NAME OF THE PROJECT

PRODUCT REVIEW RATING PREDICTION USING NL

Submitted by:

SNEHA SANTRA

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. Khushboo Garg (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 “Data trained” who are the reason behind my Internship at Fliprobo.

References use in this project:

1. SCIKIT Learn Library Documentation
2. Blogs from towardsdatascience, Analytics Vidya, Medium
3. Andrew Ng Notes on Machine Learning (GitHub)
4. Data Science Projects with Python Second Edition by Packt
5. Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron

INTRODUCTION

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

- Conceptual Background of the Domain Problem
    - The project is sub-divided following section. These are:
    - 1. Loading necessary libraries
    - 2. Loading Dataset from a CSV file
    - 3. Summarization of Data to understand Dataset (Descriptive Statistics)
    - 4. Visualization of Data to understand Dataset (Plots, Graphs etc.)
    - 5.Processing the data for modeling
    - 6.Build the model and select the right model and save it

- Review of Literature

  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 a 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 the first provided to me by Flip Robo Technologies as a part of the internship programme. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary motivation.

Analytical Problem Framing

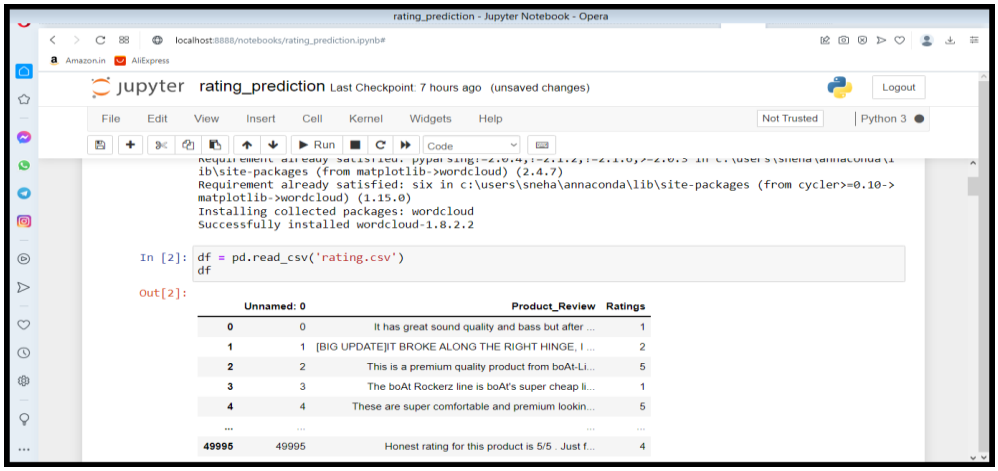
- Mathematical/ Analytical Modeling of the Problem

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.

In the field of text classification, documents are commonly treated like a Bag-of-Words (BoW), meaning that each word is independent from the others that are present in the document. 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. As its name suggests, the term frequency simply weights each term based on their occurrence frequency and does not take the discriminatory power of terms into account.

- 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.



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.

```
df.info() #Checking the datatype of all the columns present

<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
```

- <u>Data Preprocessing Done</u>

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'.

```
# Replacing missing data with 'Review Not Available' using pandas fillna()
df['Product_Review'].fillna('Review Not Available',inplace=True)
```

```
df.isnull().sum().any()  #Checking after filling them
```

```
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 i3 processor with 1.70GHZ
2. RAM — 12 GB

Software utilised -

1.Anaconda – Jupyter Notebook

2. Selenium – Web scraping

3. Google Colab – for Hyper parameter tuning

Libraries Used – General library for data wrangling & visualsation

```
#Importing warning library to avoid any warnings
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # ploting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

Libraries used for Text Mining / Text Analytics are:

```
#Importing required libraries
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 web scraping data from e-commerce website are:

```
#Importing required libraries
import pandas as pd
import selenium
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException , StaleElementReferenceException
import time
import warnings
warnings.filterwarnings('ignore')
```

Libraries used for machine learning model building:

```
#Importing Machine Learning Model Library
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)

Just make the comments more appropriate so that we'll get less word to process and get more accuracy. Removed extra spaces, converted email address into email keyword, likely wise phone number etc. Tried to make Comments small and more appropriate as much as it was possible.

