Scoring Loan Risk on Peer-to-Peer Lending Platform

CS611 Machine Learning Engineering

Group 2

Aditya Vijay, Lope Santos IV, Oh Boon Long, Vinnie Seah, Xu Pei Min, Zhong Yi

Agenda

- Problem Statement
- Business & ML Objectives
- Data & Features Overview
- Final Architecture
- Model Development Strategy
- Deployment, Inference, and Automation
- Monitoring and Maintenance
- Challenges and Mitigations

Problem Statement

Context:

We're a P2P lending startup. Lenders on our platform face high risk due to **limited visibility into borrower creditworthiness.**

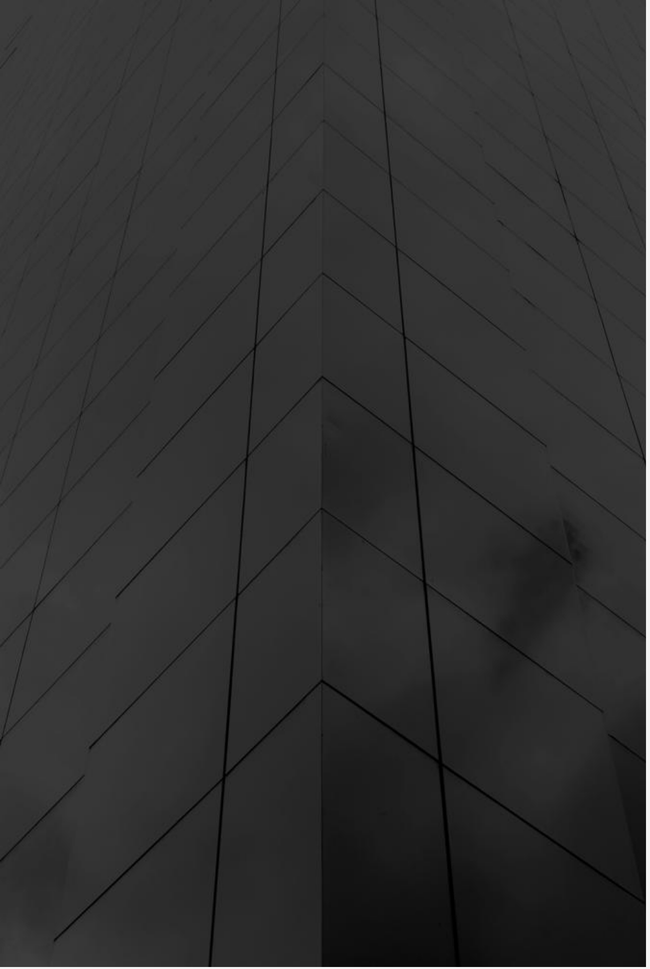
Problem:

Without reliable credit scoring, lenders may **fund high-risk borrowers,** leading to **defaults and financial losses.**

Solution

Need:

An automated, data-driven credit scoring system to assess risk before loans are issued.



Data & Features Overview

Data Sources	Target Variable
Demographics – From borrower application forms	'grade' – loan grade assigned to each borrower As shown below:
Loan Terms & Payments – From loan management system (LMS)	
Credit History – From credit bureau reports	
Financial Info – From bank statements / transaction data	

