

Project Report

On

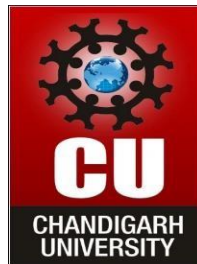
Linear Model To Predict Car Price

Submitted for the requirement of

Project course

BACHELOR OF ENGINEERING

COMPUTER SCIENCE & ENGINEERING



**Submitted to:
Ms. Parampreet kaur**

**Submitted By:
Dharmendra Yadav-
21BCS11791
Riya Battu-21BCS729**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CHANDIGARH UNIVERSITY, GHARUAN
November 2023**

CERTIFICATE

This is to certify that the work embodied in this Project Report entitled “**LINEAR MODEL TO PREDICT CAR PRICE**” being submitted by “**21BCS11791**” “**Dharmendra Yadav**”, 5th Semester for partial fulfillment of the requirement for the degree of “**Bachelor of Engineering in Computer Science & Engineering**” discipline in “**Chandigarh University**” during the academic session Aug-Dec 2021 is a record of bonafide piece of work, carried out by student under my supervision and guidance in the “**Department of Computer Science & Engineering**”, Chandigarh University.

APPROVED & GUIDED BY:

Ms. Parampreet Kaur

E10366

DECLARATION

I, student of **Bachelor of Engineering in Computer Science & Engineering, 5th Semester** , session: **Jan – June 2021, Chandigarh University**, hereby declare that the work presented in this Project Report entitled “ **LINEAR MODEL TO PREDICT CAR PRICE** ” is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Student details and Signature

<u>SN.</u>	<u>Name</u>	<u>UID</u>	<u>Contact</u>
1.	Riya Batu	21BCS11729	8756342190
2.	Dharmendra yadav	21BCS11791	7766080235

APPROVED & GUIDED BY:

MS. Parampreet kaur

E10366

Synopsis for Linear Model to Predict Car Price

1. Introduction:

Linear module to predict car price is somehow interesting and popular. Car price predictions involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, whether it has navigation or not will also influence the car price. And this module is used mostly in car companies. In this project we are using machine learning and linear regression in order to achieve higher precision of car price prediction.

This project is made in the following manner :

First we import necessary libraries on Jupyter notebook which is required in this project , then collect the valid dataset with maximum number of data which can be useful in implementing this project. After tha collected data is being analyzed. And then we build our Linear Regression Module and after module is being prepared we create a dataframe. And at last after creating the dataframe finally we implement the project.

2. Feasibility Study:

As we know feasibility study introduces on cost ,budget ,and technology wise. This case of study clearly Estimates cost and budget required for project implementation. It makes Estimation of cost of requirement and budget needed to utilize in order to development of project. If we are trying to build big project then we need lot of teams member And need to have paid software which is tremendously increases budget for the project. Incase there will be less team member budget will decrease.

3. Need of LINEAR MODULE TO PREDICT CAR PRICE:

- Helps the management to understand how exactly the prices vary with the independent variables
- Manipulate the design of the cars, the business strategy etc. to meet certain price levels.
- Helps the developers to use datasets and build more modules and develop projects.

4. Methodology/ Planning of work :

1. The project contains Number of components, modules and libraries to make coding easier and it will make developer easy to understand.
2. The project will be developed using the existing dataset collected from anywhere which is valid.
3. It uses Linear module Regression to create dataframe.
4. The module of this project will be deployed in some type of car business website.

5. Innovations in Project:

Generally, We are building Linear Module to Predict car price using linear regression and enhance its functionality and improve in performance. We would likely to implement complex module which can perform well in real world and there wouldn't be any technical error. Also, we are likely to deploy this module in the website where it can be implemented efficiently.

5. Software and Hardware Requirements:

Hardware	Type
Processor	Intel Core Processor or better performance
Primary Memory	1 GB or more
Secondary Memory	3 GB or more Not Specified
Graphics	Not Required
Printer	Not Required

Software	Type
Operating System	Windows and Linux Operating System
Compiler	Anaconda(Jupyter Notebook)

Bibliography

- <https://towardsdatascience.com/predicting-car-price-using-machine-learning-8d2df3898f16>
- <https://www.kaggle.com/ngawangchoeda/linear-models-to-predict-car-price>

Teamwise distribution of work

By Rupa kumari-19BCS2271

- Data collection along with teammate
- Data analysis
- Data cleansing
- Coding database for data interpretation

By Nitesh Kumar sah-19BCS2271

- Data collection along with teammate
- Data preprocessing'
- Building the models
- Coding in the models to find out the accuracy and statistical value

As this project is all about coding line by line and getting the output, more than doing the work separate, the work is done together by both member. At first, playing with the data is needed which is to be done by both and in the further process too, each other needs to know the output and can change according to the requirement as the project is all about getting the statistical accuracy. One's task is to completely known and understood by the other work.

ABSTRACT

As in this modern world with the advancement of modern technologies especially in the field of transport, the price of vehicles is on hype and there are lot of issue in detecting and predicting the price of vehicle especially those of which are second hand and is to be compared. Our project, Linear Model to predict the car price is all about building a models and comparing all models and find out which model is so accurate to predict the price by its features analysis and giving more accuracy. Playing with the data is much more important here that the more changes and accuracy in the data helps in giving the more proper statistical value and accuracy of the models.

Example of this model in real life : If a person goes to buy or sell the second hand car , there might be difficulty in predicting the actual price. So , the models build here will help to compare the feature of that car with the new one and detect how the features are working and according to the correlation of features , the price will be predicted.

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1 Introduction with objectives and scope
- 1.2 Problem introduction
- 1.3 Libraries used

2. REQUIREMENTS SPECIFICATION

- 2.1 Introduction
- 2.2 Hardware requirements
- 2.3 Software requirements

3. ANALYSIS

- 3.1 Existing System
- 3.2 Proposed System
- 3.3 Feasibility study

4. DESIGN

- 4.1 System Design
- 4.2 Modules in project

5. SYSTEM IMPLEMENTATION

- 5.1 Introduction
- 5.2 Sample code

6. SAMPLE SCREENSHOTS

7. CONCLUSION

8. BIBLOGRAPHY

1. INTRODUCTION:

1.1 Introduction

Linear module to predict car price is somehow interesting and popular. Car price predictions involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, whether it has navigation or not will also influence the car price. And this module is used mostly in car companies. In this project we are using machine learning and linear regression in order to achieve higher precision of car price prediction.

