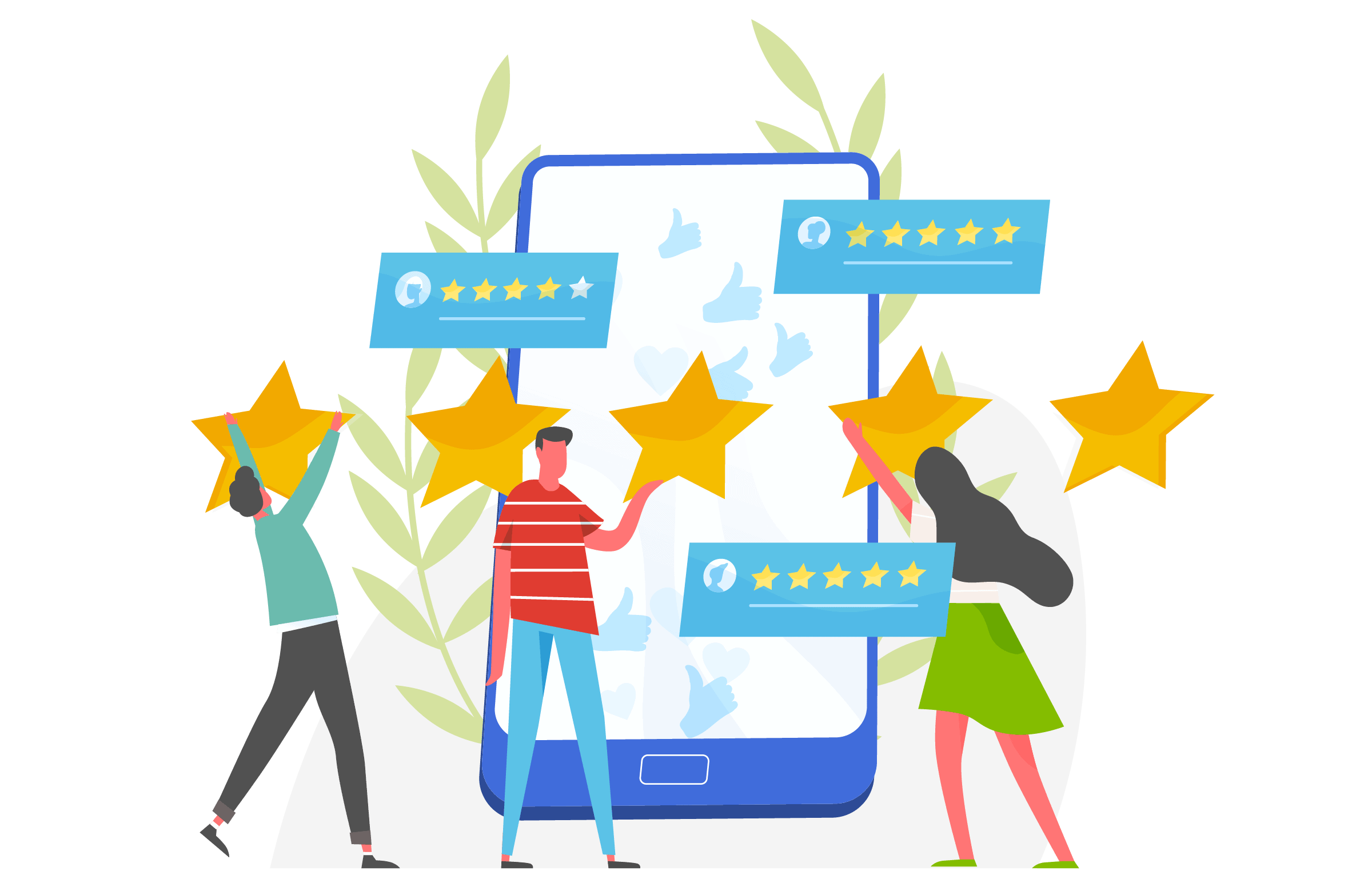


**“A PROJECT REPORT ON**

**REVIEW RATING PREDICTION”**



SUBMITTED BY

HIMAJA IJJADA

**ACKNOWLEDGMENT**

I express my sincere gratitude to FlipRobo Technologies for giving me the opportunity to work on “**A PROJECT REPORT ON REVIEW RATING PREDICTION”** using machine learning algorithms. I would also like to thank FlipRobo Technologies for providing me with the requisite datasets to work with. And I would like to express my gratitude to Mr. Mohd Kashif (SME FlipRobo) and Ms. Sapna Verma (SME FlipRobo) for being of a great help in completion of the project.

Most of the concepts used to predict the ratings of reviews are learned from Data Trained Institute and below documentations.

* https://scikit-learn.org/stable/
* https://seaborn.pydata.org/
* https://www.scipy.org/

**CONTENTS**

* **Introduction**
* Business Problem Framing
* Conceptual Background of the Domain Problem
* Review of Literature
* Motivation for the Problem Undertaken
* **Analytical problem framing**
* Mathematical/ Analytical Modeling of the Problem
* Data Sources and their formats
* Data Preprocessing Done
* Data Inputs- Logic- Output Relationships
* Assumptions
* Hardware and Software Requirements and Tools Used
* **Model/s Development and evaluation**
* Visualizations
* Identification of possible problem-solving approaches (methods)
* Testing of Identified Approaches (Algorithms)
* Run and Evaluate selected models
* Key Metrics for success in solving problem under consideration
* 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**

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.

**Business Problem Framing**

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, we have to build an application which can predict the rating by seeing the review.

**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 analyzing 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.

As far as we know, all the recent works on recommendation techniques utilizing opinions inferred from user’s reviews are either focused on the item recommendation task or use only the opinion information, completely leaving user’s ratings out of consideration. The approach proposed in this report is filling this gap, providing a simple, personalized and scalable rating prediction framework utilizing both ratings provided by users and opinions inferred from their reviews. Experimental results provided on dataset containing user ratings and reviews from the real world Amazon and Flipkart Product Review Data show the effectiveness of the proposed framework.

