T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

# TWITTER VERİLERİ KULLANILARAK KORONAVİRUS AŞILARI HAKKINDAKİ KAMU ALGISININ ZAMAN İÇİNDEKİ DEĞİŞİMİNİN YAPAY ZEKA DESTEKLİ DUYGU ANALİZİ İLE İNCELENMESİ

Danışman:
Dr. Öğr. Üyesi Muammer AKÇAY

Hazırlayan:
Uğur ERTOY

Kütahya – 2022

**Kabul ve Onay**

Lisansüstü Eğitim Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından

Bilgisayar Mühendisliği Anabilim dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

| **Tez Jürisi** | **İmza** | |
|---|---|---|
| | **Kabul** | **Red** |
| Dr. Öğr. Üyesi Muammer AKÇAY (Danışman) | | |
| Prof. Dr. Eyyüp GÜLBANDILAR | | |
| Prof. Dr. Alpaslan DUYSAK | | |

**Onay**

İmza

Doç. Dr. Arif KOLAY

Enstitü Müdürü

**Bilimsel Etik Bildirimi**

Yüksek Lisans Tezi olarak hazırladığım "Twitter Verileri Kullanılarak Koronavirus Aşıları Hakkındaki Kamu Algısının Zaman İçindeki Değişiminin Yapay Zeka Destekli Duygu Analizi İle İncelenmesi" adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel ve etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

01/05/2022

Imza

Uğur ERTOY

# ÖZET

## TWITTER VERİLERİ KULLANILARAK KORONAVİRUS AŞILARI HAKKINDAKİ KAMU ALGISININ ZAMAN İÇİNDEKİ DEĞİŞİMİNİN YAPAY ZEKA DESTEKLİ DUYGU ANALİZİ İLE İNCELENMESİ

**ERTOY, Uğur**
**Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı**
**Tez Danışmanı: Dr. Öğr. Üyesi Muammer AKÇAY**
**Mayıs, 2022, 86 Sayfa**

Sosyal medya verileri artık günümüzde organik olarak gerçekçi duygu analizleri yapmak için bir çok imkan sağlamaktadır. Bunlardan bir tanesi de insanların bir konu hakkında nasıl hissettiklerini keşfetmek için sayısız Twitter mesajından anlamlı duygu ayrımları yapmaktır. Kovid19 tam olarak anlaşılamayan bilimsel ve tıbbi olarak yeni bir hastalıktır. Bu hastalığa karşı geliştirilen aşılar hakkında insanlar fikir ayrılığına düşmüş durumdadır.

Bu çalışmada Twitter kullanıcıları tarafından İngilizce dilinde yayınlanan Kovid19 aşılarına ilişkin mesajlara göre kamu algısı analiz edilmiştir.

Twitter mesajları genelde kısa uzunlukta olmakta, emojiler, ironiler ya da çeşitli imla hataları içerebilmektedir. Bu nedenle yapılan analizlerden doğru sonuçlar alabilmek için analiz öncesi verilerin işlenmesi gerekmektedir. Doğal dil işleme algoritmaları ve Python yazılım dili kullanılarak Colab ortamında bir model geliştirilmiştir. Bu model yardımıyla veriler analize uygun hale getirilmiştir. Daha sonra bu veriler üzerinde VADER duygu analizi, zaman serisi analizi ve BERT makine öğrenmesi analizi uygulanmıştır. Aynı koşullar altında ulaşılan sonuçlar ve bu sonuçlara ait çıktılar sunulmuştur.

**Anahtar Kelimeler:** Koronavirüs, Kovid-19, Aşı, Salgın, Büyük Veri,
Denetimli Öğrenme, Duygu Analizi, Twitter, BERT, VADER.

# ABSTRACT

## INVESTIGATION OF TREND CHANGE IN PUBLIC PERCEPTION OF CORONAVIRUS VACCINES OVER TIME USING TWITTER DATA WITH ARTIFICIAL INTELLIGENCE-ASSISTED SENTIMENT ANALYSIS

**ERTOY, Uğur**

**M.S.Thesis, Department of Computer Engineering, 2020 Thesis**

**Supervisor: Asst. Prof. Muammer AKÇAY**

**May, 2022, 86 Pages**

As of today, social media data provides great opportunities and supports us to study on realistic sentiment analysis with real life data. One of the sentiment analysis is to put forward meaningful sentiment distinctions from many Tweets to explore how people feel about an issue or a topic. Covid-19 is a new disease that is not yet fully understood. People are divided about vaccination developed against this disease.

In this study, an analysis was conducted on the trend of public perception over time by using Twitter messages published in English regarding Covid-19 vaccines.

Tweets are generally short in length and may contain emoticons, ironies, allusions or various spelling mistakes in written messages. From this perspective, it is hardly requirement to process the data before the analysis in order to get accurate results from the data analysis. A model was developed in the Google Colab environment using natural language processing algorithms and Python software language. The data were maintained for the analysis with the help of this model. Later on, VADER sentiment analysis, time series analysis and BERT machine learning analysis were applied on these data. The results obtained under the same conditions totally and the outputs of the results are presented.

**Keywords:** Coronavirus, Covid-19, Vaccine, Pandemic, Big Data, AI,

Supervised Learning, Sentiment Analysis, Twitter, BERT, VADER.

# ÖNSÖZ

Yüksek lisans çalışmalarımda ilk günden beri bana hep inanan ve destekleyen, bu çalışmanın hazırlık süresince bilimsel katkısını, desteğini ve yardımlarını esirgemeyen, her türlü bilgi birikimini ve tecrübelerini paylaşan, ailesiyle geçirdiği zamanlarda dahi zamanını vakfeden kıymetli danışman hocam Dr. Öğr. Üyesi Muammer AKÇAY'a şükranlarımı sunarım.

Bu çalışma süresince yanımda olan, maddi ve manevi olarak her türlü desteği sunan ve yardımcı olan eşime, babama, anneme, kardeşime ve arkadaşlarıma da müteşekkir olduğumu belirtmek isterim.

## İÇİNDEKİLER

**Sayfa**

# TABLOLAR LİSTESİ

**Sayfa**

# ŞEKİLLER LİSTESİ

**Sayfa**

# KISALTMALAR

| | |
|---|---|
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **NLP** | Natural Language Processing |
| **AUC** | Area Under Curve |
| **ROC** | Receiver Operating Curve |
| **ANN** | Artificial Neural Network |
| **YSA** | Yapar Sinir Ağları |
| **CRISP** | Çapraz Endüstri Süreç Modeli |
| **DSÖ** | Dünya Sağlık Örgütü |
| **VADER** | Valence Aware Dictionary and Sentiment Reasoner |
| **API** | Application Programming Interface |

# GİRİŞ

# GİRİŞ

Koronavirüs, Ocak 2020'den bu yana Twitter'da hareketli gündem maddelerinden biri ve bugüne kadar incelenmeye devam etmektedir. 31 Aralık 2019'da Wuhan şehrinde karşılaşılan ilk vaka ile gündeme gelmiştir ve ilerleyen günlerde Dünya Sağlık Örgütü tarafından Yeni Tip Koronavirüs (Novel Coronavirus(COVID-19)) olarak adlandırılmıştır. Dünya Sağlık Örgütü, Mart 2020'de koronavirüsün yüz on dört ülkede yüz on sekiz binden fazla vaka sayısı ortaya çıkınca bir pandemi olarak ilan etmiştir (www.who.int, 2021) .

26 Eylül 2021'e kadar 4.75 milyon onaylanmış ölüm ve 232.28 milyon onaylanmış Kovid19 vakası ve Dünya nüfusunun %44,3'ü aşı olmuştur (ourworldindata.org, 2021). Kovid19 aşısı yaygınlaşmaya başladığından beri durum düzelmeye başlamıştır. Aşının bulaşma üzerindeki olumlu etkilerine dair daha fazla kanıt elde ettikçe, halkın güvenini güçlendirmeye yardımcı olacaktır. Bunu göz önünde bulundurarak, kamuoyunu veya duyarlılığını analiz etmek, insanları Kovid19'a karşı aşı olmaya motive etmek için çok önemlidir.

11 Aralık 2020 tarihinde ABD İlaç ve Gıda Dairesi (FDA) tarafından Pfizer-Biontech'in geliştirdiği aşının acil kullanım onayı almasının ardından dünya çapında heyecan ve rahatlama dalgaları oluşmuştur. Bununla birlikte Kovid19 aşı tartışmaları ve halk arasındaki kutuplaşmalar başlamıştır. Dünya Sağlık Örgütü (DSÖ) raporları, bazı insanların Kovid19 aşılarını yaptırma konusunda tereddüt ettiğini gösteriyor. DSÖ, 2019 yılında küresel sağlığa yönelik en büyük tehditlerden birinin aşı tereddüdü olduğunu belirtmiştir.

Bu nedenle, Kovid19 aşıları hakkındaki bilgilere erişmek ve toplumların duygu durumlarının, aşı hakkındaki düşüncelerinin ne olduğunun ve toplumdaki kutuplaşma düzeyinin anlaşılarak buna yönelik uygulamaların geliştirilmesi devletler için önem arz etmektedir. Günümüzde sosyal medyanın yaygın kullanımı nedeniyle sokaklarda anket yapmak yerine her konuda çeşitli sosyal medya ortamları aracılığıyla birçok bilgiye erişilebilmektedir.

Günümüzde sosyal medya kullanıcılarının sayısı hızla artmıştır ve görüş içerikli metinlerin elde edilmesi oldukça kolay hale gelmiştir. Facebook, Twitter, Youtube, Linkedin önceki yıllara göre ciddi artışlar gördüler. Facebook, günlük 1,9 milyar aktif kullanıcıya sahiptir (www.statista.com, 2021). Twitter sosyal medya platformu ise günlük

yaklaşık 206 milyon aktif kullanıcılı bir trafiğe sahiptir (www.statista.com, 2021). Twitter hızlı bir gelişme göstermekte ve dünyanın tüm bölgelerinde günden güne ün kazanmaktadır. Eskiden anket yaparak kamuoyu yoklaması yapmak gibi zorluklarla halktan veri toplamaya çalışılırken, günümüz dünyasında online sosyal iletişim platformları sayesinde daha büyük örnek büyüklüklerine sahip daha çeşitli ve nispeten daha sağlıklı dağılımlar gösteren veriler elde edilmektedir. Sosyal medyada yedi gün yirmi dört saat boyunca dünyanın her bir köşesindeki insanlar tarafından çok miktarda mesaj paylaşılmaktadır. Bu noktada sosyal medya üzerinden toplanan verilerden insanların görüşlerinin anlaşılması ve analiz edilmesi bütün alanlarda, belirli kullanıcılar tarafından farklı bakış açılarına yardımcı olmak için, örneğin, siyasi misyonlar ve amaçlanan bilgileri yayma gibi konularda ciddi bir kullanımı söz konusudur. Twitter ise aynı zamanda belirli karakter sınırıyla sadece mesaj içerikli bir platform olduğundan duygu analizi için de en uygun ortamlardan birisidir. Twitter, duygu analizi için gereken veri setlerini oluşturmak için her kullanıcıya bu imkanı sunmamaktadır ve bu verisetlerini oluşturabilmek için öncelikle Twitter Geliştirici hesabı edinmek gerekmektedir.

Makine öğrenmesi son yıllara kadar hayal bile edilemeyecek teknolojik başarıların ve araçların yolunu açan veri bilimindeki en yeni yöntemdir. Yüz tanıma, parmak izi tanıma, hareketli nesne tanıma, hareket türü algılama, duygu analizi, nesne tanıma, sıcaklık algılama, ürün önerileri, oltalama algılama, sosyal medya özellikleri, trafik yoğunluğu vb. hayatımızın hemen her alanında kullanmakta olduğumuz uygulamalardan bazılarıdır. Makineler yazılımlardaki algoritmalar sayesinde insan davranışlarını öğrenerek insanları taklit etmektedir ya da anlayarak tepki vermektedir. Duygu analizi, hayata dair tüm görüşlerin analizi ile ilgilenen Doğal Dil İşleme alanına dahil olup, görüşlere ait sınıflandırmalara, kutuplaşmalara odaklanmaktadır.

Duygu ve düşünce analizi araştırmaları alanında başvurulabilecek birden fazla farklı yöntem söz konusu olmakla birlikte, araştırmaların temelinde veri madenciliği bulunduğundan esası oluşturan yöntemler makine öğrenimi ve dil işleme algoritmalarıdır.

Dil işleme algoritmalarında, sözcük/sözlük temelli kelimelerin kutup değerlerine bağlı olarak olumlu, olumsuz ya da tarafsız şeklinde sınıflandırmalar bulunmaktadır. Makine öğrenimi algoritmalarında, herhangi bir mesajın, ifadenin, cümlenin tümüne yayılan duygu ve düşüncenin tanımlanması için eğitim veri dosyaları kullanılarak sınıflandırmalar yapılmaktadır.

Sözbilimsel yaklaşımın temelini sözcüklerin ifade ettiği duygu ve görüşlerin birlikte yer aldığı sözlük benzeri veri tabanları oluşturmaktadır. Wordnet, HowNet, SentiWordNet, SenticNet bu veri tabanlarından bazılarıdır.

Duygu ve düşünce analizi araştırmalarına ait önceki çalışmalar göz önünde bulundurulduğunda sınıflandırma, kutup tespiti teknikleri kullanıldığı gözlemlenmiştir. Bu çalışmada ise sözcük tabanlı VADER duygu analizi algoritması ve aynı koşullar altında Google BERT makine öğrenmesi algoritması karşılaştırması yapılmıştır.

Genel bakış, bu tezde duygu analizini ve zaman serisi analizini gerçekleştirmek için kullanılan yöntemleri ve yaklaşımları açıklar. Birinci bölümde, literatür çalışmalarına yer verilmiştir. İkinci bölümde veri madenciliği ve veri madenciliği yöntemlerine yer verilmiştir. Doğal Dil İşleme yaklaşımı, duygu analizi ve yaklaşımları, duygu analizi gerçekleştirmek için kullanılan araçlar ve zaman serisi analizi yaklaşımı gibi kullanılan çeşitli yaklaşımlarından bahsedilmiştir. Üçüncü bölümde çalışmada kullanılan materyal ve yöntemlere yer verilmiştir. Dördüncü bölümde ortam kurulumu, veri toplama, toplanan verilerin nasıl önceden işlendiği, tweet duygularının analizi ve tahmin edilen duyarlılıklar ve deneyde kullanılan diğer araçlar üzerindeki zaman serisi analizi ve deneysel kurulum bilgileri gibi deneysel çalışmalara yer verilmiştir. Beşinci bölümde, elde edilen sonuçlar gösterilmektedir. Altıncı bölümde analiz sonuçlarına ve modellerin karşılaştırılmasına yer verilmiştir.

Son bölümde çalışma sonucunda varılan sonuçlar paylaşılarak, çalışmanın literatüre katkılarından ve sonraki aşamalarda yapılabilecek önerilerden bahsedilmiştir.

# BİRİNCİ BÖLÜM

# LİTERATÜR TARAMASI

## 1.1. Literatür Taraması

(Gilbert, E. ve Hutto, C.J., 2021)'da özellikle sosyal medyada ifade edilen duygulara uyum sağlayan ve diğer alanlardaki metinlerde iyi çalışan bir sözlük ve genel duygu ve düşünce kutuplaşması için geliştirilmiş olan sözlük tabanlı VADER (Valence Aware Dictionary and Sentiment Reasoner) algoritmasını geliştirdiler. Niteliksel ve niceliksel yöntemlerin bir kombinasyonu kullanılarak, özellikle mikroblog benzeri bağlamlarda duygu yoğunluğuna göre ayarlanmış standart bir sözcük listesi oluşturmuştur. Özellikle İngilizce Sözcükler için Duygusal Normlar (ANEW), Dil Sorgulama ve Kelime Sayımı (LIWC), Genel Sorgulayıcı, SentiWordNet ve Naive Bayes, Maksimum Entropi ve Destek Vektör Makinesi (SVM) algoritmalarına dayanan makine öğrenimi odaklı teknikler gibi on bir önde gelen duygu analizi aracı kadar dikkate değer ve hatta daha iyi sonuçlar göstermiştir. Bu araştırma çalışması VADER'in oluşumunu, doğrulamasını ve test edilmesini ortaya koymuştur. VADER, mesajların duyarlılığını değerlendirmek için basit bir kural tabanlı model kullanır. Bulgulara göre VADER, LIWC gibi geleneksel duygu sözlüklerinin avantajlarını artırmıştır. VADER, sosyal medya ortamlarındaki duygu ifadelerine daha duyarlı olması ve diğer alanlara daha yakın genelleme yapmasıyla kendisini LIWC'den ayırmıştır.

(Davidov, D., Tsur, O. ve Rappoport, A., 2010)'da yazarlar, 50 Twitter etiketi ve 15 gülen emojiyi duygu eğitimi etiketi olarak kullanmaktadır. Bu şekilde Twitter mesajlarının bu özelliğinden faydalanırlar ve ";)" gülen yüz veya "#mutlu" etiketini içeren bir mesaja oldukça olumlu bir puan vermeleri kolaylaşır. Ancak makalenin kendisi bunun göründüğünden çok daha zor olduğunu açıkça ortaya koyuyor çünkü insanlar sıklıkla aynı cümlede zıtlık duygularını ifade ediyorlar ve etiketler bu noktada pek yardımcı olmuyor gibi görünüyor. Örneğin, "birkaç gün içinde bitecek mutlu eğitim günleri #üzgün #mutlu". Çoğunlukla sözlü konuşmada kullanılan söz dizimsel olarak tutarsız kelimeler ve ifadeler dışında, Twitter kullanıcıları özel etiketler ve emojiler kullanırlar, bu durum diğer belgeler için uygulanan duygu analizi yaklaşımlarını uygulamayı daha da zorlaştırır.

(Barbosa, L., Feng. J., 2010)'da, Twitter mesajlarındaki duyguları otomatik olarak tespit etmek için bir yaklaşım sunulmaktadır. Buradaki kapsam, öznel mesajları bir bütün olarak üç kategoriye (olumlu, tarafsız, olumsuz) ayırmaktır. Karakter kısıtlaması nedeniyle mesajlar her zaman söz dizimsel olarak tutarlı kelimeler içermediğinden, önyargılı ve gürültülü etiketleri girdi olarak dikkate alır.

(Lwin, M. O. ve diğerleri, 2020)'da Twitter kullanıcılarının Kovid19'a karşı genel duygu ve düşünceleri incelenmiş dört temel duyguya (korku, öfke, üzüntü ve neşe) odaklanmıştır.

(Hoang, M. Ve Bihorac, O. A., 2019)'da bakış açısına dayalı duygu analizi ve BERT modeli birlikte kullanılarak BERT modelinin tek başına olan performansından daha iyi sonuç gösterdiğinin kanıtlanması amaçlanmıştır. Ancak bu analiz hem duyguları hem de yönleri tanımlamayı içeren karmaşık bir modeldir. Önceden eğitilmiş bir BERT modelinin yanı sıra ilgili ve ilgisiz etiketlerden oluşan cümle çiftleri ile duygu sınıflandırmak için modele ek eğitimler gerekmiştir.

Önceki çalışmaları karşılaştırırken çoğu araştırmacı doğal dil işleme algoritma paketini, yapay sinir ağlarını ya da dilbilimsel yaklaşımları kullandıkları görülmektedir. Bu çalışmada, Kovid19 aşılarının gündeme gelmeye başladığı ilk aşamalarından 2021 yılı sonlarına doğru, Twitter'daki mesajlardan halkın duygularının, inançlarının ve genel düşüncelerinin ne yönde eğilim gösterdiğini anlamak için VADER duygu analizi ve BERT makine öğrenmesi kullanılmıştır. Bu modeller arasında en iyi performans gösteren modelin ortaya konması amaçlanmaktadır.

# İKİNCİ BÖLÜM

# VERİ MADENCİLİĞİ TANIMI

## 2.1. Veri Madenciliği Tanımı

Veri madenciliği, akla gelebilecek her türlü bilgiye veri denmekte ve veriler yığını içerisinden anlamlı ve kullanılabilir bilgiyi elde etmek için kullanılan yöntemlerle ilgilenen alan olarak tanımlanabilir. Veri madenciliği temeli istatistik, yapay zeka ve makine öğrenmesi olarak üç farklı bilimsel disiplinin kesişiminden oluşmaktadır. Bir başka deyişle büyük veri yığınları içindeki sonuçları tahminlemek için veri setleri içerisinde yer alan normal dışı durumları ve korelasyonları bulma süreci de denebilir.

Yapay zeka ile birlikte makine öğrenmesinin teknolojik olarak geliştirilmesinin sebebi, günümüzde artan internet kullanımıyla birlikte her dakika üstel olarak artan verilerin temizlenerek, odaklanılacak veri havuzuna yardımcı olmaktır. Dünya üzerinde artan veri miktarı, depolama ve saklama için gerekli olan alan ihtiyacında da ciddi artış oluşturması sebebiyle verilerin değerlendirilmesiyle alakalı yeni çözüm arayışları oluşmuştur. Üretilen veriler bilgisayarların işleyebileceği Büyük veri havuzlarının birçok teknikle analiz edilerek maliyetleri azaltmak, gelirleri artırmak, müşteri ilişkilerini ve deneyimlerini artırmak, çeşitli olası riskleri bertaraf etmek ya da azaltmak, stok yönetimini tam zamanında tam yerinde prensibine göre yönetmeye çalışmak gibi çeşitli amaçlara yönelik veri madenciliği yapılabilir.

Tarihsel olarak 1962 yılında ilk makine öğrenimi algoritması Jonathan D. Rosenblatt tarafından sunulmuştur. Daha sonra 1980 yılının ikinci yarısında yapay sinir ağları ortaya çıkmaya başlamıştır. Bu dönem içerisinde bazı akademisyenler karar ağacı teoremi ile yapay sinir ağlarını sınıflandırma çözümleri için kullanılabilir noktaya getirmişlerdir. İstatistik bilimi insanlık tarihi boyunca modelleme çalışmalarında bir araç olarak süregelmiş ve bu gelişmelerle birlikte bilgi işleme teknolojilerinin ilerlemesiyle beraber hesaplama yöntemlerinin bu tür analizlerdeki önemi de giderek artmaya başlamıştır (Paolo, 2003) (Luan ve Terrence, 2000).

## 2.2. Veri Madenciliği Aşamaları

### 2.2.1. Veri Toplama

Yapılacak çalışmada kullanılacak bilgilerin, verilerin ilgili veri tabanından ya da bilgi kaynağından tedarik edilmesi gerekmektedir.

Bu noktada veri tabanlarından bilgileri çekmek için SQL yazılımları kullanılarak sorgulamalar yapılabilmektedir. Python gibi yazılım dillerinde çevrimiçi bir veri tabanından çeşitli kütüphaneler kullanılarak veriler çekilerek güncel bir akış sağlanabilmektedir. Bunların yanı sıra virgülle ayrılmış veri tabanı da denilen .csv formatındaki MS Excel dosyaları da veri setleri olarak kaydedilebilmekte ve çalışmalarda kullanılabilmektedir.

Yapılacak çalışmanın amacına uygun verileri toplamak için çeşitli SQL sorguları kullanılmaktadır. Web sitelerinden verilerin toplanması için webscrap araçları kullanılmaktadır. Sosyal medya sitelerinin geliştirici hesapları edinilerek API (Application Programming Interface) bağlantıları ile bu platformlarda oluşan anlık veriler çekilebilmektedir.

Belirli bir zaman dilimine ait veri dosyalarının kullanımı için ise yaygın kullanımı bulunan Kaggle ve türevi web sitelerinde açık kaynak olarak paylaşılan virgülle ayrılmış veri dosyaları olarak adlandırılan .csv formatındaki dosyalar indirilmektedir. Temel olarak bu veri dosyaları düz metin formunda bulunmaktadır. Python gibi yazılım dillerinde bu veri dosyalarının kullanılabilir/anlaşılabilir hale gelmesi için ayrıştırıcı kullanılması gerekmektedir (www.towardsdatascience.com, 2021).

### 2.2.2. Ön İşleme

Verileri elde ettikten sonra yapılacak ilk şey verileri temizlemektir. Bu aşamada, verilerin mevcut bir türden başka türe değiştirilmesi ve standartlaştırılması gerekmektedir. Örnek vermek gerekirse, birden fazla veri tabanı dosyası ile çalışmak yerine bu veri tabanı dosyaları aynı dosyada birleştirilerek ve standartlaştırılarak analize tek bir kaynak dosya olarak girdi sağlanabilir.

