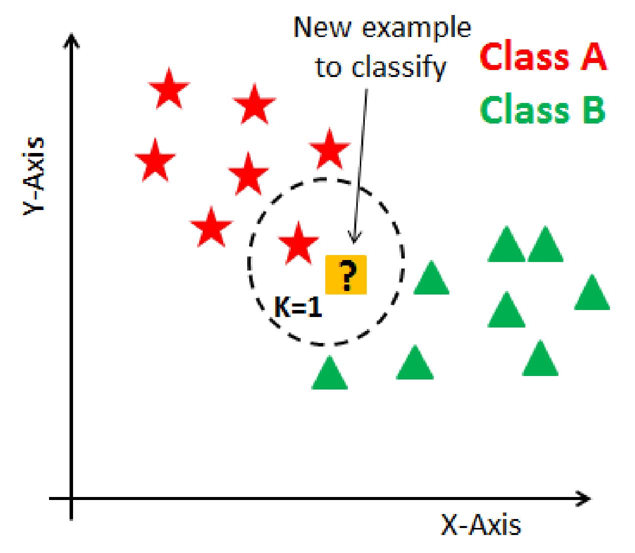
`KNN(K-nearest Neibour)` in Python.

- KNN is easy to understand,simple and versatile and one of the top most machine learning alogorithm.
- KNN used in variety of applications such as Finance,heathcare,political science,writting detection,image recognition and video recognition.
- In the finacial department predict the credit card rating of the customer,in the banking department while the loan disbustment predict whether this is going to safe or risk. In political science classifing potential voters in two classes i.e. will vote or won't vote.
- KNN algorithm use for the both regression and classification problems.
- KNN algorithm is based on the feature similarity approach

How does the KNN Algorithm work ?

