

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**DOĞAL DİL İŞLEME İLE TÜRKÇE İÇERİKLİ PAYLAŞIMLARDAN SOSYAL
MEDYA KULLANICILARININ DUYGU ANALİZİ**

Çağla BALLI

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2021**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

DOĞAL DİL İŞLEME İLE TÜRKÇE İÇERİKLİ PAYLAŞIMLARDAN SOSYAL MEDYA KULLANICILARININ DUYGU ANALİZİ

Çağla BALLI

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mehmet Serdar GÜZEL

Son yıllarda teknolojik ilerleyişin arasında yükselen ve bu ilerleyişe yön veren Yapay Zeka, Makine Öğrenmesi konularında yapılan çalışmalar, akademik araştırmaların ve sektörde bu alana olan talebin artmasıyla büyük gelişme göstermiştir. Teknolojinin günden güne bu büyük ilerleyişi ile birlikte 2020 yılından beri hayatımızın bir parçası haline gelen pandemi sonrasında bireylerin hayatlarında sosyal medya her zaman olduğundan daha büyük bir yer tutmaya başlamıştır. Bu sebeple sosyal medya paylaşımları duygu analizi alanı için büyük bir veri kaynağı haline gelmiştir. Teknolojik gelişmeden nasibini alan ve Yapay Zeka uygulamalarının en önemli başlıklarından biri olan Doğal Dil İşleme methodu bireylerin duygu analizi konusunda oldukça başarılı sonuçlar veren bir makine öğrenmesi modelidir. Bu çalışmada literatürde bulunan makine öğrenmesi konularından Doğal Dil İşleme methodu temel alınarak ile kullanıcıların Twitter üzerinde yaptıkları Türkçe paylaşımlardan farklı algoritmalar kullanılarak duygu analizleri yapılmış ve elde edilen sonuçlar ortaya konulmuştur. Çalışmada modelin eğitilmesi için hali hazırda bulunan bir veri setinin yanında pandemi, korona gibi kelimeler ile filtrelenmiş tweetlerden oluşan bir veri seti el ile işaretlenerek oluşturulup bu veri seti ile ayrıca eğitilen modelin başarısı incelenmiş ve karşılaştırılmıştır. Modelde duygu analizi verilerin pozitif veya negatif olmasına göre sınıflandırılmış ve bu aşamada makine öğrenmesinin çeşitli sınıflandırma algoritmaları kullanılmıştır. Çalışma, literatürde doğal dil işleme başlığı altında Türkçe veriler kullanılarak yapılan çalışmalara katkıda bulunmak amacıyla doğal dil işlemenin Türkçe veriler üzerindeki başarısını incelemek ve sonuçları ortaya koymak için gerçekleştirilmiştir.

Temmuz 2021, 66 pages

Anahtar Kelimeler: Yapay Zeka, Makine Öğrenmesi, Doğal Dil İşleme, Duygu Analizi, Sosyal medya, Twitter

ABSTRACT

Master Thesis

SENTIMENT ANALYSIS OF SOCIAL MEDIA USERS FROM TURKISH CONTENT WITH NATURAL LANGUAGE PROCESSING

Çağla BALLI

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Mehmet Serdar GÜZEL

In recent years, studies on Artificial Intelligence, Machine Learning, which has risen among the technological progress and directed this progress, have shown great development with the increase in academic research and the demand for this field in the sector. With this great progress of technology day by day, social media has started to occupy a larger place in the lives of individuals after the pandemic, which has become a part of our lives since 2020. For this reason, social media shares have become a great data source for the field of sentiment analysis. Natural Language Processing method, which is one of the most important topics of Artificial Intelligence applications and has its share of technological development, is a machine learning model that gives very successful results in the emotion analysis of individuals. In this study, based on the Natural Language Processing method, which is one of the machine learning topics in the literature, sentiment analyzes were made using different algorithms from the Turkish posts made by users on Twitter, and the results were revealed. In the study, besides an existing dataset for training the model, a dataset consisting of tweets filtered with words such as pandemic and corona was created by manually marking, and the success of the trained model was examined and compared with this dataset. In the model, sentiment analysis was classified according to whether the data were positive or negative, and at this stage, various classification algorithms of machine learning were used. The study was carried out to examine the success of natural language processing on Turkish data and to reveal the results in order to contribute to the studies using Turkish data under the title of natural language processing in the literature.

July 2021, 66 pages

Key Words: Artificial Intelligence, Machine Learning, Natural Language Processing, Sentiment Analysis, Social Media, Twitter

TEŞEKKÜR

İlk olarak Yüksek lisans eğitimim süresince göstermiş olduğu anlayış ile motive eden, her zaman öğrencilerine güvenen tavrı ile yol gösteren deneyimlerini paylaşan ve çalışmama önerileri ile katkıda bulunan saygıdeğer hocam Doç.Dr.Mehmet Serdar GÜZEL (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı)‘ e sonsuz teşekkürlerimi sunarım. Zorlu bir süreci beni motive etmesi ve yol göstermesi sayesinde iyi bir çalışma ile tamamladığımı düşünüyorum.

Ayrıca her koşulda yanımda olan kıymetli ailem; hiçbir konuda desteklerini esirgemeyen, bütün varlığını çocuklarına adanmış biricik annem Ayşe ve babam Tahsin Ballı’ya; üzerimdeki emeklerini hiçbir zaman ödeyemeyeceğim canım kardeşlerim ablam Sevinç ve abim Çağdaş Ballı’ya sonsuz teşekkürlerimi sunarım.

Çağla BALLI
Ankara, Temmuz 2021

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT.....	iii
TEŞEKKÜR	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
1. GİRİŞ	1
2. DOĞAL DİL İŞLEME	5
2.1 Veri Ön İşleme.....	5
2.1.1 Gürültülü veri temizleme (Removal Noisy Data).....	6
2.1.2 Duraklama kelimelerinin kaldırılması (Removal Stopwords).....	6
2.1.3 Kelime köklerinin bulunması (Stemming/ Lemmatization).....	6
2.1.4 Simgeleme (Tokenization)	7
2.1.5 Metin normalleştirme (Normalization)	8
2.1.6 Metin vektörleştirme (Text Vectorization)	8
3. TWITTER API.....	13
3.1 Twitter API Kullanımı.....	14
3.1.1 Projeler ve uygulamalar	15
3.1.2 Kimlik doğrulama yöntemleri.....	16
3.1.3 Tweepy kullanımı	18
3.1.4 Tweet işaretleme.....	19
4. MAKİNE ÖĞRENMESİ ALGORİTMALARI	21
4.1 Logistik Regresyon (Logistic Regression).....	21
4.2 Naive Bayes Sınıflandırması (Naive Bayes Classification).....	22
4.3 Destek Vektör Makineleri (Support Vector Machines).....	24
4.4 Rastgele Orman Algoritması (Random Forest Algorithm).....	25
4.5 Stokastik Gradyan İniş Algoritması (Stochastic Gradient Descent Algorithm)	27
4.6 Uzun Kısa Süreli Bellek (Long Short-Term Memory)	28

5.LİTERATÜR.....	29
6.METODOLOJİ	34
6.1 Model Mimarisi	36
6.2 Veri Oluşturma	37
6.3 Veri Ön İşleme.....	39
6.4 Model Oluşturma	44
6.4.1 Logistik Regresyon modeli	45
6.4.2 SGD Modeli	45
6.4.3 Rastgele Orman modeli	45
6.4.4 Bayesian Modeli	45
6.4.5 SVM Modeli.....	46
6.4.6 LSTM Modeli	46
7. DENEYLER	49
8. SONUÇ.....	58
KAYNAKLAR	61
ÖZGEÇMİŞ.....	66

KISALTMALAR DİZİNİ

AI	Artificial Intelligence
NLP	Natural Language Processing
SMT	Statistical Machine Translation
TF	Term Frequency
IDF	Inverse Document Frequency
API	Application Programming Interface
SVM	Support Vector Machine
SDG	Stochastic Gradient Descent
SGDC	Stochastic Gradient Descent Classifier
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
kNN	k-Nearest Neighbors
NLTK	Natural Language Toolkit
NALPRONN	Natural Language Processor with Neural Networks

ŞEKİLLER DİZİNİ

Şekil 2.1 Veri Ön İşleme	6
Şekil 2.2 Kelime Torbası Örneği	9
Şekil 2.3 Kelime Torbası Vektör Dönüşüm Örneği	9
Şekil 2.4 Kelime Torbası Vektör Dönüşüm Örneği	10
Şekil 2.5 Kelimelerin indekslenip tarafından seyreke matrise dönüştürülmesi	10
Şekil 2.6 Terim Frekansı Formülü	11
Şekil 2.7 Ters Belge Frekansı Formülü	11
Şekil 3.1 Tweepy kimlik doğrulama örneği	18
Şekil 3.2 Tweepy search methodu kullanım örneği	19
Şekil 3.3 Tweepy stream methodu kullanım örneği	19
Şekil 3.4 İşaretili veri örneği	20
Şekil 4.1 Basit Logistik Regresyon Modeli (Swaminathan 2018)	22
Şekil 4.2 Bayes Teoremi'nin formülü	22
Şekil 4.3 Naive Bayes Örnek Veri Kümesi	23
Şekil 4.4 Naive Bayes Formülü (Anonymus. 2020a)	24
Şekil 4.5 Ayırma alt düzlemi örneği (Reddy 2018)	24
Şekil 4.6 Özellik önem grafik örneği (Donges 2019)	26
Şekil 5.1 Çeşitli Türkçe Duygu Analizi Çalışmaları ve Doğruluk Oranları	34
Şekil 6.1 Model mimarisi genel şeması	36
Şekil 6.2 açık kaynaklı veri setinden alınmış örnek veri (Beyaz 2020)	38
Şekil 6.3 snscape kütüphanesi tweet toplama örneği	39
Şekil 6.4 SentimentSet veri setine ait tweet toplama örneği	39
Şekil 6.5 Gürültülü verinin temizlenmesi ve duraklama kelimelerinin kaldırılması veri ön işlem örneği	41
Şekil 6.6 Zemberek kütüphanesi kullanılarak bulunan kelime kökleri örneği	42
Şekil 6.7 Snowball kütüphanesi kullanılarak bulunan kelime kökleri örneği	42
Şekil 6.8 Metin Vektörleştirme için kullanılan Tf-idf yöntemi gösterimi	43
Şekil 6.9 LSTM için kullanılan Tokenizer gösterimi	44
Şekil 6.10 LSTM modeli ve hiper parametreleri	47
Şekil 6.11 LSTM modeli fit ve evaluate methodları ve parametreleri	48
Şekil 6.12 LSTM modeli predict_classes methodu gösterimi	48
Şekil 7.1 Açık veri setinin kategori oranlarının görselleştirmesi	49
Şekil 7.2 Açık veri seti ile makine öğrenmesi algoritmalarının doğruluk oranlarının kök bulma kütüphanelerine göre karşılaştırılması	50
Şekil 7.3 Açık veri seti ve Zemberek kök bulma kütüphanesi ile eğitilmiş Bayesian modelinin veri setine ait test verisi ile test sonuçlarının dağılımı	51
Şekil 7.4 Açık veri seti ve NLTK Snowball kök bulma kütüphanesi ile eğitilmiş Logistik Regresyon modelinin veri setine ait test verisi ile test sonuçlarının dağılımı	51
Şekil 7.5 SentimentSet veri setinin kategori oranlarının görselleştirmesi	52
Şekil 7.6 SentimentSet veri seti ile makine öğrenmesi algoritmalarının doğruluk oranlarının kök bulma methodlarına göre karşılaştırılması	53

Şekil 7.7 SentimentSet veri seti ve Zemberek kök bulma kütüphanesi ile eğitilmiş SVM modelinin veri setine ait test verisi ile test sonuçlarının dağılımı	53
Şekil 7.8 SentimentSet veri seti ve NLTK Snowball kök bulma kütüphanesi ile eğitilmiş Rastgele Orman modelinin veri setine ait test verisi ile test sonuçlarının dağılımı ...	54
Şekil 7.9 LSTM farklı verisetleri ile eğitim ve test sonuçları	54
Şekil 7.10 SentimentSet veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama doğruluk değerleri	55
Şekil 7.11 SentimentSet veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama hata oranı değerleri.....	55
Şekil 7.12 SentimentSet veri seti ile eğitilmiş LSTM modelinin veri setine ait test verisi ile test sonuçlarının dağılımı	56
Şekil 7.13 Açık veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama doğruluk değerleri	56
Şekil 7.14 Açık veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama hata oranı değerleri	57
Şekil 7.15 Açık veri seti ile eğitilmiş LSTM modelinin veri setine ait test verisi ile test sonuçlarının dağılımı.....	57

1. GİRİŞ

Yapay Zeka (AI) insan zekası ile gerçekleştirilen davranışların makine ile gerçekleştirilmesini hedef alan zekayı ifade eder. Bu hedef doğrultusunda insan zekasını taklit etmeye çalışır. Yapay Zeka konusunda yapılan çalışmaların ve bu alanda elde edilen başarı oranlarının artmasının sebebi büyük verilerin kullanılabildiği makinelerin ve süper hızlı bilgisayarların geliştirilmesidir. (IBM 2020)

Bilgisayar biliminde tarihçesi çok eskiye dayanan yapay zeka konuları teknolojik gelişmelerin içinde farklı akademik alanlarda kendini göstermeyi başarmış ve üzerinde sıkça durulan ve günden güne iyileşen gerek güncel akademik çalışmalarda gerek bilgisayar sektöründe gerekse günlük hayatta kullanılan teknolojinin büyük bir parçası haline gelmiştir. 1950 yılında Alan Turing'in (1950) Turing Testi olarak anılan çalışması akıllara sıkça gelse de Yapay Zeka'nın kavramlaştırılması esasen Ramon Llull The Ultimate General Art kitabının yayımlandığı 1308 yılına kadar dayanmaktadır. "Yapay Zeka" terimi olarak kullanımı ise John McCarthy, Marvin Minsky ve Nathaniel Rochester tarafından 1956 yılında gerçekleştirilmiştir. (Cerebro 2018)

Yapay zeka alanında yapılan çalışmaların ülke ve dünya çapında değer görmesi modellerin eğitilebileceği son derece hızlı bilgisayarların ulaşılabilir olması, bulut yöntemlerinin eklenmesi, bu alanda yazılmış kütüphane sayılarının artması, çalışma yapan kişi sayısı arttıkça verilere ulaşmanın kolaylaşmasına yol açmıştır. Son yıllarda görülen büyük gelişmeler ile birlikte bu alana duyulan arz talep ilişkisi güçlenmiş ve araştırmacıları bu konuda çalışma yapmaya itmiştir. Elde edilen hatırı sayılır sonuçlar yeni çalışmaların tabanını oluşturarak gelişmeyi takip etmesi güç olacak derecede hızlandırmıştır. Bu tez kapsamında yapay zekanın makine öğrenmesi alanında geliştirilmiş çeşitli algoritmalar, programlara araçları ve python ve java dilinde geliştirilmiş kütüphaneler kullanılarak çalışma bu hızda gelişen bu alana uygun bir bütün halinde ortaya konmuştur.

Yapay zeka, insan davranışlarını taklit etmeye çalışan bir konu iken makine öğrenmesi büyük verileri öğrenen ve bunlardan çıkarımlar yapan bilgisayar modelleridir (Safak 2017). Bu modeller istatistiksel algoritmalarla temellendirilen ve büyük verileri işleyerek bunlar üzerinden çıkardığı matematiksel sonuçlarla tahminlerde bulunan bir dizi adımdan oluşur (Anonymus 2017). Bu modeller tekrar edilip farklı veriler için kullanılabilir, farklı algoritmalarla iyileştirilebilir veya daha iyi sonuçlar elde etmek için farklı hiper parametrelerle yeniden eğitilebilir. Eğitilmiş model kullanılarak farklı test verileriyle yeniden eğitilmeden tekrar kullanılabilir. Amaç her zaman daha iyi bir tahminleme yapıp daha iyi sonuçlar elde etmektir.

Yapay zeka biliminin en önemli alt dallarından biri de Doğal dil işleme (NLP), bilgisayar ve insan dili arasındaki etkileşimlerle, özellikle doğal dil verilerini işlemek ve analiz etmek için bilgisayarların nasıl programlanacağıyla ilgili bir dilbilim, bilgisayar bilimi ve yapay zeka alt alanıdır. Amacı, metinlerin yazılmasında kullanılan doğal dilin kendine özgü yönleriyle birlikte içeriğini “anlayabilen” bir bilgisayar elde etmektir. Sonrasında metin içeriğinde yer alan bilgileri sınıflandırıp, düzenlemek bilgileri doğru biçimde çıkarmak işlemleri gerçekleştirilebilir. (Anonymus 2021a). Doğal Dil İşleme konusu 1950’li yıllara kadar dayanmaktadır. Turing testi (Turing 1950) doğal dili yorumlama ve üretme üzerinde olan ilk çalışmalardan biridir. Önceleri Doğal Dil İşleme sistemleri sembolik yöntemlerle tasarlandı. Bir dizi kuralın elle kodlanması ve sözlük aramasıyla oluşturulan bu sistemlerin yerini günümüzde sıklıkla kullanılan makine öğrenmesi algoritmalarına dayalı sistemler aldı. Ezbere dayalı elle kodlama yöntemleri yeni sözcükler ve farklı kombinasyonlara açık olmadığından otomatik öğrenme prosedürleri alışılmadık veya hatalı girdilere karşı istatistiksel çıkarım algoritmalarından yararlanarak daha doğru sonuçlar sağlayabilir.

Makine Öğrenmesi türleri denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) olarak üçe ayrılmaktadır. Denetimli öğrenme verilerin giriş ve çıkış değerlerinin belli olduğu, etiketlenen verilerle yapılan öğrenmedir.

Denetimsiz öğrenme girdilere karşı gelen çıkış değerleri bilinmemektedir. Giriş değerlerine göre verilerin kendi içinde gruplanması beklenmektedir. Pekiştirmeli Öğrenme ise etiketli verilerin olmadığı ancak çıktı değerlerinin geri bildirim olarak ağı verilip yeniden eğitildiği öğrenme biçimidir. NLP'nin en yaygın kullanım alanlarından biri olan Duygu Analizi, yazılı verilerin yorumlanması, işlenmesi ve duygulara göre sınıflandırılmasıdır. Bu alanda yapılan çalışmalar için yazı teknikleri, dil araçları farklı dillere yönelik linguistik geliştirmeler, dil bilimi üzerinde kelimenin farklı anlamlarının yorumlanması ve kelimelerin bir araya geldiğinde ifade ettiği duygunun değişmesi gibi veriden çıkarılan bu anlamların istatistik biliminin ışığı altında makine öğrenmesi algoritmalarıyla çalışılması oldukça başarılı sonuçlar oluşturmıştır. Bahsedildiği gibi, duygu analizi konusunun temelini oluşturan dil bilimi, her dile özgü farklı bir çalışmanın yapılması gerektiğini ortaya koymaktadır.

Bu tez kapsamında Duygu Analizi Doğal Dil İşleme yöntemleri ve denetimli öğrenme kullanılan makine öğrenmesi algoritmaları ile ele alınmıştır. Literatürde ve dünya çapında duygu analizi çok yaygın bir konudur ve bu konuda pek çok çalışma yapılmıştır. Bu çalışmaların çoğu İngilizce dili içindir. Bu çalışmada Türkçe veriler üzerinde duygu analizi yapılmıştır. Türkçe dilinde yapılacak bir çalışma için Türkçeye özgü dil özellikleri göz önüne alınarak morfolojik olarak kelimelerin yorumlanması ve işlenmesi için çeşitli kütüphaneler kullanılmıştır. İşaretlenmiş (etiketli) veriler dile özgü ön işlemlerden geçirilip vektörlere çevrilerek Makine öğrenmesi algoritmalarının kullanabileceği sayısal değerlere uygun hale getirilmiştir. Farklı algoritmalar üzerinde Makine Öğrenmesi modelleri geliştirilmiş ve bu modellerle yeni veriler üzerinde tahminleme yapılmıştır.

