School of Computing and Information Systems

CS610 Applied Machine Learning

Final Report

Group 3

2 July 2024

Table of Contents

1 Business Problem **1**

2 Dataset **1**

2.1 Overview 1

2.2 Preprocessing 1

2.2.1 Handling Missing Values in Modeling 1

2.2.1.1 Automatic Handling by LightGBM 1

2.2.1.2 Comparison with Traditional Models 2

2.2.2 Handling Unrealistic Values in Data 2

2.2.2.1 Identification of Unrealistic Values 2

2.2.2.2 Treatment of Unrealistic Values 2

2.3 Feature Engineering 2

2.3.1 Feature Engineering-Primary Training Set 2

2.3.1.1 New Features 2

2.3.1.2 Distribution Analysis 2

2.3.2 Feature Engineering-Auxiliary Training Set 3

2.3.2.1 Data Aggregation 3

2.3.2.2 Feature Generation 3

2.3.2.3 Reducing Complexity and Redundancy 3

3 Model **3**

3.1 First Selection 3

3.2 Experimental Analysis of Model Selection 4

3.2.1 Experimental Settings 4

3.2.2 Training and Evaluation Framework 4

3.3 Parameters 4

3.2.1 Logistic Regression 4

3.2.2 KNN (K-Nearest Neighbors) 5

3.2.3 SVM (Support Vector Machine) 5

3.2.4 XGBoost 5

3.2.5 TabNetClassifier 5

3.2.6 LGBM (Light Gradient Boosting Machine) 5

3.4 Final Selection 6

4 Results Analysis **6**

5 Conclusion **7**

6 Future Considerations and Next Steps **8**

6.1 Data 8

6.2 Model Improvement 8

References 9

1 Business Problem

Banks have been struggling financially, which means they have less money to lend out(Sermeño, 2023). This reduction in lending capacity is significant because loans are essential for helping people and businesses achieve their goals and drive economic growth. Given the current situation, it's more important than ever to accurately assess whether a loan applicant is likely to default.

Our goal is to enhance the accuracy of risk assessments for banks and lenders regarding potential loan defaults. By improving these evaluations, we aim to promote responsible lending practices. This approach enables financial institutions to extend credit to individuals in need while simultaneously safeguarding against the risk of defaults.

Ultimately, our goal is to foster a more stable lending environment through the development and implementation of improved tools and methodologies for assessing default risk. This will benefit both lenders and borrowers, and increase financial inclusion by supporting a balanced financial ecosystem.

2 Dataset

2.1 Overview

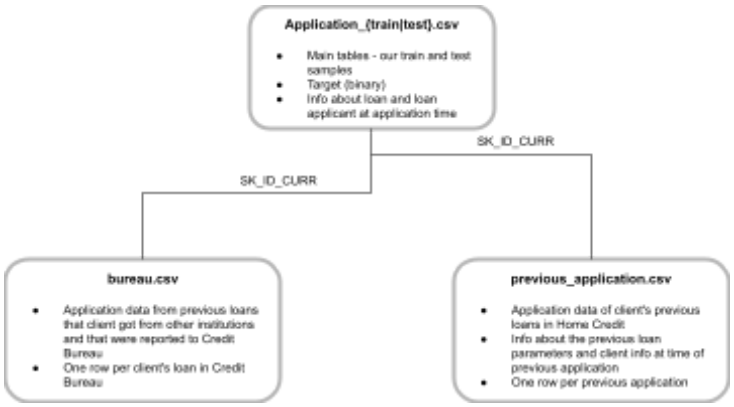

Figure1: Schema Diagram of the Datasets

Application_{train|test}.csv: This dataset serves as the primary table, segregated into two distinct sets for training (inclusive of the TARGET variable) and testing (exclusive of the TARGET variable). It encompasses static data for all applications, where each row symbolizes a singular loan within our dataset.

Bureau.csv: This dataset combines all preceding credits associated with clients, which were provided by external financial institutions and then reported to the Credit Bureau regarding clients who possess a loan in our sample. Corresponding to each loan in our sample, the dataset comprises multiple rows—each representing a credit that the client had registered with the Credit Bureau prior to the date of the application.

