



# GIẢI PHÁP QUẢN LÝ TỒN KHO HIỆU QUẢ

## **Nhóm 5TDR:**

Trần Đức Trung  
Phan Quốc Huy  
Nguyễn Minh Trí

Trần Phi Hùng  
Nguyễn Đông Hải

# NỘI DUNG

1

## INTRODUCTION

Tổng quan vấn đề

2

## PROBLEM STATEMENT

Mô tả chi tiết vấn đề

3

## SOLUTION OVERVIEW

Tổng quan giải pháp AI

4

## METHODOLOGIES

Mô tả kiến trúc và cấu trúc của mô hình AI sử dụng

# NỘI DUNG

5

## CORE FUNCTIONALITY

Liệt kê các tính năng chính của sản phẩm

6

## PERFORMANCE METRICS

Xác định các chỉ số chính để đánh giá sự thành công của sản phẩm

7

## TIMELINE & ROADMAPS

Xác định các cột mốc để phát triển và triển khai sản phẩm

8

## CONCLUSION

Tổng hợp ý chính về sản phẩm

# 1. INTRODUCTION

Hàng tồn kho là những mặt hàng, sản phẩm được doanh nghiệp giữ lại để bán sau cùng, có vai trò quan trọng trong hoạt động sản xuất kinh doanh.

Việc quản lý hàng tồn kho hiệu quả luôn là một vấn đề mà bất kỳ doanh nghiệp bán lẻ nào cũng phải quan tâm đến nếu muốn đạt được hiệu suất kinh doanh tốt nhất.



# 1. INTRODUCTION

Trong thời đại 4.0 ngày nay, với việc công nghệ ngày càng phát triển, việc áp dụng AI vào các khía cạnh của cuộc sống đang trở nên ngày càng phổ biến hơn bao giờ hết.

Trước tình hình đó, nhóm đề xuất một giải pháp AI để có thể giúp các doanh nghiệp bán lẻ có thể dễ dàng và hiệu quả hơn trong việc quản lý hàng tồn kho và tối ưu hoá lợi nhuận của mình.



## 2. PROBLEM STATEMENT

Việc quản lý tồn kho luôn là một thách thức đối với mỗi doanh nghiệp bán lẻ, cần có sự cân bằng giữa 2 yếu tố:

- **Có đủ hàng tồn kho** để duy trì hoạt động kinh doanh
- **Tránh tồn kho dư thừa** gây tổn kém chi phí

Để quản lý tồn kho một cách hiệu quả, Bộ phận kho của các doanh nghiệp cần phải **thực hiện nhiều nhiệm vụ** như theo dõi lượng tồn kho, dự đoán nhu cầu, và cập nhật định kỳ thông tin về hàng tồn kho. Tuy nhiên, với sự phát triển của **trí tuệ nhân tạo (AI)** có thể giúp bộ phận Kho có thể **tự động theo dõi, giám sát** tình trạng hàng tồn kho và **hỗ trợ đưa ra các quyết định** mua/trữ hàng tồn kho một cách hợp lý.

## 2. PROBLEM STATEMENT

**Mô hình quản lý hàng EOD (Economic Order Quantity)** là một mô hình dùng để quản lý hàng tồn kho, mang trong mình tính chất định lượng dùng để xác định **số lượng hàng hóa tối ưu** cần mua tại một thời điểm. Điều này được thực hiện trên 2 cơ sở là 2 loại chi phí tồn tại trong doanh nghiệp.

- Chi phí dùng để đặt mua hàng
- Chi phí dùng để trữ hàng tồn kho

Trong đó:

Gọi O là chi phí cho mỗi lần đặt hàng:

S là nhu cầu hàng cần trong kỳ.

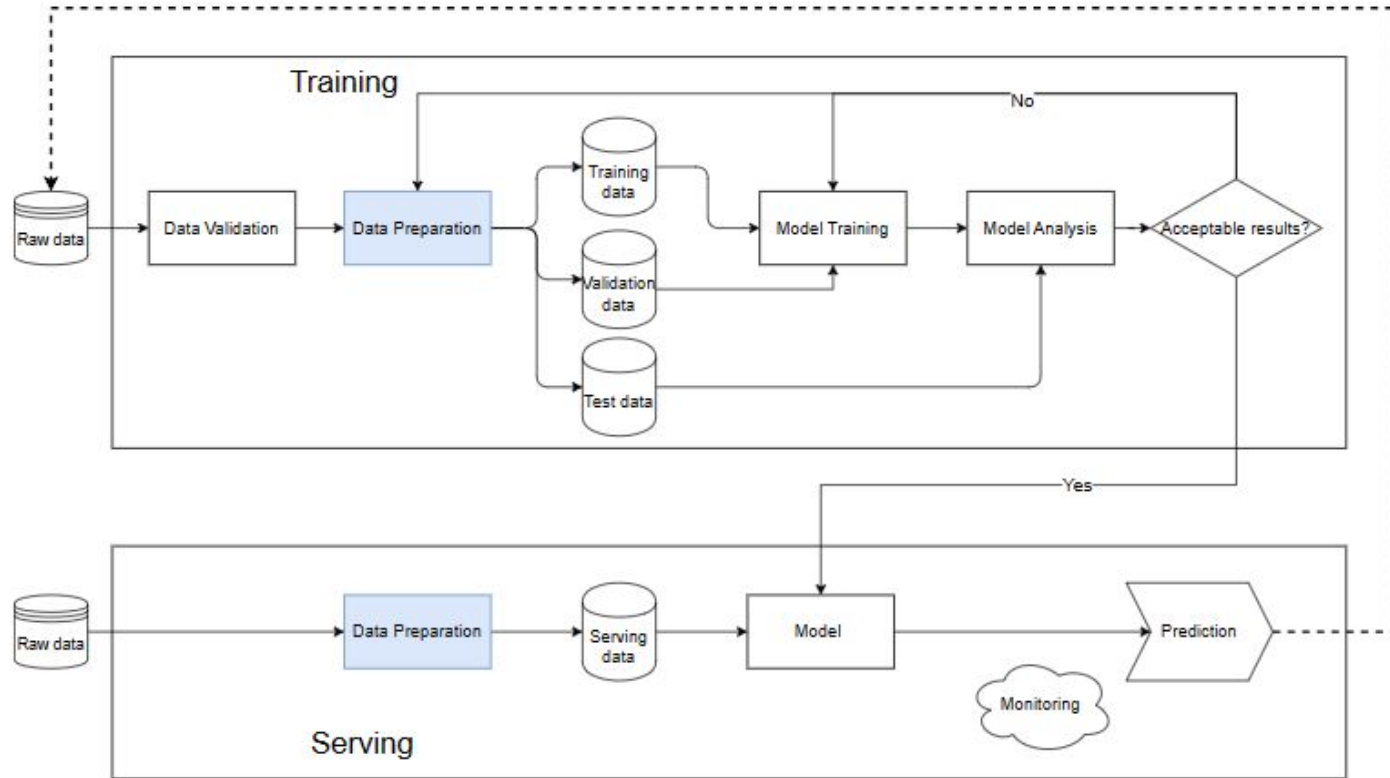
Q là lượng hàng tồn kho cho mỗi lần đặt hàng.

