Ratings Prediction for Reviews

NAME OF THE PROJECT

Salman Pradhan

Submitted by

Salman Pradhan

ACKNOWLEDGMENT

I wish to thank Flip Robo Technologies for assigning this project to me and thanks to Mr.Shubham Yadav sir for his valuable technical support on this project.

INTRODUCTION

- ## Business Problem Framing

We have to build an application in order to predict the ratings by seeing the reviews given by customers.So that we can easily identify how much is our positive reviews and how much is our negative value of that specified product.

- ## Conceptual Background of the Domain Problem

Created model can be used to predict ratings of reviews, it might be a good tool for online shopping sites such as Flipkart.com,Amazon.com etc and manufacturer companies who might look into their product ratings and reviews sothat they can make their investment according to demand of customers , which might help them to save time and earn more profits.

- ## Review of Literature

Now a days people buy more products from online sites,So by using this model,rating prediction and positive or negative review prediction of a product is easy and less time consuming.

- ## Motivation for the Problem Undertaken

DataScience help us to make predictions at areas like health sectors, auto industry, education, media etc. For our project we decided to implement Prediction of Rating of reviews.we have to create a model which will help online sites and manufacturer of a product if they have to modify their product or not according to cusomers requirement.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

The prediction will base on the dataset which is scrapped from flipkart's technical product. Scrapping dataset contains reviews and ratings of different products.

- Data Sources and their formats

To collect data we need to scrape the reviews of different laptops, Phones, Headphones, smart watches, Professional Cameras, Printers, Monitors, Home theater, Router from Flipkart.com.

Basically, we need these columns-

1) reviews of the product.

2) rating of the product.

We scrap reviews and Ratings of different products by using selenium.

```
#importing all libraries
import pandas as pd
import selenium
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException,StaleElementReferenceException
import time
```

```
driver=webdriver.Chrome(r"chromedriver.exe")
time.sleep(4)

url = "https://www.flipkart.com/"
driver.get(url)
time.sleep(3)
```

After scrapping data we save the dataset in a csv file named "flipkartreviews.csv".

We have one data set ie "**flipkartreviews.csv"** which contains reviews and ratings of different products from flipkart.comur model.
In this dataset we have 19188 rows and 2 features.Our predicted output feature is 'Ratings'.
Here's a brief version of what we'll find in the data description file.

```
df =pd.read_csv('flipkartreviews.csv')#Load dataset
df=df.drop("Unnamed: 0",axis=1)
df
```

	Full_Reviews	Ratings
0	Everything works with very fast speed due to i...	5
1	Good morning.\nIt is a nice product.\nI used i...	5
2	Performance of the laptop is good. Login scree...	3
3	Flipkart just took 2 days to deliver the produ...	5
4	Got it 40k,and at that particular price it's j...	5
...	...	...
19183	Very poor	1
19184	It was nice to wear also the feature were amaz...	4
19185	Good prodect	3
19186	It was nice to wear also the feature were amaz...	4
19187	Good prodect	3

19188 rows × 2 columns

1. The dataset has two columns one is Full_Reviews and other one is our target variable Ratings.
2. Full_Reviews column is object type and Ratings column is integer type.
3. There is no null values present in the dataset.

- Data Preprocessing Done

1. First we have to remove all other words and symbols except alphates (a-z and A-Z).
2. We have to make all the words lower.
3. We have to split words from different sentences.
4. Then we have made a list named stoplist which contains all the repetation of words whose impact is neglisible while predicting the output.
5. We have to remove all the punctuations that present inside the sentence.
6. After removing all these unnecessary noise ,we have converted words to their base form by using WordnetLemmatizer.
7. Almost 750000 unnecessary words are removed.

Data cleaning

```
ps = PorterStemmer()
wordnet=WordNetLemmatizer()
corpus = []
for i in range(0,19188):
    review = re.sub("[^a-zA-z\s']", ' ', df['Full_Reviews'][i])
    review = review.lower()
    review = review.split()
    review = [wordnet.lemmatize(word) for word in review if word not in stoplist if word not in punc]
    review = ' '.join(review)
    corpus.append(review)
```

- State the set of assumptions (if any) related to the problem under consideration

As the dataset is imbalanced we have to choose our best model according to recall,precision or f1 score.

