FLIP ROBO

# RATING PREDICTION

Submitted by:

**NEETAL TIWARI**

# ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my SME (Mohd Kashif) as well as my company (Flip Robo Technologies) who gave me the golden opportunity to do this wonderful project on the (Rating project), which also help me to in doing a lot of research and I came to know about so many things I am really thank to them.

# INTRODUCTION

## Business Problem Framing

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.

## Conceptual Background of the Domain Problem

Consumers use online user ratings because they assume these provide a good indication of product or service quality. For example, you would expect a laptop with an average rating of four out of five star to be objectively better than a laptop with an average rating of three out of five stars, 100% of the time.

In order to test this assumption, one researcher team put together an impressive dataset comprising of 344157 Amazon.com ratings for 1272 products, in 120 product categories. For each product, they obtained objective quality scores from the website Consumer Reports. They also collect data on prices, brand image measures, and two independent sources of resale values in the market for second hand or used goads.

The researchers found that average user ratings correlated poorly with the scores Consumer Reports. For example, when the difference in average user ratings between pairs of products was larger than one star, the item with the higher user rating was rated more favourably by Consumer Reports only about two-third of the time.

In other words, if you were comparing a laptop with an average rating of four out of five stars, with another laptop with an average rating of three out of five stars, the first laptop would only be objectively better 65% (not 100%) of the time. This is a far cry from a sure difference in quality. Moreover, the average user ratings did not predict resale value in the used-product marketplace.

## Review of Literature

## Motivation for the Problem Undertaken

There is a way to use the information from review and ratings despite all of these potential pitfalls. First, look for product with a high average user rating, many reviews, and not of variance in the rating scores. Beware placing too much faith in average ratings that are based on few reviews and with high variance in the ratings.

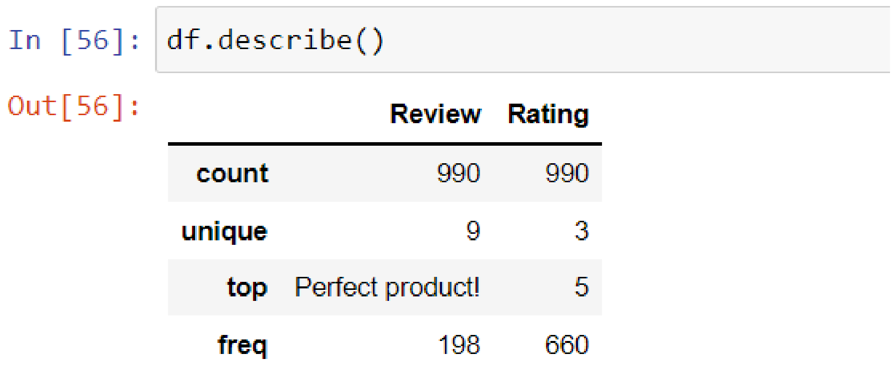
You can also consider online reviews in light of additional sources that provide objective product evaluations, from technical experts. Sources of this kind of information include Consumer Report, Choices, Consumers Union.

Where possible, you can consider employing technology designed to help you navigate the bias in online reviews.

