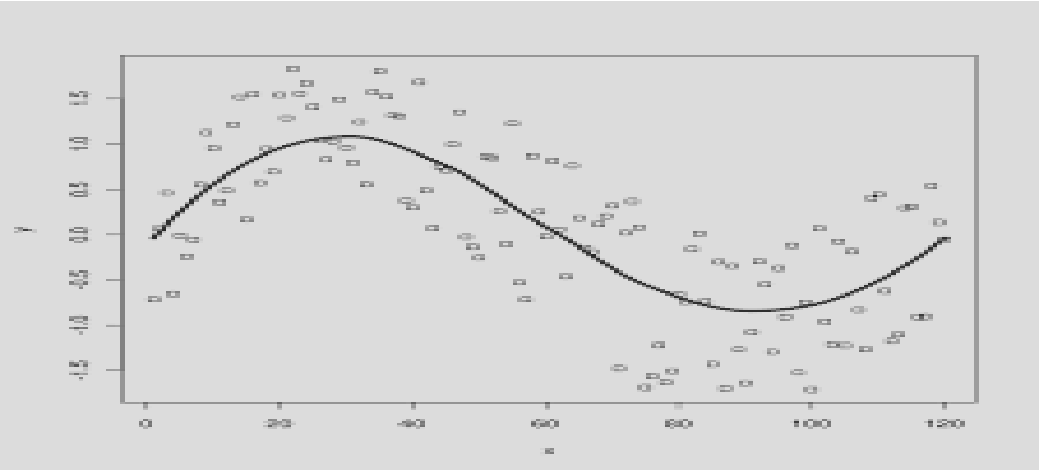
**Machine Learning:**

**What is ML?**

Machine learning plays crucial role in data science. It is ***branch of artificial intelligence*** which focuses on ***developing algorithm and models*** so that computer can ***analyze, learn and make predictions*** based on ***deterministic part of data*** without explicit programing.

***Data = Deterministic part + Random part***



Here the data points which follows sine wave is the deterministic part and the other data points are considerd as random part of data.

The main function of ML model is to ***capture deterministic part of data***.

**Key Concepts of ML:**

***Algorithms***: Sets of rules and statistical techniques that allow the machine to learn from data.

***Data***: Machine learning algorithms require data to learn from. This data can be labeled (for supervised learning) or unlabeled (for unsupervised learning).

***Training***: The process of feeding data to the algorithm and finding relationship or patterns in a data

***Models***: The output of the training process, representing the learned patterns and relationships in the data.

***Prediction***: Using the trained model to make predictions or decisions on new, unseen data

**Steps in ML:**

***1. Data collection:*** collect the data from various sources.

***2. Data preparation:*** transform and prepare the data for further analysis.

***3. Model selection:*** select the appropriate model based on the data to make prediction.

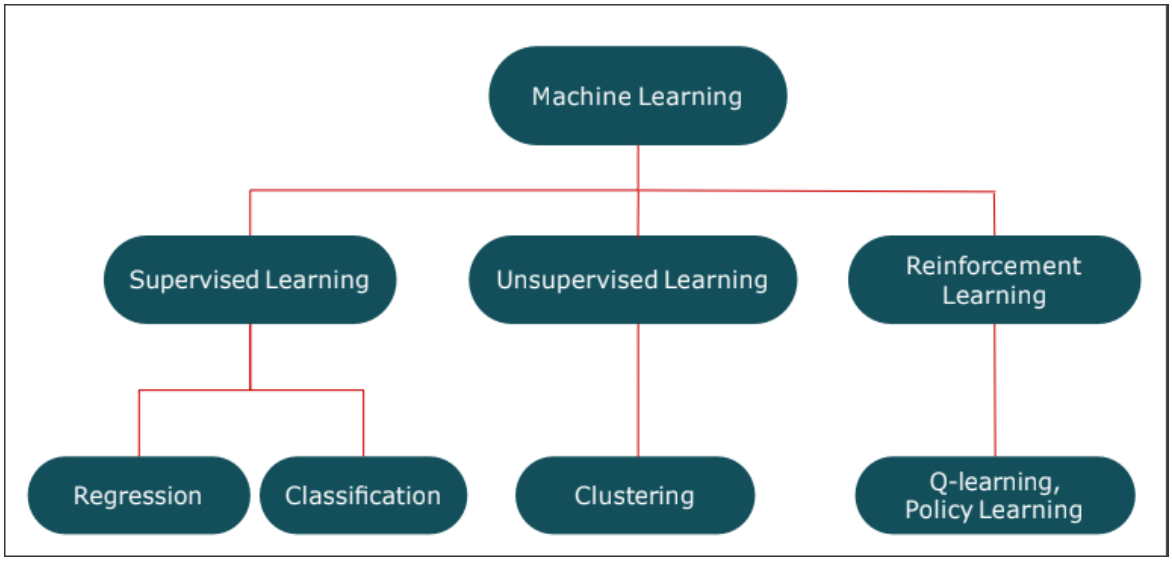
***4. Data training:*** train the data on selected model.

***5. Performance evaluation:*** evaluate the performance of model.

***6. Hyper parameter tuning and cross validation :*** if the model performance is not up to the mark then perform hyper parameter tuning and cross validation to improve its performance.

***7. Prediction:*** Make predictions or decisions.

**Classification of Machine learning:**




**1. Supervised learning:**

In supervised learning, the algorithm are learned from the ***labelled data*** to make predictions or decision. Where dependent variable or target variable is already present in the data.

*Types: - Regression and classification*

## 2. Unsupervised learning:

In unsupervised learning, the algorithms are learned from the ***unlabeled data*** where the task is to analyze pattern or relation or structure in a data to make predictions or decision.

*Type: - Clustering*

## #supervised learning Algorithms:

There are two type of supervised learning algorithms

1. ***Regression :*** In regression, The target variable is ***numeric and continuous***
2. ***Classification :*** In classification, target variable is ***categorical and discrete***

**Regression** is a statistical method that **models the relationship between dependent and independent variables** for predicting a value of dependent variable based on the values of one or more Independent variables

### 1. Linear Regression :

**1**. Linear regression is the most basic type of regression in the supervised Machine learning where the ***dependent variable and independent variable are linear*** to each other.
**2**. It is also known as ***ordinary least square or linear least square***.

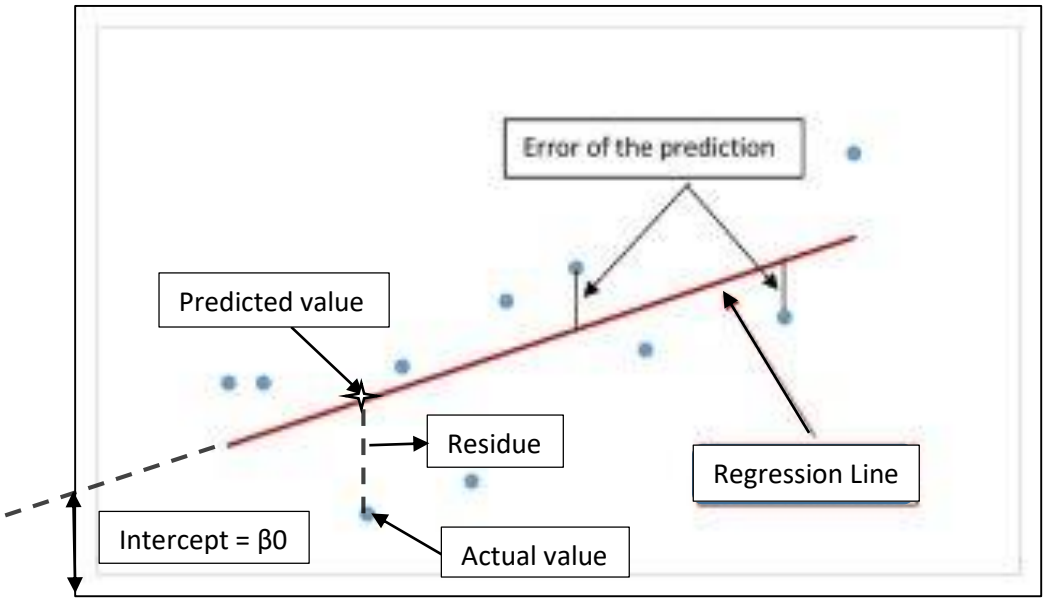
**Equation of multiple linear regression**:

$$Y'' = \beta 0 + \beta 1x1 + \beta 2x2 + ... + \beta nxn + \epsilon$$

Where,
β0 = Intercept or bias term
β1, β2, ….  βn = Model parameters or coefficients or the slope of individual features.
x1, x2, …. x3 = Features or variable or columns
Y" = Predicted output
Y = Actual output
ϵ = Random error

**3**. Main ***goal of linear regression*** is to find the ***best fit line*** or regression line.
**4**. Best fit line is the line for which the ***error*** between predicted value (Y'') and observe value (actual value $y$) is **minimum**.
**5.** ***Best fit line*** is also known as ***regression line*** and ***error*** is called ***residue***.

**6**. The ***vertical error*** between predicted value and actual value from Best fit line is called ***prediction error***.




