ĐẠI HỌC KINH TẾ QUỐC DÂN

ĐỀ TÀI : DỰ BÁO DOANH SỐ BÁN HÀNG CỦA CHUỖI SIÊU THỊ FAVORITA

Thành viên nhóm:

Bùi Văn Thái

Vũ Quốc Tấn

Lê Nguyễn Minh Duy

Hoàng Quốc Việt

MỤC LỤC

1 ĐẶT VẤN ĐỀ

1.1 Bối cảnh chung

Trong bối cảnh thị trường bán lẻ ngày càng cạnh tranh và biến động, khả năng dự báo chính xác doanh số bán hàng trở thành yếu tố then chốt để tối ưu hóa hoạt động kinh doanh. Việc nắm bắt các yếu tố ảnh hưởng đến doanh số – từ xu hướng thời gian, các dịp lễ tết, sự kiện đặc biệt, đến tác động của giá dầu hay chính sách khuyến mãi – giúp doanh nghiệp đưa ra quyết định hiệu quả trong quản lý tồn kho, nhân sự, marketing và mở rộng hệ thống cửa hàng.

Dữ liệu bán lẻ có đặc thù phức tạp: nhiều chuỗi thời gian theo từng cửa hàng và từng nhóm sản phẩm, chịu ảnh hưởng mạnh từ yếu tố ngoại sinh, và tồn tại tỷ lệ lớn ngày không phát sinh doanh số (zero sales). Do đó, việc dự báo trong bối cảnh này đòi hỏi phương pháp tiếp cận linh hoạt và mạnh mẽ, vừa xử lý được biến động mùa vụ, vừa kết hợp được các biến giải thích đa dạng.

1.2 Bối cảnh Ecuador và dữ liệu nghiên cứu

Nghiên cứu này lựa chọn bộ dữ liệu từ Corporación Favorita – một trong những tập đoàn bán lẻ lớn nhất Ecuador, sở hữu các chuỗi siêu thị như Supermaxi và Akí. Doanh số của Favorita phản ánh trực tiếp nhu cầu tiêu dùng thiết yếu của người dân và chịu tác động rõ rệt từ các yếu tố kinh tế – xã hội tại Ecuador.

Đặc điểm của thị trường bán lẻ Ecuador là sự kết hợp giữa kênh truyền thống và hiện đại, đồng thời mang tính mùa vụ cao, gắn với lịch nghỉ lễ và chu kỳ chi trả lương (ngày 15 và cuối tháng). Bên cạnh đó, nền kinh tế Ecuador sử dụng đồng USD, giúp lạm phát ổn định nhưng cũng khiến sức mua dễ bị ảnh hưởng bởi biến động giá dầu – một trong những nguồn thu chủ lực của quốc gia.

Trong bối cảnh đó, dữ liệu doanh số Favorita giai đoạn 2013–2017 vừa phản ánh rõ nét đặc trưng của ngành bán lẻ, vừa chứa nhiều thách thức: dữ liệu đa chiều (cửa hàng × sản phẩm × thời gian), ảnh hưởng mạnh từ sự kiện đặc biệt (ví dụ: động đất năm 2016), và tỷ lệ zero sales đáng kể. Đây là cơ sở thực tiễn quan trọng để kiểm định các mô hình dự báo hiện đại.

2 MỤC TIÊU NGHIÊN CỨU

2.1 Mục tiêu tổng quát:

- Mục tiêu tổng quát của đề tài là xây dựng và đánh giá một mô hình dự báo doanh số bán lẻ hiệu quả, có khả năng dự đoán chính xác cả việc một mặt hàng có bán được hay

không và nếu có thì bán được bao nhiêu, dựa trên các dữ liệu lịch sử và các yếu tố ảnh hưởng khác.

2.2 Mục tiêu cụ thể (Câu hỏi nghiên cứu):

- Để đạt được mục tiêu tổng quát, nghiên cứu này sẽ trả lời các câu hỏi sau:
- Làm thế nào để tiền xử lý và tích hợp các nguồn dữ liệu khác nhau (doanh số, cửa hàng, dầu, ngày lễ) để tạo ra một tập dữ liệu phù hợp cho mô hình dự báo?
- Những đặc trưng nào (từ dữ liệu thời gian, cửa hàng, sản phẩm, ngày lễ và giá dầu) có ảnh hưởng đáng kể đến doanh số bán lẻ?
- Làm thế nào để xây dựng một mô hình hai bước, kết hợp mô hình phân loại (dự đoán có bán được hay không) và mô hình hồi quy (dự đoán số lượng bán được khi có bán), để giải quyết bài toán dự báo doanh số bao gồm cả giá trị 0?
- Mô hình hai bước được đề xuất đạt hiệu quả như thế nào trên tập dữ liệu kiểm định (validation set) dựa trên các chỉ số đánh giá phù hợp (ví dụ: RMSLE)?
- Làm thế nào để áp dụng mô hình đã huấn luyện để dự báo doanh số trên tập dữ liệu test và so sánh kết quả với các phương pháp khác ?

