FLIP ROBO

RATINGS PREDICTION PROJECT

Submitted by:

JASMINE KAUR

DATA SCIENCE INTERN

ACKNOWLEDGMENT

This project would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am thankful to Flip Robo Technologies for their guidance and constant help provided from time to time has enabled the completion of this project. I want to thank my SME Ms. Khushboo Garg for her support in completing this project

I would like to express my gratitude towards my parents and friends as without their constant believe, this would have not been possible.

INTRODUCTION

Business Problem Framing

- This is a machine learning project performed on customer reviews. We collected reviews and processed those using NLP techniques.
- Today, millions of people use AMAZON to purchase stuff. The consumers then rate the purchase and write a review about the experience of their purchase. If they like it, they leave a positive review with some good rating. On the other side, if they don't like the product, they leave a negative review and give bad ratings. Our aim in this project is to predict star rating based on the product review.
- The range of star rating is 1 to 5 where 1 being the bad rating and 5 being the excellent rating.
- This project is similar to the sentiment analysis, but instead of predicting the positive negative, or neutral sentiment, here we need to predict the star rating.

Conceptual Background of the Domain Problem

- With the growth in internet and e-commerce, customers have found an easy and convenient way to make their purchase at the comfort of sitting at their home and getting the product delivered at the doorstep. Some of the platforms that offer these services are Amazon, Walmart, Flipkart etc. As the consumers are not able to touch and have an experience of the product personally, so the purchases rely on the reviews written by the consumers who have already bought them. Consumers have the option to compare the products based on quality, pricing, etc on different websites. The most important and reliable way to compare is to look at the consumer reviews. Also, these reviews act as a feedback to the consumer that tell them whether the product is up to the mark or they need to make some alternations.

Review of Literature

- This project is mainly about exploration, feature engineering and classification that can be done on the data. There are five possible categories i.e. 1.0 star, 2.0 stars, 3.0 stars, 4.0 stars, and 5.0 stars.
- Many initiatives have been taken in this direction to figure out the rating based on the reviews. This is done by analysing the reviews and then predicting the star rating associated with the review. This is to allow people to explain and review their purchase with each other in this increasingly digitalized world.

Motivation for the Problem Undertaken

- In the era of digital world, we come across various choices even for a single product. It is a difficult decision for a consumer to make purchase when a wide variety of range is available and at the same time, he/she has no access to the product in person. They need to make a decision based on the picture that is available on the website. But these images might be misleading sometimes. So, to have a reliable source of the product, consumers often rely on the consumers who have used the product. So, the reviews given by the consumers are used to judge whether to make a purchase or not.
- With the growing artificial intelligence, we can use Natural Language Processing (NLP) to minimize the number of false positives to encourage all constructive conversation. Employing machine learning model to predict ratings promotes easier way to distinguish between product quality, cost and related features.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

- We have scraped 50,000 data points from Amazon website. There are two columns in dataset namely Product Review and Ratings.
- The target variable is 'Rating' and this is a categorical variable which makes this a classification problem.
- This project is completed in two parts:
    - Data collection phase
    - Model building phase

Data Collection Phase:

- We have scraped 50,000 rows of data. This process was completed using Selenium. The web scraping was performed on Amazon website to fetch reviews and ratings of the product.
- Each data point contains a review and its associated rating.

Model Building Phase:

- After collecting the data, we have created a machine learning model. The initial process of this model is to perform data pre-processing that involves NLP. We have tries different classification models and then applied hyperparameter tuning on the best model. The steps involved in this process are:
    1. Data cleaning
    2. Exploratory data analysis
    3. Data pre-processing
    4. Model building
    5. Model evaluation
    6. Selecting the best model

Data Sources and their formats

- We have collected data from Amazon using web scraping technique and the framework used is Selenium.
- We have scraped 50,000 reviews and their ratings.
- The first five rows of the data looks like:

Unnammed: 0	Product_Review	Ratings
0	Evolve2 85 feels light and has adequate featur...	3.0 out of 5 stars
1	Edit 2: January 2021 the price shows these at ...	3.0 out of 5 stars
2	I bought this phone because it was rated highl...	3.0 out of 5 stars
3	The headset was comfortable even on long calls...	3.0 out of 5 stars
4	I wrote a lengthy word document comparing thes...	3.0 out of 5 stars

- We have saved the data in a csv file.

