FLIP ROBO

RATINGS PREDICTION

Submitted by:

Ibrahim Abdul Shukoor

Internship – 30

ACKNOWLEDGMENT

I want to express my gratitude to my mentor, Mr. Kashif Mohd, who always supported me and helped me develop my talents. I want to thank Flip Robo Technologies for providing me with a platform to learn, comprehend, and carry out various projects like this one. Finally, I would want to express my sincere gratitude to Data Trained for teaching me cutting-edge Python, Statistics, and machine learning approaches.

INTRODUCTION

A customer has a website where users can post various product reviews for technical items. They're including a new function onto their website called. The reviewer must also include their rating in the form of stars. There are only 5 alternatives available, and the ranking is out of 5. 1, 2, 3, 4, and 5 stars, respectively. They are attempting to forecast ratings for past reviews that have not yet received one. Therefore, we must create a program that can gauge the rating from the review.

I. Business Problem Framing

The objective of the project is to predict the rating of the review left by a customer.

Given a group of sentences or paragraphs, used as a review by a user in an online platform, must be classified into 1 star, 2 star, 3 stars, 3 stars or 5 stars with 5 stars being the highest and 1 star – the lowest.

II. Conceptual Background of the Domain Problem

Following are the study's primary goals:

a. To scrape the data from various websites
b. To clean the data scraped (if applicable)
c. To visualize different traits that can have an impact on the target variable.
d. To create machine learning models that can predict the ratings
e. To select the ideal model and use it with the test data.

III. Review of Literature

Machine Learning: With the use of machine learning (ML), which is a form of artificial intelligence (AI), software programs can predict outcomes more accurately without having to be explicitly instructed to do so. In order to forecast new output values, machine learning algorithms use historical data as input. The kind of data that data scientists wish to predict determines the kind of algorithm they use.

The three categories of machine learning algorithms are:

i. Supervised Learning: In supervised learning, data scientists describe the variables they want the algorithm to look for connections between and provide the algorithms with labelled training data. The algorithm's input and output are both described.

ii. Unsupervised learning: Algorithms trained on unlabelled data are used in this sort of machine learning. The algorithm searches through data sets in search of any significant relationships. Both the input data that algorithms use to train and the predictions or suggestions they produce are predefined.

iii. Data scientists generally employ reinforcement learning to instruct a computer to carry out a multi-step procedure for which there are set rules. An algorithm is programmed by data scientists to fulfil a goal, and they provide it with positive or negative feedback as it determines how to do so. However, the algorithm typically chooses the course of action on its own.

IV. Motivation for the Problem Undertaken

Every product used by a customer will leave a sense of satisfaction/ dissatisfaction. Customers tend to express this feeling in the reviews section of a product page which can be seen by 1000+ users a day. If this program could help provide a star-based rating system to the review let by a user, it could simplify the process of classifying the product as good/bad.

Analytical Problem Framing

I. Mathematical/ Analytical Modelling of the Problem

There are several methods for data analysis, but the two that are used most frequently are as follows:

- Supervised learning, which includes classification and regression models.
- Unsupervised learning, which includes association rules and clustering methods

Classification Model: Classification models are used to look at how different variables relate to one another. When determining which independent factors have the most impact on dependent variables, classification models are frequently utilized to gather crucial information.

II. Data Sources and their formats

The data has been scraped from Amazon and Flipkart.

Format: Comma Separated Values (CSV).

Data Types:

- int64 – 1.
- object – 2

III. Data Pre-processing Done

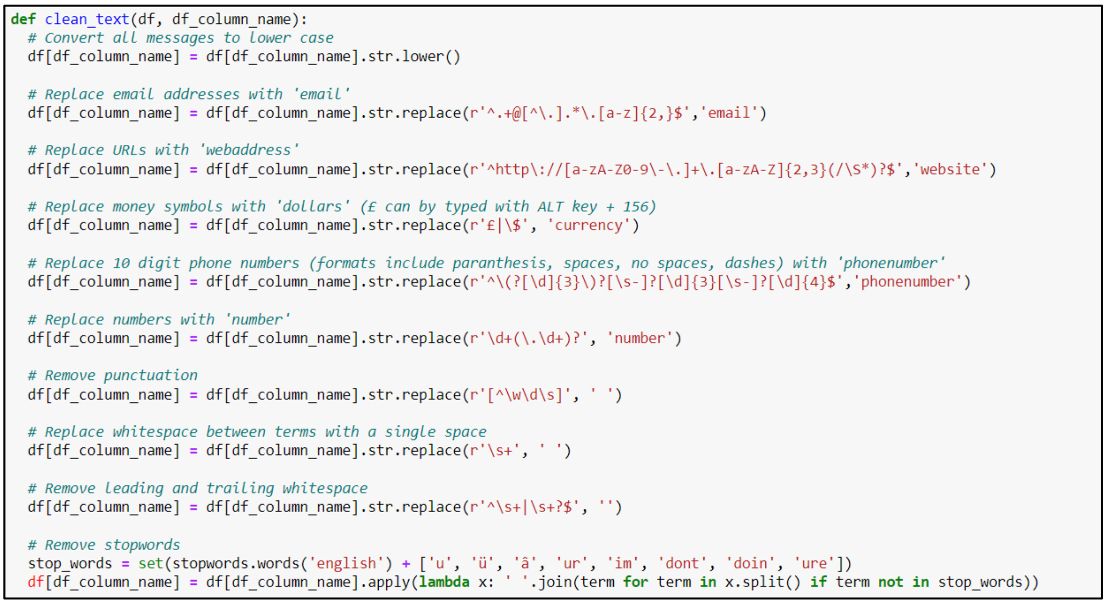
The following actions were taken during the data pre-processing:

i. The "Ratings" column was cleaned to only include the rating (For example: 1,2,3,4,5)

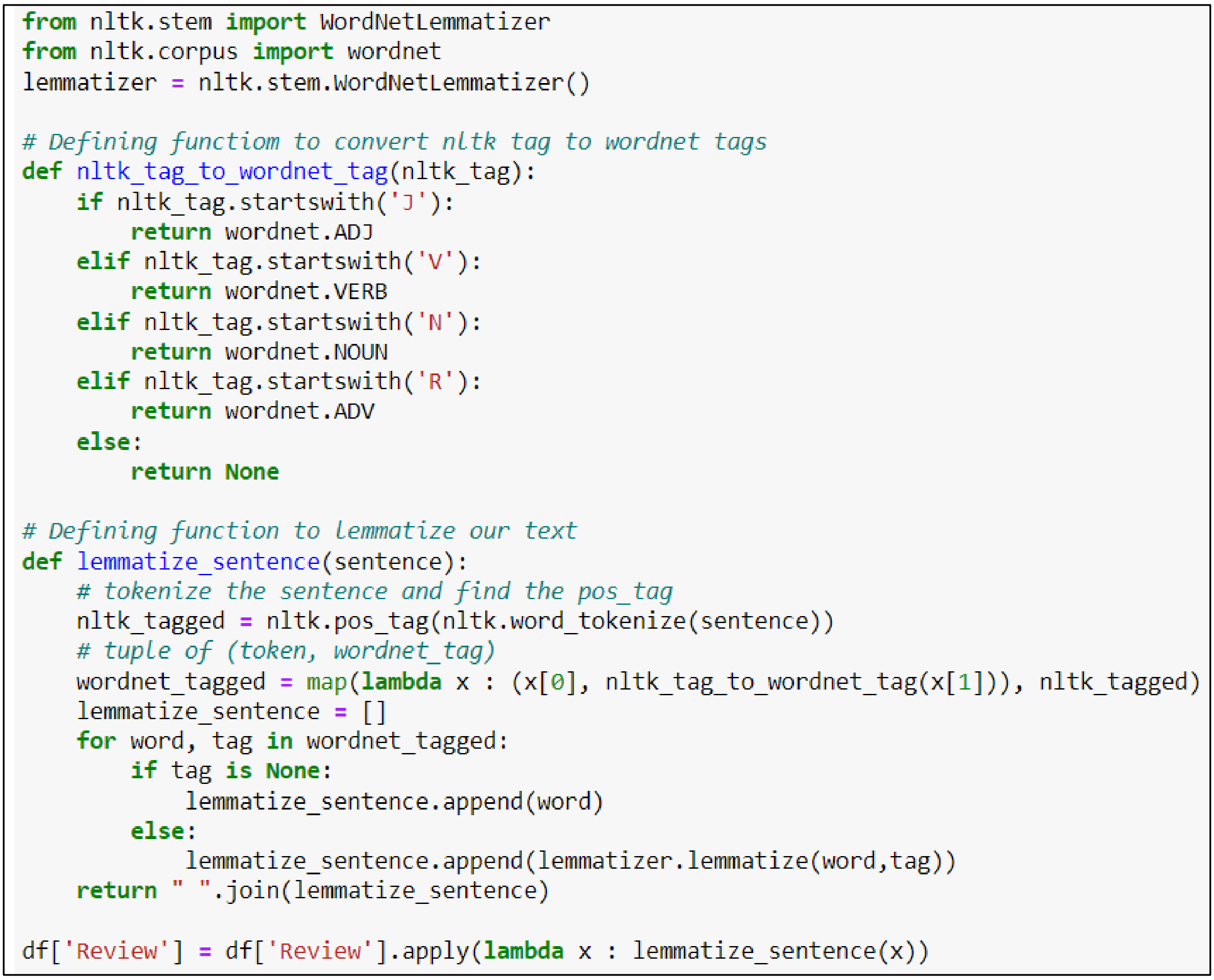
ii. Created columns called "Review_count" and "Review_char", which contains the number of words a review contains in total and the number of characters in a review in total respectively.

