EUROPA-UNIVERSITÄT VIADRINA

RECEP GÖKTUĞ ŞENGÜN

128272

Maximizing Ensemble Model Performance by Introducing Diversity for Default Prediction

Master's Thesis for the Award of the Academic Degree Master of Science, Submitted to the Chair of Finance and Capital Market Theory

22. August 2024

Advisor: Dr. Rick Steinert

Degree Program: International Business Administration
Track: Data Science & Decision Support

Table of Contents

Abstract

This thesis investigates the potential for maximizing ensemble model performance by introducing diversity in the context of default prediction in financial markets. Default prediction is essential for financial institutions, as it enables more accurate assessment of credit risk and enhances the management of lending portfolios. Traditional models often fall short in capturing the complexity of financial data, underscoring the need for more advanced techniques. This research focuses on stacking ensembles, examining how diversity among the models can enhance predictive performance. While the study includes comparisons with widely used methods like Random Forest, XGBoost and AdaBoost. The primary emphasis is on evaluating the effectiveness of stacking ensembles and exploring the potential benefits of introducing diversity within these ensembles.

The thesis also addresses key challenges in default prediction, including data imbalance, missing values and high dimensionality while exploring how diverse stacking ensembles can help mitigating these issues. The findings suggest that incorporating strong individual models like XGBoost and Random Forest in diverse stacking ensembles can lead to significant improvements in predictive performance. This research aims to contribute to the field by demonstrating the value of model diversity in maximizing ensemble performance and provides insights for future work, particularly in refining diversity measures, handling data anomalies, and developing more comprehensive performance metrics for advance predictive modeling in financial risk management.

Keywords: Ensemble Models, Default Prediction, Financial Markets, Model Diversity, Binary Classification, Heterogenous Classifier Ensembles, Advanced Predictive Modeling, Financial Risk Management.

1 Introduction

1.1 Background and Motivation

In a globalizing world, with increased connectivity and transparency in economic activities, financial institutions face significant and highly complex challenges. One critical area is default prediction which is essential for institutions whose core business is lending. When making the lending decisions, it is vital to accurately assess consumer and corporate creditworthiness, to prevent these institutions from significant loses in the event of default. These institutions must predict the probability of default to control their losses. The information technology in this area involves collecting extensive data from borrowers and analyzing credit risk objectively by developing various statistical methods.

In default prediction area, common statistical methods such as logistic regression have been widely used. Although these techniques are simple to implement and interpret, they often fail in capturing complexity and nonlinearity that exists in financial data. These limitations suggest the need for more advanced approaches.

The advancements in technology and machine learning paved the way for more sophisticated algorithms in field of predictive modeling. Refined models like decision trees, neural networks and support vector machines have been developed and these models improved predictive capabilities. Despite their strengths, these individual models have weaknesses, therefore they are commonly known as weak learners. Weak learners are accepted as models that perform

slightly better than random guessing and they tend to overfit or underfit when used for dealing with complex datasets.
The limitations of weak learners show the necessity for methods that can leverage their strengths while reducing their weaknesses. This is where ensemble methods become relevant. Ensemble methods such as bagging, boosting and stacking combine the strengths of multiple weak learners to produce more solid and accurate predictions.
Ensemble models work on the principle that a group of weak learners can outperform any individual weak learner when combined appropriately. This inference stems from the diversity of the models. It relies on the assumption that different models capture different patterns of the data, thus leads to improved overall performance. As a result, the effectiveness of ensembles is largely dependent on the dissimilarity and compatibility of the base models.

1.2 Research Objectives

The primary aim of this thesis is to investigate the role of model diversity in enhancing the performance of ensemble models for default prediction. The specific objectives of this research are:

1. To analyze the importance of accurate default prediction in the context of financial markets, particularly in relation to the development and performance of ensemble models.
2. To identify and address challenges of default prediction that impact ensemble model performance, such as data imbalance, missing values and high dimensionality.
3. To provide a comprehensive overview of ensemble models, including an analysis of bagging, boosting, and stacking techniques and their application in default prediction.
4. To explore the concept of model dissimilarity within ensemble learning, and its role in enhancing predictive accuracy and robustness.
5. To empirically evaluate the impact of model diversity on the performance of ensemble models in default prediction.

1.3 Thesis Structure

The structure of this thesis is organized as follows:

Chapter 2 Significance and Application of Ensemble Models in Default Prediction: This chapter delves into the importance of default prediction in financial markets, the challenges and the role of ensemble models in addressing these challenges. It also provides an overview of different types of ensemble models and their current applications in default prediction, supplemented by a review of recent literature.

Chapter 3 Effectiveness of Dissimilar Models and Concepts of Ensemble Diversity: Here, the concept of model dissimilarity and its benefits in ensemble learning is presented. The chapter defines ensemble diversity, introduces techniques to enhance it and provides a theoretical framework for diverse models. Empirical evidence and case studies are also presented.

Chapter 4 Data Collection and Preprocessing Steps: This chapter describes the dataset used in the study including data collection methods and preprocessing techniques. It covers handling missing values and outliers, as well as feature selection and engineering processes to prepare the data for modeling.

Chapter 5 Experimental Design: An overview of the experimental setup is provided, detailing the benchmark models and ensembles, methods for constructing ensembles and the evaluation metrics used to measure performance. This chapter also explains the integration techniques for combining base models into an ensemble.

Chapter 6 Result: This chapter presents the performance results of individual and ensemble models with a focus on the effect of diversity on the performance of constructed heterogeneous ensembles. It includes a comparison of heterogeneous vs. homogeneous ensembles and analyzes the results using various performance measures.

Chapter 7 Discussion: The findings are interpreted and discussed in this chapter. The advantages and limitations of the proposed approach are examined and comparison with state-of-the-art methods are made. Potential directions for future research are also suggested.

Chapter 8 Conclusion: The thesis concludes with a summary of key findings, contributions to the field, and recommendations for practitioners. Directions for future work are provided.

2 Significance and Application of Ensemble Models in Default Prediction

Default prediction is a critical component of risk management in the financial industry, it is essential for the stability and success of financial institutions. Ensemble models, which combine multiple predictive algorithms, are acknowledged for their ability to improve prediction performance and resilience. This section addresses the significance of default prediction, the challenges involved and the effectiveness of ensemble models in overcoming these challenges. In this section, various types of ensemble methods will be explored and their application in default prediction will be examined with a review of recent studies in the field.

2.1 Importance of Default Prediction in Financial Markets

Default prediction is essential for financial institutions and investors. Banks use probabilities of default to evaluate potential borrowers, determine loan terms and manage risks associated with lending. Investors also utilize probabilities of default and credit rating changes for bond pricing and portfolio management. These institutions aim to reduce the incidence of bad debts and allocate capital more efficiently to ensure better financial health [1].

2.2 Challenges in Default Prediction

Traditional approaches for evaluating the risk of default typically depend on human judgement, leveraging the insights from past decisions. However, increased demand for credit, along with increasing commercial competition and the improvements of information technology, have led

to the development of more advanced statistical models to help the decision of granting credit [2].
Employing econometric methods such as logistic regression offers the advantage of interpretability, because the results are derived from a solid theoretical foundation. However, the assumptions these models are based on do not always align with real-world scenarios, as traditional models struggle to adequately capture the nonlinearity of financial data [3].

2.3 Overview of Ensemble Models

To address the drawbacks of traditional models, ensemble methods have emerged as a powerful alternative. The generalization capability of an ensemble is generally much stronger than that of its individual base learners which are sometimes called weak learners due to their initial performance level. Ensemble methods are particularly attractive because they can enhance weak learners which perform just slightly better than random guessing, into strong learners capable of making highly accurate predictions [4].

2.4 Role of Ensemble Models in Default Prediction

Ensemble models can significantly enhance the accuracy of predictions, making it particularly valuable in the context of default prediction. For instance, by building an ensemble of classifiers, the algorithm can aggregate and assign different weights to their votes, resulting minimized risk of making an incorrect prediction. Moreover, they are effective for capturing complexities and structural variabilities of financial data taking advantage of dissimilarity of base learners [5]. Hence, ensemble models play an important role in default prediction, which involves forecasting the likelihood that a borrower will fail to meet debt obligations. This prediction is vital for risk management of financial institutions.
As mentioned, ensemble methods significantly enhance predictive performance by aggregating the strengths and weaknesses of multiple classifiers, thereby balancing out individual model errors. This aggregation leads to more reliable predictions, which is particularly critical in the context of default prediction where misclassifications can result in major financial losses [6].

2.5 Types of Ensemble Models

2.5.1 Bagging

Bagging is a technique used to improve the performance of predictions by creating multiple versions of a single model and then combining them. This method is applied by randomly sampling with replacement to generate multiple training sets from the original data. These new sets are called bootstrap replicates. Each of these training sets is used to train the selected model. For numerical predictions, the final prediction is made by averaging the results of all the models. For classification problems, a majority vote is taken from separately trained models' predictions [6].

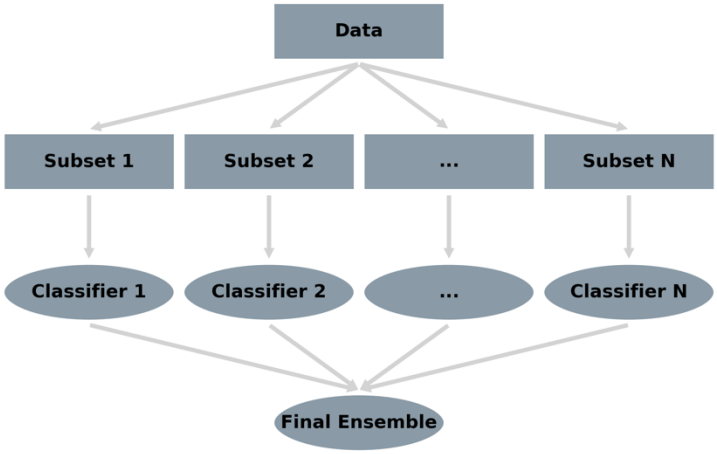

Figure 1: Bagging.

2.5.2 Boosting

Boosting is a type of sequential ensemble method. It includes a group of algorithms that can transform weak learners into strong learners. Boosting is similar to bagging, but while bagging trains multiple models in parallel, boosting trains models sequentially. In boosting, each new model focuses on correcting the errors made by the previous ones. This is achieved by emphasizing the misclassified data points in each iteration, allowing the model to improve on its weaknesses. The final prediction is made by combining the outputs of all models, usually using weighted voting. Boosting algorithm reduces both bias and variance which makes it effective for complex learning problems. Experiments demonstrate that the AdaBoost, which is a boosting algorithm, consistently outperforms bagging, especially when the base learners are simple models [7].

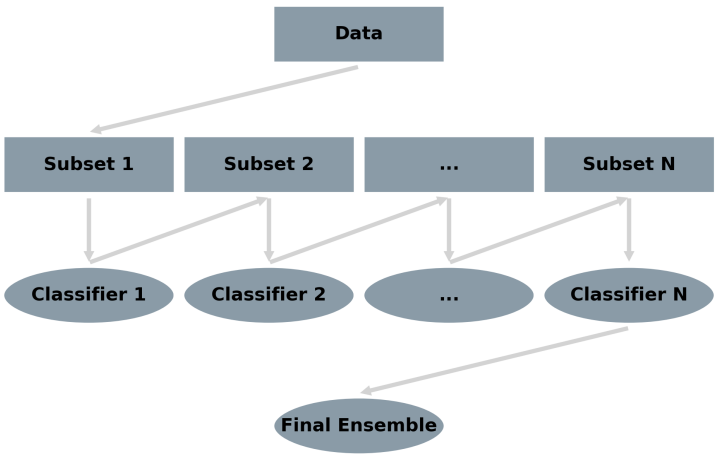

Figure 2: Boosting.

