

## Ürün Yorum Analizi Projesi

Bu proje, bir e-ticaret sitesinden alınan ürün yorumlarını inceleyerek, yorumların olumlu mu yoksa olumsuz mu olduğunu belirlemek için bir yapay zeka modeli kullanmaktadır.

### Kullanılan Teknolojiler ve Kütüphaneler:

- Python
- Pandas
- NumPy
- Seaborn
- Matplotlib
- Requests
- JSON
- TensorFlow Keras (Sequential, Dense, GRU, Embedding)
- Tokenizer ve pad\_sequences

### Projenin amacı :

Projede amaç, web scraping ile alınan ürün yorumlarını analiz ederek, her bir yorumun olumlu mu yoksa olumsuz mu olduğunu belirlemektir. Bu amaç doğrultusunda, doğal dil işleme (NLP) teknikleri ve derin öğrenme modelleri kullanılmaktadır.

### geliştiren : Hüseyin Erol

Kişisel websitem : <https://huseyineroll.com/> (<https://huseyineroll.com/>)

Github : <https://github.com/HuseyinErol24> (<https://github.com/HuseyinErol24>)

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sbn
import matplotlib.pyplot as plt
import requests
import json
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GRU, Embedding
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

WARNING:tensorflow:From C:\Users\Hüseyin\AppData\Roaming\Python\Python311\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse\_softmax\_cross\_entropy is deprecated. Please use tf.compat.v1.losses.sparse\_softmax\_cross\_entropy instead.

## Veri Çekme İşlemi

```
In [2]: for i in range(1,3):
        url = f"https://user-content-gw-hermes.hepsiburada.com/queryapi/v2/ApprovedUserContents?sku=HBV000"
        veri = requests.get(url).content
        veri = json.loads(veri)
        for i in veri["data"]["approvedUserContent"]["approvedUserContentList"]:
            print("Yorum => :",i["review"]["content"],"\\n")
```

Yorum => : Logitech m170i iade edip bunu aldım. Çünkü kablolu mousedan sonra o çok küçük geldi. büyük sevenlere gelsin :) kumpasla ölçtüm sizin için daha ne yapayım...

Yorum => : Hafif kullanışlı. Tam istediğim gibi. Tavsiye edilir.

Yorum => : Ürün fiyatına göre harika,Kucuk boy istemiyordum,boyuda gayet iyi.Tavsiye ediyorum

Yorum => : Lenovo... Kalite ve Markanın bir arada olması logitech mouse kullanıyorum aynı zamanda her i kaside çok iyi. Boyut olarak Logitech M171 den biraz büyük ama Razer DeathHadder V2 mini ile aynı boy utta. Tavsiye edilir.

Yorum => : Ürün kısa sürede elime ulaştı. Hafif ve kullanışlı bi ürün. Büyüklüğü gayet iyi. Kaliteli bi ürün. Üründen çok memnun kaldım.

Yorum => : verdiğin parayı fazlasıyla hakediyor, bilinmeyen markalara kesinlikle para vermeyin ürün t am elinize göre Marka belli kalite belli.

Yorum => : beklediğimden daha iyi çıktı tahmin etmezdim bu denli iyi olacağını.tuş tıklama sesi,pil k apağının çalılıp kapanmasının pratikliği,malzeme kalitesi ve denediğim her zeminde(tahta,koltuk üstü,v b) sorunsuz çalıştı.tam bir F/P ürünü oyun oynamıyorsanız ideal bir mouse..

Yorum => : Ürün hafif ve ergonomik. Diğer mouse'lar dan frkı açma kapatma tuşu olması beni çok memnun etti pilimi tasarruflu kullanmış oluyorum. Pil takma yerini kullanma kılavuzuna bakmasam bulamazdım b aya akıllıca olmuş. Alalı birkaç ay oldu ama memnunum

Yorum => : None

Yorum => : Ben memnun kaldım rahat ürün hızlı kargo

Yorum => : Hafif kullanışlı. Tam istediğim gibi. Tavsiye edilir.

