Machine Learning là gì?

Machine Learning (ML) là một nhánh của Artificial Intelligence (AI) cho phép máy tính học từ dữ liệu và đưa ra quyết định hay dự đoán mà không cần được lập trình một cách tường minh.

Thay vì viết ra từng bước quy tắc, ta đưa cho máy tính dữ liệu và ví dụ. Máy tính tự tìm ra các patterns trong dữ liệu, rồi dùng chúng để cho kết quả chính xác trên dữ liệu mới, chưa từng thấy.

Ví dụ: Phát hiện Email Spam

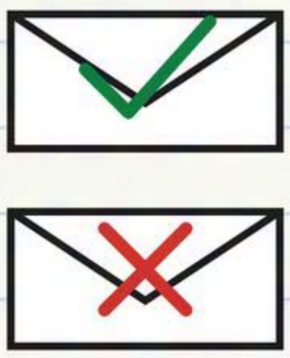
1. Input (Dữ liệu)

Nhiều email (spam hoặc không spam) kèm theo nhãn ví dụ

2. Learn (Mô hình ML)

Mô hình ML học các patterns từ dữ liệu

3. Output (Dự đoán)



Mô hình dự đoán một email mới là spam hay không spam

Tóm lại, Machine Learning giúp máy tính học từ dữ liệu và tự động cải thiện theo kinh nghiệm.

Machine Learning hoạt động như thế nào

Machine Learning cho phép máy tính học từ dữ liệu và cải thiện hiệu năng trên một tác vụ mà không cần được lập trình một cách tường minh.

Thay vì tuân theo các quy tắc cố định, mô hình ML tìm ra patterns trong dữ liệu và dùng chúng để đưa ra dự đoán hoặc quyết định.

Cách hoạt động - Từng bước một

1. Thu thập dữ liệu : Gom dữ liệu thô từ nhiều nguồn liên quan đến bài toán.

2. Chuẩn bị dữ liệu : Làm sạch, sắp xếp và biến đổi dữ liệu để sẵn sàng đưa vào sử dụng.

3. Huấn luyện mô hình : Thuật toán học các patterns từ dữ liệu huấn luyện.

4. Đưa ra dự đoán : Mô hình đã huấn luyện dùng các patterns đã học để dự đoán trên dữ liệu mới.

5. Đánh giá & Cải thiện : Kiểm tra độ chính xác của mô hình và cải thiện nó bằng nhiều dữ liệu hơn cùng việc tinh chỉnh.

Quá trình này giúp mô hình ngày càng chính xác và đáng tin cậy hơn khi nó học từ nhiều dữ liệu và kinh nghiệm hơn.

Các loại Machine Learning

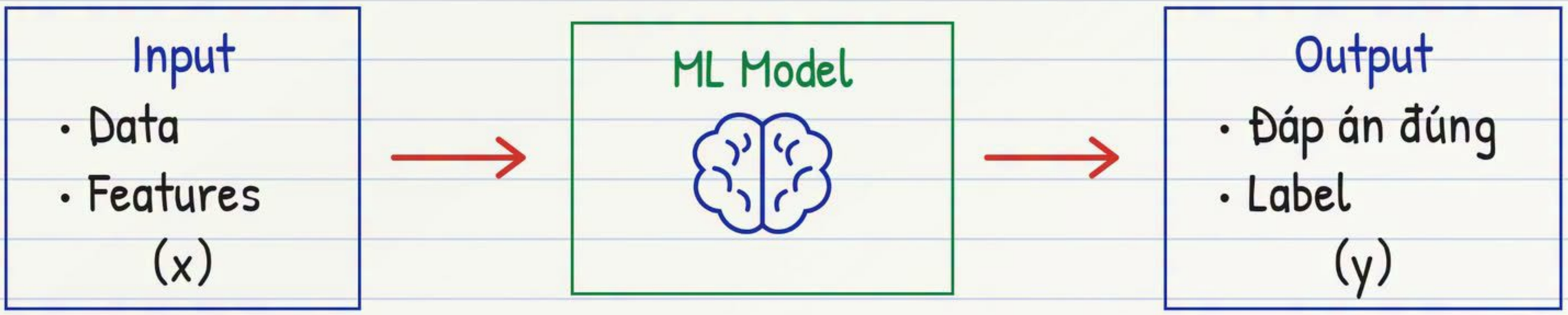
Machine Learning chủ yếu được chia thành ba loại dựa trên cách mô hình học từ dữ liệu.

