UNIT-IV

UNIT-IV

Analysis of Variance (ANOVA): Introduction to ANOVA, one way ANOVA, two way ANOVA, Post – Hoc test

Regression: Simple Linear Regression, Multiple Linear Regression, Maximum Likelihood Estimation (MLE), Logistic Regression, step-wise methods and algorithms

Reference Books:

1. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".
2. Swaroop, C. H. (2003). A Byte of Python. Python Tutorial.
3. Ken Black, sixth Editing. Business Statistics for Contemporary Decision Making. “John Wiley & Sons, Inc”.
4. Anderson Sweeney Williams (2011). Statistics for Business and Economics. “Cengage Learning”.

Introduction to ANOVA

- Analysis of Variance (ANOVA) is a statistical method used to analyze the differences among group means in a sample.
- It is particularly **useful when comparing three or more groups to determine if there are statistically significant differences between them**.
- ANOVA assesses whether the variability within groups is comparable to the variability between groups.
- **ANOVA compares the variation between group means to the variation within the groups**.
- If the variation between group means is significantly larger than the variation within groups, it suggests a significant difference between the means of the groups.
- ANOVA is used to compare treatments, analyze factors impact on a variable, or compare means across multiple groups.
- Types of ANOVA include **One-way (for comparing means of groups)** and **Two-way (for investigating effects of two independent variables on a dependent variable).**

When to Use ANOVA:

- As an analyst, you might use Analysis of Variance (ANOVA) to test a particular hypothesis. Use ANOVA to figure out how your various groups react, with the null hypothesis being that the means of the various groups are equal.
- **If the difference between the two populations is statistically significant, then the two populations are unequal**.
- Now that we have understood what ANOVA is, let's understand some important terms related to ANOVA.

Important Terms Related to ANOVA

Means (Grand and Sample):

- A sample mean is the average value for a group, whereas the grand mean is the average of sample means from various groups or the mean of all observations combined.

F-Statistics:

- F-statistic or F-ratio is a statistical measure that tells us about the extent of difference between the means of different samples. Lower the F-ratio, closer are the sample means

Sum of Squares

- The sum of squares is a technique used in regression analysis to determine the dispersion of data points. It is used in the ANOVA test to compute the value of F.

Mean Squared Error (MSE)

- The Mean Squared Error gives us the average error in the data set.

Hypothesis

- In ANOVA, we have Null Hypothesis and an Alternative Hypothesis. The Null hypothesis is valid when all the sample means are equal, or they don't have any major difference.
- The Alternate Hypothesis is valid when at least one of the sample means is different from the other.

Group Variability

- In ANOVA, a group is a set of samples within the independent variable.
- Between-group variability occurs when there is a significant variation in the sample distributions of individual groups.
- Within-group variability occurs when there are variations in the sample distribution within a single group.

One-Way ANOVA

- **What is one-way ANOVA?**
- The most common method of performing an ANOVA test is one-way ANOVA. The one-way ANOVA means that the analysis of variance has **one independent variable**.
- One-way analysis of variance (ANOVA) is a statistical method for testing for differences in the means of three or more groups.
- **How is one-way ANOVA used?**
- One-way ANOVA is typically used when you have a single independent variable, or factor, and your goal is to investigate if variations, or different levels of that factor have a measurable effect on a dependent variable.
- **Use the one-way ANOVA to see if there are any significant differences between the means of independent variables.**
- **When you know how each independent variable's mean differs from the others, you can figure out which of them is linked to your dependent variable and start to figure out what's driving that behavior.**

What are some limitations to consider?

- One-way ANOVA, or one-way analysis of variance, is a statistical method used to test for differences in the means of three or more groups based on a single independent variable.
- One-way ANOVA can only be used when investigating a single factor and a single dependent variable.
- When comparing the means of three or more groups, it can tell us if at least one pair of means is significantly different, but it can't tell us which pair. Also, it requires that the dependent variable be normally distributed in each of the groups and that the variability within groups is similar across groups.

One-way ANOVA is a test for differences in group means:

- One-way ANOVA is a statistical method to test the null hypothesis (H0) that three or more population means are equal vs. the alternative hypothesis (Ha) that at least one mean is different. Using the formal notation of statistical hypotheses, for k means we write:

  $H0: \mu 1 = \mu 2 = \cdots = \mu k$

  Ha: not all means are equal

where μi is the mean of the i-th level of the factor

- **Let's use the example of three teaching methods (Method A, Method B, and Method C) with the following test scores:**

  Method A: 85, 88, 90, 92, 87

  Method B: 78, 80, 82, 85, 81

  Method C: 88, 85, 84, 87, 90

- We'll perform a one-way ANOVA to test if there are any significant differences in the mean test scores between the teaching methods.

Step 1: Set Hypotheses

- Null Hypothesis (H0): There is no significant difference in the mean test scores between Method A, Method B, and Method C.
- Alternative Hypothesis (H1): There is a significant difference in the mean test scores between at least two teaching methods.

Step 2: Calculate Means

❖ Calculate the mean test score for each group:

Mean A = (85 + 88 + 90 + 92 + 87) / 5 = 88.4

Mean B = (78 + 80 + 82 + 85 + 81) / 5 = 81.2

Mean C = (88 + 85 + 84 + 87 + 90) / 5 = 86.8

Step 3: Calculate Sum of Squares (SSW and SSB) within the Groups / Between the Groups

❖ Let's calculate the sum of squares (SSW and SSB) for the productivity scores in each training program:

$$SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2$$
$$SSB = \sum_{i=1}^{k} n_i (\bar{X}_i - \bar{X})^2$$

Where:

- X_{ij} is the score of the j-th observation in the i-th group.
- $\bar{X}_i$ is the mean of the i-th group.
- $\bar{X}$ is the overall mean.
- n_i is the number of observations in the i-th group.
- k is the number of groups.

❖ Let's substitute the values into the formulas for SSW_A, SSB_A, SSW_B, SSB_B, SSW_C, and SSB_C to show the calculations

For Program A:

$$SSW_A = \sum_{j=1}^{5}(X_{Aj} - 88.4)^2$$
$$SSW_A = (85 - 88.4)^2 + (88 - 88.4)^2 + (90 - 88.4)^2 + (92 - 88.4)^2 + (87 - 88.4)^2$$

$$SSW_A \approx 1.6^2 + (-0.4)^2 + 1.6^2 + 3.6^2 + (-1.4)^2$$
$$SSW_A \approx 2.56 + 0.16 + 2.56 + 12.96 + 1.96$$
$$SSW_A \approx 20.2$$

- Calculate overall mean ($\bar{X}$):

$$\bar{X} = \frac{85+88+90+92+87+78+80+82+85+81+88+85+84+87+90}{15} = 86$$

$$SSB_A = 5 \times (88.4 - 86)^2 = 28.8$$

For Program B:

$$SSW_B = \sum_{j=1}^{5}(X_{Bj} - 81.2)^2$$
$$SSW_B = (78 - 81.2)^2 + (80 - 81.2)^2 + (82 - 81.2)^2 + (85 - 81.2)^2 + (81 - 81.2)^2$$

$$SSW_B \approx (-3.2)^2 + (-1.2)^2 + 0.8^2 + 3.8^2 + (-0.2)^2$$
$$SSW_B \approx 10.24 + 1.44 + 0.64 + 14.44 + 0.04$$
$$SSW_B \approx 26.8$$

$$SSB_B = 5 \times (81.2 - 86)^2 = 105.8$$

For Program C:

$$SSW_C = \sum_{j=1}^{5}(X_{Cj} - 86.8)^2$$
$$SSW_C = (88 - 86.8)^2 + (85 - 86.8)^2 + (84 - 86.8)^2 + (87 - 86.8)^2 + (90 - 86.8)^2$$

$$SSW_C \approx 1.2^2 + (-1.8)^2 + (-2.8)^2 + 0.2^2 + 3.2^2$$
$$SSW_C \approx 1.44 + 3.24 + 7.84 + 0.04 + 10.24$$
$$SSW_C \approx 22.8$$

$$SSB_C = 5 \times (86.8 - 86)^2 = 5$$

$$SSW = SSW_A + SSW_B + SSW_C \approx 69.8$$
$$SSB = SSB_A + SSB_B + SSB_C = 28.8 + 105.8 + 5 = 139.6$$

Step 4: Calculate Degrees of Freedom (DFW and DFB)

$$DFW = N - k = 15 - 3 = 12$$

$$DFB = k - 1 = 3 - 1 = 2$$

Step 5: Calculate Mean Squares (MSW and MSB)

$$MSW = \frac{SSW}{DFW} \approx \frac{69.8}{12} \approx 5.82$$

$$MSB = \frac{SSB}{DFB} = \frac{139.6}{2} = 69.8$$

Step 6: Calculate F Ratio

$$F = \frac{MSB}{MSW} \approx \frac{69.8}{5.82} \approx 11.98$$

Step 7: Decision Rule

- With $DFB = 2$ and $DFW = 12$ at a significance level of 0.05, the critical F-value is approximately 3.89.
- Since $F > 3.89$ (11.98 > 3.89), we reject the null hypothesis.

Step 8: Conclusion

❖ There is sufficient evidence to suggest that there are significant differences in the mean test scores between at least two teaching methods.

Important Note:

❖ The critical F-value of 3.89 is a hypothetical value used for illustration purposes. It doesn't correspond to any specific degrees of freedom.

Two-Way ANOVA

- An ANOVA test is the first step in identifying factors that influence a given outcome. Once an ANOVA test is performed, a tester may be able to perform further analysis on the systematic factors that are statistically contributing to the data set's variability.
- A two-way ANOVA test reveals the results of **two independent variables** on **a dependent variable**. ANOVA test results can then be used in an F-test, a statistical test used to determine whether two populations with normal distributions share variances or a standard deviation, on the significance of the regression formula overall.
- ANOVA (Analysis of Variance) is a statistical test used to analyze the difference between the means of more than two groups.
- A two-way ANOVA is used **to estimate how the mean of a quantitative variable changes according to the levels of two categorical variables**. Use a two-way ANOVA when you want to know how two independent variables, in combination, affect a dependent variable.

When to use a two-way ANOVA

- You can use a two-way ANOVA when you have collected data on a quantitative dependent variable at multiple levels of two categorical independent variables.
- A quantitative variable represents amounts or counts of things. It can be divided to find a group mean.

How Does Two-way ANOVA Test Works?

- The two-way ANOVA test is an extension of the one-way ANOVA test, although the latter is a more hypothetical one.
- An ANOVA test determines whether a statistical operation has outcomes that are useful or not.
- In essence, it allows one to determine whether to reject or accept a null hypothesis. In a two-way ANOVA test, two variables are used to determine this.
- The two-way ANOVA test reveals whether the two important variables affect the outcome or a dependent variable.
- One can then use the result to work out variances and to carry out an f-test. The two-way ANOVA test is similar to the two-sample t-test but has the benefit of having a lower chance of getting type 1 errors, which could corrupt the data collected.

- The two-way ANOVA is versatile; it can compare means and variances within-subjects, between groups, within groups, and even between test groups.
- Two-way ANOVA helps to assess whether there are significant main effects for each independent variable and whether there is an interaction effect between them. It is a powerful tool for analyzing experimental designs with two categorical factors.
- Two-way Analysis of Variance (ANOVA) is a statistical method used to analyze the influence of two independent categorical variables on a dependent variable. It is an extension of the one-way ANOVA, which deals with a single factor. The two-way ANOVA helps to understand how two factors simultaneously impact the mean of a dependent variable.

Let's consider another example involving two-way ANOVA to analyze the influence of two factors on a dependent variable.

- **Scenario:** Evaluating the Impact of Fertilizer Type and Watering Frequency on Plant Growth.
- **Independent Variables:**
- **Fertilizer Type (Factor A):** Organic fertilizer and synthetic fertilizer.
- **Watering Frequency (Factor B):** Daily watering and weekly watering.
- **Dependent Variable:**
- Plant Growth: Measured in height (in centimeters) after 8 weeks.

Hypotheses:

- Null Hypothesis (H_0): There is no significant effect of fertilizer type on plant growth.
- Null Hypothesis (H_0): There is no significant effect of watering frequency on plant growth.
- Null Hypothesis (H_0): There is no interaction between fertilizer type and watering frequency on plant growth.

Procedure:

- Plants are randomly assigned to one of four groups: organic fertilizer/daily watering, organic fertilizer/weekly watering, synthetic fertilizer/daily watering, and synthetic fertilizer/weekly watering.
- Plant growth is measured for each group after 8 weeks.

Analysis:

- Two-way ANOVA will be conducted to assess the impact of fertilizer type, watering frequency, and their interaction on plant growth.
- If the result shows a significant main effect for fertilizer type, it implies that the choice of fertilizer significantly affects plant growth.
- Similarly, a significant main effect for watering frequency suggests that the frequency of watering significantly influences plant growth.
- If there is a significant interaction, it indicates that the combined effect of fertilizer type and watering frequency is not simply additive.

