Comparative Study of Credit Score Modelling using Ensemble Techniques

Pradnya Patil[1], Tilottama Dhake[2]

[1]Computer Engineering Department K J Somaiya Institute of Technology, Mumbai,
Email: pradnya08@somaiya.edu
[2]Electronics & Telecommunication Department, K J Somaiya Institute of Technology, Mumbai,
Email: tdhake@somaiya.edu

Received: 10.07.2024 Revised: 13.08.2024 Accepted: 21.09.2024

ABSTRACT
Credit scoring, a cornerstone of modern finance, plays a pivotal role in assessing the creditworthiness of individuals and businesses. The primary motivation driving this review paper is to scrutinize the suitability and efficacy of algorithms in the context of credit scoring. This comprehensive review delves into the evolving landscape of credit risk assessment and explores the transformative impact of algorithmic decision making on reducing bias, overfitting and enhancing predictive accuracy. Over time, the accuracy of credit scoring models has seen substantial improvements, reflecting advancements in computational capabilities and data availability. By focusing on a spectrum of algorithms, from classical statistical methods to state-of-the-art machine learning and neural network approaches, this paper provides a comparative analysis of their strengths and limitations in the realm of credit scoring. The discussion encompasses considerations of interpretability, predictive accuracy, scalability, and fairness, shedding light on the trade-offs involved in algorithm selection.

Keywords: Credit Scoring, Machine learning, Classification, Creditworthiness, Default Prediction

1. INTRODUCTION

The world of credit scoring has undergone a remarkable transformation over the years, evolving from rudimentary systems of trust and collateral to sophisticated analytical methodologies that underpin financial decision-making on a global scale. This review paper begins with a comprehensive exploration of the historical evolution, contemporary issues, breakthroughs, and current trends in credit scoring analysis techniques.

A. A Brief History of Credit Scoring

The concept of credit scoring can be traced back to ancient civilizations where interpersonal trust served as the sole determinant for lending and borrowing. In the early 20th century, personal relationships gave way to more data-driven methods, exemplified by the birth of the credit bureaus. It wasn't until the mid-20th century that statistical models and computer technology began to play a pivotal role in credit scoring, making the process more objective and efficient. The advent of FICO [1] scores in 1956 marked a significant milestone in this journey, setting a standard for creditworthiness assessment.

B. Issues in Credit Scoring

However, as credit scoring became more complex, it brought with it an array of challenges. Discrimination, bias, and data privacy concerns emerged as critical issues [2]. Financial institutions faced dilemmas in reconciling their fiduciary duty to shareholders with societal expectations of equitable access to credit. Traditional scoring methods faced criticism for excluding individuals with limited credit histories or unconventional financial behaviors. These problems underscored the need for a new generation of credit scoring techniques that are not only more accurate but also fair and inclusive. Furthermore, interpretability and explainability of "black box" models raised the question of transparency [3].

C. Breakthroughs in Credit Score Techniques

The 21st century ushered in a wave of innovations that have revolutionized credit scoring. Machine learning algorithms, big data analytics, and alternative data sources have enabled the development of more robust and predictive models. Explainable AI and fairness-aware algorithms have been introduced

to mitigate biases and enhance transparency. Additionally, lending platforms and fintech companies have disrupted the traditional lending landscape, pushing the boundaries of credit scoring even further.

D. Current Ongoing Research

In the ever-evolving field of credit scoring, research is at the forefront of addressing the challenges and opportunities. Ongoing studies explore advanced modeling techniques, incorporating non-traditional data sources such as social media, and the use of blockchain for secure and transparent credit assessment. Furthermore, the ongoing discourse around regulatory frameworks, particularly in the context of data privacy and consumer protection, continues to shape the credit scoring landscape.

This review paper aims to delve into the various techniques involved in modern credit scoring, with a focus on algorithms that are flexible and accurate. The paper is divided into the following sections, Section II is a discussion of a subset of Machine Learning approaches.Section III proposes the methodology carried out by the authors as the various research steps in the implementation.Section IV is a comparative analysis of a few algorithms as applied on two standard datasets. These datasets will be described in the section alongwith the results obtained. Concluding remarks are discussed as well.Section V discusses potential future work and improvements.

2. LITERATURE REVIEW

In this study an overview of the primary computational techniques are employed in credit analysis, that belong to eitherof statistical learning, machine learning, and deep learning. Each of these techniques possesses distinct characteristics and shares some common principles. Statistical methods have traditionally been used to assess the credit behavior of customers or businesses. However, with the rapid advancement of artificial intelligence, machine learning and deep learning have gradually supplanted traditional statistical analysis. This evolution reflects the academic progression of the field and offers context for our study and the subsequent comparison of our findings.

A. Discriminant Analysis

Discriminant Analysis (DA) [4] is a multivariate statistical method used to predict group memberships and enhance separability among recognized clusters. It allows the classification of dataset classes by constructing linear functions that involve the explanatory variables. Nevertheless, critiques regarding its use in business, finance, and economics have arisen. One criticism, as indicated in [5], highlights the assumption that variables describing group members must adhere to a multivariate normal distribution. However, a counter-argument presented in [6] suggests that this assumption is a common misconception, asserting that if variables conform to a multivariate ellipsoidal distribution (of which normal distribution is a specialized case), the linear discriminant rule remains optimal. Furthermore, it has been contested whether DA can be successfully applied to discrete and identifiable groups, as opposed to continuous variables. Research presented in [7] demonstrates that the discriminant function, obtained by dividing a multivariate normal distribution into two classes, aligns with the optimal discriminant function, challenging the notion that DA works exclusively for discrete groups.