Previous_application.csv: This dataset contains records of all antecedent applications for Home Credit loans by clients who currently have loans within our sample. Each row in this dataset represents a prior application associated with the loans documented in our dataset.

2.2 Preprocessing

2.2.1 Handling Missing Values in Modeling

2.2.1.1 Automatic Handling by LightGBM

One of the strengths of LightGBM lies in its ability to automatically handle missing values during the training process. This feature eliminates the need for manual imputation steps, streamlining our workflow and preserving the integrity of our data. By using LightGBM, we can avoid additional data preprocessing steps, thereby increasing efficiency and reducing the possibility of incorrectly manipulating the data.

2.2.1.2 Comparison with Traditional Models

In contrast, many models require explicit handling of missing values through techniques like mean or median imputation. These methods involve replacing missing entries with statistical averages, such as the mean or median of the respective feature. This additional preprocessing step can introduce biases and affect the overall model performance if not handled correctly. For our project, we chose to use median over mean, to better preserve the data integrity of the dataset.

2.2.2 Handling Unrealistic Values in Data

2.2.2.1 Identification of Unrealistic Values

During our data exploration, we encountered instances where reported values, such as 365,243 in employment duration, exceeded practical norms—far surpassing typical human lifespan. Recognizing these values as implausible, we conducted thorough investigations to determine their origins and hypothesized that they might have been used to denote missing data.

2.2.2.2 Treatment of Unrealistic Values

To maintain data integrity and ensure accurate analysis, we systematically replaced these unrealistic values with NaN. This approach aligns with best practices in data cleaning and helps mitigate potential biases that could arise from erroneous or anomalous entries. By treating these values as missing data, we ensure that our dataset remains reliable and suitable for robust model training and accurate predictions.

2.3 Feature Engineering

2.3.1 Feature Engineering-Primary Training Set

In our data preprocessing phase, we created several new features aimed at enhancing the predictive power of our model. These features were derived from the existing dataset and are designed to capture various financial aspects of the applicants.

2.3.1.1 New Features

1. **CREDIT_INCOME_PERCENT**: This feature is calculated as the loan amount divided by the income. Higher ratios may indicate a higher risk of default, as they suggest a larger financial burden relative to the applicant's income.

2. **ANNUITY_INCOME_PERCENT**: This is the annual repayment divided by the income. Higher ratios suggest a greater financial burden, potentially leading to an increased risk of default.

3. **CREDIT_TERM**: This feature is computed as the annual repayment divided by the loan amount. Shorter credit terms may increase the risk of default due to higher monthly payments.

4. **DAYS_EMPLOYED_PERCENT**: This is the number of employment days divided by the age of the applicant. It provides an indication of the employment stability relative to the applicant's age.

5. I**NCOME_PER_CHILD**: This feature is calculated as the income divided by the number of children. Larger families may experience higher financial strain, impacting their ability to repay loans.

6. **HAS_HOUSE_INFORMATION**: This is a binary feature indicating whether housing information is missing. Missing housing data might correlate with a higher risk of default.

2.3.1.2 Distribution Analysis

We conducted a distribution analysis to examine how these new features are distributed among defaulters and non-defaulters. Most features showed minimal differentiation between the two groups, except for CREDIT_TERM, which exhibited more noticeable differences. Although initial visual separation might be low, the effectiveness of these features can be better evaluated within the model.

2.3.2 Feature Engineering-Auxiliary Training Set

In addition to the primary training and prediction datasets, auxiliary datasets are often used to enrich the information available for model training. These datasets can be linked to the main dataset using the common key "SK_ID_CURR."

2.3.2.1 Data Aggregation

Each row in the credit bureau dataset represents loan applications submitted by individuals across various financial institutions. Given that a single "SK_ID_CURR" can be associated with multiple "SK_ID_BUREAU" entries, indicative of numerous loans per applicant, direct merging is impracticable. Instead, we employ statistical methods such as 'group by' to aggregate auxiliary data, thereby preventing record duplication and ensuring each applicant is represented uniquely within the training set.

2.3.2.2 Feature Generation

We meticulously generate features from these auxiliary datasets:

Continuous Variables: For continuous variables, we compute aggregated statistics such as the mean, maximum, and median loan amounts. These aggregated values provide a summary of the applicant's financial interactions with various institutions.

