Machine Learning for Loan Approvals: Automating Decision-Making

Federico Cesare Cattò, Andrea Matteo Re
Università degli Studi di Milano Bicocca

Abstract
This study addresses whether loan repayment can be accurately predicted using imbalanced data. We propose a Decision Tree-based ensemble model combining SMOTE oversampling and undersampling techniques. The ensemble achieved the highest AUC, outperforming individual models and alternative methods, effectively mitigating class imbalance. By leveraging complementary resampling strategies, the model balanced the dataset, improved generalization, and avoided overfitting, enabling reliable classification. These findings highlight the potential of ensemble methods for robust predictions in applications like loan repayment, fraud detection, and churn prediction, providing a foundation for future advancements in handling imbalanced datasets.

Contents

1 Introduction

Loan approvals are a fundamental pillar of the financial industry, playing a pivotal role in connecting individuals and businesses with the resources they need to achieve their goals. This process involves a thorough evaluation by lenders to assess whether a borrower meets the criteria for receiving credit. Key factors analyzed include job stability, which demonstrates the borrower's capacity to maintain a steady income over time; income level, which indicates their ability to manage loan repayments comfortably; marital status (whether married or single), as it can influence financial obligations and repayment behavior; and the property of the house, which often serves as collateral and determines the loan's security. By carefully examining these elements, lenders aim to ensure that loans are extended to individuals and entities with a strong likelihood of repayment, thereby minimizing financial risks. With the rise of digital transformation in finance, the loan approval process has evolved significantly.Advanced algorithms and automated decision-making systems are now widely used to enhance efficiency and accuracy. These technologies enable lenders to process applications faster while maintaining high standards of risk assessment. For borrowers, understanding the nuances of this process can make a significant difference in improving their chances of securing a loan. Whether for purchasing a home, expanding a business, or addressing personal financial needs, navigating the loan approval process is a crucial step toward achieving financial objectives. In this project, we aim to apply machine learning techniques to enhance the loan approval process. By leveraging historical loan application data, our goal is to build a predictive model that can assess the likelihood of a loan being approved. We will preprocess the data, handle missing values, and balance the dataset to address any class imbalances, ensuring that the model is robust and reliable. Using various machine learning algorithms, we will train the model to identify patterns and relationships within the data that are indicative of successful loan repayments. The final goal is to create a model that can provide accurate predictions on whether a loan application should be approved or denied, helping lenders make more informed, efficient decisions. Throughout the project, we will evaluate the performance of different models using metrics such as AUC.

The results of this project could significantly streamline the loan approval process, reduce risks for financial institutions, and help individuals and businesses secure the financial support they need with greater ease and fairness.

2 Dataset description and Preprocessing

2.1 Dataset and features presentation

This dataset provides a detailed overview of loan applicants' characteristics, offering valuable insights into their profiles and associated risk levels. It encompasses a wide range of features that capture demographic, financial, and employment-related information, alongside ownership details. The dataset includes both numerical and categorical attributes, making it a versatile resource for various analytical techniques, such as predictive modeling, statistical analysis, and risk assessment.

The dataset contains several key attributes. Each applicant is identified by a unique ID. Their financial capacity is represented by their income level, while their age and years of professional experience provide context about their career stage and stability. Marital status, categorized as married or single, offers insight into household dynamics. Ownership details, such as house ownership (owning or renting a house) and car ownership, further indicate financial standing. Additionally, the dataset includes the applicant's profession, city, and state of residence, which add geographic and occupational context. Job stability is captured by the duration of employment in the current job, while residential stability is reflected in the duration of residence in the current house. Lastly, a critical feature, the Risk Flag, serves as a binary indicator of loan risk, where 1 represents a flagged risky applicant and 0 represents a non-risky applicant. With 252,000 entries, this dataset offers a comprehensive foundation for analyzing loan applicants' profiles and evaluating risk. Its structure enables lenders to develop informed, data-driven strategies for risk assessment and decision-making in the lending process, enhancing both efficiency and reliability.