2.5.3 Stacking

Ensembles can be constructed either by combining the outputs of base models or by selecting the best performing one. Stacking technique is an integration method that makes use of a meta-learning model to combine the outputs of base models. The concept of stacking was first

introduced by David H. Wolpert. In this technique, the dataset is randomly split into equal parts. For cross-validation, one part is used for testing while the remaining parts are used for training. Using these training and testing subsets, predictions from different learning models are obtained and used as meta-data to build the meta-model. The meta-model then makes the final prediction utilizing the meta-data [8].

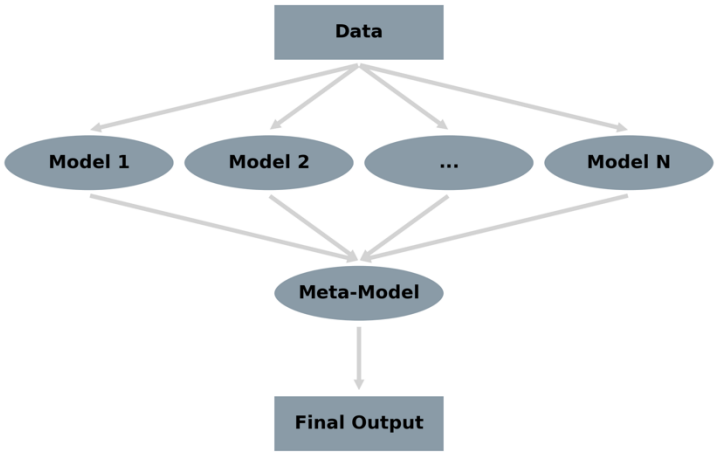

Figure 3: Stacking.

2.6 Current Applications in Default Prediction

Machine learning methods do not require assumptions or prior knowledge and can automatically extract useful information from datasets. Consequently, these methods have gained significant attention and widespread application in binary classification problems.
Tree based methods have been able to predict loan defaults and demonstrated much better performance and reliability than traditional models [9]. In addition to traditional approaches for predicting user defaults such as Logistic Regression, Decision Trees and Support Vector Machines are extensively used in default prediction [10].
For instance, Adaboost, introduced by Freund & Schapire is a popular algorithm that deterministically adjusts the weights of training samples for the next classifier based on the errors of previous classifiers.

2.7 Review of Recent Literature

Recent research highlights the effectiveness of ensemble learning algorithms in default prediction. Bifet et al. [11] presented bag ensembles composed of restricted Hoeffding Trees that were shown to outperform the best results of default prediction in the literature.
Singh and Sivasankar [12] highlighted the use of ensemble methods to improve the accuracy of credit default predictions. Their research concluded that ensemble techniques, such as Bagging and Boosting, significantly enhance prediction accuracy over single models.
Hamori et al. [13] evaluated the effectiveness of ensemble learning algorithms versus deep learning methods for predicting default risk. Their findings indicated that ensemble learning techniques, especially Random Forests and Gradient Boosting, outperformed deep learning models in default risk analysis.
The study by Li et al. [14] explored heterogeneous ensemble learning combined with feature engineering for default prediction in China's peer-to-peer lending market. They demonstrated

that heterogeneous ensembles, combined with feature engineering, provided superior performance in default prediction compared to homogeneous models.
Lu et al. [15] found that an adjusted heterogeneous ensemble approach significantly enhanced prediction accuracy and robustness. Similarly, Kun et al. [16] compared to other mainstream machine learning algorithms with stacking and showed that it generally achieves better performance than a single classifier. They found that even though XGB was the best-performing individual model and when it was used as a base model in stacking, the overall ensemble accuracy was higher than that of XGB alone.

3 Effectiveness of Dissimilar Models and Concepts of Ensemble Diversity

3.1 Concept of Model Dissimilarity

Classifiers in an ensemble must be different to provide benefits. This has been identified as a key research area, with measures derived from statistical literature. Model dissimilarity refers to the extent of variation between the models within an ensemble. This variation can arise from using different algorithms, distinct subsets of training data, or varied parameter settings [17]. Diverse models make different errors and improve overall prediction accuracy.

3.2 Benefits of Dissimilar Models in Ensembles

If there was a classifier that never makes mistakes, there would be no need for an ensemble. However, since classifiers often do make errors, the aim is to complement them with another classifier that makes errors on different instances. Therefore, the diversity of classifier outputs is important for an ensemble's success. Ideally, the members of an ensemble members are preferred to be as accurate as possible and when they do make mistakes, those mistakes should be on different instances [18].
Using dissimilar models in an ensemble enhances predictive performance by reducing the likelihood that all models will make the same errors. This diversity leads to more resilient and accurate predictions as errors made by some models can be compensated by others [19]. Moreover, dissimilar models help to capture different aspects of the data and providing a more comprehensive understanding and better generalization. An ensemble of three identical classifiers will make the same errors. However, if their errors are uncorrelated, when one makes an error, the others may still be correct, allowing a majority vote to classify accurately. This reduces the overall error rate of the ensemble compared to individual classifiers [5].

3.3 Case Studies and Examples

Based on the work of Dietterich [5], it is widely accepted that the diversity of individual members in an ensemble enhances model performance. Hansen and Salamon [19] demonstrated that training different neural network models as base learners in an ensemble improves performance, even if two of the three models are not well trained. Opitz and Maclin [20] found that boosting outperforms bagging in nearly every test. Ahmet et al. [21] used diversity measures as regulators for ensemble pruning, showing that this approach can

enhance the overall predictive power and generalization of a classifier ensemble. Li et al. [22] proposed a multi-round heterogeneous ensemble model comprising XGBoost, Deep Neural Networks, and Logistic Regression, which demonstrated improved predictive accuracy on imbalanced data.

3.4 Defining Ensemble Diversity

Ensemble diversity refers to the degree of variability among the predictions made by individual weak learners within an ensemble. High diversity indicates that the models make different errors on the same data points, whereas low diversity suggests that the models make similar errors. Diversity is important when constructing ensembles because it helps to ensure that the combined model benefits from the strengths of each individual model which leads to reduced overall error rate [5] [18]. Methods to enhance diversity include using different algorithms, varying training data subsets or applying distinct hyperparameters. The key is to balance accuracy and diversity to optimize ensemble performance [23].

3.5 Measuring Model Dissimilarity

Defining a single measure of diversity is challenging in practice and establishing a clear and meaningful link between that measure and ensemble performance is even more complex. Valuable insights often come from cross-disciplinary ideas and approaches. Model dissimilarity can be quantified using various metrics, such as pairwise correlation measures like the Q-statistic, correlation coefficient, disagreement measure, and double-fault measure [24]. These metrics assess the degree of variation in errors made by different models, providing insights into the diversity between model pairs. In addition, non-pairwise methods such as entropy and generalized can measure diversity across the entire ensemble.

3.5.1 Q-Statistic:

Consider a labeled dataset $Z = \{ z_1, z_2, \ldots, z_N \}$, where each instance z_j belongs to a classification problem. The output of a classifier D_i can be represented as an N-dimensional binary vector: $Y_i = [y_{1,i}, \ldots, y_{N,i}]^T$

where $y_{j,i} = 1$ if classifier D_i correctly classifies instance z_j and $y_{j,i} = 0$ otherwise.

The relationship between two classifiers D_i and D_k is summarized in the table:

	$D_k\ correct\ (1)$	$D_k\ wrong\ (0)$
$D_i\ correct\ (1)$	N^{11}	N^{10}
$D_i\ wrong\ (0)$	N^{01}	N^{00}

Q statistic for two classifiers D_i and D_k is calculated as:

$$Q_{i,k} = \frac{N^{11}\,N^{00} \;-\; N^{01}\,N^{10}}{N^{11}\,N^{00} \;+\; N^{01}\,N^{10}}$$

Lower Q-statistic values (closer to -1) indicate more diversity. A value of 0 indicates independence, while higher values (closer to 1) indicate less diversity [24].

3.5.2 Correlation Coefficient:

The correlation coefficient (ρ) measures the correlation between the outputs of two binary classifiers, y_i and y_k, and is calculated as follows:

$$\rho_{i,k} = \frac{N^{11}\,N^{00} \;-\; N^{01}\,N^{10}}{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}$$

For any 2 classifiers ρ and Q indicate the same sign and $|\rho| \leq |Q|$ [24].

3.5.3 The Disagreement Measure:

This measure characterizes the diversity between a base classifier and a complementary classifier by measuring the ratio of instances where one classifier is correct and the other is incorrect to the total number of instances.

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}$$

Higher disagreement measure values indicate more diversity. A higher value means that the classifiers are making different predictions more often [24].

3.5.4 The Double Fault Measure:

It is defined as the fraction of instances that both classifiers incorrectly classify.

$$Dis_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}}$$

Lower double-fault measure values indicate more diversity. A lower value means that both classifiers are not making errors on the same instances frequently [24].

3.5.5 The Entropy Measure:

Entropy measure is a non-pairwise diversity measure suggested by Kuncheva et al. [24]. According to the suggestion the greatest diversity among classifiers for a specific $z_j \in Z$ occurs when half of the votes in Z_j are one value (either 0 or 1), and the other half are the opposite value. If all votes are 0's or all are 1's, there is no disagreement, the classifiers cannot

be considered diverse. Let $l(z_j)$ represent the number of classifiers from D that correctly identify z_j, defined as $l(z_j) = \sum_{i=1}^{L} y_{j,i}$. With higher entropy values indicate more diversity, possible measure of diversity based on this concept is as follows:

$$E = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{(L - \lceil L/2 \rceil)}\min\{l(z_j), L - l(z_j)\}$$

3.5.6 Generalized Diversity:

Another non-pairwise diversity measure is Generalized Diversity. In their work, Partridge & Krzanowski [25] suggested the probabilistic approach for measuring diversity by focusing on failure patterns among classifiers. Based on this work, Kuncheva et al. [24] develops generalized diversity and it can be calculated as follows:

$$p(1) = \sum_{i=1}^{L}\frac{i}{L}p_i \quad and \quad p(2) = \sum_{i=1}^{L}\frac{i(i-1)}{L(L-1)}p_i$$

$$GD = 1 - \frac{p(2)}{p(1)}$$

Where:

- L represents the number of classifiers in an ensemble
- $p(i)$ is the probability of i classifiers failing out of L.

A higher GD value indicates greater diversity, meaning that the classifiers are making different errors, thus they are more likely to complement each other and improve the overall ensemble performance.

3.6 Techniques to Enhance Diversity in Ensembles

Beyond Bagging, Boosting and Stacking there are various methods to enhance diversity in ensembles. For instance, the random subspace method enhances diversity by training each model on a random subset of the features [26]. Additionally, training models with different hyperparameter settings can significantly enhance diversity. Different hyperparameters can lead to models that make different predictions, even if they are based on the same algorithm. Furthermore, combining models from different algorithmic families, such as decision trees, support vector machines, and neural networks, ensures diversity since these algorithms have different biases and learning patterns. This approach leverages the strengths of various algorithms to improve the ensemble's performance [5].

