G H Raisoni College of Engineering
SY AI Semester-IV AY 2023-24 Division-A
UCAIP210: Machine Learning Algorithms Practicals
Lab Manual
Practical Teacher : Dr Monika Y. Dangore

Experiment No: 5

Aim:

Implement Linear Regression using Python

Introduction:

A **linear regression** is a linear approximation of a causal relationship between two or more variables. **Regression analysis** is one of the most widely used methods for prediction. It is applied whenever we have a causal relationship between variables.

For Ex : "The amount of money you spend depends on the amount of money you earn."

The Process of Creating a Linear Regression

1. Get sample data;
2. Design a model that explains the data;
3. Use the developed model to make a prediction for the whole population.

There is a dependent variable, labeled *Y*, being predicted, and independent variables, labeled *x1*, *x2*, and so forth. These are the predictors. *Y* is a function of the *X* variables, and the **regression model** is a linear approximation of this function.

The Simple Linear Regression

The easiest regression model is the simple linear regression**:**

$Y = \beta_0 + \beta_1 * x_1 + \varepsilon.$

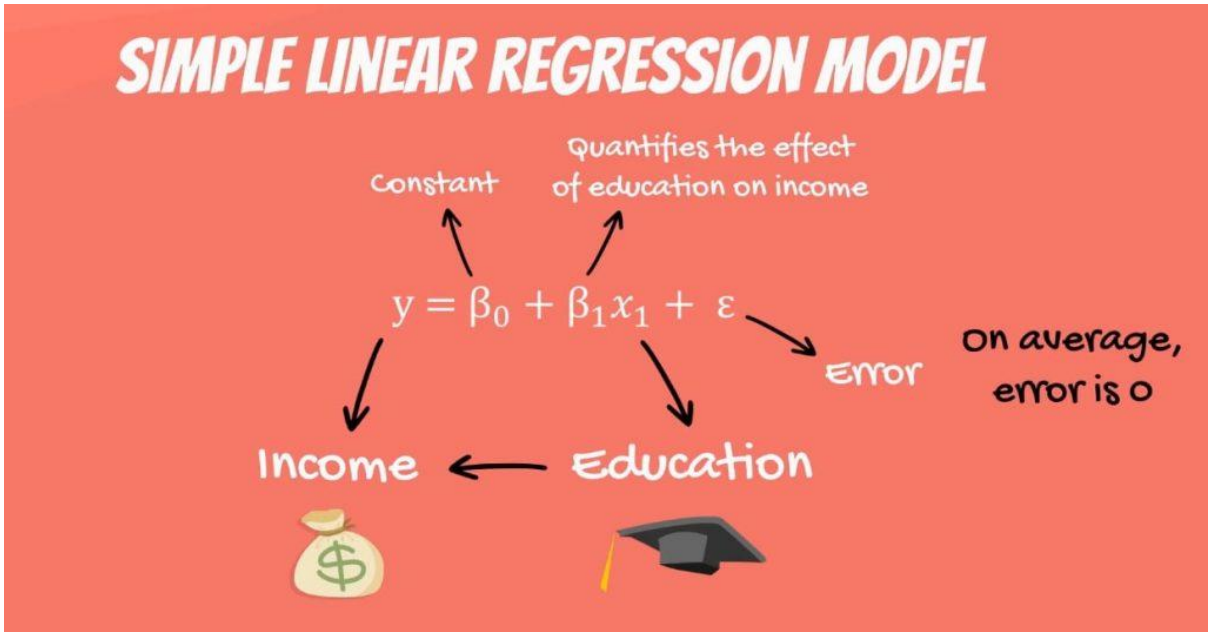
Y is the variable we are trying to predict and is called the *dependent variable*. x1 *i*s an *independent variable*.

When using **regression analysis**, we want to predict the value of *Y*, provided we have the value of *X*.

But to have a **regression**, *Y* must depend on *X* in some way. Whenever there is a change in *X*, such change must translate to a change in *Y*.

Providing a Linear Regression Example

The income a person receives depends on the number of years of education that person has received. The *dependent variable* is income, while the *independent variable* is years of education.



In our model, there are coefficients. β_1 is the coefficient that stands before the independent variable. It quantifies the effect of education on income.
If β_1 is 50, then for each additional year of education, your income would grow by $50. The other two components are the constant β_0 and the error – epsilon(ε).

In this **linear regression** example, you can think of the constant β_0 as the minimum wage. No matter your education, if you have a job, you will get the minimum wage. This is a guaranteed amount.

So, if you never went to school and plug an education value of 0 years in the formula, logically, the **regression** will predict that your income will be the minimum wage.

