**NAME OF THE PROJECT**

**Product Review Rating**

**Predication Using NLP**

**Submitted by:**

**SAUNAK MUKHERJEE**



**ACKNOWLEDGMENT**

I would like to express my special gratitude to THE “Flip Robo” team, they have given me this opportunity to deal with a beautiful dataset and it has

helped me to improve analyzation of skills. And I want to express my

huge gratitude to Ms. Sapna Verma (SME Flip Robo), she is the person

who has helped me When I am in difficulties I faced while doing the

project. And also 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

6. Lackermair, G., Kailer, D. & Kanmaz, K. (2013). Importance of online

product reviews from a consumer’s perspective. Horizon

Research Publishing, 1-5. doi: 10.13189/aeb.2013.010101

7. Baccianella, S., Esuli, A. & Sebastiani, F. (2009). Multi-facet rating

of product reviews. Proceedings of the 31st European Conference

on Information Retrieval (ECIR), 461- 472

8. Chevalier, J. & Mayzlin, D. (2006). The Effect of Word of Mouth on

Sales: Online Book Reviews. Journal of Marketing Research, 43,

345-354. doi: 10.3386/w10148

9. Pang, B., Lee, L. & Vaithyanathan, S. (2002). Thumbs up? Sentiment

Classification using Machine Learning Techniques. Proceedings of

EMNLP, 271-278. doi: 10.3115/1118693.1118704

10. Dave, K., Lawrence, S. & Pennock, D. M. (2003). Mining the Peanut

Gallery: Opinion Extraction and Semantic Classification of Product

Reviews. Proceedings of the 12th International Conference on

World Wide Web, 519–528. doi: 10.1145/775152.775226

11. McAuley, J., Pandey, R., Leskovec J. (2015). Inferring networks of

substitutable and complementary products. Knowledge Discovery

and Data Mining.3

12. A NithyaKalyani, S Ushasukhanya, TYJ Nagamalleswari, and S

Girija. 2018. Rating prediction using textual reviews. In Journal of

Physics: Conference Series, Vol. 1000. IOP Publishing, 012044.

13. Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag

of-opinions method for review rating prediction from sparse text

patterns. In Proceedings of the 23rd international conference on

computational linguistics. Association for Computational

Linguistics, 913–921

14. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Recommender

Systems Handboo,Springer, 2011.

15. Leon, Vasant, Sheldon, Using Properties of the Amazon Graph to

Better Understand Reviews

Business Problem Framing

Internet is the best source where we can find nowadays for any organisation to know the public opinions about their products and services. Many consumers form an opinion about the product just by the reading a few reviews. Online product that we reviews provided by the consumers who previously purchased products to have become the major information of the source for consumers and marketers

regarding product of the quality. Research that has been shown that consumer online

product ratings of the reflect both the customers' experience with the product

and the influence of others' ratings. Websites which has prominently display consumers' product ratings, which influence the consumers' buying why decisions and willingness to pay.

The opinion which 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. Hence, in recent years,

sentiment analysis and opinion mining have become a popular topic for

machine learning and data mining.

We have the client who has a website where people write different

reviews for technical products. Now they are adding to new feature to

their website, i.e. The reviewer will have to add star as well with

the review. The rating which 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.

Data Sources and their formats

Data is collected from flipkart.com using selenium and

saved in CSV file. Around 20000 Reviews are collected for this project.

<bound method DataFrame.info of Product\_Review Ratings

0 Fantastic ❤️ stylish and comes wd super qualit... 5

1 Very nice, can wear in night also 5

2 Worth for the money, awesome product and quali... 5

3 I'm quite happy with the product. A little bit... 5

4 The frame is good looks premium.. but i dont k... 4

... ... ...

