REVIEW RATING PREDICTION

Submitted by

Steffin Varghese

Batch - 26

In partial fulfilment of Data Science – Internship

At

Flip Robo Technologies

AI and Software Development company

Flip Robo Technologies | Indiranagar, Bengaluru - 560 038, Karnataka, India

Month of Submission

August 2022

Acknowledgement

I am highly indebted to FlipRobo Technologies for giving me this opportunity to work on a project and for the guidance and persistent supervision, as well as for providing necessary project information and assistance in completing the project.

I would want to convey my gratitude to the members of FlipRobo Technologies for their kind encouragement and support in completing this project.

I would like to extend my heartfelt gratitude and appreciation to SME, Ms Khushboo Garg for spending such close attention to me and assisting me during the course of completion of project, as well as towards others who have volunteered to assist me with their skills.

I acknowledge my gratitude towards the authors of papers: “Predicting User Ratings Using Status Models on Amazon.com”, “Rating Prediction Based on Social Sentiment from Textual Reviews” and “Predicting ratings of Amazon reviews - Techniques for imbalanced datasets” for the insights I could attain through the extensive research which enhanced my knowledge in development of the project.

CONTENTS

INTRODUCTION

- Business Problem Framing
- Conceptual Background of the Domain Problem
- Review of Literature
- Motivation for the Problem Undertaken

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem
- Data Sources and their formats
- Data Pre-processing
- Assumptions related to the problem under consideration
- Hardware and Software Requirements and Tools Used

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
- Testing of Identified Approaches (Algorithms) and evaluation of selected models
- Key Metrics for success in solving problem under consideration
- Visualizations
- Interpretation of the Results

CONCLUSION

- Key Findings and Conclusions of the Study
- Learning Outcomes of the Study in respect of Data Science
- Limitations of this work and Scope for Future Work

INTRODUCTION

Business Problem Framing

E-Commerce platforms are having a good time during the past few years as the customers are opting for digital shopping. With the impact of digitalization and pandemic in the past few years, the demand for online shopping have been always going up. As the demand is steadily increasing, a greater number of e-commerce sellers are adding up to the market to catch up with the trend. So, the competition for capturing the market is high and e-commerce sellers are finding different ways to satisfy the existing customers and expanding their reach by influencing the potential customers in online shopping.

While looking for the new customers, E-Commerce sellers always have to give much importance to one thing. The way they treat their existing customers. Because, now a days, almost all sellers are providing good quality products at best price, but they are lacking behind when it comes to customer support. Considering this crucial factor, e-commerce sellers implemented the customer rating and feedback program, where the customers can rate and provide their reviews and feedbacks on the quality of product, price, and services during and after the sale from the seller. This will help a new customer to gather information on the product as well as the customer satisfaction on the seller during and after purchasing the product.

Conceptual Background of the Domain Problem

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.

Review of Literature

Predicting ratings of Amazon reviews - Techniques for imbalanced datasets (2017)

Marie MARTIN

In this dissertation, authors investigated various models for accurately predicting a user's numerical rating from the review text content. For this, text classification has been used on two separate datasets from Amazon. Text classification allows a document to be automatically classified into a fixed set of classes after being trained over previous annotated data. These datasets are distinguished by an imbalanced distribution and relate to the experience and search product categories, respectively. They observed that binary classification offers better results for the experience products while multi-class classification performed better with the search products

Predicting User Ratings Using Status Models on Amazon.com (2019)

Borui Wang, Guan (Bell) Wang, Zhemin Li

In this study, authors have explored two models. The first model depends more on using status theory and network structure analysis. The second one focuses more on status information based on reviews. From the results, the first model performed better with a mean squared error around 0.1173 and prediction accuracy around 96.93% with three triangles employed as training features.

Rating Prediction Based on Social Sentiment from Textual Reviews (2016)

Xiaojiang Lei, Xueming Qian, Member, IEEE, and Guoshuai Zhao

In this research, a recommendation algorithm is put forth by extracting sentiment data from user reviews on social media. To complete the rating prediction challenge, we combine user sentiment similarity, interpersonal sentiment influence, and item reputation similarity into a single matrix factorization framework. We specifically use user sentiment on social media to indicate user preferences.

Motivation for the Problem Undertaken

Customer satisfaction is a vital factor in e-commerce business as the new customers tends to check on the review and feedbacks of existing customers before purchasing a product. But, to evaluate the product and support satisfaction, a customer has to go through various reviews to identify whether it is worth for money and quality of product and services. This is not just time taking process, but it can also give a mixed signals to customers as there can be many positive and negative reviews. Review rating is an effective technique that will be best to overcome this drawback. Because a customer can just check the numeric rating of customers to identify whether the product is worth for value or not. It is an easy process and customers will have an idea on whether to buy the product from this seller or not based on the rating.

So, the e-commerce companies should make it easier for the customers, by providing a system which can predict the overall numeric rating based on the review of customer. With the help of machine learning and Natural Language Processing (NLP), we can build a model that can predict the review rating based on the customer reviews.

For building this model, we will have to gather the customer reviews from various e-commerce websites for different products. Then, identify the patterns, if there is any, existing between the customer review and customer rating. This will help us to build a powerful model that can predict the customer rating based on their reviews. So, sellers can use this model to rate the customer review at the time when a customer provide a review or feedback for the product or service.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

The study is divided as three phases: -

1. Data Collection 2. Data Analysis 3. Model Building

We have to collect the data which are required for making the analysis and building a predictive model.

We need to collect the reviews of customers for various products on different ecommerce websites which includes the following important information.

1. Review of the product

2. Rating of the product

After collecting the required data, we have to build the machine learning model. Before that, we have to do all the pre-processing steps involving NLP. Finally, we have to select the best model which can predict the ratings based on the customer reviews.

Data Sources and their formats

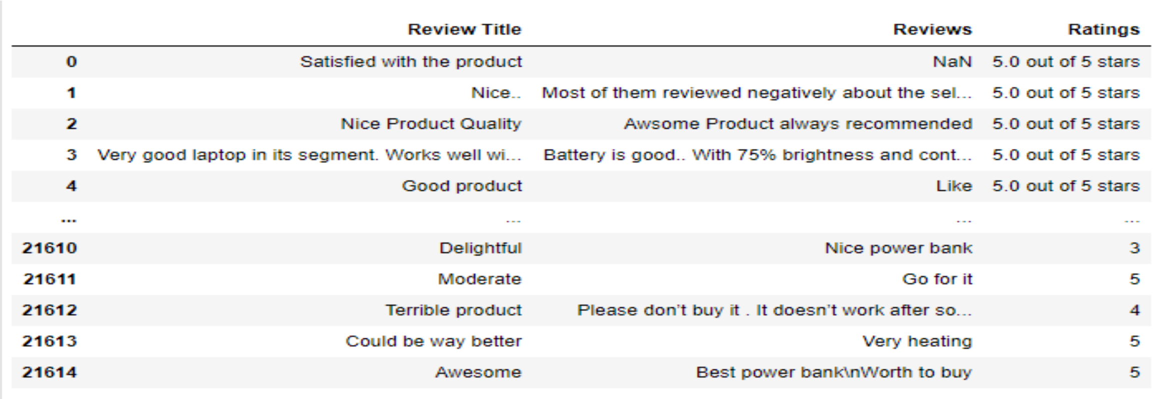
We had to scrape 20,000 records of customer reviews from two websites: - Amazon.in, Flipkart.com

We have collected customer reviews for the products:

`Laptops` `Phones` `Headphones` `Smart Watches` `Professional Cameras`

`Printers` `Monitors` `Home theater` `Router` `Power bank`

We have collected 21615 records of customer reviews and ratings.



