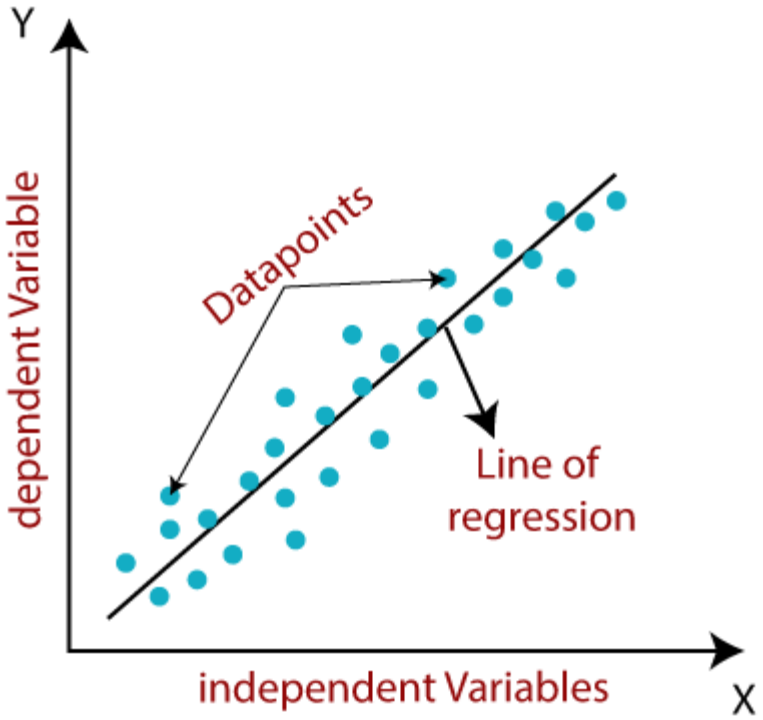
Supervised Learning

Unit 4

Supervised Learning

- Most popular Machine Learning algorithms.
- 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.
- Shows a linear relationship between a dependent (y) and one or more independent (x) 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:



Types of Linear Regression

- **Simple Linear Regression:** A single independent variable is used to predict the value of a numerical dependent variable

Equation of Simple Linear Regression

$$y = b_o + b_1 x$$

where bo is the intercept, b1 is coefficient or slope, x is the independent variable and y is the dependent variable.

Types of Linear Regression

- **Multiple Linear regression:** More than one independent variable is used to predict the value of a numerical dependent variable

$$y = b_o + b_1x_1 + b_2x_2 + b_3x_3 \ldots . + b_nx_n$$

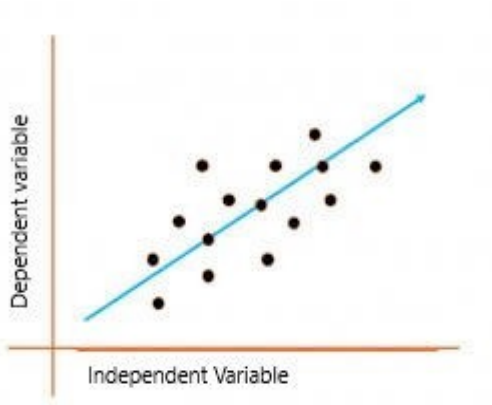
- Where bo is the intercept, $b_1, b_2, b_3, b_4 \ldots, b_n$ are coefficients or slopes of the independent variables $x_1, x_2, x_3, x_4 \ldots, x_n$ and y is the dependent variable.

Simple Linear Regression

- There is one independent variable and one dependent variable.
- The model estimates the slope and intercept of the line of best fit, which represents the relationship between the variables.
- The ***slope*** represents the change in the dependent variable for each unit change in the independent variable
- the ***intercept*** represents the predicted value of the dependent variable when the independent variable is zero.
- It shows the linear relationship between the independent(predictor) variable i.e. X-axis and the dependent(output) variable i.e. Y-axis, called linear regression.
- If there is a single input variable **X**(independent variable), such linear regression is called ***simple linear regression***.

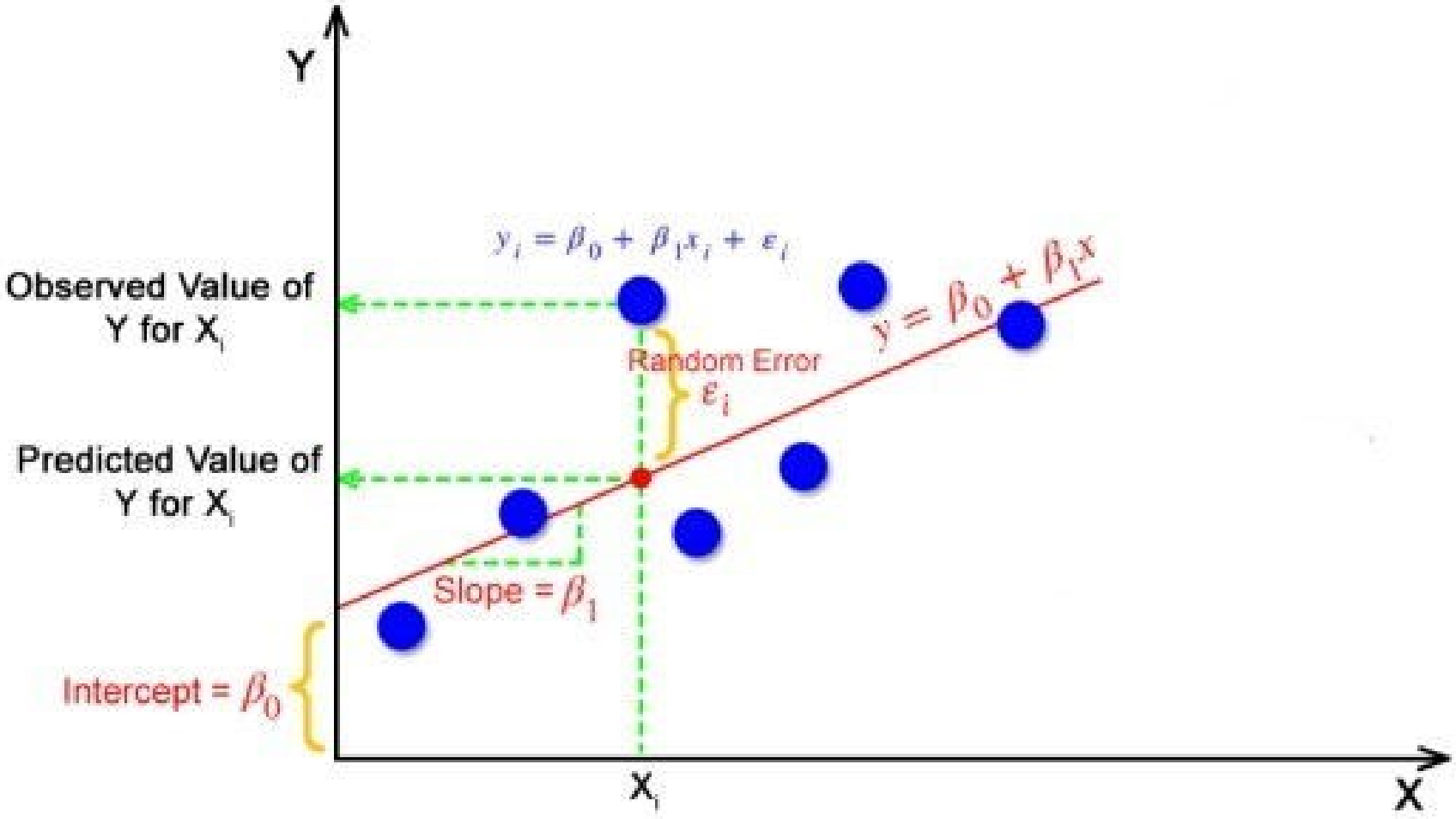
Simple Linear Regression

- The graph presents the linear relationship between the output(y) variable and predictor(X) variables.
- The blue line is referred to as the *best fit* straight line. Based on the given data points, we attempt to plot a line that fits the points the best.



$$Y_i = \beta_0 + \beta_1 X_i$$

- where Y_i = Dependent variable, β_0 = constant/Intercept, β_1 = Slope/Intercept, X_i = Independent variable.


Y
$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
Observed Value of Y for X_i
Random Error ε_i
$y = \beta_0 + \beta_1 x$
Predicted Value of Y for X_i
Slope = β_1
Intercept = β_0
X_i
X

Simple Linear Regression

But how the linear regression finds out which is the best fit line?

- The goal of the linear regression algorithm is to get the **best values for B_0 and B_1** to find the best fit line.
- The best fit line is a line that has the least error which means the error between predicted values and actual values should be minimum.

Random Error(Residuals)

- In regression, the difference between the observed value of the dependent variable(y_i) and the predicted value(**predicted**) is called the residuals.

$$\varepsilon_i = y_{predicted} - y_i$$

where $y_{predicted} = B_0 + B_1 X_i$

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 (β_0, β_1) gives a different line of regression, so we need to calculate the best values for β_0, and β_1 to find the best fit line

Cost function

- The cost function helps to work out the optimal values for B_0 and B_1, which provides the best fit line for the data points.
- In Linear Regression, generally **Mean Squared Error (MSE)** cost function is used

