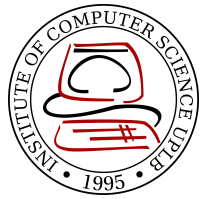


CMSC 170: Supervised Learning

K-Nearest Neighbors

Katherine Loren M. Tan
Institute of Computer Science
University of the Philippines Los Baños

LEARNING OUTCOMES

At the end of the session, the students should be able to:

- understand the K-Nearest Neighbor classification algorithm;
- implement the KNN algorithm; and
- Apply the KNN algorithm in classifying data.

K-Nearest Neighbor Classification Algorithm

It is a non-parametric machine learning algorithm that classifies new data based by finding its k-nearest neighbors in the data set.

K-Nearest Neighbor Classification Algorithm

The algorithm chooses the classification that represented the other data points the most.

HOW DOES THE ALGORITHM WORKS?

x_{00}	x_{01}	x_{02}	...	x_{0n}	->	Y_0
x_{10}	x_{11}	x_{12}	...	x_{1n}	->	Y_1
x_{20}	x_{21}	x_{22}	...	x_{2n}	->	Y_2
			⋮			
x_{m0}	x_{m1}	x_{m2}	...	x_{mn}	->	Y_m

HOW DOES THE ALGORITHM WORKS?

Data Points

x_{00}	x_{01}	x_{02}	...	x_{0n}	->	Y_0
x_{10}	x_{11}	x_{12}	...	x_{1n}	->	Y_1
x_{20}	x_{21}	x_{22}	...	x_{2n}	->	Y_2
			⋮			
x_{m0}	x_{m1}	x_{m2}	...	x_{mn}	->	Y_m

HOW DOES THE ALGORITHM WORKS?

						Label
x_{00}	x_{01}	x_{02}	...	x_{0n}	->	Y_0
x_{10}	x_{11}	x_{12}	...	x_{1n}	->	Y_1
x_{20}	x_{21}	x_{22}	...	x_{2n}	->	Y_2
			⋮			
x_{m0}	x_{m1}	x_{m2}	...	x_{mn}	->	Y_m

K-Nearest Neighbor Classification Algorithm

uses distance algorithms.

K-Nearest Neighbor Classification Algorithm

Common distance algorithms are euclidean, manhattan, and minkowski distance.

K-Nearest Neighbor Classification Algorithm

The classification of a new feature vector x=(x0,x1,x2,…,xn) is determined by computing its distance from all the other feature vectors in the training data set

K-Nearest Neighbor Classification Algorithm

choosing the *k* nearest neighbors, where *k* is a given value.

