ÜSKÜDAR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

2022-2023

FİNAL PROJE RAPORU

ÖĞRENCİ NUMARASI: 214327027

ÖĞRENCİ ADI-SOYADI: Hamza Talip Ekşi

PROJE BAŞLIĞI: IMDB Film Yorumlarının Duygu Analizi

1. **GİRİŞ**

Duygu analizi, mesajın duygusal tonunun olumlu, olumsuz veya tarafsız olup olmadığını belirlemek için dijital metnin analiz edilme sürecidir. Fikir madenciliği olarak da bilinen duygu analizi, şirketlerin ürün ve hizmetlerini geliştirmelerine yardımcı olan önemli bir iş zekâsı aracıdır.Bu çalışmada eğlence sektörü üzerinden duygu analizi yapacağız.

2. **MATERYALLER VE YÖNTEMLER**

Kaggle.com'dan elden edilen IMDB Movie Ratings veri seti ("https://www.kaggle.com/datasets/yasserh/imdb-movie-ratings-sentiment-analysis ") kullanılarak bir duygu analizi yapılmıştır.Dataset kullanıcı yorumları ve olumlu-olumsuz(0-1) puandan oluşmaktadır.Ayrıca model oluşturulduktan sonra tahmin yapıması için IMDB sitesinden veri setinde bulunmayan yorumlar alınmıştır.

3. **ARAÇLAR**

   Analizin yapılması için gerekli kod Google Colaboratory üzerinde pyhton dili ile yazılmıştır.

4. **KODLAR**

```
[ ]  !pip install --user tensorflow
     !pip install --user keras
```

-tensorflow ve keras kütüphaneleri yüklendi.

```
[ ]  from tensorflow.python.keras.models import Sequential
     from tensorflow.python.keras.layers import Dense, GRU, Embedding, CuDNNGRU
     from tensorflow.python.keras.optimizer_v2 import adam as adam_v2
     from keras.preprocessing.text import Tokenizer
     from keras.utils import pad_sequences
```

-Yapay sinir ağı oluşturacağımız için Sequential import ediyoruz.

-Katmanlarını oluşturmak için Dense,bilgi akışını düzenlemek için GRU,kelimeleri vektörleyebilmek için Embedding,GPU üzerinden çalışma için CuDNNGRU import edildi.

-Optimizer olarak adam_v2 import edildi.

-Tokenleme işlemi yapılacağından Tokenizer import edildi.

-Kelime vektörlerinin belli bir uzunlukta olmasını istediğimiz için pas_sequences import edildi.

```
[ ]  import numpy as np
     import pandas as pd
```

-Dizilerle çalışmak için numpy,veri analizi için pandas kullanıldı.

```
     from google.colab import drive
     drive.mount('/content/gdrive')
```

-Dataseti çekebilmek için drive ,colab'a bağlandı.

```
[ ]  %cd /content/gdrive/MyDrive/Colab Notebooks/

     /content/gdrive/MyDrive/Colab Notebooks
```

-Datasetin çekleceği klasör girildi.

```
     !ls

     'A TALE OF TWO CITIES.txt'
      movie.csv
      NlpFinal.ipynb
      NlpVize
     'Topic models with Gensim.ipynb adlı not defterinin kopyası'
      word2vec.ipynb
      word2vec.model
```

-Klasördeki dosyalar listelendi.

```
[ ]  dataset = pd.read_csv('movie.csv')
```

-Listeden movie.csv dosyası veri seti olarak kullanıldı.

```
[ ]  dataset
```

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1
...	...	...
39995	"Western Union" is something of a forgotten cl...	1
39996	This movie is an incredible piece of work. It ...	1
39997	My wife and I watched this movie because we pl...	0
39998	When I first watched Flatliners, I was amazed....	1
39999	Why would this film be so good, but only gross...	1

40000 rows × 2 columns

-Dataset kolonları text ve label olacak şekilde 2 tanedir.

```
[ ]  target = dataset['label'].values.tolist()
     data = dataset['text'].values.tolist()
```

-Datasette bulunan değerler listeye çevrildi.

```
[ ]  cutoff = int(len(data) * 0.80)
     x_train, x_test = data[:cutoff], data[cutoff:]
     y_train, y_test = target[:cutoff], target[cutoff:]
```

-Sahip olan datanın yüzde 80'i eğitim için kullanılacak.Bu yüzden bir cutoff noktası oluşturuldu.

```
[ ]  x_train[500]
```

'Just because an event really happened doesn't mean that it will make a good screenplay/ movie. The Cat's Meow, by Peter Bogdanovich claims to be based on a writer paid more attention to creating a bizarre cast of characters than taking time to create a story for the bizarre characters to inhabit. The key moment the producer, believing him to be Chaplin. Basing a key element of a story on someone wearing the wrong hat is trite and contrived. The story attempts to be n, comedy. There is also a lack of empathy for any of the characters. It hardly matters who is shot, who is killed, who is guilty and who is innocent. There ult to care about and the eventual outcome is incidental.'

