STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–17.
Published online in International Academic Press (www.IAPress.org)

Performance of Machine Learning Algorithms for Credit Risk Prediction with Feature Selection

Muhammad M. Seliem [1,*], Muhammad Amin [2], Mona Mahmoud Abo El Nasr [3], Emad Abdelaziz Elnaggar [4], Hany Abdelmonem Mohamed Khalifa [5], Mona Ahmed Abdelwahab Arab [6]

[1]*Department of Applied Statistics and Econometrics, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza , Egypt*
[2]*Department of Statistics, University of Sargodha, Sargodha, Pakistan (muhammad.amin@uos.edu.pk)*
[3]*Department of Management Information Systems, College of Business Administration in Yanbu, Taibah University, Madinah, Saudi Arabia (mmaboelnasr@taibahu.edu.sa)*
[4]*College of Business Administration, Afif - Shaqra University (emadelnaggar@su.edu.sa)*
[5]*College of Science and Humanities, Al- Dawadmi- Shaqra University (hkhalifa@su.edu.sa)*
[6]*Department of Management, Faculty of Managerial Sciences, Sadat Academy for Management Sciences, Egypt (Mona.Arab@sadatacademy.edu.eg)*

Abstract Financial institutions increasingly rely on machine learning (ML) models to assess credit risk and make lending decisions. Accurate prediction hinges on effective feature selection, which can significantly enhance model performance. This paper investigates the efficacy of seven supervised ML algorithms in predicting credit risk: Naive Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbor, Artificial Neural Network, Random Forest, and Logistic Regression. Using a German credit dataset comprising 1000 observations with 20 explanatory variables, we evaluated model performance using accuracy, kappa statistic, and F1 score. Two data-splitting scenarios (70-30% and 80-20%) were employed to assess robustness. We addressed outliers through imputation methods to optimize model performance and applied the Boruta algorithm for feature selection, which identified and eliminated six non-contributing features. Our findings consistently demonstrate the superiority of the Random Forest algorithm across both scenarios. Regarding accuracy, Random Forest achieved 77.3% in the 70-30% split and 80% in the 80-20% split, outperforming all other methods. These results underscore the potential of Random Forest as a valuable tool for credit risk assessment in financial institutions.

Keywords Credit Risk Assessment, Feature Selection, Classification Algorithms, Kappa statistic and Accuracy

Mathematics Subject Classification: 62E10, 60K10, 60N05

DOI: 10.19139/soic-2310-5070-2392

1. Introduction

Machine learning (ML) models, a subset of artificial intelligence (AI), are versatile tools for solving **various problems**, including classification tasks. In credit risk assessment, classification models are employed to predict the likelihood of a borrower defaulting on their loan. This information enables financial institutions to make informed lending decisions. This research provides a comprehensive review of ML algorithms applied to German credit data, focusing on their efficacy in predicting credit risk. Table (1) summarizes key studies in this area, highlighting the diverse methodologies and performance metrics employed. By understanding the state-of-the-art in ML for credit risk prediction, financial institutions can leverage these techniques to enhance their risk

*Correspondence to: Muhammad M. Seliem (Email: Seliem.m.m@hims.edu.eg). Department of Applied Statistics and Econometrics, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza , Egypt.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

management strategies and improve lending outcomes.

Table 1. A Literature Review of German Credit Data.

References	Author	Algorithmes
1	Pławiak et al. (2020)	KNN, PNN and SVM
2	Zhang et al. (2018)	KNN, RF, and SVM
3	Arora and Kaur (2020)	NB, SVM, RF, and KNN
4	Nalić et al. (2020)	NB, DT, GLM, and SVM
5	Saheed et al. (2020)	NB, RF, and SVM
6	Religia et al. (2020)	RF
7	Imron and Prasetyo (2020)	KNN
8	Mardiansyah et al. (2021)	LR, RF, and SVM
9	Arun and Venkatachalapathy (2020)	LR, DT, and SVM
10	Metawa et al. (2021)	SVM
11	Yang et al. (2021)	KNN, SVM, and LR
12	Trivedi (2020)	NB, RF, and SVM
13	Shi et al. (2022)	LR, SVM, ANN, RF, KNN and NB
14	Shen et al. (2021)	LR, KNN, NB, SVM and ANN

Data normalization rescales feature values to a standardized range of 0 to 1. This is achieved by transforming each value into its z-score, calculated using the mean and standard deviation of the feature. By standardizing the data, we ensure that all features contribute equally to the model, preventing features with larger magnitudes from dominating the learning process and improving overall model performance. Each value in a variable is replaced by its z-value; which is expressed by

$$z_{\text{norm}} = \frac{x - \bar{x}}{S_{\text{x}}} \tag{1}$$

where S_{x} is the standard deviation. We can concentrate on taking care of missing values, and outliers data. Figure 1 outlines the proposed methodology, which is structured into the following steps:

- **Step 1: Data Preprocessing:** The raw dataset is acquired from the UCI ML repository. It undergoes comprehensive preprocessing, including handling missing values and outliers using the treatment of outlier data as missing values by applying imputation methods (**TOMI**) technique, normalizing features via techniques such as Min-Max scaling or standardization (as shown in equation 1), and encoding categorical variables using one-hot or label encoding to ensure compatibility with ML algorithms.
- **Step 2: Data Partitioning:** The refined dataset is partitioned into training and testing sets using two distinct strategies: a 70-30% split for Case 1 and an 80-20% split for Case 2. These splits are stratified to maintain class distribution and randomized with a fixed seed to ensure reproducibility.
- **Step 3: Model Construction:** A diverse set of algorithms is employed to construct predictive models, including Naive Bayes (NB) for baseline performance, Support Vector Machine (SVM) for high-dimensional spaces, Decision Tree (DT) for interpretability, K-Nearest Neighbors (KNN) for instance-based learning, Logistic Regression (LR) as a linear benchmark, Random Forest (RF) for ensemble-based robustness, and Artificial Neural Network (ANN) to capture complex patterns. Hyperparameters are tuned using grid search, and models are implemented using R libraries.
- **Step 4: Performance Evaluation:** Model performance is rigorously evaluated using metrics such as Accuracy (ACC), F1 Score, Precision, and Kappa statistics, with validation conducted via 10-fold cross-validation.

- **Step 5: Final Model Selection:** The top-performing algorithm, selected based on aggregated metrics, is applied to the testing dataset to generate the final predictive model. This methodological framework ensures transparency, reproducibility, and alignment with best practices in ML research.