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

1. Supervised Learning

Trong Supervised Learning, mô hình được huấn luyện trên dữ liệu có nhãn (labeled data). Nghĩa là dữ liệu huấn luyện chứa cả input lẫn output đúng.
Mô hình học mối quan hệ giữa input và output, rồi dùng kiến thức đó để dự đoán output cho dữ liệu mới, chưa từng thấy.

Cách hoạt động:

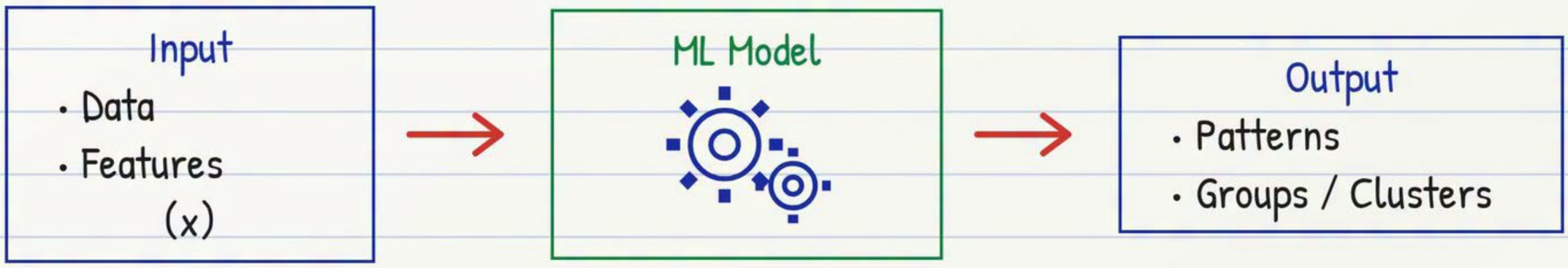


Mô hình được huấn luyện qua rất nhiều ví dụ có đáp án đúng và cố gắng tối thiểu hoá sai số.

2. Unsupervised Learning

Trong Unsupervised Learning, mô hình được huấn luyện trên dữ liệu không nhãn (unlabeled data). Nghĩa là dữ liệu huấn luyện chỉ có input mà không có output đúng.
Mô hình tự tìm ra các patterns hoặc cấu trúc ẩn trong dữ liệu.
Nó chủ yếu được dùng để clustering, gom nhóm, hoặc tìm mối quan hệ trong dữ liệu.

Cách hoạt động:

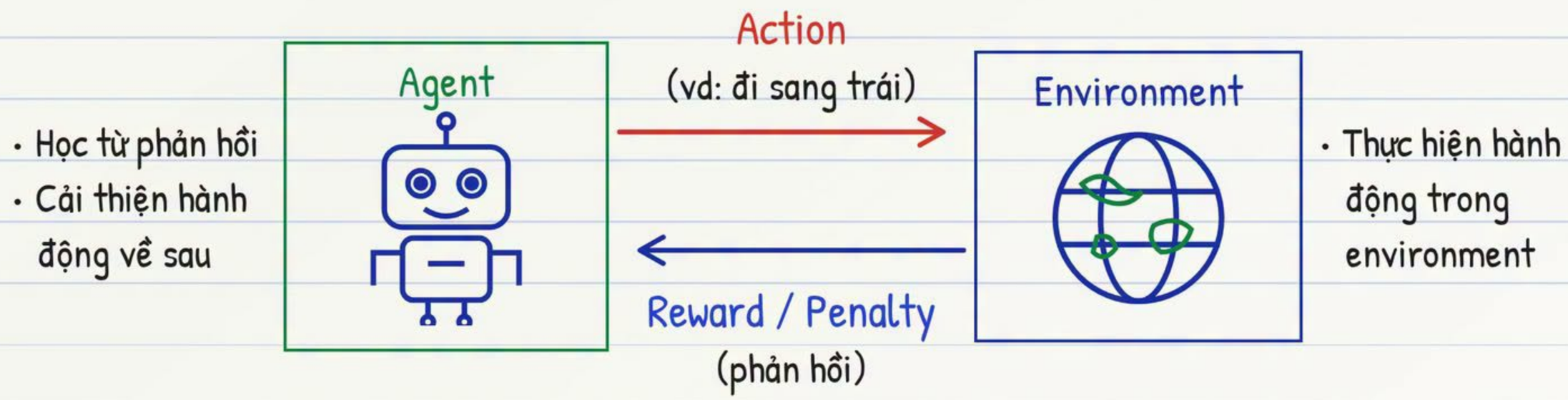


Mô hình phân tích dữ liệu và khám phá ra các patterns hoặc nhóm mà không cần bất kỳ hướng dẫn nào.