**Error:**

***Sum squared error = SSE = Σ(y_actual - y_predicted)^2***

***Mean square error = MSE = (1/n) * Σ(y_actual - y_predicted)^2***

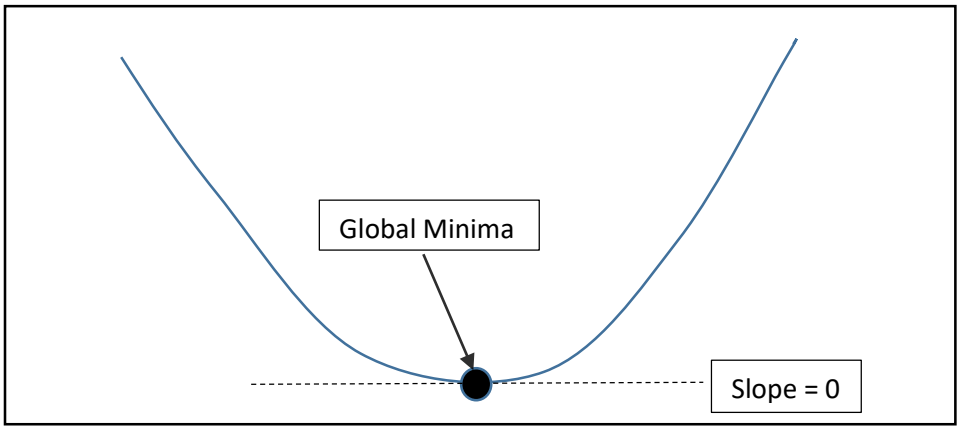
**7**. So basically the main function of linear regression is to ***minimize the error*** or residue to get the best fit line or regression line

So the objective function of linear regression is:
***Min (SSE) = Σ(y_actual - y_predicted)^2***

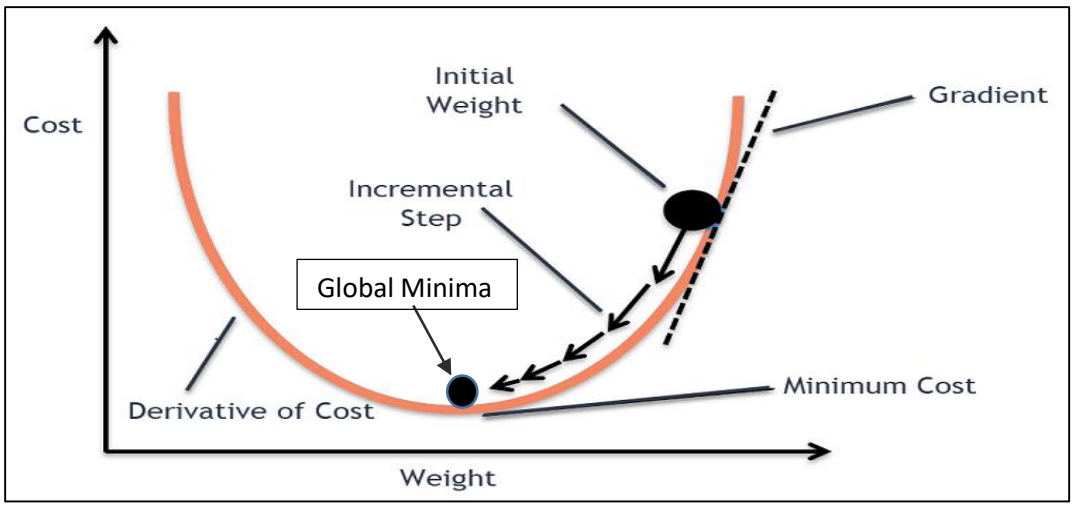
**8**. To minimize the SSE we use different ***β*** value and the ***value which gives minimum SSE*** will produce best fit line and is called ***best possible value (βp***).
**9**. The ***least square estimator method*** is used to find the best possible value of model parameter's β and intercept β0 to get the minimum value of SSE.

**<u>There are 2 method to find out the minimum error:</u>**

***1. Normal equation method:*** Normal equation method gives minimum value of error in single shot. It directly calculate optimal values of the regression coefficients using mathematical equations. (Close form solution)




***2. Gradient decent algorithm:*** while gradient descent gives the minimum value of error using repeated steps.



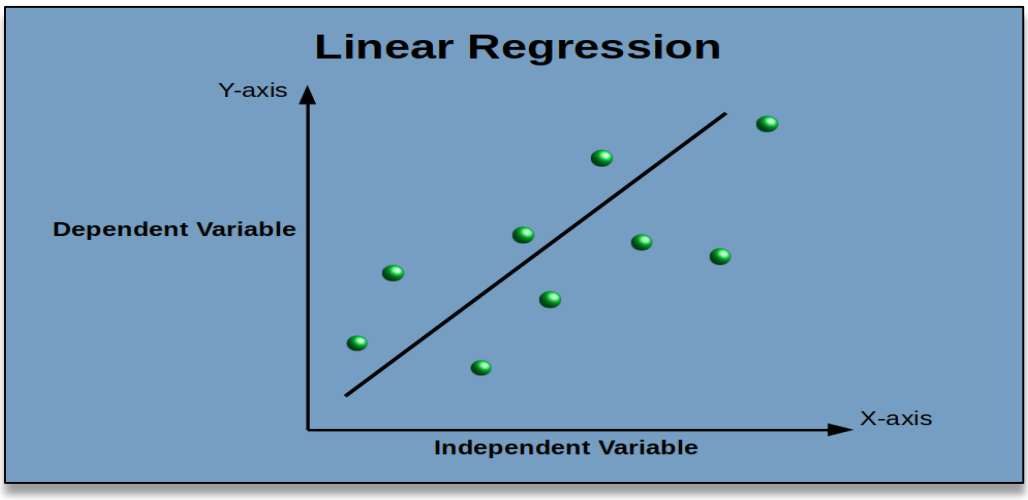

***Here's a breakdown of how gradient descent works:***

1. Start with Initial Parameters (coefficients and intercept)
2. Calculate the Gradient (partial derivatives of the cost function with respect to each parameter.)
3. Take a Step in the Opposite Direction (determined by the *learning rate* α)
4. Update the Parameters (Subtract gradient from current parameter and move closer to minimum value)
5. Repeat steps 2 to 4

**Assumption of linear regression:**

***1. Relationship between independent and dependent variable should be linear:***

There should be a linear relationship, it might be either positive or negative no matter but there should be a linearity.




***2. Multicollinearity should not be there in regression model:***

Multicollinearity occurs when Independent variables are highly correlated to each other. Making it difficult to determine the true effect of each predictor on the dependent variable

***3. The Error or residue should be normally distributed.***

***4. Homoscedasticity in the regression model is must:***

***Homoscedasticity***: - same variance -----> error term is same across all value of independent variable.

This indicates that the ***model is well-defined***,

Meaning that the dependent variable is correctly defined by the predictor variable.

***Heteroscedasticity***: - different variance ------> error term is different across all value of independent variable.

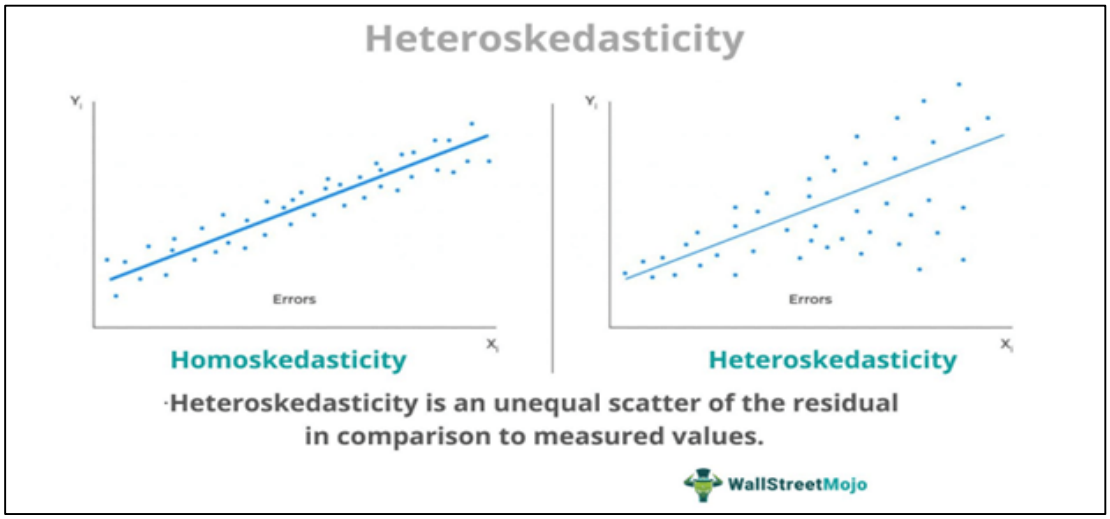
This indicates the ***model isn't well-defined***.

Too much variance indicates that there are ***other factors influencing the dependent variable.***

These factors need to be considered through further investigation or modeling.

To ***deal with heteroscedasticity*** we need to transform the target variable data, *like* ***log transformation, outlier's treatment***, etc.

***How to check if the there is homoscedasticity or not?***
Calculate Ratio between the largest variance and the smallest variance.
If the ratio is 1.5 or smaller, then the regression is homoscedastic.




**#Properties of regression line:**
1. Regression line always passed through mean of independent variable as well as mean of dependent variable.

**Effects of Multicollinearity:**

***Unstable Coefficient Estimates:*** Multicollinearity can cause the estimated regression coefficients to become highly sensitive to small changes in the model.

This means that adding or removing a variable can lead to large fluctuations in coefficient values,

Making it difficult to determine the true effect of each predictor on the dependent variable

***Increase standard error***: As multicollinearity leads to large fluctuations in coefficient values. It increase the error value and includes over fitting of the model.

for an instance we can say that while calculating house price, there are two features number of rooms and square feet area, as we know higher the number of room higher the square feet area, so they are highly correlated and there is multicollinearity, they provide similar information about the price of house, It becomes challenging to determine the individual effect of each feature on house prices since both are contributing similar information. You can consider removing one of them from the model. For instance, you might choose to keep either the number of rooms or square footage, but not both.

By doing so, you simplify the model and reduce multicollinearity, allowing for more reliable coefficient estimates and clearer interpretations.

***To detect multicollinearity: calculate VIF***

If VIF = 1 then no correlation

If VIF > 1 and VIF < 5 then moderate

If VIF > 10 then high correlation

If there is multicollinearity in the model then we need to drop the features or variable who is more correlated or we need to do feature engineering

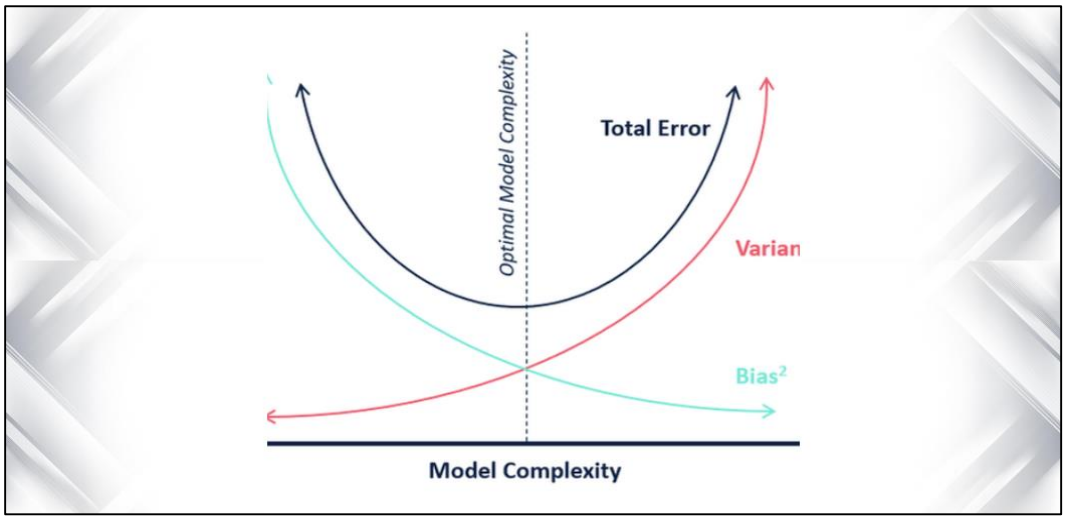
To create a new feature from existing feature.

