FLIP ROBO

Rating Prediction Project

Submitted by:

ROHIT KATTEWAR

ACKNOWLEDGEMENT

I would like to thank and express my sincere gratitude to Flip Robo Technologies for giving me the opportunity to work on this project named ‘Ratings Prediction’ using Machine Learning algorithms and NLTK suite.

Primarily, I would like to thank to the author of the paper titled: “Review-Based Rating Prediction” for providing me with invaluable knowledge and insights into the importance of contextual information of user sentiments in determining the rating of products, the role of Natural Language Processing tools and techniques in identifying the user sentiments towards various products based on their reviews and ratings and in helping build models to predict user ratings based on the input reviews.

Finally, I will thank my mentors, under whose guidance I learned a lot about Machine Learning, Natural Language Processing and much more.

INTRODUCTION

❖ Business Problem Framing

A website has a forum for writing technical reviews of products and consists of repository of 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. An application to predict the rating by seeing the review is required to be built.
Therefore, a predictive model to accurately predict a user's rating based on input review is required to be made.

❖ Conceptual Background of the Domain Problem

Predictive modelling, Classification algorithms are some of the machine learning techniques used along with the various libraries of the NLTK suite for Classification of comments.

Using NLTK tools, the frequencies of malignant words occurring in textual data were estimated and given appropriate weightage, whilst filtering out words, and other noise which do not have any impact on the semantics of the comments and reducing the words to their base lemmas for efficient processing and accurate classification of the comments.

❖ Review of the Literature

The given dataset is a part of the research paper named “Review Based Rating Prediction” by Tal Hadad. It was reviewed and studied to gain insights into the importance of contextual information of user sentiments in determining the rating of products, the role of natural language processing tools and techniques to identify the user sentiments towards the various product based on their reviews and ratings. The Contextual information about a user’s opinion of a product can be explicit or implicit and can be inferred in different ways such as user score ratings and textual reviews.

❖ Motivation for the Problem Undertaken

The motivation behind this is to understand the user experience and sentiments towards the product. The sentiment of a user towards a product is reflected in their rating score and their review of the product. This shows how the product is perceived by the consumers and in turn gives an idea about the acceptance of the product by the consumers. This study will help us understand the product’s worth and acceptance by the customer.

ANALYTICAL PROBLEM FRAMING

❖ Mathematical/Analytical Modeling of the Problem:

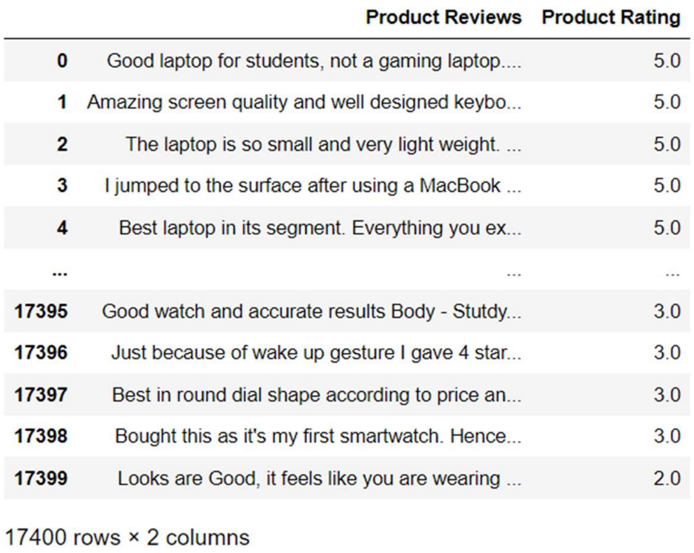
The process of building a predictive model, in this case, different classification analysis techniques were used to build such model. The model will help us understand the relationship between the user reviews of the particular product and the actual product offerings. To predict the ratings for the product, different models such as Logistic regression, Random Forest Classifier Boost Classifier, Extreme Gradient Boost Classifier, Multinomial Naïve Bayes Classifier, Complement Naïve Bayes Classifier and Passive Aggressive Classifier were used.

❖ Data Sources and their Formats:

The dataset is created by using Web Scrapping technique to scrape the user reviews and ratings for the different set of products from https://www.amazon.in/ e-commerce website.

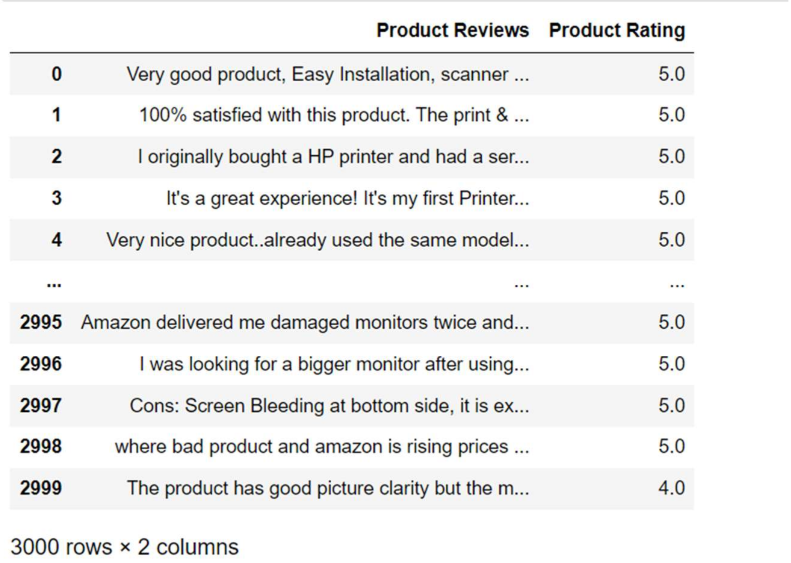
❖ Dataset Description:

The dataset is created by using web scrapping technique which involve the Selenium which is an automation software to scrape the product reviews and ratings. I, firstly, scrapped 17400 reviews and ratings of the multiple products from Amazon.in then I scrapped some more products reviews and ratings for the different set of products than the previous ones. I combined the both the data frames into one and made the sole file for the reviews and ratings for the products.



