# Comparison between non-linear and linear models to predict financial status in a low-dimensional dataset

David G. Mora Salazar, Matyáš Švehla, Marta Wolf, Hai Yen

**Abstract**

Prediction of a default among clients is one of the most common applications of machine learning models in finance. This paper applies several models. Among the linear models, logistic regression, LASSO, and Ridge regression are trained. To capture nonlinear relationships, random forest and XGBoost are applied. Our dataset consists of 10,000 observations of a target variable indicating whether a client defaulted, along with 8 client features describing their financial state. Missing values are present, and several imputation techniques are employed, including k-nearest neighbors, decision trees, logistic regression, and data generated by normal and lognormal distributions. Among the linear models, logistic regression displayed the highest AUC (0.9110), followed by LASSO (0.9108) and Ridge (0.9104). Among the nonlinear models, XGBoost (AUC = 0.916) outperforms Random Forest (AUC = 0.906). Notably, XGBoost outperforms both other nonlinear and linear models. These results confirm the effectiveness of non-linear machine learning models in banking applications, particularly when predictive accuracy is critical.

**Keywords: Variable selection algorithm, null values imputing techniques, linear models, regularization, non-linear models, learning by correcting models**

## 1. Introduction

Credit risk assessment is a crucial component of financial systems. Financial firms can better manage resources and minimize losses using risk analysis. Logistic regression has traditionally been used to predict credit risk due to its simplicity and interpretability. However, it struggles to capture complex relationships in financial data, particularly when dealing with multicollinearity, endogeneity, heteroskedasticity, and the linearity assumption. Machine learning techniques are increasingly employed as they handle non-linear interactions between variables more effectively.

S&P Global (2020) explains how machine learning algorithms utilize large datasets to identify patterns and generate meaningful insights, which can be effectively applied to credit risk modeling. Ledolter (2013) demonstrates how machine learning methods enable the analysis, exploration, and simplification of high-dimensional datasets, which is essential in modern financial analytics.

Practical financial data often contain missing values, differences in categorical variables, and an uneven distribution of the target variable. The examined dataset contains 5.3% missing values, which may impact model accuracy. Using R's extensive package ecosystem is crucial for data analysis tasks, particularly for handling missing data through methods

such as K-nearest neighbors (KNN) and tree-based approaches Toomey (2014). This study applies multiple imputation techniques, including K-means, KNN, random forests, non-linear logit, and imputation using normal and lognormal inverse functions. The objective is to clean and prepare the data for analysis and compare different models for predicting credit default.

Additionally, this research focuses on selecting the optimal set of explanatory variables by testing various combinations during cross-validation and out-of-time validation to enhance model performance.

Our research objectives are as follows.

1. To clean and adjust the dataset to resolve missing values and inconsistencies in the data using KNN.

2. To test different models for predicting credit default: Logistic Regression, LASSO, Ridge Regression, Random Forest and XGBoost.

3. To compare the performance of the models using precision, recall, Area under the curve (AUC), the Mcfadden R square deviance indicator (R2) and the Bayesian information criteria (BIC).

4. To verify the accuracy of the models on the test sample and using out-of-time validation (data from 2016–2018) as well as cross-validation.

# 2. Data Description and Preprocessing

The data set used for the research contains a standard set of variables which describe characteristics of mortgage loan applicants. There are 8 variables available for potential features (Income, Installment/Income ratio, SchufaCredit Score, Loan Amount, Occupation, Number of Applicants, Term Length, Marital Status), Observation Date and a target variable indicating whether a client defaulted on his mortgage after the given date of observation.

| Variable name | Varible name in R code | Type |
|---|---|---|
| Income | income | numeric |
| Installment/Income | install_to_inc | numeric |
| SchufaCredit Score | schufa | numeric |
| Loan Amount | loan_amount | numeric |
| Occupation | occup | categorical |
| Number of Applicants | num_applic | categorical |
| Term Length | term_length | numeric |
| Marital Status | marital | categorical |

Table 1: Overview of Feature Variables

## 2.1 Incidence and Distribution of Missing Values

The dataset contains 10000 observations and, including the target variable and an observation date, 10 columns in total. There are missing or invalid entries across 7 out of 8 features, approximately 6% values are missing in each case. In the target variable, the proportion of missing values is very similar to the proportion of missing values in the aforementioned feature variables (approximately 6%). In total, approximately 5.3% of the data points are missing across all columns. Figure 1 depicts the location of missing

values in the data set. Based on the schematic depiction, that is a constant percentage of missing values among the variables, including the target variable, we do not have a reason to assume than missing data occur in any nonrandom systematic manner and therefore, an interest to work it out is risen.