This project is made in the following manner :

First we import necessary libraries on Jupyter notebook which is required in this project , then collect the valid dataset with maximum number of data which can be useful in implementing this project. After the collected data is being analyzed. And then we build our Linear Regression Module and after module is being prepared we create a dataframe. And at last after creating the dataframe finally we implement the project.

The project is being prepared with data science , Python programming language and some specification of database management system.

Data Science: Data science is the process of building, cleaning, and structuring datasets to analyze and extract meaning. It's not to be confused with data analytics, which is the act of analyzing and interpreting data. These processes share many similarities and are both valuable in the workplace. Data science requires you to Form hypotheses, Run experiments to gather data, Assess data's quality, Clean and streamline dataset and Organize and structure data for analysis. Data science can be used to gain knowledge about behaviors and processes, write algorithms that process large amounts of information quickly and efficiently, increase security and privacy of sensitive data, and guide data-driven decision-making. Data science uses techniques such as machine learning and artificial intelligence to extract meaningful information and to predict future patterns and behaviors. Advances in technology, the internet, social media, and the use of technology have all increased access to big data. The field of data science is growing as technology advances and big data

collection and analysis techniques become more sophisticated. Data is drawn from different sectors, channels, and platforms, including cell phones, social media, e-commerce sites, healthcare surveys, and internet searches. The increase in the amount of data available opened the door to a new field of study based on big data—the massive data sets that contribute to the creation of better operational tools in all sectors. The continually increasing access to data is possible due to advancements in technology and collection techniques. Individuals buying patterns and behavior can be monitored and predictions made based on the information gathered. However, the ever-increasing data is unstructured and requires parsing for effective decision-making. This process is complex and time-consuming for companies—hence, the emergence of data science.

Python: Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. As Python continues to grow in popularity and as the number of data scientists continues to increase, the use of Python for data science will inevitably continue to grow. As we advance machine learning, deep learning, and other data science tasks, we'll likely see these advancements available for our use as libraries in Python. Python has been well-maintained and continuously growing in popularity for years, and many of the top companies use Python today. With its continued popularity and growing support, Python will be used in the industry for years to come.

Database: A database is a data structure that stores organized information. Most databases contain multiple tables, which may each include several different fields. For example, a company database may include tables for products, employees, and financial records. Each of these tables would have different fields that are relevant to the information stored in the table. Nearly all e-commerce sites use databases to store product inventory and customer information. These sites use a database management system (or DBMS), such as Microsoft Access, FileMaker Pro, or MySQL as the "back end" to the website. By storing website data in a database, the data can be easily searched, sorted, and updated. This flexibility is important for e-commerce sites and other types of dynamic websites. Early databases were relatively "flat," which means they were limited to simple rows and columns, like

a spreadsheet. (See also "flat file database"). However, today's relational databases allow users to access, update, and search information based on the relationship of data stored in different tables. Relational databases can also run queries that involve multiple databases. While early databases could only store text or numeric data, modern databases also let users store other data types such as sound clips, pictures, and videos.

Objectives:

- Helps the management to understand how exactly the prices vary with the independent variables
- the design of the cars, the business strategy etc. to meet certain price levels.
- Helps the developers to use datasets and build more modules and develop projects.

Scope:

- The project and this model can be deployed in website of the same field.
- Vehicles showrooms can use this model to predict the actual price without any difficulty
- It will make impact on trading of vehicles by making it easy to predict price.

1.2 Problem introduction

As this project is all about getting the statistical value as a result, there were lots of problems to carry on this project and after the building of the project too, there might be several issues as discussed below:

- **Variation in the dataset used:** one project developer might use one dataset where as another might use another. So, the variation in the dataset plays a vital role in giving the accuracy of the project. According to our dataset, we got the result but if others use more valid dataset than us, they might get the better result. The problem we faced is making an appropriate dataset and collecting it to get a better result which took more time.
- **Including the features:** As mentioned above in dataset, the features included in dataset also play a vital role. Different features contribute in different ways to build a model and if the major feature is missed, then there will be huge error in the result.

- **Analysis of dataset:** Data analysis and data cleansing was a quite difficult task to do. At first, we must make a proper tree of dataset either by including or excluding the features or other requirement.
- **Plotting the graph and comparing:** After the building of model, the hard task is to compare the models according to their trained score, testing score, coefficient correlation of features and graph.

1.3 Libraries Used

There are countless libraries like NumPy, Pandas, and Matplotlib available in Python to make data cleaning, data analysis, data visualization, and machine learning tasks easier. Some of the most popular libraries in this project include:

- **NumPy:** NumPy is a Python library that provides support for many mathematical tasks on large, multidimensional arrays and matrices.
- **Pandas:** The Pandas library is one of the most popular and easy-to-use libraries available. It allows for easy manipulation of tabular data for data cleaning and data analysis.
- **Matplotlib:** This library provides simple ways to create static or interactive boxplots, scatterplots, line graphs, and bar charts. It's useful for simplifying your data visualization tasks.
- **Seaborn:** Seaborn is another data visualization library built on top of Matplotlib that allows for visually appealing statistical graphs. It allows you to easily visualize beautiful confidence intervals, distributions, and other graphs.
- **Statsmodels:** This statistical modeling library builds all of your statistical models and statistical tests including linear regression, generalized linear models, and time series analysis models.
- **Scikit-learn:** This popular machine learning library is a one-stop-shop for all of your machine learning needs with support for both supervised and unsupervised tasks. Some of the machine learning algorithms available are logistic regression, k-nearest neighbors, support vector machine, random forest, gradient boosting, k-means, DBSCAN, and principal component analysis.
- **Tensorflow:** Tensorflow is a high-level library for building neural networks. Since it was mostly written in C++, this library provides us with the simplicity of Python without sacrificing power and performance. However, working with raw Tensorflow is not suited for beginners.