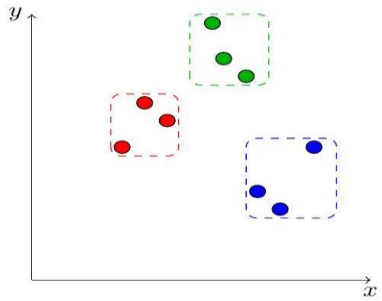

Fig 1. Representation of Individual Class Boundaries in Feature Space.
Source: Created by authors

Figure 1 depicts how DA might separate classes in the feature space. Additionally, Linear Discriminant Analysis (LDA) [4], was initially introduced by Fisher for taxonomic classification in the field of biology and has found versatile applications in the realm of credit scoring. The central premise of DA is that the factors explaining each category of the target variable adhere to a multivariate normal distribution with a shared variance-covariance matrix. DA's primary goal is to enhance the distinction among diverse groups while reducing the internal variation within each group, providing a noteworthy alternative to

logistic regression in the context of credit scoring and similar domains. LDA is mathematically represented by the equation [8],

$$d = \left((\mu_1 - \mu_2)' \frac{d}{\lambda}\right) \Sigma^{-1}(\mu_1 - \mu_2) = k \cdot \delta \#(1)$$

where k is a scalar.

Logistic Regression (LR) and Discriminant Analysis (DA) are two methods for credit risk assessment, each with its strengths under different conditions. Logistic regression performs exceptionally well when the dependent variable exhibits a binary or dichotomous nature, such as 'yes'/'no' or 'correct'/'incorrect' outcomes. In contrast, DA tends to deliver superior results when the dependent variable is multi-categorical, making it a preferred choice for assessing creditworthiness in such cases.

B. Regression

1. Linear Regression:

Linear regression analysis is frequently employed in credit scoring applications and is noteworthy for its applicability even when dealing with a twoclass response variable. This technique establishes a linear connection between the borrower attributes, represented as $X = \{X_1, \ldots, X_p\}$, and the target variable Y. The relationship is expressed as [9]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \#(2)$$

where ϵ represents random error and is independent of X. The conventional approach for estimating the coefficients $\beta = \beta_0, \ldots, \beta_p$ involves using ordinary least squares, represented as $\hat{\beta}$. When dealing with a binary response variable Y, the conditional expectation $E(Y \mid X) = x'\beta$ is utilized to classify borrowers into 'good' or 'bad' categories. However, due to the range of $-\infty < x'\beta < \infty$, the model's output cannot be interpreted as a probability. Linear regression has found applications in credit scoring, as indicated in [9] by various researchers.

2. Logistic Regression:

Logistic regression (LR) is a fundamental statistical model employed in the domain of classification and predictive analytics, proving particularly valuable when dealing with categorical target variables, making it a well-suited choice for binary classification tasks as noted in [10]. In this context, LR utilizes a linear model based on the sigmoid, also referred to as the logistic function, to estimate the probability of a binary variable, effectively determining the likelihood of a specific event occurring. Significantly, the outcome of this estimation is a probability, constraining the dependent variable to a range between 0 and 1 [11].

LR posits that a suitable function of the predicted event probability is linearly related to the observed values of the

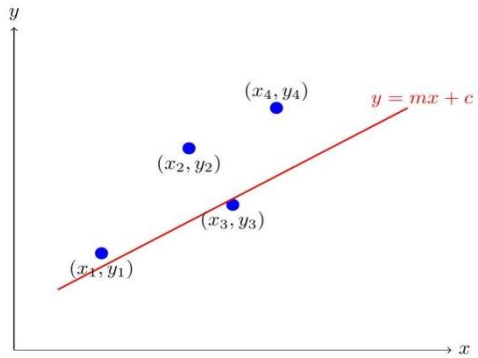

Fig 2. Linear Regression.
Source: Created by authors available explanatory variables, as discussed in [12].

One of its key advantages is its capacity to produce a straightforward probabilistic classification formula, allowing for individual examination of each feature and its corresponding coefficient. This enables a clear understanding of the factors contributing to the differentiation between creditworthy and noncreditworthy customers [10]. The LR model is mathematically defined as:

$$P(y = +1 \mid x) = \frac{1}{1 + \exp(\alpha_0 + \alpha^T x)} \#(3)$$

and

$$P(y = -1 \mid x) = 1 - P(y = +1 \mid x) = \frac{\exp(\alpha_0 + \alpha^T x)}{1 + \exp(\alpha_0 + \alpha^T x)} \#(4)$$

Here, $x \in R^n$ represents the feature vector, $P(y = +1 \mid x)$ denotes the probability of classifying x as a creditworthy borrower, $P(y = -1 \mid x)$ stands for the probability of classifying x as a non-creditworthy borrower, and the model parameters, $\{\alpha_0, \alpha\}$, are estimated through techniques like maximum likelihood estimation on the training dataset, as detailed in [13].

After estimating the model parameters, the decision pertaining to an input feature vector x is made in favor of $\hat{y} = +1$ if $P(y = +1 \mid x) \geq P(y = -1 \mid x)$, which is equivalent to the following decision rule [14] :

$$\hat{y} = \begin{cases} +1 & \text{for } 1 \geq \exp(\alpha_0 + \alpha^T x) \\ -1 & \text{otherwise} \end{cases} \#(5)$$

However, LR does have its limitations. It is particularly effective for linearly separable problems but may struggle when faced with non-linear complexities or interactive effects of explanatory variables [15]. Despite these shortcomings, LR remains a prevalent choice due to its ease of use and satisfactory predictive accuracy. Unlike other methods, it is highly interpretable, allowing for human-readable insights, and it is widely adopted for its simplicity and practicality.

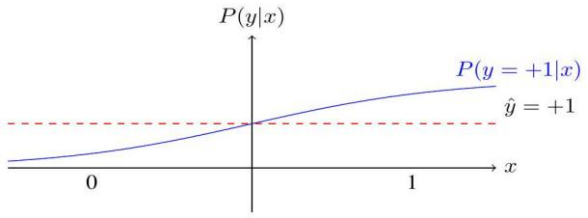

Fig 3. Logistic Regression. Source: Created by authors

C. Naive Bayes

The Bayesian classification algorithm, rooted in Bayesian theorem, is a statistical method for probabilistic classification that make effective use of Bayes' theorem, incorporating the "naive" assumption of conditional independence between all pairs of features given the class variable [16]. This approach facilitates the classification of data by establishing a probabilistic model rooted in the structure of Bayesian networks, which elucidate the conditional dependence relationships among a series of random variables, while adhering to Bayes' theorem.

