

**PROJECT REPORT:**

Rating Prediction Project

**SUBMITTED BY:**

Ankit Soran

**ACKNOWLEDGMENT**

I would like to express my special gratitude to “FlipRobo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analysing skills. And I want to express my huge gratitude to **Swati Mahaseth** (SME FlipRobo), 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.

Contents:

* Introduction

1. Business Problem Framing
2. Conceptual Background of the Domain Problem
3. Review of Literature
4. Motivation for the Problem Undertaken

* Analytical Problem Framing

1. Mathematical/ Analytical Modelling of the Problem
2. Data Sources and their formats 2.3 Data Pre-processing Done
3. Data Inputs- Logic- Output Relationships
4. Hardware and Software Requirements and Tools Used

* Data Analysis and Visualization

1. Identification of possible problem-solving approaches (methods)
2. Testing of Identified Approaches (Algorithms)
3. Key Metrics for success in solving problem under consideration
4. Visualization
5. Run and Evaluate selected models

* Conclusion

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

* INTRODUCTION

1. Business Problem Framing:

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.

1. Conceptual Background of the Domain Problem

Recommendation systems are an important units in today's e-commerce applications, such as targeted advertising, personalized marketing and information retrieval. In recent years, the importance of contextual information has motivated generation of personalized recommendations according to the available contextual information of users. Compared to the traditional systems which mainly utilize user’s rating history, review-based recommendation hopefully provide more relevant results to users. We introduce a review-based recommendation approach that obtains contextual information by mining user reviews.

The proposed approach relate to features obtained by analysing textual reviews using methods developed in Natural Language Processing (NLP) and information retrieval discipline to compute a utility function over a given item. An item utility is a measure that shows how much it is preferred according to user's current context. In our system, the context inference is modelled as similarity between the user’s reviews history and the item reviews history.

As an example application, we used our method to mine contextual data from customer’s reviews of technical products and use it to produce review-based rating prediction. The predicted ratings can generate recommendations that are item-based and should appear at the recommended items list in the product page. Our evaluations (surprisingly) suggest that our system can help produce better prediction rating scores in comparison to the standard prediction methods.

1. Review of Literature

In real life, people’s decision is often affected by friend’s action or recommendation. How to utilize social information has been extensively studied. Yang et al. propose the concept of “Trust Circles” in social network based on probabilistic matrix factorization. Jiang et al. propose another important factor, the individual preference. Some websites do not always offer structured information, and all of these methods do not leverage user’s unstructured information, i.e. reviews, explicit social networks information is not always available and it is difficult to provide a good prediction for each user. For this problem the sentiment factor term is used to improve social recommendation. The rapid development of Web 2.0 and e-commerce has led to a proliferation in the number of online user reviews. Online reviews contain a wealth of sentiment information that is important for many decision-making processes, such as personal consumption decisions, commodity quality monitoring, and social opinion mining. Mining the sentiment and opinions that are contained in online reviews has become an important topic in natural language processing, machine learning, and Web mining.

1. 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

1. Mathematical/ Analytical modelling of the Problem

In this particular 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 over fit. 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 tuned the best model and saved the best model

1. Data Sources and their formats

The data set contains nearly 94565 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
    - Review\_Text
    - Ratings

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.

1. Data Pre-processing

Data pre-processing is the process of converting raw data into a well readable format to be used by Machine Learning model. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model.

I have used following pre-processing steps:

* Importing necessary libraries and loading dataset as a data frame.
* Checked some statistical information like shape, number of unique values present, info, null values, value counts etc.
* Checked for null values and I replaced those null values using imputation method. And removed Unnamed: 0.
* Visualized each feature using seaborn and matplotlib libraries by plotting distribution plot and wordcloud for each ratings.
* Done text pre-processing techniques like Removing Punctuations and other special characters, splitting the comments into individual words, Removing Stop Words, Stemming and Lemmatization.
* After getting a cleaned data used TF-IDF vectorizer. It’ll help to transform the text data to feature vector which can be used as input in our 6 modelling. It is a common algorithm to transform text into numbers. It measures the originality of a word by comparing the frequency of appearance of a word in a document with the number of documents the words appear in. Mathematically, TF-IDF = TF(t\*d)\*IDF(t,d)
* Balanced the data using SMOTE method.

4 Data Inputs- Logic- Output Relationships

The dataset consists of 2 features with a label. The features are independent and label is dependent as our label varies the values (text) of our independent variables changes.

• I checked the distribution of skewness using dist plots and used count plots to check the counts available in each column as a part of univariate analysis.

