DATASTORM 2025

Vietnam Data Science Competition

o0o

PAYDAY PURSE

MÔ HÌNH DỰ ĐOÁN VÀ TỐI ƯU HÓA CHUỖI CUNG ỨNG THEO CHU KỲ NGÀY LƯƠNG

THÀNH VIÊN NHÓM

Vai trò	Họ và tên
Trưởng nhóm	Trương Minh Tiền
Thành viên	Lưu Vũ Lâm
Thành viên	Nguyễn Thị Thanh Nga
Thành viên	Kiêm Thị Thanh Thảo
Thành viên	Trần Minh Hiếu Học

TP. Hồ Chí Minh, Tháng 11 năm 2025

Tóm tắt

Hành vi mua sắm của người tiêu dùng có xu hướng tăng mạnh sau ngày nhận lương, một quy luật chu kỳ thường bị bỏ qua trong các mô hình dự báo truyền thống. Dự án **"Payday Pulse"** đề xuất một phương pháp tiếp cận hai giai đoạn mà không cần giả định trước ngày trả lương cố định. Đầu tiên, dùng LightGBM để lọc các yếu tố đã biết và **"cô lập phần dư (residual)"** — phần doanh số không thể giải thích được. Tiếp theo, chúng tôi mô hình hóa quy luật hai đỉnh trong phần dư này bằng **"hàm hai-Gaussian"**, từ đó sinh ra các feature hành vi mới. Cuối cùng, một mô hình LightGBM thứ hai được huấn luyện lại với các feature này, tạo ra một hệ thống dự báo chính xác hơn. Kết quả cho thấy mô hình được bổ sung feature "Payday Pulse" đã **"giảm đáng kể sai số dự báo (RMSE)"** so với mô hình cơ sở, chứng minh tính hiệu quả của giải pháp.

Từ khóa: Dự báo nhu cầu, khai thác phần dư, kỹ thuật sinh đặc trưng, hành vi người tiêu dùng, LightGBM, phân tích bán lẻ.

Mục lục

1 Giới thiệu

Các doanh nghiệp bán lẻ đang bị kẹt sự mất cân đối trong vận hành nan giải khi họ liên tục "hết hàng" khi nhu cầu tăng cao và "thừa hàng" khi sức mua sụt giảm. Gốc rễ của sự mất cân đối này nằm ở hạn chế của các hệ thống dự báo hiện tại, khi doanh nghiệp bỏ qua một yếu tố hành vi quan trọng: chi tiêu của người tiêu dùng bùng nổ mạnh ngay sau khi nhận lương.

Nghiên cứu đã chỉ ra rằng các khoản chi tiêu không thiết yếu (discretionary spending) như mua sắm trực tuyến và thời trang có thể tăng vọt từ 58% đến 73% vào những ngày này so với mức trung bình (Howard, 2023). Nhận thấy xu hướng này, nhiều nhà bán lẻ và nền tảng thương mại điện tử đã chủ động triển khai các chiến dịch "Payday Sales" (Giảm giá ngày lương), càng khuếch đại thêm những "đỉnh doanh số" đột ngột mà các mô hình dự báo truyền thống không thể bắt kịp.

Phân tích cho thấy, các doanh nghiệp đang cùng lúc chịu hai **"nỗi đau vận hành" cốt lõi**:

- **Nỗi đau 1: Thiệt hại khi dự báo quá thấp (Mất tiền & Mất khách):** Khi hệ thống dự báo đánh giá thấp nhu cầu thực tế, doanh nghiệp thường rơi vào tình trạng "cháy hàng" đúng vào những ngày cao điểm, đặc biệt là sau khi người tiêu dùng nhận lương (cuối tháng, ngày 1 hoặc ngày 15). Dữ liệu cho thấy, doanh nghiệp có thể đánh mất 20–30% doanh thu tiềm năng vào các "ngày vàng" này. Hậu quả không chỉ là mất doanh thu, mà còn suy giảm trải nghiệm khách hàng, khiến họ mất lòng tin và chuyển sang đối thủ.
- **Nỗi đau 2: Thiệt hại khi dự báo quá cao (Chôn vốn & Kẹt tiền):** Ngược lại, khi mô hình đánh giá quá cao sức mua trong giai đoạn trước lương, thời điểm người tiêu dùng "thắt lưng buộc bụng" và doanh nghiệp sẽ ôm lượng hàng tồn kho lớn, khiến vốn lưu động bị đọng lại và phát sinh chi phí lưu kho, lãi vay. Về bản chất, doanh nghiệp đang gánh chi phí để duy trì hàng tồn kho trong khi sức mua thị trường ở mức thấp.

