştır.

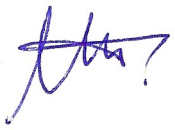
.....

Prof. Dr. Uğur YÜCEL

Fen Bilimleri Enstitüsü Müdürü

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde bilimsel etiğe ve akademik kurallara özenle riayet edildiğini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etiğe uygun olarak kaynak gösterildiğini ve alıntı yapılan çalışmalara atfedildiğine beyan ederim.

Merve ÖZDEŞ



ÖZET

BÜYÜK VERİ ARAÇLARINI KULLANARAK DUYGU ANALİZİ GERÇEKLEŞTİRİMİ
YÜKSEK LİSANS TEZİ
MERVE ÖZDEŞ
PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI:PROF. DR. SEZAİ TOKAT)

DENİZLİ, NİSAN - 2017

İnternetin yaygın olarak kullanılmasıyla birlikte veri miktarında da inanılmaz büyüklükte artış meydana gelmiştir. Veri miktarındaki bu artış, bu verilerin yönetimini zorlaştırmakla birlikte, bu veriler arasından anlamlı bilgiler elde etmeyi de gerekli kılmıştır. Geleneksel veri tabanlarıyla verilerin saklanması, işlenmesi ve analiz edilmesi gibi işlemlerin yapılamaması büyük veri kavramını ortaya çıkarmıştır. Büyük veri kavramı verinin oluşturulması, saklanması, işlenmesi ve analiz edilmesi gibi işlemlerin tümüne verilen addır. Basit bir ifadeyle, verinin anlamlı ve işlenebilir hale dönüştürülmüş biçimidir. İnternet ortamında paylaşılan video, blog, resim, web sunucularının log dosyaları, GSM operatörlerinin arama kayıtları ve buna benzer birçok kaynak büyük veri araçlarıyla işlenerek anlamlı hale dönüştürülmektedir. Üretim, pazarlama, telekomünikasyon, hükümet kaynakları, sağlık ve eğitim gibi birçok alanda büyük veri inanılmaz kolaylık sağlamaktadır. Büyük veri analizi için kullanılan pek çok araç mevcuttur. Bu tezde, büyük veri araçlarından olan Spark kullanılarak elde edilen veriler üzerinde duygu analizi işlemi gerçekleştirilmiştir. Duygu analizi, sözlüğe dayalı ve makine öğrenmesine dayalı olmak üzere iki farklı şekilde gerçekleştirilebilmektedir.

Bu tezde, makine öğrenmesi yöntemlerinden biri olan denetimli öğrenme metoduyla duygu analizi işlemi gerçekleştirilmiştir. Toplamda 57.650 adet İngilizce şarkı sözü üzerinde veri temizleme işlemleri gerçekleştirildikten sonra, pozitif ya da negatif olacak şekilde etiketleme işlemi gerçekleştirilmiştir. Etiketlenen veri pozitifse 1, negatifse 0 değeri ile skorlanarak duygu analizi işleminde kullanılacak algoritmalara uygun bir formata dönüştürülmüştür. Dönüştürülen bu veri, denetimli öğrenme algoritmalarından Naive Bayes, Logistic Regresyon ve Decision Tree olmak üzere toplamda üç farklı algoritmaya tabi tutularak, algoritmanın çalıştırılması sonucu elde edilen başarım oranları karşılaştırılmıştır. Veri, RStudio üzerinde Naive Bayes algoritmasıyla tekrar çalıştırılmış ve algoritmanın işlemesi için geçen süresi Spark üzerinde geçen süreyle karşılaştırılmıştır. Spark'ın bu karşılaştırma sonucunda çok daha hızlı olduğu görülmüştür. Son olarak da çalışmanın geliştirilmeye açık yönleri belirtilmiş ve gelecek çalışmalar için önerilerde bulunulmuştur.

ANAHTAR KELİMELER: Duygu Analizi, Büyük Veri, Spark, Öznitelik Seçme, Naive Bayes, Logistic Regression, Decision Tree

ABSTRACT

SENTIMENT ANALYSIS USING BIG DATA TOOLS
MSC THESIS
MERVE ÖZDEŞ
PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE
COMPUTER ENGİNEERİNG

(SUPERVISOR:PROF. DR. SEZAİ TOKAT)

DENİZLİ, APRIL 2017

With the widespread usage of the Internet, the amount of data has also increased enormously. This increase in the amount of data has also made it necessary to obtain meaningful information from these data, as well as making it difficult to manage this data. The fact that data can not be stored, processed and analyzed by traditional databases reveals the concept of big data. The term of big data is sum of all operations such as creating, storing, processing and analyzing the data. In simple terms, the form is transformed into meaningful and processable. The log files of web servers, videos, blogs, images shared on internet, search records of GSM operators and many other similar resources are converted into meaningful data by processing with big data tools. Big data in many fields such as production, marketing, telecommunications, government resources, health and education provide incredible convenience. There are many tools available for big data analysis. In this thesis, sentiment analysis is performed on the data obtained by Spark, which is a big data tool. Sentiment analysis can be performed in two different ways, based on dictionary and machine learning.

In this thesis, sentiment analysis process is performed with supervised learning method which is one of the machine learning methods. After a total of 57.650 songs were cleaned in the English language, labeling was performed either positively or negatively. The tagged data were converted to a form suitable for the algorithms to be used in the sentiment analysis process by scoring 1 if it is positive otherwise, with 0. The transformed data is subjected to three different algorithms, namely Naive Bayes, Logistic Regression and Decision Tree, from supervised learning algorithms, and the performance ratios obtained by running the algorithm are compared. The data was re-run on RStudio with the Naive Bayes algorithm, and the time spent for the algorithm to run was compared to the time spent on Spark. It has been found that Spark is much faster in this comparison. Finally, explicit aspects of the study were identified and suggestions for future studies were made.

KEYWORDS: Sentiment Analysis, Big Data, Spark, Feature Exctraction, Naive Bayes, Logistic Regression, Decision Tree

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

KISALTMALAR LİSTESİ

IDF	**:**	Ters Doküman Frekansı (Inverse Document Frequency)
TF-IDF	**:**	Terim Frekansı – Ters Doküman Frekansı
IDC	**:**	International Data Corporation
IoT	**:**	Nesnelerin İnterneti (Internet of Things)
HDFS	**:**	Hadoop Dağıtık Dosya Sistemi (Hadoop Distributed File System)
GFS	**:**	Google Dosya Sistemi (Google File System)
PB	**:**	Petabayt
MLlib	**:**	Machine Learning Library
SAS	**:**	Statistical Analysis System
RDD	**:**	Resilient Distributed Data

ÖNSÖZ

Bu tezin yürütülmesinde bilgi ve deneyimleriyle benden yardımlarını esirgemeyen Yrd. Doç. Dr. Gürhan GÜNDÜZ'e ve danışman hocam Prof. Dr. Sezai TOKAT'a teşekkürlerimi sunarım.

Tez süresince desteklerini aldığım meslektaşlarım ve aynı zamanda yakın arkadaşlarım Araş. Gör. Elif Gülfidan DAYIOĞLU'na, Araş. Gör. Mustafa TOSUN'a, Araş. Gör. Selahattin AKKAŞ'a, Araş. Gör. Erdi KAYA'ya ve diğer çalışma arkadaşlarıma teşekkürü bir borç bilirim.

Eğitim ve çalışma hayatım boyunca maddi manevi her zaman yanımda olan, desteklerini benden hiçbir zaman esirgemeyen değerli aileme ve en pes ettiğim durumlarda bile yanımda olamasa da telefondan bana destek olan, her aradığında tezin ne zaman bitecek diye soran biricik annem Fatma ÖZDEŞ'e teşekkürlerimi sunuyorum.

1. GİRİŞ

Teknolojinin gelişmesiyle ve kullanım alanlarının artmasıyla birlikte bilginin gücü de ön plana çıktı. İnsanların teknolojiyi etkin bir şekilde kullanmalarına bağlı olarak artan veri miktarı, bu verilerden anlamlı bilgilerin elde edilmesini de zorunlu kılmıştır. Geleneksel veri tabanı sistemleri ve algoritmaları ile sürekli artış gösteren devasa miktardaki verinin işlenmesi, saklanması, akışı ve analiz edilmesi gibi birçok konuda yaşanan sıkıntılar bu becerilere sahip olan büyük veri kavramını ortaya çıkarmıştır. Basit bir ifadeyle büyük veri; sosyal medya üzerinde yaptığımız paylaşımlar, internette gezinirken bıraktığımız izler, bloglar, internet istatistikleri, log dosyaları, fotoğraf arşivleri, mailler videolar, sağlık kayıtları, sensörler ve mobil araçlar gibi farklı kaynaklardan elde ettiğimiz, anlamlı ve işlenebilir biçime dönüştürülmüş devasa büyüklükteki veri yığınıdır.

İnsanların internetin yaygınlaşmasıyla birlikte alışkanlıklarının değişmesi ve buna bağlı olarak günlük yaşamda alınan birçok hizmetin internet üzerinden gerçekleştirilebilir olması, özellikle hizmet sektöründe satış sonrası memnuniyeti sağlama amacına yönelik olarak, son kullanıcıların her türlü bilgisine ulaşarak anlamlı ve kullanılabilir bilgi üretimi sonucunu doğurdu. Müşteri profilini daha iyi analiz ederek, onlara bireyselleştirilmiş hizmet sunabilmek için şirketlerin müşterileriyle ilgili çok sayıda bireysel bilgiyi saklaması gerekmektedir. Sağlık, hükümet kaynakları, üretim, bankacılık, sosyal medya / duygu analizi, Telekom, e-ticaret, medya, eğitim ve perakende satış gibi birçok alanda kullanılan büyük veri bu anlamda kolaylık sağlamaktadır. Birçok alan için önem kazanan Büyük Veri'nin analizi için özel şirketler ve kamu kuruluşları tarafından ciddi yatırımlar yapılmakta, yeni teknik ve yazılımlar geliştirilmektedir.

Bu tezde büyük veri araçlarından biri olan Spark ortamında makine öğrenmesine dayalı duygu analizi işlemi gerçekleştirilmiştir. 2014 yılında Apache Spark açık kaynak kodlu büyük veri deposunu kuran Databricks firmasının bulut platformunda SAS olarak sunduğu Spark kullanılmıştır. Databricks, Apache Spark'ın üstünde, kullanıcıların gelişmiş analitik çözümlerini kolayca kurmasına ve

yerleştirmesine olanak tanıyan gerçek zamanlı bir veri platformu sağlar. Spark'ın üzerine kurulu, yüksek kaliteli algoritmalar ve etkileyici hız sağlayan ölçeklenebilir bir makine öğrenme kitaplığı olan MLlib tezde kullanılan denetimli makine öğrenmesi algoritmaları için kullanılmıştır. Makine öğrenmesi algoritmalarından denetimli öğrenme algoritmalarından olan Naive Bayes(NB), Logistic Regression(LR) ve Decision Tree (DT) algoritmalarının başarım oranları karşılaştırılmıştır. Karşılaştırılan bu algoritmalardan Naive Bayes algoritması, RStudio üzerinde de gerçekleştirilerek Spark üzerinde algoritma için geçen süre ile RStudio üzerinde algoritma için geçen süre karşılaştırılmıştır. Bu tezde Duygu Analizi için kullanılan makine öğrenmesi algoritmalarından yararlanılmıştır ve verinin miktarına bağlı olarak algoritmanın işletilmesi için geçen sürenin iyileştirilmesi öngörülmüştür. Bu nedenle makine öğrenmesi algoritmaları büyük veri araçlarından biri olan Spark üzerinde gerçekleştirilmiştir. Spark üzerinde algoritmanın gerçekleştirimi RStudio'ya göre çok daha hızlıdır.

Duygu analizi ve büyük veri alanında literatürde bulunan çalışmalar incelendiğinde birçoğunda Hadoop kullanıldığı gözlenmiştir. Spark, genellikle verileri işleyiş biçimi nedeniyle Hadoop'un veri bileşeni olan MapReduce'dan çok daha hızlıdır. Aynı zamanda Spark'ta RDD kavramı, veriyi belleğe kaydetmenizi ve yalnızca gerekli olması halinde ve aynı zamanda işlemi yavaşlatabilecek herhangi bir senkronizasyon engeline sahip olmadığında diske saklamanızı sağlar. Hadoop platformunun özellikleri ile karşılaştırıldığında gerek hız bakımından daha üstün performansa sahip olması gerekse farklı dilleri desteklemesi nedeniyle bu çalışmanın Spark üzerinde gerçekleştirilmesi öngörülmüştür.

İnceleyeceğiniz bu tezde büyük veri ve büyük veri araçları, duygu analizi, duygu analizi alanında yapılan çalışmalar anlatılmıştır. 5. Bölümde uygulanan yöntemler detaylandırılmış ve 6. Bölümde yapılan uygulamadan elde edilen sonuçlar gösterilmiştir.

2. LİTERATÜR ÇALIŞMASI

1990 yıllarında ortaya çıkan Duygu analizi alanında, farklı dillerde birçok çalışma gerçekleştirilmiştir. Yapılan bu çalışmalar; öznellik sınıflandırması, duygusal sınıflandırma, istenmeyen fikir taraması, fikir özetleme ve karşılaştırmalı fikirlerin çıkarılması gibi farklı amaçlar doğrultusunda gerçekleştirilmiştir. Genellikle İngilizce ve Çince dilleri üzerinde çalışmalar gerçekleştirilmiştir. Günümüzde Arapça, Türkçe, İtalyanca gibi diğer diller üzerinde de sınıflandırma çalışmaları yapılmıştır (Ghag ve diğ. 2013).

Carlos ve Coletta (2014) çalışmalarında, sınıflandırma ve kümeleme algoritmalarını birlikte kullanan SVM algoritmasının, sadece sınıflandırma işlevini yerine getiren SVM algoritmasından daha iyi doğruluk oranı verdiğini göstermişlerdir. Bu algoritma, aynı kümelerdeki benzer örneklerin sınıf etiketini paylaşma olasılığı yüksek olduğu varsayılarak, kümeler tarafından sağlanan ek bilgilerden tweet sınıflandırmalarını hassaslaştırabilmektedir. Bu çalışma için Health Care Reform (HCR), Obama, Sanders ve Stanford gibi dört farklı gruptan tweetler kullanılmıştır.

Shankar ve diğ. (2014) çalışmalarında, Google'ın Gmail'de mesajlar için kullandığı sınıflama işlemini benzer biçimde Facebook haber akışlarının sınıflandırılmasını gerçekleştirmişlerdir. Kullanıcıların haber akışlarını, kullanıcı duvarlarında verilerin daha iyi bir şekilde gösterilmesini sağlamak için sınıflandırıcılar kullanarak çeşitli kategorilere sınıflamayı amaçlamışlardır. Bu çalışma, gelecekte Facebook sayfasına iyi düzenlenmiş ve daha çekici bir görünüm sağlayacak olan otomatik haber akışı sınıflandırma ve duygu analizi yöntemlerini değerlendirmek amacıyla veri setlerine sınıflandırma yöntemlerinin uygulanmasına yöneliktir. Facebook Restfb Java API kullanarak yaklaşık 2000 eğitim verisi toplamışlar ve bu veri üzerinde Binary Logistic Regression, Naive Bayes Classifier, Support Vector Machine(SVM), Bayes Net ve J48 gibi farklı algoritmaları uygulamışlardır.

Kang ve Park (2014) çalışmalarında, mobil hizmetler için müşteri memnuniyetinin ölçümünde duygu analizi ve çok kriterli karar verme yaklaşımlarından olan VIKOR algoritmasını birleştirerek yeni bir çerçeve geliştirmeyi hedeflemişlerdir. Önerilen çerçeve, veri toplama ve önişleme sonrasında müşteri memnuniyetinin ölçülmesi şeklinde iki aşamadan oluşmaktadır. İlk aşamada, müşteri

yorumlarını esas alan nitelik ve duygu kelimelerinden oluşan sözlükleri derlemek için veri madenciliği kullanılır. Daha sonra duygu analizi kullanılarak, her bir mobil servisin nitelikleri için hesaplanan duygu puanına göre bir vektör elde edilmiştir. Elde edilen değerler VIKOR algoritmasında kullanılarak müşterilerin hangi servisten daha memnun kaldığı belirlenmiştir (Kang ve diğ. 2014).

Mostafa (2013) çalışmasında, tüketicilerin farklı markalara yönelik duygularını değerlendirmek için rasgele 3516 tweet örneği kullanmıştır. Analiz için Hu ve Liu (2004) tarafından oluşturulmuş, 2006 pozitif ve 4783 negatif sözcüğü barındıran sözlüğü kullanmıştır. QDA Miner adında bir yazılımla havayolu firmasının Twitter verilerini çekerek, bu firma adıyla birlikte en çok hangi kelimenin geçtiğini tespit etmiştir. Daha sonra Nokia ve Phizer için atılan tweetleri karşılaştırmıştır.

Singh ve diğ. (2013) film yorumlarının duygu analizi için alana özgü özellik tabanlı yeni bir yöntem sunmuşlardır. Bir filmin metinsel incelemelerini analiz eden ve her yönü üzerinde bir duygu etiketi atayan bir sistem geliştirmişlerdir. Film yorumlarının her açıdan hesaplanmış puanları toplanarak filmin duygu profili her parametre üzerinde üretilmiştir. Sıfatlar, zarflar ve fiiller ile n-gram özelliği içeren SentiWordNet kullanılmıştır. Ayrıca incelenen her film için belge seviyesinde duyguları hesaplamak için SentiWordNet kullanmış ve sonuçları Alchemy API kullanarak elde edilen sonuçlarla karşılaştırmışlardır. Elde edilen sonuçlar, basit belge tabanlı duygu analizinden daha doğru ve odaklanmış bir duyarlılık profili ürettiğini göstermiştir.

Web dokümanlarının sınıflandırılması, terörle ilişkili belgeleri ortaya çıkaran en önemli tekniklerden biri olarak görülüyor. Web içerik miktarı arttıkça, potansiyel olarak tehlikeli olan belgeleri tanımlamak da zorlaşmıştır. Dokümanlardan anahtar kelimeler çıkarmak içeriği sınıflandırmak için yeterli değildir. Bu nedenle otomatik belge sınıflandırma sistemleri oluşturmak için bugüne kadar pek çok teknik incelenmiştir. Ancak, bu yöntemler istatistiksel ve bilgiye dayalı yaklaşımlardır. Doğal dillerin karmaşıklığı yüzünden sonuçlar tatmin edici değildir. Choi ve diğ. (2014) bu eksikliği gidermek için WordNet hiyerarşisine ve n-gram veri frekansına dayalı bir kelime benzerliği kullanma yöntemi önermiştir. Bu yöntem, dört farklı alandaki dört farklı kelimeyi sorgulayarak örneklenmiş New York Times makaleleri ile test edilmiştir.

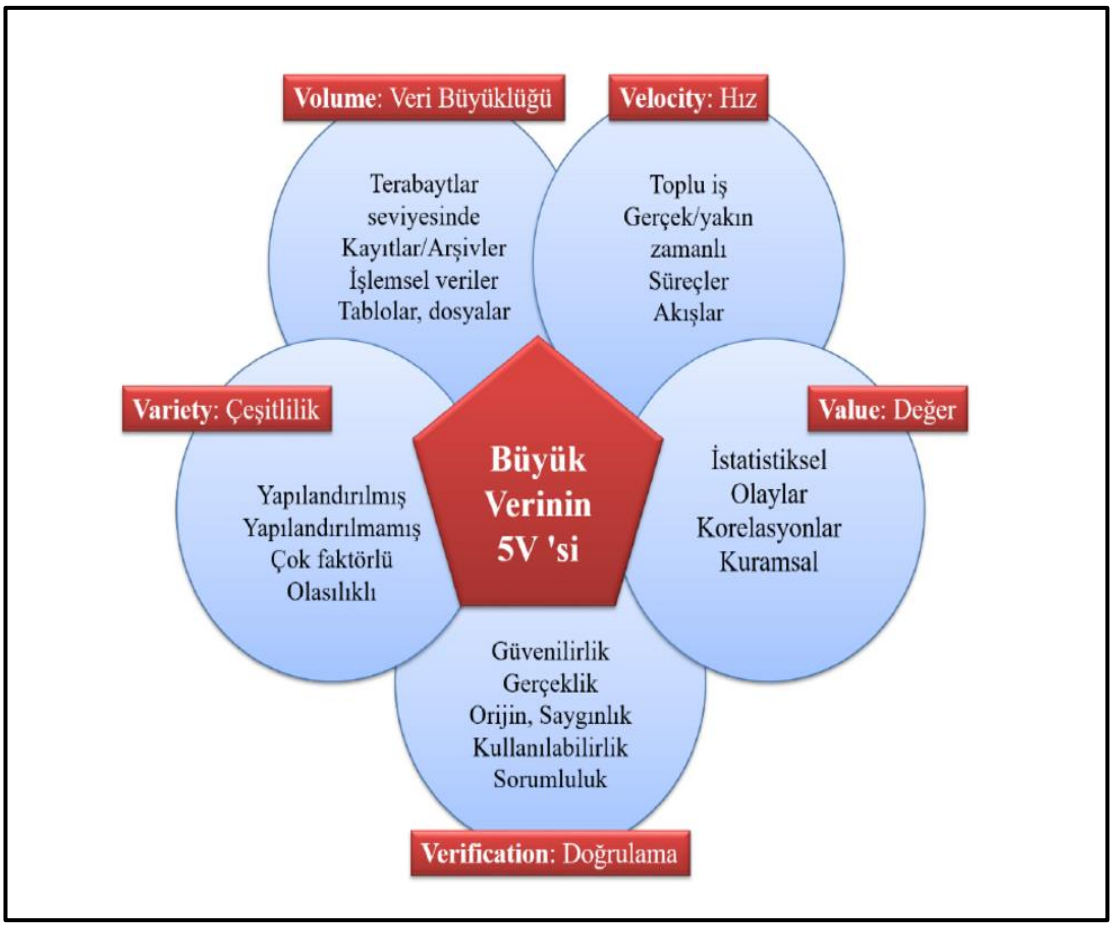
3. BÜYÜK VERİ

Büyük veri, mevcut bilgi sistemlerinin işleyemeyeceği kadar geniş ve farklı türdeki veri kümelerine verilen addır. Farklı bir deyişle, bilinen veri tabanı yönetim sistemleri ve yazılım araçlarının, verileri toplama, saklama, yönetme ve çözümleme yeteneklerini aşan büyüklükteki verilere büyük veri (big data) denir.

