Ratings Prediction

Submitted by:

VARSHA PATANKAR

ACKNOWLEDGMENT

Various references were used for the implementation of this project. Different research papers by experts were referred for the best method and for the best result. Different sites also referred for this project which are useful for data science projects like thesai, towardsdatascience and so on.

INTRODUCTION

- ## Business Problem Framing

  We have a client who has a website where people write different reviews for technical products. Now they are adding 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. So, we have to build an application which can predict the rating by seeing the review.

- ## Conceptual Background of the Domain Problem

  In their research process, consumers want to find useful information as quickly as possible. However, searching and comparing text reviews can be frustrating for users as they feel submerged with information. Indeed, the massive amount of text reviews as well as its unstructured text format prevents the user from choosing a product with ease.

  The star-rating, i.e. stars from 1 to 5 on any website, rather than its text content gives a quick overview of the product quality. This numerical information is the number one factor used in an early phase by consumers to compare products before making their purchase decision.

- ## Motivation for the Problem Undertaken

  The main objective of this project is to collect the data from different websites for various products to do the predictions based on reviews. So they have to see the previous data of each customer and check their needs and online platform for purchasing. So we have collected the data of from various online shopping websites.

Analytical Problem Framing

- ## Mathematical/ Analytical Modelling of the Problem

  The general problem one faces in the modelling is that to find best suitable algorithm and mathematical and statistical approach for our model. In real world the datasets are very large, noisy and unstructured. So to find the best strategy for our model we need such approach that combines the statistics and machine learning together.

  Statisticians are using so many approaches to find the best accuracy and we are also trying some of the machine learning algorithms along with the statistics. But in this project our main focus is to do the prediction of the ratings based on the reviews. Data analysis is the important factor to see the details of given dataset.

- ## Data Sources and their formats

  There are so many sources are available on the internet from which can get the data on which we have to work on. So for this project you can get the data from bitstream, towardsdatascience etc.

  The data you found can be in any form i.e. csv file, excel file etc.

  Here I have got the data in the form of excel file which we have to import that file using pandas library. Below is the snapshot of data and how I have load the data for further process.

```
#lets import pandas library to read the data
import pandas as pd
df = pd.read_excel('Combined.xlsx')
df
```

	Sr. No.	Site	Product	Model Name	Rating	Reviews
0	1	FlipKart	Laptop	HP 14s Core i5 11th Gen - (8 GB/512 GB SSD/Win...	4.3240	18
1	2	FlipKart	Laptop	MSI GF63 Thin Core i5 9th Gen - (8 GB/512 GB S...	4.4712	97
2	3	FlipKart	Laptop	HP Pavilion x360 Core i3 11th Gen - (8 GB/256 ...	4.4271	27
3	4	FlipKart	Laptop	Lenovo IdeaPad Flex 5 Ryzen 7 Octa Core 5700U ...	4.7300	0
4	5	FlipKart	Laptop	Lenovo Ideapad S145 Ryzen 3 Dual Core 3200U - ...	3.9500	749
...	...	...	...	...	...	...
12029	12030	ebay	Router	D-Link RangeBooster N DIR-628 54 Mbps 4-Port ...	4.5000	19

- Data Pre-processing Done

  There is some process that we have to do while pre-processing. Following are the process that we have used-

1. **Data Wrangling:** Our dataset is not always in proper format. Sometimes there maybe some missing values, sometimes there must be out of the box values. Our data is always a raw data so for that we have to do some wrangling process like.
   a. Check if there are any null values are present in the dataset or not.

```
df.isnull().sum()
```

```
Sr. No.          0
Site             0
Product          0
Model Name       0
Rating         112
Reviews          0
dtype: int64
```

   There are 112 null values present in our dataset so we have to fill that values.

