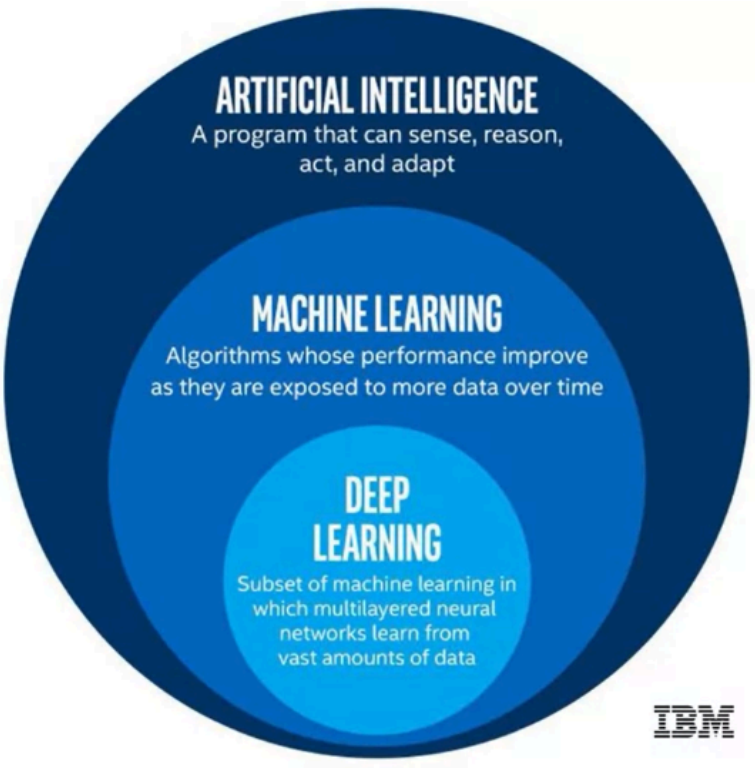
AIL303m_Course1

1. A Brief History of Modern AI and its Applications

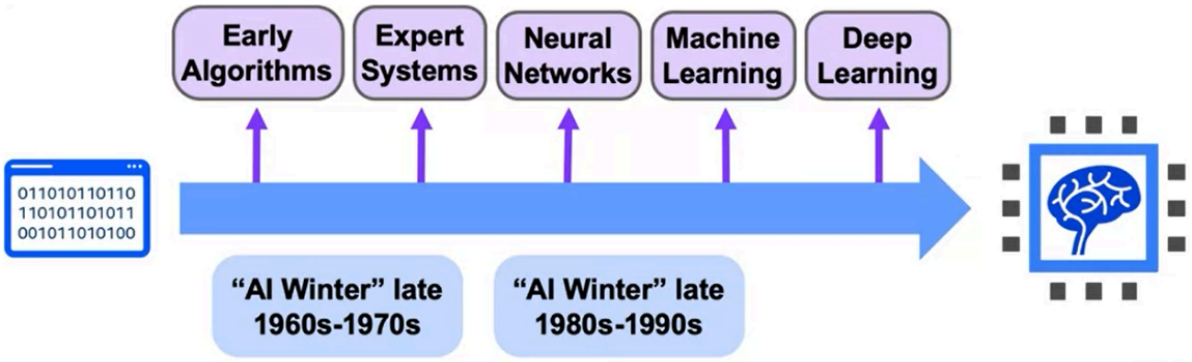
1.1. Introduction:

- AI: Cơ bản là một ngành khoa học máy tính cố gắng mô phỏng hành vi thông minh trong máy tính. Kiểu như máy học hỏi và giải quyết vấn đề như con người.
- ML: Là xây dựng các chương trình không được lập trình rõ ràng, mà chúng tự học các patterns khi tiếp xúc với dữ liệu nhiều hơn.
- DL: Là một tập con của ML, sử dụng các mô hình rất phức tạp gọi là mạng nơ ron sâu.
- DL vs ML: DL giúp mô hình tự xác định cách biểu diễn dữ liệu tốt nhất, trong khi ML cổ điển thì con người phải làm việc này.



1.2. History of AI:

- AI đã trải qua nhiều chu kỳ nổi dậy cũng như ngủ đông (AI winter) do kỳ vọng và thất vọng.
- 1950s: Alan Turing phát triển Turing test (1950). AI được chấp nhận tại Hội nghị Dartmouth (1956). Perceptron (tiền thân mạng nơ ron) ra đời (1957).
- Mùa đông AI lần 1: Sau các báo cáo tiêu cực về lợi suất (ALPAC 1966, Lighthill 1973), dẫn đến cắt giảm tài trợ.
- 1980s (AI Bùng nổ): Sự trỗi dậy của Hệ thống Chuyên gia (Expert Systems), hệ thống có quy tắc lập trình mô phỏng chuyên gia. Thuật toán Backpropagation (1986) cho phép huấn luyện perceptron đa lớp.
- Mùa đông AI lần 2: Hệ thống chuyên gia chậm tiến, mạng nơ-ron không mở rộng được cho các vấn đề lớn.
- Hiện đại (Sau 2012): Đột phá DL (sau bài báo của Hinton 2006). Được thúc đẩy bởi: dữ liệu lớn hơn, máy tính nhanh hơn, gói phần mềm mã nguồn mở. Ứng dụng rộng rãi: xe tự hành, phát hiện gian lận, giao dịch thuật toán, y tế.



