FLIP ROBO

NAME OF THE PROJECT

Ratings Prediction Project

Submitted by:

Chandan Kumar

FLIPROBO SME:

Ms. Khushboo Garg

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and otherresources that helped me and guided me in completion project.

INTRODUCTION

• Business Problem Framing

Internet is the best source nowadays for any organisation to know public opinions about their products and services. Many consumers form an opinion about a product just by reading a few reviews. Online product reviews provided by consumers who previously purchased products have become a major information source for consumers and marketers regarding product quality. Research has shown that consumer online product ratings reflect both the customers' experience with the product and the influence of others' ratings. Websites prominently display consumers' product ratings, which influence consumers' buying decisions and willingness to pay.

The opinion information is very useful for users and customers alike, many of whom typically read product or service reviews before buying them. Businesses can also use the opinion information to design better strategies for production and marketing.

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.

• Conceptual Background of the Domain Problem

Consumers want to find useful information as quickly as possible. However, searching and comparing text reviews can be frustrating for users as they feel submerged with information. Indeed, the massive amount of text reviews as well as its unstructured text format prevent the user from choosing a product with ease. The star-rating, i.e., stars from 1 to 5 on online platform, rather than its text content gives a quick overview of the product quality

Generally, the ratings and the price of the product are simple heuristics used by the customers to decide over the final purchase of the product. But often, the overall star ratings of the product reviews may not capture the exact polarity of the sentiments.

• Review of Literature

product reviews and ratings represent an important source of information for consumers and are helpful tools in order to support their buying decisions. They also found out that consumers are willing to compare both positive and negative reviews when searching for a specific product. They also found out that consumers are willing to compare both positive and negative reviews when searching for a specific product. the goal of their project is to predict the rating of a customer who reviews the same product more than once, while our project is to predict the user’s first-time rating score. Also, they use stochastic gradient descent to minimize the mean square error to capture the weights of such features.

Motivation for the Problem Undertaken

Data needed for this project is require to scrap from E-commerce platform and data cleaning operation over it. Features derived from textual reviews are used to predict its corresponding star ratings. To accomplish it, the prediction problem is transformed into a multi-class classification task to classify reviews to one of the five classes corresponding to its star rating. 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.

Analytical Problem Framing

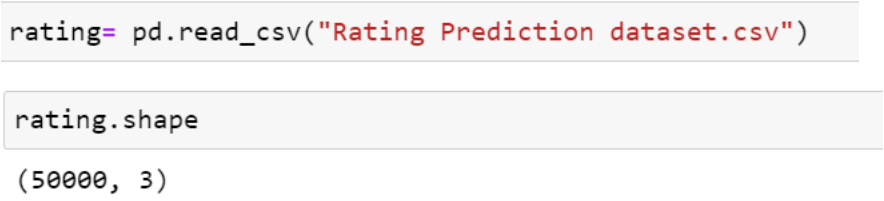
Mathematical/ Analytical Modelling of the Problem

In order to apply text classification, the unstructured format of text has to be converted into a structured format for the simple reason that it is much easier for computer to deal with numbers than text. This is mainly achieved by projecting the textual contents into Vector Space Model, where text data is converted into vectors of numbers.

They are examined without regard to grammar neither to the word order. In such a model, the termfrequency (occurrence of each word) is used as a feature in order to train the classifier. However, using the term frequency implies that all terms are considered equally important.

Data Sources and their formats

Data is collected from Amazon.in and flipkart.com using selenium and saved in CSV file. Around 50000 Reviews are collected for this project.

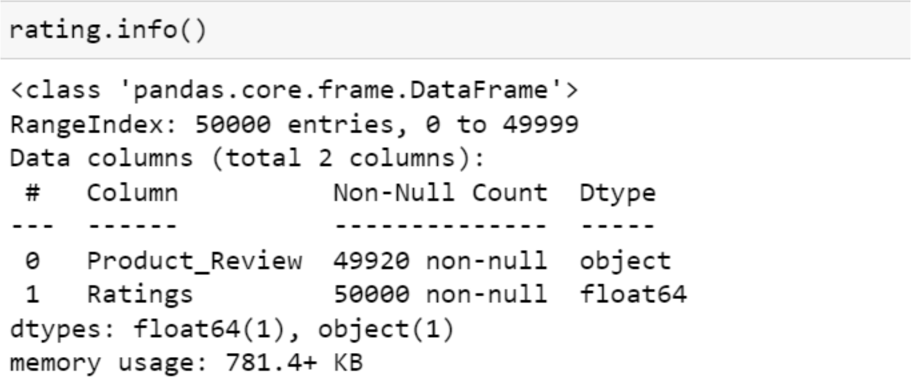


```
rating= pd.read_csv("Rating Prediction dataset.csv")
```

```
rating.shape
```

```
(50000, 3)
```

This is multi-classification problem and Rating is our target feature class to be predicated in this project. There are five different categories in feature target i.e., The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars.



```
rating.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Product_Review  49920 non-null  object 
 1   Ratings         50000 non-null  float64
dtypes: float64(1), object(1)
memory usage: 781.4+ KB
```

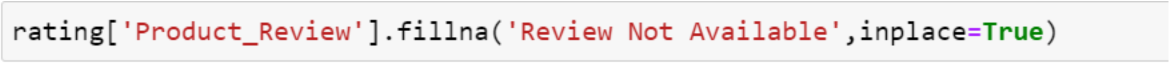
There are some missing values in product review. The datatype of Product review is object while datatypes of Ratings is float.

