



Artificial Learning Models

Lecture 4 : K-Nearest Neighbors

By : Dr. Lamri SAYAD

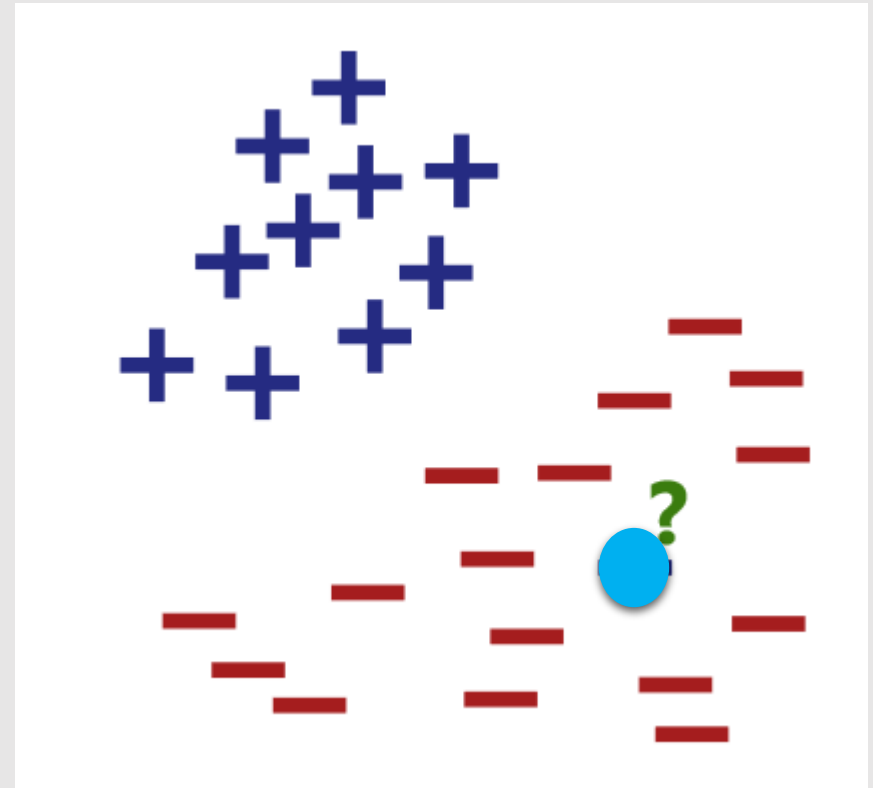
2024

Introduction

- Consider 2 classes
 - Positive vs negative
 - The blue new point
- What is the classe of the blue point ?

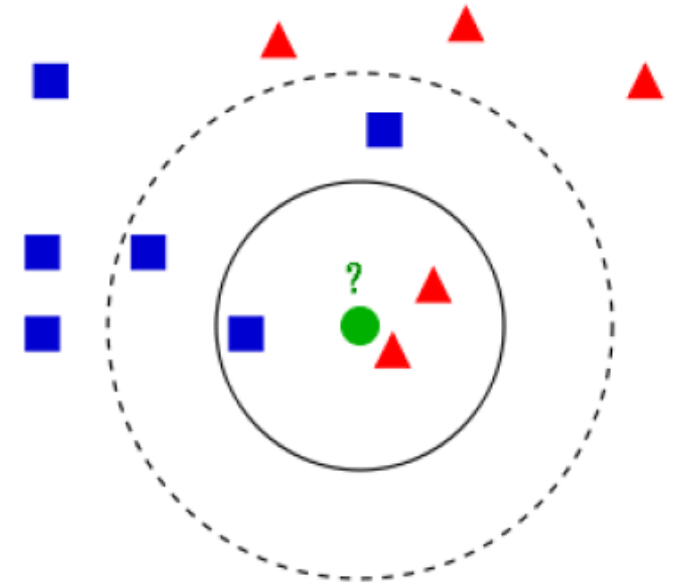


- voting system or a popularity contest



K-Nearest Neighbor (KNN)

- Supervised machine learning algorithm
 - Used for classification
 - But it can also be used for regression
 - Very simple but effective
- Used to classify new data points based on “distance” to known data
 - Need a distance measure
- Find the K nearest neighbors, based on the distance metric
 - Let them all vote on the classification



K-Nearest Neighbors (KNN)

How does it work ?

