ĐẠI NAM
UNIVERSITY

Bài 3

HỌC CÓ GIÁM SÁT

Thuật toán K-Nearest Neighbors

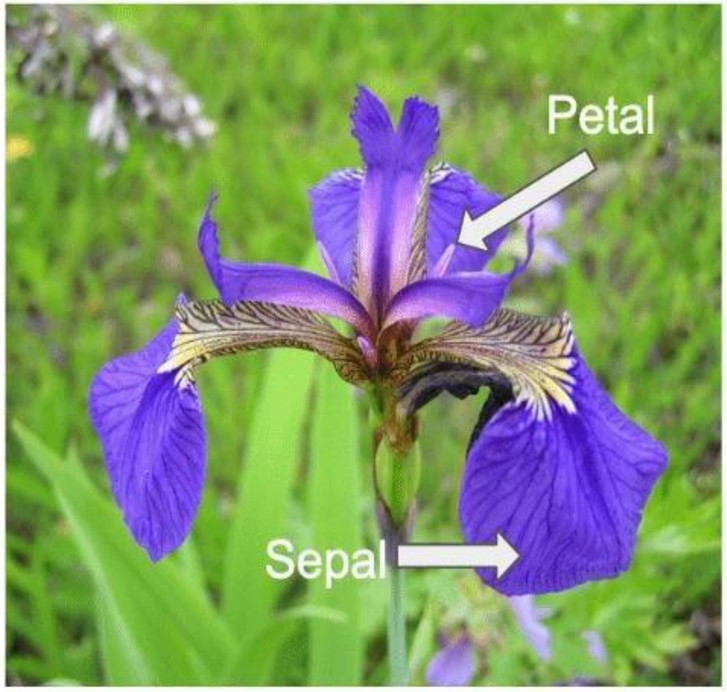
Giảng viên: TS. Nguyễn Ngọc Giang

SĐT: 0862411011

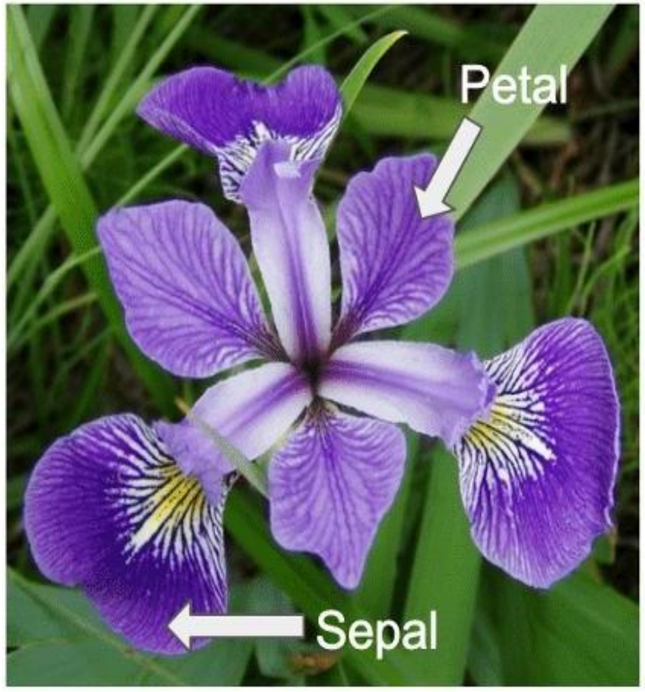
Email: giangnn.cntt@dainam.edu.vn

- **Quy trình xây dựng dữ liệu cho học máy**
- **Phân tích dữ liệu Iris**