**Review of Literature**

In real life, people are influenced by peer group 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.

**Motivation for the Problem Undertaken**

The project was 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.

**Analytical problem framing**

**Mathematical/ Analytical Modeling of the Problem**

In this 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 overfitting. 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.

I will need to build multiple classification machine learning models. Before model building will need to perform all data pre-processing steps involving NLP. After trying different classification models with different hyper parameters then will select the best model out of it. Will need to follow the complete life cycle of data science that includes steps like -

1. Data Cleaning

2. Exploratory Data Analysis

3. Data Pre-processing

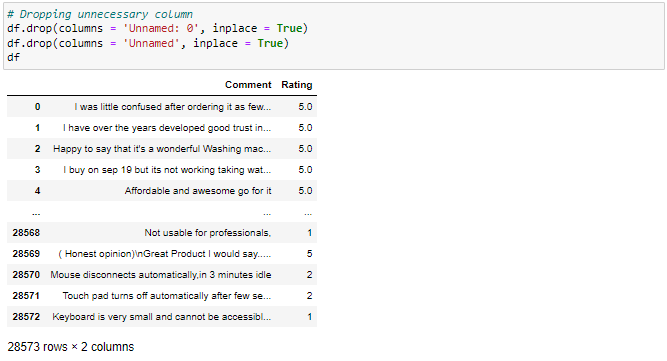
4. Model Building

5. Model Evaluation

6. Selecting the best model

Finally, we compared the results of proposed and baseline features with other machine learning algorithms. Findings of the comparison indicate the significance of the proposed features in review rating prediction.

**Data Sources and their formats**

****

The data set contains nearly 28573 samples with 2 features. Since **Rating** is my target column and it is a categorical column with 5 categories, this problem is a **Multi Classification Problem**. The Ratings can be 1, 2, 3, 4 or 5, which represents the likeliness of the product to the customer. The data set includes:

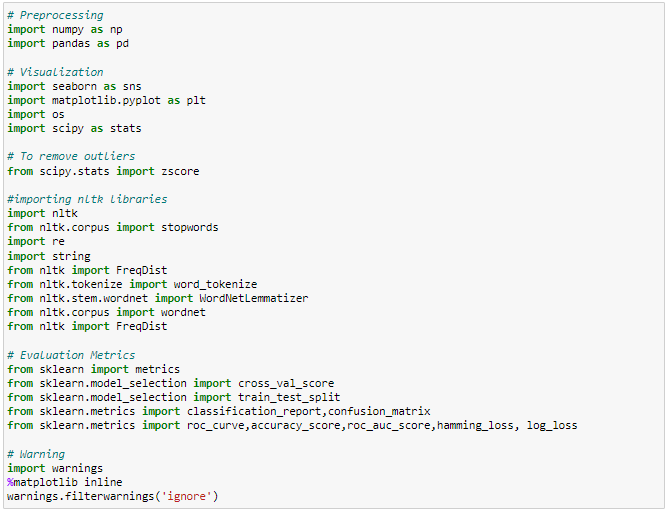
* Comment: Text Content of the Review.
* Rating: Ratings out of 5 stars.

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.

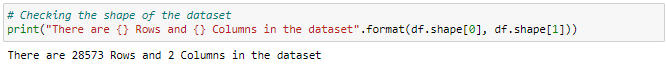
**Data Preprocessing Done**

Importing all necessary libraries and packages

****

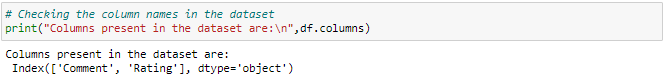
We have imported all the necessary libraries/packages.

Checking the shape of the dataset

****

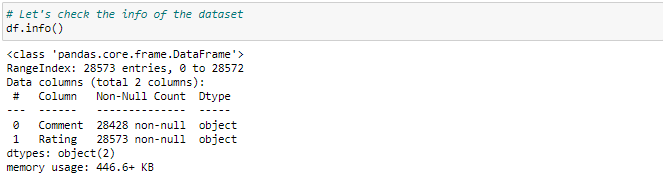
So there are 28573 rows and 2 columns in the dataset.

Checking the column names in the dataset

****

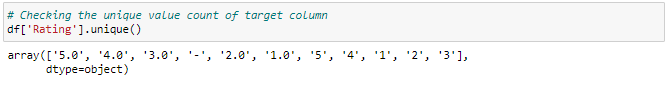
So above 2 are the column names in the dataset.

Checking the info of the dataset

****

By observing the info we can say that there are some null values in the dataset and all the columns are of object data type which means all the entries are string entries.

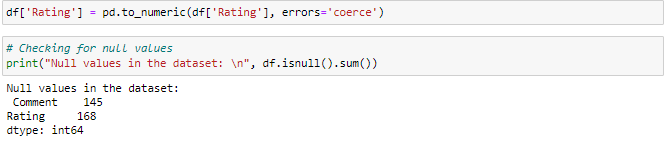
Checking the unique value count of target column

****

Looking the above entries in target column we can observe that

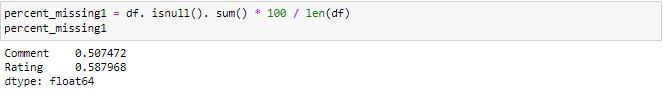
* There are some blank spaces in the column which need to be addressed
* There are string entrie which we shall replace with their respective rating values(stars).

Converting to numeric datatype and checking for null values

****

So we have a minimal amount of nan values in the Comment column of the dataset. Apart from the star ratings there is another column which has nan values. Let's view the percentage of the nan values in the both the Rating and Comment columns.

