***"A breakthrough in Machine Learning would be worth ten Microsofts."*** *-*Bill Gates

**M**achine Learning is one of the most demanding field today which uses data to solve problems, yet we are unaware that mankind has been using machine learning for a long time such as surveys based on data collected from the past era to the present advanced computers.

Every day, when we search Google, ask Alexa or Siri on weather report, take a photo using face detection or even shopping or watching movie based on recommendation system is a part of machine learning where a computer programs are learning and improving on our data to cater our needs.

Machine learning can be classified into:

**Supervised Learning:** In this method, computer is given with labeled data and desired outputs. This is for the algorithm to learn from the actual output and its calculated output.

*Example*: Lion is labeled as animal and Rose is labeled as flower. A new data point will learn from this labeled data then will be appropriately classified.

**Unsupervised Learning:** In this method, computer is given with unlabeled data set to learn the patterns and find the desired output.

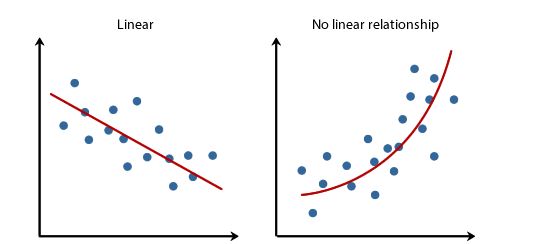
*Example:* A set of un-labeled photos of flowers will be given and algorithm will likely classify bunch of flowers together based on the features learned by the algorithm.

**Reinforcement Learning:** In this method, labeled data is not used instead a rewards system is used for learning.

*Example*: Frequently used in Gaming where proper leveling up in the game is positively rewarded and vice versa.

**Linear Regression**

Linear Regression is a method in which the relationship between a dependent variable and an independent variable are modeled and expressed in a linear form.



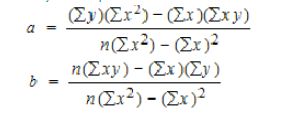
**Theory:** Here the dependent variable is ‘Y’ which is the target variable and the independent variable is ‘x’. ‘Linear’ signifies that when ‘x’ increases/decreases, then ‘Y’ also will change in a linear form. Goal is to fit a best fit line among the given data points.

Mathematically they are denoted as

**Y= ax+b**

In this equation, ‘a’ and ‘b’ are constant factors which are to be determined based on the dataset provided. Using ‘a’ and ‘b’ values, value of ‘Y’ is predicted for any given value of ‘x’. This process is known as Simple Linear Regression using only one independent ‘x’ value. If more than one independent values are used then it will be known as Multiple Linear Regression.

Mathematical formula for finding a and b:



General equation to find Multiple Linear Regression:

MLP

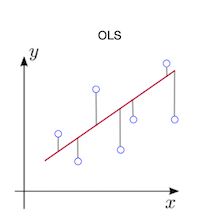
β0= Intercept

β1,β2, …βp-1 = regression coefficients

Yi= target variable

**Ordinary Least Squared**

This method is used for Multiple input in order to find the value of coefficients. OLS is used to minimize the sum of squared residuals.



In regression, the y-intercept line I.e in red color is predicted for the given dataset and distance is calculated from each data point to the regression line. Then the value is squared and added together. Aim here is to reduce this ordinary least squared to minimal in order to get a best fit line.

**Code:**

**Python**

Linear\_Reg= Linear Regression()

Linear\_Reg.fit(X\_train,y\_train)

**R**

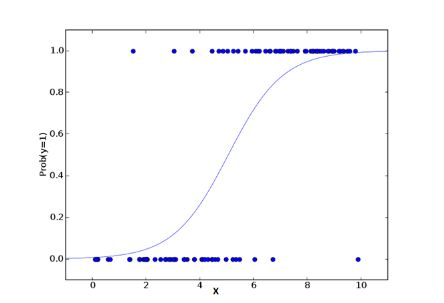
Linear\_Reg= lm(a~. , data = train\_data)

**Logistic Regression**

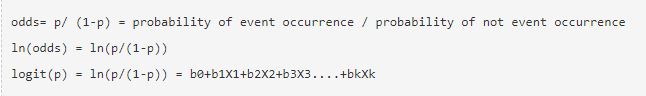
Logistic regression is one the basic classification algorithm to start in machine learning. ‘Regression’ in the name is misleading since it is used for classification problem and ‘logistic’ is derived from the ‘Logit function’ which is the base for this model.

