RATING PREDICTION PROJECT

Submitted By :

Milind B. Kuwar

ACKNOWLEDGMENT

The project would not have been built without the constant support from DataTrained and Fliprobo teams.

References:

1. Notes and classes by DataTrained Academy

INTRODUCTION

- ## Business Problem Framing

The prediction for rating for a particular review on any e-commerce site can help their listers/makers of those products to find the problems or positives that a customer is facing after buying this product. This can help these e-commerce sites to grow or retain theirbusinesses. Here, we have built a ML model that processes Review posted on Amazon using common NLP techniques. This task is similar to Sentiment Analysis, but instead of predicting the positive and negative sentiment (sometimes neutral also), here I need to predict the star rating

- ## Conceptual Background of the Domain Problem

Sentiment Analysis is the most commonly used approach to analyze data which is in the form of text and to identify sentiment content from the text. Opinion Mining is another name for sentiment analysis. A wide range of text data is getting generated in the form of suggestions, feedback's, tweets and comments. E-Commerce portals are generating a lot of data every day in the form of customer reviews. Analyzing E Commerce data will help online retailers to understand customer expectations, provide better shopping experience and to increase the sales. Sentiment Analysis can be used to identify positive, negative and neutral information from the customer reviews. Researchers have developed a lot of techniques in Sentiment Analysis. Mostly sentiment Analysis is done using a single machine learning algorithm. This work uses Amazon customer review data and focuses on finding aspect terms from each review, identifying the Parts-of-Speech, applying classification algorithms to find the score of positivity, negativity and neutrality of each review.

- ## Review of Literature

The advent of electronic commerce with growth in internet and network technologies has led customers to move to online retail platforms such as Amazon, Walmart, Flip Kart, etc. People often rely on customer reviews of products before they buy online. These reviews are often rich in information describing the product. Customers often choose to compare between various products and brands based on whether an item has a positive or negative review. More often, these reviews act as a feedback mechanism for the seller. Through this medium, sellers strategize their future sales and product improvement.

There is a client who has a website where people write different reviews for technical products. Now they want to add 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.

This project is more about exploration, feature engineering and classification that can be done on this data. Since we scrape huge amount of data that includes five stars rating, we can do better data exploration and derive some interesting features using the available columns. We can categorize the ratings as: 1.0, 2.0, 3.0, 4.0 and 5.0 stars

The goal of this project is to build an application which can predict the rating by seeing the review. In the long term, this would allow people to better explain and reviewing their purchase with each other in this increasingly digital world.

- ## Motivation for the Problem Undertaken

The motivation that driven me towards the timeline of this project was the sheer size and complexity of this dataset. I have tried my best to follow best practices of machine learning throughout the project and follow the steps that doesn't only predict the best results but also tried to keep the process simple to understand.

Analytical Problem Framing

- ## Analytical Modelling of the Problem

Mathematical Tools Used: Mean, Average, IQR, Standard deviation and Median for gaining insights of the dataset. Analytical Tools Used: Correlation and Skewness for finding the relationships of dependent and independent variable and checking the distribution of data. Packages Used: scikit-learn, pandas, seaborn etc

The following steps represents stepwise tasks that need to be completed:

- Data Collection
- Exploratory Data Analysis
- Data pre-processing
- Data Visualization
- Model Building
- Model Evaluation
- Saving the best model

- ## Data Sources and their formats

Data Sources and their formats Data was been scraped by me using Selenium and then making csv file of scraped data and using it in analysis. Whereas originally data belongs Amazon.in. The data contained 2 columns and 29535 entries which are shown below :