3. Reinforcement Learning

Trong Reinforcement Learning, mô hình (gọi là agent) học bằng cách tương tác với một environment. Agent thực hiện hành động, nhận reward cho hành động tốt và penalty cho hành động xấu. Theo thời gian, nó học được cách tốt nhất để đạt reward lớn nhất.

Cách hoạt động:

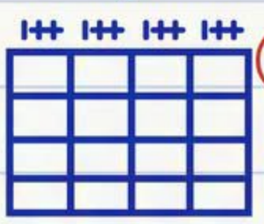


Agent học được một chiến lược mang lại reward cao nhất sau rất nhiều lần thử.

Quy trình Machine Learning

Quy trình Machine Learning là một chuỗi các bước được thực hiện để xây dựng một mô hình machine learning. Nó giúp ta đi từ dữ liệu thô đến những dự đoán hữu ích một cách có hệ thống.

1. **Thu thập dữ liệu** : Gom dữ liệu thô từ nhiều nguồn liên quan đến bài toán. Dữ liệu tốt là nền tảng của một mô hình tốt.

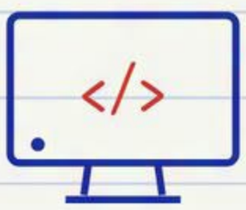


2. **Chuẩn bị dữ liệu** : Làm sạch, sắp xếp và biến đổi dữ liệu để mô hình có thể hiểu được. Bao gồm xử lý missing values, encoding, scaling, v.v.

3. **Huấn luyện mô hình** : Đưa dữ liệu đã chuẩn bị vào thuật toán để nó học các patterns và mối quan hệ trong dữ liệu.

4. **Đánh giá mô hình** : Kiểm tra hiệu năng của mô hình trên dữ liệu mới bằng các chỉ số như accuracy, precision, recall, v.v. để biết mô hình hoạt động tốt đến đâu.

5. **Tinh chỉnh mô hình** : Cải thiện mô hình bằng cách điều chỉnh các parameters và thử những kỹ thuật khác nhau để có kết quả tốt hơn.



6. **Triển khai mô hình** : Đưa mô hình đã huấn luyện vào sử dụng thực tế, nơi nó có thể đưa ra dự đoán trên dữ liệu mới.

7. **Giám sát & Cải thiện** : Liên tục theo dõi hiệu năng của mô hình và cập nhật nó với dữ liệu mới để giữ được độ chính xác theo thời gian.

Machine Learning: Training Data vs Testing Data

Trong Machine Learning, dữ liệu thường được chia thành hai phần: Training Data và Testing Data. Cả hai đều quan trọng, nhưng chúng phục vụ những mục đích khác nhau.

