**Decision Tree-**

This is a machine learning algorithm which is a tree based model and goes through a decisive structure for outputting anything , Given below is a picture of decision tree model where each circle is called as a root node and each square is called as leaf node where it can’t go more granular level to make a decision , we can also say the algorithm is based on conditional or a flow chart structure.

Decision Tree are called as the simplest machine learning algorithm, because it is very easy to visualize and interpret and the model complexity is also low

This is mostly used for both categorical and regression problem

The algorithm goes as follows

1.Place the best attribute of the dataset at the **root** of the tree.

2.Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

3.Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

So question here is how do we split the data based on what?

And the answer is that we split with two criteria one is entropy and second is GINI index

1. Gini index is used as the splitting criteria
2. The more the Gini index is the more it is homogeneous
3. It performs only binary split
4. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p²+q²).
5. Calculate Gini for split using weighted Gini score of each node of that split

Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% — 50%), it has entropy of one.

**Entropy can be calculated using formula:- Entropy = -p log2 p — q log2q**

Here p and q is probability of success and failure respectively in that node. Entropy is also used with categorical target variable. It chooses the split which has lowest entropy compared to parent node and other splits. The lesser the entropy, the better it is.

Information gain is 1- entropy

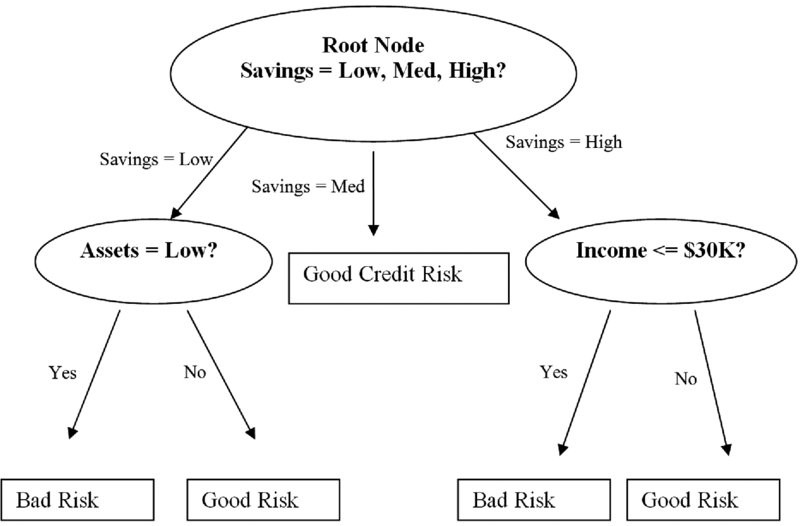
Calculate entropy of the parent node. Also calculate the entropy of individual node of the split and calculate weighted average of all the sub-nodes available in the split

Advantages and Disadvantages:

1. Decesssion tree are easy to understand.
2. It is easy to visualize

Disadvantage:-

1. It is prone to overfitting



**KNN-**

KNN stands for k nearest neighbor

KNN is a supervised algorithm which is use for both classification and regression problem

In KNN k stands for the k number of nearest neighbor , it does not make any assumption of underlying data on which it is being applied.

The algorithm basically predicts given there is a new point in the data where the data pint will lie looking at its nearest neighbor and it is based on feature similarity, this process of computation while predicting Is called as lazy execution hence it is called as a lazy model and is locally optimized

Referring to our example of friend circle in our new neighborhood. We select 3 neighbors that we want to be very close friends based on common hobbies. In this case K is 3.

For an example :-

We have age and experience(x values) in an organization along with the salaries(y values).

Problem statement:- We want to predict the salary of a new candidate whose age and experience is available.

Solution though KNN is

Step 1: **Choose a value for K**. K should be an odd number.

Step 2: **Find the distance of the new point to each of the training data**.

Step 3:**Find the K nearest neighbors to the new data point**.

Step 4: For classification, count the number of data points in each category among the k neighbors. **New data point will belong to class that has the most neighbors** which means the decision is made upon the feature similarity.

And if the problem statement is a regression, value for the **new data point will be the conditional mean of the k nearest neighbors**.