C: Chi phí tồn trữ cho một đơn vị hàng lưu kho

$$\text{Sản lượng tồn kho tối ưu } Q^* = \sqrt{\frac{2 \times S \times O}{C}}$$
$$\text{Thời gian dự trữ tối ưu } T^* = \frac{365 \times Q^*}{S} \text{ (ngày)}$$



### 3. SOLUTION OVERVIEW





### 3. SOLUTION OVERVIEW

Với ý tưởng xây dựng 1 Machine learning/Deep learning pipeline sẽ lấy những data liên quan tới doanh số mua bán sản phẩm ở phía doanh nghiệp (cụ thể ở đây là trong thời trang) cũng như là thông tin sản phẩm, thông tin quản lý kho hàng, để đưa ra dự đoán cho câu hỏi “khi nào nên mua và mua bao nhiêu hàng”

Pipeline trên sẽ có 2 khối pha chính là “**Training**” và “**Serving**” thể hiện luồng dữ liệu trong quá trình huấn luyện và chạy thực tế.

### 3. SOLUTION OVERVIEW

#### a. Pha “Training”

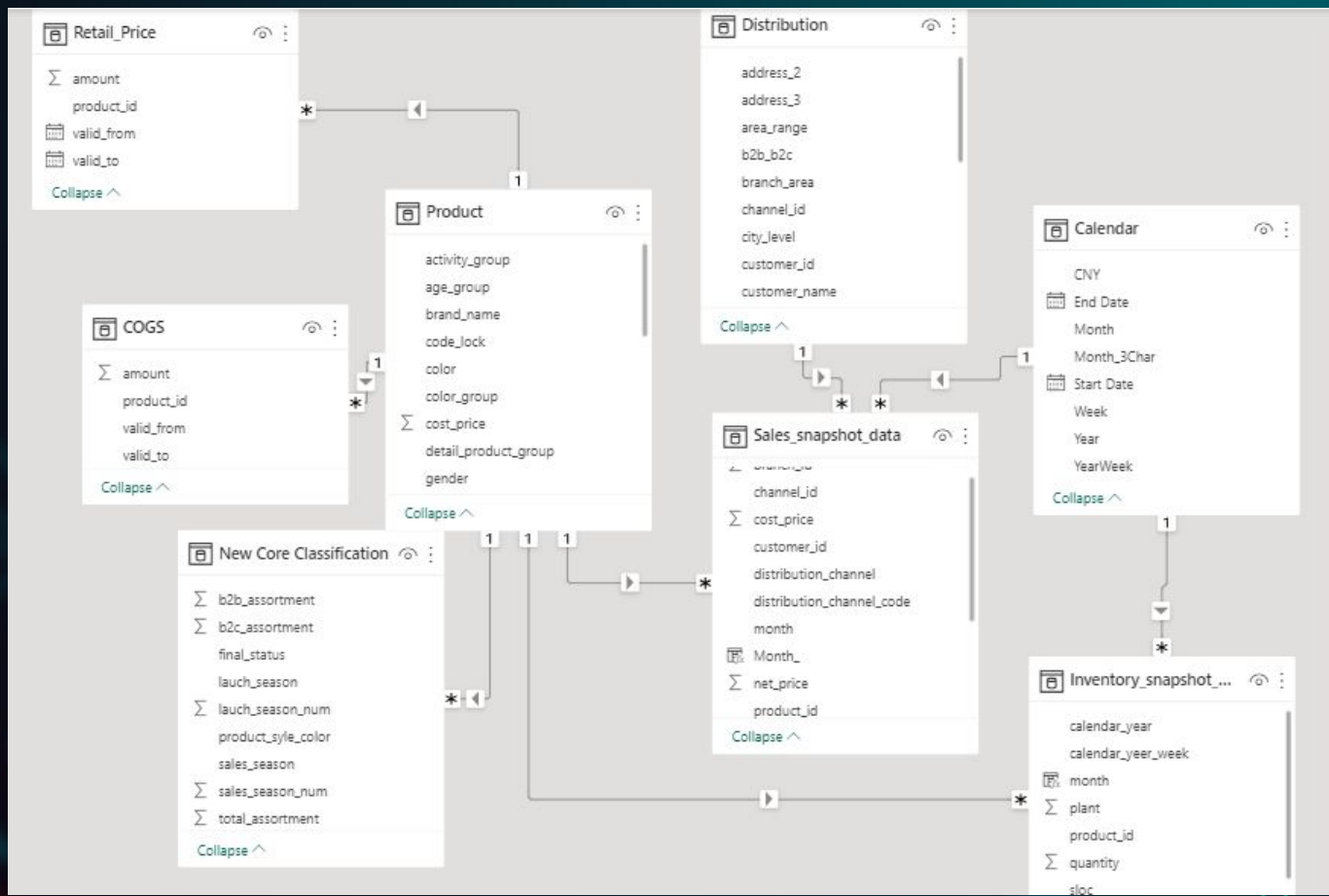
- **“Data Validation”** là bước kiểm định dữ liệu. Tại bước này, dữ liệu cần được kiểm tra xem có tương tự như dữ liệu hiện có trong cơ sở dữ liệu hay không. Lưu ý rằng trong các bài toán thực tế, dữ liệu mới thường xuyên được sinh ra, và mô hình cần thường xuyên được cập nhật dựa trên dữ liệu mới này để “bắt kịp xu thế”. Nếu để mô hình huấn luyện trên dữ liệu không nhất quán, rất nhiều khả năng mô hình sẽ có kết quả tệ.
- Việc kiểm tra dữ liệu có tương tự hay không có thể được xác định dựa trên các thông số thống kê như mean, phương sai, phân phối...
- Hay 1 số phương pháp khác như: data type check, range check, format check...