The last term is the epsilon(ε). This represents the error of estimation. The error is the actual difference between the observed income and the income the **regression** predicted. On average, across all observations, the error is 0. If you earn more than what the **regression** has predicted, then someone earns less than what the **regression** predicted. Everything evens out.

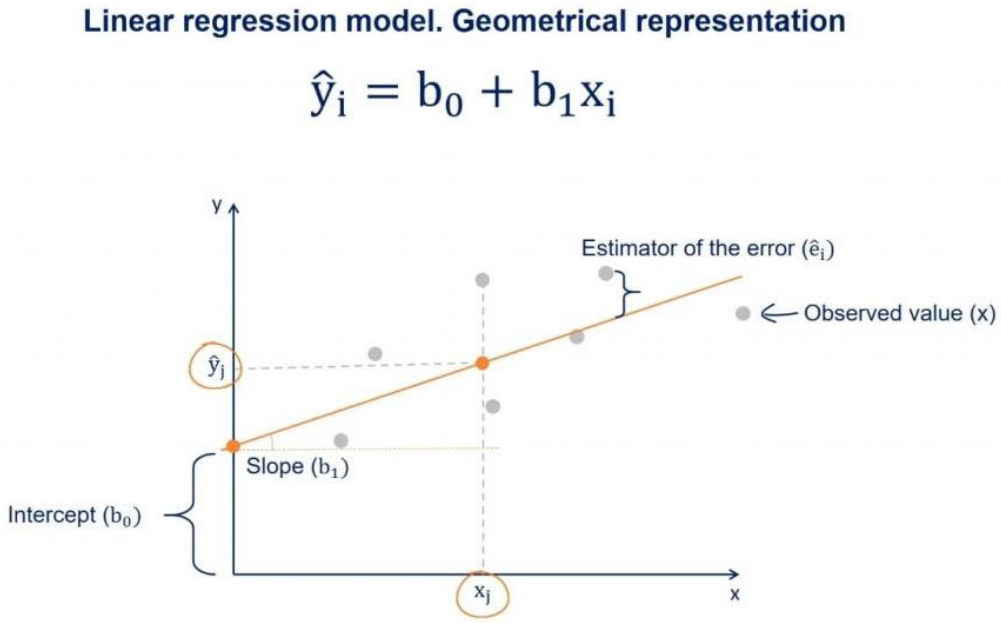
Simple Linear Regression Equation

$$\hat{y} = b_0 + b_1 x_1$$

The $\hat{y}$ here is referred to as *y hat*. Whenever we have a hat symbol, it is an estimated or predicted value.

The Regression Line

When we plot the data points on an *x-y* plane, the **regression line** is the best-fitting line through the data points.



The grey points that are scattered are the observed values. B_0 is a *constant* and is the intercept of the **regression line** with the y-axis. B_1 is the slope of the **regression line**. It shows how much *y* changes for each unit change of *x*.

The distance between the observed values and the **regression line** is the *estimator of the error term epsilon*. Its point estimate is called residual.
Now, suppose we draw a perpendicular from an observed point to the **regression line**. The intercept between that perpendicular and the **regression line** will be a point with a *y* value equal to $\hat{y}$. Given an *x*, $\hat{y}$ is the value predicted by the **regression line**.

Python Packages for Linear Regression

NumPy is a fundamental Python open source scientific package that allows many high-performance operations on single-dimensional and multidimensional arrays. It also offers many mathematical routines.

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

Matplotlib is a graph plotting library in python that serves as a visualization utility.

The package scikit-learn is a widely used Python library for machine learning, built on top of NumPy and some other packages. It provides the means for preprocessing data, reducing dimensionality, implementing regression, classifying, clustering, and more. Like NumPy, scikit-learn is also open-source.

Statsmodels isa powerful python package to implement linear regression and need functionality beyond the scope of scikit-learn. It is a powerful Python package for the estimation of statistical models, performing tests, and more. It is open-source as well.

Interpreting StatsModels Results:

R-squared value: This is a statistical measure of how well the regression line fits with the real data points. The higher the value, the better the fit.

Adj, R-squared: This is the corrected R-squared value according to the number of input features. Ideally, it should be close to the R-squareds value.

Coefficient: This gives the *'M'* value for the regression line. It tells how much the output variable y changes with a unit change in input variable x. A positive value means that the two variables are directly proportional. A negative value, however, would have meant that the two variables are inversely proportional to each other.

Std error: This tells us how accurate our coefficient value is. The lower the standard error, the higher the accuracy.

P >|t| : This is the p-value. It tells us how statistically significant the input variable is are to the output variable. A value less than 0.05 usually means that it is quite significant.

Program:

(Note: Execute the program and attach printout)

Result:

The concept of Linear Regression is studied and the program is successfully executed with the given dataset.