- x_train 500.indekste bulunan "text" verisi.

```
[ ] x_train[800]
```

```
'Contrary to most of the comments in this section, I have to say this film just barely escapes the definition 'rubbish!'.
north-east and know at first hand what the area is like. I am totally sick of films that are supposed to be 'realistic',
social background. And funny? I smiled briefly twice and laughed once, but that was at the incongruity of two boy actors
as only one likable character in the whole film, and that was the senile grandfather played quietly but very competently
ogy department the author studied at. He certainly had no ear for th...'
```

- x_train 800.indekste bulunan “text” verisi.

```
[ ] y_train[800]
```

```
0
```

-y_train 800.indekste bulunan “label” verisi.Verinin “0” olması olumsuz yorum olduğu anlamına gelir.Karşısındaki x_train[800] verisine bakarsak gerçekten olumsuz olduğunu görüyoruz.

```
num_words = 10000
tokenizer = Tokenizer(num_words=num_words)
```

-Yorumlarda en sık geçen ilk 10000 kelime alınıp tokenleştirildi.

```
[ ] tokenizer.fit_on_texts(data)
```

-İlk 10000 data ile devam edileceği belirtildi.

```
tokenizer.word_index
```

```
{'the': 1,
 'and': 2,
 'a': 3,
 'of': 4,
 'to': 5,
 'is': 6,
 'in': 7,
 'it': 8,
 'i': 9,
 'this': 10,
 'that': 11,
 'was': 12,
 'as': 13,
 'for': 14,
 'movie': 15,
 'with': 16,
 'but': 17,
```

-Bu 10000 kelime görüntülendi.

```
[ ]  x_train_tokens = tokenizer.texts_to_sequences(x_train)
```

-Tokenizer tarafından dikkate alınan 10000 kelime bir integer değerine çevrildi.

```
[ ] x_train[800]
```

```
'Contrary to most of the comments in this section, I have to say this film just barely escapes the definition 'rubbish!'. The only readers who seem to
north-east and know at first hand what the area is like. I am totally sick of films that are supposed to be 'realistic', yet portray the working class
social background. And funny? I smiled briefly twice and laughed once, but that was at the incongruity of two boy actors with Sunderland accents suppo
as only one likable character in the whole film, and that was the senile grandfather played quietly but very competently by Roy Hudd -- the only non-G
ogy department the author studied at. He certainly had no ear for th...'
```

```
print(x_train_tokens[800])
```

```
[3735, 5, 87, 4, 1, 801, 7, 10, 2460, 9, 24, 5, 130, 10, 18, 38, 1174, 3034,
```

-800.indeksteki textte bulunan kelimelerin token değerleri yazıldı.Nadir kelimeler tokenleştirmede yok sayılır.

```
x_test_tokens = tokenizer.texts_to_sequences(x_test)
```

-Test datasıda aynı işlemden geçirildi.Kelimeler integer değerine çevrilmiş oldu.

```
num_tokens = [len(tokens) for tokens in x_train_tokens + x_test_tokens]
num_tokens = np.array(num_tokens)
```

-Her “text” verisinde kaç token bulunduğu hesaplandı ve bir dizi haline getirildi.

```
np.mean(num_tokens)
```

```
217.364725
```

-Filtrelenmiş verinin ortalama token sayısı

```
np.max(num_tokens)
```

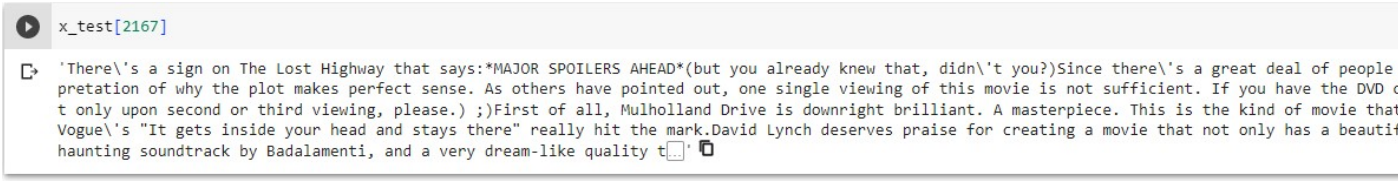
```
2122
```

-En fazla token sahibi yorum.

```
np.argmax(num_tokens)
```

```
34167
```

-En uzun veriye sahip indeksi bulur.