Yorum => : Ürün fiyatına göre harika,Kucuk boy istemiyordum,boyuda gayet iyi.Tavsiye ediyorum

Yorum => : Lenovo... Kalite ve Markanın bir arada olması logitech mouse kullanıyorum aynı zamanda her i kaside çok iyi. Boyut olarak Logitech M171 den biraz büyük ama Razer DeathHadder V2 mini ile aynı boy utta. Tavsiye edilir.

Yorum => : Ürün kısa sürede elime ulaştı. Hafif ve kullanışlı bi ürün. Büyüklüğü gayet iyi. Kaliteli bi ürün. Üründen çok memnun kaldım.

Yorum => : verdiğin parayı fazlasıyla hakediyor, bilinmeyen markalara kesinlikle para vermeyin ürün t am elinize göre Marka belli kalite belli.

Yorum => : beklediğimden daha iyi çıktı tahmin etmezdim bu denli iyi olacağını.tuş tıklama sesi,pil k apağının çalılıp kapanmasının pratikliği,malzeme kalitesi ve denediğim her zeminde(tahta,koltuk üstü,v b) sorunsuz çalıştı.tam bir F/P ürünü oyun oynamıyorsanız ideal bir mouse..

Yorum => : Ürün hafif ve ergonomik. Diğer mouse'lar dan frkı açma kapatma tuşu olması beni çok memnun etti pilimi tasarruflu kullanmış oluyorum. Pil takma yerini kullanma kılavuzuna bakmasam bulamazdım b aya akıllıca olmuş. Alalı birkaç ay oldu ama memnunum

Yorum => : None

Yorum => : Ben memnun kaldım rahat ürün hızlı kargo

Yorum => : İhtiyaç gören güzel bir ürün tavsiye ederim. Oldukça kaliteli. Thurst Mouse vardı Bir sene içerisinde bozuldu. Ondan daha kaliteli gözüküyor. Ve daha tepki süresi kısa gibi. Alabilirsiniz.

Bu şekilde istediğimiz siteden istediğimiz kadar yorumu bir araya getirebiliriz. Ne kadar çok veri kullanırsak, model o kadar doğru sonuçlar verecektir. Ben örnek olması adına bu adımı kısa tuttum.

```
In [3]: veri = pd.read_csv("yorumlar.csv")
```

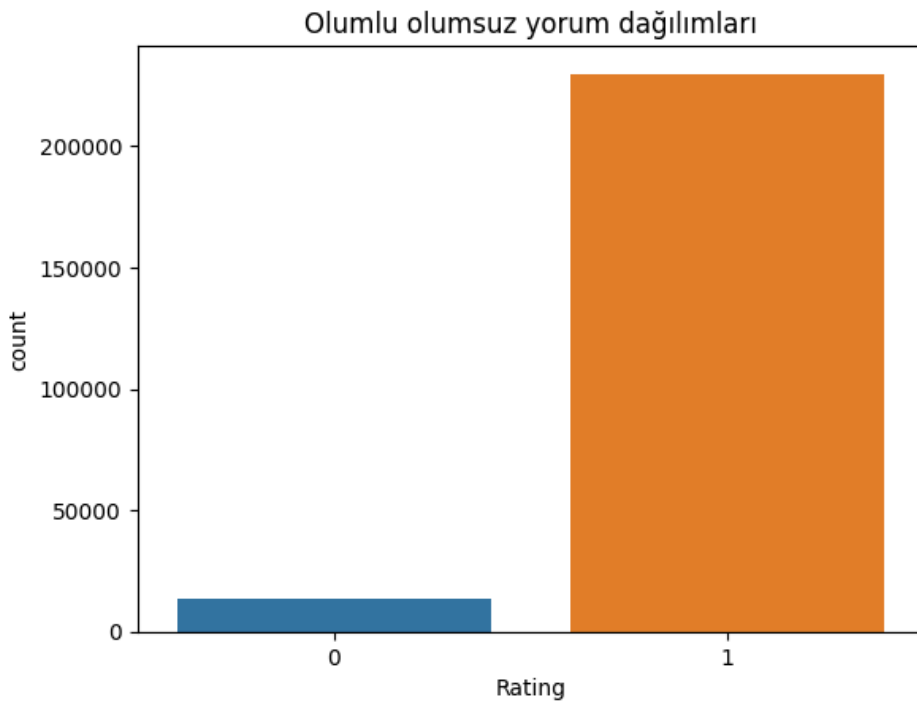
```
In [4]: veri.head() #1 degerleri 4-5 yıldızlı yorumlar 0 degerleri 1-2 yıldızlı yorumlar
#veri çekme aşamasında link üzerindeki stars kısmından bunu ayaralaya bilirsiniz
```

```
Out[4]:
```