Verilerin temizlenmesi, değerlerin çıkarılması ve değiştirilmesi de bu kısımda yapılmaktadır. Eksik veri kümeleri olduğu veya bunların değersiz gibi görünebileceği düşünülürse, bunları uygun şekilde değiştirilmesi gerekmektedir.

Sütunların bölünmesi, birleştirilmesi ve çıkarılması da gerekecektir. Örneğin, bir sütunda daha küçük bir yerleşke belirtilirken, diğer sütunda daha geniş olan yerleşke belirtiliyorsa ve çalışmada bu verilerin aynı çatı altında toplanması gerekebilir. Keza tam tersine tek sütunda yer alan benzer veriler söz konusu ise bunların ayrı sütunlara dağıtılması da gerekebilir.

Bu aşama, verilerin dizgilenmesi, düzenlenmesi, ihtiyaç duyulmayacak bilgilerin temizlenmesi, standartlaştırılması ve ortaklaştırılması olarak düşünülebilir (www.towardsdatascience.com, 2021).

### 2.2.3. İndirgeme

Veriler kullanılmaya hazır olduğunda ve yapay zeka ve/veya makine öğrenimine geçmeden hemen önce verilerin incelenmesi gerekmektedir. Veriler ve özellikleri incelenerek, verilerin farklı veri türlerine göre sayısal, sözel, ardışık vb. şeklinde sınıflandırılması işlemi uygulanır.

Ardından, önemli değişkenleri test etmek amacıyla özellikler çıkarılarak tanımlayıcı istatistikler hesaplanır, hangi değişkenlerin önem arz ettiğini bulmak için korelasyon analizi yapılır. Örnek olarak, obezite hastalığında şeker ve yağ tüketiminin ne kadar etkili olduğunun ilişkilendirilerek araştırılması gibi.

Grafikler, şekiller gibi görselleştirme elemanları kullanılarak verilerde yer alan eğilimler ve verilerdeki anlam bütünlükleri belirlenebilir.

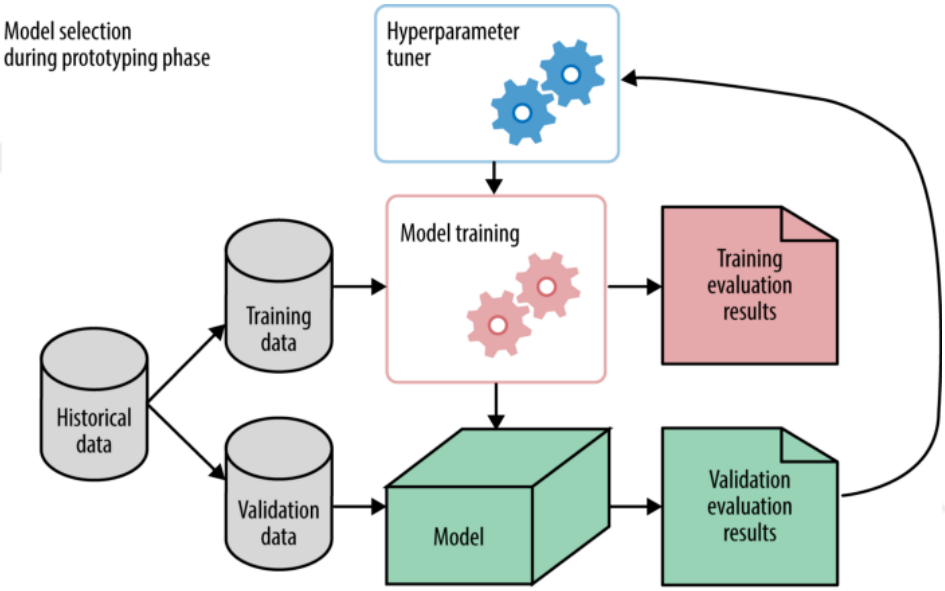
### 2.2.4. Veri Modelleme

Veri modellemenin temelinde hangi işlemlerin gerçekleştirileceğine odaklanılmasının yanı sıra hangi verilerin gerektiğine ve bu verilerin nasıl yapılandırılması gerekliliği bulunmaktadır. Herhangi bir veri tabanında saklanacak veriler için veri modelleri oluşturulmaktadır. Veri modelleme yapılarak veri dosyalarının içerisindeki verilerin ölçeği küçültülerek amaca yönelik gerekli verilerin kullanılması amaçlanmaktadır. Veri kalitesinin iyileştirilmesi, verilerin sınıflandırılması, veri nesnelerinin temsil edilip edilmediğinin kontrol edilmesi, ilişki tabloları oluşturulması, iş gereksinimlerinin ortaya çıkarılması gibi kullanım amaçları bulunmaktadır.

Veri modelleri kavramsal, mantıksal ve fiziksel olarak üç ana başlık altında toplanmaktadır. Hiyerarşik, nesneye yönelik, ağ, varlık ilişkisi ve ilişkisel teknik olarak beş farklı veri modelleme tekniği bulunmaktadır.

Modellemede gerçekleştirilebilecek birkaç görev vardır. Lojistik regresyonlar kullanarak alınan e-postaları "Gelen Kutusu" ve "Spam" olarak ayırt etmek için sınıflandırma gerçekleştirecek modeller de eğitilebilir. Doğrusal regresyonlar kullanılarak değerler de tahmin edilebilir. Bu kümelerin arkasındaki mantığın anlaşılması için verileri gruplandırma modeli de kullanılabilir. Örneğin, IBM şirketinin bilgi yönetim sistemi, öncelikle bankacılık alanında kullanılmaya başlanmış ve daha sonra işletmelerde yaygın kullanıma sahip olmuş hiyerarşik bir veri modellemedir.

**Şekil 2.1:** Veri Modelleme (www.towardsdatascience.com, 2021)




### 2.2.5. Değerlendirme

Bir veri analizinin değerlendirme aşaması algoritmanın yani modelin olgunlaştırılmasında önemli bir aşamasıdır. Üzerinde çalışılan veriler için seçilecek doğru algoritmanın seçilmesinin yanı sıra seçilmiş algoritmanın ne kadar kaliteli çalışacağının bulunmasını sağlamaktadır.

Eğitim aşamasında kullanılan verilerle model performansının değerlendirilmesi, veri biliminde kabul edilemez çünkü eğitim dosyalarında aşırı iyimser ya da uyumlu modeller üretilebilmektedir. Veri madenciliğinde veri değerlendirmenin bekletme ve çapraz doğrulama şeklinde iki yöntemi bulunmaktadır.

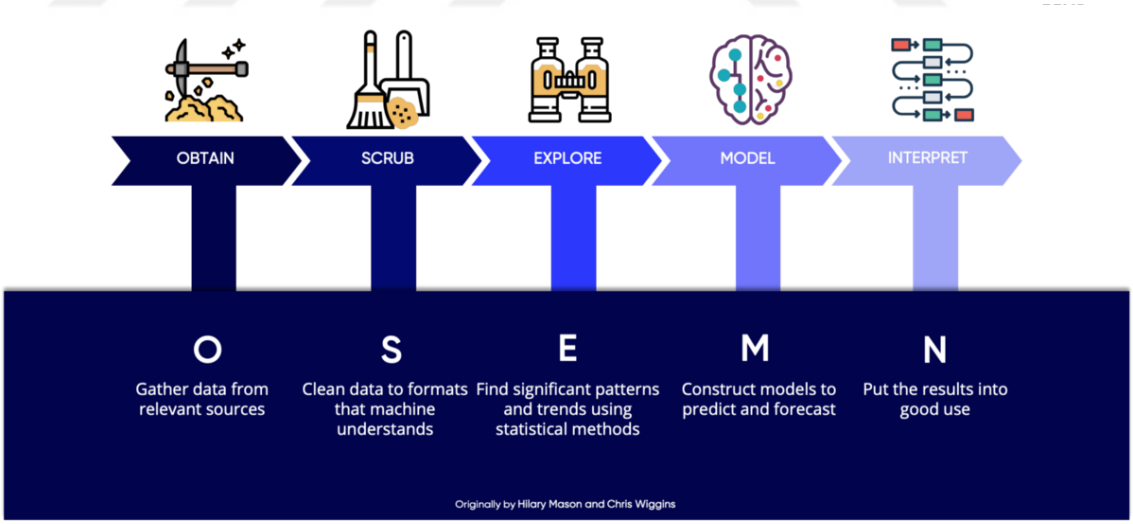
Bekletme yönteminde, genellikle büyük veri kümesi rassal olarak eğitim, doğrulama ve test seti olarak üç alt kategoriye bölünmektedir. Eğitim veri seti, tahmine

dayalı modeller oluşturmak için kullanılan veri kümesinin alt kümesidir. Doğrulama veri seti, eğitim aşamasındaki modelin performansını değerlendirmek için kullanılan veri kümesinin alt kümesidir. Model parametrelerinin ince ayarı yapmak ve en iyi performansı sunan modeli seçmek için bir test platformu sağlar. Tüm modelleme algoritmaları bir doğrulama setine ihtiyaç duymamaktadır. Test veri seti veya görünmeyen örnekler, bir modelin olası gelecekteki performansını değerlendirmek için kullanılan veri kümesidir.

Çapraz doğrulama yönteminde, sadece sınırlı miktarda veri mevcut olduğunda, model performansının tarafsız bir tahminini elde etmek için k-katlama çapraz doğrulaması kullanılmaktadır. K-katlama çapraz doğrulamada, veriler eşit büyüklükteki k alt kümeye bölünmektedir. Modeller k kez oluşturulmakta ve her döngüde bir alt küme eğitimden çıkarılmaktadır ve test seti olarak kullanılmaktadır. Eğer k örnek büyüklüğüne eşitse birini dışarda bırakma olarak adlandırılmaktadır.

Veri modelinin değerlendirilmesi sınıflandırma değerlendirmesi ve regresyon değerlendirmesi olarak iki başlığa bölünebilir (Sayad, 2010).

**Şekil 2.2:** Veri Madenciliği Aşamaları (www.towardsdatascience.com, 2021)




## 2.3. Veri Madenciliği Metodları

Veri madenciliği metodları tahminleme ve betimleme olarak iki başlık altında toplanmaktadır. Bazı kaynaklarda bu ayrım denetimli ve denetimsiz olarak ayrılmaktadır.

Denetimli ve denetimsiz öğrenme arasındaki temel farklılık, etiketlenmiş ve etiketlenmemiş veri setlerinin kullanılmasıdır.

Denetimli öğrenme metodu, etiketlenmiş veri setlerinin kullanımı olan makine öğrenimi yaklaşımıdır. Verileri sınıflandırmak veya doğru sonuçları tahminlemek için algoritmaların eğitilmesi gerekmektedir. Denetimli öğrenme yaklaşımında model, etiketlenmiş girdi ve istenen çıktı değerlerini kullanarak ve doğruluğunu kontrol ederek zamanla öğrenebilmektedir. Denetimli öğrenme metodu da kendi içinde sınıflandırma ve regresyon olarak ikiye ayrılmaktadır.

Sınıflandırma problemlerinde, test verilerinin belirli gruplar altında toplanarak tasnif edilmesi için algoritma kullanılmaktadır. Örneğin, MS Outlook'ta belirli sözcük içeren epostaların belirli bir klasöre aktarılması için denetimli öğrenme kullanılmaktadır. En yakın komşu, karar ağacı ve rassal orman, destek vektör makinesi, yapay sinir ağları, naive-bayes metodu bunlardan bazılarıdır.

Regresyon, yani eğri oluşturma problemlerinde, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi anlamak için algoritma kullanılmaktadır. Regresyon modelleri, bir şirketin satışlarının pazardaki diğer şirketlerle olan ilişkisi ya da tiroid hastalığı ile ilgili verilerden yola çıkarak, hormonlar arasındaki ilişkiyi tahmin etmek için kullanılabilmektedir. Lineer regresyon, lojistik regresyon, polinom regresyon başlıcalarıdır.

Denetimsiz öğrenme metodu, etiketlenmemiş veri setlerini analiz etmek ve kümelemek için kullanılan yaklaşımdır. Denetimsiz algoritmalar, insan müdahalesine gereksinim duymadan verilerdeki kalıpları keşfetmektedir. Kümeleme, ilişkilendirme ve boyut küçültme olarak üç ana başlık altında toplanmaktadır.

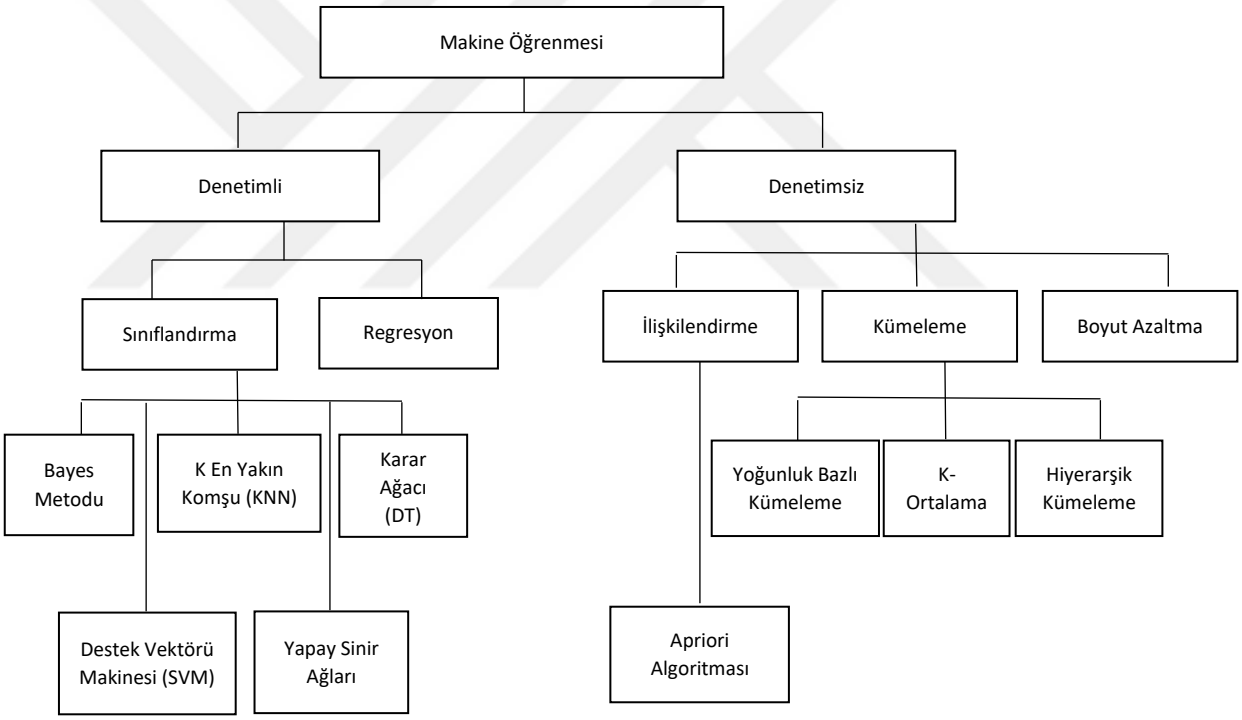
Kümeleme yönteminde, etiketlenmemiş veriler benzer veya farklı yönlerine göre ayrılmaktadır. Örneğin, k-ortalama kümeleme algoritmaları, birbirine benzeyen veri noktalarını gruplandırır, buradaki k değeri, gruplandırmanın boyutunu ve ayrıntı düzeyini ifade etmektedir.

İlişkilendirme yöntemi, belirli bir veri kümesindeki değişkenler arasındaki bağıntıyı bulmak için farklı kurallar kullanılmaktadır. Örneğin, çevrimiçi alışveriş platformlarında müşterilerin x ürününü alırken y ürününü de aldığını gösteren öneriler bu algoritmalarla çözümlenmektedir.

Boyut azaltma yöntemi, belirli bir veri setindeki özelliklerin sayısı çok fazla olduğunda kullanılmaktadır. Veri bütünlüğü korunurken, veri giriş sayısı yönetilebilir bir boyuta indirgenmektedir. Bu yöntem genelde yapay zekanın resim kalitesini iyileştirmek amacıyla görsel verilerden parazit olan verileri çıkarması için kullanılmaktadır.

Genel olarak, denetimli öğrenmede algoritma, veriler üzerinde tekrarlı tahminler yaparak ve doğru çözümü ayarlayarak eğitim veri kümesinden öğrenmektedir. Bu noktada ise verilerin uygun şekilde etiketlenmesi için insan müdahalesi gerekmektedir. Denetimsiz öğrenmede ise algoritma, etiketlenmemiş verilerin doğal yapısını çözebilmek için kendi başına çalışmaktadır (Glymour, Madıgan, Pregibon ve Smyth, 1997)(Kohavi ve Quinlan, 2002).

**Şekil 2.3:** Makine Öğrenmesi Çeşitleri




## 2.4. Duygu Analizi Teknikleri

Bu çalışma, duygu analizi yaklaşımlarıyla birlikte duygu analizi seviyelerini de içermektedir. Duygu analiziyle ilgili temel kavramlara bu bölümde yer verilmiştir.

### 2.4.1. Doğal Dil İşleme

Hem bir dizi teoriye hem de bir dizi teknolojiye dayanan metin analiz etmeye yönelik bilgisayarlı yaklaşım Doğal Dil İşleme olarak adlandırılmaktadır. Çok aktif bir

araştırma ve geliştirme alanı olduğundan, herkesi tatmin edecek, üzerinde anlaşmaya varılan tek bir tanım yoktur (Kent, 2000).

NLP, bilimsel, ekonomik, sosyal ve kültürel olarak önemlidir. NLP teorileri ve yöntemleri çeşitli yeni dil teknolojilerinde kullanıldığı için hızlı bir büyüme göstermektedir. Bu nedenle, NLP hakkında daha çok insanın çalışma bilgisine sahip olması önemlidir. Endüstri içinde buna insan-bilgisayar etkileşimi, iş bilgisi analizi ve web yazılımı geliştirmesinde çalışan insanlar dahildir. Akademik hayat içinde, beşeri bilimler, bilgi işlem ve korpus dilbiliminden bilgisayar bilimi ve yapay zekaya kadar olan alanlardaki insanları içerir. Akademideki birçok insan için NLP, "Bilgisayarlı Dilbilim" adıyla bilinir.

2010 yıllarında temsili öğrenme ve derin sinir ağı tarzı makine öğrenimi yöntemleri, kısmen bu tür tekniklerin başarılı olduğunu gösteren bir dizi sonuç nedeniyle doğal dil işlemede yaygınlaştı

Metinsel Kapsamı Tanıma (Recognizing Textual Entailment(RTE)) gibi görevlerde araştırma odaklı ilerlemelere rağmen, gerçek dünya uygulamaları için geliştirilmiş doğal dil sistemleri hala sağduyulu muhakeme gerçekleştiremez veya dünya bilgisinden genel ve sağlam bir şekilde yararlanamaz. Bu zor yapay zeka problemlerinin çözülmesini bekleyebiliriz ancak bu arada doğal dil sistemlerinin akıl yürütme ve bilgi yetenekleri üzerinde bazı ciddi sınırlamalarla yaşamak gerekiyor. Buna göre, en başından beri NLP araştırmasının en önemli hedefi, sınırsız bilgi ve akıl yürütme yetenekleri yerine yüzeysel ama güçlü teknikleri kullanarak "dili anlayan" teknolojiler inşa etmek gibi zor bir görevde ilerleme kaydetmek olmuştur (Bird, Klein ve Loper, 2009).

### 2.4.2. Doğal Dil Araç Kiti (NLTK)

Doğal Dil Araç Kiti kavramı, ilk olarak 2001 yılında Pennsylvania Üniversitesi, Bilgisayar ve Bilgi Bilimleri Bölümü'ndeki hesaplamalı dilbilim kursunun bir parçası olarak kurulmuştur. Kurulduğu günden günümüze kadar düzinelerce insanın katkısıyla birlikte bu araçlar geliştirilmiştir.

Artık onlarca üniversitede derslerde benimsenmiş ve birçok araştırma projesine temel teşkil etmektedir. En önemli NLTK modülleri Tablo-2.1'de listelenmektedir. Doğal Dil Araç Kiti, dört ana hedef göz önünde bulundurularak tasarlanmıştır:

**Basitlik:** Önemli yapı taşları ile birlikte sezgisel bir çerçeve sağlamak, kullanıcılara genellikle açıklamalı dil verilerinin işlenmesiyle ilişkilendirilen sıkıcı temizlik işlerine saplanmadan pratik bir NLP bilgisi vermek amaçlanmaktadır.

**Tutarlılık:** Tutarlı arayüzler ve veri yapıları ve kolayca tahmin edilebilir yöntem adları ile tek tip bir çerçeve sağlamak amaçlanmaktadır.

**Genişletilebilirlik:** Aynı göreve alternatif uygulamalar ve rakip yaklaşımlar dahil olmak üzere yeni yazılım modüllerinin kolayca yerleştirilebileceği bir yapı sağlamak amaçlanmaktadır.

**Modülerlik:** Araç setinin geri kalanını anlamanıza gerek kalmadan bağımsız olarak kullanılabilecek bileşenler sağlamak amaçlanmaktadır.

**Tablo 2.1: İşlevsellik Örnekleriyle Birlikte Dil İşleme Görevleri Ve İlgili NLTK Modülleri**

| Dil İşleme Görevi | NLTK Modülleri | Fonksiyonellik |
|---|---|---|
| Corporaya erişim | nltk.corpus | Corpora ve sözlüklere standartlaştırılmış arayüzler |
| Dize işleme | nltk.tokenize, nltk.stem | Belirteçler, cümle belirteçleri, kök ayırıcılar |
| Sıralama | nltk.collocations | t-testi, ki-kare, noktasal karşılıklı bilgi |
| Konuşma parçaları etiketleme | nltk.tag | ngram, geri çekilme, Brill, HMM, TnT |
| Sınıflandırma | nltk.cluster, nltk.classify | Karar Ağacı, Maksimum Entropi, k-ortalamalar, NaiveBayesian |
| Kümeleme | nltk.chunk | Normal ifade, n-gram, adlandırılmış varlık |
| Ayrıştırma | nltk.parse | Grafik, özellik tabanlı, birleştirme, olasılık, bağımlılık |
| Anlamsal yorumlama | nltk.sem, nltk.inference | Lambda hesabı, birinci dereceden mantık, model kontrolü |
| Değerleme metrikleri | nltk.metrics | Hassasiyet, geri çağırma, uyum katsayıları |
| Olasılık ve tahminleme | nltk.probability | Frekans dağılımları, düzleştirilmiş olasılık dağılımları |
| Uygulamalar | nltk.app, nltk.chat | Grafiksel bağdaştırıcı, ayrıştırıcılar, WordNet tarayıcısı, sohbet robotları |
| Dilbilimsel çalışma | nltk.toolbox | SIL Toolbox formatına veri manipülasyonu |

### 2.4.3. Duygu Analizi

Duygu analizi, insanların bir varlığa veya bir özneye yönelik duygu veya duygularını tahmin etme yöntemidir. Duygu analizi, çeşitli ilgili metin örneklerini genel olarak olumlu ve olumsuz kategorilere ayırmak için algoritmalar kullanır. Duygusal analize yönelik çoğu yaklaşım, bağlamın gücünü göz önünde bulundurarak, metinleri olumlu veya olumsuz ya da değerlik temelli olarak düzenleyen kutupsallık temelli olarak iki biçimden birini içerir. Örneğin, polariteye dayalı bir yaklaşımda, mükemmel ve iyi sözcükleri aynı şekilde ele alınacaktır. Buna karşılık, değerlik temelli yaklaşımda mükemmel, iyi olduğu kadar olumlu olarak kabul edilecektir. Açıklamalı sözcük kutupluluklarına sahip Doğal Dil İşleme yöntemleri ve sözlükler bu araştırmayı gerçekleştirebilir. İnternette kullanıcı tarafından oluşturulan bilgilerin yaygın olarak bulunması, büyük olasılıkla etkili duygu analizi araştırmalarıyla sonuçlanmıştır (Spencer ve Uchyigit, 2012).

### 2.4.4. Makine Öğrenimi Yaklaşımı

Makine Öğrenimi Yaklaşımı, sözlük tabanlı yaklaşıma kıyasla daha yavaş ve hantal olan, duygu analizi için yaygın yaklaşımlardan biridir. Bir makine öğrenimi yaklaşımı kullanılırken sınıflandırılmış eğitim verilerine ihtiyaç vardır.

