Loan Default Prediction: A Technical Case Study

ML-Driven Risk Mitigation in Financial Lending

by Nishant Sharma

Business Problem & Objective

Background

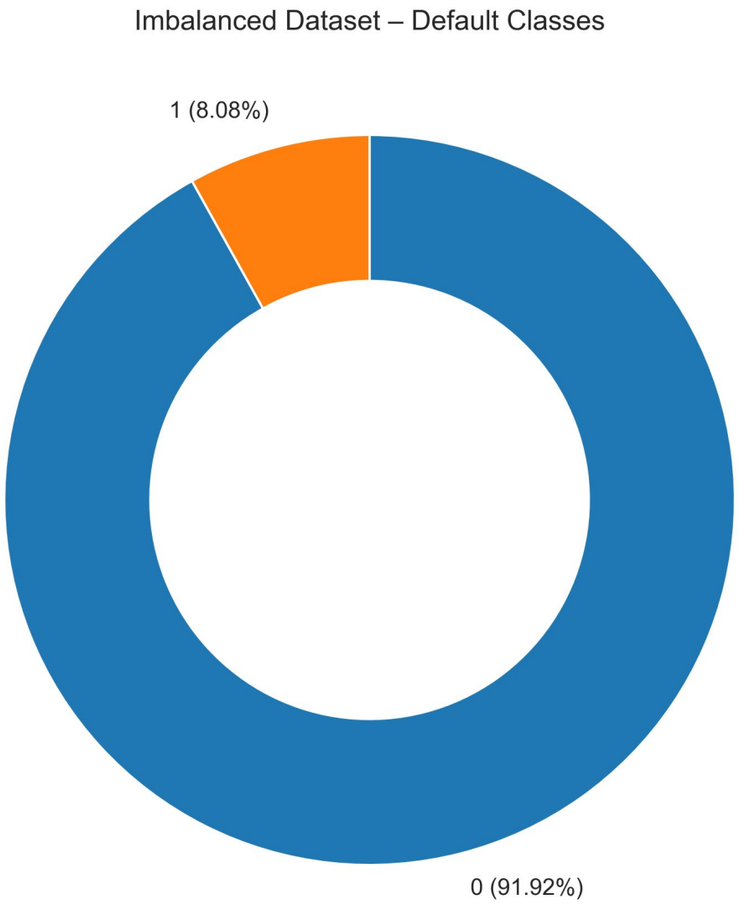
Loan defaults can significantly damage a financial institution’s profitability and credibility. Traditional risk assessment models — relying on credit scores, income, and collateral — fail to capture complex borrower behaviors.

Why Machine Learning?

Because ML can model hidden, nonlinear relationships and provide early signals for borrower risk that traditional methods miss.

Our Objective

To design a predictive model that flags high-risk borrowers early — reducing missed defaults and financial losses.



Data Landscape & Analytical Challenge

Our dataset comprises 121,856 loan records with 40 features, including income and repayment behaviour. The target variable indicates loan default (1/0).

Imbalanced Data

The dataset is highly imbalanced, with 90% non-defaults and only 10% defaults. Standard accuracy metrics would favour the majority class.

False Negatives Cost

The cost of a false negative (predicting non-default for a defaulting loan) is too high. This error directly leads to financial losses.

Focus on Recall

We emphasise 'Recall' and 'F1-score' for the minority default class. This ensures high detection of actual defaults.

Model Development & Evaluation Strategy

Our approach involves employing models robust to class imbalance. We combine algorithmic solutions with sampling-based techniques to address this challenge.

Algorithms

- Balanced Random Forest
- XGBoost & LightGBM
- RandomForest + RandomUnderSampler