	Rating	Review
0	1	3 yıldır tık demedi. :)
1	1	3 yıldır kullanıyorum müthiş
2	1	Ürün bugün elime geçti çok fazla inceleme fırs...
3	1	Almaya karar verdim. Hemencecik geldi. Keyifle...
4	1	Günlük kullanımınızı çok çok iyi karşılıyor kı...

```
In [5]: plt.title("Olumlu olumsuz yorum dağılımları")
sbn.countplot(data = veri,x="Rating")
```

```
Out[5]: <Axes: title={'center': 'Olumlu olumsuz yorum dağılımları'}, xlabel='Rating', ylabel='count'>
```



```
In [6]: y_degeri = veri["Rating"].values.tolist()
x_degerleri = veri["Review"].values.tolist()
```

```
In [7]: veri.info()
#243497 adet yorum mevcut csv formatında
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243497 entries, 0 to 243496
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype  
---  -
0   Rating  243497 non-null    int64  
1   Review  243497 non-null    object  
dtypes: int64(1), object(1)
memory usage: 3.7+ MB
```

## Veri Setini Ayırma

```
In [8]: y_degeri[30] #olumlu yorum degeri
```

```
Out[8]: 1
```

```
In [9]: x_degerleri[30] #o degere denk gelen yorum
```

```
Out[9]: 'Logitech her zamanki gibi kaliteli.'
```

```
In [10]: #veri setinin %80 test olarak ayırıcım bundan dolayı
```

```
In [11]: sayi = int(len(veri)*0.8)
```

```
In [12]: sayi # veri setinin %80 ni 194797 denk geliyor
```

```
Out[12]: 194797
```

```
In [13]: x_train = x_degerleri[0:sayi]
y_train = y_degeri[0:sayi]
x_test  = x_degerleri[sayi:]
y_test  = y_degeri[sayi:]
```

```
In [14]: print("X_train : ",len(x_train))
print("Y_train : ",len(y_train))
print("X_Test  : ",len(x_test))
print("Y_test  : ",len(y_test))
```

```
X_train : 194797
Y_train : 194797
X_Test  : 48700
Y_test  : 48700
```

## Tokenleştirme

```
In [15]: kelime_sıklığı = 10000
#en çok sık kullanılan 10K kelime ile işleme devam ediyoruz
```

```
In [16]: tokenizer = Tokenizer(num_words=kelime_sıklığı)
```

```
In [17]: tokenizer.fit_on_texts(x_degerleri)
```

```
In [18]: tokenizer.word_index
```

```
'olduğu': 122,
'teslimat': 123,
'com': 124,
'arkadaşlar': 125,
'sağlam': 126,
'oluyor': 127,
'teslim': 128,
'ettim': 129,
'zaman': 130,
'kalite': 131,
'almak': 132,
'az': 133,
'telefonu': 134,
'yeni': 135,
'cihaz': 136,
'marka': 137,
'son': 138,
'koku': 139,
'ise': 140,
'bulundum': 141
```

```
In [19]: x_train_token = tokenizer.texts_to_sequences(x_train)
```

```
In [20]: x_train_token[120] #tokenleştirilmiş yorum
```

```
Out[20]: [4, 221, 779, 30, 5155, 191, 14, 79]
```

```
In [21]: x_train[120] #tokenleştirilmeden önceki hal
```

```
Out[21]: 'Ürün şu ana kadar hatasız çalışıyor. Gayet başarılı.'
```

```
In [22]: x_test_token = tokenizer.texts_to_sequences(x_test)
```

```
In [23]: x_test_token[800]
```

```
Out[23]: [3226,
          308,
          2,
          4699,
          260,
          693,
          185,
          1556,
          1810,
          185,
          64,
          3379,
          20,
          78,
          30,
          13,
          269,
          2,
          550,
          50]
```

```
In [24]: x_test[800]
```

```
Out[24]: 'Çıkış musluğu herhangi bir darbe veya zorlanma görmediği halde su damlatıyor, açma kapama vanasının
buandığı alandan su damlatıyor. Ayrıca denildiği gibi o kadar da sessiz bir motor değil.'
```