In logistic regression, independent variables are continuous while dependent variable(target variable) is Boolean I.e True or False, Yes or No, 0 or 1. Practical examples such Cancer or not, Spam or not, Fresh or old etc.

Logistic function which is the base for Logistic regression is a Sigmoid function which takes any value between 0 and 1. Plotting the function will result in a **S** curve as shown below.

****  
Now deciding the Boolean output needs a boundary line known as Decision boundary which is based on conditional probability. Lets day **0.5** as boundary, then anything above is classified as **Positive** and anything below is classified as **Negative**.

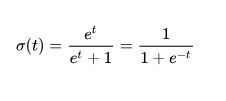
Here comes a probability concept where ‘odds’ is probability of an event occurring to event not occurring.



Log is taken in the above equation in order to remove the limits of zero to one into negative infinity to positive infinity leading to better and easy interpretation of the output.

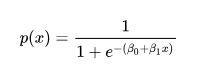
**Mathematical Illustration:**

**Logit Function**

Let’s take **t**  as a linear function with an univariate regression model then

LORRR

So substituting **t** in Logit function equation gives the **Logistic Regression Equation** as

Where β0= Intercept

β1= coefficients

**Code for Logistic Regression**

**Python:**

lr = LogisticRegression()

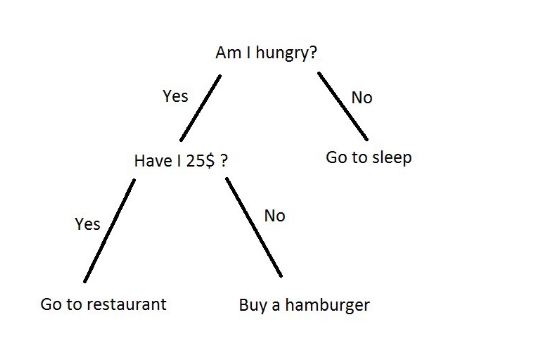
lr.fit(x\_train, y\_train)

**R:**

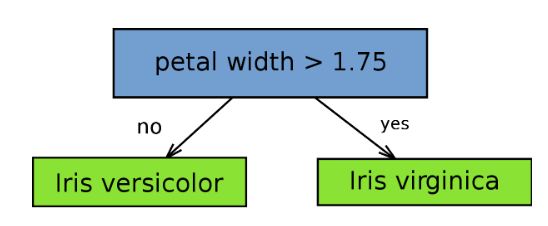
Logistic= glm(y\_train ~ . , data=data, family=’binomial’)

**Decision Tree**

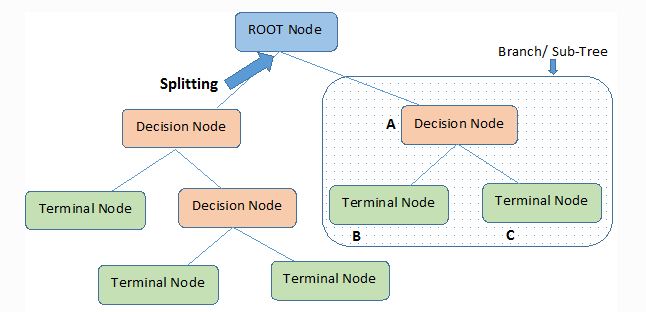
Decision tree is a supervised learning that is used for both regression and classification problems but mostly used for classification problem. The underlying principle of Decision tree is to split the population or sample into homogeneous multiple chunks based on certain cutoff values in the features. In a nutshell, Decision tree is a series of ‘if-else’ decision at each node leading to a more desirable output.



Example: Lets take the Iris dataset from Kaggle competition. When the petal width is greater than 1.75 than it is classified as *Iris virginica* or else it is classified as *Iris versicolor* based on the threshold of 1.75 as shown below.



The threshold to split a node into multiple sub nodes is based on several algorithms such as Gini, Chi-square, Entropy and so on which will be discussed later. Lets get first into basic terminology of Decision tree.



