FLIP ROBO

RATINGS PREDICTION PROJECT

Submitted by:

DEBANTI ROY

ACKNOWLEDGMENT

I, Debanti Roy, would like to convey my sincere gratitude to DataTrained Academy and Flip Robo Technologies for giving me this opportunity to do this project. I would like to thank all mentors and SME's for extending their support all through the process which helped me complete this project.

E-source: https://terakeet.com/blog/online-reviews/

INTRODUCTION

- **Business Problem Framing**

The proliferation of social media enables people to express their opinions widely online. However, it is a review of a product or service where it reflects the opinions and experiences of a customer purchasing a product or service. For customers, online reviews and testimonials are all about building trust. And that's especially true for ecommerce sales when shoppers can't ask store associates for product information before buying.

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**

Online reviews build brand trust among the audience. If potential customers know that other people had positive experiences with a brand, they're more likely to trust the brand. Reviews build brand credibility and increase the likelihood that consumers will purchase from that brand.

Online reviews also validate the company's expertise in the eyes of prospective customers. They prove that the brand/company has successfully helped other people overcome a specific challenge or accomplish a particular goal.

Anyone can make claims about how amazing their product or service is, but that doesn't mean those claims are true. By providing evidence of success through online reviews, brands can demonstrate their expertise and authority.

Most consumers trust online reviews by their peers more than the claims made by a company. Consumers are less likely to believe a claim if there isn't any third-party supporting evidence. Good reviews are a great way to back up any claims made by a company.

- **Data Set Description**

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 many categories of reviews, we can do good amount of data exploration and derive some interesting features using the comments text column available.

- **Review of Literature**

Online reviews are just as essential for businesses as they are for customers, but for different reasons. For example, they establish stronger relationships with potential customers, and they allow companies to charge a premium price for their products.

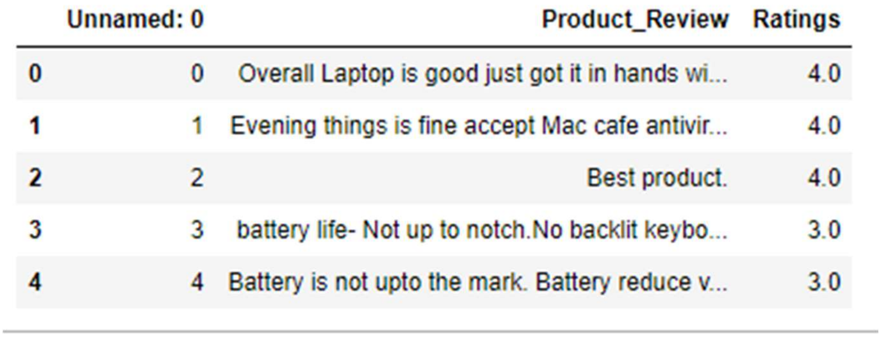
One of the more underrated advantages of online reviews is that they allow a brand to expand its online presence beyond the website and social media channels. Customers can learn about the brand on authoritative third-party sites, like Google, Angi, and Yelp. It can directly influence the buying decisions of the target market as it helps the customers to compare products or services, they often choose the ones that have more positive online reviews. A lack of reviews makes buyers feel increased risk, which makes them less likely to buy.

Analytical Problem Framing

- **Data Sources and their formats**

The data set is scrapped from online shopping websites like Amazon and Flipkart. The dataset contains 31095 rows and 3 columns. These are enclosed in CSV format. All the data samples contain 3 fields which includes ‘Unnamed:0, ‘Product_Review’ and ‘Ratings’.

In the Ratings field the comments are rated on a scale of 1 to 5.



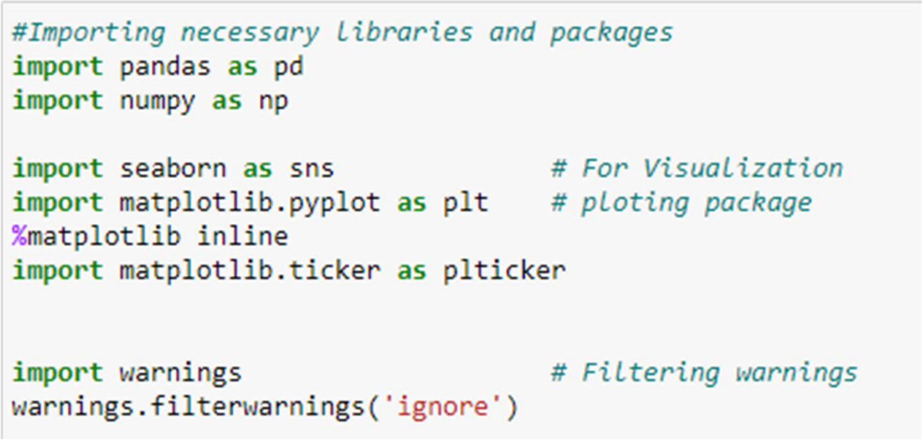
	Unnamed: 0	Product_Review	Ratings
0	0	Overall Laptop is good just got it in hands wi...	4.0
1	1	Evening things is fine accept Mac cafe antivir...	4.0
2	2	Best product.	4.0
3	3	battery life- Not up to notch.No backlit keybo...	3.0
4	4	Battery is not upto the mark. Battery reduce v...	3.0

Here,

- 1 = Poor.
- 2 = Fair.
- 3 = Good.
- 4 = Very Good.
- 5 = Excellent.