• Got to know the frequently occurring and rare occurring word with the help of count plot.

• And was able to see the words in the Review text with reference to their ratings using word cloud.

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:

Core i5 RAM: 6 GB ROM/SSD: 512 GB

Software/s required Processor:

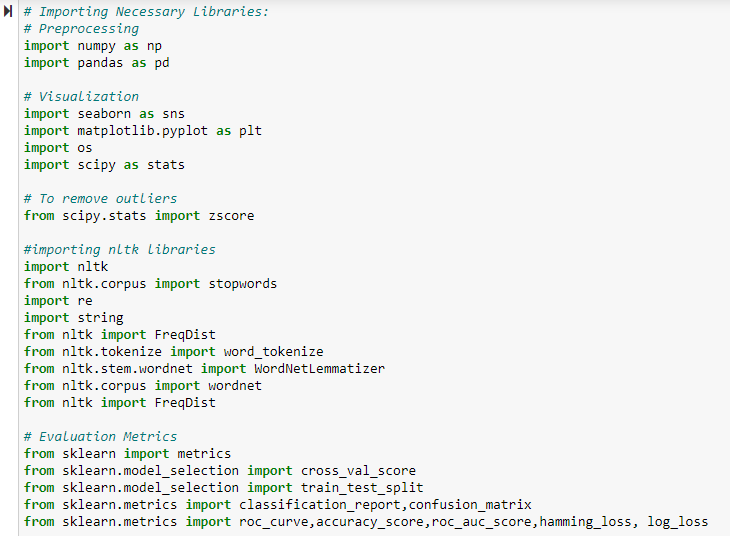
Distribution: Anaconda Navigator

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

1. Identification of possible problem-solving approaches (methods)

I have converted text into feature vectors using TF-IDF vectorizer and separated our feature and labels. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models. Just making the Reviews more appropriate so that we’ll get less word to process and get more accuracy. Removed extra spaces, converted email address into email keyword, and phone number etc. Tried to make Reviews small and more appropriate as much as possible.

1. 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

* LinearSVC
* Logistic Regression
* RandomForestClassifier
* DecisionTreeClassifier
* XGBClassifier
* SGDClassifier

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

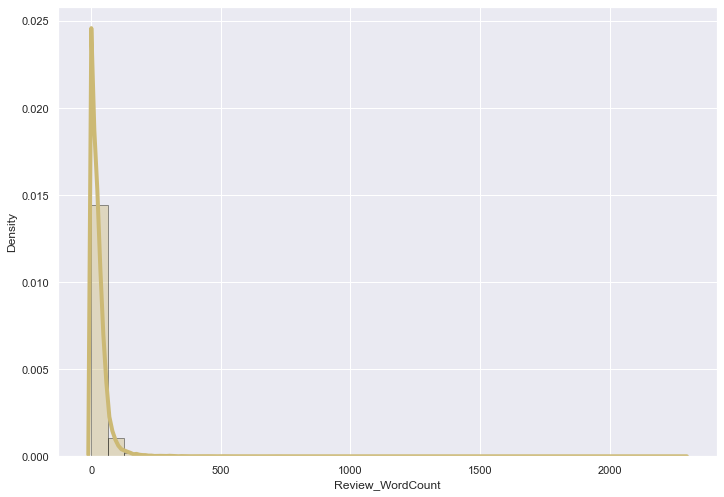
1. Key Metrics for success in solving problem under consideration

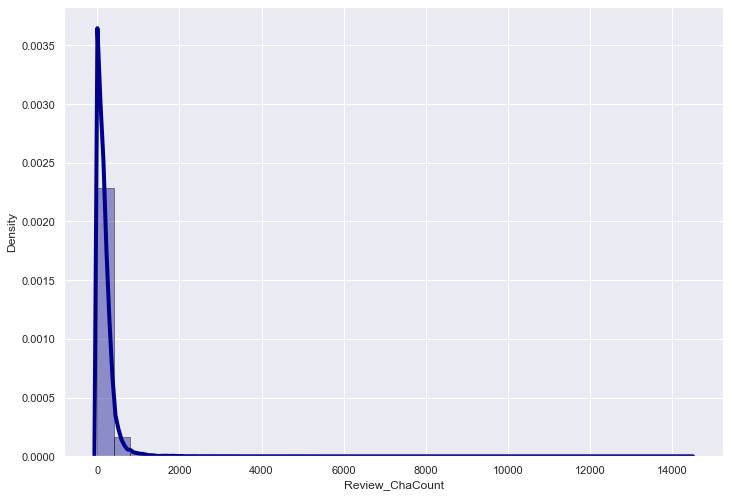
I have used the following metrics for evaluation:

* + 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.