**Terminology:**

**Root Node**: Is the main node or whole population/sample.

**Decision Node**: Sub-node spitted into further sub-nodes.

**Splitting:** Dividing a node into sub-nodes.

**Terminal Node:** End of node splitting I.e : splitting terminated.

**Pruning**: removal of sub-nodes.

**Why Decision Tree is used ?**

Decision trees are easy to understand, less cleaning required since influence of outliers and missing values are minimal and most important is it can handle both categorical and numeric variable which is a huge benefit in time saving.

Even though Decision tree has few issues such as *Over fitting* which can be addressed by tuning parameters and pruning.

**Working of Decision Tree:**

Since terminal node is at the bottom of this model, it is a top down approach. In regression problem, a new point which is a dependent continuous variable at the terminal node is the mean value of the prediction while in classification problem, it will be based on mode value. Decision tree keeps splitting until user defined criteria or threshold is reached which is a hyper-parameter to be tuned by the user.

**When to stop splitting of tree ?**

This can be answered using different models such as:

**Gini:**

When two random points are selected from the population then they must be of same class and probability is 1 if they are from pure population. Higher the Gini value means higher the homogeneity and works only as binary splits.

Example: From above Iris data set example, finding which class it belongs to

i.e: *Iris virginica* or *Iris versicolor* is based on Gini score which is calculated as

Gini=(p^2 + q^2)

Where,

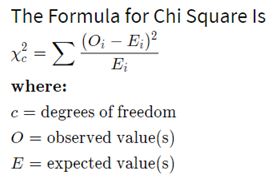
P= probability of success

Q= probability of failure

**Chi-Square:**

Is used to calculate the statistical significance between of difference between sub-node and parent node.

**Formula:**

Chi-square ranges from zero to infinity.

**Entropy**

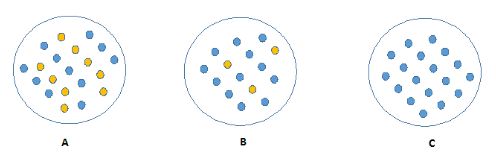
Entropy helps decision tree to decide a split



Information gain is measure of how much information a feature gives us about the class.

dt7

In a simpler words lets look at the image below:



Here seeing at these 3 figures, we can easily explain **C**  when compared to **A** and **B** because it is homogeneous when compared to other two and more information is needed to explain different colors.

Hence more homogeneous/ pure the node is, less information is needed to explain while more impure the node is, lot more information is needed to explain the node and this is *Entropy*.

**Parameters Tuning**

Minimum samples: Number of samples(minimum) required to split a node. It is used to curb Overfitting and under fitting by tuning properly.

Minimum samples for terminal node: Number of samples (minimum) required in a terminal node or leaf.

Maximum depth of tree: defines the vertical depth of the tree. Used to control over fitting since more the depth of the tree will leads to model learning very specific to given sample.

Maximum number of terminal nodes: can be defined in place of max\_depth.

Maximum features: Number of features considered while looking for best split in a node. Higher the value means Overfitting hence to be tuned as per requirement.

**Code**

Python:

DT = DecisionTreeClassifier()

DT.fit(X\_train, y\_train)

R:

library(rpart)

fit <- rpart(y\_train ~ ., data = x,method="class")

**Random Forest**

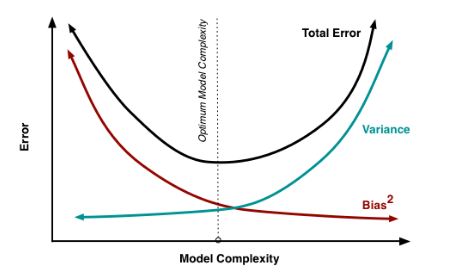
Random Forest is a method which grows multiple trees instead of single tree like Decision tree and extracts benefits of multiple forest to reach more accurate and desired output.

**Ensemble**

Ensemble is technique where a group of models are built in order to achieve better accuracy. This is the pinnacle for Random Forest since it builds many trees for better accuracy.

The con of this technique is like any other tree based model where bias and variance is suffered. Bias means, ‘how much on an average are the predicted values different from the actual value.’ Variance means, ‘how different will the predictions of the model be at the same point if different samples are taken from the same population’.

