



Veri Bilimi ve Mühendisliği

Derin Öğrenme Mimarisi ile Fine Food Reviews

Proje ilişkin kodlar:

<https://github.com/atak-05/Fine-Food-Reviews->

Hazırlayan : Gizem Çırıkka

Bölüm Numarası:Y210240060

Bölüm :Yazılım Mühendisliği Yüksek Lisans

İÇİNDEKİLER

GİRİŞ	2
1.Notebook kullanımı ve gerekli kütüphaneler	2
2-Veri Setinin Projeye Eklenmesi ve Veri Setinin İncelenmesi	2
3-Yıldızlı Yorumlara İlişkin Sonuçlar	3
4- Basit NLTK (Natural Language Toolkit)	3
5-Vader Modeli ve Sentiment(Duygusalılık) Skor ve Analizleri	5
6-Grafiksel Vader Sonuçları ve Değerlendirmesi	6
7-Roberta Önceden Eğitilmiş Modeli	7
8-Roberta Model vs Vader Modeli	8
9-Vader Modeline İlişkin Grafiksel Sonuçlar ve Değerlendirmesi	8
10-Roberta Modeline İlişkin Grafiksel Sonuçlar ve Değerlendirmesi	9
11-İki Model sonuçlarının değerlendirilmesi	10
12-Kaynakçalar	11

GİRİŞ

Proje içerisinde python da duygu analizine ilişkin bir çalışma yapılmıştır. Bu proje içerisinde veri seti olarak kaggle kütüphanesinden alınan Amazon'da "Fine Food Reviews" adlı veri seti kullanılmıştır. Bu proje içeriğinde python'da kodlanmıştır. Bu analiz yapılırken iki teknik kullanılmıştır. İlk olarak Vader modelinin kelime çantası yaklaşımını kullanılacaktır. İkinci olarak roberta tipi bir model olan önceden eğitilmiş "hugging face" modeli kullanılmıştır.

Bu model daha geliştirilmiş bir transformer modeldir. İki model arasında sonuçlar karşılaştırılmıştır. Ayrıca hugging face pipeline ları kullanılmıştır.

Veri seti olarak Amazon'da İyi Yemek İncelemeleri seti ile bu çalışma yapılmıştır. Bu veri setinde iyi yemek ile ilgili yorumlar toplanmıştır. Burada metin incelemeleri ve ayrıca 5 yıldız ile değerlendirmeler yapılmıştır. .

Bu projeye başlarken ilk olarak kaggle notebook üzerinde çalışılmıştır. Ve ilgili veri seti projeye dahil edilmiştir.

1. Notebook kullanımı ve gerekli kütüphaneler

Notebook üzerinden ilk olarak veri seti okumak için python kütüphanelerinden pandas ayrıca numpy, matplotlib ve seaborn gibi ilgili kütüphaneler projeye dahil edilmiştir.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Daha sonrasında Doğal Dil Araç Kiti (NLTK) kullanılmıştır. NLTK, istatistiksel doğal dil işlemede (NLP) uygulamak için insan dili verileriyle çalışan Python programları oluşturmak için kullanılan bir platformdur. Tokenizasyon, ayrıştırma, sınıflandırma, stemming, etiketleme ve anlamsal akıl yürütme için metin işleme kütüphaneleri içerir.

```
plt.style.use('ggplot')
import nltk
```

2-Veri Setinin Projeye Eklenmesi ve Veri Setinin İncelenmesi

Projeye veri setinin csv dosyasını okumak için pandas kütüphanesi kullanılmıştır. Veri Setinin ilk 500 ilk datası verilmiştir.

```
# Read in data
df = pd.read_csv('../input/amazon-fine-food-reviews/Reviews.csv')
print(df.shape)
df = df.head(500)
print(df.shape)
```

Burada her satırın benzersiz Id, Product Id ve User Id bulunmaktadır. Ayrıca Score olarak her datanın kendi skor puanı bulunmaktadır. Bu skor bu yorumu yapan kişinin ürünü yorumlayan kişinin ürün için verdiği skor puanıdır. Bir ve beş arasında yıldız puanı vermektedir. Burada duygu analizi için bu veri setinde yarım milyon inceleme vardır. Ancak ilk olarak veri setini bu şekilde örneklendirilmiştir.

```
df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary
0	1	B001E4KFQ0	A35GX7AUHUGW	delmarian	1	1	5	1303862400	Good Quality Dog Food

Head komutu ile , hangi sütunlara sahip olduğumuzu görebilir ve referans verebilir.