Data Types : int64

```
# Let's enter the details in the search column
p_search=input("Enter product you want to search: ")
sear= driver.find_element_by_id("twotabsearchtextbox")
sear.send_keys(p_search)
sea_but=driver.find_element_by_id("nav-search-submit-button")
sea_but.click()
```

```
Enter product you want to search: Laptop
```

```
rating = []
review = []
```

```
#Searching for laptops
#Functions for getting products in all 2000 entries
urls=[]
for k in range(17):
    for i in driver.find_elements_by_xpath("//div[@class='a-section a-spacing-medium']//h2"):
        urls.append(i.find_element_by_xpath(".//a").get_attribute('href'))
    try:
        next_page=driver.find_element_by_xpath("//div[@class='a-text-center']/ul/li[@class='a-last']/a").get_attribute('href')
        driver.get(next_page)
        driver.refresh()
    except:
        break

#Taking every URL one after one
for i in urls:
    driver.get(i)
    urls.append(i)
    driver.implicitly_wait(5)

#Scraping the rating
    try:
        driver.find_element_by_xpath("//a[1][@id='acrCustomerReviewLink']").click()
    except NoSuchElementException:
        print("No rating")
        pass

#Scraping all the reviews
    try:
        driver.find_element_by_xpath("//a[@class='a-link-emphasis a-text-bold']").click()
    except NoSuchElementException:
        pass


#Scrapping the details
    S_page=1
    E_page=60
    for page in range(S_page,E_page+1):
        try:
            reviews=driver.find_elements_by_xpath("//div[@class='a-row a-spacing-small review-data']/span/span")
            for r in reviews:
                review.append(r.text.replace('\n',''))
        except NoSuchElementException:
            review.append("Not Available")
        try:
            Rating=driver.find_elements_by_xpath("//div[@class='a-section celwidget']/div[2]/a[1]")
            for i in Rating:
                ratings=i.get_attribute('title')
                rating.append(ratings[:3])
        except NoSuchElementException:
            rating.append("Not available")

        #Printing the page scrapped
        print("Product review and rating  of page {} scraped ".format(page+1))

        #Looping for going to next page automatically
        try:
            next_page=driver.find_element_by_xpath("//div[@id='cm_cr-pagination_bar']/ul/li[2]/a")
            if next_page.text=='Next Page':
                next_page.click()
                time.sleep(2)
        except NoSuchElementException:
            pass
```

```
rating_data = pd.concat([Lapdf,Watchdf])
rating_data.head()
```

	Unnamed: 0	Reviews:	Ratings:
0	0	Was excited for this laptop until it got deliv...	1.0
1	1	As with the looks its very sleek, lightweight ...	5.0
2	2	powerful machine but a bit heavy ..	5.0
3	3	its is a perfect banlance of hardware and pric...	5.0
4	4	Updated review after 6months of usage:All the ...	5.0

```
rating_data.shape
```

```
(29535, 3)
```

```
#Saving the data into a csv file
rating_data.to_csv('Rating_Data.csv')
```

- For keeping only, the useful data we used feature engineering. We removed the data if unnecessary by qualifying them on certain conditions like their uniqueness, their correlation with target variable and the no of outliers present in that particular variable. - We then tried to treat the missing values. - After handling the missing values pre-processing was done using NLP techniques like Regex, Tokenization, Lemmatization and removing all the unnecessary stop words.

Data Preprocessing

- ## Data Preprocessing Done

  - Checked for null values and identified no null values in dataset.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29535 entries, 0 to 29534
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Reviews:   29535 non-null  object 
 1   Ratings:   29535 non-null  float64
dtypes: float64(1), object(1)
memory usage: 461.6+ KB
```

```
# We will find the missing values first
df.isnull().sum()
```

```
Reviews:     0
Ratings:     0
dtype: int64
```

```
# Unique values present in Ratings: column
df['Ratings: '].value_counts()
```

```
5.0    13287
1.0    10434
4.0     3940
2.0     1500
3.0      374
Name: Ratings: , dtype: int64
```

- Converting all message to lower case
- Replace email address with 'email'
- Replace money symbol with 'moneysymb'(£ can type with ALT key+156)
- Replace 10 digit phone number(format include paranthesis, space, no spaces,dashes) with 'phone number'
- Remove punctuation
- Remove leading and trailing whitespace
- Remove leading and trailing whitespace
- Replace whitespace between terms with a single space
- Remove stopwords

```
def clean_text(df, df_column_name):

    # Converting all messages to lowercase
    df[df_column_name] = df[df_column_name].str.lower()

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

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

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

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

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

    # 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', '4', '2', '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))
```

After performing all the above steps and also adding a new feature to check new comment length after cleaning, our dataset would look as follows:

```
# Calling the class
clean_text(df, 'Reviews: ')
df['Reviews: '].head(3)
```

```
0    excited laptop got delivered today turns huge ...
1    looks sleek lightweight colour dark grey gives...
2                          powerful machine bit heavy
Name: Reviews: , dtype: object
```