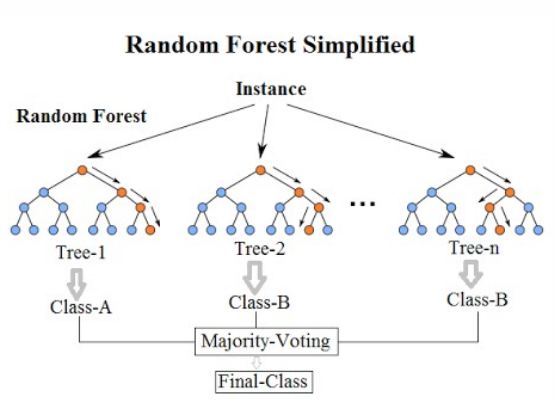
Now this issue of bias and variance can be handled using bias- variance trade-off.



The goal here is to maintain a balance between bias and variance. As from above figure, when complexity of model increases, prediction error reduces due to low bias but after a point it will lead to over fitting problem leading to high variance. A balance has to be achieved for a stable model.

In order to reduce Variance, **Bagging** is used in which variance is reduced by combining results of multiple models on multiple sub-samples of the same data. In regression, Mean is taken at the end to get the predicted output.

In a nutshell, Random Forest can be illustrated below.



Random Forest can be explained by first explaining its name. ***Random*** comes from randomness of the data it chooses. For example, it takes random rows and few random columns(features) to build multiple tree. This building of multiple tress gets its name***Forest.***

Once Random forest builds a series of decision trees, then a **majority voting** is taken if it is for classification or **Mean** is taken if it is for a regression problem while having low bias and variance leading better accuracy and stable model.

**Hyper parameter Tuning**

n\_estimators: number of trees in the foreset

max\_features: max number of features considered for splitting a node

max\_depth: max number of levels in each decision tree

min\_samples\_split: min number of data points placed in a node before the node is split

min\_samples\_leaf : min number of data points allowed in a leaf node

Bootstrap: method for sampling data points (with or without replacement)

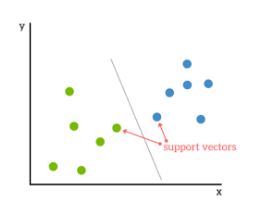
**Code**

RF = RandomForestClassifier()

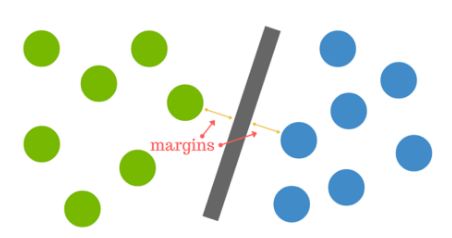
RF.fit(X\_train,y\_train)

**Support Vector Machine**

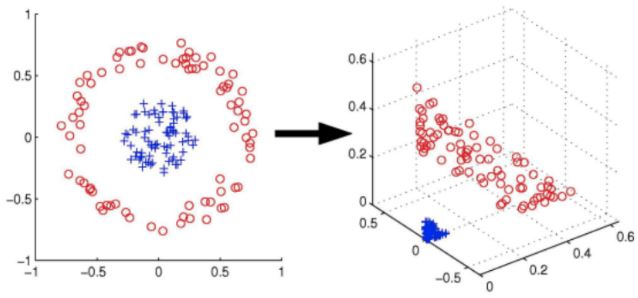
It is a supervised algorithm which can be used for both regression and classification but mostly used for classification. The main idea of SVM is to find a hyper plane which can divide the data given as accurate as possible.



A Hyper plane in a two features dataset is a simple line separating or classifying like above figure. In a dataset with more than two features would be represented by a plane(like a 3-D slab) separating the given dataset. Best fit hyper plane is one which has maximum margin. Margin is the distance between the hyper plane and the nearest point on either side of the plane.



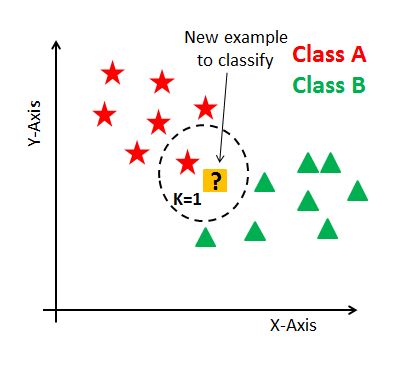
When the data is over lapping I.e green and blue dots are mixed, then we have to move away from 2-Dimension to higher dimensions like 3-Dimension. By moving to 3-D we have converted the line to plane which would help in better segregation of the data.

