Classifying Loan Applications and Optimizing Interest Rates
Final Report

Valentin Pinon
Jan Philipp Girgott

Massachusetts Institute of Technology
Sloan School of Management

Machine Learning Under a Modern Optimization Lens (15.093)

December 13th, 2023

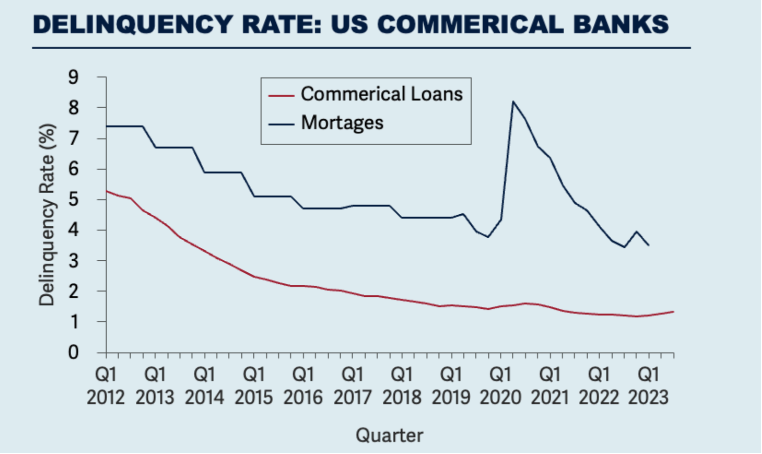
MIT MANAGEMENT SLOAN SCHOOL

Table of Contents

1 Problem Description

Loan defaults, an increasingly critical issue in financial systems, have far-reaching implications that encompass Credit Damage, Lender Losses, and Economic Turbulence. Firstly, lenders bear a significant brunt of these defaults, as they not only lose out on the principal and interest of the defaulted loan, but such occurrences also erode their profitability and stability. This situation often compels lenders to hike interest rates to offset losses, which adversely affects new borrowers, escalating the cost of borrowing across the board. Secondly, the borrowers themselves face dire consequences. A defaulted loan severely dents their credit scores, limiting their ability to secure future loans, and even impinging on their chances of obtaining housing or employment, as many sectors consider credit history in their evaluation processes.

Furthermore, the ripple effects of loan defaults extend into the wider economy. Reduced lender confidence results in tightened lending criteria and decreased access to capital, which can stifle business expansion and innovation. This, in turn, leads to slower economic growth, as businesses and individuals struggle to access the funds needed for investments and consumption. In extreme cases, high rates of loan defaults can precipitate financial crises, akin to the subprime mortgage crisis of 2007-2008, where massive loan defaults triggered a global economic downturn. This multifaceted impact of loan defaults underscores their potential to not only disrupt individual lives and businesses but also to destabilize entire economies, making it a critical issue for policymakers, financial institutions, and consumers alike to address.



The accompanying graph provides a compelling visualization of delinquency rates for commercial loans and mortgages at U.S. commercial banks from the first quarter of 2012 through to the first quarter of 2023. Over the period, commercial loans have experienced some fluctuations, with delinquency rates notably peaking before a sharp decline. In contrast, mortgage delinquencies have maintained a steady, gradual decline, maintaining a rate well below that of commercial loans throughout the period. Despite the observed trends, it remains critical to further reduce delinquency rates for both commercial loans and mortgages. Lower delinquency rates are indicative of a healthier economy where businesses and individuals are more capable of meeting their financial obligations. This financial stability is essential not only for the resilience of the banking sector but also for fostering a positive lending climate, which can stimulate investment and consumption. The machine learning approach developed in this report can help achieve this objective.

2 Data

2.1 Kaggle Dataset

The data obtained from Kaggle contains two datasets, one on current applications and one on previous applications. As part of the data preprocessing, we merged the two datasets to have an even wider range of loan applications to train on. The full merging process is described in the next section. Please find below a summary of the features of the current applications dataset which contains $p = 122$ features for $n = 307511$ loan applications.