Günümüzde Duygu Analizi için kullanılan en yaygın ve bu konuda en çeşitli veri kaynaklarından biri de sosyal medyadır. Sosyal medya verinin anlık olarak sürekli yenilendiği güncel konulara dair bilgilerin hızla toplumlar ve kişiler arası yayıldığı, siyaset, bilim, tarih gibi her konuda türlü farklı bakış açısıyla dolu, şu anki büyük veri çalışmalarında önemli bir veri kaynağı haline gelmiştir.

Twitter sosyal medyada gncel konular veya byk bařlıklar altında yazılı veri saęlama konusunda en iyi kaynak alınabilecek web sitelerinden biridir. Bu tez kapsamında kiřilerin attıęı Trke ierikli tweetler anonim olarak alınıp iřaretili veriler halinde makine ęrenmesi algoritmalarıyla beraber iřlenip yorumlanarak duygu analizi yapılmıřtır.

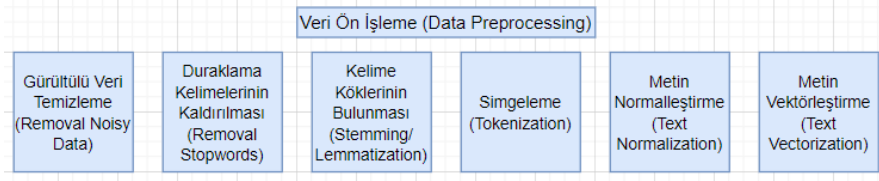


2. DOĞAL DİL İŞLEME

NLP yani Doğal Dil İşleme ‘nin, yapay zekanın alt dallarından biri olduğundan ve bilgisayar ile insan dili arasındaki etkileşimlerle verileri işlemek ve analiz etmek için bilgisayarların nasıl programlanacağı ile ilgili bir dilbilim olduğundan söz etmiştik. Doğal Dil İşleme, doğal dillerin kurallı yapısının çözümlenerek anlaşılması veya yeniden üretilmesi amacını taşır. Bu çözümlemenin insana getireceği kolaylıklar, yazılı dökümanların otomatik çevrilmesi, soru-cevap makineleri, otomatik konuşma ve komut anlama, konuşma sentezi, konuşma üretme, otomatik metin özetleme, bilgi sağlama gibi birçok başlıkla özetlenebilir. Bilgisayar teknolojisinin yaygın kullanımı, bu başlıklardan üretilen uzman yazılımların gündelik hayatımızın her alanına girmesini sağlamıştır. Örneğin, tüm kelime işlem yazılımları birer imlâ düzeltme aracı taşır. Bu araçlar aslında yazılan metni çözümleyerek dil kurallarını denetleyen doğal dil işleme yazılımlarıdır. (Anonymus. 2021a). Doğal Dil İşleme’de, veri ilk haliyle doğal dille yazılmış bir metinken, bilgisayarların anlayabileceği makine diline çevrilmesi gerekmektedir. Bu işlemler makine öğrenmesi alanında genel olarak veri ön işleme olarak adlandırılır.

2.1 Veri Ön İşleme

Bir makine öğrenmesi modeli için en önemli işlemlerden biri veri ön işleme biridir. Öyle ki, modelin başarısı çoğunlukla öğrendiği verinin kalitesine ve ne derece iyi bir ön işleme tabii tutulduğuna göre değişkenlik gösterir. Doğal Dil İşleme alanında, veri ön işleme eldeki metin verisinin sayısal değerlere dönüştürülüp makinenin anlayabileceği forma sokulması ve modellere giriş değeri olarak verilme gerekliliği bulunduğundan büyük önem taşımaktadır. Bazı veri ön işleme işlemleri Şekil 2.1’de gösterilmiştir.



Şekil 2.1 Veri Ön İşleme

2.1.1 Gürültülü Veri Temizleme (Removal Noisy Data)

Metnin içerisinde anlamlı olmayan ve istenmeyen model eğitilirken gerekli görülmeyen ve modelin başarısını kötü yönde etkileyecek verilerin çıkarılması işlemidir. Bu veriler, web sitesi linkleri, emojiler, hashtaglar, konu başlıkları, çeşitli semboller, noktalama işaretleri bu verilere örnek olarak verilebilir.

2.1.2 Duraklama Kelimelerinin Kaldırılması (Removal Stopwords)

Doğal Dil İşleme, diller arasında farklılık gösteren bir konudur. Ön işleme işlemleri her dile özgü farklı şekillerde gerçekleştirilmektedir. Duraklama kelimeleri, cümlelerin içinde cümlelerin anlamını değiştirmeyen ancak dil yapısı gereği kullanılan bazı kelimelerdir. Türkçe’de bağlaçlar bunlara örnek olarak verilebilir. Bu kelimelerin çıkarılması Doğal Dil İşleme’nin hangi amaçla yapıldığına bağlı olarak değişkenlik gösterir. Duygu analizi konusunda, cümlelerden çıkarılacak duygu ifadesini bozan ve yanlış sınıflandırmaya yol açan kelimeler de duraklama kelimeleri olarak sayılabilir.

2.1.3 Kelime Köklerinin Bulunması (Stemming/ Lemmatization)

Stemming(kök bulma), kelimelerin köklerine indirgenme yani kök haline getirilme sürecidir. Bu süreç, NLP uygulanan dilin yapısına göre değişkenlik gösterir. Dillere göre farklı stemming algoritmaları bulunmaktadır.

Türkçe’de stemming işlemi kelimenin geldiği anlamdan en sade halini almak demek olsa da stemming işlemi bundan biraz daha farklı çalışır. Örneğin, yer, yemek, yeter kelimelerinin aynı metinde bulunması durumunda stemming süreci üçünün de kökünü -ye olarak alır. Burada aslında yer kelimesinin ve yeter kelimesinin Türkçe’deki anlamsal kökü -ye değildir. Stemming işlemi yapılırken elde edilen kelimelerin anlamlı olması zorunlu değildir.

Lemma, stemming ten farklı olarak kelimenin köküne morfolojik olarak inilme işlemidir. Örneğin “Alınan” kelimesine lemmatization uygularsak “almak” kelimesini elde ederiz. Kelimelerin daha sade hallerinin sözlükteki karşılığı olarak tanımlanabilir. Lemmatization yapmak için her zaman sözlüğe ihtiyaç duyulur. Kelimeler bu sözlükteki karşılıklarına göre sadeleştirilirler. Alınmak kelimesinin lemma’sı “almak”tır, yani kelimelerin yalın halinin sözlükteki karşılığı gibi tanımlayabiliriz “istiyorlar” fiilinin çekimlenmemiş hali “istemek” tir. Bu fiil iistemek fiilinin üçüncü çoğul şahısa çekimlenmiş halidir. Kelimedeki bu çekim kaldırılarak kelimenin ilk haline “lemma” denir. Örnekteki istemek bir lemmadır. İngilizce bir örnek olarak “feeds”, feed fiilinin üçüncü tekil şahsının geniş zamanda çekimlenmiş halidir. (Noyan, 2019)

2.1.4 Simgeleme (Tokenization)

Simgeleme cümlelerin farklı kurallara göre daha küçük anlamlı parçalara bölünmesi işlemine denir. Cümle kelimelere, deyimlere ve söz öbeklerine ayrılır. Bu işlem ayrıca metin bölümlleme (text segmentation) veya sözcük analizi (lexical analysis) olarak da adlandırılır. Örneğin, “Sporun önemi, günlük yaşamın teknoloji ile birlikte daha hareketsiz ve durağan bir hale gelmesiyle beraber artıyor.” Kelimelere ayrılan bu cümle şu hale gelir; ['Sporun', 'önemi', ',', 'günlük', 'yaşamın', 'teknoloji', 'ile', 'birlikte', 'daha', 'hareketsiz', 've', 'durağan', 'bir', 'hale', 'gelmesiyle', 'beraber', 'giderek' 'artıyor', '.']. Bu cümle sonrasında duraklama kelimelerinin kaldırılması, kök bulma işlemlerinin yapılması adımlarına tabii tutulmalıdır.

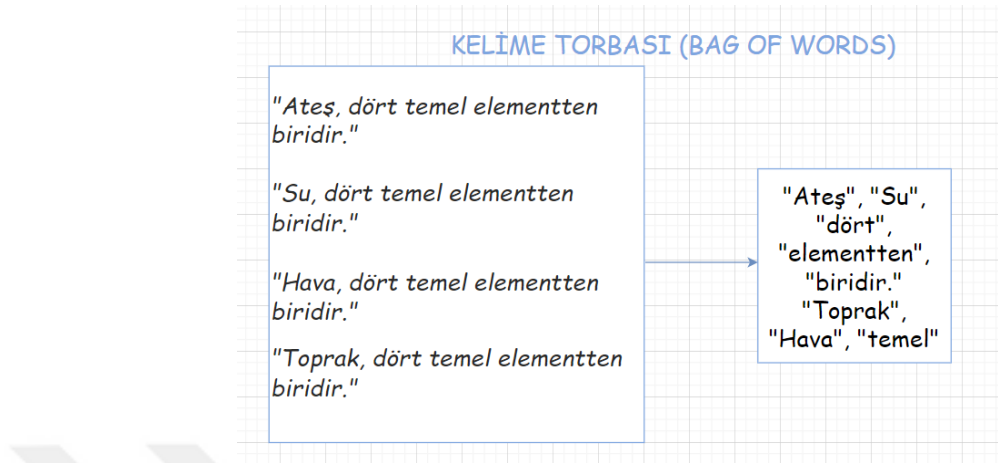
2.1.5 Metin Normalleştirme (Normalization)

Veri ön işleme adımları arasında olup çoğunlukla uygulanmayan adımlardan biri de metin normalleştirmesidir. Metin normalleştirme, basitçe ifade edilirse metnin standart bir forma dönüştürülmesidir. Örneğin, Türkçe atılmış bir tweette “tmm”, “tamaaammm” kelimelerinin kanonik formu olan “tamam” kelimesine dönüştürülmesidir. Fakat metin normalleştirmenin kök bulma işlemleri gibi standart bir yolu yoktur. Çoğunlukla göreve bağlı olarak farklı yaklaşımlar gerektirir. İstatiksel makine çevirisi (SMT), sözlük eşleştirmeleri ve yazım düzeltme tabanlı yaklaşımlar gibi çeşitli metin normalleştirme yaklaşımları bulunmaktadır (Ganesan 2019).

2.1.6 Metin Vektörleştirme (Text Vectorization)

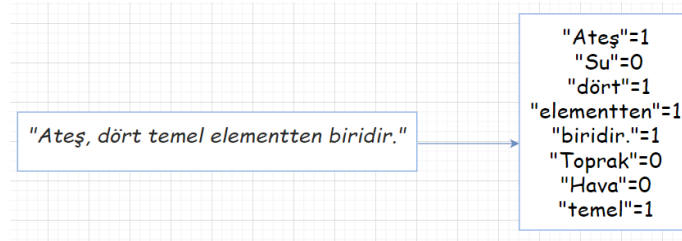
Metin Vektörleştirme, NLP alanında çalışma yapılırken metin halindeki doğal dili bilgisayarın anlayıp yorumlaması ve makine öğrenmesi algoritmalarını çalıştırmak için sayısallaştırma yani verileri vektörler haline getirme işlemine denir. Makine öğreniminde nitelikler (feature) matris faktörleştirme gibi bazı matematiksel işlemleri gerçekleştirebilecek sayısal özelliklerdir. Bu işlemin kök bulma, duraklama kelimelerinin kaldırılması gibi veri ön işleme işlemlerinden sonra yapılması gerekmektedir. Bu işlemden sonra ortaya çıkan sonuç sayısal değerlerdir. Bu sayısal değerler kullanılacak makine öğrenmesi algoritmasının giriş değerlerini oluşturur. Kullanılacak almortimaya göre vektörleştirme işleminin yapısı ve parametreleri değışkenlik gösterir.

Kelime torbası (Bag of words), Doğal dil işlemede kullanılan temel metin vektörleştirme modellerinden biridir. Kelime torbası denme sebebi ise belgedeki kelimelerin herhangi bir sırası olmadan konulmasıdır. Bu modelin temel amacı kelimelerin belgede mevcut olup olmadığını göstermektir. Şekil 2.2 ‘de gösterilen örnekte gösterilen her bir cümle ayrı bir belgedir, eğer tekrar olmadan bir kelime listesi yapılırsa karşısındaki kutu gibi görünecektir. Örnekten anlaşıldığı gibi model, kelime sırası önemli olmadan liste haline getirildi.



Şekil 2.2 Kelime Torbası Örneği

Öyleyse, bir kelimenin vektöre nasıl dönüştürülebileceği, bir belgedeki kelime oluşumunu liste olarak saydığımız basit kelime sayısı örneği ile anlaşılabilir.



Şekil 2.3 Kelime Torbası Vektör Dönüşüm Örneği

Örneğin, "Ateş dört temel elementten biridir." cümlesinin vektör dönüşümü Şekil 2.3'te gösterildiği gibi temsil edilebilir. Yani burada temelde kelime vektöre çevrildi. Aynı yaklaşım izlenerek diğer belgeler vektöre çevrildiğinde Şekil 2.4'teki sonuç elde edilir.

"Ateş, dört temel elementten biridir." = [1,0,1,1,1,0,0,1]
"Su, dört temel elementten biridir." = [0,1,1,1,1,0,0,1]
"Hava, dört temel elementten biridir." = [0,0,1,1,1,0,1,1]
"Toprak, dört temel elementten biridir." = [0,0,1,1,1,1,0,1]

Şekil 2.4 Kelime Torbası Vektör Dönüşüm Örneği

Yukarıda söz edilen yaklaşıma; her seferinde yalnızca bir kelime ele alındığı için verilen isim unigramdır. Benzer şekilde bir yaklaşım olan biagram, bir seferinde iki kelime kullanarak kelime listelerini oluşturur. Örneğin; “Ateş dört”, “dört temel”, “temel elementten” gibi. Trigram ise bir seferde üç kelime kullanılarak belgelerden kelime listelerinin oluşturulmasıdır. Örneğin, “Ateş dört temel”, “dört temel elementten” gibi. Ngram ise bir seferde n kelime kullanarak kelime listesi oluşturmaktır.

CountVectorizer methodu; sklearn kütüphanesine ait olan ve kelimelerin indexlenip her belgedeki kelime sayısı ile bir matrix oluşturan bir fonksiyondur. Matrix işlevini kullanılarak metin belgesini kelime sayısı matrisine dönüştürebilir. Burada üretilen matris seyrek matristir (sparse matrix).

Kelime İndeksleri

```
{'Ateş': 0,
'biridir.': 1,
'dört': 2,
'elementten': 3,
'Hava': 4,
'Su': 5,
'temel': 6,
'Toprak': 7, }
```

Kelime sayısı matrisi (seyrek matris)

```
array([[1,1,1,1,0,0,1,0],
[0,1,1,1,0,1,1,0],
[0,1,1,1,1,0,1,0],
[0,1,1,1,0,0,1,1]], dtype=int64)
```

Şekil 2.5 Kelimelerin indekslenip tarafından seyrek matrise dönüştürülmesi

Yukarıdaki belgede CountVectorizer kullanılarak Şekil 2.5’ te gösterildiği gibi numpy.int64 türünde 4 * 8 seyrek matris elde edilir. Bazı makine öğrenmesi modellerinde seyrek matris kullanılmadığından seyrek matrisin yoğun matrise dönüştürülmesi gerekmektedir (Pantola, 2018). TF-IDF, doğal dil işleme’de kullanılan metin vektörleştirme yöntemlerinden biridir. TF-Idf; kelimenin içinde bulunduğu derlem (corpus) veya veri kümesindeki önemini anlatan Terim Frekansı-Ters Belge Frekansı anlamına gelmektedir. TF-IDF, Terim Frekansı (TF) ve Ters Belge Frekansı (IDF) olmak üzere iki kavramdan oluşur. Terim Frekansı, Şekil 2.6’da gösterildiği gibi kelimenin veri (belge veya derlem) içinde ne sıklıkla tekrarlandığını ifade eder. Her cümle aynı uzunlukta olmadığından, uzun cümlelerde terim frekansı daha yüksek olabilir. Çok büyük belgelerde kelimenin frekansı çok yüksek değerler olabilir. Bu değeri düşürmek için loglanarak formulize edilir.

$$TF = \frac{\text{Terimin belgedeki sayısı}}{\text{Belgedeki toplam kelime sayısı}}$$

Şekil 2.6 Terim Frekansı Formülü

Ters Belge Frekansı, daha seyrek bulunan kelimelerin daha önemli olduğuna dayanan bir kavramdır. Şekil 2.7’de gösterildiği gibi belge sayısının kelimenin geçtiği belge sayısına oranını loglanması ile formulüze edilir.

$$IDF = \log_{10} \frac{\text{Belge sayısı}}{\text{Kelimenin geçtiği belge sayısı}}$$

Şekil 2.7 Ters Belge Frekansı Formülü

TF-IDF, basitçe ifade edilirse terim frekansı ve ters belge frekansının çarpımıdır. Farklı belgelerde kullanılan ortak kelimelerin değerini azaltır. Bir diğer vektörleştirme yöntemi olan Word2vec, iki katmanlı sinir ağı kullanan bir derin öğrenme tekniğidir. Word2vec

girdi olarak büyük bir veri alır (Google veri seti etc.) ve bunu vektör uzayına dönüştürür. Word2vec, basitçe kelimeyi özellik(feature) uzayına yerleştirir, kelimelerin konumları anlamlarına göre belirlenir, yani benzer anlama sahip kelimeler bir arada kümelenir. Google Word2Vec Google veri seti ile önceden eğitilmiş bir tekniktir. En benzer kelimeyi bulmak için kosinüs benzerliğini kullanır.

Kosinüs benzerliği iki vektör arasındaki benzerliği ifade eden bir hesaplama. Kosinüs benzerliği için kelime frekansı önemsizdir, gün kelimesinin bir belgede 100 diğer belgede 500 tane olmasına rağmen iki kelime arasındaki açı çok yakın olacaktır. İki vektörün uzayda arasındaki açı θ olsun, eğer θ sıfıra yakınsa vektörlerin birbirine çok benzediğini ifade eder. Fakat θ , 90 derece ise vektörlerin ortogonal ve birbirleri ile ilgisiz olduğunu anlatır. Eğer θ 180 derece olursa her iki vektörün birbirine zıt olduğunu ifade eder. Kosinüs benzerliği'nin bu formülizasyonu kullanılarak özellik uzayında Google Word2Vec tekniği anlamsal olarak yakın kelimeleri kümelere ayırır ve metin vektörleştirme işlemini gerçekleştirir. Bu yöntemde ön işleme olarak kullanılan Google veri seti çok fazla kelime içermemesine karşın pek çok uzman tarafından kullanılmaktadır. (Pantola, 2018)

3. TWITTER API

Sosyal medya son dönemde pek çok haber kanalından çok daha hızlı ve kolay bir şekilde ulaşılan ve kamuoyunun siyasi, bilimsel, günlük hemen her konuda görüşlerini paylaştığı, kişilerin hem birbirlerine hem kurumlara ulaşmak için aracı olan ve pek çok kişinin artık yaptığı paylaşımlarla ve takipçi sayısının fazla olmasıyla beraber ortaya çıkan ‘influencer’ gibi yeni mesleklere mensup olarak geçimini sağlayıp çeşitli akımlara yön verdiği büyük bir mecra haline geldi. Sosyal medya, doğru yanlış fark etmeksizin çoğu zaman nesnel olmayan haber ve görüşlerle kişilerin hayatlarını ve psikolojik durumlarını etkileyen büyük bir etken halini aldı.

