Machine Learning Project

Credit Risk Classification Using Machine Learning Models

Jack Liu - Benjamin Marguin - Ambroise Megel - Marcellin Milcent
ESILV – De Vinci Higher Education

December 2025

Contents

1 Business Scope and Problem Definition

1.1 Context and Business Case

Credit risk management is a key activity for banks and lending institutions. When a bank grants a loan to a customer, it is exposed to the risk that the borrower may not be able to repay the debt according to the agreed conditions. This situation is referred to as credit default and can generate direct financial losses for the institution, such as unpaid principal and interest, as well as indirect costs related to collection procedures.

A reliable assessment of credit risk is essential for several reasons. First, from a profitability perspective, approving loans to high-risk customers increases default rates and reduces the overall financial performance of the institution. Second, from a risk management point of view, banks must maintain a balanced loan portfolio in order to remain financially stable over time and comply with regulatory constraints. Finally, accurate credit risk evaluation also contributes to fairer decision-making, as it helps avoid rejecting low-risk customers who would have been able to repay their loans.

Because financial institutions process a large number of loan applications, manual evaluation is often insufficient or inconsistent. As a result, banks increasingly rely on data-driven approaches. Machine learning models make it possible to analyze historical customer data and identify patterns associated with loan repayment behavior. These models help institutions make faster, more consistent, and more objective credit decisions.

1.2 Project Objectives

The objective of this project is to predict the credit risk of a loan applicant using financial and personal information. More precisely, the goal is to determine whether a customer is likely to repay a loan or to default. This prediction task directly supports the decision-making process involved in loan approval.

The target variable of the project is the credit risk label, which takes two possible values: Good, indicating a low-risk customer, and Bad, indicating a high-risk customer. In the dataset used in this project, the distribution of the target variable is imbalanced, with 700 customers labeled as Good and 300 labeled as Bad. This imbalance reflects real-world credit datasets, where default events are less frequent than successful repayments.

The input data consists of 20 features describing the customer profile. These features include both numerical and categorical variables related to the customer's financial situation, employment status, credit history, and personal characteristics. From a machine learning perspective, this project addresses a supervised learning problem, and more specifically a binary classification task.

From a business point of view, particular attention must be given to the Bad class. Incorrectly predicting a high-risk customer as low risk can lead to direct financial losses for the bank. Therefore, beyond overall accuracy, it is important to evaluate model performance using metrics that focus on the correct identification of risky customers.

2 Dataset Description

2.1 Dataset Source

This project is based on the Statlog German Credit Data dataset from the UCI Machine Learning Repository. The dataset describes loan applicants in a German credit context and is designed to classify each applicant as either a good or a bad credit risk. In the notebook associated with this project, the data is imported using the ucimlrepo Python package with the function fetch_ucirepo(id=144), and then transformed into a clean and structured tabular format suitable for machine learning analysis.

The dataset is highly relevant to the business problem because it contains the type of information commonly used by banks when evaluating credit applications. These variables include account status, credit history, loan purpose, employment situation, savings, housing conditions, and basic demographic information such as age. In addition, this dataset is widely used as a benchmark in credit risk modeling, which makes it particularly suitable for academic analysis and model comparison.

2.2 Variables Description

The dataset contains 1,000 observations and 21 columns in total. Among these columns, 20 correspond to input features and one corresponds to the target variable, Credit_risk, which indicates whether the customer is classified as Good or Bad. Most of the categorical variables are encoded using symbolic codes such as A11, A12, and similar values. These codes originate from the original UCI documentation and represent different qualitative categories, for example various levels of checking account status or credit history quality.

The numerical variables represent measurable quantities related to the customer and the loan. They include the loan duration expressed in months, the amount of credit requested, the installment rate. These variables provide important information about financial exposure, repayment capacity, and customer stability.

The categorical variables describe qualitative aspects of the customer profile. They include information about the status of the checking account, past credit history, the purpose of the loan. Together, these variables capture behavioral, financial, and socio-economic factors that influence the probability of loan repayment.

The target variable, Credit_risk, represents the final credit decision. Customers labeled as Good are considered likely to repay the loan, while customers labeled as Bad are considered more likely to default. Since the number of Bad cases is significantly lower than the number of Good cases, the dataset is imbalanced. For this reason, relying only on accuracy can be misleading, and additional evaluation metrics such as precision, recall, and F1-score are required to properly assess model performance, especially for the identification of high-risk customers.