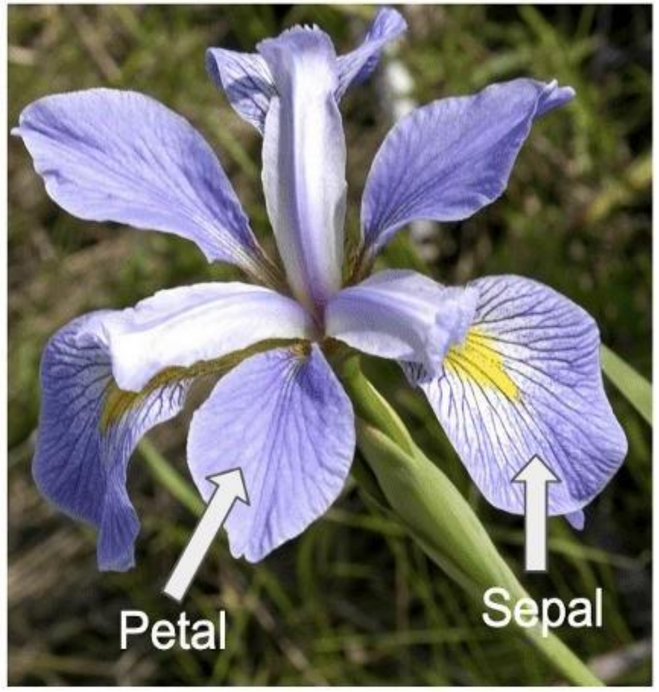
Iris setosa



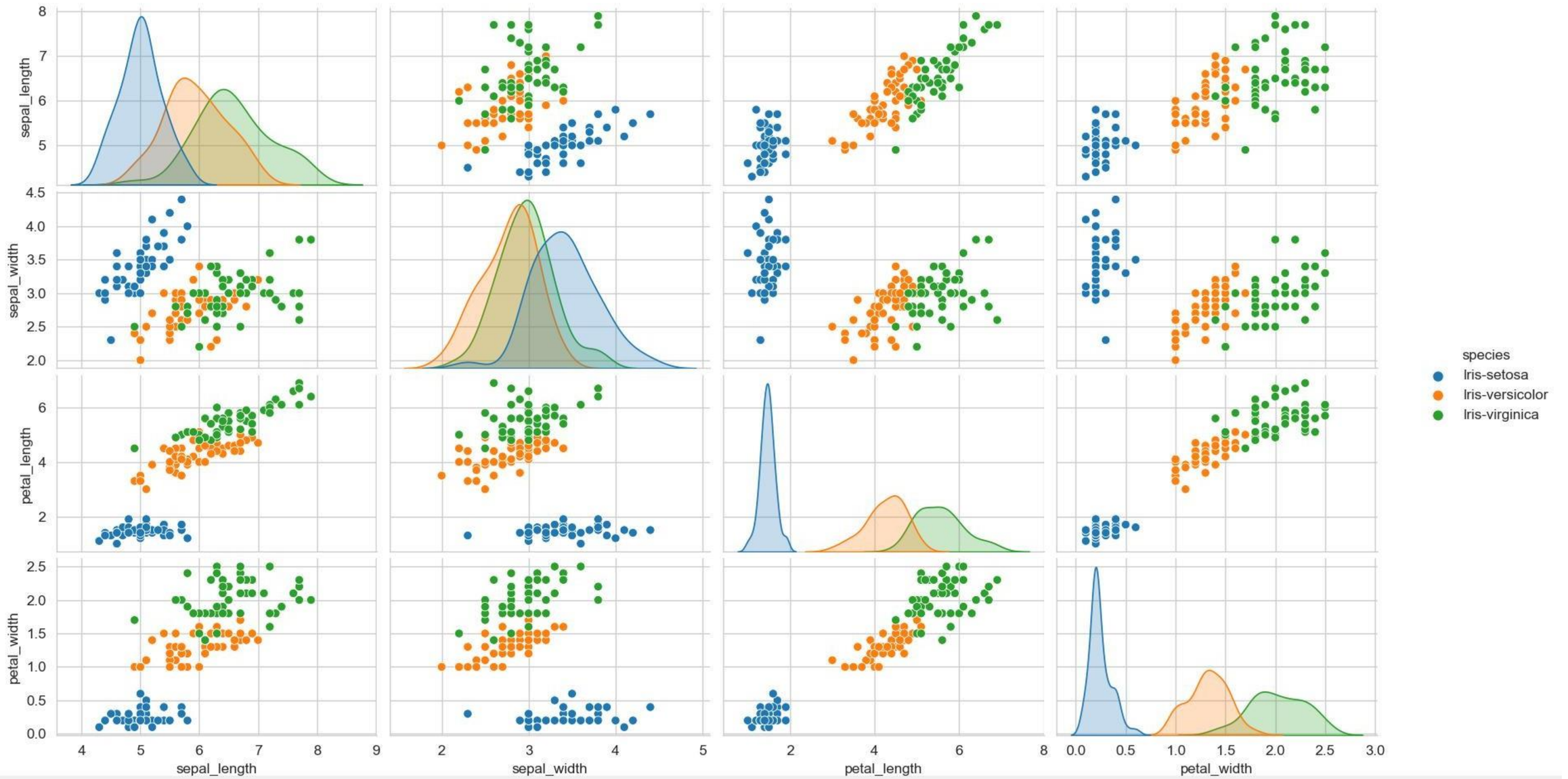
Iris versicolor



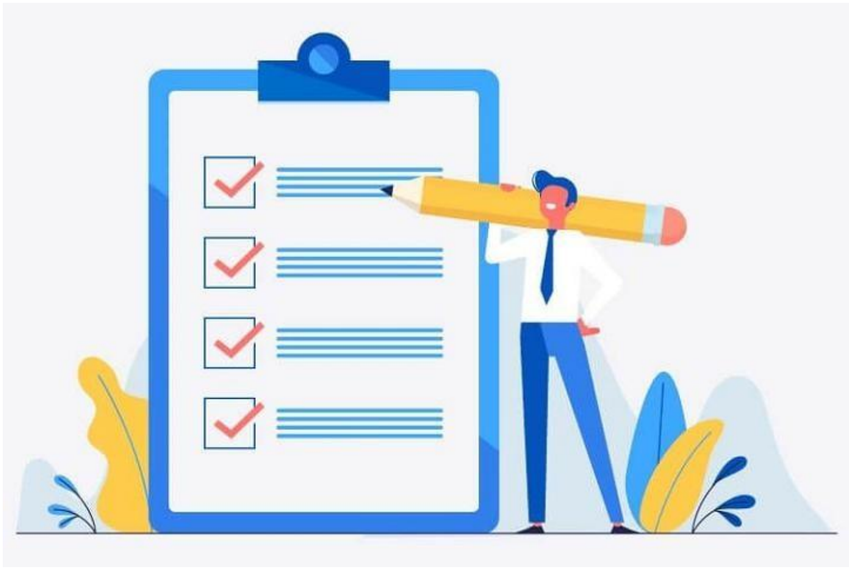
Iris virginica



Ôn lại bài cũ

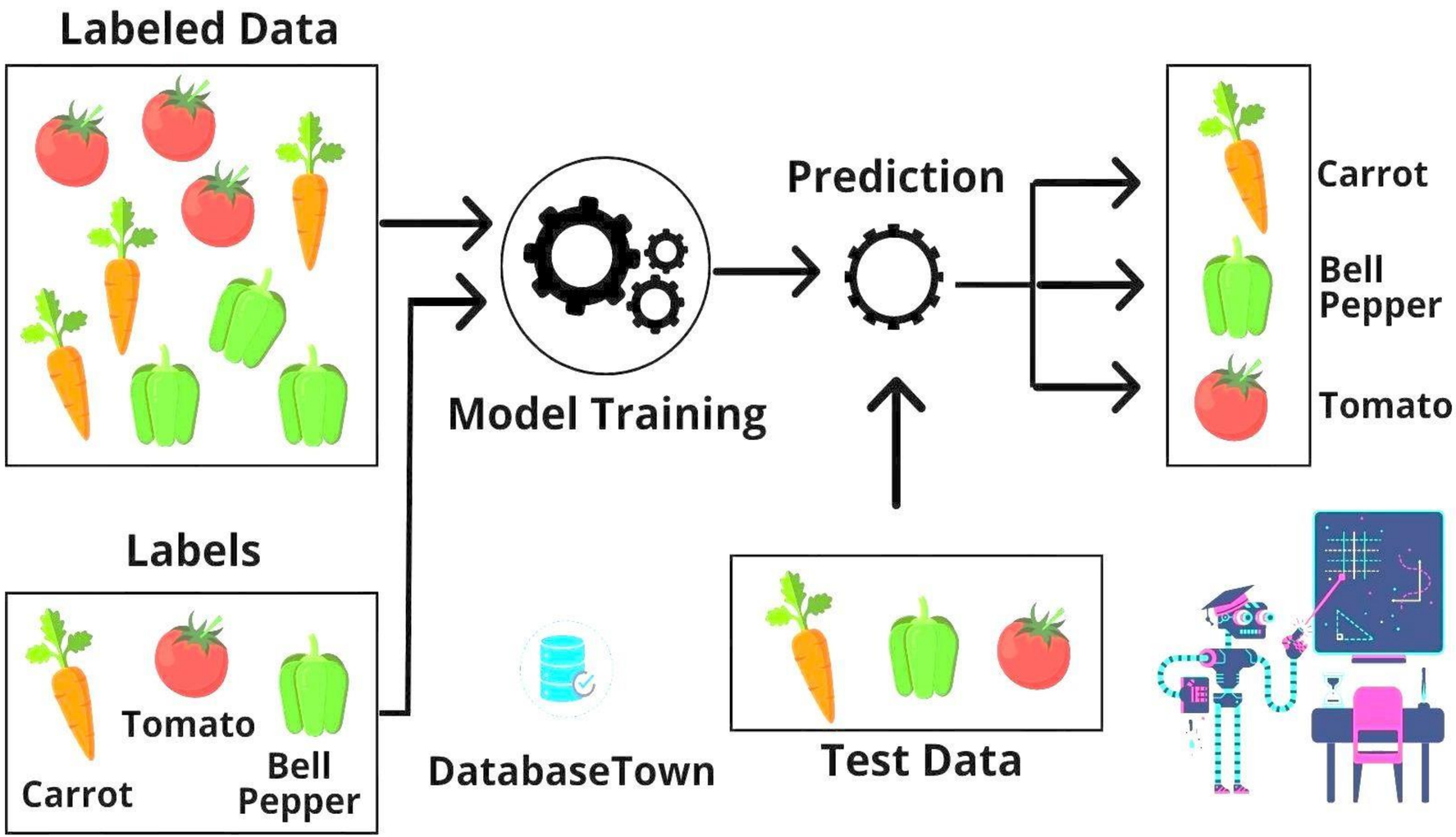

sepal_length
sepal_width
petal_length
petal_width
species
Iris-setosa
Iris-versicolor
Iris-virginica

