Important Tips:

CDF = Cumulative Distribution Function.

PPF = Probability Point Function.

**Hypothesis Testing :-**

[Colab Notebook](https://colab.research.google.com/drive/1acBk4ho5b6lFTmsV5Y2s9AhCqlr3gJja?usp=sharing)

* T-test & Z-test

-We use Z-test when we have (σ) population std deviation, and sample size > 30.

Z-value = (X - µ)/(σ/√n)

-We use T-test when we don’t have (σ), but we have (S) sample std deviation.

T-value = (X - µ)/(S/√n)

* When we have Condition of :

H0 = Something

Ha ≠ Something -- Then we use 2-tail test

& When

H0 ≥ Something

Ha ≤ Something -- Then we use 1-tail test

* For 2-tail test

p-value = p/2

σ = σ/2

* Compare p-value with α (significance level)

If P < α -> Reject null hypothesis(Ho)

If P > α -> Reject alternative hypothesis.

Hypothesis Testing Python Code:

import scipy.stats as stats

p\_value = stats.ttest\_ind(df['col'], df['col'], alternative = 'two-sided')

* Anova Test :-

-Anova test is mainly used to check the means of two or more groups are statistically same or different.

-It is used on numerical data.

-In this Q-Q plot is used to check weather the distribution is normal or not.

import statsmodels.api as sm

sm.qqplot(df['col'], line = 'q')

#line "q" is 45 degree line.

-Python code:

import scipy.stats as stats

test\_statistics, p\_value = stats.f\_oneway(df.iloc[:,0],df.iloc[:,1],df.iloc[:,2],df.iloc[:,3])

* Chi-Squared :-

-It is used to check weather two or more variables are statistically independent or not

-It is also known as Test of independence.

-It is used mainly for Categorical data

-In this contingency table is used to convert observed values into expected values.

-Python code:

import scipy.stats as stats

stat, p, dof, exp = stats.chi2\_contingency(table)

statistics, p\_value = stats.chisquare(observed, expected, ddof = 3)

**Regression:-**

ols - Ordinary List Square method.

* Simple Linear Regression(SLR):-

-Used for determining relation between dependent and independent variables

-It has only one independent variable(Y).

-Equation:- Y = βo + β1 X + ε

-r-squared value should be more than 0.85, It is stated by thumb rule

-r-square is Coefficient of Determination.

-Homoscedasticity & Heteroscedasticity is process of checking pattern in scatter plot

-Data transformation is used only when data is skewed or not normal

-F-statistic should be higher - 500+

-log-likelihood should be close to 0

-Python code:

[Colab Notebook](https://colab.research.google.com/drive/1f3KjwhSbfFe2tM-B2QpSwoYqmvOqDYUC?usp=sharing)

import statsmodels.formula.api as smf

model = smf.ols('Y ~ X', data=df).fit()

-> #Data transformation on model is like

model1 = smf.ols('np.sqrt(Y)~np.sqrt(X)', data = df).fit()

Model2 = smf.ols('np.cbrt(Y)~np.cbrt(X)', data = df).fit()

Model3 = smf.ols('np.log(Y)~np.log(X)', data = df).fit()

-> Select model with least root mean squared error for predictions:-

model1\_pred\_y = np.square(model1.predict(df['X']))

model2\_pred\_y = pow(model2.predict(df['X']),3)

model3\_pred\_y = np.exp(model3.predict(df['X']))

model = np.sqrt(model.mse\_resid)

model1\_rmse = np.sqrt(mean\_squared\_error(df['Y'], model1\_pred\_y))

model2\_rmse = np.sqrt(mean\_squared\_error(df['Y'], model2\_pred\_y))

model3\_rmse = np.sqrt(mean\_squared\_error(df['Y'], model3\_pred\_y))

-> During predictions normalize the data from log to exp.:

predicted = pd.DataFrame(np.exp(model3.predict(predicted.X)))

* Multi Linear Regression:-

[\*\*Colab Notebook\*\*](https://colab.research.google.com/drive/1JWOlta1fYP7HotIRK2hPCUbVSdRx42gu?usp=sharing)

-It has one dependent and multiple independent variables.