Categorical Variables: For categorical variables, we transform them into dummy variables and calculate their occurrence frequencies. This process provides insights into the applicant's financial behaviors and interactions with different types of loans and institutions.

2.3.2.3 Reducing Complexity and Redundancy

Adding numerous features can lead to increased complexity and potential redundancy in the dataset. To address this, we conduct a thorough correlation analysis, setting a stringent threshold of 0.8. This allows us to remove highly correlated features, reducing dimensionality and focusing on the most impactful variables. Finally, we get the following feature importance chart:

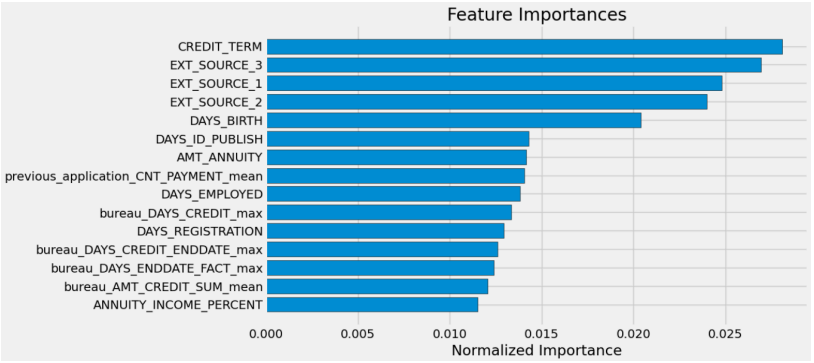

Figure2: Feature Importance After Feature Screening

3 Model

3.1 First Selection

In this project, the initial model selection encompassed Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), XGBoost, TabNet Classifier, and Light Gradient Boosting Machine (LGBM), chosen for their distinct capabilities in addressing various aspects of data analysis pertinent to this financial dataset.

Logistic Regression and KNN were selected for their robustness in binary classification tasks, crucial for determining loan approval outcomes. Logistic Regression is particularly esteemed for its interpretability—a key requirement in financial applications where decision transparency is imperative. Conversely, KNN is appreciated for its effectiveness in capturing non-linear relationships via its instance-based learning paradigm.

SVM was included for its proficiency in high-dimensional spaces, typical of financial datasets with multiple simultaneous feature analyses. The kernel trick enables SVM to manage complex, non-linear decision boundaries efficiently.

Furthermore, XGBoost and LGBM are noted for their exemplary performance in extensive datasets, a frequent attribute in the financial sector. These models incorporate advanced functionalities such as native handling of missing data and robust regularization techniques to avert overfitting, thereby ensuring model reliability and accuracy.

The TabNet Classifier, distinguished by its novel architecture that utilizes decision layers and attention mechanisms, offers a progressive method for feature selection in tabular data, enhancing the model's ability to identify pivotal predictors of loan default.

The selection process excluded Linear Regression, ResNet, and a bespoke Convolutional Neural Network (CNN) due to their relative ineffectiveness in preliminary evaluations or the complexities and challenges associated with their practical implementation in a financial context.

The assessment of these models was rigorously executed using the ROC-AUC score, a metric that quantifies the ability to differentiate between classes—vital for evaluating the risk of loan default. This methodical approach to model evaluation ensures that the selected models not only fulfill the performance criteria but also conform to the operational requirements and objectives of the financial institution.

3.2 Experimental Analysis of Model Selection

3.2.1 Experimental Settings

The models evaluated in this study included a mix of traditional machine learning algorithms and more advanced ensemble methods. The selected models for our final evaluation were Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), XGBoost, TabNet Classifier, and Light Gradient Boosting Machine (LGBM). These models were chosen for their robustness in handling large, complex datasets typically found in financial applications and their performance measured by the ROC-AUC score, which evaluates class differentiation.

3.2.2 Training and Evaluation Framework

The models were trained using a dataset that was divided into training and validation sets, with each model configured with specific parameters to optimize performance for binary classification tasks related to financial loan approvals. The primary metric for evaluation was the ROC-AUC score, assessing each model's ability to distinguish between classes.