POST HOC TEST ANOVA

- A post hoc test, also known as a post hoc analysis or pairwise comparison, is conducted after an analysis of variance (ANOVA) to determine specific group differences when the overall ANOVA results indicate a significant difference among three or more groups.
- ANOVA is a statistical method used to analyze the variation between group means in a dataset to determine if there are statistically significant differences among the groups.
- When the ANOVA test indicates a significant difference, it doesn't specify which groups are different from each other.
- Post hoc tests are employed to make pairwise comparisons between individual groups and identify where the differences lie.
- The need for post hoc tests arises because conducting multiple pairwise comparisons without adjustment increases the likelihood of making a Type I error (incorrectly rejecting a true null hypothesis).

- There are various post hoc tests available, and the choice of a specific test may depend on the design of the study and assumptions. Common post hoc tests include Tukey's Honestly Significant Difference (HSD), Bonferroni correction, Scheffé's method, and others. Here's a brief explanation of Tukey's HSD, which is widely used:

Here's a simplified step-by-step process for using Tukey's HSD:

- **Conduct ANOVA:** Start by performing the ANOVA to determine if there are significant differences among group means.
- **Identify Significance:** If the ANOVA results indicate a significant difference, proceed to post hoc testing.
- **Tukey's HSD:** Apply Tukey's HSD test to compare all possible pairs of group means.
- **Critical Value:** Calculate a critical value based on the number of groups and observations. If the difference between the means of two groups exceeds this critical value, the difference is considered significant.
- **Interpretation:** Identify which specific groups have significantly different means.

Regression: Simple Linear Regression

- Machine learning is a subset of AI, which enables the machine to automatically learn from data, improve performance from past experiences, and make predictions.
- Machine learning contains a set of algorithms that work on a huge amount of data. Data is fed to these algorithms to train them, and on the basis of training, they build the model & perform a specific task.
- Machine learning (ML) is a type of artificial intelligence (AI) focused on building computer systems that learn from data.
- The broad range of techniques ML encompasses enables software applications to improve their performance over time.
- Machine learning algorithms are trained to find relationships and patterns in data.
- They use historical data as input to **make predictions, classify information, cluster data points**, reduce dimensionality and even help generate new content.

Regression: Simple Linear Regression

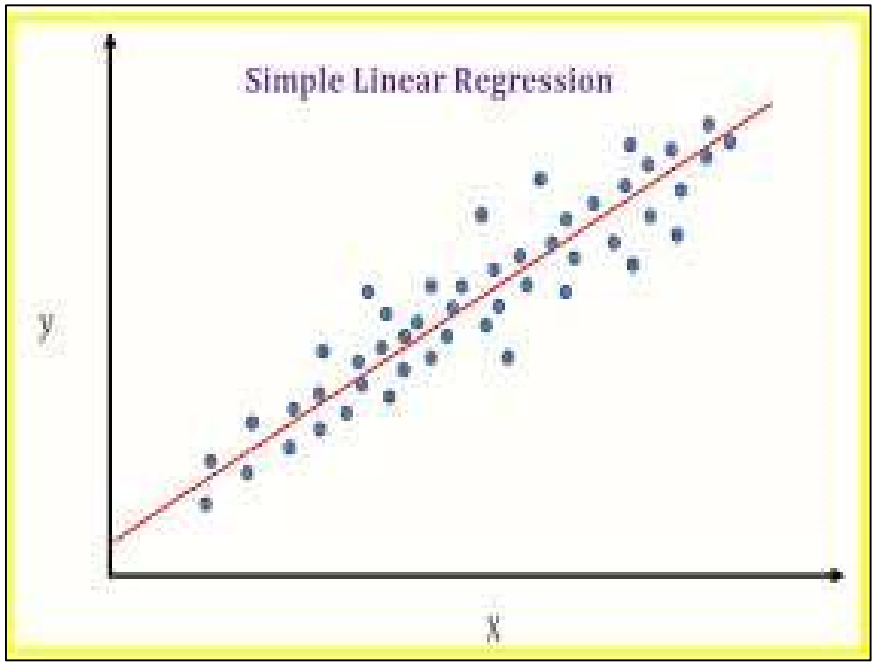
- Linear regression predicts the relationship between two variables by assuming a linear connection between the independent and dependent variables.
- It try to find the optimal line that minimizes the sum of squared differences between **predicted and actual values**.
- Applied in various domains like economics and finance, this method analyzes and forecasts data trends. It can extend to multiple linear regression involving several independent variables and logistic regression, suitable for binary classification problems
- **Linear regression is a Machine Learning model that depends on the linear relationship between a dependent variable and one or more independent variables.**

Problem Statement example for Simple Linear Regression:

- Here we are taking a dataset that has two variables: salary (dependent variable) and experience (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 independent variable.

- Let us understand the word ‘linear relationship’. We can say two variables are in linear relationship when their values can be represented using a straight line. That means, the data points (or values of the variables) lie on the straight line.
- **When there is only one independent variable, it is called Simple Linear regression.**
- **For more than one independent variables, the model is called Multiple Linear regression**.

Simple linear regression is a regression model that estimates the relationship between one independent variable and one dependent variable using a straight line. Both variables should be quantitative.



Finding the best fit line:

- When working with linear regression, our main goal is to **find the best fit line** that means the error between predicted values and actual values should be minimized. **The best fit line will have the least error.**
- The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

Cost function-

- The different values for weights or coefficient of lines (a_0, a_1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
- Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
- We can use the cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.

- For Linear Regression, we use the **Mean Squared Error (MSE) cost function**, which is the average of squared error occurred between the predicted values and actual values. It can be written as:
- For the above linear equation, MSE can be calculated as:

$$MSE = 1\frac{1}{N}\sum_{i=1}^{n}(y_i - (a_1x_i + a_0))^2$$

Where,

N=Total number of observation

yi = Actual value

$(a_1x_i+a_0)$= Predicted value.

- **Residuals: The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.**

R-squared method:

- R-squared is a statistical method that determines the goodness of fit.
- It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.
- The **high value of R-square determines the less difference between the predicted** values and actual values and hence represents a **good model**.
- It is also called a coefficient of determination, or coefficient of multiple determination for multiple regression.
- It can be calculated from the below formula:

$$\text{R-squared} = \frac{\text{Explained variation}}{\text{Total Variation}}$$

Gradient Descent:

- Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
- A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
- It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

The Linear Equation

- In mathematics, to draw a line, we use the equation:

  $y=mx+b$

- Where m is slope and b is a constant.
- This equation is useful to find y value depending on x value. Here, **y is called the dependent variable** and **x is known as independent variable**.
- In Statistics, we write the same equation as:
- Here, the **slope is β1**. The **constant value β0** is called intercept. β0, indicates the distance on y axis.
- Let us take a linear equation: **$y = 4+2x$**. Compare this with **$Y=\beta0+\beta1X$.**
- Here, **β0=4, axis and β1=2**. By substituting x value into this equation, we can find the value of y.
- So, x is called independent variable and y is called dependent variable since **y value is dependent on x value.**

y = 4+2x

If x = 0, then y = 4+2(0) = 4.

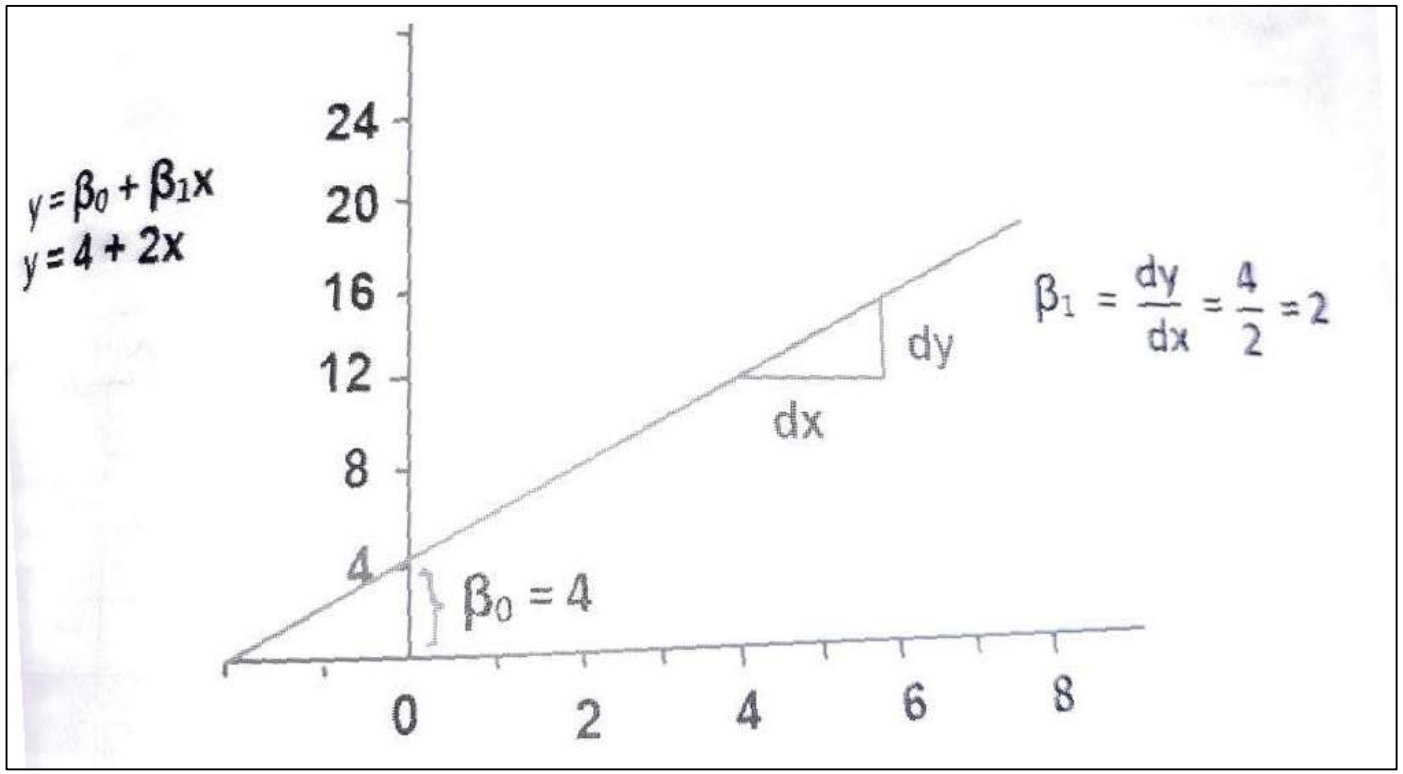
If x=2, then y = 4+2(2) = 8.

If x=4, then y = 4+2(4) = 12.

If x = 6, then y = 4+2(6) = 16.

- In this manner, when x is increased in steps of 2, y values are increasing in steps of 4. These values are shown in the Figure 1. From the Figure, we can calculate the **slope β1**, and **intercept β0**, values, as:
- Slope β1 = deviation in y / deviation in x = dy / dx = 4/ 2=2
- Intercept β0= distance on y axis where the line crosses = 4.
- **Why we called y = 4+2x a linear equation?** Because when the x values and y values are shown in the form of a graph as in Figure 1, it will show a straight line.
- **That means the relation between independent variable [x] and dependent variable [y] is linear.**
- **When such a relation exists between the data, we can apply Linear regression analyze the data**.

Figure 1: Understanding Linear Equation



Squared Value

- After reading the data from the dataset, we can plot them in the form of a graph. The points may not be exactly on the straight line. There will be deviations from the straight line. This is called error E. The linear regression should consider this error also Hence the formula will be:

  $Y=\beta 0+\beta 1X+E$

- Let us discuss about this error term "E" now. Suppose the expected values of y are given as:

  y= [1, 2, 3, 4, 5]

- But the y values are deviated due to the deviations of the data points from the straight line. These deviated y values are:

  y1= [0.8, 2.5, 3, 4.8, 4.4]

- That means we should get 1 but we got 0.8 as y value. This difference **is called residual error (E_1).**

- **We should square these errors. If we do not square them, while finding their total, the positive and negative values may cancel out. Hence squaring is needed (E_1^2).**

- Similarly, we have to calculate the differences of y values from their mean. These are deviations from the mean value (E_2 = y-mean). We have to square this value (E_2^2).
- **Now, r squared value = 1- (Sum of $E1^2$ / Sum of $E2^2$)**
- The above formula can be used to find the value of r squared.
- Now observe the following table to understand how to calculate r squared value

y	y1	E1 = y – y1	$E1^2$	E2 = y - Mean	$E2^2$
1	0.8	0.2	0.04	-2	4
2	2.5	-0.5	0.25	-1	1
3	3	0	0	0	0
4	4.8	-0.8	0.64	1	1
5	4.4	0.6	0.36	2	4
Mean = 3			Sum1 = 1.29		Sum2= 10

- Using the above table data, the formula will be:

  r squared =1 - (Sum1 / Sum2) = 1 - (1.29 / 10)=0.871

- r squared value is also called 'Coefficient of determination'.
- The r squared value obtained by us is 0.871. In percentage, it is 0.871X100=87.1. This indicates 87% accuracy level for the model.
- That means the Linear regression model in this example can explain 87% of deviations for successfully, whereas the remaining 23% cannot be explained by the model. That means there is chance of error level at 23%. The accuracy level of the model is 87%.
- **R squared value will be in the range of 0 to 1.**
- **If r squared value is closer to 1, then the actual value and predicted values (on the line) will be very close.**
- **It represents high accuracy of the model.**
- **When r squared value is nearer to 0, they are much apart.**
- **So, the prediction may not be correct.**

- Based on the r squared value prediction will change.
- The coefficient of determination is a statistical measurement that examines how differences in one variable can be explained by the difference in a second variable, when predicting the outcome of a given event.
- In other words, this coefficient, which is more commonly known as R-squared (or R2), assesses how strong the linear relationship is between two variables.
- What happens when there are no deviations between **y and y1**. That means, El value will be 0. Then $E1^2$ will also be 0.
- Then the sum of squares of errors (Sum1) will be 0.
- This indicates 100% accuracy for the model. So, the point is this: in Linear regression models, the sum of squares should have least value and that represents high accuracy.
- This is the reason, **'Simple Linear Regression model'** is also called Least Squares Regression model'.