Basic Applicant Information: This includes the unique ID of the loan (SK_ID_CURR), the target variable indicating payment difficulties (TARGET), contract type (NAME_CONTRACT_TYPE), gender (CODE_GENDER), car ownership (FLAG_OWN_CAR), real estate ownership (FLAG_OWN_REALTY), number of children (CNT_CHILDREN), and total income (AMT_INCOME_TOTAL).

Loan Details: This covers the loan amount (AMT_CREDIT), annuity (AMT_ANNUITY), and the price of goods for consumer loans (AMT_GOODS_PRICE).

Applicant's Background: Detailed information about the client's background includes their company type (ORGANIZATION_TYPE), occupation type (OCCUPATION_TYPE), education level (NAME_EDUCATION_TYPE), family status (NAME_FAMILY_STATUS), housing situation (NAME_HOUSING_TYPE), and age in days at the time of application (DAYS_BIRTH).

Contact Information: This encompasses various flags indicating whether the client provided a mobile phone (FLAG_MOBIL), work phone (FLAG_EMP_PHONE), home phone (FLAG_WORK_PHONE), and if these were reachable (FLAG_CONT_MOBILE), along with email availability (FLAG_EMAIL).

Employment and Financial History: Data related to days employed before the application (DAYS_EMPLOYED), registration changes (DAYS_REGISTRATION), and the last change in identity document (DAYS_ID_PUBLISH). It also includes data on past enquiries to Credit Bureaus at various time frames before the application (AMT_REQ_CREDIT_BUREAU_HOUR, DAY, WEEK, etc.).

Residential Information: This includes normalized data about the client's residence, such as the average, modus, and median size of the apartment, common area, living area, age of building, number of elevators, entrances, state of the building, and floor number (APARTMENTS_AVG, BASEMENTAREA_AVG, YEARS_BEGINEXPLUATATION_AVG, etc.).

Social and Environmental Factors: This covers the client's social circle observations (OBS_30_CNT_SOCIAL_CIRCLE), defaults in social circle (DEF_30_CNT_SOCIAL_CIRCLE), region population relative to the client (REGION_POPULATION_RELATIVE), and our rating of the region where the client lives (REGION_RATING_CLIENT, REGION_RATING_CLIENT_W_CITY).

Document Verification: Information about whether the client provided various documents (FLAG_DOCUMENT_2, FLAG_DOCUMENT_3, etc.).

Application Process Details: This includes the weekday and hour of the loan application (WEEKDAY_APPR_PROCESS_START, HOUR_APPR_PROCESS_START), and various flags indicating mismatches between the client's permanent, contact, and work addresses.

2.2 Data Preprocessing

We first preprocessed the applications dataset. We started by looking up all the features which had 40% or more missing values. These are features for which we cannot impute any values as there are too many missing values. These features were in majority features related to the apartment of the applicant and its area of living. These features were thus removed from the dataset as we could not deal with them.

We then looked at the remaining features with missing values and classified them in different groups. First, there was the features related to a timeline of the application such as how many queries about the client were there in the 7 days prior to the application. As we do not have the same information in the previous application dataset and we want to merge both datasets, we had to get rid of those. Then, there were the features for which a missing value would actually give us information. For instance, in the feature OWN_CAR_AGE, which describes the number of days a client has owned a car for, if there was a missing value it meant that the client did not own a car. The same goes for EMPLOYMENT_TYPE: a missing value meant the client was unemployed. We therefore filled those with a 'missing' string if the feature type was a string, or with a zero and created a "Flag" column with 1 were the missing value was, to indicate to the model that it is not a zero like the others. Finally, we only had a few features with less than 0.1% of missing values and decided to remove the rows concerned. We then transformed all categorical features into dummy variables and the time features remaining from "days since..." to "year starting": instead of having the number of days a client has owned a car we put the year it bought it, which would allow us to use this feature in the previous application dataset.