2. REQUIREMENT SPECIFICATION

2.1- INTRODUCTION: To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

2.2-HARDWARE REQUIREMENTS: The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

Hardware	Type
Processor	Intel Core Processor or better performance
Primary Memory	1 GB or more
Secondary Memory	3 GB or more Not Specified
Graphics	Not Required
Printer	Not Required

2.3 SOFTWARE REQUIREMENTS: Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT

Software	Type
Operating System	Windows and Linux Operating System
Compiler	Anaconda(Jupyter Notebook)
Language	Python, Database

3. ANALYSIS

3.1 Existing system

There might be lots of projects and models based on predicting the car price. The existing models might be different in comparisons to others. As we have discussed above, this model is completely dependent upon the type of dataset used. If the developers had used better dataset, then the accuracy and statistical result might be appropriate of the existing models. As researched by us, the developers building this model lags behind the proper dataset used but their statistical values are very good as they had cleansed the data so appropriately though the data was improper.

3.2 Proposed System

We have proposed to build our model appropriate which can give us the fruitful result. We have tried to prepare a proper appropriate dataset than the existing models. We cleansed the data properly and have got a proper accuracy and statistical values better than most of the existing developers.

3.3 Feasibility study

As we know feasibility study introduces on cost ,budget, and technology wise. This case of study clearly Estimates cost and budget required for project implementation. It makes Estimation of cost of requirement and budget needed to utilize in order to development of project. If we are trying to build big project then we need lot of teams member And need to have paid software which is tremendously increases budget for the project. In case there will be less team member budget will decrease.

4. DESIGN

4.1 system Design

We are building Linear Module to Predict car price using linear regression and enhance its functionality and improve in performance. We would likely to implement complex module which can perform well in real world and there wouldn't be any technical error. Linear model to predict car price is a project designed under the linear model. With the models , it contains number of packages like matplotlib, seaborn, sklearn, etc from which different modules are imported. The project is designed as per the discussed algorithm below as it does not have any kinds of DFDs and UMLs. The innovative idea in this project is that the more huge and validate dataset has been used with more various features which will make impact on the accuracy and statistical output of this model. As the linear regression model with Ridge and Lasso are used here, more comparisons among the features can be made and the proper features are taken out which makes the high impact on the output.

Instead of DFDs, UML and flowchart, this model can be described with its algorithm in a effective way. The algorithm is discussed below:

1. Start
2. Import libraries from packages
3. Input

Here, we input the collected valid dataset which is the most required in this model.

4. Exploratory Data Analysis(EDA)

Here, the data is being analysed like how many features does it contains with its rows and columns. Its shows the information regarding the datasets. Also, from this analysis we can differentiate the independent and dependent data used here.

5. Data Cleansing

We can differentiate the independent and dependent data used here by cleaning the data and its related to EDA.

6. Data Preprocessing

Here, the data is being separated as the numeric data and the categorical data also called the string data. The string data is being converted into the numeric through binary process and then concatenate with numeric data.

7. Model Building

Three linear models are being built called Linear Regression, ridge and Lasso and those models are regularized and plotted into the graph which shows the statistical data.

8. Training Model

The data is being trained with the help of ridge model which is being further analysed and tested.

9. Testing Model

After the data is trained, it is tested with the help of lasso model which makes the impact on output.

10. Output(form of accuracy of the model)

This model can also be deployed in any kinds of related website and used in the practical life.

4.2 Modules in project

Linear model to predict car price is a project designed under the linear model. With the models, it contains number of packages like matplotlib, seaborn, sklearn, etc from which different modules are imported. In this project, we have used three linear models imported from sklearn.linear_model which are **Linear Regression()**, **Ridge()** and **Lasso()**. These three models are discussed below:

- **Linear Regression()**

Linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables. The case of one explanatory variable is called simple linear regression and for more than one is called multiple linear Regression. In this project where the statistical values are to be calculated, linear representation of data is much needed. Linear Regression is an attractive model because the representation is simple. The representation is an linear equation that

combines a specific set of input values (x) the solution to which is the predicted output for that set of input values(y). **Linear Regression(aka ordinary least squares)**: Simplest & most classic linear method for regression. It finds the parameters w & b that minimize the mean square error between predicted value and true value.

$$y=wx + b$$

w->Weights associated with individual independent features(Slope of a line)

b->y intercept

- **Ridge()**

Ridge Regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values. For any type of Ridge Regression machine learning models, the usual regression equation forms the base which is written as:

$$Y = XB + e$$

Where Y is the dependent variable, X represents the independent variables, B is the regression coefficients to be estimated, and e represents the errors or residuals. It is a linear model which uses **L2 regularization** technique.

Regularization techniques explicitly restrict a model to avoid overfitting.

LinearRegression() does not allow us to control its complexity so it's very likely that it will overfit the models when the dataset is relatively small. **L2**

regularization reduces the coefficient of the independent features to small magnitude as possible i.e. all entries of w should be close to zero. Ridge has an alpha parameter which makes a trade-off between the simplicity of the model and its performance on training set & hence tuning it will yield different model performance.

- **Lasso()**

The word “LASSO” stands for **Least Absolute Shrinkage and Selection Operator**. It is a statistical formula for the regularisation of data models and feature selection. Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. It is a linear model which uses **L1** however unlike Ridge it reduces magnitude of some of the features to zero. Hence it neglects some of the features completely. Hence it is also used for automatic feature selection as it ignores some of the features. Lasso also have alpha parameter which makes a trade-off between the simplicity of the model and its performance on training set & hence tuning it will yield different model performance.

Comparing with the others existing models of this project, the models are quite similar but the project is whole dependent in the accuracy of the dataset used and how much the accuracy we can get as a output. Here, we have tried to use the huge and validate dataset through which the statistical accuracy and results can be more efficient than the existing ones. If the project give the proper statistical value and in the end result if more accuracy is obtained, the project can be deployed in at any kinds of website of relatable field.