Checking the percentage of missing / nan values in the columns

****

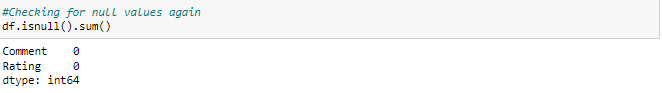
There are approximately 0.5% null values in both the columns. We can use imputation methods to fill these nan values but this could impact the model, so these rows with null values can be dropped from the dataset

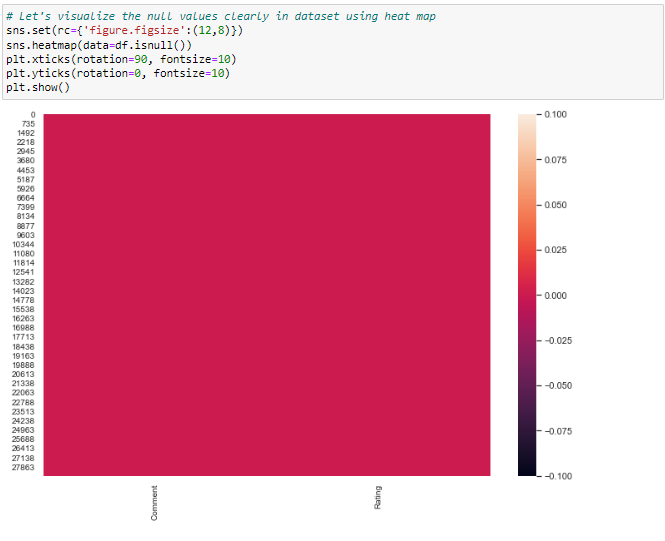
Dropping the rows with null values

****

Now after dropping the null values we have 28260 rows and 2 columns in the dataset

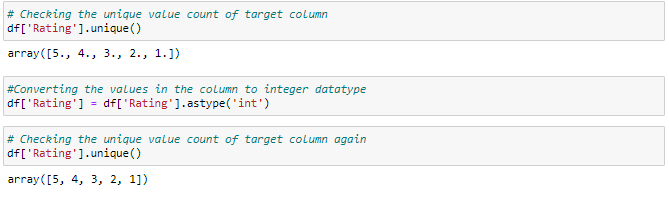
Checking for null values again

****

****

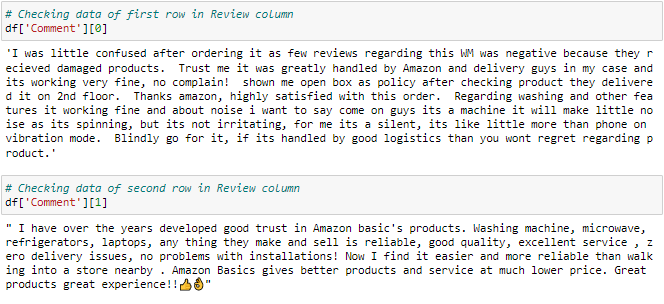
Now we can observe that there are no null values in the dataset

Checking the unique value count of target column and converting the values in the column to integer datatype

****

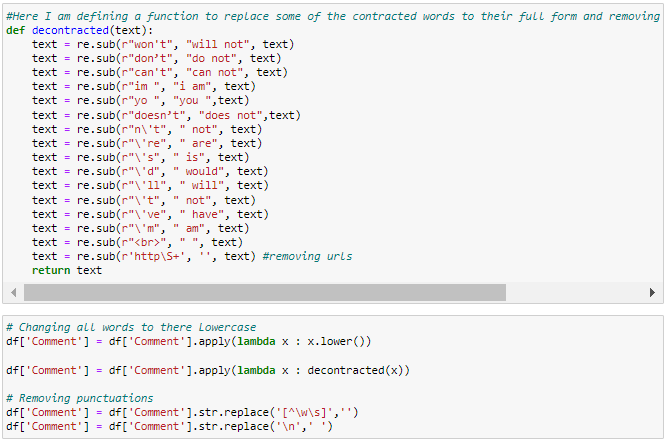
Now we can see that the entries in the rating column are in integers

Let's have a look into our Review column and see first 2 entries how the data looks:

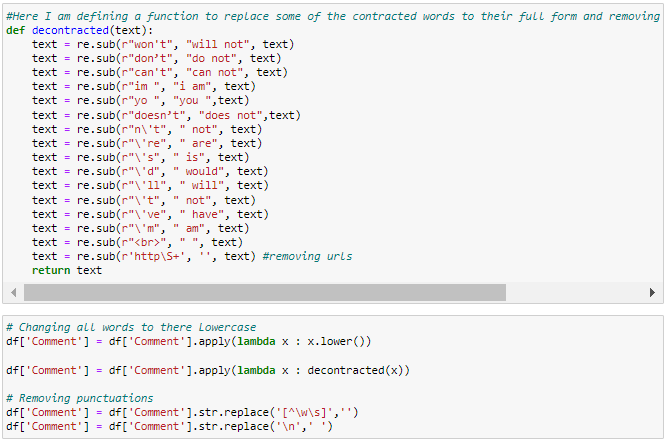
****

# By observing the Reviews we can say that there are many words, numbers, as well as punctuations which are not important for our predictions. So we need to do good text processing.

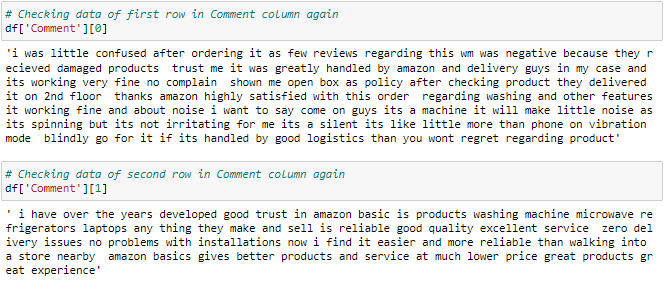
# Text Processing:

****

Changing all words to Lowercase

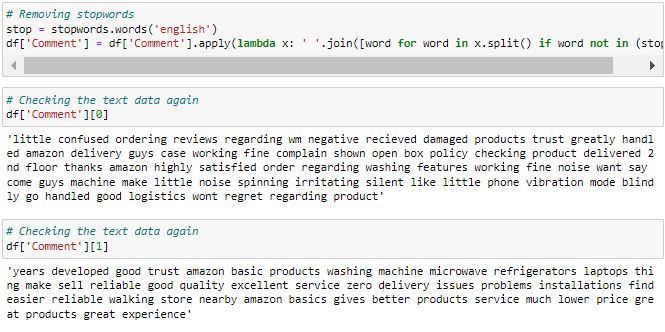
****

### Let's have a look into our text again:

****

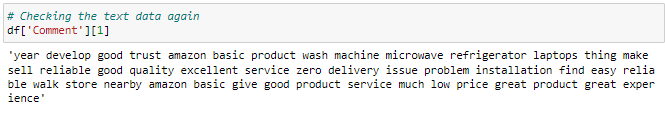
Now the data looks far better than previous. And we have successfully removed punctuations and unwanted text from our text and lowercased all the text data.

