CS611 – MLE

Assignment 2
ML Pipeline
Jun 2025

SUBMITTED BY:
ALVIN LEE ZE XIAN

Background & Objective

CONTEXT

As a financial institution providing cash loans, a key business challenge is to mitigate the risk of **loan default** by accurately assessing the creditworthiness of applicants **at the point of loan application**.

OBJECTIVE

To develop a machine learning model that predicts whether a customer is likely to default on a loan **before the loan is approved**. Enabling more informed, data-driven credit decisions and helps reduce financial risk.

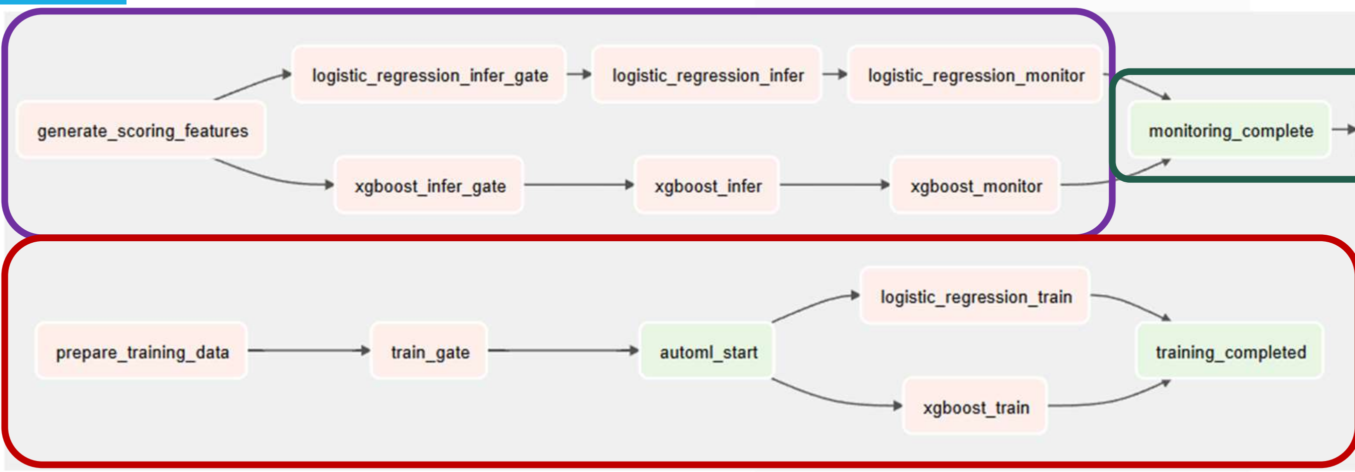
<u>Recap on Assignment 1</u>:

- Data processing pipeline built
- Raw data processed
- Gold standard data prepared

<u>What was achieved in Assignment 2</u>:

- Machine learning pipeline developed
- Model training, inference, selection
- Performance monitoring and dashboarding