```
df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary
0	1	B001E4KFQ0	A35GX7AUHUGW	delmarian	1	1	5	1303862400	Good Quality Dog Food

Şimdi bu veri setinin neye benzediği hakkında bir fikir edinmek için biraz hızlı bir“eda” yapılmıştır. Bu eda aracı python dilinde oluşturulmuş ve verilerin özetlenmesi için kullanılan bir araçtır. Veri seti içerisindeki bir ile beş arasında bir değer olduğunu bildiğimiz bu “puan” sütununu kullanılacak ve üzerinde sayılan bir değer yapılacaktır. Buradaki yıldız score puanlarını sayısal bir puana çevrilecek ve çubuklu grafik şeklinde gösterilecektir. Bu grafikte yıldızlara göre inceleme sayısı gösterilmiştir.



Aşağıda bir örnek yorum görülmektedir, daha sonra kullanabilmek için bunu bir değer olarak “example” atanmıştır ve örnek olarak sadece 50. değeri seçilmiştir. Bu yorum’a baktığımızda negatif bir yorum olduğu anlaşılmaktadır. Ve olumsuz bir duygu gibi görünmektedir.

4- Basit NLTK (Natural Language Toolkit)

3-Yıldızlı Yorumlara İlişkin Sonuçlar

```
Quick EDA
```

```
ax = df['Score'].value_counts().sort_index() \
    .plot(kind='bar',
          title='Count of Reviews by Stars',
          figsize=(10, 5))
ax.set_xlabel('Review Stars')
```

[11] <AxesSubplot:title={'center':'Count of Reviews by Stars'}>

Ve burada x_label etiketi kullanılmıştır.

Burada incelemelerin çoğunun aslında 5 yıldız olduğunu görülmektedir. Yorumların çoğunun olumlu olduğu görülmektedir. Bu çıkarımı yıldız skor değerlendirmelerine göre söylenilmektedir.

```
Basic NLTK
```

```
(13): example = df['Text'][50]
      print(example)
```

This oatmeal is not good. Its mushy, soft. I don't like it. Quaker Oats is the way to go.

Aşağıda görülen kod dizisinde daha önceden import ettiğimiz nltk modelinden tokenizer aracı bir önceki kod dizisinde atanan example değişkeni için çalıştırılmış. Ve aşağıdaki sonu. gözlenmiştir. Bu sonuca göre yorumdaki tüm kelimeler tek tek ayrılmıştır.

```

> nltk.word_tokenize(example)
[14... ['This',
'oatmeal',
'is',
'not',
'good',
'Its',
'mushy',
'soft',
'I',
'do',
'n't',
'like',
'it',
'Quaker',
'Oats',
'is',
'the',
'way']

```

Projemiz içerisinde başka bir nltk aracı olan pos tags ile bir konuşmanın belirli bir bölümü için sözcükleri metin biçiminde işaretleme işlemidir . Bazı NLTK POS etiketleme örnekleri şunlardır: CC, CD, EX, JJ, MD, NNP, PDT, PRP\$, TO, vb. POS etiketleyici, cümlelerin her bir kelimesinin dilbilgisel bilgilerini atamak için kullanılır.

