# Module1:

**Khái niệm về Machine Learning:**

**Machine Learning** là một nhánh của **Trí tuệ nhân tạo** cho phép các thuật toán học các mẫu từ dữ liệu mà không cần lập trình rõ ràng. **Hiệu suất** của các thuật toán Machine Learning **cải thiện theo thời gian khi được tiếp xúc với nhiều dữ liệu hơn**, nhưng có thể **đạt đến điểm bão hòa sau một lượng dữ liệu nhất định**, cho thấy hiệu quả giảm dần.

**Các khái niệm thường gặp:**

**Features** (đặc trưng) là các **biến đầu vào** được **sử dụng để dự đoán**, trong khi **target** (mục tiêu) là **kết quả** mà chúng ta muốn dự đoán.

**Các loại Machine Learning:**

1. **Supervised Learning** - Học có giám sát

Supervised Learning học từ dữ liệu có nhãn để dự đoán kết quả.

**Đặc điểm chính:**

**Dữ liệu có nhãn**: Dữ liệu huấn luyện có nhãn được định trước

**Thuật toán:** Linear Regression, Logistic Regression, SVM, Decision Trees, Neural Networks

**Ứng dụng thực tế:**

- Phát hiện spam email
- Chẩn đoán y tế
- Phát hiện gian lận
- Nhận dạng khuôn mặt

2. **Unsupervised Learning** - Học không giám sát

Unsupervised Learning làm việc với dữ liệu không có nhãn để tìm ra các cấu trúc ẩn

**Đặc điểm chính**:

**Dữ liệu không nhãn**: Không có đầu ra được định trước

**Thuật toán**: K-Means, Hierarchical Clustering, PCA, Autoencoders

**Ứng dụng thực tế:**

- Phân khúc khách hàng
- Phát hiện bất thường
- Giảm chiều dữ liệu
- Tối ưu hóa mạng xã hội

3. **Reinforcement Learning** - Học tăng cường

Reinforcement Learning học thông qua phần thưởng và phạt để tối đa hóa thành công

**Đặc điểm chính:**

**Học dựa trên tương tác:** học bằng cách thực hiện hành động và nhận phản hồi

**Không có dữ liệu nhãn:** Học từ thử và sai

**Thuật toán:** Q-learning, SARSA, Deep Q-Networks (DQN)

**Ứng dụng thực tế:**

- Xe tự lái: Học các chính sách lái xe để điều hướng, chuyển làn và tránh chướng ngại
- Giao dịch thuật toán: Tối ưu hóa chiến lược mua bán để tối đa hóa lợi nhuận
- Robot công nghiệp: Học cách điều hướng môi trường và nắm bắt các vật thể

4. **Semi-Supervised Learning** - Học bán giám sát

Semi-supervised Learning kết hợp cả dữ liệu có nhãn và không nhãn để huấn luyện mô hình.

**Đặc điểm chính:**

**Dữ liệu hỗn hợp:** Sử dụng một lượng nhỏ dữ liệu có nhãn kết hợp với một lượng lớn dữ liệu không nhãn

**Kỹ thuật**: Self-training, Co-training, Graph-based labeling

**Quy trình:** Huấn luyện mô hình ban đầu trên dữ liệu có nhãn, sau đó áp dụng cho dữ liệu không nhãn với pseudo-labeling

**Ứng dụng thực tế:**

- Phân loại trang web
- Nhận dạng giọng nói
- Phát hiện gian lận

**Deep Learning so với Traditional Machine Learning**

**Traditional Machine Learning**

Điểm mạnh:

- Hoạt động tốt với dataset nhỏ, sạch
- Có thể diễn giải và minh bạch
- Mô hình đơn giản hơn, huấn luyện nhanh hơn
- Chạy trên phần cứng tiêu chuẩn

Hạn chế:

- Feature engineering thủ công tốn thời gian
- Hiệu suất hạn chế, ngay cả với nhiều dữ liệu hơn
- Tập trung hẹp, ít linh hoạt với các thay đổi

**Deep Learning**

Điểm mạnh:

- Học features tự động: Deep Learning tự động trích xuất các đặc trưng hữu ích từ dữ liệu thô
- Tăng trưởng liên tục với nhiều dữ liệu hơn
- Kết quả tối tân trên các tác vụ nhận thức
- Features có thể tái sử dụng trên các tác vụ khác
- Khả năng chịu đựng cao với dữ liệu nhiễu

