NAME OF THE PROJECT

“Malignant Comment Classification”

Submitted by:

Sharad Yadav

ACKNOWLEDGMEN

I would like to thank fliprobo team that provided “Rating prediction” is current demanding topic that how to predict rating on amazon flipkart and other site. Here dataset extracted by selenium form amazon site in csv format, to predict rating of review products.

INTRODUCTION

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.

Data Set Description

The data set contains the training set, which has approximately 1195 samples. All the data samples contain 5 fields which includes 'Name', 'Brand', 'No_of_Ratings', 'Price', 'Rating'.

The data set includes:

- **Name:** It is the Name of product.
- **Brand:** It give you which Brand product you are selecting.
- **No_of_ratings:** It give you total no of rating.
- **Price:** It reflect price of product.
- **Rating:** It is reflect rating of product.

This project is more about exploration, feature engineering and classification that can be done on this data. Since the data is about rating prediction based on review. All features paly important role to predict rating.

- **Conceptual Background of the Domain Problem**

The idea behind rating prediction the rating is out 5 stars option available. we have to build an application which can predict the rating by seeing the review.

Data

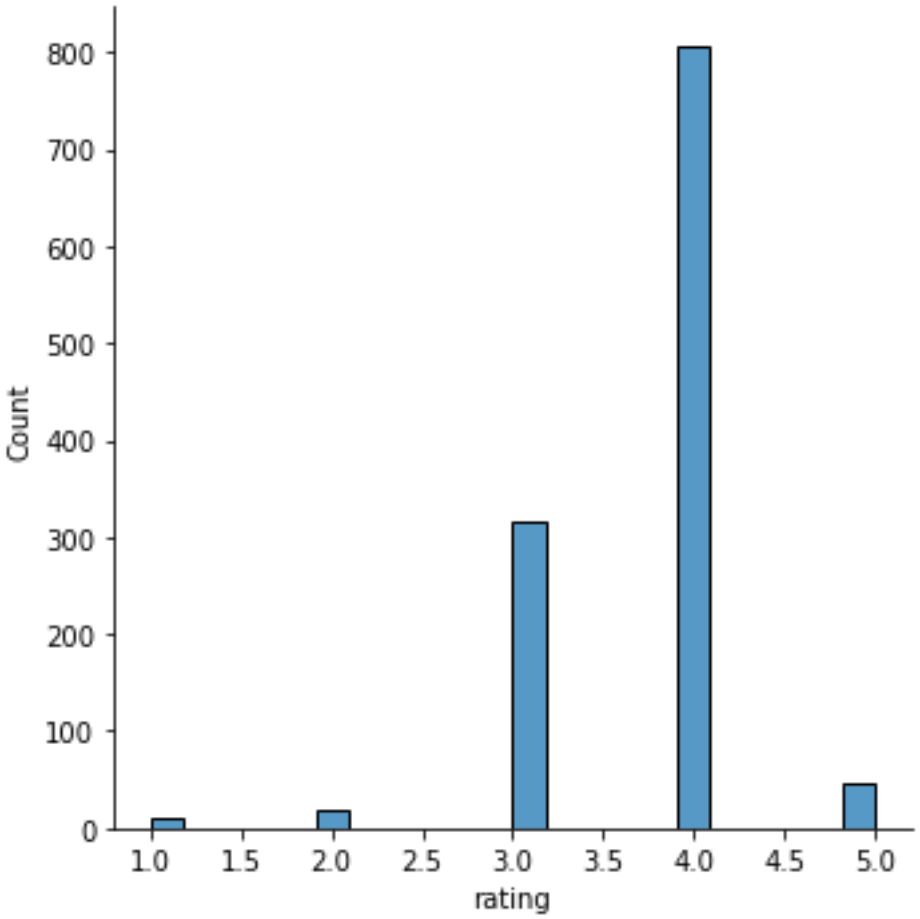
Data set 1195 rows and 5 column id review name and price ,total no of review to build rating prediction model.

- **Motivation for the Problem Undertaken**
  In recent times as online purchase of product buying based on rating came quite interesting, many of company focused on marketing but online rating paly significant role to choose best product in e-retail segment

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

I have to predict rating based on comment price and product review .



- **Data Sources and their formats**

- Dataset is in csv file, to read that data I have to used pandas library to read file further describe method to analysis get overview of data distribution, data info to identify object integer data types. Here this is all about predict

[2]:

```
# lets start with importing necessary library
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import statsmodels as sm
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import re
import warnings
warnings.filterwarnings("ignore")
```

[3]:

```
data=pd.read_csv(r"C:\Users\INPshy\Desktop\DATA Science\Product_rating.csv")
data.head()
```

Out[3]:

	Unnamed: 0	Name	Rating	Brand	NO_OF_RATINGS	PRICE
0	0	NIBOSI Chronograph Men's Watch (Black Dial & ...	4.1 out of 5	-	246 ratings	₹2,290.00
1	1	TIMEWEAR Digital Men's Watch	3.8 out of 5	-	12,814 ratings	₹739.00
2	2	RAW STAR Day and Date Functioning Colored Dial...	-	-	-	₹879.00
3	3	Victorinox Chrono Classic Analog Black Dial Me...	4.4 out of 5	-	27 ratings	₹38,760.00
4	4	Gionee STYLFIT GSW6 Smartwatch with Bluetooth ...	-	GIONEE	-	₹2,999.00

- Data Preprocessing Done

The following steps were taken to process the data

1. I have extracted numeric values from object data
2. Convert rest object data in numeric by imputing.
3. Splitting data set into Training and testing by train_test_split method

```
df=data.Rating
```

```
newList = []

progress = True
i = 0;

while (progress):
    progress = False
    sublist = []

    for list in df:
        if len(list) <= i:
            continue
        else:
            sublist.append(list[i])
            progress = True

    if not progress:
        break

    newList.append(sublist)
    i = i+1


print(newList[0])
```

Extract numeric value form object

- Data Inputs- Logic- Output Relationships
  Since rating decide by review product features price and other various features.

- **Hardware and Software Requirements and Tools Used**
  Below required sklearn library and various other python techniques required to build model

```python
# lets start with importing necessary library
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import statsmodels as sm
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import re
import warnings
warnings.filterwarnings("ignore")
```

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

- Testing of Identified Approaches (Algorithms)
  1. Linear Regression
  2. Decision Tree Regression
  3. Gradient Boosting Regression
  4. Random Forest Regression
  5. XGB

- **Run and Evaluate selected models**
  1. XGB

```
import xgboost as xgb
xgb=xgb.XGBRegressor()
```

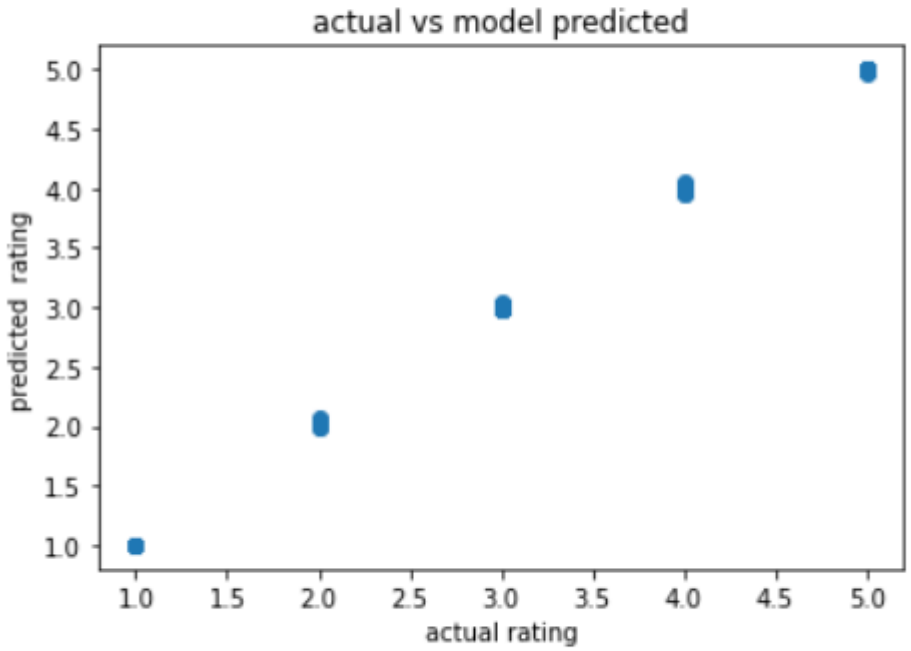
```
xgb.fit(x_train,y_train)
```

0]:
```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.300000012, max_delta_step=0, max_depth=6,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=0,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

```
y_pred=xgb.predict(x_train)
print(r2_score(y_train,y_pred))
```

```
0.9998885680848175
```

```
#Visualization of actual vs predicted rating
plt.scatter(y_train,y_pred)
plt.xlabel('actual rating')
plt.ylabel('predicted  rating')
plt.title('actual vs model predicted')
plt.show()
```




```
cross_val_score(xgb,X_scaler,Y,cv=5).mean()
```

]:
```
0.009280226415890214
```

```
1 cross_val_score(xgb,X_scaler,Y,cv=5).mean()
```

```
3]: 0.009280226415890214
```

```
1 print('**** accuracy  ****')
2 print(r2_score(y_train,y_pred))
```

```
**** accuracy  ****
0.9998885680848175
```

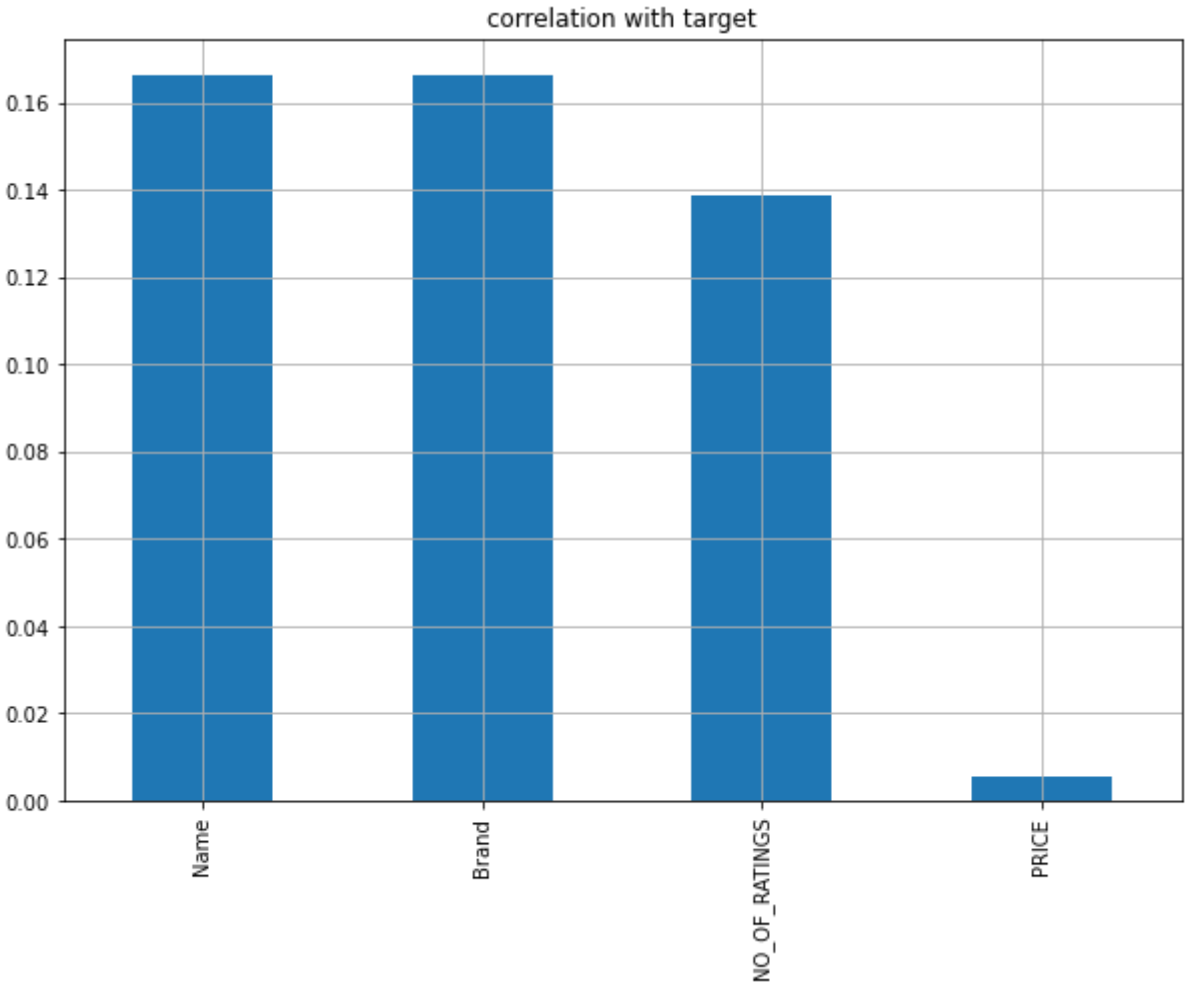
Model saving

```
1 #saving model AdaBoost model
2 import pickle
3 file='pikle_ada_model'
4 with open(file ,'wb') as file:
5     pickle.dump(xgb, file)
```

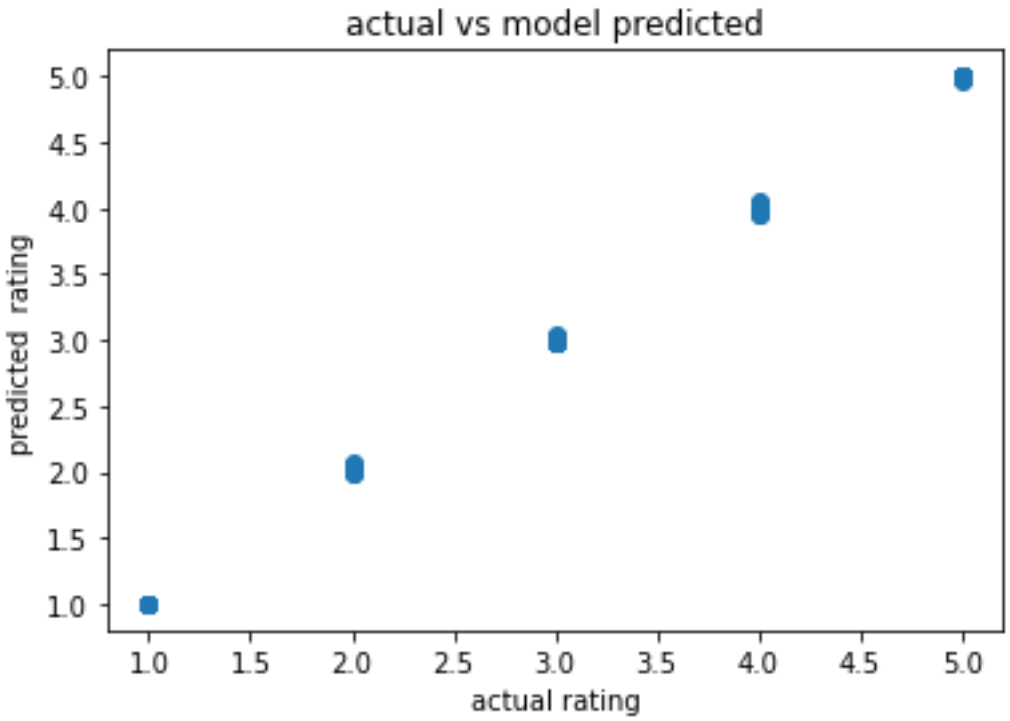
- **Key Metrics for success in solving problem under consideration**

  Our final Model is XGB chosen best among all because of its perform better than other actual vs predicted data, low R2 error.

- **Visualizations**



Correlation with Target Variable



Actual Vs predicted

- **Interpretation of the Results**

CONCLUSION

- **Key Findings and Conclusions of the Study**
  I have found some rating between 4 to 5 category which is top rated product and price is high among belonging category.
- **Learning Outcomes of the Study in respect of Data Science**

  This project is quite different and relatable to present scenario, but data mining and extracting meaningful data is difficult, pre-processing and cleaning is most time consuming part.

- **Limitations of this work and Scope for Future Work**

  The current project to identify rating, future I will add following points

  E-commerce website increasing every year and product sales mostly based on rating ,every user check rating and compare product, I will add more features in this project

  Build feedback model to increase efficiency