- **Giới thiệu**
- **Thuật toán K-NN**
- **Luyện tập**

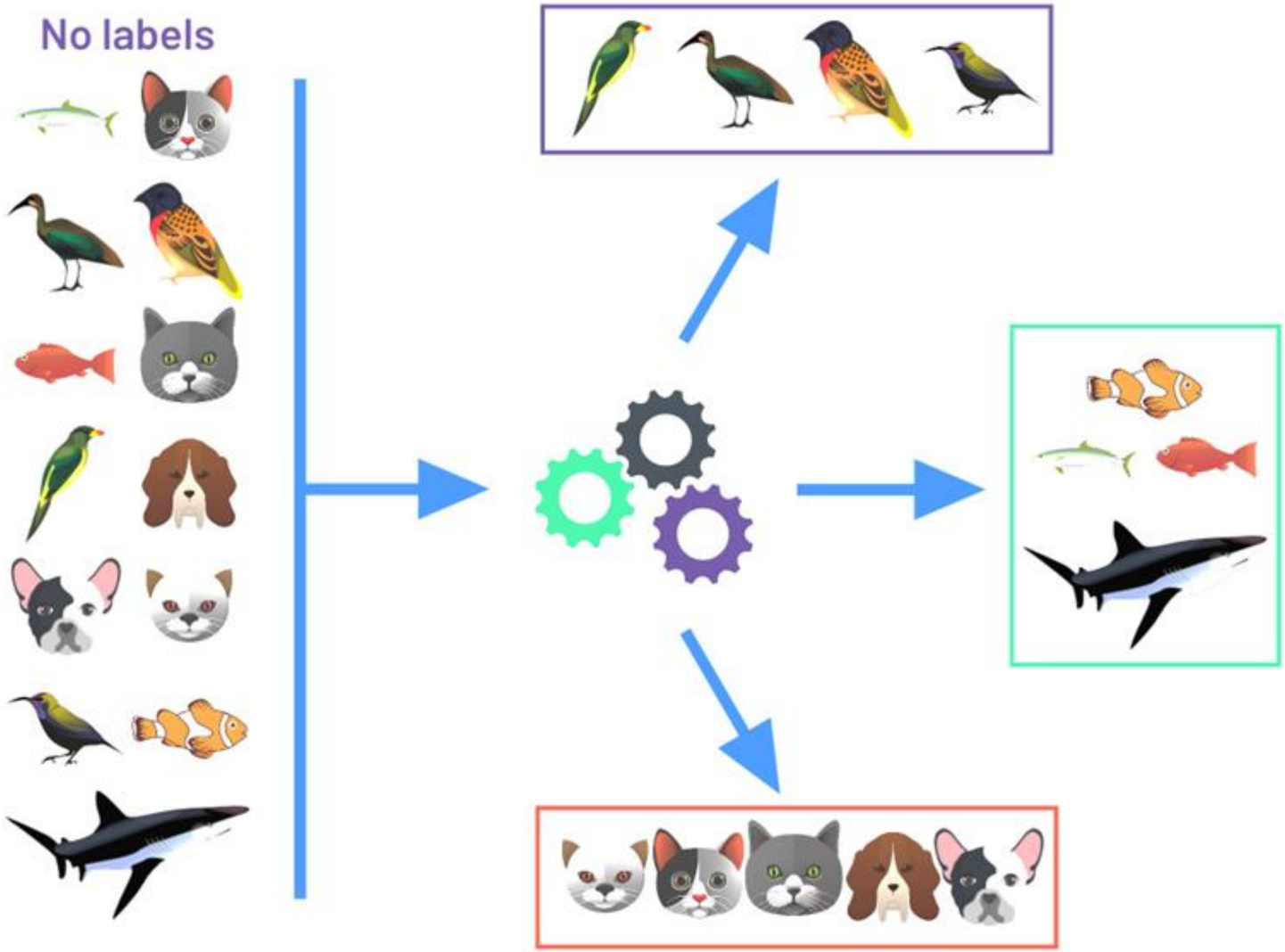


GIỚI THIỆU

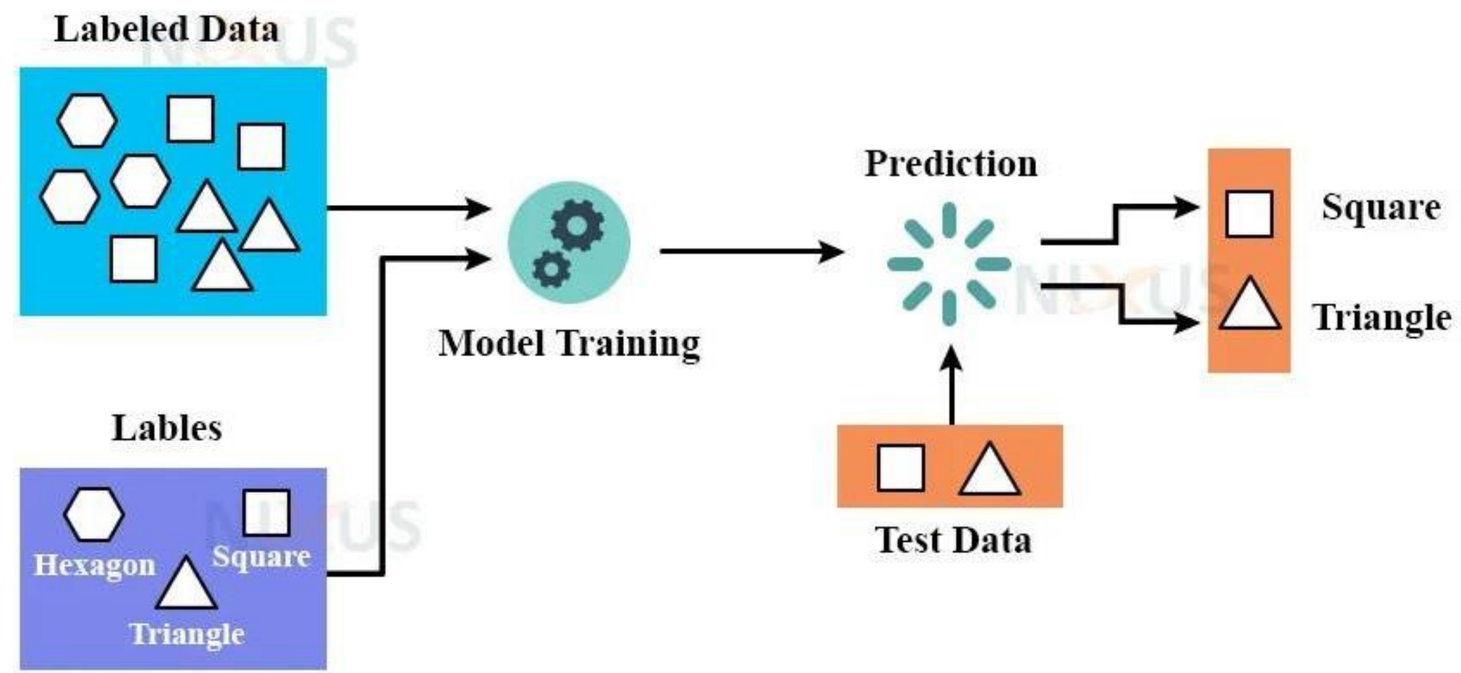
- **Học có giám sát (Supervised Learning)**



- **Học không giám sát (Unsupervised Learning)**



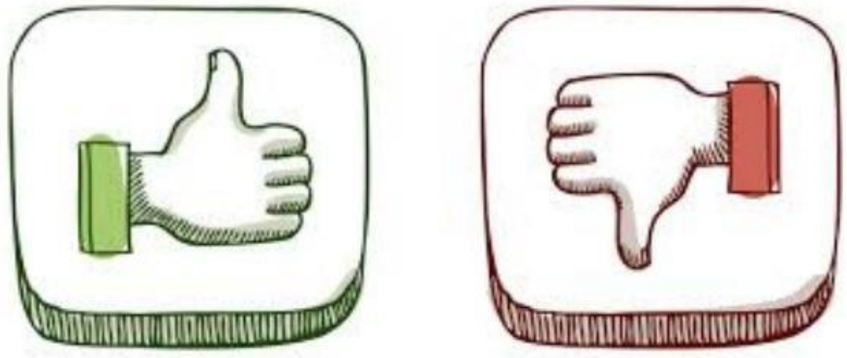
- **Thuật toán học có giám sát (Supervised Learning)** là một phương pháp trong lĩnh vực học máy (machine learning), trong đó mô hình được huấn luyện bằng cách sử dụng một bộ dữ liệu đã được gán nhãn sẵn.



- **Ưu điểm và nhược điểm:**

- **Ưu điểm:** Hiệu quả cao khi có dữ liệu huấn luyện chất lượng, mô hình có thể đưa ra các dự đoán chính xác.

- **Nhược điểm:** Yêu cầu bộ dữ liệu đã được gán nhãn, điều này đôi khi khó khăn và tốn kém. Mô hình cũng có thể gặp vấn đề khi phải làm việc với các dữ liệu không nhìn thấy trong quá trình huấn luyện.