Data Pre-processing

- *Checking null values*: We got that there are null values in the data:
- `Product_Review     400`
- `Ratings           2400`

- *Checking values counts of the Ratings column*:

```
 5.0    16760
1.0    11760
4.0     7210
3.0     3970
2.0     3200
iCl     1600
HiF      600
euf      400
Ter      200
Mao      200
Log      200
Lap      200
ICC      200
pTr      200
Ama      200
WK       200
iGR      200
SWA      200
(Re      100
```

We have realized that there are more categories in the ratings column than expected. So, we dropped all the categories except 1.0, 2.0, 3.0, 4.0, and 5.0.

- *Calculating average*: It comes to be 3.326.
- *Pre-processing using NLP:* We cleaned the data using regex, matched the patterns in the comment and replaced them with

more organized counterparts. Cleaner data leads to more efficient model and higher frequency. The steps involved in this are:

1. Removing punctuations and other special characters.
2. Splitting the comments into individual words.
3. Removing stop words.

This was followed by:

i. Stemming
ii. Lemmatizing
iii. Word cloud
iv. Feature extraction

Hardware and Software Requirements and Tools Used

The hardware used in this project:

- Laptop with high end specification and a stable internet connection.

The software used in this project:

- Anaconda Navigator
- Jupytor notebook
- MS Excel
- Various libraries in python

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

Testing of Identified Approaches (Algorithms)

- Logistic Regression

- Multinomial NB
- Decision Tree Classifier
- KNeighbors Classifier
- Random Forest Classifier
- AdaBoost Classifier
- Gradient Boosting Classifier