## Boyut Belirleme işlemi

```
In [25]: max_token_uzunluk = 50
```

```
In [26]: x_train_pad = pad_sequences(x_train_token,max_token_uzunluk)
x_test_pad = pad_sequences(x_test_token,max_token_uzunluk)
```

```
In [27]: x_train_pad[45] #bütün yorumları aynı boyuta getirmek için bu adımı gerçekleştiriyoruz
```

```
Out[27]: 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,  5,  30, 4261,
                  3, 5494,  1,  7,  2, 587])
```

```
In [28]: x_train_token[45]
```

```
Out[28]: [5, 30, 4261, 3, 5494, 1, 7, 2, 587]
```

```
In [29]: key_value = tokenizer.word_index
ters_cevir =dict(zip(key_value.values(),key_value.keys()))
# key-value yapısını yer değiştiriyoruz yani sayı karşılığına kelime gelecek şekilde
#düzenliyoruz
```

```
In [30]: def vektor_cevir(vektor):
text = ""
for i in vektor:
    if i == 0:
        text += "" #0 degeri gelirse ekleme yapmıyoruz
    else:
        text += " " +ters_cevir[i] #0 dan farklı sayı gelirse onun karşılığındaki kelimneyi ekliyoruz.
return text.strip()
```

```
In [31]: vektor_cevir(x_train_pad[45]) #tokenleşmiş veri verdik ve o bize normal bir çıktı verdi
```

```
Out[31]: 'bu kadar yorumlarını ve puanı çok güzel bir mouse'
```

In [32]: `model = Sequential()`

WARNING:tensorflow:From C:\Users\Hüseyin\AppData\Roaming\Python\Python311\site-packages\keras\src\backend\tf\_keras\_backend.py:873: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

In [33]: `embedding_size = 50 #token boyutuna eşit`

In [34]: `model.add(Embedding(input_dim=kelime_sıklığı,  
output_dim=embedding_size,  
input_length=embedding_size,))`

In [35]: `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"))  
# sigmoid ile çıktı değerlerini 0-1 arasına yerleştiriyoruz`

In [36]: `optimizer = Adam(learning_rate=0.001)`

In [37]: `model.compile(  
optimizer=optimizer,  
loss='binary_crossentropy',  
metrics=['accuracy']  
)`

In [38]: `y_train = np.array(y_train)`

In [39]: `model.fit(x_train_pad,y_train,epochs=5,batch_size=256)`

Epoch 1/5

WARNING:tensorflow:From C:\Users\Hüseyin\AppData\Roaming\Python\Python311\site-packages\keras\src\utils\tf\_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\Hüseyin\AppData\Roaming\Python\Python311\site-packages\keras\src\engine\base\_layer\_utils.py:384: The name tf.executing\_eagerly\_outside\_functions is deprecated. Please use tf.compat.v1.executing\_eagerly\_outside\_functions instead.