Bilişim dünyası hızla gelişmekte ve bu da verinin üssel bir biçimde artmasına yol açmaktadır. Mevcut verinin %92'si geçtiğimiz iki yılda oluşturulmuş olması gelecekte verilerin ne hızla büyüyeceği konusunda bize fikir vermektedir. IDC (International Data Corporation) istatistiklerine göre, 2020 yılında elde edilecek olan veri miktarı 44 kat daha fazla olacaktır. Twitter'da günlük 500 milyon tweet atılmakta, günün her dakikasında 570'in üzerinde yeni web sitesi kurulmakta, 2013 yılında 2.712.239.573 internet kullanıcısı, Google'da yaklaşık olarak 1,2 trilyon arama yapmış ve dijital dünyanın %70'i olan 900 exabyte veri kullanıcılar tarafından oluşturulmaktadır. Yukarıda sıralanan istatistikler, artan bu veri miktarının ne denli önemli ve hızlı olduğunun bir göstergesidir. Büyük veri; farklı türde oluşturulan bu verilerin toplanıp, çözümlenmesiyle müşteri profilini belirleme, tahminler yapma ve işletmelerin sakladıkları verilerinden yola çıkarak akıllı yönetim imkânı sağlayarak, karar alma sürecinde işletmelere ciddi bir kolaylık sağlar.

3.1 Büyük Veri Bileşenleri

Büyük veri kavramını daha iyi anlamak için onun oluşumundaki beş bileşeni incelemek gerekir. Büyüklük kavramı yalnızca hacim ile ilgili değildir. Gerek kapsadığı alan (volume) gerekse çok hızlı hareket etmesi (velocity) veya kullanılabilir bir şekilde yapılandırılmamış çeşitlikte (variety) olmasıyla da ifade edilebilir. Ayrıca bu bileşenlerin tamamlayıcısı olarak değerlendirebileceğimiz, verinin güvenli oluşu yani doğrulanabilir olması (verification) ile anlamlı bir değere (value) sahip olması da önemli bileşenlerdir. Şekil 3.1'de büyük verinin bileşenleri gösterilmiştir.



Şekil 3.1: Büyük Veri Bileşenleri (5V).

Hacim (Volume) : Hacim, verinin geleneksel yöntemlerle ele alınamayacak düzeyde büyük olduğunu ifade eder. Teknolojinin gelişip, hayatımızın vazgeçilmez bir parçası olduğundan beri ürettiğimiz verinin boyutunda büyük oranda artış olmuştur. Veri büyüklüğü terabyte ve petabytedan daha büyük hale geldiğinden, geleneksel depolama ve analiz araçları yetersiz kalmıştır. Bunun için yeni teknik ve araçların geliştirilmesi kaçınılmaz olmuştur.

Hız (Velocity) : Verinin üretilme hızının sürekli artış göstermesi veriyi kullanacak işlem sayısının ve çeşitliliğinin de aynı oranda artması sonucunu doğuruyor.

Çeşitlilik (Variety) : Farklı veri kaynaklarından elde edilen verinin %80'ine yakın bir kısmı yapısal değildir ve farklı formattadırlar. Farklı türdeki bu verilerin birbirlerine dönüşmeleri gerekir.

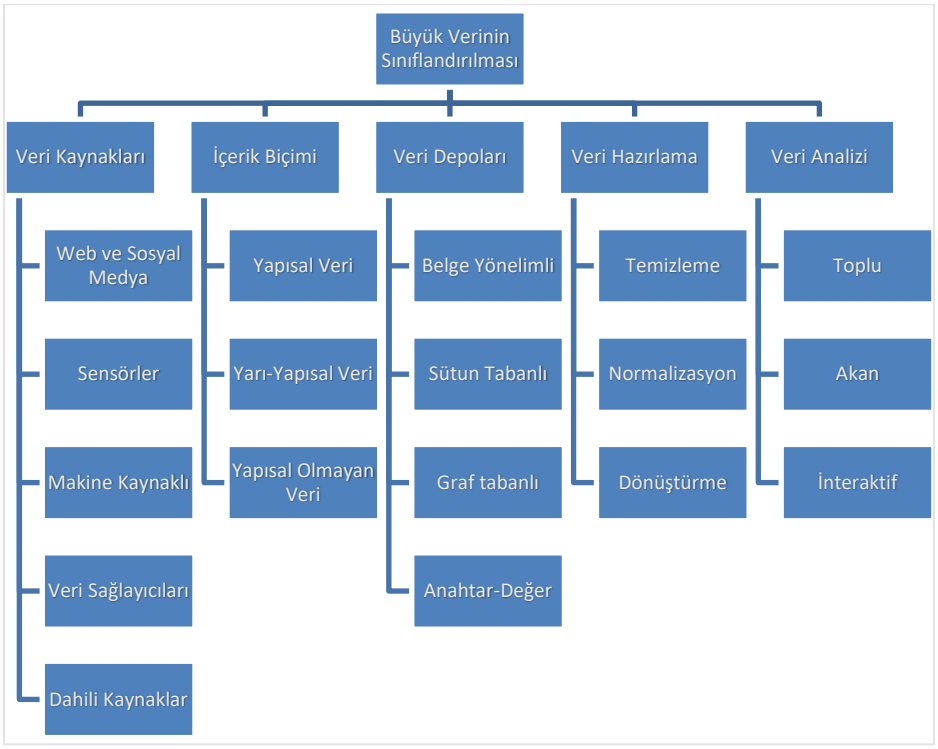
Doğrulama (Verification) : Verinin doğruluk ve güvenliğiyle ilgilidir. Veri miktarı arttıkça, bu verinin doğruluğunu kontrol etmek de güçleşmektedir. Veri akışı

sırasında verinin gerektiği güvenlik seviyesinde izlenmesi, doğru kişiler tarafından görünebilir veya gizli kalması gerekiyor.

Value (Değer) : Büyük veri bileşenlerinden son ve en önemli bileşendir. Verinin üretim ve işleme adımlarından sonra anlamlandırılarak kurum için artı bir katkı sağlaması gerekmektedir. Karar verme aşamasında anında etki etmesi, bu kararı vermede ulaşılabilir olması gerekmektedir.

3.2 Büyük Verinin Sınıflandırılması

Büyük verinin karakteristik özelliklerini daha iyi anlamak için veri, farklı kategorilerde sınıflandırılmıştır. Şekil 3.2'de gösterildiği gibi veri kaynağı, içerik biçimi ve analiz türü gibi beş farklı kategori bulunmaktadır.



Şekil 3.2: Büyük Verinin Sınıflandırılması.

Veri Kaynakları: Sosyal medya, makine kaynaklı veriler, dahili kaynaklar ve sensörler gibi bir çok kaynaktan ham veri elde edebiliriz. Sosyal medya; bloglar,

mikrobloglar, Facebook ve Twitter gibi sanal topluluklar ve ağlardaki bilgi paylaşımında veya fikir alışverişinde bulunmak için URL yoluyla üretilen bilgi kaynağıdır. Makine verileri, insan müdahalesi olmadan bilgisayarlar, tıbbi cihazlar veya diğer makineler gibi donanım ya da yazılımdan otomatik olarak üretilen bilgilerdir. Günümüzde giderek yaygınlaşan, Nesnelerin İnterneti (Internet of Things, IoT) olarak adlandırdığımız teknoloji de bir veri kaynağıdır. IoT, internetin bir parçası olarak tanımlanabilen, çeşitli haberleşme protokollerini kullanarak haberleşen ve birbirlerine bağlanarak bilgi paylaşımında bulunan akıllı bir ağ oluşturmuş cihazlar kümesidir. İnternete bağlı çok sayıda cihaz, birçok türde hizmet sunar ve büyük miktarda veri üretir (Abaker ve diğ. 2015).

İçerik Biçimi: Yapısal, yarı-yapısal ve yapısal olmayan veri türü olmak üzere üç tür veri biçimi bulunmaktadır. Yapılandırılmış veriler, ilişkisel bir veri tabanındaki bilgiler gibi yüksek oranda organizasyona sahip veri türleri anlamına gelir. Bu verilerin girilmesi, sorgulanması, depolanması ve analiz edilmesi kolaydır. Bu verilere örnek olarak sayılar, kelimeler ve tarihler verilebilir. Yarı yapısal veri, bir veri tabanı gibi özel bir depoya yerleştirilmemiş, ilişkili veri modellerinin biçimsel yapısına uymayan ancak işlenmemiş verilere göre analiz işlemini daha kolay hale getiren ilişkili bilgilere sahip yapılandırılmış verinin bir formudur. Metin mesajları, konum bilgileri, videolar ve sosyal medya verileri gibi yapılandırılmamış veriler, tanımlanabilir içyapıya sahip değildir (Comput ve diğ. 2015).

Veri Depoları: Belge odaklı veri depoları, belge ve bilgilerin koleksiyonlarını depolamak, almak ve karmaşık veri formlarını JSON, XML ve ikili formlar (PDF ve MS Word) gibi çeşitli standart formatlarda desteklemek üzere tasarlanmıştır. Sütun odaklı bir veri tabanı, içeriğini satırların yanı sıra sütunlarda depolar. Aynı sütuna ait öznitelik değerleri bitişik olarak depolanır. Neo4j gibi bir grafik veri tabanı düğümler ve kenarlarla birbirleriyle ilişkili olan grafik modelini kullanan verileri depolamak ve temsil etmek üzere tasarlanmıştır. Anahtar-Değer, çok büyük ölçekte ölçeklendirilmek üzere tasarlanan, veriyi depolayan ve bunlara erişen alternatif bir ilişkisel veritabanı sistemidir. Dinamo, Apache Hbase, Apache Cassandra ve Voldemort bu veritabanı türüne örnek olarak verilebilir (Comput ve diğ. 2015).

Veri Hazırlama: Temizleme, eksik ve mantıksız verilerin belirlenmesi işlemidir. Dönüşüm, veriyi analiz için uygun bir forma dönüştürme işlemidir.

Normalleştirme ise veri tabanının yanlış tasarımı sonucu ortaya çıkan kötü ilişkileri en aza indirgemek için veri tabanı şemasını yapılandırma yöntemidir (Comput ve diğ. 2015).

Veri Analizi: MapReduce tabanlı sistemler, uzun süren toplu işler için son yıllarda birçok organizasyon tarafından benimsenmiştir. Bu tür sistemler, binlerce düğümden oluşan büyük kümelerde bulunan makinelerdeki uygulamaların ölçeklendirilmesine olanak tanır. En güçlü ve yaygın olarak kullanılan gerçek zamanlı işlem temelli büyük veri araçlarından biri basit ölçeklenebilir akış sistemidir (Comput ve diğ. 2015).

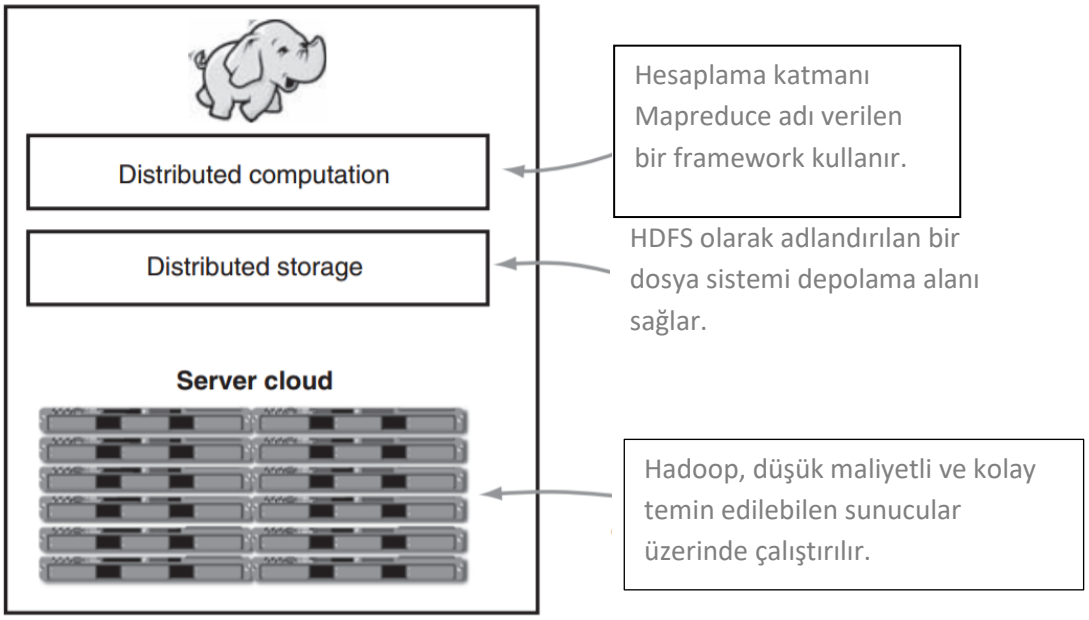
3.3 Büyük Veri Araçları

Büyük veri kavramı yukarıda da bahsedildiği gibi mevcut sistemlerin işleyemeyeceği türdeki veriler olduğundan analiz işlemi için yeni teknikler ve teknolojiler geliştirilmiştir. Şimdiye kadar bilim adamları, büyük verileri saklamak, analiz etmek ve görselleştirmek için çok çeşitli teknikler ve teknolojiler geliştirdiler. Buna rağmen çeşitli ihtiyaçların karşılanmasında bu teknik ve teknolojiler yetersiz kaldı. Büyük veri içerisindeki değerli bilgiye erişmek için çok disiplinli yöntemlere ihtiyaç vardır. Büyük veriyi anlamaya yönelik de farklı araçlar geliştirilmiştir. Mevcut araçlar, toplu işleme (batch processing) araçları, akış işleme (stream processing) ve etkileşimli analiz araçları gibi üç farklı sınıfa odaklanmaktadır. Çoğu yığın işleme aracı Mahout ve Dryad gibi Apache Hadoop altyapısına sahiptir. Storm ve S4, büyük ölçekli akışlı veri analitik platformlarına verilebilecek örneklerdendir.

3.3.1 HADOOP

Hadoop, dağıtık depolama ve hesaplama imkânı sağlayan bir küme mimarisidir. Yaygın olarak kullanılan, arama motoru kütüphanesi Apache Lucene projesinin de geliştiricisi olan Doug Cutting tarafından geliştirilmiştir. Google arama motorunun 2003 yılında tanıttığı dağıtık veri işleme modeli olan Mapreduce programlama modelinden esinlenilerek, açık kaynak arama motoru projesi olan Apache Nutch için dağıtık dosya sistemi ve Mapreduce gerçekleştirimi yapılmıştır. Bu

altyapının arama işlevinin yanı sıra başka işlevleri de yerine getirebileceği düşüncesiyle, Hadoop adı altında bağımsız bir alt proje oluşturulmuştur. Hadoop, depolama için Hadoop Dağıtık Dosya Sistemi'ni (HDFS) ve hesaplama işlemleri için MapReduce kullanan master node ve slave node olarak adlandırılmış node'lara sahip bir mimaridir. Master node, bir ya da birden fazla slave olarak adlandırılan prosesi kontrol eden prosestir. Depolama ve hesaplama yetenekleri, bir Hadoop kümesine ana bilgisayarların eklenmesiyle ölçeklenebilir ve binlerce bilgisayarların bulunduğu kümeler üzerinde hacim boyutlarına petabayt olarak erişilebilir.



Şekil 3.3: Hadoop'un temel yapısı.

Günümüzde birçok başarılı firma Hadoop teknolojisini kullanmaktadır. Bunlardan bazıları aşağıda kullanım amaçları ve özellikleriyle birlikte verilmiştir.

EBay

Arama optimizasyonu ve araştırmalar için kullanılmıştır. 532 nodes cluster(8*532 çekirdek, 5.3 PB)

Facebook

Dahili log verilerinin saklanmasında, raporlanmasında, analizinde ve makine öğrenimi için kullanılmıştır.8800 çekirdeğe ve yaklaşık 12 PB ham depolamaya sahip

1100 adet makine kümesi, 2400 çekirdeğe ve yaklaşık 3 PB ham depolamaya sahip 300 makine kümesi bulunmaktadır. Her düğüm 8 çekirdek ve 12 TB depolama alanına sahiptir.

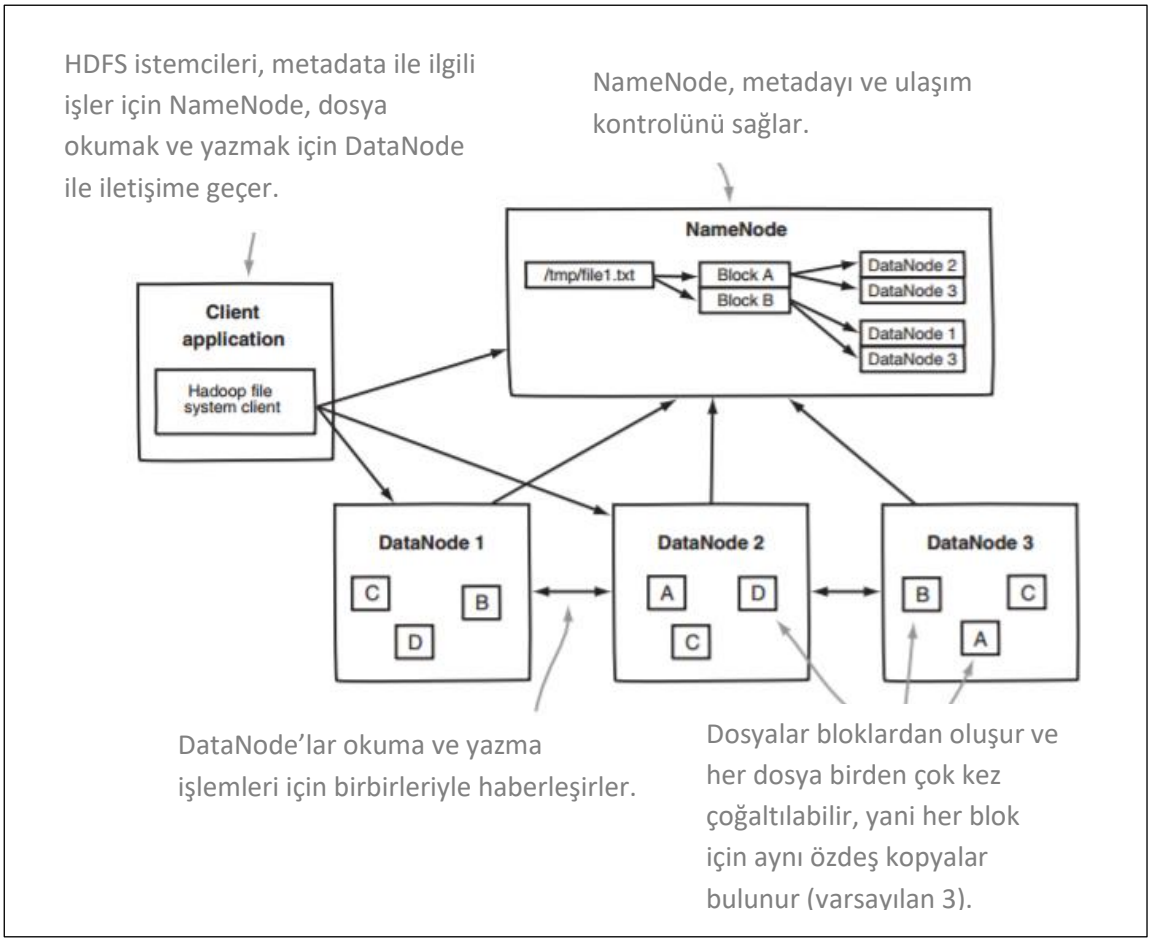
Twitter

Tweetleri, log dosyalarını ve Twitter'da üretilen diğer birçok veriyi depolamak ve işlemek için kullanılmıştır. Tüm veriler sıkıştırılmış LZO dosyaları olarak depolanmıştır. Hadoop'un MapReduce API'lerine erişmek için de hem Scala hem de Java kullanılmıştır. Aynı zamanda Apache Pig, zamanlanmış ve ad-hoc işler için yoğun şekilde kullanılmıştır.

Yukarıda örnek olarak verilen firmaların yanısıra IBM,Yahoo!, LinkedIn gibi birçok firma da Hadoop teknolojisini kullanmaktadır. Hadoop, HDFS ve MapReduce olmak üzere temelde iki bileşene sahiptir.

3.3.2 HDFS (Hadoop Distributed File System)

HDFS, Hadoop'un büyük verileri depolamak için kullanılan bileşenidir. Google Dosya Sistemi (GFS) yayınlandıktan sonra modellenen bu dosya sistemi, yüksek verim için optimize edilmiştir ve büyük dosyaları okuma yazma işleminde en iyi sonucu alır(Holmes, n.d.). HDFS, Namenode ve buna bağlı olarak birden fazla Datanode olarak adlandırılan özel düğümlerden oluşmaktadır. NameNode, her dosya için hangi DataNode'un hangi blokları kontrol ettiğinin bilgisini bellekte tutar. Kısaca master görevindedir. DataNode ise HDFS için depolama blokları sunar. HDFS yüksek hata toleransına sahiptir. HDFS üzerindeki veriler, farklı düğümlerde de kaydedilmektedir. Yani birçok makine arasında veri kopyalanır.Bu da herhangi bir hata durumunda veri kaybını önler. Şekil 3.4'te de görüldüğü gibi D bloğu hem DataNode1 hem de DataNode2'de tutulmaktadır.



Şekil 3.4: HDFS mimarisi.

3.3.3 MapReduce

MapReduce, bir Hadoop kümesindeki yüzlerce veya binlerce sunucuda ölçeklenebilirlik sağlayan bir fonksiyonel programlama uygulamasıdır. Bu terim, aslında Hadoop programlarının gerçekleştirdiği map ve reduce olarak adlandırılan iki ayrı ve farklı görevi temsil etmektedir.

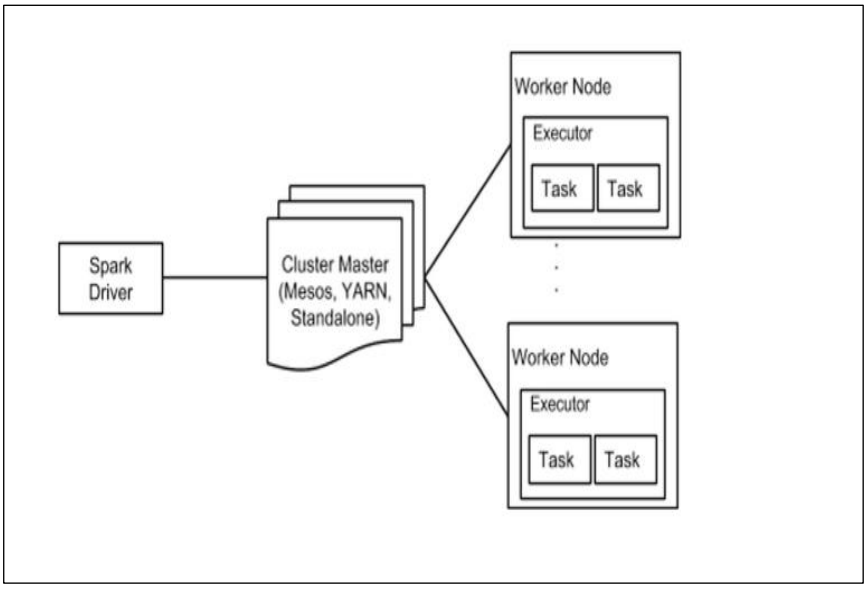
Haritalama (Map) işlemi, bir dizi veri alır ve bunu ayrı öğelerin tüple (anahtar / değer çifti) olarak ayrıldığı başka bir veri grubuna dönüştürür. İndirgeme (Reduce) işlemi ise map işleminden elde edilen çıktıyı girdi olarak alır ve bu anahtar / değer çiftlerini daha küçük tüple grubuna birleştirir.