iii. Comments were cleaned:

   a. Emails were replaced with "email"
   b. Websites were replaced with "website"
   c. Currencies were replaced with "currency"
   d. Phone numbers were replaced with "phonenumber"
   e. Cleaned additional things such as trailing, leading white spaces, removed stop words.



```python
def clean_text(df, df_column_name):
  # Convert all messages to lower case
  df[df_column_name] = df[df_column_name].str.lower()

  # Replace email addresses with 'email'
  df[df_column_name] = df[df_column_name].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$','email')

  # Replace URLs with 'webaddress'
  df[df_column_name] = df[df_column_name].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$','website')

  # Replace money symbols with 'dollars' (£ can by typed with ALT key + 156)
  df[df_column_name] = df[df_column_name].str.replace(r'£|\$', 'currency')

  # Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'
  df[df_column_name] = df[df_column_name].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$','phonenumber')

  # Replace numbers with 'number'
  df[df_column_name] = df[df_column_name].str.replace(r'\d+(\.\d+)?', 'number')

  # Remove punctuation
  df[df_column_name] = df[df_column_name].str.replace(r'[^\w\d\s]', ' ')

  # Replace whitespace between terms with a single space
  df[df_column_name] = df[df_column_name].str.replace(r'\s+', ' ')

  # Remove leading and trailing whitespace
  df[df_column_name] = df[df_column_name].str.replace(r'^\s+|\s+?$', '')

  # Remove stopwords
  stop_words = set(stopwords.words('english') + ['u', 'ü', 'â', 'ur', 'im', 'dont', 'doin', 'ure'])
  df[df_column_name] = df[df_column_name].apply(lambda x: ' '.join(term for term in x.split() if term not in stop_words))
```



```python
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet
lemmatizer = nltk.stem.WordNetLemmatizer()

# Defining functiom to convert nltk tag to wordnet tags
def nltk_tag_to_wordnet_tag(nltk_tag):
    if nltk_tag.startswith('J'):
        return wordnet.ADJ
    elif nltk_tag.startswith('V'):
        return wordnet.VERB
    elif nltk_tag.startswith('N'):
        return wordnet.NOUN
    elif nltk_tag.startswith('R'):
        return wordnet.ADV
    else:
        return None

# Defining function to lemmatize our text
def lemmatize_sentence(sentence):
    # tokenize the sentence and find the pos_tag
    nltk_tagged = nltk.pos_tag(nltk.word_tokenize(sentence))
    # tuple of (token, wordnet_tag)
    wordnet_tagged = map(lambda x : (x[0], nltk_tag_to_wordnet_tag(x[1])), nltk_tagged)
    lemmatize_sentence = []
    for word, tag in wordnet_tagged:
        if tag is None:
            lemmatize_sentence.append(word)
        else:
            lemmatize_sentence.append(lemmatizer.lemmatize(word,tag))
    return " ".join(lemmatize_sentence)

df['Review'] = df['Review'].apply(lambda x : lemmatize_sentence(x))
```

