T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

VERİ MADENCİLİĞİ YÖNTEMLERİYLE SOSYAL MEDYA DUYGU ANALİZİ

BATUHAN CEM ÖĞE

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN
DR. ÖĞR. ÜYESİ FATİH KAYAALP

DÜZCE, 2021

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

VERİ MADENCİLİĞİ YÖNTEMLERİYLE SOSYAL MEDYA DUYGU ANALİZİ

Batuhan Cem ÖĞE tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Fatih KAYAALP

Düzce Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Fatih KAYAALP
Düzce Üniversitesi _____

Prof. Dr. Kemal POLAT
Bolu Abant İzzet Baysal Üniversitesi _____

Doç. Dr. Abdullah Talha KABAKUŞ
Düzce Üniversitesi _____

Tez Savunma Tarihi: 27/08/2021

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

27 Ağustos 2021

Batuhan Cem ÖĞE

TEŞEKKÜR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı danışman hocam Dr. Öğr. Üyesi Fatih KAYAALP'e en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen başta kardeşim olmak üzere sevgili aileme sonsuz teşekkürlerimi sunarım.

27 Ağustos 2021 **Batuhan Cem ÖĞE**

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

ÇİZELGE LİSTESİ

Sayfa No

KISALTMALAR

ANN	Artificial Neural Network
DDİ	Doğal Dil İşleme
DT	Decision Tree
IMDB	Internet Movie Database
LR	Logistic Regression
NB	Naive Bayes
RF	Random Forest
SVM	Support Vector Machine

ÖZET

VERİ MADENCİLİĞİ YÖNTEMLERİYLE SOSYAL MEDYA DUYGU ANALİZİ

Batuhan Cem ÖĞE
Düzce Üniversitesi
Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi
Danışman: Dr. Öğr. Üyesi Fatih KAYAALP
Ağustos 2021, 44 sayfa

Son yıllarda internete erişim imkanlarının artması ve kullanıcılardaki akıllı telefon kullanımının yaygınlaşması sebebiyle sosyal medya olarak adlandırılan ve insanların çeşitli konulardaki fikirlerini paylaştığı servisler çok yaygın olarak kullanılmaya başlamıştır. Kullanıcılar tarafından bu servislere girilen ve birçok farklı platformda depolanmakta olan veriler çeşitli veri madenciliği yöntemleriyle analiz edilerek anlamlı bilgi çıkarımları yapılmaya çalışılmaktadır. Sosyal medya verilerinin analiz edilmesiyle insanların farklı konulardaki duygularına dair anlamlı çıkarımlarda bulunulması anlamına gelen Duygu Analizi çalışmaları da bu konuda öne çıkan çalışma alanlarından biridir. Duygu Analizi, insanların görüşlerinin olumlu, olumsuz veya nötr gibi çeşitli sınıflara göre kategorize edilmesi işlemidir. Ve işletmeler açısından müşterilerinin davranış eğilimlerinin anlaşılması, hastaların ruh sağlığının değerlendirilmesi, insanların çeşitli toplumsal olaylarda verdiği tepkilerin ortaya çıkarılması gibi birçok sektörde aktif olarak kullanılmaktadır. Bu tez çalışmasında, kullanıcıların IMDB internet sitesinde paylaşmış oldukları film yorumlarından oluşan etiketli bir veri seti, çeşitli veri madenciliği yöntemleri kullanılarak sınıflandırılmış, Python, Matlab ve R programlama dilleri ile duygu analizi çalışması gerçekleştirilmiş ve elde edilen sonuçlar farklı değerlendirme kriterlerine göre karşılaştırılmıştır.

Anahtar sözcükler: Doğal dil işleme, Duygu analizi, Metin temsil, Sınıflandırma, Veri madenciliği.

ABSTRACT

SOCIAL MEDIA SENTIMENT ANALYSIS WITH DATA MINING TECHNIQUES

Batuhan Cem ÖĞE
Düzce University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering
Master's Thesis
Supervisor: Asst. Prof. Fatih KAYAALP
August 2021, 44 pages

Due to the increase in internet access opportunities and the widespread use of smartphones in recent years, the services called social media, where people share their opinions on various issues, have started to be used widely. The data entered by users into these services and stored on many different platforms are analyzed by various data mining methods and meaningful information inferences are tried to be made. Sentiment Analysis studies, which means making meaningful inferences about people's feelings on different subjects by analyzing social media data, is one of the prominent fields of study in this regard. Sentiment Analysis is the process of categorizing people's opinions according to various classes such as positive, negative or neutral. And in terms of businesses, it is actively used in many sectors such as understanding the behavioral tendencies of its customers, evaluating the mental health of patients, revealing the reactions of people in various social events. In this thesis, a labeled data set consisting of movie reviews shared by users on the IMDB website was classified using various data mining methods, a sentiment analysis study was carried out with Python, Matlab and R programming languages, and the results were compared according to different evaluation criteria.

Keywords: Classification, Data mining, Natural language processing, Sentiment analysis, Text representation.

1. GİRİŞ

1.1. ÇALIŞMANIN AMACI

Son yıllarda internete erişim imkanlarının artması ve kullanımı gittikçe artan, insanların çeşitli konulardaki fikirlerini paylaştığı servisler olarak adlandırılan sosyal medya platformlarının çoğalması ile çok büyük miktarda işlenmemiş veri ortaya çıkmıştır. Kullanıcılar tarafından bu servislere girilen ve birçok farklı platformda depolanmakta olan veriler çeşitli veri madenciliği yöntemleriyle analiz edilerek anlamlı bilgi çıkarımları yapılmaya çalışılmaktadır. Sosyal medya verilerinin analiz edilmesiyle insanların farklı konulardaki duygularına dair anlamlı çıkarımlarda bulunulması anlamına gelen Duygu Analizi çalışmaları da bu konuda öne çıkan çalışma alanlarından biridir. İnsanların sanal ortamlarda gerçek düşüncelerini daha rahat ve özgürce paylaştığı düşünüldüğünde, paylaşılan mesajlardan duygu analizi yapabilmek, yüz yüze sohbet edilen bir insanın duygusunu anlamaktan çok daha kolay olabilmektedir. Çünkü herhangi bir sokak röportajında ya da karşılıklı bir sohbet esnasında insanlar gerçek duygularını gizleyebilmekte, bir şey anlatmak isterken aslında başka bir şey ifade edebilmektedirler. Ancak sosyal medyaya girildiğinde insanlar kendilerini daha özgür ve ulaşılmaz hissettiklerinden bu tür ortamlarda görüşlerini daha doğal olarak ortaya atabilmektedirler. Bu durum sosyal medya platformlarında paylaşılan her türlü görüntülü, sesli veya yazılı mesajın analizinin sonucunda insanların duyguları, görüşleri hakkında daha sağlıklı bir bilgi elde etmemize olanak sağlamaktadır. Duygu analizi yardımı ile insanların yaptıkları paylaşımlardan anlamlı sonuçlar elde edilebilmektedir.

Duygu analizi, bir metinden fikir çıkarmak, dönüştürmek ve yorumlamak ve bunları olumlu, olumsuz veya nötr olarak sınıflandırmak için Doğal Dil İşleme'yi (DDİ) kullanan bir yaklaşımdır [1].

Önceki çalışmaların çoğunda duygu analizi, müşterilerini daha iyi anlamak ve ürün veya hizmetlerini geliştirmek için gerekli kararı vermek adına bir ürün veya film incelemesinde kullanılmıştır [2].

Duygu analizi, cümle düzeyi, belge düzeyi ve özellik düzeyi olmak üzere üç farklı düzeye

ayrılmıştır. Amaç, düşünceyi cümle, belge veya özelliklerden olumlu ve olumsuz duygu olarak sınıflandırmaktır [3].

Makine öğrenmesi yaklaşımı ve sözlük tabanlı yaklaşım olarak adlandırılan iki ana duygu analizi yöntemi mevcuttur. Makine öğrenmesi yaklaşımı, bir veriden duyguyu çıkarmak ve tespit etmek için algoritmalar kullanırken, sözlük tabanlı yaklaşım, verilerle ilgili olumlu ve olumsuz kelimeleri sayarak çalışır. Bilim insanları, duygu analizinde yeni, etkili ve doğru bir model geliştirmek için çalışmalarını sürdürürken tasarımın çoğunun İngilizce dili için olduğu düşünüldüğünde diğer diller açısından bazı sorunlar ortaya çıkabilmektedir [4].

Duygu analizinin uygulanmasına gelince, iş ve pazarlama, siyaset ve kamusal eylem bağlamında yapıldığı görülmektedir. Uygulama alanları olarak e-ticaret siteleri, oylama uygulamaları ve insanların dünya olayları hakkındaki görüşleri verilebilir [5].

1.2. LİTERATÜR TARAMASI

2018 yılında yayınlanan bir çalışmada araştırmacılar Amazon tarafından sunulan ürünler ile ilgili yorumların yer aldığı veri setini kullanarak duygu analizi çalışması yapmışlar ve Destek Vektör Makinesi (Support Vector Machine) ile Rastgele Orman (Random Forest) algoritmalarının karışımı olan bir algoritma geliştirmişlerdir. Elde ettikleri sonuç iki yöntemin bireysel olarak kullanılmasına göre daha iyi sonuç vermiştir [6].

Yapılan bir diğer çalışmada, 400 binden fazla tüketici yorumu ile duygu analizinde, kullanıcı yorumları word2vec yöntemiyle vektör gösterim şekline dönüştürülerek analiz edilmiş, 10 katlı çapraz doğrulama (10-fold cross validation) yöntemi kullanılarak Destek Vektör Makinesi (Support Vector Machine), Naif Bayes (Naive Bayes), Lojistik Regresyon (Logistic Regression) ve Rastgele Orman (Random Forest) algoritmaları yardımıyla sonuçlar elde edilmiştir. Elde edilen sonuçlar incelendiğinde en yüksek doğruluk değerine Rastgele Orman (Random Forest) algoritması ile ulaşılmıştır [7].

Veri ön işleme adımlarının değerlendirmesinin yapıldığı bir diğer çalışmada, Twitter mesajlarından oluşan veri seti için duygu analizi çalışması yapılmış, Destek Vektör Makinesi (Support Vector Machine), Naif Bayes (Naive Bayes), Lojistik Regresyon (Logistic Regression) ve Evrişimli Sinir Ağır (Convolutional Neural Network) gibi 4 popüler makine öğrenmesi algoritması yardımı ile elde edilen sonuçlar karşılaştırılmıştır. Yapılan gözlemler sonucunda kelimeleri kök hallerine dönüştürme, rakamları silme,

kesme işaretini kaldırma gibi yöntemlerin doğruluk oranını olumlu yönde etkilediği, noktalama işaretleri kaldırmanın ise doğruluk değerine çok fazla katkı sağlamadığı görülmüştür [8].

Yapılan bir diğer çalışmada, Twitter mesajlarından kullanıcıların duygularını otomatik olarak çıkarmayı hedefleyen bir model geliştirilmiş ve Destek Vektör Makinesi (Support Vector Machine) yöntemi uygulanarak geliştirilen bu model %98 doğruluk değerine ulaşmıştır [9].

Facebook mesajlarının veri seti olarak kullanıldığı bir diğer çalışmada, Endonezya başkanlık seçimleri öncesinde sosyal medyada paylaşılan mesajlar Naif Bayes (Naive Bayes) sınıflandırma algoritması kullanılarak analiz edilmiş ve elde edilen sonuçlar ile gerçek sonuçlar karşılaştırılmıştır [10].

Yapılan bir diğer çalışmada, davranışsal ekonomi ve online ekonomi çevrelerinin karakteristikleri dikkate alınarak ekonomi blog'ları ve online gazete yayınları incelenmiş ve piyasalar için dinamik halk görüşü geliştirilmiştir. 18 aylık bir süreçte çeşitli internet siteleri ve borsa bilgilerinin paylaşıldığı platformlardan elde edilen veri seti ile geliştirilen model, günlük gelişen olayların hisse fiyat tahminlerinde etkili olduğunu ortaya çıkarmıştır [11].

Aşı ve aşılanma ile ilgili tweet'lerden oluşan bir veri setinin kullanıldığı çalışmada, insanların aşı ile ilgili görüşleri kelime torbası (bag of words) ve Destek Vektör Makinesi (Support Vector Machine) yöntemleri kullanılarak analiz edilmiş ve sonuçlar sunulmuştur [12].

Yapılan bir başka çalışmada, LGBT karşıtı eylemlerin Endonezya'da geniş çaplı şekilde tartışıldığı dönemde Twitter'da yapılan paylaşımlar incelenmiş, insanların paylaştığı tweet'lerden oluşan veri seti Naif Bayes (Naive Bayes), Karar Ağacı (Decision Tree) ve Rastgele Orman (Random Forest) yöntemleriyle analiz edilmiş ve sonuçlar elde edilmiştir. Paylaşılan tweet'lerin büyük çoğunluğu konu ile ilgili nötr duygu içerirken en iyi sonucu Naif Bayes (Naive Bayes) algoritması vermiştir [13].

2010 Şili depremi ve 2017 Katalan Bağımsızlık referandumu ile ilgili Twitter üzerinde yapılan paylaşımların incelendiği bir çalışmada, duygu analizi için Bayes Ağları sınıflandırıcısı kullanılmış, daha gerçekçi ağlar elde edilebilmesi için Bayes faktör yaklaşımı uygulanmıştır. Geliştirilen bu model Destek Vektör Makinesi (Support Vector Machine) ve Rastgele Orman (Random Forest) algoritmalarına göre daha iyi sonuçlar

vermiştir [14].

Duygu analizinde özellik çıkarımının etkisinin incelendiği bir çalışmada, araştırmacılar Twitter paylaşımlarından oluşan bir veri seti için TF-IDF ve n-gram özellik çıkarım yöntemlerini incelemiş ve 6 farklı sınıflandırma algoritması ile performans kriterlerini elde ederek karşılaştırmalar yapmıştır. Elde edilen sonuçlara göre TF-IDF %3-4 oranında n-gram yöntemine göre daha iyi sonuçlar vermiştir [15].

