CAR_PURCHASING_DATA
Project Using
Multiple Linear Regression

Project Overview

Introduction

The goal of this project is to analyse a dataset related to car purchases using multiple linear regression. The primary objective is to understand the factors that influence car prices and to predict car prices based on various features. This analysis aims to provide valuable insights for potential buyers, car dealerships, and market analysts by identifying key determinants of car prices and offering a model for price prediction.

Objectives

1. Understand Influencing Factors: Identify and quantify the impact of different features on car prices. This includes examining variables such as car make, model, age, mileage, engine size, and other relevant attributes.

2. Predict Car Prices: Develop a predictive model that can estimate car prices based on the features provided. This model will help potential buyers and sellers make informed decisions.

3. Evaluate Model Performance: Assess the effectiveness of the regression model using appropriate evaluation metrics, ensuring that it provides accurate and reliable predictions.

Scope

1. Data Collection: Utilize a dataset containing information on car purchases, including both the features of the cars and their corresponding prices.

2. Feature Analysis: Analyse various features to determine their relevance and impact on car prices.

3. Model Development: Implement multiple linear regression to build a predictive model.

4. Validation: Use statistical methods to validate the model's performance and reliability.

Expected Outcomes

1. Feature Importance: Insights into which car attributes are most influential in determining prices.

2. Price Prediction Model: A robust regression model capable of predicting car prices with high accuracy.

3. Actionable Insights: Recommendations for buyers and sellers based on the model's findings, such as the impact of specific features on price.

Data Description

Data Source

The data source is not a private one we can access the data set from the website called Kaggle which is very much useful for us to download any type of dataset in easily. coming to this particular project dataset just go to the Kaggle website and search for the car purchasing data set in the search but and then you get much more datasets regarding to them. just open the first dataset and click on the download button. That’s your dataset will be downloaded and stores in the device in the form of zip file. You just extract the zip file. That’s the data source for our project will get ready.

Data Description

The dataset will be in the form of excel sheet and it consists of 500 rows and 9 columns. the column names are given below.

They are:

1. Customer Name
2. Customer E-mail Id
3. Country
4. Gender
5. Age
6. Annual Salary
7. Credit Card Debt
8. Net Worth
9. Car Purchase Amount

Data Cleaning

Now , we have to check weather the data in the dataset is clean or not. If it is clean then there will be no issue. We can go further but, if you are having any wanted features or null values or unnamed features in the dataset means we will get errors. So, we have clean the data. From our dataset we are having some unwanted features so, we can remove them. The unwanted features are shown below.

They are :

1. Customer Name
2. Customer E-mail Id
3. Country

The above 3 features are useless since, there will be no use for us regarding the customer name, e-mail, country since, we are just predicting the purchase amount of the car. So, we can remove those features.