Moreover, introducing randomness into the learning process can create diverse models. For example, using stochastic gradient descent (SGD) with different random seeds or injecting noise into the training data can lead to diverse model behaviors [9]. Similarly, using cross-validation to train multiple models on different subsets of the data enhances diversity [27]. Finally, negative correlation learning encourages diversity by penalizing models that make correlated errors. This technique ensures that models learn different aspects of the data, further enhancing the ensemble's diversity [28].

3.7 Theoretical Framework

In ensemble techniques, an aggregate of machine learning algorithms is used to derive better predictive performance than could be obtained from individual algorithms separately. These models achieve better accuracy, especially when there is significant diversity among the ensemble models [12].

The bias-variance trade-off is essential for understanding ensemble methods. Ensemble methods like bagging [5], aim to reduce variance without significantly increasing bias. By averaging predictions from multiple models, ensembles reduce overfitting and improve generalization.

3.8 Bias-Variance Decomposition

In classification, the error can be decomposed into bias, variance, and noise. Ensembles aim to reduce the variance component without significantly increasing bias. This can be represented as:

$$Total\ Error = Bias^2 + Variance + Noise$$

$$Bias^2 = E[\hat{f}(x)] - f(x)$$

Where:

- $\hat{f}(x)$ is the model's predicted probability for a specific input x.
- $E[\hat{f}(x)]$ is the expected prediction across different training datasets.
- $f(x)$ is the true probability of the positive class.

$$Variance = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

Where:

- $\hat{f}(x)$ is the predicted probability of the positive class (i.e., $P(y = 1 \mid x)$ for a given input x.
- $E[\hat{f}(x)]$ is the expected predicted probability over different training datasets. [29]

3.9 Empirical Evidence

Empirical studies validate the effectiveness of bagging in reducing variance and enhancing accuracy in binary classification models. Breiman's foundational work showed that bagging significantly reduces variance by averaging predictions from multiple models trained on different bootstrap samples, benefiting high-variance models like decision trees [9]. Dietterich confirmed these results, demonstrating that bagging consistently improves accuracy across various classifiers and datasets [5]. Opitz and Maclin further highlighted the effect of bagging in reducing variance [20]. Additionally, Bauer and Kohavi's analysis of bias-variance decomposition provided empirical evidence that bagging effectively reduces variance while maintaining or slightly increasing bias, leading to improved performance in binary classification tasks [29].

4 Data Collection and Preprocessing Steps

4.1 Description of the Dataset

The dataset utilized in this study was sourced from *Bondora*. The original dataset includes *338,814* observations and 112 variables, covering the period from February 28, 2009 to November 8, 2023.
Bondora.com, a leading non-bank digital lender in Continental Europe since 2008, offers consumer loans in Finland, Spain, and Estonia through a fully digital platform powered by advanced credit analytics. These loans are financed by selling receivables to a retail investor base spanning 40 countries worldwide. The dataset provided from *Bondora.com* contains detailed information about loans, borrowers, and repayment statuses, providing a rich source of data for analyzing lending patterns, default rates, and other financial metrics.

4.2 Data Quality

Steps Taken to Ensure Data Quality and Integrity:

- **Duplicate Entries**: The dataset was checked for duplicate entries to ensure that each record was unique.
- **Target Variable Extraction**: The *DefaultDate* column was used to extract the target variable. If a date was present, it indicated that the loan had defaulted. However, some loans marked as defaulted were shown as repaid in the *Status* column, causing confusion. To resolve this, entries where the *Status* was 'repaid' and a *DefaultDate* was present were removed. The left entries were only the ones where the *Status* was 'current' or 'late'. The target variable *Default* was then created, with 1 indicating default and 0 indicating non-default. The *DefaultDate* and *Status* columns were subsequently removed.

4.3 Preprocessing Techniques

Specific Data Cleaning Steps Taken: Columns with more than 80% missing values were removed. These included *DateOfBirth*, *County*, *City* which have 100% missing values and *EmploymentPosition, NrOfDependants, WorkExperience, PlannedPrincipalTillDate, EL_V0, Rating_V0, EL_V1, Rating_V1, Rating_V2, CreditScoreEsEquifaxRisk* and *PreviousEarlyRepaymentsBeforeLoan*.[1]

Data Transformation Methods Used: To understand data types, unique values in each column were analyzed. The final aim was to convert all data to numerical format to avoid errors during model implementation. Data was separated according to their types:

- **Categorical Variables**: Categorical variables were handled separately. Nominal categories were converted to numerical values using mapping techniques, while ordinal categories were also mapped but according to their inherent ranks.

[1] For a visual representation of the columns with high missing values see *Figure A.1.* in Appendix.

- **Datetime Columns**: Datetime values were transformed into durations and binary variables indicating occurrence to make them interpretable by models.
- **Numeric Columns**: All numeric columns were converted to floats, and their logarithmic transformations were applied.

4.3.1 Handling Missing Values and Outliers

Methods for Detecting and Handling Missing Data:

Missing data was handled based on the different data types:

- **Datetime Features**: Missing and infinite values were replaced with 0 to indicate the event did not occur.
- **Numeric Values**: Missing numeric values were imputed using the median to minimize the influence of outliers. The median was chosen due to the data distribution and the selected approach for handling outliers.
- **Categorical Variables**: Missing values in categorical variables were replaced with the mode.
- **Boolean Values**: Converted to 1 and 0, with missing values replaced with the mode.

Techniques for Identifying and Treating Outliers:

The Tukey IQR (Interquartile Range) method was used to detect and handle outliers. This method identifies outliers as values lying below $Q_1 - 1.5\ x\ IQR$ or above $Q_3 + 1.5\ x\ IQR$,

Where:

- $Q_1 = 25th\ percentile\ of\ the\ data$
- $Q_3 = 75th\ percentile\ of\ the\ data$
- $IQR = Q_3 - Q_1$

Outliers were capped to these limits to retain valuable information while minimizing their impact.[2]

4.3.2 Feature Selection and Engineering

Criteria for Selecting Features

Initially, columns containing unique IDs were removed, as they did not offer meaningful information for the analysis. No other features were excluded based solely on their names or definitions; instead, statistical methods were employed to assess the relevance of the remaining features.

[2] See *Figure A.2.* in the Appendix for the distribution of data after outlier capping.

Methods for Creating New Features

Datetime Conversion: To enable arithmetic operations on datetime columns, all relevant datetime fields were first converted to the datetime format. This conversion was essential for accurate duration calculations and to leverage the datetime features effectively. Thereafter, the following calculations were made, and the corresponding features were created:

Duration Calculations: Several durations were calculated to provide insights into various stages and periods related to loans. Each duration was expressed in months to standardize the timeframes.

- **Loan Application to Approval Duration:**
  - *Calculation:* The duration between *LoanApplicationStartedDate* and *LoanDate*.
  - Purpose: To understand the time required for loan applications to be approved.

- **Grace Period Duration:**
  - Calculation: The duration from *GracePeriodStart* to *GracePeriodEnd*.
  - Purpose: To measure the length of the grace period provided to borrowers.

- **Current Stage Duration:**
  - **Calculation:** The time a loan has been in its current stage, from *StageActiveSince* to *ReportAsOfEOD*.
  - **Purpose:** To determine the duration a loan remains in its present status.

- **Listed to Approval Duration:**
  - **Calculation:** The duration from *ListedOnUTC* to *LoanDate*.
  - **Purpose:** To measure the time taken for a loan to be approved after being listed.

- **Original Loan Term Duration:**
  - **Calculation:** The original loan term, from *LoanDate* to *MaturityDate_Original*.
  - **Purpose:** To understand the initially agreed-upon duration of the loan.

- **Bidding to Loan Conversion Duration:**
  - **Calculation:** The time taken for bidding to convert into a loan, from *BiddingStartedOn* to *LoanDate*.
  - **Purpose:** To evaluate the efficiency of the bidding process in converting bids into loans.

Event Occurrences: Tracking specific events was essential for understanding changes in the loan status. To achieve this, binary indicators were created for the following events by analyzing missing values in the dataset.

- **Principal Debt Occurrence:**
  - **Indicator:** *DebtOccurred*
  - **Purpose:** To indicate when the principal amount of the loan was confirmed as a debt.

- **Secondary Debt Occurrence:**
  - **Indicator:** *DebtOccurredForSecondary*
  - **Purpose:** To track the occurrence of additional debts, such as interest or other secondary liabilities.

- **Loan Rescheduling:**
  - **Indicator:** *ReScheduled*
  - **Purpose:** To track whether a loan had been rescheduled, indicating adjustments to the repayment terms.

Seasonality Effect: To understand the potential impact of seasonality on loan events, the month in which bidding started and the loan date were extracted and converted to numeric formats:

- **Bidding Start Month:**
  - **Calculation:** The month when the bidding process for a loan began was extracted from the *BiddingStartedOn* date and converted into a numeric format.
  - **Purpose:** To assess whether the timing of bidding start exhibits any seasonal patterns affecting loan approval or success rates.

- **Loan Date Month:**
  - **Calculation:** The month of the actual loan date was extracted from the *LoanDate* and converted into a numeric format.
  - **Purpose:** To determine if there are seasonal trends in loan issuance that could influence loan outcomes.

Feature Selection and Engineering

Given the large dataset, consisting of 306,268 rows and 86 columns, processing it was computationally expensive. Initially, to address the class imbalance within the dataset, a Random Under Sampler method was applied, which reduced the majority class to balance it with the minority class. Following this, a 20% sample of the resampled data was used in feature selection algorithms to streamline the dataset by identifying the most relevant features. In the feature selection and engineering process, irrelevant columns such as unique IDs and the datetime features which are not used for the presented calculations were first removed to streamline the dataset. The remaining features were then evaluated using statistical methods:

- **Categorical Features:** These features were initially assessed using the Chi-square test to determine their significance in relation to the target variable. Chi-square limits were set between 3.84 and 1000 to prevent overfitting. Subsequently, Mutual Information was applied to retain only the most informative features. For this, limits were established between 0.01 and 0.3, and features with values within this interval were kept.

