K- Nearest Neighbors

simple, easy to use supervised Machine Learning Algorithm, used to solve both the classification and regression problem.

KNN is a non-parametric, lazy algo.

Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new data point.

---

when we say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. ie. the model structure is determined from the data.

It is indeed useful as real world data doesn't obey the typical theoretical assumptions.

KNN is also a lazy algo as opposed to eager. It means it doesn't use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This also means that the training phase is fast.

Lack of generalization means that KNN keeps all the training data. To be more exact, all or most of the data (training) is needed during the testing phase.

KNN is based on feature similarity, how closely out of sample features resemble our training set.

The output is a discrete value ie. the predicted class.

An object is classified by a majority vote of its neighbours with the object being assigned to the class most common among its K nearest neighbours.

It can also be used for regression → output is the value for the object (predicts continuous values) → can be mean/ median of the values of its K-Nearest Neighbours.

Applications:

① should the bank give loan to an individual? would an individual default on his/her loan? is the person closer in characteristics to people who defaulted or not?

② handwriting recognition (OCR), image recognition and even video recognition.

Advantages:

① No assumptions about data.

② simple algo.

③ High accuracy: (not better compared to supervised)

④ versatile.

Disadvantages:

① Computationally expensive. (stores all training data)

② High memory requirement.

③ sensitive to irrelevant features and scale of the data.

Notes

= There are two types of learners in classification tasks:

Lazy   Eager

⇒ Lazy learners simply store the training data and wait until a testing data appears.

Compared to eager learners, lazy learners have less training time but takes more time in predicting.

ex: KNN, case-based learning.

⇒ Eager learners construct a classification model based on the given training data before receiving data for classification.

ex: Decision Tree, Naive Bayes, ANN.

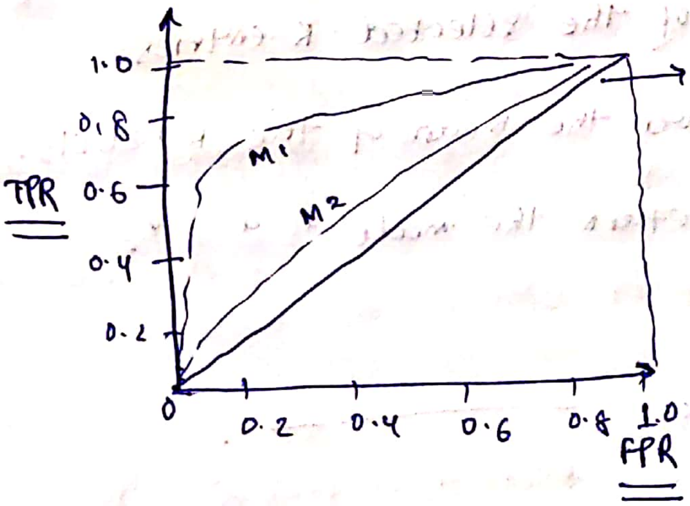
——— x ——— x ———

Roc curve (Receiver operating characteristics) :

① Roc curve is used for visual comparison of classification models which show the trade off between ~~TR~~ TPR and FPR.

② The area under the ROC curve is a measure of the accuracy of the model.

③ when a model is closer to the diagonal, it is less accurate and the model with perfect accuracy will have an area of 1.0.



M1 has better accuracy than M2.

The KNN algo assumes that similar things exist in close Proximity.

In other words, similar things are near to each other.

" Birds of a feather flock together."

Algorithm:

1. Load the data.
2. initialize 'K' to your chosen no. of neighbors.
3. for each example in the data:
    1. calc. distance between the query example and the current example from the data.
    2. add the distance and index of the example to an ordered collection.
3. Sort in ascending order.
4. Pick the first K entries from the sorted collection
5. get the labels of the selected K entries
6. if regression, return the mean of the K labels.
7. if classification, return the mode " " ".

—— X —— X —— .

To find optimal value of K:

1. use bruteforce
2. use elbow method
3. K Fold CV.