RATING PROJECT

SUBMITTED BY:

NEETAL TIWARI

INTRODUCTION

A lot of consumers, when searching online for something to buy, will take a look at an online review or rating for a product. It seems like a great way to get an unfiltered view on quality but research indicates most online reviews are too simple and may misguide consumers.

According to one united states survey, 78.5% of American consumers looked for information online about a product or service, and 34% had posed an online review. A global Nielsen survey, found 70% of consumers trust online product review and use them in making decisions.

As a result, the average user rating of products has become a significant factor in driving sales across many product categories and industries. The proliferation of online reviews from many consumers sounds like a positive development for consumer welfare but some research shows otherwise.

MATHEMATICAL/ ANALYTICAL MODELLING OF THE PROBLEM

STATISTICAL SUMMARY

In [56]:
```
df.describe()
```

Out[56]:

	Review	Rating
count	990	990
unique	9	3
top	Perfect product!	5
freq	198	660

DATA SOURCES AND THEIR FORMATS

I have taken the review and rating data from the online retailer which is flipkart.com. Dataset has 990 rows and 2 columns Review and Rating.

Review is Review of laptop given by the consumers and Rating is the Rating of the laptop given by the consumers.

In [43]:
```
print(len(Review_of_laptops))
print(len(Ratings_of_laptops))
```
```
990
990
```

In [44]:
```
df=pd.DataFrame({"Review":Review_of_laptops, "Rating":Ratings_of_laptops})
df
```

Out[44]:

	Review	Rating
0	Perfect product!	5
1	Classy product	5
2	Pretty good	4
3	Terrific purchase	5
4	Brilliant	5
...	...	...
985	Nice	5
986	Terrific	3
987	Waste of money!	5
988	Value-for-money	4
989	Perfect product!	5

990 rows × 2 columns

DATA PRE-PROCESSING DONE

In Data pre-processing I have used the Label Encoding method to change the objects into integer so that I can analysis the dataset and I can use the algorithm for the prediction.

Label Encoding

In [83]:
```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()
objects=["Review","Rating"]
for i in objects:
    df[i]=le.fit_transform(df[i])

objects
```

