*****RATINGS Prediction Project***



***Submitted By:***

***KOUSHIK BISWAS***

***(Internship:Flip Robo Techknology@Bangalore)***

***Acknowledgements***

The completion of this thesis study would not have been possible without the support and extensive knowledge of several website.

I cannot begin without expressing my thanks to my SME from *Flip Robo Technologies* , ***Khushboo Garg***, for the valuable experience and insight in conducting and writing scientific reports. Thank you for always answering all questions and giving me helpful practical suggestions.

Someone whose help cannot be overestimated is my advisor from Career Coach, *Dr. Deepika Sharma VP, Learning and Development at DataTrained.* Thank you for always taking your time to support and guide me through the project. Without your extensive knowledge and insightful suggestions, the final result would not be what it is today.

Finally, I would like to express my sincere gratitude to my family for their constant encouragement and support throughout my time on this thesis.

***Introduction***

Here, we get the Rating of many products from the website **https://amazon.com/**, **https://Flipkart.com/** filled with information On different types reviews of product. We realize that we can use this data to make sure we get a good idea of selected product. In particular, we can figure out exactly how much to pay for a specific product.

In this article, I am going to walk you through how we can train a model that will help us predict a specific product how much good its for people. you’ll practice the machine learning workflow you’ve learned so far to predict Rating of a particular product using its reviews. The dataset we will be working with contains information on various reviews of many products.

Anyone who has buy a product from any online shoping store knows how unexpectedly the prices vary. Thousands of products are sold everyday. There are some questions every buyer asks himself like: What is the actual price that product deserves? Am I paying a fair price? Is it good product? How to choose cheap and best product from online?

So, they are looking for new machine learning models from new data. We have to make Rating prediction model. This project contains two phase-

* Data Collection Phase
* Model Building Phase

Artificial Intelligence is an integral part of all major e-commerce companies today. Today's online retail platforms are heavily powered by algorithms and applications that use AI. Machine learning is used in a variety of ways, from inventory control and quality assurance in the warehouse to product recommendations and sales demographics on the website. Let’s say you want to create a promotional campaign for an e-commerce store and offer discounts to customers in the hopes that this might increase your sales. You have been provided descriptions of products on Amazon and Flipkart, including details like product title, ratings, reviews, and actual prices. In this challenge, you will predict discounted prices of the listed products based on their ratings and actual prices. That's why I want to do this project and share my knowledge which I have learned from my institute.

***Analytical***

***Problem Framing***

***Mathematical/ Analytical Modeling of the Problem:-*** In this project, we will develop and evaluate the performance and predictability of trained and tested models based on reviews and ratings data collected from **Amazone and Flipkart**. Once we get a good fit, we will use this model to predict the rating of a specific product.

In here we will use various classification algorithm to predict our target. Let's have an overview of the algorithms we will use for our predictions. To read more about these algorithms , just click on the algorithms name.