## Analytical Problem Framing

## Mathematical/ Analytical Modelling of the Problem

### STATISTICAL SUMMARY

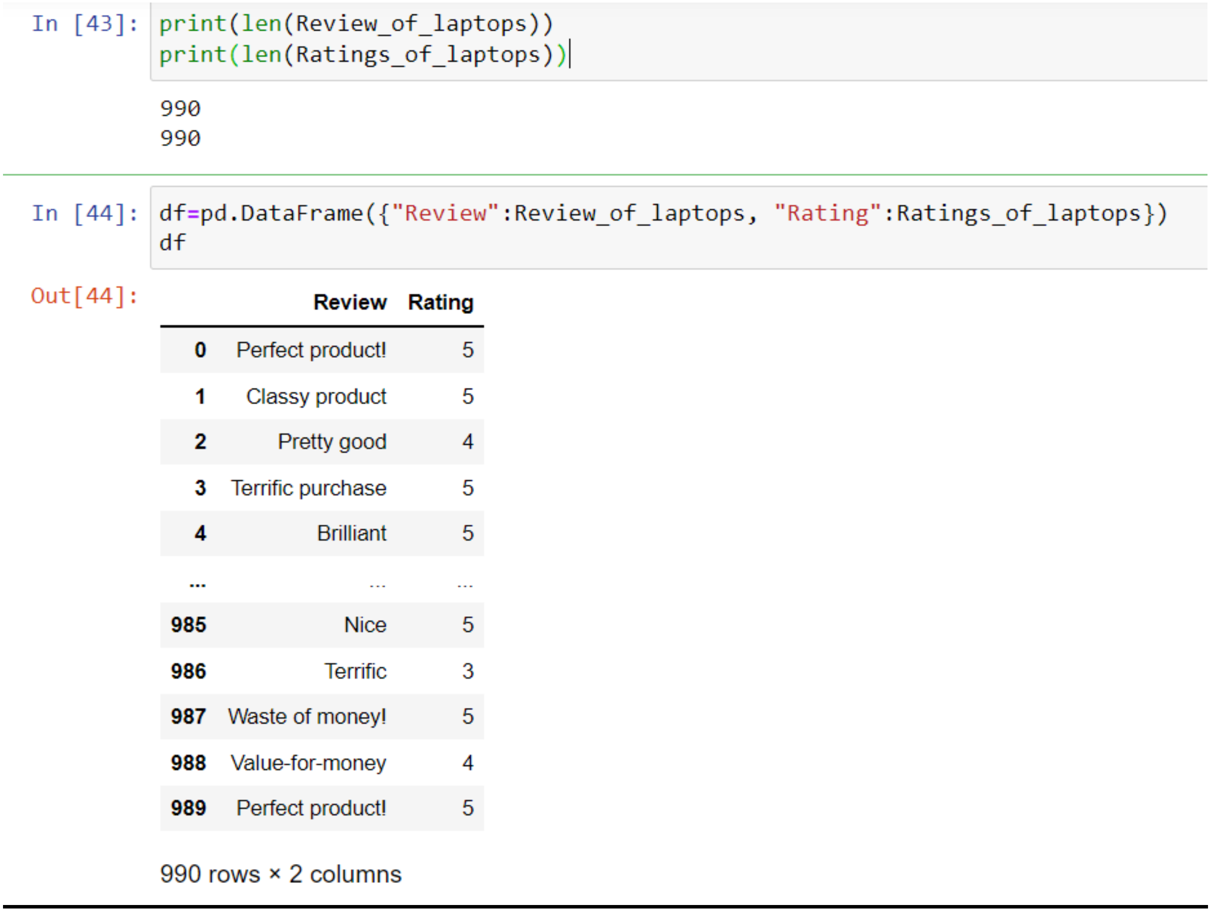


```
In [56]: df.describe()
Out[56]:
                  Review   Rating
   count             990      990
  unique               9        3
     top Perfect product!       5
    freq             198      660
```

