RATING PREDICTION PROJECT

Submitted by:

ANITA THAPA

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my SME Swati Ma’am for giving me an opportunity to work on the Rating Prediction Project. During my research on this project internet sites like towardsdatascience, medium, Amazon.in, flipkart.com, scikit learn and Stack Overflow helped me a lot in the project.

INTRODUCTION

- ## Business Problem Framing

  The project is based on the reviews and ratings of the technical products like smartphones, printers, laptops etc. available on the ecommerce websites. The rating prediction model is to be built based on the reviews given by the customers.

- ## Conceptual Background of the Domain Problem

  Reviews and Ratings are important as they help to understand the customers feedback in real time. Building a model to predict the ratings based on the reviews available on the technical product will help to give a clear idea about the customers feedback after using those products. This will ultimately help companies to improve upon their products for their growth in this competitive world.

- ## Review of Literature

  The reviews scrapped from ecommerce websites shows that customers used emojis, numeric along with the long description about the product which sometimes become difficult to analyse the rating for the product as its very time consuming. Stackoverflow website helped a lot during data analysis.

- ## Motivation for the Problem Undertaken

  The main objective was to analyse the reviews and then predict the rating for the products. This project really helped to understand the importance of rating and review for the companies as it helps to understand the customer choices well onto which companies can work for achieving growth.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

The data was present in excel format so by using pandas the excel file was read. Then the dataset details were analysed using. shape () to find the number of rows and columns, using dtypes() we found the dataset datatype like object and numeric datatype. Dataset has 4749Rows and 3 Columns; 2 columns are numeric and 1 column is object type and the Target column is Rating. Dataset is a Classification model.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42405 entries, 0 to 42404
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  42405 non-null  int64 
 1   Rating      42405 non-null  int64 
 2   Review      42376 non-null  object
dtypes: int64(2), object(1)
memory usage: 994.0+ KB
```

- Data Sources and their formats

What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data. Data was scraped from websites like Amazon.in and Flipkart.com using Selenium library in Python and saved as excel file. Then data was imported in python using pandas . Information of dataset is found using datasetname.info(). As per the information each column has count of 42405 rows and 2 columns.

- Data Preprocessing Done

Review column had 29 Nan values so they are dropped along with Unnamed: 0 column. Then the data in Review column had punctuation, uppercase and lowercase, emoji and numeric terms which were corrected using Natural Language Processing.

```
# Removing Nan Values:

def preprocessore_inputs(data):
    data=data.copy()
    missing_reviews=data[data['Review'].isna()].index
    data=data.drop(missing_reviews, axis=0).reset_index(drop=True)
    return data
df1=preprocessore_inputs(df)
print(df1)
```

```
       Unnamed: 0  Rating                                             Review
0            6365       5  It's really nice product worth to buy and I'm ...
1            2455       5                                          Very nice
2           14681       5                                               good
3            2443       5                      Nice product and very comfortable 👌
4            2445       5              Very lightweight , thin , classic product.
...           ...     ...                                                ...
42371       53823       1                                       Third grade
42372       53824       1                                         Not Happy
42373       53825       1                                    Bat quality.....
42374         926       1  cartridges are very low. After printing 10-15 ...
42375       53855       1  This such a bad printer. Stopped working after...

[42376 rows x 3 columns]
```

```
import re
import nltk
import string
```

```
def get_clean(x):
    x=str(x).lower().replace('\\','').replace('_',' ')
    x=re.sub('\[.*?\]','',x)
    x=re.sub('[%s]'%re.escape(string.punctuation),'',x)
    x=re.sub('\w*\d\w*','',x)
    x=re.sub("(.)\\1{2,}","\\1",x)
    return x
```

```
df1['Review']=df1['Review'].apply(lambda x: get_clean(x))
```

```
df1
```

	Rating	Review
0	5	its really nice product worth to buy and im ha...
1	5	very nice
2	5	good
3	5	nice product and very comfortable 👌

- Data Inputs- Logic- Output Relationships

  Input is the Review Column and output is Rating column. Rating is given on the based of the Reviews given by customers. 5 Star rating means very positive Reviews like Superb etc and 1 Star Rating means the most disliked product as it has reviews like very bad, don’t buy it etc. Input is related to output.

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

  Here no such assumptions are considered.

- Hardware and Software Requirements and Tools Used

  Libraries used were:

  1. Numpy : It is a Numerical Python library used for numerical computation like arrays in Python.

2. Panda: It was used for reading and converting excel/csv file into dataframe.

3. matplotlib: this library was used for plotting the graph in the EDA.

4. Seaborn: This library is also used for visualization as it has helped to plot different types of plots like countplot in the notebook.

5. nltk, re: Used for Natural Language Processing of review column.

6. sklearn.preprocessing and power transform: It was used to remove the skewness from the dataset.

7. Tfidf Vectorizer : It was used for Review column.

8. sklearn: This library was used to import train_test_split, metrics like accuracy score, classification report and also importing Algorithms likes Linear classification, Decision Tree classification, SV classification, Random Forest classification and K-Neighbors classification.

9. pickle: This library was used to save the final model.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  The Tfidf Vectorizer is used in Model building. As the model is Classification so Logistic Regressor library along with accuracy, cross validation score, confusion matrix was imported. Then 4 Algorithms were used which were Decision Tree Classification, SV Classification, Random Forest Classification and K-Neighbors Classification. Then after selecting best model, then the final model was deployed and the accuracy of model was 82% as it predicted values approximately equal to actual values.

Model Selection: ¶

