# 

# **Yapay Zeka Yöntemleri**

# **2021-2022 Bahar Yarıyılı**

# **Proje 2 Raporu**

# **19.06.2022**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Doğal Dil İşleme (Natural Language Processing)***

***ile***

***Görüş / Duygu Analizi (Sentiment Analysis)***

***Bir konu hakkında gönderilen tweet’lerin olumlu mu olumsuz mu olduğunun bulunması***

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Mustafa Kaya - 05190000059

Veli Yaşar - 05190000841

Şükriye Züleyha Alkan - 05190000107

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**İçindekiler:**

1. Problemin Tanımı
2. Araştırma/Ön Çalışma
3. Kullanılan Ortam, Yöntem ve Kütüphaneler
4. Kaynak Kodlar ve Açıklamaları
5. Ek 1: Başarı artırmak için kullanılan yöntemler ve artış miktarı
6. Ek 2: Daha önce yapılmış çalışmalar ile karşılaştırma
7. Ek 3: Kullanılan hazır kodlar/bağlantılar
8. Ek 4:
   1. Kümeleme Algoritmaları için performans ölçütleri
   2. Derin öğrenmede Transformer Model
9. İş Bölümü
10. Öz Değerlendirme Tablosu

### 0. NPL.ipnb’nin çalışması için gerekli .txt dosyası

İndirme Linki: [download (677 MB)](https://drive.google.com/drive/folders/1My6w9RQJXVp0tccfdVT-FW2hgwWopd6e?usp=sharing)

### 1. Problemin Tanımı

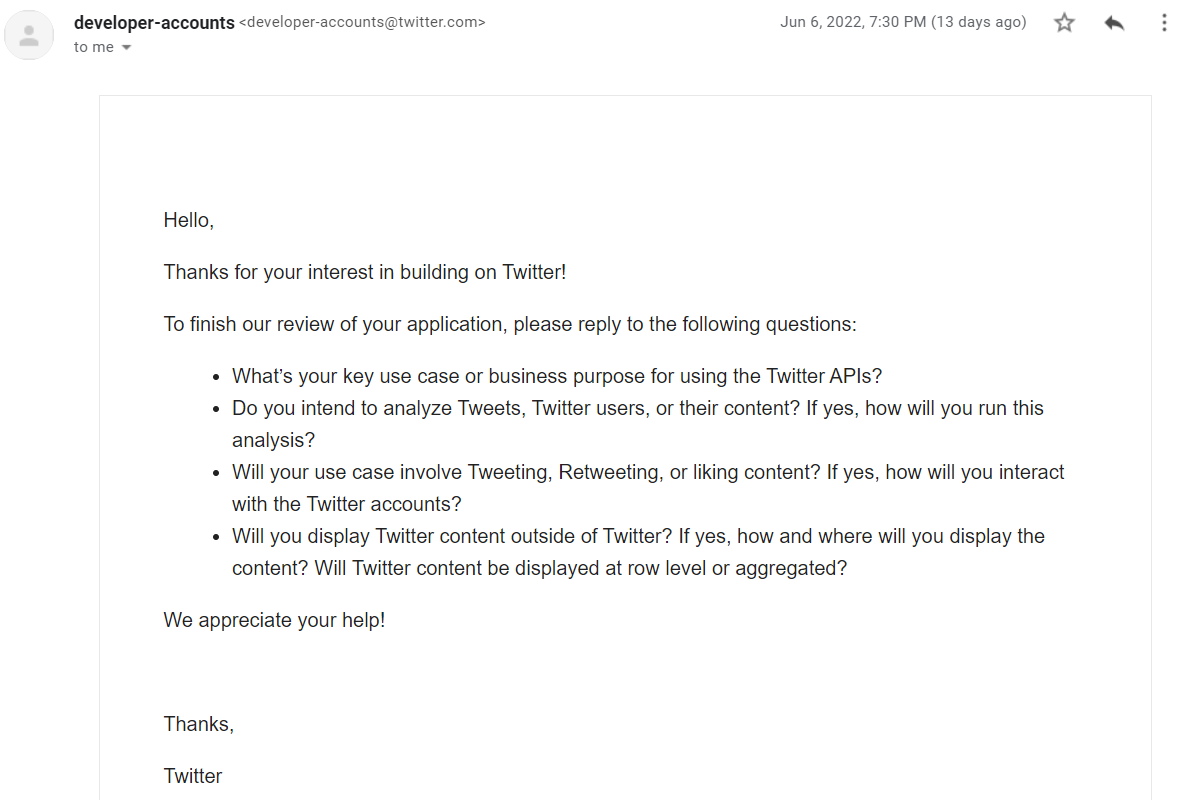
Projemizde verilen bir zaman aralığında bir konu/hashtag hakkındaki tweet’ler için duygu analizi yapmaya çalıştık. Konuyu “Lightyear”(2022) filmi, zaman aralığını ise son 24 saat olarak belirledik.