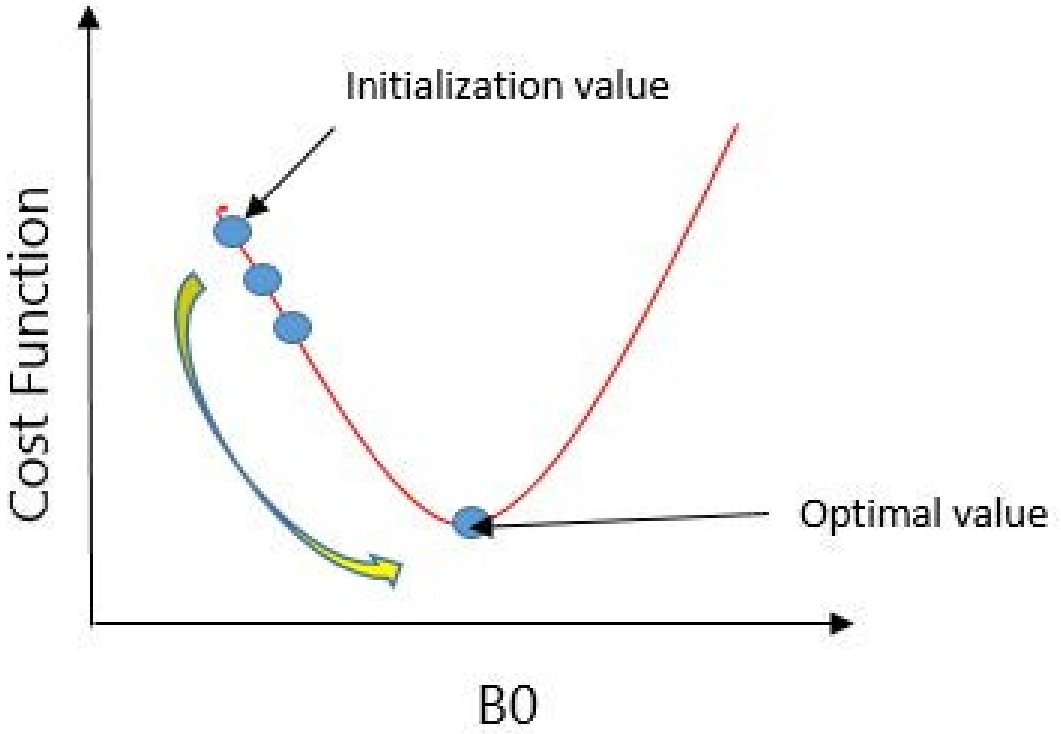
$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (B1x_i + B0))^2$$

- Using the MSE function, we'll update the values of B_0 and B_1 such that the MSE value settles at the minima.
- These parameters can be determined using the gradient descent method such that the value for the cost function is minimum.

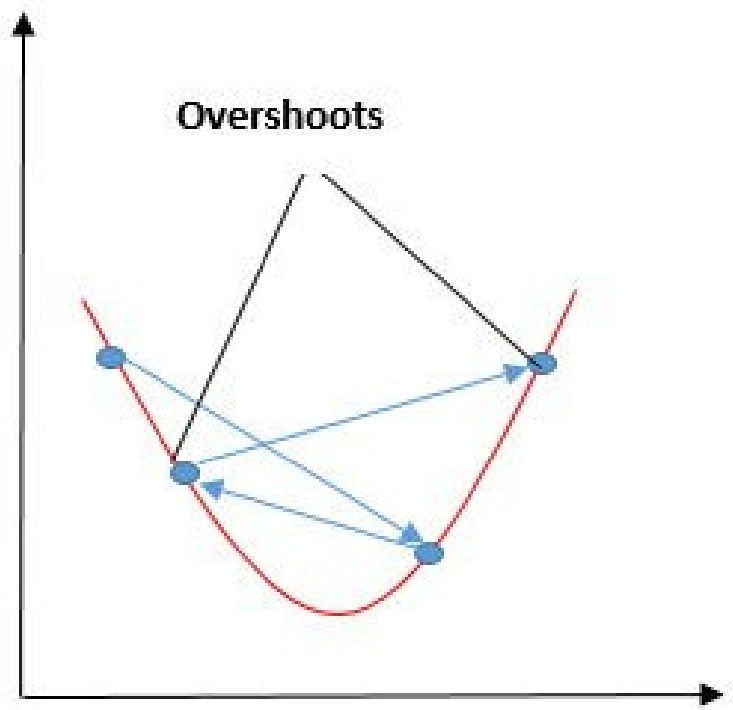
Gradient Descent for Linear Regression

- One of the optimization algorithms
- It optimize the cost function(objective function) to reach the optimal minimal solution.
- To find the optimum solution we need to reduce the cost function(MSE) for all data points.
- This is done by updating the values of B_0 and B_1 iteratively until we get an optimal solution.
- A regression model optimizes the gradient descent algorithm to update the coefficients of the line by reducing the cost function by randomly selecting coefficient values and then iteratively updating the values to reach the minimum cost function.

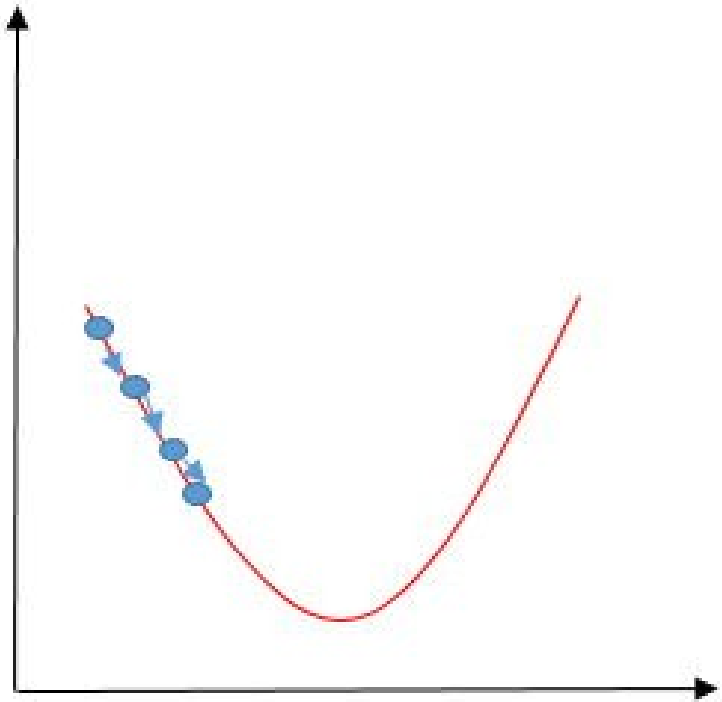
Gradient Descent for Linear Regression



Gradient Descent for Linear Regression



High learning rate



Low learning rate

Gradient Descent for Linear Regression

- To update B_0 and B_1, we take gradients from the cost function. To find these gradients, we take partial derivatives for B_0 and B_1.

$$J = \frac{1}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial B_0} = \frac{2}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i)$$

$$\frac{\partial J}{\partial B_1} = \frac{2}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i) \cdot x_i$$

Gradient Descent for Linear Regression

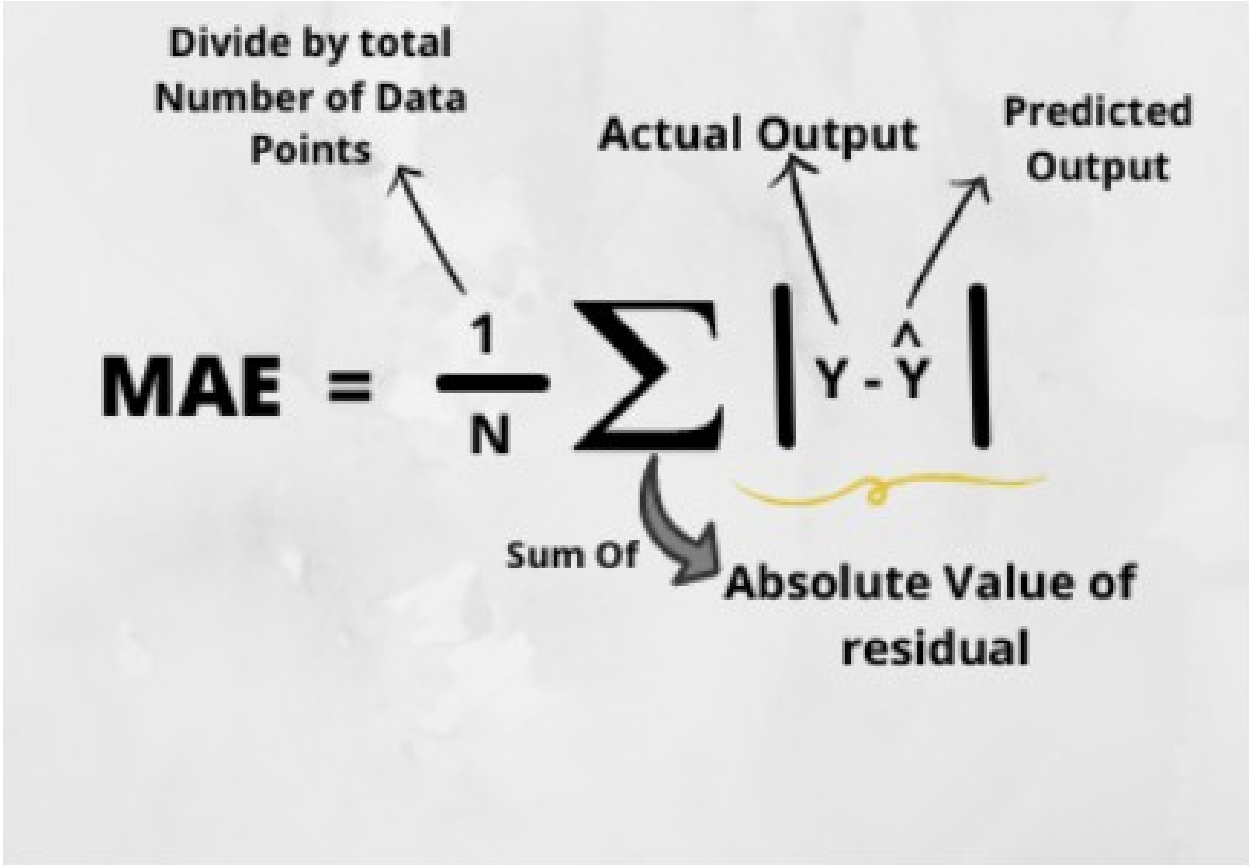
$$B_0 = B_0 - \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (pred_i - y_i)$$

$$B_1 = B_1 - \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (pred_i - y_i) \cdot x_i$$

Evaluation Metrics for Linear Regression

- Linear regression model can be assessed using various evaluation metrics.
- These evaluation metrics usually provide a measure of how well the observed outputs are being generated by the model.

