

Đại học quốc gia TP.HCM
Trường đại học Công nghệ Thông tin



Ngành Khoa học máy tính

Môn học: Lập trình Python cho máy học – CS116.P22

Walmart Sales Forecasting

Sinh viên:

23520520 – Lê Chí Hoàng
23520526 – Ngô Lê Nhật Hoàng
23520937 – Nguyễn Hoàng Minh
23521825 – Đặng Văn Vy

Giảng viên:

TS. Nguyễn Vinh Tiệp
KS. Đàm Vũ Trọng Tài

Lời mở đầu

Tài liệu này trình bày kết quả đồ án môn học CS116 – Lập trình Python cho Máy học, với đề tài là về Walmart Sales Forecasting, bao gồm các yêu cầu, thiết kế, triển khai và đánh giá kết quả. Mục tiêu của đồ án là áp dụng các kiến thức đã học để xây dựng model, đồng thời rèn luyện kỹ năng làm việc nhóm, báo cáo và trình bày kết quả một cách bài bản.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Vinh Tiệp, anh Đàm Vũ Trọng Tài đã quan tâm, hướng dẫn, truyền đạt những kiến thức và kinh nghiệm cho chúng em trong suốt thời gian học tập môn Lập trình Python cho Máy học.

Trong quá trình làm đồ án môn không tránh khỏi được những sai sót, chúng em mong nhận được sự góp ý của thầy và các bạn để được hoàn thiện hơn.

Contents

Lời mở đầu	1
1 Giới thiệu bài toán	4
1.1 Sơ lược về bộ dữ liệu	4
1.2 Giới thiệu các đặc trưng (feature) trong bộ dữ liệu	4
1.3 Lí do chọn bộ dữ liệu	5
1.3.1 Ứng dụng thực tế cao	5
1.3.2 Dữ liệu đa chiều, chứa nhiều yếu tố ảnh hưởng	5
1.3.3 Phù hợp để thử nghiệm nhiều kỹ thuật học máy	5
1.3.4 Hỗ trợ đánh giá chiến lược kinh doanh	6
1.3.5 Nhận xét tổng quan	6
2 EDA	6
2.1 Phân tích đơn biến	6
2.1.1 Weekly_Sales	7
2.1.2 Temperature	8
2.1.3 Fuel_Price	8
2.1.4 CPI	9
2.1.5 Size	9
2.1.6 IsHoliday	10
2.1.7 Nhận xét chung	10
2.2 Phân tích hai biến	11
2.2.1 Phân tích ảnh hưởng của các biến số	11
2.2.1.1 Weekly_Sales theo Month	11
2.2.1.2 Weekly_Sales theo Temperature	12
2.2.1.3 Weekly_Sales theo Fuel_Price	13
2.2.1.4 Weekly_Sales theo CPI	14
2.2.1.5 Weekly_Sales theo Unemployment	15
2.2.2 Phân tích ảnh hưởng của các biến phân loại	16
2.2.2.1 Weekly_Sales theo IsHoliday	16
2.2.2.2 Weekly_Sales theo Type	17
2.3 Phân tích đa biến	18
2.3.1 Phân tích từng biến	19
2.3.1.1 Weekly_Sales	19
2.3.1.2 Temperature	19
2.3.1.3 Fuel_Price	19
2.3.1.4 CPI	19
2.3.1.5 Unemployment	19
2.3.1.6 Size	19
2.3.2 Nhận xét chung	20
3 Tiền xử lí dữ liệu	20
3.1 Xử lí NULL	20
3.2 Xử lí Outliers	21
3.3 Mã hóa dữ liệu	23
3.4 Chuẩn hóa dữ liệu	23

4 Feature engineering	24
4.1 Loại bỏ các đặc trưng không có nhiều giá trị dự đoán	24
4.2 Tạo mới đặc trưng	25
4.3 Chọn lựa đặc trưng	26
5 Xây dựng mô hình	28
5.1 Thủ các mô hình ban đầu	28
5.2 Tìm bộ siêu tham số tốt nhất cho mô hình Random Forest, XGBoost . .	30
5.2.1 Tìm bộ siêu tham số tốt nhất cho mô hình Random Forest	30
5.2.2 Tìm bộ siêu tham số tốt nhất cho mô hình XGBoost	32
5.3 Kết hợp mô hình	34
6 Tài liệu tham khảo	35

1 Giới thiệu bài toán

- Link competition: [Walmart Sales Forecasting](#)
- Link Google Colab: [Google Colab](#)

1.1 Sơ lược về bộ dữ liệu

Bộ dữ liệu được sử dụng trong đề tài là dữ liệu bán lẻ của hệ thống siêu thị Walmart – một trong những chuỗi bán lẻ lớn nhất tại Hoa Kỳ. Dữ liệu bao gồm doanh thu hàng tuần của các cửa hàng theo từng phòng ban, kết hợp với các yếu tố ảnh hưởng từ bên ngoài như kinh tế vĩ mô, thời tiết, chương trình khuyến mãi và thời điểm trong năm (ngày lễ).

Bộ dữ liệu được chia thành bốn tệp chính:

- **train.csv**: Gồm 421.570 dòng và 5 cột, cung cấp thông tin về doanh thu hàng tuần của từng cửa hàng và phòng ban. Tệp dữ liệu được thu thập từ ngày 5 tháng 2 năm 2010 đến ngày 26 tháng 10 năm 2012.
- **test.csv**: Gồm 115.064 dòng và 4 cột, là tập kiểm tra không có thông tin doanh thu (Weekly_Sales). Tệp dữ liệu được thu thập từ ngày 2 tháng 11 năm 2012 đến ngày 26 tháng 7 năm 2013.
- **features.csv**: Gồm 8.190 dòng và 12 cột, chứa các yếu tố ảnh hưởng như điều kiện thời tiết (Temperature), giá nhiên liệu (Fuel_Price), chỉ số giá tiêu dùng (CPI), tỷ lệ thất nghiệp (Unemployment), mức giảm giá khuyến mãi (MarkDown1 đến MarkDown5), và chỉ báo ngày lễ (IsHoliday). Tệp dữ liệu được thu thập từ ngày 5 tháng 2 năm 2010 đến ngày 26 tháng 7 năm 2013, bao phủ cả train và test.
- **stores.csv**: Gồm 45 dòng và 3 cột, cung cấp thông tin về loại cửa hàng (Type) và diện tích (Size). Tệp dữ liệu chứa thông tin tĩnh, không gắn với mốc thời gian.

Chúng ta cần sử dụng thông tin ở 3 tập train.csv, test.csv, features.csv để dự đoán doanh thu cho từng hàng ở tập test.csv

Qua việc kết hợp và xử lý các tệp dữ liệu trên, có thể xây dựng một tập dữ liệu đầy đủ giúp mô hình học được mối quan hệ giữa các đặc trưng đầu vào và doanh số đầu ra.

1.2 Giới thiệu các đặc trưng (feature) trong bộ dữ liệu

Các tệp dữ liệu ban đầu được tổ chức riêng biệt, mỗi tệp chứa một khía cạnh thông tin cụ thể như doanh thu, đặc điểm cửa hàng, hoặc các yếu tố ảnh hưởng bên ngoài. Sau khi kết hợp các tập tin train.csv, features.csv, stores.csv và test.csv, bộ dữ liệu hoàn chỉnh bao gồm các đặc trưng đầu vào (features) như sau:

Tên đặc trưng	Mô tả
Store	Mã số cửa hàng (1–45).
Dept	Mã số phòng ban (ví dụ: đồ ăn, quần áo, điện tử...).
Date	Ngày trong tuần đại diện cho từng dòng dữ liệu.
Weekly_Sales	Doanh thu trong tuần (chỉ có trong tập huấn luyện).
IsHoliday	Biến nhị phân cho biết tuần đó có rơi vào kỳ nghỉ lễ hay không.
Temperature	Nhiệt độ trung bình trong tuần ($^{\circ}\text{F}$).
Fuel_Price	Giá nhiên liệu trung bình (USD/gallon).
CPI	Chỉ số giá tiêu dùng – đo lường mức độ lạm phát.
Unemployment	Tỷ lệ thất nghiệp ở khu vực hoạt động của cửa hàng (%).
MarkDown1–5	Mức giảm giá trong tuần (nếu có), chia theo 5 loại chương trình khác nhau.
Type	Loại cửa hàng: A, B, hoặc C – phản ánh quy mô và mô hình hoạt động.
Size	Diện tích cửa hàng (tính theo đơn vị foot vuông).

Table 1: Bảng các đặc trưng đầu vào trong bộ dữ liệu

1.3 Lí do chọn bộ dữ liệu

1.3.1 Ứng dụng thực tế cao

Dự báo doanh thu là một bài toán kinh điển và có tính ứng dụng rất cao trong lĩnh vực bán lẻ. Việc dự đoán chính xác doanh thu giúp doanh nghiệp tối ưu hóa quản lý hàng tồn kho, nhân sự, và kế hoạch tài chính. Với quy mô và độ phức tạp của hệ thống Walmart, bộ dữ liệu phản ánh một cách chân thực bối cảnh vận hành thực tế của một chuỗi bán lẻ lớn.

1.3.2 Dữ liệu đa chiều, chứa nhiều yếu tố ảnh hưởng

Không giống các bộ dữ liệu bán lẻ đơn giản khác, bộ dữ liệu này bao gồm nhiều loại biến đầu vào đa dạng, giúp phân tích sâu hơn:

- **Yếu tố kinh tế:** giá nhiên liệu (Fuel_Price), chỉ số CPI, tỷ lệ thất nghiệp.
- **Yếu tố thời tiết:** nhiệt độ.
- **Yếu tố khuyến mãi:** thông qua 5 mức MarkDown khác nhau.
- **Thời gian đặc biệt:** tuần có ngày lễ (IsHoliday).

Sự hiện diện của nhiều yếu tố giúp kiểm tra và phân tích ảnh hưởng của các biến ngoại sinh lên doanh thu.

1.3.3 Phù hợp để thử nghiệm nhiều kỹ thuật học máy

Dữ liệu phù hợp cho cả bài toán hồi quy (dự đoán doanh số theo nhiều yếu tố) và chuỗi thời gian (time series forecasting). Điều này cho phép thử nghiệm nhiều mô hình khác nhau như:

- **Các mô hình truyền thống:** Linear Regression, Random Forest, XGBoost.
- **Các mô hình chuỗi thời gian:** Prophet, ARIMA, LSTM.
- **Các kỹ thuật tiền xử lý:** xử lý missing values, phát hiện outliers, encoding và chuẩn hóa.

Việc này giúp người học củng cố kỹ năng xử lý dữ liệu, trích xuất đặc trưng và xây dựng pipeline mô hình hoàn chỉnh.

1.3.4 Hỗ trợ đánh giá chiến lược kinh doanh

Bộ dữ liệu cho phép thực hiện các phân tích chuyên sâu như:

- Dánh giá tác động của chương trình khuyến mãi đến doanh thu.
- Phân tích ảnh hưởng của các chỉ số kinh tế vĩ mô đến hành vi tiêu dùng.
- Xây dựng hệ thống dự báo giúp ra quyết định nhập hàng, lên kế hoạch bán hàng cho các kỳ nghỉ lễ.

1.3.5 Nhận xét tổng quan

Bộ dữ liệu có chất lượng cao, phù hợp với mục tiêu học thuật và thực tiễn. Tuy nhiên, dữ liệu cũng tồn tại một số thách thức như giá trị thiếu trong các cột Markdown, CPI, Unemployment, và thiếu thông tin chi tiết ở cấp độ sản phẩm. Tuy vậy, đây cũng chính là cơ hội để rèn luyện kỹ năng xử lý dữ liệu và thiết kế mô hình học máy phù hợp trong điều kiện thực tế.

