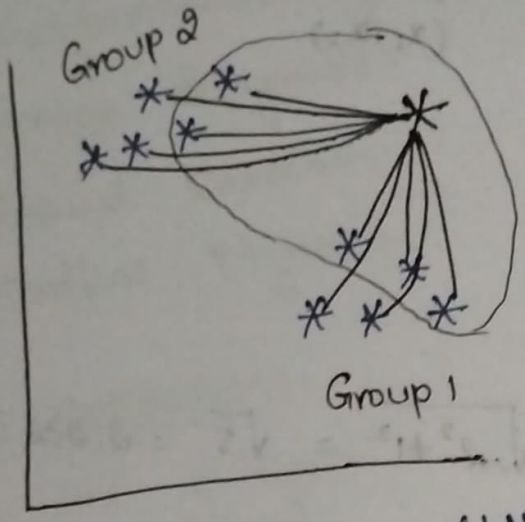
02/18/25

(K-N-N) kNN Algorithm: (Classification)

kNN → k - Nearest - Neighbours

Also called as Lazy Algorithm.



We have to define no. of neighbours, usually we will take odd number

Ex: 5

* k-nearest neighbours (kNN) is a supervised algorithm used for classification and regression based on distance b/w data points.

* Choose the k-value
* Find k nearest data points to the test point
* For classification we will go with majority vote, in the regression it will go with average distance value.

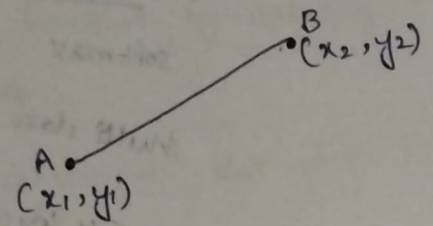
classification - majority vote
regression - avg. distance value.

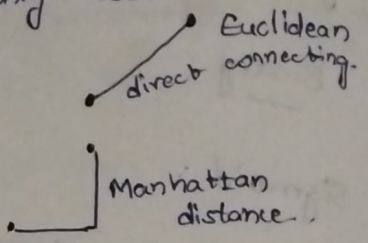
* Why it is called as lazy learner?

- No training phase.
- Stores entire dataset.
- Computes everything during prediction

Distance Metrics:

1. Euclidean Distance / L2 distance : commonly used.





$$\text{distance } d = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

Incase of more coordinates,

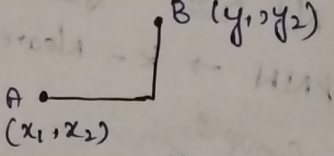
$$d = \sqrt{\Sigma(x_i-y_i)^2} = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$$

A(x_1, x_2) B(y_1, y_2)

2. Manhatton Distance / L1 distance :

$$d = |x_1-y_1| + |x_2-y_2|$$

$$d = \Sigma|x_i-y_i|$$



Example

A = (5, 3) B = (3, 2)

Euclidean $d = \sqrt{(5-3)^2+(3-2)^2} = \sqrt{2^2+1^2} = \sqrt{5} = 2.236$

Manhatton $d = |5-3| + |3-2| = 2+1 = 3.$

How to choose k-value?

* Small k value leads to overfitting problem.
* Large k value leads to Underfitting problem.
* Choose odd k value for binary classes.
* Use cross-validation to select best k.

* Note: Feature Scaling is required (standard / minmax) as KNN is distance based algorithm.

Pros & Cons of KNN:

Pros:
* Simple
* No training time
* Works well on small dataset.

Cons:
* Slow prediction
* High Memory Usage
* Sensitive to noisy data (Outliers)
* Poor in high dimension datas.

Applications:
1. Recommendation System
2. Image Recognition
3. Fraud Detection.
4. Medical Diagnosis.

Model validation:
There are 3 types:

1. Train Test Split
2. k-fold Cross Validation
3. Leave one Out cross validation.

k-fold cross validation:

* for example : have 1000 records
* first perform train-test split 80:20 ratio
  800 - train
  200 - test.
* for train data alone we will perform k-fold cross validation.
* In k-fold we have two standard k-values 5 and 10.
* Based on k value the train data is further divided into k groups.

$k = 5$

$800/5 = 160$

D_1 D_2 D_3 D_4 D_5 – every group has 160 records each.

Train	Test
D_1, D_2, D_3, D_4	D_5
D_1, D_2, D_3, D_5	D_4
D_1, D_2, D_4, D_5	D_3
D_1, D_3, D_4, D_5	D_2
D_2, D_3, D_4, D_5	D_1

* The model will undergo 5 level trainings, but it is time consuming and more efficient compared to train test split.

Leave-One Out Cross Validation:

*Never used generally.

* In the 800 train data every time 1 will be left out.

* The model will undergo training for 800 times.

* More time consuming, never used, not efficient.