FLIP ROBO

RATINGS PREDICTION

Submitted by:

NITIN SINGH TATRARI

Batch: 1825

ACKNOWLEDGMENT

I have read article on rating and review which covers – how they are important for business scenario, how customers depend on them for purchasing decision making.

I have gone through ecommerce websites ‘Flipkart’, read different reviews and try to understand the relation of reviews and rating.

INTRODUCTION

- Business Problem Framing

Here, we are trying to develop a rating system to predict the rating for the reviews. The rating will be given out of 5 stars- 1star, 2star, 3star, 4star & 5star. Websites where there were only reviews given, we can now classify the reviews into five categories by giving them ratings (1 to 5).

As there will be many reviews on a product, customer may not want to read all the reviews. So he may choose to read the reviews based on rating as per his thinking.

- Conceptual Background of the Domain Problem

Since 92% of consumers now read online reviews, e-Commerce businesses need to rethink their strategies for implementing the proper digital strategy in order to take advantage of the viral effects that customer reviews have. Thus, collection of ratings and reviews can be used as an immensely powerful tool for the business.

About 40 percent of customers even say they wouldn't buy electronics without reading online reviews. Reviews become even more important in the absence of the ability to 'test' products prior to purchasing online, which is not a concern in brick-and-mortar stores.

Customers will first look for product ratings, i.e., ratings out of five stars, to see which products deserve their attention. Once a product has been clicked on, prospective customers will then compare reviews to one another and depend on the feedback from other customers.

Even bad reviews can be helpful for stores. It helps customers know what the worst scenario they can expect and serve as risk mitigators.

- ## Review of Literature

This is a comprehensive summary of the research done on the topic. The review should enumerate, describe, summarize, evaluate and clarify the research done.

Today’s shoppers trust the reviews of strangers more than they trust the advertisement from a company. According to a 2016 study by Brightlocal, a survey found that “84% of consumers trust online reviews as much as they would a personal recommendation from a friend.”

By taking advantage of your customer reviews & ratings, your business can monitor and improve brand awareness, reputation, and loyalty. Generating conversations through digital word-of-mouth is extremely important for influencing and guiding purchasing decisions.

By collecting and publishing authentic customer feedback, businesses can help drive higher SEO, generate new customers, and increase sales. A study by BrightLocal found that 74% of consumers stated that after seeing customer reviews about a product on a landing page, they were willing to take the next step in the customer journey.

Google is extremely important for the collection and advertisement of customer reviews and ratings. Since Google provides the opportunity for businesses to showcase customer feedback as an average star rating alongside their ads, businesses are able to increase online visibility and enhance their trust with the customer. As Chris Anderson, businessman and current head of TED, puts it, “Your brand isn’t what you say it is — it’s what Google says it is”.

Through reviews, businesses can utilize this meaningful and useful information from their own customers to help create more effective strategies. By integrating customer feedback into the brand and marketing strategies of your business, you will stay ahead of the competition and enhance experience management.

- Motivation for the Problem Undertaken

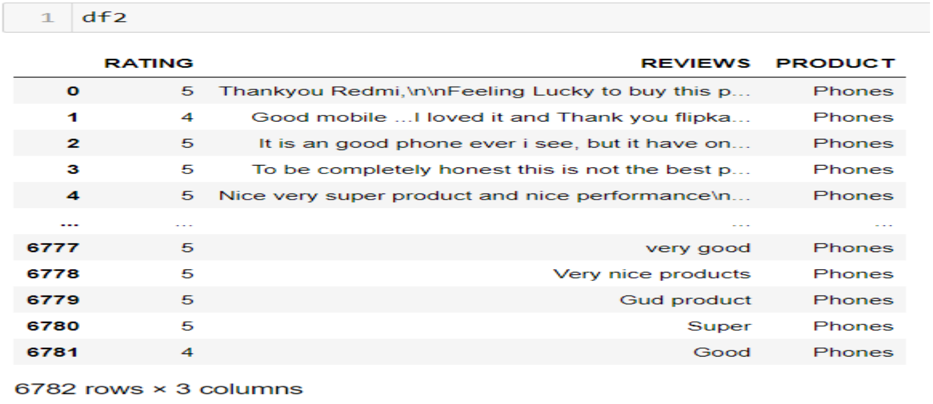
Since e-commerce is one of the fastest growing industry and post pandemic, people are getting more dependent on e-commerce shopping platforms to avoid physical contact at marketplaces, a good rating and reviews system strategy is very crucial for future business. This challenge is going to be common in upcoming market scenario.

Thus, studying customer behaviour through ratings and reviews will be the new tread in coming years for businesses.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

1) We have scraped rating & reviews of different products from Flipkart website. Let us consider below example for phones.

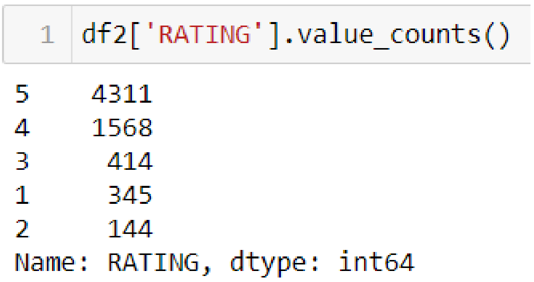


```
df2
```

	RATING	REVIEWS	PRODUCT
0	5	Thankyou Redmi,\n\nFeeling Lucky to buy this p...	Phones
1	4	Good mobile ...I loved it and Thank you flipka...	Phones
2	5	It is an good phone ever i see, but it have on...	Phones
3	5	To be completely honest this is not the best p...	Phones
4	5	Nice very super product and nice performance\n...	Phones
...	...	...	...
6777	5	very good	Phones
6778	5	Very nice products	Phones
6779	5	Gud product	Phones
6780	5	Super	Phones
6781	4	Good	Phones

6782 rows × 3 columns

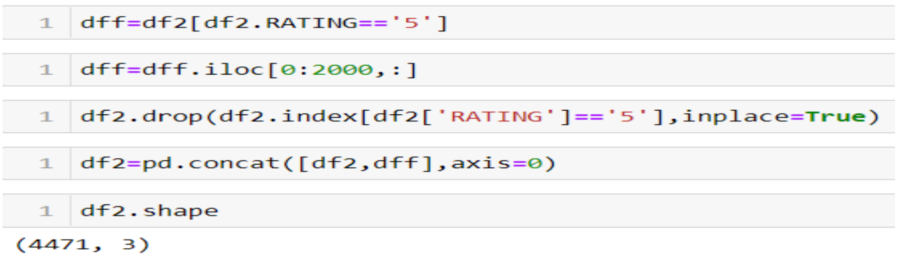
Here, we have store all the data in Data Frame df2 which consist of 6782 rows and 3 columns.



```
df2['RATING'].value_counts()
```

```
5    4311
4    1568
3     414
1     345
2     144
Name: RATING, dtype: int64
```

Here, we can see that data is not balanced. So we try to balance the data by reducing rows having ‘5’ star rating.



```
dff=df2[df2.RATING=='5']
```

```
dff=dff.iloc[0:2000,:]
```

```
df2.drop(df2.index[df2['RATING']=='5'],inplace=True)
```

```
df2=pd.concat([df2,dff],axis=0)
```

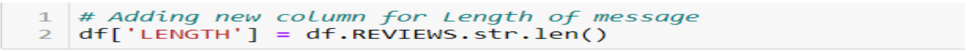
```
df2.shape
```

```
(4471, 3)
```

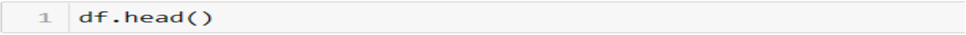
Thus, now the rows have been reduced from 6782 to 4471. Similarly we apply the same methodology with other products.