```
# Using RegexpTokenizer for tokenizing the data
from nltk.tokenize import RegexpTokenizer
tokenizer=RegexpTokenizer(r'\w+')
df['Reviews: '] = df['Reviews: '].apply(lambda x: tokenizer.tokenize(x.lower()))
df.head()
```

	Reviews:	Ratings:
0	[excited, laptop, got, delivered, today, turns...	1.0
1	[looks, sleek, lightweight, colour, dark, grey...	5.0
2	[powerful, machine, bit, heavy]	5.0
3	[perfect, banlance, hardware, price, post, ful...	5.0
4	[updated, review, numbrmonths, usage, points, ...	5.0

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

  Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

2. Lenovo ideapad145 laptop
3. Jupyter Notebook - The Jupyter Notebook is an interactive environment for running code in the browser. It is a great tool for exploratory data analysis and is widely used by data scientists.
4. MS powerpoint - For preparing presentation of project.
5. MS word - For preparing report
6. Matplotlib - For data visualization to produce high quality plots, charts and graphs.
7. Pandas - Python library for data wrangling and analysis. It is built around a data structure called DataFrame.
8. Numpy - NumPy is one of the fundamental packages for scientific computing in Python.

Model/s Development and Evaluation

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

  I have use statistical models to find insights given set of data. Predictions are assessed differently depending on its type: Based on the data structure of Target variable, Model type was identified

  First and important step was to identify the model which was done by analysing the target variable . Best Acuuracy score was achieved using Logistic Regression

- ## Testing of Identified Approaches (Algorithms)

  Build Models based on learning it was a supervised classification problem.
  I built 5 models to evaluate performance of each of them:

  1. **Logistic Regression**
  2. **MultinomialNB**
  3. **KNeighbors Classifier**
  4. **DecisionTree Classifier**
  5. **AdaBoost Classifier**
  6. **RandomForest Classifer**
  7. **GradientBoosting Classsifer**

- **Run and Evaluate selected models**

  Describing all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

Logistic Regression

```
Score of LogisticRegression() is: 0.9616477272727273
Scores:
Accuracy Score: 0.9706818181818182
Cross Val Score: 0.6017727272727273
Classification Report:               precision    recall  f1-score   support

         1.0       0.95      0.99      0.97      1179
         2.0       0.85      0.88      0.87       204
         3.0       0.99      0.93      0.96       395
         4.0       0.98      0.97      0.97       872
         5.0       0.99      0.98      0.98      1750

    accuracy                           0.97      4400
   macro avg       0.95      0.95      0.95      4400
weighted avg       0.97      0.97      0.97      4400

Confusion Matrix: [[1167    9    1    2    0]
 [  11  180    2    9    2]
 [   9   12  369    5    0]
 [   0   11    1  842   18]
 [  37    0    0    0 1713]]
************************************************************
```

MultinomialNB

```
Score of MultinomialNB() is: 0.941875
Scores:
Accuracy Score: 0.9513636363636364
Cross Val Score: 0.5637272727272726
Classification Report:               precision    recall  f1-score   support

         1.0       0.95      0.96      0.96      1179
         2.0       0.96      0.77      0.86       204
         3.0       1.00      0.86      0.93       395
         4.0       0.93      0.94      0.94       872
         5.0       0.95      0.99      0.97      1750

    accuracy                           0.95      4400
   macro avg       0.96      0.91      0.93      4400
weighted avg       0.95      0.95      0.95      4400

Confusion Matrix: [[1137    1    0   24   17]
 [  17  158    1   24    4]
 [  19    3  341   11   21]
 [   3    2    0  822   45]
 [  22    0    0    0 1728]]
************************************************************
```

`Decision Tree Classifier

```
Score of DecisionTreeClassifier() is: 0.9646022727272727
Scores:
Accuracy Score: 0.9740909090909091
Cross Val Score: 0.5279090909090909
Classification Report:               precision    recall  f1-score   support

         1.0       0.96      0.99      0.98      1179
         2.0       0.83      0.92      0.87       204
         3.0       0.99      0.93      0.96       395
         4.0       0.99      0.96      0.98       872
         5.0       0.99      0.98      0.99      1750

    accuracy                           0.97      4400
   macro avg       0.95      0.96      0.95      4400
weighted avg       0.98      0.97      0.97      4400

Confusion Matrix: [[1167   10    1    1    0]
 [  11  187    2    2    2]
 [   9   15  369    2    0]
 [   0   13    1  840   18]
 [  27    0    0    0 1723]]
************************************************************
```

KNeighbors Classifier

