Tackling Fraudulent Financial Transactions

Objectives

Implement Machine Learning Pipeline Adaptive to Fast Evolving Fraudulent Behaviour

1. Real-time ingestion of high-volume financial data
2. Build & maintain robust data lakes
3. Automated model training & inference
4. Continuous monitoring of model performance

Metrics

Primary Goal : Detect frauds while minimize false alarms

- F1-Score balances both objectives as our main success measure

Key Trade-offs Monitored :

- Recall (fraud detection rate) – ensure actual fraud not missed
- Precision (accuracy of alerts) – keeps investigation workload manageable

Drift Detection :

- Alerts triggered once accuracy < 0.70
- Commercial team threshold is 0.60 – this buffer provides lead time for model retraining

Data Schema

Transactions

- Core stream of transaction events
- 13M rows
- Base table

transactions_data.csv

Cards

- Card Metadata – card type, credit limit, chip presence etc.
- 6.1K rows

cards_data.csv

User Profiles

- Demographics – age, income, credit score, debt-related features
- 2K rows

users_data.csv

Merchants

- Merchant Category – merchant codes & descriptions
- 109 rows

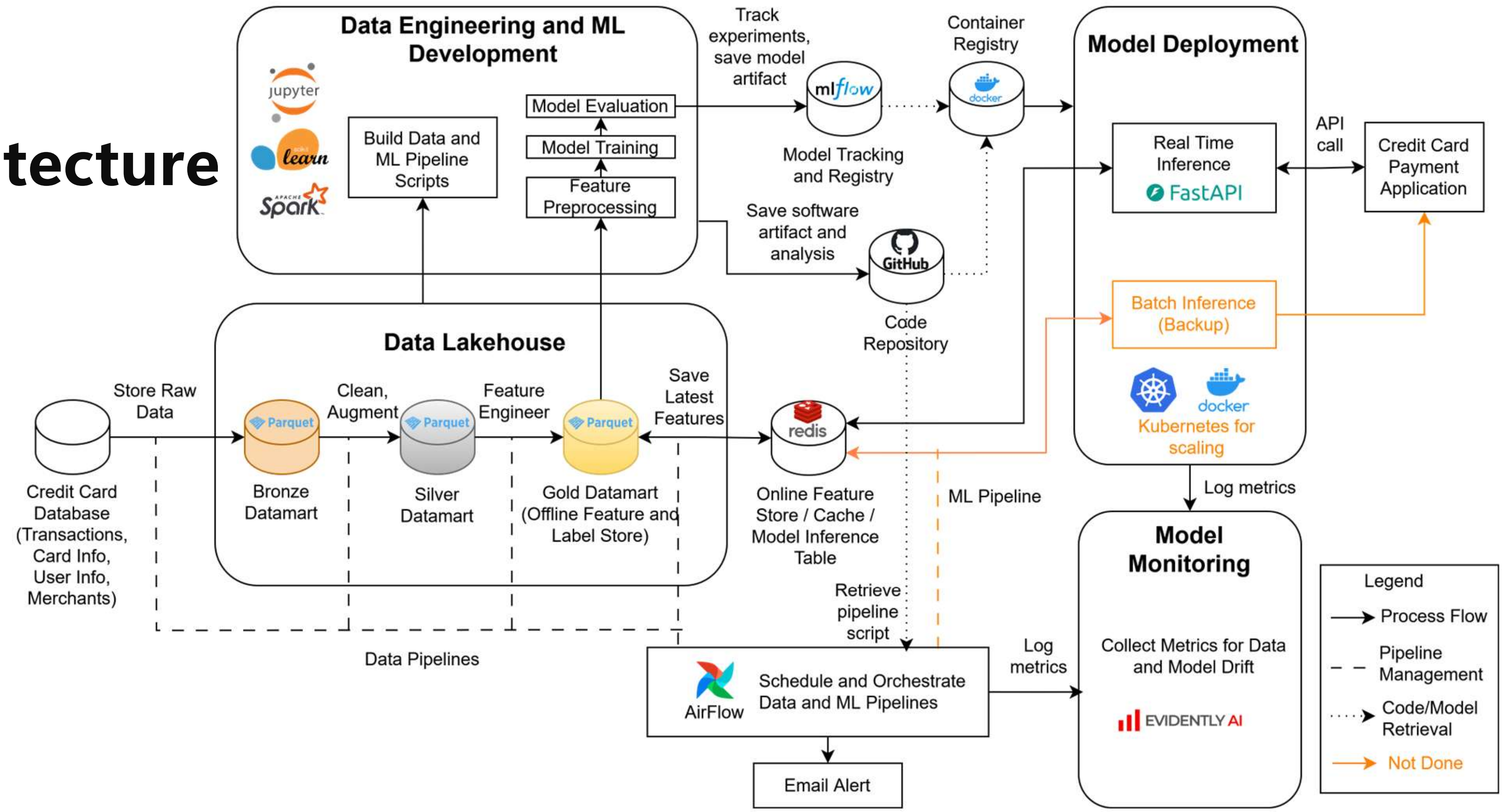
mcc_codes.json

Labels

- Classification Target (fraud vs non-fraud)
- Highly imbalanced (0.15% fraud)

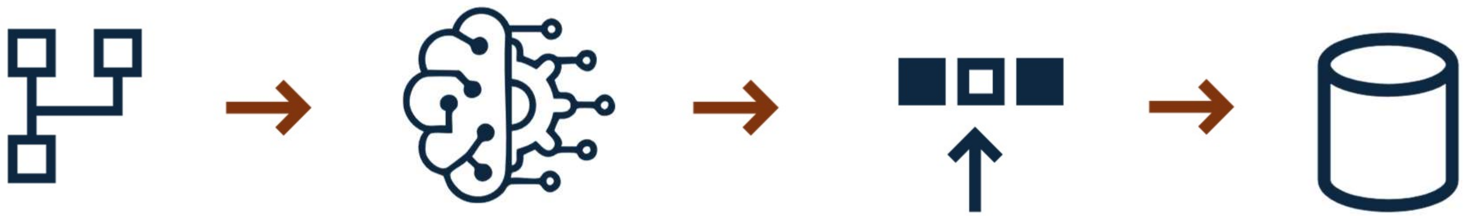
train_fraud_labels.json

Architecture



Implementation Pipeline

- Orchestrated via Apache Airflow -



Data Processing (ETL) → **Machine Learning Pipeline** → **Model Deployment & Inference** → **Model Monitoring**

3 Core Design Principles

Modularity | **Scalability** | **Abstraction**

- **Modularity**: Isolated containers for each stage