The data in the application dataset can be categorized into 2 types: the features regarding the client (job, family status,...) and the features regarding the application itself. In the previous application dataset, we only had features regarding the applications. However, the clients in the previous application dataset are the same as in the application dataset. We therefore used the personal features from the application datasets and added them to the previous application dataset.

Finally, we now needed to create our training and test sets. We needed 4 of them: one to train the prediction model, one to validate it, one to actually predict the default probabilities on and then train the optimization model on and finally one to test our optimization model on. The 2 first datasets need to be from the application dataset as they need the default as target feature. The 2 other datasets need to be from the previous application dataset so we can now which applications have been approved and which rejected. Therefore, we divided all the applicants ID in 4 groups, 2 of them with 40% each and 2 others with 10% each: the first were training sets, the second test sets. We then took the first 40% of the IDs from the application dataset to make our first training set, the 10%

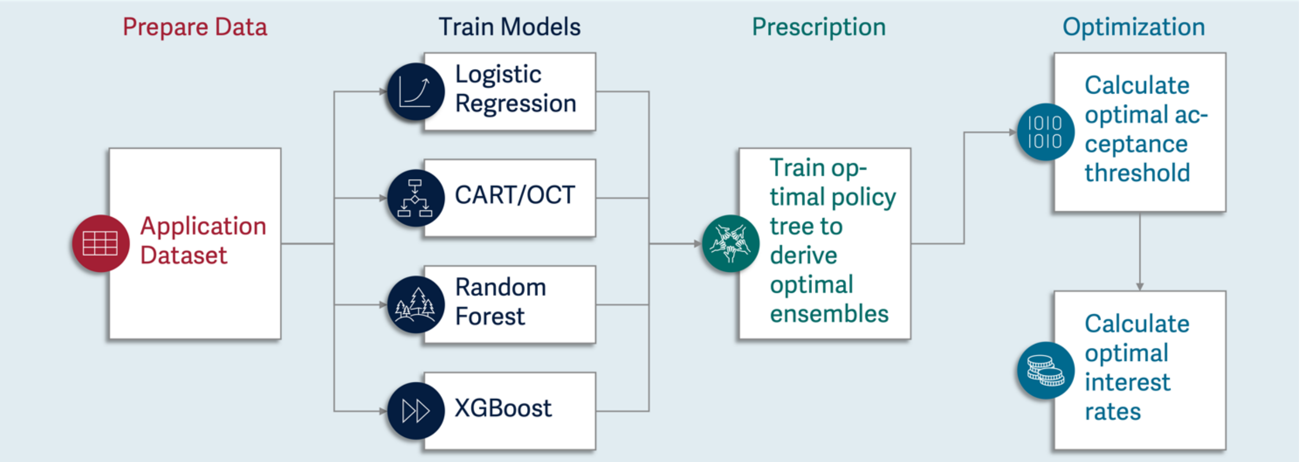
following to make our first test set, then the second 40% of the IDs from the previous application dataset for our second training set and finally the last 10% from the previous application to make our second test set. That way, there is no data leakage due to the "personal features" being the same in both datasets.

To make up for the class imbalance in our datasets, we then used a SMOTE method to resample our minority class. This allowed us to have more training data for our models and be more accurate.

3 Methodology

3.1 Overview

Our overall methodology is summarized in the graphic below. The approach is a fusion of classification techniques, prescriptive analytics, and optimization procedures.


Prepare Data
Train Models
Prescription
Optimization
Application Dataset
Logistic Regression
CART/OCT
Random Forest
XGBoost
Train optimal policy tree to derive optimal ensembles
Calculate optimal acceptance threshold
Calculate optimal interest rates

After data preparation, a suite of predictive models is trained to assess loan applications. This suite includes robust Logistic Regression, decision tree algorithms like CART/OCT, ensemble methods such as Random Forest, and the gradient boosting technique XGBoost. The methodology advances to a 'Prescription' phase, where an Optimal Policy Tree (OPT) is trained. The purpose of this step is to distill from the predictive models an ensemble that yields the best prescription for decision-making, effectively combining the predictive outputs into actionable strategies.