Film yorumlarından ve Twitter paylaşımlarından oluşan iki farklı veri seti ile yapılan bir çalışmada, Python programlama diliyle farklı sınıflandırma modelleri (Naif Bayes (Naive Bayes), Destek Vektör Makinesi (Support Vector Machine), Yapay Sinir Ağları (Artificial Neural Network)) ve özellik çıkarım yöntemleri (TF-IDF ve word2vec) kullanılmış ve elde edilen sonuçlar karşılaştırılmıştır. Elde edilen sonuçlar karşılaştırıldığında Yapay Sinir Ağları (Artificial Neural Network) yöntemi diğer yöntemlere göre daha iyi performans göstermiştir [16].

1.3. TEZİN ORGANİZASYONU

Tez çalışmasının ilk bölümünde, seçilen konu başlığının önemi ve bu konu başlığı ile ilgili daha önce yapılmış olan çalışmalardan bahsedilmiştir.

Tezin ikinci bölümünde, tez çalışması ile ilgili teknik arka plandan bahsedilmiştir. Veri madenciliği, veri madenciliği kullanım alanları, veri madenciliği süreçleri, veri madenciliği fonksiyonları gibi kavramlar açıklanmış, tez çalışmasında kullanılan sınıflandırma yöntemlerinden bahsedilmiştir. Ardından, metin madenciliği, veri seti oluşturma, etiketleme, metin ön işleme adımları, veri setinin bölünmesi, vektörleştirme gibi kavramlar açıklanmış, tez çalışmasında kullanılan metin temsil yöntemlerinden bahsedilmiştir. Son olarak, kullanılan programlama dilleri ve bu programlama dillerinin hangi platformlar aracılığı ile kullanıldığından bahsedilmiştir.

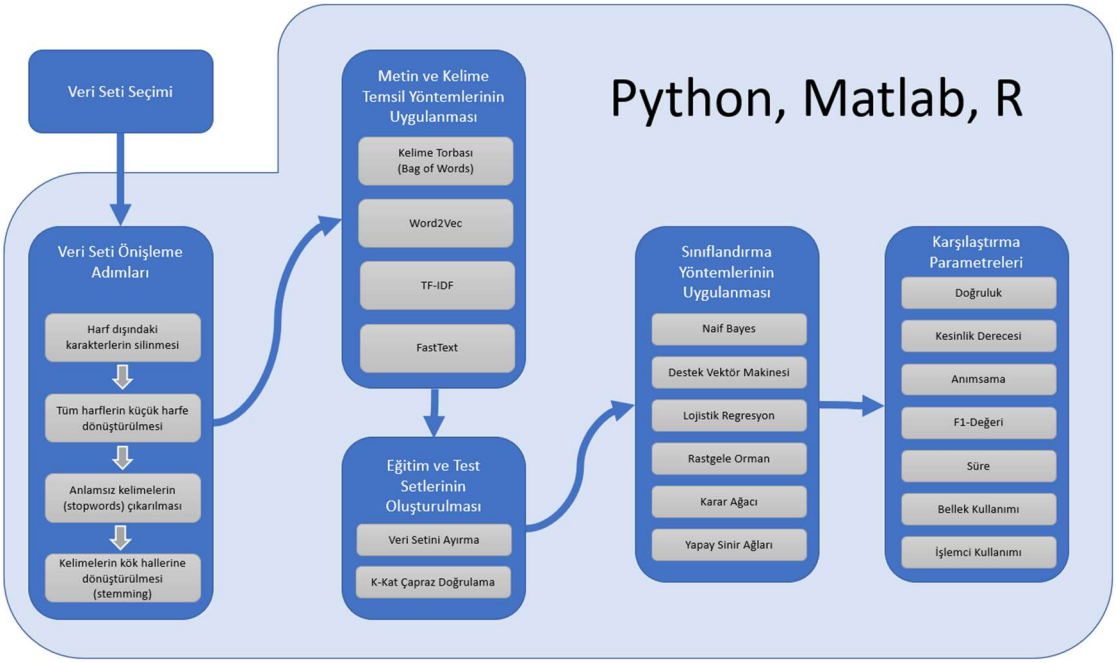
Tezin üçüncü bölümünde, duygu analizi yapılan veri seti hakkında bilgi verilmiş, kullanılan yöntem ve analiz sürecinin basamakları anlatılmıştır. Veri setinin kullanıma hazır hale getirilmesi, veri temizleme aşamaları detaylı şekilde açıklanmış, veri setinin eğitim ve test setleri olarak nasıl ayrıldığından bahsedilmiş, metin ve kelime temsil yöntemlerinin uygulanması hakkında bilgi verilmiş, kullanılan sınıflandırma yöntemlerinin nasıl uygulandığı ve gerçeklendiğinden bahsedilmiştir. Ayrıca, kullanılan sınıflandırma yöntemleri açıklanmış, tüm simülasyonların yapıldığı bilgisayar ortamının

donanım bilgisi verilmiş ve elde edilen sonuçların hangi değerler kullanılarak karşılaştırılacağı açıklanmıştır.

Tezin dördüncü bölümünde, değerlendirme ölçütleri açıklanmış ve değerlendirme ölçütlerine göre her bir programlama dili için elde edilen sonuçlar kullanılan metin temsil ve sınıflandırma yöntemleri ile birlikte tablo halinde verilmiş, en iyi ve en kötü performansı gösteren sınıflandırma yöntemleri belirtilmiştir.

Tezin beşinci bölümünde ise elde edilen sonuçlar değerlendirilerek ileride yapılabilecek çalışmalar hakkında önerilerde bulunulmuştur.

Tüm tez boyunca uygulanan işlemlerin blok diyagramı Şekil 1.1'de gösterilmiştir.



Şekil 1.1. Tezin Blok Diyagramı.

2. TEKNİK ARKA PLAN

2.1. VERİ MADENCİLİĞİ

Veri madenciliği, sonuçları tahmin etmek ve tanımlama işlemleri için büyük veri kümeleri içindeki anormallikleri, kalıpları ve bağlantıları bulma sürecidir. Çeşitli yöntemler kullanılarak, bu bilgileri gelirleri artırmak, maliyetleri azaltmak, müşteri ilişkilerini geliştirmek, riskleri azaltmak ve çok daha fazlası için kullanmak mümkündür.

2.1.1. Veri Madenciliği Kullanım Alanları

Veri Madenciliği çeşitli alanlarda kullanılmaktadır. Telekomünikasyon, medya ve teknoloji şirketleri, müşteri davranışlarını tahmin etmek, hedefe yönelik ve müşteri odaklı kampanyalar sunmak adına veri madenciliği yöntemlerini kullanırlar.

Veri madenciliği, eğitimcilerin öğrenci verilerine erişmesine, başarı düzeylerini tahmin etmesine ve ekstra dikkat gerektiren öğrencileri veya öğrenci gruplarını belirlemesine yardımcı olur.

Otomatik algoritmalar, bankaların müşteri portföylerini ve finansal sistemin kalbindeki milyarlarca işlemi anlamalarına yardımcı olur. Veri madenciliği, finansal hizmet şirketlerinin piyasa risklerini daha iyi görmelerine, sahtekarlığı daha hızlı tespit etmelerine, mevzuata uygunluk yükümlülüklerini yönetmelerine ve pazarlama yatırımlarından optimum getiri elde etmelerine yardımcı olur.

Tedarik planlarını talep tahminleriyle uyumlu hale getirmek, sorunların erken tespiti, kalite güvencesi ve marka değerine yapılan yatırım kadar önemlidir. Üreticiler, veri madenciliği yöntemleri ile üretilen ürünlerin yıpranma ve bakım sürelerini tahmin edebilirler. Bu durum çalışma süresini en üst düzeye çıkarır ve üretim hattının plana uygun şekilde çalışmasına olanak sağlar.

2.1.2. Veri Madenciliği Süreçleri

Veri madenciliği süreci, veri toplama, veri temizleme ve dönüştürme, model kurma, model değerlendirme, raporlama, değerlendirme, uygulama entegrasyonu ve model yönetimi gibi adımlardan oluşur.

2.1.3. Veri Madenciliği Fonksiyonları

Veri madenciliği fonksiyonları temelde iki başlık altında gruplandırılabilir. Bu fonksiyonlar alt başlıklarıyla birlikte aşağıdaki listelenmiştir:

- Tahmin Edici (Predictive) Fonksiyonlar
    - Sınıflandırma
    - Regresyon
- Tanımlayıcı (Descriptive) Fonksiyonlar
    - Kümeleme
    - Birliktelik Kuralları
    - Ardışık Zamanlı Örüntüler

Bu tez çalışmasında tahmin edici fonksiyonlar içerisinde yer alan sınıflandırma yöntemi kullanılmıştır. Kullanılan sınıflandırma yöntemlerinden başlıklar halinde aşağıda bahsedilmiştir.

2.1.3.1. *Naif Bayes (Naive Bayes)*

Özellikleri arasında güçlü bağımsızlık varsayımlarının olduğu Bayes Teoremine dayanan bir sınıflandırma yöntemidir. Bir Naif Bayes sınıflandırıcısı, bir sınıftaki belirli bir özelliğin (elemanın) yakınlığının diğer bazı elemanların yakınlığıyla bağlantısının kesilmesini bekler. Örneğin, rengi kırmızı, şekli yuvarlak ve genişliği yaklaşık üç santimetre olan organik bir meyve elma olarak kabul edilebilir. Bu özelliklerin birbirine veya diğer özelliklerin varlığına bağlı olup olmadığına bakılmaksızın, bir Naif Bayes sınıflandırıcısı, bu doğal meyvenin bir elma olma olasılığı nedeniyle bu özellikleri bağımsız olarak değerlendirecektir. Naif Bayes'in, kolaylığın yanı sıra son derece modern sıralama stratejilerini bile yerine getirdiği bilinmektedir. Naif Bayes, metinleri birden çok sınıfa sınıflandırma görevinde yaygın olarak kullanılmaktadır ve yakın zamanda duygu analizi sınıflandırması için kullanılmıştır [17].

2.1.3.2. *Destek Vektör Makinesi (Support Vector Machine)*

Destek Vektör Makinesi (Support Vector Machine) yönteminin duygu analizinde iyi performans gösterdiği bilinmektedir [18]. Destek Vektör Makinesi bilgiyi araştırır, seçim sınırlarını karakterize eder ve girdi uzayında gerçekleştirilen hesaplama için bileşenleri kullanır. Kritik bilgi, her biri m boyutunda olan iki vektör düzenlemesinde sunulur. Bu

noktada, her veri (vektör olarak ifade edilir) bir sınıfa sıralanır. Daha sonra makine, eğitim örneklerinde herhangi bir yerden uzak olan iki sınıf arasındaki sınırı tanımlar. Ayrım, sınıflandırma sınırını karakterize eder, kenarı genişletmek kararsız seçenekleri azaltır. Destek Vektör Makinesi'nin çeşitli metin sınıflandırma problemlerinde Naif Bayes sınıflandırıcısından daha etkili bir şekilde çalıştığı kanıtlanmıştır [19].

2.1.3.3. *Lojistik Regresyon (Logistic Regression)*

Lojistik Regresyon (Logistic Regression), adına rağmen Genelleştirilmiş Doğrusal Modeller yöntemlerine ait popüler bir algoritmadır ve Maksimum Entropi olarak da bilinir. Bu modelde, tek bir denemenin olası sonuçlarını tanımlayan olasılıklar, bir lojistik fonksiyon kullanılarak modellenir. Lojistik Regresyon yöntemi, bir veya birden fazla bağımsız değişken olduğunda çıktıyı veya sonucu belirlemek için kullanılır. Çıkış değeri 0 veya 1, yani ikili biçimde olur [20].

2.1.3.4. *Karar Ağacı (Decision Tree)*

Karar Ağacı (Decision Tree) sınıflandırmasında, verileri bölmek için bir koşul kullanılır. Koşulu sağlayan veriler bir sınıfa, kalan veriler diğer sınıfa yerleştirilir. Bu yinelemeli bir süreçtir. Birden fazla ayırma yöntemi vardır. Bunlar, sınıflandırma yapmak için belirli kelime varlığını veya yokluğunu arayan tek öznitelik bölme ve belgedeki kelimeleri önceden tanımlanmış kelimelerle eşleştiren benzerlik tabanlı çoklu öznitelik bölmedir [21].

2.1.3.5. *Rastgele Orman (Random Forest)*

Rastgele Orman (Random Forest), sınıflandırma ve regresyon için öğrenme yöntemidir. Eğitim aşamasında bir dizi karar ağacı oluşturulur. Yeni gelen durumu sınıflandırmak için yeni durum ağaçların her birine gönderilir. Her ağaç sınıflandırma yapar ve sonuç olarak bir sınıf çıkarır. Çıktı sınıfı, çoğunluk oylamasına dayalı olarak çeşitli ağaçlar tarafından oluşturulan maksimum benzer sınıf sayısı dikkate alınarak seçilir. Rastgele Orman yöntemi, hem profesyoneller hem de sıradan insanlar için çok az araştırma ve programlama gerektirir, öğrenmesi ve kullanması kolaydır. Güçlü bir istatistiksel altyapıya sahip olmayan kişiler tarafından bile rahatlıkla kullanılabilir [22].