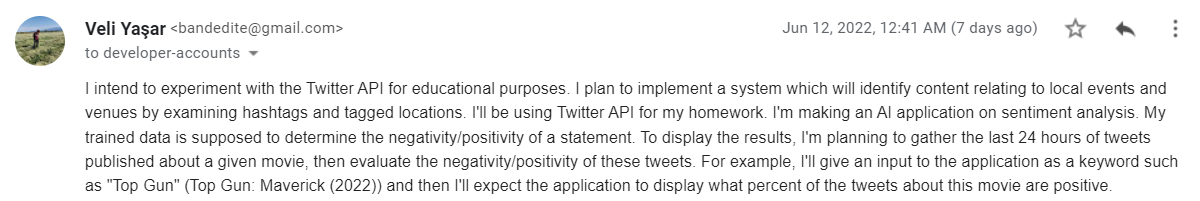
Projenin öneminden bahsetmek gerekirse; geliştirilen bir ürünün insanlar tarafından nasıl karşılandığını bilmek üretici için kritik geri bildirimler sağlar. Üretici bu sayede hatalarını fark edip daha iyisini hedefleyebilir. O yüzden çıkartılan ürünün kamuoyu tarafından nasıl karşılandığının analizinin yapılması kritik bir öneme sahiptir. Twitter da insanların en hızlı tepki verdiği platformlardan biri olarak duygu analizi için ciddi bir kaynak sağlıyor.

### 2. Araştırma/Ön Çalışma

Seçtiğimiz konu hakkında kendimizi geliştirmek için YouTube’dan birtakım eğitici videolar izledik. Algoritmaları araştırarak hangilerinin bize daha hızlı ve etkin bir şekilde sonuç verebileceğini araştırdık.

Elde ettiğimiz eğitimli sistemde kullanacağımız tweet’leri çekebilmek için Twitter API ile Twitter verisini kullanmamız gerekiyor. Twitter ilk aşamada verilere erişebilmemiz için developer hesabı açmamızı istiyor. Bunun için başvurularımızı yaptık. Twitter sadece birimize developer hesabı vermeyi onayladı.





### 3. Kullanılan Ortam, Yöntem ve Kütüphaneler

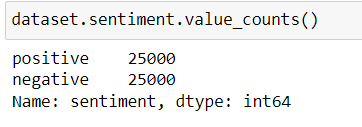
Bu projeyi Python, Anaconda, Jupyter Notebook teknolojilerini kullanarak geliştirdik.

İlk aşamada IMDB-Dataset.csv dosyasındaki yorumları kullanarak bir cümlenin olumlu ya da olumsuz olmasını tespit edecek sistemin eğitilmesini sağladık. Temel olarak veri setinin çekilmesi için BeautifulSoup algoritmasını, veri setinin eğitimi için Tensorflow/Keras kütüphanesini kullandık. Bunların yanında Tweepy kütüphanesiyle Twitter API’ya bağlanıp tweet’lere eriştik ve bu verilerin analizini yaptık.

### 4. Kaynak Kodlar ve Açıklamaları

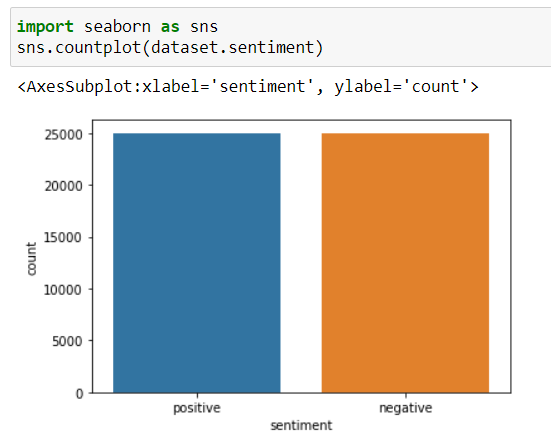
Öncelikle *pandas* ve *numpy* kütüphaneleri import ettik ve sonrasında da bize verilen “imdb\_dataset.csv” dosyasını okuduk.



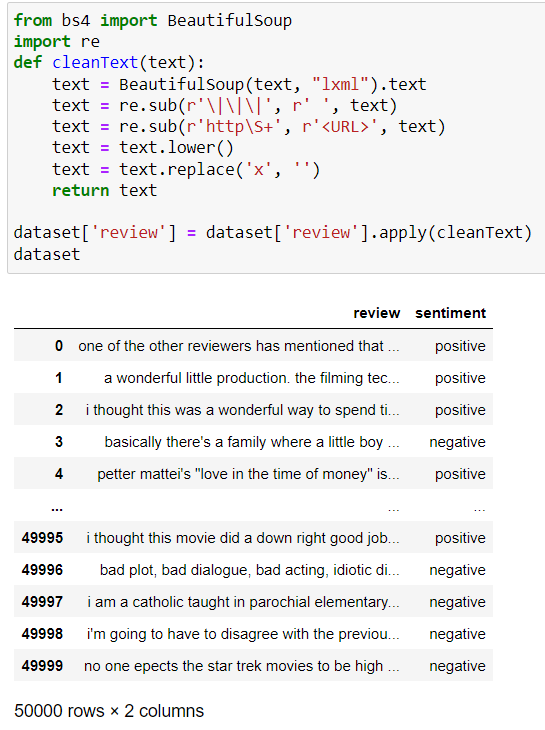


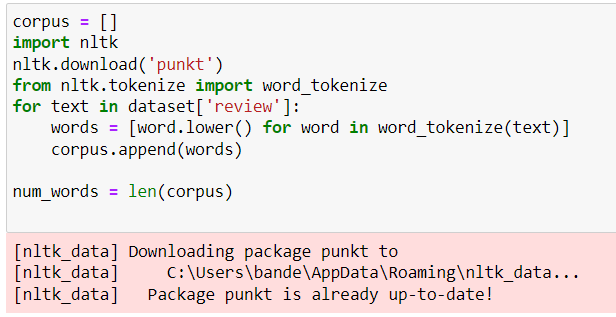
Okuduğumuz dosyamız ile alaklı datasetlerini okuyup bunların değerlerini ve sayılarını yazdırdık ve output olarak bizlere 25000 pozitif ve 25000 negatif değer döndürdü.