761/761 [=====] - 39s 44ms/step - loss: 0.1665 - accuracy: 0.9547

Epoch 2/5

761/761 [=====] - 35s 46ms/step - loss: 0.0897 - accuracy: 0.9716

Epoch 3/5

761/761 [=====] - 35s 46ms/step - loss: 0.0707 - accuracy: 0.9783

Epoch 4/5

761/761 [=====] - 34s 45ms/step - loss: 0.0562 - accuracy: 0.9842

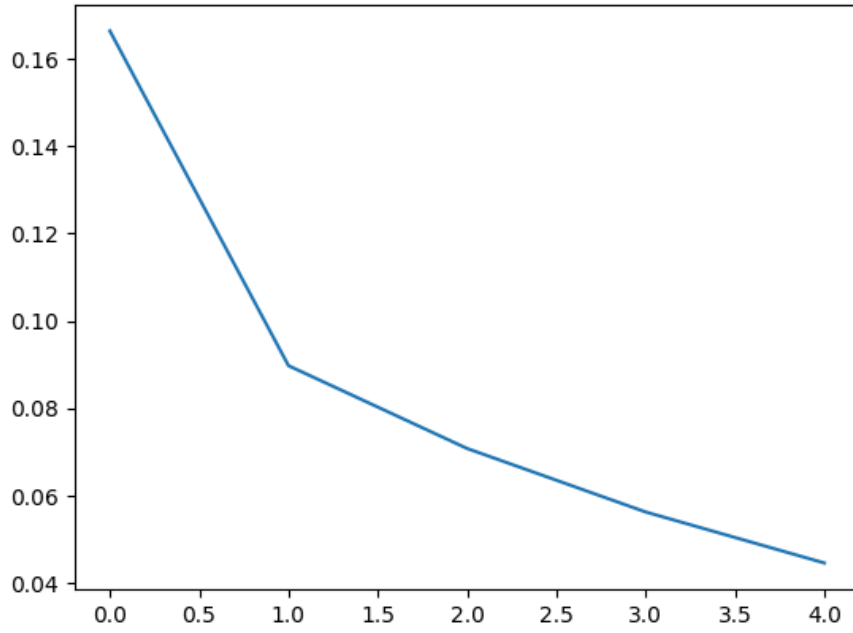
Epoch 5/5

761/761 [=====] - 34s 45ms/step - loss: 0.0445 - accuracy: 0.9882

Out[39]: <keras.src.callbacks.History at 0x1f2387f9f90>

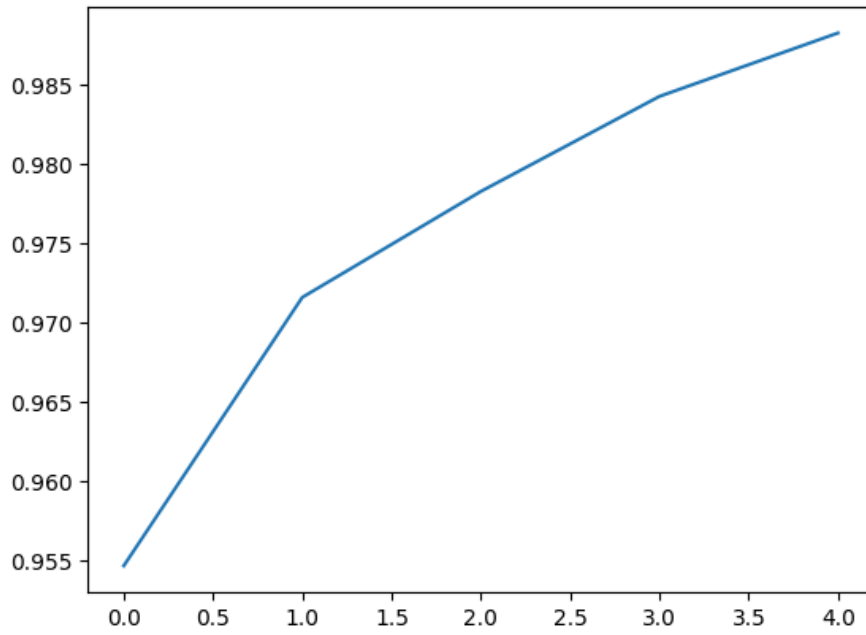
```
In [40]: plt.plot(model.history.history["loss"])
```

```
Out[40]: [<matplotlib.lines.Line2D at 0x1f2457d4f50>]
```



```
In [41]: plt.plot(model.history.history["accuracy"])
```

```
Out[41]: [<matplotlib.lines.Line2D at 0x1f2457a3490>]
```