# 3. SOLUTION OVERVIEW

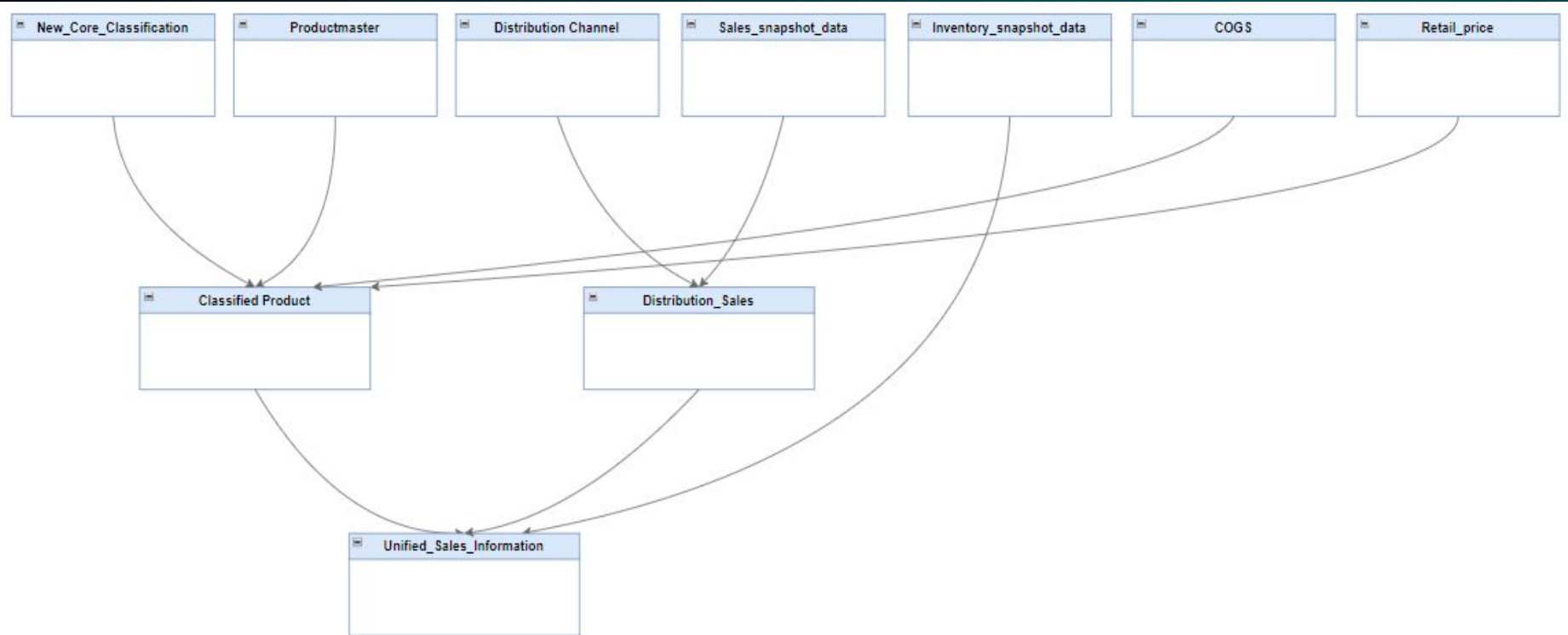
## a. Pha “Training”

- Sau bước Data Validation, dữ liệu thô được làm sạch, tạo đặc trưng trong bước “**Data Preparation**”. Ta chia dữ liệu thành 3 domain chính:
  - Detailed information of product: bao gồm Retail\_price, COGS, New Core classification, and Productmaster
  - Sales and distribution information: bao gồm Distribution, Sales\_snapshots\_data
  - Inventory information: Inventory\_snapshot\_data.

- Sau khi data collection, ta được dataset chứa các thông tin về tồn kho, sản phẩm, nhà phân phối ...



- Tiến hành quá trình ETL để tổ chức lại dữ liệu, chuẩn hóa định dạng và chuyển đổi nó để đáp ứng nhu cầu phân tích và quản lý dữ liệu.