2 EDA

Trong phần này, chúng ta sẽ thực hiện phân tích dữ liệu khám phá (Exploratory Data Analysis - EDA) nhằm tìm hiểu các đặc điểm tổng quát của tập dữ liệu, phát hiện các mẫu, xu hướng, giá trị ngoại lai và mối liên hệ giữa các biến. Phân tích EDA bao gồm ba mức độ: đơn biến, hai biến và đa biến

Để dễ quan sát, thực hiện merge 3 bộ data Features, Store và Train để tạo bộ data `df` (data train), Features, Store và Train để tạo bộ data `df_test` (data test) nhằm thuận tiện cho việc phân tích hai biến và đa biến

2.1 Phân tích đơn biến

Phân tích đơn biến tập trung vào việc khảo sát đặc điểm phân bố của từng biến độc lập thông qua việc sử dụng một số biểu đồ biểu diễn phân bố dữ liệu như histogram, boxplot.

2.1.1 Weekly_Sales

Đây là biến mục tiêu chính trong bài toán, phản ánh doanh số hàng tuần của từng cửa hàng và phòng ban.

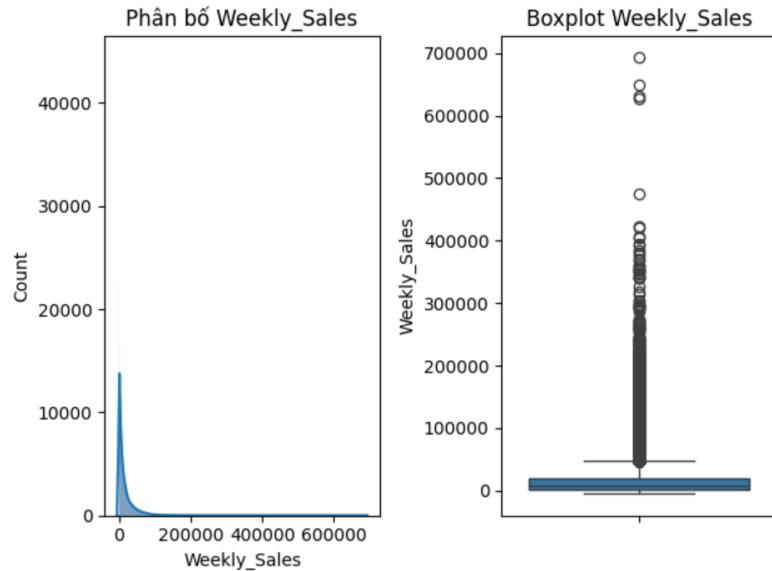


Figure 1: Biểu đồ histogram và boxplot của biến Weekly_Sales

Nhận xét:

- Dựa vào biểu đồ histogram, có thể thấy rằng Weekly_Sales có phân bố lệch phải mạnh (right-skewed), phần lớn giá trị nằm ở mức thấp (dưới 20.000), nhưng vẫn tồn tại một số giá trị doanh số rất lớn, vượt trên 600.000.
- Biểu đồ boxplot bên phải cũng minh họa rõ sự tồn tại của nhiều điểm ngoại lai (outliers) — đây có thể là những tuần đặc biệt có nhu cầu tăng vọt như dịp lễ, khuyến mãi lớn.

2.1.2 Temperature

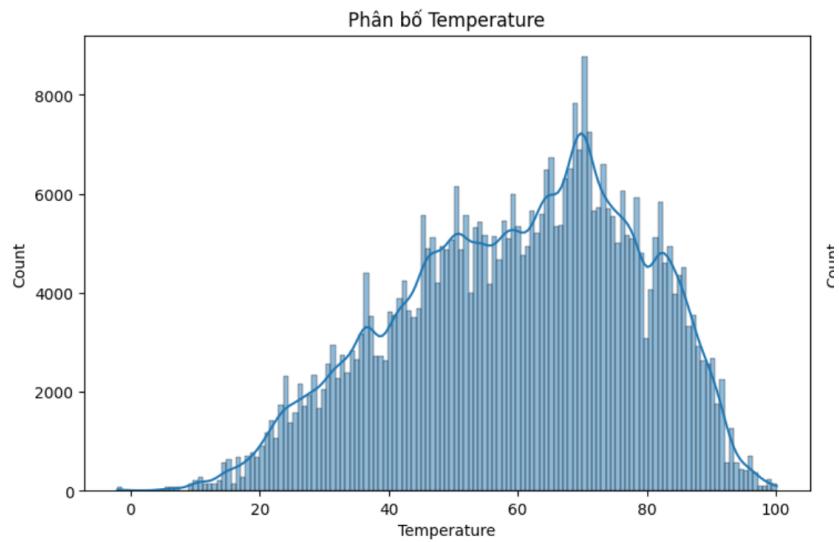


Figure 2: Biểu đồ histogram của biến Temperature

Nhận xét:

- Phân bố gần chuẩn, tập trung nhiều ở khoảng 60–80 độ F.
- Một số điểm dữ liệu rải rác ở hai đầu, phản ánh những tình huống thời tiết cực đoan có thể ảnh hưởng đến doanh số (do ít người đi mua sắm hơn).

2.1.3 Fuel_Price

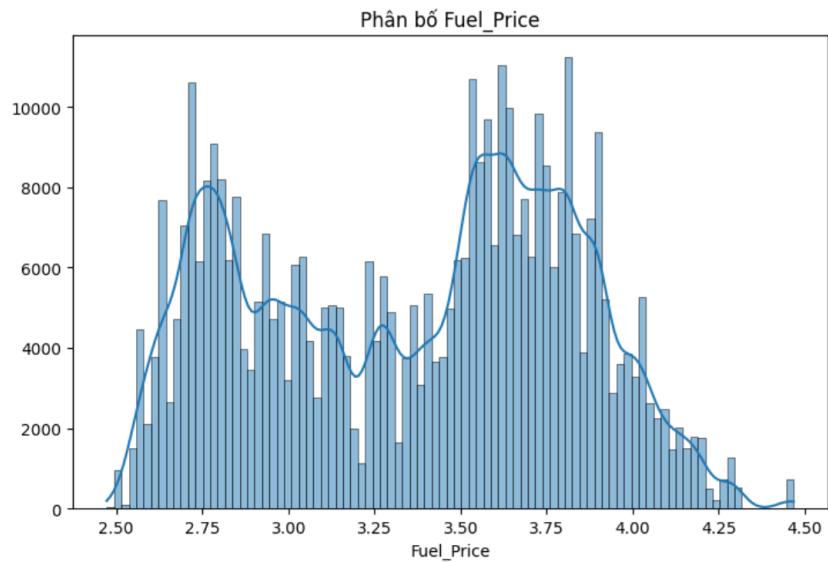


Figure 3: Biểu đồ histogram của biến Fuel_Price

Nhận xét:

- Phân bố không chuẩn và có nhiều đỉnh phụ, cho thấy giá nhiên liệu biến động theo từng thời điểm cụ thể.
- Phần lớn giá nằm trong khoảng từ 2.5 đến 4.2 USD/gallon.

2.1.4 CPI

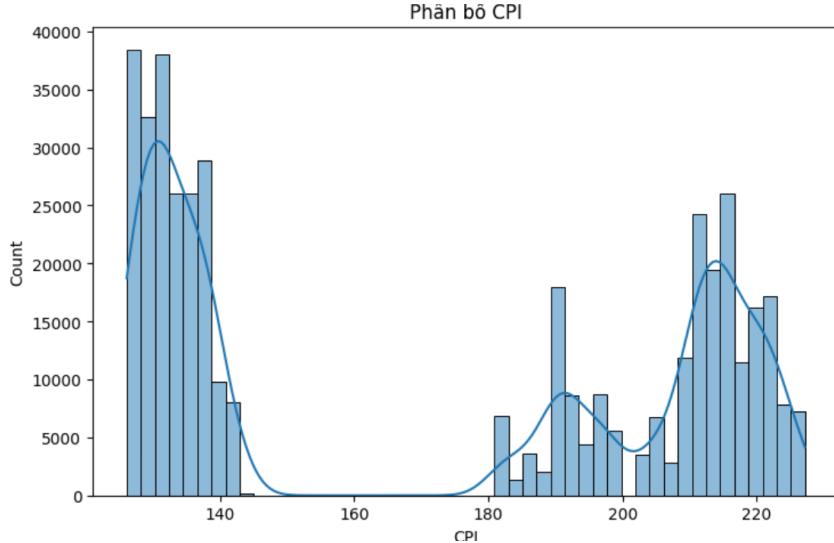


Figure 4: Biểu đồ histogram của biến CPI

Nhận xét:

- Phân bố lệch mạnh với hai cụm giá trị tách biệt rõ ràng, có thể do các nhóm cửa hàng thuộc các vùng có nền kinh tế khác nhau.
- Điều này cho thấy ảnh hưởng vùng miền có thể là một yếu tố quan trọng trong dự đoán doanh số.

2.1.5 Size

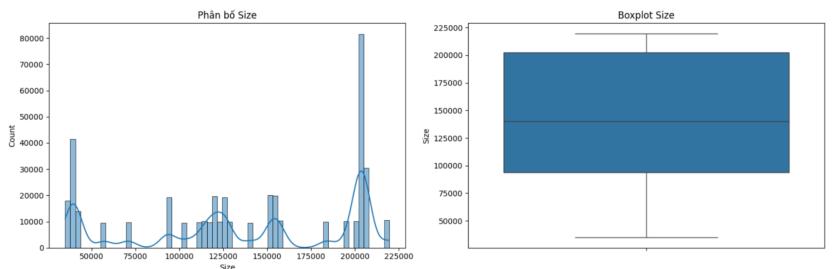


Figure 5: Biểu đồ histogram và box plot của biến Size

Nhận xét:

- Biểu đồ histogram cho thấy phân bố không chuẩn, với nhiều đỉnh nhỏ phản ánh các cụm diện tích phổ biến – điều này cho thấy Walmart có các mẫu cửa hàng chuẩn về quy mô (ví dụ như cụm quanh 40,000 và 200,000). Có xu hướng hơi lệch phải, nghĩa là tồn tại một số cửa hàng có diện tích rất lớn.

- Boxplot cho thấy các giá trị được phân bố khá đều, không có outlier rõ rệt. Khoảng từ phân vị trung位 từ 90,000 đến 200,000, phản ánh sự đa dạng trong kích thước cửa hàng.

2.1.6 IsHoliday

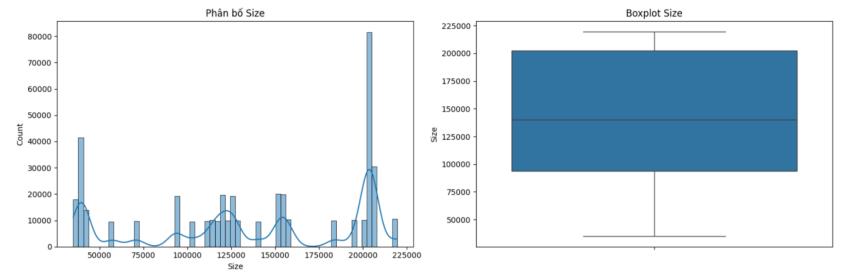


Figure 6: Biểu đồ histogram và box plot của biến IsHoliday

Nhận xét:

- Biểu đồ tần suất cho thấy phần lớn các tuần trong tập dữ liệu là tuần bình thường (False), chiếm khoảng 93%, trong khi các tuần lễ (True) chỉ chiếm khoảng 7%.