Run and Evaluate selected models

After running the code, the result is:

```
************* Logistic Regression *********************


LogisticRegression()


accuracy_score:  0.48399558498896245


cross_val_score:  0.2862913907284768


Classification report:

              precision    recall  f1-score   support

         1.0       0.46      0.52      0.49      2361
         2.0       0.48      0.16      0.24       680
         3.0       0.41      0.17      0.24      1222
         4.0       0.41      0.26      0.32      1415
         5.0       0.52      0.73      0.61      3382

    accuracy                           0.48      9060
   macro avg       0.46      0.37      0.38      9060
weighted avg       0.47      0.48      0.45      9060



Confusion matrix:

[[1233   49   96  153  830]
 [ 253  110    2   93  222]
 [ 340   15  205   92  570]
 [ 269   22   62  361  701]
 [ 567   33  131  175 2476]]



********************** MultinomialNB ***************************
```

```
MultinomialNB()


accuracy_score:  0.46357615894039733


cross_val_score:  0.24852097130242828


Classification report:

              precision    recall  f1-score   support

         1.0       0.50      0.41      0.45      2361
         2.0       0.34      0.35      0.35       680
         3.0       0.35      0.25      0.29      1222
         4.0       0.36      0.32      0.34      1415
         5.0       0.52      0.66      0.58      3382

    accuracy                           0.46      9060
   macro avg       0.42      0.40      0.40      9060
weighted avg       0.45      0.46      0.45      9060



Confusion matrix:

[[ 966  172  192  230  801]
 [ 150  239   19   79  193]
 [ 258   65  300  121  478]
 [ 175   79   98  456  607]
 [ 392  142  244  365 2239]]



****************** DecisionTreeClassifier ***************************


DecisionTreeClassifier()


accuracy_score:  0.48730684326710816


cross_val_score:  0.27474613686534216


Classification report:

              precision    recall  f1-score   support

         1.0       0.47      0.53      0.50      2361
         2.0       0.45      0.19      0.27       680
         3.0       0.42      0.19      0.26      1222
         4.0       0.42      0.27      0.33      1415
         5.0       0.52      0.72      0.61      3382
```

```
    accuracy                           0.49      9060
   macro avg       0.46      0.38      0.39      9060
weighted avg       0.47      0.49      0.46      9060


Confusion matrix:

[[1250   59  101  166  785]
 [ 253  131    2   93  201]
 [ 347   28  229   92  526]
 [ 248   31   65  382  689]
 [ 590   42  144  183 2423]]



********** KNeighborsClassifier **************************


KNeighborsClassifier()


accuracy_score:  0.37649006622516556


cross_val_score:  0.23247240618101545


Classification report:

              precision    recall  f1-score   support

         1.0       0.37      0.53      0.44      2361
         2.0       0.23      0.15      0.18       680
         3.0       0.24      0.17      0.20      1222
         4.0       0.28      0.30      0.29      1415
         5.0       0.50      0.42      0.46      3382

    accuracy                           0.38      9060
   macro avg       0.32      0.31      0.31      9060
weighted avg       0.38      0.38      0.37      9060


Confusion matrix:

[[1254   74  231  225  577]
 [ 280  104   56  129  111]
 [ 455   53  204  183  327]
 [ 409   76   93  422  415]
 [ 985  143  265  562 1427]]



********** RandomForestClassifier **********
```

```
RandomForestClassifier()


accuracy_score:  0.4878587196467991


cross_val_score:  0.2778587196467991


Classification report:

              precision    recall  f1-score   support

         1.0       0.47      0.52      0.49      2361
         2.0       0.47      0.18      0.26       680
         3.0       0.42      0.18      0.25      1222
         4.0       0.41      0.27      0.32      1415
         5.0       0.52      0.73      0.61      3382

    accuracy                           0.49      9060
   macro avg       0.46      0.38      0.39      9060
weighted avg       0.47      0.49      0.46      9060



Confusion matrix:

[[1221   56  107  165  812]
 [ 253  122    2  102  201]
 [ 341   21  221   98  541]
 [ 248   22   65  377  703]
 [ 561   36  137  169 2479]]



********** AdaBoostClassifier **************************


AdaBoostClassifier()


accuracy_score:  0.39867549668874175


cross_val_score:  0.31863134657836645


Classification report:

              precision    recall  f1-score   support

         1.0       0.35      0.17      0.23      2361
         2.0       0.41      0.09      0.15       680
         3.0       0.00      0.00      0.00      1222
         4.0       0.35      0.08      0.12      1415
         5.0       0.41      0.90      0.56      3382
```

```
    accuracy                           0.40      9060
   macro avg       0.30      0.25      0.21      9060
weighted avg       0.33      0.40      0.30      9060


Confusion matrix:

[[ 412   33    0   38 1878]
 [ 211   63    0   93  313]
 [ 159   15    0   26 1022]
 [ 129   11    0  107 1168]
 [ 276   30    0   46 3030]]


********** GradientBoostingClassifier **************************

GradientBoostingClassifier()

accuracy_score:  0.4746136865342163

cross_val_score:  0.29578366445916116

Classification report:

              precision    recall  f1-score   support

         1.0       0.49      0.45      0.47      2361
         2.0       0.48      0.16      0.24       680
         3.0       0.45      0.14      0.21      1222
         4.0       0.45      0.16      0.24      1415
         5.0       0.47      0.81      0.60      3382

    accuracy                           0.47      9060
   macro avg       0.47      0.34      0.35      9060
weighted avg       0.47      0.47      0.43      9060


Confusion matrix:

[[1064   49   75   96 1077]
 [ 191  110    2   41  336]
 [ 300   15  165   39  703]
 [ 205   22   35  228  925]
 [ 425   33   90  101 2733]]
```

Key Metrics for success in solving problem under consideration

The key metrics used in this project were accuracy_score, cross_val_score, classification report, and confusion matrix. We tried to find out the best parameters and also to increase our scores by using hyperparameter tuning and we will be using GridSearchCV.