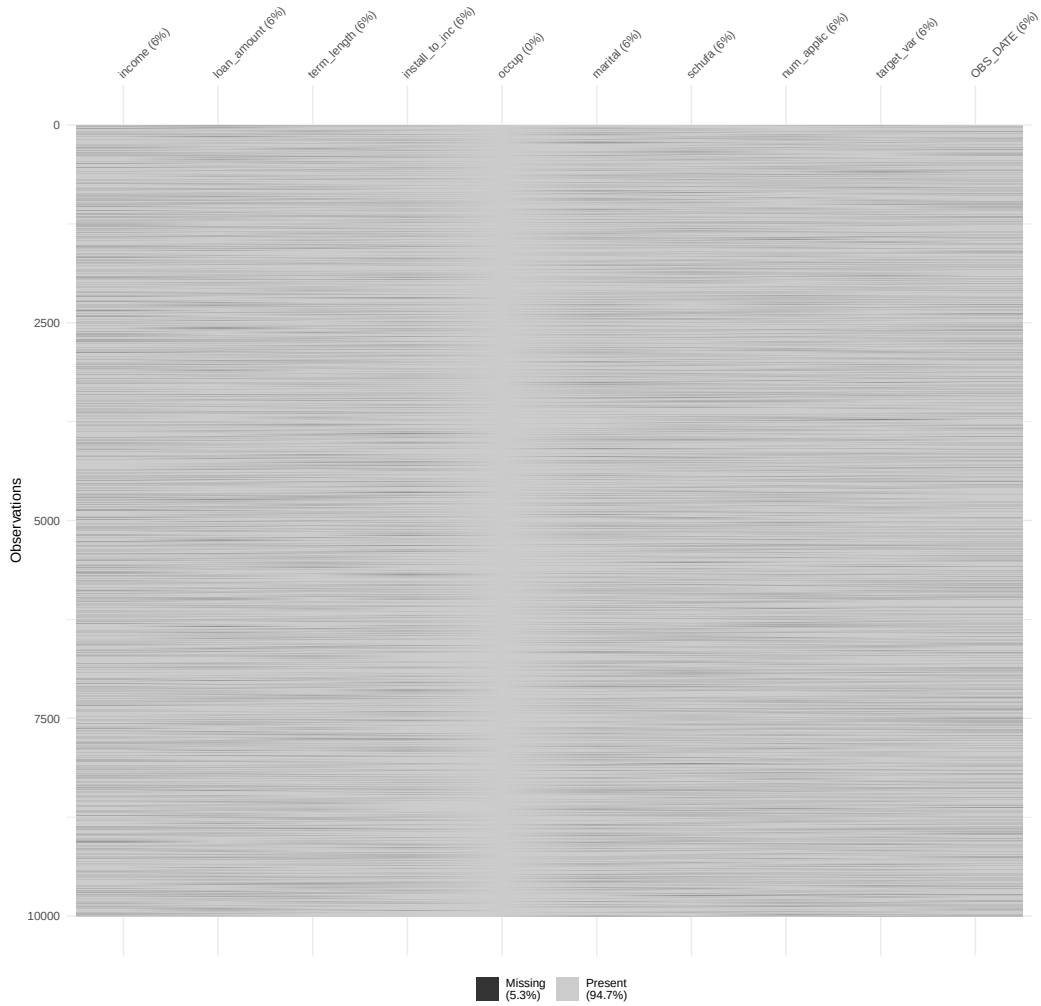



Figure 1: Visualization of location of missing values in the data set

## 2.2 Missing Value Imputation

Due to the differing nature of the features, a careful approach is needed for missing value imputation. For numerical data, we analyze the properties of non-missing values to generate random data points that align with the original distribution. Skewness is particularly important in determining whether the data follows a normal or log-normal distribution. For Income, Term Length, Installment/Income, and Schufa Credit Score, we apply a log-normal distribution, while for Loan Amount, we use a normal distribution.

