

**EGE ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY ZEKA YÖNTEMLERİ**

**2021-2022 BAHAR YARIYILI**

**PROJE-2**

**(NLP & Sentiment Analysis)**

**TESLİM TARİHİ**

19/06/2022

**HAZIRLAYANLAR**

05190000025, Ersin ELMAS

05180000120, Cemil CESUR

İçindekiler

[Problem Tanımı 3](#_Toc106497148)

[Araştırma 3](#_Toc106497149)

[Kullanılan Ortam ve Kütüphaneler 4](#_Toc106497150)

[Kullanılan Yöntemler 4](#_Toc106497151)

[Önerilen Yöntemler 8](#_Toc106497152)

[Deneysel Çalışmalar 8](#_Toc106497153)

[Sonuç 10](#_Toc106497154)

[Ek 1: 10](#_Toc106497155)

[Ek 2: 11](#_Toc106497156)

[Ek 3: 11](#_Toc106497157)

[EK 4: 12](#_Toc106497158)

[Kaynakça 13](#_Toc106497159)

[Özdeğerlendirme Tablosu 14](#_Toc106497160)

[İş Bölümü 15](#_Toc106497161)

# Problem Tanımı

Bu proje kapsamında “artificial intelligence”, “deep learning”, “machine learning” ve “natural language processing” konularının insanlar tarafından nasıl değerlendirildiğini ve insanların duygularına nasıl yansıdığını çeşitli analizlerle gözlemlemek amaçlanmıştır. Bu analizleri gerçekleştirebilmek için Doğal Dil İşleme (Natural Language Processing) ve Duygu Analizi (Sentiment Analysis) yöntemleri kullanılmıştır. Duygu Analizini gerçekleştirmek için Twitter’dan paylaşımlar toplanmıştır ve bunların üzerinden duygu analizi çalışmaları yapılmıştır.

# Araştırma

Duygu analizi temel olarak bir metin analizi olup verilen metnin duygusal olarak ifade etmek istediği sınıfı (pozitif, negatif ve nötr) belirlemeyi amaçlar. Kelimeleri analiz etmek ve gerçek anlamlarını çıkarmak için duygu ve bağlamı dikkate alarak tanımlanmış örüntüler ve temalar sonucu duygu analizini verir. Doğal Dil İşlemede amaç veri kümesi içerisinden ve dilin barındırdığı özelliklerden faydalanılarak bağlamları doğru yakalayabilen ve genelleyici olduğu kabul edilen bir istatistiksel ya da olasılıksal model ortaya koymaktır. Bu sayede veri seti içinden dile ve göreve dair tüm dinamikler sistem kendi kendine öğrenecektir. Bu proje için kullanabileceğimiz veri setinin oluşturulması aşamasında Twitter’dan veri çekmek için Twint kullandık. Twint, Twitter'ın API'sini kullanmadan ve geliştirici hesabı almaya gerek olmadan tweetleri çekmemize olanak veren açık kaynak kodlu gelişmiş bir Twitter scraping aracıdır.

Twint, sağladığı parametrelerle sadece tweetleri değil bunun yanında veriyi daha iyi analiz etmeye yardımcı olabilecek lokasyon, zaman aralığı ve en popüler olan tweetin çekilmesi gibi benzer birçok özellik de sunar. Aynı zamanda Twitter API’lerinde olan 3200 tweet çekebilme sınırı twint’de yoktur.

# Kullanılan Ortam ve Kütüphaneler

Veri çekme ve veri seti oluşturmak için twint (2.1.21. sürümü) kütüphanesi; veri seti üzerindeki temel işlemler için pandas kütüphanesi; görselleştirme için seaborn, matplotlib kütüphaneleri; stopwords belirlerme işlemi için nltk kütüphanesi; veriyi temizlemek için string, re ve emoji kütüphaneleri; sentiment analiz için TextBlob, wordcloud oluşturmak için WordCloud kütüphaneleri kullanılmıştır. Projenin gerçekleştirilmesi aşamasında JupyterNotebook (6.4.8 sürümü) ortamı kullanılmıştır.