- **In training phase :**
 - We define number of neighbors (k) and distance metric to be used.
 - There is no training
 - KNN model stores the training data with its labels/categories.
- **In Prediction (or test) phase :** If we provide a query point to the model :
 - It will find the distance of all the training data points with the query point,
 - Sort the distances in ascending order.
 - Then choose k number of nearest neighbor.
 - Whichever group will have more data points out of k neighbors, query point will be assigned to that group.

K-Nearest Neighbor (KNN) Distance

- In D-dimensional space,

- Minkowski distance

$$d_{minkowski} = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

- the Euclidean distance

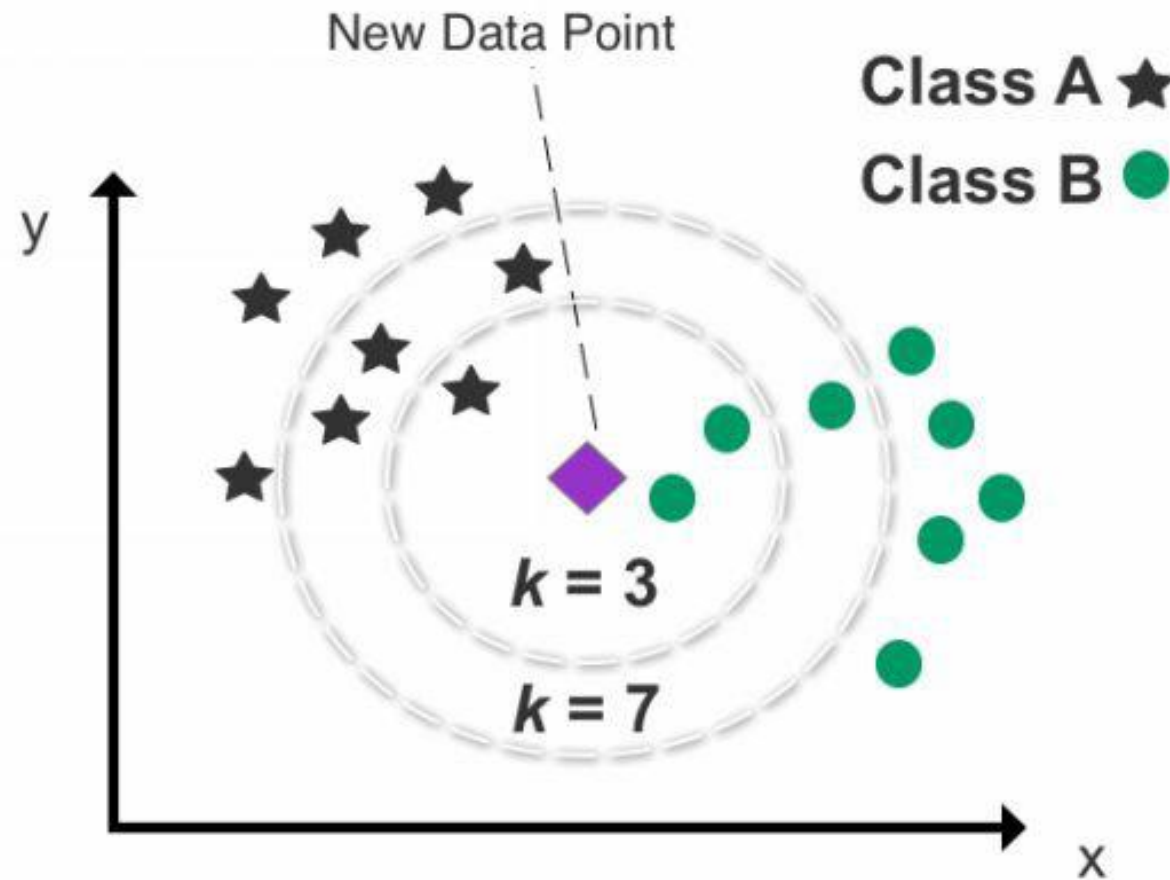
$$d_{euclidean} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Manhattan distance