Hạn chế:

- Cần nhiều dữ liệu
- Thiếu khả năng diễn giải
- Tốn kém về mặt tính toán
- Cấu trúc cứng nhắc, khó thay đổi

**Khi nào sử dụng phương pháp nào?**

Sử dụng Traditional Machine Learning khi:

-Dữ liệu có cấu trúc và dạng bảng
-Dataset nhỏ
-Cần khả năng diễn giải
-Tài nguyên tính toán hạn chế

Sử dụng Deep Learning khi:

-Dữ liệu không có cấu trúc như hình ảnh, âm thanh, văn bản
-Dataset lớn
-Cần hiệu suất cao trên các tác vụ phức tạp
-Có tài nguyên tính toán mạnh

**Feature Engineering** - Kỹ thuật đặc trưng

Feature Engineering là quá trình **biến đổi dữ liệu thô thành các đặc trưng hữu ích** giúp **cải thiện hiệu suất** của các mô hình machine learning. Đây là một bước **quan trọng trong traditional machine learning**, trong khi **deep learning có thể tự động thực hiện** quá trình này.

Các kỹ thuật Feature Engineering

1. Coordinate Transformation - Biến đổi tọa độ:

Ví dụ chuyển đổi từ tọa độ Cartesian (x, y) sang tọa độ cực (r, θ) để giải quyết bài toán phân loại theo khoảng cách.

2. Feature Selection - Lựa chọn đặc trưng:

**Chọn ra** một số lượng nhỏ hơn **các feature phù hợp** với bài toán **từ một tập lớn các feature ban đầu**.

3. Categorical Encoding - Mã hóa phân loại:

**Chuyển đổi** các **biến phân loại thành dạng số**, như one-hot encoding.

4. Normalization - Chuẩn hóa:

Đưa các feature **về cùng một phạm vi giá trị** để **tránh bias do scale khác nhau**.

**Tìm hiểu về Features trong hình ảnh**

Traditional Machine Learning **gặp nhiều khó khăn khi xử lý hình ảnh** do **đặc tính phức tạp của dữ liệu hình ảnh**. Trong **traditional computer vision**, các kỹ sư **phải thiết kế thủ công** các **đặc trưng** (hand-crafted features) trước khi có thể huấn luyện mô hình.

**Mỗi pixel như một feature**

Khi coi mỗi pixel như một feature riêng biệt, các thách thức chính bao gồm:

-**Số lượng features khổng lồ**: Một hình ảnh 256x256 pixel có 65,536 features (pixels), làm cho quá trình phân tích trở nên cực kỳ phức tạp

**-Thiếu thông tin không gian**: Khi flatten hình ảnh thành vector 1D, mất đi mối quan hệ không gian giữa các pixel

-**Không có ý nghĩa ngữ cảnh**: Mỗi pixel riêng lẻ không mang nhiều thông tin ý nghĩa về đối tượng trong hình ảnh

Traditional Feature Extraction Methods

Để giải quyết vấn đề này, traditional computer vision phát triển các phương pháp trích xuất đặc trưng chuyên biệt:

SIFT (Scale-Invariant Feature Transform)

Cách hoạt động: **Phát hiện các keypoints đặc biệt** trong hình ảnh và **tạo descriptors không thay đổi theo scale và rotation**

Ưu điểm: Bất biến với scale, rotation, và thay đổi illumination

Ứng dụng: Panorama stitching, object tracking, robot navigation

SURF (Speeded-Up Robust Features)

Đặc điểm: Phiên bản tối ưu tốc độ của SIFT, phù hợp cho real-time keypoint matching

Lợi ích: Nhanh hơn SIFT nhưng vẫn duy trì độ chính xác cao

HOG (Histogram of Oriented Gradients)

Nguyên lý: **Chia hình ảnh thành các cell nhỏ** và **tính histogram của gradient directions** trong mỗi cell

Công dụng: Đặc biệt hiệu quả cho object detection, nhất là pedestrian detection

Quy trình:

Tính gradients → Chia thành cells → Tạo histograms → Normalization

LBP (Local Binary Patterns)

Phương pháp: **So sánh mỗi pixel với các pixel lân cận** để **tạo binary pattern**

Ưu thế: **Đơn giản tính toán** nhưng **mạnh trong texture classification**

Hạn chế của Traditional Methods

Thiết kế thủ công phức tạp: **Kỹ sư phải có domain knowledge để chọn features phù hợp** cho từng bài toán cụ thể.

