UNIVERSITY OF PISA

Predicting Bank Loan Default

Project Presentation – Data Mining & Machine Learning

Nicolò Bacherotti

The problem

Loan default prediction is a crucial task for financial institutions.
Identifying in advance whether a client will default on a loan allows banks to reduce financial risk, optimize credit approval processes, and minimize losses.

This project aims to build a classification pipeline that can accurately predict loan default probability based on both numerical and categorical borrower information.

This type of analysis plays a central role in **risk management strategies**, making loan default prediction a key component of modern banking and financial analytics.

Dataset Overview

45,000 loan applications with **14** columns, providing detailed information on a large population of loan applicants.
Goal: Binary Classification for loan default prediction.

Features overview:

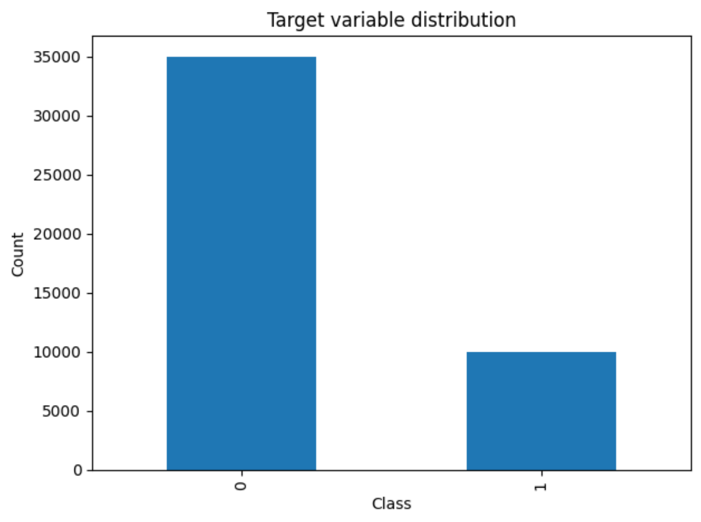
Personal Information:
person_age
person_gender
person_education
person_income
person_emp_exp
person_home_ownership

Loan Details:
loan_amnt
loan_intent
loan_int_rate
loan_percent_income

Credit & Loan History:
cb_person_cred_hist_length
credit_score
previous_loan_defaults_on_file

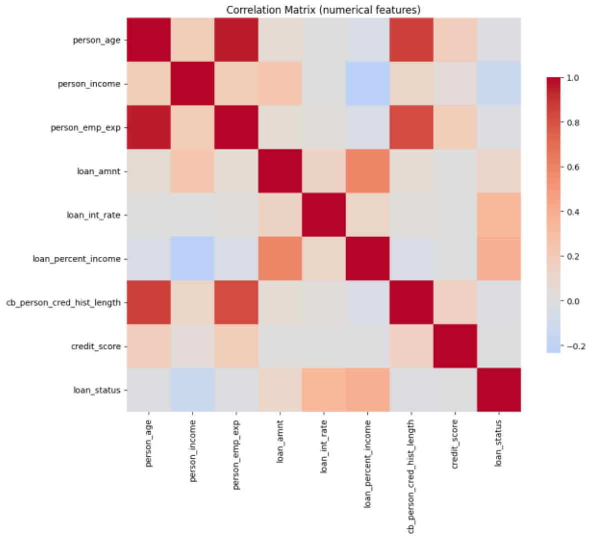
Target:
loan_status (0 = repaid, 1 = default)