3 Exploratory Data Analysis

3.1 General Statistics

An initial exploratory analysis is performed to better understand the structure and quality of the dataset before building any machine learning model. The dataset contains a total of 1,000 observations, where each observation corresponds to a loan applicant. For each applicant, 20 input features are available, describing financial, professional, and personal characteristics, along with one target variable representing the credit risk label.

During the initial inspection of the data, no missing values are detected. This confirms that the dataset is complete and does not require imputation. However, preprocessing is still necessary to handle categorical encoding and feature scaling.

3.2 Data Visualization

Data visualization is used to gain deeper insight into the distribution of variables and to detect potential anomalies. Histograms are created for numerical variables in order to analyze their distributions. These visualizations reveal that some variables, such as credit amount and loan duration, are skewed and contain extreme values. Such distributions are common in financial data and may influence model performance if not properly handled.

Boxplots are also used to visually identify outliers within numerical variables. Several features show the presence of extreme values, particularly for credit amount and duration. These outliers may correspond to uncommon but legitimate financial situations, or they may reflect atypical cases that could distort the learning process. Identifying these values is essential to decide whether outlier treatment is necessary.

In addition, a correlation heatmap is generated to examine relationships between numerical variables. The heatmap shows that most numerical features are weakly correlated, suggesting that they provide complementary information to the model. Some moderate correlations are observed, such as between credit amount and loan duration, which is expected since larger loans are often associated with longer repayment periods. Overall, no strong multicollinearity issues are detected.

The distribution of the target variable is analyzed to evaluate the presence of class imbalance. The target variable shows a clear class imbalance, with a majority of Good credit risk observations and a smaller proportion of Bad cases. This imbalance reflects real-world credit data, where default events occur less frequently than successful repayments.

Class imbalance can significantly affect the performance evaluation of machine learning models. A model that predicts the majority class for most observations may achieve high accuracy while failing to correctly identify high-risk customers.

For this reason, accuracy alone is not sufficient to evaluate model performance in this project. Additional metrics such as precision, recall, and F1-score are required, with particular attention given to the Bad class. This approach ensures a more reliable assessment of the model's ability to detect risky customers and supports better credit risk decision-making.

4 Data Pre-processing

4.1 Data Cleaning

The dataset is inspected for missing values and duplicated observations. The analysis shows that there are no missing values across all features and no duplicated rows, confirming that the dataset is clean and does not require additional cleaning steps.

4.2 Encoding and Scaling

Ordinal categorical variables are mapped to numerical values to preserve their inherent ordering, while nominal categorical features are encoded using one-hot encoding. Numerical features are then standardized using StandardScaler to ensure proper convergence of scale-sensitive models.

4.3 Train-Test Split

The dataset is split into training and test sets using an 80/20 ratio. Feature scaling is fitted only on the training set and applied to both subsets to prevent information leakage and ensure fair model evaluation.

5 Baseline Model

5.1 Logistic Regression

At this stage, a Logistic Regression model is selected as the baseline classifier. This model is well suited for binary classification problems such as credit risk assessment and is widely used due to its simplicity, interpretability, and efficiency. The model is trained on the preprocessed and scaled dataset using an 80/20 train–test split.

5.2 Baseline Results

The baseline achieves an accuracy of 79%, indicating good general performance. However, the classification report reveals a clear imbalance between classes. While the model performs well on the majority "Good" class, the recall for the minority "Bad" class is limited (recall ≈ 0.54). This result shows that the baseline model tends to favor the majority class, leading to a significant number of risky clients being misclassified. These limitations motivate the need for class imbalance handling and more advanced modeling techniques in the next steps.

6 Handling Class Imbalance

6.1 Imbalance Problem

The class distribution of the target variable reveals a strong imbalance, with a majority of "Good" credit risks compared to "Bad" ones. This imbalance leads the baseline Logistic Regression model to favor the majority class, resulting in a low recall for bad payers. In a credit-risk context, this behavior is problematic, as misclassifying risky clients represents a significant business risk.

6.2 Resampling Techniques

To address this issue, we use RandomOverSampler, which balances the dataset by duplicating minority class samples, while SMOTE generates synthetic samples based on feature-space interpolation. In both cases, the class distribution becomes perfectly balanced after resampling.