- The following Python code explains how to calculate r squared value for the above example. ‘sklearn' is a package from scikit-learn.org that contains many machine learning related modules.
- In sklearn, we have a module by the name metrics. This module contains a function r2_score(). By calling this function and passing the original data and predicted data, we can find the r squared value as shown below:

- Scikit-learn (Sklearn) is **the most useful and robust library for machine learning in Python**.
- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

```
from sklearn.metrics import r2_score
y =[1, 2, 3, 4, 5]
y1 =[0.8,2.5,3,4.8,4.4]
R_Square = r2_score (y, y1)
print('Coefficient of Determination', R_Square)
```

```
Coefficient of Determination 0.871
```

Practical Use of Simple Linear Regression

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

- pandas is used for data manipulation and data analysis.
- numpy is used for numerical operations.
- matplotlib.pyplot is used for data visualization.
- train_test_split from sklearn.model_selection is used to split the dataset into training and testing sets.
- LinearRegression from sklearn.linear_model is used to perform linear regression.
- mean_squared_error from sklearn.metrics is used to calculate Mean Squared Error.

Loading the Dataset:

```
data = pd.read_csv("Salary_Data.csv")
```

❖ This line reads the dataset from the file "Salary_Data.csv" into a pandas DataFrame called data.

Out[5]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525

Extracting Features and Target Variable:

```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

❖ X contains the features (independent variables), which are the 'YearsExperience' column from the dataset.

❖ y contains the target variable (dependent variable), which is the 'Salary' column from the dataset.

❖ data.iloc[:, :-1] selects all rows and all columns except the last one from the DataFrame data.

❖ This means we're selecting all the features (independent variables) in the dataset, except for the last column, which is typically the target variable.

Splitting the Dataset:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

- splits the dataset into training and testing sets.
- test_size=0.2 indicates that 20% of the data will be used for testing, and the rest for training.
- random_state=0 ensures reproducibility by fixing the random seed.

Training the Model:

```
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

- LinearRegression is instantiated to create a linear regression model.
- fit() is used to train the model on the training data (X_train and y_train).

Making Predictions:

```
y_pred = regressor.predict(X_test)
```

❖ predict() is used to make predictions on the test data (X_test).

Calculating Mean Squared Error (MSE):

```
mse = mean_squared_error(y_test, y_pred)
```

❖ mean_squared_error() calculates the mean squared error between the actual target values (y_test) and the predicted values (y_pred).

Intercept and Coefficient:

```
intercept = regressor.intercept_
coefficient = regressor.coef_[0]
```

❖ The intercept and coefficient of the linear regression model are obtained from the intercept_ and coef_ attributes of the regressor object, respectively.

Predicting a new value

```
new_experience = 5
# Example: predicting salary for someone with 5 years of experience
predicted_salary = regressor.predict([[new_experience]])
print("Predicted Salary for {} years of experience: {:.2f}".format(new_experience, predicted_salary[0]))
```

- This code predicts the salary for a new value of years of experience (new_experience) using the trained linear regression model (regressor).
- Replace new_experience with the desired value for prediction. The predicted salary is printed to the console.

Visualizing the Training Set Results:

```
print("MSE:", mse)
print("Intercept:", intercept)
print("Coefficient:", coefficient)
```

```
MSE: 12823412.298126562
Intercept: 26780.099150628186
Coefficient: 9312.575126729187
```

```
plt.scatter(X_train, y_train, color='red')
plt.plot(X_train, regressor.predict(X_train), color='blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

MULTIPLE LINEAR REGRESSION

- In Simple Linear Regression model, we take only 1 independent variable (x) that is useful to predict the dependent variable (y) value. For example, 'area' of the house is useful to predict the house price.
- But in many practical cases, house price' does not dependent only on area. The price of a house can be decided depending on various factors like 'area', the number of bedrooms', the age of the house', etc. These variables are called independent variables and they are useful to predict the price of the house. When we use multiple (more than 1) variables in Linear Regression model, it is called Multiple Linear Regression Model'.
- **In Multiple Linear Regression, the target variable (y) value can be calculated based on several independent variables (xl, x2, x3, ...). So, the equation used by this model will be in the form of:**

  y = mix1 + m2x2 +m3x3 + b

- Here, y is called the dependent variable or target variable. x1, x2, x3, .. are called independent variables or features. ml, m2, m3, .. are called quotients associated with the dependent variables. b is called intercept.

- **The relationship between y and x1 should be linear. That means it can be shown in the form of a straight line. Similarly, the relationship between y and x2 should be linear. Also, the relationship between y and x3 should be linear.**
- As usual, when Multiple Linear Regression model is applied on data, there will be certain deviations between the predicted values and original values which can be calculated through r squared value.
- The **r squared value** should be between 0 and 1. If it is nearer to 0, then the model is not performing well. If it is nearer to 1, then the model is doing well and accuracy level is high.
- Let us apply Multiple Linear Regression model on the house prices data. We will first look at the data and then understand how to use the model on the data.

The dataset: homeprices.csv

- This dataset represents home prices in Monroe Township, New Jersey, USA. This dataset is a sample dataset that contains only 6 rows and 4 columns.
- The columns are the area of the house in square foot, the number of bed rooms, age of the house in months and price of the house in dollars. This is shown in Figure:

area	bedrooms	age	price
2600	3	20	550000
3000	4	15	565000
3200		18	610000
3600	3	30	595000
4000	5	8	760000
4400	5	8	795000

- We are supposed to calculate the home price' depending on the 'area', bedrooms' and age columns. So, the Multiple Linear Regression model uses the following formula:

  y = mlx1 + m2x2 + m3x3 + b

  price m1*area + m2*bedrooms +m3*age + b

- Please observe that this dataset has a missing value in bedrooms' column. So, first of all let us clean the data or make the data ready for the model. For this purpose, we have to either delete that row that contains the missing value or substitute appropriate value in that place.

❖ To find out the missing values in the data frame (df), we can use:

df.isnul1().sum()

Output:

```
area        0
bedrooms    1
age         0
price       0
dtype: int64
```

❖ The output clearly tells us that there is 1 missing value found in the ‘bedrooms' column. Our aim is to calculate median value of all the other values in that column and then substitute that value in the place of missing value. So, first let us find the median value for ‘bedrooms’ column as:

df.bedrooms.median()

Output:4

4.0

❖ We may get a float number as a result of executing the above statement. Convert that into an integer using floor() function of math module, as:

import math

med = math.floor(df. df.bedrooms.median())

med

Output:

4

❖ Now, let us fill the median value into missing place of 'b

df.bedrooms= df.bedrooms.fillna(med)

df

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	4.0	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4400	5.0	8	795000

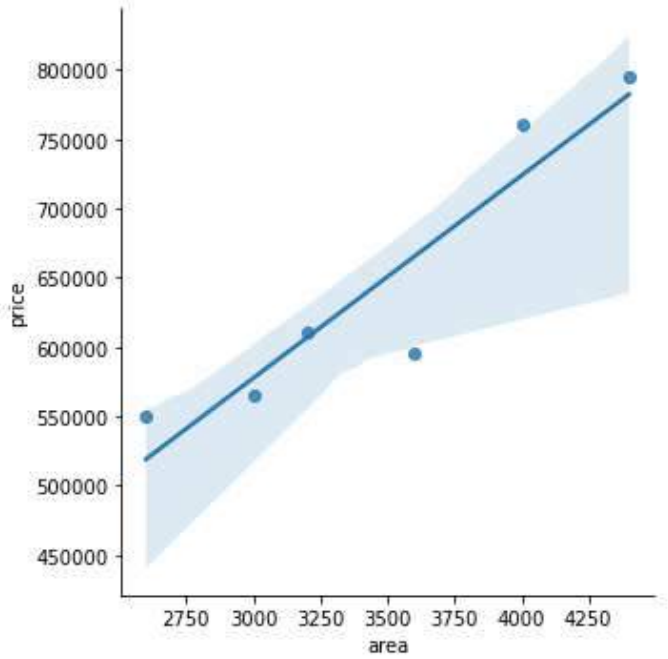
- Since the data is alright, we can check if we can apply Linear Regression model on this data of not.
- This can be done by checking the relationships between 'area' and price, between ‘bedrooms and price', and between age' and price'.
- To view these relationships, we can draw Implot that displays a scattered data points along with regression (relationship) line.
- Implots can be drawn using Implot() function of seaborn module. Now, let's use the Implot() function in 3 ways:
- To find the relation between constructed area and price by drawing lmplot as:

```
Import seaborn as sns
sns. Implot (x='area', y='price , data=df)
```

Figure Relationship between area and price of house

```
sns.lmplot(x='area', y='price',data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x24bb583d3a0>
```



- The output shows positive relationship.
- That means, if the constructed area of the house increases, the price of the house will also increase

❖ Find the relationship between the number of bedrooms and price by drawing a Implot as:

sns.lmplot(x='bedrooms', y='price',data=df)

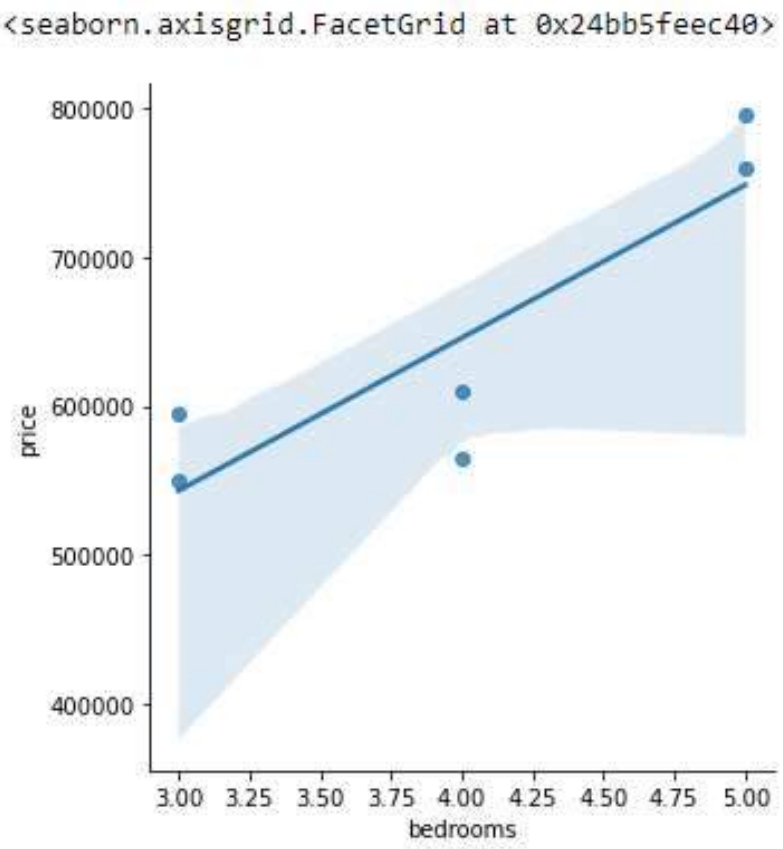


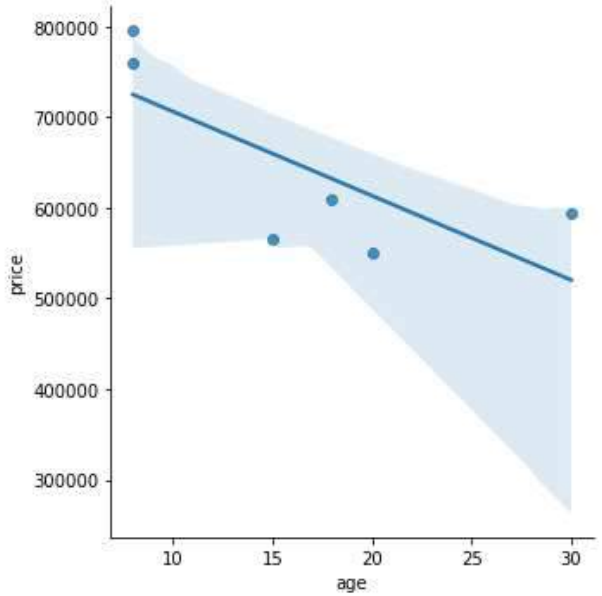
Figure:
Relationship between bedrooms and price of house

- ❖ The output shows that a **positive relationship** exists between the number of bedrooms and price.
- ❖ That means, if the number of bedrooms is increased, the price of the house will also increase

❖ To find the relation between age of the house and price by drawing Implot as:

sns.lmplot(x='age', y='price',data=df)

```
: sns.lmplot(x='age', y='price',data=df)
: <seaborn.axisgrid.FacetGrid at 0x24bb6055340>
```



❖ This output shows that there is **negative relationship** between age and price. If the age of the house is increased, the price will decrease as it represents an older house.

❖ Since there is linear (or straight line) relationship between the multiple independent:

❖ variables (i.e. area, bedrooms and age) with the dependent variable (price), we can apply Multiple Linear Regression model on the data.

- Multiple Linear Regression model is also nothing but Regression model with multiple variables. Hence, we have to create an object to LinearRegression class as:

  from sklearn.linear_model import LinearRegression

  reg=LinearRegression()

- Apply the model on the ta using fit() method. let should remember that while passing the inputs ((or independent variables), we have to pass them in the form of 2D array and the output (or dependent variable) should be given as 1D array.

  reg.fit(df[['area','bedrooms','age']], df['price'])

- To see the coefficient values used by the model, we can display coef_variable, as:

  reg.coef_array([142.895644 , -48591.66405516, -8529.30115951])

- To see the intercept value used by the model, we can display intercept_ variable as:

```
reg. intercept_
        485561.8928233979
```

- Since the model has been trained, we can predict the house price for given area, bedrooms and age values.
- This can be done using predict) method to which we have to pass the area, bedrooms and age values in the form of a 2D array.