• Data Pre-processing Done

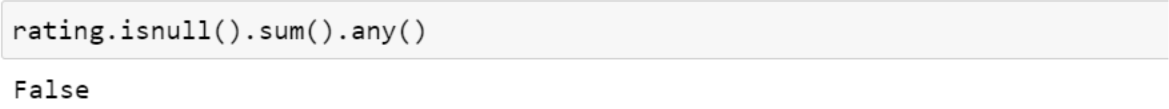
The dataset is large and it may contain some data error. In order to reach clean, error free data some data cleaning & data pre-processing performed data.

• Missing Value Imputation:

Missing value in product reviews are replace with 'Review Not Available'.



```
rating['Product_Review'].fillna('Review Not Available',inplace=True)
```



```
rating.isnull().sum().any()
```

```
False
```

• Data is pre-processed using the following techniques:

1. Convert the text to lowercase

2. Remove the punctuations, digits and special characters

3. Tokenize the text, filter out the adjectives used in the review and create a new column in data frame

4. Remove the stop words

5. Stemming and Lemmatising

6. Applying Text Vectorization to convert text into numeric

• 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 variable's changes. Using word cloud, we can see most occurring word for different categories.

• Hardware and Software Requirements and Tools Used

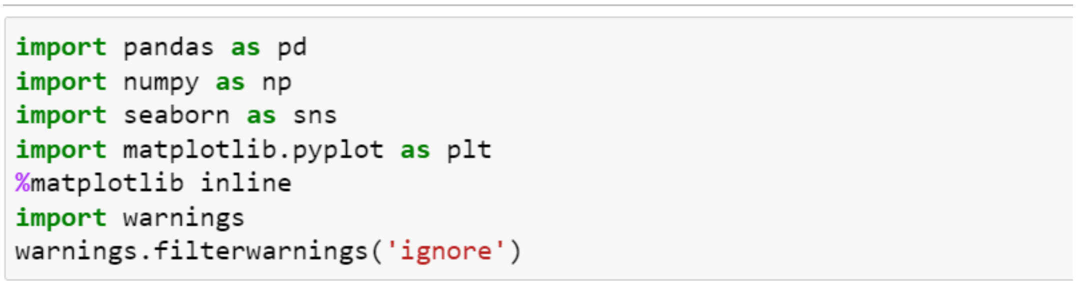
Hardware Used –

1. Processor — Intel i5 processor with 2.4GHZ
2. RAM — 8 GB
3. GPU — 2GB AMD Radeon Graphics card

Software utilized –

1. Anaconda – Jupyter Notebook
2. Selenium – Web scraping
3. Google Colab – for Hyper parameter tuning

Libraries Used – General library for data wrangling & visualization

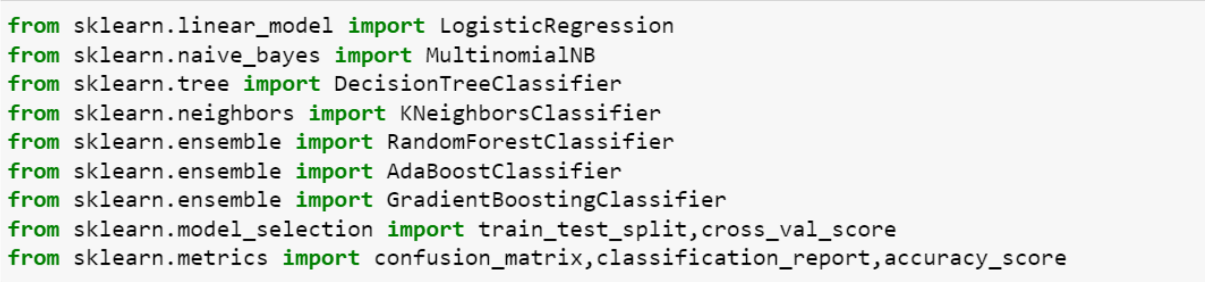


```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Libraries used for Text Mining / Text Analytics are:

```
import re
import string
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from wordcloud import WordCloud
```

Libraries used for machine learning model building



