Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 4, August 2024, pp. 453-467

DOI: https://doi.org/10.52436/1.jutif.2024.5.4.2022
p-ISSN: 2723-3863
e-ISSN: 2723-3871

STACKING ENSEMBLE LEARNING AND INSTANCE HARDNESS THRESHOLD FOR BANK TERM DEPOSIT ACCEPTANCE CLASSIFICATION ON IMBALANCED DATASET

Bangun Watono[*1], Ema Utami[2], Dhani Ariatmanto[3]

[1,2,3]Magister of Informatics Engineering, Universitas AMIKOM Yogyakarta, Indonesia
Email: [1]bangun.watono@students.amikom.ac.id, [2]ema.u@amikom.ac.id, [3]dhani@amikom.ac.id

(Article received: April 28, 2024; Revision: May 26, 2024; published: August 07, 2024)

Abstract

Bank term deposits are a popular banking product with relatively high interest rates. Predicting potential customers is crucial for banks to maximize revenue from this product. Therefore, bank term deposits acceptance classification is an important challenge in the banking industry to optimize marketing strategies. Previous studies have been conducted using machine learning classification techniques with the imbalanced Bank Marketing Dataset from the UCI Repository. However, the accuracy results obtained still need to be improved. Using the same dataset, this study proposes an Instance Hardness Threshold (IHT) undersampling technique to handle imbalanced datasets and Stacking Ensemble Learning (SEL) for classification. In this SEL, Decision Tree, Random Forest, and XGBoost are selected as base classifiers and Logistic Regression as meta classifier. The model trained on SEL with the dataset undersampled using IHT shows a high accuracy rate of 98.80% and an AUC-ROC of 0.9821. This performance is significantly better than the model trained with the dataset without undersampling, which achieved an accuracy of 90.30% and an AUC-ROC of 0.6898. The findings of this research demonstrate that implementing of the suggested IHT undersampling technique combined with SEL has been evaluated to effectively enhance the performance of term deposit classification on the dataset.

Keywords: *Bank Term Deposit, Classification, Instance Hardness Threshold, Machine Learning, Stacking Ensemble Learning, Undersampling.*

1. INTRODUCTION

Bank term deposits provide a secure and reliable way to earn higher interest compared to regular savings accounts. Their attractiveness stems from the fixed interest rate over a predetermined period, making them suitable for achieving short-term or medium-term financial goals [1]. However, offering term deposits requires banks to accurately predict customer acceptance to maximize revenue from this product. Customer rejection of term deposit offers translates into lost interest income for the bank. As such, banks must conduct thorough customer research to understand their profiles, needs, and preferences regarding term deposit products [2]. Predicting bank term deposit acceptance is a complex task due to the multitude of factors influencing customer decisions. The complexity is further amplified when dealing with imbalanced datasets [3].

Multiple research endeavors have investigated imbalanced data [4] classification methods within the banking sector, particularly for customer profiling, telemarketing success prediction, and time deposit acceptance forecasting. Guo et al. [5] employed an Artificial Neural Network (ANN) optimized with four metaheuristic optimization algorithms, namely Electromagnetic Field Optimization (EFO), Social Ski-Driver (SSD), Harmony Search Algorithm (HSA) and Future Search Algorithm (FSA). Their findings indicated that the EFO-ANN combination yielded the highest predictive performance, achieving an accuracy of 88.76% and an AUC of 0.79.

This section reviews previous research on feature selection and imbalanced data handling techniques for time deposit acceptance prediction. Ghatasheh et al. [6] proposed a hybrid approach combining Genetic Algorithm (GA) for feature selection and Extreme Boosting algorithm for classification. Their method achieved an average accuracy of 89.07% using 10-fold cross-validation. Additionally, Hayder et al. [7] employed the SMOTE (Synthetic Minority Oversampling Technique) method to address the imbalanced dataset and compared the performance of four classification algorithms: Support Vector Machine (SVM), Decision Tree, Naïve Bayes and K-Nearest Neighbor (KNN). Their results demonstrated that SMOTE improved classification accuracy, with Decision Tree achieving the highest accuracy of 91%.

This section reviews previous research on ensemble selection and feature analysis for telemarketing success prediction in time deposit sales. Feng et al. [8] introduced META-DES-AAP, a novel dynamic ensemble selection methodology, for predicting the performance of bank time deposit telemarketing initiatives. Their inquiry further sought

to examine the variables impacting telemarketing effectiveness and average profit under the META-DES-AAP framework. The proposed method yielded an accuracy rate of 89.39% and an AUC of 89.44%.

This segment reviews previous research on optimizing bank marketing data classification using Genetic Algorithm (GA), Bagging, and Naive Bayes. Religia et al. [9] compared the performance of Random Forest Classifier with and without GA and Bagging techniques. Their findings suggest that combining GA and Bagging with Random Forest did not improve classification accuracy, as all four methods achieved 88.30%. Nugroho and Religia Religia [10] investigated Naive Bayes optimization, comparing its performance with GA, Bagging, and the combination of both. Their results showed that the combination of Bagging and GA significantly improved Naive Bayes effectiveness, achieving a peak accuracy of 89.73% (an increase of 4.57%). Additionally, Alsolami et al. [11] explored Logistic Regression, Decision Tree, and Multilayer Perceptron models for bank marketing data classification. Their research identified Logistic Regression as the most effective method, achieving an accuracy of 91.48%.

Previous research on imbalanced data classification for time deposit acceptance prediction has consistently utilized the Bank Marketing Dataset [4] and explored various methods to improve classification accuracy. However, there remains room for further investigation and potential enhancement. Therefore, this study proposes an Instance Hardness Threshold (IHT) and Stacking Ensemble Learning (SEL) method to tackle the drawbacks of prevailing approaches and contribute to the advancement of imbalanced data classification techniques in this domain.

The IHT tackles the challenge of imbalanced datasets in classification tasks, where one class (often the majority class) has significantly more data points than the others. IHT specifically addresses this issue by employing an undersampling technique that focuses on removing the most difficult-to-classify data points from the majority class. These "hard" data points are those that are frequently misclassified by the model, leading to a decrease in overall accuracy [12]. To identify these challenging instances, IHT utilizes an auxiliary classifier trained on the imbalanced dataset itself. This auxiliary classifier assigns a "hardness score" to each data point, reflecting the difficulty the model has in correctly classifying it. Based on these hardness scores, IHT selectively discards data points belonging to the majority class that are considered particularly problematic to classify [13]. This selective removal process helps to balance the dataset and reduces the model's bias towards the majority class. Consequently, IHT provides numerous benefits: it enhances classification accuracy by prioritizing easily classifiable data, optimizes training data usage, and diminishes bias towards the majority class.

Then, for classification purposes, one approach in machine learning is the utilization of ensemble learning [14]. Ensemble learning stands out as a robust machine learning strategy, involving various models to address specific challenges. Rather than relying on a single model, ensemble approaches leverage a diverse array of models and consolidate their predictions to enhance overall performance [15]. Ensemble learning offers several advantages, including enhanced prediction accuracy, feature selection, improved understanding of metrics, and effective handling of large-scale data [16]. Numerous studies have demonstrated the superiority of ensemble methods over single models. For instance, Divina et al. [17] and Baccouche et al. [18] have shown that ensemble learning techniques consistently outperform single models.

Several commonly utilized ensemble learning methodologies include Bagging, Boosting, and Stacking [19]. In the study by Lazzarini et al. [20], they implemented Stacking Ensemble Learning (SEL) by employing MLP, DNN, CNN, LSTM as base classifiers, and integrating DNN as a meta-classifier, which resulted in an impressive accuracy of 99.60% in IoT intrusion detection. Similarly, Almulihi et al. [21] proposed SEL using CNN-LSTM and CNN-GRU as base classifiers, followed by employing SVM as a meta-classifier, attaining a remarkable accuracy rate of 97.17% in the prompt detection of heart disease utilizing the Cleveland Heart Disease dataset. Furthermore, AlJame et al. [22] explored SEL by leveraging ExtraTrees, Random Forest, and Logistic Regression as base classifiers, while utilizing XGBoost as a meta-classifier, culminating in a remarkable accuracy of 99.88% for COVID-19 diagnosis based on routine blood examination data. Drawing insights from these studies, Stacking Ensemble Learning (SEL) is a particular ensemble method that involves training a meta-classifier on the predictions of multiple base classifiers. This hierarchical structure allows SEL to capture higher-order relationships between the data and achieve superior performance. Given the success of SEL in various applications, this study adopts SEL as the ensemble learning method for the proposed approach. The goal is to harness the power of SEL to enhance the classification performance and address the challenges of imbalanced data in the context of time deposit acceptance prediction.

