

**“Rating Prediction Project”**

**SUBMITTED BY**

**Abhinav Verma**

**ACKNOWLEDGMENT**

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Sapna Verma (SME Flip Robo), she is the person who has helped me to get out of all the difficulties I faced while doing the project.

A huge thanks to my academic team “Data trained” who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life. And also thank you for many other persons who has helped me directly or indirectly to complete the project.

**Business Problem Framing:**

Rating prediction is a well-known recommendation task aiming to predict a user’s rating for those items which were not rated yet by her. Predictions are computed from users’ explicit feedback, i.e. their ratings provided on some items in the past. Another type of feedback are user reviews provided on items which implicitly express users’ opinions on items. Recent studies indicate that opinions inferred from users’ reviews on items are strong predictors of user’s implicit feedback or even ratings and thus, should be utilized in computation. The rise in E-commerce has brought a significant rise in the importance of customer reviews. There are hundreds of review sites online and massive amounts of reviews for every product. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches. The ability to successfully decide whether a review will be helpful to other customers and

thus give the product more exposure is vital to companies that support these reviews, companies like Google, Amazon and Yelp!. There are two main methods to approach this problem. The first one is based on review text content analysis and uses the principles of natural language process (the NLP method). This method lacks the insights that can be drawn from the relationship between costumers and items. The second one is based on recommender systems, specifically on collaborative filtering, and focuses on the reviewer’s point of view. 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 rating. So, we have to build an application which can predict the rating by seeing the review.

**Motivation for the Problem Undertaken:**

The project was first provided to me by FlipRobo as a part of the internship program. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary objective. 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. However, the motivation for taking this project was that it is relatively a new field of research. Here we have many options but less concrete solutions. The main motivation is to build a prototype of online hate and abuse review classifier which can used to classify hate and good comments so that it can be controlled and corrected according to the reviewer’s choice.

**2.Analytical Problem Framing**

**Mathematical/ Analytical Modeling of the Problem:**

In this perticular problem the Ratings can be 1, 2, 3, 4 or 5, which represents the likely ness of the product to the customer. So clearly it is a multi classification problem and I have to use all classification algorithms while building the model.

We would perform one type of supervised learning algorithms: Classification. Here, we will only perform classification. Since there only 1 feature in the dataset, filtering the words is needed to prevent overfit. In order to determine the regularization parameter, throughout the project in classification part, we would first remove email, phone number, web address, spaces and stops words etc. In order to further improve our models, we also performed TFID in order to convert the tokens from the train documents into vectors so that machine can do further processing. I have used all the classification algorithms while building model then tunned the best model and saved the best model.

**Data Sources and their formats:**

The data set contains 23,405 samples with 3 features. Since **Ratings** is my target column and it is a categorical column with 5 categories so this problem is a **Multi Classification Problem**. The Ratings can be 1, 2, 3, 4 or 5, which represents the likely ness of the product to the customer. The data set includes:

• Review\_Title : Title of the Review.

• Review\_Text : Text Content of the Review.

• Ratings : Ratings out of 5 stars.

This project is more about exploration, feature engineering and classification that can be done on this data. Since the data set is huge and includes multi classification of ratings, we can do good amount of data exploration and derive some interesting features using the Review column available.

We need to build a model that can predict Ratings of the reviewer.

**Hardware & Software Requirements & Tools Used:**

While taking up the project we should be familiar with the Hardware and software required for the successful completion of the project. Here we need the following hardware and software.