-40000 verinin yüzde sekseni 32000 veri yaptığından 34167 indeksindeki veri cutoff noktasından itibaren test verisidir.En uzun veriye sahip indeks bulundu.

```
max_tokens = np.mean(num_tokens) + 2 * np.std(num_tokens)
max_tokens = int(max_tokens)
max_tokens
```

```
533
```

-Ortalama ve iki standart sapmasını toplayarak yaptığımız hesapla elde ettiğimiz 533 değeri , en fazla 533 token uzunluğuna sahip verileri dikkate alarak verimizin yüzde 95'ini dikkate almış olacağımız anlamına geliyor.

```
[ ] np.sum(num_tokens < max_tokens) / len(num_tokens)
```

```
0.945225
```

-533'ten küçük tokenları toplayıp toplam token sayısına bölerek üstteki işlemin sağlaması yapıldı.

```
[ ] x_train_pad = pad_sequences(x_train_tokens, maxlen=max_tokens)
```

```
[ ] x_test_pad = pad_sequences(x_test_tokens, maxlen=max_tokens)
```

-533'ten küçük verileri doldurmamız,büyük verileri ise kesmemiz gerekiyor.pad_sequences ile 533'ten küçük verilerde boşluklar yerine "0" yazılırken, büyük verilerde ilk 533 dikkate alınıp geri kalan kısım kesilir.

```
[ ] x_train_pad.shape
```

```
(32000, 533)
```

```
[ ] x_test_pad.shape
```

```
(8000, 533)
```

-pad.shape'e baktığımızda train için 32000-533 değeri test için 8000-533 değeri görüldü.

```
[ ] np.array(x_train_tokens[800])
```

```
array([3735,    5,   87,    4,    1,  801,    7,   10, 2460,    9,   24,
          5,  130,   10,   18,   38, 1174, 3034,    1, 4945,  864,    1,
         60, 6070,   34,  303,    5,   25, 1854,    4,   47,    3,    4,
       1692,    8,    6,   22,  142,   34,   36,   67,  411,    7,    1,
       2187, 2476,    2,  118,   29,   86,  523,   47,    1, 1635,    6,
         36,    9,  236,  460, 1157,    4,  103,   11,   22,  440,    5,
         25,  243, 2056,    1,  765,  716,    4, 7187,   13,  359, 1831,
        351,   17,   90, 1378,   89,   83,    4,   64, 1128,  955,    2,
        153,    9, 3031, 1472,    2, 1437,  280,   17,   11,   12,   29,
          1,    4,  104,  419,  149,   16, 2530, 1539,  263,    5, 2485,
         64, 2530,   35, 2030,  454,   45,   12,   60,   26, 1496,  107,
          7,    1,  223,   18,    2,   11,   12,    1, 3573,  251, 5312,
         17,   51, 9559,   30, 2681,    1,   60,  687,    7,    1,  176,
         13,   14,    1,  485,   69,    9,   38,  581,   47, 3675, 2447,
          1, 2212, 6879,   29,   27,  418,   65,   53, 4591,   14,    1,
        693, 1214,    2,    9,   95,   24,  110,  545,    1, 2688,    2,
         21,   60,  236,    9,    3, 2098,  144,    3,  883,  406,   19,
          1,  693,   88,  446,  124,   54,    2,   19,   10, 1169,    8,
        210,  153,    8,  210,  789,    8,  210,  425,    2,    8,    6])
```

```
[ ] x_train_pad[800]
```

```
array([   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0, 3735,    5,   87,    4,    1,  801,
          7,   10, 2460,    9,   24,    5,  130,   10,   18,   38, 1174,
       3034,    1, 4945,  864,    1,   60, 6070,   34,  303,    5,   25,
       1854,    4,   47,    3,    4, 1692,    8,    6,   22,  142,   34,
         36,   67,  411,    7,    1, 2187, 2476,    2,  118,   29,   86,
        523,   47,    1, 1635,    6,   36,    9,  236,  460, 1157,    4,
        103,   11,   22,  440,    5,   25,  243, 2056,    1,  765,  716,
          4, 7187,   13,  359, 1831,  351,   17,   90, 1378,   89,   83,
          4,   64, 1128,  955,    2,  153,    9, 3031, 1472,    2, 1437,
        280,   17,   11,   12,   29,    1,    4,  104,  419,  149,   16,
       2530, 1539,  263,    5, 2485,   64, 2530,   35, 2030,  454,   45,
         12,   60,   26, 1496,  107,    7,    1,  223,   18,    2,   11,
         12,    1, 3573,  251, 5312,   17,   51, 9559,   30, 2681,    1,
         60,  687,    7,    1,  176,   13,   14,    1,  485,   69,    9,
         38,  581,   47, 3675, 2447,    1, 2212, 6879,   29,   27,  418,
         65,   53, 4591,   14,    1,  693, 1214,    2,    9,   95,   24,
        110,  545,    1, 2688,    2,   21,   60,  236,    9,    3, 2098,
        144,    3,  883,  406,   19,    1,  693,   88,  446,  124,   54,
          2,   19,   10, 1169,    8,  210,  153,    8,  210,  789,    8,
        210,  425,    2,    8,    6], dtype=int32)
```