   b. Check the data type of each column.

```
pd.options.display.max_info_columns = 5
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12034 entries, 0 to 12033
Columns: 6 entries, Sr. No. to Reviews
dtypes: float64(1), int64(2), object(3)
memory usage: 564.2+ KB
```

   c. Check the summary of our dataset using

```
df.describe()
```

	Sr. No.	Rating	Reviews
count	12034.000000	11922.000000	12034.000000
mean	6017.500000	4.629389	328.451720
std	3474.060904	0.312361	1285.077686
min	1.000000	3.740000	0.000000
25%	3009.250000	4.500000	4.000000
50%	6017.500000	4.500000	20.000000
75%	9025.750000	5.000000	94.000000
max	12034.000000	5.000000	14203.000000

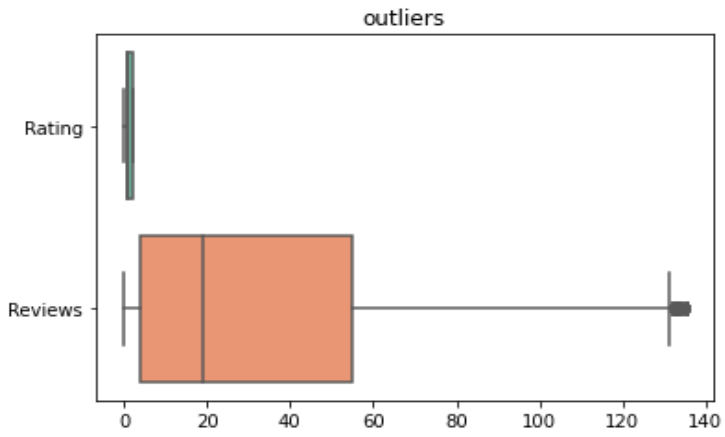
From above we can say that all values for each column is different.

1. For Ratings column max value is 5 and 75% value is 5. So we can say that in such case data is ok.

2. But for the Review column max value is 14203 and 75% value is 94 so in this case we can say that some outliers are present in this column.

Checking Outliers: Our data is a raw data so we have to check whether any value is out of the box or not. If any outliers are present in any column, so in that case we have remove that particular value from our dataset. Otherwise it will give us wrong predictions.

```
sns.boxplot(data=df[['Rating','Reviews']],orient='h',palette='Set2')
plt.title('outliers')
plt.show()
```



- Data Inputs- Logic- Output Relationships

  In this step we have to do the analysis of each column using univariate method. And analysis of each column with target variable using bivariate method. In this method suppose we have any target variable then we check each column with our target variable whether they have proper correlation or not. But in this no target variable is present, so we have consider two target variables 'gender' and 'age' as per data given.

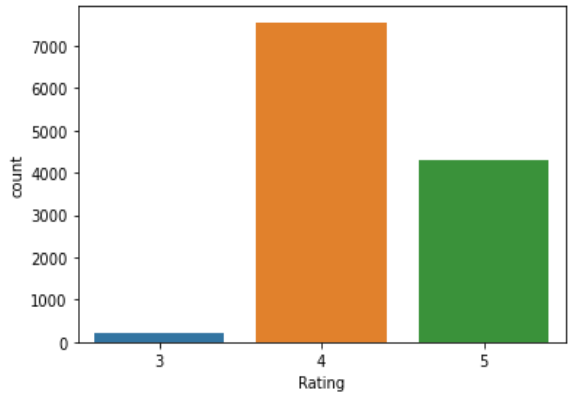
```
df['Rating'].value_counts()
```

```
4    7555
5    4283
3     196
Name: Rating, dtype: int64
```