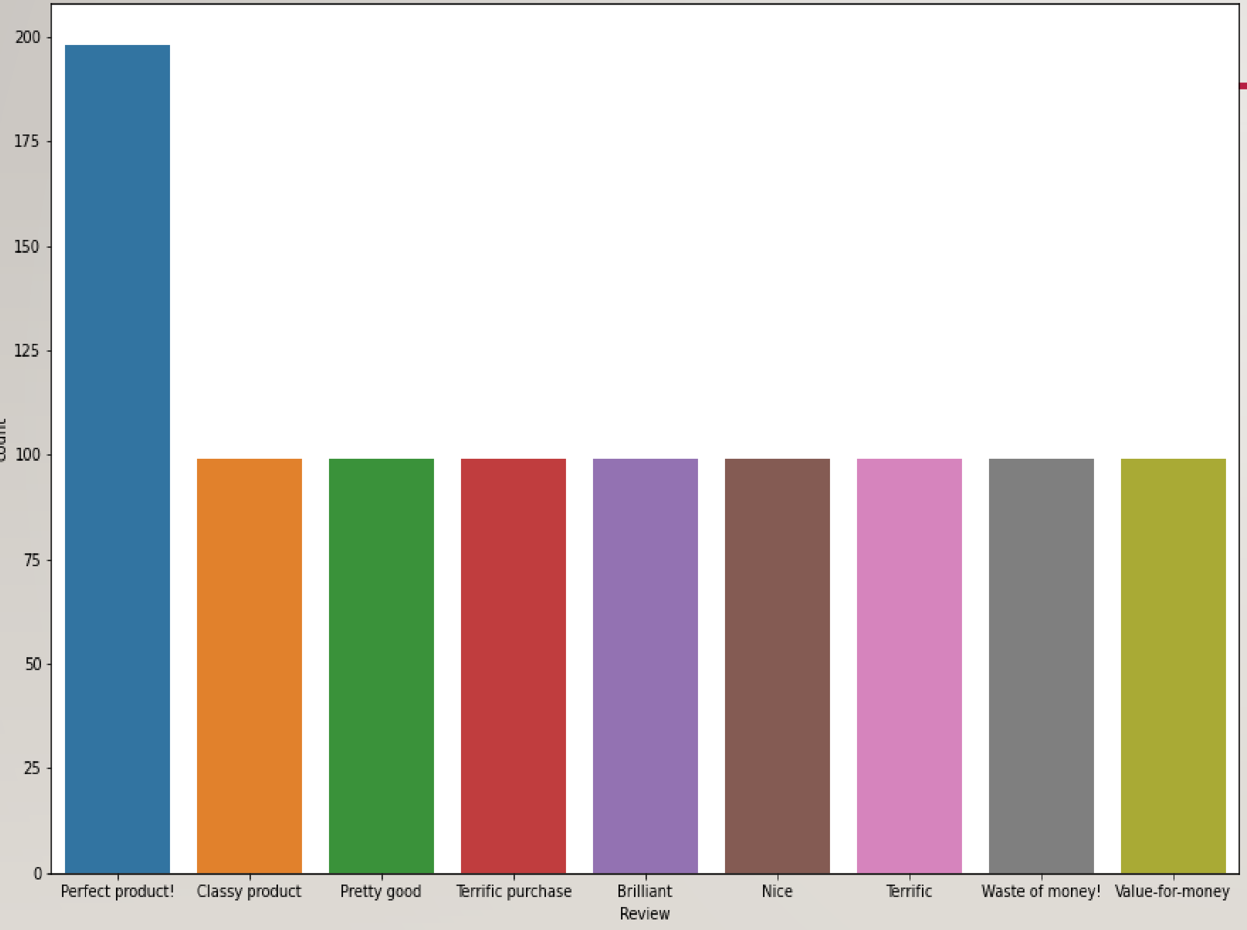
Out[83]:
```
['Review', 'Rating']
```

In [84]:
```
df.head()
```

Out[84]:

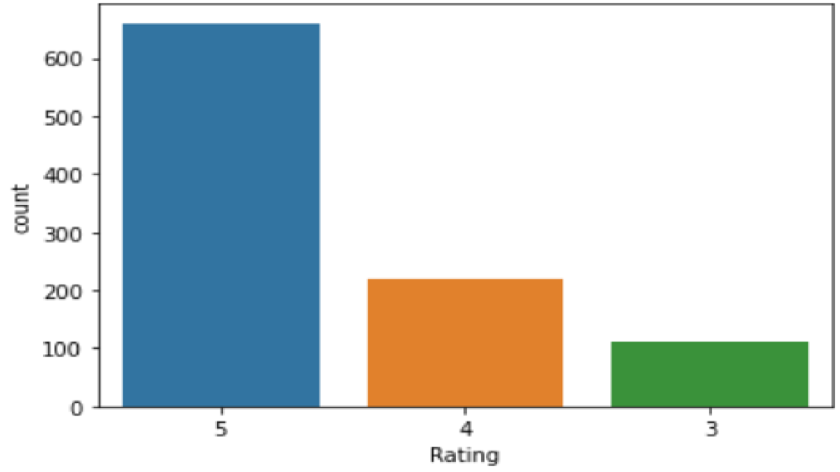
	Review	Rating
0	3	2
1	1	2
2	4	1
3	6	2
4	0	2

VISUALIZATIONS




```
In [64]: df["Rating"].value_counts()
Out[64]: 5    660
         4    220
         3    110
         Name: Rating, dtype: int64

In [65]: sns.countplot(x="Rating",data=df)
Out[65]: <AxesSubplot: xlabel='Rating', ylabel='count'>
```



Perfect product Review is highest, it means review of laptop is good.