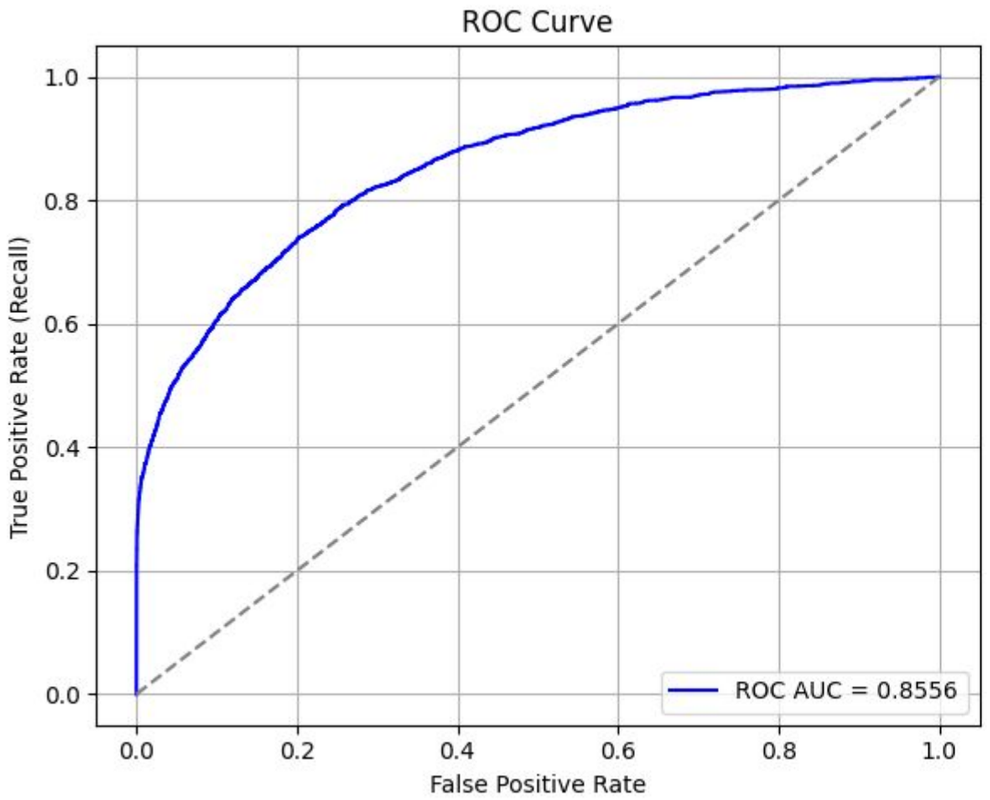
Evaluation

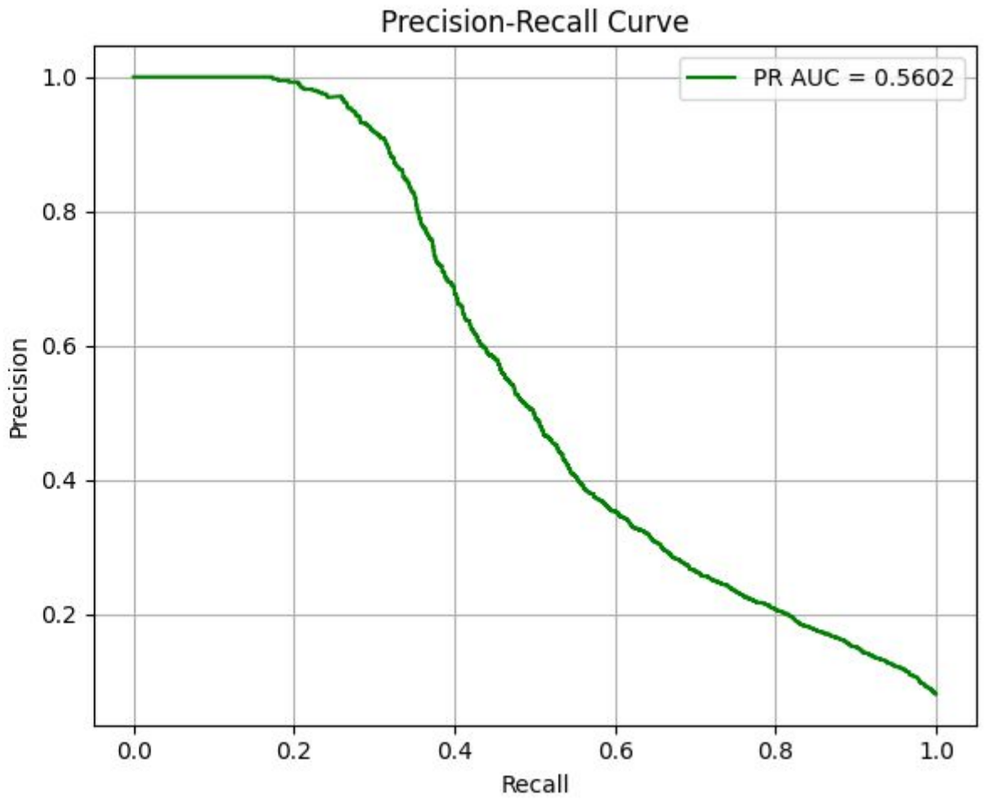
- Stratified train/test split
- MLflow for experiment tracking
- DVC/Git for dataset/code versioning
- Optimise for high recall
- Maintain acceptable precision

