**Car Purchasing Annually Spending Money using Multiple Linear Regression Algorithm**

**Documentation about the program:**

In this program we are going to develop the Multiple Linear Regression using Oops concept for Car Purchasing data set Mainly we Focus on Accuracy of Data and Losses of Data.

**Understand Data Set:**

Open that data file and understand what the Colum’s represent and what are Colum’s names and inside data and what are unique labels present and how many rows present in data set.

**Working on Project:**

1.open your IDLE for working on project and import what essential library you required and load the file into you IDLE and create a Data Frame that to see the Structured data like excel means Rows and Colum’s.

2.Manily modifying means unnecessary data Colum’s like Name, Id, Country, Age…. Those Colum’s remove.

& if this is any null value are present or not check and modify and if any categorical data present that one also change into numerical data .

3.Split the data using Scikit learn library’s.

4.In library there is function is train test split function using to split data set into 2 parts

X=Independent Colum’s

Y=Dependent Colum’s

i)Training data having X\_train,y\_train

ii)Testing data having X\_test,y\_test

**Training Data:**

5. we need to focus on First Training Data so that Train data is give the algorithm is Linear Regression.

6.fiting the training data to the Algorithm then giving Model .

**Training Performance:**

7.the Model is giving y\_train\_predicted values then create a new Data Frame and giving X\_Train, y\_train, y\_train\_predicted collum’s to Data Frame .

8.**R2\_score function(Accuracy):**

**From sklearn.metrics import r2\_score**

**Syntax: r2\_score(y\_train ,y\_train\_predicted)**

Then it will giving Accuracy of the train Data is 0.9999999812764105

9.**Mean\_square\_error(Loss):**

**From sklearn.metrics import mean\_square\_error**

**Syntax:mean\_square\_error(y\_train,y\_train\_predicted)**

Then it will give loss of train Data is 2.1990016683893954

Without using r2\_score best to using score function

Syntax: reg.score(X\_train,y\_train)

Then it will calculate the predict function and then get predicted values and next calculate Accuracy is the train data giving Accuracy value only not showing predicted values.

10.Manually Calculating Accuracy and Losses and Formula’s

i)r2\_score-> 1-summation of (Actual value -predicted value )\*\*2/(Actual value-mean of actual value)\*\*2

ii)mean\_square\_error -> 1/n\*summation of (actual value – predicted value)\*\*2

iii)root\_mean\_square\_error -> square root of (mean\_square\_error)

iv)absolute\_mean\_square\_error -> 1/n \* summation of (actual value -predicted value)

**Note: project output in training data of r2\_score & MSE & RMSE & AMSE**

**Training data Performance:**

**Train Accuracy with r2\_score fun is :0.9999999812764105**

**Train Loss with MSE fun is : 2.1990016683893954**

**Train data r2\_score means the Accuracy is calculate manually : 0.9999999169519312**

**Mean square error is calculate manually : 2.2049242371757685**

**Root Mean Square Error in Train is calculated manually :1.484898729602719**

**Absolute Mean Square Error is calculate Manually : 2.199001668389395**

**Testing Data:**

5. we need to focus on Final assumption Testing Data so that Test data is give the algorithm is Linear Regression.

6.fiting the test data to the Algorithm then giving Model .

**Testing Performance:**

7.the Model is giving y\_test\_predicted values then create a new Data Frame and giving X\_test, y\_test, y\_test\_predicted collum’s to Data Frame .

8.**R2\_score function(Accuracy):**

**From sklearn.metrics import r2\_score**

**Syntax: r2\_score(y\_test ,y\_test\_predicted)**

Then it will giving Accuracy of the test Data is 0.9999999806028682

9.**Mean\_square\_error(Loss):**

**From sklearn.metrics import mean\_square\_error**

**Syntax:mean\_square\_error(y\_test,y\_test\_predicted)**

Then it will give loss of test Data is 2.0943696031557573

Without using r2\_score best to using score function

Syntax: reg.score(X\_test,y\_test)

Then it will calculate the predict function and then get predicted values and next calculate Accuracy is the test data giving Accuracy value only not showing predicted values.

10.Manually Calculating Accuracy and Losses and Formula’s

i)r2\_score-> 1-summation of (Actual value -predicted value )\*\*2/(Actual value-mean of actual value)\*\*2

ii)mean\_square\_error -> 1/n\*summation of (actual value – predicted value)\*\*2

iii)root\_mean\_square\_error -> square root of (mean\_square\_error)

iv)absolute\_mean\_square\_error -> 1/n \* summation of (actual value -predicted value)

**Note: project output in testing data of r2\_score & MSE & RMSE & AMSE**

**Testing data Performance:**

**Test Accuracy with functions is : 0.9999999806028682**

**Test Loss is with functions is : 2.0943696031557573**

**Test data r2\_score (Accuracy) calculated manually: 0.9999998837487859**

**Mean Square Error (Test) calculated manually: 2.094369603155757**

**Root Mean Square Error (Test) calculated manually: 1.4471936992523693**

**Absolute Mean Square Error (Test) calculated manually: 2.094369603155757**

**Finaly Conclusion:**

After calculating Training and Testing data performances and comparation between them there is very very low difference so the data set is best for working in project we will get the best Accuracy and very very low Loss values .