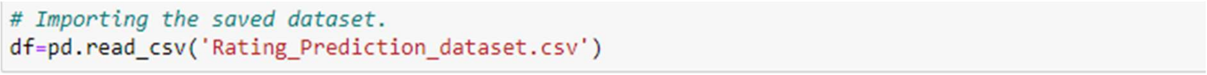
Importing the necessary libraries and packages

First we have imported the necessary libraries.

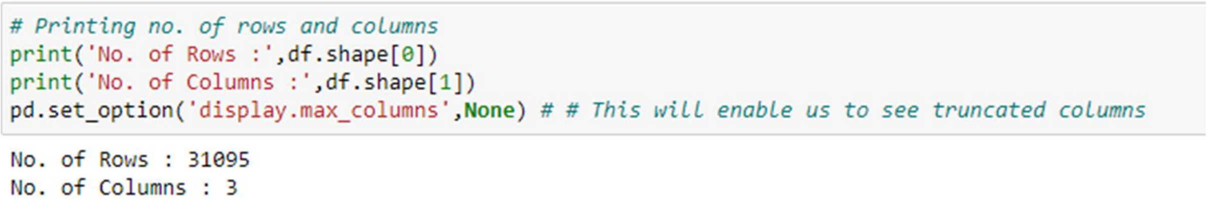


```
#Importing necessary libraries and packages
import pandas as pd
import numpy as np

import seaborn as sns                 # For Visualization
import matplotlib.pyplot as plt       # ploting package
%matplotlib inline
import matplotlib.ticker as plticker


import warnings                       # Filtering warnings
warnings.filterwarnings('ignore')
```

Then we have imported our dataset which was in CSV format and printed the shape of the dataset, i.e., the total rows and columns.



```
# Importing the saved dataset.
df=pd.read_csv('Rating_Prediction_dataset.csv')
```

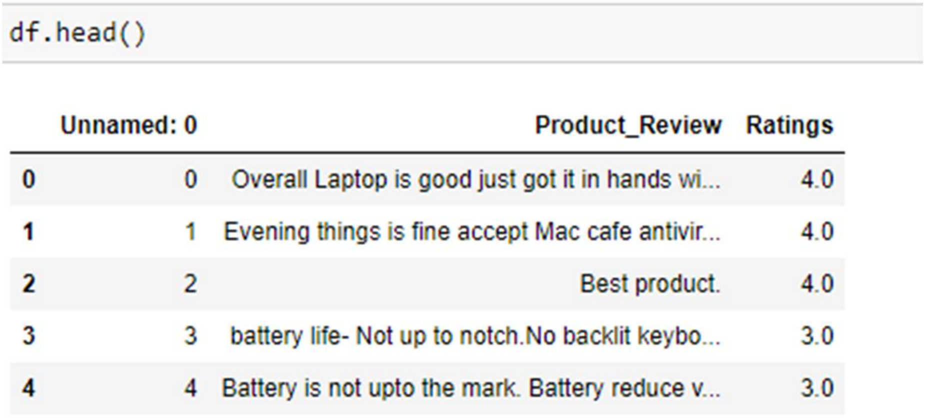


```
# Printing no. of rows and columns
print('No. of Rows :',df.shape[0])
print('No. of Columns :',df.shape[1])
pd.set_option('display.max_columns',None) # # This will enable us to see truncated columns
```

```
No. of Rows : 31095
No. of Columns : 3
```

Exploratory Data Analysis (EDA)

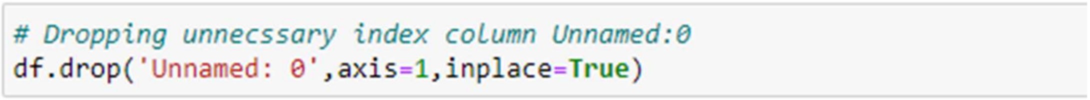
Next, we have printed the head to get a general understanding of the data and their values.



