

**NHÓM 19**

# **WALMART SALES FORECAST ANALYSIS**

CS116.P22

**BÁO CÁO**



**NHÓM 19**

# DANH SÁCH THÀNH VIÊN

**LÊ CHÍ HOÀNG**

235205020

**ĐẶNG VĂN VỸ**

23521825

**NGÔ LÊ NHẬT HOÀNG**

23520526

**NGUYỄN HOÀNG MINH**

23520937



# MỤC LỤC

**I**

**GIỚI THIỆU  
BỘ DỮ LIỆU**

**II**

**EXPLORATORY DATA  
ANALYSIS**

**III**

**DATA PREPROCESSING**

**IV**

**FEATURE ENGINEERING**

**V**

**MODEL BUILDING**



# **I. BỘ DỮ LIỆU WALMART SALES FORECASTING**





# WALMART SALES FORECASTING

## GIỚI THIỆU

- Bộ dữ liệu Walmart Sales Forecast được xây dựng nhằm mục đích dự báo doanh thu hàng tuần của các cửa hàng thuộc chuỗi bán lẻ Walmart.
- Đây là một bài toán quan trọng trong lĩnh vực bán lẻ, giúp các nhà quản lý có thể tối ưu hóa kho hàng, nhân sự và lập kế hoạch kinh doanh dựa trên các yếu tố tác động từ thị trường.



## MỤC TIÊU DỰ ÁN

- Phân tích và dự báo doanh thu hàng tuần của chuỗi bán lẻ Walmart.
- Hỗ trợ quản lý trong việc lập kế hoạch kho hàng, nhân sự, chiến lược kinh doanh dựa trên dự báo chính xác.





# LÝ DO CHỌN BỘ DỮ LIỆU

01.

## ỨNG DỤNG THỰC TẾ CAO

- Là bài toán quan trọng trong ngành bán lẻ
- Dữ liệu được lấy từ Walmart- chuỗi bán lẻ lớn

02.

## CHỨA NHIỀU YẾU TỐ ẢNH HƯỞNG

- Kinh tế: Giá nhiên liệu, CPI, tỷ lệ thất nghiệp
- Thời tiết : Nhiệt độ
- Khuyến mãi : Markdown
- Ngày lễ : IsHoliday

03.

## CÓ THỂ THỬ NHIỀU PHƯƠNG PHÁP MACHINE LEARNING

- Random Forest, XGBoost
- Mạng nơ-ron, LSTM, ARIMA
- Xử lý dữ liệu bị thiếu, trích xuất đặc trưng



## THÔNG TIN VỀ BỘ DỮ LIỆU

### 1. Input và Output của bài toán

- Input: Các yếu tố ảnh hưởng đến doanh thu, bao gồm thông tin về cửa hàng, thời gian, điều kiện thời tiết, CPI, tỷ lệ thất nghiệp, chương trình giảm giá...
- Output: Doanh thu hàng tuần của từng cửa hàng và từng phòng ban (Weekly\_Sales).

### 2. Chi tiết các tệp trong bộ dữ liệu

Tên file	Số dòng	Số cột	Mô tả
train.csv	421.570	5	Dữ liệu huấn luyện, chứa thông tin doanh thu hàng tuần theo cửa hàng và phòng ban
test.csv	115.064	4	Dữ liệu kiểm tra, giống train.csv nhưng không có cột doanh thu
features.csv	8.190	12	Các yếu tố ảnh hưởng đến doanh thu: thời tiết, CPI, khuyến mãi, giá nhiên liệu, v.v.
stores.csv	45	3	Thông tin về từng cửa hàng: loại cửa hàng và diện tích



## **II. EXPLORATORY DATA ANALYSIS**





# PHƯƠNG PHÁP PHÂN TÍCH

01.

**PHÂN TÍCH ĐƠN BIẾN**

02.

**PHÂN TÍCH TƯƠNG QUAN  
HAI BIẾN**

03.

**PHÂN TÍCH TƯƠNG QUAN  
ĐA BIẾN**



# PHƯƠNG PHÁP PHÂN TÍCH

Để tiện cho việc phân tích và xử lý sau này, chúng ta sẽ gộp file train, features, stores thành data train. Đối với file test cũng tương tự

## 1. PHÂN TÍCH ĐƠN BIẾN

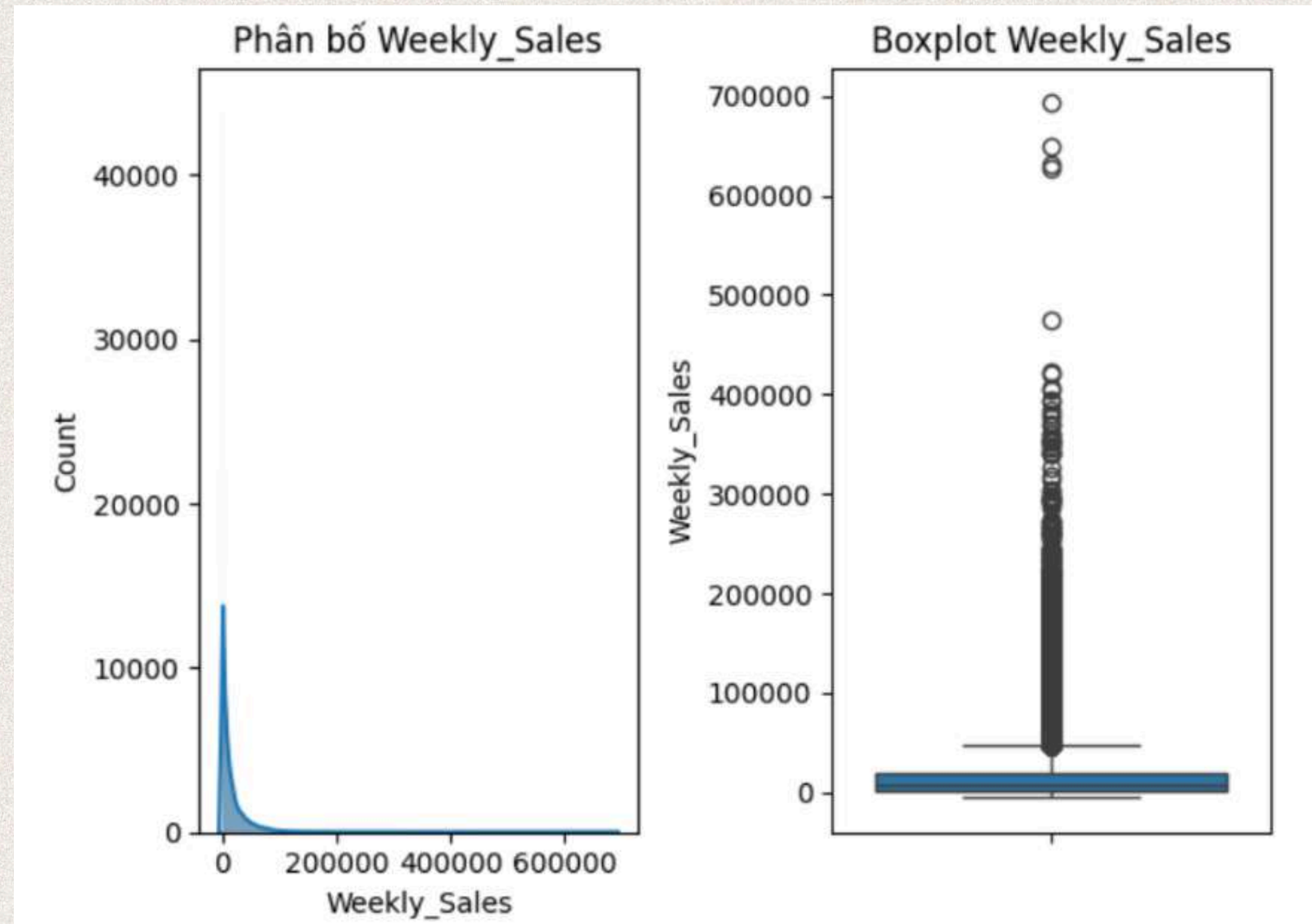
Phân tích phân bố của từng biến riêng lẻ:

- Histogram, Boxplot: dùng cho biến Weekly\_Sales, Size, Temperature, v.v.
- Mục đích:
  - Phát hiện outliers
  - Hiểu phân bố (lệch trái/phải, tập trung, cực trị)