3 PHẠM VI NGHIÊN CỨU

Đề tài này tập trung vào việc phân tích và dự báo doanh số bán lẻ dựa trên tập dữ liệu được cung cấp, bao gồm thông tin về doanh số hàng ngày của các mặt hàng tại các cửa hàng cụ thể, thông tin chi tiết về cửa hàng, giá dầu hàng ngày và thông tin về các ngày lễ/sự kiện.

Phạm vi dữ liệu nghiên cứu bao gồm:

- Dữ liệu doanh số bán hàng từ năm 2013 đến năm 2017.
- Thông tin về 54 cửa hàng.
- Thông tin về 33 nhóm mặt hàng (families).
- Dữ liệu giá dầu hàng ngày.
- Dữ liệu về các ngày lễ và sự kiện tại Ecuador.

Nghiên cứu sẽ tập trung vào việc xây dựng và đánh giá mô hình dự báo trên tập dữ liệu đã cho, không mở rộng ra các yếu tố ngoại sinh khác ngoài những dữ liệu được cung cấp. Mô hình được phát triển sẽ là mô hình dự báo chuỗi thời gian đa biến (multivariate time series forecasting) ở mức độ chi tiết theo từng cửa hàng và từng mặt hàng.

4 PHƯƠNG PHÁP NGHIÊN CỨU

- Nghiên cứu này sử dụng kết hợp các phương pháp phân tích dữ liệu và mô hình học máy để đạt được các mục tiêu đã đề ra. Cụ thể, phương pháp nghiên cứu bao gồm:

4.1 Phân tích định tính và thống kê mô tả:

- Sử dụng các kỹ thuật thống kê mô tả để khám phá và hiểu rõ cấu trúc dữ liệu, xu hướng doanh số theo thời gian (ngày, tuần, tháng, năm), sự phân bố doanh số theo cửa hàng và nhóm mặt hàng, cũng như tác động ban đầu của các yếu tố ngoại sinh như ngày lễ/sự kiện và giá dầu.

- Trực quan hóa dữ liệu thông qua các biểu đồ (biểu đồ đường, biểu đồ cột, box plot) để nhận diện các mẫu hình, tính thời vụ, xu hướng và các điểm bất thường trong dữ liệu doanh số và các biến liên quan.

4.2 Tiền xử lý dữ liệu

- Chia tập train thành 2 phần nhỏ hơn : train và validation (tỷ lệ 80/20) để huấn luyện và đánh giá mô hình trước khi dự báo cho tập test.
- Xử lý dữ liệu thiếu (ví dụ: điền giá trị cho cột giá dầu) bằng các phương pháp phù hợp (ví dụ: forward fill, backward fill).
- Tích hợp các nguồn dữ liệu khác nhau (train, validation, test, stores, oil, holidays) dựa trên các cột chung (date, store_nbr, city, state).
- Xử lý các đặc điểm dữ liệu cụ thể như ngày lễ/sự kiện bằng cách tạo các biến dummy hoặc biến đặc trưng mới dựa trên phạm vi và loại ngày lễ.
- Tạo các đặc trưng kỹ thuật (feature engineering) từ cột ngày tháng (ví dụ: tháng, ngày trong tuần, cuối tuần).
- Áp dụng mã hóa (encoding) cho các biến phân loại (ví dụ: Label Encoding cho store_nbr và family, One-Hot Encoding cho các biến ngày lễ/sự kiện, loại cửa hàng, nhóm thành phố).
- Chuẩn hóa hoặc biến đổi dữ liệu (ví dụ: áp dụng log1p cho cột sales và các biến lag) để cải thiện hiệu suất mô hình.

4.3 Xây dựng mô hình dự báo hai bước

- Bước 1 (Phân loại): Xây dựng một mô hình phân loại (XGBoost Classifier) để dự đoán khả năng một danh mục sản phẩm có doanh thu trong một ngày cụ thể hay không ('has_sales' = 1) dựa trên các đặc trưng đã được tiền xử lý.

- Bước 2 (Hồi quy): Xây dựng một mô hình hồi quy (XGBoost Regressor, LightGBM Regressor, Linear Regression) trên tập dữ liệu con chỉ chứa các trường hợp được dự đoán

là có bán hàng ('has_sales' = 1) để dự đoán giá trị doanh số thực tế (trên thang đo đã biến đổi logarit).