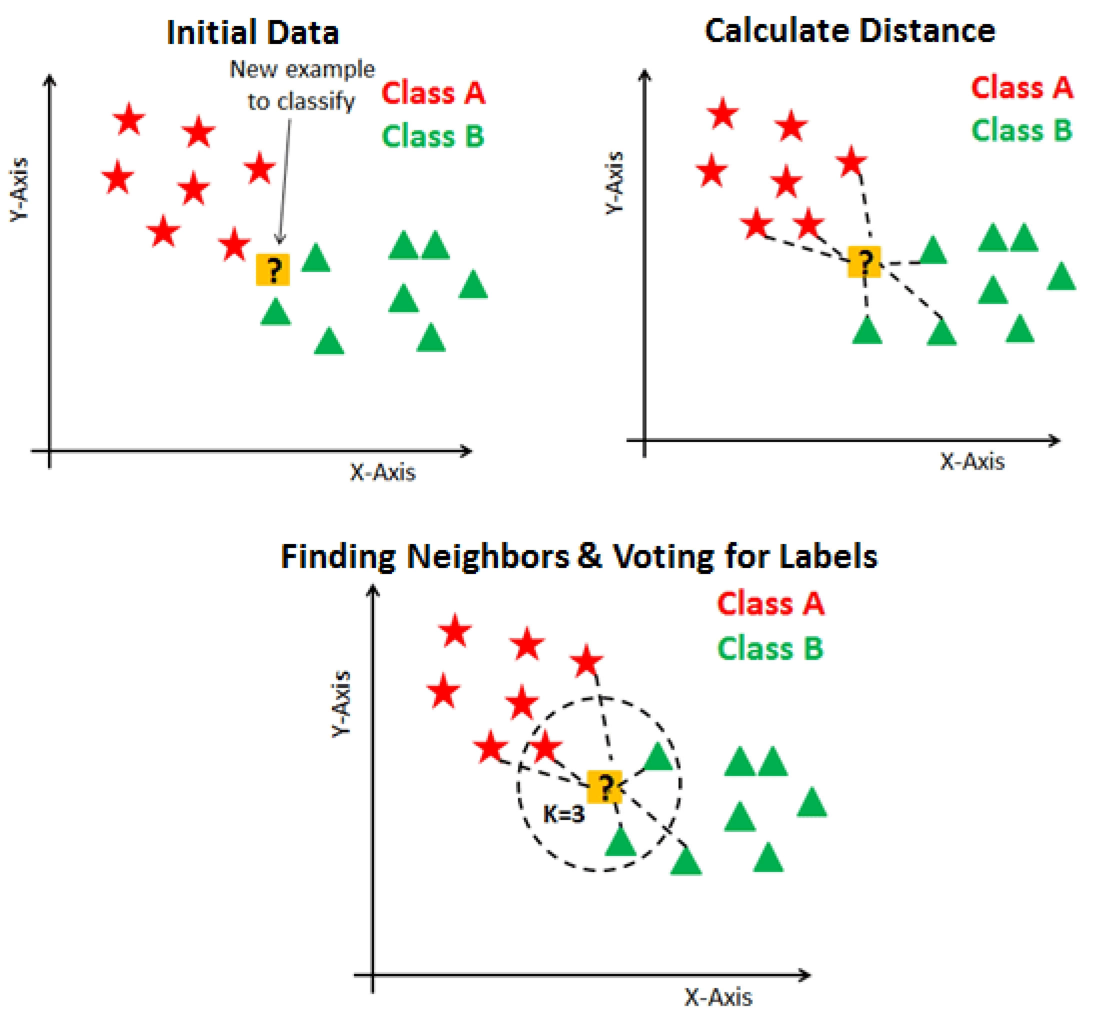


- In KNN Algorithm,
- **K is the number of nearest neighbors.**
- The numbers of neighbors is core deciding factor,K is generally an odd number if the number of classes is 2 or even.
- When the K=1,then the algorithm is known as nearest neighbor algorithm.This is the simplest case.Suppose we have `P1` is the data point inside the data and we want to classify that point or label that point with the help of prediction.
- First,it will find out the one closest data points from avaliable data to unseen P1 data point and then the label of the nearest point assigned to the P1 unseen data point.
- Let's take a example of the scatttered data points with some distribution in random fashion along with two categories.If algorithm got the some unseen data that time a unseen data points has to be classify according to which points are highly nearer from the avaliable points.
- Let suppose we have P1 unseen data point and we having two categories that are avaliable inside data and want to clasify that P1 point during this situation algorithm first find out how many data points from each categories are closed that P1 point by using some distance paramter called as euclidean distance or manhattan distance.If majority of data point from any category belong to that point that means the new P1 will be belong to particular category.

With an prictical example :-

- Let's suppose we having the customer list in that we consider the limit of tax those had not paid yet.If the people are not paid tax from last 1 yr then those people consider as defulter and if not then these people consider as geninue. That means there if any person come accross then we can say easily this person faulty.
- For fiinding closest similar points,you find the distance between the points uisng the **`distance measures such as Euclidean distance ,Hamming distance,Manhattam distance and Minkowski distance.`**
- KNN has following basic steps :-
  - Calculate the first distance.
  - Find the cloest neighbors.
  - Vote for labels.