Daha önce görülmemiş verilerin sınıflandırılmasını tahmin etmek ve yapılandırmak için sınıflandırılmış eğitim verileri üzerinde algoritmayı eğitecektir. Bu amaçla kullanılan birkaç standart algoritma örneği, Naive Bayes ve Destek Vektör Makineleridir. Tüm bu makine öğrenimi algoritmalarını araştırmak ve test etmek zaman alıcı ve yavaştır. Raporun kapsamının daha iyi olması için sözlük tabanlı yaklaşım seçilmiştir. Makine öğrenimi yaklaşımı daha fazla araştırılmayacaktır.

### 2.4.5. Dilbilimsel Yaklaşım

Adından da anlaşılacağı gibi, sözlük tabanlı yaklaşım sözlük veya sözlükleri kullanır. Bu adımda, bir belgenin yönünü hesaplamak için kelimelerin veya cümlelerin anlamsal yönelimi veya kutupluluğu kullanılır. Bir makine öğrenimi yaklaşımından farklı olarak, sözlük tabanlı yöntem, büyük bir veri kitaplığına depolanmasını gerektirmez. Sözlük veya sözlükler kullanarak bir belgenin yönünü değerlendirir. Anlamsal Yönelim, metindeki öznellik ve görüşün ölçüsüdür ve kelimelerin veya ifadelerin kutupluluğunu ve gücünü yakalar. Tüm bu cümleler, belgenin tam duygu yönelimini yönetir. Düşünce,

duygu sözlüğü manuel veya otomatik olarak oluşturulur. Düşünce, duygu sözlüğü geliştirmeye yönelik manuel yaklaşım zaman alıcı olabilir. Diğer otomatik yöntemlerle birleştirilmelidir, bu ayrıntılar büyük ölçüde iki sözlük kategorisine ayrılır: ortak sözlük ve kategoriye özel sözlük. Sözlüğe dayalı yaklaşımı kullanmanın avantajı, duygu sözcüklerinin listesinin ve bunların kutupluluğunun çok hızlı bir şekilde aranabilmesidir. Standart veri sözlüğü, aynı duygu değerine sahip varsayılan duygu sözcüklerini, bölünmüş sözcükleri, müzakere sözcüklerini ve kapalı müzakere sözcüklerini içerir. Kapsamlı, yüksek kaliteli bir sözlük, genellikle büyük ölçeklerde hızlı, doğru duygu analizi için gereklidir (Gustafsson ve Davidsson, 2020) (Hutto ve Gilbert, 2014).

### 2.4.6. Valence Aware Dictionary and Sentiment Reasoner

Sözlük tabanlı sözbilimsel bir duygu analizi algoritmasıdır. Sosyal medyada ifade edilen duygular için özel olarak tasarlanmıştır ve diğer alanlardaki metinlerde de iyi çalışır.

VADER, sözlükteki her kelimenin bileşik puanlar alarak, pozitif, negatif veya nötr bir kategoriye giren metin oranları için oranlar olarak çıktı sağlar ve hepsi birlikte 1'e eşittir. Bileşik puan, duygu analizi için en yaygın kullanılan ölçümdür.

Bileşik puan, [-1, 1] aralığında kayan bir noktadır. Bileşik puan, sözlükteki her kelimenin değerlik puanlarının toplanmasıyla hesaplanır, kurallara göre ayarlanır ve daha sonra -1 ile +1 arasında olacak şekilde normalleştirilir. Bileşik puan >= 0,05 olumlu duyguyu belirtir ve bileşik puan <= -0.05 olumsuz duyguyu belirtir. Tarafsız duyarlılık, -0.05 < bileşik puan < 0.05 ile tanımlanır.

Bu sözlüğü kullanmak, zaman alıcı olsa bile manuel olarak kontrol ederek ortadan kaldırabilecek hatalar verir. VADER tabanlı yaklaşımın bir dezavantajı, etki alanından bağımsız olmasıdır. Örneğin, "sessiz" kelimesi bir arabayı olumlu bir şekilde tanımlar, ancak bir hoparlörü tanımlarken olumsuz olur.

Uygulama büyük ölçüde VADER uygulaması ile gerçekleştirilmektedir. Açık kaynaklı bir algoritma olan VADER, metnin hem kutup değerine (olumlu/olumsuz) hem de yoğunluğuna (şiddetine) duyarlı bir metin duygu analizi modelidir. VADER algoritması bize bir ifadenin olumlu ya da olumsuz ya da tarafsız mı olduğunu ve olumlu, olumsuz ya da tarafsız olmasının değerini verir. VADER, sistematik olarak oluşturulmuş bir duygu sözlüğü ile duygu analizi geliştirmek için bazı söz dizimsel kuralları içerir.

VADER, özellikle tweet verileri için oluşturulmuştur ve hem kısaltmalar hem de emojiler içerir.

Emojiler, internette yaygın olarak kullanılan duygusal simgelerdir. Tamamen ücretsiz bir açık kaynaklı araçtır. VADER, kelime sırası ve derece değiştiricileri de dikkate alır (Hutto ve Gilbert, 2014)(Elbagir ve Yang, 2019).

### 2.4.7. Bidirectional Representation for Transformers (BERT) Modeli

BERT Modeli, Google Research'teki araştırmacılar tarafından 2018 yılında önerilen bir Doğal Dil İşleme modelidir. Arama motorunun insan dilini anlamasını artıran bir yapay sinir ağı algoritmasıdır. Sinir ağları, bir hayvanın merkezi sinir sisteminden ilham alan, kalıpları öğrenebilen ve tanıyabilen bilgisayar modelleridir. Bunlar makine öğreniminin bir parçasıdır. BERT örneğinde, sinir ağı insan dilinin ifade biçimlerini öğrenme yeteneğine sahiptir. Transformer adlı bir Doğal Dil İşleme (NLP) modeline dayanmaktadır ve bu model, bir cümledeki kelimeler arasındaki ilişkileri sırayla tek tek görüntülemek yerine anlamaktadır.

Önerildiği dönemde genel dil anlama değerlendirme, Stanford soru cevap veri seti SquAD versiyon 1.1 ve versiyon 2.0 gibi birçok doğal dil işleme ver doğal dil anlama araştırmasında en son teknoloji doğruluklar sağlamaktadır. Yayınlandıktan birkaç gün sonra yayınlana kod, önceden eğitilmiş BERTbase ve büyük bir veri kümesinin üzerinde eğitilmiş BERTlarge sürümüyle açık kaynaklı hale getirilmiştir. BERT ayrıca yarı denetimli eğitim, OpenAI transformatörleri, ELMo Embeddings, ULMFit, Transformers gibi önceki birçok NLP algoritmasını ve mimarisini kullanır. Bu modelin asıl amacı Google Arama ile ilgili sorguların anlamının anlaşılmasını geliştirmektir. İsminden de anlaşılacağı gibi diğer modellerden farklı olarak cümleyi hem sağdan sola hem soldan sağa değerlendirmektedir. Böylece metinlerin anlamlarını ve birbirleriyle oluşturdukları ilişkileri yüksek bir güvenilirlik düzeyinde ortaya çıkarmayı hedeflemekte ve sonuçlarda da bunun karşılığını almaktadır. Örneğin, "bank" kelimesi, "banka hesabı" ve "nehir kıyısındaki bank" için bağlamdan bağımsız aynı temsile sahip olacaktır. Bunun yerine bağlamsal modeller, cümledeki diğer kelimelere dayanan her kelimenin bir temsilini oluşturur. Örneğin, "Banka hesabına eriştim" cümlesinde, tek yönlü bir bağlamsal model, "eriştim"e dayalı olarak "banka"yı temsil eder ancak "hesap" sözcüğünü temsil etmez. Bununla birlikte, BERT, derin bir yapay sinir ağının en altından başlayarak "banka ...

eriştim" cümlesinde hem önceki hem de sonraki bağlamını kullanarak “hesap” sözcüğünü temsil etmektedir (ai.googleblog.com, 2021).

Ön eğitim prosedürü, büyük ölçüde, dil modeli ön eğitimi ile ilgili mevcut literatürü takip etmektedir. Eğitim öncesi derlem için 800 milyonluk sözcük barındıran BooksCorpus ve iki buçuk milyar sözcük içeren Wikipedia kullanılmaktadır. BERT (large) ve BERT (base) şeklinde iki farklı temel algoritma sunulmaktadır. BERT_large için on altı adetlik Tensor Processing Unit (TPU) ve BERT_base için dört adetlik Tensor Processing Unit (TPU) kullanılarak dört gün süresince eğitilmiştir. Google, aynı zamanda base model olan algoritmayı OpenAI modeli ile karşılaştırma yapmak üzere aynı özelliklerde geliştirdiğini açıklamaktadır. BERT kendi başına GLM adı verilen, birden fazla problemde kullanılabilecek şekilde tasarlanmış bir modeldir. Chatbot, metin sınıflandırma vb. problemlerin çözümünde kullanmak için modelin üstüne ekstra katmanlar eklenmesi gerekmektedir. Wikipedia için listeleri, tabloları ve başlıkları yok saymakta ve yalnızca metin pasajlarını çıkarmaktadır. Uzun bitişik dizileri çıkarmak için Billion Word Benchmark gibi karıştırılmış cümle düzeyinde bir sözlük yerine belge düzeyinde bir sözlük kullanmak çok önemlidir. BERT, çift-yönlü olması dışında Maskelenmiş Dil Modelleme (MLM) ve Sıradaki Cümleyi Tahminleme (NSP) olarak adlandırılmış iki farklı yöntemle eğitilmektedir. BERT modelinin karşılaştığı ilk cümlede, cümlede yer alan kelimelerin yüzde on beşinde maskelenmiş dil modelleme yöntemi kullanılmaktadır. Bu yöntemin kullanıldığı kelimelerin yüzde sekseni MASK jetonu ile maskelenirken, yüzde onluk kısmı başka bir kelime ile yer değiştirmektedir, geriye kalan yüzde onluk kısım ise olduğu gibi bırakılmaktadır. Yüzde on beşlik değer seçilmesini sebebi olarak, çok fazla kelimenin maskelenmesinin eğitimi çok zorlaştırdığı, çok az kelimeyi maskelemenin de cümledeki içeriğin çok iyi kavranamamasına sebep olduğu belirtilmektedir. MLM yönteminde, yüzde seksenlik kısımda maskelenen kelimeler tahmin edilerek bulunmaya çalışılır, diğer yüzde yirmilik kısımla ilgili işlem yapılmamaktadır. Bu nedenle kayıp değeri yalnızca bu yüzde seksenlik maskelenen kısım üzerinden hesaplanmaktadır. MLM yönteminde, cümle içerisindeki kelimeler arasındaki bağıntılar tahminlenmeye çalışılırken, NSP yönteminde ise cümleler arasında bağıntı tahminlenmeye çalışılmaktadır. Eğitim esnasında ikili olarak gelen cümle çiftinde, ikinci cümlenin ilk cümlenin devamı olup olmadığı tahmin edilmektedir. Bu teknikten önce ikinci cümlelerin yüzde ellisi rassal olarak değiştirilmekte, diğer yüzde ellisi ise aynı

şekilde bırakılmaktadır. Eğitim esnasındaki optimizasyon, bu iki teknik kullanılırken ortaya çıkan kaybın minimuma indirilmesidir (Devlin, Chang, Lee ve Toutanova, 2019).

Google, her gün aldığı sorguların %15'nin yeni aramalar olduğunu belirtiyor. Bu nedenle, arama sorgusunu anlamak için Google arama motorunun dili çok daha iyi anlaması gerekmektedir. Modelin dil anlayışını geliştirmek için BERT, farklı bir mimaride farklı görevler için eğitilmiş ve test edilmiştir (www.geeksforgeeks.com, 2021). BERT, şu anda Google'da, arama sorgularının yorumlanmasını optimize etmek için kullanılmaktadır. BERT, aşağıdakiler dahil, bunu mümkün kılan çeşitli işlevlerde kullanılabilmektedir.

Sıradan sıraya dayalı dil oluşturma görevleri:

- Soru cevaplama
- Özet oluşturma
- Cümle tahmini
- Sohbet yanıtı oluşturma

Doğal dil anlama görevleri:

- Çokanlamlılık çözümleme (anlamları farklı olan kelimeler)
- Kelime anlamı belirsizliği
- Doğal dil çıkarımı
- Duygu sınıflandırması

### 2.4.8. Kernel Yoğunluk Tahminlemesi (KDE)

Kernel yoğunluk tahminlemesi (KDE) yöntemleri, olay kalıplarını anlama ve potansiyel olarak tahmin etme amacıyla, mekansal verilerin görselleştirilmesinde ve analiz edilmesinde sıklıkla kullanılmaktadır. Bu yöntemlerin risk değerlendirmesi ve hasar analizi, yangın ve kurtarma hizmetleri için acil durum planlaması, yol kazaları gibi çok çeşitli uygulamaları bulunmaktadır (Silverman, 1986) (Smith, Goodchild ve Longley, 2015) (Anderson, 2009).

KDE, tüm nokta deseni üzerine yerleştirilmiş bir ızgara üzerinde yapılan bir dizi tahminden dolayı özellikle sıcak noktaları tespit etmede kullanılışlı bir tahminleme olarak öne çıkmaktadır. Bu tahminlerin her biri belirli bir konumdaki yoğunluğu göstermekte ve bu nedenle nokta desen yoğunluklarının en yüksek ve en düşük değerlerini tespit etmektedir. Kullanıcının tek rolü, belirleyici bir rol oynayan tahmin için uygun bant

genişliğini belirlemektir. Bant genişliği çok büyük ayarlandığında önemli bilgiler kaybolabilmektedir. Küçük bir bant genişliği durumunda yerel veri bilgisinin sonuç üzerinde daha önemli bir etkisi bulunmaktadır. Bu işlemi kolaylaştırmak için önceden işlenmiş KDE haritalarını belirtilen bant genişliği ile aynı anda çıkaran bir bant genişliği kaydırma aracı önermektedir. Bu şekilde, kernel bant genişliğinin KDE'ye etkisi açıkça gösterilebilmekte ve uygun bir bant genişliği görsel olarak belirlenebilmektedir (Krisp, Peters, Murphy ve Fan, 2009)(Krisp ve Spatenkova, 2009).

Olasılık yoğunluk fonksiyonu:

$$f(x) = \frac{1}{nh} \Sigma_{i=1}^{n} K\,(x-xi)/h,$$

n'nin örnek boyutunu gösterdiği, h kernel bant genişliğini temsil eder ve K, kernel işlevi anlamına gelir. Bant genişliği h, yaklaşık yoğunluğun düzgünlüğünü belirler. Daha küçük bir h bant genişliği daha değişken bir tahmini yoğunluğa yol açarken, daha büyük bir h daha kabul edilebilir bir tahmini yoğunluğa yol açar. Rastgele değişken yoğunluk, kernel fonksiyonları kullanılarak hesaplanır. Burada kullanılan kernel işlevi, olarak tanımlanan Gauss kernel işlevidir.

*Gaussian kernel fonksiyonu*

$$K(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Fonksiyon artan bir fonksiyon oldugunda, normal kümülatif dağılım işlevi (CDF), normal dağılım işlevinin belirtilen rastgele değişkene eşit veya ondan küçük yüzdesini döndüren bir tekniktir. Φ islevi, standart normal dağılımın kümülatif dağılım fonksiyonunu belirtir:

*CDF standart normal dağılımı:*

$$\Phi(x) = P(Z \leq x) = -\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} exp\left\{-u^{2}/_{2}\right\} dx$$

## 2.5. Zaman Serileri Analizi

Zaman serileri analizi, muayyen bir zaman dilimindeki verileri veya verilerin eğilimlerini analiz eden istatistiksel yöntemdir. Bir zaman dizisi, birbirini takip eden

muayyen bir zaman süresinde belirlenmiş sıralı veri noktalarını içermekte, zaman serisindeki veri noktalarının altındaki temel sebepleri anlayabilmek, önermek veya tahminlerde bulunmak üzere bir zaman serisini tahmin etmeye çalışan yöntemleri içermektedir. Zaman serisi analizini kullanan tahmin verileri, bilinen geçmiş sonuçlara dayanarak gelecekteki sonuçları tahmin etmek için bazı önemli modellerin kullanımını içermektedir. Bu analizin bir amacı, trendler, döngüler ve düzensiz hareketler dahil olmak üzere bir zaman serisinin bileşenlerini ifade ettiği zaman içindeki değişikliklerdeki kalıpları keşfetmek ve anlamaktır.

Örneğin, müşterilerin geçmiş verilerine bağlı olarak belirli bir zaman diliminde kaç müşterinin geleceği gibi daha fazla müşterinin de ne zaman geleceğinin tahminlenmesi için zaman serisi analizi kullanılmaktadır. Tahmin edilen değerler, hangi verinin uzun ya da kısa kaldığını, bağlantılar ve ilişkilendirme, günlük, mevsimsel eğilimler ve davranışları incelemek için zaman serileri analizleri kullanılmaktadır.

### 2.5.1. Zaman Serileri İçin Otokorelasyon Analizi

Otokorelasyon bir tür seri bağımlılıktır. Otokorelasyon, bir zaman serisi kendisinin gecikmiş bir versiyonuyla lineer olarak ilişkili olduğunda meydana gelir. Korelasyon ise basitçe iki bağımsız değişkenin doğrusal olarak ilişkili olduğu zamandır.

Bazı zaman serisi tahmin yöntemleri örneğin, regresyon modellemesi gibi, gözlem ve hesap sonuçları arasındaki farkların otokorelasyondan (ayarlanmış model ile veriler arasındaki fark) bağımsız olduğu varsayımına dayanır. Otokorelasyon analizinin belki de en zorlayıcı yönü, verilerimizdeki gizli detayları ortaya çıkarmada ve uygun tahmin yöntemlerini seçmemizde bize nasıl yardımcı olabileceğinin tespit edilmesidir. Bunu, zaman serisi verilerindeki dönemselliği ve eğilimi belirlemek için özel olarak kullanabiliriz (Shumway ve Stoffer, 2005)(Hyndman ve Athanasopoulos, 2018).

### 2.5.2. Zaman Serileri İçin Dönemsel Ayrışma

Zaman serisi, seviye, eğilim, mevsimsellik ve gürültü olarak dört bileşenden oluşmaktadır. Seviye, serinin temel ortalama değerini ifade etmektedir. Eğilim, serinin zaman içinde isteğe bağlı ve genelde doğrusal artan ya da azalan değeridir. Mevsimsellik, zaman içinde isteğe bağlı yinelenen davranış benzerlikleri ve döngüleridir. Gürültü ise verilerde, model tarafından açıklanamayan isteğe bağlı rassal değişimdir.

Tüm zaman serilerinin bir seviyesi bulunmaktadır ve çoğunlukla gürültü içermektedir, trend ve mevsimsellik ise isteğe bağlıdır. Bazı kaynaklarda gürültü modellenemeyen bir bileşen olması sebebiyle sistematik olmayan bir bileşen olarak ayrılırken, seviye, eğilim, mevsimsellik ise tutarlılığı veya tekrarı olan, tanımlanabilen ve modellenebilen zaman serileri bileşenleri, yani sistematik zaman serileri bileşenleri olarak tanımlanmaktadır (Shumway ve Stoffer, 2005)(Hyndman ve Athanasopoulos, 2018).

# ÜÇÜNCÜ BÖLÜM

# MATERYAL VE YÖNTEM

## 3.1. Kullanılan Araçlar

Bu çalışmada kullanılan programlama diline, yazılım platformuna, Python'da kurulan bilimsel kütüphanelere, kullanılan bilgisayar özelliklerine, seçilen algoritmalara bu bölümde yer verilmiştir.

### 3.1.1. Python Programlama Dili

Bu çalışmanın makine öğrenmesi modellerinin geliştirilmesi için Python yazılım dili ve kütüphaneleri kullanılmıştır.

Python, dilsel verileri işlemek için mükemmel işlevselliğe sahip basit ama güçlü bir programlama dilidir. Python, http://www.python.org/ adresinden ücretsiz olarak indirilebilir. Tüm platformlar için yükleme seçenekleri mevcuttur.

Aşağıda file.txt dosyasını işleyen ve "ing" ile biten tüm kelimeleri yazdıran beş satırlı bir Python kod bloğu bulunmaktadır.

```
for line in open("file.txt"):
        for word in line.split():
        if word.endswith('ing'):
print word
```

Bu kod bloğu Python'ın bazı temel özelliklerini göstermektedir. İlk olarak, boşluk kod satırlarını iç içe geçirmek için kullanılır; böylece if ile başlayan satır, for ile başlayan önceki satırın kapsamına girer. Bu kod her kelime için ing testinin yapılmasını sağlar. İkinci husus ise Python nesne yönelimlidir, her değişken, belirli tanımlanmış niteliklere ve yöntemlere sahip bir varlıktır. Örneğin, değişken satırının değeri bir karakter dizisinden daha fazladır. Bir satırı sözcüklerine bölmek için kullanabileceğimiz split() adlı bir yöntemi olan bir dize nesnesidir. Bir nesneye yöntem uygulamak için nesne adını, ardından bir nokta ve ardından yöntem adını, yani line.split() yazarız. Üçüncü husus, yöntemlerin parantez içinde ifade edilen argümanları vardır. Örneğin, word.endswith('ing'), başka bir şey değil, ing ile biten kelimeler istediğimizi belirtmek için 'ing' argümanına sahiptir. Son olarak ve en önemlisi, Python son derece okunabilirdir, daha önce hiç kodlama yapmamış olsanız bile bu programın ne yaptığını tahmin etmek oldukça kolaydır.

Yorumlanmış bir dil olarak Python, etkileşimli keşfi kolaylaştırır. Nesne yönelimli bir dil olarak Python, verilerin ve yöntemlerin kapsüllenmesine ve kolayca

yeniden kullanılmasına izin vermektedir. Dinamik bir dil olarak Python, nesnelere anında özniteliklerin eklenmesine izin verir ve değişkenlerin dinamik olarak yazılmasına izin vererek hızlı geliştirmeyi kolaylaştırır. Python, grafik programlama, sayısal işleme ve web bağlantısı için bileşenleri içeren kapsamlı bir standart kitaplıkla birlikte gelir.

Python, dünya çapında endüstride, bilimsel araştırmalarda ve eğitimde yoğun olarak kullanılmaktadır. Python, yazılımın üretkenliğini, kalitesini ve sürdürülebilirliğini kolaylaştırdığı için genellikle övülür. Python başarı öykülerinin bir koleksiyonu http://www.python.org/about/success/ adresinde yayınlanmaktadır. NLTK, Python'da NLP programları oluşturmak için kullanılabilecek bir altyapı tanımlar. Doğal dil işlemeyle ilgili verileri temsil etmek için temel sınıflar sağlar; konuşma bölümü etiketleme, sözdizimsel ayrıştırma ve metin sınıflandırma gibi görevleri gerçekleştirmek için standart arayüzler; ve karmaşık sorunları çözmek için birleştirilebilen her görev için standart uygulamalar.

NLTK kapsamlı belgelerle birlikte gelmektedir. Bu kitaba ek olarak, http://www.nltk.org/ adresindeki web sitesi, araç setindeki her modülü, sınıfı ve işlevi kapsayan, parametreleri belirten ve kullanım örnekleri veren API belgeleri sağlamaktadır. Web sitesi ayrıca kullanıcılar, geliştiriciler ve eğitmenler için tasarlanmış kapsamlı örnekler ve test senaryoları içeren birçok “Nasıl” belgesi sunmaktadır.

Bu çalışmada makine öğrenimi modellerini geliştirmek için aşağıdaki Python kitaplıkları kullanılmıştır.

### 3.1.2. NLTK Kütüphanesi

- İnsan dili verileriyle çalışan ve WordNet ve metin işleme kitaplıkları gibi çeşitli sözlük kaynaklarına basit bir arayüz sağlayan bir Python paketidir.
- Sınıflandırma, jetonlaştırma, kaynak oluşturma, etiketleme, ayrıştırma ve anlamsal akıl yürütme bu sözlüksel kaynaklar kullanılarak gerçekleştirilir.

### 3.1.3. Google Colab

- Kısa ismiyle Colab olarak bilinen Google Colaboratory, internet tarayıcı üzerinde Python dilinde programlar yazmayı ve çalıştırmayı sağlamaktadır.
- Herhangi bir yapılandırma yapılmasına gerek duymamaktadır.
- GPU'lara ücretsiz erişim imkanı sunmaktadır.
- Kolay paylaşım imkanı sunmaktadır.