	Product Reviews	Product Rating
0	Good laptop for students, not a gaming laptop....	5.0
1	Amazing screen quality and well designed keybo...	5.0
2	The laptop is so small and very light weight. ...	5.0
3	I jumped to the surface after using a MacBook ...	5.0
4	Best laptop in its segment. Everything you ex...	5.0
...	...	...
17395	Good watch and accurate results Body - Stutdy...	3.0
17396	Just because of wake up gesture I gave 4 star...	3.0
17397	Best in round dial shape according to price an...	3.0
17398	Bought this as it's my first smartwatch. Hence...	3.0
17399	Looks are Good, it feels like you are wearing ...	2.0

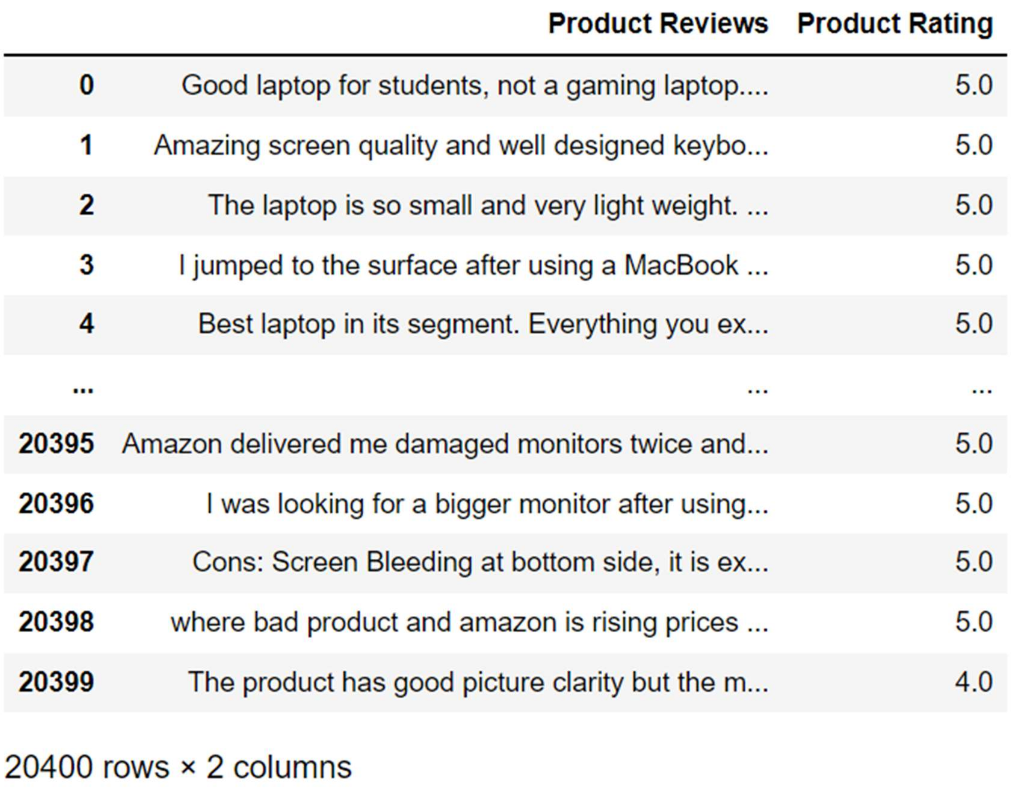
17400 rows × 2 columns



	Product Reviews	Product Rating
0	Very good product, Easy Installation, scanner ...	5.0
1	100% satisfied with this product. The print & ...	5.0
2	I originally bought a HP printer and had a ser...	5.0
3	It's a great experience! It's my first Printer...	5.0
4	Very nice product..already used the same model...	5.0
...	...	...
2995	Amazon delivered me damaged monitors twice and...	5.0
2996	I was looking for a bigger monitor after using...	5.0
2997	Cons: Screen Bleeding at bottom side, it is ex...	5.0
2998	where bad product and amazon is rising prices ...	5.0
2999	The product has good picture clarity but the m...	4.0

3000 rows × 2 columns

We combined the above two data frames and merged them.



	Product Reviews	Product Rating
0	Good laptop for students, not a gaming laptop....	5.0
1	Amazing screen quality and well designed keybo...	5.0
2	The laptop is so small and very light weight. ...	5.0
3	I jumped to the surface after using a MacBook ...	5.0
4	Best laptop in its segment. Everything you ex...	5.0
...	...	...
20395	Amazon delivered me damaged monitors twice and...	5.0
20396	I was looking for a bigger monitor after using...	5.0
20397	Cons: Screen Bleeding at bottom side, it is ex...	5.0
20398	where bad product and amazon is rising prices ...	5.0
20399	The product has good picture clarity but the m...	4.0

20400 rows × 2 columns

The image above shows the combined data frame which has 20400 the reviews and ratings of the various products.