# PHƯƠNG PHÁP PHÂN TÍCH

## 1. PHÂN TÍCH ĐƠN BIẾN

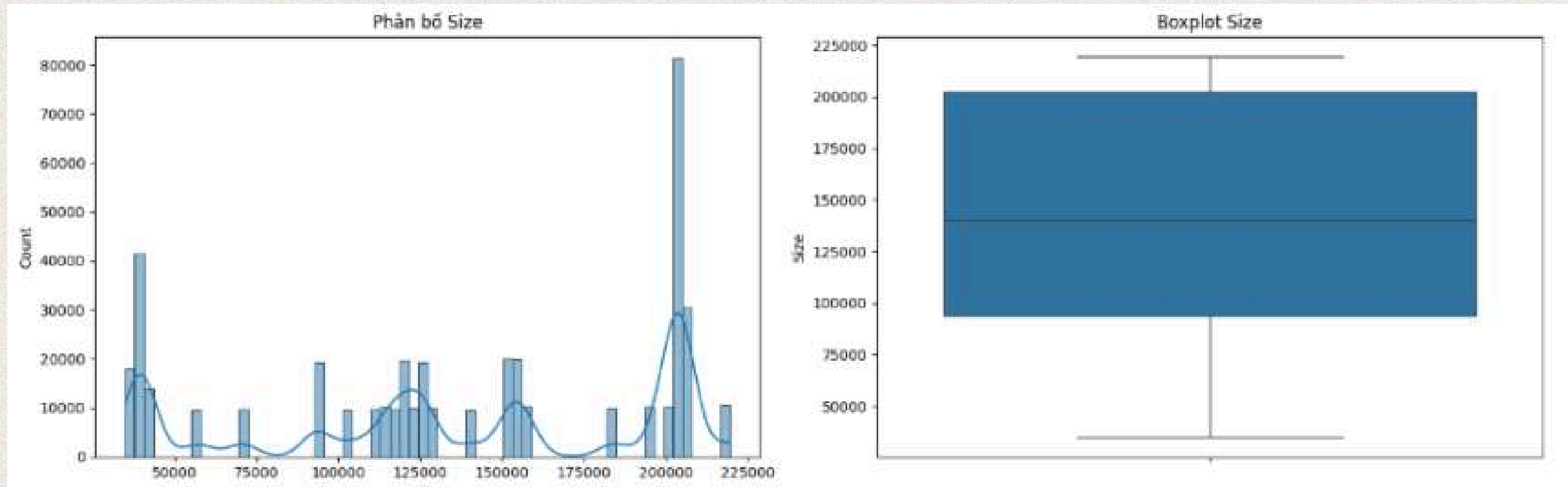


**Biểu đồ phân bố dữ liệu doanh số hàng tuần  
(Weekly\_Sales)**



# PHƯƠNG PHÁP PHÂN TÍCH

## 1. PHÂN TÍCH ĐƠN BIẾN

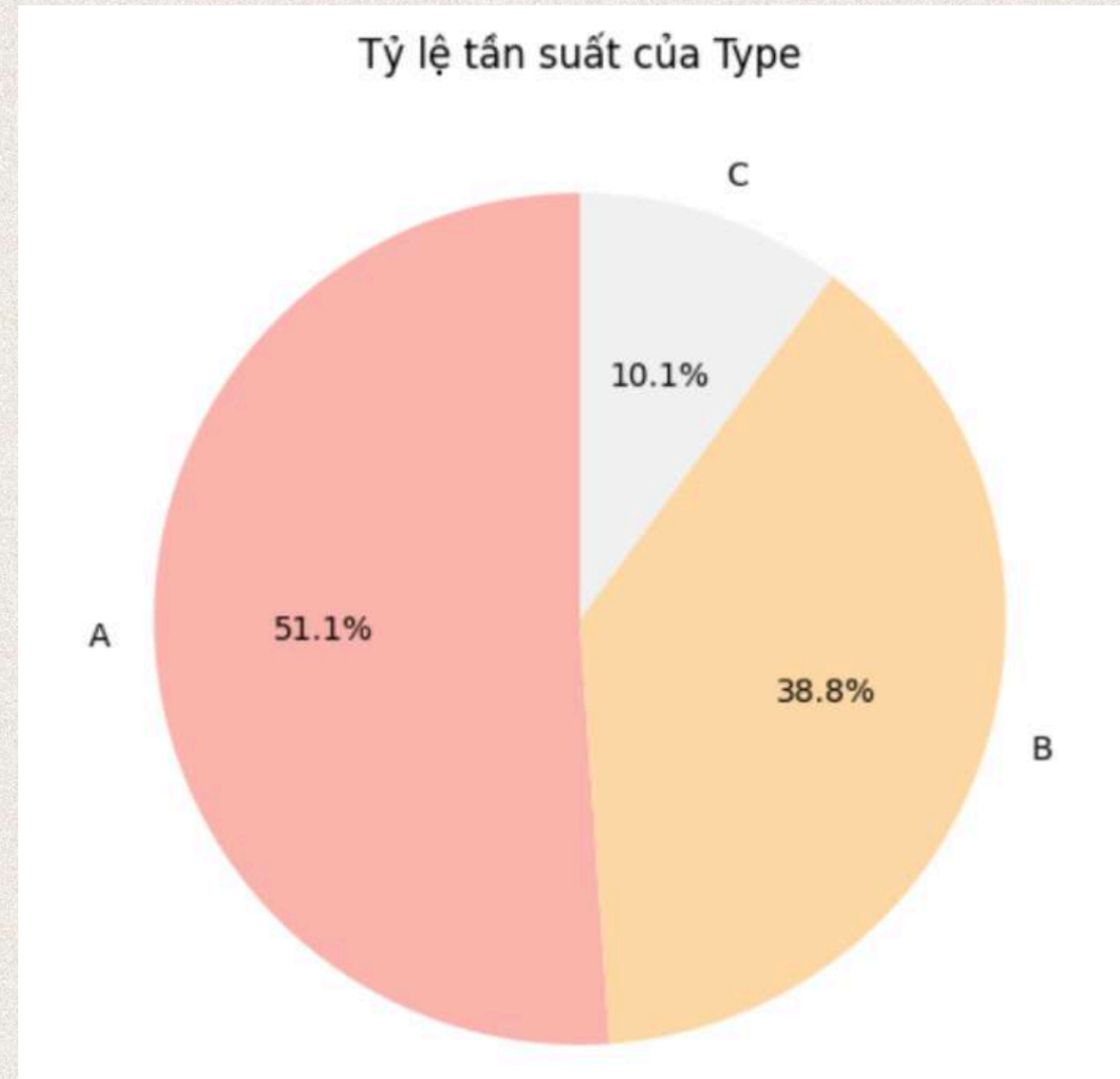


**Biểu đồ phân bố Size trong tập stores:**



# PHƯƠNG PHÁP PHÂN TÍCH

## 1. PHÂN TÍCH ĐƠN BIẾN



**Biểu đồ tần suất của Type**



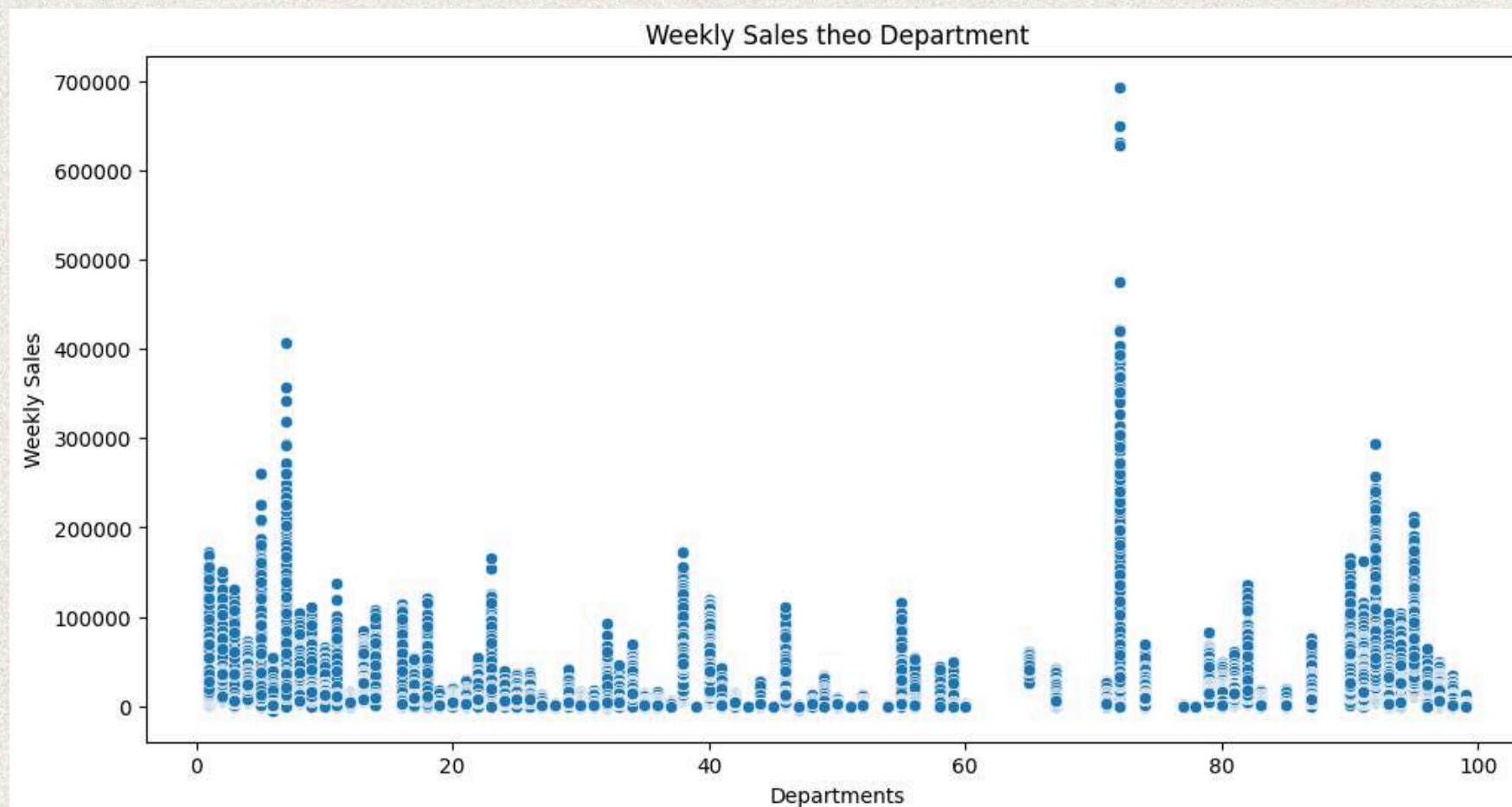
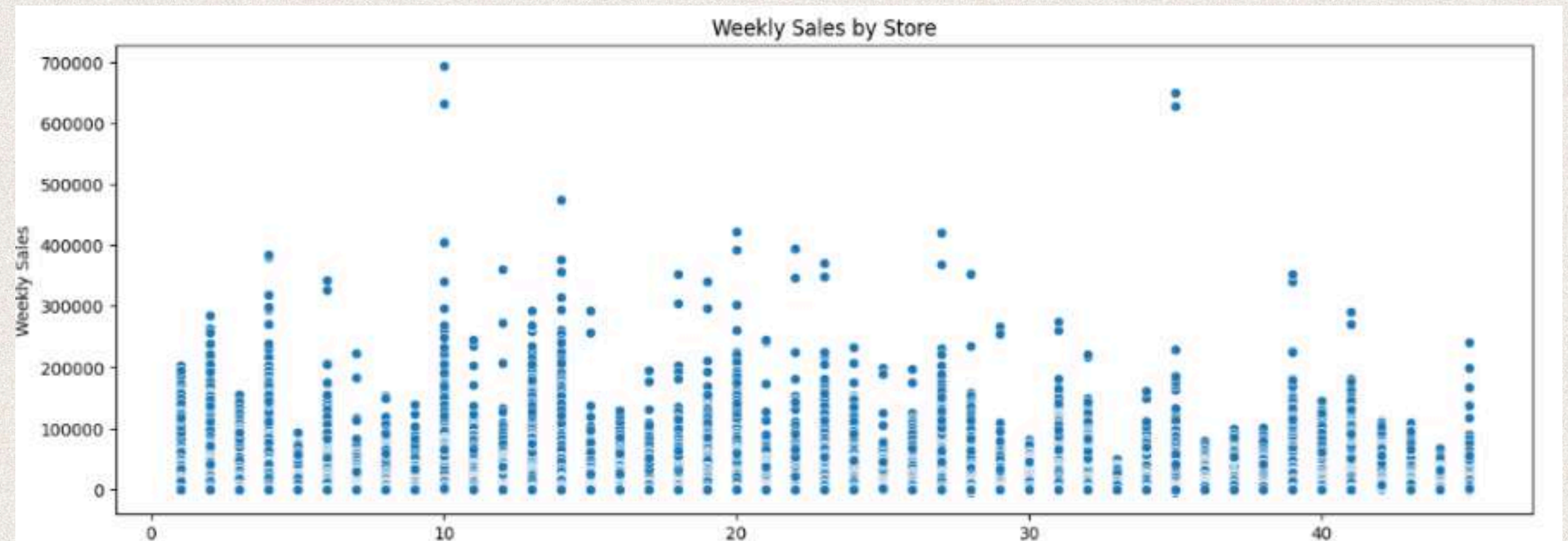
# 2. PHÂN TÍCH TƯƠNG QUAN HAI BIẾN

Phân tích Weekly\_Sales theo từng biến ảnh hưởng:

- Biến phân loại đến biến số:
  - Weekly\_Sales theo Dept, Store, Type, IsHoliday
  - Dùng Barplot, Boxplot, Pivot Table
- Biến thời gian đến biến số
  - Weekly\_Sales theo Month, Year, Week
  - Tạo biến Month, Week, Year từ Date
- Biến số đến biến số:
  - Weekly\_Sales theo Temperature, Fuel\_Price, CPI, Unemployment
  - Dùng biểu đồ scatterplot, lineplot