- Colab not defterleri; yürütülebilir kod, zengin metin, resimler, HTML, LaTeX ve diğer öğeleri tek bir dokümanda birleştirmeye imkan sağlamaktadır.
- Oluşturulan Colab not defterleri Google Drive hesabında saklanabilmektedir.
- Colab not defterleri diğer insanlarla kolayca paylaşılabilir, paylaşılan kişilerin not defterleri üzerinde yorum yapmaları, hatta düzenlemeleri sağlanabilir (colab.research.google.com, 2021).

### 3.1.4. Scikit-Learn Kütüphanesi

- Scikit-Learn (skLearn), Python yazılım dilinde makine öğrenmesi için geliştirilmiş yayın kütüphanelerden birisidir.
- Herkes tarafından erişime açık ve ulaşılabilirdir.
- Sınıflandırma, regresyon, kümeleme, boyut azaltma, model seçimi, yeniden işleme problemlerinde kullanabilir.
- Tahmine dayalı veri analizi çalışmaları için basit ve verimli araçlar sunmaktadır.
- NumPy, SciPy ve matplotlib üzerinde kuruludur.
- Açık kaynak ve ticari olarak kullanıma uygundur (scikit-learn.org, 2021).

### 3.1.5. Numpy (Numerical Python) Kütüphanesi

- NumPy, temel Python bilgi işlem paketidir. Kapsamlı bir üst düzey matematiksel işlevler koleksiyonu sağlayarak çok boyutlu dizilerin ve matrislerin işlevselliğini genişletmek için kullanılır.
- Pandas ve matplotlib dahil olmak üzere bir çok Python kütüphanesiyle uyumlu çalışmaktadır.
- Diziler üzerinde matematiksel, mantıksal, şekil işleme, sıralama, seçme gibi işlemler için temel matematik ve istatistik işlemlerini yapabilmektedir.
- Bellekte yeni bir dizi oluştururken, orjinali sildiğinden sabit ve küçük bir boyutta kalmaktadır.
- Standart matematiksel yapıdaki kodlamalar için daha uygundur.
- Açık kaynak olarak sunulmaktadır (numpy.org, 2021) .

### 3.1.6. Pandas Kütüphanesi

- Veri analiz aracı olarak işlev gören ve veri yapılarıyla çalışan bir Python paketi. Pandalar, tüm veri analizi iş akışını python'da gerçekleştirerek R gibi daha alana özgü bir dile geçme ihtiyacını ortadan kaldırır.

- Otomatik ve açık veri hizalama yapabilmektedir.
- Hem verileri toplamak hem de dönüştürmek için veri kümeleri üzerinde işlevselliğe göre güçlü ve esnek gruplama yapabilmektedir.
- Diğer Python ve NumPy veri yapılarındaki düzensiz dizine sahip verileri DataFrame nesnelerine dönüştürebilmektedir.
- Büyük veri dosyalarının alt kümelenmelerinin yapılabilmesine olanak sağlamaktadır.
- Veri dosyalarının özet dosya olarak yeniden şekillendirilmesine esneklik sağlamaktadır.
- Zaman dilimi, tarih aralığı oluşturmak, periyot değiştirme, tarih değiştirme ve geciktirme, kayan pencere istatistikleri oluşturulabilmektedir.
- Düz metin dosyalarından, excel dosyalarından, veri tabanlaırndan veri yükleme işlemini çok hızlı yapabilmektedir (pandas.pydata.org, 2021).

### 3.1.7. Matplotlib Kütüphanesi

- Matplotlib, iki boyutlu çizimlerin Python'da yapılabilmesi için geliştirilmiş popüler bir görselleştirme kütüphanesidir.
- John Hunter tarafından 2002 yılında tanıtılmıştır.
- MathWorks tarafından geliştirilen tescilli bir programlama dili olan Matlab'a benzer olmak üzere tasarlanmıştır.
- Açık kaynak ve herkes tarafından erişilebilir bir kütüphanedir.
- Dizilerdeki verilerden iki boyutlu çizimler yapmak için platformlar arası kullanıma sahiptir.
- Python'da yazılmıştır ve sayısal mekanik uzantısı olan NumPy'ı kullanmaktadır.
- Python ve IPython terminallerinde, Jupyter notebook ve web uygulama sunucularında kullanılabilmektedir.
- Matplotlib, dağılım, çizgi, pasta, histogram, güç spektrumları, çubuk grafikler vb. üretebilmektedir (matplotlib.org, 2021).

### 3.1.8. Vader

- Açık kaynaklı bir algoritma olan VADER, sözcüklerin olumlu veya olumsuz olarak kutup değerine ve bu değerin şiddetine göre metinlerde duygu analizi yapan bir modeldir.

- VADER algoritması bize bir ifadenin olumlu ya da olumsuz ya da tarafsız mı olduğunu ve olumlu, olumsuz ya da tarafsız olmasının değerini verir.
- VADER, sistematik olarak oluşturulmuş bir duygu sözlüğü ile duygu analizi geliştirmek için bazı söz dizimsel kuralları içerir.
- Sözlük tabanlı bir model olduğundan cümledeki anlam bütünlüğüne göre yorumlayamaması söz konusudur.
- VADER, özellikle tweet verileri için oluşturulmuştur ve hem kısaltmalar hem de emojiler içerir. Emojiler, internette yaygın olarak kullanılan duygusal simgelerdir. Tamamen ücretsiz bir açık kaynaklı araçtır.
- VADER, kelime sırası ve derece değiştiricileri de dikkate almaktadır.

### 3.1.9. Bert

- Açık kaynak bir model olan BERT, transformer mekanizmasıyla çalışan, çift yönlü maskelenmiş dil modelini kullanan, bir veri seti ile eğitilerek dili anlaması sağlanan yapay sinir ağı modelidir.
- Transformer mekanizması, çoklu işlem yapma algoritması ile kelimenin önünden ve arkasından gelen kelimeyle olan ilişkisine odaklanarak çift yönlü değerlendirme yapılmasını sağlamaktadır.
- Soldan sağa, sağdan sola cümledeki kelimelerin yerlerini değiştirerek, maskeleyerek ve sonradan gelen cümleler arasındaki ilişkileri hesaplayarak anlam bütünlüğüne odaklanan bir dil modelidir.
- Hazır eğitilmiş modeller ince ayar tekniğiyle yeni problem çözümlerinde kullanılabilmektedir. Soru cevaplama, metin sınıflandırma ve gruplandırma, görüntü işleme, ses tanıma gibi alanlarda çözüm üretebilmektedir.
- Cümle içerisindeki kelimeleri sırayla değerlendirdiği için uzun cümle yapılarında kelimeler arasındaki ilişkilerde kayıplar söz konusudur.
- Modelin birden çok algoritmayı içerisinde barındırıyor olması ve bu işlemlerin hem iyi bir donanım gerektirmesi hem de zaman alması bakımından yüksek maliyetli bir modeldir.

### 3.1.10. Transformers Kütüphanesi

- Hugging Face Transformers (dönüştürücü) paketi, çeşitli doğal dil işleme (NLP) görevleri için olağanüstü yararlı olan önceden eğitilmiş modeller sağlayan son derece popüler bir Python kitaplığıdır.

- Önceden yalnızca PyTorch'u destekliyordu, ancak 2019'un sonlarından itibaren TensorFlow2 de destekleniyor.
- Kütüphane, Doğal Dil Çıkarımından (NLI) soru sorup cevaplamaya kadar birçok görev için kullanılabilirken, metin sınıflandırma en popüler ve pratik kullanım örneklerinden birisidir.

### 3.1.11. Tensorflow Kütüphanesi

- Açık kaynak kod olarak Google tarafından geliştirilmiş, derin öğrenme destekli yapay zeka çalışmalarında kullanılmaktadır.
- Ses tanıma, sesli komut, sesli arama, duygu analizi, otomotiv ve havacılık sanayilerinde akım kaçağı tespit etme teknolojilerinde kullanılmaktadır.
- Apple Siri, Google Now, MS Cortana gibi sesli komut teknolojilerinde kullanılmaktadır.
- Dil tanıma, metin özetleme gibi metin tabanlı uygulamalarda kullanılmaktadır.
- Google Translate tarafından dil tanıma teknolojisinde kullanılmaktadır.
- Fotoğraf ve/veya video görüntü tanıma, işlemede kullanılmaktadır.
- İki boyutlu görsellerden üç boyutlu uzay modelleri oluşturmada, sosyal ağlarda fotoğraf etiketleme önerilerinde, deep face, deep fake uygulamalarında kullanılmaktadır.
- Zaman serileri analizinde kullanılmaktadır.
- Python yanı sıra Java, JavaScript, C#, C++ yazılım dilleriyle birlikte kullanılabilmektedir.
- Sistem üzerindeki merkezi ya da grafik işlemcileri kullanabilmektedir.
- NumPy ile birlikte kullanılmalıdır (tensorflow.com, 2021).

### 3.1.12. Torch Kütüphanesi

- PyTorch, çok boyutlu vektörel bir makine öğrenme kütüphanesidir.
- Öncelikle GPU ve CPU'ları kullanan derin öğrenme uygulamaları için iyileştirilmiş bir bilimsel hesaplama, tensör kütüphanesidir.
- Torch, genel olarak yapay zeka algoritmalarında öğrenim için kapsamlı bir çözüm üreten hesaplama yapısıdır.
- Facebook tarafından 2017 yılında sunulmuştur.
- Açık kaynak kodludur ve erişime açıktır.

- Temelindeki C/CUDA uygulaması ve LuaJIT yazılım dili nedeniyle kullanımı hem basit hem de verimli olmaktadır.
- Torch, algoritmaların hazırlığını yüksek bir esneklik ve hızda yapmayı amaçlarken, yapılan işlemleri de basite indirgemektedir.
- Torch, alınan sinyallerin dönüştürülmesi, sensörlerden aktarılan ses, video ve görüntülerin işlenmesi gibi büyük işlem yükü olan çeşitli alanlarda kullanılmaktadır.
- Grafik işlemciler ve merkezi işlemciler arasında etkili bir paralel işlem yapabilmekte ve yapay sinir ağına ait isteğe bağlı grafiklerini oluşturulabilmektedir (torch.ch, 2021)

# DÖRDÜNCÜ BÖLÜM

# VERİ ANALİZİ AŞAMALARI

## 4.1. Veri Analizi Aşamaları

### 4.1.1. Veri Toplama

Çalışmada kullanılan veri setlerini toplamak için sosyal medya platformu olarak Twitter seçilmiştir. Temel adım olarak Python ve Twitter Mikroblog arasında bağlantı kurularak veriler indirilebilir. Twitter, verilerini URL'ler aracılığıyla erişilebilen ve genel API'ler aracılığıyla kullanıma sunmaktadır. Ancak API hesabı açtırabilmek için özel olarak başvuru yaparak Twitter'dan onay alınması gerekmektedir ve sonrasında bu API'nin kullanılabilmesi söz konusudur. Python, Twitter'ın verilerine API aracılığıyla erişmenizi sağlayan çok basit bir paket içerir. API hesabınızın kullanıcı bilgilerini kod bloğuna yazarak Tweepy gibi gerekli kitaplıkları yardımıyla verileri indirebilirsiniz.

Günümüzde Twitter kullanıcıları metinle ifadelerinin yanı sıra, duygularını ifade etmek için gülme, ağlama, utanma, sinirlenme, üzülme gibi duygularını ifade etmek için emojiler kullanmaktadır. Twitter'dan indirilen veriler belirli bir zaman aralığı boyunca gerçekleştirilir ve ilgili dönemin verileri farklı CSV dosyalarında kaydedilir. Hedeflenen bilgiler bu dosyalarda yer alan içerik ve tweet'lerin zaman damgalarıdır. Ana işlem, tweet'lerin Twitter'dan alınması ve tweet'lerin python kütüphanesini kullanarak duygu analizi yapan bir algoritma ile değerlendirilmesidir. Twitter verilerini çıkarmak için kullanılan yöntemler, herkese açık ham tweet'lere gerçek zamanlı erişime izin vermektedir. Twitter API, kullanıcıların resmi olarak bir kullanıcı hesabından tweet indirmelerini ve tweetleri .csv gibi uygun bir dosya formatında kaydetmelerini sağlamaktadır. Ancak bu çalışmada ise verisetlerini düzenli aralıklarla güncelleyerek açık kaynak olarak kamuya sunan Gabriel Preda'nın verisetinden yararlanılmıştır. (). Twitter'ın genel mesaj panosunda yayınlanan COVID-19 aşısı ile ilgili toplam 212.983 tweet bulunan bır .csv veri seti kullanılmıştır (www.kaggle.com, 2021).

Gabriel Preda'nın veri setlerini çekmek için kullandığı kod bloğu ise aşağıda paylaşılmıştır (www.github.com, 2021)

```
tweets_pfizer = create_cursor(api, "#PfizerBioNTech -filter:retweets", "2021-02-02")
tweets_sinopharm = create_cursor(api, "#Sinopharm -filter:retweets", "2021-02-02")
tweets_sinovac = create_cursor(api, "#Sinovac -filter:retweets", "2021-02-02")
tweets_moderna = create_cursor(api, "#Moderna -filter:retweets", "2021-02-02")
tweets_oaz = create_cursor(api, "#oxfordastrazeneca -filter:retweets", "2021-02-02")
```

*tweets_covaxin = create_cursor(api, "#Covaxin -filter:retweets", "2021-02-02")*

*tweets_sputnikv = create_cursor(api, "#SputnikV -filter:retweets", "2021-02-02")*

*tweets_jandj = create_cursor(api, "#j -filter:retweets", "2021-02-02")*

### 4.1.2. Veriye Genel Bakış

Kullanılan veri seti, çeşitli bilgi sütunlarından oluşmaktadır. Toplamda 16 alan ve 212.983 tweet bulunmaktadır.

Bu sütun başlıkları aşağıdaki gibidir;

*id, user_name user_location, user_description, user_created, user_followers, user_friends, user_favourite, user_verified, date, text, hashtags, source, retweets, favorites, is_retweet*

Temel veri alanları, *kullanıcı_kimliği, kullanıcı_adı, tarih, metin, etiketler* olarak belirtilmiştir ve bu bilgiler duygu analizi için büyük ölçüde gerekli olan bilgilerdir.

### 4.1.3. Verinin Ön İşlenmesi

Twitter'da tweet, bir mikroblog mesajıdır. Twitter'da yıllarca 140 karakter olan mesaj uzunluğu limiti son yıllarda 280 karakter limitine artırılmıştır. Tweetlerin çoğu, gömülü bağlantı linklerinin, fotoğrafların, kullanıcı adlarının ve ifadelerin yanı sıra metin içerir. Tweetlerde yazım ve imla hataları bulunabilmektedir. Bu çalışmada sunulan modelde, yeni bir koronavirüs (2019-nCoV) ile ilgili yapılandırılmamış veriler Twitter verileri kullanılarak, ardından tarama veya filtreleme olarak adlandırılan metin temizleme ve sınıflandırma işlemlerine tabi tutulmaktadır. Bu nedenle, tweet'lerden alakasız bilgileri kaldırmak için bir dizi ön işleme adımı gerçekleştirilmektedir. Metnin analizi için HTML karakterlerinin, argo kelimelerin, duraklamaların, noktalamaların, bağlantı linklerinin kaldırılması gerekmektedir. Birbirine ulanmış kelimelerin ayrılması da temizlik için önem arz etmektedir (Jianqiang ve Xiaolin, 2017). Veriler ne kadar temiz olursa, madencilik ve özellik çıkarma için o kadar uygun olmaları ve bu da sonuçların daha doğru olmasını sağlamaktadır. Tweet'ler ayrıca, veri setinden yinelenen tweet'leri ve retweetleri ortadan kaldırmak için ön işleme tabi tutulmaktadır. Bu verileri önceden işlemek için Python Natural Language Toolkit'i (NLTK) kullanılmaktadır. İlk olarak, URL'ler ("http://url"), retweet (RT), kullanıcıdan bahsetme (@) ve istenmeyen noktalama işaretleri gibi tweet'lerin özel karakterlerini algılamak ve atmak için Python'da normal bir ifade (Regex) çalıştırılmaktadır. Hashtag'ler (#) genellikle tweet'in konusunu açıkladığı ve tweet'in konusu ile ilgili faydalı bilgiler içerdiği için tweet'in bir parçası olarak

eklenmiştir, ancak "#" sembolü kaldırılmıştır (Jianqiang ve Xiaolin, 2017)(Symeonidis, Efforsynidis ve Arampatzis, 2018).

```
cleaned_tweets = []

for tweet in tweets:

# String search - remove searched substring from string

# RE for links: r'http\S+

# RE for @mentions: @[A-Za-z0-9]

cleaned_tweet = re.sub(r"http\S+|@[A-Za-z0-9]+", "", tweet[0])

# Store in a new list of lists with cleaned tweets

cleaned_tweets.append([cleaned_tweet, tweet[1]])
```

Daha sonra tweet'leri küçük harfe dönüştürülerek, durdurma sözcükleri yani, hiçbir anlamı olmayan is, a, the, he, onlar vb. sözcükler kaldırılmaktadır, tweet'ler tek tek sözcüklere veya simgelere dönüştürülerek porter kök ayrıştırıcısı kullanılarak tweet'ler köklendirilmektedir. Ön işleme adımları tamamlandığında veri seti duyarlılık sınıflandırması için hazır olmaktadır (Wagh, Shinde ve Kale, 2018).

### 4.2. VADER Modeli İle Veri Analizi

Tweetlerde ifade edilen duygular, veri setine uygulanan VADER Sentiment Analyzer ile sınıflandırılmıştır. İlk olarak, veri setimizi kategorize etmek için bir duyarlılık yoğunluğu analizcisi (SIA) oluşturulmuştur. Daha sonra duyguları belirlemek için polarite puanları yöntemi kullanılmıştır. Ardından önceden işlenmiş tweetleri pozitif, negatif, nötr veya bileşik olarak sınıflandırmak için VADER Sentiment Analyzer'ı kullanılmıştır. Bileşik değer, belirli bir tweet metnindeki duyguyu ölçmek için önemli bir ölçümdür. Bileşik puan, sözlükteki her kelimenin değerlik puanlarının toplanmasıyla hesaplanıp, kurallara göre ayarlanmaktadır ve daha sonra -1 ile +1 arasında olacak şekilde normalleştirilmektedir. Eşik değerleri tweetleri pozitif, negatif veya nötr olarak sınıflandırmaktadır (Hutto ve Gilbert, 2014).

Duyguların sınıflandırılması:

**Olumlu görüş:** bileşik değer > 0,000001, değer = 1

**Nötr görüş:** (bileşik değer > -0.000001) ve (bileşik değer < 0.000001), değer =0

**Olumsuz görüş:** bileşik değer < -0.0000001, değer = -1

### 4.2.1. Analiz Edilen Veriye KDE Dağılımını Uygulanması

Eşik değerinden daha belirgin bir bileşik değere sahip bir tweet, pozitif bir tweet olarak sınıflandırılır. Buna karşılık, eşik değerinden daha düşük bir bileşik değere sahip bir tweet, mevcut çalışma boyunca olumsuz bir tweet olarak sınıflandırılmaktadır. Tweet metni, kalan durumlarda tarafsız olarak kabul edilmektedir. Toplam duygular duygusal değerlerine göre üç kategoriye ayrılmaktadır: olumlu, tarafsız ve olumsuz. Model girişinin uzunluğu, model büyümesi için çok önemli olan duyarlılık değeri tarafından belirlenmektedir. Bunu takiben, tüm duyguların özet bir dağılımı sunulmaktadır. Dağılım gösterilmeden önce kernel yoğunluğu hesaplamaları uygulanacaktır.

Matplotlib üzerine kurulu bir Python veri görselleştirme kütüphanesi olan Seaborn'u kullanmak, KDE grafiğini uygularken KDE grafiklerini çizmek için üst düzey bir arayüz sağlamaktadır. Daha sonra, verilerdeki pozitif, nötr ve negatif duyguların gücündeki önemli değişimleri gözlemlemek için Kümülatif Dağılım Fonksiyonu (CDF) kullanılarak duygu değerlerine dayalı olarak ölçülmektedir. Belirtilen rastgele değişkenden küçük veya küçük eşit normal dağılım işlevinin yüzdesini göstermektedir. Bu nedenle standart normal dağılımın CDF'sine göre, duyarlılık değerlerine ve yoğunluklarına göre pozitif, nötr ve negatif olarak genel duygular yorumlanmaktadır (Krisp, Peters, Murphy ve Fan, 2009) (Krisp ve Spatenkova, 2009).

### 4.2.2. Kelime Bulutu ile Duygu Analizi

En sık rastlanan kelimeler, bu analizde yukarıda bahsedilen duygulardan bulunur ve tweet'lerle ilgili hem olumlu hem de olumsuz duygular vermektedir. Sözcük bulutunda temsil edilen yorumlar, metinlerde en sık alıntılanan sözcükleri vurgulamaya yardımcı olan bir cümle olasılığı kümesiyle temsil edilmektedir. Kelime bulutu, en üstteki olumlu ve olumsuz duyguların her biri için “Word Cloud” paketleri kullanılarak oluşturulmaktadır.

### 4.2.3. Zaman Serileri Analizi Günlük Duygu Dağılımı

Günlük Twitter hacmine ilişkin bir zaman serisi genel kabulü kullanılarak, örnek zaman çizelgesi daha küçük zaman aralıklarına bölünür. Zaman serisi analizi, belirli bir dönemde konuya ait temel süreci açıklamak için Twitter mesajlarındaki zirveleri göstermektedir. Sürekli verileri geri bildirim olarak kullanan bu tür bir çalışma, bir konu hakkındaki durumsal bilgilerde zaman içinde meydana gelen değişiklikleri tespit edebilmemize yardımcı olmaktadır. Bu zaman serisi analizi, olayları gerçek zamanlı olarak tanımlar ve ekonomi, çevre, bilim ve tıp gibi çeşitli uygulamalarda kullanılmaktadır. Değişikliklerin nerede olduğunu analiz edilerek duyguların otokorelasyonu ve dönemsel ayrışması dahil olmak üzere birden fazla yöntem kullanarak meydana gelme zamanları tanımlanmaktadır. Ayrı zaman serilerine ayrılmak için, hem göreceli hacimdeki hem de olaylardaki ani değişimler gösterilmektedir.

İlk olarak, günlük duyguların her bölüm için zaman çizelgesi üzerinde gösterilmesi için üç bölüme halinde olumlu duygular ve olumsuz duygular için ortalama ve standart sapma hesaplanmaktadır. Bu tweetleri böldükten sonra, standart sapma ile olumlu ve olumsuz duyguların ortalamasını göstermek için bir model oluşturulmaktadır.

### 4.2.4. Otokorelasyon Analizi ile Duyguların Bileşenlere Ayrılması

Yapılandırılmış (yerleşik) modeldeki gecikmeleri ortadan kaldırmak için otokorelasyon analizi yapılmaktadır. Pearson korelasyon katsayısının değerini döndüren “Pandas.Series.autocorr()” fonksiyonu kullanılmaktadır. Pearson korelasyon katsayısı, iki değişkenin doğrusal korelasyonunu ifade etmektedir. Pearson korelasyon katsayısı -1 ile 1 arasında değişir; 0, doğrusal bir korelasyon olmadığını, >0 pozitif bir korelasyonu ve <0 negatif bir korelasyon olduğunu belirtmektedir. Pozitif korelasyon katsayısı, Pozitif korelasyon katsayısı, bir döngüde iki değişkenin değiştiğini, negatif korelasyon katsayısı ise değişkenlerin ters yönde değiştiğini göstermektedir. Verileri (t-2) karşılaştırmak için bir gecikme=1 (veya data(t)/data(t-1)) ve bir lag=2 (veya data(t)/data(t-2)) kullanılmaktadır. Daha sonra, çeşitli gecikme boyutlarına karşı otokorelasyon fonksiyonu (AFC) değerlerini ölçmek için otokorelasyon grafiği kullanılmaktadır. Gecikme değerinde bir artış olduğu için daha az ve daha az gözlem karşılaştırılmaktadır. Genel bir kural, toplam gözlem sayısının (T) en az 50 olması gerektiğidir en büyük gecikme değerinin (k), T/k'den küçük veya ona eşit olması gerekmektedir (Siebet, Gross ve Schroth, 2021)(www.towardsdatascience.com, 2021).