3.3.4 Spark

Büyük veri kümelerinin art arda gelen sorgularını daha iyi işlemek için geliştirilmiş olan Spark, Scala dili ile yazılmış, bellek içi veri işleme özelliğine sahip

açık kaynak kodlu bir veri işleme aracıdır. Hadoop, Mapreduce'u temel alır ve petabyte'larca veriye ölçeklenebilir, esnek, hata toleranslı ve uygun maliyetli bir bilgi işlem çözümü sağlar. Ancak buradaki temel sorun her işlemin bir önceki işleme bağlı olmasından kaynaklanan bekleme süresidir. Spark, Hadoop'un zayıf kaldığı bazı konulardaki eksiklerini tamamlamaktadır. Bellek içi bilgi işlem özelliği ile verinin bellekte olması ve tekrar eden süreçlerde çok hızlı olması Spark'ı ön plana çıkarmıştır.

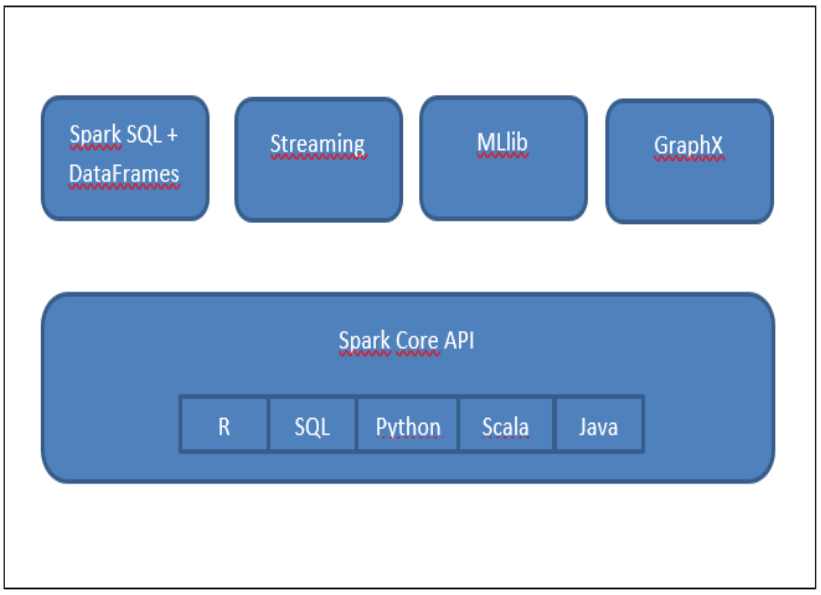
Spark'ın kendi dağıtık dosya sistemi yoktur. Bunun yerine, HDFS ve HBase gibi Hadoop tarafından desteklenen depolama sistemlerini kullanır. Verileri önbelleklemek için esnek dağıtılmış veri (Resilient Distributed Data-RDD) kümelerini kullanır. Spark'ın çalışma zamanı bir sürücü (driver) ve birden fazla işçiden (worker) oluşur. Bu işçiler, dağıtılmış bir dosya sisteminden veri bloklarını okur ve bunları RDD'nin bölümleri olarak kendi hafızalarında saklarlar. Kullanıcılar, yeniden kullanılan verileri ve ara sonuçları RDD'ler biçiminde belleğe kaydedebilir ve veri yerleşimini optimize etmek için bölümlemelerini kontrol edebilir. RDD'lerin hata toleransı vardır; bu da, bir bölümün kaybolması durumunda yeniden oluşturulabilir demektir (Gu ve Li 2013).



Şekil 3.5: Spark çalışma mantığı

Spark mimarisi temelde Spark SQL, Streaming, MLlib, GraphX ve Spark Core API bileşenlerinden oluşur. Spark Core, diğer tüm işlevlerin üstünde kurulu olan Spark

platformunun temel yürütme altyapısıdır. Hız sağlamak için bellek içi işlem yetenekleri, çeşitli uygulamaları desteklemek için bir yürütme modeli ve geliştirme kolaylığı için Java, Scala ve Python gibi farklı API desteği sunar. GraphX, Spark üzerine kurulmuş bir grafik hesaplama altyapısıdır ve büyük grafik problemleri olarak görülebilen öğrenme algoritmalarını uygulamak için güçlü bir API sunar. MLlib, Spark'ın üzerine kurulu, hem yüksek kaliteli algoritmalar ve etkileyici hız sağlayan ölçeklenebilir bir makine öğrenme kitaplığıdır. Kütüphane, Java, Scala ve Python'da Spark uygulamalarının bir parçası olarak kullanılabilir. Streaming, yalnızca toplu veriyi değil aynı zamanda gerçek zamanlı olarak yeni verilerin akışlarını işleme ve analiz etme ihtiyacı duyan uygulamalarda, hem anlık hem de diğer veriler arasında güçlü interaktif ve analitik çözümler sağlamaktadır. HDFS, Flume, Kafka ve Twitter da dahil olmak üzere çeşitli veri kaynakları ile kolayca bütünleşmiş olur. SQL tabanlı veriler üzerinde işlemler yapmamızı sağlayan Spark SQL, Parquet, Hive, JSON ve ilişkisel veritabanı üzerinde bulunan verilerden SQL tabanlı sorgulama imkânı verir.



Şekil 3.6: Spark ekosistemi

3.3.4.1 Spark ve Hadoop Arasındaki Farklar

Hadoop ve Apache Spark büyük veri araçlarındandır. Ancak ikisi de aynı amaçlara hizmet etmezler. Hadoop temelde dağıtılmış veri altyapısıdır ve büyük miktardaki veri koleksiyonlarını, özel donanım satın alınmasını ve bakım

gerektirmeyen sunucular kümesindeki birçok düğüme dağıtır. Aynı zamanda, bu verileri endeksler, izler ve büyük veri işlemeyi etkili bir biçimde gerçekleştirebilir. Spark, diğer taraftan bu dağıtılmış veri koleksiyonlarında çalışan bir veri işleme aracıdır. Dağıtılmış depolama alanı yapmaz.

Spark, genellikle verileri işleyiş biçimi nedeniyle Hadoop'un veri bileşeni olan MapReduce'dan çok daha hızlıdır. Spark'da RDD kavramı, veriyi belleğe kaydetmenizi ve yalnızca gerekli olması halinde ve aynı zamanda işlemi yavaşlatabilecek herhangi bir senkronizasyon engeline sahip olmadığında diske saklamanızı sağlar. Bu nedenle, Spark'ın genel yürütme motoru Hadoop MapReduce'dan daha çok hızlıdır.

Hadoop, her işlemden sonra veriler diske yazıldığından sistem hatalarına ya da arızalarına dayanıklıdır. Spark ise veri nesnelerinin, veri kümesinde dağıtılan RDD yapılarında depolanması nedeniyle benzer esnekliğe sahiptir.

4. DUYGU ANALİZİ

4.1 Duygu Analizi Nedir?

Fikir Madenciliği olarak da bilinen Duygu Analizi (Sentiment Analysis), bir varlık üzerinde insanların tutum, düşünce ve duygularının bilgisayar bilimleri kullanılarak ortaya çıkarılmasını amaçlayan bir araştırma alanıdır (Medhat ve diğ. 2014). Belirli bir konu veya hedefin özelliğine göre metinler olumlu, olumsuz veya tarafsız olup olmadığına göre araştırılır.

Günümüzde internetin gelişmesi, insanların duygu ve fikirlerini paylaşmak istemesiyle sosyal medyanın önemi artmıştır. Sosyal medya artık sadece bir iletişim aracı olmaktan çıkmış, belirli bir ürün veya konu hakkında, insanların görüşlerini paylaştığı önemli bir bilgi kaynağı haline gelmiştir (Onan ve Korukoğlu 2016). Medya takibi yapan kişiler ve kurumlar, duygu analizine metinleri olumlu, olumsuz veya tarafsız olarak sınıflandırarak verimli ve verimsiz veriyi birbirinden ayırmak için ihtiyaç duymaktadırlar. Duygu analizi bir sınıflandırma işlemi olarak da düşünülebilir. Üç farklı sınıflandırma seviyesi bulunmaktadır. Bunlar:

-Doküman seviyesinde (Document Level)

-Cümle seviyesinde (Sentence Level)

-Özellik temelli (Aspect Based)

Doküman seviyesinde sınıflandırma, bir konu üzerinde yazılmış olan dokümanı ele alır. Tek bir doküman ele alınarak, dokümanın olumlu ya da olumsuz olduğu tespit edilir.

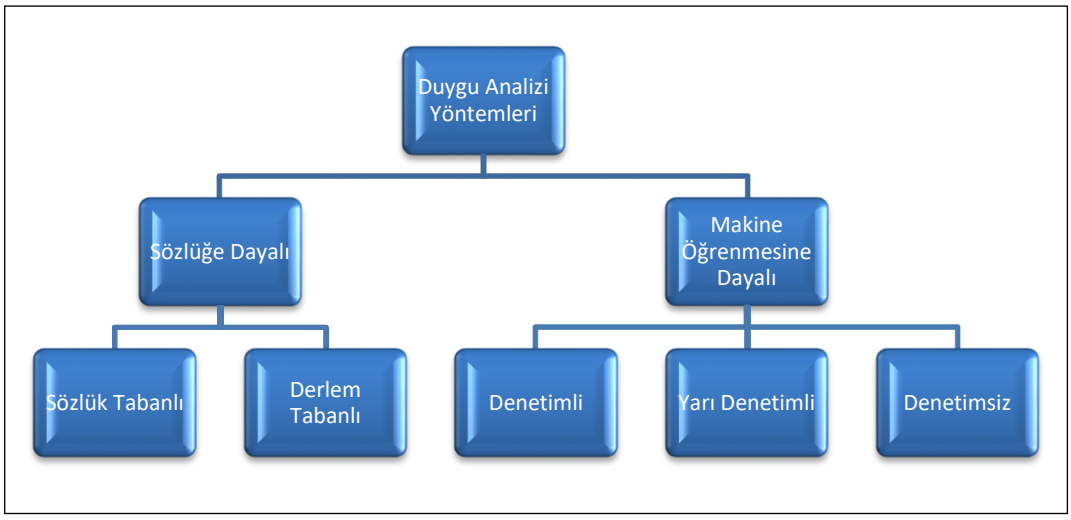
Cümle seviyesinde sınıflandırma, her cümlenin duygu analizini gerçekleştirmeyi amaçlar. İlk adım cümlenin objektif ya da sübjektif olup olmadığını belirlemektir. Eğer cümle sübjektif ise cümlenin olumlu veya olumsuz olduğu belirlenir (Medhat ve diğ. 2014)

Bir metni tüm yönleriyle ele alıp sınıflandırma işlemini gerçekleştirmek için cümle seviyesinde sınıflandırma ve doküman seviyesinde sınıflandırma yöntemleri yeterli değildir. Bunun için özellik temelli duygu analizine ihtiyaç vardır.

Özellik temelli duygu analizinde, diğer sınıflandırma seviyelerine göre yaklaşım biraz daha farklıdır. Özellik temelli (Aspect-level) duygu analizi varlığın tüm yönleriyle ele alınmasını amaçlar. Bir nesnenin belli özelliklerine göre sınıflandırma yapılır. Bir nesnenin hangi bakış açısına göre olumlu ya da olumsuz olduğu önemlidir (Medhat ve diğ. 2014).

4.2 Duygu Analizi Yöntemleri

Duygu analizi alanında yapılan çalışmalar makine öğrenmesine dayalı yöntemler ve sözlüğe dayalı yöntemler olmak üzere iki temel sınıfta incelenebilir (Medhat ve diğ. 2014). Bazı kaynaklarda duygu analizi yöntemlerinde, yukarıda belirtilen iki yaklaşımın birlikte kullanılmasını esas alan hibrit yöntemi de ele alınmıştır. Biz yalnızca iki temel yöntemi ele alacağız. Duygu analizi çalışmalarının uygulama alanına göre iki temel sınıf arasında seçim yapılmaktadır. Örneğin, makine öğrenmesi algoritmalarının basit yapısı, farklı alanlardaki sınıflandırma uygulamalarına kısmen kolay uygulanabilir olmasıyla birlikte istatistiksel yaklaşımların görece daha hızlı çalışması sosyal ağlar gibi akan veri üzerinde gerçek zamanlı olarak yapılan çalışmalarda tercih sebebi olmuştur(Seker 2016).



Şekil 4.1: Duygu Analizi yöntemleri.

Makine öğrenmesine dayalı yöntemde, makine öğrenmesi algoritmaları kullanılarak metin üzerinden istatistiksel özellik çıkarımı yapılır. Elde edilen sayısal değerler karar vermeye destek için kullanılır(Seker 2016).

4.2.1 Sözlüğe Dayalı

Bu yöntemde, doğal dil işleme yöntemlerini kullanılır. İlgili doğal dil (Türkçe, İngilizce gibi) için daha önceden oluşturulmuş görüş sözcükleri içeren görüş sözlüğü kullanılarak anlam bilimsel sonuçlar elde edilerek duygu analizi işlemi gerçekleştirilir. Sözlüğe dayalı görüş madenciliği yöntemleri, yüksek ölçeklenebilirlikleri nedeniyle görüş sınıflandırmada sıklıkla uygulanmaktadır (Onan ve Korukoğlu 2016). Bu yaklaşımda sözlük tabanlı ve derlem tabanlı olmak üzere iki farklı metot bulunmaktadır.

Sözlük tabanlı yöntemde, eş anlamlı ve zıt anlamlı kelimeler kullanılarak metnin görüş kutbu belirlenir (Senthamarai ve Mary 2015).

Derlem tabanlı yöntemde ise görüş kutbu belirlenirken istatistiksel ya da semantik yöntemler kullanılır.

4.2.2 Makine Öğrenmesine Dayalı

Makine öğrenmesine dayalı yöntemde, makine öğrenmesi algoritmaları ve özellik çıkarım yöntemleri kullanılarak duygu analizi hedeflenmiştir. Bu teknikte eğitim ve test için kullanılan iki farklı veri seti bulunmaktadır. Eğitim için kullanılan verinin özellikleri ve çıktı değerleri önceden bilinmektedir. Bazı kaynaklarda eğitim verisi için etiketli veri de denilmektedir. Farklı sınıflandırma algoritmaları kullanılarak eğitim verisi eğitilir. Daha önceden kategorisi ve özellikleri bilinmeyen test verisi bu modele tabi tutularak sınıflandırma işlemi gerçekleştirilir (Amolik ve diğ. 2016). Bundan sonraki bölümlerde makine öğrenmesine dayalı yöntemlerden, denetimli, yarı denetimli ve denetimsiz makine yöntemleri açıklanacaktır.

4.2.2.1 Denetimli Öğrenme

Denetimli öğrenme (supervised learning) yöntemleri var olan etiketlenmiş eğitim verisine dayanır. Etiketlenmiş veride her bir giriş değeri için bir hedef değeri bulunmaktadır. Test verisini eğitmek için eğitim verisinde bulunan çıkış değerlerine göre bir sınıflandırma fonksiyonu(modeli) oluşturulur. Bu model giriş değerleri ile 3çıkış değerleri arasında bir ilişki oluşturur. Öğrenme işlemi gerçekleştirildikten sonra modelin doğruluk değeri test verisiyle kontrol edilir. Modelin doğruluk değeri, test verisindeki doğru sınıflandırma sayısının test kümesindeki toplam örnek sayısına oranıyla belirlenir.

Literatürde Naive Bayes, Destek Vektör Makinesi, Maksimum Entropi, k-En Yakın Komşu, Karar Ağaçları, Lojistik Regresyon ve Lineer sınıflandırma gibi sınıflandırma algoritmaları kullanılarak yapılmış birçok çalışma bulunmaktadır. Bu yöntemleri kullanarak yapılmış en temel çalışma Pang ve diğ. (2002) tarafından film yorumları içeren veri seti üzerinde gerçekleştirilmiştir. Bu çalışmada maximum entropi, destek vektör makineleri ve naive bayes yöntemleri kullanılmıştır. Başarım oranı verinin türüne ve boyutuna bağlı olarak değişiklik gösterdiği için bu çalışmada veriye çeşitli önişlemler uygulanmıştır. Veri setini 1-gram, 1-gram ve 2-gram, 1-gram ve sözcük etiketleri gibi farklı veri seti temsilini kullanarak algoritmalara tabi tutmuşlardır. En yüksek başarım oranını veri 1-gram yöntemi ile temsil edilip destek vektör makineleri yöntemini kullanarak elde etmişlerdir.

Denetimli öğrenme yöntemleri kullanılarak yapılan birçok çalışma literatürde mevcuttur. Farklı veri ve değerlendirme ölçütleri kullanılarak algoritmaların başarım oranları elde edilmiştir. Film değerlendirmeleri, ürün değerlendirmeleri, eğitim değerlendirmeleri, Çince ve Romanca görüş değerlendirmeleri gibi çalışmalar yapılan en popüler çalışmalardandır. Tablo 4.1'de denetimli öğrenme yöntemleri kullanılarak yapılan çalışmalara örnekler detaylı olarak verilmiştir.

Tablo 4. 1: Denetimli öğrenme yöntemiyle yapılan çalışmalar

Yıl	Yöntemler	Veri Alanı	Değerlendirme Ölçütü	Değer
2002	Destek vektör makineleri, Naive Bayes, maksimum entropi, N-gram temsili	Film Değerlendirmesi	Doğru Sınıflandırma	82.90%
2004	Tümcelerin öznel/nesnel olarak sınıflandırılması	Film Değerlendirmesi	Doğru Sınıflandırma	87.00%
2005	Tutum bildiren kelime toplulukları ile anlamsal ilişki temsili. Destek vektör makineleri	Film Değerlendirmesi	Doğru Sınıflandırma	90.20%
2005	Sözcük sırası ve sentaks ilişkilerine dayalı temsil, ağaç yapısı	Film Değerlendirmesi	Doğru Sınıflandırma	93.70%
2007	Viterbi algoritması, tümce/belge seviyesi sınıflandırma	Çevrimiçi Ürün Değerlendirmesi	Doğru Sınıflandırma	82.80%
2007	Görüş kutbu etiketleri, kişilerden elde edilen ek açıklamalar. Destek vektör makineleri	Film Değerlendirmesi	Doğru Sınıflandırma	92.20%
2008	Belge sıklığı ölçütü ki-kare ölçülü, karşılıklı bilgi ve bilgi kazancı yöntemleri, kitle merkezi sınıflandırıcısı, k-en yakın komşu sınıflandıncısı, destek vektör makineleri, Naive Bayes, Winnow sınıflandırma yöntemleri	Çince Görüş Değerlendirmeleri	F- ölçütü (Makro/Mikro)	86.64%
2009	Kural tabanlı smıflandırma.öğreticili öğrenme, makine öğrenmesi (Destek vektör makineleri)	Film ve Urun Değerlendirmesi	F- ölçütü (Makro/Mikro)	91.00%
2010	Çok katmanlı oğreticili mimari, Tümce seviyesi etiketler, Destek vektör makineleri	Film Değerlendirmesi	Doğru Sınıflandırma	93.22%
2010	Sözdizimsel ayrıştırma, görüş sözlüğü, kural tabanlı sınıflandırma	Web forumları	Doğru Sınıflandırma	55.00%
2010	Belge içi ve belgeler arası öznitelikler. Çizge-tabanlı yayılım algoritması, destek vektör makineleri, maksimum entropi, öğreticisiz öğrenme	Kamera Değerlendirmesi	Doğru Sınıflandırma	67.23%
2011	Markov model, Görüş sözlüğü. Tabu arama algoritması, Destek vektör makineleri, Naive Bayes, maksimum entropi	Film Değerlendirmesi	Doğru Sınıflandırma	92.70%
2011	Bire-karşı-tüm destek vektör makinesi, tek-makine çok-sınıflı destek vektör makinesi, Bilgi niteliği çatısı	Ürün Değerlendirmeleri	F- ölçütü (Makro/Mikro)	91.40%
2011	Sınıflandırıcı toplulukları. Naive Bayes,maksimum entropi ve destek vektör makineleri, sözcük tipi bilgisi, sözcük ilişkileri ve özellik ağırlıklandırma yöntemleri	Ürün Değerlendirmeleri	Doğru Sınıflandırma	88.65%
2012	Görüş Sözlüğü, iyileştirilmiş Naive Bayes	Restoran Değerlendirmeleri	Pozitif/Negatif Sınıf Doğru Sınıflandırma yüzdesi arası fark	3.60%
2013	Dilsel özelliklere dayalı öznitelik çıkarımı, TF-IDF terim puanlama, destek vektör makineleri	Sosyal Medya	Doğru Sınıflandırma	90.40%
2013	Destek vektör makineleri, yapay sinir ağları, bilgi kazancı öznitelik çıkarımı	Ürün Değerlendirmeleri	Doğru Sınıflandırma	90.30%
2013	Naive Bayes, maksimum entropi, karar ağacı, k-en yakın komşu. Destek vektör makineleri, bagging, boosting ve random subspace yöntemleri	Film ve Ürün Değerlendirmesi	Doğru Sınıflandırma	92.20%

4.2.2.2 Denetimsiz Öğrenme

Denetimsiz öğrenmede denetimli öğrenmeden farklı olarak sistem eğitilirken etiketsiz veriler kullanılır. Amaç tanıma ve sınıflandırma değildir. Genellikle kümeleme, olasılık yoğunluk tahmini ve boyut indirgeme gibi amaçlarla kullanılmaktadır. Başarım oranı denetimli öğrenme algoritmalarına kıyasla daha düşüktür. Ancak denetimli öğrenme ve yarı-denetimli öğrenme algoritmalarında karşılaşılan alan bağımlılık problemini ortadan kaldırmaktadır.