NLTK POS Tags Examples are as below:

Abbreviation	Meaning
CC	coordinating conjunction
CD	cardinal digit
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/subordinating conjunction
JJ	This NLTK POS Tag is an adjective (large)
JJR	adjective, comparative (larger)
JJS	adjective, superlative (largest)
LS	list marker
MD	modal (could, will)
NN	<u>noun, singular (sarah)</u>
NNS	noun plural (books)
NNP	proper noun, singular (sarah)
NNPS	proper noun, plural (indians or americans)
PDT	predeterminer (all, both, half)
POS	possessive ending ('s, 's)

Belirlenen örnek için bu araç çalışır ise etiketlemeler aşağıdaki görseledeki gibi olmaktadır.Bu sayede her kelimenin etiketlemesi yapılmıştır.

```

> nltk.pos_tag(tokens)
[16... [('This', 'DT'),
('oatmeal', 'NN'),
('is', 'VBZ'),
('not', 'RB'),
('good', 'JJ'),
('Its', 'PRP$'),
('mushy', 'NN'),
('soft', 'JJ'),
('I', 'PRP'),
('do', 'VBP'),
('n't', 'RB'),
('like', 'VB'),
('it', 'PRP'),
('Quaker', 'NNP'),
('Oats', 'NNPS')]

```

Burada örnek yorum için ilk 10 etiketlemeyi görmektedir. Aslında her etiket veya kısaltmanın bir anlamı bulunmaktadır. Tekil, Çoğul, İsim, Sıfat vb şeklinde açıklanılmaktadır.

```

[17]: tagged = nltk.pos_tag(tokens)
tagged[10]

[17... [('This', 'DT'),
('oatmeal', 'NN'),
('is', 'VBZ'),
('not', 'RB'),
('good', 'JJ'),
('Its', 'PRP$'),
('mushy', 'NN'),
('soft', 'JJ')]

```

Bu işlemde ise etiketlemeleri sanki yorum'un bir parçasıymış gibi yorumun içine yerleştirdik. Ve bunu entities değeri olarak atanmıştır.Bunu nltk araçlarından biri olan chunk metodu ile gerçekleştirildi. Bu metot ile tokenizer sonuçlarına etiketlemedeki sonuçları ön ad olarak eklemektedir.

```

> entities = nltk.chunk.ne_chunk(tagged)
entities.pprint()

(S
  This/DT
  oatmeal/NN
  is/VBZ
  not/RB
  good/JJ
  Its/PRP$
  mushy/NN
  soft/JJ
  I/PRP
  do/VBP
  n't/RB
  like/VB
  it/PRP
  (ORGANIZATION Quaker/NNP Oats/NNPS)
  is/VBZ
  )

```

Buraya kadar nltk modeli üzerinde çalışmalar yapıldı. Şimdi duygu analize ilişkin çalışmalar yapılmaya başlanacaktır. Bunun için ilk olarak Vader modelini kullanılacaktır. Bu modeli açıklanacak olursa Vader modeli;VADER, Valence Aware Dictionary and SEntiment Reasoner anlamına gelir. Lexicon ve kurallara dayalı bir duyarlılık analiz kütüphanesidir. Kütüphane, Sentiment Analytics alanında popülerdir.VADER basit bir kural tabanlı duyarlılık analizörüdür. (dergipark #)

Aslında vader modeli, var olan kelimelerden belirli matematiksel denklemler sonucu belirli kelimelerin pozitif, nötr ve negatif olarak çıkarımlar yapmaktadır.

Bu modelin dezavantajlarından biri de konuşmalar arasındaki bağlantıyı hesaba katmamasıdır. Bu yüzden cümlelerin gidişatına uygun bir çıkarım yapamamaktadır.Vader

modelinin teorik anlatımından sonra şimdi uygulamaya geçilebilir.

İlk olarak model kullanımı için gerekli importlar projeye eklendi.

İkinci sırada bulunan tqdm modeli sadece veri setini çalıştırırken bir progress bar kullanılmak için projeye dahil edilmiştir. Daha sonrasında import ettiğimiz Sentiment Intensity Analyzer fonksiyonunu sia adında bir nesne olarak oluşturulmuştur.

5-Vader Modeli ve Sentiment(Duygusalılık) Skor ve Analizleri

```
[23]: from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm

sia = SentimentIntensityAnalyzer()
```

Bu metin üzerinde çalıştırabileceğimiz ve görebileceğimiz duygu analiz hesaplayıcı (Sentiment Intensity Analyzer) nesne vardır.