Mean Absolute Error(MAE)

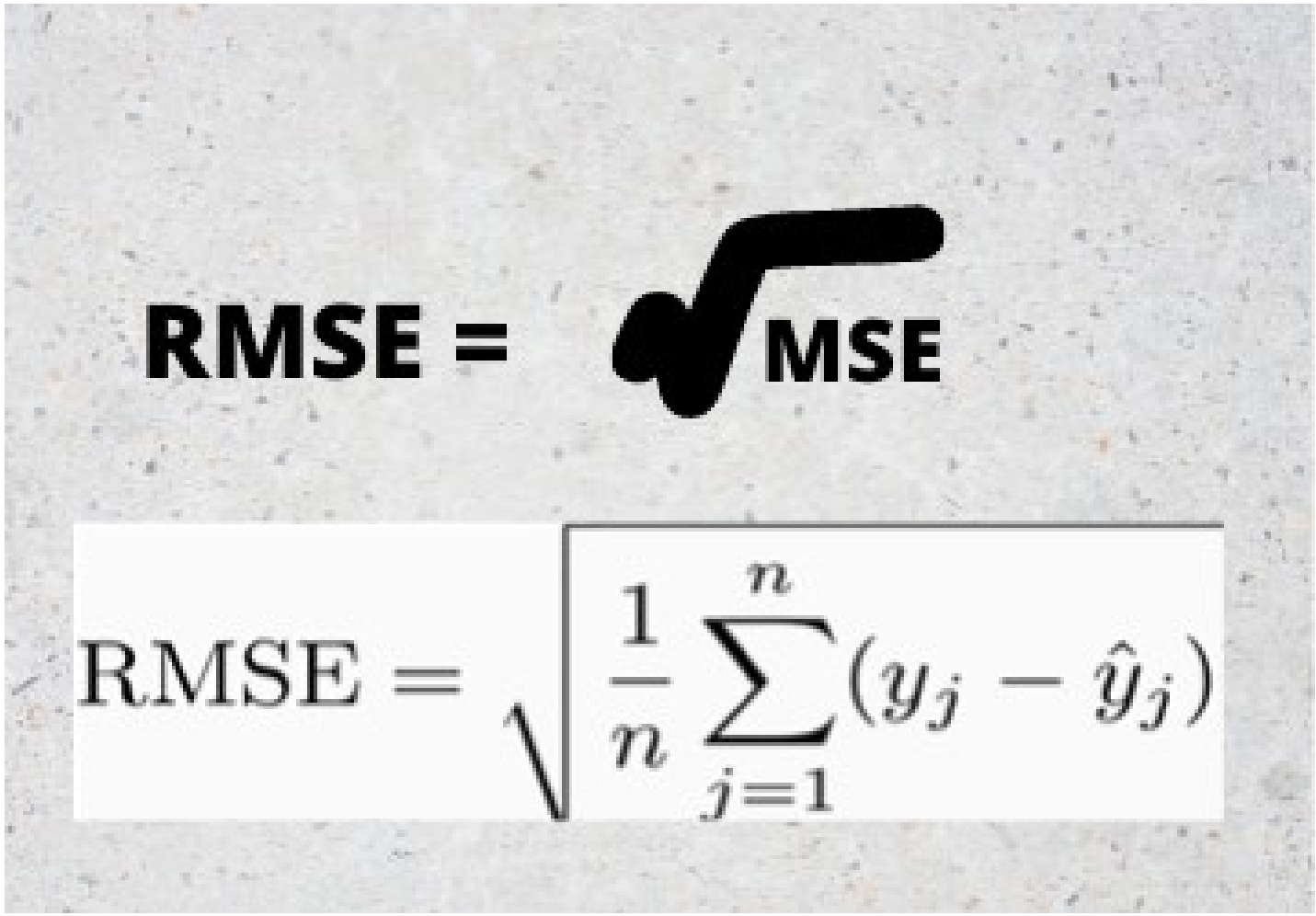


Mean Squared Error(MSE)

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)}^{2}$$

The square of the difference between actual and predicted

Root Mean Squared Error(RMSE)



Logistic Regression

Unit 4

Logistic Regression

- Most popular Machine Learning algorithms
- which comes under the Supervised Learning technique
- used for predicting the categorical dependent variable using a given set of independent variables
- predicts the output of a categorical dependent variable. outcome must be a categorical or discrete value.
- It can be either **Yes** or **No,** 0 or 1, true or False, etc. but instead of giving the exact value as **0** and **1**, **it gives the probabilistic values which lie between 0 and 1**.

Logistic Regression

- Much similar to the Linear Regression except that how they are used
- Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- It has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- It can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification

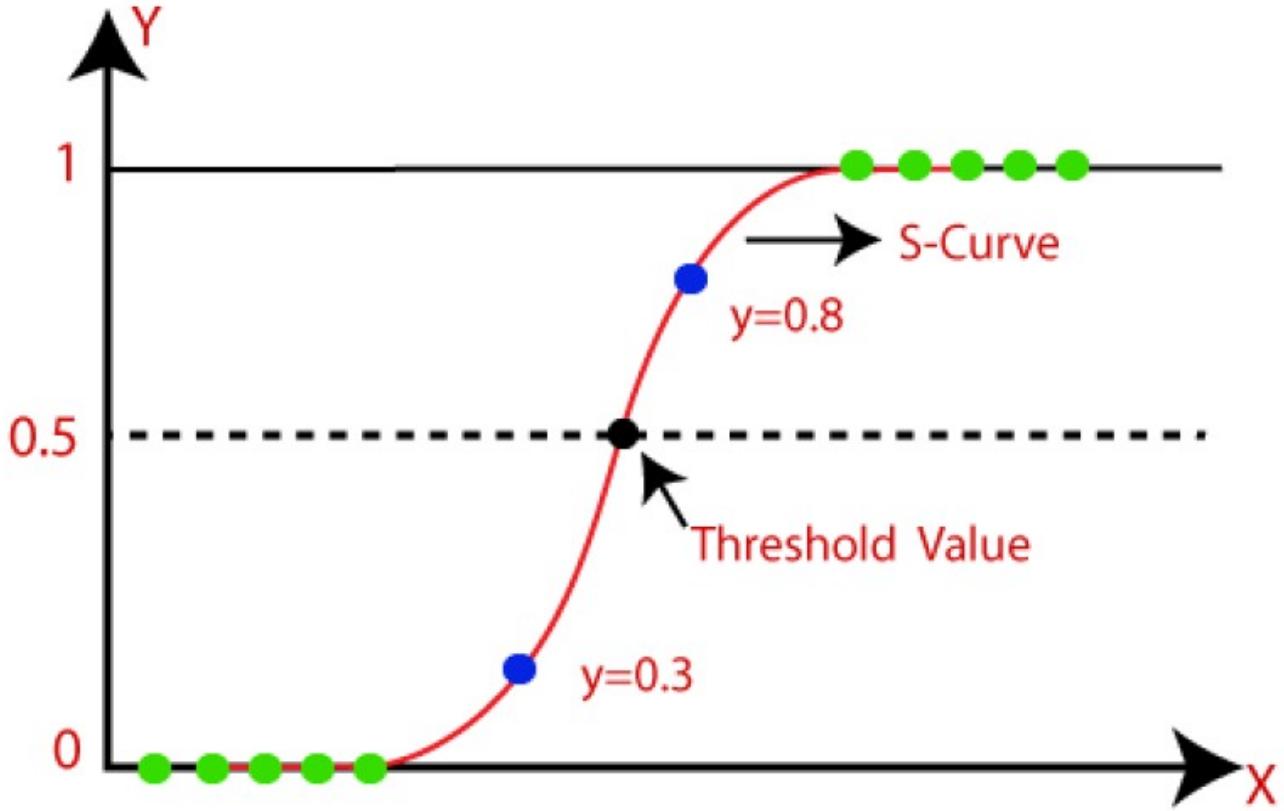
Type of Logistic Regression

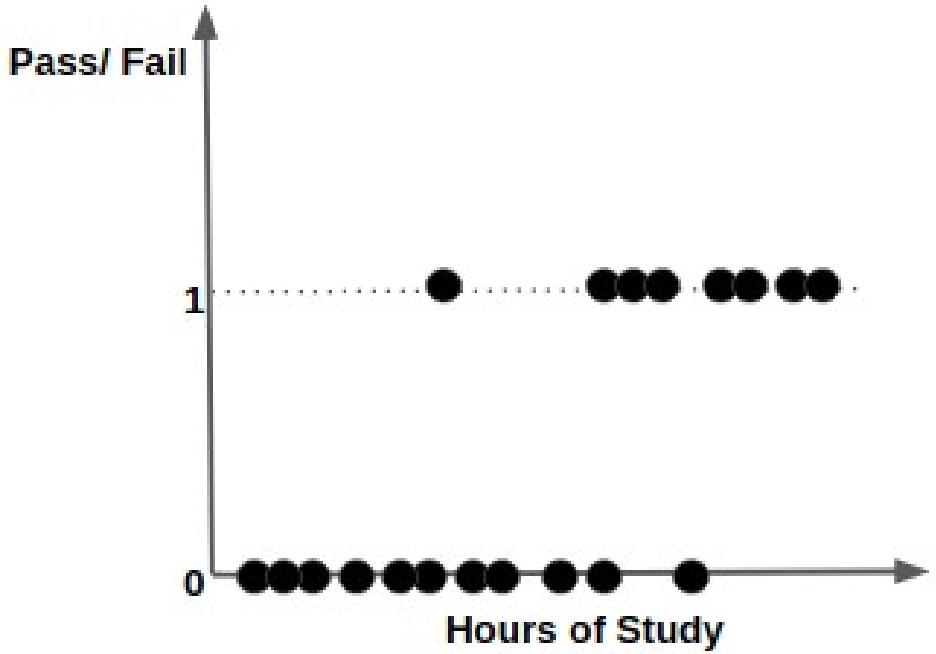
- Classified into three types

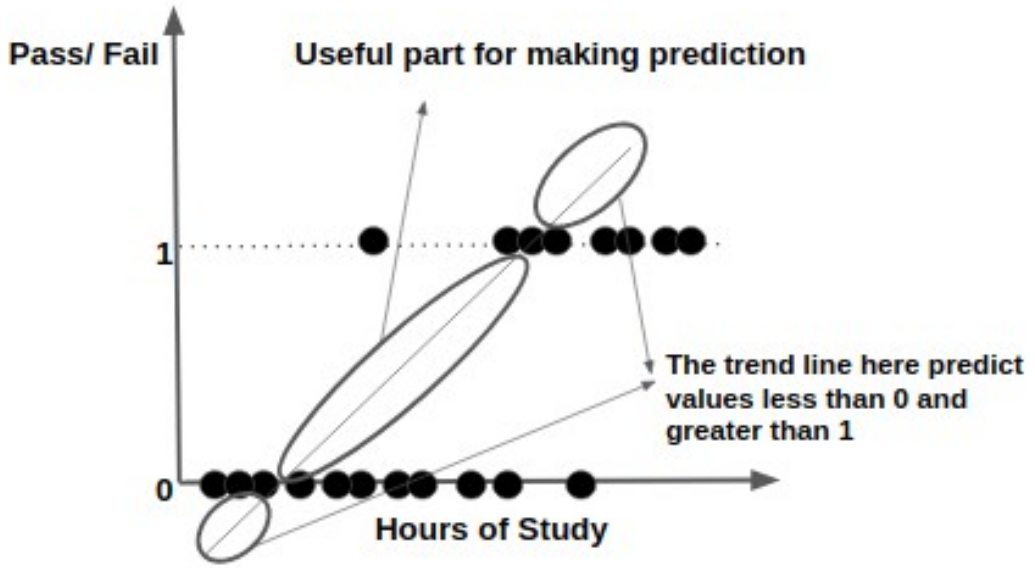
1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
3. **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