One of the notable features of Naive Bayes is its applicability to both continuous and categorical data, making it a versatile tool for classification tasks [17]. During the learning process, Naive Bayes constructs a decision tree from training samples, which are labeled with their respective classes. In this model, "observations" are characterized as "conjunctions of features," while "corresponding target values" are represented as "leaf nodes." Notably, Naive Bayes with standard entropy criteria is employed to categorize continuous attributes as categorical, differing from a traditional approach where only a single class is considered [12].

Naive Bayesian classification, among the simplest and most widely used Bayesian classifiers as stated in [16], offers a straightforward approach to construct a probability model for estimating the posterior probability, $P(y \mid x)$, with the goal of classifying testing samples, denoted as x. It is termed "naïve" because the Bayesian classifier rests on the most basic and simplistic assumptions: that the features of samples are independent. This assumption, is mathematically expressed as

$$P(C_k \mid x) = \frac{P(x \mid C_k) \cdot P(C_k)}{P(x)} \#(6)$$

D. Linear Programming

As described in [18], one of the advantages of utilizing Linear Programming (LP) methods in credit scoring is their adaptability in incorporating domain expertise or pre-existing biases by introducing supplementary constraints. A very popular formulation goes as follows [19]:

$$\begin{aligned} & \min_{w,g} \sum_{i=1}^{N} \xi^2 \\ \text{subject to} \quad & w^T x_i \geq c - \xi \quad y_i = +1 \\ & w^T x_i \leq c + \xi \quad y_i = -1 \\ & \xi \geq 0, \quad i = 1, \ldots, N \end{aligned} \#(7)$$

whereby **n** represents the vector of x_i values. In the first set of inequalities, an effort is made to distinguish positive cases, and in the second set, negative cases, by assigning them a score denoted as $w^T x_i$ that surpasses or falls below a predefined threshold value, such as $c = 1$. To address potential misclassifications, positive slack variables x_i are introduced. The primary aim is to minimize the count of misclassifications by reducing the total sum of these slack variables x_i [18]. It is worth noting that variations of this approach have been proposed in existing literature. For instance, Glen's work [20] introduced a mixed-integer programming method for classification. A notable advantage of applying Linear Programming (LP) techniques in credit scoring is their versatility in accommodating domain-specific knowledge or a priori biases through the inclusion of extra constraints [19].

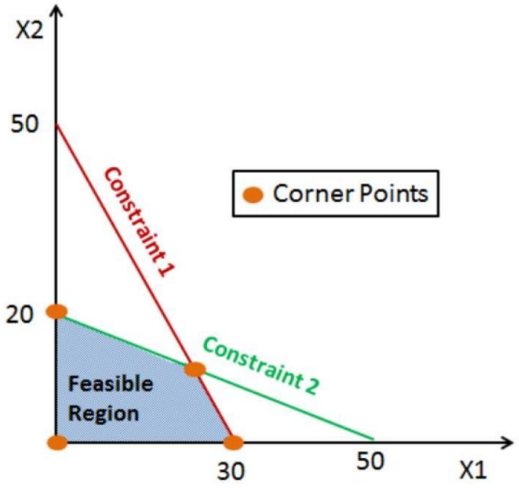

Fig 4. Linear Programming [21].

E. Decision Trees

Decision Trees (DT) are tree-structured classifiers widely used for classification and regression tasks [11], a classification method that constructs decision rules from historical data, forming tree-like structures. The primary objective is to establish a set of if-then logical conditions for predictive or classification purposes [9]. They are built fast and are easy to interpret [22] and their computational complexity is low [16].

They employ "if" statement-like branches to split datasets based on their ability to classify data accurately. The tree comprises internal nodes, branches, and leaf nodes, representing dataset features, decision rules, and outcomes or classes, respectively. At each node, the classifier selects the feature that best separates the data into classes, akin to an "if" statement. To prevent overfitting, tree depth is typically limited [23]. Decision trees offer a powerful and interpretable supervised learning approach, with each internal node corresponding to an input attribute and each leaf node assigned a class or probability distribution [11].

Three commonly employed tree algorithms are the chisquare automatic interaction detector CHAID, classification and regression tree CART, and C5, which vary based on the tree construction criterion. CART employs the Gini index as the splitting criterion, while C3, C4, C5 use entropy, and CHAID relies on the chi-square test for tree construction, as discussed in [9]. Among these, C4.5 stands out as one of the most popular, as noted in [16]. It is defined as:

$$E(S) = -\sum_{i=1}^{c} p_i \log_2 p_i \#(8)$$

Where $E(S)$ represents the entropy of a set S, c is the number of classes in the dataset and p_i is the proportion of instances in class i within set S.

$$\text{Gain}(S, A) = E(S) - \sum_{v \in V} \frac{|S_v|}{|S|} E(S_v) \#(9)$$

Where $\text{Gain}(S, A)$ is the gain achieved by splitting set S using attribute A, $E(S)$ is the entropy of set S, S_v is a subset of S for each value v of attribute A and $|S|$ is the total number of instances in set S.

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInformation}(S, A)} \#(10)$$

Here, $\text{GainRatio}(S, A)$ represents the gain ratio for attribute A on set S, $\text{Gain}(S, A)$ denotes the gain resulting from splitting set S using attribute A, and $\text{SplitInformation}(S, A)$ quantifies the potential

information generated by the split involving attribute A. In the context of **C4. 5**, it prioritizes splits with the highest Gain-Ratio, subject to the condition that the information gain must be at least as significant as the average gain observed across all examined splits, as outlined in [18].