```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
```

Model/s Development and Evaluation

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

First part of problem solving is to scrap data from amazon.in and flipkart.com website which we already done. Second is performing text mining operation to convert textual review in ML algorithm useable form. Third part of problem building machine learning model to predict rating on review. This problem can be solve using classification-based machine learning algorithm like logistics regression. Further Hyper parameter tuning performed to build more accurate model out of best model.

- ### Testing of Identified Approaches (Algorithms)

The different classification algorithm used in this project to build ML model are as below:

❖ Random Forest classifier

❖ Decision Tree classifier

❖ Logistics Regression

❖ AdaBoost Classifier

❖ Gradient Boosting Classifier

. KEY METRICS FOR SUCCESS IN SOLVING PROBLEMUNDER CONSIDERATION

▪ 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.

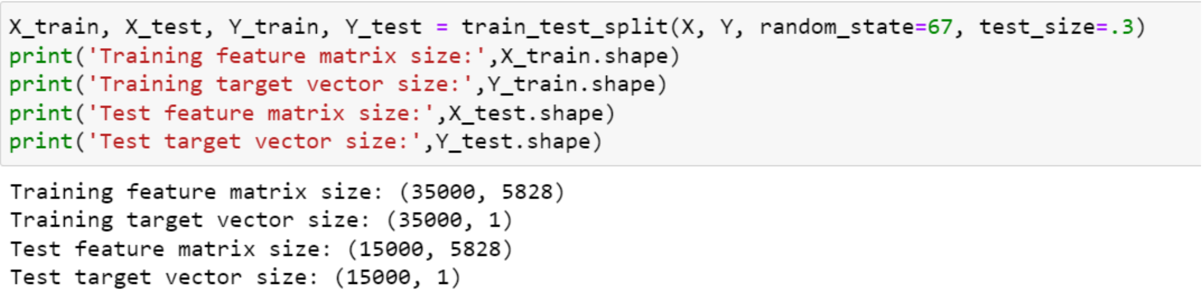
▪ Cross validation Score: To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of cross-validation for diagnostic purposes. Make a scorer from a performance metric or loss function.

▪ AUC_ROC _score: ROC curve. It is a plot of the false positive rate (xaxis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0

▪ We have used Accuracy Score and Cross validation score as key parameter for model evaluation in this project since balancing of data is perform.

Run and Evaluate Selected Models

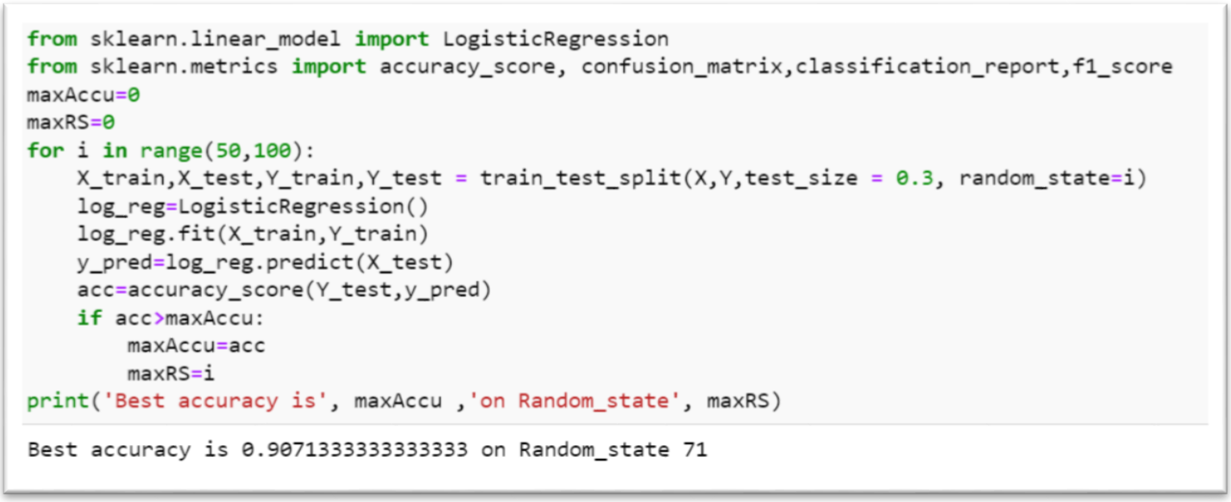
1. Linear Regression:



```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state=67, test_size=.3)
print('Training feature matrix size:',X_train.shape)
print('Training target vector size:',Y_train.shape)
print('Test feature matrix size:',X_test.shape)
print('Test target vector size:',Y_test.shape)
```