Thuật toán

K-Nearest Neighbors

- **K-Nearest Neighbors (KNN): Là thuật toán học có giám sát (Supervised Learning)**
- **Ý tưởng: Lấy ý tưởng từ phương pháp nhìn bài bạn**

- Nhìn bài để lấy kết quả chép bài?

- Ở đây, ***K*** là số người xung quanh bị nhìn bài.

Ref: https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn

k-Nearest Neighbors

The *k*-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training dataset. To make a prediction for a new data point, the algorithm finds the closest data points in the training dataset—its "nearest neighbors."

k-Neighbors classification

In its simplest version, the *k*-NN algorithm only considers exactly one nearest neighbor, which is the closest training data point to the point we want to make a prediction for. The prediction is then simply the known output for this training point. Figure 2-4 illustrates this for the case of classification on the `forge` dataset:

Ví dụ về thời tiết đi chơi

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

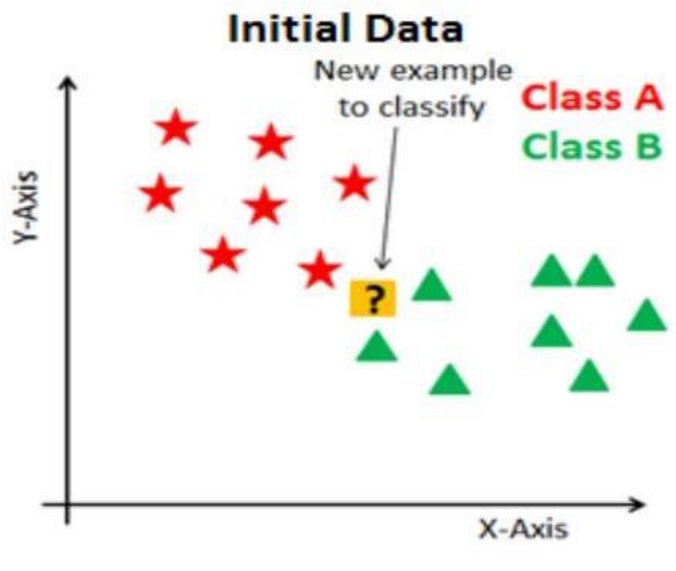
Ví dụ về xếp loại học tập

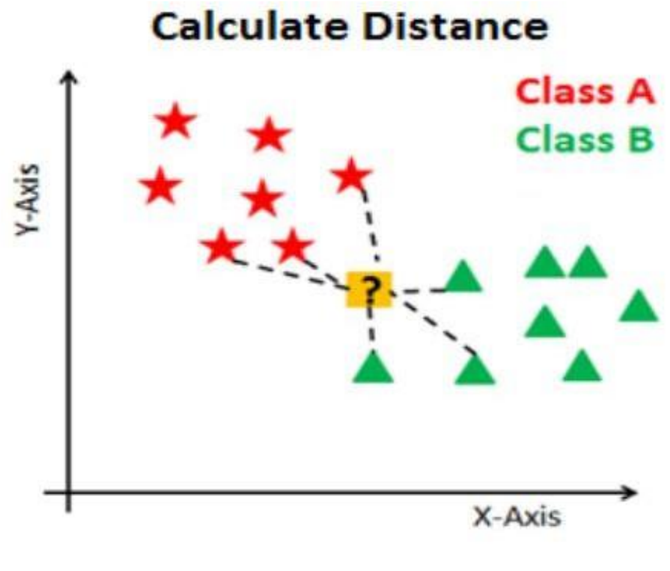
Điểm của tôi: 7

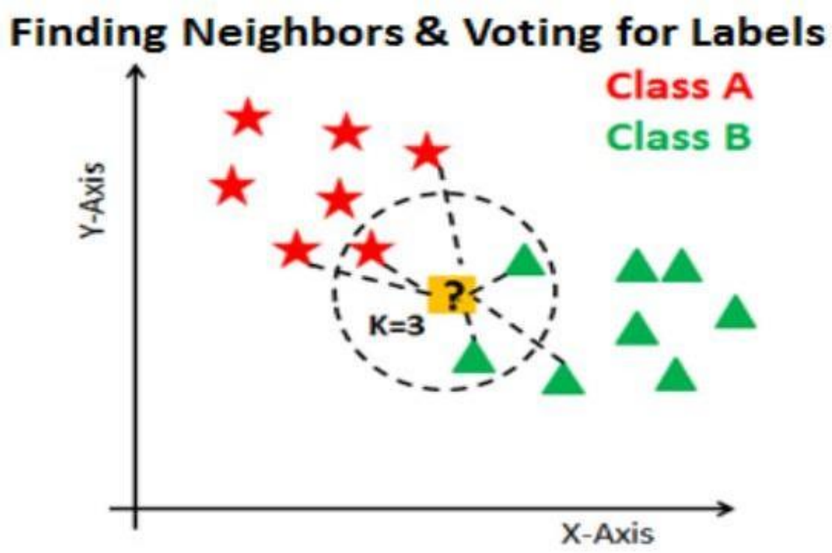
Điểm của bạn tôi:

- 7.1 => Khá
- 7.2 => Khá
- 6.7 => Khá
- 6.6 => Khá
- 6.4 => Trung bình

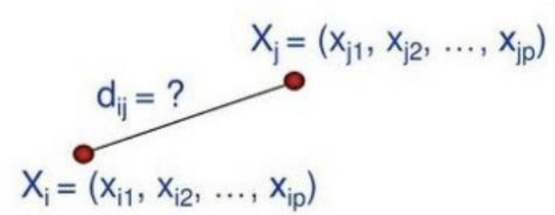
MÔ TẢ HOẠT ĐỘNG THUẬT TOÁN







- **Xác định khoảng cách?**



- **Minkowski distance**

$$d(i,j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q}$$

1st dimension 2nd dimension pth dimension

- **Euclidean distance**

q = 2

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2}$$

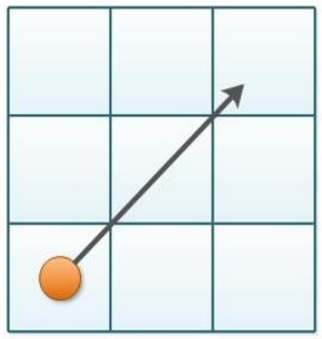
- **Manhattan distance**

q = 1

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

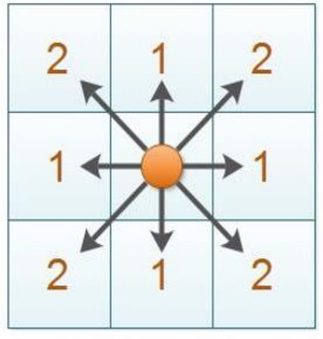
- **Xác định khoảng cách?**

Euclidean Distance



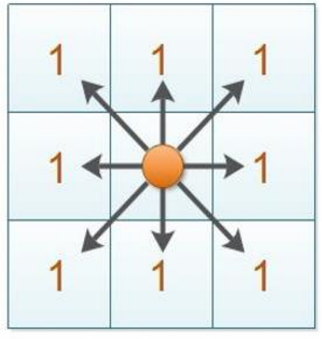
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan Distance

