UTRECHT UNIVERSITY

Department of Information and Computing Science

Applied Data Science Master thesis

Deploying Deep Learning Techniques and Ensemble Methods to Challenge Logistic Regression for Probability of Default Estimation

First examiner:

Ramón Rico Cuevas

Second examiner:

Dr. I.R. Karnstedt-Hulpus

Candidate:

Allard Willem Tilma

6489990

In cooperation with:

BridgeFund BV

July 4, 2024

Abstract

Correct estimation of probability of default (PD) for credit loans is an essential task for BridgeFund, an online loan broker operating in the Dutch Small and Medium-Sized Enterprizes (SME) market . Advanced machine learning techniques are increasingly being explored to enhance prediction accuracy. Traditional models like logistic regression offer clear interpretability but often lack predictive power compared to more complex algorithms. Ensemble methods and deep learning techniques show potential for significant performance improvements in PD quantification. This study compares XGBoost, Random Forest, Feedforward Neural Networks (FNN) and Tabular Networks (TabNet) against logistic regression to determine their efficacy. The results show that XGBoost outperforms logistic regression and all other models, in all evaluation metrics for PD scoring. However, the "black box" nature of XGBoost raises concerns about model transparency and stakeholder trust, necessitating careful implementation. Developing techniques to demystify XGBoost's decision-making process such as calculation of SHAP values will enhance the model's interpretability and, therefore, applicability.

Contents

1. Introduction

1.1 Scope of BridgeFund

For banks and commercial financial institutes alike, there is a perpetual need to quantify the potential risk of outstanding and newly given loans. Banks aim to integrate the bearing risk in loan pricing by setting interest rate premiums while trying to maintain market competitive prices.

BridgeFund is an online financial institution specializing in providing loans to small and medium-sized enterprises (SMEs) within the Netherlands. SMEs inherently face greater financial risks compared to large corporations due to their typically limited capital reserves and fewer available resources (Cathcart et al., 2020). BridgeFund secures its capital through a network of investors. The company currently has over 4,500 businesses as clients, spanning more than 30 sectors. This broad diversification across various sectors mitigates the risk exposure for BridgeFund, enhancing the stability and resilience of its loan portfolio. Nevertheless, there is an existential need for adequately modeling credit risk given the additional risk that is synonymous with the client pool BridgeFund operates in.

Additionally, BridgeFund aspires to become a data driven FinTech where its products are processed largely automated. This ambition can strictly be met with a well incorporated model that estimates economic consequences of accepting applying businesses adequately. For starters, an estimation should be made of the probability a loan is defaulted.

1.1.1 Use of PD model

Correctly estimating a Probability of Default (PD) model is not the end of the road for BridgeFund; in fact, it is merely the beginning of quantifying its exposed risk and the loss that is accompanied with the defaulted loan.

Expected Loss (EL) quantifies the anticipated average loss a lender, in this case BridgeFund, can expect over a specific period, considering the likelihood of default and the severity of loss if a default occurs. EL is formulated as follows:

$$\text{EL} = \text{PD} \times \text{LGD} \times \text{EAD} \tag{1.1}$$

where,

$$\text{LGD} = \frac{\text{EAD} - \text{Recovery}}{\text{EAD}} \tag{1.2}$$

and

$$\text{EAD} = \text{Current Balance} + (\text{Credit Conversion Factor} \times \text{Undrawn Commitments}) \tag{1.3}$$

Here, Loss Given Default is noted as LGD and Exposure At Default is given as EAD. Recovery is the amount recovered from the defaulted loan, which may include collateral liquidation, guarantees, and any other recoverable amounts.

As can be derived from equation 1.3, EAD is a dynamic value that changes in the direction that Undrawn Commitments (UC) changes. Current Balance (CB) is the amount of money that has been drawn or borrowed by the borrower and is currently owed to BridgeFund. This includes the principal amount and any accrued interest or fees up to the point of calculation. In the context of EAD, this represents the portion of the credit line or loan that is actively being used. Credit Conversion Factor (CCF) is a regulatory multiplier used to convert off-balance sheet exposures (like undrawn commit-

ments) into an equivalent on-balance sheet exposure. This factor accounts for the likelihood that the undrawn portion of a credit line will be drawn down before or at the time of default. The CCF is expressed as a percentage. For instance, a CCF of 40% means that 40% of the undrawn commitment is expected to be utilized if the borrower defaults.

Accurately estimating EL scores and establishing business-level thresholds for these scores enables BridgeFund to implement an automated workflow. In this workflow, incoming loan applications can be accepted or declined based on a pre-trained model. This automation enhances BridgeFund's capacity to process a higher volume of loan applications, reduces labor costs by decreasing the need for risk analysts, and allows for a more personalized product for the customer.

Institutionalized banks are required to measure and apply capital charges in respect of their market risks in addition to their credit risks after the introduction of the first Basel Capital Accord (Basle Committee on Banking Supervision, 1996). This accord has been extended significantly over the years to regulate individual and market risk in the banking's operations. BridgeFund is not an institutionalized bank and therefore has fewer restrictions and obligations regarding their risk strategy and scoring. This allows BridgeFund to deploy more opaque credit scoring techniques than its competitors.

1.2 Literature Review

The Z-score model is an early multivariate formula used to predict the probability of bankruptcy among companies. This model analyzes a combination of five financial ratios, weighted by coefficients derived through discriminant analysis, to classify firms into solvent and insolvent categories. Altman's model was one of the first to use quantitative financial metrics to predict bankruptcy, setting a precedent for the use of statistical and mathematical models in credit risk modeling (CRM). It demonstrated that financial ratios could be systematically combined to assess credit risk, paving the way for more sophisticated models. The use of discriminant analysis

in the Z-score model highlighted the importance of multivariate techniques in CRM. This approach showed that considering multiple financial indicators simultaneously could provide a more accurate prediction of financial distress than univariate analysis (Altman, 1968).

Another proposed technique is the application of Merton's Option Pricing Model (1973) to estimate PD scores. This approach is assessed against the backdrop of Basel II's recommendations for risk capital estimation, which include the standardized approach based on external credit ratings and the Internal Ratings-Based (IRB) approach relying on internal PD estimates. The study finds that the PDs estimated using Merton's model generally increase as credit ratings decline (from AAA to BBB). However, there is considerable overlap in PDs across different rating categories, suggesting that credit ratings may not accurately reflect the true default risk. The PDs show high volatility over time, reflecting the influence of equity market fluctuations on the model's inputs. This high volatility translates to highly variable capital requirements under the IRB approach, posing challenges for bank capital planning. Nevertheless, structural models like Merton's offer a theoretically grounded method to estimate PDs by linking a firm's default risk to its capital structure and asset volatility. This model captures the dynamic nature of default risk more effectively than static credit ratings (Jacob & Gupta, 2005).

Logistic regression is used to develop the O-score model, which estimates the probability of a firm defaulting (PD) within a specific period. The logistic regression approach does not assume linear relationships between variables and the outcome, making it more flexible and robust for real-world applications. Ohlson's model used a set of nine financial ratios and firm characteristics, weighted by coefficients derived through logistic regression, to predict bankruptcy. Logistic regression provides probabilities of default, offering a nuanced view of credit risk rather than a binary classification. This probabilistic approach allows for more flexible decision-making and risk management. Furthermore, the coefficients in a logistic regression model are interpretable as log-odds ratios, making it easier to understand the impact of each predictor on the probability of default. This interpretability is crucial for explaining the model's decisions to stakehold-

ers and regulators (Ohlson, 1980).

The exploration of various methods and applications of credit scoring highlights the widespread adoption of logistic regression in this domain. Logistic regression is particularly favored due to its capability to handle binary outcomes (default/no default) and to provide probabilities of default, which are crucial for risk assessment and decision-making in lending. It is emphasized that logistic regression does not necessitate the stringent assumptions required by discriminant analysis, such as multivariate normality and equal covariance matrices. This flexibility renders logistic regression more robust and applicable to a broader range of datasets, including those with non-normal distributions and heteroscedasticity. (Thomas et al, 2002)

In their examination of the importance of interpretability in logistic regression for PD estimation, various studies emphasize that logistic regression is particularly advantageous due to its clear interpretability. For instance, a study on feature importance measures discusses how logistic regression's coefficients directly indicate the relevance of each predictor variable to the model's output, making it straightforward for analysts to understand and communicate the factors driving default predictions (Saarela, 2021).

1.2.1 Ensemble methods

Traditional models such as logistic regression and decision trees offer transparency but often lack predictive power when dealing with complex, high-dimensional data. In contrast, ensemble methods like Random Forests provide superior accuracy but are criticized for their black-box nature, making it difficult to interpret the results and understand the underlying drivers of credit risk as noted by Aria et al (2021). Random Forests operate by creating an ensemble of decision trees, each constructed from a random subset of the training data through a process called bootstrap sampling. Additionally, at each node of the trees, only a random subset of features is considered for splitting. This method reduces overfitting and improves generalization by ensuring that the trees in the forest are de-correlated. The final predic-

tion of the RF is made by aggregating the predictions of all individual trees, typically through majority voting for classification tasks or averaging for regression tasks (Breiman, 2001).

The interpretability of RFs is hampered by their complex ensemble structure and the way they handle feature interactions. Each RF model can consist of hundreds or thousands of decision trees, each making splits based on different features. This results in an ensemble with numerous decision paths, making it nearly impossible to trace how a specific prediction was derived. Furthermore, RFs model intricate interactions between features, which, while enhancing predictive performance, obscure the direct relationships between input variables and the predicted outcome (Hasties et al, 2009). In credit risk modeling, the need for interpretability is paramount, not only for regulatory compliance but also for ensuring that credit decisions are transparent and justifiable. The application of methods like inTrees and NodeHarvest, as proposed by the authors, can significantly enhance the interpretability of RF models in this domain. By extracting understandable rules and visualizing feature effects, these methods provide insights into the decision-making process, helping stakeholders understand the factors driving credit risk predictions (Aria et al, 2021).

Another widely implemented machine learning technique is XGBoost and LightGBM. The use of LightGBM and XGBoost represents an advanced approach to PD estimation, leveraging ensemble learning techniques to improve predictive accuracy. These models build on the principles of Gradient Boosting Decision Trees (GBDT), enhancing the ability to handle high-dimensional data and complex interactions among variables. By utilizing real transaction data from Lending Club, a peer-to-peer lending platform at the time, the data-driven nature of modern credit risk modeling is exemplified. The algorithms can process vast amounts of data to identify patterns and predictors of default, providing more nuanced and timely risk assessments compared to traditional methods. The significant reduction in default rates achieved by the LightGBM model highlights the potential for machine learning algorithms to enhance the predictive power of PD models. This aligns with the ongoing shift towards more sophisticated risk modeling

techniques in the financial industry, driven by the need for more accurate and reliable credit risk assessments (Ma et al., 2018).

The limitations of existing personal credit risk assessment models, particularly the weak explanatory power and suboptimal feature selection processes, are addressed by proposing a two-fold approach: effective feature selection using Logistic Regression methods and robust default discrimination using the XGBoost algorithm. Effective feature selection is crucial in PD estimation as it determines which borrower characteristics and loan attributes are most predictive of default. By using Logistic Regression and its variants (AIC and BIC), the most relevant features are identified, ensuring that the model is both parsimonious and predictive. The XGBoost algorithm was chosen for its advanced capabilities in handling structured data and capturing complex interactions between features. This approach demonstrated that XGBoost outperformed traditional models like Decision Trees and K-Nearest Neighbors (KNN) in terms of key performance metrics (Wang et al., 2021).

1.2.2 Deep learning

The promising capability of neural networks was highlighted as they were compared to traditional linear scoring models, specifically logistic regression, for predicting credit risk in the credit union context.In the feedforward neural network (FNN), the input layer consisted of various financial and demographic variables related to the credit applicants. The network included one or more hidden layers, with the exact architecture (e.g., the number of hidden layers and the number of neurons per layer) determined based on experimentation to optimize performance. The output layer provided a single output, representing the probability of default. The neural network model demonstrated higher predictive accuracy compared to logistic regression. This was measured using standard metrics such as classification accuracy, sensitivity, specificity, and the area under the ROC curve (AUC). The neural network's ability to capture complex, non-linear relationships among variables contributed to its superior performance. Nevertheless, it

was noted that the FNN is also more prone to overfitting, especially when the dataset is small or imbalanced. Proper regularization techniques and cross-validation were necessary to mitigate overfitting (Desai et al., 1996).

Yeh and Lien (2009) investigate the default payment behavior of credit card clients in Taiwan by comparing the predictive accuracy of six data mining techniques. These techniques include discriminant analysis, logistic regression, Bayes classifier, nearest neighbor, artificial neural networks, and classification trees. The study utilizes a novel "Sorting Smoothing Method" to estimate the real probability of default, given that the actual probability is unknown. The study finds significant differences in classification accuracy among the six data mining techniques. The area ratio proves to be a more sensitive measure than error rates for evaluating model performance. Among the techniques, artificial neural networks (ANNs) demonstrate superior performance with the highest area ratio in the validation set (0.54) and a low error rate (0.17).