- **Scalability**: Horizontal scaling via Docker & Airflow
- **Abstraction**: Clear interfaces using FastAPI & Airflow

Medallion Architecture: From Raw to ML-Ready Data

BRONZE

Raw Ingestion

- Ingest source data (CSV/JSON)
- No transformation or cleaning
- Partition by snapshot date
- Save as Parquet to Bronze datamart

→

SILVER

Clean & Standardize

- Enforce schema & cast data types
- Trim, lowercase, remove symbols
- Handle nulls
- Save as Parquet to Silver datamart

→

GOLD

ML-Ready Enrichment & Feature Engineering

- Join: transactions + users + cards + MCC
- Compute feature store per snapshot date
- Feature engineering
  - acct_opened_months
  - yrs_since_pin_changed
- Create label store
- Save to Gold datamart

Machine Learning Pipeline

1. Model Selection & Assessment

Logistic Regression

- Simple and interpretable
- Efficient with engineered features
- **Fastest inference**

XGBoost

- Robust to class imbalance
- Sequential learning improves rare fraud detection
- **Fastest inference**

MLP (Neural Network)

- Capture complex, non-linear relationships
- Highest model capacity
- Higher risk of overfitting
- **Slower inference**

Decision Criteria : Real-time inference requirement

Model Development focused on Logistic Regression & XGBoost

Machine Learning Pipeline

2. Training Strategy

Preprocessing

Imputation, one-hot encoding, scaling

Handling Imbalance

SMOTE used to generate synthetic minority (fraud) samples

Data Split

80% training, 20% test, **7-day sequential OOT period.**

Hyperparameter Tuning

Used **Optuna** for efficient exploration, faster convergence, adaptively focuses on promising regions