2) We have added two columns-

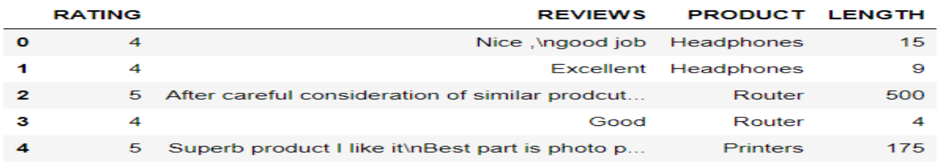
a) LENGTH: It shows the count of Reviews



```
# Adding new column for Length of message
df['LENGTH'] = df.REVIEWS.str.len()
```

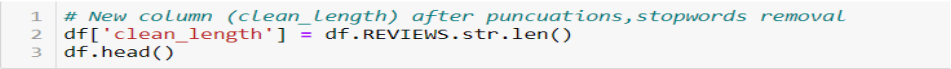


```
df.head()
```

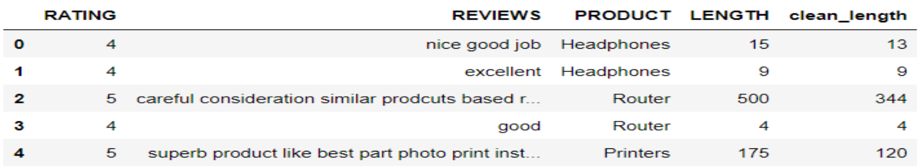


	RATING	REVIEWS	PRODUCT	LENGTH
0	4	Nice ,\ngood job	Headphones	15
1	4	Excellent	Headphones	9
2	5	After careful consideration of similar prodcut...	Router	500
3	4	Good	Router	4
4	5	Superb product I like it\nBest part is photo p...	Printers	175

b) Clean_length: It shows the count of review after data processing.



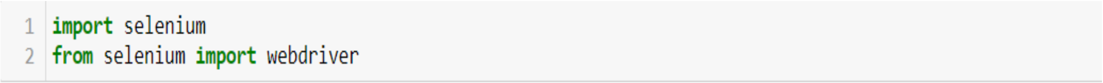
```
# New column (clean_length) after puncuations,stopwords removal
df['clean_length'] = df.REVIEWS.str.len()
df.head()
```



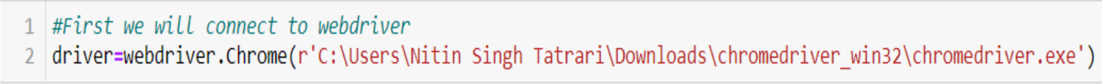
	RATING	REVIEWS	PRODUCT	LENGTH	clean_length
0	4	nice good job	Headphones	15	13
1	4	excellent	Headphones	9	9
2	5	careful consideration similar prodcuts based r...	Router	500	344
3	4	good	Router	4	4
4	5	superb product like best part photo print inst...	Printers	175	120

- Data Sources and their formats

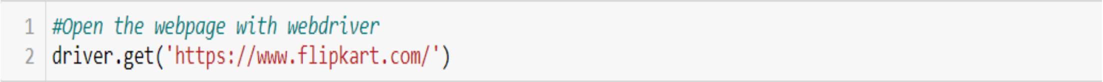
1) The data have been scrapped from Flipkart using selenium library.
Here, we are trying to scrap all rating and reviews for ‘Laptop’.



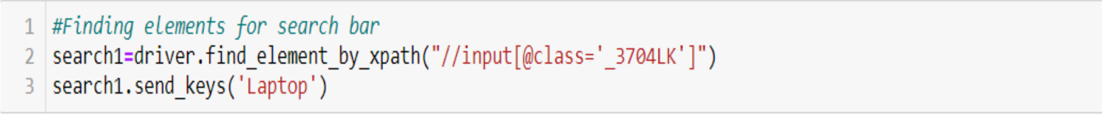
```
import selenium
from selenium import webdriver
```



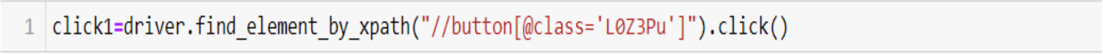
```
#First we will connect to webdriver
driver=webdriver.Chrome(r'C:\Users\Nitin Singh Tatrari\Downloads\chromedriver_win32\chromedriver.exe')
```



```
#Open the webpage with webdriver
driver.get('https://www.flipkart.com/')
```



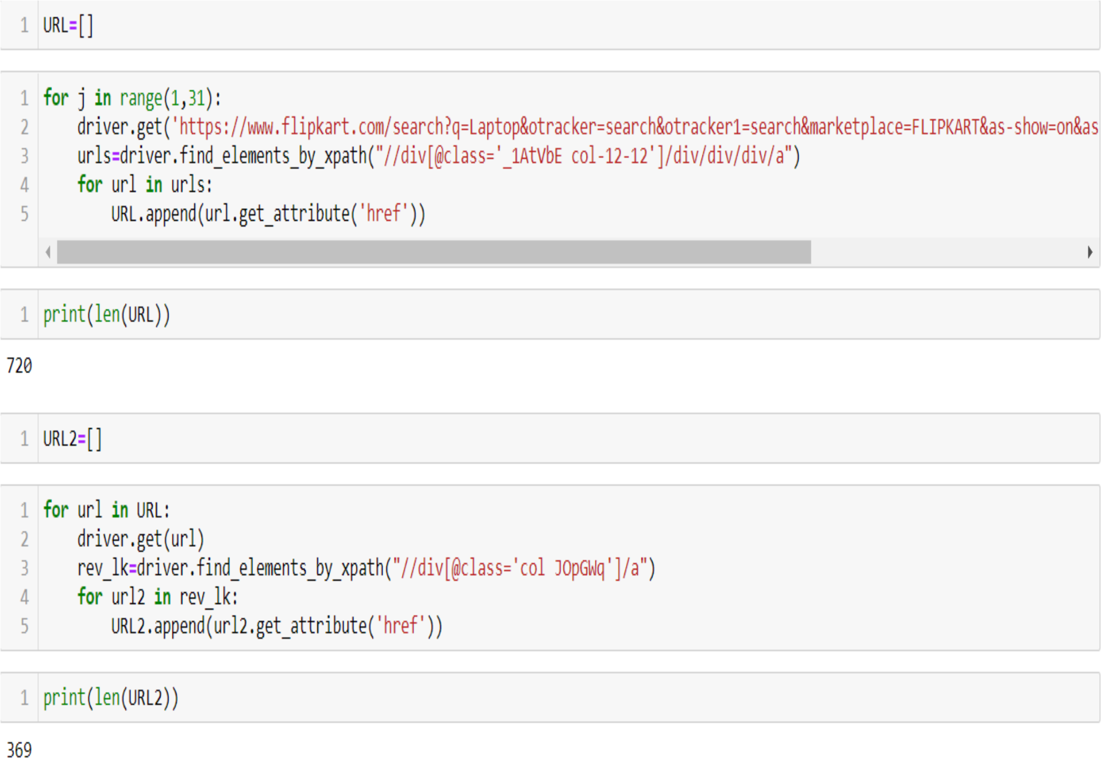
```
#Finding elements for search bar
search1=driver.find_element_by_xpath("//input[@class='_3704LK']")
search1.send_keys('Laptop')
```



```
click1=driver.find_element_by_xpath("//button[@class='L0Z3Pu']").click()
```

We save all url of laptops page in URL list.