1. Visualizations

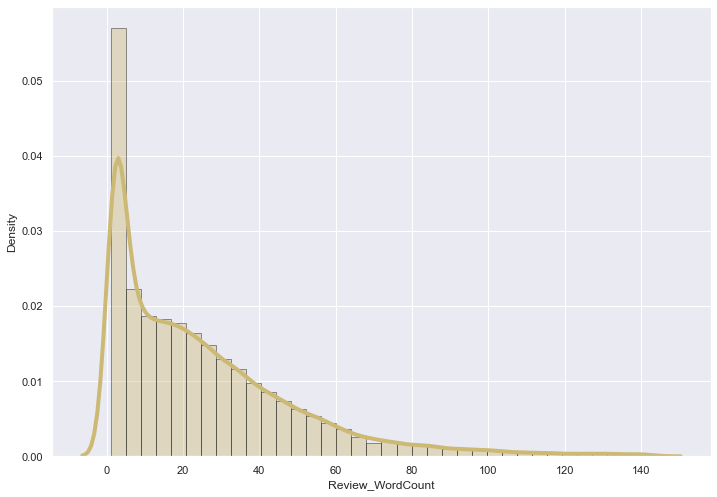
Plotting word count and character count using hist plot:

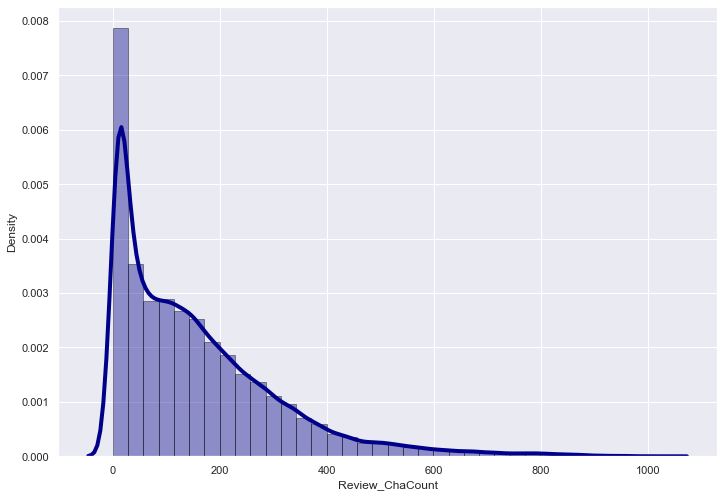




Observations:

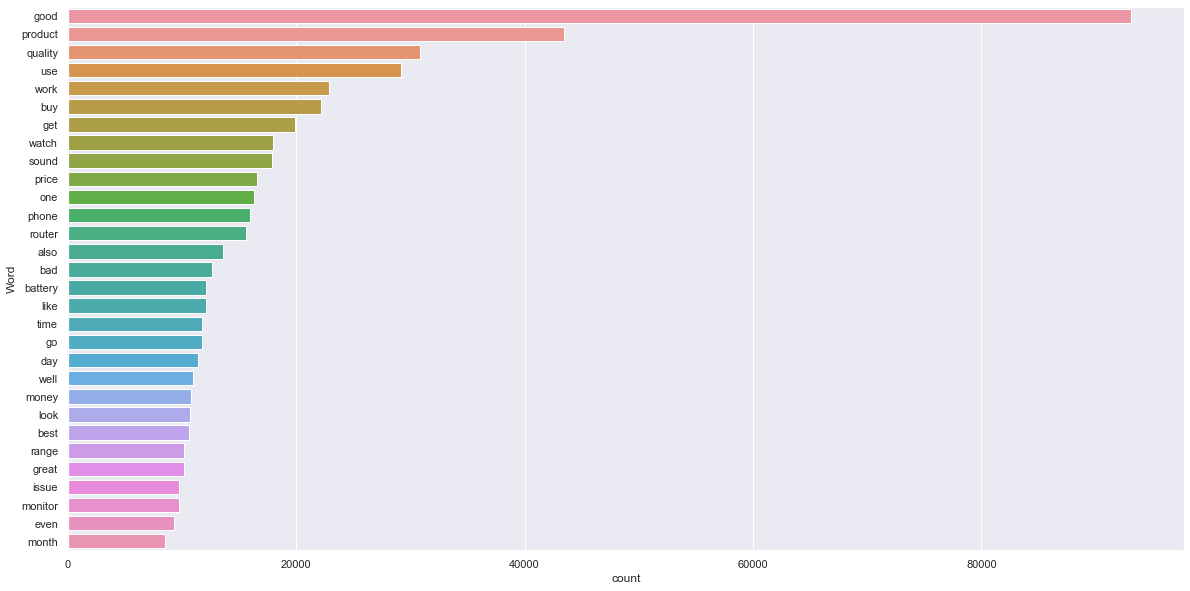
* By observing the histogram we can clearly see that most of our text is having the number of words in the range of 0 to 200, but some of the reviews are too lengthy which may act like outliers in our data.
* Above plot represents histogram for character count of Review text, which is quite similar to the histogram of word count.
* As we know that some of the review are too lengthy, so I have to treat them as outliers and remove them using Z-score method. After removing the outliers the word count and character count looks as below