3. Optimization & Operations

Model Storage

Training parameters stored to **MLflow**

Monitoring

Final **training set persisted as reference** for future monitoring

Machine Learning Pipeline

4. Model Validation - Data Split Strategy

- 12 months of training data to **learn long-term patterns**
- 3 OOT test periods to **simulate real-world performance**
- Forward-looking evaluation **detects temporal drift** & **supports weekly monitoring**

Machine Learning Pipeline

4. Model Validation – Model Comparison & Selection

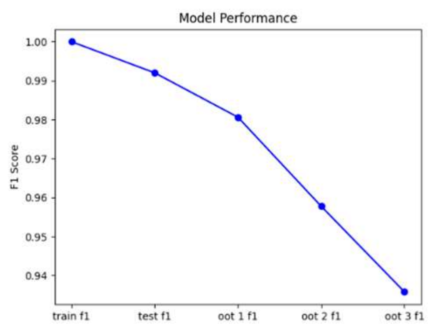
F1-Score: Primary Evaluation Metric

Chosen to **balance fraud detection sensitivity (recall)** and **false alarm minimization (precision)** under severe class imbalance.

(Optimizes both recall (catching frauds) and precision (avoiding false alarms), which is critical in fraud detection)

XGBoost

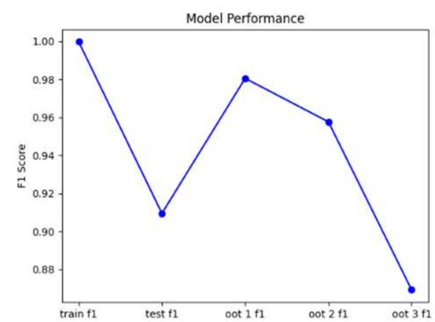
- F1-score gradually decays
- Smooth and stable degradation
- No overfitting, generalizes well
- Consistent under temporal shift