Figure 1: Data Pre-processing

I. Hardware/Software and Tools

i. Jupyter Notebook
ii. NumPy
iii. Pandas
iv. Matplotlib
v. Seaborn
vi. nltk
vii. string

Model/s Development and Evaluation

I. Testing of Identified Approaches (Algorithms)

Five algorithms altogether were executed.

i. Logistic Regression
ii. Linear Support Vector Classifier
iii. Bernoulli Naive Bayes
iv. Multinomial Naive Bayes
v. SGD Classifier
vi. LGBM Classifier

II. Run and evaluate selected models

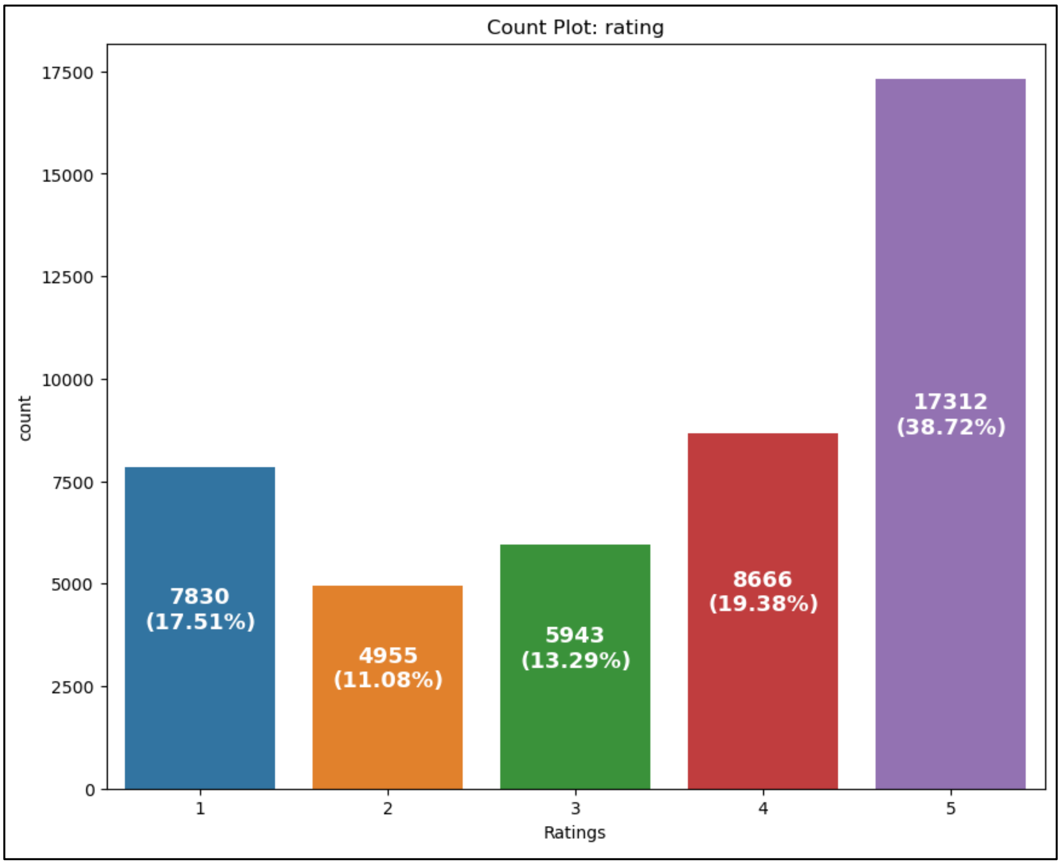
vii. Logistic Regression
viii. Linear Support Vector Classifier **(BEST SCORE)**
ix. Bernoulli Naive Bayes
x. Multinomial Naive Bayes
xi. SGD Classifier