```
#predict the price of 3000 sqft area, 3 bed rooms, 40 years old house
reg.predict([[3000,3,40]])    # 427301
Output:
        array([427301.78627387])
```

Program 1:

Develop a Python program utilizing Multiple Linear Regression to predict home prices in Monroe Township, NJ (USA). Given the attributes of square footage area, number of bedrooms, and age of the home, the program should predict the prices for the following properties:

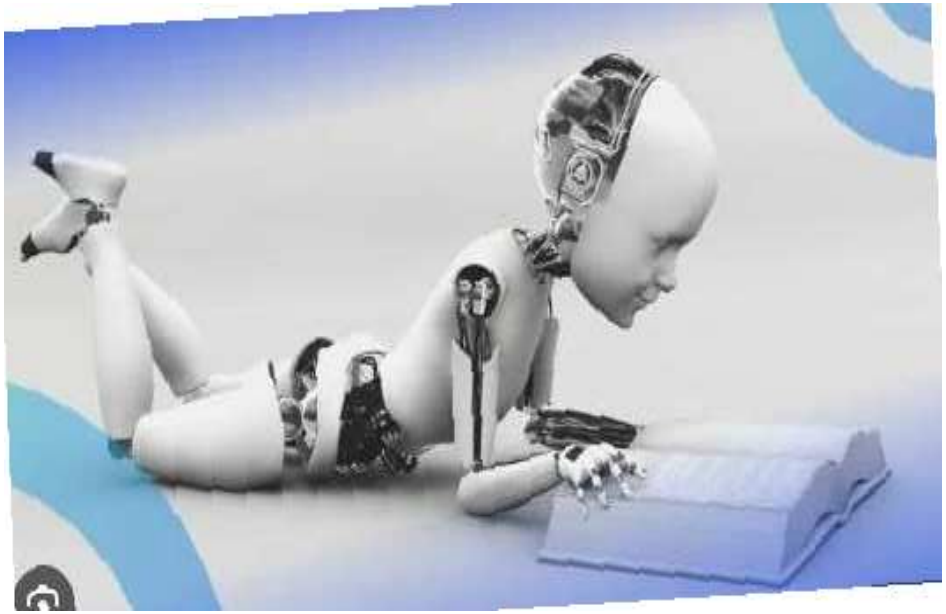
a. 3000 sqft area, 3 bedrooms, 40 years old

b. 2500 sqft area, 4 bedrooms, 5 years old

Maximum Likelihood Estimation (MLE)

EM Algorithm in Machine Learning

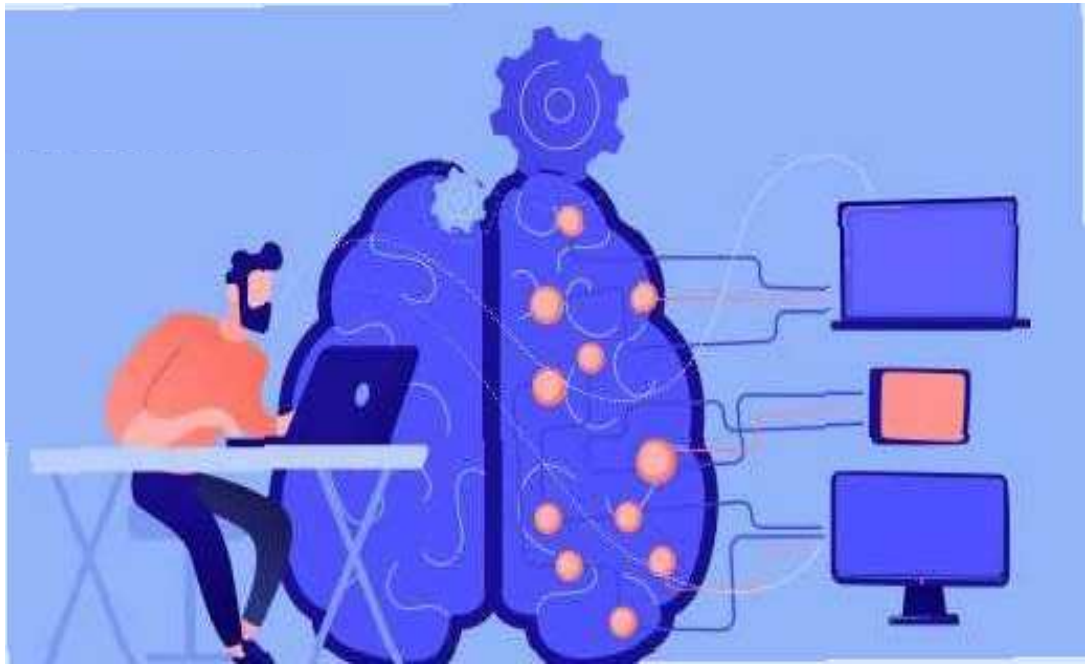
- The EM algorithm is considered a latent variable model to find the local maximum likelihood parameters of a statistical model, proposed by **Arthur Dempster, Nan Laird, and Donald Rubin in 1977**.
- The EM (Expectation-Maximization) algorithm is one of the most commonly used terms in machine learning to obtain maximum likelihood estimates of variables that are sometimes observable and sometimes not.
- However, it is also applicable to unobserved data or sometimes called latent.
- It has various real-world applications in statistics, including obtaining the mode of the posterior marginal distribution of parameters in machine learning and data mining applications.



What is an EM algorithm?

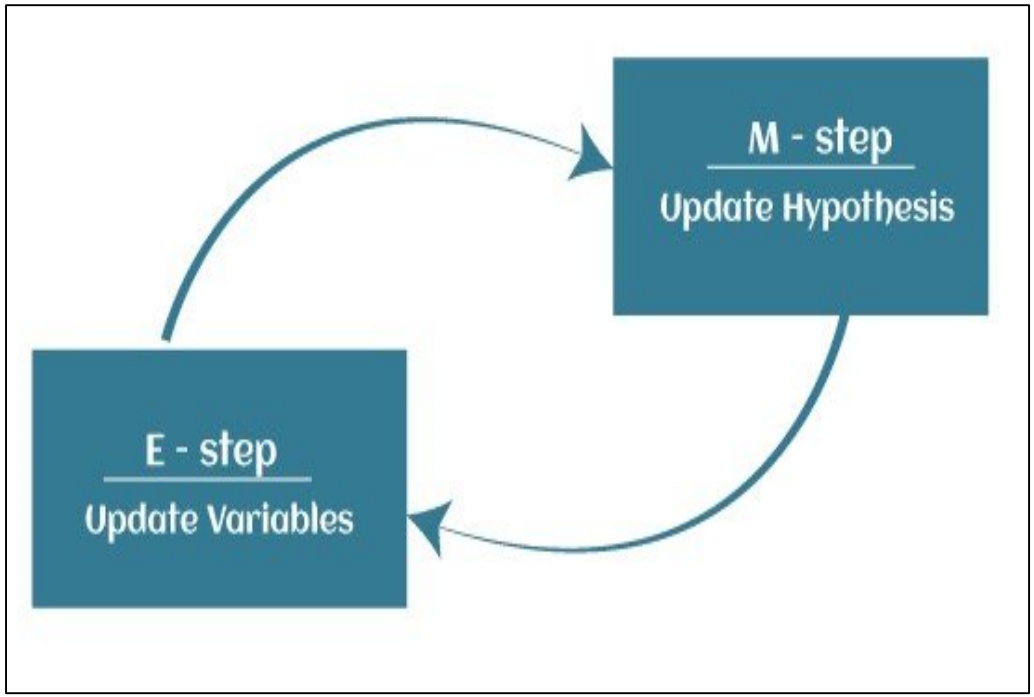
- The Expectation-Maximization (EM) algorithm is defined as the combination of various unsupervised machine learning algorithms, which is used to determine the local maximum likelihood estimates (MLE) or maximum a posteriori estimates (MAP) for unobservable variables in statistical models.
- Further, it is a technique to find maximum likelihood estimation when the latent variables are present.

- It is also referred to as the latent variable model.
- A latent variable model consists of both observable and unobservable variables where observable can be predicted while unobserved are inferred from the observed variable.
- These unobservable variables are known as latent variables.



EM Algorithm

- The EM algorithm is the combination of various unsupervised ML algorithms, such as the k-means clustering algorithm.
- Being an iterative approach, it consists of two modes.
- In the first mode, we estimate the missing or latent variables. Hence it is referred to as the Expectation/estimation step (E-step).
- Further, the other mode is used to optimize the parameters of the models so that it can explain the data more clearly.
- The second mode is known as the maximization-step or M-step.



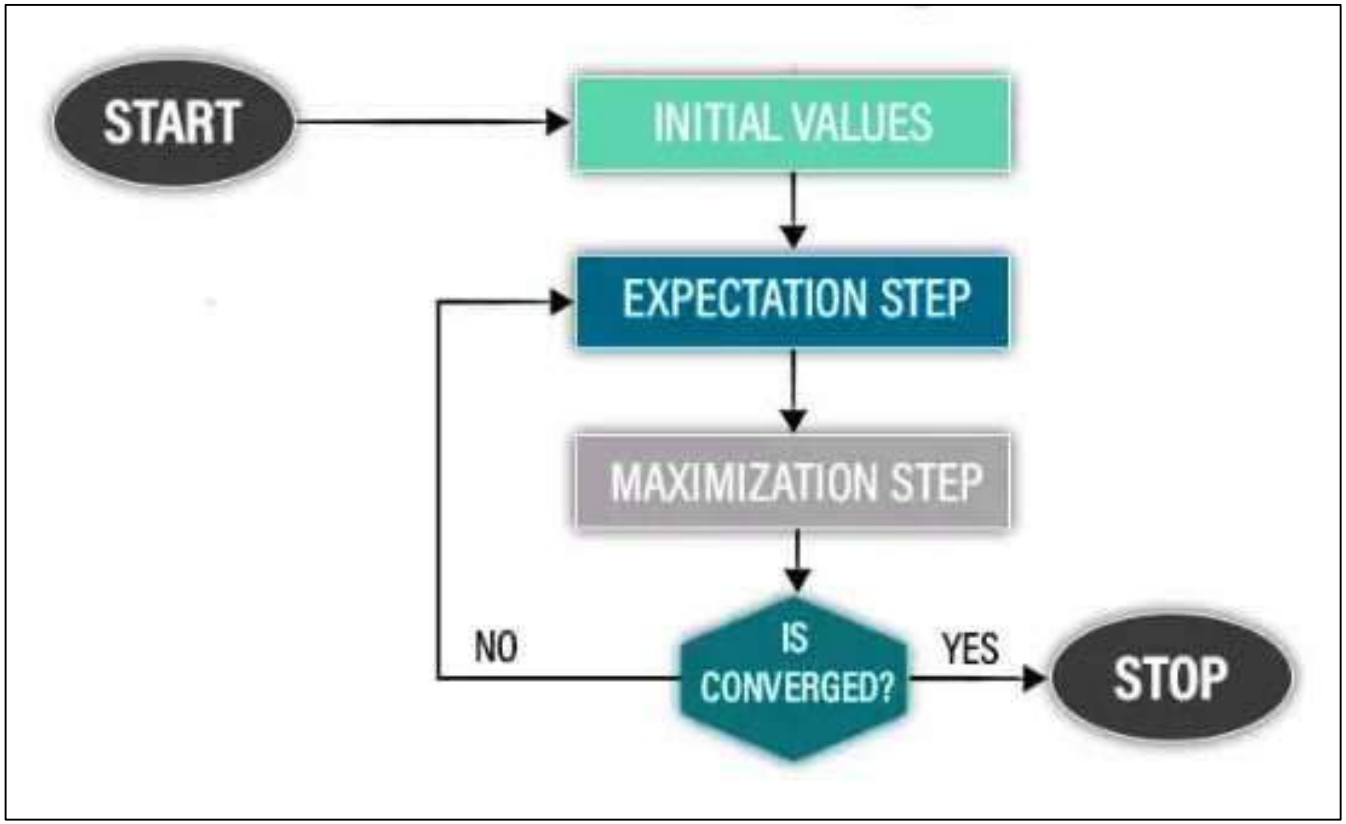
- **Expectation step (E - step):** It involves the estimation (guess) of all missing values in the dataset so that after completing this step, there should not be any missing value.
- **Maximization step (M - step):** This step involves the use of estimated data in the E-step and updating the parameters.
- Repeat E-step and M-step until the convergence of the values occurs.
- The primary goal of the EM algorithm is to use the available observed data of the dataset to estimate the missing data of the latent variables and then use that data to update the values of the parameters in the M-step.

What is Convergence in the EM algorithm?

- Convergence is defined as the specific situation in probability based on intuition, e.g., if there are two random variables that have very less difference in their probability, then they are known as converged.
- In other words, whenever the values of given variables are matched with each other, it is called convergence.

Steps in EM Algorithm

- The EM algorithm is completed mainly in 4 steps, which include Initialization Step, Expectation Step, Maximization Step, and convergence Step. These steps are explained as follows:



1st Step:

- The very first step is to initialize the parameter values. Further, the system is provided with incomplete observed data with the assumption that data is obtained from a specific model.

2nd Step:

- This step is known as Expectation or E-Step, which is used to estimate or guess the values of the missing or incomplete data using the observed data. Further, E-step primarily updates the variables.

3rd Step:

- This step is known as Maximization or M-step, where we use complete data obtained from the 2nd step to update the parameter values. Further, M-step primarily updates the hypothesis.

4th step:

- The last step is to check if the values of latent variables are converging or not. If it gets "yes", then stop the process; else, repeat the process from step 2 until the convergence occurs.

Applications of EM Algorithm:

1. Gaussian Mixture Model (GMMs):

- The EM algorithm is widely used in clustering algorithms to estimate data with similar characteristics and handle uncertainties.
- It calculates the Gaussian density, weight, shape, and location of each Gaussian components cluster by capturing clustering characteristics.
- For example, it is used in target marketing strategies to identify different groups of customers based on their purchasing behavior, such as product preference, amount spent, buying frequency, etc.

2. Hidden Markov Models (HMMs):

- The Hidden Markov Model is a probabilistic model mainly used for sequential data analysis where hidden states influence the observed data. EM estimates the parameters of HMMs, which are Emission probabilities and Transition probabilities.
- Used for speech recognition, bioinformatics, and finance.
- For example, used in speech recognition for identifying phonemes and acoustic features of audio signals, the EM algorithm can transcribe the verbal word into text.

3. Natural Language Processing (NLP):

- In natural language processing, practitioners utilize the Expectation-Maximization algorithm for diverse tasks, including text categorization, sentiment analysis, machine translation, topic modeling, and part-of-speech tagging.
- EM algorithm aids in extracting meaningful insights from text content by analyzing and learning its pattern.
- For example, NPL can efficiently identify sentiments in text by reviewing and classifying social media posts into positive, negative, or neutral.

