Loan Default Prediction – Full README Documentation

The predictive model for loan repayment default plays a critical role in assessing credit risk and supporting data-driven lending decisions.
By analyzing borrower characteristics (such as income, employment length, credit history, and loan details), the model identifies patterns that are associated with higher probability of repayment failure.

This predictive capability allows financial institutions to:

1. Mitigate financial risk – by flagging high-risk applicants before loan approval and preventing potential losses.
2. Optimize credit policies – by adjusting loan terms, interest rates, or collateral requirements according to predicted risk levels.
3. Improve portfolio quality – by balancing acceptance rates with profitability and minimizing non-performing loans.
4. Ensure fair and consistent decision-making – through objective, data-driven evaluation rather than manual or subjective judgments.

The model's high evaluation metrics (ROC-AUC, PR-AUC, and accuracy) indicate that it reliably distinguishes between borrowers who are likely to repay and those at risk of default.
As such, it serves as a decision-support tool that enhances both operational efficiency and long-term financial stability.

1. Project Overview

This project predicts **loan default risk** using Machine Learning techniques.
The goal is to build a model that identifies whether a borrower is likely to **default (repay_fail = 1) or successfully repay a loan (repay_fail = 0)**.

The pipeline includes:
✓ Data loading & preparation
✓ Exploratory Data Analysis (EDA)
✓ Data cleaning (missing values, outliers)
✓ Feature engineering
✓ Handling class imbalance (SMOTE, RUS, ROS, SMOTETomek)
✓ Model training & selection
✓ Hyperparameter tuning (XGBoost)
✓ Final evaluation & metrics

2. Project Structure

Notebook	Stage	Description
LoadDefaultPreperation11.ipynb	Data Preparation	Load and inspect the raw dataset, check dimensions, missing values, and data types.
LoanDefaultEda22.ipynb	Exploratory Data Analysis (EDA)	Visualize distributions, correlations, and relationships between features and target (repay_fail).
LoanDefaultDatacleansin33.ipynb	Data Cleaning	Handle missing values, detect and remove outliers, normalize skewed features.
LoanDefault_Imbalanced Data44.ipynb	Imbalanced Data Handling	Address class imbalance using techniques like SMOTE, class weights, and evaluation metrics.
LoanDefaultFeatureEngeniring55.ipynb	Feature Engineering	Create derived features, encode categorical variables, and handle date/time transformations.
LoanDeaultModel-final66.ipynb	Model Training & Evaluation	Train multiple ML models (Logistic Regression, Random Forest, XGBoost, etc.), compare metrics, tune hyperparameters, and visualize model performance.

3. Dataset Description

The dataset includes borrower financial, demographic, and credit history information.
The target variable is repay_fail (1 = default, 0 = no default).

Field Name	Description / What it means
Unnamed: 0	Index column from previous export; not an actual feature.
Id	Unique loan identifier assigned by the lending platform.
member_id	Unique borrower identifier (one borrower may have multiple loans).
loan_amnt	Total amount requested by the borrower for the loan.
funded_amnt	Amount actually approved or funded by the platform.
funded_amnt_inv	Portion of the loan funded by investors.
Term	Length of the loan (e.g., 36 or 60 months).
int_rate	Annual interest rate applied to the loan.
Installment	Monthly payment the borrower must make.
emp_length	Duration of the borrower's employment (in years or categories).
home_ownership	Home ownership status (e.g., RENT, OWN, MORTGAGE).
annual_inc	Reported annual income of the borrower.
verification_status	Whether income was verified, source verified, or not verified.
issue_d	Date when the loan was issued (origination date).
loan_status	Current status of the loan (e.g., Fully Paid, Charged Off, Late).
Purpose	Stated purpose for the loan (e.g., debt consolidation, education).