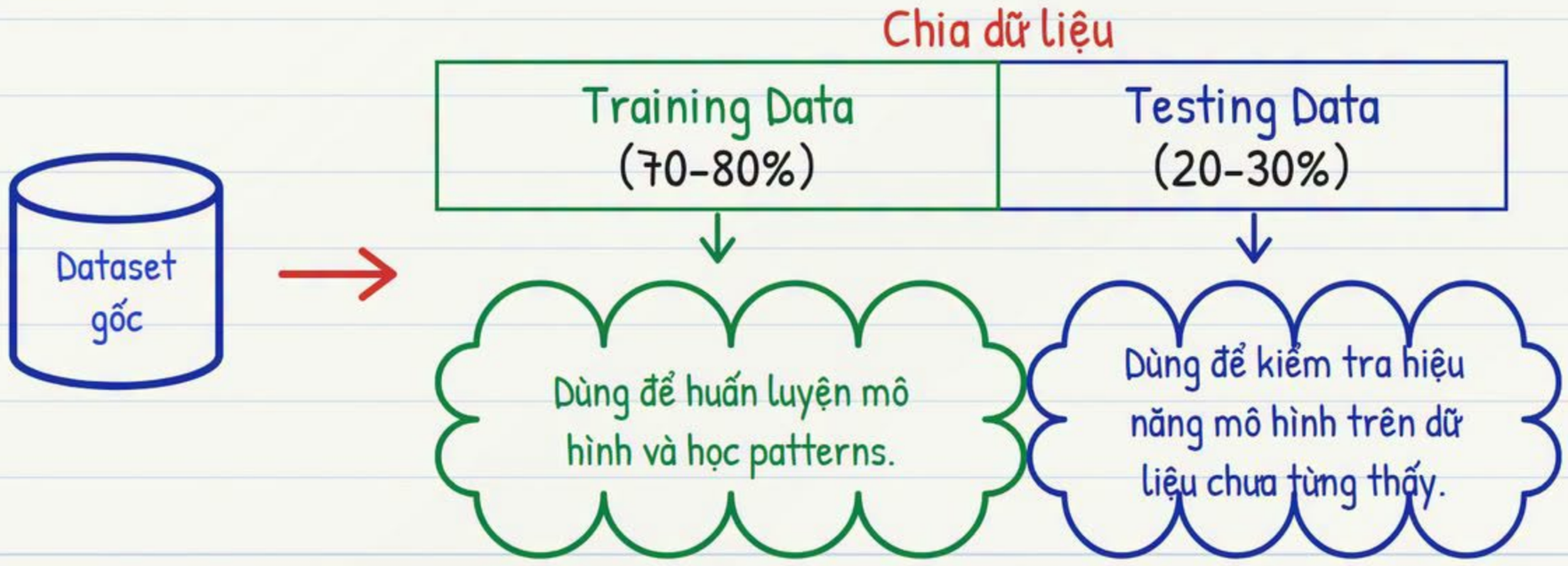
1. Training Data

Training data là phần dữ liệu được dùng để huấn luyện mô hình machine learning. Mô hình xem dữ liệu này, học các patterns và mối quan hệ, rồi điều chỉnh các parameters bên trong để dự đoán tốt hơn. Mục tiêu là giúp mô hình hiểu được bài toán tốt nhất có thể.

2. Testing Data

Testing data là phần dữ liệu mà mô hình chưa từng nhìn thấy. Nó được dùng để đánh giá xem mô hình đã huấn luyện hoạt động tốt đến đâu trên dữ liệu mới. Nó cho ta hình dung về hiệu năng của mô hình trong tình huống thực tế.

3. Cách hoạt động



4. Ví dụ đơn giản

Giả sử ta có 1000 ảnh mèo và chó.

Dùng cả training data và testing data giúp ta xây dựng một mô hình không chỉ học tốt mà còn hoạt động tốt trên dữ liệu mới.

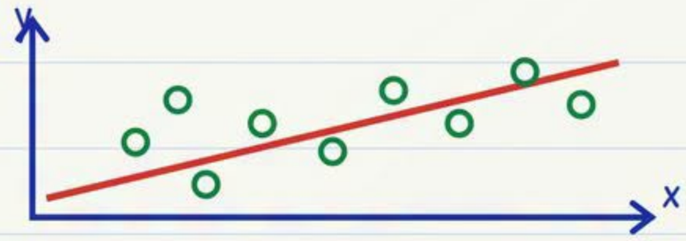
Các thuật toán Machine Learning phổ biến

Có rất nhiều thuật toán machine learning, mỗi loại hữu ích cho một dạng bài toán khác nhau. Dưới đây là một số thuật toán phổ biến nhất:

1. Linear Regression

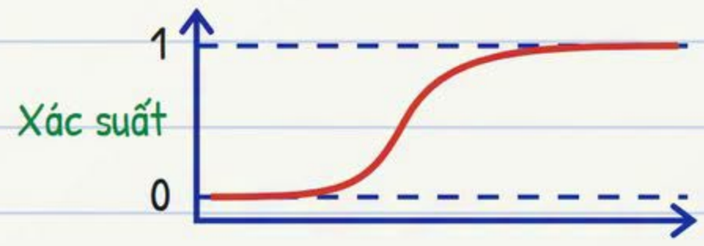
Dùng để dự đoán một giá trị liên tục.