- Testing of Identified Approaches (Algorithms)

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

1.Logistic Regression

2.Random Forest Classifier

3.DecisionTree Classifier

4.AdaBoost Classifier

5.Gradient boosting classifier

- Run and Evaluate selected models

1. **Logistics 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
```

```
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
```

DecisionTree Classifier:

```
Accuracy Score of Decision Tree Classifier : 0.8967333333333334


Confusion matrix of Decision Tree Classifier :
 [[3338    7    3    9   21]
 [  35  573    0    2    7]
 [  38    1  825   36  207]
 [  38    7   23 1806  595]
 [ 125    9  103  283 6909]]


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

         1.0       0.93      0.99      0.96      3378
         2.0       0.96      0.93      0.94       617
         3.0       0.86      0.75      0.80      1107
         4.0       0.85      0.73      0.78      2469
         5.0       0.89      0.93      0.91      7429

    accuracy                           0.90     15000
   macro avg       0.90      0.86      0.88     15000
weighted avg       0.89      0.90      0.89     15000
```

Random Forest Classifier:

```
Accuracy Score of Random Forest Classifier : 0.9133333333333333


Confusion matrix of Random Forest Classifier :
 [[3334    7    3    9   25]
 [  35  573    0    2    7]
 [  36    0  821   11  239]
 [  23    3    7 1742  694]
 [  79    4   17   99 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       0.94      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
```

AdaBoost Classifier:

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
```

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    8  254]
 [  39    3    8 1700  719]
 [ 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       0.98      0.93      0.95       617
         3.0       0.98      0.74      0.84      1107
         4.0       0.97      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
```

5-fold Cross validation performed over all model. We can see that Random Forest Classifier gives us good Accuracy and maximum f1 score along with best Cross-validation score. Hyperparameter tuning is applied over Random Forest model and used it as final model.

```
parameter = { 'max_features': ['auto', 'log2'],
              'criterion':['gini','entropy'],
              'n_estimators': [75,100,150]}
```

```
GCV = GridSearchCV(RandomForestClassifier(),parameter,verbose=10)
GCV.fit(X_train,Y_train)
```

```
[CV 4/5; 11/12] END criterion=entropy, max_features=log2, n_estimators=100;, score=0.905 total time= 2.3min
[CV 5/5; 11/12] START criterion=entropy, max_features=log2, n_estimators=100....
[CV 5/5; 11/12] END criterion=entropy, max_features=log2, n_estimators=100;, score=0.906 total time= 2.3min
[CV 1/5; 12/12] START criterion=entropy, max_features=log2, n_estimators=150....
[CV 1/5; 12/12] END criterion=entropy, max_features=log2, n_estimators=150;, score=0.905 total time= 3.6min
[CV 2/5; 12/12] START criterion=entropy, max_features=log2, n_estimators=150....
[CV 2/5; 12/12] END criterion=entropy, max_features=log2, n_estimators=150;, score=0.911 total time= 3.5min
[CV 3/5; 12/12] START criterion=entropy, max_features=log2, n_estimators=150....
[CV 3/5; 12/12] END criterion=entropy, max_features=log2, n_estimators=150;, score=0.905 total time= 3.5min
[CV 4/5; 12/12] START criterion=entropy, max_features=log2, n_estimators=150....
[CV 4/5; 12/12] END criterion=entropy, max_features=log2, n_estimators=150;, score=0.905 total time= 3.6min
[CV 5/5; 12/12] START criterion=entropy, max_features=log2, n_estimators=150....
[CV 5/5; 12/12] END criterion=entropy, max_features=log2, n_estimators=150;, score=0.906 total time= 3.5min

GridSearchCV(estimator=RandomForestClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_features': ['auto', 'log2'],
                         'n_estimators': [75, 100, 150]},
             verbose=10)
```

