AI VIET NAM – AIO2024

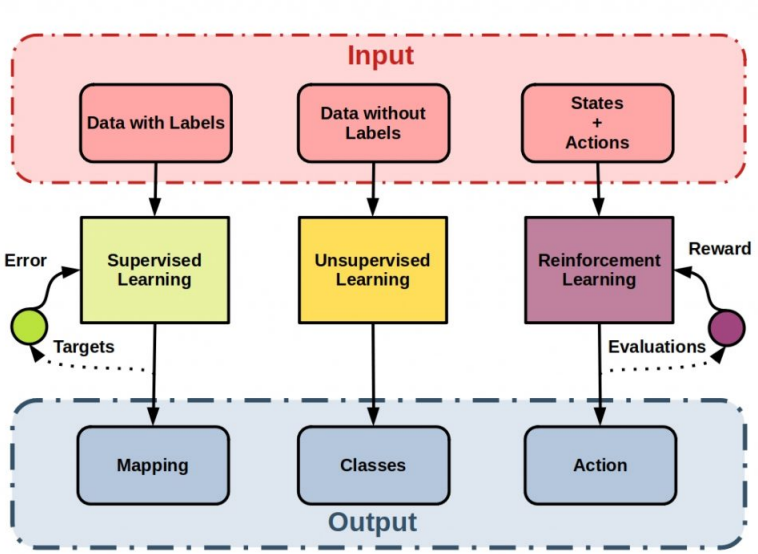
K-Nearest Neighbors - KNN

Ngày 15 tháng 2 năm 2024

Người tóm tắt	Ngọc Trúc
Nguồn dữ liệu:	K-Nearest Neighbor - KNN
Từ khóa:	Machine Learning,KNN, K-Nearest Neighbor, Brute force, K-D Tree, Ball Tree

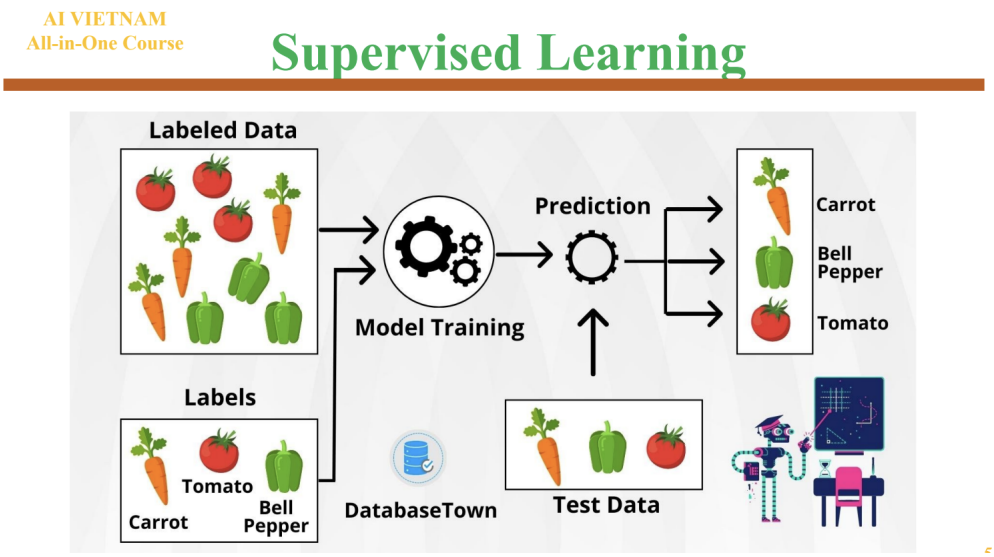
1 Overview Machine Learning

- Có 4 phương pháp máy học chính: Học máy có giám sát (Supervised Learning), Học máy không giám sát (Unsupervised Learning), Học máy tăng cường (Reinforcement Learning) và Học máy bán giám sát (Semi-supervised learning)