- Hardware and Software Requirements and Tools Used

1.NumPy: Base n-dimensional array package
2.Matplotlib: Comprehensive 2D/3D plotting
3Seaborn: For plotting graph
4.Pandas: Data structures and analysis
5.SciPy: Fundamental library for scientific computing
6.Nltk:for converting text to vectors
7.String: For data cleaning
8.Re: For data cleaning and remove noise from dataset
9.Scikit-learrn: provides a range of algorithm

```
# importing libraries

import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
import matplotlib.pyplot as plt
import seaborn as sns
import string
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

These are the approaches to solve this problem
1.Understand business problem
2.Scrapping data from flipkart.com using selenium
3.Data analysis
4.Data Cleaning
5.Data preprocessing
6.Model Building
7. Model Evaluation
8. Selecting the best model

- Testing of Identified Approaches (Algorithms)

Algorithms used for this dataset are

1.MultinomialNB

2.RandomForestClassifier

3.DecisionTreeClassifier

4.LogisticRegression

5.AdaBoostClassifier

6.GradientBoostingClassifier

7.XGBClassifier

8.SVC

- Run and Evaluate selected models

We have to choose best random_state for our model.

```
# Train Test Split
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.model_selection import train_test_split
# Training model using Naive bayes classifier

from sklearn.naive_bayes import MultinomialNB
mn=MultinomialNB()
maxacc=0
maxrs=0
for i in range(50):
    x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    mn.fit(x_train,y_train)
    y_pred=mn.predict(x_test)
    acc=accuracy_score(y_test,y_pred)
    if acc>maxacc:
        maxacc=acc
        maxrs=i
print("best accuracy score is",maxacc,"for random_state",maxrs)
```

```
best accuracy score is 0.6675351745700886 for random_state 42
```

Then we have to use random_state 42 on other algorithms to see where my accuracy is good.

```
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
```

```
model=[SVC(),DecisionTreeClassifier(),LogisticRegression()]
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=42)
```

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
for i in model:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    print(i)
    print(accuracy_score(y_test,y_pred))
    print(classification_report(y_test,y_pred))
    print(confusion_matrix(y_test,y_pred))
```

```
SVC()
0.7451797811360084
              precision    recall  f1-score   support

           1       0.74      0.93      0.82       325
           2       0.89      0.18      0.30        95
           3       0.59      0.21      0.32       135
           4       0.76      0.32      0.45       370
           5       0.75      0.97      0.84       994

    accuracy                           0.75      1919
   macro avg       0.75      0.52      0.55      1919
weighted avg       0.75      0.75      0.70      1919

[[302   1   1   2  19]
 [ 52  17   8   5  13]
 [ 33   0  29  13  60]
 [  9   0  10 119 232]
 [ 12   1   1  17 963]]
```

```
DecisionTreeClassifier()
0.7107868681605003
              precision    recall  f1-score   support

           1       0.75      0.84      0.79       325
           2       0.53      0.32      0.39        95
           3       0.41      0.33      0.36       135
           4       0.57      0.39      0.46       370
           5       0.76      0.88      0.82       994

    accuracy                           0.71      1919
   macro avg       0.61      0.55      0.57      1919
weighted avg       0.69      0.71      0.69      1919

[[273  12  14  10  16]
 [ 40  30   6   4  15]
 [ 23   5  44  17  46]
 [  9   5  21 143 192]
 [ 17   5  23  75 874]]
LogisticRegression()
0.7076602397081814
              precision    recall  f1-score   support

           1       0.71      0.90      0.79       325
           2       0.64      0.09      0.17        95
           3       0.50      0.16      0.25       135
           4       0.61      0.23      0.34       370
           5       0.72      0.95      0.82       994

    accuracy                           0.71      1919
   macro avg       0.64      0.47      0.47      1919
weighted avg       0.68      0.71      0.65      1919

[[292   3   2   5  23]
 [ 56   9   9   5  16]
 [ 39   0  22  16  58]
 [ 11   0   9  86 264]
 [ 13   2   2  28 949]]
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

```
rf=RandomForestClassifier()
rf.fit(x_train,y_train)
y_pred=rf.predict(x_test)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

```
0.7550807712350183
              precision    recall  f1-score   support

           1       0.76      0.91      0.83       325
           2       0.85      0.29      0.44        95
           3       0.67      0.29      0.40       135
           4       0.80      0.34      0.48       370
           5       0.75      0.97      0.84       994

    accuracy                           0.76      1919
   macro avg       0.77      0.56      0.60      1919
weighted avg       0.76      0.76      0.72      1919

[[296   2   3   4  20]
 [ 46  28   5   3  13]
 [ 32   3  39   8  53]
 [  4   0   7 126 233]
 [ 13   0   4  17 960]]
```

- Key Metrics for success in solving problem under consideration

As the dataset is imbalanced,I use cross_val_score validation to check overfitting of models.Precision,Recall and f1 score of RandomForestClassifier is good compared to all other algorithms we used in the dataset.I conclude that RandomForestClassifier is the best model for this dataset.