2.2 Data Preprocessing

In this chapter, we detail the steps undertaken to prepare the dataset for analysis and model training. Data preprocessing serves as a crucial stage in machine learning workflows, as it ensures the dataset is suitable for modeling by addressing issues like missing values, inconsistent formats, and redundant information. The processes described here—Data Exploration, Feature Encoding, Feature Engineering and Standardization, and Feature Selection—are designed to enhance data quality, reduce noise, and maximize the predictive power of machine learning models.

2.2.1 Data Exploration

The data exploration phase focused on understanding the dataset's structure and identifying potential issues. The dataset was checked for missing entries, and appropriate strategies such as imputation or removal were considered to handle any detected missing values. Missing data can introduce bias or reduce the statistical power of a model, and imputation helps maintain the dataset's integrity by filling in reasonable estimates based on available information. Outlier detection methods, such using boxplots, were employed to identify data points that deviated significantly from the expected distribution. Outliers can distort statistical analyses and degrade model performance, so their treatment was tailored to their potential impact. The distribution of the target variable was examined to assess balance or imbalance [1], as an imbalanced target can lead to biased predictions. Addressing imbalance typically involves resampling techniques such as oversampling the minority class or undersampling the majority class. An initial analysis suggested the presence of many duplicate entries; however, a more detailed investigation, which included the "Id" variable as a unique identifier for each individual, confirmed that no duplicates were present.

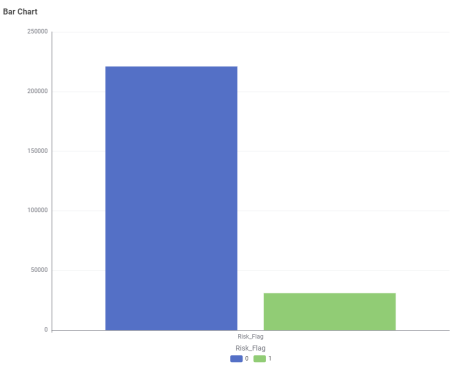

Figure 1: Risk_Flag Bar Chart: Unbalanced Class

2.2.2 Feature Engineering and Standardization

Feature engineering involved the creation of new variables from the existing ones to enhance the dataset's predictive power. This process leverages domain knowledge to uncover relationships or patterns that may not be apparent in the original variables. Examples of feature engineering include combining variables, creating interaction terms, or transforming data into new formats. The following variables are derived from the original dataset:

- Income per Age: The average earnings per year of life, calculated as the ratio of income to age. This metric helps assess how much a person earns relative to their age, providing insight into their economic standing.
- Experience per Age: The percentage of life spent working, calculated as the ratio of work experience to age. This can identify individuals with more or less experience compared to their age group.
- Job Stability: The ratio of years spent in the current job to the total years of experience. This indicator helps determine if a person is likely to stay in a job or change positions frequently.
- House Stability: The ratio of years spent in the current house to total years of life. This measure helps to understand if a person tends to move frequently, which may indicate a lack of stability.
- Age-Job Interaction: The interaction between age and job stability is analyzed to assess whether an individual's employment stability is influenced by their age.
- Income and Experience Interaction: The ratio of income to experience, adjusted by adding 1 to experience to prevent division by zero. This balances income with years of experience.
- Average Time in Previous Households: This metric is derived from the difference between age and years in the current house. It shows the average duration a person stays in each previous household.

These variables help to offer deeper insights into the relationship between age, income, experience, job stability, and housing stability.

Additionally, the dataset was standardized to address the issue of differing ranges and measurement scales across variables. Standardization involves rescaling features to have a mean of zero and a standard deviation of one, ensuring consistency across variables and enabling the effective training of machine learning models . Standardization ensures that all features contribute equally to the model, preventing variables with larger scales from dominating those with smaller scales.

2.2.3 Feature Encoding

To enable the use of categorical variables in machine learning algorithms, we converted categorical features into numerical representations. Machine learning models often require numerical inputs as they perform mathematical computations that cannot directly interpret categorical data. Specifically, Car_Ownership and House_Ownership were encoded using binary encoding, which assigns 0 and 1 to the two possible categories, maintaining simplicity and interpretability. The Married/Single variable was similarly transformed into binary format. For Profession, a label encoding technique was employed to assign unique numerical values to each profession. Label encoding is efficient but can introduce ordinal relationships between categories, which may not always be desirable. To represent geographical categories such as State and City, label encoding was applied as it effectively condenses information while retaining the distinctiveness of each category.