Denetimsiz öğrenme alanında literatürde birçok çalışma bulunmaktadır. Bu çalışmalardan bazıları Tablo 4.2'de gösterilmiştir. Bu çalışmalardan en temeli Turney ve Littman (2002) tarafından gerçekleştirilen çalışmadır. Bu çalışmada bir belgenin olumlu ya da olumsuz olarak sınıflandırılması, her kelime için belirlenen semantik yönüne bağlı olarak gerçekleştirilmiştir. Kelimelerin semantik yönü belirlenirken yedisi pozitif ("good", "nice", "excellent", "positive", "fortunate", "correct", "superior) ve yedisi negatif ("bad", "nasty", "poor", "unfortunate", "wrong", "inferior") olmak üzere toplamda on dört kelime dikkate alınmıştır. Kelimelerin duygu yönlerini belirlemede, pozitif ve negatif kelimelerle olan ilişkileri önemlidir. Bu ilişkileri belirlemede noktasal karşılıklı bilgi (pointwise mutual information) ve gizli anlamsal çözümleme (latent semantic analysis) yöntemleri kullanılmıştır. Toplamda 3596 (1614 pozitif ve 1982 negatif) kelime ile test işlemi gerçekleştirilerek başarım oranı %80 olarak elde edilmiştir.

Tablo 4. 2: Denetimsiz öğrenme yöntemleriyle yapılan çalışmalar

Yıl	Yöntemler	Veri Alanı	Değerlendirme Ölçütü	Değer
2002	Belgede geçen belirteç ve sıfatlara dayalı yön belirleme, Noktasal Karşılıklı Bilgi, Gizli Anlamsal Çözümleme	General Inquirer Lexicon	Doğru Sınıflandırma	80.00%
2008	WordNet sözcük veritabanı, sözlük tabanlı sınıflandırma	Film, Haber, Blog Değerlendirmeleri	Doğru Sınıflandırma	78.00%
2008	Öğreticisiz öğrenme, Kademeli olarak yeniden eğitime dayalı sözlük geliştirme	Çince Görüş Değerlendirmeleri	F-Ölçütü	87.00%
2008	Öğreticisiz öğrenme, Otomatik sözcük seçimi, Sezgisel bilgi, Tekrarlamalı yeniden eğitim	Çince Görüş Değerlendirmeleri	F-Ölçütü	92.00%
2009	Derlem, Görüş Sözlüğü, Öğreticisiz Öğrenme	Görüş Değerlendirmeleri	F-Ölçütü	89.35%
2015	Öğreticisiz Öğrenme	Film Değerlendirmeleri	Doğru Sınıflandırma	64.50%

4.2.2.3 Yarı-Denetimli Öğrenme

Yarı denetimli öğrenme, denetimli öğrenmede karşılaşılan problemleri çözümlemede tamamlayıcıdır. Denetimli öğrenme yöntemleri, eğitim seti yeterli miktarda büyük olduğu zaman genellikle iyi performans verir. Ancak yeterli miktarda eğitim verisi bulunmadığı zamanlarda yarı-denetimli öğrenme yöntemleri tercih edilmektedir (Abd AL-BNDI 2015). Yetersiz veri miktarının yanı sıra çok boyutlu veri örnekleri de denetimli öğrenme yöntemleri için kısıtlamalar oluşturmaktadır ve performans açısından kötü sonuçlar elde edilmektedir. Yarı-denetimli öğrenme yöntemleri eğitim verisi içerisinde etiketlenmemiş veriye olanak sağlamaktadır.

Yarı denetimli öğrenme yöntemleri kullanılarak yapılan en temel çalışma Aue ve Gamon (2005) tarafından gerçekleştirilmiştir. Görüş sınıflandırma, alana özgü bir problemdir. Bir alanda iyi performans gösteren bir sınıflandırıcı diğer alanda aynı performansı gösteremeyebilir. Yeterli etiketli verinin bulunmadığı alanlarda, sınıflandırıcı eğitimleri için farklı yaklaşımlar kullanılmaktadır. Bu çalışmada dört farklı yaklaşım başarım oranları, avantajları ve dezavantajları bakımından karşılaştırılmıştır. Karşılaştırma işlemi için “movie”, “book”, “product support services” ve “knowledge base” olmak üzere dört farklı kaynaktan elde edilmiş veriler kullanılmıştır. Her doküman özellik vektörü şeklinde ve veriler de 1-gram, 2-gram ve 3-gram şeklinde temsil edilmiştir. Dört farklı yaklaşım için de beklenti maksimizaysonu (expectation-maximization) algoritması en yüksek başarım oranını elde ettiği gözlenmiştir (Aue ve diğ. 2005).

Yarı-Denetimli öğrenme yöntemleri kullanılarak yapılan birçok çalışma literatürde mevcuttur. Farklı veri ve değerlendirme ölçütleri kullanılarak algoritmaların başarım oranları elde edilmiştir. Film değerlendirmeleri, ürün değerlendirmeleri, eğitim değerlendirmeleri, Çince ve Romanca görüş değerlendirmeleri gibi çalışmalar yapılan en popüler çalışmalardandır. Tablo 4.3’te yarı denetimli öğrenme yöntemleri kullanılarak yapılan çalışmalara örnekler detaylı olarak gösterilmiştir.

Tablo 4. 3: Yarı denetimli öğrenme yöntemlerinin kullanıldığı çalışmalar

Yıl	Yöntemler	Veri Alanı	Değerlendirme Ölçütü	Değer
2005	Başka alandaki etiketli verilerin eğitimde kullanılması, Sınıflandırıcı Toplulukları, Beklenti- Maksimizasyonu	Film ve Ürün Değerlendirmeleri	Doğru Sınıflandırma	82.39%
2007	Kosinüs benzerlik ölçütü, Benzerlik Sıralama Yöntemi, Bağıl Benzerlik Sıralama Yöntemi, Transduktif Destek Vektör Makineleri	Bilgisayar, Eğitim ve Ev Değerlendirmeleri	Doğru Sınıflandırma	89.93%
2007	Yapısal yazışma öğrenme, karşılıklı bilgi ölçütü	Ürün Değerlendirmeleri	Doğru Sınıflandırma	85.90%
2007	İngilizce görüş sözlüğü, duygu analizi araçları, paralel derlem	Romanca Görüş Değerlendirmesi	F-ölçütü	72.68%
2008	Farklı alanlardaki özellik setlerinin bir araya getirilmesi, farklı alanlardaki veri setleri üzerinde sınıflandırıcı eğitimi, Meta Öğrenme Makine Çevirisi, Açıklama Eklenmiş derlem	Ürün Değerlendirmeleri	Doğru Sınıflandırma	85.00%
2008	Görüş Açıklamaları Ekleme, Naive Bayes, Destek vektör makineleri	Romanca Görüş Değerlendirmesi	F-ölçütü	69.44%
2009	Spektral Kümeleme, Aktif Öğrenme, Transdüktif Öğrenme	Film ve Ürün Değerlendirmeleri	Doğru Sınıflandırma	76.20%
2009	Eş-eğitim, İngilizce görüş ifadeleri, etiketli İngilizce değerlendirme ifadeleri	Çince Görüş Değerlendirmeleri	Doğru Sınıflandırma	81.30%
2010	Hedef nesneye ilişkin kişisel duygu, tercih, tutum ifadeleri, nesnel-alana özgü bilgiler, eş-öğretim	Ürün Değerlendirmeleri	Doğru Sınıflandırma	86.75%
2011	Kendi-kendine eğitim, Veri Sözlüğü, alana özgü özellikler	Film Değerlendirmesi, Çok Alanlı Görüş Değerlendirmesi	Doğru Sınıflandırma	75.00%
2011	Ortak görüş-konu modeli, gizli Dirchlet tahsis modeli. Eş Zamanlı Görüş ve Konu Belirleme	Film Değerlendirmesi, Çok Alanlı Görüş Değerlendirmesi	Doğru Sınıflandırma	90.00%
2013	Öznellik, Görüş Kutbu ve Etkileme Durumu Belirleme, Bayes Ağları, Beklenti Maksimizasyonu	Asomo veri seti	Doğru Sınıflandırma	83.63%
2014	Aktif Öğrenme, Yan-öğreticili eş-eğitim	Kitap Değerlendirmesi	Doğru Sınıflandırma	82.17%
2014	Yarı-öğreticili öğrenme, çoğunluğun azınlığı eğitmesi kuralı	Kitap Değerlendirmesi	Doğru Sınıflandırma	81.00%

5. MATERYAL VE YÖNTEM

Bu bölümde tezde kullanılan yöntemler ve araçlar açıklanmıştır. Makine öğrenmesine dayalı duygu analizi için kullanılan tüm algoritmalar özellikleriyle birlikte detaylandırılmıştır. Makine öğrenmesi denetimli öğrenme algoritmalarından olan üç farklı algoritma üzerinde duygu analizi işlemi gerçekleştirilmiş olup başarım oranları karşılaştırılmıştır. Algoritmaların başarım oranları da bir sonraki bölümde verilmiştir. Tezde, Databricks firmasının SAS olarak sunmuş olduğu Spark platformu üzerinde makine öğrenmesi algoritmalarıyla duygu analizi işlemi gerçekleştirilmiştir. 6GB'lık bir bellek ve tek bir ana düğüm (master node) üzerinde işlemler gerçekleştirilmiştir.

5.1 Veri Seti

Bu çalışmada, duygu analizinin gerçekleştirilmesi için hazır veri seti kullanılmıştır. Toplamda 57650 İngilizce şarkı sözü kullanılarak duygu analizi gerçekleştirilmiştir. www.kaggle.com sitesinden csv formatında çekilen veri seti, veri önişleme aşamalarından geçirilerek makine öğrenmesine dayalı duygu analizine tabi tutulmuştur.

Tablo 5.1'de veride bulunan toplam şarkı sayısı, artist sayısı, en olumlu ve en olumsuz şarkıların detayları gösterilmiştir. Veri toplamda 57650 adet satırdan oluşmaktadır. Ancak aynı şarkılar farklı sanatçılar tarafından da söylendiği için 57494 farklı şarkı sözü bulunmaktadır.

Tablo 5. 1: Veri setinin detayları

	Artist	Şarkı	Link
Adet	57650	57650	57650
Farklı	643	57494	57494
En Olumlu Şarkı	John Martyn	CoolTide	/j/john+martyn/cooltide_20823887.html
En Olumsuz Şarkı	Gucci Mane	I Shook Them Haters Off	/g/gucci+mane/i+shook+them+haters+off_20736743.html

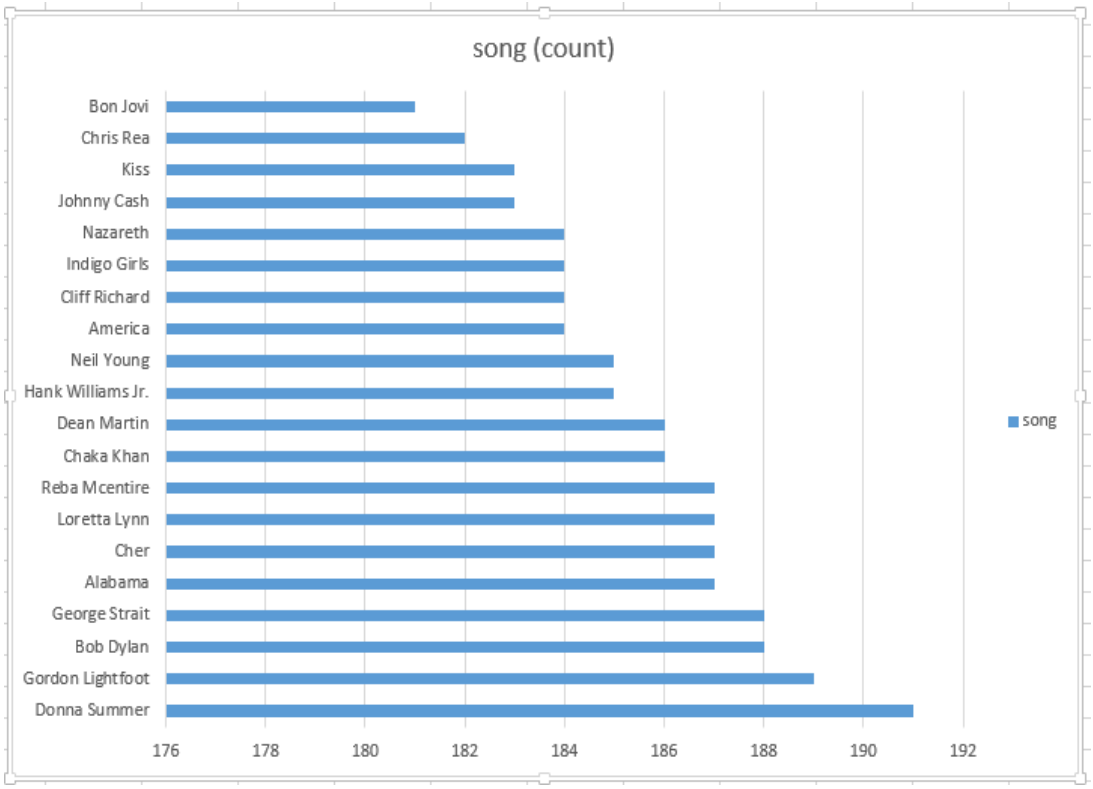
Tablo 5. 2: En olumlu yirmi şarkının detayları

artist	song	text	sentiment	score
John Martyn	CoolTide	So cool what a cool time It is so cool what a cool…	positive	89
Lil Wayne	Hot Boy	[Intro Fuck wrong with you Verse 1 Come through…	positive	87
Nicki Minaj	Super Bass	This one is for the boys with the boomin system …	positive	87
Fabolous	I Shine You Shine	Them other niggas is cool but I just got that glow…	positive	83
Kiss	Do You Love Me	You really like my limousine You like the way …	positive	80
Cat Stevens	Ready	I love I love I am ready to love yes I love I love …	positive	78
Fatboy Slim	Fat Boy Slim - Right Here Right Now	Right here right now right here right now Right …	positive	77
Christina Aguilera	Beautiful People	Burlesque Beautiful beautiful beautiful beautiful…	positive	72
Kirk Franklin	A God Like You	Everybody wanna be like you they Want power…	positive	70
Quincy Jones	Stuff Like That	Walked in the joint They were lined up back to ….	positive	70
Lauryn Hill	Turn Your Lights Down Low	(With Bob Marley Bob Marley Lauryn - Uh Turn …	positive	70
Rod Stewart	It Was Love That We Needed	It was love that we needed Hmm hmm We needed…	positive	66
Diana Ross	If You are Not Gonna Love Me Right	Hey oh Hey yeah Hey baby hey baby The phone …	positive	64
R. Kelly	Just Like That	Oh baby If I could Explain the joy I feel Oh If I …	positive	64
Ellie Goulding	Love Me Like You Do	[Verse 1 You are the light you are the night You…	positive	64
Kanye West	Diamonds	We the cause of all the commotion Your mouth …	positive	63
James Taylor	How Sweet It Is	How sweet it is to be loved by you How sweet it…	positive	62
Selena Gomez	Like A Champion	Walk like a champion talk like a champion Ram …	positive	62
Who	Here tis	Whoa whoa whoa whoa whoa whoa) I said whoa…	positive	61
Mud	Tiger Feet	Yeah yeah All night long you have been looking …	positive	61

Tablo 5. 3: En olumsuz yirmi şarkının detayları

artist	song	text	sentiment	score
Gucci Mane	I Shook Them Haters Off	[Chorus I shook dem haters off I shook dem haters off …	negative	-74
Devo	S.I.B. Swelling Itching Brain)	Gotta nervous kind of feeling Gotta painful yellow …	negative	-68
Michael Jackson	2 Bad	Told me that you are doin wrong Word out shockin all …	negative	-66
Usher	Bump	[Intro Lil Jon At-at what at-at-at what At-at-at what at …	negative	-66
Fatboy Slim	In Heaven	Fatboy Slim is fucking in heaven Fatboy Slim is fucking in...	negative	-66
Yoko Ono	Midsummer New York	Wake up in the morning my hands cold in fear…	negative	-65
Chris Brown	100 Bottles	We are in the mother fucking building! A hundred fucking...	negative	-64
Insane Clown Posse	I did not Mean To Kill Em	This is the story of a murderer A cold blooded killer a…	negative	-62
Vanilla Ice	Dirty South	Chorus Here come the south shit dirty south shit…	negative	-61
Flo-Rida	Broke It Down	This time we going in Gonna get get what get wild…	negative	-60
Dolly Parton	Go To Hell	GO TO HELL WRITER DOLLY PARTON Go to Hell go to …	negative	-59
Insane Clown Posse	Bugs On My Nuts	Well I don t understand the phenomenon We fucking…	negative	-56
Metallica	St. Anger	Saint Anger round my neck Saint Anger round my …	negative	-56
Stevie Wonder	All Day Sucker	Come on up you say Cause you can feel your …	negative	-55
Pitbull	Damn It Man	Damn it man them just D-damn it man pitbull D-d-damn …	negative	-55
The Weeknd	Live For	Getting sober for a day got me feeling too low …	negative	-54
Rihanna	Disturbia	Bum bum be-dum bum bum be-dum bum Bum bum…	negative	-53
Rihanna	Man Down	I did not mean to end his life I know it wasn t right I …	negative	-51
Dusty Springfield	Silly Silly Fool	Such a silly silly silly silly fool am I Oh I just a silly …	negative	-51
Depeche Mode	Wrong	I was born with the wrong sign In the wrong house With …	negative	-51

Tablo 5.2 ve Tablo 5.3'te pozitif ya da negatif olarak etiketlenen verinin skorlarına göre en olumsuz yirmi şarkı ve en olumlu yirmi şarkının detayları gösterilmiştir.



Şekil 5.1: Şarkı sayılarına göre sanatçıların grafik gösterimi

Şekil 5.1'de en çok şarkısı bulunan yirmi sanatçı grafik şeklinde gösterilmiştir. Grafiğe göre 191 adet şarkıya sahip olan Donna Summer en çok şarkısı bulunan sanatçıdır.

Sınıflandırma işlemi, Spark'ın desteklemiş olduğu MLlib kütüphanesinde bulunan makine öğrenmesi algoritmalarıyla gerçekleştirilmiştir. MLlib'in amacı, pratik makine öğrenmesini ölçeklenebilir ve kolay hale getirmektir. Spark Core'a benzer şekilde, üç farklı dilde API sunar: Python, Java ve Scala. Apache Spark 1.2'de Databricks, AMPLab ile birlikte pratik ML boru hatlarının (pipeline) kolay oluşturulması ve ayarlanması için MLlib'e bir boru hattı API'si sunmuştur. Bunun için de aşağıdaki adımlar izlendi:

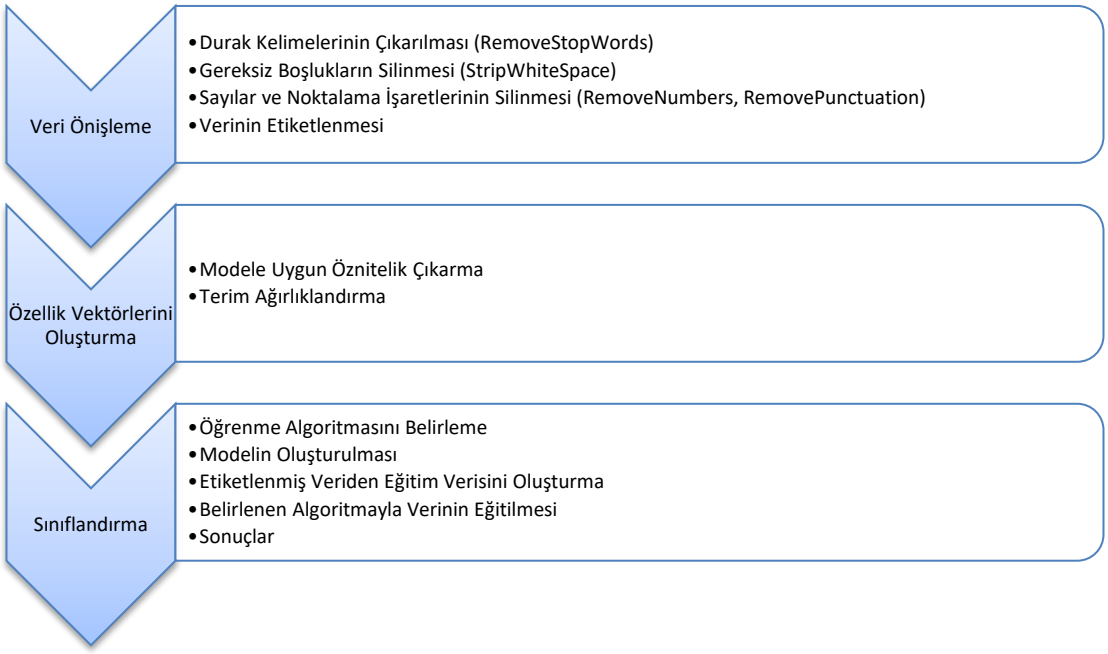
- Her doküman kelimelere ayrıldı
- Her dökümanın kelimeleri özellik vektörüne dönüştürüldü

- Özellik vektörleri ve etiketleri kullanılarak oluşturulan tahminleme modeli eğitildi.

5.1.1 Verinin Hazırlanması

Duygu analizi gerçekleştirimi için verinin uygun hale getirilmesi gerekmektedir. Bunun için veri bazı işlemlere tabi tutulur. Şekil 5.2'de duygu analizi gerçekleştiriminde kullanılacak olan verinin hazırlanması için gerekli adımlar verilmiştir.

Verinin makine öğrenmesine dayalı duygu analizi için belirli işlemlerden geçirilmesi gerekmektedir. Bu işlemler üç ana başlık altında ele alınmıştır. Veri önişleme, özellik vektörlerinin oluşturulması ve sınıflandırma adımlarından sonra duygu analizi işlemi tamamlanmış olmaktadır. Bu üç farklı adım aşağıda detaylandırılmıştır.



Şekil 5.2: Verinin hazırlanması için uygulanan adımlar

5.1.2 Veri Önişleme

Metinler üzerinde yapılan duygu analizi işleminde, verinin kalitesi önemli bir faktördür. Verinin kalitesine bağlı olarak yapılan analizin başarısı da değişmektedir (Çoban 2015). Bu nedenle veri önişleme adımları duygu analizi için en önemli adımlardan biridir. Bu aşamada, analiz sonucunu yanlış yönlendirecek verinin temizlenmesinin yanı sıra veri duygu analizi için uygun formata da dönüştürülmüş olmaktadır.

Bu tezde veri önişleme adımları R dili kullanılarak RStudio'da gerçekleştirilmiştir. R web sitesinden temin edilebilen paketler, önceden yazılmış fonksiyonlara sahiptir. Ara yüz aracılığıyla da yüklenebilen bu paketlerle yalnızca gerekli olanla çalışılarak daha az bellek kullanımı ve hızlı işlem gücü sağlanmış olur. Metin madenciliği uygulamalarında yaygın olarak kullanılan "tm" paketi verinin temizlenmesi ve duygu analizi işlemlerine hazır hale getirilmesi için kullanılmıştır. Bu pakette verinin temizlenmesi, bir fonksiyonu verinin tüm unsurlarına uygulayan tm_map() işlevi ile yapılır.