- Kết hợp kết quả từ hai mô hình: Đối với mỗi dự báo, nếu mô hình phân loại dự đoán không có bán hàng, doanh số dự báo cuối cùng sẽ là 0. Nếu mô hình phân loại dự đoán có bán hàng, doanh số dự báo cuối cùng sẽ là kết quả từ mô hình hồi quy (sau khi chuyển đổi ngược về thang đo gốc).

- Đánh giá mô hình:

- Sử dụng các chỉ số đánh giá phù hợp cho cả mô hình phân loại (ví dụ: Accuracy, Precision, Recall, F1-score, ROC AUC) và mô hình hồi quy (ví dụ: MSE, RMSE, MAE, R2, RMSLE).
- Đánh giá hiệu suất của mô hình hai bước trên tập validation bằng chỉ số Root Mean Squared Logarithmic Error (RMSLE), đây là chỉ số chính được sử dụng trong cuộc thi Kaggle cho bài toán này.
- Kiểm tra hiện tượng underfitting/overfitting bằng cách so sánh hiệu suất của mô hình trên tập huấn luyện và tập validation.

4.4 Cơ sở logic, toán học và công thức của các mô hình và chỉ số

4.4.1. Các mô hình

a) XGBoost (Extreme Gradient Boosting)

- XGBoost là một thuật toán boosting cây quyết định (gradient boosting) được tối ưu hóa. Nó xây dựng mô hình một cách tuần tự, mỗi cây mới cố gắng sửa chữa lỗi của các cây trước đó.

Cơ sở:

- Boosting: Ý tưởng cốt lõi là kết hợp các "weak learners" (mô hình yếu, thường là cây quyết định nông) để tạo ra một "strong learner" (mô hình mạnh). Mỗi weak learner được huấn luyện trên phần dữ liệu mà mô hình kết hợp hiện tại đang dự đoán sai.
- Gradient Boosting: Sử dụng gradient descent để tối ưu hóa hàm mất mát. Ở mỗi bước, thuật toán xây dựng một cây mới để dự đoán gradient (đạo hàm) của hàm mất mát đối với dự đoán hiện tại.
- Regularization: XGBoost bao gồm các kỹ thuật regularization (L1 và L2) để tránh overfitting, giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới.
- Tree Pruning: Cắt tỉa cây sau khi xây dựng để cải thiện khả năng tổng quát hóa.
- Handle Missing Values: Có khả năng xử lý các giá trị thiếu một cách tự nhiên.

Giải thích:

- XGBoost hoạt động bằng cách bắt đầu với một dự đoán ban đầu (thường là giá trị trung bình của biến mục tiêu). Sau đó, nó lặp đi lặp lại việc thêm các cây quyết định mới. Mỗi cây mới được huấn luyện để dự đoán phần "residual" (phần sai sót) của dự đoán hiện tại. Quá trình này tiếp tục cho đến khi đạt được tiêu chí dừng hoặc số lượng cây tối đa được xây dựng. XGBoost sử dụng các cải tiến như xử lý song song, phân phối dữ liệu và các chức năng mục tiêu/đánh giá tùy chỉnh để đạt hiệu suất cao.

b) LightGBM (Light Gradient Boosting Machine)

- LightGBM cũng là một framework gradient boosting cây quyết định, được thiết kế để nhanh hơn và hiệu quả hơn XGBoost, đặc biệt trên các tập dữ liệu lớn.

Cơ sở:

- Gradient-based One-Side Sampling (GOSS): LightGBM sử dụng GOSS để lấy mẫu dữ liệu hiệu quả hơn. Nó giữ lại tất cả các instance có gradient lớn (những instance mà mô hình hiện tại dự đoán sai nhiều) và lấy mẫu ngẫu nhiên một phần nhỏ các instance có gradient nhỏ. Điều này giúp tập trung vào các instance khó học mà không làm mất đi quá nhiều thông tin từ các instance dễ học.
- Exclusive Feature Bundling (EFB): Kỹ thuật này nhóm các đặc trưng "độc quyền" (exclusive features - những đặc trưng hiếm khi có giá trị khác 0 cùng lúc) thành một bó duy nhất để giảm số lượng đặc trưng, tăng tốc độ huấn luyện.
- Leaf-wise (Best-first) Tree Growth: Không như các thuật toán phát triển cây theo cấp độ (level-wise), LightGBM phát triển cây theo lá (leaf-wise). Nó chọn lá có mức giảm mất mát lớn nhất để phát triển tiếp, có thể dẫn đến cây mất cân bằng nhưng thường đạt được độ chính xác cao hơn với số lượng lá/nút tương đương.

