B~~a~~hr~~ia~~ Un~~i~~vers~~i~~ty,



K~~a~~r~~a~~chi C~~a~~mpus

LAB EXPERIMENT NO.

\_\_\_\_\_\_\_10\_\_\_\_\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| **1** | Using python implement Linear Regression Algorithm on Car price prediction dataset to predict the price based on these individual features.  ⦁ Kilometers\_Driven  ⦁ Power ⦁ Mileage  Note: Three different machine learning models are required to be created using Linear Regression model on the said features respectively. Write a 250 words paragraph to compare the outcome. |
| **2** | Using python implement Linear Regression Algorithm on Car price prediction dataset using multiple attributes to predict the price. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Submitted On:**

\_\_\_\_20/12/2022\_\_\_\_\_\_

**(Date: DD/MM/YY)**

**Task# 01: - Using python implement Linear Regression Algorithm on Car price prediction dataset to predict the price based on these individual features.**

⦁ **Kilometers\_Driven**

⦁ **Power** ⦁ **Mileage**

**Note: Three different machine learning models are required to be created using Linear Regression model on the said features respectively. Write a 250 words paragraph to compare the outcome.**

**Solution: -**

**General Code: -**

import numpy as np import pandas as pd import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression dataset = pd.read\_csv('/content/drive/MyDrive/train-data.csv')

# Kilometers\_driven model: -

X = dataset["Kilometers\_Driven"].values y = dataset["Price"].values

X = X.reshape(-1,1)

X\_test = pd.read\_csv('/content/test-data.csv')["Kilometers\_Driven"].values.reshape(-1,1) regressor = LinearRegression() regressor.fit(X,y)

y\_pred = regressor.predict(X\_test)

plt.scatter(X, y, color='red') # plotting the observation line

plt.plot(X, regressor.predict(X), color='blue') # plotting the regression line plt.title("Kilometers\_driven vs Price") # stating the title of the graph plt.xlabel("Kilometers\_driven") # adding the name of x-axis plt.ylabel("Price") # adding the name of y-axis plt.show() # specifies end of graph plt.scatter(X\_test, y\_pred, color='red')

plt.plot(X, regressor.predict(X), color='blue') # plotting the regression line plt.title("Kilometers\_driven vs Price (Testing set)") plt.xlabel("Kilometers\_Driven")

plt.ylabel("Price") plt.show()

**Mileage Model:** X = dataset["Mileage"].values y = dataset["Price"].values

X = X.reshape(-1,1)

#removing nan values in the training set y = y[~pd.isnull(X).any(axis=1)] X = X[~pd.isnull(X).any(axis=1)] for x in range(len(X)):

X[x][0] = float(X[x][0].split()[0]) regressor = LinearRegression() regressor.fit(X,y)

X\_test = pd.read\_csv('/content/test-data.csv')["Mileage"].values.reshape(-1,1)

#removing nan values in the test set

X\_test = X\_test[~pd.isnull(X\_test).any(axis=1)] for x in range(len(X\_test)):

X\_test[x][0] = float(X\_test[x][0].split()[0]) y\_pred = regressor.predict(X\_test)

plt.scatter(X, y, color='red') # plotting the observation line plt.plot(X, regressor.predict(X), color='blue') # plotting the regression line plt.title("Mileage vs Price (Training set)") # stating the title of the graph plt.xlabel("Mileage") # adding the name of x-axis plt.ylabel("Price") # adding the name of y-axis plt.show() # specifies end of graph plt.scatter(X\_test, y\_pred, color='red')

plt.plot(X, regressor.predict(X), color='blue') # plotting the regression line plt.title("Mileage vs Price (Testing set)") plt.xlabel("Mileage") plt.ylabel("Price") plt.show()

# Power Model: -

X = dataset["Power"].values

y = dataset["Price"].values

X = X.reshape(-1,1)

#removing nan values in the training set y = y[np.where(X[:,0] != 'null bhp')] X = X[np.where(X[:,0] != 'null bhp')] y = y[~pd.isnull(X).any(axis=1)] X = X[~pd.isnull(X).any(axis=1)] for x in range(len(X)):

X[x][0] = float(X[x][0].split()[0]) regressor = LinearRegression() regressor.fit(X,y)

X\_test = pd.read\_csv('/content/test-data.csv')["Power"].values.reshape(-1,1)

#removing nan values in the test set

X\_test = X\_test[np.where(X\_test[:,0] != 'null bhp')] X\_test = X\_test[~pd.isnull(X\_test).any(axis=1)] for x in range(len(X\_test)):

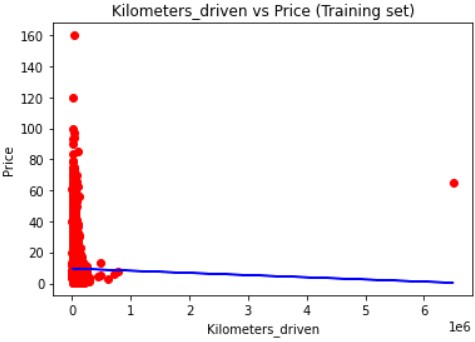
X\_test[x][0] = float(X\_test[x][0].split()[0]) y\_pred = regressor.predict(X\_test)

