FLIP ROBO

Rating Price Prediction

Submitted by:

Matta Akhil Reddy

ACKNOWLEDGMENT

I would like to thank **FlipRobo Technologies** for providing me with this opportunity for writing the blog, and also in providing the guidance and constant support in providing necessary information

regarding the project, by providing their support in completing the project. I also want to thank my **SME Ms.Khushboo garg** for their instant response in solving the problems and providing the valuable information by addressing out our query in right time.

INTRODUCTION

Business Problem Framing

- We have a client who has a website where people write different reviews for technical products.
- Now they are adding a new feature to their website i.e. The reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars,3 stars, 4 stars, 5 stars.
- Now they want to predict ratings for the reviews which were written in the past and they don't have a rating. So, we have to build an application which can predict the rating by seeing the review.

Conceptual Background of the Domain Problem

Nowadays, a massive amount of reviews is available online. Besides offering a valuable source of information, these informational contents generated by users, also called User GeneratedContents (UGC) strongly impact the purchase decision of customers. As a matter of fact, a recent survey (Hinckley, 2015) revealed that 67.7% of consumers are effectively influenced by online reviews when making their purchase decisions. More precisely, 54.7% recognized that these reviews were either fairly, very or absolutely important in their purchase decision making. Relying on online reviews has thus become a second nature for consumers

Review of Literature

- E-commerce is one of the fastest growing segments in the Indian Economy.
- Though marked by high growth rate, the Indian e-commerce industry has been behind its counterparts in many developed and emerging economies, primarily due to a relatively low internet user base.
- In a study conducted by global management consultancy firm AT Kearney in 2015, there were only 39 million online buyers in India; a tiny fraction of the 1.2 billion who live in the country. However, increased technological proliferation combined with internet and mobile penetration, presents a favorable ecosystem for the development of e-commerce in India.

Motivation for the Problem Undertaken

- Many product reviews are not accompanied by a scale rating system, consisting only of a textual evaluation.
- In this case, it becomes daunting and time-consuming to compare different products in order to eventually make a choice between them.
- Therefore, models able to predict the user rating from the text review are critically important.
- Getting an overall sense of a textual review could in turn improve consumer experience.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

- There are in total 57389 rows and 3 columns of ratings and reviews are present in our dataset.

Data Sources and their formats

- We can observe that our dataset is quite imbalanced.

Rating	counts
5	35524
4	15438
3	2831
1	2397
2	1199

```
df['Product_description'] = df['Product_description'].apply(clean_text)
df
```

	Product_description	Product_Rating	Type_of_product
0	days usage laptop quite lightweight well desig...	4	Laptop
1	performance best better price range display ex...	5	Laptop
2	nice product new feature price range works rea...	4	Laptop
3	every thing gud laptop cons fell display aspec...	5	Laptop
4	working really well satisfied product battery ...	5	Laptop
...	...	...	...
57384	good performance good value	5	Phone
57385	good	5	Phone
57386	performance best part features money price best	5	Phone
57387	good condition mobile	3	Phone
57388	design super quality low price best phone	4	Phone

57389 rows × 3 columns

We have 3 columns states product description, rating, type of product

Data Preprocessing Done

We first looked for the null values present in the dataset. We noticed that there were no null values present in our dataset. Then we performed text processing. Data usually comes from a variety of sources and often in different formats. For this reason transforming your raw data is essential. However, this is not a simple process, as text data often contains redundant and repetitive words. This means that processing the text data is the first step in our solution. The fundamental steps involved in text preprocessing are, Cleaning the raw data Tokenizing the cleaned data.

Data Inputs- Logic- Output Relationships

This phase involves the deletion of words or characters that do not add value to the meaning of the text. Some of the standard cleaning steps are listed below:

- Lowering case
- Removal of special characters
- Removal of stopwords
- Removal of hyperlinks
- Removal of numbers
- Removal of whitespaces

Lowering Case

- Lowering the case of text is essential for the following reasons: The words, ‘TEXT’, ‘Text’, ‘text’ all add the same value to a sentence Lowering the case of all the words is very helpful for reducing the dimensions by decreasing the size of the vocabulary.

Removal of special characters

- This is another text processing technique that will help to treat words like ‘hurray’ and ‘hurray!’ in the same way.