**Hardware required**

Processor: core i5

RAM: 8 GB

SSD: 512 GB

**Software/s required**

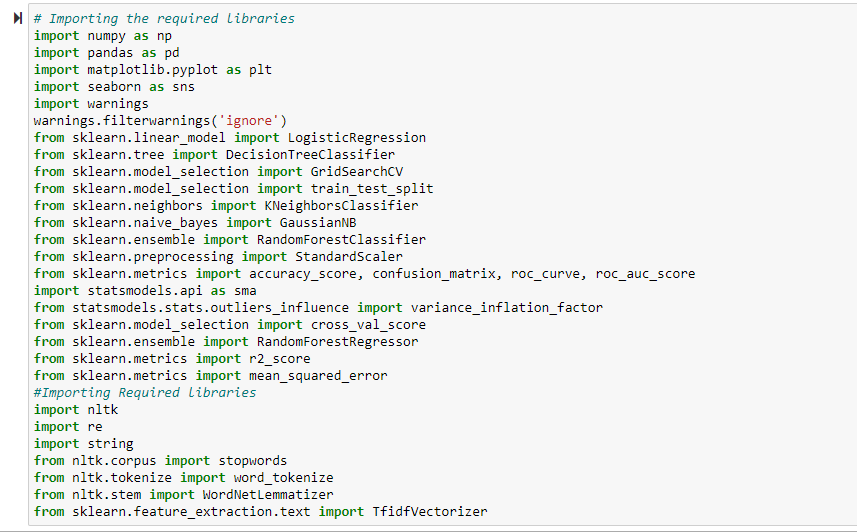
Distribution: Anaconda Navigator

Programming language: Python

Browser based language shell: Jupyter Notebook

**Libraries required :-**

To run the program and to build the model we need some basic libraries as follows:



**3.Data Analysis and Visualization**

**Testing of Identified Approaches (Algorithms)**

In this nlp based project we need to predict Ratings which is a multiclassification problem. I have converted the text into vectors using TFIDF vectorizer and separated our feature and labels then build the model using One Vs Rest Classifier. Among all the algorithms which I have used for this purpose I have chosen SGDClassifier as best suitable algorithm for our final model as it is performing well compared to other algorithms while evaluating with different metrics I have used following algorithms and evaluated them

➢ KNN Classifier

➢ LogisticRegression

➢ RandomForestClassifier

➢ DecisionTreeClassifier

From all of these above models RandomForestClassifier was giving me good performance with less difference in accuracy score and cv score.

**Key Metrics for success in solving problem under consideration:**

I have used the following metrics for evaluation:

• I have used f1\_score, precision\_score, recall\_score, multilabel\_confusion\_matrix and hamming loss all these evaluation metrics to select best suitable algorithm for our final model.

• **Precision** can be seen as a measure of quality, higher precision means that an algorithm returns more relevant results than irrelevant ones.

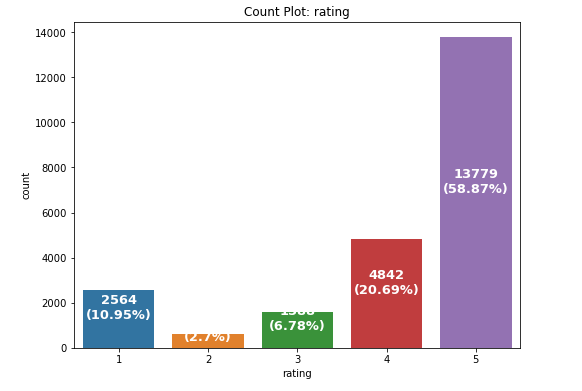
• **Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.

• **Accuracy score** is used when the True Positives and True negatives are more important. Accuracy can be used when the class distribution is similar.

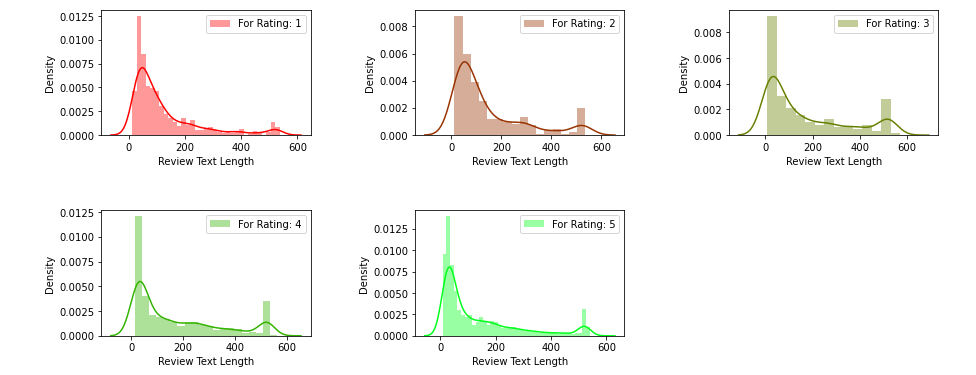
• **F1**-**score** is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.