Now the next question is mind is how do you determine the value of K, the answer is quite simple K is a hyperparameter and we have to choose K on a trial and error method , there is now hard and fast rule to choose K.

Coming to the point how do you define nearest, To answer this question we can think about what nearest means, nearest means the distance between 2 points should be small, Now we have to define distance and distance can be calculated using various method

Distance can be

1. Euclidean distance
2. Manhattan distance
3. Hamming Distance

* Euclidean distance is the square root of the sum of squared distance between two points. It is also known as L2 norm.
* Manhattan distance is the sum of the absolute values of the differences between two points
* Hamming distance is used for cat
* categorical variables. In simple terms it tells us if the two categorical variables are same or not.

Advantages :-

* model complexity is low
* can be use for both regression and classification problem
* there is no assumption that must be consider for the given data like linear regression, in other ways model free algorithm

Disadvantages:-

* computational complexity is high
* can be suffered from overfitting if value of k is less also the overfitting problem can be solved by Wilson editing
* in KNN we have to hold all the data in ram, unlike linear regression, because every data to be fitted in a local function
* we can not do feature selection on the data

please Note:-

if the number of p are more than its advisable to have large number of data points to have a better result.

Also please be mindful that the computational complexity does not depend on K but depends highly on N and very little on P because , N is the number of data points and once a new data point enters ,we have to find out the N distance from which takes time than we have to sort them and that does take time, but once its done whether you take top 3 bottom 3 that doesn’t increase the computational complexity.

Also as the number of parameters grows in KNN then the concept of likelihood seems to fade away as the distance become high between two points and this is called curse of dimensionality

The improvised version of KNN is called condensed KNN where computation complexity will be reduced by keeping the model accuracy same . this includes categorization of points into different bucket like prototype point ,absorbed point and outlier, where proto type point is basically selected with hert’s algorithm,

**Support Vector Machine.**

SVMs are supervised learning models , Support vector machines (SVMs) are a type of linear classifier,

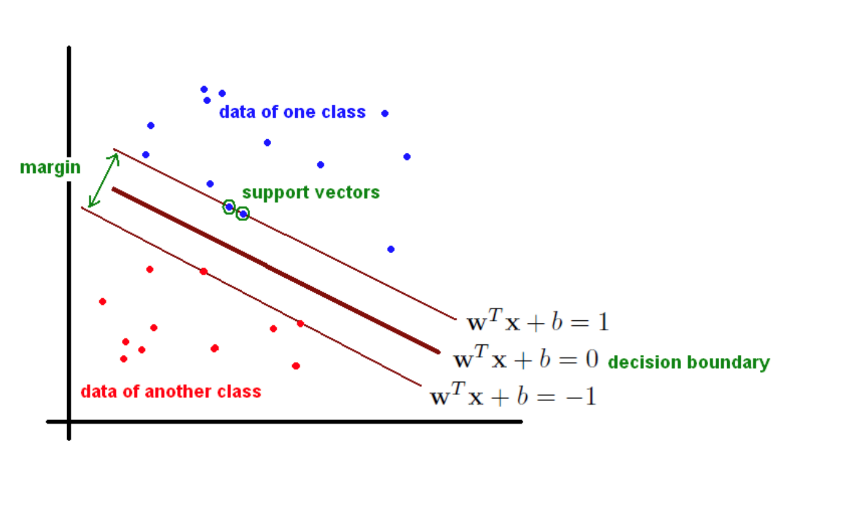
They are especially effective at **classification, numeral prediction**  SVMs find a line (or hyperplane in dimensions greater than 2) in between different classes of data such that the distance on either side of that line or hyperplane to the next-closest data points is maximized also called as margin maximization. which in turn helps the algorithm to differentiate between 2 homogeneous set of data. Also consider the separating boundary as a slab rather than a line for better understanding of margin.