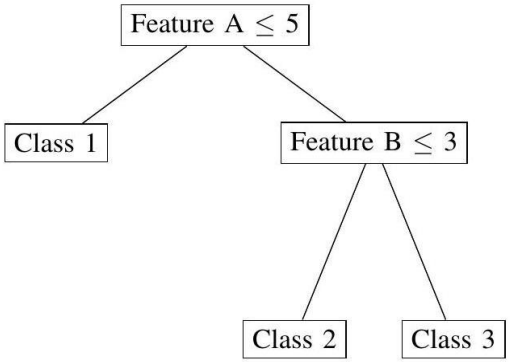

Fig 5. Simple Decision Tree. Source: Created by authors

F. Support Vector Machines

The Support Vector Machine (SVM) is a statistical classification approach that aims to identify an optimal hyperplane for separating binary classified data, with the objective of minimizing the margin between the hyperplane and the classes. In SVM construction, a hyperplane or a set of hyperplanes can be employed for various tasks, including classification and regression, especially in high-dimensional or infinite dimensional scenarios, as discussed in [16], [13], [24], [9], and [11].

For a given training set comprising l sample points, denoted as $(1, x_1, y_1), (2, x_2, y_2), \dots, (l, x_l, y_l)$, where x_i represents an input vector with m-dimensional features, and $y_i \in -1,1$ is the corresponding class label, SVM seeks a classification mapping $y(x) = \mathrm{sgn}(w \cdot x + b)$ by solving a quadratic programming problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i^2 \\ \text{such that} \quad & y_i (f w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,2, \dots, l \\ & \xi_i \geq 0, \quad i = 1,2, \dots, l \end{aligned} \qquad \#(11)$$

The parameter pair (w, b) comprises the weight vector and the bias term, while non-negative variables ξ_i are introduced to account for cases where data points are not linearly separable, as indicated in [24].

However, conventional SVMs still exhibit several limitations, including high computational complexity, sensitivity to outliers, and interpretability challenges. The standard SVM model assumes that all features contribute equally to the final classification, which does not align with the reality that different features have varying impacts on data classification, as discussed in [24].

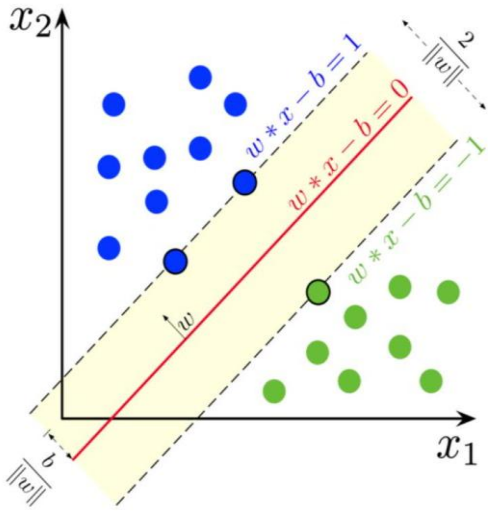

Fig 6. SVM representation in two dimensions.
Source: [25]

G. K-Nearest Neighbors

K-Nearest Neighbor (KNN) classifier calculates the similarity between the new data and its neighboring k data points [23]. It is a versatile and widely applied supervised machine learning technique capable of resolving both classification and regression tasks. This method operates on the principle of learning by analogy, searching for the K-nearest data points to an unknown sample within the pattern space based on a specified distance metric, typically Euclidean distance. Subsequently, the majority class among these K-neighbors determines the class of the unknown data point. KNN is celebrated for its simplicity, theoretical transparency, and nonparametric nature, making it suitable for scenarios with categorical variables, though numerical variables are often normalized to eliminate scale disparities [11].

KNN classifier's function memory-based, demanding no model establishment before classification. One of the primary advantages of KNN is its capacity to operate without the need for a pre-defined predictive model. However, it does not provide a straightforward classification probability formula. The predictive accuracy of KNN is notably influenced by the choice of distance metric and the cardinality of the neighborhood, K, which affects the accuracy of majority voting [26]. Despite these considerations, KNN remains a widely employed technique in various domains, from genetics to recommendation systems, owing to its adaptability and straightforward implementation.

H. Artificial Neural Networks

An Artificial Neural Network (ANN) is inspired by biological neural networks in animal brains except that the information-processing units in a neural network are artificial [19] [27] [28]. These artificial neurons are connected through edges, similar to synapses in biological brains, enabling the transmission of signals between them. An artificial neuron receives incoming signals, processes them, and can, in turn, send signals to connected neurons. These signals are represented as reals, and each neuronal output is determined through a nonlinear function applied to the sum of its inputs. Neurons and edges possess weights that are adjusted as the network learns, affecting the strength of signal transmission. Additionally, neurons may have a threshold, ensuring a signal is only sent if the cumulative signal surpasses this threshold.

In typical neural networks, neurons are organized into layers, each potentially performing unique transformations on their inputs. Signals originate from the input layer, traverse through intermediary layers, and ultimately reach the output layer, potentially undergoing multiple iterations through the layers. The network's design allows for complex data transformations and learning patterns through the adjustment of weights and signal thresholds. The application of ANN in credit scoring was first introduced by [29].

For a given input feature vector $\mathbf{x}$, a three-layer ANN computes the output $\hat{y}$ according to:

$$\hat{y} = a_2\left(a_1\left(\alpha^{(1)}\mathbf{x} + \alpha_0^{(1)}\right)\alpha^{(2)} + \alpha_0^{(2)}\right) \#(12)$$

where $\alpha_0^{(1)}, \alpha^{(1)}, \alpha_0^{(2)}, \alpha^{(2)}$ are weights, and a_1, a_2 are activation functions between the input and hidden layer, and between the hidden layer and the output layer, respectively. The parameters are learned through a training set. The ANN makes the final decision by applying a decision function, such as Soft-Max, to $\hat{y}$ [13]. One needs to be careful about Neural networks with too many weights will overfit the data at the global minimum of the error and generalization will not be good [15].