The Columns are:

1. **Product Reviews**: User review of the Product

2. **Product Rating**: User rating of the Product

❖ Data Pre-processing Done:

1. Rows with null values were removed.

2. Columns: Unnamed: 0(just a series of numbers) was dropped since it doesn't contribute to building a good model for predicting the target variable values.

3. The train and test dataset contents were then converted into lowercase.

4. Punctuations, unnecessary characters etc were removed, currency symbols, phone numbers, web URLs, email addresses etc were replaced with single words.

5. Tokens that contributed nothing to semantics of the messages were removed as Stop words. Finally retained tokens were lemmatized using WordNetLemmatizer().

6. The string lengths of original comments and the cleaned comments were then compared.

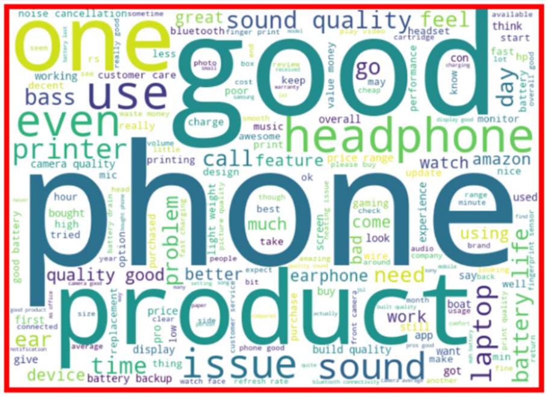
❖ Data Inputs- Logic- Output Relationships:

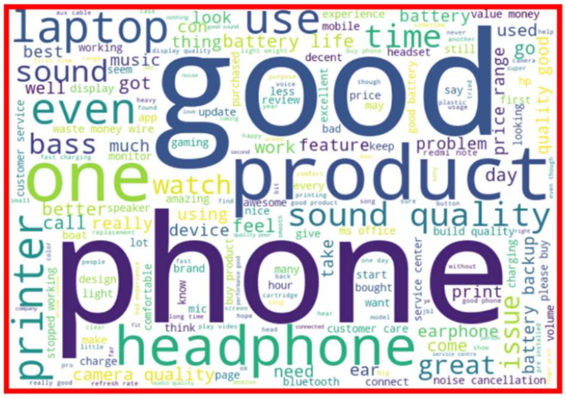
The comment tokens so vectorised using TF-IDF Vectorizer are input and the corresponding rating is predicted based on their context as output by classification models.

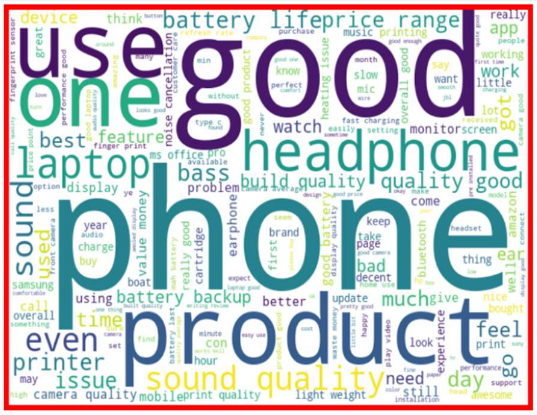
❖ Exploratory Data Analysis:

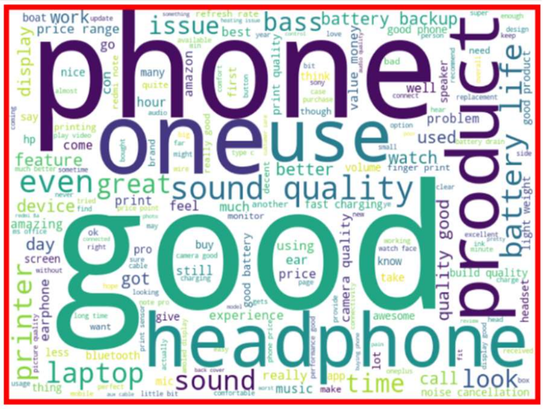
Visualizations: Bar plots, Count plots, Word Clouds were used to visualise the data of all the columns and their relationships with Target variable.

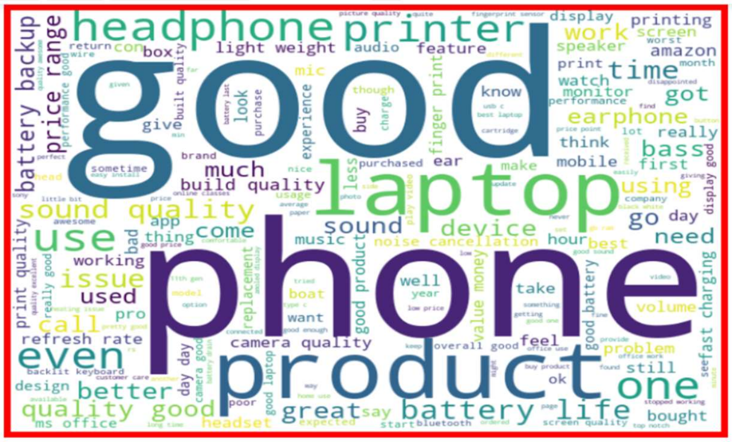
Word Clouds of the most frequent words used in reviews corresponding to the various Rating Scores:











From the graphs above the following observations are made:

1. Reviews corresponding to 5.0 rating frequently carry words like: ‘great’,’best’,’perfect’,’better,’good’ etc indicating very high customer satisfaction and high-quality product.