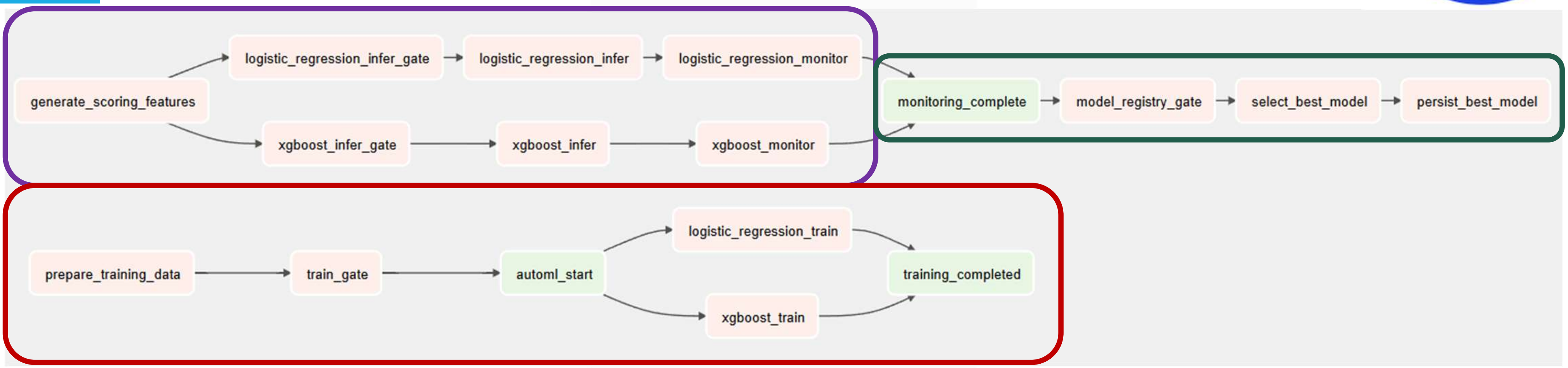
End-to-End ML Pipeline



This DAG orchestrated a full machine learning pipeline for loan default prediction using **Apache Airflow**, handling **both training and inference paths** for two models in parallel — **Logistic Regression** and **XGBoost**.

- Consist of 3 key segments – 1) Training, 2) Inference & Monitoring, and 3) Model Selection
- Parallelization
  - Both **Logistic Regression** and **XGBoost** were handled independently from training to monitoring.
  - DAG ensured **modularity and fault isolation** - if one model fails, the other can still complete.
- Backfilling and Scheduling
  - Airflow supported **monthly execution and backfilling** across historical time windows.
  - Pipeline is **stateless** across runs - governed by snapshot month.

End-to-End ML Pipeline



Training Segment

Executes only on the scheduled training snapshot

- **prepare_training_data**: Loads and combines monthly data
- **train_gate**: Ensures training runs only once
- **automl_start**: Marks training kickoff for both models
- **logistic_regression_train/ xgboost_train**: Trains both models and logs them to Mlflow
- **training_completed**: Marks training success

Inference & Monitoring Segment

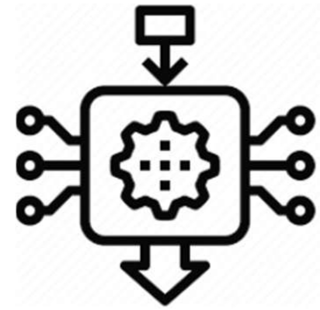
Runs for OOT months

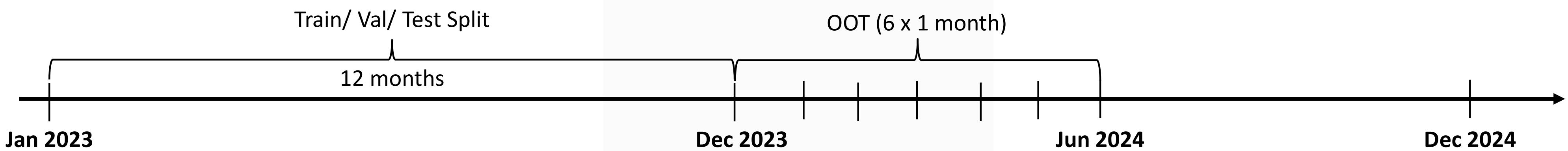
- **generate_scoring_features**: Loads current month's snapshot for inference
- ***_infer_gate**: Ensures inference only runs post-training
- ***_infer**: Scores data using persisted models
- ***_monitor**: Evaluates prediction quality and logs performance/drift

Model Selection Segment

- **monitoring_complete**: Waits for both model monitors to finish
- **model_registry_gate**: Logic to determine if new best model should be selected
- **select_best_model**: Compares OOT performance (F1 score, etc.)
- **persist_best_model**: Saves the selected model to the model registry

Model Training & Inference Strategy





Training Strategy

- **Training Data Period:** Jan 2023 – Dec 2023 (12 months)
- **Labels:** Labels were **revealed 6 months later** (label lag)
- **Models Trained:**
    - Logistic Regression – Interpretable, fast to train and baseline for comparison, useful for understanding linear relationships and coefficient weights
    - XGBoost – Handles nonlinear interactions and feature interactions well, strong performance on structured/tabular data, robust to class imbalance, missing values, and outliers
- **Parallel Training:** Both models trained simultaneously
- **Evaluation Metrics:** F1 score, AUC, accuracy (via MLflow logging)
- **Logged Artifacts:** Feature schema, model weights, config, plots
- **Output:** Models stored as versioned artifacts in model_store