III. Key Metrics for success in solving problem under consideration

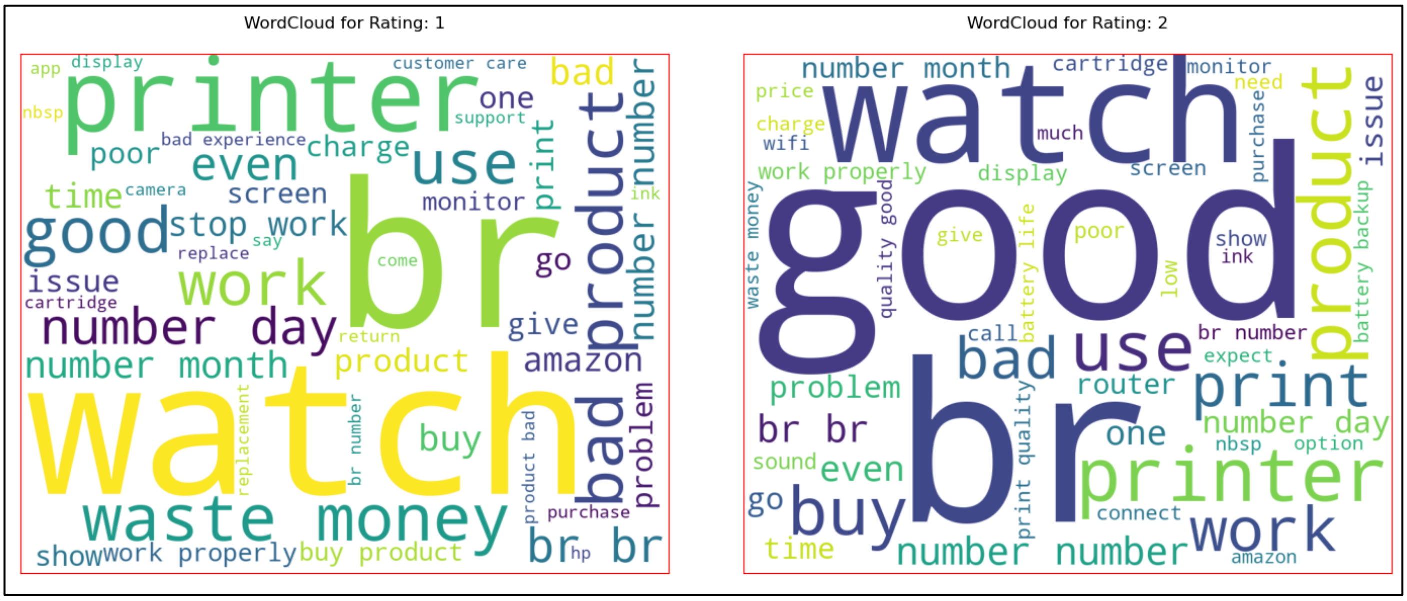
The measures utilized to assess the model's effectiveness included the F1 score, CV, accuracy and confusion matrix.

