Experiment :

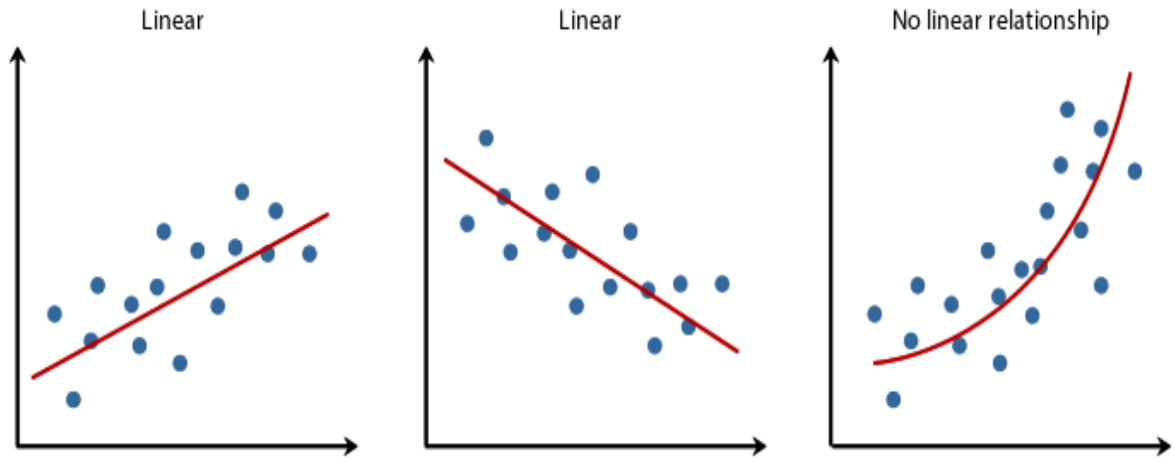
AIM : Write a program to implement Linear Regression.

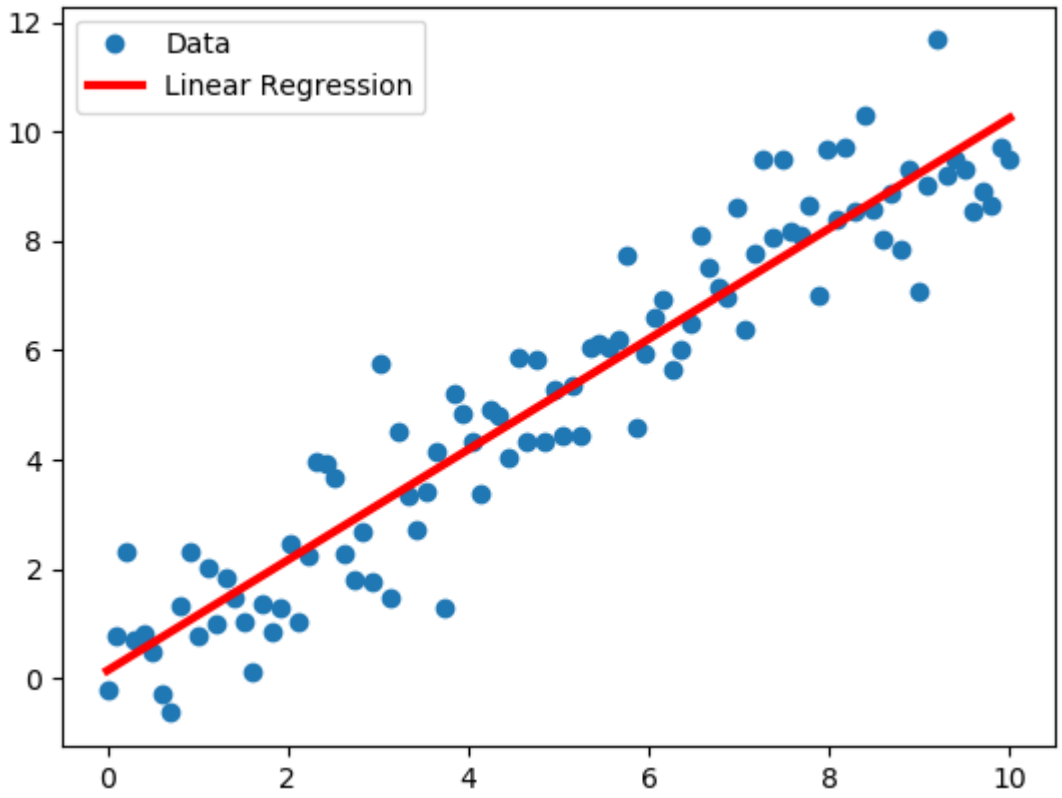
THEORY :

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.





Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

- If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.

- If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

Linear regression models are often fitted using the least squares approach, but they may also be fitted in other ways, such as by minimizing the "lack of fit" in some other norm (as with least absolute deviations regression), or by minimizing a penalized version of the least squares cost function as in ridge regression (L^2-norm penalty) and lasso (L^1-norm penalty). Conversely, the least squares approach can be used to fit models that are not linear models. Thus, although the terms "least squares" and "linear model" are closely linked, they are not synonymous.

Given a data set of n statistical units, a linear regression model assumes that the relationship between the dependent variable y and the p-vector of regressors x is linear. This relationship is modeled through a disturbance term or error variable ε — an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors. Thus, the model takes the form

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where $^{\mathsf{T}}$ denotes the transpose, so that $x_i^{\mathsf{T}}\beta$ is the inner product between vectors x_i and β.

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$X = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Some remarks on notation and terminology:

- **y** is a vector of observed values **yi** of the variable called the regress and, endogenous variable, response variable, measured variable, criterion variable, or dependent variable.
- **X** may be seen as a matrix of row-vectors **xi** or of n-dimensional column-vectors **Xj** , which are known as regressors, exogenous variables, explanatory variables, covariates, input variables, predictor variables, or independent variables (not to be confused with the concept of independent random variables). The matrix **X** is sometimes called the design matrix.
- **beta** is a (p+1)-dimensional parameter vector, where **beta0** is the intercept term. Its elements are known as effects or regression coefficients (although the latter term is sometimes reserved for the estimated effects.
- **epsilen** is a vector of values which is a part of the model is called the error term, disturbance term, or sometimes noise (in contrast with the "signal" provided by the rest of the model).

ALGORITHM :

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Where θ_0 and θ_1 are intercept and slope of regression line(called parameters of regression) of assumed hypothesis $h_\theta(x)$ which in case of simple linear regression is a straight line,

And $J(\theta_0, \theta_1)$ is the mean squared cost function which is to be minimized by using gradient descent algorithm.

Gradient descent algorithm

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$
$$(\text{for } j = 1 \text{ and } j = 0)$$
$$\}$$

Both the parameters are updated simultaneously.

CODE :

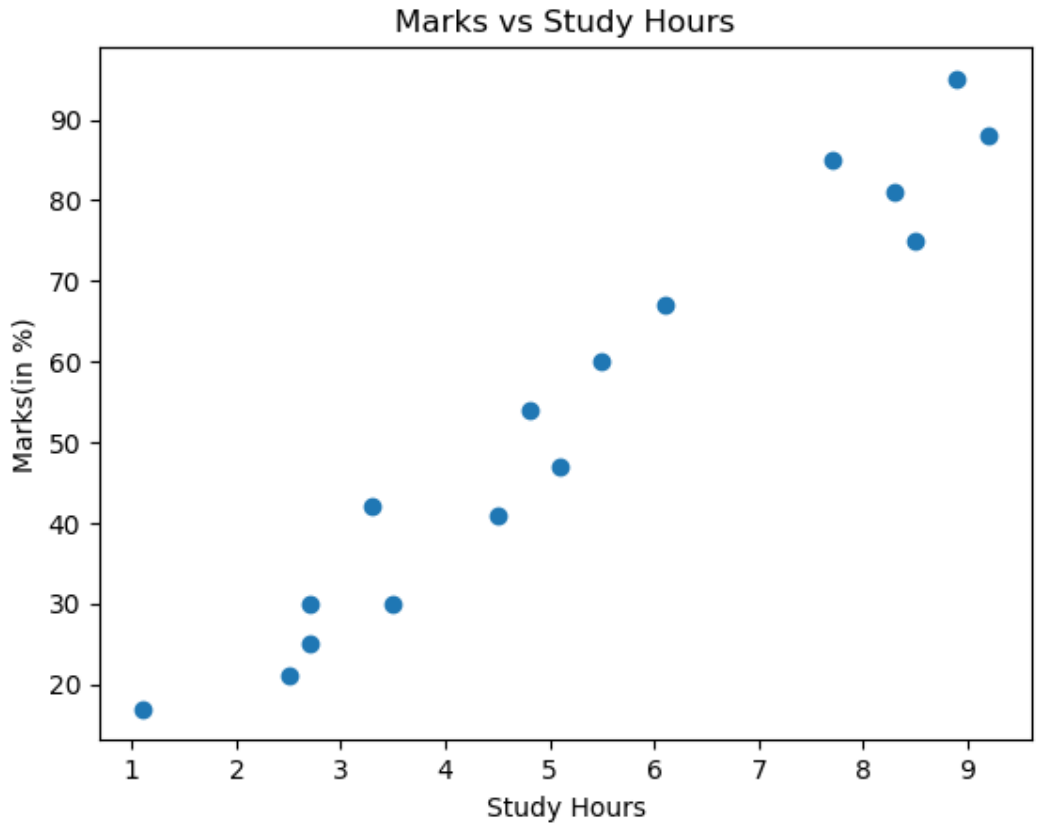
```
#Importing the libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('student_scores.csv')
x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values

#x = (x-x.mean())/x.std()
plt.title('Marks vs Study Hours')
plt.xlabel('Study Hours')
plt.ylabel('Marks(in %)')
plt.scatter(x,y)
plt.show()

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
x, xtest, y, ytest = train_test_split(x, y, test_size = 1/3, random_state = 0)

def hypothesis(x,theta):
    return theta[1]*x + theta[0]
    #return np.dot(theta,X)
def gradient(x,y,theta):

    grad = np.array([0,0])
    m = x.shape[0]

    for i in range(m):
        hx = hypothesis(x[i],theta)
        grad[1] += (hx-y[i])*x[i]
        grad[0] += (hx-y[i])

    return grad/m

def error(x,y,theta):

    e = 0
    for i in range(x.shape[0]):
        e += (y[i] - hypothesis(x[i],theta))**2

    #print(type(e))
    return 0.5*e/x.shape[0]

def gradientDescent(x,y,lr=.01, threshold=0.001):
    theta = np.array([0.1,0.2])
    error_list = []
```

