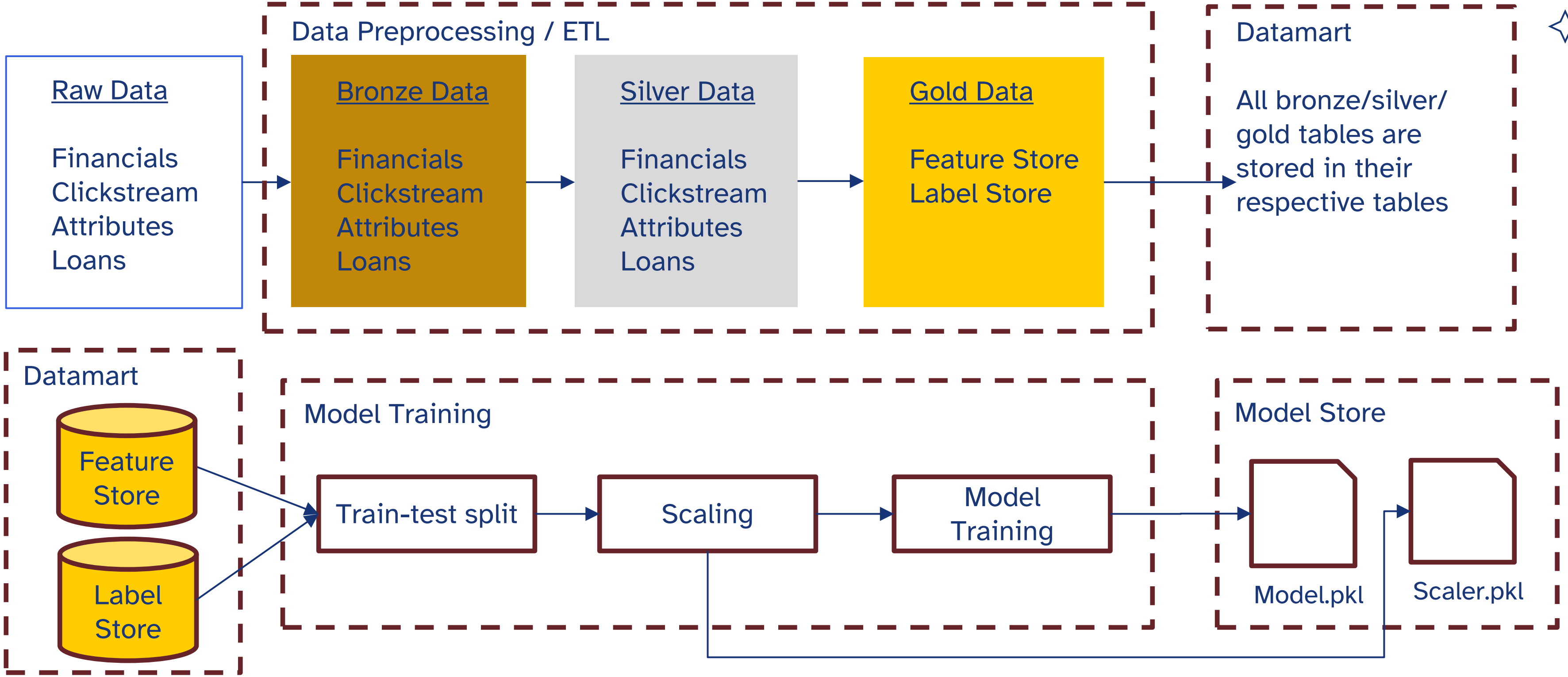
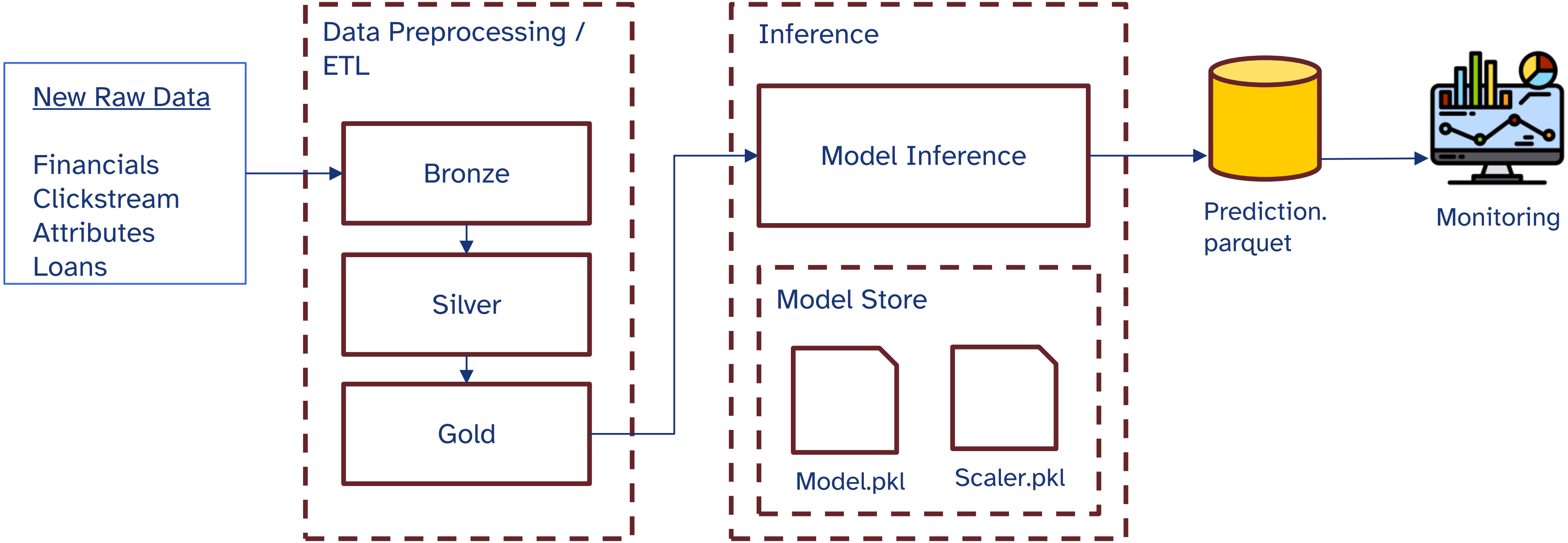
Machine Learning Pipeline for Loan Default Prediction