- Ta kết hợp data ở New\_Core\_Classification và Productmaster được dataset mới chứa các thông tin chi tiết về sản phẩm và phân loại đặc tả về cách sản phẩm được bán ra.
- Dựa vào data ở COGS và Retail\_price ta có thể fill missing values cho các giá trị bị thiếu ở listing\_price và cost\_price còn thiếu ở Classified Product.
- Kết hợp data ở Sales\_snapshot\_data và Distribution Channel được dataset mới là Distribution\_Sales chứa thông tin bán hàng của mỗi nhà phân phối đối với từng sản phẩm ở mỗi tháng và thông tin về các nhà phân phối đó.
- Từ Classified Product, Distribution\_Sales và Inventory\_snapshot data ta kết hợp lại thành Unified\_Sales\_Information chứa dữ liệu về thông tin bán hàng của mỗi nhà phân phối với từng sản phẩm ở mỗi tháng, thông tin chi tiết về sản phẩm đó, cách sản phẩm được bán ra và thông tin về nhà phân phối đó.

- Ta thu được các thông tin cần thiết :

Unified_Sales_Information	
color	channel_id
color_group	region
price_group	city_level
gender	store_concept
product_group	trade_term
detail_product_group	area_range
shoe_product	store_type
size_group	urbanization
size	branch_area
age_group	warehouse_area
activity_group	start_month
image_copyright	start_year
lifestyle_group	month
heel_height	week
cost_price	site
product_id	branch_id
product_syle_color	distribution_channel_code
brand_name	sold_quantity
vendor_name	cost_price
lauch_season_num	net_price
sales_season	product_id
sales_season_num	plant
final_status	calendar_year
b2c_assortment	calendar_year_week
b2b_assortment	sloc
total_assortment	quantity
b2b_b2c	total_amount



### 3. SOLUTION OVERVIEW

#### a. Pha “Training”

- Sau bước “Data Preparation”, ta tách data ra thành tập huấn luyện (training data), tập kiểm định (validation data), tập kiểm tra (test data).
- Dữ liệu huấn luyện được đưa vào “Model Training” để huấn luyện mô hình. Dữ liệu kiểm định được sử dụng trong quá trình huấn luyện này để tinh chỉnh tham số.
- Mô hình sau khi huấn luyện được đưa vào “Model Analysis” để phân tích chất lượng của mô hình sau khi huấn luyện.
- Nếu kết quả trong bước “Model Analysis” đáp ứng được yêu cầu của bài toán thì sẽ được dùng chạy dữ liệu thực tế ở bước “Serving”. Nếu không, ta cần điều chỉnh hoặc xây dựng lại mô hình, coi lại và tìm các đặc trưng phù hợp hơn hoặc lấy thêm dữ liệu.

### 3. SOLUTION OVERVIEW

#### b. Pha “Serving”

- Sau khi huấn luyện mô hình, ta không áp dụng ngay cho toàn bộ dữ liệu thực tế mà chỉ áp dụng trên một phần nhỏ để so sánh với mô hình hiện tại. Nếu chất lượng chấp nhận được, tỷ lệ dữ liệu mà mô hình mới dự đoán được nâng dần lên cho đến khi chạy trên toàn bộ dữ liệu.
- Data Preparation là bước làm sạch và tạo dữ liệu phải giống hệt như những gì đã xây dựng mô hình trong pha “Training”. Ta cần đảm bảo dữ liệu đầu vào của mô hình có những tính chất giống hệt như những gì nó được nhìn thấy khi huấn luyện.
- Kết quả mà mô hình dự đoán có thể ảnh hưởng trực tiếp đến hành vi người dùng. Phản ứng của người dùng, trong rất nhiều trường hợp được sử dụng để huấn luyện những mô hình tiếp theo.

## 4. METHODOLOGIES

Chi tiết về những model mà nhóm dự định sẽ áp dụng trong pipeline, để giải quyết cho câu hỏi “khi nào mua và cần mua bao nhiêu ?” trong bài toán quản lý kho hàng, nhóm sẽ tập chung vào các thuật toán regression để dự đoán các output bên. Bên cạnh đó với dữ liệu bán hàng và hàng tồn kho, các thuật toán về ensemble algorithm tương đối phù hợp đặc biệt với những dữ liệu lớn và độ phức tạp cao

Gọi  $Q^*$  là sản lượng hàng tồn kho tối ưu

$T^*$  là thời gian dự trữ tối ưu

$$\text{Sản lượng tồn kho tối ưu } Q^* = \sqrt{\frac{2 \times S \times O}{C}}$$

$$\text{Thời gian dự trữ tối ưu } T^* = \frac{365 \times Q^*}{S} (\text{ngày})$$

công thức

Gọi  $O$  là chi phí cho mỗi lần đặt hàng.

$S$  là nhu cầu hàng cần trong kỳ.

$Q$  là lượng hàng tồn kho cho mỗi lần đặt hàng.

$C$ : Chi phí tồn trữ cho một đơn vị hàng lưu kho

## 4. METHODOLOGIES

- Các thuật toán nhóm dự định sử dụng trong giai đoạn model training:
  - + Linear/Ridge/Lasso regression
  - + Decision tree
  - + Random forest
  - + CatBoost
  - + XGBoost
  - + Stacking model
  - + Multilayer perceptron

## 4. METHODOLOGIES

- **Linear Regression** là một thuật toán trong machine learning để mô hình hóa mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc.
- Mục tiêu: Tìm đường thẳng (hoặc siêu phẳng) sao cho tổng bình phương sai số là nhỏ nhất.

Ưu điểm Linear Regression:

- Dễ hiểu và triển khai.
- Thực hiện tốt trên dữ liệu có mối quan hệ tuyến tính đơn giản.
- Yêu cầu ít siêu tham số để điều chỉnh.