For categorical data, classification methods are used to impute the missing data. In the case of Marital Status, a decision tree is employed (see Figure 3). Logistic regression is

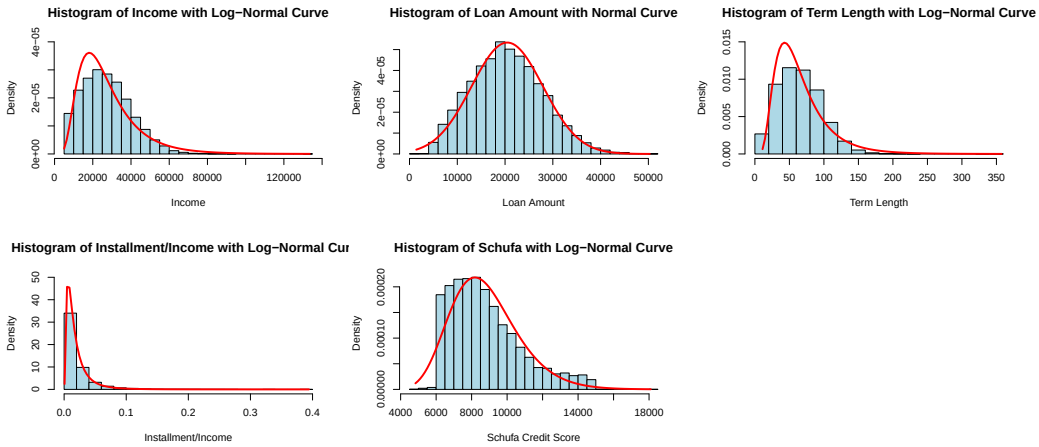



Figure 2: Histograms for numeric variables after data imputation and illustration of densities to which were used to conduct the missing data imputation

used for the imputation of missing values in the case of Number of Applicants (see Table 2).

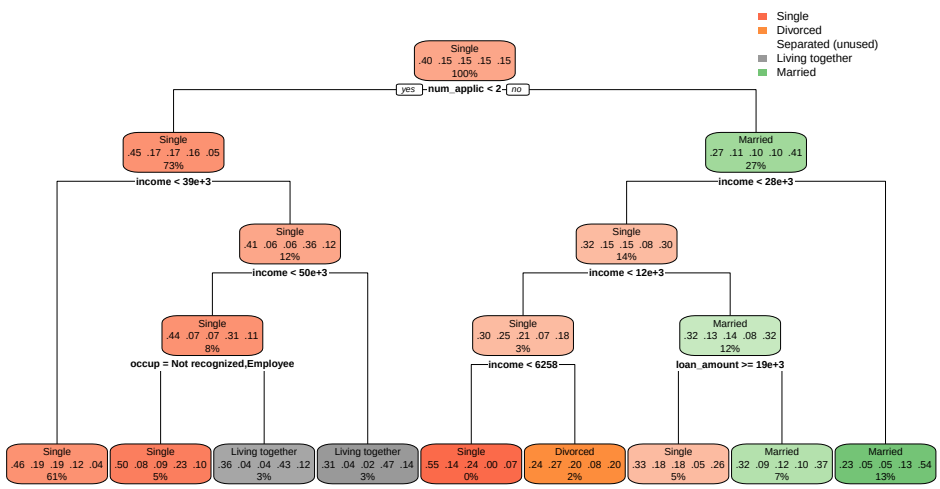



Figure 3: Visualization of the Tree model model used for imputation of missing values in Marital status

The only feature without any missing data is Occupation. It needs to be noted that, while not containing missing values explicitly, one of the levels of Occupation variable is "Not recognized". This level accounts for 8.76% cases, slightly higher than for missing values in other variables. In this case, however, we do not attempt to conduct missing value imputation. Instead, we treat these values as a new level, which may contain useful information about the subjects by itself and may help to predict since it can be an important category.