2. Reviews corresponding to 4.0 rating frequently carry words like: ‘good’,’better’,’nice’,’value money’,’look’, ‘quality’,’awesome’ etc indicating high customer satisfaction and good quality product.

3. Reviews corresponding to 3.0 rating frequently carry words like: ‘good’,’well’,’sound,’bad quality’,’issue’ etc indicating customer dissatisfaction and average to below average product quality.

4. Reviews corresponding to 2.0 rating frequently carry words like: ‘problem’, ‘replacement,’stopped working’,’worst experience’,’quality’ etc indicating high customer dissatisfaction and below average product quality.

5. Reviews corresponding to 1.0 rating frequently carry words like: ‘issue sound’,’working’,’cheap’,’return’,’issue’,’waste money’,’poor quality’,’customer care’,’bad’,’used’,’worst’, ‘poor build quality’ etc indicate very high customer dissatisfaction and poor quality product.

❖ Model Building:

Following are the model algorithms that were used:

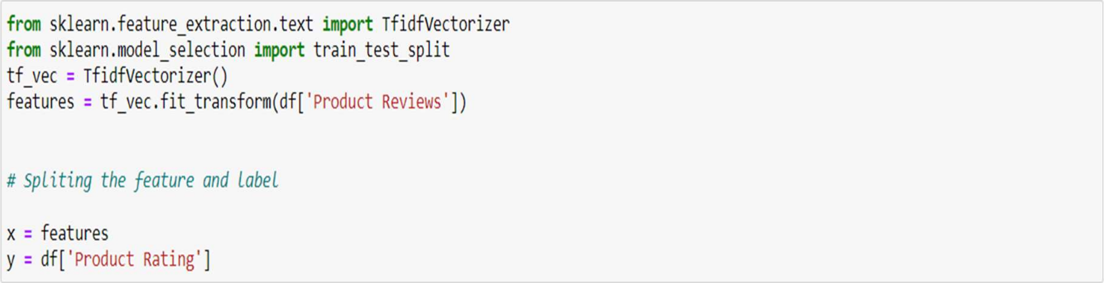
1. Logistic Regression: It is a classification algorithm used to find the probability of event success and event failure. It is used when the dependent variable is binary(0/1, True/False, Yes/No) in nature. It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data. It learns a linear relationship from the given dataset and then introduces a nonlinearity in the form of the Sigmoid function. It not only provides a measure of how appropriate a predictor(coefficient size)is, but also its direction of association (positive or negative).
2. Multinomial Naïve Bayes Classifier: Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

3. Random Forest Classifier: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A random forest produces good predictions that can be understood easily. It reduces overfitting and can handle large datasets efficiently. The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.
4. Complement Naïve Bayes Classifier: Complement Naive Bayes is somewhat an adaptation of the standard Multinomial Naive Bayes algorithm. Complement Naive Bayes is particularly suited to work with imbalanced datasets. In complement Naive Bayes, instead of calculating the probability of an item belonging to a certain class, we calculate the probability of the item belonging to all the classes

5. Passive Aggressive Classifier: Passive-Aggressive algorithms do not require a learning rate and are called so because if the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model. If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

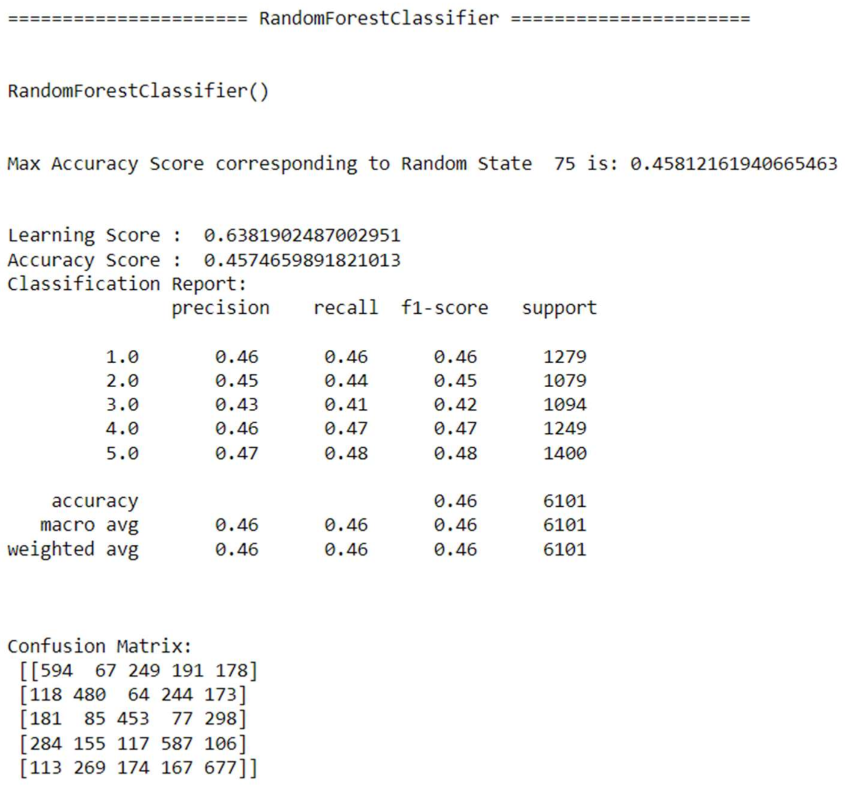
6. AdaBoost Classifier: The basis of this algorithm is the Boosting main core: give more weight to the misclassified observations. the meta-learner adapts based upon the results of the weak classifiers, giving more weight to the misclassified observations of the last weak learner. The individual learners can be weak, but as long as the performance of each weak learner is better than random guessing, the final model can converge to a strong learner (a learner not influenced by outliers and with a great generalization power, in order to have strong performances on unknown data)