Sonrasında *seaborn* kütüphanesinden faydalanarak bizlere grafiksel olarak pozitif ve negatif verilerinin adedini bu şekilde gösterdim.

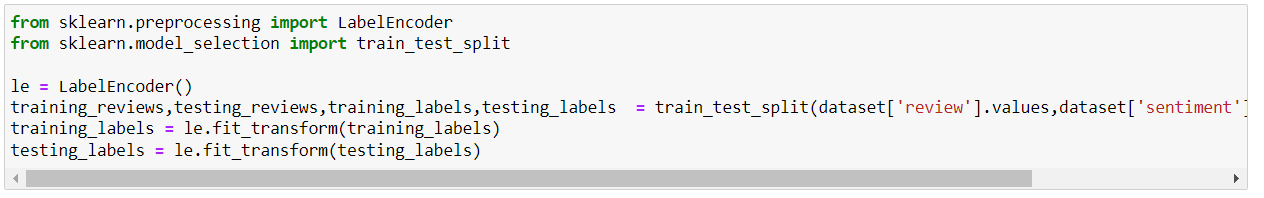


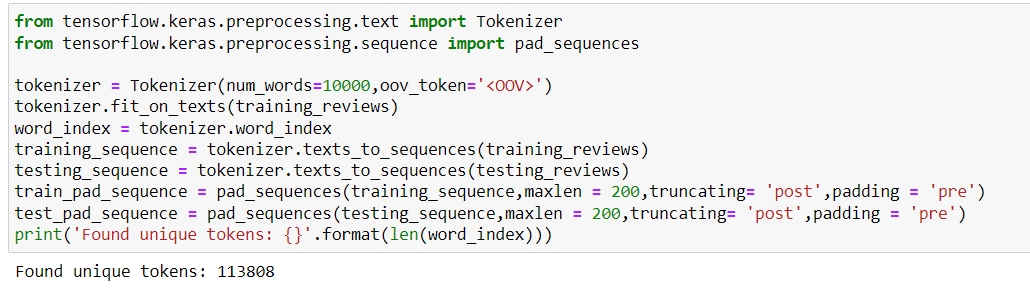
*BeautifulSoup* algoritmalarından faydalanarak textlerdeki gereksiz ifadeleri sildik. Bunu *cleantext* fonksiyonu ile yaptık. Sonrasında da output olarak tekrardan bizlere 50000 adet veri geri döndü.





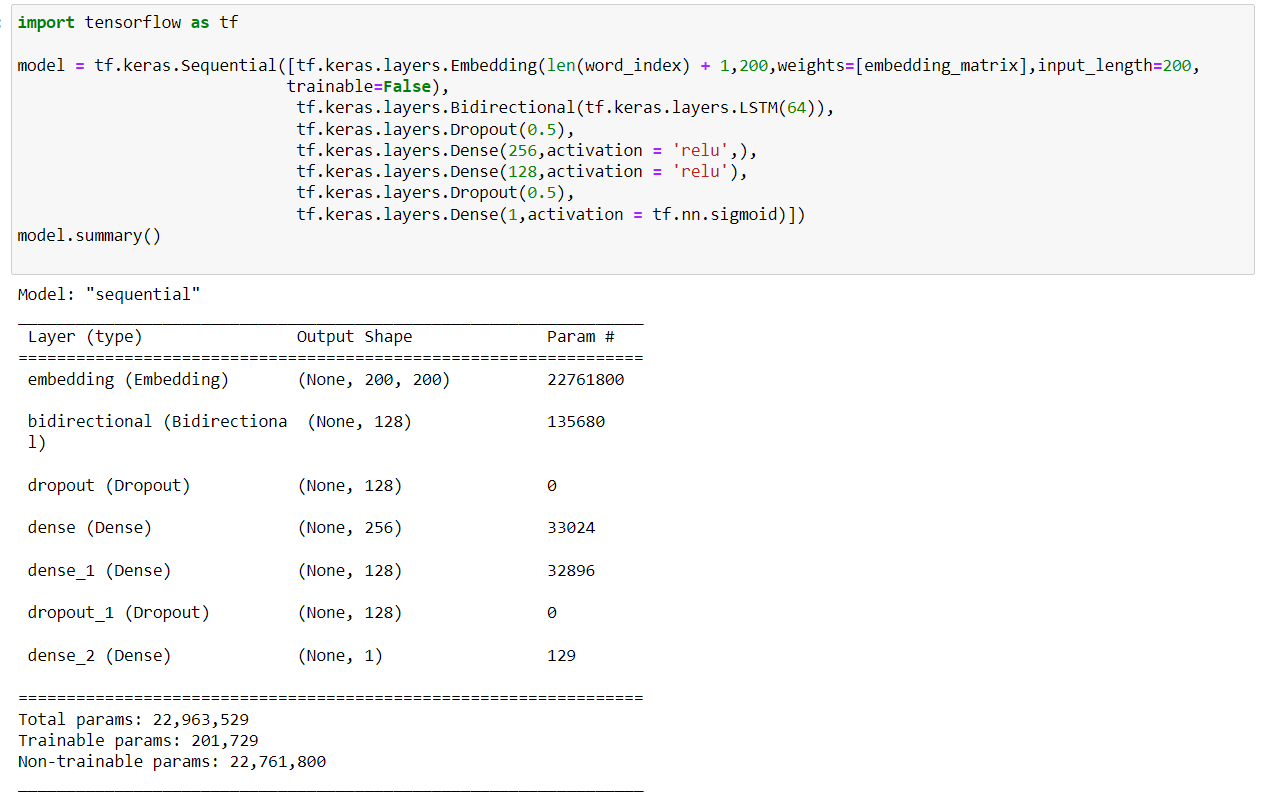
Burada yapılan işlemler genel itibari ile önce cümleleri tokenize işlemi ile kelimeler haline getirmektir. Sonrasında verisetimizi train etmeye başlıyoruz. Bir sonraki kısımda artık ve işleyişini göreceğiz.



