# Bahria University,

## Karachi Campus

## 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.<br>• Kilometers_Driven<br>• Power<br>• Mileage<br>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: -**

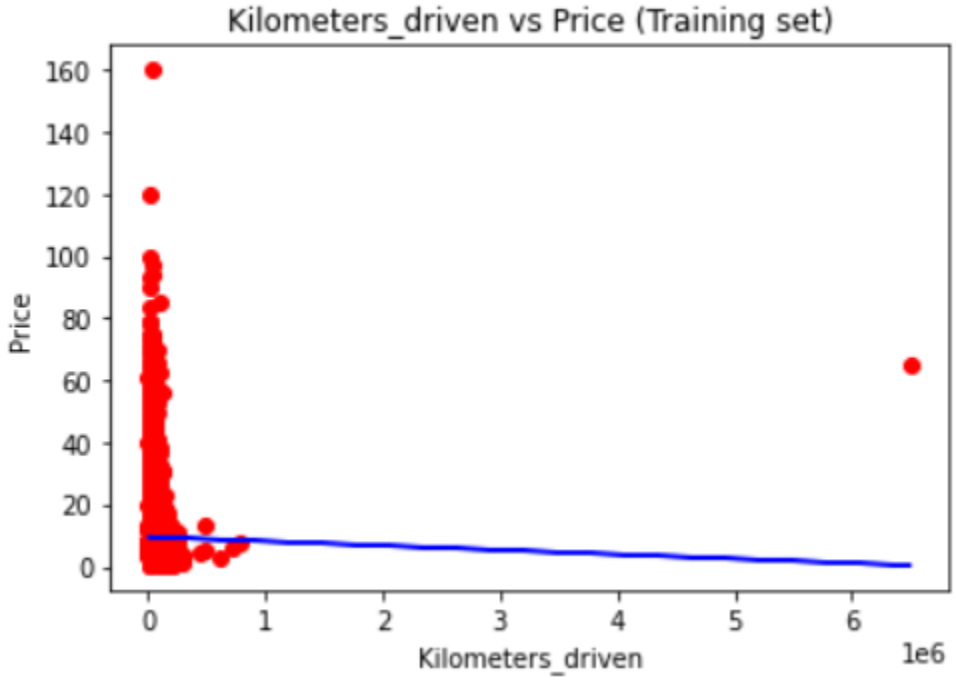
```
X = dataset["Mileage"].values
y = dataset["Price"].values
X = X.reshape(-1,1)
#removing nan values in the training set
y = y[~pd.isnull(X).any(axis=1)]
X = X[~pd.isnull(X).any(axis=1)]
for x in range(len(X)):
  X[x][0] = float(X[x][0].split()[0])
regressor = LinearRegression()
regressor.fit(X,y)
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()
```

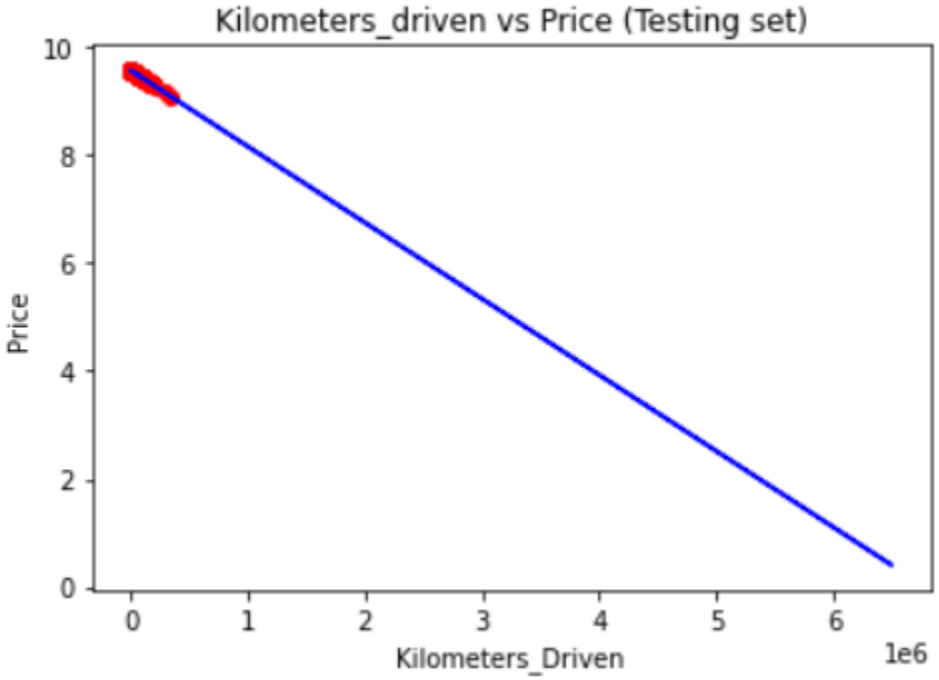
**Power Model: -**

```
X = dataset["Power"].values
```