In the final stage, 'Optimization', the focus shifts to the application of analytical methods to fine-tune decision thresholds and financial terms. This involves calculating the most advantageous acceptance threshold for loan approvals and determining the optimal interest rates to offer. These calculations are likely aimed at balancing risk and profitability, ensuring competitive loan products while maintaining sound fiscal management.

3.2 Models Applied: Robust Logistic Regression, CART, OCTs, Random Forest, XGBoost

To predict the probability of default for each loan application, we applied different models. The first model that came to mind as we are predicting probabilities was logistic regression. However, as we have data from different datasets that we are merging, this data is probably not 100% correct. That is the reason we decided to implement robust logistic regression.

To add to this, we then tried different tree methods, ranging from OCTs, to XGBoost and Random Forest. All these models were fine-tuned using a GridSearch cross-validation method. However, for XGBoost, as the dataset we have has a lot of entries, the cross-validation computation time was excessive (several hours). Thus, we chose to instead implement a random GridSearch, which does not try all combination for the whole range of parameters we gave it but random combinations instead. We limited the search to 500 combinations for a 4-folds cross-validation.

Finally, we also wanted to be able to compare the benefits of optimization on this matter, which is why we trained a CART model as well.

3.3 Prescription Tree

We used an OPT to act as a meta-model that intelligently combines the predictions from the various models. In particular, we considered the following policies: (1) 100% of the weight on one model, (2) combination of two models with 50% each, (3) combination of three models with 33% each, (4) combination of four models with 25% each, (5) combination of all models with equal weights (20%). This diverse set of ensembles allows the optimal policy tree to evaluate numerous policies to ascertain which ensemble best predicts loan defaults, leading to the most informed and strategic decision-making policy.

The outcomes are derived using a logloss function. The log loss is calculated by comparing the true labels with the predicted probabilities, providing a measure of the cost associated with the prediction accuracy of the models. Eventually, hyperparameter tuning is accomplished through a grid search strategy, utilizing the GridSearch functionality from the Interpretable AI package. This method systematically explores a range of hyperparameters, including the minimum number of samples in the leaf node (minbucket) and the depth of the tree (max_depth). It uses a cross-validation approach with 10 folds (fit_cv!) to ensure that the model's performance is validated across different subsets of the data.

3.4 Optimization Model for Interest Rate Assignment and Loan Acceptance

The mathematical model (for full formulation see Appendix 1) provides an approach to optimizing the bank's process of evaluating and accepting loan applications. It balances the dual objectives of minimizing the risk of default and maximizing profitability through loan acceptances and interest rate assignments.

At the heart of this model are several key variables. The binary decision variable x indicates whether a particular loan is accepted by the bank. This is complemented by c, which denotes the probability that a client will accept the offer associated with their loan. The interest rate for each loan is represented by r. Lastly, the model introduces a threshold variable t, which sets the minimum acceptance standard based on the risk assessment. The model also takes into account essential input data like the amount requested for each loan a and the associated probability of default p. The objective function aims to maximize the expected profit from the loan portfolio. This is calculated by summing up the products of the loan amount, interest rate, and adjusted risk factor across all loans,

The first two constraints ensure that only loans with a probability of default below the acceptance threshold are accepted. Constraints three, four, and five linearize the product of the variables x and c which is a product of a binary variable and a continuous variable. To do so, the model introduces an auxiliary variable z. Constraint six stipulates a linear relationship between the client acceptance probability and the interest rate assigned (blue line in graphic in Appendix 2). An even more reasonable approach would be to assume a quadratic relationship (red line in graphic in Appendix 2), but this would make the model not tractable since it would become a non-convex optimization problem with convex constraints. The remaining constraints model x as a binary variable and ensure that c and t are values between 0 and 1. Furthermore, they set a minimum and maximum interest rate and mandate that the auxiliary variable z is non-negative.