Giải thích:

- LightGBM cũng xây dựng mô hình tuần tự như XGBoost. Tuy nhiên, các kỹ thuật GOSS và EFB giúp nó xử lý dữ liệu lớn nhanh hơn và sử dụng ít bộ nhớ hơn. Phương pháp phát triển cây leaf-wise có thể dẫn đến overfitting trên các tập dữ liệu nhỏ, nhưng trên các tập dữ liệu lớn, nó thường hiệu quả hơn.

c) Linear Regression

- Linear Regression là một mô hình hồi quy tuyến tính đơn giản, giả định mối quan hệ tuyến tính giữa các biến độc lập (features) và biến phụ thuộc (target).

Cơ sở:

- Mục tiêu của Linear Regression là tìm ra một đường (trong trường hợp một biến độc lập) hoặc một siêu phẳng (trong trường hợp nhiều biến độc lập) phù hợp nhất với dữ liệu. Mô hình có dạng:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

Trong đó:

- y: Biến phụ thuộc (sales).
- x_i: Các biến độc lập (features).
- β_0: Hệ số chặn (intercept).
- β_i: Hệ số (coefficients) cho biến độc lập xi, biểu thị mức độ ảnh hưởng của xi lên y.
- ε: Sai số (error term), biểu thị phần biến thiên của y không được giải thích bởi mô hình.

Giải thích:

Mô hình tìm cách ước lượng các hệ số βi sao cho tổng bình phương sai số (Sum of Squared Errors - SSE) giữa giá trị thực tế và giá trị dự đoán là nhỏ nhất. Phương pháp phổ biến nhất để tìm các hệ số này là Ordinary Least Squares (OLS).

4.4.2 Công thức các thông số đánh giá mô hình

a) Mô hình phân loại (đánh giá 'has_sales')

- Accuracy (Độ chính xác): Tỷ lệ các trường hợp dự đoán đúng (cả có sales và không có sales) trên tổng số trường hợp.

$$Accuracy = \frac{Số\ lượng\ dự\ đoán\ đúng}{Tổng\ số\ lượng\ dự\ đoán}$$

- Precision (Độ chính xác dương): Tỷ lệ các trường hợp dự đoán là có sales thực sự là có sales. Quan trọng khi chi phí của việc dự đoán sai là có sales (False Positive) cao.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- Recall (Độ nhạy/Tỷ lệ True Positive): Tỷ lệ các trường hợp thực sự có sales được mô hình dự đoán đúng là có sales. Quan trọng khi chi phí của việc bỏ sót trường hợp có sales (False Negative) cao.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- F1 Score: Trung bình điều hòa của Precision và Recall. Cân bằng giữa Precision và Recall, hữu ích khi phân bố lớp không đồng đều.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- ROC AUC Score (Area Under the Receiver Operating Characteristic Curve): Đo lường khả năng phân biệt giữa các lớp dương và âm của mô hình. Giá trị càng gần 1, khả năng phân biệt càng tốt.

b) Mô hình hồi quy (đánh giá 'sales')

- MSE (Mean Squared Error - Sai số bình phương trung bình): Trung bình của bình phương sai số giữa giá trị thực tế và giá trị dự đoán. Giá trị nhỏ hơn cho thấy mô hình tốt hơn.

$$MSE = \frac{1}{\mathrm{n}}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

Trong đó yi là giá trị thực tế, $\widehat{y_i}$ là giá trị dự đoán, và n là số lượng mẫu.

- RMSE (Root Mean Squared Error - Căn bậc hai của Sai số bình phương trung bình): Căn bậc hai của MSE. Có cùng đơn vị với biến mục tiêu, dễ diễn giải hơn MSE. Giá trị nhỏ hơn cho thấy mô hình tốt hơn.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

- MAE (Mean Absolute Error - Sai số tuyệt đối trung bình): Trung bình của giá trị tuyệt đối của sai số giữa giá trị thực tế và giá trị dự đoán. Ít nhạy cảm với các giá trị ngoại lai hơn MSE/RMSE. Giá trị nhỏ hơn cho thấy mô hình tốt hơn.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- R2 Score (Coefficient of Determination): Biểu thị tỷ lệ phương sai trong biến phụ thuộc có thể được giải thích bởi các biến độc lập trong mô hình. Giá trị nằm trong khoảng từ 0 đến 1 (hoặc có thể âm nếu mô hình rất tệ). Giá trị gần 1 cho thấy mô hình phù hợp tốt với dữ liệu.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Trong đó SSE là tổng bình phương sai số, SST là tổng bình phương toàn bộ, và $\bar{y}$ là giá trị trung bình của biến phụ thuộc.