Nhược điểm Linear Regression:

- Giả sử mối quan hệ tuyến tính, không linh hoạt với mối quan hệ phi tuyến tính.
- Nhạy cảm với nhiễu (outliers) trong dữ liệu.
- Không thích hợp cho dữ liệu phức tạp với nhiều biến độc lập và mối quan hệ phức tạp.

## 4. METHODOLOGIES

- **Ridge Regression** là một biến thể của linear regression với mục tiêu là giảm overfitting bằng cách thêm vào hàm mục tiêu một thành phần kiểm soát (regularization term). hình hóa mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc.
- Mục tiêu là tối thiểu hóa tổng bình phương sai số và tổng bình phương của hệ số (sử dụng tham số kiểm soát  $\alpha$ ).

Ưu điểm Ridge Regression:

- Giảm overfitting và ổn định mô hình trên dữ liệu nhiễu.
- Hiệu quả khi có multicollinearity (tương quan cao giữa các biến độc lập).

Nhược điểm Ridge Regression:

- Không loại bỏ các biến không quan trọng mà thực sự đóng góp ít vào mô hình (không thực hiện lựa chọn đặc trưng).

## 4. METHODOLOGIES

- **Lasso Regression** là một biến thể của linear regression được thiết kế để giảm overfitting và thực hiện lựa chọn đặc trưng bằng cách thêm vào hàm mục tiêu một thành phần kiểm soát (regularization term).
- Mục tiêu là tối thiểu hóa tổng bình phương sai số và tổng giá trị tuyệt đối của hệ số (sử dụng tham số kiểm soát  $\alpha$ ).

Ưu điểm Lasso Regression:

- Giảm overfitting và giúp trong việc lựa chọn đặc trưng bằng cách đặt một số hệ số bằng 0.
- Hiệu quả khi có nhiều biến độc lập không quan trọng.

Nhược điểm Lasso Regression:

- Có thể gặp khó khăn khi có multicollinearity cao (tương quan cao giữa các biến độc lập).



## 4. METHODOLOGIES

- **Decision Tree Regression** là một phương pháp trong machine learning để dự đoán giá trị liên tục thay vì nhãn lớp.
- Mô hình dựa trên cây quyết định, trong đó mỗi nút lá của cây đại diện cho một giá trị dự đoán.
- Mục tiêu là tạo ra các quy tắc tuyến tính để phân chia dữ liệu thành các phần nhỏ, mỗi phần có một giá trị dự đoán.

Ưu điểm Decision Tree Regression:

- Dễ hiểu và giải thích.
- Không đòi hỏi chuẩn hóa dữ liệu.
- Xử lý tốt dữ liệu phi tuyến tính và tương đối ổn định với nhiễu.

Nhược điểm Decision Tree Regression:

- Dễ bị overfitting, đặc biệt khi cây quyết định quá sâu.
- Khả năng không tổng quát hoá tốt trên dữ liệu mới nếu không kiểm soát overfitting.

## 4. METHODOLOGIES

- **Random Forest Regression** là một phương pháp ensemble learning dựa trên nhiều cây quyết định (Decision Trees).
- Mỗi cây quyết định được xây dựng trên một tập dữ liệu con được lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện (bootstrap sampling).
- Kết quả dự đoán là sự kết hợp của dự đoán từ nhiều cây, thường thông qua việc trung bình (regression) hoặc đa số phiếu (classification).

Ưu điểm Random Forest Regression:

- Hiệu suất tốt trên nhiều loại dữ liệu và mô hình có khả năng tự động kiểm soát overfitting.
- Khả năng xử lý nhiễu và dữ liệu phi tuyến tính.
- Cung cấp mức độ quan trọng của các đặc trưng.

Nhược điểm Random Forest Regression:

- Khó diễn giải so với một cây quyết định đơn lẻ. Đòi hỏi nhiều tài nguyên tính toán so với một cây quyết định.

## 4. METHODOLOGIES

- **CatBoost** là một thuật toán machine learning dựa trên ensemble learning, được thiết kế đặc biệt cho việc xử lý dữ liệu hạng mục (categorical features) mà không cần mã hóa trước.
- Sử dụng Gradient Boosting trên cây quyết định, với khả năng xử lý tự động missing values và làm việc tốt trên dữ liệu có nhiễu.
- Tự động cân bằng giữa các lớp trong dữ liệu phân loại và cung cấp các tính năng như early stopping để tránh overfitting.

Ưu điểm CatBoost:

- Tự động xử lý dữ liệu hạng mục, giảm công đoạn tiền xử lý.
- Hiệu suất tốt trên dữ liệu không đồng nhất và có nhiễu.
- Có thể chạy hiệu quả mà không đòi hỏi nhiều tinh chỉnh siêu tham số.

Nhược điểm CatBoost:

- Yêu cầu tài nguyên tính toán khá lớn, đặc biệt khi có nhiều cây quyết định và dữ liệu lớn.

## 4. METHODOLOGIES

- **XGBoost** (Extreme Gradient Boosting) là một thuật toán machine learning dựa trên ensemble learning, chủ yếu sử dụng Gradient Boosting Framework.
- XGBoost được thiết kế để tối ưu hóa hiệu suất và độ chính xác của mô hình thông qua các cây quyết định.
- Sử dụng phương pháp tối ưu hóa chất lượng cây và cân bằng giữa bias và variance. Hỗ trợ cả các bài toán phân loại và regression.

Ưu điểm XGBoost:

- Hiệu suất cao và ổn định, thích hợp cho nhiều loại dữ liệu và bài toán.
- Xử lý tốt overfitting thông qua các tham số như learning rate và regularization term. Hỗ trợ parallel processing, làm tăng tốc độ huấn luyện.