Train Test Split the data:

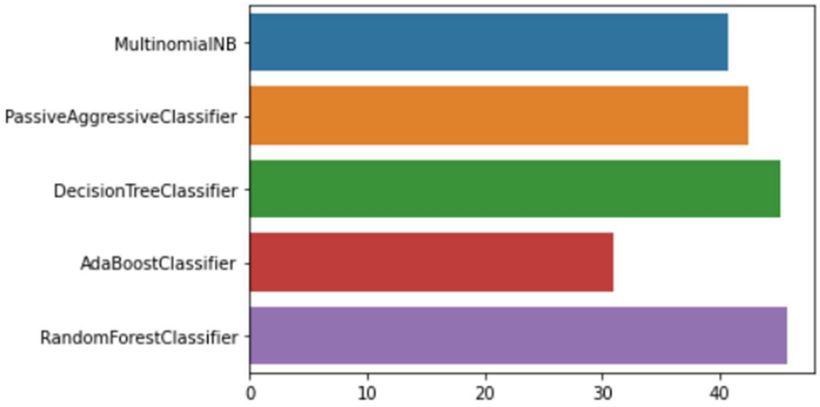


```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
tf_vec = TfidfVectorizer()
features = tf_vec.fit_transform(df['Product Reviews'])


# Spliting the feature and label

x = features
y = df['Product Rating']
```

Best Random State for Random Forest Model:

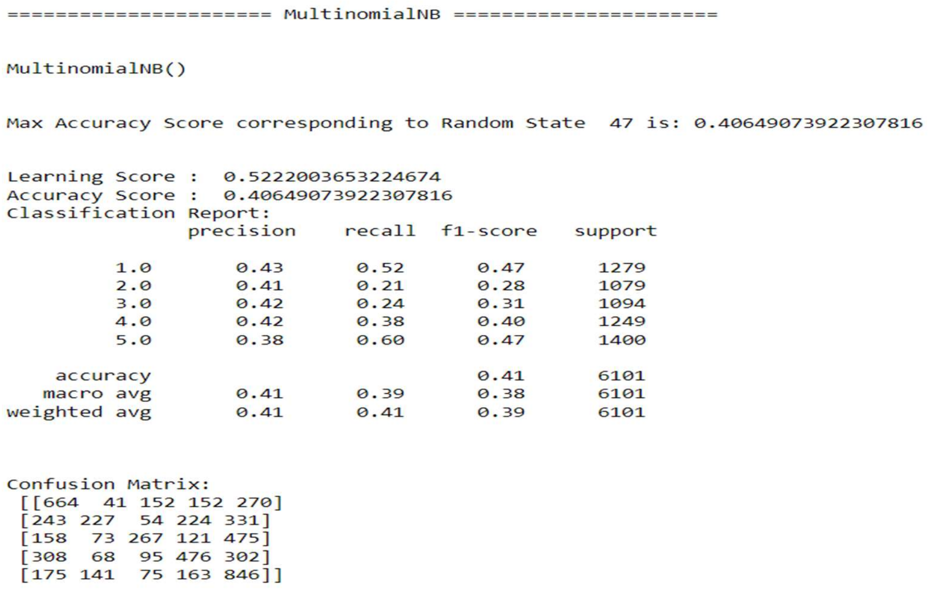


```
===================== RandomForestClassifier =====================


RandomForestClassifier()


Max Accuracy Score corresponding to Random State  75 is: 0.45812161940665463


Learning Score :  0.6381902487002951
Accuracy Score :  0.4574659891821013
Classification Report:
              precision    recall  f1-score   support

         1.0       0.46      0.46      0.46      1279
         2.0       0.45      0.44      0.45      1079
         3.0       0.43      0.41      0.42      1094
         4.0       0.46      0.47      0.47      1249
         5.0       0.47      0.48      0.48      1400

    accuracy                           0.46      6101
   macro avg       0.46      0.46      0.46      6101
weighted avg       0.46      0.46      0.46      6101



Confusion Matrix:
 [[594  67 249 191 178]
 [118 480  64 244 173]
 [181  85 453  77 298]
 [284 155 117 587 106]
 [113 269 174 167 677]]
```

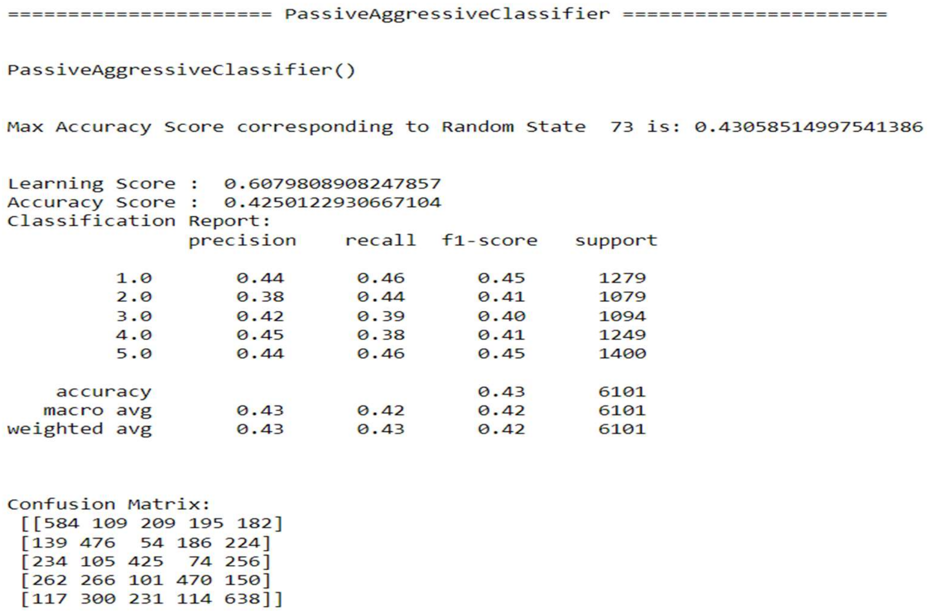


Above image shows the different accuracy scores of the different models.