Durak kelimelerin çıkarılması:

Durak kelimeler (stopwords), genellikle bir dildeki tek başına herhangi bir anlam ifade etmeyen ancak son derece yaygın olarak kullanılan kelimelerdir. İngilizce dili için bu kelimelere örnek olarak to be, do, the, and, is gibi çok sık kullanılan kelimeler verilebilir. Tüm doğal dil işleme araçlarıyla kullanılan durdurma kelimelerinin tek bir evrensel listesi yoktur. Bu tezde kullanılan verideki durak kelimeler, R programlama dilinin sahip olduğu "tm" yani "Text Mining" paketinde bulunan fonksiyonlar kullanılarak temizlenmiştir. Paket içerisinde İngilizce dili için "stopwords" listesi bulunmaktadır.

Gereksiz Boşlukların Silinmesi

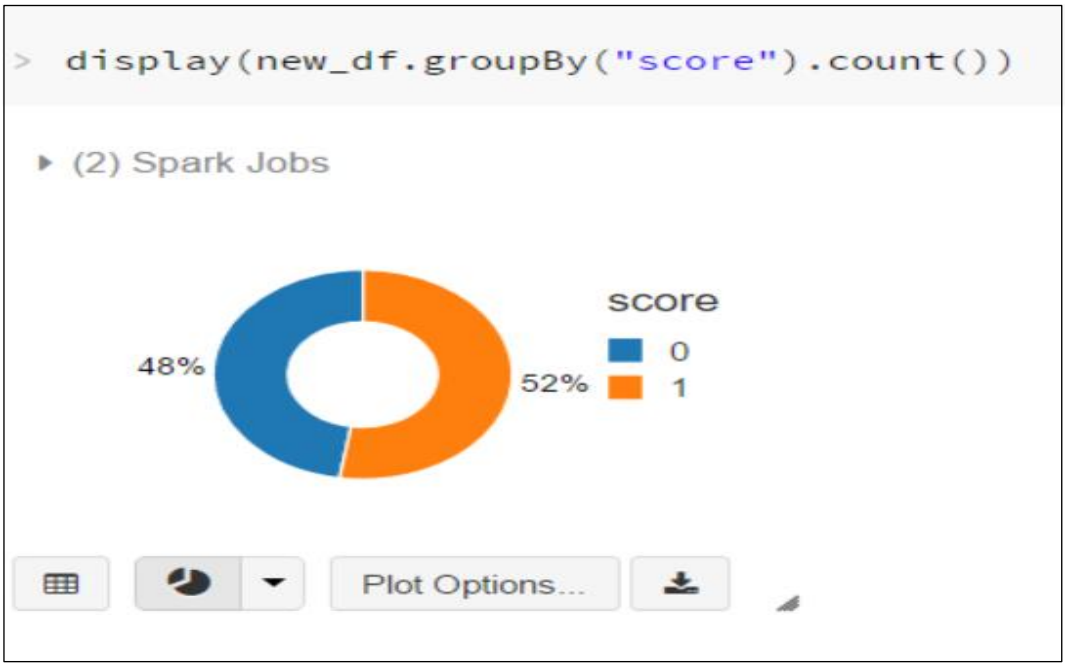
stripWhitespace metodu ile birden fazla boşluk karakteri tek bir boşluğa daraltılarak kelimeler arasındaki boşluklar eşitlenmiştir ve content_transformer(tolower) metoduyla da büyük harfler küçük harflere dönüştürülmüştür.

Sayılar ve Noktalama İşaretlerinin Silinmesi

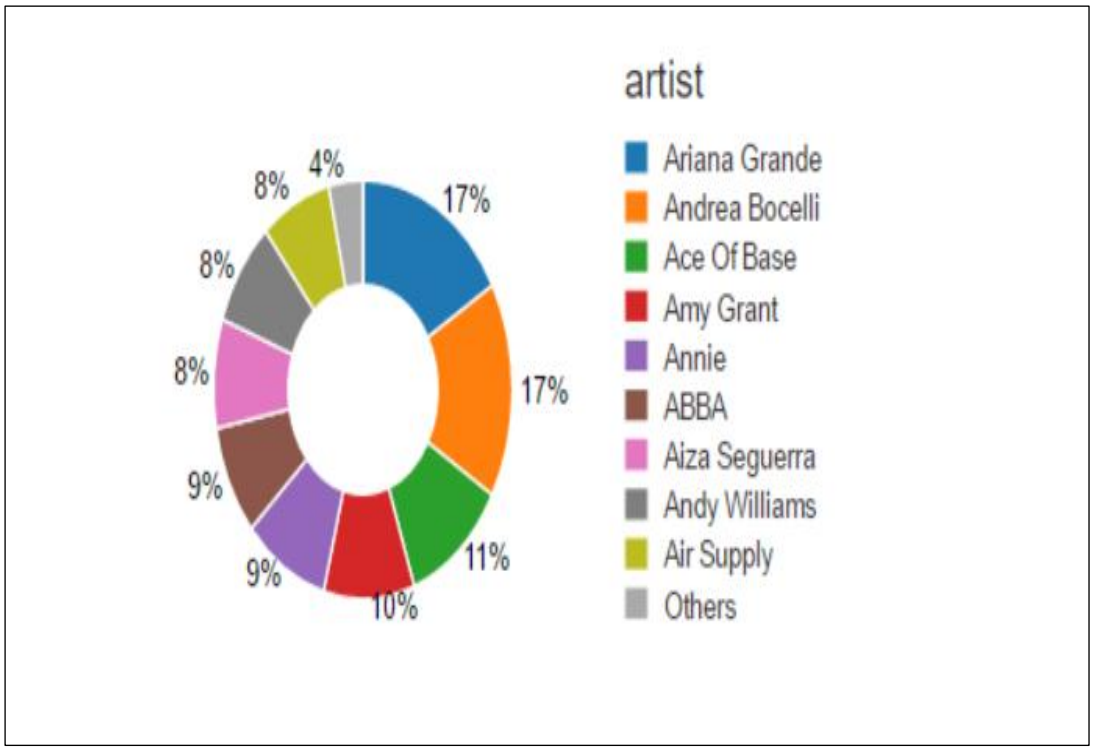
"tm" paketinde yer alan removeNumbers ve removePunctuation fonksiyonlarıyla da duygu analizi sonucunda etkili olmayan farklı semboller, tüm noktalama işaretleri, İngilizce olmayan harfler ve anlamsız karakterler kaldırılmıştır.

Verinin Etiketlenmesi

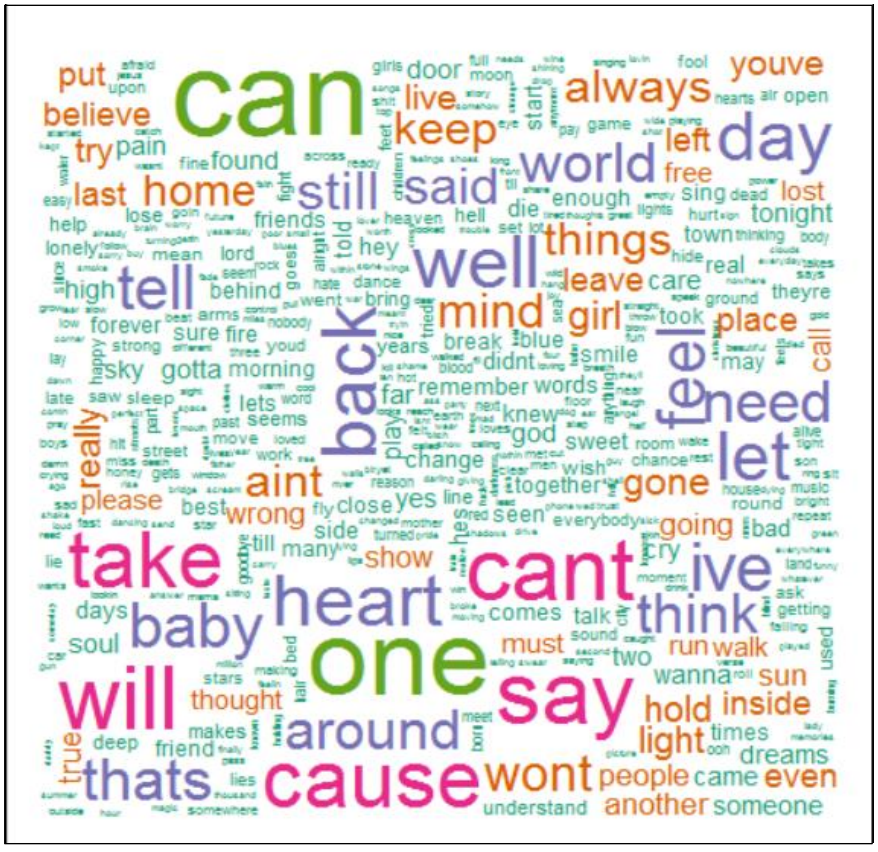
Veriyi parçalayarak veriden anlamlı bilgi elde etmeyi amaçlar. Veri, kelimelerine ayrılarak her bir şarkı sözü için verinin olumlu ya da olumsuz olduğuna bakılmıştır. StringR paketinde bulunan str_split metoduyla parçalarına ayrılan veri, 2006 pozitif kelime bulunan text dosyasındaki ve 4783 adet negatif kelime bulunan text dosyasındaki kelimelerle karşılaştırılarak her bir şarkı sözünün skor değeri belirlenmiştir. Skor belirleme işlemi pozitif kelimelerin sayısından negatif kelimelerin sayısı çıkartılarak elde edilmiştir. Skor değeri sıfır ve daha küçük değerler için negatif, sıfırdan büyük değerler için de pozitif olarak etiketlenmiştir. Verinin %52'lik kısmı pozitif, %48'lik kısmı ise negatif olarak etiketlenmiştir.



Şekil 5.3: Verinin olumluluk ve olumsuzluk oranı



Şekil 5.4: Artistlerin şarkı sayılarına göre şarkıların olumluluk oranları



Şekil 5.5: Kelime bulutu

artist	song	text	sentiment	label
ABBA	Ahe is My Kind Of...	Look at her face ...	positive	3.0
ABBA	Andante Andante	Take it easy with...	positive	1.0
ABBA	As Good As New	I will never know...	positive	17.0
ABBA	Bang	Making somebody h...	positive	7.0
ABBA	Bang-A-Boomerang	Making somebody h...	positive	7.0
ABBA	Burning My Bridges	Well you hoot and...	negative	-3.0
ABBA	Cassandra	Down in the stree...	negative	-8.0
ABBA	Chiquitita	Chiquitita tell m...	negative	-4.0
ABBA	Crazy World	I was out with th...	negative	-5.0
ABBA	Crying Over You	I am waitin for y...	negative	-2.0
ABBA	Dance	Oh my love it mak...	positive	4.0
ABBA	Dancing Queen	You can dance you...	positive	2.0
ABBA	Disillusion	Changing moving i...	negative	-8.0
ABBA	Does Your Mother ...	You are so hot te...	positive	12.0
ABBA	Dream World	Agnetha We are no...	positive	5.0
ABBA	Dum Dum Diddle	I can hear how yo...	negative	-5.0
ABBA	Eagle	They came flying ...	negative	0.0
ABBA	Every Good Man	Every good man ne...	positive	6.0
ABBA	Fernando	Can you hear the ...	positive	3.0

Şekil 5.6: Önişlemden geçirilmiş veri

2014 yılında Apache Spark açık kaynak kodlu büyük veri deposunu kuran Databricks firması Spark'ı bulut platformunda SAS olarak sunuyor. Databricks, Apache Spark'ın üstünde, kullanıcıların gelişmiş analitik çözümlerini kolayca kurmasına ve yerleştirmesine olanak tanıyan gerçek zamanlı bir veri platformu sağlar. Bu tezde de Databricks'in bulut platformunda sunmuş olduğu Spark kullanılmıştır. Veri etiketleme işlemi bittikten sonra veri, makine öğrenmesi algoritmaları ile eğitilmesi için Databricks platformuna yüklenmiştir. İkili (binary) sınıflandırma işlemi için veri yüklendikten sonra verinin "label" alanındaki değerler 0 ya da 1 olarak değiştirilmiştir. Şekil 5.7'de label alanının ikili sınıflandırma için uygun formata dönüştürülmüş hali gösterilmektedir.

▸ (1) Spark Jobs

artist	song	text	sentiment	label
ABBA	Ahe is My Kind Of...	Look at her face ...	positive	1.0
ABBA	Andante Andante	Take it easy with...	positive	1.0
ABBA	As Good As New	I will never know...	positive	1.0
ABBA	Bang	Making somebody h...	positive	1.0
ABBA	Bang-A-Boomerang	Making somebody h...	positive	1.0
ABBA	Burning My Bridges	Well you hoot and...	negative	0.0
ABBA	Cassandra	Down in the stree...	negative	0.0
ABBA	Chiquitita	Chiquitita tell m...	negative	0.0
ABBA	Crazy World	I was out with th...	negative	0.0
ABBA	Crying Over You	I am waitin for y...	negative	0.0
ABBA	Dance	Oh my love it mak...	positive	1.0
ABBA	Dancing Queen	You can dance you...	positive	1.0
ABBA	Disillusion	Changing moving i...	negative	0.0
ABBA	Does Your Mother ...	You are so hot te...	positive	1.0
ABBA	Dream World	Agnetha We are no...	positive	1.0
ABBA	Dum Dum Diddle	I can hear how yo...	negative	0.0
ABBA	Eagle	They came flying ...	negative	0.0
ABBA	Every Good Man	Every good man ne...	positive	1.0
ABBA	Fernando	Can you hear the ...	positive	1.0

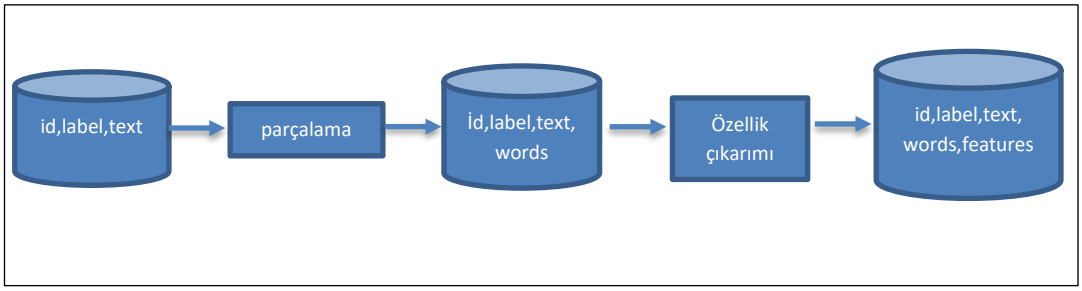
Şekil 5.7: Label alanının uygun formata dönüştürülmüş hali

5.1.3 Özellik Vektörlerinin Oluşturulması

Makine öğrenmesi algoritmalarının gerçekleştirimi için kullanılan MLlib Kütüphanesi birkaç özel veri tipini desteklemektedir. Bunlardan bir tanesi de vektörlerdir. Sınıflandırma, regresyon ve kümeleme gibi birçok makine öğrenme algoritmaları için veri, genellikle bir özellik vektörü olarak temsil edilir. Bu nedenle bu özellik vektörlerinin oluşturulması önemli bir adımdır. Vektör oluşturma işlemi sonucunda algoritma öğrenme kararını temsil eden bir model oluşturulur.

Vektörün her koordinatı, veri noktasının belirli bir özelliğine karşılık gelir (Dave ve diğ. 2003). Spark'ın makine öğrenme modülü olan MLlib, yoğun (dense) ve seyrek (sparse) olmak üzere iki tür vektörü destekler. Yoğun bir vektör, giriş değerlerini temsil eden çift bir dizi ile desteklenirken, seyrek bir vektör, iki paralel diziyle desteklenir (dizinler ve değerler). Örneğin; [1.0 , 0.0 , 3.0] yoğun vektör, (3 ,

[0 , 2] , [1.0 , 3.0]) ise seyrek vektöre örnektir. Seyrek vektördeki ilk değer olan 3 vektörün boyutunu temsil eder.



Şekil 5.8: Özellik vektörü oluşturma evreleri

Yukarıdaki şekilde özellik çıkarımı için gerçekleştirilen adımlar gösterilmiştir. Farklı özellik çıkarımı yöntemleri bulunmaktadır. Bunlardan metin madenciliğinde ve sınıflandırma algoritmalarında en sık kullanılanı bilgi getirimi (information retrieval) yöntemidir (Boiy ve diğ. 2007). Bu tezde de düşük hesaplama maliyetleri ve kolay uygulanabilirlikleri açısından bilgi getirimi konu başlığı altında yer alan TF-IDF yöntemi kullanılmıştır. Veri setlerindeki kirliliği azaltmak için kullanılan bu yöntem, cümlelerde yer alan kelimelerin ne sıklıkla kullanıldığını ve diğer dokümanlarda geçme sıklıklarını birlikte hesaplayarak, kategoriler için en önemli kelimeleri belirler.

TF-IDF

TF-IDF (Term Frequency- Inverse Document Frequency) kavramı, metin madenciliğinde yaygın olarak kullanılan ve bir terimin bir belgede bulunan cümle içerisindeki önemini yansıtacak şekilde kullanılan bir özellik vektörizasyon yöntemidir. Bir d belgesindeki t teriminin kaç kez gösterildiği TF(t,d) ifadesiyle, D derlemindeki (çok sayıdaki metnin düzenli ve yapısal olarak bir arada bulunması durumu, corpus) belge frekansı DF(t,D) ifadesiyle gösterilir. TF-IDF ters belge frekansı ve terim frekansı olmak üzere iki istatistiğin çarpımıdır. Her iki istatistiğin kesin değerlerini belirlemenin çeşitli yolları vardır.

- Terim Frekansı (TF): Bir terimin bir belgede ne sıklıkta oluştuğudur. Her belge farklı uzunlukta olduğundan, bir terimin uzun belgelerde daha kısa belgelerden çok daha fazla görünmesi mümkündür. Dolayısıyla, terim frekansı normalleştirme yöntemi olarak sıklıkla belge uzunluğuna (belgedeki toplam terim sayısı) bölünür.

TF(t) = (t teriminin belgede kaç kez geçtiğinin sayısı) / (Belgedeki toplam kelime sayısı)

- Ters Metin Frekansı (IDF): Bir terimin ne kadar önemli olduğunu ölçer. Terim frekansı (TF) hesaplanırken her kelimenin önem derecesi eşit olarak alınır. Bununla birlikte bazı terimlerin çok fazla görülebileceği ancak çok az önemi olduğu bilinmektedir. Her dilde bu tarz kelimeler mevcuttur ve tek başlarına bir anlam ifade etmezler. İngilizce dili için "is", "of" ve "that" gibi kelimeler örnek verilebilir. Bu nedenle, nadir olanları ölçeklendirirken sık terimleri de tartmamız gerekir.

$$IDF(t) = \log \frac{Toplam\ döküman\ sayısı}{Terimi\ içeren\ doküman\ sayısı} \tag{5.1}$$

Yukarıdaki formülde kullanılan logaritma tabanının bir önemi yoktur. Burada önemli olan üstel fonksiyonun tersi yönünde hesap yapmak olduğundan logaritma kökü e, 2 veya 10 gibi değerler olabilir. "Terimi içeren doküman sayısı" değerinin 0 (sıfır) olma ihtimali vardır. Böylesi bir durumda sonuç sıfıra bölüm belirsizliğine götürebileceğinden sıkça yapılan programlama yaklaşımlarından biri de bu değere 1 eklemektir. Ters terim frekansında terimin farklı dokümanlarda kaç kez geçtiği de önemli bir rol oynamaktadır. Eğer terim az miktarda dokuman içerisinde yüksek miktarda geçiyorsa TF-IDF değeri yüksek, her dokümanda geçiyorsa TF-IDF değeri en düşük değerini alır. Yukarıdaki verilen formülleri açıkça belirtmek mümkündür:

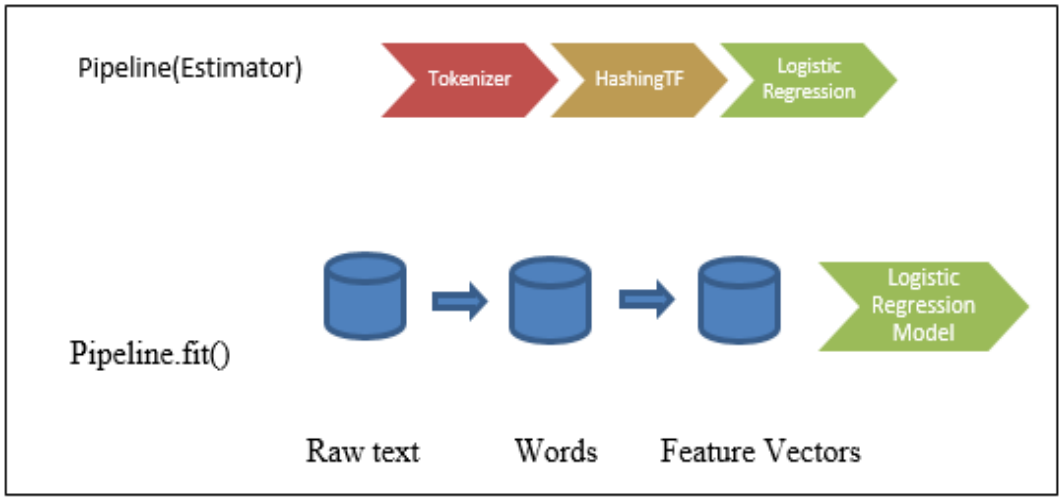
$$w_{i,d} = \mathrm{t}f_{i,d} \times \log \frac{n}{df_i} \tag{5.2}$$

i terimi için ve d dokümanı için yukarıdaki gibi hesaplama yapılmaktadır. Formülde n ile toplam doküman sayısı, df ise doküman frekansını (i teriminin kaç farklı dokumanda geçtiğinin sayısını) vermektedir. Spark ortamında, TF-IDF hesaplaması için MLlib üzerinde hazır olarak sunulan kütüphaneler bulunmaktadır. Bu tezde de Spark'ın sunmuş olduğu kütüphanelerden yararlanılmıştır.

5.1.4 Sınıflandırma

Bu tezde makine öğrenmesi algoritmalarının etkin bir şekilde işletilebilmesi için makine öğrenmesi boru hattı API'sı(ML Pipeline API) kullanılmıştır. Bir makine öğrenmesi iş akışını belirtmek için birden fazla dönüştürücü ve tahminleyiciyi bir araya getiren boru hattı(pipeline), DataFrame'lerin üzerine kurulmuş üst düzey API'lerin bir setini sağlar. Bu API, "spark.ml" adında bir paket kapsamında bulunmaktadır. Bir boru hattı, dönüştürücü(transformer) ve tahminleyici(estimator) olmak üzere bir dizi aşamadan oluşur. Dönüştürücü, veri kümesini girdi olarak alır ve çıktı olarak genişletilmiş veri kümesi üretir. Bir ayrıştırıcı(tokenizer), bir veri setini kelimelerine ayrılmış bir veri kümesine dönüştüren bir dönüştürücüdür. Tahminleyici(estimator), bir model oluşturabilmek için önce giriş veri setine uymalıdır. Örneğin; lojistik regresyon, etiketleri ve özellikleri olan bir veri setinde eğitilerek lojistik regresyon modeli üreten bir tahminleyicidir.