STEPS IN KNN

1. Provide a k and a feature vector x.

STEPS IN KNN

1. Provide a k and a feature vector x.
2. For each feature vector v in the training set, compute the Euclidean distance.

$$d = \sqrt{\sum_{i}^{n} (x_i - v_i)^2}$$

STEPS IN KNN

1. Provide a k and a feature vector x.
2. For each feature vector v in the training set, compute the Euclidean distance.

$$d = \sqrt{\sum_{i}^{n} (x_i - v_i)^2}$$

3. If it is one of the k nearest neighbors, it is remembered.
4.

STEPS IN KNN

1. Provide a k and a feature vector x.
2. For each feature vector v in the training set, compute the Euclidean distance.

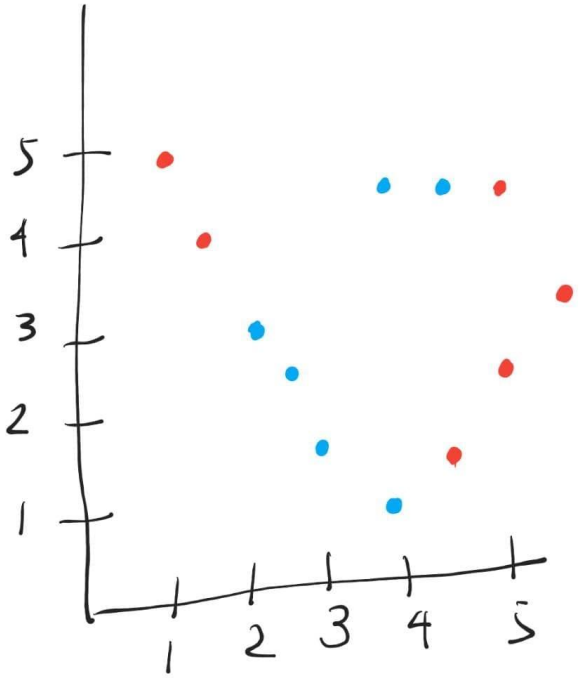
$$d = \sqrt{\sum_{i}^{n} (x_i - v_i)^2}$$

3. If it is one of the k nearest neighbors, it is remembered.
4. The class with the maximum count will be the classification of x.

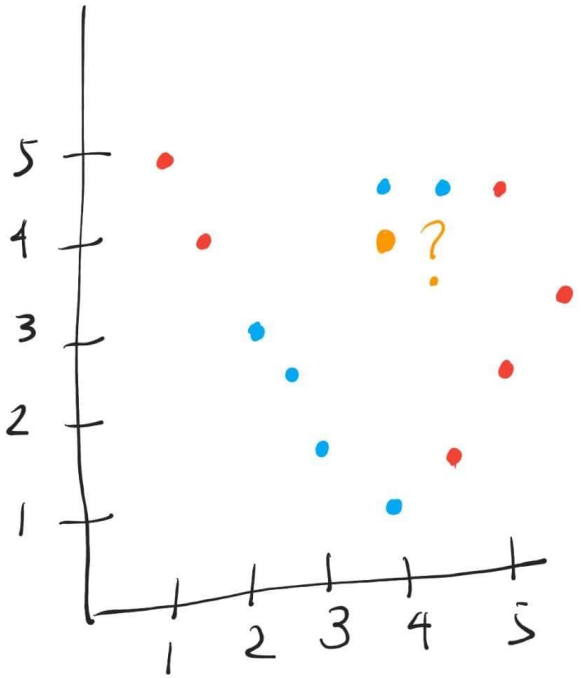
TRAINING DATASET

x	y	class
1	5	0
1.5	4	0
2	3	1
2.5	2.5	1
3	1.5	1
4	1	1
4	4.5	1

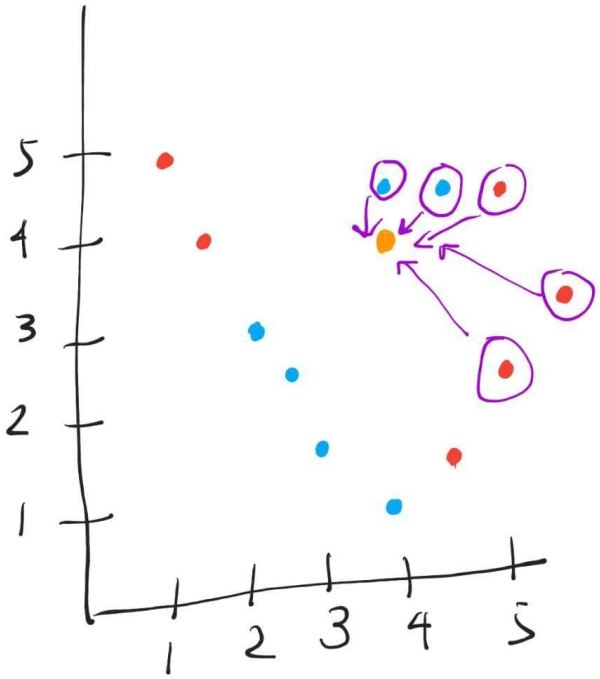
x	y	class
4.5	1.5	0
4.5	4.5	1
5	2.5	0
5	4.5	0
5.5	3.5	0