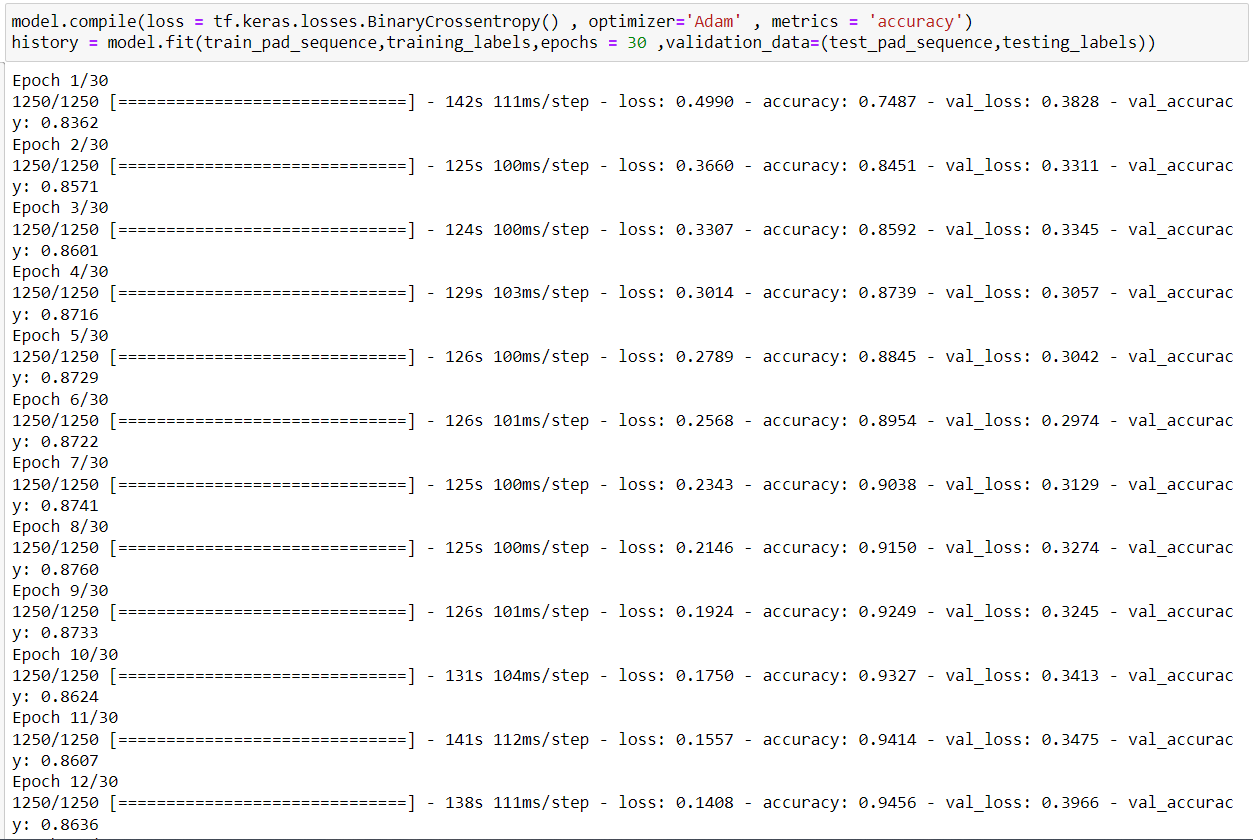




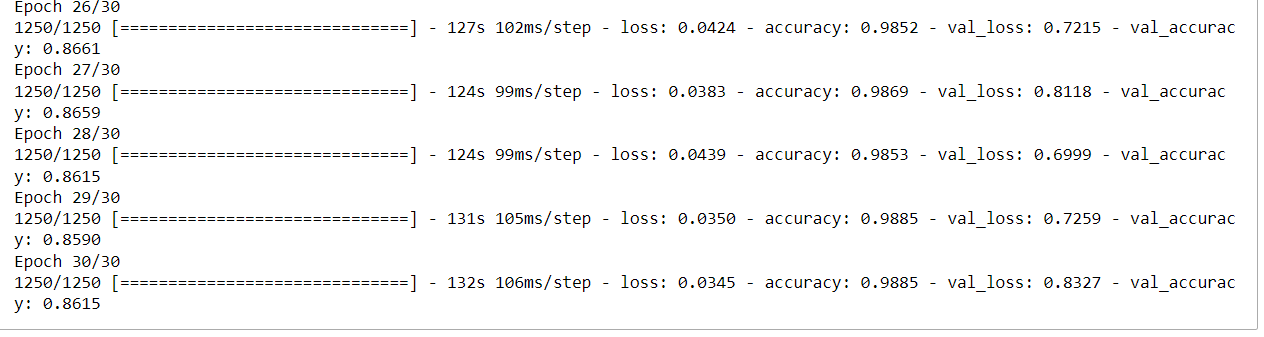
Sequentiallar’dan faydalanarak burada modelimizi oluşturmuş bulunuyoruz.



Burada verilerimiz eğitime başladı ve görüldüğü üzere “Epoch 1/30” deki accurancy = 0.7573 ve validation accurancy = 0.8338 fakat ilerleyen “Epoch”larımızda verilerimizdeki artış