Lau Li Qing
Assignment 2
CS611 Machine Learning Engineering

Architecture Overview



Architecture Overview



Dataset

- Used the dataset that I had previously created as:

- **Domain Expertise & Context**
  - Deep understanding of business logic behind each feature
  - Familiar with data quirks, edge cases, and limitations
  - Can explain feature definitions to stakeholders confidently

- **Risk Mitigation**
  - Lower risk of data leakage (know what went into feature creation)
  - Easier debugging when issues arise
  - Can trace back to source systems if needed

- **Reproducibility & Maintenance**
  - Clear ownership and accountability for data pipeline
  - Easier to maintain and update as business requirements change
  - Consistent feature definitions across train/test/production

Data Pipeline: Adobe Acrobat Document

Model Training

Purpose: Predict if a person would default on their loan during application based on their financial history, attributes and loan history
Training: GridSearchCV was used to find the best attributes for the models
Total dataset: 18 months (July 2023 to Dec 2024), split into 80% train 20% test based on stratification of default label, last month (Dec 2024) reserved for out-of-time testing

Logistic Regression

Explainable on how each factor increases or reduces default risk

Best parameters selected:
- C: 0.01
- Solver: liblinear

Random Forest

Captures complex patterns better, **less sensitive to noise** and **outliers**

Best parameters selected:
- n_estimators: 200
- max_depth: 10
- min_samples_split: 2
- min_samples_leaf: 2

Selection Criteria: AUC score, it was selected due to its abilities to account for imbalanced dataset (~30% default rate) which existed for this loan default dataset
Best model selected: **Random Forest (Test AUC:** 0.6988 vs Logistic regression Test AUC: 0.6649 **)**

Airflow Schedulers

Data extraction [1st of each month, 12am] → Inference [1st of each month, 1am] → Monitoring [1st of each month, 2am]

1) Model training pipeline
    - Runs annually on 1st January
2) Data extraction pipeline
    - Runs at 12am on the 1st of each month
    - Extract new data using the data preprocessing and store them into bronze, silver, gold tables
3) Model inference pipeline
    - Runs at 1am on the 1st of each month
    - Loads latest gold features and trained model to generate predictions
    - Stores prediction in gold prediction table
4) Model monitoring pipeline
    - Runs at 2am on the 1st of each month
    - Calculates the performance metrics, input feature drifts using Population Stability Index and prediction stability via statistics
    - Generates automated dashboards and alerts