Training triggered only once using a train_gate to avoid duplicate runs

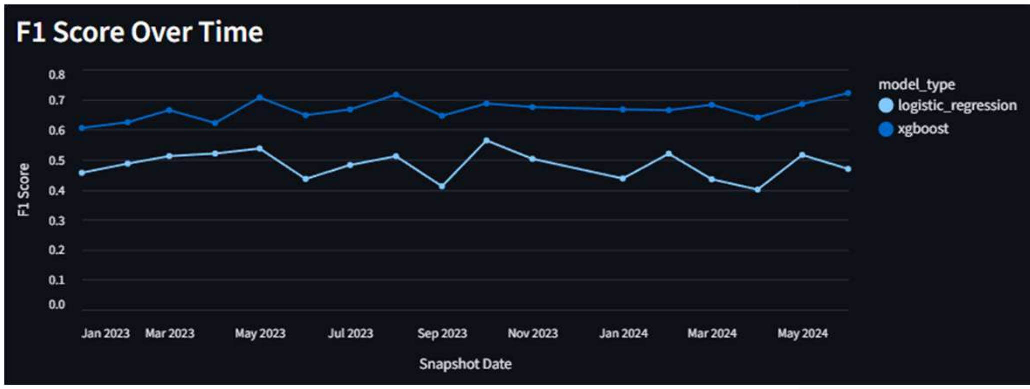
Inference Strategy

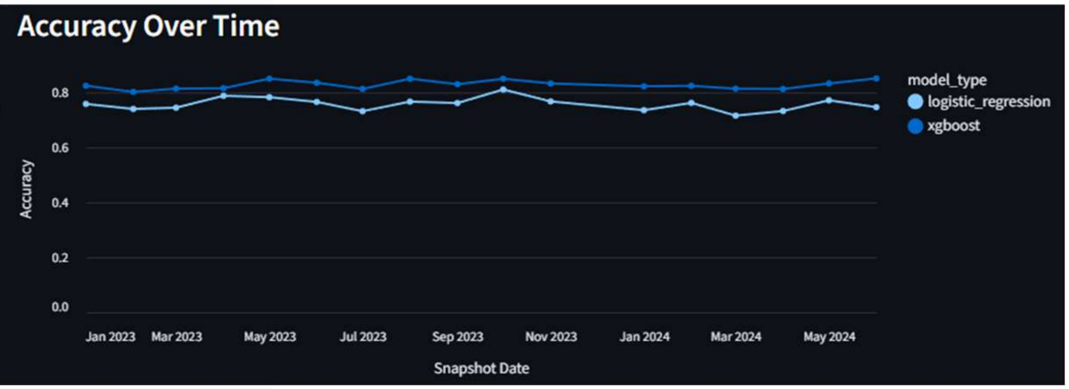
- **Inference Period:** Jan 2024 – Jun 2024 (OOT window)
- **Labels:** Labels were **revealed 6 months later** (label lag)
- **Feature Snapshot:** Monthly gold datasets (scoring features)
- **Pipeline Steps:**
    - Load trained model from registry
    - Align input schema
    - Score batch data for each customer
    - Output predictions with probability scores
- **Storage:** Predictions saved as prediction_{snapshot_date}.parquet

Models are applied independently per month, allowing backfilling and tracking.

Model Monitoring

Tracking model behavior both **pre-deployment** (backfilled) and **post-deployment** (live inference)





Timeline

- **Jan–Dec 2023:** Retrospective (backfilled) evaluation using historical data
- **Jan–Jun 2024:** Live OOT inference with delayed labels

Performance Metrics

- **F1 Score:** Measures prediction quality under class imbalance
- **Accuracy:** Measures overall correctness