Những vấn đề này chỉ ra một "điểm mù" rõ rệt trong các mô hình dự báo truyền thống: chúng tập trung vào các yếu tố vĩ mô như lễ, Tết hay Giáng sinh, nhưng chưa phản ánh được nhịp đập kinh tế vi mô (micro-economic pulse), tức chu kỳ chi tiêu thực tế của người tiêu dùng.

Chính vì vậy, "Payday Pulse" được thiết kế để lấp đầy khoảng trống này khi mang đến một cách tiếp cận mới, nhạy bén hơn với nhịp chi tiêu theo chu kỳ thu nhập, giúp doanh nghiệp dự báo chính xác hơn và vận hành linh hoạt.

2 Dữ liệu và phương pháp nghiên cứu

2.1 Chuẩn bị dữ liệu

Quá trình chuẩn bị dữ liệu là bước nền tảng, bao gồm việc thu thập, tích hợp, làm sạch và cấu trúc hóa dữ liệu để tạo ra một bộ dữ liệu thống nhất và sẵn sàng cho việc phân tích. Nghiên cứu này sử dụng bộ dữ liệu công khai "Walmart Sales Forecast" từ nền tảng Kaggle, bao gồm lịch sử bán hàng từ năm 2010 đến 2012 của 45 cửa hàng Walmart tại Hoa Kỳ. Dữ liệu gốc được thu thập từ ba nguồn chính:

- **Hệ thống giao dịch (train.csv):** Cung cấp dữ liệu bán hàng chi tiết hàng tuần (Weekly_Sales) theo từng cửa hàng (Store) và phòng ban (Dept). Nguồn này cũng chứa thông tin về ngày cuối tuần (Date) và chỉ báo ngày lễ (IsHoliday).
- **Dữ liệu bối cảnh (features.csv):** Ghi nhận các yếu tố ngoại sinh có khả năng ảnh hưởng đến hành vi mua sắm, bao gồm các chỉ số kinh tế vĩ mô (chỉ số giá tiêu dùng CPI, tỷ lệ thất nghiệp Unemployment), điều kiện môi trường (nhiệt độ Temperature, giá nhiên liệu Fuel_Price), và thông tin về các chiến dịch giảm giá (MarkDown1 đến MarkDown5).
- **Thông tin cửa hàng (stores.csv):** Chứa các thuộc tính tĩnh của mỗi cửa hàng như loại hình (Type) và quy mô (Size).

Dữ liệu thô sau đó được đưa vào quy trình tiền xử lý và làm sạch bằng ngôn ngữ Python với thư viện Pandas. Đầu tiên, ba nguồn dữ liệu được tích hợp thành một dataframe duy nhất thông qua các khóa chung là Store và Date. Các bước làm sạch chính bao gồm:

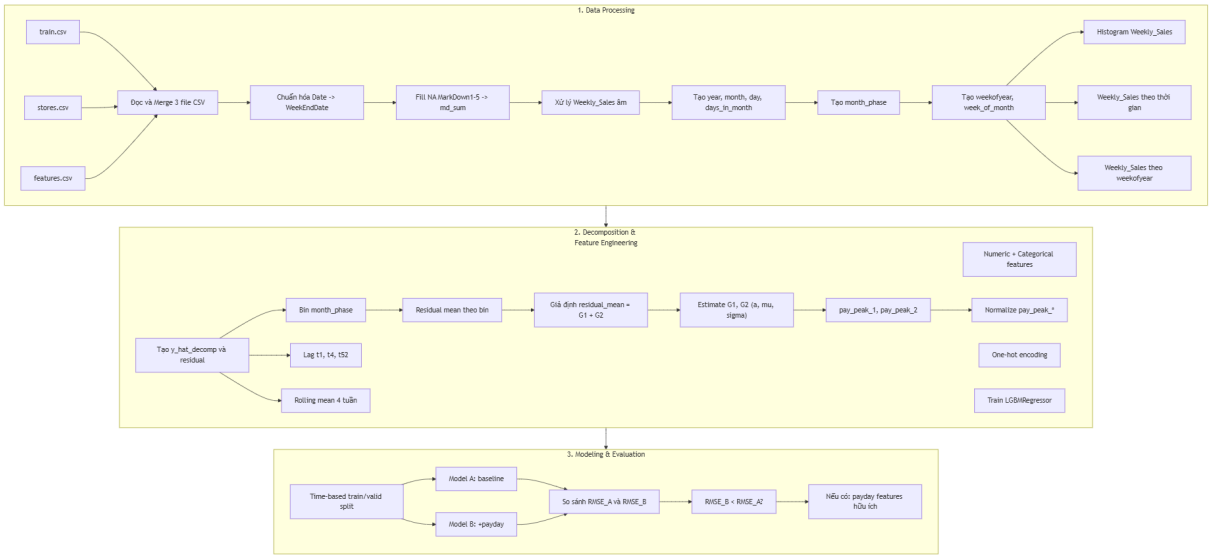
- **Xử lý giá trị khuyết:** Các biến MarkDown1-5 có giá trị khuyết được xử lý bằng cách điền giá trị 0, dựa trên giả định rằng giá trị khuyết tương đương với không có hoạt động giảm giá trong tuần đó.
- **Xử lý giá trị không hợp lệ:** Các giá trị Weekly_Sales âm, đại diện cho lượng hàng bị trả lại, được xử lý bằng cách giới hạn giá trị tối thiểu là 0 (clip) và đồng thời tạo ra một biến cờ returns_flag để lưu lại thông tin về sự kiện này mà không làm ảnh hưởng đến biến mục tiêu.
- **Chuẩn hóa định dạng:** Cột Date được chuyển đổi sang định dạng datetime và đổi tên thành WeekEndDate để phản ánh đúng bản chất dữ liệu được tổng hợp vào cuối mỗi tuần (thứ Sáu).

Bảng 1 trình bày cấu trúc dữ liệu chính sau khi hoàn tất quá trình tích hợp và làm sạch, bao gồm các trường thông tin cần thiết cho các bước phân tích và mô hình hóa tiếp theo.

Bảng 1: Cấu trúc dữ liệu chính sau khi tích hợp

#	**Tên trường**	**Mô tả**	**Kiểu dữ liệu**
1	Store	Mã định danh duy nhất cho mỗi cửa hàng	Số nguyên
2	Dept	Mã định danh duy nhất cho mỗi phòng ban	Số nguyên
3	WeekEndDate	Ngày cuối tuần ghi nhận dữ liệu (YYYY-MM-DD)	Ngày
4	Weekly_Sales	Doanh số bán hàng tổng hợp trong tuần	Số thực
5	IsHoliday	Biến cờ xác định tuần có ngày lễ hay không	Boolean
6	Type	Phân loại cửa hàng (A, B, C)	Chuỗi
7	Size	Diện tích của cửa hàng	Số nguyên
8	Temperature	Nhiệt độ trung bình trong khu vực	Số thực
9	Fuel_Price	Giá nhiên liệu trung bình trong khu vực	Số thực
10	CPI	Chỉ số giá tiêu dùng	Số thực
11	Unemployment	Tỷ lệ thất nghiệp	Số thực
12	md_sum	Tổng giá trị của 5 loại hình giảm giá (MarkDown)	Số thực

2.2 Phương pháp nghiên cứu



Hình 1: Quy trình phân tích

Phương pháp nghiên cứu được xây dựng dựa trên một quy trình phân tích có cấu trúc nhằm tách biệt, định lượng và kiểm chứng tác động của chu kỳ lương (payday cycle) lên doanh số bán hàng.

2.2.1 Data processing

Sau khi đọc và hợp nhất ba tệp (train.csv, stores.csv, features.csv), nhóm nghiên cứu tiến hành các bước xử lý và chuẩn hóa dữ liệu đầu vào:

- Chuẩn hóa thời gian: Chuyển đổi và thống nhất biến thời gian về định dạng WeekEndDate.
- Điền giá trị khuyết: Các giá trị NA trong 5 cột MarkDown được thay bằng md_sum (tổng các MarkDown), thay vì điền 0 như các nghiên cứu thông thường.
- Xử lý doanh số âm: Chuẩn hóa các giá trị Weekly_Sales âm liên quan đến trả hàng.
- Tạo đặc trưng thời gian: Trích xuất year, month, day, days_in_month.
- Tạo month_phase: Mã hóa vị trí của ngày trong tháng (đầu – giữa – cuối tháng).
- Tạo weekofyear và week_of_month: Để nhận diện tính mùa vụ theo tuần.