## 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
```

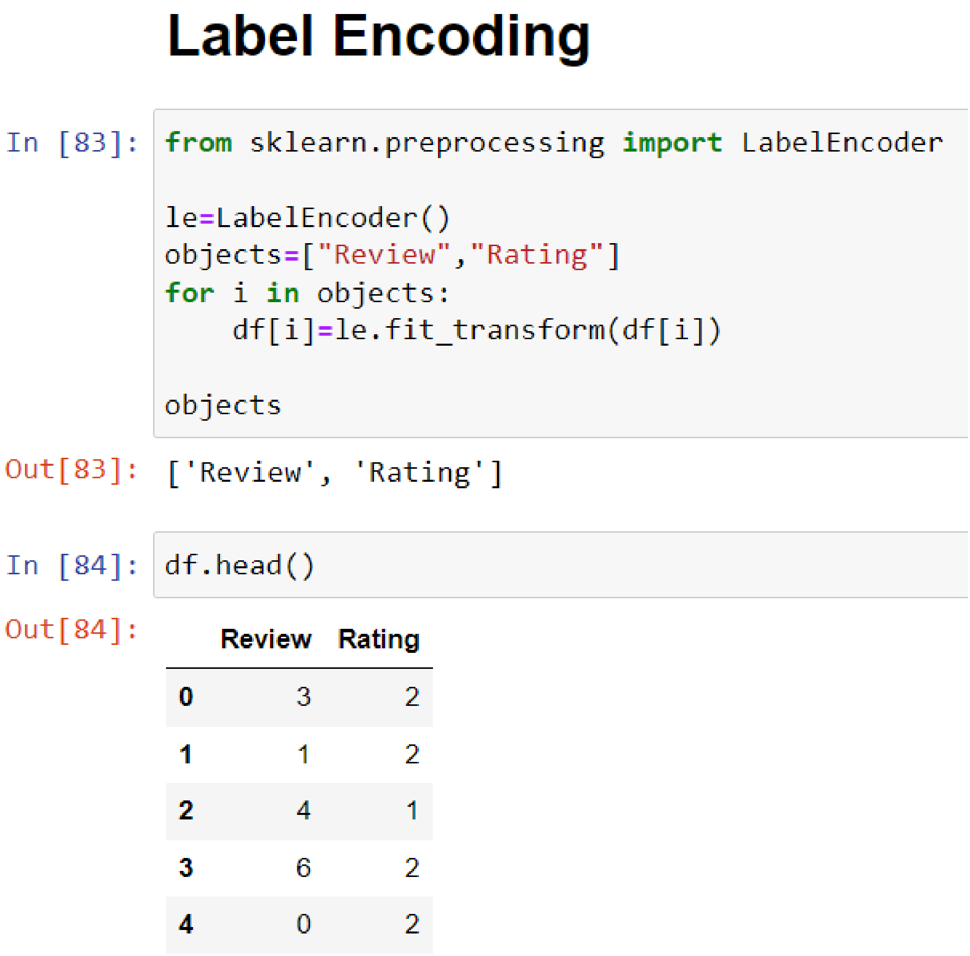
Out[44]:

| | 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 Preprocessing Done

In Data preprocessing 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 |

## Data Inputs- Logic- Output Relationships

There is only one input and one output in the dataset. Input directly affects the output because if reviews are good than ratings are also good if reviews are bad than ratings are also bad.

# Model/s Development and Evaluation

## Testing of Identified Approaches (Algorithms)

I had used 5 different algorithms to predict the model. Such as

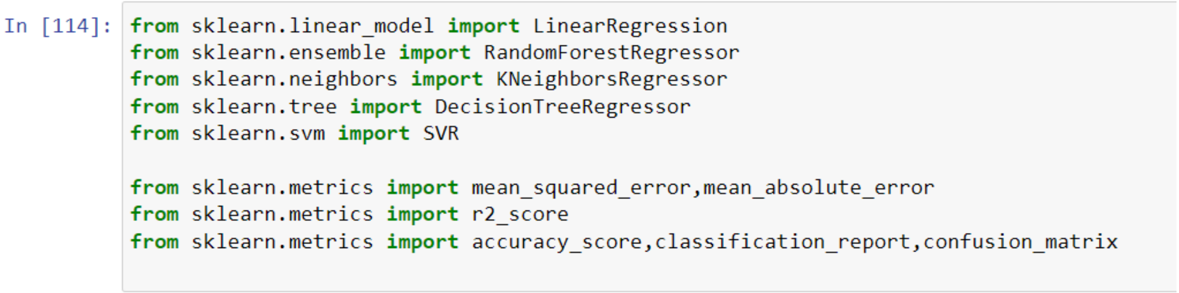
1) Linear Regression

2) Random Forest Regressor

3) KNeighbors Regressor

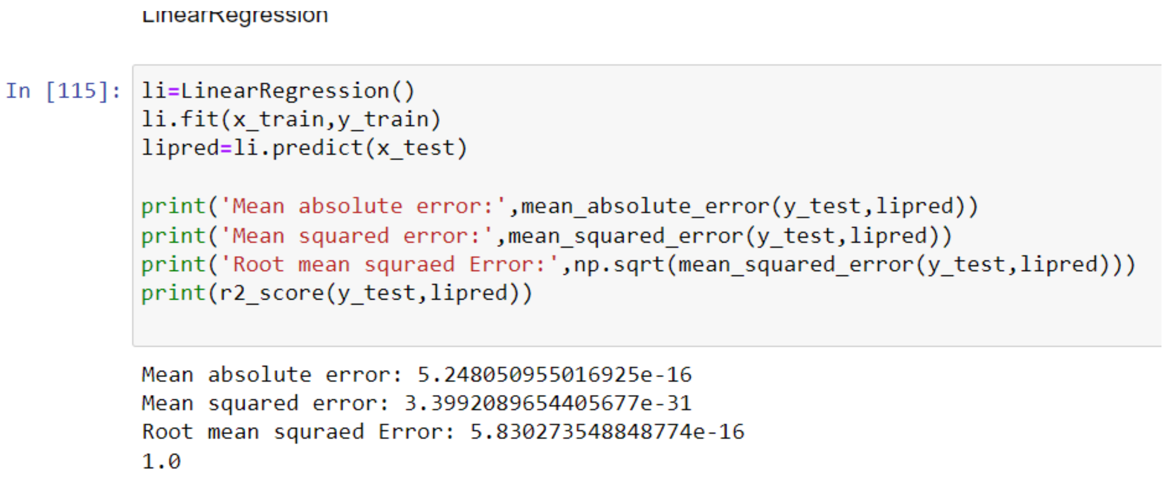
4) Decision Tree Regressor

5) SVR