Basic characteristics of numerical variables after the imputation of missing values may be found in the table of descriptive statistics (see Table 3) and in the correlation matrix (see Table 4). Besides providing an intuition about the scale and interrelations between the variables, it is vital to ensure that assumption of no perfect multicollinearity is not

Table 2: Logistic Regression model used for missing values imputation in Number of Applicants variable

| | *Dependent variable:* |
|---|---|
| | num_applic |
| income | −0.00000 |
| | (0.00000) |
| loan_amount | −0.00000 |
| | (0.00000) |
| occupStudent | 0.191* |
| | (0.109) |
| occupWorker | 0.073 |
| | (0.098) |
| occupEmployee | −0.015 |
| | (0.105) |
| maritalDivorced | 0.146* |
| | (0.079) |
| maritalSeparated | 0.084 |
| | (0.080) |
| maritalLiving together | 0.118 |
| | (0.082) |
| maritalMarried | 3.067*** |
| | (0.084) |
| Constant | −1.355*** |
| | (0.123) |
| Observations | 9,397 |
| Log Likelihood | −4,588.955 |
| Akaike Inf. Crit. | 9,197.911 |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

violated for the correct estimation process of linear models (especially logistic and lasso models). Correlation values between variables are either low or moderate, the highest value being 0.527, therefore, we can be reasonably confident that included variables are not collinear.

## 2.3 Training and Testing Set

The data set spans the period from January 2, 2008 to December 31, 2018. The data split for training and testing is conducted based on the observations dates. However, the Observation date column contains missing values. Therefore, such null observations, instead of being archived, are randomly divided into training and testing sets, with training accounting for 70% of the data and testing for 30% of the data. For the rest of the sample for which the Observation date is available, we define the testing data split for the dates from January 1, 2016 to December 31, 2018. Combining these two splits, we obtain a training sample containing 7198 observations and a testing sample containing 2802 observations.

Table 3: Descriptive Statistics of numerical variables after the imputation of missing values

| Statistic | N | Mean | St. Dev. | Min | Max |
|---|---|---|---|---|---|
| income | 10,000 | 27,208.210 | 12,798.620 | 5,021.691 | 133,522.000 |
| loan_amount | 10,000 | 20,480.740 | 7,484.806 | 1,414.171 | 50,300.000 |
| term_length | 10,000 | 65.433 | 31.239 | 11.083 | 357.695 |
| install_to_inc | 10,000 | 0.021 | 0.024 | 0.001 | 0.394 |
| schufa | 10,000 | 8,798.115 | 2,025.717 | 4,834.059 | 18,098.360 |

Table 4: Correlation matrix of numerical variables afther the imputation of missing values

| | income | loan_amount | term_length | install_to_inc | schufa |
|---|---|---|---|---|---|
| income | 1 | 0.337 | 0.348 | −0.429 | 0.004 |
| loan_amount | 0.337 | 1 | 0.003 | 0.134 | 0.005 |
| term_length | 0.348 | 0.003 | 1 | 0.527 | 0.006 |
| install_to_inc | −0.429 | 0.134 | −0.527 | 1 | 0.007 |
| schufa | 0.004 | 0.005 | 0.006 | 0.007 | 1 |

### 2.4 KNN Data Imputation for Nonlinear models

After being replaced all the null values for every explanatory variable with feasible values, now target variable's null values can be also handled by replacing them using a very flexible model that requires the minimum assumptions as possible. For the given reasons, K-Nearest Neighbors (KNN) data imputation is performed for use in nonlinear modeling (Random Forests, XGBoost). There are 564 missing values in target variable. Therefore, there are 9436 observations used in the training sample of the model. Cross-validation with 5 folds is applied and k (number of nearest neighbors) is determined based on the model with the highest accuracy value (0.884) to be 29.

## 3. Methodology

This study applies various methodologies for data preprocessing, imputation, and modeling using eight main variables.