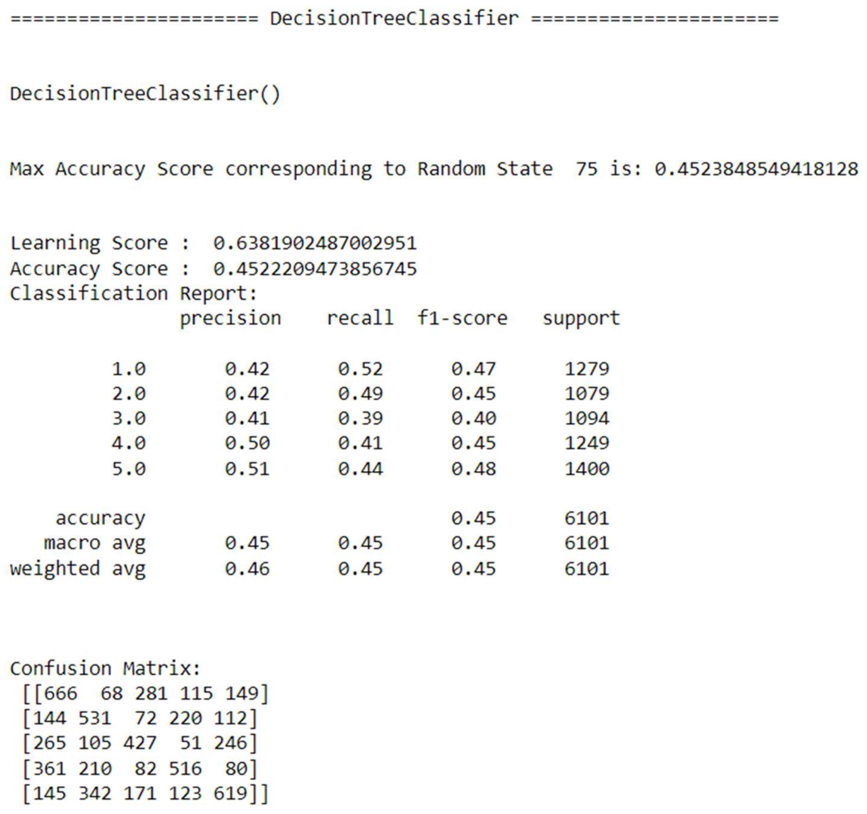
Analysing the accuracy of the models:



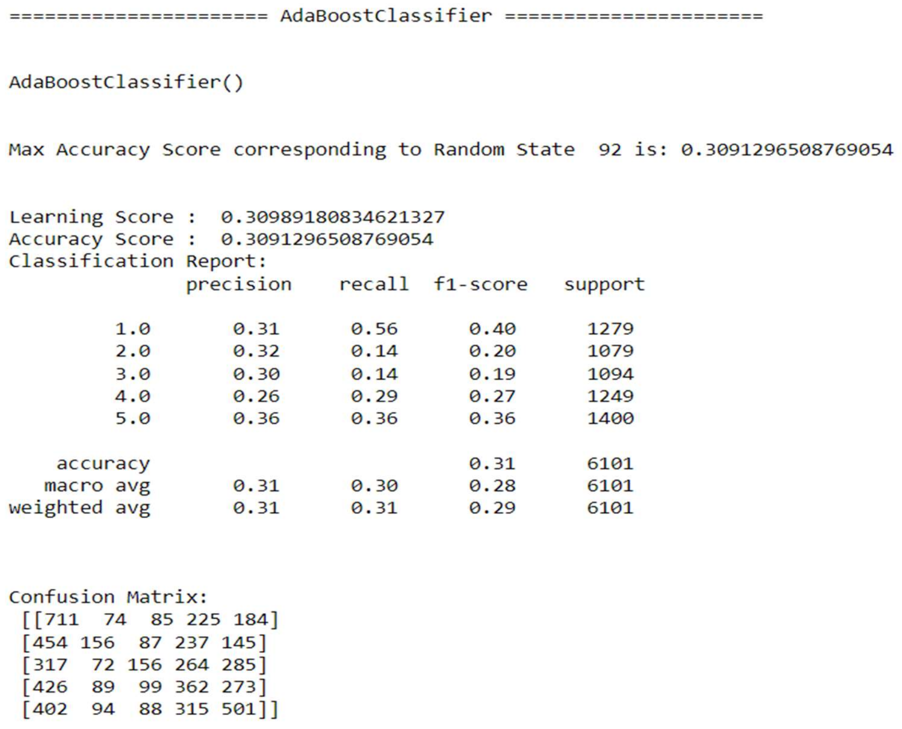
```
====================== MultinomialNB ======================


MultinomialNB()


Max Accuracy Score corresponding to Random State  47 is: 0.40649073922307816


Learning Score :  0.5222003653224674
Accuracy Score :  0.40649073922307816
Classification Report:
               precision    recall  f1-score   support

         1.0       0.43      0.52      0.47      1279
         2.0       0.41      0.21      0.28      1079
         3.0       0.42      0.24      0.31      1094
         4.0       0.42      0.38      0.40      1249
         5.0       0.38      0.60      0.47      1400

    accuracy                           0.41      6101
   macro avg       0.41      0.39      0.38      6101
weighted avg       0.41      0.41      0.39      6101


Confusion Matrix:
 [[664  41 152 152 270]
 [243 227  54 224 331]
 [158  73 267 121 475]
 [308  68  95 476 302]
 [175 141  75 163 846]]
```



```
====================== PassiveAggressiveClassifier ======================


PassiveAggressiveClassifier()


Max Accuracy Score corresponding to Random State  73 is: 0.43058514997541386


Learning Score :  0.6079808908247857
Accuracy Score :  0.4250122930667104
Classification Report:
               precision    recall  f1-score   support

         1.0       0.44      0.46      0.45      1279
         2.0       0.38      0.44      0.41      1079
         3.0       0.42      0.39      0.40      1094
         4.0       0.45      0.38      0.41      1249
         5.0       0.44      0.46      0.45      1400

    accuracy                           0.43      6101
   macro avg       0.43      0.42      0.42      6101
weighted avg       0.43      0.43      0.42      6101


Confusion Matrix:
 [[584 109 209 195 182]
 [139 476  54 186 224]
 [234 105 425  74 256]
 [262 266 101 470 150]
 [117 300 231 114 638]]
```