# Removing Stop Words:

****

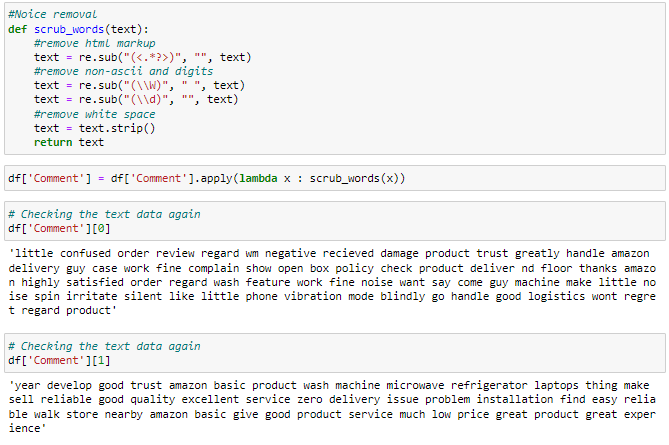
Now we have removed all stop words from the text data.

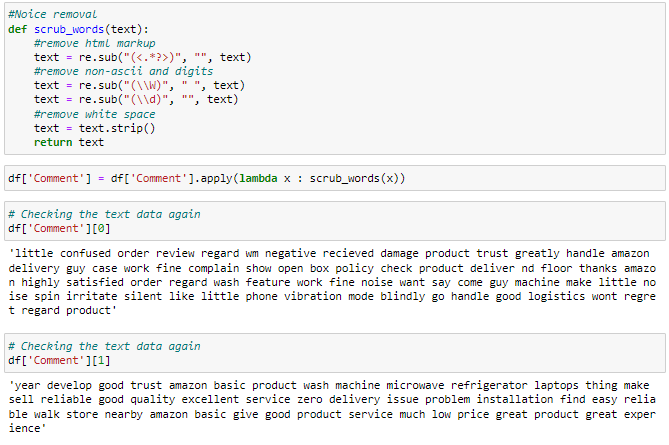
# Lemmatization:

****

So now we have removed the inflectional endings and left out with the base or dictionary form of a word.

# Text Normalization - Standardization:

****

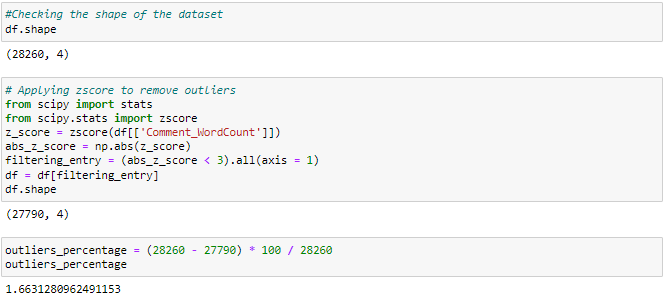
****

Finally I have defined a function scrub\_words for removing the noise from the text. It will remove any html markups, digits and white spaces from the text.

Now we did all the text-processing steps and got required input for our model. We will get into Visualization part now.

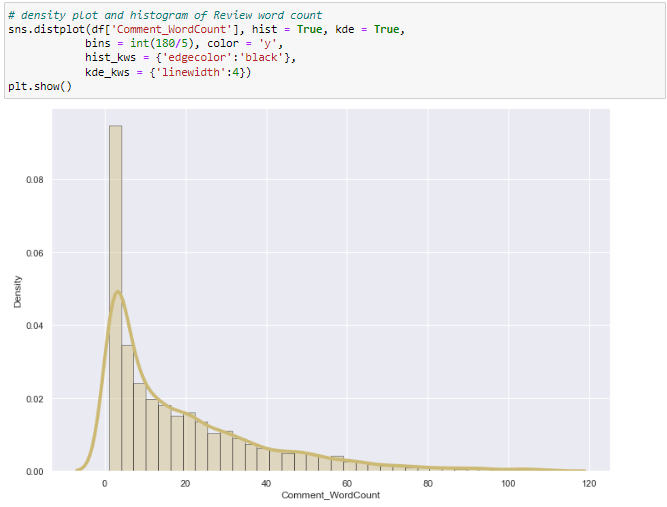
# Removing Outliers:

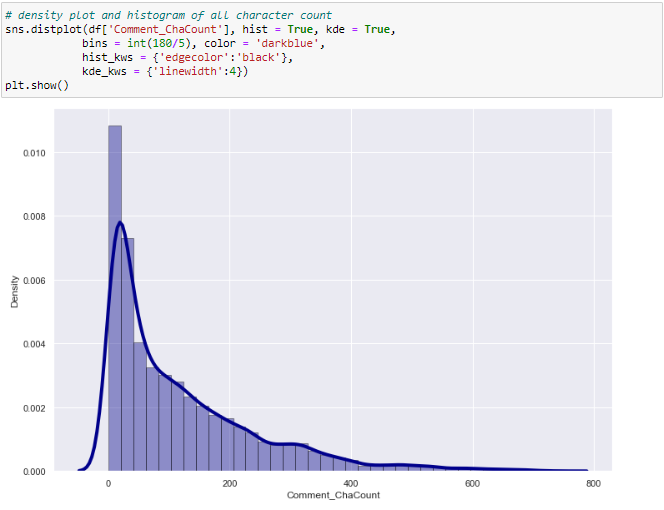
As we know that some of the review are too lengthy, so i have to treat them as outliers and remove them using z\_score method.

