Predicting Risk of Customer Loan Default

Loan Pirates — Navigating Credit Risk in High-Stakes Waters

Frank Roth
Linda Carmichael
Christina Ntenekou

ETH Zürich

June 11, 2026

Acknowledgements
We thank Clara Isabel Meister for her guidance throughout the course and Nada Ezziti for her valuable support as teaching assistant.

Contents

List of Figures

List of Tables

1 Introduction & Problem Statement

1.1 Project Title and Team

Project Title: Loan Pirates — Navigating Credit Risk in High-Stakes Waters

The "Loan Pirates" crew consists of Linda (Canada, Google), Christina (Greece, UBS), and Frank (Germany, Swiss Re). Coming from leading global institutions in technology, banking, and reinsurance, we bring perspectives shaped by organizations operating at the forefront of their respective fields. This diverse background allows us to approach the project from complementary vantage points — spanning scalable AI systems, regulated financial environments, and quantitative risk management.

As part of the inaugural cohort of the MAS ETH in AI and Digital Technology, we frequently sailed into uncharted waters. Being among the first to complete the program meant charting our own course through evolving project structures, experimental notebooks, and the occasional hidden reef disguised as a "working" templates.

Our mission was clear: build an AI-driven credit risk system that does not simply guard the treasure chest of capital, but distributes it wisely. By enabling faster and fairer loan decisions while maintaining disciplined risk standards, the model aims to prevent worthy borrowers from being lost overboard in lengthy manual processes, while still keeping the ship financially afloat.

As students navigating regulatory storms and model-risk tides, we adjusted our sails when necessary and delivered a system designed to withstand the shifting currents of AI in regulated financial seas.

1.2 Business Context and Real-World Problem

SwissCredit Bank processes more than 50,000 loan applications per month across Switzerland and the European Union. The existing assessment process relies heavily on manual review and requires approximately three to five business days to reach a decision. This delay results in a customer attrition rate of roughly 35% among otherwise qualified applicants and generates operational costs exceeding CHF 85 per application [1].

At the same time, digital-native competitors approve loans within hours, increasing competitive pressure and customer churn. Beyond operational inefficiencies, the regulatory landscape has become increasingly demanding. Creditworthiness evaluation of natural persons is classified as a high-risk AI use case under the European Union AI Act [2]. Furthermore, automated decision-making in credit approval is subject to the safeguards of Article 22 of the General Data Protection Regulation (GDPR) [3]. In Switzerland, supervisory expectations from FINMA and the revised Federal Act on Data Protection (revFADP) impose additional requirements concerning transparency, governance, and accountability in algorithmic decision-making [4, 5].

Against this backdrop, the central problem addressed in this project can be formulated as follows:

> How can SwissCredit reduce loan decision time from 3–5 days to under 30 minutes for the majority of applications, while maintaining portfolio default rates below the current 20% threshold and ensuring full regulatory compliance?

1.3 Key Stakeholders

The development of an AI-driven credit risk assessment system involves multiple stakeholder groups whose priorities shape both the technical design and evaluation criteria of the solution. Given the high-risk classification of creditworthiness evaluation under European regulation, the system must balance predictive performance with fairness, interpretability, regulatory compliance, and operational efficiency.

Stakeholder	**Primary Concerns**
CEO / Board	Regulatory compliance; competitive positioning.
Chief Risk Officer	Model accuracy; portfolio quality.
Head of Retail Banking	Customer satisfaction; processing speed.
Compliance Officer	Regulatory adherence; audit readiness.
IT Director	System reliability; integration complexity; lifecycle stability.
Loan Officers	Tool usability; clarity of explanations.
Customers	Fair treatment; quick decisions; interpretability.

Table 1: Key Stakeholders and Primary Concerns

As shown in Table 1, stakeholder priorities introduce inherent trade-offs that directly influence system architecture and methodological choices. For example, while risk management functions prioritize predictive accuracy and portfolio stability, regulatory and compliance stakeholders require transparency, auditability, and demonstrable fairness. Operational stakeholders emphasize reliability and speed, whereas customers expect equitable treatment and understandable decision-making.

These competing objectives guided the selection of modeling approaches, the integration of interpretability mechanisms, and the inclusion of fairness monitoring within the overall system design.