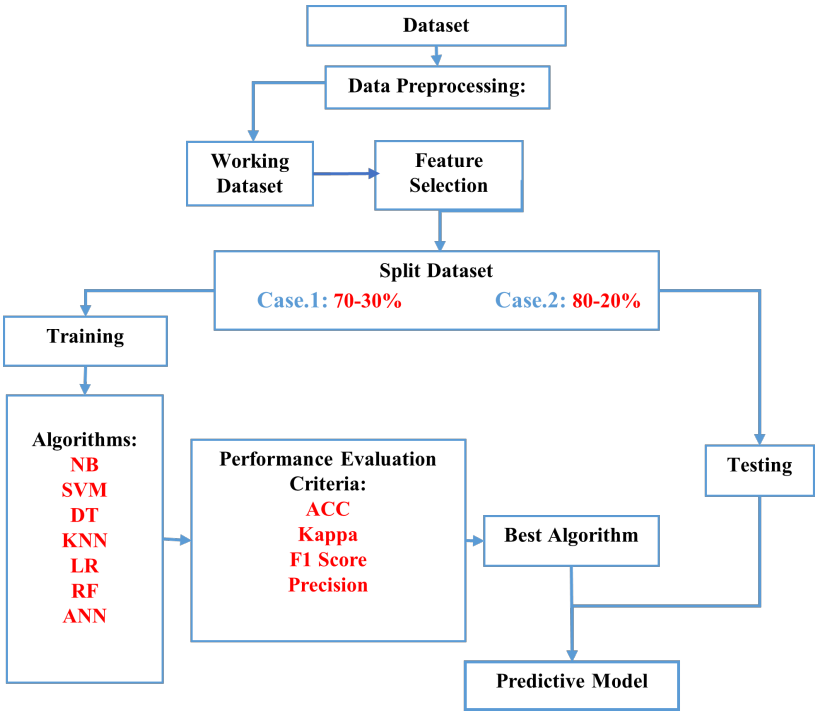

Figure 1. Methodology of the research work.

The literature review highlights the effectiveness of ML algorithms like SVM, RF, KNN, and NB in credit risk assessment. Hybrid and ensemble models, such as the deep genetic hierarchical network proposed by Pławiak et al. (2020) and the Bolasso-based feature selection method by Arora and Kaur (2020), outperform traditional approaches by addressing challenges like imbalanced datasets and high-dimensional data. Interpretable models (e.g., LR, DT) combined with advanced techniques like Social Spider Optimization Arun and Venkatachalapathy (2020) and SMOTE-XGBoost Mardiansyah et al. (2021) remain prominent. Advanced methods, including deep learning (e.g., Deep Belief Networks by Metawa et al. (2021)) and evolutionary algorithms (e.g., neural architecture search by Yang et al. (2021)), are increasingly applied, showcasing their potential to tackle complex credit risk challenges. Overall, integrating ML with feature selection, ensemble learning, and deep learning significantly enhances prediction accuracy and robustness. Seliem (2022) introduced a novel technique called **the**treatment of outlier data as missing values using imputation methods (**TOMI**), which innovatively addresses outliers by treating them as missing values rather than eliminating them through conventional approaches. The TOMI technique effectively bridges the gap between outlier detection and missing value imputation, establishing a comprehensive framework for data preprocessing in **ML** applications. The methodology operates through a systematic three-phase process: **a)** outlier detection utilizing a sophisticated hybrid approach combining Z-score and IQR methods, **b)** strategic transformation of identified outliers into missing values while preserving data structure, and **c)** implementation of advanced imputation methods to estimate and replace these values. This methodological approach distinguishes itself by preserving both the dataset's volume and intrinsic variable relationships, thereby overcoming significant limitations inherent in traditional methods such as deletion or winsorization. Extensive empirical analyses demonstrate TOMI's capacity to enhance model accuracy and prediction stability across diverse **ML** contexts. The technique represents a significant advancement in data preprocessing methodologies, offering a robust solution for managing outliers while maintaining data integrity in scenarios where precise analysis and

reliable prediction are paramount. Additionally, TOMI's adaptability to various data types and its systematic approach to outlier handling make it particularly valuable for real-world applications where data quality directly impacts analytical outcomes. We can separate our strategy into four primary segments as follows: data collection, data pre-processing, data training, and applications of ML algorithms. This paper is structured as follows: **Section 2:** Presents an overview of the **ML** algorithms used in the study. **Section 3:** Discusses the performance metrics employed to evaluate model accuracy. **Section 4:** Introduces the German credit dataset and its key characteristics. **Section 5:** Delves into the analysis of real-world credit data using the selected ML algorithms. **Section 6:** Offers concluding remarks and insights based on the research findings

2. The ML Algorithms

The ML and data mining are powerful tools for addressing a wide range of complex problems. As a branch of artificial intelligence, ML plays a critical role in data science, enabling the extraction of meaningful insights and solutions from data. It is an interdisciplinary field that integrates concepts from computer science, statistics, cognitive science, engineering, and various domains of mathematics and science. One of the most common applications of ML is predictive modeling, where the goal is to estimate an outcome (the dependent variable) based on patterns identified in existing data (the independent variables). By learning these patterns from a known dataset, ML algorithms can generalize and apply them to new, unseen data, thereby making accurate predictions. This capability makes ML an indispensable tool for decision-making and problem-solving across numerous industries.

2.1. Logistic Regression

Logistic regression (LR) is a widely used supervised classification algorithm that models the relationship between a categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function. It is a special case of the **Generalized Linear Model (GLM)** and shares similarities with linear regression (Mohamed et al., 2023; Seliem et al., 2025a). The LR is particularly suited for modeling binary outcomes, where the response variable Y takes one of two possible values (e.g., 0 or 1). The connection between logistic regression and the **exponential family** of distributions is fundamental, as it provides the theoretical foundation for its formulation and estimation. The exponential family is a broad class of probability distributions that includes many common distributions, such as the Gaussian (normal), binomial, Poisson, and gamma distributions. A distribution belongs to the exponential family if its probability mass or density function can be expressed in the form:

$$f(y_i, \theta_i, \phi) = \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right]; \quad i = 1, 2, ..., n \tag{2}$$

where θ is the natural parameter (related to the mean of the distribution), ϕ is the dispersion parameter (often a constant for binary data), $b(\theta)$ is the cumulant function (related to the moment-generating function), $a(\phi)$ and $c(y_i, \phi)$ are known functions. For binary data, the **Bernoulli distribution** is a member of the exponential family. Its probability mass function is:

$$P(Y = y) = p^y(1 - p)^{1-y}, \tag{3}$$

where $p = P(Y = 1)$. Rewriting this in **the** exponential family form:

$$\ln P(Y = y) = y \ln\left(\frac{p}{1 - p}\right) + \ln(1 - p). \tag{4}$$

The **natural parameter** is $\theta = \ln\left(\frac{p}{1-p}\right)$, which is the **log-odds** or **logit** of p. The **cumulant function** is $b(\theta) = \ln(1 + e^\theta)$. The LR models the probability p of a binary outcome as a function of predictor variables $x_1, x_2, \ldots, x_k$. It consists of three key components:

1. **Random Component**: The response Y follows a Bernoulli distribution, which is a member of the exponential family.

2. **Systematic Component**: A linear predictor $\eta = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$.
3. **Link Function**: The **logit function** (canonical link for the Bernoulli distribution) connects the mean p to the linear predictor:

$$g(p) = \ln\left(\frac{p}{1-p}\right) = \eta. \tag{5}$$

The inverse of the logit function is the **logistic function**:

$$p = \frac{1}{1+e^{-\eta}}. \tag{6}$$

The exponential family provides a unified framework for GLMs, including logistic regression. Key advantages include:

- **Canonical Link Function**: The logit link is the natural choice for binary data, ensuring interpretability and computational efficiency.
- **Mean and Variance Relationship**: The variance of Y is $p(1-p)$, which is directly derived from the exponential family structure.
- **Estimation**: Maximum likelihood estimation (MLE) is straightforward due to the exponential family's properties, often implemented using iteratively reweighted least squares (IRLS).