5. SYSTEM IMPLEMENTATION

5.1 Introduction

As the project is built on jupyter lab, after every code and line there is output. The codes are ran on jupyter lab because the model has to give a statistical value and it must change according to the developers change in dataset .

5.2 Sample Code

As there is no particular single set of code, after every code there is some result, so the code is shown in screenshots in the below section.

The samples of some are below:

Blackboard Learn Set Status Reading list

Settings Help

linear-models-to-predict-car X CarPrice.csv

Python 3

[1]:
import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

Import necessary libraries for the project

[2]:
#matplotlib and seaborn are imported for visualization
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

#splitting the dataset into train & test data
from sklearn.model_selection import train_test_split

#GridSearchCV is used for hyperparameter tuning in Lasso & Ridge
from sklearn.model_selection import GridSearchCV

#three Linear models used in the project
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso

#StandardScaler for preprocessing the dataset
from sklearn.preprocessing import StandardScaler

#metrics to evaluate the Linear regression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error

#import warnings to ignore any warnings during execution
import warnings
warnings.filterwarnings('ignore')

Idle Saving completed Mode: Command Ln 1, Col 1 linear-models-to-predict-car-price.ipynb 15:33 36°C Haze ENG 11-10-2021

Blackboard Learn Set Status Reading list

Settings Help

linear-models-to-predict-car X CarPrice.csv

Python 3

[24]:
#the function takes model, train & test split as an argument
def fit_model_getScores(model,X_train,y_train,X_test,y_test):
 #fit the model with training dataset
 model.fit(X_train,y_train)

 #score the training data
 train_score=model.score(X_train,y_train)
 #score the test data
 test_score=model.score(X_test,y_test)

 #Display the scores
 print("Scores of {}".format(model),"\n")
 print("Training Score:{:.2f}".format(train_score))
 print("Testing Score:{:.2f}".format(test_score))

The function given below will help return the **metrics** used for **evaluating linear models** & that includes **mse,mae,rmse,r2_score**

[25]:
#function takes model, and test data split as an argument
def get_metrics(model,X_test,y_test):
 #calculate the predicted value of y
 y_pred=model.predict(X_test)
 mse=mean_squared_error(y_test,y_pred)#mse
 r2_score=r2_score(y_test,y_pred)#r2_score
 mae=mean_absolute_error(y_test,y_pred)#mae
 rmse=mean_squared_error(y_test,y_pred,squared=False)#rmse