Then, we save all url of laptops having reviews in URL2 list.



```
URL=[]
```

```
for j in range(1,31):
    driver.get('https://www.flipkart.com/search?q=Laptop&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as
    urls=driver.find_elements_by_xpath("//div[@class='_1AtVbE col-12-12']/div/div/div/a")
    for url in urls:
        URL.append(url.get_attribute('href'))
```

```
print(len(URL))
```

```
720
```

```
URL2=[]
```

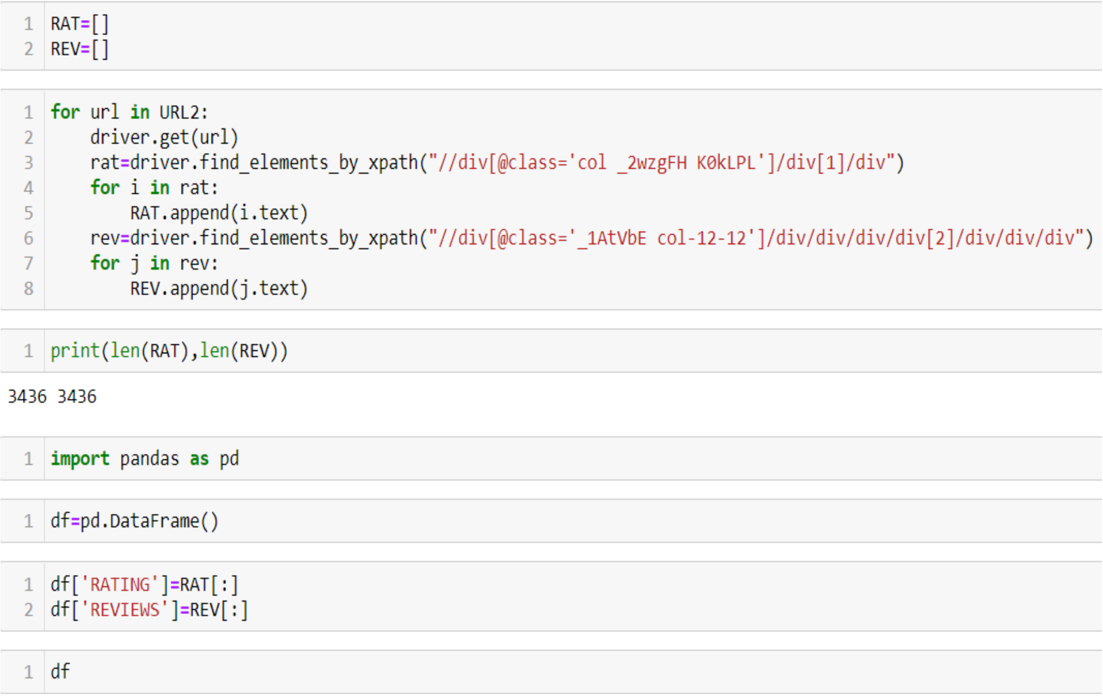
```
for url in URL:
    driver.get(url)
    rev_lk=driver.find_elements_by_xpath("//div[@class='col JOpGWq']/a")
    for url2 in rev_lk:
        URL2.append(url2.get_attribute('href'))
```

```
print(len(URL2))
```

```
369
```

Now, we scrap all the data and save it in RAT & REV list.

We load the data into Data Frame df



```
RAT=[]
REV=[]
```

```
for url in URL2:
    driver.get(url)
    rat=driver.find_elements_by_xpath("//div[@class='col _2wzgFH K0kLPL']/div[1]/div")
    for i in rat:
        RAT.append(i.text)
    rev=driver.find_elements_by_xpath("//div[@class='_1AtVbE col-12-12']/div/div/div[2]/div/div/div")
    for j in rev:
        REV.append(j.text)
```

```
print(len(RAT),len(REV))
```

```
3436 3436
```

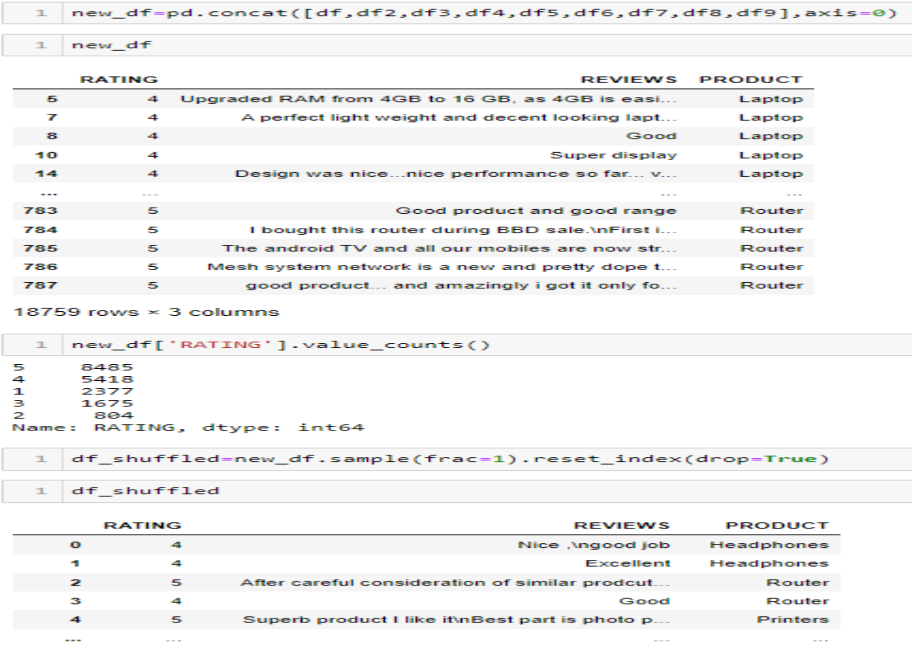
```
import pandas as pd
```

```
df=pd.DataFrame()
```

```
df['RATING']=RAT[:]
df['REVIEWS']=REV[:]
```

```
df
```

Thus, similarly we have scrapped the data of all products and Combine all data frame into one



```
new_df=pd.concat([df,df2,df3,df4,df5,df6,df7,df8,df9],axis=0)
```

```
new_df
```

	RATING	REVIEWS	PRODUCT
5	4	Upgraded RAM from 4GB to 16 GB, as 4GB is easi...	Laptop
7	4	A perfect light weight and decent looking lapt...	Laptop
8	4	Good	Laptop
10	4	Super display	Laptop
14	4	Design was nice...nice performance so far... v...	Laptop
...	...	...	...
783	5	Good product and good range	Router
784	5	I bought this router during BBD sale.\nFirst i...	Router
785	5	The android TV and all our mobiles are now str...	Router
786	5	Mesh system network is a new and pretty dope t...	Router
787	5	good product... and amazingly i got it only fo...	Router

18759 rows × 3 columns

```
new_df['RATING'].value_counts()
```

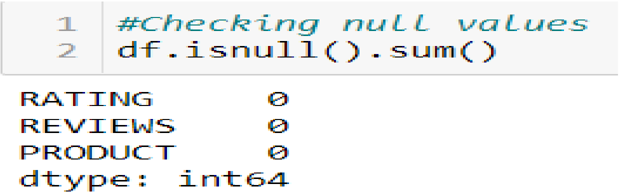
```
5     8485
4     5418
1     2377
3     1675
2      804
Name: RATING, dtype: int64
```

```
df_shuffled=new_df.sample(frac=1).reset_index(drop=True)
```

```
df_shuffled
```

	RATING	REVIEWS	PRODUCT
0	4	Nice ,\ngood job	Headphones
1	4	Excellent	Headphones
2	5	After careful consideration of similar prodcut...	Router
3	4	Good	Router
4	5	Superb product I like it\nBest part is photo p...	Printers
...	...	...	...

2) Checking null values

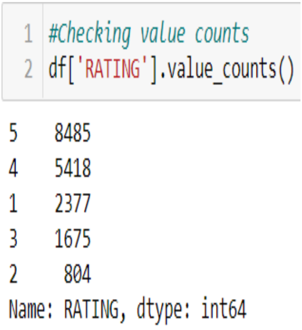


```
#Checking null values
df.isnull().sum()
```

```
RATING      0
REVIEWS     0
PRODUCT     0
dtype: int64
```

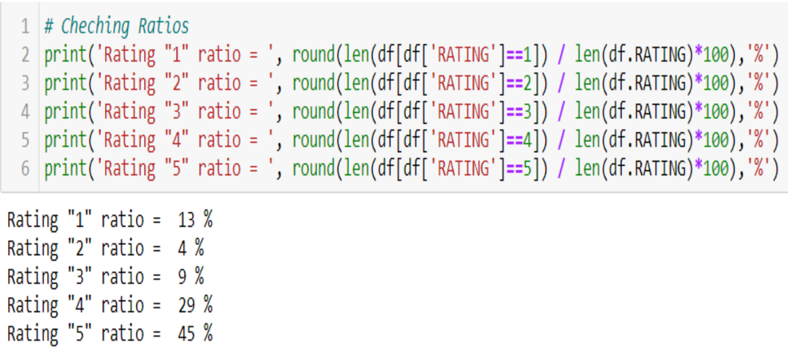
There are no null values in the data frame.