Nó tìm ra đường thẳng khớp nhất với dữ liệu.



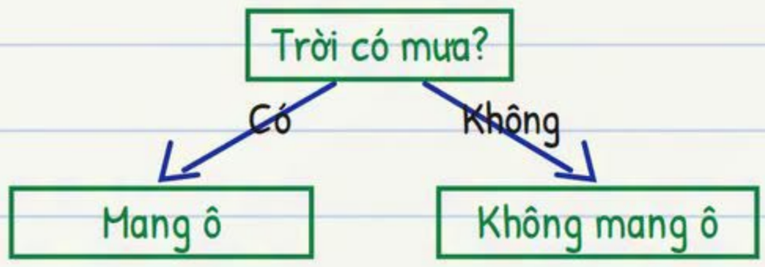
2. Logistic Regression

Dùng cho bài toán classification. Nó dự đoán xác suất một điểm dữ liệu thuộc về một lớp (0 hoặc 1).



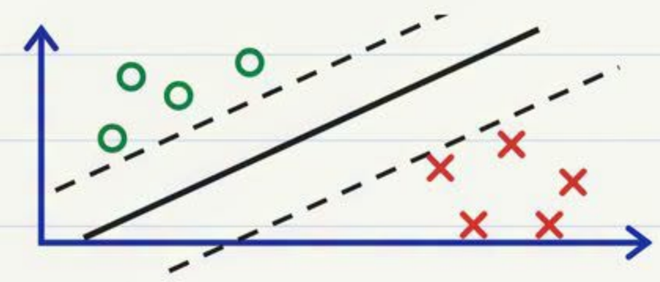
3. Decision Tree

Chia dữ liệu thành các nhánh bằng những câu hỏi dựa trên giá trị của features để đưa ra quyết định.



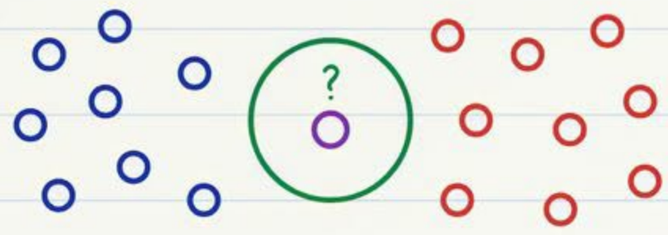
4. Support Vector Machine (SVM)

Tìm ranh giới tốt nhất (đường thẳng hoặc hyperplane) phân tách các lớp với margin lớn nhất.



5. K-Nearest Neighbors (KNN)

Phân loại một điểm dữ liệu dựa trên lớp chiếm đa số trong 'K' điểm lân cận gần nhất.



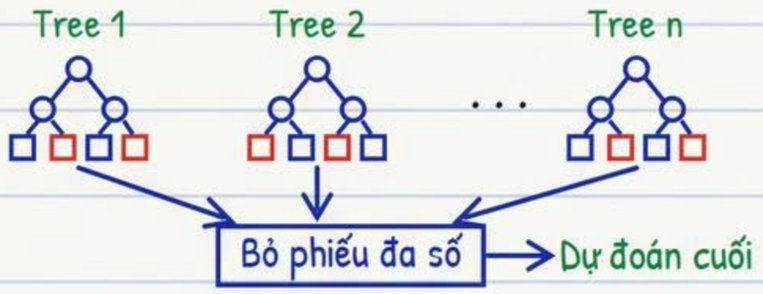
6. Naive Bayes

Dùng xác suất và định lý Bayes với giả định các features độc lập với nhau.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

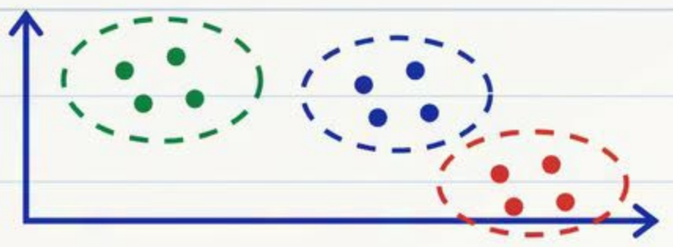
7. Random Forest

Tập hợp (ensemble) của nhiều decision tree. Nó kết hợp các dự đoán để cho kết quả chính xác và ổn định hơn.