 #print the Metrics
 print("The Metrics for {}".format(model))
 print("-----")
 print("Mean Squared Error:{:.2f}".format(mse))
 print("Root Mean Squared Error:{:.2f}".format(rmse))

```
[24]: #the function takes model, train & test split as an argument
def fit_model_getScores(model,X_train,y_train,X_test,y_test):
    #fit the model with training dataset
    model.fit(X_train,y_train)

    #score the training data
    train_score=model.score(X_train,y_train)
    #score the test data
    test_score=model.score(X_test,y_test)

    #Display the scores
    print("Scores of {}".format(model),"\n")
    print("Training Score: {:.2f}".format(train_score))
    print("Testing Score: {:.2f}".format(test_score))

The function given below will help return the metrics used for evaluating linear models & that includes mse,mae,rmse,r2 score

[25]: #function takes model, and test data split as an argument
def get_metrics(model,X_test,y_test):
    #calculate the predicted value of y
    y_pred=model.predict(X_test)
    mse=mean_squared_error(y_test,y_pred)#mse
    r2_score=r2_score(y_test,y_pred)#r2_score
    mae=mean_absolute_error(y_test,y_pred)#mae
    rmse=mean_squared_error(y_test,y_pred,squared=False)#rmse

    #print the metrics
    print("The Metrics for {}".format(model))
    print("-----")
    print("Mean Squared Error: {:.2f}".format(mse))
    print("Root Mean Squared Error: {:.2f}".format(rmse))
```

```
We got alpha=100 & max_iter=1000 for the lasso model

[42]: params={
    "alpha": [1e-9, 1e-6, 1e-3, 1, 100, 1000, 10000],
    "max_iter": [1e3, 1e4, 1e5, 1e6] #maximum number of iterations to run
}

lasso_best_params=gridSearch(Lasso_model,params,X_train_scaled,y_train)
lasso_best_params

[42]: {'alpha': 100, 'max_iter': 1000.0}

Lets try fitting the Lasso model using the parameters that have been returned from Hypertuning

[43]: lasso1_model=Lasso(*lasso_best_params)

The scores are 93% for training set and 86% for testing set which is better generalized model than the above two models i.e LinearRegression() & Ridge()

[44]: fit_model_getScores(lasso1_model,
    X_train_scaled,y_train,
    X_test_scaled,y_test
)

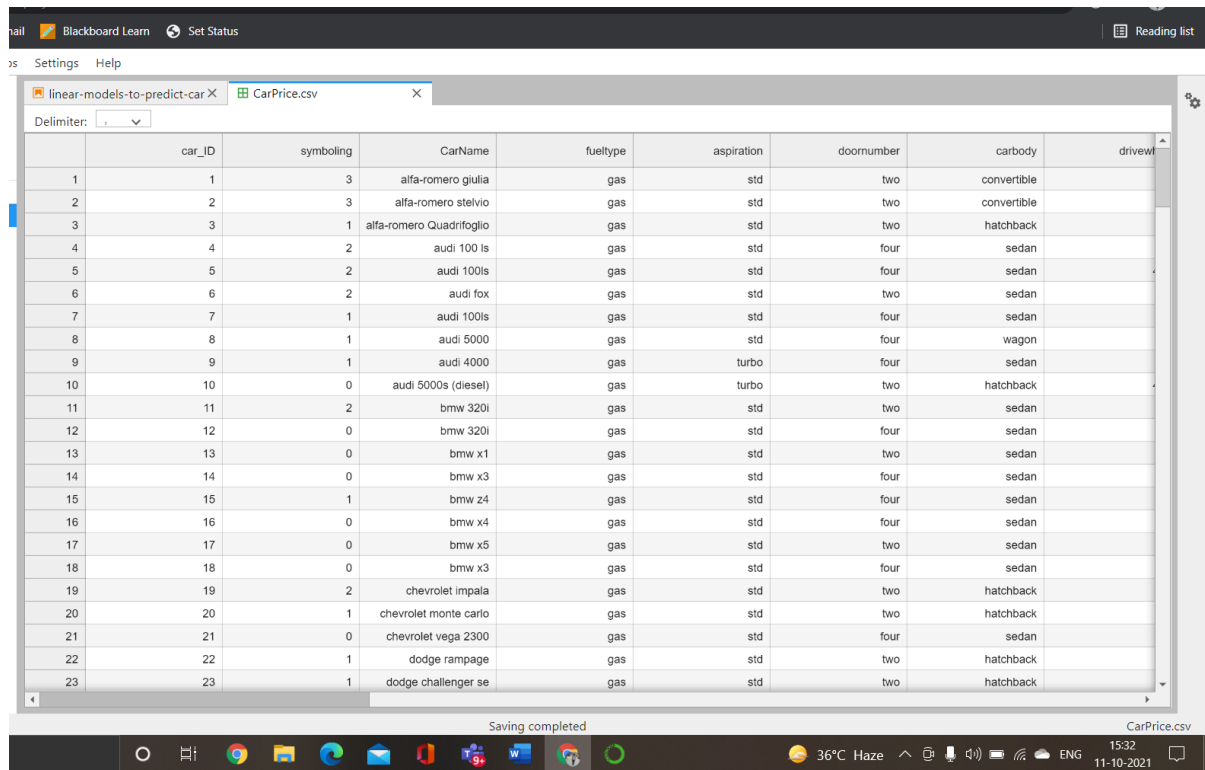
Scores of Lasso(alpha=100, max_iter=1000.0)

Training Score:0.93
Testing Score:0.86

Print the metrics of lasso model

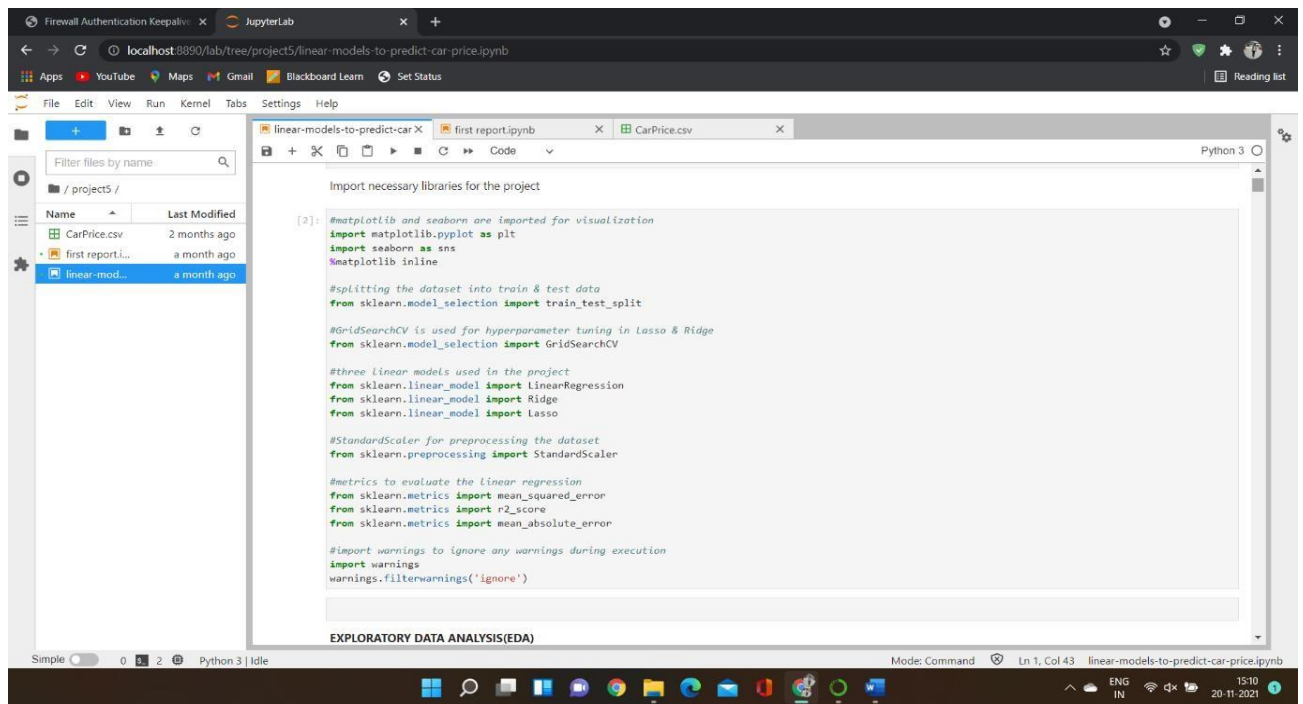
[45]: get_metrics(lasso1_model,
    X_test_scaled,y_test
)
```