	Review Title	Reviews	Ratings
0	Satisfied with the product	NaN	5.0 out of 5 stars
1	Nice..	Most of them reviewed negatively about the sel...	5.0 out of 5 stars
2	Nice Product Quality	Awsome Product always recommended	5.0 out of 5 stars
3	Very good laptop in its segment. Works well wi...	Battery is good.. With 75% brightness and cont...	5.0 out of 5 stars
4	Good product	Like	5.0 out of 5 stars
...	...	...	...
21610	Delightful	Nice power bank	3
21611	Moderate	Go for it	5
21612	Terrible product	Please don't buy it . It doesn't work after so...	4
21613	Could be way better	Very heating	5
21614	Awesome	Best power bank\nWorth to buy	5

Dataset

We have two features and one target column in the dataset.

Description of the dataset

Features in Dataset (Independent Variable)

`Review Title` = Title/Summary of the review.

`Reviews` = Detailed review

Target in dataset (Dependent Variable)

`Ratings` = Ratings based on the review.

Data Pre-processing

Text Processing

While exploring the categorical variables, there are many words, numbers, as well as punctuations which are not important for our predictions. So, we had to process the text.

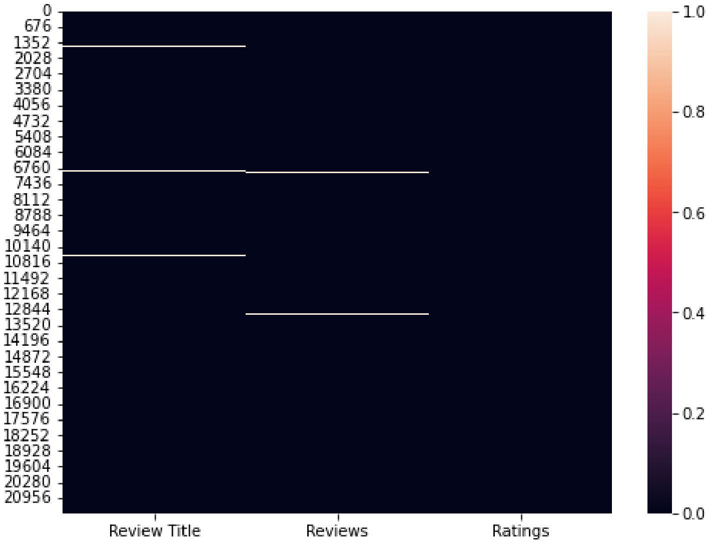
- Using the user defined function “clean_text”, we removed numbers, URLs, punctuations, and other unwanted characters from the data and unshrink the words which were mentioned in short forms.

- Removed **StopWords**

- **Lemmatization** – Using wordnet lemmatizer and word tokenize, we removed the inflected forms of words and converted them to their ‘lemma’ or dictionary form.

- **Text Normalization – Standardization –** We used user defined function ‘scrub_words’ to remove html tags, non-ascii and digits and white spaces from the text data.

Checking for Missing Values



Missing Data in Dataset using heatmap

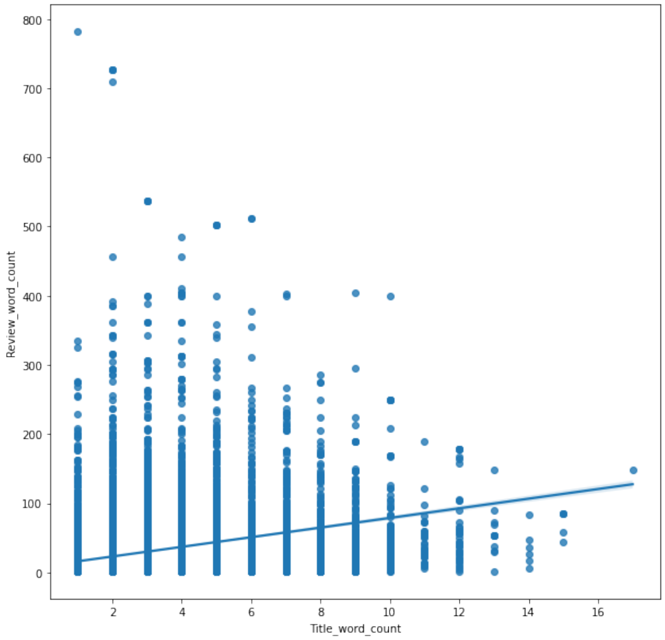
Since the consolidated missing records were only around 1.03%, we dropped these records from the dataset as this will not have much impact on the model building.

1
811
1619
2414
3216
4022
4831
5633
6434
7252
8051
8853
9657
10450
11247
12042
12839
13633
14436
15234
16037
16835
17628
18435
19229
20032
20830
Review Title
Reviews
Ratings
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100

Dataset after removing missing values

Data Cleansing

Removed Outliers in the datasets

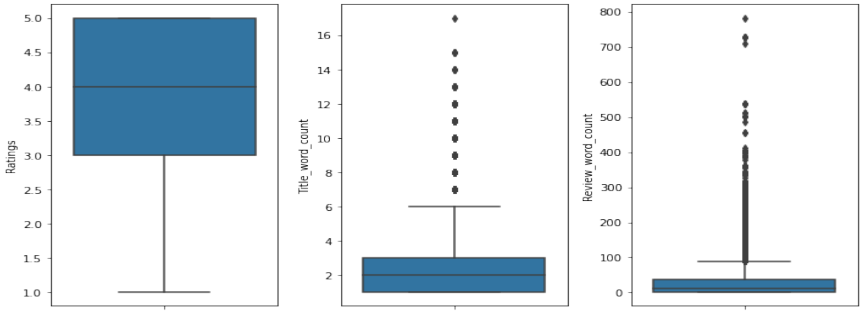


Distribution of word counts for each record in dataset

We have used ZScore outlier removal method for removing the outliers data from the dataset. By outliers in this context, we mean the records with word counts which has huge difference from the average. After removing the outlier data from the dataset, we were losing only 3.53% of records from the dataset.