```
In [114]: from sklearn.linear_model import LinearRegression
          from sklearn.ensemble import RandomForestRegressor
          from sklearn.neighbors import KNeighborsRegressor
          from sklearn.tree import DecisionTreeRegressor
          from sklearn.svm import SVR

          from sklearn.metrics import mean_squared_error,mean_absolute_error
          from sklearn.metrics import r2_score
          from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

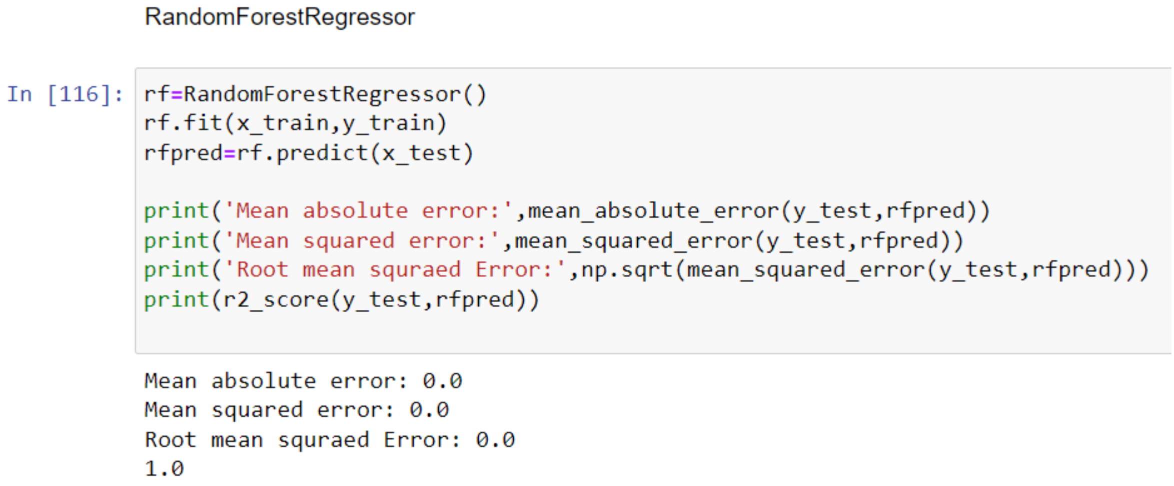
## Run and evaluate selected models



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
```

In Linear Regression r2 score is 1.0.

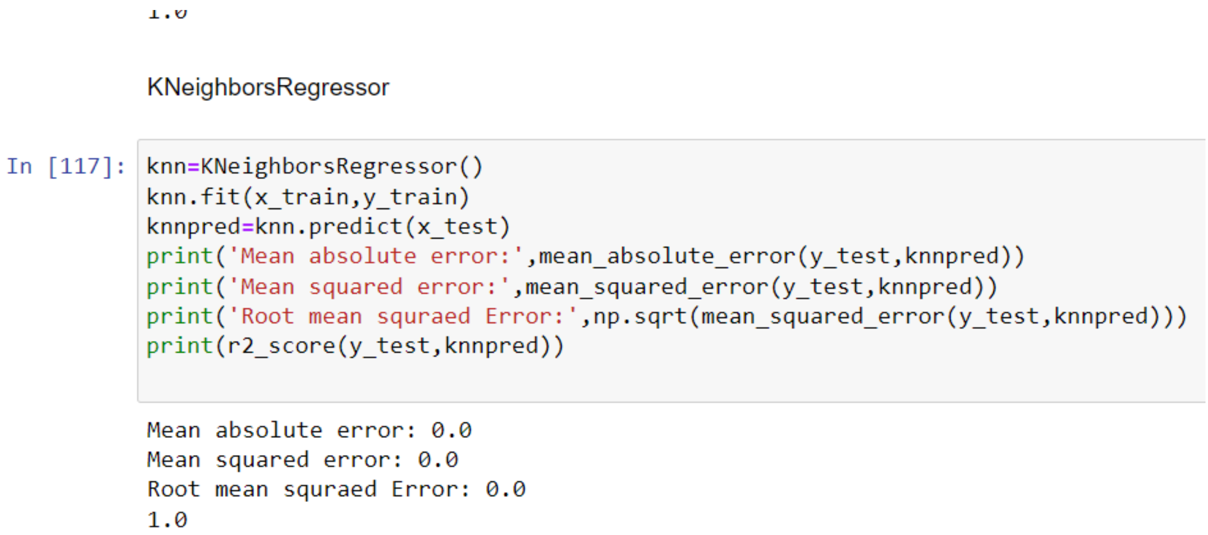


RandomForestRegressor

```
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 Random Forest Regressor r2 score is 1.0.