3.3 Parameters

In this section, we delve into the specifics of parameter selection for each model included in our analysis, aiming to enhance their performance for the task of classifying financial loan approvals. The selection of parameters is pivotal as it directly influences a model's ability to learn from the training data without overfitting, thereby improving its generalization to unseen data.

3.2.1 Logistic Regression

C (Regularization Strength): Values varied from 0.01 to 10 to control the trade-off between achieving lower training errors and maintaining a low generalization error.

Penalty: Employed L1 and L2 regularization to prevent overfitting by penalizing large coefficients.

Solver: Utilized various optimization algorithms including 'liblinear', 'lbfgs', 'saga', 'newton-cg', 'sag' to find optimal coefficients.

Max Iterations: Set from 100 to 10,000 to ensure convergence of the logistic model to the optimal coefficients. However, it was realized that anything less won’t converge, so manually set to 10,000.

Class Weight: Adjusted to balance the dataset, particularly useful in scenarios with imbalanced class distributions.

3.2.2 KNN (K-Nearest Neighbors)

Number of Neighbors (`n_neighbors`): Evaluated settings like 3, 5, 7, and 10 to determine the number of nearest neighbors that influence the prediction.

Weights: Configured to either 'uniform' (where all neighbors have equal influence) or 'distance' (where closer neighbors have more influence).

Metric: Used distance metrics such as 'euclidean', 'manhattan', 'minkowski' to measure the proximity between instances.

3.2.3 SVM (Support Vector Machine)

C (Regularization Parameter): Tested levels from 0.1 to 100 to balance the classification margin and misclassification error.

Kernel: Experimented with 'linear', 'poly', 'rbf', 'sigmoid' to transform data into a suitable format.

Degree: This parameter is critical when using 'poly' kernel to determine the flexibility of the decision boundary.

Gamma: Adjusted for 'rbf', 'poly', and 'sigmoid' kernels to define how far the influence of a single training example reaches.

3.2.4 XGBoost

N Estimators: Set at 1000 to specify the number of boosting rounds or trees to build.

Learning Rate: Tuned at 0.05 to control how quickly the model adapts to the problem.

Max Depth: Configured at 6 to control the depth of each tree, affecting the model complexity and likelihood of overfitting.

Subsample and Colsample_bytree: Both set to 0.8 to specify the fraction of samples and features to use for building each tree, which helps in preventing overfitting.

Regularization (Alpha and Lambda): Used to add penalty terms to the loss function to smooth the final learned weights and avoid overfitting.

3.2.5 TabNetClassifier

Decision Prediction Layer Dimension (n_d) and Attention Embedding Dimension (n_a): Tuned to 16 and 8, respectively, to control feature transformation and selection dynamically.

Number of Steps (n_steps): Set at 3 to define the complexity of the sequential decision-making process in the model.

Gamma and Lambda_sparse: These parameters manage the entropy of outputs and encourage feature selection sparsity, enhancing model interpretability and efficiency.

3.2.6 LGBM (Light Gradient Boosting Machine)

Number of Estimators: 1000 rounds to optimize the number of trees in the boosting process.

Learning Rate: At 0.05 to ensure gradual learning using each tree added.

Max Depth of Trees: Not explicitly mentioned but typically tuned to manage model complexity.

Regularization: Applied via `reg_alpha` and `reg_lambda` to control overfitting.

Subsampling: Employed to make the training process more robust against noise.

Each model's parameters were carefully selected based on their theoretical implications and practical outcomes observed in preliminary tests. To ensure the effectiveness of these parameters, we employed a

random search methodology, which systematically explores a range of values for each parameter and selects those that optimize performance. This meticulous approach ensures each model operates within an optimal parameter space to maximize its predictive accuracy and generalization capability. The random search process not only provides a robust statistical foundation for our parameter choices but also enhances the likelihood of discovering high-performing configurations that might be missed through more traditional, manual tuning methods.

3.4 Final Selection

Upon thorough evaluation, as depicted in the Model Comparison Chart, the Light Gradient Boosting Machine (LGBM) has been selected as our final model. This decision is supported by a comprehensive analysis of the model's performance metrics, its adaptability, and efficiency in handling our dataset's unique characteristics.

