Introduction

- k Nearest Neighbors(kNN) is a non-parametric method used for classification
- It is a lazy learning algorithm where all computation is deferred until classification
- It is also an instance based learning algorithm where the function is approximated locally

Why kNN and when does one use it?

- Why kNN ?
  - Simplest of all classification algorithms and easy to implement
  - There is no explicit training phase and the algorithm does not perform any generalization of the training data
- When does one use this algorithm?
  - When there are nonlinear decision boundaries between classes
  - When the amount of data is large

k Nearest Neighbors

- Input features
  - Input features can be both quantitative and qualitative
- Outputs
  - Outputs are categorical values, which typically are the classes of the data
- kNN explains a categorical value using the majority votes of nearest neighbors

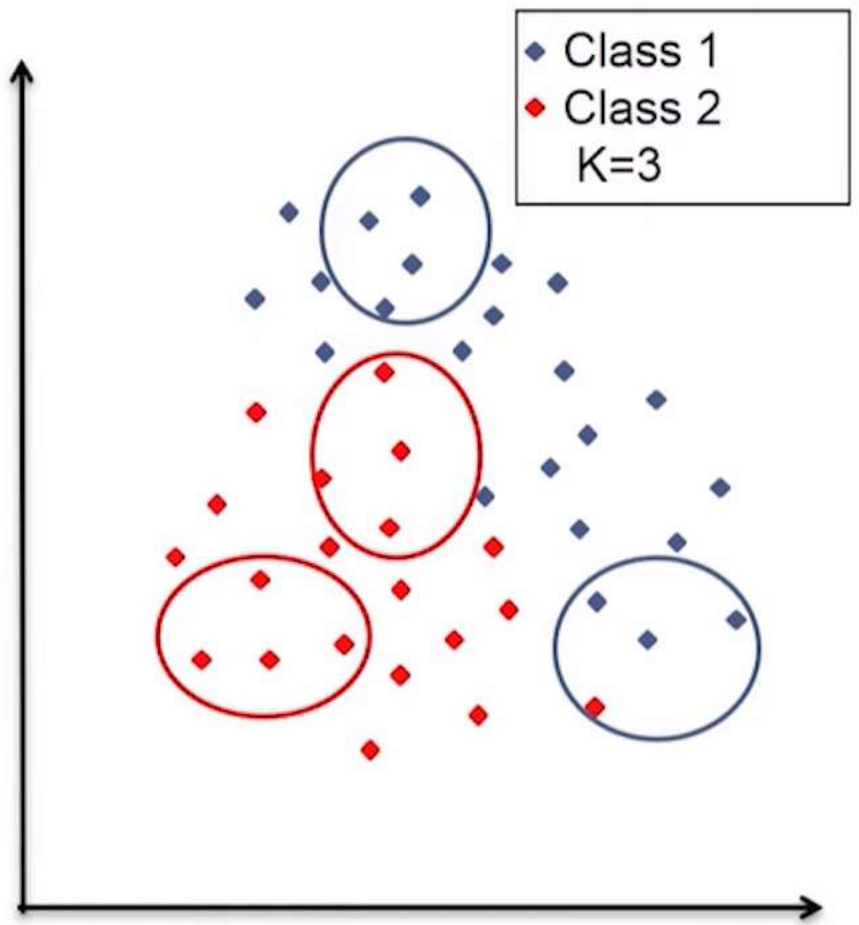
Assumptions

- Being nonparametric, the algorithm does not make any assumptions about the underlying data distribution
- Select the parameter k based on the data
- Requires a distance metric to define proximity between any two data points
  - Example: Euclidean distance, Mahalanobis distance or Hamming distance

Algorithm

- The kNN classification is performed using the following four steps
  - Compute the distance metric between the test data point and all the labeled data points
  - Order the labeled data points in the increasing order of this distance metric
  - Select the top k labeled data points and look at the class labels
  - Find the class label that the majority of these k labeled data points have and assign it to the test data point