Imbalanced dataset

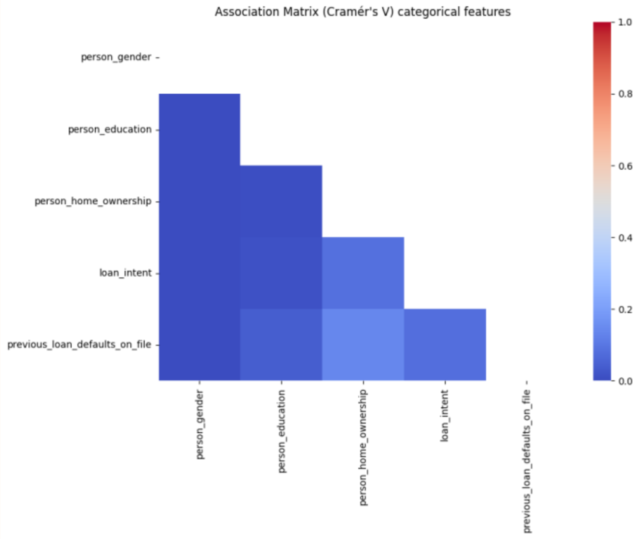


Dataset Overview

Weak correlations across numerical variables and limited correlation with the target variable



Low association among categorical variables

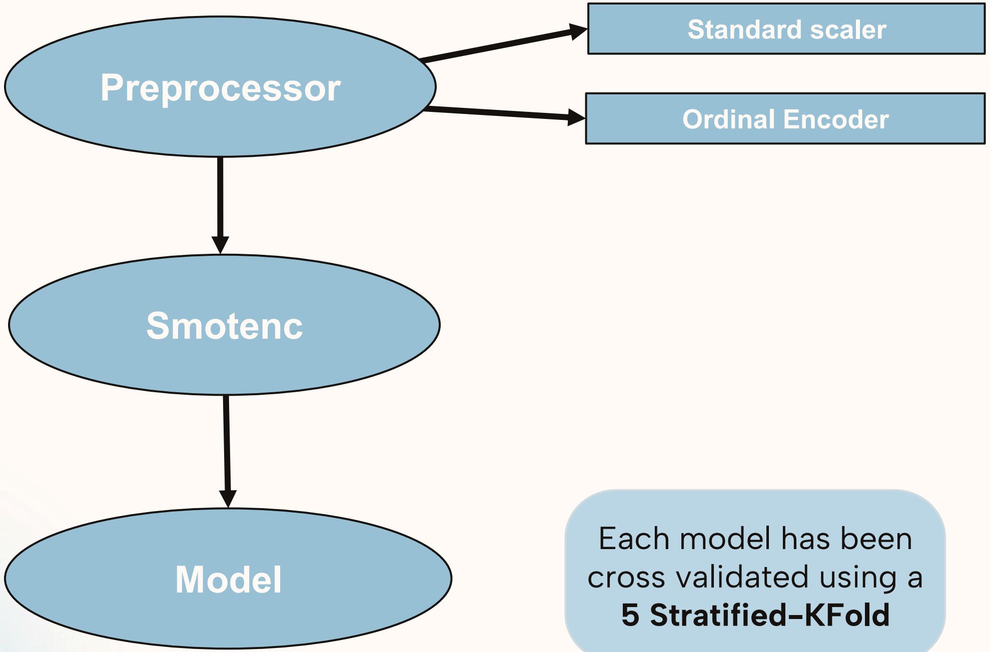


Feature Engineering & Pipeline Build

Feature Engineering:
- Person_age_bin
- Loan_int_rate_bin
- Income_to_loan
- Emp_exp_x_age
- Loan_over_score

Feature Deletion:
- previous_loan_defaults_on_file

Applied Log-Transform for skewed features:
- Detects **heavily skewed** numeric features (|skew|> 1)
- Applies **log1p(x)** to reduce skewness.



Each model has been cross validated using a **5 Stratified-KFold**

In one case a pipeline without Smotenc as intermediary is used.