3) Checking value count and Ratio



```
#Checking value counts
df['RATING'].value_counts()
```

```
5    8485
4    5418
1    2377
3    1675
2     804
Name: RATING, dtype: int64
```



```
# Cheching Ratios
print('Rating "1" ratio = ', round(len(df[df['RATING']==1]) / len(df.RATING)*100),'%')
print('Rating "2" ratio = ', round(len(df[df['RATING']==2]) / len(df.RATING)*100),'%')
print('Rating "3" ratio = ', round(len(df[df['RATING']==3]) / len(df.RATING)*100),'%')
print('Rating "4" ratio = ', round(len(df[df['RATING']==4]) / len(df.RATING)*100),'%')
print('Rating "5" ratio = ', round(len(df[df['RATING']==5]) / len(df.RATING)*100),'%')
```

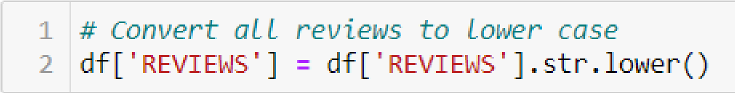
```
Rating "1" ratio =  13 %
Rating "2" ratio =  4 %
Rating "3" ratio =  9 %
Rating "4" ratio =  29 %
Rating "5" ratio =  45 %
```

We have try to balance the data at the scrapping by eliminating rows having 5 star rating as more than 80% rating was of 5 star.

Thus we get the above value count and rating ratio, after data balancing.

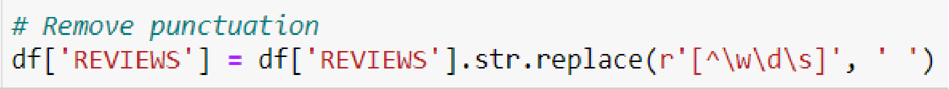
- ## Data Pre-processing Done

1) Converting all data in REVIEW column to lower case

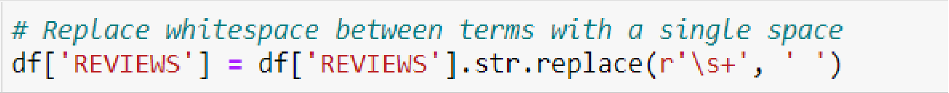


```
# Convert all reviews to lower case
df['REVIEWS'] = df['REVIEWS'].str.lower()
```

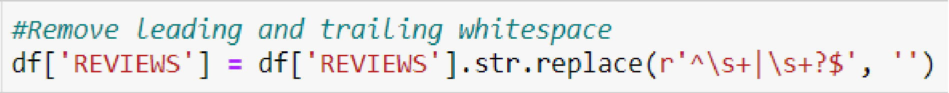
2) Removing punctuations and white spaces



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

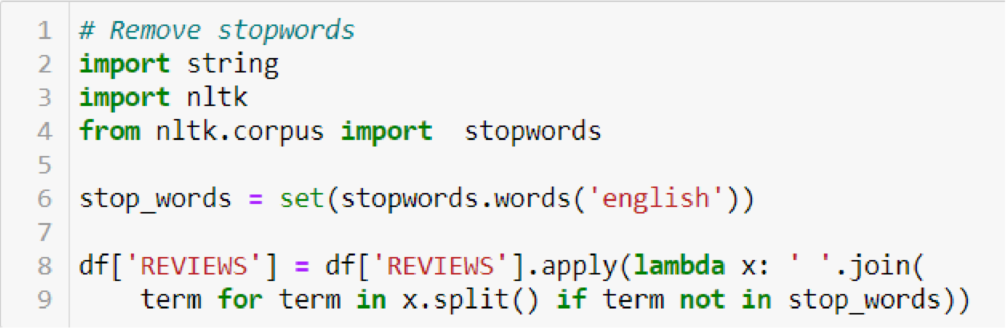


```
# Replace whitespace between terms with a single space
df['REVIEWS'] = df['REVIEWS'].str.replace(r'\s+', ' ')
```



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

3) Removing stop words



```
# Remove stopwords
import string
import nltk
from nltk.corpus import  stopwords

stop_words = set(stopwords.words('english'))

df['REVIEWS'] = df['REVIEWS'].apply(lambda x: ' '.join(
    term for term in x.split() if term not in stop_words))
```

- ## Data Inputs- Logic- Output Relationships

1) Term Frequency Inverse Document Frequency Vectorizer(TF-IDF)

We have converted the text in REVIEW column into meaningful representation of numbers which is used to fit machine algorithm for prediction.

- ## State the set of assumptions (if any) related to the problem under consideration

  We have not segregated reviews on the basic product type for machine learning. We have considered reviews of all products as one type data.

  We have only scrape reviews on first page for all products.

- ## Hardware and Software Requirements and Tools Used

  The libraries used are: pandas, numpy, matplotlib.pyplot, seaborn and scikit_learn. The laptop used is with Intel I5 10th generation, 4GB RAM, 4GB GPU.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  I have cleaned the data by removing punctuation, whitespace and stop words.

  Then I have changed the review into vector using TF-DIF vectorizer.

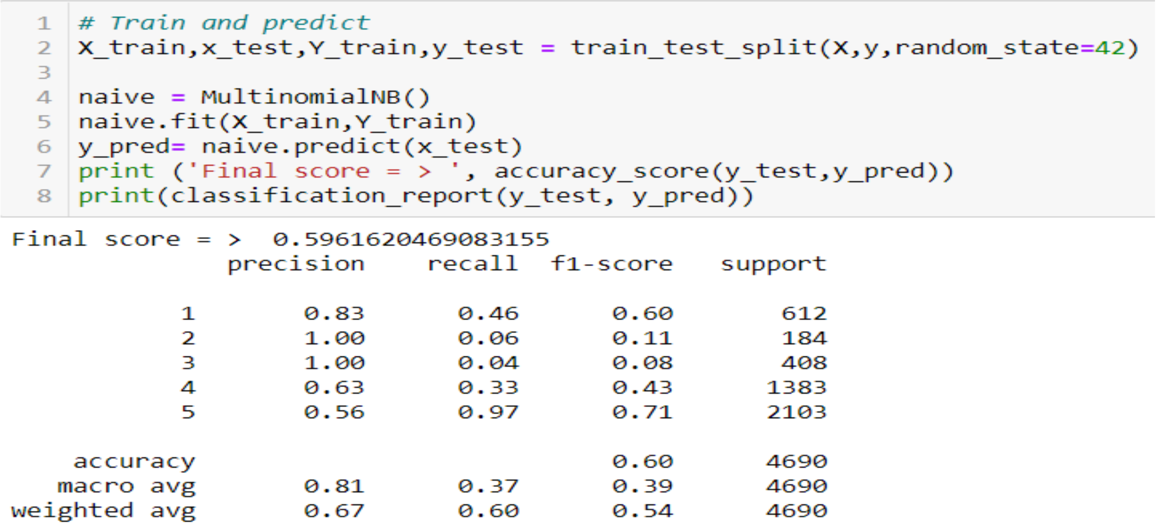
- Testing of Identified Approaches (Algorithms)

  The algorithms used for testing are as follows:-

  1) Term Frequency Inverse Document Frequency Vectorizer(TF-IDF)
  2) Space Vector Classifier
  3) Multinomial Naïve Bayes

- Run and Evaluate selected models

  We have converted the text in REVIEW column into meaningful representation of numbers (vectors) by TF-IDF vectorizer which is used to fit machine algorithm for prediction.