An overall approach of this type of model is shown in Fig. 7. The model consists of several input values connected to hidden nodes through weighing factors. The hidden nodes sum the input values using weighing factors that are determined during the training step. The weighing factors imitate the strength of the neural connections. Each hidden node can be biased by an activating signal such that it produces an output only if the sum of the input signals exceeds a pre-set threshold condition.

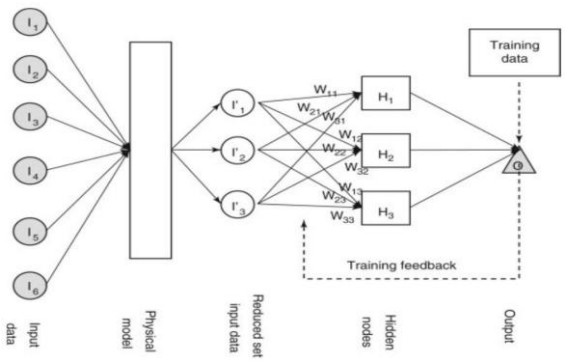

Fig 7. Structure of an ANN. Source: [30]

I. Ensemble Methods

The process of merging predictions from multiple classifiers to create a single, more accurate classifier is referred to as ensemble learning. Both theoretical analyses and practical experiments have demonstrated that a well-constructed ensemble outperforms individual classifiers. This is primarily because ensembles effectively leverage the diversity of errors exhibited by individual classifiers across different regions of the input space. In a manner akin to human decision-making, where we consider and weigh various opinions before arriving at a decision, ensembles harness the strengths of multiple classifiers to enhance predictive accuracy [31] [32]. Common ensemble types include the Bayes optimal classifier, Bootstrap aggregating (bagging) [33], Boosting [34], Stacking, Voting, each with its own unique characteristics and applications. To mitigate the risk of overfitting, diversity is introduced into the process through randomness. This diversity is achieved in two main ways. Firstly, each decision tree is created using a process called "bootstrapping," which involves presenting each tree with a random sample of the training set. Secondly, additional randomness is introduced by randomly selecting which inputs each tree considers during training, especially at the stage of partitioning. The essential aspect of this approach lies in the differences among the resulting decision trees and their collective performance as individual base models. This diversity, introduced by bootstrapping and random input selection, plays a crucial role in the ensemble's effectiveness [35].

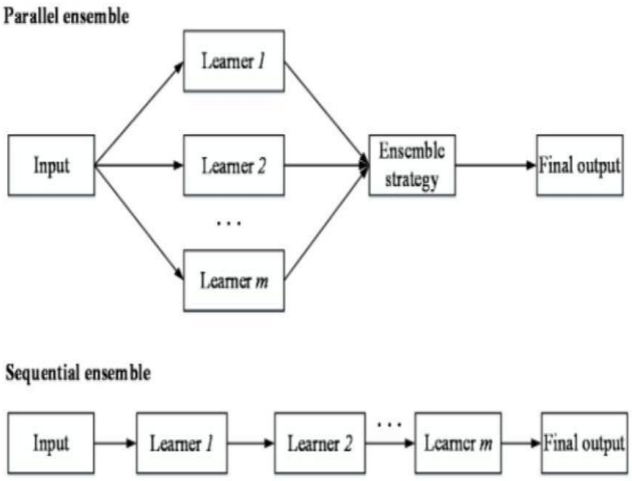

Fig 8. Flowchart of Parallel and Sequential ensemble [16].

1. Bagging

This approach is commonly referred to as Bootstrap AGGregatING. Bagging involves generating M samples, denoted as $S_1, \dots, S_m$, which are randomly selected from the original training set with replacement. Each sample S_i, where $1 \leq i \leq M$, matches the size of the original set. Consequently, within each sample S_i, some instances may appear more than once, while others may not appear at all. A distinct classifier C_i is trained using each sample S_i, for all $i = 1, \dots, M$. As each classifier is constructed from a different training set, the learners often exhibit variability. To classify an instance, the predictions generated by the classifiers are combined through a majority voting mechanism, as discussed in [36].

As demonstrated in [33], Bagging proves to be effective when applied to "unstable learning algorithms," where minor alterations in the training set lead to significant fluctuations in predictions.

2. Random Forest

A Random Forest (RF) is an ensemble of decision trees for classification and regression, incorporating randomness to select the best feature subsets rather than making arbitrary choices. The Random Forest classifier constructs multiple trees using a randomization technique.It gathers decision trees through random data selection and determines classification based on majority votes, effectively reducing overfitting and enhancing overall accuracy.

To make an overall classification, RF employs a majority vote based on the decision trees' outputs [13]. The RF method, proposed by [37], involves multiple decision trees generated through bagging and random-subspace methods. Each decision tree contributes a vote to predict the test set, and the majority voting strategy combines the results. RF offers various benefits, including the ability to handle high-dimensional data, maintaining some interpretability through feature importance analysis, and low computational cost. It can be adapted to parallel computing, making it a valuable ensemble method [38].

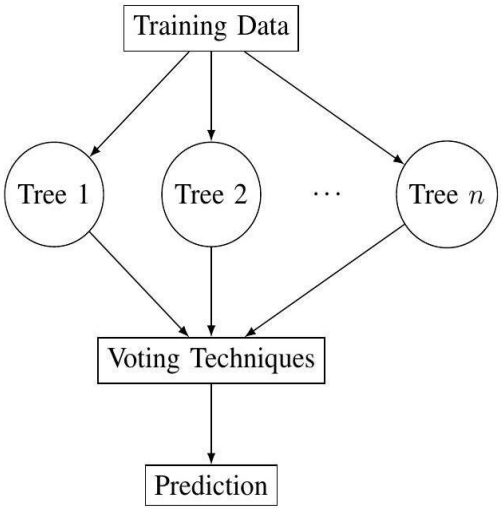

Fig 9. Random Forest Approach. Source: Created by authors

RF outperforms Decision Trees by employing an ensemble of decision trees through majority voting, which enhances its predictive accuracy while simultaneously mitigating the issue of overfitting [12]. RF constructs this ensemble using the bagging method and decision trees as base classifiers. This approach bolsters robustness by altering data samples and randomly selecting input features, enabling it to deliver a collective decision based on votes from the ensemble of classifiers [12]. Furthermore, RF excels when dealing with high-dimensional data, thanks to its ability to operate on data subsets. It boasts faster training than Decision Trees, focusing on a subset of features [23]. The effectiveness of Random Forest depends on the strength and interdependence of each classifier, with each tree contributing to the final prediction based on dataset-specific characteristics [11].