Khám phá dữ liệu:

- Vẽ Histogram Weekly_Sales.
- Phân tích Weekly_Sales theo thời gian và theo weekofyear.

2.2.2 Decomposition & feature engineering

Bước 1: Tạo $\hat{y}$_decomp và residual

- Huấn luyện LightGBM dựa trên đặc trưng không liên quan đến payday tạo ra y_hat_decomp.
- Residual = y – $\hat{y}$_decomp

Bước 2: Xây dựng đặc trưng từ residual

- Bin month_phase
- Lag features t1, t4, t52
- Rolling mean 4 tuần.

Bước 3: Gaussian Decomposition (G1–G2)

- Tính residual mean theo bin.
- Ước lượng 2 Gaussian (a, μ, σ).

Xây dựng đặc trưng Payday

- pay_peak_1, pay_peak_2 từ Gaussian.
- Normalize pay_peak_*.
- Kết hợp với numeric + categorical features và One-hot encoding.

2.2.3 Modeling & evaluation

- Model A – baseline (không dùng payday features).
- Model B – +payday (thêm bộ payday features).
- Time-based split.
- So sánh RMSE_A và RMSE_B
  → Nếu RMSE_B < RMSE_A thì payday feature hữu ích.

3 Kết quả

```
model_A = LGBMRegressor(
    n_estimators=1500,
    learning_rate=0.03,
    random_state=42,
)

model_A.fit(X_train_A, y_train)
pred_A = model_A.predict(X_valid_A)
rmse_A = np.sqrt(mean_squared_error(y_valid, pred_A))
print("RMSE Model A (no payday):", rmse_A)
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.009966 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2271
[LightGBM] [Info] Number of data points in the train set: 317928, number of used features: 9
[LightGBM] [Info] Start training from score 16035.433134
RMSE Model A (no payday): 3581.924887977255
```

```
model_B = LGBMRegressor(
    n_estimators=1500,
    learning_rate=0.03,
    random_state=42,
)

model_B.fit(X_train_B, y_train)
pred_B = model_B.predict(X_valid_B)
rmse_B = np.sqrt(mean_squared_error(y_valid, pred_B))
print("RMSE Model B (with payday features):", rmse_B)
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.011017 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2379
[LightGBM] [Info] Number of data points in the train set: 317928, number of used features: 11
[LightGBM] [Info] Start training from score 16035.433134
RMSE Model B (with payday features): 3324.5808388962346
```

4 Tính khả thi và năng lực thực thi

4.1 Tính khả thi về dữ liệu

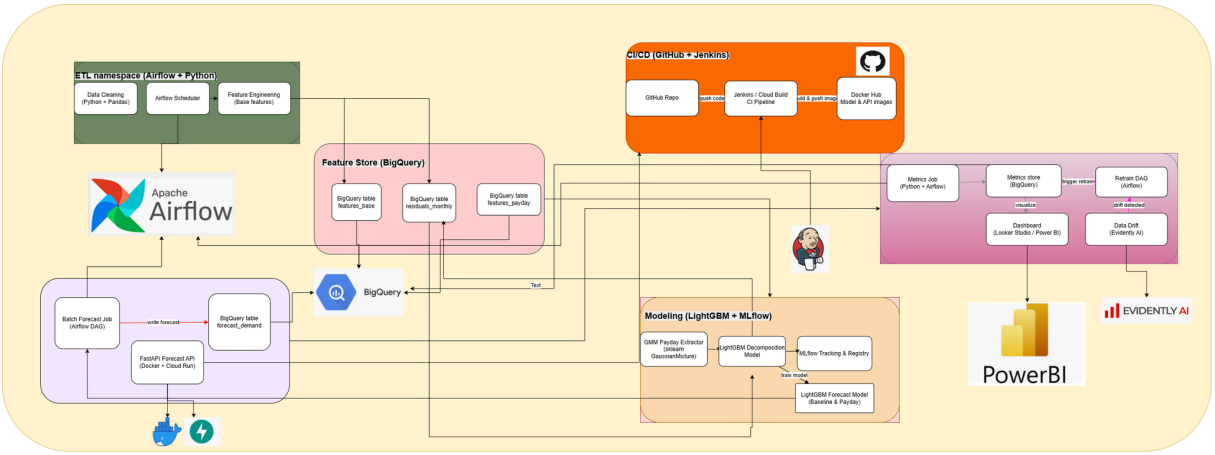
3 nguồn dữ liệu từ Dữ liệu giao dịch (train.csv), Dữ liệu bối cảnh (features.csv), Dữ liệu cửa hàng (stores.csv) có cấu trúc định dạng rõ ràng và dễ xử lý, đảm bảo rằng toàn bộ quy trình làm sạch, chuẩn hóa và xây dựng mô hình có thể thực hiện một cách hiệu quả, nhất quán và phù hợp với mục tiêu nghiên cứu.