$$d_{manhattan} = \sum_{i=1}^n |x_i - y_i|$$

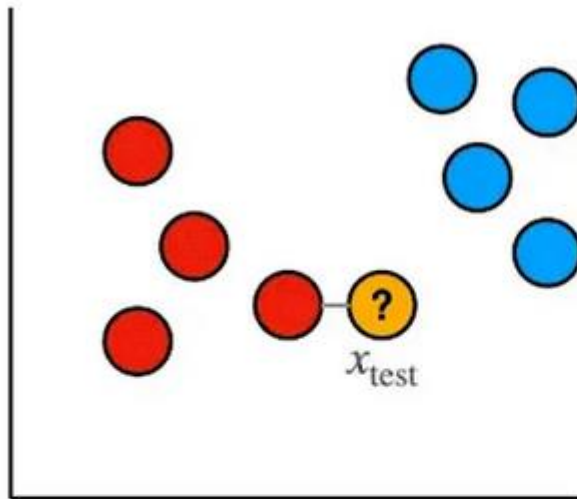
K-Nearest Neighbor (KNN)

Impact of K



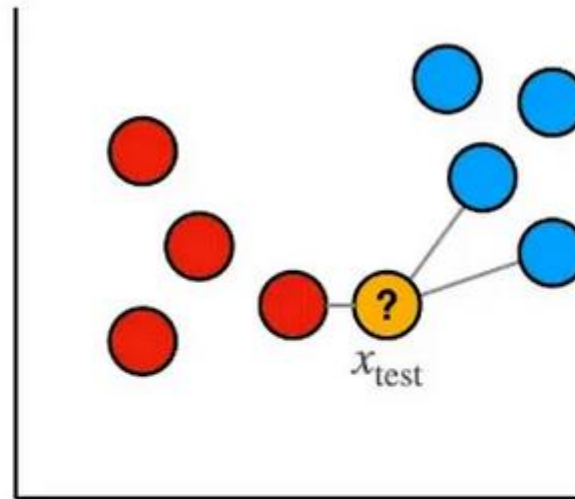
K-Nearest Neighbor (KNN)

Impact of K



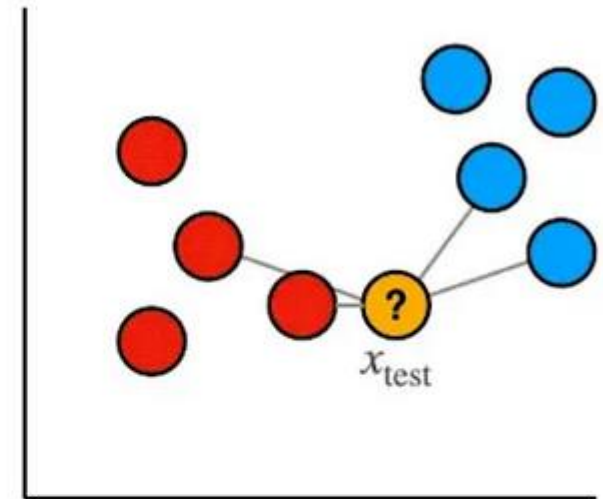
$k = 1$

Nearest point is **red**, so x_{test} classified as **red**



$k = 3$

Nearest points are {**red**, **blue**, **blue**} so x_{test} classified as **blue**



$k = 4$

Nearest points are {**red**, **red**, **blue**, **blue**} so classification of x_{test} is not properly defined

K-Nearest Neighbor (KNN)

How to choose K ?