The LGBM model demonstrated a robust balance in performance, achieving a Train AUC of 0.8079 and a Val AUC of 0.7659. This performance indicates effective learning from the training data and admirable generalization to unseen data in the validation set. In comparison, models like SVM, despite their high training scores, showed significantly lower validation scores, highlighting a potential overfitting issue. LGBM, however, maintained a more consistent performance across training and validation phases, suggesting a lower propensity for overfitting and a stable performance across various data subsets.

The selection of LGBM was also influenced by its compatibility with our dataset, which comprises a mix of categorical and continuous variables typical in financial datasets. LGBM's capability to directly handle categorical features without extensive preprocessing significantly enhances its applicability in financial contexts, where data can vary widely in format and complexity. Moreover, its proficiency with large datasets and its ability to efficiently manage unbalanced data underline its utility in scenarios where data volume and integrity are paramount.

Notably, LGBM is recognized for its computational efficiency and speed in processing large datasets—crucial attributes in real-world applications requiring timely and accurate responses. The model employs Gradient-based One-Side Sampling (GOSS), which prioritizes more challenging and informative cases during training, thus reducing noise and enhancing accuracy. Additionally, the Exclusive Feature Bundling (EFB) technique reduces the feature space without compromising performance, facilitating faster computations.

In conclusion, the selection of LGBM as the optimal model for predicting financial loan approvals was driven by its strong performance metrics, suitability for the data, and computational advantages. As we move towards deployment in a production environment, LGBM is poised to enable rapid and precise decision-making crucial for loan processing. Ongoing monitoring and adaptive tuning of the model's parameters will be essential to maintain its efficacy and adaptability to evolving dataset characteristics, ensuring sustained high levels of predictive accuracy.

4 Results Analysis

The evaluation of model performance using the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) provides a robust metric for assessing the ability of various models to discriminate between classes, such as distinguishing likely loan defaulters from non-defaulters in this case.

Logistic Regression recorded a Train AUC score of 0.7676, which slightly declined to a Validation AUC of 0.7642. This small reduction suggests that Logistic Regression maintains a consistent performance between the training and validation datasets, indicating good generalizability and minimal overfitting.

KNN recorded a perfect Train AUC score of 1.0, which significantly declined to a Test AUC of 0.5626. This substantial reduction implies that the KNN model, while perfect on training data, is likely overfitting by capturing noise or non-generalizable patterns, which adversely affects its performance on unseen data.

SVM recorded a Train AUC score of 0.5891, which slightly increased to a Validation AUC of 0.6122. This indicates that the SVM model performs moderately on the training data but improves slightly on the validation data, suggesting that it may generalize better without significant overfitting.

XGBoost recorded a Train AUC score of 0.9109, which declined to a Validation AUC of 0.7563. This reduction suggests that while XGBoost performs well on the training data, it may still be overfitting to some extent, as evidenced by the decrease in performance when applied to validation data.

TabNetClassifier recorded a Train AUC score of 0.7865, which slightly declined to a Validation AUC of 0.7587. This small reduction indicates that TabNetClassifier maintains a consistent performance between the training and validation datasets, showing good generalizability and minimal overfitting.

LGBM recorded a Train AUC score of 0.8456, which declined to a Validation AUC of 0.7773. This reduction suggests that while LGBM performs well on the training data, it may still be overfitting to some degree, as indicated by the decrease in performance on the validation data.

Model	**Train AUC**	**Validation AUC**
Logistic Regression	0.7676	0.7642
KNN	1.0000	0.5626
SVM	0.5891	0.6122
XGBoost	0.9109	0.7563
TabNetClassifier	0.7865	0.7587
LGBM	0.8456	0.7773

Given these evaluations, LGBM emerges as the optimal model. It achieves a superior balance in generalizing from training to validation data and is proficient in handling both categorical and continuous features. Its capabilities in managing large and potentially unbalanced datasets, combined with its computational efficiency, render it particularly suitable for financial applications where accuracy and processing speed are paramount.

For models like XGBoost, despite their impressive training performance, significant tuning is necessary to curtail overfitting and enhance their generalizability. Techniques such as parameter regularization, pruning, or more robust validation methods like cross-validation could be instrumental in this regard.

In conclusion, the choice of LGBM for deploying financial loan approval predictions is strongly supported by its performance metrics and adaptability to data dynamics. However, the need for continuous monitoring and adaptive tuning of the model's parameters is highlighted to maintain and potentially improve its predictive performance in response to evolving data characteristics.

5 Conclusion

This project aimed to enhance the accuracy of risk assessments for banks and lenders regarding potential loan defaults. By improving these evaluations, our goal was to promote responsible lending practices, allowing financial institutions to extend credit to those in need while minimizing the risk of defaults. We explored multiple machine learning models and preprocessing techniques, ultimately selecting the Light Gradient Boosting Machine (LGBM) for its efficiency and adaptability.

Our preprocessing steps addressed missing and unrealistic values to maintain data integrity, and feature engineering efforts added meaningful features to enhance predictive power. LGBM's ability to handle large datasets and various feature types without extensive preprocessing made it particularly suitable for our needs.

In conclusion, this project's successful application of LGBM highlights the importance of robust preprocessing and appropriate model selection in financial risk assessment. By improving default risk

predictions, financial institutions can foster responsible lending practices, contributing to a stable and inclusive financial ecosystem.

6 Future Considerations and Next Steps

6.1 Data

In future considerations, we plan to integrate additional data sources such as social media activity, mobile phone usage, and utility payments to gain a comprehensive view of borrower behaviors. Using public financial datasets and alternative credit scoring data will capture a broader range of borrower characteristics. Regularly updating datasets is crucial for relevance.

Creating features that capture spending habits, payment punctuality, and macroeconomic indicators like inflation and unemployment rates will provide deeper insights. Interaction features can reveal complex relationships in the data.

Advanced feature engineering techniques, such as polynomial features, binning, PCA, and ICA, will enhance pattern recognition. Using domain-specific knowledge to create features and testing methods like target encoding for categorical variables will further optimize the model.

Feature selection through techniques like Recursive Feature Elimination (RFE) and Lasso regularization will identify impactful features, while correlation analysis will reduce redundancy. Evaluating feature importance with Random Forests and SHAP values will prioritize critical features.

Focusing on these improvements will boost prediction accuracy, enhance loan approval rates, reduce default rates, and foster greater financial inclusion.

6.2 Model Improvement

To augment the predictive capacity, robustness, and transparency of our loan risk assessment model, we propose an integrated strategy. This strategy involves the experimentation with models such as CatBoost, which is recognized for its efficacy in handling categorical data and addressing imbalances, and the adoption of pre-trained models for transfer learning to enhance performance while reducing the reliance on extensive training datasets.

We plan to implement model ensembling techniques, integrating predictions from models like XGBoost and CatBoost, and employing advanced methodologies such as stacking to leverage the individual strengths of each model. Furthermore, we intend to explore the application of boosting algorithms, specifically AdaBoost, for targeted scenarios and to calibrate probability estimates to yield accurate predictions of default likelihood.

For enhanced interpretability, we will utilize SHAP (SHapley Additive exPlanations) values to delineate feature contributions and LIME (Local Interpretable Model-agnostic Explanations) for elucidating individual prediction rationales. This approach aims to foster greater transparency and trust among stakeholders.

These enhancements are designed to refine our model into a state-of-the-art tool that equips lenders with the insights necessary to make informed decisions. By doing so, we aspire to expand financial inclusion and contribute to a safer, more equitable financial ecosystem, thus reinforcing the connection between advanced analytical techniques and practical financial applications.

References

Sermeño, R. (2023, September 1). Banks lost billions on bad loans last quarter: Kiplinger economic forecasts. Kiplinger. https://www.kiplinger.com/personal-finance/banking/banks-lost-billions-on-bad-loans-kiplinger-economic-forecasts

Aziz, R. M., Baluch, M. F., Patel, S., & Ganie, A. H. (2022). LGBM: a machine learning approach for Ethereum fraud detection. International Journal of Information Technology, 14(7), 3321-3331.

Abou Omar, K. B. (2018). XGBoost and LGBM for Porto Seguro's Kaggle challenge: A comparison. Preprint Semester Project.