**Bias Variance Tradeoff:**

The **bias-variance trade-off** is about finding the right balance between simplicity and complexity (Bias and Variance) in a machine learning model. Basically, you want a model that is complex enough to capture the underlying patterns in the data but not so complex that it overfits the training data.

**Bias**: Bias in machine learning refers to the ***systematic error*** that occurs when a model makes ***incorrect assumptions about the data***,
Leading to ***inaccurate predictions***.
In the simplest terms, Bias is the difference between the Predicted Value and the Expected Value.

**Variance**: It essentially measures how ***sensitive*** the model is to ***fluctuations*** in the training data.
The model captures noise and fluctuations, leading to over fitting (good performance on training data but poor generalization on testing data).



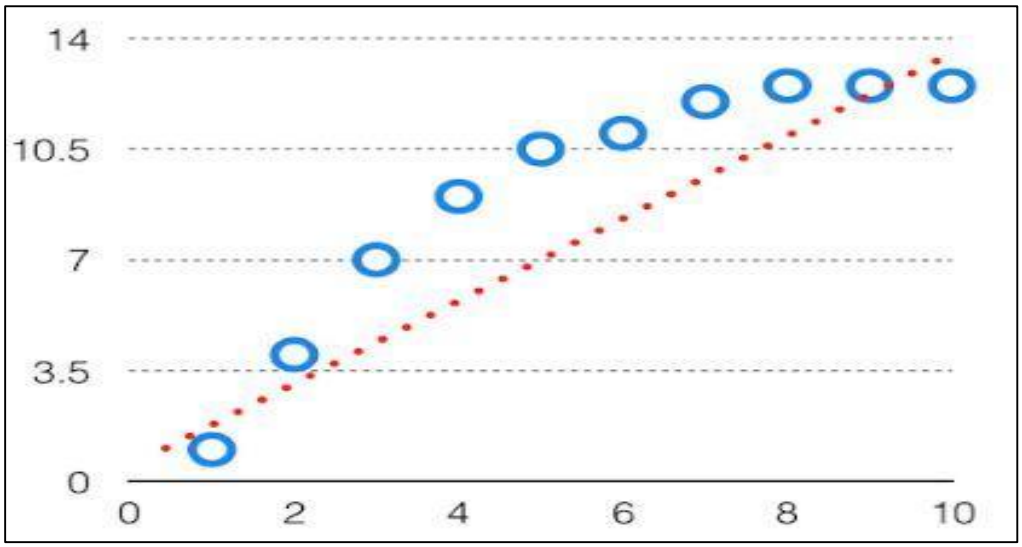

X-axis represents model complexity, and the y-axis represents error
As complexity increases, bias tends to decrease and variance tends to increase.
The total error is the sum of bias, variance, and irreducible error, and often shows a U-shaped curve where the optimal model complexity is at the bottom of the U.

**<u>Underfitting</u>**: (***High bias and Low variance***)
Underfitting occurs when model is too simple to capture underlying patterns in data, resulting in poor performance on both training and testing data

***Sign of underfitting:***
1. Low accuracy on training data
2. Low accuracy on testing data

***Causes of underfitting*:**
1. Model is too simple
2. Less training time or insufficient training time
3. Ignoring important features. It take bias decisions based on some features and ignore other features.

***Overcome underfitting:***
1. Try using more complex model
2. Add more features
3. Increase training time



**<u>Over fitting</u>*: (Low bias and High variance)***

Over fitting occurs when model is to complex and it captures noise or fluctuations in the training data rather than learning true underlying patterns As a result it perform well on training but fail to perform well on testing data.

***Sign of overfitting:***

1. High accuracy on training data
2. Low accuracy on testing data

***Causes***:

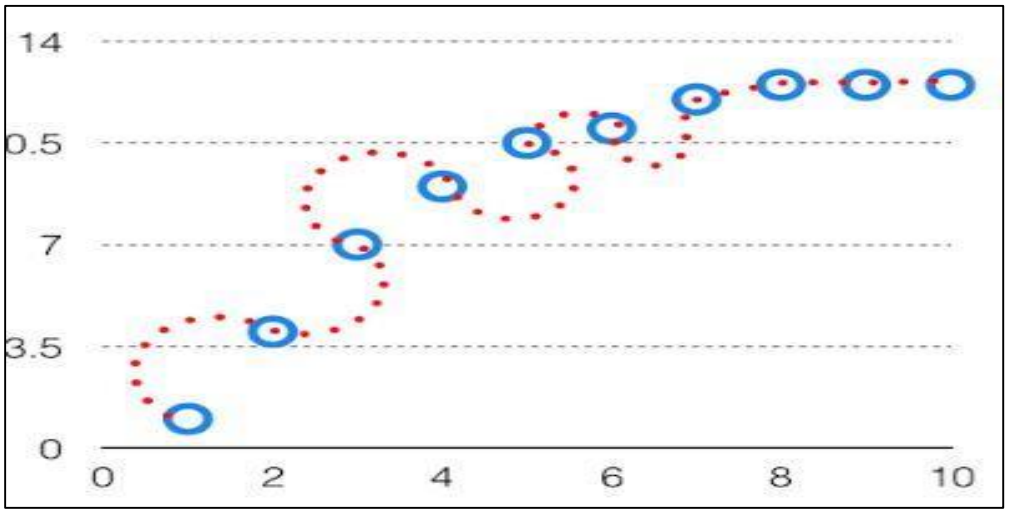
1. Model is too complex
2. Model has many features
3. High coefficient values

***Overcome overfitting:***

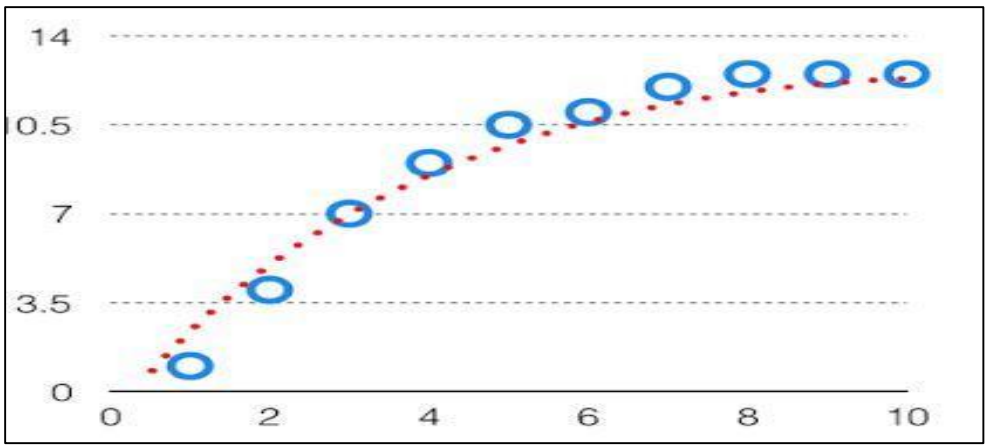
1. Increase size of training data (which increase bias and decrease variance)

If the dataset is having fewer data points then Model tries to capture all the data points as shown in the below fig. so increase the size of training data will capture the underlying pattern rather than all data points and will reduce variance.

2. Use regularization technique such as lasso and ridge which minimizes the coefficient and reduce overfitting.
3. Reduce complexity of model.
4. Cross validation techniques



***Optimal Fit: (Balance between bias and variance)***




**Regularization Techniques:**

As we know as the model complexity increases the chances of overfitting also increases.

To overcome overfitting , we use lasso and ridge regression which takes care overfitting due to high value of coefficient and outliers.

Penalty term decrease the coefficient value to 0 or up to 0

***Lasso*** = L1 regularization technique ---- reduced coefficient value to 0 and remove from dataset (Absolute coefficient)

$$\text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

***Ridge*** = L2 regularization technique ---- reduced coefficient value up to 0 not exactly zero (Squared values)

$$\text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Tuning parameter used in the penalty term is lambda (λ)

**Cross validation:**

For large dataset we have sufficient data to split into train and test the model, so we can train good and test good. No need of cross validation

For the small dataset with limited data, cross validation is crucial, which allows you to repeatedly train and test your model on different subset of data.

***Cross validation*** *is a technique to* ***split the data into subset of k fold*** *where we* ***train the data on kn fold****s and* ***test the data on remaining folds****.*
*Cross validation allows you to train and test the model repeatedly.*

**Steps:**

1. Split the data set into K folds of equal size
2. Take one fold as the reserve or test data set.
3. Train the data on remaining folds
4. Evaluate the performance of the model using the test set.
5. Repeat the process till all folds tested.

**Hyper parameter tuning:**

Hyper parameter tuning involves ***finding the best combination of hyper parameters*** that leads to ***optimal model performance***. (Balance bias and variance trade-off)
Each machine learning algorithm favours its own respective set of hyper parameters

We have ***GridSearchCV and RandomSearchCV***

***GridSearchCV:***

It searches for the best set of hyper parameters from a grid of hyper parameters values.
But the slowness is a disadvantage. It often takes a lot of processing power and time to fit the model with every potential combination,

***RandomSearchCV:***

Instead of testing every possible combination, it randomly selects a set of hyper parameter values from defined ranges or distributions.
For most scenarios, "random search" is generally considered the more effective hyper parameter tuning technique compared to grid search, especially when dealing with a large number of hyper parameters or a complex search space

**Performance evaluation of linear regression:**

1. Mean square error
2. Root mean square error
3. Mean absolute error
4. Mean absolute percentage error
5. R2 : R-squared indicates how well data points fit a statistical model.
6. Adjusted R square: Adjusted R-squared adjusts R-squared for the number of predictors in a model, providing a more accurate measure when comparing models with different numbers of predictors.

***Acceptable Values for Errors***

Generally, lower values for all error metrics indicate better model performance.