1. CROSS VALIDATION:

Cross validation helps to find out the over fitting and under fitting of the model. In the cross validation, the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part being 20% of full dataset. While running the cross-validation the 1st part (20%) of the 5 parts will be kept out as a holdout set for validation and everything else is used for training data. This way we will get the estimate of the dataset,

In the similar way, further iterations are made for the second 20% of the dataset is held as a holdout set and remaining 4 parts are used for training data during process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross-validation process to get the remaining estimate of the model quality.

2. CONFUSION MATRIX:

A confusion matrix, also known as error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). It is a special kind of contingency table, with two dimensions ('actual' and 'predicted'). The name stems from the fact that it makes it easy to see whether the system is confusing two classes(i.e. commonly mislabelling one as another).

3. CLASSIFICATION REPORT

The classification report displays the precision, recall, F1, and support scores for the model.

Precision = True positive/(True positive + False positive)

Recall = True positive/(True positive + False Negative)

F1 score = 2*(Recall*Precision)/(Recall + Precision)

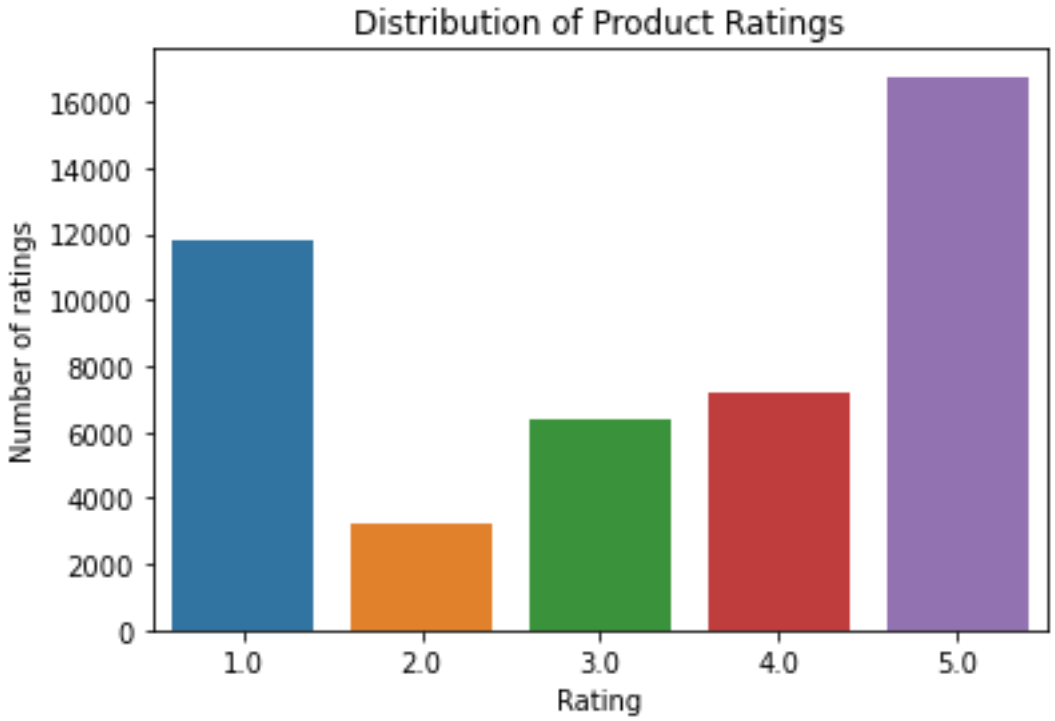
Support: It is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn't change between models but instead diagnoses the evaluation process.

4. Hyperparameter Tuning:

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for the model. You must select from a specific list of hyperparameters for a given model as it varies from model to model.