Logistic Regression

- The below image is showing the logistic function:




Pass/ Fail
1
0
Hours of Study


Pass/ Fail
Useful part for making prediction
1
0
The trend line here predict values less than 0 and greater than 1
Hours of Study

- We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the '**Sigmoid function**' or also known as the 'logistic function' instead of a linear function.
- The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

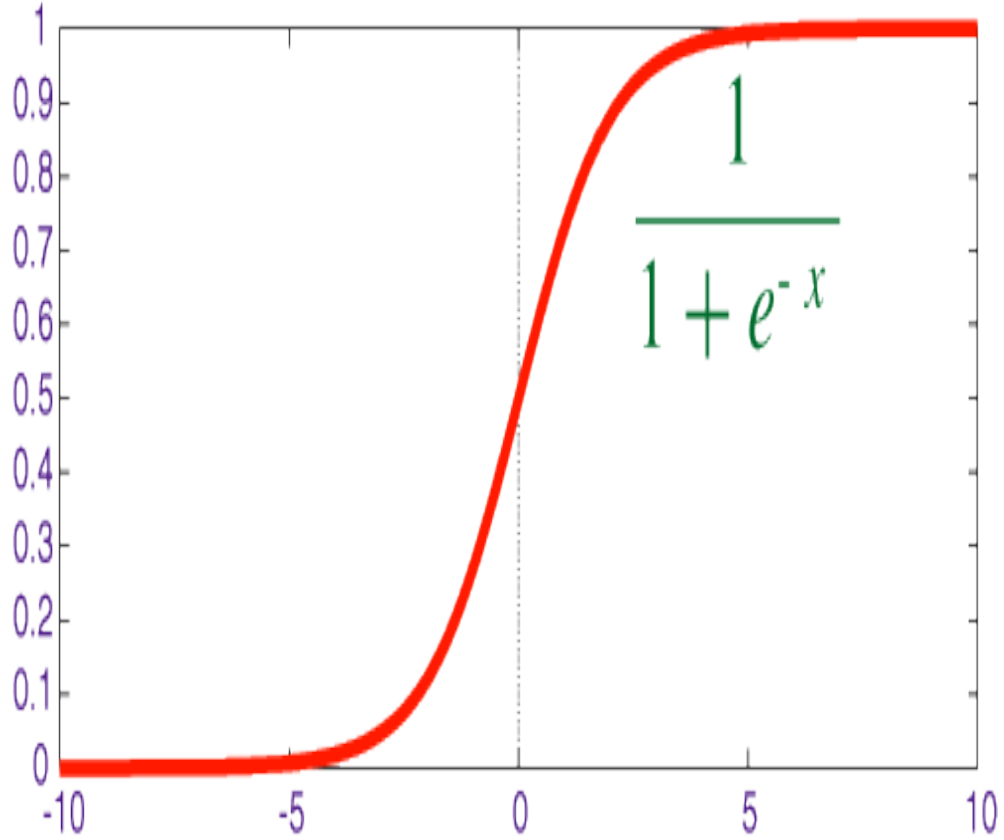
$$0 \leq h_\theta(x) \leq 1$$

Logistic regression hypothesis expectation

Logistic Function (Sigmoid Function)

- Mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0

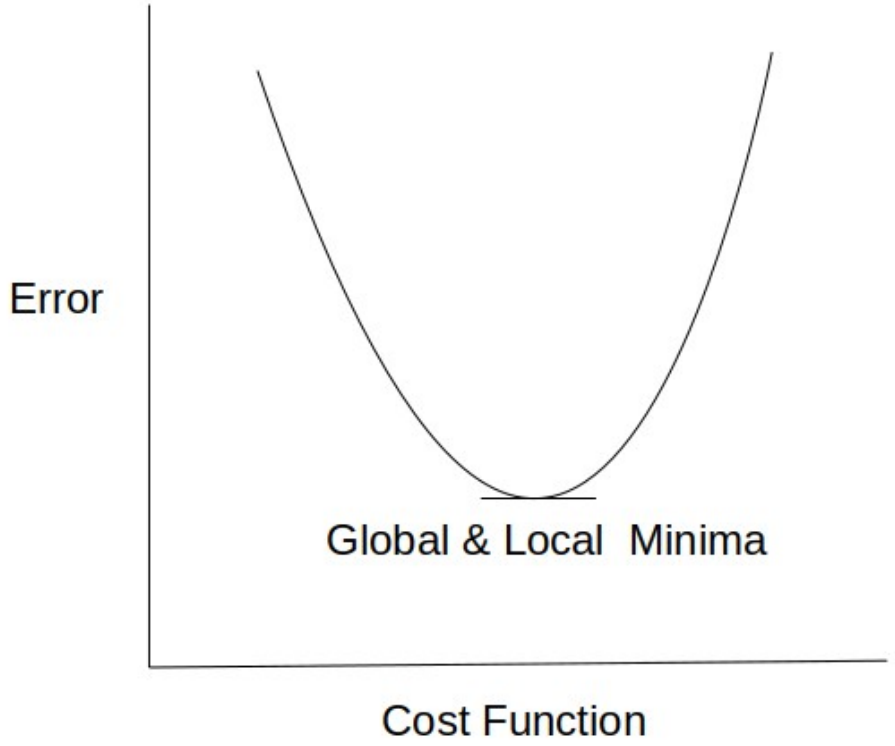
Sigmoid Function



Logistic Function

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

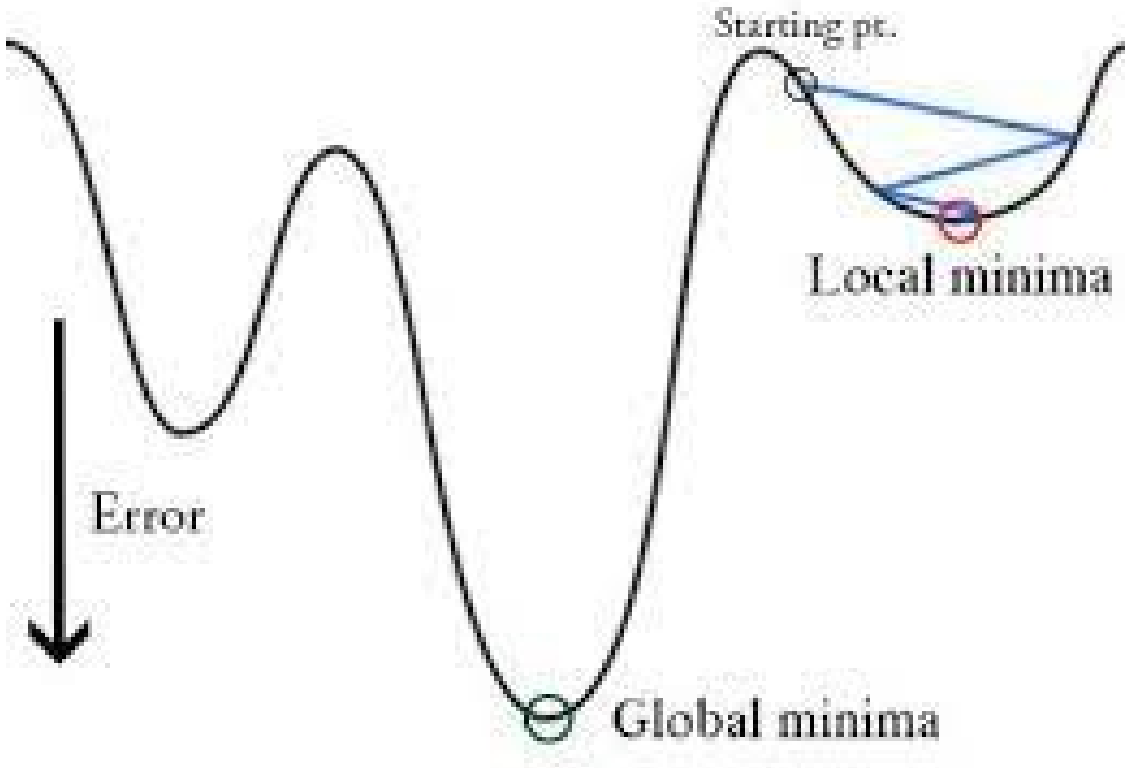
Cost Function in Logistic Regression



Linear Regression
Cost Function

$$J = \frac{\sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}{n}$$

- In logistic regression Yi is a non-linear function ($\hat{Y}=1/1+e^{-z}$). If we use this in the above MSE equation then it will give a non-convex graph with many local minima as shown



- The problem here is that this cost function will give results with local minima, which is a big problem because then we'll miss out on our global minima and our error will increase.
- In order to solve this problem, we derive a different cost function for logistic regression called ***log loss*** which is also derived from the *maximum likelihood estimation* method.