* [LogisticRegression](https://www.google.com/search?q=linear+regression&rlz=1C1CHBF_enIN997IN998&oq=&aqs=chrome.1.69i59i450l8.734952339j1j15&sourceid=chrome&ie=UTF-8):- Logistic regression analysis is valuable for predicting the likelihood of an event. It helps determine the probabilities between any two classes. In a nutshell, by looking at historical data, logistic regression can predict whether: An email is a spam.
* [DecisionTreeClassifier](https://www.google.com/search?q=about+DecisionTreeRegressor&rlz=1C1CHBF_enIN997IN998&ei=7kG5YoWNM6fA3LUPqcGy8AQ&ved=0ahUKEwiFvLry9Mz4AhUnILcAHamgDE4Q4dUDCA4&uact=5&oq=about+DecisionTreeRegressor&gs_lcp=Cgdnd3Mtd2l6EAM6BAgAEA1KBAhBGABKBAhGGABQAFjfEWDdFWgAcAF4AIABqQKIAZYLkgEDMi02mAEAoAEBwAEB&sclient=gws-wiz):- Decision trees **help you to evaluate your options**. Decision Trees are excellent tools for helping you to choose between several courses of action. They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options.
* [SVR](https://www.google.com/search?q=about+SVR&rlz=1C1CHBF_enIN997IN998&oq=about+SVR&aqs=chrome..69i57j0i10i22i30l6j0i390l3.4767j1j15&sourceid=chrome&ie=UTF-8):- The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points. Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value.
* [KNeighborsClassifier](https://www.google.com/search?q=about+KNeighborsRegressor&rlz=1C1CHBF_enIN997IN998&oq=about+KNeighborsRegressor&aqs=chrome..69i57j33i160.3952j1j15&sourceid=chrome&ie=UTF-8):- By default, the KNeighborsClassifier looks for the 5 nearest neighbors. We must explicitly tell the classifier to use Euclidean distance for determining the proximity between neighboring points. Using our newly trained model, we predict whether a tumor is benign or not given its mean compactness and area..
* [RandomForestClassifier](https://www.google.com/search?q=about+RandomForestRegressor&rlz=1C1CHBF_enIN997IN998&ei=n0a5Yq5xxJWx4w_O07lA&ved=0ahUKEwjuvN-u-cz4AhXESmwGHc5pDggQ4dUDCA4&uact=5&oq=about+RandomForestRegressor&gs_lcp=Cgdnd3Mtd2l6EAMyBwghEAoQoAEyBwghEAoQoAE6BwgAEEcQsAM6CggAEOQCELADGAE6BQgAEIAEOggIABCxAxCDAToICAAQgAQQsQM6CwgAEIAEELEDEIMBOgUIABCRAjoLCC4QgAQQsQMQgwFKBAhBGABKBAhGGAFQywhYjiBgrzJoAXABeAKAAbQDiAH5EJIBCTAuMi4yLjAuM5gBAKABAaABAsgBDcABAdoBBggBEAEYCQ&sclient=gws-wiz):- What is Randomforestclassifier in Python?A random forest classifier. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting..

You can find the complete project, documentation and dataset on [my GitHub page](https://github.com/KBkoushik/Flight-Price-Prediction-Project.git)

https://github.com/KBkoushik/RATINGS-PREDICTION.git

***Data Sources and their formats:-*** I have scraped the product reviews data from amazon and flipkart. The dataset contains 37324 rows and 3 columns having both numerical and categorical data.

In this dataset " Ratings" is our target variable which has only 5 type(1,2,3,4,5) of data. So this is a “Classification type" problem.

you can download my dataset from the below link.

**Dataset link:**

***https://github.com/KBkoushik/Flight-Price-Prediction-Project/blob/5e7dfa256e1b30e53e0c57432ab29791b12ae95b/Flight\_Prices(web%20scraping).xlsx***

Dataset looks as follows:-



***About The DataSet:-***

**Rating:** Rating of the specific product

**Review:**Heading of the Review

**Long\_Review:**complete review of the product

***Data Preprocessing Done:-***

For the purpose of the project the dataset has been preprocessed as follows:

* We have few missing values in the dataset. We drop this missing value
* we divide all the columns into categorical and numerical types
* Univariate Analysis Of Categorical Columns and numerical columns.
* All the text in Review and Long Review columns changes into lower case
* Adding length of Review and Long\_review columns as a new columns
* Replace Email address,url and phone number with email,webaddress and phonenumber respectively
* Removing the pucntuations
* writing function for the entire dataset
* Lemmatizing and then Stemming with Snowball to get root words and further reducing characters
* Plot Word cloud of each rating
* Dividing data into features and vectors
* Oversampling for data balancing

We’ll now open a python 3 Jupyter Notebook and execute the following code snippet to load the dataset and remove the non-essential features. Recieving a success message if the actions were correclty performed.

***Hardware and Software Requirements and Tools Used:-***

* ***Hardware:- Desktop/Laptop***
* ***Software:- Anaconda***
* ***Libraries:-*** ***Numpy,Pandas,Matplot,Seaborn,nltk,etc***

***Model/s Development and Evaluation***

* ***Identification of possible problem-solving approaches (methods):***
* ***Missing Value Handling:***

**There are 2 primary ways of handling missing values:**

* Deleting the Missing values:-Generally, this approach is not recommended. It is one of the quick and dirty techniques one can use to deal with missing values.
* Imputing the Missing Values:- There are different ways of replacing the missing values
* Replacing With Mean
* Replacing With Mode
* Replacing With Median,etc.