Không tự thích ứng: Khi **số lượng classes tăng lên**, **feature extraction trở nên cồng kềnh** và khó khăn hơn.

Hiệu suất hạn chế: Các **handcrafted features có thể không generalize tốt** trên các scenario khác nhau.

**Deep Learning và tự động Feature Extraction**

**Automatic Feature Learning**

Deep Learning, đặc biệt là Convolutional Neural Networks (CNNs), tự động học và trích xuất features từ dữ liệu thô mà không cần sự can thiệp thủ công.

**Cách CNNs hoạt động**

**Hierarchical Feature Learning**: CNNs **học các representations theo từng tầng từ thấp đến cao**:

**Early Layers**: **Phát hiện** các l**ow-level features** như edges, textures, color gradients

**Middle Layers**: **Kết hợp basic features để nhận diện higher-level structures** như shapes, corners

**Deep Layers**: **Học complex objects và semantic concepts** như eyes, wheels, faces

**Spatial Relationships:** CNNs **bảo tồn mối quan hệ không gian giữa các pixels**:

**Local Connectivity: Mỗi neuron chỉ kết nối với một vùng nhỏ của layer trước** (receptive field)

**Shared Weights:** Cùng **một filter được áp dụng trên toàn bộ hình ảnh**, giúp **phát hiện cùng một pattern ở các vị trí khác nhau**

**Translation Equivariance**: CNN có khả năng **phát hiện features bất kể chúng xuất hiện ở đâu trong hình ảnh**

So sánh Deep Learning vs Traditional ML trong Image Processing

| Khía cạnh | Traditional Machine Learning | Deep Learning |
|---|---|---|
| Feature Extraction | Thủ công, cần domain expertise | Tự động, học từ dữ liệu |
| Spatial Relationships | Bị mất khi flatten thành vector | Được bảo tồn qua convolution |
| Scalability | Khó khăn khi tăng số classes | Tự thích ứng với dataset lớn |
| Performance | Tốt với dataset nhỏ, structured | Vượt trội với dataset lớn, complex |
| Computational Requirements | Thấp hơn | Cần GPU mạnh |
| Development Time | Nhiều thời gian cho feature engineering | Lâu training nhưng ít manual work |

**Các cột mốc lịch sử của AI**
**Thuật ngữ "Artificial Intelligence" và Turing Test**

"Artificial Intelligence" (Trí tuệ nhân tạo) được đặt ra vào năm 1956 tại Hội nghị Dartmouth, đánh dấu sự ra đời của lĩnh vực AI như một ngành nghiên cứu độc lập.

Turing Test được Alan Turing giới thiệu năm 1950 trong bài báo "Computing Machinery and Intelligence". Đây là một tiêu chuẩn nền tảng để đánh giá khả năng "suy nghĩ" của máy móc: nếu một máy có thể giao tiếp bằng ngôn ngữ tự nhiên mà người đánh giá không phân biệt được với con người, máy đó được coi là có trí tuệ nhân tạo.

**Chu kỳ đầu tư và các "AI Winter"**
Thập niên 1960-1970: AI trải qua "AI Winter" đầu tiên do kỳ vọng quá cao và tiến bộ hạn chế, đặc biệt trong lĩnh vực dịch máy.
Thập niên 1980: AI hồi sinh nhờ hệ chuyên gia (expert systems) và thuật toán Backpropagation cho mạng nơ-ron. Tuy nhiên, cuối thập niên 80 và 90 lại xuất hiện "AI Winter" thứ hai do các hạn chế thực tế về hiệu suất và chi phí.

**Phát triển gần đây**
Cuối thập niên 90 và đầu 2000: Machine Learning bắt đầu thành công trong các ứng dụng thực tế như nhận diện giọng nói.
Hiện nay: Deep Learning nổi lên như công cụ mạnh mẽ, vượt qua các giới hạn trước đây và đạt thành tích xuất sắc trong các bài toán phức tạp như phân loại hình ảnh, dịch máy, nhận diện giọng nói.
Kiến thức nền tảng và công cụ
Thư viện Python phổ biến: NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, TensorFlow, Keras.
Kiến thức nền tảng: Thống kê cơ bản và đại số tuyến tính rất quan trọng để hiểu và tránh diễn giải sai các mô hình.