# Kullanılan Yöntemler

* Twint yardımı ile belirlediğimiz keyword’leri içeren tweetler çekildi. Ardından .csv dosyası oluşturup içerisine yazıldı.

c = twint.Config()

c.Search = "\"Machine Learning\" OR \"Artificial Intelligence\" OR \"Deep Learning\" OR \"Natural Language Processing\""

c.Limit = 10000

#c.Until = "2022-01-01"

#c.Near = "New York"

c.Store\_csv = True

c.Output = "./dataset.csv" # csv dosyasına çıktı olarak verilir

# Searching

twint.run.Search(c)

* Oluşturduğumuz csv dosyasından okuma yapıldı.

df=pd.read\_csv("dataset.csv") # dosyadan dataframe'e okuma yapıldı

* İngilizcede kullanılan önemsiz kelimeler nltk kütüphanesinden çekildi. Ardından metnin içerisinden url/hashtag/retweet gibi işlemlerin temizlemesi yapıldı.

stopwords = nltk.corpus.stopwords.words("english") # ingilizcedeki etkisiz kelimeleri aldık

stopwords

def remove\_noise(text):

text = re.sub('@[\w]+','',text)

text = re.sub('https:\/\/\S+', '', text) # url'leri kaldırma

text = re.sub('#', '', text) # hashtagleri kaldırma

text = re.sub('RT[\s]+', '', text) # retweetleri kaldırma

text = "".join([char for char in text if char not in string.punctuation]) #noktalama işaretlerini kaldırma

text = re.sub('[0-9]+', '', text)

text=re.sub(emoji.get\_emoji\_regexp(),r"",text) # emojileri kaldırma

text = text.lower()

text = re.sub('[‘’“”…]', '', text)

text = re.sub('\n', '', text)

text = re.sub('\[.\*?\]', '', text)

text = re.sub('[%s]' % re.escape(string.punctuation), '', text)

text = re.sub('\w\*\d\w\*', '', text)

return text

df['tweet\_cleaned'] = df['tweet'].apply(lambda x: remove\_noise(x))

# Tweet'in eski ve yeni hali

df[['tweet','tweet\_cleaned']]

df = df.dropna(axis=1) # Eksik değer içeren sütunları kaldırma

df.head()

* Textblob aracılığıyla sentiment analiz yapıldı.

df\_clean["Polarity"]=df\_clean["tweet\_cleaned"].apply(lambda word: TextBlob(word).sentiment.polarity)

df\_clean["Subjectivity"]=df\_clean["tweet\_cleaned"].apply(lambda word: TextBlob(word).sentiment.subjectivity)

df\_clean[["Polarity","Subjectivity"]].head(10)

* Polarity değerine göre değerlendirme işlemi yapıldı.

def polarity\_analyzer(polarity):

if polarity > 0:

return "Positive"

if polarity == 0:

return "Neutral"

if polarity < 0:

return "Negative"

* Sentiment analizine göre pasta grafik şeklinde negatif,pozitif ve nötr duygu durumları gösterildi.

plt.figure(figsize=(10,10))

df\_clean['Sentiment'].value\_counts().plot.pie(autopct='%1.1f%%', startangle=60)

plt.title('Sentiment Dağılım Grafiği')

plt.legend(loc='upper right')

* Belirlenen pozitif ve negatif kelimelere göre WordCloud gösterimi yapıldı.

text\_positive=df\_clean[df\_clean['Sentiment']=="Positive"]

text\_positive\_all = " ".join(review for review in text\_positive.tweet\_cleaned) # tek bir metin elde edildi

text\_negative=df\_clean[df\_clean['Sentiment']=="Negative"]

text\_negative\_all = " ".join(review for review in text\_negative.tweet\_cleaned) # tek bir metin elde edildi

tw\_mask = np.array(Image.open("twitter.png"))

wordcloud = WordCloud(contour\_width=0, contour\_color='Blue',

background\_color ='white', mask = tw\_mask , colormap = "Blues",

max\_words = 5000, normalize\_plurals = True)

wordcloud.generate(text\_positive\_all)