Assumptions Related to the Problem Under Consideration

Outliers

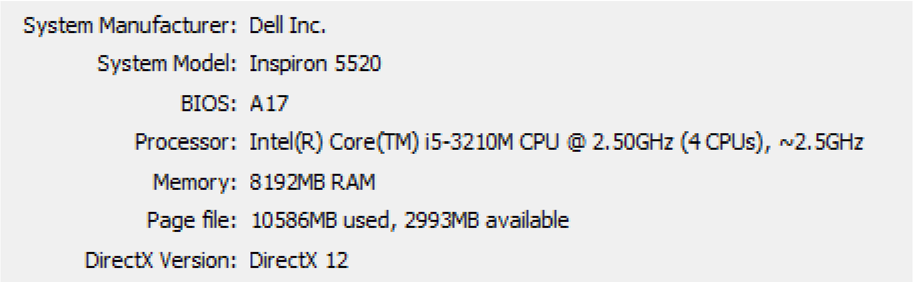


Outliers in Dataset

We can see that the data was having records with high word counts which is far from the 75%.

Hardware and Software Requirements and Tools Used

Hardware Requirement:



```
System Manufacturer: Dell Inc.
       System Model: Inspiron 5520
               BIOS: A17
          Processor: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (4 CPUs), ~2.5GHz
             Memory: 8192MB RAM
          Page file: 10586MB used, 2993MB available
    DirectX Version: DirectX 12
```

Hardware Configuration

Software Requirements:

- Windows Version: Windows 10 Pro
- Anaconda Navigator: 2.0.3
  Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Jupyter Notebook: 6.3.0
  Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Python3: Python 3.9.9
  Python3 is used as the base environment for performing the machine learning and data analysis.

  Python Libraries Used:
- Pandas: Data manipulation and analysis
- NumPy: Adding support for large, multi-dimensional arrays and matrices, along with an enormous collection of high-level mathematical functions to operate on these arrays.
- Matplotlib, Seaborn: For visualization of variable relations and data distribution, and analysis.
- Wordcloud : For plotting the word cloud
- Sklearn: Simple and efficient tools for predictive data analysis.
- SciPy: SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics, and many other classes of problems.
- Statsmodels: Provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Nltk : For natural language processing.
- Xgboost, catboost, lightgbm: Gradient boosting framework that uses tree-based learning algorithms.
- Pickle: Implements binary protocols for serializing and de-serializing

Model/s Development and Evaluation

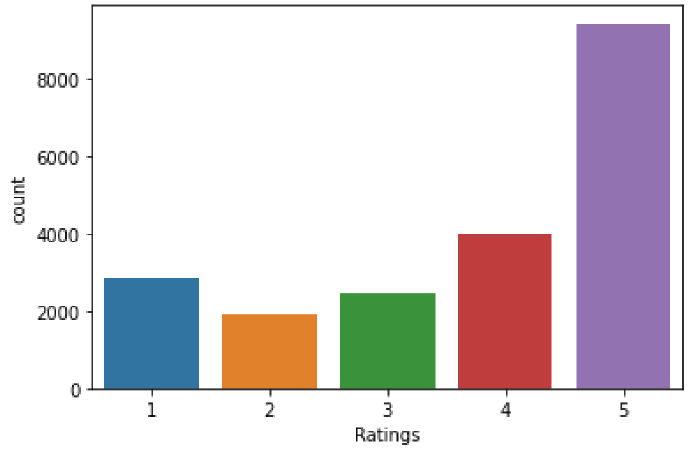
Identification of possible problem-solving approaches (methods)

Vectorizing the Text Data in Dataset

We used tfidf vectorizer for vectorizing the text data in the dataset.

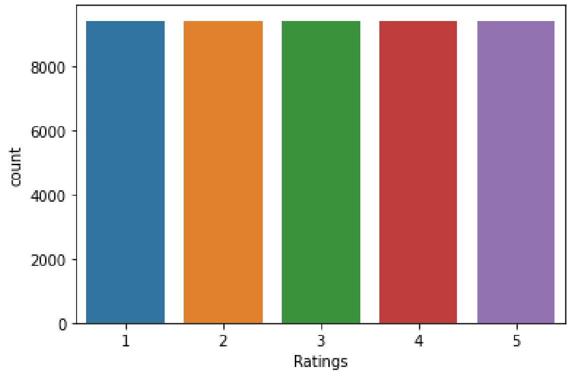
Balancing the dataset

We used smote technique to oversample the data in dataset to equalize the number of records in each category.



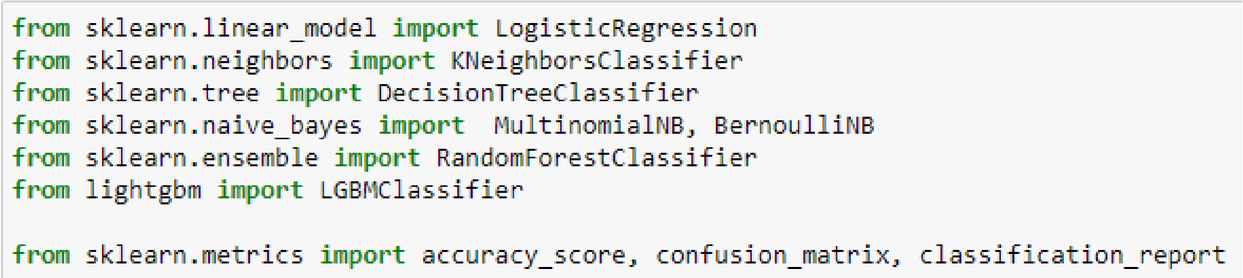
```
5     9425
4     3988
1     2873
3     2469
2     1882
```

Number of records for each category before balancing



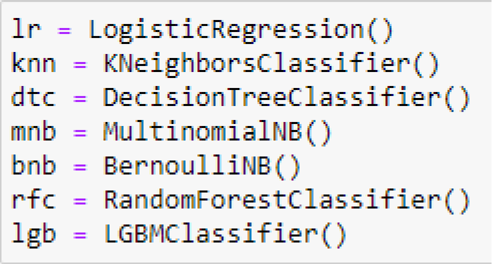
```
5     9425
3     9425
2     9425
4     9425
1     9425
```

Number of records for each category after balancing



```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import  MultinomialNB, BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from lightgbm import LGBMClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Importing required libraries for building the model

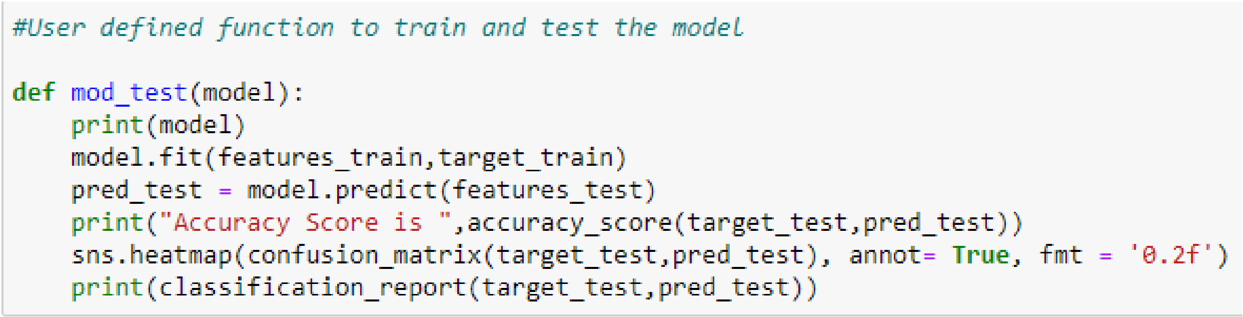