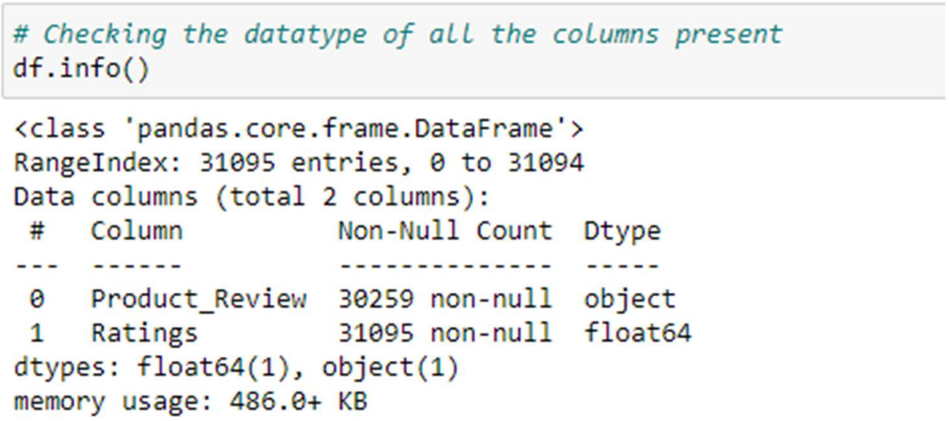
```
df.head()
```

	Unnamed: 0	Product_Review	Ratings
0	0	Overall Laptop is good just got it in hands wi...	4.0
1	1	Evening things is fine accept Mac cafe antivir...	4.0
2	2	Best product.	4.0
3	3	battery life- Not up to notch.No backlit keybo...	3.0
4	4	Battery is not upto the mark. Battery reduce v...	3.0

We dropped the ‘Unnamed: 0’ column because it was unnecessary for our prediction.



```
# Dropping unnecssary index column Unnamed:0
df.drop('Unnamed: 0',axis=1,inplace=True)
```

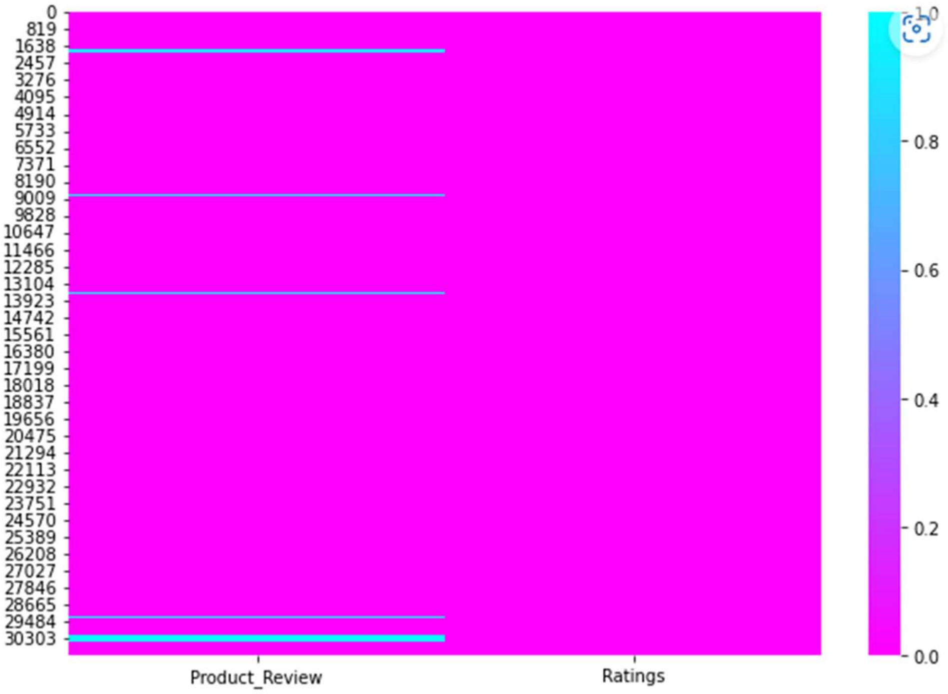


```
# Checking the datatype of all the columns present
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31095 entries, 0 to 31094
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Product_Review  30259 non-null  object 
 1   Ratings         31095 non-null  float64
dtypes: float64(1), object(1)
memory usage: 486.0+ KB
```

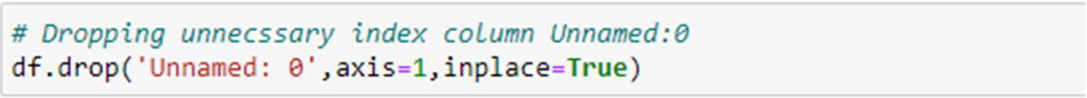
We observe that the ‘Product_Review’ column is an object datatype & the Ratings column is a float datatype.

It also came to our notice that there were certain null values in the Product review column. We illustrated the same with a heatmap.



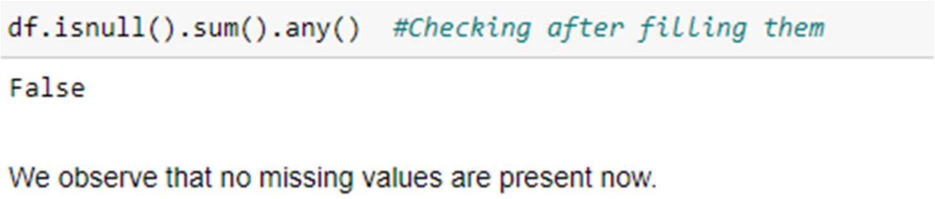
Data pre-processing

We dropped the 'Unnamed: 0' column because it was unnecessary for our prediction.



```
# Dropping unnecssary index column Unnamed:0
df.drop('Unnamed: 0',axis=1,inplace=True)
```

We also observed that this column had some null values which we replaced by inserting 'Reviews not available'. Thus, the null values were eliminated.