### 4.2.5. Günlük Duygu Eğitiminin Belirli Bir Olayla İlişkilendirilmesi

Çalışmanın bu aşamasında dönemsel ayrıştırma ve otokorelasyon analizi yapılarak, verilerden günlük duygular tahmin edilmiştir. Veri setimizi "kullanıcı adları", "tarih", "metin" ve "hashtag" alanlarına bölünerek ve "sayım" olarak ek bir alan eklenerek yeni veriler oluşturulmuştur. Verilerde yer alan tweet'lerin dönemsel analizini görmek için tarih alanı baz alınarak birleştirilmektedir.

### 4.2.6. BERT Modeli İle Veri Analizi

İlk olarak Hugging Face tarafından kullanıma sunulan Trasformers kütüphanesini kurulmuştur (https://huggingface.co/). Pandas, tensorflow, torch, numpy gibi sıralı kütüphaneler kurulmuştur. Öncelikle torch kütüphanesi yardımıyla GPU kullanımını modelimizde kullanmak üzere aktif hale getiriyoruz, bu işlem sadece işlemci tabanlı değil, grafik donanım desteğini de modele dahil ederek, cihazımızın maksimum düzeyde çözüm üretmesini sağlayacaktır (www.huggingface.co, 2021).

BERT belirteçleştirici (BERTtokenizer) ve BERT sınıflandırıcı (TFBertForSequenceClassification) kütüphaneleri indirilmiştir. BERT Tokenizer ile cümleler kelimelerine ayrıştırılmıştır. Problemin çözümlenmesi için belirteçler (token) cümlenin başına ve sonuna eklenmiştir, cümleler kısaysa boşluklar doldurulmuştur, uzunsa limite göre kelimeler ile ifade edilmiştir, akabinde maskelemeler oluşturulmuş ve tensor verisi oluşturularak, duygu eğitilerek etiketlere aktarılmıştır. Oluşturulan veriler dataloader değişkenine dönüştürülmüştür. Önceden eğitilmiş modele, Hugging Face'den indirilen BertForSequenceClassification yardımıyla ince ayar (fine-tune) yapılmıştır. Eğitim öncesi son adımda ise eğitim iterasyon sayısı belirtilmiş, eğitim verimini artırmak için öğrenme oranı iyileştiricisi eklenmiş ve Adam Optimizer kullanılmıştır (Hutto ve Gilbert, 2014).

Eğitim aşamasına geçmeden önce seed değeri sabit bir değere eşitlenmiştir, böylece tüm denemelerde aynı sonucun alınması amaçlanmıştır. Eğitim iterasyonu (epoch=4) toplam bölüm sayısı kadar yapılmıştır. Eğitim veri seti dataloader fonksiyonuna aktarılmıştır ve girdiler otuz ikişerli olarak modeli besleyerek eğitimi başlatmıştır. Her bölüm başlamadan önce optimize edilecek loss değeri sıfırlanmaktadır. Bu aşamada train metodu çağırılmıştır. Test aşamasında ise eval metodu çağırılmıştır. Çünkü modelin katmanları train ve eval metotlarında farklı olarak davranmaktadır.

Dataloader fonksiyonundaki değerler GPU'ya aktarılmıştır, gradient değerleri sıfırlanmıştır ve çıktı değerleri oluşmuştur. Bu çıktı (logit) değerlerine bağlı olarak kayıp (loss) değeri hesaplanmıştır. Backpropogation ile gradient'ler tekrar hesaplanmıştır ve son olarak da öğrenme oranıyla beraber parametreler de optimize edilmiştir.

Eğitimdeki model performansını incelemek için eğitim kaybındaki düşüşü incelenmiştir.

Eğitim veri setinde olduğu gibi, test veri seti için de bir dataloader oluşturulmuştur.

Test verisi kullanılarak modele sonuçları tahmin ettirilmiştir. Batch değerimiz 32 olduğu için, model eğitimde olduğu gibi prediction kısmında da girdiler modele otuz ikişerli verilmiştir. Bu nedenle flatten fonksiyonu ile bütün sonuçlar tek bir listede toplanmıştır ve *"prediction_set"* değişkeninde saklanmıştır.

Bu bir sınıflandırma problemi olduğu için performans metriklerinden F-score kullanılmıştır. Bu kısımda Precision, Recall ve F-score değerleri çıkarılmıştır.

# BEŞİNCİ BÖLÜM

# DENEYSEL ÇALIŞMA SONUÇLARI

## 5.1. Deneysel Çalışma Sonuçları

### 5.1.1. Veriseti Toplama Kod ve Sonuçları

Twitter API'sinden toplanan veri kümesinin genel görünümünü Şekil 5.1.1. göstermektedir. Bu veri kümesi, temel alanlardan oluşur: kimlik, kullanıcı_adı, tarih, metin, hashtag'ler, bunların büyük ölçüde gerekli olduğu ve duygu analizi için verileri analiz etmekle meşgul olduğu kısımlardır.

```
df=pd.read_csv("/content/gvac1024.csv")
df.head(5)
```

**Şekil 5.1.1:** Veri Seti Genel Görünümü

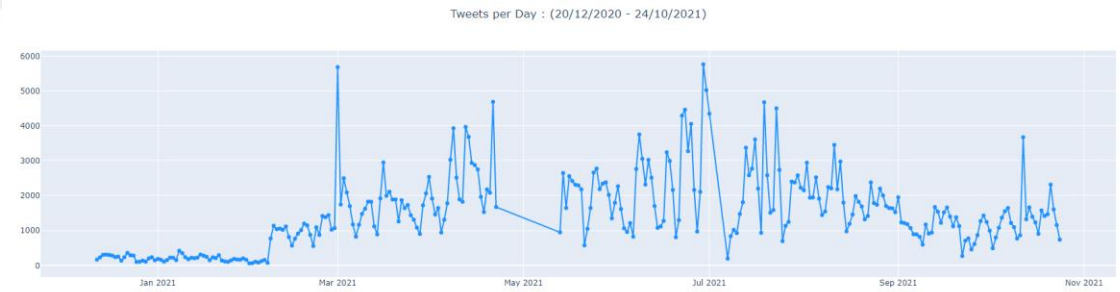
| | id | user_name | user_location | user_description | user_created | user_followers | user_friends | user_favourites | user_verified | date | text | hashtags | source | retweets | favorites | is_retweet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1340539111971516416 | Rachel Roh | La Crescenta-Montrose, CA | Aggregator of Asian American news; scanning di... | 2009-04-08 17:52:46 | 405 | 1692 | 3247 | False | 2020-12-20 06:06:44 | Same folks said daikon paste could treat a cyt... | ['PfizerBioNTech'] | Twitter for Android | 0 | 0 | False |
| 1 | 1338158543359250433 | Albert Fong | San Francisco, CA | Marketing dude, tech geek, heavy metal & '80s ... | 2009-09-21 15:27:30 | 834 | 666 | 178 | False | 2020-12-13 16:27:13 | While the world has been on the wrong side of ... | NaN | Twitter Web App | 1 | 1 | False |
| 2 | 1337858199140118533 | eli.txt | Your Bed | hell, hydra | 2020-06-25 23:30:28 | 10 | 88 | 155 | False | 2020-12-12 20:33:45 | #coronavirus #SputnikV #AstraZeneca #PfizerBio... | ['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf... | Twitter for Android | 0 | 0 | False |
| 3 | 1337855739918835717 | Charles Adler | Vancouver, BC - Canada | Hosting "CharlesAdlerTonight" Global News Radi... | 2008-09-10 11:28:53 | 49165 | 3933 | 21853 | True | 2020-12-12 20:23:59 | Facts are immutable, Senator, even when you're... | NaN | Twitter Web App | 446 | 2129 | False |
| 4 | 1337854064604966912 | Citizen News Channel | NaN | Citizen News Channel bringing you an alternati... | 2020-04-23 17:58:42 | 152 | 580 | 1473 | False | 2020-12-12 20:17:19 | Explain to me again why we need a vaccine @Bor... | ['whereareallthesickpeople', 'PfizerBioNTech'] | Twitter for iPhone | 0 | 0 | False |

Şekil 5.1.2.'de tweetlerin günlere göre dağılımı görülmektedir.

```
import plotly.graph_objects as go
df['tweet_date']=pd.to_datetime(df['date']).dt.date
tweet_date=df['tweet_date'].value_counts().to_frame().reset_index().rename(columns={'index':'date','tweet_date':'count'})
tweet_date['date']=pd.to_datetime(tweet_date['date'])
tweet_date=tweet_date.sort_values('date',ascending=False)

fig=go.Figure(go.Scatter(x=tweet_date['date'],
                         y=tweet_date['count'],
                        mode='markers+lines',
                        name="Submissions",
```

```
                                    marker_color='dodgerblue'))

fig.update_layout(
    title_text='Tweets per Day : ({} - {})'.format(df['tweet_date'].sort_values()[0].str
ftime("%d/%m/%Y"),
                                                   df['tweet_date'].sort_values().il
oc[-1].strftime("%d/%m/%Y")),
    title_x=0.5)


fig.show(renderer="colab")
```

**Şekil 5.1.2:** Günlere Göre Tweet Sayısı Dağılımı

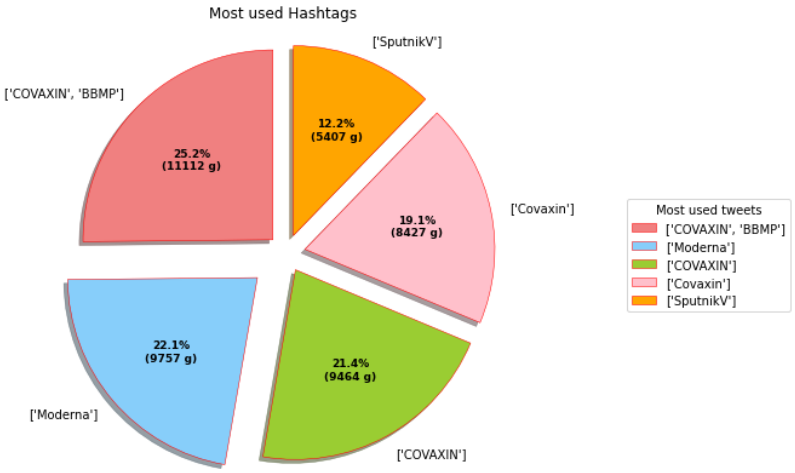



Şekil 5.1.3'de ise twitter mesajlarıyla birlikte en çok kullanılan hashtaglerin pasta grafiği gösterilmiştir.

```
MostUsedTweets = df.hashtags.value_counts().sort_values(ascending=False)[:5]
colors = ['lightcoral', 'lightskyblue', 'yellowgreen', 'pink', 'orange']
explode = (0.1, 0.2, 0.1, 0.1, 0.1)

wp = { 'linewidth' : 0.5, 'edgecolor' : "red" }

def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return "{:.1f}%\n({:d} g)".format(pct, absolute)

fig, ax = plt.subplots(figsize =(10, 7))
wedges, texts, autotexts = ax.pie(MostUsedTweets,
```

```
                                autopct = lambda pct: func(pct, MostUsedTweets),
                                explode = explode,
                                labels = MostUsedTweets.keys(),
                                shadow = True,
                                colors = colors,
                                startangle = 90,
                                wedgeprops = wp,
                                textprops = dict(color ="black"))
ax.legend(wedges, MostUsedTweets.keys(),
          title ="Most used tweets",
          loc ="center left",
          bbox_to_anchor =(1, 0, 0.5, 1))

plt.setp(autotexts, size=9, weight="bold")
ax.set_title("Most used Hashtags")
plt.axis('equal')
plt.show()
```

**Şekil 5.1.3:** En Sık Kullanılan Hashtagler




### 5.1.2. Ön İşleme Kod ve Sonuçları

Veri seti üzerinde gerçekleştirilen ön işleme stratejileri, Şekil 5.2.'te özetlenen sonuçları üretmektedir. Bu işlem, incelemelerdeki kelime sayısını ve kelime dağarcığındaki kelime sayısını önemli ölçüde azaltmaktadır. Ön işleme aşaması, gereksiz kelimeleri temizlenmesine ve kaldırılmasına yardımcı olmak için gereklidir.

```
df.text = df.text.apply(lambda x:re.sub('@[^\s]+','',x)) #remove twitter handlers

df.text = df.text.apply(lambda x:re.sub(r'\B#\S+','',x)) #remove hashtags

df.text = df.text.apply(lambda x:re.sub(r"http\S+", "", x)) #remove URLs

df.text = df.text.apply(lambda x:' '.join(re.findall(r'\w+', x))) #remove all the specia
l characters

df.text = df.text.apply(lambda x:re.sub(r'\s+[a-zA-
Z]\s+', '', x)) #remove all single characters

df.text = df.text.apply(lambda x:re.sub(r'\s+', ' ', x, flags=re.I)) #Substituting multi
ple spaces with single space

df.head(5)
```

**Şekil 5.2:** Ön İşleme Sonrası Veriseti Genel Görünümü

| | text | tokenized | No_stopwords | stemmed_porter | stemmed_snowball | lemmatized |
|---|---|---|---|---|---|---|
| 0 | same folks said daikon paste could treatcytoki... | [same, folks, said, daikon, paste, could, trea... | [folks, said, daikon, paste, could, treatcytok... | [folk, said, daikon, past, could, treatcytokin... | [folk, said, daikon, past, could, treatcytokin... | [folk, said, daikon, paste, could, treatcytoki... |
| 1 | while the world has been on the wrong side of ... | [while, the, world, has, been, on, the, wrong,... | [world, wrong, side, history, year, hopefully,... | [world, wrong, side, histori, year, hope, bigg... | [world, wrong, side, histori, year, hope, bigg... | [world, wrong, side, history, year, hopefully,... |
| 2 | russian vaccine is created to last 2 4 years | [russian, vaccine, is, created, to, last, 2, 4... | [russian, vaccine, created, last, 2, 4, years] | [russian, vaccin, creat, last, 2, 4, year] | [russian, vaccin, creat, last, 2, 4, year] | [russian, vaccine, created, last, 2, 4, year] |
| 3 | facts are immutable senator even when you re n... | [facts, are, immutable, senator, even, when, y... | [facts, immutable, senator, even, ethically, s... | [fact, immut, senat, even, ethic, sturdi, enou... | [fact, immut, senat, even, ethic, sturdi, enou... | [fact, immutable, senator, even, ethically, st... |
| 4 | explain to me again why we needvaccine | [explain, to, me, again, why, we, needvaccine] | [explain, needvaccine] | [explain, needvaccin] | [explain, needvaccin] | [explain, needvaccine] |

| | id | user_name | user_location | user_description | user_created | user_followers | user_friends | user_favourites | user_verified | date | text | hashtags | source | retweets | favorites | is_retweet | number of words in text | tweet_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1340539111971516416 | Rachel Roh | La Crescenta-Montrose, CA | Aggregator of Asian American news; scanning di... | 2009-04-08 17:52:46 | 405 | 1692 | 3247 | False | 2020-12-20 06:06:44 | Same folks said daikon paste could treatcytoki... | ['PfizerBioNTech'] | Twitter for Android | 0 | 0 | False | 97 | 2020-1 |
| 1 | 1338158543359250433 | Albert Fong | San Francisco, CA | Marketing dude, tech geek, heavy metal & '80s ... | 2009-09-21 15:27:30 | 834 | 666 | 178 | False | 2020-12-13 16:27:13 | While the world has been on the wrong side of ... | NaN | Twitter Web App | 1 | 1 | False | 140 | 2020-1 |
| 2 | 1337858199140118533 | eliｚｔｅｕ | Your Bed | heil, hydra | 2020-06-25 23:30:28 | 10 | 88 | 155 | False | 2020-12-12 20:33:45 | Russian vaccine is created to last 2 4 years | ['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf... | Twitter for Android | 0 | 0 | False | 140 | 2020-1 |
| 2 | 1337858199140118533 | eliｚｔｅｕ | Your Bed | heil, hydra | 2020-06-25 23:30:28 | 10 | 88 | 155 | False | 2020-12-12 20:33:45 | Russian vaccine is created to last 2 4 years | ['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf... | Twitter for Android | 0 | 0 | False | 140 | 2020-1 |
| 2 | 1337858199140118533 | eliｚｔｅｕ | Your Bed | heil, hydra | 2020-06-25 23:30:28 | 10 | 88 | 155 | False | 2020-12-12 20:33:45 | Russian vaccine is created to last 2 4 years | ['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf... | Twitter for Android | 0 | 0 | False | 140 | 2020-1 |

### 5.1.3. VADER Kod ve Sonuçları

Şekil 5.3.1, VADER Sentiment Analyzer tarafından elde edilen pozitif, negatif, nötr veya bileşik olarak her bir tweet'in duygu puanını göstermektedir.

**Şekil 5.3.1:** Vader Kullanan Tweetlerin Duygu Puanı

```
Summary statistics of numerical features :

id  user_followers  ...  Number_Of_Words  Mean_Word_Length

count  2.129820e+05    2.129820e+05  ...    212982.000000     212982.000000

mean   1.403890e+18    9.464582e+04  ...        13.453311          4.627621

std    2.657192e+16    8.271028e+05  ...         5.372148          1.028193

min    1.337728e+18    0.000000e+00  ...         1.000000          0.000000

25%    1.381004e+18    5.700000e+01  ...        10.000000          3.950000

50%    1.406080e+18    3.340000e+02  ...        14.000000          4.500000

75%    1.424304e+18    1.515000e+03  ...        17.000000          5.150000

max    1.452377e+18    1.622620e+07  ...        42.000000         45.000000


[8 rows x 11 columns]
```

Şekil 5.3.2, eşikler uygulandıktan sonra tweet'lerin pozitif, nötr veya negatif olarak sınıflandırılmasını göstermektedir. VADER kullanarak tweet'leri doğrudan pozitif, negatif veya nötr olarak kategorize eden uygun bir eşik değeri seçilmiştir.

Şekil 5.3.2, puanlama kuralına bağlı olarak her tweet'in genel duygu puanları değerini ve polaritesini göstermektedir ve tweet'leri Twitter veri setinde olumlu, olumsuz veya tarafsız bir duygu olarak sınıflandırmaktadır.

```
sid = SIA()


data['sentiments']           = data['text'].apply(lambda x: sid.polarity_scores(' '.join
(re.findall(r'\w+',x.lower()))))
data['Positive Sentiment']   = data['sentiments'].apply(lambda x: x['pos']+1*(10**-6))
data['Neutral Sentiment']    = data['sentiments'].apply(lambda x: x['neu']+1*(10**-6))
data['Negative Sentiment']   = data['sentiments'].apply(lambda x: x['neg']+1*(10**-6))


data.drop(columns=['sentiments'],inplace=True)
```

**Şekil 5.3.2:** Her Tweet İçin Genel Duygu Polaritesi

| date | text | hashtags | source | retweets | favorites | is_retweet | Tidy Tweet | Tidy hashtags | Sentiment | Positive Sentiment | Neutral Sentiment | Negative Sentiment |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2020-12-20 | Same folks said daikon paste could treatcytoki... | ['PfizerBioNTech'] | Twitter for Android | 0 | 0 | False | folk said daikon past could treat cytokin stor... | | Positive | 0.000001 | 1.000001 | 0.000001 |
| 2020-12-13 | While the world has been on the wrong side of ... | | Twitter Web App | 1 | 1 | False | world wrong side histori year hope biggest vac... | | Negative | 0.109001 | 0.766001 | 0.125001 |
| 2020-12-12 | Russian vaccine is created to last 2 4 years | ['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf... | Twitter for Android | 0 | 0 | False | coronavirus sputnikv astrazeneca pfizerbiontec... | sputnikv astrazeneca pfizerbiontech moderna | Neutral | 0.250001 | 0.750001 | 0.000001 |
| 2020-12-12 | Facts are immutable Senator even when you re n... | | Twitter Web App | 446 | 2129 | False | fact immut senat even your ethic sturdi enough... | | Neutral | 0.000001 | 1.000001 | 0.000001 |
| 2020-12-12 | Explain to me again why we needvaccine | ['whereareallthesickpeople', 'PfizerBioNTech'] | Twitter for iPhone | 0 | 0 | False | explain need vaccin whereareallthesickpeopl | | Neutral | 0.000001 | 1.000001 | 0.000001 |

```
def user_popularity(sir):
    teta = 0
    if sir['user_verified'] == 'True':
        teta=1
    else:
        teta=0.5
    rho = np.sqrt(sir['user_followers']*0.65+sir['user_friends']*0.25+sir['user_favourites']*0.10)*teta
    return np.round(rho)

df['user_popularity'] = df.apply(user_popularity,axis=1)

user_df = df.groupby(by='user_name').mean().reset_index()
user_df[['user_name','user_popularity','Positive Sentiment','Negative Sentiment']].sample(5)
```

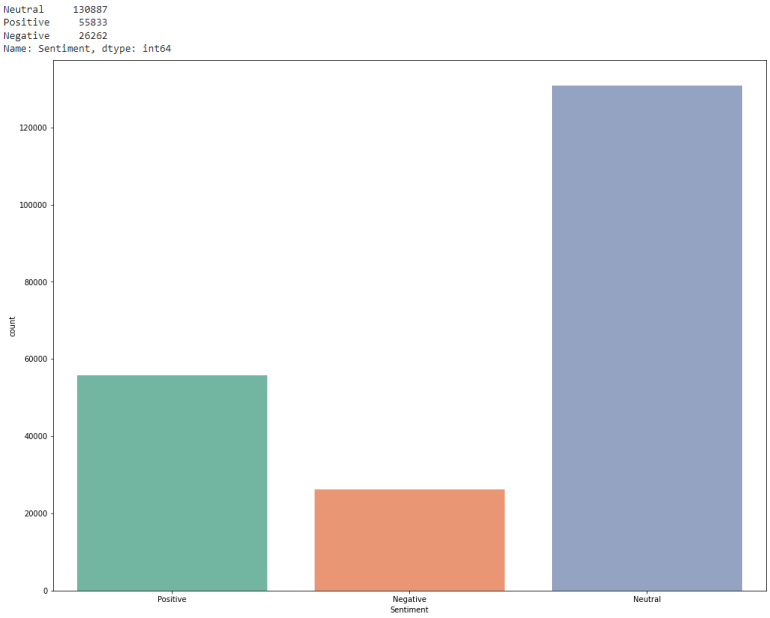
**Şekil 5.3.3:** Her Kullanıcı İçin Duygu Polaritesi

| | user_name | user_popularity | Positive Sentiment | Negative Sentiment |
|---|---|---|---|---|
| **26673** | Indian Advocate | 38.142857 | 0.060858 | 0.097858 |
| **5824** | Anne Antonio-Teves | 5.000000 | 0.512001 | 0.000001 |
| **67389** | WinniePooh | 8.500000 | 0.275501 | 0.000001 |
| **31026** | Judith Berkens | 30.000000 | 0.217001 | 0.000001 |
| **2624** | Adrian Murphy | 18.000000 | 0.000001 | 0.000001 |
| **79843** | Boyabenyathi | 41.000000 | 0.116751 | 0.079001 |
| **23832** | Greg #ClimateEmergency #ZeroCOVID19 #onpoli | 78.000000 | 0.000001 | 0.000001 |
| **79753** | library music • news•my poetry 音楽中心•私の詩 | 19.000000 | 0.000001 | 0.000001 |
| **62711** | The3ndEYE1 | 20.000000 | 0.077001 | 0.165501 |
| **59978** | Sunshine Pictures | 8.000000 | 0.066001 | 0.000001 |

Genel duyguların üç farklı sınıfa dağılımı, Şekil 5.3.3'te gösterilen duyarlılık değerlerine göre pozitif, nötr ve negatif şeklindedir. Şekil 5.5.3, bu veri setindeki duyarlılık değerlerine göre üç sınıfa ait toplam tweet sayısını göstermektedir: pozitif, nötr, negatif. Burada gösterilen sonuçlara dayanarak, veri kümemizdeki çoğu tweet, COVID-19 aşısı hakkında olumlu veya tarafsız görüşler ifade etmekte ve negatif görüşlerin azınlıkta olduğu yorumlanmaktadır.

```
sentiment = SentimentIntensityAnalyzer()

def get_sentiment(data):

    sentiment_list = []

    for text in list(data['Tidy Tweet'].values):

        if sentiment.polarity_scores(text)["compound"] > 0:

            sentiment_list.append("Positive")

        elif sentiment.polarity_scores(text)["compound"] < 0:

            sentiment_list.append("Negative")

        else:

            sentiment_list.append("Neutral")

    return sentiment_list


data['Sentiment'] = get_sentiment(data)

sns.countplot(x="Sentiment", data=data, palette="Set2")

print(data.Sentiment.value_counts())
```