The strong mathematical power provided by DNNs is further emphasized through modeling the asset value process of a firm using a stochastic differential equation, specifically an Ornstein-Uhlenbeck (OU) process with jumps. This process reflects the random nature of asset value fluctuations and incorporates sudden changes (jumps), which are realistic in financial markets. The primary objective is to estimate the probability that the asset value falls below a certain threshold (default) within a specified time frame. The PD estimation problem is formulated as a first-passage-time problem, focusing on the first time the asset value hits the default threshold. This leads to the derivation of Partial Integro-Differential Equations (PIDEs) that describe the evolution of the default probability over time. DNNs are utilized to approximate the solutions to these PIDEs. The rationale behind using DNNs lies in their ability to approximate complex, high-dimensional functions without suffering from the curse of dimensionality, where the complexity and computational cost of analyzing and organizing data increase exponentially as the number of dimensions (features) grows. By leveraging the power of DNNs to solve PIDEs derived from complex asset value processes, the approach provides accurate and efficient PD esti-

mates. This method not only enhances the predictive capabilities of credit risk models but also aligns with regulatory requirements, ensuring its practical applicability in financial institutions (Georgiou et al., 2023).

Although FNN have shown promising accuracy for classification, and thus PD estimation, it should be noted that neural networks struggle with the tabular structure of datasets. One of the primary struggles for neural networks on tabular data is their difficulty in effectively capturing and prioritizing interactions between features. Tree-based models, such as Random Forests and Gradient Boosting Machines, naturally handle interactions between features due to their hierarchical structure, which splits the data based on feature values in a way that inherently captures these interactions. The study concludes that while neural networks can perform well on tabular data, they often require extensive hyperparameter tuning and regularization. (Borisov et al, 2021) Furthermore, tabular data often includes a mix of continuous and categorical variables. Neural networks generally require extensive preprocessing, such as one-hot encoding, to handle categorical data. This can lead to high-dimensional feature spaces and increased computational complexity. In contrast, tree-based models can directly handle categorical variables without extensive preprocessing, maintaining efficiency and often leading to better performance (Grinsztajn & Oyallon, 2022)

In a comparative study, the performance of ensemble methods was analyzed against deep learning techniques for PD estimation. The conclusion was that ensemble methods, particularly boosting, provide more consistent and higher predictive performance compared to deep neural networks. Neural networks, while flexible and capable of capturing complex patterns, are sensitive to hyper-parameter settings and may not always outperform simpler ensemble methods in small sample scenarios. Ensemble methods like boosting aggregate predictions from multiple base models, enhancing the robustness and accuracy of PD estimates. Boosting, in particular, improves classification accuracy by sequentially adjusting weights based on previous errors, making it effective in handling imbalanced datasets, common in credit risk modeling. The practical applicability of machine learning models in credit risk management is underscored by their ability to pro-

cess large datasets and uncover patterns that traditional statistical methods might miss. Ensemble learning methods, due to their simplicity and effectiveness, offer a robust alternative to more complex neural networks, especially in scenarios with limited data (Hamori et al., 2018).

A novel approach to predicting the PD for mid-cap companies using transformer models focuses on companies with market capitalizations between USD 1 billion and USD 10 billion, observed over a 30-year period. The transformer-based model demonstrated superior performance in predicting default probabilities compared to traditional models, evidenced by a significant improvement in the Area Under the Curve (AUC) metric. The model effectively integrated various data sources, including fundamental, market, and pricing data. This multi-channel approach allowed the model to utilize a comprehensive set of inputs, enhancing its predictive accuracy. The core of the model is the transformer encoder, which employs a multi-head self-attention mechanism. This mechanism allows the model to consider the entire sequence of input data, capturing complex temporal dependencies and relationships. The use of transformer models represents a cutting-edge advancement in machine learning applied to credit risk modeling. Transformers, originally developed for natural language processing, have shown exceptional performance in capturing complex patterns in sequential data, making them well-suited for PD estimation (Korangi et al., 2023).

To tackle the shortcomings of neural networks on tabular data, an innovative method was presented at the AAAI Conference on Artificial Intelligence. TabNet, a novel deep learning model specifically designed for tabular data, combines the strengths of neural networks with the interpretability of decision trees, offering a new approach to handling structured data commonly found in many practical applications, including credit risk modeling. TabNet uses sequential attention to select the most relevant features at each decision step, allowing the model to focus on different subsets of features for different samples. The model employs a sparse selection mechanism, which reduces redundancy and focuses on the most important features. This mechanism is inspired by decision trees, where only relevant features

are used for splits. For classification problems in tabular data, TabNet outperforms XGBoost, classification trees, and other traditional classification techniques (Arik & Pfister, 2021).

1.3 Formulation of research question

This study aims to challenge the current standard, Logistic Regression, for estimating probability of default scores. The study will deploy ensemble techniques – XGBoost and Random Forest – as well as deep learning techniques – FNN and TabNet – to challenge the benchmark method. Based on this challenge, the following research question can be constructed:

"*To what extent can Logistic Regression be outperformed to estimate probability of default scores by ensemble methods XGBoost and Random Forest and deep learning techniques FNN and TabNet in the Dutch SME market*"

Based on the theoretic framework supplied in section 1.2, it is hypothesized that both ensemble methods and deep learning techniques will outperform Logistic Regression. Moreover, deep learning techniques FNN and TabNet are expected to outperform all other techniques due to their excellent performance in non-linear, big datasets.

2. Method

2.1 Logistic Regression

The use of logistic regression in banking has been well documented. Logistic regression has proven accurate for a great volume of datasets to estimate PD scores. In logistic regression, the relationship between the predictor variables and the probability of an event occurrence is estimated. In this study, the occurring event is a loan going into default. A logit function is used to estimate binary outcomes.

$$\text{logit}(P) = \log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{2.1}$$

where:

- P is the probability of the event occurring (e.g., loan default),
- β_0 is the intercept,
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for the predictor variables $X_1, X_2, \ldots, X_n$.

Logistic regression is a form of supervised learning, meaning the labels are known when training the model. The coefficients are estimated using Maximum Likelihood Estimation (MLE) to maximize the likelihood of observing the given sample. Once the model is trained, it can be used to make predictions on new data. For any new input, the model calculates the linear combination of the input features using the learned coefficients and then applies the logistic function to predict the probability of the outcome.

The coefficients (β) represent the change in the log odds of the outcome for a one-unit increase in the corresponding input variable. In simpler terms,

they indicate how each feature affects the probability of the outcome.

Logistic regression models are well-established and accepted by regulators. Regulatory frameworks often require models that can be easily explained and validated, criteria that logistic regression meets well. Logistic Regression will serve as a performance benchmark for PD estimation due to its simplicity and its proven sufficiency.

2.2 Random Forest

Random Forests are found to be very effective, particularly for classification and regression tasks. A Random Forest is an ensemble learning method that constructs multiple decision trees during training and merges their results to enhance predictive accuracy and control overfitting. A decision tree is a flowchart-like structure where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The core idea is to create a 'forest' of decision trees, each trained on a bootstrap sample of the data—a technique known as bagging (bootstrap aggregating). In each tree, a random subset of features is selected at each split, ensuring that the trees are decorrelated. This randomness in feature selection and data sampling reduces the variance of the model, making it more robust to overfitting compared to single decision trees, which can become highly sensitive to the training data. (Breiman, 2001)

The process of building a Random Forest begins with the generation of multiple bootstrap samples from the original dataset. The construction of each tree involves selecting a random subset of features at each split point, a method that prevents the trees from being overly similar to each other. This randomness introduces diversity among the trees, enhancing the overall generalization ability of the model. Once the forest of trees is constructed, predictions are made by aggregating the results of individual trees. For classification tasks, this is typically done through majority voting, where the class that receives the most votes from all the trees is selected as the final prediction.

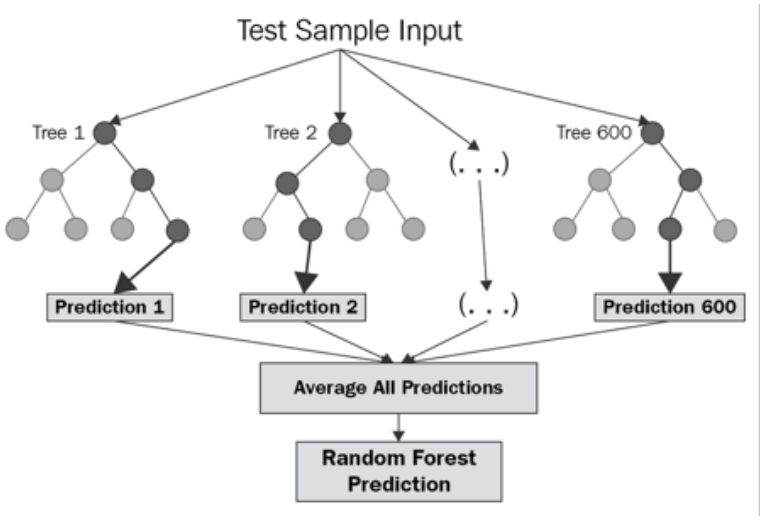

Figure 2.1: Random Forest Architecture (Rudd, 2020)

The strength of Random Forests lies in their ability to handle large datasets with higher dimensionality and their robustness to noise and outliers. They are particularly effective when the relationship between the features and the target variable is complex and nonlinear. (Louppe, 2014)

2.3 XGBoost

XGBoost, short for eXtreme Gradient Boosting, is a highly sophisticated machine learning algorithm that converts weak learners into strong ones, a fundamental concept in ensemble learning.

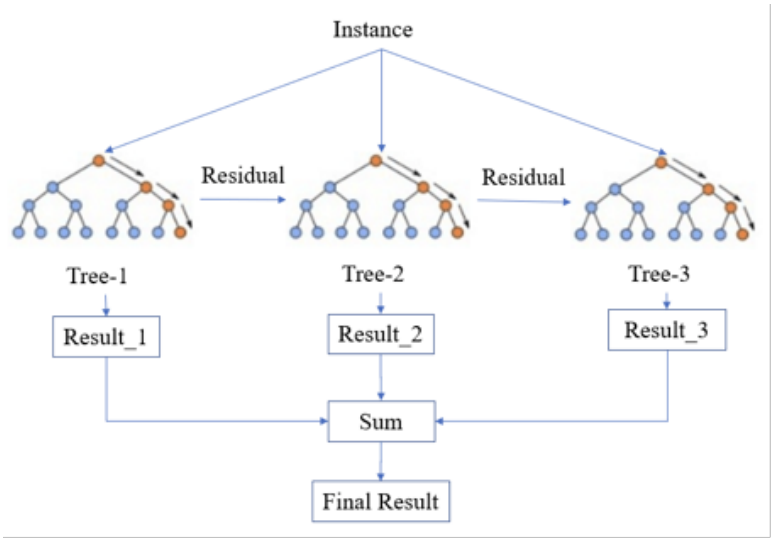

Figure 2.2: XGBoost Architecture (Wang et al, 2020)

XGBoost constructs an ensemble of decision trees in a sequential manner (see figure 2.2). Each new tree in the sequence is built to model and correct the residuals (errors) of the sum of the previously built trees.

The process starts with an instance (data point) that needs to be predicted. Initially, a simple model (which can be a single leaf or a base prediction like the mean of the target variable) provides a preliminary prediction. This initial prediction is used to calculate the residuals. The residuals represent the difference between the actual target values and the predicted values from the initial model. These residuals serve as the new targets for the subsequent tree:

Tree-1: The first decision tree is trained to predict the residuals from the initial model. The results (Result_1) are then summed to improve the initial predictions.

Tree-2: The second tree (Tree-2) is built to predict the new residuals, which are the differences between the actual target values and the updated predictions from Tree-1. The results (Result_2) are added to the sum of previous results.

Tree-3: This process continues iteratively, with each new tree predicting the residuals from the previous step. Tree-3, for instance, predicts the residuals after accounting for the contributions of Tree-1 and Tree-2. The results

(Result_3) are then added to the cumulative sum.

The objective function in XGBoost includes both a loss function and a regularization term. For each tree added to the model, the goal is to minimize the following objective:

$$\text{Objective} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{2.2}$$

Where:

- ℓ is the loss function (e.g., mean squared error for regression tasks).
- y_i is the actual target.
- $\hat{y}_i^{(t)}$ is the predicted target after t trees.
- Ω is the regularization term to prevent overfitting, which includes the complexity of each tree f_k.

The final result is obtained by summing the predictions of all individual trees. Each tree contributes to refining the model's accuracy by focusing on the errors of the previous ensemble. (Ramraj et al, 2016)

2.3.1 Feedforward Neural Network

Feedforward Neural Networks (FNNs) have been a cornerstone in the development of neural network models for various tasks, including classification and regression. An FNN is composed of multiple layers of nodes (neurons), where each layer is fully connected to the next. The basic building block of an FNN is the perceptron, a simple model of a biological neuron that receives inputs, applies a weighted sum, adds a bias, and passes the result through an activation function (Rosenblatt, 1958).