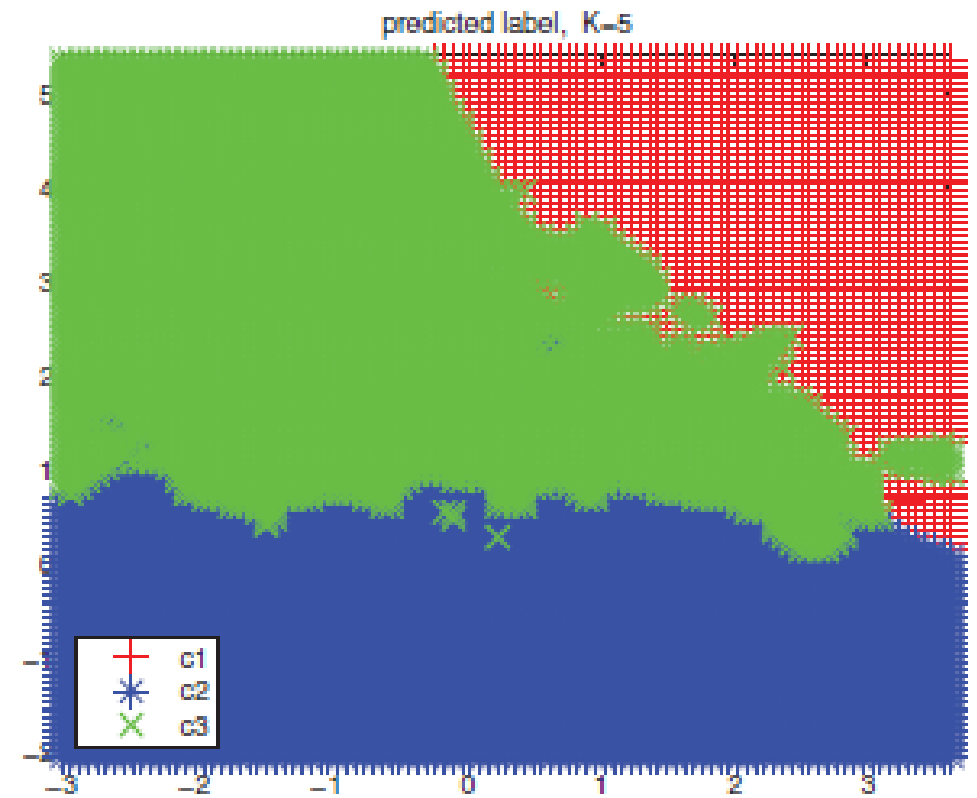
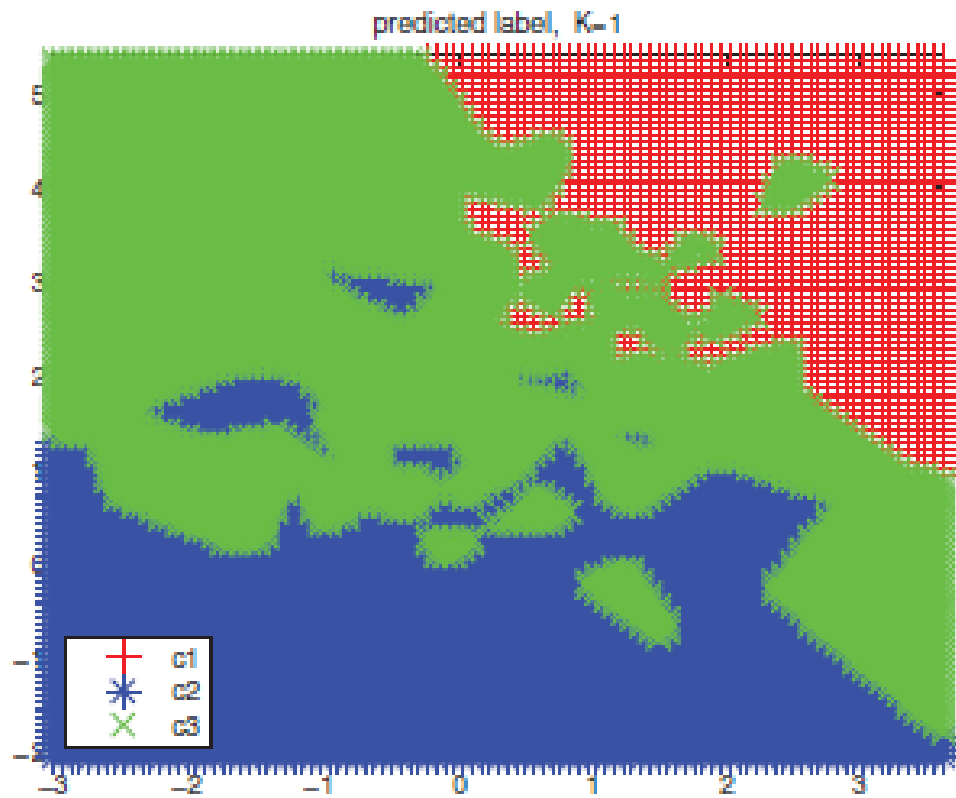
- choose odd value of k to avoid ties in classification.
- Approach :
 - Create different KNN model for $k = 1, k = 3, \dots, k = 21 \dots$
 - Train these different KNN model on training data.
 - Run trained model on test data.
 - Find accuracy score.
 - Select one with high accuracy.

K-Nearest Neighbor (KNN)

value of K vs overfitting and underfitting ?