plt.figure(figsize = (20, 10))

plt.imshow(wordcloud)

plt.axis("off")

plt.tight\_layout(pad = 0)

plt.title('Positive Reviews')

plt.savefig("twitter.png", dpi=300)

plt.show()

wordcloud.to\_file("cloud1.png")

wordcloud.generate(text\_negative\_all)

plt.figure(figsize = (20, 10))

plt.imshow(wordcloud)

plt.axis("off")

plt.tight\_layout(pad = 0)

plt.title('Negative Reviews')

plt.savefig("twitter.png", dpi=300)

plt.show()

wordcloud.to\_file("cloud2.png")

# Önerilen Yöntemler

Sentiment analizinde kullanılan yöntem, ortamlara yukarıda yer alan “***Kullanılan Ortam, Yöntem ve Kütüphaneler”*** ve ***“Araştırma”*** bölümlerinde yer verilmiştir.

Word cloud oluşturma aşamasından sonra sırasıyla tokanization, lemmatization ve vectorization aşamalarını uygulayarak veriyi eğitim için modele hazır hale getirdik. Tokanization işleminde tweetler kelimelerine ayrılırmış, sonrasında lemma işlemi ile kelimelerin kökü bulunmuştur. Ancak bu aşamalardan sonra elde edilen veri hala kategori tipinde olduğu için modele vermek için uygun değildir. Temizlenmiş ve işlemden geçirilmiş verinin classification modellerine verilebilmesi için son olarak vektörize işleminden de geçmesi gerekir. Vectorization işleminde kategorik veriler vektörler şeklinde ifade edilerek sayısal değerlere dönüştürüldü ve sonrasında eğitim için modellere verildi. Bu modellerin eğitilmesi ve çeşitli deneysel çalışmalar ile başarı ölçümleri yapılmış ve sonuçlar elde edilmiştir.

# Deneysel Çalışmalar

Veri setini twint aracılığıyla Twitter’dan çekerek oluşturduk. Örnek sayısı 10000 olarak belirledik. Sınıflarımız pozitif,negatif ve nötr olmak üzere 3 tanedir.

Öznitelikler,

*'id', 'conversation\_id', 'created\_at', 'date', 'time', 'timezone',*

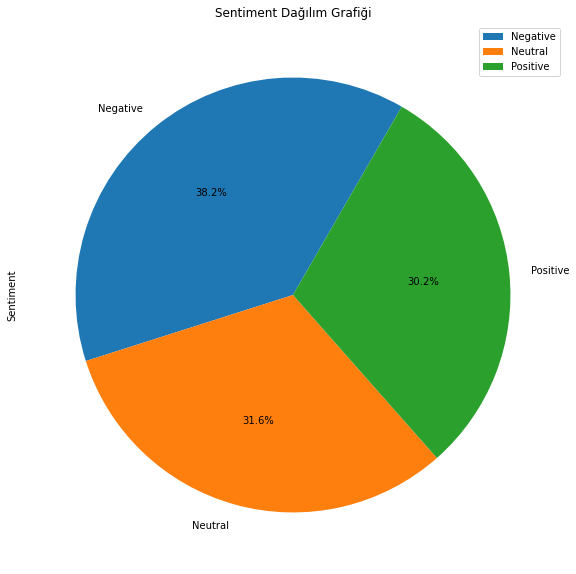
*'user\_id', 'username', 'name', 'tweet', 'language', 'mentions', 'urls',*