The construction of an FNN begins with an input layer that receives the raw data. This layer is followed by one or more hidden layers, which are responsible for capturing the complex patterns in the data. Each neuron in a hidden layer receives inputs from all neurons in the previous layer, pro-

cesses them using an activation function (ReLU, sigmoid, tanh), and sends the output to all neurons in the next layer. The final layer is the output layer, which provides the model's prediction. In a classification task, the output layer typically uses a softmax activation function to generate probabilities for each class (Goodfellow, Bengio, & Courville, 2016).

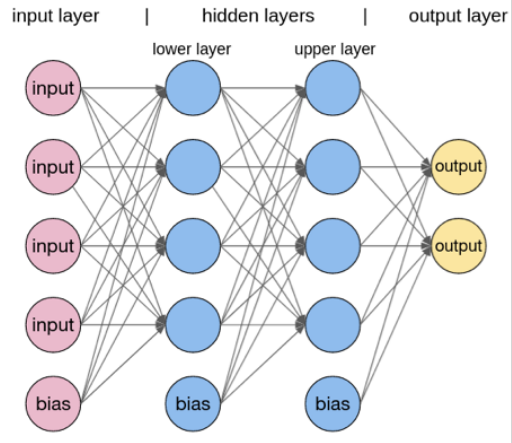

Figure 2.3: Feedforward Neural Network Architecture (An, 2021

Training an FNN involves adjusting the weights and biases of the neurons to minimize a loss function, which quantifies the difference between the predicted and actual outputs. For binary classification tasks, the binary cross-entropy loss function is commonly used. Binary cross-entropy measures the performance of a classification model whose output is a probability value between 0 and 1, as is the case in PD scoring. The function is defined as:

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

where y_i is the actual label, and p_i is the predicted probability. This process is done using backpropagation, an algorithm that computes the gradient of the loss function with respect to each weight by applying the chain rule of calculus. The gradients are then used to update the weights through an optimization algorithm, such as stochastic gradient descent (SGD) or Adam (Kingma & Ba, 2014).

In more mathemetical terms, the FNN architecture can be described as follows. Given an input data vector $x \in \mathbb{R}^p$ and a categorical variable $y \in \{0, 1\}$, the goal is to predict the output $\hat{y} = P(Y = 1 \mid X = x)$. This prediction process involves multiple layers of neurons, indexed from $k = 1$ to l. Layers where $k < l$ are referred to as hidden layers, and the layer l is the output layer. Each of these layers includes a bias term $b_k \in \mathbb{R}$ and multiple units, each represented by an activation $h_i^k \in \mathbb{R}$.

The neurons in each layer k are connected to the neurons in the previous layer, $k - 1$, via a set of weights $w_{ij}^k \in \mathbb{R}$. These weights define how much influence each neuron from the previous layer has on each neuron in the current layer. The initial input data x is weighted by these weights w_{ij}^k as it passes through the network, and each neuron i in every layer k applies these weights to its inputs.

In the final output layer, there is an additional bias term $b_{l+1} \in \mathbb{R}$, and the neurons in this layer are connected to the previous layer by weights $w_j^{l+1} \in \mathbb{R}$. Each neuron's response is determined by an activation function $s_i^k(\cdot)$, which processes the weighted sum of inputs. This activation function generates an output that serves as the input for the next layer. This process repeats iteratively through all layers, continuously adjusting until the network's predictions converge to an optimal value, either local or global. This way, the network learns to accurately map the input data to the desired output. (Sariev & Germano, 2020)

2.3.2 Tabular Network

Feedforward Neural Networks often face difficulties in identifying interactions among features in tabular data. To address this issue, a new training methodology for neural networks tailored for tabular data, known as Tabular Networks (TabNet) is developed. This architecture is specifically designed to effectively capture the relationships among tabular features.

TabNet's architecture is mainly composed of two key components: the Attentive Transformer (AT) and the Feature Transformer (FT). The AT is responsible for selecting the most pertinent features for further processing

stages, while the FT transforms these selected features into more informative representations.

The AT plays a critical role in feature selection at each stage, utilizing sparsemax activation to consider prior scales. Prior scales manage the frequency of feature selection by the model, influenced by their usage in earlier stages. The input to the prior scales comes from the previous AT, providing information on the features' usage history. Similar to the FT, the AT can be implemented as a TensorFlow model and integrated into a broader neural network framework.

An FT consists of a series of feature blocks applied sequentially. Each FT includes two shared blocks (with weights reused across steps) and two step-specific blocks. Using shared weights reduces the number of model parameters, thereby enhancing generalization capabilities. Prior scales regulate the frequency of feature selection by the model. These scales are calculated using the activations from the preceding AT and incorporate a relaxation factor (γ).

$$\mathbf{P}[i] = \prod_{j=1}^{i} (\gamma - \mathbf{M}[j]) \tag{2.3}$$

where:

- $\mathbf{P}[i]$ is the resulting importance of feature i,
- γ is the relaxation factor, and
- $\mathbf{M}[j]$ represents previous sparsemax activations at step j.

Equation 2.3 explains the update mechanism for prior scales. The update involves a product over all preceding steps up to the current step i. Conceptually, if a feature has been heavily utilized in earlier steps, the model shifts focus to the remaining features to avoid overfitting. For instance, with $\gamma = 1$, features with high multiplicative activations (e.g., 0.8) will result in small prior scales $(1 - 0.8 = 0.2)$. Thus, small prior scales reduce the likeli-

hood of such features being selected in the current step.

Hence, the initial input features are processed through the Feature Transformer to generate initial feature representations. The output from the Feature Transformer then serves as input to the Attentive Transformer, which selects a subset of features for the next step. This process repeats for the specified number of steps. The model makes final predictions using outputs from the Feature Transformer at each decision step. By aggregating attention masks at each step, one can identify which features were used in making predictions, thus determining both local and global feature importance.

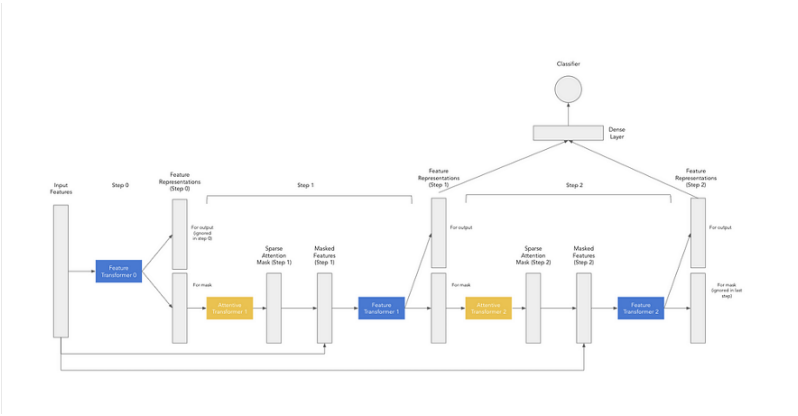

Figure 2.4: TabNet Architecture (Tocilins-Ruberts, 2022)

Sequential attention allows TabNet to dynamically adjust the importance of features at each step, leading to better utilization of the data. For the probability of default estimation, this means the model can more effectively focus on the most indicative features of default risk. By processing features sequentially and re-weighting them at each step, TabNet can capture complex, non-linear relationships that might be missed by simpler models. This capability is crucial for accurately estimating the probability of default, which often involves intricate interactions between financial indicators. The attention mechanism also enhances interpretability by providing insights into which features are most influential at each decision step. This transparency is valuable for understanding the factors driving the model's predictions and for building trust in its estimates of default probability. (Arik et al., 2022)

3. Experimental evaluation

The objective of this study is to challenge logistic regression for PD modeling with deep learning techniques and ensemble methods. Specifically, XGBoost, Random Forest, a fully connected feedforward network (FNN) and TabNet will be deployed to challenge the logistic regression, which has become the standard to estimate PD scores for most banks. A set of explanatory variables are fitted on the dependent variable *'label'* where expression of 1 indicates a default and expression of 0 indicates a non-default.

Performance of each model was measured using the Area Under Curve (AUC), Area Under Curve Precision Recall (AUC-PR), F-score and Brier scores

3.1 Experimental setup

The data used in this study was obtained from loan management system VTiger. VTiger is a Customer Relationship Management (CRM) platform that integrates various functionalities for managing customer data, including loan management. The raw data is migrated from VTiger to AWS S3. Here, the data is stored in the cloud.

To extract raw data from AWS S3 to DataBricks, the DataBricks integrated API and built-in data import features were utilized. The API allowed for programmatically retrieval of data directly from the system, ensuring that the most recent and accurate information was obtained. This data included customer information, loan applications, loan statuses, repayment schedules, and transactional data from applicants. Feature engineering was performed on this data to extract relevant features for the models.

Databricks was utilized extensively in this study to streamline the computation of model predictions and the performance of statistical tests in a

shared environment. Databricks provides a collaborative platform that integrates seamlessly with Apache Spark, allowing for efficient data processing and version control.

For the predictive modeling, both TensorFlow and PyTorch frameworks were employed to build and train neural networks. TensorFlow was used for the initial implementation due to its ease of use and comprehensive documentation, while PyTorch was utilized for its dynamic computational graph and flexibility in model customization. Using TensorFlow, a FNN to predict loan default probabilities was constructed. The model architecture included multiple hidden layers with ReLU activation functions, dropout layers for regularization, batch normalization and a final sigmoid activation for binary classification.

The models were trained on the accepted loans dataset. This is because only for those observations there were available labels. It should be noted that training solely on the accepted loans introduces a selection bias since all loans were deemed low-risk by the risk analyst during the application process.

The best fitting hyperparameters were determined through cross-validation, optimizing for the highest precision.

3.1.1 Weight of Evidence Encoding

The deployed classification methods necessitate numerical input. Therefore, categorical data will be encoded using Weight of Evidence (WOE). WOE encoding transforms categorical data based on the distribution of a binary target variable, in this scenario '*label*', within each category.

One of the primary reasons WOE is widely used in credit risk modeling is that it enhances the model's explainability, which is crucial for regulatory compliance. Financial institutions are required to provide transparent and justifiable risk models. WOE encoding helps in this by creating a monotonic relationship between predictor variables and the target variable, making it easier to interpret and explain the model's decisions. (Lund, 2016)

It is particularly useful for creating features that are linearly separable and thus more compliant to the linear transformations within neural networks. WOE is calculated as:

$$\text{WOE}_i = \log\left(\frac{\frac{N_i^{Good}}{N^{Good}}}{\frac{N_i^{Bad}}{N^{Bad}}}\right) \tag{3.1}$$

where:

$$\begin{aligned} N_i^{Good} &= \text{Number of good observations in group } i \\ N^{Good} &= \text{Total number of good observations} \\ N_i^{Bad} &= \text{Number of bad observations in group } i \\ N^{Bad} &= \text{Total number of bad observations} \end{aligned}$$

3.1.2 Z-Score normalization

In credit risk modeling, Z-score normalization is particularly beneficial because it standardizes financial metrics and personal information features, making the model less susceptible to the distorting effects of outliers. Credit risk models often deal with data that can have extreme values due to atypical financial behaviors or data entry errors. By transforming the data to have a mean of zero and a standard deviation of one, Z-score normalization reduces the influence of these extreme values. This standardization process helps to mitigate the impact of outliers, leading to more reliable credit risk predictions. (Hand & Henley, 1997)

In neural network training, Z-score normalization is equally crucial. Neural networks, particularly deep learning models, are sensitive to the scale of input data. Disparities in feature scales can lead to inefficient training and suboptimal model performance.(LeCun et al. , 1998)

The normalization of input data ensures that the gradients during training are more balanced, thereby facilitating faster convergence of the gradient descent optimization process. This balancing act is crucial for avoiding problems such as exploding or vanishing gradients, which can severely impede the training process of deep neural networks. Furthermore, standardized data enables the activation functions,, in this studt ReLU, to operate more effectively by ensuring the input values fall within an optimal range, preventing issues like saturation that can occur with unnormalized data.

The process of Z-score normalization involves calculating the mean and standard deviation of the dataset, then transforming each data point using the formula:

$$z = \frac{x - \mu}{\sigma} \tag{3.2}$$

where:

$z =$ The z-score (normalized value)

$x =$ The raw score (original value)

$\mu =$ The mean of the population

$\sigma =$ The standard deviation of the population

3.2 Evaluation Metrics

3.2.1 AUC

AUC is an essential metric for classification models. Calculating AUC requires the predicted scores column and the actual label column, the so-called ground truth. AUC can be derived by calculating the True Positive Rate (TPR) and False Positive Rate (FPR). The calculation of AUC can be

written as follows:

$$\text{AUC} = \frac{\text{TPR}}{\text{FPR}} \tag{3.3}$$

where

$$\text{TPR (Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.4}$$

and

$$\text{FPR (Specificity)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3.5}$$

As can be derived from equation (3.3), a high AUC indicates a higher discriminatory capability of the model (A combination of high TPR and low FPR will result in a high AUC)

3.2.2 F1 Score

Another metric to estimate the performance of classification models is the F1 score. F1 consists of two main components:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.6}$$

and

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.7}$$

The F1 score is the harmonic mean of these two metrics. The harmonic mean is used instead of the arithmetic mean because it punishes extreme values more, ensuring that a high F1 score can only be achieved if both precision and recall are reasonably high. The formula for the F1 score can be given as:

$$\text{F1 Score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{3.8}$$

There is an inherent trade-off between precision and recall. Increasing precision typically reduces recall and vice versa. The F1 score provides a single metric that captures this trade-off.