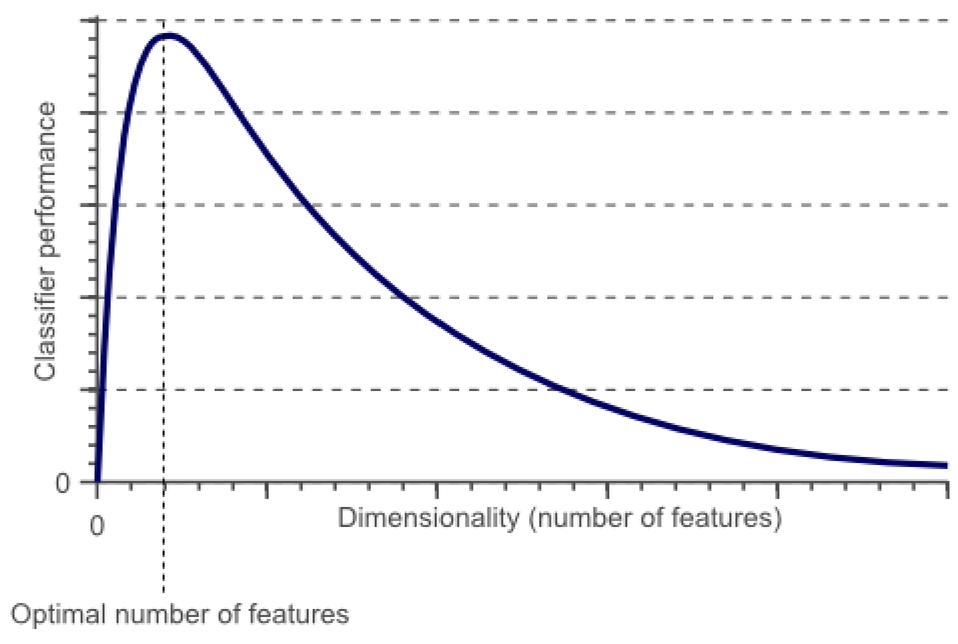
Steps Of The KNN



- **Calculate the first distance:-** Suppose we having the two types of class in that we having the point inbetween the class A and class B but we don't know whether this point belongs to the class A or Class B.So,that's why will calculate the distances from that point inbetween the class and try to comparision which is nearer it.
- **Find the closest nieghbors :-** Afterwards,we will find the majority of cloeset observation around that point.suppose that majority of cloest observation around that point which is of the class A then,
- **Vote for the Labels :-** Then,that class will label that point and consider as it's own contain.That's means it's giving it's label to that unknown point.Based on the majority of points surounding that unknown points it will decide that the avaliable point is belonging to that category.

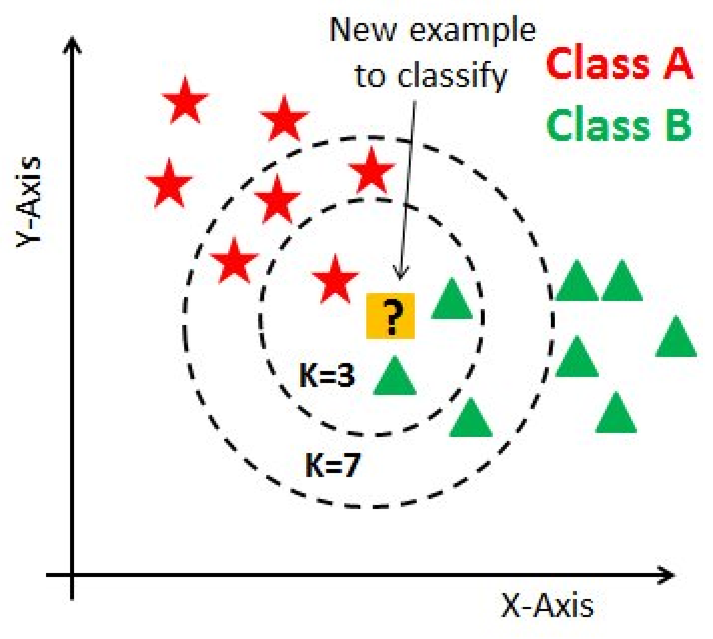
Curse Of Dimensionality

- **KNN perform better on the lower number of features than the large number of features.**
- We can say that when the number of features increses then it requires more data.increse in dimensions also leads to the problem of overfitting.
- At the some specific point increase in dimentions increasing the performance of the model but after the cartain points classifier performance get start decreasing.
- That is the case of overfitting and to avoid overfitting the requied data will need to grow exponentialy as we increase the number of the dimensions.
- This problem of higher dimensions wis known as "Curse of dimensionality"
- To deal with the problem of the curse of dimensionality,we need to perform the principle component analysis before apply any machine learning algorithm or we can also use feature selection approach.
- Research has shown that in large dimensions Euclidean distance is not usefull anymore.Therefore,we can prefer other measures such as cosine similarity,which get decidedly less affected by the high dimension.



How do we decided the number of neighbors in KNN ?

- Now,we will understand the KNN algorithm working mechanism.At this point the quentions arries that how to choose the optimal number of Neighbors and what are it's effect on classifier number of neighbors (K) in KNN is a hyperparameter that you need to choose at the time of modeling building.We can think of K as controlling variable for prediction model.
- Research has shown that no optimal number of neighbors suits all kind of datasets.Each datasets has it's own requirement.In the case of small number of neighbours,the noise will have higher influence on the result,and a large number of neighbors makes it computationally expensive.
- Research has also shown that a small number neighbors are most flexible fit which will have low bias and high variance and high number of neighbors will have a smoother decision boundary which means lower variance but high bias.
- So,It is always advisable to calculate mean squared error and performace of the model with the variable number of nearest neighbors of problem datasets and the make list and then plot that list.
- Once you make list and plot vs neighbours and performance your model then we can optimally choose your number of neighbours.
- There is elbow method is used to get the optimal number of K.


New example
to classify
Class A
Class B
Y-Axis
?
K=3
K=7
X-Axis