-We include only those variables in model who has p-value less than 0.05

-VIF (Varience Inflation Factor) should be less than 20.

-r-squared is used to measure of goodness.

-Correlation (r) of variables:

r = 0.1 to 0.3 - weak corr

r = 0.4 to 0.7 - moderate corr

r = 0.8 to 1.0 - strong corr

-Outlier Detection methods:-

-Boxplot

-Cook's Distance

-Leverage Cut-off = 3\*((k+1)/n) using STEM plot

-Outlier Treatment methods:-

-Data Transformation using (log, sqrt, cbrt)

-Mean-Imputation and Median-Imputation to handle outlier

-Normal Distribution check methods:-

-Q-Q plot

-Shapiro-Wilk test

-Standard Scaler scales the values such that mean is 0 and std dev is 1.

-However Standard Scaler should be used only when distribution is normal, if not convert distribution to normal using different methods.

-For building multi linear regression model there are assumptions regarding the dataset:-

-Feature should be independent of each other there shouldn't be any dependency upon each other

-There shouldn't be any other relation but Linear relation amongst model parameters (Hyperparameters of the model the intercept and coefficient)

-Each Feature and Model Error (residuals) should be independent of each other

-Constant Variance (Homoscedasticity) in Error, it should have Normal / Gaussian distribution~N(0,1) and indenpendently and identically distributed.

-There should be a linear relation between the dependent variable and Independent variables.

* Logistic Regression:-

[\*\*Colab Notebook\*\*](https://colab.research.google.com/drive/1FSvlns0kfjyaItCjPNGjEoR_ebavTyX_?usp=sharing)

[\*\*Colab Notebook version-2\*\*](https://colab.research.google.com/drive/1Yl2xfMMOzi4zzIcPZp1amt5NqlWFsiE1?usp=sharing)

-Logistic regression is used to predict categorical data of dependent variable

-It is used for solution of classification problems.

-Methods to check model accuracy:-

1. Confusion Matrix

2. ROC-curve

**Clustering:-**

[\*\*Colab Notebook\_Crime\_dataset\*\*](https://colab.research.google.com/drive/1jtRLLgDwIQA9UMGgD41m_aY7Ni4JJ-WX?usp=sharing)

[\*\*Colab\_Notebook\_Airlines\_Dataset\*\*](https://colab.research.google.com/drive/1ONE0I_ko6_NZ0SVG1bklbAOsx5uLqg3s?usp=sharing)

-Clustering is unsupervised-ML technique in Data mining.

-Cluster analysis(data segmentation) is exploratory method for identifying homogenious groups(clusters) of record.

-Similar records should belong to same cluster.

-Euclidean distance method is used to find distance between two points or records.

* Hierarchical Clustering:-

-In this algorithm we develop the hierarchy of clusters, which results in inverted tree from, which is known as **dendrogram**.

-Quick Notes:

-linkage{“ward”, “complete”, “average”, “single”}, default=”ward” Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

-ward minimizes the variance of the clusters being merged.

-average uses the average of the distances of each observation of the two sets.

-complete or maximum linkage uses the maximum distances between all observations of the two sets.

-single uses the minimum of the distances between all observations of the two sets.

* KMeans Clustering:-

-KMeans clustering is iterative algorithm that divides dataset into K different clusters in such a way that each dataset belongs only one group that has similar properties.

-K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

-The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid. The first step in k-means is to pick the number of clusters, k. Then keep the iterations running until one of the below satisfies:

-Centroids of newly formed clusters do not change

-Points remain in the same cluster

-Maximum number of iterations are reached.

-In K Means clustering we need to first assign the K number of clusters which can be taken up using the thumb rule of "Sqrt(n)/2" or plotting the Elbow curve, aka Scree plot.