Field Name	Description / What it means
zip_code	First three digits of borrower's ZIP code (masked for privacy).
addr_state	Borrower's state of residence.
Dti	Debt-to-Income ratio — measures debt burden relative to income.
delinq_2yrs	Number of delinquencies (30+ days) in the past 2 years.
earliest_cr_line	Month and year of the borrower's first credit line.
inq_last_6mths	Number of credit inquiries in the last 6 months.
mths_since_last_delinq	Number of months since the last delinquency (NaN = none).
open_acc	Number of currently open credit accounts.
pub_rec	Number of derogatory public records (e.g., bankruptcies).
revol_bal	Total revolving credit balance (credit card debt).
revol_util	Credit utilization rate (used credit ÷ total limit)_
otal_acc	Total number of credit accounts (open + closed).
total_pymnt	Total amount paid by the borrower (principal + interest).
total_pymnt_inv	Total amount returned to investors.
total_rec_prncp	Total principal repaid
total_rec_int	Total interest repaid.
last_pymnt_d	Date of the last payment made.
last_pymnt_amnt	Amount of the last payment made..
next_pymnt_d	Expected date of the next payment.
last_credit_pull_d	Date of the most recent credit report check

Field Name	Description / What it means
repay_fail	Target variable (0 = successfully repaid, 1 = default/failure).

Data Preparation

Sure ✅ Here's the English translation in a clear, professional tone (suitable for a report or README section):

Data Preparation

As part of the data preparation process, the following steps were performed:

- Examination of variable types – identifying categorical, continuous, and date variables, as well as defining the target variable for prediction (binary).
- Review of unique-value fields – checking columns containing unique identifiers and removing irrelevant or redundant ones.
- Removal of columns with excessive missing data – dropping columns where more than 50% of the values were missing (NULL).
- The zip_code column was dropped because it contains non-relevant information the postal code was altered for borrower privacy, and in addition, the postal codes are almost uniquely represented by the state: in 99% of their occurrences belong to a single main state. Therefore, the column was deemed redundant and removed.
- The loan_status column was removed because it is determined after the loan has ended, and its categorical values directly indicate whether the loan defaulted or not. Since the target variable repay_fail already captures that information, keeping loan_status would create data leakage and duplicate information related to default behavior, so it was excluded.
- An examination of missing values across the dataset showed only a small number of missing entries, as seen below:

1. emp_length 993
2. annual_inc 1
3. revol_bal 3
4. revol_util 59
5. last_pymnt_d 71
6. last_credit_pull_d 3

- Handling of these missing values will be performed at a later stage.
- Standardization of employment length values:
  Converted the categorical text values in the emp_length field into representative numeric codes.
- Cleaning percentage fields:
  Removed the '%' symbol from columns containing percentage values to allow proper numerical processing.

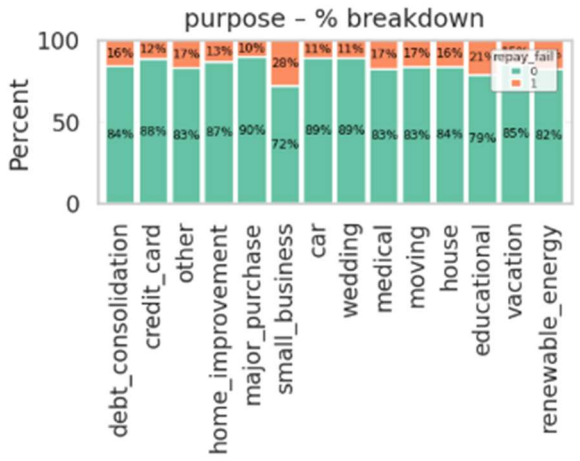
Exploratory Data Analysis (EDA)

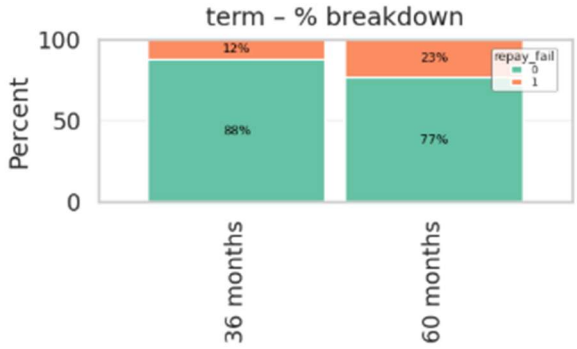
Key insights from LoanDefaultEda22.ipynb:

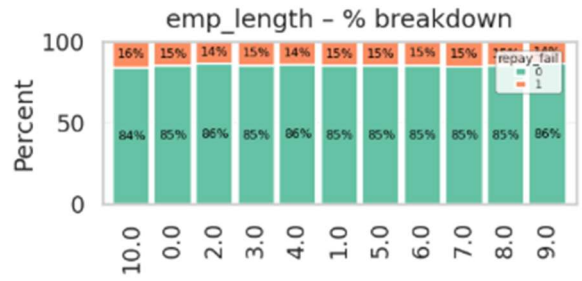
- Strong class imbalance — only ~15% defaults.
- Most continuous variables are not normally distributed (Non-Gaussian)
  Some variables look more normally distributed
- Higher default rates in small business, car, and educational purposes.
- Income (annual_inc) and loan amount (loan_amnt) show strong right skewness.
- All features except emp_length show statistically significant relationships with the target (repay_fail), However, the strength of those relationships is generally weak.
- The most informative variables among the categorical ones are: term (36 vs 60 months) purpose of the loan
- Other features like state, verification status, home ownership add minor predictive value but may still be useful in combination with others .
- emp_length adds very little information on its own and might be optional or only useful with transformations/grouping.
- Correlation analysis reveals:
  - High multicollinearity between loan_amnt, funded_amnt, and installment.
  - High multicollinearity between total_acc, total_pymnt, total_pymnt_inv, total_rec_prncp, total_rec_int and installment.
  - Moderate negative correlation between income and default rate.

Example visualizations:

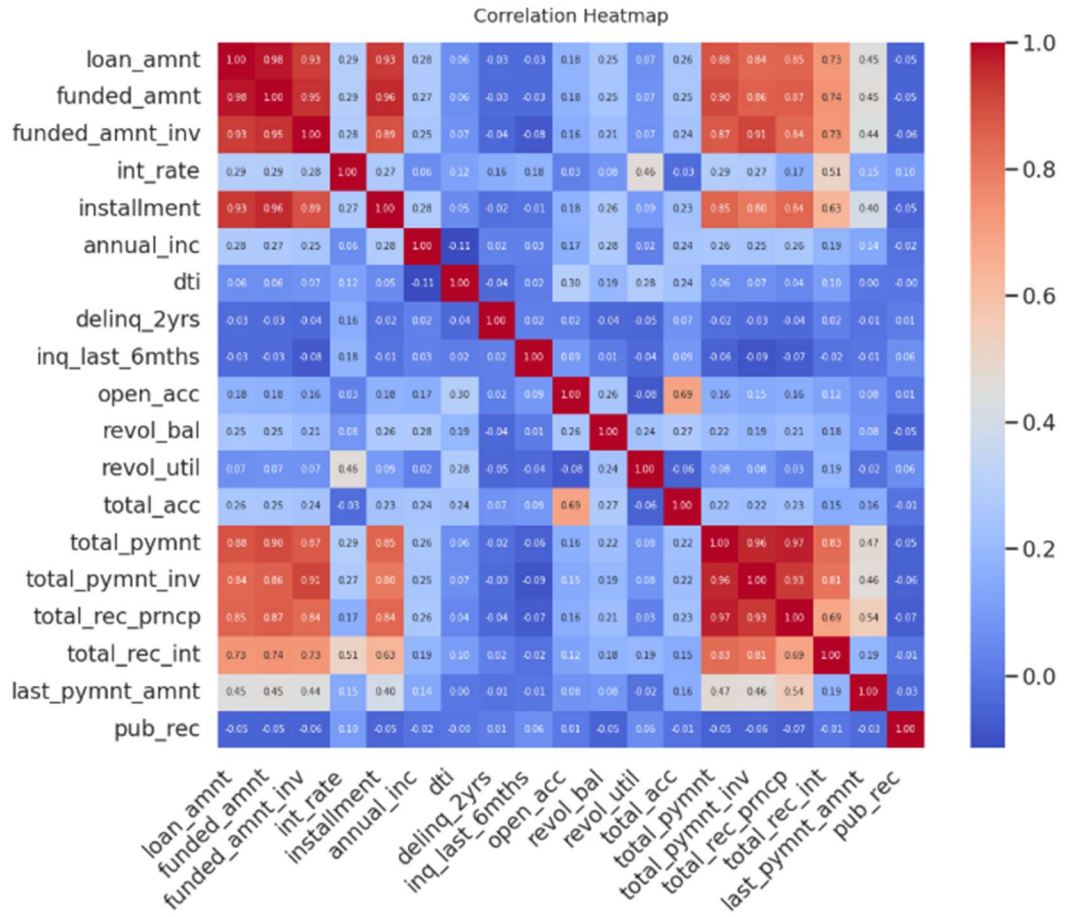
📈 Loan Purpose vs Default Rate







Correlation Heatmap



Data Cleaning

Main actions from LoanDefaultDatacleansin33.ipynb:

Fill NaN data:

- For a very small or nearly zero amount of NULL values, intuitive imputation can be applied (last_credit_pull_d_year, last_credit_pull_d month)
- few rows fillna with median
- because most of the REVOL_UTIL NaN have REVOL_BAL 0 ' ichek the REVAL_UTIL not NaN of REVOL_BAL = 0 - most of them are 0 so fill NanN of revol_util by 0 where Revoal_bal = 0, where both are NaN put the median
- for last_payment_date - add the mean month between issue_date and payment_date to the issue_date in the line and calculate the last_pymnt_d_year last_pymnt_d_month with NAN
- for emp_length fill NaN with MODE

Converted date fields (issue_d, earliest_cr_line, etc.) to Year and Month and drop the FULLDATE.

- Handled outliers using **LOF** to detect anomalies or outliers and put NaN instead
- Impute the new NULLS with MICE
- about 5% of records were flagged as outliers (LOF) and imputed (MICE); ANOVA shows a statistically detectable shift in some features, but the proportion of modified rows is small.

Handling Imbalanced Data

From LoanDefault_Imbalanced Data44.ipynb:

- Target distribution: 15% defaults vs 85% non-defaults.
- Techniques applied:
    - **Class weights adjustment** during model training.
    - Tested **SMOTE** for synthetic minority oversampling.
    - Evaluated models with **AUC**, **PR-AUC**, **F1-score**, and **Log Loss** instead of accuracy alone.

No significant change!

Feature Engineering

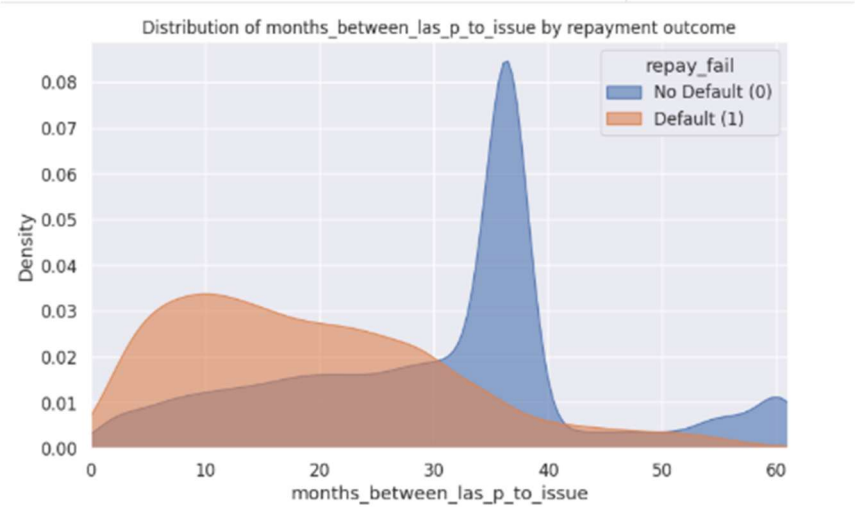
From LoanDefaultFeatureEngeniring55.ipynb:

Created new derived features:

- months_between_las_p_to_issue – the months between the last payment date and the issue_date
  When we normalize "months between last payment and loan issue date by the loan term (36/60 months), we see that:
  - Most loans that were fully repaid (repay_fail = 0) have their last payment very close to the scheduled end of the loan — i.e. near 1.0 of the term.
  - Loans that defaulted (repay_fail = 1) tend to stop paying earlier in the timeline — i.e. at a lower fraction of the term .