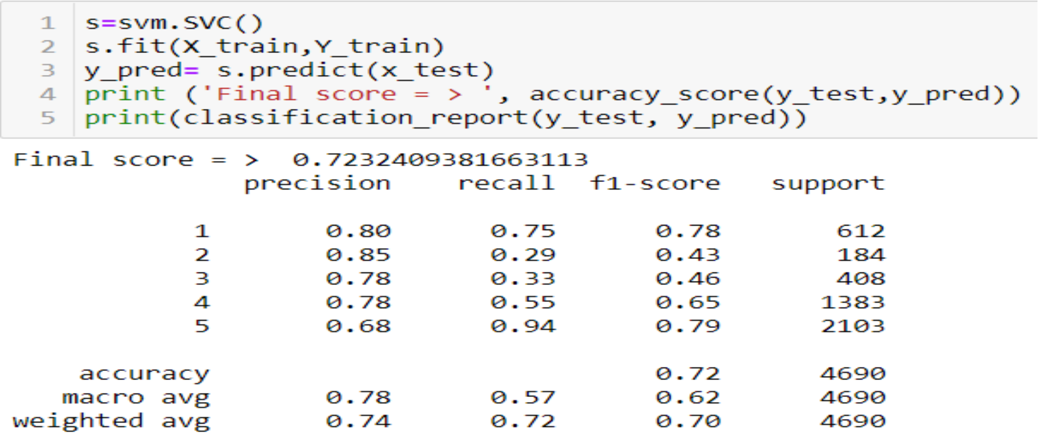


```
# Train and predict
X_train,x_test,Y_train,y_test = train_test_split(X,y,random_state=42)

naive = MultinomialNB()
naive.fit(X_train,Y_train)
y_pred= naive.predict(x_test)
print ('Final score = > ', accuracy_score(y_test,y_pred))
print(classification_report(y_test, y_pred))
```

```
Final score = >  0.5961620469083155
              precision    recall  f1-score   support

           1       0.83      0.46      0.60       612
           2       1.00      0.06      0.11       184
           3       1.00      0.04      0.08       408
           4       0.63      0.33      0.43      1383
           5       0.56      0.97      0.71      2103

    accuracy                           0.60      4690
   macro avg       0.81      0.37      0.39      4690
weighted avg       0.67      0.60      0.54      4690
```

We split the data into train and test.

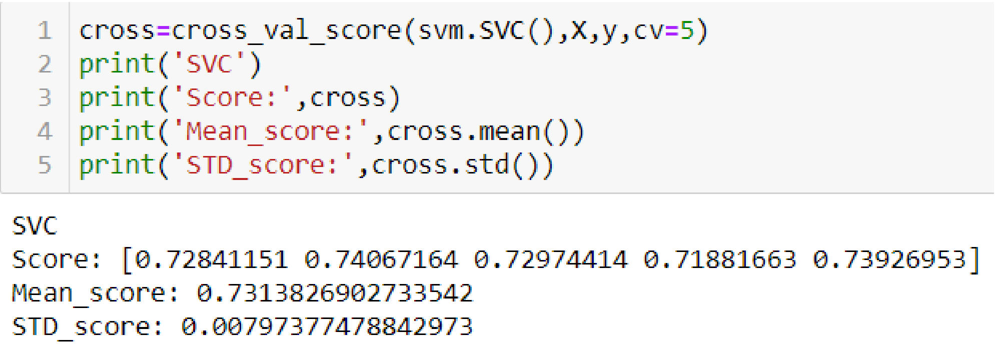
We apply Multinomial Naives Bayes and gets accuracy score of 59%.



```
s=svm.SVC()
s.fit(X_train,Y_train)
y_pred= s.predict(x_test)
print ('Final score = > ', accuracy_score(y_test,y_pred))
print(classification_report(y_test, y_pred))
```

```
Final score = >  0.7232409381663113
              precision    recall  f1-score   support

           1       0.80      0.75      0.78       612
           2       0.85      0.29      0.43       184
           3       0.78      0.33      0.46       408
           4       0.78      0.55      0.65      1383
           5       0.68      0.94      0.79      2103

    accuracy                           0.72      4690
   macro avg       0.78      0.57      0.62      4690
weighted avg       0.74      0.72      0.70      4690
```

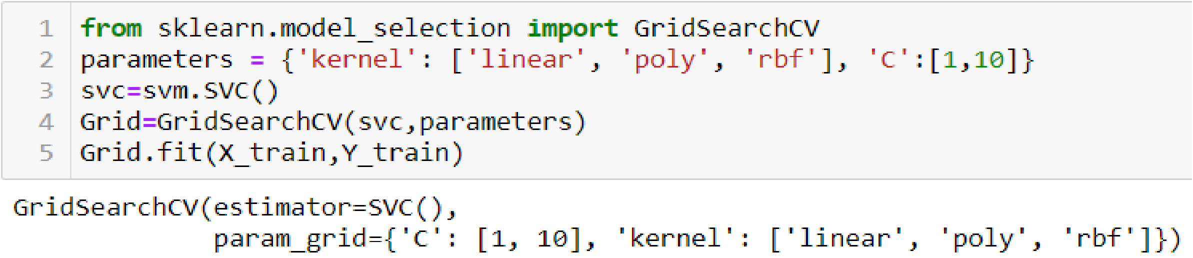
We now apply Space vector classifier and get an accuracy score of 72%.



```
cross=cross_val_score(svm.SVC(),X,y,cv=5)
print('SVC')
print('Score:',cross)
print('Mean_score:',cross.mean())
print('STD_score:',cross.std())
```

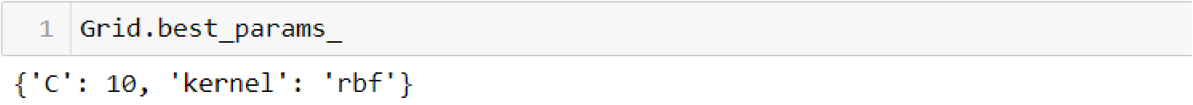
```
SVC
Score: [0.72841151 0.74067164 0.72974414 0.71881663 0.73926953]
Mean_score: 0.7313826902733542
STD_score: 0.00797377478842973
```

We have cross validated the score from SVC. The result is satisfactory with Space vector classifier.



```
from sklearn.model_selection import GridSearchCV
parameters = {'kernel': ['linear', 'poly', 'rbf'], 'C':[1,10]}
svc=svm.SVC()
Grid=GridSearchCV(svc,parameters)
Grid.fit(X_train,Y_train)
```

```
GridSearchCV(estimator=SVC(),
             param_grid={'C': [1, 10], 'kernel': ['linear', 'poly', 'rbf']})
```



```
Grid.best_params_
```

```
{'C': 10, 'kernel': 'rbf'}
```

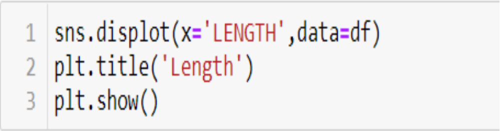
By applying Grid search CV, we found the best parameter for SVC algorithm is in radial basis function kernel and penalty parameter ‘C’ =10.

- Key Metrics for success in solving problem under consideration

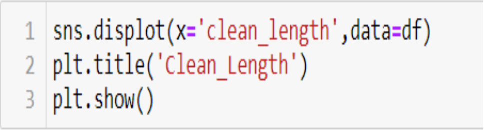
  The metrics used are accuracy_score, confusion_matrix, classification_report.