2. Quy trình ML và thuật ngữ

2.1. Quy trình Học máy (6 bước):

1. Problem Statement: Xác định vấn đề.

2. Data Collection: Thu thập dữ liệu.

3. Data Exploration & Preprocessing: Khám phá và tiền xử lý dữ liệu (rất quan trọng để làm sạch).

4. Modeling: Xây dựng mô hình.

5. Validation: Xác thực.

6. Decision Making & Deployment: Đưa ra quyết định và triển khai.

2.2. Thuật ngữ ML:

- Target (Mục tiêu): Cái mình đang cố dự đoán (giá trị hoặc category).
- Features (Tính năng): Các thuộc tính của dữ liệu dùng để dự đoán (biến giải thích).
- Example / Observation (Quan sát): Một điểm dữ liệu duy nhất (một hàng).
- Label (Nhãn): Giá trị của target cho một điểm dữ liệu cụ thể.

3. Retrieving and Cleaning Data

3.1. Retrieving Data:

Nguồn dữ liệu đa dạng:

- SQL Databases: Cơ sở dữ liệu quan hệ, schema cố định.
- NoSQL Databases: Không quan hệ, cấu trúc đa dạng, thường lưu trữ dữ liệu dưới định dạng JSON.
- File phẳng: Phổ biến nhất là CSV (phân tách bằng dấu phẩy). JSON (giống Python dictionary) cũng thường dùng.
- APIs và Cloud data sources.

3.2. Cleaning Data (quan trọng):

- Lý do: Dữ liệu lộn xộn dẫn đến hiệu ứng "garbage-in, garbage-out" và kết quả không đáng tin cậy.
- Các vấn đề cần xử lý:

  1. Dữ liệu trùng lặp (Duplicate Data): Rất có hại! Nó gây ra trọng lượng không cân xứng khi huấn luyện và làm sai lệch việc chia tập train/test, dẫn đến ước tính hiệu suất bị thiên vị.

  → *Cách fix:* Dễ nhất là dùng `drop_duplicates` của Pandas.

2. Dữ liệu không cần thiết (Unnecessary Data): Các thứ cần bỏ là PII (thông tin nhận dạng cá nhân), URL, tracking codes, khoảng trắng thừa.

→ *Cách fix:* Dùng `filter` hoặc `drop` cột/hàng.

3. Dữ liệu thiếu (Missing Data): có 3 cách:

- Remove: Xóa hàng/cột chứa dữ liệu thiếu.
- Impute: Thay thế bằng giá trị thay thế (ví dụ: trung bình, phổ biến nhất).
- Mask: Tạo một category riêng cho các giá trị thiếu.

4. Giá trị ngoại lai (Outliers): Quan sát cách xa các quan sát khác, có thể tác động lớn đến mô hình.

- *Cách tìm:* Thông qua Plots, Statistics, hoặc Residuals.
- *Cách xử lý:* Xóa chúng, gán mean/median, biến đổi biến, hoặc dùng mô hình chịu lỗi (resistant).

4. EDA and Feature Engineering

4.1. Phân tích khám phá dữ liệu (EDA):

- Mục đích: Tóm tắt các đặc điểm chính của tập dữ liệu, thường dùng trực quan hóa. Giúp mình có cảm nhận ban đầu và tìm ra mẫu/xu hướng.
- Kỹ thuật: Tính Summary Statistics (mean, median, correlation) và trực quan hóa (Histograms, Scatter Plots, Box Plots).
- Lấy mẫu (Sampling): Hữu ích khi dữ liệu quá lớn, hoặc khi cần xử lý các kết quả không đồng đều.

4.2. Feature Engineering:

- Mục tiêu: Chuyển đổi dữ liệu để phù hợp với giả định của mô hình (ví dụ: tạo mối quan hệ tuyến tính).

  1. Chuyển đổi phân phối:

  - Log Transformation: Dùng khi dữ liệu bị lệch (skewed).
  - Polynomial Features: Thêm các tính năng bậc cao hơn (vd: bậc 2) để mô hình 'tuyến tính' có thể ước tính mối quan hệ phi tuyến tính.