- RMSLE (Root Mean Squared Logarithmic Error - Căn bậc hai của Sai số logarit bình phương trung bình): Tương tự như RMSE nhưng áp dụng phép biến đổi logarit cho cả giá trị thực tế và dự đoán. Thường được sử dụng khi biến mục tiêu có phân phối bị lệch phải (như sales) hoặc khi các sai số tương đối quan trọng hơn các sai số tuyệt đối. Nó giảm thiểu ảnh hưởng của các sai số lớn trên các giá trị lớn.

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(1 + y_i) - \log(1 + \hat{y}_i))^2}$$

Cộng 1 vào y_i và $\hat{y}_i$ để xử lý trường hợp giá trị bằng 0.

c) Underfitting và Overfitting

- Trong học máy, mục tiêu là xây dựng một mô hình không chỉ hoạt động tốt trên dữ liệu huấn luyện mà còn có khả năng tổng quát hóa (generalize) tốt trên dữ liệu mới, chưa từng thấy. Underfitting và Overfitting là hai vấn đề phổ biến liên quan đến khả năng tổng quát hóa của mô hình.

Underfitting (Thiếu khớp)

- Giải thích:

  - Underfitting xảy ra khi mô hình quá đơn giản để nắm bắt được mối quan hệ phức tạp trong dữ liệu. Mô hình không học đủ từ dữ liệu huấn luyện, dẫn đến hiệu suất kém không chỉ trên dữ liệu huấn luyện mà còn trên dữ liệu validation và test.

- Dấu hiệu:

  - Hiệu suất (ví dụ: độ chính xác, R2 score) thấp trên cả tập huấn luyện và tập validation.
  - Mô hình quá đơn giản (ví dụ: sử dụng Linear Regression cho dữ liệu có mối quan hệ phi tuyến tính rõ rệt, sử dụng cây quyết định quá nông).

- Nguyên nhân:

  - Mô hình quá đơn giản so với độ phức tạp của dữ liệu.
  - Thiếu đặc trưng quan trọng.
  - Dữ liệu huấn luyện không đủ lớn hoặc không đại diện cho vấn đề.
  - Regularization quá mạnh.

Overfitting (Quá khớp)

- Giải thích:

  - Overfitting xảy ra khi mô hình quá phức tạp và học "noise" (nhiễu) hoặc các chi tiết không quan trọng trong dữ liệu huấn luyện. Mô hình hoạt động rất tốt trên dữ liệu huấn luyện nhưng kém hiệu quả trên dữ liệu validation và test vì nó đã học thuộc lòng dữ liệu huấn luyện thay vì học được các mẫu hình tổng quát.

- Dấu hiệu:

  - Hiệu suất rất cao trên tập huấn luyện.
  - Hiệu suất thấp hơn đáng kể trên tập validation và test so với tập huấn luyện.
  - Mô hình quá phức tạp (ví dụ: sử dụng cây quyết định rất sâu, mạng nơ-ron với quá nhiều lớp hoặc nơ-ron).

- Nguyên nhân:

  - Mô hình quá phức tạp so với độ phức tạp của dữ liệu.
  - Dữ liệu huấn luyện quá ít.
  - Thiếu regularization.
  - Lựa chọn đặc trưng kém (bao gồm cả noise).

- Kiểm tra Underfit/Overfit:

Việc so sánh hiệu suất của mô hình trên tập huấn luyện và tập validation (hoặc test) là cách phổ biến nhất để kiểm tra underfitting và overfitting.

- Nếu hiệu suất thấp trên cả hai tập: Khả năng cao là underfitting.
- Nếu hiệu suất cao trên tập huấn luyện nhưng thấp hơn đáng kể trên tập validation/test: Khả năng cao là overfitting.
- Nếu hiệu suất tương tự và chấp nhận được trên cả hai tập: Mô hình có khả năng tổng quát hóa tốt.

5 KẾT QUẢ HUẤN LUYỆN MÔ HÌNH

5.1 Triển khai và dự báo:

- Áp dụng quy trình tiền xử lý và mô hình hai bước đã huấn luyện trên tập dữ liệu test để tạo ra các dự báo doanh số cuối cùng.
- Trình bày kết quả dự báo dưới dạng bảng và trực quan hóa để phân tích xu hướng dự báo.
- Công cụ và Phần mềm:
    - Ngôn ngữ lập trình: Python
    - Các thư viện chính: pandas (xử lý dữ liệu), numpy (tính toán số học), scikit-learn (tiền xử lý, mô hình hóa cơ bản), xgboost, lightgbm (mô hình gradient boosting), matplotlib, seaborn (trực quan hóa).
    - Môi trường phát triển: Google Colab.