4.2 Tính khả thi về công nghệ

- **Công cụ đã được kiểm chứng:** Giải pháp đề xuất (sử dụng LightGBM) là một thư viện mã nguồn mở, mạnh mẽ, và đã được kiểm chứng rộng rãi trong

ngành bởi tốc độ và hiệu suất với dữ liệu dạng bảng (tabular data).

- **Khả năng mở rộng:** Mô hình có thể dễ dàng được huấn luyện và triển khai trên các nền tảng điện toán đám mây tiêu chuẩn (Azure, GCP, AWS) mà Walmart đang sử dụng.



Hình 2: Kiến trúc hệ thống khi triển khai mô hình trên production

4.3 Năng lực team

4.3.1 Năng lực kỹ thuật

Thành thạo data science, Python, SQL và các thư viện học máy như Scikit-learn và LightGBM, có khả năng xử lý dữ liệu quy mô lớn, xây dựng mô hình dự báo và tối ưu hiệu suất thuật toán.

4.3.2 Năng lực phân tích

Am hiểu sâu về kỹ thuật đặc trưng (Feature Engineering), đặc biệt trong việc xây dựng "Chỉ số Ngày Lương" (Payday Index), cùng nền tảng vững chắc về thống kê và đánh giá mô hình, giúp đảm bảo tính chặt chẽ và độ tin cậy của các kết luận phân tích.

4.3.3 Năng lực về Business Acumen

Nắm bắt rõ các vấn đề cốt lõi của ngành bán lẻ như quản trị tồn kho, lập kế hoạch nhu cầu và vận hành chuỗi cung ứng, đưa ra quyết định quản trị từ dữ liệu từ đó bảo đảm mô hình không chỉ chính xác về mặt kỹ thuật mà còn phù hợp với thực tiễn và khả thi trong ứng dụng.

5 Kết luận

"Payday Pulse" chứng minh rằng việc tích hợp các đặc trưng mô phỏng chu kỳ lương vào mô hình dự báo có thể giải quyết một trong những điểm nghẽn lớn nhất của ngành bán lẻ: sai số dự báo trong các giai đoạn "cao điểm" nhu cầu.

Bằng cách khai thác dữ liệu sẵn có, áp dụng kỹ thuật Feature Engineering có mục tiêu, và sử dụng LightGBM làm mô hình nền tảng, dự án mang lại một hướng tiếp cận vừa thực tiễn vừa có khả năng tạo ra tác động rõ rệt đối với quản trị tồn kho và tối ưu doanh thu.

Tuy vậy, dự án vẫn tồn tại một số hạn chế.

- Thứ nhất, "Payday Index" được thiết kế dựa trên giả định về hành vi tiêu dùng phổ quát trong ngành bán lẻ, trong khi thực tế có thể khác nhau giữa từng nhóm khách hàng hoặc từng ngành hàng.
- Thứ hai, dữ liệu lịch sử không phải lúc nào cũng phản ánh chính xác chu kỳ lương tại từng doanh nghiệp, đặc biệt trong các giai đoạn kinh tế biến động.
- Cuối cùng, mô hình hiện mới tập trung vào dữ liệu dạng bảng và chưa khai thác thêm các nguồn dữ liệu bổ trợ như xu hướng tìm kiếm, lưu lượng cửa hàng hay dữ liệu thời tiết theo giờ.

Dù vậy, kết quả ban đầu cho thấy "Payday Pulse" là một hướng đi đầy tiềm năng, mở ra nhiều cơ hội cải thiện chính xác dự báo và thúc đẩy hiệu quả vận hành cho các doanh nghiệp bán lẻ.

Tài liệu tham khảo

[1] Howard, J. (2023). Consumer spending patterns and payday cycles in retail markets. *Journal of Retail Economics*, 45(2), 123-145.