-pad işleminden öncesinin ve sonrasının kıyaslaması yapıldı.Boş kısımların “0” ile doldurularak uzunluğunun 533’e tamamlandığı görülüyor.

```
[ ]  idx = tokenizer.word_index
     inverse_map = dict(zip(idx.values(), idx.keys()))
```

-Sayıları tekrardan kelimelere çevirebilmek istiyoruz.Bunun için bir sözlük yapısı oluşturuldu.

```
[ ]  def tokens_to_string(tokens):
         words = [inverse_map[token] for token in tokens if token!=0]
         text0 = ' '.join(words)
         return text0
```

-Sayılardan oluşan bir listeyi alıp bu sayıları kelimeye dönüştüren bir fonksiyon tanımlandı.

```
[ ]  x_train[800]
```

```
'Contrary to most of the comments in this section, I have to say this film just barely escapes the definition 'ru
north-east and know at first hand what the area is like. I am totally sick of films that are supposed to be 'real
social background. And funny? I smiled briefly twice and laughed once, but that was at the incongruity of two boy
as only one likable character in the whole film, and that was the senile grandfather played quietly but very comp
ogy department the author studied at. He certainly had no ear for th...'
```

```
[ ]  tokens_to_string(x_train_tokens[800])
```

```
'contrary to most of the comments in this section i have to say this film just barely escapes the definition ' the only readers
first hand what the area is like i am totally sick of films that are supposed to be yet portray the working class of wherever a
d laughed once but that was at the of two boy actors with accents supposedly trying to hide their accents from football fans th
competently by roy the only non in the cast as for the writing well i just wonder what university department the author studied
i a native i'm a who's written on the local don't waste your time an...'
```

-Ham ve filtrelenmiş halleriyle 800.indeks verisi.

```
[ ]  model = Sequential()
```

```
[ ]  embedding_size = 50
```

-Model oluşturuldu ve embedding size olarak “50“ verildi.Rastgele kelime vektörleri başlatıp model içinde eğiteceğiz.

```
[ ]  model.add(Embedding(input_dim=num_words,
                         output_dim=embedding_size,
                         input_length=max_tokens,
                         name='embedding_layer'))
```

-10000x50 kelime modele eklendi.Verideki kelime vektörlerini sonraki katmana input olarak verecek.

```
[ ]  model.add(GRU(units=16, return_sequences=True))
     model.add(GRU(units=8, return_sequences=True))
     model.add(GRU(units=4))
     model.add(Dense(1, activation='sigmoid'))
```

-GRU yapısı oluşur.Katmanlar arası çıktı alışverişi sağlandı.

```
[ ]  model.compile(loss='binary_crossentropy',
                   optimizer='rmsprop',
                   metrics=['accuracy'])
```

-Modeli compile etmek için bazı özellikler verildi.

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_layer (Embedding)  (None, 533, 50)           500000
_________________________________________________________________
gru (GRU)                    (None, 533, 16)           3216
_________________________________________________________________
gru_1 (GRU)                  (None, 533, 8)            600
_________________________________________________________________
gru_2 (GRU)                  (None, 4)                 156
_________________________________________________________________
dense (Dense)                (None, 1)                 5
=================================================================
Total params: 503,977
Trainable params: 503,977
Non-trainable params: 0
_________________________________________________________________
```

-Modelin özeti.