3.5 Prediction Results

The models have been assessed based on three key metrics: Area Under the Receiver Operating Characteristic Curve (AUC), Sensitivity (True Positive Rate, TPR), and Specificity (True Negative Rate, TNR). The results are as follows:

- The Classification and Regression Tree (CART) model has an AUC of 0.918, 85.1% sensitivity, and 98.5% specificity.
- The Optimal Classification Tree (OCT) has a slightly lower AUC of 0.909, with a sensitivity of 78.1% and a higher specificity of 99.8%.
- The Random Forest model shows an AUC of 0.93, sensitivity of 82.8%, and the lowest specificity of 87.5% among the models.
- XGBoost outperforms the other models with the highest AUC of 0.959, sensitivity at 90.3%, and nearly perfect specificity at 99.9%.
- Logistic Regression also shows robust performance with an AUC of 0.95, 90% sensitivity, and a perfect specificity score of 100%.

This emphasizes that the XGBoost model achieves the highest AUC and an almost perfect specificity, indicating that it is the most effective model among those presented for distinguishing between the positive and negative classes in the dataset. Despite the great performance of the individual models, a strategic combination could lead to a more nuanced and accurate classification system, particularly with an Optimal Policy Tree. In this way, we can capitalize on the high specificity of one model, the sensitivity of another, and the superior AUC of yet another. For instance, the near-perfect specificity of the XGBoost model could be harmoniously blended with the perfect specificity of Logistic Regression, ensuring that the ensemble not only accurately identifies positive cases but also minimizes false positives. The OPT shows the best performance with an AUC of 0.964, sensitivity of 91.8% and specificity of 94.3%. This is a great case to show that ensembling can improve model performance even further.

Apart from the model performances, we also analyzed the role individual features play in classifying an application. To do this, we looked at images of the resulting trees and at importance scores. Overall, the importance scores illustrate that demographic factors (like gender and marital status), socioeconomic status (like income type, education level, and property ownership), and lifestyle indicators (like car ownership and age) are critical in predicting loan default risk according to the machine learning model's learned patterns from the data. Please find below an interpretation of the five most important features presented by the XGBoost model.

1. **CODE_GENDER_F**: Gender being female has the highest importance score of approximately 0.112, suggesting that the gender of the applicant plays a significant role in predicting loan defaults, with female applicants perhaps showing different default patterns than male applicants.
2. **NAME_INCOME_TYPE_Working**: Being in a working income type has the second-highest importance of around 0.077, indicating that the employment status of the applicant is a strong predictor, with those in regular employment potentially having a different risk profile than those who are not.
3. **FLAG_OWN_REALTY_Y**: Owning realty (i.e., property or real estate) with a score of about 0.076 implies that property ownership is closely associated with the ability to service a loan, possibly due to the financial stability or additional collateral that comes with property ownership.
4. **NAME_INCOME_TYPE_Commercial associate**: This category has an importance of around 0.071, which suggests that applicants who are commercial associates have distinctive characteristics that affect their likelihood of defaulting on a loan.
5. **NAME_EDUCATION_TYPE_Higher education**: The importance score of approximately 0.064 for this variable shows that the level of education, specifically higher education, is a considerable factor in predicting loan defaults, potentially due to the impact of education on earning potential and financial management skills.

3.6 Optimization Results

Our optimization model allows us to have great results as well as good insights on which loans to approve or reject and if needed which interest rate to prescribe. The first optimization model we used was optimizing the threshold to decide from what default probability we would reject loans, as well as the interest rates at the same time. It landed on an acceptance threshold of 25.9%: above a

25.9% probability of default the model automatically rejects the loan and give it the smallest possible interest rate as it does not matter which interest rate it gives. We compared to whether the loans were approved in our datasets, we obtain an 82.7% of loans approved that we approved and 84.2% of loans rejected that we rejected.

