GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

MACHINE LEARNING

COURSE CODE: BCSE3093

LAB ETE

Submitted by- Sushant Kumar Bhartiya

Adms. No. - 18SCSE1010019

Semester :- 5

LAB ETE

AIM:-

Simple Linear Regression

Objective:-

A program to demonstrate the working of the Simple Linear Regression. Use an appropriate data set the implementation.

Concept:- Simple Linear Regression is a type of Regression algorithms that models the relationship between a dependent variable and a single independent variable. The relationship shown by a Simple Linear Regression model is linear or a sloped straight line, hence it is called Simple Linear Regression.

The key point in Simple Linear Regression is that the dependent variable must be a continuous/real value. However, the independent variable can be measured on continuous or categorical values.

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.

The Simple Linear Regression model can be represented using the below equation:

$y= a_0+a_1x+ \varepsilon$

Where,

a0= It is the intercept of the Regression line (can be obtained putting x=0)
a1= It is the slope of the regression line, which tells whether the line is increasing or decreasing.
ε = The error term. (For a good model it will be negligible.

Algorithm:-

Problem Statement example for Simple Linear Regression:

Here we are taking a dataset that has two variables: dependent variable and Independent variable. The goals of this problem is:

- **We want to find out if there is any correlation between these two variables**
- **We will find the best fit line for the dataset.**
- **How the dependent variable is changing by changing the dependent variable.**

In this section, we will create a Simple Linear Regression model to find out the best fitting line for representing the relationship between these two variables.

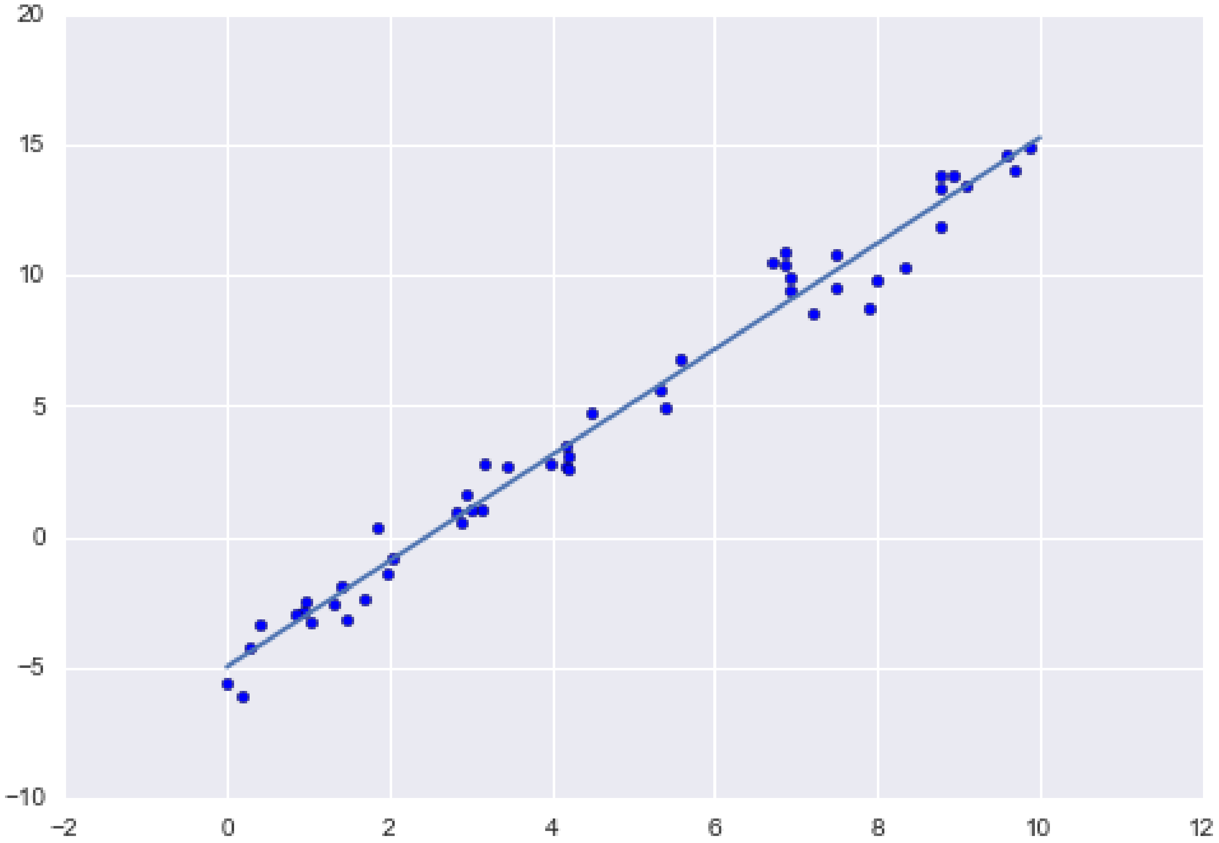
To implement the Simple Linear regression model in machine learning using Python, we need to follow the below steps:

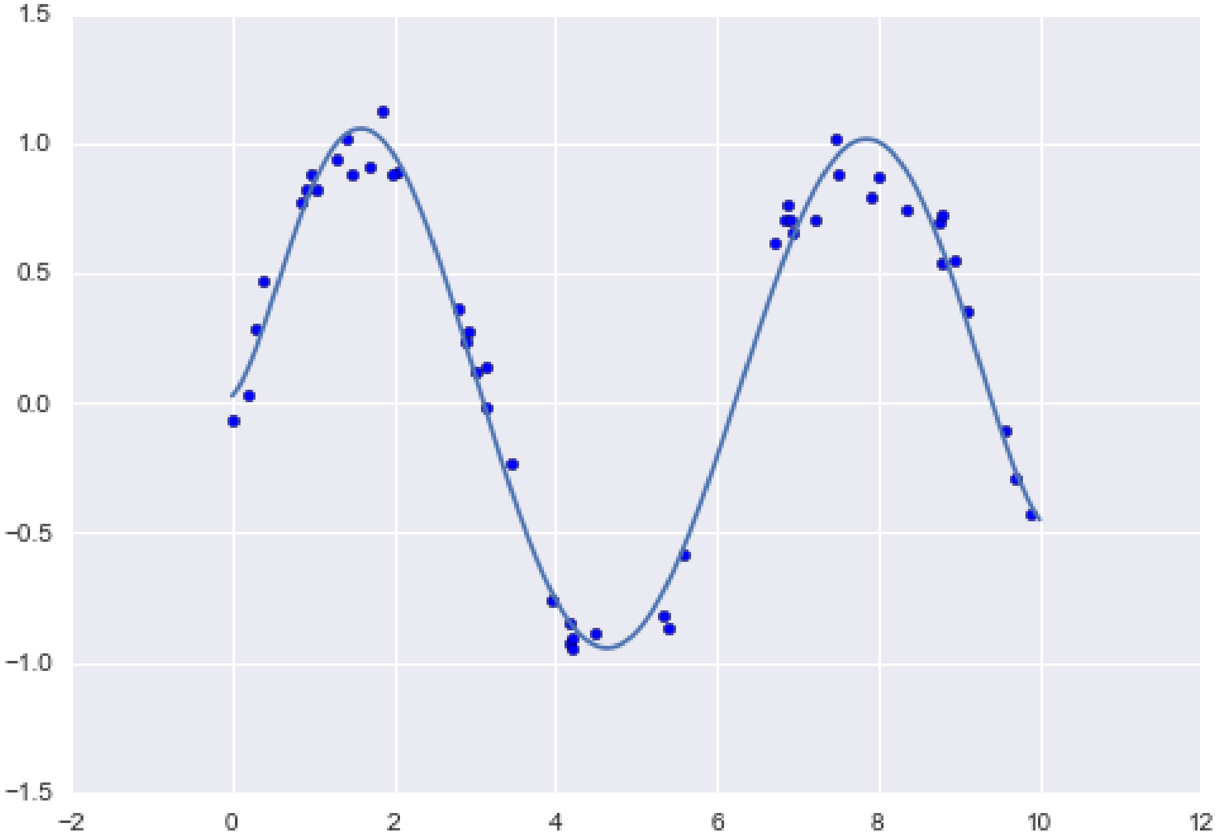
Coding / Logic:-

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import numpy as np

rng = np.random.RandomState(1)
x = 10 * rng.rand(50)
y = 2 * x - 5 + rng.randn(50)
plt.scatter(x, y);
from sklearn.linear_model import LinearRegression
model = LinearRegression(fit_intercept=True)
```

```
model.fit(x[:, np.newaxis], y)

xfit = np.linspace(0, 10, 1000)
yfit = model.predict(xfit[:, np.newaxis])

plt.scatter(x, y)
plt.plot(xfit, yfit);
print("Model slope:    ", model.coef_[0])
print("Model intercept:", model.intercept_)
rng = np.random.RandomState(1)
X = 10 * rng.rand(100, 3)
y = 0.5 + np.dot(X, [1.5, -2., 1.])

model.fit(X, y)
print(model.intercept_)
print(model.coef_)
from sklearn.preprocessing import PolynomialFeatures
x = np.array([2, 3, 4])
poly = PolynomialFeatures(3, include_bias=False)
poly.fit_transform(x[:, None])
from sklearn.pipeline import make_pipeline
poly_model = make_pipeline(PolynomialFeatures(7),
                           LinearRegression())
rng = np.random.RandomState(1)
x = 10 * rng.rand(50)
y = np.sin(x) + 0.1 * rng.randn(50)

poly_model.fit(x[:, np.newaxis], y)
yfit = poly_model.predict(xfit[:, np.newaxis])

plt.scatter(x, y)
plt.plot(xfit, yfit);
```

Output (Screenshots):-


20
15
10
5
0
−5
−10
−2
0
2
4
6
8
10
12



Result:-

The program is successfully run…