5 Rating is highest and 3 Rating is lowest.

MODEL/S DEVELOPMENT AND EVALUATION

LinearRegression

```
In [115]: li=LinearRegression()
          li.fit(x_train,y_train)
          lipred=li.predict(x_test)

          print('Mean absolute error:',mean_absolute_error(y_test,lipred))
          print('Mean squared error:',mean_squared_error(y_test,lipred))
          print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,lipred)))
          print(r2_score(y_test,lipred))
```

```
Mean absolute error: 5.248050955016925e-16
Mean squared error: 3.3992089654405677e-31
Root mean squraed Error: 5.830273548848774e-16
1.0
```

RandomForestRegressor

```
In [116]: rf=RandomForestRegressor()
          rf.fit(x_train,y_train)
          rfpred=rf.predict(x_test)

          print('Mean absolute error:',mean_absolute_error(y_test,rfpred))
          print('Mean squared error:',mean_squared_error(y_test,rfpred))
          print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,rfpred)))
          print(r2_score(y_test,rfpred))
```

```
Mean absolute error: 0.0
Mean squared error: 0.0
Root mean squraed Error: 0.0
1.0
```

In Linear Regression r2 score is 1.0.

In Random Forest Regressor r2 score is 1.0.

MODEL/S DEVELOPMENT AND EVALUATION

1.0

KNeighborsRegressor

In [117]:
```
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
knnpred=knn.predict(x_test)
print('Mean absolute error:',mean_absolute_error(y_test,knnpred))
print('Mean squared error:',mean_squared_error(y_test,knnpred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,knnpred)))
print(r2_score(y_test,knnpred))
```

```
Mean absolute error: 0.0
Mean squared error: 0.0
Root mean squraed Error: 0.0
1.0
```

DecisionTreeRegressor

In [118]:
```
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
dtrpred=dtr.predict(x_test)
print('Mean absolute error:',mean_absolute_error(y_test,dtrpred))
print('Mean squared error:',mean_squared_error(y_test,dtrpred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,dtrpred)))
print(r2_score(y_test,dtrpred))
```

```
Mean absolute error: 0.0
Mean squared error: 0.0
Root mean squraed Error: 0.0
1.0
```

SVR

In [119]:
```
svr=SVR()
svr.fit(x_train,y_train)
svrpred=svr.predict(x_test)

print('Mean absolute error:',mean_absolute_error(y_test,svrpred))
print('Mean squared error:',mean_squared_error(y_test,svrpred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,svrpred)))
print(r2_score(y_test,svrpred))
```

```
Mean absolute error: 0.060865827224416164
Mean squared error: 0.005266855911588853
Root mean squraed Error: 0.07257310735795218
0.9877782197098463
```

In KNeighbors Regressor r2 score is 1.0.

In Decision Tree Regressor r2 score is 1.0.

In SVR r2 score is 0.98%

CONCLUSION

This study mainly reflects the importance of mining online product reviews and also analysing the impact these reviews create on third party sellers. This study is beneficial to both the consumer and the seller. Though this study, it is made clear that the seller reviews also carry equal importance as product review. The seller review does not only mean the reviews on the whole but also the reviews given by the customers to the sellers in the product review itself. The seller should also consider them in order to take further decisions on how and in what areas to improve.

Considering the analysis done in this study, the insights observed/gathered help both the consumer and the seller. The consumer benefits from the fact that instead of going through a lot of reviews in order to know about the pros and cons of the product, this analysis helps him/her to directly view the percentage of positive, negative, and neutral reviews and the relevant frequent words in each category, and also drives home the fact that there are certain topics that the endures can view directly with the corresponding words in each topic. This would help him/her to get an idea of the product in less time. The seller comparison on the whole and on a particular product would also help the consumer to decide whether to select that particular seller or choose another.

From the sellers point to view, in order to improve upon product sales performance, the seller review analysis would be of great help.

This study, by considering seller reviews, whish was not done in previous analysis, help us to not only get a clear idea about the pros and cons of the seller but also to analysis the performance and decide on the seller. The techniques used and the method followed can be utilized for various kinds of the product too in order to analysis the impact the reviews create on third-party sellers.