Kişilerin dışarıda yaptıkları aktiviteler, arkadaş ortamları, sosyalleştikleri alanlar bile paylaşmak istedikleri, var olmak istedikleri ve ‘buradayım’ demek istedikleri sosyal medyada birer paylaşım olarak insan hayatına girdi. Yemek yapmak, çay içmek, uyumak yeni bir şeyler almak, arkadaşlar ile vakit geçirmek bile bireylerin sosyal medyada paylaşım yapmak üzerine yaptığı aktiviteler haline geldi. Hemen her anını sosyal medyada paylaşmak o kadar normalleşti ki bireylerin paylaşımları hayatlarına çok uzak kişilerin bile onlar hakkında kimi zaman gerçekçi olmayan bilgiler edinmesine sebep oldu. Sosyal medyada popüler olan kişilere olan özenme duygusu bireylerin kendi kişiliklerini ve dış görünüşlerini sorgulama ve onlar gibi olmak isteme gibi duygular hissetmesine ve bu sebepten psikolojik anlamda kendilerini kötü ve dışlanmış hissetmelerine neden oldu. Öte yandan siyasi rolü olan kişilerin sosyal medya paylaşımları adeta birer siyasi söylev niteliğinde değerlendirilip kimi zaman oy toplamasına veya takipçisi olduğu kişileri kaybetmesine yol açtı.

Sosyal medyanın siyasi etkisi o kadar büyüdü ki, gündeme oturan her konuda başlık olarak sosyal medya üzerinde kişilerin görüşleri sıralanabilir ve kişiler tepkilerini bu başlık altında toplayarak hükümetlere sesini duyurabilir. Sosyal medyanın insan hayatındaki önemi bu derece artınca, akademik alanda artık sosyal medya konusu bilimsel olarak ele alınmaya ve bu konuda çalışma yapılmaya başlandı. Sosyal medya büyük bir veri kaynağı haline geldi.

Twitter bu büyük veri kaynağını kullanıcılara ve bu alanda akademik çalışma yapacak kişilere, sektörde çalışan yazılımcılara belli şartları karşılayıp gerekli izinleri almaları dahilinde kişisel verileri koruma koşulu altında anonim olarak paylaşmaya başladı.

Twitter API, kişilere bu arayüzü kullanarak tweet atma, geçmişe yönelik veya anlık olarak atılan tweetleri; belli başlıklara ait veya belli bir dilde, belli bir lokasyonda olacak şekilde filtreleme olanağı ile sağlamaya başladı. Twitter API, Twitter'ın geliştiricilere özel ürettiği ürünlerden biridir. Twitter API'si, verileri programlı olarak almak ve analiz etmek ve ayrıca Twitter'daki konuşmaya katılmak için kullanılabilir. Bu API, tweetler, kullanıcılar, direkt mesajlar, listeler, eğilimler, medya ve yerler dahil olmak üzere çeşitli farklı kaynaklara erişim sağlar. Twitter API, şu anda desteklenen iki sürümden (TwitterAPI ve TwitterAPI v2) ve farklı erişim katmanlarından oluşmaktadır.

3.1 Twitter API Kullanımı

Twitter API özelliklerini kullanabilmek için öncelikle Twitter'dan yetki alınması gerekmektedir. (Twitter) Bu yetkinin alınması için Twitter'da bir geliştirici hesabının (Twitter Developer Account) açılması ve kullanım alanının Twitter tarafından onaylanması gerekmektedir. Standart ve Akademik Araştırma olmak üzere iki çeşit ürün bulunmaktadır. Standart, iyi bir amaç için, öğrenmek veya öğretmek için bir şeyler geliştirenler de dahil olmak üzere çoğu geliştiricinin varsayılan ürün kanalıdır. Onaylanan tüm geliştiriciler bir Standart Proje oluşturabilir ve kullanabilir. Akademik Araştırma, nitelikli akademik araştırmacılara; tam arşiv arama yetkisi, daha yüksek aylık tweet sınırı ve filtrelenmiş akış ve son arama yetkisi ile gelişmiş filtreleme yetenekleri dahil olmak üzere, yükseltilmiş erişim ve gelişmiş işlevlere erişim sağlar. Twitter, geliştirici hesabı başvurularını çeşitli koşullarla kabul etmektedir, kullanım politikasına uygun olmayan başvurular reddedilmektedir.

Onaylandıktan sonra, standart veya akademik araştırma projesi ve API'ye yönelik tüm isteklerin doğrulanması için kullanılacak bir dizi kimlik bilgisi sağlayacak olan ilişkili bir geliştirici uygulaması oluşturulmalıdır. (Twitter)

3.1.1 Projeler ve uygulamalar

Projeler, Twitter API'nin kullanım amacına göre yapılacak çalışmanın organize edilmesine olanak tanır. Böylece API'ye erişim etkili bir şekilde yönetilebilir ve kullanımını izlenebilir. Her Proje, kimlik doğrulama bilgilerinin (anahtarlar ve belirteçler) oluşturulabileceği bir uygulama içerebilir. “Twitter API”, “Twitter Ads API” veya “Labs” 'a erişimi için herhangi bir uygulamadaki anahtarlar ve belirteçler kullanabilir, ancak yeni “Twitter API v2” servislerini(endpointlerini) kullanabilmek için bir projeyle ilişkili bir uygulamadaki anahtarların ve belirteçlerin kullanılması gerekir. Twitter geliştirici hesabından, projeler Projeler ve Uygulamalar sayfasından görüntülenebilir ve yönetebilir.

Twitter API' nin destek gösterdiği iki tip proje olan Standart Proje ve Akademik Araştırma Projesi arasından kullanılan projeye bağlı olarak ek işlevlerin ve erişim düzeylerinin kilidi açabilir. Örneğin, erişim isteği ile birlikte Akademik Araştırma Projesi ile ilişkili kimlik doğrulama bilgileri iletildiğinde tam arşiv arama servisine (endpoint) erişim sağlanır. Ayrıca, arama tweetleri (search tweets) ve filtrelenmiş akış servisleri (stream endpoints) ile sonuçları filtrelerken gelişmiş işlevlerden yararlanılabilir.

Twitter tarafından onaylanmış bir geliştirici hesabıyla bir Standart Proje oluşturabilir. Bir Akademik Araştırma Projesi oluşturmak için, önce Akademik Araştırma ürün yolunu kullanmak için başvurulması ve onay alınması gerekir. Akademik Araştırma yolu özel, daha yüksek düzeyde erişim içerdiğinden, yalnızca ticari olmayan kullanım için ayrılmıştır. Akademik Araştırma alanı, kamuya açık Twitter verilerine ücretsiz olarak uzmanlaşmış, daha yüksek düzeylerde erişim yoluyla akademik araştırma topluluğunun ihtiyaçlarına hizmet etmek için oluşturulmuştur. Yeni ve mevcut Twitter geliştiricilerinin bu parçaya

eriřim elde etmek için bir Akademik Arařtırma bařvurusu tamamlamaları gerekecektir. Bu uygulamada, akademik arařtırmacının, kimlik bilgileri, arařtırma projesi, Twitter verilerini nasıl kullanmayı planladığı ve çalışmayı nasıl paylaşmayı planladığı sorulur. Twitter, verilen bilgilerin ekipleri dışındaki hiç kimseyle paylaşılmayacağını sözünü vermektedir. Twitter Ekipleri, ticari olmayan kullanım şartları da dahil olmak üzere, her projenin Twitter Geliřtirici Sözleşmesine ve Politikasına uygun olduğundan emin olmak için her uygulamayı dikkatlice inceler. Bu süreç ve bilgilerin, Twitter'daki ve geliřtirici platformundaki kiřilerin güvenliğini ve mahremiyetini korumaya yönelik olduğu ortaya atılır. Onaylandıktan sonra, bařvuruda gönderilen ayrıntılara dayalı olarak bir Akademik Arařtırma Projesi alınmış olur.

3.1.2 Kimlik doęrulama yöntemleri

Twitter API'leri muazzam miktarda veriyi işler. Bu verilerin güvenliğini sağlamak amacıyla çeřitli kimlik doęrulama yöntemleri kullanılmaktadır. Duygu Analizi konusunda çalışma yapmak için veriye ihtiyaç duyulduğunda en sık kullanılan kimlik doęrulama yöntemi OAuth 1.0'a Kullanıcı Bağlamı kimlik doęrulası (User Context Authentication) gerektiren isteklerde bulunmak için API Anahtarı, API Anahtar Sırđı, Eriřim Belirteci ve Eriřim Belirtici Sırđı'nın kullanılması gerekmektedir. Bařka bir kullanıcı adına istekte bulunulması doęrultusunda gereken yetkinin alınması için üç aşamalı OAuth akışının kullanılması gerekir.

Geliřtirici hesabı Twitter tarafından onaylandıktan ve bir proje ve uygulama oluşturulduktan sonra, geliřtirici uygulamada gereken kimlik bilgileri bulunabilir veya üretilebilir. Bu kimlik bilgileri řunlardır,

- API Anahtarı (API Key) aslında bir kullanıcı adıdır ve uygulama adına bir talepte bulunulmasını sağlar.
- API Anahtar Sırđı (API Key Secret) bir řifredir ve uygulama adına bir talepte bulunulmasına izin verir.

- Eriřim Belirteci (Access Token) ve Eriřim Belirteci Sırrı (Access Token Secret) uygulamanın sahibi olan Twitter hesabını temsil eder ve bu Twitter hesabı adına bir talepte bulunulmasına izin verir.
- Tařıyıcı Belirteci (Bearer Token) uygulamanın kendisini temsil eder ve OAuth 2.0 Tařıyıcı Belirteci kimlik doęrulaması gerektiren isteklerin doęrulanmasına olanak tanır.

Oluřturulan tüm bu anahtarların ve belirteçlerin belirli bir kullanım süresi olmadığından, yani yeniden oluřturulana kadar aynı bilgiler kullanılabildiğinden, gereken kimlik bilgileri alındıktan sonra bilgi güvenlięi için kullanıldıęı uygulamada gereken güvenlik önlemlerinin alınması gerekmektedir. Anahtarlar ve belirteçler kullanılan hesabın geliřtirici portalında yalnızca bir kez görüntülenebilir, bu nedenle bu kimlik bilgileri oluřturulduęu anda eklenecek uygulamaya kaydedilmez.

Geliřtirici portalında daha önceden üretilmiř parolalar görüntülenememektedir. Eęer anahtarlar ve belirteçler bir sebepten kaybedilirse veya yanlış yazılırsa, geliřtirici portalından yeniden oluřturulması gerekir ve daha önce oluřturulan kimlik bilgileri geçersiz hale gelir. Bu da eski kimlik bilgilerinin kullanıldıęı her yerde güncelleme yapılması gerektięi anlamına gelir. Twitter API'nin iki farklı sürümünde farklı doęrulama yöntemleri gerektiren çeřitli katmanlar bulunmaktadır. Bunlardan bazıları, “Standart v1.1”, “Premium v1.1” ’tir. Gereken kimlik bilgileri bu katmanlara göre deęiřkenlik göstermektedir (Twitter 2021a). Twitter Geliřtirici Hesabı’na bařvurduktan ve gerekli onaylar alındıktan sonra, oluřturulan proje ve uygulamanın verdięi kimlik bilgileri kullanılacak uygulamaya eklendikten sonra, API'nin saęladığı servislerin nasıl kullanılacaęına dair açık kaynaklı bir örnek kod bulunmaktadır. (Twitter 2021b) Tweetleri çekebilmek veya iřleyebilmek için Twitter ‘ın çeřitli kütüphaneleri bulunmaktadır. Tweepy bu iřlemleri saęlayan Python’da yazılmıř kütüphanelerden biridir.

3.1.3 Tweepy kullanımı

Tweepy kütüphanesi, hem OAuth 1a (uygulama kullanıcısı) hem de OAuth 2 (yalnızca uygulama) kimlik doğrulamasını desteklemektedir. Kimlik doğrulama işlemi, tweepy kütüphanesine ait AuthHandler sınıfı tarafından gerçekleştirilir. OAuth 2, bir uygulamadan API istekleri yapılarak gerçekleştirilen ve kullanıcı bağlamı gerektirmeyen bir kimlik doğrulama yöntemidir. Herkese açık bilgilere salt okunur erişim gerektiğinde bu yöntem kullanılır. Uygulama oluşturulduktan ve bir tüketici anahtarı (api_key veya consumer_key) ve sırrı (api_secret veya consumer_secret) alındıktan sonra Tweepy aracılığıyla şekil 3.1'deki gibi bir AppAuthHandler örneği oluşturulur. (Yener 2020)

```
consumer_key="*****"  
consumer_secret="*****"  
access_key="*****"  
access_secret="*****"  
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)  
auth.set_access_token(access_key, access_secret)  
api = tweepy.API(auth,parser=tweepy.parsers.JSONParser(),)
```

Şekil 3.1 Tweepy kimlik doğrulama örneği

Duygu analizi konusunda, Twitter API, sosyal medya mecralarından biri olan Twitter'daki verileri işlemek için büyük miktarda veri çekimine olanak tanıyan bir arayüzdür. Tweepy ise çoğunlukla python dilinde yapılan çalışmalarda kullanılan Twitter'ın sağlamış olduğu açık kaynak kodlu bir kütüphanedir. Tweepy ile kimlik doğrulama işlemi yapıldıktan sonra ihtiyaç duyulan veriyi elde etmek için iki farklı method kullanılır. Bunlar Tweepy kütüphanesine ait “search” ve “stream” methodlarıdır.

Twitter Arama API'si (Twitter Search API) Twitter'ın verilerini bir kullanıcı adı ara veya arama aracılığıyla sorgulamayı sağlar. Hali hazırda var olan bir veri kümesine erişilmesine olanak tanır. Kullanıcılar belli kriterlerle (anahtar kelimeler, kullanıcı adları, konumlar vb.)

eşleşen tweetleri talep edebilir. Şekil 3.2’de Tweepy search methodu kullanım örneği gösterilmiştir.

```
search_words = "bisiklet"
date_since = "2019-11-16"
tweets = tweepy.Cursor(api.search,
                        q=search_words,
                        lang="tr",
                        since=date_since).items(5)
```

Şekil 3.2 Tweepy search methodu kullanım örneği

Twitter Yayın API’si (Twitter Streaming API) gerçek zamanlı olarak Twitter’ın verilerini tweetler atıldığı anda istek atan kullanıcıya iletir. Hazır veri kümesinin içinde sorgu yapıp göndermez (Anonymus, 2013). Şekil 3.3’te Tweepy’ ye ait “stream” methodunun bir kullanım örneği gösterilmiştir.

```
streamListener = StreamListener()
stream = tweepy.Stream(auth=api.auth, listener=streamListener, tweet_mode='extended', count=5,)
tags = ["korona", "covid-19", "Covid19", "Covid-19", "Corona Virus", "pandemic", "pandemi", "COVID-19"]
stream.filter(track=tags, languages = ["tr"], encoding="utf-8")
```

Şekil 3.3 Tweepy stream methodu kullanım örneği

3.1.4 Tweet İşaretleme

Doğal Dil İşleme alt konularından olan Duygu Analizi konusunda veri oluşturmak ile ilgili olarak Twitter üzerinden nasıl veri oluşturulacağından yani nasıl tweet çekileceğinden yukarıda bahsedildi. Duygu analizi konusunda çalışma yapmak için denetimli öğrenme kullanan makine öğrenmesi algoritmalarından yararlanılabilir. Bu durumda ham veriden işaretli veri oluşturmak gerekmektedir.

Veriler duygu analizinde Pozitif, Negatif veya Nötr olarak işaretlenebilir. Denetimli Öğrenme’ de işaretleme yapmak için farklı ve daha hızlı yöntemler olsa da Duygu Analizi konusunda genellikle bu işaretleme manuel olarak yani her tweetin pozitif, negatif veya nötr olduğu bir birey tarafından değerlendirilip işaretlenerek gerçekleştirilir. Şekil 3.4’ te gösterildiği gibi tweetler birer birer değerlendirilir ve tablo oluşturularak 3 farklı duyguda etiketlenir. Duygu Analizi çalışmalarındaki işaretli veriler dil yapısına ve yorumlanma şekline göre özgünlük taşımaktadır. Bazı çalışmalarda veri sadece Pozitif ve Negatif olarak ele alınmaktadır. Makine Öğrenmesi algoritmalarının gösterilen bu veriyi öğrenmesi ve yeni karşılaştığı verilerde bu durumlar içinde tahminleme yapması beklenmektedir.

Duygu	Tweet
Pozitif	Güneşli ve çok güzel bir gündü.
Negatif	Güneşli havaları hiç sevmem.
Nötr	Hava bugün güneşli idi.

Şekil 3.4 İşaretli veri örneği

4. MAKİNE ÖĞRENMESİ ALGORİTMALARI

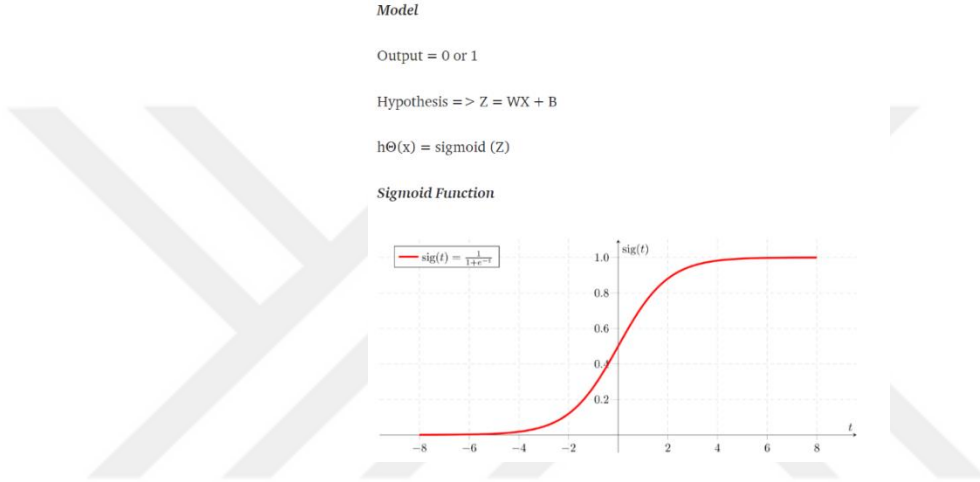
Duygu Analizi, Makine Öğrenmesi alanında adından sıkça söz ettiren ve üzerinde pek çok çalışma yapılmış bir konudur. Günümüzde bu konu farklı dillerde Denetimli Öğrenme yöntemi ile pek çok başarılı sonuç elde edilen ve reklamlara yansiyacak derecede insan hayatına etki etmiş ve dolayısıyla yazılım sektörüne fazlasıyla girmiştir. Bu alanda kullanılan pek çok farklı Makine Öğrenmesi algoritması bulunmaktadır. Aşağıda bu algoritmalar arasından Denetimli Öğrenme ile kullanılan bazı örnekleri sırasıyla gösterilmiştir.

4.1 Logistik Regresyon (Logistic Regression)

Logistik Regresyon, yirminci yüzyılın başlarında biyolojik bilimlerde sonrasında ise pek çok sosyal bilim uygulamasında kullanıldı. Logistik Regresyon çıkış verisi bir kategoriye ifade ettiğinde kullanılır. Örnek verilirse, bir e-posta'nın spam olup olmaması kategorilerinden birini sonuç olarak verir. Linear Regresyon'dan farkı ise Logistik Regresyon doğrusal sınıflandırmada kullanılırken, linear regresyon kategorik verilerde kullanılamaz, girdi ve çıktı arasında ilişki kurarak bir sonraki ayın faiz oranı gibi tahminleme yapmayı sağlar. (Hatipoğlu 2018)

Logistik Regresyon temel olarak Şekil 2.11'de belirtilen formüle dayanır. Duygu Analizi konusu kapsamında verilen cümlelerin pozitif veya negatif olarak sınıflandırılması söz konusudur. Logistik Regresyon'un üç farklı tipi vardır. Veri İşaretleme işlemlerinin veya veri ön işlemlerinin bu tiplere göre yapılması gerekmektedir. İkili Logistik Regresyon; kategorik yanıtın yalnızca iki olası sonucu vardır. Örnek: spam veya değil. Bu logistik regresyon tipinin kullanılması durumunda yalnızca iki kategorili (Pozitif ve Negatif gibi) tweetlerin verilmesi gerekir. Bir diğeri olan Çok Terimli Logistik Regresyon, sıralama olmadan üç veya daha fazla kategoride sonuç verebilir. Örnek: Hangi yiyeceğin daha çok tercih edildiğini tahmin etmek (Sebzeli, Sebzeli Olmayan, Vegan gibi) olabilir.