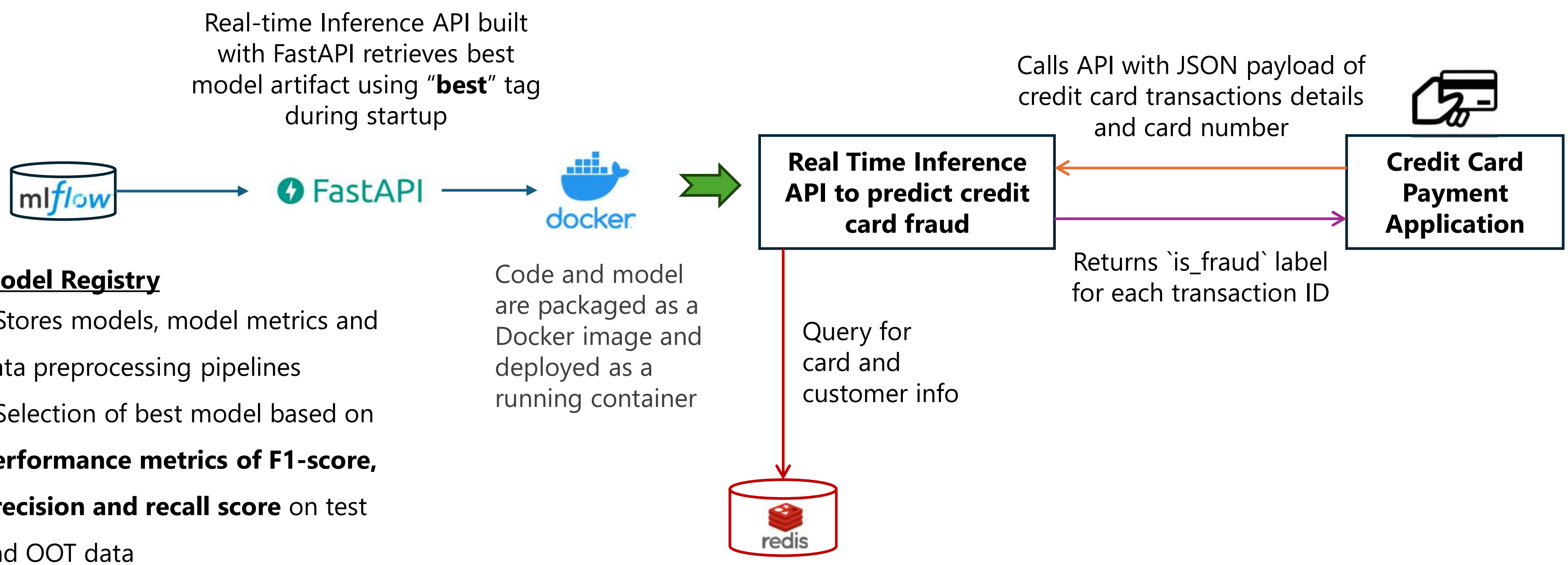
Selected

Logistic Regression

- F1-score shows high variance
- Indicates overfitting or model instability
- Not reliable for deployment