2.2.4 Feature Selection

Feature selection was performed to eliminate redundant or irrelevant variables, enhancing computational efficiency and potentially improving model performance. Irrelevant features add noise, while redundant features can lead to multicollinearity, where highly correlated predictors degrade model stability. The correlation matrix was used to identify and remove highly correlated variables. This technique is based on Pearson's correlation coefficient, which measures the linear relationship between two variables. Features with high correlation were flagged as candidates for removal to avoid redundancy. Feature selection not only simplifies the model but also reduces the risk of overfitting, leading to better generalization on unseen data.

By completing these pre-processing steps, the data set was made ready for model training, ensuring both efficiency and effectiveness in subsequent analysis stages.

3 Modeling

In this chapter, we describe the modeling phase of the machine learning pipeline, focusing on the selection and evaluation of models. Several algorithms were considered, including XGBoost, Logistic Regression, Naive Bayes, Decision Trees, Random Forests and Neural Networks. These models were selected for their diversity in approach, interpretability, and proven effectiveness in classification tasks. The modeling process involved the application of different sampling techniques to address class imbalance and the exploration of ensemble methods to improve predictive performance.

3.1 Machine Learning Models

The models chosen span various methodological paradigms, offering a range of strengths:

- **Logistic Regression** is a statistical method used for binary classification problems. It models the relationship between the dependent variable (target) and one or more independent variables (features) using a sigmoid function. The model outputs probabilities that can be thresholded to classify observations into distinct categories. Its interpretable coefficients provide insights into the influence of individual predictors on the target[4].

- **Naive Bayes** is a probabilistic classifier based on Bayes' theorem. It assumes that features are conditionally independent given the target class, an assumption that simplifies computations and allows the model to perform efficiently even with high-dimensional data. Despite its simplicity, Naive Bayes often delivers strong performance on text classification and other problems with clear probabilistic relationships[1].

- **Decision Trees** create a tree-like model of decisions by recursively splitting data based on feature values to maximize information gain or reduce impurity (e.g., Gini index or entropy). They are highly interpretable, as the path from the root to a leaf represents a decision rule, making them particularly useful for understanding the decision-making process[3].

- **Random Forests** build an ensemble of decision trees by training each tree on a random subset of the data and features. This approach reduces overfitting and increases robustness by aggregating predictions from multiple trees. Random Forests are versatile and can handle

both classification and regression tasks effectively[5].

- **XGBoost** (Extreme Gradient Boosting) is a powerful implementation of gradient boosting that optimizes model performance through advanced regularization techniques, such as L1 and L2 penalties, and efficient handling of missing data. It iteratively builds models that correct the errors of previous iterations, focusing on difficult-to-classify instances, making it particularly effective for structured datasets[6].
- **Neural networks** are machine learning models inspired by the human brain's structure and function. They consist of interconnected layers of artificial neurons that process data and learn to recognize patterns. Each neuron receives input, applies a transformation, and passes the output to the next layer. Through training, neural networks adjust the weights of these connections to minimize errors, enabling them to perform complex tasks such as image and speech recognition, natural language processing, and more[2].

3.2 Metrics

In this chapter, we present the primary metrics used to evaluate the performance of machine learning (ML) models.

- **Confusion Matrix:** A confusion matrix is a table that shows the model's predictions compared to the actual values, highlighting true positives, false positives, true negatives, and false negatives. It helps in understanding the model's errors.
- **Accuracy:** Accuracy represents the percentage of correct predictions out of the total predictions made. It is defined as:

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

where:

  - TP: True Positives
  - TN: True Negatives
  - FP: False Positives
  - FN: False Negatives

  This metric is useful when classes are balanced but can be misleading in the presence of class imbalance.

- **Precision:** Precision measures the proportion of true positives among all instances classified as positive:

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

  It is crucial when false positives have a significant impact.