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -(y_i * \log(\hat{Y}_i) + (1 - y_i) * \log(1 - \hat{Y}_i))$$

Gradient Decent

- Now the question arises, how do we reduce the cost value.
- this can be done by using **Gradient Descent.**
- The main goal of Gradient descent is to **minimize the cost value.** i.e. min J(θ).
- Now to minimize our cost function we need to run the gradient descent function on each parameter i.e.

$$\theta j := \theta j - \alpha \frac{\partial}{\partial \theta j} J(\theta)$$

Objective: To minimize the cost function we have to run the gradient descent function on each parameter

Cost Function

- For logistic regression, the Cost function is defined as:

$$\begin{cases} -log(h\theta(x)) \text{ if } y = 1 \\ -log(1 - h\theta(x)) \text{ if } y = 0 \end{cases}$$

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Cost function of Logistic Regression

Gradient Decent

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Gradient Descent Simplified | Image: Andrew Ng Course

RANDOM FOREST

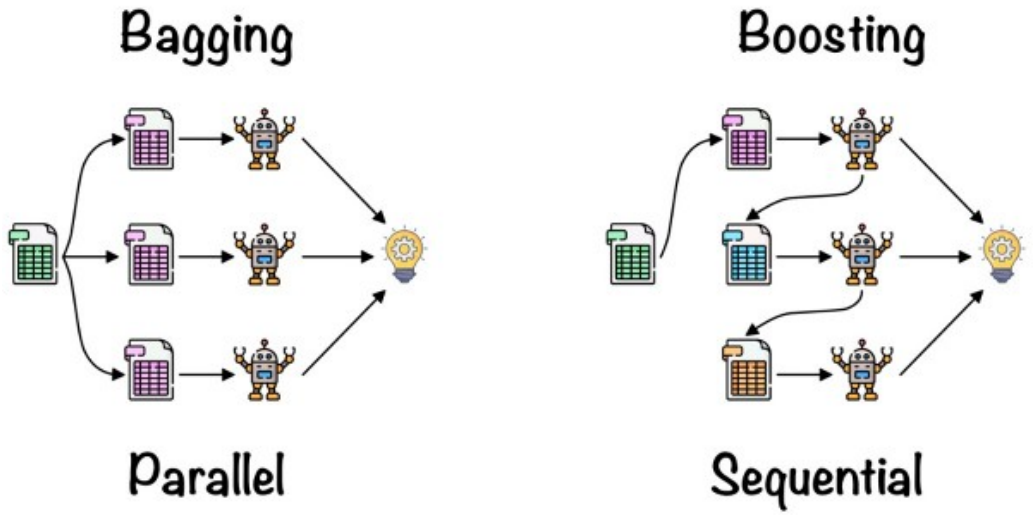
- Supervised Machine Learning Algorithm
- Used widely in Classification and Regression problems
- It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Working –

- Ensemble technique - ***Ensemble*** simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model.
- **Ensemble uses two types of methods:** Bagging and Boosting

- **Bagging -** It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

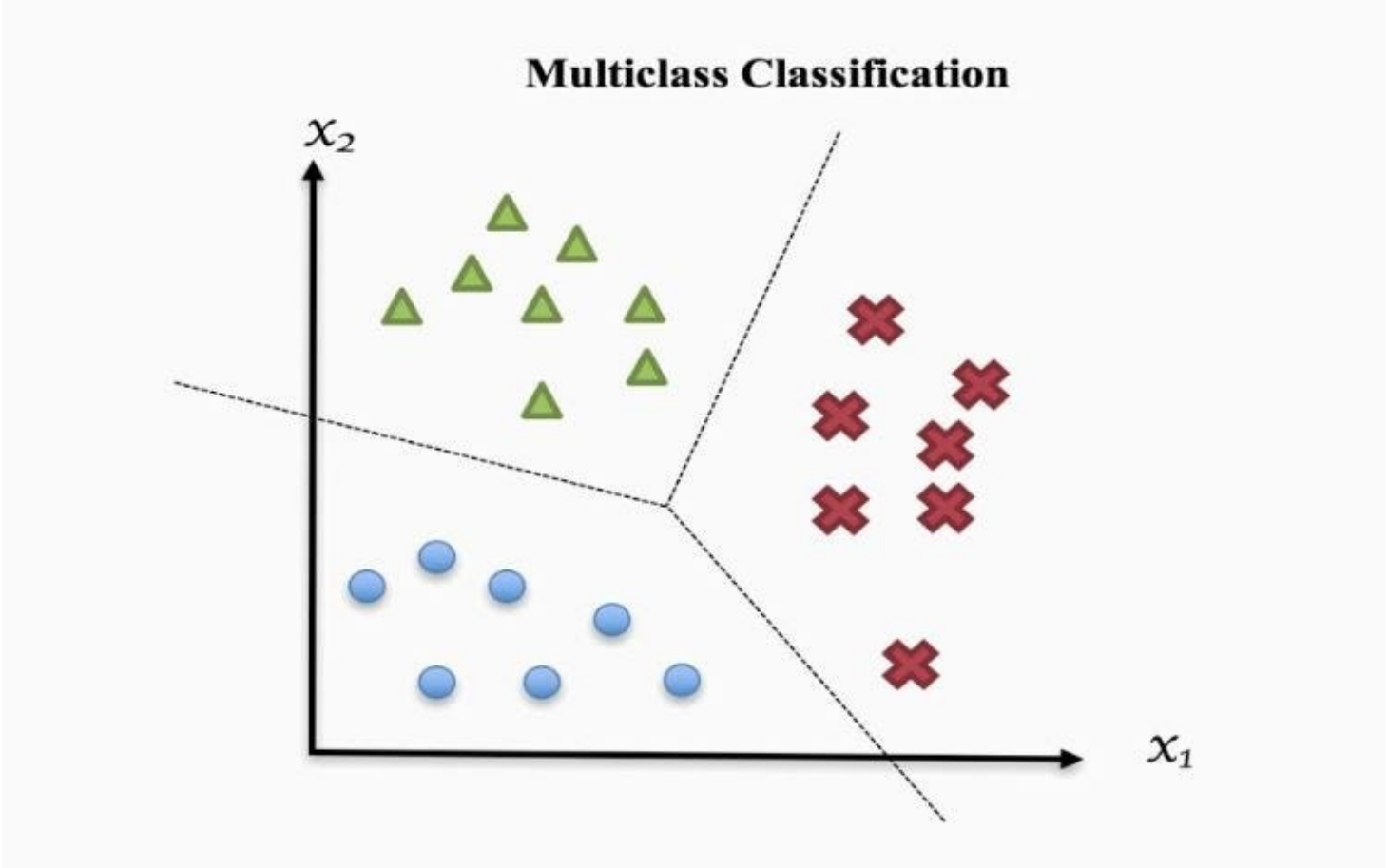
- **Boosting -** It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.



Multiclassification

- Supervised machine learning categorizes into **regression** and **classification**
- regression technique to predict the target values of continuous variables
- classification technique for predicting the class labels for given input data.
- In classification, we design the classifier model, then train it using input train data and then categorize the test data into multiple class labels present in the dataset.

Multiclassification



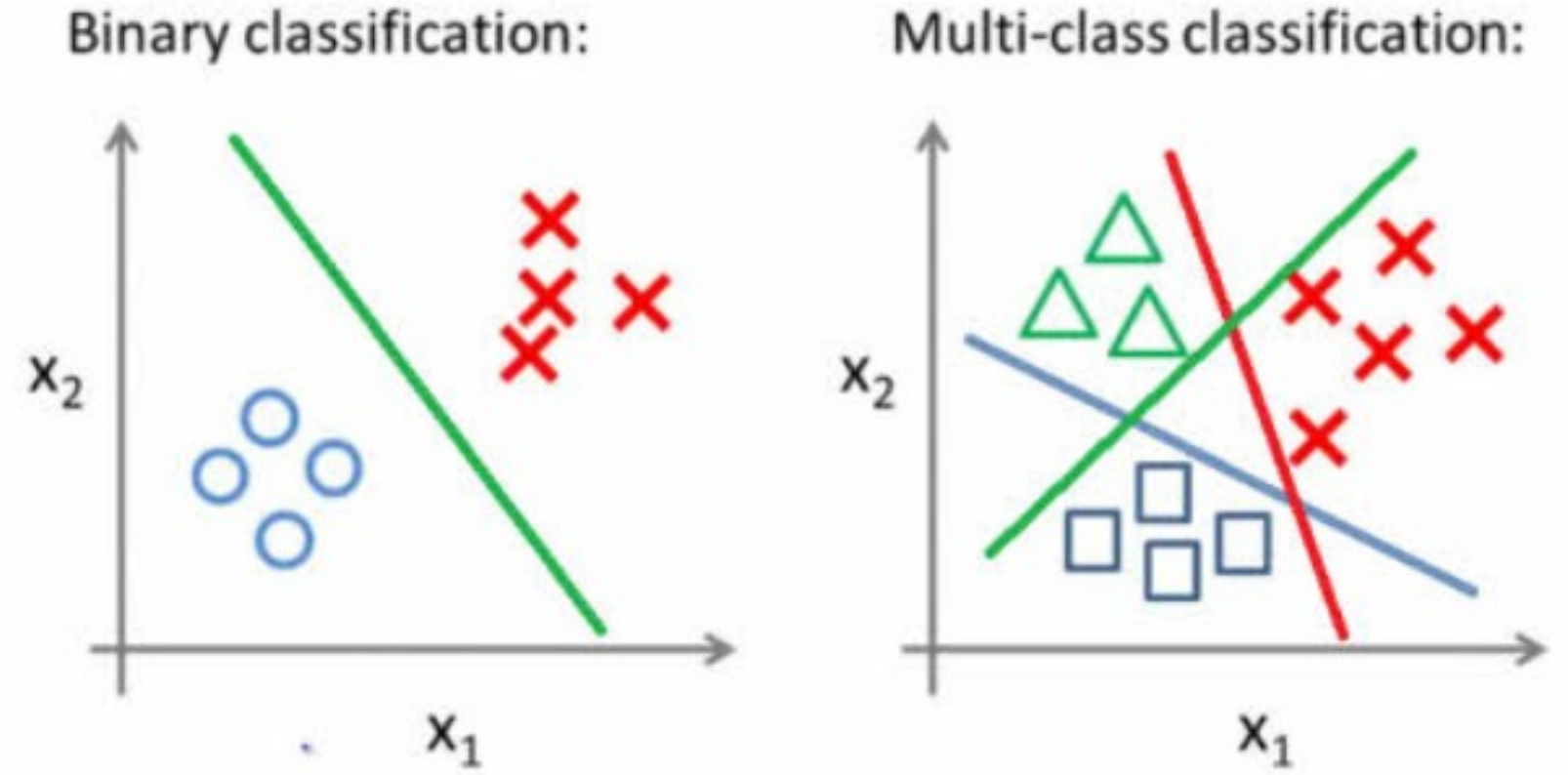
Multiclassification

- When we solve a classification problem having only two class labels, then it becomes easy for us to filter the data, apply any classification algorithm, train the model with filtered data, and predict the outcomes
- But when we have more than two class instances in input train data, then it might get complex to analyze the data, train the model, and predict relatively accurate results. To handle these multiple class instances, we use multi-class classification
- Multi-class classification is the classification technique that allows us to categorize the test data into multiple class labels present in trained data as a model prediction.