Duygunun kelimelere bağlı olduğu bu vader yaklaşımında “I am so happy!” cümlesi vader modeli tarafından analiz edildiğinde pozitif bir sonuç verdiğini görülmektedir. Vader bu kelime için negatifliği sıfır olarak hesaplamıştır. Ve ayrıca compound değeri bulunmaktadır. Bu değer ile negatiften pozitive doğru ne kadar negatif olduğunu temsil ediyor

Vader modeli burada doğru bir sonuç verdiği söylenebilir.

```
[23]: sia.polarity_scores('I am so happy!')

[23_... {'neg': 0.0, 'neu': 0.318, 'pos': 0.682, 'compound': 0.6468}
```

tam tersi denenecek olursa sia.polarity_scores("this is the worst thing ever") cümlesini analiz edilecek olursa sonucun çoğunlukla negatif olduğu görülmektedir..

```
[24]: sia.polarity_scores('This is the worst thing ever.')

[24_... {'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}

[26_... {'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}
```

bu yüzden tüm veri setinde polarite puanını çalıştırılırsa bu kod bloğu ile bir döngü oluşturuldu ve 500 adet yorumun için bu analiz çalıştırıldı.

Bu analizin sonuçları için “res” adında bir değer olarak atanmıştır.

```
[31]: # Run the polarity score on the entire dataset
res = {}
for i, row in tqdm(df.iterrows(), total=len(df)):
    text = row['Text']
    myid = row['Id']
    res[myid] = sia.polarity_scores(text)

100% 500/500 [00:00<00:00, 893.78it/s]
```

Şimdi bu aşağıdaki sonuca bakılacak olursa hem negatif hem neutral ve pozitif sonuçlarını görülmektedir. Burada 500 adet yorumun hepsinin sia polarity skorlarını görülmektedir.

```
[34_... vaders = pd.DataFrame(res).T

      neg  neu  pos  compound
1  0.000  0.695  0.305    0.9441
2  0.079  0.853  0.068   -0.1027
3  0.091  0.754  0.155    0.8265
4  0.000  1.000  0.000    0.0000
5  0.000  0.552  0.448    0.9468
...
496 0.000  0.554  0.446    0.9725
497 0.059  0.791  0.150    0.8215
498 0.025  0.762  0.212    0.9848
499 0.041  0.904  0.055    0.1280
```

```
vaders = pd.DataFrame(res).T
vaders = vaders.reset_index().rename(columns={'index': 'Id'})
vaders = vaders.merge(df, how='left')
```

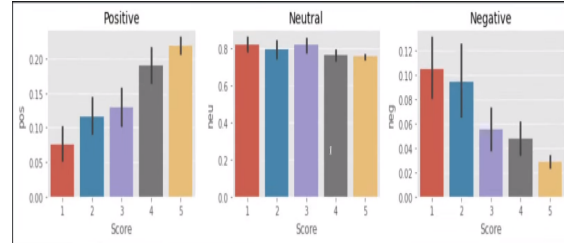
	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator
0	1	0.000	0.695	0.305	0.9441	B001E4KF60	A3SGXHTAUHURGW	celmartian	1	
1	2	0.079	0.853	0.068	-0.1027	B00813GRG4	A1D87FRZCVE5NK	dl pa	0	

bulunan bu vadersler elde tutularak diğer dizini güncellenecektir. Burada kolon adını yeniden “id”

şeklinde adlandırıldı.Ve dataframe ile birleştirilmiştir. Ve kod bloğu çalıştırılırsa ortaya sentiment score ve metadata çıkacaktır.

compound değeri daha düşük değerlerde seyredecektir. Bu noktada plt.show denilerek bu çizgili grafik gözlemlenebilir.

```
fig, axes = plt.subplots(1, 3, figsize=(12, 3))
sns.barplot(data=vaders, x='Score', y='pos', ax=axes[0])
sns.barplot(data=vaders, x='Score', y='neu', ax=axes[1])
sns.barplot(data=vaders, x='Score', y='neg', ax=axes[2])
axes[0].set_title('Positive')
axes[1].set_title('Neutral')
axes[2].set_title('Negative')
plt.tight_layout()
plt.show()
```