Final Model Evaluation Summary

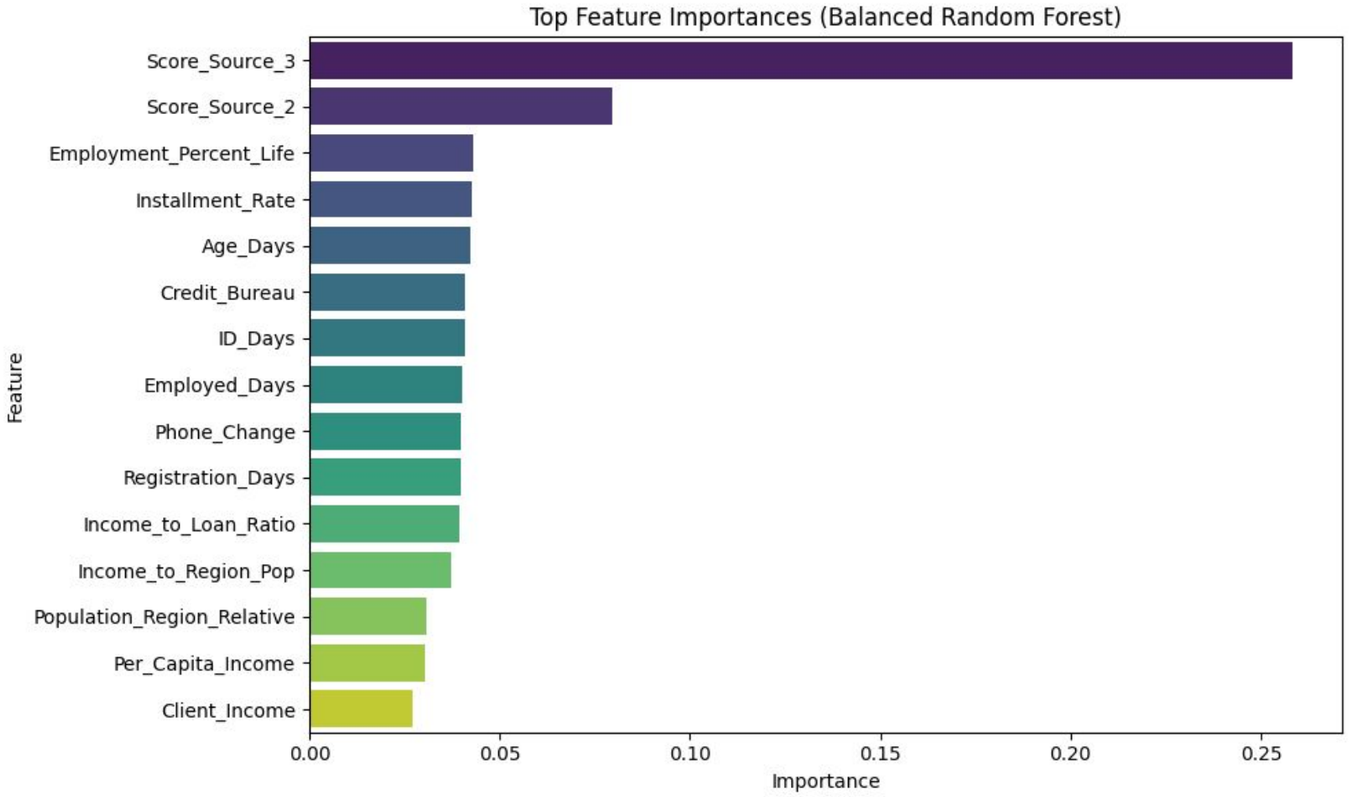
We selected a **Balanced Random Forest Classifier** as the final model for our **loan default prediction** use case, prioritizing **high recall** to minimize missed defaulters (false negatives). The composite metric used for hyper-parameter tuning was:

composite_score = 0.6 × Recall + 0.3 × F1 Score + 0.1 × PR AUC

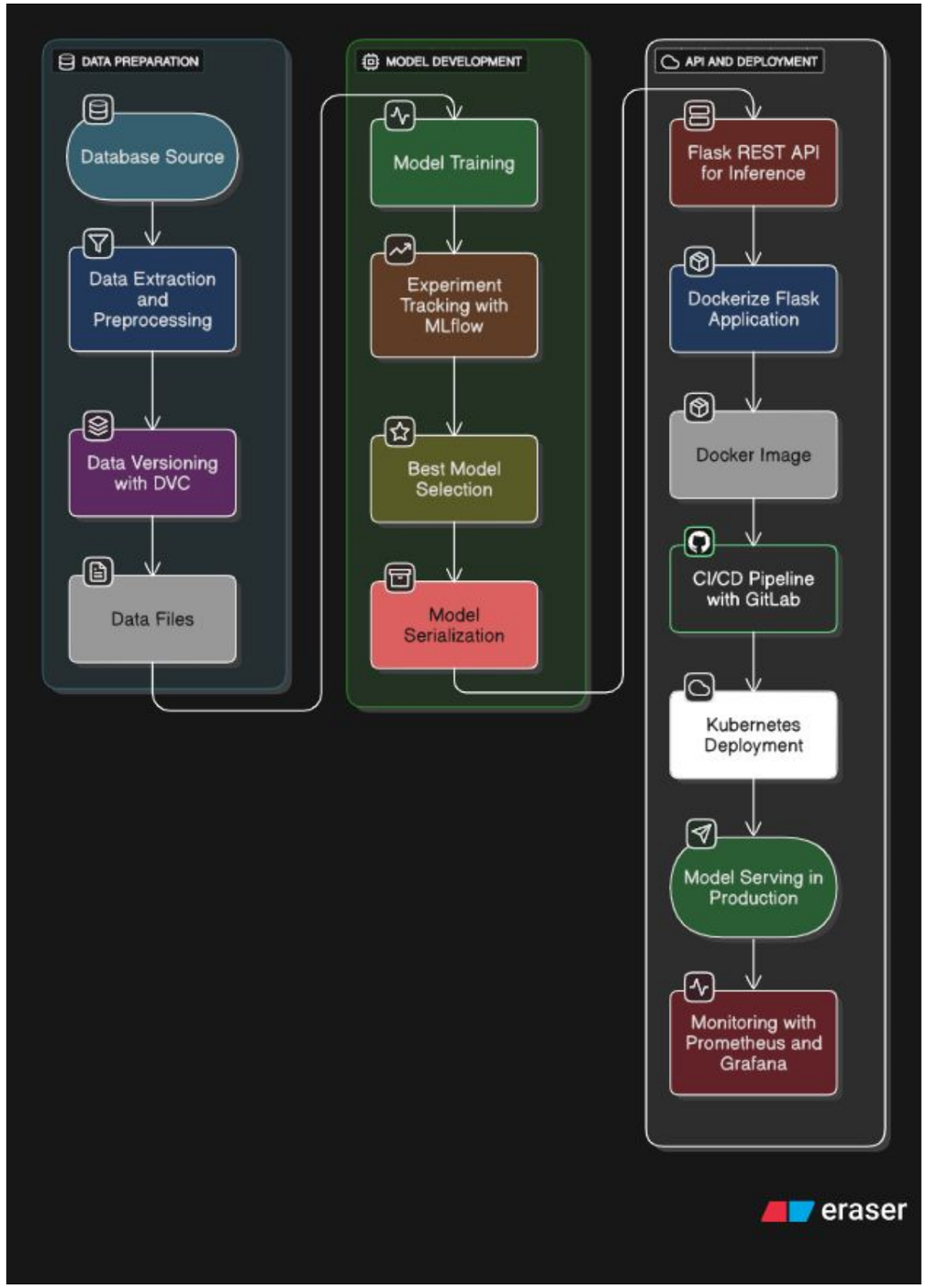




Feature Importance



- After training, we evaluated feature importance using the **Balanced Random Forest model.**
- **Score_Source_3,** a normalized external score, emerged as the most influential feature.
- Several high-impact features like **Employment_Percent_Life, Income_to_Loan_Ratio, and Installment_Rate** were **created during EDA** and proved highly valuable.
- This highlights the importance of feature engineering in improving model performance beyond the raw dataset.



Deployment Architecture & MLOps

Our solution architecture ensures robust deployment and continuous operation. This integrates model serving with comprehensive CI/CD pipelines.

Model Serving

Flask REST API, Docker, Kubernetes (GKE/EKS) ensure scalable model serving.

CI/CD

Gitlab CI/CD automates build, test, and deployment processes via DockerHub.

Monitoring

Prometheus and Grafana monitor latency and errors, with alerts for performance drops.

Drift Detection

Real-time monitoring detects model performance degradation and data drift.

Final Plan, Governance & Scalability

Our final plan focuses on deploying a robust, recall-optimised loan default predictor. This system is designed for real-time risk mitigation with strong governance.

Steps Ahead

- Finalise data transformation
- Train and log all models
- Select best model
- Deploy on Kubernetes
- Configure real-time monitoring

Security & Governance

- Role-based access controls
- Full audit trails
- API key security
- Docker image signing

The outcome is a resilient system capable of preventing missed high-risk cases effectively.