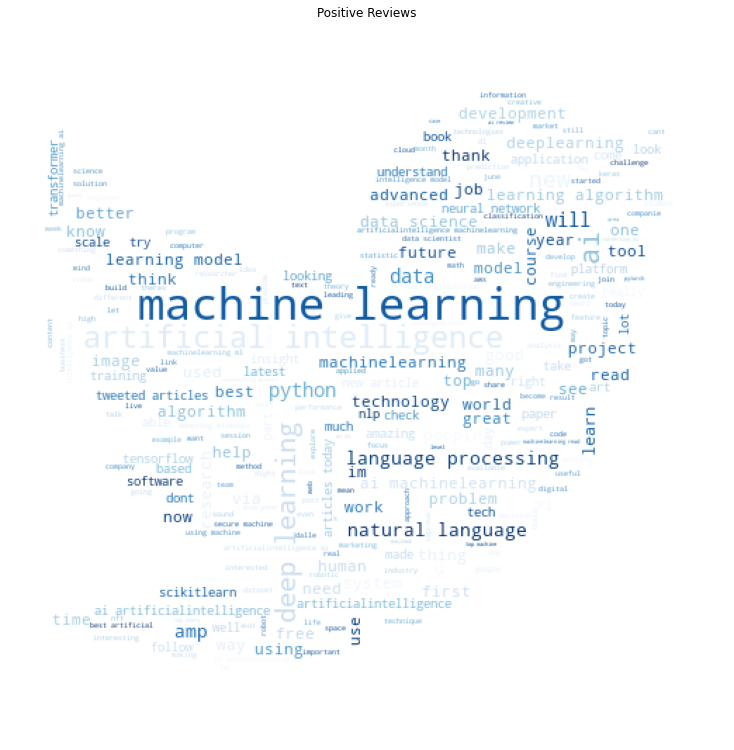
*'photos', 'replies\_count', 'retweets\_count', 'likes\_count', 'hashtags',*

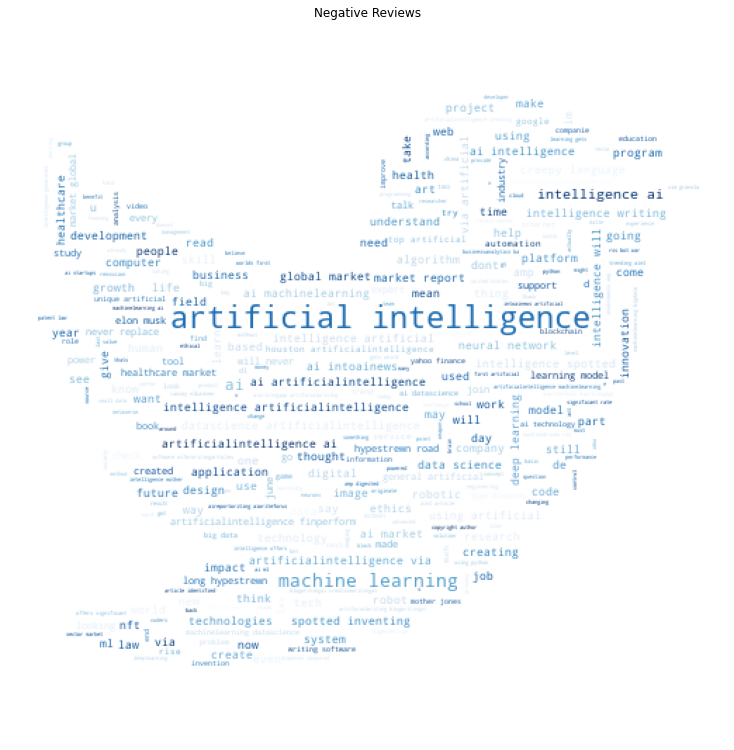
*'cashtags', 'link', 'retweet', 'video', 'reply\_to', 'tweet\_cleaned'*

olmak üzere 24 tanedir.

Sonuç Çıktıları:



****

****

**Kaynakça**

# Sonuç

Kullanılan modellerde elde edilen Precision, Accuracy, Recall ve F1 skorlarını incelediğimizde projede gerçekleştirilen sentiment analizin ve yapılan tahminlerin genel olarak başarılı olduğu gözlemlenmektedir. En iyi sonuç Random Forest Classifier tarafından sağlanmıştır. Proje gerçek hayatta kullanım için uygundur. Yapay zeka hakkındaki görüşler sentiment analiziyle tespit edilip twitter üzerinden görüş bildiren insanların büyük bir bölümümün bu konu hakkındaki duygularının tespit edilmesini sağlar. Ayrıca bu proje farklı konulara da uyarlanarak filmler, şirketler, ürünler hakkındaki yorumların değerlendirilmesinde ve gerekli analizlerin yapılmasında da kullanılabilir.