6. SAMPLE SCREESSHOTS WITH EXPLANATION

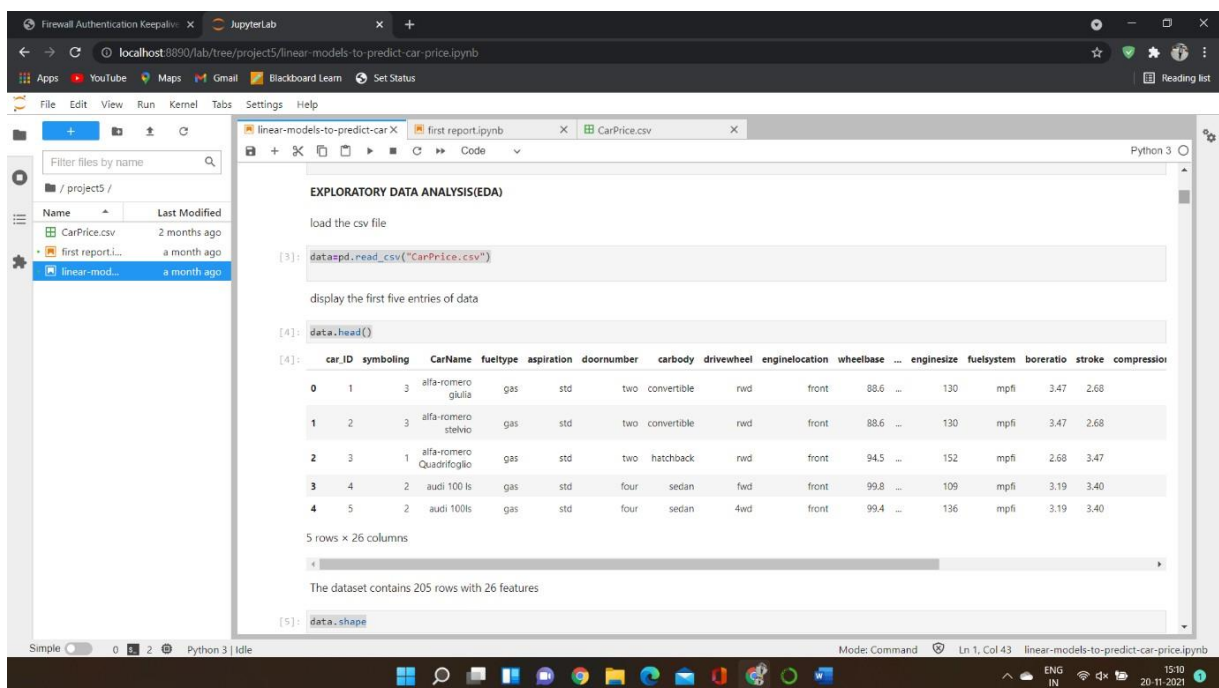


	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel
1	1	3	alfa-romero giulia	gas	std	two	convertible	
2	2	3	alfa-romero stelvio	gas	std	two	convertible	
3	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	
4	4	2	audi 100 ls	gas	std	four	sedan	
5	5	2	audi 100ls	gas	std	four	sedan	
6	6	2	audi fox	gas	std	two	sedan	
7	7	1	audi 100ls	gas	std	four	sedan	
8	8	1	audi 5000	gas	std	four	wagon	
9	9	1	audi 4000	gas	turbo	four	sedan	
10	10	0	audi 5000s (diesel)	gas	turbo	two	hatchback	
11	11	2	bmw 320i	gas	std	two	sedan	
12	12	0	bmw 320i	gas	std	four	sedan	
13	13	0	bmw x1	gas	std	two	sedan	
14	14	0	bmw x3	gas	std	four	sedan	
15	15	1	bmw 24	gas	std	four	sedan	
16	16	0	bmw x4	gas	std	four	sedan	
17	17	0	bmw x5	gas	std	two	sedan	
18	18	0	bmw x3	gas	std	four	sedan	
19	19	2	chevrolet impala	gas	std	two	hatchback	
20	20	1	chevrolet monte carlo	gas	std	two	hatchback	
21	21	0	chevrolet vega 2300	gas	std	four	sedan	
22	22	1	dodge rampage	gas	std	two	hatchback	
23	23	1	dodge challenger se	gas	std	two	hatchback	

This is the type of dataset being used in this model. It contains 11 rows and 26 columns. This dataset contains different features which can be used in predicting the car price after building the model. It is somewhere appropriate dataset which includes the proper values and requirements used in the project.

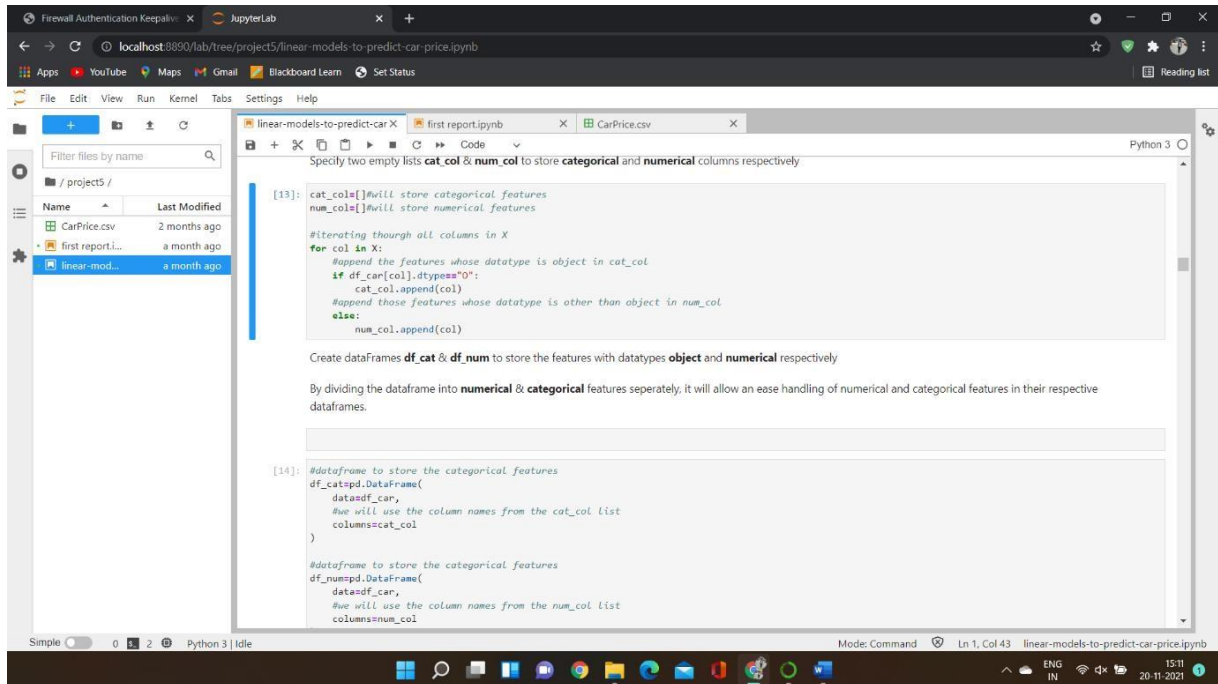


Different libraries such as matplotlib, seaborn, linear regression, ridge, lasso and more are imported from the packages.