**Quy trình Machine Learning điển hình**

**Xác định vấn đề:** Định nghĩa rõ bài toán cần giải quyết (ví dụ: phân loại hình ảnh).
**Thu thập dữ liệu:** Tìm kiếm và thu thập đủ dữ liệu có nhãn để huấn luyện, chú ý đến các điều kiện khác nhau.
**Khám phá và tiền xử lý dữ liệu:** Làm sạch, chuẩn hóa, phân tích phân phối dữ liệu, chuyển đổi định dạng dữ liệu phù hợp cho mô hình.

**Từ vựng cơ bản trong Machine Learning**
**Target Variable** (Biến mục tiêu): Giá trị cần dự đoán (ví dụ: loài hoa iris).
**Features** (Đặc trưng): Các biến giải thích dùng để dự đoán biến mục tiêu (ví dụ: chiều dài và chiều rộng đài hoa, cánh hoa).
**Examples/Observations** (Ví dụ/Quan sát): Mỗi dòng dữ liệu, đại diện cho một điểm dữ liệu riêng biệt.
**Label** (Nhãn): Giá trị cụ thể của biến mục tiêu cho một quan sát (ví dụ: tên loài hoa).

# Module3

**EDA (Phân tích dữ liệu khám phá)**

1. **EDA là bước đầu tiên trong phân tích dữ liệu**, giúp **tóm tắt các đặc điểm chính của tập dữ liệu** bằng các phương pháp thống kê và trực quan hóa.

> Mục tiêu là **nhận diện mẫu**, **xu hướng**, **phát hiện dữ liệu bất thường** và **xác định nhu cầu làm sạch dữ liệu.**
>
> **Các thống kê thường dùng**: trung bình, trung vị, min, max, tương quan giữa các cột.
> **Các biểu đồ phổ biến:** histogram (phân phối), scatter plot (mối quan hệ giữa hai biến), box plot (phát hiện outlier).
> Lấy mẫu ngẫu nhiên giúp quản lý tập dữ liệu lớn và đảm bảo mẫu đại diện; lấy mẫu phân tầng giữ tỷ lệ các nhóm trong dữ liệu mất cân bằng.

**2. Thư viện trực quan hóa dữ liệu**

Matplotlib: Thư viện nền tảng, linh hoạt, tạo nhiều loại biểu đồ.

Pandas: Có hàm plot wrapper cho Matplotlib, dễ dùng nhưng ít tùy biến hơn.

Seaborn: Xây dựng trên Matplotlib, giúp tạo biểu đồ đẹp và có ý nghĩa thống kê, như pair plot (quan hệ nhiều biến), hexbin plot (mật độ điểm dữ liệu), facet grid (so sánh phân phối giữa các nhóm).

**3. Kỹ thuật biến đổi và xử lý đặc trưng**

**Feature Engineering:** Biến đổi dữ liệu thô thành đặc trưng hữu ích để tối ưu hóa hiệu suất mô hình.
Biến đổi biến số: Log transformation giúp chuẩn hóa dữ liệu lệch, Box-Cox cũng dùng để làm dữ liệu gần phân phối chuẩn hơn.

**Polynomial Features:** Thêm các đặc trưng bậc cao ($x^2$, $x^3$) để mô hình hóa quan hệ phi tuyến.

**4. Mã hóa đặc trưng** (Feature Encoding)
**Mã hóa nhị phân:** Dùng cho dữ liệu phân loại hai giá trị (True/False).
**One-hot encoding**: Tạo cột nhị phân cho từng giá trị phân loại, phù hợp với dữ liệu không có thứ tự.
**Ordinal encoding**: Dùng cho dữ liệu phân loại có thứ tự (ví dụ: thấp, vừa, cao), chuyển thành số phản ánh thứ tự thực tế.

**5.Chuẩn hóa đặc trưng** (Feature Scaling)

**Dữ liệu thực tế** thường **có các biến với đơn vị và phạm vi khác nhau**, cần **chuẩn hóa để các thuật toán học máy hoạt động hiệu quả**.