```
x_train_pad = np.array(x_train_pad)
y_train = np.array(y_train)
model.fit(x_train_pad, y_train, epochs=5, batch_size=256)
```

```
Epoch 1/5
125/125 [==============================] - 142s 1s/step - loss: 0.5117 - accuracy: 0.7526
Epoch 2/5
125/125 [==============================] - 137s 1s/step - loss: 0.3305 - accuracy: 0.8753
Epoch 3/5
125/125 [==============================] - 137s 1s/step - loss: 0.2707 - accuracy: 0.9024
Epoch 4/5
125/125 [==============================] - 137s 1s/step - loss: 0.2356 - accuracy: 0.9168
Epoch 5/5
125/125 [==============================] - 135s 1s/step - loss: 0.2105 - accuracy: 0.9260
<tensorflow.python.keras.callbacks.History at 0x7f5321e7ca00>
```

-Filtrelenmiş veri bir diziye dönüştürülüp modele yerleştirildi ve eğitime başlandı.

```
x_test_pad = np.array(x_test_pad)
y_test = np.array(y_test)
result = model.evaluate(x_test_pad, y_test)
```

```
250/250 [==============================] - 55s 209ms/step - loss: 0.2913 - accuracy: 0.8840
```

```
[ ] result[1]
```

```
0.8840000033378601
```

-Modele test edildi ve %88lik bir doğruluk görüldü.

```
[ ] y_pred = model.predict(x=x_test_pad[0:1000])
    y_pred = y_pred.T[0]
```

-İlk 1000 satır için tahminde bulunuldu.

```
[ ] cls_pred = np.array([1.0 if p>0.5 else 0.0 for p in y_pred])
```

-Gelen sonuç 0,5 üzerindeyse 1 yap,altındaysa 0 yap diyerek eşikten geçirildi ve tahmin sınıfı yapısı oluşturuldu.

```
[ ] cls_true = np.array(y_test[0:1000])
```

-Doğruluğu tespit etmek için gerçek sınıflar elde edildi.

```
[ ]  incorrect = np.where(cls_pred != cls_true)
     incorrect = incorrect[0]
```

-İki yapı karşılaştırılarak eşleşmeyenlerin indeksleri “incorrect”’e atıldı.

```
[ ]  len(incorrect)

     127
```

-İlk 1000 satır için kaç yanlış tahmin yapıldığı bulundu.

```
     idx = incorrect[0]
     idx

     0
```

-İlk yanlış tahminin yapıldığı indeks.

```
[ ]  text0 = x_test[idx]
     text0

     'This movie was a major bait and switch. I rented it because of Rebecca St. James, a popular Christian singer. I have met her
     elp out a friend, or a friend of a friend. My first clue that this movie wasn't what it was supposed to be was when I witness
     ors were. It was funny how almost everyone in the movie wore solid colors. (There are a few exceptions).Rebecca was verrryyy
     se husband has return to the fold. Doesn't she ever leave the house? I had to turn off the movie several times in order to fi
     r and go to hell. Jesus spread a message of love and hope. His messag...'
```

-İlk yanlış tahmin “0” indeksinde yapılmış.Bu indeks görüntülendi.Yorumun olumsuz olduğu açık.

```
[ ]  y_pred[idx]

     0.9440327
```

-Bizim modelimizin bu olumsuz yorumu olumlu olarak yanlış yorumladığı görülüyor.

```
[ ]  cls_true[idx]

     0
```

-Gerçek sınıfının “0” yani olumsuz olduğu görülüyor.