4. Computer Vision:

- Computer vision employs the EM algorithm to recognize objects and reconstruct images. It enables us to extract valuable information from the visual data and image content.
- For example, the EM algorithm aids in reconstructing a three-dimensional image from a two-dimensional image, and these images can be any medical scans like CT or MRI, which can enhance visualization of any injuries or disease by reconstructing the process.

5. Health and Medical Industry:

- The EM algorithm has a valuable use in the health care and medical industry for improving patient treatment and care by aiding the detailed information for diagnosis by reconstructing and enhancing the diagnosed image quality.
- For example, the EM algorithm helps oncologists in cancer radiation therapy reconstruct a three-dimensional tumor image from a CT scan to target only the affected area and avoid any healthy tissue damage.

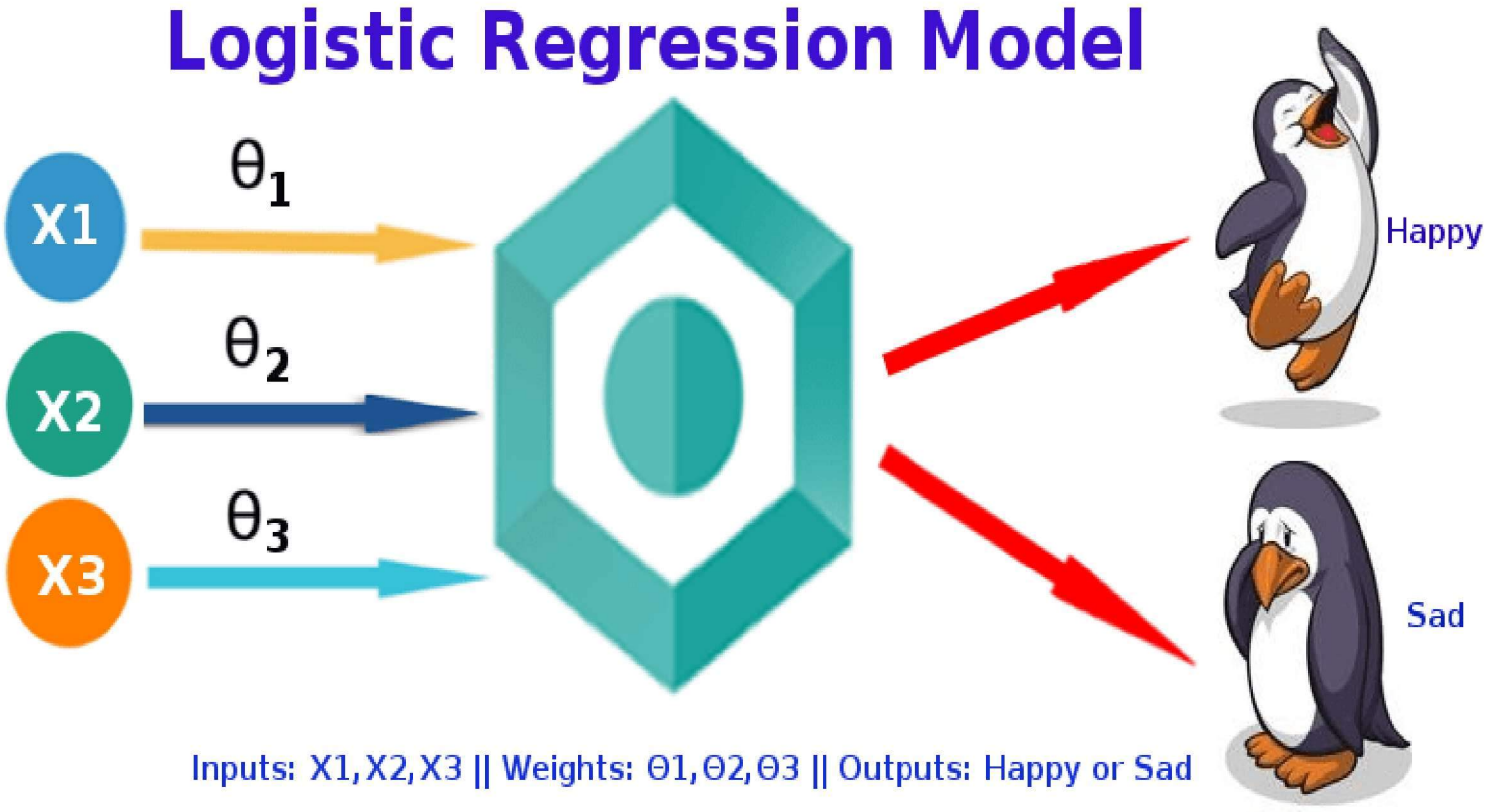
Advantages of EM algorithm

- It is very easy to implement the first two basic steps of the EM algorithm in various machine learning problems, which are E-step and M- step.
- It is mostly guaranteed that likelihood will enhance after each iteration.
- It often generates a solution for the M-step in the closed form.

Logistic Regression Model

- All the **Linear Regression** models like Simple, Multiple or Polynomial Regression models are useful when we want to predict certain value.
- For example, we can predict the home price at Boston or salary of an employee using these models. **These values are called continuous values.**
- In some cases, we want to predict values which are called 'categorical'. That means, the values representing limited number of possibilities will come under categorical values. For example,
  - An email is spam or not spam (only 2 possibilities like Spam/Not-spam)
  - Gender of a person (only 2 possibilities like male/Female)
  - The martial status of a person (only 4 possibilities like Single / Married / Widowed / Divorced)
  - Which party a citizen is going to vote for (only 3 possibilities like Congress party. BJP Party, Bharat Rashtra Samithi).
- These values come under categorical type of values. From the above examples, we can understand that the data contains limited possibilities. That means the data can be classified into 2 or 3 or 4 groups or classes etc.

❖ When we want to predict the categorical data, that means where the data is classified, we can think about Logistic Regression model. So, Logistic Regression is one of the techniques for classification.



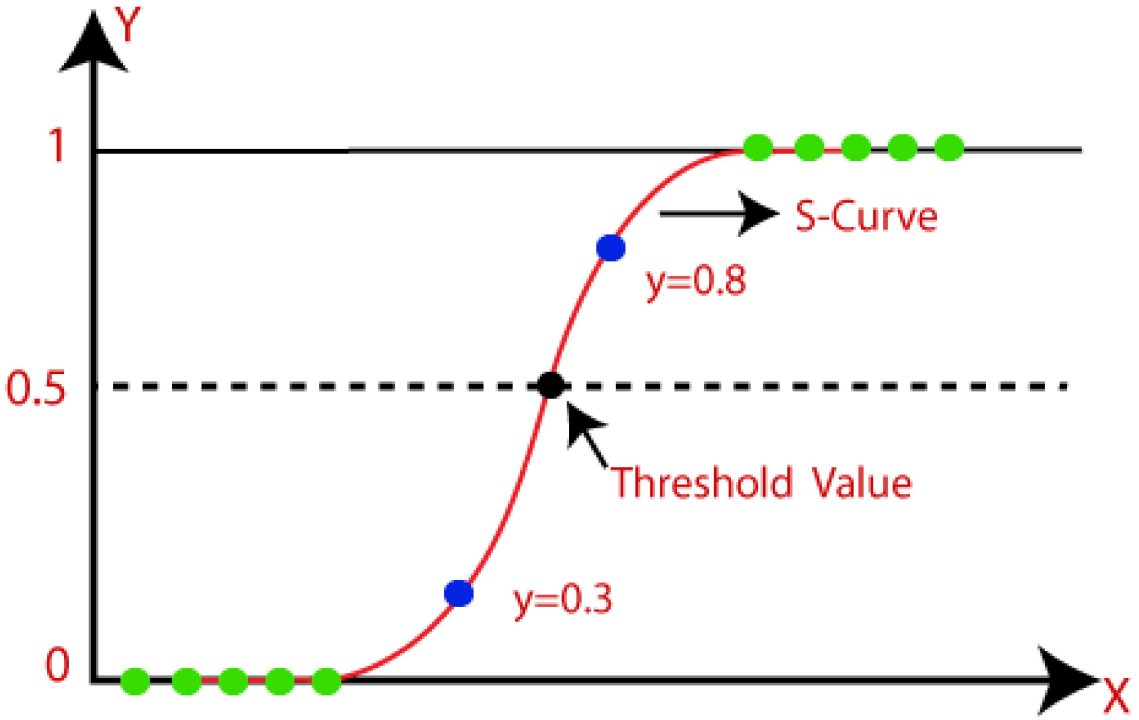
What Does Logistic Regression Mean?

- Logistic regression is a supervised learning algorithm used in machine learning to **predict the probability of a binary outcome**. A binary outcome is limited to one of two possible outcomes. Examples include yes/no, 0/1 and true/false.
- Logical regression is used in predictive modeling to analyze large datasets in **which one or more independent variables can determine an outcome.**
- The outcome is expressed as a **dichotomous variable** that has one of two possible outcomes.
- Essentially, logistic regression works by estimating the mathematical probability that an instance belongs to a specified class -- or not.

- Logistic regression uses something called the **Sigmoid function** to map predicted predictions and their probabilities.
- On a graph, if the estimated probability is **greater than a pre-defined** acceptance threshold, then the model will predict that the instance **belongs to that class**. If the estimated probability is **less than the pre-defined** threshold on the graph, then the model will predict the instance does not belong to the class.
- **The Sigmoid function performs the role of an activation function in machine learning which is used to add non-linearity in a machine learning model. Basically, the function determines which value to pass as output and what not to pass as output.**

Logistic Regression:

- Logistic regression is one of the most popular Machine learning algorithm that comes under Supervised Learning techniques.
- It can be used for **Classification as well as for Regression problems**, but mainly used for Classification problems.
- Logistic regression is used to predict the **categorical dependent variable** with the help of independent variables.
- The output of Logistic Regression problem can be only between the 0 and 1.
- Logistic regression can be used where the probabilities between two classes is required. Such as whether it will rain today or not, either 0 or 1, true or false etc.
- Logistic regression is based on the concept of Maximum Likelihood estimation. According to this estimation, the observed data should be most probable.
- In logistic regression, we pass the weighted sum of inputs through an activation function that can map values in between 0 and 1. Such activation function is known as **sigmoid function** and the curve obtained is called as **sigmoid curve or S-curve**. Consider the below image:



The equation for logistic regression is

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

In statistics, there are three basic types of logistic regression:

- **Binary logistic regression** -- useful for predicting the relationship between a binary dependent variable (Y) and an independent variable (X).
- **Multinomial logistic regression** -- useful for making predictions when the dependent variable has two or more discrete outcomes and the order of the outcomes doesn't matter.
- **Ordinal logistic regression** -- useful for making predictions when the dependent variable has more than two discrete outcomes and the order of the outcomes has some significance.

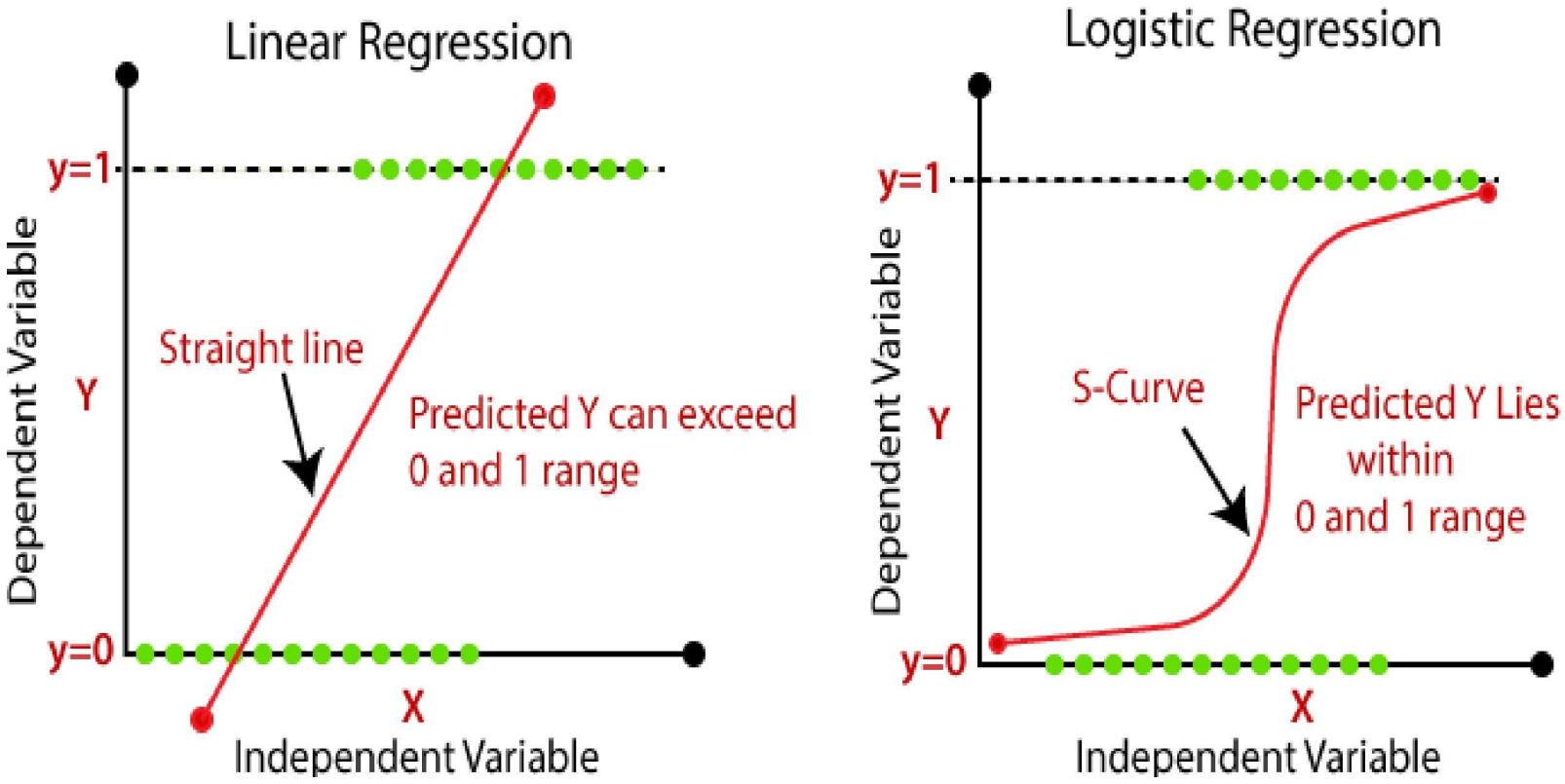
Why do we use Logistic Regression rather than Linear Regression?

- If you have this doubt, then you're in the right place,
- After reading the definition of logistic regression we now know that it is only used when our dependent variable is binary and in linear regression this dependent variable is continuous.
- The second problem is that if we add an outlier in our dataset, the best fit line in linear regression shifts to fit that point.