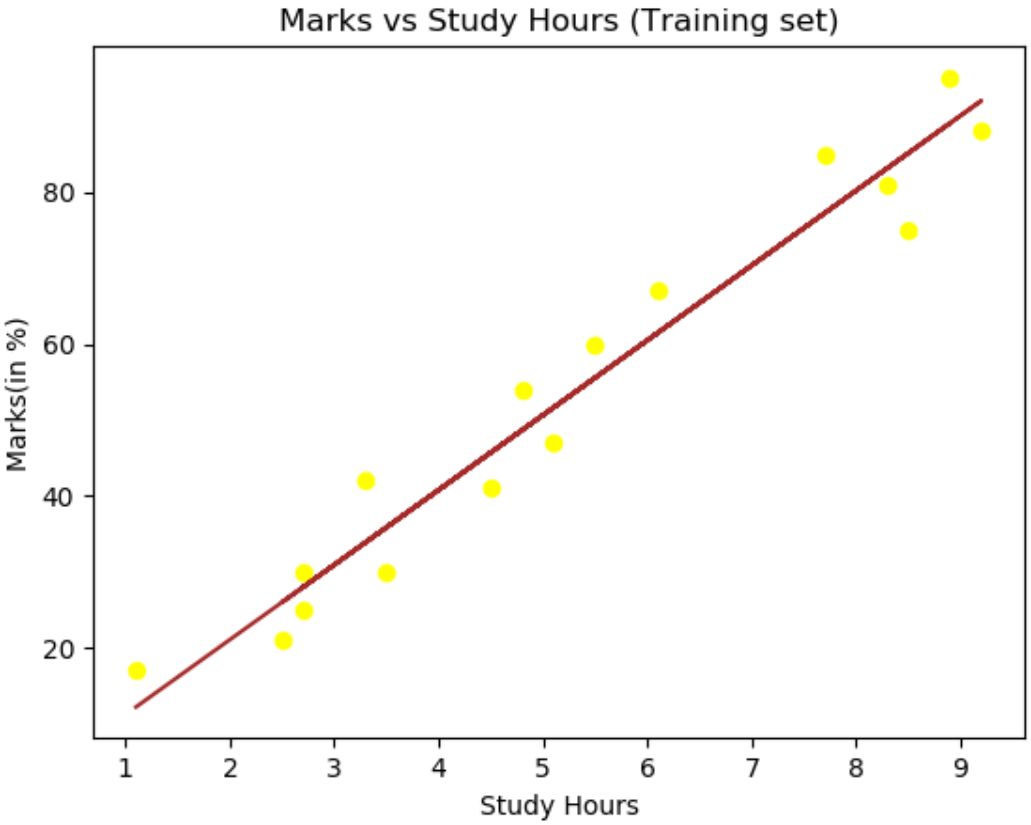
```
    # Homework - define converge criteria to break the loop
    # Change in error < threshold
    for i in range(800):
        grad = gradient(x,y,theta)
        e = error(x,y,theta)
        error_list.append(e)
        theta[0] = theta[0] - lr*grad[0]
        theta[1] = theta[1] - lr*grad[1]

    return theta,error_list

#Printing theta values
print(theta)

#Running gradient descent
theta,err = gradientDescent(x,y)

# Visualising the Training set results
plt.scatter(x, y, color = 'yellow')
plt.plot(x, hypothesis(x,theta), color = 'brown')
plt.title('Marks vs Study Hours (Training set)')
plt.xlabel('Study Hours')
plt.ylabel('Marks(in %)')
plt.show()

# Visualising the Test set results
plt.scatter(x, y, color = 'yellow')
plt.plot(x, hypothesis(x,theta), color = 'brown')
plt.title('Marks vs Study Hours (Test set)')
plt.xlabel('Study Hours')
plt.ylabel('Marks(in %)')
plt.show()
```

