NAME: Meith Navlakha
SAP: 60004190068
DWM Practical

EXPERIMENT 3

AIM: Implementation of Linear Regression

1. Single Variate
2. Multi variate

THEORY:

Linear Regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables, but it has been borrowed by machine learning. It is both a statistical algorithm and a machine learning algorithm.

In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.

The representation is a linear equation that combines a specific set of input values (X) the solution to which is the predicted output for that set of input values (y). As such, both the input values (X) and the output value are numeric. The linear equation assigns one scale factor to each input value or column, called a **coefficient,** and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the **bias coefficient**.

For example, in a simple regression problem (for X and y), the form of the model would be:

$$y = B_0 + B_1 * x \quad \ldots \text{ univariate}$$

$$y = B_0 + B_1 * x_1 + B_2 * x_2 + \ldots\ldots + B_n * x_n \quad \ldots. \text{ multivariate}$$

NAME: Meith Navlakha
SAP: 60004190068
DWM Practical

PART A: Univariate Salary Dataset

Salary Dataset is a univariate dataset having 2 columns 'YearsExperience' and Salary. I have trained a Linear Regression model that predicts the Salary based on the Years of Experience of the Employee.

CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import LinearRegression
from google.colab import files
from sklearn.metrics import mean_squared_error, r2_score
from google.colab import files
import io
uploaded = files.upload()
# Load the diabetes dataset
salary_data = pd.read_csv(io.BytesIO(uploaded['Salary_Data.csv']))
salary_data.head()

sal_X = salary_data[['YearsExperience']]
sal_y = salary_data.Salary

# Test Train Split
from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(sal_X , sal_y, test_size = 0.3 ,
random_state= 2 )

# Create linear regression object
sal_reg = LinearRegression()

# Train the model using the training sets
sal_reg.fit(X_train, y_train)

#  PREDICT UNSEEN
y_pred = sal_reg.predict(X_test)
print('PREDICT UNSEEN SAMPLES : \n' , y_pred)
# Coefficients
print('\nCoefficient (C0): ', sal_reg.coef_)
```

```
# MSE
print('Mean squared error(MSE) : %.2f' % mean_squared_error(y_test, y_pred))
# R2 Score
print('R2 SCORE : %.2f' % r2_score(y_test, y_pred))

# Plot outputs
plt.scatter(X_test, y_test,  color='green')
plt.plot(X_test, y_pred, color='blue', linewidth=3)
plt.title("Salary vs Experience")
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.xticks(())
plt.yticks(())

plt.show()
```

OUTPUT:

DataSet

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0

```
PREDICT UNSEEN SAMPLES :
 [ 36143.62176044  34237.05465324  66648.69547576  59022.42704693
  91434.06786946  80947.94877982 101920.1869591   52349.44217171
  42816.60663567]

Coefficient (C0):  [9532.83553604]
Mean squared error(MSE) : 64406629.39
R2 SCORE : 0.90
```

PLOT Linear Regression Model

PART B: Multivariate Dataset

Melbourne Housing Prices

This is a multivariate dataset that depends on attributes like Area, Number of Rooms, Real Estate Agent, Distance, Council Area, Type of House and other features. I have implemented a Linear Regression Model along with data pre-processing before it.

CODE:

```
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
```

```
dataset = pd.read_csv('./Melbourne_housing_FULL.csv')
dataset.head()

# DATA CLEANING
cols_to_use = ['Suburb' , 'Rooms' , 'Type' , 'Method' , 'SellerG' , 'Regionname' ,
'Propertycount' , 'Distance' , 'CouncilArea' , 'Bedroom2' ,'Bathroom' , 'Car' , 'Landsize'
, 'BuildingArea' , 'Price' ]

dataset = dataset[cols_to_use]

# N.A values in the dataset
dataset.isna().sum()

# Fill with Zero
cols_to_fill_zero = ['Propertycount' , 'Distance' , 'Bedroom2' , 'Bathroom' , 'Car']
dataset[cols_to_fill_zero] = dataset[cols_to_fill_zero].fillna(0)

# Fill with mean
dataset['Landsize'] = dataset['Landsize'].fillna(dataset.Landsize.mean())
dataset['BuildingArea'] = dataset['BuildingArea'].fillna(dataset.BuildingArea.mean())

# Drop the remaining
dataset.dropna(inplace=True)

dataset = pd.get_dummies(dataset , drop_first = True)

# X and Y data columns
X = dataset.drop('Price' , axis=1)
Y = dataset['Price']

train_X , test_X , train_Y , test_Y = train_test_split(X , Y, test_size = 0.3 ,
random_state= 2 )

# LINEAR REGRESSION
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

lr = LinearRegression().fit(train_X ,train_Y)
lr_r2 = lr.score(test_X , test_Y)
```

```
#  PREDICT UNSEEN
y_pred = lr.predict(test_X)
print('PREDICT UNSEEN SAMPLES(First 20) : \n' , y_pred[:20])
# Coefficients
# print('\nCoefficient (C0): ', lr.coef_)
# MSE
print('\nMean squared error(MSE) : %.2f' % mean_squared_error(test_Y, y_pred))
# R2 Score
print('R2 SCORE : %.2f' % lr_r2)