In summary, LR model is a powerful tool for modeling binary outcomes, rooted in the exponential family of distributions. By leveraging the Bernoulli distribution and the logit link function, it provides a flexible and interpretable framework for understanding the relationship between predictors and binary responses. The exponential family underpins its theoretical and computational foundations, making LR a cornerstone of modern statistical modeling.

2.2. Naive Bayes

The NB algorithm is a probabilistic classification method based on **Bayes' theorem**, which operates under the assumption that the features are conditionally independent of each other given the class label. This assumption, known as the "naive" assumption, simplifies the computation and makes the algorithm computationally efficient, even for large datasets. NB calculates the probability of each potential classification and assigns the class with the highest probability to the given instance (Seliem, 2022; Seliem et al., 2025b). The core of the algorithm relies on the following equation derived from Bayes' theorem:

$$P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right) \cdot P(A)}{P(B)}, \tag{7}$$

where $P\left(\frac{A}{B}\right)$ is the posterior probability of class A given the features B, $P\left(\frac{B}{A}\right)$ is the likelihood of observing the features B given class A, $P(A)$ is the prior probability of class A, $P(B)$ is the marginal probability of the features B, which acts as a normalizing constant. By leveraging this probabilistic framework, NB is particularly effective for tasks such as text classification, spam filtering, and other applications where feature independence is a reasonable assumption. For example, in spam filtering, the algorithm calculates the probability of an email being spam based on the presence of certain words, assuming that the occurrence of each word is independent of others. Despite its simplicity, NB often performs competitively with more complex models, especially in high-dimensional spaces. However, its performance may degrade if the feature independence assumption is significantly violated.

2.3. Support Vector Machine

The SVMs are a powerful and versatile machine learning algorithm widely used for classification tasks. They excel at handling both linear and nonlinear data by finding an optimal hyperplane that separates data points into distinct classes while maximizing the margin—the distance between the hyperplane and the nearest data points

from each class (Abd El-Salam et al., 2019). This margin maximization helps improve the model's generalization to unseen data. SVMs leverage kernel functions to transform input data into higher-dimensional spaces, enabling the classification of data that is not linearly separable in its original feature space. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. For instance, the RBF kernel measures the similarity between data points, allowing SVMs to capture complex patterns and nonlinear relationships in the data. The optimization problem in SVMs is formulated as follows:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \quad \text{for all} \quad i = 1, 2, \ldots, n, \tag{8}$$

where $\mathbf{w}$ is the weight vector defining the orientation of the hyperplane, b is the bias term that shifts the hyperplane, $\mathbf{x}_i$ represents the feature vector of the i-th data point, y_i is the class label of the i-th data point ($y_i \in \{-1, 1\}$), and n is the total number of data points. Let us consider a dataset $(A_1, B_1, \ldots, A_n, B_n)$, where $(A_1, \ldots, A_n)$ is the set of input variables, $(B_1, \ldots, B_n)$ is the output variable, and 'C' is the intercept. Then, the SVM classifier is given as the following equation:

$$SVM = \sum_{i=1}^{n} \beta_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_i b_j C(a_i, a_j)\beta_i\beta_j \tag{9}$$

In equation (9), $i = 1, 2, 3, \ldots, n$, and $C = b_i\beta_i + b_j\beta_j$. By solving this optimization problem, SVMs identify the hyperplane that not only separates the classes but also ensures the largest possible margin, thereby improving generalization to unseen data. This makes SVMs highly effective for a wide range of applications, including image classification, text categorization, and bioinformatics. The primary advantage of SVM lies in its ability to handle a wide variety of classification problems, including high-dimensional and non-linearly separable datasets, by leveraging kernel functions to transform the data into a higher-dimensional space. However, one of the major drawbacks of SVM is that it depends on the careful selection of key parameters, such as the regularization parameter C and the kernel parameters, to achieve optimal classification performance Reddy et al. (2019). The general workflow of SVM involves two main steps: (1) identifying boundaries that correctly classify the training dataset, and (2) selecting the boundary with the maximum margin from the closest data points Rawal (2020).

2.4. Decision Tree

The DT algorithm is a robust, nonparametric supervised learning method widely used for classification and regression tasks. It operates by recursively partitioning the dataset based on attribute selection, which is determined by calculating information gain. Information gain is derived from entropy, a measure of impurity or uncertainty in the dataset, as defined by the equation:

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{10}$$

where p_i represents the proportion of instances in the dataset S that belong to class i, and c is the total number of classes. The algorithm evaluates the information gain for each attribute using the formula:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{\vartheta \in \text{values}(A)} \frac{|S_\vartheta|}{|S|}\text{Entropy}(S_\vartheta) \tag{11}$$

Here, A is an attribute, S_ϑ is the subset of S where attribute A has value ϑ, and $|S|$ denotes the size of the dataset. The attribute with the highest gain is selected for splitting. The attribute with the highest gain is selected for splitting. This process begins by identifying the best attribute to place at the root of the tree, followed by recursively splitting nodes based on the highest information gain until all attributes are assigned as leaf nodes across the tree's branches. The decision tree's hierarchical structure makes it an interpretable and efficient model, capable of handling both categorical and numerical data while providing clear insights into the decision-making process (Seliem, 2022).

2.5. K-Nearest Neighbor

The KNN algorithm is a simple yet effective ML method widely used for both classification and regression tasks. As a nonparametric algorithm, KNN does not make assumptions about the underlying data distribution, making it highly flexible for various applications (El-sayed et al., 2019; Prasannavenkatesan et al., 2021). Given an input $\mathbf{x}$, KNN identifies the k closest data points (neighbors) in the training set based on a distance metric. The most commonly used metric is the **Euclidean distance**, which measures the straight-line distance between two points in a multi-dimensional space:

$$d(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{p} (x_j - x_{ij})^2}, \tag{12}$$

where $\mathbf{x}_i$ represents a training point and p is the number of features. For **classification**, KNN predicts the class of $\mathbf{x}$ by taking a majority vote among the k neighbors:

$$\hat{y} = \text{mode}\{y_i \mid i \in N_k(\mathbf{x})\}, \tag{13}$$

where $N_k(\mathbf{x})$ is the set of indices of the k nearest neighbors. For **regression**, it predicts the output as the average of the target values of the k neighbors:

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i. \tag{14}$$

The choice of k is a critical hyperparameter that controls the model's flexibility. A small k leads to a more complex fit that may capture noise in the data, while a large k results in a smoother model that may oversimplify the decision boundary. In practice, k is typically chosen through cross-validation. Despite its simplicity, KNN has some limitations. It can be computationally expensive for large datasets, as it requires calculating the distance between the test point and every training point. Additionally, KNN is sensitive to the scale of features, so it is often recommended to normalize or standardize the data before applying the algorithm. Other distance metrics, such as Manhattan or Minkowski distance, can also be used depending on the specific application.

2.6. Artificial Neural Network

The ANNs are computational models inspired by the human brain's structure and function. Their research and application have grown significantly in recent decades. ANNs consist of interconnected layers: an input layer, one or more hidden layers, and an output layer. Data enters the network at the input layer, is processed through the hidden layers, and generates the output at the output layer (Eltalhi and Kutrani, 2019). ANNs are computational models inspired by the structure and function of the human brain. Their research and application have grown significantly in recent decades, driven by advancements in computational power and the availability of large datasets. ANNs consist of interconnected layers: an input layer, one or more hidden layers, and an output layer. Data enters the network at the input layer, is processed through the hidden layers via weighted connections and activation functions, and generates the final output at the output layer. This layered architecture allows ANNs to learn complex patterns and relationships in data, making them highly effective in performing tasks such as classification, regression, and pattern recognition.