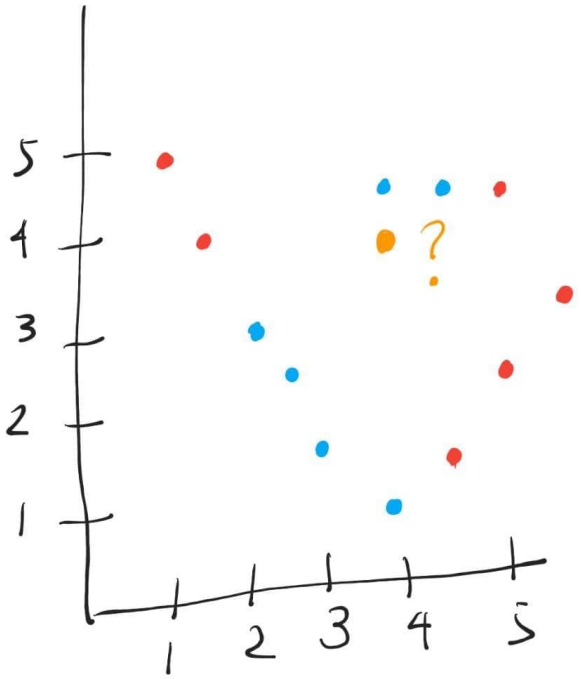


VISUAL REPRESENTATION OF TRAINING SET

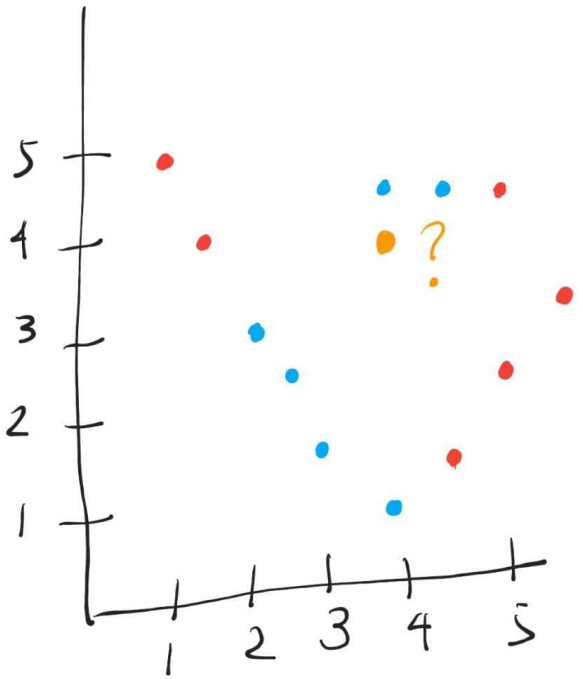


Input
4 4 What is its class?


5
4
3
2
1
1
2
3
4
5

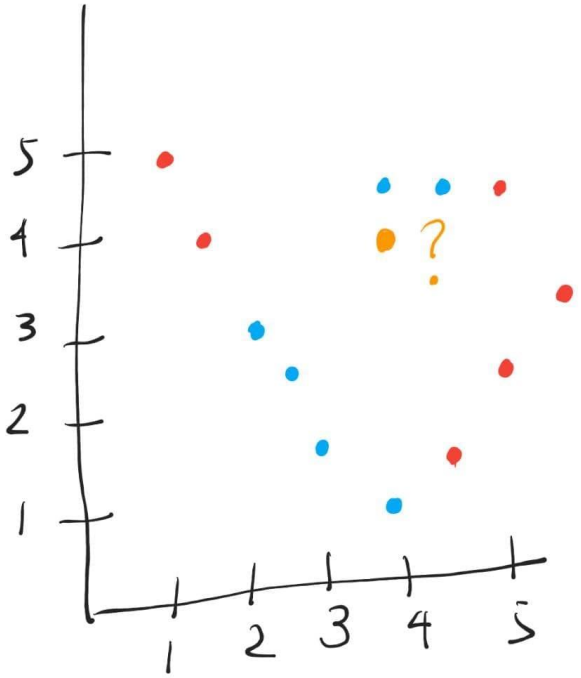


x	y	class	Distance from 4 4
1	5	0	3.1622
1.5	4	0	2.5000
2	3	1	2.2360
2.5	2.5	1	2.1213
3	1.5	1	2.6925
4	1	1	3
4	4.5	1	0.5