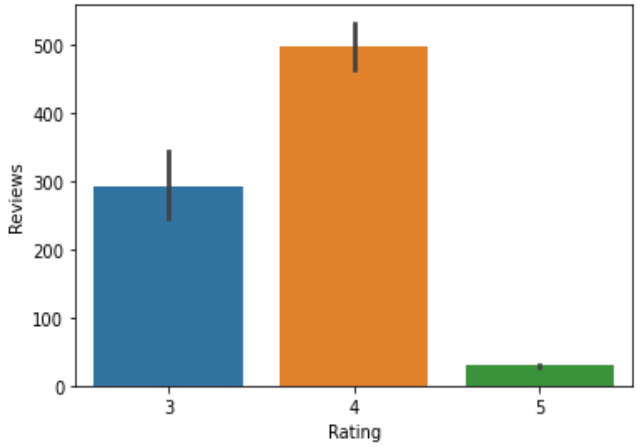
```
df['Reviews'].value_counts()
```

```
1       1412
5        756
2        660
3        520
4        499
        ...
665       20
1133      20
257       20
209       20
1075      20
Name: Reviews, Length: 136, dtype: int64
```

```
#lets see the label graphically to understand more clearly using some visualisation libraries
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x='Rating',data=df)
plt.show()
sns.countplot(x='Reviews',data=df)
plt.show()
sns.countplot(x='Site',data=df)
plt.show()
sns.countplot(x='Product',data=df)
plt.show()
```




```
sns.barplot(x='Rating',y='Reviews',data=df)
plt.show()
```



Now we have to find the correlation between all variables.

```
df.corr()
```

	Site	Product	Rating	Reviews
Site	1.000000	-0.133894	0.411921	-0.549836
Product	-0.133894	1.000000	0.050788	0.004930
Rating	0.411921	0.050788	1.000000	-0.367349
Reviews	-0.549836	0.004930	-0.367349	1.000000

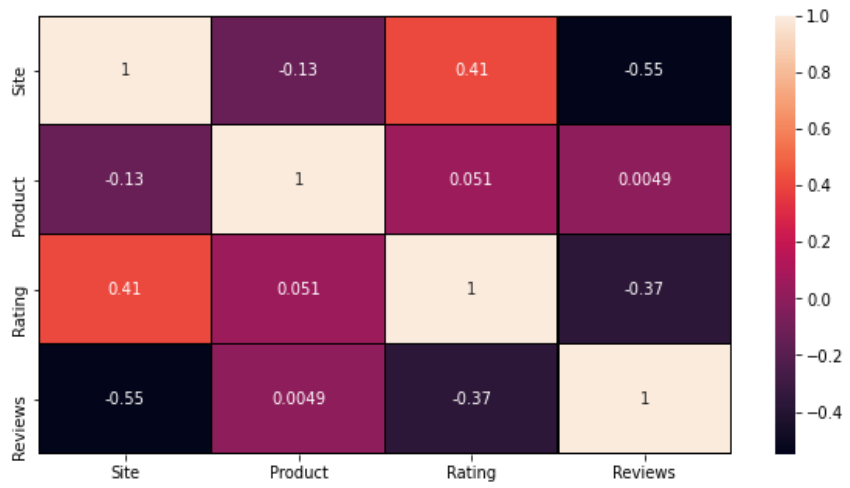
This above method is used to check the correlation between all columns. Heatmap method gives the graphical visualization from which we can easily understand the correlation.

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

  This above method is used to check the correlation between all columns. Heatmap method gives the graphical visualization from which we can easily understand the correlation. So from the diagram below we can say that-

```
#Lets check the correlation using heatmap for better understanding
plt.subplots(figsize=(10,5))
sns.heatmap(df.corr(),linewidths=.1,linecolor='black', annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2029bebfbb0>
```



Some columns are making good positive correlation with other columns and some has negative correlation. So the variables that doesn't make any good correlation with any variable so we to drop that column from the dataset.

- Hardware and Software Requirements and Tools Used

  No external hardware required for this project. Only you have to do in laptop.

  Jupyter notebook with any latest version is required as software.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Machine learning helps to solve the all above problems. It will not only used for the recent data but also helpful to predict the future outcomes with the help of past outcomes.

  Machine learning algorithms help to create a new model using existing and historical data that we can use for the training and testing our model to predict the future outcome.

  In this project we have used various machine learning algorithms to predict the proper outcome which gives as much as accuracy for our model.

- Testing of Identified Approaches (Algorithms)

Following algorithms we have used for our model which gives better performance for the dataset.

1. Decision tree classifier – Decision tree is a powerful algorithm in machine learning which can used when working on the real world datasets.

   Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

- Run and Evaluate selected models

  Following are the methods used for our model-

```
from sklearn.tree import DecisionTreeClassifier
dc=DecisionTreeClassifier()
dc.fit(x_train,y_train)
dc.score(x_train,y_train)
pred=dc.predict(x_test)
print(accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
82.18677301429047
[[  43    0    0]
 [   0 1576  295]
 [   0  241  854]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        43
           1       0.87      0.84      0.85      1871
           2       0.74      0.78      0.76      1095

    accuracy                           0.82      3009
   macro avg       0.87      0.87      0.87      3009
weighted avg       0.82      0.82      0.82      3009
```

From above diagram we can say that decision tree method gives the accuracy above 82%.

2. Random forest classifier- The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

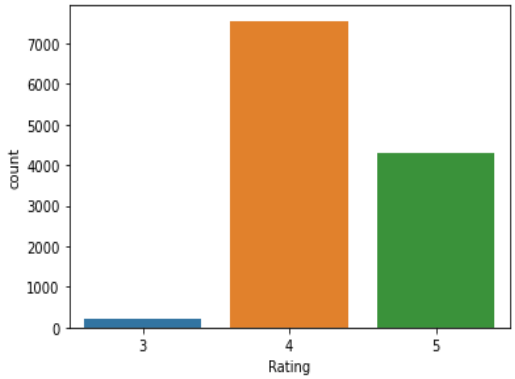
```
from sklearn.ensemble import RandomForestClassifier
rc=RandomForestClassifier()
rc.fit(x_train,y_train)
rc.score(x_train,y_train)
pred=rc.predict(x_test)
print(accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
82.25324027916251
[[  43    0    0]
 [   0 1572  299]
 [   0  235  860]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        43
           1       0.87      0.84      0.85      1871
           2       0.74      0.79      0.76      1095

    accuracy                           0.82      3009
   macro avg       0.87      0.88      0.87      3009
weighted avg       0.83      0.82      0.82      3009
```

From above diagram we can say that Random forest classifier method gives the accuracy above 82%.

- Visualizations

```
#lets see the label graphically to understand more clearly using some visualisation libraries
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x='Rating',data=df)
plt.show()
sns.countplot(x='Reviews',data=df)
plt.show()
sns.countplot(x='Site',data=df)
plt.show()
sns.countplot(x='Product',data=df)
plt.show()
```



Seaborn and Matplotlib are two visualization libraries that I used to visualize graphically my data.

This is the bivariate method because I have taken two variables one is target variable (rating) and other variable. In this way we have to visualize all variables with other variable.

- Interpretation of the Results

From all above process we can say that whatever algorithms or method we used for prediction of our model all process with visualization is very beneficial. Using any visualization library it is easy for us to understand the all data.

CONCLUSION

- Key Findings and Conclusions of the Study

1. We have checked the details of each column and check whether any null values present in our dataset or not. So we have found that our data is proper and no null values or any other different values present in our dataset.
2. Then we have check for the data type of each column, all values are in integer format except city column. So we have converted that object type into numeric values.
3. Next step is important one to do analysis. Here we have done two analysis processes, one is univariate. In this process we have checked each column one by one and see the count of each element present in that column. We have also used seaborn and matplotlib two visualization methods to see the data graphically for better understanding.
4. Other analysis method is bivariate method. In this method suppose we have any target variable then we check each column with our target variable whether they have proper correltion or not.
5. Then lastly we have checked the correlation of all columns with each other. Graphically it shows better result. So from above heatmap diagram we can say that there are some columns which make good correlation with other columns.

- Learning Outcomes of the Study in respect of Data Science

1. In the experimentation different machine learning algorithms were employed for the proposed model development and their performances were evaluated on various parameters.

2. From all above process we can say that whatever algorithms or method we used for prediction of our model all process with visualization is very beneficial. Using any visualization library it is easy for us to understand the all data.

3. For this data Randomforest classifier and decision tree classifier proves the best and others we have used also has good result.