3.2.3 AUC-PR

Another metric to estimate the performance of classification models is the Area Under the Precision-Recall Curve (AUC-PR). The AUC-PR evaluates the trade-off between precision and recall across different threshold settings.

The precision-recall curve plots precision (y-axis) against recall (x-axis) at various threshold levels. Unlike the ROC curve, which can be overly optimistic for imbalanced datasets, the precision-recall curve provides a more informative picture when dealing with skewed classes.

The formula for AUC-PR is given by:

$$\text{AUCPR} = \sum_{i=1}^{n-1} (R_{i+1} - R_i) \cdot \frac{P_{i+1} + P_i}{2} \tag{3.9}$$

where:

- P_i is the precision at the i-th threshold,
- R_i is the recall at the i-th threshold,
- n is the number of precision-recall points.

In formula 3.9, the precision-recall pairs (P_i, R_i) are sorted by recall values. The AUC-PR is approximated using the trapezoidal rule, which sums up the areas of the trapezoids formed by consecutive points.

Higher AUC-PR values indicate better model performance, reflecting a higher precision and recall trade-off. The AUC-PR metric is particularly useful for evaluating models in contexts where the positive class is of greater interest and more critical than the negative class, as is the case in this study.

3.2.4 Brier Score

The Brier score is another critical metric for evaluating the performance of probabilistic predictions in classification models. It assesses the mean squared difference between the predicted probability and the actual outcome, providing insights into both the accuracy and calibration of the model's predictions. The score ranges from 0 to 1, with 0 indicating perfect predictions and 1 representing the worst possible accuracy. This renders the Brier score especially valuable in PD estimation, where the output is probabilistic rather than binary.

The Brier score is mathematically defined as follows:

$$\text{Brier Score} = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2 \tag{3.10}$$

where

- N is the number of forecasts,
- f_i is the forecast probability for the i-th instance,

- o_i is the actual outcome for the i-th instance (1 if the event occurred, 0 if it did not).

3.3 Evaluation of Data

3.3.1 Dependent variable

The variable to be estimated is noted as *'label'*. The variable indicates a state of default for the loan. The accuracy of PD models significantly hinges on the precise definition of what constitutes a "default." An incorrect or ambiguous definition can lead to flawed risk assessments, inappropriate capital allocation, and ultimately, financial instability for BridgeFund. Therefore, a clear and consistent definition of default is paramount. A well-defined default criterion is essential for the robustness of statistical models used in PD estimation, ensuring that the predicted probabilities align with the actual risk of default. (Bellotti & Crook, 2009)

In the context of defining defaults for PD estimation, a common threshold used in the financial industry is the number of days past due. (Basel, 2006) (EBA, 2019) BridgeFund has set an internal business rule stating that loans with a days past due over 60 days is considered a defaulted loan. Therefore, this rule will be maintained in this study.

Depicted in Figure 3.1, is the vast class imbalance in the dependent variable. There is a clear majority in the non-default class, as can be expected.

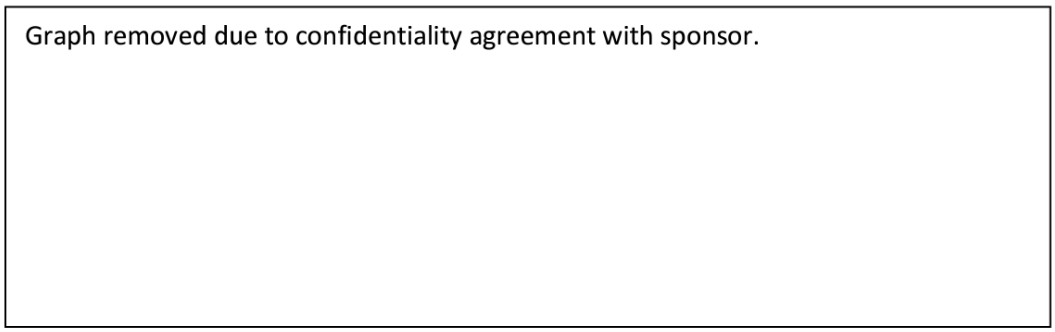

Figure 3.1: Class imbalance in the dependent variable

To address this issue, cross-validation, particularly stratified k-fold cross-

validation, is applied when training the models. Cross-validation mitigates the risk of biased performance estimates by ensuring that each fold used for training and validation maintains the original class distribution, thereby providing a balanced evaluation environment (Kohavi, 1995). This method helps in understanding of model performance across different subsets of the data, including the minority class, which might otherwise be underrepresented in a single train-test split (Wong & Yeh, 2020). Furthermore, cross-validation is employed here to facilitate robust hyperparameter tuning, as the model is tested on multiple data splits, enhancing its ability to generalize well on unseen data (Kohavi, 1995). By averaging evaluation metrics across folds, cross-validation offers a reliable measure of the model's capability to handle imbalanced datasets, thus ensuring a more accurate and fair assessment of its performance (Japkowicz & Stephen, 2002). A deliberate choice is made when opting not to correct for the aforementioned class imbalance. The implementation of techniques such as random undersampling, random oversampling, and SMOTE in PD estimation lead to poorly calibrated models, with a significant overestimation of the likelihood of defaults for minority classes. These resampling methods do not enhance the area under the ROC curve compared to models that do not address class imbalance (see Appendix for proof). This indicates that while resampling techniques aim to rectify class imbalance, they may not necessarily contribute to more accurate PD predictions.(Van den Goorbergh, 2022)

3.3.2 Missing data

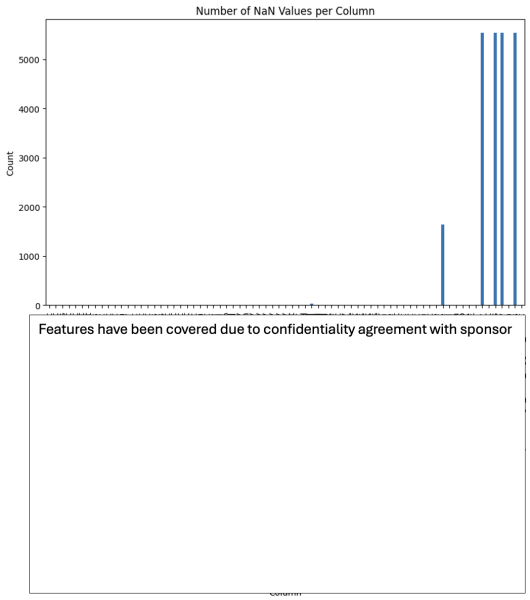

Figure 3.2: Missing values

Figure 3.2 shows the sum of missing values per column. The four features on the right side of the graph show missing entry for all observations, therefore these features are simply removed from the dataset. Some other features show some cases of missing values. The proportion of missings in these features are negligible and therefore, are simply replaced with their respective median for numerical features and respective mode for categorical variables.

3.3.3 Multivariate analysis

The cross-correlation matrix visualized in figure 3.3 provides a detailed view of the relationships between multiple features used in the study. This analysis aims to identify significant correlations, potential multicollinearity issues, and insights that can be drawn from the data.

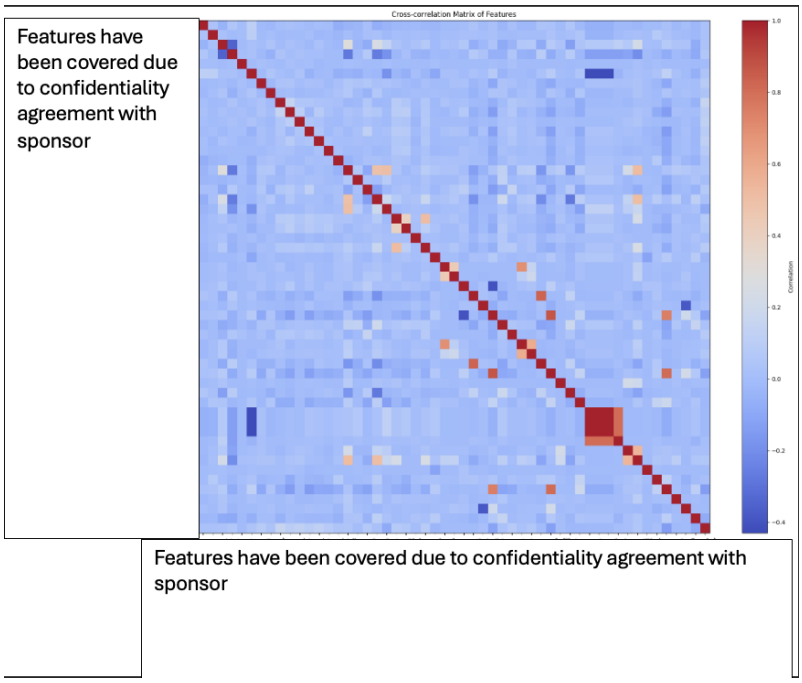

Figure 3.3: Correlation Matrix

Multicollinearity, a pair of extremely highly correlated explanatory variables, can distort the results of a regression analysis, leading to unreliable and unstable estimates of regression coefficients. In classification models, multicollinearity can decrease the precision of the estimated coefficients, making it difficult to assess the relative importance of each predictor. When multicollinearity is present, the standard errors of the coefficients tend to increase. This means that even if the overall model fits well, individual predictors might appear insignificant, leading to potential misinterpretation of the model. Additionally, multicollinearity can make the model more sensitive to changes in the model or the data, which can reduce the model's predictive performance.

To address multicollinearity, one feature was systemetically removed from each pair of highly correlated features. The criterion for removal was a correlation coefficient greater than 0.8. The goal was to retain as much unique information as possible while minimizing redundancy and improving the model's performance. The refined correlation matrix is depicted in figure 3.4.

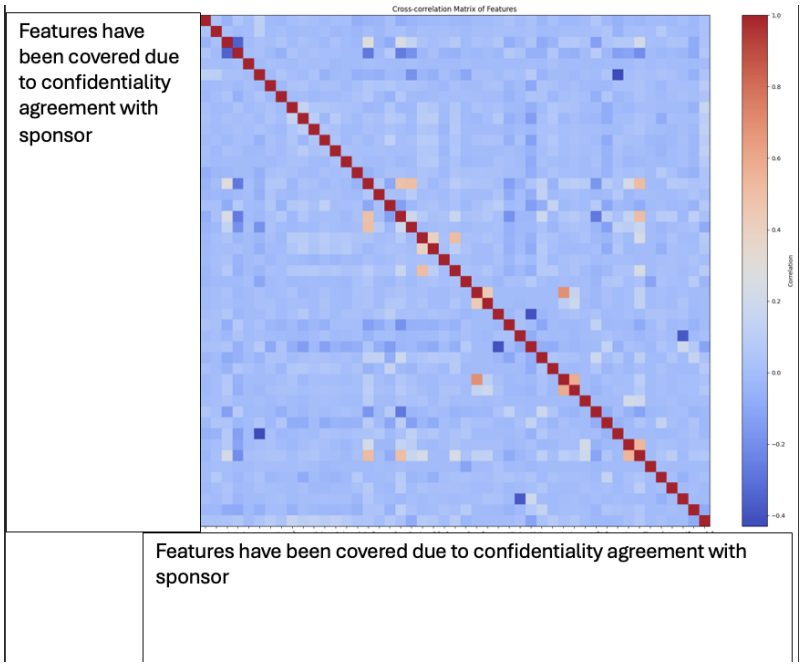

Figure 3.4: Refined Correlation Matrix

3.3.4 Univariate analysis

The array of boxplots provided in Appendix A gives a visual summary of each feature's distribution, highlighting central tendency, spread, and outliers. These outliers are corrected for using Z-Score normalization. In the boxplot for label it can be visually assessed there is a class imbalance, as the label = 1 are depicted as outliuers.

3.4 Experimental results

In this section, a comparison of performance of five different classification models used to estimate the PD scores is presented. The models evaluated include Logistic Regression, Random Forest, XGBoost, Feedforward Neural Network (FNN), and TabNet. The performance of these models is assessed using four key metrics: Area Under the Curve (AUC), F1 Score, Area Under Curve Precision Recall (AUC-PR) and Brier Score. The performance will be visually assessed using ROC, PR curves, Kernel Density Estimation (KDE) plots and calibration plots. The statistical results are summarized in Table 3.1.

3.4.1 Probability distributions

The distribution of PD scores exhibits a notable shape that offers insights into the risk assessment process of loan applications. The PD scores are distinctly peaked between the range of 0.05 to 0.10, suggesting that the majority of loans are perceived to carry a low risk of default. This peak aligns with the underlying class imbalance present in the dataset. Such a distribution indicates a conservative lending strategy where loans are primarily approved for applicants who are considered low-risk by the risk analysts.