1. MSE and RMSE should ideally approach zero.

2. R-squared values range from 0 to 1, with higher indicating better fit.

3. MAE and MAPE should also be minimized for optimal predictions.

**Importing Important Libraries:**

***# Data manipulation and numerical computation***

Import pandas as pd
import numpy as np

***# Visualization***

import matplotlib.pyplot as plt
% inline matplotlib
import seaboarn as sns

***# Model building and evaluation***

from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet

from sklearn.preprocessing import StandardScaler, PolynomialFeatures from sklearn.metrics import mean_squared_error, r2_score

## 2. Logistic Regression: (Only classification)

1. ***Logistic regression*** is a type of supervised learning algorithm belongs to ***classification*** problem. Where the target variable is a categorical column and have *only* 2 values (yes-no, true-false) etc.
As the target variable is having only 2 values hence it is called as ***dichotomous variable***.
2. Logistic Regression is also known as ***binary classification algorithm*** *where it* ***predicts the probability*** *of an instance belonging to a particular* ***class***.
3. Although the target variable contains categorical values still we called it as Logistic Regression, because we are going to ***assign a numerical value to target variable.***

***Linear regression equation is:***

$Y'' = \beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta nxn$

Now for the logistic regression ***the logistic function*** is

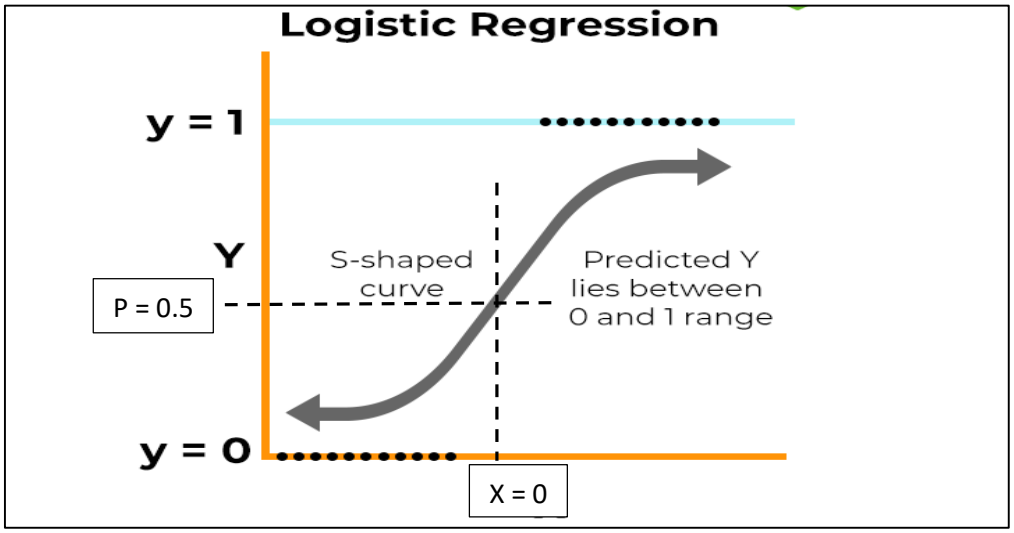
$p = 1 / (1 + \exp(-mtx))$

Where,

$mtx = \beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta nxn$

4. Now when we plot the ***logistic function***, we get a ***sigmoid curve*** which states that
At ***x=0, p = 0.5***
As value of ***x increases the p also increases***
As value of ***x decreases the p also decreases***.

***At any value of x the p will be lies between 0 to 1***



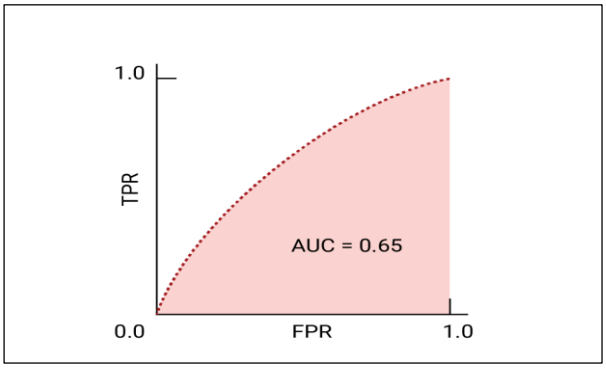
**Imp Note:**
The cost function which used in logistic regression is called as maximum likehood estimation (MLE) function.

**#Assumption of logistic regression**
1. Dependent variable should be dichotomous
2. Multicollinearity should not be there in the model (Independent variable should not be highly correlated to each other)

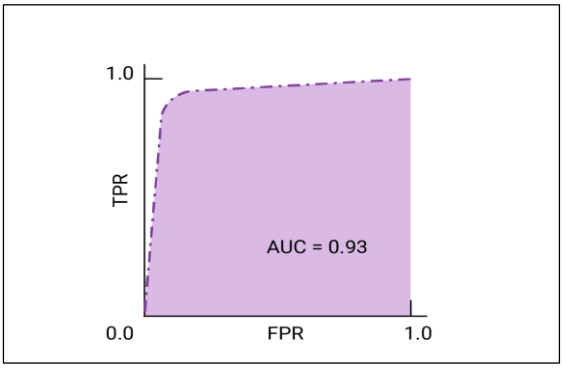
**Regularization:** L1 (LASSO) and L2 (Ridge) regularization can be applied to logistic regression to prevent over fitting, just like in linear regression.

**Performance evaluation of logistic regression includes**
**a. *Accuracy*** = correctly predicted / total number of data points
**b**. ***Precision*** = true positive / predicted positive
**c. *Recall*** = true positive / actual positive
**d**. ***F1_score*** = harmonic mean of precision and recall i.e 2 * (precision * recall) / (precision + recall)
**e. *Log_loss (Logistic loss): (measure uncertainty of model prediction)***
Higher log loss --→ bad fit
Lower log_loss --→ best fit

**f**. ***AUC*** (Area under the Curve):
Overall ability of model to classify positive and negative classes.
AUC ranges between 0 to 1
1 = perfect classification
0.5 = moderate classification
Below 0.5 = poor classification
AUC is a useful measure for comparing the performance of two different models.
The model with greater area under the curve is generally the better one.



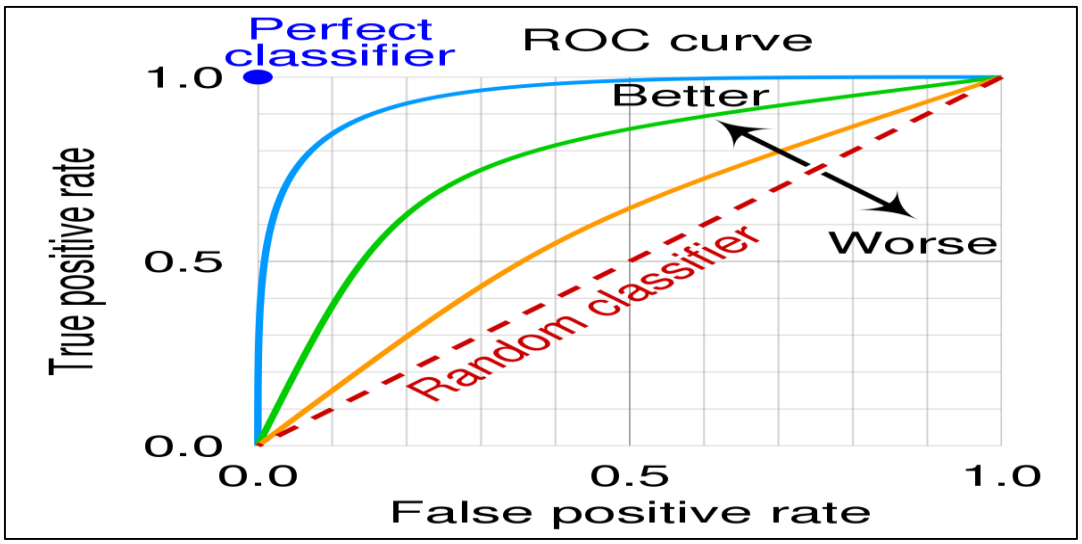
Good model



Better model

**g**. ***ROC Curve*** (Receiver Operating Characteristic):

1. ROC plots the True Positive Rate (TPR, or Recall) against the False Positive Rate (FPR)
2. An ideal ROC curve hugs the top left corner, indicating a high TPR and low FPR.

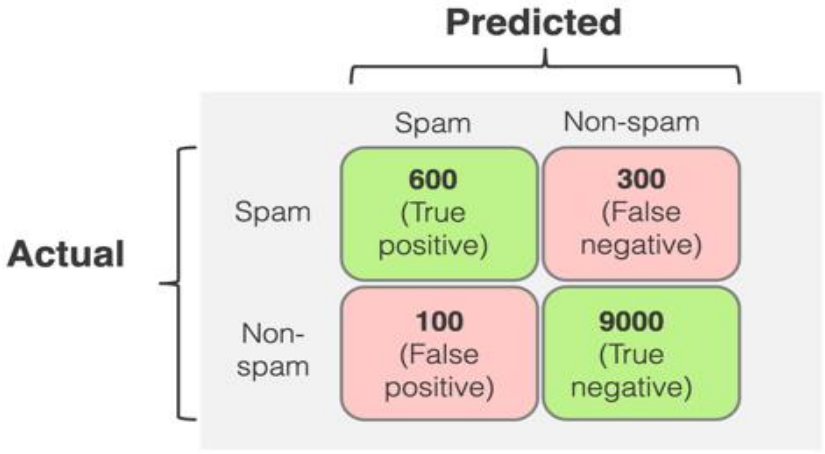



***To calculate this metrics we need to calculate confusion matrix.***

**<u>What is confusion matrix?</u>**

A confusion matrix is a crucial tool in machine learning, particularly for evaluating the performance of classification models.

It provides a detailed breakdown of the correct and incorrect predictions made by the model.



3. **<u>Decision Tree</u>: ( both classification and regression task )**

1. Decision tree is a supervised learning algorithm used for both classification and regression task.
2. The tree used for ***classification task is classification tree*** and the tree used for ***regression task is called regression tree***.