-**Silhouette Score:-**

-The silhouette analysis measures how well an observation is clustered and it estimates the average distance between clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighbouring clusters.

* DBSCAN Clustering:-

-Density Based Spatial Clustering of Application with Noise.

-It is a density based clustering method that discovers clusters of non-spherical shape.

-It handles Noise better than other clustering algorithms.

-DBSCAN Parameters:-

-**Epsilon**(ε) :- It defines the size and borders of each neighbourhood.

-**MinPoints** :- It is density threshold, minimum number of points clustered together for a region to be considered dense.

**PCA :-**

[\*\*Colab\_Notebook\*\*](https://colab.research.google.com/drive/1k2kHKIxEpO-7cJAIdlEM3zTeani7xt8i?usp=sharing)

-Principle Component Analysis.

-It is used to reduce dimensions of datasets.

-It is used to :-

-Identify relation between columns.

-Reduce no of columns.

-Visualize dataset in 2D.

-PCA works by finding a low dimensional subspace that maximizes the variance of the data in that subspace and performing a linear projection.

-This basically means the data will be as spread out as possible, without changing the relationship between the data points. This allows us to find patterns or identify clusters if any exist.

-**Variance** :- It is the measure of spread of data in dataset with mean(x)

**Association Rules:-**

[\*\*Colab\_Notebook\_BookDataset\*\*](https://colab.research.google.com/drive/1PO52G48elBZ7BLyyfSSqMlldWd5PPP5v?usp=sharing)

[\*\*Colab\_Notebook\_MoviesDataset\*\*](https://colab.research.google.com/drive/1jBk2aMfkTmqJTS3k5EnVo7tGWod80zKd?usp=sharing)

-It is the study of association between two products.

-In simple words study of what goes with what, also called as market basket analysis.

-Antecedent -> Consequent [support, confidence]

-if Antecedent is product1 p1 and Consequent is product2 p2, then

-If p1 is purchased then customer will also purchase p2

-Terminology in Association rules:-

-**Support** :-Support is an indication of how frequently the itemset appears in dataset.

Support = no of transactions with both antecedents and consequent item sets / total transactions.

-**Confidence** :- confidence is an indication of how often the rule has been found true.

Confidence = no of transactions with both antecedents and consequent item sets / no of transaction with antecedents item set.

-**Lift ratio** :- The ratio of observed support to that expected if x and y were independent

Lift = confidence / benchmark confidence

-lift ratio more than 1 is better.

-Apriori Algorithm:-

-apriori() function evaluates support value for each product

-association\_rules() function help us to understand relationship between antecedent and consequent products

-apriori algorithm finds the most frequent itemsets or elements in a transaction database and identifies association rules between items.

**Recommendation System :-**

[\*\*Colab\_NoteBook\*\*](https://colab.research.google.com/drive/1ZZfSJ9schjUWpDb3zoMcGr8_U6egAJn7?usp=sharing)

-A recommender system, or a recommendation system, is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications.

-Mainly three types of recommendation systems in machine learning based on filtering are used to suggest product and services to the consumers.

-Content Filtering

-Collaborative Filtering

-Hybrid Filtering

-**Cosine Similarity** :-

-Cosine similarity score is independent of magnitude, and easy and fast.

-cos(A,B) = A.B/|A|\*|B|

**Text Mining (NLP) :-**

**Natural Language Processing.**

-The research on how to make computers understand and manipulate natural languages, draws from several fields including computer science, maths, linguistics, and neuroscience and the resulting interdisciplinary area of research is called NLP.

-It is estimated that 80% of worlds data is in unstructured form.

-Corpus :- Refers to collection of documents to be processed / analysed.

**-N-grams :-** N-grams are the combination of multiple words used together. Ngrams with N=1 are called unigrams. Similarly, bigrams (N=2), trigrams (N=3) and so on can also be used.