The tail of the distribution, which extends from around 0.3 to 1, reflects a smaller subset of loans that are perceived to be high-risk. This tail is relatively sparse, which is consistent with the class imbalance and further underscores the predominance of low-risk loans within the accepted pool. The presence of this tail, albeit thin, is crucial as it highlights that the risk model is capable of identifying loans with a high likelihood of default.

The nature of this distribution can be attributed to the preliminary manual approval process by risk analysts. Before any loan is included in the dataset and subsequently scored by the model, it has already passed an initial human evaluation. This manual screening process likely filters out the riskiest loan applications, resulting in a dataset where most loans are inherently low-risk. Therefore, the model's PD scores are skewed towards the lower end.

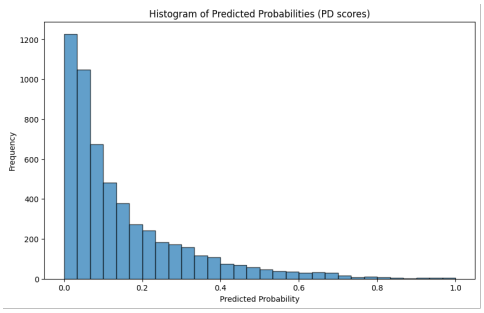

Figure 3.5: Distribution Logistic Regression

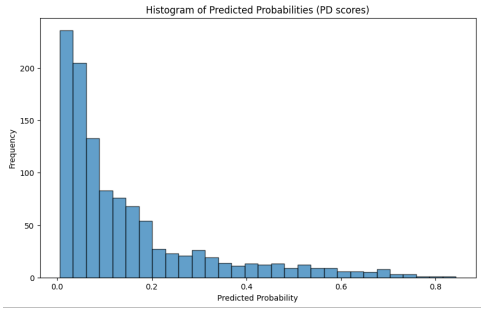

Figure 3.6: Distribution XGBoost

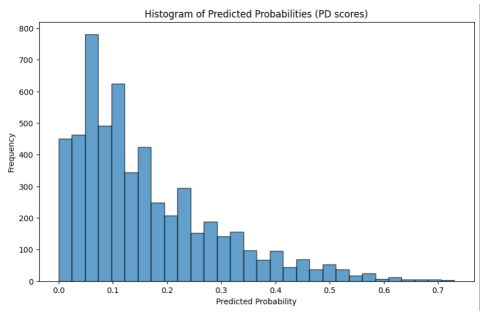

Figure 3.7: Distribution Random Forest

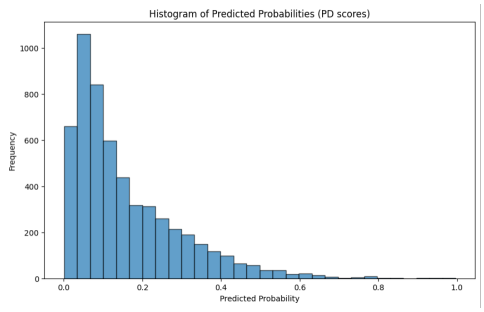

Figure 3.8: Distribution FNN

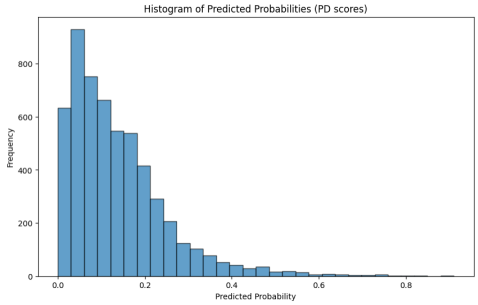

Figure 3.9: Distribution TabNet

The histogram of predicted PDs from the Random Forest model, as shown in figure 3.7, exhibits a distribution with some noticeable characteristics, including random local peaks. This pattern can be attributed to the inherent properties of the Random Forest algorithm and the nature of the data it processes.Random Forest, being an ensemble learning method, constructs multiple decision trees and aggregates their predictions to form a final output. Each tree is trained on a different subset of the data, selected through bootstrapping, and considers a random subset of features for splitting nodes. This randomness introduces variability in the model's predictions, leading

to the observed local peaks in the distribution of PD scores. These peaks represent areas where individual trees in the forest are likely to agree on certain predictions more strongly due to the specific splits and data subsets they were trained on.

The differences in the distributions of PD scores among various models stem from their distinct mechanisms of learning and prediction. Logistic regression, being a linear model, tends to produce a smoother probability distribution as it fits a single linear decision boundary across the feature space. XGBoost, which is a gradient boosting algorithm, builds trees sequentially to correct the errors of the previous ones, resulting in more refined predictions. FNN and TabNet display more continuous distributions of PD scores. They learn through backpropagation and gradient descent, which optimizes the entire network's weights simultaneously, contributing to the overall smoothness of their prediction distributions.

3.4.2 Kernel density plots

The Kernel Density Estimate (KDE) plot of predicted probabilities provides an approximation of the distribution of the predicted PD scores for the two classes: non-defaulted loans and defaulted loans.

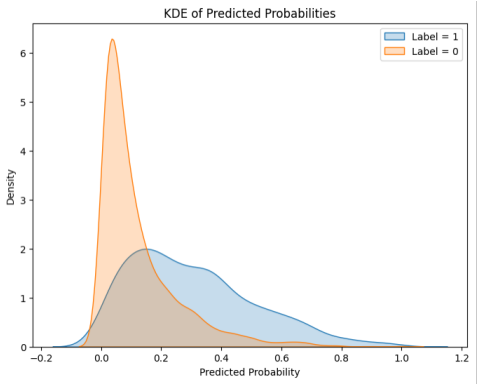

Figure 3.10: KDE for Logistic Regression

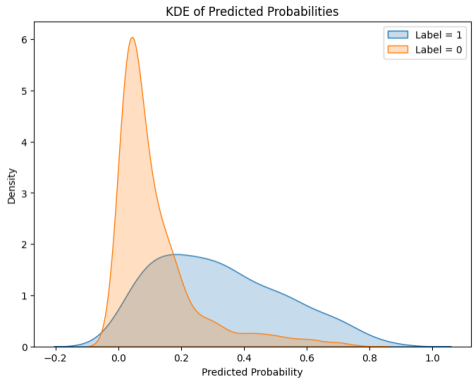

Figure 3.11: KDE for XGBoost

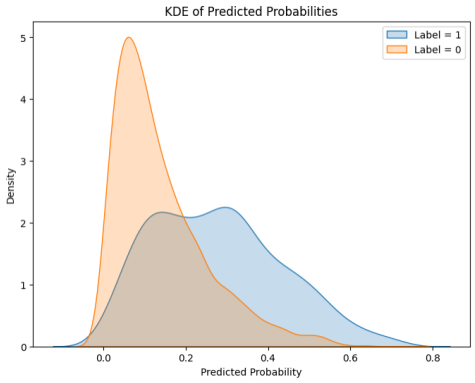

Figure 3.12: KDE for Random Forest

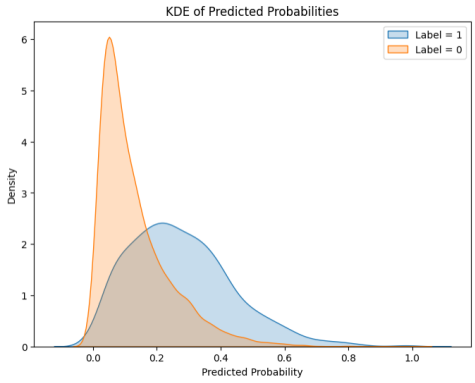

Figure 3.13: KDE for FNN

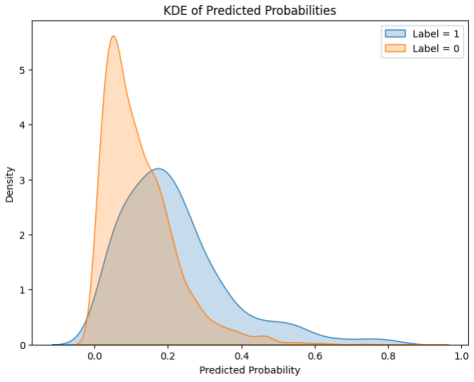

Figure 3.14: KDE for TabNet

The KDE is a non-parametric way to estimate the probability density function of a variable. It works by placing a Gaussian kernel on each data point and summing these kernels to produce a smooth curve. The bandwidth of the kernel controls the smoothness of the KDE: a smaller band-

width leads to a more sensitive, less smooth estimate, while a larger bandwidth results in a smoother estimate. (Gan & Bailis , 2017)

The KDE plots show a prominent peak for non-defaulted loans (Label = 0) around a predicted probability close to 0. This indicates that the models frequently assign very low probabilities of default to non-defaulted loans. For defaulted loans (Label = 1), the density is more spread out with a peak around a predicted probability of approximately 0.1 to 0.2. This suggests that the model is more conservative and assigns a range of probabilities rather than clustering at low values.

The area under the orange curve is larger, indicating the higher frequency of non-defaulted loans in the dataset. There is a region where the densities of the two classes overlap, particularly around the predicted probability of 0.1 to 0.3. This overlap indicates the model's uncertainty in distinguishing between defaulted and non-defaulted loans within this probability range. The overlap suggests that some defaulted loans receive low predicted probabilities (leading to false negatives), and some non-defaulted loans receive higher predicted probabilities (leading to false positives).

3.4.3 ROC curves

The ROC curve is a graphical representation of the model's performance across various threshold settings. It plots the TPR against the FPR at different thresholds, illustrating the trade-off between sensitivity and specificity. XGBoost exhibits the best performance with the highest AUC of 0.83, effectively distinguishing between defaults and non-defaults. FNN and Logistic Regression have comparable performances with AUCs of 0.79, capturing substantial patterns but missing some complexities. TabNet and Random Forest, both with AUCs of 0.77, show good performance but with more variability and slightly lower discriminative power.

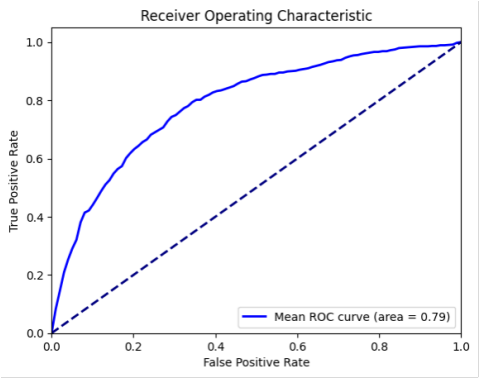

Figure 3.15: ROC Curve for Logistic Regression

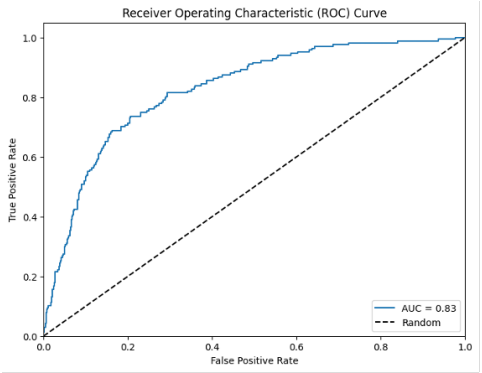

Figure 3.16: ROC Curve for XGBoost

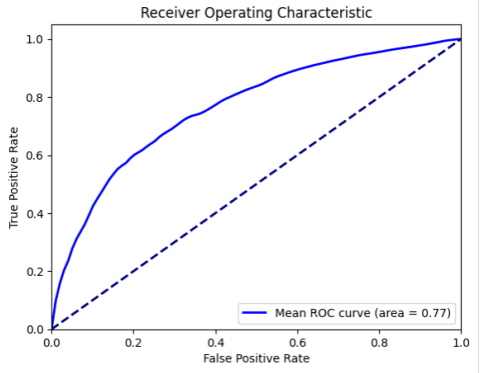

Figure 3.17: ROC Curve for Random Forest

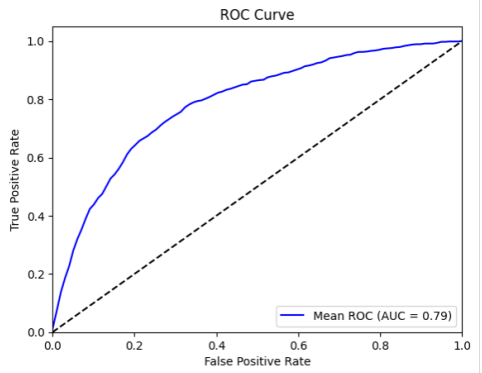

Figure 3.18: ROC Curve for FNN

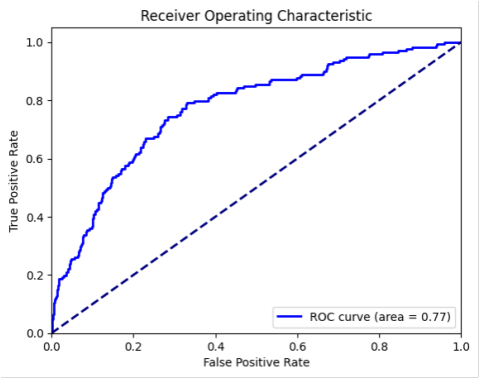

Figure 3.19: ROC Curve for TabNet

3.4.4 Precision-Recall Curves

Due to the class imbalance present in the dependent variable, looking at AUC curves alone would give a biased estimate. AUC measures the over-

all ability of the model to discriminate between classes. However, it does not account for the skewed distribution of classes. In a highly imbalanced dataset, AUC can present an overly optimistic view of model performance because it treats all FPR and TPR equally, regardless of the actual class proportions. For instance, a model predicting strictly non-defaults in a dataset with 95% non-defaults, an accuracy of 95% would still be measured, gravely overestimating the model's performance. (Buda et al, 2018)

Hence, an investigation will be made of the Precison-Recall (PR) curves. Unlike ROC curves, PR curves ignore true negatives and provide a clearer picture of the classifier's performance on the positive class. This makes PR curves more appropriate for highly imbalanced datasets where the number of true negatives overwhelms the number of true positives and false positives. (Grau et al, 2015)

Based on the analysis of the PR curves, the XGBoost model (Figure 3.23) demonstrates the best performance, with the highest average precision of 0.48. This model maintains a higher precision across a broader range of recall values compared to the other models evaluated. In contrast, the FNN (Figure 3.20) and Tabular Network (Figure 3.21) show similar and lower performance, with average precisions of 0.34 and 0.31, respectively. The Logistic Regression model (Figure 3.22) performs slightly better with an average precision of 0.36, maintaining relatively higher precision over a longer range of recall values. The Random Forest model (Figure 3.24), with an average precision of 0.39, performs better than the FNN Tabular Network, and Logistic Regression but falls short of XGBoost.

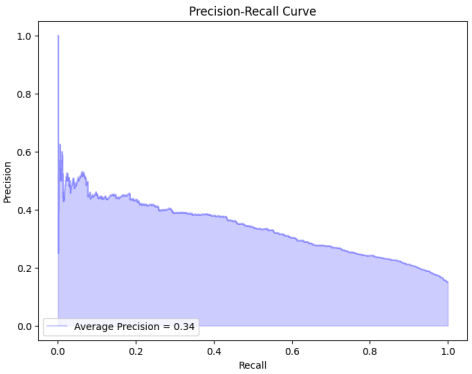

Figure 3.20: PR Curve Feedforward Neural Network

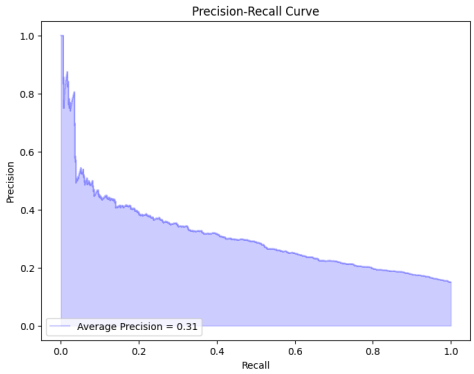

Figure 3.21: PR Curve Tabular Network

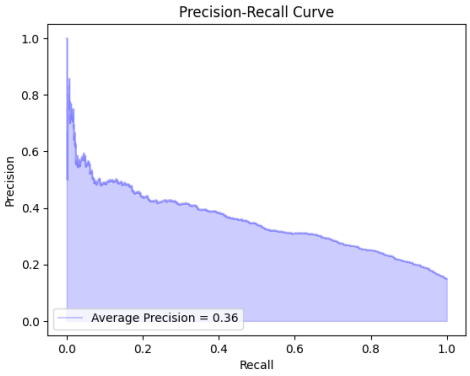

Figure 3.22: PR Curve Logistic Regression

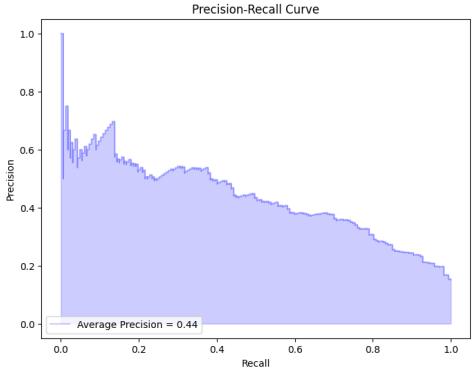

Figure 3.23: PR Curve XGBoost

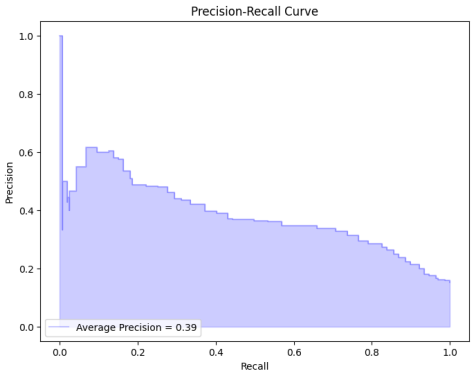

Figure 3.24: PR Curve Random Forest

3.4.5 Calibration slopes

Calibration refers to the agreement between predicted probabilities and actual outcomes. A well-calibrated model means that when it predicts a 20%

probability of an event, that event occurs about 20% of the time. For all models, the calibration slopes indicate that the models are well-calibrated for predicted probabilities up to around 0.50. Beyond 0.50, the models exhibit overconfidence and underconfidence, meaning that their predicted probabilities deviate from the actual observed frequencies. Overconfidence occurs when the model's predicted probabilities are higher than the observed frequencies, while underconfidence occurs when the predicted probabilities are lower.

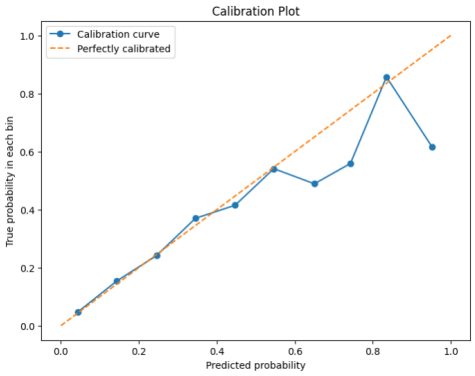

Figure 3.25: Calibration Plot for Logistic Regression

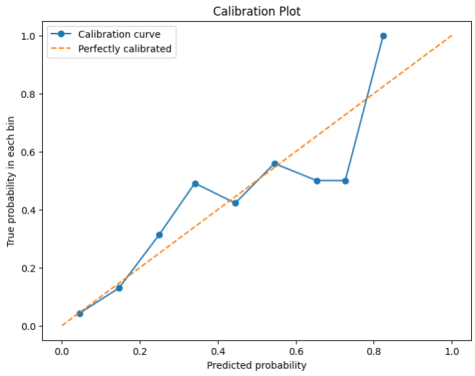

Figure 3.26: Calibration Plot for XGBoost

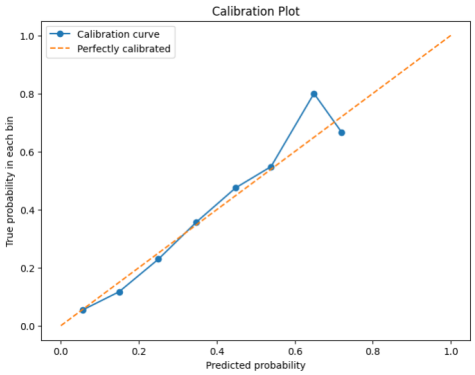

Figure 3.27: Calibration Plot for Random Forest

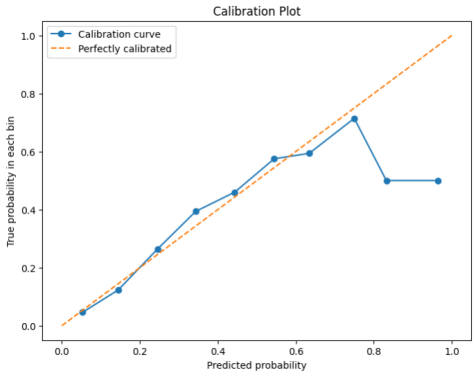

Figure 3.28: Calibration Plot for FNN

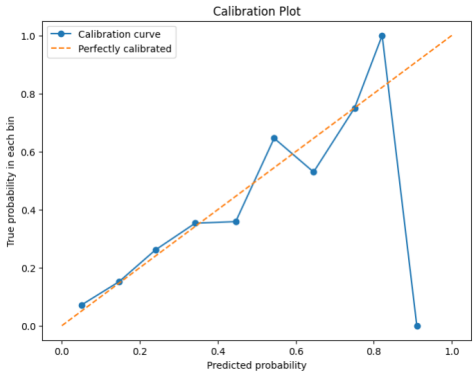

Figure 3.29: Calibration Plot for TabNet

The primary reason for the models' poor calibration beyond 0.50 is the scarcity of observations with high PD scores. Since most loans are deemed low-risk and receive low PD scores, there is insufficient data to reliably estimate probabilities in the higher range. This lack of high-PD observations

leads to inaccurate probability estimates, resulting in overconfidence and underconfidence.

Still, there is a notably difference in how poorly-calibrated the models are from the 0.5 point onwards. Logistic Regression is a linear model, and its probability predictions are based on a sigmoid function applied to a linear combination of features. This simplicity ensures good calibration in the lower probability ranges but can struggle with more complex relationships at higher probabilities, leading to underconfident predictions. In Random Forests, each tree is trained on a random subset of the data, leading to variability in predictions. This ensemble approach results in good average performance but also introduces randomness, which may cause the erratic behavior seen in the calibration curve at higher probabilities. Nevertheless, based on the depicted plots, it can be concluded that Random Forest is the best calibrated model.

For TabNet, the curve shows a steep rise and then a sharp decline, indicating significant overconfidence followed by underconfidence at higher predicted probabilities. TabNet's attention mechanism might emphasize certain features disproportionately in high-risk cases, resulting in overconfident predictions. When these features do not generalize well, the model's confidence drops sharply, leading to the observed decline.

Poor calibration at higher PD scores affects the model's generalizability, particularly in BridgeFund's business context, where accurate high-risk predictions are crucial for business longevity. Models that become overconfident or underconfident at higher PDs may not perform reliably when applied to new data, particularly if that data includes more high-risk observations.

3.4.6 Statistical results

The results in Table 1 provide insights into the models' ability to distinguish between labels (AUC and AUC-PR), their balance between precision and recall (F1-score), and the accuracy of their predicted probabilities (Brier score).

Logistic Regression demonstrates a commendable performance with an

Method	AUC	F1 Score	Brier Score	Precision
Logistic Regression	0.79	0.27	0.11	0.36
Random Forest	0.77	0.16	0.11	0.39
XGBoost	0.83	0.31	0.10	0.48
FNN	0.79	0.27	0.10	0.34
TabNet	0.77	0.27	0.12	0.31

Table 3.1: Performance Metrics of Different Methods

AUC of 0.79, indicating a good capability to distinguish between defaulted and non-defaulted loans. Its F1-score is 0.27, which, although not high, suggests a reasonable balance between precision and recall. The Brier score of 0.11 indicates that the probability predictions are fairly accurate. Logistic Regression's performance is reliable and consistent, reflecting its strength in handling linear relationships within the data.

Random Forest shows slightly weaker performance with an AUC of 0.77, implying a somewhat reduced ability to differentiate between the two default states compared to Logistic Regression and FNN. The F1-score is notably lower at 0.16, indicating challenges in achieving a balanced trade-off between precision and recall, possibly due to overfitting issues inherent in its ensemble nature. However, the Brier score of 0.11 is similar to that of Logistic Regression, suggesting reasonable accuracy in its probability predictions.

XGBoost emerges as the best-performing model across all metrics. It has the highest AUC of 0.83, showcasing its superior ability to distinguish between defaulted and non-defaulted loans. The F1-score of 0.31 is also the highest, indicating the best balance between precision and recall among the models tested. Additionally, the Brier score of 0.10, the lowest among all models, signifies the most accurate probability predictions.

The FNN exhibits similar performance to Logistic Regression, with an AUC of 0.79, reflecting good discriminatory power. The F1-score is 0.27, indicating a reasonable balance between precision and recall, comparable to Logistic Regression. The Brier score of 0.10 suggests accurate probability predictions, slightly better than Logistic Regression. The FNN's ability to capture non-linear relationships in the data contributes to its strong perfor-

mance, although it does not surpass XGBoost.

TabNet shows an AUC of 0.77, on par with Random Forest, indicating a decent but not outstanding ability to distinguish between the labels. The F1-score is 0.27, similar to Logistic Regression and FNN, suggesting a reasonable balance between precision and recall. However, its Brier score of 0.12 is the highest among all models, indicating less accurate probability predictions.

3.4.7 Feature Importance

Table 6.1 in the Appendix shows the feature importance table derived from the logistic regression. The table illustrates which factors most strongly influence the likelihood of default. The coefficients, along with their standard errors (SE), highlight the relative importance and reliability of these predictors.

Understanding the economic implications of these features is crucial for BridgeFund for effective risk assessment and management. Features like `Feature 2` highlight the increased risk associated with borrowers consolidating existing debts, indicating potential financial distress. Higher `Feature 4` reflects greater financial burdens, thereby elevating default risk. Larger `Feature 16` correlate with a higher risk of default due to the substantial obligations they represent. The feature `Feature 25` indicates that more extended transaction histories facilitate better credit assessments, reducing default risk. A higher `Feature 31` denotes stable and compliant financial behavior, which reduces default risk. Finally, `Feature 37` suggests that while reliance on a major client can be risky, a higher percentage from a reliable client may indicate stability and lower default risk.

The selection bias arising from using only accepted loans implies that the model may not generalize well to all loan applicants. The requirement of a revenue threshold of at least 45.000EUR signifies that features related to lower revenue may be deemed insignificant. Likewise, the strict threshold of company age exceeding 12 months could result in features related to company age are regarded insignificant for PD estimation.

3.5 Discussion

In this study, an evaluation is made of the performance of five different classification models in estimating PD scores: Logistic Regression, Random Forest, XGBoost, Feedforward Neural Network (FNN), and TabNet. The models were assessed using key metrics including AUC, F1 Score, Brier Score, and Precision. The results, summarized in Table 3.1, provide insights into each model's ability to distinguish between defaulted and non-defaulted loans, balance precision and recall, and accurately predict probabilities.

The histogram of predicted PD scores, particularly from the Random Forest model, reveals a distinctive distribution (Figure 3.7). The scores predominantly cluster between 0.05 and 0.10, indicating a conservative risk assessment approach where most loans are considered low-risk. This distribution reflects the underlying class imbalance. The thin tail extending to 1 suggests the model's ability to identify high-risk loans, albeit these instances are rare due to initial human vetting of loan applications.

Random Forest, as an ensemble learning method, introduces variability through bootstrapping and random feature selection, which can create local peaks in the distribution of PD scores. These peaks represent the model's consensus on certain predictions due to specific data subsets used in training individual trees.

The KDE plots (Figures 3.10-3.14) offer a smooth approximation of PD score distributions for non-defaulted and defaulted loans. Non-defaulted loans show a prominent peak near 0, indicating low predicted probabilities. Defaulted loans have a more spread-out density, with peaks around 0.1 to 0.2, reflecting the model's conservative nature in assigning PD scores. The overlap between the densities of the two classes (0.1 to 0.3) indicates areas of uncertainty, leading to potential false positives and false negatives. This overlap underscores the challenge in perfectly distinguishing between the two classes.

Calibration plots (Figures 3.25-3.29) reveal the models' calibration quality, comparing predicted probabilities to observed frequencies. All models

exhibit reasonable calibration up to 0.50 but show overconfidence or underconfidence beyond this point. The scarcity of high PD score observations likely causes this poor calibration at higher probabilities, as models lack sufficient data to make accurate high-risk predictions. Logistic Regression, due to its simplicity, remains well-calibrated at lower probabilities but struggles with more complex relationships at higher ranges. TabNet's sharp overconfidence and subsequent underconfidence likely results from its attention mechanism, which can disproportionately weigh certain features in high-risk cases.

Regarding precision, XGBoost stands out with the highest precision value of 0.48, underscoring its superior performance in correctly identifying positive instances across the range of thresholds. Logistic Regression and FNN both exhibit precision values of 0.36 and 0.34, respectively, indicating their comparable effectiveness in distinguishing between positive and negative classes, with Logistic Regression being slightly better. Random Forest achieves a precision of 0.39, reflecting its ability to maintain higher accuracy in positive instance identification despite its lower F1 Score. TabNet, with a precision of 0.31, demonstrates the lowest effectiveness in identifying positive instances among the evaluated models.

In conclusion, XGBoost stands out as the best-performing model, excelling in F1-score, Brier Score, and precision. Its superior ability to handle both linear and non-linear relationships, coupled with its robust gradient boosting framework, makes it highly effective for this task. Logistic Regression and FNN also perform well, providing reliable and consistent results. However, Random Forest and TabNet, while decent, show weaknesses, particularly in achieving a balanced precision-recall trade-off and accurate probability predictions.

4. Conclusion

This study aimed to estimate to what extent deep learning techniques and ensemble methods outperform logistic regression in calculating probability of default scores for credit loans. The experiments conducted in this research make a strong case for the application of XGBoost in credit risk modeling as it outperforms the other techniques in all evaluation metrics.

However, despite these promising results, it is crucial to incorporate XGBoost with caution due to its inherent complexity and "black box" nature. Unlike logistic regression, which offers clear insights into the relationships between predictors and outcomes, XGBoost's decisions are not as easily interpretable. This opacity can pose challenges in the context of credit risk modeling where model transparency, stakeholder trust and interpretability are essential.

This tradeoff between transparency and performance is in line with the No Free Lunch (NFL) theorem. The NFL theorem posits that no single model works best for every problem; rather, the effectiveness of a model depends on the specific nature of the data and the problem at hand. While XGBoost has shown superior performance in this particular study, there is no guarantee that it will consistently outperform other models in all scenarios. Each model, including FNN, RF, TabNet, and logistic regression, has its strengths and weaknesses, and their performance can vary significantly depending on the dataset. (Wolpert & Macready, 1997)

For future research, it is recommended to delve deeper into the opaque nature of XGBoost to improve its explainability for stakeholders and risk analysts who will be utilizing the PD scoring model. For BridgeFund, obtaining risk-drivers through feature importance is highly valuable. One such technique is the use of SHAP values, which provide a unified measure of feature importance. SHAP values are derived from cooperative game the-

ory and indicate how each feature contributes to the model's predictions. (Liu et al, 2022) By applying SHAP values to XGBoost, BridgeFund can pinpoint the influence of each variable on the classification of default , thereby enhancing transparency.

5. Bibliography

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, *23*(4), 589-609.

Aria, M., Cuccurullo, C., & Gnasso, A. (2021). A comparison among interpretative proposals for Random Forests. *Machine Learning with Applications*, *6*, 100094.

Arik, S. Ö., & Pfister, T. (2021, May). Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 8, pp. 6679-6687).

Basel Committee on Banking Supervision. (2006). *International convergence of capital measurement and capital standards: A revised framework comprehensive version*. Retrieved from `https://www.bis.org/publ/bcbs128.htm`

Basle Committee on Banking Supervision. (1996). *Amendment to the capital accord to incorporate market risks*. Bank for International Settlements.

Borisov, A., & others. (2021). Retrieved from `https://arxiv.org/abs/2110.01889`

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249-259.

Cathcart, L., Dufour, A., Rossi, L., & Varotto, S. (2020). The differential impact of leverage on the default risk of small and large firms. *Journal of Corporate Finance*, *60*, 101541.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, *95*(1), 24-37.

European Banking Authority. (2019). *Guidelines on the application of the definition of default*. Retrieved from https://www.eba.europa.eu/sites/default/documents/files/documents/10180/2427115/7c38f076-b8d4-4144-a2a4-c0dbf70a7b5b/Final%20report%20on%20Guidelines%20on%20default%20definition%20(EBA-GL-2019-07).pdf

Galindo, J., & Tamayo, P. (2000). Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications. *Computational Economics*, *15*, 107-143.

Gan, E., & Bailis, P. (2017, May). Scalable kernel density classification via threshold-based pruning. In *Proceedings of the 2017 ACM International Conference on Management of Data* (pp. 945-959).

Georgiou, K., & Yannacopoulos, A. N. (2024). Deep neural networks for probability of default modelling. *Journal of Industrial and Management Optimization*, 0-0.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Grau, J., Grosse, I., & Keilwagen, J. (2015). PRROC: Computing and visualizing precision-recall and receiver operating characteristic curves in R. *Bioinformatics*, *31*(15), 2595-2597.

Grinsztajn, S., & others. (2022). Retrieved from https://ar5iv.labs.arxiv.org/html/2207.08815

Hamori, S., Kawai, M., Kume, T., Murakami, Y., & Watanabe, C. (2018). Ensemble learning or deep learning? Application to default risk analysis. *Journal of Risk and Financial Management*, *11*(1), 12.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *160*(3), 523-541.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, *33*(1), 3-12.

Hooman, A., Marthandan, G., Yusoff, W. F. W., Omid, M., & Karamizadeh, S. (2016). Statistical and data mining methods in credit scoring. *The Journal of Developing Areas*, *50*(5), 371-381. Retrieved from `https://www.jstor.org/stable/26415601`

Jacob, J., & Gupta, P. (2005). Estimation of probability of default using Merton's option pricing approach: An empirical analysis. *[Publisher/Journal Name]*, *[Volume(Issue)]*, Pages.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429-449.

Jemai, J., & Zarrad, A. (2023). Feature selection engineering for credit risk assessment in retail banking. *Information*, *14*(3), 200.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)* (Vol. 14, No. 2, pp. 1137-1145).

Korangi, K., Mues, C., & Bravo, C. (2023). A transformer-based model for default prediction in mid-cap corporate markets. *European Journal of Operational Research*, *308*(1), 306-320.

Li, J., Liu, H., Yang, Z., & Han, L. (2021). A credit risk model with small sample data based on G-XGBoost. *Applied Artificial Intelligence*, *35*(15), 1550-1566. `https://doi.org/10.1080/08839514.2021.1987707`

Liu, Y., Liu, Z., Luo, X., & Zhao, H. (2022). Diagnosis of Parkinson's disease based on SHAP value feature selection. *Biocybernetics and Biomedical Engineering*, *42*(3), 856-869.

Louppe, G. (2014). Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*.

Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q., & Niu, X. (2018). Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGBoost algorithms according to different high dimensional data cleaning. *Electronic Commerce Research and Applications*, *31*, 24-39. https://doi.org/10.1016/j.elerap.2018.08.002

Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 141-183.

Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, 109-131.

Ramraj, S., Uzir, N., Sunil, R., & Banerjee, S. (2016). Experimenting XGBoost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, *9*(40), 651-662.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

Rudd, J. M. (2020). *An empirical study of downstream analysis effects of model pre-processing choices*. Retrieved from https://www.researchgate.net/figure/Random-forest-structure-15_fig3_346411178

Sang, A. (2021). Introduction to how a multilayer perceptron works but without complicated math. Retrieved from https://medium.com/codex/introduction-to-how-an-multilayer-perceptron-works-but-without-complicated-math-a423979897ac

Sariev, E., & Germano, G. (2020). Bayesian regularized artificial neural networks for the estimation of the probability of default. *Quantitative Finance*, *20*(2), 311-328.

Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). Credit scoring and its applications: SIAM monographs on mathematical modeling and computation. *Philadelphia: University City Science Center, SIAM*.

van den Goorbergh, R., van Smeden, M., Timmerman, D., & Van Calster, B. (2022). The harm of class imbalance corrections for risk prediction models: Illustration and simulation using logistic regression. *Journal of the*

American Medical Informatics Association, *29*(9), 1525-1534.

Wang, J., Wang, H., Nie, F., & Li, X. (2023). Feature selection with multi-class logistic regression. *Neurocomputing*, *543*, 126268.

Wang, W., Chakraborty, G., & Chakraborty, B. (2020). Predicting the risk of chronic kidney disease (CKD) using machine learning algorithm. *Applied Sciences*, *11*, 202. Retrieved from `https://www.researchgate.net/figure/Simplified-structure-of-XGBoost_fig2_348025909`

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67-82.

Yang, H. I., Yeh, M. L., Wong, G. L., Peng, C. Y., Chen, C. H., Trinh, H. N., ... & Nguyen, M. H. (2020). Real-world effectiveness from the Asia Pacific Rim Liver Consortium for HBV risk score for the prediction of hepatocellular carcinoma in chronic hepatitis B patients treated with oral antiviral therapy. *The Journal of Infectious Diseases*, *221*(3), 389-399.

Tocilins-Ruberts. (2022). Classification with TabNet: Deep Dive. Retrieved from `https://syslog.ravelin.com/classification-with-tabnet-deep-dive-49a0dcc8f7e8`

6. Appendix

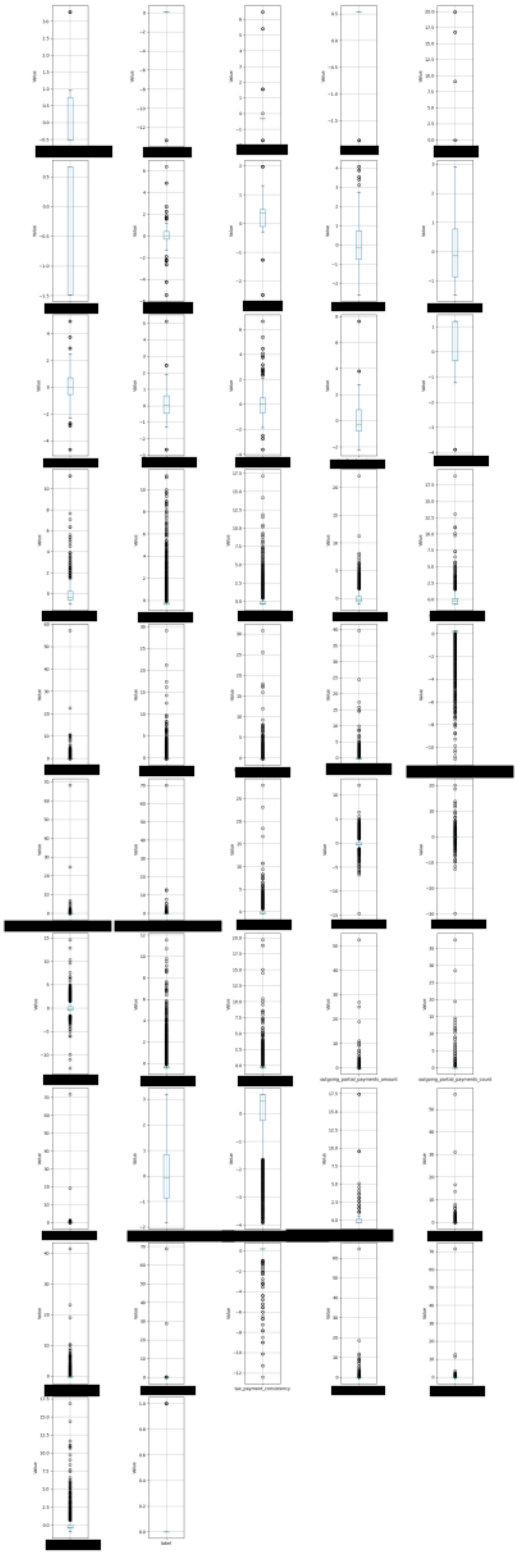

Figure 6.1: Boxplots

6.1 Adjustment for class imbalance with random undersampling

6.1.1 Logistic Regression

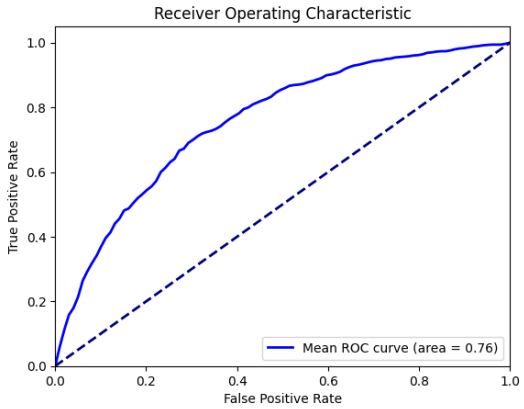

Figure 6.2: ROC for logistic regression with random undersampling

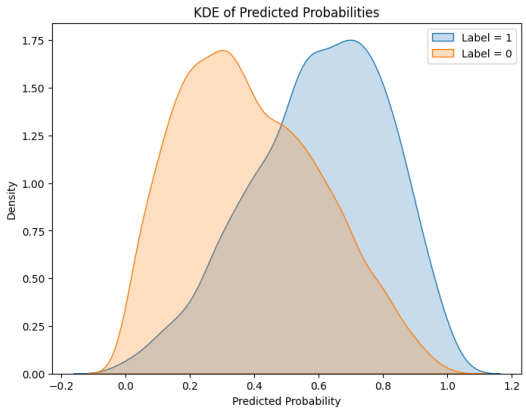

Figure 6.3: KDE for logistic regression with random undersampling

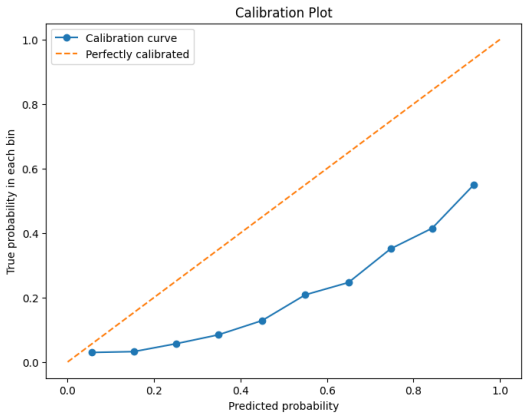

Figure 6.4: Calibration for logistic regression with random undersampling

6.1.2 XGBoost

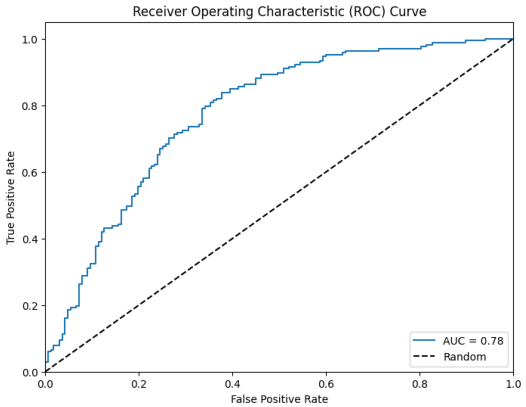

Figure 6.5: ROC for XGBoost with random undersampling

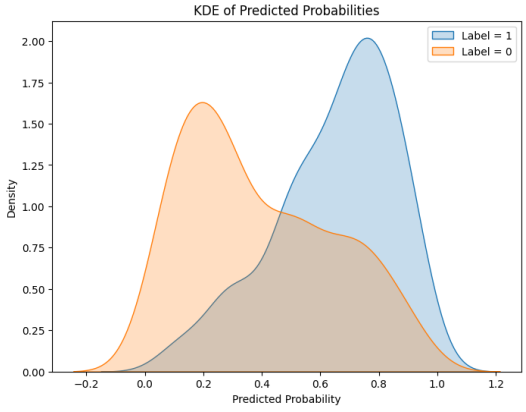

Figure 6.6: KDE for XGBoost with random undersampling

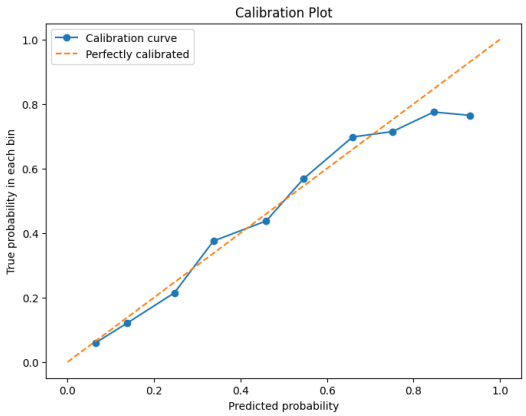

Figure 6.7: Calibration for XGBoost with random undersampling

6.1.3 Random Forest

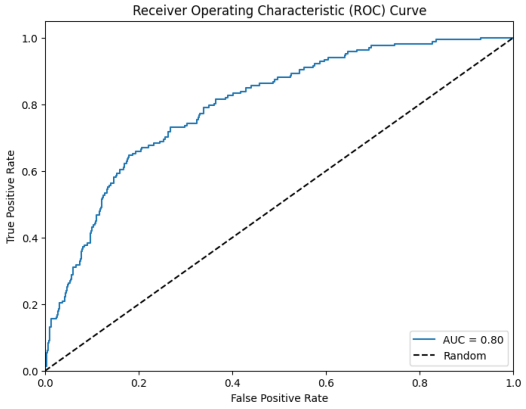

Figure 6.8: ROC for RF with undersampling

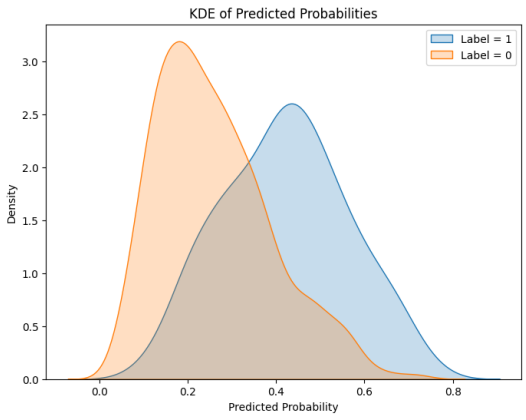

Figure 6.9: KDE for RF with undersampling

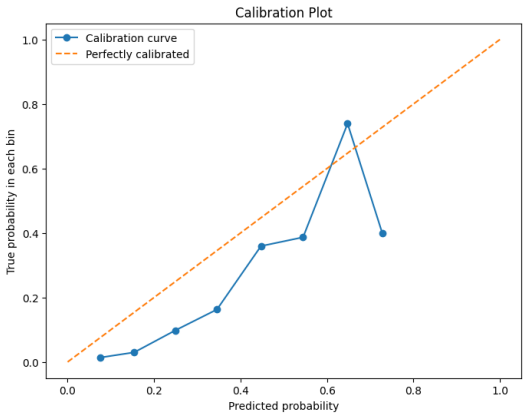

Figure 6.10: Calibration for RF with undersampling

6.1.4 FNN

6.1.5 TabNet

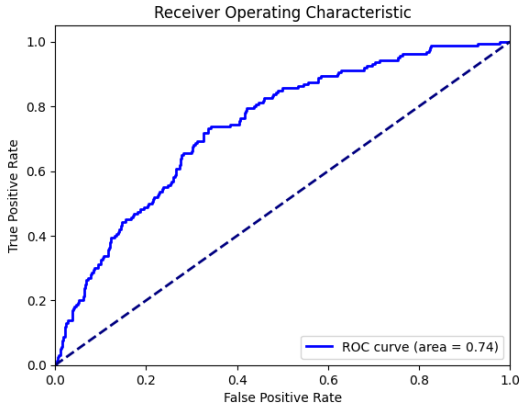

Figure 6.11: ROC for TabNet with random undersampling

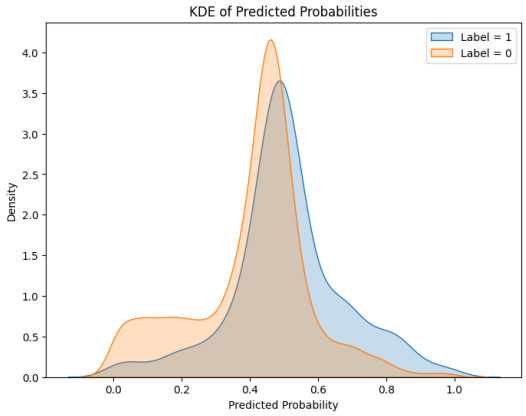

Figure 6.12: KDE for TabNet with random undersampling

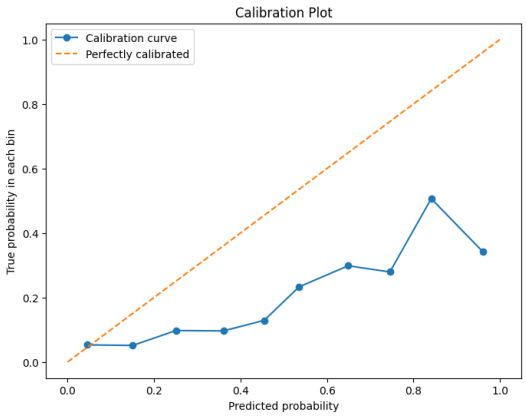

Figure 6.13: Calibration for TabNet with random undersampling

Feature	Coefficient	Standard Error	P-value	Significance
Feature 1	0.058	0.015	0.001	**
Feature 2	0.153	0.015	0.000	***
Feature 3	0.146	0.017	0.000	***
Feature 4	0.272	0.017	0.000	***
Feature 5	-0.020	0.015	0.298	
Feature 6	0.104	0.017	0.000	***
Feature 7	0.264	0.015	0.000	***
Feature 8	0.115	0.015	0.000	***
Feature 9	0.375	0.015	0.000	***
Feature 10	0.113	0.016	0.000	***
Feature 11	0.304	0.015	0.000	***
Feature 12	0.177	0.016	0.000	***
Feature 13	0.264	0.015	0.000	***
Feature 14	0.173	0.015	0.000	***
Feature 15	0.100	0.015	0.000	***
Feature 16	0.326	0.022	0.000	***
Feature 17	0.015	0.015	0.381	
Feature 18	0.076	0.016	0.001	**
Feature 19	-0.090	0.021	0.005	**
Feature 20	-0.422	0.018	0.000	***
Feature 21	0.050	0.022	0.121	
Feature 22	0.037	0.017	0.236	
Feature 23	0.052	0.015	0.004	**
Feature 24	0.095	0.018	0.000	***
Feature 25	-0.076	0.015	0.000	***
Feature 26	-0.078	0.042	0.306	
Feature 27	-0.153	0.028	0.036	*
Feature 28	-0.295	0.017	0.000	***
Feature 29	0.036	0.016	0.183	
Feature 30	0.093	0.016	0.093	
Feature 31	-0.504	0.018	0.000	***
Feature 32	0.050	0.017	0.064	
Feature 33	0.084	0.016	0.000	***
Feature 34	-0.113	0.034	0.154	
Feature 35	0.167	0.024	0.000	***
Feature 36	-0.030	0.073	0.211	
Feature 37	-0.106	0.017	0.000	***
Feature 38	-0.015	0.016	0.308	
Feature 39	0.023	0.017	0.216	
Feature 40	-0.016	0.020	0.449	
Feature 41	0.069	0.025	0.162	
Feature 42	-0.126	0.018	0.000	***
Feature 43	0.109	0.016	0.000	***
Feature 44	-0.221	0.017	0.000	***
Feature 45	-0.067	0.027	0.101	
Feature 46	-0.283	0.016	0.000	***

Table 6.1: Coefficients, Standard Errors, and P-values for Logistic Regression Model. Significance levels: * $p < 0.05$, ** $p < 0.005$, *** $p < 0.0005$