For the approved loans, the prescribed interest then depends on the probability of default: the higher the probability of default, the higher the interest rate. This makes intuitive sense: a more risky client will have a higher interest rate because we are not sure we want to loan them money and if we do might as well have a big return on investment, whereas a less risky client will have a lower interest rate as we want to prioritize landing their business and make sure they do not leave as they are "guaranteed" profits. The interest rates decrease linearly with the probability of default due to our definition of the interest rate acceptance by the client function. As the probability of default is directly linked to the amount asked for a loan (all things equal meaning the same client applying twice for the same type of loan but with different amounts, the loan with the higher amount will have a higher probability of default.

4 Discussion of Results

4.1 Managerial Implications and Recommendations

This model would allow banks to have a rough idea on which loans to approve and reject, and which interest rates to prescribe if approved. However, this is just a proof of concept to be used as an extra tool in a multimodal decision-making context. This model also can give insights on which habits or characteristics to look at in a client application when taking the decision to approve it or not, using the feature importance analysis that we did.

4.2 Limitations

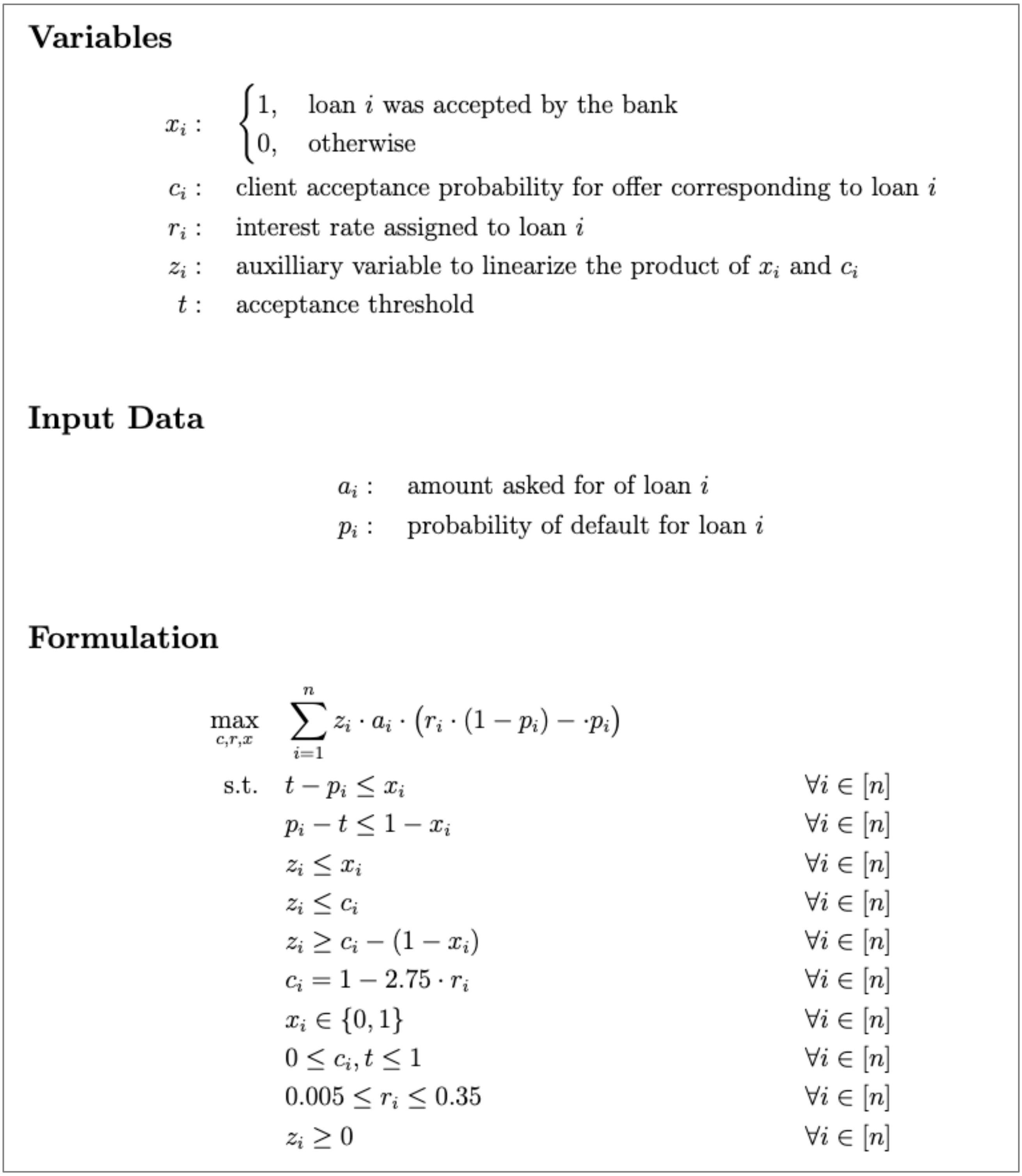
This model can however be greatly improved with more fine-tuning of our parameters and additional data. For instance, we believe that including data on money habits from checking accounts and past granted loans would give even more information to our model. We could also enrich our ensembling model by adding other prediction models such as SVMs or Neural Networks.