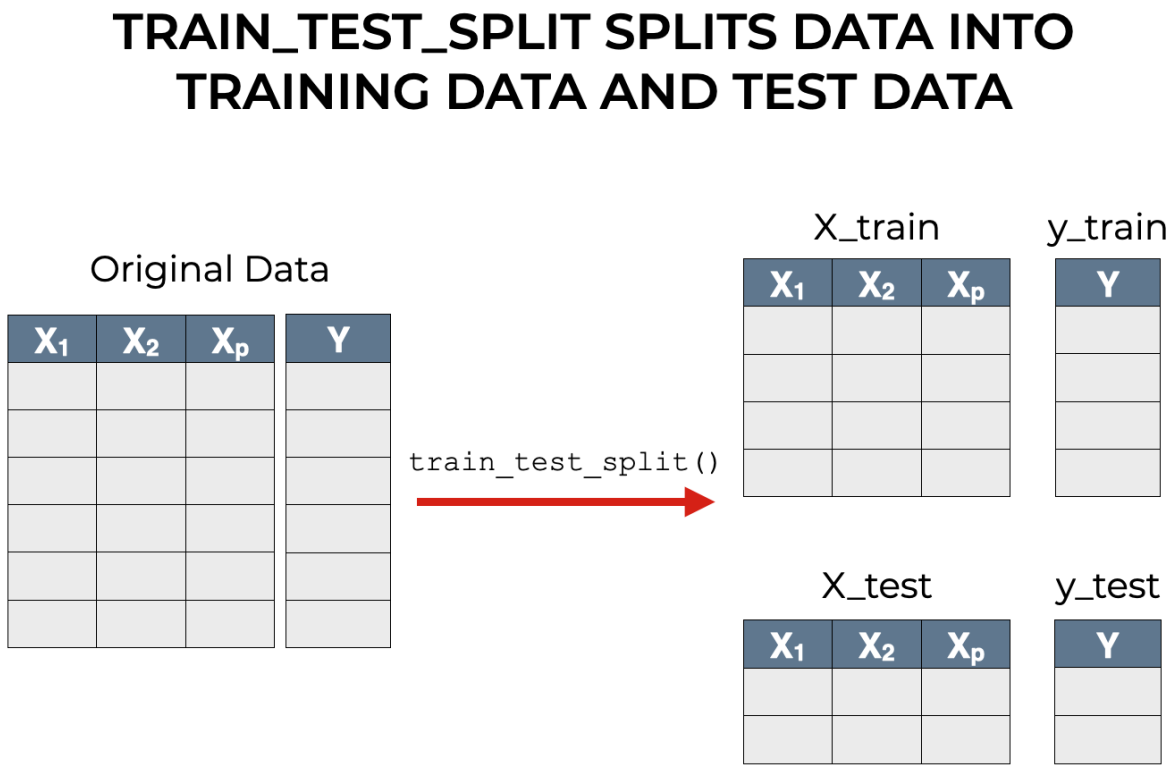
By using pandas library we read the dataset in our idle software and in pandas there is a function called drop to remove the features from the existing data set. Upto Now we have cleaned the unwanted data from the dataset.

Methodology

Model Selection

The first and foremost thing is we should split the cleaned data set to give the data to the algorithm for the purpose of getting the module and the prediction values. This **train test split model** should be imported from the sklearn library which is used to perform operation in machine learning. It will do all types of mathematical operations.

This train test split will divides the independent and dependent data to the training part and the testing part. This divided data we will give to the regression.



Regression

Regression is a statistical approach used to analyse the relationship between a dependent variable (target variable) and one or more independent variables (predictor variables). The objective is to determine the most suitable function that characterizes the connection between these variables. It is a supervised machine learning technique, used to predict the value of the dependent variable for new, unseen data. It models the relationship between the input features and the target variable, allowing for the estimation or prediction of numerical values.

Mainly there are 4 types of regression

There are :

1. Simple Linear Regression
2. Multiple Linear Regression
3. Regularization
4. Polynomial Linear Regression

Firstly, linear regression is nothing but a straight line.

i.e. y=mx + c

To the operation on linear regression. We have to **import linear regression** class from the **model selection** function from the **sklearn library**.

Simple Linear Regression:

We have only one independent and one dependent feature for the simple linear regression.

Multiple Linear Regression:

We have only one dependent and multiple independent features for multiple linear regression.

Regularization:

It is used to reduce the overfitting problem. It can be implemented in two different mathematics i.e. regularization is divided into two types .

They are:

1. Ridge Regression
2. Lasso Regression

Ridge Regression:

It is also known as L2 Regression. It tries to increase the accuracy of the data and decreases the loss in the data.

Lasso Regression:

It is also known as L1 Regression. It tries to increase the accuracy of the data and decreases the loss in the data and also in any case the slope values becomes zero then it will remove that particular column from the data.

Polynomial Regression:

A form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial in x.

Complete Operation of the dataset:

Step 1:

Firstly, we read the dataset by giving it path and also by using pandas we can view the dataset in it’s original form.

Step 2:

Then we should check the dependent and independent features by checking this we can come to know weather it is simple linear regression or multiple linear regression.

Step 3:

Now, check weather there is unwanted features in the dataset. If it is there then remove the feature.

Step 4:

Divide the independent and dependent features into two different variables i.e. X , y and then give these two variables to the train test split module and in it add test size and random state. Here, test size is nothing but giving value that how much the test data should be divided and random state will protect the data which is already executed without changing them every step of running.