- **Recall (Sensitivity):** Recall indicates the proportion of true positives among all actual positive instances:

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

  It is essential when false negatives are critical.

- **F1-Score:** The F1-Score is the harmonic mean of precision and recall, providing a balance between the two metrics:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

  It is useful when a compromise between precision and recall is necessary.

- **Area Under the ROC Curve (AUC-ROC):** The AUC-ROC represents the model's ability to distinguish between classes. An AUC close to 1 indicates an excellent model, while an AUC near 0.5 suggests a model with no discriminative power.

3.3 Classification and Results

Classification was approached using various methodologies to maximize model effectiveness and address challenges posed by the dataset, such as class imbalance.

3.3.1 The Hold out method

The holdout method was employed as a baseline validation technique. The dataset was split into training (70%) and testing (30%) subsets, ensuring the independence of evaluation data from the training process. This approach provides an unbiased estimate of model performance on unseen data. Results from this approach highlighted the baseline performance of the models, serving as a benchmark for subsequent techniques [1].

Model	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.88	0.07	0.00	0.00	0.50
Decision Tree	0.89	0.53	0.57	0.55	0.75
Random Forest	0.90	0.62	0.50	0.55	0.73
Naïve Bayes	0.88	0.00	0.00	0.00	0.50
Neural Network	0.88	0.59	0.00	0.00	0.50
XGBoost	0.88	0,00	0.00	0.00	0.50

Table 1: Performance metrics, including AUC, for various machine learning models.

Among the analyzed models, Decision Tree and Random Forest emerge as the most promising. In particular, Decision Tree combines good performance in accuracy and recall with a higher AUC value than the other models, making it an ideal choice for the problem at hand. The results of the Logistic Regression, Naïve Bayes, Neural Network and XGBoost models suggest the need for further optimization or better feature selection to improve their performance.

3.3.2 Hold out method and SMOTE

To address the class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training set. SMOTE generates synthetic examples for the minority class by interpolating between existing samples, effectively reducing the risk of overfitting compared to simple oversampling. This technique was used to train models, particularly those sensitive to class distributions, to improve their ability to predict minority class instances. The performance of each model trained with SMOTE is detailed in the results section, showing improved recall for minority class predictions [2].

Model	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.13	0.12	0.99	0.22	0.50
Decision Tree	0.88	0.50	0.82	0.62	0.85
Random Forest	0.90	0.56	0.77	0.65	0.84
Naïve Bayes	0.33	0.13	0.79	0.22	0.53
Neural Network	0.33	0.13	0.80	0.23	0.53
XGBoost	0.53	0,14	0.53	0.21	0.53

Table 2: Performance metrics, including AUC, for various machine learning models.

Decision Tree and Random Forest are confirmed as the best performing models. Decision Tree shows a good balance between precision (50%) and recall (82%), with an F1-score of 0.62 and an AUC of 0.85, demonstrating a remarkable ability to separate classes. Random Forest further improves accuracy (56%) and F1-score (0.65), while maintaining a competitive AUC (0.84), making it an excellent choice for the problem analyzed. In contrast, Logistic Regression, Naïve Bayes, Neural Network and XGBoost show unsatisfactory performance. Although Logistic Regression achieves 99% recall, its very low precision (12%) and F1-score (0.22) limit its effectiveness. The other models achieve similar results, with an AUC just above that of a random classifier (0.53). Even in this configuration, Decision Tree and Random Forest clearly emerge as the most reliable models, with Random Forest offering an overall improvement in key evaluation metrics.

3.3.3 Hold out method and Undersampling

In addition to SMOTE, undersampling of the majority class was explored as an alternative strategy. By reducing the number of majority class examples, the class distribution was balanced, simplifying the decision boundary for certain algorithms. However, undersampling carries the risk of discarding potentially valuable data, necessitating a careful evaluation of its impact on model performance. Model performance metrics are presented to assess the trade-offs associated with this approach [3].

Model	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.16	0.12	0.96	0.23	0.50
Decision Tree	0.84	0.43	0.86	0.57	0.85
Random Forest	0.88	0.49	0.82	0.62	0.85
Naïve Bayes	0.33	0.13	0.79	0.22	0.53
Neural Network	0.79	0.16	0.17	0.17	0.53
XGBoost	0.51	0,14	0.59	0.23	0.54

Table 3: Performance metrics, including AUC, for various machine learning models.