**Visualizations:**

Count plot for Rating:



checking review text length distribution for each rating:

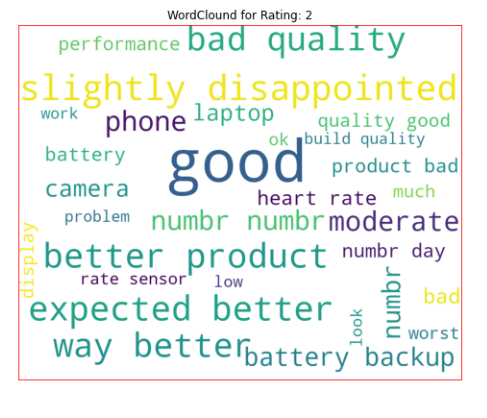


**Word Cloud for particular ratings:**

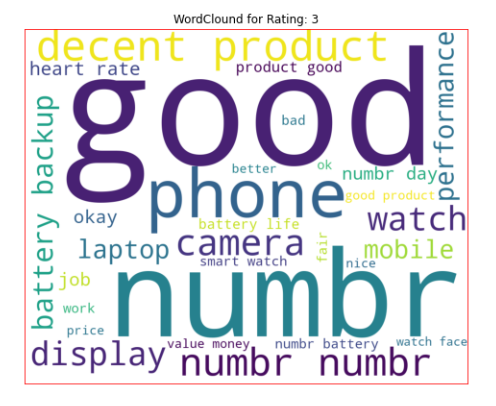
**Rating 1:**



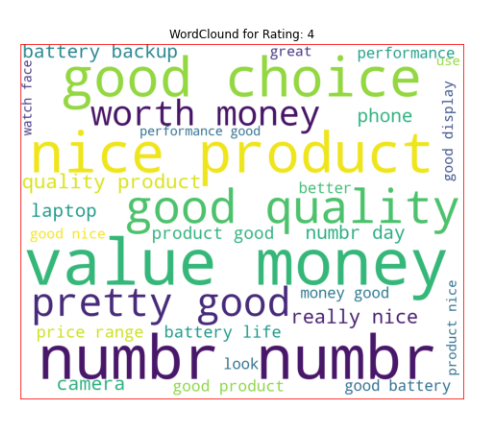
**Rating 2:**



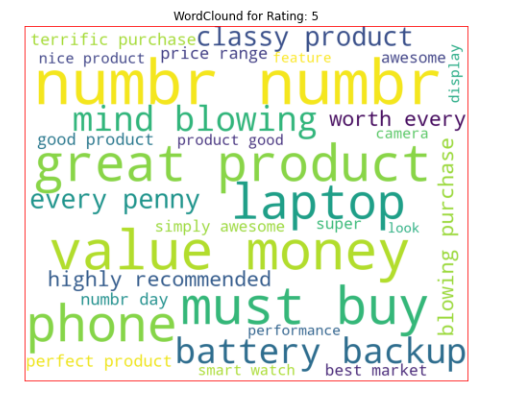
**Rating 3 :**



**Rating 4 :**



**Rating 5 :**



• From the above plots we can clearly see the words which are indication

of Reviewer's opinion on products.

• Here most frequent words used for each Rating is displayed in the word

cloud.