2.1.7 Nhận xét chung

Từ kết quả phân tích đơn biến, có thể rút ra một số đặc điểm quan trọng về cấu trúc dữ liệu:

- Biến mục tiêu Weekly_Sales** có phân bối lệch rõ rệt, với phần lớn giá trị doanh thu ở mức thấp và một số lượng nhỏ giá trị cực cao đóng vai trò ngoại lai. Điều này phản ánh tính chất biến động mạnh của doanh thu, đặc biệt trong các dịp lễ hoặc sự kiện khuyến mãi. Mô hình cần có khả năng xử lý tốt các phân bối lệch và giá trị outlier.
- Các biến liên tục** như Temperature, Fuel_Price, CPI, Unemployment, Size thể hiện đặc điểm phân bối khác nhau. Một số biến có phân bối gần chuẩn (Temperature), trong khi một số khác có phân bối lệch hoặc đa cực (CPI, Fuel_Price). Điều này cho thấy sự đa dạng về điều kiện kinh tế – xã hội và môi trường của các cửa hàng, cần được xử lý chuẩn hóa hoặc tạo nhóm phù hợp.
- Biến IsHoliday** là một biến nhị phân có tần suất xuất hiện đáng kể, với phần lớn các tuần không phải dịp lễ. Tuy nhiên, đây là biến có ý nghĩa quan trọng do doanh thu thường tăng đột biến trong các tuần lễ. Việc giữ lại biến này, cũng như tạo các đặc trưng thời gian bổ sung như SuperBowlWeek, Christmas,... là cần thiết để mô hình có thể học được các xu hướng mùa vụ.
- Biến Size** cho thấy sự phân tầng rõ rệt trong quy mô cửa hàng. Các cửa hàng lớn có thể có năng lực phục vụ khách hàng và trưng bày hàng hóa cao hơn, ảnh hưởng trực tiếp đến doanh thu. Phân bối biến này không đồng đều, nhưng không có outlier rõ rệt, phù hợp để đưa trực tiếp vào mô hình sau khi chuẩn hóa.

2.2 Phân tích hai biến

Trong phần này, chúng ta tập trung khảo sát mối quan hệ giữa biến mục tiêu Weekly_Sales và các biến đầu vào còn lại trong tập dữ liệu. Việc phân tích hai biến giúp xác định mức độ ảnh hưởng của từng đặc trưng lên doanh thu, từ đó làm cơ sở cho việc chọn lựa đặc trưng và xây dựng mô hình hiệu quả hơn.

Để thuận tiện trong việc đánh giá và phân tích, chúng em chia phần này thành hai nhóm chính: phân tích ảnh hưởng của các biến số lên Weekly_Sales và phân tích ảnh hưởng của các biến phân loại lên Weekly_Sales.

2.2.1 Phân tích ảnh hưởng của các biến số

2.2.1.1 Weekly_Sales theo Month

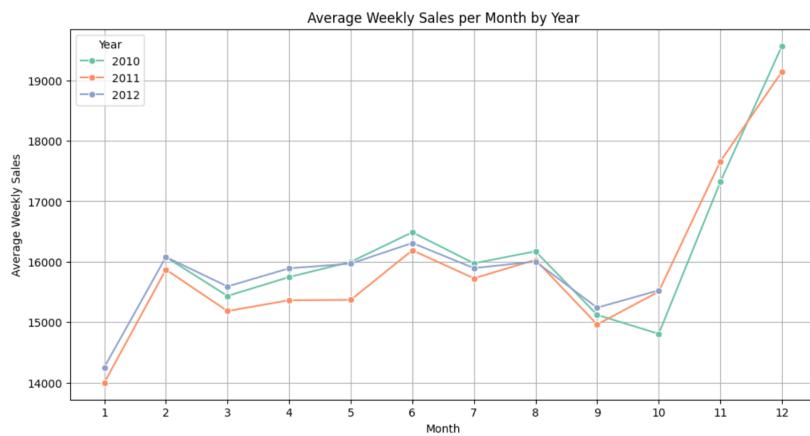


Figure 7: Biểu đồ doanh số trung bình theo tháng trong năm

Nhận xét:

- Về phân bố dữ liệu:

- Doanh số trung bình hàng tuần dao động trong khoảng 14.000 - 19.500.
- Tháng 12 có mức doanh số cao nhất, cho thấy ảnh hưởng của các dịp mua sắm lớn như Giáng Sinh hoặc Black Friday.
- Xu hướng doanh số theo từng tháng tương đối giống nhau giữa các năm, nhưng vẫn có sự khác biệt nhỏ ở một số thời điểm.

- Về tính tương quan giữa Weekly_Sales và Month:

- Không có mối quan hệ tuyến tính rõ ràng giữa thời gian và doanh số.
- Xu hướng doanh số theo tháng khá tương đồng giữa các năm, nhưng vẫn có sự khác biệt nhỏ ở một số tháng nhất định.
- Tăng mạnh vào tháng 2, giảm vào tháng 9 và tăng đột biến vào tháng 12, cho thấy yếu tố mùa vụ ảnh hưởng mạnh đến doanh số. Biến Month có ảnh hưởng đến Weekly_Sales nhưng không theo dạng tuyến tính.

- So sánh giữa các năm:

- Năm 2011 có doanh số thấp hơn năm 2010 khi xem xét toàn bộ xu hướng.
- Năm 2010 có mức doanh số trung bình cao nhất trong số các tháng đã có dữ liệu.
- Năm 2012 có doanh số trung bình gần bằng năm 2010, mặc dù chưa có dữ liệu của tháng 11 và 12.

2.2.1.2 Weekly_Sales theo Temperature

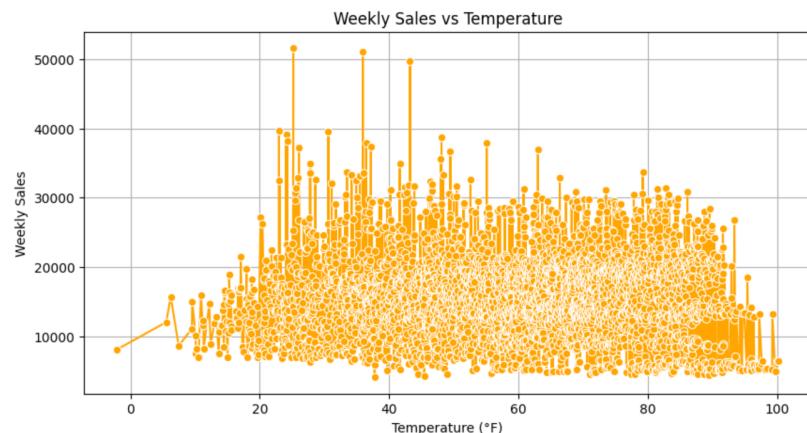


Figure 8: Biểu đồ doanh số theo nhiệt độ

Nhận xét:

• **Phân bố dữ liệu:**

- Doanh số có sự phân tán rộng trên toàn bộ dải nhiệt độ, từ 0°F đến 100°F.
- Phần lớn điểm dữ liệu tập trung trong khoảng 20°F - 80°F, cho thấy đây là khoảng nhiệt độ phổ biến nhất.
- Có một số điểm ngoại lệ với doanh số rất cao (>50.000) nhưng không xuất hiện thường xuyên.

• **Mối quan hệ tương quan giữa nhiệt độ và doanh số:**

- Không có mối tương quan tuyến tính rõ ràng giữa nhiệt độ và doanh số.
- Ở mức nhiệt độ dưới 20°F, doanh số có xu hướng thấp hơn, nhưng vẫn có một số điểm ngoại lệ với doanh số cao.
- Khi nhiệt độ tăng lên từ 20°F - 80°F, doanh số dao động mạnh và không theo một xu hướng cố định.
- Khi nhiệt độ vượt 80°F, doanh số có xu hướng giảm nhẹ, nhưng vẫn có nhiều điểm dữ liệu với doanh số cao.

• **Xu hướng tổng thể:**

- Doanh số không bị ảnh hưởng mạnh bởi nhiệt độ, nhưng có thể có một số tác động gián tiếp (ví dụ: mùa đông có thể ảnh hưởng đến hành vi mua sắm).

- Doanh số đạt mức cao nhất trong khoảng 30°F - 70°F , có thể phản ánh mức nhiệt độ phù hợp cho hoạt động mua sắm.
- Khi nhiệt độ cực đoan (quá lạnh hoặc quá nóng), doanh số có xu hướng giảm, nhưng vẫn có nhiều biến động.

2.2.1.3 Weekly_Sales theo Fuel_Price

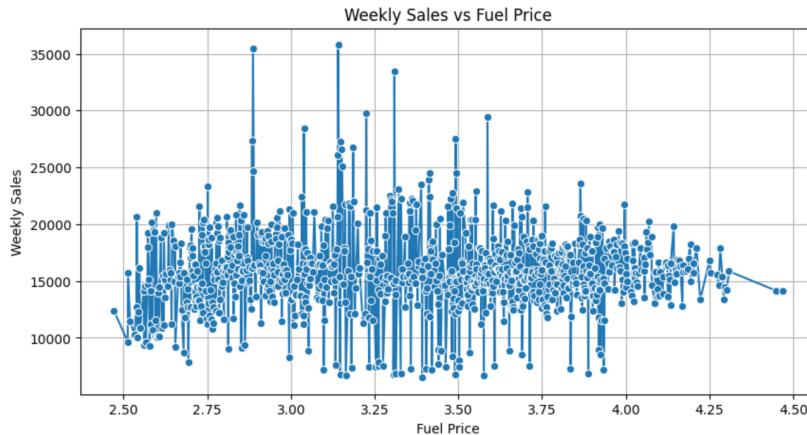


Figure 9: Biểu đồ doanh số theo giá nhiên liệu

Nhận xét:

• **Phân bố dữ liệu:**

- Biểu đồ thể hiện sự phân bố của doanh số bán hàng hàng tuần theo mức giá nhiên liệu.
- Dữ liệu có sự phân tán rộng, phản ánh mức độ biến động lớn trong doanh số.
- Phần lớn doanh số nằm trong khoảng từ 10.000 đến 20.000, nhưng có một số điểm ngoại lệ vượt 30.000.

• **Mối quan hệ giữa giá nhiên liệu và doanh số:**

- Không có mối tương quan tuyến tính rõ ràng giữa giá nhiên liệu và doanh số bán hàng.
- Dữ liệu phân tán rộng, doanh số dao động mạnh ở nhiều mức giá nhiên liệu khác nhau.
- Khi giá nhiên liệu thấp ($2.5 - 3.5$), doanh số có sự biến động lớn (10.000 - 35.000).
- Khi giá nhiên liệu cao (>4.0), doanh số có xu hướng tập trung ở mức thấp hơn (10.000 - 15.000), nhưng không có quy luật rõ ràng.
- Giá nhiên liệu không ảnh hưởng đáng kể đến doanh số.

• **Xu hướng tổng thể:**

- Khi giá nhiên liệu nằm trong khoảng 2.5 đến 4.0, doanh số dao động mạnh mà không theo một hướng cụ thể.

- Khi giá nhiên liệu vượt 4.0, có dấu hiệu doanh số giảm nhẹ, nhưng mức độ ảnh hưởng không đáng kể.
- Điều này gợi ý rằng tác động của giá nhiên liệu lên doanh số không thực sự mạnh.