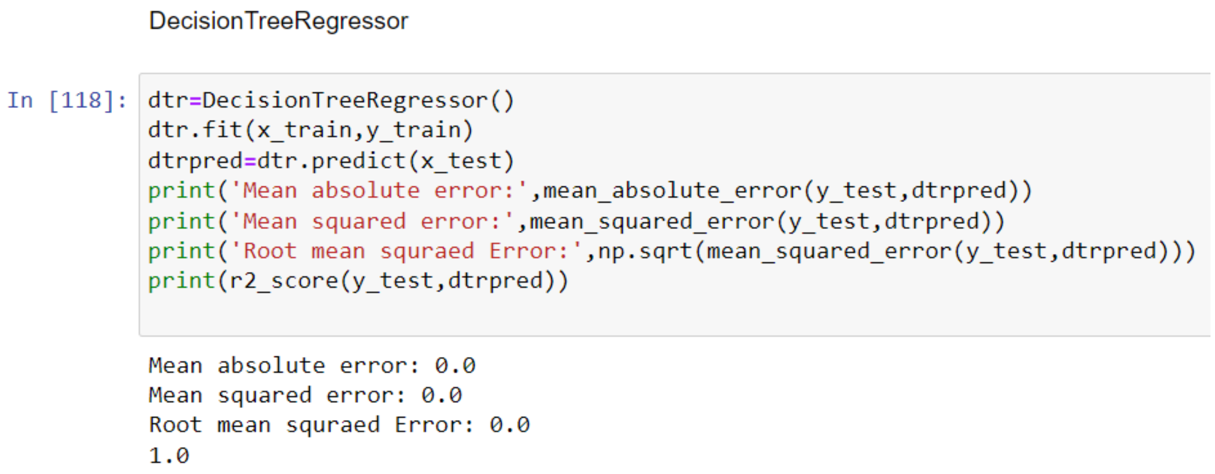
```
1.0
```

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
```

In KNeighbors Regressor r2 score is 1.0.

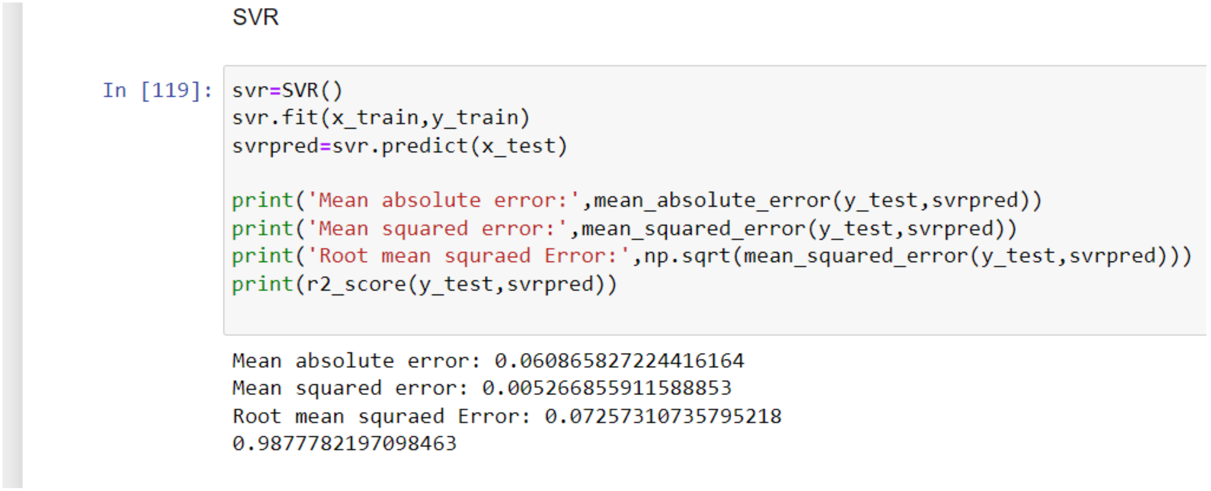


DecisionTreeRegressor

```
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
```

In Decision Tree Regressor r2 score is 1.0.



SVR

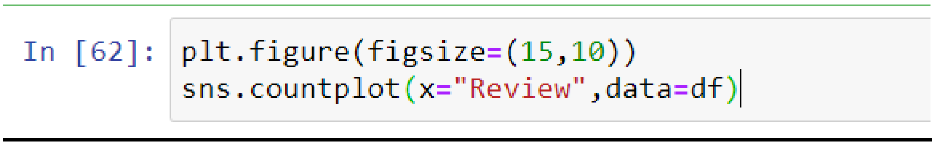
```
In [119]: svr=SVR()
          svr.fit(x_train,y_train)
          svrpred=svr.predict(x_test)