3. Boosting

Boosting is a machine learning technique that iteratively trains models, giving greater emphasis to data points that were misclassified by earlier models. It operates by sequentially constructing multiple models and adjusting the weights of data instances. The process initiates with a less robust model, frequently a shallow decision tree, and subsequently constructs an improved model that rectifies the mistakes of the preceding model. Instances inaccurately categorized by the prior model receive heightened emphasis in the subsequent stages. Initially, all data points are assigned equal weight D1 and used to create a base model M1. Data points that M1 misclassifies are then assigned higher weights, creating a boosted dataset D2, which is used to train a second base model M2, and so on. Inferences are made by combining the predictions through a voting process. A widely used boosting technique is Adaptive Boosting (Ada-boost).

a. AdaBoost: Adaptive Boosting (AdaBoost), a widelyused variant of the Boosting algorithm, was introduced by Freund and Schapire [34]. AdaBoost, particularly known for its use of the exponential loss function and forward stage-wise additive modeling, leads to significant performance improvements for several reasons [16].

Firstly, AdaBoost effectively reduces the misclassification rate of the final classifier by combining multiple base classifiers, even if individual base classifiers exhibit high misclassification rates. Secondly, the ensemble generated by AdaBoost maintains a substantially lower variance compared to that of weak base learners. Despite these advantages, AdaBoost may sometimes fail to enhance the base inducer's performance, and this is typically attributed to overfitting [32].

To counter overfitting, it is crucial to select weak classifiers that maximize the diversity within the ensemble. If two weak learners produce highly similar outputs, it can be beneficial to remove one of them and increase the weight of the remaining weak learner [39].

In this context, the problem involves a collection of m labeled training examples, represented as $(x_1, y_1), \ldots, (x_m, y_m)$, with the x_i belonging to a specific domain X, and the labels y_i taking values in the set $\{-1,1\}$. For each iteration within the range of $t = 1,2, \ldots, T$, a distribution D_t is established over the m training examples. Subsequently, a given weak learner or weak learning algorithm is employed to determine a weak hypothesis $h_t: X \to \{-1,1\}$. The primary goal of the weak learner is to identify a weak hypothesis with a low weighted error ϵ_t concerning D_t," as elucidated in [40].

The final or combined hypothesis H calculates the sign of a weighted combination of weak hypotheses, given by:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

This can be equivalently expressed as asserting that H is computed as a weighted majority vote among the weak hypotheses h_t, with each hypothesis being assigned a weight α_t.

An advantageous characteristic of AdaBoost is its capacity to detect outliers, which are examples that may either be mislabeled in the training data or inherently ambiguous and challenging to categorize, as noted in [34].

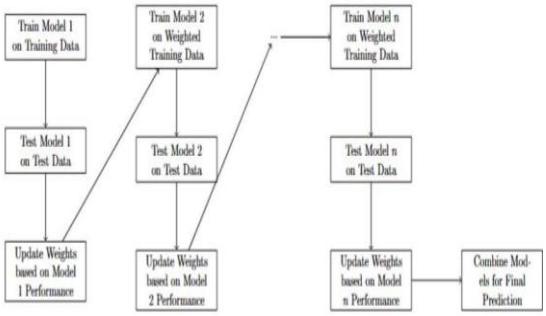

Fig 10. AdaBoost Sequential Model Execution

b. XGBoost: XGBoost [41], an abbreviation for "extreme gradient boosting," is widely recognized for its exceptional processing speed and performance [13]. While it shares similarities with gradient boosting, XGBoost distinguishes itself by constructing decision trees in parallel, as opposed to the sequential construction of trees [42]. It employs gradient descent to iteratively enhance the performance of weak treebased learners, amalgamating numerous decision tree models into a highly effective composite model [23]. A key advantage of XGBoost lies in its ability to perform parallel computations on a single machine, rendering it a favorable choice for constructing accurate models on extensive datasets without necessitating access to extensive computational resources. In comparison to Gradient Boosting, XGBoost excels in both performance and speed, achieving this optimization through distributed implementation that focuses on an objective function encompassing a loss function and regularization term [38].

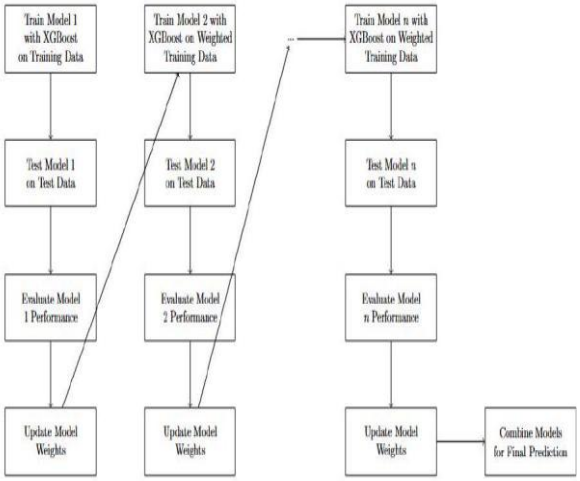

Fig 11. XGBoost Sequential Model Execution.
Source: Authors

3. Comparative Analysis and Proposed Methodology

A. Empirical Design

In order to perform a fair evaluation of the effectiveness of five distinct algorithms, specifically Logistic Regression, Support Vector Machine (SVM), Naive Bayes, Random Forest, and XGBoost, the complete experiment was implemented using Python 3.6 on Google Colab with a T4 GPU on a Windows-based system.