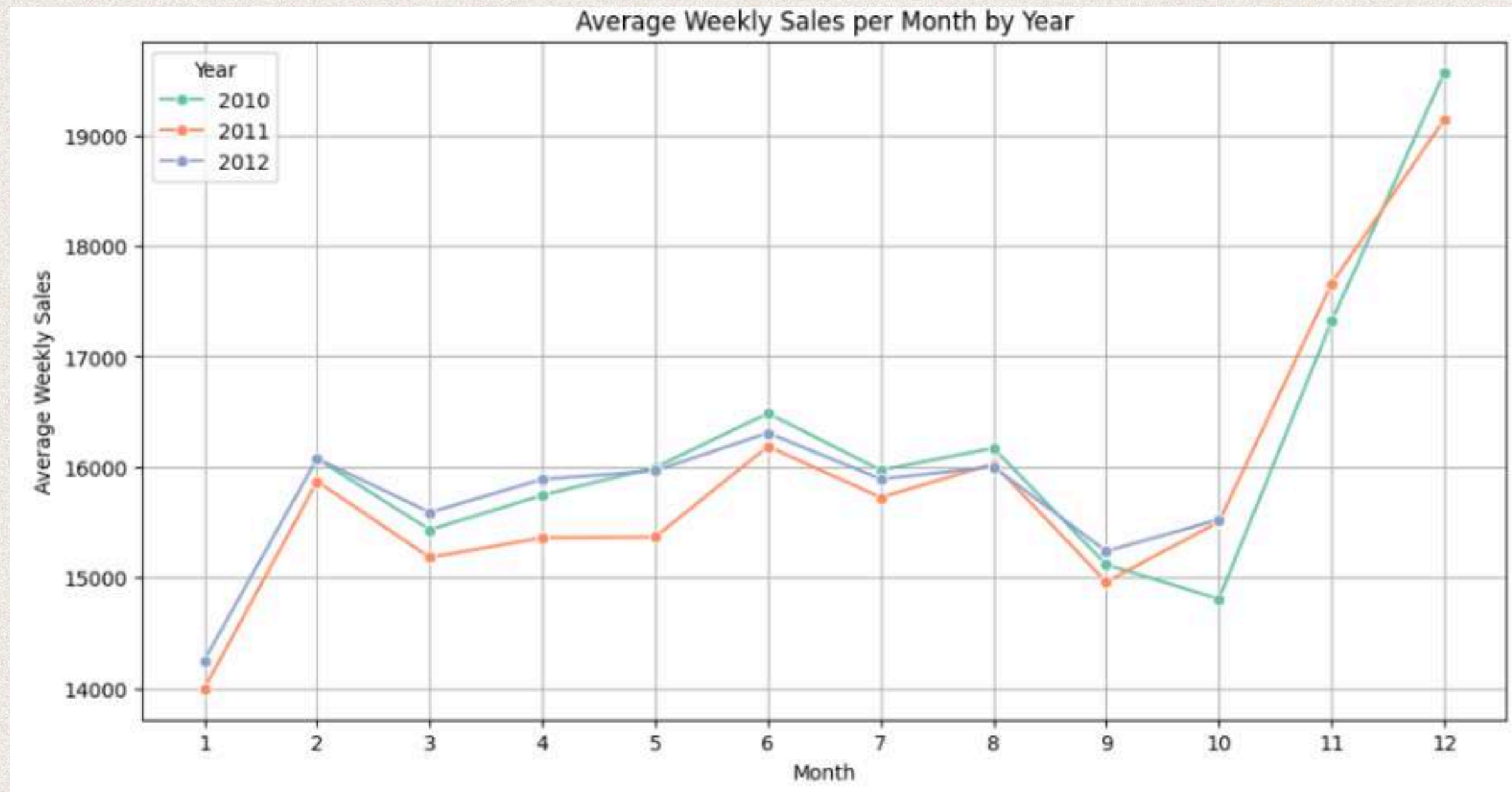
Biểu đồ phân  
bố dữ liệu của  
Weekly\_Sales  
theo Dept và  
Store





# PHƯƠNG PHÁP PHÂN TÍCH

## 2. PHÂN TÍCH TƯƠNG QUAN HAI BIẾN



Biểu đồ phân bố của biến Weekly\_Sales theo tháng trong các năm



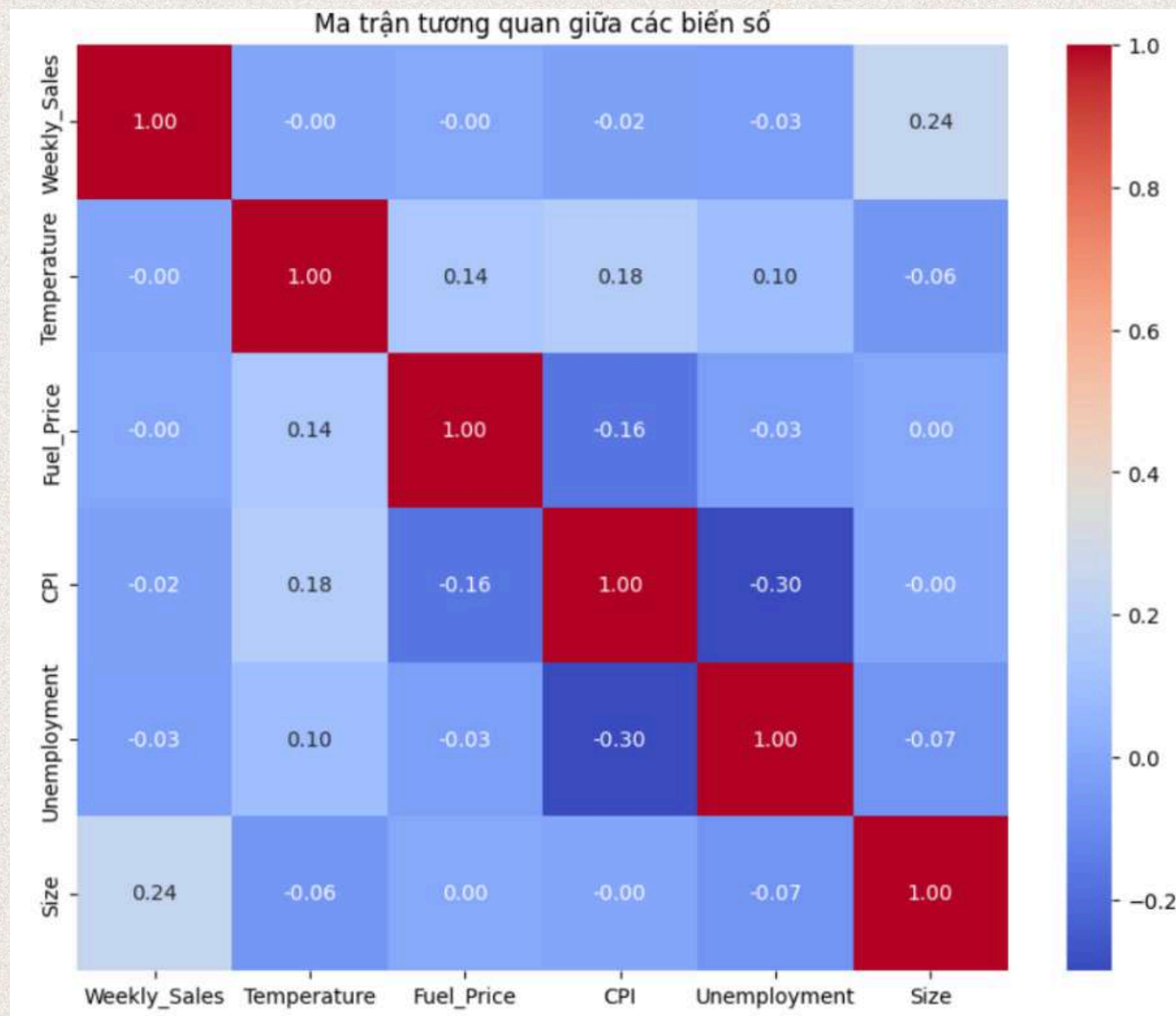
### 3. PHÂN TÍCH ĐA BIẾN

- Dùng `.corr()` và biểu đồ seaborn heatmap
- Mục tiêu: phát hiện mối tương quan tuyến tính giữa các biến:
  - Ví dụ: Size ~ Weekly\_Sales: tương quan yếu dương
  - Unemployment, CPI: tương quan âm với nhau



# PHƯƠNG PHÁP PHÂN TÍCH

## 3. PHÂN TÍCH ĐA BIẾN



Heatmap tương quan giữa các biến số

### Nhận xét

- Biến Size có tương quan dương yếu với Weekly ( $r \approx 0.24$ ), cho thấy quy mô cửa hàng ảnh hưởng phần nào đến doanh thu.
- Các biến như CPI, Unemployment, Fuel\_Price và Temperature gần như không có tương quan rõ rệt với doanh số.
- Hai biến Unemployment và CPI có tương quan âm tương đối với nhau ( $r \approx -0.30$ ).
- Hầu hết các biến số đều không tuyến tính mạnh với doanh số, cho thấy cần mô hình phi tuyến như Random Forest hoặc XGBoost để khai thác mối quan hệ ẩn.



### III. TIỀN XỬ LÝ DỮ LIỆU





01.

### XỬ LÝ GIÁ TRỊ NULL

- Điền NaN bằng 0
- Điền NaN bằng giá trị trung bình hoặc giá trị trước đó

02.

### XỬ LÝ OUTLIERS

- Chỉnh sửa hoặc loại bỏ các outliers để dữ liệu có phân phối gần chuẩn hơn, giảm ảnh hưởng của các giá trị cực đoan.

03.

### MÃ HÓA DỮ LIỆU

- Mã hóa (encode) các biến phân loại



## 1. XỬ LÝ GIÁ TRỊ NULL

	Số lượng	Phần trăm
Store	0	0.000000
Dept	0	0.000000
Date	0	0.000000
Weekly_Sales	0	0.000000
IsHoliday	0	0.000000
Temperature	0	0.000000
Fuel_Price	0	0.000000
MarkDown1	270889	64.257181
MarkDown2	310322	73.611025
MarkDown3	284479	67.480845
MarkDown4	286603	67.984676
MarkDown5	270138	64.079038
CPI	0	0.000000
Unemployment	0	0.000000
Type	0	0.000000
Size	0	0.000000




Chúng ta thấy có 5 cột có giá trị NULL là Markdown 1, Markdown 2, Markdown 3, Markdown 4, Markdown 5, đây là các loại giảm giá khác nhau được áp dụng bởi Walmart

Xử lý các giá trị NULL này bằng cách điền khuyết thành giá trị 0. Điều này giả định rằng nếu không có thông tin về việc giảm giá, thì không có giảm giá

Thống kê số lượng giá trị NULL ở từng cột trong data train



## XỬ LÝ GIÁ TRỊ NULL

	Số lượng	Phần trăm	  
Store	0	0.000000	
Dept	0	0.000000	
Date	0	0.000000	
IsHoliday	0	0.000000	
Temperature	0	0.000000	
Fuel_Price	0	0.000000	
Markdown1	149	0.129493	
Markdown2	28627	24.879198	
Markdown3	9829	8.542203	
Markdown4	12888	11.200723	
Markdown5	0	0.000000	
CPI	38162	33.165890	
Unemployment	38162	33.165890	
Type	0	0.000000	
Size	0	0.000000	

Data test cũng có các cột Markdown 1 → 5 là NULL, ngoài ra còn có thêm các cột CPI, Unemployment

Cách xử lý giá trị NULL:

1. Xử lý các cột Markdown 1-→5 tương tự nha data train, điền 0 vào tất cả giá trị NULL
2. Đối với các cột CPI, Unemployment, điền giá trị mean CPI, Unemployment ở data train

Thống kê số lượng giá trị  
NULL ở từng cột trong data test



## XỬ LÝ GIÁ TRỊ NULL

Ngoài ra, ở cột Weekly\_Sales ở file train có các giá trị  $< 0$

```
Số lượng giá trị Weekly_Sales < 0: 1358  
Tỷ lệ phần trăm: 0.32%
```

Việc có giá trị âm hoặc bằng 0 ở cột giá trị Weekly\_Sales là bất hợp lí. Vì vậy chúng ta sẽ xóa những cột này vì doanh thu không thể âm, nếu bằng 0 thì có thể đó là lỗi nhập liệu. Tỷ lệ phần trăm giá trị Weekly\_Sales âm hoặc bằng 0 chỉ là 0,32%, do đó sẽ không gây ảnh hưởng lớn đến tập dữ liệu



## 2. XỬ LÝ OUTLIER

	Số lượng	Phần trăm
Store	0	0.000000
Dept	0	0.000000
Weekly_Sales	35381	8.418335
Temperature	67	0.015942
Fuel_Price	0	0.000000
Markdown1	56183	13.367834
Markdown2	101695	24.196676
Markdown3	84436	20.090177
Markdown4	78897	18.772262
Markdown5	40334	9.596821
CPI	0	0.000000
Unemployment	32050	7.625778
Size	0	0.000000