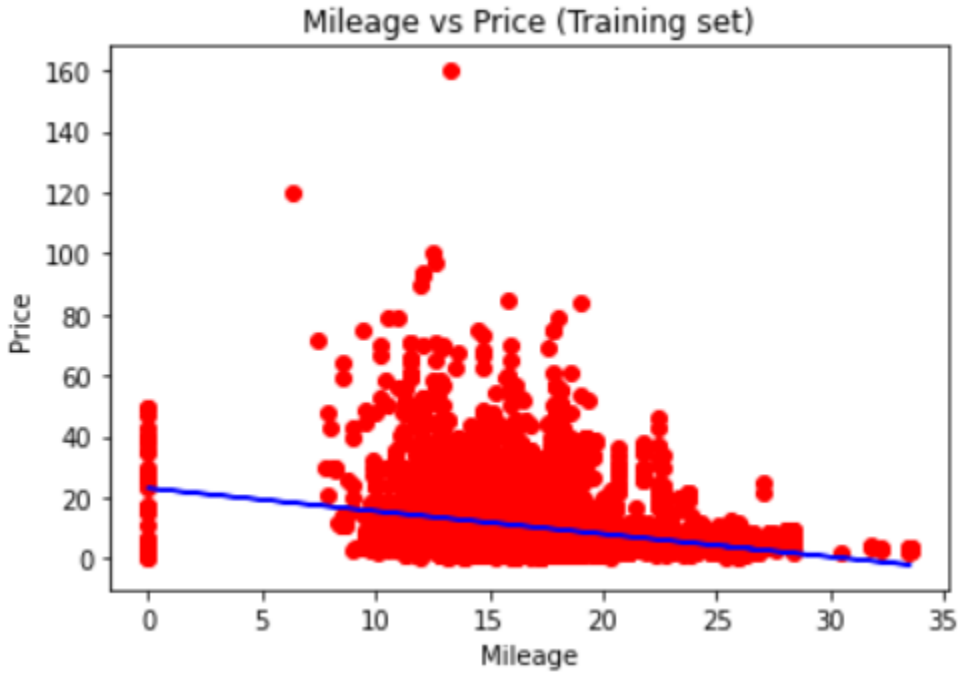
```
y = dataset["Price"].values
X = X.reshape(-1,1)
#removing nan values in the training set
y = y[np.where(X[:,0] != 'null bhp')]
X = X[np.where(X[:,0] != 'null bhp')]
y = y[~pd.isnull(X).any(axis=1)]
X = X[~pd.isnull(X).any(axis=1)]
for x in range(len(X)):
  X[x][0] = float(X[x][0].split()[0])
regressor = 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()
```

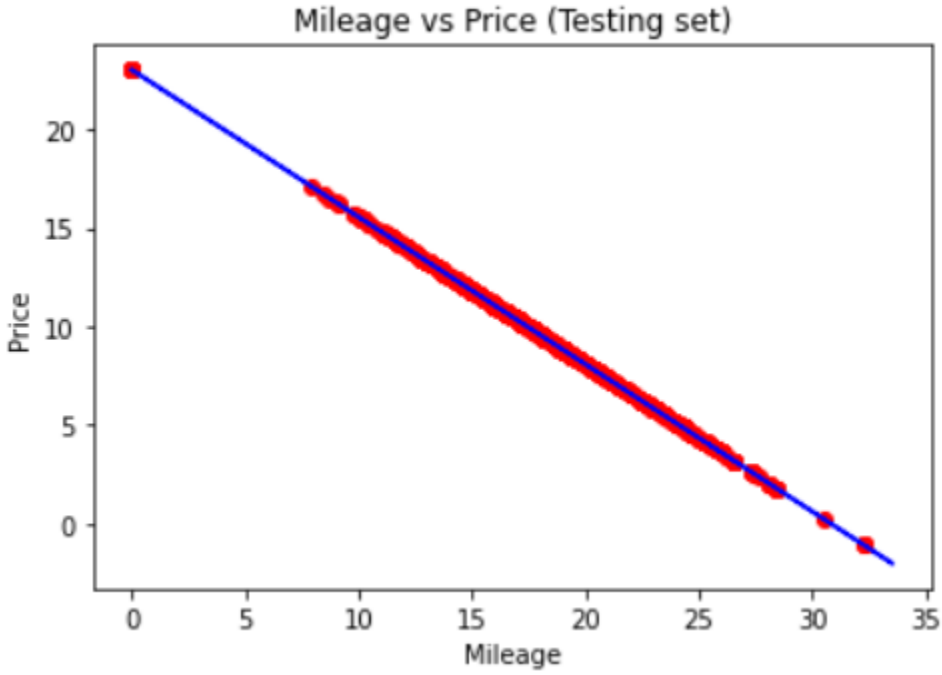
**Output: -**
**Kilometers_driven model: -**



**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: -**

```
array([ 3.85868512, -0.09673436, 14.44452091, ...,  2.30394664,
        7.66153098, 17.16794189])
```