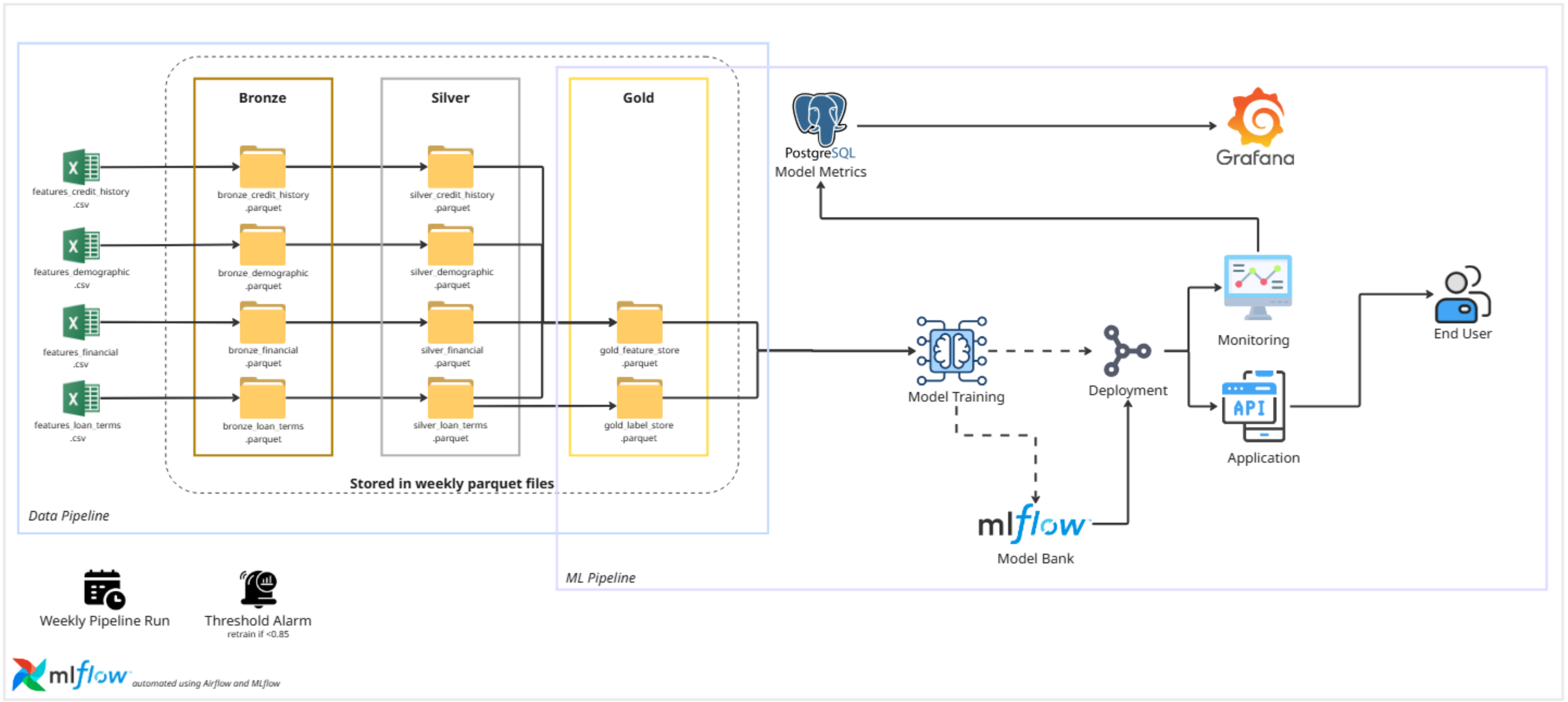
Low Risk

High Risk

Data Volume & Granularity:

- 2.3 million samples
- Daily snapshots from 2022 to 2024
- Each sample tied to member_id + loan_id

Architecture Diagram



Model Development Strategy

CatBoost

Boosting model optimized for categorical features.

LightGBM

Fast gradient boosting using leaf-wise tree growth

XGBoost

Gradient boosting with regularization for high accuracy

Decision Tree

Single interpretable model with rule-based splits.

Deployment & Inference

Registry and Deployment

Model training metadata gets stored in MLFlow, where the better performing model gets deployed to production, which is used to infer on current week's data.

Real time vs batch

Weekly pipeline runs ensure new data from this week will have their credit score by the following week.

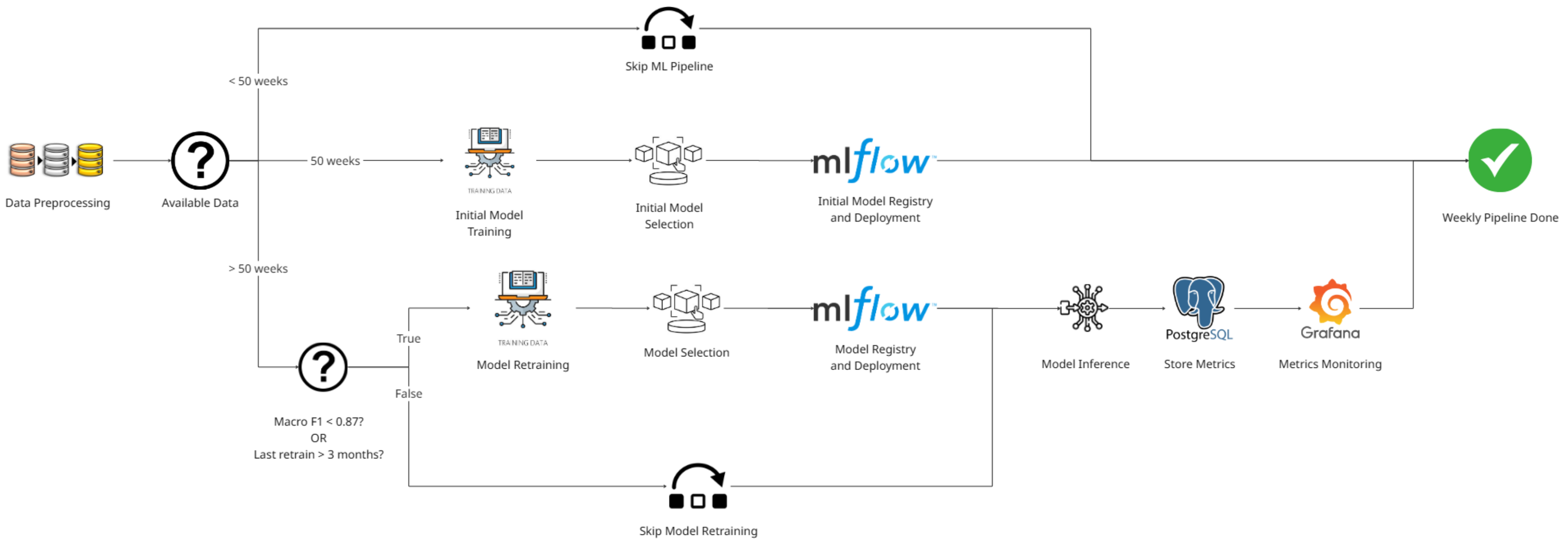
Model Monitoring

Macro F1 score over time is monitored using Grafana pulling data from the PostgreSQL database.