```
GCV.best_params_
```

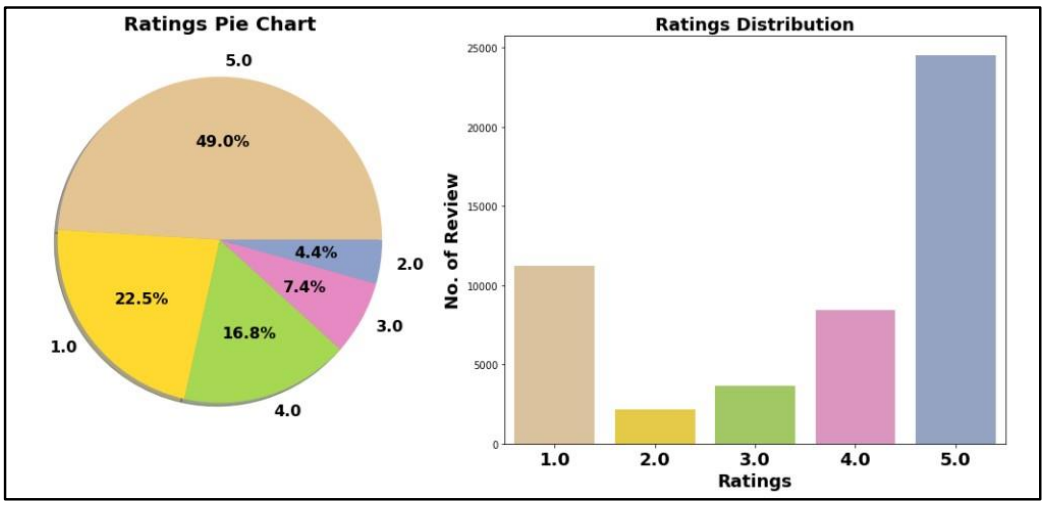
```
{'criterion': 'entropy', 'max_features': 'auto', 'n_estimators': 150}
```

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

```
Final Random Forest Classifier Model
Accuracy Score :
 0.9136


Confusion matrix of Random Forest Classifier :
 [[3334    7    3    9   25]
 [  35  573    0    2    7]
 [  35    0  821   11  240]
 [  23    3    7 1736  700]
 [  79    4   17   89 7240]]