- Small K value (like 1 or 3):
 - ☺ Capture fine details in the data
 - ☹ May also be sensitive to noise and outliers.
- Large value of K :
 - ☺ Provides more generalized predictions
 - ☺ Reduces the risk of overfitting
 - ☹ Could underfit if taken to an extreme.
- If the data is densely packed,
 - a smaller 'k' might suffice
- sparse datasets
 - might benefit from a larger 'k'

K-Nearest Neighbor (KNN) Decision boundary



K-Nearest Neighbor (KNN)

Variants : Weighted KNN

- In standard KNN
 - Each of the K neighbors contributes equally to the final decision (vote)
 - Distance to the query point change → it is unfair to give them the same importance



Weighted KNN

- neighbors are assigned weights based on their distance to the query point.
- closer neighbors are given more influence in determining the output than those further away.

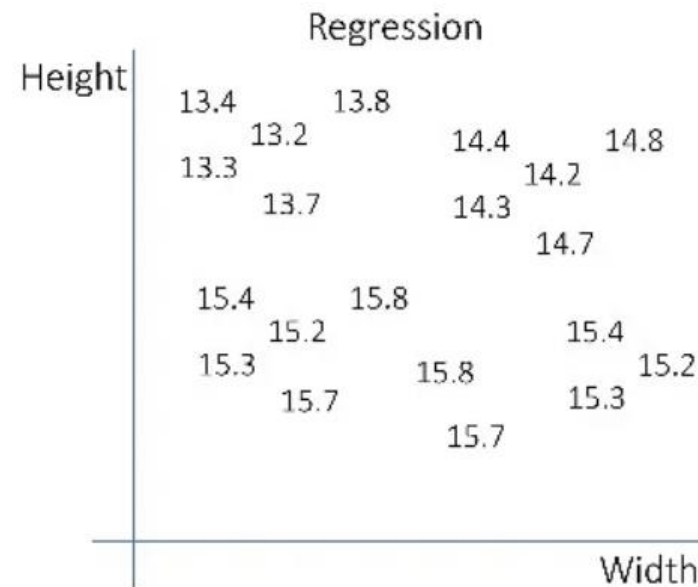
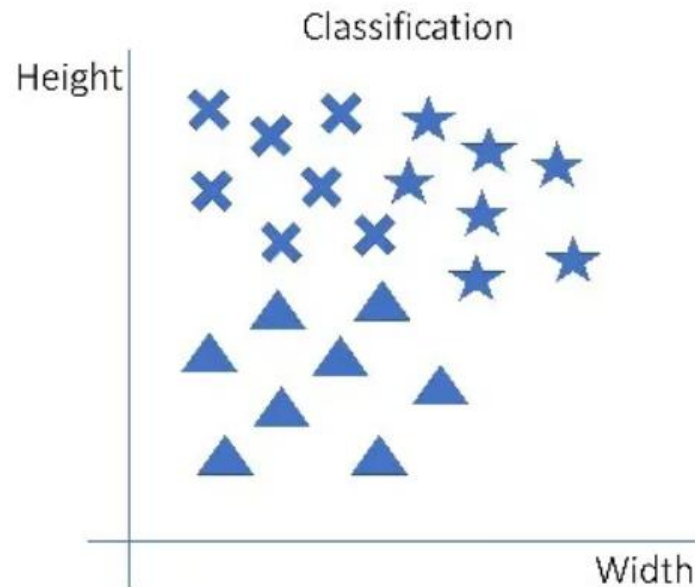
K-Nearest Neighbor (KNN) Applications

- Prediction
- Imputing missing data
 - If you have a small amount of data, predict the missing values using k-nearest neighbors (KNN)

K-Nearest Neighbor (KNN)

KNN for regression

- For classification problems,
 - the algorithm assigns a class label based on a majority vote
- For regression problems,
 - Continuous values are applied
 - The average is used to identify the k nearest neighbors.



K-Nearest Neighbor (KNN)

Advantages and disadvantages

- Advantages:

- ☺ Simple
- ☺ Training process is very fast
- ☺ Easy implementation

- Disadvantages:

- ☹ Huge memory consumption (because it needs to store all the data).
- ☹ Time complexity at testing
- ☹ Does not work well with large datasets
- ☹ Does not work well with high dimensionality
- ☹ Sensitive to noisy data