Again, Random Forest and Decision Tree stand out as the most effective models. Decision Tree achieves 86% recall and an AUC of 0.85, showing a good ability to correctly identify classes, with an F1-score of 0.57. The Random Forest further improves accuracy (49%) and F1-score (0.62), while maintaining the same level of AUC (0.85), confirming itself as a robust and balanced choice. The other models, including Logistic Regression, Naïve Bayes, Neural Network and XGBoost, continue to show lower performance. Logistic Regression has high recall (96%), but low precision (12%) and AUC of 0.50 indicate poor overall effectiveness. Neural Network and XGBoost achieve an AUC slightly higher than that of a random classifier (0.53 and 0.54), highlighting the need for further optimization. In conclusion, Random Forest and Decision Tree are confirmed as the best solutions, with Random Forest providing an optimal balance between the metrics analyzed.

3.4 Ensemble Modeling

Ensemble models are machine learning techniques that combine multiple basic models to improve predictive performance compared to using a single model. The idea behind them is that by aggregating predictions from different models, individual errors can be compensated for, resulting in more accurate and robust results. Among the main ensemble techniques are bagging, which reduces variance by combining models trained on different subsets of the dataset, and boosting, which aims to reduce bias by sequentially building models focused on the errors of previous ones. These approaches are particularly effective in complex scenarios, where individual models may not adequately capture the structure of the data. After careful comparative analysis, the model we decided to create is an ensemble of two Decision Trees, each trained with different sampling techniques to handle data imbalance. The decision to combine these two models was dictated by the excellent performance previously considered.

The ensemble model combines the predictions of two separate Decision Trees:

- **Decision Tree with Undersampling**: This tree was trained on a subset of the data obtained by reducing the amount of the majority class to balance the class distribution. Undersampling was accomplished by randomly selecting a number of samples from the majority class equal to the number of samples from the minority class.
- **Decision Tree with SMOTE**: In this case, the tree was trained on a dataset obtained by increasing the number of samples from the minority class. Oversampling was performed using the Synthetic Minority Over-sampling Technique (SMOTE), which generates new synthetic samples by interpolating between existing minority class samples.

To evaluate the effectiveness of the ensem-

ble model, several performance metrics were again used [4], with a focus on AUC, which measures the model's ability to distinguish between different classes. A higher area-under-curve value indicates a better discrimination ability of the model [2].

Model	Accuracy	Precision	Recall	F1-score	AUC
Ensemble Model	0.89	0.53	0.90	0.67	0.89

Table 4: Performance metrics, including AUC, for various machine learning models.

The Ensemble Model emerges as the best performing model among those analyzed. With an accuracy of 89%, precision of 53%, and recall of 90%, the model demonstrates a remarkable ability to correctly identify classes while maintaining a good balance between sensitivity and specificity. The F1-score of 0.67 reflects this effectiveness, while the AUC of 0.89 confirms the excellent discriminatory ability of the model.

In summary, the Ensemble Model stands out as the optimal solution due to its superior overall performance compared to the other models analyzed.

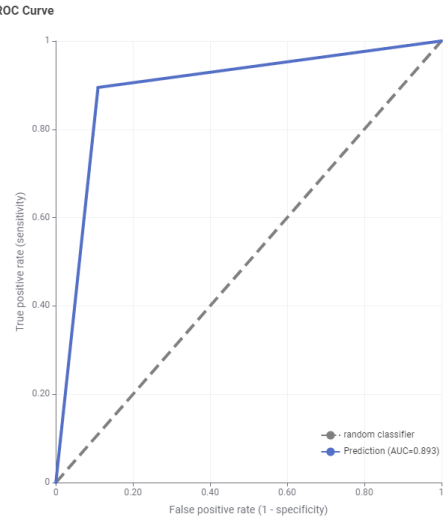

Figure 2: ROC cruve of the Decision Trees combination

All analyses were performed with Knime (5.4.0 version).