5.2 Kết quả và nhận xét

5.2.1 Kết quả

a) Mô hình phân loại

- Accuracy: 0.9343
- Precision: 0.9690
- Recall: 0.9531
- F1 Score: 0.9610
- ROC AUC Score: 0.9705

b) Linear Regression

- Kết quả:
    - MSE: 311331.97
    - RMSE: 557.97
    - MAE: 132.05
    - R2 Squared: 0.85
- Nhận xét:

- RMSE cao, nhiều giá trị âm, không phù hợp cho dữ liệu có phân bố lệch
- R-square khá thấp so với 2 mô hình kia, phù hợp với baseline nhưng không đủ để dự báo chính xác.

c) XGBoost

- Kết quả:
  - MSE: 1154708.62
  - RMSE: 393.33
  - MAE: 105.81
  - R2 Squared: 0.93
- Nhận xét:
  - Cải thiện đáng kể so với Linear Regression.
  - Tuy nhiên vẫn bị ảnh hưởng bởi outliers, một số giá trị dự báo âm.

d) LightGBM

- Kết quả:
  - MSE: 158242.47
  - RMSE: 397.80
  - MAE: 104.66
  - R2 Squared: 0.93
- Nhận xét :
  - Cho kết quả tốt, dự báo ổn định hơn, không tạo ra giá trị âm.
  - Khả năng nắm bắt tốt biến mùa vụ và đặc trưng phi tuyến

e) Mô hình 2 bước

- Kết quả:
  - Accuracy (mô hình phân loại): 0.9348
  - RMSLE: 0.4822
  - MSE: 131455.39
  - RMSE: 362.57
  - MAE: 90
  - R2 Squared: 0.93
- Nhận xét :
  - Cho kết quả tốt hơn hẳn, dự báo ổn định hơn, xử lí được các giá trị gần với 0
  - Sai số từ dự báo giảm đi khá nhiều so với việc dùng mô hình 1 bước

f) Kiểm tra overfit/underfit

- Chênh lệch RMSLE (Validation - Training): 0.0453
- Chênh lệch RMSE (Validation - Training): 25.76
- Chênh lệch MAE (Validation - Training): 23.03

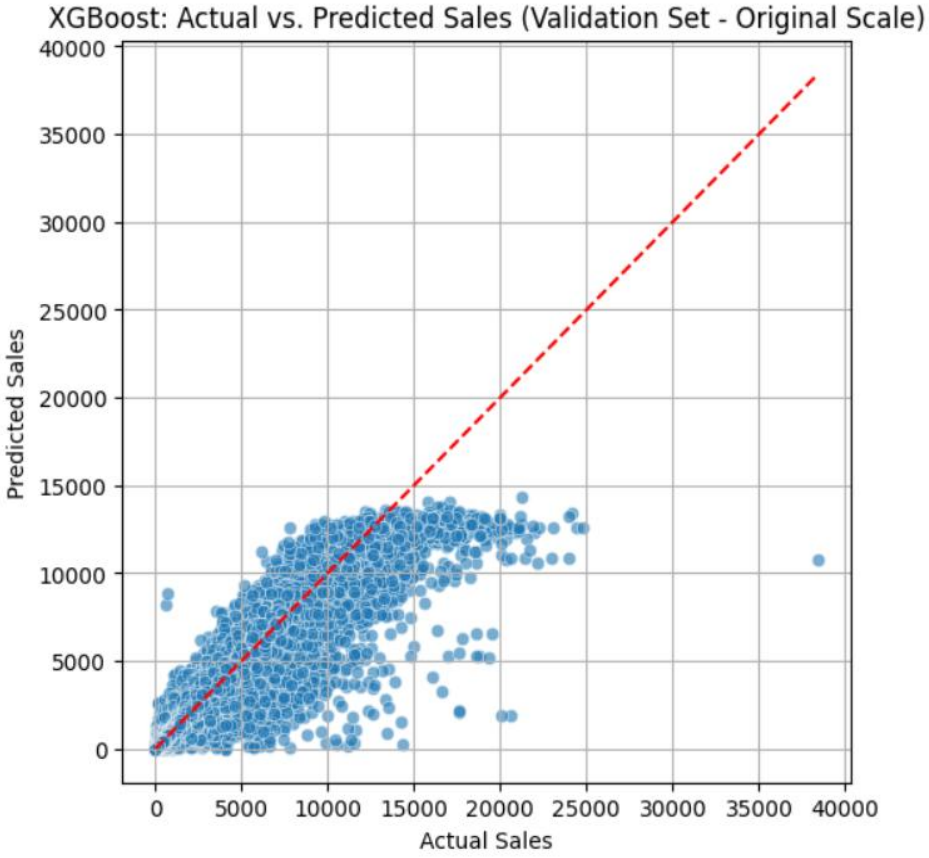
=> Sự khác biệt giữa tập huấn luyện và validation là chấp nhận được.