Tìm outlier bằng phương pháp IQR: tính hai phần vị Q1 (25%) và Q3 (75%) của tập dữ liệu, sau đó xác định khoảng liên phần vị  $IQR = Q3 - Q1$ , mọi giá trị nằm dưới  $Q1 - 1,5 * IQR$  hoặc trên  $Q3 + 1,5 * IQR$  được xem là outlier

Có các cột Weekly\_Sales, Markdown 1-> 5, Unemployment, Temperature có outlier

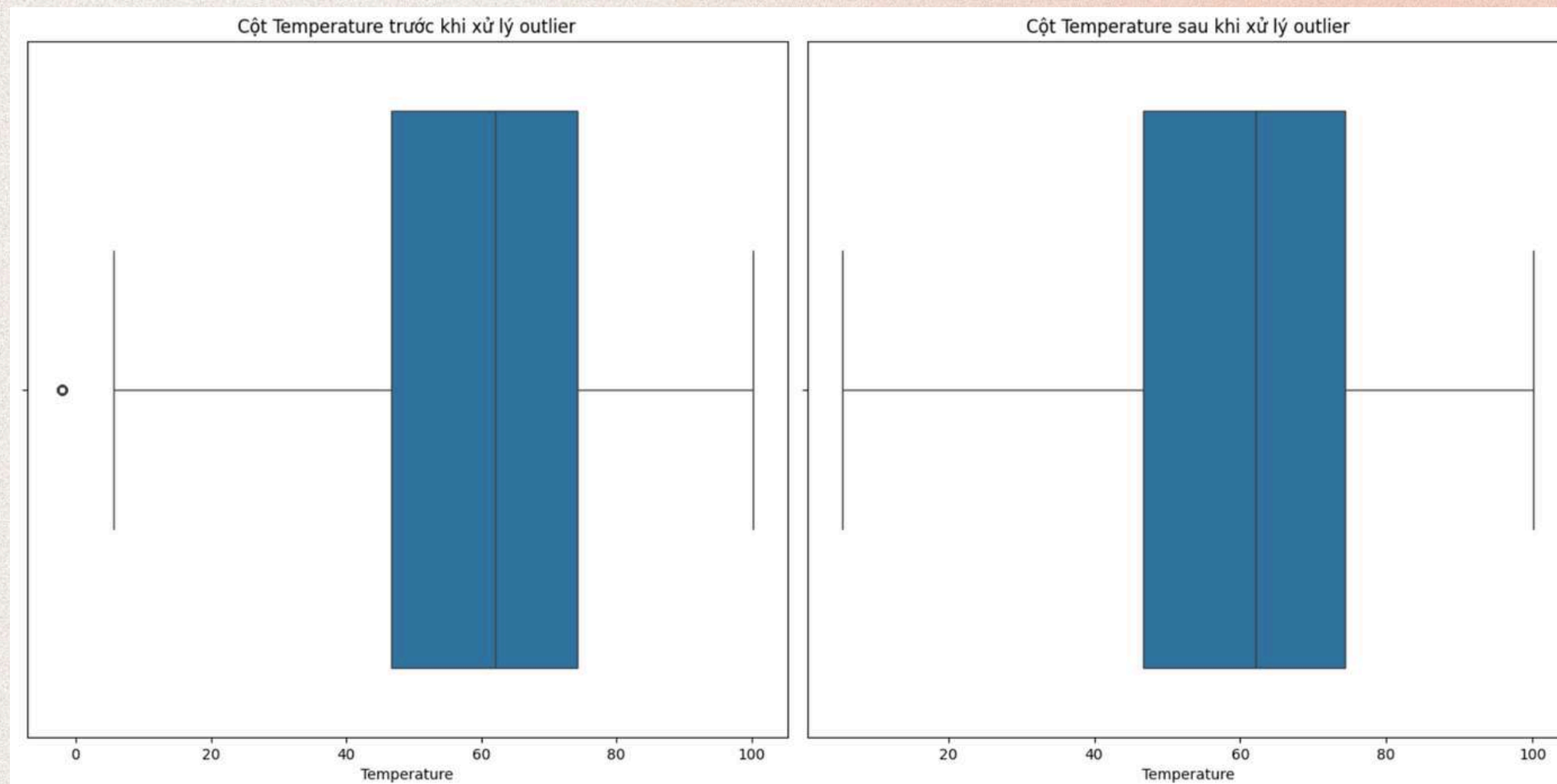
Với các cột Weekly\_Sales, Markdown 1-> 5, mặc dù có outlier, nhưng các mô hình sử dụng trong bài ít ảnh hưởng bởi outlier nên không xử lý

**Thống kê số lượng outlier ở từng cột trong data train**



## XỬ LÝ OUTLIERS

Với cột Temperature, các giá trị outlier chỉ lệch phải, số lượng ít. Xử lý outlier bằng cách thay các giá trị thành mean của cột Temperature

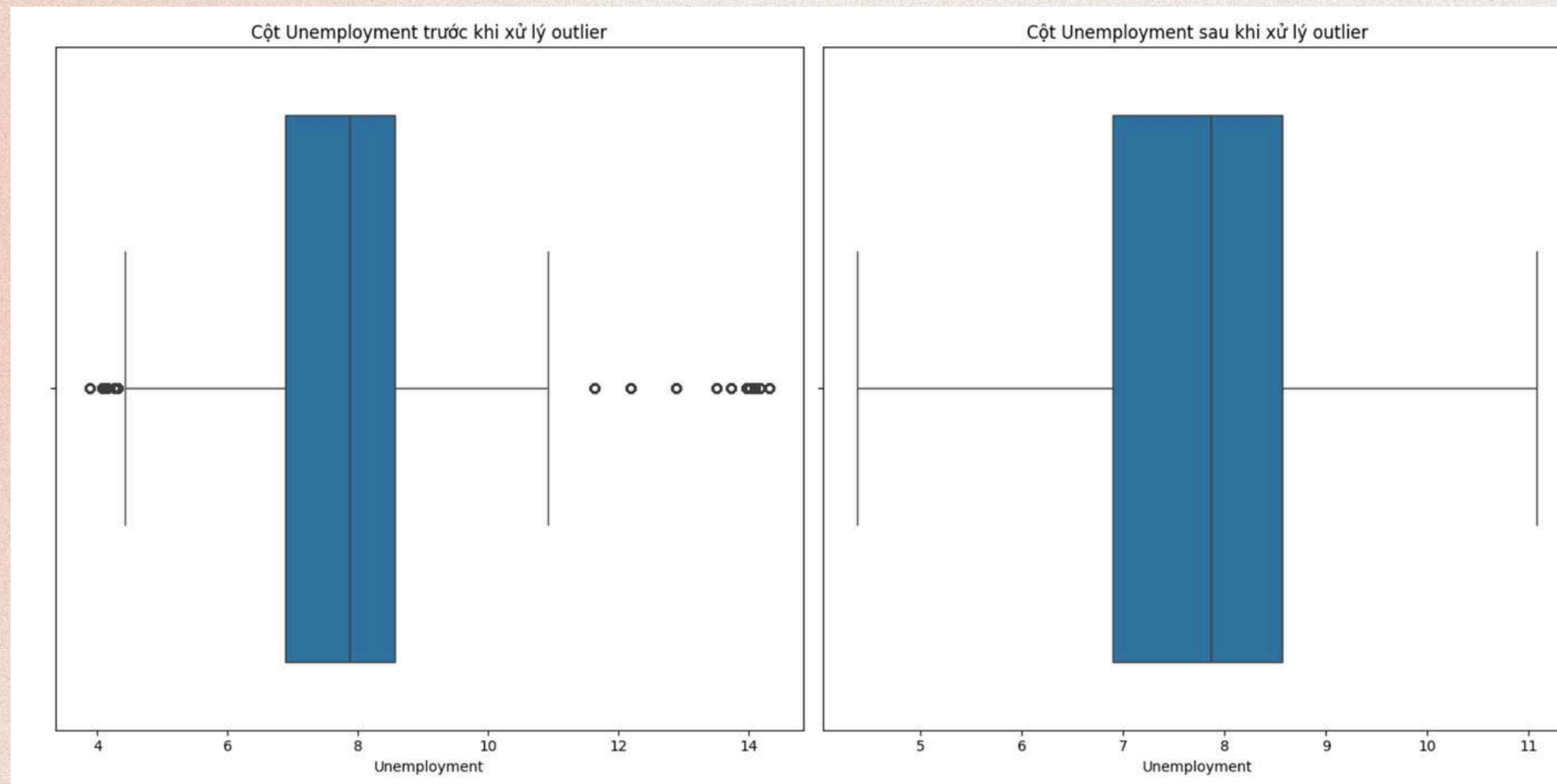


**Giá trị của cột Temperature trước và sau khi xử lý outlier**



## XỬ LÝ OUTLIERS

Với cột Unemployment, các giá trị outlier lệch cao về hai bên có thiên hướng lệch phải. Xử lý outlier bằng cách thay các giá trị quá hai bên thành giá trị biên của IQR



**Giá trị của cột Unemployment trước và sau khi xử lý outlier**



### 3. MÃ HÓA DỮ LIỆU

Trong các cột data test và train, có cột Type gồm các giá trị: A, B, C. Ta cần chuyển các giá trị này thành số để mô hình có thể hiểu được. Chúng ta sẽ sử dụng Label encoding lên các giá trị này, chuyển thành giá trị số 0, 1, 2

Mặc dù có những mô hình hiểu nhầm sự phân cấp giữa các giá trị số dẫn đến làm sai lệch dự đoán nhưng những mô hình trong bài (Random Forest, XGBoost, LightGBM) thì không bị ảnh hưởng

Giá trị ở cột IsHoliday có dạng True/False. Chúng ta chuyển giá trị của cột trong data test và train thành dạng số 0/1 thay vì True/False giúp đảm bảo tính nhất quán, tương thích và hiệu quả trong xử lý và huấn luyện mô hình



## 4. CHUẨN HÓA DỮ LIỆU

Sử dụng Min-Max Scaller cho các cột Markdown 1-> 5. Lí do là vì các cột Markdown 1-5 có rất nhiều giá trị bằng 0 do việc điền NULL từ trước, làm cho giá trị không bị nhiễu bởi phân phối không đối xứng, giúp cho việc so sánh và phân tích dễ dàng hơn bằng cách đưa dữ liệu về khoảng [0, 1]

Sử dụng Standard Scaller cho các cột CPI, Fuel\_Price, Size, Temperature, Unemployment. Lí do là vì các cột này có phân phối đa dạng, Standard Scaller đưa các feature về cùng một chuẩn với trung bình bằng 0 và độ lệch chuẩn bằng 1, làm đồng bộ hóa và bảo đảm công bằng giữa các feature

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	Markdown1	Markdown2	Markdown3	Markdown4	Markdown5	CPI	Unemployment	Type	Size	Day	Month	Year
0	1	1	2010-02-05	24924.50	0	-0.965231	-1.720502	0.0	0.002536	0.000205	0.0	0.0	1.018435	0.183839	0	0.238806	5	2	2010
1	1	1	2010-02-12	46039.49	1	-1.171399	-1.772844	0.0	0.002536	0.000205	0.0	0.0	1.022159	0.183839	0	0.238806	12	2	2010
2	1	1	2010-02-19	41595.55	0	-1.094357	-1.846995	0.0	0.002536	0.000205	0.0	0.0	1.023358	0.183839	0	0.238806	19	2	2010
3	1	1	2010-02-26	19403.54	0	-0.730850	-1.744492	0.0	0.002536	0.000205	0.0	0.0	1.024137	0.183839	0	0.238806	26	2	2010
4	1	1	2010-03-05	21827.90	0	-0.737903	-1.604913	0.0	0.002536	0.000205	0.0	0.0	1.024916	0.183839	0	0.238806	5	3	2010

Một số giá trị đầu tiên sau khi tiền xử lý dữ liệu



## **IV. FEATURE ENGINEERING**





## FEATURE ENGINEERING

01.