2.1.3.6. *Yapay Sinir Ağları (Artificial Neural Network)*

Farklı makine öğrenimi algoritmaları arasında Yapay Sinir Ağları (Artificial Neural Network) bu alanda daha az ilgi görmüş ancak son zamanlarda daha fazla ilgi ve

popülerlik kazanmıştır. Yapay Sinir Ağları'nın ana fikri, girdi verilerinin doğrusal birleşiminden öznitelikleri çıkarmak ve ardından çıktıyı bu özniteliklerin doğrusal olmayan bir işlevi olarak modellemektir. Sinir ağları genellikle, bağlantılarla bağlanan düğümleri içeren bir ağ şeması olarak görüntülenir. Düğümler bir katmanda düzenlenir ve ortak sinir ağlarının mimarisi üç katman içerir: giriş katmanı, çıkış katmanı ve gizli katman. İki tür sinir ağı vardır, ileri ve geri besleme. İleri ağda düğümler yalnızca bir yönde bağlı olduğundan, duygu sınıflandırması için uygundur. Her bağlantının, bir gradyan iniş eğitim sürecinde global bir hata fonksiyonunu en aza indirerek tahmin edilen, karşılık gelen bir ağırlık değeri vardır. Bir nöron, iki adımda bir değer veren basit bir matematiksel modeldir. İlk adımda, nöron girdisinin ağırlıklı toplamını hesaplar ve ardından bu toplama bir aktivasyon fonksiyonu uygulayarak çıktısını alır. Aktivasyon işlevi tipik olarak doğrusal olmayan bir işlevdir ve tüm ağın giriş verilerinden önceden öğrenilen doğrusal olmayan bir işlevi tahmin edebilmesini sağlar [23].

2.2. METİN MADENCİLİĞİ

Metin madenciliği, belgelerdeki ve veri tabanlarındaki yapılandırılmamış metni, analize uygun normalleştirilmiş, yapılandırılmış verilere dönüştürmek için doğal dil işlemeyi kullanan bir yapay zeka teknolojisidir.

2.2.1. Veri Seti Oluşturma

Metin madenciliği çalışmalarında kullanılan veri setleri oluşturulurken genellikle insanların sosyal medya platformlarında paylaştığı mesajlar kullanılır. Veri toplama aşamasında sosyal medya platformlarında paylaşılan (Facebook, Twitter, internet sayfalarındaki yorumlar vb.) kullanıcıların paylaştıkları görüşler, yorumlar alınarak veri setleri oluşturulur.

2.2.2. Etiketleme

Metin madenciliği çalışmalarında kullanılan veri setleri ile verimli sonuçlar elde edilebilmesi, insanların paylaştıkları yorumlardan yeni şeylerin tahmin edilebilmesi için elde edilen veri setinin etiketlenmesi gerekir. Özellikle duygu analizi kapsamında yapılan çoğu çalışma ikili sınıflandırma üzerine olduğu için (pozitif, negatif), veri setinde yer alan yorumların pozitif veya negatif şeklinde etiketlenmeleri gerekir. Veri seti oluşturulurken etiketlenmeyen yorumlar analizi zorlaştıracak ve sonucu olumsuz etkileyecektir.

2.2.3. Metin Ön İşleme İşlemleri

Sosyal medya uygulamalarında kullanıcıların yaptığı yorumlardan oluşturulan veri setlerinin analiz edilebilmesi ve verimli sonuçların elde edilebilmesi için oluşturulan veri setinin bir takım ön işlemlere tabi tutulması gerekir. Kullanıcıların yaptıkları yorumlarda veya paylaşımlarda duyguyu ifade etmeyen, analize herhangi bir anlam katmayacak, aksine çalışma süresini ve donanımı zorlayacak şekilde ifadeler mevcuttur. Bu nedenle tüm veri setinden analize katkı sağlamayacak ve anlam karmaşasına yol açacak tüm ifadelerin çıkarılması, verinin temizlenmesi gerekir. En sık kullanılan metin ön işleme yöntemleri şöyledir:

2.2.3.1. *Unicode Dizileri ve Gürültü Temizleme*

Veri seti oluşturulurken veriye eklenen kod satırları veya İngiliz alfabesinde yer almayan tüm karakterler (gürültü) veri setinden temizlenir.

2.2.3.2. *URL Adresleri ve Kullanıcı Adları Silme*

Kullanılan veri setinin alındığı platforma göre yorumlar içerisinde kullanıcı adları veya internet sayfalarına ait linkler yer alabilir. Bu tür ifadelerin analizlere bir katkısı olmayacağından silinmeleri gerekmektedir.

2.2.3.3. *Argo ve Kısaltma İfadelerin Silinmesi*

Yorumlarda yer alan argo ifadeler veya kısaltmalar özellikle duygu analizi için bir anlam ifade etmediğinden bu tarz ifadeler veri setinden silinir.

2.2.3.4. *Kesme İşaretlerinin Kaldırılması*

Yorum satırlarında yer alabilecek “don’t”, “won’t” gibi ifadelerin verimli bir şekilde analiz edilebilmesi için “do not” ve “will not” şekline dönüştürülmesi gerekir. Bu nedenle veri setindeki tüm kesme işaretleri kaldırılır.

2.2.3.5. *Sayıların Silinmesi*

Herhangi bir duygu ifadesi içermedikleri için veri setinden tüm sayıların çıkarılması gerekmektedir.

2.2.3.6. *Tekrar Eden Noktalama İşaretlerinin Düzenlenmesi*

Veri setinde yer alan ifadeler içerisinde bulunan noktalama işaretlerinden arka arkaya kullanılanlardan sadece bir tane kalacak şekilde veri setinin düzenlenmesi gerekir. Aksi takdirde gereksiz karakterler analizi olumsuz yönde etkileyecektir.

2.2.3.7. *Zıtlık İfade Eden Kelimelerin Düzenlenmesi*

İngilizce dilinde yorumların yer aldığı veri setleri için “not good” (iyi değil) tarzında ifadeler “not” kelimesi olumsuz, “good” kelimesi olumlu bir duygu ifade ettiği için zorluk çıkarabilir. Bu nedenle bu tarz bir ifadenin “not good” yerine “bad” (kötü) olacak şekilde değiştirilmesi gerekmektedir.

2.2.3.8. *Noktalama İşaretlerinin Kaldırılması*

Veri setinde yer alan tüm noktalama işaretlerinin kaldırılması işlemidir.

2.2.3.9. *Büyük Harfle Yazılan Kelimelerin Küçük Harfe Dönüştürülmesi*

Bir veri seti içerisinde geçen “good”, “Good”, “GOOd”, “GOOD” tarzı ifadelerin hepsi aynı kelime ve aynı anlamı ifade etmektedir. Anlam karmaşasını ortadan kaldırmak ve aynı kelimenin tek bir ifade ile temsil edilebilmesi için veri setinde yer alan tüm harfler küçük harfe dönüştürülür.

2.2.3.10. *Stopwords (Anlamsız) Kelimelerin Çıkarılması*

Duygu analizi çalışmalarında veri setlerinde yer alan ve herhangi bir duygu ifade etmeyen kelimelerin veri setinden çıkarılması işlemidir. Bunun için “stopwords” sözlükleri kullanılır ve bu sözlük içerisinde yer alan tüm kelimeler çalışılan veri setinden çıkarılır.

2.2.3.11. *Uzatılmış Kelimelerin Düzenlenmesi*

Yorum yapılırken vurgu amaçlı veya yazım hatası nedeniyle olması gerektiği gibi yazılmayan (“good” yerine “ggooooddd” gibi) kelimelerin düzenlenmesi işlemidir.

2.2.3.12. *Yazım Hatalarının Düzeltilmesi*

Veri setinde yer alan yorumlar içerisindeki bazı kelimeler kullanıcılar tarafından istemli veya istemsiz olarak yanlış yazılabilir. Bu durumun yapılan çalışmayı olumsuz etkilememesi için “good” yerine “g00d” şeklinde yazılan kelimelerin düzenlenmesi ve olması gerektiği hallerine dönüştürülmesi gerekir.

2.2.3.13. *Kelime Köklerinin Elde Edilmesi*

Veri seti analiz edilmeden önce içerisinde yer alan tüm kelimelerin kök hallerine dönüştürülmesi gerekir. Aksi halde analiz sonuçları olumsuz yönde etkilenebilir. Kelime kökünün elde edilmesi metin ön işleme aşamalarından en sık kullanılanıdır.

2.2.4. Veri Setinin Bölünmesi

Metin madenciliği çalışmalarında kullanılan veri setinin eğitim ve test olarak ayrılması gerekmektedir. Tüm verinin eğitim için kullanılmayıp, bir bölümünün test için kullanılmasına metin madenciliğinde "hold-out" (veri setini bölme) denir. Veri seti analiz için farklı oranlarda bölünebilir. Veri seti eğitim ve test için %50-%50, %60-%40, %70-%30 veya %80-%20 gibi çeşitli oranlarda ayrılabilir. Bu oranlardan en sık kullanılanı %80 ve %20'dir. Veri setinin eğitim ve test için bölünmesinde karşılaşılabilecek sorunlardan bir tanesi verinin sadece belirli bir bölümü ile eğitim yapılması ve belirli bir bölümü ile test edilmesidir. Bu şekilde yapılan bir çalışmada tüm veri eğitim ve test için kullanılmamış olur. Sadece istenen bir bölümü ile eğitilmiş ve istenen bir bölümü ile testler yapılmış olur. Bu da elde edilen sonuçlar hakkında soru işaretleri doğurabilir. Bu nedenle veri setinin bölünmesi aşamasında k katlı çapraz doğrulama (k-fold cross validation) yöntemi kullanılarak bu sorunun önüne geçilebilir. Bu yöntem ile veri seti k parçaya bölünür. Her bir parça eğitim ve test için kullanılarak ortalama sonuçlar elde edilir. Örnek olarak k=10 yapıldığında veri seti 10 eşit parçaya bölünür ve bu parçalardan 9 tanesi eğitim, 1 tanesi test için kullanılır. Her bir parça sırasıyla eğitim ve test için kullanılır. Veri seti 10 kez eğitilmiş ve 10 kez test edilmiş olur. Her bir katın ortalaması alınarak veri setinin eğitim ve test sonuçları elde edilir. Bu sayede tüm veri seti eğitim ve test için kullanılmış olur.

2.2.5. Vektörleştirme

Kelimeleri sayılara dönüştürme işlemi vektörleştirme olarak adlandırılır. Kelime Gömme veya Kelime Vektörleştirme, bir sözlük aracılığı ile kelimeleri gerçek sayılardan oluşan ve kelime tahminlerini, kelime benzerliklerini elde etmeye yarayan uygun vektörlere haritalama işlemidir.

2.2.6. Metin ve Kelime Temsil Yöntemleri

Metin madenciliği çalışmalarında kullanılmak üzere tasarlanmış pek çok metin temsil (text representation) ve kelime temsil (word embedding) yöntemi mevcuttur. Bu yöntemlerden tez çalışmasında kullanılan 4 tanesi aşağıda açıklanmıştır:

2.2.6.1. *Kelime Torbası (Bag of Words)*

Kelime Torbası (Bag of Words), doğal dil işleme ve bilgi almada kullanılan basitleştirilmiş metin veya verileri temsil ederek özelliklerin çıkarılması işlemidir. Bu

modelde, bir metin veya belge, kendi içerisinde yer alan kelimelerin çantası olarak temsil edilir. Bu nedenle, duygu analizinde Kelime Torbası modeli kullanmak, faydalı kelimelerin bir listesini oluşturmaktır [24].

2.2.6.2. *Word2Vec*

Word2vec, büyük bir veri kümesinden sözcük yerleştirmeyi inceleyen sinir ağı tabanlı bir modeldir. Yüksek boyutlu bir uzayda her kelime için bir vektör üretir. Word2Vec, vektör temsili oluşturmak için iki mimari içerir: sürekli kelime torbası (continuous bag of words) ve skip-gram. Sürekli kelime torbası modeli, kelimeleri çevreleyen bağlamlarını kullanarak tahmin eder. Skip-gram modeli, mevcut kelimeyi çevreleyen bağlamı tahmin etmek için kullanır [25].

2.2.6.3. *Tf-Idf*

TF-IDF (Terim Frekansı-Ters Belge Frekansı) (Term Frequency-Inverse Document Frequency), bir terimin frekansını (TF) ve ayrıca ters belge frekansını (IDF) ağırlıklandıran bir bilgi alma tekniğidir. Her kelime veya terimin kendi TF ve IDF puanı vardır. Bir terimin TF ve IDF çarpım puanları, o terimin TF*IDF ağırlığını belirler. Basitçe anlatmak gerekirse, bir terimin TF*IDF puanı (ağırlık değeri) ne kadar yüksekse terim o kadar nadirdir, bu değer ne kadar düşük ise bu terim o kadar yaygındır. Bir kelimenin Terim Frekansı değeri bir kelimenin frekansının, Ters Belge Frekansı değeri ise o terimin sözlük içerisinde ne kadar önemli olduğunun ölçüsüdür [24].

2.2.6.4. *FastText*

Son zamanlarda Facebook Research, kelime temsillerini öğrenmek ve metin sınıflandırması yapmak için hızlı ve etkili bir yöntem olan FastText'i açık kaynak olarak yayınlamıştır. FastText'in temel amacı, kelime temsillerini öğrenmek yerine kelimelerin içyapısını dikkate almaktır. Bu durum, farklı morfolojik kelimelerin temsillerinin bağımsız olarak öğrenilebilme olanağını sağladığından morfolojik olarak zengin diller için oldukça faydalıdır.

FastText, giriş metninin üzerine bir pencere açarak tüm bağlamdan merkez kelimeyi veya merkezdeki tüm bağlam kelimelerini tek bir skip-gram şeklinde öğrenerek çalışır. Öğrenme, iki katman ağırlık ve üç katman nöron içeren bir sinir ağında bir dizi güncelleme olarak görülebilir; burada iki dış katmanın her birinde kelime dağarcığındaki her bir kelime için bir nöron bulunur ve orta katmanda gömme alanının boyutları kadar nöron bulunur. Bu yaklaşım Word2Vec'e çok benzer. Ancak, Word2Vec'in aksine,

FastText karakter n-gramları olarak da adlandırılabilecek kelimelerin alt bölümleri için de vektörleri öğrenebilir. Bu durum, sevgi, sevilen ve sevgili gibi sözcüklerin farklı bağlamlarda karşımıza çıkabilecek olsalar bile benzer vektör temsillerine sahip olmasını sağlar. Bu özellik, yoğun çekimli dillerde öğrenmeyi geliştirir [26].