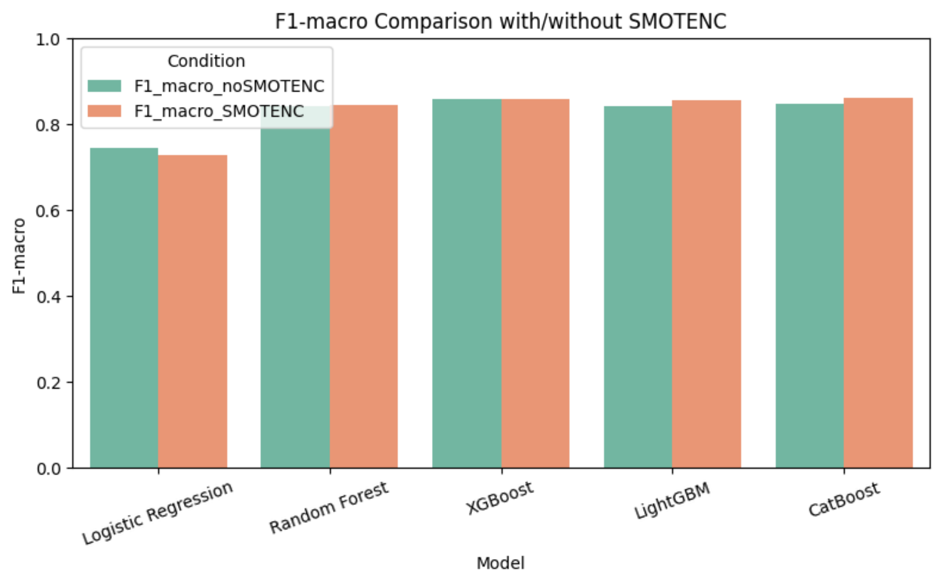
Initial Model Selection

Five models evaluated in **two pipelines**: Baseline vs with SMOTENC:

- Random Forest
- LightGBM
- Logistic Regression
- XGBoost
- CatBoost

XGBoost (0.859) and **CatBoost** (0.861) are the top performers.

Smotenc was mantained in the pipeline of the subsequent evaluations, leading in small but consistent F1_macro gains for the best models.



Grid Search Evauation

XGBoost params:

- n_estimators: [200, 400, 600],
- max_depth: [3, 5, 7, 10]
- learning_rate: [0.01, 0.05, 0.1]
- subsample: [0.6, 0.8, 1.0]

Best XGBoost params:
n_estimators: 600,
max_depth: 5,
learning_rate: 0.1,
subsample: 0.8

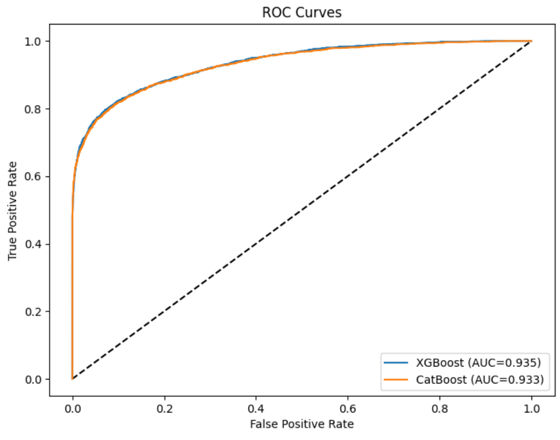
CatBoost params:

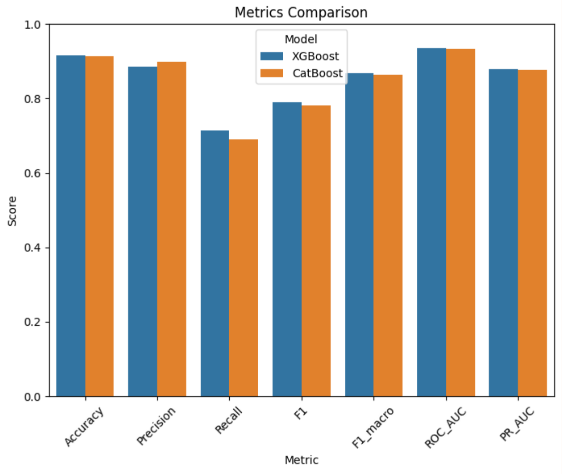
- iterations: [200, 400, 600]
- depth: [4, 6, 8, 10]
- learning_rate: [0.01, 0.05, 0.1]
- l2_leaf_reg: [1, 3, 5, 7]