Bu veri tipinde duygu analizi verileri üç kategoride (Pozitif, Negatif, Nötr) olacak şekilde verilebilir. Son olarak, Sıralı Lojistik Regresyon; sıralama ile üç veya daha fazla kategoride sonuç verir. Örnek: Bir filmin beğeni olarak 1'den 5'e kadar derecelendirmesi verilebilir. (Swaminathan 2018)



Şekil 4.1 Basit Lojistik Regresyon Modeli (Swaminathan 2018)

4.2 Naive Bayes Sınıflandırması (Naive Bayes Classification)

Naive Bayes, Bayes Teoremi'ne dayanan, Makine Öğrenmesi alanında sıkça kullanılan ve Duygu Analizi konusunda başarılı algoritmalarından biridir. Naive Bayes, tahmine dayalı bir modellemeye sahip bir sınıflandırma algoritmasıdır. Makine öğrenmesi, eldeki verilerle en olası sonucu elde etmek için en iyi seçimi yapmak ile ilgilenir. Hipotez, sınıflandırma problemlerinde verinin atanacağı sınıfı ifade eder. En olası hipotezi seçebilmek için, veriden elde edilen önceki bilgiler kullanılabilir. Bayes Teoremi basitçe, önceki bilgiler ışığında hipotez olasılığının hesaplanması için iyi bir yoldur.

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

Şekil 4.2 Bayes Teoremi'nin formülü

Bayes teoremi, Şekil 4.2’ deki formülle ifade edilir. Formülde görülen $P(h|d)$ sonraki olasılığı (posterior probability), d veriyi, h ise hipotezi ifade etmektedir. Sonuç ise, d ’ ye göre h ’nin yani veriye göre hipotezin olasılığıdır. Hipotezin doğru olduğu varsayıldığında verinin olasılığı $P(d|h)$ ile ifade edilir. $P(d)$ ve $P(h)$, birbirilerinden bağımsız olarak sırayla verinin olasılığı ve hipotezin doğru olma olasılığıdır. Buna önceki olasılık (prior probability) denir (Brownlee 2020).

Bisiklet sürmek için hava koşullarını tanımlayan Şekil 4.3’ teki gibi bir veri kümesinde; hava koşulları göz önüne alındığında, her bir satır, bisiklet sürmek için uygun ("Evet") veya uygun değil ("Hayır") olarak koşulları sınıflandırır. Bu veri kümesinde, özellikler 'Görünüm', 'Sıcaklık', 'Nem' ve 'Rüzgarlı'dır (Anonymus 2020a).

	Görünüm	Sıcaklık	Nem	Rüzgarlı	Bisiklet Sür
0	Yağmurlu	Sıcak	Yüksek	Yanlış	Hayır
1	Yağmurlu	Sıcak	Yüksek	Doğru	Hayır
2	Bulutlu	Sıcak	Yüksek	Yanlış	Evet
3	Güneşli	Hafif	Yüksek	Yanlış	Evet

Şekil 4.3 Naive Bayes Örnek Veri Kümesi

Özellik değerlerini içeren vektörlerden oluşan matris özellik matrisi (feature matrix)’ dir. Özellik matrisindeki her bir vektörün tahminlerinden oluşan vektör ise yanıt vektörüdür (response matrix). Şekil 4.3’ teki veri kümesi, 'Görünüm', 'Sıcaklık', 'Nem' ve 'Rüzgarlı' özellik değerlerinden oluşan vektörleri içeren özellik matrisi ve ‘Bisiklet Sür’ değerini içeren yanıt vektöründen oluşur.

Hiçbir özellik ikilisinin bağımlı olmadığını varsayıldığından her özellik eşit ağırlıkta kabul edilir. Örnek verilirse, nemli olmasının görünüm ile bir ilgisi yoktur. Bu varsayımla Bayes Teoremi'nin birleşmesi sonucu ortaya çıkan Naive Bayes formülü Şekil 4.4' te gösterilmektedir. Bu varsayımlar gerçek dünyadaki durumlarda her ne kadar doğru olmasa da pratikte başarılıdır (Anonymus. 2020a).

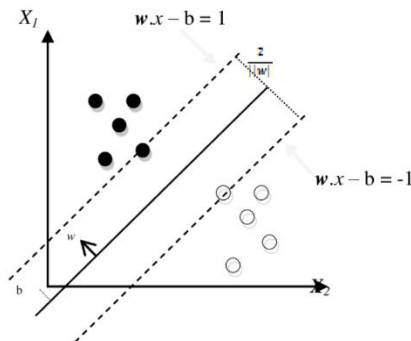
$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)}$$

Şekil 4.4 Naive Bayes Formülü (Anonymus. 2020a)

Duygu analizi yapılırken verinin uğradığı ön işlemler sonrasında elde edilen veri Naive Bayes sınıflandırması ile Pozitif, Negatif veya Nötr olarak sınıflandırılabilir. Bu sınıflandırma kullanılarak duygu analizi yapılacak yazıdan çıkarılan bir özellik matrisi ile duygu olarak ifade ettiğimiz yanıt vektörü bulunur.

4.3 Destek Vektör Makineleri (Support Vector Machines)

SVM, verilen uzayda verileri farklılaştıran sınırı belirlemek için hiper düzlem (hyper plane) bularak, bir grubu veya etiketi tahmin etme işlemi olan sınıflandırma veya sürekli bir değeri tahmin etme işlemi olan regresyon problemlerinde kullanılabilen denetimli bir makine öğrenme algoritmasıdır.



Şekil 4.5 Ayırma alt düzlemi örneği (Reddy 2018)

Şekil 4.5 'te iki sınıf arasında en uygun ayırma hiper düzlemi gösterilmiştir. SVM, Hiper düzlem çizebilmek için "Kernels" adı verilen matematiksel fonksiyonları kullanır. Kernel tipleri, doğrusal, sigmoid, RBF, doğrusal olmayan, polinom vb. olabilir. Doğrusal olmayan problemler için ayarlama parametresi olarak kullanılan Kernel "RBF"; veriler hakkında ön bilgi olmadığında kullanılan genel amaçlı bir kernel tipidir. "Kernel doğrusal" (kernel linear) ise Kernel "RBF"'nin aksine doğrusal ayrılabilir problemlerde kullanılır. SVM ile Duygu Analizi çalışması yapılırken verilerin kaç kategoride sınıflandırılacağına göre kernel seçimi yapılmalıdır. Sadece pozitif ve negatif gibi kategorilerde ikili sınıflandırma için doğrusal kernel parametresi kullanılırken, nötr gibi farklı kategoriler eklenip çoklu sınıflandırma yapılması için "Kernel RBF" kullanılır. (Reddy 2018)

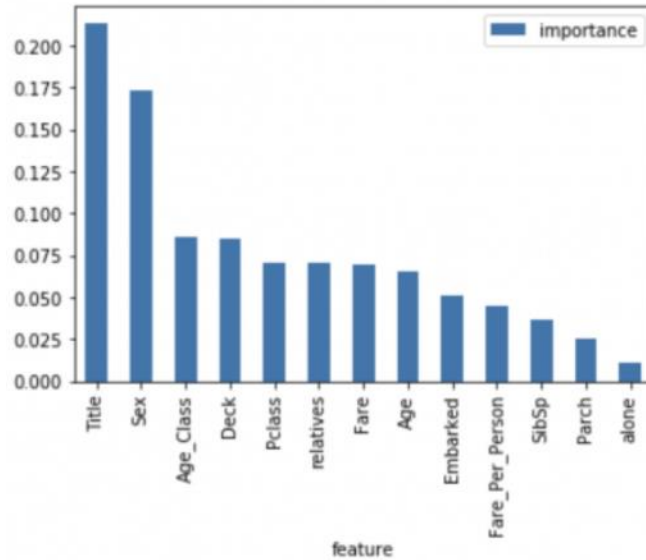
4.4 Rastgele Orman Algoritması (Random Forest Algorithm)

Rastgele orman, torbalama (bagging) yöntemiyle eğitilmiş karar ağaçlarından orman oluşturan bir denetimli makine öğrenmesi algoritmasıdır. Torbalama yönteminin genel fikri, öğrenme modellerinin bir kombinasyonunun genel sonucu artırmasıdır. Rastgele orman algoritması daha doğru tahminler elde edebilmek için karar ağaçları oluşturup bunları birleştirir. Sınıflandırma ve Regresyon problemlerinin çoğunda kullanılabildiğinden Duygu Analizi problemlerinde de sıkça kullanılan bir algoritmadır.

Rastgele Orman Algoritması, karar ağaçlarını fazladan rastgelelik ekleyerek birleştirir. Dğümleri alt dğümlere ayırma işlemini gerçekleştirirken en önemli özelliği aramaz. Bunun yerine rastgele bir özellik alt kümesi (subset of features) arasında en iyi özelliği arar. Bu, genellikle daha iyi bir modelle ve geniş bir çeşitlilikle sonuçlanır. Rastgele Orman Algoritması ile çalışılırken karar ağaçlarının (decision tree) yaptığı gibi olası en iyi eşikleri (thresholds) aramak yerine her özellik için ek olarak rastgele eşikler (thresholds) kullanılarak ağaçlar daha da rastgele hale getirilebilir. Rastgele orman algoritması'da tahmin üzerindeki her bir özelliğin diğerlerine göre önemini ölçmek oldukça kolaydır.

Eğitimden sonra her özellik (feature) için bu puan otomatik olarak hesaplar ve sonuçları ölçeklendirir, böylece tüm önemlerin toplamı bire eşit olur. Şekil 4.6'da örnek bir özellik önem grafiği gösterilmiştir.

Bir karar ağacında her bir düğüm bir testi ifade eder. Bu ağaca ait her bir dal bu testin sonucunu temsil ederken alt düğümü olmayan yapraklar ise yapılan sınıflandırma sonucunda kategoriye yani verinin ait olduğu sınıfı ifade eder. Özelliğin önemine bakarak, tahmin sürecine yeterince katkıda bulunmayan (veya bazen hiçbir şey yapmayan) hangi özelliklerin kaldırılacağına karar verilir ve bu işlem oldukça önemlidir. Bunun sebebi makine öğreniminde genel bir kuraldır. Özellik sayısı arttıkça, modelde aşırı uyum (overfit) gerçekleşme ihtimali o kadar artar (Donges 2019). Rastgele Orman Algoritması birden fazla kategoride sınıflandırma için kullanılabilir böylece 3 farklı sınıf (pozitif, negatif, nötr) için yapılan Duygu Analizi çalışmaları için uygundur.



Şekil 4.6 Özellik önem grafik örneği (Donges 2019)

4.5 Stokastik Gradyan İniş Algoritması (Stochastic Gradient Descent Algorithm)

Gradyan İniş, Makine Öğrenimi ve Derin Öğrenmede popüler bir optimizasyon tekniğidir. Öğrenme algoritmalarının tamamında olmamakla birlikte çoğunda kullanılabilir. Bir fonksiyonun eğimi olarak tanımlanan Gradyan, bir değişkenin geçirdiği değişiklikler karşısında bir başka değişkenin geçirdiği miktarını hesaplar. Başka bir değişkenin değişikliklerine yanıt olarak bir değişkenin değişim derecesini ölçer. Gradyan İniş'i maliyet (cost) fonksiyonunun minimum değerini bulmak amacıyla yinelemeli olarak çalışır. 'Stokastik' rastgele bir olasılığa sahip süreç veya sistem olarak tanımlanır. Stokastik Gradyan İnişinde, her bir yineleme için tüm veri seti yerine rastgele birkaç örnek seçilir. Toplu Gradyan İniş (Batch Gradient Descent) gibi tipik Gradyan İniş algoritmalarında her iterasyonda parti (batch, yani her iterasyonda veri setinden alınan örnek sayısı olarak) olarak tüm veri seti alınır.

Bununla birlikte, tüm veri kümesini kullanmak minimuma daha az gürültülü ve daha az rastgele bir şekilde ulaşmak için gerçekten yararlı olsa da, veri kümesi büyüdükçe işlem maliyeti artar. Bu problem Stokastik Gradient Descent ile çözülmüştür.

SGD algoritmasında, her yinelemeyi gerçekleştirmek için bir grup veri (batch) kullanılır. Bu veri grubu her yinelemede rastgele karıştırılıp tekrar seçilir (Anonymus 2020b).

SGD, bir optimizasyon algoritmasıdır. SGD Sınıflandırıcı (SGD Classifier) ise, basit bir öğrenme rutinine sahiptir. Bu rutin, sınıflandırma işlemini gerçekleştirmek için çeşitli kayıp fonksiyonlarını (loss functions) ve cezaları destekler (Anonymus 2021b). SGD Sınıflandırıcı çok sınıflı sınıflandırma yapmak için birden fazla ikili sınıflandırmayı birleştirebilir (Anonymus 2020c). Bu sebeple çoklu kategoride (pozitif, negatif, nötr) Duygu Analizi yapılmasına olanak tanır.

4.6 Uzun Kısa Süreli Bellek (Long Short-Term Memory)

Long Short Term Memory, 1997 yılında Hocheriter & Schminhuber tarafından piyasaya sürülen LSTM olarak da bilinir.

LSTM, uzun vadeli bağımlılığı kavrayabilen bir RNN ağı türüdür. Günümüzde duygu analizi, konuşma tanıma, metin sınıflandırması gibi konularda yapılan çalışmalarda sıkça rastlanmaktadır. LSTM'den önce var olan, ileri beslemeli ağlar, (feed-forward-networks) girişlerin bir ağırlıkla çarpıldığı ve sonra buna bias eklendiği ve son katmandan çıkış değerini aldığı ağlardır. Bu ağlarda karşılaşılan problem ise sıralı verilerde kullanılamaması ve bellek depolaması yapılmamasıdır. Bu tür bir ağın girişi ve çıkışı bile sabittir. Bu tür ağları Hisse Senedi Fiyatı tahmini ve benzeri sorunlar için kullanamayız.

Tekrarlayan Sinir Ağları'nın (Recurrent Neural Networks) tanıtılmasının nedeni budur. RNN, sıralı / zaman serisi verilerini yakalayabilecek şekilde tasarlanmıştır. RNN' de bir önceki durumun (w_1) girdisi ile ilişkili ağırlık ile aynı durumun çıktısı ile ilişkili ağırlık çarpılır ve yeni durumu elde etmek için Tanh fonksiyonundan geçirilir. Sonrasında çıktı vektörünü elde etmek için; yeni durum Tanh fonksiyonunun bir çıktısı ile çarpılır. RNN'de derin ağlar tercih edilmez. Ancak RNN, modelin öğrenmesine yardımcı olmayan ağırlıklarda çok önemli değişiklikler oluşması durumunu ifade eden kaybolan bir gradyan probleminden (vanishing gradient problem) muzdariptir. LSTM, bu soruna çözüm getirilmek amacıyla tanıtılmıştır (Dwivedi 2020).

5. LİTERATÜR

Yapay Zeka'nın başlangıcı oldukça eski tarihlere dayanmasına rağmen, Doğal Dil İşleme başlangıcı çok eski tarihlere dayanan ve günümüzde ise Yapay Zeka'nın alt dalı Makine Öğrenmesi alanında ele alınan bir konudur. Doğal Dil İşleme ilk olarak Alan Turing (1950) tarafından yayınlanan ve Yapay Zeka konusunda ufuk açıcı bir makale olan ve Turing Testi olarak bilinen makalesini yayınladı. Turing, makinenin görevini ve zeka kriterini doğal dilin otomatik olarak yorumlanmasını ve üretilmesini olarak belirlemişti. Fakat henüz o zamanlar, Doğal Dil İşleme başlığı altında incelenmemiştir. Sonrasında Doğal Dil İşleme (NLP) adı farklı isimlerle ve yaklaşımlarla, Sembolik NLP (Symbolic NLP), İstatiksel NLP (Statistical NLP) ve günümüzde var olan Sinirsel NLP türleri olarak tarihsel biçimde sıralanmaktadır. Sembolik NLP'nin önermesi, John Searle'nin (1980) Çin Oda Deneyi'ni ortaya attığı Minds, Brains, and Programs adlı 1980' de yayınladığı makalesine göre, bilgisayara bir dizi kural örneğin bir Çince öğrenme klavuzu verildiğinde NLP görevlerinin bilgisayar tarafından taklit edilmesidir.

1980'lerin sonuna kadar karmaşık kurallar ile oluşturulmuş doğal dil işleme sistemleri kullanılırken, hesaplama gücündeki sürekli artış ve Makine Öğrenmesi algoritmalarının ortaya çıkması ile birlikte Doğal Dil İşleme alanında büyük bir gelişme yaşandı. 1980lerin sonlarından itibaren bu algoritmalarının kullanılması ile oluşturulan sistemler İstatiksel NLP başlığı altında yer aldı. İstatistiksel yöntemler kullanarak otomatik konuşma tanıma Bahl ve diğerlerinin (1983) yayınladıkları "A Maximum Likelihood Approach to Continuous Speech Recognition," makalesi ile büyük ses getirmişlerdir. Makaleye göre, konuşma tanıma görevi için maksimum olasılıkla kod çözme (maksimum likelihood decoding) formülasyonu oluşturulmuştur. Çalışmada konuşma tanıma görevinde kullanılması amacıyla bir dizi istatistiksel model açıklanmıştır. 1990 yıllarına ait bir diğer istatistiksel çalışma, Peter Brown ve diğerlerinin (1990) yayınladıkları A Statistical Approach to Machine Translation bu alanda yapılmış önemli araştırmalardan biridir. Makalede, Fransızca'dan İngilizce'ye Bayes Teoremi kullanılarak istatistiksel bir çeviri yaklaşımı sunulmuştur.

Literatürde NLP alanında Makine Öğrenmesi algoritmaları ile ilgili pek çok çalışma bulunmaktadır. Günden güne gelişen Makine Öğrenmesi algoritmaları ve hızlı bilgisayarlar ile yapay zeka ağlarının bu alanda kullanılabileceği ortaya atılmıştır. Gelişen Makine Öğrenmesi algoritmalarıyla birlikte Derin Sinir Ağları (Deep Neural Network) kullanarak oluşturulan modeller yaygınlaşmaya ve NLP alanında kullanılmaya başlandı. Neural NLP bu ağlar ile kullanılan Doğal Dil İşleme Sistemleri'ni ifade etmektedir. NLP alanında yapay sinir ağları kullanılarak, Martinez (1995) tarafından geliştirilen ve çok ses getirmiş bir çalışma olan NALPRONN (Natural Language Processor with Neural Networks) modeli, Neural NLP alanındaki kilometre taşlarından biridir. NALPRONN sistemi, çok modlu çok dilli bir bilgisayar arayüzüdür (interface). Yapay nöron ağlarından oluşan bir sistemdir. Bu sistem, giriş çıkış gösterimleri, giriş alt sistemi, çıkış alt sistemi, sözlük alt sistemi ve monitör alt sistemi gibi geri yayılım ağını içeren işleme modüllerine sahiptir. Aynı zamanda özellik haritası ağlarına (feature mapping networks) sahip olan bellek modülleri bulunmaktadır. Sisteme göre, görevler modüller tarafından gerçekleştirilir. Bu modüller aynı veri ile birbirinden bağımsız olarak eğitilmiştir. NALPRONN ile yapay sinir ağları kullanılarak genelleştirilmiş bir NLP sistemi ortaya konulmuştur.