This study proposes a Stacking Ensemble Learning (SEL) approach to improve classification accuracy for time deposit acceptance prediction using the Bank Marketing Dataset [4]. The selection of DT Classifier, RF Classifier, XGBoost Classifier, and Logistic Regression for the SEL method is based on their demonstrated effectiveness in prior research and their unique advantages. Decision Trees excel at capturing complex non-linear relationships within

data [20]. Random Forests combine multiple Decision Trees to enhance accuracy and reduce overfitting [21]. XGBoost, a powerful boosting algorithm, is particularly adept at handling high-dimensional and complex data [22]. Finally, Logistic Regression, a widely used and interpretable statistical method [23], proves valuable for diverse classification tasks [24].

To contextualize the proposed approach, this section highlights the performance of single algorithms applied to the Bank Marketing Dataset in previous studies: Hayder et al. [7] using Decision Tree and SMOTE achieved 91% accuracy. Religia et al. [9] using Random Forest achieved 88.30%. Ghatasheh et al. [6] using XGBoost achieved 89.07% and Alsolami et al. [25] using Logistic Regression achieved 91.48%. To comprehensively evaluate the SEL approach, the study will consider various classification scenarios, including using the original imbalanced data, resampled data to address class distribution issues, and comparing the performance of individual models to the ensemble model. Furthermore, the evaluation will extend beyond accuracy by incorporating additional metrics like precision, recall, F1-score and AUC to assess the model's ability to distinguish between positive and negative classes. along with the intended evaluation approach, seeks to thoroughly evaluate the efficiency of the Stacking Ensemble Learning model in addressing the hurdles of imbalanced data classification for bank term deposit acceptance.

2. RESEARCH METHODOLOGY

In Figure 1, the research flowchart is depicted, encompassing data collection, data understanding, preprocessing, without or with undersampling, data splitting, classification scenario and evaluation.

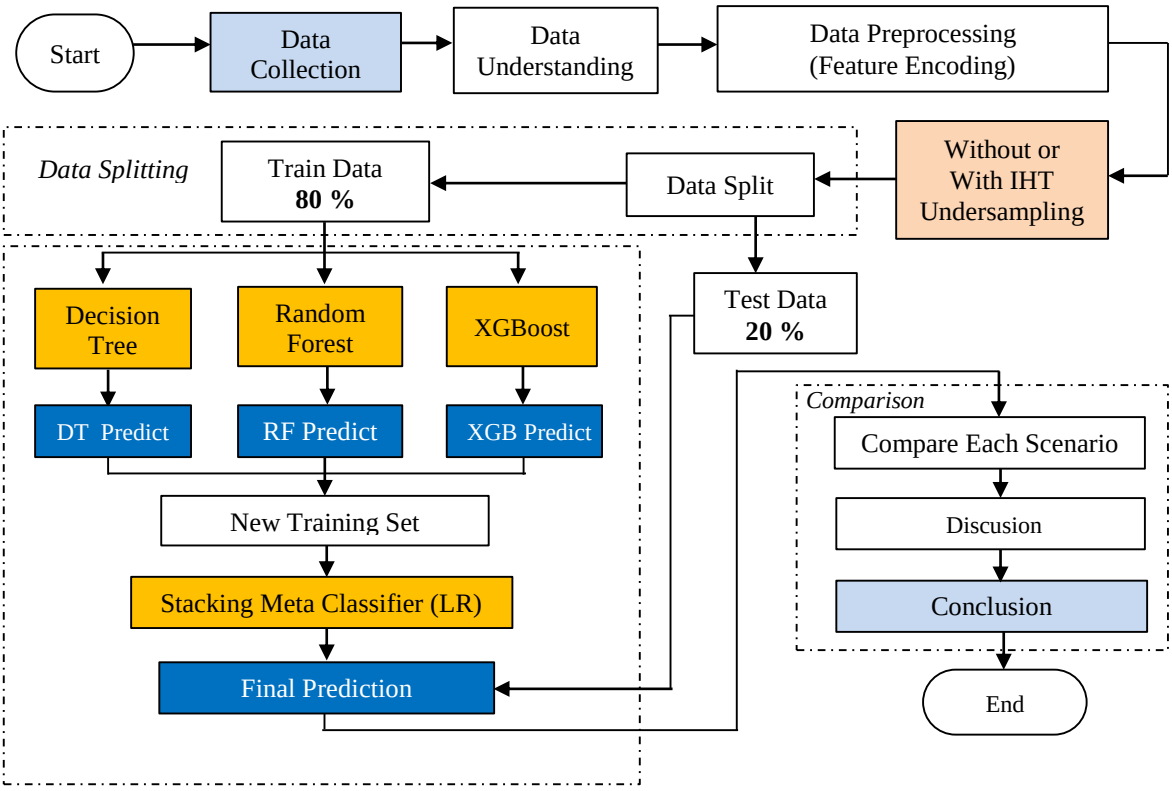

Figure 1. Research Flow

2.1. Data Collection

Data collection is the process of gathering data that will be used in the research. This is done by conducting a literature study to obtain theories and previous research as reference materials for this research. Next, the dataset is collected from the UCI related to bank term deposit acceptance.

2.2. Data Understanding

Data understanding involves iteratively analyzing and visualizing data to uncover characteristics, patterns, and relationships. This crucial step in data science allows for informed model building and decision-making.

2.3. Data Preprocessing

Data preprocessing is crucial for optimizing data for machine learning. Cleaning, transforming, and engineering data enhance model performance, efficiency, and generalizability [28]. In the Bank Marketing Dataset, preprocessing techniques such as label encoding and ordinal encoding can greatly enhance the effectiveness of time deposit acceptance prediction models [29].

2.3.1. Label Encoding

Label encoding simplifies data by converting categories to numbers, making it machine-learning friendly. However, it can lose order information and impact model performance, so use it cautiously. [30]. For example, label encoding assigns numbers to categories. "Yes" might be 1 and "No" might be 0 for credit status.

2.3.2. Ordinal Encoding

Unlike label encoding, ordinal encoding translates categorical data into numerical sequences, enabling machine learning models to understand relationships such as "greater than" and "lesser than" [31]. Ordinal encoding assigns values based on order, like 5 for "very satisfied" and 1 for "very dissatisfied" in customer reviews. This helps models make better predictions. [32]. Ordinal encoding can also be applied to non-order-sensitive attributes.

2.4. Instance Hardness Threshold (IHT)

The IHT approach is a type of undersampling technique utilized for identifying data instances with elevated instance hardness within a dataset, namely data that has a high probability of being misclassified by the machine learning model. In the data selection process, this "instance hardness" value, which is abbreviated as IH, is obtained from calculating the probability $p(h|t)$ using Bayes' theorem [12]. $p(h|t)$: probability of function mapping (h) producing a particular label (t) based on training data (t). h: function that maps input features (raw data) to corresponding labels. t: represents the training data utilized for training the machine learning model [11]. Equation (1) is a written equation to find the IHT value.

$$IH_h(\langle x_i, y_i \rangle) = 1 - p(y_i|x_i, h) \quad (1)$$

In short, IHT works by resampling an imbalanced dataset by eliminating data instances from the majority class that have high IH values. This elimination process is carried out until the desired balance ratio is reached [33].

2.5. Split Data

In machine learning studies, dividing data into a training set and a test set is a crucial step to ensure the model is well trained and produces accurate performance. This is done by dividing the data by certain proportions [34].

2.6. Stacking Ensemble Learning (SEL)

Stacked Generalization, commonly referred to as Stacking, was originally introduced by Wolpert in 1992 [35]. The primary goal of Stacking is to combine the prediction results from multiple classification models (called base classifiers) and then use another model (called a meta classifier) to process these prediction results to achieve a more optimal final result. The primary benefit of this approach lies in its capacity to incorporate diverse models utilizing different algorithms as base learners to construct an ensemble [17]. The prediction results of these base models are then combined using another classification technique, which is trained using the output from the ensemble members. This process produces a stacking model that is expected to perform better than the base models used [36]. For a clearer understanding, the Stacking Ensemble Learning architecture can be seen in Figure 2.