Daha öncede bahsedildiği gibi bir yıldızlı bir inceleme daha düşük compound değerine sahiptir aynı şekilde beş yıldızlı bir inceleme daha yüksek bir compound değerine sahiptir. Aslında beklenen sonuç compound değeri ne kadar olumlu olursa buda daha fazla metnin yada yorumun da olumlu olduğunu gösterir. Ve ayrıca sadece compound değerine bu şekilde bakılması yerine compound değerini parçalanabilir. Her biri için pozitif, nötr ve negatif skorlar incelenebilir. Bunun için bir üst kod bloğunda ki gibi sns.barplot, plt.subplot metotlarından yararlanılacaktır. Burada plt.subplot için (1,3) şeklinde 1'e 3'lük bir ızgara oluşturulacaktır. İlk kısımda pozitif , ikinci kısımda nötr ve son olarak üçüncü kısımda negatif için değerlendirmeler görselleştirilecektir. Set_title'ları için sırasıyla aynı isimlendirmeler kullanılmıştır.

Yıldızlar açısında puan daha yüksek olduğu için pozitif daha yüksektir. Nötr bir tür düzdür. Ve negatif aşağıya doğru eğilimlidir. Yıldız skorunun daha yüksek olmasıyla compound değeri daha az negatif olur.

6-Grafiksel Vader ve Sonuçları Değerlendirmesi



Bulunan verilerin gerçek ile uyumlu olup olmadığını görsel olarak gözlemlemek için seaborn kütüphanesi kullanılarak bir çizgili bir grafik oluşturulacaktır. Burada x ekseninde herhangi bir yemek için yapılan değerlendirme skoru y ekseninde ise compound değeri konumlandırılmıştır.

Bu noktada eğer yorumlayıcılar 5 puanlık bir skor verdiler ise compound değeri olumlu yönde ilerleyecektir. Ancak daha düşük skorlar da ise

7-Roberta Önceden Eğitilmiş Modeli

```
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
```

Bu noktada vader modeli yorumlardaki cümleler ve puanlamalar için her kelimeye ayrı ayrı incelendi. Ve her biri için ayrı puanlama yapıldı ancak insanlar konuşma dilinde bağlaçlar veya mecazi anlamlar kullanabilmektedir. Ve bu mecazi anlamlar bazen model tarafından olumsuz bir kelime olumlu veya olumlu bir kelime olumsuz olarak puanlandırılmaktadır. Vader model konuşma dilindeki bazı kelimeler arası bağlantıları da kurulmamaktadır ve bu tür ilişkiyi anlayamamaktadır.

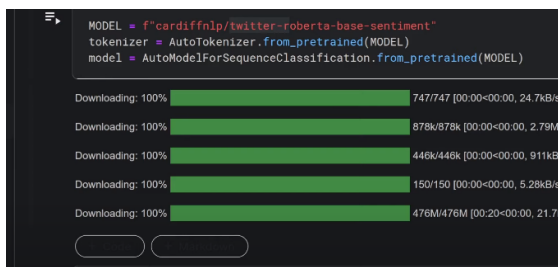
Bu yüzden son zamanlar “transformers” tabanlı derin öğrenme modelleri çok popüler hale gelmektedir. Bunun en önemli nedeni vader gibi mecazi anlam veya kelimeler arası bağlantı kuramayan modellerin bu eksini kapatıyor olmasından kaynaklanmaktadır. Şimdi bu popüler modellerden biri olan “hugging face” modeli kullanarak aynı veri seti için aynı işlemler yapılacak ve sonuçlar karşılaştırılacaktır.

Bu modelleri daha kolayca kullanabilir hale getirebilmek için transformers kitaplığına modele dahil edilmiştir.

Ve daha sonra doğal dil işleme kütüphanelerinden biri olan “nltk” kütüphanesi de modele dahil edilmiştir.

Bu noktada auto modeli de proje içine import edilmiştir..

Ve son olarak spacy kütüphanesinden aktivasyon için softmax metodunu da dahil edilmiştir.