```
Score of KNeighborsClassifier() is: 0.9598295454545455
Scores:
Accuracy Score: 0.9677272727272728
Cross Val Score: 0.48395454545454547
Classification Report:               precision    recall  f1-score   support

         1.0       0.96      0.99      0.98      1179
         2.0       0.91      0.79      0.85       204
         3.0       0.93      0.95      0.94       395
         4.0       0.95      0.99      0.97       872
         5.0       1.00      0.97      0.98      1750

    accuracy                           0.97      4400
   macro avg       0.95      0.94      0.94      4400
weighted avg       0.97      0.97      0.97      4400

Confusion Matrix: [[1163    3    9    4    0]
 [  11  162   14   15    2]
 [   8    7  374    6    0]
 [   0    7    3  859    3]
 [  24    0    3   23 1700]]
************************************************************
```

AdaBoost Classifier

```
Score of AdaBoostClassifier() is: 0.5553977272727273
Scores:
Accuracy Score: 0.5493181818181818
Cross Val Score: 0.4959545454545454
Classification Report:               precision    recall  f1-score   support

         1.0       0.55      0.72      0.63      1179
         2.0       1.00      0.12      0.22       204
         3.0       0.41      0.36      0.38       395
         4.0       0.39      0.29      0.34       872
         5.0       0.62      0.65      0.64      1750

    accuracy                           0.55      4400
   macro avg       0.59      0.43      0.44      4400
weighted avg       0.56      0.55      0.53      4400

Confusion Matrix: [[ 850    0   75  142  112]
 [ 103   25   22   24   30]
 [ 107    0  141   36  111]
 [ 114    0   66  256  436]
 [ 364    0   44  197 1145]]
************************************************************
```

GradientBoosting Classsifer

```
Score of GradientBoostingClassifier() is: 0.9574431818181818
Scores:
Accuracy Score: 0.9668181818181818
Cross Val Score: 0.5553636363636364
Classification Report:               precision    recall  f1-score   support

         1.0       0.93      0.99      0.96      1179
         2.0       0.96      0.76      0.85       204
         3.0       0.99      0.93      0.96       395
         4.0       0.98      0.97      0.97       872
         5.0       0.99      0.98      0.98      1750

    accuracy                           0.97      4400
   macro avg       0.97      0.93      0.94      4400
weighted avg       0.97      0.97      0.97      4400

Confusion Matrix: [[1170    1    1    2    5]
 [  36  155    2    9    2]
 [  18    2  369    5    1]
 [   8    3    1  842   18]
 [  32    0    0    0 1718]]
************************************************************
```

RandomForest Classifer

```
Score of RandomForestClassifier() is: 0.9646022727272727
Scores:
Accuracy Score: 0.9740909090909091
Cross Val Score: 0.606090909090909
Classification Report:               precision    recall  f1-score   support

         1.0       0.96      0.99      0.98      1179
         2.0       0.83      0.92      0.87       204
         3.0       0.99      0.93      0.96       395
         4.0       0.99      0.96      0.98       872
         5.0       0.99      0.98      0.99      1750

    accuracy                           0.97      4400
   macro avg       0.95      0.96      0.95      4400
weighted avg       0.98      0.97      0.97      4400

