NAME OF THE PROJECT
Rating-Prediction-Project

INTRODUCTION**

. This dataset is collected from amazon and flipkart Ecommerce website through webscrapping
. this dataset has two variables and 39000 rows
. This data is related to reviews and rating of e-commerce product. when a customer buys an item from ecommerce, how satisfied he is with that item, he gives reviews and ratings about it on e-commerce website. while ratings show how satisfied the customer is with the item. 5-star rating show more satisfy, 1-star rating show absolutely not satisfied with products

Business Problem Framing

In Today's time, there has been a lot of growth in the number of ecommerce companies in the market and along with this, in some 4 to 5 years, the number of customers has also increased a lot towards online shopping. Sometimes customers get confused
as to which e-retailer the products are from it would be good to buy, then here the reviews and ratings of the products of the
e-retailer helps the customers in choosing the e-retailer for online shopping. Which means that the ecommerce whose ratings and reviews are more positive, then customers like to buy their products more

Review of Literature

. I checked the reviews and ratings of many customers on many ecommerce websites, which customers themselves write about
the products, so I found that it is difficult to predict the rating of the products exactly to the reviews.

Analytical Problem Framing

Data Sources and their formats

. This dataset is collected from amazon and flipkart Ecommerce website through Web scrapping
. format of the data is in csv and it is text data
It is the snapshot of dataset

```
print(data.shape)   ## Dataset shape

data.sample(12)
```

```
(39360, 2)
```

	review	rating
4546	[']	5-star
20486	[']	1-star
31536	Voice quality is good. But carry option is not...	4-star
9790	["Please I request don't by this product phone...	1-star
31109	Poor bass,Bass is to poor	3-star
18627	['यह घड़ी बहुत सुंदर है और इसके फीचर्स बहुत अच...	5-star
22427	["Watch hangs at least twice in a day. It woul...	3-star
32526	Worth buying but little hassel to use it every...	3-star
20337	[']	1-star
28138	good.black ink not working properly	3-star
38581	Best touch screen or connectivity..Nice quality	4-star
2207	[']	5-star

Data Preprocessing Done

cleaning the dataset by Using nltk and regex library. describe the word-length of the data.
Visualize the reviews text with WordCloud according rating wise
vectorization the text data with Tf-idf

Data Inputs- Logic- Output Relationships

Input data ('review') is not properly related or distributed with all rating's values wise.
As if a reviews has 5-star ratings and the almost same review also has 4-star ratings

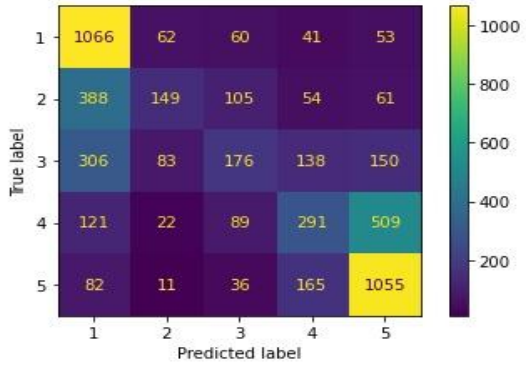
Model/s Development and Evaluation

Run and Evaluate selected models

. I did build a model on this dataset by using MultinomialNB, RandomForestClassifier and ExtraTreesClassifier
. First, we built a model with taking target variable then came about 50 percent

```
:  #               RANDOM FOREST CLASSIFIER
   rf= RandomForestClassifier(n_jobs=-1,oob_score=True,n_estimators=200)
   rf.fit(x_train,y_train)
   pred= rf.predict(x_test)
   print('accuracy_score',accuracy_score(y_test,pred))
   print('classification_report',classification_report(y_test,pred))

   # PLOT CONFUSION MATRIX
   cm= confusion_matrix(y_test,pred, labels= rf.classes_)
   disp= ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=rf.classes_)
   disp.plot()
   plt.show()
```

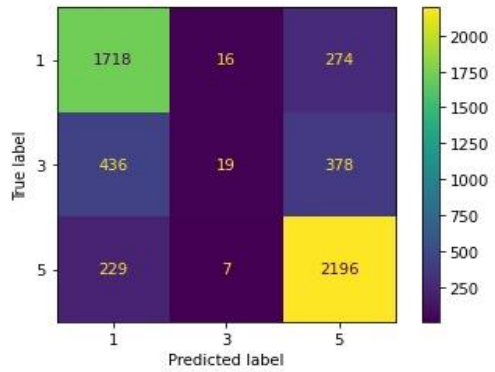
```
accuracy_score 0.5190593589986725
classification_report               precision    recall  f1-score   support

           1       0.54      0.83      0.66      1282
           2       0.46      0.20      0.27       757
           3       0.38      0.21      0.27       853
           4       0.42      0.28      0.34      1032
           5       0.58      0.78      0.66      1349

    accuracy                           0.52      5273
   macro avg       0.48      0.46      0.44      5273
weighted avg       0.49      0.52      0.48      5273
```



. After that I convert the multiclass values of the target variable into 3 types values and now built a model again and I got 74 percent accuracy

```
#                          RANDOMFOREST CLASSIFIER
rf= RandomForestClassifier(n_jobs=-1,oob_score=True,n_estimators=200,min_samples_split=4)
rf.fit(x_train,y_train)
pred= rf.predict(x_test)
print('accuracy_score',accuracy_score(y_test,pred))
print('classification_report \n',classification_report(y_test,pred))

# PLOT CONFUSION MATRIX
cm= confusion_matrix(y_test,pred, labels= rf.classes_)
disp= ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=rf.classes_)
disp.plot()
plt.show()
```

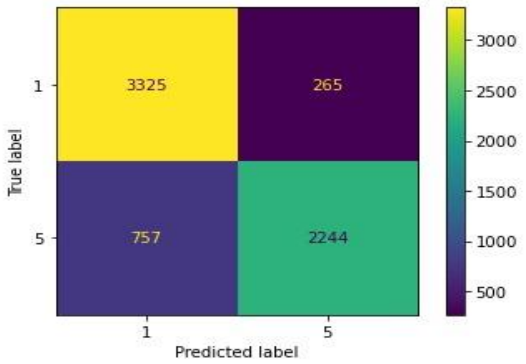
```
accuracy_score 0.7458752133510336
classification_report
              precision    recall  f1-score   support

           1       0.72      0.86      0.78      2008
           3       0.45      0.02      0.04       833
           5       0.77      0.90      0.83      2432

    accuracy                           0.75      5273
   macro avg       0.65      0.59      0.55      5273
weighted avg       0.70      0.75      0.69      5273
```



. And last, I convert the multiclass values (5,4,3,2,1) into binary (5 & 1) of the target variable, then get 84 percent accuracy

```
:
## MULTINOMIAL NB

mnb= MultinomialNB()
mnb.fit(x_train,y_train)
predict= mnb.predict(x_test)
print('accuracy_score',accuracy_score(y_test,predict))
print('classification_report \n',classification_report(y_test,predict))

# PLOT CONFUSION MATRIX
cm= confusion_matrix(y_test,predict, labels= mnb.classes_)
disp= ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=mnb.classes_)
disp.plot()
plt.show()
```

```
accuracy_score 0.8449400697921408
classification_report
               precision    recall  f1-score   support

           1       0.81      0.93      0.87      3590
           5       0.89      0.75      0.81      3001

    accuracy                           0.84      6591
   macro avg       0.85      0.84      0.84      6591
weighted avg       0.85      0.84      0.84      6591
```



CONCLUSION

Describe The Key findings, Inferences Observations From the whole Problem

. The target column in this project is a multiclass and when I made the model on it, the accuracy was around 70% then I changed the values of the target to 3 types, then some accuracy increased. After that I changed the target to binary class as the positive and negative rating, then accuracy came at 84 percent.

. On this data it is difficult to predict multiclass target with good accuracy because some customer reviews in this data are almost sane but their ratings are different, that is why after converting this multiclass Target attribute to binary class, the accuracy is getting better

Limitation

The limitation of this project is that if we gave 100 positive reviews to be rating predication, then the expected prediction result is likely to be incorrect with about 17 to 20 reviews of this model