Tam olarak yapmaya çalıştığımız, duyarlılık için bir dizi veri üzerinde önceden eğitilmiş çok özel bir model çekilecektir. Bu hugging face ile sağlanmaktadır ve otomatik sınıflandırılmış model yöntemlerinde çalıştırıldığında ve önceki eğitilmiş model yüklendiğinde, depolanmış

model ağırlıklarına göre işlemler yapılacaktır. Bu model eğitilirken twitter yorumları üzerinden eğitilmiştir. Aslında bu noktada güzel olan kısım bu modeli eğitmemiz gerekmiyor. Sadece var olan veri seti üzerinden çalıştırılacak ve sonuçlar elde edilecektir.

```
# VADER results on example
print(example)
sia.polarity_scores(example)

This oatmeal is not good. Its mushy, soft, I don't like it. Quaker Oats is the way to go.
{'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.3448}
```

Bu kod bloğunda da görüldüğü üzere bu model için daha önce değişken olarak atılan example burada kullanılacaktır. Bu example'da yulaf yorumu yapılmıştır.

Eski model sonuçları yukarıdaki gibi bulunmuştur. Bunlar vader modelinin sonuçlarıdır. Şimdi bunu roberta modeli ile karşılaştırmak için roberta modelinin kod bloğu oluşturulacaktır.

```
encoded_text = tokenizer(example, return_tensors='pt')
output = model(**encoded_text)
scores = output[0][0].detach().numpy()
scores = softmax(scores)
scores_dict = {
    'roberta_neg': scores[0],
    'roberta_neu': scores[1],
    'roberta_pos': scores[2]
}
print(scores_dict)

{'roberta_neg': 0.9763551, 'roberta_neu': 0.028687457, 'roberta_pos': 0.0029573673}
```

İlk olarak tokenizeri çekilecektir. İkinci olarak return_tensors='pt' şeklinde kodlanan kısımda yorum metnini alıyor ve modelin anlayacağı bir şekilde kaydediyor.

Daha sonra bu kaydedilen metni yada model tarafından anlaşılması için dönüştürülmüş verileri modeli dahil ederek çalıştırılacaktır. Çalıştırıldıktan sonra tensors olmaktan çıkarılıp bir numpy haline dönüştürülecektir. Böylece bu kaydedilmesi yada saklanması daha kolay olacaktır. Ve son olarak bu saklanan değerleri softmax fonksiyonunu uygulanacaktır. Şimdi ise bu çıktılar yazdırılır ise üç farklı değer ile karşılaşacaktır. Bunlar temelde negatif, nötr ve pozitif olarak değerlendirilmektedir. Roberta modeline göre çoğu yorumun olumsuz olduğunu görülmektedir. Ve bu veri seti incelenildiğinde daha mantıklı bir sonuç olduğu söylenebilir.


```
[26]: sia.polarity_scores(example)

[26...] {'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}

# Run the polarity score on the entire dataset
res = {}
for i, row in tqdm(df.iterrows(), total=len(df)):
    text = row['Text']
    myid = row['Id']
    res[myid] = sia.polarity_scores(text)

100% 500/500 [00:00<00:00, 893.78it/s]
```

8-Roberta Model vs Vader Modeli

Bu noktada iki model karşılaştırıldığında Roberta modelinin Vader modeline göre ne kadar daha güçlü olduğu görülmektedir.Şimdi bu modeli daha önce Vader modelinde de yaptığımız gibi tüm veri seti için çalıştırılacak olur ise bu sayede karşılaştırılması daha kolay olmaktadır.Bunu için daha öncede yaptığımız kod satırı “polarity_scores_roberta(example)” bu kod örnek alıyor ve hepsini for in döngüsü ile çalışacaktır ve tüm verilerin sonuçları elde edilecektir.