2 Data & Methodology

2.1 Data Characteristics

The dataset used in this project reflects a U.S.-based consumer lending environment and contains historical loan application data including financial indicators, repayment outcomes, and demographic attributes. The prediction target is a binary variable indicating whether a borrower fully repaid the loan (non-default) or not (default).

The data exhibits characteristics typical of real-world credit datasets:

- Class imbalance, with defaults representing a minority class (see Section 2.5);
- Mixed feature types (numerical, categorical, temporal);
- Missing values and heavy-tailed financial distributions.

A notable feature of the dataset is the inclusion of a precomputed FICO score. FICO scores are widely used within U.S. lending practices as aggregate indicators of creditworthiness. However, because they are proprietary composite metrics, their internal computation logic is not fully transparent. Under European regulatory expectations regarding explainability and accountability [2, 3], reliance on opaque third-party indicators must be critically assessed.

Given that the dataset represents a realistic American lending context, the FICO score was retained. Rather than excluding it, we monitored its contribution to model predictions through interpretability analysis and fairness evaluation.

Explicit protected attributes related to race were excluded from model training to prevent direct discriminatory effects:

```
PROTECTED_FEATURES = [
    "race_american_indian",
    "race_asian",
    "race_black",
    "race_latino",
    "race_other",
    "race_white",
]
```

This exclusion aligns with GDPR restrictions on special-category data and supervisory expectations under FINMA and revFADP concerning non-discrimination and data minimization [3, 4, 5]. Protected features were retained solely for fairness auditing in the evaluation phase to detect indirect bias via correlated variables.

2.2 Regulatory and Interpretability Requirements

Creditworthiness evaluation is classified as a high-risk AI use case under the EU AI Act [2]. This classification requires documentation, risk management processes, human oversight mechanisms, and bias monitoring.

Article 22 of the GDPR establishes safeguards for individuals subject to automated decision-making, including the right to obtain meaningful information about the logic involved and to request human review [3]. Similarly, FINMA and revFADP impose expectations regarding

understandable outputs, lifecycle governance, model change documentation, and proof of non-discriminatory outcomes [4, 5].

These regulatory drivers directly influenced both feature selection and model design, particularly in the removal of protected attributes, implementation of fairness metrics, and integration of interpretability mechanisms.

2.3 Preprocessing Pipeline

A structured preprocessing pipeline was implemented to ensure reproducibility and prevent data leakage.

Removal of Leakage-Prone Features Certain variables present in the raw dataset are unavailable at decision time and would introduce data leakage if used during training. Examples include post-origination repayment information, recovery fees, settlement indicators, and future payment dates. These features were removed to ensure that model training reflects only information available at application time.

Additionally, selected runtime-available but text-heavy variables (e.g., free-text descriptions, employment titles, zip codes) were excluded in the baseline model to maintain interpretability and reduce preprocessing complexity.

Categorical Encoding and Text Conversion Categorical variables such as loan term, home ownership, verification status, purpose, and application type were converted using one-hot encoding. Dummy encoding with reference categories was applied to avoid multicollinearity.

Structured textual attributes were converted into numerical representations:

- Employment length strings were mapped to numerical years;
- Date strings were reduced to year values;
- Loan status was mapped to a binary target variable.

Missing Value Imputation Missing numerical values were imputed using median values to ensure robustness against skewed financial distributions. Categorical variables were imputed using an explicit `"Missing"` category. Infinite values were replaced with NaN prior to imputation to avoid numerical instability.

Imputation parameters derived from the training set were stored and consistently applied to validation and test data.

2.4 Feature Engineering

Feature engineering focused on constructing financially meaningful and behaviorally interpretable indicators tailored to the binary classification task. All engineered variables were derived from existing credit bureau aggregates and designed to reflect standard credit risk assessment logic.

Financial Ratio Features

Two utilization-based ratios were implemented to capture borrower leverage and credit exposure dynamics:

Installment Account Utilization Ratio (IAUR) The Installment Account Utilization Ratio measures the proportion of outstanding installment balances relative to the total granted installment credit limit:

$$\text{IAUR} = \frac{\texttt{total\_bal\_il}}{\texttt{total\_il\_high\_credit\_limit}} \tag{1}$$

where:

- `total_bal_il` represents the total current balance across all installment accounts,
- `total_il_high_credit_limit` represents the aggregated high credit / credit limit of installment accounts.