21828 Low price and very good quality I'm very happy... 5

21829 Great Experience and Nice shoes\nIt's awesome ... 4

21830 Grt product only little size issue 5

21831 Very Nice Product I love it 5

21832 Excellent fit and comfortable as well as stylish 5

[21833 rows x 2 columns]>

This are 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. There are also some missing values are there in product review. The datatype of the

Product review is object while datatypes of Ratings is float.

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

**Libraries used for Text Mining**

import re

import string

import nltk

from nltk.corpus import stopwords

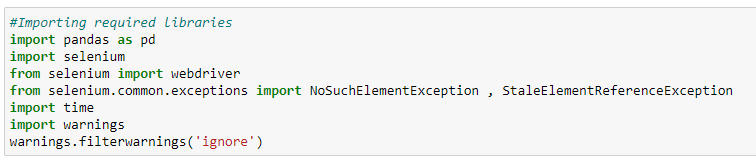
from nltk.tokenize import word\_tokenize

from nltk.stem import SnowballStemmer, WordNetLemmatizer

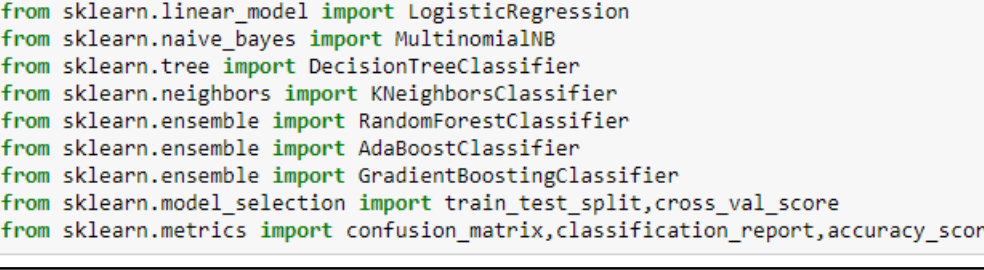
from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer

from wordcloud import WordCloud

Libraries used for web scraping data from e-commerce website are



Libraries used for machine learning model building



Testing of Identified Approaches (Algorithms)

The different classification algorithm used in this project to build ML

model are as below:

❖ Decision Tree classifier

❖ Logistics Regression

❖ AdaBoost Classifier

❖ Gradient Boosting Classifier

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

Key Metrics for Success in Solving Problem

Under Consideration

▪ Precision that can be seen as a measure of the quality; higher precision means which is an algorithm returns more relevant results than

irrelevant ones.

▪ Recall that also used as a measure of quantity and high recall means that an

algorithm returns most of the relevant results.

▪ Accuracy score which is also used when the True Positives and True negatives

are more important. Accuracy can be used when the class

distribution is similar.

▪ F1-score that is used when the False Negatives and False Positives are

crucial. While F1-score is a better metric when there are imbalanced

classes

Run And Evaluate Selected Models

Logistics Regression

[{"metadata":{"trusted":false},"id":"cc5f2b4d","cell\_type":"code","source":"X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, random\_state=67, test\_size=.3)\nprint('Training feature matrix size:',X\_train.shape)\nprint('Training target vector size:',Y\_train.shape)\nprint('Test feature matrix size:',X\_test.shape)\nprint('Test target vector size:',Y\_test.shape)","execution\_count":50,"outputs":[{"name":"stdout","output\_type":"stream","text":"Training feature matrix size: (15283, 5417)\nTraining target vector size: (15283, 1)\nTest feature matrix size: (6550, 5417)\nTest target vector size: (6550, 1)\n"}]}]

Train-test split is used to split data into the training data & testing data.

