Name: Jigar Siddhpura **SAPID:** 60004200155
DIV: C/C2 **Branch:** Computer Engineering

DMW EXP 4

Jigar Siddhpura
60004210155

DMW - Experiment 4

Aim : To implement Linear regression
1. Single variate 2. Multi variate.

Theory :
1. Linear regression is a machine learning based on supervised learning.
2. Regression models a target prediction value based on independent variables.
3. It is mostly used for finding out the relationship btw variables & forecasting.
4. Different regression model differs based on - the kind of relationship btw dependent independent variables they are considering.
5. It performs task to predict a dependent variable (y) based on independent variable (x). So it finds out relationship btw x & y.
6. Regression line is the Best fit line for our model.
7. Hypothesis function for linear regression :
$$y_E = \theta_1 + \theta_2 \cdot x$$
8. While training, x: input training data (univariate)
y: labels to data.
9. While training it fits the best line to predict value of y for a given x. The model gets the best regression fit line by finding best θ_1 & θ_2.
θ_1 : intercept
θ_2 : co-efficient of x

Cost function (J) : 1. By achieving the best-fit regression line, the model predicts (y) value such that error difference predicted value & true value is minimum.

2. It is very imp. to update θ_1 & θ_2 values, to reach the best value that minimize the error btw predicted $\hat{y}$ & & true y.

→ $$\text{minimize} \left[\frac{1}{n} \sum_{i=1}^{n} (\text{pred}_i - y_i)^2 \right]$$

→ $$J = \frac{1}{n} \sum_{i=1}^{n} (\text{pred}_i - y_i)^2$$

Cost function (J) of linear regression is the Root Mean Squared Error (RMSE) btw $\hat{y}$ & y.

CODE:

```
from google.colab import drive
drive.mount("/content/gdrive")

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error

df =
pd.read_csv("/content/gdrive/MyDrive/DMW/datasets/StudentsPerformance.csv")
df.head()

df['final_score'] = df.apply(lambda x: (x["math score'] + x['reading score']
+ x["writing score'])/3, axis=1)

df2 = pd.get_dummies(df, columns=["gender","lunch","parental level of
education','race/ethnicity','test preparation course'])
df2 = df2.drop(["math score','reading score","writing score'],axis=1)

# multi-variate
y = df2['final_score']
X = df2.drop(['final_score'],axis=1)
xtrain, xtest, ytrain, ytest =
train_test_split(X,y,test_size=0.25,random_state=10)
sns.boxplot(data=df2['final_score'],orient='h')
```

```
model = linearRegression()
model.fit(xtrain,ytrain)
score = model.score(xtest,ytest)
print(score)
ypred = model.predict(xtest)

sns.scatterplot(data=df,x=df['reading  score'],y=df['final_score'])

#regression line
m,b = np.polyfit(x=df['reading score'],y=df['final_score'],deg=1)
X = df['reading score']
plt.plot(X, m*X+b)

# univariate
X_uni = df['reading score']
y_uni = df['final_score']
x_uni_train, x_uni_test, y_uni_train, y_uni_test =
train_test_split(X_uni,y_uni,test_size=0.25,random_state=10)
x_uni_train = x_uni_train.values.reshape(-1,1)
x_uni_test = x_uni_test.values.reshape(-1,1)
uni_model = linearRegression()
uni_model.fit(x_uni_train,y_uni_train)
uni_score = uni_model.score(x_uni_test,y_uni_test)
print(uni_score)
y_uni_pred = uni_model.predict(x_uni_test)
```

OUTPUT:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

head() of the database

	gender	race/ethnicity	parental level of education	lunch	test preparation course	final_score
0	female	group B	bachelor's degree	standard	none	72.666667
1	female	group C	some college	standard	completed	82.333333
2	female	group B	master's degree	standard	none	92.666667
3	male	group A	associate's degree	free/reduced	none	49.333333
4	male	group C	some college	standard	none	76.333333

df.head() after adding a final score column

	final_score	gender_female	gender_male	lunch_free/reduced	lunch_standard	parental level of education_associate's degree	parental level of education_bachelor's degree	parental level of education_high school	parental level of education_master's degree	parental level of education_some college	parental educatic high
0	72.666667	1	0	0	1	0	1	0	0	0	
1	82.333333	1	0	0	1	0	0	0	0	1	
2	92.666667	1	0	0	1	0	0	0	1	0	
3	49.333333	0	1	1	0	1	0	0	0	0	
4	76.333333	0	1	0	1	0	0	0	0	1	

df.head() after applying One-hot encoding to the dataset

Considering Multivariate Linear Regression

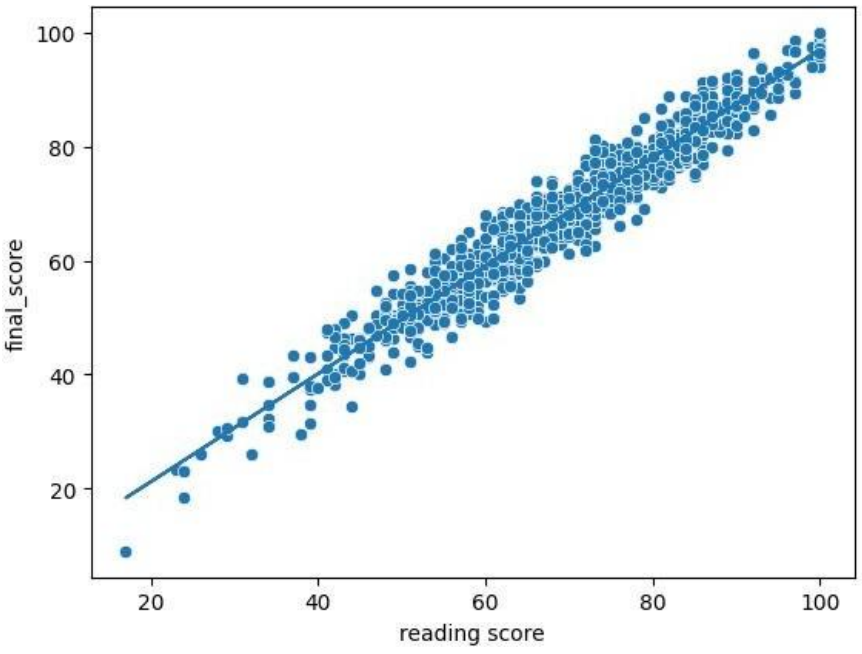
Prediction Score

```
score = model.score(xtest,ytest)
print(score)
ypred = model.predict(xtest)
```

```
0.19674412629893356
```

Now considering Univariate Linear Regression with Reading Score as the feature

Scatter plot



Prediction Score of Univariate LR

```
uni_score = uni_model.score(x_uni_test,y_uni_test)
print(uni_score)
y_uni_pred = uni_model.predict(x_uni_test)
```

```
0.944431815987387
```

CONCLUSION:

We have implemented Multivariate and Univariate Linear
Regression on a dataset and have observed the differences in their AccuracyScore and
Mean Squared Errors. We observe 19.67% accuracy in the case of
Multivariate whereas in the case ofUnivariate, the accuracy score is
94.43%

94.21% and the Mean Squared Error is 109.48. Therefore we can conclude that using
Multivariate Linear Regression is better than using Univariate but nevertheless the efciency of Univariate is still great.