Datasets Used:Credit datasets from both Australia and Germany, which are frequently encountered in the field of credit scoring research and are readily accessible from the UC Irvine (UCI) Machine Learning Repository.
Implementation: Commonly used Python libraries, such as XGBClassifier for XGBoost, and scikit-learn (sklearn) for all other algorithms, were employed in the study. In the subsequent phase, the model development endeavor took center stage. We embarked on an exploration of various machine learning algorithms to uncover their predictive potential. Logistic Regression, a model well-suited for binary classification tasks, was harnessed to kick off our analysis. An in-depth investigation into its parameters and fine-tuning enabled us to attain an accuracy rate of 86.67%, signifying its efficacy in the context of this problem. Parallelly, the KNN algorithm

Table 1: Results for Australian Dataset

Models/Indicator	Accuracy	Precision	Recall	F1-Score
Logistic	0.8623	0.8076	0.8235	0.8155
SVM	0.8406	0.7377	0.8823	0.8036
Naive Bayes	0.8261	0.8649	0.6275	0.7273
RF	0.8768	0.84	0.8235	0.8317
XGBoost	0.8551	0.8039	0.8039	0.8039

Table 2: Results for German Dataset

Models/Indicator	Accuracy	Precision	Recall	F1-Score
Logistic	0.775	0.6458	0.5254	0.5794
SVM	0.77	0.6066	0.6271	0.6167
Naive Bayes	0.735	0.5385	0.7119	0.6131
RF	0.775	0.7692	0.339	0.4706
XGBoost	0.805	0.7174	0.5593	0.6286

was enlisted, with a keen focus on identifying the optimal value of ' K,' representing the number of neighbors considered for classification. This meticulous approach led to an accuracy rate of 84.44%, further bolstering this study's repository of predictive tools. SVM, a robust classifier, entered the fray, flexing its classification muscles with an accuracy of 84.44%. The kernel choice, an RBF kernel, played a pivotal role in shaping the model's performance. Naive Bayes, both Categorical and Gaussian variations, were subjected to scrutiny. Categorical Naive Bayes emerged as a robust performer with an accuracy of 84.44%, while Gaussian Naive Bayes exhibited reasonable predictive power with a 71.11% accuracy.
A decision tree model, dynamically optimized for maximum leaf nodes, yielded a commendable accuracy of 84.44%. Simultaneously, RF, a collection of decision trees, displayed its prowess, attaining an accuracy of 86.67%. The iterative manipulation of leaf nodes and the incorporation of 1000 estimators were key contributors to its success. Lastly, Gradient Boosting, a boosted ensemble technique, was unveiled. Leveraging Randomized Search optimization, the model reached an accuracy of 77.78%. This panorama of machine learning models, with their respective accuracies, offered a comprehensive view of the landscape, providing valuable insights for prospective model selection. In the denouement of this phase, a meticulous comparison of these models unveiled Logistic Regression and RF as front-runners, each clocking an accuracy of 86.67%. These findings underscore the pivotal role of judicious data preprocessing and algorithmic selection in the realm of loan approval prediction.
Four comprehensive evaluation model indicators were used to evaluate the model performance, namely Accuracy, Precision, Recall, F1 Score. They are summarized in I and II

4. CONCLUSION

In conclusion, the comparative analysis of five credit scoring models on both the Australian and German datasets has provided valuable insights into their performance. On the
Australian dataset, the Random Forest model exhibited the highest accuracy (88%), while Naive Bayes showed the best precision (86%) and F1 score (83%). Support Vector Machine displayed the highest recall (88%). However, on the German dataset, XGBoost stood out with the highest accuracy (80.5%), while Random Forest achieved the best precision (76.9%) and F1 score (62.8%). Support Vector Machine demonstrated the highest recall (62.7%). These findings underscore the variations in performance specific to each model when applied to diverse datasets, underscoring the significance of meticulously choosing the right model for credit scoring applications, taking into account the unique characteristics of the dataset and the preferred evaluation metrics. Additionally, this research enhances comprehension of

the pros and cons of different machine learning models within the realm of credit risk assessment, providing valuable insights for financial institutions and lending platforms to inform their decision-making processes.

5. Future Scope

In the complex landscape of loan applications, ambiguity surrounding approval or rejection decisions breeds frustration and mistrust among applicants. To address this issue, this solution envisions a fusion of BigQuery, H2O Driverless AI, and PaLM from Google AI. BigQuery serves as the foundation, creating a robust repository for diverse loan denial data. H2O AI takes centre stage, utilizing algorithms like logistic regression and gradient boosting to meticulously analyse the dataset, predicting approvals and denials with remarkable precision. The key innovation lies in transparency, as H 2 O AI's Explainability feature unveils the inner workings of the model, providing crystal-clear explanations for application outcomes. Integrating seamlessly with BigQuery, this system enables real-time predictions and explanations, empowering users with immediate insights. PaLM, the linguistic prodigy, refines explanations through natural language processing, identifying patterns and biases. This triad aims to revolutionize loan decisions, fostering trust, financial literacy, and a more equitable financial landscape.