Bu projeyi geliştirirken NLP kullanılabilecek alanlar, gerçekleştirim için kullanılan yöntem, ortam ve kütüphaneler hakkında daha kapsamlı bilgiler edindik. Hazır bir veri seti kullanmadan veri çekerek kendi veri setimizi nasıl oluşturabileceğimizi ve bu veriyi nasıl kullanılabilir hale getirebileceğimizi öğrendik.

# Ek 1:

Başarı oranını artırmak için RandomForestClassifier, DecisionTreeClassifier, LogisticRegression, KneighborsClassifier; classiffication modelleri kullanılarak test edilmiş ve en başarılı sonuç veren model: RandomForestClassifier seçilmiştir.

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

RandomForestClassifier, DecisionTreeClassifier, LogisticRegression classiffication modelleri yakın sonuç vermekle birlikte, KneighborsClassifier’a göre yaklaşık %52 daha başarılı sonuç vermişlerdir.

# Ek 2:

Sentiment analizi için yapılan diğer çalışmalarda genel olarak sonuçlar tek bir classification modeline odaklanarak elde edilirken, bizim yaptığımız çalışmada en başarılı olan modeli belirlemek için farklı classification modelleri üzerinde eğitim yapılarak sonuçlar elde edilmiştir.

# Ek 3:

<https://github.com/Bhasfe/distance_learning/blob/master/Distance_Learning.ipynb> ,

<https://github.com/twintproject> adreslerindeki çalışmalardan faydalanılmıştır.

İlk adresteki çalışmada BernoulliNB() modeli kullanılırken, bizim yaptığımız çalışmada RandomForestClassifier, DecisionTreeClassifier, LogisticRegression, KneighborsClassifier gibi farklı classiffication modelleri kullanılmış, ayrıca çeşitli deneysel çalışmalar yapılarak en başarılı sonuç veren model ve kütüphaneler belirlenerek kullanılmıştır.

İkinci adreste ise açık kaynak olarak sunulan twint projesinde oluşturulmuş olan kütüphane ve kodlardan faydalandık

# EK 4:

***A)***

• Silhouette Coefficient



Silhouette Coefficient her örnek için tanımlanır ve iki skordan oluşur:

a: Bir numune ile aynı kümedeki diğer tüm noktalar arasındaki ortalama mesafe.

b: Bir numune ile bir sonraki en yakın kümedeki diğer tüm noktalar arasındaki ortalama mesafe.

Bir numune seti için Silüet Katsayısı, her numune için Silüet Katsayının ortalaması olarak verilir. Skor, yanlış kümeleme için -1 ve çok yoğun kümeleme için +1 arasında sınırlandırılmıştır. Sıfır civarındaki skorlar, örtüşen kümeleri gösterir. Kümeler yoğun ve iyi ayrılmış olduğunda skor daha yüksektir, bu da standart bir küme kavramıyla ilgilidir.

• Dunn’s Index

Dunn's Index (DI), bir kümeleme algoritmasını değerlendirmek için kullanılan başka bir ölçüttür. Dunn's Index, maksimum küme boyutuna bölünen minimum kümeler arası mesafeye eşittir. Büyük kümeler arası mesafelerin (daha iyi ayırma) ve daha küçük küme boyutlarının (daha kompakt kümeler) daha yüksek bir DI değerine yol açtığına dikkat edilmelidir. Daha yüksek bir DI, daha iyi kümeleme anlamına gelir. Daha iyi kümelemenin, kümelerin kompakt ve diğer kümelerden iyi ayrılmış olduğu anlamına geldiğini varsayar.