Removal of stop words

- Stopwords are commonly occurring words in a language like ‘the’, ‘a’, and so on. Most of the time they can be removed from the text because they don’t provide valuable information.

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

- By looking into the target variable label we assumed that it was a Multiclass classification type of problem.
- We observed that dataset was imbalance so we will have to balance the dataset for better outcome.

Hardware and Software Requirements and Tools Used

- This project was done on Mac OS which has 8-core GPU
- Softwares used are Anaconda, jupyter notebook.
- The tools, libraries and packages we used for accomplishing this project are pandas, numpy, matplotlib, seaborn, wordcloud, tfidf vectorizer, smote, Gridsearchcv, joblib.
- Through pandas library we loaded our csv file into dataframe and performed data manipulation and analysis.
- With the help of numpy we worked with arrays.
- With the help of matplotlib and seaborn we did plot various graphs and figures and done data visualization.
- With wordcloud we got sense of loud words present in the dataset.
- Through tfidf vectorizer we converted text into vectors.
- Through smote technique we handled the imbalanced dataset.
- Through Gridsearchcv we tried to find the best parameters of
- random forest classifier.
- Through joblib we saved our model in csv format.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Pre-processing involved the following steps:

  - Removing Punctuations and other special characters
  - Removing Stop Words
  - Stemming and Lemmatising
  - Applying tfidf Vectorizer
  - Splitting dataset into Training and Testing

- Testing of Identified Approaches (Algorithms)
- The algorithms we used for the training and testing are as follows :-
  - Decision tree classifier
  - Kneighbors classifier
  - MultinomialNB
  - Random forest classifier
  - Adaboost classifier
- Run and Evaluate selected models

```
#Importing all the model library

from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB

#Importing Boosting models
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier

#Importing error metrics
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,roc_curve,auc
from sklearn.model_selection import GridSearchCV,cross_val_score
```

```
KNN=KNeighborsClassifier(n_neighbors=6)
DT=DecisionTreeClassifier(random_state=6)
RF=RandomForestClassifier()
ADA=AdaBoostClassifier()
MNB=MultinomialNB()
```

```
models= []
models.append(('KNeighborsClassifier', KNN))
models.append(('DecisionTreeClassifier', DT))
models.append(('RandomForestClassifier', RF))
models.append(('AdaBoostClassifier', ADA))
models.append(('MultinomialNB', MNB))
```

- The results observed over different evaluation metrics are shown in fig,

```
R2 score of the model :  KNeighborsClassifier(n_neighbors=6) = 95.9 %
mean squared error for model :  KNeighborsClassifier(n_neighbors=6) = 0.08136134972667672
cross val score for model :  KNeighborsClassifier(n_neighbors=6) = 0.8662763202342079
Accuracy :               precision    recall  f1-score   support

           1       1.00      1.00      1.00     11119
           2       1.00      1.00      1.00     11064
           3       0.87      1.00      0.93     10908
           4       0.97      0.92      0.94     11063
           5       1.00      0.89      0.94     10909

    accuracy                           0.96     55063
   macro avg       0.97      0.96      0.96     55063
weighted avg       0.97      0.96      0.96     55063

_______________________________________________
R2 score of the model :  DecisionTreeClassifier(random_state=6) = 96.1 %
mean squared error for model :  DecisionTreeClassifier(random_state=6) = 0.0785645533298222
cross val score for model :  DecisionTreeClassifier(random_state=6) = 0.913810381713771
Accuracy :               precision    recall  f1-score   support

           1       1.00      1.00      1.00     11119
           2       1.00      1.00      1.00     11064
           3       0.87      1.00      0.93     10908
           4       0.98      0.91      0.95     11063
           5       0.99      0.91      0.95     10909

    accuracy                           0.96     55063
   macro avg       0.97      0.96      0.96     55063
weighted avg       0.97      0.96      0.97     55063

_______________________________________________
R2 score of the model :  RandomForestClassifier() = 96.1 %
mean squared error for model :  RandomForestClassifier() = 0.0785645533298222
cross val score for model :  RandomForestClassifier() = 0.9440490935705439
Accuracy :               precision    recall  f1-score   support

           1       1.00      1.00      1.00     11119
           2       1.00      1.00      1.00     11064
           3       0.87      1.00      0.93     10908
           4       0.98      0.91      0.95     11063
           5       0.99      0.91      0.95     10909

    accuracy                           0.96     55063
   macro avg       0.97      0.96      0.96     55063
weighted avg       0.97      0.96      0.97     55063

_______________________________________________
R2 score of the model :  AdaBoostClassifier() = -56.9 %
mean squared error for model :  AdaBoostClassifier() = 3.1407478706209253
cross val score for model :  AdaBoostClassifier() = 0.41681679990992004
Accuracy :               precision    recall  f1-score   support

           1       0.62      0.32      0.42     11119
           2       0.96      0.50      0.66     11064
           3       0.41      1.00      0.58     10908
           4       0.19      0.13      0.16     11063
           5       0.27      0.22      0.24     10909

    accuracy                           0.43     55063
   macro avg       0.49      0.43      0.41     55063
weighted avg       0.49      0.43      0.41     55063

_______________________________________________
R2 score of the model :  MultinomialNB() = 92.5 %
mean squared error for model :  MultinomialNB() = 0.15082723425893976
cross val score for model :  MultinomialNB() = 0.8747550951469429
Accuracy :               precision    recall  f1-score   support

           1       0.98      1.00      0.99     11119
           2       1.00      1.00      1.00     11064
           3       0.88      0.94      0.91     10908
           4       0.87      0.88      0.87     11063
           5       0.94      0.85      0.89     10909

    accuracy                           0.93     55063
   macro avg       0.93      0.93      0.93     55063
weighted avg       0.93      0.93      0.93     55063

_______________________________________________
```