****

Great by removing the outliers we are losing 1.66% of data which is very less and it is in acceptable range.

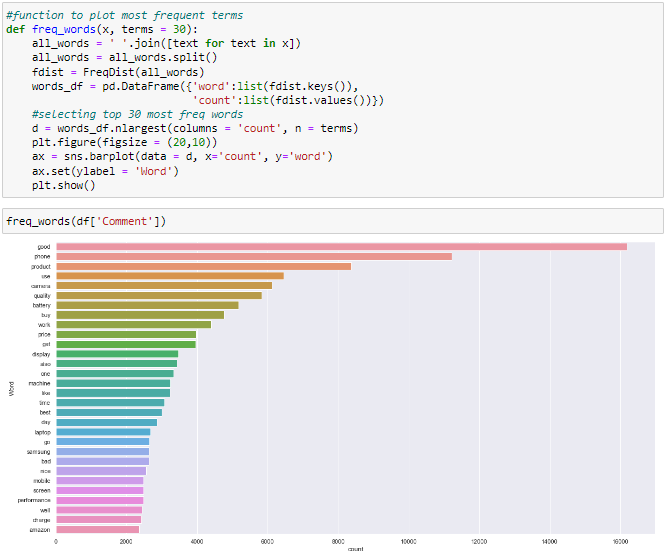
Plotting histograms for word count and character counts again after removing outliers:

****

****

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.

# iii) Top 30 most frequently occuring words:

****

By seeing the above plot we can see that Good, phone, product, use......are occurring frequently.

# iv) Top 30 Rare words:

****

Above list of words are have rare occurrence in Review.

**Data Inputs- Logic- Output Relationships**

# Word cloud:

****

**Rating 1:**

****

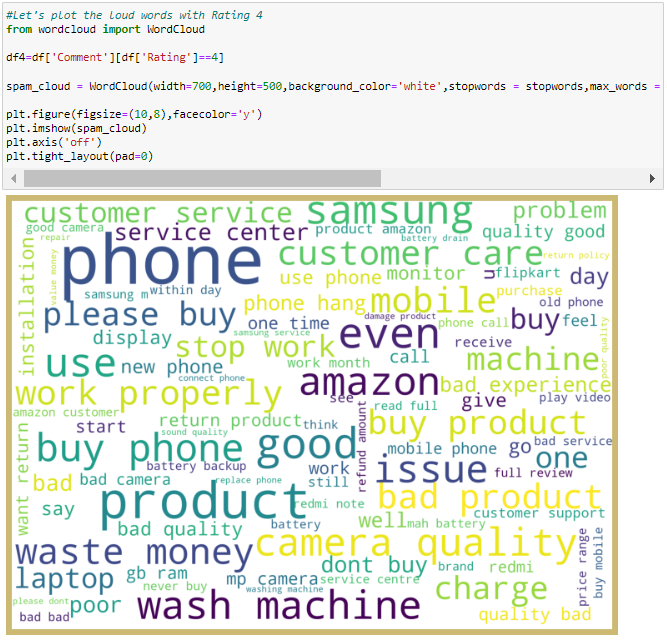
**Rating 2:**

****

**Rating 3:**

****

**Rating 4:**

****

**Rating 5:**

****

**Assumptions**

- 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.

**Hardware and Software Requirements and Tools Used**

To build the machine learning projects it is important to have the following hardware and software requirements and tools.

**Hardware required:**

* Processor: core i5 or above
* RAM: 8 GB or above
* ROM/SSD: 250 GB or above

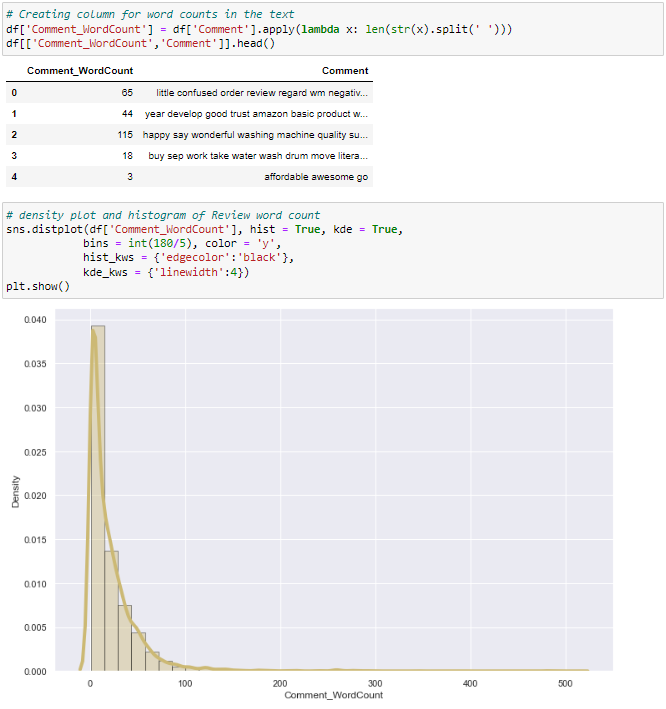
**Software required:**

* Distribution: Anaconda Navigator
* Programming language: Python
* Browser based language shell: Jupyter Notebook
* Word cloud: For visual display of text data
* Libraries/Packages specifically being used - Pandas, NumPy, matplotlib, seaborn, scikit-learn, pandas-profiling, missingno, NLTK