Nhược điểm XGBoost:

- Đòi hỏi tài nguyên tính toán cao, đặc biệt là khi xử lý dữ liệu lớn và số cây lớn. Cần sự điều chỉnh tham số kỹ lưỡng để đạt được hiệu suất tốt nhất.

## 4. METHODOLOGIES

- **Stacking** là một phương pháp trong ensemble learning, kết hợp dự đoán từ nhiều mô hình để tạo ra một mô hình cuối cùng mạnh mẽ hơn.
- Quá trình này bao gồm việc huấn luyện một mô hình "meta-learner" để kết hợp dự đoán từ các mô hình cơ bản.
- Các mô hình cơ bản có thể là bất kỳ thuật toán machine learning nào, và dự đoán của chúng được sử dụng để tạo ra tập dữ liệu mới cho meta-learner.
- Stacking giúp cải thiện tính chính xác và ổn định của mô hình.

Ưu điểm Stacking:

- Cải thiện khả năng dự đoán bằng cách kết hợp sức mạnh của nhiều mô hình.
- Linh hoạt và có thể được tùy chỉnh cho nhiều loại dữ liệu và bài toán.
- Hiệu quả đặc biệt khi sử dụng các mô hình có sự đa dạng cao.

Nhược điểm Stacking:

- Đòi hỏi nhiều tài nguyên tính toán và thời gian huấn luyện so với một mô hình đơn lẻ. Cần chú ý đến việc quản lý overfitting và lựa chọn tham số.

## 4. METHODOLOGIES

- **Multilayer Perceptron** là một kiểu mô hình fully connected neural network được tổ chức thành nhiều tầng (lớp) nơ-ron.
- Bao gồm tầng đầu vào, một hoặc nhiều tầng ẩn, và tầng đầu ra.
- Mỗi tầng ẩn chứa nhiều nơ-ron và có các trọng số tương ứng.
- Sử dụng hàm kích hoạt để tạo ra đầu ra của mỗi nơ-ron và học các biểu diễn phức tạp từ dữ liệu. Dùng backpropagation để điều chỉnh trọng số và giảm thiểu hàm mất mát, hoặc sử dụng các phương pháp tối ưu khác nhau

Ưu điểm Multilayer Perceptron:

- Có khả năng học được biểu diễn phức tạp và giải quyết nhiều bài toán máy học.
- Linh hoạt và có thể áp dụng cho nhiều loại dữ liệu.

Nhược điểm Multilayer Perceptron:

- Yêu cầu nhiều dữ liệu để đạt hiệu suất tốt.
- Nhạy cảm với sự chọn lựa tham số và có thể bị overfitting nếu không kiểm soát.
- Có thể đòi hỏi nhiều tài nguyên tính toán, đặc biệt là khi mô hình lớn.

## 4. METHODOLOGIES

Bên cạnh đó để tối ưu quá trình training model, 1 số thuật toán tối ưu mà nhóm dự định sử dụng để giảm bớt gánh nặng tính toán so với Gradient Descent thông thường:

### **Mini-batch Gradient Descent**

Ưu điểm của mini-batch GD bao gồm:

- Tăng tốc độ hội tụ: So với Gradient Descent truyền thống, mini-batch GD thường hội tụ nhanh hơn do sử dụng tính toán vector hóa và được tận dụng từ các thư viện tính toán hiệu quả.
- Ứng dụng tốt cho dữ liệu lớn: Mini-batch GD cho phép xử lý hiệu quả dữ liệu lớn mà không cần đưa toàn bộ dữ liệu vào bộ nhớ.
- Giảm nhiễu: So với SGD, mini-batch GD giảm đi nhiễu trong quá trình cập nhật trọng số do sử dụng một lượng dữ liệu lớn hơn trong mỗi bước.