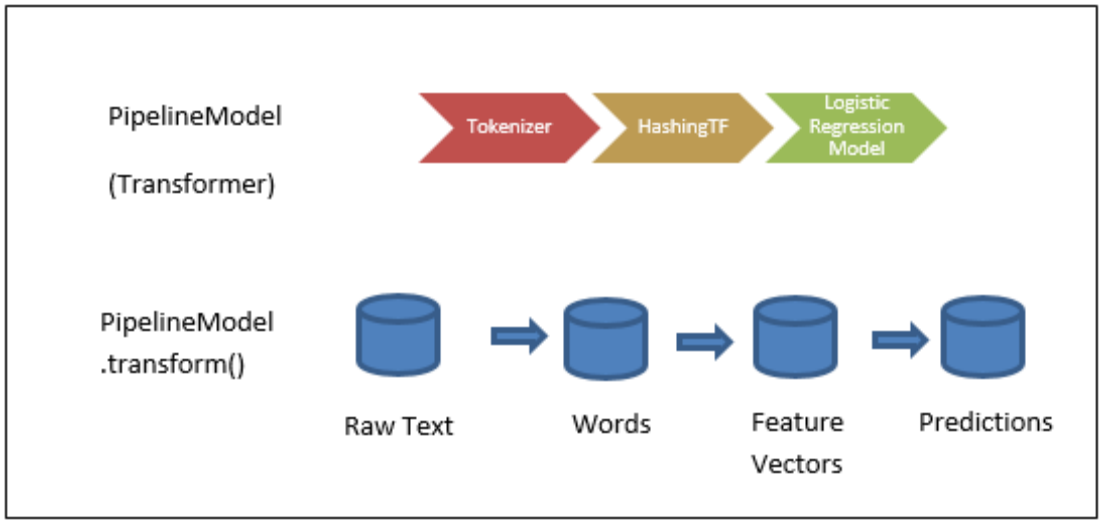
Aşamalar dizisi olarak belirtilen boru hattında veri seti, her bir aşamadan geçtikçe dönüştürülür. Dönüştürücü(Transformatör) aşamaları için transform() metodu, tahminleyici(Estimator) aşamaları için bir Transformer üretmek için fit() yöntemi çağırılır. Tezde kullanılan algoritmalardan biri olan lojistik regresyonun boru hattı üzerindeki işleyişi Şekil 5.9'de gösterilmiştir.



Şekil 5.9: Boru hattı işleyişi (Estimator)

Şekil 5.9'de ilk satırda üç aşama gösterilmiştir. İlk ikisi (Tokenizer ve HashingTF) dönüştürücü üçüncü (Logistic Regression) bir tahminleyicidir. Alt satırda

ise verinin boru hattı üzerindeki akışı gösterilmiştir. Ham metin belgeleri ve etiketler içeren orijinal DataFrame'de Pipeline.fit() yöntemi çağırılır. Tokenizer.transform() metodu ham veriyi kelimelerine ayırarak, Dataframe'e kelimelerin olduğu yeni bir sütun ekler. HashingTF.transform() metodu da bir önceki adımda eklenen kelime sütununu özellik vektörüne dönüştürür ve DataFrame'e bu vektörlerin bulunduğu yeni bir sütun ekler. Logistic Regression bir tahminleyici olduğundan, boru hattı bir logistic regression model oluşturmak için LogisticRegression.fit() yöntemini çağırır. Bu işlemler boru hattının eğitim aşaması için kullanılan adımlarıdır. Boru hattının fit() yöntemi çalıştırıldıktan sonra, bir dönüştürücü olan PipelineModel oluşturur. Bu boru hattı modeli test zamanında kullanılır. Şekil 5.10'da bu kullanım gösterilmektedir.



Şekil 5.10: Boru hattı işleyişi (Transformator)

Şekil 5.10'da gösterildiği gibi PipelineModel orijinal boru hattıyla aynı sayıda aşamaya sahiptir, ancak orijinal boru hattındaki tüm tahminleyiciler dönüştürücüler haline gelmiştir. Bir test veri kümesinde PipelineModel.transform() yöntemi çağırıldığında, veriler uygun boru hattından sırayla geçirilir. Her aşamadaki transform() yöntemi, veri kümesini günceller ve bir sonraki aşamaya geçirir. Veri tüm aşamalardan geçtikten sonra en son tahminler oluşturulur. Tahminleme sonucuna göre de algoritmanın başarım oranları elde edilir.

5.2 Uygulanan Yöntemler

5.2.1 Naive Bayes

Naive Bayes algoritması, uygulanabilirliği ve performansı açısından yıllarca popüler makine öğrenme yöntemlerinden biri olmuştur. Naive Bayes sınıflandırıcısı, Bayes teoremine dayanan istatistiksel bir yöntemdir. Basit bir ifadeyle, Naive Bayes sınıflandırıcısı, bir sınıfın özelliğinin varlığının (veya yokluğunun) başka herhangi bir özelliğin varlığı (veya yokluğu) ile ilgili olup olmadığını varsayar. Sınıflandırılması gereken kümeler ve bir örneğin hangi sınıfa ait olduğu önceden bilinmektedir.

Bu algoritmanın bir avantajı, sınıflandırma için gerekli olan parametreleri (değişkenlerin ortalama ve varyanslarını) tahmin etmek için az miktarda eğitim verisine ihtiyacı olmasıdır (Bhargavi ve diğ. 2009). Eğitim verisinde bulunan her bir özelliğin koşullu olasılık dağılımını hesaplar ve daha sonra verilen koşullu olasılık dağılımını hesaplamak için Bayes teoremi uygular ve bunu tahmin için kullanır.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{5.3}$$

P(A|B): B olayınının gerçekleştiği durumda A olayının gerçekleşme olasılığı,

P(B|A): A olayının gerçekleştiği durumda B olayının gerçekleşme olasılığı (sonrasal olasılığı) ,

P(A) ve P(B): A ve B olaylarının önsel olasılıklarıdır.

Bir sınıf değişkeni y ve $x_1, \ldots, x_n$ bağımlı bir özellik vektörü verildiğinde, Bayes teoremi aşağıdaki ilişkiyi belirtir:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)} \tag{5.4}$$

Saf (naive) bağımsızlık varsayımını kullanarak aşağıdaki ifade elde edilir:

$$P(x_i|y, x_1, \ldots\ldots, x_{i-1}, x_{i+1}, \ldots\ldots, x_n) = P(x_i|y) \quad (5.5)$$

Her i değeri için ifade basitleştirilirse,

$$P(y|x_1, \ldots\ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots\ldots, x_n)} \quad (5.6)$$

Naive Bayes modeli bir karar verme kuralı ile birleştirilir ve genel olarak kullanılan kural en olasılıklı hipotezin sonuç olarak seçilmesidir. $P(x_1, \ldots\ldots, x_n)$ girdisi sabit olduğu için, aşağıdaki sınıflandırma formülünü kullanabiliriz:

$$P(y|x_1, \ldots\ldots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \quad (5.7)$$

$$\tilde{y} = arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y) \quad (5.8)$$

Spark'ın makine öğrenmesi için sunduğu makine öğrenmesi kütüphanesi (spark.mllib), Multinomial Naive Bayes ve Bernoulli Naive Bayes modellerini destekler. Bu tezde Bernoulli modeli kullanılmıştır. Bu modeller genellikle doküman sınıflandırması için kullanılır. Bu bağlamda, her gözlem bir belgedir ve her özellik teriminin frekansı (Multinomial Naive Bayes'de) frekansı olan bir terimi ya da terimin belgede (Bernoulli Naive Bayes'de) bulunup bulunmadığını gösteren sıfır ya da bir terimi temsil eder. Özellik değerleri negatif olmamalıdır.

Naive Bayes sınıflandırıcısı, modelin özelliklerinin her birinin koşullu bağımsızlığını ifade eden genel bir terim iken, Multinomial Naive Bayes sınıflandırıcısı, özelliklerin her biri için çok terimli bir dağılım kullanan Naive Bayes sınıflandırıcısının belirli bir özelliğidir. Bu varyasyon, bir sınıfta belirli bir kelimenin

koşullu olasılığını, c sınıfına ait belgelerde t teriminin göreceli sıklığı olarak hesaplar. Bernoulli Naive Bayes, çok değişkenli Bernoulli dağılımlarına göre dağıtılan veriler için Naive Bayes eğitim ve sınıflandırma algoritmalarını uygular. Örneklerin ikili değerli özellik vektörleri olarak temsil edilmesini gerektirir. Bu varyasyon, terimin incelenen belgeye ait olması durumunda 1'e eşit, yoksa 0'a eşit bir Boolean değer üretir. Her bir sözcüğün tekrarlanma sayılarını hesaba katmasının yanı sıra belgedeki geçmeyen terimleri de hesaba kattığı için Multinomial Naive Bayes modelinden farklıdır. Multinomial modelde bulunmayan terimler tamamen yok sayılır.

5.2.2 Lojistik Regresyon

Denetimli öğrenme algoritmalarından biri olan Lojistik Regresyon isminin aksine bir regresyon algoritması değil sınıflandırma algoritmasıdır. İkili (binary) veriler, kategorik verilerin en yaygın biçimidir. Sosyal bilimlerde, eğitim araştırmalarında, meteorolojide, askeri ve daha birçok alanda karşılaşılan ikili veriler için kullanılan en uygun model lojistik regresyondur (Agresti 2007). Lojistik regresyon, sayısal değerler yerine (standart regresyon gibi) kategorileri (evet/hayır, doğru/yanlış) tahmin etmek için kullanılır. Diğer değişkenler tarafından etkilenen (bağımlı) değişkenler ile bağımlı değişkenleri etkileyen (bağımsız) değişkenler arasındaki ilişkiyi tanımlamak için en uygun modeli bulur. Lojistik regresyonda, doğrusal regresyondan farklı olarak bağımlı değişkenin değeri yerine bağımlı değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilir. Bunun yanı sıra doğrusal regresyon analizinde tahmin edilecek bağımlı değişkenler süreklidir. Lojistik regresyonda ise bağımlı değişken kesikli bir değer almaktadır.

Lojistik regresyon, ikili değerler olarak adlandırdığımız 1 veya 0'ın (başarı veya başarısızlık) olasılık oranlarının (odds ratio) doğal logaritmasıdır (Hosmer ve diğ. 2013) Başarı durum olasılığının (p) lojistik dönüşümü aşağıdaki gibidir:

$$Logit(P_i) = \log(\frac{P_i}{1 - P_i}) \tag{5.9}$$

Doğrusal model çerçevesinde, (5.9) nolu denklem bir bağıntı fonksiyonu olarak ele alındığında logit model elde edilir. Aşağıdaki logit modelde, x'ler bağımsız değişkenleri göstermektedir:

$$Log\left(\frac{P_i}{1-P_i}\right) = Z_i = \sum_{k=0}^{p} \beta_k X_{ik} \tag{5.10}$$

(5.10) nolu denklemde k ile bağımsız değişken sayısı, X ile bağımsız değişkenleri β ile bağımsız değişkenlerin regresyon katsayıları ifade edilmektedir. Bu denklemle ifade ettiğimiz p olasılığının lojistik dönüşümünde; p 0'a yaklaştıkça, logit (P_i) -∞' a, p 1'e yaklaştıkça da logit (P_i) +∞' a yaklaşmaktadır. P arttıkça logit(p) de artar. Bağımlı değişkenin 1 değerini alma olasılığı (P_i) ile gösterilirken, 0 değerini alma olasılığı $(1 - P_i)$ ile gösterilmektedir. Bu oran ise bahis oranı (odds ratio) olarak adlandırılmaktadır ve aşağıdaki (5.11) nolu eşitlikle ifade edilir:

$$\frac{P_i}{1-P_i} = \frac{1+e^{Z_i}}{1+e^{-Z_i}} \tag{5.11}$$

Bahis oranı (odds ratio), Exp(β) olarak ifade edilir. Bahis oranının sonucuna göre bağımsız değişkenlerin Y ile ifade edilen çıktı değerine olan katkısı elde edilebilir. Yani Exp(β), Y değişkeninin bağımsız değişkenin/değişkenlerin etkisi ile kaç kat daha fazla gözlenme olasılığına sahip olduğunu belirtir.

Bu tezde Apache Spark'ın makine öğrenmesi algoritmaları için sunduğu MLlib kütüphanesinde bulunan Logistic Regression algoritması kullanılarak duygu analizi gerçekleştirilmiştir.

5.2.3 Karar Ağaçları

Karar ağaçları, sınıflandırma ve regresyonun makine öğrenme işlemleri için kullanılan popüler yöntemlerdir. Kolay yorumlanması, sürekli ve ayrık özellikleri ele alması, özellik ölçeklendirmesi gerektirmemesi gibi birçok avantajları bakımından yaygın olarak kullanılmaktadır. Diğer sınıflandırma metodolojilerinde de olduğu gibi bu yöntemde de öğrenme ve sınıflandırma olmak üzere iki basamaklı bir işlem gerçekleştirilir. Öğrenme aşamasında, önceden etiketlenmiş eğitim verisi model oluşturmak amacıyla sınıflandırma algoritması tarafından analiz edilir. Öğrenilen bilgi bir ağaç üzerinde modellenerek bu model karar ağacı olarak gösterilir. Ağaç yapısında bulunan kök, ara düğümler ve yapraklar bir çeşit karar verme kuralları ile oluşturulur. Bağlaç görevini üstlenen bu kurallar, belirli koşullarda elde edilecek sonucu ifade eder. Sınıflama aşamasında ise test verisi kullanılarak sınıflama kurallarının veya karar ağacının doğruluğu belirlenir.

Öğrenme kümesi oluşturulduktan sonra, kümedeki örnekleri en iyi ayıran nitelik belirlenir. Seçilen nitelik ile ağacın bir düğümü oluşturulur ve bu düğümden çocuk düğümleri veya ağacın yaprakları oluşturulur. Bu işlem özyinelemeli olarak tekrarlanır ve tekrarlama işleminin tahmin üzerinde bir etkisi kalmayana kadar sürer. Yapraklarda sonuç değerleri bulunmaktadır. Aşağıda karar ağaçlarına bir örnek verilmiştir.

Tablo 5. 4: Karar tablosu örneği

Hafta(Örnek)	Hava	Aile	Para	Sonuç
1	Güneşli	Evet	Var	Sinema
2	Güneşli	Hayır	Var	Tenis
3	Rüzgârlı	Evet	Var	Sinema
4	Yağmurlu	Evet	Yok	Sinema
5	Yağmurlu	Hayır	Var	Ev
6	Yağmurlu	Evet	Yok	Sinema
7	Rüzgârlı	Hayır	Yok	Sinema
8	Rüzgârlı	Hayır	Var	Alışveriş
9	Rüzgârlı	Evet	Var	Sinema
10	Güneşli	Hayır	Var	Tenis

Tablo 5.4'de verilen değerlerden en ayırt edici nitelik belirlenir ve ağacın kökü olarak alınır. Son olarak da ağacın çocuk düğümü olan A düğümüne ait alt veri kümesi belirlenir. Alt veri kümesi belirlenirken, örneklerin hepsi aynı sınıfa aitse, örnekleri ayıracak başka bir nitelik kalmamışsa ve kalan niteliklerin değerini taşıyan örnek yoksa işlem sonlandırılır. Bu özelliklerin geçersiz olduğu durumlarda da tekrar örnekleri en iyi ayıran nitelik belirlenir. Yani her bölüm, bir ağaç düğümünde bilgi kazanımını en üst düzeye çıkarmak için olası bölünmelerden en iyi bölünmeyi seçerek gerçekleştirilir.

5.2.3.1 Bilgi Kazancı

Karar ağaçları oluşturulurken en ayırt edici özellik bulunur. Bu da her özellik için bilgi kazancının ölçülmesi ile elde edilir. Bilgi kazancı ölçümünde Entropi (Entropy) kullanılır (Safavian ve diğ. 1991). Kelime anlamı olarak düzensizlik (impurity) anlamına gelen Entropi, sınıf etiketlerine dayanarak hesaplanır. Belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir. Entropi ile bilgi kazancı birbirlerinin tersidir. Yani Entropi değeri 0 ise bilgi kazanımı 1 çıkacaktır. Örneğin, on farklı sınıf için özelliğimiz on farklı değer alıyorsa bu durumda Entropi 0'dır. Aşağıda Entropi hesaplamanın formülü verilmiştir.

$$H(X) = E\big(I(X)\big) = \sum_{i=1}^{n} \mathrm{p}(x_i) \log_2 \left(\frac{1}{p(x_i)}\right) \qquad (5.12)$$

$$= -\sum_{i=1}^{n} \mathrm{p}(x_i) \log_2 \left(\frac{1}{p(x_i)}\right)$$

Yukarıda verilen karar tablosuna (T kümesi) göre entropi değeri:

$$H(T) = -\left(\frac{6}{10}\right)\log_2\left(\frac{6}{10}\right) - \left(\frac{2}{10}\right)\log_2\left(\frac{2}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right)$$

$$H(T) = 1{,}571$$

A niteliğinin T veri kümesindeki bilgi kazancı:

Gain(T,A)=Entropy(T) - $\sum$ P(v)$Entropy(T(v))$

V: Values of A

P(v)=|T(v)| / |T|

Her bir özellik için bilgi kazanımı hesaplanır;

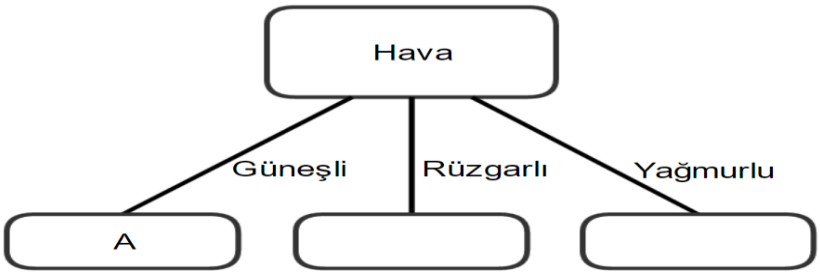
$$Gain(T, hava) = Entropy(T) - ((P(güneşli)Entropy(T_{güneşli}) + P(rüzgarlı)Entropy(T_{rüzgarlı}) + P(yağmurlu)Entropy(T_{yağmurlu})$$

= 0,70

$Gain(T, aile) = 0{,}61$

$Gain(T, para) = 0{,}2816$

Yukarıdaki hesaplanan bilgi kazançlarına göre hava özelliği en büyük bilgi kazancına sahip olduğu için en ayırt edici özellik olarak seçilmiştir.



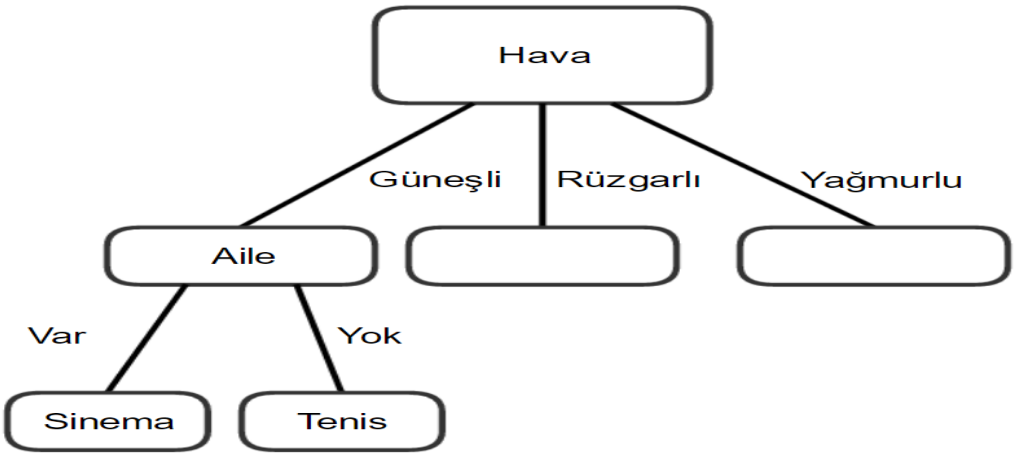
Şekil 5.11: Karar ağacı

Ağacın çocuk düğümü olan A düğümüne ait alt veri kümesi belirlenir ve her alt kümse için tekrar bilgi kazancı hesaplanarak en ayırt edici özellik belirlenir.

Aşağıda verilen tabloda havanın güneşli olduğu durumlar ele alınmıştır. Bu tablodan elde edilen bilgi kazançlarına göre en ayırt edici özellik aile olarak bulunmuştur. Örneklerin hepsi aynı sınıfa ait, örnekleri bölecek ayırt edici özellik kalmamış veya kalan özelliklerin değerini taşıyan örnek olmayana dek ayırt edici özellik bulma işlemi her bir düğüm için devam eder.

Tablo 5. 5: Ayırt edici özellik bulunduktan sonra karar tablosu

Hafta(Örnek)	Hava	Aile	Para	Sonuç
1	Güneşli	Evet	Var	Sinema
2	Güneşli	Hayır	Var	Tenis
10	Güneşli	Hayır	Var	Tenis



Şekil 5.12: Karar ağacı (İkinci adım)

Yukarıda örnek olarak verilen karar ağacı uygulamasına göre yinelemeli olarak bulunan ayırt edici özellikler son bulununcaya kadar işleme devam edilmiştir ve sonuç olarak ağacın kökünde bulunan hava parametresi en ayırt edici özellik olarak bulunmuştur. Tablo 5.5'de verilen havanın güneşli olma durumuna göre bilgi kazanımlarına bakıldığında ise aile parametresi en ayırt edici özellik olarak bulunmuştur. Havanın her bir farklı parametresine göre bu işlemler tekrarlanır ancak yukarıdaki şekilde yalnızca havanın güneşli olma durumu ele alınmıştır.

6. UYGULAMA SONUÇLARI

Duygu analizi alanında birçok çalışma yapılmıştır. Bu çalışmalar, veriler üzerinde sözlüğe dayalı ya da makine öğrenmesine dayalı olmak üzere farklı şekillerde gerçekleştirilmiştir. Kullanılan verinin boyutu, dili ve yapısı yapılan çalışmanın performansını etkileyen önemli faktörlerdendir. Bölüm 4'te makine öğrenmesine dayalı duygu analizi alanında yapılan çalışmalar ve uygulamada kullanılan veriler, algoritmalar ve algoritmaların başarım oranları verilmiştir. Bu çalışmada toplamda 57.650 adet şarkı sözü verisi üzerinde duygu analizi işlemi gerçekleştirilmiştir. Makine öğrenmesi algoritmalarından kullanım ve yorumlana bilirlik açısından kolay olduğu düşünülen üç farklı algoritma kullanılmıştır. Aşağıda, tezde kullanılan başarım metrikleri açıklanmıştır.