**Standard Scaling**: Trừ trung bình, chia cho độ lệch chuẩn (có thể bị ảnh hưởng bởi outlier).

**Min-Max Scaling**: Đưa giá trị về khoảng 0-1, nhưng nhạy với outlier.

**Robust Scaling**: Dựa vào median và IQR, giảm ảnh hưởng của outlier nhưng không đảm bảo giá trị nằm trong khoảng 0-1.

Chuẩn hóa đặc trưng đặc biệt quan trọng với các thuật toán dựa trên khoảng cách như KNN, hoặc khi các biến có phạm vi rất khác nhau (ví dụ: tuổi tính bằng giây so với số lần phẫu thuật).

# Module4

**Hiểu rõ về Estimation vs Inference**

Ước lượng (Estimation) là quá trình sử dụng dữ liệu mẫu để tính ra giá trị ước lượng cho các tham số của quần thể, chẳng hạn như trung bình hoặc tỷ lệ.
Point Estimation: Đưa ra một giá trị duy nhất (ví dụ: trung bình mẫu = 25)
Interval Estimation: Đưa ra khoảng tin cậy (ví dụ: khoảng tin cậy 95% cho trung bình từ 23-27)

Suy luận (Inference) là quá trình hiểu về phân phối quần thể, bao gồm các tham số như sai số chuẩn (standard error), và sử dụng dữ liệu mẫu để rút ra kết luận về đặc điểm của toàn bộ quần thể.

**Tham số vs Phi tham số**

**Phương pháp Tham số** (Parametric)
Giả định cụ thể về phân phối: Yêu cầu dữ liệu tuân theo một phân phối xác định (thường là phân phối chuẩn)
Số tham số hữu hạn: Được định nghĩa bởi số lượng tham số giới hạn (ví dụ: phân phối chuẩn có mean và variance)
Kích thước mẫu: Thường yêu cầu mẫu lớn (≥30)
Ví dụ: t-test, ANOVA, Linear Regression

**Phương pháp Phi tham số** (Non-parametric)
Không giả định phân phối: Không yêu cầu giả định về hình dạng phân phối của dữ liệu
Linh hoạt hơn: Cho phép linh hoạt khi dữ liệu không tuân theo phân phối chuẩn
Robust to outliers: Ít bị ảnh hưởng bởi outliers
Ví dụ: Mann-Whitney U test, Kruskal-Wallis test, sử dụng histogram để tạo phân phối

**Frequentist vs. Bayesian Statistics**

**Frequentist Statistics**
Long-run frequencies: Tập trung vào tần suất dài hạn của các sự kiện
Objective approach: Dựa hoàn toàn vào dữ liệu quan sát, không sử dụng thông tin tiên nghiệm
Fixed parameters: Coi tham số là giá trị cố định, chưa biết
Confidence intervals: Cung cấp khoảng tin cậy dựa trên repeated sampling
**Bayesian Statistics**
Degree of belief: Xác suất thể hiện mức độ tin tưởng vào một kết quả cụ thể
Prior beliefs: Kết hợp thông tin tiên nghiệm với dữ liệu mới
Posterior distribution: Cập nhật prior beliefs với evidence để tạo posterior
Parameter distributions: Cho phép tham số có phân phối xác suất riêng

**Các phân phối phổ biến và ứng dụng**
**Phân phối Đều** (Uniform)
Đặc điểm: Mọi giá trị trong khoảng có xác suất bằng nhau, như tung xúc xắc công bằng
Hình dạng: Phẳng, không thiên vị về giá trị nào

**Phân phối Chuẩn** (Normal/Gaussian)
Đặc điểm: Giá trị tập trung quanh trung bình, tạo đường cong hình chuông
Central Limit Theorem: Trung bình mẫu sẽ xấp xỉ phân phối chuẩn khi kích thước mẫu tăng

**Phân phối Log-Normal**
Định nghĩa: Nếu logarit của biến có phân phối chuẩn, biến gốc có phân phối log-normal
Ứng dụng: Thường gặp trong dữ liệu tài chính với nhiều outliers lớn

**Phân phối Mũ** (Exponential)
Mục đích: Mô hình hóa thời gian đến sự kiện tiếp theo
Ví dụ: Thời gian đến lượt khách hàng, khoảng cách giữa các sự cố

