CLASS: B.E. E &TC **SUBJECT:** ML
EXPT. NO.: 9 **DATE:** 01/02/21
ROLL NO.: 42428

TITLE: **To create a machine learning algorithm of Linear Regression**

CO 1: Understand the basic concepts in machine learning like parametric/non- parametric modeling, classification, clustering, linear/ nonlinear regression and supervised/unsupervised learning to broadly classify various types of machine learning algorithms. Given an applicable feature vector, select an appropriate machine learning approach to design an analytical model to make expected predictions.

CO 4: Carry out experiments as an individual and in a team, comprehend and write a laboratory record and draw conclusions at a technical level.

AIM:
To implement:
To create a Machine Learning Algorithm of linear regression for classification

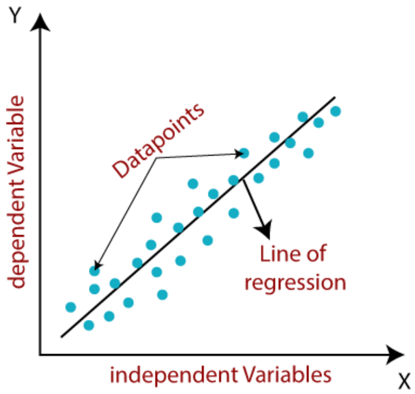
SOFTWARES REQUIRED: MATLAB 7.0 or Python

THEORY:

It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:



Mathematically, we can represent a linear regression as:

$y= a0+a1x+ \varepsilon$

Here,

Y= Dependent Variable (Target Variable)
X= Independent Variable (Predictor Variable)
a0=intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value). ε = random error
The values for x and y variables are training datasets for Linear Regression model representation.

Types of Linear Regression
Linear regression can be further divided into two types of the algorithm:

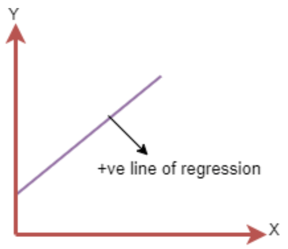
- **Simple Linear Regression:**
  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- **Multiple Linear Regression:**

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Linear Regression Line

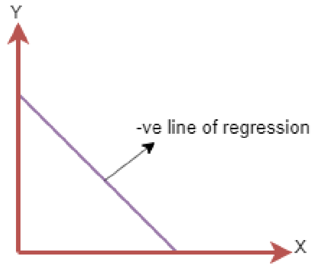
A linear line showing the relationship between the dependent and independent variables is called a regression line. A regression line can show two types of relationship:

- **Positive Linear Relationship:**
  If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



The line equation will be: $\mathbf{Y= a_0+a_1x}$

- **Negative Linear Relationship:**
  If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



The line of equation will be: $\mathbf{Y= -a_0+a_1x}$

Simple Linear regression algorithm has mainly two objectives:

- **Model the relationship between the two variables.** Such as the relationship between Income and expenditure, experience and Salary, etc.
- **Forecasting new observations.** Such as Weather forecasting according to temperature, Revenue of a company according to the investments in a year, etc.

Terminologies Related to the Regression Analysis:

- **Dependent Variable:** The main factor in Regression analysis which we want to predict or understand is called the dependent variable. It is also called target variable.
- **Independent Variable:** The factors which affect the dependent variables or which are used to predict the values of the dependent variables are called independent variable, also called as a predictor.
- **Outliers:** Outlier is an observation which contains either very low value or very high value in comparison to other observed values. An outlier may hamper the result, so it should be avoided.
- **Multicollinearity:** If the independent variables are highly correlated with each other than other variables, then such condition is called Multicollinearity. It should not be present in the dataset, because it creates problem while ranking the most affecting variable.
- **Underfitting and Overfitting:** If our algorithm works well with the training dataset but not well with test dataset, then such problem is called Overfitting. And if our algorithm does not perform well even with training dataset, then such problem is called underfitting.

Strengths and Weakness of Linear Regression approach

The strengths of decision tree methods are:

- Regression estimates the relationship between the target and the independent variable.

- It is used to find the trends in data.
- It helps to predict real/continuous values.
- By performing the regression, we can confidently determine the **most important factor, the least important factor, and how each factor is affecting the other factors.**

The weaknesses of Linear Regression methods:
- Linear regression most of the time can be only used when we deal with relationships that graphically look like a line.
- Linear Regression only looks at the mean of the dependent variable.

ALGORITHM:
1. Calculate sum of all dependent variables(∑y), SOP slope and independent variables (∑mixi), sum of constants (∑c), SOP dependent and independent variables (∑yixi), SOP slope and dependent variable square(∑mixi2), SOP constant and independent variables(∑cxi)

2. We will have 2 equations with m and c unknowns.

3. Solve the equations linearly for m and c.

4. We will get a linear equation in y=mx+c format. Value for any dependent variable can be found by substituting the value for independent variable.

5. Error can thus be found by MSE.

CONCLUSION:

Regression models a target prediction value based on independent variables. It is mostly used for finding 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.