Exploratory data analysis, the dataset used is analysed and cleansed. Data analysis like, the number of rows and column present in dataset, the required columns and rows. Data cleansing means to get the appropriate rows and

columns which are genuinely required for building the models and the rows columns which doesnot make impact on the building model are ignored.



```
[13]: cat_col=[]#will store categorical features
      num_col=[]#will store numerical features

      #iterating through all columns in X
      for col in X:
          #append the features whose datatype is object in cat_col
          if df_car[col].dtype=="O":
              cat_col.append(col)
          #append those features whose datatype is other than object in num_col
          else:
              num_col.append(col)

      Create dataframes df_cat & df_num to store the features with datatypes object and numerical respectively

      By dividing the dataframe into numerical & categorical features seperately, it will allow an ease handling of numerical and categorical features in their respective dataframes.

[14]: #dataframe to store the categorical features
      df_cat=pd.DataFrame(
          data=df_car,
          #we will use the column names from the cat_col list
          columns=cat_col
      )

      #dataframe to store the numerical features
      df_num=pd.DataFrame(
          data=df_car,
          #we will use the column names from the num_col list
          columns=num_col
```

The shown code above is the code for data framing. Data framing means to get the every data in same language. The data includes the features which might be of character value as well as numerical values. All the values are converted to binary form and then concatenated so that the further process can go on.

The screenshot shows a JupyterLab environment with a file browser on the left and a code editor on the right. The file browser shows a directory named 'project5' containing files 'CarPrice.csv', 'first report...', and 'linear-mod...'. The code editor displays the following Python code:

```
[24]: #the function takes model, train & test split as an argument
def fit_model_getScores(model,X_train,y_train,X_test,y_test):
    #fit the model with training dataset
    model.fit(X_train,y_train)

    #score the training data
    train_score=model.score(X_train,y_train)
    #score the test data
    test_score=model.score(X_test,y_test)

    #Display the scores
    print("Scores of {}".format(model),"\n")
    print("Training Score: {:.2f}".format(train_score))
    print("Testing Score: {:.2f}".format(test_score))

The function given below will help return the metrics used for evaluating linear models & that includes mse,mae,rmse,r2_score

[25]: #function takes model, and test data split as an argument
def get_metrics(model,X_test,y_test):
    #calculate the predicted value of y
    y_pred=model.predict(X_test)
    mse=mean_squared_error(y_test,y_pred)#mse
    r2_score=r2_score(y_test,y_pred)#r2_score
    mae=mean_absolute_error(y_test,y_pred)#mae
    rmse=mean_squared_error(y_test,y_pred,squared=False)#rmse

    #print the metrics
    print("The Metrics for {}".format(model))
    print("-----")
    print("Mean Squared Error: {:.2f}".format(mse))
    print("Root Mean Squared Error: {:.2f}".format(rmse))
    print("Mean Absolute Error: {:.2f}".format(mae))
    print("R2 Score: {:.2f}".format(r2_score))
```

The above code is to analyse and get the training and test score of the data of different models . more the training and test score,the more fruitful in building the accurate model.

The screenshot shows a JupyterLab environment with a file browser on the left and a code editor on the right. The file browser shows a directory named 'project5' containing files 'CarPrice.csv', 'first report...', and 'linear-mod...'. The code editor displays the following Python code:

```
b->y intercept

[29]: #initializing the model
linear_model=LinearRegression()

Lets fit the LinearRegression and fetch training and testing scores

[30]: fit_model_getScores(linear_model,
                          X_train_scaled,y_train,
                          X_test_scaled,y_test
                          )

Scores of LinearRegression()

Training Score:0.95
Testing Score:0.89

Get the metrics to evaluate LinearRegression

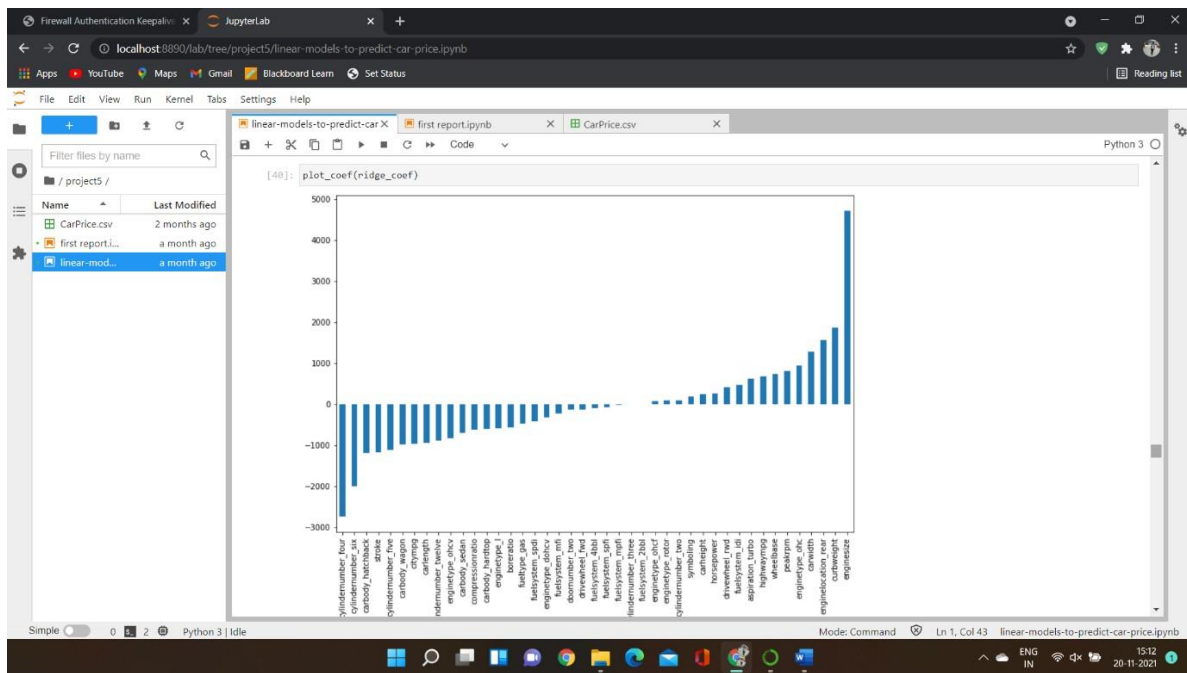
[31]: get_metrics(linear_model,
                  X_test_scaled,y_test
                  )

The Metrics for LinearRegression():
-----
Mean Squared Error:8482008.48
Root Mean Squared Error:2912.39
Mean Absolute Error:2089.38
r2_score:0.89

The function return_coef_series will return the series of coefficient along with features as its index

[32]: linear_coef=feature_coef_series(linear_model.X).coef.values()
```

Linear Regression training and test score with the mean squared, Root mean squared and mean absolute error which is evaluated from the metricses.