Multiclassification

- There are mainly two types of multi-class classification techniques:-
- **One vs. All (one-vs-rest)**
- **One vs. One**

Binary classification vs. Multi-class classification



Binary classification vs. Multi-class classification

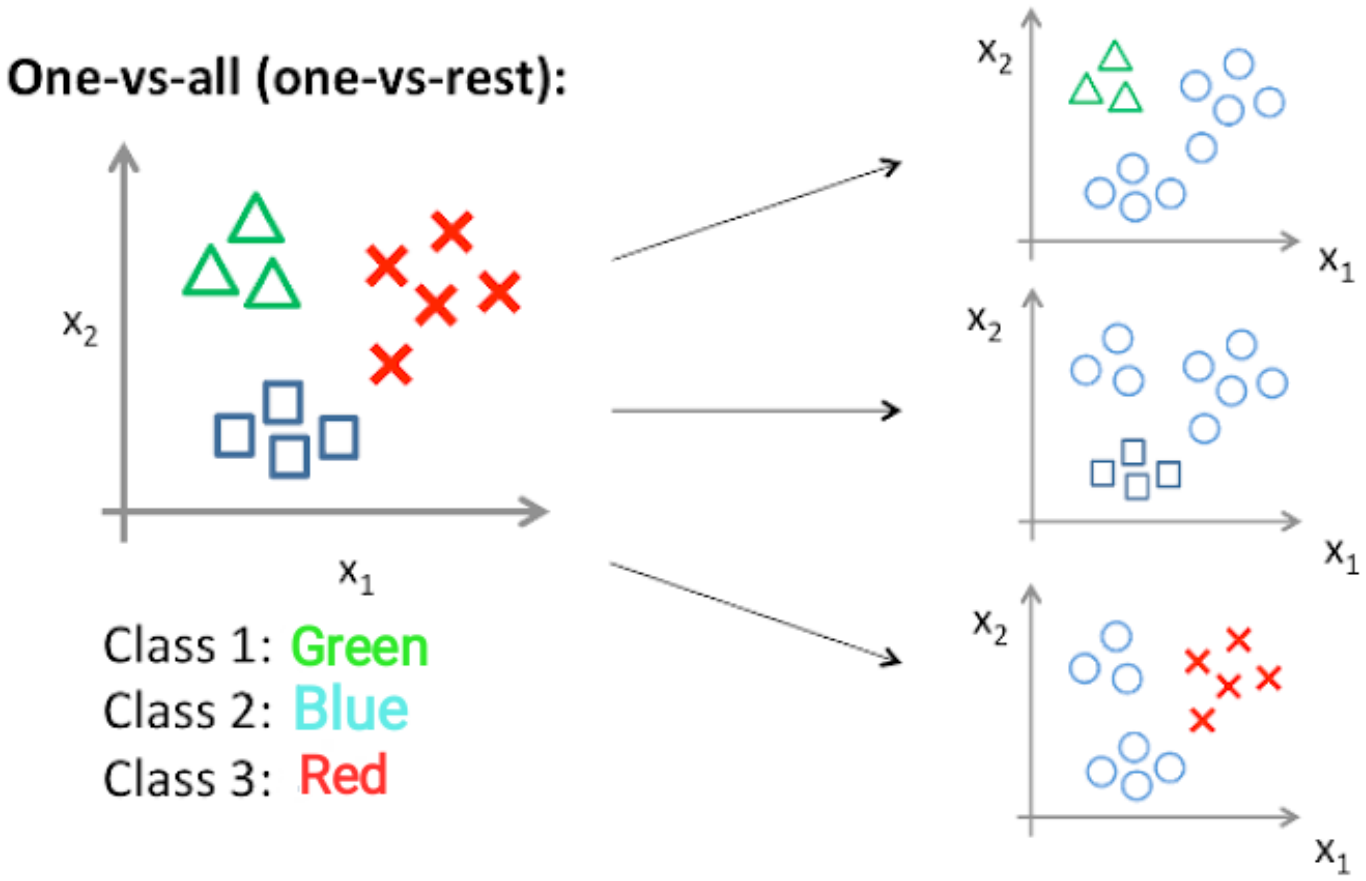
- **Binary Classification**
- Only two class instances are present in the dataset.
- It requires only one classifier model.
- Confusion Matrix is easy to derive and understand.
- Example:- Check email is spam or not, predicting gender based on height and weight.

Binary classification vs. Multi-class classification

- Multiple class labels are present in the dataset.
- The number of classifier models depends on the classification technique we are applying to.
- One vs. All:- **N-class instances** then **N binary classifier models**
- One vs. One:- **N-class instances** then **N* (N-1)/2 binary classifier models**
- The Confusion matrix is easy to derive but complex to understand.
- Example:- Check whether the fruit is apple, banana, or orange.

One vs. All (One-vs-Rest)

- In one-vs-All classification, for the N-class instances dataset, we have to generate the N-binary classifier models. The number of class labels present in the dataset and the number of generated binary classifiers must be the same.



One vs. All (One-vs-Rest)

- Now to train these three classifiers, we need to create three training datasets. So let's consider our primary dataset is as follows,

Main Dataset

Features			Classes
x1	x2	x3	G
x4	x5	x6	B
x7	x8	x9	R
x10	x11	x12	G
x13	x14	x15	B
x16	x17	x18	R

Class 1 :- Green Class 2 :- Blue Class 3 :- Red

Training Dataset 1
Class :- Green

Features			Green
x1	x2	x3	**+1**
x4	x5	x6	-1
x7	x8	x9	-1
x10	x11	x12	**+1**
x13	x14	x15	-1
x16	x17	x18	-1

Figure 6: Training dataset for **Green** class

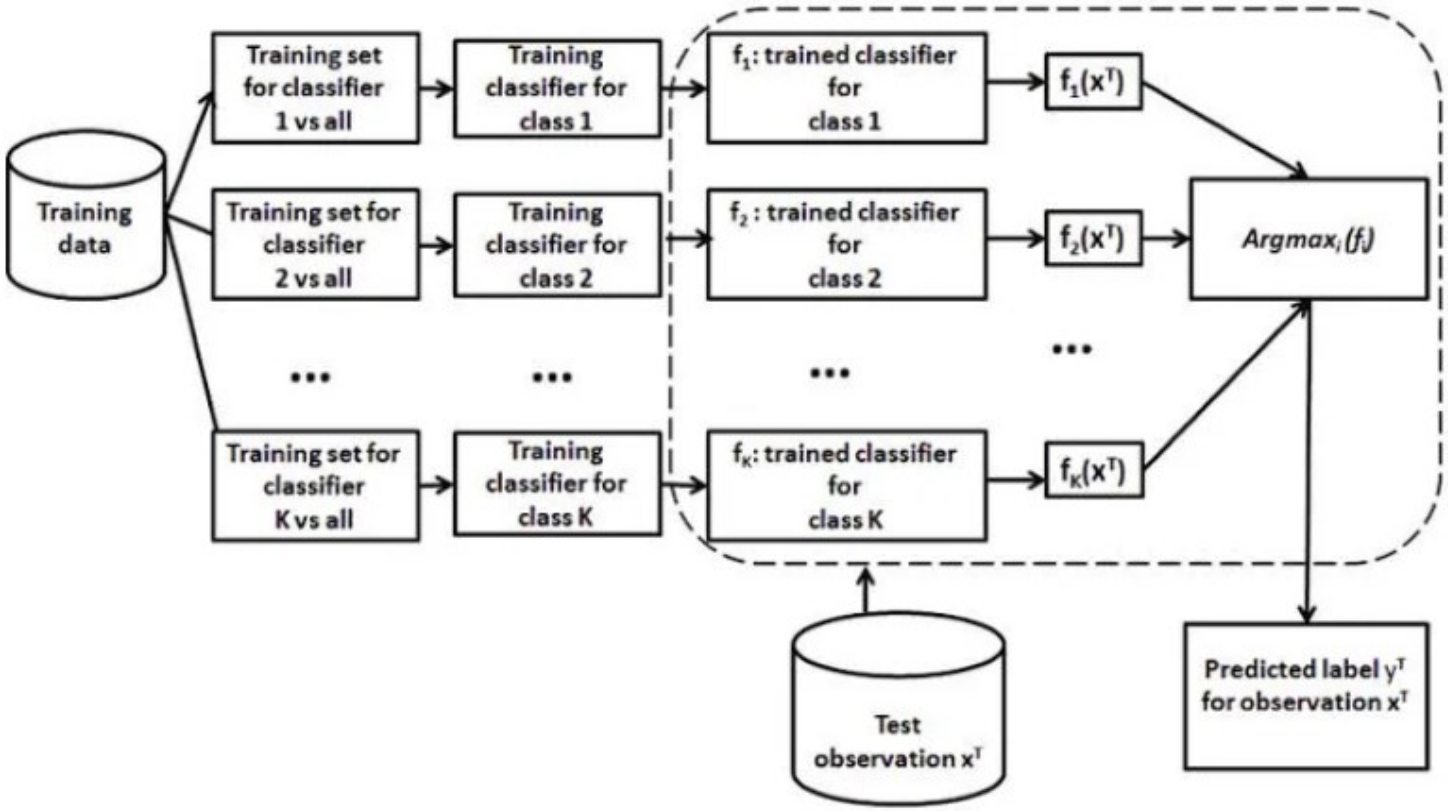
Training Dataset 2
Class :- Blue

Features			Blue
x1	x2	x3	-1
x4	x5	x6	**+1**
x7	x8	x9	-1
x10	x11	x12	-1
x13	x14	x15	**+1**
x16	x17	x18	-1