-Text Transformation techniques :-

**-Tokenization :-** Tokenization is breaking the sentence or document into chunks called tokens, each token carries semantic meaning associated with it.

**-Normalization of words :-** Transform various forms of the same term into common normalized form

lower\_words=[Text.lower() for Text in no\_stop\_tokens]

Eg- Apple, apple, APPLE -> apple

**-Stop-Words :-** Stop words are those words that (on their own) do not carry any meaning

from nltk.corpus import stopwords

stop = stopwords.words('english')

Eg- a, is, the, this, an, or, in, be, etc.

**-Lemmatization :-** Lemmatization refers to use of a vocabulary and morphological analysis of words, aiming to return the base or dictionary form of a word, which is known as lemma.

import spacy

nlp=spacy.load('en\_core\_web\_sm')

doc=nlp(' '.join(lower\_words))

lemmas=[token.lemma\_ for token in doc]

Eg- argue, argued, argues, arguing -> argue.

**-Stemming :-** Stemming is a crude heuristic process that chops off the ends of words without considering linguistic meaning of words.

from nltk.stem import PorterStemmer

ps=PorterStemmer()

stemmed\_tokens=[ps.stem(word) for word in lower\_words]

Eg- argue, argued, argues, arguing -> argu.

**Naive Bayes :-**

[\*\*Colab\_Notebook\*\*](https://colab.research.google.com/drive/1PAQABxtT14d8YiCpmoYM-tEXPyDPBTkq?usp=sharing)

-It is used for classification model building tasks/problems.

-Cardinality :- The number of labels within categorical variable is known as cardinality.

-Naïve bayes classifier uses the bayes theorem to predict the results of classification problem.

-Types of Naïve Bayes algorithm:-

-Gaussian Naïve Bayes:-

-Multinomial Naïve Bayes

-Categorical Naïve Bayes

**Confusion Matrix:-**

->A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give us a clear picture of classification model performance and the types of errors produced by the model.

->It gives us a summary of correct and incorrect predictions broken down by each category. The summary is represented in a tabular form.

->Four types of outcomes are possible while evaluating a classification model performance. These four outcomes are described below:-

-True Positives (TP) – True Positives occur when we predict an observation belongs to a certain class and the observation actually belongs to that class.

-True Negatives (TN) – True Negatives occur when we predict an observation does not belong to a certain class and the observation actually does not belong to that class.

-False Positives (FP) – False Positives occur when we predict an observation belongs to a certain class but the observation actually does not belong to that class. This type of error is called Type I error.

-False Negatives (FN) – False Negatives occur when we predict an observation does not belong to a certain class but the observation actually belongs to that class. This is a very serious error and it is called Type II error.

**Accuracy :-** (TP + TN) / float(TP + TN + FP + FN)

Classification error :- (FP + FN) / float(TP + TN + FP + FN)

**Recall:-** When the actual value is positive, how often is prediction is correct?

recall = TP / float(FN + TP)

**Specificity:** When the actual value is negative, how often is prediction is correct?

specificity = TN / (TN + FP)

**Precision:-** When a positive value is predicted, how often is prediction is correct?

precision = TP / float(TP + FP)

**F1-Score :-**

f1-score is the weighted harmonic mean of precision and recall. The best possible f1-score would be 1.0 and the worst would be 0.0. f1-score is the harmonic mean of precision and recall. So, f1-score is always lower than accuracy measures as they embed precision and recall into their computation. The weighted average of f1-score should be used to compare classifier models, not global accuracy.

**KNN(K-Nearest Neighbour):-**

[\*\*Colab NoteBook\_ZooDataset\*\*](https://colab.research.google.com/drive/1YqtNTmYciCrk5_YRR9SQAOwuzmAyTqDm?usp=sharing)

[\*\*Colab\_NoteBook\_GlassDataset\*\*](https://colab.research.google.com/drive/12Cmz9zuxMK3baeqdW2xO9DoRWMpmT3eG?usp=sharing)

-It is a machine learning algorithm used for both classification and regression problems.