6.1 Sınıflandırma Algoritmalarının Karşılaştırılması

6.1.1 Model Başarım Ölçütleri

Model başarımının ölçülmesinde kullanılan birçok yöntem mevcuttur. Eğitim kümesinde girişler ve çıkışlar arasında bir fonksiyon oluşturan sınıflandırma modeli, yeni girişler için anlamlı sınıflandırma etiketleri belirler. Farklı sınıflandırma yaklaşımları kullanılabilir. Bir sınıflandırıcının ne derece doğru sınıflandırma yaptığı birçok parametreye bağlıdır. Her sınıflandırıcı aynı eğitim kümesinde aynı sonuçları üretmeyebilir ve ya aynı sınıflandırıcı modeli bir eğitim kümesi için farklı parametrelerle farklı çözümler üretebilir. Sınıflandırma algoritmasının kullanım amaçlarına ve problemin büyüklüğüne bağlı olarak farklı sonuçlar kullanılarak sınıflandırma algoritmaları değerlendirilebilir.

Bu tezde uygulanan her bir sınıflandırma algoritması için k-katlamalı çapraz doğrulama tekniği kullanılmıştır ve k değeri 10 olarak alınmıştır.

6.1.1.1 K-Katlamalı Çapraz Doğrulama

Bir sınıflandırma probleminde kullanılacak verinin ne kadarı eğitim verisi ne kadarı test verisi olarak kullanılacağı verinin boyutuna göre farklılık gösterebilir. Literatürde veriyi eğitim ve test verisi olarak ayırmada farklı teknikler bulunmaktadır. Daha iyi öğrenme ve genelleme sağlanabilmesi için eğitim verisinin büyük olması, sınıflandırıcının da hata olasılığını daha iyi tahmin edebilmek için test verisinin büyük olması gerekmektedir. Eğitim verisi ile aynı değerlere sahip test kümesi kullanılmamalıdır. Eğitim ve test verisi ayrı olarak oluşturulmalıdır ve değerlerinin birbirlerinden farklı olması gerekmektedir.

Bu yöntemde veri kümesi, önceden belirlenen k adet eşit boyutta alt kümeye bölünmektedir. Literatürde k için en çok kullanılan değer 10'dur. Bu tezde de veri bu yöntemle eğitim ve test verisi olarak ayrılmıştır. k değeri 10 olarak alınmıştır. Veri k adet eşit parçaya ayrıldıktan sonra rastgele seçilen bir parça test kümesi geri kalan parçalar eğitim kümesi olarak kullanılarak sınıflandırma algoritmasından bir sonuç elde edilir. Ardından aynı işlem ikinci parça seçilerek tekrarlanır ve yine sınıflandırma algoritması çalıştırılarak bir sonuç daha elde edilir. Bu şekilde k kere aynı yöntem, k farklı eğitim ve test kümeleri için çalıştırılmış olur. Sistemin genel başarısı bu k farklı sonucun ortalamasından elde edilir. Bu işlemi aşağıdaki gibi formülleştirebiliriz:

$$t_i \in D \; olmak \; \text{üzere}, \qquad Başarım = \frac{\sum_{i=0}^{k} F(t_i, D - t_i)}{k} \tag{6.1}$$

Buradaki F(test, eğitim), sınıflandırma için kullanılan fonksiyonu, D veri kümesi, k kaç parça katlama olduğunu ve t ise veri kümesi üzerinden seçilen her bir test kümesini temsil etmektedir. Yukarıda verilen 6.1'de de olduğu üzere her bir k değeri için sınıflandırıcının ürettiği başarı değerine göre elde edilen tüm performansların toplamının k sayısına bölünerek ortalaması alınarak sistemin genel başarım değeri elde edilmiştir.

6.1.1.2 Doğruluk Oranı(Accuracy)

Doğru sınıflandırılmış örnek sayısının toplam örnek sayısına oranıyla elde edilen bu oran, model başarımının ölçülmesinde kullanılan en popüler ve basit yöntemdir. Yanlış değerlendirilmiş örnek sayısının, toplam örnek sayısına oranı da hata oranıdır. Yani doğruluk oranının 1'e tamlayanıdır.

Tablo 6. 1: Karışıklık Matrisi (Class Confusion Matrix)

		Öngörülen Sınıf	
		S1(Pozitif)	S2(Negatif)
Doğru Sınıf	S1(Pozitif)	True Positive **TP**	False Negative **FN**
	S2(Negatif)	False Positive **FP**	True Negative **TN**

$$Doğruluk = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.2}$$

$$Hata\ Oranı = \frac{FP + FN}{TP + FP + FN + TN} \tag{6.3}$$

6.1.1.3 Kesinlik (Precision)

Kesinlik, doğru pozitif öngörülerin toplam pozitif öngörü sayısına bölünmesiyle hesaplanır.

$$Kesinlik = \frac{TP}{TP + FP} \tag{6.4}$$

6.1.1.4 Duyarlılık (Recall)

Doğru pozitif tahminlerin (TP) sayısının, toplam pozitif örnek sayısına (TP+FN) bölünmesiyle elde edilir.

$$Duyarlılık = \frac{TP}{TP + FN} \tag{6.5}$$

6.1.1.5 F-Ölçütü (F-Measure)

Kesinlik ve duyarlılık ölçütlerinin bir arada kullanılarak hesaplandığı bir yöntemdir. Tek başına anlamlı bir karşılaştırma sonucu çıkarmada yetersiz kalan bu iki değerin harmonik ortalaması hesaplanarak F-ölçütü elde edilir.

$$F - \text{Ölçütü} = \frac{2\ X\ Duyarlılık\ X\ Kesinlik}{Duyarlılık + Kesinlik} \tag{6.6}$$

6.1.2 Sınıflandırma Algoritmalarının Sonuçları

Bu tezde kullanılan üç farklı sınıflandırma algoritmasına göre en yüksek başarım oranına sahip algoritma 0.85 başarım oranıyla Naive Bayes algoritması olmuştur. Logistic Regresyon ve Decision Tree algoritmalarının başarım oranları sırasıyla 0.82 ve 0.84'tür. Tablo 6.2'de tezde kullanılan üç farklı algoritmanın sonuçları verilmiştir. Her algoritma için 10 kez çalıştırılıp sonuçların ortalamaları elde edilmiştir. Tablo 6.2'de bu ortalamalar verilmiştir. Sonuçlar elde edilirken k katlamalı çapraz doğrulama yöntemi kullanılmıştır. Bu yönteme göre her bir çalıştırma sonucunda elde edilen değerlerin ortalaması ve standart sapması da hesaplanmıştır. Algoritmaların başarım oranları doğruluk(accuracy), kesinlik(precision), duyarlık(recall) ve f-ölçütü(f-measure) model başarım ölçütlerine göre karşılaştırılmıştır ve değerleri tabloda verilmiştir.

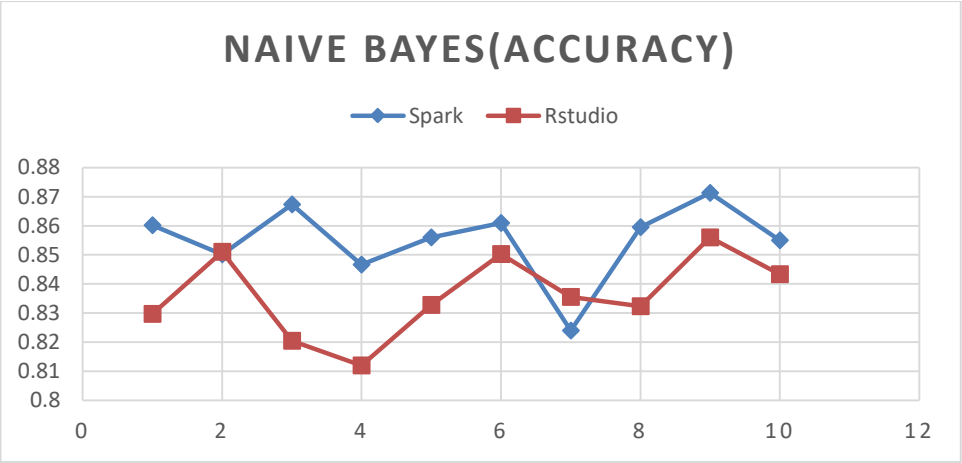
Tablo 6. 2: Spark üzerinde algoritma sonuçları (A(Accuracy), P(Precision), R(Recall), F(F-Measure))

	n	Veri miktarı	Accuracy	Precision	Recall	F-Measure
Naive Bayes	n=1	5765	0.860191	0.8504	0.86726	0.85875
	n=2	5765	0.85013	0.875	0.8152	0.84404
	n=3	5765	0.867303	0.8981	0.84691	0.87175
	n=4	5765	0.846661	0.83367	0.86607	0.84956
	n=5	5765	0.856028	0.86128	0.85119	0.8562
	n=6	5765	0.860885	0.87925	0.85065	0.86471
	n=7	5765	0.823938	0.85714	0.82367	0.84007
	n=8	5765	0.859497	0.88433	0.86996	0.87709
	n=9	5765	0.871292	0.85163	0.89843	0.87441
	n=10	5765	0.854987	0.8593	0.89968	0.87903
	Ortalama	5765	0.855091	0.86501	0.8589	0.86156
	Standart Sapma	0	0.013156	0.01901	0.02767	0.01404
	n	Veri miktarı	Accuracy	Precision	Recall	F-Measure
Logistic Regression	n=1	5765	0.826886	0.90538	0.87911	0.89205
	n=2	5765	0.839029	0.8557	0.82792	0.84158
	n=3	5765	0.8366	0.84458	0.78607	0.81427
	n=4	5765	0.856895	0.86586	0.84517	0.85539
	n=5	5765	0.805724	0.78998	0.83877	0.81364
	n=6	5765	0.822203	0.83661	0.80468	0.82033
	n=7	5765	0.804337	0.81185	0.81749	0.81466
	n=8	5765	0.811795	0.8642	0.8501	0.85709
	n=9	5765	0.797745	0.80249	0.78738	0.79486
	n=10	5765	0.817173	0.85776	0.89352	0.87527
	Ortalama	5765	0.821839	0.84344	0.83302	0.83792
	Standart Sapma	0	0.01837	0.03449	0.03592	0.03138
	n	Veri miktarı	Accuracy	Precision	Recall	F-Measure
Decision Tree	n=1	5765	0.842151	0.86154	0.8589	0.86022
	n=2	5765	0.848049	0.88951	0.83225	0.85993
	n=3	5765	0.853599	0.88189	0.83772	0.85924
	n=4	5765	0.840416	0.8533	0.84608	0.84967
	n=5	5765	0.846834	0.87567	0.82013	0.84699
	n=6	5765	0.865568	0.86619	0.85916	0.86266
	n=7	5765	0.845967	0.8644	0.83812	0.85106
	n=8	5765	0.820642	0.83157	0.84367	0.83757
	n=9	5765	0.813183	0.81201	0.8083	0.81015
	n=10	5765	0.829662	0.84694	0.82396	0.83529
	Ortalama	5765	0.840607	0.8583	0.83683	0.84728
	Standart Sapma	0	0.015591	0.02346	0.01633	0.01614

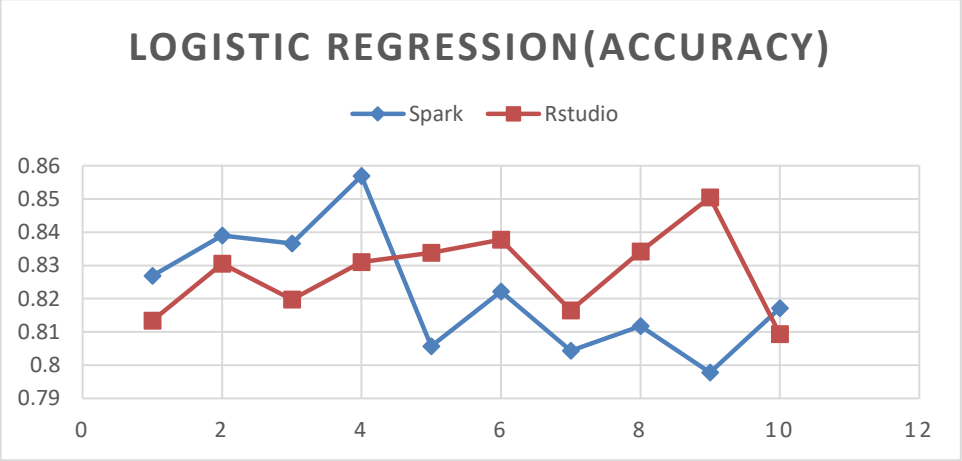
Tablo 6.2'de verilen üç farklı algoritmanın doğruluk sonuçlarına göre en yüksek başarım oranına 0.85 oranla Naive Bayes algoritması sahiptir. Decision Tree ve Logistic Regresyon algoritmaları da sırasıyla ikinci ve üçüncü sırada yer almaktadır.

Tablo 6. 3: RStudio'daki algoritma sonuçları (A(Accuracy), P (Precision), R(Recall), F(F-Measure))

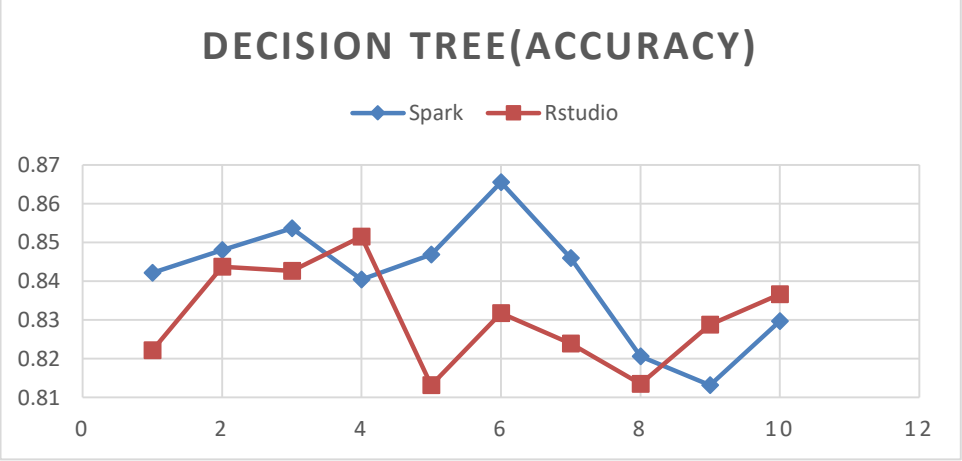
	k	Veri miktarı	Accuracy	Precision	Recall	F-Measure
Naive Bayes	k=1	5765	0.829662	0.85033	0.84756	0.84894
	k=2	5765	0.850997	0.81858	0.82215	0.82036
	k=3	5765	0.820468	0.87523	0.77435	0.82171
	k=4	5765	0.811969	0.76325	0.85757	0.80767
	k=5	5765	0.832784	0.8466	0.82555	0.83594
	k=6	5765	0.850304	0.87169	0.83796	0.85449
	k=7	5765	0.835559	0.85885	0.80374	0.83038
	k=8	5765	0.832264	0.85529	0.85266	0.85397
	k=9	5765	0.856028	0.87509	0.83493	0.85454
	k=10	5765	0.843365	0.86584	0.89481	0.88009
	Ortalama	5765	0.83634	0.84807	0.83513	0.84081
	Standart Sapma	0	0.014012	0.03429	0.03245	0.02153
	k	Veri miktarı	Accuracy	Precision	Recall	F-Measure
Logistic Regression	k=1	5765	0.813356	0.89067	0.86462	0.87745
	k=2	5765	0.830529	0.88004	0.82433	0.85128
	k=3	5765	0.819775	0.85886	0.74747	0.7993
	k=4	5765	0.831049	0.8876	0.80111	0.84214
	k=5	5765	0.833825	0.82935	0.83286	0.8311
	k=6	5765	0.837814	0.87878	0.81194	0.84404
	k=7	5765	0.816479	0.81967	0.79261	0.80591
	k=8	5765	0.834172	0.86345	0.85286	0.85812
	k=9	5765	0.850477	0.84553	0.85032	0.84792
	k=10	5765	0.809367	0.85098	0.88898	0.86957
	Ortalama	5765	0.827684	0.86049	0.82671	0.84268
	Standart Sapma	0	0.012694	0.02432	0.04071	0.02504
	k	Veri miktarı	Accuracy	Precision	Recall	F-Measure
Decision Tree	k=1	5765	0.822203	0.87028	0.80793	0.83794
	k=2	5765	0.843712	0.8727	0.82951	0.85056
	k=3	5765	0.842671	0.8707	0.81933	0.84424
	k=4	5765	0.851518	0.85616	0.8515	0.85383
	k=5	5765	0.813183	0.86054	0.74004	0.79575
	k=6	5765	0.831743	0.85965	0.81126	0.83475
	k=7	5765	0.823938	0.85396	0.8131	0.83303
	k=8	5765	0.81353	0.83396	0.80736	0.82044
	k=9	5765	0.828794	0.83897	0.82391	0.83137
	k=10	5765	0.8366	0.84005	0.84289	0.84147
	Ortalama	5765	0.830789	0.8557	0.81468	0.83434
	Standart Sapma	0	0.012932	0.01401	0.03012	0.01666



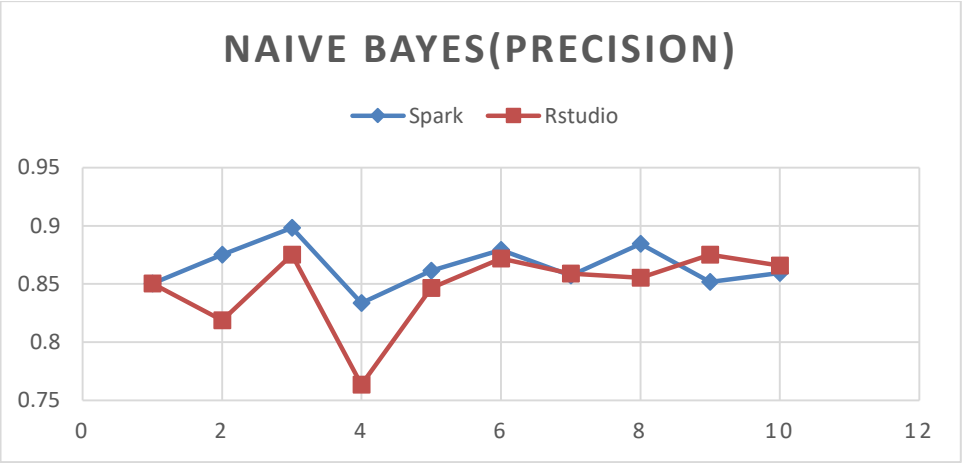
Şekil 6.1: Naive Bayes algoritması için doğruluk değerleri



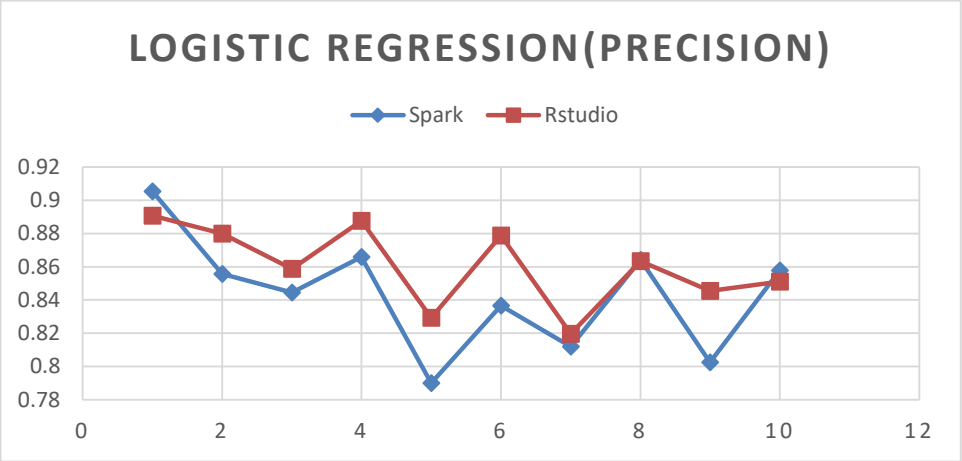
Şekil 6.2: Logistic Regression algoritması için doğruluk değerleri



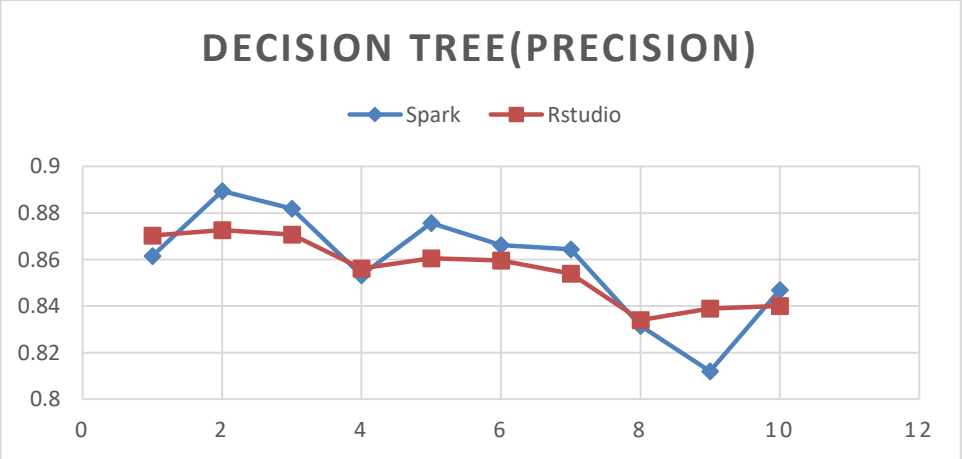
Şekil 6.3: Decision Tree algoritması için doğruluk değerleri



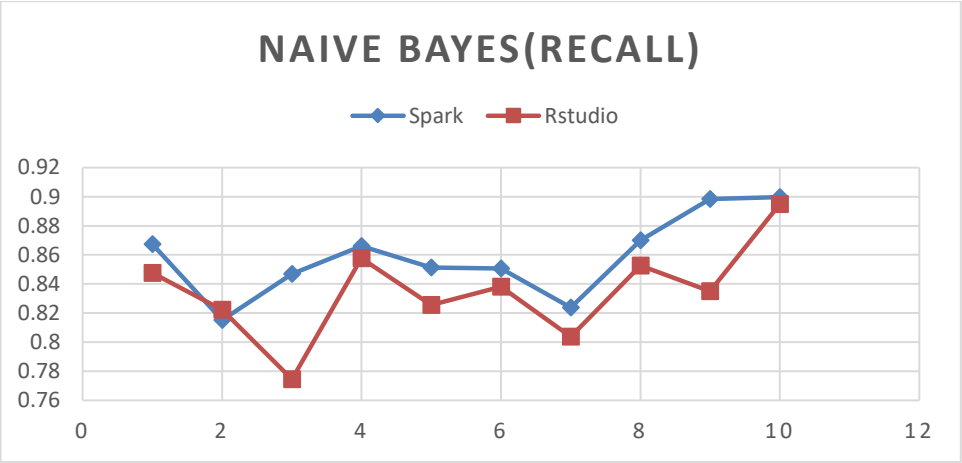
Şekil 6.4: Naive Bayes algoritması için kesinlik(precision) değerleri



