



End-to-End ML Pipeline for Loan Default Prediction

A comprehensive system orchestrating data processing, model training, inference, and automated governance using Airflow, Docker, and Python to predict loan defaults at application time with continuous monitoring and drift detection.

Overview: End-to-End ML Pipeline (3 DAGs)

Goal

Predict loan default at application time, monitor performance & drift, and enforce automated governance.

Tech Stack

Airflow (3 DAGs), Docker, Python (pandas, scikit-learn, XGBoost), Parquet (pyarrow).

Data Layers

Bronze → Silver → Gold; Model artifacts in model_store; results in results/.

Temporal Integrity

- Features at application time (MOB=0), labels at +6 months (MOB=6).
- Windows: Train 2023-01→12, Val 2024-01→03, Test 2024-04→05, OOT 2024-06.

Selection

Best model chosen by AUC, used for inference across a date range.

Monitoring

Performance metrics + PSI, saved monthly; visualizations generated.

DAG 1: Data Processing (Bronze → Silver → Gold)

Schedule

Monthly backfillable; orchestrates per-snapshot processing.

Inputs (Bronze)

clickstream, attributes, financials, and lms_loan_daily snapshots.

Outputs

Silver

cleaned, standardized, de-duplicated tables
(clickstream/attributes/financials/loan_daily).

Gold

- feature_store_YYYY_MM_DD.parquet (MOB=0)
- label_store_YYYY_MM_DD.parquet (MOB=6; DPD threshold configured)



Silver Data Cleaning (Highlights)



Type Standardization

- Cast numeric columns
- coerce invalid values
- handle $\pm\text{inf}$
- consistent date types



Missing Values

- Impute numeric columns (median), safe coercion.
- Drop/flag records missing core identifiers (loan_id, Customer_ID).



De-duplication

stable keys (loan_id, Customer_ID, snapshot_date).



Sanity Checks

non-negative amounts/tenure; valid income ranges (truncate or clip if needed).



Join Readiness

conforming column names and keys for downstream merges to Gold.


Feature Engineering (Gold Feature Store @ MOB=0)

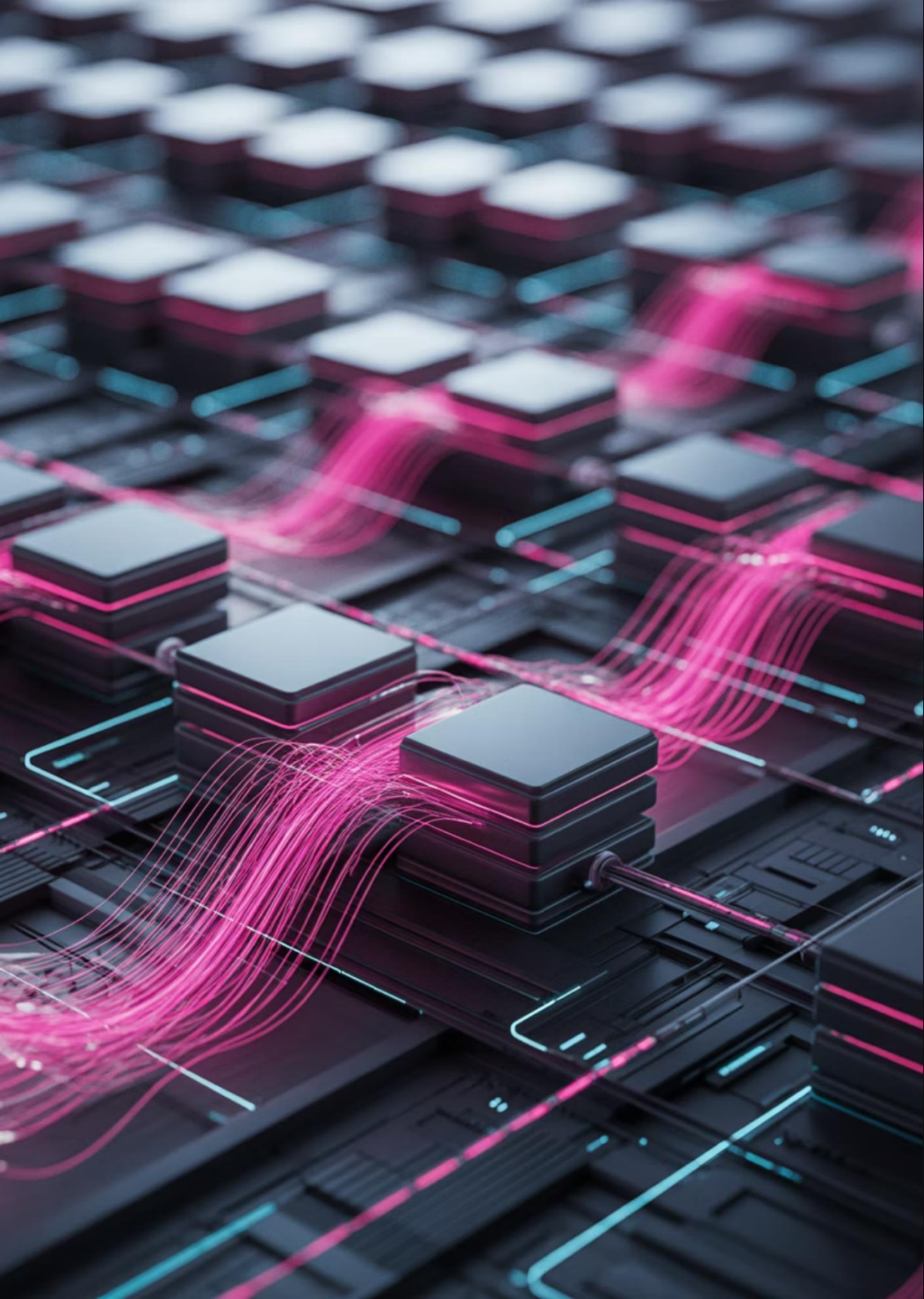
Feature groups included (config-driven, leakage-safe at application time):