This ratio captures the remaining repayment burden relative to the total installment credit exposure.

Bank Card Utilization Ratio (BCUR) The Bank Card Utilization Ratio quantifies revolving credit pressure on bank-issued credit cards. In the implementation, utilization was derived using the open-to-buy variable:

$$\text{BCUR} = \frac{\texttt{total\_bc\_limit} - \texttt{total\_bc\_open\_to\_buy}}{\texttt{total\_bc\_limit}} \tag{2}$$

where:

- `total_bc_limit` denotes the total bank card credit limit,
- `total_bc_open_to_buy` denotes the unused available credit.

Algebraically, this corresponds to the share of utilized bank card credit relative to the total available limit.

Leverage and Affordability Ratios

Debt-to-Income Ratio (DTI) The Debt-to-Income ratio measures revolving debt relative to annual income:

$$\text{DTI} = \frac{\texttt{revol\_bal}}{\texttt{annual\_inc}} \tag{3}$$

Payment-to-Income Ratio (PTI) A proxy for installment burden was constructed by approximating monthly principal payments:

$$\text{Monthly Principal Proxy} = \frac{\texttt{loan\_amnt}}{\text{Term}_{\text{months}}} \tag{4}$$

where:

$$\text{Term}_{\text{months}} = \begin{cases} 60 & \text{if 60-month term} \\ 36 & \text{otherwise} \end{cases}$$

The Payment-to-Income ratio was then defined as:

$$\text{PTI} = \frac{12 \cdot \text{Monthly Principal Proxy}}{\texttt{annual\_inc}} \tag{5}$$

This feature approximates annualized installment burden relative to borrower income.

Credit Utilization Metrics

Revolving Utilization

$$\text{Revolving Utilization} = \frac{\texttt{revol\_bal}}{\texttt{total\_rev\_hi\_lim}} \tag{6}$$

Installment Account Utilization (IAUR)

$$\text{IAUR} = \frac{\texttt{total\_bal\_il}}{\texttt{total\_il\_high\_credit\_limit}} \tag{7}$$

Bank Card Utilization (BCUR)

$$\text{BCUR} = \frac{\texttt{total\_bc\_limit} - \texttt{bc\_open\_to\_buy}}{\texttt{total\_bc\_limit}} \tag{8}$$

These utilization measures quantify exposure across revolving and installment credit products.

Non-Linear Transformations

To allow linear models to capture curvature effects without sacrificing interpretability, selected quadratic terms were introduced:

$$\text{Squared Income} = \texttt{annual\_inc}^2 \tag{9}$$

$$\text{Squared DTI} = \text{DTI}^2 \tag{10}$$

$$\text{Squared Revolving Utilization} = \text{Revolving Utilization}^2 \tag{11}$$

Interaction Indicator

To capture joint risk amplification effects, a binary interaction indicator was defined:

$$\text{High DTI and High Utilization} = \mathbb{1}\left(\text{DTI} > 0.4 \ \wedge \ \text{Revolving Utilization} > 0.8\right) \tag{12}$$

This indicator captures scenarios in which high leverage and high credit utilization jointly occur, reflecting elevated liquidity stress.

Implementation Design

All feature transformations were implemented within a unified and reusable preprocessing function to ensure:

- deterministic transformations,
- prevention of data leakage,
- consistent application during training and inference,
- full auditability of feature construction logic.

No opaque embeddings or high-dimensional latent feature mappings were introduced. All engineered features remain economically interpretable and aligned with established credit risk modeling practice.

2.5 Handling of Class Imbalance

The dataset exhibits a pronounced class imbalance, with loan defaults representing a minority class. This imbalance can bias supervised learning algorithms toward the majority class, resulting in reduced sensitivity to default events, which are of primary business and regulatory interest.

To mitigate this issue, a data-level balancing strategy was applied during model training. Specifically, the minority class was oversampled using a controlled duplication approach implemented via a custom function (`balance_classes`).