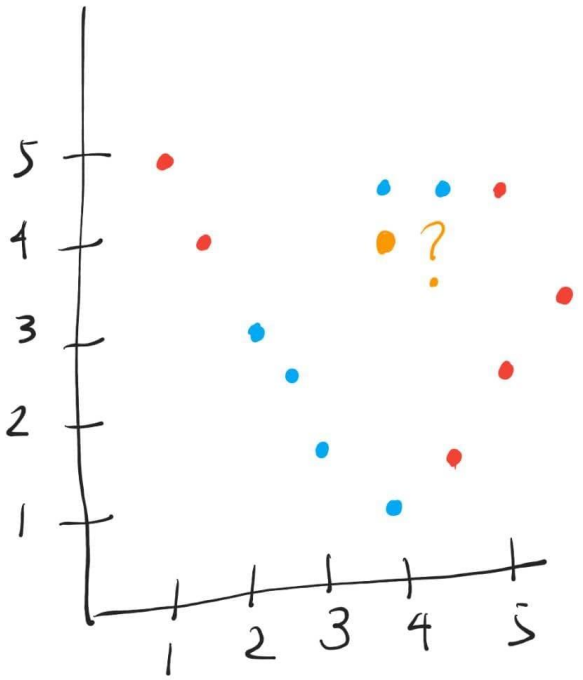
x	y	class	Distance from 4 4
4.5	1.5	0	2.5495
4.5	4.5	1	0.7071
5	2.5	0	1.8027
5	4.5	0	1.1180
5.5	3.5	0	1.5811

k=5



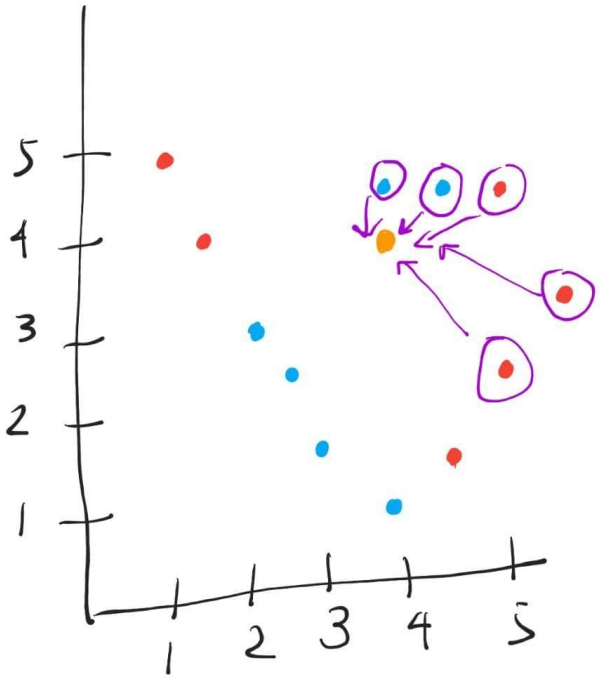
x	y	class	Distance from 4 4
1	5	0	3.1622
1.5	4	0	2.5000
2	3	1	2.2360
2.5	2.5	1	2.1213
3	1.5	1	2.6925
4	1	1	3
4	4.5	1	0.5