          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
```

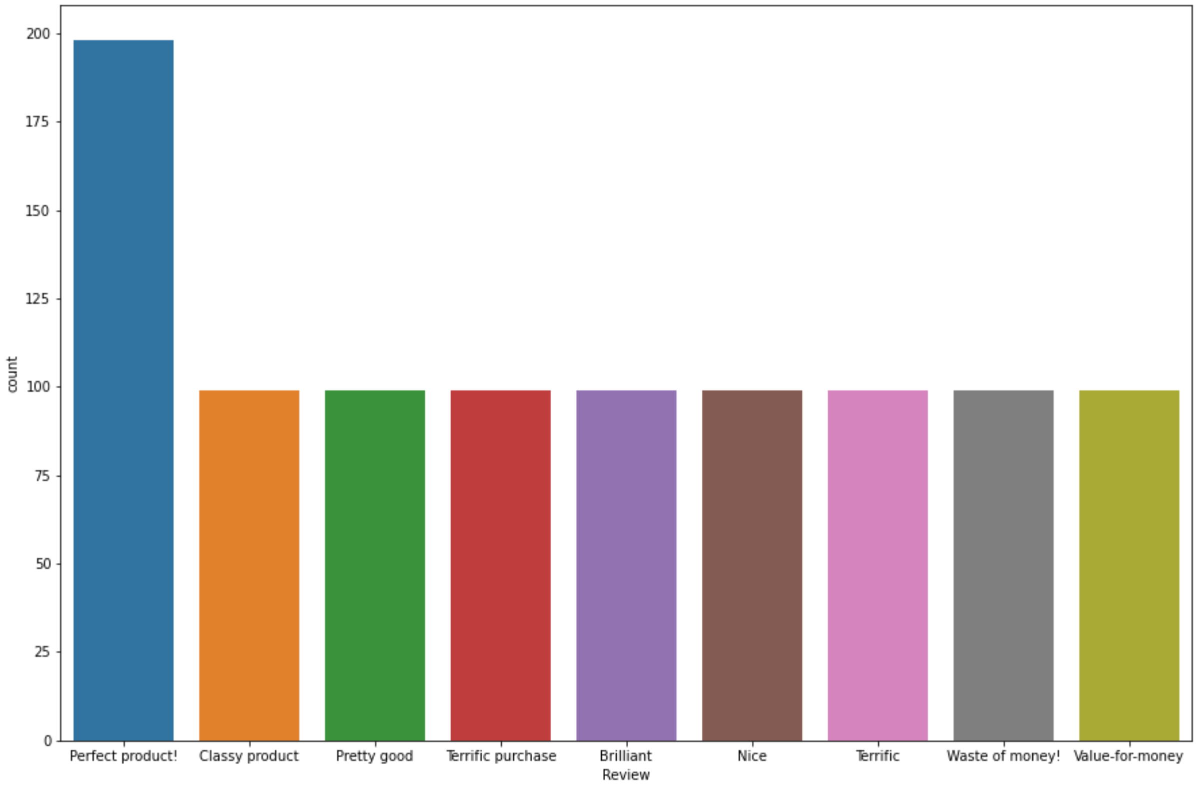
In SVR r2 score is 0.98%