Oversampling Strategy The implemented approach performs random oversampling by duplicating existing observations from the minority class until both classes reach equal size. Formally, let N_{major} and N_{minor} denote the number of samples in the majority and minority classes, respectively. The procedure generates $N_{\text{major}} - N_{\text{minor}}$ additional samples by randomly sampling (with replacement) from the minority class:

$$X_{\text{minor}}^{\text{aug}} \sim \text{Sample}\left(X_{\text{minor}},\ N_{\text{major}} - N_{\text{minor}}\right) \tag{13}$$

The augmented dataset is then defined as:

$$X_{\text{balanced}} = X_{\text{original}} \cup X_{\text{minor}}^{\text{aug}} \tag{14}$$

A detailed description of the oversampling procedure is provided in Appendix B. After augmentation, the dataset is randomly shuffled to remove any ordering bias introduced during resampling.

Implementation Considerations The oversampling procedure was implemented with the following design principles:

- **Reproducibility:** A fixed random seed ensures deterministic resampling.
- **Pipeline Integration:** Balancing is encapsulated in a reusable preprocessing function to ensure consistent application.
- **Data Integrity:** Feature-target alignment is preserved throughout the resampling and shuffling process.

Training-Only Application To prevent data leakage and ensure unbiased model evaluation, class balancing was applied exclusively to the training dataset:

- The training set was oversampled to achieve class balance;
- Validation and test sets were left unchanged, preserving the original class distribution.

This design ensures that performance metrics reflect real-world conditions, where class imbalance naturally occurs.

Alternative Methods Considered In addition to random oversampling, more advanced imbalance handling techniques were implemented and evaluated within the codebase, including:

- Synthetic Minority Over-sampling Technique (SMOTE);
- Adaptive Synthetic Sampling (ADASYN);
- Random undersampling of the majority class;
- Tomek Links for boundary cleaning.

While these methods provide more sophisticated mechanisms for addressing imbalance (e.g., synthetic data generation or class boundary refinement), the duplication-based approach was selected as a baseline due to its simplicity, interpretability, and minimal impact on the original data distribution.

Regulatory Perspective From a regulatory standpoint, the chosen approach supports transparency and auditability, as it does not introduce synthetic feature values that could complicate explainability. This aligns with requirements under the EU AI Act and GDPR for understandable model behavior and traceable data transformations [2, 3].

2.6 Model Candidates and Selection Rationale

Multiple classification algorithms were implemented and compared:

- Logistic Regression (interpretable baseline);
- Random Forest;
- Gradient Boosting (e.g., XGBoost).

Logistic Regression served as a transparent benchmark model. Tree-based ensemble methods were evaluated to assess potential improvements in predictive performance.

Model selection criteria included:

- Statistical performance (AUC-ROC, precision-recall);
- Fairness impact across demographic groups;
- Interpretability requirements;
- Computational efficiency;
- Regulatory compliance considerations.

2.7 Training Process and Validation

The dataset was partitioned into training and test sets. Cross-validation was applied during model development to ensure robustness and reduce variance in performance estimates.

Given class imbalance, class weighting and threshold optimization were applied to prevent majority-class dominance. Performance evaluation included both statistical metrics and business-oriented indicators such as portfolio default rate.

Fairness metrics were computed across demographic groups to detect disparities in error rates and outcome distributions.

2.8 Business Constraints

Model development was guided by explicit business objectives:

- Reduce loan decision time from 3–5 days to under 30 minutes for the majority of applications;
- Maintain portfolio default rates below the current 20% threshold;
- Provide meaningful explanations for automated decisions;
- Enable human oversight consistent with regulatory safeguards.

These constraints influenced threshold selection, model complexity decisions, and the integration of interpretability and fairness monitoring mechanisms within the overall system design.

3 Implementation & Technical Architecture

3.1 Model Selection

The following models have been tested and evaulated

- Random Forest
- XGBoost
- MLP (sklearn)
- MLP (PyTorch)

3.1.1 Overall Model Performance

Across all evaluated models, XGBoost achieved the highest AUC, demonstrating superior capability in distinguishing between default and non-default loans across varying decision thresholds. This confirms the widely observed advantage of gradient boosting methods for structured tabular data, where complex feature interactions and non-linear relationships can be efficiently captured.

Random Forest also performed strongly, providing stable and reliable results. However, its performance was consistently below that of XGBoost. This is expected, as Random Forest builds trees independently, whereas XGBoost improves performance iteratively by focusing on previously misclassified observations.