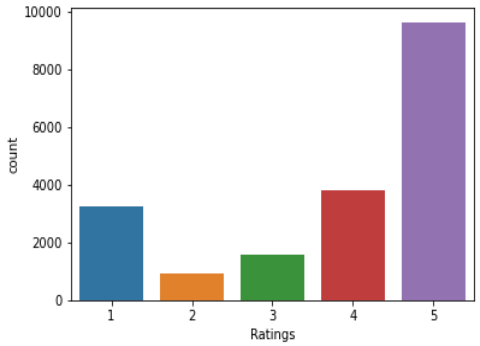
- Visualizations

Count of Ratings

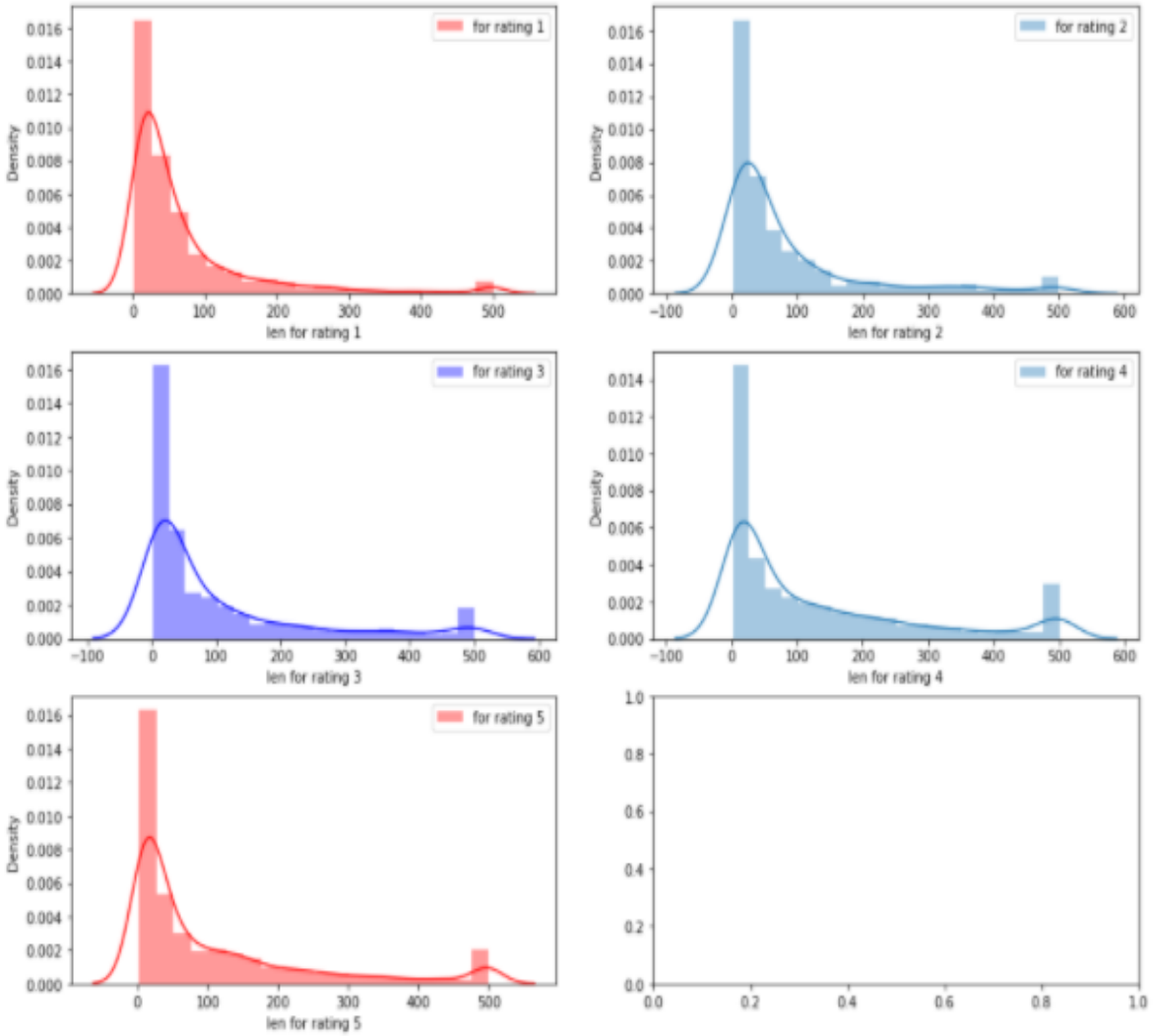
```
sns.countplot(df['Ratings'])
print ('Ratings counts','\n',df.Ratings.value_counts())
```

```
Ratings counts
 5    9639
4    3800
1    3246
3    1570
2     933
Name: Ratings, dtype: int64
```