```
Training feature matrix size: (35000, 5828)
Training target vector size: (35000, 1)
Test feature matrix size: (15000, 5828)
Test target vector size: (15000, 1)
```

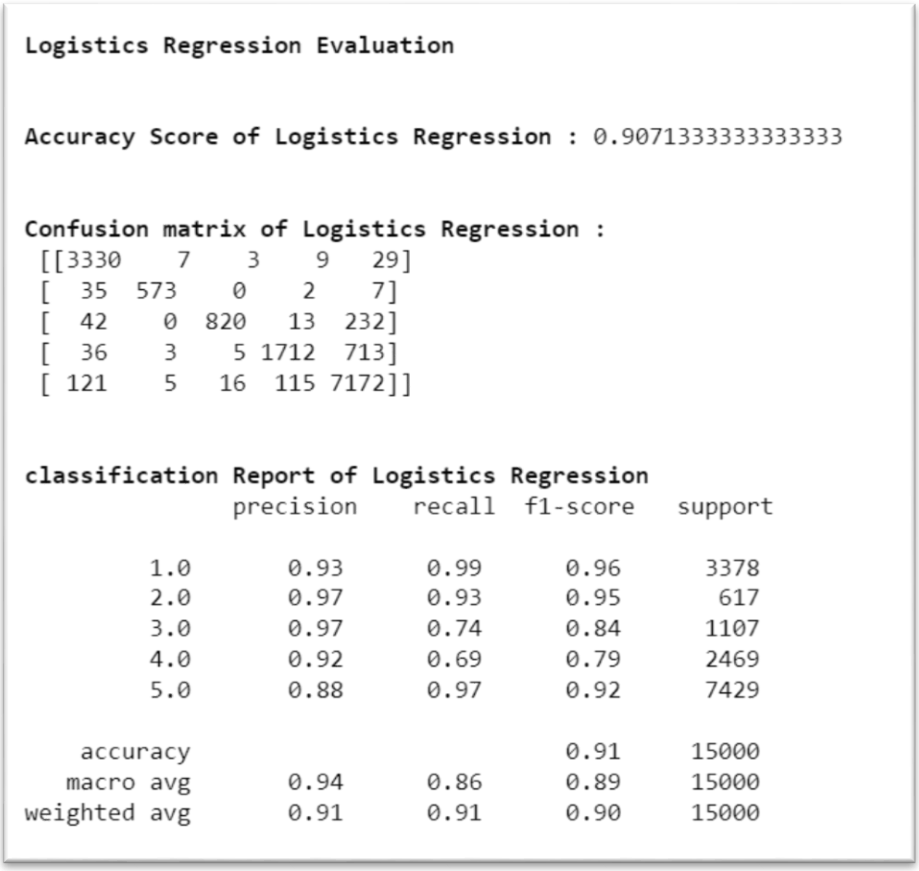
Train-test split is used to split data into training data & testing data. Further best random state is investigated through loop



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report,f1_score
maxAccu=0
maxRS=0
for i in range(50,100):
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.3, random_state=i)
    log_reg=LogisticRegression()
    log_reg.fit(X_train,Y_train)
    y_pred=log_reg.predict(X_test)
    acc=accuracy_score(Y_test,y_pred)
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print('Best accuracy is', maxAccu ,'on Random_state', maxRS)
```

```
Best accuracy is 0.9071333333333333 on Random_state 71
```

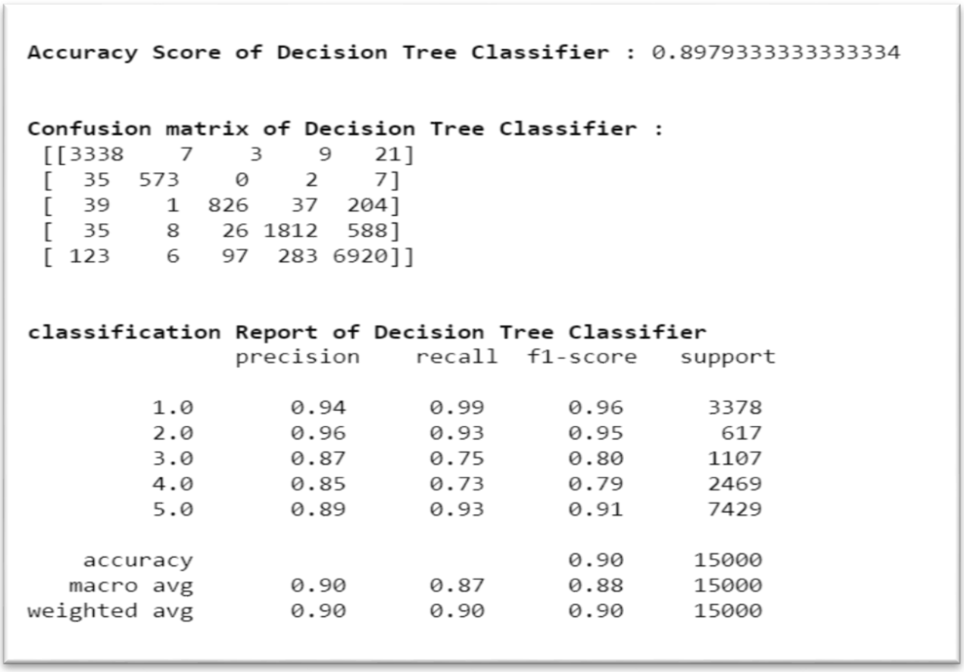
Logistics regression evaluation matrix is shown below:



```
Logistics Regression Evaluation


Accuracy Score of Logistics Regression : 0.9071333333333333


Confusion matrix of Logistics Regression :
 [[3330    7    3    9   29]
 [  35  573    0    2    7]
 [  42    0  820   13  232]
 [  36    3    5 1712  713]
 [ 121    5   16  115 7172]]


classification Report of Logistics Regression
               precision    recall  f1-score   support

         1.0       0.93      0.99      0.96      3378
         2.0       0.97      0.93      0.95       617
         3.0       0.97      0.74      0.84      1107
         4.0       0.92      0.69      0.79      2469
         5.0       0.88      0.97      0.92      7429

    accuracy                           0.91     15000
   macro avg       0.94      0.86      0.89     15000
weighted avg       0.91      0.91      0.90     15000
```

