Name: OM M PATEL
Roll No.: 21BCP094
Group: G3
Division: D2

LAB ASSIGNMENT-3

Title: Implement simple and multi-linear regression to predict profits for a food truck. Compare the performance of the model on linear and multi-linear regression

Objective:

This lab assignment aims to implement simple and multi-linear regression models to predict profits for a food truck business. By comparing the performance of these two regression models, you will gain insights into when and how to use simple and multi-linear regression techniques.

Dataset Format:

Population	Years in business	Profit
10,000	5	10000
15000	6	12000
20000	6	13000
9000	5	12000
12000	4	?

Tasks:

1. Apply Simple Linear Regression Selecting.

```
import pandas as pd
import numpy as np


data = {
"population" : [10000,15000,20000,9000,12000], "experience" : [5,6,6,5,4], "profit"
[10000,12000,13000,12000,np.nan],
# "profit" :[10000,12000,13000,12000,13000], }

df = pd.DataFrame(data)
df1 = df.drop(df.index[-1])
# df1 =df
```

```
import numpy as np
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

X = df1['population'].values.reshape(-1,1)
y = df1['profit'].values
model = lr.fit(X,y)
predict_lr = model.predict(X)
predict_lr
```

Output:
array([11136.36363636, 12012.98701299, 12889.61038961, 10961.03896104])

2. Performance Evaluation (Simple Linear Regression).

```
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y,predict_lr)
r2 = r2_score(y,predict_lr)
rmse = np.sqrt(mse)
print(f"Evaluation: \n Mean Square Error: {mse} \n Root Mean Square Error: {rmse} \n R2_score: {r2} ")
```

```
Evaluation:
 Mean Square Error: 595779.2207792206
 Root Mean Square Error: 771.8673595762556
 R2_score: 0.498291182501709
```

3. Multi-Linear Regression.

```
X_m = df1[['population', 'experience']].values
y_m = df1['profit'].values
mlr = LinearRegression()
mlr.fit(X_m,y_m)
predict_mlr = mlr.predict(X_m)
predict_mlr
```

#Output
array([11057.69230769, 12211.53846154, 12788.46153846, 10942.30769231])

4. Performance Evaluation (Multi-Linear Regression).

```
mlr_mse = mean_squared_error(y_m,predict_mlr)
mlr_rmse = np.sqrt(mlr_mse)
mlr_r2_Score = r2_score(y_m,predict_mlr)
print(f"Evaluation for Multi-Linear Regression: \n Mean Square Error: {mlr_mse} \n Root Mean Square Error:
{mlr_rmse} \n R2_score: {mlr_r2_Score} \n\n ")
```

```
Evaluation for Multi-Linear Regression:
 Mean Square Error: 581730.7692307702
 Root Mean Square Error: 762.7127698096907
 R2_score: 0.5101214574898778
```

5. Model Comparison and Interpretation

print(f"Evaluation for Multi-Linear Regression: \n Mean Square Error: {mlr_mse} \n Root Mean Square Error: {mlr_rmse} \n R2_score: {mlr_r2_Score} \n\n “)

print(f"Evaluation for Linear Regression: \n Mean Square Error: {mse} \n Root Mean Square Error: {rmse} \n R2_score: {r2} ")

```
Evaluation for Multi-Linear Regression:
 Mean Square Error: 581730.7692307702
 Root Mean Square Error: 762.7127698096907
 R2_score: 0.5101214574898778


Evaluation for Linear Regression:
 Mean Square Error: 595779.2207792206
 Root Mean Square Error: 771.8673595762556
 R2_score: 0.498291182501709
```

From the above codes and evaluation measures, we can interpret that multi-linear regression will provide us with better prediction than simple linear regression, we will validate our interpretation by predicting the missing value of the last index in our original dataset.

Simple Linear Regression	Multi- Linear Regression
#Missing Value Prediction arr = np.array([12000]).reshape(-1, 1) predict_lastrow_lr = model.predict(arr) predict_lastrow_lr	**#Missing Value Prediction** arr_mlr = np.array([[12000, 4]]) predict_lastrow_mlr = mlr.predict(arr_mlr) predict_lastrow_mlr
#Output: array([11487.01298701])	**#Output:** array([10711.53846154])