K -Nearest Neighbors

- K-NN is supervised machine learning classification and Regression algorithm.
- K-NN classifier works on majority of vote.
- K-NN Regression works on the Aveage of the vote
- In this kind of machine learning algorithm we will classify the number of classes acorrding to their similarities.
- In that core deciding memeber is the k value that is nothing but number of neighbors.\
- If we consider any new random input point and k=3 that means arround that point we will get the 3 neighbour point.
- If suppose we having data in that we contaning the N number of classes.If we add new Random input datapoint and tries to predict this in which class this point is belonging we cannot say simply this point is belongs to this this category.
- This will derived with the help of KNN algorithm and k values this will measure the eucladian distance from the new random datapoint to the avalible datapoint of each classes.
- We will consider the shortest distance of the new random input datapoint with respect to all among classes datapoint in order to predict the class of that particular random input datapoint.
- Out of the those distances which class datapoint give us shortest distance this will be predicted class with respect to new random datapoint.

- **Steps of the KNN Algorithm**

- Identify the data.
- Select the Target Variable
- Caculate the Optimal k value which can give us the better prediction.
- Find the the closest distance from the Input random datapoint to the closest class point.
- Vote that random input datapint as that particular closiest class name. ### About K

- K is the is value of number of neighbors around the new random input datapoint.

- K is generally an odd number if our number is even.It will help us to give a weight on the majority side.

Curse of dimensinality

- Dimensions of the data is number feature.
- KNN works well when we have low dimensions avalible in data
- If we more number feature that time should have more data in index position.
- If we increase the dimensions of the data it will leads to overfitting because it will confused the model to make prediction in our favour.
- Upto the some feature KNN model will give us the better peformance.
- But if we will increase the feature then it start decreasing the performace of model or classifier just because of the overfitting of model.
- To reduce the overfitting, our data should move expoentially as we increase the data.
- This problem overfitting of model due to large feature is known as "Curse Of Dimensinality".
- To get the rid of 'Curse of dimensinality' we have to perform the priciple component analysis or feature reduction technique before applying any machine learning algorithm.

Optimal Number of K value.

- In the KNN the k value is the controlling variable for the prediction model.
- While performing the model each k-value has it's own accuaracy but we want suh k value which can give us best prediction.
- As we said very earlier,K can work for the all the datasets.Each datasets has it's own requirement.
- If we choose the minimum value of k then it highly influential by the noise for the final result that is Prediction.
- Similarly if we take large number of the k value it will the model computatinaly very high it will give us high training time.
- Accroding the Research, If we choose the small number of k then it will helps to fit the model but will not give us better prediction that is overfitting of the model due low bias and high variance occurance.
- Similarly,If we choose the higher number of k then make the undefit due to high bias and low variance.

Why is Nearest Neighbor a Lazy Algorithm?

https://sebastianraschka.com/faq/docs/lazy-knn.html#why-is-nearest-neighbor-a-lazy-algorithm

- The nearest neighbor algorithms, for example, K-Nearest Neighbors (K-NN) are very "simple" algorithms, but that's not why they are called lazy ;).
- K-NN is a lazy learner because it doesn't learn a discriminative function (Categorical) from the training data but memorizes the training dataset instead.
- For example, the logistic regression algorithm learns the model weights during training time. In contrast, there is no training time in K-NN.
- On the other hand, the "prediction" step in K-NN is relatively expensive: Each time you want to make a prediction, you are searching for the nearest neighbor in the entire training set (note that there are tricks such as BallTrees and KDtrees to speed this up a bit.
- To summarize: An eager learner has a model fitting or training step. A lazy learner does not have a training phase.