***Classification task:***

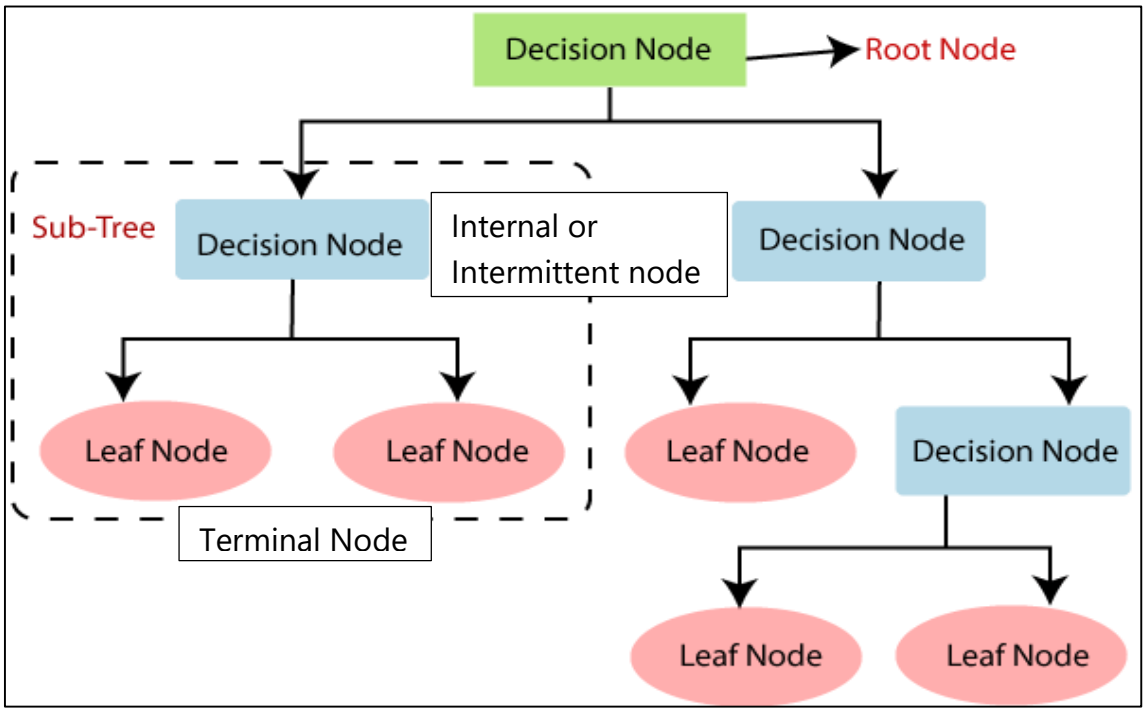
Target variable is categorical and discrete.

Classification tree classify the target variable into different classes.

***Regression task:***

Target variable is numerical or continuous.

Regression tree is used to predict the value of target variable.



***The structure of decision tree includes:***

***1. Root node:*** It is a starting node of decision tree. Root node represent entire population and gets divided into two or more homogeneous subset. (Represented by Square)

***2. Decision node:*** The sub node which gets divided into further sub node bases on specific feature or specific condition then it is called decision node. Decision node is also called as **internal, intermittent node.**
(Represented by Square)

***3. Leaf node:*** The node which do not splits into sub node is called leaf node. Leaf node also called as **terminal node where decision or prediction made**.
(Represented by Triangle)

***4. Branch or sub tree:*** it is part of tree or sub section of decision tree.

***5. Parent and child node:*** the node which splits into sub node is called parent node and the sub node is called child node of parent node

***6. Pruning:*** pruning is a technique to reduce the size of tree by **removing branches** or sub tree which do not contribute more in improving the accuracy of prediction.
**Pruning** helps to **prevent overfitting**

**How to split the node:**
The feature which is more homogeneous is best suited for the split

***Methods to determine best split:***
1. Information Gain
2. Gini score
3. Chi Square
4. Reduction in Variance

The most common methods are ***Information gain and Gini score*** which uses entropy as criteria to determine homogeneity of data

**Entropy:**

A ***measure of randomness or impurity*** in data is called entropy.

Higher the entropy --→ higher the randomness --→ data is less homogenous

Lower the entropy --→ lower the randomness --→ data is more homogenous

***Formula***:

***Entropy = -plog2p -qlog2q***

---

**Information gain**:

1. Information gain measures how much the information a particular feature provides about class label.
2. Information gain measures reduction in entropy after splitting the data based on particular feature.

Higher the IG --→ higher the reduction in entropy --→ Data is more homogeneous

Lower the IG --→ Lower the reduction in entropy --→ Data is heterogeneous

***The feature which is having highest I.G is best suited for the split***

***Steps to calculate IG:***

***1. Calculate entropy of entire dataset based on the value in target variable***

Total entropy S (total) = -plog2p -qlo2q

Where

p = probability of class A

q = probability of class B or probability of not class A

Suppose there are 100 rows out of which 70 is yes and 30 is no

Then

p = 70 / 100 = 0.7

q = 30 / 100 = 0.3

S (total) = -plog2p -qlo2q

***S (total)*** = -0.7 * log2 (0.7) - 0.3 * log2 (0.3) = ***0.8816***

***2. Calculate entropy of each individual node of split***

I.e. feature is gender and have 2 value male and female

Entropy of female

Entropy of male

Suppose there are 30 female and 70 male
P (male) = 70 / 100
P (female) = 30 /100
***Entropy of female;***
No of female = 30
Female (yes) = 10
Female (no) = 20

Then
P (female yes) = 10/30
P (female no) = q = 20 / 30

Entropy of female = -plog2p -1log2q
***S (female)*** = -0.333 * log2 (0.333) - 0.667 * log2 (0.667) = ***0.918***

Similarly
***Entropy of male:***
No of male = 70
Male (yes) = 60
Male (no) = 10

P (male yes) = p = 60/70
P (male no) = q = 10/70

Entropy of male = -plog2p -qlog2q
***S (Male)*** = -0.857 * log2 (0.857) - 0.143 * log2 (0.143) = ***0.580***

Now we have entropy of both sub node (male and female)

***3. Calculate weighted entropy of node***
Weighted entropy of Gender:
S (gender) = P (male) * s (male) + p (female) *s (female)
S (gender) = 70 / 100 * s (male) + 30/100 * s (female)
***S (gender)*** = 0.7 * 0.580 + 0.3 * 0.918 = ***0.681***

***4. Calculate IG = total entropy - weighted entropy***
IG (Gender) = S (total) - S (gender)
***IG (Gender)*** = 0.8816 - 0.681 = ***0.201***

***Similarly perform the same to all feature and feature which is having highest IG is best suited for split.***

**Gini score:**

Gini score is even simpler than IG.
It performs only binary splits.

***GS = P square + q square***

---

**Steps**

1. Calculate Gini score of individual subset ---- male and female
2. Calculate weighted average of gini score ---- gini score of gender
3. Similarly do this for all feature and their sub node.

gini score (feature1) > gini score (feature 2)

Then feature 1 is best suited for split

---

***Gini impurity = 1- gini score***

Higher the gini score ------> less is the impurity --------> Data is more homogeneous

Gini impurity is also called as ***gini index***

---

**IMP note:**

1. If target variable has binary classification, go for both IG and Gini score, and which gives better result use for further analysis
2. If more than 2 target variable then go for IG only

**Disadvantages of decision tree:**

**1. Over fitting:** Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning

4. **<u>Ensemble of decision tree</u>**:
   We know that disadvantage of decision tree is ***overfitting***

   ***Why overfitting?***
   Because decision tries to mug up to homogenous classification and uses only 1 dataset for training

   Technique like *pruning* and setting *maximum depth of tree* is used to prevent overfitting as follows:

   ***1. Limit tree depth:***
   The most simple way to prevent overfitting is to limit the maximum depth of tree

   ***2. Pruning:***
   Pruning is the technique to reduce the size of tree by removing branches or sub tree which do not contribute more in improving the accuracy of prediction

   ***3. Cross validation:***
   Cross validation is a technique to split the dataset into subset of k fold where we train the data on kn folds and test the data on remaining folds.

   ***4. Ensemble methods:***
   Using ensemble method like gradient boosting and random forest can help to reduce overfitting.
   This methods create multiple decision tree and combine their predictions which help to generalize the model.

**Various ensemble techniques as follows:**

**1. Bagging: (Bootstrap + Aggregation)**

Bagging stands for bootstrap aggregation

***Bootstrapping***: bootstrapping involve creating a multiple subset from the original dataset. This subset are created randomly by sampling with replacement.
Sampling with replacement is nothing but the subset may contain more than 1 common data points

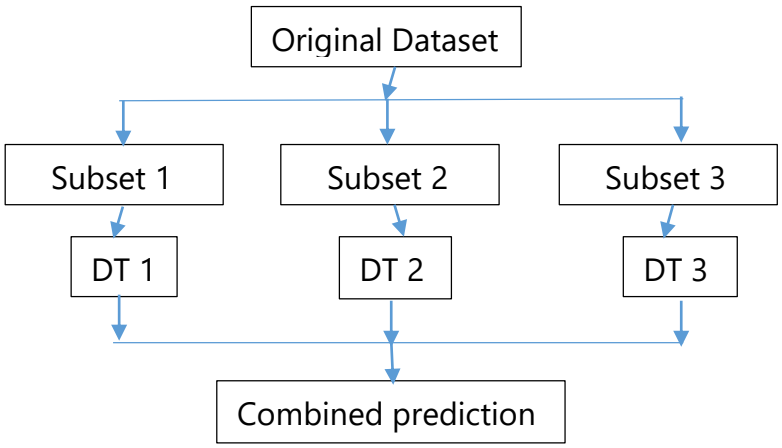
Number of column of subset = number of column of original dataset
Number of rows of subset <= number of rows of original dataset.

Original data set contains m column and N rows then
Subset should contain m columns and n' rows where
n'<= n

***Aggregation***: in bootstrapping we have created multiple subset from the original dataset.
Now in aggregation we are creating individual decision tree for each subset and combine the prediction of each decision tree by using mean, median, and mode

For categorical values ------> mode
For numerical values -------> mean

***IMP Note***:
Higher number of subset are always better
***Variance*** of combine prediction ***reduce to 1/n of original variance***



***1.1 Random forest***:

Random forest is a special case of bootstrap aggregation or bagging where the subsets are created by ***random combination of rows and columns both***.