2. Mã hóa tính năng (Feature Encoding): Chuyển tính năng phân loại (Categorical) thành số:

- Không thứ tự (*Nominal)*: Dùng One-hot encoding (tạo biến nhị phân 0/1 cho mỗi category) hoặc Binary encoding (cho biến chỉ có 2 giá trị).
- Có thứ tự (*Ordinal):* Dùng Ordinal encoding (gán số nguyên có thứ tự 0, 1, 2...).

3. Điều chỉnh tỷ lệ tính năng (Feature Scaling): Giúp các biến số liên tục có thể so sánh được.

- Standard Scaling: Chuẩn hóa biến về phân phối chuẩn.
- Min-max Scaling: Chuyển biến về khoảng (0, 1), nhưng nhạy cảm với outliers.
- Robust Scaling: Ít nhạy cảm với outliers hơn (dùng khoảng tứ phân vị).

5. Inferential Statistics and Hypothesis Testing

5.1. Ước tính (Estimation) và Suy luận (Inference):

- Estimation: Áp dụng thuật toán để xác định tham số quần thể (ví dụ: tính trung bình).
- Inference: Đưa ra độ chính xác cho ước tính đó (ví dụ: lỗi chuẩn).

5.2. Mô hình Thống kê:

- Parametric: Giả định dữ liệu có một tập hợp các phân phối với số lượng tham số hữu hạn (ví dụ: Phân phối Chuẩn). Ước tính thường dùng Maximum Likelihood Estimation (MLE).
- Non-parametric: Đưa ra ít giả định hơn, không cần giả định dữ liệu thuộc về phân phối cụ thể nào (distribution-free).

5.3. Frequentist vs Bayesian:

- Frequentist: Tập trung vào các quan sát lặp lại. Áp dụng xác suất trực tiếp vào dữ liệu quan sát được.

- Bayesian: Mô tả tham số bằng phân phối xác suất. Bắt đầu với phân phối prior (dựa trên niềm tin) và cập nhật nó sau khi xem dữ liệu để tạo ra phân phối posterior.

5.4. Kiểm định Giả thuyết (Hypothesis Testing):

- Giả thuyết: Một tuyên bố về tham số quần thể.
  - H0 (Null Hypothesis): Giả thuyết mặc định (ví dụ: không có bệnh).
  - H1 (Alternative Hypothesis): Giả thuyết đối lập (ví dụ: có bệnh).
- Lỗi Loại I & Loại II:
  - Lỗi Loại I: Bác bỏ H0 khi H0 là đúng (không có bệnh nhưng nói có bệnh).
  - Lỗi Loại II: Không bác bỏ H0 khi H0 là sai (có bệnh nhưng nói không có bệnh).
- Ngưỡng xác suất (Significance Level): Phải chọn trước khi tính toán (thường là 0.01 hoặc 0.05).
- P-value: Mức ý nghĩa nhỏ nhất mà tại đó H0 sẽ bị bác bỏ.

Priors: $P(H_1) = 1/2 = P(H_2) = 1/2$

Updating priors after seeing the data 3 heads (Bayes' Rule):

$$P(H_1|x) = \frac{P(x|H_1)P(H_1)}{P(x)}$$

We can write out the ratio:

$$\frac{P(H_1|x)}{P(H_2|x)} = \frac{P(H_1)P(x|H_1)}{P(H_2)P(x|H_2)}$$

- “The priors are multiplied by the likelihood ratio, which does not depend on the priors.”
  - LR chỉ phụ thuộc vào dữ liệu và giả thuyết, **không phụ thuộc vào prior**.

    - Ta dùng LR như “hệ số điều chỉnh” để cập nhật prior thành posterior.
    - Nghĩa là dữ liệu tác động đến niềm tin ban đầu thông qua LR, không phải thông qua chính bản thân prior.
- “The likelihood ratio tells us how we should update the priors in reaction to seeing a given set of data”
    - LR cho ta biết dữ liệu này ủng hộ giả thuyết nào mạnh hơn.
        - Nếu LR > 1: dữ liệu ủng hộ H1 hơn H0.
        - Nếu LR < 1: dữ liệu ủng hộ H0 hơn H1.
    - Mức độ điều chỉnh niềm tin (từ prior → posterior) phụ thuộc trực tiếp vào LR.

5.5. Tương quan và Nhân quả (Correlation vs Causation):

- Correlation: Nếu X và Y tương quan, thì X hữu ích để dự đoán Y.
- Lưu ý: Tương quan không ngụ ý nhân quả.
- Biến Nhiễu (Confounding Variables): Là yếu tố thứ ba gây ra cả X và Y thay đổi, khiến chúng tương quan dù không có quan hệ trực tiếp.
- Tương quan Ngẫu nhiên (Spurious Correlations): Chỉ là sự trùng hợp do mẫu dữ liệu cụ thể, có thể không đúng trên các mẫu khác.