Further best random state is investigated through loop.

dtc=DecisionTreeClassifier()

dtc.fit(X\_train,Y\_train)

y\_pred=dtc.predict(X\_test)

print('\033[1m'+'Decision Tree Classifier Evaluation'+'\033[0m')

print('\n')

print('\033[1m'+'Accuracy Score of Decision Tree Classifier :'+'\033[0m', accuracy\_score(Y\_test, y\_pred))

print('\n')

print('\033[1m'+'Confusion matrix of Decision Tree Classifier :'+'\033[0m \n',confusion\_matrix(Y\_test, y\_pred))

print('\n')

print('\033[1m'+'classification Report of Decision Tree Classifier'+'\033[0m \n',classification\_report(Y\_test, y\_pred))

Decision Tree Classifier

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

below:

dtc=DecisionTreeClassifier()

dtc.fit(X\_train,Y\_train)

y\_pred=dtc.predict(X\_test)

print('\033[1m'+'Decision Tree Classifier Evaluation'+'\033[0m')

print('\n')

print('\033[1m'+'Accuracy Score of Decision Tree Classifier :'+'\033[0m', accuracy\_score(Y\_test, y\_pred))

print('\n')

print('\033[1m'+'Confusion matrix of Decision Tree Classifier :'+'\033[0m \n',confusion\_matrix(Y\_test, y\_pred))

print('\n')

print('\033[1m'+'classification Report of Decision Tree Classifier'+'\033[0m \n',classification\_report(Y\_test, y\_pred))

**Decision Tree Classifier Evaluation**

**Accuracy Score of Decision Tree Classifier :** 0.7409160305343512

**Confusion matrix of Decision Tree Classifier :**

[[ 155 45 286]

[ 36 645 917]

[ 95 318 4053]]

**classification Report of Decision Tree Classifier**

precision recall f1-score support

3 0.54 0.32 0.40 486

4 0.64 0.40 0.50 1598

5 0.77 0.91 0.83 4466

accuracy 0.74 6550

macro avg 0.65 0.54 0.58 6550

weighted avg 0.72 0.74 0.72 6550

Ada Boost Classifier

**AdaBoost Classifier Evaluation**

**Accuracy Score of AdaBoost Classifier :** 0.7331297709923664

**Confusion matrix of AdaBoost Classifier :**

[[ 106 4 376]

[ 5 320 1273]

[ 22 68 4376]]

**classification Report of AdaBoost Classifier**

precision recall f1-score support

3 0.80 0.22 0.34 486

4 0.82 0.20 0.32 1598

5 0.73 0.98 0.83 4466

accuracy 0.73 6550

macro avg 0.78 0.47 0.50 6550

weighted avg 0.75 0.73 0.67 6550

Gradient Boosting Classifier

**Gradient Boosting Classifier Evaluation**

**Accuracy Score of Gradient Boosting Classifier :** 0.7506870229007634

**Confusion matrix of Gradient Boosting Classifier :**

[[ 103 1 382]

[ 1 365 1232]

[ 0 17 4449]]

**classification Report of Gradient Boosting Classifier**

precision recall f1-score support

3 0.99 0.21 0.35 486

4 0.95 0.23 0.37 1598

5 0.73 1.00 0.85 4466

accuracy 0.75 6550

macro avg 0.89 0.48 0.52 6550

weighted avg 0.81 0.75 0.69 6550

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.

Fitting 5 folds for each of 12 candidates, totalling 60 fits

[CV 1/5; 1/12] START criterion=gini, max\_features=auto, n\_estimators=75.........

[CV 1/5; 1/12] END criterion=gini, max\_features=auto, n\_estimators=75;, score=0.768 total time= 14.3s

[CV 2/5; 1/12] START criterion=gini, max\_features=auto, n\_estimators=75.........

[CV 2/5; 1/12] END criterion=gini, max\_features=auto, n\_estimators=75;, score=0.775 total time= 14.1s

[CV 3/5; 1/12] START criterion=gini, max\_features=auto, n\_estimators=75.........

[CV 3/5; 1/12] END criterion=gini, max\_features=auto, n\_estimators=75;, score=0.765 total time= 14.0s

[CV 4/5; 1/12] START criterion=gini, max\_features=auto, n\_estimators=75.........