8. K-Means Clustering

Gom các điểm dữ liệu tương tự vào K cụm. Nó hoạt động bằng cách tìm tâm cụm rồi gán mỗi điểm vào tâm gần nhất.



Machine Learning: Features và Labels

Trong Machine Learning, dữ liệu thường gồm features và labels. Cả hai đóng vai trò khác nhau nhưng quan trọng như nhau trong việc huấn luyện một mô hình.

1. Features (Input)

Features là các biến đầu vào hay đặc điểm của dữ liệu.
Chúng mô tả từng ví dụ và giúp mô hình học ra các patterns.
Hãy xem features như thông tin mà ta đưa vào cho mô hình.

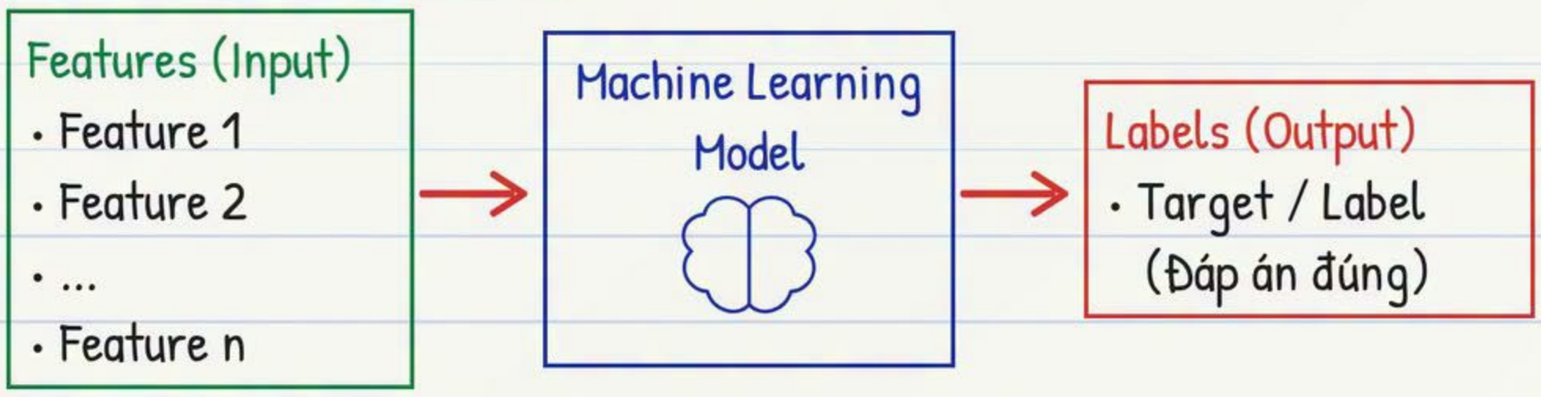
Key Points
- Còn gọi là input variables
- Được mô hình dùng để đưa ra dự đoán
- Có thể là số, văn bản, danh mục, v.v.

2. Labels (Output / Target)

Labels là đáp án đúng hay giá trị mục tiêu gắn với mỗi ví dụ. Mô hình học từ những label này trong quá trình huấn luyện.
Hãy xem labels như kết quả mà ta muốn mô hình dự đoán.

Key Points
- Còn gọi là output hoặc target
- Đại diện cho đáp án đúng
- Dùng để đánh giá và định hướng việc học

3. Cách chúng phối hợp



Mô hình dùng features để dự đoán label. Khi huấn luyện, nó so sánh dự đoán với label thực tế và tự cải thiện.

4. Ví dụ đơn giản

Giả sử ta muốn dự đoán một học sinh sẽ Đậu hay Trượt.

Học sinh	Giờ học (Feature 1)	Chuyên cần % (Feature 2)	Điểm trước đó (Feature 3)	Kết quả (Label)
A	5	80	70	Đậu
B	2	60	40	Trượt
C	6	90	85	Đậu
...	...	...	...	...

Ở đây, Giờ học, Chuyên cần %, Điểm trước đó là features (inputs).
Đậu / Trượt là label (output).

Mô hình học từ những ví dụ này (features + labels), rồi dùng features để dự đoán label cho dữ liệu mới.

