Name: Marwin Shroff
Div: B/B1
SAP:60004200097
Branch: Computer Engineering

DWM EXPERIMENT 4

LINEAR REGRESSION

AIM: Implementation of Linear Regression

1. Single Variate

2. Multi Variate

THEORY:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for fnding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering and the number of independent variables being used.
Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique fnds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. In the fgure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best ft line for our model.

Hypothesis function for Linear Regression :

$$y = \theta_1 + \theta_2.x$$

While training the model we are given :

x: input training data (univariate – one input variable(parameter))

y: labels to data (supervised learning)

When training the model – it fts the best line to predict the value of y for a given value of x. The model gets the best regression ft line by fnding the best $\theta 1$ and $\theta 2$ values.
$\theta 1$: intercept

$\theta 2$: coefficient of x

Once we find the best $\theta 1$ and $\theta 2$ values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.
Cost Function (J):

By achieving the best-ft regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the $\theta 1$ and $\theta 2$ values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

CODE:

```
import warnings

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import
mean_squared_error from
sklearn.linear_model import
LinearRegression from sklearn.preprocessing
import LabelEncoder from sklearn.metrics
import accuracy_score import
matplotlib.pyplot as plb
from sklearn.model_selection import train_test_split

sns.set()

warnings.simpleflter("ignore")

df = pd.read_csv("StudentsPerformance.csv")

df.head()

print(df.info())

df['final score'] = df.apply(lambda x : (x['math score'] + x['reading score'] +
x['writing score'])

/ 3, axis=1)

df.head()

data2 = df.drop('fnal score', axis=1)

plt.fgure(fgsize=(16, 6))

sns.boxplot(data=data2)

df = df.apply(LabelEncoder().ft_transform)

# MULTIVARIATE

X = df.drop('fnal score', axis=1)
```

```
y = df['fnal score']

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.2)

lr = LinearRegression()

lr.ft(X_train, y_train)

pred = lr.predict(X_test)

lr.score(X_test, y_test)

accuracy = mean_squared_error(y_test, pred)

print('Mean Squared Error: ', accuracy)

# UNIVARIATE

sns.scatterplot(df["writing score"],df["fnal score"])

plt.savefg('scp-1', dpi=500)

m, b = np.polyft(df["writing score"], df["fnal score"], 1)

plt.plot(df["writing score"], m*df["writing score"] + b)

X_uni = df['writing score']

y_uni = df['fnal score']

X_uni_train, X_uni_test, y_uni_train, y_uni_test =
train_test_split(X_uni,y_uni,test_size = 0.2)

lr2 = LinearRegression()

X_uni_train = X_uni_train.reshape(-1,1)

X_uni_test = X_uni_test.values.reshape(-1,1)

lr2.ft(X_uni_train, y_uni_train)

pred_uni = lr2.predict(X_uni_test)

lr2.score(X_uni_test, y_uni_test)

accuracy_uni = mean_squared_error(y_uni_test, pred_uni)

print('Mean Squared Error: ', accuracy_uni)
```

OUTPUT:

head() of the database:

	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

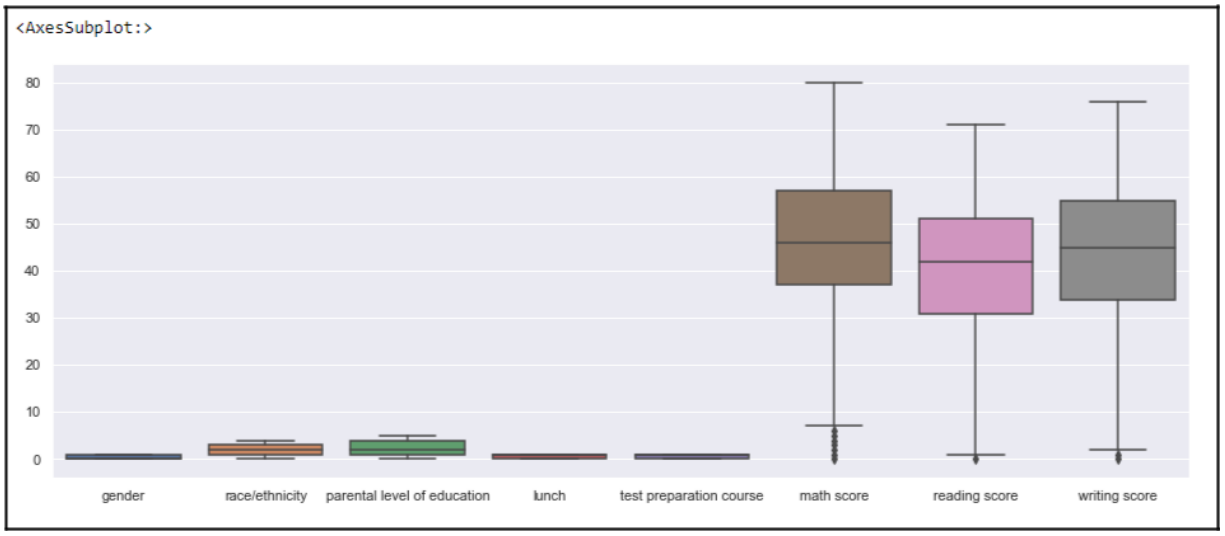
After running df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64 
 6   reading score                1000 non-null   int64 
 7   writing score                1000 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
None
```

df.head() after adding a final score column

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	final score
0	female	group B	bachelor's degree	standard	none	72	72	74	72.666667
1	female	group C	some college	standard	completed	69	90	88	82.333333
2	female	group B	master's degree	standard	none	90	95	93	92.666667
3	male	group A	associate's degree	free/reduced	none	47	57	44	49.333333
4	male	group C	some college	standard	none	76	78	75	76.333333

Boxplot of the features



df.head() after applying LabelEncoder to the dataset

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	final score
0	0	1	1	1	1	52	44	50	118
1	0	2	4	1	0	49	62	64	147
2	0	1	3	1	1	70	67	69	178
3	1	0	0	0	1	27	29	20	48
4	1	2	4	1	1	56	50	51	129

df.info() after applying LabelEncoder to the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   int64
 1   race/ethnicity               1000 non-null   int64
 2   parental level of education  1000 non-null   int64
 3   lunch                        1000 non-null   int64
 4   test preparation course      1000 non-null   int64
 5   math score                   1000 non-null   int64
 6   reading score                1000 non-null   int64
 7   writing score                1000 non-null   int64
 8   final score                  1000 non-null   int64
dtypes: int64(9)
memory usage: 70.4 KB
None
```

Considering Multivariate Linear Regression

Prediction Score of MultiVariate Linear Regression

```
lr.score(X_test, y_test)
```

```
0.9992194766540022
```

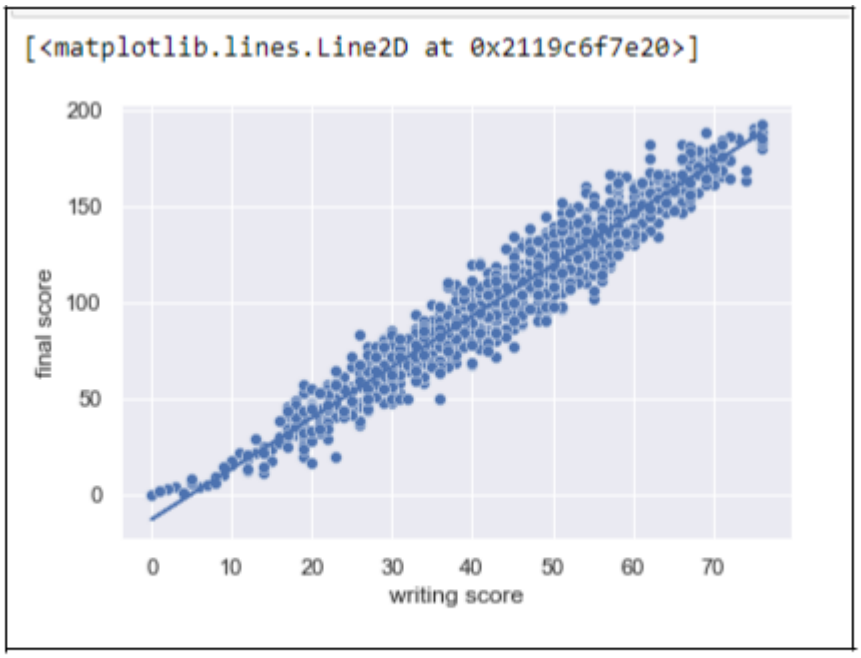
Mean Square Error of MultiVariate Linear Regression

```
print('Mean Squared Error: ', accuracy)
```

```
Mean Squared Error:  1.692171227952071
```

Now considering Univariate Linear Regression with Writing Score as the feature

Scatter Plot of the dataset



Prediction Score of Univariate LR

```
lr2.score(X_uni_test, y_uni_test)
```

```
0.9421228773316737
```

Mean Square Error of Univariate LR

```
accuracy_uni = mean_squared_error(y_uni_test, pred_uni)
print('Mean Squared Error: ', accuracy_uni)
```

```
Mean Squared Error:  109.48409107917793
```

CONCLUSION: We have implemented Multivariate and Univariate Linear Regression on a dataset and have observed the differences in their Accuracy Score and Mean Squared Errors. We observe 99.92% accuracy in the case of Multivariate with a Mean Squared Error of 1.62 whereas in the case of Univariate, the accuracy score is

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.