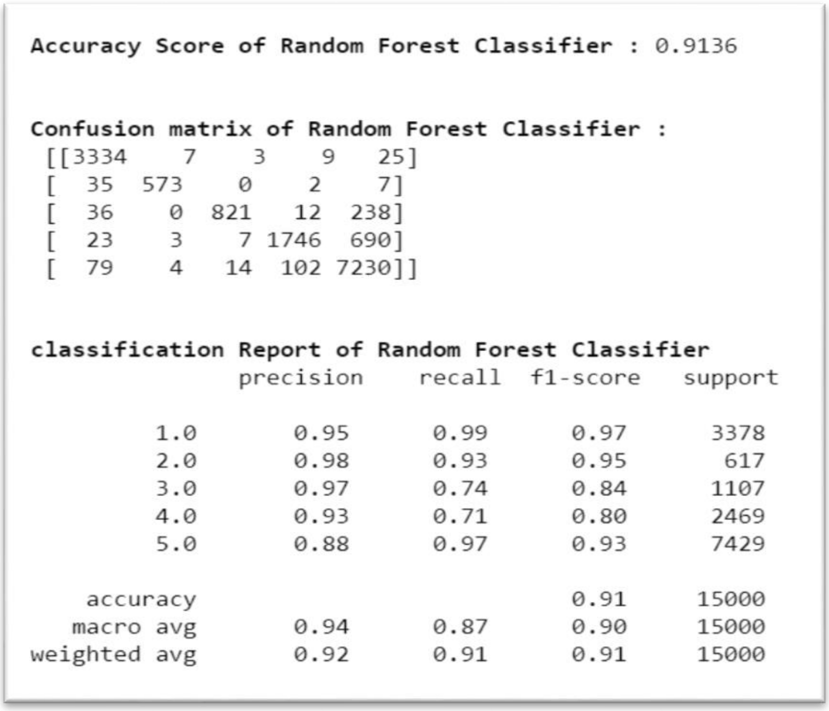
2. Decision Tree Classifier

Decision Tree Classifier model is built and evaluation matrix is shown as below:



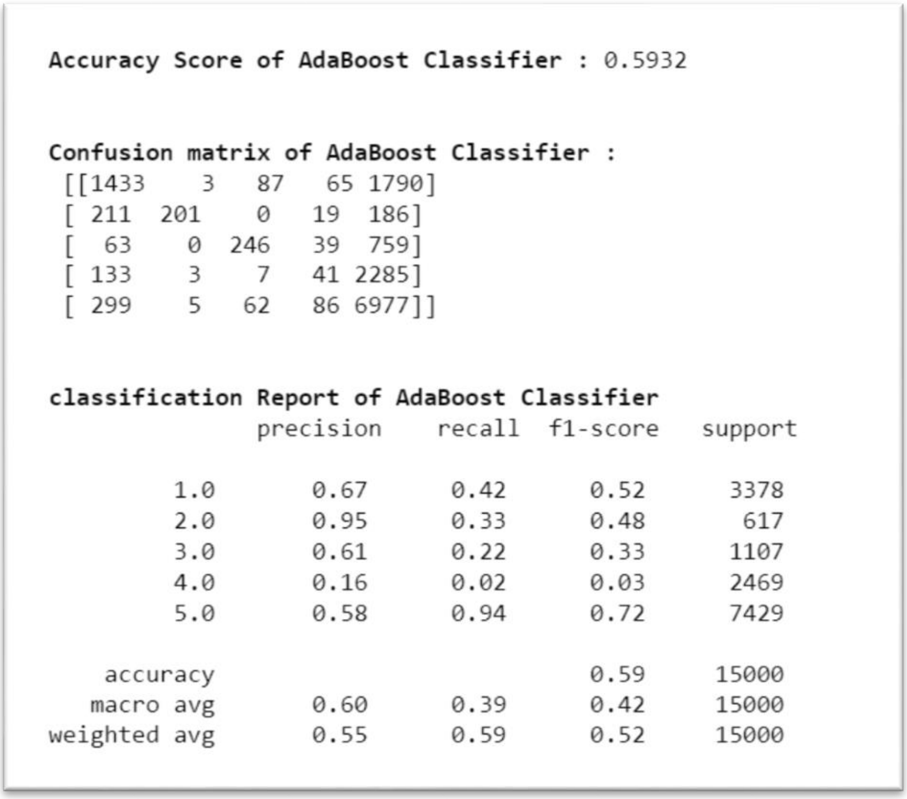
```
Accuracy Score of Decision Tree Classifier : 0.8979333333333334


Confusion matrix of Decision Tree Classifier :
 [[3338    7    3    9   21]
 [  35  573    0    2    7]
 [  39    1  826   37  204]
 [  35    8   26 1812  588]
 [ 123    6   97  283 6920]]


classification Report of Decision Tree Classifier
               precision    recall  f1-score   support

         1.0       0.94      0.99      0.96      3378
         2.0       0.96      0.93      0.95       617
         3.0       0.87      0.75      0.80      1107
         4.0       0.85      0.73      0.79      2469
         5.0       0.89      0.93      0.91      7429

    accuracy                           0.90     15000
   macro avg       0.90      0.87      0.88     15000
weighted avg       0.90      0.90      0.90     15000
```