Bu kod blogu ile 500 örneğin tümü üzerinden çalıştırılacaktır.Bu çalışma bittiğinde 2 örnek için kırıldığı görülmektedir.Bu boyutu cpu üzerinden çalıştırıldığından daha fazla örnek cpu daha fazla zorlayacaktır.

```
vaders = pd.DataFrame(res).T
vaders = vaders.reset_index().rename(columns={'index': 'Id'})
vaders = vaders.merge(df, how='left')
```

Bu noktadan sonra bu iki model de karşılaştırılması için yukarıdaki kod blokları çalıştırılmıştır.

```
results_df = pd.DataFrame(res).T
results_df = results_df.reset_index().rename(columns={'index': 'Id'})
results_df = results_df.merge(df, how='left')
```

Yukarıdaki kod bloğunda Vader ve Roberta modelleri için sonuçları karşılaştırılabilir. Aslında seaborn kütüphanesinden yararlanılarak daha güzel bir görsel ile bu karşılaştırmayı kolaylaştırılabilir.Bu yüzden pairplot metodu kullanılarak gösterilecektir.

```
results_df.columns

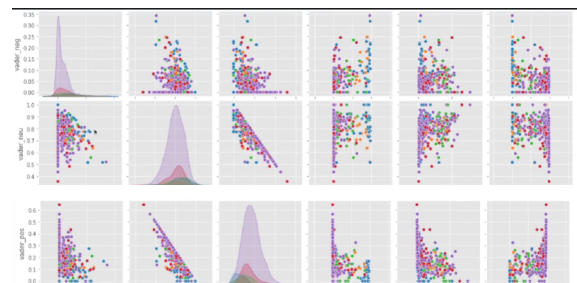
[76...] Index(['Id', 'vader_neg', 'vader_neu', 'vader_pos', 'vader_compound', 'roberta_neg', 'roberta_neu', 'roberta_pos', 'ProductId', 'UserId', 'ProfileName', 'HelpfulnessNumerator', 'HelpfulnessDenominator', 'Score', 'Time', 'Summary', 'Text'], dtype='object')
```

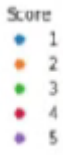
Burada “result_df.columns” kodu ile sütun isimleri görülmektedir. Ancak biz grafikler de Vader ve Roberta için negatif, nötr ve pozitif değerlerine göre karşılaştırmalar yapılacağından bu değerler seçilmiştir.

```
sns.pairplot(data=results_df,
             vars=['vader_neg', 'vader_neu', 'vader_pos',
                  'roberta_neg', 'roberta_neu', 'roberta_pos'],
             hue='Score',
             palette='tab10')
plt.show()
```

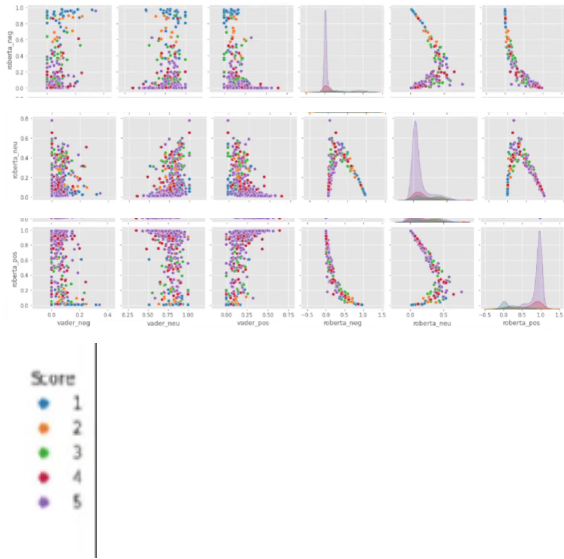
Roberta ve Vader için sütun değerleri belirlendi. Ayrıca daha görsel olması için palette değişkeni ile renklendirme işlemi de yapılmıştır.Ton rengi 1 ile 5 yıldız puanlarına göre değişecektir.

9-Vader Modeline İlişkin Grafiksel Sonuçlar ve Değerlendirmesi





10-Roberta Modeline İlişkin Grafiksel Sonuçlar ve Değerlendirmesi



modelinin tahminlemesinin daha gerçeğe yakın olduğunu göstermektedir.

```
results_df.query('Score == 1') \
.sort_values('roberta_pos', ascending=False)[0].Text.values[0]
```

'I felt energized within five minutes, but it lasted for about 45 minutes. I paid \$3.99 for this drink. I could have just drunk a cup of coffee and saved my money.'