For my dataset,I will use mode for categorical features and median for numerical features by the code below.

******Since our dataset has only (63+72)=135 missing values only and the dataset contents approximately 38000 rows, we choose the first option(Deleting the Missing values). The code of this operation is

* ***Word Cloud of each rating***
  + ***Word Cloud of 1 Rating***
  + ***Word Cloud of 2 Rating***

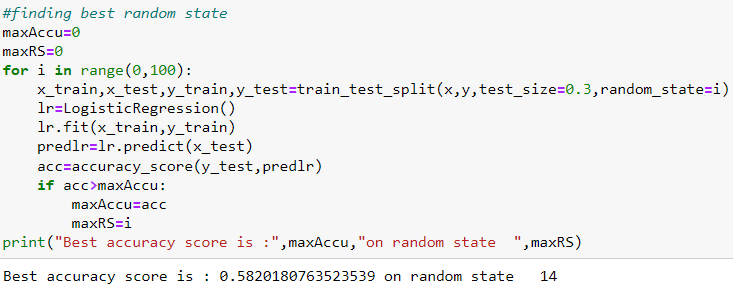
******

* + ***Word Cloud of 3 Rating***

******

* + ***Word Cloud of 4 Rating***
  + ***Word Cloud of 5 Rating***
* ***Testing of Identified Approaches (Algorithms):-***
  + ***Finding Best Random State:-***

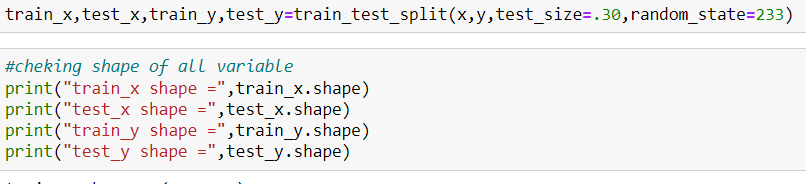
An algorithm might have multiple points that introduce randomness to the process and thus introduce randomness to the result. One method to make sure our results are constant is to set every possible *random\_state* available in the functions that we use. We can find the best random state by the code below.

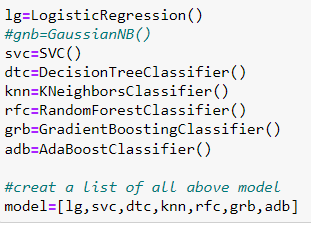
******

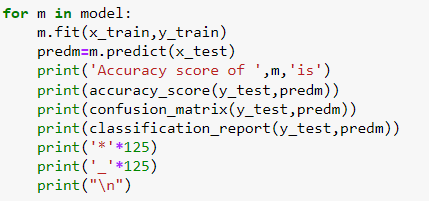
### After we have found the value for best random state, we proceeded with the train test split function to create new training and testing datasets and fit them into the models to find our ideal models.

### ***Train\_Test\_Split:-***

The [Train\_Test\_Split](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.



Now I will be using 7 differents classification algorithms to get the ideal one for prediction.

* ***Run and Evaluate selected models:-***

Accuracy score of LogisticRegression() is

0.5820180763523539

[[2020 373 438 53 79]

[ 630 1548 546 142 75]

[ 356 282 1671 144 538]

[ 104 230 468 1121 1027]

[ 73 70 266 303 2269]]

precision recall f1-score support

1 0.63 0.68 0.66 2963

2 0.62 0.53 0.57 2941

3 0.49 0.56 0.52 2991

4 0.64 0.38 0.48 2950

5 0.57 0.76 0.65 2981

accuracy 0.58 14826

macro avg 0.59 0.58 0.58 14826

weighted avg 0.59 0.58 0.58 14826

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Accuracy score of SVC() is

0.6546607311479833

[[2422 205 251 17 68]

[ 640 1776 374 88 63]

[ 396 247 1772 91 485]

[ 110 206 358 1318 958]

[ 114 24 215 210 2418]]

precision recall f1-score support

1 0.66 0.82 0.73 2963

2 0.72 0.60 0.66 2941

3 0.60 0.59 0.59 2991

4 0.76 0.45 0.56 2950

5 0.61 0.81 0.69 2981

accuracy 0.65 14826

macro avg 0.67 0.65 0.65 14826

weighted avg 0.67 0.65 0.65 14826

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Accuracy score of DecisionTreeClassifier() is