3. Random Forest Classifier



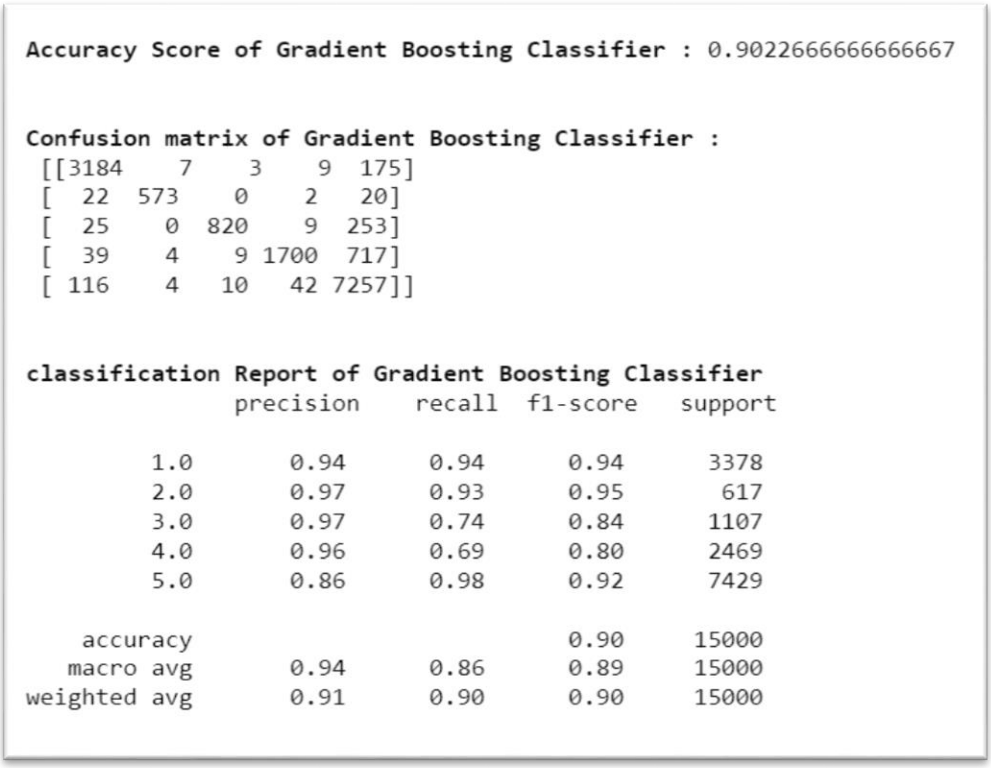
```
Accuracy Score of Random Forest Classifier : 0.9136


Confusion matrix of Random Forest Classifier :
 [[3334    7    3    9   25]
 [  35  573    0    2    7]
 [  36    0  821   12  238]
 [  23    3    7 1746  690]
 [  79    4   14  102 7230]]


classification Report of Random Forest Classifier
               precision    recall  f1-score   support

         1.0       0.95      0.99      0.97      3378
         2.0       0.98      0.93      0.95       617
         3.0       0.97      0.74      0.84      1107
         4.0       0.93      0.71      0.80      2469
         5.0       0.88      0.97      0.93      7429

    accuracy                           0.91     15000
   macro avg       0.94      0.87      0.90     15000
weighted avg       0.92      0.91      0.91     15000
```