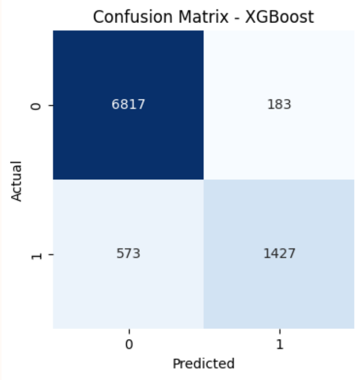
Best CatBoost params:
iterations: 600,
depth: 6,
learning_rate: 0.1,
l2_leaf_reg: 1

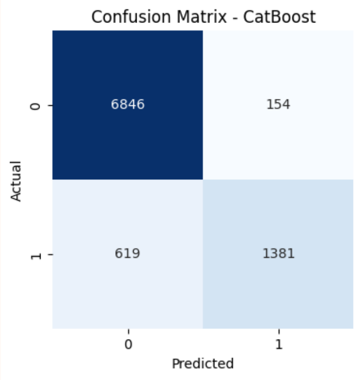
Each model was then retrained with the optimal settings and assessed performance on the test set

Models Evaluation



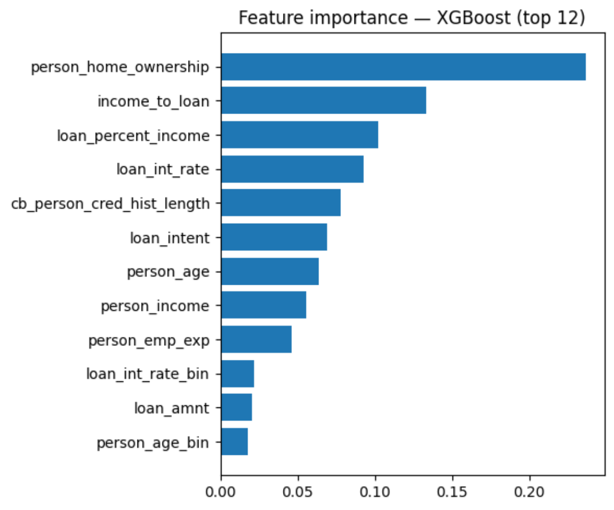


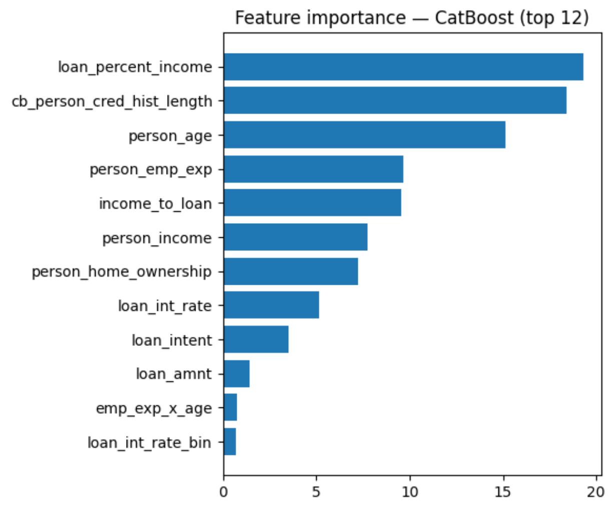




	Accuracy	Precision	Recall	F1_score	F1_macro	ROC_AUC
XGBoost	0.916	0.886	0.713	0.790	**0.869**	0.935
CatBoost	0.914	0.899	0.690	0.781	**0.863**	0.933

Feature Importance





Comparison with the State of the Art

Following the reported studies, i decided to analyze my dataset also with pipelines containing **One-Hot Encoding** and **SMOTE**.