- **Numerical Features:** Correlations with the target variable were analyzed, and highly correlated features were eliminated to prevent overfitting. Features with correlation values below 0.05 were excluded as they were considered too weakly correlated to be meaningful. Conversely, features with correlation values above 0.7 were removed due to their strong correlation and potential overfitting.[3]

To ensure feature independence, the Variance Inflation Factor (VIF) was calculated and features with high multicollinearity were iteratively removed until all remaining features met the threshold criteria which was set to 5. Finally, Recursive Feature Elimination with Cross-Validation (RFECV) was applied, leading to the selection of nine final features for model development: *HomeOwnershipType*, *LanguageCode*, *CreditScoreEsMicroL*, *Gender*, *Restructured*[4], *PrincipalBalance*, *BidsManual*, *LossGivenDefault*, and *NoOfPreviousLoansBeforeLoan*.[5]

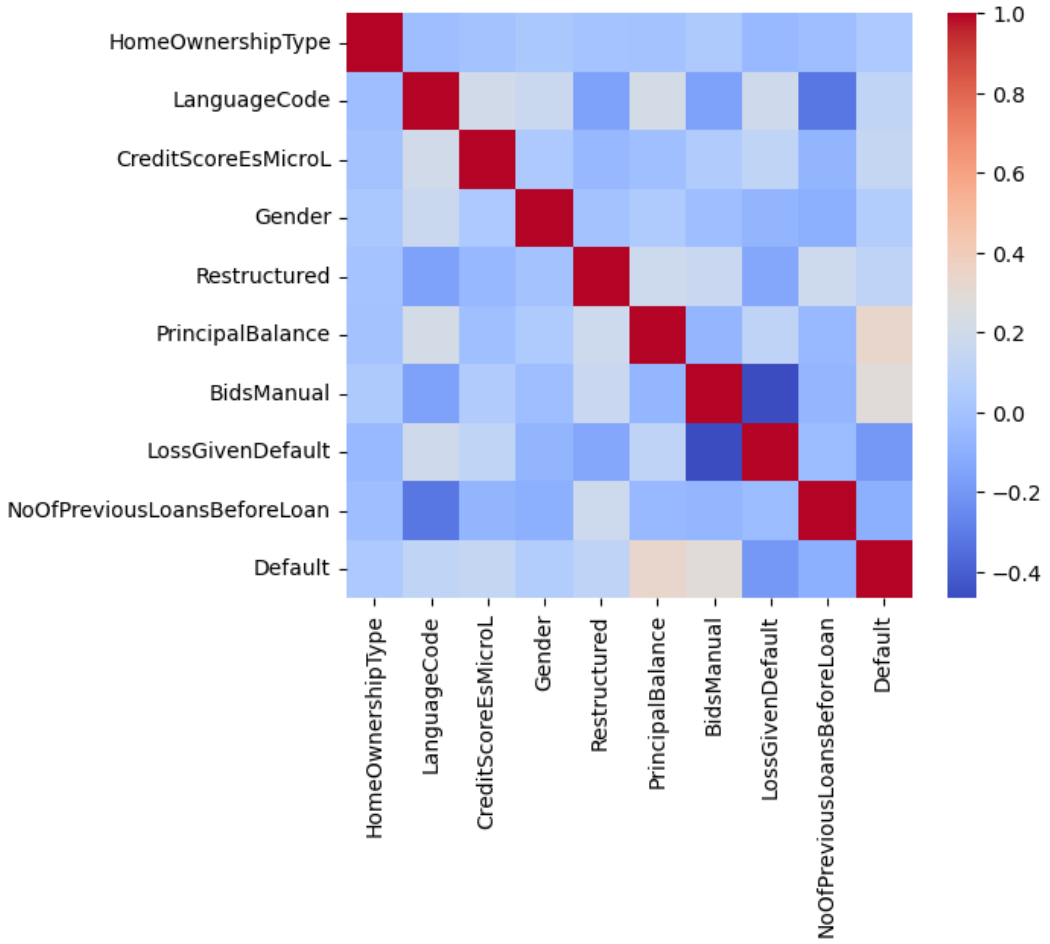

Figure 4: Correlation Heatmap of Final Features After Feature Selection.

[3] See *Figure A.3.* in the Appendix for the heatmap showing correlations after the removal of weakly and highly correlated numeric features.
[4] *Restructured* is a feature created by addressing missing values from the dataset.
[5] See *Figure A.4.* in the Appendix for the RFECV plot illustrating the selection of features based on cross-validation F1-scores.

5 Experimental Design

5.1 Overview of Experimental Setup

The study is focused on improving the predictive performance of binary classification in default prediction tasks by utilizing ensemble methods that leverage the diversity of several machine learning algorithms. These algorithms, belonging to different families with varying prediction formulas, often provide weak default predictions when used individually. However, when combined in ensembles, their diverse strengths contribute to enhanced overall prediction performance.

The main objective of this experiment is to compare the performance of several weak learners, measure their diversity and construct ensembles. The constructed ensembles will then be evaluated based on their diversity and standard performance metrics accuracy, precision, recall, F1-score, log loss, and ROC-AUC. The study highlights the effectiveness of diverse ensemble methods, showing how they can outperform individual weak learners, emphasizing the importance of diversity in improving predictive performance in default prediction tasks.

This experimental work was conducted using loan data from Bondora.com. The original dataset contained many features, from which the most relevant ones were selected based on their relationship with the target variable and their predictive power. Due to computational constraints, nine of the most relevant features were chosen for the default prediction tasks. These features were used in processes such as hyperparameter tuning, individual model predictions, ensemble construction, and validation.

Various weak learners, including logistic regression, k-nearest neighbors, and support vector machines were employed. The pairwise diversity of these models was measured using metrics such as correlation, disagreement, and double-fault measures.[6] However, the ensembles are randomly constructed, and their diversity is measured using non pairwise diversity metrics such as entropy measure and generalized diversity. The performances of the constructed ensemble models over individual weak learners underscore the critical role of diversity in achieving better predictive outcomes.

5.2 Evaluation Metrics and Criteria

To evaluate the effectiveness of both individual and ensemble models, various performance metrics are employed. This section of the paper will briefly explain the performance measures and their corresponding formulas.[7]

Accuracy: Accuracy is the percentage of classifications that are correct. The measure is calculated as follows.

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

[6] See *Table A.1* for pairwise diversity values.
[7] The definitions and formulas are sourced from [32].

Precision: Precision, also known as positive predictive value, is the ratio of correctly predicted positive cases to the total number of cases that were predicted as positive.

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

Recall: Recall or the true positive rate (TPR), is the ratio of correctly predicted positive cases to the total number of actual positive cases.

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

F1-Score: The F-measure, also known as the F1-score, is the harmonic mean of precision and recall.

$$F1\ Score = 2 \times \frac{1}{\frac{1}{precision} + \frac{1}{recall}} = 2 \times \frac{precision \times recall}{precision + recall}$$

Cross Entropy (Log Loss): For a set of n instances, where each instance x_i has a true label y_i and a predicted probability p_i of belonging to the positive class, the log loss is calculated as follows:

$$log\ loss = -\frac{1}{N}\sum_{i=1}^{n} y_i log_2(p_i) + (1 - y_i) log_2(1 - p_i)$$

ROC and AUC: The AUC is formally defined as the area under the curve that plots the true positive rate (TPR) against the false positive rate (FPR) across all possible thresholds t_i.

$$AUC = P(p_+ > p_- \mid x_+ \ and \ x_-) = \int_0^1 TPR(t_i) dFPR(t_i)$$

5.3 Benchmark Models and Ensembles

Logistic Regression

Logistic regression is a useful statistical method for analyzing outcomes with two possible results, like yes/no or success/failure. It includes both continuous and categorical predictors while adjusting for multiple factors. Unlike linear regression, logistic regression predicts the probability of an event happening by modeling the log of odds. This makes it reliable for providing accurate estimates. This method is widely used in fields where predicting the likelihood of an event is essential.

Support Vector Machines
SVM aims to create the best possible separation between classes by maximizing the margin around the separating hyperplane. It effectively handles separable and non-separable cases by using slack variables and solving a convex optimization problem. The support vectors play a crucial role in defining the decision boundary and the method ensures that the model is robust and generalizes well to new data [30].

K-Nearest Neighbours
KNN is a non-parametric, supervised learning classifier that uses distances in predictions of grouping an individual data point. One of the popular and simplest machine learning classification and regression classifiers used today is k-nearest neighbors. Though the KNN algorithm can be used for either regression or classification, it usually is implemented as a classification algorithm because of the assumption that similar points should align close to each other [30].

Neural Networks
Neural networks often work well for binary classification. In such a setup, layers of inputs come in through an input layer and go to one or more hidden layers. At the top, there is only one output unit that can represent the probability of one of two classes. The target variable is binary valued, as 0 or 1. The input features are combined linearly to create derived features, which are then used to model the probability through a non-linear activation function like the sigmoid function. This architecture allows the learning of complex, nonlinear relationships between the inputs and a binary target, enabling effective classification of data into two distinct categories [30].

Naive Bayes
The Naive Bayes classifier assumes that the features are independent given the class and treats the joint probability of the features as the product of the individual probabilities, significantly simplifying the computation [30].

Stochastic Gradient Descent
SGD is an optimization algorithm used to minimize a model's loss function by updating its parameters iteratively. Unlike traditional gradient descent, which uses the entire dataset to compute gradients, SGD updates the model using only one data point at a time, making it faster and more suitable for large datasets.

Linear Discriminant Analysis
LDA is a statistical method used for classification and dimensionality reduction. It is particularly effective when the classes are linearly separable, meaning that the decision boundary between classes can be represented by a straight line. LDA assumes that each class k has its own probability density function $f_k(x)$, which represents how likely it is for a given observation x to belong to that class [30].

Random Forests
Random forests are an ensemble learning method that combines multiple decision tree predictors, where each tree is built using a random subset of features. This random selection of features at each split reduces the correlation between trees, leading to better models. The

generalization error of a random forest converges to a limit as the number of trees increases, and it depends on the strength of the individual trees and their correlations. Additionally, internal estimates within the model can monitor error rates, tree strength, correlations and measure the importance of variables, making this approach applicable to both classification and regression tasks [6].

AdaBoost
AdaBoost, or Adaptive Boosting, is an ensemble learning algorithm introduced by Freund and Schapire in 1996, designed to enhance the performance of weak classifiers. It operates by sequentially applying a weak classification algorithm to multiple modified versions of the training data, each time adjusting the weights of the training samples based on the accuracy of the previous classifier. Misclassified samples receive higher weights, forcing subsequent classifiers to focus on these harder cases. The final prediction is made by combining the outputs of all weak classifiers through a weighted majority vote. This method effectively reduces bias and variance, producing a strong classifier that outperforms individual weak classifiers [30].

XGBoost
XGBoost (eXtreme Gradient Boosting) is a scalable machine learning algorithm based on gradient boosting and constructed for performance and speed. In conventional gradient boosting, one constructs an additive model in a forward stepwise manner by minimizing the loss function to the existing model; XGBoost changes the method. Only decision trees are modeled as the base learners and a new form of the loss function is introduced that comprises training loss and a regularization term. This regularization term penalizes the complexity of a tree so that it will not overfit, based on the number of leaves and the output scores derived from those leaves [31].

5.4 Methods for Constructing Ensembles

5.4.1 Selection of Base Models

For this task, a cluster of 10 binary classification models was used, with each model trained on a 20% sample of the dataset. To address class imbalance during hyperparameter tuning, a random under sampling technique was employed. Random search and grid search were applied to optimize each model's performance. Once tuning was complete, the dataset was reverted to its original imbalanced form, allowing each model to handle class imbalance according to its own strengths, acknowledging that some models are better at managing imbalance than others.
The predicted values from each model were used to calculate performance metrics such as accuracy, log loss, and F-1 score. To assess and enhance model diversity, initially pairwise diversity measures were calculated, providing insight into the diversity between two models. However, determining the third model in a group proved challenging, particularly as the study aimed to form groups of 3, 5, and 7 models. Consequently, non-pairwise diversity measures, such as entropy and general diversity were employed to evaluate all possible combinations of models in these groups. Ultimately, entropy was selected as the primary measure of diversity.

5.4.2 Integration Techniques

To effectively combine the base models and enhance predictive performance, the stacking ensemble method was employed. Stacking allows multiple models to contribute to the final prediction by leveraging a meta-learner to combine their outputs.

Stacking Approach

- **Meta-Learner**: A Logistic Regression model was used as the meta-learner, trained solely on the predicted probabilities from the base models. This choice enables the meta-learner to weigh the confidence of each model's predictions, leading to a more accurate final output.
- **Predicted Probabilities**: By utilizing predicted probabilities rather than binary predictions, the aim was to leverage the confidence levels of the predictions, allowing the meta-learner to more effectively capture the nuances in the base models' outputs.
- **Cross-Validation**: A 5-fold cross-validation process was applied, where base models were trained on subsets of the data, and their out-of-fold predictions were used to train the meta-learner. This approach minimized overfitting and ensured the meta-learner was built on diverse and unbiased predictions.
- **Model Diversity**: The stacking process carefully considers model diversity, leveraging it to improve resilience and reduce the impact of individual model errors.