k=5



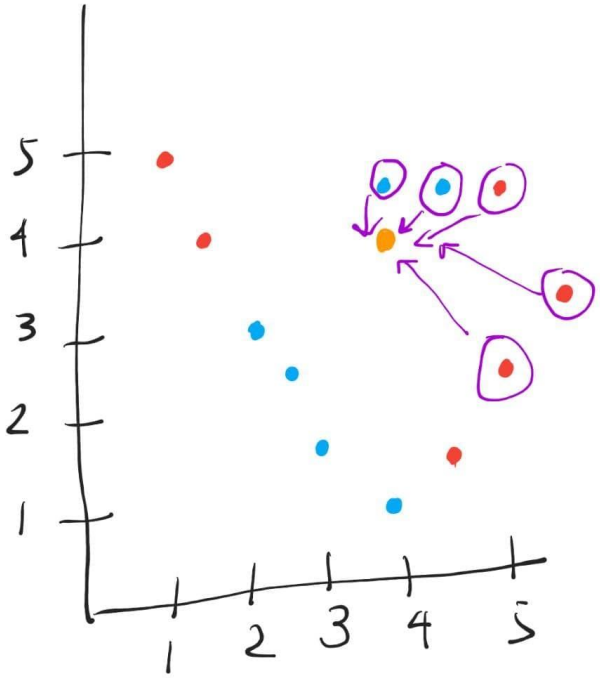
x	y	class	Distance from 4 4
4.5	1.5	0	2.5495
4.5	4.5	1	0.7071
5	2.5	0	1.8027
5	4.5	0	1.1180
5.5	3.5	0	1.5811

k=5



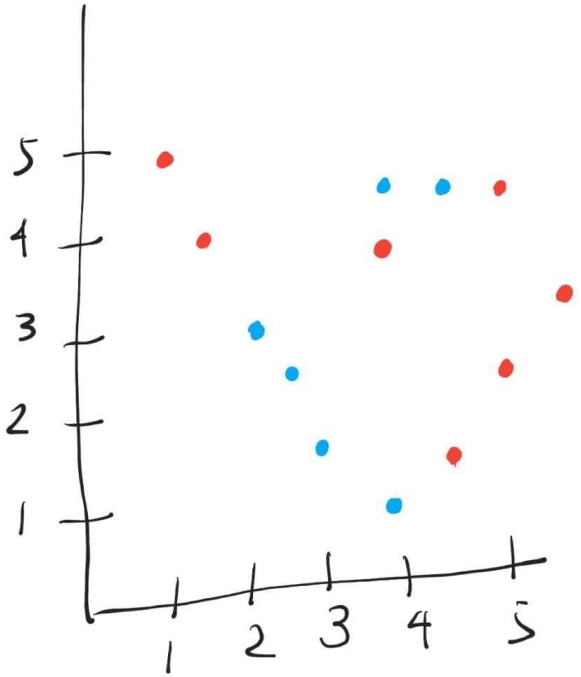
5 nearest neighbors of (4,4) are:
(4, 4.5), (4.5, 4.5), (5, 2.5), (5, 4.5) and (5.5, 3.5)

Get the classes of the nearest neighbors



The class of the 5 neighbors are:
(4, 4.5) 1,
(4.5, 4.5) 1,
(5, 2.5) 0,
(5, 4.5) 0 and
(5.5, 3.5) 0

Get the classes of the nearest neighbors



The new point (4,4) is labelled as class 0.

For questions and inquiries, you can email me at

kmtan4@up.edu.ph

EXERCISE on KNN

A dataset containing diabetes information will be used. The task is to classify the next points from **diabetes.csv**. The test file contains information regarding the no of pregnancies, glucose value, blood pressure, skin thickness, insulin value, bmi, diabetes pedigree function, age, and outcome of a person. The person can be classified as either diabetic or non-diabetic. The results must be placed on **output.txt**.

SOME REMINDERS:

- Naming convention for exercise: surname_knn.
- Python or Java can only be used for the exercise.
- Do not used built-in libraries for KNN.
- Do not forget to put a journal in your ReadMe file in Github.
- Lastly, **Honor and Excellence.**