2.2.1.4 Weekly_Sales theo CPI

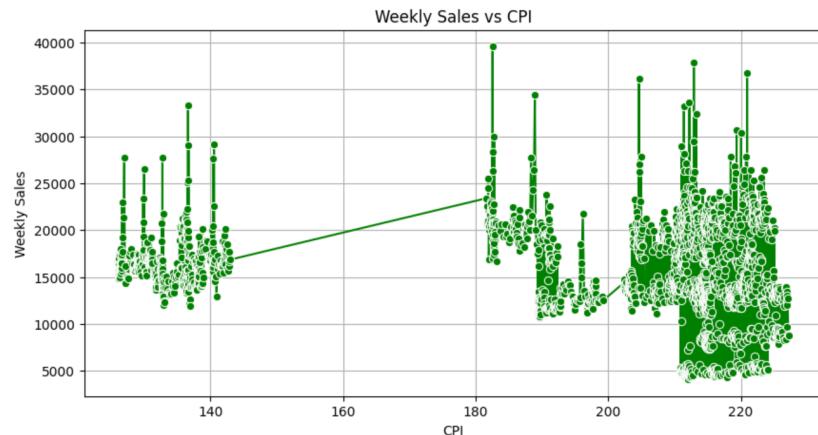


Figure 10: Biểu đồ doanh số theo CPI

Nhận xét:

• **Phân bố dữ liệu:**

- Dữ liệu được chia thành hai nhóm chính:
- Nhóm thứ nhất tập trung quanh CPI $\approx 130 - 140$ với doanh số dao động chủ yếu từ 10.000 - 30.000.
- Nhóm thứ hai nằm trong khoảng CPI $\approx 180 - 220$, với mức doanh số có sự phân tán rộng hơn.
- Ở cả hai nhóm, có nhiều điểm ngoại lệ với doanh số vượt 35.000, nhưng không xuất hiện thường xuyên.

• **Mối quan hệ tương quan giữa CPI và doanh số:**

- Nhóm CPI 130 - 140 có doanh số khá ổn định và không có xu hướng rõ ràng.
- Nhóm CPI 180 - 220 có sự phân tán mạnh hơn, với doanh số dao động lớn.
- Nhìn tổng thể, đường xu hướng có độ dốc nhẹ, cho thấy CPI có thể có mối tương quan dương yếu với doanh số, nhưng không rõ ràng.

• **Xu hướng tổng thể:**

- Khi CPI tăng từ 130 lên 140, doanh số không có sự thay đổi đáng kể.
- Khi CPI vượt 180, doanh số trở nên biến động hơn, có cả mức thấp lẫn cao hơn so với nhóm trước.
- Nhìn chung, CPI không ảnh hưởng rõ ràng đến doanh số, nhưng sự thay đổi của CPI có thể liên quan đến các yếu tố khác như lạm phát, giá cả hàng hóa, ảnh hưởng đến hành vi mua sắm của khách hàng.

2.2.1.5 Weekly_Sales theo Unemployment



Figure 11: Biểu đồ doanh số theo tỉ lệ thất nghiệp

Nhận xét:

- **Phân bố dữ liệu:**

- Dữ liệu được chia thành hai nhóm chính:
- Nhóm thứ nhất tập trung quanh Unemployment ≈ 4% - 6% với doanh số dao động chủ yếu từ 15.000 - 30.000.
- Nhóm thứ hai nằm trong khoảng Unemployment ≈ 6% - 10%, với mức doanh số có sự phân tán rộng hơn.
- Một số điểm ngoại lệ với doanh số vượt 30.000, nhưng không xuất hiện thường xuyên.

- **Mối quan hệ tương quan giữa Unemployment và doanh số:**

- Nhóm Unemployment 4% - 6% có doanh số khá cao và ổn định ở ngưỡng trên.
- Nhóm Unemployment 6% - 10% có sự phân tán mạnh hơn, với doanh số dao động lớn.
- Nhìn tổng thể, đường xu hướng có độ dốc giảm, cho thấy Unemployment có thể có mối tương quan nghịch với doanh số, nhưng không rõ ràng.

- **Xu hướng tổng thể:**

- Khi Unemployment tăng từ 4% lên 6%, doanh số có xu hướng giảm đáng kể.
- Khi Unemployment vượt 6%, doanh số trở nên biến động hơn, có cả mức thấp lẫn cao hơn so với nhóm trước.
- Nhìn chung, Unemployment có ảnh hưởng tiêu cực đến doanh số, thể hiện qua xu hướng giảm của doanh số khi tỷ lệ thất nghiệp tăng, đặc biệt rõ trong khoảng 4-10%. Tuy nhiên, khi Unemployment > 10%, doanh số có xu hướng ổn định hơn ở mức thấp.

2.2.2 Phân tích ảnh hưởng của các biến phân loại

2.2.2.1 Weekly_Sales theo IsHoliday

Để đánh giá mức độ ảnh hưởng của các tuần lễ đến doanh thu theo từng tháng trong năm, chúng tôi tiến hành so sánh xu hướng Weekly_Sales giữa các tuần lễ (IsHoliday = True) và tuần thường (IsHoliday = False) trong từng tháng.

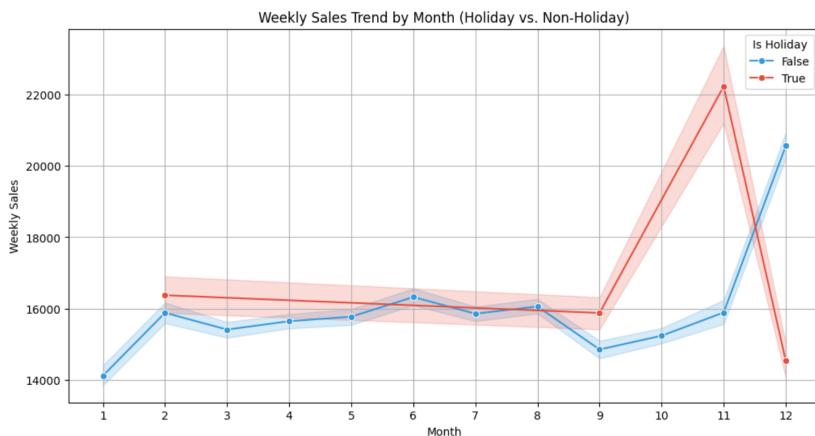


Figure 12: Biểu đồ doanh số theo tuần thường và tuần lễ

Nhận xét:

- **Phân bố dữ liệu:**

- Dữ liệu Weekly Sales được chia thành hai nhóm chính:
- Nhóm Non-Holiday (không phải ngày lễ): Có xu hướng ổn định trong phần lớn các tháng, dao động trong khoảng 14.000 - 16.000, với đỉnh đặc biệt vào tháng 12 (khoảng 20.000).
- Nhóm Holiday (ngày lễ): Xuất hiện chủ yếu ở các tháng 2, 9, 11 và 12, với mức biến động lớn hơn, đặc biệt là đỉnh cao nhất vào tháng 11 (trên 22.000).

- **Mối quan hệ giữa biến phân loại Holiday/Non-holiday và Weekly Sales:**

- Nhóm Holiday có mức doanh số trung bình cao hơn so với nhóm Non-Holiday ở hầu hết các thời điểm xuất hiện (tháng 2, 9, 11), ngoại trừ tháng 12.
- Sự chênh lệch rõ rệt nhất giữa hai nhóm xuất hiện ở tháng 11, khi doanh số ngày lễ vượt trội hơn hẳn (khoảng 22.000 so với 16.000).
- Nhìn tổng thể, Holiday có tương quan thuận với doanh số Weekly Sales, nhưng mức độ ảnh hưởng thay đổi theo tháng, thể hiện rõ sự tương tác giữa hai biến phân loại: Holiday và tháng trong năm.

- **Ảnh hưởng của biến phân loại đến biến số Weekly Sales:**

- Tác động của Holiday đến Weekly Sales thay đổi theo mùa vụ rõ rệt và ảnh hưởng rõ nhất trong 4 tháng có các ngày lễ lớn:
- Tháng 2, 9: Holiday cao hơn 5-7%
- Tháng 11: Holiday cao hơn 40% (ảnh hưởng mạnh nhất)

- Tháng 12: Holiday thấp hơn Non-Holiday
- Biểu đồ cho thấy rõ độ biến động của doanh số Holiday lớn hơn Non-Holiday, đặc biệt vào các tháng 11 và 12, cho thấy tính không ổn định của doanh số trong thời gian.

2.2.2.2 Weekly_Sales theo Type

Bên cạnh yếu tố thời gian và khuyến mãi, cấu trúc và quy mô của từng loại hình cửa hàng cũng có thể ảnh hưởng đáng kể đến doanh số. Do đó, chúng tôi tiến hành phân tích doanh số hàng tuần (Weekly_Sales) theo biến Type, trong đó mỗi giá trị A, B, C tương ứng với một loại cửa hàng khác nhau trong hệ thống của Walmart.

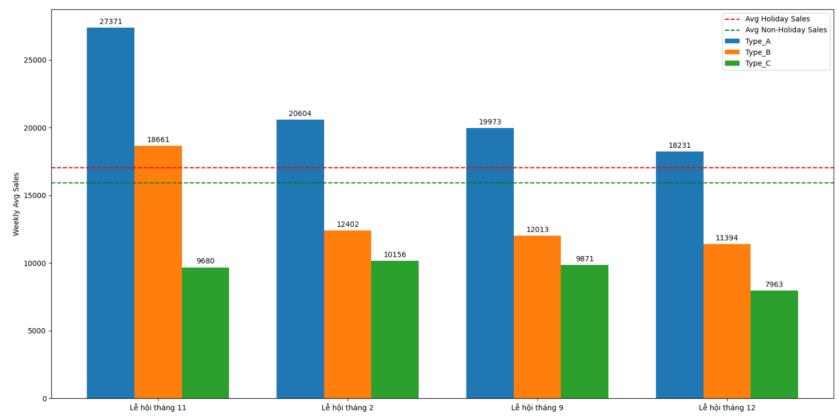


Figure 13: Biểu đồ doanh số theo nhiệt độ

Nhận xét:

- **So sánh giữa các loại cửa hàng:**

- Cửa hàng Type_A luôn có doanh số cao nhất, gấp 1.5-2 lần so với Type_B và 2-3 lần so với Type_C
- Cửa hàng Type_B có doanh số trung bình
- Cửa hàng Type_C có doanh số thấp nhất trong tất cả các dịp lễ

- **So sánh giữa các dịp lễ:**

- Lễ hội tháng 11 (Thanksgiving) mang lại doanh số cao nhất cho cả ba loại cửa hàng, đặc biệt là Type_A (27.371) và Type_B (18.661)
- Lễ hội tháng 2 (Super Bowl) có doanh số cao thứ hai, với Type_A đạt 20.604
- Lễ hội tháng 9 (Labor Day) và tháng 12 (Christmas) có doanh số thấp hơn

- **Kết luận chung:**

- Biểu đồ cho thấy hiệu quả kinh doanh khác nhau rõ rệt giữa các loại cửa hàng và các dịp lễ, với Thanksgiving là dịp mang lại doanh thu cao nhất và Type_A là loại hình cửa hàng hiệu quả nhất.

2.3 Phân tích đa biến

Phân tích đa biến được thực hiện nhằm đánh giá đồng thời mối quan hệ giữa nhiều biến trong tập dữ liệu, đặc biệt là mức độ tương quan giữa các đặc trưng đầu vào và biến mục tiêu Weekly_Sales. Trong phần này, chúng ta sử dụng biểu đồ heatmap ma trận tương quan để trực quan hóa hệ số tương quan Pearson giữa các biến số.