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      0.70      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
```

- Visualizations

5. Visualizations



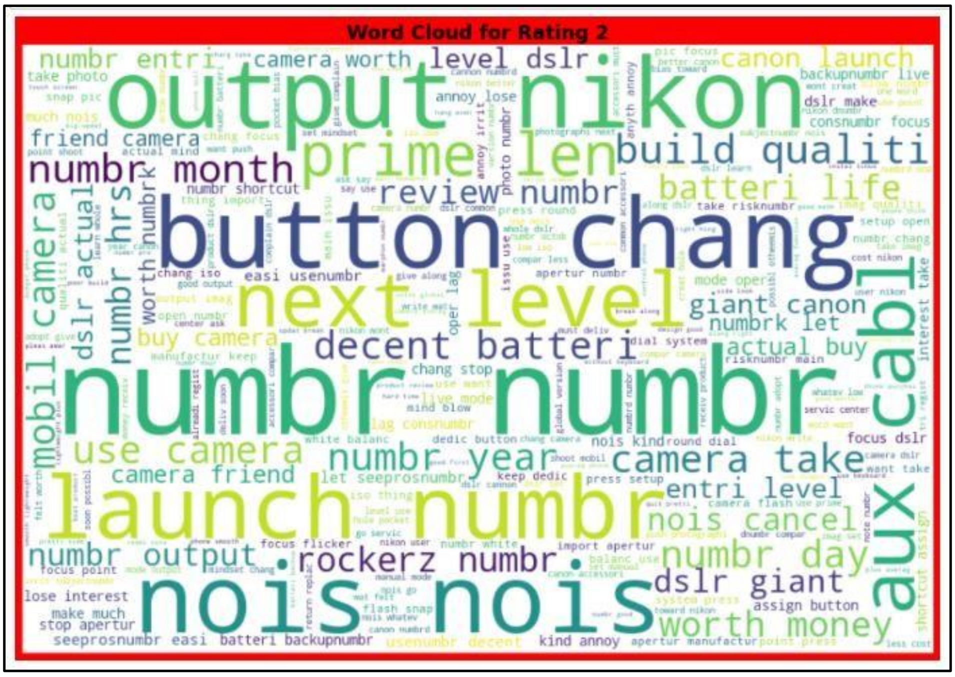
Comment:

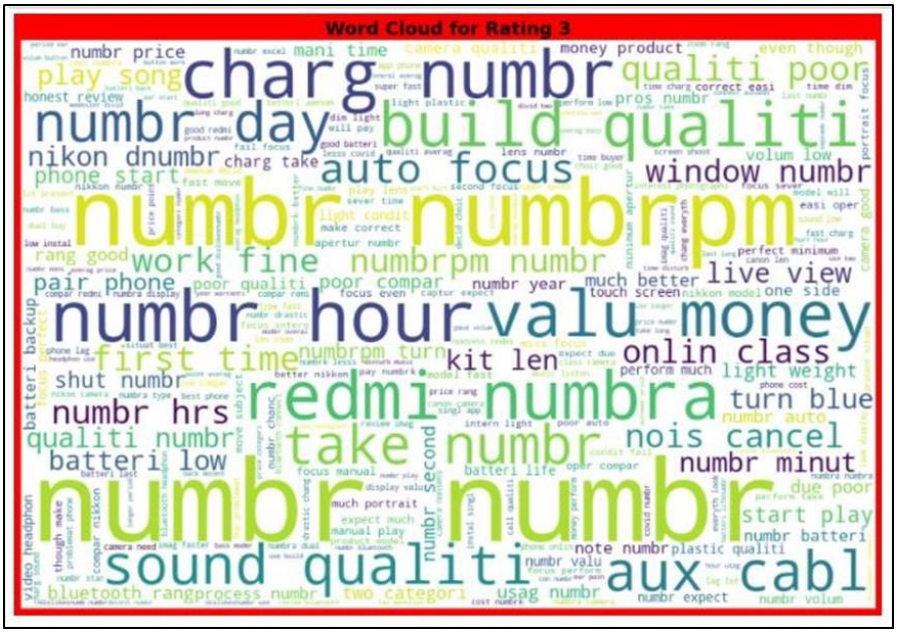
1. Around 49% customer given 5- star rating followed by 22.5% customer given lowest 1-star rating.
2. Average Rating is 3.65.

Wordcloud:

- Word Cloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency.
- The more commonly the term appears within the text being analysed, the larger the word appears in the image generated.
- The enlarged texts are the greatest number of words used there and small texts are the smaller number of words used.


Word Cloud for Rating 1
numbr numbr
wast money
numbr day
camera qualiti
qualiti good
sound qualiti
pleas buy
servic center
soon possibl
buy product
nois cancel
buy mobil
build qualiti
batteri life
use numbr
usb cabl
replac camera
numbr month
front camera
transfer photo
use headphon
numbrgb ram
numbr sec
apart usb
buy phone
heat issu
requir transfer
juli numbr
buy laptop
photo laptop
cabl requir
first time
want replac
two day
deliveri numbrth
numbr order
local servic
bad qualiti
boat rockerz
samsung mnumbr
laptop desktop
numbrmp
camera
numbr year
inform order
dslr camera
nikon dnumbr
dnumbr dslr
surpris find
reciev deliveri
box want
borosil toaster
make complaint
video order
amazon reciev
aux cabl
packag surpris
toaster proof
numbr pro


Word Cloud for Rating 2
output nikon
button chang
next level
numbr numbr
launch numbr
nois nois
prime len
build qualiti
numbr month
batteri life
review
decent batteri
aux cabl
camera take
use camera
numbr year
worth money
numbr day
nois cancel
entri level
dslr giant
rockerz numbr
numbr output
giant canon
actual buy
mobil camera
dslr actual
numbr hrs
camera friend
friend camera
numbr entri
camera worth
level dslr
canon launch
buy camera
numbrk let
lose interest
make much
stop apertur
seeprosnumbr easi batteri backupnumbr
kind annoy
apertur manufactur
focus dslr



CONCLUSION

- Key Findings and Conclusions of the Study

Algorithm	Accuracy Recall Score		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 Boosting Classifier	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	0.5730

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.