6 Results

6.1 Performance of Individual Models

The performance of the models was calculated using several metrics accuracy, precision, recall, F-1 score, log loss, and ROC-AUC. The results of this evaluation are summarized in *Table 1* and visually represented in *Figure 5* below.

Table 1: Performance of Base Models.

Model	Accuracy	Precision	Recall	F1 Score	Log Loss	ROC-AUC
XGB	0.875	0.793	0.699	0.743	0.255	0.943
LR	0.812	0.717	0.448	0.552	0.435	0.815
RF	0.875	0.807	0.68	0.738	0.259	0.943
SVM	0.816	0.729	0.46	0.564	0.436	0.813
KNN	0.847	0.753	0.608	0.673	0.78	0.894
NN	0.849	0.836	0.515	0.637	0.313	0.912
ADA	0.823	0.697	0.554	0.618	0.482	0.883
NB	0.752	0.76	0.058	0.107	0.53	0.784
LDA	0.81	0.708	0.447	0.548	0.436	0.814
SGD	0.798	0.799	0.291	0.427	0.568	0.814

The highest rankings were achieved by the ensemble models XGBoost and Random Forest, which performed strongly across most metrics, particularly with ROC-AUC scores of 0.943 each. These models also showed balanced performance across other metrics, making them reliable choices for various applications.

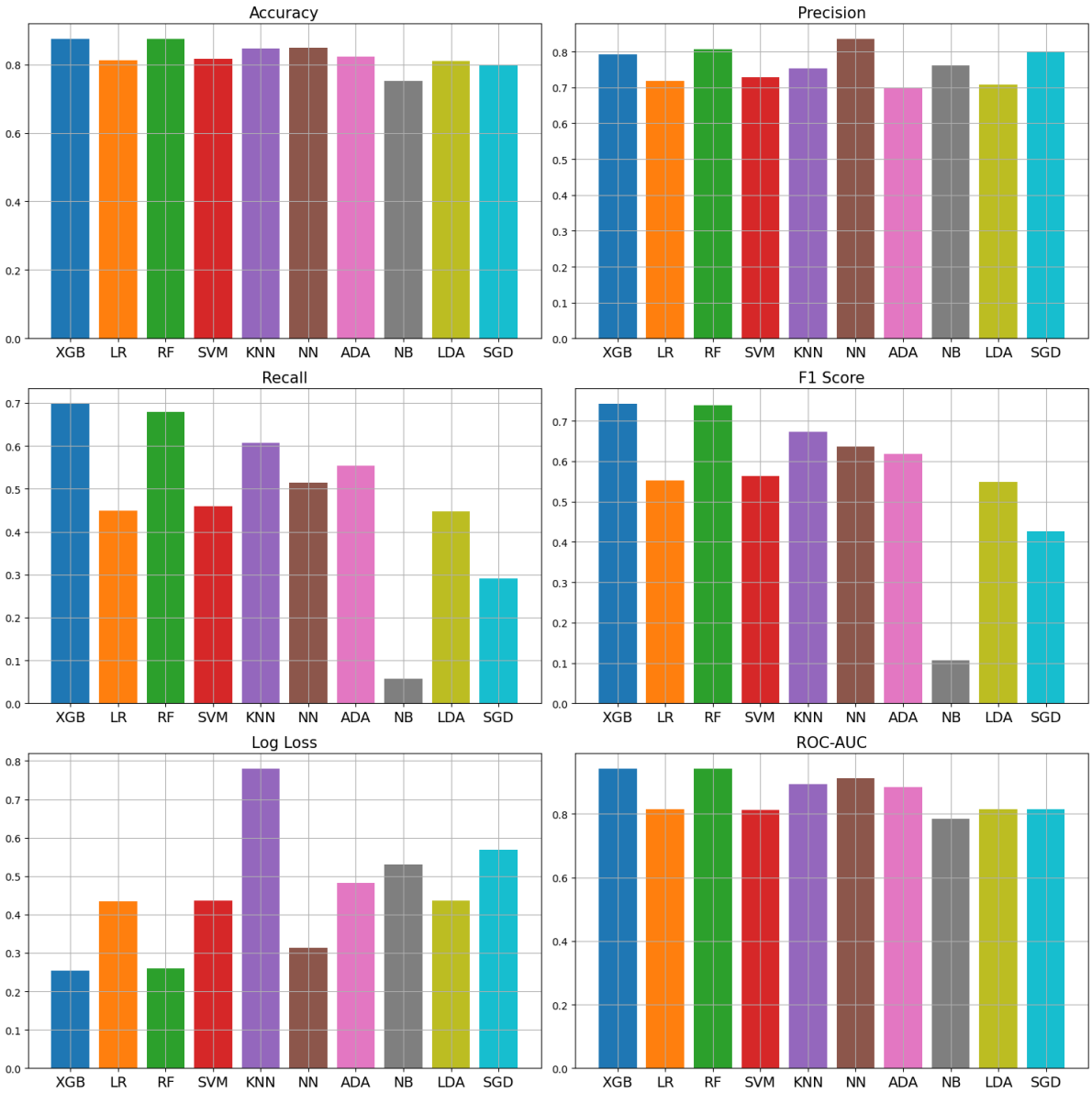

Figure 5: Histograms of Base Learners' Performance Across Different Metrics.

As shown in *Figure 5*, the Neural Network classifier demonstrated the highest precision, indicating its effectiveness in minimizing false positives. However, it had a lower recall, meaning it missed more true positives compared to other models like XGBoost, Random Forest and KNN. This means that while the Neural Network is good at identifying positive cases, it might miss some actual positive cases.

A balanced performance in precision and recall was demonstrated by XGBoost and Random Forest, making them suitable for minimizing both false positives and false negatives. XGBoost also recorded the lowest log loss, indicating more accurate probability predictions.

In summary, XGBoost and Random Forest were identified as the top performers across most metrics, though the Neural Network model exhibited the highest precision, which could be advantageous in scenarios where minimizing false positives is crucial.

6.2 Performance of Ensemble Models

The performance of various ensemble model combinations was evaluated using key. As shown in the table below, the combination of XGBoost, LR, KNN achieved the highest accuracy at 0.875 and the highest ROC-AUC at 0.943, highlighting its effectiveness in correctly classifying instances and distinguishing between classes.

Table 2: Performance of Ensemble Models.

Combination	Accuracy	Precision	Recall	F1 Score	Log Loss	ROC-AUC
XGB, LR, KNN	0.875	0.795	0.696	0.742	0.275	0.943
NN, NB, SGD	0.839	0.722	0.613	0.663	0.328	0.907
LR, RF, SGD	0.876	0.803	0.691	0.743	0.271	0.942
ADA, NB, SGD	0.833	0.691	0.639	0.664	0.344	0.891
LR, KNN, NB	0.851	0.772	0.602	0.677	0.344	0.899
XGB, LR, RF, NN, LDA	0.876	0.792	0.705	0.746	0.268	0.943
RF, KNN, ADA, NB, SGD	0.873	0.792	0.691	0.738	0.266	0.941
LR, RF, SVM, ADA, SGD	0.874	0.787	0.7	0.741	0.265	0.941
XGB, SVM, NN, NB, LDA	0.874	0.791	0.695	0.74	0.272	0.942
LR, RF, SVM, NN, ADA, LDA, SGD	0.873	0.787	0.697	0.739	0.265	0.942
XGB, LR, RF, ADA, NB, LDA, SGD	0.875	0.787	0.706	0.744	0.262	0.943
SVM, KNN, NN, ADA, NB, LDA, SGD	0.857	0.762	0.647	0.7	0.301	0.92

For precision, the combination of LR, RF, SGD stood out with a score of 0.803, making it particularly strong in minimizing false positives. In terms of recall, the ensemble XGBoost, LR, RF, NN, LDA led with a value of 0.705, indicating its capability to capture the most true positives. This same combination also excelled in F1-score with a value of 0.746, demonstrating a balanced performance between precision and recall.

When it comes to log loss, which measures the accuracy and confidence of probability predictions, the ensemble XGBoost, LR, RF, NN, LDA again performed the best, recording the lowest log loss at 0.262. This suggests that not only does this combination predict accurately, it also does so with high confidence.

These results suggest that different combinations may be preferable depending on the specific requirements of the task. For tasks that prioritize overall accuracy, the combination of XGBoost, LR, KNN would be ideal. For scenarios where minimizing false positives or capturing true positives is critical, the combinations LR, RF, SGD and XGBoost, LR, RF, NN, LDA are particularly effective.

6.3 Analysis of Results Using Performance Measures

In this study, the performance of various ensemble learning techniques is examined, with attention given to both homogeneous and heterogeneous approaches. Although the highest focus was on heterogeneous stacking ensembles, bagging and boosting are highlighted as examples of homogeneous ensemble algorithms. Random Forest is identified as a representative of bagging, while XGBoost and AdaBoost are used as examples of boosting algorithms. Although these algorithms are employed as base learners within the constructed

stacking ensembles, their individual performances are also significant and can be compared directly through the *Figure 6* presented below.

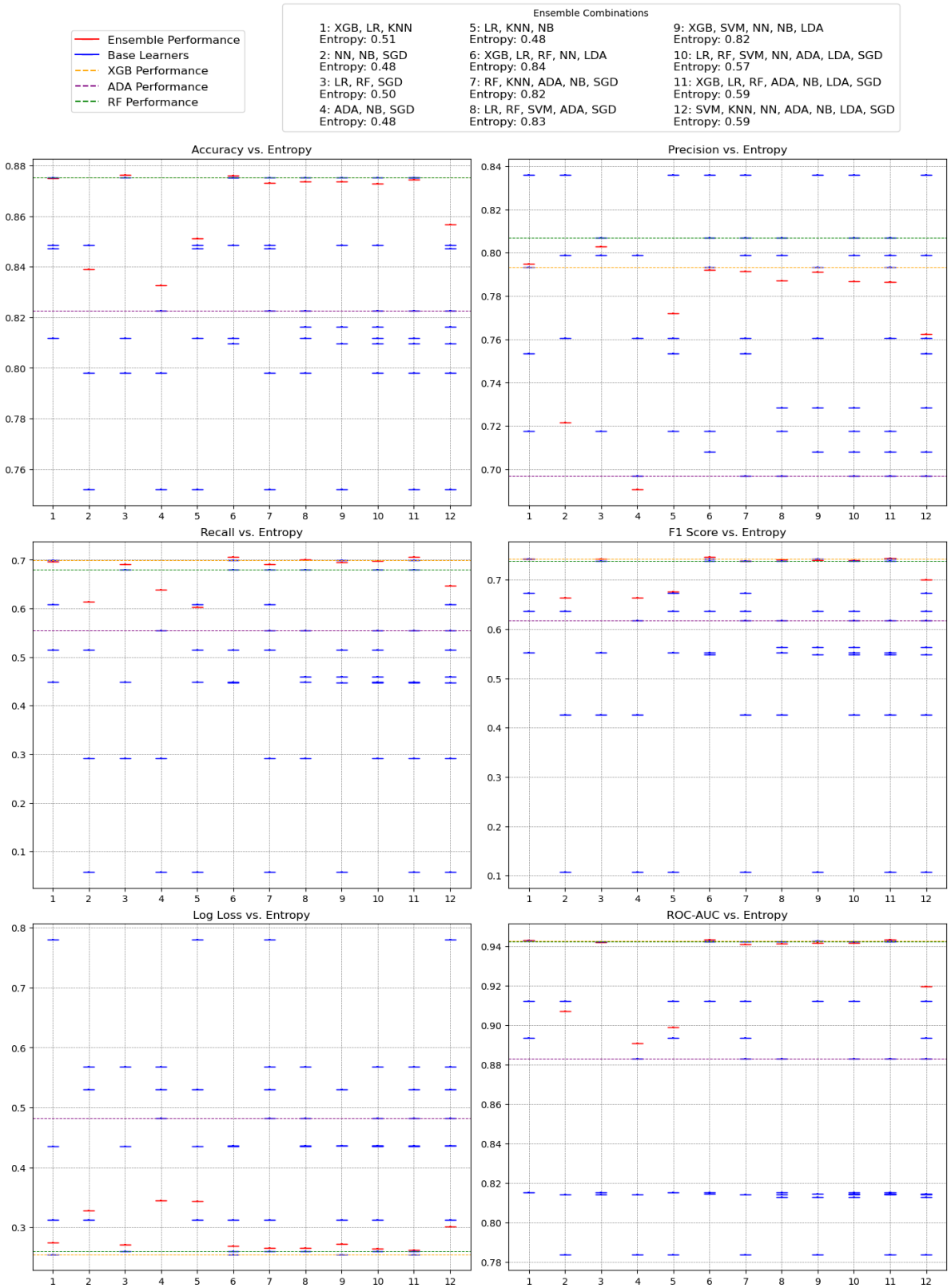

Figure 6: Performance comparison of Base and Ensemble Models across different metrics.