4. Ada Boost Classifier



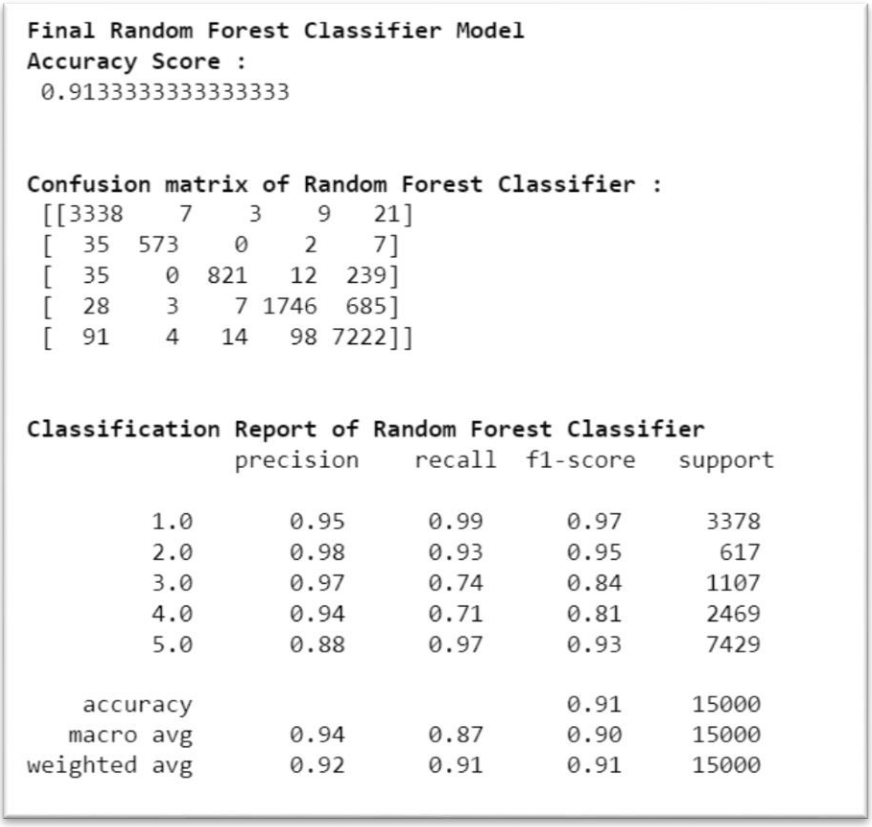
```
Accuracy Score of AdaBoost Classifier : 0.5932


Confusion matrix of AdaBoost Classifier :
 [[1433    3   87   65 1790]
 [ 211  201    0   19  186]
 [  63    0  246   39  759]
 [ 133    3    7   41 2285]
 [ 299    5   62   86 6977]]


classification Report of AdaBoost Classifier
               precision    recall  f1-score   support

         1.0       0.67      0.42      0.52      3378
         2.0       0.95      0.33      0.48       617
         3.0       0.61      0.22      0.33      1107
         4.0       0.16      0.02      0.03      2469
         5.0       0.58      0.94      0.72      7429

    accuracy                           0.59     15000
   macro avg       0.60      0.39      0.42     15000
weighted avg       0.55      0.59      0.52     15000
```

5. Gradient Boosting Classifier



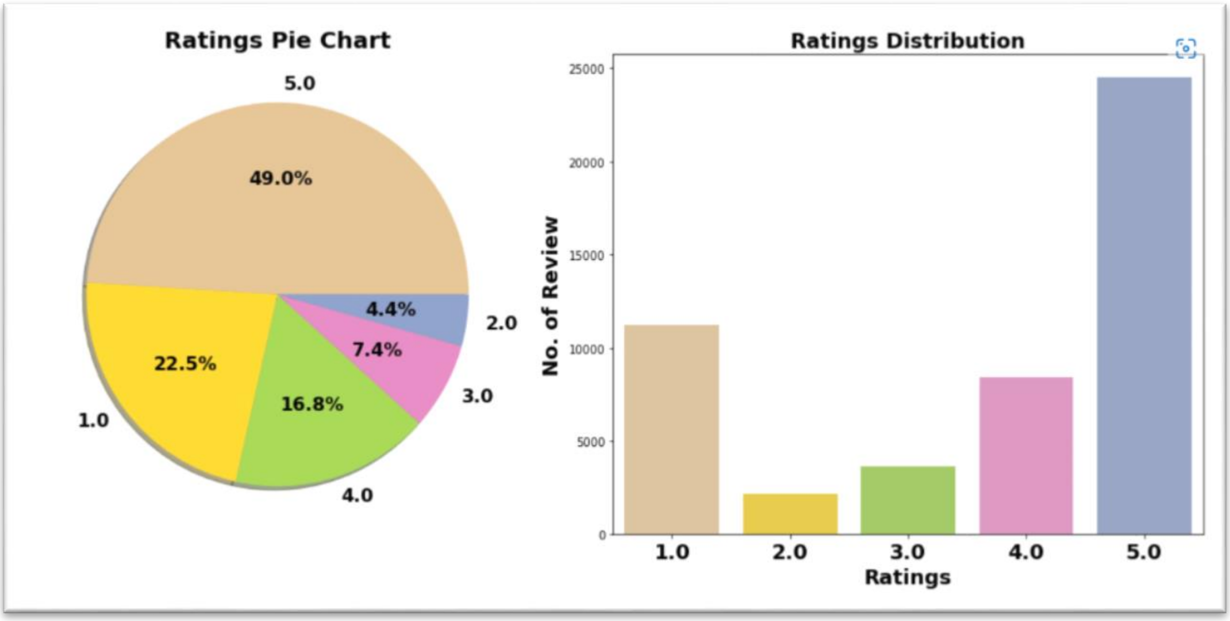
```
Accuracy Score of Gradient Boosting Classifier : 0.9022666666666667


Confusion matrix of Gradient Boosting Classifier :
 [[3184    7    3    9  175]
 [  22  573    0    2   20]
 [  25    0  820    9  253]
 [  39    4    9 1700  717]
 [ 116    4   10   42 7257]]


classification Report of Gradient Boosting Classifier
              precision    recall  f1-score   support

         1.0       0.94      0.94      0.94      3378
         2.0       0.97      0.93      0.95       617
         3.0       0.97      0.74      0.84      1107
         4.0       0.96      0.69      0.80      2469
         5.0       0.86      0.98      0.92      7429

    accuracy                           0.90     15000
   macro avg       0.94      0.86      0.89     15000
weighted avg       0.91      0.90      0.90     15000
```