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

```
False
```

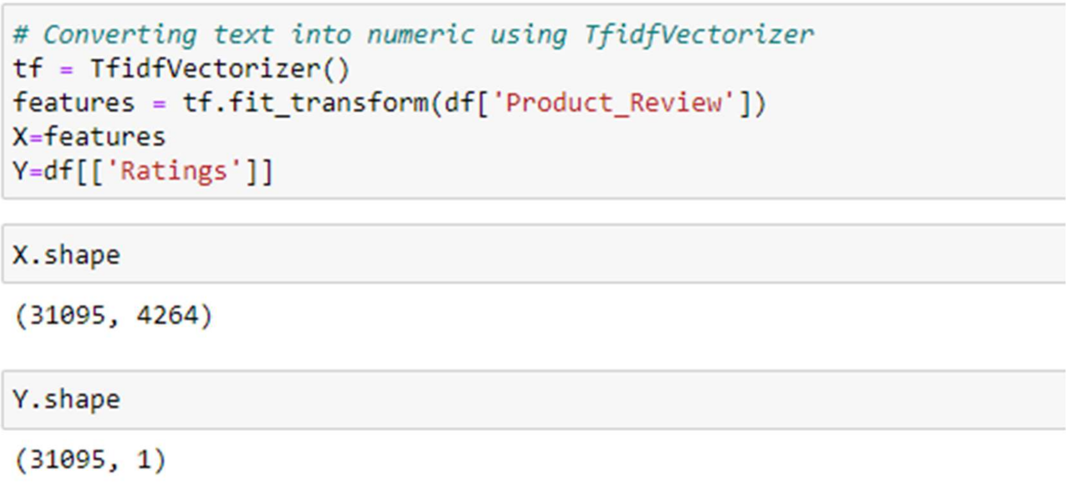
We observe that no missing values are present now.

We also applied the following steps as a part of text mining:

- Removing Punctuations and other special characters
- Word Tokenzation
- Removing Stop Words
- Stemming and Lemmatising
- Applying Count Vectorizer

Feature and Target Value

We are now ready to prepare our data for model building. Let's start with separating the target value (in y) from the feature variables (in x).



```
# Converting text into numeric using TfidfVectorizer
tf = TfidfVectorizer()
features = tf.fit_transform(df['Product_Review'])
X=features
Y=df[['Ratings']]
```

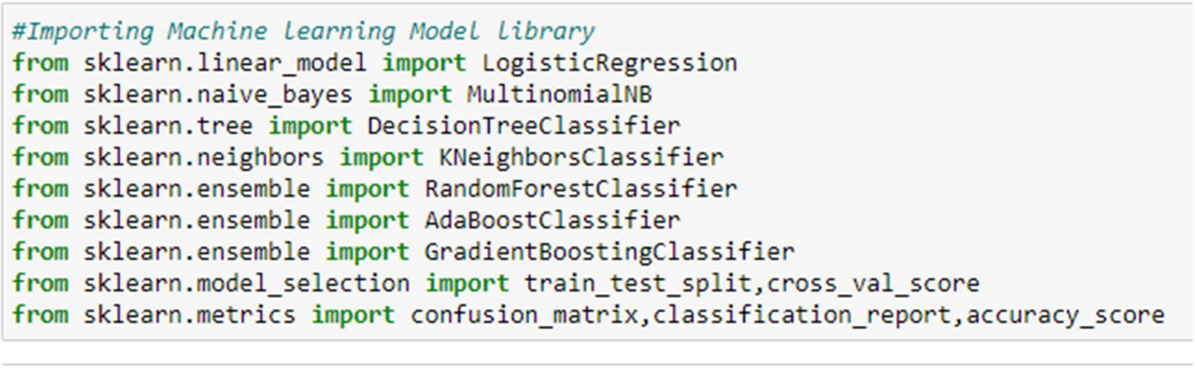
```
X.shape
```

```
(31095, 4264)
```

```
Y.shape
```

```
(31095, 1)
```

To build the model we imported the necessary packages that enable training and testing.



```
#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
```

- **Hardware and Software Requirements and Tools Used**

Hardware required:

a. Processor: core i5 or above
b. RAM: 8 GB or above
c. ROM/SSD: 250 GB or above

Software required:

d. Anaconda 3- language used Python 3
e. Microsoft Excel Libraries: The important libraries that I have used for this project are below:

import numpy as np

It is defined as a Python package used for performing various numerical computations and processing of the multidimensional and single dimensional array elements. The calculations using Numpy arrays are faster than the normal Python array.

import pandas as pd

Pandas is a Python library that is used for faster data analysis, data cleaning and data pre-processing. The data-frame term is coming from Pandas only.

import matplotlib.pyplot as plt and import seaborn as sns

Matplotlib and Seaborn acts as the backbone of data visualization through Python.