[CV 4/5; 1/12] END criterion=gini, max\_features=auto, n\_estimators=75;, score=0.779 total time= 14.1s

[CV 5/5; 1/12] START criterion=gini, max\_features=auto, n\_estimators=75.........

[CV 5/5; 1/12] END criterion=gini, max\_features=auto, n\_estimators=75;, score=0.777 total time= 14.1s

[CV 1/5; 2/12] START criterion=gini, max\_features=auto, n\_estimators=100........

[CV 1/5; 2/12] END criterion=gini, max\_features=auto, n\_estimators=100;, score=0.769 total time= 18.7s

[CV 2/5; 2/12] START criterion=gini, max\_features=auto, n\_estimators=100........

[CV 2/5; 2/12] END criterion=gini, max\_features=auto, n\_estimators=100;, score=0.774 total time= 19.0s

[CV 3/5; 2/12] START criterion=gini, max\_features=auto, n\_estimators=100........

[CV 3/5; 2/12] END criterion=gini, max\_features=auto, n\_estimators=100;, score=0.764 total time= 18.9s

[CV 4/5; 2/12] START criterion=gini, max\_features=auto, n\_estimators=100........

[CV 4/5; 2/12] END criterion=gini, max\_features=auto, n\_estimators=100;, score=0.776 total time= 18.8s

[CV 5/5; 2/12] START criterion=gini, max\_features=auto, n\_estimators=100........

[CV 5/5; 2/12] END criterion=gini, max\_features=auto, n\_estimators=100;, score=0.777 total time= 19.0s

[CV 1/5; 3/12] START criterion=gini, max\_features=auto, n\_estimators=150........

[CV 1/5; 3/12] END criterion=gini, max\_features=auto, n\_estimators=150;, score=0.768 total time= 28.0s

[CV 2/5; 3/12] START criterion=gini, max\_features=auto, n\_estimators=150........

[CV 2/5; 3/12] END criterion=gini, max\_features=auto, n\_estimators=150;, score=0.775 total time= 28.4s

[CV 3/5; 3/12] START criterion=gini, max\_features=auto, n\_estimators=150........

[CV 3/5; 3/12] END criterion=gini, max\_features=auto, n\_estimators=150;, score=0.763 total time= 28.2s

[CV 4/5; 3/12] START criterion=gini, max\_features=auto, n\_estimators=150........

[CV 4/5; 3/12] END criterion=gini, max\_features=auto, n\_estimators=150;, score=0.776 total time= 28.4s

[CV 5/5; 3/12] START criterion=gini, max\_features=auto, n\_estimators=150........

[CV 5/5; 3/12] END criterion=gini, max\_features=auto, n\_estimators=150;, score=0.776 total time= 28.4s

[CV 1/5; 4/12] START criterion=gini, max\_features=log2, n\_estimators=75.........

[CV 1/5; 4/12] END criterion=gini, max\_features=log2, n\_estimators=75;, score=0.768 total time= 16.8s

[CV 2/5; 4/12] START criterion=gini, max\_features=log2, n\_estimators=75.........

[CV 2/5; 4/12] END criterion=gini, max\_features=log2, n\_estimators=75;, score=0.776 total time= 16.4s

[CV 3/5; 4/12] START criterion=gini, max\_features=log2, n\_estimators=75.........

[CV 3/5; 4/12] END criterion=gini, max\_features=log2, n\_estimators=75;, score=0.768 total time= 16.9s

[CV 4/5; 4/12] START criterion=gini, max\_features=log2, n\_estimators=75.........

[CV 4/5; 4/12] END criterion=gini, max\_features=log2, n\_estimators=75;, score=0.778 total time= 17.0s

[CV 5/5; 4/12] START criterion=gini, max\_features=log2, n\_estimators=75.........

[CV 5/5; 4/12] END criterion=gini, max\_features=log2, n\_estimators=75;, score=0.777 total time= 16.5s

[CV 1/5; 5/12] START criterion=gini, max\_features=log2, n\_estimators=100........