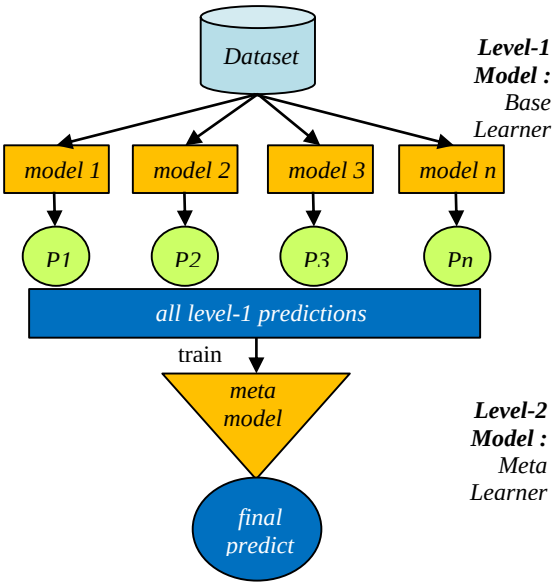

Figure 2. Stacking Ensemble Learning Architecture

In the context of this paper, the SEL approach utilizes three distinct classification models as base learners: Decision Tree, Random Forest, and XGBoost. These base learners are responsible for generating initial predictions, which are then fed into a meta learner, Logistic Regression, for further processing. The meta learner combines the predictions from the base learners to produce a final prediction that is expected to be more accurate than the individual predictions. The SEL approach [36] is particularly well-suited for scenarios where different models have varying strengths and weaknesses. By combining these models, The SEL approach is expected to harness the strengths of individual models to overcome their weaknesses and achieve overall better performance.

2.6.1. Decision Tree

The DT Classifier is a classification model composed of test nodes and decision nodes. The test nodes, starting from the root node, serve to divide the data into separate groups based on specific attributes. This division process continues recursively until the data can no longer be divided further [37]. Decision trees use information gain to choose the best feature for splitting data. Information gain measures how

well a feature separates data into different categories [38]. Information gain itself can be calculated mathematically as shown in Equation (2).

$$IG = Entropy\,(s) - [(WeightedAvg.) * Entropy\,(Every\ Feature)] \quad (2)$$

Entropy (S): Initial measure of uncertainty in the entire dataset. It is typically calculated using the proportion of each class present and can be computed using Equation (3). The more diverse the classes in the data, the higher the Entropy value. WeightedAvg.: Accounts for the proportion of data that falls into each branch after splitting. Entropy (Every Feature): Remaining uncertainty in each branch after splitting [38].

$$Entropy(s) = -P(yes)log_2\,P(yes) - P(no)log_2\,(no) \quad (3)$$

In the provided formula, S: Overall sample count in the dataset. This represents the total number of data points being considered. P(yes): Probability of obtaining the class "yes." This represents the distribution of data points within the dataset that belong to the "yes" class. P(no): Probability of obtaining the class "no." This denotes the share of data points in the dataset attributed to the "no" class.

2.6.2. Random Forest

Random Forest [39] is a powerful and popular non-parametric classification algorithm that builds multiple decision trees from different training data samples. Each tree is trained on a subset of data and features, reducing correlation between trees and improving prediction diversity. The final prediction is determined by majority voting from all decision trees. Random Forest excels in producing accurate and robust predictions due to its ensemble learning approach, random subsampling, and OOB error estimation. It is widely used in classification tasks, feature engineering, and anomaly detection.

For example [40] $\{h(x,\Theta_k), k = 1, \dots\}$ where $\{\Theta_k\}$ is an independent and identically distributed random vector and each tree then selects the class most represented in the data, based on the principle "most votes". Consider an ensemble $h_1(x), h_2(x), \dots, h_k(x)$ with training data chosen randomly from the distribution of random vectors Y and X. The margin function $(mg(X,Y))$ of Random Forest is defined as in Equation (4).

$$(mg(X,Y)) = \frac{\sum_{k=1}^{K} I(h_k(X)=Y)}{K} - \max_{j \neq Y}\left[\frac{\sum_{k=1}^{K} I(h_k(X)=j)}{K}\right] \quad (4)$$

In Equation (4), I represents the indicator function and K represents the number of trees. The margin function serves to measure the difference between the average votes for X and Y and the average votes from other classes. Strength is the average value (expectation) of a measure of the accuracy of a single tree. The greater the value of s, the better the prediction accuracy. The definition of the s value can be seen in Equation (5).

$$s = E_{X,F} mg(X,Y) \quad (5)$$

$$\bar{\rho} = \frac{E_{A.A'}(\rho(\theta,\theta')\mathrm{sd}(\theta)\mathrm{sd}(\theta'))}{E_{\theta.\theta'}(sd(\theta)\mathrm{sd}(\theta'))} \quad (6)$$

Equation (6) represents the correlation between two trees, where $\rho(\theta,\theta')$ denotes the correlation between trees with parameters θ and θ'. The terms $sd(\theta)$ and $sd(\theta')$ represent the standard deviations of the trees with parameters θ and θ', respectively. Equation (7) defines the prediction error of Random Forest in classifying new data points. It measures the average error across all trees in the ensemble.

$$\varepsilon\mathrm{RF} \leq \bar{\rho}\,(\frac{1-s^2}{s^2}) \quad (7)$$

A low value of εRF indicates that the Random Forest model is effectively predicting the labels of new data points. This implies that the ensemble of decision trees is working in harmony to produce accurate classifications. A high value of εRF suggests that the Random Forest model is struggling to accurately predict the labels of new data points. This may be due to factors such as overfitting, underfitting, or insufficient training data. Equation (7) highlights the interplay between correlation and strength in determining the prediction error of Random Forest. To achieve low error, a combination of low correlation and strong individual trees is desirable. This can be achieved by adjusting the values of parameters *m* and *ntree*. It is important to note that there is no single optimal value for m and ntree that applies to all datasets. The best values for these parameters need to be determined experimentally for each specific dataset. Additionally, Random Forest offers other tunable parameters, such as max_depth and min_samples_split, which can be adjusted to further optimize model performance.

2.6.3. XGBoost

XGBoost, an acronym for Extreme Gradient Boosting, is a robust machine learning technique that employs a combination of decision trees to achieve improved predictive accuracy [37]. It builds upon the Gradient Boosting algorithm, originally proposed by Friedman in 2001, and has emerged as a popular choice for various machine learning tasks, particularly in the realm of supervised learning.

The core principle of XGBoost are [25][37][41]: 1) Ensemble Learning: XGB builds an ensemble of decision trees, each sequentially improving on its predecessors, leveraging strengths while mitigating weaknesses. 2) Gradient Boosting: XGB uses Gradient Boosting, where each new tree corrects the

inaccuracies of the preceding ones, minimizing the overall loss function iteratively for a more accurate model. 3) Loss Function Optimization: XGBoost supports various loss functions, such as the squared error loss and the logistic loss, to optimize the training process for specific tasks, such as regression and classification, respectively.

The mathematical representation [25] of the XGBoost model is given by Equation (8):

$$\hat{y}_i = \Sigma_{k=1}^{K} f_k(x_i), f_k \in F \quad (8)$$

In this model, K describes the number of existing trees, y_i is the predicted value and f_k indicates the k_{th} tree. The predicted value is the total score projected by the K tree. Meanwhile F is the space in the regression tree / CART. To solve this problem, a set of functions is needed as in Equation (9) which is applied in the model by reducing loss and regularization.

$$obj\ (\theta) = \Sigma_{i=1}^{n} l(y_i, \hat{y}_i) + \Sigma_{k=1}^{K} \Omega(\ f_k) \quad (9)$$

In Equation (9) $l(y_i, \hat{y}_i)$ is a loss function, while $\Omega(\ f_k)$ is called the regularization term. Then Ω is used to help reduce overfitting in the model which can be calculated as in equation (10).

$$\Omega(\ f_k) = \Upsilon T + \frac{1}{2} \lambda\ ||W||^2 \quad (10)$$

In equation (10), T represents the total number of leaves on the tree. Meanwhile, W represents the weight per leaf that the tree has.

2.6.4. Logistic Regression

Logistic Regression predicts the probability of binary outcomes, contrasting with Linear Regression, which forecasts continuous numerical outcomes. It generates probability estimates between 0 and 1, indicating the likelihood of the event's occurrence [42].