Accuracy: XGBoost and Random Forest both achieve an accuracy of 0.875, which is the highest among the individual models. However, some stacking ensembles, such as LR, RF, SGD slightly surpass this with an Accuracy of 0.876. Another ensemble, XGB, LR, RF, NN, LDA also matches this top performance, showcasing how combining different models can enhance accuracy beyond what is possible with a single ensemble algorithm.
Precision: When examining precision, the Neural Network model reaches the value of 0.836, XGBoost achieves a value of 0.793 and RF slightly outperforms it with 0.807. The stacking ensemble LR, RF, SGD is able to maintain a high precision of 0.803, demonstrating that while stacking maintains the precision of strong individual models, it also has the potential to improve other performance measures. The ensemble XGB, LR, KNN also performs closely with a precision of 0.795.
Recall: Recall is a critical metric, particularly in scenarios where it is important to identify as many relevant instances as possible. XGBoost records a recall of 0.699, RF 0.680, and ADA 0.554. Notably, the stacking ensembles XGB, LR, RF, NN, LDA and XGB, LR, RF, ADA, NB, LDA, SGD surpasses these with a recall of 0.706 and 0.705 respectively.
F1-Score: For the F1-Score, which balances precision and recall, XGB achieves 0.743, RF 0.738, and ADA 0.618. The stacking ensemble XGB, LR, RF, NN, LDA slightly exceeds XGB with an F-1 Score of 0.746, indicating that stacking can provide a more balanced performance, making it more effective in applications where both precision and recall are crucial.
Log Loss: In terms of log loss, a metric that measures the accuracy of probability estimates, XGB performs well with a log loss of 0.255, and RF is close behind at 0.259. The stacking ensemble XGB, LR, RF, ADA, NB, LDA, SGD achieves a log loss of 0.262, which is competitive with XGB's performance while benefiting from the added diversity of the ensemble.
ROC-AUC: Finally, in the ROC-AUC metric, which assesses the ability to distinguish between classes, both XGB and RF score 0.943. The stacking ensemble XGB, LR, RF, ADA, NB, LDA, SGD matches this performance with a ROC-AUC of 0.943, demonstrating that stacking can be as effective as the best individual models in ranking predictions.

An interesting observation from the *Figure 7* below is the consistent relationship between the diversity measure, entropy, and all performance measures. As entropy increases, indicating greater diversity within the ensemble, there is a notable positive trend across most performance metrics. This suggests that more diverse ensembles tend to perform better overall. The log loss graph reveals a negative trend, indicating that higher entropy is associated with lower log loss, meaning more accurate probabilistic predictions. These trends highlight the importance of model diversity in enhancing the effectiveness and reliability of ensemble methods.

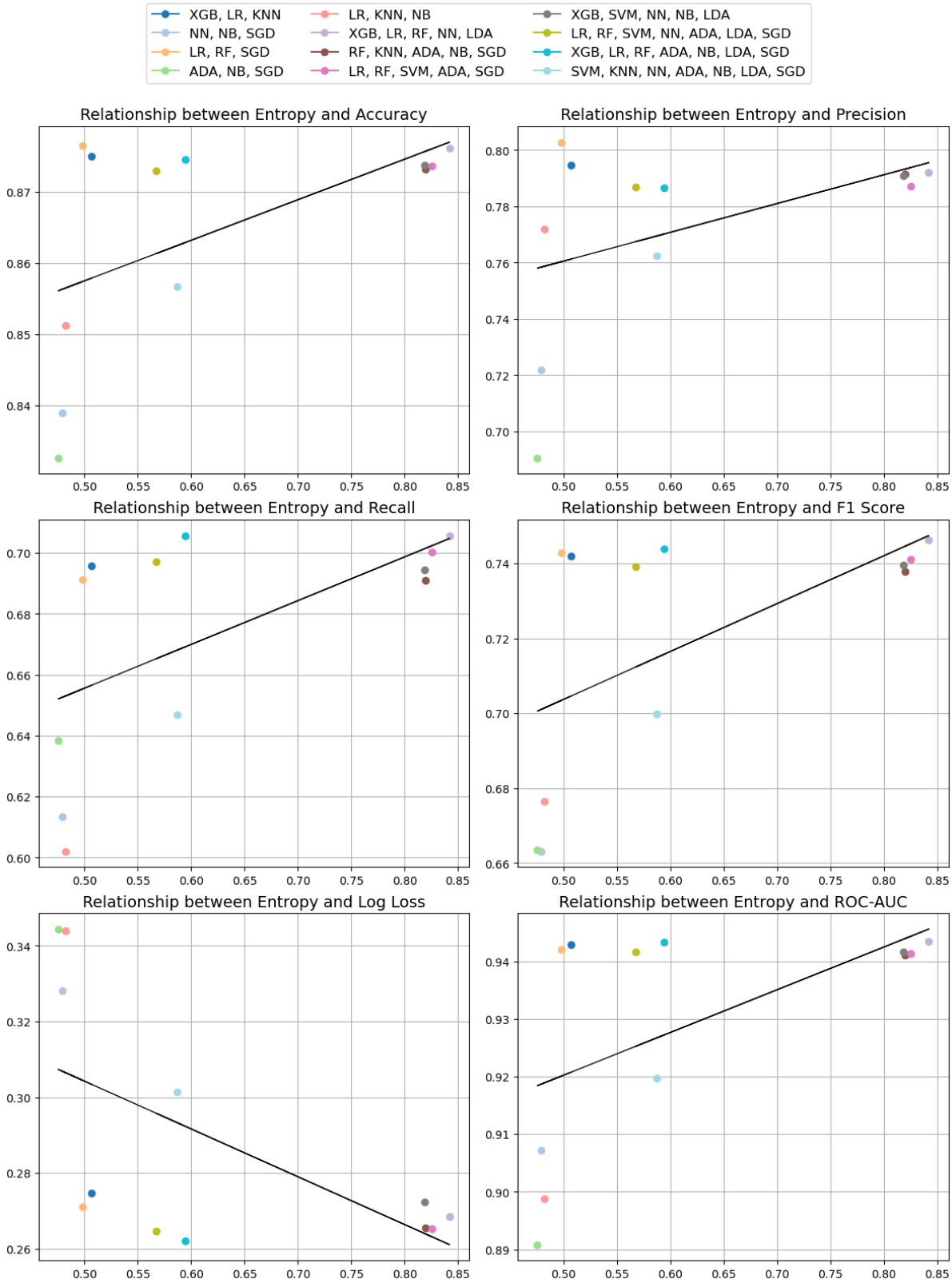

Figure 7: Diversity and Performance of Ensemble Models.

In summary, the analysis reveals that higher ensemble diversity, as indicated by entropy, generally correlates with better performance. Notably, some stacking ensembles were able to outperform homogeneous ensemble algorithms like XGBoost, AdaBoost, and Random Forest.

This suggests that stacking diverse models may provide a significant advantage over relying solely on strong but homogenous algorithms. However, it is important to note that these conclusions are based on a small sample size of only 12 ensembles, which limits the strength of the findings. The relationship between diversity and performance is complex, and the initial performance of individual models also plays a critical role. Given these limitations, the results should be interpreted with caution and further research with a larger sample size is needed to validate the potential benefits of diversity in ensemble effectiveness.

7 Discussion

7.1 Interpretation of Findings

The analysis suggests that ensemble methods, especially those that integrate various types of models, have the potential to enhance the accuracy and reliability of default predictions. The results indicate a possible positive correlation between ensemble diversity and improved performance across key metric. While these findings imply that combining models with different perspectives could contribute to more accurate predictions, the relationship is not definitive and may vary depending on specific circumstances.

Notably, some stacked ensembles outperformed individual models which are considered high performing in this field. This underscores the potential of heterogenous stacking approach to utilize the strengths of multiple models, leading to superior performance compared to homogeneous ensembles constructed with bagging and boosting.

However, the relationship between diversity and performance is complex. While increased diversity might align with better outcomes according to findings, the effectiveness of individual models within the ensemble remains a crucial factor. In certain cases, less diverse ensembles with particularly strong base models performed comparably well, and increased the performance even more, indicating that the benefits of diversity are context-dependent and influenced by the characteristics of the data and models involved.

7.2 Implications for Default Prediction

These findings have significant implications for the field of default prediction. Financial institutions and researchers are encouraged to adopt ensemble methods that prioritize diversity, as this can enhance predictive accuracy and reduce lending risks. By integrating a variety of models into predictive frameworks, institutions can achieve more accurate assessments of credit risk, leading to better decision making and increased financial stability. The success of stacked ensembles in surpassing traditional models like XGBoost and Random Forest despite being used as base models suggests that the future of default prediction may lie in more complex, heterogeneous approaches rather than relying solely on single, well established algorithms. This could pave the way for the development of more robust models, better equipped to handle the complexities of financial data.

7.3 Advantages and Limitations of the Approach

A key advantage of the approach taken in this study is its emphasis on model diversity to enhance ensemble performance. By using the heterogenous stacking metohd, the study

provides evidence suggesting that combining different models can improve predictive efficiency.
However, several limitations must be acknowledged. One significant drawback is the computational complexity involved in creating and evaluating diverse ensembles, particularly with large datasets or a high number of models. Additionally, while diversity was shown to be beneficial, it does not guarantee improved performance. The initial quality of the base models and their relevance to the specific prediction task are critical factors that must be carefully considered.
Moreover, the selection of both performance and diversity measures is crucial. Depending on the context, the most appropriate performance metric and diversity measure might vary, and relying on a single measure could lead to biased evaluations. In some cases, a more generalized or context specific measure may need to be identified to accurately assess the effectiveness and diversity of the models.

