MEDALLION DATA WAREHOUSE & PREDICTIVE MODELING

TRÊN MICROSOFT FABRIC

1. GIỚI THIỆU CHUNG

1.1. Bối cảnh chuyển đổi dữ liệu trong doanh nghiệp

Trong bối cảnh doanh nghiệp ngày càng phụ thuộc vào dữ liệu để ra quyết định, việc chỉ dừng lại ở các báo cáo mô tả đơn giản (descriptive analytics) là không còn đủ. Các tổ chức hiện đại cần một hệ thống dữ liệu có khả năng:

- Thu thập dữ liệu từ nhiều nguồn
- Chuẩn hóa và kiểm soát chất lượng dữ liệu
- Phục vụ phân tích đa chiều (OLAP, BI)
- Là nền tảng cho các mô hình dự đoán (predictive analytics)

Đặc biệt trong lĩnh vực **vận chuyển / đặt xe công nghệ (ride-hailing)**, dữ liệu giao dịch phát sinh liên tục với quy mô lớn, yêu cầu hệ thống phải **mở rộng tốt, xử lý incremental và hỗ trợ real-time hoặc near real-time analytics**.

1.2. Bối cảnh bài toán nghiệp vụ

Giả định sinh viên là thành viên của **đội Data** tại một công ty cung cấp dịch vụ vận chuyển tương tự Grab, Be hoặc Gojek. Hệ thống hiện tại của công ty có các hạn chế:

- Dữ liệu lưu trữ rời rạc, thiếu chuẩn hóa
- Chưa có kiến trúc phân tầng rõ ràng
- Báo cáo chỉ dừng ở mức doanh thu theo ngày, theo thành phố
- Chưa tận dụng dữ liệu để dự đoán giá cước (fare) hoặc tối ưu vận hành

Ban lãnh đạo đặt ra các yêu cầu chiến lược:

1. Chuẩn hóa dữ liệu theo **Medallion Architecture**
2. Xây dựng **Data Warehouse/Lakehouse Gold** phục vụ BI
3. Tận dụng dữ liệu Gold để xây dựng **mô hình Machine Learning**

2. MỤC TIÊU ĐỒ ÁN

2.1. Mục tiêu tổng quát

Xây dựng một hệ thống **end-to-end** trên Microsoft Fabric bao gồm:

- Data ingestion
- Data transformation
- Data modeling
- Analytics
- Predictive modeling

Toàn bộ pipeline phải có khả năng **mở rộng, incremental và tái sử dụng**.

2.2. Mục tiêu kỹ thuật cụ thể

- Thiết kế pipeline **Bronze → Silver → Gold**
- Áp dụng **Delta Lake** cho incremental load
- Xây dựng **Star Schema** tại Gold layer
- Phát triển mô hình dự đoán **fare_amount**
- Lưu kết quả dự đoán và metrics để trực quan hóa bằng Power BI

3. KIẾN TRÚC TỔNG THỂ HỆ THỐNG

3.1. Tổng quan kiến trúc Microsoft Fabric

Microsoft Fabric là nền tảng hợp nhất (unified analytics platform) bao gồm:

- Lakehouse
- Delta Lake
- Spark Engine
- Semantic Model
- Power BI

Trong đồ án này, Fabric đóng vai trò trung tâm để triển khai toàn bộ pipeline dữ liệu.

3.2. Medallion Architecture

Medallion Architecture chia dữ liệu thành ba tầng:

Layer	Vai trò
Bronze	Raw data, dữ liệu thô
Silver	Dữ liệu đã làm sạch, chuẩn hóa
Gold	Dữ liệu mô hình hóa cho BI & ML

Cách tiếp cận này giúp:

- Tách biệt trách nhiệm từng tầng
- Dễ kiểm soát chất lượng
- Tăng khả năng mở rộng