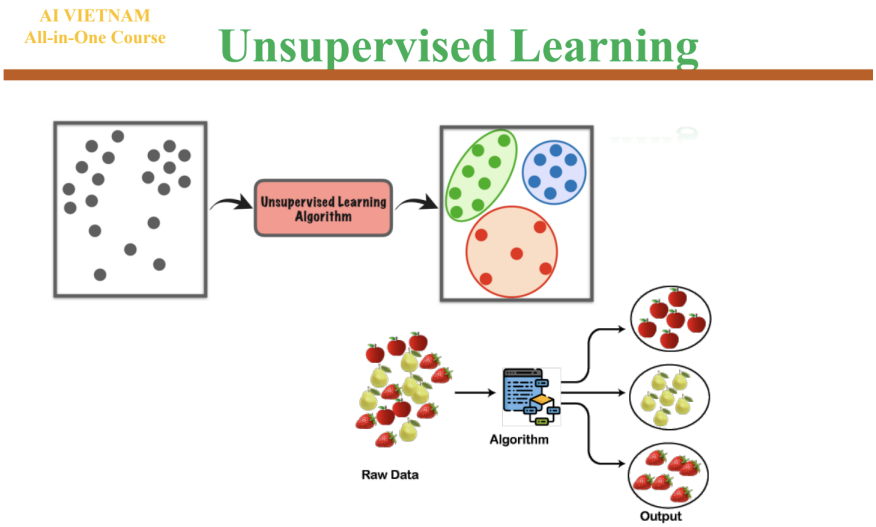
Hình 1: So sánh input-output của 3 loại học máy chính

- Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước.
  - Ví dụ: Thuật toán dò các khuôn mặt trong một bức ảnh đã được phát triển từ rất lâu. Thời gian đầu, facebook sử dụng thuật toán này để chỉ ra các khuôn mặt trong một bức ảnh và yêu cầu người dùng tag friends - tức gán nhãn cho mỗi khuôn mặt. Số lượng cặp dữ liệu (khuôn mặt, tên người) càng lớn, độ chính xác ở những lần tự động tag tiếp theo sẽ càng lớn.



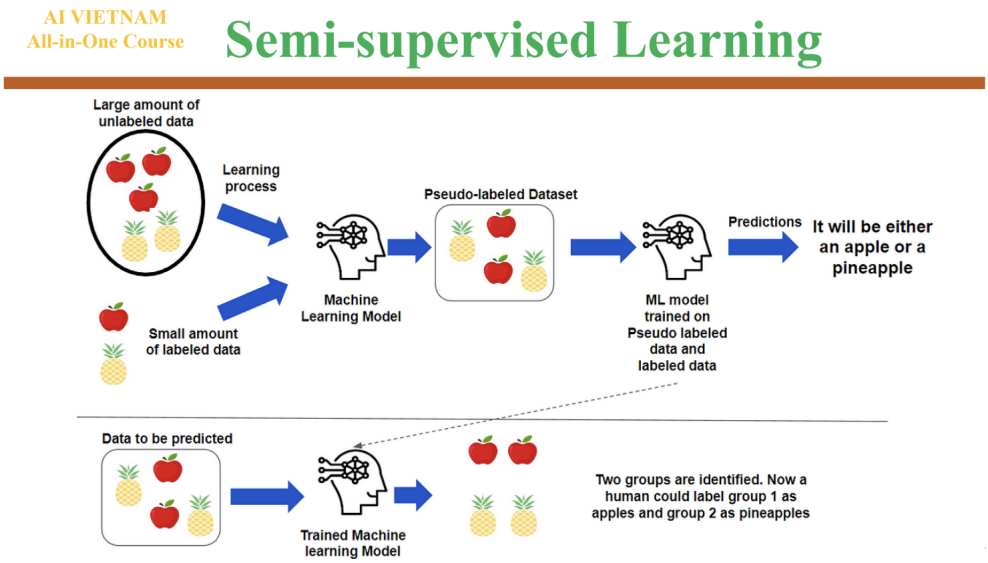
Hình 2: Supervised Learning

- Thuật toán Unsupervised Learning không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào, nó sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.
  - Ví dụ bài toán phân nhóm (Clustering) khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.