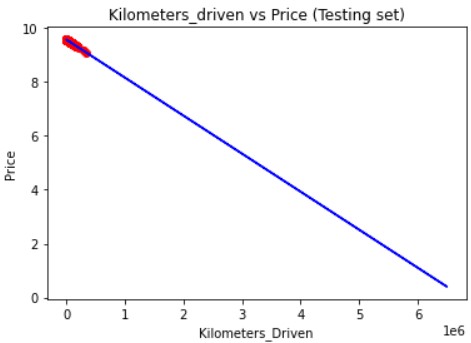
plt.scatter(X, y, color='red') # plotting the observation line plt.plot(X, regressor.predict(X), color='blue') # plotting the regression line plt.title("Power vs Price (Training set)") # stating the title of the graph plt.xlabel("Power") # adding the name of x-axis plt.ylabel("Price") # adding the name of y-axis plt.show() # specifies end of graph plt.scatter(X\_test, y\_pred, color='red')

plt.plot(X, regressor.predict(X), color='blue') # plotting the regression line plt.title("Power vs Price (Testing set)") plt.xlabel("Power") plt.ylabel("Price") plt.show()

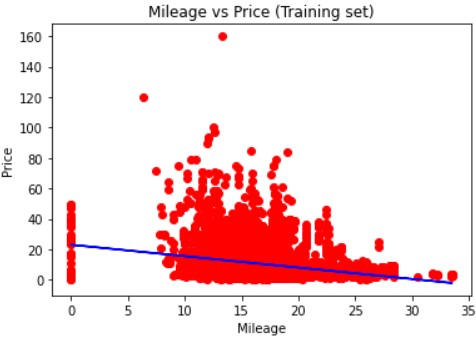
**Output: -**

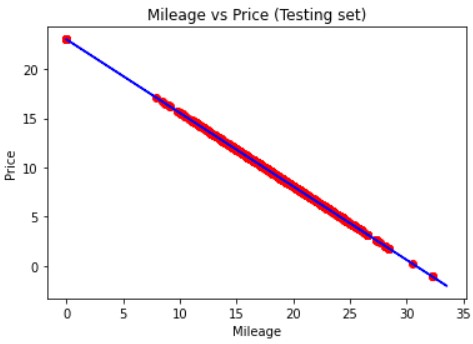
**Kilometers\_driven model: -**



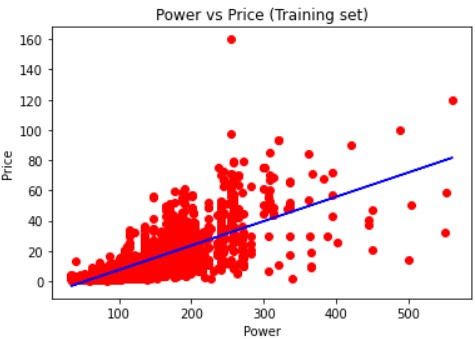


**Mileage Model: -**





**Power Model: -**





**Comparison Between three models: -**

Here, By observing the above graphs, I conclude that the model made with power feature is doing best as compared to the other two models. We can increase the accuracy of these models by the following two ways: -

1. By associating other features of datasets with these models.
2. Removing the outliers from the above data as we can see that some data points are too much far from other points.

**Task 02: - Using python implement Linear Regression Algorithm on Car price prediction dataset using multiple attributes to predict the price.**

# Solution: -

import numpy as np import pandas as pd import matplotlib.pyplot as plt

dataset = pd.read\_csv('/content/drive/MyDrive/train-data.csv') temp = [] for x in range(len(dataset)):

if pd.isnull(dataset["Mileage"][x]):

temp.append(x) else:

dataset["Mileage"][x] = float(dataset["Mileage"][x].split()[0])

dataset.drop(temp,axis=0,inplace=True) temp = [] for x in dataset["Unnamed: 0"]:

if pd.isnull(dataset["Power"][x]) or dataset["Power"][x] == 'null bhp':

temp.append(x) else:

dataset["Power"][x] = float(dataset["Power"][x].split()[0])

dataset.drop(temp,axis=0,inplace=True)

X = dataset[["Kilometers\_Driven","Power","Mileage"]] y = dataset["Price"]

X\_test = pd.read\_csv('/content/test-data.csv')[["Unnamed: 0","Kilometers\_Driven","Power","Mileage"]] temp = []

# X\_test["Unnamed: 0"] for x in X\_test["Unnamed: 0"]:

if pd.isnull(X\_test["Mileage"][x]):

temp.append(x) else:

X\_test["Mileage"][x] = float(X\_test["Mileage"][x].split()[0]) X\_test.drop(temp,axis=0,inplace=True) temp = [] for x in X\_test["Unnamed: 0"]: if pd.isnull(X\_test["Power"][x]) or X\_test["Power"][x] == 'null bhp': temp.append(x) else:

X\_test["Power"][x] = float(X\_test["Power"][x].split()[0]) X\_test.drop(temp,axis=0,inplace=True) from sklearn.linear\_model import LinearRegression regressor = LinearRegression() regressor.fit(X,y)

y\_pred = regressor.predict(X\_test[["Kilometers\_Driven","Power","Mileage"]]) y\_pred

# Output: -