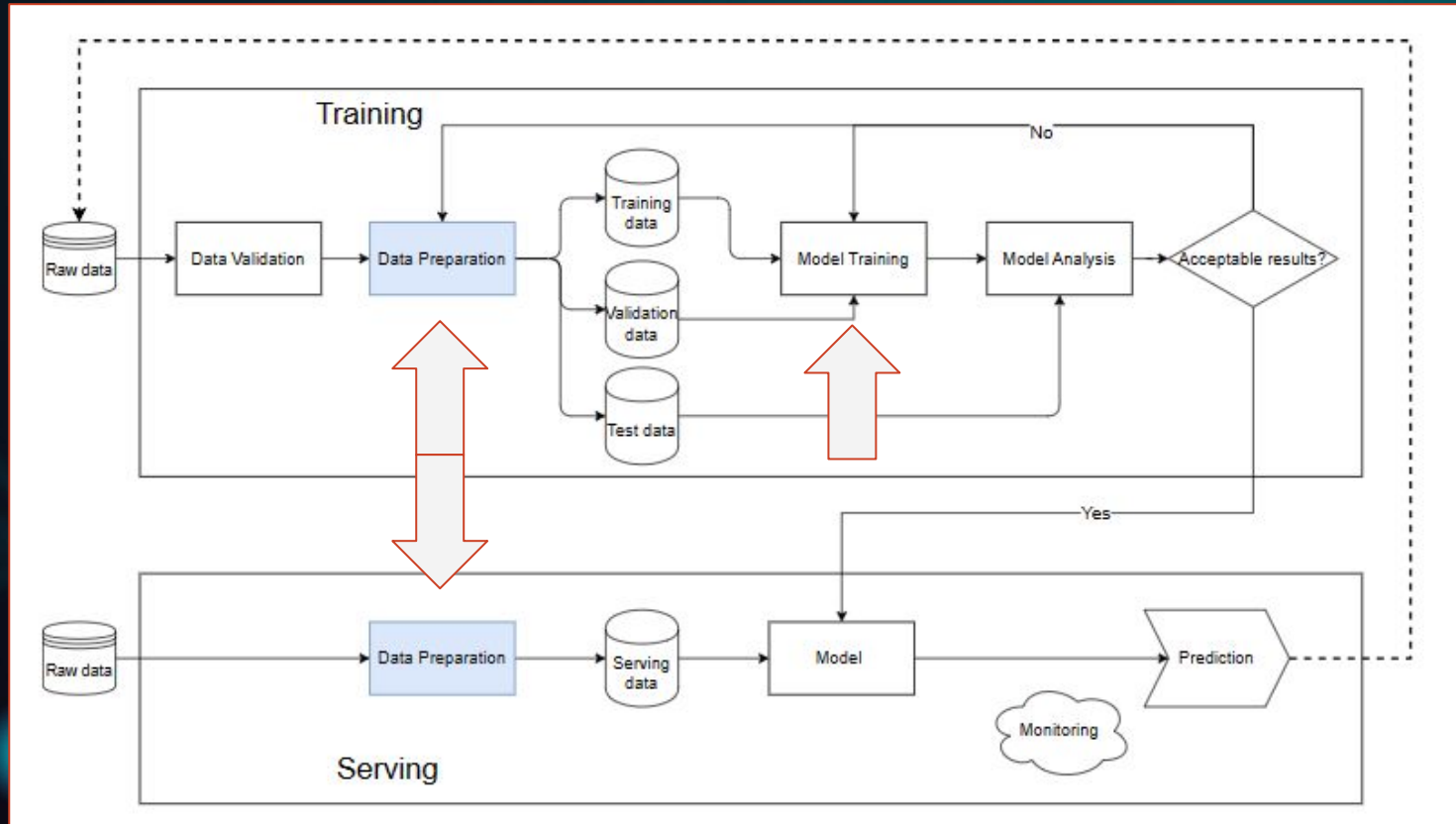
## 4. METHODOLOGIES

### **Stochastic Gradient Descent (SGD)**

Ưu điểm của SGD bao gồm:

- Hiệu quả với dữ liệu lớn: SGD thường hiệu quả với dữ liệu lớn vì nó chỉ cần tính toán độ dốc trên một mẫu dữ liệu (ngẫu nhiên) trong mỗi bước, giảm bớt gánh nặng tính toán so với Gradient Descent truyền thống.
- Khả năng làm việc với dữ liệu động: SGD có thể được sử dụng hiệu quả khi dữ liệu đang thay đổi liên tục, vì nó có thể được cập nhật mỗi khi có một mẫu mới được thêm vào.
- Tránh bị kẹt ở các điểm địa phương: Do việc sử dụng mẫu ngẫu nhiên trong mỗi bước, SGD có khả năng tránh bị kẹt ở các điểm địa phương của không gian tham số.

# 5. CORE FUNCTIONALITY



## 6. PERFORMANCE METRICS

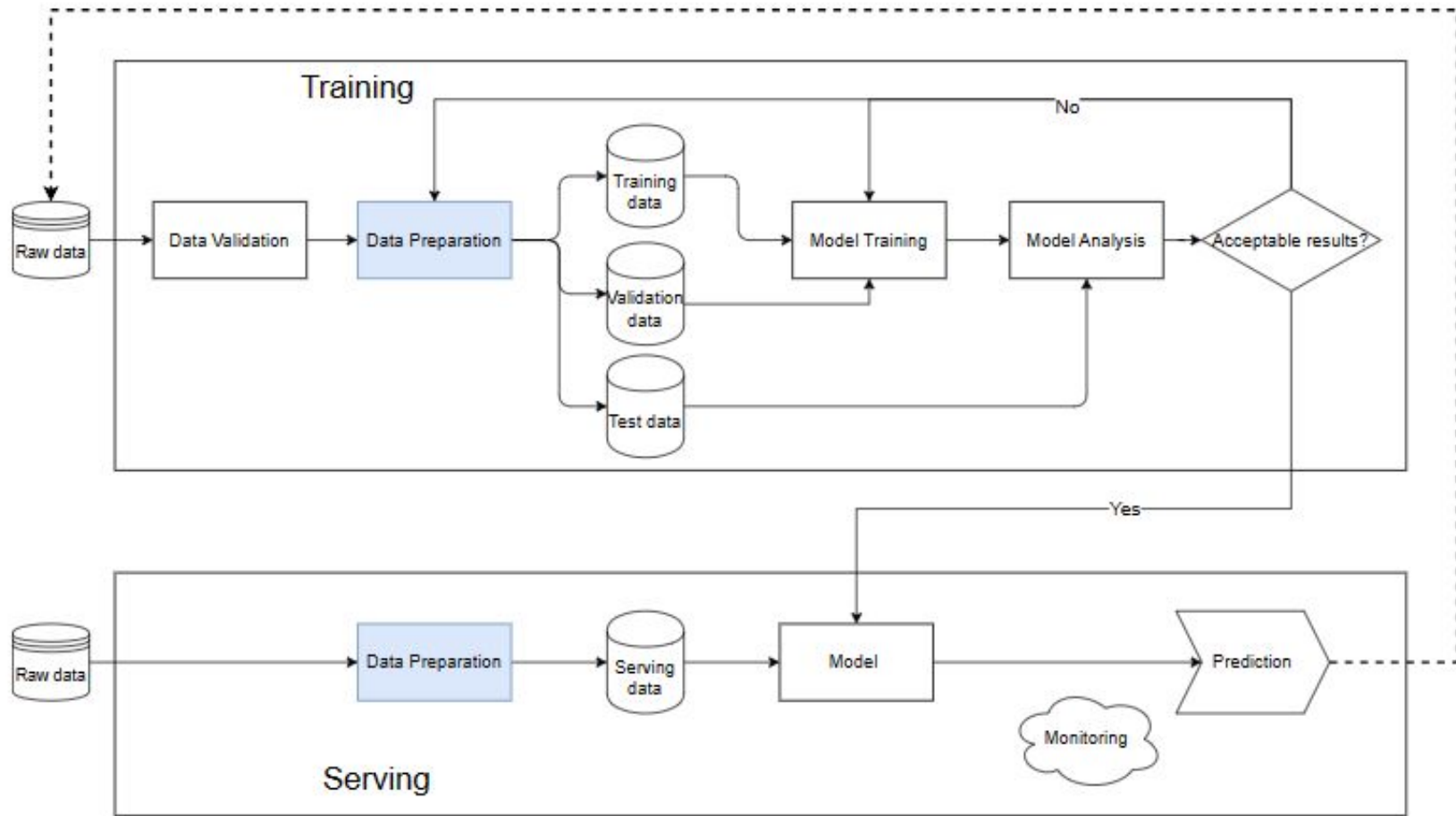
**Metric sẽ được sử dụng để đánh giá MVP:**