On the optimization side, we believe that we could improve our interest rate acceptance from the client function by making it non-linear: an exponential negative for instance or a quadratic function would probably better represent reality. However, these functions being non-linear greatly increase the computational time or even make it infeasible for Gurobi to solve. We could also incorporate different stages of defaults and fine-tune the interest rates range.

Finally, if we had a dataset with approved loans and the given interest rates that we could compare outcomes with that would help us fine-tune our model and parameters.

5 Appendix

5.1 Full Optimization Model Formulation



Variables

$$x_i : \begin{cases} 1, & \text{loan } i \text{ was accepted by the bank} \\ 0, & \text{otherwise} \end{cases}$$

c_i : client acceptance probability for offer corresponding to loan i

r_i : interest rate assigned to loan i

z_i : auxilliary variable to linearize the product of x_i and c_i

t : acceptance threshold

Input Data

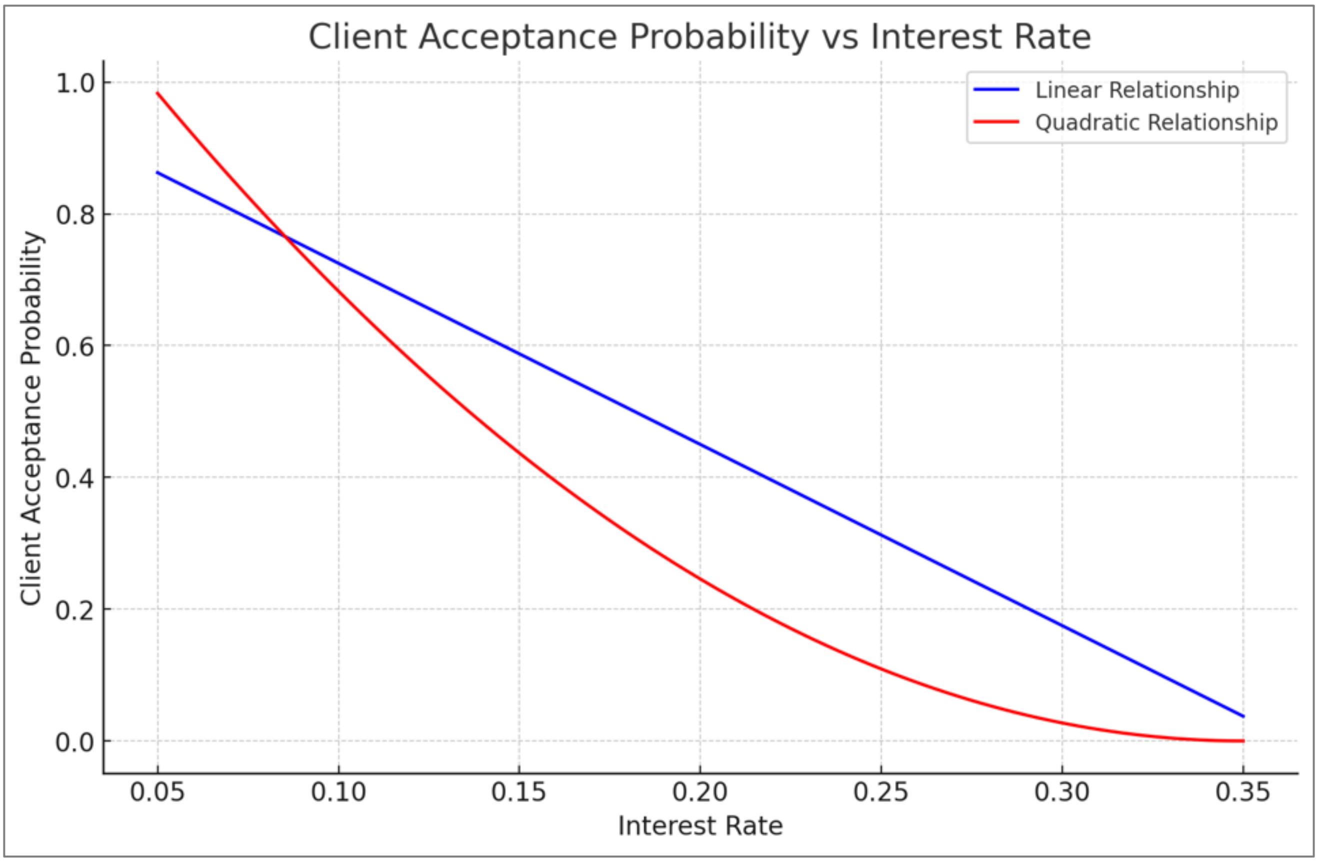