**LOẠI BỎ ĐẶC TRƯNG KHÔNG CÓ NHIỀU GIÁ TRỊ DỰ ĐOÁN**

02.

**TẠO MỚI ĐẶC TRƯNG**

02.

**CHỌN LỰA ĐẶC TRƯNG**



## FEATURE ENGINEERING

# 1. LOẠI BỎ ĐẶC TRƯNG KHÔNG CÓ NHIỀU GIÁ TRỊ DỰ ĐOÁN

	Cột	Số lượng giá trị riêng biệt	Tỉ lệ
0	Store	45	0.000107
1	Dept	81	0.000193
2	Date	143	0.000340
3	Weekly_Sales	358786	0.853673
4	IsHoliday	2	0.000005
5	Temperature	3527	0.008392
6	Fuel_Price	892	0.002122
7	MarkDown1	2278	0.005420
8	MarkDown2	1499	0.003567
9	MarkDown3	1662	0.003954
10	MarkDown4	1945	0.004628
11	MarkDown5	2294	0.005458
12	CPI	2145	0.005104
13	Unemployment	334	0.000795
14	Type	3	0.000007
15	Size	40	0.000095
16	Day	31	0.000074
17	Month	12	0.000029
18	Year	3	0.000007

Chúng ta thấy trong dữ liệu có các cột Store, Dept có thể là cột định danh vì các giá trị này ghi thứ tự cửa hàng, phòng ban đang tính toán. Nhưng sau khi kiểm tra chúng ta thấy Store chỉ có 45 giá trị riêng biệt, Dept có 81 giá trị riêng biệt. Vì vậy không nên xóa các cột này vì tỉ lệ giá trị riêng biệt trên toàn bộ dữ liệu quá thấp

Trong dữ liệu cũng không có cột nào chỉ nhận một giá trị nên chúng ta cũng sẽ không xóa các cột khác

**Thống kê số lượng giá trị riêng biệt ở từng cột trong data train**



## 2. TẠO MỚI ĐẶC TRƯNG

### a) Đặc trưng tổng hợp từ các đặc trưng cũ

Chúng ta cũng cần tách cột Date thành 3 cột mới: Day, Month, Year và thêm đặc trưng Week là tuần trong năm vào dữ liệu. Điều này giúp mô hình nhìn ra mối quan hệ danh số theo các đơn vị thời gian trong năm, làm dự đoán chuẩn xác hơn

Trong dữ liệu đã cho, có 5 loại giảm giá khác nhau nhưng không có mối quan hệ rõ ràng với biến cần dự đoán. Vì vậy chúng ta sẽ tạo đặc trưng mới Total\_Markdown là tổng tất cả giảm giá của Markdown từ 1 - 5 để biết mức độ giảm giá tổng thể

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	Markdown1	Markdown2	Markdown3	...	Markdown5	CPI	Unemployment	Type	Size	Day	Month	Year	Total_Markdown	Week
0	1	1	2010-02-05	24924.50	0	-0.965231	-1.720502	0.0	0.002536	0.000205	...	0.0	1.018435	0.183839	0	0.238806	5	2	2010	0.002742	5
1	1	1	2010-02-12	46039.49	1	-1.171399	-1.772844	0.0	0.002536	0.000205	...	0.0	1.022159	0.183839	0	0.238806	12	2	2010	0.002742	6
2	1	1	2010-02-19	41595.55	0	-1.094357	-1.846995	0.0	0.002536	0.000205	...	0.0	1.023358	0.183839	0	0.238806	19	2	2010	0.002742	7
3	1	1	2010-02-26	19403.54	0	-0.730850	-1.744492	0.0	0.002536	0.000205	...	0.0	1.024137	0.183839	0	0.238806	26	2	2010	0.002742	8
4	1	1	2010-03-05	21827.90	0	-0.737903	-1.604913	0.0	0.002536	0.000205	...	0.0	1.024916	0.183839	0	0.238806	5	3	2010	0.002742	9

5 rows x 21 columns



# FEATURE ENGINEERING

## b) Đặc trưng dựa theo thời gian

Chúng ta tạo đặc trưng mới đánh dấu các tuần lễ hội trong năm vì theo như phân tích ở phần trước, doanh thu vào thời điểm lễ thường có mức tiêu thụ đột biến do khuyến mãi, mua sắm quà tặng, nhu cầu tăng cao, giúp mô hình không cần phải tự học chuỗi thời gian rời rạc, cải thiện khả năng dự báo các đỉnh doanh số

Chúng ta sẽ tạo thêm biến 'SuperBowlWeek', 'LaborDay', 'Thanksgiving', 'Christmas' trả về giá trị 1 nếu tương ứng tuần thứ 6, 36, 47, 52 trong năm, 0 với các trường hợp còn lại. Đây là các ngày lễ lớn trong năm có doanh thu tăng đột biến

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	...	Size	Day	Month	Year	Total_Markdown	Week	SuperBowlWeek	LaborDay	Thanksgiving	Christmas
0	1	1	2010-02-05	24924.50	0	-0.965231	-1.720502	0.0	0.002536	0.000205	...	0.238806	5	2	2010	0.002742	5	0	0	0	0
1	1	1	2010-02-12	46039.49	1	-1.171399	-1.772844	0.0	0.002536	0.000205	...	0.238806	12	2	2010	0.002742	6	1	0	0	0
2	1	1	2010-02-19	41595.55	0	-1.094357	-1.846995	0.0	0.002536	0.000205	...	0.238806	19	2	2010	0.002742	7	0	0	0	0
3	1	1	2010-02-26	19403.54	0	-0.730850	-1.744492	0.0	0.002536	0.000205	...	0.238806	26	2	2010	0.002742	8	0	0	0	0
4	1	1	2010-03-05	21827.90	0	-0.737903	-1.604913	0.0	0.002536	0.000205	...	0.238806	5	3	2010	0.002742	9	0	0	0	0



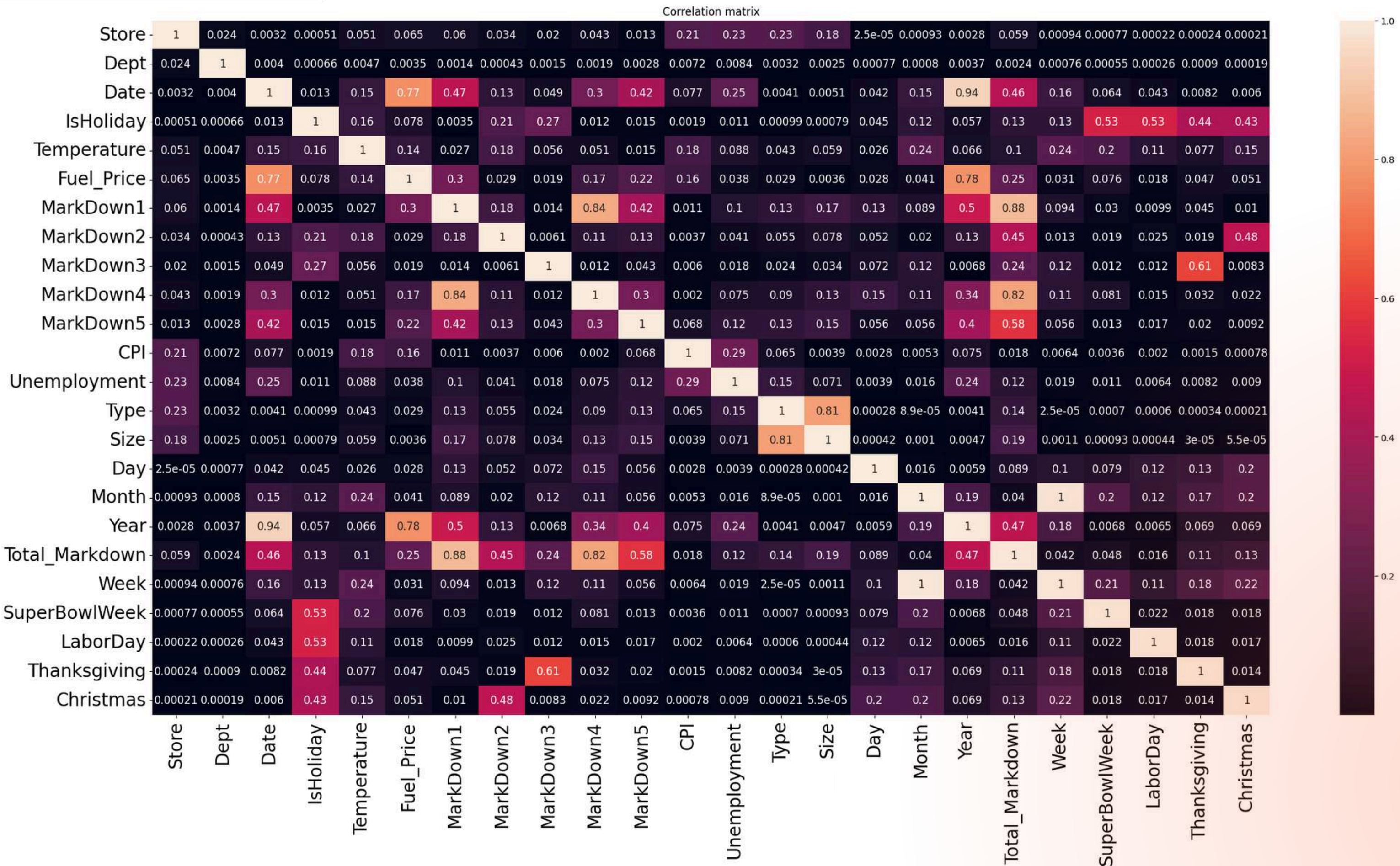
### 3. CHỌN LỰA ĐẶC TRƯNG

#### a) Sử dụng ma trận tương quan

Chúng ta sẽ vẽ ma trận tương quan giữa các feature không phải biến cần dự đoán. Mục đích là loại các feature có mức độ đa cộng tuyến cao, những feature có giá trị lớn hơn một ngưỡng nhất định, chỉ giữ lại các biến có ít tương quan với nhau



# LỰA CHỌN ĐẶC TRƯNG





## LỰA CHỌN ĐẶC TRƯNG

Chúng ta có các nhận xét sau:

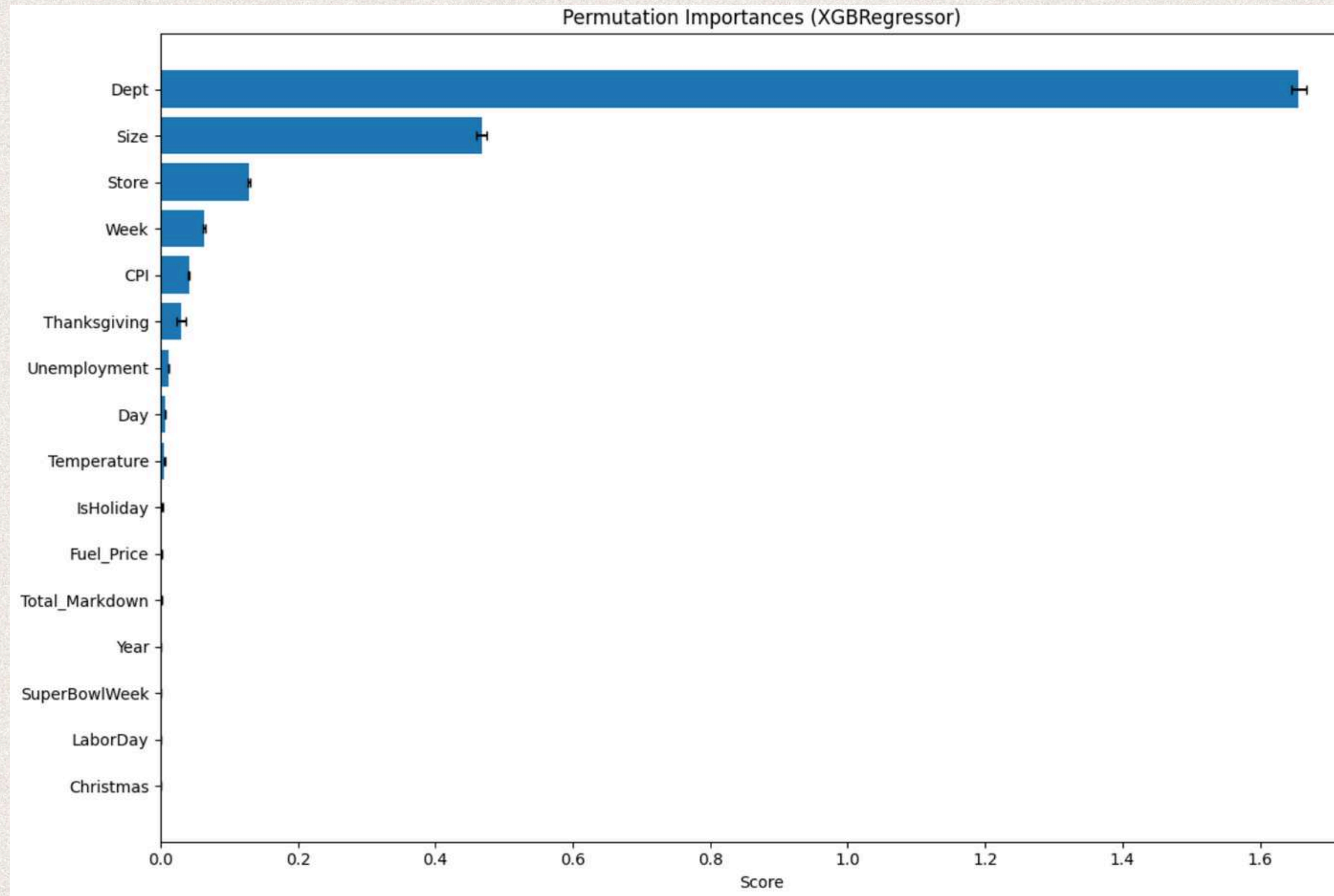
- Hai cột Week và Month có độ tương quan cao do thông tin của Week cũng có trong Month, vì vậy chúng ta sẽ bỏ cột Month đi
- Tương tự với Size và Type, ở đây chúng ta bỏ cột Type
- Tương tự với Year và Date, ở đây chúng ta bỏ cột Date
- Các biến Markdown1, Markdown2, Markdown3, Markdown4, Markdown5 có tương quan khá lớn với một số feature khác nên chúng ta cũng bỏ các cột này đi

## b) Sử dụng model-based

Sau đó để hiểu hơn về độ quan trọng của các feature, chúng ta sẽ thử train trên một model XGBoost, tuy nhiên chúng ta sẽ thử xáo trộn các giá trị trong feature khi train 10 lần, xem sai số dự đoán được tăng nhiều hay ít. Từ đó sẽ biết được sự đóng góp của từng feature vào tính chính xác mô hình



# LỰA CHỌN ĐẶC TRƯNG





## LỰA CHỌN ĐẶC TRƯNG

Chúng ta có thể thấy rằng 3 features: Dept, Size, Store, Week có ảnh hưởng lớn đến mô hình. Có nghĩa là doanh thu của một cửa hàng phụ thuộc lớn vào tên cửa hàng, quy mô cửa hàng, các lĩnh vực cửa hàng nhắm tới, số tuần hiện tại trong năm. Một số features cũng có ảnh hưởng đến doanh thu cửa hàng nhưng không đáng kể



# V.XÂY DỰNG MÔ HÌNH





# 1. Sử dụng các mô hình cơ bản

Chúng ta sẽ sử dụng các mô hình sau

- Random Forest
- XGBoost
- LightGBM

Các thang đo sử dụng:

- MSE: Tính trung bình sai số trong quá trình dự đoán
- MAE: Phạt nặng các dự đoán có độ lệch cao so với giá trị thật
- RMSE: Tương tự MSE
- WMAE(thang đo của competition): Phạt nặng các dự đoán có sai số trong các tuần nghỉ lễ hơn gấp 5 lần so với các dự đoán có sai số trong các tuần không phải tuần nghỉ lễ



SỬ DỤNG MÔ HÌNH CƠ BẢN

Mô hình	MSE	MAE	RMSE	WMAE
Random Forest	8667219.43	1234.31	2944.01	1370.91
XGBoost	26150054.35	2954.91	5113.71	3081.07
LightGBM	46454371.52	4153.51	6815.74	4248.17

So sánh các mô hình khi chưa tunning

Chúng ta thấy hai mô hình Random Forest, XGBoost cho kết quả khá tốt khi chạy thử, nên chúng ta sẽ tunning bộ siêu tham số tốt nhất cho hai mô hình này để cải thiện kết quả



## 2. Tìm bộ siêu tham số tốt nhất cho mô hình

Chúng ta sẽ tìm bộ siêu tham số tốt nhất cho các mô hình bằng Validation Curves, xem sai số khi thay đổi từng siêu tham số trong lúc train và test, chúng ta sẽ chọn giá trị sao cho sai số sao cho cả lúc train và test thấp nhất. Phương pháp này giúp tránh overfitting, tìm được khoảng tốt nhất của một siêu tham số. Nhược điểm là giá trị tìm được khi kết hợp chưa chắc là bộ siêu tham số tốt nhất

### a) Random Forest

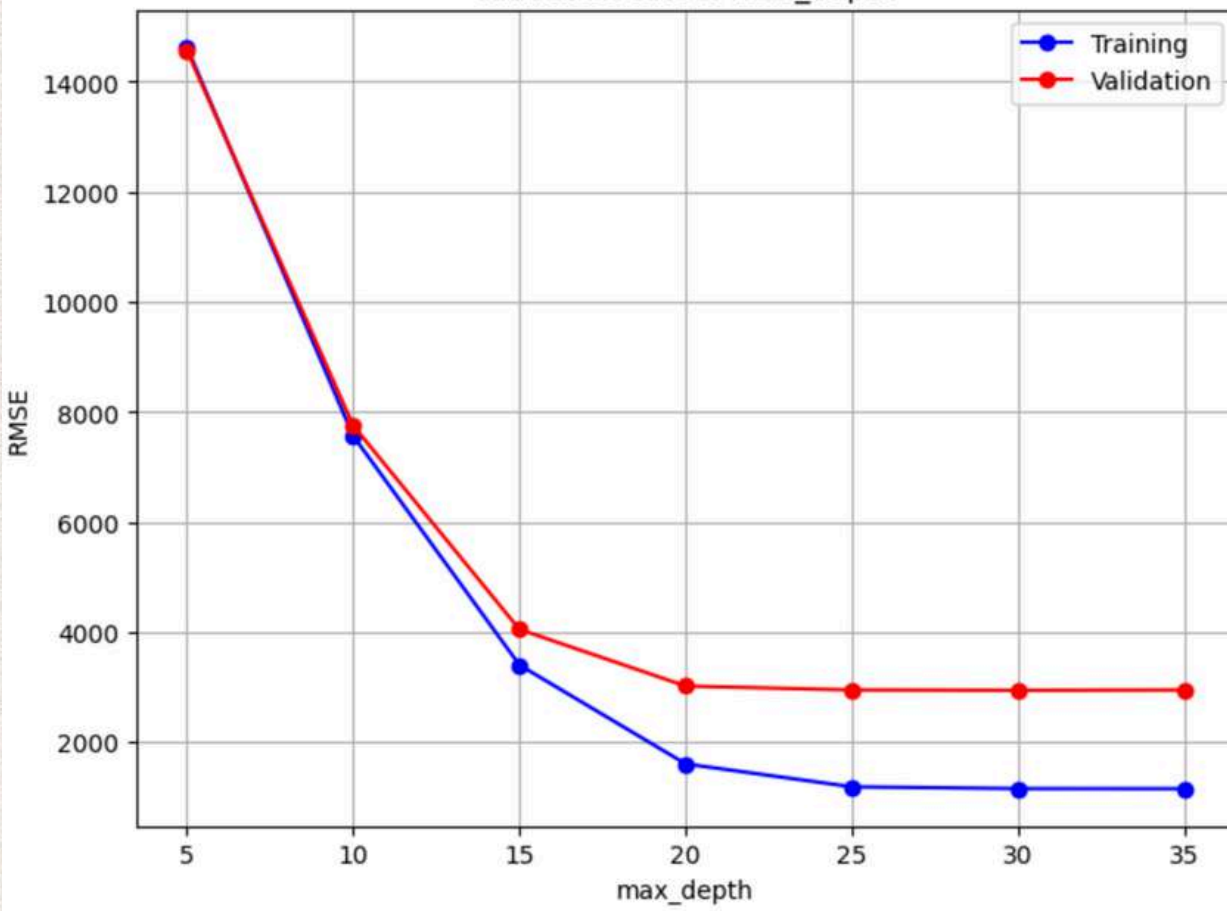
Chúng ta sẽ tuning các siêu tham số sau:

- `n_estimators`: số lượng cây trong rừng
- `max_depth`: độ sâu tối đa của mỗi cây
- `min_samples_split`: số mẫu tối thiểu tại một nút để thực hiện phép tách.
- `min_samples_leaf`: số mẫu tối thiểu tại mỗi nút lá, đảm bảo lá không quá nhỏ.
- `max_features`: số thuộc tính ngẫu nhiên được chọn tại mỗi nút để đánh giá phép tách.
- `max_samples`: khi xây dựng mỗi cây, ta chọn ngẫu nhiên và có hoàn lại khoảng bao nhiêu % mẫu từ tập dữ liệu gốc