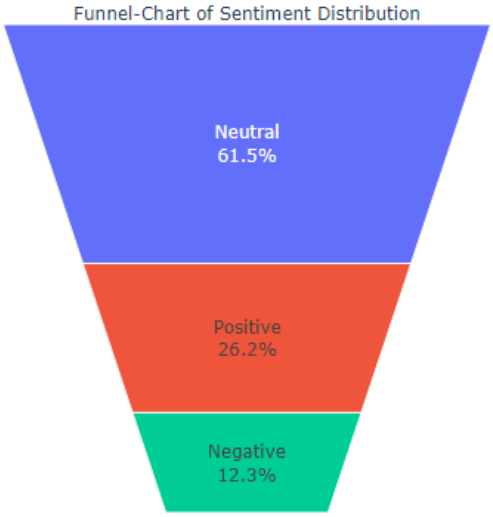
**Şekil 5.3.4:** Genel Duygular Dağılımı




Ancak ilginç bir şekilde, Şekil 5.3.5.'te görüldüğü gibi tweetlerin %26,2'sı olumlu, %12,3'ü olumsuz, %61,5'i ise tarafsız görüş ile sonuçlanmaktadır.

```
plt.figure(figsize=(12,6))
sns.countplot(x='Sentiment',data=data)
fig = go.Figure(go.Funnelarea(
    text =temp.Sentiment,
    values = temp.text,
    title = {"position": "top center", "text": "Funnel-Chart of Sentiment Distribution"}
    ))
fig.show(renderer="colab")
```

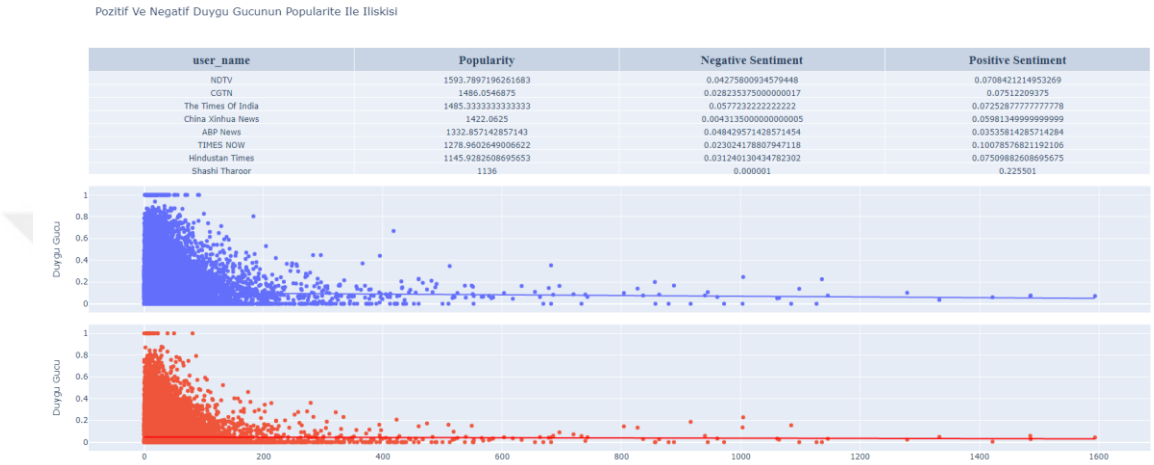
**Şekil 5.3.5:** Duyarlılık Sınıflandırması Dağılımı Huni Grafiği




```
user_df=user_df.sort_values(by='user_popularity',ascending=False)

fig = make_subplots(rows=3, cols=1, shared_xaxes=True, vertical_spacing=0.03, specs=[[{"
type": "table"}], [{"type": "scatter"}], [{"type": "scatter"}]])


fig.add_trace(go.Scatter(x=user_df["user_popularity"], y=user_df["Positive Sentiment"],
mode="markers", name="Positive Sentiment"), row=2, col=1)


fig.add_trace(go.Scatter(x=user_df["user_popularity"], y=user_df["Negative Sentiment"],
mode="markers", name="Negative Sentiment"), row=3, col=1)


fig.add_trace(go.Table( header=dict( values=['<b>user_name<b>',"<b>Popularity<b>",'<b>Ne
gative Sentiment<b>','<b>Positive Sentiment<b>'],

            font=dict(size=19,family="Lato"), align="center"), cells=dict(values=[user_d
f[k].tolist() for k in ['user_name',"user_popularity",'Negative Sentiment','Positive Sen
timent']],

            align = "center", font=dict(size=12)),), row=1, col=1)


figg = ex.scatter(user_df, x='user_popularity', y='Positive Sentiment', trendline='ols')

trendline = figg.data[1]

fig.add_trace(trendline)

figg = ex.scatter(user_df, x='user_popularity', y='Negative Sentiment', trendline='ols',
 color_discrete_sequence=['red'])

trendline = figg.data[1]

fig.add_trace(trendline,3,1)
```

```
fig.update_layout(height=800, showlegend=False, title_text="Pozitif Ve Negatif Duygu Guc
unun Popularite Ile Iliskisi",)

fig.update_yaxes(title_text="Duygu Gucu")

fig.show(renderer="colab")
```

**Şekil 5.3.6:** Pozitif ve Negatif Duygu Gücünün Popularite İle İlişkisi



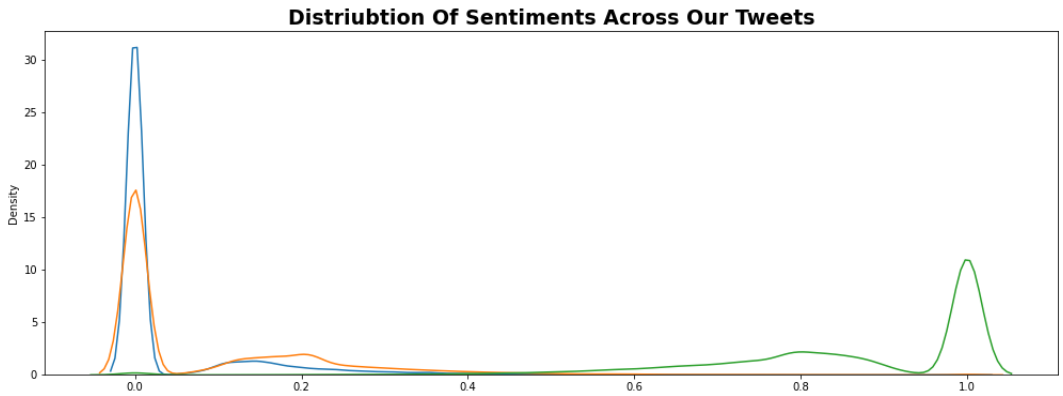

### 5.1.4. KDE Dağılımının Kod ve Sonuçları

Etiket (hashtag) verilerine dayalı KDE grafiği, her bir duyarlılığın tahmini dağılımını sağlayan Şekil 5.4.1'te sunulmaktadır. KDE grafiğini uygulamak için üst düzey arayüz, Matplotlib tabanlı bir Python veri görselleştirme kitaplığı olan Seaborn tarafından KDE grafiklerinin çizilmesi için sağlanmıştır. Şekil 5.4.1, tweet'lerdeki olumsuz, tarafsız ve olumlu duyguların duygu değerlerine göre normal dağılımını göstermektedir. Temelde duyarlılık değerleri -0.5 ile 1.5 arasındadır. Pozitif, negatif ve nötr değerler için sırasıyla yeşil, kırmızı ve turuncu renkler kullanılmıştır. Baskın duygunun tarafsız olduğu da görülmektedir. Aşağıdaki grafikten, tweet'lerdeki olumlu ve olumsuz duygulara göre tarafsız duyguların dağılımının daha yüksek olduğu yorumlanabilmektedir.

```
plt.subplot(2,1,1)

plt.title('Distriubtion Of Sentiments Across Our Tweets',fontsize=19,fontweight='bold')

sns.kdeplot(data['Negative Sentiment'],bw=0.1)

sns.kdeplot(data['Positive Sentiment'],bw=0.1)

sns.kdeplot(data['Neutral Sentiment'],bw=0.1)
```

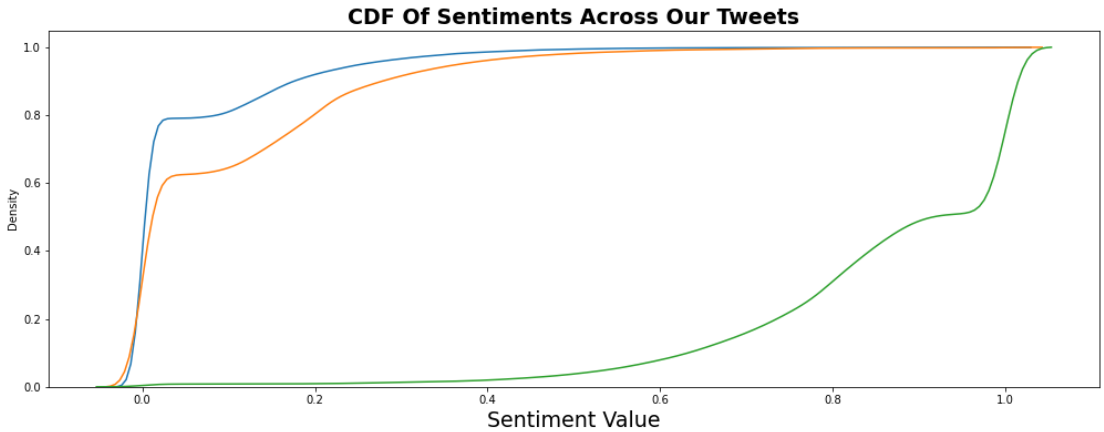
```
plt.subplot(2,1,2)
plt.title('CDF Of Sentiments Across Our Tweets',fontsize=19,fontweight='bold')
sns.kdeplot(data['Negative Sentiment'],bw=0.1,cumulative=True)
sns.kdeplot(data['Positive Sentiment'],bw=0.1,cumulative=True)
sns.kdeplot(data['Neutral Sentiment'],bw=0.1,cumulative=True)
plt.xlabel('Sentiment Value',fontsize=19)
plt.show()
```

**Şekil 5.4.1:** Tweetlerimiz Arasındaki Duyguların Normal Dağılımı




Şekil 5.4.2, standart normal dağılımın kümülatif dağılım fonksiyonunu göstermektedir. Genel duygular, duyarlılık değerlerine ve yoğunluklarına göre olumlu, tarafsız ve olumsuz olarak dağılmaktadır.

**Şekil 5.4.2:** Tweetlerimizdeki Duyguların Kümülatif Dağılım Fonksiyonu



### 5.1.5. Kelime Bulutu Kod ve Sonuçları

Aşağıdaki Şekil 5.5.1 ve Şekil 5.5.2, ilk on olumlu ve olumsuz tweet kelimesinden biriyle başlayan 15 cümlenin trigramını göstermektedir. Cümlenin olasılığı rastgele 'son derece' olumsuz tweet'te görünecektir. Sözcükler, olumlu veya olumsuz çağrışımlar veya olumluluk ve olumsuzluk dereceleri olarak sınıflandırılmaktadır. Bir cümlenin genel duyarlılığı, kelimelerin duyguları toplanarak hesaplanmaktadır. Birkaç tweet'i daha okursak, bunun genellikle kusurlu olduğu sonucuna varabiliriz, ancak ortalama olarak doğru sonuçlara varmaktadır.

```
import nltk
nltk.download('punkt')


token=nltk.word_tokenize(' '.join(df.text))
pos_bigram=ngrams(token,2)
pos_bigram_dict = dict()
pos_trigram =ngrams(token,3)
pos_trigram = [k for k in pos_trigram if k[0] in top_10_pos]


token=nltk.word_tokenize(' '.join(df.text))
neg_bigram=ngrams(token,2)
neg_bigram_dict = dict()
neg_trigram =ngrams(token,3)
neg_trigram = [k for k in neg_trigram if k[0] in top_10_neg]


for i in neg_bigram:
    neg_bigram_dict[i] = neg_bigram_dict.get(i,0)+1
for i in pos_bigram:
    pos_bigram_dict[i] = pos_bigram_dict.get(i,0)+1


pos_trigram_dict = dict()
neg_trigram_dict = dict()


for i in pos_trigram:
    pos_trigram_dict[i] = pos_trigram_dict.get(i,0)+1
```

```
for i in neg_trigram:
    neg_trigram_dict[i] = neg_trigram_dict.get(i,0)+1


pos_trigram_df = pd.DataFrame(random.sample(list(pos_trigram_dict.keys()),k=15),columns=
['One Of Top 10 Words','Second Word','Third Word'])


def get_prob(sir):
    key = (sir['One Of Top 10 Words'],sir['Second Word'],sir['Third Word'])
    w3 = pos_trigram_dict[key]
    w2 = pos_bigram_dict[(sir['One Of Top 10 Words'],sir['Second Word'])]
    return w3/w2


pos_trigram_df['Probabilty Of Sentence'] = pos_trigram_df.apply(get_prob,axis=1)


pos_trigram_df.style.background_gradient(subset='Probabilty Of Sentence',cmap='vlag')
```

**Şekil 5.5.1:** En Sık Kullanılan 10 Pozitif Kelimeyle Başlayan 15 Cümle ve Olasılıkları

| | One Of Top 10 Words | Second Word | Third Word | Probabilty Of Sentence |
|---|---|---|---|---|
| 0 | vaccine | developed | by | 0.894260 |
| 1 | vaccine | shortage | across | 0.011364 |
| 2 | vaccine | can | drop | 0.022388 |
| 3 | vaccine | doses | Jan | 0.000952 |
| 4 | vaccine | Itthe | second | 0.111111 |
| 5 | vaccinated | so | grateful | 0.108108 |
| 6 | amp | Do | youdid | 0.200000 |
| 7 | vaccine | attacking | Covid | 1.000000 |
| 8 | vaccinated | and | vaccinated | 0.002825 |
| 9 | vaccine | at | A | 0.001642 |
| 10 | vaccine | dose | less | 0.006186 |
| 11 | vaccine | Berhad | has | 1.000000 |
| 12 | vaccine | wit | Study | 0.333333 |
| 13 | vaccine | to | children | 0.038908 |
| 14 | vaccine | out | to | 0.058824 |

```
eg_trigram_df = pd.DataFrame(random.sample(list(neg_trigram_dict.keys()),k=15),columns=[
'One Of Top 10 Words','Second Word','Third Word'])


def get_prob(sir):
    key = (sir['One Of Top 10 Words'],sir['Second Word'],sir['Third Word'])
    w3 = neg_trigram_dict[key]
    w2 = neg_bigram_dict[(sir['One Of Top 10 Words'],sir['Second Word'])]
    return w3/w2


neg_trigram_df['Probabilty Of Sentence'] = neg_trigram_df.apply(get_prob,axis=1)


neg_trigram_df.style.background_gradient(subset='Probabilty Of Sentence',cmap='vlag')
```

**Şekil 5.5.2:** En Sık Kullanılan 10 Negatif Kelimeyle Başlayan 15 Cümle ve Olasılıkları

| | One Of Top 10 Words | Second Word | Third Word | Probabilty Of Sentence |
|---|---|---|---|---|
| 0 | amp | approved | delay | 0.107143 |
| 1 | amp | Moderna | not | 0.036364 |
| 2 | people | craving | for | 1.000000 |
| 3 | vaccine | today | Fantastic | 0.000712 |
| 4 | vaccine | unlike | and | 0.571429 |
| 5 | vaccine | 08 | 44 | 1.000000 |
| 6 | amp | safe | vaccine | 0.066667 |
| 7 | vaccine | 36 | hours | 1.000000 |
| 8 | vaccine | and | there | 0.006276 |
| 9 | vaccine | program | performed | 0.038462 |
| 10 | No | waste | Got | 0.333333 |
| 11 | amp | 100 | 5 | 0.066667 |
| 12 | vaccine | Chose | not | 1.000000 |
| 13 | The | current | manufacturing | 0.033333 |
| 14 | The | Canton | KS | 1.000000 |

```
l_t = Most_Positive_text


w1_dict = dict()
for word in l_t.split():
    w= word.strip()
    if w in STOPWORDS:
        continue
    else:
        w1_dict[w] = w1_dict.get(w,0)+1
 w1_dict = {k: v for k, v in sorted(w1_dict.items(), key=lambda item: item[1],reverse=Tr
ue)}


l_t = Most_Negative_text
w2_dict = dict()
for word in l_t.split():
    w= word.strip()
    if w in STOPWORDS:
        continue
    else:
        w2_dict[w] = w2_dict.get(w,0)+1
w2_dict = {k: v for k, v in sorted(w2_dict.items(), key=lambda item: item[1],reverse=Tru
e)}


top_10_pos = list(w1_dict.keys())[:10]
top_10_neg = list(w2_dict.keys())[:10]


plt.subplot(1,2,1)
w_c = WordCloud(width=600,height=400,collocations = False,colormap='nipy_spectral',backg
round_color='white').generate(' '.join(top_10_pos))
plt.title('Top 10 Words In Most Positive Tweets',fontsize=19,fontweight='bold')
plt.imshow(w_c)
plt.axis('off')
plt.subplot(1,2,2)
w_c = WordCloud(width=600,height=400,collocations = False,colormap='nipy_spectral',backg
round_color='white').generate(' '.join(top_10_neg))
plt.title('Top 10 Words In Most Negative Tweets',fontsize=19,fontweight='bold')
plt.imshow(w_c)
```

```
plt.axis('off')

plt.show()
```

**Şekil 5.5.3:** Pozitif ve Negatif On Duygunun Kelime Bulutu


Şekil 5.5.3., kelime bulutunu kullanarak en olumsuz duyguları ve en olumlu duyguları göstermektedir. Şekil 5.5.4 ise en iyi olumlu ve en olumsuz duyguları oluşturan en sık kullanılan kelimeleri göstermektedir. Şekil 5.5.5 ise en iyi olumlu ve en olumsuz duyguları oluşturan en sık kullanılan etiketleri göstermektedir.

```
Most_Positive = df[df['Positive Sentiment'].between(0.4,1)]

Most_Negative = df[df['Negative Sentiment'].between(0.25,1)]

Most_Positive_text = ' '.join(Most_Positive.text)

Most_Negative_text = ' '.join(Most_Negative.text)

pwc = WordCloud(width=600,height=400,collocations = False,background_color='white').gene
rate(Most_Positive_text)

nwc = WordCloud(width=600,height=400,collocations = False,background_color='white').gene
rate(Most_Negative_text)


plt.subplot(1,2,1)

plt.title('Common Words Among Most Positive Tweets',fontsize=16,fontweight='bold')

plt.imshow(pwc)

plt.axis('off')

plt.subplot(1,2,2)

plt.title('Common Words Among Most Negative Tweets',fontsize=16,fontweight='bold')

plt.imshow(nwc)

plt.axis('off')

plt.show()
```

**Şekil 5.5.4:** En İyi Olumlu Ve Olumsuz Duyguların Kelime Bulutu


```
Most_Positive_ht = ' '.join(Most_Positive[Most_Positive.hashtags.notna()].hashtags)

Most_Negative_ht = ' '.join(Most_Negative[Most_Negative.hashtags.notna()].hashtags)


Most_Positive_ht = re.sub(r'\W', ' ', Most_Positive_ht)

Most_Negative_ht = re.sub(r'\W', ' ', Most_Negative_ht)

Most_Positive_ht = re.sub(r'\s+', ' ', Most_Positive_ht, flags=re.I)

Most_Negative_ht = re.sub(r'\s+', ' ', Most_Negative_ht, flags=re.I)


pwc = WordCloud(width=600,height=400,collocations = False,background_color='white').gene
rate(Most_Positive_ht)

nwc = WordCloud(width=600,height=400,collocations = False,background_color='white').gene
rate(Most_Negative_ht)


plt.subplot(1,2,1)

plt.title('Common Hashtags Among Most Positive Tweets',fontsize=16,fontweight='bold')

plt.imshow(pwc)

plt.axis('off')

plt.subplot(1,2,2)

plt.title('Common Hashtags Among Most Negative Tweets',fontsize=16,fontweight='bold')

plt.imshow(nwc)

plt.axis('off')

plt.show()
```

**Şekil 5.5.5:** En İyi Olumlu Ve Olumsuz Duyguları Oluşturan Etiketlerin Kelime Bulutu


### 5.1.6. Günlük Dağılım Analizi Kod ve Sonuçları

Burada Şekil 5.6.1., pozitif ve negatif duygular için ortalama ve standart sapmayı (SD) göstermektedir ve her bir bölüm için zaman çizelgesi üzerinde günlük duyguları dağıtmak üzere üç bölüm periyoduna bölünmüştür.

**Şekil 5.6.1:** Pozitif Ve Negatif Duyguların Ortalama ve Standart Sapmaları

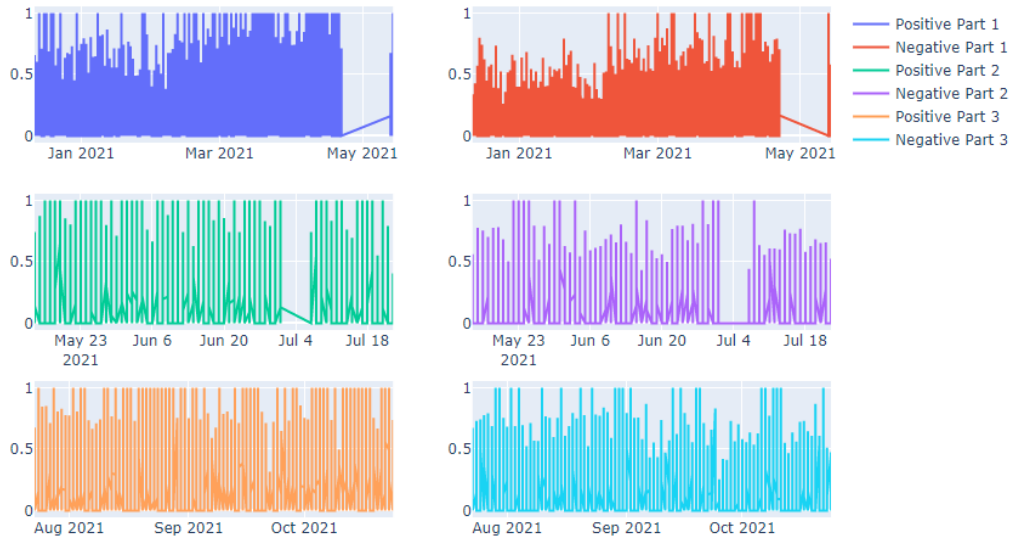
| | Partition_1_Mean | Partition_2_Mean | Partition_3_Mean | Partition_1_SD | Partition_2_SD | Partition_3_SD |
|---|---|---|---|---|---|---|
| **Positive Sentiment** | 0.097982 | 0.080880 | 0.103484 | 0.158884 | 0.144021 | 0.149381 |
| **Negative Sentiment** | 0.048714 | 0.037541 | 0.038407 | 0.109696 | 0.098036 | 0.102786 |

```
 fig = make_subplots(rows=4, cols=2, subplot_titles=('Observed Pos', 'Observed Neg', 'Tr
end Pos','Trend Neg','Seasonal Pos','Seasonal Neg','Residual Pos','Residual Neg'))

b_date_mean = df.groupby(by='date').mean().reset_index()


lbl = ['Positive','Negative']

from statsmodels.tsa.seasonal import seasonal_decompose

for idx,column in enumerate(['Positive Sentiment','Negative Sentiment']):

    res = seasonal_decompose(b_date_mean[column], freq=3)


    fig.add_trace(

    go.Scatter(x=np.arange(0,len(res.observed)), y=res.observed,name='{} Observed'.forma
t(lbl[idx])),

    row=1, col=idx+1)


    fig.add_trace(

    go.Scatter(x=np.arange(0,len(res.trend)), y=res.trend,name='{} Trend'.format(lbl[idx
])),

    row=2, col=idx+1)
```

```
    fig.add_trace(
    go.Scatter(x=np.arange(0,len(res.seasonal)), y=res.seasonal,name='{} Seasonal'.forma
t(lbl[idx])),
    row=3, col=idx+1)


    fig.add_trace(
    go.Scatter(x=np.arange(0,len(res.resid)), y=res.resid,name='{} Residual'.format(lbl[
idx])),
    row=4, col=idx+1)


fig.update_layout(height=600, width=900, title_text="Decomposition Of Our Sentiments int
o Trend,Level,Seasonality and Residuals")
fig.show(renderer="colab")
```

Şekil 5.6.2'da görüldüğü üzere tweet'ler zaman içinde farklı noktalarda arttığı tespit edilen olumlu ve olumsuz duyguları ifade ettiğinden, duygular her bölüme günlük olarak dağıtılmaktadır. Tweetlerin duyarlılığının sabit olmayan ortalama ve varyans açısından durağanlık gereksinimlerini karşılamadığını görüyoruz. Yukarıdaki kod hücresinde, hipotezimizi verilerimizin üç bölümü üzerinde test ettik. Verilerde bazı eğilimler olduğu gözlemlenmektedir.