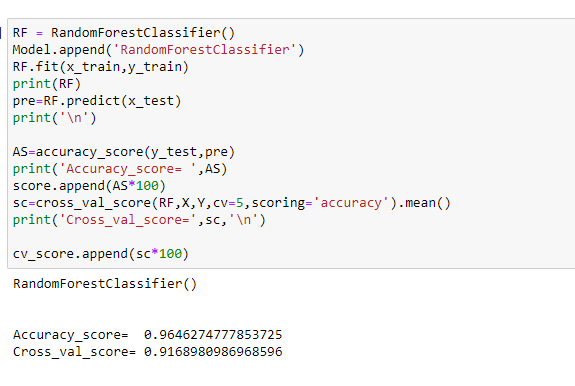
**Model Building:**

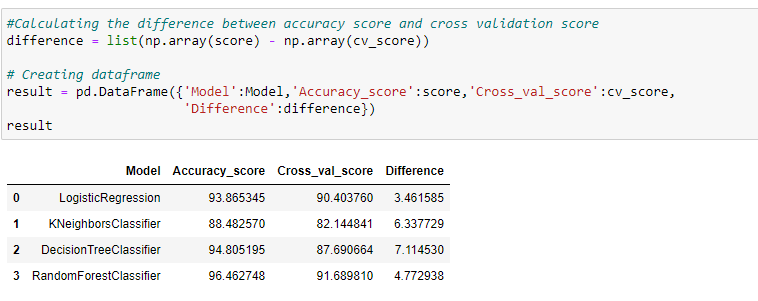
I have used 4 classification algorithms. First, I have created 6 different classification algorithms and are appended in the variable models. Followed by TFIDF vectorization and data balancing. Then, ran a for loop which contained the accuracy of the models along with different evaluation metrics.



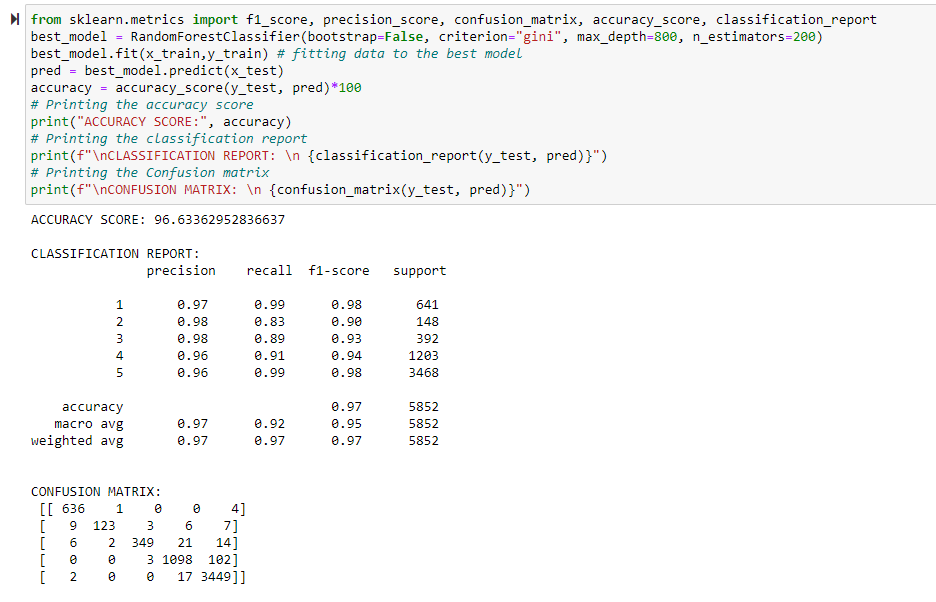








**Grid search CV:**



**Saving the model and Predictions:**



**CONCLUSION:**

• In this project I have collected data of reviews and ratings for different

products from amazon.in and flipkart.com.

• 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 the users. We made use of natural

language processing and machine learning algorithms in order to do so.

• Then I have done different text processing for reviews column and chose

equal number of text from each rating class to eliminate problem of

imbalance. By doing different EDA steps I have analysed the text.

• We have checked frequently occurring words in our data as well as rarely

occurring words.

• After all these steps I have built function to train and test different

algorithms and using various evaluation metrics I have selected

SGDClassifier for our final model.

• Finally by doing hyperparameter tuning we got optimum parameters for

our final model.