The Core Principles of Logistic Regression [43][44]: 1) Binary Outcome Prediction: LR classifies data into two categories, useful for yes/no scenarios like classifying an email as spam or determining if a patient has a disease. 2) Probability Estimation: LR estimates the probability of each outcome, offering nuanced insights into the event's potential occurrence. 3) Logistic Function: LR uses the sigmoid function to transform the linear relationship between independent variables and the dependent variable into a probability distribution. 4) Model Optimization: LR uses optimization algorithms to minimize errors between predicted probabilities and actual outcomes.

The mathematical representation [44] of the LR model is given by Equation (11):

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m \quad (11)$$

In Equation (11), π (pi) represents the probability of an event occurring. β (beta) is the regression coefficient. Each independent variable (x_1 to x_m) has an associated regression coefficient (β_i). This coefficient shows how the independent variable affects the probability of the event. β_0 is a special regression coefficient related to the reference group. The reference group is the basic category used to compare the effects of other independent variables [42].

2.7. Evaluation Metrics

Analyzing the performance of machine learning models is fundamental for assessing their effectiveness and identifying areas for improvement. The Confusion Matrix [45] is a vital tool for summarizing and visualizing classification model performance, offering a detailed overview of predictions versus true labels, revealing both strengths and weaknesses. The Confusion Matrix is a square table with dimensions matching the number of classes in the classification task. Each cell represents a combination of actual and predicted class labels, as depicted in Figure 3.

		Actual Values	
		Positive	*Negative*
Predicted Values	*Positive*	TP (*True Positive*)	FP (*False Positive*)
	Negative	FN (*False Negative*)	TN (*True Negative*)

Figure 3. Confusion Matrix

As shown in Figure 3, the four main categories of cells in the Confusion Matrix are [46]:

1) True Positives (TP): Correctly classified positive cases, such as predicting a customer will subscribe and they actually do.
2) True Negatives (TN): Correctly classified negative cases, like predicting a customer won't subscribe, and they don't.
3) False Positives (FP): Incorrectly predicted positives, such as predicting a customer will subscribe, but they don't.
4) False Negatives (FN): Missed positives, like predicting a customer won't subscribe, but they do.

Several performance metrics can be acquired from the Confusion Matrix [47], such as *Accuracy, Precission, Recall, F1-Score* dan *AUC-ROC*.

2.7.1. Accuracy

Accuracy is the percentage of correct predictions made by the model, calculated by dividing the number of correct predictions by the total number of predictions. Mathematically, accuracy can be represented as Equation (12) :

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (12)$$

2.7.2. Precision

Precision is the percentage of correctly classified positive predictions, calculated by dividing the number of true positives by the total positive predictions. Mathematically, precision can be represented as Equation (13):

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

2.7.3. Recall

Recall is the percentage of true positives correctly identified by the model, calculated by dividing the number of true positives by the total actual positives. Mathematically, recall can be represented as Equation (14) :

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

2.7.4. F1-Score

The F1-Score, the harmonic mean of precision and recall, is crucial for evaluating classification models, balancing true positive identification and minimizing false positives. Mathematically, recall is represented as Equation (15) :

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (15)$$

2.7.5. AUC-ROC

The AUC-ROC represents the area under the ROC curve, which plots the true positive rate against the false positive rate across all classification thresholds.. Mathematically, AUC-ROC can be represented as Equation (16) :

$$AUC = \frac{1 - \sum(FPR[i+1] - FPR[i]) * (1 - TPR[i+1])}{2} \quad (16)$$

Where:
FPR[i] : False positive rate at point i
FPR[i+1] : False positive rate at point i+1
TPR[i] : True positive rate at point i
TPR[i+1] : True positive rate at point i+1.

3. RESULT

3.1. Data Collection

The publicly available Bank Marketing Dataset [4] downloaded from UCI Machine Learning Repository serves as the foundation for this research. It comprises 45,211 records, each featuring 16 attributes and a single class label, 'y'. The dataset stems from a direct marketing initiative executed by a Portuguese bank, where telemarketing was used to promote bank deposit products.

3.2. Data Understanding

A crucial aspect of the dataset is its imbalanced class distribution. The 'yes' class (minority class) represents only 5,289 instances, while the 'no' class (majority class) encompasses a significantly larger number of instances (39,922). Figure 4 presents a visual representation of the data distribution within the Bank Marketing Dataset.

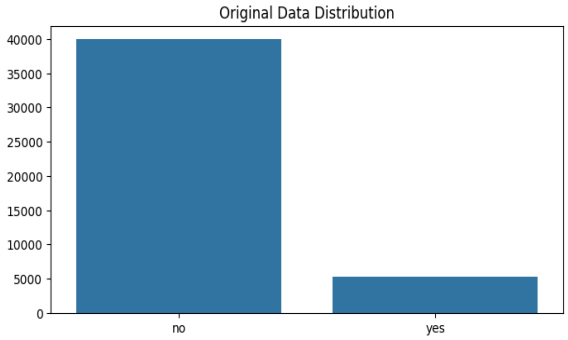

Figure 4. Original Data Distribution

The graphical representation in Figure 4 showcases the distribution of data classes in the Dataset. The 'no' class, representing customers who did not accept the term deposit product, dominates the dataset with a share of 88.3%. In contrast, the 'yes' class, representing customers who accepted the product, accounts for only 11.7%. This imbalanced class distribution necessitates careful consideration in the selection and evaluation of classification algorithms to ensure fair and accurate prediction for both classes. The dataset includes various demographic and financial attributes of bank customers, offering insights into factors influencing time deposit acceptance.

Table 1 provides a detailed breakdown of these attributes, while Figure 5 showcases a sample of data records, illustrating the dataset's structure and content.

Table 1. Attributes

Variable	Attribute	Data Type	Description
input	age	numeric	age of the customer in years
	job	categorical	type of job ('unknown', 'unemployed', 'technician','self-employed','student','retired', 'management','services','housemaid','blue-collar','entrepreneur','admin')
	marital	categorical	marital status ('unknown', 'single','married','divorced')
	education	categorical	educational level ('unknown', 'university.degree', 'professional.course', 'illiterate', 'high.school', 'basic.9y','basic.6y', 'basic.4y')
	default	categorical	is there any defaulted credit?
	balance	numeric	mean yearly balance
	housing	categorical	does the individual have a mortgage?

Variable	Attribute	Data Type	Description
	loan	categorical	does the individual have a personal loan?
	contact	categorical	preferred communication channel for reaching the contact ('telephone','cellular')
	day	numeric	weekday of last interaction
	month	categorical	month of last contact
	duration	numeric	last interaction duration (seconds)
	campaign	numeric	contact frequency for this client (current campaign)
	pday	numeric	days since last contact (previous campaign)
	previous	numeric	prior contact frequency for this client
	poutcome	categorical	result of prior campaign ('success', 'nonexistent','failure')
output	y	binary	has the client signup a time deposit? (yes or no)

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

45211 rows × 17 columns

Figure 5. Example of Record Data in Dataset

3.3. Data Preprocessing

3.3.1. Label Encoding

In the context of the Bank Marketing Dataset, label encoding can effectively transform categorical attributes like default, loan, housing, contact, and "y" into numerical representations. Table 2 shows the categories were transformed using label encoding.

Table 2. Example of Data Transformation Results with Label Encoding

Default	Housing	Loan	Contact	Y
0	1	0	2	0
0	1	1	2	0
0	0	0	2	0
0	0	1	0	1
0	1	0	1	1

3.3.2. Ordinal Encoding

Table 3 shows the categories like job, marital status, education, month, and outcome are commonly transformed using ordinal encoding.

Table 3. Example of Data Transformation Results with Ordinal Encoding

Job	Marital	Education	Month	Poutcome
4.0	1.0	2.0	8.0	3.0
0.0	1.0	1.0	11.0	0.0
11.0	2.0	3.0	8.0	3.0
7.0	1.0	1.0	9.0	1.0
9.0	2.0	1.0	8.0	3.0

3.3.3. Preprocessed Dataset

As depicted in Figure 6, the data preprocessing has successfully transformed categorical features into numerical representations. This was achieved using label encoding and ordinal encoding techniques. The features that have been converted to numerical are job, marital, education, default, housing, loan, contact, month, poutcome, and the label class y. This transformation is crucial to prepare the data for effective processing by machine learning algorithms, which typically operate on numerical data.