Statistical graph generated from Ridge model which is dependent upon the correlation coefficient of the features of dataset.

```
[41]: lasso_model=Lasso()

We got alpha=100 & max_iter=1000 for the lasso model

[42]: params={
    "alpha": [1e-9, 1e-6, 1e-3, 1, 100, 1000, 10000],
    "max_iter": [1e3, 1e4, 1e5, 1e6] #maximum number of iterations to run
}

lasso_best_params=gridSearch(lasso_model, params, X_train_scaled, y_train)
lasso_best_params

[42]: {'alpha': 100, 'max_iter': 1000.0}

Lets try fitting the Lasso model using the parameters that have been returned from Hypertuning

[43]: lasso1_model=Lasso(**lasso_best_params)

The scores are 93% for training set and 86% for testing set which is better generalized model than the above two models (i.e LinearRegression() & Ridge)

[44]: fit_model_getScores(lasso1_model,
    X_train_scaled, y_train,
    X_test_scaled, y_test)

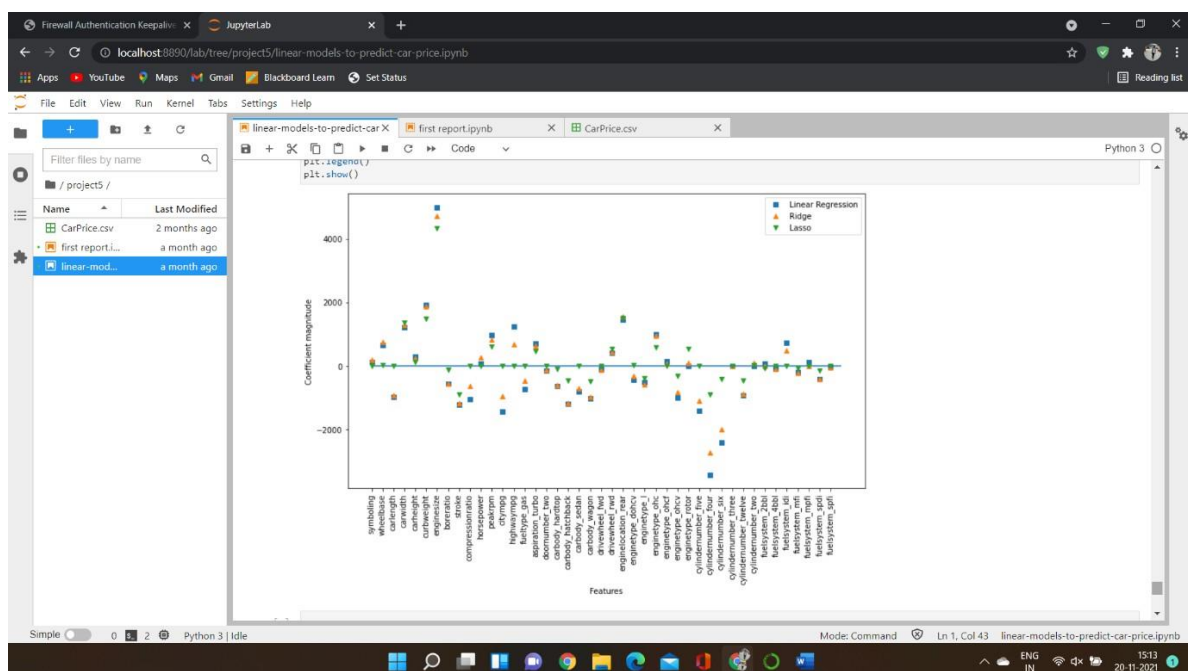
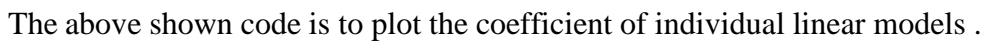
Scores of lasso(alpha=100, max_iter=1000.0)

Training Score:0.93
Testing Score:0.86

Print the metrics of lasso model
```

Lasso training and test score with the mean squared, Root mean squared and mean absolute error which is evaluated from the metrics

Greater the magnitude of linear coefficient, better the correlation of features with the accuracy. In the graphs we can see that which features impacts much more in the model and which features impacts less and the unused features are negligible.



In the above graph, it can be seen there is three shapes with different color on every features. The three shapes are of three different model called linear regression, ridge and Lasso. It shows in which feature, with how much magnitude of coefficient the three models are making the impact and in this way it can be compared that which model is suitable to be deployed and used in predicting the car price.

7. CONCLUSION

The project, linear model to predict the Car price is the project based on data science using python programming and Database. The output of this project is in the form of statistical value as we detect the accuracy of the models being built here. We collected the data set and build the model as it is based on data science. This model being prepared can be further deployed online in any kinds of related website and can be used in a physical life nut unfortunately we are not able to deploy this project as the data set was not enough though it was proper. It requires a huge amount of dataset which dioesnot create any error after being deployed in any websites.

8. BIBLIOGRAPHY

- <https://towardsdatascience.com/predicting-car-price-using-machine-learning-8d2df3898f16>
- <https://www.kaggle.com/ngawangchoeda/linear-models-to-predict-car-price>
- <https://github.com/sahidul-shaikh/car-price-prediction-linear-regression>