Đánh giá mô hình

Đánh giá mô hình là quá trình kiểm tra xem một mô hình machine learning hoạt động tốt đến đâu trên dữ liệu chưa từng thấy. Nó giúp ta hiểu được mô hình đang tốt, tệ, hay cần cải thiện.

1. Vì sao đánh giá mô hình lại quan trọng?

- Cho ta biết mô hình chính xác và đáng tin cậy đến mức nào.
- Giúp so sánh các mô hình khác nhau.
- Cho thấy mô hình có bị overfitting hay underfitting không.

2. Các loại đánh giá

- Training Evaluation : Kiểm tra trên chính dữ liệu đã dùng để huấn luyện mô hình.
- Testing Evaluation : Kiểm tra trên dữ liệu mới, chưa từng thấy để biết mô hình sẽ hoạt động thế nào trong thực tế.

3. Đánh giá cho bài toán Classification

→ Confusion Matrix

Bảng thể hiện số lượng dự đoán đúng và sai.

Giá trị dự đoán \ Giá trị thực tế	Positive	Negative
Positive	TP (True Positive)	FP (False Positive)
Negative	FN (False Negative)	TN (True Negative)

TP : Dự đoán đúng là Positive

TN : Dự đoán đúng là Negative

FP : Dự đoán sai thành Positive

FN : Dự đoán sai thành Negative

→ Các chỉ số phổ biến

- **Accuracy** : (TP + TN) / (TP + TN + FP + FN)
  Tỉ lệ dự đoán đúng trên tổng số dự đoán.
- **Precision** : TP / (TP + FP)
  Trong tất cả dự đoán là positive, có bao nhiêu thực sự là positive.
- **Recall (Sensitivity)** : TP / (TP + FN)
  Trong tất cả positive thực tế, mô hình nhận ra đúng được bao nhiêu.
- **F1 Score** : 2 × (Precision × Recall) / (Precision + Recall)
  Sự cân bằng giữa Precision và Recall.

Ví dụ (Confusion Matrix)

Giả sử mô hình dự đoán email là Spam hay Not Spam.

		Thực tế: Spam	Thực tế: Not Spam
Dự đoán	Spam	90	10
	Not Spam	15	85

Accuracy = (90 + 85) / (90 + 10 + 15 + 85)
= 175 / 200 = 87.5%

Precision = 90 / (90 + 10) = 90%

Recall = 90 / (90 + 15) = 85.7%

F1 Score ≈ 87.7%

4. Đánh giá cho bài toán Regression

Ở đây ta đo sai số giữa giá trị dự đoán và giá trị thực tế.

- **MAE (Mean Absolute Error)** : Trung bình các sai số tuyệt đối.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- **MSE (Mean Squared Error)** : Trung bình bình phương sai số.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **RMSE (Root Mean Squared Error)** : Căn bậc hai của MSE.

$$RMSE = \sqrt{MSE}$$

- **R^2 Score (R-squared)** : Cho biết mô hình giải thích được bao nhiêu phần biến thiên của dữ liệu. Càng cao càng tốt (tối đa = 1).

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \right)$$

Ví dụ Regression

Đường dự đoán càng gần các điểm thực tế thì mô hình càng tốt.

Các thư viện Python phổ biến cho Machine Learning

Python có rất nhiều thư viện mạnh mẽ giúp việc làm machine learning trở nên dễ dàng và nhanh hơn. Dưới đây là một số thư viện phổ biến nhất:

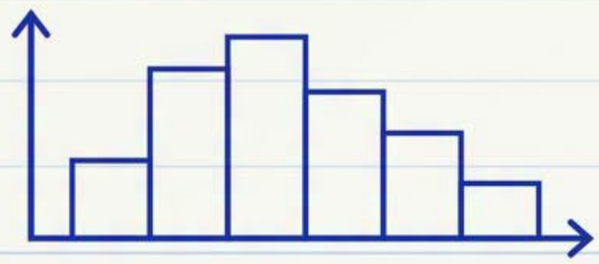
1. NumPy : Dùng cho tính toán số học. Nó hỗ trợ các mảng lớn, ma trận và các hàm toán học.
   Ví dụ : Tạo và thao tác với mảng.