```
lr = LogisticRegression()
knn = KNeighborsClassifier()
dtc = DecisionTreeClassifier()
mnb = MultinomialNB()
bnb = BernoulliNB()
rfc = RandomForestClassifier()
lgb = LGBMClassifier()
```

Creating the instances for the algorithms

- mod_test: Training and testing the model
- cross_val: Finding the best cross validation mean score for each model.

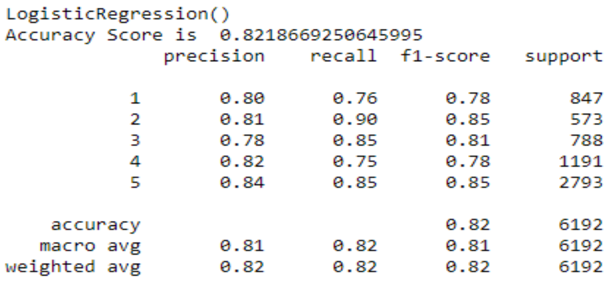
Testing of Identified Approaches (Algorithms) and evaluation of selected models



```
#User defined function to train and test the model

def mod_test(model):
    print(model)
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    print("Accuracy Score is ",accuracy_score(target_test,pred_test))
    sns.heatmap(confusion_matrix(target_test,pred_test), annot= True, fmt = '0.2f')
    print(classification_report(target_test,pred_test))
```

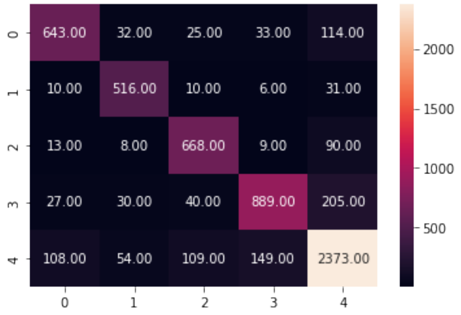
Code Snippet for function to test the model

➢ Logistic Regression



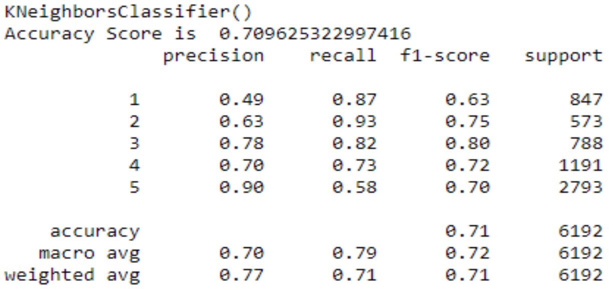
```
LogisticRegression()
Accuracy Score is  0.8218669250645995
              precision    recall  f1-score   support

           1       0.80      0.76      0.78       847
           2       0.81      0.90      0.85       573
           3       0.78      0.85      0.81       788
           4       0.82      0.75      0.78      1191
           5       0.84      0.85      0.85      2793

    accuracy                           0.82      6192
   macro avg       0.81      0.82      0.81      6192
weighted avg       0.82      0.82      0.82      6192
```

Testing the model performance



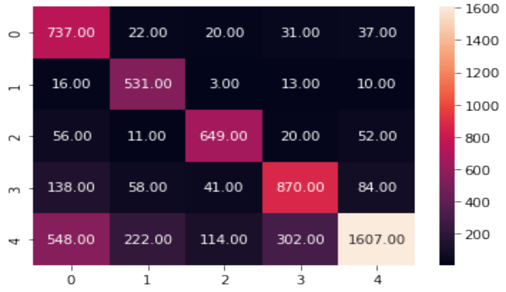
Confusion Matrix

➢ KNeighborsClassifier



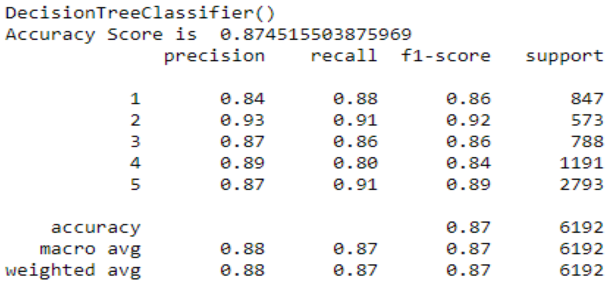
```
KNeighborsClassifier()
Accuracy Score is  0.709625322997416
              precision    recall  f1-score   support

           1       0.49      0.87      0.63       847
           2       0.63      0.93      0.75       573
           3       0.78      0.82      0.80       788
           4       0.70      0.73      0.72      1191
           5       0.90      0.58      0.70      2793

    accuracy                           0.71      6192
   macro avg       0.70      0.79      0.72      6192
weighted avg       0.77      0.71      0.71      6192
```

Testing the model performance



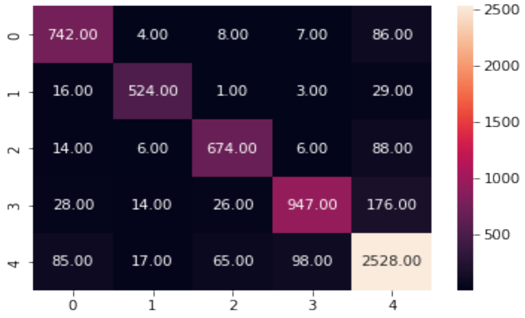
Confusion Matrix

➢ DecisionTreeClassifier



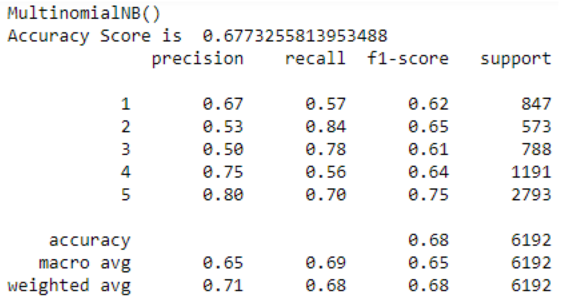
```
DecisionTreeClassifier()
Accuracy Score is  0.874515503875969
              precision    recall  f1-score   support