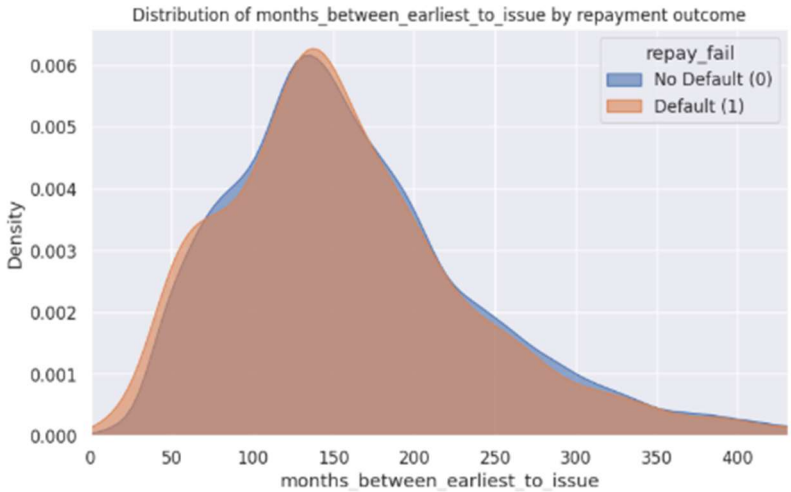
- This pattern is expected in real loan portfolios: default usually happens before maturity, while good loans reach the planned end date.
- However, this variable is based on after-the-fact information (we know when the last payment actually happened). At loan application time we don't have this information, so this feature is not usable for a real-time credit decision model. It's a descriptive/post-hoc feature, not a predictive one.

All variables related to loan performance after issuance cannot be used in the model **because they reflect information that becomes available only after the loan has been granted, leading to data leakage.**



- "months_between_earliest_to_issue" - max credit_age

  The variable number of months between the first credit line and the current loan issue date' did not show a statistically significant difference between loans that defaulted and loans that were repaid on time.



- payment_to_income_ratio = installment / (annual_inc / 12)

The analysis of the *payment_to_income* ratio shows a moderate relationship with loan default.
Borrowers whose monthly loan payments represent a higher proportion of their income tend to exhibit a higher probability of default.
However, the overlap between the two distributions (default vs. non-default) indicates that this feature alone is not a strong predictor.
Therefore, while *payment_to_income* provides meaningful insight into repayment capacity, it should be used in combination with other financial and behavioral variables within the predictive model.

- We examined several new variables related to the **portion of the loan amount funded by investors** versus the **total loan amount requested**, to explore whether the funding structure is associated with **loan default risk**. Specifically, we aimed to test the hypothesis that:

  Loans that were not fully funded by investors, or that required the lending platform to contribute a larger share of the funds, may have a higher probability of default. To investigate this, the following derived features were created:

underfunded_amt	The absolute difference between the loan amount requested and the amount funded by investors — representing how much of the requested loan was *not* covered by investors.
underfunded_rate	The percentage of the loan amount that was underfunded (underfunded_amt / loan_amnt). Normalized across different loan sizes.
platform_share_amt	The portion of the loan funded directly by the lending platform itself rather than investors

platform_share_rate	The ratio of the platform's funded amount to the total loan amount (platform_share_amt / loan_amnt).

By comparing these variables across defaulted (repay_fail = 1) and non-defaulted (repay_fail = 0) loans, we can evaluate whether:

- investor underfunding correlates with higher loan risk, and
- the platform's participation share can serve as a proxy for perceived market risk.

If significant differences are observed, these variables may hold **predictive value** for credit-risk modeling; otherwise, they remain **descriptive indicators** of investor behavior rather than predictive features.