[CV 1/5; 5/12] END criterion=gini, max\_features=log2, n\_estimators=100;, score=0.767 total time= 22.4s

[CV 2/5; 5/12] START criterion=gini, max\_features=log2, n\_estimators=100........

[CV 2/5; 5/12] END criterion=gini, max\_features=log2, n\_estimators=100;, score=0.775 total time= 22.6s

[CV 3/5; 5/12] START criterion=gini, max\_features=log2, n\_estimators=100........

[CV 3/5; 5/12] END criterion=gini, max\_features=log2, n\_estimators=100;, score=0.765 total time= 22.4s

[CV 4/5; 5/12] START criterion=gini, max\_features=log2, n\_estimators=100........

[CV 4/5; 5/12] END criterion=gini, max\_features=log2, n\_estimators=100;, score=0.774 total time= 22.2s

[CV 5/5; 5/12] START criterion=gini, max\_features=log2, n\_estimators=100........

[CV 5/5; 5/12] END criterion=gini, max\_features=log2, n\_estimators=100;, score=0.777 total time= 21.9s

[CV 1/5; 6/12] START criterion=gini, max\_features=log2, n\_estimators=150........

[CV 1/5; 6/12] END criterion=gini, max\_features=log2, n\_estimators=150;, score=0.769 total time= 33.5s

[CV 2/5; 6/12] START criterion=gini, max\_features=log2, n\_estimators=150........

[CV 2/5; 6/12] END criterion=gini, max\_features=log2, n\_estimators=150;, score=0.776 total time= 33.7s

[CV 3/5; 6/12] START criterion=gini, max\_features=log2, n\_estimators=150........

[CV 3/5; 6/12] END criterion=gini, max\_features=log2, n\_estimators=150;, score=0.765 total time= 33.5s

[CV 4/5; 6/12] START criterion=gini, max\_features=log2, n\_estimators=150........

[CV 4/5; 6/12] END criterion=gini, max\_features=log2, n\_estimators=150;, score=0.777 total time= 33.8s

[CV 5/5; 6/12] START criterion=gini, max\_features=log2, n\_estimators=150........

[CV 5/5; 6/12] END criterion=gini, max\_features=log2, n\_estimators=150;, score=0.777 total time= 33.0s

[CV 1/5; 7/12] START criterion=entropy, max\_features=auto, n\_estimators=75......

[CV 1/5; 7/12] END criterion=entropy, max\_features=auto, n\_estimators=75;, score=0.768 total time= 13.8s

[CV 2/5; 7/12] START criterion=entropy, max\_features=auto, n\_estimators=75......

[CV 2/5; 7/12] END criterion=entropy, max\_features=auto, n\_estimators=75;, score=0.775 total time= 14.0s

[CV 3/5; 7/12] START criterion=entropy, max\_features=auto, n\_estimators=75......

[CV 3/5; 7/12] END criterion=entropy, max\_features=auto, n\_estimators=75;, score=0.766 total time= 13.8s

[CV 4/5; 7/12] START criterion=entropy, max\_features=auto, n\_estimators=75......

[CV 4/5; 7/12] END criterion=entropy, max\_features=auto, n\_estimators=75;, score=0.774 total time= 14.0s

[CV 5/5; 7/12] START criterion=entropy, max\_features=auto, n\_estimators=75......

[CV 5/5; 7/12] END criterion=entropy, max\_features=auto, n\_estimators=75;, score=0.775 total time= 13.9s

[CV 1/5; 8/12] START criterion=entropy, max\_features=auto, n\_estimators=100.....

[CV 1/5; 8/12] END criterion=entropy, max\_features=auto, n\_estimators=100;, score=0.768 total time= 18.4s

[CV 2/5; 8/12] START criterion=entropy, max\_features=auto, n\_estimators=100.....