6.3 Impact on Model Performance

Applying ROS significantly increases the recall of the “Bad” class (from 0.54 to 0.76), leading to a higher F1-score for minority class detection. However, this improvement comes with a substantial drop in overall accuracy (around 71%). SMOTE provides a more balanced outcome: minority class recall remains high (around 0.71), while overall accuracy improves compared to ROS. The ROC-AUC score of approximately 0.81 further confirms the improved discriminative ability of the model.

7 Advanced Machine Learning Models

After establishing a baseline with Logistic Regression, several advanced machine learning models were implemented and evaluated in order to capture more complex, non-linear relationships in the data.

7.1 Decision Tree

The Decision Tree model was introduced as a non-linear classifier capable of learning hierarchical decision rules. Unlike linear models, decision trees can naturally model interactions between variables and handle non-linear decision boundaries.

However, a single decision tree is known to be highly sensitive to noise and prone to overfitting, especially when the tree grows deep. This behavior was observed in our experiments, where the baseline decision tree achieved strong performance on the training data but showed weaker generalization on the test set.

After hyperparameter tuning, constraints such as maximum depth and minimum number of samples per leaf helped reduce overfitting, but the Decision Tree still remained less robust than ensemble-based methods. This model therefore mainly serves as a reference point to highlight the benefits of ensemble learning.

7.2 Random Forest

Random Forest is an ensemble learning method based on the bagging principle, where multiple decision trees are trained on different bootstrap samples of the data. By aggregating the predictions of many decorrelated trees, Random Forest significantly reduces variance compared to a single decision tree.

In this project, the Random Forest model achieved strong and stable performance on the test set. Compared to Logistic Regression, it demonstrated a clear improvement in classification metrics, particularly in terms of recall and F1-score for the minority class. This indicates a better ability to detect risky clients.

One important observation is that Random Forest sometimes achieved very high performance scores, which may suggest a risk of overfitting or data leakage. This highlights the importance of careful cross-validation and the use of separate test data to validate generalization performance.

Overall, Random Forest proved to be a robust and effective model for this credit risk classification task.

7.3 Gradient Boosting

Gradient Boosting is another ensemble method, based on the boosting principle. Unlike Random Forest, which builds trees independently, Gradient Boosting trains trees sequentially, where each new tree focuses on correcting the errors made by the previous ones.

This sequential learning strategy allows Gradient Boosting to achieve high predictive accuracy and capture complex patterns in the data. In our experiments, the Gradient Boosting model showed competitive performance compared to Random Forest, with strong ROC-AUC and F1-score values.

However, Gradient Boosting is more sensitive to hyperparameter choices such as learning rate, number of estimators, and tree depth. Careful tuning was therefore required to avoid overfitting. When properly configured, Gradient Boosting provided a good balance between bias and variance, making it a strong candidate for credit risk modeling.

8 Model Optimization and Overfitting Control

8.1 Hyperparameter Optimization Strategy

Hyperparameter optimization was carried out using GridSearchCV combined with cross-validation in order to improve model performance and generalization. GridSearchCV was chosen because the number of hyperparameters per model was limited, allowing a systematic and reproducible exploration of a compact search space.

For Logistic Regression, the regularization strength and penalty type were optimized, as these parameters directly control the bias–variance trade-off. For Random Forest, hyperparameters such as the number of trees and tree depth were tuned to reduce overfitting and improve stability.

A 5-fold stratified cross-validation strategy was used to preserve class imbalance across folds. The F1-score was selected as the optimization metric to ensure balanced performance between classes and avoid favoring the majority class. Overall, hyperparameter tuning improved the detection of the minority class, sometimes at the cost of a slight decrease in accuracy, which is acceptable in a credit risk context.

8.2 Model Complexity and Overfitting Analysis

After implementing ensemble methods such as Random Forest, Gradient Boosting, and Bagging, an analysis of model complexity was conducted to address observed performance gaps between training and testing phases. Given the relatively small size of the German Credit dataset (1,000 observations) and the use of RandomOverSampler, the models initially showed a high tendency to overfit by capturing noise and duplicated samples from the minority class.

This behavior highlights the importance of controlling model complexity, particularly for tree-based models that can easily memorize training data when allowed to grow deep. Overfitting in this context leads to poor generalization and unreliable predictions on unseen loan applicants, which is problematic from both a technical and business perspective.