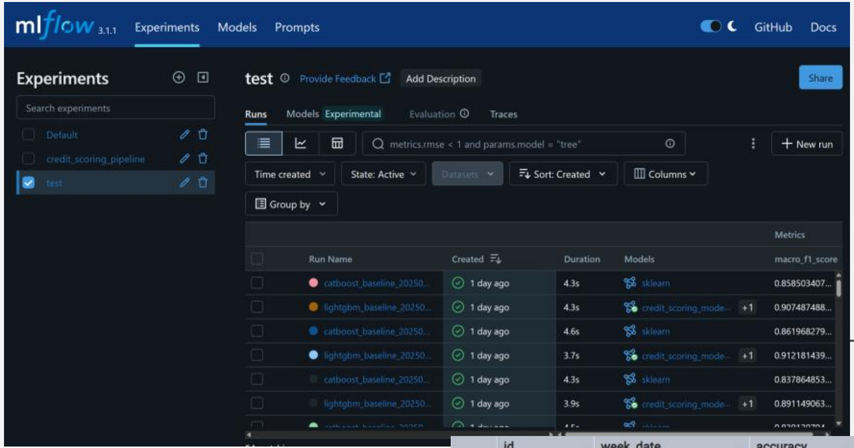
Model Inference

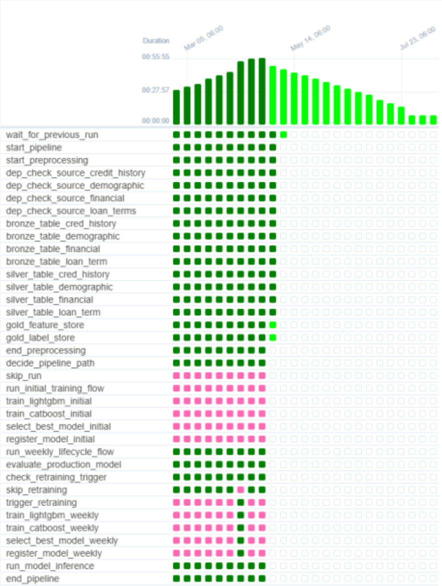
Metrics for each week is stored in a PostgreSQL database.

Automation & Orchestration



Project Artefacts





Model Pipeline

Airflow

Model Registry

MLFlow

Model Metrics

PostgreSQL

	id [PK] integer	week_date character varying (10)	accuracy numeric (5,4)	macro_f1 numeric (5,4)	weighted_f1 numeric (5,4)	total_samples integer	created_at timestamp without time zone
1	1161	2023_06_11	0.9506	0.9019	0.9504	14503	2025-06-24 08:23:10.373031
2	1160	2023_06_04	0.9504	0.8960	0.9503	14407	2025-06-24 08:20:28.590334
3	1159	2023_05_28	0.9486	0.8932	0.9484	14481	2025-06-24 08:18:01.435412
4	1158	2023_05_21	0.9476	0.8792	0.9473	14491	2025-06-24 08:15:34.842419
5	1157	2023_05_14	0.9488	0.8910	0.9486	14524	2025-06-24 08:13:10.982328
6	1156	2023_05_07	0.9474	0.8969	0.9472	14465	2025-06-24 08:10:34.741367
7	1155	2023_04_30	0.9481	0.8934	0.9479	14563	2025-06-24 08:08:11.244944
8	1154	2023_04_23	0.9502	0.8964	0.9500	14491	2025-06-24 08:05:47.865653
9	1153	2023_04_16	0.9489	0.8994	0.9488	14352	2025-06-24 08:03:13.65332
10	1152	2023_04_09	0.9517	0.8974	0.9516	14442	2025-06-24 08:00:36.571364
11	1151	2023_04_02	0.9492	0.8966	0.9490	14324	2025-06-24 07:57:50.38476
12	1150	2023_03_26	0.9501	0.8969	0.9499	14509	2025-06-24 07:48:57.490324

Model Monitoring

Challenges & Mitigations

Performance Improvements

- Training bottleneck when doing sequential training – explore solutions to computational and memory limits and try parallel processing. This will also allow us to do daily data processing and inference, creating a more real-time model.
- Add performance ceiling tracking to better evaluate model performance

Economic/Market Changes

- Model will need to be retrained after these changes take place to continue making accurate predictions

Demographic Shifts

- Users in different stages of life have different needs and financial ability
- We will track shifts in demographic changes so that we can retrain the model as needed

Thank you

Q&A