Confusion Matrix: [[1167   10    1    1    0]
 [  11  187    2    2    2]
 [   9   15  369    2    0]
 [   0   13    1  840   18]
 [  27    0    0    0 1723]]
************************************************************
```

We are getting best scores and cross-validation scores using Logistic regression and Random Forest Classifier but we will be using RFC for building our model because of its better scores.

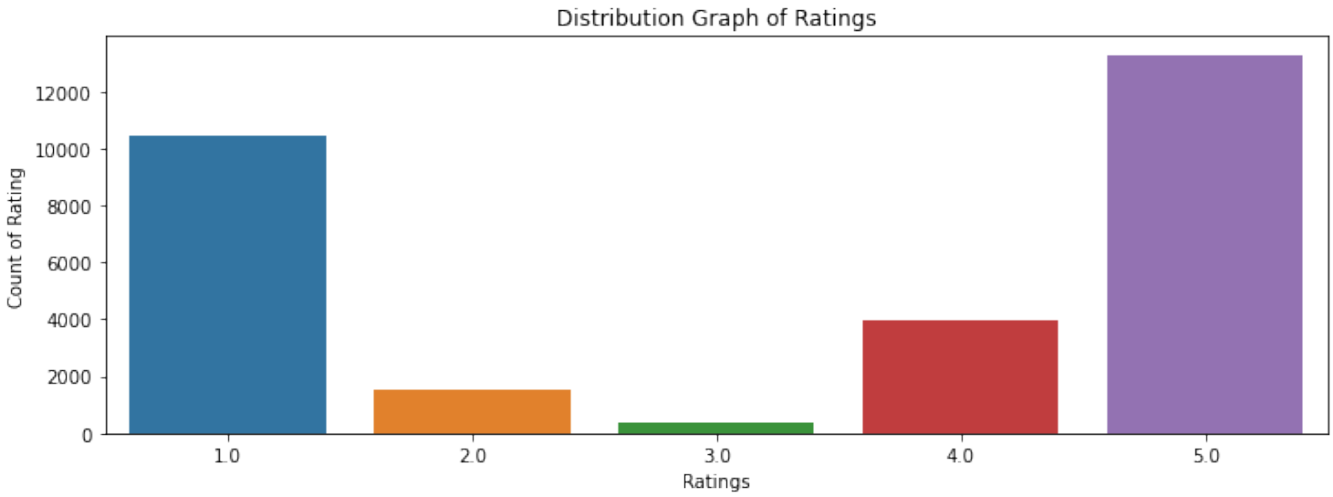
Now, we will improve their accuracy using Hyper-parameter tuning.¶

Saving Model:

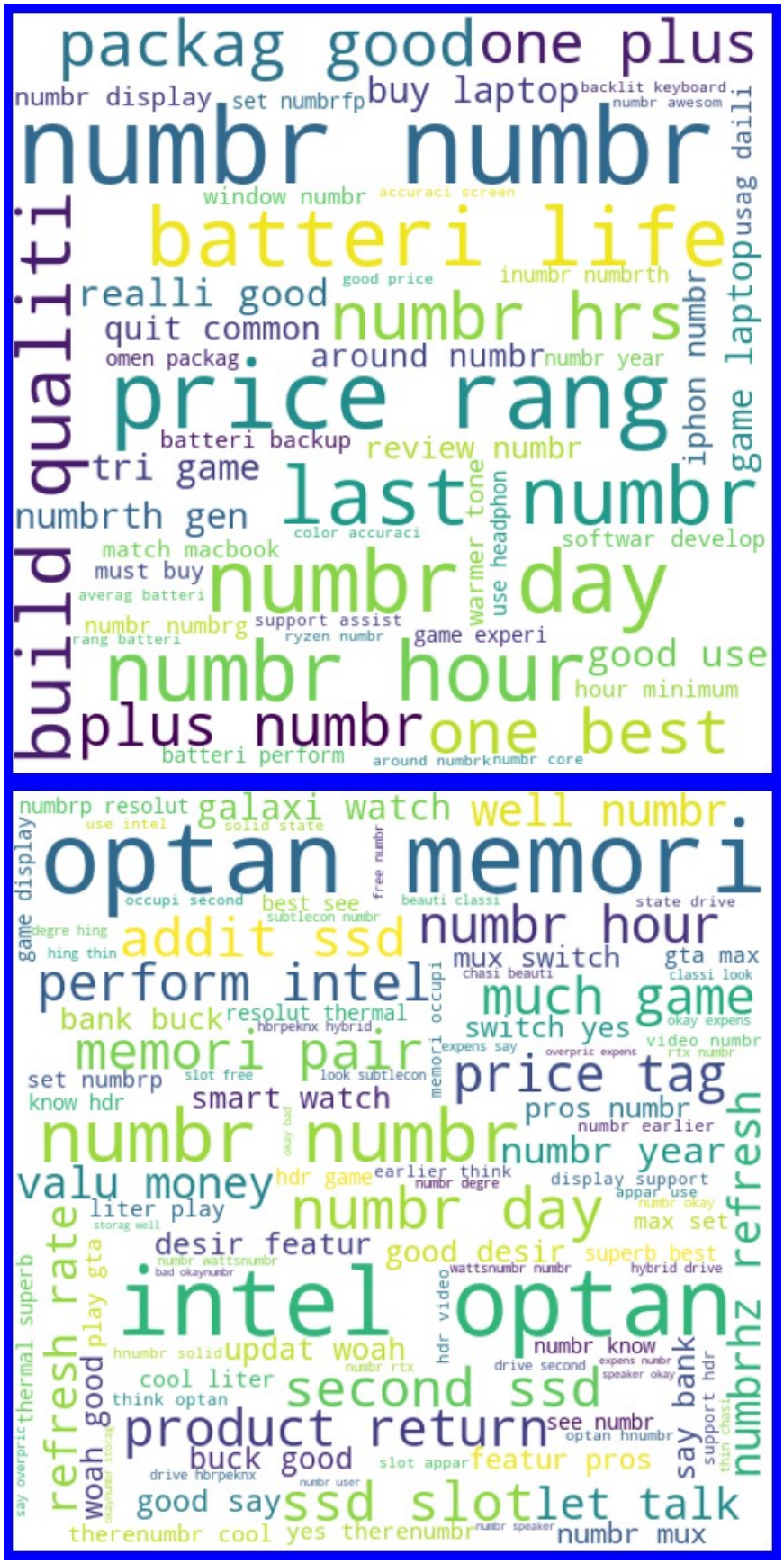
```
# Saving the best Ridge model
import joblib
joblib.dump(gb,'Rating_Pred.pkl')
```

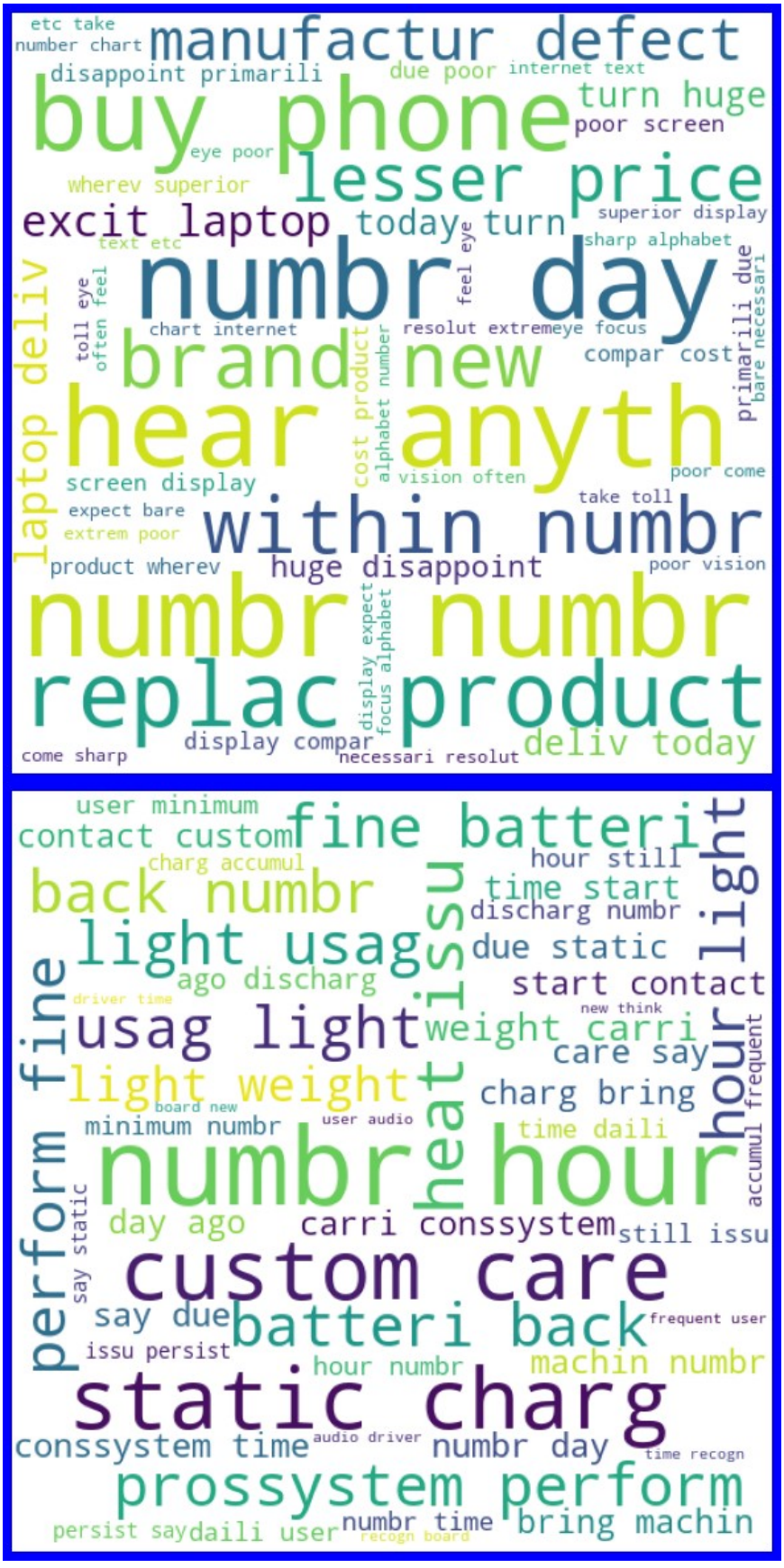
```
['Rating_Pred.pkl']
```

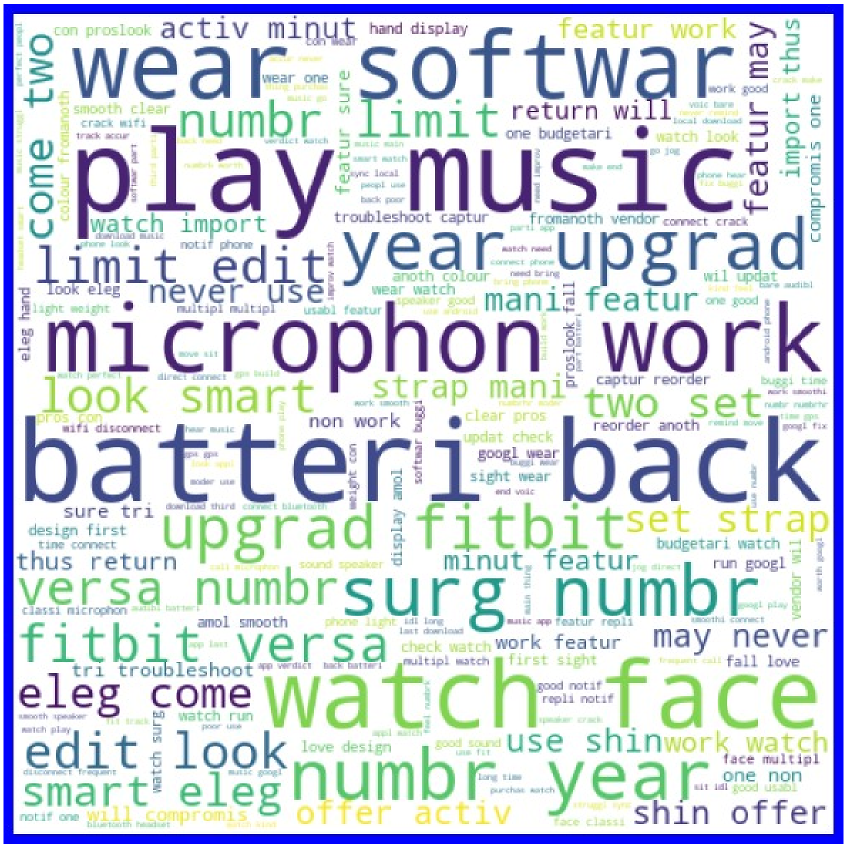
- **Visualizations & Interpretation of the Results**



We have also observed most frequent words in positive and negative comments through word-cloud:


packag good
one plus
numbr display
set numbrfp
buy laptop
backlit keyboard
numbr awesom
numbr numbr
window numbr
accuraci screen
batteri life
good price
inumbr numbrth
usag daili
realli good
numbr hrs
quit common
laptop
iphon numbr
omen packag
around numbr
numbr year
price rang
batteri backup
review numbr
game
tri game
last numbr
numbrth gen
color accuraci
softwar develop
match macbook
must buy
numbr day
warmer tone
use headphon
averag batteri
numbr numbrg
support assist
ryzen numbr
game experi
numbr hour
good use
hour minimum
plus numbr
one best
batteri perform
around numbrk
numbr core
numbrp resolut
galaxi watch
well numbr.
use intel
solid state