2.3. KULLANILAN PROGRAMLAMA DİLLERİ VE PLATFORMLAR

Bu tez çalışmasında 3 farklı programlama dili ile sonuçlar elde edilmiştir. Kullanılan programlama dillerinin sürüm bilgileri Çizelge 2.1'de görülebilir.

Çizelge 2.1. Yazılım sürüm bilgileri.

Program	Sürüm
Python	3.8.11
Matlab	2020a
R	4.0.5

2.3.1. Python

Python, okunması ve anlaşılması kolay, güçlü bir programlama dilidir. Diğer birçok programlama dilinde ortak olan özelliklerin çoğunu gösterir ve gerçek dünya uygulamaları için kullanışlıdır. Kullanım kolaylığı, hızlı öğrenme eğrisi ve veri bilimi ve makine öğrenimi için çok sayıda yüksek kaliteli paketi sayesinde şu anda dünyanın en hızlı büyüyen programlama dilidir [27]. Web geliştirme, veri bilimi, yazılım prototipleri oluşturma vb. alanlarda kullanılır. Bu tez çalışmasında Python programlama dili için Anaconda Navigator üzerinden Jupyter Notebook kullanılmıştır.

2.3.2. Matlab

MATLAB, pozitif bilim ve mühendislik hesaplamaları için sıklıkla kullanılan bir bilgisayar programıdır. MathWorks firması tarafından geliştirilen MATLAB aynı zamanda bir programlama dilidir. İngilizce "Matrix Laboratory" kelimelerinin birleşmesiyle oluşan MATLAB, adından da anlaşılacağı gibi matris tabanlı bir çalışma sistemine sahiptir. Bu tez çalışmasında Matlab programlama dili için MATLAB R2020a programı kullanılmıştır.

2.3.3. R

R, istatistiksel hesaplama, veri analitiği ve bilimsel araştırmalarda yaygın olarak kullanılan bir programlama dili ve ortamıdır. İstatistikçiler, veri analistleri, araştırmacılar ve pazarlamacılar tarafından verileri almak, temizlemek, analiz etmek, görselleştirmek ve sunmak için kullanılan en popüler dillerden biridir. Dışavurumcu sözdizimi ve kullanımı kolay ara yüzü nedeniyle son yıllarda popülaritesi gitgide artmıştır. Bu tez çalışmasında R programlama dili için RStudio programı kullanılmıştır.

3. METOT

3.1. KULLANILAN VERİ SETİ

Bu tez çalışmasında kullanılan veri seti, IMDB (Internet Movie Database) (İnternet Film Veri Tabanı) internet sayfasında yer alan film yorumlarından oluşmaktadır. Toplam 50.000 adet yorumdan oluşan veri setinde 25.000 adet pozitif, 25.000 adet negatif olarak etiketlenmiş yorum bulunmaktadır [28].

3.2. VERİ SETİNİ KULLANIMA HAZIR HALE GETİRME

IMDB internet sayfasında yer alan film yorumlarından oluşan veri setinin analize hazır hale getirilebilmesi için öncelikli olarak 50.000 adet yorumun tek bir dosyada toplanması gerekmektedir. Orijinal veri seti 25.000 adet pozitif ve 25.000 adet negatif olmak üzere toplam 50.000 adet bireysel .txt dosyasından oluşmaktadır. Tüm bu yorumlar MATLAB programı kullanılarak yazılan kod sayesinde tek bir dosya haline getirilmiştir. Veri setinin son hali 50.000 satır ve 2 sütundan oluşmakta olup ilk sütun film yorumunu, ikinci sütun ise yorumun etiketini (pozitiflik veya negatiflik) belirtmektedir.

3.3. VERİ İÇE AKTARIMI

Veri seti tek bir dosya haline getirildikten sonra Jupyter Notebook, MATLAB R2020a ve RStudio programlarına verinin aktarılması yani analizlerin yapılabilmesi için verinin bu programlara tanıtılması gerekmektedir. Bu işlem için Jupyter Notebook'ta (Python dilinde) "pandas" kütüphanesi kullanılarak "pd.read_csv" fonksiyonu yardımı ile veri içeri aktarılmıştır. MATLAB R2020a programında (Matlab dilinde) "readtable" fonksiyonu ile veri seti tablo olarak programa tanıtılmıştır. RStudio programında (R dilinde) "read.delim" fonksiyonu kullanılarak veri içeri aktarılmış, veri seti programa tanıtılmıştır.

3.4. VERİ TEMİZLEME

Veri seti programlara tanıtılmış olsa bile şu anki haliyle analize tam olarak hazır değildir. Duygu analizinin verimli bir şekilde yapılabilmesi için veri setinin bir takım temizleme aşamalarından geçmesi gerekmektedir.

3.4.1. Duygu Analizinde Veri Temizleme ile İlgili Bilinen Kısıtlar

Günümüzde, kişilerin duygularını aktardığı metinlerde kelimelerin yanı sıra diğer semboller de kullanılabilmektedir. Bu semboller arasında duygu durumunu en çok belli eden semboller "emoji" denilen belirli yüz ifadelerinin sergilendiği sembollerdir. Bu tez çalışmasında veri temizleme aşamalarında harf dışındaki bütün sembollerin silinmesi tercih edildiği için, bu tarz duygu bildiren sembollerden dolayı belirli sınıflandırma kayıpları yaşanabilir. Bu tez genelinde kelime bazlı duygu analizi öncelikli olarak tercih edilmiştir. Başka çalışmalarda harf dışındaki sembollerin de entegrasyonu sağlanarak farklı analizler yapılabilir. Ayrıca, insanların yaptıkları yorumlar içerisinde eksik harfli veya hatalı yazılan kelimeler de mevcut olabilmektedir. Bu durumda bir kelime birden fazla şekilde ifade edilebileceği için bazı sınıflandırma kayıpları yaşanabilir.

3.4.2. Harf Dışındaki Tüm Sembollerin Metinden Çıkarılması

Duygu analizi yapılırken sonucu olumsuz yönde etkileyebilecek karakterlerin metinden temizlenmesi gerekir. Rakamlar, noktalama işaretleri ve diğer tüm semboller bir duygu ifade etmez. Bu nedenle analiz sonuçlarını olumsuz etkilememeleri için metinden temizlenmeleri gerekir. Bunun için Python dilinde "re.sub" fonksiyonu kullanılarak gerekli temizleme işlemi yapılmıştır. Matlab dilinde "regexprep" fonksiyonu kullanılarak gerekli temizleme işlemi yapılmıştır. R dilinde "tm_map" fonksiyonu içerisine gerekli parametreler girilerek gerekli temizleme işlemi yapılmıştır. Bu işlemlerin sonucunda harf dışındaki tüm karakterler yorumlardan silinmiştir.

3.4.3. Tüm Harflerin Küçük Harfe Çevrilmesi

Bir önceki aşamada harf dışındaki tüm sembollerden arındırılan veri setinin bir düzenleme aşamasından daha geçmesi gerekmektedir. Veri setinde yer alan tüm harflerin küçük harfe dönüştürülmesi, aynı kelimenin büyük veya küçük harflerle veri setinde bulunması durumunda (bir yorumda "Duygu", bir yorumda "duygu" şeklinde yazılan iki kelime aslında aynı etkiye sahip olmalıyken bu şekilde bırakılırsa sonuç farklı çıkabilir)

duygu analizi sonucunu etkileyebileceğinden bir bütünlük, tutarlılık sağlanması açısından oldukça büyük önem arz eder. Bunun için Python ve Matlab dillerinde “lower”, R dilinde ise “tolower” isimli fonksiyon kullanılarak veri setindeki tüm harfler küçük harfe çevrilmiştir.

3.4.4. Etkisiz Kelimelerin (Stopwords) Çıkarılması

Veri temizleme aşamalarından bir tanesi de etkisiz kelimelerin çıkarılmasıdır. Herhangi bir duygu ifade etmeyen tüm karakter, kelime ve ifadelerin veri setinden temizlenmesi gerekmektedir. Burada dikkat edilmesi gereken nokta, İngilizce dilindeki “not” kelimesidir. Bir yorumda geçen “good” ifadesi olumlu bir duyguyu ifade ederken “not good” ifadesi olumsuz bir duyguyu ifade eder. Yani İngilizce dilindeki etkisiz kelimeler veri setinden temizlenirken bu duruma dikkat edilmeli ve kod yazılırken “not” ifadesinin veri setinden çıkarılmamasına dikkat edilmesi gerekmektedir. Bu işlemin gerçekleştirilebilmesi için Python “remove_stopwords” fonksiyonu, Matlab dilinde “removeStopWords” fonksiyonu, R dilinde “removeWords” ve “stopwords” fonksiyonları kullanılmıştır. Bu işlemlerin sonucunda veri seti etkisiz kelimelerden arındırılmıştır. Etkisiz kelimelerin (stopwords) çıkarılması aşamasında kullanılan etkisiz kelimelerin listesi (stopwords) Çizelge 3.1’de listelenmiştir.

Çizelge 3.1. Etkisiz Kelime (Stopwords) Listesi.

Etkisiz Kelime (Stopwords) Listesi					
i	me	my	myself	we	our
you'd	your	yours	yourself	yourselves	he
hers	herself	it	it's	its	itself
which	who	whom	this	that	that'll
were	be	been	being	have	has
a	an	the	and	but	if
at	by	for	with	about	against
above	below	to	from	up	down
again	further	then	once	here	there
both'	each	few	more	most	other
same	so	than	too	very	s
should	should've	now	d	ll	m
aren't	couldn'	couldn't	didn'	didn't	doesn'
haven't	isn'	isn't	ma'	mightn'	mightn't
shouldn'	shouldn't	wasn'	wasn't	weren'	weren't
ours	ourselves	you	you're	you've	you'll
him	his	himself	she	she's	her
they	them	their	theirs	themselves	what
these	those	am	is	are	was
had	having	do	does	did	doing
or	because	as	until	while	of
between	into	through	during	before	after
in	out	on	off	over	under
when	where	why	how	all	any
some	such	no	nor	only	own
t	can	will	just	don'	don't
o	re	ve	y	ain'	aren'
doesn't	hadn'	hadn't	hasn'	hasn't	haven'
mustn'	mustn't	needn'	needn't	shan'	shan't
won'	won't	wouldn'	wouldn't		

3.4.5. Kelimelerin Kök Hallerine Getirilmesi (Stemming)

Veri setinden etkisiz kelimelerin de çıkarılmasının ardından son bir temizleme işlemine daha ihtiyaç duyulmaktadır. Bu da "stemming" denilen, kelimelerin sadece kök halinin kalacak şekilde aldığı diğer eklerden arındırılarak temizlenmesi işlemidir. Örnek olarak, "oynuyorum", "oynadım" gibi kelimeler "oyna" kök haline getirilir. Bu işlem için Python dilinde "nltk" kütüphanesi içerisinde yer alan "stem" fonksiyonu, Matlab dilinde "Text Analytics Toolbox" içerisinde yer alan "normalizeWords" fonksiyonu, R dilinde "tm" kütüphanesi içerisinde yer alan "stemDocument" fonksiyonu kullanılmış, veri setindeki tüm kelimeler kök hallerine dönüştürülmüştür.

3.5. VERİ SETİNİN EĞİTİM ve TEST SETLERİ OLARAK AYRILMASI

Veri seti temizlendikten ve analize hazır hale getirildikten sonra yapılması gereken bir sonraki adım veri setinin eğitim ve test setleri olarak ayrılmasıdır. Bu aşamada farklı yaklaşımlar izlenebilir. Bu tez çalışmasında k katlı çapraz doğrulama yöntemi kullanılmıştır. Veri seti 25.000 pozitif ve 25.000 negatif yorumdan oluştuğu için dengeli dağılan bir veri setidir. Veri setinin orijinal halinde ilk 25.000 satır pozitif, kalan 25.000 satır negatif yorumlardan oluşmaktadır. Veri setinden parçalar alarak eğitim ve test setleri şeklinde ayırmadan önce veri seti rastgele karıştırılmış ve ardından bu karıştırılan veri seti veri setini ayırma (hold-out) ve çapraz doğrulama kriterlerini karşılamak için k=5 olacak şekilde 5 parçaya ayrılarak %80 eğitim, %20 test olacak şekilde işlenmiştir. Bu yöntemin avantajı tüm veri setinin hem eğitim hem de test için kullanılması ve analiz sonuçlarının her bir kat için ortalamasının alınarak optimum şekilde elde edilebilmesidir. Kullanılan üç programlama dilinde de k katlı çapraz doğrulama yapılabilmesi için gerekli kodlar yazılmış, gerekli döngü yapısı kurularak her bir k değeri için eğitim ve test setleri oluşturulmuştur. Esas analizler, literatürde de en çok kullanılan k = 5 (%80 - %20) üzerinden yapılmakla birlikte, aynı analizler k = 2 (%50 - %50), k = 3 (%67 - %33), k = 4 (%75 - %25) olacak şekilde de Python dilinde uygulanmış, en yüksek doğruluk sonuçlarını veren Lojistik Regresyon sınıflandırma yöntemi ile elde edilen sonuçlar farkları göstermek adına raporlanmıştır.

3.6. METİN TEMSİL YÖNTEMLERİNİN UYGULANMASI

3.6.1. Kelime Torbası (Bag of Words)

Kelime Torbası modelinin oluşturulabilmesi için 3 programlama dilinde de bulunan "bag of words" fonksiyonları kullanılmıştır. Burada dikkat edilmesi gereken nokta Kelime Torbası modeli oluşturulduktan sonra seçilecek kelime sayısıdır. Kelime Torbası modeli oluştuğunda elimizdeki veri setinde yer alan tüm kelimelerin ve bu kelimelerin veri setinde kaç kez geçtiği bilgisi elde edilmiş olunur. Kelime Torbası metodunun uygulanması için gerekli sadece bir parametre vardır. Bu parametre en çok geçen ilk kaç kelimenin sayısının (max_count) tutulacağıdır. Bu tezde uygulan Kelime Torbası metodu için Çizelge 3.2'de belirtilen değer seçilmiştir.

Çizelge 3.2. Kelime Torbası (Bag of Words) parametreleri.