All algorithms are working so well in this model.

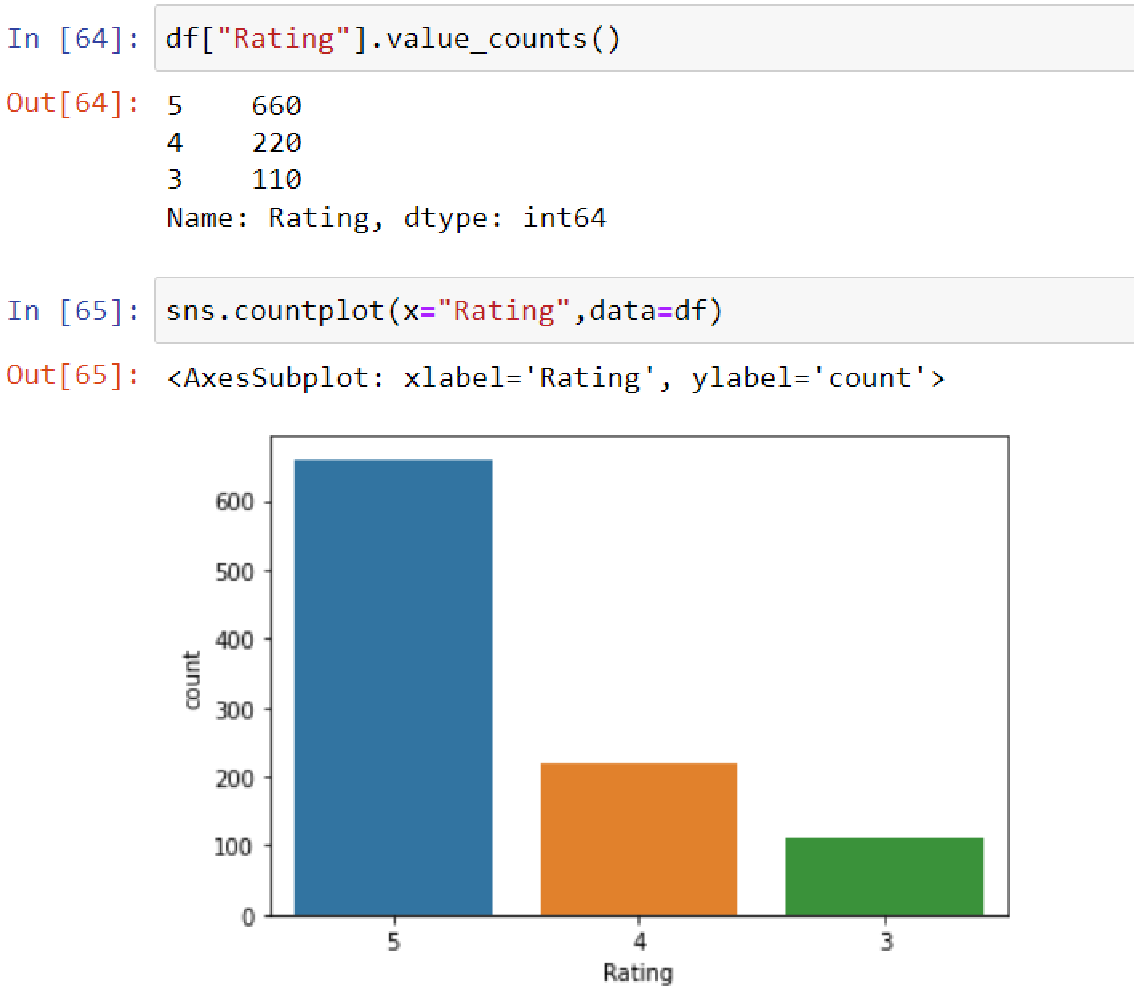
# Visualizations