Final model is built using best parameter in hyper parameters tuning. The corresponding evaluation matrix shown below:



```
Final Random Forest Classifier Model
Accuracy Score :
 0.9133333333333333


Confusion matrix of Random Forest Classifier :
 [[3338    7    3    9   21]
 [  35  573    0    2    7]
 [  35    0  821   12  239]
 [  28    3    7 1746  685]
 [  91    4   14   98 7222]]


Classification Report of Random Forest Classifier
              precision    recall  f1-score   support

         1.0       0.95      0.99      0.97      3378
         2.0       0.98      0.93      0.95       617
         3.0       0.97      0.74      0.84      1107
         4.0       0.94      0.71      0.81      2469
         5.0       0.88      0.97      0.93      7429

    accuracy                           0.91     15000
   macro avg       0.94      0.87      0.90     15000
weighted avg       0.92      0.91      0.91     15000
```

5.Visualizations

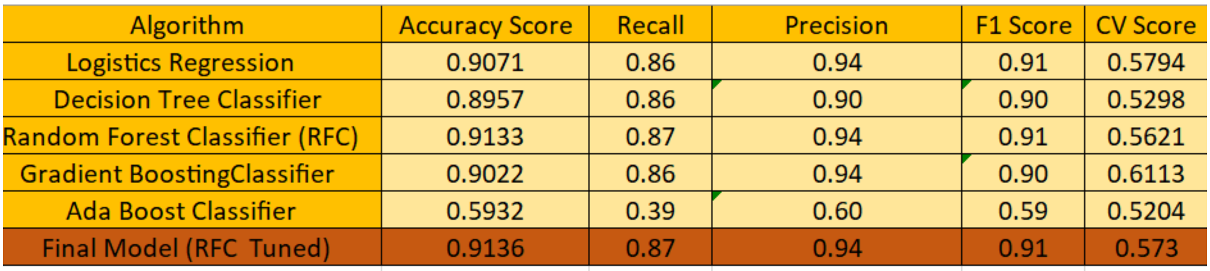


Comment:

1. Around 49% customer given 5- star rating followed by 22.5% customer given lowest 1-star rating.
2. Average Rating is 3.65.

Conclusion

1. Key Findings and Conclusions of the Study



Algorithm	Accuracy Score	Recall	Precision	F1 Score	CV Score
Logistics Regression	0.9071	0.86	0.94	0.91	0.5794
Decision Tree Classifier	0.8957	0.86	0.90	0.90	0.5298
Random Forest Classifier (RFC)	0.9133	0.87	0.94	0.91	0.5621
Gradient BoostingClassifier	0.9022	0.86	0.94	0.90	0.6113
Ada Boost Classifier	0.5932	0.39	0.60	0.59	0.5204
Final Model (RFC Tuned)	0.9136	0.87	0.94	0.91	0.573

Final Model is giving us Accuracy score of 91.36% which is slightly improved compare to earlier Accuracy score of 91.33%

Learning Outcomes of the Study in respect of Data Science

➢ Hands on chance to enhance my web scraping skillset.

➢ 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.

Limitations of this work and Scope for Future Work

➢ More input features can be scrap to build predication model.

➢ There is scope for application of advanced deep learning NLP tool to enhanced text mining operation which eventually help in building more accurate model with good cross validation score