Observations

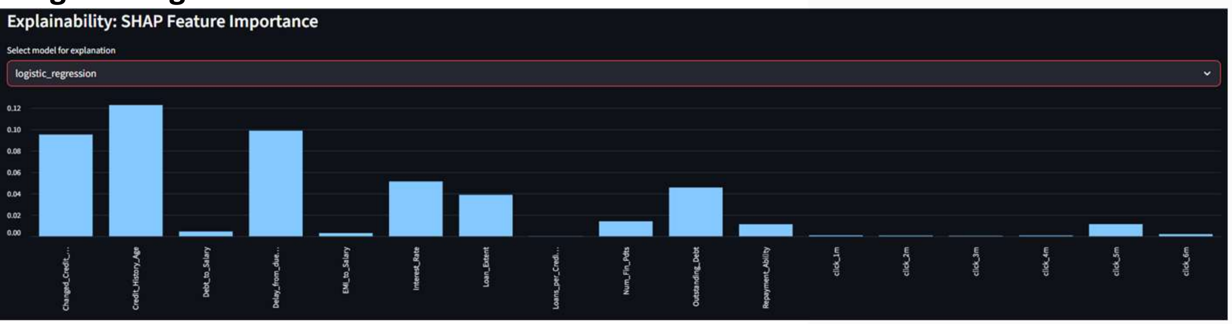
- **XGBoost** consistently outperformed Logistic Regression on **F1 score**
- **Accuracy** remained stable for both models across time

Backfilled insights enabled us to benchmark baseline performance prior to deployment, facilitated the assessment of model robustness on unseen historical data, providing confidence model selection

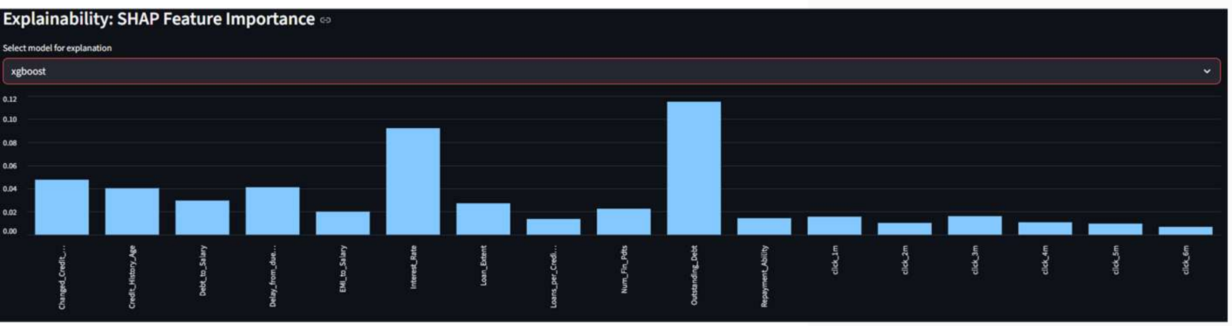
Model Monitoring

Showcasing **model interpretability** using SHAP values

Logistic Regression



XGBoost



SHAP values **quantified the average contribution** of each feature to the model's prediction.

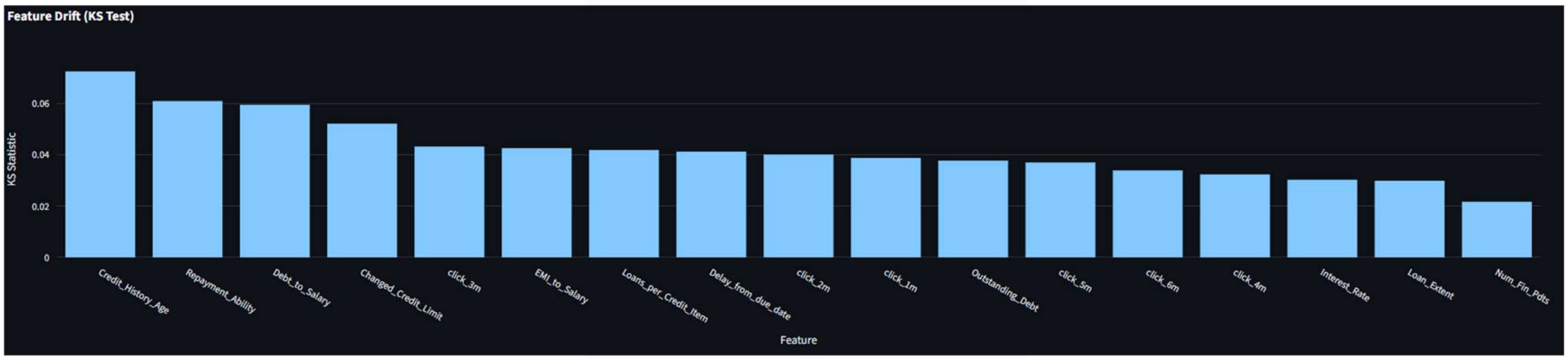
- Logistic Regression (**Interpretability**):
  - Top contributors: Credit_History_Age, Delay_from_due_date, Changed_Credit_Limit
  - Indicates reliance on well-understood, linear financial risk drivers