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	4.0	1.0	2.0	0	2143	1	0	2	5	8.0	261	1	-1	0	3.0	0
1	44	9.0	2.0	1.0	0	29	1	0	2	5	8.0	151	1	-1	0	3.0	0
2	33	2.0	1.0	1.0	0	2	1	1	2	5	8.0	76	1	-1	0	3.0	0
3	47	1.0	1.0	3.0	0	1506	1	0	2	5	8.0	92	1	-1	0	3.0	0
4	33	11.0	2.0	3.0	0	1	0	0	2	5	8.0	198	1	-1	0	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	9.0	1.0	2.0	0	825	0	0	0	17	9.0	977	3	-1	0	3.0	1
45207	71	5.0	0.0	0.0	0	1729	0	0	0	17	9.0	456	2	-1	0	3.0	1
45208	72	5.0	1.0	1.0	0	5715	0	0	0	17	9.0	1127	5	184	3	2.0	1
45209	57	1.0	1.0	1.0	0	668	0	0	1	17	9.0	508	4	-1	0	3.0	0
45210	37	2.0	1.0	1.0	0	2971	0	0	0	17	9.0	361	2	188	11	1.0	0

45211 rows × 17 columns

Figure 6. Illustration of A Preprocessed Data Record in The Dataset

3.4. Classification on Dataset without Undersampling

In the initial phase of the study, a series of comprehensive experiments were performed on the bank-full.csv dataset without applying any undersampling techniques. The experiments employed various individual classification algorithms, including Decision Tree, Random Forest, and XGBoost. Additionally, the study utilized the advanced Stacking Ensemble Learning method, which combined these individual algorithms—Decision Tree, Random Forest, and XGBoost—as base classifiers. This ensemble was further enhanced by incorporating Logistic Regression as the meta-classifier to improve the overall predictive performance. Following these setups, the dataset was meticulously divided into training and testing subsets, with 80% of the data allocated for training and the remaining 20% for testing, as detailed in Table 4.

Table 4. Split Data Distribution without Undersampling

Label	Train Data	Test Data
0	31.970	7.952
1	4.198	1.091
Amount	36.168	9.043

Table 4 illustrates the distribution of training data, consisting of 36,168 instances, with 31,970 records classified as class 0 and 4,198 records classified as class 1. Additionally, the testing data comprises 9,043 instances, with 7,952 records classified as class 0 and 1,091 records classified as class 1.

Figure 7 depicts the Confusion Matrix results of the four method scenarios, which were employed as the basis for determining Accuracy, Precision, Recall, and F1-Score values.

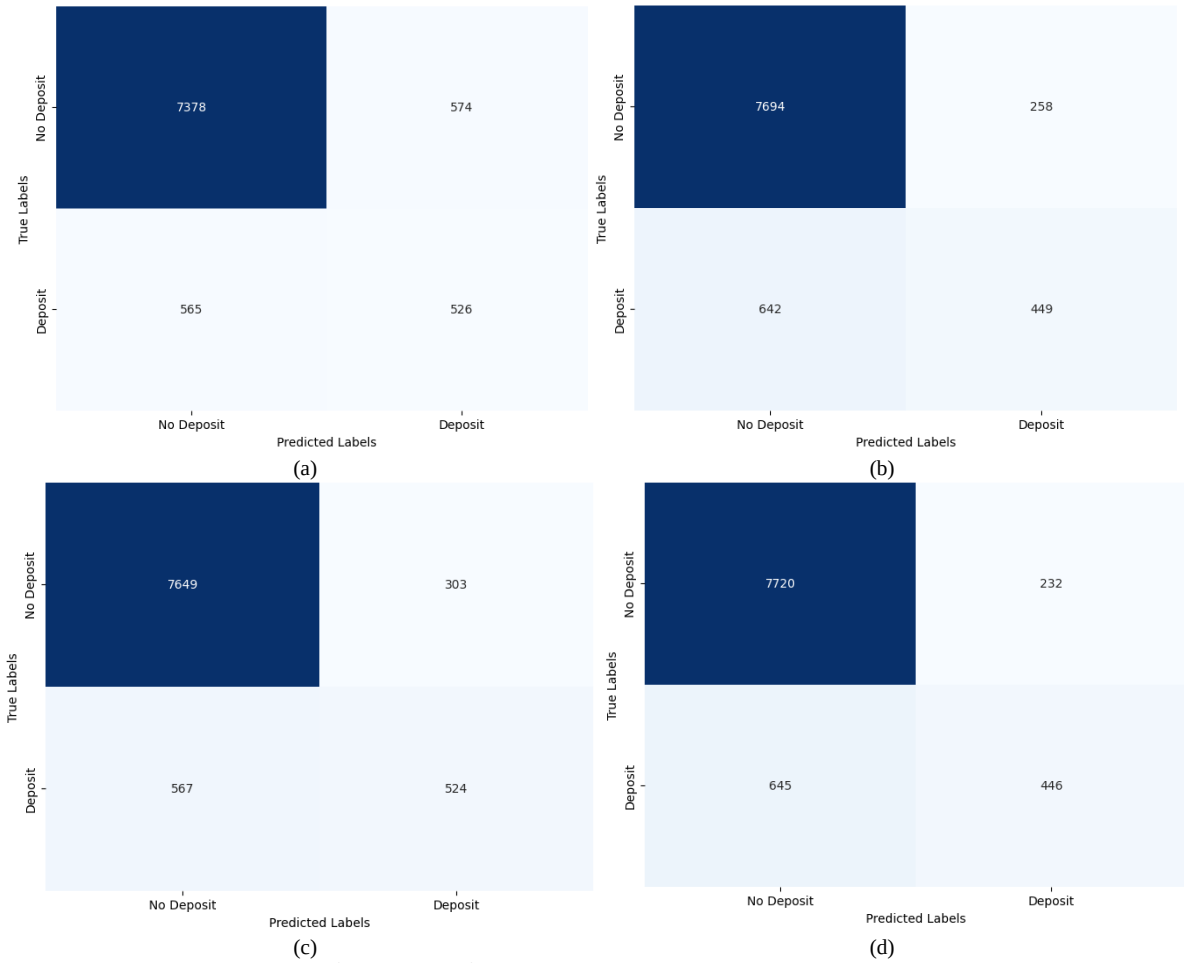

Figure 7. Confusion Matrix of (a) DT, (b) RF, (c) XGB, (d) SEL without Undersampling

Figure 7(a) depicts the performance of the Decision Tree algorithm. For class 0 (No), the algorithm accurately classified 7,378 out of 7,952 data points, with 574 misclassified as class 1 (Yes). For class 1 (Yes), it correctly classified 526 out of 1,091 data points, while 565 were misclassified as class 0 (No). Figure 7(b) shows the Random Forest algorithm's results, where 7,694 out of 7,952 class 0 (No) instances were correctly classified, and 258 were misclassified as class 1 (Yes). For class 1 (Yes), 449 out of 1,091 instances were correctly classified, with 642 misclassified as class 0 (No). According to Figure 7(c), the XGBoost algorithm correctly classified 7,649 out of 7,952 class 0 (No) instances,

with 303 misclassified as class 1 (Yes). For class 1 (Yes), 524 out of 1,091 instances were correctly classified, while 567 were misclassified as class 0 (No). Finally, Figure 7(d) shows the performance of the SEL algorithm, with 7,720 out of 7,952 class 0 (No) instances correctly classified, and 232 misclassified as class 1 (Yes). For class 1 (Yes), 446 out of 1,091 instances were correctly classified, with 645 misclassified as class 0 (No).

Based on the confusion matrix results in Figure 7, the accuracy, precision, recall, and F1-score values were calculated and are shown in Table 5. These metrics comprehensively assess the algorithms' classification performance.

Table 5. Experiment Results Without Undersampling

Method	Accuracy	Precision	Recall	F1-Score
DT	87,40	87,45	87,40	87,43
RF	90,05	88,82	90,05	89,10
XGB	**90,38**	**89,51**	**90,38**	**89,80**
SEL	90,30	89,09	90,30	89,29

As seen in Table 5, the performance of each method wherein Decision Tree achieved an Accuracy of 87.40%, Precision of 87.45%, Recall of 87.40%, and F1-Score of 87.43%. Random Forest attained an Accuracy of 90.05%, Precision of 88.82%, Recall of 90.05%, and F1-Score of 89.10%. XGBoost yielded an Accuracy of 90.38%, Precision of 89.51%, Recall of 90.38%, and F1-Score of 89.80%. The SEL approach obtained an Accuracy of 90.30%, Precision of 89.09%, Recall of 90.30%, and F1-Score of 89.29%. This experiment reveals that classification using the XGBoost single algorithm outperforms the other two single algorithms, namely DT and RF, in all four measured metrics. Furthermore, it even outperforms the SEL approach, which utilizes all three algorithms as base classifiers and for the meta classifier, LR was applied. In addition to these four metrics, the evaluation process also incorporated AUC-ROC analysis, as visualized in Figure 8. These results highlight the effectiveness of XGB in handling this classification task.