- Visualizations

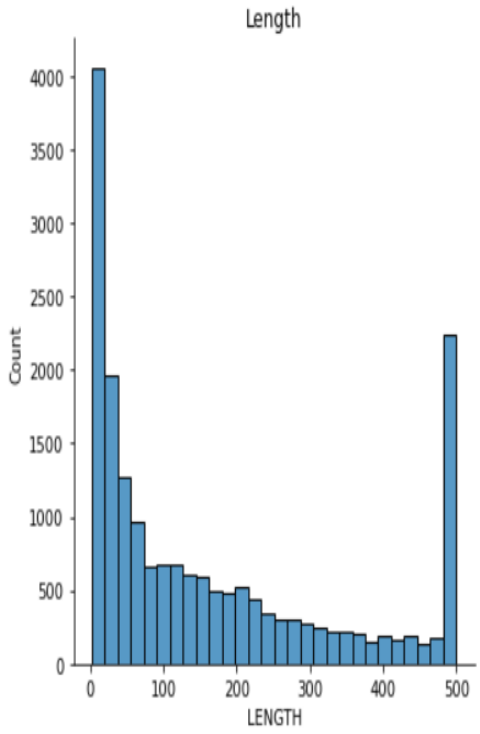
  A) Plotting graphs

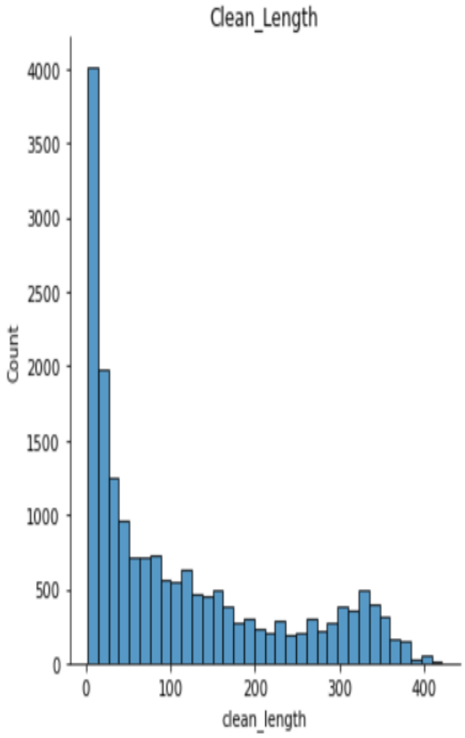


```
sns.displot(x='LENGTH',data=df)
plt.title('Length')
plt.show()
```

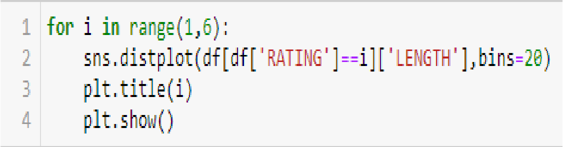


```
sns.displot(x='clean_length',data=df)
plt.title('Clean_Length')
plt.show()
```

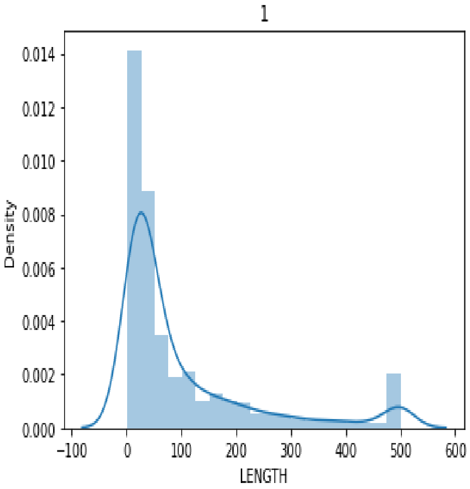


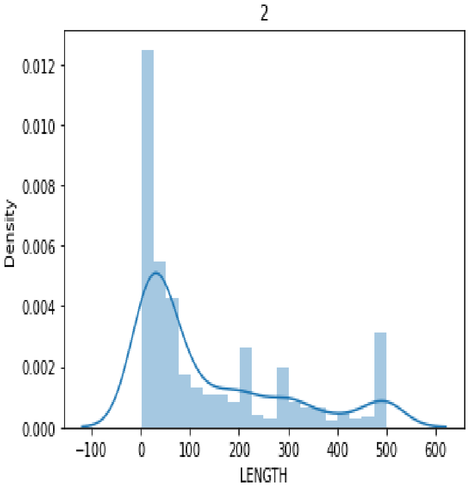


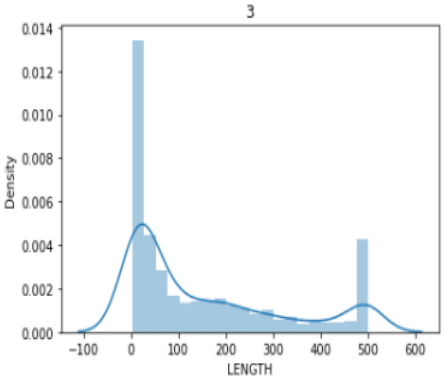
We can see the length is reduced after cleaning. Maximum length of 500 is reduced to around 400 after cleaning.

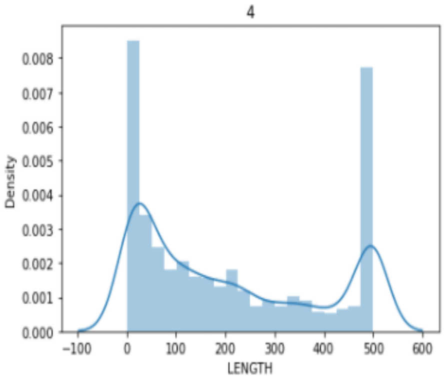


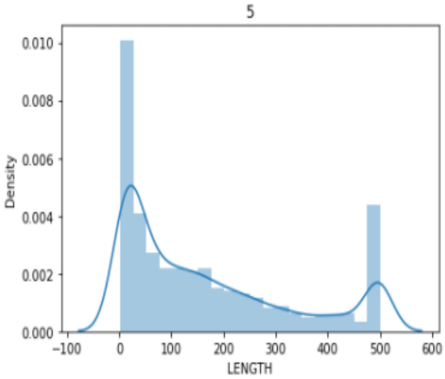
```python
for i in range(1,6):
    sns.distplot(df[df['RATING']==i]['LENGTH'],bins=20)
    plt.title(i)
    plt.show()
```



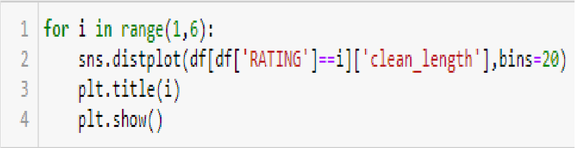




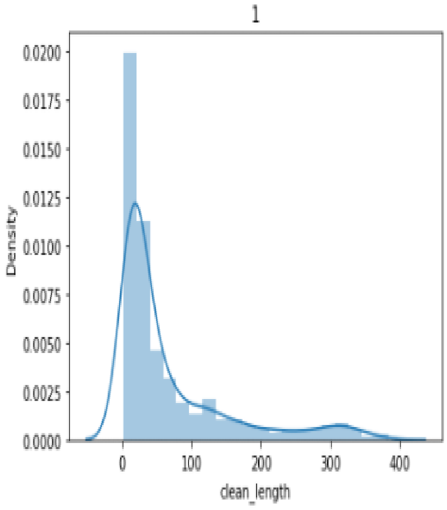


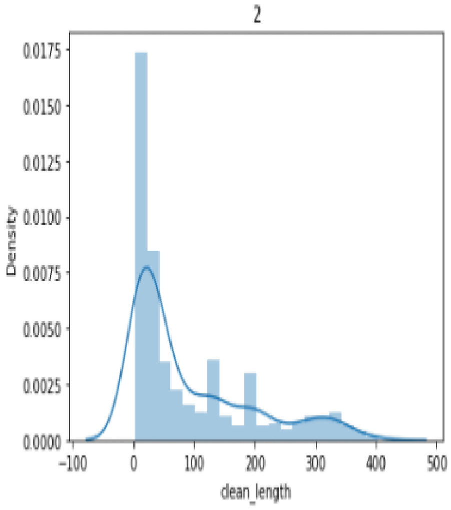


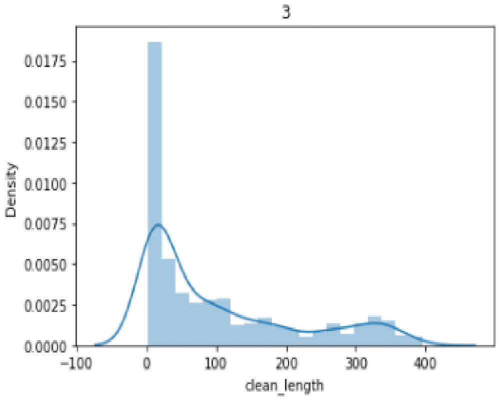
All ratings have maximum frequency at 0-20 length. Second highest frequency in 4 & 5 rating at range of 500 lengths.