**Şekil 5.6.2:** Her Bölümün Zaman Çizelgesi Üzerinden Günlük Duyguların Dağılımı

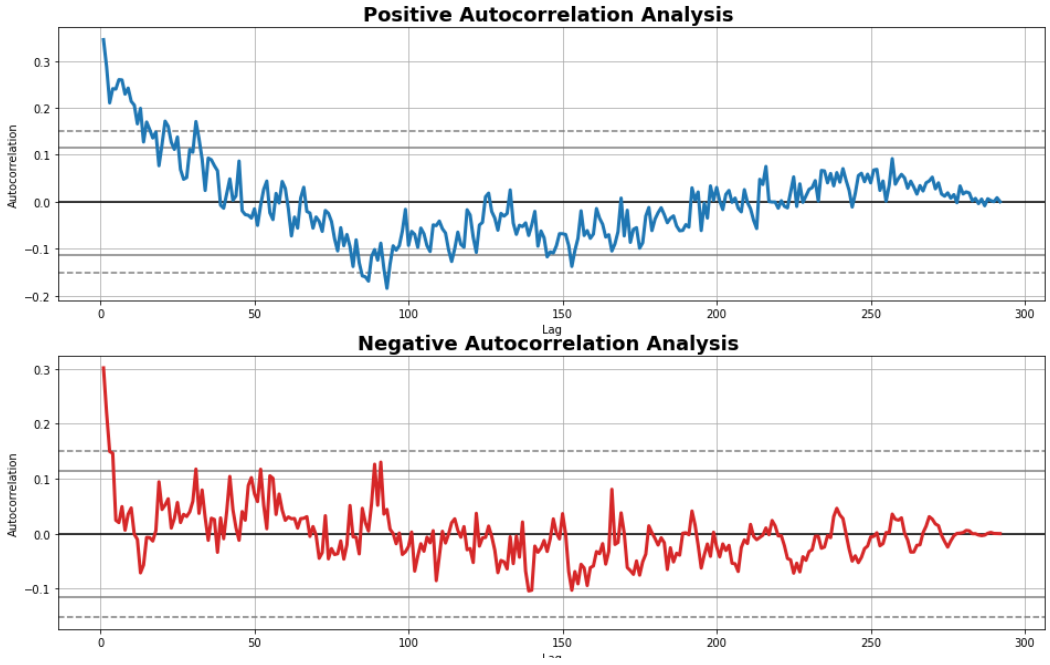


### 5.1.7. Otokorelasyon Analizi ve Duyguların Bileşenlerine Ayrılması İçin Kod ve Sonuçları

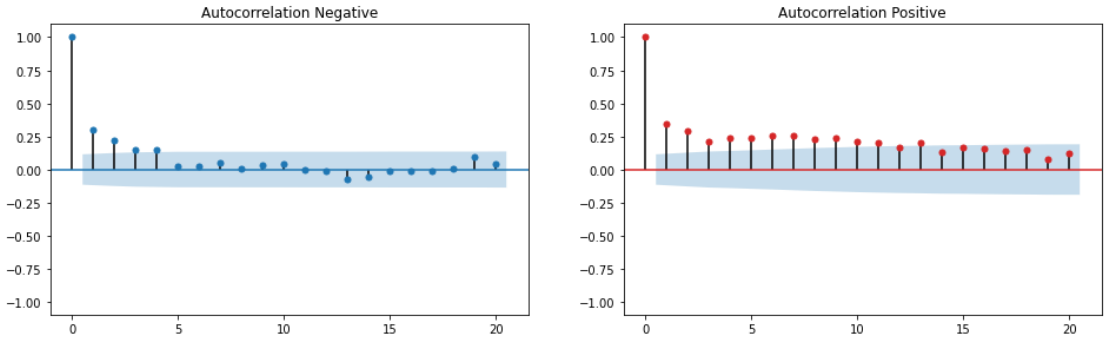
Şekil 5.7.1.'den, otokorelasyon için bu değerlerin %95 güven aralığında olduğu (düz gri çizgi ile temsil edilmektedir) gözlemlenmektedir. Gecikme > 0 için, verilerimizin herhangi bir otokorelasyonu olmadığını doğrulanmaktadır.

```
f, ax = plt.subplots(nrows=2, ncols=1, figsize=(16, 10))

ax[0].set_title('Positive Autocorrelation Analysis ',fontsize=18,fontweight='bold')
autocorrelation_plot(b_date_mean['Positive Sentiment'],ax=ax[0],lw=3)
ax[1].set_title('Negative Autocorrelation Analysis ',fontsize=18,fontweight='bold')
autocorrelation_plot(b_date_mean['Negative Sentiment'],ax=ax[1],color='tab:red',lw=3)
plt.show()
```

**Şekil 5.7.1:** Pozitif ve Negatif Otokorelasyon Analizi



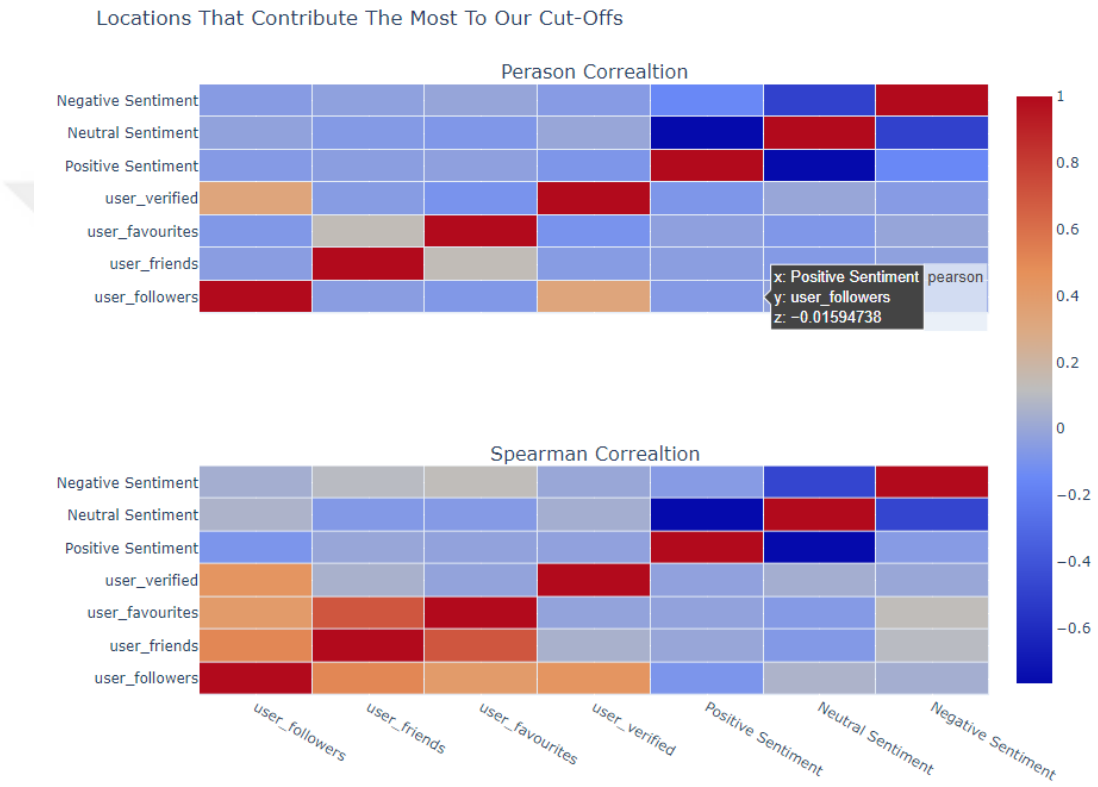
**Şekil 5.7.2:** Olumlu ve Olumsuz Duyguların Otokorelasyonu




```
 fig = make_subplots(rows=2, cols=1,shared_xaxes=True,subplot_titles=('Perason Correalti
on',  'Spearman Correaltion'))

colorscale=     [[1.0              , "rgb(165,0,38)"],

                [0.8888888888888888, "rgb(215,48,39)"],

                [0.7777777777777778, "rgb(244,109,67)"],

                [0.6666666666666666, "rgb(253,174,97)"],

                [0.5555555555555556, "rgb(254,224,144)"],

                [0.4444444444444444, "rgb(224,243,248)"],

                [0.3333333333333333, "rgb(171,217,233)"],

                [0.2222222222222222, "rgb(116,173,209)"],

                [0.1111111111111111, "rgb(69,117,180)"],

                [0.0               , "rgb(49,54,149)"]]


s_val =df[['user_followers','user_friends','user_favourites','user_verified','Positive S
entiment','Neutral Sentiment','Negative Sentiment']].corr('pearson')

s_idx = s_val.index

s_col = s_val.columns

s_val = s_val.values

fig.add_trace(go.Heatmap(x=s_col,y=s_idx,z=s_val,name='pearson',showscale=False,xgap=1,y
gap=1,colorscale=colorscale),row=1, col=1)


 s_val =df[['user_followers','user_friends','user_favourites','user_verified','Positive
Sentiment','Neutral Sentiment','Negative Sentiment']].corr('spearman')

s_idx = s_val.index

s_col = s_val.columns

s_val = s_val.values
```

```
fig.add_trace(go.Heatmap(x=s_col,y=s_idx,z=s_val,xgap=1,ygap=1,colorscale=colorscale), r
ow=2, col=1)


 fig.update_layout(height=700, width=900, title_text="Locations That Contribute The Most
 To Our Cut-Offs")

fig.show(renderer="colab")
```

**Şekil 5.7.3:** Pearson ve Spearman Korelasyon Analizi




Şekil 5.7.4'de, seriden çıkarılan eğilim (trend) ve mevsimsellik (seasonal) bilgilerinin makul olduğu gözlemlenmektedir. Kalıntı bilgiler (residual) de serideki yüksek değişkenlik eğilimlerini göstermektedir.

**Şekil 5.7.4:** Duyguların Eğilimler, Düzey, Mevsimsellik ve Kalıntılar Olarak Ayrıştırılması


### 5.1.8. Günlük Eğilim Analizi Kod ve Sonuçları

Zaman serisi analizinin uygulanması, tarihler üzerinden günlük tweet'leri gösteren Şekil 5.8.'de bir grafikle temsil edilmektedir. Günlük tweetler y ekseninde, tarihler ise x ekseninde gösterilir. Genel olarak on aylık veriler burada toplanır, burada her bir gün belirli sayıda tweet'e sahiptir. Twitter etkinliklerinin zirvelerini belirleyen trend analizini kullanarak en son haber güncellemelerini elde edilmektedir. Bu zamanlara ait haberler araştırılarak zamansal noktalarla birleştirilmektedir.

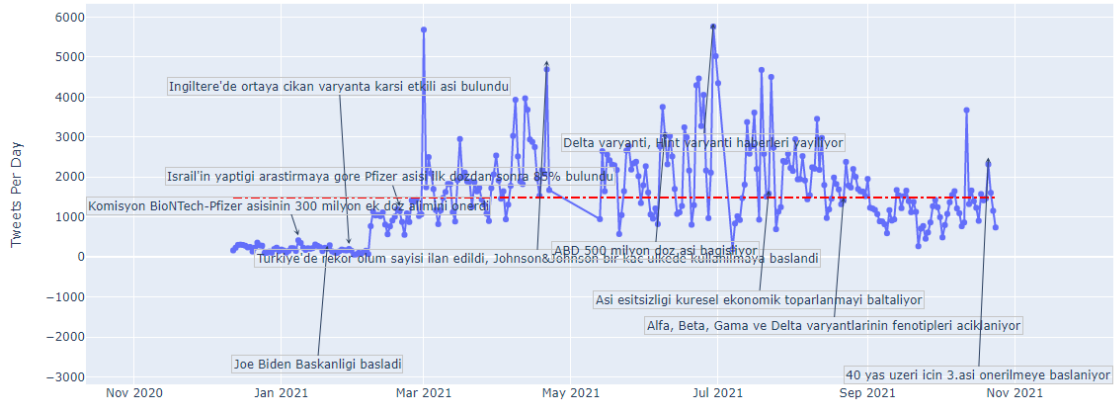
Haber bilgileri şöyledir, (1) Komisyon 300 milyona kadar ek BioNTech-Pfizer aşısı satın almayı teklif ediyor, (2) Joe Biden başkanlığı başladı, (3) Aşının Birleşik Krallık'ta keşfedilen varyanta karşı etkili olduğu bulundu, (4) İsrail'de yapılan bir araştırmaya Pfizer aşısını ilk dozdan sonra %85 etkili bulundu, (5) Türkiye'de rekor ölüm sayısı ilan edildi, Johnson&Johnson aşısı bir kaç ülkede kullanılmaya başlandı, (6) ABD 500 milyon doz aşı bağışladı, (7) Delta varyantı, Hint varyantı haberleri yayılmaya başladı, (8) Aşı eşitsizliği küresel ekonomik toparlanmayı baltalıyor, (9) Alfa, Beta, Gama ve Delta varyantlarının fenotipleri açıklanıyor, (10) 40 yaş üzeri için 3.doz aşı önerilmeye başlanıyor. Bunlar, Aralık 2020'den Ekim 2021'e kadar olan aylarda toplanan ve elde edilen en son haberlerin güncellemeleridir. Tweet sayısında birçok iniş ve çıkış vardır, varyantların gündeme geldiği 29 Haziran tarihinde zirve yapan tweet sayıları

başlangıca göre giderek artış gösterse de Ekim 2021 ayına kadar ortalama 1500 civarında seyretmektedir.

```
import datetime

b_date_count = df.groupby(by='date').count().reset_index()

b_date_count = b_date_count.rename(columns={'id':'Tweets Per Day'})

fig = ex.line(b_date_count,x='date',y='Tweets Per Day')


 fig.add_shape(type="line", x0=b_date_count['date'].values[0], y0=b_date_count['Negative
 Sentiment'].mean(), x1=b_date_count['date'].values[-
1], y1=b_date_count['Negative Sentiment'].mean(),

    line=dict(color="Red", width=2, dash="dashdot",), name='Mean',)


 fig.update_traces(mode="markers+lines")

fig.update_layout(hovermode= 'x')


#annots

b_date_count.date = pd.to_datetime(b_date_count.date)

b_date_count_dt = b_date_count.set_index('date')


fig.add_annotation(x=datetime.datetime(2021,2,19), y=b_date_count_dt.loc[pd.Timestamp('2
021-02-19'),'year'],

            text=r"Israil'in yaptigi arastirmaya gore Pfizer asisi ilk dozdan sonra 85%
bulundu",

            showarrow=True, arrowhead=3, yshift=5,bordercolor="#c7c7c7")


fig.add_annotation(x=datetime.datetime(2021,1,29), y=b_date_count_dt.loc[pd.Timestamp('2
021-01-29'),'year'],

            text=r"Ingiltere'de ortaya cikan varyanta karsi etkili asi bulundu",

            showarrow=True, arrowhead=3, yshift=5,ay=-160,bordercolor="#c7c7c7")


fig.add_annotation(x=datetime.datetime(2021,1,8), y=b_date_count_dt.loc[pd.Timestamp('20
21-01-8'),'year'],

            text=r"Komisyon BioNTech-Pfizer asisinin 300 milyon ek doz alimini onerdi",

            showarrow=True, arrowhead=3, yshift=5,ay=-30,bordercolor="#c7c7c7")
```

```
fig.add_annotation(x=datetime.datetime(2021,1,20), y=b_date_count_dt.loc[pd.Timestamp('2021-01-20'),'year'],
            text=r"Joe Biden Baskanligi basladi",
            showarrow=True, arrowhead=3, yshift=3,ay=120,bordercolor="#c7c7c7")

fig.add_annotation(x=datetime.datetime(2021,4,21), y=b_date_count_dt.loc[pd.Timestamp('2021-04-21'),'year'],
            text=r"Turkiye'de rekor olum sayisi ilan edildi, Johnson&Johnson bir kac ulkede kullanilmaya baslandi",
            showarrow=True, arrowhead=3, yshift=9,ay=200,bordercolor="#c7c7c7")

fig.add_annotation(x=datetime.datetime(2021,6,9), y=b_date_count_dt.loc[pd.Timestamp('2021-06-09'),'year'],
            text=r"ABD 500 milyon doz asi bagisliyor",
            showarrow=True, arrowhead=3, yshift=3,ay=120,bordercolor="#c7c7c7")

fig.add_annotation(x=datetime.datetime(2021,6,29), y=b_date_count_dt.loc[pd.Timestamp('2021-06-29'),'year'],
            text=r"Delta varyanti, Hint varyanti haberleri yayiliyor",
            showarrow=True, arrowhead=3, yshift=3,ay=120,bordercolor="#c7c7c7")

fig.add_annotation(x=datetime.datetime(2021,7,22), y=b_date_count_dt.loc[pd.Timestamp('2021-07-22'),'year'],
            text=r"Asi esitsizligi kuresel ekonomik toparlanmayi baltaliyor",
            showarrow=True, arrowhead=3, yshift=3,ay=110,bordercolor="#c7c7c7")

fig.add_annotation(x=datetime.datetime(2021,8,22), y=b_date_count_dt.loc[pd.Timestamp('2021-08-22'),'year'],
            text=r"Alfa, Beta, Gama ve Delta varyantlarinin fenotipleri aciklaniyor",
            showarrow=True, arrowhead=3, yshift=3,ay=130,bordercolor="#c7c7c7")

fig.add_annotation(x=datetime.datetime(2021,10,21), y=b_date_count_dt.loc[pd.Timestamp('2021-10-21'),'year'],
            text=r"40 yas uzeri icin 3.asi onerilmeye baslaniyor",
            showarrow=True, arrowhead=3, yshift=7,ay=220,bordercolor="#c7c7c7")

fig.update_layout(title='<b>Daily Tweet Count<b>',width=1200)
fig.show(renderer="colab")
```

**Şekil 5.8:** Belirli Tarihlerdeki Olaylarla Günlük Eğilim Analizi




### 5.1.9. Lokasyon Bazlı Duyguların Analizi Kod ve Sonuçları

Şekil 5.9.'da görüldüğü üzere en pozitif mesajların gönderildiği yerler Hindistan, New Delhi, Amerika Birleşik Devletleri, Londra şeklinde devam ederken, en negatif mesajların gönderildiği yerler ise Toronto, Hindistan, Türkiye, Amerika Birleşik Devletleri olarak gözlemlenmektedir.

```
fig = make_subplots(rows=2, cols=1,shared_xaxes=True,subplot_titles=('Top 10 Most Positive Locations Contributes',  'Top 10 Most Negative Locations Contributes'))


fig.add_trace(
    go.Bar(x=Most_Positive.user_location.value_counts()[:10].index, y=Most_Positive.user_location.value_counts()[:10].values,name='Number Of Tweets'),
    row=1, col=1
)


fig.add_trace(
    go.Bar(x=Most_Negative.user_location.value_counts()[:10].index, y=Most_Negative.user_location.value_counts()[:10].values,name='Number Of Tweets'),
    row=2, col=1
)


fig.update_layout(height=700, width=900, title_text="Locations That Contribute The Most To Our Cut-Offs")
fig.show(renderer="colab")
```

**Şekil 5.9:** En Çok Pozitif ve Negatif Tweet Gönderilen Ülkelerin Sıralaması


## 5.2. BERT Analizi Kod ve Sonuçları

BERT algoritması, Colab platformunda VADER algoritmasının çalıştırıldığı aynı bilgisayarda farklı varyasyonlarla öğrenme veri kümesi ve 212 bin tweet içeren esas veri kümesi azaltılarak kontrollü olarak defalarca çalıştırılmıştır. Ancak Intel Core i5*9300H CPU @2.40GHz işlemciye, 16 GB RAM'e, 500 GB harddiske, Intel UHD Graphics 630 entegre ekran kartına sahip bilgisayarda bu algoritma ile sonuçlar başarısız olmuştur.

40 bin tweet içeren bir eğitim veri setiyle eğitim aşaması ortalama 8 saat sürdüğü görülmüştür. Colab ile bu bilgisayarda 40 bin tweet ile eğitim birden fazla seferde yarıda kesilmiş ve yeniden başlatılmak zorunda kalmıştır. Colab çalışma süresi 8 saatten fazla izin vermediği için eğitim aşamasından öteye gidilememiştir. İnce ayar ve sınıflandırma aşamalarına geçebilmek için eğitim veri seti aşama aşama yüzde elli azaltılarak 10 bin tweet içeren bir eğitim veri setine kadar indirgenmiştir. Şekil 5.10.1. 'de görüleceği üzere her eğitim setinde %2.2'lik bir batch veri ile eğitime devam ettiği görülmüştür. Her batch için yaklaşık 20 dakika eğitim süresi gözlemlenmiştir.

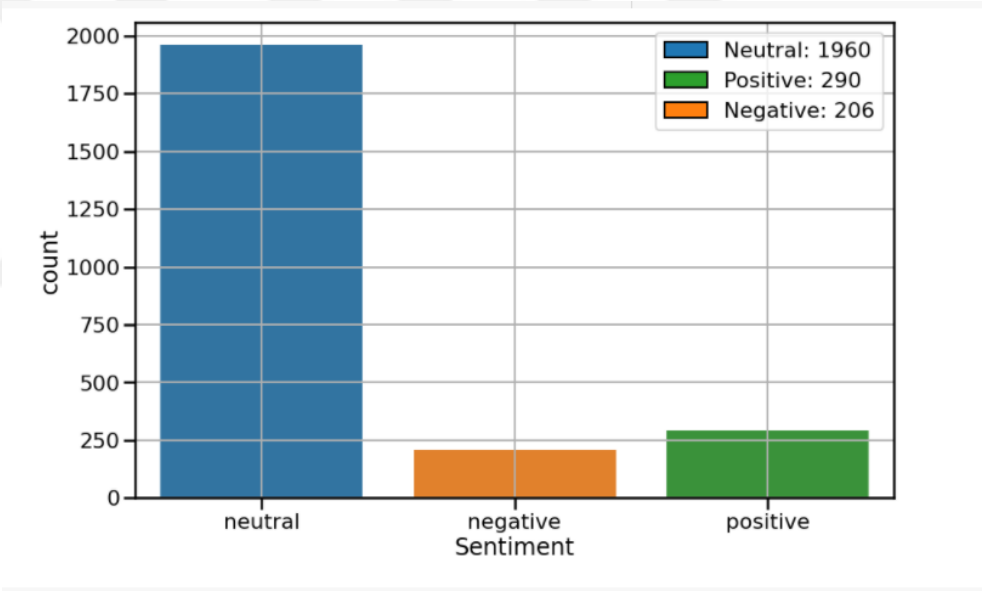
2 bin tweet ile eğitim aşaması 2 saat 10 dakikalık bir çalışma atlatılmıştır. Tahminleme aşamasında Aralık 2020 ve Ekim 2021 dönemindeki 212 bin tweeti içeren veri seti kullanılarak başlanan denemelerden de sınıflandırma aşamasında sonuç alınamadığı için kontrollü deneylerle veri setinde azaltmaya gidilerek yenilenmiştir. Sırasıyla 212 bin tweet, 150 bin tweet, 60 bin tweet, 30 bin tweet, 10 bin tweet ve son olarak 3109 tweet ile denemeler yapılmıştır. 3 saat süren çalışma sonunda 3109 tweet için sınıflandırma gerçekleşmiştir.