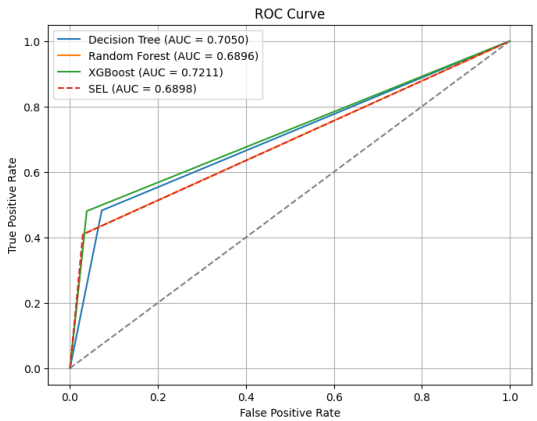

Figure 8. AUC-ROC on Original Dataset

Figure 8 shows that XGBoost again got the highest score with an AUC-ROC of **0.7211**, followed by Decision Tree of 0.7050, SEL of 0.6898 and Random Forest 0.6896.

3.5. Classification on Resampled Dataset with IHT

In the second stage, experiments were conducted on the bank-full.csv dataset, which had undergone undersampling using Instance Hardness Threshold (IHT). The classification algorithms employed remained consistent, utilizing single models such as DT, RF, and XGB. Additionally, SEL was implemented, incorporating DT, RF, and XGB as base classifiers, with Logistic Regression serving as the meta-classifier.

In Figure 9, we can see the distribution of data that has been resampled using IHT, where the number of majority (no) classes which originally reached 39,922 data records has become 15,070 data records.

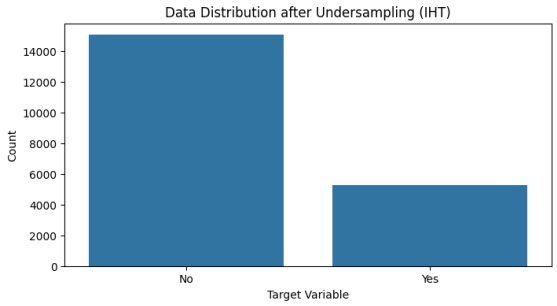

Figure 9. Data Distribution after Undersampling with IHT

The dataset was partitioned into two subsets, with 80% allocated for training and the remaining 20% reserved for testing. The detailed distributions of these subsets are illustrated in Table 6, ensuring clarity in data allocation and supporting the reproducibility of the experimental results.

Table 6. Split Data Distribution with IHT Undersampling

Label	Train Data	Test Data
0	12.062	3.008
1	4.225	1.064
Amount	16.287	4.072

In Table 6, the distribution of training data can be observed, comprising 16,287 instances, with 12,062 records classified as class 0 and 4,225 records classified as class 1. Additionally, the testing data consists of 4,072 instances, with 3,008 records classified as class 0 and 1,064 records classified as class 1.

Figure 8 illustrates the Confusion Matrix results of the four algorithm scenarios utilized as the basis for determining Accuracy, Precision, Recall, and F1-Score values.

Figure 8(a) displays the performance results of the Decision Tree algorithm. For class 0 (No), the algorithm accurately classified 2,956 out of 3,008 instances, misclassifying 52 as class 1 (Yes). In the case of class 1 (Yes), it correctly classified 1,001 out of 1,064 instances, with 63 misclassified as class 0 (No). Moving to Figure 8(b), the Random Forest

algorithm's results are detailed as follows: out of 3,008 class 0 (No) instances, 2,994 were correctly classified, and 14 were misclassified as class 1 (Yes). For class 1 (Yes), it correctly classified 1,006 out of 1,064 instances, with 58 misclassified as class 0 (No). In Figure 8(c), the XGBoost algorithm's performance shows that out of 3,008 class 0 (No) instances, 2,994 were correctly classified, and 14 were misclassified as class 1 (Yes). For class 1 (Yes), 1,019 out of 1,064 instances were correctly classified, with 45 misclassified as class 0 (No). Finally, Figure 8(d) illustrates the SEL algorithm's results: out of 3,008 class 0 (No) instances, 2,991 were correctly classified, with 17 misclassified as class 1 (Yes). For class 1 (Yes), 1,032 out of 1,064 instances were correctly classified, with 32 misclassified as class 0 (No).

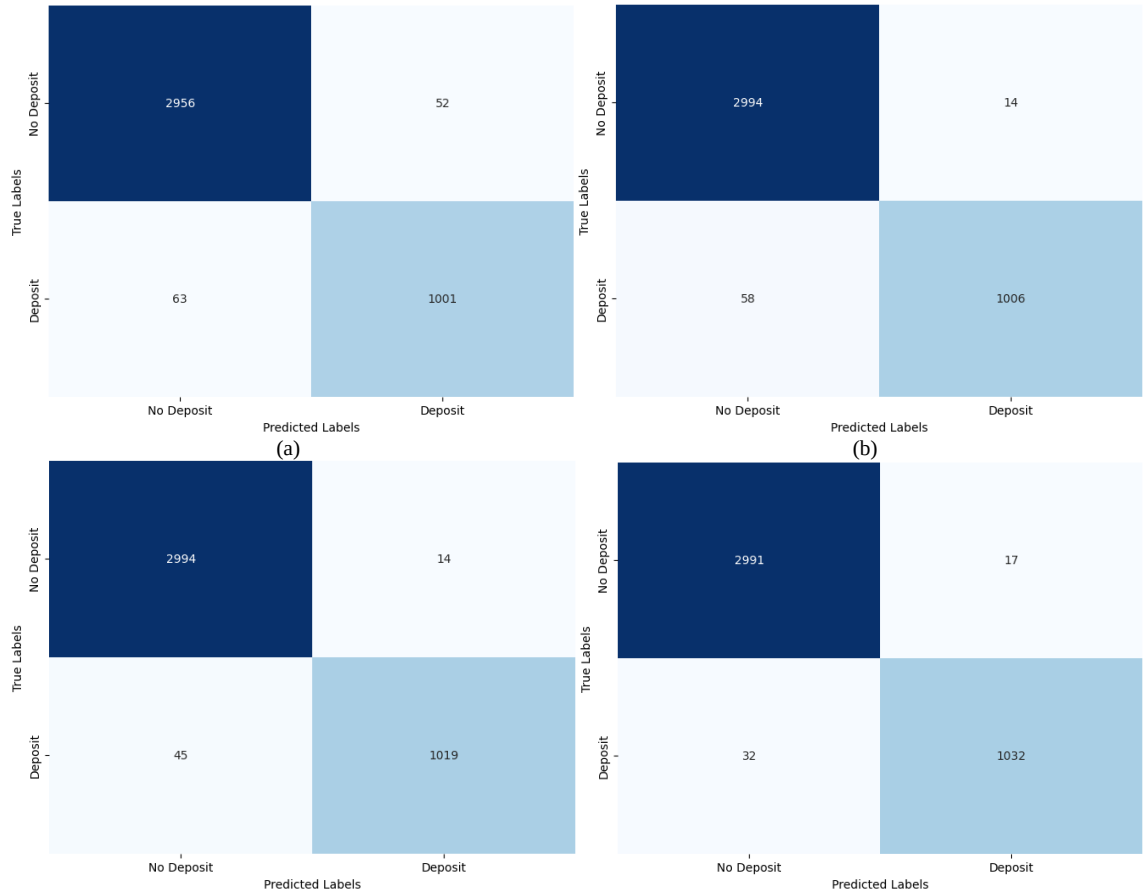

Figure 8. Confusion Matrix of (a) DT, (b) RF, (c) XGB, (d) SEL with IHT Undersampling

By analyzing the confusion matrix depicted in Figure 8, we can derive various evaluation metrics, which are elaborated upon and presented in Table 7. These metrics provide insights into the performance of the classification algorithms, aiding in the assessment of their effectiveness in accurately predicting class labels.