```
text1 = " A thrilling masterpiece worth the long run time even better in IMAX! Important to note Rip Lance Reddick he was amazing in his career!"
text2 = "Hands down the best action movie franchise ever."
text3 = "John Wick: Chapter 4 offered me 169 minutes of pure entertainment with an exciting cast of characters and action sequences that'll impress anyone, a perfect action film"
text4 = "Bad acting, bad casting, bad adaptation. Do not watch this if you enjoyed the book. From start to finish it's bad, there's no redeeming qualities to this film! Parker looks
text5 = "This movie was nothing like the book, except for the character's names. The acting was atrocious and there was barely any chemistry...I came for the steam, but there is no
text6 = "I'm a huge fan of the book. I was so excited when the movie was announced. Man, did the screenwriter even read the book? This is the biggest jumbled mess of crap! The dialog
text7 = "It all leads back to where we once started off as all great trilogies have indicated from the past. But GOTG surpasses expectations with what is nothing short of phenomenal
text8 = "This. This is what I've wanted. Yeah some of the jokes are a bit too silly and the tone is a bit confusing at times. But the end result for me is probably the most heartfelt
text9 = "I think for what this movie sets out to do, Super Mario Bros. Movie pretty much ticks the boxes of an entertaining, reference fan-service heavy movie. A lot of the jokes are
text10 = "The story was simple, but worked wonders with the excellent characters that perfectly reflected their game counterparts. They even had unique aspects given to them that he
texts = [text1, text2, text3, text4, text5, text6, text7, text8, text9, text10]
```

-IMDB sitesinden 2023 yılına ait 3 filme yapılan yorumları alarak modelimizde olumlu mu olumsuz mu oldukları test edildi.Bu yorumlar bir texts listesine eklendi.

```
[ ]  tokens = tokenizer.texts_to_sequences(texts)
```

-Bu texts listesi tokenize edildi.

```
[ ]  tokens_pad = pad_sequences(tokens, maxlen=max_tokens)
     tokens_pad.shape

     (10, 533)
```

-Bu tokenlar daha sonra sınırımız olan 533’e pad edildi.Yani boşluuklar 0’lar ile doldurulurken fazlalıklar silindi.10 yorumumuz var hepsinin 533 token’ı var.

```
[ ]  model.predict(tokens_pad)

     array([[0.89164174],
            [0.50121   ],
            [0.6166402 ],
            [0.03234053],
            [0.07595197],
            [0.0389641 ],
            [0.962625  ],
            [0.86469615],
            [0.36488223],
            [0.96354073]], dtype=float32)
```

-Modelimizin yaptığı tahminler ortaya çıkmış oldu.

SONUÇLAR

```
[ ]  model.predict(tokens_pad)

     array([[0.89164174],
            [0.50121   ],
            [0.6166402 ],
            [0.03234053],
            [0.07595197],
            [0.0389641 ],
            [0.962625  ],
            [0.86469615],
            [0.36488223],
            [0.96354073]], dtype=float32)
```

-Tahmin yapılması istenen on yorumun ilk üçünün çok olumlu ,sonraki üç yorumun olumsuz, yedinci yorumun olumlu, sekizinci-dokuzuncu-onuncu yorumların ise nötre yakın olumlu yorumlar olduğu görülüyor.Modelimiz birinci yorumda doğru tahminde bulunsa da sonraki iki yorumu nötre yakın görmüş.Sonraki üç yorumda ise bu yorumların olumsuz olduğunu tahmin edebilmiş.Yedinci yorumu da doğru bir şekilde olumlu olarak tahmin etmiş.Sekiz ve onuncu yorumları çok olumlu bulmuş.Yorumların olumluya kaydığı doğru ama olumsuz ifadelerde var.Bu sebeple oran biraz daha düşük olabilirdi.Dokuzuncu yorumu olumsuza yakın görmüş ama nötr olması gerekirdi.

Sonuç olarak olumsuz yorumlarda gerçekten başarı yüzde yüz iken olumlu yorumların bir kısmında ve nötr yorumlarda tam olarak istenen başarıya sahip değil.

5. ÖNERİ VE TARTIŞMALAR

Modeli daha başarılı hale getirmemiz için eğitim kalitesi artmalı.Bunu sağlamak için ya veri miktarı artırılabilir ya da eğitim yapılırken kullanılan epoch sayısı artırılabilir.

KAYNAKLAR

-https://www.kaggle.com/datasets/yasserh/imdb-movie-ratings-sentiment-analysis

-https://www.imdb.com/title/tt10366206/?ref_=adv_li_tt

-https://m.imdb.com/title/tt2316548/?ref_=m_tt_urv

-https://www.imdb.com/title/tt6718170/?ref_=adv_li_tt

KONTROL LİSTESİ

	EVET /HAYIR
Raporunuzu şablonda belirtildiği gibi hazırladınız mı?	Evet
Çalışmanızın sonuçlarını (print screen) rapora eklediniz mi?	Evet
Rapor dosyanızı şablondaki gibi yeniden adlandırdınız mı?	Evet
Raporu sisteme yüklediniz mi?	Evet
Kodları sisteme yüklediniz mi?	Evet