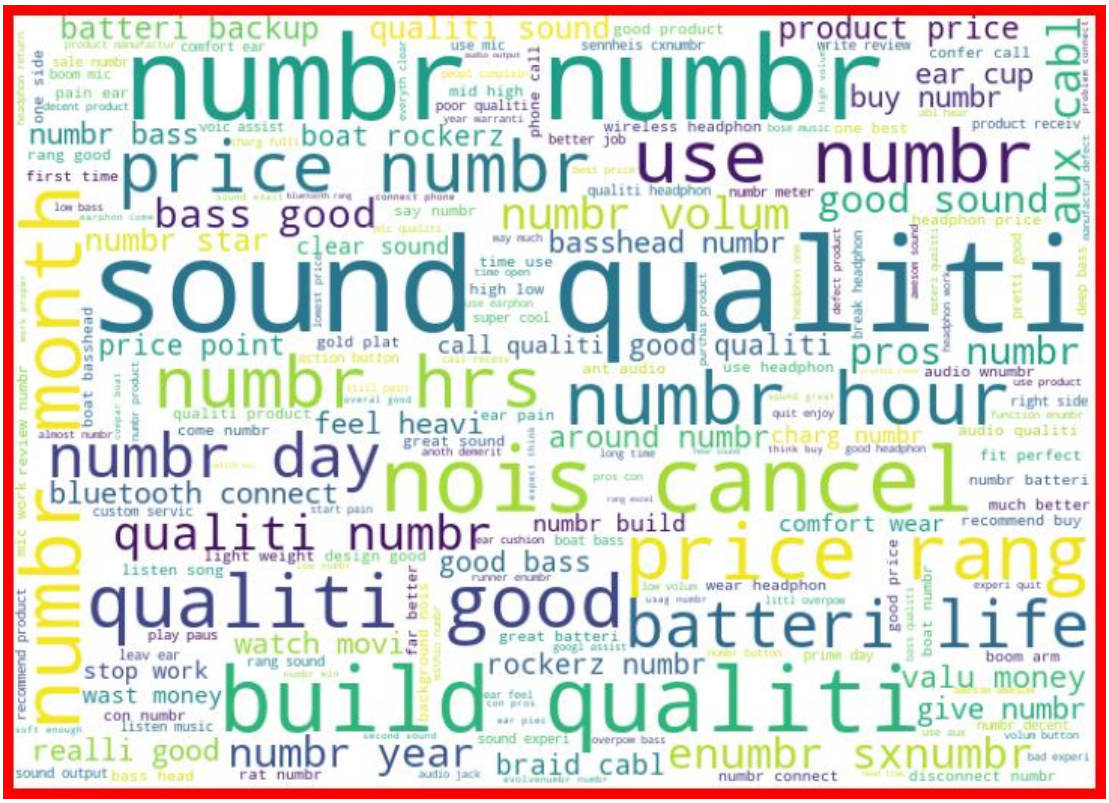
We are not aware of optimal values for hyperparameters which would generate the best model output. So, for this, GridSearchCV is used to find best parameters.

Visualizations



The maximum number of rating is 5.0 which are followed by 1.0 and then 2.0, 3.0 and the least is 4.0.

EXAMPLE OF WORD CLOUD:



CONCLUSION

Key Findings and Conclusions of the Study

- After the completion of this project, we got an insight of how to collect data, how to do pre-processing and analysing the data and at the end building the model.
- Collecting the data is challenging task as getting the right data is an important part of the project.
- Cleaning the data and removing inefficiency requires a lot of time and attention to reach the final clean data which is ready to get processed.
- Performing NLP technique requires careful attention to the data and irregularity present in the data.
- Applying train test split and comparing each model is an important task to reach at the final model.
- Hyperparameter helps us get the best parameters and this increases the efficiency of the model.
- Finally, we saved the model in pkl format.

Limitations of this work and Scope for Future Work

- The data could have been scraped from different websites.
- Some of the reviews were misleading and thus leads to wrong labelling.
- Wordcloud was not showing proper text which had more positive and negative weightage.