SVM uses a method called dot product duality, on either side of the margin there is data points called as a support vector by which the margin line is being defined

Consider we have a set of negative and positive values in a two-dimensional space, along with an initial straight line between the two classes of data points, the circled points are called as support vectors(fig below)

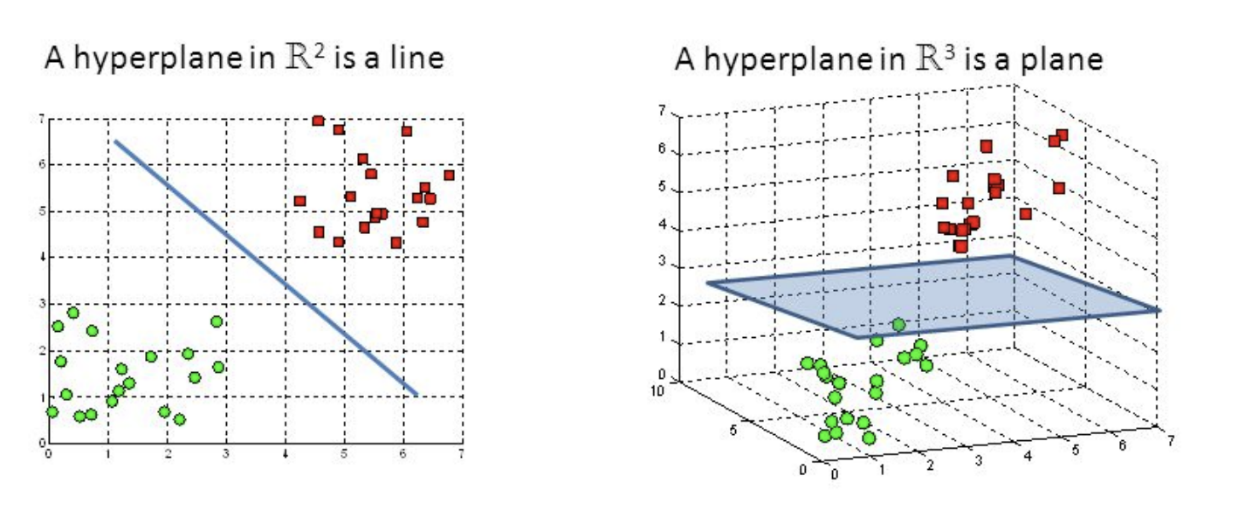
The SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM’s theory. Therefore, the optimal separating hyperplane maximizes the margin of the training data.

Now the task to find the optimal hyperplane is simple in the below case as the data points as linearly separable. But if the data points are not linearly separable In that case how do we find the optimal hyperplane?



In SVM, it is easy to have a linear hyper-plane between these two classes. SVM includes a technique called the [**kernel**](https://en.wikipedia.org/wiki/Kernel_method)**trick**. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear.

There are various types of kernel available in the SVM algorithm , few examples are Gussian kernel, polynomial kernel, rbf kernel.



Advantages:

* It works really well with clear margin of separation.
* It is effective in high dimensional spaces.
* It is effective in cases where number of dimensions is greater than the number of samples.
* It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Disadvantages:

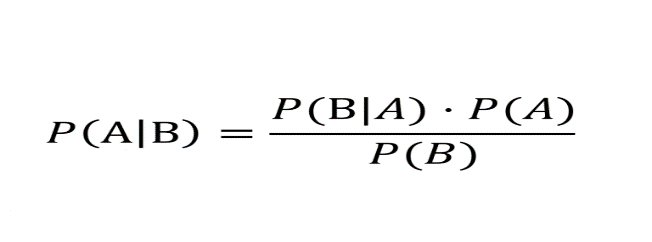
* It doesn’t perform well, when we have large data set because the required training time is higher
* It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping.

**Naïve Bayes-**

This is a supervised learning algorithm which is used for classification problems . The algorithm used Bayes theorem as it’s base to classify.

The Bayes theorem states that , given the probability for occurring of the event A given the probability of event B, what the probability of occurring B given the probability of A

The formula of the Bayes theorem is given below .



* Naïve Bayes assumes that all the features are independent
* The algorithm also work great with multi class classifier on a large data set.
* Even if Having model complexity simple, this algorithm performs exceptionally well
* And also Used for text mining