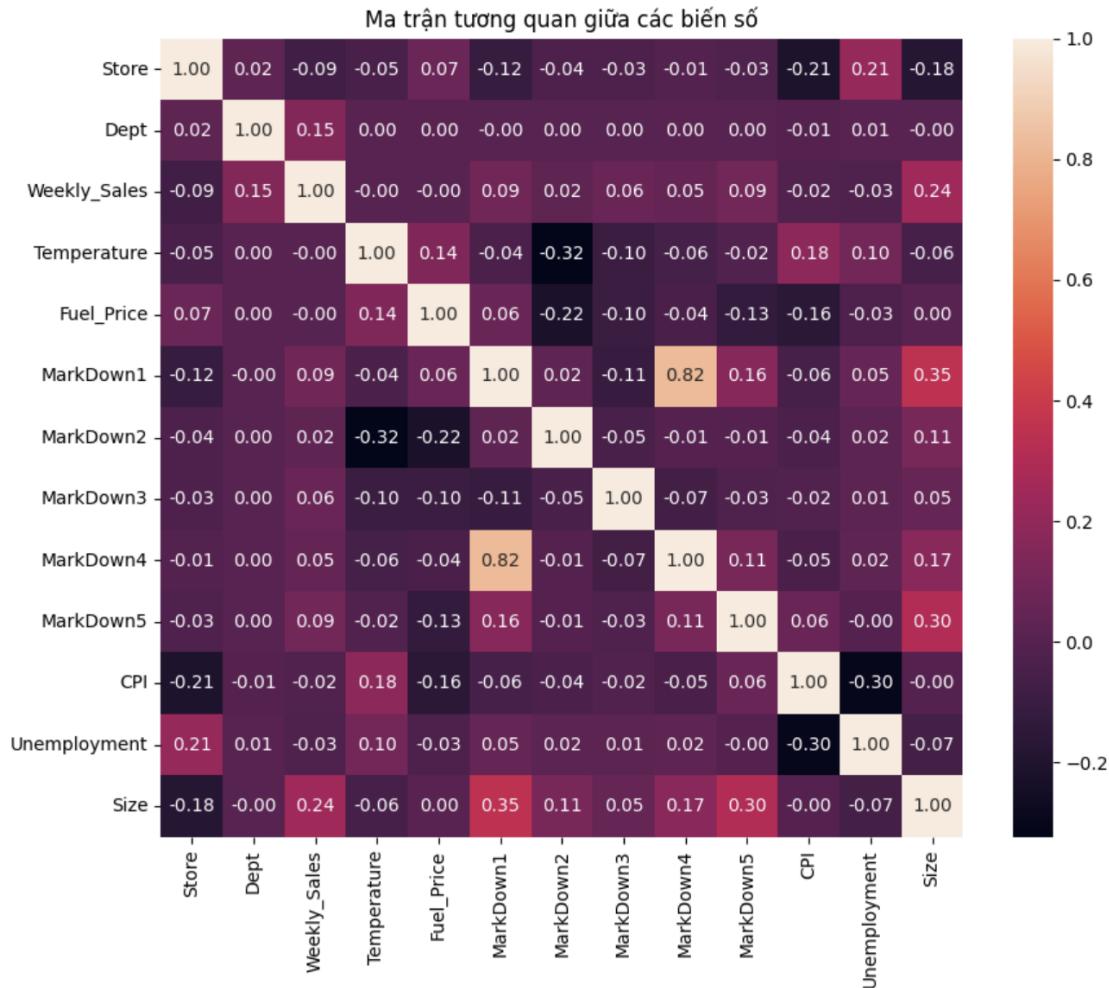


Figure 14: Ma trận tương quan giữa các biến

Qua biểu đồ heatmap, có thể xác định được:

- Các đặc trưng có mối tương quan tuyến tính mạnh hoặc yếu với biến mục tiêu.
- Mối liên hệ giữa các đặc trưng đầu vào, từ đó phát hiện những biến có thể gây ra hiện tượng đa cộng tuyến.
- Các nhóm biến có xu hướng cùng thay đổi, giúp định hướng việc chọn lựa đặc trưng đầu vào hoặc thiết kế đặc trưng tổng hợp.

Nhận xét:

2.3.1 Phân tích từng biến

2.3.1.1 Weekly_Sales

- Tương quan yếu với các biến còn lại:
- Tương quan dương yếu với biến Size (0.24) → Các cửa hàng lớn hơn có xu hướng có doanh số cao hơn.
- Với các biến khác như Temperature, Fuel_Price, CPI, Unemployment: tương quan rất yếu hoặc gần như bằng 0 → các yếu tố này không ảnh hưởng nhiều trực tiếp đến doanh số.

2.3.1.2 Temperature

- Có tương quan dương nhẹ với CPI (0.18) và Fuel_Price (0.14) → Yếu tố thời tiết có ảnh hưởng nhẹ đến chi tiêu và giá nhiên liệu.
- Tương quan yếu hoặc âm rất nhẹ với các biến khác.

2.3.1.3 Fuel_Price

- Tương quan âm nhẹ với CPI (-0.16) → khi giá nhiên liệu tăng, chi tiêu tiêu dùng có thể giảm.
- Không có mối liên hệ đáng kể nào với các biến khác.

2.3.1.4 CPI

- Tương quan âm tương đối với Unemployment (-0.30) → cho thấy mối quan hệ thường thấy trong kinh tế: khi thất nghiệp tăng, chi tiêu giảm (CPI giảm).
- Có mối tương quan yếu với các biến còn lại.

2.3.1.5 Unemployment

- Tương quan âm trung bình với CPI (-0.30) như trên.
- Gần như không có tương quan với các biến còn lại, đặc biệt là với biến Weekly_Sales.

2.3.1.6 Size

- Có tương quan yếu với Weekly_Sales (0.24) → các cửa hàng lớn thường bán được nhiều hơn.
- Không có mối tương quan đáng kể với các yếu tố khác.

2.3.2 Nhận xét chung

- **Không có mối tương quan mạnh giữa các biến:** Hầu hết các hệ số tương quan đều ở mức yếu hoặc trung bình, cho thấy các biến độc lập tương đối với nhau.
- **Các biến vĩ mô như CPI, thất nghiệp không ảnh hưởng mạnh mẽ đến doanh số:** Điều này có thể do tác động của các yếu tố vĩ mô thường có độ trễ hoặc ảnh hưởng gián tiếp.
- **Quy mô cửa hàng là yếu tố có tương quan đáng kể nhất với doanh số:** Cần ưu tiên yếu tố này trong mô hình dự báo, cùng với các yếu tố khác như loại cửa hàng và vị trí địa lý.

3 Tiền xử lí dữ liệu

3.1 Xử lí NULL

Các cột có giá trị NULL ở tập train và test như sau:

Cột	Số lượng NULL	Tỉ lệ %
MarkDown1	270889	64.257181
MarkDown2	310322	73.611025
MarkDown3	284479	67.480845
MarkDown4	286603	67.984676
MarkDown5	270138	64.079038

Table 2: Số lượng và tỉ lệ giá trị NULL trong tập train

Cột	Số lượng NULL	Tỉ lệ %
MarkDown1	149	0.129493
MarkDown2	28627	24.879198
MarkDown3	9829	8.542203
MarkDown4	12888	11.200723
CPI	38162	33.165890
Unemployment	38162	33.165890

Table 3: Số lượng và tỉ lệ giá trị NULL trong tập test

Cách xử lí giá trị NULL:

- **Đối với tập train:** chúng ta thấy có 5 cột có giá trị NULL là Markdown1, Markdown2, Markdown3, Markdown4, Markdown5, đây là các loại giảm giá khác như được áp dụng bởi Walmart. Chúng ta sẽ xử lí 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

```
1 markdown_col = ['MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4',
2                      , 'MarkDown5']
3
4 df[markdown_col] = df[markdown_col].fillna(0)
```

- **Đối với tập test:** chúng ta điền các giá trị NULL ở các cột Markdown bằng 0 như ở tập train. Ngoài ra ở tập này còn có thêm các giá trị NULL ở cột CPI, Unemployment, chúng ta xử lí bằng cách điền vào các giá trị này median của tập train

```

1 df_test[markdown_col]      = df_test[markdown_col].fillna(0)
2
3 median_cpi_train         = df['CPI'].median()
4 median_unemployment_train = df['Unemployment'].median()
5
6 df_test['CPI']            = df_test['CPI'].fillna(median_cpi_train)
7 df_test['Unemployment']   = df_test['Unemployment'].fillna(
8                                         median_unemployment_train)

```

Có một số phương pháp xử lí giá trị NULL khác như loại bỏ, nội suy nhưng không phù hợp với bộ dữ liệu này vì:

- **Loại bỏ:** tỷ lệ NaN trong các cột này rất cao, việc loại bỏ sẽ làm mất đi một lượng lớn dữ liệu quan trọng giảm đáng kể kích thước bộ dữ liệu và ảnh hưởng đến độ tin cậy của mô hình
- **Nội suy:** các giá trị Markdown không tuân theo một qui luật cụ thể nào và ít phụ thuộc vào các biến khác nên dùng nội suy sẽ có tính chính xác thấp

3.2 Xử lí Outliers

Đầu tiên chúng ta cần xử lí các cột Weekly_Sales có giá trị âm hoặc bằng 0. Có 1358 giá trị ở cột Weekly_Sales có giá trị âm hoặc bằng 0, chiếm 0.32%. Điều này là bất hợp lý theo phân tích ở phần EDA nên ta bỏ các phần này đi.

Chúng ta xem ở mỗi cột có bao nhiêu outliers(chỉ tập train). Ở đây chúng ta sẽ tìm bằng IQR, phương pháp IQR phát hiện outlier bằng cách 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.

Phương pháp xử lí Outliers:

- **Cột Temperature:** Biến này có một số outliers ở cả hai phía, điều này có thể làm sai lệch giá trị trung bình. Chúng ta sẽ thay thế các giá trị này bằng trung vị (median) để loại bỏ outliers mà không làm thay đổi đáng kể phân bố của dữ liệu
- **Cột Unemployment:** Biến này có phân bố lệch trái với nhiều outliers ở giá trị cao nhưng tỉ lệ giá trị outlier thấp. Chúng ta thay thế các giá trị bằng cách chỉnh giá trị vượt quá giá trị cho phép thành giá trị ở ngay biên
- **Cột Weekly_Sales, Markdown15** mặc dù cũng có giá trị outlier, tuy nhiên các mô hình sử dụng (Random Forest, XGBoost) ít nhạy cảm với outlier nên chúng ta không xử lí