- MAE: Trung bình độ lệch tuyệt đối
- MSE: Trung bình lỗi bình phương
- RMSE: Căn bậc 2 của MSE
- R-squared: hệ số xác định, càng cao càng tốt

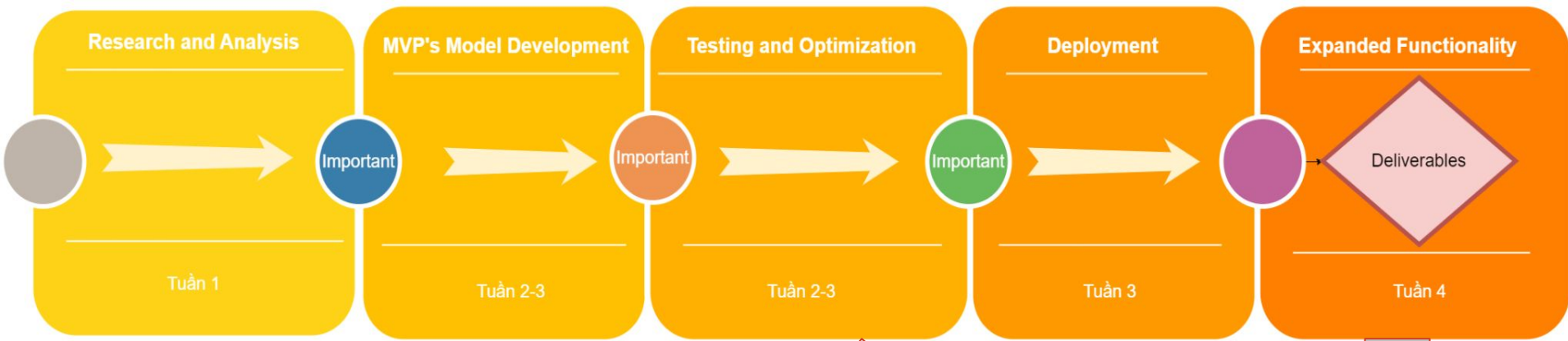
## 6. PERFORMANCE METRICS

**MVP's performance sẽ được đánh giá tại bước model analysis trong pha “Training”**

- Mô hình sau khi huấn luyện được đưa vào “Model Analysis” để phân tích chất lượng của mô hình sau khi huấn luyện.
- Nếu kết quả trong bước “Model Analysis” đáp ứng được yêu cầu của bài toán thì sẽ được dùng chạy dữ liệu thực tế ở bước “Serving”. Nếu không, ta cần điều chỉnh hoặc xây dựng lại mô hình, coi lại và tìm các đặc trưng phù hợp hơn hoặc lấy thêm dữ liệu.



# 7. TIMELINE AND ROADMAP



Có thể quay lại bước 3 nếu có sự thay đổi, tinh chỉnh về mục tiêu cũng như mong muốn nhu cầu khác không phù hợp so với mục tiêu hiện tại.

## 8. Limitations and Future Enhancements

- Đầu tiên là về cơ sở hạ tầng phát triển mô hình, do mô hình hoạt động dựa trên những dữ liệu nhưng đối với những công ty bán lẻ tại Việt Nam hiện nay, dữ liệu lưu trữ rất lớn. Vì vậy cần phải có nơi lưu trữ an toàn và đủ sức chứa, và đặc biệt nơi lưu trữ lớn - cần phải chi trả những chi phí vô cùng lớn để duy trì và triển khai mô hình song hành.
- Data không đủ phù hợp với mô hình đang được triển khai, đặc biệt là triển khai thực tiễn.
- Model lỗi thời, chưa đủ nguồn nhân lực phát triển mở rộng mô hình đáp ứng toàn diện với nhu cầu của thị trường trong tương lai.



## 9. Conclusion

Tóm lại, sản phẩm AI quản lý tồn kho hứa hẹn mang lại lợi ích đáng kể trong việc **tối ưu hóa chi phí** và **nâng cao hiệu suất kinh doanh** cho cửa hàng bán lẻ. Tuy nhiên, đây chỉ là một ý tưởng ở giai đoạn **lý thuyết** và việc chuyển ý tưởng này thành hiện thực trong thực tế kinh doanh đòi hỏi **sự hỗ trợ và hướng dẫn chặt chẽ** từ các mentor, giúp nhóm cụ thể hóa và hoàn thiện thêm về sản phẩm của mình. Đây là một hành trình mà sự đồng hành và sự hỗ trợ từ người đi trước sẽ góp phần quan trọng vào sự thành công của dự án.

**Thank you for reading!**