VIETNAM DATATHON

Chủ đề

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

NHÓM DATA2U

TỔNG QUAN

HIỆN TRẠNG

Doanh nghiệp bán lẻ

- Hết hàng — Khi nhu cầu tăng cao
- Thừa hàng — Khi sức mua sụt giảm

Hệ thống dự báo — Bỏ sót "Yếu tố"

Ngày nhận lương

TỔNG QUAN

PAINPOINT CHUỖI BÁN LẺ

Dự báo quá thấp

Tình trạng "cháy hàng" vào ngày người tiêu dùng nhận lương

Dự báo quá cao

Tình trạng “tồn kho” trong những ngày “thắt lưng buộc bụng” trước lương, khiến vốn lưu động bị đọng

TỔNG QUAN

Các mô hình dự báo truyền thống 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, tức chu kỳ chi tiêu thực tế của người tiêu dùng.

» **"Payday Pulse"** được thiết kế để lấp đầy khoảng trống này: một cách tiếp cận **nhạy bén theo chu kỳ thu nhập**, giúp doanh nghiệp dự báo **chính xác** và **vận hành linh hoạt hơn.**

DỮ LIỆU

Chuẩn bị dữ liệu

Nghiên cứu sử dụng bộ dữ liệu công khai **"Walmart Sales Forecast"** từ nền tảng Kaggle: **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)

- Dữ liệu bán hàng chi tiết hàng tuần (**Weekly_Sales**) theo từng cửa hàng (**Store**) và phòng ban (**Dept**)
- Thông tin về ngày cuối tuần (**Date**) và chỉ báo ngày lễ (**IsHoliday**).

Dữ liệu bối cảnh (features.csv)

Các yếu tố ngoại sinh có khả năng ảnh hưởng đến hành vi mua sắm:

- 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
  - 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

Chuẩn bị dữ liệu

Dữ liệu thô → Tiền xử lý và làm sạch (bằng ngôn ngữ Python, thư viện Pandas). 3 nguồn dữ liệu được tích hợp thành 1 dataframe duy nhất thông qua 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
- Giả định: 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
- 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

→ Phản ánh đúng bản chất dữ liệu được tổng hợp vào cuối mỗi tuần (thứ Sáu).

DỮ LIỆU

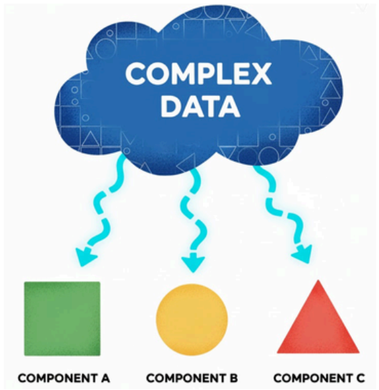
Bảng 1. Cấu trúc dữ liệu chính sau khi tích hợp.