Illustration of kNN



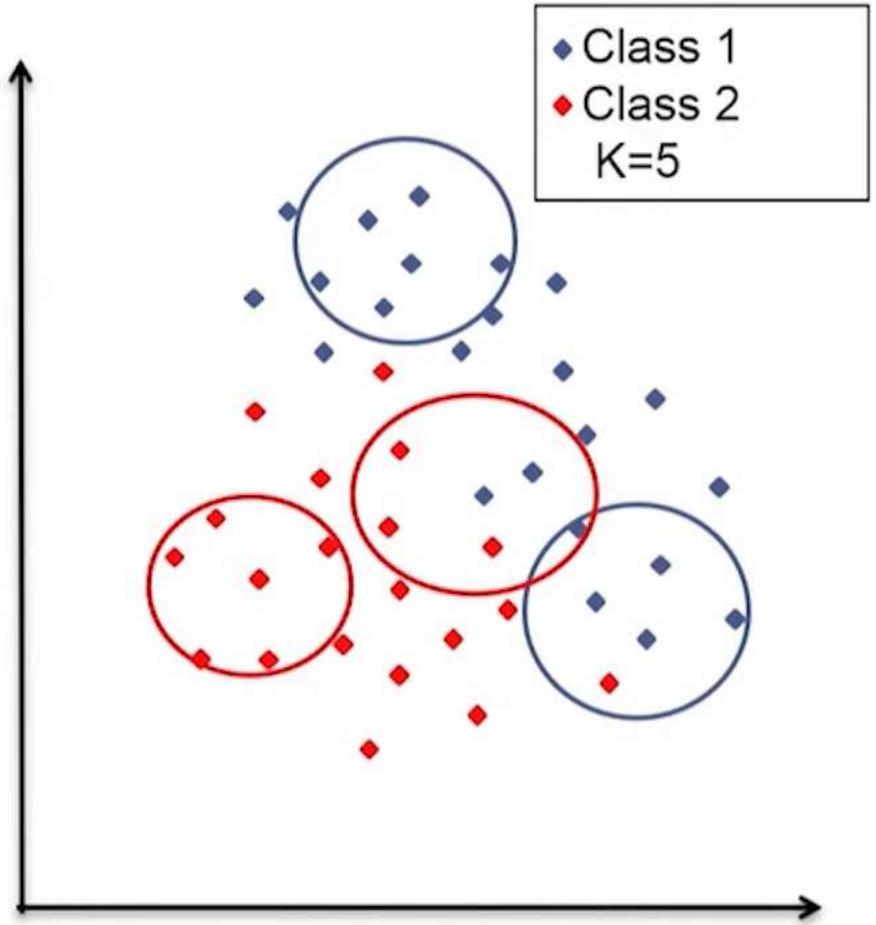
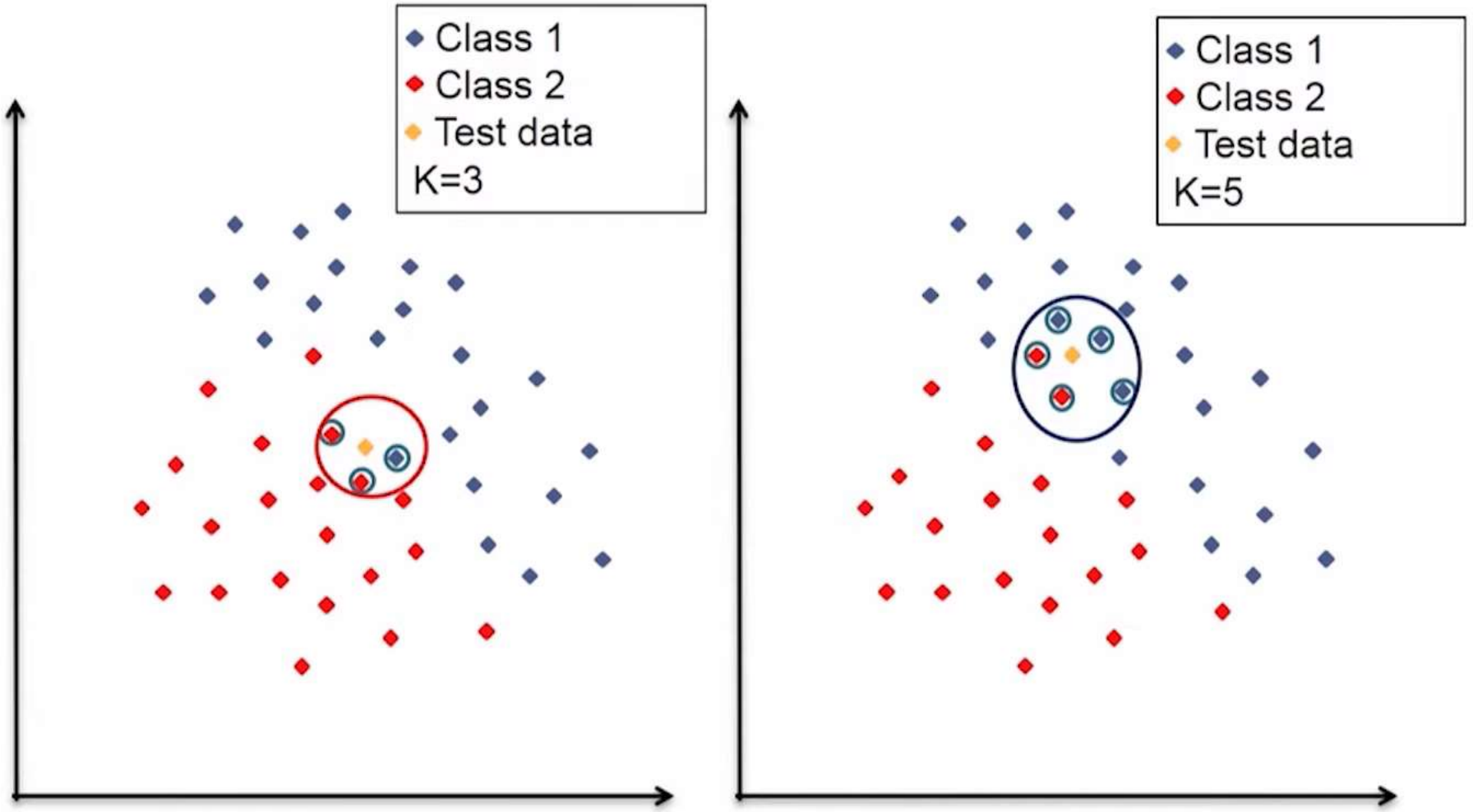


Illustration of kNN (Testing)



Things to consider

- Following are some things one should consider before applying kNN algorithm
  - Parameter selection
  - Presence of noise
  - Feature selection and scaling
  - Curse of dimensionality

Parameter selection

- The best choice of k depends on the data
- Larger values of k reduce the effect of noise on classification but makes the decision boundaries between classes less distinct
- Smaller values of k tend to be affected by the noise with clear separation between classes

Feature selection and scaling

- It is important to remove irrelevant features
- When the number of features is too large, and suspected to be highly redundant, feature extraction is required
- If the features are carefully chosen then it is expected that the classification will be better