<div><div>Loan Features</div><div>tenure, loan_amt</div></div>	<div><div>Demographics</div><div>customer_age</div></div>
<div><div>Financials</div><div>Annual_Income, Monthly_Inhand_Salary, Num_Bank_Accounts, Num_Credit_Card, Interest_Rate, Num_of_Loan, Num_of_Delayed_Payment, Outstanding_Debt, Credit_Utilization_Ratio, Total_EMI_per_month, Amount_invested_monthly, Monthly_Balance, debt_to_income_ratio</div></div>	<div><div>Clickstream</div><div>fe_1 ... fe_20</div></div>

Label Engineering

label_store at MOB=6 using DPD_THRESHOLD=30; aligned to features via snapshot_date + 6 months.

 **Leakage guardrails:** strict MOB filtering (features: t, labels: t+6) to mirror real-time availability.



DAG 2: Model Training (Manual, Readiness-Gated)

Readiness Checks

full feature months + aligned label months available for Train/Val/Test.

Models

LogisticRegression, RandomForest, XGBoost (balanced handling; XGB tuned via scale awareness).

Splits (absolute)

- Train: 2023-01→12, Val: 2024-01→03, Test: 2024-04→05.



Selection

Best model by AUC



Saves

model.pkl, scaler.pkl, metadata.json



Writes

model_config.json and model_evaluation.json

Stability: forbids empty splits; robust metrics (nan-safe log_loss/AUC in degenerate cases).

Model Comparison & Selection

RandomForest was selected for deployment due to its highest Test AUC and best F1 balance, demonstrating manageable overfitting.

Performance Metrics (Test Set)

Model	AUC	Accuracy	Precision	Recall	F1	Log Loss
Logistic Regression	0.7251	0.6873	0.4707	0.6357	0.5409	0.6162
Random	0.7716	0.7480	0.5572	0.6357	0.5939	0.5278
XGBoost	0.7669	0.7361	0.5387	0.6220	0.5774	0.5388

Generalization (Train → Test AUC Drop)

- **LogisticRegression:** 0.753 → 0.725 (Δ -0.029) – Minimal overfit.
- **RandomForest:** 0.949 → 0.772 (Δ -0.178) – Moderate, acceptable overfit.
- **XGBoost:** 0.989 → 0.767 (Δ -0.222) – Largest gap, strong overfit signal.

Key Observations

- RandomForest achieves the top Test AUC and highest F1.
- XGBoost nearly matches RandomForest AUC but shows heavier overfit.
- LogisticRegression is stable but lags in precision and AUC.

Selection Rationale

- **Primary Metric (AUC):** RandomForest edges out others on unseen data.
- **Secondary Metric (F1):** Highest harmonic mean indicates balanced trade-off between missed defaults and false positives.
- **Overfit:** RandomForest's generalization gap is acceptable; XGBoost requires additional tuning for regularization.

❏ Decision Statement

Deploy **RandomForest** as current champion. Retain **XGBoost** as a candidate for future tuning and keep **LogisticRegression** as an interpretability baseline.

DAG 3: Inference & Monitoring (Range-Aware)

Inference

- Check Data Availability.
- Loads best model; features in trained order; numeric coercion; safe imputation.
- Output: predictions_<MODEL>_<YYYY_MM_DD>.parquet with prediction_proba, prediction_label, threshold, model_name, inference timestamps.

Monitoring

- Merges predictions with labels at snapshot_date + 6 months on loan_id + Customer_ID.
- Metrics per month: AUC, Accuracy, Precision, Recall, F1, Log Loss, Confusion Matrix.
- Drift: PSI vs baseline (first available predictions); saved monthly JSON + Parquet; cumulative model_monitoring.json.
- Robust: handles empty joins, missing labels (PSI-only mode), flexible file schemas, JSON-safe outputs.

Visualizations & Reporting (Gold → Results)

Inputs: cumulative monitoring history + monthly predictions in Gold.

Charts saved to `results/monitoring_visualizations`

- Performance Metrics Over Time

(AUC, Acc, Prec, Rec, F1, Log Loss) with thresholds.

- PSI Trend

with warning/critical bands (0.1/0.2).

- Confusion Components Trend

(TP/FP/TN/FN).