REFERENCES:

i. Laurene Fausett ,"Fundamentals of Neural Networks: Architectures, Algorithms and Applications", Pearson Education, Inc, 2008.

ii. S. N. Sivanandam , S. Sumathi, S. N. Deepa, "Introduction to Neural Networks using MATLAB", McGraw Hill, 2006.

iii. S. N. Sivanandam, S. N. Deepa, "Principles of Soft Computing" , John Wiley & Sons, 2007

iv. Phil Kim, "MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence", a Press 2017.

(Course Teacher)

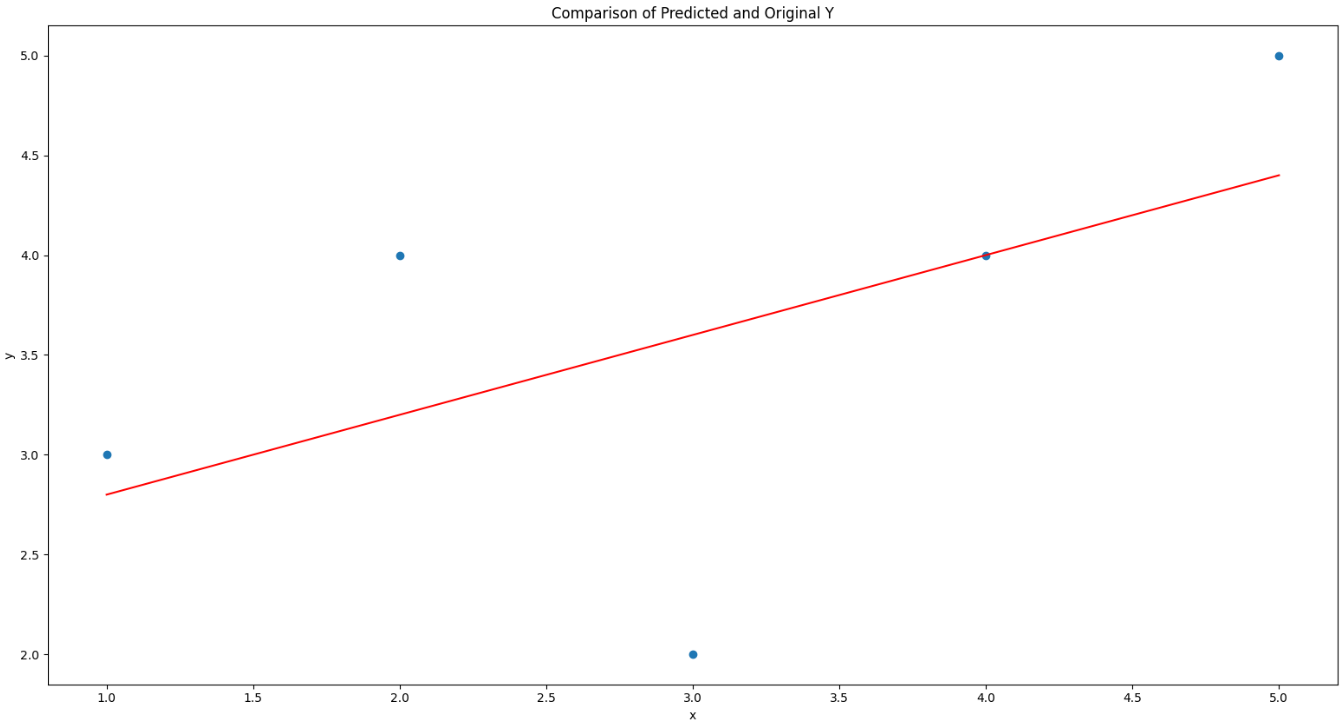
Code :

```
import numpy as np
from matplotlib import pyplot as plt

x = np.array([1,2,3,4,5])
y = np.array([3,4,2,4,5])

mean_x = np.mean(x)
mean_y = np.mean(y)

numerator = 0
denominator = 0

for i in range(x.size):
    numerator += (x[i] - mean_x)*(y[i] - mean_y)
    denominator += (x[i] - mean_x)**2

m = numerator/denominator

c = mean_y - (m * mean_x)

print("\nsolpe = ",m)
print("\nintercept = ",c)
print("\nLine fitted : y =",m,"x +",c)

y_predicted = np.zeros(y.size)

for i in range(x.size):
    y_predicted[i] = (m * x[i]) + c

print("\nx = ",x)
print("\ny = ",y)
print("\ny_predicted = ",y_predicted)

numerator = 0
denominator = 0

for i in range(x.size):
    numerator += (y_predicted[i] - y[i])**2
    denominator += (y[i] - mean_y)**2
```

```
R_sq_coeff = numerator/denominator

print("\nR_square Coefficient = ",R_sq_coeff)

plt.scatter(x,y)
plt.plot(x,y_predicted,'r')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Comparison of Predicted and Original Y')
plt.show()
```

Output :

```
solpe =  0.4

intercept =  2.4

Line fitted : y = 0.4 x + 2.4

x =  [1 2 3 4 5]

y =  [3 4 2 4 5]

y_predicted =  [2.8 3.2 3.6 4.  4.4]

R_square Coefficient =  0.6923076923076922
PS C:\Users\chand>
```


Comparison of Predicted and Original Y
5.0
4.5
4.0
3.5
3.0
2.5
2.0
y
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x