Linear Regression vs Logistic Regression

- Linear Regression and Logistic Regression are the two famous Machine Learning Algorithms which come under **supervised learning technique**.
- Since both the algorithms are of supervised in nature hence these algorithms use labeled dataset to make the predictions. But the main difference between them is how they are being used.
- The **Linear Regression** is used for solving **Regression problems** whereas **Logistic Regression** is used for solving the **Classification problems**.
- **Supervised Learning Technique**
- **It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately.**
- **A supervised learning algorithm takes a known set of input data (the learning set) and known responses to the data (the output), and forms a model to generate reasonable predictions for the response to the new input data. Use supervised learning if you have existing data for the output you are trying to predict.**

The description of both the algorithms is given below along with difference table.



Classification Types

- When the categorical data shows only 2 possibilities, then it is called '**Binary Classification'** For example,
  - Will the customer buy life insurance? 1. Yes, 2. No.
  - What is the gender of the person? 1. Male, 2. Female.
- When we have more than 2 possibilities, then it is called '**Multiclass Classification**'. In this the number of possibilities can be 3, 4 or even more. For example,
  - Which party a person is going to vote for? 1. BRS, 2. BJP, 3. Congress
  - What is this written digit? May be any digit from 0 to 9. Hence 10 possibilities.

Binary Classification using Logistic Regression

- In Binary classification, we have to predict one of the two classes or groups of data.
- For example, we are given age of the person and whether he bought insurance or not. Depending on age, we have to predict the person will buy insurance or not. That means the variable is : buying insurance and its possibilities are : 'Yes' or 'No'.
- **Data set given: insurance_data.csv**
- This file contains 'age' and 'bought_insurance'.
- The independent column is '**age**' and the dependent column is '**bought_insurance**'.
- This target column contains only two values 0 and 1.
- Here, 0 represents 'No' (did not buy insurance) and 1 represents 'Yes' (bought insurance).
- Hence predicting this column value comes under Binary classification.
- Please see the first few line of this data set in figure 26.1:

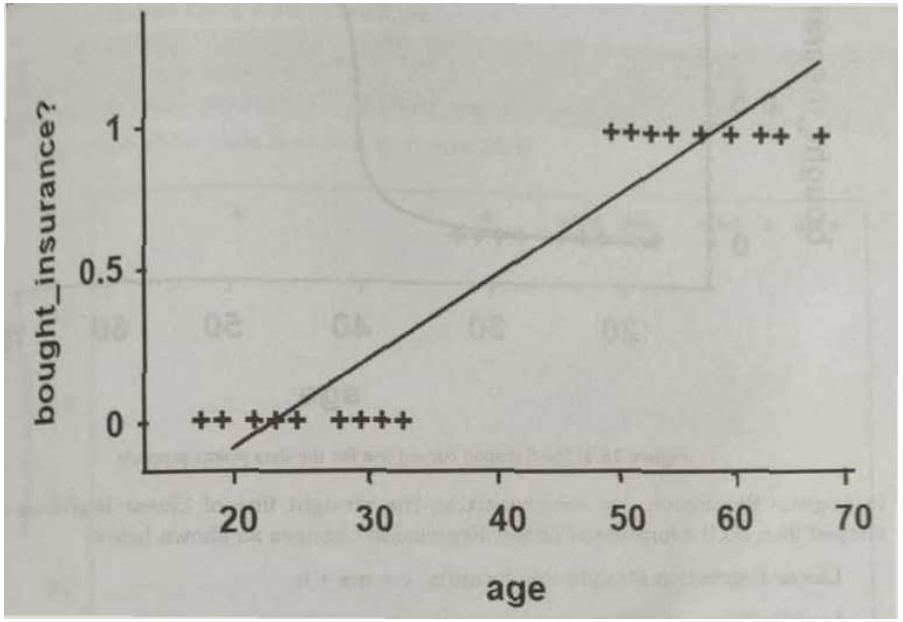
	A	B
1	age	bought_insurance
2	22	0
3	25	0
4	47	1
5	52	0
6	46	1
7	56	1
8	55	0
9	60	1
10	62	1
11	61	1
12	18	0
13	28	0
14	27	0
15	29	0
16	49	1

Figure 26.1: The *insurance data set*

❖ When we draw a line graph showing relationship between ‘age’ and ‘bought_insurance’, we can except the graph as shown in Figure 26.2.

❖ Since ‘bought_insurance’ has only two values 0 or 1, we can expect the points either at bottom(at 0) or at top(at 1) as shown in the Figure.

❖ When we connect the points using a straight line as done by the Linear Regression Model, the points will not fit on the line properly. Mostly, the points will be out of the line.

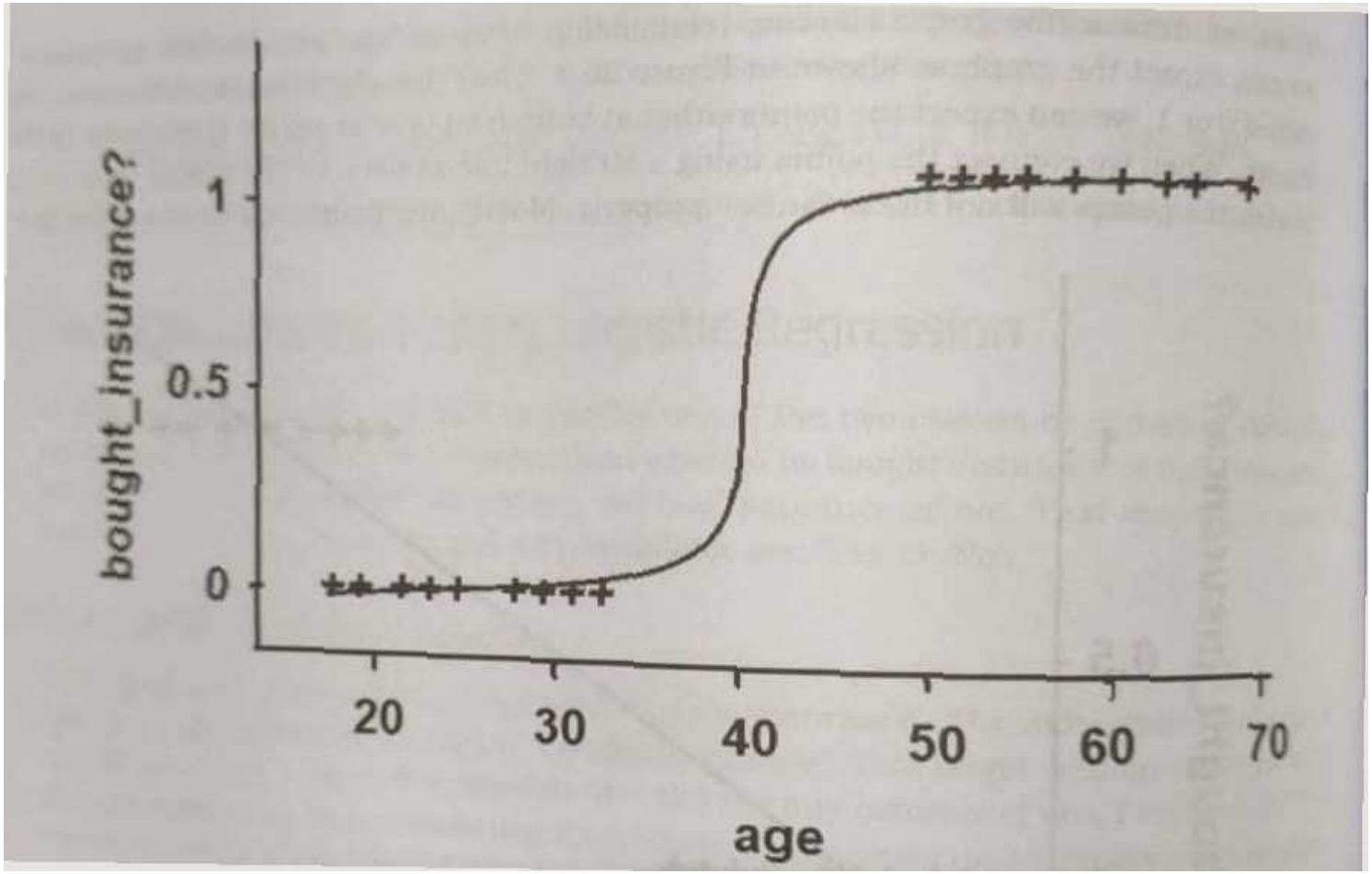


- To connect such types of points, we can use an ‘S’ shaped curve.
- This curved line fits most of the data points and becomes best fit as shown in Figure 26.3. To draw ‘S’ shaped smooth curve, we can use **Sigmoid** or **Logit Function** which is given by the formula:

$$\text{sigmoid}(t) = \frac{1}{1 + e^{-t}}$$

- Where **t** indicates Euler’s number that nearly equals to **2.71828**.
- If we observe the above formula, we can see that it is 1 divided by some quantity. Hence its maximum value can reach up to 1 only.
- So, the purpose of sigmoid function is to convert the input into a range from 0 to 1. It shows a smooth curve when plotted.

Figure 26.3 : The S shaped curved line fits the data points properly



- In Logistic Regression , we are converting the straight line of Linear Regression into **s shaped line**. So the formula of Linear Regression changes as shown below:

  Linear regression straight line formula: y=mx+b

  Linear regression s shaped curved line formula: y=1/(1+e^-(mx+b))

- Now let us develop a machine Learning program to apply Logistic Regression on insurance data set and predict whether a person will buy insurance or not depending on the age of the person.
- Since the data set has only 2 columns, we should take the 0th column(i.e. age) as the independent variable x, as:

  x= df.iloc[:, :-1].values

- Here, observe -1 in the columns. It represents all columns except the last column. Since there are only 2 columns, this represents the 0th column. The '.values' property will convert the column into an array. We get a 2D array finally. Why we get a 2D array? This is because we gave an expression to iloc[] suspects that there is possibility of many columns and hence it returns the data frame as a table that is converted by '.values' into a 2D array.

- Let us now retrieve the dependent variable y values (i.e. the 1st column) as:

  y = df. iloc[:, 1].values

- This gives us the 1st column values in the form of 1D array. Why we get a 1D array.
- Why we get a 1D array here? Hence we clearly mentioned to pick only one column to the iloc[], it returns a Series object with that column alone.
- This is converted by '.values' into a 1D array as there is only one column.
- To know how the data points are aligned, we can display a scatter plot that represents the data points scattered along x and y axes, as :

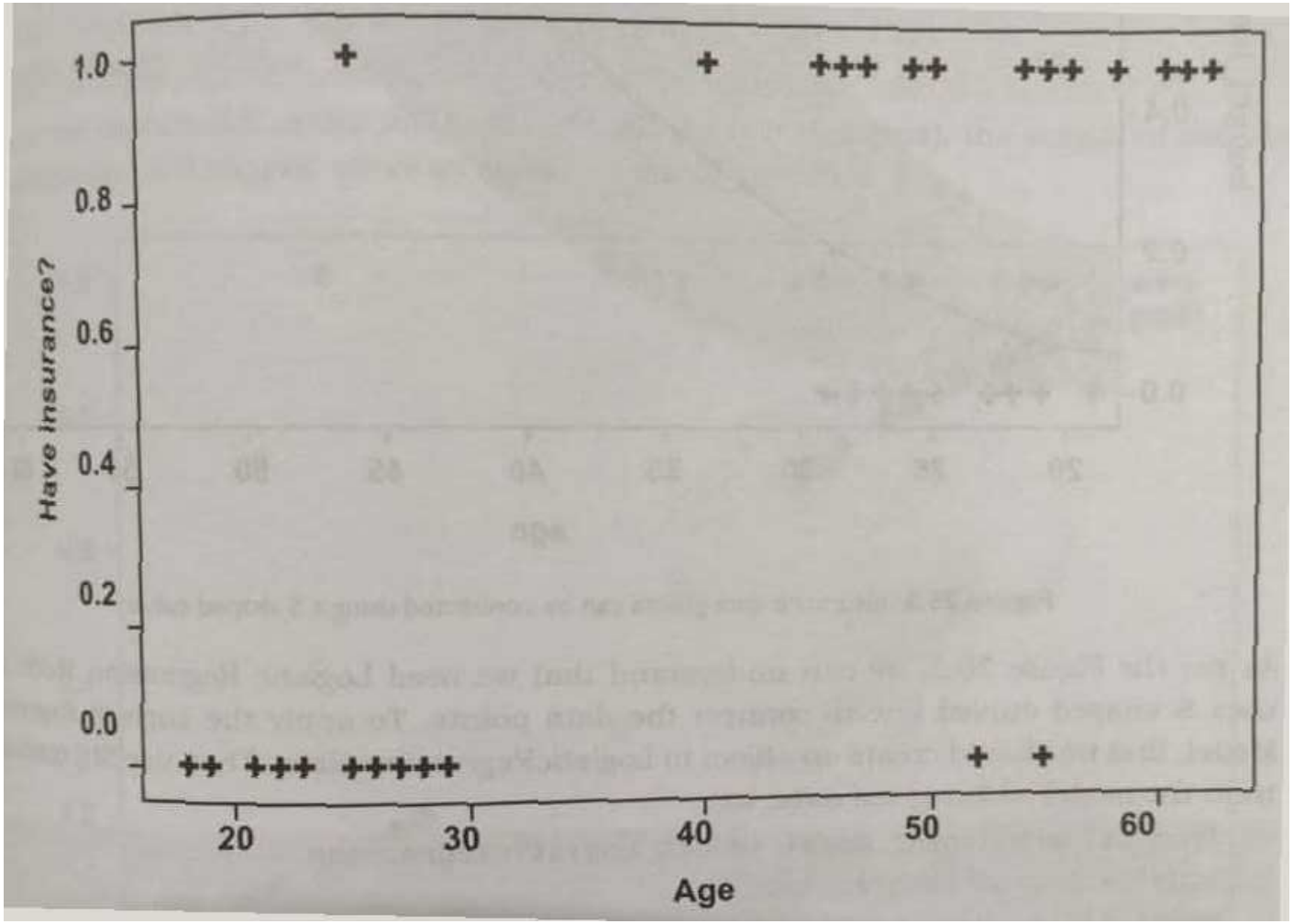
  plt. xlabel('Age')

  plt.ylabel('Have insurance?')

  plt.scatter(x, y, marker= '+', color='red')

- This is the result of the above code is shown in Figure 26.4:

Figure 26.4: The data points from insurance data

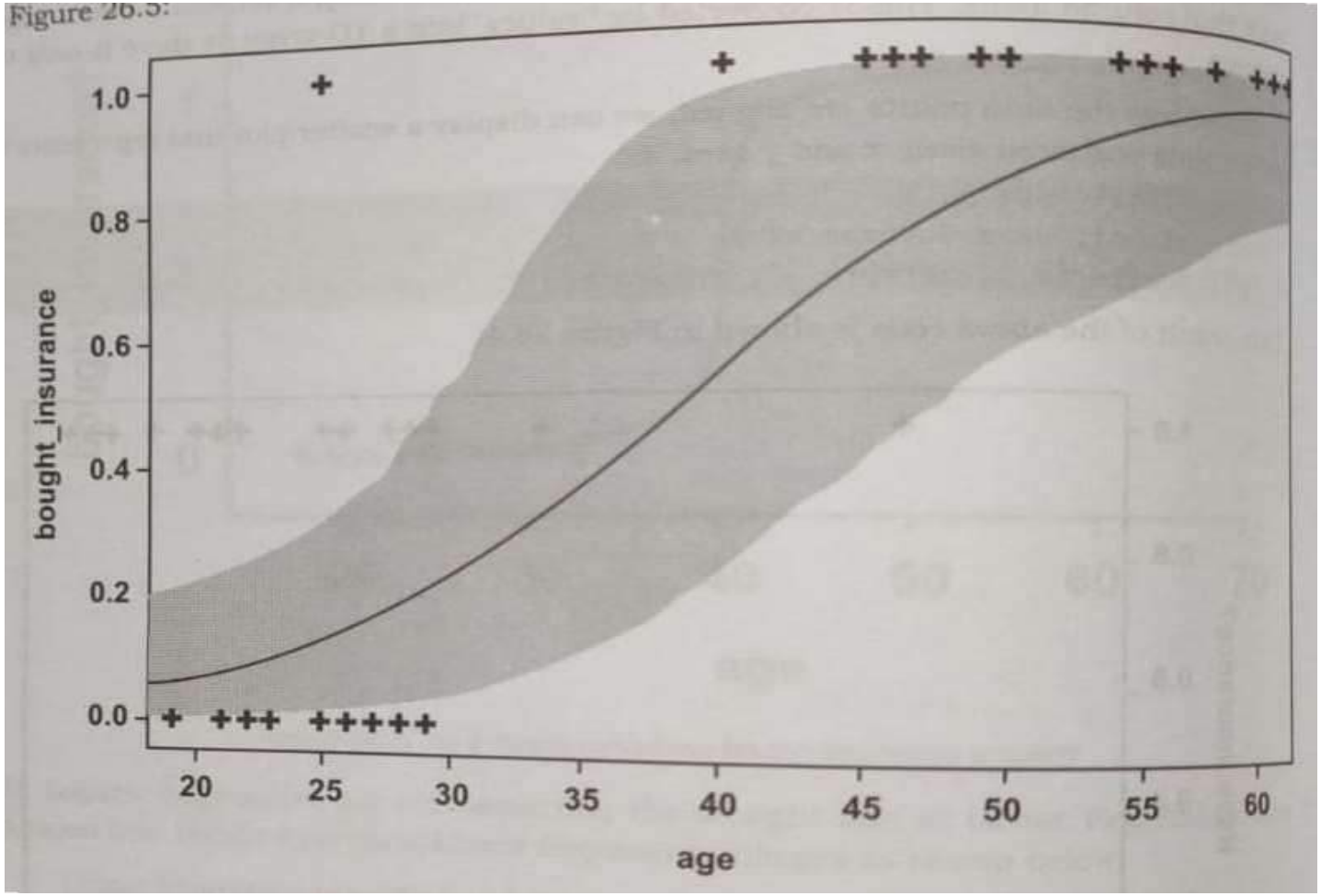


- From the Figure 26.4, we can understand that we cannot use a straight line to properly fit all these data points. That means linear Regression Model cannot be used on this data.
- Let us try now a regression line to fit this data. This can be done using regplot() function available in seaborn package.