başlamaktadır. Verilerimizdeki doğruluk oranları da artmaya başlamaktadır.

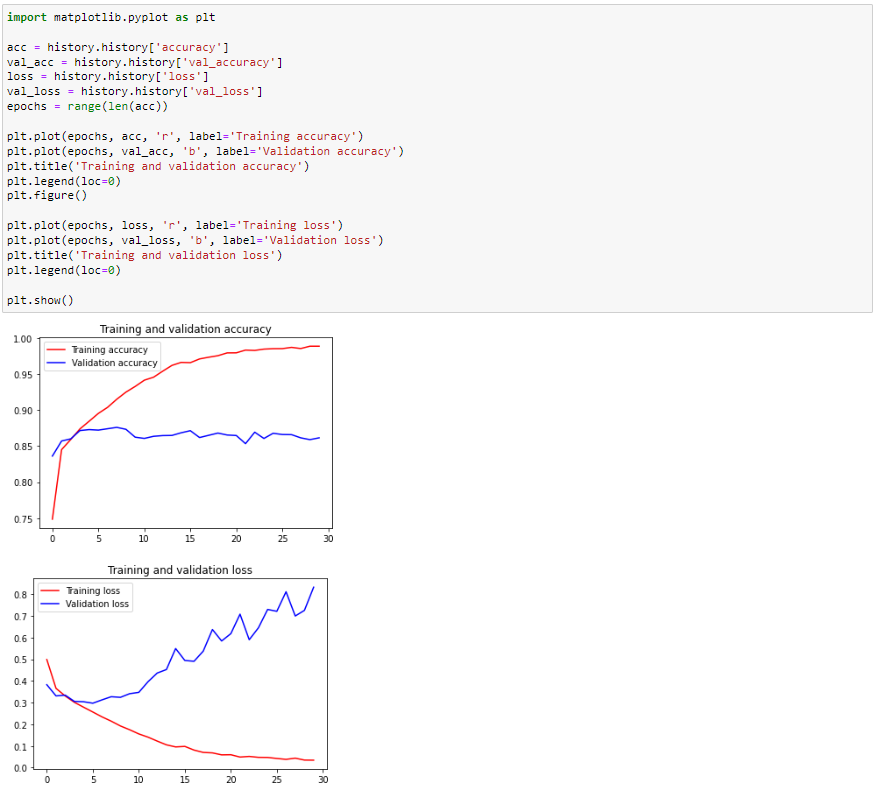


…

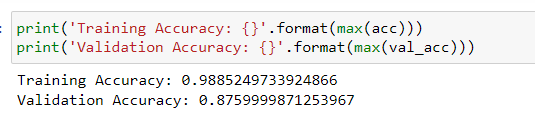


Burada veri setimimizin sonunu görüyoruz.

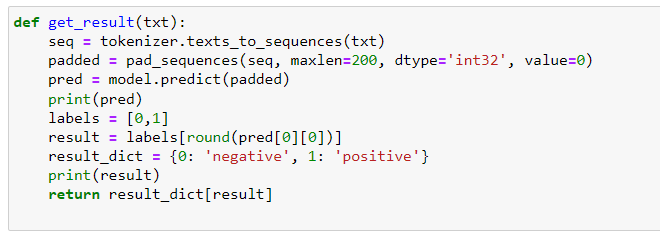
Sonrasında yapılan işlemlerle *matploit* kütüphanesi yardımı ile oluşan sonuçları grafiklerde göstermekteyiz.