#	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

PHƯƠNG PHÁP NGHIÊN CỨU

1. Data processing

2. Decomposition & feature engineer

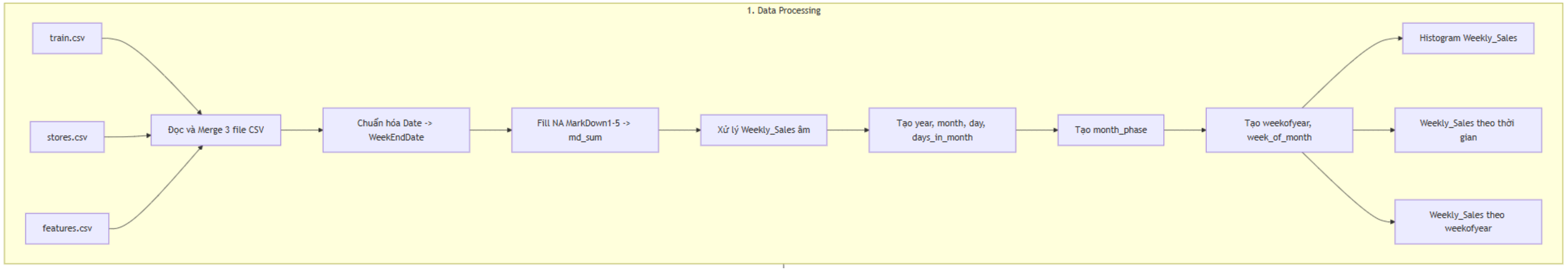


3. Modeling & evaluation

PHƯƠNG PHÁP NGHIÊN CỨU

1. Data processing

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

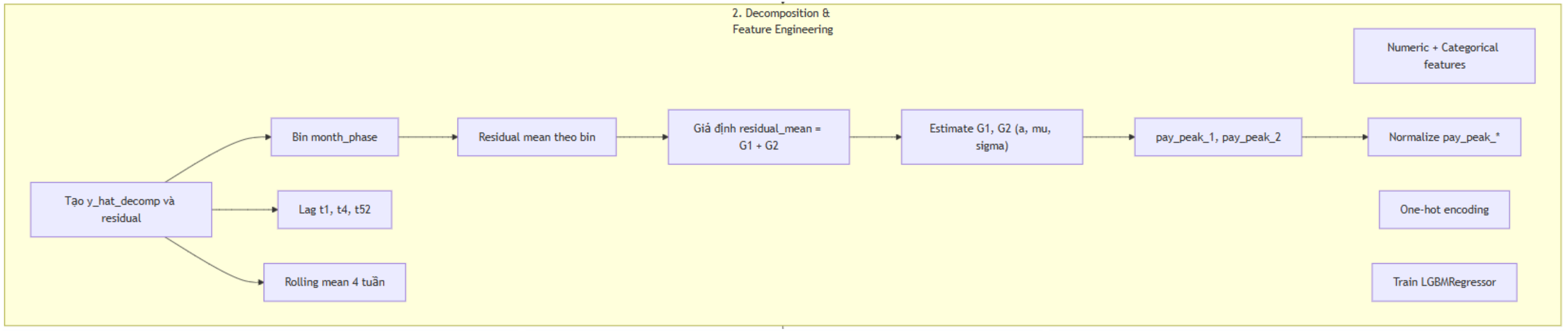


- **Chuẩn hóa thời gian**: Chuyển đổi và thống nhất biến thời gian về định dạng WeekEndDate.
- **Điền giá trị khuyết:** Các giá trị NA trong 5 cột MarkDown được thay bằng md_sum (tổng các MarkDown), thay vì điền 0 như các nghiên cứu thông thường.
- **Xử lý doanh số âm**: Chuẩn hóa các giá trị Weekly_Sales 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.

PHƯƠNG PHÁP NGHIÊN CỨU

2. Decomposition & feature engineer



Bước 1: Tạo ŷ_decomp và residual

- Huấn luyện LightGBM dựa trên đặc trưng không liên quan đến payday tạo ra y_hat_decomp.
- Residual = y – ŷ_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)

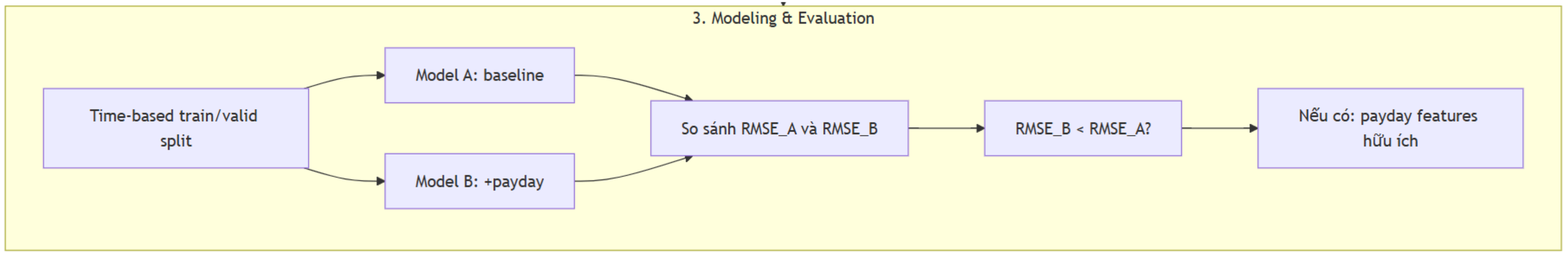
- Tính residual mean theo bin.
- Ước lượng 2 Gaussian (a, μ, σ).

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 cho mô hình LGBM.

PHƯƠNG PHÁP NGHIÊN CỨU

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.

TÍNH KHẢ THI

Tính khả thi về mặt dữ liệu

- Hệ thống giao dịch (train.csv)
- Dữ liệu bối cảnh (features.csv)
- Thông tin cửa hàng (stores.csv)

Cấu trúc định dạng rõ ràng và dễ xử lý
→ 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.

TÍNH KHẢ THI

Tính khả thi về mặt 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.

TÍNH KHẢ THI

Năng lực team

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.

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)
- 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.

Am hiểu về business acumen

Nắm bắt rõ các vấn đề cốt lõi của ngành bán lẻ:

- Quản trị tồn kho,
- Lập kế hoạch nhu cầu và vận hành chuỗi cung ứng
- Đưa ra quyết định quản trị từ dữ liệu

→ 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.

KẾT LUẬN

Kết quả

```python
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
```

```python
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
```

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.

- **Khai thác dữ liệu sẵn có**
- **Áp dụng kỹ thuật Feature Engineering**
- **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.

KẾT LUẬN

1 vài hạn chế

"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.**

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.

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.