Parametre	Değer
Max_count	1500

3 programlama dili için de aynı değer alınarak sınıflandırma algoritmaları gerçeklenmiştir. Bu aşamada kullanılan sınıflandırma algoritmaları ve simülasyonların gerçekleştirildiği bilgisayarın özelliklerine bağlı olarak farklı parametre değerleri seçilebilir. Farklı değerlerin seçilmesi sırasında, değerin düşürülmesi, anlamlı olabilecek kelimeleri dışarıda bırakabilirken, değerin yükseltilmesi, sisteme duygu ifade etmeyebilecek ve sık kullanılmayan kelimeleri ekleyerek sınıflandırma analizinin doğruluğunu düşürebilecektir. Bu tezde kullanılan 1500 değeri, farklı parametrelerle yapılan denemeler sonucunda, ilgili veri setine uygun olarak seçilmiştir.

3.6.2. Word2Vec

Word2Vec modelinin oluşturulabilmesi için 3 programlama dilinde de bulunan "word2vec" fonksiyonları kullanılmıştır. Burada dikkat edilmesi gereken nokta model oluşturulurken "word2vec" fonksiyonunun içine girilecek boyut (dimension) parametresinin değeridir. Bu tez çalışmasında bu değer 300 olarak belirlenmiş ve tüm programlarda model bu şekilde oluşturulmuştur. Bu yöntem ile veri setindeki her bir kelime 300'lük bir vektör ile ifade edilir ve duygu analizi her bir yorumdaki tüm kelimelerin vektör değerleri karşılaştırılarak yapılır. "word2vec" fonksiyonu kullanılırken kullanılan parametreler Çizelge 3.3'te gösterilmiştir.

Çizelge 3.3. Word2Vec parametreleri.

Parametre	Değer
Vector_size	300
Window	5
Min_count	1
workers	4

3.6.3. Tf-Idf

TF-IDF modelinin oluşturulabilmesi için 3 programlama dilinde de bulunan "tf-idf" fonksiyonları kullanılmıştır. TF-IDF metodunun uygulanması için gerekli sadece bir parametre vardır. Bu parametre en çok geçen ilk kaç kelimenin sayısının (max_feature) tutulacağı bilgisini içerir. Bu tezde uygulan TF-IDF metodu için Çizelge 3.4'te belirtilen değer seçilmiştir.

Çizelge 3.4. TF-IDF parametreleri.

Parametre	Değer
Max_feature	1500

Kelime Torbası modeliyle benzerlik göstermesi ve karşılaştırmanın daha verimli yapılabilmesi için bu yöntemde de "max_feature" değeri 1500 olarak alınmış ve 3 programlama dili için sınıflandırma algoritmaları gerçeklenmiştir.

3.6.4. FastText

FastText modelinin oluşturulabilmesi için 3 programlama dilinde de bulunan "fasttext" kütüphaneleri kullanılmış ve hazır olarak kullanılabilen fasttext modeli içe aktarılarak (import edilerek) veri setinin fastText modeli oluşturulmuştur [29]. Word2Vec modeli ile benzerlik göstermesi ve daha sağlıklı bir karşılaştırma yapılabilmesi için bu yöntemde de vektör sayısı 300 olarak belirlenmiştir.

3.6.5. Metin Temsil Yöntemlerinin Ön Değerlendirmesi

Yukarıda bahsedilen bu tez çalışmasında kullanılan metin ve kelime temsil yöntemlerinin birbirlerine göre avantaj ve dezavantajlarının [30] kısaca açıklandığı tablo Çizelge 3.5'te görülebilir.

Çizelge 3.5. Metin ve Kelime Temsil Yöntemlerinin Karşılaştırması.

Yöntem	Avantaj	Dezavantaj
Kelime Torbası (Bag of Words)	Basit uygulama yöntemi	Büyük dosyalarda karmaşıklık, Semantik bağlantıları çözememe
TF-IDF		
Word2Vec	Her boyutta dosyada uyumluluk, Semantik bağlantı kurma	Karmaşık uygulama
FastText		

3.7. SINIFLANDIRMA YÖNTEMLERİNİN UYGULANMASI

3.7.1. Naif Bayes (Naive Bayes)

Naif Bayes sınıflandırma algoritması, Python dilinde "GaussianNB" fonksiyonu, Matlab dilinde "fitcnb" fonksiyonu, R dilinde "naiveBayes" fonksiyonu kullanılarak gerçeklenmiştir. 5-kat çapraz doğrulama yöntemi kullanıldığı için her bir kat ile algoritma çalıştırılmış, her bir kat ayrı ayrı eğitilmiş ve test edilmiştir. 5 kat için elde edilen hata matrislerinin ortalaması alınarak Naif Bayes yöntemi için ortalama hata matrisi kaydedilmiştir.

3.7.2. Destek Vektör Makinesi (Support Vector Machine)

Destek Vektör Makinesi sınıflandırma algoritması, Python dilinde "SVC" fonksiyonu, Matlab dilinde "fitcsvm" fonksiyonu, R dilinde "svm" fonksiyonu kullanılarak gerçeklenmiştir. 5-kat çapraz doğrulama yöntemi kullanıldığı için her bir kat ile algoritma çalıştırılmış, her bir kat ayrı ayrı eğitilmiş ve test edilmiştir. 5 kat için elde edilen hata matrislerinin ortalaması alınarak Destek Vektör Makinesi yöntemi için ortalama hata matrisi kaydedilmiştir. 3 dilde de Kernel fonksiyonu olarak "linear" kernel fonksiyonu kullanılmıştır. Linear Kernel, verilerin tek bir çizgi ile ayrılabildiği (doğrusal olarak) durumlarda kullanılır. En çok kullanılan kernel'lerden biridir. Çoğunlukla, belirli bir veri kümesinde çok sayıda özellik olduğunda kullanılır [31]. Diğer kernel fonksiyonlarının (rbf ve polynomial) etkilerini görmek için, Python dilinde k = 5 (%80 - %20) çapraz doğrulama kullanılarak, word2vec metin temsil yöntemi kullanılarak, diğer iki kernel fonksiyonu da uygulanmış ve sonuçlar 4. Bölümde sunulmuştur.

3.7.3. Lojistik Regresyon (Logistic Regression)

Lojistik Regresyon sınıflandırma algoritması, Python dilinde "LogisticRegression" fonksiyonu, Matlab dilinde "fitglm" fonksiyonu, R dilinde "glm" fonksiyonu kullanılarak gerçeklenmiştir. 5-kat çapraz doğrulama yöntemi kullanıldığı için her bir kat ile algoritma çalıştırılmış, her bir kat ayrı ayrı eğitilmiş ve test edilmiştir. 5 kat için elde edilen hata matrislerinin ortalaması alınarak Lojistik Regresyon yöntemi için ortalama hata matrisi kaydedilmiştir.

3.7.4. Karar Ağacı (Decision Tree)

Karar Ağacı sınıflandırma algoritması, Python dilinde "DecisionTreeClassifier" fonksiyonu, Matlab dilinde "fitctree" fonksiyonu, R dilinde "rpart" fonksiyonu kullanılarak gerçeklenmiştir. 5-kat çapraz doğrulama yöntemi kullanıldığı için her bir kat ile algoritma çalıştırılmış, her bir kat ayrı ayrı eğitilmiş ve test edilmiştir. 5 kat için elde edilen hata matrislerinin ortalaması alınarak Karar Ağacı yöntemi için ortalama hata matrisi kaydedilmiştir.

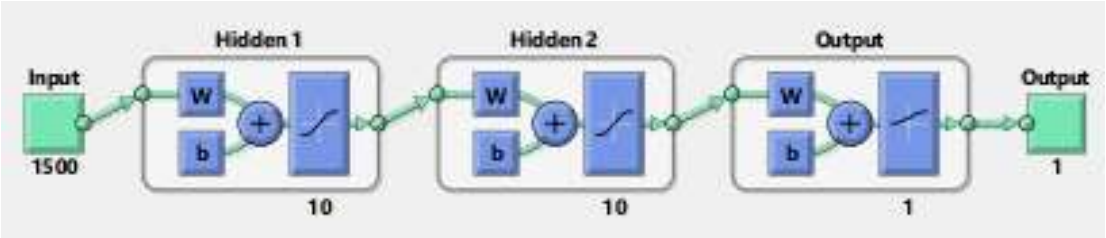
3.7.5. Rastgele Orman (Random Forest)

Rastgele Orman sınıflandırma algoritması, Python dilinde "RandomForestClassifier" fonksiyonu, Matlab dilinde "fitcensemble" fonksiyonu, R dilinde "randomForest" fonksiyonu kullanılarak gerçeklenmiştir. 5-kat çapraz doğrulama yöntemi kullanıldığı için her bir kat ile algoritma çalıştırılmış, her bir kat ayrı ayrı eğitilmiş ve test edilmiştir. 5 kat için elde edilen hata matrislerinin ortalaması alınarak Rastgele Orman yöntemi için ortalama hata matrisi kaydedilmiştir.

3.7.6. Yapay Sinir Ağları (Artificial Neural Network)

Bu tez çalışmasında kullanılan diğer sınıflandırma yöntemlerinden farklı olarak Yapay Sinir Ağları sınıflandırma algoritmasının çalıştırılabilmesi için öncelikli olarak bir yapay sinir ağı yapısının oluşturulması gerekmektedir. Doğrultulmuş Doğrusal Birim (Rectified Linear Unit – ReLU) yapay sinir ağlarında aktivasyon fonksiyonu olarak yaygınca kullanılır. Bu birimler, derin öğrenme devrimindeki birkaç kilometre taşından biridir. Basit ve güçlüdür, ileri beslemeli ağların performansını büyük ölçüde artırır. Bu nedenle, birçok başarılı mimaride yaygın olarak kullanılmaktadır [32]. Bu tezde kullanılan sinir ağı yapısında bir adet giriş katmanı, iki adet boyutu 10 olan gizli katman ve bir adet de çıkış katmanı kullanılmıştır. Çıkış katmanı dışındaki katmanların aktivasyon

fonksiyonları ReLU, çıkış katmanının aktivasyon fonksiyonu da ikili sınıflandırmaya uygun olarak sigmoid fonksiyonu olarak uygulanmıştır. Oluşturulan sinir ağının yapısı Şekil 3.1'de görülebilir.



Şekil 3.1. Yapay Sinir Ağı Yapısı.

5-kat çapraz doğrulama yöntemi kullanıldığı için her bir kat ile algoritma çalıştırılmış, her bir kat ayrı ayrı eğitilmiş ve test edilmiştir. Her bir k değeri için döngü başa döndüğünde sinir ağır tekrar oluşturularak herhangi bir tutarsızlık oluşmaması sağlanmıştır. 5 kat için elde edilen hata matrislerinin ortalaması alınarak Yapay Sinir Ağları yöntemi için ortalama hata matrisi kaydedilmiştir.

3.8. KARŞILAŞTIRMA PARAMETRELERİ

Bu tez çalışmasında 4 farklı metin temsil yöntemi, 6 farklı sınıflandırma yöntemi ve 3 farklı programlama dili kullanılarak elde edilen sonuçların sağlıklı şekilde karşılaştırılabilmesi ve hangi yöntemin nasıl sonuç verdiğinin verimli bir şekilde anlaşılabilmesi için öncelikli olarak her bir sınıflandırma algoritmasından hata matrisleri elde edilmiştir. Hata matrisinde yer alan değerler kullanılarak doğruluk (accuracy), kesinlik derecesi (precision), anımsama (recall) ve f1-değeri (f1-score) değerleri hesaplanmış, her bir yöntemin ne kadar sürdüğü sonuç tablosuna eklenmiştir. Ayrıca, HWINFO isimli program yardımıyla bellek ve işlemci kullanımları hesaplanmıştır [33].

3.9. DONANIM BİLGİSİ

Bu tez çalışmasında tüm simülasyon sonuçları Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz, 12GB RAM'e sahip bir laptop kullanılarak elde edilmiştir. İşletim sistemi olarak Windows kurulu olan bilgisayarda kapasitesi 480GB, okuma hızı 540 MB/sn, yazma hızı 500 MB/sn, bağlantı tipi SATA 3.0, boyutu 2.5" olan bir SSD (Solid State Drive – Katı Hal Sürücüsü) yer almaktadır.

4. PERFORMANS KRİTERLERİ VE DENEY SONUÇLARI

4.1. KULLANILAN PROGRAMLAMA DİLLERİNİN PERFORMANSLARININ VERİ TEMİZLEME ADIMLARI İÇİN KARŞILAŞTIRMASI

Bu başlık altında Python, Matlab ve R programlama dillerinde uygulanan veri temizleme adımlarının sonucunda, kullanılan veri setinden örnek bir verinin geçtiği aşamalar gösterilmiştir. Bu aşamalar sırasında geçen süre, kullanılan işlemci ve bellek değerleri raporlanmıştır. Çizelge 4.1'de tüm programlarda metnin geçtiği temizleme aşamaları gösterilmiştir.

Çizelge 4.1. Veri Temizleme Adımlarının Örnek Gösterimi.

Ham Metin		Before Dogma 95: when Lars used movies as art, not just a story. A beautiful painting about love and death. This is one of my favorite movies of all time. The color... The music... Just perfect.
Harf Dışındaki Tüm Sembollerin Metinden Çıkarılması	Python	Before Dogma when Lars used movies as art not just a story A beautiful painting about love and death This is one of my favorite movies of all time The color The music Just perfect
	Matlab	
	R	
Tüm Harflerin Küçük Harfe Çevrilmesi	Python	before dogma when lars used movies as art not just a story a beautiful painting about love and death this is one of my favorite movies of all time the color the music just perfect
	Matlab	
	R	
Etkisiz Kelimelerin (Stopwords) Çıkarılması	Python	dogma lars movies art not just story beautiful painting love death favorite movies time color music just perfect
	Matlab	
	R	
Kelimelerin Kök Hallerine Getirilmesi (Stemming)	Python	dogma lar use movi art not stori beauti paint love death one favorit movi time color music perfect
	Matlab	
	R	

Çizelge 4.2'de kullanılan üç programlama dili için veri temizleme sırasındaki performans karşılaştırmaları yer almaktadır. Bellek ve işlemci kullanımı açısından ölçümler yapılırken bazı adımların çok kısa sürmesi ve verimli bir şekilde bellek ve işlemci kullanım bilgileri tespit edilemediği için tüm süreç boyunca kullanılan bellek ve işlemci bilgileri verilirken süre açısından her bir adımın işlem süresi tabloya eklenmiştir.