4 Limitations

4.1 Computational Constraints

One of the most significant limitations was the lack of sufficient computational resources. This constraint affected the ability to train and test more complex models, as some of them exceeded the hardware capacity of the available system. Specifically:

- **Model Complexity**: Certain advanced models could not be tested due to memory limitations and processing power.
- **Training Time**: The training times for some models were prohibitively long, which hindered their inclusion in the comparative analysis. As a result, the exploration of additional architectures and hyperparameter tuning was constrained.
- **Iterate Hold-Out**: Due to computational limitations, it was not feasible to implement the Iterate Hold-Out method for evaluation. This prevented a more robust assessment of model performance across multiple train-test splits, which could have provided deeper insights and reduced potential bias in performance metrics.

The computational constraints also limited the diversity of models explored. While simpler models and a few ensemble methods were implemented, the inability to experiment with more resource-intensive methods potentially restricted the identification of higher-

performing alternatives.

4.2 Potential Mitigations

Although these limitations were unavoidable given the available resources, several measures could address them in future iterations:

- Access to more powerful hardware, such as cloud-based computing platforms, to enable the training of computationally demanding models.
- Optimization of model architectures and preprocessing pipelines to reduce training time without significantly impacting performance.
- Implementation of distributed computing techniques to parallelize training and improve efficiency.
- Adoption of alternative evaluation techniques, such as stratified k-fold cross-validation, which could provide robust assessments without requiring computationally intensive iterations.

These limitations highlight how important it is to have sufficient resources to explore and test machine learning models effectively.

5 Conclusions

The results of our study demonstrated that the ensemble model, trained using both undersampling and oversampling techniques, outperformed the individual models and other configurations tested, achieving a higher AUC. This finding underscores the effectiveness of combining multiple sampling strategies, which proves essential for enhancing model performance in the presence of imbalanced data. By integrating these techniques, the ensemble model successfully addresses the challenges posed by class imbalance, ensuring that the model remains robust even when confronted with a skewed distribution of classes.

Specifically, the Decision Tree-based ensemble model effectively mitigated the impact of class imbalance, leading to a significant boost in the predictive performance of the model. The use of SMOTE (Synthetic Minority Over-sampling Technique) and undersampling not only balanced the training dataset but also improved the model's ability to generalize to unseen data, thereby enhancing its discriminatory power between the different classes. This dual approach resulted in better classification outcomes, as it allowed the model to learn from a more representative dataset while avoiding overfitting to the majority class.

The success of this approach highlights the importance of adopting innovative strategies, such as ensemble learning, in addressing problems where class imbalance is a significant challenge. It demonstrates the flexibility and adaptability of ensemble methods, which can improve model robustness by leveraging the complementary strengths of various learning algorithms and resampling techniques. Moreover, the use of ensemble methods in this context is not merely an enhancement but rather a necessity, as it enables models to tackle complex classification problems with greater accuracy and reliability.

Furthermore, this methodology has broader implications for fields where unbalanced datasets are common, such as fraud detection, medical diagnosis, and customer churn prediction. By integrating resampling techniques within the ensemble framework, we can ensure that the models are not only accurate but also reliable in real-world scenarios where data imbalance is often inevitable. The findings from this study set a solid foundation for future research, offering a benchmark for developing more sophisticated ensemble configurations. As these techniques

evolve and advance, further exploration into their application could yield even greater improvements in performance, providing valuable insights for both academic research and practical applications.

References

[1] Thomas Bayes. Naive bayes classifier. *Article Sources and Contributors*, pages 1–9, 1968.

[2] Raphael Féraud and Fabrice Clérot. A methodology to explain neural network classification. *Neural networks*, 15(2):237–246, 2002.

[3] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39:261–283, 2013.

[4] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[5] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. A review on random forest: An ensemble classifier. In *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, pages 758–763. Springer, 2019.

[6] Junyi Zhang, Xianglong Ma, Jialan Zhang, Deliang Sun, Xinzhi Zhou, Changlin Mi, and Haijia Wen. Insights into geospatial heterogeneity of landslide susceptibility based on the shap-xgboost model. *Journal of environmental management*, 332:117357, 2023.