

# K Nearest Neighbor and MLE

# Recap from Last session

- Decision Tree working example with gini index
- Ensemble learning intuition
- Bagging, Boosting and Stacking
- Random Forest
- Precision and recall with confused matrix

# Agenda for today

- K nearest neighbor
- Maximum likelihood estimate

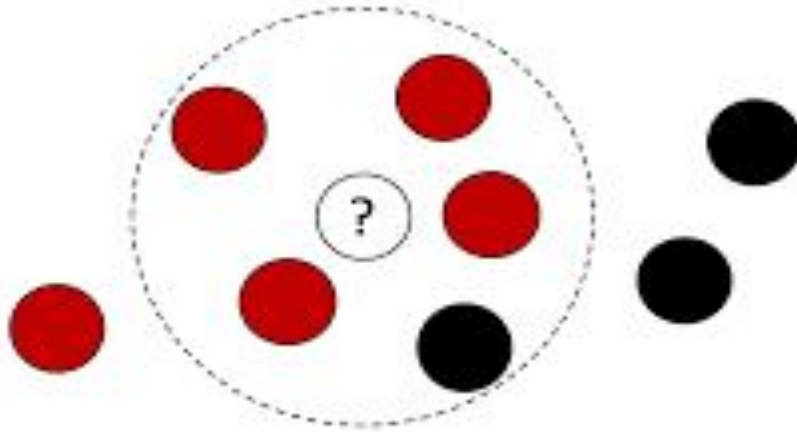
# Introduction to KNN

- Non parametric algorithm
- Both for classification and regression
- One of the most easier and simpler models to understand
- Lazy learner

# KNN

- Used to classify objects based on closest training examples in the feature space
- Feature space: raw data transformed into sample vectors of fixed length using feature extraction (**Important**)

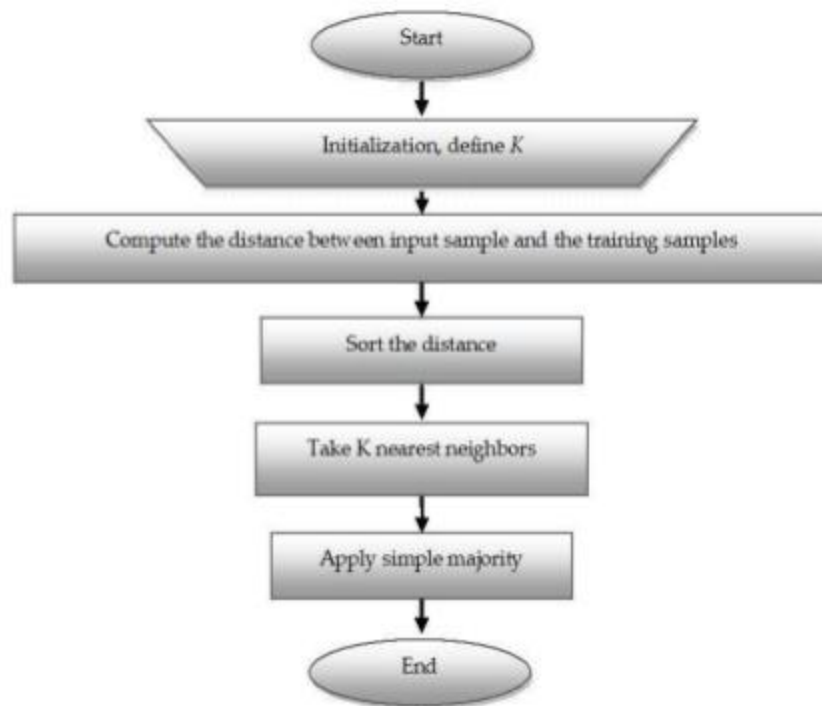
Prediction on test time



# How does it work?

- It basically requires three things:
  - Feature space
  - Distance metric
    - Euclidean Distance
    - Manhattan Distance
    - Hamming Distance
  - Number of neighbors to consider
- **No Learning in the algorithm**

# KNN Classifier Algorithm





# How to classify at test time?

- Compute distance to other training records
- Identify  $k$  nearest neighbors (hence, the name)
- For classification:
  - Assign the mode label of the  $k$  neighbors
- For regression:
  - Assign the average of the  $k$  neighbors

## Distance Measures

$$\textit{Euclidean distance} : d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

$$\textit{Squared Euclidean distance} : d(x, y) = \sum (x_i - y_i)^2$$

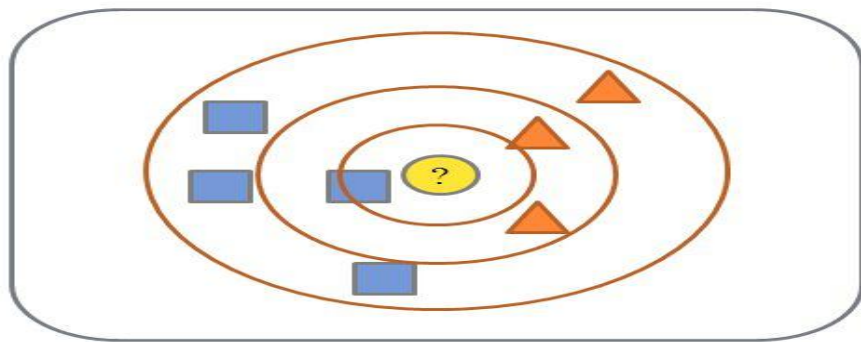
$$\textit{Manhattan distance} : d(x, y) = \sum |x_i - y_i|$$

### **Which distance measure to use?**

We use Euclidean Distance as it treats each feature as equally important.

# Impact of K

## $k$ NEAREST NEIGHBOR



- $k = 1$ :
  - Belongs to square class
- $k = 3$ :
  - Belongs to triangle class
- $k = 7$ :
  - Belongs to square class

- Choosing the value of  $k$ :
  - If  $k$  is too small, sensitive to noise points
  - If  $k$  is too large, neighborhood may include points from other classes
  - Choose an odd value for  $k$ , to eliminate ties

# Maximum Likelihood Estimate