Çizelge 4.2. Veri Temizleme Adımlarının Performans Karşılaştırması.

Veri Madenciliği Programı/Platform	Yapılan Önişleme Adımı	Birim İşlem Süresi (Sn)	Toplam Süre (Sn)	Bellek Kullanımı (GB)	İşlemci Kullanımı (GHz)
Python	Harf Dışındaki Tüm Sembollerin Metinden Çıkarılması	7,89	180,38	5,19	0,34
	Tüm Harflerin Küçük Harfe Çevrilmesi	0,93			
	Etkisiz Kelimelerin (Stopwords) Çıkarılması	34,3			
	Kelimelerin Kök Hallerine Getirilmesi (Stemming)	137,26			
Matlab	Harf Dışındaki Tüm Sembollerin Metinden Çıkarılması	12	31,88	5,78	0,37
	Tüm Harflerin Küçük Harfe Çevrilmesi	0,29			
	Etkisiz Kelimelerin (Stopwords) Çıkarılması	8,52			
	Kelimelerin Kök Hallerine Getirilmesi (Stemming)	11,07			
R	Harf Dışındaki Tüm Sembollerin Metinden Çıkarılması	6	87	4,35	0,33
	Tüm Harflerin Küçük Harfe Çevrilmesi	3			
	Etkisiz Kelimelerin (Stopwords) Çıkarılması	62			
	Kelimelerin Kök Hallerine Getirilmesi (Stemming)	16			

Veri temizleme adımları açısından programların performansları incelendiğinde toplam süre açısından en hızlı performansı Matlab dili göstermiştir. İşlemlerin en uzun sürede tamamlandığı dil ise Python olmuştur. Bellek ve işlemci kullanımı açısından R dili en az bellek ve işlemci kullanan dil olurken en çok bellek ve işlemci kullanan dil Matlab olmuştur. Süre farkları incelendiğinde kullanılan programlama dillerinin altyapıları ve bu matris tabanlı veriye yaklaşım tarzları nedeniyle sürelerin farklılık gösterdiği sonucu ortaya çıkmıştır.

4.2. SINIFLANDIRMA YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRMASI

Bu tez çalışmasında 4 farklı metin temsil yöntemi ile 6 farklı sınıflandırma yöntemi kullanılarak gerçeklenen duygu analizleri sonucunda 5-kat çapraz doğrulama yöntemi kullanılarak hata matrisleri elde edilmiş, bu hata matrisleri yardımıyla karşılaştırmaların yapılacağı değerler hesaplanmıştır.

Hata matrisi, 2x2 boyutunda bir matris olup içerisinde yer alan değerler aşağıdaki gibidir:

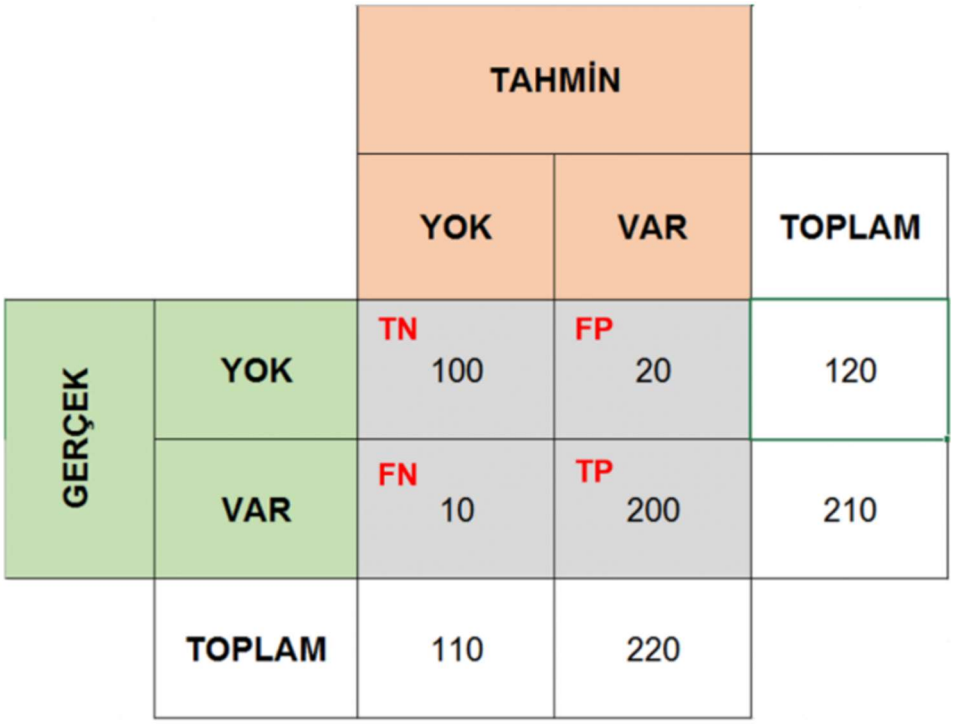
Gerçek Pozitif (True Positive): Doğru şekilde pozitif olarak sınıflandırılan yorum sayısı.

Gerçek Negatif (True Negative): Doğru şekilde negatif olarak sınıflandırılan yorum sayısı.

Yanlış Pozitif (False Positive): Yanlış şekilde pozitif olarak sınıflandırılan yorum sayısı.

Yanlış Negatif (False Negative): Yanlış şekilde negatif olarak sınıflandırılan yorum sayısı.[14]

Hata matrisinin örnek bir görünümü Şekil 4.1'de görülebilir.



Şekil 4.1. Hata Matrisi Yapısı.

4.2.1. Doğruluk (Accuracy)

Doğruluk değerinin hesaplanabilmesi için aşağıdaki formül kullanılmıştır.

$$Doğruluk = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

4.2.2. Kesinlik Derecesi (Precision)

Kesinlik Derecesi değerinin hesaplanabilmesi için aşağıdaki formül kullanılmıştır.

$$Kesinlik\ Derecesi = \frac{TP}{TP + FP} \tag{4.2}$$

4.2.3. Anımsama (Recall)

Anımsama değerinin hesaplanabilmesi için aşağıdaki formül kullanılmıştır.

$$Anımsama = \frac{TP}{TP + FN} \tag{4.3}$$

4.2.4. F1-Değeri (F1-Score)

F1-Değeri'nin hesaplanabilmesi için aşağıdaki formül kullanılmıştır.

$$F1\ Değeri = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

4.2.5. Çalışma Süreleri

Her bir sınıflandırma yöntemi için Python dilinde "datetime" fonksiyonu, Matlab dilinde "clock" fonksiyonu, R dilinde "tic" ve "toc" fonksiyonları kullanılarak çalışma süreleri kaydedilmiş ve sonuç tablosuna eklenmiştir.

4.2.6. Bellek ve İşlemci Kullanımı

Sınıflandırma algoritmalarının performansları hata matrisinden elde edilen doğruluk, kesinlik derecesi, anımsama ve f1-değeri kullanılarak karşılaştırılırken elde edilen sonuçlara ek olarak simülasyonların çalıştırıldığı bilgisayarda simülasyon süresi boyunca kullandıkları bellek ve işlemci değerleri de sonuç tablosuna eklenmiştir.

4.3. PYTHON İLE ELDE EDİLEN SONUÇLAR

Bu bölümde Python programlama dili ile elde edilen sonuçlar yer almaktadır. Doğruluk, kesinlik derecesi, anımsama ve f1-değeri için en iyi sonucu veren iki yöntem Lojistik

Regresyon ve Destek Vektör Makinesi olmuştur. Çalışma süresi açısından en hızlı yöntem word2vec ve fastText ile Naif Bayes, en yavaş yöntem ise kelime torbası ile Destek Vektör Makinesi olmuştur. Ortalama bellek kullanımı açısından en fazla bellek kullanan yöntem fastText ile Lojistik Regresyon, en az bellek kullanan yöntem ise word2vec ile Destek Vektör Makinesi olmuştur. Ortalama işlemci kullanımı açısından en fazla işlemci kullanan yöntem TF-IDF ile Lojistik Regresyon, en az işlemci kullanan yöntem word2vec ile Rastgele Orman olmuştur. Python ile elde edilen tüm sonuçlar Çizelge 4.3'de görülebilir.

Çizelge 4.3. Python ile Elde Edilen Tüm Sonuçlar (k = 5, %80 - %20).

		Doğruluk (Accuracy)	Kesinlik Derecesi (Precision)	Anımsama (Recall)	F1-Değeri (F1-Score)	Süre (Sn)	Bellek Kullanımı (GB)	İşlemci Kullanımı (GHz)
Naif Bayes (Naive Bayes)	Kelime Torbası (Bag of Words)	0,76	0,82	0,67	0,74	11,34	7,93	0,37
	TF-IDF	0,81	0,81	0,81	0,81	11,39	8,33	0,44
	Word2vec	0,77	0,77	0,78	0,77	2,48	6,35	0,35
	FastText	0,70	0,70	0,71	0,70	2,52	9,82	0,40
Destek Vektör Makinesi (Support Vector Machine)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	87968,68	7,76	0,34
	TF-IDF	0,87	0,86	0,88	0,87	4342,28	7,68	0,34
	Word2vec	0,87	0,87	0,88	0,87	1400,32	6,07	0,34
	FastText	0,84	0,83	0,85	0,84	1707,09	8,08	0,35
Lojistik Regresyon (Logistic Regression)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	38,64	8,11	1,27
	TF-IDF	0,87	0,87	0,88	0,88	33,08	8,65	1,35
	Word2vec	0,87	0,87	0,88	0,87	9,7	6,55	1,27
	FastText	0,83	0,82	0,84	0,83	8,74	10,08	1,26
Rastgele Orman (Random Forest)	Kelime Torbası (Bag of Words)	0,79	0,82	0,73	0,78	35,52	7,78	0,33
	TF-IDF	0,79	0,82	0,74	0,78	47,54	8,22	0,35
	Word2vec	0,81	0,83	0,77	0,80	46,7	6,33	0,33
	FastText	0,73	0,77	0,68	0,72	49,38	9,87	0,34
Karar Ağacı (Decision Tree)	Kelime Torbası (Bag of Words)	0,72	0,72	0,71	0,72	158,32	7,68	0,33
	TF-IDF	0,72	0,72	0,72	0,72	217,13	8,10	0,35
	Word2vec	0,74	0,74	0,74	0,74	126,98	6,62	0,34
	FastText	0,67	0,67	0,67	0,67	128,31	9,90	0,33
Yapay Sinir Ağları (Artificial Neural Network)	Kelime Torbası (Bag of Words)	0,82	0,83	0,81	0,82	838,96	7,71	0,77
	TF-IDF	0,82	0,83	0,80	0,81	911,13	7,91	0,79
	Word2vec	0,87	0,87	0,88	0,87	578,78	7,53	0,75
	FastText	0,84	0,84	0,85	0,84	575,75	9,88	0,76

Python'da farklı k değerlerinin (k = 2, k = 3, k = 4, k = 5) sonuçları Çizelge 4.4'de raporlanmıştır.

Çizelge 4.4. 4 Farklı k değerleri için Karşılaştırmalı Sonuçlar.

		Python						
		Lojistik Regresyon (Logistic Regression)						
		Doğruluk (Accuracy)	Kesinlik Derecesi (Precision)	Anımsama (Recall)	F1-Değeri (F1-Score)	Süre (Sn)	Bellek Kullanımı (GB)	İşlemci Kullanımı (GHz)
k = 2 (%50 - %50)	Kelime Torbası (Bag of Words)	0,86	0,86	0,87	0,87	9,29	7,09	1,22
	TF-IDF	0,87	0,86	0,88	0,87	5,19	7,09	1,14
	Word2vec	0,87	0,87	0,88	0,87	2,45	5,84	1,16
	FastText	0,80	0,79	0,81	0,80	2,46	9,60	1,22
k = 3 (%67 - %33)	Kelime Torbası (Bag of Words)	0,87	0,86	0,87	0,87	16,37	8,46	1,28
	TF-IDF	0,87	0,87	0,88	0,87	11,62	7,11	1,36
	Word2vec	0,87	0,87	0,88	0,87	4,97	5,91	1,18
	FastText	0,81	0,80	0,82	0,81	3,78	9,71	1,23
k = 4 (%75 - %25)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	37,00	7,98	1,30
	TF-IDF	0,87	0,87	0,88	0,88	14,89	7,30	1,42
	Word2vec	0,87	0,87	0,88	0,87	6,84	6,12	1,23
	FastText	0,81	0,80	0,82	0,81	5,22	9,80	1,22
k = 5 (%80 - %20)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	38,64	8,11	1,27
	TF-IDF	0,87	0,87	0,88	0,88	33,08	8,65	1,35
	Word2vec	0,87	0,87	0,88	0,87	9,70	6,55	1,27
	FastText	0,83	0,82	0,84	0,83	8,74	10,08	1,26

Python'da Destek Vektör Makinesi sınıflandırma yöntemi için farklı kernel fonksiyonlarının sonuçlarını gösteren tablo Çizelge 4.5'de raporlanmıştır.

Çizelge 4.5. Destek Vektör Makinesi Farklı Kernel Fonksiyonları için Sonuçlar.