optan memori
occupi second
best see
beauti classi
state drive
subtlecon numbr
addit ssd
numbr hour
hing thin
mux switch
gta max
perform intel
chasi beauti
classi look
much game
bank buck
resolut thermal
switch yes
okay expens
hbrpeknx hybrid
video numbr
memori pair
expens say
overpric expens
rtx numbr
set numbrp
slot free
look subtlecon
price tag
know hdr
smart watch
pros numbr
numbr earlier
numbr numbr
numbr year
valu money
hdr game
earlier think
numbr degre
display support
appar use
liter play
numbr day
numbr okay
storag well
max set
desir featur
good desir
superb best
numbr wattsnumbr
bad okaynumbr
wattsnumbr numbr
hybrid drive
intel optan
refresh rate
numbrhz refresh
thermal superb
play gta
hdr video
updat woah
numbr know
hnumbr solid
numbr rtx
drive second
expens numbr
speaker okay
say bank
support hdr
cool liter
second ssd
think optan
product return
see numbr
optan hnumbr
buck good
slot appar
featur pros
drive hbrpeknx
numbr user
good say
ssd slot
let talk
therenumbr cool yes therenumbr
numbr speaker
numbr mux


etc take
number chart
manufactur defect
disappoint primarili
due poor internet text
buy phone
turn huge
poor screen
eye poor
lesser price
wherev superior
excit laptop
today
turn
superior display
text etc
sharp alphabet
numbr day
feel eye
toll eye
often feel
chart internet
resolut extremeye focus
primarili due
bare necessari
brand new
compar cost
laptop deliv
cost product
alphabet number
hear anyth
vision often
poor come
screen display
take toll
expect bare
within numbr
extrem poor
product wherev
huge disappoint
poor vision
numbr numbr
display expect