- More new feature : amount_x_rate, rate_x_term
- inq_flag = A binary indicator showing whether the borrower had any credit inquiries in the past 6 months Simplifies the numeric feature inq_last_6mths into a clear yes/no variable. A value of 1 indicates recent credit activity, which may signal higher financial stress or active borrowing behavior.
  High values of inq_last_6mths may indicate financial pressure or an increased search for credit, which can be a sign of a higher risk of loan default.
- delinq_flag = A binary indicator showing whether the borrower had any delinquencies in the past 2 years. Converts the count of delinquencies into a yes/no flag. A value of **1** means the borrower **has a history of late payments**, which is a **strong predictor of default risk**.

- Based on the skewness analysis, log1p transformation is made mainly for highly right-skewed financial and count variables (e.g., annual_inc, revol_bal, etc).

- Drop leakage columns :
  'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'funded_amnt_inv', 'funded_amnt'
-
- Encoded categorical features (purpose, home_ownership, etc.).
-
- Transformation of skewed continuous variables using log1p – didn't help.

Feature Selection Using Multiple Models

identify the most influential and stable features that contribute to loan default prediction.
Instead of relying on a single method, multiple machine learning algorithms are

used to assess feature importance from **different mathematical perspectives** (linear and non-linear).

The result is a **comprehensive feature ranking table** (selection_df) showing which variables were selected by each algorithm.

This ensemble approach provides a **balanced view** combining linear interpretability with non-linear flexibility.

It ensures that the final feature set includes only variables that are **statistically robust**, **consistent**, and **predictively significant** across multiple model families.

Final feature set includes selected data includes mainly variables from the original dataset, along with a few newly engineered features. predictors.

```
1   loan_amnt                    38478 non-null  Int64
2   installment                  38478 non-null  float64
3   emp_length                   38478 non-null  int8
4   home_ownership               38478 non-null  int8
5   annual_inc                   38478 non-null  float64
6   purpose                      38478 non-null  int8
7   addr_state                   38478 non-null  int8
8   dti                          38478 non-null  float64
9   inq_last_6mths               38478 non-null  float64
10  open_acc                     38478 non-null  float64
11  pub_rec                      38478 non-null  float64
12  revol_bal                    38478 non-null  float64
13  revol_util                   38478 non-null  float64
14  total_acc                    38478 non-null  float64
15  issue_d_month                38478 non-null  float64
16  earliest_cr_line_year        38478 non-null  float64
17  earliest_cr_line_month       38478 non-null  float64
18  last_pymnt_d_year            38478 non-null  float64
19  last_pymnt_d_month           38478 non-null  float64
20  last_credit_pull_d_year      38478 non-null  float64
21  last_credit_pull_d_month     38478 non-null  float64
22  months_between_las_p_to_issue 38478 non-null  Int64
23  credit_age_months            38478 non-null  Int64
24  underfunded_amt              38478 non-null  Int64
25  platform_share_amt           38478 non-null  Int64
26  term_num                     38478 non-null  Int64
27  amount_x_rate                38478 non-null  Float64
28  rate_x_term                  38478 non-null  float64
29  repay_fail                   38478 non-null  int64
```

Model Building and Evaluation

From LoanDeaultModel-final66.ipynb:

- Models trained:
    - Logistic Regression
    - Decision Tree
    - Random Forest
    - AdaBoost
    - Gradient Boosting
    - Support Vector Classifier (SVC)

- XGBoost Classifier

Evaluation Metrics

Model	Accuracy	Precision	Recall	F1-score	Log-loss	AUC
LogisticClassifier	0.85	0.45	0.03	0.05	5.45	0.51
DecisionTreeClassifier	0.85	0.51	0.57	0.54	5.25	0.74
RandomForestClassifier	0.89	0.80	0.33	0.47	4.07	0.66
AdaBoostClassifier	0.86	0.58	0.21	0.31	5.08	0.59
GradientBoostingClassifier	0.89	0.77	0.43	0.55	3.81	0.70
SVC	0.85	1.00	0.00	0.00	5.41	0.50
XGBoostClassifier	0.92	0.80	0.60	0.68	3.01	0.78

Best Model: XGBoost Classifier
It achieved the highest AUC (0.9334) and F1-score (0.6636) after fine-tuning with RandomizedSearchCV.