Python			
Destek Vektör Makinesi - Word2Vec			
Kernel	Linear	RBF	Polynomial
Doğruluk	0,87	0,87	0,87
Kesinlik Derecesi	0,87	0,87	0,86
Anımsama	0,88	0,88	0,89
F1-Değeri	0,87	0,88	0,87
Süre (sn)	1400,32	1264,31	1152,31
Bellek Kullanımı (GB)	6,07	8,29	9,66
İşlemci Kullanımı (GHz)	0,34	0,47	0,49

4.4. MATLAB İLE ELDE EDİLEN SONUÇLAR

Bu bölümde Matlab programlama dili ile elde edilen tüm sonuçlar yer almaktadır. Doğruluk, kesinlik derecesi, anımsama ve f1-değeri için en iyi sonucu veren iki yöntem Lojistik Regresyon ve Yapay Sinir Ağları olmuştur. Çalışma süresi açısından en hızlı yöntem word2vec ve fastText ile Naif Bayes, en yavaş yöntem ise TF-IDF ile Destek Vektör Makinesi olmuştur. Ortalama bellek kullanımı açısından en fazla bellek kullanan yöntem word2vec ile Karar Ağacı, en az bellek kullanan yöntem ise kelime torbası ile Destek Vektör Makinesi olmuştur. Ortalama işlemci kullanımı açısından en fazla işlemci kullanan yöntem kelime torbası ile Naif Bayes, en az işlemci kullanan yöntem word2vec ve fastText ile Lojistik Regresyon olmuştur. Matlab ile elde edilen tüm sonuçlar Çizelge 4.6'da görülebilir.

Çizelge 4.6. Matlab ile Elde Edilen Tüm Sonuçlar (k = 5, %80 - %20).

		Doğruluk (Accuracy)	Kesinlik Derecesi (Precision)	Anımsama (Recall)	F1-Değeri (F1-Score)	Süre (Sn)	Bellek Kullanımı (GB)	İşlemci Kullanımı (GHz)
Naif Bayes (Naive Bayes)	Kelime Torbası (Bag of Words)	0,77	0,82	0,70	0,75	23,83	10,76	1,66
	TF-IDF	0,77	0,82	0,70	0,75	25,55	9,84	1,60
	Word2vec	0,77	0,78	0,75	0,76	7,64	11,16	1,07
	FastText	0,75	0,77	0,72	0,74	7,16	10,60	1,26
Destek Vektör Makinesi (Support Vector Machine)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	67743,5	7,43	1,19
	TF-IDF	0,71	0,76	0,63	0,69	201366,48	7,60	1,19
	Word2vec	0,88	0,88	0,88	0,88	666,81	9,75	1,37
	FastText	0,85	0,85	0,86	0,86	821,83	9,45	1,38
Lojistik Regresyon (Logistic Regression)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	632,33	8,97	1,13
	TF-IDF	0,87	0,86	0,88	0,87	646,65	9,77	1,13
	Word2vec	0,88	0,88	0,88	0,88	71,91	9,92	1,00
	FastText	0,86	0,86	0,86	0,86	50,17	8,97	1,00
Rastgele Orman (Random Forest)	Kelime Torbası (Bag of Words)	0,84	0,83	0,85	0,84	667,4	9,08	1,23
	TF-IDF	0,84	0,83	0,85	0,84	668,56	7,56	1,22
	Word2vec	0,85	0,84	0,85	0,85	1675,91	9,00	1,21
	FastText	0,82	0,82	0,82	0,82	1662,02	8,49	1,22
Karar Ağacı (Decision Tree)	Kelime Torbası (Bag of Words)	0,72	0,72	0,72	0,72	166,46	10,86	1,27
	TF-IDF	0,72	0,72	0,72	0,72	171,83	10,04	1,21
	Word2vec	0,71	0,72	0,71	0,71	47,33	11,34	1,19
	FastText	0,69	0,69	0,69	0,69	50,41	9,47	1,36
Yapay Sinir Ağları (Artificial Neural Network)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	1091,88	8,19	1,22
	TF-IDF	0,87	0,87	0,87	0,87	1120,45	7,67	1,23
	Word2vec	0,85	0,84	0,85	0,85	469,58	9,07	1,21
	FastText	0,83	0,82	0,83	0,83	475,62	8,94	1,21

4.5. R İLE ELDE EDİLEN SONUÇLAR

Bu bölümde R programlama dili ile elde edilen tüm sonuçlar yer almaktadır. Doğruluk, kesinlik derecesi, anımsama ve f1-değeri için en iyi sonucu veren iki yöntem Lojistik Regresyon ve Yapay Sinir Ağları olmuştur. Çalışma süresi açısından en hızlı yöntem word2vec ile Rastgele Orman, en yavaş yöntem ise TF-IDF ile Destek Vektör Makinesi olmuştur. Ortalama bellek kullanımı açısından en fazla bellek kullanan yöntem kelime torbası ile Lojistik Regresyon, en az bellek kullanan yöntem ise word2vec ile Naif Bayes olmuştur. Ortalama işlemci kullanımı açısından en fazla işlemci kullanan yöntem TF-IDF ile Yapay Sinir Ağları, en az işlemci kullanan yöntem fastText ile Naif Bayes olmuştur. R ile elde edilen tüm sonuçlar Çizelge 4.7'de görülebilir.

Çizelge 4.7. R ile Elde Edilen Tüm Sonuçlar (k = 5, %80 - %20).

		Doğruluk (Accuracy)	Kesinlik Derecesi (Precision)	Anımsama (Recall)	F1-Değeri (F1-Score)	Süre (Sn)	Bellek Kullanımı (GB)	İşlemci Kullanımı (GHz)
Naif Bayes (Naive Bayes)	Kelime Torbası (Bag of Words)	0,77	0,82	0,69	0,75	1903,33	10,00	0,33
	TF-IDF	0,77	0,82	0,69	0,75	1042,18	9,29	0,35
	Word2vec	0,74	0,77	0,68	0,72	197,2	5,98	0,33
	FastText	0,72	0,72	0,71	0,71	193,72	7,99	0,31
Destek Vektör Makinesi (Support Vector Machine)	Kelime Torbası (Bag of Words)	0,87	0,86	0,88	0,87	8428,81	9,81	0,33
	TF-IDF	0,87	0,86	0,87	0,87	20769,76	8,71	0,32
	Word2vec	0,88	0,88	0,88	0,88	4732,64	7,73	0,33
	FastText	0,66	0,62	0,83	0,71	12652,88	7,49	0,62
Lojistik Regresyon (Logistic Regression)	Kelime Torbası (Bag of Words)	0,87	0,86	0,87	0,87	3503,74	10,30	0,44
	TF-IDF	0,87	0,86	0,87	0,87	3587,39	10,07	0,32
	Word2vec	0,88	0,88	0,88	0,88	126,07	6,53	0,35
	FastText	0,84	0,84	0,85	0,85	102,7	8,69	0,34
Rastgele Orman (Random Forest)	Kelime Torbası (Bag of Words)	0,77	0,78	0,75	0,77	2679,9	10,26	0,36
	TF-IDF	0,77	0,78	0,75	0,77	1434,59	10,08	0,34
	Word2vec	0,80	0,80	0,80	0,80	95	6,16	0,33
	FastText	0,72	0,73	0,71	0,72	91,29	8,11	0,34
Karar Ağacı (Decision Tree)	Kelime Torbası (Bag of Words)	0,71	0,67	0,82	0,74	1754,94	10,21	0,47
	TF-IDF	0,71	0,67	0,82	0,74	1989,78	9,28	0,34
	Word2vec	0,74	0,73	0,74	0,74	182,11	6,43	0,33
	FastText	0,66	0,64	0,73	0,68	180,63	8,79	0,32
Yapay Sinir Ağları (Artificial Neural Network)	Kelime Torbası (Bag of Words)	0,82	0,82	0,81	0,82	2029,24	9,35	0,75
	TF-IDF	0,82	0,82	0,82	0,82	1996,28	7,67	1,02
	Word2vec	0,86	0,86	0,87	0,87	451,03	6,63	0,64
	FastText	0,83	0,84	0,83	0,83	434,39	6,96	0,60

4.6. KARŞILAŞTIRMALI SONUÇLAR

Üç farklı programlama dilinde 6 farklı sınıflandırma yöntemi için elde edilen sonuçlar incelendiğinde kelime torbası ile Naif Bayes yöntemi için doğruluk değerleri %76-%77 seviyelerinde benzerlik gösterse de geçen süre açısından Python 11,34 sn ile en iyi performansı verirken R dilinde geçen süre 1903 sn olarak gözlemlenmiştir. TF-IDF ile Naife Bayes için doğruluk değerleri Python için %81 çıkarken, Matlab ve R dillerinde %77 seviyelerinde kalmıştır. Geçen süre açısından Python 11,39 sn ile en iyi performansı verirken R dilinde geçen süre 1042 sn olarak gözlemlenmiştir. Word2Vec ile Naif Bayes için doğruluk değerleri Python ve Matlab dillerinde %77 seviyelerindeyken R için sonuç %74 seviyelerinde gözlemlenmiştir. Geçen süre açısından Python 2,48 sn ile en iyi performansı verirken R dilinde geçen süre 197 sn olarak gözlemlenmiştir. FastText ile Naif Bayes için doğruluk değerleri Python'da %70, Matlab'da %75 ve R'de %72 olarak ortaya çıkmıştır. Süre açısından yine en hızlı dil 2,52 sn ile Python olurken en yavaş performans 193 sn ile R'de gözlemlenmiştir. Bellek ve işlemci kullanımı açısından en çok bellek ve işlemci kullanan dil Matlab olarak gözlenmiştir.

Kelime torbası ile Destek Vektör Makinesi yöntemi için üç dilde de doğruluk değerleri %87 seviyelerinde çıkmıştır. Python dilinde simülasyonlar 87968 sn sürerken R dilinde işlemler 8428 sn'de tamamlanmıştır. TF-IDF ile Destek Vektör Makinesi için doğruluk değerleri Python ve R'de %87 seviyelerinde çıkarken Matlab dilinde %71 olarak hesaplanmıştır. Geçen süre açısından en hızlı dil 4342 sn ile Python olurken Matlab dilinde simülasyonlar 201366 sn sürmüştür. Word2Vec ile Destek Vektör Makinesi için üç dilde de sonuçlar %87-%88 seviyelerinde benzer performans gösterirken geçen süre açısından en hızlı dil 666 sn ile Matlab olurken en yavaş performans 4732 sn ile R dilinde gözlemlenmiştir. FastText ile Destek Vektör Makinesi için doğruluk değerleri Python ve Matlab dillerinde %84-%85 seviyelerinde hesaplanırken R dilinde sonuç %66 seviyelerinde kalmıştır. Geçen süre açısından en hızlı dil 821 sn ile Matlab olurken, R dili 12652 sn ile en yavaş performansı göstermiştir. Bellek ve işlemci kullanımı açısından üç programlama dili de benzer performans göstermiştir.

Kelime torbası ile Lojistik Regresyon yöntemi için üç programlama dilinde de doğruluk değerleri %87 olarak hesaplanmıştır. Geçen süre açısından en hızlı dil 38 sn ile Python olurken en yavaş performans 3503 sn ile R dilinde gözlemlenmiştir. TF-IDF ile Lojistik Regresyon yöntemi için üç programlama dilinde de doğruluk değerleri %87 olarak

hesaplanmıştır. Geçen süre açısından en hızlı dil 33 sn ile Python olurken en yavaş performans 3587 sn ile R dilinde gözlemlenmiştir. Word2Vec ile Lojistik Regresyon yöntemi için doğruluk değeri %87-%88 seviyelerinde hesaplanmıştır. Geçen süre açısından en hızlı dil 9 sn ile Python olurken en yavaş dil 126 sn ile R olmuştur. FastText ile Lojistik Regresyon yöntemi için doğruluk değerleri Python dilinde %83, Matlab dilinde %86 ve R dilinde %84 olarak ölçülmüştür. Süre açısından Python dili 8,74 sn ile en hızlı performansı gösterirken en yavaş dil 102 sn ile R olmuştur. Bellek ve işlemci kullanımı açısından en az bellek ve işlemci kullanımı R dilinde gözlemlenirken en çok bellek ve işlemci kullanımı Matlab dilinde gözlemlenmiştir.

Kelime Torbası ile Rastgele Orman yöntemi için doğruluk değerleri Python dilinde %79, Matlab dilinde %84, R dilinde %77 olarak hesaplanmıştır. Süre bakımından en hızlı dil 35 sn ile Python olurken en yavaş performans 2679 sn ile R dilinde gözlemlenmiştir. TF-IDF ile Rastgele Orman yöntemi için doğruluk değerleri Python dilinde %79, Matlab dilinde %84, R dilinde %77 olarak hesaplanmıştır. Geçen süre açısından 47 sn ile Python en hızlı performansı gösterirken R dilinde işlemler 1434 sn'de tamamlanmıştır. Word2Vec ile Rastgele Orman yöntemi için doğruluk değerleri Python dilinde %81, Matlab dilinde %85 ve R dilinde %80 olarak hesaplanmıştır. Süre açısından en yavaş dil 1675 sn ile Matlab olurken en hızlı performans 46 sn ile Python'da gözlemlenmiştir. FastText ile Rastgele Orman yöntemi için doğruluk değerleri Python dilinde %73, Matlab dilinde %82, R dilinde %72 olarak hesaplanmıştır. Geçen süre açısından Python dili 49 sn ile en hızlı performansı gösterirken Matlab dilinde geçen süre 1662 sn olarak ölçülmüştür. Bellek ve işlemci kullanımı açısından üç programlama dili de benzer performanslar göstermiştir.

Kelime Torbası ile Karar Ağacı yöntemi için doğruluk değerleri %71-%72 seviyelerinde hesaplanmıştır. Çalışma süresi açısından en hızlı dil 158 sn ile Python olurken en yavaş performans 1754 sn ile R dilinde ölçülmüştür. TF-IDF ile Karar Ağacı yöntemi için doğruluk değerleri %71-%72 seviyelerinde hesaplanmıştır. Çalışma süresi açısından en hızlı dil 171 sn ile Matlab olurken en yavaş dil 1989 sn ile R olmuştur. Word2Vec ile Karar Ağacı yöntemi için doğruluk değerleri Python ve R dillerinde %74 olurken Matlab dilinde sonuç %71 seviyesinde çıkmıştır. Süre bakımından en hızlı dil 47 sn ile Matlab olurken en yavaş dil 182 sn ile R olmuştur. FastText ile Karar Ağacı yöntemi için doğruluk değerleri Python dilinde %67, Matlab dilinde %69 ve R dilinde %66 olarak hesaplanmıştır. Çalışma süresi açısından en hızlı dil 50 sn ile Matlab olurken en yavaş dil

180 sn ile R olmuştur. Bellek ve işlemci kullanımı açısından üç programlama dili de benzer performanslar göstermiştir.