We have got god scores Random forest classifier

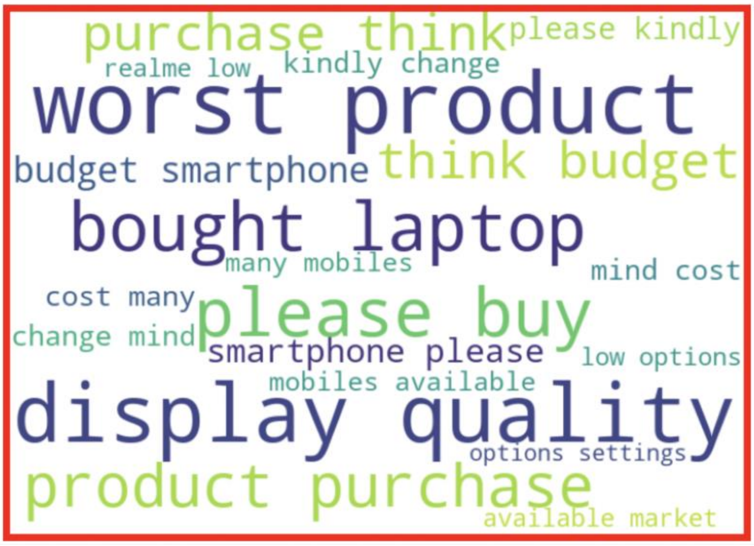
Key Metrics for success in solving problem under consideration

- On the basis of accuracy and confusion matrix we save Random forest classifier as our final model.

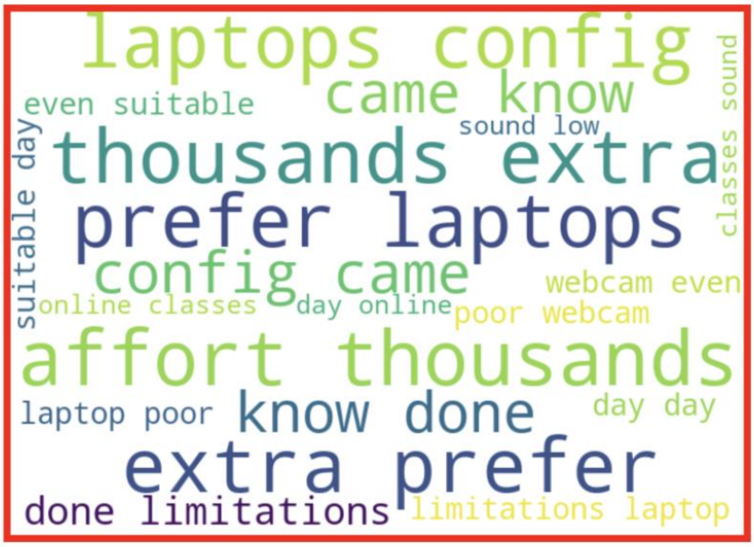
Visualizations

-

```python
#getting sense of review Loud words in Rating 1
from wordcloud import WordCloud

Rating1=df['Product_description'][df['Product_Rating']==1]

spam_cloud = WordCloud(width=700,height=500,background_color='white',max_words=20).generate(' '.join(Rating1))

plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```