Step 5:

Then import the linear regression class and give the splitted features by the train test split by this the dataset will be given to the algorithm and then the algorithm will be trained**.**

Step 6:

And then predict the values by giving the independent features. And then find the accuracy for the original values and predicted values by importing r2_score function from the metrics method from the sklearn library.

Step 7:

Also find the loss by giving the original values and predicted values to the loss functions. There are the loss functions they are mean squared error, absolute

square error, R squared error.by importing the loss function from metrics from the sklearn.

Step 8:

Repeat the above 5,6,7 steps by giving test data instead of train data so, that you will get test data accuracy and loss.

Step 9:

From the above steps we will get the exact results of the given dataset so, that we can decide weather the dataset is good or bad.

Source Code For The Project

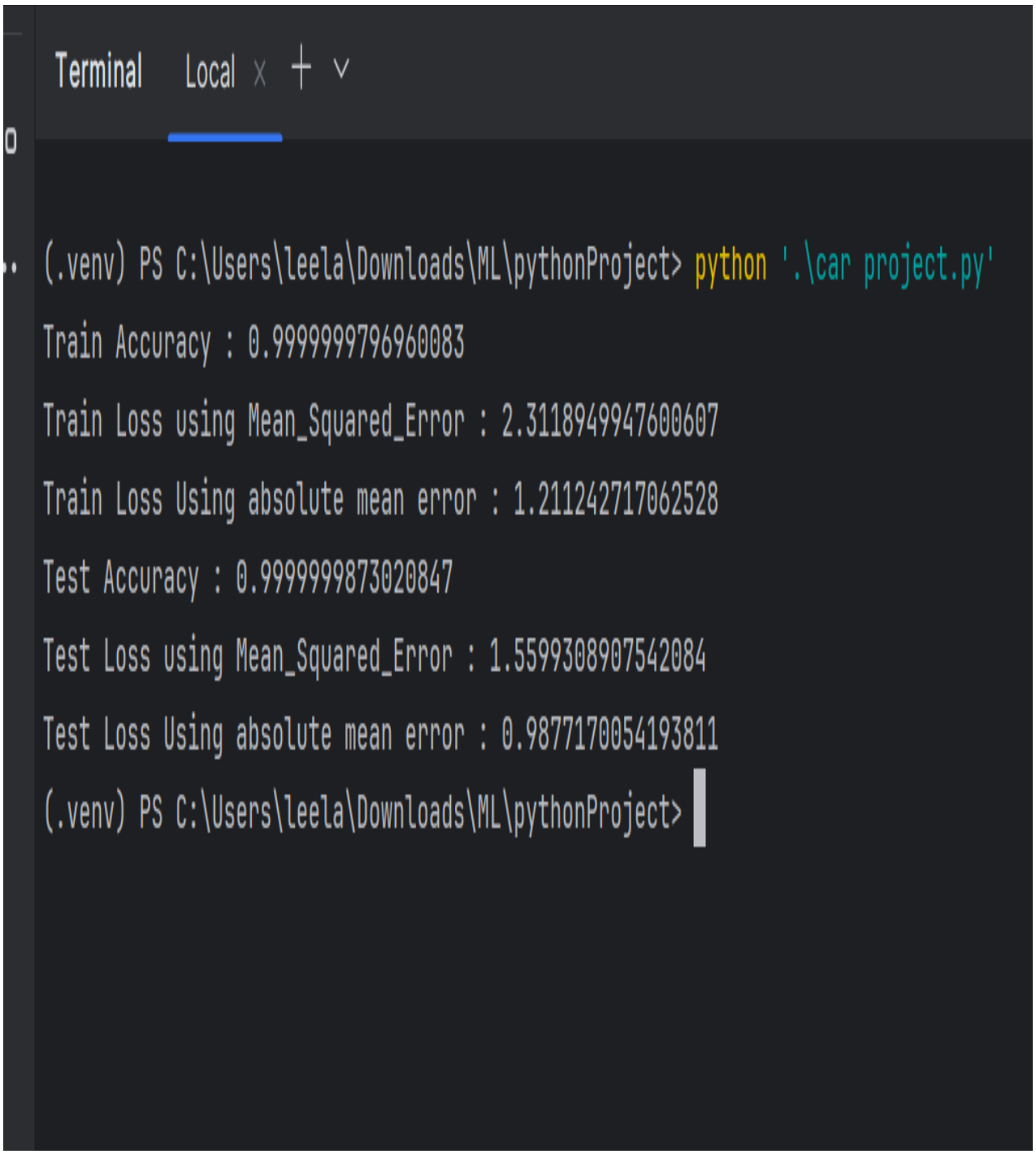
```
'''
In this file we are going to develop and Multiple Linear Regression with Oops concept
'''
import numpy as np
import pandas as pd
import sklearn
import sys
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error



class TRAIN:
```

```
    def __init__(self, file):
        try:
            self.data = pd.read_csv(file, encoding='latin1')
            # self.print(data)
            self.data = self.data.drop(['Customer Name', 'Customer e-mail', 'Country'], axis=1)
            # print(self.data)
            self.X = self.data.iloc[:, :-1]
            self.y = self.data.iloc[:, -1]
            # print(self.X)
            self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(self.X, self.y,
test_size=0.2, random_state=2)
        except Exception as e:
            error_type, error_msg, err_line = sys.exc_info()
            print(f'Error from Line {err_line.tb_lineno} -> type {error_type} -> Error msg ->
{error_msg}')


    def TRAINING(self):
        try:
            self.reg = LinearRegression()
            self.reg.fit(self.X_train, self.y_train)
        except Exception as e:
            error_type, error_msg, err_line = sys.exc_info()
            print(f'Error from Line {err_line.tb_lineno} -> type {error_type} -> Error msg ->
{error_msg}')
```

```