Matplotlib: It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statical interferences and plotting 2D graphs of arrays.

Seaborn: It is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy. It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data.

- **Data Inputs- Logic- Output Relationships**

We are here considering the comments from users as the input data and evaluating their responses on the basis of the words they have used. The output indicates the level of harshness each comment carries categorised as highly malignant, malignant, rude, abuse, threat and loathe.

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

a. We used ".drop()" function to drop unwanted entries in the columns.
b. Described the datatypes using ".info()" method.
c. To check null values we have used ".isnull().sum().any()".
d. Removed Punctuations and other special characters
e. Used Word Tokenzation
f. Removed Stop Words
g. Applied Stemming and Lemmatising
h. Applied Count Vectorizer

i. Used WordCloud to visualize the most and the least used words in each type of comment.

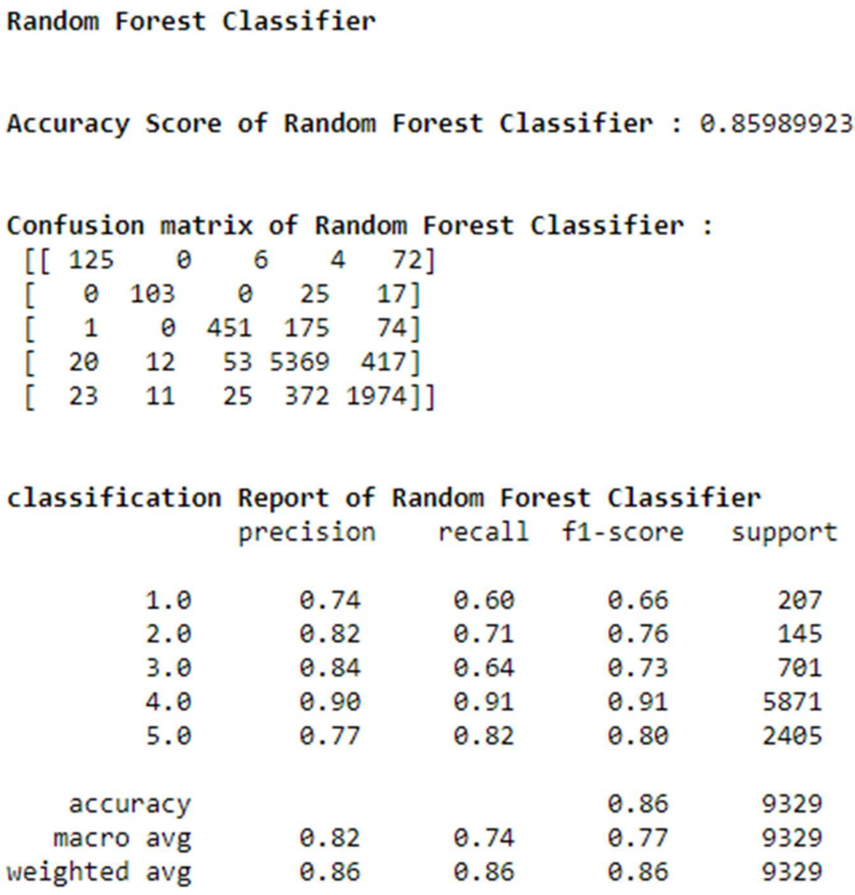
Testing of Identified Approaches (Algorithms)

We tested the data on the following models:

Models	Accuracy Score
Logistic Regression	= 85.31%
Decision Tree Classifier	= 85.96%
Random Forest Classifier	= 85.98%
Ada Boost Classifier	= 63.44%
Gradient Boosting Classifier	= 83.80%

Run and Evaluate selected models

Thereafter, we found that Random Forest Classifier performs better with Accuracy Score: 85.98992389323614 % than the other classification models.