Number of column of subset < number of column of original dataset.
Number of rows of subset <= number of rows of original dataset

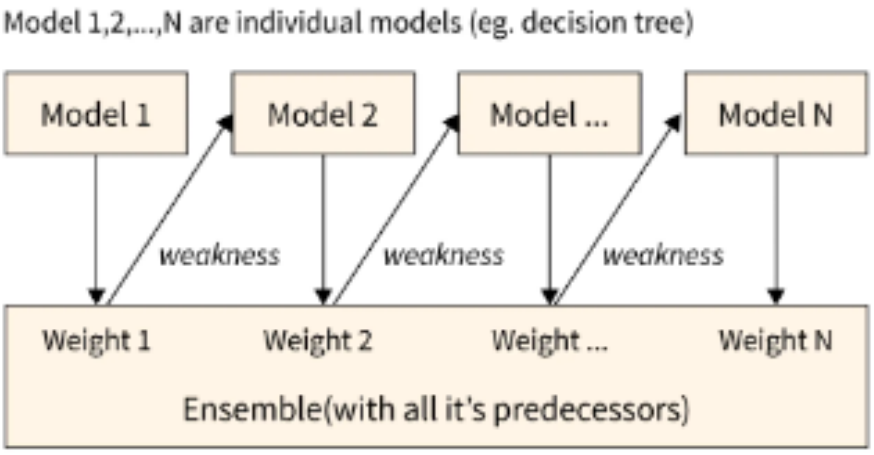
***IMP note:***
1. The subset may contain more than 1 common datapoint in it.
2. The subset should not contain any common column or feature in it.
3. Random forest gives better result in case of classification problem rather than regression problem

**2. Boosting:**
Boosting is an ensemble technique where ***decision tree learns from the mistake*** and improve the predictive accuracy of model by reducing error at each iteration.
It ***finds week learner*** from previous iteration and provide ***more weightage to week learners*** in the next iteration
And tries to minimize the error.

***There are many boosting algorithms as follows***
1. XG boost
2. Ada boost
3. Light boost
4. Cat boost

***XG Boost: -***
1. XG boost is an optimized implementation of gradient boosting that ***enhance performance and speed by allowing parallel computation***.
2. XG boost incorporates regularization techniques to prevent overfitting
3. XG boost allows ***tree to grow till depth and then prune it*** to improve predictive accuracy of model
4. It also automatically ***handles missing value*** during training without requiring explicit imputation.

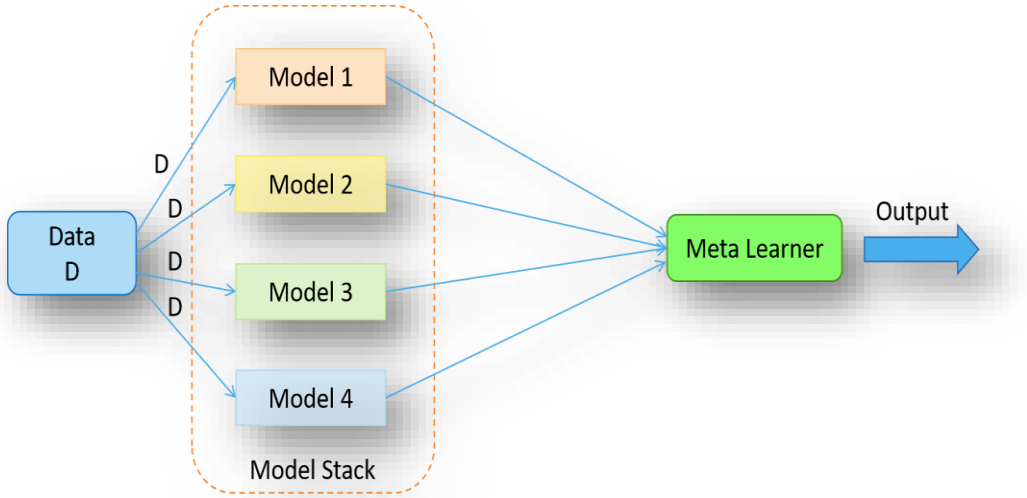
- **Ensemble Method:**
  Boosting, like bagging, is an ensemble method, meaning it combines the predictions of multiple models (learners) to make a final prediction.
- **Sequential Training:**
  Unlike bagging, which trains models independently, boosting trains models sequentially. Each new model is trained to correct the errors of the previous model(s).
- **Focus on Misclassified Instances:**
  Boosting assigns weights to training instances, giving more weight to those that were previously misclassified. This allows subsequent models to focus on the difficult cases, leading to improved accuracy.
- **Weak Learners to Strong Learners:**
  The goal of boosting is to transform a collection of weak learners (models that perform only slightly better than random guessing) into a strong learner (a model with high accuracy).
- **Common Boosting Algorithms:**
  Popular boosting algorithms include AdaBoost (Adaptive Boosting), Gradient Boosting, and XGBoost.
- **Benefits:**
  Boosting can significantly improve the accuracy and predictive power of machine learning models, especially for complex problems.
- **Reducing Bias:**
  Boosting is particularly effective at reducing bias in machine learning model



**3.Stacking:**

Stacking in machine learning, also known as stacked generalization, is an ensemble technique that combines predictions from multiple base models to improve overall predictive accuracy by training a meta-model on the predictions of the base models

- **How it works:**
    1. **Train Base Models:** Train multiple different machine learning models on the training data.
    2. **Make Predictions:** Use the trained base models to make predictions on the training data (or a validation set).
    3. **Train Meta-Model:** Use the predictions from the base models as input to train a meta-model (e.g., logistic regression, linear regression).
    4. **Final Prediction:** Use the meta-model to make the final prediction on new, unseen data.



## 5. K- Nearest neighbours:

1. K nearest neighbours is a supervised learning algorithm used for ***both classification and regression problems***. Where task is to classify unknown or unseen datapoint based on nearest datapoints using KNN.

2. ***K nearest Neighbours*** is used for a ***classification problem*** where it uses ***majority voting principle*** to classify the unseen data point

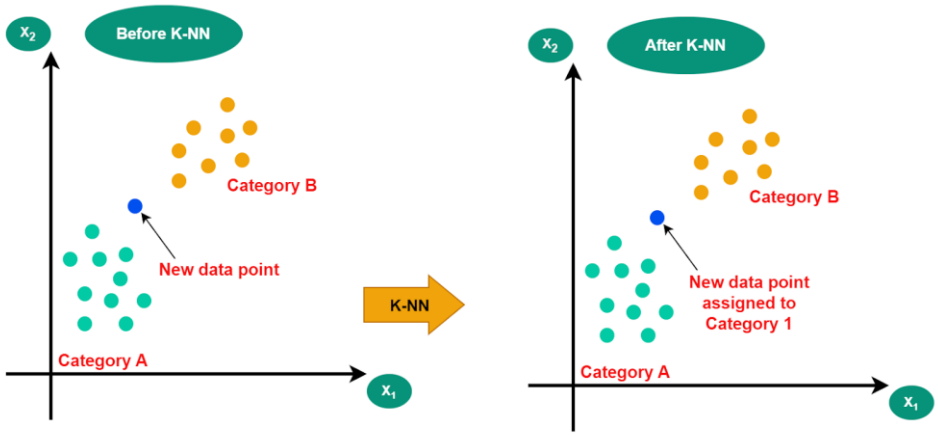
***For example***:
If datapoint is surrounded by the 5 datapoints out of which 3 of type A and 2 of type B then the datapoint is classify as a type A class.

3. ***K nearest Regressor*** is used for a ***regression problem*** where it uses ***mean or average*** of the surrounded datapoints to predict the value of unknown or unseen datapoint.

***For example:***
If datapoint is surrounded by 5 datapoints which values are [10, 5, 7, 3, 2] then the value of unknown datapoint is average of all the values
That is (10+5+7+3+2) / 5 = 5.44

4. the most used or preferred value ***of K is 5***
To calculate optimum value of K we can use gridserchcv or randomsearchcv

***IMP:***
***AS the value of K increases the bias increases and variance decreases***



## 6. Naive Bayes classifier:

1. Naive Bayes classifier is a supervised learning algorithm based on ***Bayes theorem.***
2. As the name suggest it is only used for the ***classification problem***.
3. It is a probabilistic models that predict probability of class labels based on the probabilities of the features given the classes.

***Bayes theorem:-***

P (A/B) = P (A) * P (B/A) / P (B)

Where P (B) != zero

P (label/feature) = P (label) * P (feature/label) / P(feature)

Let say we are calculating the probability of yes label then

P (label = yes / feature) = P (label = yes) * P (feature1/label= yes) * P (feature2/label= yes) * P (feature3/label= yes)

7. **Support vector machine**:

1. ***Support vector machine*** is a supervised learning algorithm used for both ***classification and regression*** problem
2. ***Support vector machine*** can ***handle High dimensional and complex data***
3. High-dimensional data refers to datasets where the number of features (or variables) is significantly larger than the number of observations (or samples).

4. Concept of SVM is to ***classify the datapoints*** using a plane called as ***hyperplane***.
5. The hyperplane which is having ***the maximum (margin)*** from both side is the ***best hyperplane***

***What is margin?***

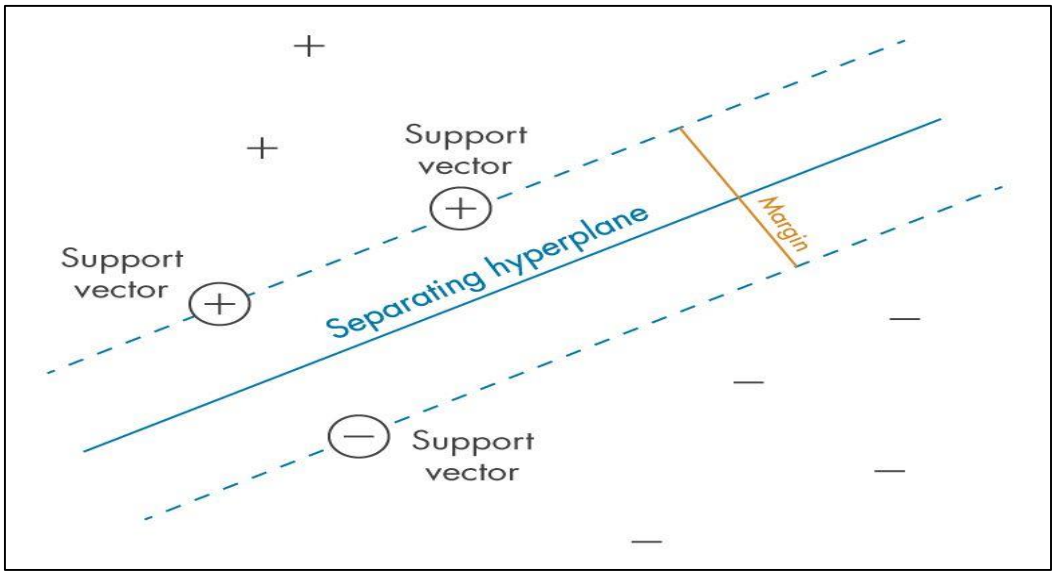
Margin is nothing but the distance between hyperplane and nearest data points.

When the Line of margin passes through the nearest points then these points are called as support vectors.

**Type of support vector machine:**

***1. Linear Support Vector machine:***

if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.




Here the datapoints are classified using a single straight line hence called as linear support vector machine classifier.