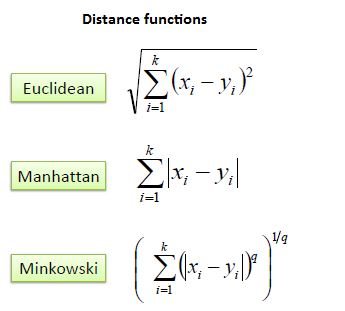


Even though SVM is accurate but it is not a good choice for larger dataset due to restriction of dimensions and their associated interpretation.

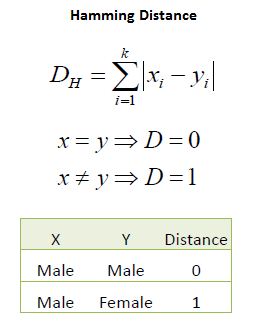
**k- Nearest Neighbors**

KNN is a simple algorithm which classifies data based on its nearest neighbors. KNN will classify given data into similar groups based on similarity with its nearest points. It is the crux of recommendation system semantic searching and anomaly detection.

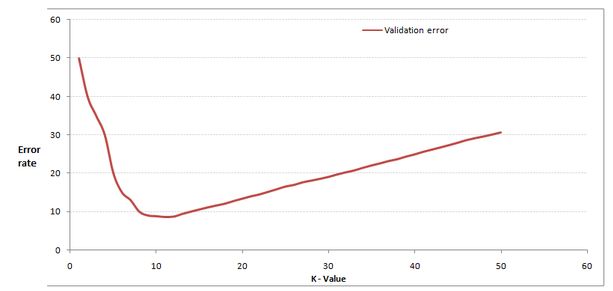
The way KNN determines its appropriate neighbor is using similarity measure such **Euclidean, Cosine Similarity, Manhattan, Minkowshi , Hamming** and so on.



The above distance functions are applicable only for continuous variables while **Hamming distance** is applicable for Categorical variables.



Now **K** value is a hyper parameter which has to be tuned for a optimum result. It can be determined using cross-validation and if **k** is small will leads to over fitting while higher values of **k** will lead to under fitting and hence proper **k** value is essential.



In the above image cross-validation is used and optimum k value is determined using **Elbow curve** which in the above example is **10.**

**Statistics for Data Science**

Statistics is the art of learning from data. It involves collecting data, analysis and arriving at a conclusion based on the data. Some of the examples of data collection such as survey, polling, voting, web scraping, cache etc.

**Population:** Is the total values of the data I.e complete data. Ex: Population of the country or world.

**Sample:** Sample is the subset of the population. It represents the population but not entirely since it was taken randomly. Ex: Testing samples in the laboratory.

The following concepts are used for imputation I.e. during missing values.

**Mean:** Average of all the values in the sample. It is used when the variable is continuous.

Mean <x> = ∑x/N

∑x = Sum of all the values

N= Total number of samples

**Median:** Is the middle/ central value of the sample which is an ordered form.It is used when the variable is continuous.

Median = (n+1)/2

**Mode:** Is the value which has occurred most number of times in the sample. Used for imputing categorical variable during imputation.

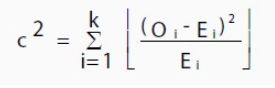
**Range:** Is the spread of the data in the sample = Max -Min. Range is used a data to find the range.

Ex: Range of ‘ID’ column gives the spread/ total count of data.

**Chi-square Test**

Chi-square test is used to show relationship between two categorical variables. It tells how much difference exists between those variables. A low value on the test signifies high correlation between two sets of data while high value on test signifies low correlation between data.

This can be used during classification problem where relation between target and independent (categorical) is to be found for implementing a model. Chi-square test gives p-value which tells if the result are significant or not.

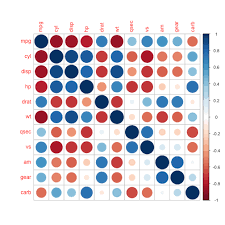
****

**O =** Observed value ; **E=** Expected value.

**Correlation**

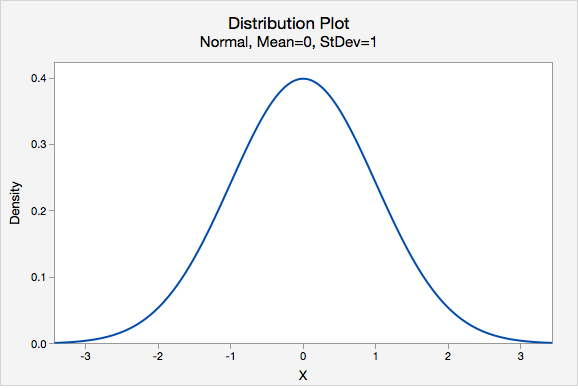
Like Chi-square test, correlation is used to find the relation between two numeric variables. Ex: In regression, relation between target variable (y) and the independent variable(x) can be found which can be either positively correlated I.e. when ‘x’ changes, ‘y’ also changes in the same direction