Code xử lí Outliers:

```

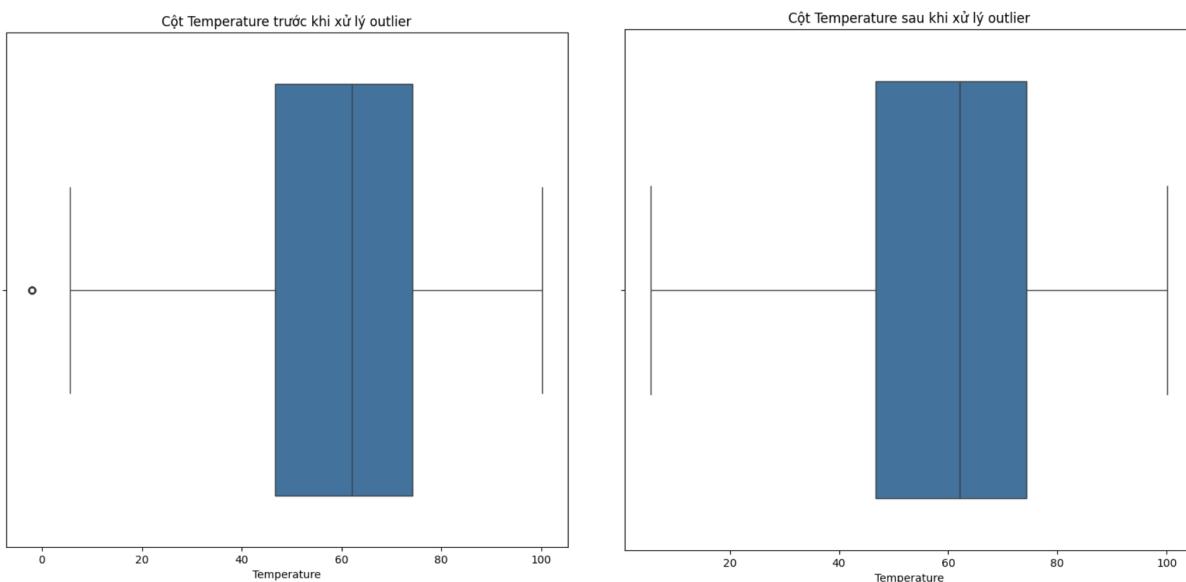
1 median_temp = df['Temperature'].median()
2 q1_temp = df['Temperature'].quantile(0.25)
3 q3_temp = df['Temperature'].quantile(0.75)
4 iqr_temp = q3_temp - q1_temp
5
6 upper_bound_temp = q3_temp + 1.5 * iqr_temp
7 lower_bound_temp = q1_temp - 1.5 * iqr_temp
8
9 df.loc[(df['Temperature'] > upper_bound_temp) | (df['Temperature'] <
       lower_bound_temp), 'Temperature'] = median_temp
10 df_test.loc[(df_test['Temperature'] > upper_bound_temp) | (df_test['Temperature'] <
       lower_bound_temp), 'Temperature'] = median_temp

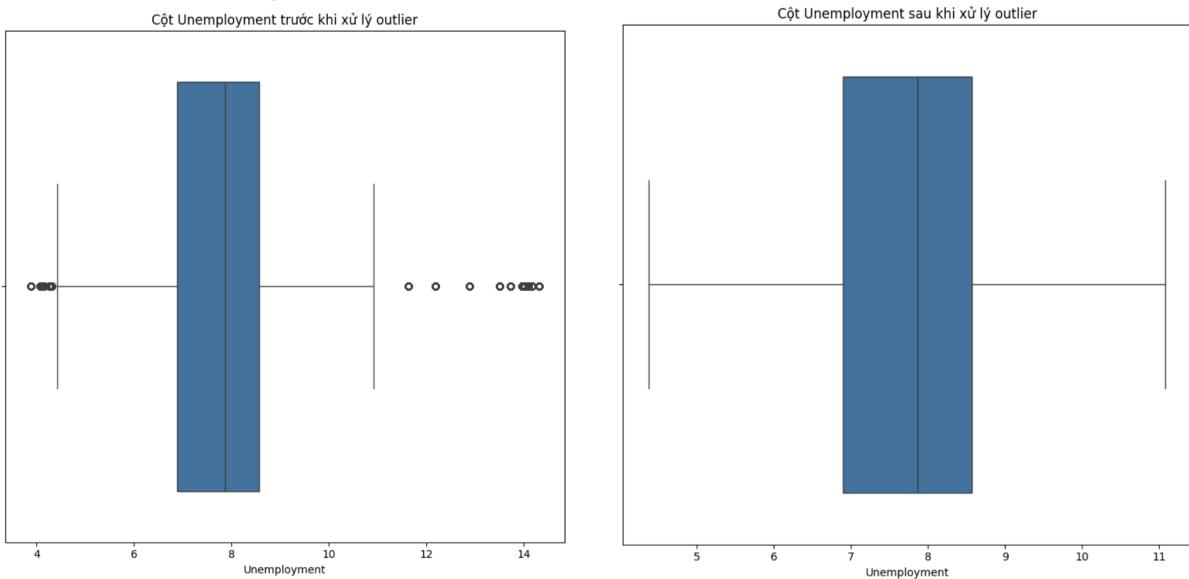
```

```

1 q1_unemp = df['Unemployment'].quantile(0.25)
2 q3_unemp = df['Unemployment'].quantile(0.75)
3 iqr_unemp = q3_unemp - q1_unemp
4
5 lower_bound_unemp = q1_unemp - 1.5 * iqr_unemp
6 upper_bound_unemp = q3_unemp + 1.5 * iqr_unemp
7
8 df['Unemployment'] = df['Unemployment'].clip(lower=lower_bound_unemp,
                                                upper=upper_bound_unemp)
9
10 df_test['Unemployment'] = df_test['Unemployment'].clip(lower=lower_bound_unemp,
                                                upper=upper_bound_unemp)
11

```





Sau khi xử lý giá trị outliers, chúng ta thấy giá trị của các cột thay đổi như sau:

- **Cột Temparature:** Khoảng giá trị giảm xuống, phân bố giá trị tập trung hơn quanh giá trị trung tâm, phân bố giá trị ít biến động hơn giữa các điểm tiếp
- **Cột Unemployment:** Phân bố giá trị ít lệch trái hơn, tập trung về giá trị trung tâm, dù vậy dữ liệu vẫn mang tính đại diện và mô hình ít bị ảnh hưởng hơn bởi các giá trị outliers

3.3 Mã hóa dữ liệu

Trong các cột dữ liệu, 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. Mặc dù có những mô hình hiểu nhầm sự phân cấp giữa các giá trị số nhưng những mô hình trong bài (Random Forest, XGBoost, LightGBM) thì không có vấn đề

```

1 le = LabelEncoder()
2
3 df['Type']      = le.fit_transform(df['Type'])
4 df_test['Type'] = le.transform(df_test['Type'])

```

Chúng ta cũng chuyển giá trị IsHoliday trong df 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

```

1 df['IsHoliday']      = df['IsHoliday'].apply(lambda x: 1 if x else 0)
2
3 df_test['IsHoliday'] = df_test['IsHoliday'].apply(lambda x: 1 if x else 0)

```

3.4 Chuẩn hóa dữ liệu

Chúng ta sẽ chuẩn hóa dữ liệu một số biến trong dữ liệu, mục đích là để đưa các biến số về một phạm vi cụ thể, tránh tình trạng các biến có thang đo lớn hơn chi phối mô hình học máy, giúp mô hình hoạt động chính xác hơn

```

1 standard_scaler = StandardScaler()
2 minmax_scaler = MinMaxScaler()
3
4 grp_standard = ['CPI', 'Fuel_Price', 'Size', 'Temperature', 'Unemployment']
5 grp_minmax = ['MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5']
6
7 df[grp_minmax] = minmax_scaler.fit_transform(df[grp_minmax])
8 df[grp_standard] = standard_scaler.fit_transform(df[grp_standard])
9
10 df_test[grp_minmax] = minmax_scaler.transform(df_test[grp_minmax])
11 df_test[grp_standard] = standard_scaler.transform(df_test[grp_standard])

```

Giải thích việc chuẩn hóa dữ liệu với các biến:

1. Min-Max Scaling:

- **Phân bố:** Các cột MarkDown 1-5 có rất nhiều giá trị bằng 0 do việc điền NULL
- **Lý do chọn:** Min-Max Scaling được sử dụng để đưa dữ liệu về một khoảng cố định, thường là từ 0 đến 1. Điều này đặc biệt hữu ích cho các biến có nhiều giá trị bằng 0, làm cho giá trị không bị nhiễu bởi phân phối không đối xứng, giúp cho việc so sánh và phân tích dễ dàng hơn

2. Standard Scaling:

- **Phân bố:** Các cột CPI, Fuel_Price, Size, Temperature, Unemployment có phân bố đa dạng
- **Lý do chọn:** Standard Scaling là lựa chọn phù hợp cho các biến này vì nó 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

4 Feature engineering

4.1 Loại bỏ các đặc trưng không có nhiều giá trị dự đoán

Các đặc trưng không có nhiều giá trị dự đoán là các đặc trưng chỉ định danh, các đặc trưng chỉ nhận một giá trị duy nhất. Vì vậy chúng ta sẽ xem tỉ lệ giá trị riêng biệt trên tổng giá trị ở từng cột

Cột	Số lượng giá trị riêng biệt	Tỉ lệ
Store	45	0.000107
Dept	81	0.000193
Date	143	0.000340
Weekly_Sales	358786	0.853673
IsHoliday	2	0.000005
Temperature	3527	0.008392
Fuel_Price	892	0.002122
MarkDown1	2278	0.005420
MarkDown2	1499	0.003567
MarkDown3	1662	0.003954
MarkDown4	1945	0.004628
MarkDown5	2294	0.005458
CPI	2145	0.005104
Unemployment	334	0.000795
Type	3	0.000007
Size	40	0.000095

Table 4: Số lượng giá trị riêng biệt và tỉ lệ của các cột

Nhận xét:

- 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 cột nào.

4.2 Tạo mới đặc trưng

Chúng ta tách cột Date thành 3 cột mới: Day, Month, Year. Điều này giúp xử lý các dữ liệu theo khoảng ngày, tháng, năm cụ thể dễ dàng hơn. Việc tách ra cũng giúp cải thiện độ chính xác mô hình vì có thể tìm ra các xu hướng biến động của giá trị theo mốc thời gian cụ thể. Chúng ta cũng sẽ thêm đặc trưng tuần trong năm, vào dữ liệu. Điều này giúp mô hình nhìn ra mối quan hệ danh số theo tuần trong năm, làm dự đoán chuẩn xác hơn

```

1 df['Date'] = pd.to_datetime(df['Date'])
2 df_test['Date'] = pd.to_datetime(df_test['Date'])
3
4 df['Day'] = df['Date'].dt.day
5 df['Month'] = df['Date'].dt.month
6 df['Year'] = df['Date'].dt.year
7
8 df_test['Day'] = df_test['Date'].dt.day
9 df_test['Month'] = df_test['Date'].dt.month
10 df_test['Year'] = df_test['Date'].dt.year
11
12 df['Week'] = df['Date'].dt.isocalendar().week
13
14 df_test['Week'] = df_test['Date'].dt.isocalendar().week

```

Chúng ta sẽ tạo đặc trưng mới Total_Markdown là tổng tất cả các giá trị giảm giá từ MarkDown1 đến MarkDown5. Lý do là vì các mức giảm giá được đưa theo từng mục khác nhau, chưa có mức tổng giảm giá của tất cả các mục. Điều này giúp phản ánh mức độ giảm giá tổng thể trong tuần và có thể cải thiện khả năng dự đoán doanh số.

```

1 df['Total_Markdown'] = df['MarkDown1'] + df['MarkDown2'] + df['MarkDown3'] +
2                                         df['MarkDown4'] + df['MarkDown5']
3
4 df_test['Total_Markdown'] = df_test['MarkDown1'] + df_test['MarkDown2'] +
5                                         df_test['MarkDown3'] + df_test['MarkDown4'] +
6                                         df_test['MarkDown5']

```

Chúng ta tạo đặc trưng mới đánh dấu các tuần lễ hội trong năm vì theo phân tích ở EDA, các tuần lễ hội 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

```

1 df['SuperBowlWeek'] = (df['Week'] == 6).astype(int)
2 df['LaborDay'] = (df['Week'] == 36).astype(int)
3 df['Thanksgiving'] = (df['Week'] == 47).astype(int)
4 df['Christmas'] = (df['Week'] == 52).astype(int)
5
6 df_test['SuperBowlWeek'] = (df_test['Week'] == 6).astype(int)
7 df_test['LaborDay'] = (df_test['Week'] == 36).astype(int)
8 df_test['Thanksgiving'] = (df_test['Week'] == 47).astype(int)
9 df_test['Christmas'] = (df_test['Week'] == 52).astype(int)