**Model/s Development and evaluation**

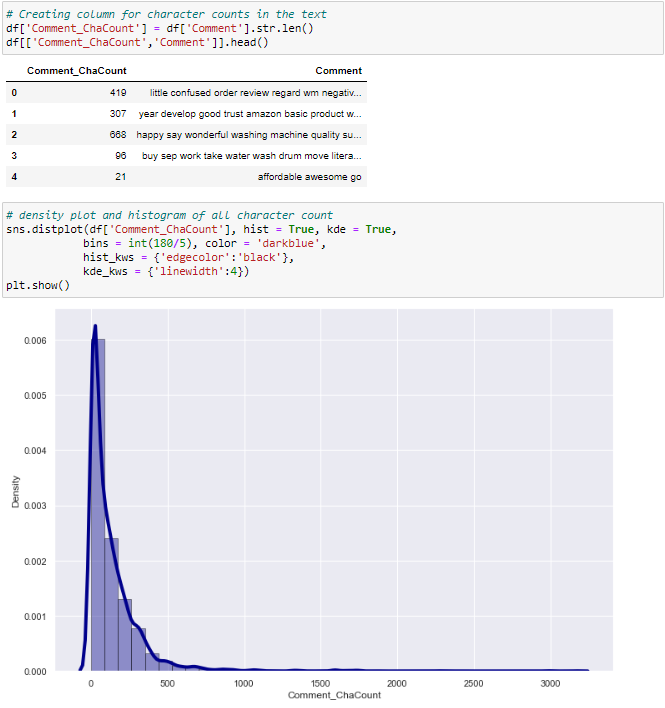
**Visualizations**

## i) Word Counts:

****

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 100 but some of the reviews are too lengthy which may act like outliers in our data.

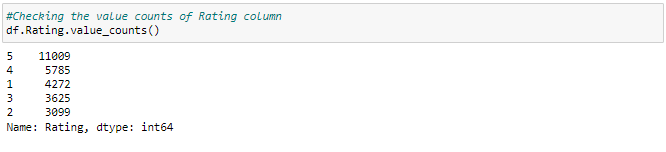
## ii) Character count:

****

Above plot represents histogram for character count of Review text, which is quite similar to the histogram of word count.

**Identification of possible problem-solving approaches (methods)**

Checking the value counts of Rating column



# Converting text data into vectors using Tfidf Vectorizer:



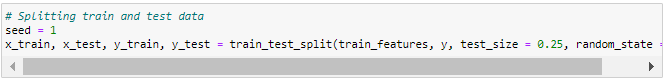
I have converted text into feature vectors using TF-IDF vectorizer and separated our feature and labels. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models. Just making the Reviews more appropriate so that we’ll get 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.

**Testing of Identified Approaches (Algorithms)**

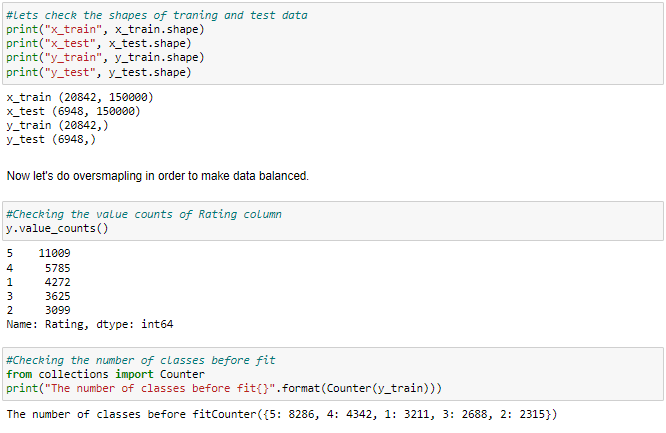
# Model Building and Evaluation:

# 

# Splitting the data into train and test:

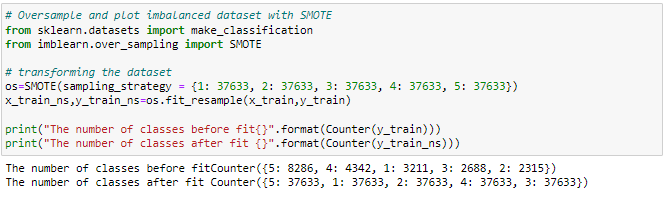


Data Balancing:

****

So we have maximum count 8286 for 5 rating which may hamper the model accuracy. Hence we shall balance the data using SMOTE

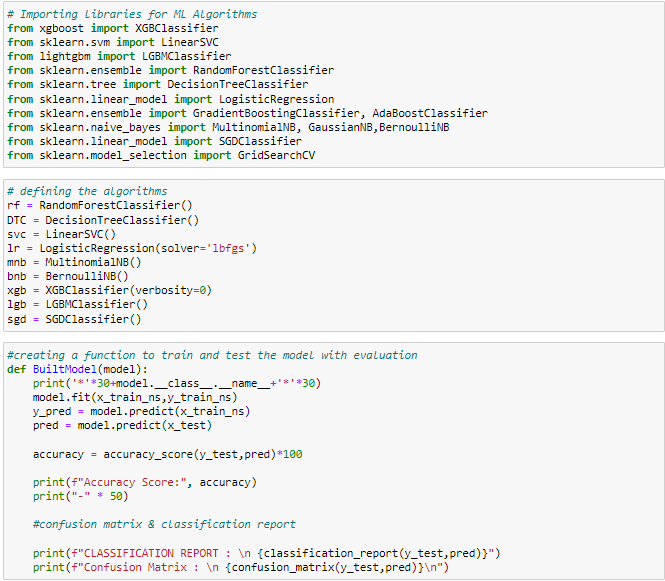
Oversample and plot imbalanced dataset with SMOTE

****

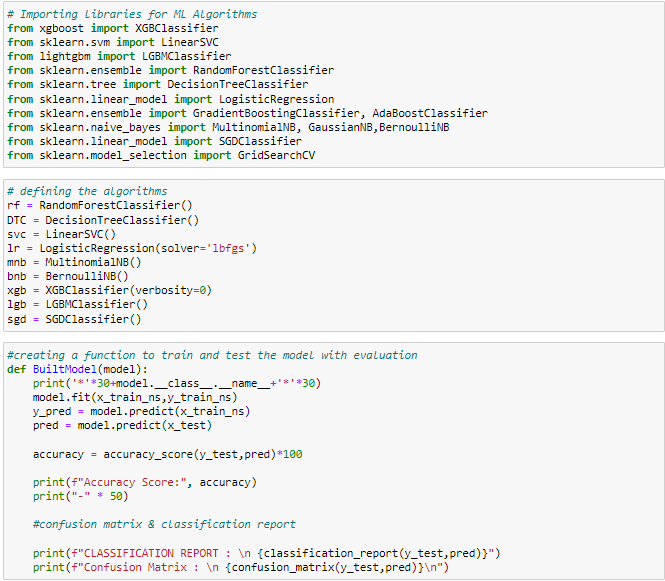
So now we have successfully balanced the data. Let's proceed with model building.