Yapay sinir ağlarının kullanımına yönelik bu artışa katkıda bulunan makalelerden biri 2002 yılında yayınlanan “Natural Language Processing with Neural Networks” adlı makaledir. (Ma 2002). Makalede, öğrenmeye dayalı doğal dil işleme araştırmalarının ilerlediği bir zamanda sinir ağlarının (neural networks) bu alanda daha çok kullanılması gerektiğine değinilmiştir. Sinir Ağları'nın denetimli veya denetimsiz öğrenme yöntemleri ile hata tespiti (error detection) konuşma parçası (part of speech) gibi NLP görevlerindeki başarısı ortaya konulmaya çalışılmıştır.

Günümüzde Sinir Ağları'na pek çok makine öğrenmesi algoritmasının kullanıldığı yapay zeka alanında olduğu gibi NLP alanında da sıkça rastlanmaktadır. Makale Yapay Sinir Ağı kullanılarak bir NLP sistemi oluşturmanın mümkün olduğunu ortaya koyar.

Bu tezde NLP görevleri denetimli öğrenme kullanılarak gerçekleştirilmeye çalışılmıştır. Bu sebeple literatürdeki denetimli öğrenme ile yapılan çalışmaların incelenmesine ağırlık verilmiştir. Denetimli öğrenme kullanılarak NLP'nin Duygu Analizi görevini ele alan çalışmalardan biri olan, Yi ve diğerlerinin (2003) yayınladığı “Sentiment Analyzer: extracting sentiments about a given topic using natural language processing techniques” isimli makalesidir. Makalede veri oluştururken dijital kamera ve müzik inceleme olmak üzere iki konu tanımlanmıştır. Bu iki veri grubu altında toplanan belgeler web taramalarından toplanan web sayfalarından rastgele seçilen belgelerdir. Bu belgeler kartıştırılarak konu başlıklarının altına rastgele konulmuştur. Her konu için iki farklı etiketleme yapılmıştır. Konu ile ilgili belgeler (D+) olarak işaretlenirken konu dışı belgeler (D-) olarak işaretlenmiştir. Bu veri sistemi ile beraber Mixture model ve Likelihood testi gibi çeşitli feature selection algoritmaları kullanılmıştır.

Tez kapsamında yazı sınıflandırma için araştırılan denetimli öğrenme kullanılarak gerçekleştirilen çalışmalardan biri diğeri Liu ve diğerlerinin (2010) yaptıkları SVM 'in kNN Desicion Tree ve Naive Bayesian methodları ile karşılaştırılması üzerinedir. Çalışmada, SVM tabanlı bir sınıflandırma modeli önerilmiştir. Verilen diğer algoritmalarla yapılan deneylerin sonucunda SVM sınıflandırıcısının F1 değerinin %86,26'nın üzerine çıktığı ortaya konulmuştur. Denetimli Öğrenme kullanılarak yapılan bir başka Duygu Analizi çalışması Shi ve Li'nin 2011 yılında yayınlamış olduğu “A Sentiment Analysis Model For Hotel Reviews Based on Supervised Learning” isimli makalesidir. Kullanıcıların otel incelemeleri ele alınarak bir makine öğrenmesi yaklaşımı ile duygu sınıflandırılması yapılmaya çalışılmıştır. Makalede denetimli öğrenme algoritmalarından olan SVM kullanılmıştır. (Shi ve Li 2011)

Literatürde NLP'ye dair çok fazla makale bulunmasına karşın İngilizce dilinde yapılan çalışma sayısı Türkçe'ye oranla bir hayli fazladır. Fakat son yıllarda Türkçe dilinde yapılan çalışma sayısı artmıştır. Kaya ve diğerlerinin (2012) Türkçe siyasi haberler üzerine Duygu

analizi üzerine yaptıkları çalışmada, farklı Türk haber sitelerindeki köşe yazılarında yer alan siyasi haberler toplanmıştır.

Denetimli Öğrenme kapsamında olan dört algoritma Duygu Sınıflandırması için karşılaştırılmıştır. Bunlar Naïve Bayes, Maximum Entropy, SVM ve karakter tabanlı N-Gram Dil Modeli'dir. Ampirik bulgulardan Maksimum Entropi ve N-Gram Dil Modelinin SVM ve Naive Bayes'ten daha iyi performans gösterdiğini gözlemlenmiştir. Farklı özellikler kullanılarak, tüm yaklaşımların %65 ila %77 doğruluk oranlarına ulaştığı ortaya konulmuştur.

Akba (2014) tarafından yapılan çalışmada film değerlendirmeleri ile eğitilen modeller üzerinden %83.9'a varan F1 skor elde edilmiştir. Sonuçlar Information Gain ve Chi Square metrikleri ile ölçülmüştür. Çalışmada veriler oluşturulurken ön işlem olarak Zemberek; verileri sınıflandırmak için ise SVM kullanılmıştır. Çoban ve diğerleri (2015)'nin yaptıkları çalışmada Twitter üzerinden alınan tweetlerin emojiye göre etiketlenip Bag of words ve N-gram model olmak üzere iki farklı öz nitelik çıkarma yöntemi kullanarak çeşitli makine öğrenmesi algoritmaları ile sınıflandırma işlemi yapılarak %66'ya varan doğruluk elde edilmiştir. Karamollaoğlu ve diğerleri (2018) tarafından yapılan çalışmada, çeşitli internet sitelerinden toplanan kullanıcı yorumlarına Lexicon-Based yöntemi kullanılarak duygu analizi süreçleri uygulanmış; sınıflandırma ve duygu analizi süreci ortalama olarak %80 başarı oranı ile gerçekleştirilmiştir. Bir başka Türkçe ile yapılan çalışma ise, Bozyiğit ve diğerleri (2019) tarafından yayınlanan Türkçe siber zorbalık tespiti'dir. Çalışma kapsamında yapay sinir ağları kullanılmıştır. Türkçe Doğal Dil İşlemi için hali hazırda mevcut kütüphaneler kullanılmamıştır. Çalışmada, "Zararlı Terimler" isimli bir liste oluşturularak Levenshtein algoritması ile terim ile girdi arasındaki yanlış yazımlar düzeltilmeye çalışılmıştır. Metin madenciliği bölümü için bu tezde de bahsedilen TF-IDF kullanılmıştır. Yapay Sinir ağı için iki gizli katman sayısı öngörülmüştür. Çalışmanın sonucunda %91 F1 oranında skor elde edilmiştir.

Pervan (2019) tarafından yapılan web siteleri üzerinden toplanan Türkçe kullanıcı yorumları üzerinde word2vec modeli kullanılarak LSTM ve farklı CNN ağları ile gerçekleştirilen çalışmada LSTM’de %94’e varan doğruluk değeri elde edilmiştir. Türkçe üzerine yapılan NLP çalışmalarından biri de Rumelli ve diğerleri (2019) tarafından yayınlanmış olan, Makine Öğrenmesi algoritmalarıyla gerçekleştirilen bir Duygu Analizi çalışmasıdır. Çalışma sırasında Türkçe Doğal Dil İşleme üzerine hazırlanmış Zemberek (Akin ve Akin 2019) gibi açık kaynaklı kütüphanelerden yararlanılmıştır. Bir e-ticaret web sitesi üzerinden alınan veriler kullanıcıların verdikleri puanlara göre negatif, nötr ve pozitif olarak işaretlenmiş ve Naive Bayesian, Random Forest, SVM gibi Makine Öğrenmesi algoritmalarıyla eğitilmiştir. Çalışmanın sonucunda %73.8 skor elde edilmiştir. Çöltekin (2020) tarafından yayınlanan ‘A Corpus of Turkish Offensive Language on Social Media’ isimli sosyal medyadaki Türkçe saldırgan dilinin tespitine yönelik bir çalışmadır. Çalışmada %19’u saldırgan dil olarak etiketlenmiş mesajlardan oluşan bir veri seti kullanılmıştır. Oluşturulan sınıflandırıcılar saldırgan tweetleri belirlemede %77.3 F1 skor elde etmiştir. Son yıllarda yapılan bir diğer Türkçe çalışma ise Kirelli ve Arslankaya (2020) tarafından gerçekleştirilen kişilerin küresel ısınma ve iklim değişikliği konularına dair duygu analizidir. Literatürde Türkçe üzerine yapılan Duygu Analizi çalışmalarından bazılarının doğruluk değerleri Şekil 5.1’de gösterilmiştir. Yao ve Guan (2020) yayınladıkları makalede, gelişmiş bir NLP yöntemi önermektedir. Bu yöntem LSTM yapısına dayanmaktadır. Çalışmada, Temel ve diğer LSTM ile karşılaştırıldığında, geliştirilmiş yöntem, Wall Street Journal veri setinde daha iyi F1 skoru sonuçlarına sahiptir, Ortaya konulan yöntemin, sınırlı bilgi işlem kaynaklarında ve yüksek miktarda veri olduğunda NLP için daha uygun olduğu sonucuna varılmıştır. Tam ve diğerleri (2021) tarafından 2021 yılında yapılan Twitter Duygu Analizi konulu çalışmada İngilizce dilinde olan tweetler üzerinde CNN ve çift yönlü LSTM (Bi-LSTM) modellerinin birarada kullanılması gerçekleştirilen sınıflandırmada %91.13 doğruluk oranı elde edilmiştir.

Çalışma	Doğruluk
Kaya vd. (2012)	%77
Akba (2014)	%83.9 F1 skor
Coban vd. (2015)	%66,06
Karamollaoğlu vd. (2018)	%80
Bozyiğit vd. (2019)	%91 F1 skor
Rumelli vd. (2019)	%73.8
Pervan (2019)	%94,21
Çöltekin (2020)	%77.3 F1 skor
Kirelli ve Arslankaya (2020)	%74.63

Şekil 5.1 Çeşitli Türkçe Duygu Analizi Çalışmaları ve Doğruluk Oranları

6. METODOLOJİ

Bilgisayarların hızlanmasıyla birlikte çok sayıda Makine Öğrenmesi algoritmasının büyük verilerle işlenebildiği görülmüştür. Doğal Dil İşleme alanında Makine Öğrenmesi algoritmalarının kullanılmaya başlanmasıyla, bu alanda yaşanan gelişimlerden Doğal Dil İşleme alanı da aynı şekilde etkilenmiştir. Literatürde Doğal Dil İşleme'ye dair pek çok örnek bulunmaktadır. Bu alanda yapılan çalışmalar günden güne iyileşmektedir. NLP'nin metin üzerine yapılan çalışmaları günümüzde reklam, anket, dilden dile çeviri gibi sayısız alanda bilgisayar sektöründe uygulanmaktadır. Bu alanda yapılan akademik çalışmalar NLP'nin yapılış amacına göre değişerek, farklı makine öğrenmesi algoritmaları ile gerçekleştirilmektedir.

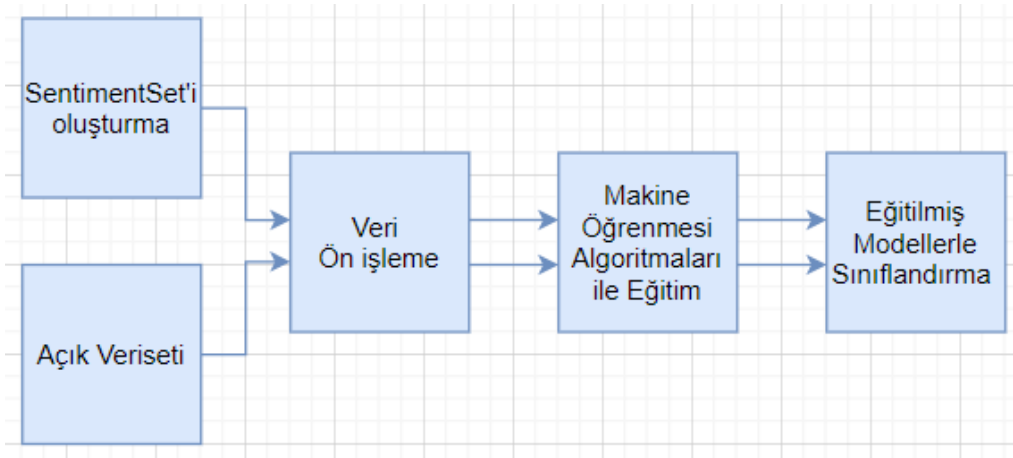
Duygu Analizi, bir konudaki doğal dil kullanılarak belirtilen sözlü veya yazılı bir ifadenin pozitif veya negatif biçimde sınıflandırılmasıdır. Bu sınıflandırma çok pozitif, az pozitif, nötr, az negatif çok negatif gibi derece sayısı artırılarak gerçekleştirilebilir. Duygu Analizi kavramı, Duygu Sınıflandırma, Fikir Madenciliği gibi farklı isimlerle de anılmaktadır. (Pervan 2019) Sosyal medya, kullanımının artması ve kişilerin bu alandaki etkinliğinin ve paylaşımlarının çeşitlenmesiyle beraber büyük bir veri kaynağı haline gelmiştir. Twitter bu alanda yapılan çalışmalarda kullanılan ve büyük bir veri kaynağı olarak görülen sitelerden biridir. Twitter'ın sağladığı verilerle literatürde pek çok çalışma bulunmaktadır. Türkçe içeriklerle yapılan çalışma sayısı ise günden güne artmaktadır.

Bu tez kapsamında, Duygu Analizi çalışması yapmak amacıyla doğal dil olarak metin verisi ile çalışmak için iki farklı veri seti kullanılmıştır. Bu veri setleri Twitter'dan alınan önceden etiketlenmiş pozitif ve negatif tweetlerden oluşmaktadır. Veri setleri çeşitli Makine Öğrenmesi algoritmalarıyla eğitilmiştir. Oluşturulan modellerle sosyal medya kullanıcılarının duygu durumları pozitif veya negatif olarak sınıflandırılmaya çalışılmıştır.

Elde edilen sonuçlarla kullanılan algoritmaların başarı oranları karşılaştırılmış ve doğruluk oranları ortaya konulmuştur. Modellerin eğitildiği veri setlerinin dışında rastgele tweetlerden oluşan küçük bir veri seti ile verilen modeller ayrıca karşılaştırılmış ve sonuçlar raporlanmıştır.

6.1 Model Mimarisi

Çalışma kapsamında, modellerin eğitebilmesi için kullanılacak tweetlerden oluşan veri setlerinin hazırlanma işlemleri gerçekleştirilmiştir. Her veri seti oluşturulurken çeşitli ön işlemlerden geçirilmiştir. Veri setleri oluşturulurken geçirdikleri ön işlem haricinde, veriler kullanılmadan önce tekrar ön işlemlerden geçirilmiştir. Ön işlemlerden geçirilen veriler makine öğrenmesi algoritmalarına giriş değeri olarak verilerek eğitilmiştir. Farklı algoritmalarla eğitilen modellerle tahminleme ve sınıflandırma işlemi yapılmıştır. Şekil 6.1’de basitçe model mimarisi ifade edilmiştir.



Şekil 6.1 Model mimarisi genel şeması

Tez kapsamında iki farklı veri seti kullanılmıştır. Oluşturulan modellerle her iki veri seti için de eğitilip farklı sonuçlar elde edilip algoritmaların modeller üzerindeki başarısı karşılaştırılmıştır. Duygu analizi çalışmaları yapılırken tweetlerden çıkarılacak duygu

durumunu sınıflandırma işlemi birey bazında da zorlu bir görevdir. Bu alanda tez içerisinde literatürde belirtildiği gibi fazlasıyla çalışma bulunmaktadır.

Türkçe çalışma sayısı ise günden güne artış göstermektedir. Kullanılan dilin morfolojik yapısı ve kültürel yaklaşım gibi etkenler bu analizin başarısı konusunda büyük rol oynamaktadır. Türkçe dili, deyim, ata sözleri, ima ve kelimelerin yan yana geldiğinde farklı anlamlara yol açması gibi sebeplerle bir cümleden öznel anlamlar çıkarmak oldukça mümkündür. Veri seti olarak değerlendirilecek olursa bu durum veri setleri içerisinde farklı kişilerin tweetleri farklı duygulara yorumlaması gibi sonuçlara sebep olabilir. Tez kapsamında oluşturulan modellerle, veri kalitesi ile birlikte farklılaşabilecek bu zorlu göreve karşılık bir başarı ortaya konulmaya çalışılmıştır.

6.2 Veri Oluşturma

Makine Öğrenmesi alanında modellerin başarısını en çok etkileyen faktörlerden biri de kullanılan veridir. Doğru veri ile doğru şekilde eğitilen modellerin başarılı olma ihtimali yükselir. Öyle ki, yeterince veri olduğunda çok iyi hazırlanmamış bir model bile başarılı sonuçlanabilir. Verinin kalitesinin önemi bu derece yüksekken, veriyi doğru şekilde kullanmak da aynı derecede önemlidir. Bu çalışma kapsamında, sosyal medya verileri üzerinde Duygu Analizi yapan modeller geliştirildi. Daha önce bahsedildiği gibi sosyal medya verileri için kaynak olarak Twitter API kullanıldı. Tweetler çeşitli ön işlemlerden geçirilerek kalitesi artırıldı ve algoritmalara giriş değeri olarak verildi.

Bu tez kapsamında, iki çeşit veri seti ile eğitim yapılmıştır. İlki Beyaz (2020) tarafından oluşturulmuş, çeşitli ön işlemlerden geçirilmiş ve kaggle'a eklenmiş olan açık kaynaklı bir veri setidir. İçeriğinde yaklaşık 11.000 veri içeren bu veri seti pozitif ve negatif olarak işaretlenmiş Türkçe tweetlerden oluşmaktadır. Veri seti, Beyaz (2020) tarafından sosyal medyada zorbalık tespiti ile ilgili bir proje için geliştirilmiştir. Kullanılan veri setinde bazı ön işlemlerin olduğu görülmektedir ancak modeller oluşturulurken veriler ekta çeşitli veri ön

işlemlerden geçirilmiştir. Şekil 6.2’de açık kaynak kodlu veri setine ait bir örnek gösterilmektedir.

sentiment	tweets
Pozitif	uçur beni rüzgar
Negatif	sana küfür ederim

Şekil 6.2 açık kaynaklı veri setinden alınmış örnek veri (Beyaz 2020)

Bu tez kapsamında modelleri eğitmek için kullanılan açık kaynaklı veri setinin yanı sıra kullanılan ikinci veri seti ise, çalışma kapsamında geliştirilmiş SentimentSet (Balli ve Balli 2021) isimli veri setidir. Veri seti ilk toplandığında yaklaşık 2800 Türkçe tweetten oluşmaktadır. Her bir tweet pozitif negatif ve nötr olarak işaretlenmiştir. Sonradan nötr tweetler çıkarılmıştır. Veri setini oluşturan tweetler toplanırken iki yöntemden yararlanılmıştır. Bu yöntemlerden birincisi Twitter API’ye ait stream methodunun, ikincisi ise için açık kaynak kodlu olan snsrape (Anonymus 2021d) kütüphanesinin kullanılmasıdır. Tweetler herhangi bir düzen takip edilmeden rastgele olarak seçilmiştir.