2. Pandas : Dùng để xử lý và phân tích dữ liệu. Nó cung cấp cấu trúc DataFrame để làm việc với dữ liệu có cấu trúc một cách dễ dàng.
   Ví dụ : Đọc dữ liệu từ file CSV và phân tích.

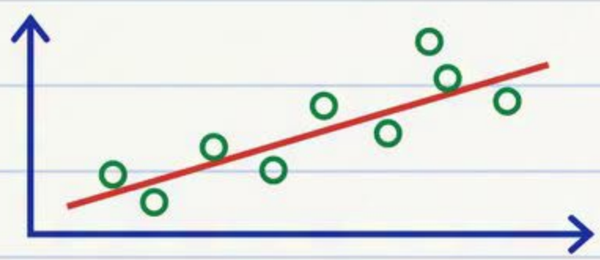
ID	Age	Salary
1	25	50000
2	30	60000
3	28	55000

3. Matplotlib : Dùng để trực quan hoá dữ liệu. Nó giúp tạo ra các đồ thị và biểu đồ.
   Ví dụ : Vẽ một đồ thị đường.

4. Seaborn : Được xây dựng trên Matplotlib. Nó cho các biểu đồ thống kê đẹp và giàu thông tin.
   Ví dụ : Tạo một histogram.

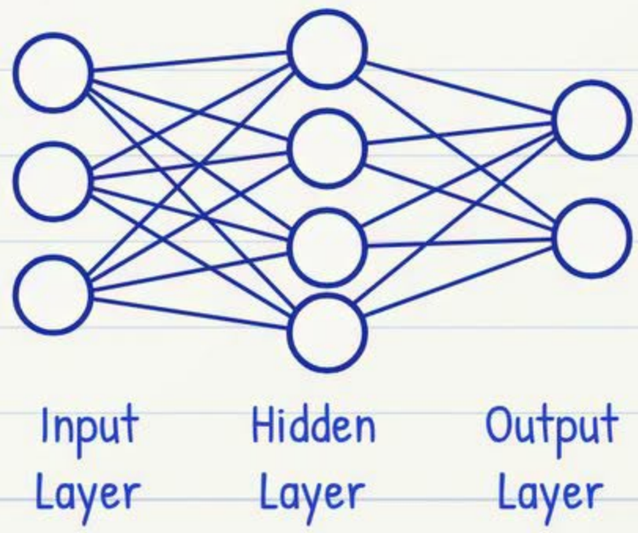


5. Scikit-learn (sklearn) : Thư viện phổ biến nhất cho machine learning. Nó có các công cụ đơn giản, hiệu quả cho classification, regression, clustering, v.v.
   Ví dụ : Huấn luyện mô hình bằng Linear Regression.



6. TensorFlow : Thư viện mã nguồn mở của Google cho deep learning và machine learning. Rất tốt để xây dựng và huấn luyện các mạng neural sâu.

Ví dụ : Xây dựng mô hình neural network.



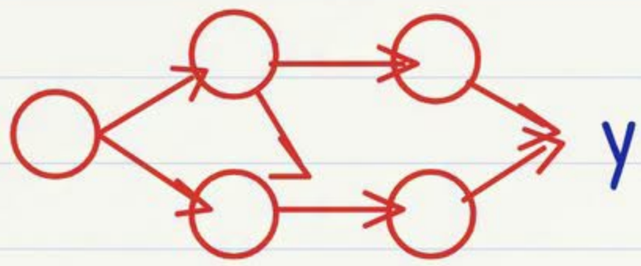
7. Keras : Một API cấp cao chạy bên trên TensorFlow. Nó giúp việc xây dựng các mô hình deep learning dễ và nhanh hơn.

Ví dụ : Tạo một mô hình Sequential.

```
model = Sequential([
  Dense(64, activation='relu'),
  Dense(1, activation='sigmoid')
])
```

8. PyTorch : Thư viện deep learning của Facebook (Meta). Nó mang lại sự linh hoạt và đồ thị tính toán động.

Ví dụ : Xây dựng và huấn luyện các mô hình deep learning.



9. XGBoost : Thư viện được tối ưu cho các thuật toán gradient boosting. Nó nhanh, hiệu quả và hoạt động tốt với dữ liệu có cấu trúc.

Ví dụ : Huấn luyện mô hình XGBoost cho classification.