DATA SCIENCE

Pulling amazon.com Product Reviews and Predicting Scores

Ömer Kılıççeker 180101016

Emre Büyüktaş 180101009

Yunus Ziya Arıkan 180101020

CONTENTS

TABLE

INTRODUCTION

We first started with creating a dataset. We used the web scraping method to get product comments and product ratings. We completed this process by getting 10000 comments from 20 products from the Amazon electronics category.

As the second step, we went through the preprocessing steps of the dataset we obtained. Here, we cleaned our data from punctuation marks, if there is more than one of the same sentence, stopwords and extra spaces in words. We converted all capital letters in the dataset to lowercase.

In the last step, we trained the machine with the dataset we had. At this stage, we used the sklearn library. After the preprocessing eliminations, we used 7767 of the 8443 data remaining for training and 676 of them for testing. In total, our machine has a correct prediction rate of 76%. While the average rating of the data set we have is 4.388, our machine knows it as 4.383.

CREATING DATASET

USED LIBRARY

- *Pandas*
- *request*
- *BeautifulSoup*
- *Datetime*

```
u="https://www.amazon.com.tr/Samsung-NVMe-Dahili-
Kat%C4%B1/product-
reviews/B08GS7748F/ref=cm_cr_getr_d_paging_btm_prev_{btmNext}?
ie=UTF8&reviewerType=all_reviews&pageNumber={pageNumber}".form
at(btmNext=x,pageNumber=x)
```

We take the product links and dynamically give the btnNext and page Numbers with the page numbers, and we get as many pages as we want.

As the last operation, we save the data we have taken to the csv file.

```
save_name=f"{datetime.now().strftime('%Y-%m-%d-%m')}.csv"out.t
o_csv(save_name,index=False,mode='a')
```

```
df['stars'].value_counts()
✓ 0.3s
5,0    6699
4,0    1073
1,0     704
3,0     449
2,0     273
```

PREPROCESSING

USED LIBRARY

- *Pandas*
- *Numpy*
- *Nltk*
- *Re*

✓ *First, we delete multiple duplicate comments.*

```
df.drop_duplicates(subset ="body",keep ='first', inplace =
True)
```

✓ *Secondly, we clean Turkish stopwords.*

```
stopword = set(stopwords.words("turkish"))

def stopwords_temizleme (metin):

    return " ".join([kelime for kelime in str(metin).split()
if kelime not in stopword])

df["body"] = df["body"].apply(lambda metin :
stopwords_temizleme(metin) )
```

✓ *Third step we clean up punctuation marks*

```
df['body'] = df['body'].apply(lambda
x:re.sub("[,\.!?:()']",'',str(x)))
```

✓ *Forth step we clean multiple spaces in the comments.*

```
df['body'] = df['body'].apply(lambda x: str(x).strip())
```

✓ *Step five we change uppercase letters to lowercase*

```
df['body'] = df['body'].apply(lambda x: str(x).lower())
```

Finally, we save the pre-processed dataset to a file named "Cleaningdataset"

MACHINE TRAINING

USED LIBRARY

- *Sklearn*

✓ *We use 7767 of the data we have for training and 676 for testing.*

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state=0)
```

Machine results like this.

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
1,0	0.53	0.57	0.55	54
2,0	0.25	0.16	0.19	19
3,0	0.29	0.25	0.27	32
4,0	0.35	0.31	0.33	74
5,0	0.87	0.90	0.89	497
accuracy			0.76	676
macro avg	0.46	0.44	0.45	676
weighted avg	0.74	0.76	0.75	676

The average rating score of the dataset we have is the rating score that the machine predicts after training the machine.

	REAL RATING AVERAGE	MACHİNE ESTIMATION
Average	4.3888	4.3836

When we give a group of comments with an average of three that we do not use in the dataset, our machine estimates it as 3.3 .

References

https://scikit-learn.org/stable/index.html

https://opensource.com/article/21/9/web-scraping-python-beautiful-soup

https://realpython.com/python-nltk-sentiment-analysis/

https://www.veribilimiokulu.com/derin-ogrenme-ile-duygu-analizi/

Ümit Hocanın kodları

https://medium.com/data-science-lab-spring-2021/amazon-review-rating-prediction-with-nlp-28a4acdd4352

https://medium.com/t%C3%BCrkiye/google-playstore-uygulamalar%C4%B1n-ne-kadar-y%C4%B1ld%C4%B1z-ald%C4%B1%C4%9F%C4%B1n%C4%B1n-tahmini-7efbbc02d4d2