2.7. Random Forest

The RF is a widely used supervised ML algorithm, applicable to both regression and classification tasks, though it typically excels in classification. It is particularly effective for large datasets with high dimensionality, making it a versatile tool in various domains. The core principle of RF is to combine multiple weak learners (decision trees) to create a strong learner, leveraging the power of ensemble learning. The RF algorithm can be summarized as follows(Darst et al., 2018):

1. Randomly select K data points from the training set.
2. Construct a decision tree using the selected K data points.
3. Repeat steps 1 and 2 to generate N decision trees.
4. For a new data point, predict its category by aggregating the predictions of all N trees and assigning the category with the highest probability.

This ensemble approach ensures high accuracy and generalization, making RF a powerful and reliable algorithm for predictive modeling.

3. Performance evaluation criteria

The performance of machine learning (ML) prediction algorithms, especially for classification tasks, is typically assessed using specific metrics. In this research, we employed a comprehensive set of metrics to evaluate our models' performance (Afzal et al., 2021; Seliem, 2022):

- **Confusion Matrix:** Table 2 summarizes the model's predictions and actual classifications, providing insights into the types of errors made.

Table 2. Confusion Matrix.

Actual	Predicted	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

where:

  - **True Positive (TP):** The number of positive samples correctly predicted as positive.
  - **False Negative (FN):** The number of positive samples incorrectly predicted as negative.
  - **False Positive (FP):** The number of negative samples incorrectly predicted as positive.
  - **True Negative (TN):** The number of negative samples correctly predicted as negative.

- **Kappa statistic**

$$\frac{[\frac{TP+TN}{N}] - [\frac{(TP+FN)(TP+FP)(TN+FN)}{N^2}]}{1 - [\frac{(TP+FN)(TP+FP)(TN+FN)}{N^2}]} \tag{15}$$

- **Accuracy**

$$\frac{(TP+TN)}{TP+FP+TN+FN} \tag{16}$$

- **Precision**

$$\frac{TP}{TP+FP} \tag{17}$$

- **Recall**

$$\frac{TP}{TP+FN} \tag{18}$$

- **F1 Score**

$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

4. The German Credit Dataset

We consider the widely used German credit dataset from the UCI Machine Learning Repository, given by German professor Hans Hofmann via the European Statlog project. The dataset has been widely used in machine learning research. Several R packages include this data i.e., evtree, CollapseLevels, caret, gamclass, klaR, and rchallenge. The data consists of 1000 credits and a stratified sample of 1000 credits to 300 bad ones and 700 good ones. This data was collected from 500 branches of a large regional bank in southern Germany's urban and rural areas from 1973 to 1975. The data consists of 1000 observations with one response variable

$$(Y)$$

: [0 for good risks and 1 for bad risks] and 20 explanatory variables (from X_1 to X_{20}). The 20 explanatory variables in the data set originally contained categorical and numerical variables; for example, the numerical variables are Credit amount and Age. The complete information of variables is presented in Table (3).

Table 3. The Description of Variables of the German Credit Dataset

No	Variable name	Attribute	Description
X_1	Status	The account status of the debtor with a bank	categorical
X_2	Duration	The duration of credit in months	Quantitative
X_3	Credit History	The contract's history of previous or current credit	categorical
X_4	Purpose	The reason behind credit	categorical
X_5	Amount	The total amount of credit	Quantitative
X_6	Savings	Total savings of debtor	categorical
X_7	Employment	Debtor's tenure with current organization	Ordinal
X_8	Installment Rate	The credit installments of debtor's throwaway income	Ordinal
X_9	Personal Status Sex	The information about both sex and marital status	categorical
X_{10}	Other Debtors	Another debtor for the credit	categorical
X_{11}	Present Residence	The duration of living in the present residence	Ordinal
X_{12}	Property	The ranking of debtor's property in ascending order	Ordinal
X_{13}	Age	The age of the debtor	Quantitative
X_{14}	Other Installment Plans	Installment loans from other sources	categorical
X_{15}	Housing	Status of current residence	categorical
X_{16}	Number Credits	The complete history of the credits taken	Ordinal
X_{17}	Job	The level of the debtor's job	Ordinal
X_{18}	People Liable	The total number of peers depends on the debtor financially	Quantitative
X_{19}	Telephone	The status of a registered landline on the debtor's name	Binary
X_{20}	Foreign Worker	Is the debtor a foreign worker	Binary
Y	Credit Risk	Good or Bad	Binary

Table 4. Summary statistics of the variables.

Sympol	Attribute& Description	Coding	Level	Freq.	Percentage	
Y	Dependent	0	Good Risks	700	70%	
		1	Bad Risks	300	30%	
Numerical Independent		Min	Max	Mean	SD	VIF
X_2	Duration	4	72	20.9	12.05	1.64
X_5	Amount	250	18424	3271	282.73	1.65
X_{13}	Age	19	75	35.55	11.37	1.01

In Table (4), our analysis is based on the complete data and some descriptive statistics for all quantitative independent and dependent variables. For summary statistics about the categorical variables, see Groemping (2019). Before doing the ML algorithms, the multicollinearity between the independent variables should be checked. We consider the correlation matrix and variance inflation factor (VIF) to diagnose this problem. Table (4) displays the VIF between the independent variables.

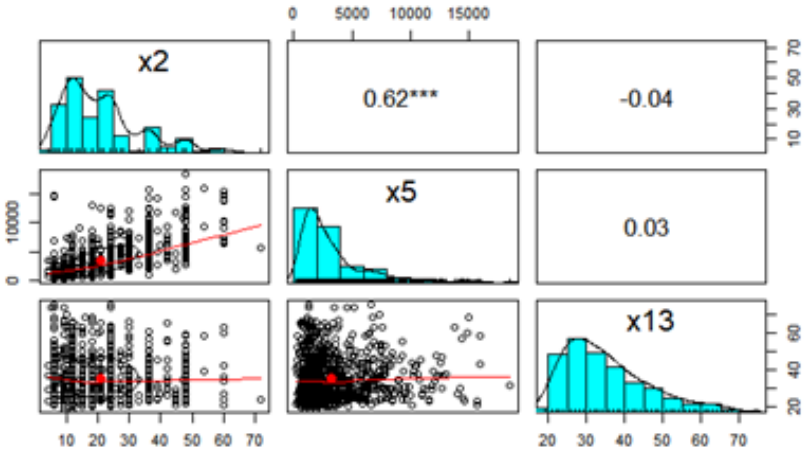

Figure 2. Correlation Matrix.