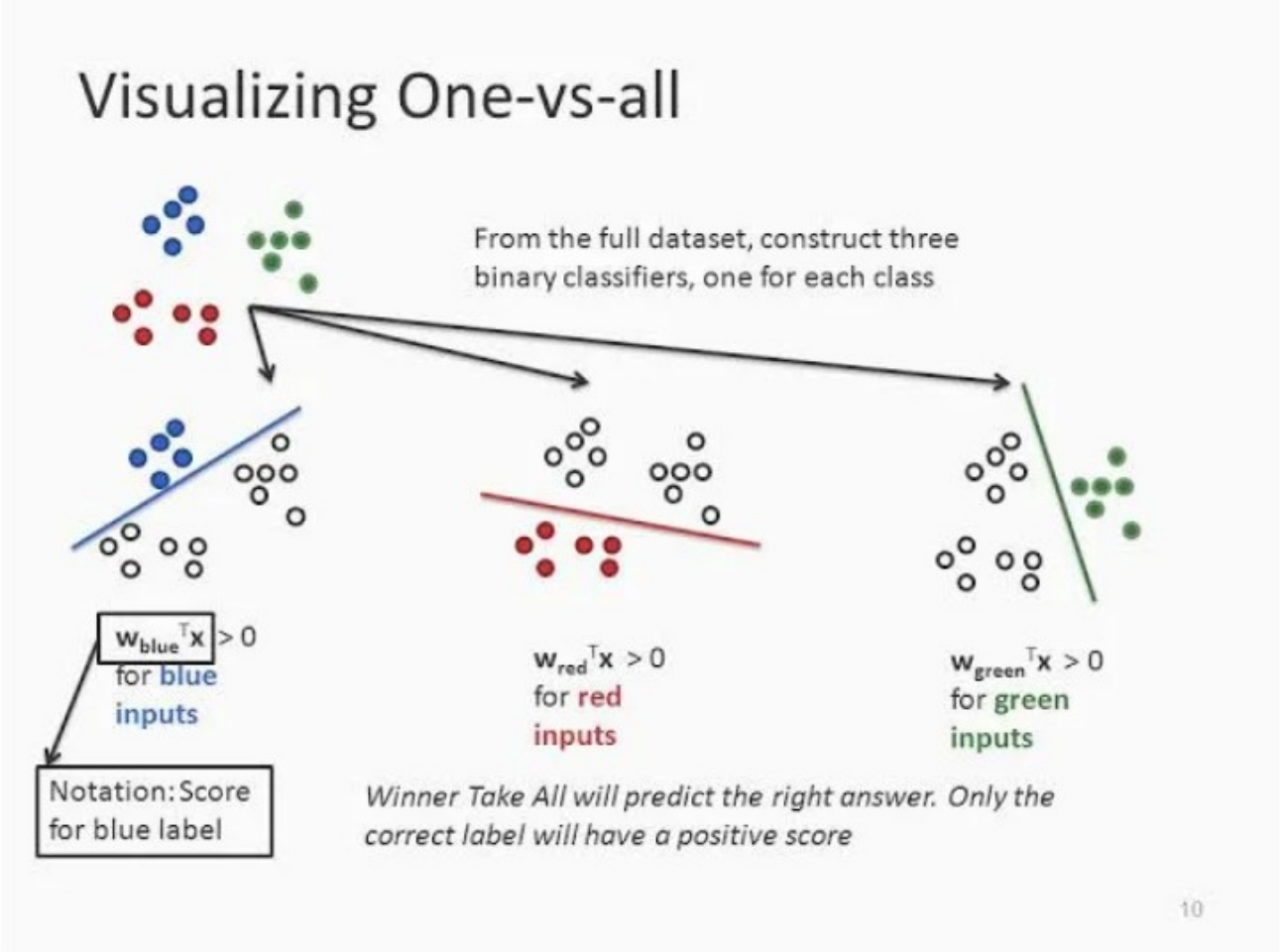
Training Dataset 3
Class :- Red

Features			Red
x1	x2	x3	-1
x4	x5	x6	-1
x7	x8	x9	**+1**
x10	x11	x12	-1
x13	x14	x15	-1
x16	x17	x18	**+1**

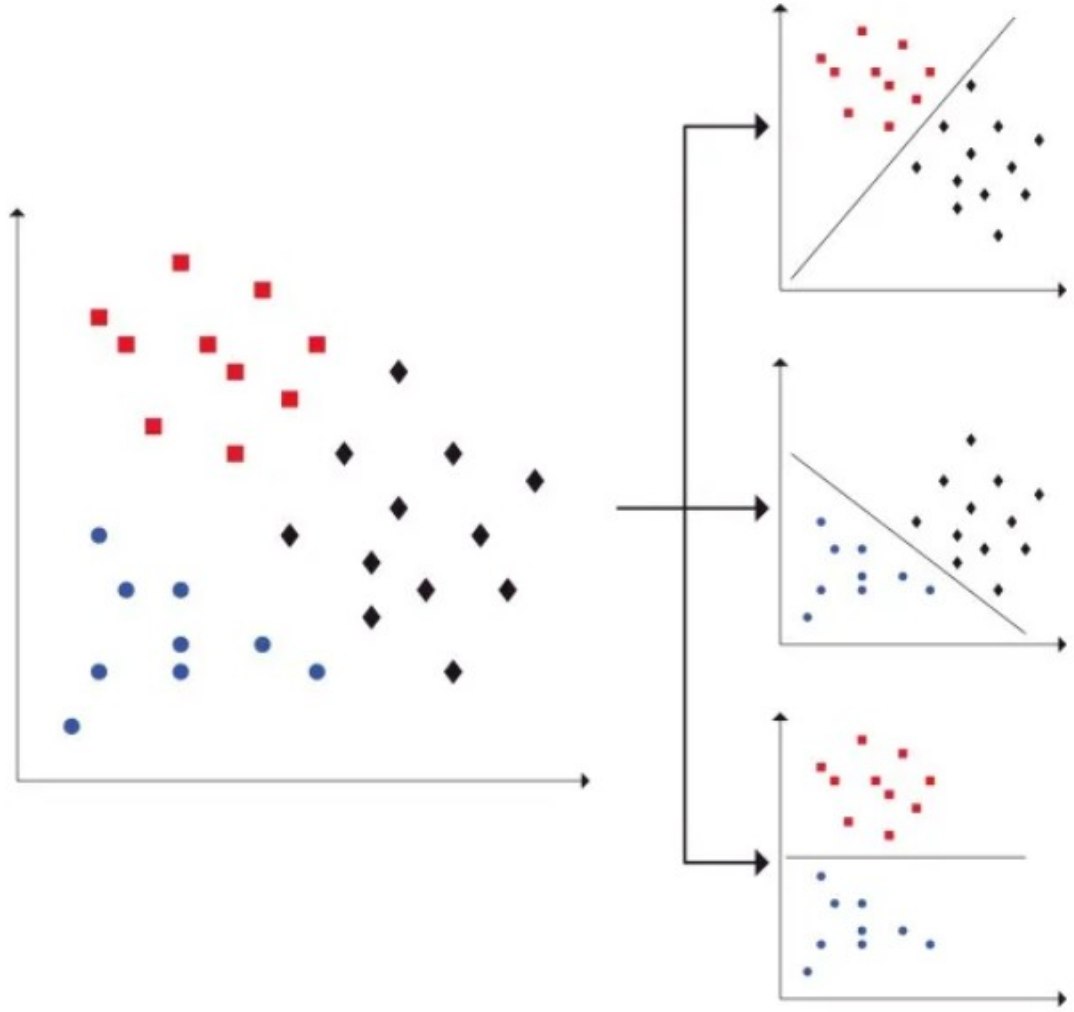
One vs. All (One-vs-Rest)



One vs. All (One-vs-Rest)



One vs. One (OvO)

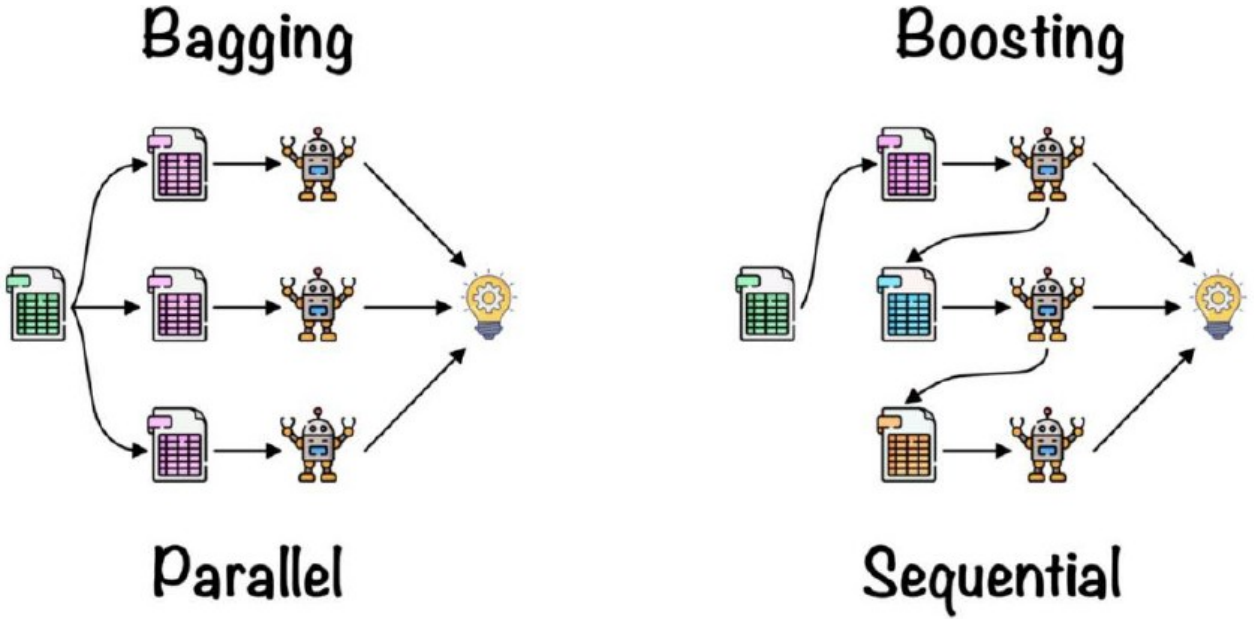


One vs. One (OvO)

- In One-vs-One classification, for the **N-class** instances dataset, we have to generate the **N* (N-1)/2** binary classifier models. Using this classification approach, we split the primary dataset into one dataset for each class opposite to every other class.
- Taking the above example, we have a classification problem having three types: **Green**, **Blue**, and **Red (N=3).**
- We divide this problem into **N* (N-1)/2 = 3** binary classifier problems:
- Classifier 1: Green vs. Blue
- Classifier 2: Green vs. Red
- Classifier 3: Blue vs. Red

RANDOM FOREST

- **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
- **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST



RANDOM FOREST

- **Bagging –**

✓ Bagging, also known as ***Bootstrap Aggregation*** is the ensemble technique used by random forest.