The sklearn-based MLP delivered competitive results, indicating that neural networks can learn meaningful representations even in tabular datasets. However, it did not surpass tree-based methods, highlighting the limitations of standard feedforward architectures in this domain without extensive feature engineering or architectural tuning.

The PyTorch MLP, while more flexible and computationally efficient due to mini-batch training and hardware acceleration, showed lower predictive performance. This suggests that faster training alone is insufficient, and that achieving competitive results with neural networks requires careful optimization of architecture, learning dynamics, and regularization.

Model	AUC	Accuracy	Recall	Precision	F1	Train time (sec)
Random Forest	0.7109	0.6800	0.6996	0.8753	0.7776	69.9540
XGBoost	0.7279	0.6681	0.6704	0.8871	0.7637	12.8490
MLP (sklearn)	0.7156	0.6689	0.6788	0.8798	0.7663	229.1967
MLP (PyTorch)	0.7200	0.6353	0.6203	0.8905	0.7313	598.7070

Table 2: Model performance comparison

3.1.2 Model Evaluation Metrics

The following metrics are used:

Area Under the Curve (AUC)

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{15}$$

Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

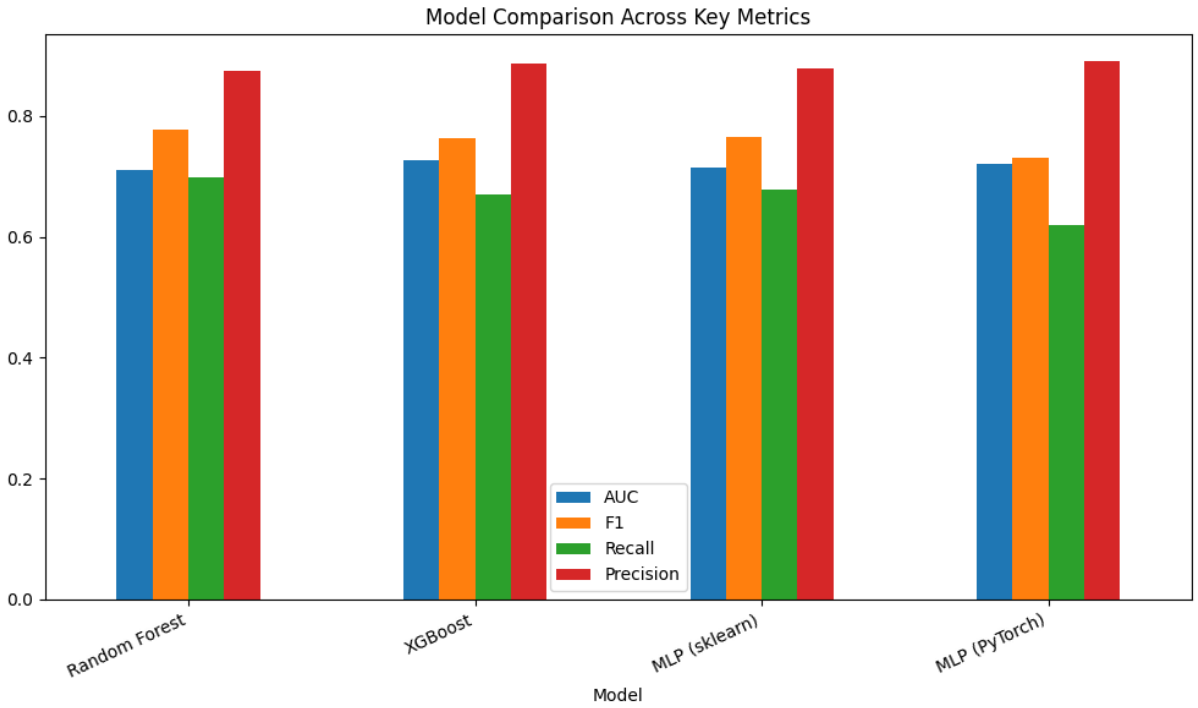

Figure 1: Model Comparison Across Key Metrics

Recall

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

Precision

$$\text{Precision} = \frac{TP}{TP + FP} \tag{18}$$

F1 Score

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

Training Time (sec)

$$\text{Train Time} = t_{\text{end}} - t_{\text{start}} \tag{20}$$

These metrics evaluate classification performance, balancing correctness, error types, and computational efficiency. A detailed definition and interpretation of these metrics is provided in Appendix C.