***2. Non-linear Support Vector Machine:***

If a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

***Case No 01***: ***When datapoints are radially distributed then we need to increase the dimension from 2D to 3D***

***Use radial basis function (RBF)***

r= e^ (- γ (x'2 + y'2))

Where

x and y are the co-ordinates of the datapoints

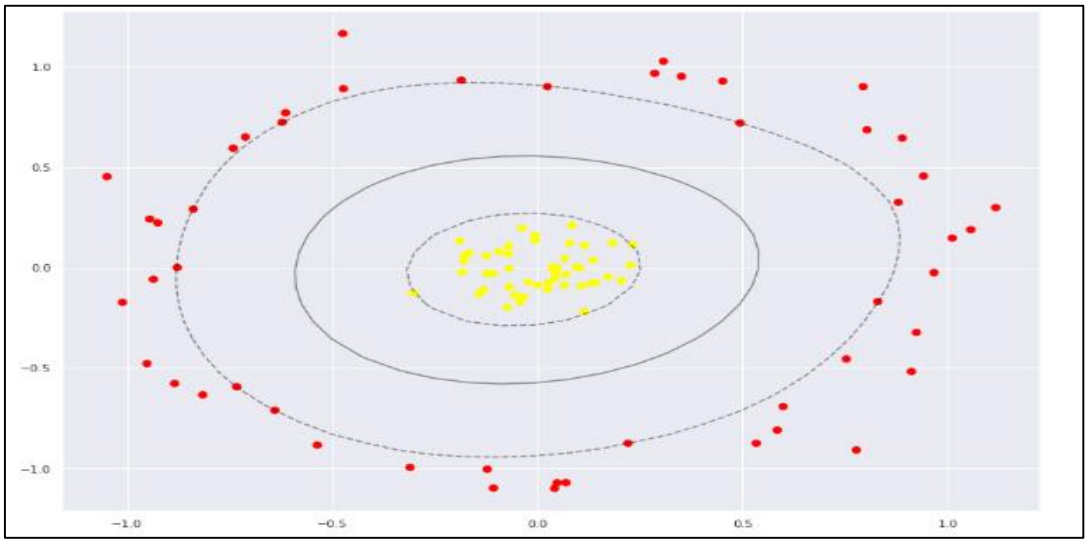
γ = γ (gamma) is a hyperparameter that controls the "width" of the RBF

***Gamma (γ) is very important.***

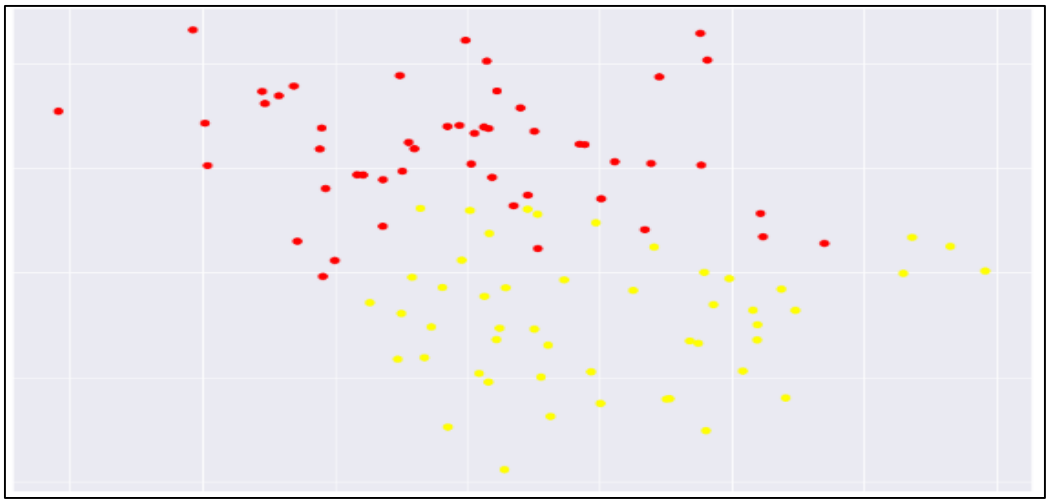
1. A **small gamma** makes the RBF kernel more localized. The decision boundary will be more ***sensitive to individual data points***. This can lead to ***overfitting*** if gamma is too small.
2. A **large gamma** makes the RBF kernel smoother. The decision boundary will be less sensitive to individual data points. This can lead to ***underfitting*** if gamma is too large.

In Scikit-Learn, we can apply kernelized SVM simply by changing our linear kernel to an RBF (radial basis function) kernel, using the kernel model hyperparameter:

clf **=** SVC(**kernel=****'rbf'**, C**=**1E6)



***Case No 02: when the datapoints are completely mixed in such as case tuning of SVM technique is used.***

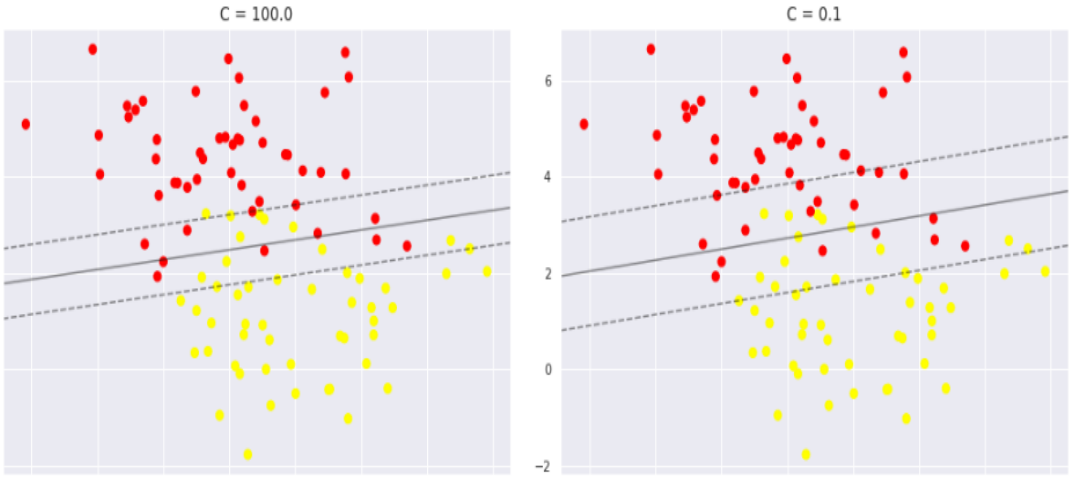


In this technique we are ***softening a margin*** and allowing some datapoints to cross the Hyperplane.
Hence we can say that
The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

***Parameter C is used for the tuning of margin***

***Large value of C*** --------> margin is very hard and ***datapoints not allowed to cross the hyperplane***. In this case we chances of overfitting increases

***Less value of C*** ---------> margin is softer and ***datapoints are allowed to cross the hyperplane***. In this case chances of underfitting increases.



To calculate optimal value of C, use cross validation and hyperparameter tuning technique such as GridSearchCV

## 8. Neural Network:

1. Neural network is a supervised learning algorithm used for both classification and regression task
2. It is a representation of human brain and essentially made to recognize patterns

***Perceptron:***

Perceptron is a functional building block of neural network which contains

***1. Input layer:*** The input layer is the first layer in a neural network. It contains neurons which represents input features. So basically
The number of neurons in the input layer equals the number of features in the dataset (these neuron don't have weight and bias)
Here we feed input features to neurons in the input layer. No computation happens here.

***Neurons:*** Neurons are the basic computational units of a neural network. They receive inputs (which can be features from the input layer or outputs from previous layers), apply weights and biases, and produce an output through an activation function. (These neurons are in the hidden layer if any and output layer).

***2. Activation function:*** Activation functions allow neural networks to model non-linear relationships between inputs and outputs. Also called transfer fn. Without them, the network would only perform linear transformations, limiting its ability to learn complex patterns in real-world data.
Activation function applies to output of neuron after computing the weighted sum.

***There are various activation functions such as:***
***Sigmoid function***
***SoftMax***
***RLU: Rectified linear unit***
***RELU: Exponential linear unit***

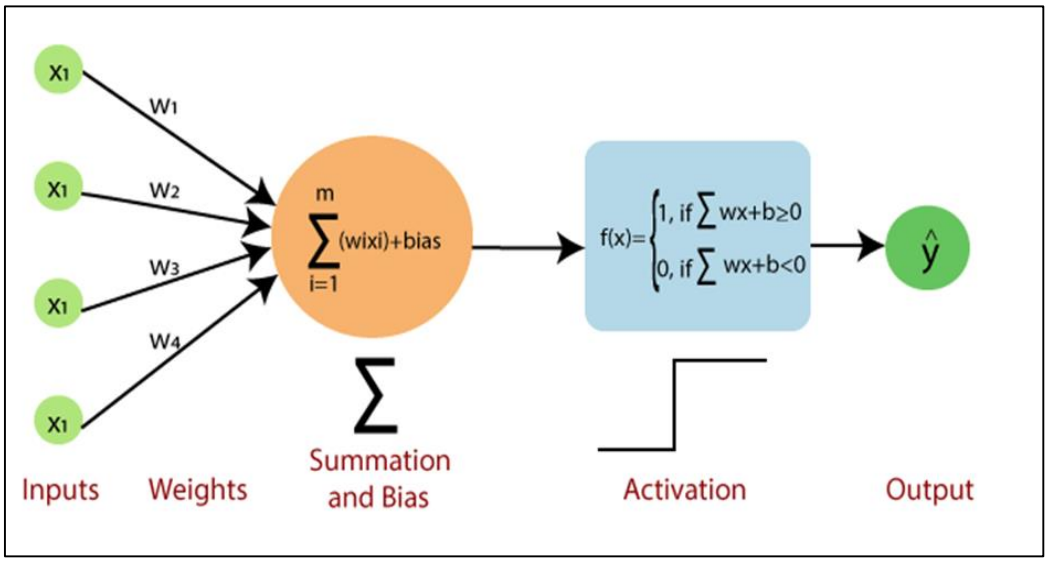
***IMP note:***
***Activation function is there for classification problem only other wise for regression problem linear function is sufficient***

Different activation functions constrain the output of neurons to specific ranges.