REFERENCES

[1] "Fico," https://www.fico.com/en, accessed : 23-11-2023.
[2] A. K. Reichert, C.-C. Cho, and G. M. Wagner, "An Examination of the Conceptual Issues Involved in Developing CreditScoring Models," J. Bus. Econom. Statist., pp. 101-114, July 2012. [Online]. Available: https://www.tandfonline.com/doi/epdf/10.1080/07350015.1983.10509329?needAccess=true
[3] M. Bücker, G. Szepannek, A. Gosiewska, and P. Biecek, "Transparency, auditability, and explainability of machine learning models in credit scoring," J. Oper. Res. Soc., pp. 70-90, June 2021.
[4] R. A. Fisher, "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS," Annals of Eugenics, vol. 7, no. 2, pp. 179-188, Sept. 1936.
[5] R. A. Eisenbeis, "Problems in applying discriminant analysis in credit scoring models," p. 205, 1978.
[6] D. J. Hand and W. E. Henley, "Statistical classification methods in consumer credit scoring: a review."
[7] D. J. Hand, R. Jonathan, J. Olivers, and A. D. Lunnr, "Discriminant analysis when the classes arise from a continuum linear discriminant analysis classification rule optimal decision surface error rate," pp. 641650, 1998
[8] "Kapitel 7 discriminant analysis."
[9] F. Louzada, A. Ara, and G. B. Fernandes, "Classification methods applied to credit scoring: Systematic review and overall comparison," pp. 117-134, 122016.
[10] J. P. Barddal, L. Loezer, F. Enembreck, and R. Lanzuolo, "Lessons learned from data stream classification applied to credit scoring," Expert Systems with Applications, vol. 162, 122020.
[11] S. Varun, A. Theagarajan, and M. Shobana, "Credit score analysis using machine learning." Institute of Electrical and Electronics Engineers Inc., 2023.
[12] D. Tripathi, A. K. Shukla, B. R. Reddy, G. S. Bopche, and D. Chandramohan, "Credit scoring models using ensemble learning and classification approaches: A comprehensive survey," pp. 785-812, 32022.
[13] X. Dastile, T. Celik, and M. Potsane, "Statistical and machine learning models in credit scoring: A systematic literature survey," Applied Soft Computing Journal, vol. 91, 62020.
[14] V. B. Djeundje, J. Crook, R. Calabrese, and M. Hamid, "Enhancing credit scoring with alternative data," Expert Systems with Applications, vol. 163, 12021.
[15] B. Chitambira, "Credit scoring using machine learning approaches."
[16] Y. Li and W. Chen, "A comparative performance assessment of ensemble learning for credit scoring," Mathematics, vol. 8, pp. 1-19, 102020.
[17] D. Tripathi, B. R. Reddy, and A. K. Shukla, "Cfr: collaborative feature ranking for improving the performance of credit scoring data classification," Computing, vol. 104, pp. 893-923, 42022.
[18] B. Baesens, T. V. Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," Journal of the Operational Research Society, vol. 54, pp. 627-635, 2003.
[19] L. C. Thomas, "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers," International Journal of Forecasting, vol. 16, no. 2, pp. 149-172, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207000000340

[20] J. J. Glen, "Classification accuracy in discriminant analysis: A mixed integer programming approach," The Journal of the Operational Research Society, vol. 52, no. 3, pp. 328-339, 2001. [Online]. Available: http://www.jstor.org/stable/254070
[21] A. Navlani, "Linear Programming with Spreadsheets," DataCamp, July 2019. [Online]. Available: https://www.datacamp.com/tutorial/ linear-programming-with-spreadsheets
[22] S. T. Moe and T. T. Nwe, "A hybrid approach of logistic regression with grid search optimization in credit scoring modeling for financial institutions," vol. 2023-February. IEEE Computer Society, 2023, pp. 62 − 66.
[23] S. Bhatia, "Pragmatic segmentation-based credit risk management using machine learning." Institute of Electrical and Electronics Engineers Inc., 2022.
[24] A. I. C. Conference, E. Engineering, C. . .-. S. I. C. on Mechatronic Sciences, M. . .-. Shenyang, E. Engineering, and C. . .-. S. I. C. on Mechatronic Sciences, Proceedings / 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC) 20-22 Dec. 2013, Shenyang, China.
[25] Contributors to Wikimedia projects, "Support vector machine - Wikipedia," Nov. 2023, [Online; accessed 27. Nov. 2023]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=1183475870
[26] I.-C. Yeh and C. hui Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," Expert Systems with Applications, vol. 36, no. 2, Part 1, pp. 2473-2480, 2009. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S0957417407006719
[27] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," Expert Systems with Applications, vol. 83, pp. 405-417, 2017. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S0957417417302415
[28] C.-F. Tsai, Y.-F. Hsu, and D. Yen, "A comparative study of classifier ensembles for bankruptcy prediction," Applied Soft Computing, vol. 24, pp. 977-984, 112014.
[29] M. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in 1990 IJCNN International Joint Conference on Neural Networks, June 1990, pp. 163-168 vol.2.
[30] "Artificial Neural Network Model - an overview | ScienceDirect Topics," Nov. 2023, [Online; accessed 27. Nov. 2023].
[31] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study, " Journal of Artificial Intelligence Research, vol. 11, no. 1, pp. 169-198, 1999
[32] L. Rokach, "Ensemble-based classifiers," Artificial Intelligence Review, vol. 33, pp. 1-39, 2010.
[33] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, pp. 123140, 1996.
[34] Y. Freund and R. E. Schapire, "A short introduction to boosting," Journal of Japanese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771780, September 1999, (In Japanese, translation by Naoki Abe.).
[35] B. R. Gunnarsson, S. vandenBroucke, B. Baesens, M. Óskarsdóttir, and W. Lemahieu, "Deep learning for credit scoring: Do or don't?" European Journal of Operational Research, vol. 295, pp. 292-305, 112021.
[36] S. Moral-Garcia and J. Abellan, "Improving the results in credit scoring by increasing diversity in ensembles of classifiers," IEEE Access, vol. 11, pp. 58451 − 58461,2023.
[37] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5 − 32, Oct. 2001.
[38] Y. Jin, W. Zhang, X. Wu, Y. Liu, and Z. Hu, "A novel multi-stage ensemble model with a hybrid genetic algorithm for credit scoring on imbalanced data," IEEE Access, vol. 9, pp. 143 593-143 607, 2021.
[39] C. Tamon and J. Xiang, "On the boosting pruning problem," in Machine Learning: ECML 2000, R. López de Mántaras and E. Plaza, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 404-412.
[40] R. E. Schapire, Explaining AdaBoost. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 37-52. [Online]. Available: https: //doi.org/10.1007/978-3-642-41136-6_5
[41] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785794.
[42] J. Nobre and R. F. Neves, "Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets," Expert Systems with Applications, vol. 125, pp. 181-194, 2019.