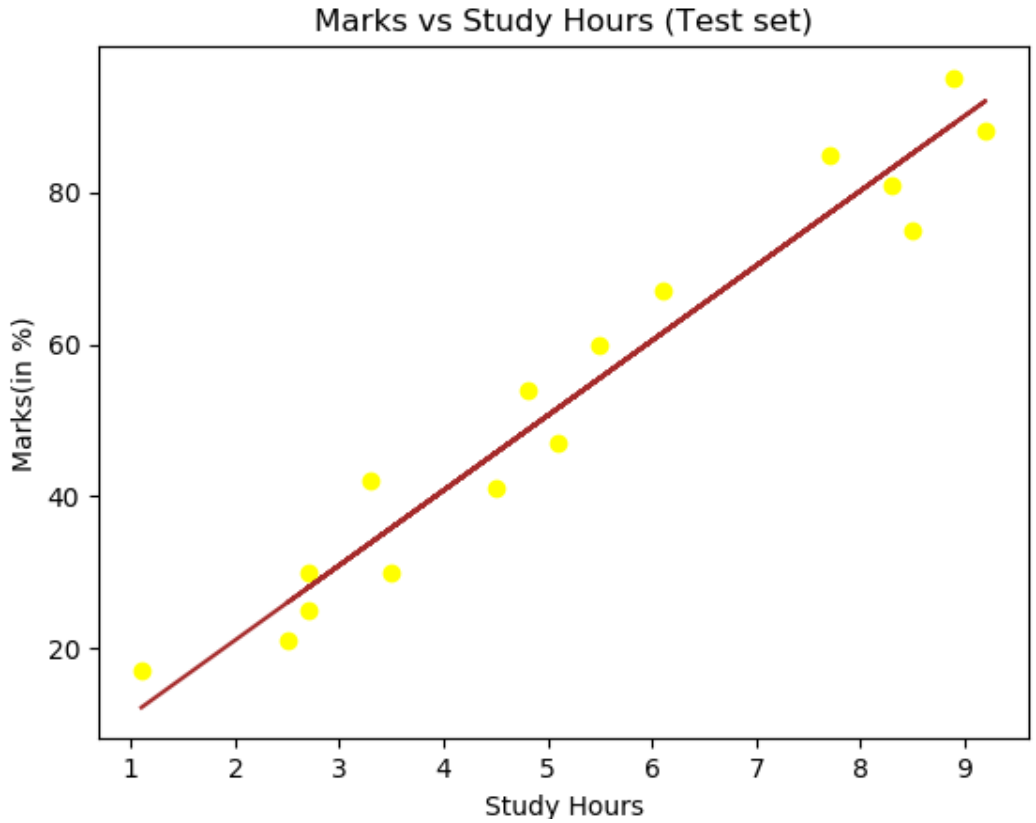
OUTPUT :

Hours	Scores
2.5	21
5.1	47
3.2	27
8.5	75
3.5	30
1.5	20
9.2	88

Theta values：[1.399375 9.851875]







Above graphs show that there seems to be a linear relationship between Marks and Study Hours.

DISCUSSION :

Linear regression is a common statistical data analysis technique. It is used to determine the extent to which there is a linear relationship between a dependent variable and one or more independent variables. There are two types of linear regression, simple linear regression and multiple linear regression.
In simple linear regression a single independent variable is used to predict the value of a dependent variable. In multiple linear regression two or more independent variables are used to predict the value of a dependent variable. The difference between the two is the number of independent variables. In both cases there is only a single dependent variable.

FINDING AND LEARNING :

In this experiment, by using a dataset we learnt about implementing and understanding the concept of linear regression for continuous variable type. Here we determined a relationship between a dependent variable ‘Marks’ with an independent variable ‘Study Hours’.