0.5631323350870093

[[1871 473 448 73 98]

[ 560 1624 464 194 99]

[ 346 428 1434 231 552]

[ 109 288 374 1260 919]

[ 100 77 310 334 2160]]

precision recall f1-score support

1 0.63 0.63 0.63 2963

2 0.56 0.55 0.56 2941

3 0.47 0.48 0.48 2991

4 0.60 0.43 0.50 2950

5 0.56 0.72 0.63 2981

accuracy 0.56 14826

macro avg 0.57 0.56 0.56 14826

weighted avg 0.57 0.56 0.56 14826

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Accuracy score of KNeighborsClassifier() is

0.44165654930527454

[[ 860 1251 673 136 43]

[ 277 2004 494 136 30]

[ 137 796 1750 192 116]

[ 54 657 915 1037 287]

[ 58 331 1157 538 897]]

precision recall f1-score support

1 0.62 0.29 0.40 2963

2 0.40 0.68 0.50 2941

3 0.35 0.59 0.44 2991

4 0.51 0.35 0.42 2950

5 0.65 0.30 0.41 2981

accuracy 0.44 14826

macro avg 0.51 0.44 0.43 14826

weighted avg 0.51 0.44 0.43 14826

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Accuracy score of RandomForestClassifier() is

0.6587076757048429

[[2143 394 328 25 73]

[ 471 2017 323 66 64]

[ 293 309 1816 109 464]

[ 75 266 269 1486 854]

[ 74 65 261 277 2304]]

precision recall f1-score support

1 0.70 0.72 0.71 2963

2 0.66 0.69 0.67 2941

3 0.61 0.61 0.61 2991

4 0.76 0.50 0.60 2950

5 0.61 0.77 0.68 2981

accuracy 0.66 14826

macro avg 0.67 0.66 0.66 14826

weighted avg 0.67 0.66 0.66 14826

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Accuracy score of GradientBoostingClassifier() is

0.5265074868474302

[[1907 344 595 44 73]

[ 714 1298 630 216 83]

[ 389 421 1383 223 575]

[ 116 272 436 1046 1080]

[ 91 54 407 257 2172]]

precision recall f1-score support

1 0.59 0.64 0.62 2963

2 0.54 0.44 0.49 2941

3 0.40 0.46 0.43 2991

4 0.59 0.35 0.44 2950

5 0.55 0.73 0.62 2981

accuracy 0.53 14826

macro avg 0.53 0.53 0.52 14826

weighted avg 0.53 0.53 0.52 14826

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Accuracy score of AdaBoostClassifier() is

0.4825306893295562

[[1737 274 853 43 56]

[ 680 1038 907 253 63]

[ 419 267 1443 359 503]

[ 111 268 551 1051 969]

[ 77 27 526 466 1885]]

precision recall f1-score support

1 0.57 0.59 0.58 2963

2 0.55 0.35 0.43 2941

3 0.34 0.48 0.40 2991

4 0.48 0.36 0.41 2950

5 0.54 0.63 0.58 2981

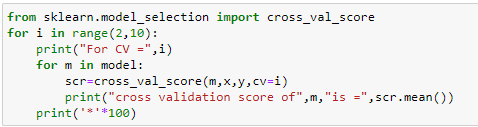
accuracy 0.48 14826

macro avg 0.50 0.48 0.48 14826

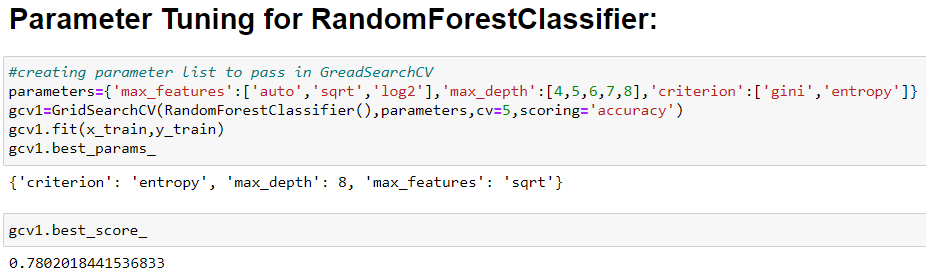
weighted avg 0.50 0.48 0.48 14826 we get best accuracy score from XGBRegressor, RandomForestRegressor, DecisionTreeRegressor