3.1.3 Metric Interpretation and Class Imbalance

A critical aspect of this problem is class imbalance, which significantly affects how model performance should be interpreted. While AUC provides a threshold-independent measure of ranking quality, it does not fully reflect performance at a fixed classification threshold.

The results show consistently high precision across models, indicating that predicted defaults are generally reliable. However, recall is lower, meaning that a notable proportion of actual defaults is not identified. This imbalance has important implications in a real-world credit risk context:

- High precision reduces false positives, avoiding unnecessary rejection of good customers - Low recall increases the risk of undetected defaults, which may lead to financial losses

The F1-score provides a balanced view of this trade-off, combining precision and recall into a single metric. The observed F1 values indicate that while models perform reasonably well, there is still room for improvement in detecting default cases.

3.1.4 Computational Efficiency and Scalability

The comparison of training times reveals significant differences between modeling approaches. Tree-based methods such as XGBoost offer a strong balance between speed and performance, scaling efficiently even with large datasets.

The sklearn MLP required substantially longer training time, especially during hyperparameter tuning, due to its iterative optimization and lack of GPU support. In contrast, the PyTorch implementation demonstrated much faster training, leveraging mini-batching and hardware acceleration via Apple's Metal Performance Shaders (MPS).

However, this computational advantage did not translate into improved predictive performance. This highlights a key principle in machine learning:

> Improvements in computational efficiency do not automatically lead to better model quality.

3.1.5 Model Selection and Practical Implications

From a practical perspective, XGBoost emerges as the most suitable model for this task. It combines strong predictive performance, robustness to feature scaling, and relatively efficient training. Its ability to handle tabular data effectively makes it a natural choice for credit risk modeling.

Neural networks, while flexible and powerful in other domains such as image and text processing, require more careful tuning and may not provide additional benefits for structured tabular datasets without significant effort.

The PyTorch experiment nevertheless provides valuable insight into engineering trade-offs, demonstrating how modern deep learning frameworks can reduce runtime and offer greater flexibility, even if they do not always improve predictive performance in this context.

3.1.6 Final Assessment Model Selection

In conclusion, this analysis confirms that tree-based ensemble methods, particularly XGBoost, remain the most effective approach for loan default prediction on structured data. Neural networks offer alternative modeling strategies but require additional tuning and computational considerations to achieve comparable results.

The results emphasize the importance of evaluating models across multiple dimensions, including predictive accuracy, metric trade-offs, and computational efficiency, rather than relying on a single performance measure.

Overall, **XGBoost** provides the best balance between model quality, interpretability, and operational feasibility, making it the preferred choice for this application.

3.2 System Architecture

DRAFT Describe pipeline flow and system components.

3.3 Testing and Continuous Integration

The GIT repository hosted on Github includes:

- Unit tests for preprocessing, feature engineering, model inference, and fairness metrics
- Integration tests validating end-to-end pipeline execution
- Automated GitHub Actions workflow executing tests on each pull request

This testing framework ensures robustness, regression prevention, and alignment with regulatory lifecycle expectations (e.g., FINMA governance requirements).

4 Results & Evaluation

4.1 Performance Metrics and Benchmarks

4.1.1 Statistical Metrics

AUC-ROC, Precision-Recall, F1-score, etc.

4.1.2 Business-Oriented Metrics and Rationale

Default rate, expected revenue per loan, portfolio risk impact. Explain why these metrics were chosen.

4.2 Baseline Comparison

Comparison of logistic regression (baseline) with advanced models. Include performance tables and percentage improvements.

4.3 Post-Deployment Drift Assessment

To assess whether the selected XGBoost model remains stable after deployment, the trained model was evaluated on a new drift dataset representing incoming loan applications after the original training period. The drift dataset uses the same schema as the original loan dataset, allowing the existing preprocessing and feature engineering pipeline to be reused without structural changes.

The evaluation shows that the model's ranking quality remains stable under the new data distribution. The AUC decreases only marginally from 0.7279 on the original test set to 0.7272 on the drift dataset. This indicates that the model still separates higher-risk from lower-risk applications with similar effectiveness.