- XGBoost (**Performance**):
  - Broader spread of feature contributions
  - Heavier weight on: Interest_Rate, Outstanding_Debt, Repayment_Ability
  - Captures complex, nonlinear interactions

SHAP supports explainability required in regulated environments,
explainability boosts trust with credit risk analysts & compliance teams

Model Monitoring

Showcasing **data stability** using KS-statistics



- KS-statistics measured how much a **feature's distribution shifted** compared to the training set (2023-12-01 baseline)
- Drift is monitored monthly; KS > 0.2 triggers alerts
- Current snapshot: All features show KS < 0.1 → **No severe drift**
- Notable minor drift: Credit_History_Age, Repayment_Ability, and Debt_to_Salary with KS ≈ 0.06 - 0.07

KS-statistics validates input stability; model still operating within learned distribution, drift monitoring ensures predictions remain reliable as data evolves

Model Governance

Ensuring **robust management** of models in production with **clear SOPs** for refreshing, evaluating, and deploying models responsibly.

Model Governance Framework

- **Versioning & Registry**
  Models tracked and versioned using Mlflow, enables reproducibility, rollback, and auditability
- **Model Selection Logic**
  After each inference cycle, models are evaluated based on F1 and Accuracy, best-performing model (on latest OOT) is auto-selected
- **Artifact Management**
  Artifacts (model weights, schema, config) stored, standardized storage with metadata

Model Refresh SOP

- **Trigger Conditions for Retraining**
  - **Performance Degradation**: F1 score drop > 5% from trailing average
  - **Feature Drift Detected**: KS-stat > 0.2 on multiple key features
  - **Business Changes**: Introduction of new credit products, regulation updates
- **Retraining Steps**
  - Collect most recent 12-months labeled data
  - Re-run training pipeline
  - Log & compare with existing models in registry
  - Deploy only if new model outperforms previous in OOT evaluation

Deployment Options

- **Batch Scoring (Current)**
  Monthly prediction using Airflow DAG, suitable for periodic loan approvals
- **Realtime API (Future-Ready)**
  Wrap model inference in FastAPI or Flask for integration into credit app workflow, enables instant loan decisioning

Conclusion

End-to-End ML Pipeline Achieved

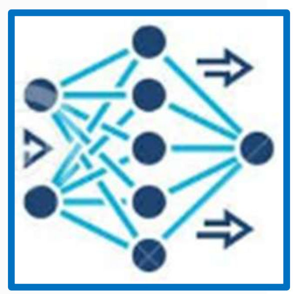
- Built a full pipeline covering **training**, **inference**, **monitoring**, and **explainability**.
- Modular structure orchestrated using **Airflow**, integrated with **MLflow** for tracking and reproducibility.

Model Performance & Monitoring

- **XGBoost** outperformed Logistic Regression on F1 score across backfilled and OOT periods.
- Monitoring dashboards provide visibility on:
  - **Performance trends** (F1, accuracy)
  - **Feature impact explainability** (via SHAP)
  - **Data drift detection** (KS-statistics)

Business Trust through Transparency

- **SHAP visualizations** promote interpretability and trust for credit risk teams.
- **KS-statistics** confirm model stability over time, ensuring reliable deployment.



Scalable Design

- Monthly batch inference supports backfilling and future scoring use cases.
- **Models versioned** and persisted using a registry for **reproducibility and governance**.

Thank You