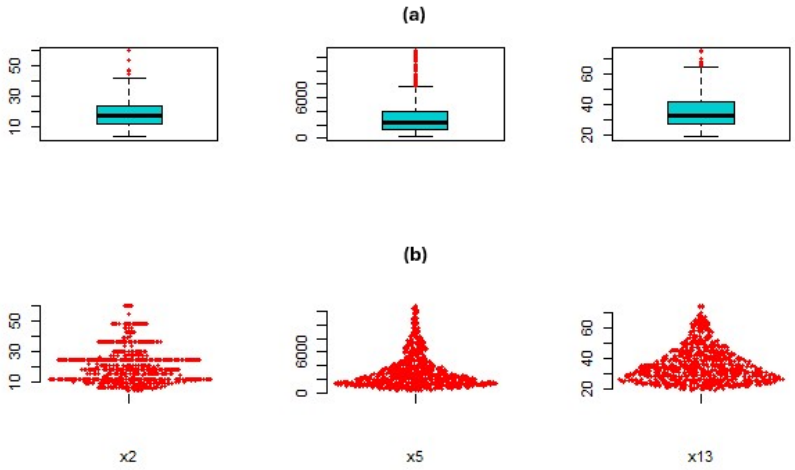

Figure 3. (a): Boxplots. (b):Violin Plots.

From Figure(2), we observed that some independent variables are weakly correlated since all correlation coefficients are less than 0.8. Moreover, the VIF for all independent variables is less than 5. This means that we have no multicollinearity problem (Hair Jr et al., 2021). Furthermore, we also check the outliers in the German credit dataset. For this purpose, we consider the boxplot as Figure 3 and Rosner's test as Tabel 5. Figure 3 presents a comparative visualization of the distributions for three variables, X_2, X_5, and X_{13}, using both boxplots (a) and violin plots (b). The boxplots in (a) offer a concise summary of the data's central tendency, spread, and potential outliers through the median, quartiles, and whiskers. Notably, X_5 exhibits a substantially higher median value compared to X_2 and X_{13}, suggesting a rightward shift in its distribution. The violin plots in (b) provide a more nuanced depiction of the data's density and shape, revealing the underlying distribution's multimodality and skewness. For instance, X_2 shows a distinct multi-peaked structure, indicating potential clustering within the

data. The violin plots also highlight the presence of outliers, consistent with the boxplots, further enriching the understanding of each variable's distributional characteristics. This dual representation allows for a comprehensive assessment of the data, leveraging the strengths of both boxplots and violin plots to reveal both summary statistics and detailed distributional information.

Table 5. Results of Rosner's Test for Outliers.

Var	i	Mean.i	SD.i	Value	Obs.Num	R.i.1	lambda.i.1	Outlier
X_2	0	20.90	12.06	72.	678.	4.24	4.04	TRUE(bad)
	1	20.85	11.96	60.	30.00	3.27	4.04	FALSE
X_5	0	3271.26	2822.74	18424.	916.	5.37	4.04	TRUE (bad)
	1	3256.09	2783.08	15945	96.	4.56	4.04	TRUE (bad)
	2	3243.38	2755.29	15857	819	4.58	4.04	TRUE (bad)
	3	3230.72	2727.52	15672	888	4.56	4.04	TRUE (bad)
	4	3218.23	2700.21	15653	638	4.61	4.04	TRUE (bad)
	5	3205.74	2672.59	14896	918	4.37	4.04	TRUE (bad)
	6	3193.97	2648.05	14782	375	4.38	4.04	TRUE (bad)
	7	3182.31	2623.69	14555	237	4.33	4.04	TRUE (bad)
	8	3170.84	2600.01	14421	64	4.33	4.04	TRUE (bad)
	9	3159.49	2576.61	14318	379	4.33	4.04	TRUE (bad)
	10	3148.22	2553.35	14179	745	4.32	4.04	TRUE (bad)
	11	3137.06	2530.40	14027	715	4.30	4.04	TRUE (bad)
	12	3126.04	2507.81	13756	374	4.24	4.04	TRUE (bad)
	13	3115.27	2486.12	12976	382	3.97	4.04	FALSE
	14	3105.27	2467.44	12749	922	3.91	4.04	FALSE
X_{13}	0	35.55	11.38	75	331.00	3.47	4.04	FALSE
	1	35.51	11.31	75	537.00	3.49	4.04	FALSE
Handling Outliers by TOMI Technique								
X_2	0	20.85	11.96	60	30	3.27	4.04	FALSE
	1	20.81	11.90	60	135	3.29	4.04	FALSE
X_5	0	3115.27	2486.12	12976	382	3.97	4.04	FALSE
	1	3105.27	2467.44	12749	922	3.91	4.04	FALSE

Table 5 delineates the outcomes of Rosner's test for outlier detection applied to variables X_2, X_5, and X_{13} within the dataset. Rosner's test, an iterative statistical procedure, evaluates the presence of outliers by computing the mean (`Mean.i`), standard deviation (`SD.i`), and test statistic (`R.i.1`) for each observation. Observations exceeding the critical value (`lambda.i.1`) are classified as outliers (`TRUE`), as highlighted in red. Notably, **X5** demonstrates a pronounced prevalence of outliers across multiple iterations, indicative of substantial variability or extreme values within this variable. In contrast, X_{13} exhibits no outliers, suggesting its relative stability. Following the application of the TOMI methodology, the outlier status for X_2 and X_5 transitions to `FALSE`, signifying the successful mitigation of anomalous data points. This underscores the efficacy of the TOMI technique in addressing outlier-induced distortions, thereby enhancing the dataset's robustness for subsequent analytical procedures. The iterative resolution of outliers in X_5 further highlights the technique's capability to iteratively refine data quality, ensuring the integrity of statistical inferences.

Table 6. Performance Evaluation of Imputation Methods.

	Algorithm				
Metric	Cart	PMM	RF	Mean	Median
MAE	55	59	60	78	83
RMSE	736	763	772	972	1038

Table 6 presents a comparative evaluation of five imputation methods: CART, PMM, RF, mean Imputation, and median Imputation, based on two performance metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Lower values for both metrics indicate superior imputation accuracy. The results demonstrate that the **CART** method outperforms the others, achieving the lowest MAE (55) and RMSE (736). This suggests that CART is the most effective technique for minimizing imputation errors in the dataset.

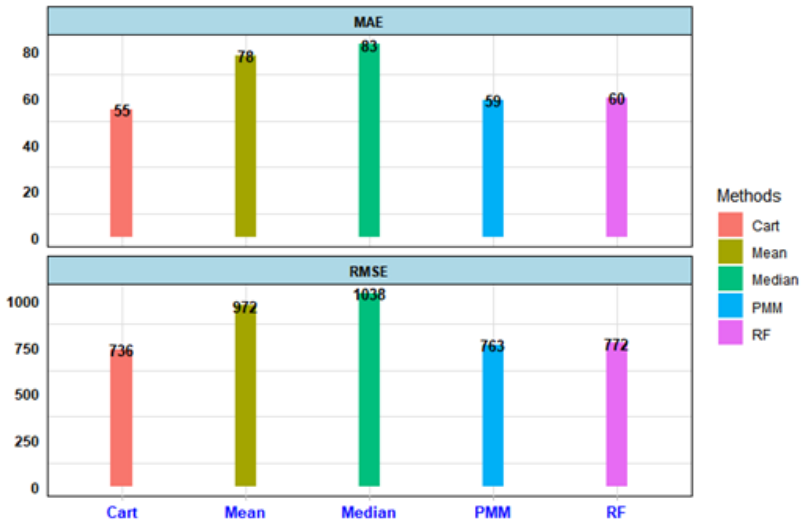

Figure 4. The MAE and RMSE of imputation Methods.