```
import seaborn as sns
sns.regplot(x='age', y= 'bought_insurance', data=df , logistic=True,
                                                marker='+',      color='red')
```

- The above code will produce the S shaped regression line to fit the data as shown Figure 26.5:

Figure 26.5: Insurance data points can be connected using a S shaped curve



- As per the Figure 26.5, we can understand that we need Logistic Regression Model uses S shaped curved line to connect the data points. To apply the Logistic Regression Model, first we should create an object to Logistic Regression class. Then use fit() method to train the model on the given data, as:

```
from sklearn. linear_model import LogisticRegression
model= LogisticRegression()
model.fit(x,y)
```

- While creating LogisticRegression object, we can pass the parameter 'solver' values as
- **solver= 'liblinear'-** useful for small datasets and for binary classification. Applies L1 and L2 penalities.
- **solver='newton-cg', 'lbfgs', 'sag'** - useful for multiclass classification. Applies only L2 penality. Default solver 'lbfgs'.
- **solver= 'saga'-** useful for multiclass classification. Applies L1 and L2 penality. For example

model=LogisticRegression(solver='liblinear')

- Once the model is trained, we can use predict() method to predict the class (either 0 or 1). But there is another method by the name **predict_proba()** that gives an array of probabilities regarding **how far the data point comes under the class 0 or 1**.
- See how to use this method.

  y1= model.predict_proba(x)[:,1]
- In the above statement, predict_proba(x) gives a 2D array with 2 columns.
- The **0th column indicates** the probability of how far the data point belongs to class 0 and the **1st column indicates** the probability that it belongs to class 1.
- For drawing the points in the form of a regression plot, we should take class 1 type of points. Hence we used [:, 1] in the above statement. Now let us draw the scatter plots with original data and with probabilities as:

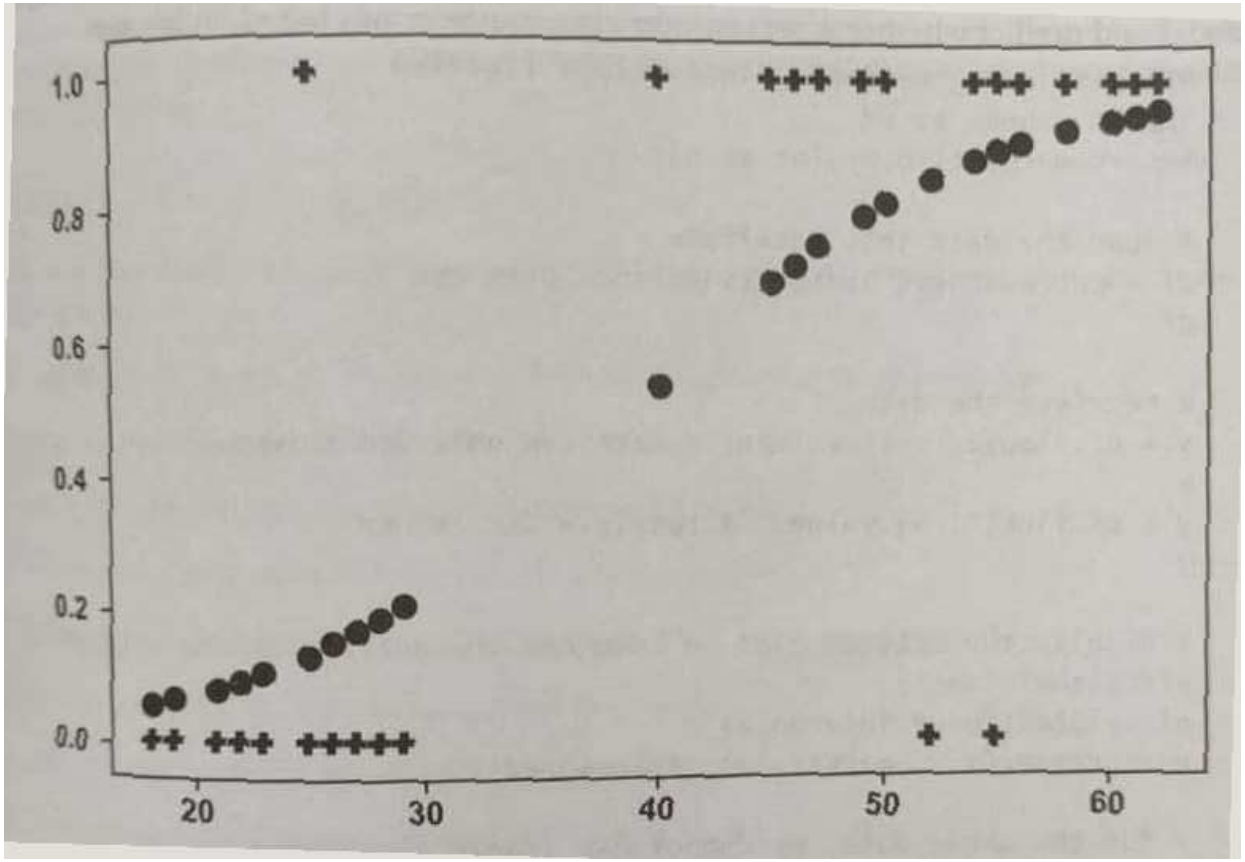
  plt.scatter(x,y, marker='+', color='red') # original data

  plt.scatter (x, y1. color='blue') # probabilities data

  plt.show () # show both the plots

❖ We execute the above block of code (all the 3 statements), the output of the above code displays a S shaped curve as shown in the Figure 26.6:

❖ Figure 26.6: The Logistic Regression line in the form of S curve



- The predictions or output given by Logistic Regression model will be in the form of 0 or 1. Here 0 represents ‘No' and 1 indicates ‘Yes’. To apply the model on new data, we call predict() method and pass the data as a 2D array. For example, to know a person with 56 years age will buy insurance or not, we can call predict() method as:

  model.predict([[56]])

  Output:

  array([1])

- Observe the output. It is showing 1. This represents ‘Yes'. That means there are high chances that a 56 years aged person will purchase insurance policy. Similarly , we want to know a person with 36 years age will purchase insurance or not. Let us call predict() method as

  model.predict ([[36]])

  Output:

  array([0])

- Since the output is a 0, we can understand that the 36 years aged person may not purchase insurance policy.

Program 1:

We are given data like age and bought_insurance.

Apply Logistic Regression Model and predict whether a person takes insurance or not based on his age.

#1ogistic regression - binary classification