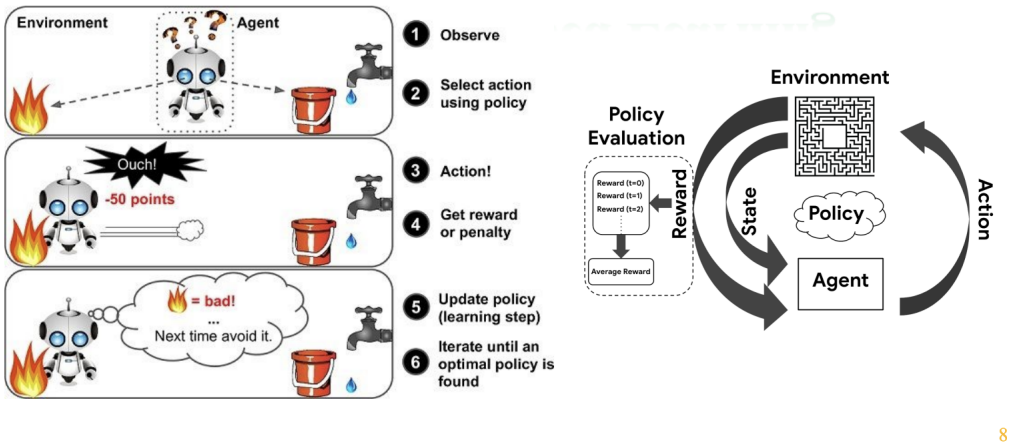
Hình 3: Unsupervised Learning

- Thuật toán Semi-Supervised Learning (Học bán giám sát) là các bài toán khi chúng ta có một lượng lớn dữ liệu X nhưng chỉ một phần trong chúng được gán nhãn. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên (Supervised Learning và Unsupervised Learning)
  - Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet.



Hình 4: Semi Supervised Learning

- Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance)



Hình 5: Reinforcement Learning

2 KNN Motivation

- Lazy Motivation hoạt động bằng cách ghi nhớ dữ liệu huấn luyện thay vì xây dựng một mô hình chung. Khi nhận được một truy vấn mới, thuật toán này sẽ truy xuất các trường hợp tương tự từ tập huấn luyện và sử dụng chúng để tạo dự đoán.

- Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học. Một trong những thuật toán Lazy Motivation phổ biến nhất là KNN (K-Nearest Neighbors)

- KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát (Supervised Learning) đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy, mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới. Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

- Khác với Lazy Motivation, Eager learning hay còn gọi là học dựa trên mô hình, là phương pháp trong học máy xây dựng một mô hình tổng quát từ dữ liệu huấn luyện, cố gắng khám phá mối quan hệ và mẫu ẩn.

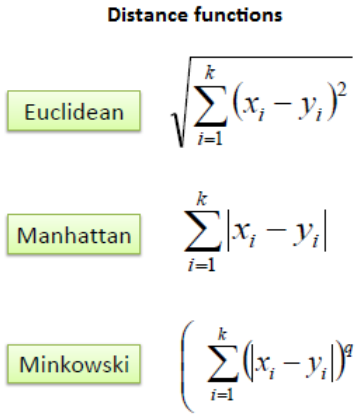
Feature	Lazy Learning	Eager Learning
Generalization	Adapts quickly	Less flexible
Model complexity	Less complex	More complex
Training time	Minimal	Longer
Prediction time	Slower	Faster
Memory usage	Higher	Lower
Interpretability	More interpretable	Varies
Online Learning	Well-suited	Less suitable
Robustness	Less robust	More robust

Hình 6: Lazy Learning and Eager Learning

3 KNN for Classification

- Các bước để thực hiện thuật toán KNN:
  - Bước 1: Xác định số láng giềng gần nhất (K)

– Bước 2: Tính toán khoảng cách
Có 3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính, thông dụng nhất là cách tính Euclid



Hình 7: Công thức tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính

– Bước 3: Xác định K láng giềng gần nhất
– Bước 4: Phiếu bầu và xác định nhãn dự đoán

4 How to select k in KNN

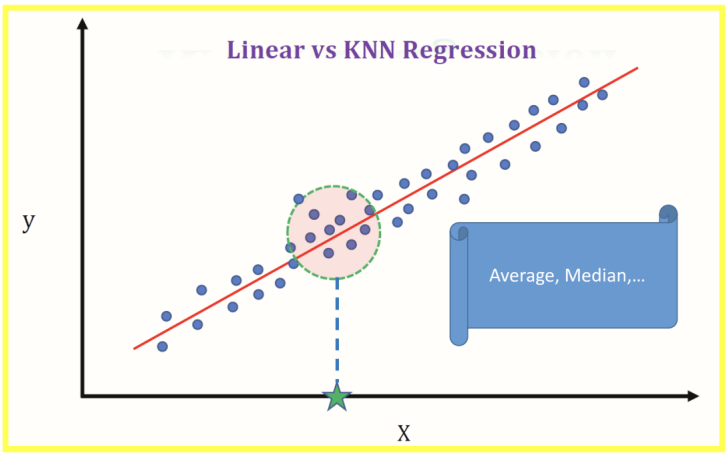
- Giá trị k là tham số ảnh hưởng đến độ phức tạp của mô hình.
  – Nếu k nhỏ, mô hình phức tạp hơn, sai số khớp trên mẫu xây dựng nhỏ hơn, dễ bị overfitting.
  – Nếu k lớn, kết quả dự báo ổn định hơn, do có "sự bình chọn giữa nhiều quan sát.