The models tested in the cross-validation were:

- Random Forest (F1_macro=0.850)
- XGBoost (F1_macro=0.863)
- CatBoost (F1_macro=0.864)

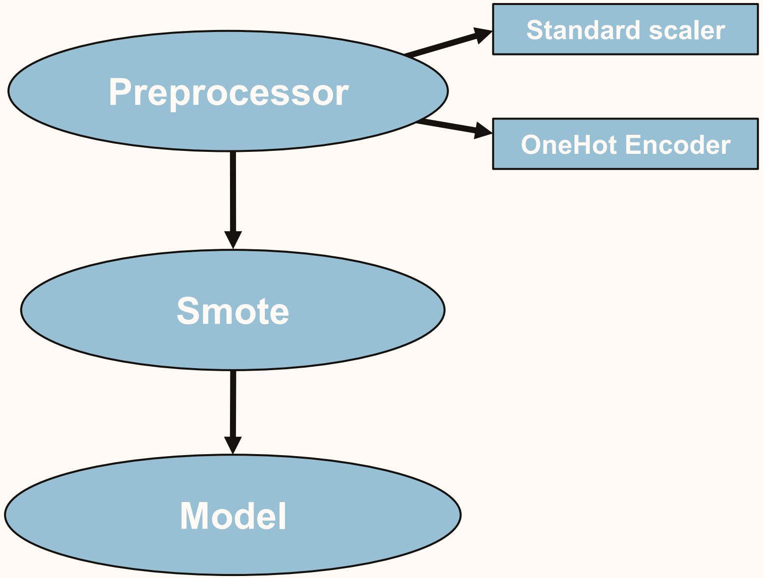
A grid search was carried out on the 2 best models:

XGBoost params:
- n_estimators: [300, 600],
- max_depth: [4,6]
- learning_rate: [0.05, 0.1]
- subsample: [0.8, 1.0]

Best XGBoost params:
n_estimators: 600,
max_depth: 6,
learning_rate: 0.1,
subsample: 0.8

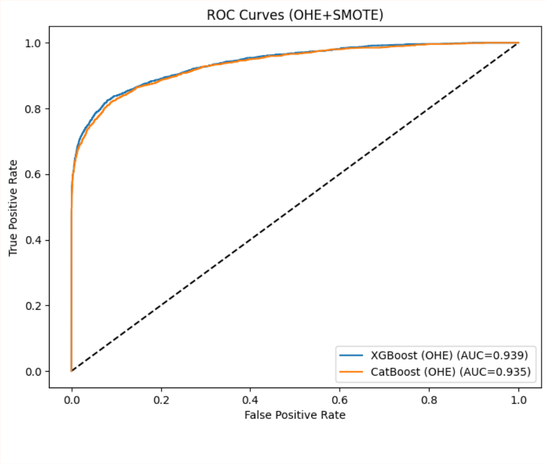
CatBoost params:
- iterations: [400, 600]
- depth: [6, 8]
- learning_rate: [0.05, 0.1]
- l2_leaf_reg: [3, 5]

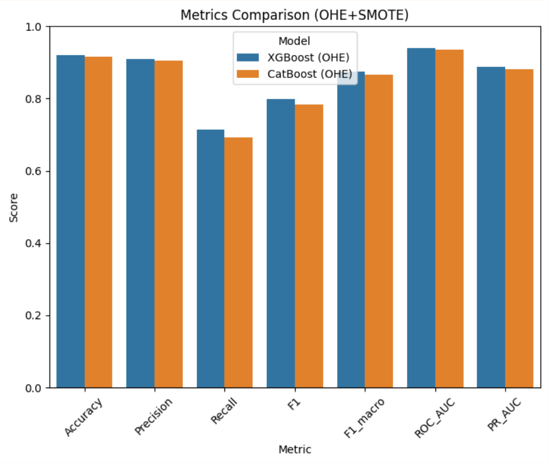
Best CatBoost params:
iterations: 600,
depth: 6,
learning_rate: 0.1,
l2_leaf_reg: 3