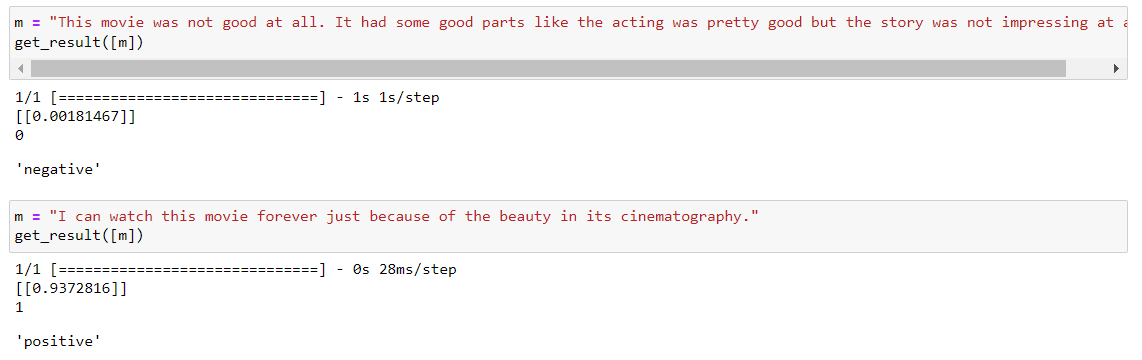


Bu kısımda yaptığımız işlemle sonuçları grafiklerde gösterdikten sonra, “training accurancy” ve “validation accuracy” kısımları ile alakalı bize dönen verilerde maksimum değerleri yazdırdık.



Bu kısımda ise bir “get\_result” metodu yazdık, bu metoda bir cümle göndereceğiz ve bize bu cümlenin pozitif ya da negatif mi olduğunu söyleyecek. Örnek olarak ise “I can watch this movie forever just because of the beauty in its cinematography.” cümlesini verdik ve *get\_result* metoduna gönderdiğimizde bu text’i bize “positive” olarak bir sonuç döndü.





Proje için Twitter aracılığı ile verilen bir key ya da hashtag sonrasında o kelime ile alakalı son 1 gün içerisinde atılmış bütün tweetleri döndürüp sonrasında bunların ne kadarı pozitif ve negatif olduğunun analizini bizlere sunmaktadır.

Bunun için öncelikle Twitter developer hesabı aldık ve Twitter tarafından api ye bağlanmak için bizlere bir takım “developer key”ler verildi ve bu “key”ler aracılığı ile erişim sağladık.

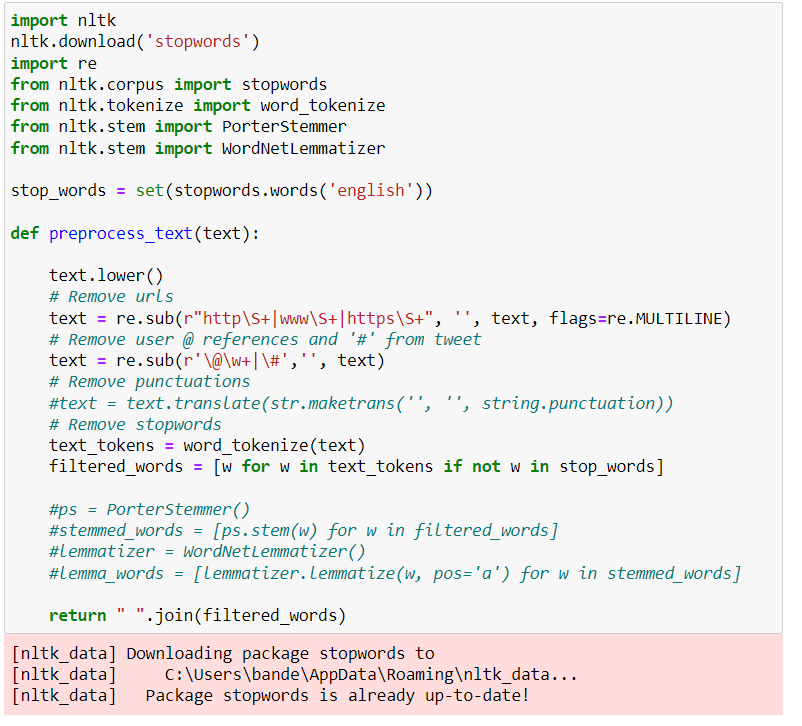
Bir *percentage* fonksiyonu oluşturduk. Bunun amacı projenin en sonunda kullanacağımız grafiği görüntülemek.

*isEnglish* fonksiyonu ise twitlerin ingilizce olup olmadığını anlamak içindir.

Sonrasında ise 24 saat içerisinde atılan tweetleri bulmak için *datetime* ve *timedelta* classlarını import ediyoruz.