- Để chọn giá trị của k trong thuật toán KNN (K-Nearest Neighbors), có thể sử dụng các phương pháp thử nghiệm và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.
  – Tuy nhiên, việc đánh giá hiệu suất mô hình không chỉ dựa trên độ chính xác (accuracy) mà còn phải sử dụng các số đo khác như precision, recall thông qua Confusion Matrix. Bằng cách sử dụng Confusion Matrix, có thể xác định số lượng dự đoán đúng và sai cho mỗi lớp, từ đó đưa ra cái nhìn tổng quan về hiệu suất của mô hình.

- Việc chọn giá trị k là số lẻ hoặc số chẵn trong thuật toán KNN ảnh hưởng đến quá trình bỏ phiếu (voting) và có thể ảnh hưởng đến kết quả dự đoán của mô hình
  – Khi k là số lẻ, quá trình bỏ phiếu sẽ luôn cho ra kết quả với số phiếu bầu cao hơn, giúp việc kết luận trở nên dễ dàng hơn.
  – Tuy nhiên, khi k là số chẵn, có thể xảy ra tình trạng mỗi lớp nhận được số phiếu bầu bằng nhau, dẫn đến kết quả dự đoán không hiệu quả. Để khắc phục điều này, chúng ta có thể sử dụng trọng số (weights), bao gồm uniform weight, distance weight, và customize weight, để điều chỉnh quá trình bỏ phiếu dựa trên mức độ quan trọng của mỗi láng giềng. Ví dụ, khi sử dụng distance weight, mô hình sẽ xem xét cả khoảng cách giữa các láng giềng để xác định trọng số cho các phiếu bầu.

5 KNN for Regression

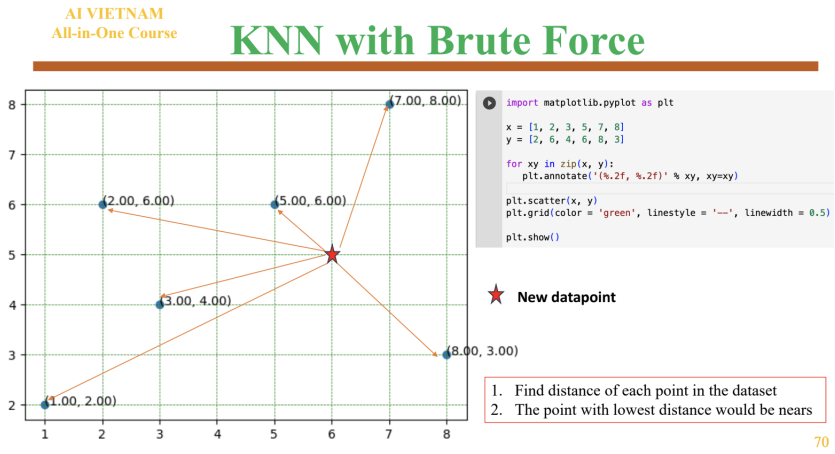
- Trong bài toán regression, thuật toán K-Nearest Neighbors (KNN) cũng có thể được sử dụng để dự đoán giá trị của một biến liên tục dựa trên các giá trị của các biến đầu vào. Quá trình hoạt động của KNN trong bài toán regression tương tự như trong bài toán phân loại, nhưng có một số điểm khác biệt chính:
  - Sau khi chọn K lân cận, thay vì sử dụng bước bỏ phiếu như trong bài toán phân loại, chúng ta thực hiện bước tính trung bình (hoặc trung vị) của các giá trị đầu ra của các điểm lân cận. Điều này có nghĩa là chúng ta tính trung bình của các giá trị của biến mục tiêu (target variable) cho các điểm dữ liệu lân cận và sử dụng giá trị này làm dự đoán cho điểm mới.