Monitoring

1. Performance Monitoring

- Retrospective with 2 months lag as we assumed that within 2 months, we would be able to get the ground truth
- Generates AUC, F1-score, precision and recall dashboard visualizations

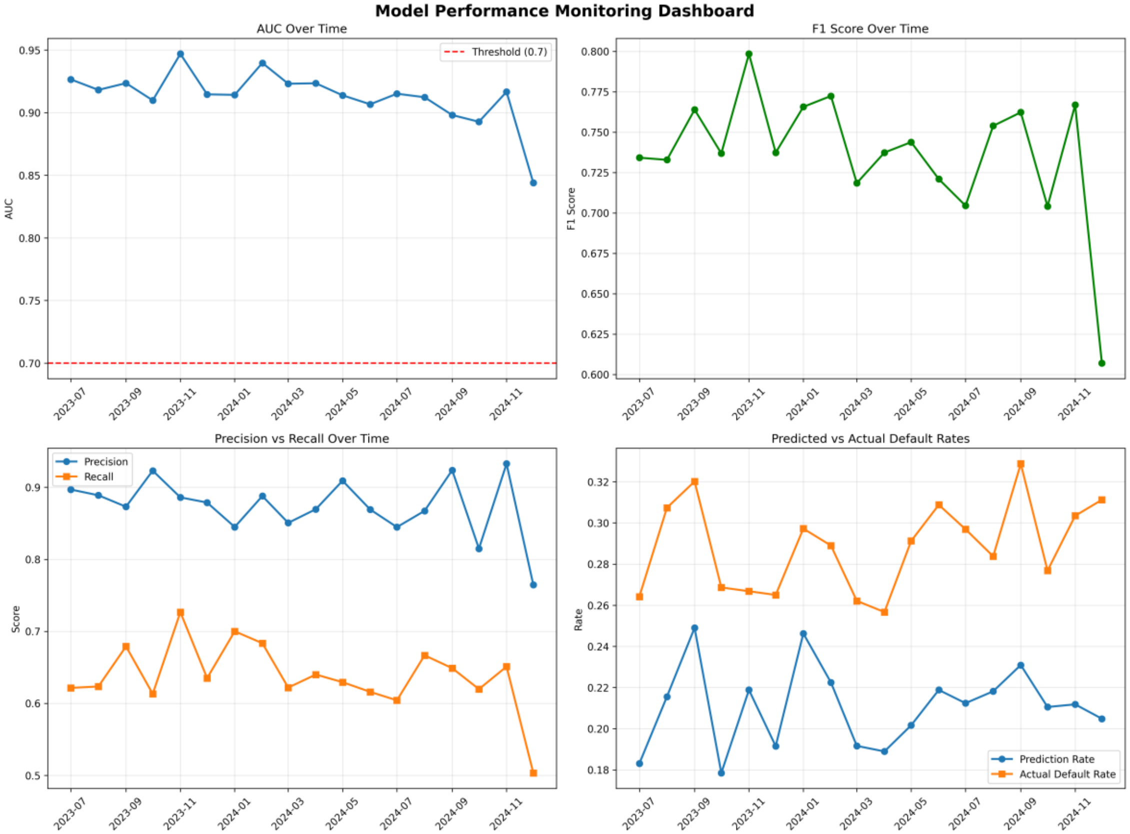
2. Input Feature Drift (PSI)

- Check if the input data distribution has been shifted
- Eg: Training data for monthly salary are around $5000/month but new data coming in are closer to $10,000/month
- To be investigated further when 30% of features show high drift
- **Shows alerts** when PSI > **0.25**