Table 7. Experimental Results on Resampled Dataset

Method	Accuracy	Precision	Recall	F1-Score
DT	97,18	97,17	97,18	97,17
RF	98,23	98,24	98,23	98,22
XGB	98,55	98,55	98,55	98,54
SEL	**98,80**	**98,79**	**98,80**	**98,79**

In Table 7, the performance of each algorithm can be observed, with Decision Tree achieving an Accuracy of 97.18%, Precision of 97.17%, Recall of 97.18%, and F1-Score of 97.17%. Random Forest achieved an Accuracy of 98.23%, Precision of 98.34%, Recall of 98.55%, and F1-Score of 98.54%. XGBoost attained an Accuracy of 98.55%, Precision of 98.55%, Recall of 98.55%, and F1-Score of 98.54%. SEL obtained an Accuracy of 98.80%, Precision of 98.79%, Recall of 98.80%, and F1-Score of 98.79%. This experiment demonstrates that classification using SEL yields the best results in terms of Accuracy, Precision, Recall, and F1-Score.

Additionally, AUC-ROC analysis, shown in Figure 9, complements these metrics, offering insights into the model's performance by depicting the trade-offs between sensitivity and specificity across various threshold values, emphasizing the algorithms' ability to distinguish between classes.

In Figure 9, it can be observed that SEL achieved the highest AUC-ROC level of 0.9821, followed by XGBoost with 0.9765, Random Forest with 0.9704, and Decision Tree with 0.9618. This trend aligns with the findings from the other

performance metrics, further solidifying the superior performance of SEL.

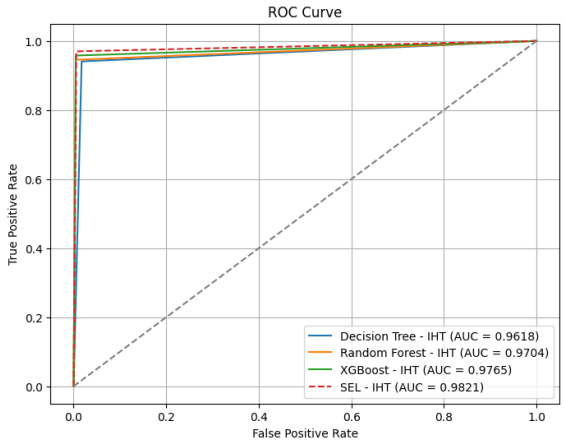

Figure 9. AUC-ROC on Resampling Dataset

4. DISCUSION

This study found that combining SEL and IHT undersampling techniques improved term deposit acceptance classification accuracy in the imbalanced Bank Marketing Dataset to 98.80%, with an AUC-ROC of 0.9821. While the impact of SEL on classification performance in this study is relatively small, these results are consistent with several studies in other classification cases, such as those conducted by Divina et al., Baccouche et al., and Almulihi et al., which indicate that ensemble learning can yield better classification performance than single classification models [17][18][21]. In this study, applying IHT undersampling had the most significant and effective impact on addressing class imbalance, leading to an approximately 8% improvement in classification model performance compared to without IHT undersampling.

Compared to several similar studies utilizing the same dataset, this research achieves higher accuracy levels. Table 8 compares the performance of SEL-IHT with the optimal performance gained from previous studies using the same dataset. The studies by Zaki et al. [48], Fitriani and Febrianto [49], Nugroho and Religia [10], Religia et al. [9] and Alsolami et al. [11] achieved their highest accuracy rates of 87.50%, 92.61%, 89.73%, 88.30%, and 91.48%, respectively, using RF Classifier, RF + SMOTE, GA + Naïve Bayes + Bagging, GA + RF + Bagging, and LR. None of these five studies utilized IHT undersampling and SEL techniques as employed in this study.

Table 8. Comparison with Previous Research on the Same Dataset

Author	Accuracy	Best Method	Year
Zaki et al. [48]	87,50%	RF Classifier	2023
Fitriani & Febrianto [49]	92,61%	RF + Smote	2021
Nugroho & Religia [10]	89,73%	GA + NB + Bagging	2021
Religia, et al. [9]	88,30%	GA + RF + Bagging	2021
Alsolami et al. [11]	91,48%	Logistic Regression	2020
SEL + IHT APPROACH	**98,80%**	**SEL + IHT**	**2024**

Aside from the accuracy advantages mentioned, this study also has several limitations. Firstly, it only utilizes one dataset. Secondly, it does not compare the IHT undersampling technique with other undersampling techniques. Thirdly, it does not investigate the impact of parameters on model performance. Future research can address these limitations by using datasets from different banks, comparing the IHT undersampling technique with other undersampling techniques, investigating the impact of parameters on model performance, performing feature selection, and exploring alternative algorithm combinations in ensemble learning architectures to achieve more accurate and comprehensive results. Additionally, considering the dynamic nature of financial data, longitudinal studies may provide valuable insights into the stability and generalizability of the proposed model over time.

5. CONCLUSION

This research investigates techniques to enhance the classification performance of term deposit acceptance on imbalanced datasets. Such classification is crucial for banks, aiming to optimize term deposit marketing strategies. The study compares single classification models (Decision Tree, Random Forest, and XGBoost) with Stacking Ensemble Learning (SEL), which integrates these models as base classifiers and and for the meta classifier, Logistic Regression was chosen. Undersampling Instance Hardness Threshold (IHT) is applied to address class imbalance in the dataset.

The findings indicate that SEL with IHT undersampling significantly improves classification accuracy compared to models without undersampling. In datasets without undersampling, XGBoost achieves the highest accuracy among single classification models. However, in datasets with undersampling, SEL surpasses all other models in terms of Accuracy, Precision, Recall, F1-Score, and AUC-ROC. These findings suggest that implementing SEL with IHT undersampling can yield superior models for term deposit acceptance classification on imbalanced datasets.

Future research may explore the generalizability of these findings using different datasets. Moreover, other undersampling techniques and ensemble learning methods could be investigated to determine if they can achieve better performance. Lastly, this study opens avenues to examine move forward classification method such as deep learning to further enhance classification accuracy. With further

research, it is anticipated that banks can leverage the most accurate classification models to assist in marketing term deposit products to potentially more prospective customers.

REFERENCES

[1] N. L. Muliawati and T. Maryati, "Analaisis Pengaruh Inflasi, Kurs, Suku Bunga dan Bagi Hasil Terhadap Deposito Pada PT. Bank Syariah Mandiri 2007-2012," *Semin. Nas. Cendekiawan*, no. 7, pp. 735–745, 2015.

[2] E. E. Ene, S. Atong, and J. C. Ene, "Effect of Interest Rates Deregulation on the Performance of Deposit Money Banks in Nigeria," *Int. J. Manag. Stud. Res.*, vol. 3, no. 9, pp. 164–176, 2015, [Online]. Available: www.arcjournals.org.

[3] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016, doi: 10.1007/s13748-016-0094-0.

[4] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decis. Support Syst.*, vol. 62, pp. 22–31, 2014, doi: 10.1016/j.dss.2014.03.001.

[5] G. Guo, Y. Yao, L. Liu, and T. Shen, "A novel ensemble approach for estimating the competency of bank telemarketing," *Sci. Rep.*, vol. 13, no. 1, pp. 1–10, 2023, doi: 10.1038/s41598-023-47177-7.

[6] N. Ghatasheh, I. Altaharwa, and K. Aldebei, "Modeling the Telemarketing Process Using Genetic Algorithms and Extreme Boosting: Feature Selection and Cost-Sensitive Analytical Approach," *IEEE Access*, vol. 11, no. June, pp. 67806–67824, 2023, doi: 10.1109/ACCESS.2023.3292840.

[7] I. M. Hayder, G. A. N. Al Ali, and H. A. Younis, "Predicting reaction based on customer's transaction using machine learning approaches," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 1, pp. 1086–1096, 2023, doi: 10.11591/ijece.v13i1.pp1086-1096.

[8] Y. Feng, Y. Yin, D. Wang, and L. Dhamotharan, "A dynamic ensemble selection method for bank telemarketing sales prediction," *J. Bus. Res.*, vol. 139, pp. 368–382, 2022, doi: 10.1016/j.jbusres.2021.09.067.

[9] Y. Religia, A. Nugroho, and W. Hadikristanto, "Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk Klasifikasi Data Bank Marketing," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 187–192, 2021.

[10] A. Nugroho and Y. Religia, "Analisis Optimasi Algoritma Klasifikasi Naive Bayes menggunakan Genetic Algorithm dan Bagging," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 3, pp. 504–510, 2021, doi: 10.29207/resti.v5i3.3067.