The **PMM** and **RF** methods exhibit comparable performance, with slightly higher MAE and RMSE values, indicating moderate effectiveness. In contrast, **Mean** and **Median Imputation** yield significantly higher errors (MAE: 78 and 83; RMSE: 972 and 1038, respectively), highlighting their limitations in preserving data accuracy. These findings underscore the importance of selecting advanced imputation methods, such as CART, PMM, or RF, over simpler techniques like Mean or Median Imputation, particularly in datasets where precision is critical. The superior performance of CART can be attributed to its ability to model complex relationships within the data, thereby reducing imputation bias and error. Additionally, Figure 4 corroborates these conclusions, further validating the effectiveness of the CART method.

5. Results and Discussion

Feature selection is a critical step in ML algorithms, as selecting optimal features significantly enhances the accuracy of predictive models. To identify the most significant variables, we employed the Boruta algorithm (Mousa et al., 2022; Alam et al., 2020). Table 7 presents the results of this feature selection process, which evaluates the importance of independent variables for predictive modeling. The Boruta algorithm assesses each variable using importance scores, including mean importance (`meanImp`), median importance (`medianImp`), minimum importance (`minImp`), maximum importance (`maxImp`), and normalized hits (`normHits`). Based on these scores, variables are classified as either **Confirmed** (important) or **Rejected** (unimportant), with rejected variables highlighted in red.

The results indicate that 14 out of 20 variables (e.g., X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{10}, X_{12}, X_{13}, X_{14}, X_{15}, and X_{17}) are confirmed as significant, with high importance scores and normalized hits close to 1. These variables are likely to contribute meaningfully to the predictive model. In contrast, variables such as X_9, X_{11}, X_{16}, X_{18}, X_{19}, and X_{20} are rejected due to low importance scores and normalized hits, suggesting they have limited predictive power.

Table 7. The selection of independent variables using Boruta algorithm

Variables	meanImp	medianImp	minImp	maxImp	normHits	decision
X_1	30.58	30.47	27.37	34.70	1.00	Confirmed
X_2	15.83	15.65	12.45	18.65	1.00	Confirmed
X_3	14.16	14.05	11.96	17.25	1.00	Confirmed
X_4	4.38	4.30	1.39	7.97	0.88	Confirmed
X_5	9.19	9.23	6.63	11.64	1.00	Confirmed
X_6	8.83	8.83	6.16	12.98	1.00	Confirmed
X_7	5.04	5.12	1.78	7.93	0.90	Confirmed
X_8	2.67	2.59	0.76	6.19	0.55	Confirmed
X_9	1.08	1.05	-0.58	3.03	0.03	Rejected
X_{10}	7.41	7.60	4.49	10.43	1.00	Confirmed
X_{11}	1.24	1.21	0.12	2.14	0.00	Rejected
X_{12}	6.74	6.65	4.45	10.14	1.00	Confirmed
X_{13}	4.99	4.93	2.33	7.96	0.98	Confirmed
X_{14}	5.94	5.78	3.45	8.31	1.00	Confirmed
X_{15}	3.25	3.39	0.74	6.26	0.67	Confirmed
X_{16}	1.99	1.81	0.48	4.18	0.16	Rejected
X_{17}	2.49	2.43	-0.44	6.03	0.52	Confirmed
X_{18}	1.39	1.68	-0.70	3.14	0.00	Rejected
X_{19}	0.87	0.89	-0.27	2.30	0.00	Rejected
X_{20}	1.86	1.83	-1.18	4.39	0.18	Rejected

By distinguishing relevant from irrelevant features, the Boruta algorithm ensures a robust and interpretable feature set, improving model performance and generalizability. This feature selection process is critical for reducing dimensionality, mitigating overfitting, and improving computational efficiency in subsequent modeling steps. As shown in Table 7, the 14 confirmed variables were determined to be statistically significant at the 0.01 level in predicting the dependent variable. These findings align with the visual representation in Figure 5, further validating the results.

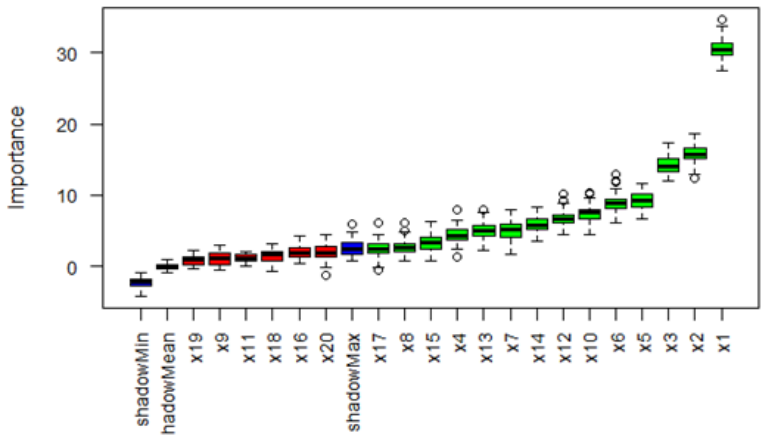

Figure 5. The important Variables based on Boruta algorithm.

By focusing on the identified key predictors, we have established a strong foundation for constructing a precise and dependable predictive model. The decision is based on a comparison between each variable's importance score

and the importance scores of randomly permuted shadow features. These results provide valuable guidance for model building by identifying the most informative variables. The generic form of the predictive model is:

$$\hat{Y} = f(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{10}, X_{12}, X_{13}, X_{14}, X_{15}, X_{17}) \tag{20}$$

where $\hat{Y}$: Predicted output (dependent variable), $f(\cdot)$: The function representing the model, which varies depending on the algorithm and $X_1, X_2, \ldots, X_{17}$: Confirmed independent variables (features). Imbalanced datasets are a common problem in ML, often causing poor performance in classification tasks. To address this, resampling techniques are used to balance the classes. This can be done by either oversampling the minority class or undersampling the majority class. Oversampling is usually preferred because undersampling can lose important data. Random oversampling duplicates minority class examples, but this can lead to overfitting. A better method is SMOTE, which creates new synthetic examples for the minority class by combining nearby instances. This helps improve performance without overfitting. However, since the imbalance was not strong, we also performed the analysis with the original imbalanced data. After feature selection to build a classification model, the combined dataset with 14 attributes is divided into training and testing data with a percentage split of 70%-30% and 80%-20%. In case 1, data is split below into two subsets: training (70%) and testing (30%) while in case 2, data is split below into two subsets: training (80%) and testing (20%). The confusion matrix obtained by seven different supervised ML algorithms is given below. The performance measures follow the accuracy of each classification algorithm. Ten-fold cross-validation was utilized to evaluate the performance of the classification models. In this approach, the entire dataset is divided into ten subsets and processed ten times where nine subsets are used as testing sets and the remaining subset is used as training. Finally, the results are obtained by averaging every ten iterations.

Table 8. Case 1 splitting data to70%-30%.