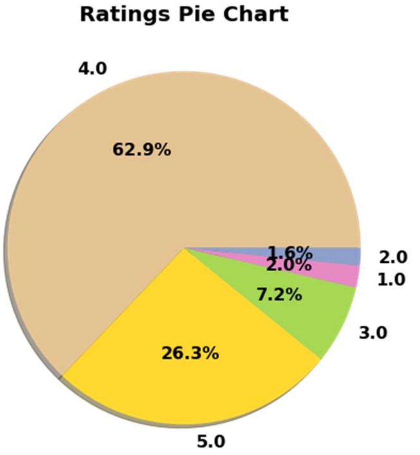


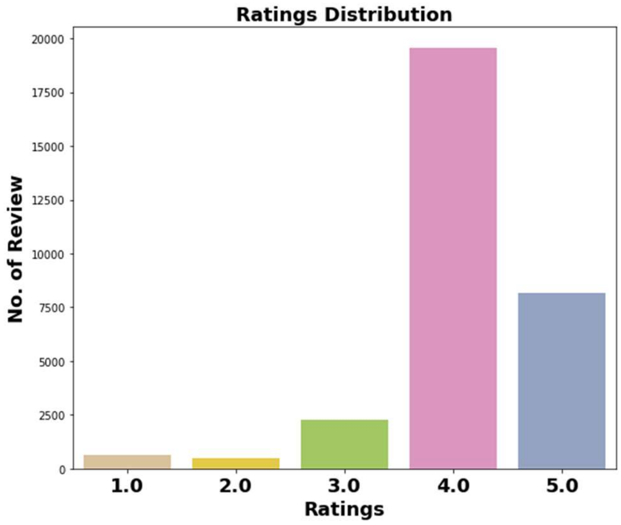
```
Random Forest Classifier


Accuracy Score of Random Forest Classifier : 0.8598992389323614


Confusion matrix of Random Forest Classifier :
 [[ 125    0    6    4   72]
 [    0  103    0   25   17]
 [    1    0  451  175   74]
 [   20   12   53 5369  417]
 [   23   11   25  372 1974]]


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

         1.0       0.74      0.60      0.66       207
         2.0       0.82      0.71      0.76       145
         3.0       0.84      0.64      0.73       701
         4.0       0.90      0.91      0.91      5871
         5.0       0.77      0.82      0.80      2405

    accuracy                           0.86      9329
   macro avg       0.82      0.74      0.77      9329
weighted avg       0.86      0.86      0.86      9329
```

Therefore, we considered Random Forest Classifier for further Hyperparameter tuning which further enhanced the accuracy to 86.11%.

Visualizations

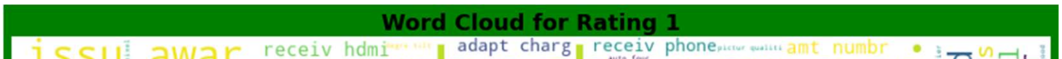
We have visualized the rating data in a count plot and pie plot:



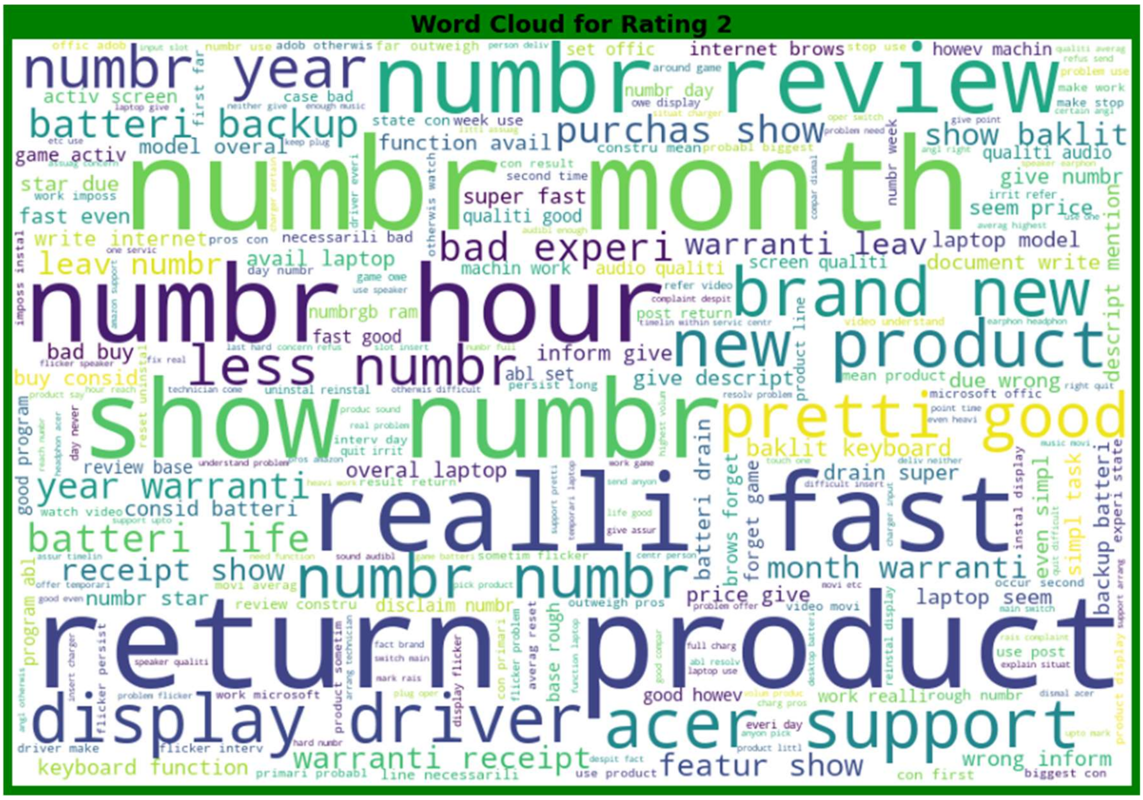


We can see that Rating 4 has the highest count followed by Rating 5, 3, 1, 2.