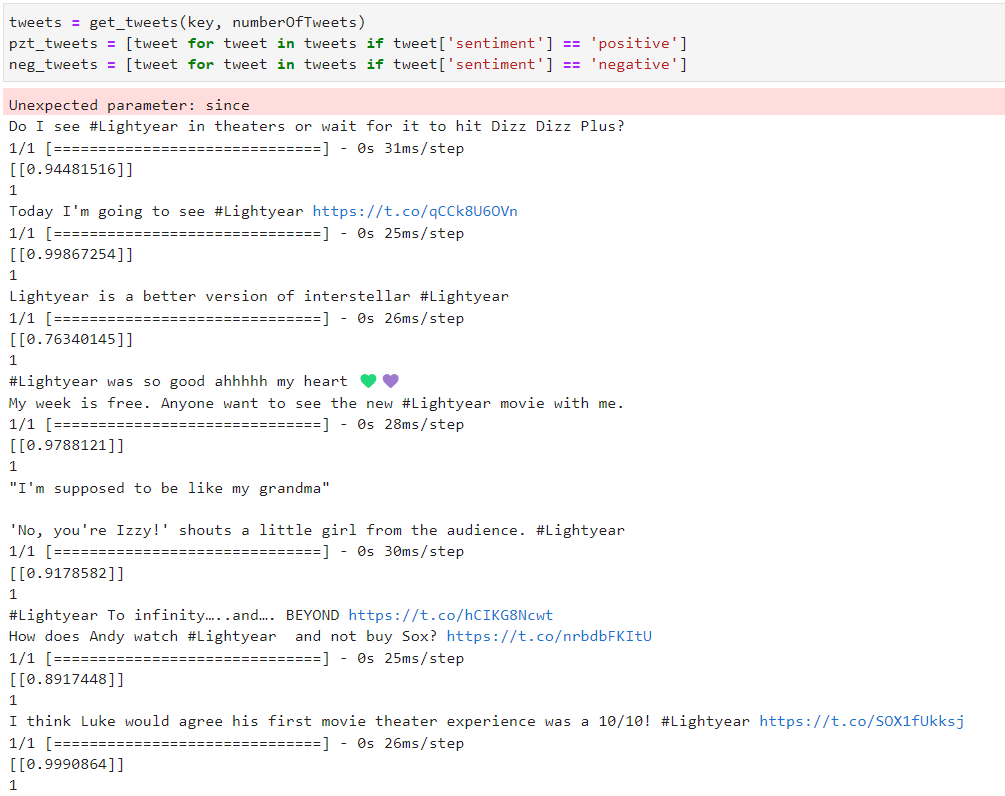
Tekrardan cümleleri tokenize ederek kelime haline dönüştürüyoruz ve bütün cümleyi *lower* ile küçültüyoruz.



“get\_tweets” fonksiyonu oluşturuyoruz ve sonrasında da text’leri çekip sırasıyla yukarıda belirtilen fonksiyonlara gönderiyoruz. Gerekli işlemler yapıldıktan sonra da işimize yarayan verileri kendi dizimize aktarıyoruz.

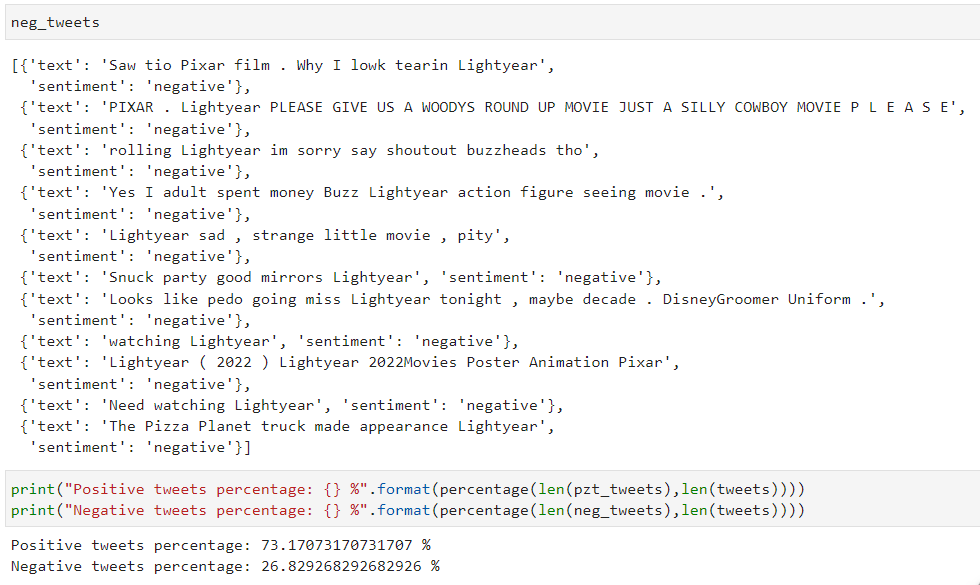


Sonrasındaki kısımda ise pozitif ve negatif tweetleri bir diziye atıp “Lightyear” filmi ile alakalı belirtilen parametrelere göre gelen tweetler aşağıda gösterilmektedir.



Burada pozitif ve negatif array’lerle atılan tweetlerin bir nevi sağlamasını yapmaktayız.

Sonrasında bu tweetler ile alakalı pozitif ve negatiflik yüzdelik oranları yazdırılmıştır.



Ve son olarak, yüzdelik kısımlarını düzenlemek adına, noktadan sonra 2 basamak alınacak şekilde sonuçlar yazdırıldı ve bu oranlara göre de bir pasta grafiği şekildeki gibi oluşturuldu.

### 5. Ek 1: Başarı artırmak için kullanılan yöntemler

Başarımı arttırmak için farklı algoritmalar kullanmayı denedim ve en verimli sonucu BeautifulSoap algoritması ile yakaladık.