	Algorithms	Performances					
		Confusion matrix			Metric		
		TYPE	No Pre.	Pre.	ACC	Kapp	F1.Score
Phase Testing	**NB**	No Pre.	189	52	0.756	0.357	0.838
		Pre.	21	38			
	SVM	No Pre.	198	67	0.736	0.24	0.833
		Pre.	12	23			
	DT	No Pre.	192	59	0.743	0.297	0.833
		Pre.	18	31			
	KNN	No Pre.	195	62	0.743	0.281	0.835
		Pre.	15	28			
	ANN	No Pre.	187	52	0.75	0.344	0.833
		Pre.	23	33			
	LR	No Pre.	200	69	0.736	0.228	0.835
		Pre.	10	21			
	RF	No Pre.	199	57	**0.7733**	**0.368**	**0.854**
		Pre.	11	33			

Table 8 presents the performance evaluation of seven ML algorithms: NB, SVM, DT, KNN, ANN, LR, and RF; on a dataset split into 70% training and 30% testing (Case 1). The evaluation is based on confusion matrix metrics and performance metrics, including Accuracy, Kappa statistics, and F1 Score. The results demonstrate that the **Random Forest (RF)** algorithm outperforms the others, achieving the highest accuracy (77.3%), kappa statistic (36.8%), and F1 score (85.4%). This indicates that RF is the most effective model for this dataset, balancing precision and recall effectively. In contrast, **SVM** and **LR** exhibit relatively lower performance, with accuracy values of 0.736 and 0.736, respectively, and lower Kappa statistics, suggesting weaker predictive capabilities. The confusion matrix values further reveal the algorithms' ability to correctly classify instances. For example,

RF correctly predicts 199 instances as ”No Pre.” and 33 as ”Pre.,” demonstrating its robustness in handling both classes. These findings highlight the importance of selecting an appropriate algorithm for predictive modeling, as performance can vary significantly across methods. The superior performance of RF underscores its suitability for this dataset, particularly in scenarios requiring high accuracy and balanced class predictions. Additionally, Figure 6 corroborates these conclusions, further validating the effectiveness of the RF algorithm.

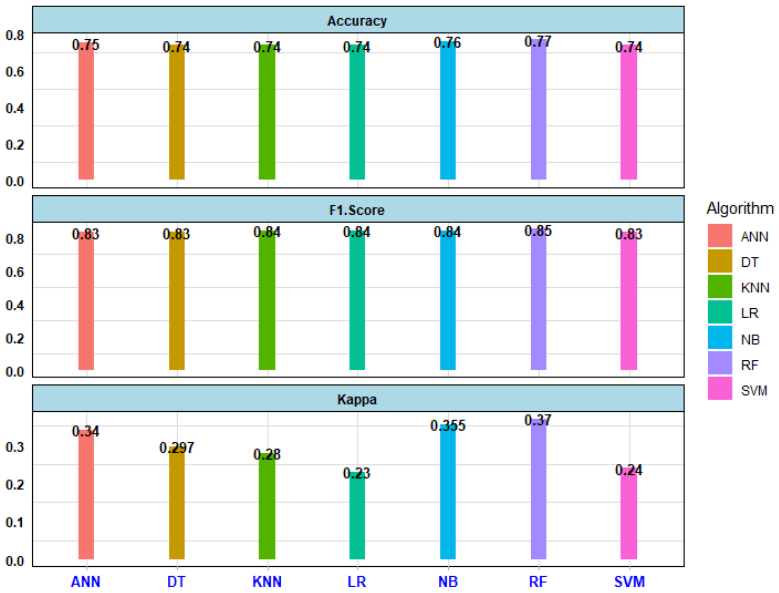

Figure 6. : Performance Evaluation for Case 1.

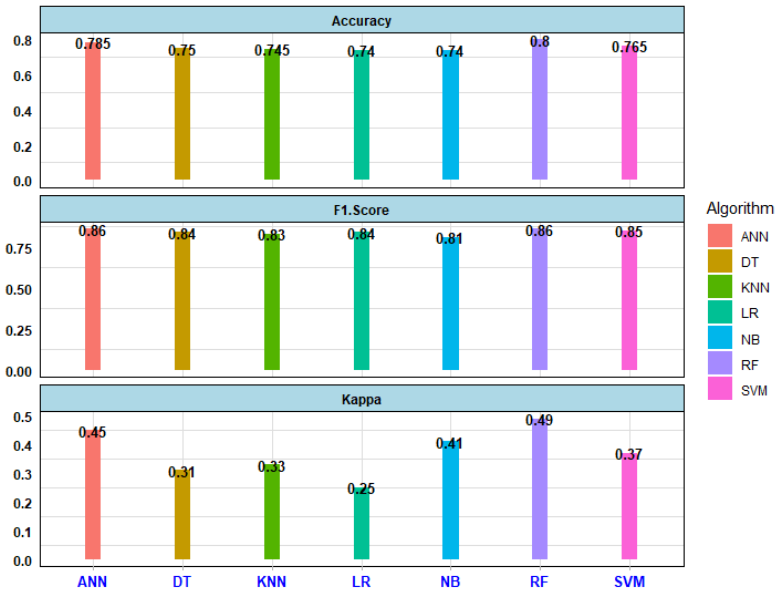

Figure 7. : Performance Evaluation for Case 2.

Building on the results from Table 8 (70%-30% split), Table 9 evaluates the performance of the same seven machine learning algorithms on a dataset split into 80% training and 20% testing (Case 2). This comparison allows for an assessment of how the training-testing ratio impacts model performance.

Table 9. Case 2 splitting data to 80%-20%.

	Algorithms	Performances					
		Confusion matrix			Metric		
		TYPE	No Pre.	Pre.	ACC	Kapp	F1.Score
Phase Testing	**NB**	No Pre.	109	21	0.74	0.409	0.807
		Pre.	31	39			
	SVM	No Pre.	128	35	0.765	0.372	0.845
		Pre.	12	25			
	DT	No Pre.	130	40	0.75	0.305	0.839
		Pre.	10	20			
	KNN	No Pre.	125	36	0.745	0.325	0.831
		Pre.	15	24			
	ANN	No Pre.	127	30	0.785	0.443	0.855
		Pre.	13	30			
	LR	No Pre.	132	44	0.74	0.252	0.835
		Pre.	8	16			
	RF	No Pre.	126	26	**0.80**	**0.494**	**0.863**
		Pre.	14	34			

The results in Table 9 show that the **RF** algorithm continues to outperform the others, achieving the highest accuracy (0.80), Kappa coefficient (0.494), and F1 Score (0.863). This is consistent with its superior performance in Case 1 (Table 8), further validating RF's robustness and effectiveness across different data splits. Similarly, the **(ANN)** maintains strong performance, with an accuracy of 0.785 and an F1 Score of 0.855, reinforcing its suitability for complex datasets. Notably, the performance gap between RF and other algorithms, such as **(LR)** and **(DT)**, remains pronounced in Case 2, with LR and DT achieving lower accuracy (0.74 and 0.75, respectively) and Kappa coefficients. This aligns with the findings in Table 8, where RF consistently demonstrated superior predictive capabilities. These findings align with the visual representation in Figure (7).

A key observation is the consistent improvement in overall performance metrics across most algorithms in Case 2 compared to Case 1. For instance, the accuracy of the Random Forest (RF) model increased from 0.7733 in Case 1 to 0.80 in Case 2, while the Artificial Neural Network (ANN) model showed an improvement from 0.75 to 0.785. This trend suggests that a larger training set (80% compared to 70%) enhances model performance, likely due to the increased availability of data for learning underlying patterns. These findings highlight the critical role of both algorithm selection and training-testing ratios in predictive modeling.