focus alphabet
replac product
come sharp
display compar
necessari resolut
deliv today
user minimum
contact custom
fine batteri
charg accumul
hour still
back numbr
time start
discharg numbr
light usag
due static
ago discharg
start contact
driver time
new think
usag light
weight carri
care say
light weight
charg bring
board new
minimum numbr
user audio
time daili
numbr hour
heat issu
light hour
perform fine
accumul frequent
say static
day ago
carri conssystem
still issu
custom care
say due
batteri back
frequent user
issu persist
hour numbr
machin numbr
static charg
conssystem time
audio driver
numbr day
time recogn
prossystem perform
persist say
daili user
numbr time
recogn board
bring machin


batteri back
play music
microphon work
wear softwar
watch face
year upgrad
numbr year
surg numbr
upgrad fitbit
fitbit versa
versa numbr
limit edit
numbr limit
look smart
edit look
smart eleg
eleg come
two set
set strap
strap mani
mani featur
come two
activ minut
featur work
featur may
return will
never use
offer activ
shin offer
use shin
work watch
may never
minut featur
watch import
import thus
compromis one
thus return
tri troubleshoot
will compromis
one non
non work
watch look
work featur
love design
troubleshoot captur
fromanoth vendor
captur reorder
reorder anoth
budgetari watch
one budgetari
wear watch
anoth colour
colour fromanoth
hand display
con proslook
smooth clear
crack wifi
wil updat
updat check
googl wear
sight wear
clear pros
pros con
sure tri
design first
check watch
first sight
fall love
amol smooth
good sound
face multipl
watch surg
proslook fall
display amol
software buggi
run googl
vendor wil
multipl multipl
usabl featur
speaker good
one good
buggi time
wifi disconnect
light weight
look eleg
sound speaker
good usabl
notif one
watch run
weight con

CONCLUSION

- **Key Findings and Conclusions of the Study –**
  After using different models for classification, we concluded that Random Forest Classifier was best suited to train the dataset for Rating Prediction Analysis. - The dataset was imbalanced in nature. - Some data had to be omitted from the datasets in order to remove data irrelevant to the dataset.
- **Learning Outcomes of the Study in respect of Data Science –**
  As the datasets was huge it created problems when we tried to manipulate it or cleanse it. As when we tried to cleanse it we had to take care that data doesn't loose its relevancyand structure. So, we tried to remove much of the unnecessary data from the datasets that was in our reach during the given time. - And finally, we came to know that the best algorithm used to train the machine for this the dataset is Random Forest Classifier as all the values along the metrics were highest.
- **Limitations of this work and Scope for Future Work –**
  More time consumption during hyperparameter tuning for both models, as the data was large. - Less number of parameters were used during tuning. - Scrapping of data from different websites were of different process and the length of data were differing in most cases so I sticked to Amazon and Scrapped data which are famous in the site. - Some of the reviews were bad and the text had morewrong information about the product