### 

### 6. Ek 2: Daha önce yapılmış çalışmalar ile karşılaştırma

[Sentiment Analysis of Twitter Texts Using Machine Learning Algorithms, Sakarya University](https://dergipark.org.tr/en/download/article-file/1776091)

Türkiye’de benzer bir çalışma yapılıp yayınlanmış, kaynak kodları olmadan. Bu çalışma, kapsamının genişliği ve yöntemleri sebebiyle bize fazla yardımcı olamadı. Fakat projemizle karşılaştırmak gerekirse, bu çalışmada Support Vector Machines, Random Forest ve Gaussian Naive Bayes kullanılıyor, yorumlar pozitif/negatif/nötr olarak ayrılıyor, çok daha kapsamlı bir veri temizliği ve hazırlığı sağlanıyor, sistemi eğitmek için farklı bir “Trump’s tweets” adında bir veriseti kullanıyor.

### 7. Ek 3: Kullanılan hazır kodlar/bağlantılar

Bütün kelimeleri ayırıp ve içerisindeki olumlu olumsuz örneklemleri çıkarmak için internette daha hızlı sonuç veren algoritmaları araştırdık ve bunları da kendi koduma uyacak şekilde implemente ettik. Bu kod bloklarındaki gereksiz yerleri çıkardık ve kendi kod bloğumuza uygun şekilde düzenlemeler yaptık. Bunun yanında, farklılıklarımız aslında fonksiyonlarda yeterli bir şekilde hızlı ve işimi görecek şekilde çalışmalarıydı. Kullandığımız çoğu fonksiyon bizim tarafımızdan yazıldı.

### 8. Ek 4:

**a) Kümeleme Algoritmaları İçin Kullanılan Performans Ölçütleri**

**Silhouette Score**

Siluet Puanı ve Siluet Grafiği, kümeler arasındaki ayırma mesafesini ölçmek için kullanılır. Bir kümedeki her noktanın komşu kümelerdeki noktalara ne kadar yakın olduğunun bir ölçüsünü gösterir. Bu ölçü [-1, 1] aralığına sahiptir ve kümeler içindeki benzerlikleri ve kümeler arasındaki farklılıkları görsel olarak incelemek için kullanılan bir araçtır.

**Rand Index**

Bir kümedeki verinin değerini hesaplarken o veriyi aynı kümede olan diğer verilerle ikili olarak karşılaştırır. Bu işlemi aynı şekilde gerçek sınıf etiketleri için de yapar. Sonrasında bu yapılan işlemi ikili olarak karşılaştırır. Aynı mı yoksa farklı mı olduğunu bir tabloda tutar. Bu tablo üzerinden RI değerini hesaplar.

**b)** **Derin Öğrenmede Transformer Model Nedir?**

Transformer, giriş verilerinin her bir bölümünün önemini farklı şekilde ağırlıklandıran, kendi kendine dikkat mekanizmasını benimseyen bir derin öğrenme modelidir. Yapay zekâ (YZ) modellerinin girdilerinin belli kısımlarına seçici olarak odaklanmasını ve böylece daha hızlı ve etkin bir öğrenme amaçlanmıştır. Doğal dil işleme, bilgisayarla görme, konuşma tanıma, sembolik matematik ve pekiştirmeli öğrenme gibi çeşitli görevlerde kullanılmaktadır.

### 9. İş Bölümü

**Mustafa Kaya:** Ek 4, *preprocess\_text()*, *get\_result()*, debugging

**Veli Yaşar:** Rapor yazımı, Ek 2, Ek 3, *tweepy(Twitter API)*, *get\_tweets()*

**Şükriye Züleyha Alkan:** Model için IMDB veriseti temini, TensorFlow/Keras, BeautifulSoup, *matplotlib* ile görüntüleme

### 10. Öz Değerlendirme Tablosu

|  | İstenen Madde | Var | Açıklama | Tahmini Not |
| --- | --- | --- | --- | --- |
| 1 | Kapak Sayfası, Problemin Tanımı, Kullanılan Ortam, Yöntem ve Kütüphaneler, Araştırma (10) | + | Yapıldı | 10 |
| 2 | Önerilen Yöntem (10) | + | Yapıldı | 10 |
| 3 | Deneysel Çalışmalar (10) | + | Yapıldı | 10 |
| 4 | Proje Rapor Biçimi, Organizasyonu, Boyutu, Kalitesi, Kaynakça ve atıflar (10) | + | Yapıldı | 10 |
| 5 | Sonuç (10) | + | Yapıldı. Kodlar açıklandı. | 10 |
| 6 | Ek 1: Başarım İyileştirme (10) | - | Yapıldı fakat sonuçları kaydetmeyi unuttuğumuz için accuracy değerlerini paylaşamadık. | 2 |
| 7 | Ek 2 (10) | + | Yapıldı | 10 |
| 8 | Ek 3 (10) | + | Yapıldı | 10 |
| 9 | Ek 4 (10): Her madde 5’er puan | + | Yapıldı | 10 |
| 10 | Özdeğerlendirme Tablosu (10) | + | Yapıldı | 10 |
| 100 üzerinde Toplam Not: | | | | 92 |