The consistent superiority of RF across both cases underscores its robustness and reliability as a modeling approach. Furthermore, the improved performance in Case 2 emphasizes the value of larger training datasets in achieving higher accuracy and model generalizability. The Random Forest model, which aggregates predictions from multiple decision trees, can be expressed as:

$$\hat{Y} = \frac{1}{N} \sum_{i=1}^{N} \text{Tree}_i(X_1, X_2, \ldots, X_{17}), \tag{21}$$

where N is the number of trees in the ensemble, and $\text{Tree}_i(\cdot)$ represents the i-th decision tree. This ensemble approach contributes to the model's robustness and superior performance across both cases.

6. Conclusion

This paper emphasizes the significance of data preprocessing in optimizing ML models for credit risk prediction. By employing the TOMI technique for outlier removal and the Boruta algorithm for feature selection, five non-contributing features were identified and eliminated, enhancing prediction accuracy. Seven ML algorithms (NB, SVM, DT, KNN, LR, and RF) were evaluated using metrics such as accuracy, kappa statistics, and F1-score. The RF algorithm emerged as the most effective model, achieving accuracy rates of 77.3% and 80% for the $70\% - 30\%$ and $80\% - 20\%$ splits, respectively, along with kappa statistics of 36.8% and 49.4% and F1-scores of 85.4% and 86.3%. These findings showcase how effective preprocessing techniques can substantially enhance the performance of ML models in predicting loan defaults, with RF outperforming other algorithms.

REFERENCES

Abd El-Salam, S. M., Ezz, M. M., Hashem, S., Elakel, W., Salama, R., ElMakhzangy, H., and ElHefnawi, M. (2019). Performance of machine learning approaches on prediction of esophageal varices for egyptian chronic hepatitis c patients. *Informatics in Medicine Unlocked*, 17:100267.

Afzal, S., Afzal, A., Amin, M., Saleem, S., Ali, N., and Sajid, M. (2021). A novel approach for outlier detection in multivariate data. *Mathematical Problems in Engineering*, 2021(1):1899225.

Alam, T. M., Shaukat, K., Hameed, I. A., Luo, S., Sarwar, M. U., Shabbir, S., Li, J., and Khushi, M. (2020). An investigation of credit card default prediction in the imbalanced datasets. *Ieee Access*, 8:201173–201198.

Arora, N. and Kaur, P. D. (2020). A bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. *Applied Soft Computing*, 86:105936.

Arun, G. K. and Venkatachalapathy, K. (2020). Intelligent feature selection with social spider optimization based artificial neural network model for credit card fraud detection. *IIOABJ*, 11(2):85–91.

Darst, B. F., Malecki, K. C., and Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC genetics*, 19:1–6.

El-sayed, S. M., Abonazel, M. R., and Seliem, M. M. (2019). B-spline speckman estimator of partially linear model. *International Journal of Systems Science and Applied Mathematics*, 4:53.

Eltalhi, S. and Kutrani, H. (2019). Breast cancer diagnosis and prediction using machine learning and data mining techniques: A review. *IOSR Journal of Dental and Medical Sciences*, 18(4):85–94.

Groemping, U. (2019). South german credit data: Correcting a widely used data set. *Rep. Math., Phys. Chem., Berlin, Germany, Tech. Rep*, 4:2019.

Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Danks, N. P., and Ray, S. (2021). *Partial least squares structural equation modeling (PLS-SEM) using R: A workbook*. Springer Nature.

Imron, M. A. and Prasetyo, B. (2020). Improving algorithm accuracy k-nearest neighbor using z-score normalization and particle swarm optimization to predict customer churn. *Journal of Soft Computing Exploration*, 1(1):56–62.

Mardiansyah, H., Sembiring, R. W., and Efendi, S. (2021). Handling problems of credit data for imbalanced classes using smotexgboost. In *Journal of Physics: Conference Series*, volume 1830, page 012011. IOP Publishing.

Metawa, N., Pustokhina, I. V., Pustokhin, D. A., Shankar, K., and Elhoseny, M. (2021). Computational intelligence-based financial crisis prediction model using feature subset selection with optimal deep belief network. *Big Data*, 9(2):100–115.

Mohamed, S. M., Abonazel, M. R., and Ghallab, M. G. (2023). Performance evaluation of imputation methods for missing data in logistic regression model: simulation and application. *Thailand Statistician*, 21(4):926–942.

Mousa, G. A., Elamir, E. A., and Hussainey, K. (2022). Using machine learning methods to predict financial performance: Does disclosure tone matter? *International Journal of Disclosure and Governance*, 19(1):93–112.

Nalić, J., Martinović, G., and Žagar, D. (2020). New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers. *Advanced Engineering Informatics*, 45:101130.

Pławiak, P., Abdar, M., Pławiak, J., Makarenkov, V., and Acharya, U. R. (2020). Dghnl: A new deep genetic hierarchical network of learners for prediction of credit scoring. *Information sciences*, 516:401–418.

Prasannavenkatesan, T., Jacob, J., Ruby, U., and Vamsidhar, Y. (2021). Prediction of covid-19 possibilities using knn classification algorithm. *Accident Analysis and Prevention. Int J Cur Res Rev*, 13(06):156.

Rawal, R. (2020). Breast cancer prediction using machine learning. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 13(24):7.

Reddy, N. S. C., Nee, S. S., Min, L. Z., and Ying, C. X. (2019). Classification and feature selection approaches by machine learning techniques: Heart disease prediction. *International Journal of Innovative Computing*, 9(1).

Religia, Y., Pranoto, G. T., and Santosa, E. D. (2020). South german credit data classification using random forest algorithm to predict bank credit receipts. *JISA (Jurnal Informatika dan Sains)*, 3(2):62–66.

Saheed, Y. K., Hambali, M. A., Arowolo, M. O., and Olasupo, Y. A. (2020). Application of ga feature selection on naive bayes, random forest and svm for credit card fraud detection. In *2020 international conference on decision aid sciences and application (DASA)*, pages 1091–1097. IEEE.

Seliem, M. M. (2022). Handling outlier data as missing values by imputation methods: application of machine learning algorithms. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 13(1):273–286.

Seliem, M. M., El-Sayed, S. M., and Abonazel, M. R. (2025a). Developing a semi-parametric zero-inflated beta regression model using p-splines: Simulation and application. *Statistics, Optimization & Information Computing*, 13(3):1103–1119.

Seliem, M. M., Kamel, A. R., Taha, I. M., and El-Nasr, M. M. A. (2025b). A note on prior selection in bayesian estimation. *Statistics, Optimization & Information Computing*, 13(2):795–806.

Shen, F., Zhao, X., Kou, G., and Alsaadi, F. E. (2021). A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing*, 98:106852.

Shi, S., Tse, R., Luo, W., D'Addona, S., and Pau, G. (2022). Machine learning-driven credit risk: a systemic review. *Neural Computing and Applications*, 34(17):14327–14339.

Trivedi, S. K. (2020). A study on credit scoring modeling with different feature selection and machine learning approaches. *Technology in Society*, 63:101413.

Yang, F., Qiao, Y., Huang, C., Wang, S., and Wang, X. (2021). An automatic credit scoring strategy (acss) using memetic evolutionary algorithm and neural architecture search. *Applied Soft Computing*, 113:107871.

Zhang, H., He, H., and Zhang, W. (2018). Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. *Neurocomputing*, 316:210–221.