           1       0.84      0.88      0.86       847
           2       0.93      0.91      0.92       573
           3       0.87      0.86      0.86       788
           4       0.89      0.80      0.84      1191
           5       0.87      0.91      0.89      2793

    accuracy                           0.87      6192
   macro avg       0.88      0.87      0.87      6192
weighted avg       0.88      0.87      0.87      6192
```

Testing the model performance



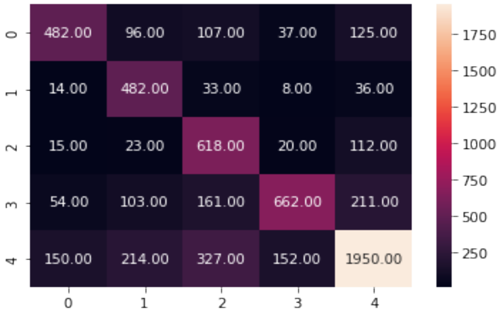
Confusion Matrix

MultinomialNB



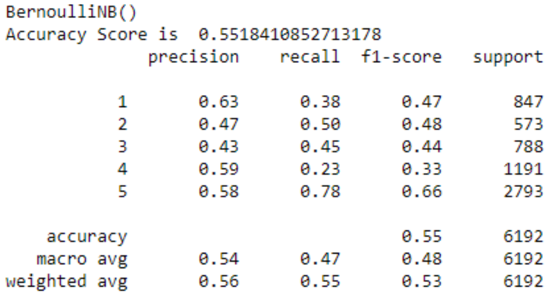
```
MultinomialNB()
Accuracy Score is  0.6773255813953488
              precision    recall  f1-score   support

           1       0.67      0.57      0.62       847
           2       0.53      0.84      0.65       573
           3       0.50      0.78      0.61       788
           4       0.75      0.56      0.64      1191
           5       0.80      0.70      0.75      2793

    accuracy                           0.68      6192
   macro avg       0.65      0.69      0.65      6192
weighted avg       0.71      0.68      0.68      6192
```

Testing the model performance



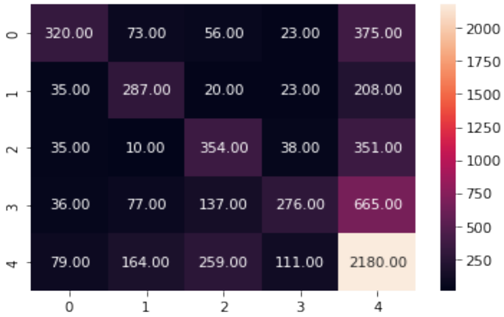
Confusion Matrix

BernoulliNB



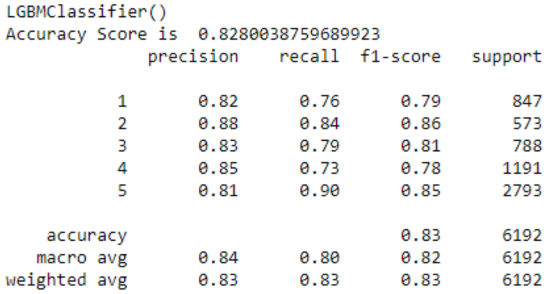
```
BernoulliNB()
Accuracy Score is  0.5518410852713178
              precision    recall  f1-score   support

           1       0.63      0.38      0.47       847
           2       0.47      0.50      0.48       573
           3       0.43      0.45      0.44       788
           4       0.59      0.23      0.33      1191
           5       0.58      0.78      0.66      2793

    accuracy                           0.55      6192
   macro avg       0.54      0.47      0.48      6192
weighted avg       0.56      0.55      0.53      6192
```

Testing the model performance



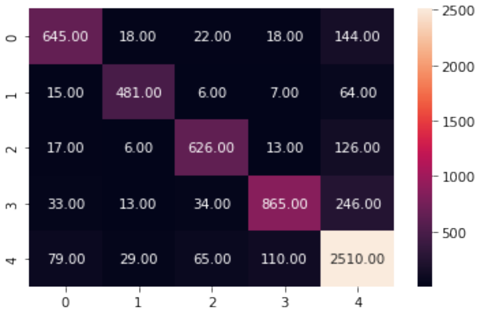
Confusion Matrix

LGBMClassifier



```
LGBMClassifier()
Accuracy Score is  0.8280038759689923
              precision    recall  f1-score   support

           1       0.82      0.76      0.79       847
           2       0.88      0.84      0.86       573
           3       0.83      0.79      0.81       788
           4       0.85      0.73      0.78      1191
           5       0.81      0.90      0.85      2793

    accuracy                           0.83      6192
   macro avg       0.84      0.80      0.82      6192
weighted avg       0.83      0.83      0.83      6192
```

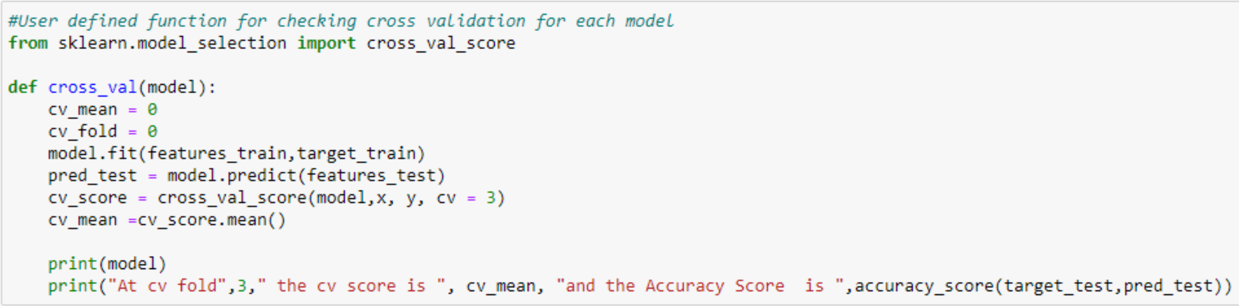
Testing the model performance



Confusion Matrix

After training and testing the models, the DecisionTree Classifier(dtc) and RandomForest Classifier (rfc) are performing well and providing the maximum accuracy score. Now let's check the cross-validation score to find the best performing model.

Cross Validation