3.3. KIẾN TRÚC THƯ MỤC LAKEHOUSE TRÊN MICROSOFT FABRIC

Để đảm bảo dữ liệu được tổ chức khoa học, dễ quản lý và phù hợp với Medallion Architecture, Lakehouse trong đồ án được thiết kế với cấu trúc thư mục như sau:

```
mhai_lakehouse
│
├── Tables
│   ├── bronze
│   │   ├── trips_bronze
│   │   ├── customers_bronze
│   │
│   ├── silver
```

```
│   │   ├── trips_silver
│   │   ├── customers_silver
│   │
│   └── gold
│       ├── dim_customer
│       ├── dim_city
│       ├── dim_date
│       ├── dim_payment
│       ├── fact_trips
│       ├── fact_prediction
│       └── ml_metrics
│
└── Files
    ├── raw
    │   └── synthetic_data
    │
    ├── processed
    │   └── feature_engineering
    │
    └── ml_features
        ├── train
        └── test
```

3.4. Ý nghĩa kiến trúc thư mục

Thư mục Tables chứa toàn bộ bảng Delta Lake, được phân tách rõ ràng theo Bronze, Silver và Gold. Điều này giúp đảm bảo tính nhất quán giữa kiến trúc logic và kiến trúc vật lý.

Thư mục Files được sử dụng để lưu trữ dữ liệu dạng file, bao gồm dữ liệu thô, dữ liệu trung gian và các feature phục vụ machine learning. Cách tổ chức này giúp tách biệt dữ liệu bảng (structured tables) và dữ liệu file-based (semi-structured / intermediate).

4. THIẾT KẾ DỮ LIỆU BRONZE

4.1. Mục đích Bronze Layer

Bronze layer lưu trữ dữ liệu **gần với nguồn nhất**, chưa xử lý hoặc chỉ xử lý tối thiểu.

Trong đồ án, dữ liệu được **giả lập (synthetic data)** để mô phỏng dữ liệu sản xuất.

4.2. Bảng trips_bronze

Bảng `bronze.trips_bronze` chứa dữ liệu giao dịch chuyến đi:

- trip_id
- customer_id
- city
- segment
- distance_km
- duration_min
- base_fare
- surge_multiplier
- fare_amount

- payment_type
- trip_datetime

Đây là bảng **fact raw**, đại diện cho mỗi chuyến đi.

4.3. Mô phỏng dữ liệu incremental

Dữ liệu được sinh thành **2 batch**:

- Batch 1: 20.000 bản ghi
- Batch 2: 5.000 bản ghi mới

Việc này giúp mô phỏng kịch bản dữ liệu đến theo thời gian thực tế.

5. THIẾT KẾ SILVER LAYER

5.1. Vai trò Silver Layer

Silver layer chịu trách nhiệm:

- Làm sạch dữ liệu
- Chuẩn hóa schema
- Thêm các thuộc tính dẫn xuất (derived columns)
- Loại bỏ dữ liệu lỗi

5.2. Các bước xử lý chính

Trong file `01_pipeline_medallion.py`, Silver layer thực hiện:

- Loại bỏ bản ghi có `fare_amount <= 0`
- Loại bỏ chuyến có `distance_km <= 0`
- Chuẩn hóa thời gian
- Trích xuất đặc trưng thời gian:
    - hour
    - day_of_week
    - is_weekend

5.3. Feature phục vụ ML từ Silver

Silver layer đóng vai trò **cầu nối giữa raw data và analytics/ML**, giúp:

- Giảm khối lượng xử lý ở Gold
- Tạo feature ổn định cho ML

6. THIẾT KẾ GOLD LAYER – STAR SCHEMA

6.1. Lý do chọn Star Schema

Star Schema được lựa chọn vì:

- Tối ưu cho BI & Power BI
- Join đơn giản
- Hiệu năng cao
- Dễ hiểu với business user

6.2. Các Dimension Tables

6.2.1. dim_date

Chứa thông tin thời gian:

- date_key
- date
- year
- month
- day

6.2.2. dim_customer

- customer_key (surrogate key)
- customer_id
- segment

6.2.3. dim_city

- city_key
- city

6.2.4. dim_payment

- payment_key
- payment_type

6.3. Fact Table – fact_trips

Bảng trung tâm lưu các metric chính:

- fare_amount
- distance_km
- duration_min
- base_fare
- surge_multiplier

Fact table chỉ chứa **key và metric**, không chứa thuộc tính mô tả.