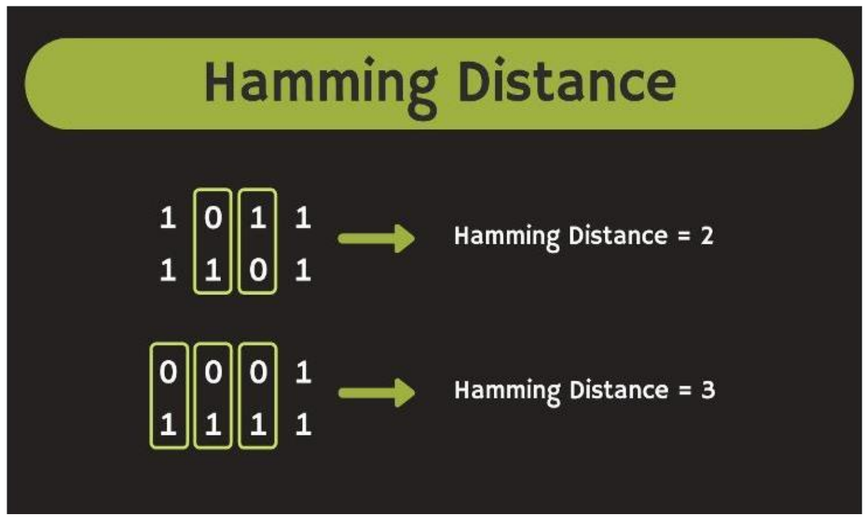


$$|x_1 - x_2| + |y_1 - y_2|$$

Chebyshev Distance

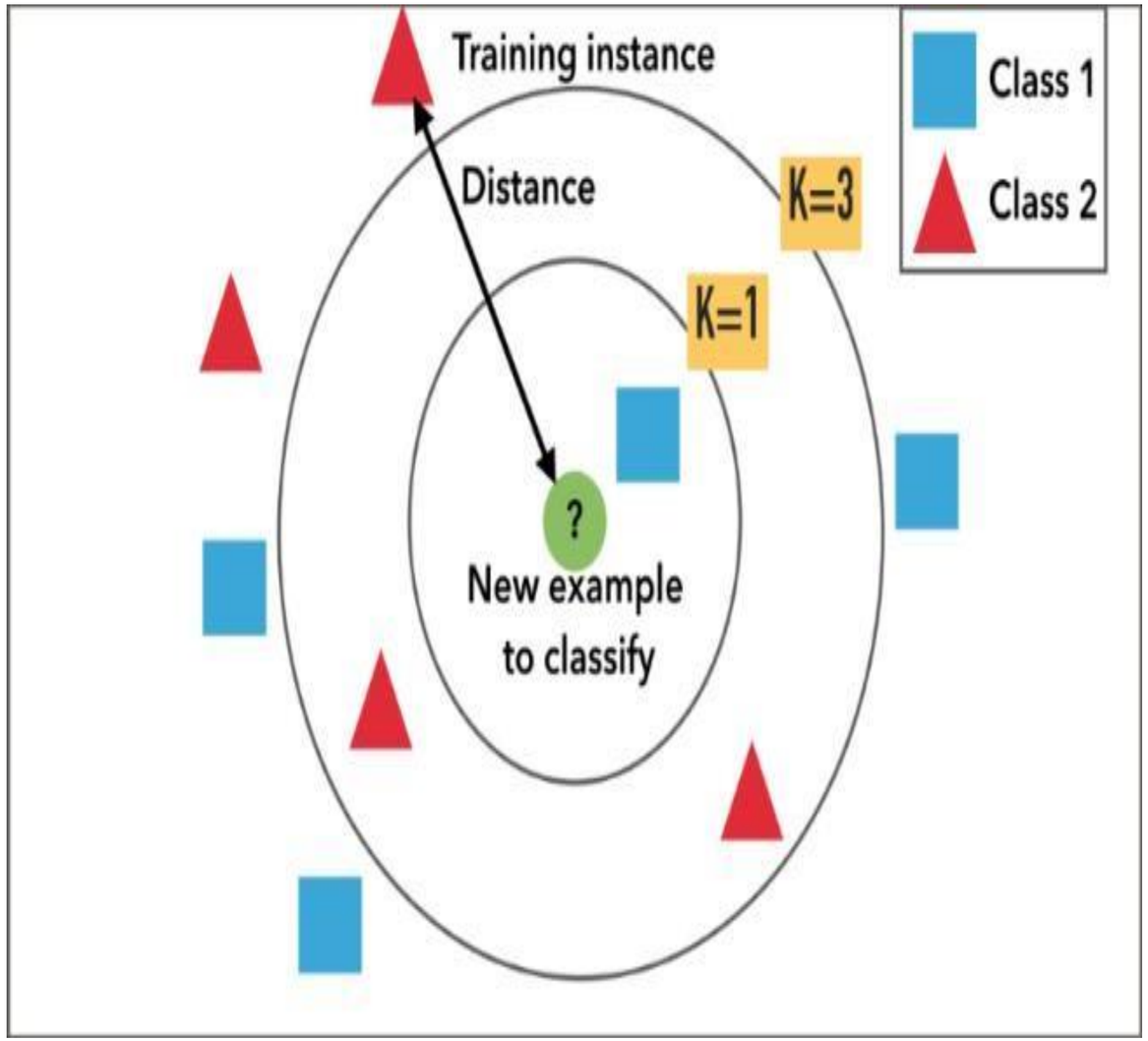


$$\max(|x_1 - x_2|, |y_1 - y_2|)$$



ĐÁNH GIÁ ĐỘ CHÍNH XÁC

- **K = ?**
- **Accuracy = ?**



Tự lấy ví dụ và xây dựng dữ liệu

- **Input: *K, trainset, pointX***
- **Output: class of *pointX***

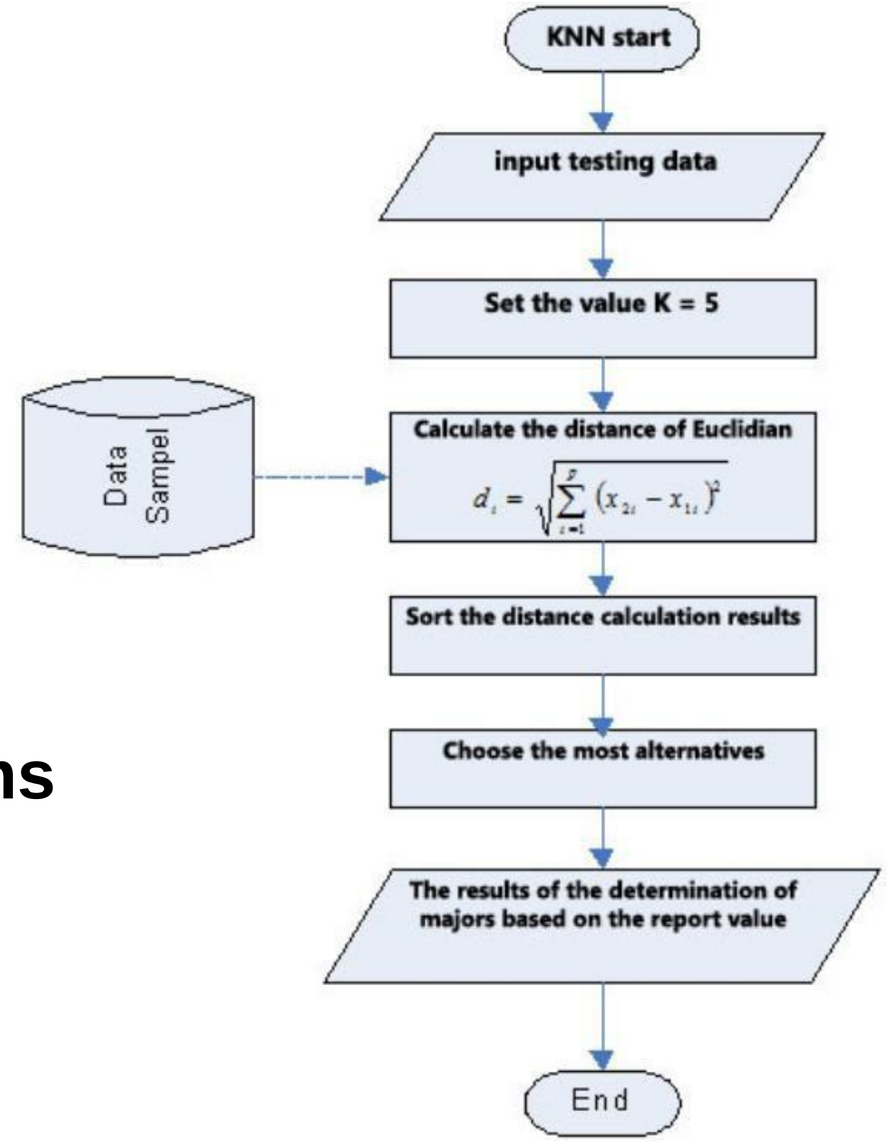
1. **For each *itemN* in *trainSet***

   - Calculate distance of *itemN* and *pointX*

2. **Find *K* items of shortest distance**

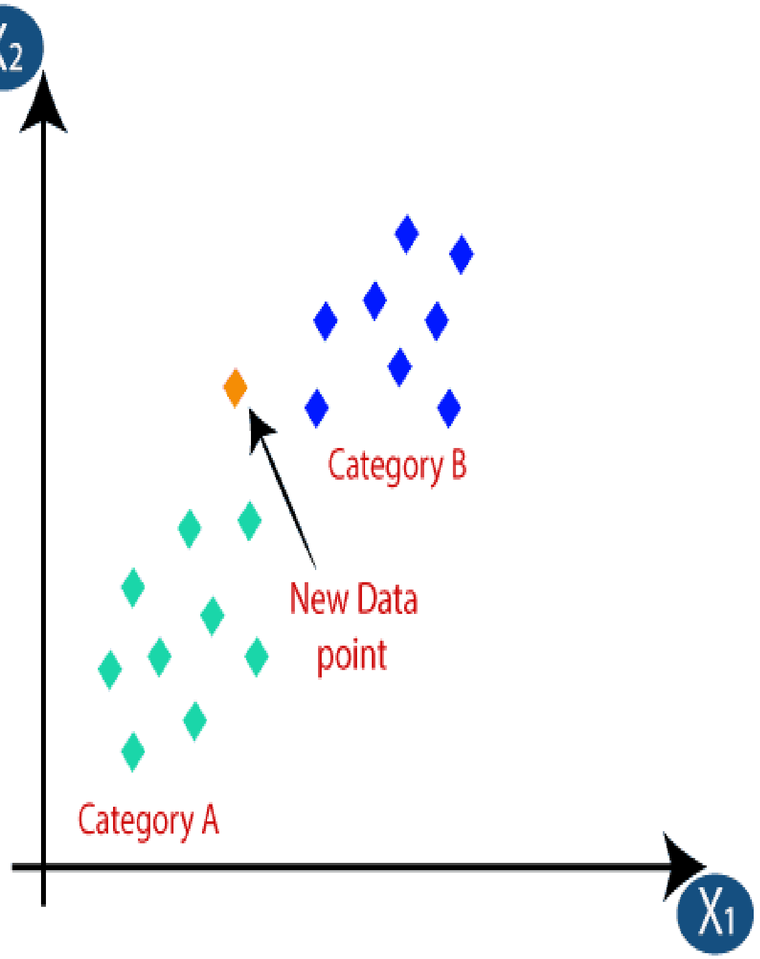
3. **Find class of the most occur items in *K* items**