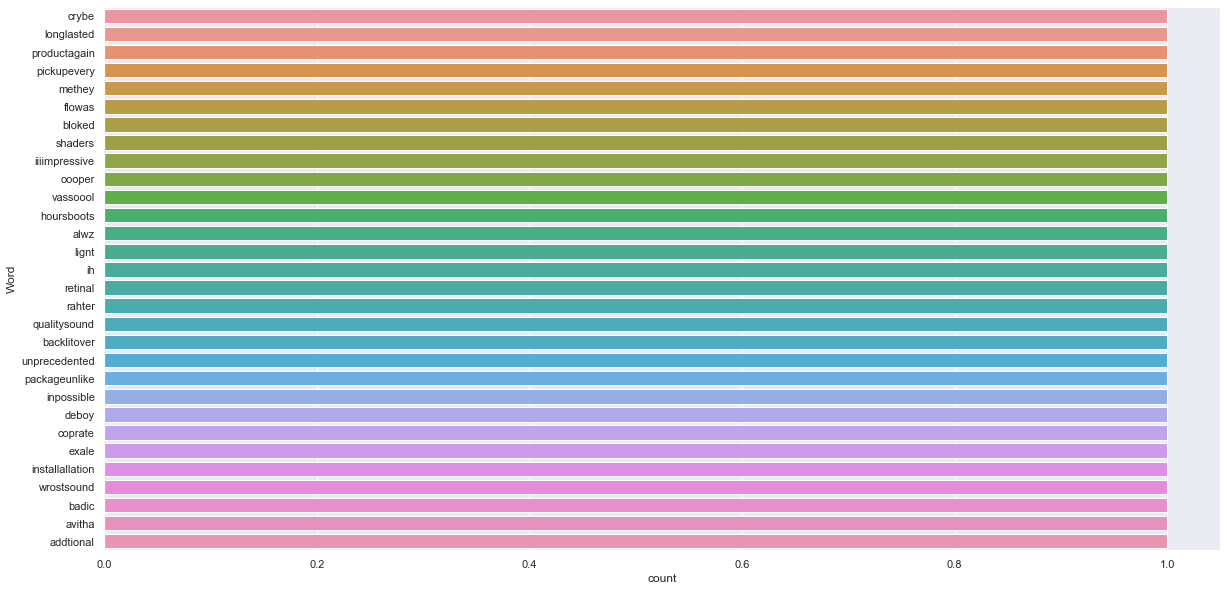




* After plotting histograms for word counts and character counts and after removing outliers we can see we are left out with good range of number of words and characters.

Top 30 most frequently occurring and rarely occurring words:





* By seeing the above plot we can see that Good, product, quality......are occurring frequently. And the second plot shows rarely occurring words.

Word Cloud for particular ratings:











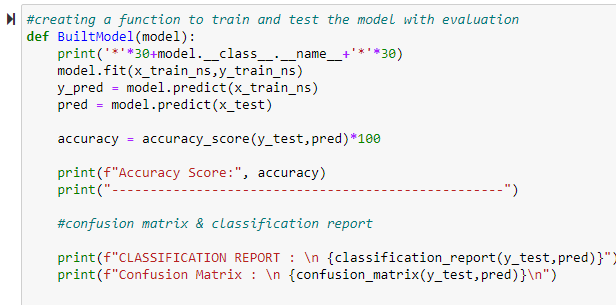
* From the above plots we can clearly see the words which are indication of Reviewer's opinion on products.
* Here most frequent words used for each Rating is displayed in the word cloud.