    def TRAINED_PERFORMANCE(self):
        try:
            self.y_train_pred = self.reg.predict(self.X_train)
            print(f'Train Accuracy : {r2_score(self.y_train, self.y_train_pred)}')
            print(f'Train Loss using Mean_Squared_Error : {mean_squared_error(self.y_train, self.y_train_pred)}')
            print(f'Train Loss Using absolute mean error : {mean_absolute_error(self.y_train, self.y_train_pred)}')
        except Exception as e:
            error_type, error_msg, err_line = sys.exc_info()
            print(f'Error from Line {err_line.tb_lineno} -> type {error_type} -> Error msg -> {error_msg}')

    def TESTING(self):
        try:
            self.reg = LinearRegression()
            self.reg.fit(self.X_test, self.y_test)
        except Exception as e:
            error_type, error_msg, err_line = sys.exc_info()
            print(f'Error from Line {err_line.tb_lineno} -> type {error_type} -> Error msg -> {error_msg}')

    def TEST_PERFORMANCE(self):
        try:
            self.y_test_pred = self.reg.predict(self.X_test)
```

```
        print(f'Test Accuracy : {r2_score(self.y_test, self.y_test_pred)}')
        print(f'Test Loss using Mean_Squared_Error : {mean_squared_error(self.y_test, self.y_test_pred)}')
        print(f'Test Loss Using absolute mean error : {mean_absolute_error(self.y_test, self.y_test_pred)}')
    except Exception as e:
        error_type, error_msg, err_line = sys.exc_info()
        print(f'Error from Line {err_line.tb_lineno} -> type {error_type} -> Error msg -> {error_msg}')


if __name__ == "__main__":
  try:
    sri = TRAIN('C:\\Users\\leela\\Downloads\\ML\\pythonProject\\Car_Purchasing_Data.csv')
    sri.TRAINING()
    sri.TRAINED_PERFORMANCE()
    sri.TESTING()
    sri.TEST_PERFORMANCE()
  except Exception as e:
    error_type, error_msg, err_line = sys.exc_info()
    print(f'Error from Line {err_line.tb_lineno} -> type {error_type} -> Error msg -> {error_msg}')
```

Outcome of the project



Conclusion

In this project, we applied a multiple linear regression model to predict car purchase amount based on various factors such as age, gender, annual salary, credit card debt, net worth. The goal was to understand how these variables influence car purchase amount and to develop a model that can estimate the purchase amount of a car given its characteristics.