```
#importing libraries for model selection

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
from sklearn.metrics import r2_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

```
# tfidf vectorizer is initialised
tfidf=TfidfVectorizer(ngram_range=(1,3),analyzer='char')
```

```
x=tfidf.fit_transform(df1['Review'])
y = df1["Rating"]
```

```
x.shape
```

```
3]: (42376, 10409)
```

- Testing of Identified Approaches (Algorithms)

  Algorithms used for the training and testing are:

  1. K-Neighbors Classifier
  2. Decision Tree Classifier
  3. SV Classifier
  4. Random Forest Classifier

- Run and Evaluate selected models

  Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

RandomForestClassifier

```
#Using GridsearchCV on RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from random import randint
from sklearn.datasets import make_classification
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()

forest_params = [{'max_depth': [5, 10, None]}]

clf = GridSearchCV(rfc, forest_params,scoring='f1_weighted',cv =9, n_jobs = -1, verbose = 3,refit = False)

clf.fit(train_x,train_y)

print(clf.best_params_)
```

```
Fitting 9 folds for each of 3 candidates, totalling 27 fits
{'max_depth': None}
```

```
#using best parameters:
rfc= RandomForestClassifier(max_depth=None)
rfc.fit(train_x,train_y)
rfc.score(train_x,train_y)
```

```
37]: 0.9248967551622419
```

```
#finding cross val score

rfscore = cross_val_score(rfc,x,y,cv=9)
rfs = rfscore.mean()
print('Cross Val Score:',rfs*100)
```

```
Cross Val Score: 81.33879214842692
```

```
#Predicting value
pred_rfc = rfc.predict(test_x)
from sklearn.metrics import accuracy_score, confusion_matrix
accuracy=accuracy_score(test_y,pred_rfc)
confusion=confusion_matrix(test_y,pred_rfc)
print("Accuracy of the model is: ",accuracy)
print("Confusion Matrix: ", confusion)
```

```
Accuracy of the model is:  0.8219679093912223
Confusion Matrix:  [[1537   68   40   32    8]
 [ 234 1070  297   44   29]
 [ 137  157 1210  137   25]
 [   6    5   37 1601   57]
 [   2    0   16  178 1549]]
```

```
#printing classification report
print(classification_report(test_y,pred_rfc))
```

```
              precision    recall  f1-score   support

           1       0.80      0.91      0.85      1685
           2       0.82      0.64      0.72      1674
           3       0.76      0.73      0.74      1666
           4       0.80      0.94      0.87      1706
           5       0.93      0.89      0.91      1745

    accuracy                           0.82      8476
   macro avg       0.82      0.82      0.82      8476
weighted avg       0.82      0.82      0.82      8476
```

- Key Metrics for success in solving problem under consideration

  Validation score, Accuracy, confusion matrix and the Classification report helped in building project.

- Visualizations

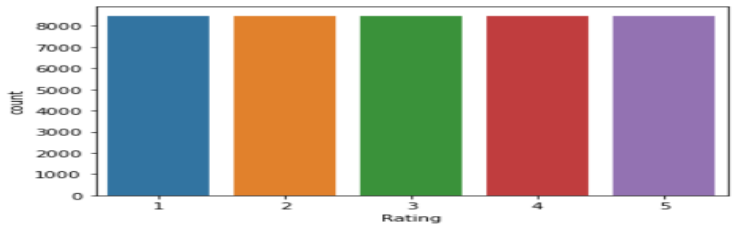
  Used only one plot which is catplot.

```
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
# Plotting Target Lable: Rating

ax=sns.countplot(x="Rating", data=df)
print(df["Rating"].value_counts())
```

```
5    8481
4    8481
3    8481
2    8481
1    8481
Name: Rating, dtype: int64
```



- Interpretation of the Results

  From the visualizations, pre-processing and modelling of the data it was interpreted that Reviews help in predicting Rating by the use of Natural Language Processing. As per the model the data can be predicted with 82% accuracy.

CONCLUSION

- Key Findings and Conclusions of the Study

  Key Findings from the problem is that the rating is important for companies for connecting with their consumers. Natural language Processing really helped to predict the rating based on the sentences written in reviews.

- Learning Outcomes of the Study in respect of Data Science

  During this project I learned about the Natural language Processing and how it can be used for sentiment analysis. The best algorithm is Random Forest Classifier as the classification report of it has higher precision value, f1 score and recall value as compare to other Algorithms.

- Limitations of this work and Scope for Future Work

  Study of Reviews and Ratings is very useful for all the businesses in today's world. This project is very much useful for Companies to connect with their consumer base so that companies can design and make changes as per their consumers requirement. The limitations are some consumers post negative reviews for defaming companies' reputation for that purpose more study needs to be done.