[CV 2/5; 8/12] END criterion=entropy, max\_features=auto, n\_estimators=100;, score=0.775 total time= 18.5s

[CV 3/5; 8/12] START criterion=entropy, max\_features=auto, n\_estimators=100.....

[CV 3/5; 8/12] END criterion=entropy, max\_features=auto, n\_estimators=100;, score=0.764 total time= 18.3s

[CV 4/5; 8/12] START criterion=entropy, max\_features=auto, n\_estimators=100.....

[CV 4/5; 8/12] END criterion=entropy, max\_features=auto, n\_estimators=100;, score=0.777 total time= 18.5s

[CV 5/5; 8/12] START criterion=entropy, max\_features=auto, n\_estimators=100.....

[CV 5/5; 8/12] END criterion=entropy, max\_features=auto, n\_estimators=100;, score=0.775 total time= 18.5s

[CV 1/5; 9/12] START criterion=entropy, max\_features=auto, n\_estimators=150.....

[CV 1/5; 9/12] END criterion=entropy, max\_features=auto, n\_estimators=150;, score=0.769 total time= 27.4s

[CV 2/5; 9/12] START criterion=entropy, max\_features=auto, n\_estimators=150.....

[CV 2/5; 9/12] END criterion=entropy, max\_features=auto, n\_estimators=150;, score=0.775 total time= 27.8s

[CV 3/5; 9/12] START criterion=entropy, max\_features=auto, n\_estimators=150.....

[CV 3/5; 9/12] END criterion=entropy, max\_features=auto, n\_estimators=150;, score=0.767 total time= 27.4s

[CV 4/5; 9/12] START criterion=entropy, max\_features=auto, n\_estimators=150.....

[CV 4/5; 9/12] END criterion=entropy, max\_features=auto, n\_estimators=150;, score=0.775 total time= 27.9s

[CV 5/5; 9/12] START criterion=entropy, max\_features=auto, n\_estimators=150.....

[CV 5/5; 9/12] END criterion=entropy, max\_features=auto, n\_estimators=150;, score=0.776 total time= 27.8s

[CV 1/5; 10/12] START criterion=entropy, max\_features=log2, n\_estimators=75.....

[CV 1/5; 10/12] END criterion=entropy, max\_features=log2, n\_estimators=75;, score=0.770 total time= 16.5s

[CV 2/5; 10/12] START criterion=entropy, max\_features=log2, n\_estimators=75.....

[CV 2/5; 10/12] END criterion=entropy, max\_features=log2, n\_estimators=75;, score=0.775 total time= 16.5s

[CV 3/5; 10/12] START criterion=entropy, max\_features=log2, n\_estimators=75.....

[CV 3/5; 10/12] END criterion=entropy, max\_features=log2, n\_estimators=75;, score=0.763 total time= 16.4s

[CV 4/5; 10/12] START criterion=entropy, max\_features=log2, n\_estimators=75.....

[CV 4/5; 10/12] END criterion=entropy, max\_features=log2, n\_estimators=75;, score=0.774 total time= 16.4s

[CV 5/5; 10/12] START criterion=entropy, max\_features=log2, n\_estimators=75.....

[CV 5/5; 10/12] END criterion=entropy, max\_features=log2, n\_estimators=75;, score=0.777 total time= 16.7s

[CV 1/5; 11/12] START criterion=entropy, max\_features=log2, n\_estimators=100....

[CV 1/5; 11/12] END criterion=entropy, max\_features=log2, n\_estimators=100;, score=0.767 total time= 22.0s

[CV 2/5; 11/12] START criterion=entropy, max\_features=log2, n\_estimators=100....

[CV 2/5; 11/12] END criterion=entropy, max\_features=log2, n\_estimators=100;, score=0.776 total time= 22.1s

[CV 3/5; 11/12] START criterion=entropy, max\_features=log2, n\_estimators=100....