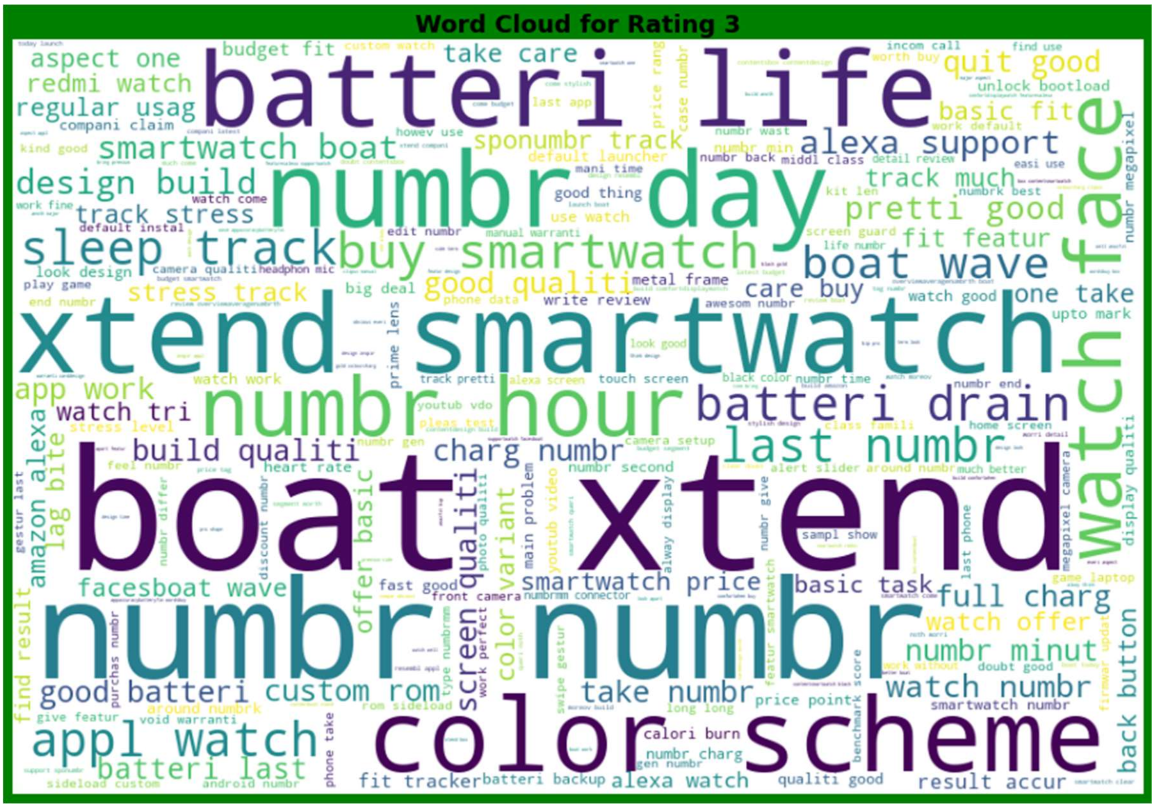
Here are some word clouds we created with each of the review categories to check the words that categorised them into the particular type:



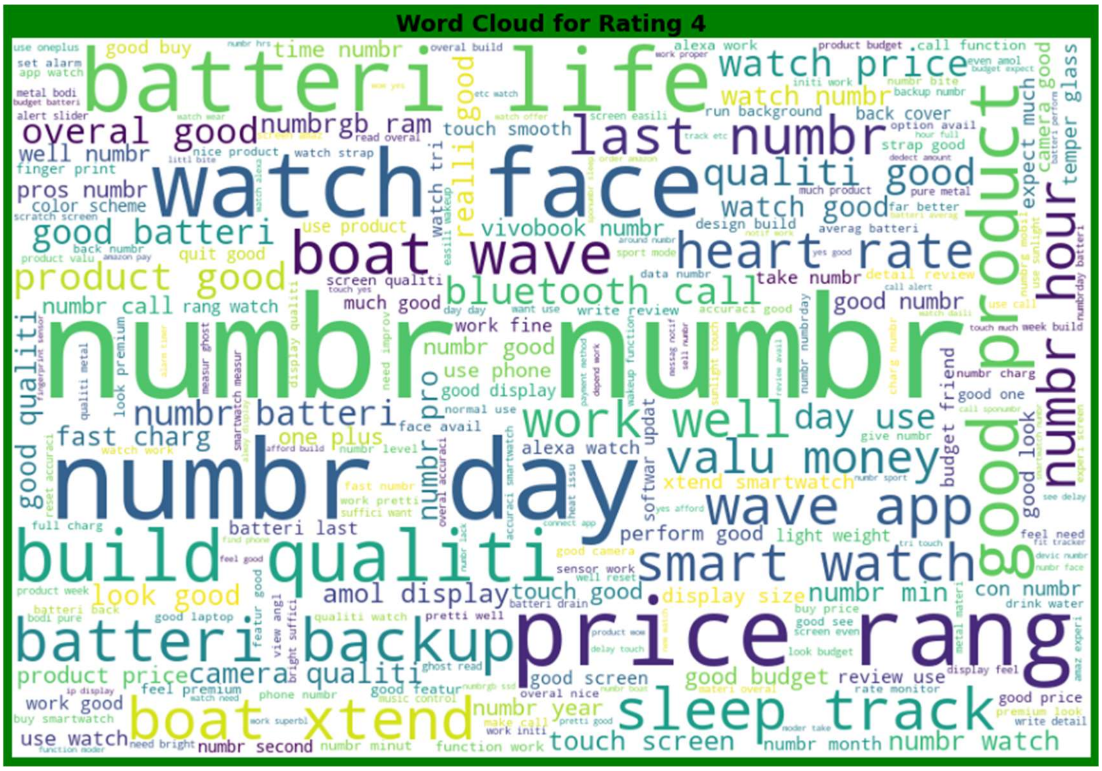
We observe for reviews with Rating 1- the most used words are numbr, memori, card, nikon, etc.