However, some degradation is visible in threshold-dependent metrics. Accuracy decreases from 0.6681 to 0.6364, recall decreases from 0.6704 to 0.6165, and the F1-score decreases from 0.7637 to 0.7275. Precision remains stable at approximately 0.887, suggesting that loans predicted as default remain highly reliable. The main operational concern is therefore the lower recall, as more actual defaults may be missed under the drifted data distribution.

Dataset	AUC	Accuracy	Recall	Precision	F1
Original test set	0.7279	0.6681	0.6704	0.8871	0.7637
Drift dataset	0.7272	0.6364	0.6165	0.8872	0.7275

Table 3: XGBoost performance on original and drift data

A subgroup fairness check was also performed using the protected race indicators retained for audit purposes. The subgroup results show broadly consistent AUC and recall values across protected groups, with no severe subgroup-specific degradation detected. This supports the conclusion that the observed drift affects overall threshold performance more than it creates a clear protected-group-specific failure pattern.

Based on these results, immediate retraining is not strictly required. The model continues to provide stable ranking quality and does not show evidence of severe fairness degradation.

Nevertheless, the decline in recall should be monitored closely, because false negatives are particularly costly in credit-risk applications. The model should therefore remain under active monitoring, with regular checks for AUC, recall, false positive rate, subgroup performance, and data-distribution changes. Retraining should be triggered if recall continues to deteriorate, if subgroup disparities increase, or if business default-rate constraints are no longer met.

4.4 Error Analysis and Model Limitations

Analysis of false positives and false negatives. Discussion of known weaknesses and edge cases.

4.5 Bias Assessment and Mitigation Strategies

4.5.1 Fairness Metrics

Demographic parity, equalized odds, disparate impact.

4.5.2 Performance by Demographics

Comparison tables across protected groups.

4.5.3 Mitigation Strategies

Reweighing, threshold optimization, fairness-aware training. Show impact on fairness vs performance.

4.6 Privacy and Security Considerations

4.6.1 EU AI Act

High-risk AI obligations and system compliance.

4.6.2 GDPR Article 22

Automated decision safeguards and human review.

4.6.3 FINMA & Swiss Requirements

Lifecycle governance, audit readiness, explainability expectations.

4.7 Interpretability and Explainability Measures

Global explanations (e.g., SHAP feature importance). Local explanations for individual loan decisions. Edge case examples.

4.8 Final Model Choice and Business Rationale

Main drivers of selection: accuracy, fairness, interpretability, computational efficiency, and alignment with business constraints. Explicit justification of prioritization trade-offs.

5 Key Takeaways

Summarize lessons learned, recommendations, and reflections.(DRAFT)

- Structured credit-risk data favors strong tree-based methods over generic neural networks in this experiment.
- XGBoost provides the best practical balance of AUC, training efficiency, interpretability path, and deployability.
- Accuracy alone is insufficient in regulated finance: fairness, explainability, auditability, and human review must be designed in from the start.
- The proposed system should be introduced as governed decision support, not unchecked fully automated credit approval.

References

[1] SwissCredit Bank AG. *Predicting Risk of Customer Loan Default – Project Briefing and Executive Summary*. Internal project documentation, MAS ETH AI and Digital Technology. 2026.

[2] European Parliament and Council of the European Union. *Regulation (EU) 2024/... Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)*. Official Journal of the European Union. 2024.

[3] European Parliament and Council of the European Union. *Regulation (EU) 2016/679 (General Data Protection Regulation)*. Official Journal of the European Union. 2016.

[4] Swiss Financial Market Supervisory Authority (FINMA). *Supervisory Guidance on Governance and Risk Management in Financial Institutions*. FINMA Publications. 2023.

[5] Swiss Confederation. *Federal Act on Data Protection (revFADP)*. SR 235.1. 2023.

A Project Repository

The complete source code, documentation, and CI configuration for this project are available at:

`https://github.com/MAS-AID-AI-Project/loan_default_prediction-the-loan-pirates`

The repository contains the full machine learning pipeline, fairness evaluation modules, model interpretability components, API implementation, and automated testing framework.