## Modelin Test Aşaması

```
In [44]: from colorama import Fore #yazıları renkli yazmak için kullandığımız kütüphane fonksiyonu => Fore
def yorum_analiz(model,yorum):
    yorum_token = tokenizer.texts_to_sequences(yorum)
    tokens_pad = pad_sequences(yorum_token,50)
    sonuç = model.predict(tokens_pad)
    if sonuç > 0.9:
        print(Fore.GREEN)
        print(yorum[0], " ==> yorumu son derece olumlu, değeri: ", sonuç)
    elif 0.8 < sonuç <= 0.9:
        print(Fore.GREEN)
        print(yorum[0], " ==> yorumu gayet olumlu, değeri: ", sonuç)
    elif 0.7 < sonuç <= 0.8:
        print(Fore.BLUE)
        print(yorum[0], " ==> yorumu olumluya yakın, değeri: ", sonuç)
    elif 0.6 < sonuç <= 0.7:
        print(Fore.BLACK)
        print(yorum[0], " ==> yorumu kararsız, değeri: ", sonuç)
    elif 0.5 < sonuç <= 0.6:
        print(Fore.BLACK)
        print(yorum[0], " ==> yorumu nötrdür, değeri: ", sonuç)
    elif 0.4 < sonuç <= 0.5:
        print(Fore.RED)
        print(yorum[0], " ==> yorumu olumsuz a yakın, değeri: ", sonuç)
    elif 0.3 < sonuç <= 0.4:
        print(yorum[0], " ==> yorumu oldukça olumsuz, değeri: ", sonuç)
    else:
        print(Fore.RED)
        print(yorum[0], " ==> yorumu aşırı olumsuz, değeri: ", sonuç)
```

```
In [45]: yorum = [  
    ["fiyat performans bir ürün gerçekten begendim"],  
    ["kalitesiz bir üründü malesef"],  
    ["Bu ürün gerçekten harika, kesinlikle tavsiye ederim"],  
    ["Fiyatına göre çok iyi bir performans, memnun kaldım"],  
    ["Hızlı teslimat ve mükemmel kalite, teşekkürler"],  
    ["Beklentilerimi fazlasıyla karşıladı, tekrar alırım"],  
    ["Kullanımı çok rahat, herkesin denemesini öneririm"],  
    ["Ürün beklediğim gibi değildi, iade ettim"],  
    ["Ürünün kalitesi gerçekten düşük, hayal kırıklığına uğradım"],  
    ["Müşteri hizmetleri çok yardımcı oldu, üründen memnun kaldım"],  
    ["Ürün açıklamalarda anlatıldığı gibi değildi, hayal kırıklığı yaşadım"],  
    ["Hızlı kargo ve güzel paketleme, teşekkürler"],  
    ["Ürünü beğenmedim, çok kalitesiz"],  
    ["Ürün fiyatına göre oldukça iyi, tavsiye ederim"],  
    ["Maalesef ürün beklediğim kalitede değil"],  
    ["Malzeme kalitesi çok düşük, tavsiye etmiyorum"],  
    ["Ürün çok çabuk bozuldu, memnun kalmadım"],  
    ["Paketleme çok kötüydü, ürün hasar görmüş"],  
    ["Parasına değmez, beklentimi karşılamadı"],  
    ["Renk soldu ve dikişler açıldı, kaliteli değil"],  
    ["Beklediğim gibi değil, çok dayanıksız"],  
    ["Ürünün fotoğrafları yanıltıcı, hayal kırıklığına uğradım"],  
    ["Kargo çok geç geldi, ürün hasar görmüştü"],  
    ["Ürün açıklamalara uymuyor, iade etmek zorunda kaldım"]  
]  
  
#modeli test etmek amacıyla daha önce görmediği verileri veriyoruz bu aşamada  
for i in yorum:  
    print(yorum_analiz(model,i))  
  
#modelin Tahmin ettiği sonuçlar alt kısımda yer alıyor.  
#yeşil renkerer olumlu kırmızı renkteki yorumlar modele göre olumsuz metinlerdir
```

```
1/1 [=====] - 0s 25ms/step

fiyat performans bir ürün gerçekten begendim ==> yorumu son derece olumlu, değeri: [[0.9971269]]