5.2.2 Trực quan kết quả dự báo và nhận xét

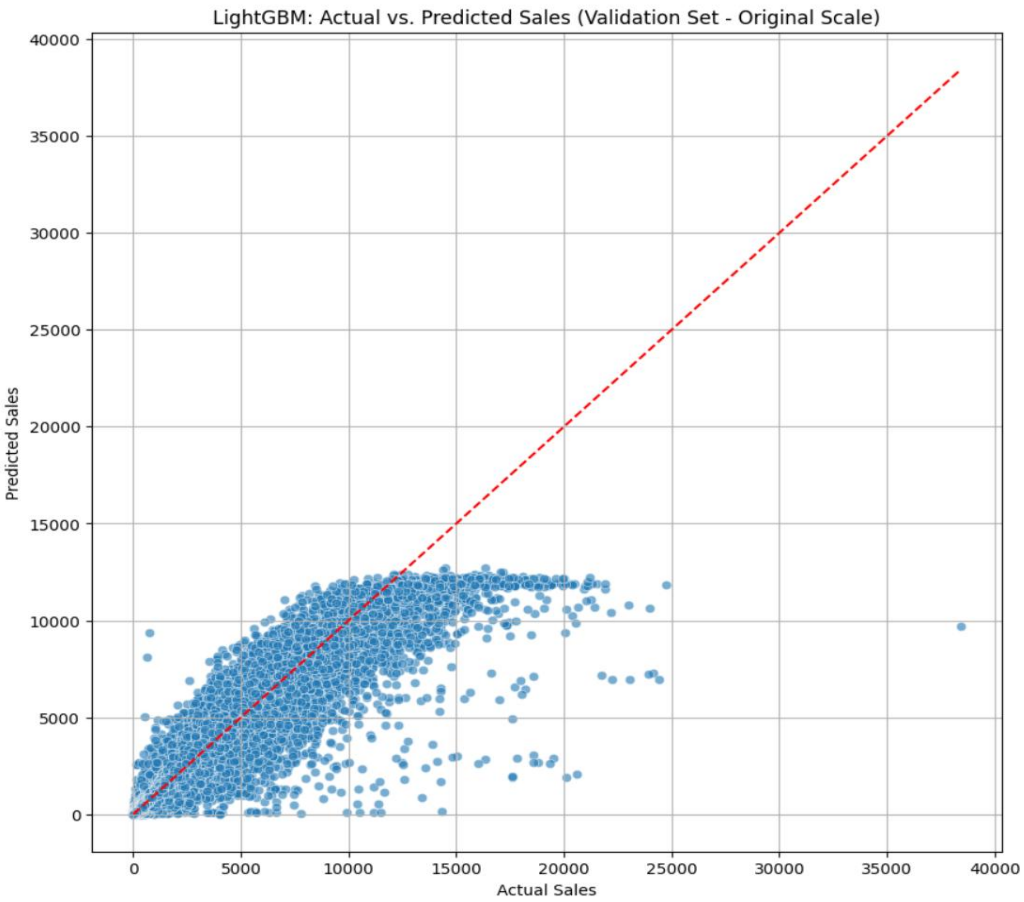
a) Trực quan kết quả dự báo của tập validation

- Biểu đồ Actual vs Predicted Sales (test set):

- Mô hình XGBoost:

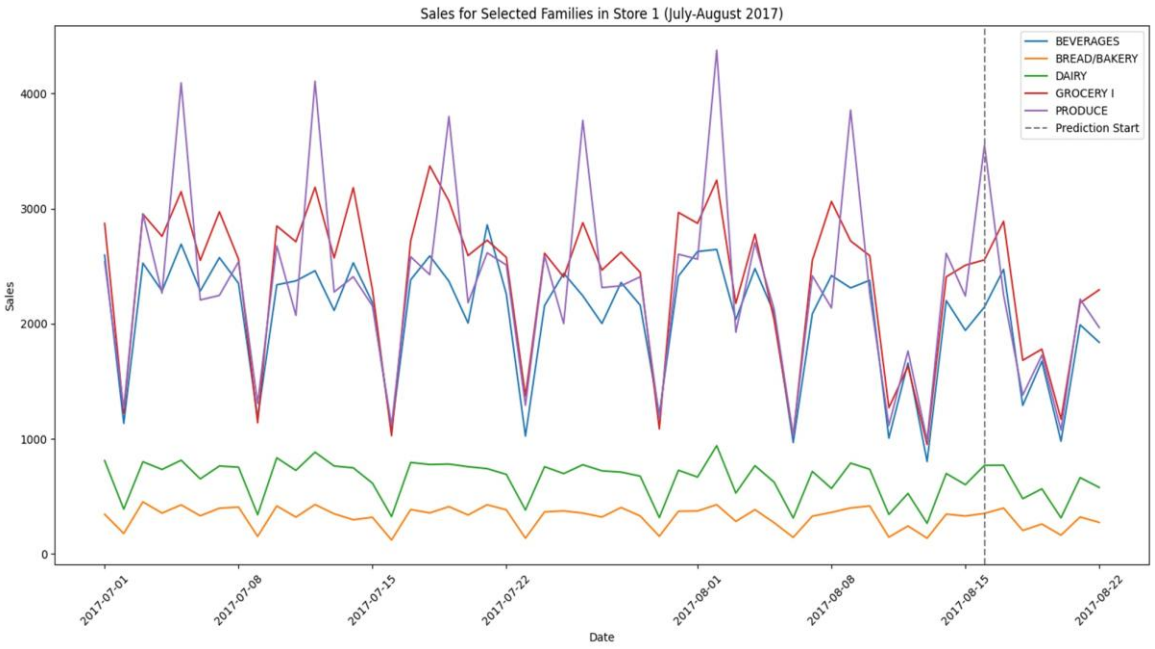


- Mô hình LightGBM



b) Trực quan Kết quả dự báo cho tập test

- Dự báo được doanh số cho từng mặt hàng trong từng cửa hàng vào các ngày 16 – 31/8/2017
- Biểu đồ dưới cho thấy doanh thu dự báo cho cửa hàng 1 trên 1 số mặt hàng:



- Nhận xét : Mô hình dự đoán sales cho tập test bám sát yếu tố mùa vụ trong quá khứ, đặc biệt là theo tháng.

6 KẾT LUẬN VÀ GIẢI PHÁP ĐỀ XUẤT

6.1 Kết luận

- LightGBM là mô hình hiệu quả ít bị ảnh hưởng bởi các giá trị ngoại lai trong dự báo doanh thu cho tập dữ liệu Store Sales.
- Việc xử lý dữ liệu ngoại lai và phân phối lệch phải là bước then chốt.
- Mô hình hai bước giúp cải thiện độ chính xác trong dự báo khi dữ liệu có nhiều giá trị 0.
- Vấn đề tích lũy sai số: Mô hình có thể có độ chính xác tốt ở cấp độ vi mô (từng mặt hàng, từng cửa hàng, từng ngày), nhưng các sai số nhỏ khi cộng dồn lại đã gây ra sự sai lệch lớn ở cấp độ vĩ mô (tổng doanh số hàng ngày). Việc biểu đồ cho thấy dự báo thấp hơn nhưng tổng lại cao hơn là do những sai số dương nhỏ ở các giao dịch lớn

6.2 Hướng phát triển

- Áp dụng các mô hình Deep Learning (LSTM, Transformer): Các mô hình này rất mạnh trong việc xử lý dữ liệu chuỗi thời gian (time-series data) và có khả năng

nắm bắt các xu hướng phức tạp trong dài hạn.

- Thêm dữ liệu ngoài (thời tiết, macro, Google Trends): Việc kết hợp dữ liệu ngoài như thời tiết (mưa/nắng ảnh hưởng đến doanh số bán lẻ), dữ liệu vĩ mô (tình hình kinh tế) và xu hướng tìm kiếm trên Google (Google Trends) có thể cung cấp thêm thông tin quý giá cho mô hình, giúp tăng độ chính xác đáng kể.

- Kết hợp nhiều mô hình (ensemble): Kết hợp kết quả từ nhiều mô hình khác nhau. Ví dụ, bạn có thể lấy trung bình của dự báo từ một mô hình hồi quy (ví dụ LightGBM) và một mô hình chuỗi thời gian (như Prophet) để có kết quả dự báo ổn định hơn.

6.3 Những điều đã làm được tốt hơn so với các bài làm khác

- Chia train/ valid từ đầu , không bị data leakage
- Xử lý triệt để các ngày có event/holiday theo phạm vi ảnh hưởng thay vì chỉ định danh ngày đó có event/holiday hay không : Local, Regional, National
- Xử lý các trường hợp trong 1 ngày mà có nhiều hơn 1 event/holiday.
- Các chỉ số đánh giá mô hình tốt: R2 squared: 0.93, không bị overfit/underfit
- Cách xử lý sale lag 7,30, quý chính xác.
- Phân cụm các thành phố thành các điểm nóng doanh số.
- Sử dụng mô hình hai bước thay vì chỉ dung mỗi Linear ngay từ đầu (xử lý dự báo các ngày sale bằng 0 tốt hơn)