Kelime Torbası ile Yapay Sinir Ağları yöntemi için doğruluk değerleri Python dilinde %82, Matlab dilinde %87, R dilinde %82 olarak hesaplanmıştır. Çalışma süresi açısından en hızlı dil 838 sn ile Python olurken en yavaş performans 2029 sn ile R dilinde ölçülmüştür. TF-IDF ile Yapay Sinir Ağları yöntemi için doğruluk değerleri Python ve R dillerinde %82 iken Matlab dilinde %87 seviyesinde hesaplanmıştır. Çalışma süresi açısından en hızlı dil 911 sn ile Python olurken en yavaş dil 1996 sn ile R olmuştur. Word2Vec ile Yapay Sinir Ağları yöntemi için doğruluk değerleri Python'da %87, Matlab'da %85 ve R'de %86 olarak ölçülmüştür. Süre açısından en hızlı dil 451 sn ile R olurken en yavaş performans 578 sn ile Python'da ölçülmüştür. FastText ile Yapay Sinir Ağları yöntemi için doğruluk değerleri %83-%84 seviyelerinde hesaplanmıştır. Çalışma süresi açısından en hızlı dil 434 sn ile R olurken en yavaş dil 575 sn ile Python olmuştur. Bellek ve işlemci kullanımı açısından üç programlama dili de benzer performanslar göstermiştir.

Üç farklı programlama dilinde gerçekleştirilen veri temizleme aşamaları incelendiğinde toplam süre açısından en hızlı performansı 31,88 sn ile Matlab dili göstermiştir. Veri temizleme işlemlerinin en yavaş gerçekleştiği dil ise Python olmuştur. Veri temizleme aşamaları tek tek incelendiğinde ise birinci temizleme aşaması olan harf dışındaki tüm sembollerin metinden çıkarılması işlemi R dilinde 6 sn, Python dilinde 7,89 sn, Matlab dilinde 12 sn sürmüştür. Tüm harflerin küçük harfe çevrilmesi işlemi Matlab dilinde 0,29 sn, Python dilinde 0,93 sn, R dilinde ise 3 sn sürmüştür. Bu iki aşama için süreler birbirlerine yakın olsa da etkisiz kelimelerin çıkarılması ve kelimelerin kök hallerine getirilmesi işlemleri süreler açısından farklılık göstermektedir. Etkisiz kelimelerin çıkarılması işlemi Matlab dilinde 8,52 sn sürerken, R dilinde 62 sn, Python dilinde ise 34,3 sn sürmüştür. Kelimelerin kök hallerine getirilmesi işlemi Matlab dilinde 11,07 sn, R dilinde 16 sn sürerken Python dilinde 137,26 sn sürmüştür.

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında sosyal medya kullanıcıları tarafından internet sitesinde paylaşılan film yorumlarından oluşan bir veri seti kullanılmıştır. Kullanılan veri seti 25.000 adet pozitif ve 25.000 adet negatif olarak etiketlenmiş yorum içermektedir. Veri setinin duygu analizine hazır hale getirilebilmesi için öncelikli olarak veri temizleme işlemleri uygulanmış, veri setinde yer alan harf dışındaki tüm karakterler çıkarılmış, veri setinde yer alan tüm kelimeler küçük harfe dönüştürülmüş, duygu analizine katkı sağlamayacak etkisiz kelimeler çıkarılmış ve tüm kelimeler kök hallerine dönüştürülerek veri temizleme aşaması tamamlanmıştır.

Veri setinin sınıflandırma yöntemleriyle duygu analizinin yapılabilmesi için öncelikli olarak bazı temsil yöntemlerine tabi tutulması gerekmektedir. Bu çalışmada metin ve kelime temsil yöntemleri olarak kelime torbası, TF-IDF, Word2Vec ve FastText kullanılmış, 3 farklı programlama dilinde her bir yöntem için veri seti duygu analizine hazır hale getirilmiştir. Duygu analizinin yapılabileceği farklı sınıflandırma yöntemlerinden 6 tanesi bu tez çalışmasında kullanılmıştır. Bunlar; Naif Bayes, Destek Vektör Makinesi, Lojistik Regresyon, Karar Ağacı, Rastgele Orman ve Yapay Sinir Ağları yöntemleridir. Python, Matlab ve R programlama dillerinde tüm sınıflandırma yöntemleri için k katlı çapraz doğrulama yöntemi kullanılarak hata matrisleri elde edilmiş ve verimli bir şekilde karşılaştırma yapılabilmesi için doğruluk, kesinlik derecesi, anımsama ve f1-değeri hesaplanmıştır. Yapılan hesaplamalar sonucunda en verimli sonucu veren yöntemler Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları olmuştur. Tüm programlar, sınıflandırmalar ve metin temsil yöntemleri arasında en iyi sonucu aşağıda verilen grup sağlamıştır:

- R ve Matlab programlama dillerinde, Lojistik Regresyon (Logistic Regression) sınıflandırma yönteminde, word2vec metin temsil yöntemi kullanılarak, en iyi doğruluk, kesinlik, anımsama ve f1-değeri olarak %88 değeri elde edilmiştir.

Bu tez çalışmasından elde edilen sonuçlar ışığında, gelecekte farklı metin temsil yöntemleri kullanılarak sınıflandırma yöntemlerinin nasıl sonuçlar vereceği incelenebilir, farklı veri temizleme yöntemleri kullanılarak daha iyi sonuçlar elde edilebilir.

6. KAYNAKLAR

[1] B. Agarwal, N. Mittal, P. Bansal, ve S. Garg, "Sentiment analysis using common-sense and context information", *Computational Intelligence and Neuroscience*, c. 2015, ss. 1–9, 2015, doi: 10.1155/2015/715730.

[2] U. T. Gürsoy, D. Bulut, C. Yiğit, ve S. Co, "Social Media Mining and Sentiment Analysis for Brand Management", *Global Journal of Emerging Trends in e-Business, Marketing and Consumer Psychology (GJETeMCP)*, c. 3, sayı 1, ss. 497–511, 2017.

[3] N. Mishra ve C. K. Jha, "Classification of Opinion Mining Techniques", *International Journal of Computer Applications*, c. 56, sayı 13, ss. 1–6, 2012, doi: 10.5120/8948-3122.

[4] Z. Drus ve H. Khalid, "Sentiment analysis in social media and its application: Systematic literature review", *Procedia Computer Science*, c. 161, ss. 707–714, 2019, doi: 10.1016/j.procs.2019.11.174.

[5] M. Ebrahimi, A. H. Yazdavar, ve A. Sheth, "Challenges of Sentiment Analysis for Dynamic Events", *IEEE Intelligent Systems*, c. 32, sayı 5, ss. 70–75, 2017, doi: 10.1109/MIS.2017.3711649.

[6] Y. Al Amrani, M. Lazaar, ve K. E. El Kadirp, "Random forest and support vector machine based hybrid approach to sentiment analysis", *Procedia Computer Science*, c. 127, ss. 511–520, 2018, doi: 10.1016/j.procs.2018.01.150.

[7] B. Bansal ve S. Srivastava, "Sentiment classification of online consumer reviews using word vector representations", *Procedia Computer Science*, c. 132, ss. 1147–1153, 2018, doi: 10.1016/j.procs.2018.05.029.

[8] S. Symeonidis, D. Effrosynidis, ve A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis", *Expert Systems with Applications*, c. 110, ss. 298–310, 2018, doi: 10.1016/j.eswa.2018.06.022.

[9] J. Ranganathan ve A. Tzacheva, "Emotion mining in social media data", *Procedia Computer Science*, c. 159, ss. 58–66, 2019, doi: 10.1016/j.procs.2019.09.160.

[10] B. Haryanto, Y. Ruldeviyani, F. Rohman, T. N. Julius Dimas, R. Magdalena, ve F. Muhamad Yasil, "Facebook analysis of community sentiment on 2019 Indonesian presidential candidates from Facebook opinion data", *Procedia Computer Science*, c. 161, ss. 715–722, 2019, doi: 10.1016/j.procs.2019.11.175.

[11] M. Y. Chen ve T. H. Chen, "Modeling public mood and emotion: Blog and news sentiment and socio-economic phenomena", *Future Generation Computer Systems*, c. 96, ss. 692–699, 2019, doi: 10.1016/j.future.2017.10.028.

[12] E. D'Andrea, P. Ducange, A. Bechini, A. Renda, ve F. Marcelloni, "Monitoring the public opinion about the vaccination topic from tweets analysis", *Expert Systems with Applications*, c. 116, ss. 209–226, 2019, doi: 10.1016/j.eswa.2018.09.009.

[13] V. A. Fitri, R. Andreswari, ve M. A. Hasibuan, "Sentiment analysis of social media

Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm", *Procedia Computer Science*, c. 161, ss. 765–772, 2019, doi: 10.1016/j.procs.2019.11.181.

[14] G. A. Ruz, P. A. Henríquez, ve A. Mascareño, "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers", *Future Generation Computer Systems*, c. 106, ss. 92–104, 2020, doi: 10.1016/j.future.2020.01.005.

[15] R. Ahuja, A. Chug, S. Kohli, S. Gupta, ve P. Ahuja, "The impact of features extraction on the sentiment analysis", *Procedia Computer Science*, c. 152, ss. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.

[16] M. S. Basarslan ve F. Kayaalp, "Sentiment Analysis with Machine Learning Methods on Social Media", *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, c. 9, sayı 3, ss. 5–15, 2020, doi: 10.14201/adcaij202093515.

[17] A. Alsaeedi ve M. Z. Khan, "A study on sentiment analysis techniques of Twitter data", *International Journal of Advanced Computer Science and Applications*, c. 10, sayı 2, ss. 361–374, 2019, doi: 10.14569/ijacsa.2019.0100248.

[18] A. M. Abirami ve V. Gayathri, "A survey on sentiment analysis methods and approach", *2016 8th International Conference on Advanced Computing, ICoAC 2016*, ss. 72–76, 2017, doi: 10.1109/ICoAC.2017.7951748.

[19] J. Khairnar ve M. Kinikar, "Machine Learning Algorithms for Opinion Mining and Sentiment Classification", *International Journal of Scientific and Research Publications*, c. 3, sayı 6, ss. 1–6, 2013.

[20] A. Tyagi ve N. Sharma, "Sentiment Analysis using logistic regression and effective word score heuristic", *International Journal of Engineering and Technology(UAE)*, c. 7, sayı 2, ss. 20–23, 2018, doi: 10.14419/ijet.v7i2.24.11991.

[21] H. Kaur, V. Mangat, ve Nidhi, "A survey of sentiment analysis techniques", *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, ss. 921–925, 2017, doi: 10.1109/I-SMAC.2017.8058315.

[22] M. M ve S. Mehla, "Sentiment Analysis of Movie Reviews using Machine Learning Classifiers", *International Journal of Computer Applications*, c. 182, sayı 50, ss. 25–28, 2019, doi: 10.5120/ijca2019918756.

[23] F. Hemmatian ve M. K. Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis", *Artificial Intelligence Review*, c. 52, sayı 3, ss. 1495–1545, 2019, doi: 10.1007/s10462-017-9599-6.

[24] T. U. Haque, N. N. Saber, ve F. M. Shah, "Sentiment analysis on large scale Amazon product reviews", *2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018*, sayı May, ss. 1–6, 2018, doi: 10.1109/ICIRD.2018.8376299.

[25] F. Ali *vd.*, "Transportation sentiment analysis using word embedding and ontology-based topic modeling", *Knowledge-Based Systems*, c. 174, ss. 27–42, 2019, doi: 10.1016/j.knosys.2019.02.033.

[26] N. Nedjah, I. Santos, ve L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network via word embeddings", *Evolutionary Intelligence*, ss. 2–6, 2019, doi: 10.1007/s12065-019-00227-4.

[27] R. Vallat, “Pingouin: Statistics in Python”, *Journal of Open Source Software*, c. 3, sayı 31, s. 1026, 2018, doi: 10.21105/joss.01026.

[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, ve C. Potts, “Learning word vectors for sentiment analysis”, *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, c. 1, ss. 142–150, 2011.

[29] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, ve T. Mikolov, “Learning word vectors for 157 languages”, *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, ss. 3483–3487, 2019.

[30] B. S. Harish, D. S. Guru, ve S. Manjunath, “Representation and classification of text documents: A brief review”, *IJCA, Special Issue on Recent Trends in Image Processing and Pattern Recogniton*, sayı 2, ss. 110–119, 2010.

[31] B. Yekkehkhany, A. Safari, S. Homayouni, ve M. Hasanlou, “A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, c. 40, sayı 2W3, ss. 281–285, 2014, doi: 10.5194/isprsarchives-XL-2-W3-281-2014.

[32] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, ve Z. Liu, “Dynamic ReLU”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, c. 12364 LNCS, ss. 351–367, 2020, doi: 10.1007/978-3-030-58529-7_21.

[33] Anonim (2021, 02 Ağustos), "*HWiNFO - Free System Information, Monitoring and Diagnostics*”. [Online], Erişim: https://www.hwinfo.com/.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Batuhan Cem ÖĞE

Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Yüksek Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2021
Lisans	Bilgisayar Mühendisliği	İstanbul Kültür Üniversitesi	2014
Lise		Samsun Anadolu Lisesi	2009

YAYINLAR

B. C. Öğe, F. Kayaalp (2021), "Farklı Sınıflandırma Algoritmaları ve Metin Temsil Yöntemlerinin Duygu Analizinde Performans Karşılaştırılması", *Uluslararası Mühendislikte Yapay Zeka ve Uygulamalı Matematik Konferansı,* Basımda.