**Phân phối Poisson**
Đại diện: Số sự kiện xảy ra trong khoảng thời gian cố định
Tham số lambda: Tỷ lệ trung bình của sự kiện

**Kiểm định Giả thuyết** (Hypothesis Testing)

Các thành phần cơ bản
Null Hypothesis ($H_o$): Khẳng định về tham số quần thể với giá trị cụ thể
Alternative Hypothesis ($H_1$): Khẳng định giá trị khác với $H_o$

**Các thuật ngữ quan trọng**
Test Statistic: Giá trị tính từ dữ liệu mẫu để quyết định chấp nhận hay bác bỏ $H_o$
Rejection Region: Tập hợp giá trị test statistic dẫn đến bác bỏ $H_o$
Acceptance Region: Tập hợp giá trị test statistic dẫn đến chấp nhận $H_o$
Null Distribution: Phân phối của test statistic giả sử $H_o$ đúng

**Ví dụ thực tế trong kinh doanh**
Marketing Campaign: $H_o$: chiến dịch không ảnh hưởng đến mua hàng
Website Layout: $H_o$: thay đổi layout không tác động đến traffic
Product Quality: $H_o$: sản phẩm đạt tiêu chuẩn kích thước

**Type I và Type II Errors**

Type I Error (False Positive)
**Định nghĩa**: Bác bỏ $H_o$ khi $H_o$ đúng
Ví dụ: Kết luận đồng xu lệch khi thực tế nó công bằng
**Xác suất**: Được ký hiệu α (alpha), thường = 0.05
Type II Error (False Negative)
**Định nghĩa**: Không bác bỏ $H_o$ khi $H_o$ sai
Ví dụ: Kết luận đồng xu công bằng khi thực tế nó lệch
Xác suất: Được ký hiệu β (beta)

**Power of Test**
**Công thức**: Power = 1 - β
**Ý nghĩa:** Khả năng bác bỏ đúng $H_o$ khi $H_o$ sai

**P-values và Significance Levels**

P-value
**Định nghĩa:** Xác suất quan sát được kết quả cực đoan như dữ liệu mẫu nếu $H_o$ đúng
Ngưỡng phổ biến: 0.1, 0.05, 0.01

Significance Level (α)
**Chọn trước**: Phải được xác định trước khi kiểm định để tránh P-hacking
Trade-off: α thấp giảm Type I error nhưng có thể tăng Type II error

**Bonferroni Correction**

Mục đích: Kiểm soát Family-Wise Error Rate khi thực hiện nhiều kiểm định đồng thời
Công thức: α_adjusted = α / n (n = số lượng kiểm định)

Ví dụ: Nếu α = 0.05 và có 10 kiểm định, thì α_adjusted = 0.05/10 = 0.005
Trade-offs:
Ưu điểm: Giảm Type I errors
Nhược điểm: Tăng Type II errors, yêu cầu effect size hoặc sample size lớn hơn

**Correlation vs. Causation**

**Correlation (Tương quan)**
**Định nghĩa**: Mối liên hệ thống kê giữa hai biến số
Không có nghĩa là nhân quả: Hai biến thay đổi cùng nhau nhưng không chứng minh mối quan hệ nhân-quả

**Causation (Nhân quả)**
**Định nghĩa:** Thay đổi trong một biến trực tiếp gây ra thay đổi trong biến khác
Yêu cầu chứng minh: Cần controlled experiments hoặc longitudinal studies

**Confounding Variables**
**Định nghĩa**: Biến thứ ba ảnh hưởng đến cả X và Y, tạo ra tương quan misleading
**Ví dụ kinh điển**: Bán kem và đuối nước tăng cùng nhau do thời tiết nóng
**Spurious Correlations**
**Định nghĩa**: Tương quan xuất hiện do trùng hợp, không phản ánh mối quan hệ nhân quả thực
**Ví dụ nổi tiếng:**

Tuổi Miss America và tỷ lệ giết người
Số lượng cò và số trẻ em sinh ra ở Hà Lan
Doanh số rạp chiếu phim và giá vé tăng cùng nhau

**Cách nhận biết và xử lý**
**Third Variable Problem:** Tìm kiếm biến nhiễu có thể ảnh hưởng cả hai biến
**Directionality Problem:** Xác định hướng nhân quả thực sự
**Experimental Design:** Sử dụng randomized controlled trials để establish causation