-It predicts the label or value of new data point by considering its closest neighbour in the training dataset.

-It is supervised machine learning algorithm

-Grid search is used to choose value of K.

-It is recommended to choose the odd value of K.

**Decision Tree :-**

->Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems.

->But mostly it is used for classification problems.

->In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

->In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

**Terminologies:-**

**Root Nodes –** It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

**Decision Nodes –** the nodes we get after splitting the root nodes are called Decision Node

Leaf Nodes – Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

**Branch/Sub-tree –** just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.

**Pruning –** is nothing but cutting down some nodes to stop overfitting.

**Splitting -** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Pruning - Pruning is the process of removing the unwanted branches from the tree

**Parent/Child node -** The root node of the tree is called the parent node, and other nodes are called the child nodes.

**When to Stop Splitting :-**

->We can tackle this problem using hyperparameter tuning.

->We can set the maximum depth of our decision tree using the **max\_depth** parameter.

->The more the value of max\_depth, the more complex your tree will be. The training error will off-course decrease if we increase the max\_depth value but when our test data comes into the picture, we will get a very bad accuracy. Hence you need a value that will not overfit as well as underfit our data and for this, you can use GridSearchCV.

->Another way is to set the minimum number of samples for each spilt. It is denoted by **min\_samples\_split**. Here we specify the minimum number of samples required to do a spilt. For example, we can use a minimum of 10 samples to reach a decision. That means if a node has less than 10 samples then using this parameter, we can stop the further splitting of this node and make it a leaf node.

There are more hyperparameters such as :

**min\_samples\_leaf –** represents the minimum number of samples required to be in the leaf node. The more you increase the number, the more is the possibility of overfitting.

**max\_features –** it helps us decide what number of features to consider when looking for the best split.

* Pruning:-

->It is another method that can help us avoid overfitting. It helps in improving the performance of the tree by cutting the nodes or sub-nodes which are not significant. It removes the branches which have very low importance.

->The performance of a tree can be further increased by pruning. It involves removing the branches that make use of features having low importance. This way, we reduce the complexity of tree, and thus increasing its predictive power by reducing overfitting.

->There are mainly 2 ways for pruning:-

**(i)** **Pre-pruning** – we can stop growing the tree earlier, which means we can prune/remove/cut a node if it has low importance while growing the tree.

**(ii)** **Post-pruning** – once our tree is built to its depth, we can start pruning the nodes based on their significance.

* CART method:-

(Classification And Regression Tree)

->It is used for numerical data as well as categorical data.

->CART uses Gini-impurity method for classification purpose.

->CART in Regression cases uses 'Least squares' method.

->Also used for multi output problems.

->Overfitting and Underfitting is most usual problems in Decision tree.

Gini Impurity/Index:-

->Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.

->An attribute with the low Gini index should be preferred as compared to the high Gini index.

->It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

->Gini index can be calculated using the below formula:

Gini Index= 1- ∑jPj2

* C5.0 :-

-> Used only for classification problems.

->It uses 'entropy' concept borrowed from information theory.

->Dataset with high entropy are very diverse datasets.

Entropy:-

->It is measured in bits.

->for 'n' classes, entropy ranges from 0 to 1.

->minimum entropy value indicates that sample is homogenous.

->Where maximum value indicates diverse data

\*->Zero entropy means pure regions

Information gain:-

->To use the entropy to determine the optimal feature to split upon, the algorithm calculates the change in homogeneity that would result form 0 split on each possible feature, which is a measure known as Information Gain.

->Information gain for feature 'F' is calculated as the difference between entropy in segment before split(s1) and partitions resulting from split(s2).