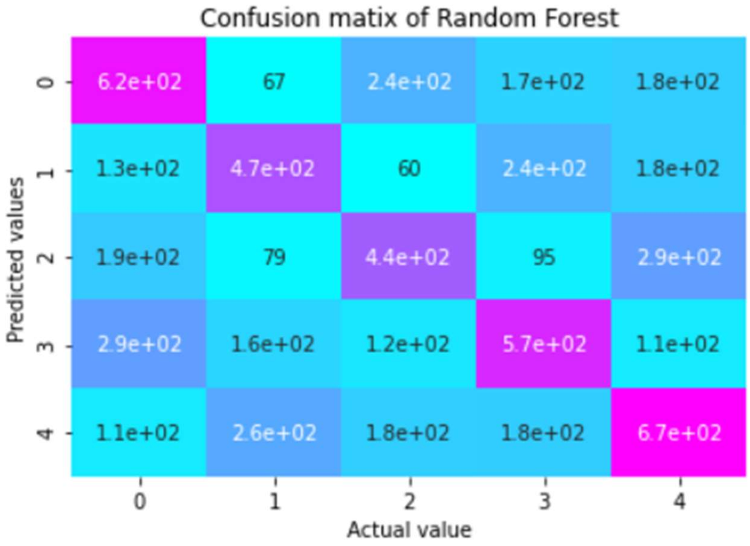


```
===================== DecisionTreeClassifier =====================


DecisionTreeClassifier()


Max Accuracy Score corresponding to Random State  75 is: 0.4523848549418128


Learning Score :  0.6381902487002951
Accuracy Score :  0.4522209473856745
Classification Report:
               precision    recall  f1-score   support

         1.0       0.42      0.52      0.47      1279
         2.0       0.42      0.49      0.45      1079
         3.0       0.41      0.39      0.40      1094
         4.0       0.50      0.41      0.45      1249
         5.0       0.51      0.44      0.48      1400

    accuracy                           0.45      6101
   macro avg       0.45      0.45      0.45      6101
weighted avg       0.46      0.45      0.45      6101



Confusion Matrix:
 [[666  68 281 115 149]
 [144 531  72 220 112]
 [265 105 427  51 246]
 [361 210  82 516  80]
 [145 342 171 123 619]]
```



```
===================== AdaBoostClassifier =====================


AdaBoostClassifier()


Max Accuracy Score corresponding to Random State  92 is: 0.3091296508769054


Learning Score :  0.30989180834621327
Accuracy Score :  0.3091296508769054
Classification Report:
               precision    recall  f1-score   support

         1.0       0.31      0.56      0.40      1279
         2.0       0.32      0.14      0.20      1079
         3.0       0.30      0.14      0.19      1094
         4.0       0.26      0.29      0.27      1249
         5.0       0.36      0.36      0.36      1400

    accuracy                           0.31      6101
   macro avg       0.31      0.30      0.28      6101
weighted avg       0.31      0.31      0.29      6101


Confusion Matrix:
 [[711  74  85 225 184]
 [454 156  87 237 145]
 [317  72 156 264 285]
 [426  89  99 362 273]
 [402  94  88 315 501]]
```

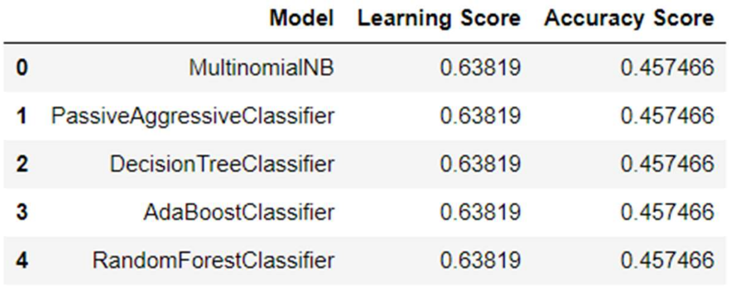
Confusion metrix for the Random Forest Model:



	0	1	2	3	4
0	6.2e+02	67	2.4e+02	1.7e+02	1.8e+02
1	1.3e+02	4.7e+02	60	2.4e+02	1.8e+02
2	1.9e+02	79	4.4e+02	95	2.9e+02
3	2.9e+02	1.6e+02	1.2e+02	5.7e+02	1.1e+02
4	1.1e+02	2.6e+02	1.8e+02	1.8e+02	6.7e+02

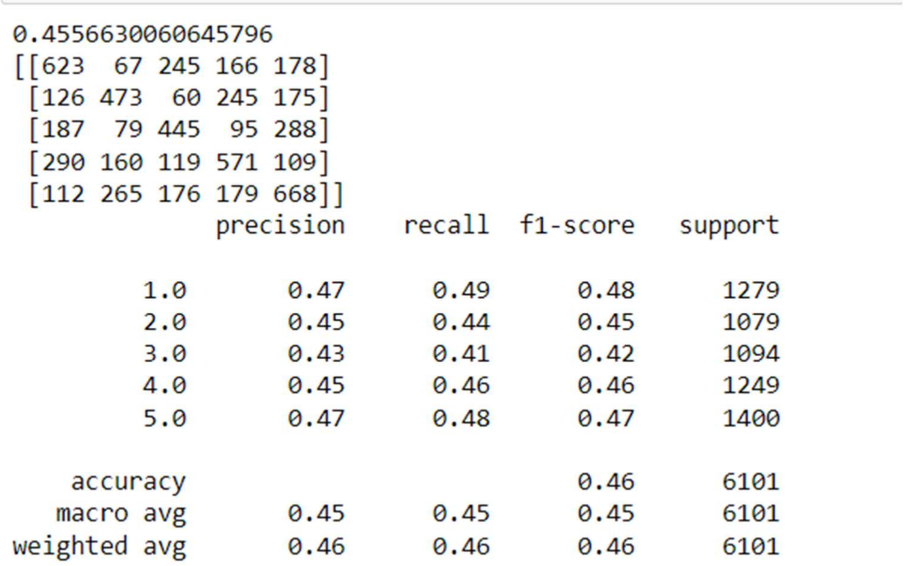
Conclusion:

The following image shows the Learning Score and the Accuracy Score of the different models.

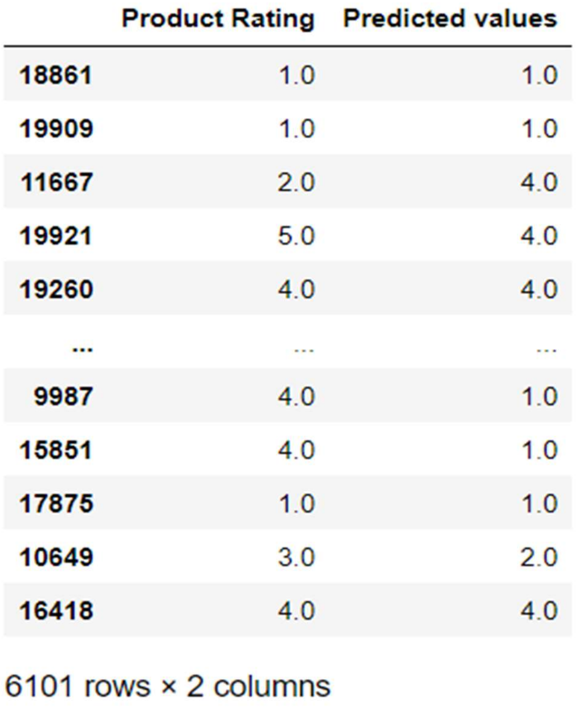


	Model	Learning Score	Accuracy Score
0	MultinomialNB	0.63819	0.457466
1	PassiveAggressiveClassifier	0.63819	0.457466
2	DecisionTreeClassifier	0.63819	0.457466
3	AdaBoostClassifier	0.63819	0.457466
4	RandomForestClassifier	0.63819	0.457466

The following image shows the predicted values after training the model:



```
0.4556630060645796
[[623  67 245 166 178]
 [126 473  60 245 175]
 [187  79 445  95 288]
 [290 160 119 571 109]
 [112 265 176 179 668]]
              precision    recall  f1-score   support

         1.0       0.47      0.49      0.48      1279
         2.0       0.45      0.44      0.45      1079
         3.0       0.43      0.41      0.42      1094
         4.0       0.45      0.46      0.46      1249
         5.0       0.47      0.48      0.47      1400

    accuracy                           0.46      6101
   macro avg       0.45      0.45      0.45      6101
weighted avg       0.46      0.46      0.46      6101
```



	Product Rating	Predicted values
18861	1.0	1.0
19909	1.0	1.0
11667	2.0	4.0
19921	5.0	4.0
19260	4.0	4.0
...	...	...
9987	4.0	1.0
15851	4.0	1.0
17875	1.0	1.0
10649	3.0	2.0
16418	4.0	4.0

6101 rows × 2 columns

1. The various data pre-processing and feature engineering steps in the project lent cognizance to various efficient methods for processing textual data. The NLTK suite is very useful in pre-processing text-based data and building classification models.

2. Data cleaning was a very important step in removing null values from the dataset.
3. By training the models on more diverse data sets, longer comments, and a more balanced dataset, more accurate and efficient classification models can be built.

----- --:-- ----- ----- --:-- ----- ❄ ----- --:-- ---------- --:-- -----