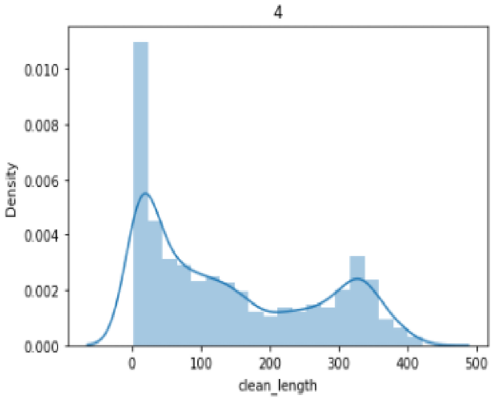


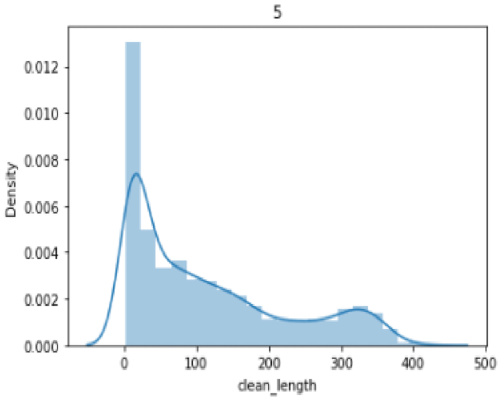
```
for i in range(1,6):
    sns.distplot(df[df['RATING']==i]['clean_length'],bins=20)
    plt.title(i)
    plt.show()
```





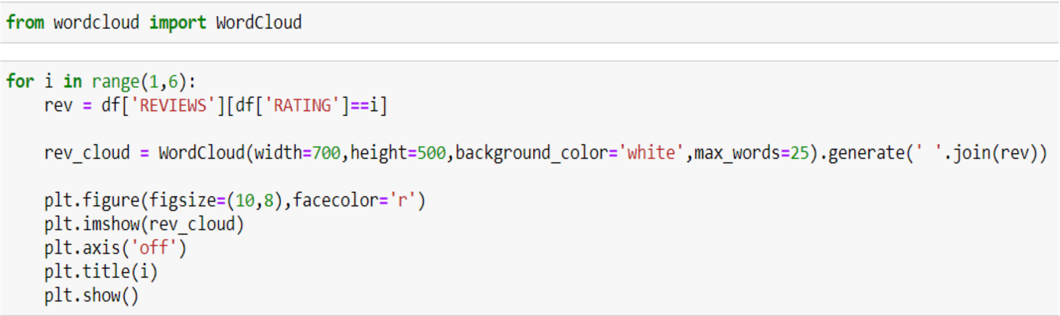






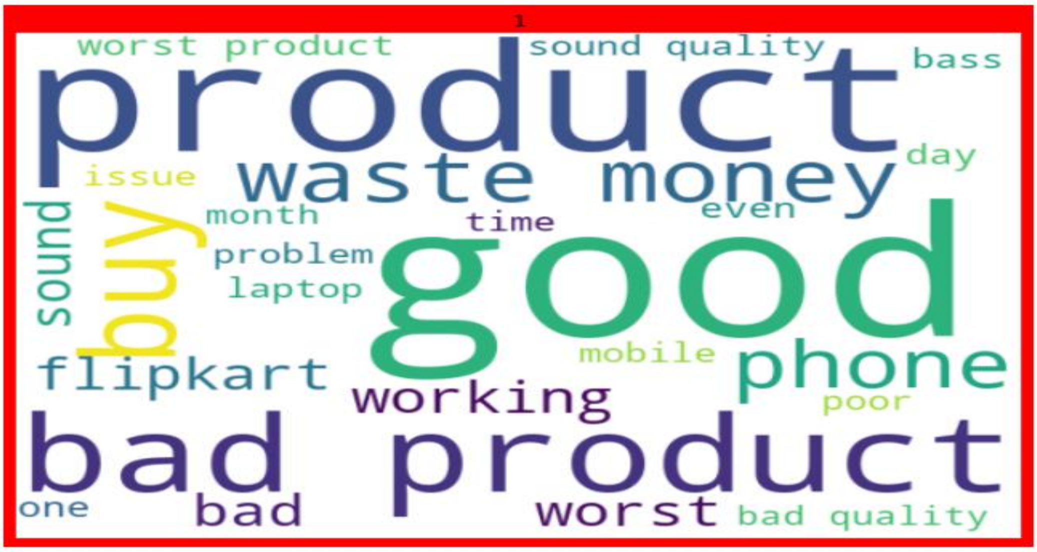
The clean length also has the maximum frequency at 0-20 range.

Second highest frequency is also reduced and it’s now in range in 20-60 range.

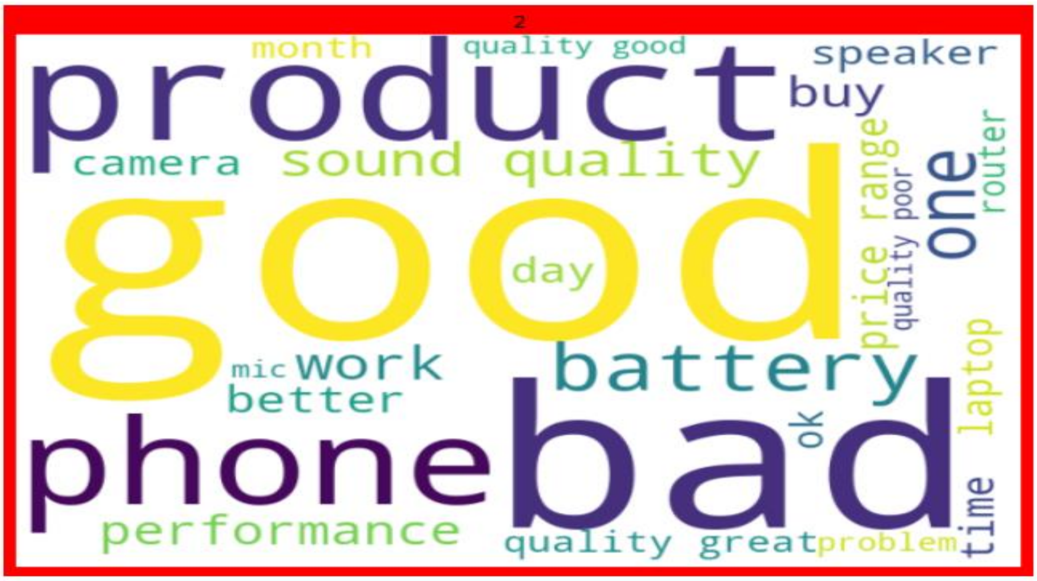


```
from wordcloud import WordCloud
```