Model Inference



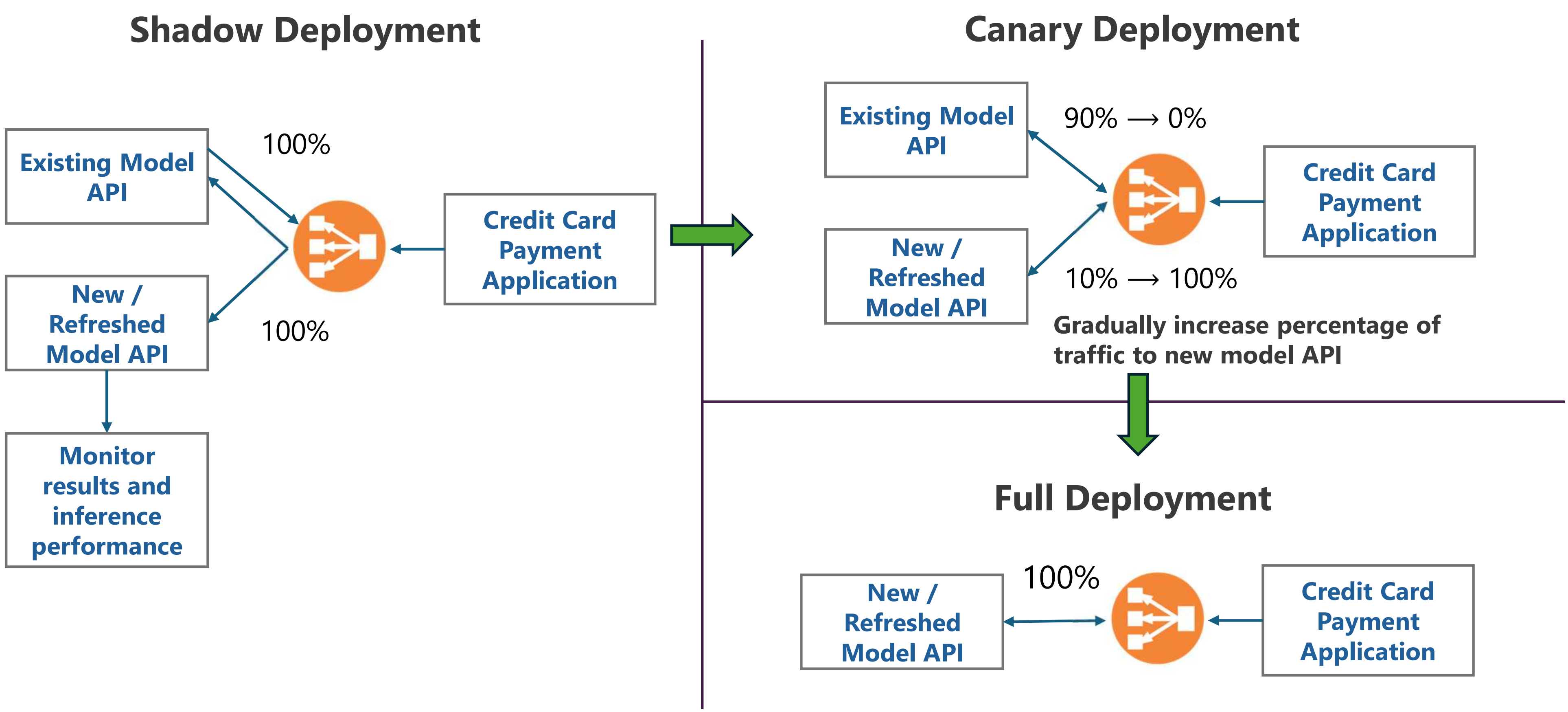
Model Registry

- Stores models, model metrics and data preprocessing pipelines
- Selection of best model based on **performance metrics of F1-score, precision and recall score** on test and OOT data
- Data scientist tags **"best"** model for deployment

Online Feature Store

- Latest card and customer information are stored in Redis key-value database for **fast data retrieval**, with card number as key
- Stores model inference results

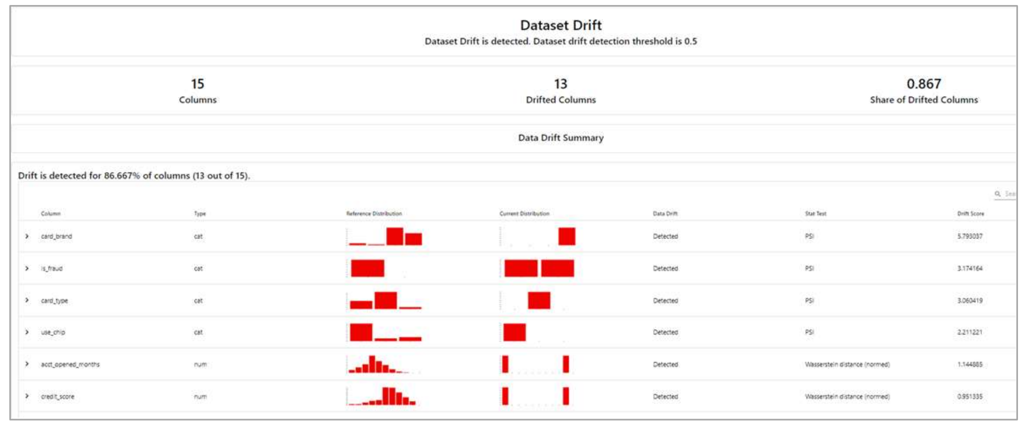
Model Deployment Strategy

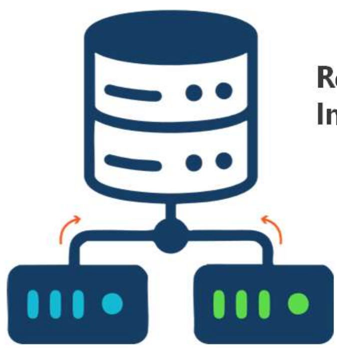


Model Monitoring

Detecting Data Drift Over Time

- Drift is calculated using **EvidentlyAI** and **PSI**
- Monitoring results are visualized in the dashboard