✓ Bagging chooses a random sample from the data set.

✓ Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as ***row sampling***.

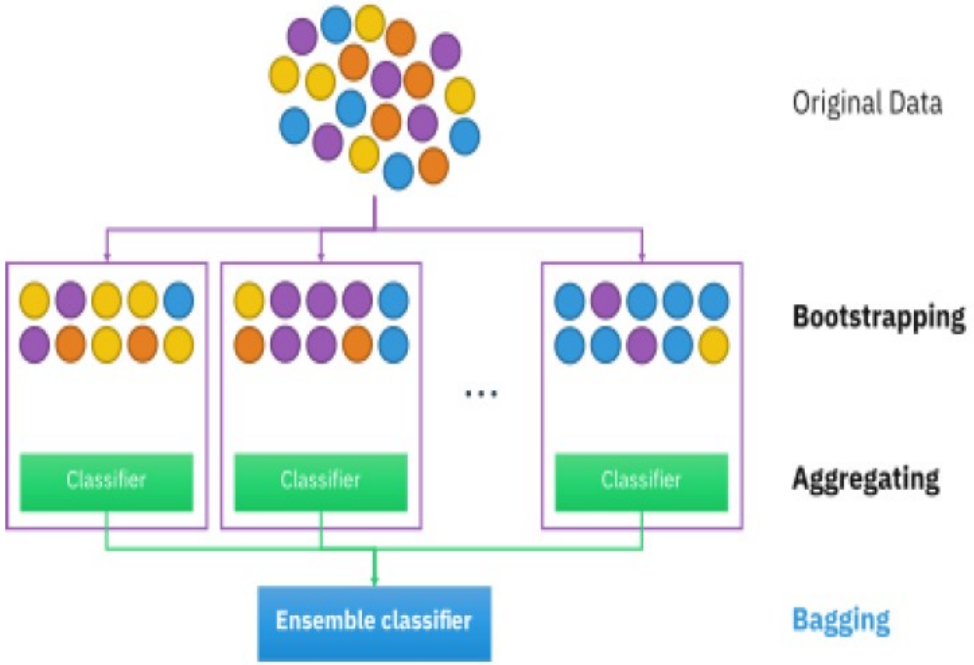
✓ This step of row sampling with replacement is called ***bootstrap***

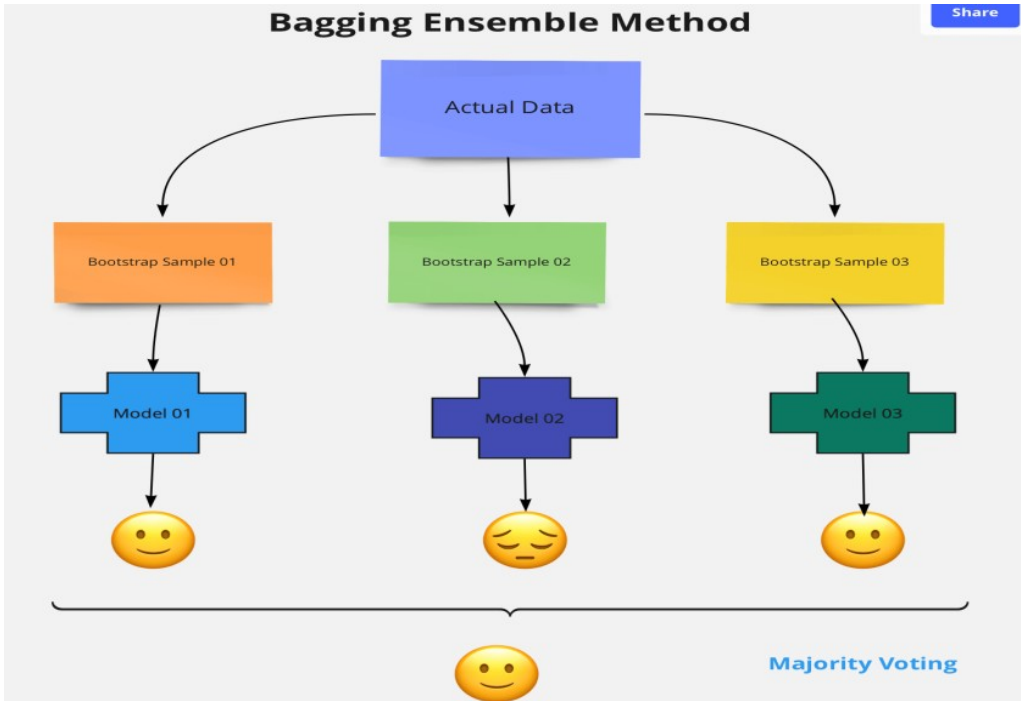
✓ Now each model is trained independently which generates results

✓ The final output is based on majority voting after combining the results of all models.

✓ This step which involves combining all the results and generating output based on majority voting is known as ***aggregation***.

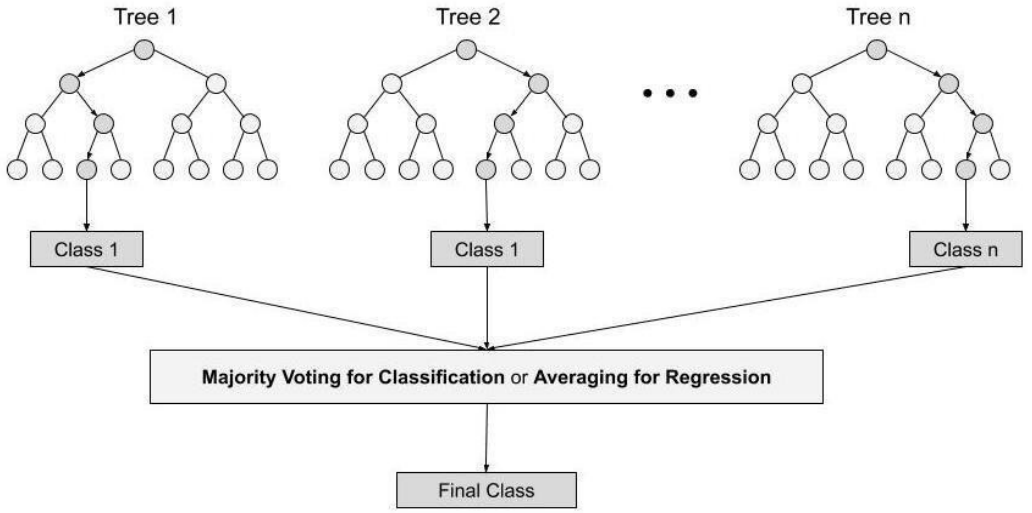
RANDOM FOREST



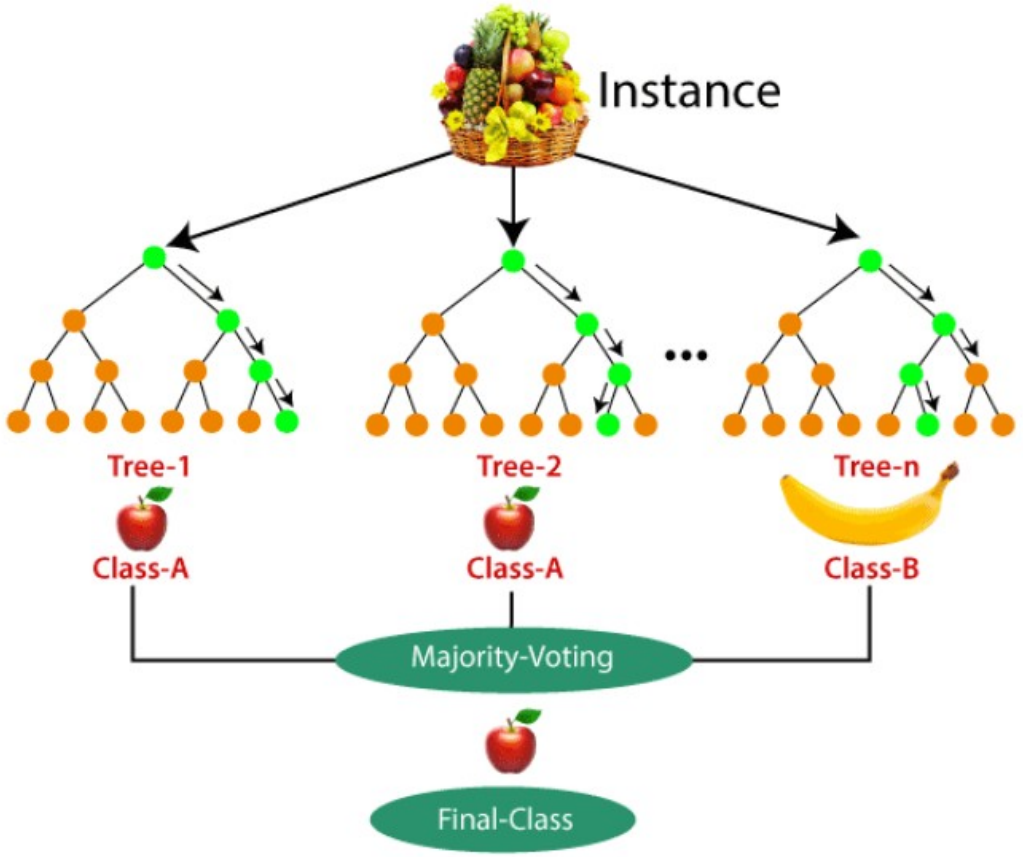


Steps involved in random forest algorithm

- Step 1: In Random forest n number of random records are taken from the data set having k number of records.
- Step 2: Individual decision trees are constructed for each sample.
- Step 3: Each decision tree will generate an output.
- Step 4: Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression respectively.



Steps involved in random forest algorithm



Important Features of Random Forest

1. **Diversity-** Not all attributes/variables/features are considered while making an individual tree, each tree is different.
2. **Immune to the curse of dimensionality-** Since each tree does not consider all the features, the feature space is reduced.

3. **Parallelization-**Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

4. Train-Test split- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

5. **Stability-** Stability arises because the result is based on majority voting/ averaging.