Tweetleri toplamak için kullanılan birinci yöntem için, öncelikle Twitter’dan tweetleri API aracılığıyla çekebilmek için gereken izinlere başvurulmuştur. Twitter tarafından alınan Twitter geliştirici hesabı yetkisi ile beraber elde edilen anahtar ve şifreler Tweepy methodunun kullanılabilmesi için kaynak koda eklenmiştir. Tweepy kütüphanesine ait olan Stream methodu verilen parametrelere uygun veri elde etmeyi sağlamaktadır. Veri seti 'aşı', 'aşılacak', 'aşı olmak', 'vaccine', 'vaccinated', 'vaccination', 'stayhome', 'stay home', 'covid', 'corona', 'coronavirus', 'korona', 'covid-19', 'Covid19', 'Covid-19', 'Corona Virus', 'pandemic', 'pandemi', 'COVID-19' kelimesi aratılarak seçilen gerçek zamanlı tweetlerden oluşturulmuştur. Parametrelerden bir diğeri ise sadece Türkçe tweetlerin çekilmesi için kullanılan “languages = ‘tr’” dir. Alınan tweetler el ile bir tabloya aktarılıp burada manuel işaretleme yapılmıştır. Tweetler toplanırken kullanılan ikinci yöntemde, tweetler Şekil 6.3 ‘te gösterildiği gibi 2020 mart ayından itibaren rastgele aylardan çekilerek ‘pandemi’ kelimesi aratılarak rastgele zaman aralıklarında resimdeki filtrelerle oluşturulmuştur.

```
TwitterSearchScraper('pandemi' + ' ' + 'lang:tr since:2020-03-11 until:2021-03-01'+  
    '-filter:RT -filter:links -filter:replies').get_items()):
```

Şekil 6.3 snscape kütüphanesi tweet toplama örneği

Belirtilen yöntemlerle oluşturulan tweetler bir araya getirilmiştir. Biraraya gelen tweetler çeşitli ön işlemlerden geçirildikten sonra manuel olarak işaretlenip SentimentSet veri seti oluşturulmuştur. Tweeler işaretlenirken, pozitif, negatif ve nötr olarak işaretlenmiştir. Duygu ifadesi olan ancak tam olarak nitelendirilemeyen ve anlaşılamayan veya pozitif veya negatif bir duygu ifade etmeyen tweetler nötr olarak işaretlenmiştir. Şekil 6.4'te SentimentSet veri setine ait bir örnek gösterilmektedir. Çalışma kapsamında, Duygu Analizi pozitif veya negatif sınıflandırmaya dayalı olarak yapıldığından, nötr olarak işaretlenen tweetler işleme alınmamıştır.

sentiment	tweets
pozitif	pandemi bitse karakoçan spor fk'mızın maçına gitsek
nötr	kanıtlanmayan sebep sorusu aşı konusuna girmeden ilk sorulacak soru
negatif	abi korona maskeyi cikariyorsun tukuruyorsun sonra baska birisi gelip aliyor gectim cocuksu hareket nefretinde bogul

Şekil 6.4 SentimentSet veri setine ait tweet toplama örneği

6.3 Veri Ön İşleme

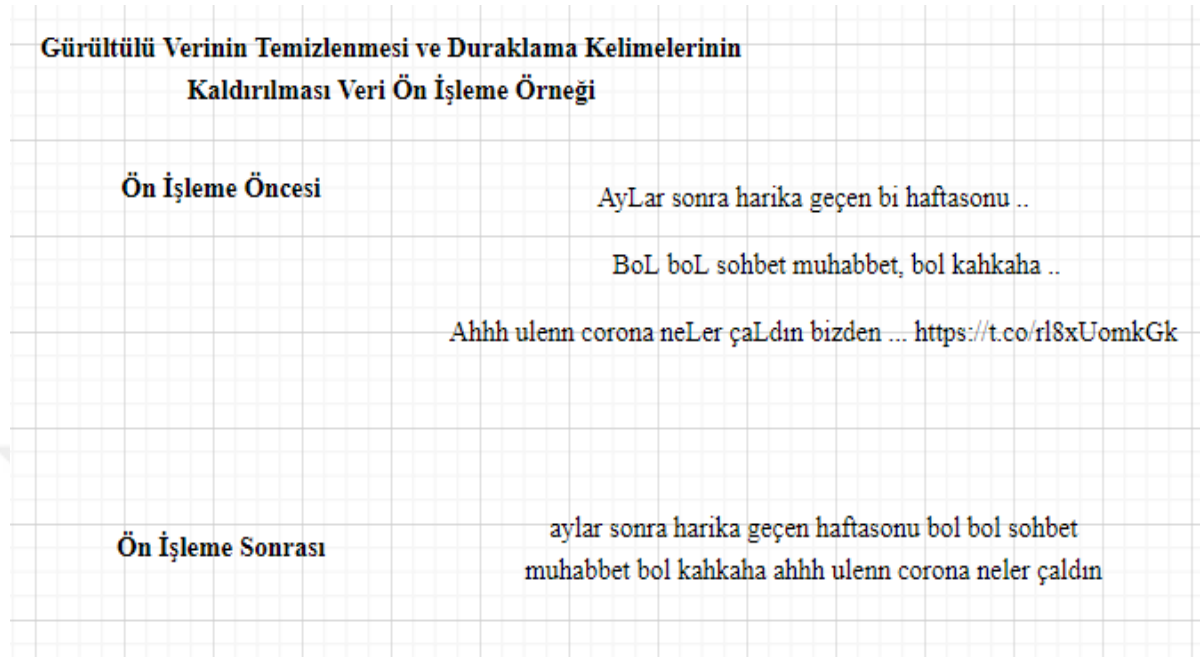
Tez kapsamında iki çeşit ön işleme yapılmıştır. Bunlardan biri Twitter API kullanılarak oluşturulan SentimentSet'in, tweet toplama aşamasından sonra çeşitli ön işlemlerden geçirilmesidir. Sonrasında işaretleme ve veri seti haline getirme işlemi yapılmıştır. Veri seti oluşturma aşamasında yapılan ön işlemenin Beyaz (2020) hazırladıkları açık kaynak kodlu diğer veri seti incelendiğinde yapıldığı görülmüştür. Ancak bu veri seti bu tez kapsamında oluşturulmadığından buradaki ön işlemler de bu tez kapsamında yapılmamıştır. Tez kapsamında yapılan diğer ön işleme ise verilerle Makine Öğrenmesi algoritmalarında kullanılmadan önce yapılan ön işlemlerdir. Bu ön işleme eğitimden önce kullanılacak bütün verilere uygulanabilir. Çalışma kapsamında veri setleri kullanılmadan önce bu ön

işlemlerden geçirilip algoritmalarla giriş değeri olarak verilmiştir. Bu aşamada Zeyrek (2020a) 'nın yaptığı ön işleme adımlarından yararlanılmıştır.

Genel anlamda veri ön işleme veri setlerini makine öğrenmesi algoritmalarına hazırlar ve veri kalitesinin artırarak daha yüksek başarı elde etmeyi hedefler. Tez kapsamında hazırlanan SentimentSet veri setinde ise tweet toplama işleminden sonra veri seti oluşturma aşamasında çeşitli ön işleme işlemleri yapılmıştır. Toplanan tweetler, öncelikle tez içeriğinde daha önce bahsedilmiş olan Gürültülü Veri Temizleme işleminden geçirilmiştir. Bu işlemin amacı veriyi gerekli ve Duygu Analizi başlığı altında anlamlı olmayan verilerden temizlemektir. Tweetler içeriğindeki emojiler, http ile başlayan linkler kaldırılmıştır. Noktalama işaretleri gibi çeşitli semboller kaldırılmıştır. Twitter' a ait olan tweetlerde geçen ve (@ işareti ile başlayan) kullanıcı isimleri kaldırılmıştır. Bu işlemten sonra tweeti düzenlemek için tweet içeriği tamamen küçük harfe dönüştürülmüştür. Sonrasında tweetlerin başında ve sonunda bulunan boşluklar (whitespaces) silinmiştir.

Bir sonraki veri ön işleme işlemi ise tez kapsamında daha önce bahsedilen duraklama kelimelerinin silinmesidir. Çalışma kapsamında duraklama kelimelerinin pozitif veya negatif anlam bozukluğu yaratması veya gerekli olmaması gibi sebeplerle silindiğinden bahsedilmişti. Dile özgü değişkenlik gösteren bu kelimeler, literatürde yapılan çalışmalarda değişkenlik gösterebilir. Bu çalışma kapsamında ise kullanılan duraklama kelimeleri imjunsung (2015) tarafından listesi LUCENE-559 Türkçe duraklama kelimeleri ile "Information Retrieval on Turkish Texts" listesi ile birleştirilmesi ile oluşturulan Türkçe diline ait stopword listesinden alınmıştır.

Duraklama kelimeleri silindikten sonra veri oluşturma aşamasında yapılan veri ön işleme işlemi tamamlanmıştır. Sonrasında veriler işaretlenip SentimentSet veri seti oluşturulur. Şekil 6.5 'te gürültülü verinin temizlenmesi ve duraklama kelimelerinin kaldırılması ile yapılan bir ön işleme örneği gösterilmektedir.



Şekil 6.5 Gürültülü verinin temizlenmesi ve duraklama kelimelerinin kaldırılması veri ön işlem örneği

Tez kapsamında iki çeşit ön işleme yapıldığından bahsedilmiştir. İlki, veri oluşturma aşamasında yapılan ve sadece tez kapsamında kullanılan SentimentSet veri seti oluşturulurken gerçekleştirilen ön işlemdir. Veri oluşturma aşamasında yapılan ön işleme dışında bir diğeri de Makine Öğrenmesi algoritmalarından önce uygulanan veri ön işleme işlemleridir. Makine Öğrenmesi algoritmalarına giriş değeri olarak verilmeden önce kullanılan veri setleri üzerinde gerçekleştirilen ön işlem çalışma kapsamında kullanılan her veri setine uygulanmıştır. Bu ön işlemden öncelikle gürültülü veri temizleme ve duraklama kelimelerinin kaldırılması işlemleri gerçekleştirilmiştir. Bu iki işlem zaten bu şekilde hazırlanmış olan SentimentSet'i etkilememektedir. Ancak Beyaz (2020) tarafından hazırlanan veri seti de böylece bu işlemlerden geçirilmiştir.

Sonrasında ise veri kök bulma (stemming/lemmatization) işlemi yapılmıştır. Bu işlem, iki farklı kütüphane kullanılarak gerçekleştirilmiştir. Bunlardan birisi Zemberek (Anonymus 2019) kütüphanesidir. Diğeri ise NLTK'ya ait olan SnowBall kütüphanesidir. Zemberek (Anonymus 2019) kütüphanesi javada yazılmış bir kütüphanedir.

Tez içeriğindeki modeller python ile yazılmıştır. Zemberek kütüphanesini kullanabilmek için bir JVM çalıştırılması gerekmektedir. Sonrasında kütüphaneye ait jar dosyası çalıştırılarak methodlar kullanılabilir. Tez kapsamında lemmatization işleminden bahsedilmiştir. Kütüphaneye ait TurkishMorphology sınıfı yardımıyla kelimelerin lemmaları yani kökleri bulunmuştur. Şekil 6.6'da Zemberek yardımıyla bulunan kelime kökleri örneği gösterilmektedir.

```
Tweet : teketekbilim fatih bey sebepden olmamız gereken doz aşı temin edilemez  
farklı ülkenin aşısı olunabilir  
Kelime Kökleri : UNK fatih bey UNK ol gerek doz aşı temin et fark ülke aşı ol
```

Şekil 6.6 Zemberek kütüphanesi kullanılarak bulunan kelime kökleri örneği

Kullanılan bir diğer kök bulma kütüphanesi olan NLTK' ya ait Snowball kütüphanesine ait TurkishStemmer isimli sınıf kullanılarak kelimelerin köklerini stemming yaparak bulunmuştur. Tez kapsamında stemming işleminden daha önce bahsedilmiştir. Şekil 6.7'de Snowball kütüphanesi ile stemming ile yapılan kök bulma işlemi gösterilmektedir.

```
Tweet: toplu gidilen özledim pandemi bitsin kalabalığa giricem  
Kelime Kökleri: toplu gidile özledi pande bit kalabalık girice
```

Şekil 6.7 Snowball kütüphanesi kullanılarak bulunan kelime kökleri örneği

Tez kapsamında kök bulma işlemleri arasındaki farklardan daha önce bahsedilmiştir. Bu kütüphaneler ile lemmatization ve stemming olmak üzere iki farklı yöntem kullanarak kök bulma işlemi gerçekleştirilmektedir. Modeller oluşturulurken her iki yöntem de denenmiş ve sonuçları raporlanmıştır.

Kökleri bulunan verilere bu işlemten sonra bir diğer veri ön işlemi olan Metin Vektörleştirme (Text Vectorization) işlemi uygulanmıştır. Bu işleme geçilmeden önce verinin eğitim ve test verisi olarak ayrılması gerekmektedir. Çalışma kapsamında oluşturulan bütün modellerde verinin %20 si test için ayrılmıştır. Kalan kısım ise eğitim için kullanılmıştır. Eğitim ve test verisi ayrıldıktan sonra Metin Vektörleştirme işlemi yapılmıştır. Bu işlemin amacı basitçe, çeşitli ön işlemlerden geçirilmiş insan dilinde olan veriyi, makinenin anlayacağı dile çevirmektir. Bu işlemin sonucunda elde edilen veriler makine öğrenmesi algoritmalarına girdi olarak verilmiştir.

Bu çalışma kapsamında Metin Vektörleştirme işlemi için TF-IDF tekniği kullanılmıştır. Şekil 6.8’ de Tf-idf tekniğinin nasıl kullanıldığı gösterilmiştir. Tf-idf bir kelime torbası (bag of words) tekniği olduğundan “ngram_range = (1,2)” parametresi, tez kapsamında daha önce bahsedilmiş olan kelime torbası yöntemine ait unigram ve biagram yaklaşımlarının birlikte kullanıldığını ifade eder. Şekilde gösterilen ‘max_df = 0.9’ parametresi belgelerin %90’ından daha fazlasında görünen terimleri yok say demektir. Aynı şekilde ‘min_df = 5’ parametresi de 5’ten daha az sayıda belgede görünen terimleri yok say demektir. Şekilde gösterildiği gibi vectorizer nesnesi belirtilen parametrelerle oluşturulduktan sonra eğitim için ayrılan ‘x_train’ verisine göre ölçeklendirmek ve bu ölçeklendirme parametrelerini öğrenme için fit_transform methodu çağrılır. Eğitim setinin özelliklerinin ortalaması ve varyansı öğrenilir. Sonrasında ‘x_test’ öğrenilmiş bu parametrelere göre transform() methodu çağrılarak ölçeklendirilir. (Khanna 2020)

```
tfidf_vectorizer = TfidfVectorizer(ngram_range=(1,2), max_df=0.9, min_df=5)
x_train_tfidf = tfidf_vectorizer.fit_transform(x_train)
x_test_tfidf = tfidf_vectorizer.transform(x_test)
```

Şekil 6.8 Metin Vektörleştirme için kullanılan Tf-idf yöntemi gösterimi

Metin Vektörleştirme işleminden sonra veriler Makine Öğrenmesi algoritmalarında kullanılmıştır. Ancak LSTM için bu veri ön işlemleri diğerlerinden farklıdır.

LSTM’de kök bulma işlemi kullanılmamıştır. Vektörleştirme ve Simgeleme (Tokenization) işlemleri için Keras kütüphanesi’ne ait Tokenizer sınıfı kullanılmıştır. (Hallaç 2018)

```
max_features = 2000
tokenizer = Tokenizer(num_words=max_features, split=' ')
tokenizer.fit_on_texts(data['tweets'].values)
X = tokenizer.texts_to_sequences(data['tweets'].values)
X = pad_sequences(X)
```

Şekil 6.9 LSTM için kullanılan Tokenizer gösterimi

LSTM için kullanılan Tokenizer sınıfı ve parametreleri Şekil 6.9’da gösterilmiştir. Şekilde gösterilen parametrelerden ‘num_words = 2000’ değeri kaç kelimenin işleme alınacağını belirlemektedir. Kelime ayırma işlemleri verilen “split = ‘ ’” parametresi ile boşluğa göre yapılmıştır. Şekilde görüldüğü gibi fit_on_text methoduna veri seti parametre olarak verilmiştir. Böylece tokenizer veriye dair frekans bilgilerine sahip olmaktadır. Bu method frekansa göre kelime indeksi oluşturmaktadır. Her kelimenin kendine ait bir integer değeri bulunmaktadır. Şekildeki text_to_sequences, her kelimeyi word_index sözlüğündeki karşılık gelen tamsayı ile değiştirir. Şekildeki pad_sequences ise, bir listedeki tüm dizilerin aynı uzunluğa sahip olmasını sağlamak için kullanılır. Varsayılan olarak bu, her bir dizi en uzun dizi ile aynı uzunluğa sahip olana kadar her dizinin başlangıcına 0 eklenerek yapılır.

6.4 Model Oluşturma

Veri kalitesi artırılıp veriler ön işlemlerden geçirilip makine öğrenmesi algoritmalarının kullanabileceği bir forma getirilmiştir. Tez kapsamında oluşturulan modeller python dili ile yazılmıştır. Modeller oluşturulurken Keras kütüphanesinden ve Zeyrek (2020b) örnek kodlarından yararlanılmıştır. Çalışma ortamı olarak Google Colab seçilmiştir. Google Colab üzerinden gerçekleştirilen çalışmada hazırlanan ve ön işlemlerden geçirilen veriler pek çok makine öğrenmesi algoritmasında kullanılmıştır.

6.4.1 Logistik Regresyon Modeli

Çalışma kapsamında kullanılan Makine Öğrenmesi algoritmalarından biridir. Model hiper parameterleri varsayılan şekilde kullanılmıştır. Eğitim veri seti ve eğitim vektörü modele fit methodu ile eklenmiştir. Bu model ile predict methodu kullanılarak test verisi üzerinde tahminleme veya sınıflandırma yapılmaya çalışılmıştır.

6.4.2 SGD Modeli

Çalışma kapsamında kullanılan bir diğer Makine Öğrenmesi algoritmalarından biri de Stokastik Gradyan İniş algoritmasıdır. SGDClassifier modeli hiper parameterleri 'max_iter = 5' parametresi dışında varsayılan şekilde kullanılmıştır. Varsayılan değeri 1000 olan 'max_iter' parametresi, eğitim sırasında verilerin üzerinden maksimum kaç kere geçileceğini ifade eder. Eğitim veri seti ve eğitim vektörü modele fit methodu ile eklendikten sonra test verisi üzerinde üzerinde predict methodu kullanılarak tahminleme ve sınıflandırma yapılmaya çalışılmıştır.

6.4.3 Rastgele Orman Modeli

Bu çalışma kapsamında model eğitmek için kullanılan Makine Öğrenmesi algoritmalarından biridir. RandomForestClassifier modeli hiper parametlerinden olan 'n_estimators = 20' olarak kullanılmıştır. Varsayılan değeri 100 olan bu parametre, ormanda kaç adet ağaç olduğunu ifade etmektedir. Kullanılan diğer parametre 'random_state = 0' ise her çalışmada aynı sonucun alınmasını sağlamak için eklenmiştir. Diğer hiper parametreler varsayılan olarak kullanılmıştır. Diğer algoritmalarda da olduğu gibi eğitim seti ve eğitim vektörü modele fit methodu ile eklenmiştir. Bu model ile predict methodu kullanılarak tahminleme ve sınıflandırma yapılmaya çalışılmıştır.

6.4.4 Bayesian Modeli

Tez kapsamında ikili sınıflandırma yapıldığı için BernoulliNB modeli varsayılan hiper parametrelerle birlikte kullanılmıştır. Eğitim veri seti ve eğitim vektörü modele fit methodu ile birlikte eklenmiştir.

Bu model ile predict methodu kullanılarak test verisi üzerinde tahminleme veya sınıflandırma yapılmaya çalışılmıştır.

6.4.5 SVM Modeli

Tez kapsamında Destek Vektör Makinesi algoritmasına ait bir sınıflandırma modeli olan SVC modeli, hiper parametrelerden 'kernel = linear' dışındaki parametreleri varsayılan haliyle kullanılmıştır. Veriler lineer olarak ayrılabilirdiği ve veriden çıkarılan özellik (feature) sayısı fazla olduğu için metin sınıflandırma problemlerinde genellikle bu parametre değeri kullanılmaktadır. (Anonymus 2018) Eğitim veri seti ve eğitim vektörü modele fit methodu ile birlikte eklenmiştir. Bu model ile predict methodu kullanılarak test verisi üzerinde tahminleme veya sınıflandırma yapılmaya çalışılmıştır.

