Credit Default Prediction

Abstract: This report outlines the methodology and findings of a credit default prediction project using the UCI Credit Default dataset. Multiple machine learning models were implemented, including Logistic Regression, Random Forest, Adaboost and XGBoost algorithms. To address class imbalance in the dataset, resampling techniques such as SMOTE was applied. The performance of each model was evaluated with and without these resampling methods to determine the most effective approach for identifying potential defaulters.

Introduction

The dataset consists of the following primary columns:

- **ID**: ID of each client
- **LIMIT_BAL**: Amount of given credit in NT dollars(includes individual and family / supplementary credit)
- **SEX:** Gender (1=male, 2=female)
- **EDUCATION:** Educational Status (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- **MARRIAGE:** Marital status (1=married, 2=single, 3=others)
- **AGE:** Age in years
- **PAY 0 to PAY 6:** Repayment status from April to September 2005
- **BILL AMT1 to BILL AMT6:** Amount of bill statement from April to September 2005
- **PAY AMT1 to PAY AMT6:** Amount of previous payment from April to September 2005
- **default.payment.next.month:** Default payment (1=yes, 0=no)

Feature Engineering

- **TOTAL_BILL:** Sum of BILL _AMT1 to BILL_ AMT6
- **TOTAL_PAYMENT:** Sum of PAY _AMT1 to PAY_AMT6
- **PAY_TO_BILL:** Ratio of TOTAL_-PAY to TOTAL_BILL

Data Splitting

The data was splitted into 80% training and 20% testing sets.

Resampling Techniques

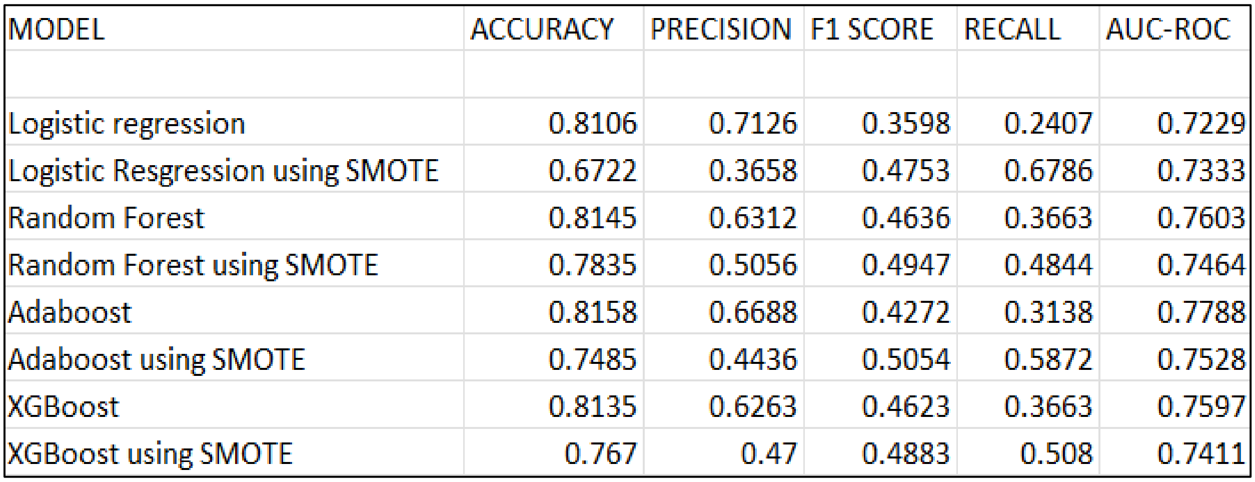
As in the Credit Default dataset classes are imbalanced, we used a resampling technique called SMOTE.

Synthetic Minority Over-sampling Technique: When the classes are imbalanced model ignore the minority class (in most of the credit default cases it is Default customer: Class1). This leads to low recall and poor performance. Instead of just duplicating the minority class which may causes overfitting SMOTE creates new synthetic samples by interpolating between existing minority class. Thus using SMOTE the model learns to catch defaulters using a balanced performance.

Modelling

- **Logistic Regression:** Logistic Regression is a linear model used for binary classification problems. The model was trained with scaled data and feature importance was analyzed to select significant features.
- **Random Forest Classifier:** It is an ensemble-based machine learning algorithm that builds multiple decision trees during training and aggregates their predictions (by majority voting in classification tasks). It is known for its robustness against overfitting and its ability to provide insights through feature importance scores.
- **AdaBoost (Adaptive Boosting):** It is an ensemble technique that combines multiple weak learners, typically decision stumps, in sequence. Each model focuses more on the errors of the previous one by adjusting weights, helping the algorithm improve its performance on difficult cases. It is effective for improving accuracy on imbalanced or noisy datasets.
- **XGBoost (Extreme Gradient Boosting):** It is an advanced boosting algorithm that builds decision trees sequentially, optimizing a loss function using gradient descent. It includes regularization techniques to prevent overfitting and is highly efficient in terms of speed and performance.

Result



MODEL	ACCURACY	PRECISION	F1 SCORE	RECALL	AUC-ROC
Logistic regression	0.8106	0.7126	0.3598	0.2407	0.7229
Logistic Resgression using SMOTE	0.6722	0.3658	0.4753	0.6786	0.7333
Random Forest	0.8145	0.6312	0.4636	0.3663	0.7603
Random Forest using SMOTE	0.7835	0.5056	0.4947	0.4844	0.7464
Adaboost	0.8158	0.6688	0.4272	0.3138	0.7788
Adaboost using SMOTE	0.7485	0.4436	0.5054	0.5872	0.7528
XGBoost	0.8135	0.6263	0.4623	0.3663	0.7597
XGBoost using SMOTE	0.767	0.47	0.4883	0.508	0.7411

The above table is showing the performance of the model considered for comparison.

- **Accuracy:** Adaboost and XGBoost without SMOTE give the highest accuracy indicating these models are effective at correctly predicting both default and non-default cases.
- **Precision:** Precision is highest for Logistic regression wherever Adaboost can also be considered as a model with good precision.
- **Recall:** Recall performance is also best for Logistic with SMOTE, that is it is better at identifying actual default cases.
- **F1 Score:** It is considered as the best metric to compare among models as it balance both recall and precision. Adaboost with SMOTE is doing good in this case.
- **AUC:** The are under ROC curve is consistent across the models with highest value in Adaboost.

Conclusion

After evaluating multiple models — including Logistic Regression, Random Forest, and boosting techniques — both with and without resampling strategies like SMOTE, **AdaBoost combined with SMOTE** emerged as the most effective model. It achieved the best balance between recall and F1-score, demonstrating its ability to accurately identify defaulters while minimizing false positives.

By effectively predicting potential defaulters, it can assist banks and credit companies in making informed lending decisions, reducing credit risk, and enhancing financial stability. Moreover, it can contribute to better credit scoring systems and early warning mechanisms for high-risk customers. The developed AdaBoost model demonstrates strong potential for supporting **data-driven risk management** in the credit industry.

For future work this model can be further improved by more diverse financial dataset , fine-tuning hyperparameters and developing more effective algorithms.

Shreya Saha

24.06.2025