4. **Set found class to *pointX***



Bước 1

Lựa chọn K - Số Nearest neighbors

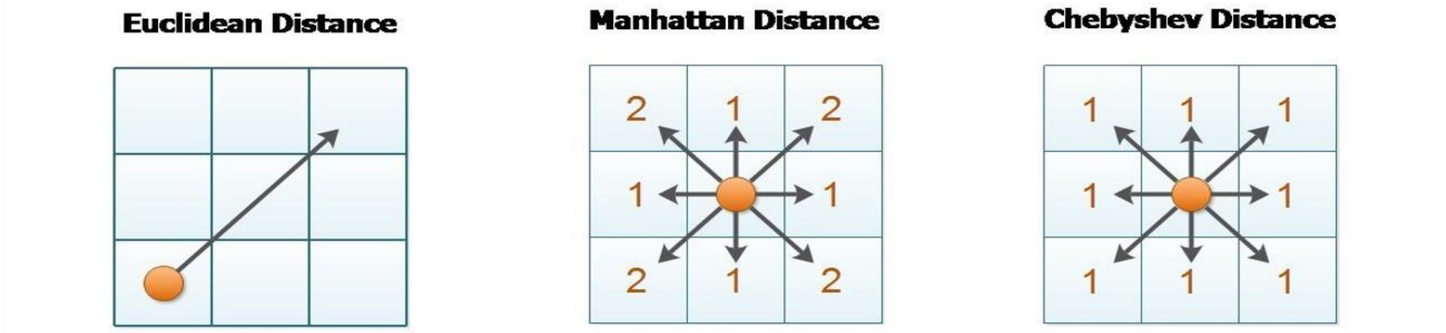


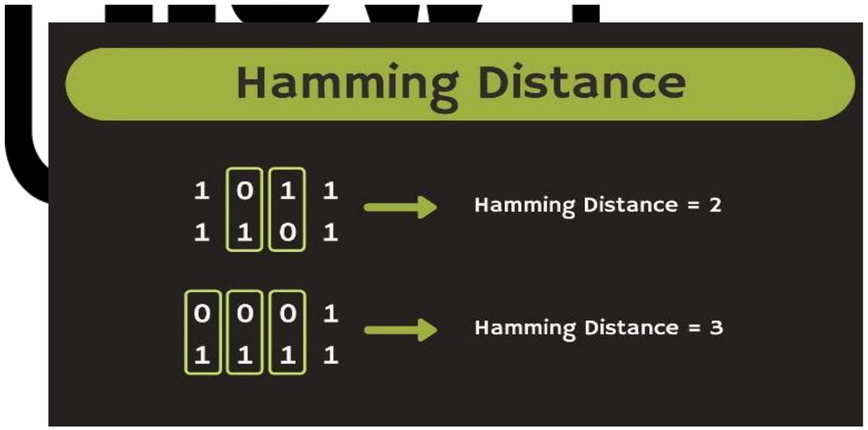
K = 5

Làm sao để tìm được 5 nearest neighbors

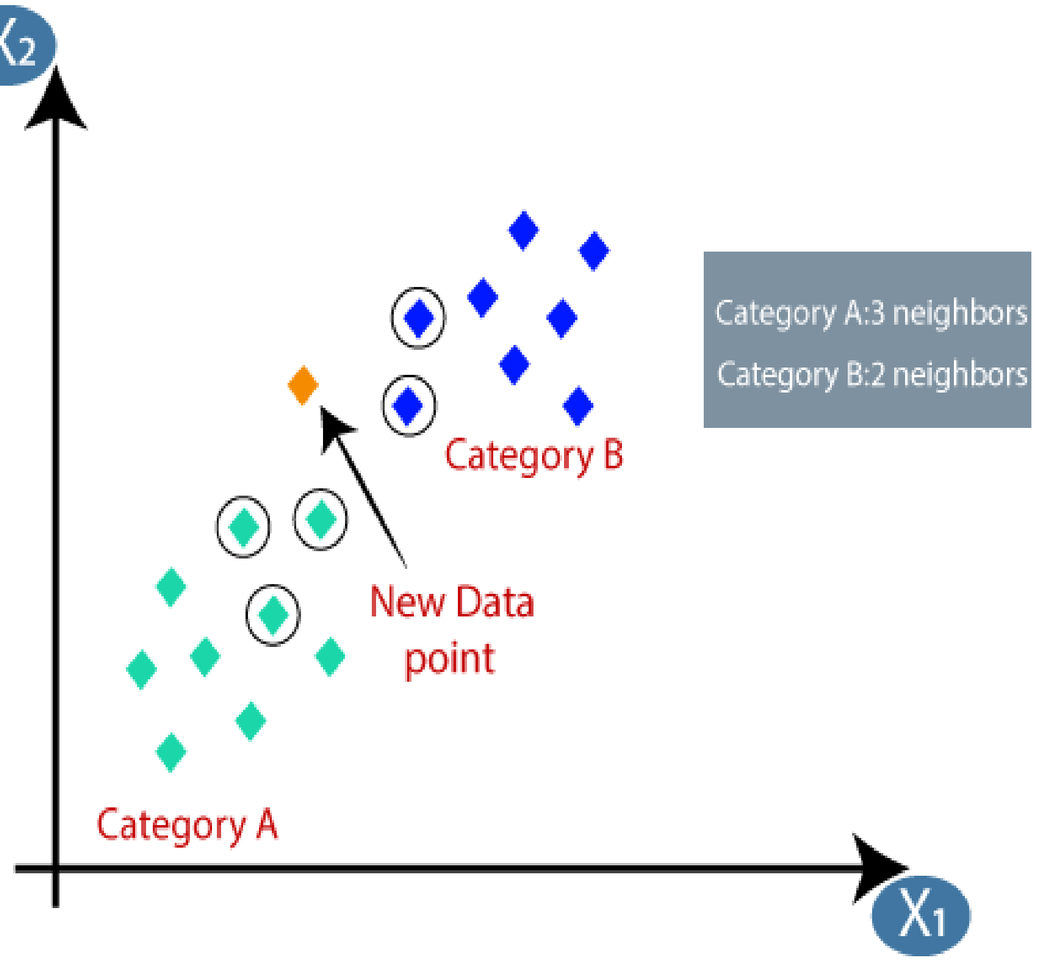
Bước 2