Same way, when ‘x; changes, ‘y’ changes in opposite direction. Knowing this relation between variables is very important in feature selection of the data. In python or R, we can use COR plot to find the correlation.



**Normal Distribution**

Normal distribution is also known as Gaussian distribution which is a probability function that defines how the data is distributed. It is a symmetric distribution where most of the values are accumulated around the center.



Normal distribution has two factors,

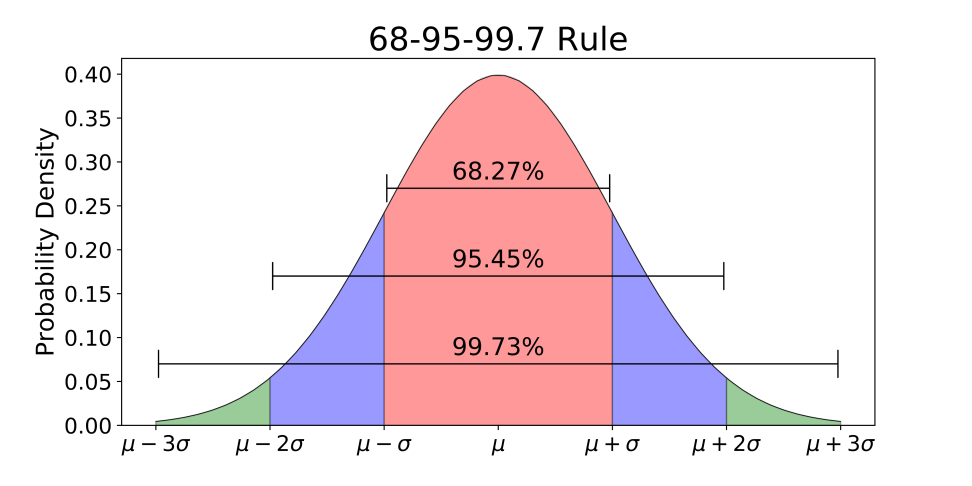
**Mean:** Is the average of complete data and in the above figure, mean is at the center and shift in the mean position will shift whole graph.

**Standard Deviation**: Is used to see variability of the data and in the above figure, it defines the width of the normal distribution. When compared to mean, SD tells how far away is a point from mean. Change in the SD will narrow or flatten the bell curve in the distribution.

Normal distribution has 68-95-99.7 rule which states that

68 % of data falls within 1-SD

95 % data falls within 2-SD

99.7@ data falls within 3-SD of the mean.

**Confusion Matrix**

Confusion matrix is a measure to check the performance of a classification problem. The total number of correct and wrong predictions are summarized.

**True Positive:** Is the total positive predicted which are actually positive.

**True Negative:** Is the total negative predicted which are actually negative.

**False Positive:** Is the total positive predicted which are actually negative.

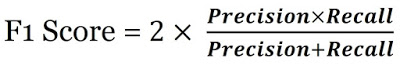
**False Negative:**Is the total negative predicted which are actually positive.

**Accuracy:** Is the overall accuracy of the confusion matrix.

**Recall:** Is the ratio of total positives correctly predicted over total number of positive examples.

**Precision:** Is the ratio of total positives correctly predicted over total number of predicted positive examples.

**F1 Score:** is the weighted average of recall and precision.



**Outliers**

Outliers are those values which deviates result from the actual result due to their presence in the data. There are two types of outliers: Univariate and multivariate.

*Univariate* : Can be found while looking at a distribution at single feature space.

*Multivariate*:Is found while looking at the distribution at multidimensional space.

Outliers can be detected in several ways. One of them is:

1. **Score:**

Standard score or Z -score is a metric which indicates how many standard deviation a data point is from the sample mean.

zscore

Where:

x= value

mean= mean

Sigma= standard deviation