```python
Rating1=df['Product_description'][df['Product_Rating']==2]

spam_cloud = WordCloud(width=700,height=500,background_color='white',max_words=20).generate(' '.join(Rating1))

plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

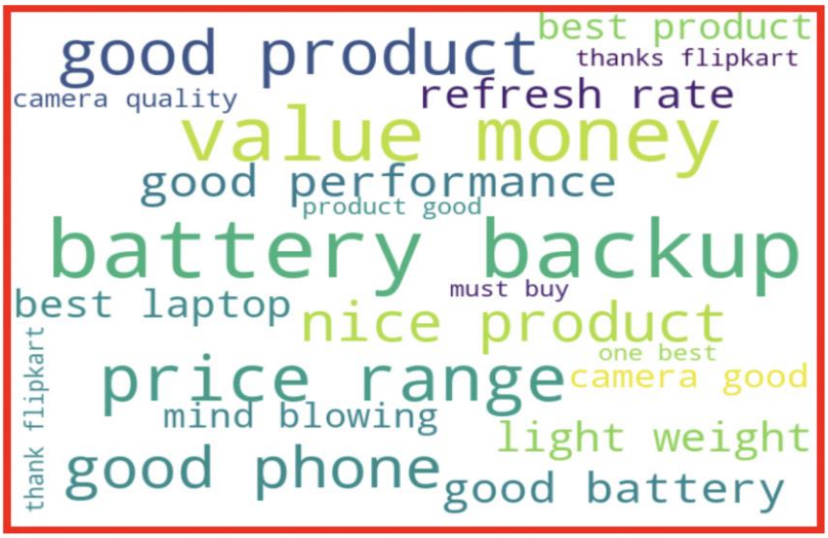



```python
Rating1=df['Product_description'][df['Product_Rating']==3]
```

```
Rating1=df['Product_description'][df['Product_Rating']==3]

spam_cloud = WordCloud(width=700,height=500,background_color='white',max_words=20).generate(' '.join(Rating1))

plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```


```
Rating1=df['Product_description'][df['Product_Rating']==4]

spam_cloud = WordCloud(width=700,height=500,background_color='white',max_words=20).generate(' '.join(Rating1))

plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```


```
Rating1=df['Product_description'][df['Product_Rating']==5]

spam_cloud = WordCloud(width=700,height=500,background_color='white',max_words=20).generate(' '.join(Rating1))

plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



CONCLUSION

Key Findings and Conclusions of the Study

- In this project we have tried to detect the Ratings in commercial websites on a scale of 1 to 5 on the basis of the reviews given by he users.
- We made use of natural language processing and machine learning algorithms in order to do so. We interpreted that Random forest classifier model is giving us best results

Learning Outcomes of the Study in respect of Data Science

- Through this project we were able to learn various Natural language processing techniques like lemmatization, stemming, removal of stopwords.
- This project has demonstrated the importance of sampling effectively, modelling and predicting data.
- Through different powerful tools of visualization we were able to analyse and interpret different hidden insights about the data.
- The few challenges while working on this project where:-
    a. Imbalanced dataset
    b. Lots of text data
- The dataset was highly imbalanced so we balanced the dataset using smote technique.
- We converted text data into vectors with the help of tfidf vectorizer.

- ## Limitations of this work and Scope for Future Work

While we couldn’t reach out goal of maximum accuracy in Ratings prediction proje, we did end up creating a system that can with some improvement and deep learning algorithms get very close to that goal. As with any project there is room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and vesatility to the project.

Thank You

Akhil Reddy