For example, the **sigmoid function** outputs values between **0 and 1**, making it suitable for binary classification tasks,
While the SoftMax function outputs probabilities for multi-class classification

***3. Output layer:*** Output layer contains final Output after the processed information

***Basically perceptron has only two layers input and output layer along with activation function.***




Neural network contains number of perceptron in a series
So data refines number of times according to number of perceptron's

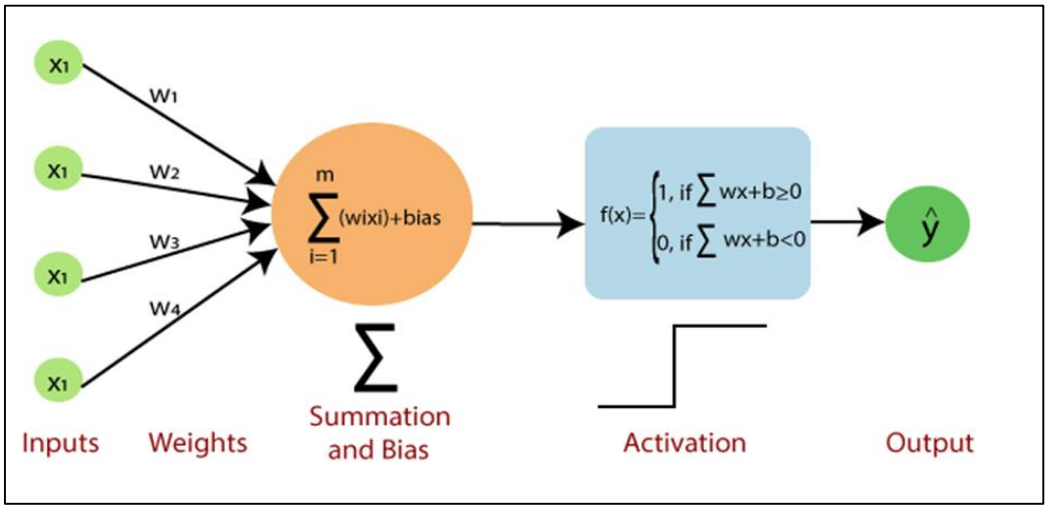
If
Number of perceptron's = 1000
Then data refines 1000 times
Weights are also refines 1000 times

Which Increase accuracy

**Working of neural network:**

***1. Forward propagation***: process of going from input layer to output layer

***Inputs (neurons) -------> Weights and bias term ---------> activation function ----------> Output***

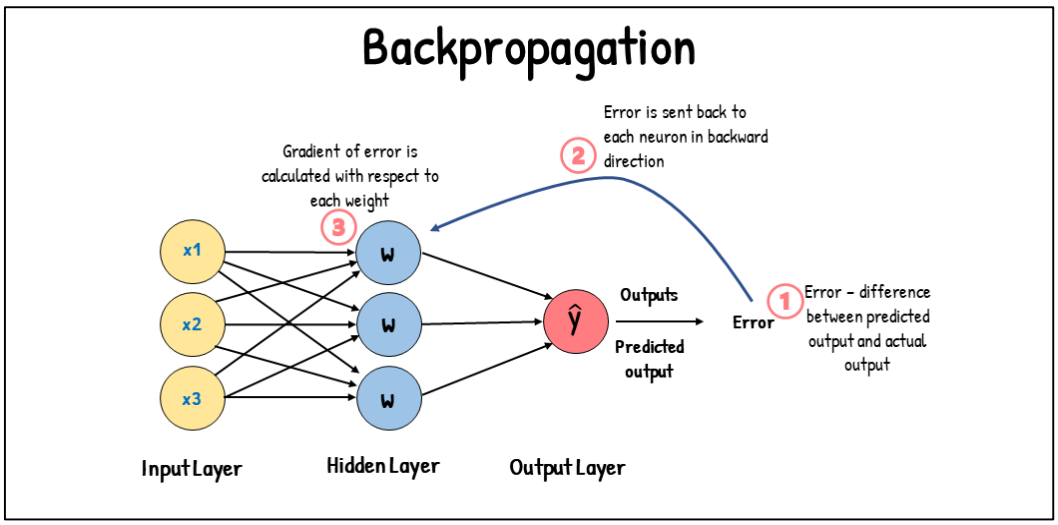



***2. Backward propagation:***

Back propagation is the crucial process in neural networks that allows the network to learn from its mistakes.

After forward propagation generates an output, the error (loss) is calculated using a predefined loss function.

The error is then propagated ***backwards*** through the network, layer by layer and Apply gradient descent to minimize the loss function or error by changing the weights and the cycle continues till the error goes to minimum possible and then we can calculate final output.



***#Artificial Neural network:***
Similar to perceptron except they have one or more extra layers called as ***hidden layers***.
Hidden layer contains the weight matrix. Weights are initialize and back propagation or adjustment of weight happens inside the hidden layer itself.

***Convolutional Neural Networks (CNNs):*** Specialized for image processing.
***Recurrent Neural Networks (RNNs):*** Designed for sequential data (text, time series).

ANN might contains more than one hidden layers

Learning process:
Model learn parameters through the process of forward propagation of data and back propagation of errors

Random weight and bias is assume ---→ hidden layer modifies the input till it reaches to output ----→ output is compared to expected output ---→ error is recorded and then propagated backward through the layers to adjust the parameters ----→ process continues

9. **Handling class imbalance:**

Sometimes the dataset contains unequal data of the different classes

***For example:***

If the dataset contains 1000 rows then out of 1000 rows 980 data points of class A and only 20 data points of Class B

This is the case of imbalance dataset. Training of such dataset is very difficult as there might be chances that they ignored the minority class datapoints

***There are various techniques to handle class imbalance as follows:***

***1. Undersampling:***

Randomly removing the samples from the majority class.

The number of samples remove should be equal to number of samples in the minority class

***Example:***

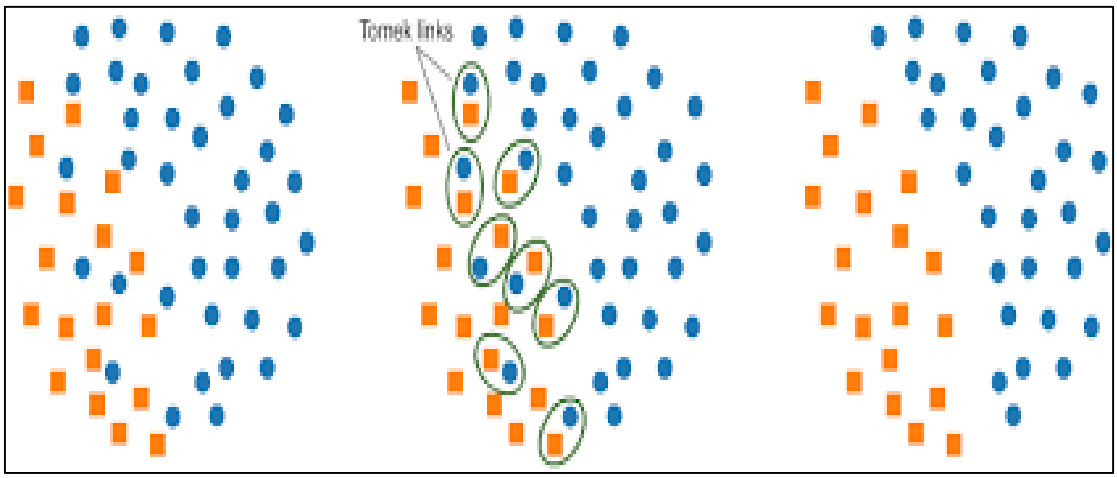
If majority class contains 1000 datapoints and minority class contains 5o datapoints then we can remove 50 datapoints from the majority class

Due to under sampling there is chances of missing data so underfitting may be happens.

***Tomek link technique is used for undersampling***.

***Tomek Link:***

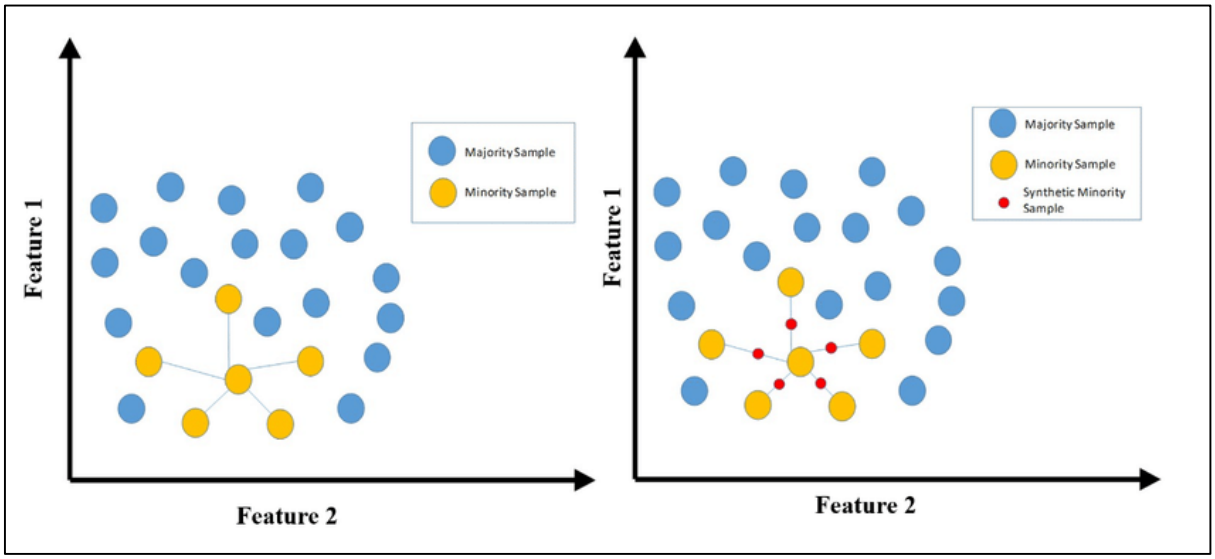
Tomek link is a pair of datapoints of opposite classes which are very close to each other and removing the datapoint of majority class from each pair.



***2. Oversampling:***

Randomly adding the duplicate samples in the minority class such that
***Number of majority samples = number of minority samples***
Duplicate samples are taking from minority class only.
Due to oversampling there is chance of overfitting
***SMOTE Technique is used for oversampling***

***SMOTE (Synthetic Minority Over-sampling Technique):***
It generate synthetic datapoints for minority class. The number of datapoints generated is equal to number of datapoints in the majority class




***How datapoints are generated?***
1. Randomly select any data point from minority class
2. Calculate K nearest neighbours
3. Choose any neighbour and place synthetic data point anywhere in the line joining the point

Repeat the process until data is balanced

Here in above diagram we can say that we have generated 5 datapoints of minority class by using SMOTE technique.

***Advantage of SMOTE:***
Avoids simply duplicating existing minority class data points, which can lead to overfitting