### 3.1 Prepocessing

First, we employ data preprocessing to extract raw data, filter, and analyze it. This process is implemented in Step 1 by reading the dataset from the CSV file "Quant Challenge data amended.csv" using the read_csv function from the readr package. Additionally, to ensure reproducibility, the study sets a random seed using set.seed(123). The data preprocessing phase includes handling missing values, formatting date variables, and converting data types to ensure consistency. Missing values represented as "Not avail."(except for categorical variables), "", and "NA"are standardized and replaced with NA. Additionally, the occup column values coded as 1, 2, and 3 are recategorized as "Not recognized."The OBS_DATE column is extracted, stored separately, and reloaded with a formatted datetime structure ( "%d%b%Y - %H:%M:%S") to maintain compatibility for time-based analysis. To facilitate statistical modeling, numerical variables such as "income", "loan_amount", "term_length" , "install_to_inc", "schufa", and "num_applic"are explicitly defined as numerical in the code. The target_var column is treated as a boolean variable, while occup and marital are considered categorical. Data integrity is verified using validation functions such as sum(is.na()), str(), and table() to ensure structural consistency.

### 3.2 Imputation

In the imputation stage, continuous variables with missing values are treated using various statistical techniques. In this model, log-normal transformation is applied to correct for income bias and missing values are imputed with log-normal random values. In addition, the "loan_amount"variable is treated using normal distribution-based imputation, while the "term_length", "install_to_inc"and "schufa"variables undergo log-normal imputation to maintain distribution integrity. We used imputed categorical variables using predictive modeling techniques. A decision tree model (rpart) was trained to predict missing values in marital status using features such as "income", "loan_amount", "occup", "schufa" , and "num_applic". Logistic regression (glm) was applied to impute missing values for "num_applic"based on relevant predictors.

### 3.3 Model training and variable selection algorithm

From a data science perspective, predictability is the primary goal. Ensuring the model generalizes well without overfitting is more critical than the individual contribution of

variables. A variable selection algorithm evaluates all predictor combinations and retains the subset that maximizes predictive performance.

Variable selection is conducted using a regression-based algorithm, testing all possible feature combinations across linear and non-linear models to identify the optimal subset based on performance metrics. Each variable contributes to model performance, even with regularization methods.

Testing all model combinations was computationally efficient, incurring no significant time or cost overhead.

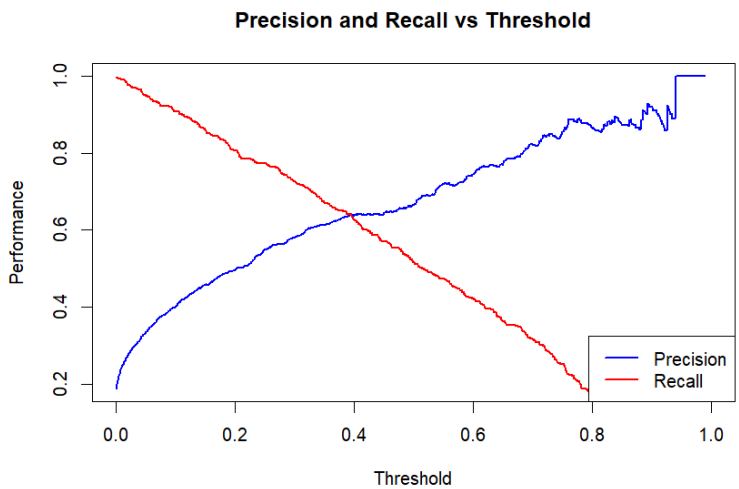
A logistic regression model is trained to predict target_var, evaluated using Precision, Recall, AUC (Area Under the Curve), McFadden's $R^2$, and Bayesian Information Criterion (BIC). To enhance generalization, LASSO (glmnet) and Ridge regression are applied with cross-validation to optimize the regularization parameter (lambda). K-Nearest Neighbors (caret) imputes missing values in target_var, with the optimal k determined via cross-validation.

Machine learning models such as Random Forest (ranger) and XGBoost (xgboost) are trained using cross-validation and hyperparameter tuning to ensure robust predictive performance..

## Selection of the performance metrics:

For classification and regression models in data science, the selection of performance metrics depends on the type of predictions necessary from a business perspective. For example, both recall and precision are commonly used metrics, but there is a known trade-off between them. It is up to the scientist to select one based on the business objective. If predicting default conditions with high accuracy is more beneficial despite the risk of false positives, then recall should be prioritized.

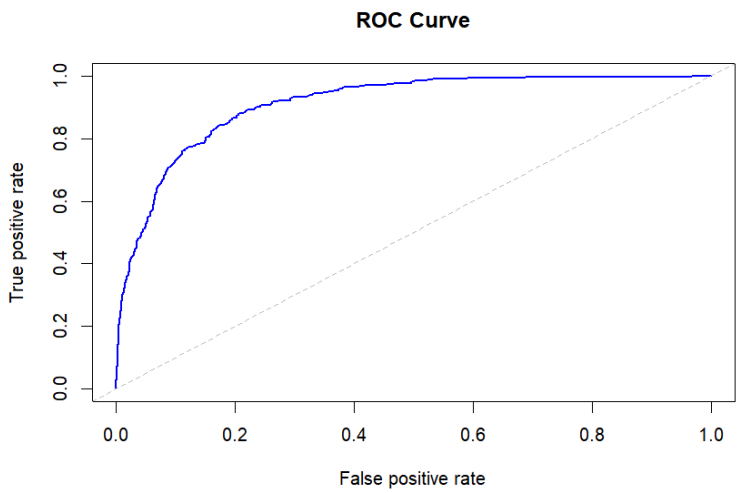
For this project, we assume that both default and non-default predictions are equally important since the bank aims to balance economic loss without penalizing potentially good clients. Therefore, the optimal threshold will be chosen where precision and recall intersect. That is why the recall and precision metrics are calculated for every threshold and are maximized in the threshold where both metrics are equal, that is, in their intersection, as follows in the next plot:




Later on this project, this assumption will be relaxed and we will give intuition to

the fact that in financial analysis where risk aversion behavior is present, weighting the Later in this project, we will explore how risk aversion in financial analysis may justify prioritizing recall. Once the threshold is defined, the model is ready to predict. However, evaluating a model at a specific threshold may be misleading when comparing its general performance across different models. To address this, three additional performance metrics are used: Area Under the Curve (AUC) of the ROC curve, McFadden's $R^2$, and Bayesian Information Criterion (BIC).

$R^2$ is useful when evaluating how well variability is explained compared to the null model and is more aligned with econometric approaches. BIC is effective in preventing overfitting, particularly when dealing with many explanatory variables. Since our dataset consists of only eight well-defined variables, the primary performance metric will be AUC, as it measures overall predictive power independent of threshold selection, making it a reliable basis for model comparison.




ROC curve of the best combination of explanatory variables in the logistic model

## 3.4 Model's tuning and cross-validation parameters

All three methodological frameworks ensure a systematic and rigorous approach to data preprocessing, missing value imputation, and model development and validation. The integration of various statistical and machine learning techniques results in an extremely powerful predictive framework. Model performance is evaluated across multiple evaluation metrics to determine the most effective prediction strategy for a given dataset.

The random forest model is built using mtry = 3, 500 trees, and 10 observations per leaf. Mtry is set as the square root of the number of explanatory variables. Random forests are used for classification.

For boosting, a balanced learning rate with 500 trees is applied. The optimal tune grid is selected by evaluating possible combinations for all folds from cross-validation. A three-level boosting grid is set, and the ROC-AUC metric is used to determine the optimal grid. Cross-validation is set to five folds for both non-linear models. These metrics were selected to reduce model complexity, which is why four-fold cross-validation is used.

For both Lasso and Ridge models, lambda is selected using the "best lambda" technique. A cross-validation algorithm is run with ten folds.

# 4. Results

The aim of this study was to compare different models for predicting credit defaults.

## 4.1 Logistic Regression

| ID | Threshold | Precision | Recall | AUC | $R^2$ | BIC |
|---|---|---|---|---|---|---|
| 255 | 0.394 | 0.6413 | 0.6413 | 0.9110 | 0.4063 | 1438.806 |

Table 5: Logistic Regression Model Performance

With a high AUC, logistic regression demonstrated strong predictive power. Every explanatory variable is used in the model. The trade-off between recall and precision indicates that while the model successfully detects defaulters, some non-defaulters may still be misclassified.

## 4.2 LASSO Regression

| ID | Threshold | Precision | Recall | AUC | $R^2$ | BIC |
|---|---|---|---|---|---|---|
| 255 | 0.392 | 0.6413 | 0.6413 | 0.9108 | 0.4066 | 1437.996 |

Table 6: LASSO Regression Model Performance

LASSO regression performed similarly to logistic regression but had a slightly lower AUC. However, its smaller BIC indicates a better balance between model complexity and fit. LASSO's regularization reduces the influence of less important variables, enhancing generalization.

## 4.3 Ridge Regression

| ID | Threshold | Precision | Recall | AUC | $R^2$ | BIC |
|---|---|---|---|---|---|---|
| 255 | 0.364 | 0.6243 | 0.6270 | 0.9104 | 0.3966 | 1461.165 |

Table 7: Ridge Regression Model Performance

Ridge regression applies regularization to reduce overfitting while maintaining all predictor variables. However, its lower recall and precision suggest it may be less effective at distinguishing defaulters from non-defaulters.

## 4.4 Random Forest

Random Forest captures non-linear relationships well. Its AUC of 0.906 is slightly lower than that of the linear models, but feature importance scores provide insights into which variables influence default prediction. If not carefully tuned, it can be computationally expensive and prone to overfitting.

## 4.5 XGBoost

With the highest AUC of 0.916, XGBoost outperformed all other models. It effectively manages non-linearities and intricate feature interactions. The boosting mechanism cor-

rects errors from previous iterations, improving predictive accuracy. However, it requires significant computational resources and careful tuning.

### 4.6 Evaluation and Validation

We tested the models on new data from 2016–2018. XGBoost performed best with an AUC of 0.916. LASSO and Logistic Regression were closely behind with simple and explainable models. Ridge Regression and Random Forest had slightly lower accuracy but remained viable options.

### 4.7 Cross Validation

The findings from 5-fold cross-validation show that XGBoost achieves an average AUC of 0.9160 with a low standard error of 0.00306, indicating consistent predictive performance across different data splits.

### 4.8 The Precision-Recall Tradeoff

All models exhibited high recall but low precision, meaning they successfully identified most defaulters, reducing financial risk, but also flagged many non-defaulters as defaulters, potentially leading to revenue loss. The best model choice depends on whether the bank prioritizes minimizing defaults or maximizing customer retention.

## 5. Discussion and Conclusion

The results of the research illuminate how a variety of machine learning models differ in terms of predictive performance of a client's default. We have performed a comparison of several linear (logistic, lasso, ridge) and nonlinear (random forrest, XGBoost) methods. AUC was used as a primary measure of predictive power of the models. Among the linear models, logistic regression performed the best (AUC = 0.911). However, nonlinear XGBoost model turned out to be the best performing model overall (AUC = 0.916).

While the AUC has been used as the primary measure of performance, in the context of risk management application, it may be beneficial to consider recall as a second-order measure. Risk-neutral agent would be satisfied with the overall performance represented by the AUC. However, should the financial institution in question display risk-averse behavior, they will be disproportionately focused on correct identification of negative outcomes, which can be measured by recall. Therefore, assigning more weight to the recall in the decision-making process may result into outcome which is more closely aligned to the risk preference of more risk-averse agents. Even-though the numbers differ, still the best model to use is the XGBoost. This result is in line with the expected conclusion regarding model selection, since boosting gives a pallette of flexibility when data predictive-ness is needed and a higher variance is accepted.

## References

Ledolter, J. (2013). *Data mining and business analytics with R.* John Wiley & Sons.

S&P Global (2020). Machine learning and credit risk modelling. Technical report.

Toomey, D. (2014). *R for Data Science.* Packt Publishing Ltd.