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
* LogisticRegression
* RandomForestClassifier
* DecisionTreeClassifier
* XGBClassifier
* SGDClassifier

****

**Run and Evaluate selected models**

****

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LogisticRegression\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy Score: 82.74323546344272

--------------------------------------------------

CLASSIFICATION REPORT :

precision recall f1-score support

1 0.89 0.91 0.90 1061

2 0.88 0.89 0.88 784

3 0.75 0.80 0.77 937

4 0.73 0.72 0.72 1443

5 0.87 0.84 0.86 2723

accuracy 0.83 6948

macro avg 0.82 0.83 0.83 6948

weighted avg 0.83 0.83 0.83 6948

Confusion Matrix :

[[ 969 28 37 17 10]

[ 36 695 31 15 7]

[ 36 33 749 80 39]

[ 29 23 76 1043 272]

[ 21 14 112 283 2293]]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LinearSVC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy Score: 83.7651122625216

--------------------------------------------------

CLASSIFICATION REPORT :

precision recall f1-score support

1 0.90 0.92 0.91 1061

2 0.90 0.89 0.90 784

3 0.77 0.80 0.79 937

4 0.71 0.77 0.74 1443

5 0.89 0.84 0.86 2723

accuracy 0.84 6948

macro avg 0.84 0.84 0.84 6948

weighted avg 0.84 0.84 0.84 6948

Confusion Matrix :

[[ 972 20 30 23 16]

[ 33 697 29 18 7]

[ 34 28 750 85 40]

[ 20 20 69 1113 221]

[ 18 8 91 318 2288]]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DecisionTreeClassifier\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy Score: 81.72135866436385

--------------------------------------------------

CLASSIFICATION REPORT :

precision recall f1-score support

1 0.86 0.88 0.87 1061

2 0.87 0.90 0.88 784

3 0.77 0.79 0.78 937

4 0.71 0.73 0.72 1443

5 0.86 0.83 0.84 2723

accuracy 0.82 6948

macro avg 0.81 0.82 0.82 6948

weighted avg 0.82 0.82 0.82 6948

Confusion Matrix :

[[ 935 26 25 35 40]

[ 34 702 13 21 14]

[ 41 19 743 82 52]

[ 34 23 78 1049 259]

[ 48 38 107 281 2249]]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SGDClassifier\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy Score: 79.74956822107082

--------------------------------------------------

CLASSIFICATION REPORT :

precision recall f1-score support

1 0.86 0.91 0.88 1061

2 0.86 0.87 0.87 784

3 0.70 0.75 0.72 937

4 0.73 0.62 0.67 1443

5 0.82 0.84 0.83 2723

accuracy 0.80 6948

macro avg 0.79 0.80 0.79 6948

weighted avg 0.80 0.80 0.80 6948

Confusion Matrix :

[[ 967 26 38 14 16]

[ 43 685 29 8 19]

[ 56 40 700 68 73]

[ 33 30 96 899 385]

[ 31 17 144 241 2290]]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RandomForestClassifier\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy Score: 85.03166378814048

--------------------------------------------------

CLASSIFICATION REPORT :

precision recall f1-score support

1 0.88 0.93 0.90 1061

2 0.96 0.89 0.92 784

3 0.83 0.79 0.81 937

4 0.78 0.72 0.75 1443

5 0.85 0.90 0.87 2723

accuracy 0.85 6948

macro avg 0.86 0.84 0.85 6948

weighted avg 0.85 0.85 0.85 6948

Confusion Matrix :

[[ 984 14 21 16 26]

[ 42 696 15 18 13]

[ 44 7 740 69 77]

[ 25 7 52 1041 318]

[ 19 2 68 187 2447]]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*XGBClassifier\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy Score: 80.15256188831317

--------------------------------------------------

CLASSIFICATION REPORT :

precision recall f1-score support

1 0.84 0.92 0.87 1061

2 0.87 0.86 0.86 784

3 0.73 0.75 0.74 937

4 0.70 0.68 0.69 1443

5 0.85 0.82 0.83 2723

accuracy 0.80 6948

macro avg 0.80 0.81 0.80 6948

weighted avg 0.80 0.80 0.80 6948

Confusion Matrix :

[[ 971 34 23 12 21]

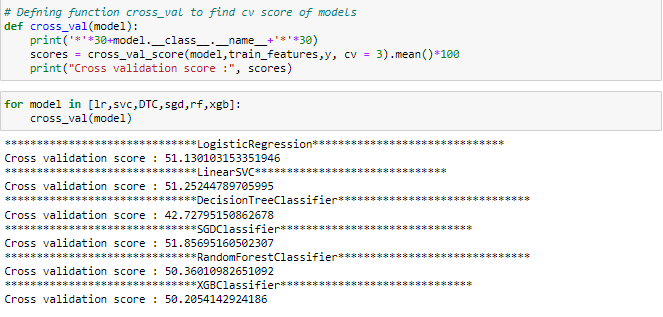
[ 54 673 28 20 9]

[ 56 30 704 91 56]

[ 36 20 86 981 320]

[ 45 18 123 297 2240]]

# Cross validation score:

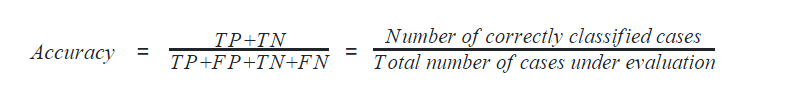


All our algorithms are giving approximately 50% accuracy as cross validation scores due to less number of the features and data imbalance. Among these algorithms I am selecting SGD Classifier as best fitting algorithm for our final model as it is giving least difference between accuracy and cv score.