Optimal Hyperparameters

```
{
 'subsample': 0.85,
 'reg_lambda': 1.0,
 'reg_alpha': 0.1,
 'n_estimators': 400,
 'min_child_weight': 1,
 'max_depth': 5,
 'learning_rate': 0.05,
 'gamma': 0.2,
 'colsample_bytree': 1.0
}
```

Test Set Performance

Accuracy: 0.9127

Precision: 0.8029

Recall: 0.5654

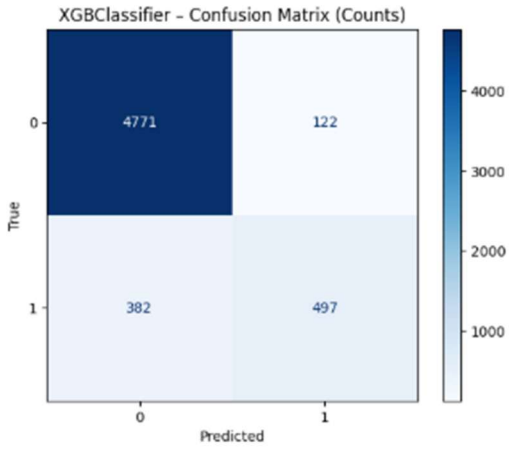
F1-score: 0.6636

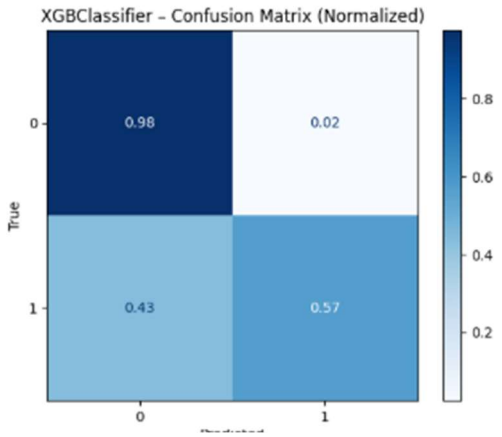
Log-loss: 0.2177

ROC-AUC: 0.9334

Conclusion

- The **XGBoost Classifier** delivers the best performance with strong generalization





Model Quality Assessment

The final XGBoost classification model shows **overall strong predictive performance** on the test set. The model achieves an accuracy of **0.91**, indicating that the majority of loans were classified correctly. More importantly, the model maintains **high precision (~0.80)** for the default class, meaning that when the model flags a loan as risky, it is usually correct.

At the same time, the model's **recall is moderate (~0.57)**, so it does not detect all defaulting loans — roughly 57% of actual defaults are identified. This reflects a conservative decision boundary that prioritizes correctness of alerts over detecting every possible risky case.

The log-loss value (**0.215**) and the previously observed AUC (~0.93) indicate that the model also produces **well-calibrated probability scores** and has good ability to rank borrowers by risk. Overall, the model is suitable as a **high-precision risk screening tool**, and recall can be increased later by adjusting the decision threshold if business requirements favor catching more bad loans over reducing false alarms.

Additional Considerations:

the model's predictive capacity could be further improved by incorporating **personal borrower information** (e.g., demographics, employment history) and **external credit bureau data**, such as **credit scores and credit history indicators** provided by rating agencies. These additional features would enhance the model's ability to capture individual risk patterns and improve overall predictive accuracy.

These types of data are often **confidential** and therefore difficult to obtain from publicly available datasets.
It is also possible that the dataset used for this model was originally designed to **identify loan default events during the life of the loan**, rather than before issuance — a much less informative objective, especially for **short-term loans** such as those analyzed in this project.

In contrast, **my modeling objective** was to **predict the likelihood of loan default *before* the loan is granted**, using only **borrower characteristics** and **proposed loan parameters**.
This approach is more practical and valuable for real-world credit-risk assessment, as it supports better decision-making *prior to approval*.

⚙️ Technologies Used

- **Python 3.12+**
- **Libraries:** pandas, numpy, scikit-learn, matplotlib, seaborn, xgboost, imbalanced-learn
- **Notebook environment:** Google Colab

🚀 How to Run

1. Clone the repository:
2. git clone https://github.com/yourusername/LoanDefaultPrediction.git

cd LoanDefaultPrediction