***B)***

Transformer, son yıllarda özellikle (Natural Language Processing — NLP) mimarilerinin temel yapı taşını oluşturan derin öğrenme modelleri ailesidir. Dikkat mekanizmalarına dayanan basit bir ağ mimarisidir. Yapay zekâ modellerinin girdilerinin belli kısımlarına seçici olarak odaklanmasını ve böylece daha hızlı ve etkin bir öğrenme amaçlanmıştır. Ayrıca konuşma tanıma, sembolik matematik ve pekiştirmeli öğrenme gibi çeşitli görevlerde de transformersları görebiliyoruz. Makine dil çevirileri konusunda özellikle tercih edilirken bu çalışmada, belki de şaşırtıcı bir şekilde, bilgisayarlı görü konusu olan görüntülerden nesne tanıma problemini çözmek üzere karşımıza çıkıyor.

# Kaynakça

* <https://github.com/Bhasfe/distance_learning/blob/master/Distance_Learning.ipynb>
* [**https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk**](https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk)
* [**https://www.youtube.com/watch?v=d6GsBAgzD-I**](https://www.youtube.com/watch?v=d6GsBAgzD-I)
* [**https://towardsdatascience.com/analysis-on-tweets-using-python-and-twint-c7e6ebce8805**](https://towardsdatascience.com/analysis-on-tweets-using-python-and-twint-c7e6ebce8805)
* [**https://github.com/twintproject**](https://github.com/twintproject)
* [**https://pages.mtu.edu/~shanem/psy5220/daily/Day13/treesforestsKNN.html**](https://pages.mtu.edu/~shanem/psy5220/daily/Day13/treesforestsKNN.html)
* [**https://analyticsindiamag.com/explained-stemming-vs-lemmatization-in-nlp/**](https://analyticsindiamag.com/explained-stemming-vs-lemmatization-in-nlp/)
* [**https://neptune.ai/blog/vectorization-techniques-in-nlp-guide**](https://neptune.ai/blog/vectorization-techniques-in-nlp-guide)
* [**https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4**](https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4)

# Özdeğerlendirme Tablosu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **İstenen Madde** | **Var** | **Açıklama** | **Tahmini Not** |
| **1** | **Kapak Sayfası, Problemin Tanımı, Kullanılan Ortam, Yöntem ve Kütüphaneler, Araştırma (10)** | X | Maddede istenilen her iş yapılmıştır. | 10 |
| **2** | **Önerilen Yöntem (10)** | X | Önerilen yöntemler açık bir şekilde belirtildi. | 10 |
| **3** | **Deneysel Çalışmalar (10)** | X | Araştırmalar sonucunda, problemimizin çözümüne uygun yollar izlendi. | 10 |
| **4** | **Proje Rapor Biçimi, Organizasyonu, Boyutu, Kalitesi, Kaynakça ve atıflar (10)** | X | Rapor projenin tüm yapım aşamalarını içerecek şekilde oluşturuldu ve düzene dikkat edildi. | 10 |
| **5** | **Sonuç (10)** | X | Projede genel olarak yapılanlar, gerçek hayatta kullanılabileceği yerler hakkında bilgi verildi ve elde edilen başarı yorumlandı. | 10 |
| **6** | **Ek 1: Başarım İyileştirme (10)** | X | Maddede istenilenler açık ve anlaşılır bir şekilde yapıldı, rapora eklendi. | 10 |
| **7** | **Ek 2 (10)** | X | Maddede istenilenler açık ve anlaşılır bir şekilde yapıldı, rapora eklendi. | 10 |
| **8** | **Ek 3 (10)** | X | Maddede istenilenler açık ve anlaşılır bir şekilde yapıldı, rapora eklendi. | 10 |
| **9** | **Ek 4 (10): Her madde 5’er puan.** | X | Maddede istenilenler açık ve anlaşılır bir şekilde yapıldı, rapora eklendi. | 10 |
| **10** | **Özdeğerlendirme Tablosu (10)** | X | Tablo ayrıntılı bir şekilde ve eksiksiz olarak dolduruldu. | 10 |
| **100 üzerinden Toplam Not:** | | | | 100 |

# İş Bölümü

Projenin konu seçiminden, rapor oluşturulmasına kadar olan tüm süreçlerde iki grup üyesi eşzamanlı ortak olarak çalıştı. Günde 4-5 saat olmak üzere toplam 7 günde proje tamamlandı.