```
import pandas as pd
import matplotlib.pyplot as plt
# load the data into dataframe
df = pd.read_csv("C:\Exp\LR\insurance_data.csv")
df
```

In [7]:
```
df = pd.read_csv("C:\Exp\LR\insurance_data.csv")
df
```

Out[7]:

	age	bought_insurance
0	22	0
1	25	0
2	47	1
3	52	0
4	46	1
5	56	1
6	55	0
7	60	1
8	62	1
9	61	1
10	18	0
11	28	0
12	27	0

```
# retrieve the data
x=df.iloc[:, :-1].values #retrieve only 0th column
x
y=df.iloc[:,1].values  #retrieve the 1st column
y
```

```
In [8]: # retrieve the data
        x=df.iloc[:, :-1].values #retrieve only 0th column
        x

Out[8]: array([[22],
               [25],
               [47],
               [52],
               [46],
               [56],
               [55],
               [60],
               [62],
               [61],
               [18],
               [28],
               [27],
               [29],
               [49],
               [55],
               [25],
               [58],
               [19],
               [18],
               [21],
               [26],
               [40],
               [45],
               [50],
               [54],
               [23]], dtype=int64)
```

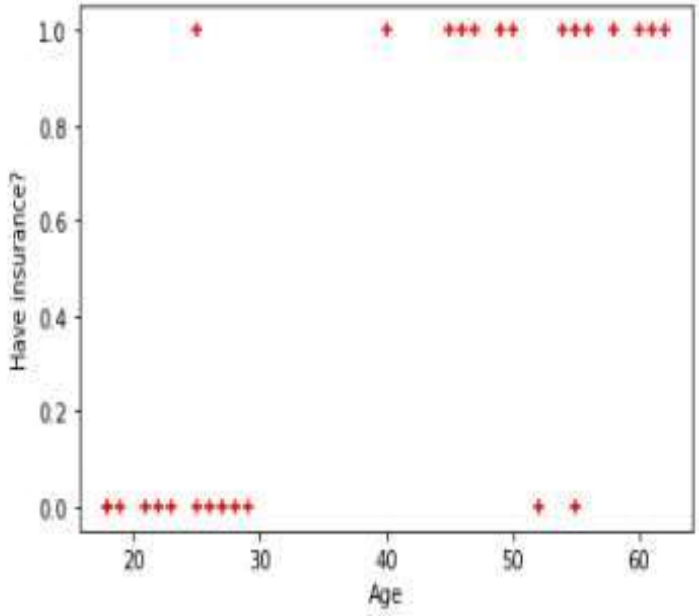
```
In [11]: y=df.iloc[:,1].values  #retrieve the 1st column
         y

Out[11]: array([0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,
                1, 1, 1, 1, 0], dtype=int64)
```

# display the scatter plot to know how the datapoints are aligned
plt.xlabel('Age')
plt.ylabel('Have insurance?')
plt.scatter(x,y, marker='+', color='red')

```
In [15]: # display the scatter plot to know how the datapoints are aligned
         plt.xlabel('Age')
         plt.ylabel('Have insurance?')
         plt.scatter(x,y, marker='+', color='red')

Out[15]: <matplotlib.collections.PathCollection at 0x1cd179cc730>
```



for the above data, we cannot use linear regression

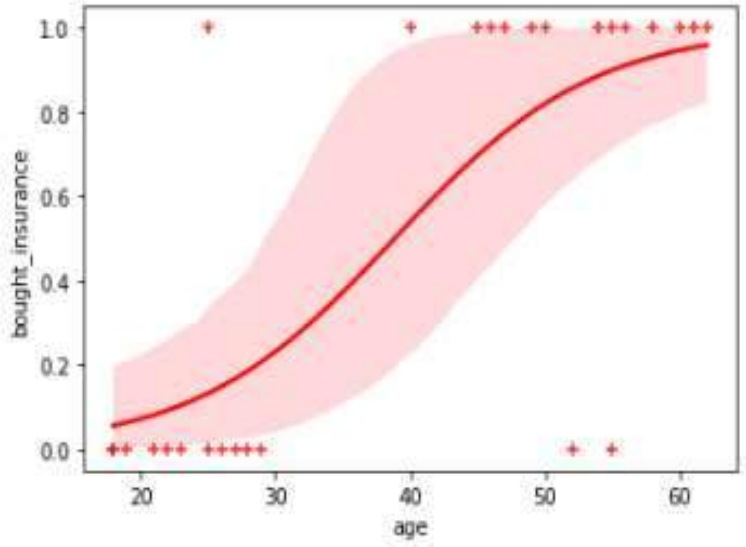
#show the data as logistic regression plot

import seaborn as sns

sns.regplot(x='age', y='bought_insurance', data=df, logistic=True, marker='+', color='red')

```
In [19]: # for the above data, we cannot use linear regression
         #show the data as logistic regression plot
         import seaborn as sns
         sns.regplot(x='age', y='bought_insurance', data=df, logistic=True, marker='+', color='red')

Out[19]: <AxesSubplot:xlabel='age', ylabel='bought_insurance'>
```



```
# create logistic regression model
from sklearn. linear_model import LogisticRegression
model = LogisticRegression()
# train the model
model.fit(x,y)
```

Output:

```
LogisticRegression()
```

```
In [23]: # find accuracy of the model - gives 88.8%
         model. score (x, y)

Out[23]: 0.8888888888888888
```

predict if 56 years aged person will buy insurance or not
model.predict([[56]]) # array([1]) yes

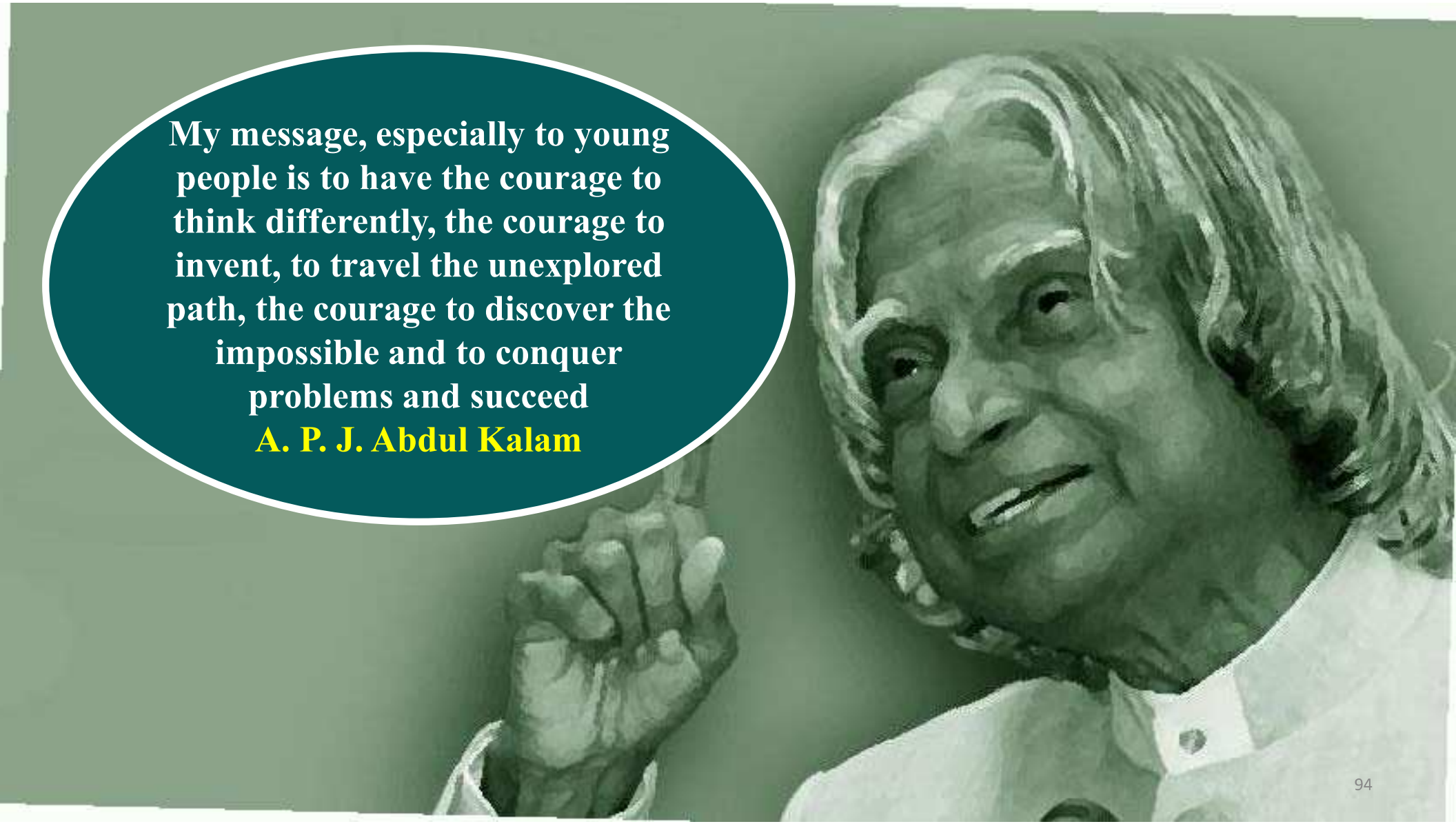
```
In [25]: # predict if 56 years aged person will buy insurance or not
         model.predict([[56]])  # array([1])       yes

Out[25]: array([1], dtype=int64)
```

predict if 36 years aged person will buy insurance or not
model.predict([[36]]) #array([O]) No

```
In [26]: # predict  if 36 years aged person will buy insurance or not
         model.predict([[36]])   #array([O])   No

Out[26]: array([0], dtype=int64)
```


My message, especially to young people is to have the courage to think differently, the courage to invent, to travel the unexplored path, the courage to discover the impossible and to conquer problems and succeed
A. P. J. Abdul Kalam


Thank
You!