7. INCREMENTAL LOAD STRATEGY

7.1. Vấn đề incremental

Trong hệ thống thực tế, dữ liệu không đến một lần mà đến liên tục. Nếu mỗi lần load lại toàn bộ dữ liệu sẽ:

- Tốn tài nguyên
- Tăng latency
- Không scale được

7.2. Chiến lược áp dụng

Đồ án sử dụng kết hợp:

- Delta Lake MERGE
- So sánh `trip_id` lớn nhất đã tồn tại

Chỉ insert những bản ghi mới chưa tồn tại trong Gold.

7.3. Đảm bảo không trùng lặp

- `trip_id` đóng vai trò khóa tự nhiên
- MERGE với điều kiện `trip_id`
- Không update dữ liệu lịch sử

8. SEMANTIC MODEL & POWER BI

8.1. Semantic Model

Semantic model được xây dựng từ Gold layer, với:

- fact_trips ↔ dim_customer
- fact_trips ↔ dim_city
- fact_trips ↔ dim_date
- fact_trips ↔ dim_payment

Tất cả join qua surrogate key.

8.2. Báo cáo Power BI

Trang 1:

- Doanh thu theo ngày/tháng
- Doanh thu theo thành phố
- Top customer / city

Trang 2:

- Actual vs Predicted Fare
- Top error trips
- RMSE KPI

9. BÀI TOÁN MACHINE LEARNING

9.1. Lý do chọn bài toán Fare Prediction

Fare là chỉ số:

- Ảnh hưởng trực tiếp doanh thu
- Phản ánh hành vi khách hàng
- Dùng cho dynamic pricing

9.2. Target Variable

Target được chọn là:

`fare_amount`

Đây là biến liên tục → bài toán hồi quy (regression).

10. FEATURE ENGINEERING

10.1. Feature số (Numeric)

- distance_km
- duration_min
- base_fare
- surge_multiplier
- hour
- day_of_week
- is_weekend

10.2. Feature phân loại (Categorical)

- city
- segment
- payment_type

Các feature này phản ánh:

- Điều kiện địa lý
- Phân khúc khách hàng
- Hành vi thanh toán

10.3. Encoding

- StringIndexer
- OneHotEncoder

Đảm bảo mô hình học được thông tin danh mục.

11. MODEL TRAINING

11.1. Chia tập dữ liệu

- 80% training
- 20% testing

Seed cố định để đảm bảo tái lập kết quả.

11.2. Linear Regression (Baseline)

- Dễ giải thích
- Làm chuẩn so sánh

11.3. Random Forest Regression

- Bắt được quan hệ phi tuyến
- Phù hợp pricing problem

12. ĐÁNH GIÁ MÔ HÌNH

12.1. Metrics sử dụng

- RMSE
- MAE
- R^2

12.2. Kết quả & ý nghĩa

Random Forest cho RMSE thấp hơn, chứng tỏ:

- Giá cước phụ thuộc phi tuyến vào distance & surge
- Linear model chưa đủ mạnh

13. LƯU KẾT QUẢ MODEL

13.1. Bảng ml_metrics

Lưu:

- model_name
- metric_name
- metric_value
- run_datetime

13.2. Bảng fact_prediction

- trip_id
- y_true
- y_pred
- abs_error

Phục vụ phân tích sai số.

14. ỨNG DỤNG BUSINESS

- Dynamic pricing

- Ước lượng doanh thu
- Cảnh báo chuyến bất thường

15. HẠN CHẾ & GIẢ ĐỊNH

- Dữ liệu giả lập
- Không có yếu tố thời tiết, giao thông
- Không xử lý concept drift