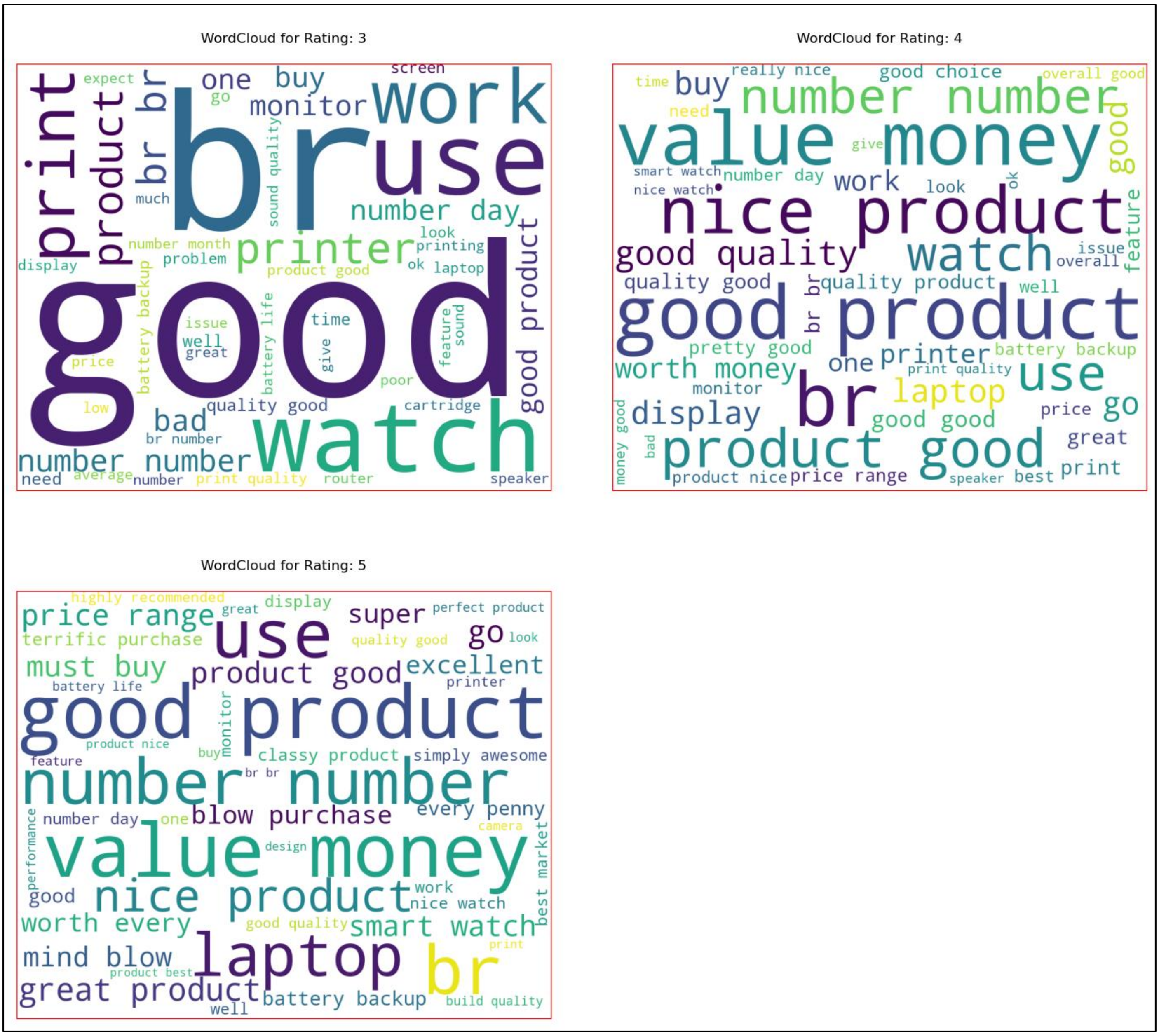
IV. Visualizations

i. Plotting the Ratings and the count

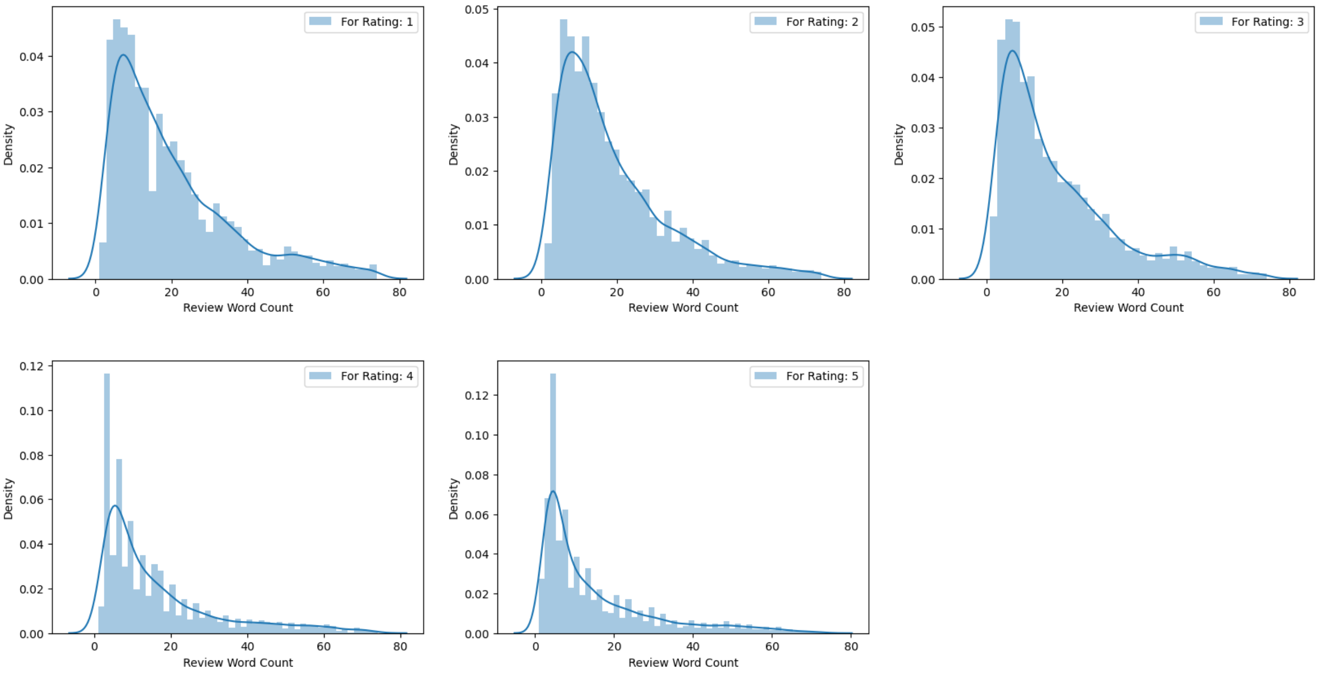


ii. Plotted the word clouds of the Reviews based on Ratings.





iii. Plotting the word counts based on the ratings



V. Interpretation of the Results

a. Looking at the count plot above for our target variable (Ratings), we can see that the majority of the data set's reviews are rated as 5 stars, while the number of reviews rated as 2 stars is extremely low. It will lead to an imbalance problem and bias in our machine learning model.

b. **Rating 1:** It generally contains words like "watch," "use," "poor product," "waste of time and money," "problem," and "issue."

c. **Rating 2:** The phrases good, phone, use, watch, poor, issue, waste money, quality good, bad, problem, etc. are the most common ones.

d. **Rating 3:** Words like sound quality, good, usage, time, camera quality, display, buy, build quality, etc. are the most common ones.

e. **Rating 4:** Words like "use," "purchase," "phone," "watch," "good product," "good quality," "good choice," and "lovely product" are frequently used.

f. **Rating 5:** Words like "price range," "value for money," "good product," "well, go," "absolutely wonderful," "great product," etc. are frequently used.

g. According to the word count plot, consumers are more likely to write descriptive reviews when they are unhappy with a service than they are to do so when they are pleased.

CONCLUSION

I. Learning Outcomes of the Study in respect of Data Science and Limitations of this work and Scope for Future Work

a. Further Hyperparameter tuning can be performed to increase accuracy.

b. Better pre-processing can be performed to increase the efficiency.