Bu noktadan sonra beş yıldızlı incelemeleri bilindiğine göre tek yıldızlı incelemeleri de bulunabilir. Bunun için puan değeri 1'e eşit olanlar sorgulanacak ve çıkan sonuçların metni alınacaktır. Ve bunlar içinde en yüksek değerleri yazdırılacaktır.

Model tarafından olumlu olduğu söylenen ancak gerçek eleştirmenin verdiği puana göre bir puan olan bir metin " i felt energized within 5 minutes but it lasts about 45 minutes i paid 3.99 for this drink i should have just drunk a cup of coffee and saved my money "" bu metin de görüldüğü gibi cümle olumlu başlıyor ancak olumsuz bitiyor ve bu da modelin kafasını karıştırabilir. Ancak görüldüğü gibi en yüksek bir puanlıklardan biri olan bu metin için en olumlu ama sonuçta olumsuz bir cümle olduğunu model etkileyici bir şekilde tespit etmektedir.

```
results_df.query('Score == 1') \
.sort_values('vader_pos', ascending=False)[0].Text.values[0]
```

'So we cancelled the order. It was cancelled without any problem. That is a positive note...'

11-İki Model sonuçlarının değerlendirilmesi

Her değer arasındaki fark kolayca görülmektedir. Bu değişkenler ile karşılaştırma yapılmaktadır. Bu arada ortaya çıkan ilk şey vader için beş yıldızlı incelemelerin morumsu bir renk alması, Roberta modeli için ise olumlu eleştirilerin daha sağda olmasıdır. Ayrıca Roberta modeli ile Vader modeli arasında bazı korelasyonlar olduğunu da görülmektedir. Vader modelinde tahmin edilen her değer için pozitif, nötr ve negatif puanları Roberta modeline göre daha az gerçeği yansıtmaktadır. Bu da Roberta

Aynı şeyi Vader modelini üzerinden yapılacak olur ise en olumlu olana bakılır ise bir yıldızlı derecelendirme için "" so we canceled the order. it was canceled without any problem. That is a positive note.. "" olumlu görünüyor ama olumsuz bir yorum ve biraz alaycı ancak model bunu anlamıyor. Bu cümle için tüm kelimelerin ayrı ayrı puanlamasını yapmaktadır. Ve buda çok gerçekçi sonuçlar oluşturmuyor.

```
results_df.query('Score == 5') \
.sort_values('roberta_neg', ascending=False)[0].Text.values[0]
```

'this was sooooo delicious but too bad i ate em too fast and gained 2 pds! my fault'

```
results_df.query('Score == 5') \
.sort_values('vader_neg', ascending=False)[0].Text.values[0]
```

'this was sooooo delicious but too bad i ate em too fast and gained 2 pds! my fault'

Şimdi de "" 5 Yıldızlı Olumsuz Duygulu Metin" bunun için ilk olarak Roberta modelinde çalışma

yapılır ise skoru beş olarak seçilmeli ve olumsuz bir yorum görünmektedir. “” this was so delicious but too bad i ate them too fast and gained 2 pds! My fault okay”” bu cümle de aslında olumsuz bir duygu ama cümle genelinde olumlu bir cümledir. Aynı şeyi Vader Modeli için de çalıştırılır ise aynı metin sonucu bulunmuştur. Bu noktada iki model de iyi sonuçlar çıkardığı söylenebilir.

SON

KAYNAKÇALAR

- 1- <https://docs.python.org/3/library/statistics.html>
- 2- <https://towardsdatascience.com/an-introduction-to-statistical-analysis-and-modelling-with-python-ef816b67f8ff>
- 3- <https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a>
- 4- <https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews>
- 5- <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment?text=I+like+you.+I+love+you>
- 6- <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>
- 7- <https://www.analyticsvidhya.com/blog/2021/06/vader-for-sentiment-analysis/>
- 8- <https://seaborn.pydata.org/tutorial.html>
- 9- https://numpy.org/doc/stable/user/absolute_beginners.html
- 10- <https://monkeylearn.com/sentiment-analysis/>
- 11- www.kdnuggets.com
- 12- <https://dergipark.org.tr/tr/>