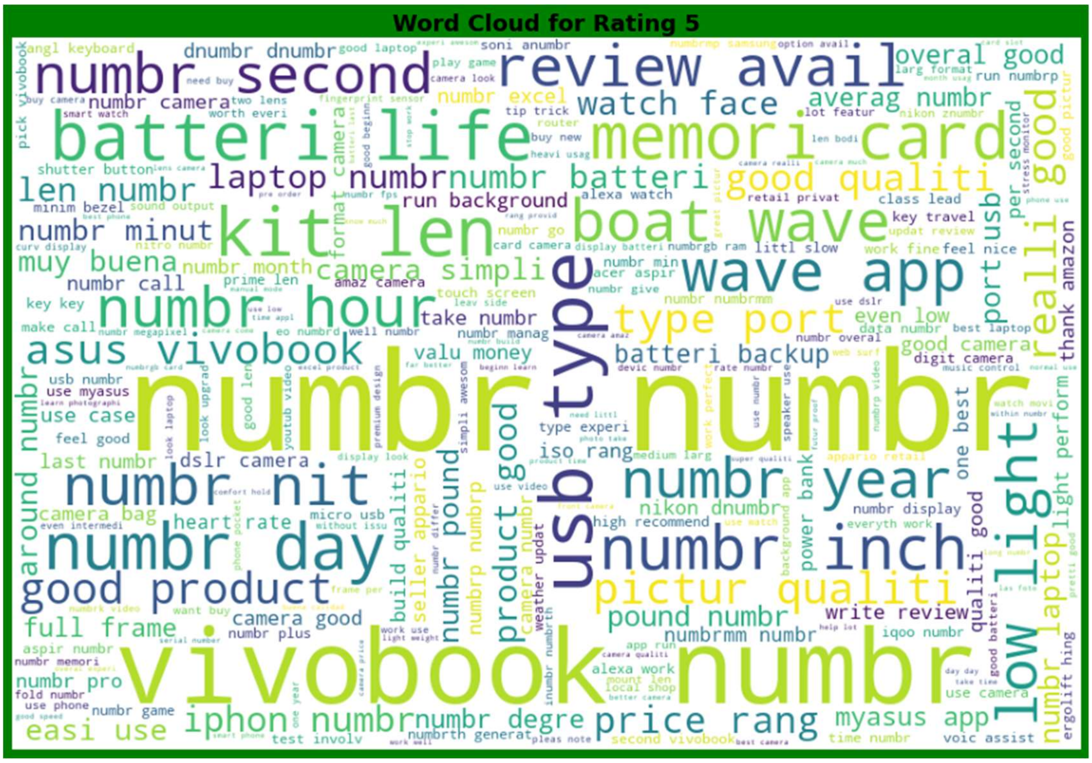
We observe for reviews with Rating 2- the most used words are return, product, numbr, realli, month, etc.



We observe for reviews with Rating 3- the most used words are boat, xtend, batteri, life, color, scheme, etc.



We observe for reviews with Rating 4- the most used words are numbr, day, price, rang, batteri life, etc.



We observe for reviews with Rating 5- the most used words are numbr, vivobook, usb type, etc.

Interpretation of the Results

- The data is imbalanced.
- The dataset reveals that most of the people who reviewed have given 4-star rating.
- After preprocessing we are able to build models for testing.
- We achieved 85.98%.accuracy and hence we can say that with advanced techniques the results can be more accurate.

Conclusion

- Random Forest Classifier performs better with Accuracy Score: 85.98992389323614%.
- Final Model (Hyperparameter Tuning) is giving us Accuracy score of 86.11% which is slightly improved compare to earlier Accuracy score of 85.98%.
- Here Random forest classifier is the most accurate algorithm as compared to others.

Limitations of work and Scope for Future Work

- The data is imbalanced but we couldn't apply balancing techniques due to computational limitations.
- Deep learning CNN, ANN can be used to build more accurate models.