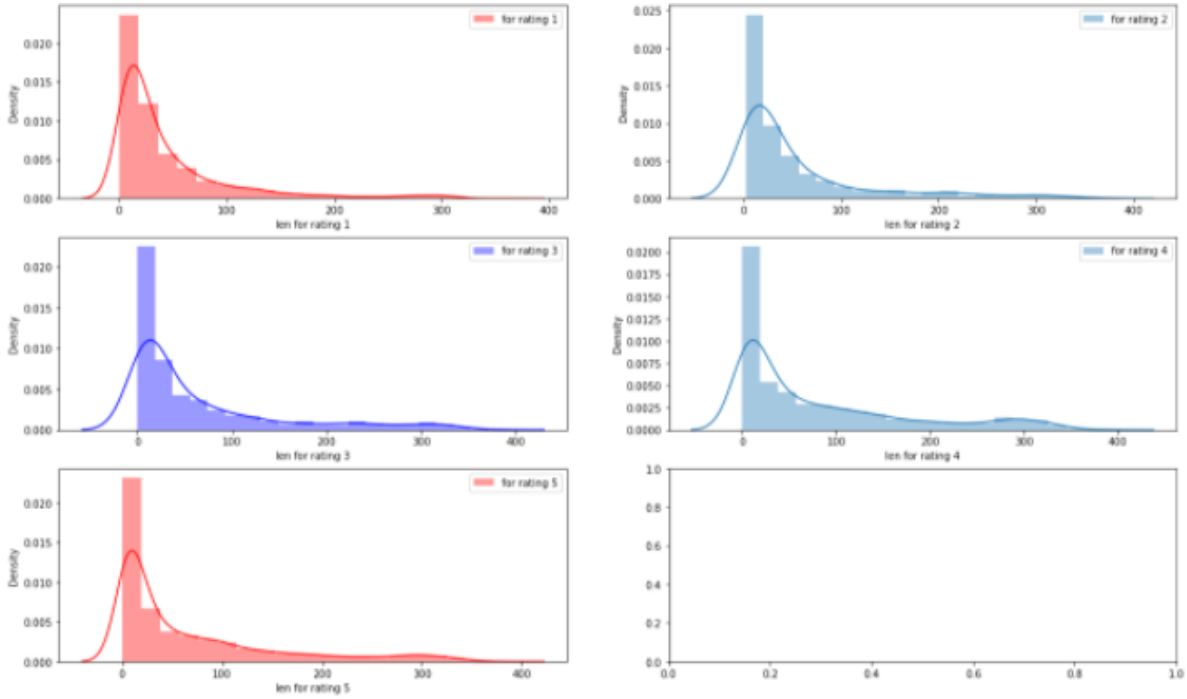
```
C:\Users\Lenovo\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword
arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```



Distribution of length before cleaning



Distribution of length after cleaning



- Interpretation of the Results

  Comparing recall,precision,f1score of all models,we choose RandomForestClassifier is our best model.It gives accuracy score of approximately 75% and after hyperparameter tuning we get an accuracy score of 75.61%.

CONCLUSION

- Key Findings and Conclusions of the Study
  1. First we scrap Reviews and Ratings data of different technical products from flipkart.com using selenium webdriver.After scrapping,we save this dataset in a csv file named “flipkartreviews.csv”.

2. Then we preprocess data by cleaning duplicates,noise and unnecessary words which are not helpful for this project.
3. Then we convert text into vectors by using tfidfvectorizer as we know our ml model understands only integers.
4. We choose our best random_state.Then we build our model.
5. Though our dataset is imbalanced,we clearly see that f1 score of RandomForestClassifier is good compared to all other algorithms.
6. So we check croos_val_score of this model to check if it is overfitting or not.
7. We confirm and conclude that this is the best model.we hypertune it by using GridSearchCV.After hypertuning we get an accuracy score of 75.61% from 75%.
8. We save the model by using joblib.

- Learning Outcomes of the Study in respect of Data Science

RandomForestClassifier is the best model for this dataset.

- Limitations of this work and Scope for Future Work

As we know data is increasing in every seconds in our day today life.So more the data beter the model.

If we make this dataset for sentiment analysis,we choose ratings 3 or more as our threshold for being helpful reviews or good or positive reviews and below 3 we choose reviews is not helpful or bad reviews or negative reviews.

For example: two people give their reviews on a product as ‘a nice product’.But their ratings are ‘4’ and ‘5’ respectively.This is where the model fails to predict whether to choose rating ‘4’ or’5’

Due to increase in data in our daily basics,this model can be used to predict ratings of reviews.It might be a good tool for online shopping sites and manufacturer companies who may predict their customers ratings so that they can make their investment according to the demand of customers,which migt help them to save time and earn more profits.