```
In [62]: plt.figure(figsize=(15,10))
         sns.countplot(x="Review",data=df)
```



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



```
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'>
```

5 Rating is highest and 3 Rating is lowest.

# CONCLUSION

## Key Findings and Conclusions of the Study

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.

## Learning Outcomes of the Study in respect of Data Science

Although sentiment analysis was used in this study, it is not limited to it. Many others method in NLP can be implemented for analysing and understanding online reviews. Some of them include summarizing reviews into a paragraph or bullet points, identifying which product are popular by extracting entities from reviews, and identifying emerging trends based on the timestamp of the reviews. Speech recognition tools and chatbots can also be created in order to answer the specific question of customers. This study can also be extended to creating a recommendation model for consumers based on their previous purchases. Finally, though the future looks extremely challenging, there are many advancements in this discipline (NLP and also machine learning in particular), and in the coming years it is very likely that these developments would make complex applications look possible.

## Limitations of this work and Scope for Future Work

Though everything seems to be in place and the process seems to be easy, there are certain limitations in this study.

One limitation of this study is transparency. When a consumer purchases a product and writes a review only mentioning the product but not the seller, online retailer like Amazon would have the control to display the sellers name under the title of the review. Though they mark the review with a Verified Purchase tag, mentioning the seller's name also would benefit the consumer.

Though the recommendation of the seller by Amazon is based on many factors such as ratings, the seller list is not clearly visible and a small link is available to view the other sellers. This causes a huge impact on the other third-party sellers who go unnoticed. Also, many sellers like Cloudtail and Appario are owned by Amazon or have a stake in them. These sellers are usually recommended by Amazon for the majority of the product. Though it appears that a giant retailer like Amazon provides a platform wherein there are equal opportunities for third-party sellers, this does not seem to be the actual case. Due to all such reasons, people usually do not notice the third-party sellers. Since the purchases are less from these sellers, the ratings and reviews are even lower when compare to the best seller.