```
#User defined function for checking cross validation for each model
from sklearn.model_selection import cross_val_score

def cross_val(model):
    cv_mean = 0
    cv_fold = 0
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    cv_score = cross_val_score(model,x, y, cv = 3)
    cv_mean =cv_score.mean()

    print(model)
    print("At cv fold",3," the cv score is ", cv_mean, "and the Accuracy Score  is ",accuracy_score(target_test,pred_test))
```

Code Snippet for function to find the cross validation mean score

- LogisticRegression
  At cv fold 3 the cv score is 0.43572224645054997 and the Accuracy Score is 0.8218669250645995

- KNeighbors Classifier
  At cv fold 3 the cv score is 0.4229781460483597 and the Accuracy Score is 0.709625322997416

- DecisionTree Classifier
  At cv fold 3 the cv score is 0.45117991956195186 and the Accuracy Score is 0.874515503875969

- MultinomialNB
  At cv fold 3 the cv score is 0.45229442263894953 and the Accuracy Score is 0.6773255813953488

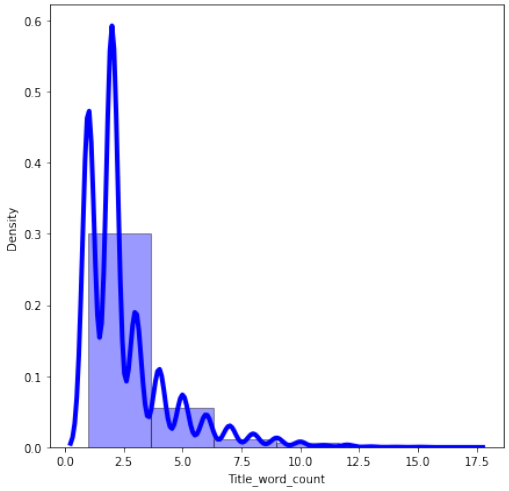
- BernoulliNB
  At cv fold 3 the cv score is 0.4032078305955323 and the Accuracy Score is 0.5518410852713178

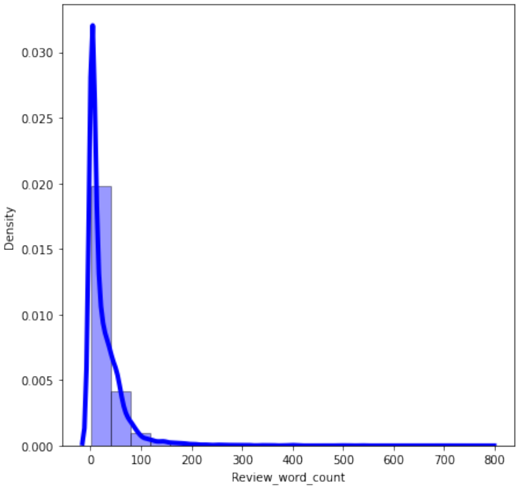
- LGBMClassifier
  At cv fold 3 the cv score is 0.42985899113243203 and the Accuracy Score is 0.8280038759689923

Key Metrics for success in solving problem under consideration

- Accuracy Score: Metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions.
- Confusion Matrix: Performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.
- Classification Report: Displays your model’s precision, recall, F1 score and support. It provides a better understanding of the overall performance of our trained model.

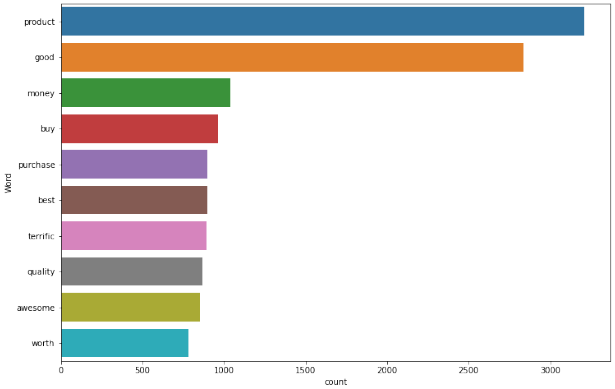
Visualization

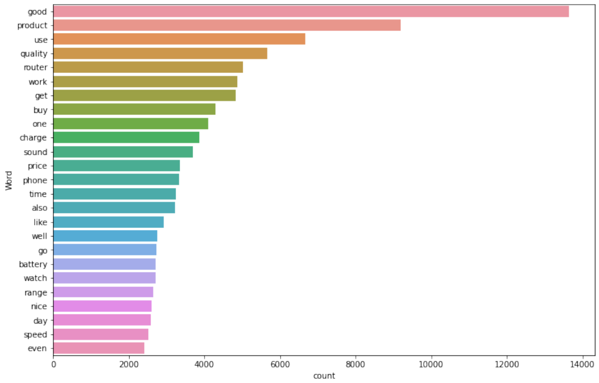




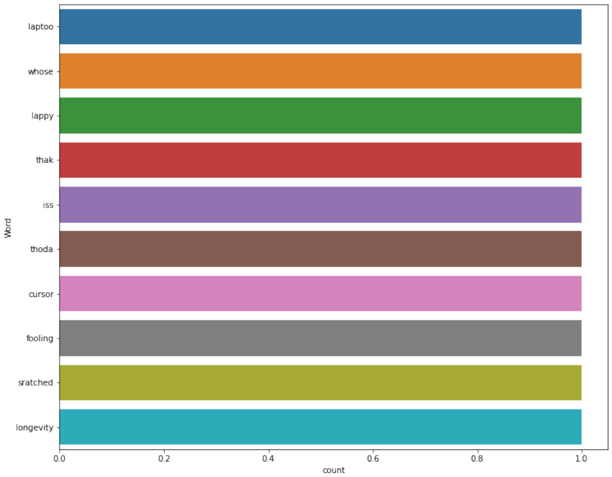
Observations:

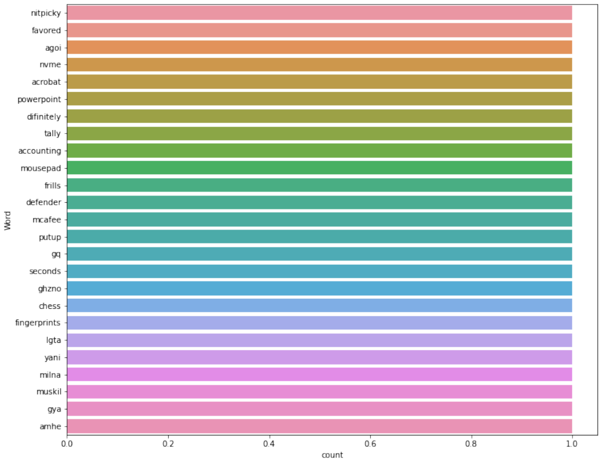
- Most of the values in column 'Review Title' are having a word count between 2-7 words. But there are records which are having around 17 words in this column. So, there can be presence of outliers in these records.
- Most of the records are having 50-100 words in column 'Reviews', but there are some records which are having higher word counts than the average. So, this can be considered as a presence of outliers in the data.



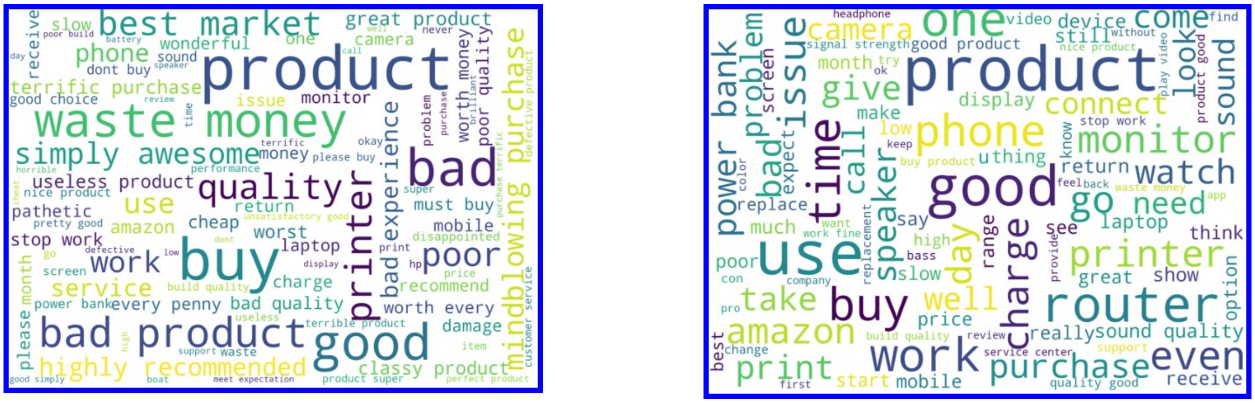


- The words "product", "good" are the most frequent words used by the customers for rating the products as review title.

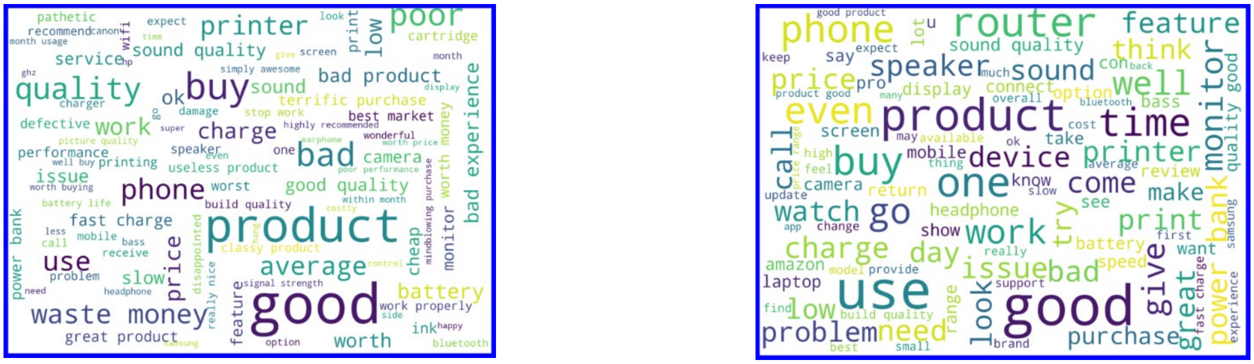




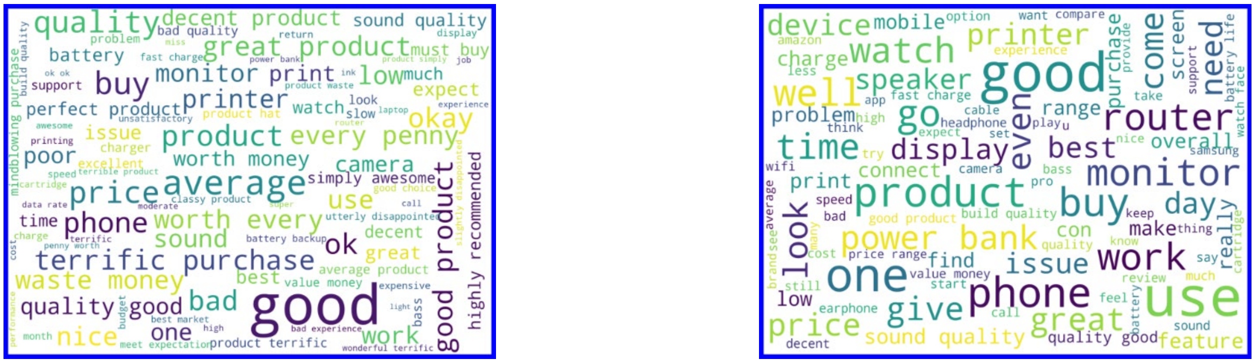
- These are the least words used by customers for review.



Word Cloud for reviews with rating 1



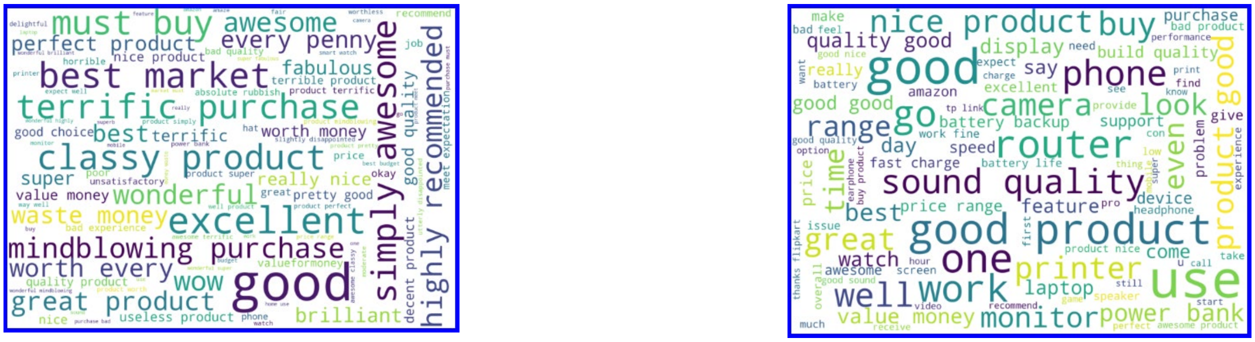
Word Cloud for reviews with rating 2



Word Cloud for reviews with rating 3

must buy
price
good quality
nice product
terrific purchase
really nice
value money
simply awesome
great product
brilliant
best
nice
highly recommended
wonderful
good product
worth every
best market
pretty good
waste money
excellent
good
great
awesome
every penny
good product
product
best
look
power bank
printer
buy
give
camera
phone
time
well
watch
price
monitor
display
headphone
issue
work
day
router
use
charge
problem
one
good
great
quality
speaker

Word Cloud for reviews with rating 4



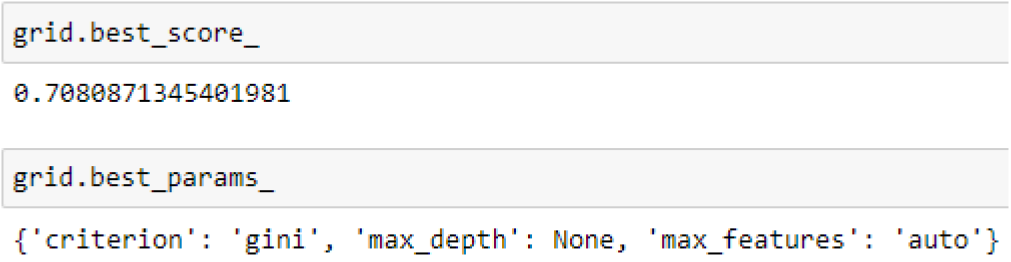
Word Cloud for reviews with rating 5

Observations:

- With word cloud we were able to identify the frequent words which were used by customers for review of different ratings.
- Most of the frequent words are related to the quality of the product irrespective of the rating.

Interpretation of the Results

Hyperparameter Tuning

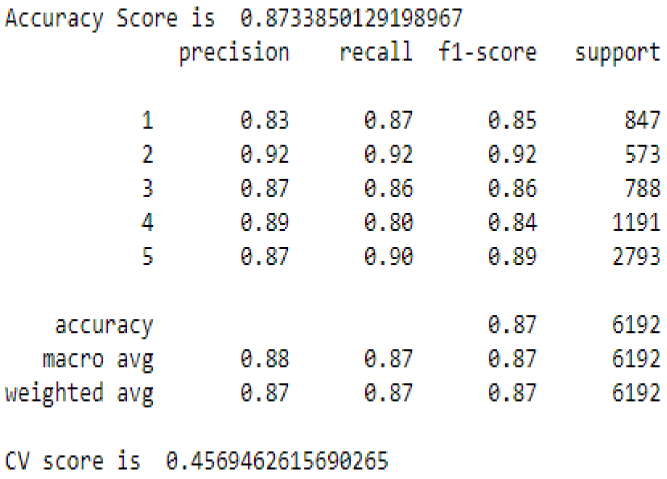