Information Gain(F) = Entropy(s1) - Entropy(S2)

F - feature

S1 - entropy before split

S2 - entropy after split

**Random Forest :-**

[\*\*Colab\_Notebook\_CompanyData\*\*](https://colab.research.google.com/drive/1RDtUEjqib9EPSoWvwpHte5wlU41FwWj_?usp=sharing)

* It is a Bagging based method (Bagging - bootstrap Aggregation).
* Random forest is a Supervised Machine Learning Algorithm that is 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.
* The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
* 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.
* It uses Gini-Impurity method
* Gini importance(or mean decrease impurity), which is computed from the Random Forest structure. Let’s look how the Random Forest is constructed. It is a set of Decision Trees. Each Decision Tree is a set of internal nodes and leaves. In the internal node, the selected feature is used to make decision how to divide the data set into two separate sets with similar responses within.

Random forest algorithm parameters:

* **n\_estimators:-** The number of trees in the forest.
* **criterion:-** The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.
* **max\_depth:-** The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.
* **min\_samples\_split:-** The minimum number of samples required to split an internal node.
* **min\_samples\_leaf:-** The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves least min\_samples\_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.
* **min\_weight\_fraction\_leaf:-** The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample\_weight is not provided.
* **max\_features:-** The number of features to consider when looking for the best split.
* **max\_leaf\_nodes:-** Grow a tree with max\_leaf\_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity. If None then unlimited number of leaf nodes.
* **min\_impurity\_decrease:-** A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
* **min\_impurity\_split:-** Threshold for early stopping in tree growth. A node will split if its impurity is above the threshold, otherwise it is a leaf.
* **bootstrap:-** Whether bootstrap samples are used when building trees. If False, the whole datset is used to build each tree.
* **oob\_score:-** Whether to use out-of-bag samples to estimate the generalization accuracy.

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

**Neural Networks :-**

[\*\*Colab\_NoteBook\_GasTurbine\*\*](https://colab.research.google.com/drive/1AGRN8RAOwfFQh2NgLpmEbzI7ei7bnnJw?usp=sharing)

* These are also known as Artificial neural networks or Simulated neural networks(SNN)
* Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.
* Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer.
* Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.
* Artificial neuron has weighted inputs, Threshold values, activation function and output.
* There are different activation functions :- (sigmoid, tanh, relu, elu, leaky relu, maxout etc).

Terminology in NN :-

* **Loss Function:-** Loss / Error function used for back propagation.
* **Batch Size:-** The number of training examples in one forward/backward pass, Higher the batch size the more memory space is needed.
* **Iterations:-** No of passes, each pass in batch size passed to the network how many times is Iterations. [one pass = one forward pass + one backward pass].
* **Epoch:-** When every record in training data set is been used in network in spite of iterations, then it is a Epoch.
* **Learning rate(η):-** Size of step in direction of the negative gradient.

Perceptron:-

* Perceptron is an artificial version of neuron
* It calculates the weighted sum of input values.
* It is used to draw a decision boundary
* Sometime one decision boundary is not enough to form pure regions so at that time we go for "Feed forward networks".

Feed Forward Network:-

* Perceptron has input and output layers, middle layer has no connection with external world hence called as hidden layers
* Each perceptron in one layer is connected to every perceptron in another layer hence information is constantly feed-forward, and this network is called feed forward network
* There is no connection among perceptron in same layer.

Back Propagation :-

* Back propagation is widely used algorithm for training feedforward neural networks.
* It computes gradient of loss function with respect to networks weight using 'chain rule'.

Hyperparameters :-

* When we use neural network to solve a problem then we come to a confusion for several parameters as how they should be used, such as:-

-Number of hidden layers

-Number of neurons in each layer

-Activation function

-Error / loss function

-Gradient descent methods

* These all are hyperparameters.
* So we have to empirically (trial & error) determine these parameters, which best suits to our problems solution.
* Gradient descent methods are used to update the networks weight, there are 3 methods

-Batch Gradient descent

-Stochastic gradient descent

-Mini-batch gradient descent

* End.