8.3 Tree Depth Tuning and Impact on Generalization

To mitigate overfitting, an evaluation of the maximum depth hyperparameter was conducted by comparing the results obtained for different depth values. This analysis was made feasible by the relatively small size of the dataset. By constraining the depth of individual decision trees, the models were forced to learn broader and more general patterns in the credit data rather than memorizing specific training instances.

The results of the depth-tuning process revealed that shallower trees led to better overall performance, particularly in terms of recall for the minority "Bad" class and overall accuracy. For the Bagging Classifier, reducing the maximum depth resulted in a significant improvement in generalization, with the test accuracy reaching 75

This adjustment ensured that the model remained sensitive to high-risk customers without being misled by the synthetic variance introduced during the resampling phase. The final tuned parameters therefore allowed the models to focus on the most relevant credit risk patterns rather than noise.

9 Dimensionality Reduction

Principal Component Analysis (PCA) was applied to reduce the dimensionality of the dataset and analyze potential redundancy among features. The explained variance analysis shows that a relatively large number of principal components is required to preserve most of the information, indicating that the variance is widely distributed across features.

A Logistic Regression model was trained using the PCA-transformed data. The results are similar to those obtained without PCA, with no significant improvement in classification performance, particularly for the minority class. Moreover, PCA reduces model interpretability, as principal components do not have a direct business meaning.

Therefore, PCA is included mainly for methodological completeness rather than as a technique to improve the final predictive performance in this credit risk application.

10 Ensemble learning

In an effort to further stabilize predictions and use the strengths of different algorithmic approaches, a Voting Classifier was implemented. This ensemble strategy combined the predictions of the Decision Tree, Random Forest, and Gradient Boosting models.

A Soft Voting mechanism was utilized, which aggregates the predicted probabilities of each class rather than a simple majority vote. The ensemble was further refined by optimizing the weights assigned to each constituent model. After testing various combinations, a weight ratio of 1:1:2 was selected to prioritize the most accurate model.

The models used inside seem to have a high degree of error correlation and the performance of the ensemble was as a result not any better than the best-tuned standalone model.

11 General Conclusion

The objective of this project was to develop a robust predictive framework for credit risk assessment, a critical component for financial stability and institutional profitability.

Through a structured data science workflow, we addressed the inherent challenges of credit datasets, ranging from class imbalance to the risk of model overfitting.

The study underscored that handling class imbalance is crucial. The initial baseline models were insufficient for detecting high-risk customers, but through the application of SMOTE/ROS and hyperparameter optimization, we successfully shifted the model's focus toward the "Bad" credit class. The analysis of tree depth was particularly useful in transforming an overfit model into a more reliable and accurate tool.

Based on the experimental results, the Optimized Bagging Decision Tree was selected as the final model. It achieved the most effective balance between Overall Accuracy (75%) and Recall for the "Bad" Class (84.75%), ensuring high reliability for standard loan approvals and maximizing the detection of risky applicants to minimize direct financial losses.

From a business perspective, this model provides a consistent and objective foundation for loan approval processes. It allows the institution to move beyond manual, inconsistent evaluations toward a data-driven strategy. We recommend the deployment of the Bagging-based model as a primary decision-support tool with periodic retraining to account for changes that will occur in the economic and social landscape. By implementing this solution, the bank can optimize its loan portfolio, reduce default rates, and maintain a competitive edge in risk management.

12 Limitations and Future Perspectives

Despite the strong performance achieved by the proposed models, several limitations should be acknowledged. The German Credit dataset is relatively small, which limits the ability of complex models to generalize to new data, especially when resampling techniques such as RandomOverSampler and SMOTE are used. Although model complexity was carefully controlled to reduce overfitting, the presence of duplicated or synthetic observations remains a structural limitation of the dataset. In addition, the dataset provides a simplified view of credit risk, whereas real-world credit decisions usually rely on richer information, including time-dependent variables and broader economic factors.

These limitations suggest several directions for future work. Cost-sensitive learning approaches could be introduced to better reflect the financial impact of classification errors. Finally, more advanced ensemble methods such as XGBoost or LightGBM, combined with regular model monitoring and retraining, could further improve robustness and performance in a real operational environment.

13 References

A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects.
I. Mienye & Y. Sun (2022) explain the use of ensemble models to improve predictive stability and performance beyond linear classifiers.

A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation.
A. A. Khan, O. Chaudhari & R. Chandra (2023) explain the combination of ensemble learning and data balancing techniques for imbalanced credit-risk problems.