# Graph Actual Values VS Predicted Values
f =plt.figure(figsize= (6,6))
ax = plt.axes()
ax.plot(test_Y , y_pred , marker='o' , ls='' , ms=3.0 , label= 'Linear')
leg = plt.legend(frameon =True)
leg.get_frame().set_edgecolor('black')
leg.get_frame().set_linewidth(1.0)
lim = (0 , test_Y.max())
ax.set(xlabel = 'Actual Price', ylabel='Predicted Value' , xlim =lim , ylim =lim , title=
'Housing Price Predictions Model')
```

OUTPUT:

Load Melbourne Housing Dataset

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Post
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	2.5	3
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016	2.5	3
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3
...	...	...	...	...	...	...	...	...	...	
34852	Yarraville	13 Burns St	4	h	1480000.0	PI	Jas	24/02/2018	6.3	3
34853	Yarraville	29A Murray St	2	h	888000.0	SP	Sweeney	24/02/2018	6.3	3
34854	Yarraville	147A Severn St	2	t	705000.0	S	Jas	24/02/2018	6.3	3

NAME: Meith Navlakha
SAP: 60004190068
DWM Practical

DATA Preprocessing and Cleaning (isNA , isNull)

Before

```
Suburb             0
Rooms              0
Type               0
Method             0
SellerG            0
Regionname         3
Propertycount      3
Distance           1
CouncilArea        3
Bedroom2        8217
Bathroom        8226
Car             8728
Landsize       11810
BuildingArea   21115
Price           7610
dtype: int64
```

After

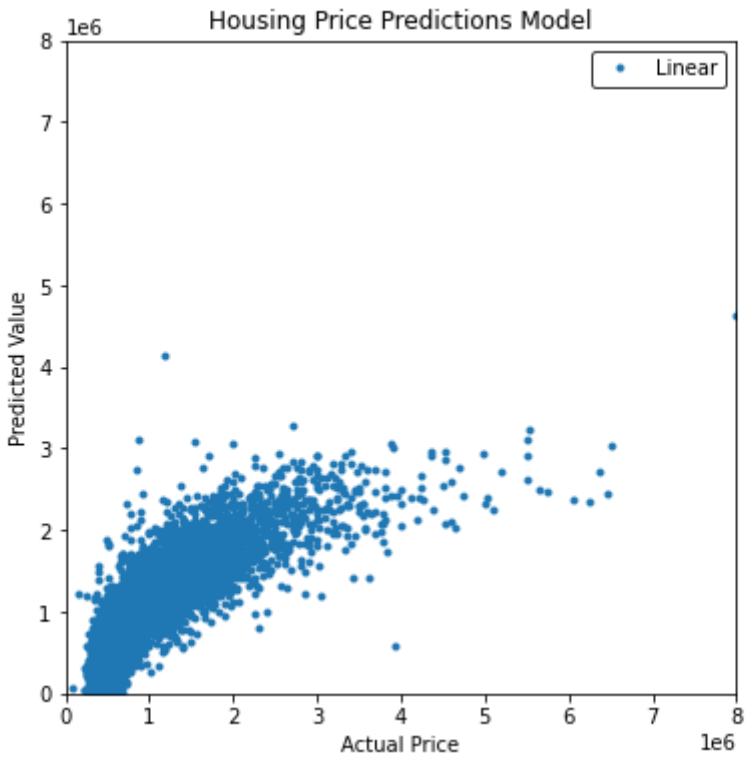
```
Suburb          0
Rooms           0
Type            0
Method          0
SellerG         0
Regionname      0
Propertycount   0
Distance        0
CouncilArea     0
Bedroom2        0
Bathroom        0
Car             0
Landsize        0
BuildingArea    0
Price           0
dtype: int64
```

```
PREDICT UNSEEN SAMPLES(First 20) :
 [1672988.74421108 1086049.72134445  852828.79001551  271308.33769275
  626189.36204137  891181.38519527  893662.90064913  686588.40443425
  904609.05338973 2137514.73886276 1456932.26794679 1243302.3151183
 2290006.13130356 2969081.89910789 2640893.31429078 1027576.89019241
 1295875.65815463 1477978.34232549 1444372.58421122 1044079.41092246]

Mean squared error(MSE) : 342275472104.02
R2 SCORE : 0.74
```

NAME: Meith Navlakha
SAP: 60004190068
DWM Practical

PLOT Actual vs Predicted Prices



CONCLUSION: In this experiment I implemented Linear Regression on Univariate as well as Multivariate dataset. I have predicted the salary for unseen samples and also plotted the Linear Regression for the Salary Dataset. In Salary Dataset the linear model had an accuracy of 90%. In the Melbourne Dataset, I performed data pre-processing by filtering out the superfluous columns which does not affect the pricing. Then I performed data cleaning by removing the null values and filling the N/A values by the mean. I have also predicted the house prices for unseen samples and displayed the first 20 values. Finally, the Melbourne House Prediction linear model had an accuracy of 74%.