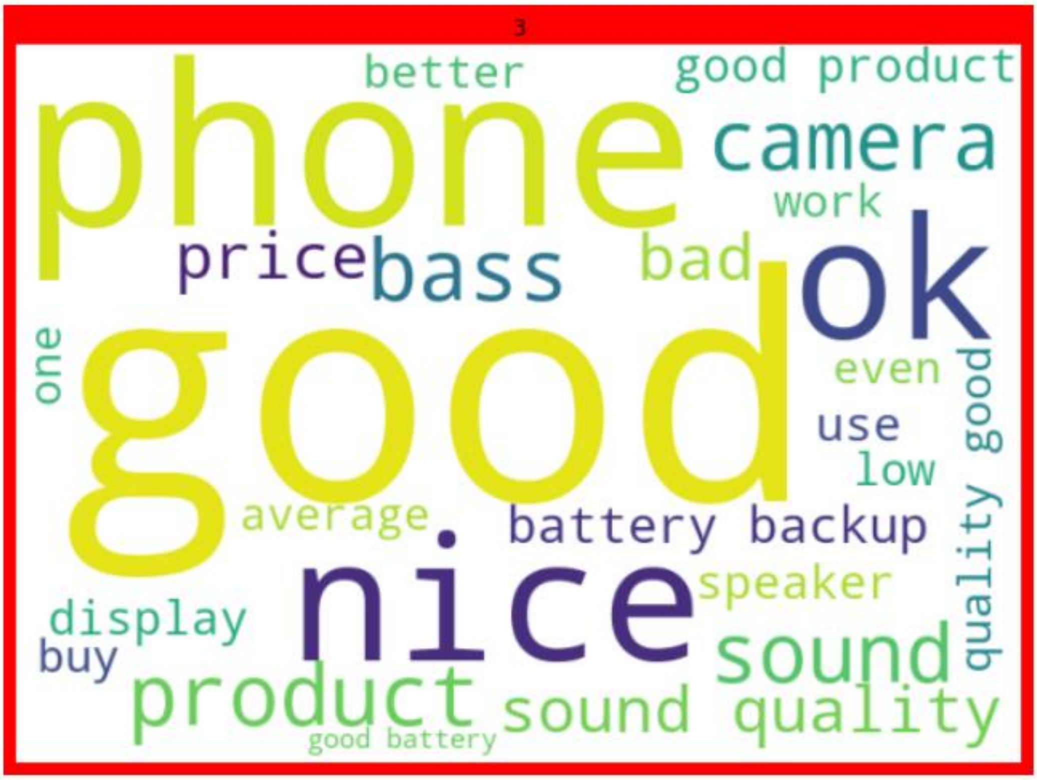
```
for i in range(1,6):
    rev = df['REVIEWS'][df['RATING']==i]

    rev_cloud = WordCloud(width=700,height=500,background_color='white',max_words=25).generate(' '.join(rev))

    plt.figure(figsize=(10,8),facecolor='r')
    plt.imshow(rev_cloud)
    plt.axis('off')
    plt.title(i)
    plt.show()
```



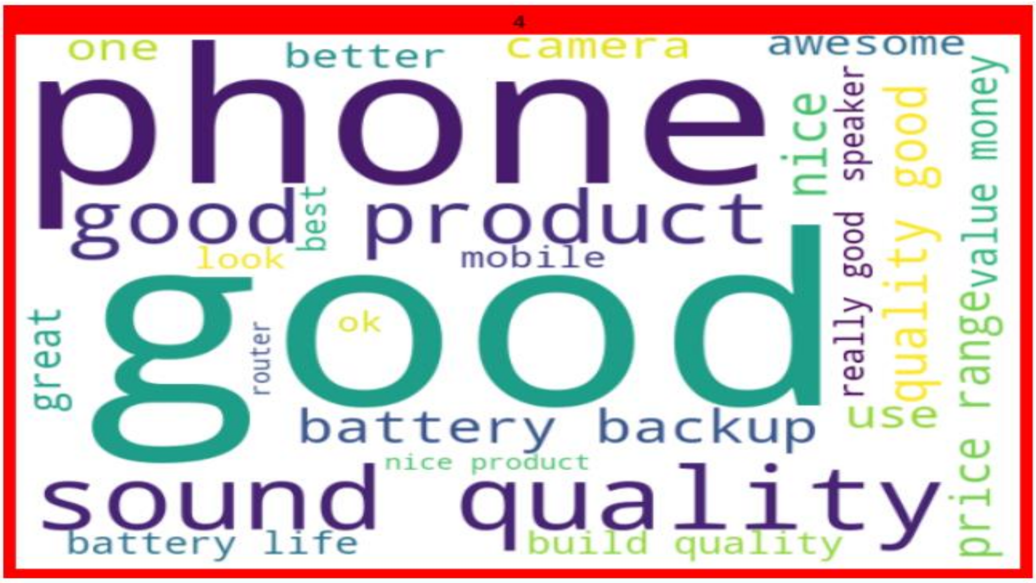
Above we can see 25 most occurring word in review with 1 star rating.



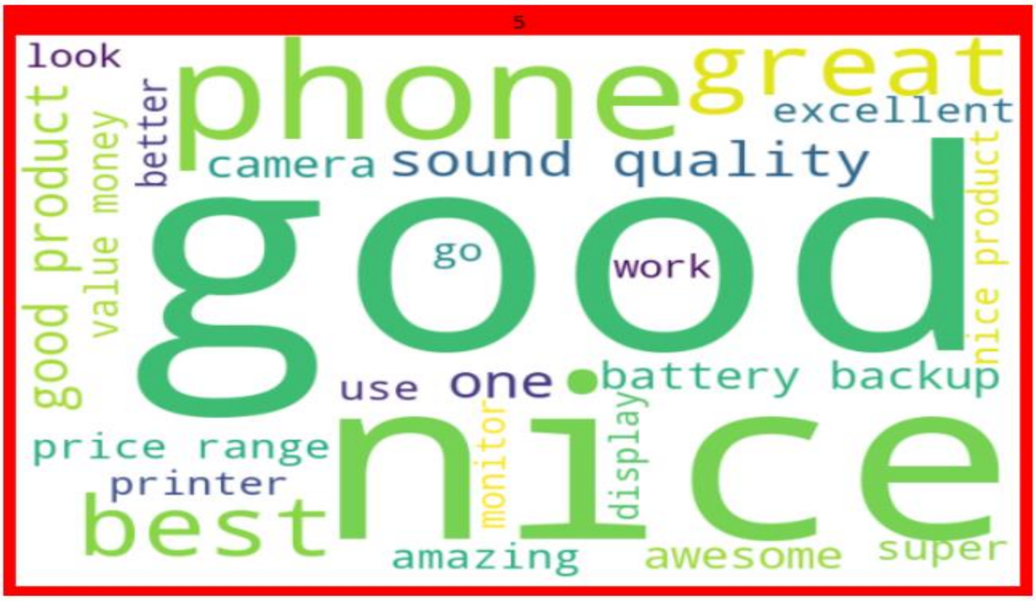
Above we can see 25 most occurring word in review with 2 star rating.



Above we can see 25 most occurring word in review with 3 star rating.



Above we can see 25 most occurring word in review with 4 star rating.



Above we can see 25 most occurring word in review with 5 star rating.

- Interpretation of the Results

Give a summary of what results were interpreted from the visualizations, pre-processing and modelling.

'phone','monitor','camera','laptop','router','printer','product', 'speaker' are repeating in the reviews. We can drop them as rating does not depend on type of product.

CONCLUSION

- Key Findings and Conclusions of the Study

  Reviews with 4 & 5 rating are comparatively longer in length.

  Insignificant words - 'phone', 'monitor', 'camera', 'laptop', 'router', 'printer', 'product', 'speaker' are repeating in the reviews. Since rating is not related to type of product.

- Learning Outcomes of the Study in respect of Data Science

  Space vector classifier works the best. Even thought I try to balance the data at scrapping stage, still the data was not balanced properly which affected the accuracy of the model.

- Limitations of this work and Scope for Future Work

  Taking reviews and ratings from Amazon website and with better balanced data, I want to train the model again to increase its accuracy.