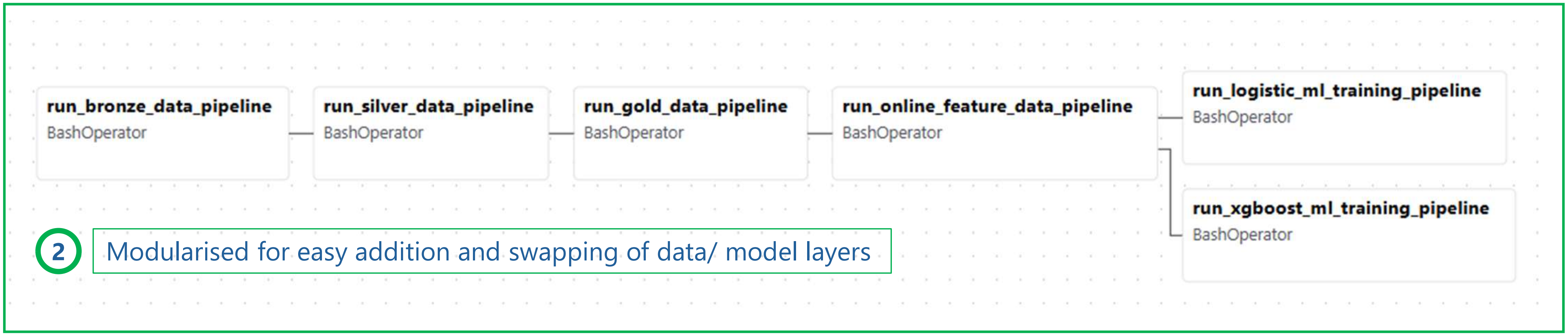
Reference & Inference dataset

Monitoring Pipeline

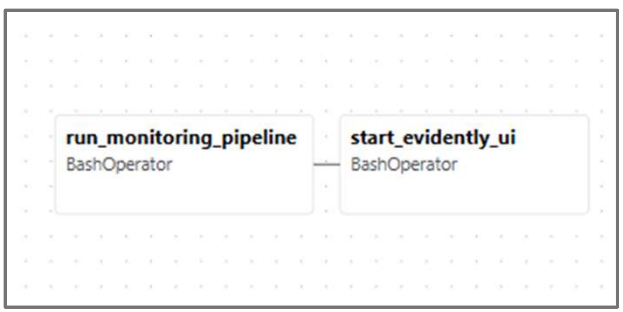
ML Pipeline **Inference**

Airflow DAGs

1 Every stage has email alert for visibility

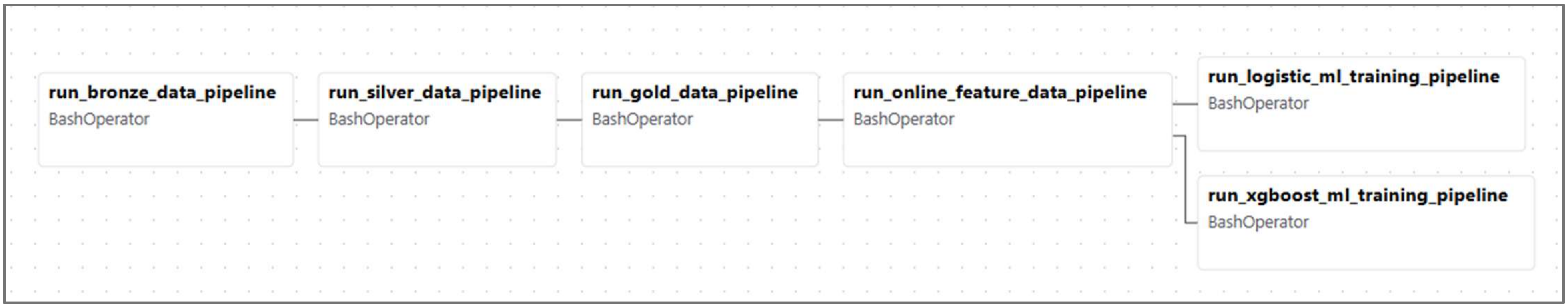


2 Modularised for easy addition and swapping of data/ model layers



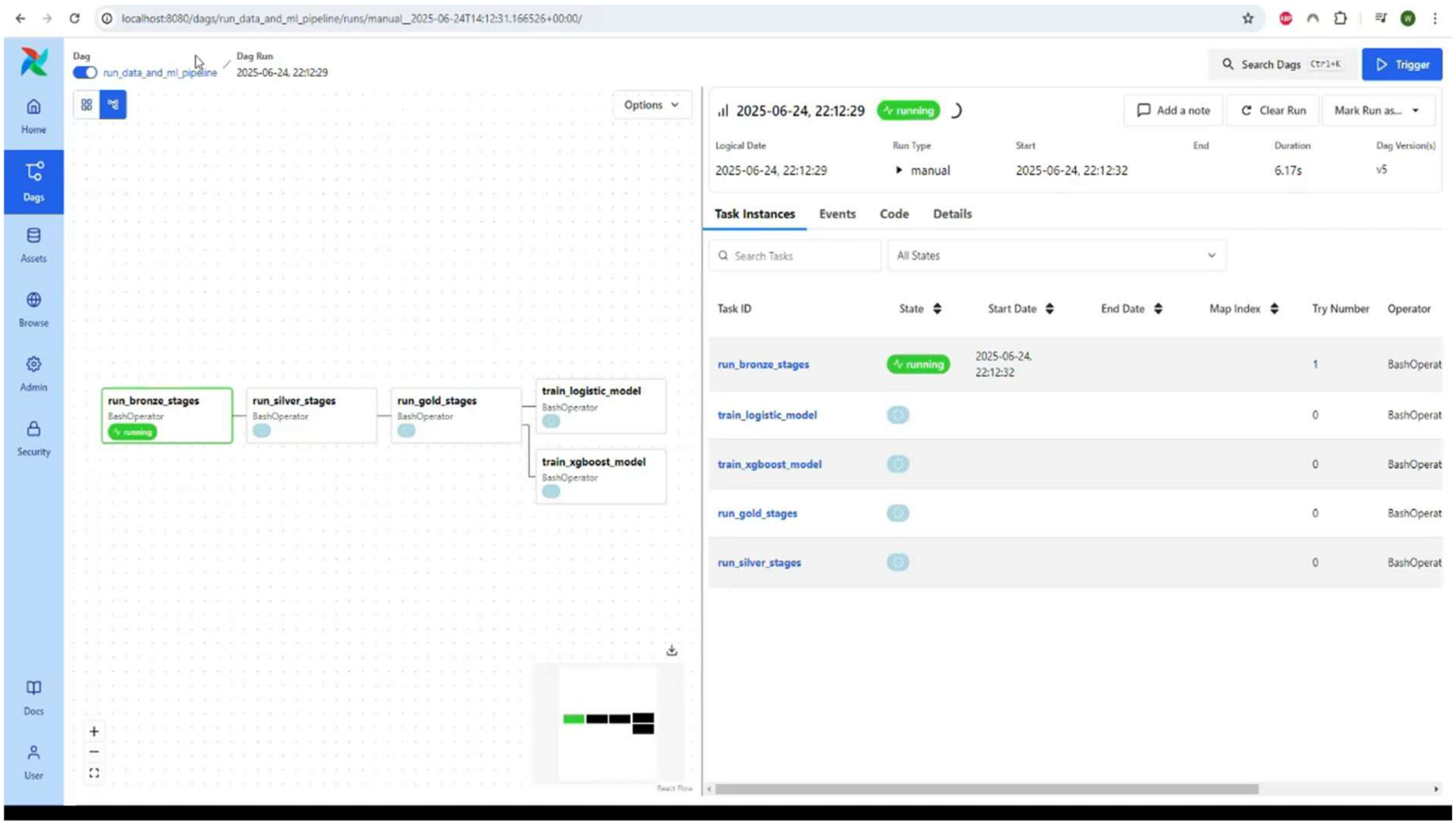
3 All DAGs are schedule ready with no backfill (clean start for each run)

Airflow DAGs



1. Manual trigger once data and model training pipeline is finished
2. Manual retrain for human-in-the-loop oversight
3. Data drift flagged when more than 50% of the columns show significant change

ETL & ML DAG


localhost:8080/dags/run_data_and_ml_pipeline/runs/manual__2025-06-24T14:12:31.166526+00:00/
Dag
run_data_and_ml_pipeline
Dag Run
2025-06-24, 22:12:29
Search Dags
Trigger
Home
Dags
Assets
Browse
Admin
Security
Docs
User
Options
run_bronze_stages
BashOperator
running
run_silver_stages
BashOperator
run_gold_stages
BashOperator
train_logistic_model
BashOperator
train_xgboost_model
BashOperator
2025-06-24, 22:12:29
running
Add a note
Clear Run
Mark Run as...
Logical Date
2025-06-24, 22:12:29
Run Type
manual
Start
2025-06-24, 22:12:32
End
Duration
6.17s
Dag Version(s)
v5
Task Instances
Events
Code
Details
Search Tasks
All States
Task ID
State
Start Date
End Date
Map Index
Try Number
Operator
run_bronze_stages
running
2025-06-24, 22:12:32
1
BashOperat
train_logistic_model
0
BashOperat
train_xgboost_model
0
BashOperat
run_gold_stages
0
BashOperat
run_silver_stages
0
BashOperat