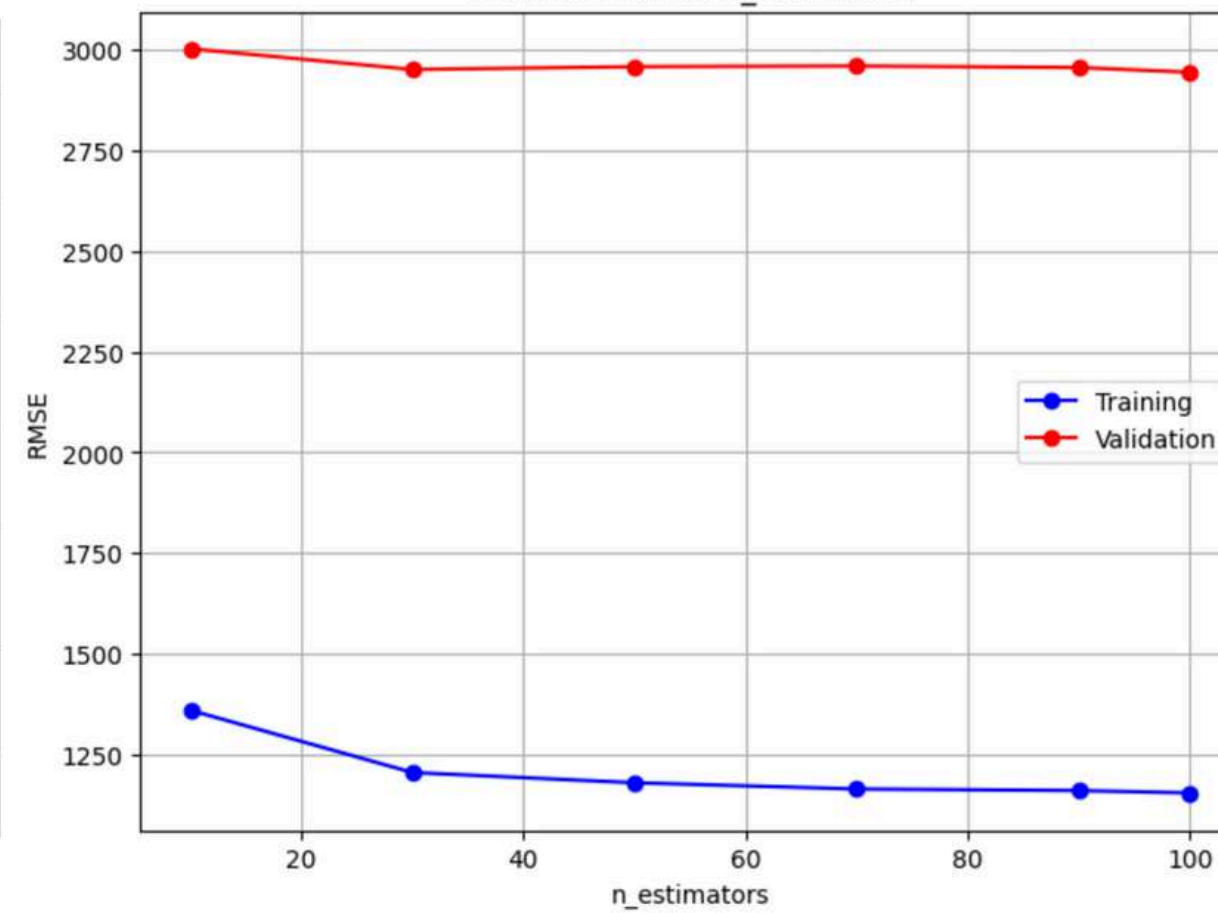


# TÌM BỘ SIÊU THAM SỐ TỐT NHẤT

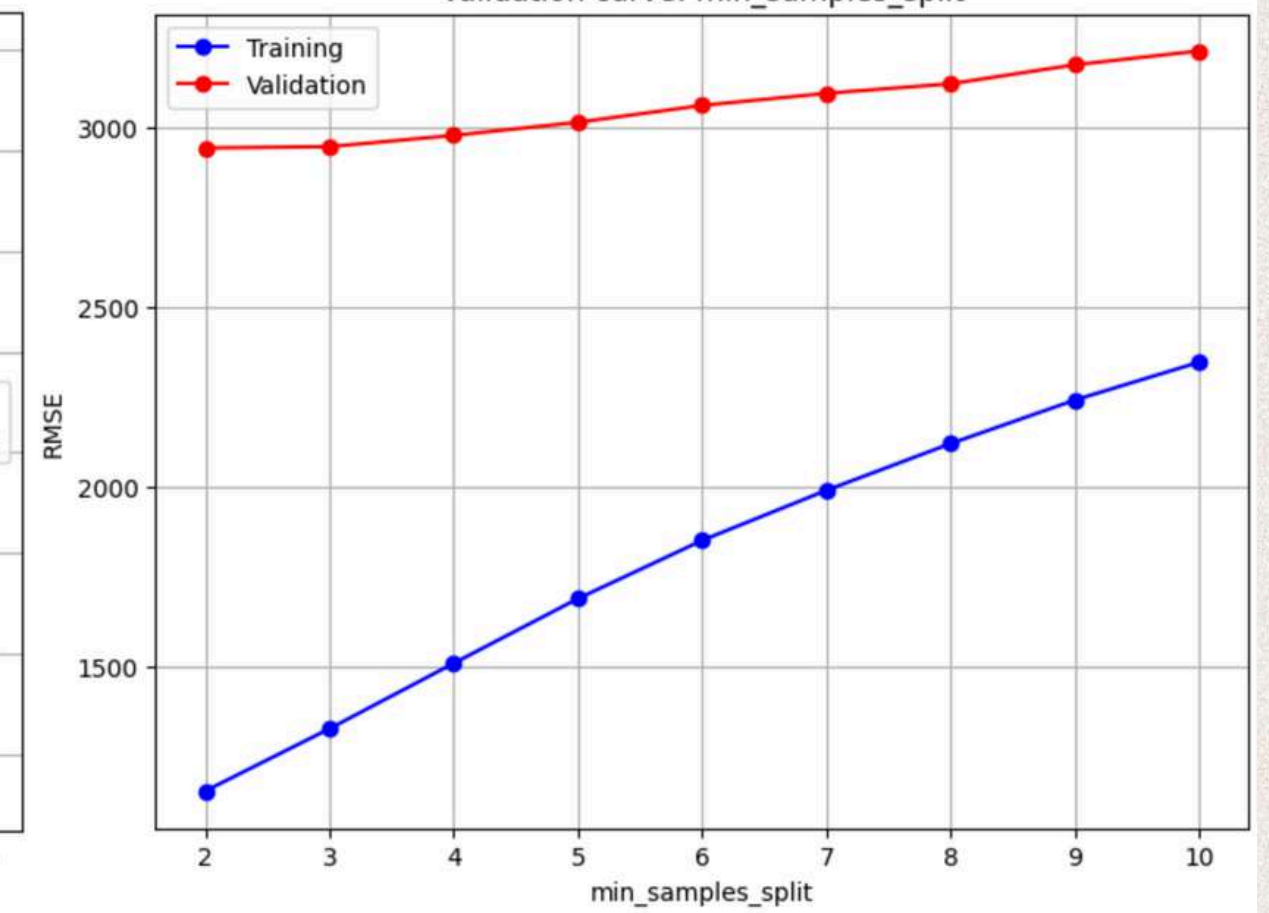
Validation curve: max\_depth



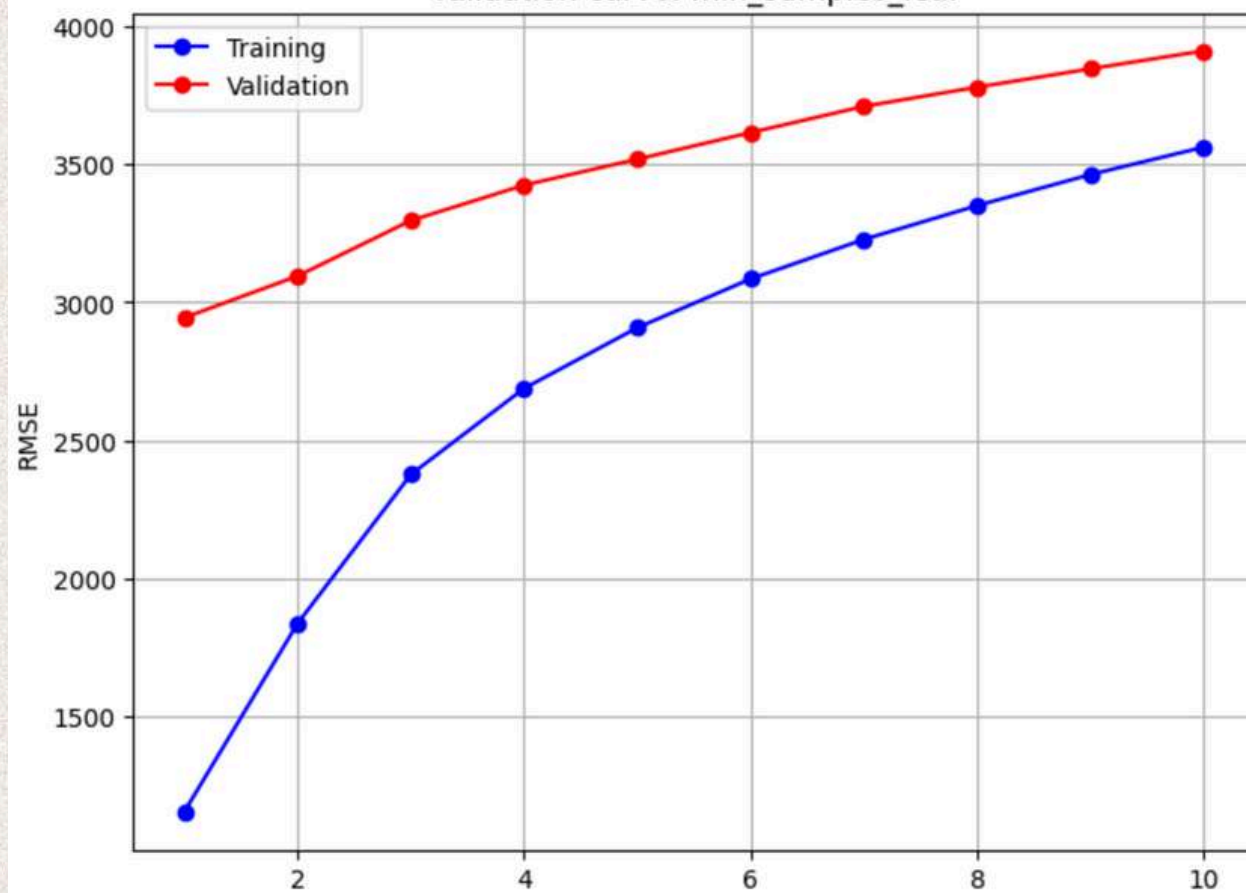
Validation curve: n\_estimators



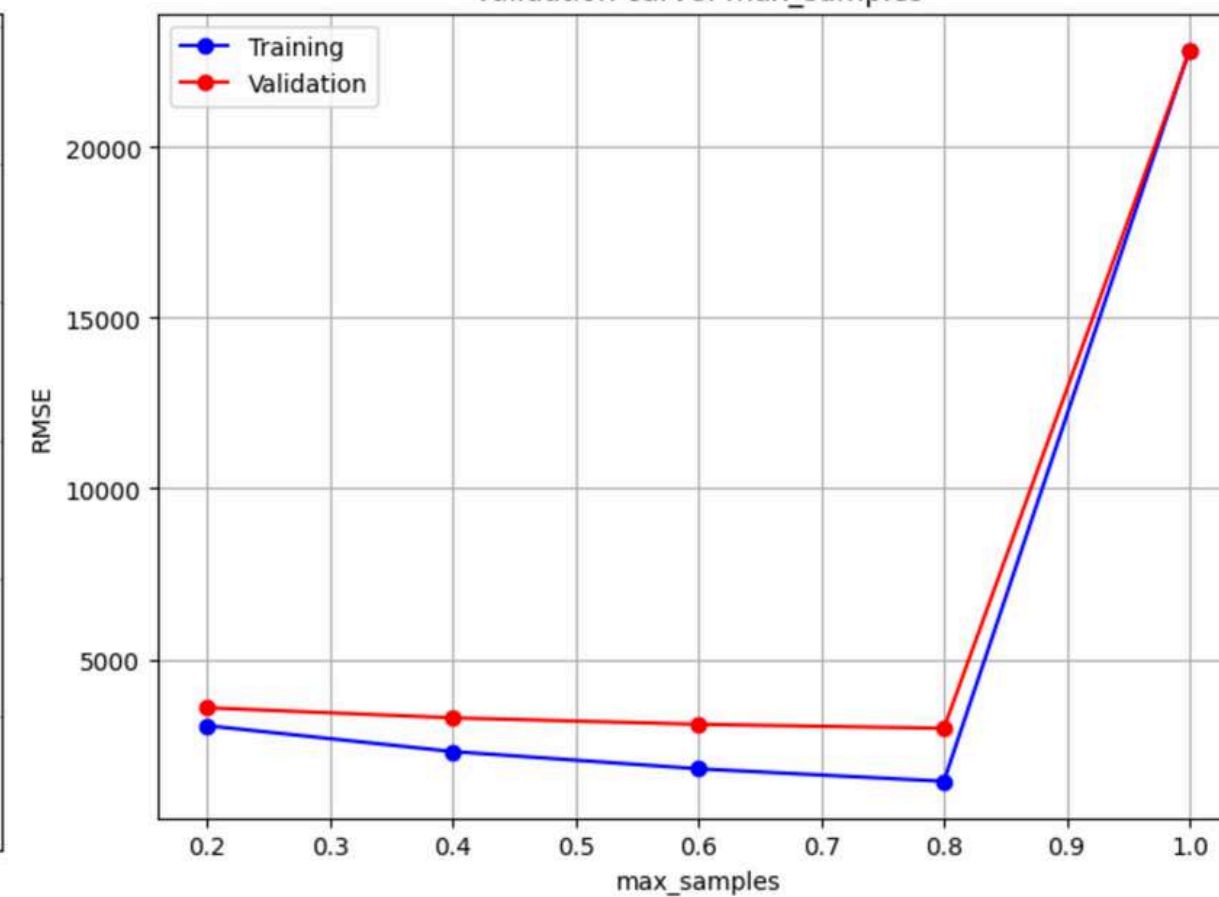
Validation curve: min\_samples\_split



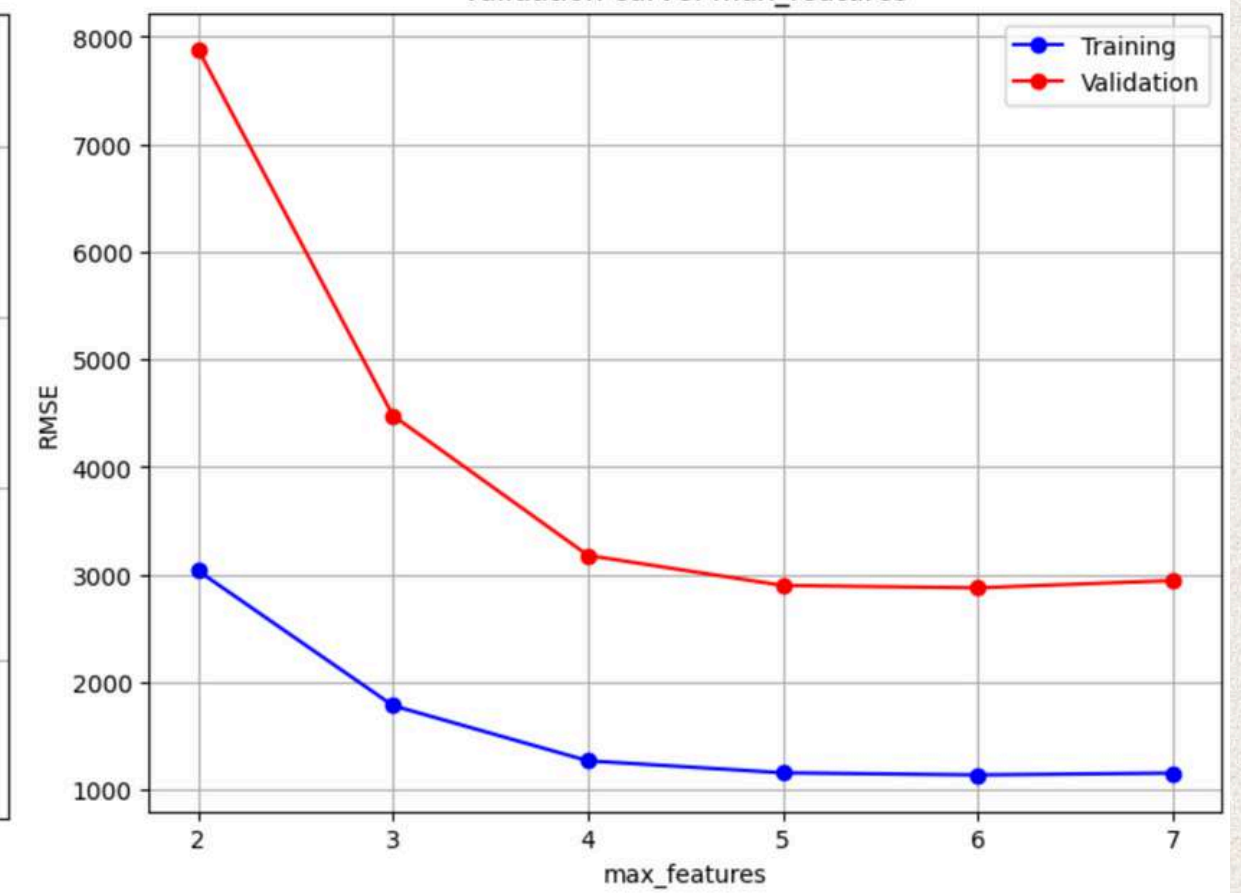
Validation curve: min\_samples\_leaf



Validation curve: max\_samples



Validation curve: max\_features





## TÌM BỘ SIÊU THAM SỐ TỐT NHẤT

Qua các biểu đồ trên, chúng ta thấy bộ siêu tham số tốt nhất là:

- max\_depth: 30
- n\_estimators: 100
- min\_samples\_split: 2
- min\_samples\_leaf: 1
- max\_samples: 0.8
- max\_features: 7

Ngoài ra bằng cách thử nghiệm các bộ siêu tham số khác, đã tìm ra một bộ siêu tham số cho kết quả tốt hơn như sau:

- max\_depth: 30
- n\_estimators: 130
- min\_samples\_split: 2
- min\_samples\_leaf: 1
- max\_samples: 0.9999
- max\_features: 6

Điều này xảy ra là vì các siêu tham số tương tác lẫn nhau, giá trị tốt nhất khi xét riêng lẻ không đảm bảo tạo ra tổ hợp tốt nhất cho mô hình



# TÌM BỘ SIÊU THAM SỐ TỐT NHẤT

Kết quả của mô hình sau khi tunning bộ siêu tham số như sau

Mô hình	MSE	MAE	RMSE	WMAE
Random Forest	8667219.43	1234.31	2944.01	1370.91
Random Forest (hyperparameter tuning)	8449648.91	1220.08	2906.82	1352.00

So sánh với mô hình ban đầu, đã có cải thiện hơn so với mô hình ban đầu, tuy nhiên không đáng kể. Mô hình Random Forest mặc định đã có khả năng dự đoán chính xác cao gần bằng so với mô hình đã chỉnh siêu tham số tối ưu

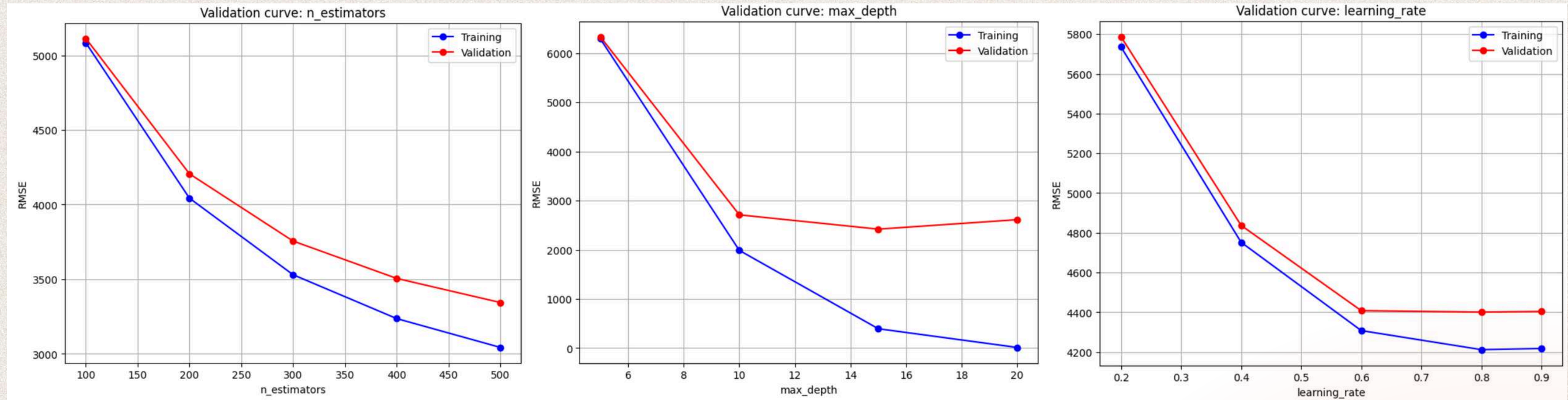
## b) XGBoost

Chúng ta tối ưu các tham số sau:

- n\_estimators : thêm cây vào mô hình sau bao nhiêu vòng lặp
- max\_depth: độ sâu tối đa của mỗi cây
- learning\_rate: mức đóng góp của mỗi cây mới vào dự đoán chung



# TÌM BỘ SIÊU THAM SỐ TỐT NHẤT



Qua các biểu đồ trên và thử nghiệm trên các bộ tham số khác nhau, tìm được bộ tham số tốt nhất là:

- $n\_estimators = 500$
- $max\_depth = 20$
- $learning\_rate = 0.8$



## TÌM BỘ SIÊU THAM SỐ TỐT NHẤT

Kết quả của mô hình sau khi tuning bộ siêu tham số như sau

Mô hình	MSE	MAE	RMSE	WMAE
XGBoost	26150054.35	2954.91	5113.71	3081.07
XGBoost (hyperparameter tuning)	8448900.79	1368.28	2906.69	1470.96

Sau khi chọn bộ siêu tham số, sai số của mô hình XGBoost đã cải thiện rất nhiều so với mô hình ban đầu, điều này là vì XGBoost có rất nhiều siêu tham số ảnh hưởng trực tiếp đến khả năng học và khái quát hoá của mô hình. Sau khi chọn được bộ siêu tham số phù hợp với dữ liệu, kết quả đã cải thiện

### 3) Kết hợp mô hình

Sau khi tìm ra bộ siêu tham số tốt nhất cho mô hình Random Forest và XGBoost, chúng ta sẽ thử kết hợp kết quả của hai mô hình. XGBoost có khả năng tìm ra các quan hệ phức tạp trong dữ liệu rất tốt, trong khi Random Forest ít bị ảnh hưởng bởi dữ liệu nhiễu. Việc kết hợp giúp mô hình vừa ổn định trước nhiễu, vừa học được các mẫu phức tạp, tăng tính chính xác và khả năng lệch giá trị so với việc chỉ dùng riêng mỗi mô hình



# KẾT HỢP MÔ HÌNH

Chúng ta kết hợp bằng Weight Average theo tỉ lệ 60 : 40 cho Random Forest và XGBoost

Sai số khi train mô hình so sánh với khi chỉ dùng một mô hình tốt hơn đáng kể

Mô hình	MSE	MAE	RMSE	WMAE
Random Forest (hyperparameter tuning)	8449648.91	1220.08	2906.82	1352.00
XGBoost (hyperparameter tuning)	8448900.79	1368.28	2906.69	1470.96
Weighted Average (RF + XGBoost)	7063884.80	1173.90	2657.79	1282.54



# XÂY DỰNG MÔ HÌNH

Kết quả của tất cả các mô hình

Mô hình	MSE	MAE	RMSE	WMAE
LightGBM	46454371.52	4153.51	6815.74	4248.17
XGBoost	26150054.35	2954.91	5113.71	3081.07
Random Forest	8667219.43	1234.31	2944.01	1370.91
XGBoost (hyperparameter tuning)	8448900.79	1368.28	2906.69	1470.96
Random Forest (hyperparameter tuning)	8449648.91	1220.08	2906.82	1352.00
Weighted Average (RF + XGBoost)	7063884.80	1173.90	2657.79	1282.54



THANK  
YOU