```
grid.best_score_

0.7080871345401981

grid.best_params_

{'criterion': 'gini', 'max_depth': None, 'max_features': 'auto'}
```

Best score and hyper parameters

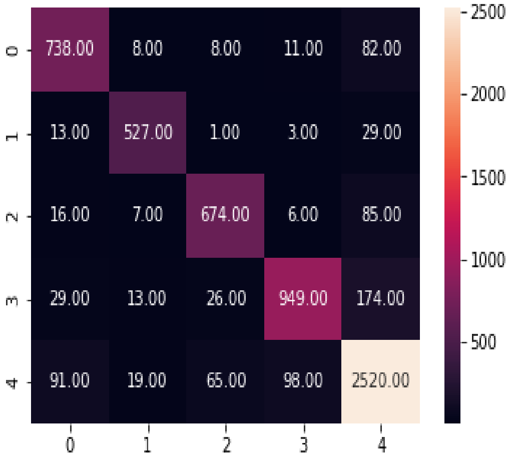
The hyperparameter tuning didn't improve the score. So, we can stick to our existing model. The DecisionTree Classifier Model is performing well after all the testing, cross validations and tuning. So, we can consider this model as best performing model.

Finalized Model Performance with Tuned Parameters



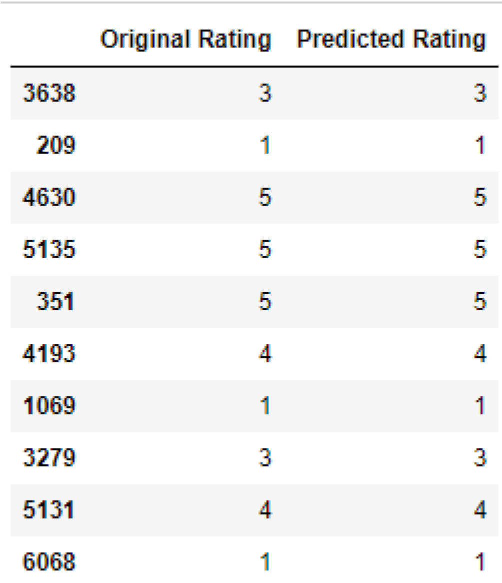
```
Accuracy Score is  0.8733850129198967
              precision    recall  f1-score   support

           1       0.83      0.87      0.85       847
           2       0.92      0.92      0.92       573
           3       0.87      0.86      0.86       788
           4       0.89      0.80      0.84      1191
           5       0.87      0.90      0.89      2793

    accuracy                           0.87      6192
   macro avg       0.88      0.87      0.87      6192
weighted avg       0.87      0.87      0.87      6192

CV score is  0.4569462615690265
```

Final Model Performance



Confusion Matrix

Now we have trained our model and it is ready to test with the actual data to cross verify the performance.



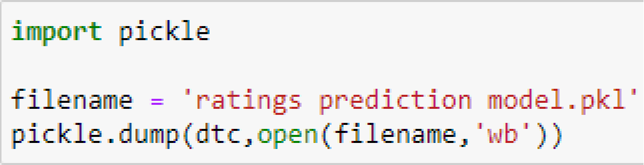
	Original Rating	Predicted Rating
3638	3	3
209	1	1
4630	5	5
5135	5	5
351	5	5
4193	4	4
1069	1	1
3279	3	3
5131	4	4
6068	1	1

Model Predictions and Actual Ratings

Our model is performing well with predictions and provided accurate results.

The dtc model is providing an accuracy score of 87.34% with a cross validation mean score of 45.7%.

Saving the best model



```
import pickle

filename = 'ratings prediction model.pkl'
pickle.dump(dtc,open(filename,'wb'))
```

We have saved the machine learning model for future predictions. We have serialized and saved the binary file as “ratings prediction model.pkl” using the pickle library.

CONCLUSION

Key Findings and Conclusions of the Study

With the help of data science and machine learning, we were able to create a machine learning model using DecisionTree algorithm, which can predict the review ratings based on customer reviews.

Now this model can be used by the client to predict the ratings of customer’s review.

Learning Outcomes of the Study in respect of Data Science

As the E-commerce market is booming, customers are giving with various choices for purchasing the same product from different e-commerce websites. Review ratings helps the customers to easily analyse whether a product is worth buying without inspecting the customer reviews. Using the customer review rating, a customer can identify whether it is good to go for buying a product or not in a glance.

With the help of natural language processing (NLP) we were able to create a machine learning model which can predict the review rating based on customer reviews. With the help of advanced technologies, we will be able to recalibrate this model to suit with the changing trends and extreme text processing, machine will be able to analyse and identify the patterns and could help us to build much more powerful model which can accurately predict the customer ratings based on the customer reviews.

Limitations of this work and Scope for Future Work

Limitations

- We tried to include as many records as possible in the dataset, but since the market we were studying was vast, we couldn't include all the categories and had to let down many other products and their valuable comments for building this model.
- In this study we were primarily concentrated on customer reviews, thus we didn't include many other saliant features which could have been impacted the customer ratings such as product information, user experience on placing the order, order fulfilment, after sale services etc.
- We only included the data from major e-commerce websites in India, but there are many other e-commerce websites which could have been included in the study.

Scope

- In future, to build a powerful machine learning model, we can include much larger dataset by including many other features about the customer reviews and product information. This will help the machine to learn much more complex patterns and avoid the chances of underfitting.
- As a future goal, it is better to include many more product categories and collect data from many other e-commerce websites which will help us to collect powerful dataset which can be used for building a powerful predictive model.
- In future as the technologies are advancing, we can use many advanced features which can process the text in a way that machine can learn the complex text patters without losing out data from the dataset.

Thank You