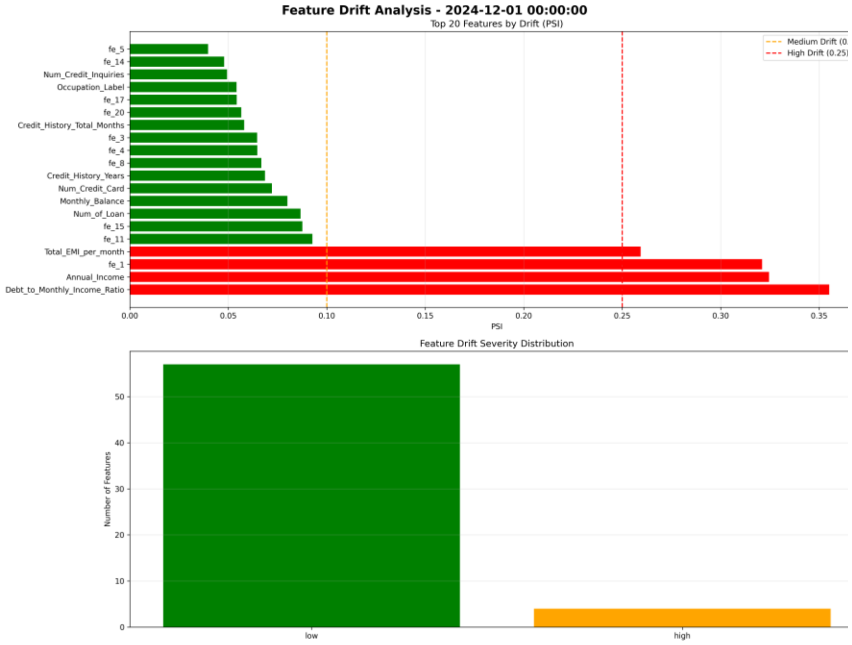
3. Prediction Stability

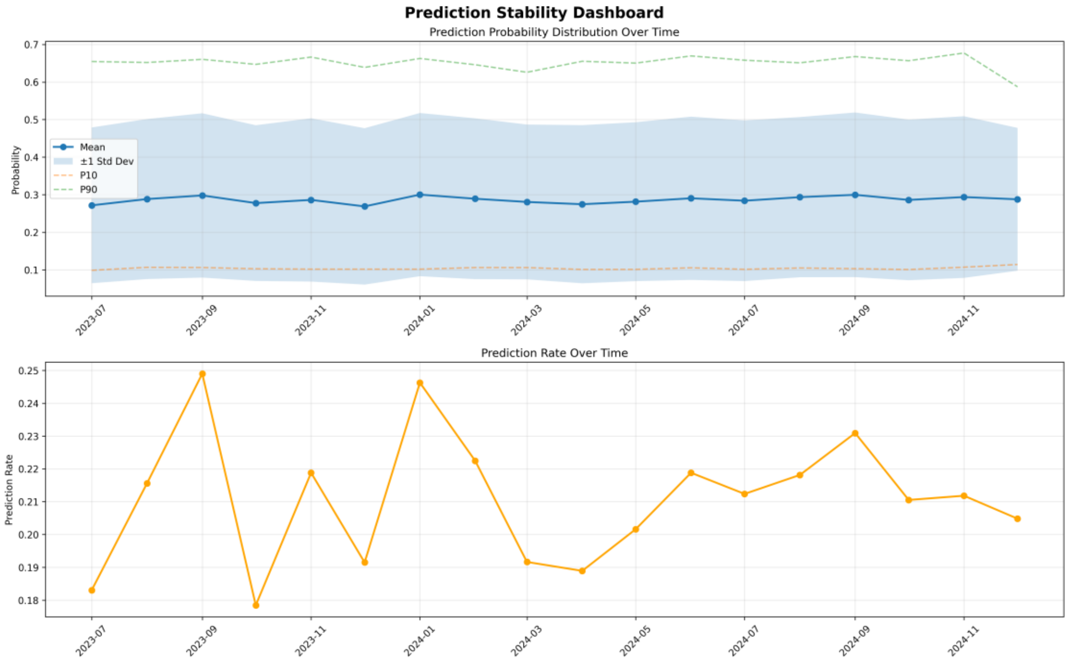
- Tracks:
    - Mean probability – should be stable month to month
    - Prediction rate - % of defaults
    - Distribution metrics (10/50/90^{th} percentile)
- This detects if model behaviour changes even without labels available

Monitoring – Model Performance



Monitoring – Stability





Model SOP

- **Refresh Triggers:**
  - Automatic: AUC < 0.7 (2 months), >30% high drift, prediction rate change >10%
  - Scheduled: Annually every January
  - Manual: Regulatory changes, market shifts, policy updates

- **Refresh Process (5 weeks):**
  - Investigation & approval (Model Risk Committee)
  - Data validation & feature engineering
  - Model training & OOT testing
  - Shadow deployment (2 weeks parallel running)
  - When shadow deployment is shown to be working better than current model, we should switch over

- **Governance:**
  - Data Science team: Develops and validates
  - Model Risk Committee: Approves deployment
  - Rollback: 15-min revert if critical issues
  - Documentation: Model validation report, audit trail maintained