[CV 3/5; 11/12] END criterion=entropy, max\_features=log2, n\_estimators=100;, score=0.765 total time= 21.8s

[CV 4/5; 11/12] START criterion=entropy, max\_features=log2, n\_estimators=100....

[CV 4/5; 11/12] END criterion=entropy, max\_features=log2, n\_estimators=100;, score=0.777 total time= 22.1s

[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=0.775 total time= 21.7s

[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=0.768 total time= 32.8s

[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=0.776 total time= 33.0s

[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=0.766 total time= 33.1s

[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=0.775 total time= 33.0s

[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=0.775 total time= 32.4s

GridSearchCV(estimator=RandomForestClassifier(),

param\_grid={'criterion': ['gini', 'entropy'],

'max\_features': ['auto', 'log2'],

'n\_estimators': [75, 100, 150]},

verbose=10)

Result:

**Final Random Forest Classifier ModelAccuracy Score :**

0.7819847328244275

**Confusion matrix of Random Forest Classifier :**

[[ 139 8 339]

[ 6 559 1033]

[ 14 28 4424]]

**Classification Report of Random Forest Classifier**

precision recall f1-score support

3 0.87 0.29 0.43 486

4 0.94 0.35 0.51 1598

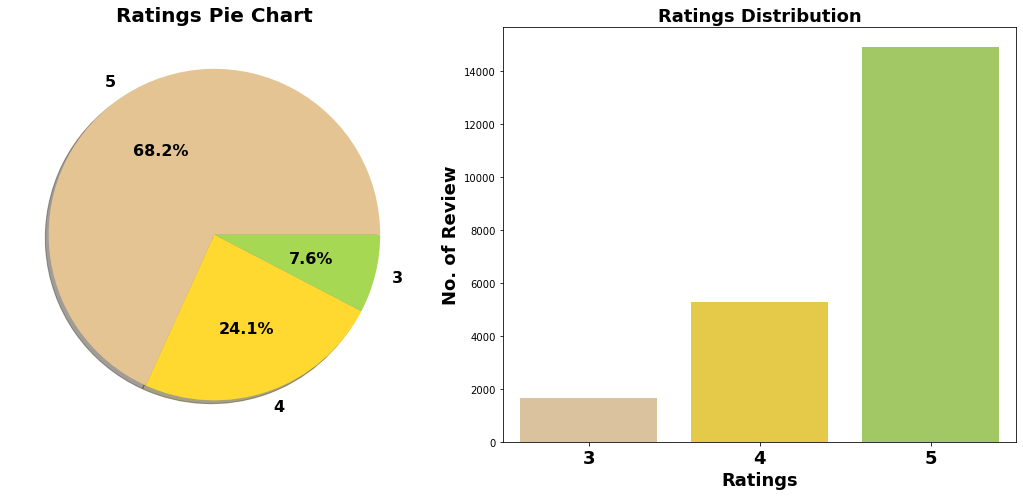
5 0.76 0.99 0.86 4466

accuracy 0.78 6550

macro avg 0.86 0.54 0.60 6550

weighted avg 0.81 0.78 0.74 6550

Visualizations



1. Around 68.2% customer given 5- star rating followed by 24.11% customer given 4-star rating.

2. 7.6% customer given 3 star

Key Findings and Conclusions of the Study

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Precision** | **Recall** | **F1 Score** | **CV Score** |
| **Logistics Regression** | 0.74 | 0.99 | 0.84 | 0.63 |
| **Decision Tree Classifier** | 0.77 | 0.91 | 0.83 | 0.61 |
| **Ada Boost Classifier** | 0.73 | 0.98 | 0.83 | 0.63 |
| **Gradient Boosting Classifier** | 0.73 | 1.00 | 0.85 | 0.64 |
| **Final Model (RFC- Tuned)** | 0.76 | 0.99 | 0.86 |  |

Final Model is giving us Accuracy score of 78% which is slightly

improved compare to earlier Accuracy score of 78%