A.1 Repository Structure

```
loan_default_prediction-the-loan-pirates/
|
|-- data/                  # Raw and processed datasets
|-- notebooks/             # Exploratory analysis
|-- src/                   # Core ML pipeline code
|   |-- preprocessing/
|   |-- feature_engineering/
|   |-- models/
|   |-- fairness/
|   |-- explainability/
|   |-- api/
|
|-- tests/                 # Unit and integration tests
|-- .github/workflows/     # CI configuration
|-- requirements.txt
\-- README.md
```

This testing framework ensures robustness, regression prevention, and alignment with regulatory lifecycle expectations (e.g., FINMA governance requirements).

B Class Imbalance Handling

This appendix provides a detailed description of the data balancing strategy applied during model training.

B.1 Problem Definition

Let X_{original} denote the original dataset with corresponding binary target variable $y \in \{0, 1\}$. Let N_{major} and N_{minor} denote the number of samples in the majority and minority classes, respectively, where $N_{\text{major}} > N_{\text{minor}}$.

Class imbalance is defined as:

$$N_{\text{major}} \gg N_{\text{minor}} \tag{21}$$

B.2 Random Oversampling by Duplication

To address this imbalance, additional samples are generated by randomly duplicating observations from the minority class.

As defined in Equation 13, a set of synthetic samples is drawn from the minority class:

$$X_{\text{minor}}^{\text{aug}} \sim \text{Sample}\left(X_{\text{minor}},\ N_{\text{major}} - N_{\text{minor}}\right) \tag{22}$$

Here, the sampling is performed with replacement, meaning that existing observations from the minority class may be selected multiple times.

The balanced dataset is then constructed as the union of the original dataset and the augmented minority samples (Equation 14):

$$X_{\text{balanced}} = X_{\text{original}} \cup X_{\text{minor}}^{\text{aug}} \tag{23}$$

B.3 Algorithmic Interpretation

The implemented oversampling procedure follows these steps:

1. Identify minority and majority classes;
2. Compute the required number of additional samples:
$$N_{\text{aug}} = N_{\text{major}} - N_{\text{minor}};$$
3. Randomly sample N_{aug} observations from the minority class with replacement;
4. Append the sampled observations to the original dataset;
5. Shuffle the resulting dataset to avoid ordering bias.

B.4 Properties and Limitations

The duplication-based oversampling approach has the following characteristics:

- **Preservation of original data distribution:** No synthetic feature values are introduced;
- **Low computational complexity:** The method is efficient and easy to implement;
- **Risk of overfitting:** Repeated samples may lead to increased model sensitivity to specific observations;
- **High interpretability:** All observations correspond to real data points, supporting auditability.

B.5 Consistency with Implementation

The described procedure corresponds directly to the implementation of the `balance_classes` function, which performs random oversampling via duplication and applies a deterministic random seed to ensure reproducibility.

C Detailed Definition of Evaluation Metrics

This appendix provides formal definitions and interpretations of the evaluation metrics used in the model assessment.

C.1 Confusion Matrix Notation

All classification metrics are derived from the confusion matrix:

- True Positives (TP): correctly predicted defaults
- True Negatives (TN): correctly predicted non-defaults
- False Positives (FP): non-defaults incorrectly predicted as defaults
- False Negatives (FN): defaults incorrectly predicted as non-defaults

C.2 Area Under the ROC Curve (AUC)

The Area Under the Curve (AUC), defined in Equation 15, measures the model's ability to discriminate between classes across all classification thresholds.

- AUC = 0.5: no discriminative power (random classifier)
- AUC = 1.0: perfect discrimination

It is based on the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR).

C.3 Accuracy

Accuracy (Equation 16) measures the proportion of correctly classified observations:

- Sensitive to class imbalance
- May be misleading when the majority class dominates

C.4 Recall (Sensitivity)

Recall (Equation 17) measures the proportion of actual defaults that are correctly identified:

- High recall reduces false negatives
- Particularly important in credit risk, where missing a default is costly

C.5 Precision

Precision (Equation 18) measures the proportion of predicted defaults that are actually defaults:

- High precision reduces false positives
- Important for minimizing unnecessary loan rejections

C.6 F1 Score

The F1 score (Equation 19) is the harmonic mean of precision and recall:

- Balances false positives and false negatives
- Useful when both types of errors are relevant

C.7 Training Time

Training time (Equation 20) measures computational efficiency:

- Relevant for operational deployment constraints
- Impacts scalability and real-time decision-making capability