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

1. **Accuracy**

Accuracy can also be defined as the ratio of the number of correctly classified cases to the total of cases under evaluation. The best value of accuracy is 1 and the worst value is 0.



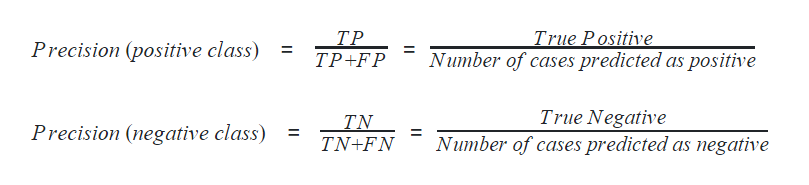
In python, the following code calculates the accuracy of the machine learning model.

Accuracy = metrics.accuracy\_score(y\_test, preds)

Accuracy

1. **Precision**

Precision can be defined with respect to either of the classes. The precision of negative class is intuitively the ability of the classifier not to label as positive a sample that is negative. The precision of positive class is intuitively the ability of the classifier not to label as negative a sample that is positive. The best value of precision is 1 and the worst value is 0.

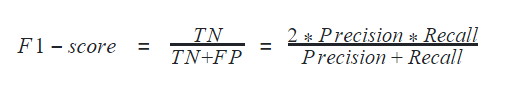


1. **Recall**

Recall can also be defined with respect to either of the classes. Recall of positive class is also termed sensitivity and is defined as the ratio of the True Positive to the number of actual positive cases. It can intuitively be expressed as the ability of the classifier to capture all the positive cases. It is also called the True Positive Rate (TPR).

1. **F1-score**

F1-score is considered one of the best metrics for classification models regardless of class imbalance. F1-score is the weighted average of recall and precision of the respective class. Its best value is 1 and the worst value is 0.



In python, F1-score can be determined for a classification model using

f1\_positive = metrics.f1\_score(y\_test, preds, pos\_label=1)

f1\_negative = metrics.f1\_score(y\_test, preds, pos\_label=0)

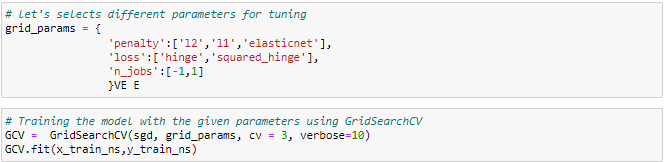
f1\_positive, f1\_negative

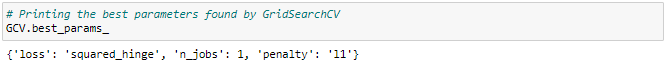
Accuracy, Precision, Recall, and F1-score can altogether be calculated using the method classification\_report in python

**Interpretation of the Results**

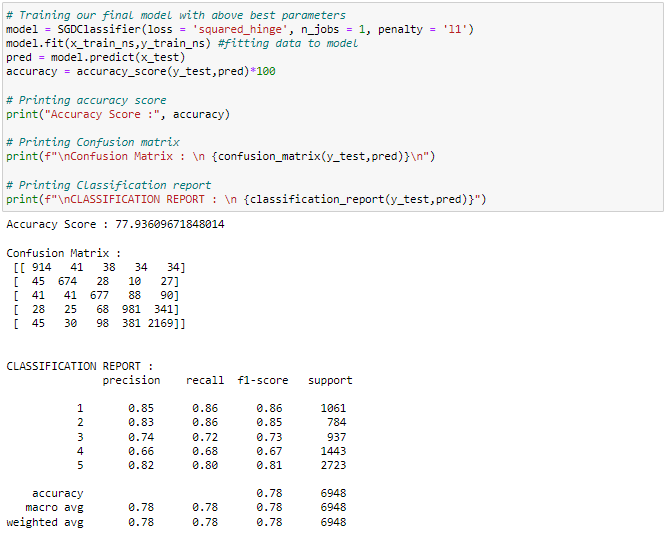
From all of these above models SGDClassifier is giving me better performance with less difference between the accuracy score and cv score.

# HyperParameter Tuning:



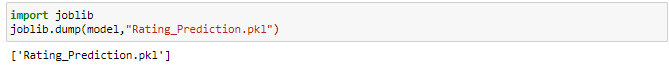


# Final Model:



After hyperparameter tuning we are unable to improve our model accuracy due to less number of features and the data imbalance.

# Model Saving:



Finally I have saved the model into .pkl file.

**Conclusion**

**Key Findings and Conclusions of the Study**

* 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 analyzed 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 SGD Classifier for our final model.
* Finally by doing hyperparameter tuning we got optimum parameters for our final model.

**Learning Outcomes of the Study in respect of Data Science**

I have scrapped the reviews and ratings of different technical products from flipkart.com and amazon.in websites. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analyzed. New analytical techniques (NLP) of machine learning can be used in property research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values, punctuations, URLs, email address and stop words. This study is an exploratory attempt to use 6 machine learning algorithms in estimating Rating, and then compare their results.

To conclude, the application of NLP in rating classification is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to crediting institutes, and presenting an alternative approach to the valuation of Ratings.

In this project we were able to learn various Natural language processing techniques like lemmatization, stemming, removal of Stop words. This project has demonstrated the importance of sampling effectively, modelling and predicting data. Through different powerful tools of visualization we were able to analyses and interpret different hidden insights about the data. The few challenges while working on this project are:-

* Imbalanced dataset
* Lots of text data

**Limitations of this work and Scope for Future Work**

As we know the content of text in reviews is totally depends on the reviewer and they may rate differently which is totally depends on that particular person. So it is difficult to predict ratings based on the reviews with higher accuracies. Still we can improve our accuracy by fetching more data and by doing extensive hyperparameter tuning.

While we couldn’t reach out goal of maximum accuracy in Ratings prediction project, we did end up creating a system that can with some improvement and deep learning algorithms get very close to that goal. As with any project there is room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project.