```

4.3 Chọn lựa đặc trưng

Đầu tiên chúng ta sử dụng ma trận tương quan nhưng không có biến mục tiêu, mục đích là loại các feature có mức độ đa cộng tuyển cao, những feature có giá trị lớn hơn một ngưỡng nhất định

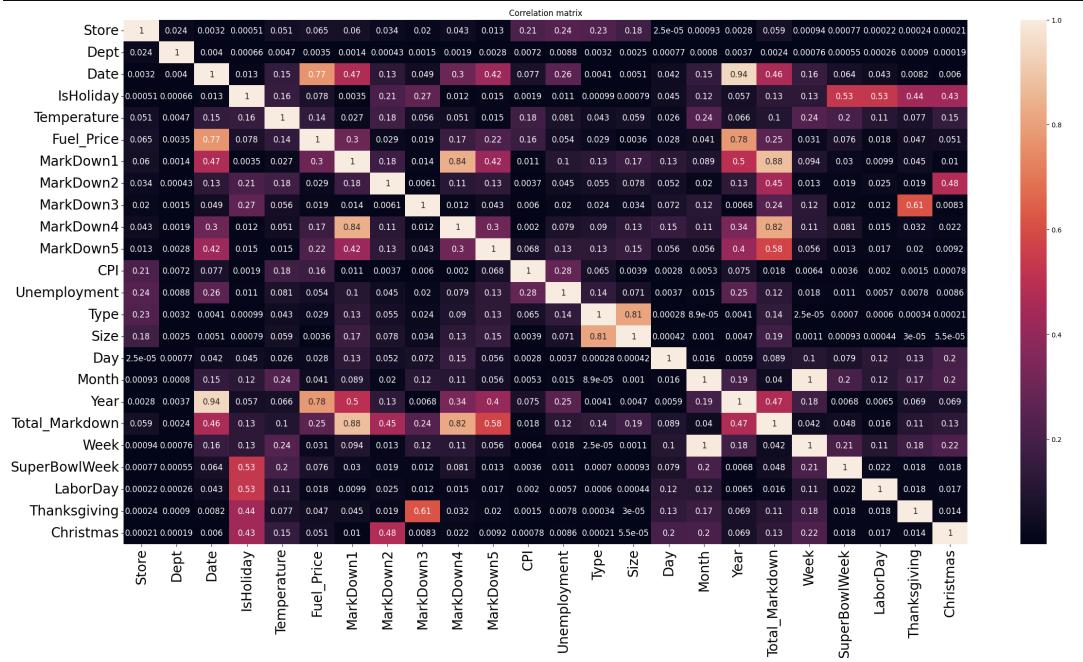


Figure 15: Ma trận tương quan giữa các đặc trưng không phải biến mục tiêu

Dựa vào ma trận tương quan, chúng ta có những 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

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

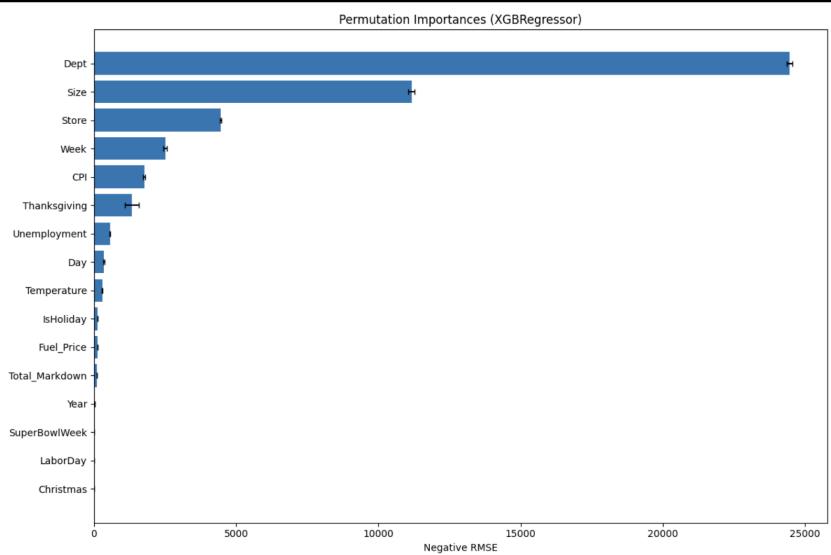


Figure 16: Độ quan trọng của các đặc trưng

Chúng ta có thể thấy rằng 3 features: Dept, Size, Store 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. Một số features cũng có ảnh hưởng đến doanh thu cửa hàng nhưng không đáng kể

Sau khi xem xét độ quan trọng của các feature theo từng phương pháp. Chúng ta sẽ lấy các biến sau để dự đoán

```
1 features = ['Store', 'Dept', 'IsHoliday', 'Size', 'Week', 'Year', 'Day']
```

5 Xây dựng mô hình

Chúng ta sẽ chia data train để huấn luyện mô hình theo kiểu train / test split với tỉ lệ 75 : 15

```
1 X = df[features]
2 y = df['Weekly_Sales']
3
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.15,
5 random_state = 42)
```

Mặc dù đây là dữ liệu Time-series, cách làm này đảm bảo không có data leakage vì thời gian của data test được định nghĩa ở phía sau thời gian của data train (đã đổi chiều với các notebook Kaggle khác)

5.1 Thủ các mô hình ban đầu

Chúng ta sẽ sử dụng các mô hình sau:

- **Random Forest:** một thuật toán học máy sử dụng nhiều cây quyết định để đưa ra dự đoán. Mỗi cây được huấn luyện trên một tập dữ liệu con được lấy mẫu ngẫu nhiên từ tập dữ liệu gốc và sử dụng một tập con các đặc trưng tại mỗi nút phân chia. Kết quả dự đoán cuối cùng được xác định bằng cách tổng hợp kết quả từ tất cả các cây

- **XGBoost (Extreme Gradient Boosting):** một thuật toán tăng cường gradient được thiết kế để tối ưu hóa tốc độ và hiệu suất. Nó xây dựng các cây quyết định tuần tự, mỗi cây mới cố gắng sửa lỗi của các cây trước bằng cách tối thiểu hóa hàm mất mát thông qua phương pháp gradient descent. XGBoost sử dụng các kỹ thuật như regularization để ngăn chặn overfitting và hỗ trợ tính toán song song, giúp xử lý hiệu quả các tập dữ liệu lớn. XGBoost là một mô hình rất mạnh trong việc dự đoán doanh thu, đã được chứng minh qua các kì thi
- **LightGBM (Light Gradient Boosting Machine):** một thuật toán học máy nổi bật với khả năng xử lý nhanh chóng và hiệu quả trên các tập dữ liệu lớn. Trong dự đoán doanh thu, LightGBM phù hợp vì khả năng xử lý dữ liệu phức tạp, bao gồm yếu tố mùa vụ, xu hướng và ảnh hưởng bên ngoài. Mô hình này có thể mô hình hóa mối quan hệ phi tuyến tính và tương tác giữa nhiều đặc trưng, giúp cải thiện độ chính xác trong dự đoán

Dựa theo tính chất của bài toán, chúng ta sẽ dùng các thang đo sau để đo lường tính chính xác của mô hình:

- **MSE:** Phạt nặng các dự đoán có độ lệch cao so với giá trị thật
- **MAE:** Tính trung bình sai số trong quá trình dự đoán
- **RMSE:** Tương tự MSE
- **WMAE:** Đây là thang đo của competition, phạt nặng các dự đoán có sai số trong các tuần nghỉ lẽ hơn gấp 5 lần so với các dự đoán có sai số trong các tuần không phải tuần nghỉ lẽ. Công thức được tính như sau:

$$\text{WMAE} = \frac{1}{\sum w_i} \sum_{i=1}^n w_i |y_i - \hat{y}_i|$$

trong đó:

- n là số lượng dòng dữ liệu
- \hat{y}_i là giá trị doanh số dự đoán
- y_i là giá trị doanh số thực tế
- w_i là trọng số. $w = 5$ nếu tuần đó là tuần nghỉ lẽ, ngược lại $w = 1$

Kết quả đánh giá các mô hình như sau:

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

Table 5: So sánh kết quả giữa các mô hình đơn giản

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

5.2 Tìm bộ siêu tham số tốt nhất cho mô hình Random Forest, XGBoost

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 từng siêu tham số tối ưu khi kết hợp chưa chắc là bộ siêu tham số tốt nhất

5.2.1 Tìm bộ siêu tham số tốt nhất cho mô hình Random Forest

Chúng ta sẽ tuning các siêu tham số sau:

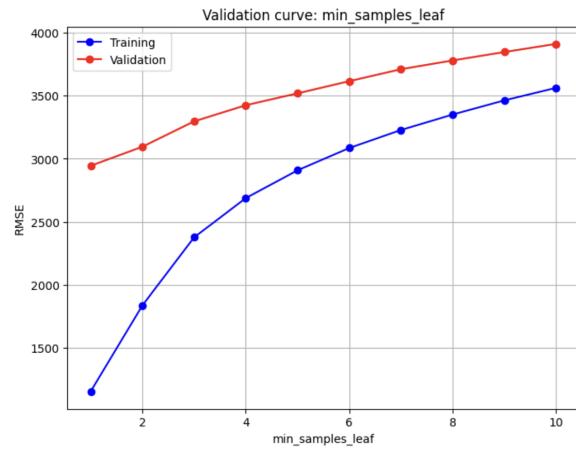
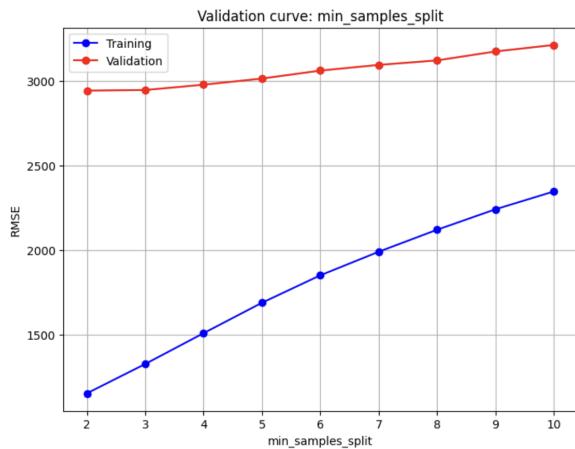
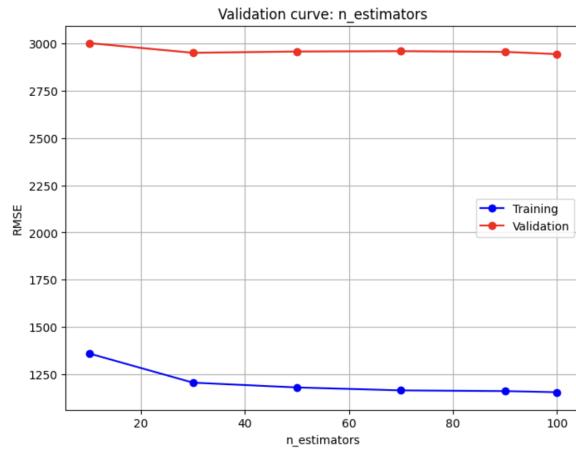
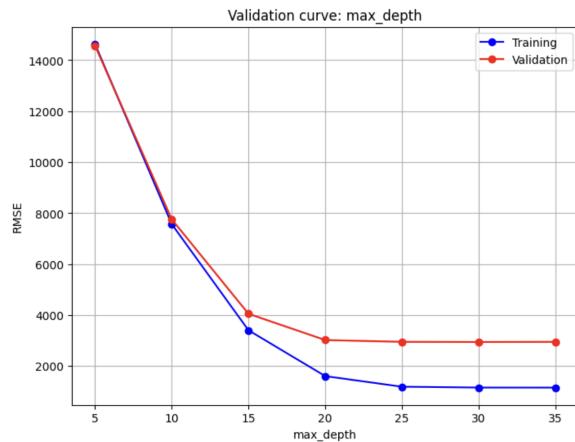
- **n_estimators:** số lượng cây trong rừng
- **max_depth:** độ sâu tối đa của mỗi cây
- **min_samples_split:** số mẫu tối thiểu tại một nút để thực hiện phép tách
- **min_samples_leaf:** số mẫu tối thiểu tại mỗi nút lá, đảm bảo lá không quá nhỏ
- **max_features:** số thuộc tính ngẫu nhiên được chọn 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