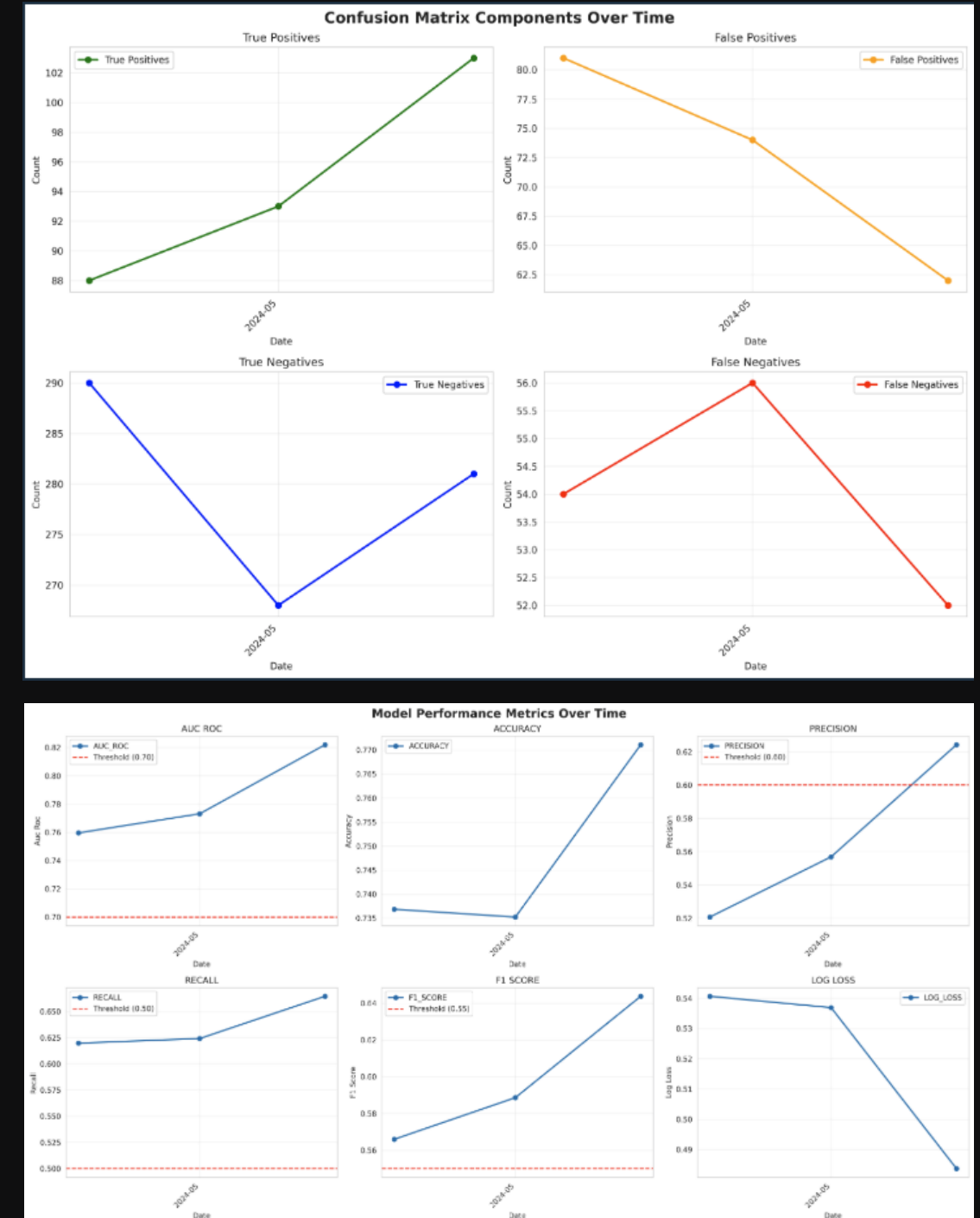
- Threshold Compliance Trend

(pass rate across AUC/Precision/Recall/F1).

- Prediction Distribution Over Time

(normalized to prediction_proba).

Summary report: `monitoring_summary_report.txt` (latest + averages, compliance).



Governance: Automated Monitoring & Retraining

A critical feedback loop ensuring model performance and data integrity are continuously upheld through automated checks and re-triggering of the training pipeline.

Governance Triggers (Configurable)

Performance Degradation

AUC < 0.70 Precision < 0.60 Recall < 0.50

Priority Levels

P0 : ROC-AUC (Critical Business Metrics)

P1 : Accuracy (Important but not critical)

P2 : F1, Precision, Recall

Data Drift

PSI (Population Stability Index) ≥ 0.20

Automated Actions

A `BranchPythonOperator` checks these rules monthly after monitoring:

- If all rules pass: The pipeline concludes with no action.
- If any rule fails: The full Training pipeline (DAG 2) is automatically re-triggered, followed by deployment of the best model and resumption of inference/monitoring.

Guardrails

- MOB integrity enforced (MOB=0 for features, MOB=6 for labels).
- Range-aware execution: Only months with aligned labels contribute to performance metrics.
- Cumulative audit trail in `model_monitoring.json` for traceability.