6.4.6 LSTM Modeli

Tez kapsamında LSTM modeli ile sınıflandırma yapılmaya çalışılmıştır. Bu model oluşturulurken veri ön işleme işlemlerinden kök bulma ve metin vektörleştirme işlemleri uygulanmamıştır. Gürültülü veri temizleme, duraklama kelimelerinin kaldırılması işlemleri sonrasında daha önce bahsedilmiş olan simgeleme (tokenization) veri ön işleme yöntemi kullanılmıştır. Kullanılan modelin parametreleri Şekil 6.10'da gösterilmiştir. Şekilde gösterildiği gibi Sequential() modeli kullanılmıştır. Modele birer adet embedding katmanı, dropout katmanı LSTM katmanı ve Dense katmanı eklenmiştir. Şekilde gösterilen max_features, embed_dim, lstm_out hiperparametreleri kullanılmıştır. Model geliştirilirken Nagy (2017) 'ye ait kod örneklerinden yararlanılmıştır.

```
max_features = 2000
embed_dim = 128
lstm_out = 196
model = Sequential()
model.add(Embedding(max_features, embed_dim,input_length = X.shape[1]))
model.add(Dropout(0.8))
model.add(LSTM(lstm_out))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
```

Şekil 6.10 LSTM modeli ve hiper parametreleri

Sequential modeli sıralı ve katmanlı bir yapı sağlar. Her katmanın bir giriş bir de çıkış değeri bulunur. Katmanlar add methodu ile modele eklenir (fchollet 2020). Keras metin verisi için sıkça kullanılan Embedding katmanı eklenmiştir.

Bu method daki parametrelerden, 'max_features' kelime dağarcığının boyutunu, 'input_length' girdi dizilerinin uzunluğunu ifade eder. 'embed_dim' ise kelimelerin gömüleceği vektör uzayının boyutunu tanımlar. Ayrıca 'embed_dim' parametresi her kelime için ilgili katmandaki çıktı vektörlerinin boyutunu ifade eder (Brownlee 2017).

Dropout katmanı overfitting problemini önlemek için eklenmiştir. Sonrasında modele LSTM katmanı eklenmiştir. Bu methodda kullanılan lstm_out parametresi çıkış uzayının boyutunu ifade eder. Dense katmanı, tez kapsamında 2 li sınıflandırma yapıldığından 2 parametresini alır. Bu katmanda Aktivasyon fonksiyonu için 'softmax' kullanılmıştır. Model compile methoduna parametre olarak 'categorical_crossentropy' loss fonksiyonu, Adam algoritması ve ayrıca accuracy metriği eklenmiştir. Eğitim verisi bu modele şekil 6.11 'de gösterildiği gibi fit methodu ile eklenmiştir. Fit_methoduna eğitim verileri, bütün veri setinin üzerinden kaç kere geçileceğini gösteren 'epoch' sayısı, her yinelemede kullanılan eğitim örneği sayısını ifade eden 'batch_size' ve ayrıntılı bir output görmek için kullanılan verbose parametresi eklenmiştir.


```
batch_size = 16
model.fit(X_train, Y_train, epochs = 6, batch_size=batch_size, verbose = 2)
score,acc = model.evaluate(X_test, Y_test, verbose = 2, batch_size = batch_size)
```

Şekil 6.11 LSTM modeli fit ve evalute methodları ve parametreleri

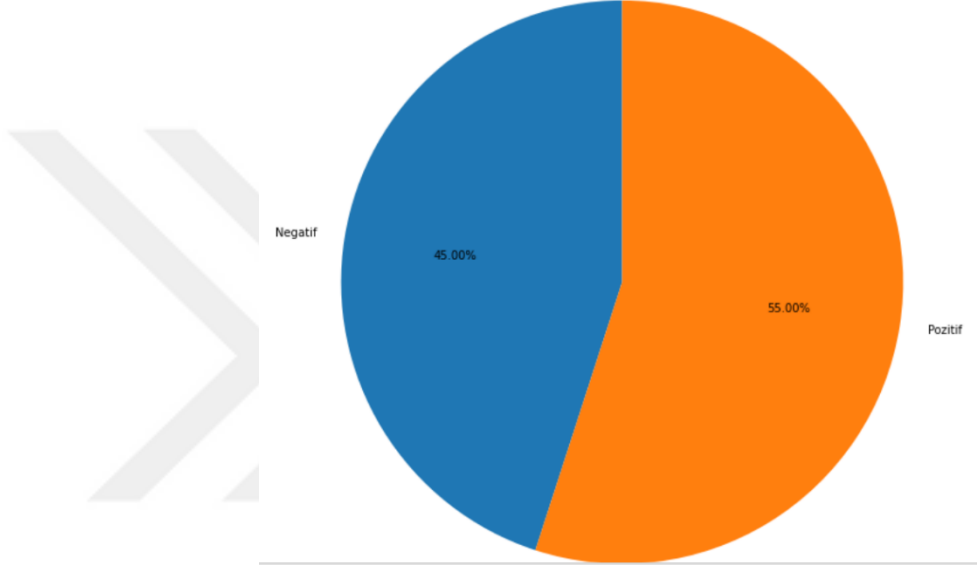
Sonrasında model ile evaluate methodu kullanılarak test verisi üzerinde sınıflandırma yapılmaya çalışılmıştır. Model ile tahmin edilen sınıfları veren method ise Şekil 6.12’ de gösterilmiştir. Modelde kullanılan evalute methodu loss fonksiyonunu verirken, predict methodu tahmin edilen değerleri verir.

```
data_list = pd.DataFrame (model.predict_classes(X), columns = ['sentiment'])
```

Şekil 6.12 LSTM modeli predict_classes methodu gösterimi

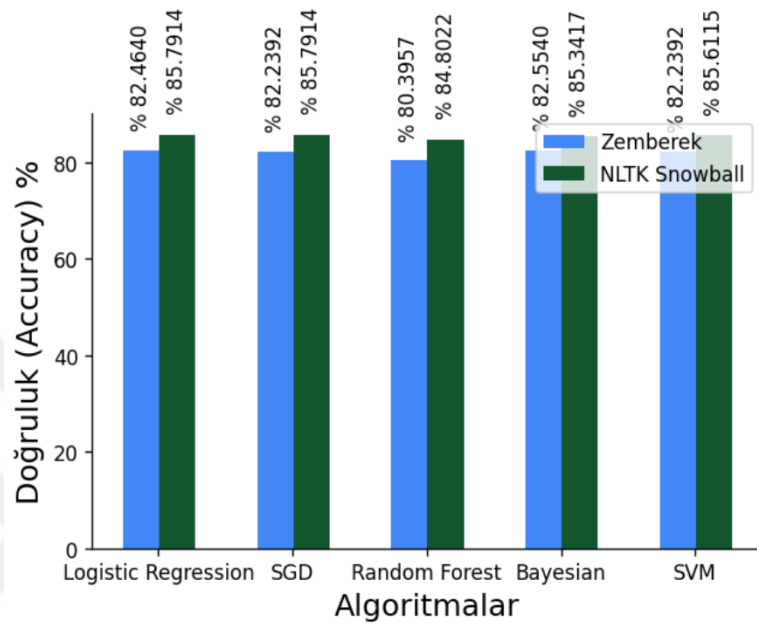
7. DENEYLER

Tez kapsamında eğitim için kullanılan işaretli açık veri setinde 11119 adet tweet bulunmaktadır. Bu veri setinin negatif, pozitif kategori oranları Şekil 7.1’ de gösterilmiştir.



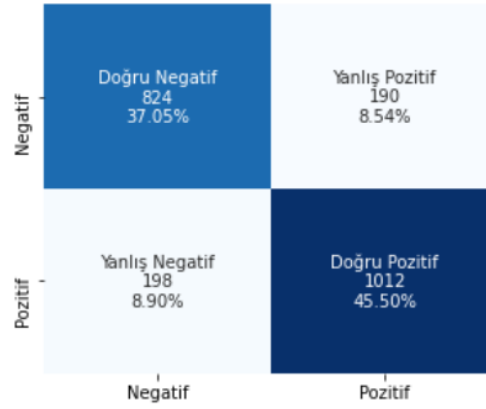
Şekil 7.1 Açık veri setinin kategori oranlarının görselleştirilmesi

Bu veri seti üzerinde daha önce bahsedilen makine öğrenmesi algoritmaları ile oluşturulan modellerin bu veri setinin %20 lik kadarının ayrılması ile oluşturulmuş test verisi üzerindeki sonuçları Şekil 7.2’de gösterilmiştir. Bu görsel modellerin doğruluk oranlarını ve kök bulma algoritması olarak Zemberek veya Snowball kütüphaneleri kullanıldığında elde edilen sonuçların karşılaştırmasını göstermektedir. Bu veri seti ile kök bulma algoritması olarak NLTK’ya ait olan Snowball kütüphanesi kullanıldığında modellerin başarı oranının arttığı gözlemlenmiştir.

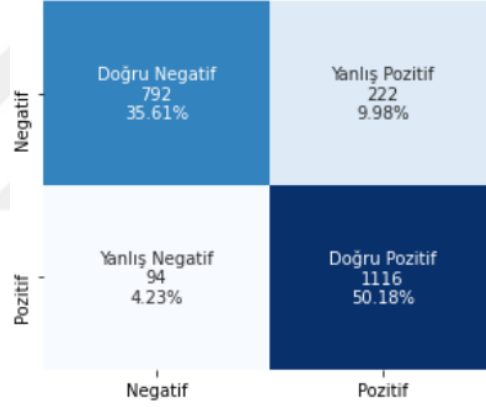


Şekil 7.2 Açık veri seti ile makine öğrenmesi algoritmalarının doğruluk oranlarının kök bulma kütüphanelerine göre karşılaştırılması

Açık veri seti ile Zemberek kütüphanesi kullanılarak eğilmiş modellerden Bayesian modeli veri setinden ayrılmış test verisi ile test edildiğinde elde edilen doğru yanlış tahminler Şekil 7.3'te yer almaktadır. Açık veri seti ile NLTK Snowball kütüphanesi kullanılarak eğilmiş modellerden Logistik Regresyon modeli, veri setinden ayrılmış test verisi ile test edildiğinde elde edilen doğru yanlış tahminler Şekil 7.4'te yer almaktadır.

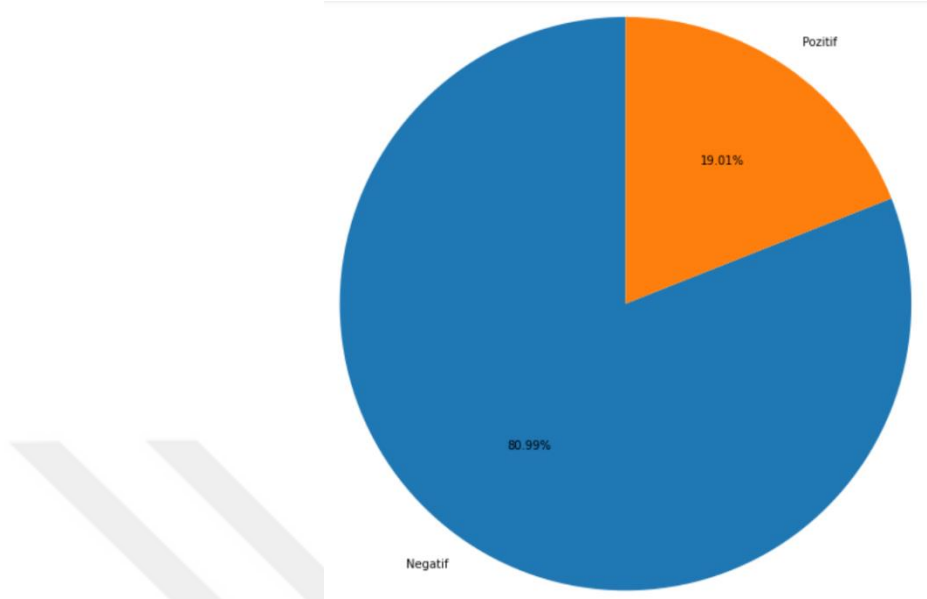


Şekil 7.3 Açık veri seti ve Zemberek kök bulma kütüphanesi ile eğitilmiş Bayesian modelinin veri setine ait test verisi ile test sonuçlarının dağılımı



Şekil 7.4 Açık veri seti ve NLTK Snowball kök bulma kütüphanesi ile eğitilmiş Logistik Regresyon modelinin veri setine ait test verisi ile test sonuçlarının dağılımı

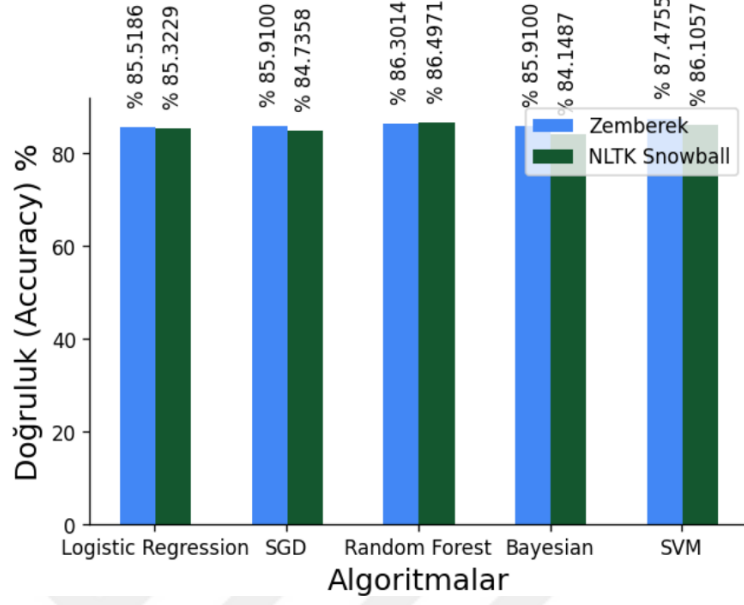
Tez kapsamında oluşturulan veri setinde yaklaşık 2767 veri bulunmaktadır. Bu veri setinden nötr kategorisi çıkarıldığında 2551 tane tweet içeren şekil 7.5’ te gösterilen pozitif negatif kategori oranlarına sahip veri seti oluşturulmuştur. Bu veri setinin %20 lik kısmı test verisi için ayrılmış, %80 lik kısmı ise eğitim için kullanılmıştır.



Şekil 7.5 SentimentSet veri setinin kategori oranlarının görselleştirilmesi

Bu veri seti üzerinde daha önce bahsedilen makine öğrenmesi algoritmaları ile oluşturulan modellerin bu veri setinin %20 lik kadarının ayrılması ile oluşturulmuş test verisi üzerindeki sonuçları Şekil 7.6’da gösterilmiştir. Bu görsel modellerin doğruluk oranlarını ve kök bulma algoritması olarak Zemberek veya Snowball kütüphaneleri kullanıldığında elde edilen sonuçların karşılaştırmasını göstermektedir. Bu veri seti ile kök bulma algoritması olarak Zemberek kütüphanesi kullanıldığında modellerin başarı oranının arttığı gözlemlenmiştir.

SentimentSet veri seti ile Zemberek kütüphanesi kullanılarak eğilmiş modellerden SVM modeli veri setinden ayrılmış test verisi ile test edildiğinde elde edilen doğru yanlış tahminler Şekil 7.7’te yer almaktadır.

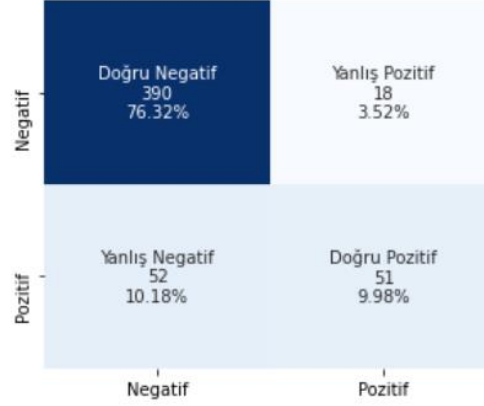


Şekil 7.6 SentimentSet veri seti ile makine öğrenmesi algoritmalarının doğruluk oranlarının kök bulma methodlarına göre karşılaştırılması



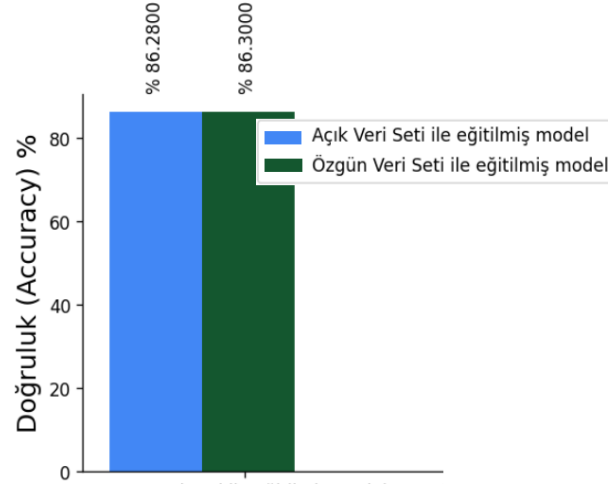
Şekil 7.7 SentimentSet veri seti ve Zemberek kök bulma kütüphanesi ile eğitilmiş SVM modelinin veri setine ait test verisi ile test sonuçlarının dağılımı

SentimentSet veri seti ile NLTK Snowball kütüphanesi kullanılarak eğitilmiş modellerden Rastgele Orman modeli veri setinden ayrılmış test verisi ile test edildiğinde elde edilen doğru yanlış tahminler Şekil 7.8’te yer almaktadır.



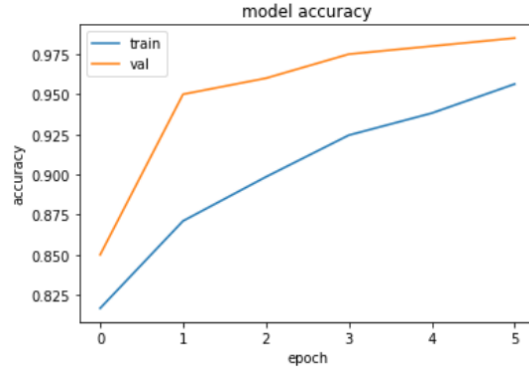
Şekil 7.8 SentimentSet veri seti ve NLTK Snowball kök bulma kütüphanesi ile eğitilmiş Rastgele Orman modelinin veri setine ait test verisi ile test sonuçlarının dağılımı

LSTM ile yapılan çalışmalarda kök bulma ön işlemi yapılmamıştır. Model eğitilirken açık veri seti için de SentimentSet veri seti için de oluşturulan modellerde veri setinin %20 lik kısmı test verisi için ayrılmış, %80 lik kısmı ise eğitim için kullanılmıştır. Açık veri seti veya SentimentSet veri seti kullanılarak eğitilen modeller, bu modellerle veri setine ait test verisi ile test edildiğinde elde edilen sonuçlar şekil 7.15’ te gösterilmiştir.

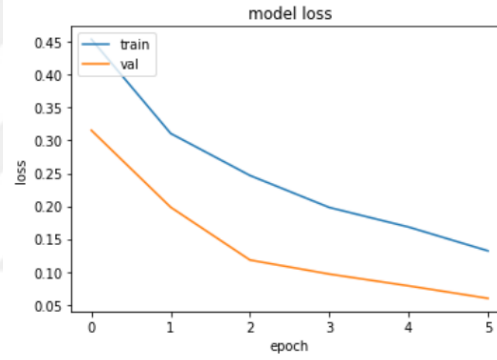


Şekil 7.9 LSTM farklı verisetleri ile eğitim ve test sonuçları

SentimentSet veri seti üzerinde LSTM modeli eğitilirken elde edilen eğitim ve doğrulama doğruluk değerleri Şekil 7.16’da, eğitim sırasında elde edilen eğitim ve doğrulama hata oranı değerleri ise Şekil 7.17’da gösterilmiştir.

