Rating Prediction Project

Submitted by:

Gargi Saha Samanta

ACKNOWLEDGMENT

I would like to express my gratitude towards FlipRobo Technologies for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons and my mentor(Ms. Swati Mahaseth) for giving me such attention and time as and whenever required.

INTRODUCTION

• Business Problem Statement

- 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

- There are many users who purchase products through E-commerce websites.
- Through online shopping many E-commerce enterprises were unable to know whether the customers are satisfied by the services provided by the firm.
- This boosts us to develop a system where various customers give reviews about the product and online shopping services, which in turn help the E-commerce enterprises and manufacturers to get customer opinion to improve service and merchandise through mining customer reviews.
- An algorithm could be used to track and manage customer reviews, through mining topics and sentiment orientation from online customer reviews. In this system user will view various products and can purchase products online.
- Customer gives review about the merchandise and online shopping services.
- Certain keywords mentioned in the customer review will be mined and will be matched with the keywords which are already exist in the database based on the comparison, system will rate the product and services provided by the enterprise.

- **Analytical Problem Framing**

Dataset Representation:

```
data=pd.read_excel('Ratings_Reviews.xlsx')
data.head()
```

	Unnamed: 0	Reviews	Ratings
0	0	Do not buy iphone or expensive product from Am...	1
1	1	Don't buy it from this seller	1
2	2	First Time iPhone User Review :-)	5
3	3	Worst Experience Ever.!	1
4	4	iPhone 11	1

```
data.drop("Unnamed: 0",axis=1,inplace=True)
```

```
print('Data shape is ',data.shape)

print('Data info',data.info())
```

```
Data shape is  (38438, 2)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38438 entries, 0 to 38437
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Reviews  38436 non-null  object
 1   Ratings  38438 non-null  int64
dtypes: int64(1), object(1)
memory usage: 600.7+ KB
Data info None
```

```
print('Data Set description',data.describe())
```

```
Data Set description            Ratings
count  38438.000000
mean       3.506686
std        1.592234
min        1.000000
25%        2.000000
50%        4.000000
```

So clearly it is classification problem.

- **Data Sources and their formats & inferences**

- Reviews of the product
- Rating of the products

Observation:

- 1 Independent variables with 1 target variable.
- From the dataset we can infer that it is clearly a classification problem.

- **Assumption for the problem:**
  - From the dataset we can infer that it is clearly a Classification problem.
  - This system will use text mining algorithm in order to mine keywords. The System takes review of various users, based on the review, system will specify whether the products and services provided by the E-commerce enterprise is good, bad, or worst.
  - We use a database of sentiment-based keywords along with positivity or negativity weight in database and then based on these sentiment keywords mined in user review is ranked.
  - This system is a web application where user will view various products and purchase products online and can give review about the merchandise and online shopping services.
  - This system will help many E-commerce enterprises to improve or maintain their services based on the customer review as well as to improve the merchandise based on the customer review.

- Hardware and Software Requirements and Tools Used

Software Used:

- Jupyter Notebook
- MS-Paint
- MS-PowerPoint
- MS-Word

Hardware used:

- Laptop
- Good internet connectivity

Model/s Development and Evaluation

- **Testing of Identified Approaches (Algorithms)**

- LogisticRegression()
- KNeighborsClassifier()
- RandomForestClassifier()
- DecisionTreeClassifier()

- **Running the selected Models:**

```
Linear=LogisticRegression()
knn=KNeighborsClassifier()
RandomForest=RandomForestClassifier()
DT=DecisionTreeClassifier()

algo=[Linear,RandomForest,DT,knn]
maximum_acc=[]

X_train,X_test,Y_train,Y_test=train_test_split(X,y,test_size=0.33,random_state=110)

for model in algo:
    model.fit(X_train,Y_train)
    Y_pred=model.predict(X_test)
    accuracy=round(accuracy_score(Y_test,Y_pred),4)*100
    confusionMatrix=confusion_matrix(Y_test,Y_pred)
    classificationReport=classification_report(Y_test,Y_pred)
    maximum_acc.append(accuracy)
    print(f"{model}:\n---------------------------\n---------------------------\n")
    print(f"The accuracy is {accuracy} of model {model} at random state 110")
    print("\n\nConfusion Matrix:\n\n",confusionMatrix)
    print(f"\n\n\n Classification report for the model:\n",classificationReport)

```

Accuracy of the Best Model (Random Forest Classifier)

```
Confusion Matrix:

[[4124  395  311  227  217]
 [ 444 4026  358  277  187]
 [ 255  291 3876  515  360]
 [ 172  218  466 2823 1180]
 [ 298  170  261 1257 3362]]


Classification report for the model:
              precision    recall  f1-score   support

           1       0.78      0.78      0.78      5274
           2       0.79      0.76      0.77      5292
           3       0.74      0.73      0.73      5297
           4       0.55      0.58      0.57      4859
           5       0.63      0.63      0.63      5348

    accuracy                           0.70     26070
```

Hyper Tuning The Random Forest Model:

```python
reg=RandomForestClassifier()
param={
    "n_estimators":[550,250],
    "min_samples_split":[4],
    "min_samples_leaf":[2],
    "max_depth":[27]

}
grd=GridSearchCV(reg,param_grid=param,cv=5)
grd.fit(X_train,Y_train)
print("Best Pramaeters:",grd.best_params_)

reg=grd.best_estimator_   #reinstantiating the best parameter to algo

reg.fit(X_train,Y_train)
ypred=reg.predict(X_test)

print(f"The accuracy is {round(accuracy_score(ypred,Y_test)*100,2)}% of model Random Forest.")


print("\nClassification Report:",classification_report(ypred,Y_test))

print(f"\n Confusion Matrix for the model:",confusion_matrix(ypred,Y_test))

```

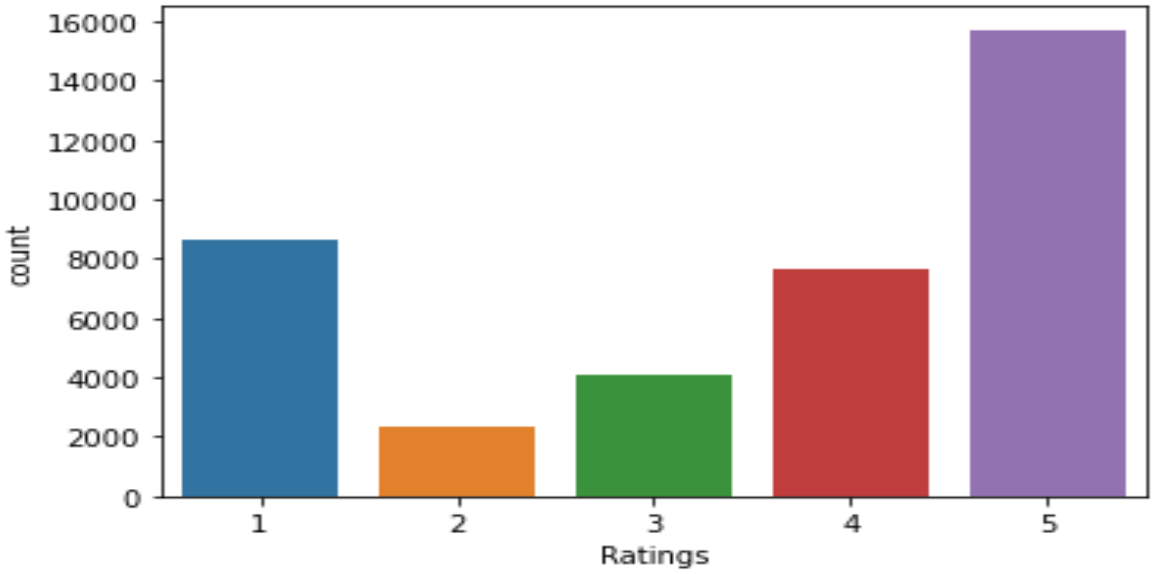
```
Best Pramaeters: {'max_depth': 27, 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 250}
The accuracy is 48.97% of model Random Forest.

Classification Report:               precision    recall  f1-score   support

           1       0.65      0.56      0.60      6061
           2       0.59      0.41      0.48      7634
           3       0.38      0.54      0.45      3785
           4       0.33      0.39      0.36      4084
           5       0.49      0.58      0.53      4506

    accuracy                           0.49     26070
   macro avg       0.49      0.50      0.48     26070
weighted avg       0.51      0.49      0.49     26070


 Confusion Matrix for the model: [[3417 1095  763  385  401]
 [1275 3117 1399  953  890]
 [ 290  514 2028  675  278]
 [ 165  414  726 1602 1177]
 [ 127  152  381 1244 2602]]
```

OBSERVATION:

- Finally we have saved the Random Forest Classifier Model.

- **Visualizations**

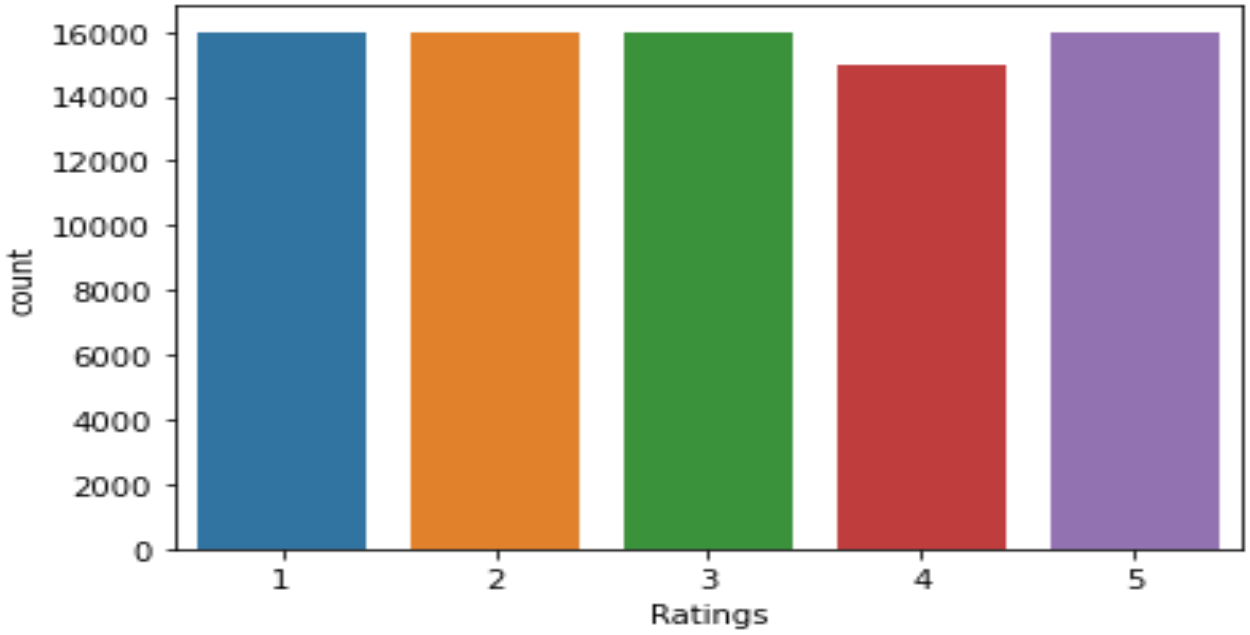
Distribution of Original Ratings:



Observation :

- The ratings are imbalanced.
- 5-star ratings are highest in number.
- It seems customers are quite satisfied with the products and giving good ratings.

Distribution of Ratings after oversampling:



Interpretation of the Results:

- Hence, we can go with normal Random Forest Classifier model with is also giving good accuracy.
- As we have decrease in accuracy after hyper tuning.
- So, it's better to use Random Forest with default parameters.

CONCLUSION

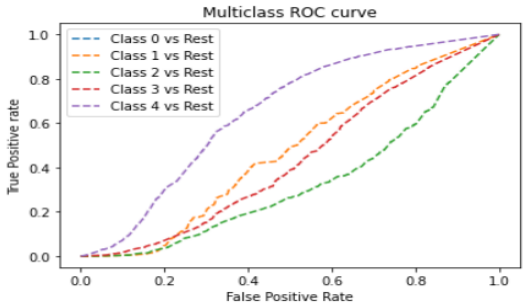
Finally we have saved the Random Forest Classifier Model.

• Learning Outcomes of the Study in respect of Data Science

- Hence we can go with normal Random Forest Classifier model which is also giving good accuracy.
- Finally we have saved the Random Forest Regressor Model.

```
Y_pred_prob_rf=RandomForest.predict_proba(X_test)

print("AUC ROC Score of RandomForest :",roc_auc_score(Y_test, Y_pred_prob_rf, multi_class='ovo', average='weighted'))

fpr = {}
tpr = {}
thresh ={}

n_class = len(Y_test.unique().tolist())

for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(Y_test, Y_pred_prob_rf[:,i], pos_label=i)

    # plotting
    plt.plot(fpr[i], tpr[i], linestyle='--', label=f"Class {i} vs Rest")
plt.title('Multiclass ROC curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive rate')
plt.legend(loc='best')
plt.savefig('Multiclass ROC',dpi=300);
```

```
AUC ROC Score of RandomForest : 0.9167320329307972
```



ROC_AUV CURVE

• Limitations of this work and Scope for Future Work

- In future this machine learning model using NLP may bind with various website which can provide real time data for Rating Review prediction.
- This system is a web application where user will view various products and purchase products online and can give review about the merchandise and online shopping services.
- This system will help many E-commerce enterprises to improve or maintain their services based on the customer review as well as to improve the merchandise based on the customer review.

****