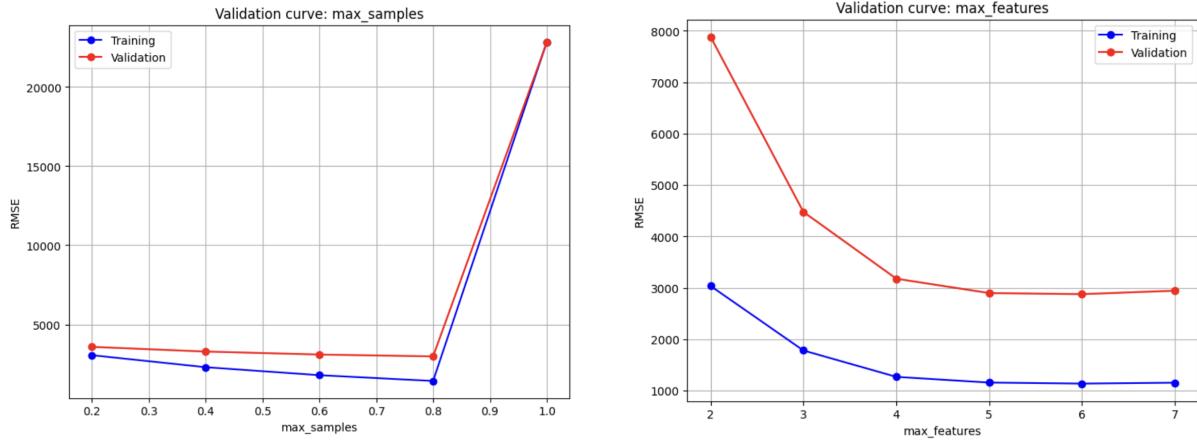
Hàm tính sai số và vẽ Validation Curves với từng siêu tham số như sau:

```

1 def test_params(**params):
2     model = RandomForestRegressor(random_state=42, n_jobs=-1, **params).fit(X_train,
3         , y_train)
4     train_rmse = RMSE(y_train, model.predict(X_train))
5     val_rmse = RMSE(y_test, model.predict(X_test))
6     return train_rmse, val_rmse
7
8
9 def test_param_and_plot(param_name, param_values):
10    train_errors, val_errors = [], []
11    for value in param_values:
12        params = {param_name: value}
13        train_rmse, val_rmse = test_params(**params)
14        train_errors.append(train_rmse)
15        val_errors.append(val_rmse)
16    plt.figure(figsize=(8, 6))
17    plt.title('Validation curve: ' + param_name)
18    plt.plot(param_values, train_errors, 'b-o')
19    plt.plot(param_values, val_errors, 'r-o')
20    plt.xlabel(param_name)
21    plt.ylabel('RMSE')
22    plt.legend(['Training', 'Validation'])
23    plt.grid(True)

```





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)

```

1 rf = RandomForestRegressor(n_jobs=-1,
2                             max_depth=30,
3                             n_estimators=130,
4                             min_samples_split=2,
5                             min_samples_leaf=1,
6                             max_samples=0.9999,
7                             max_features=6,
8                             random_state=42
9 )

```

Kết quả của mô hình sau khi tuning 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

Table 6: So sánh kết quả Random Forest trước và sau khi tuning hyperparameter

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

5.2.2 Tìm bộ siêu tham số tốt nhất cho mô hình XGBoost

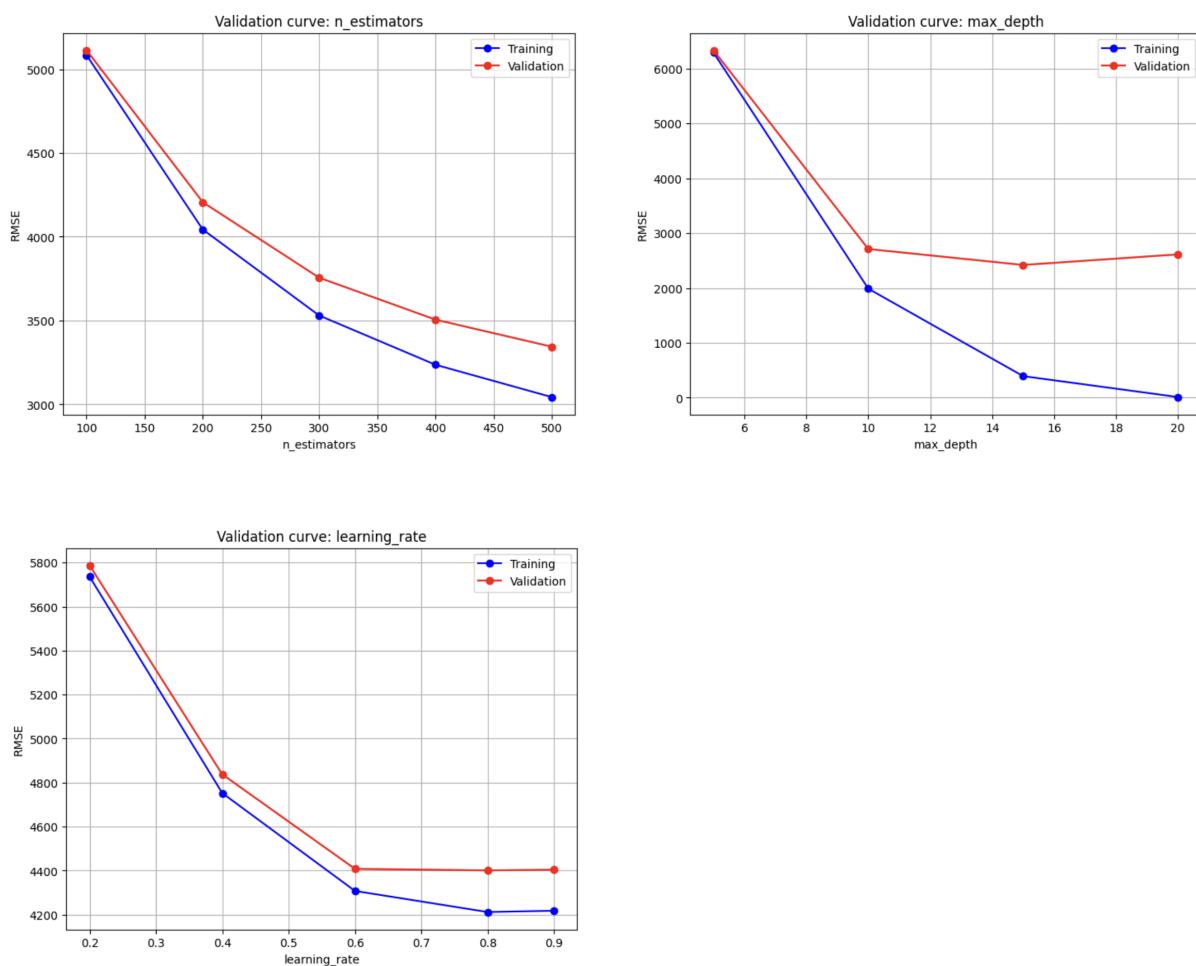
Chúng ta sẽ tuning các siêu tham số sau:

- **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

```

1 def test_params_xgb(**params):
2     model = XGBRegressor(random_state=42, n_jobs=-1, **params).fit(X_train, y_train)
3     train_rmse = RMSE(y_train, model.predict(X_train))
4     val_rmse = RMSE(y_test, model.predict(X_test))
5     return train_rmse, val_rmse
6
7 def test_param_and_plot_xgb(param_name, param_values):
8     train_errors, val_errors = [], []
9     for value in param_values:
10         params = {param_name: value}
11         train_rmse, val_rmse = test_params_xgb(**params)
12         train_errors.append(train_rmse)
13         val_errors.append(val_rmse)
14     plt.figure(figsize=(8, 6))
15     plt.title('Validation curve: ' + param_name)
16     plt.plot(param_values, train_errors, 'b-o')
17     plt.plot(param_values, val_errors, 'r-o')
18     plt.xlabel(param_name)
19     plt.ylabel('RMSE')
20     plt.legend(['Training', 'Validation'])
21     plt.grid(True)

```



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)

```

1 xgb = XGBRegressor(random_state=42,
2                     n_jobs=-1,
3                     n_estimators=500,
4                     max_depth=20,
5                     learning_rate=0.8
6 )

```

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)	8449900.79	1368.28	2906.69	1470.96

Table 7: So sánh kết quả XGBoost trước và sau khi tuning hyperparameter

So sánh với mô hình ban đầu, XGBoost sau khi tuning hyperparameter đã có cải thiện đáng kể trên tất cả các thước đo. Việc tối ưu hóa siêu tham số đã giúp giảm đáng kể sai số dự đoán và nâng cao hiệu suất của mô hình.

5.3 Kết hợp mô hình

Sau khi tìm ra bộ siêu tham số tốt nhất cho mô hình Random Forest và XGBoost, chúng ta sẽ dùng kỹ thuật Weighted Average để tăng độ chính xác. Weighted Average là một kỹ thuật ensemble đơn giản, trong đó mỗi kết quả của mỗi mô hình được nhân với một trọng số nhất định vào kết quả dự đoán tổng hợp. Điều này giúp 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

Chúng ta sẽ kết hợp kết quả dự đoán của mô hình XGBoost và Random Forest. 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. Tỉ lệ kết hợp là 60 : 40 cho Random Forest và XGBoost

```

1 y_pred_avg = y_pred_rf * 0.6 + y_pred_xgb * 0.4

```

Kết quả so sánh các mô hình như sau:

Mô hình	MSE	MAE	RMSE	WMAE
Random Forest (hyperparameter tuning)	8449648.91	1220.08	2906.82	1352.00
XGBoost (hyperparameter tuning)	8449900.79	1368.28	2906.69	1470.96
Weighted Average (RF + XGBoost)	7063884.80	1173.90	2657.79	1282.54

Table 8: So sánh kết quả giữa các mô hình đã tối ưu và mô hình ensemble

Kết quả cho thấy mô hình Weighted Average kết hợp Random Forest và XGBoost đã đạt được hiệu suất tốt nhất trên tất cả các thước đo. Việc kết hợp đã giúp giảm đáng kể sai số dự đoán so với từng mô hình riêng lẻ, chứng minh hiệu quả của phương pháp ensemble learning.

Tổng hợp kết quả của các mô hình như sau:

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)	8449900.79	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

Table 9: Tổng hợp kết quả so sánh các mô hình

Sau đó đã nộp kết quả mô hình Weighted Average lên Kaggle



Kết quả dự đoán được top 5% submission có dự đoán tốt nhất (huy chương đồng)

6 Tài liệu tham khảo

- [1] <https://scikit-learn.org/stable/index.html>
- [2] <https://www.kaggle.com/code/maxdiazbattan/walmart-sales-top-3-eda-feature-engineering>
- [3] <https://www.kaggle.com/code/yasirhussain1987/eda-and-store-sales-predictions-using>
- [4] <https://www.kaggle.com/code/avelinocaio/walmart-store-sales-forecasting#4-Christmas-Adjustment>