* + ***Cross Validation Score:***

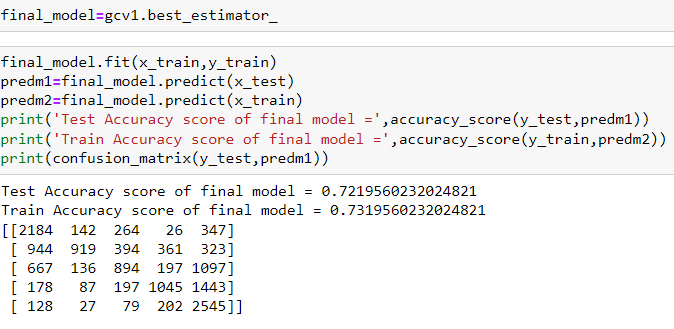
The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

Now we check cross val score of the above models by the code below:

#### After checking cross val score we can see the difference between accuracy score and cross val score is less and cross val score is maximum compare to other model at CV=5. So, we choose RandomForest classifier for parameter tuning. **Hyperparameter tuning of GradientBoostingRegressor:-**

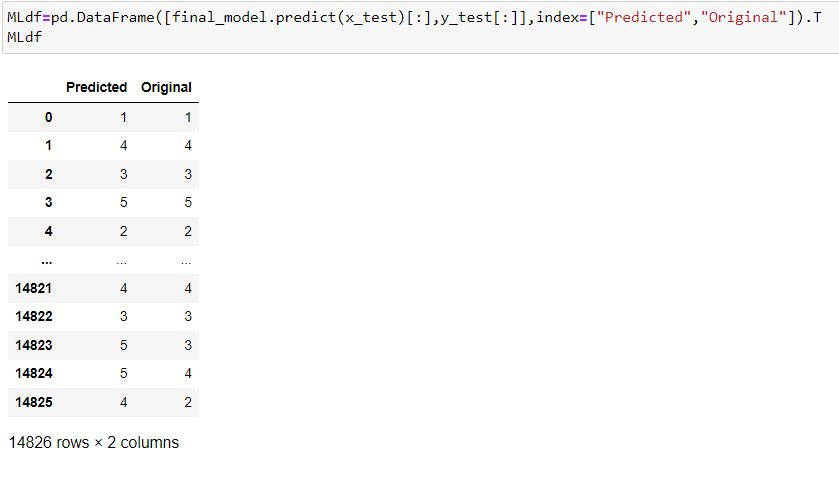


after parameter tuning for RandomForestClassifier, we get best accuracy score from it.so we creat a final model with this parameter.



since difference between train accuracy and test accuracy is very less ,our model is not overfit or underfit.

* ***SHOW PREDICTED AND ORIGINAL RATINGS:-***

******

We can see that the original old car price and final model’s predicted price are almost same which means model performance is good

* ***Interpretation of the Results:-***
* We see that our model accuracy only 72%.we have to improve accuracy.Its possible only if I have more knowledge about NLP function.
* At first we get only 58% accuracy from Logistic Regression.But finaly after parameter tuning we get 72% accuracy
* Using hyper parameter tunning we can improve our model accuracy, for instance in this model the accuracy increased.
* It is always advised to all of us that atleast we need to use 5 Algorithm in order to figure out which one is performing best among them and we choose that one and we send that for hyper parameter tuning to know that best parameter .

***CONCLUSION***

* When I predict the ratings from the reviews we can see that a lot of complexity was involved. Some comments in the review are not meaningful due to some spelling mistakes. This is very critical to increase the accuracy score of a model.
* For any of machine learning project my suggestion is first you have to understand the problem on ground level .if you don’t allow yourself to work with diligence .if you don’ t work harder anything that you are doing or will do , not only in case of machine learning but also in life cycle would be futile. Maybe, my endeavour assist you when ever you will get stuck
* For future improvements, following step we thought to took-
  + Replacing model with a latest/different model
  + Using other robust datasets
  + More focus on NLP properties
* It would seem that better performance might be achieved if multiple learners were combined.