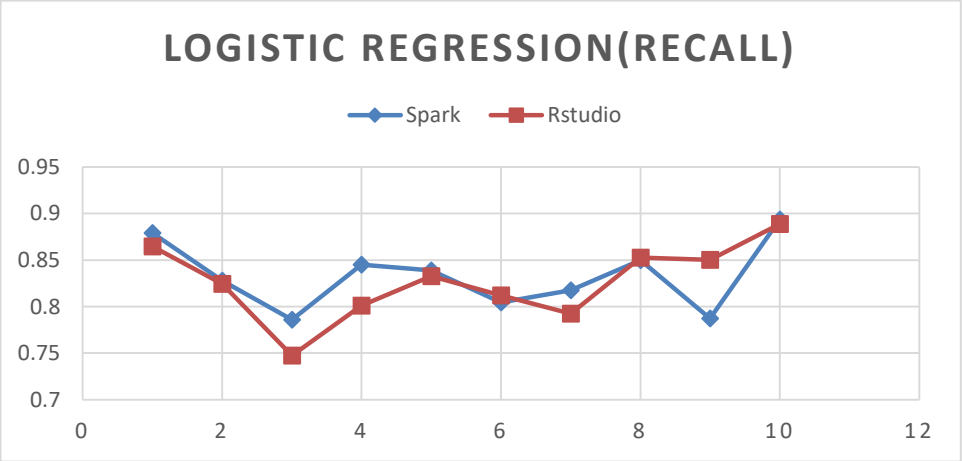
Şekil 6.5: Logistic Regression algoritması için kesinlik(precision) değerleri



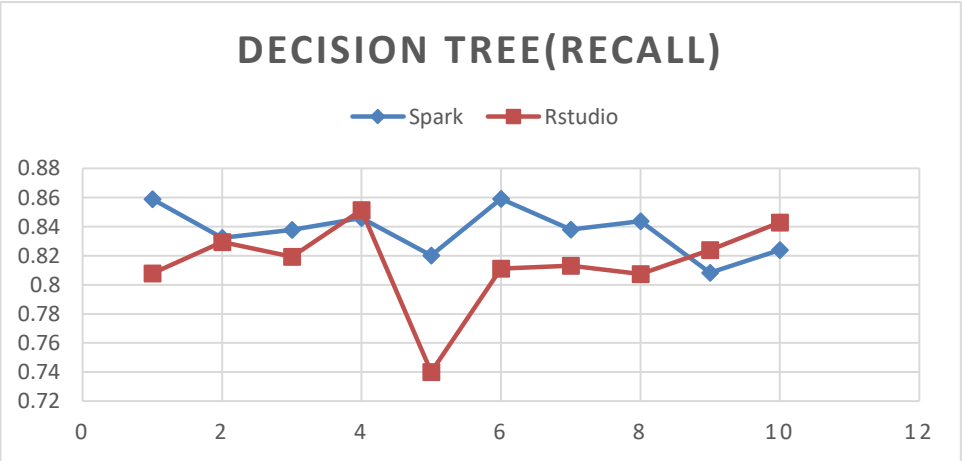
Şekil 6.6: Decision Tree algoritması için kesinlik(precision) değerleri



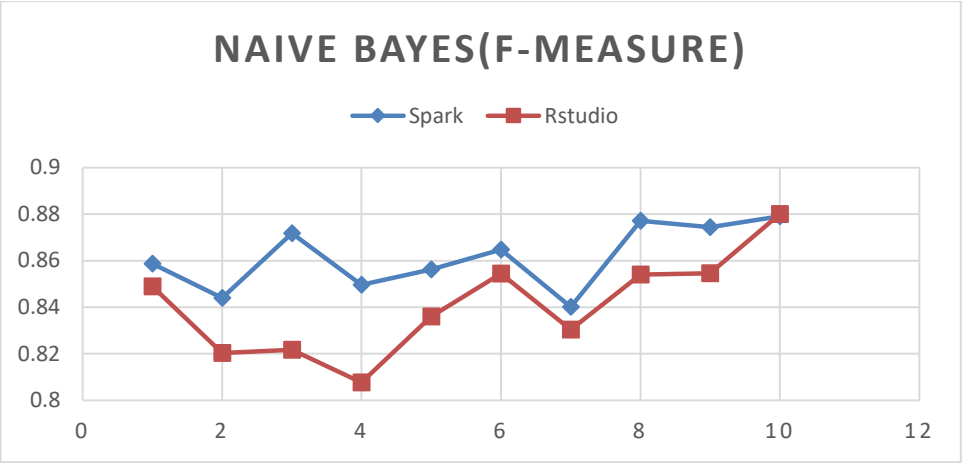
Şekil 6.7: Naive Bayes algoritması için duyarlık(recall) değerleri



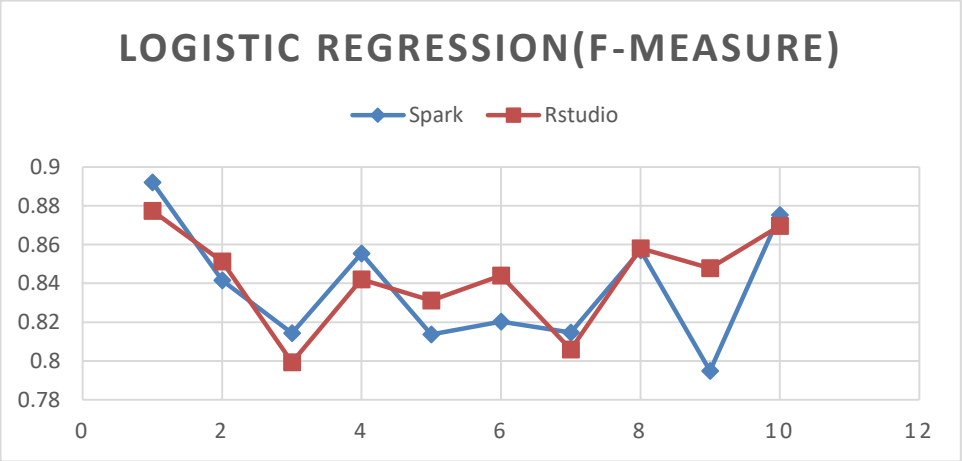
Şekil 6.8: Naive Bayes algoritması için duyarlık(recall) değerleri



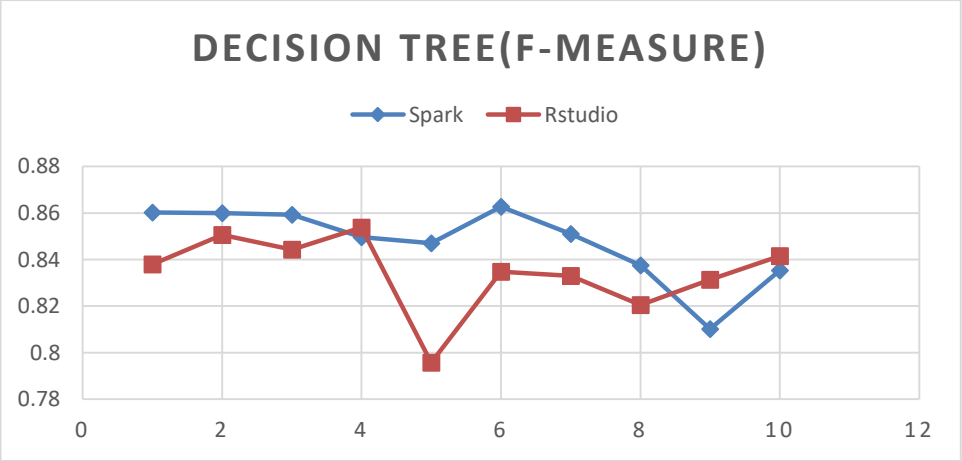
Şekil 6.9: Decision Tree algoritması için duyarlık(recall) değerleri



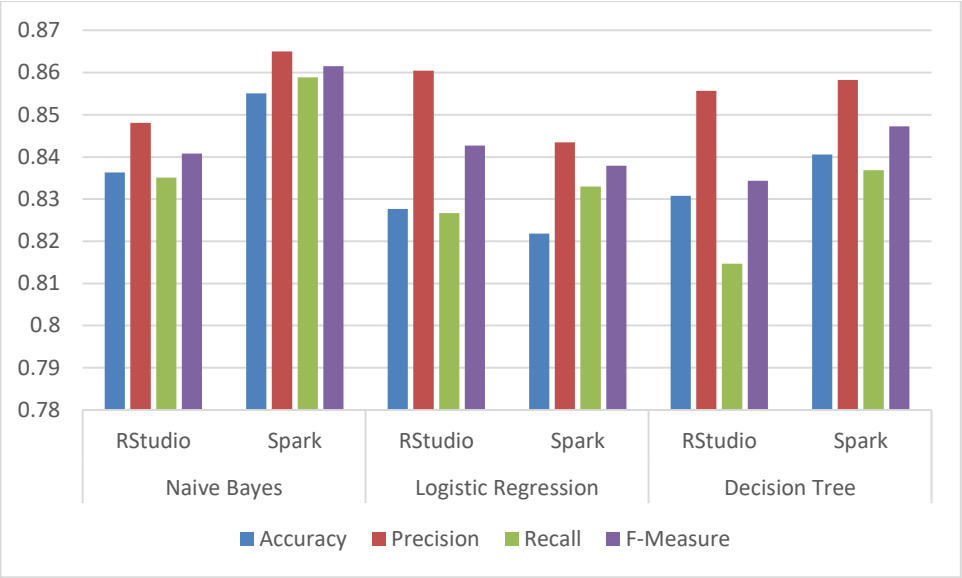
Şekil 6.10: Naive Bayes algoritması için f-ölçütü(f-measure) değerleri



Şekil 6.11: Logistic Regression algoritması için f-ölçütü(f-measure) değerleri



Şekil 6.12: Decision Tree algoritması için f-ölçütü(f-measure) değerleri



Şekil 6.13: Algoritmaların çalıştırılmaları sonucu elde edilen değerlerin ortalamaları

Şekil 6.13'te her bir algoritmanın Spark ve Rstudio üzerinde onar kez çalıştırılmaları sonucunda elde edilen değerlerin ortalamaları grafik halinde gösterilmiştir.

Sınıflandırma işlemi için uygulanan üç farklı algoritma doğruluk oranlarına (accuracy) göre karşılaştırılmıştır. Algoritmaların başarım oranları, verinin büyüklüğüne, düzenli olup olmadığına, test ve eğitim kümesinde bulunan veri miktarlarına, seçilen kelime ağırlıklandırma yöntemlerine ve bunlar gibi birçok faktöre bağlı olarak farklılık gösterebilir. Ancak bu tezde uygulanan yöntemler, literatürde de sıklıkla kullanılan ve tezde kullanılan veriye en uygun olanı seçilerek belirlenmiştir.

6.2 Test Ortamlarının Karşılaştırılması

Bu tezde Spark ve RStudio olmak üzere iki farklı test ortamı kullanılmıştır. Uygulanan algoritmaların eğitim ve test için geçen süreleri iki farklı platformda da karşılaştırılmıştır. Süreler karşılaştırılırken verinin makine öğrenmesi algoritmasına uygun hale getirilmesi için geçen süreler dikkate alınmamıştır. Algoritmaya hazır hale getirilen verinin eğitilmesi ve test edilmesi için geçen toplam süreler karşılatırılmıştır. Elde edilen sonuçlara göre Spark üzerinde çok daha kısa sürede algoritma gerçekleştirilmiştir ve sonuçlar elde edilmiştir.

Tezde kullanılan test ortamlarının genel özellikleri tablo halinde verilmiştir. Tablo 6.4'te Spark ortamının genel özellikleri verilmiştir.

Tablo 6. 4: Spark ortamının genel özellikleri

	SPARK	
	Name	Value
Runtime Information	Java Home	/usr/lib/jvm/java-8-openjdk-amd64/jre
	Java Version	1.8.0_111 (Oracle Corporation)
	Scala Version	version 2.10.6
Driver Type	Community Optimized	6.0 GB Memory, 0.88 Cores
Spark Cluster	Spark Version	2.1.x-scala2.10

Tablo 6.5'te RStudio ve üzerinde çalıştırıldığı makinenin genel özellikleri gösterilmiştir.

Tablo 6. 5: RStudio ortamının genel özellikleri

	Rstudio	
	Name	Value
Runtime Information	R Version	version 3.4.0
	Platform	x86-w64-mingw32
	Running under	Windows >= 8x64
System Information	System Type	64-bit Operating System
	Processor	Intel(R) Xeon(R) CPU E5-2620 2.00Ghz
	Installed Memory (RAM)	32.00 GB

Tezde kullanılan algoritmalar, Tablo 6.4 ve Tablo 6.5'te özellikleri verilen Spark ve RStudio üzerinde ayrı ayrı çalıştırılmıştır. Spark ve RStudio'da 10'ar kez çalıştırılarak eğitim ve test için geçen sürelerin ortalamaları alınmıştır. Tablo 6.6'te ortalaması alınan bu süreler gösterilmiştir.

Tablo 6. 6: Spark ve RStudio'da çalıştırılan algoritmalar için geçen süre

	Rstudio	Spark
Naive Bayes	953.68 sn	56.75 sn
Logistic Regression	1083.51 sn	82.04 sn
Decision Tree	1187.49 sn	75.68 sn

Tablo 6.6'te gösterildiği üzere Naive Bayes algoritması her iki platform için de diğer algoritmalara göre daha hızlı sonuç üretmiştir. Algoritmaların genel işleyiş

yapısının yanı sıra verinin algoritmaya uygunluğu ve verinin boyutu da algoritmanın hızını etkilemektedir. Veri miktarı arttıkça da verinin eğitimi ve test için geçen süre de artmaktadır. Tablo 6.6'te verilen değerler toplamda 57650 adet verinin eğitim ve testi için geçen sürelerdir. Kullanılan yöntemler ve algoritmalara bağlı olarak da elde edilen sonuçlar değişmektedir.

7. SONUÇ VE ÖNERİLER

Bu tezde, makine öğrenmesine dayalı duygu analizi işlemi büyük veri araçlarından Spark kullanılarak gerçekleştirilmiştir. Toplamda 57650 adet şarkı sözü İngilizce şarkı sözü kullanılmış ve her bir şarkı sözü polarlama işlemlerinden geçirilerek olumlu ya da olumsuz olarak etiketlenmiştir. Denetimli öğrenme algoritmalarından Naive Bayes, Logistic Regression ve Decision Tree algoritmaları kullanılarak başarım oranları elde edilmiştir. Şarkı sözü verisi üzerinde gerçekleştirilmiş ve uygulanan denetimli öğrenme algoritmalarının başarım oranları karşılaştırılarak en iyi başarım oranına sahip algoritma bulunmuştur.

Duygu analizi alanında yapılan çalışmalar incelenmiştir ve genellikle çalışmaların daha küçük boyutlu veriler üzerinde yapıldığı görülmüştür. Herhangi bir büyük veri aracı kullanılarak yapılan duygu analizi gerçekleştirimine çok az rastlanmıştır. Bu tezde makine öğrenmesine dayalı duygu analizi işleminin büyük veri aracı üzerinde gerçekleştirilmesi amaçlanmıştır. Kullanılan makine öğrenmesi algoritmaların başarım oranları karşılaştırılarak en iyi başarım oranına sahip algoritma bulunmuştur. Kullanılan makine öğrenmesi algoritmalarından Naive Bayes algoritmasının Lojistik Regresyon ve Karar Ağaçları algoritmalarının başarım oranlarına göre daha yüksek bir başarım oranına sahip olduğu görülmüştür.

Verinin boyutuna göre algoritmanın gerçekleştirilmesi için geçen süre de farklılık göstermektedir. Düşük boyutlu verilerde makine öğrenmesi algoritmalarının uygulanması herhangi bir platform ile kısa sürede gerçekleştirilebilirken, verinin boyutu arttıkça bu süre de verinin miktarına bağlı olarak artmaktadır. Bu nedenle verinin işlenmesi ve analiz edilmesi açısından büyük kolaylık sağlayan büyük veri araçlarına ihtiyaç duyulmaktadır. Büyük veri araçları, işlemleri dağıtık bir biçimde gerçekleştirdiğinden süre açısından büyük kazanç sağlanmaktadır. Bu uygulamada da veri boyutu büyük olduğundan algoritmaların gerçekleştirimi için büyük veri aracı kullanılmıştır. R, Python ve Java gibi farklı dilleri desteklediğinden, büyük veri aracı olarak Databricks firmasının SAS (Statistical Analysis System) olarak sunduğu Spark platformu kullanılmıştır. Zaman bakımından bu platformun sağladığı performans RStudio ile karşılaştırılmıştır. Naive Bayes algoritması hem Spark üzerinde hem de RStudio üzerinde çalıştırılarak algoritmanın gerçekleştirilmesi için geçen süreler

karşılaştırılmıştır ve Spark'ın RStudio'ya göre çok daha iyi performans sağladığı gözlenmiştir.

İleriki çalışmalarda, tez kapsamında büyük veri aracı olarak kullanılan Spark platformunun yanı sıra farklı büyük veri araçları da kullanılarak performans karşılaştırılması yapılabilir. Aynı zamanda bu algoritmaların farklı veriler üzerinde de başarım oranları elde edilerek, verinin yapısının algoritma başarım oranına ne derecede etkili olduğu da tespit edilebilir.

8. KAYNAKLAR

Amolik, A., Jivane, N., Bhandari, M., & Venkatesan, M., "Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques", *International Journal of Engineering and Technology*, *7*(6), 2038-2044, (2015).

Aue, A., & Gamon, M., "Customizing sentiment classifiers to new domains: A case study.", In *Proceedings of recent advances in natural language processing (RANLP),* (Vol. 1, No. 1-3, pp. 2-1), (2005).

Bhargavi, P., & Jyothi, S., "Applying naive bayes data mining technique for classification of agricultural land soils", *International journal of computer science and network security*, *9*(8), 117-122, (2009).

Boiy, E., Hens, P., Deschacht, K., & Moens, M. F., "Automatic Sentiment Analysis in On-line Text", In *ELPUB*, 349-360, (2007).

Choi, D., Ko, B., Kim, H., & Kim, P., "Text analysis for detecting terrorism-related articles on the web", *Journal of Network and Computer Applications*, *38*, 16-21, (2014).

Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A., and Buyya, R., "Big Data computing and clouds: Trends and future directions", *Journal of Parallel and Distributed Computing*, *79*, 3-15, (2015).

Coletta, L. F. S., da Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R., "Combining classification and clustering for tweet sentiment analysis", In *Intelligent Systems (BRACIS), 2014 Brazilian Conference on* IEEE, 210-215, (2014).

Çoban, Ö., Özyer, B., & Özyer, G. T., "Sentiment analysis for Turkish Twitter feeds", In *Signal Processing and Communications Applications Conference (SIU), 2015 23th* IEEE, 2388-2391, (2015).

Dave, K., Lawrence, S., & Pennock, D. M., "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews", In *Proceedings of the 12th international conference on World Wide Web*, 519-528, (2003).

Ghag, K., and Shah, K., "Comparative analysis of the techniques for Sentiment Analysis", In *Advances in Technology and Engineering (ICATE), 1-7*, (2013).

Gu, L., & Li, H., "Memory or time: Performance evaluation for iterative operation on hadoop and spark", In *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC)*, 721-727, (2013).

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

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X., *Applied logistic regression* (Vol. 398), (2013).

Kang, D., and Park, Y., "Review-based measurement of customer satisfaction in mobile service: Sentiment analysis and VIKOR approach", *Expert Systems with Applications*, *41*(4), 1041-1050, (2014).

Medhat, W., Hassan, A., and Korashy, H., "Sentiment analysis algorithms and applications: A survey", *Ain Shams Engineering Journal*, *5*(4), 1093-1113, (2014).

Mostafa, M. M., "More than words: Social networks' text mining for consumer brand sentiments", *Expert Systems with Applications*, *40*(10), 4241-4251, (2013).

Onan, A., & Korukoğlu, S., "A review of literature on the use of machine learning methods for opinion mining", *Pamukkale University Journal of Engineering Sciences*, *22*(2), 111–122, (2016).

Pang, B., Lee, L., & Vaithyanathan, S., "Thumbs up?: sentiment classification using machine learning techniques", *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 79-86, (2002).

Razzaq, M. A., Qamar, A. M., & Bilal, H. S. M., "Prediction and analysis of Pakistan election 2013 based on sentiment analysis", In *Advances in Social Networks Analysis and Mining (ASONAM),* 700-703, (2014).

Safavian, S. R., & Landgrebe, D., "A survey of decision tree classifier methodology", *IEEE transactions on systems, man, and cybernetics*, *21*(3), 660-674, (1991).

Seker, S. E., "Sosyal Ağlarda Veri Madenciliği (Data Mining on Social Networks)", *YBS Ansiklopedi*, *2*(2), 30-39, (2015).

Setty, S., Jadi, R., Shaikh, S., Mattikalli, C., & Mudenagudi, U., "Classification of facebook news feeds and sentiment analysis", In *Advances in Computing, Communications and Informatics (ICACCI),*18-23, (2014).

Shahheidari, S., Dong, H., & Daud, M. N. R. B., "Twitter sentiment mining: A multi domain analysis", In *Complex, Intelligent, and Software Intensive Systems (CISIS),* 144-149, (2013).

Vinodhini, G., & Chandrasekaran, R. M., "Sentiment analysis and opinion mining: a survey", *International Journal*, *2*(6), 282-292, (2012).

Vishwanathan, S., "Sentiment Analysis of Movie Reviews", In *Proceedings of 3rd IRF International Conference*, (2014).

Wöllmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K., & Morency, L. P., "Youtube movie reviews: Sentiment analysis in an audio-visual context", *IEEE Intelligent Systems*, *28*(3), 46-53, (2013).

9. ÖZGEÇMİŞ

Adı Soyadı	:	Merve ÖZDEŞ
Doğum Yeri ve Tarihi	:	Ankara, 21.03.1990
Lisans Üniversite	:	Trakya Üniversitesi
Elektronik posta	:	mozdes@pau.edu.tr
İletişim Adresi	:	Pamukkale Üniversitesi