**Şekil 5.10.1:** Eğitim Sonuçları

```
print("")
print("Training complete!")

print("Total training took {:} (h:mm:ss)".format(format_time(time.time()-total_t0)))
```

```
======== Epoch 1 / 1 ========
Training...
  Batch    40  of    221.    Elapsed: 0:22:31.
  Batch    80  of    221.    Elapsed: 0:45:44.
  Batch   120  of    221.    Elapsed: 1:08:43.
  Batch   160  of    221.    Elapsed: 1:31:26.
  Batch   200  of    221.    Elapsed: 1:54:11.

  Average training loss: 0.78
  Training epcoh took: 2:05:48

Running Validation...
  Accuracy: 0.75
  Validation Loss: 0.61
  Validation took: 0:04:29

Training complete!
Total training took 2:10:17 (h:mm:ss)
```

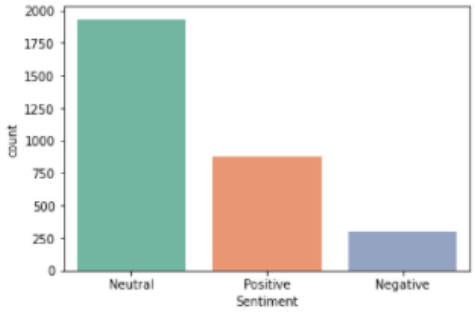
**Şekil 5.10.2:** BERT 3109 Sınıflandırma Sonuçları



**Şekil 5.10.3:** VADER 3109 Sınıflandırma Sonuçları

```
[ ]  import nltk
     nltk.download('vader_lexicon')

     sentiment = SentimentIntensityAnalyzer()
     def get_sentiment(data):
         sentiment_list = []
         for text in list(data['Tidy Tweet'].values):
             if sentiment.polarity_scores(text)["compound"] > 0:
                 sentiment_list.append("Positive")
             elif sentiment.polarity_scores(text)["compound"] < 0:
                 sentiment_list.append("Negative")
             else:
                 sentiment_list.append("Neutral")
         return sentiment_list

     data['Sentiment'] = get_sentiment(data)
     sns.countplot(x="Sentiment", data=data, palette="Set2")
     print(data.Sentiment.value_counts())
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
Neutral     1933
Positive     876
Negative     300
Name: Sentiment, dtype: int64
```



# ALTINCI BÖLÜM

# PERFORMANS ANALİZ SONUÇLARI

## 6.1. Performans Analiz Sonuçları

Çeşitli makine öğrenimi yaklaşımları ile sözlük tabanlı yaklaşımlar arasında doğrudan bir karşılaştırma olarak yapılması amaçlandı ve literatür incelememizin çoğunda gördük ki sözlük tabanlı yöntemlerin kullanımı yaygındır. Çoklu araştırma makalelerinde literatür incelemesinin sonuçları göz önüne alındığında, VADER duygu analizcisi sözlük tabanlı belirli bir yaklaşım olarak seçilmiştir. Şimdiye kadar ki literatüre bakıldığında, duygu analizini gerçekleştirmek için en uygun yöntemin VADER algoritması olduğu sonucuna varılmıştır.

Bir makine öğrenimi yaklaşımı kullanmak için gerekli eğitim verilerinin sınıflandırıldığından bahsedilen literatür inceleme sonuçlarından yola çıkarak, bu verileri şekillendirmek ve bilinmeyen veri sınıflandırmasını tahmin etmek için bu veriler üzerinde bir algoritma eğitilmesi gerektiği görülmüştür. Makine öğrenimi algoritmalarının araştırılması ve test edilmesi, sonuçların daha anlamlı yorumlar üretmesinin daha maliyetli olduğu ortaya çıkmıştır. Bu nedenle, makine öğrenme yöntemi yerine VADER yaklaşımının seçilmesi daha düşük maliyetli sonuçlar üretecektir.

Sözlük tabanlı algoritmaların üretmiş olduğu sonuçların güvenilirlik oranları düşük olmaktadır. Adından da anlaşılacağı üzere sözcük ya da sözlük tabanlı algoritmaların temelinde kelimelerin anlamına puanlama yapılarak pozitif ya da negatif görüş olarak sınıflandırma bulunmaktadır. Ancak tweet metinlerinin içerisinde mecazlar, kinayeler, tersine manalar gibi insanların günlük hayatta kullandığı anlamsal çeşitlilikleri anlayamamaktadır. Google BERT algoritmasının ise bu noktada önceden verilen bilgiyle kendisinin eğiterek daha sonra cümlelerdeki kelimelerin yerlerini değiştirerek daha güvenilir çözümler ürettiği iddia edilmektedir.

Otokorelasyon analizi ve duyguların ayrıştırılması, eğilimleri, düzeyi, mevsimselliği ve artıkları belirlemeye adanmış sonuçları göstermek, olumlu ve olumsuz duyguların mevsimsel kalıplarını izlemek ve ayrıca modeldeki gecikmeleri düzeltmek için gerçekleştirilmiştir. Son olarak, otokorelasyon analizi sonuçlarından, verilerde herhangi bir gecikme olmadığını doğrulayan %95'lik bir güven aralığı olduğundan, verilerimizde herhangi bir gecikme olmadığı sonucuna varılmıştır.

Bu konudaki sonuçlar, değerleri gösteren bir grafik şeklinde gösterilir. Son olarak, 2020'den 2021'e kadar on aylık Twitter verilerinde gösterilen günlük tweet'leri gösteren Şekil 5.8. belirli tarihlerle ilişkili olaylarla günlük trend analizini tamamladıktan

sonra gösterilmiştir. Süreç, on aylık süre içinde günlük tweet'ler ve o aylardaki haber ve duyurular alınarak gerçekleştirilmiştir.

### 6.1.1. Çalışmanın Katkıları

Çalışmada kullanılan bilgisayarın donanım yetenekleri, performansı ve zaman baskısı nedeniyle süre kısıtları altında birkaç detay ortaya çıkmıştır.

Tahmin edilen duyarlılıklar ve ayrıca büyük veri kümeleri kullanılarak zaman serisi analizi için daha fazla bölüm düşünülebilir. Bu çalışma sadece COVID-19 aşısı ile ilgili verilerle sınırlıdır. Bu modeller ve algoritmalar benzer türdeki durumlarda belirli toplulukların duygu tespitlerinin ortaya çıkarılması için kullanılabilir. Zaman serisi analizi aylık olarak sınıflandırılırken, gelecekte bu daha verimli sonuçlar elde etmek için saatlik olarak sınıflandırılabilir. Bu çalışmada duygu analizi sadece İngilizce dilindeki veriler için uygulanmıştır. Gelecekteki uygulamalar için duygu analizi diğer dillerde de yapılabilir. Gelecekteki uygulamalar için BERT algoritması kullanılmak isteniyorsa GPU işlemciye sahip, daha yüksek donanımlı bilgisayarlar tercih edilebilir.

Ek olarak, BERT algoritmasının yanısıra, 2019 yılında BERT mimarisinin eğitim ve sonuçlarını iyileştirmek üzere geliştirlen ALBERT mimarisi kullanılarak da çalışma değiştirilebilir. ALBERT modeli, karşılık gelen BERT modellerine kıyasla daha küçük bir parametre boyutuna sahiptir. Örneğin, baseBERT, baseALBERT'ten 9 kat daha fazla parametreye sahiptir ve largeBERT, largeALBERT'ten 18 kat daha fazla parametreye sahiptir.

# YEDİNCİ BÖLÜM

# SONUÇ

## 7.1. Sonuç

“Bu çalışmada, COVID-19 aşısı ile ilgili “PfizerBioNTech, Sinopharm, Sinovac, Moderna, oxfordastrazeneca, Covaxin, SputnikV” arama filtre metinleriyle milyonlarca tweet arasından hedef etiketlerin kullanıldığı mesajlar indirilerek, twitter metinlerinin duygu, görüş tespitlerini yapabilmek için sözcük odaklı bir algoritma olan VADER ve makine öğrenimi algoritması olan Google BERT modeli çalıştırılarak elde edilen çıktılar paylaşılmıştır. COVID-19 aşısı ile ilgili metin duygu analizine dair uygun yaklaşımı belirlemek için sistematik bir literatür taraması yapılmıştır. VADER algoritmasının duygu analizi için sözlük tabanlı algoritmalar arasında hem polarite hem yoğunluk değerlendirmesi sebebiyle seçkin olması, tweetleri çoklu sınıflandırma sistemine göre kategorize edebilmesi nedeniyle NLTK ve VADER algoritması Twitter verilerinin duygu analizini yapmak için seçilmiştir. Duyarlılık değerlerine göre sonuçlar, her bir duyguya ilişkin KDE dağılımının olumlu, olumsuz ve nötr olduğunu göstermiştir. Bu çalışmadan yola çıkarak insanların duygularını sosyal medyada, özellikle de Twitter'da paylaşmalarının, gösterdikleri tepkilerinin günden güne değiştiği söylenebilmektedir. COVID-19 aşısı haberlerinin yayılmasıyla birlikte aşı ile ilgili bu veriler bize insanların, devlet kurumlarının ve sosyal medya kuruluşlarının durumları nasıl yayınladığını göstermektedir.

Zaman serisi analizine gelince, sonuçları bir bakış açısıyla ele alırsak, her bölümün zaman çizelgesi üzerinde günlük duyguların dağılımını gerçekleştirdikten sonra, standart sapma ve ortalama değerleri hesaplayarak bazı gecikmeler ve eğilimler bulduğumuz sonucuna varılmıştır. Verilerde bulunan gecikmeleri düzeltmek için otokorelasyon analizi yapılmıştır ve ayrıca duyguları ayrıştırarak eğilimleri, düzeyi, mevsimselliği ve artıkları bulmak mümkündür. Belirli bir tarihle ilişkili olaylarla günlük trend analizi, verilerimizin belirli tarihlerdeki haberlerle ilişkisine bakarak daha anlamlı sonuçlar göstermiştir. Bu nedenle, zaman serisi analizinin, verileri kolayca bölümlere ayırarak ve etkinleştirerek önemli sonuçlar gösterdiği çıkarımını yapmamız doğru olacaktır.

Yapılan çalışmada sözlük tabanlı algoritma kullanılarak gerçekleştirilen duygu analizinde, pozitif metinlerin miktarı, negatif metinlerin miktarının iki katı kadar gözlenmiştir. Ortaya çıkan bu sonuçtan yola çıkarak koronavirüs hakkında kamuoyunun genel duygu düşünce durumunun olumlu olduğu yorumu yapılabilir. Koronavirüs

pandemisinin vaka oranlarının tekrar artmaya başladığı son aylarda dahi olumlu görüşlerin yüksek olması, halkın bilinç düzeyinin arttığı, farkındalık oranının yükseldiği, virüsle mücadele sürecinde, belirli koşullar altında normal hayatı sürmeye devam ettiği, toplumun duygu durumunun iyi yönde ilerlediği yorumu çıkarılabilir. Aynı koşullar altında deneyleri yapılan VADER ve BERT algoritmasının karşılaştırılması ile herhangi bir anlamlı sonuç üretilemeyen BERT algoritmasının maliyetinin VADER algoritmasına kıyasla yüksek olduğu ortaya konmuş ancak ortaya çıkan sonuçların güvenilirlikleri açısından herhangi bir yorum ortaya konamamıştır. Bu nedenle eğitim veri seti ve ana veri seti küçültülerek her iki yöntemin de kıyaslanabilmesi için ikinci bir çalışma yapılmıştır.

İlk çalışma sonucundaki veriler incelendiğinde, koronavirüs hakkında indirilen 212.982 tweet mesajının, 55.833 tanesi olumlu çıkarken, 26.262 tanesi olumsuz çıkmakta ve 130.887 tanesi nötr olarak sonuçlanmış olmaktadır.

İkinci çalışma sonucundaki veriler incelendiğinde, koronavirüs hakkında indirilen 3109 tweet mesajının, VADER algoritması ile 876 tanesi olumlu çıkarken, 300 tanesi olumsuz çıkmakta ve 1933 tanesi nötr olarak sonuçlanmış olmaktadır. BERT algoritması ile 290 tanesi olumlu çıkarken, 206 tanesi olumsuz çıkmakta ve 1960 tanesi nötr olarak sonuçlanmış olmaktadır, yanısıra BERT algoritmasının 653 tane tweet mesajını değerlendirmeye almadığı gözlemlenmiştir. Bu miktar veri setinin yüzde yirmibirini oluşturmaktadır ve önemli bir örnek büyüklüğüdür. Burdaki sapma ile olumsuz görüşlerin olumlu görüşlere oranı da yüzde otuzyedi oranında artmış olarak gözlemlenmektedir.

Bu çalışma kapsamında VADER ve BERT modelinin çalıştırıldığı cihazın donanım özelliklerinin yüksek hızlar sunabilecek olması, veri setinde kullanılan etiketlerin değiştirilmesi, veri setinin genişletilmesi, çalışma için ayrılan sürenin artırılması bu çalışmadan alınabilecek sonuçların iyileştirilmesini sağlayacaktır.

# KAYNAKÇA

Anderson, T. K. (2009). Kernel density estimation and k-means clustering to profile road accident hotspots. *Accident Analysis and Prevention, 41*(3), 359–364

Bird, S., Klein, E. & Loper, E. (2009). *Natural language processing with python*. California: O'Reilly.

Bidirectional Encoder Representations from Transformers. (2021, 30 Kasım). Erişim adresi : https://www.geeksforgeeks.org/understanding-bert-nlp/

Bert State. (2021, 30 Kasım). Erişim adresi : https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

Colab. (2021, 30 Kasım). Erişim adresi : https://colab.research.google.com/notebooks/welcome.ipynb

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019, 2-7 Haziran). *BERT: Pre-training of deep bidirectional transformers for language understanding* [Sözlü Bildiri]. The 17th North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics. Minnesota

Dünya Sağlık Örgütü. (2021, 30 Kasım). Erişim adresi : https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

Luan, J., & Terrence, W. (2000). *Data mining and knowledge management: a system analysis for establishing a tiered knowledge management model*. California: Cabrillo College CA.

Gabriel Preda. (2021, 30 Kasım). Erişim adresi: Kaggle, https://www.kaggle.com/gpreda/all-covid19-vaccines-tweets

Github Gabriel Preda. (2021, 30 Kasım). Erişim adresi : https://github.com/gabrielpreda/Kaggle/blob/master/vaccination_tweets/vaccination-all-tweets.py

Glymour, C., Madigan, D., Pregibon, D. & Smyth, P. (1997). Statistical themes and lessons for data mining, *Data Mining and Knowledge Discov*ery, *1*(1), 11-28. https://doi.org/10.1023/A:1009773905005.

Gustafsson, M. & Davidsson, M. (2020). *Sentiment analysis for tweets in Swedish: Using a sentiment lexicon with syntactic rules* (Unpublished Bsc Dissertation). Linnaeus University, Vaxjo, Sweden.

HuggingFace. (2021, 30 Kasım). Erişim adresi : https://huggingface.co/models

Hutto, C., Gilbert, E. (2014, 1-4 Haziran). *Vader: A parsimonious rule-based model for sentiment analysis of social media text* [Sözlü Bildiri]. The International AAAI Conference on Web and Social Media. Association for the Advancement of Artificial Intelligence, Michigan. https://ojs.aaai.org/index.php/ICWSM/article/view/14550

Hyndman, R. J. & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). Australia: Monash University.

Kent, A. (2000). *Encyclopedia of library and information science* (2nd Ed.). New York: Marcel Dekker Inc.

Maimon, O. & Rokach, L. (2002). *Data mining and knowledge discovery handbook* (2nd ed.). New York: Springer.

Krisp, J. M., Peters S., Murphy C. E. & Fan, H. (2009). *Visual bandwidth selection for kernel density maps*. Photogrammetrie Fernerkundung Geoinformation. *5*(1). 445–454. doi: 10.1127/1432-8364/2009/0032

Krisp, J. M. & Spatenkova, O. (2009, 19-22 Ocak). *Kernel density estimations and their application in visualizing mission density for fire & rescue services* [Sözlü Bildiri]. Cartography and Geoinformatics for Early Warning and Emergency Management: Towards Better Solutions, Prague.

Matplotlib. (2021, 30 Kasım). Erişim adresi : https://matplotlib.org/

Numpy. (2021, 30 Kasım). Erişim adresi : https://numpy.org/

Our World In Data. (2021, 30 Kasım). Erişim adresi : https://ourworldindata.org/coronavirus

Pandas. (2021, 30 Kasım). Erişim adresi : https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html

Paolo, G. (2003). *Applied data mining statistical methods for business and industry*. West Sussex. Willey.

Sayad, S. (2010). *An introduction to data mining*. Toronto: University of Toronto.

Scikit-Learn. (2021, 30 Kasım). Erişim adresi : https://scikit-learn.org/

Shumway, R. H. & Stoffer, D. S. (2005). *Time series analysis and its applications with R examples* (2nd ed.). New York: Springer.

Siebert, J., Groß, J. & Schroth, C. (2021). A systematic review of python packages for time series analysis. *Engineering Proceedings*, *5*(1), 22-34. https://doi.org/10.3390/engproc2021005022

Silverman, B. W. (1986). *Density estimation for statistics and data analysis, monographs on statistics and applied probability*. London: Chapman and Hall.

Elbagir, S. & Yang, J. (2019, 13-15 Mart). *Twitter sentiment analysis using natural language toolkit and VADER sentiment* [Sözlü Bildiri]. The International Multi Conference of Engineers and Computer Scientists, International Association of Engineers, Hong Kong.

Lansley, G., de Smith, M., Goodchild, M., Longley, P. (2018). *Geospatial analysis: A comprehensive guide to principles, techniques and software tools* (6th ed.). Edinburgh: The Winchelsea Press.

Spencer, J., & Uchyigit, G. (2012, 26-27 Ocak). *Sentimentor: Sentiment analysis of twitter data* [Sözlü Bildiri]. Central Europe Workshop, Bari, İtalya.

Symeonidis, S., Effrosynidis, D. ve Arampatzis, A. (2018). A comparative evaluation of pre-processing techniques and their ınteractions for twitter sentiment analysis. *Expert Systems With Applications, 110,* 298-310.

Statista. (2021, 30 Kasım). Erişim adresi : https://www.statista.com/statistics/346167/facebook-global-dau/

Statista. (2021, 30 Kasım). Erişim adresi : https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/

Tensorflow. (2021, 30 Kasım). Erişim adresi : https://www.tensorflow.org/learn

Towardsdatascience. (2021, 30 Kasım). Erişim adresi : https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Towardsdatascience. (2021, 30 Kasım). Erişim adresi : https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

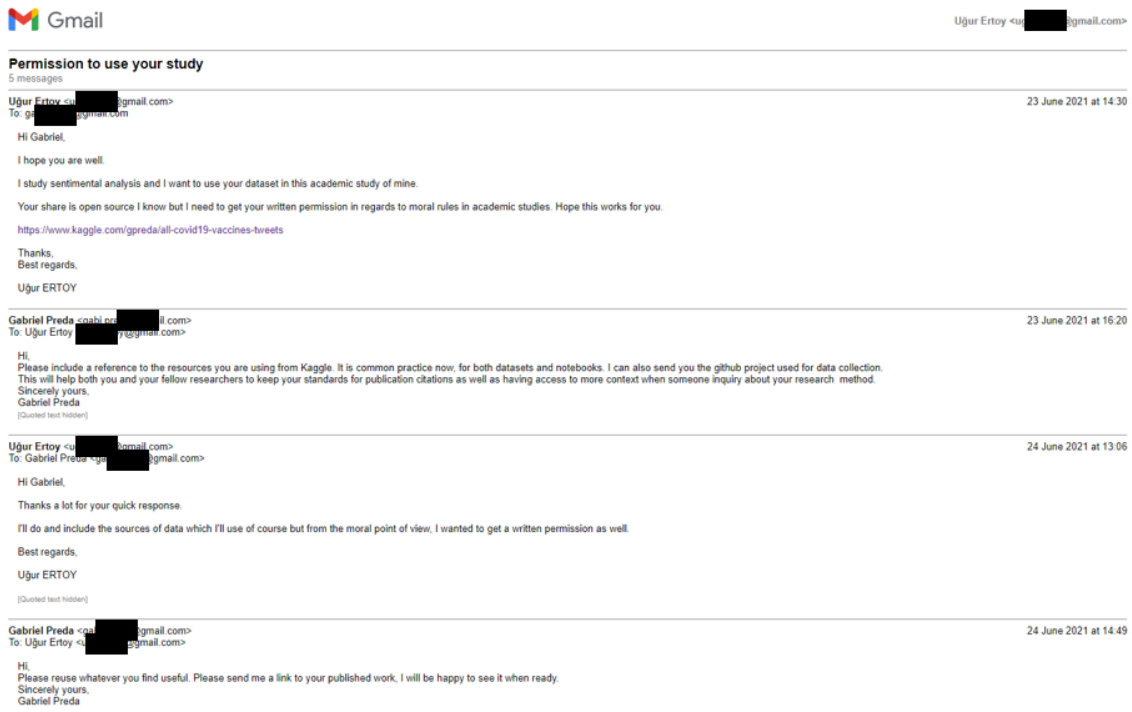
Towardsdatascience. (2021, 30 Kasım). Erişim adresi : Erişim adresi : https://towardsdatascience.com/time-series-analysis-in-python-an-introduction-70d5a5b1d52a

Torch. (2021, 30 Kasım). Erişim adresi : http://torch.ch/

Wagh, B., Shinde, J. & Kale, P.A. (2018). A Twitter sentiment analysis using NLTK and machine learning techniques. *International Journal of Emerging Research in Management and Technology*, *6*(12), 37-44. doi: 10.23956/ijermt.v6i12.32.

Jianqiang, Z. & Xiaolin, G. (2017). Comparison research on text pre-processing metods on Twitter sentiment analysis. *IEEE Access*, *5*(1), 2870-2879. doi: 10.1109/ACCESS.2017.2672677

# EK-1 ÇALIŞMAYA AİT VERİ KULLANIM İZNİ



Gmail

Uğur Ertoy <u[redacted]gmail.com>

**Permission to use your study**

5 messages

**Uğur Ertoy** <u[redacted]gmail.com> 23 June 2021 at 14:30
To: g[redacted]gmail.com

Hi Gabriel,

I hope you are well.

I study sentimental analysis and I want to use your dataset in this academic study of mine.

Your share is open source I know but I need to get your written permission in regards to moral rules in academic studies. Hope this works for you.

https://www.kaggle.com/gpreda/all-covid19-vaccines-tweets

Thanks,
Best regards,

Uğur ERTOY

**Gabriel Preda** <gabi.pr[redacted]il.com> 23 June 2021 at 16:20
To: Uğur Ertoy [redacted]gmail.com>

Hi,
Please include a reference to the resources you are using from Kaggle. It is common practice now, for both datasets and notebooks. I can also send you the github project used for data collection.
This will help both you and your fellow researchers to keep your standards for publication citations as well as having access to more context when someone inquiry about your research method.
Sincerely yours,
Gabriel Preda
[Quoted text hidden]

**Uğur Ertoy** <u[redacted]gmail.com> 24 June 2021 at 13:06
To: Gabriel Preda <g[redacted]gmail.com>

Hi Gabriel,

Thanks a lot for your quick response.

I'll do and include the sources of data which I'll use of course but from the moral point of view, I wanted to get a written permission as well.

Best regards,

Uğur ERTOY

[Quoted text hidden]

**Gabriel Preda** <ga[redacted]gmail.com> 24 June 2021 at 14:49
To: Uğur Ertoy <u[redacted]gmail.com>

Hi,
Please reuse whatever you find useful. Please send me a link to your published work, I will be happy to see it when ready.
Sincerely yours,
Gabriel Preda

# ÖZGEÇMİŞ

## Kişisel Bilgiler

Adı Soyadı : Uğur Ertoy

## Eğitim

| Derece | Eğitim Birimi |
|---|---|
| Lisans | Endüstri Mühendisliği, 2011 |
| Lise | Eskişehir Yunus Emre YDAL, 2007 |

## Profesyonel Tecrübe

| | |
|---|---|
| Buyer, McLaren Automotive Ltd., London, UK | Şubat, 2020 – ... |
| Senior Buyer, Borusan Mannessman Boru, Bursa, Turkey | Kasım, 2019 – Ocak, 2020 |
| Commodity Buyer, Vaillant Group, Istanbul, Turkey | Ocak, 2015 – Kasım, 2019 |
| Technical Buyer, Vaillant Group, Istanbul, Turkey | Ağustos, 2012 – Ocak, 2015 |
| Liutenant, Turkish Army, Cyprus | Ağustos, 2011 – Temmuz, 2012 |

## Yabancı Dil

İngilizce