Tính khoảng cách giữa test_data_point và toàn bộ dữ liệu huấn luyện

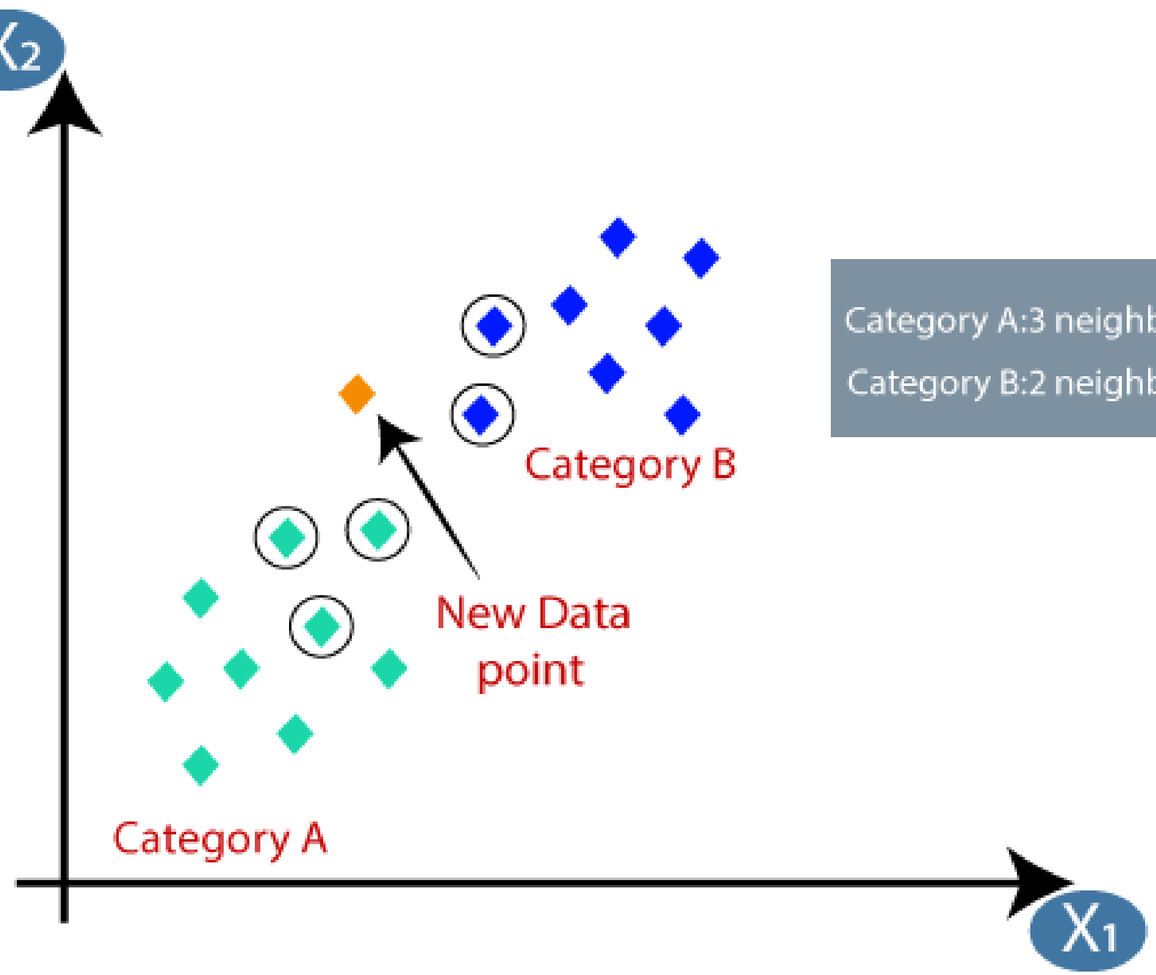




Bước 3: Chọn ra 5 nearest neighbors dựa vào khoảng cách tính được



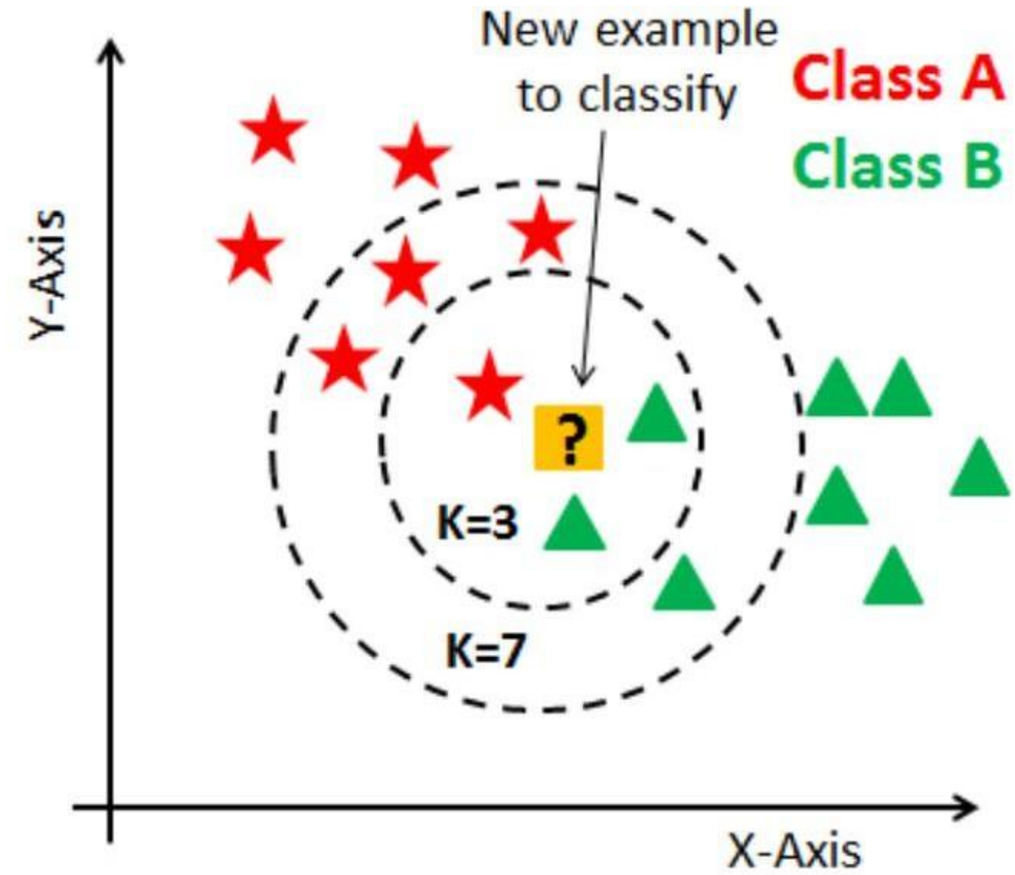
Bước 4: Chọn nhãn chiếm đa số làm kết quả dự đoán