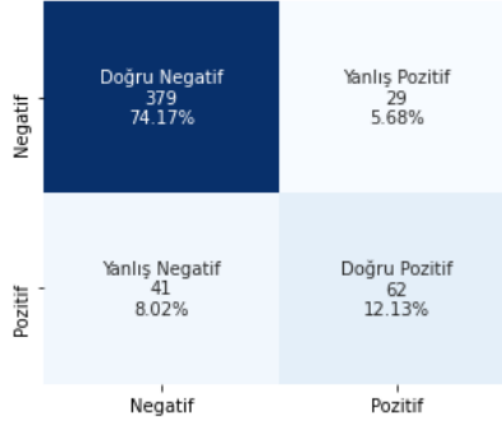


Şekil 7.10 SentimentSet veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama doğruluk değerleri



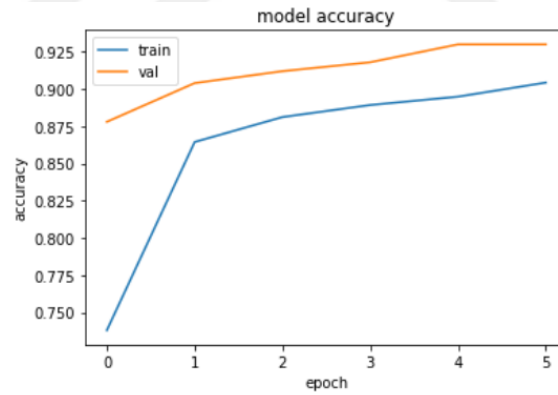
Şekil 7.11 SentimentSet veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama hata oranı değerleri

SentimentSet veri seti ile eğitilmiş LSTM modeli veri setine ait test verisi ile test edildiğinde elde edilen doğru yanlış tahminler Şekil 7.18’de yer almaktadır.



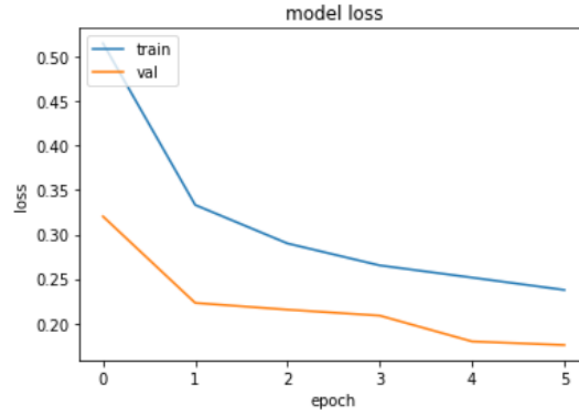
Şekil 7.12 SentimentSet veri seti ile eğitilmiş LSTM modelinin veri setine ait test verisi ile test sonuçlarının dağılımı

Açık veri seti üzerinde LSTM modeli eğitilirken elde edilen eğitim ve doğrulama doğruluk değerleri Şekil 7.20’de, eğitim sırasında elde edilen eğitim ve doğrulama hata oranı değerleri ise Şekil 7.21’de gösterilmiştir.

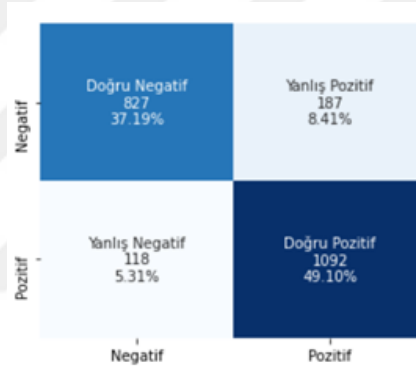


Şekil 7.13 Açık veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama doğruluk değerleri

Açık veri seti ile eğitilmiş LSTM modeli veri setine ait test verisi ile test edildiğinde elde edilen doğru yanlış tahminler Şekil 7.22’de yer almaktadır.



Şekil 7.14 Açık veri seti ile LSTM modeli eğitimi esnasında eğitim ve doğrulama hata oranı değerleri



Şekil 7.15 Açık veri seti ile eğitilmiş LSTM modelinin veri setine ait test verisi ile test sonuçlarının dağılımı

8. SONUÇ

Teknolojinin günden güne geçirdiği gelişim ve yapay zeka konularının sektörde ve akademik çalışmalarda yakaladığı ivmeyle literatürde bu alandaki çeşitlilik ve sayı günümüzde oldukça fazladır. Kullanım alanı giderek büyüyen doğal dil işleme, bu alanda ilgi toplayan konulardan biridir. Sosyal medya çok çeşitli verinin çok hızlı bir şekilde ulaşılabilmesi ve açık kaynak olması gibi sebeplerle doğal dil işleme çalışmaları için oldukça iyi bir veri kaynağıdır. Twitter API gerekli izinler alındıktan sonra akademik çalışmalar veya araştırmalar gibi konularda Twitter kullanıcılarının tweetlerinin anonim olarak kullanılmasını sağlamaktadır. Türkçe yapılan çalışma sayısının geçtiğimiz senelerde oldukça azken günümüzde giderek arttığı gözlemlenmektedir.

Tez kapsamında doğal dil işleme ile Türkçe tweetler üzerinde Duygu Analizi çalışmaları yapılmış, bu alanda çeşitli makine öğrenmesi algoritmaları kullanılarak elde edilen sonuçlar karşılaştırılmıştır. Çalışma kapsamında biri açık diğeri çalışma kapsamında oluşturulmuş iki farklı veri seti kullanılmıştır. Bu verisetleri algoritmalarla kullanılmadan önce çeşitli ön işlemlerden geçirilmiştir. Bu ön işlemlerden biri olan kök bulma ön işleminde iki farklı kütüphaneden yararlanılmış ve bunların karşılaştırması yapılmıştır. Duygu analizi yapmak amacıyla bu veri setleri Logistik Regresyon, SVM, Bayesian, Rastgele Orman, SGD algoritmaları ile eğitilmiş ve modeller üretilmiştir. Bu algoritmalar dışında bir derin öğrenme ağı olan LSTM ile veri setleri ile eğitim yapılarak modeller üretilmiş ve elde edilen sonuçlar karşılaştırılmıştır. Bu çalışma ile literatürde Türkçe dilinde yapılan doğal dil işleme çalışmalarına katkıda bulunmak hedeflenmiştir.

SentimentSet veri seti oluşturulurken tweetlerin içerisinde korona, pandemi gibi bazı kelimeler aranarak oluşturulmuştur. Çalışmada kullanılan verisetlerinin pozitif negatif kategori ağırlıkları verilmiştir. Bu verisetleri ile yapılan eğitimlerde aynı kütüphaneler ve aynı algoritmalar üzerinde yapılan karşılaştırmalarda ele alınan konu sebebiyle negatif veri

ağırlığı yüksek olan SentimentSet veri setinin negatif tahmin doğruluk oranının pozitif tahmin doğruluk oranından yüksek olduğu görülmüştür. Açık veri seti ile eğitilen modellerde pozitif tahmin doğruluk oranının SentimentSet veri setine göre daha yüksek olduğu görülmüştür. Bu durum şu bilineni doğrulamaktadır, sınıflandırma problemlerinde veri setinin eşit ağırlıklara sahip olması makine öğrenmesi algoritmaları için oldukça önemlidir. Ancak gerçek hayat problemlerinde bu eşit sayıyı yakalamak daha zordur. ‘Pandemi’, ‘Korona’ gibi kelimelerin olduğu tweetler genel olarak negatif anlam taşımaktadır.

Tez kapsamında görüldüğü gibi verinin kalitesi en az modellerin oluşturulması kadar önemlidir. Öte yandan kök bulma algoritmalarının başarısı veri setine ve test edilen veriye göre farklılık göstermektedir. SentimentSet ile eğitilen modellerin, açık veri seti ile eğitilen modellere göre kendi içindeki veri setinden ayrılan test verisi ile başarı oranlarının daha yüksek olduğu görülmektedir. Fakat bu durum SentimentSet veri setinde negatif kategori ağırlığının fazla olmasının bir sonucu olabilir. Tez kapsamında üretilen modeller iyileştirilebilir ve daha iyi sonuçlar veren modeller üretilebilir. Bunun için izlenmesi gereken yollardan biri de verinin kalitesini artırmaktır. Daha iyi filtreleme, ön işlemlerin detaylandırılması ve çeşitlendirilmesi gibi çözümler üretilebilir.

Çalışma kapsamında kullanılan algoritmaların hiper parametreleri değiştirilerek daha iyi eğitim sağlanabilir. Tez kapsamında kullanılan LSTM algoritmasının hiper parametreleri deneme yanılma yöntemi ile belirlenmiştir. Gelecekteki çalışmalarda bu modelin hiper parametreleri seçilirken genetik algoritma gibi yöntemlerden yararlanarak en uygun değerler belirlenebilir.

Tez kapsamında oluşturulan modeller ve sonuçlar göstermektedir ki Türkçe dili ile duygu analizi konusunda makine öğrenmesi algoritmaları oldukça başarılıdır. Bu başarının artırılması, daha büyük veriler ve daha iyi modeller üretilmesi elbette ki mümkündür. Modellerin zafiyetleri giderilip veri kalitesi artırılarak yapılan Türkçe çalışmalar daha da geliştirilebilir. Böylece Türkçe duygu analizi görevlerini gerçekleştiren modeller çok daha

yüksek başarı oranları ile hayatımıza etki etmeye ve teknoloji ilerledikçe onunla beraber gelişmeye devam edebilirler.



KAYNAKLAR

- Anonymus. 2013. Web Sitesi: <https://web.archive.org/web/20210308175448/https://brightplanet.com/2013/06/25/twitter-firehose-vs-twitter-api-whats-the-difference-and-why-should-you-care/>, Erişim tarihi: 15.05.2021
- Anonymus. 2017. Web Sitesi: <https://web.archive.org/web/20210302201905/https://www.exastax.com.tr/makine-ogrenimi/yapay-zeka-ve-makine-ogrenimi-arasindaki-fark-nedir/>, Erişim tarihi: 06.06.2021
- Anonymus. 2018. Web Sitesi: <https://web.archive.org/web/20201028004630/https://www.geeksforgeeks.org/creating-linear-kernel-svm-in-python/>, Erişim tarihi: 06.06.2021
- Anonymus. 2020a. Web Sitesi: <https://web.archive.org/web/20210226004046/https://www.geeksforgeeks.org/naive-bayes-classifiers/>, Erişim tarihi: 16.05.2021
- Anonymus. 2020b. Web Sitesi: <https://web.archive.org/web/20201212182100/https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>, Erişim tarihi: 16.05.2021
- Anonymus. 2020c. Web Sitesi: <https://web.archive.org/web/20210502164622/https://scikit-learn.org/stable/modules/sgd.html>, Erişim tarihi: 16.05.2021
- Anonymus. 2021a. Web Sitesi: https://web.archive.org/web/20210506094022/https://en.wikipedia.org/wiki/Natural_language_processing, Erişim tarihi: 06.05.2021
- Anonymus. 2021b. Web Sitesi: https://web.archive.org/web/20201112224440/https://www.tutorialspoint.com/scikit_learn/scikit_learn_stochastic_gradient_descent.htm, Erişim tarihi: 16.05.2021
- Anonymus. 2021d. Web Sitesi: <https://web.archive.org/web/20210507030818/https://github.com/JustAnotherArchivist/snsrape/blob/master/snsrape/modules/twitter.py>, Erişim tarihi: 27.05.2021
- Akba, F. 2014. Assesment of Feature Selection Metrics for Sentiment Analysis: Turkish Movie Reviews. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Ankara.
- Akin, A. A., Akin, M. D. 2019. Web Sitesi: <https://web.archive.org/web/20210415054058/https://github.com/ahmetaa/zemberek-nlp>, Erişim tarihi: 11.05.2021

- Bahl, L. R., Jelinek, F. ve Mercer, R. L. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(2), 179-190.
- Ballı, Ç. ve Ballı, T. 2021. Web Sitesi: <https://www.kaggle.com/caglaballi/sentimentset>, Erişim tarihi: 15.07.2021
- Beyaz, M. 2020. Web Sitesi: <https://www.kaggle.com/mrtbeyz/trke-sosyal-medya-paylam-veri-seti>, Erişim tarihi: 26.05.2021
- Bozyiğit, A., Utku, S ve Nasiboğlu, E. 2019. Cyberbullying Detection by Using Artificial Neural Network Models. In Proceedings of 2019 4th International Conference on Computer Science and Engineering (UBMK). IEEE, 520–524.
- Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J., Mercer, R. ve Roossin, P. 1990. A Statistical Approach to Machine Translation. Comput. Linguistics, 16(2), 79-85.
- Brownlee, J. 2017. Web Sitesi: <https://web.archive.org/web/20210116064720/https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/>, Erişim tarihi: 06.06.2021
- Brownlee, J. 2020. Web Sitesi: <https://web.archive.org/web/20210506173431/https://machinelearningmastery.com/naive-bayes-for-machine-learning/>, Erişim tarihi: 15.05.2021
- Cerebro. 2018. Web Sitesi: <https://web.archive.org/web/20201125200831/https://medium.com/türkiye/yapay-zekanın-tarihçesi-ve-gelişim-süreci-cb4c73deb01d>, Erişim tarihi: 03.05.2021
- Çoban, Ö., Özyer, B. and Özyer, G. T., Sentiment analysis for Turkish Twitter feeds. In Proceedings of the 23rd Signal Processing and Communications Applications Conference (SIU), 2015, 2388-2391.
- Çöltekin, Ç. 2020. A Corpus of Turkish Offensive Language on Social Media. In Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), 6174-6184.
- Donges, N. 2019. Web Sitesi: <https://web.archive.org/web/20210415111142/https://builtin.com/data-science/random-forest-algorithm>, Erişim tarihi: 16.05.2021
- Dwivedi, R. 2020. Web Sitesi: <https://web.archive.org/web/20210321215410/https://analyticsindiamag.com/how-to-implement-lstm-rnn-network-for-sentiment-analysis/>, Erişim tarihi: 18.05.2021

- fchollet. 2020. Web Sitesi: https://web.archive.org/web/20210507003609/https://keras.io/guides/sequential_model/, Eriřim tarihi: 06.06.2021
- Ganesan, K. 2019. <https://web.archive.org/web/20210418120546/https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html>, Eriřim tarihi: 08.05.2021
- Hallaç, İ. R. 2018. Web Sitesi: <https://web.archive.org/web/20200718020415/http://buyukveri.firat.edu.tr/2018/06/04/metin-on-isleme-adimlari-icin-keras-tokenizer-sinifi-kullanimi/>, Eriřim tarihi: 06.06.2021
- Hatipoğlu, E. 2018. Web Sitesi: <https://medium.com/@ekrem.hatipoglu/machine-learning-classification-logistic-regression-part-8-b77d2a61aae1>, Eriřim tarihi: 15.05.2021
- IBM. 2020. Web Sitesi: <https://web.archive.org/web/20210503203042/https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>, Eriřim tarihi: 03.05.2021
- Karamollaoğlu, H., Doğru, İ. A., Dörterler M., Utku, A. ve Yıldız, O. 2018. Sentiment Analysis on Turkish Social Media Shares through Lexicon Based Approach, In Proceedings of 3rd International Conference on Computer Science and Engineering (UBMK), 2018, 45-49.
- Kaya, M., Fidan, G. ve Toroslu, I. H. 2012. Sentiment Analysis of Turkish Political News, In proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2012, 174-180.
- Khanna, C. 2020. Web Sitesi: <https://towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe>, Eriřim tarihi: 06.06.2021
- Kirelli, Y. ve Arslankaya, S. 2020, Sentiment analysis of shared tweets on global warming on Twitter with data mining methods: A case study on Turkish language, Computational Intelligence and Neuroscience, vol. 2020, 1904172:1-1904172:9.
- Liu, Z., Lv, X., Liu, K ve Shi, S. 2010. Study on SVM Compared with the other Text Classification Methods. In proceedings of 2010 Second International Workshop on Education Technology and Computer Science, 2010, 219-222.
- Ma, Q. 2002. Natural language processing with neural networks. In Proceedings of Language Engineering Conference, 2002, 45-56.
- Martinez, W. M. 1995. A natural language processor with neural networks, In Proceedings

- of 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, 1995, 3156-3161.
- Nagy, P. 2017. Web Sitesi: <https://www.kaggle.com/ngyptr/lstm-sentiment-analysis-keras>, Erişim tarihi: 07.06.2021
- Noyan, M. 2019. Web Sitesi: <https://merveenoyan.medium.com/doğal-dil-işleme-natural-language-processing-2d7c72daf245>, Erişim tarihi: 08.05.2021
- Pantola, P. 2018. Web Sitesi: https://web.archive.org/web/20190615122850/https://medium.com/@paritosh_30025/natural-language-processing-text-data-vectorization-af2520529cf7, Erişim tarihi: 08.05.2021
- Pervan, N. 2019. Derin öğrenme yaklaşımları kullanarak Türkçe metinlerden anlamsal çıkarım yapma. Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Ankara.
- Reddy, V. 2018. Web Sitesi: <https://medium.com/@vasista/sentiment-analysis-using-svm-338d418e3ff1>, Erişim tarihi: 16.05.2021
- Rumelli, M., Akkuş, D., Kart, Ö. ve Isik Z. 2019. Sentiment Analysis in Turkish Text with Machine Learning Algorithms, In Proceedings of 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), 2019, 1-5.
- Safak, H. I. 2017. Web Sitesi: <https://web.archive.org/web/20190127071042/https://medium.com/t%C3%BCrkiye/makine-%C3%B6%C4%9Frenmesi-nedir-20dee450b56e>, Erişim tarihi: 06.06.2021
- Son, J. 2015. Web Sitesi: <https://github.com/Samsung/KnowledgeSharingPlatform/blob/master/sameas/lib/lucene-analyzers-common-5.0.0/org/apache/lucene/analysis/tr/stopwords.txt>, Erişim tarihi: 29.05.2021
- Searle, J. 1980. Minds, brains, and programs. Behavioral and Brain Sciences, 3(3), 417–424.
- Shi H. ve Li X. 2011. A sentiment analysis model for hotel reviews based on supervised learning. In Proceedings of 2011 International Conference on Machine Learning and Cybernetics, 2011, 950-954.
- Swaminathan, S. 2018. Web Sitesi: <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>, Erişim tarihi: 15.05.2021
- Tam, S., Said, R., B. ve Tanrıöver, Ö., Ö. 2021. A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification, in IEEE Access, vol. 9, 41283-41293.

- Turing, A. M. 1950. Computing Machinery and Intelligence. *Mind*, 59(236), 433–460.
- Twitter. 2021a. Web Sitesi: <https://web.archive.org/web/20210405155212/https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api>, Erişim tarihi: 13.05.2021
- Twitter. 2021b. Web Sitesi: <https://web.archive.org/web/20210326114416/https://github.com/twitterdev/Twitter-API-v2-sample-code>, Erişim tarihi: 15.05.2021
- Yao, L. ve Guan. Y., 2018. An Improved LSTM Structure for Natural Language Processing. In *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, 2018, 565-569,
- Yener, Y. 2020. Web Sitesi: <https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d>, Erişim tarihi: 15.05.2021
- Yi, J., Nasukawa, T., Bunescu, R ve Niblack, W. 2003. Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of Third IEEE International Conference on Data Mining*, 427-434.
- Zeyrek, B. 2020a. Web Sitesi: <https://medium.com/@bernazeyrekk/makine-öğrenmesi-ile-covid-19-türkçe-tweetlerde-duygu-analizi-79f1acd8cbdf>, Erişim tarihi: 06.06.2021
- Zeyrek, B. 2020b. Web Sitesi: <https://github.com/BernaGulZeyrek/Covid-19/blob/master/covid-19.ipynb>, Erişim tarihi: 06.06.2021