[11] F. J. Alsolami, F. Saleem, and A. Al-Malaise Al-Ghamdi, "Predicting the Accuracy for Telemarketing Process in Banks Using Data Mining," *JKAU Comp. IT. Sci*, vol. 9, no. 2, pp. 69–83, 2020, doi: 10.4197/Comp.

[12] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Mach. Learn.*, vol. 95, no. 2, pp. 225–256, 2014, doi: 10.1007/s10994-013-5422-z.

[13] N. A. Verdikha, T. B. Adji, and A. E. Permanasari, "Study of Undersampling Method: Instance Hardness Threshold with Various Estimators for Hate Speech Classification," *IJITEE (International J. Inf. Technol. Electr. Eng.*, vol. 2, no. 2, 2018, doi: 10.22146/ijitee.42152.

[14] T. G. Dietterichl, "The handbook of brain theory and neural networks-ensemble learning," *MIT Press*, vol. 40, 2002, [Online]. Available: https://courses.cs.washington.edu/courses/cse446/12wi/tgd-ensembles.pdf.

[15] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Front. Comput. Sci.*, vol. 14, no. 2, pp. 241–258, 2020, doi: 10.1007/s11704-019-8208-z.

[16] U. R. Salunkhe and S. N. Mali, "Classifier Ensemble Design for Imbalanced Data Classification: A Hybrid Approach," *Procedia Comput. Sci.*, vol. 85, no. Cms, pp. 725–732, 2016, doi: 10.1016/j.procs.2016.05.259.

[17] F. Divina, A. Gilson, F. Goméz-Vela, M. G. Torres, and J. F. Torres, "Stacking ensemble learning for short-term electricity consumption forecasting," *Energies*, vol. 11, no. 4, pp. 1–31, 2018, doi: 10.3390/en11040949.

[18] A. Baccouche, B. Garcia-Zapirain, C. C. Olea, and A. Elmaghraby, "Ensemble deep learning models for heart disease classification: A case study from Mexico," *Inf.*, vol. 11, no. 4, pp. 1–28, 2020, doi: 10.3390/INFO11040207.

[19] M. H. D. M. Ribeiro and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Appl. Soft Comput. J.*, vol. 86, p. 105837, 2020, doi: 10.1016/j.asoc.2019.105837.

[20] R. Lazzarini, H. Tianfield, and V. Charissis, "A stacking ensemble of deep learning

models for IoT intrusion detection," *Knowledge-Based Syst.*, vol. 279, p. 110941, 2023, doi: 10.1016/j.knosys.2023.110941.

[21] A. Almulihi *et al.*, "Ensemble Learning Based on Hybrid Deep Learning Model for Heart Disease Early Prediction," *Diagnostics*, vol. 12, no. 12, pp. 1–17, 2022, doi: 10.3390/diagnostics12123215.

[22] M. AlJame, I. Ahmad, A. Imtiaz, and A. Mohammed, "Ensemble learning model for diagnosing COVID-19 from routine blood tests," *Informatics Med. Unlocked*, vol. 21, p. 100449, 2020, doi: 10.1016/j.imu.2020. 100449.

[23] E. Demirovic and P. J. Stuckey, "Optimal Decision Trees for Nonlinear Metrics," *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 5A, pp. 3733–3741, 2021, doi: 10.1609/aaai.v35i5.16490.

[24] S. Ren, X. Cao, Y. Wei, and J. Sun, "Global refinement of random forest," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 723–730, 2015, doi: 10.1109/CVPR.2015.7298672.

[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[26] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic Regression Model Optimization and Case Analysis," *Proc. IEEE 7th Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2019*, pp. 135–139, 2019, doi: 10.1109/ICCSNT47585.2019.8962457.

[27] Y. Zizi, M. Oudgou, and A. El Moudden, "Determinants and predictors of smes' financial failure: A logistic regression approach," *Risks*, vol. 8, no. 4, pp. 1–21, 2020, doi: 10.3390/risks8040107.

[28] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, 2016, doi: 10.1186/s13634-016-0355-x.

[29] P. Cerda and G. Varoquaux, "Encoding High-Cardinality String Categorical Variables," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1164–1176, 2022, doi: 10.1109/TKDE.2020.2992529.

[30] M. X. Low *et al.*, "Comparison of Label Encoding and Evidence Counting for Malware Classification," *J. Syst. Manag. Sci.*, vol. 12, no. 6, pp. 17–30, 2022, doi: 10.33168/JSMS.2022.0602.

[31] F. Reusser, "Tabular Learning: Encoding for Entity and Context Embeddings," pp. 1–13, 2024, [Online]. Available: http://arxiv.org/abs/2403.19405.

[32] K. Takayama, "Encoding Categorical Variables with Ambiguity," *Int. Work. NFMCP conjunction with ECML-PKDD*, 2019, [Online]. Available: http://contrib.scikit-learn.org/categorical-encoding/.

[33] T. Le, L. H. Son, M. T. Vo, M. Y. Lee, and S. W. Baik, "SS symmetry A Cluster-Based Boosting Algorithm for Bankruptcy Prediction in a Highly Imbalanced Dataset," pp. 1–12, 2018, doi: 10.3390/sym10070250.

[34] R. Medar, V. S. Rajpurohit, and B. Rashmi, "Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning," *2017 Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2017*, pp. 1–6, 2017, doi: 10.1109/ICCUBEA.2017.8463779.

[35] D. Wolpert, "Stacked Generalization (Stacking)," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[36] R. Polikar, *Ensemble Machine Learning*. 2012.

[37] O. Sagi and L. Rokach, "Approximating XGBoost with an interpretable decision tree," *Inf. Sci. (Ny).*, vol. 572, pp. 522–542, 2021, doi: 10.1016/j.ins.2021.05.055.

[38] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decis. Anal. J.*, vol. 3, no. May, p. 100071, 2022, doi: 10.1016/j.dajour.2022.100071.

[39] L. Khaidem, S. Saha, and S. R. Dey, "Predicting the direction of stock market prices using random forest," vol. 00, no. 00, pp. 1–20, 2016, [Online]. Available: http://arxiv.org/abs/1605.00003.

[40] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[41] M. Guo, Z. Yuan, B. Janson, Y. Peng, Y. Yang, and W. Wang, "Older pedestrian traffic crashes severity analysis based on an emerging machine learning xgboost," *Sustain.*, vol. 13, no. 2, pp. 1–26, 2021, doi: 10.3390/su13020926.

[42] J. Phillips, E. Cripps, J. W. Lau, and M. R. Hodkiewicz, "Classifying machinery condition using oil samples and binary logistic regression," *Mech. Syst. Signal Process.*, vol. 60, pp. 316–325, 2015, doi: 10.1016/j.ymssp.2014.12.020.

[43] T. Denœux, "Logistic regression, neural networks and Dempster–Shafer theory: A

new perspective," *Knowledge-Based Syst.*, vol. 176, pp. 54–67, 2019, doi: 10.1016/j.knosys.2019.03.030.

[44] S. Sperandei, "Understanding logistic regression analysis," *Biochem. Medica*, vol. 24, no. 1, pp. 12–18, 2014, doi: 10.11613/BM.2014.003.

[45] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," *Commun. Comput. Inf. Sci.*, vol. 51, pp. 461–471, 2009, doi: 10.1007/978-3-642-04962-0_53.

[46] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci. (Ny).*, vol. 513, pp. 429–441, 2020, doi: 10.1016/j.ins.2019.11.004.

[47] R. R. Sanni and H. S. Guruprasad, "Analysis of performance metrics of heart failured patients using Python and machine learning algorithms," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 233–237, 2021, doi: 10.1016/j.gltp.2021.08.028.

[48] A. M. Zaki, N. Khodadadi, W. H. Lim, and S. K. Towfek, "Predictive Analytics and Machine Learning in Direct Marketing for Anticipating Bank Term Deposit Subscriptions," *Am. J. Bus. Oper. Res.*, vol. 11, no. 1, pp. 79–88, 2024, doi: 10.54216/ajbor.110110.

[49] M. A. Fitriani and D. C. Febrianto, "Data Mining for Potential Customer Segmentation in the Marketing Bank Dataset," *JUITA J. Inform.*, vol. 9, no. 1, p. 25, 2021, doi: 10.30595/juita.v9i1.7983.