7.4 Comparison with State-of-the-Art Methods

When compared to state-of-the-art methods such as XGBoost and Random Forest, although ensembles consisting solely of non-ensemble learning algorithms could not surpass the performance of XGBoost or Random Forest, incorporating these strong single learners into the ensemble construction is shown to be beneficial for prediction performance.
These findings suggest that while traditional single models remain valuable, integrating them into advanced ensemble techniques that emphasize diversity can lead to even greater improvements in default prediction.

7.5 Potential for Future Research

The findings of this study open several promising areas for future research. One potential direction is the exploration of different diversity measures to refine the selection and combination of models in an ensemble. Understanding how various diversity metrics influence ensemble performance could lead to more precise strategies for model integration and boost predictive accuracy.
Additionally, addressing data anomalies presents another important area for future work. Investigating how different ensemble approaches handle outliers and irregularities in the data could lead to stronger models that perform consistently well.
Another critical aspect for future research is the development of better or more accurate performance measures. Current metrics might not fully capture the nuances of model performance, especially in complex predictive tasks like default prediction. Identifying or designing metrics that provide a more comprehensive evaluation of model success could improve the assessment and selection of predictive models.
Moreover, the integration of more ensemble models using heterogeneous algorithms offers a valuable research path. By combining a broader range of algorithms, studies could explore the potential for even greater performance.

8 Conclusion

8.1 Summary of Key Findings

This thesis explored the role of heterogenous ensemble methods in enhancing default prediction, with a particular focus on the importance of model diversity. Through empirical analysis, it was found that ensembles integrating various models with stacking method can improve prediction capability. The findings indicated that while traditional single models like XGBoost and Random Forest are highly effective, ensembles incorporating these strong learners alongside other weaker models can achieve even better results.
An interesting observation was the positive correlation between ensemble diversity, measured by entropy, and improvements in key performance. Higher entropy, implying greater model diversity, generally aligned with better performance. This suggests the potential importance of diversity within ensembles for enhancing their predictive capabilities.

8.2 Contributions to the Field

This study aimed to explore the field of predictive modeling and risk management by applying advanced ensemble techniques and empirically evaluating these methods against state-of-the-art models with the intention of adding insights to the ongoing research in this area. Furthermore, this work underlines the potential benefits of integrating strong single models into diverse ensembles to improve predictive performance, particularly in the context of financial risk assessment.

8.3 Recommendations for Practitioners

Practitioners in the financial industry are advised to consider adopting ensemble methods that prioritize diversity when developing predictive models for default risk. The study's findings suggest that ensembles that combine heterogeneous algorithms and strong single learners like XGBoost and Random Forest can lead to more accurate and reliable predictions. Additionally, careful selection of performance and diversity measures or a comprehensive approach that combines the strengths of multiple measures could be beneficial for optimizing model performance. Practitioners should also consider the computational cost in creating and evaluating diverse ensembles and balance these challenges against the potential benefits of predictive performance.

8.4 Directions for Future Work

Several areas for future research have been identified based on the findings of this thesis. First, further exploration of different diversity measures could lead to more refined and effective strategies for model selection and integration within ensembles. Second, investigating how ensemble models handle data anomalies and outliers could help in developing better predictive systems. Third, the development of more appropriate performance metrics that can better capture the complexities of model performance should be considered in default prediction. Finally, future research could investigate the integration of a wider variety of ensemble models with a particular focus on heterogeneous algorithms.

9 Bibliography

[1] F. P. S. N. a. G. V. M. Moscatelli, "Corporate default forecasting with machine learning," *Expert Systems with Applications,* vol. 161, p. 113567, 2020.

[2] D. J. &. H. W. E. Hand, "Statistical classification methods in consumer credit scoring: A review.," *Journal of the Royal Statistical Society: Series A (Statistics in Society),* vol. 160, no. 3, pp. 523-541, 1997.

[3] T. &. K. H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data.," *PloS One,* vol. 14, no. 2, p. Article e0212320, 2019.

[4] Z.-H. Zhou, Ensemble methods: Foundations and algorithms, Chapman and Hall/CRC, 2012.

[5] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, Berlin, Heidelberg, Springer, 2000, pp. 1-15.

[6] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996.

[7] Y. &. S. R. E. Freund, "Experiments with a new boosting algorithm," in *Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)*, 1996.

[8] D. H. Wolpert, "Stacked generalization," *Neural Networks,* vol. 5, no. 2, pp. 241-259, 1992.

[9] L. Breiman, "Random forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[10] H. &. F. Y. He, "A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction," *Expert Systems with Applications ,* vol. 183, p. 114899, 2021.

[11] A. F. E. H. G. &. P. B. Bifet, "Ensembles of restricted Hoeffding trees," *ACM Transactions on Intelligent Systems and Technology (TIST) ,* vol. 3, no. 2, pp. Article 30, 20 pages, 2012.

[12] B. E. R. &. S. E. Singh, "Enhancing Prediction Accuracy of Default of Credit Using Ensemble Techniques," in *First International Conference on Computational Intelligence and Informatics*, Singapore, 2019.

[13] S. K. M. K. T. &. M. Y. Hamori, "Ensemble learning or deep learning? Application to default risk analysis," *Journal of Risk and Financial Management ,* vol. 11, no. 1, p. 12, 2018.

[14] W. D. S. W. H. C. Y. &. Y. S. Li, "Heterogeneous ensemble learning with feature engineering for default prediction in peer-to-peer lending in China," *World Wide Web,* vol. 23, no. 1, pp. 23-45, 2020.

[15] Y. W. K. S. H. Q. H. C. J. L. W. &. C. C. Lu, "Research on user default prediction algorithm based on adjusted homogenous and heterogeneous ensemble learning," *Applied Sciences,* vol. 14, no. 13, p. 5711, 2024.

[16] Z. W. F. &. J. W. Kun, " Default identification of P2P lending based on stacking ensemble learning," in *Proceedings of the 2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*, 2020.

[17] L. I. Kuncheva, "That elusive diversity in classifier ensembles," in *Pattern Recognition and Image Analysis. IbPRIA 2003*, Berlin, Heidelberg, Springer, 2003, pp. 1126-1138.

[18] L. I. Kuncheva, "Diversity in classifier ensembles," in *Combining pattern classifiers: Methods and algorithms*, John Wiley & Sons, 2004, pp. 295-326.

[19] L. K. &. S. P. Hansen, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, no. 10, pp. 993-1001, 1990.

[20] D. &. M. R. Opitz, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research,* vol. 11, pp. 169-198, 1999.

[21] M. A. O. D. L. F. G. &. L. B. Ahmed, "Using diversity for classifier ensemble pruning: An empirical investigation," *Theoretical and Applied Informatics,* vol. 29, no. 1-2, pp. 25-39, 2017.

[22] W. D. S. C. Y. &. Y. S. Li, "Heterogeneous Ensemble for Default Prediction of Peer-to-Peer Lending in China," *IEEE Access,* 2018.

[23] G. W. J. L. H. R. &. Y. X. Brown, "Diversity creation methods: A survey and categorization," *Information Fusion,* vol. 6, no. 1, pp. 5-20, 2005.

[24] L. I. &. W. C. J. Kuncheva, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning,* vol. 51, no. 2, pp. 181-207, 2003.

[25] D. &. K. W. Partridge, "Software diversity: Practical statistics for its measurement and exploitation," *Information and Software Technology,* vol. 39, no. 19, pp. 707-717, 1997.

[26] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 832-844, 1998.

[27] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *In IJCAI (International Joint Conference on Artificial Intelligence)*, 1995.

[28] Y. &. Y. X. Liu, "Ensemble learning via negative correlation," *Neural Networks,* vol. 12, no. 10, pp. 1399-1404, 1999.

[29] E. &. K. R. Bauer, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning,* vol. 39, no. 1-2, p. 105–139, 1999.

[30] T. T. R. &. F. J. Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009.

[31] T. &. G. C. Chen, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 2016.

[32] D. Berrar, "Performance Measures for Binary Classification," 2018.

10 Appendix

Additional Figures and Tables

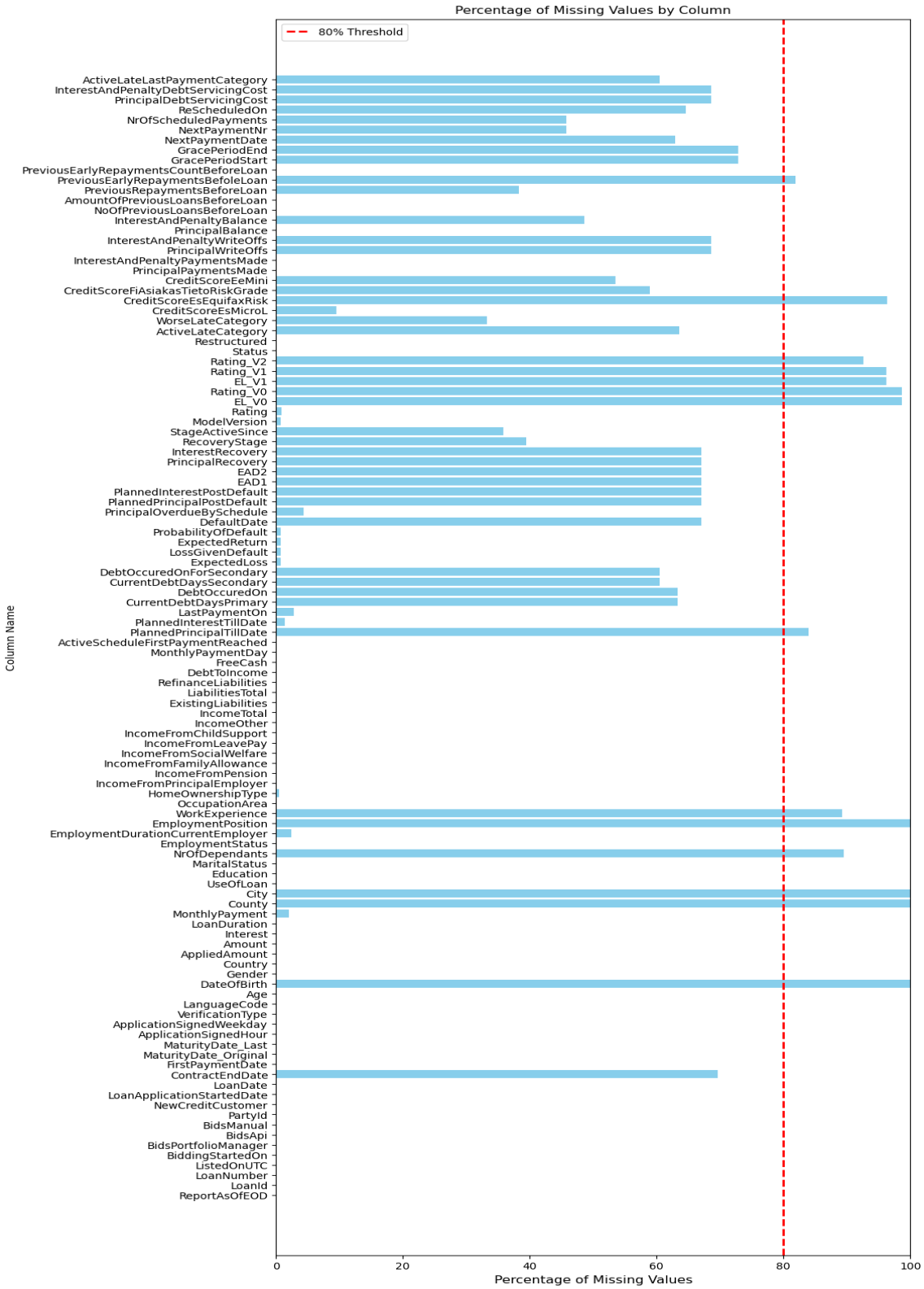

Figure A.1.: Percentages of missing values.