None
1/1 [=====] - 0s 25ms/step

kalitesiz bir ürünü malesef ==> yorumu aşırı olumsuz, değeri: [[0.07484412]]
None
1/1 [=====] - 0s 25ms/step

Bu ürün gerçekten harika, kesinlikle tavsiye ederim ==> yorumu son derece olumlu, değeri: [[0.99722856]]
None
1/1 [=====] - 0s 26ms/step

Fiyatına göre çok iyi bir performans, memnun kaldım ==> yorumu son derece olumlu, değeri: [[0.9971991]]
None
1/1 [=====] - 0s 26ms/step

Hızlı teslimat ve mükemmel kalite, teşekkürler ==> yorumu son derece olumlu, değeri: [[0.99715537]]
None
1/1 [=====] - 0s 24ms/step

Beklentilerimi fazlasıyla karşıladı, tekrar alırım ==> yorumu son derece olumlu, değeri: [[0.9971396]]
None
1/1 [=====] - 0s 24ms/step

Kullanımı çok rahat, herkesin denemesini öneririm ==> yorumu son derece olumlu, değeri: [[0.9951724]]
None
1/1 [=====] - 0s 25ms/step

Ürün beklediğim gibi değildi, iade ettim ==> yorumu aşırı olumsuz, değeri: [[0.07847032]]
None
1/1 [=====] - 0s 25ms/step

Ürünün kalitesi gerçekten düşük, hayal kırıklığına uğradım ==> yorumu aşırı olumsuz, değeri: [[0.02175852]]
None
1/1 [=====] - 0s 24ms/step

Müşteri hizmetleri çok yardımcı oldu, üründen memnun kaldım ==> yorumu son derece olumlu, değeri: [[0.9895117]]
None
1/1 [=====] - 0s 25ms/step

Ürün açıklamalarda anlatıldığı gibi değildi, hayal kırıklığı yaşadım ==> yorumu aşırı olumsuz, değeri: [[0.08123326]]
None
1/1 [=====] - 0s 25ms/step

Hızlı kargo ve güzel paketleme, teşekkürler ==> yorumu son derece olumlu, değeri: [[0.9970301]]
None
1/1 [=====] - 0s 24ms/step

Ürünü beğenmedim, çok kalitesiz ==> yorumu aşırı olumsuz, değeri: [[0.00980569]]
None
1/1 [=====] - 0s 27ms/step

Ürün fiyatına göre oldukça iyi, tavsiye ederim ==> yorumu son derece olumlu, değeri: [[0.99709964]]
None
1/1 [=====] - 0s 25ms/step

Maalesef ürün beklediğim kalitede değil ==> yorumu aşırı olumsuz, değeri: [[0.01904464]]
None
1/1 [=====] - 0s 29ms/step

Malzeme kalitesi çok düşük, tavsiye etmiyorum ==> yorumu aşırı olumsuz, değeri: [[0.01355513]]
None
1/1 [=====] - 0s 26ms/step

Ürün çok çabuk bozuldu, memnun kalmadım ==> yorumu aşırı olumsuz, değeri: [[0.05344826]]
None
1/1 [=====] - 0s 26ms/step
```

```
Paketleme çok kötüydü, ürün hasar görmüş ==> yorumu aşırı olumsuz, değeri: [[0.13211456]]
None
1/1 [=====] - 0s 26ms/step
```

```
Parasına değmez, beklentimi karşılamadı ==> yorumu aşırı olumsuz, değeri: [[0.01280491]]
None
1/1 [=====] - 0s 25ms/step
```

```
Renk soldu ve dikişler açıldı, kaliteli değil ==> yorumu aşırı olumsuz, değeri: [[0.17141478]]
None
1/1 [=====] - 0s 25ms/step
```

```
Beklediğim gibi değil, çok dayanıksız ==> yorumu aşırı olumsuz, değeri: [[0.02304285]]
None
1/1 [=====] - 0s 25ms/step
```

```
Ürünün fotoğrafları yanıltıcı, hayal kırıklığına uğradım ==> yorumu aşırı olumsuz, değeri: [[0.0247
9012]]
None
1/1 [=====] - 0s 25ms/step
```

```
Kargo çok geç geldi, ürün hasar görmüştü ==> yorumu aşırı olumsuz, değeri: [[0.26969114]]
None
1/1 [=====] - 0s 27ms/step
```

```
Ürün açıklamalara uymuyor, iade etmek zorunda kaldım ==> yorumu aşırı olumsuz, değeri: [[0.0121102
1]]
None
```

In [ ]: