RATING PREDICTION PROJECT

SUBMITTED BY : **SINDHU SHREE N**

INTERNSHIP BATCH : **19**

ACKNOWLEDGMENT

With great pleasure, I sincerely express my deep sense of gratitude and heartfelt thanks to several individuals from whom I received impetus motivation during the internship project work. I am very grateful to **SHUBHAM YADAV** sir internship 19th batch guide, for the help and I do express my deep gratitude to sir for his guidance, keen interest and constant encouragement throughout the internship project work. I am thanking him for personal concern, affinity and great spirit with which he has guided the work.

INTRODUCTION

The rise in E -commerce, has brought a significant rise in the importance of customer reviews. There are hundreds of review sites online and massive amounts of reviews for every product. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches.

The ability to successfully decide whether a review will be helpful to other customers and thus give the product more exposure is vital to companies that support these reviews, companies like Amazon , Flipkart, E-Bay etc.

The proliferation of online shopping enables people to express their opinions widely online. And the online shopping helps the customers to get their product on door. But inorder to write their opinion these online platforms are providing rating option along with review and these ratings range from 1 to 5 stars. If the product satisfied by the customer then they may give their highest rating of 5 stars. And if not then they can rate a lowest of one star. It helps the online retailers to analyse their product quality and to understand the customers' requirements.

Business Problem Framing

Inorder to build a model that predicts the rating of products from different online platforms such as amazon, flipkart, E-bay etc. using sentimental analysis.

Conceptual Background of the Domain Problem

Sentiment classification regarding rating and review has been intensively researched in the past few years, largely in the context of E-commerce websites data where researches have applied various machine learning systems to try and tackle the problem as well as the related, which is a task of sentiment analysis.

Review of Literature

Rating prediction using review research initially began with Yin et al's application of combining TF-IDF with sentiment features. They compared the performance of this model with a simple TF-IDF model.

Motivation for the Problem Undertaken

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 prevent the user from choosing a product with ease. The star-rating, i.e. stars from 1 to 5 on Amazon, 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.

Our main goal is to collect the data from different E-commerce websites and to train the model using various machine learning models and to get predict the rating using the best model.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

1. Included in 'Data-cleaning ipynb'.
2. Selecting relevant features.
3. Exploratory the data analysis and data cleaning.
4. Null value imputation
5. Training a machine learning model
6. Hyperparameter tunning

7. Evaluate the model
8. Predictions of the model

Data Sources and their formats

The data is obtained from various online platforms like amzon, flipkart etc. and stored it as a CSV file. Which was imported and performed the data cleaning process.

Data Pre-processing Done

The data set for building the Regression model was acquired from the competition site. The steps elaborated below will describe the entire process from Data Pre-Processing to Model Testing.

1. Checking for missing values

   The first and foremost step after importing the training and testing sets in pandas dataframe is to check for the null values. Using 'isnull()' function on the dataset. Here I discovered no missing values so the data is clean and hence I can proceed to the next step.

2. Text Normalising

   As I did not found any missing values now I can proceed for the data preprocessing. Since our data is directly extracted from online platforms it contains special characters, numbers, unnecessary spaces etc., hence the text has to be normalize.

   1) Converting data to lower case
   2) Removing Punctuation

3. Removing stopwords:

   As we all know, it is one of the most crucial steps in text preprocessing for use-cases that involve text classification. Removing stopwords ensures that more focus is on those words that define the meaning of the text.
   To remove stopwords from my data, first we imported stopwords from nltk.corpus then we removed them.

4. Converting rating feature from object type to the integer type:
   As the rating are directly extracted from the online platforms it is of object datatype so we have rounded them to their nearest integer and thus it is converted to integer datatype (Ex: 4.3 out of 5 stars to 4)

Data Inputs- Logic- Output Relationships

Since it is a regression model where the target feature is continuous data, and no more features we just did a sentimental analysis by analysing the ratings.

Set of assumptions (if any) related to the problem under consideration

Since it is a real-world problem and based on the users reviews no assumptions are needed. So I have made no assumptions here.

Hardware and Software Requirements and Tools Used

Hardware requirements:

PROCESSOR: Intel(R) Core(TM) i3 CPU

MONITOR : Any display unit

HARD DISK : 240GB SSD

RAM : 8.00GB

Software requirements:

OPERATING SYSTEM: Windows 10 Pro

FRONT END : Jupyter Notebook (Anaconda3)

BACK END : Excel 2013

Tools Used:

1) Pandas Library

2) Numpy Library

3) Seaborn Library

4) Matplotlib

5) Scikit_learn

6) nltk library

7) TF-IDF

Model/s Development and Evaluation

Since we have completed the data pre-processing and feature engineering part of our project, we move on to the model creation and model assessment part of the project. Before trying to fit a regression models on the training data, I randomly split the data into train-set and test-set. The test set accounts for 20% of the training data.

Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

1) Linear Regression
2) Decision Tree Regressor
3) Gradient Boosting Regressor
4) XG Boost Regressor

Run and Evaluate selected models

1) **Linear Regression**: It is the most commonly used type of regression model. In this model we can observe that the r2_score of training and testing sets are 85% and 84% respectively.

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
y_pred_lr=lr.predict(x_test)
```

```
training_data_prediction=lr.predict(x_train)
```

```
from sklearn.metrics import r2_score
error=r2_score(y_train,training_data_prediction)
print("R squared error :",error)
```

```
R squared error : 0.8568213077968965
```

```
testing_data_prediction=lr.predict(x_test)
```

```
error=r2_score(y_test,testing_data_prediction)
print("R squared error :",error)
```

```
R squared error : 0.84333593010761
```

2) **Decision Tree Regressor:** It is a type of regression model by building a decision tree. In this case it has a r2_score of training set is 92% and that of testing set is 91%.

```
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt.fit(x_train,y_train)
y_pred_dt = dt.predict(x_test)
```

```
training_data_prediction=dt.predict(x_train)
```

```
error=r2_score(y_train,training_data_prediction)
print("R squared error :",error)
```

```
R squared error : 0.9209487552292118
```

```
testing_data_prediction=dt.predict(x_test)
```

```
error=r2_score(y_test,testing_data_prediction)
print("R squared error :",error)
```

```
R squared error : 0.9153383806646845
```

3) **Gradient Boosting Regressor:** It is a type of ensemble learning. It has a training set r2_score of 49% and testing set r2_score of 46%.

```
from sklearn.ensemble import GradientBoostingRegressor
gbr=GradientBoostingRegressor()
gbr.fit(x_train,y_train)
```

```
GradientBoostingRegressor()
```

```
training_data_prediction=gbr.predict(x_train)
```

```
error=r2_score(y_train,training_data_prediction)
print("R squared error :",error)
```

```
R squared error : 0.49458450362441175
```

```
testing_data_prediction=gbr.predict(x_test)
```

```
error=r2_score(y_test,testing_data_prediction)
print("R squared error :",error)
```

```
R squared error : 0.4613694368867851
```

4) **XG Boost Regressor:** The extreme gradient boosting regressor is a type of regressor which can boost the gradient boost model. For this dataset it is giving a r2_score on training set 79% is and the testing set is 77%.

```
import xgboost as xgb
xg=xgb.XGBRegressor()
xg.fit(x_train,y_train)
```

```
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=4, 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)
```

```
training_data_prediction=xg.predict(x_train)
```

```
error=r2_score(y_train,training_data_prediction)
print("R Squared Error:",error)
```

```
R Squared Error: 0.7944545276014449
```

```
testing_data_prediction=xg.predict(x_test)
```

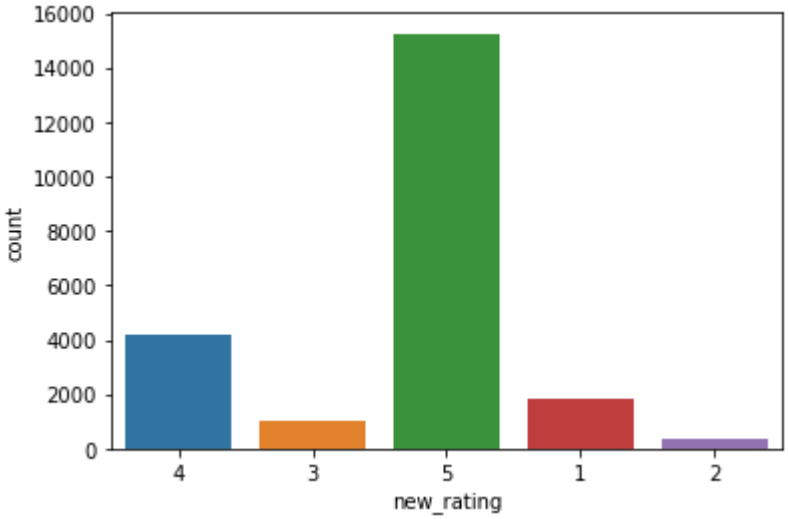
```
error=r2_score(y_test,testing_data_prediction)
print("R Squared Error:",error)
```

```
R Squared Error: 0.7692717367381614
```

Visualization:

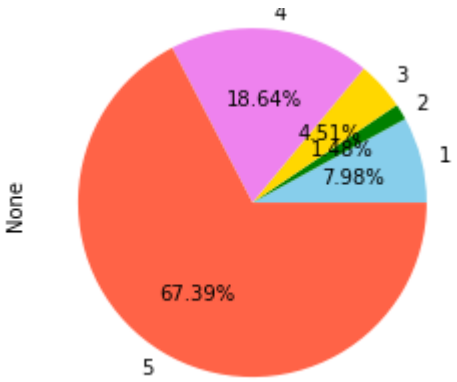
I have visualized the data in different aspects and used pie charts, barplots, heatmaps and word cloud which are showed below.

It shows that the dataset does not contains the null values.



This graph shows that the count of ratings variance the rating 5 has a highest count than that of other ratings.

This diagram shows the sense of good words used by the users who purchased the products.



Here we can see the amount of ratings given by the customers. So almost 67% of customers rated the products to the highest of 5 and 18% rated 4, nearly 8% rated 1, 4% customers rated 3 and remaining 1% rated 2.

Interpretation of the Results

On comparing all the results of these four models mentioned above we have observed that all models are performing well but the training set and testing set score is high almost near in Decision tree regressor and also has a comparative cross validation score. Thus we have choosed this model for hyperparameter tuning and there also we got a improved result. Hence **Decision Tree Regressor** is the best fit for the present dataset.

CONCLUSION

Key Findings and Conclusions of the Study:

After evaluating the results procured during the training phase of my project and the results that I received from the websites, I can claim that the Decision Tree Regressor performs better than other regression models. And using it we have predicted the output for the testing set and compared the actual and predicted rating which is almost near.

Learning Outcomes of the Study in respect of Data Science:

This project allowed me to work with four different regression models and further, I was able to implement them on Natural Language Processing use-case. The various data pre-processing and feature engineering steps in the project made me to have knowledge of the efficient methods that can be used to clean textual data. I understood the working of various classification models such as Decision Tree regressor, Gradient Boosting Regressor, Linear Regression, XGboost Regressor. I got introduced to the concepts of wordcloud,

stopwords and advantages of using it in textual data. Finally doing hyper-parameter tuning for the model helped me to achieve optimum results.

Limitations of this work and Scope for Future Work

- Limited Access to information
- Time Limits
- Conflicts on biased views and personal issues
- How to structure my project research limitation correctly
- How to set my project research limitation.
- Formulation of my objectives and aims
- Implementation of my data collection methods
- Scope of discussions
- Finding my error on the codes

****