Figure A.2.: Distribution of data after outlier capping.

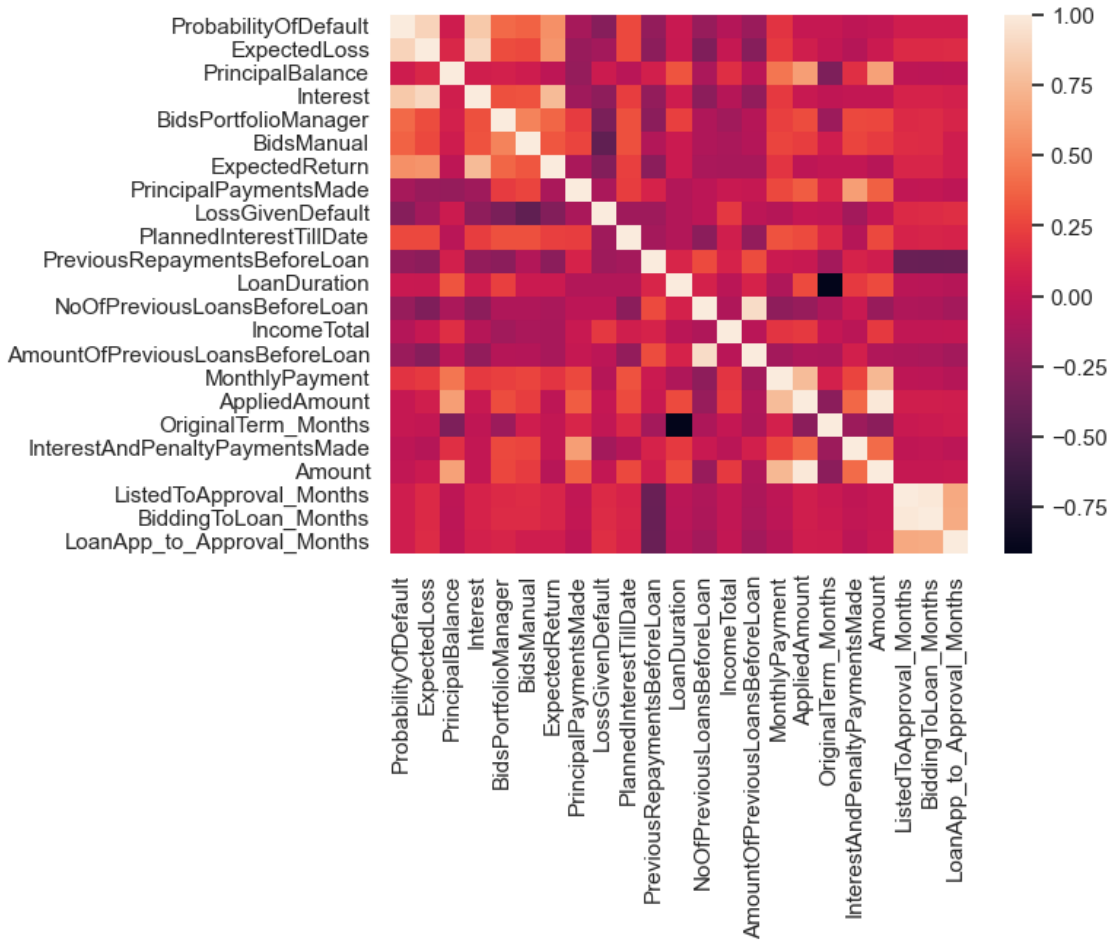

Figure A.3.: Correlation heatmap of numeric features before feature selection.

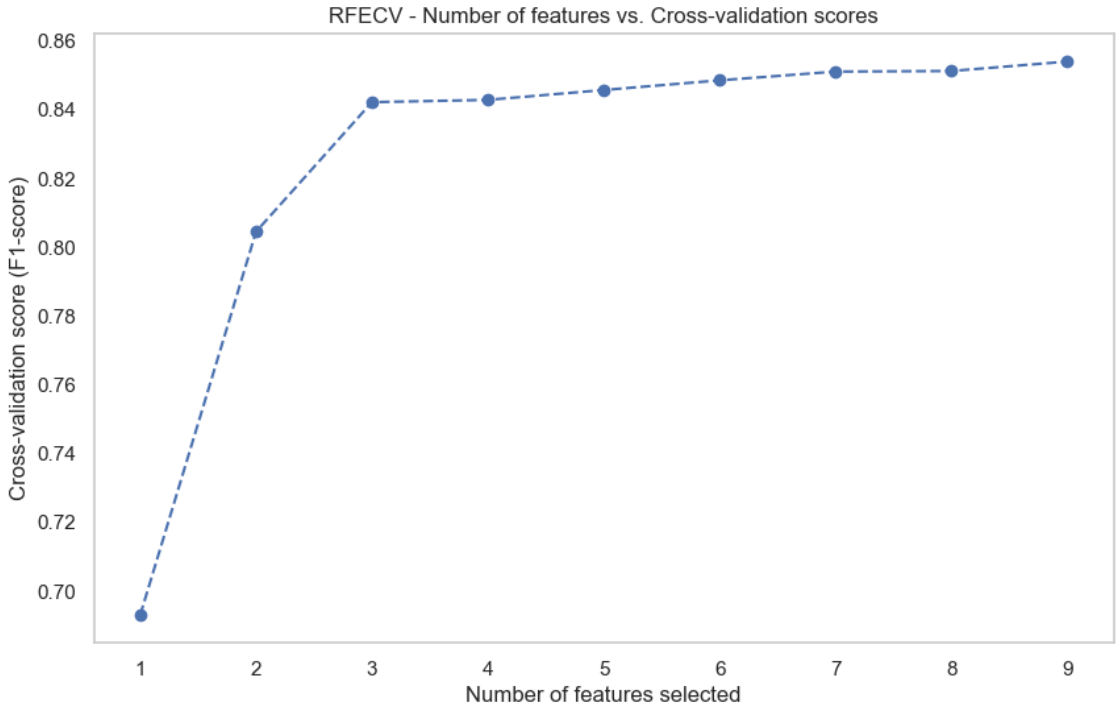

Figure A.4.: RFECV results showing the relationship between the number of features selected and the cross-validation F1-score.

Table A.1.: Pairwise Diversity of Base Models

Combinations	Q-score	Rho	Disagreement Score	Double Fault	Entropy	Generalized Diversity
(LDA, SGD)	1.0	0.72984	0.06911	0.83674	0.32151	0.2685
(LR, SGD)	0.99991	0.73432	0.06737	0.83854	0.32196	0.26363
(SVM, SGD)	0.99962	0.72855	0.06928	0.8368	0.32282	0.26942
(LR, LDA)	0.99955	0.9596	0.01099	0.83217	0.32429	0.03386
(LR, SVM)	0.99774	0.91251	0.02378	0.82591	0.32561	0.07331
(SVM, LDA)	0.99651	0.89297	0.02922	0.82227	0.32515	0.08958
(XGB, RF)	0.99396	0.87334	0.04397	0.75539	0.35006	0.09875
(RF, NN)	0.97844	0.71832	0.09066	0.76633	0.34475	0.24068
(XGB, NN)	0.97452	0.69425	0.10133	0.75604	0.34473	0.26211
(RF, KNN)	0.9704	0.7395	0.08745	0.74325	0.3445	0.20526
(XGB, KNN)	0.95948	0.70135	0.10225	0.7309	0.34448	0.23455
(KNN, NN)	0.95794	0.65995	0.10508	0.76377	0.33917	0.28603
(ADA, SGD)	0.95049	0.52405	0.1373	0.78151	0.32411	0.45814
(NB, SGD)	0.94856	0.32611	0.08364	0.90134	0.30996	0.73576
(NN, ADA)	0.94503	0.6273	0.11379	0.76083	0.33423	0.31214
(LR, ADA)	0.94498	0.62972	0.11335	0.75985	0.32689	0.30891
(SVM, ADA)	0.94378	0.62837	0.11395	0.75876	0.32775	0.30921
(ADA, LDA)	0.94304	0.62684	0.11444	0.75838	0.32644	0.31031
(RF, SGD)	0.93615	0.49014	0.1527	0.76774	0.33463	0.4897
(NN, SGD)	0.93597	0.53246	0.10949	0.81868	0.3293	0.4325
(XGB, SGD)	0.93535	0.48103	0.16108	0.7586	0.33461	0.50068
(RF, ADA)	0.93023	0.62423	0.12554	0.72562	0.33956	0.29664
(SVM, NN)	0.92935	0.59177	0.11031	0.78385	0.33294	0.34257
(KNN, SGD)	0.92538	0.48108	0.14916	0.77416	0.32905	0.49307
(NB, LDA)	0.92479	0.25042	0.1508	0.83321	0.31229	0.82495
(LR, NN)	0.92186	0.57452	0.1145	0.78254	0.33208	0.35734
(NN, LDA)	0.92176	0.57494	0.11493	0.7814	0.33163	0.35664
(XGB, ADA)	0.91986	0.60303	0.13523	0.71582	0.33954	0.31222
(RF, SVM)	0.91282	0.56021	0.13893	0.7402	0.33828	0.36498
(SVM, NB)	0.9121	0.24111	0.1515	0.83299	0.31361	0.83005
(LR, NB)	0.9117	0.2416	0.15003	0.83451	0.31274	0.82917
(XGB, SVM)	0.91164	0.55537	0.14437	0.73253	0.33826	0.36965
(SVM, KNN)	0.90544	0.54885	0.13877	0.74494	0.3327	0.37368
(LR, RF)	0.90368	0.54281	0.14399	0.73846	0.33742	0.37984
(KNN, ADA)	0.90211	0.567	0.14214	0.72197	0.33398	0.34341
(RF, LDA)	0.90173	0.54125	0.14475	0.73716	0.33696	0.38
(XGB, LR)	0.89783	0.53095	0.15172	0.72965	0.3374	0.39004
(NN, NB)	0.89699	0.23216	0.14884	0.83631	0.32008	0.83359
(KNN, LDA)	0.89524	0.53179	0.14394	0.74222	0.33138	0.38732
(XGB, LDA)	0.8952	0.52861	0.1527	0.72823	0.33694	0.3907
(LR, KNN)	0.89502	0.5299	0.14416	0.74303	0.33184	0.38984
(KNN, NB)	0.87651	0.20156	0.1971	0.7875	0.31983	0.86485
(RF, NB)	0.84816	0.18566	0.2075	0.77764	0.32541	0.87474
(XGB, NB)	0.83944	0.1794	0.2174	0.76774	0.32539	0.87976
(ADA, NB)	0.55669	0.09167	0.20679	0.78407	0.31489	0.91876

Libraries

The following Python libraries were used:

- Pandas: https://pandas.pydata.org/
- NumPy: https://numpy.org/
- Scikit-learn: https://scikit-learn.org/
- TensorFlow: https://www.tensorflow.org/
- SciKeras: https://www.adriangb.com/scikeras/stable/
- imbalanced-learn: https://imbalanced-learn.org/stable/
- XGBoost: https://xgboost.readthedocs.io/
- Statsmodels: https://www.statsmodels.org/
- Matplotlib: https://matplotlib.org/
- Itertools: https://docs.python.org/3/library/itertools.html