Hình 8: KNN for regression

6 KNN with Brute force

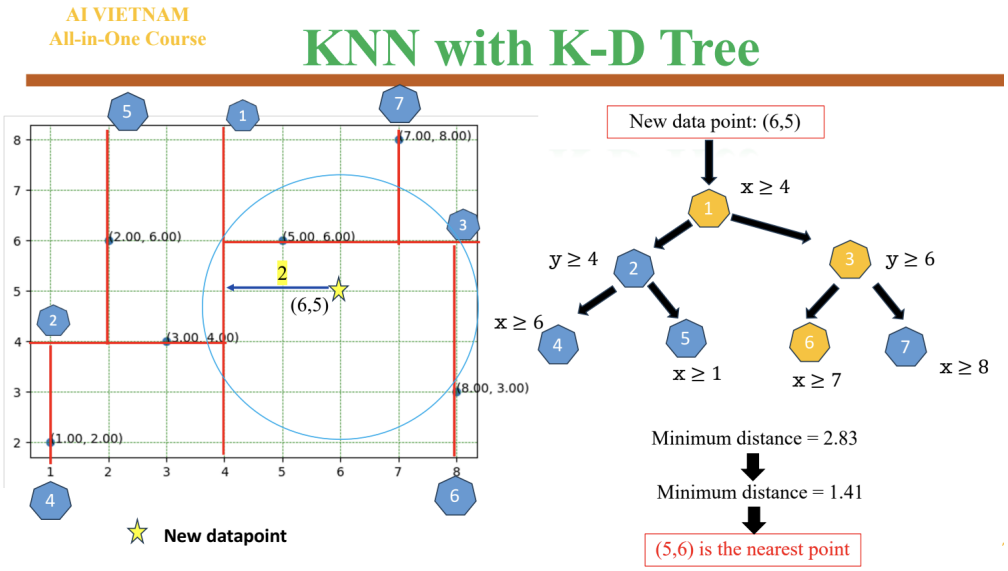
- KNN Brute Force: Mỗi điểm trong tập huấn luyện được lưu và dùng tính khoảng cách. Hiệu quả cho tập nhỏ, nhưng không hiệu quả cho dữ liệu lớn hoặc có số chiều cao.



Hình 9: KNN with Brute force

7 KNN with K-D Tree

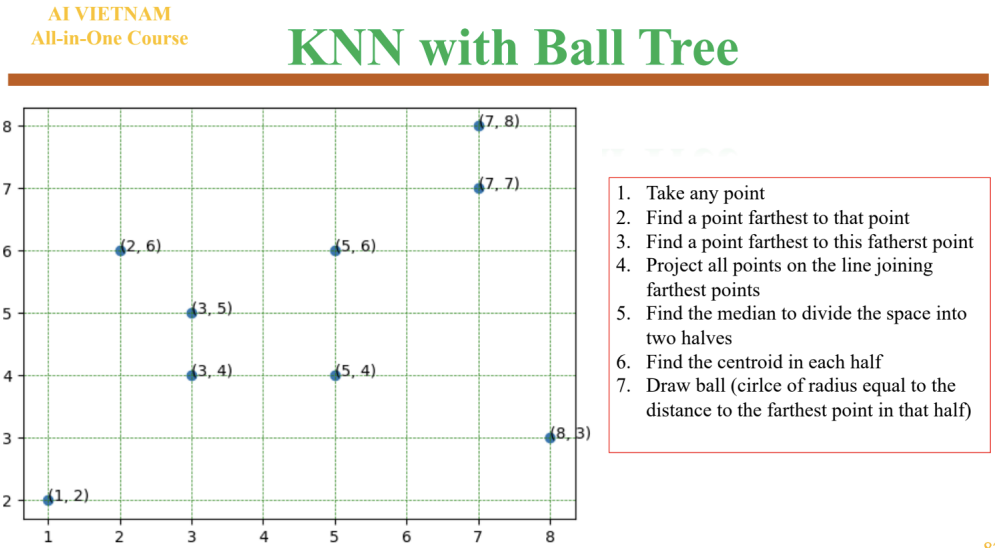
- KNN with K-D trees: Sử dụng cấu trúc dữ liệu K-D tree để tăng tốc quá trình tìm kiếm láng giềng. Hiệu quả hơn Brute Force đối với dữ liệu lớn hoặc có số chiều cao.



Hình 10: KNN with K-D Tree

8 KNN with Ball Tree

- Tương tự như K-D Tree nhưng đối với Ball Tree, hướng tiếp cận và cách tính sẽ có sự khác biệt, cụ thể được thể hiện trong hình minh họa.



Hình 11: KNN with Ball Tree