a_i : amount asked for of loan i

p_i : probability of default for loan i

Formulation

$$\begin{aligned}
\max_{c,r,x} \quad & \sum_{i=1}^{n} z_i \cdot a_i \cdot \big(r_i \cdot (1 - p_i) - \cdot p_i\big) & \\
\text{s.t.} \quad & t - p_i \leq x_i & \forall i \in [n] \\
& p_i - t \leq 1 - x_i & \forall i \in [n] \\
& z_i \leq x_i & \forall i \in [n] \\
& z_i \leq c_i & \forall i \in [n] \\
& z_i \geq c_i - (1 - x_i) & \forall i \in [n] \\
& c_i = 1 - 2.75 \cdot r_i & \forall i \in [n] \\
& x_i \in \{0, 1\} & \forall i \in [n] \\
& 0 \leq c_i, t \leq 1 & \forall i \in [n] \\
& 0.005 \leq r_i \leq 0.35 & \forall i \in [n] \\
& z_i \geq 0 & \forall i \in [n]
\end{aligned}$$

5.2 Modeling the Client Acceptance Probability



5.3 Split of Tasks

Task	Valentin Pinon	Jan Philipp Girgott
Conception of methodology	✓	✓
Data cleaning and preprocessing	✓	
Model training: OCT, Random Forest, XGBoost	✓	
Model training: Robust Logistic Regression, CART		✓
Ensembling via an OPT		✓
Optimization formulation	✓	
Optimization implementation		✓
Slides creation		✓
Report writing	✓	✓