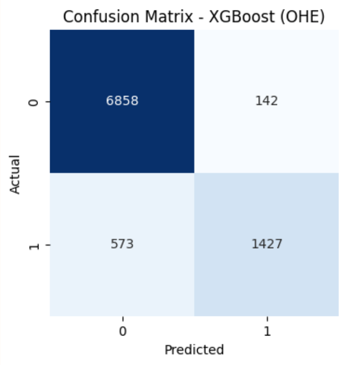


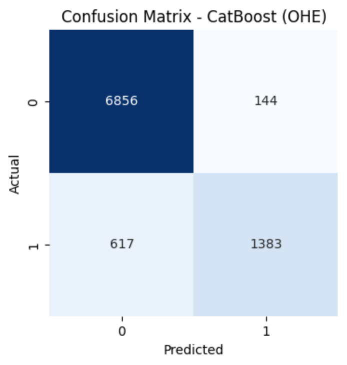
Comparison with state of art

Models Evaluation



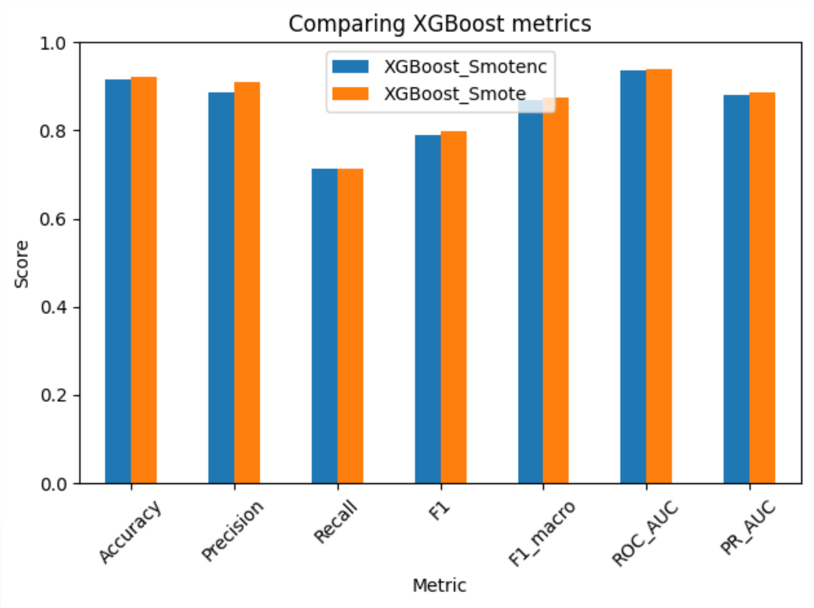






	Accuracy	Precision	Recall	F1_score	F1_macro	ROC_AUC
XGBoost	0.920	0.909	0.713	0.799	**0.875**	0.939
CatBoost	0.915	0.905	0.691	0.784	**0.865**	0.935

Comparison of the best model under the two configurations



XGBoost appears robust under both strategies;
Configuration with SMOTE has slightly
better performances in almost all the metrics.

User Interface

References

Dataset: https://www.kaggle.com/datasets/udaymalviya/bank-loan-data?select=loan_data.csv

Monje, L., Carrasco, R.A. & Sánchez-Montañés, M. Machine Learning XAI for Early Loan Default Prediction. *Comput Econ* (2025). https://doi.org/10.1007/s10614-025-10962-9

Fekadu, R., Getachew, A., Tadele, Y., Ali, N., & Goytom, I. (2022). Machine Learning Models Evaluation and Feature Importance Analysis on NPL Dataset. arXiv:2209.09638. https://arxiv.org/abs/2209.09638