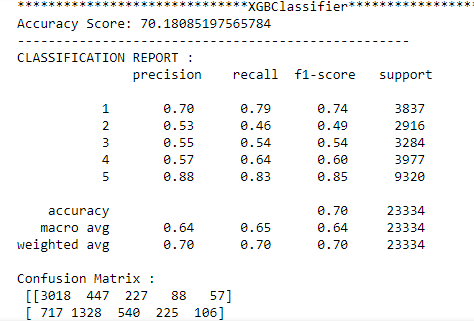
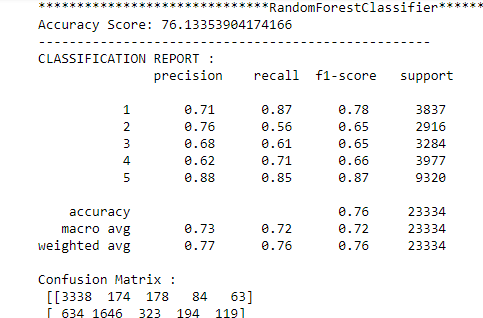
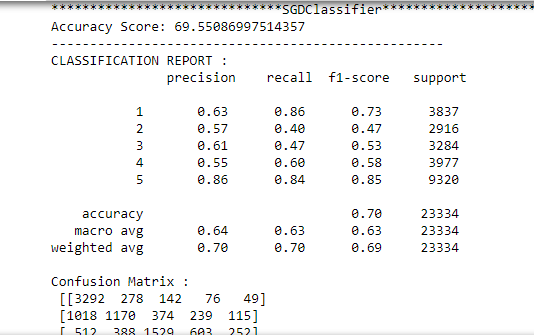
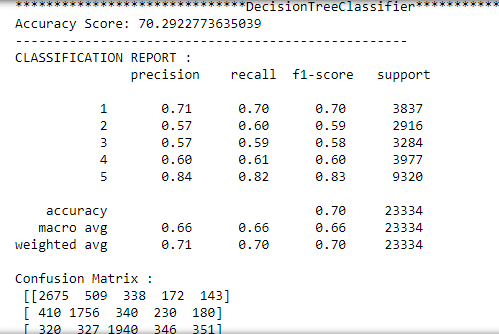
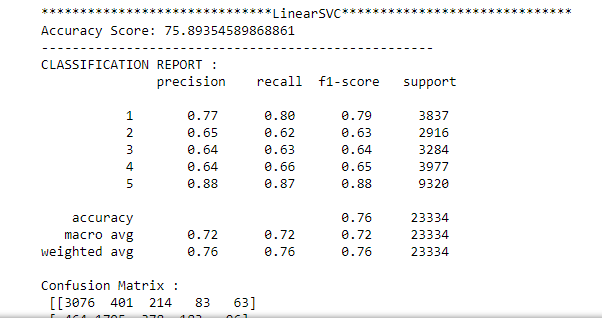
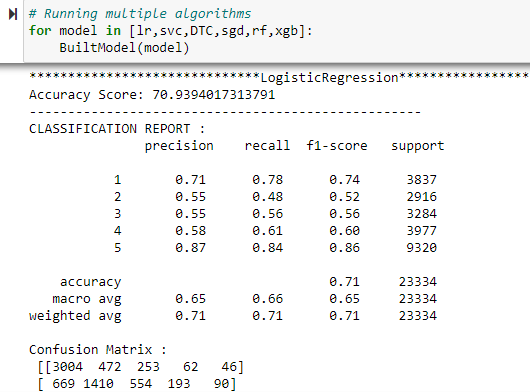
5 Run and Evaluate selected models

1. Model Building:

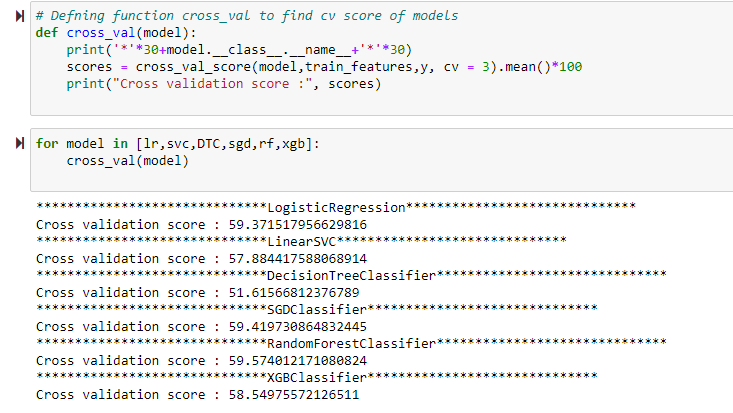
I have used 6 classification algorithms. First, I have created 6 different classification algorithms and are appended in the variable models. Followed by TFIDF vectorization and data balancing. Then, ran a for loop which contained the accuracy of the models along with different evaluation metrics.







1. Cross Validation score



**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.

THANK YOU