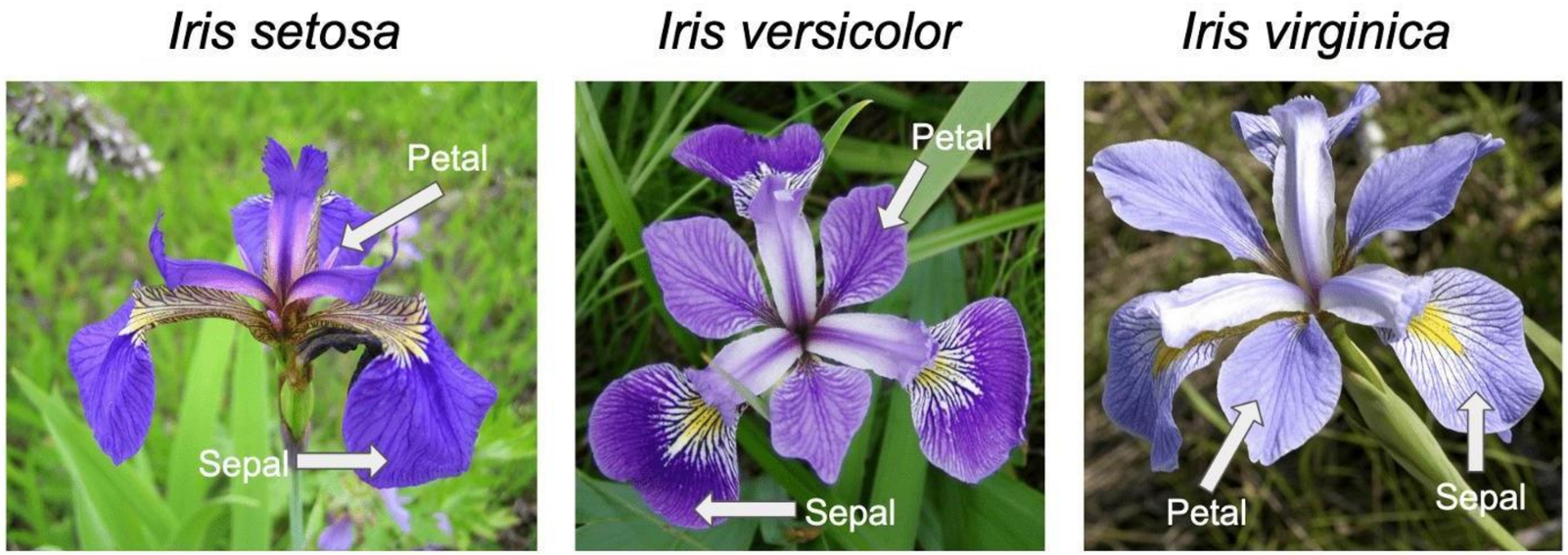
- Số lượng phần tử tham chiếu giữa các lớp

- Sự mất cân đối dữ liệu huấn luyện giữa các lớp

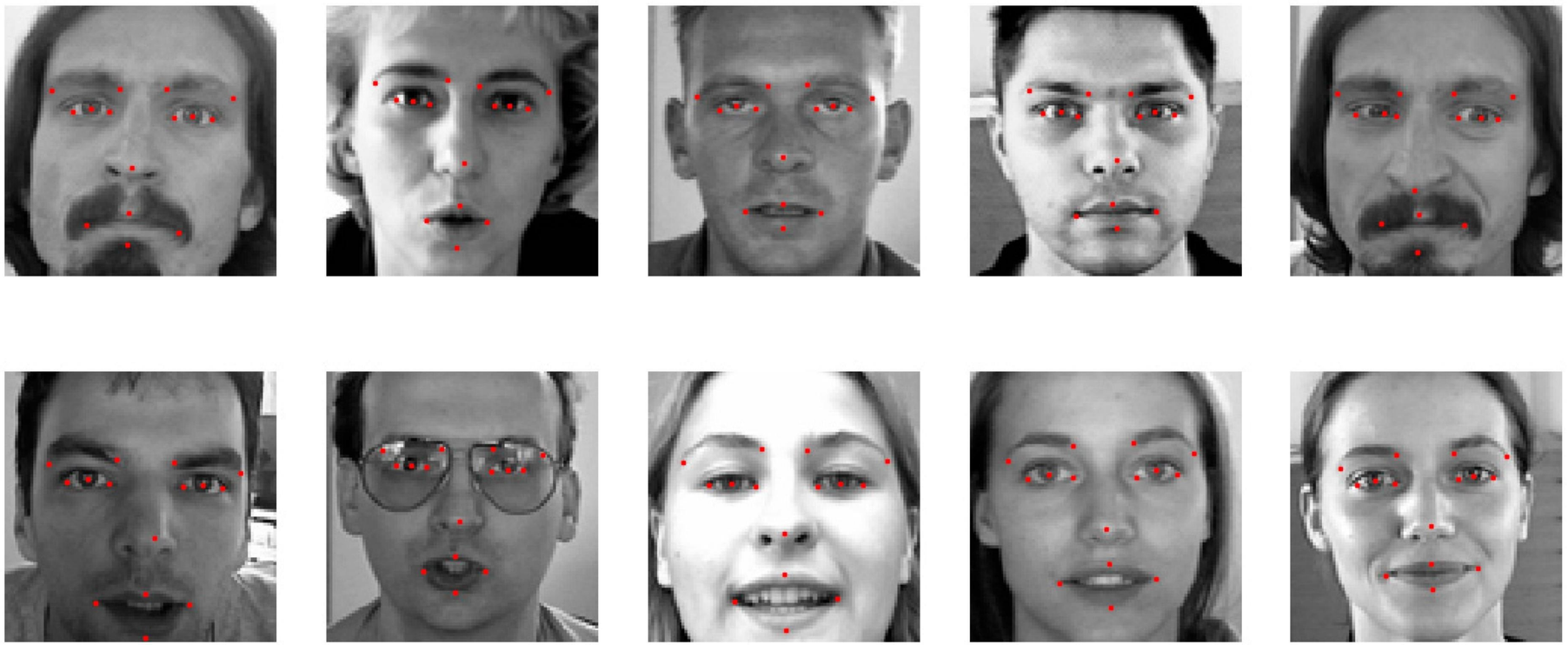
- Thời gian thực hiện



Lập trình với kho dữ liệu Iris



LẬP TRÌNH NHẬN DIỆN KHUÔN MẶT



SUMMARY

- Thuật toán học có giám sát
- Thuật toán K-NN
- Luyện tập


PRACTICE
ENGLISH
EVERYDAY
30 Minutes

Analyzing KNeighborsClassifier

For two-dimensional datasets, we can also illustrate the prediction for all possible test points in the xy-plane. We color the plane according to the class that would be assigned to a point in this region. This lets us view the *decision boundary*, which is the divide between where the algorithm assigns class 0 versus where it assigns class 1.

The following code produces the visualizations of the decision boundaries for one, three, and nine neighbors shown in Figure 2-6:

As you can see on the left in the figure, using a single neighbor results in a decision boundary that follows the training data closely. Considering more and more neighbors leads to a smoother decision boundary. A smoother boundary corresponds to a simpler model. In other words, using few neighbors corresponds to high model complexity (as shown on the right side of Figure 2-1), and using many neighbors corresponds to low model complexity (as shown on the left side of Figure 2-1). If you consider the extreme case where the number of neighbors is the number of all data points in the training set, each test point would have exactly the same neighbors (all training points) and all predictions would be the same: the class that is most frequent in the training set.

Chuẩn bị bài học sau

- **Thuật toán Naive Bayes**
- **Bài luyện tập tiếng Anh**


Thank
you!