Information Technology in Economics and Business
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8943
RESEARCH ARTICLE
Vol.1 / No.1 / 2024 / pp.7-13
https://doi.org/10.69882/adba.iteb.2024072

Enhancing Financial Decision-Making: Predictive Modeling for Personal Loan Eligibility with Gradient Boosting, XGBoost, and AdaBoost

Cem Özkurt*,[1]

*Sakarya University of Applied Sciences, Artificial Intelligence and Data Science Application and Research Center, 54050, Sakarya, Türkiye.

ABSTRACT This study aims to improve the prediction of personal loan eligibility through the application of advanced machine learning techniques. Accurate prediction of creditworthiness is crucial for financial institutions to mitigate risks and optimize their lending processes. We evaluated three algorithms Gradient Boosting, XGBoost, and AdaBoost using a comprehensive dataset containing demographic and banking information. Among these, XGBoost proved to be the most effective model, achieving an accuracy of 0.95, precision of 0.95, recall of 0.95, and an F1 score of 0.95. These results demonstrate XGBoost's superior ability to accurately identify individuals likely to repay loans, making it an invaluable tool for enhancing decision-making in loan approvals. By leveraging XGBoost, banks can reduce the risk of defaults, streamline their operations, and provide better customer service, ultimately leading to more efficient and reliable lending strategies.

KEYWORDS
Loan eligibility prediction
Machine learning techniques
XGBoost
Creditworthiness

INTRODUCTION

In recent years, the ability to accurately predict financial behaviors and needs has become increasingly crucial in the banking sector. One key area where predictive modeling can have a significant impact is in the assessment of individual creditworthiness. With the growing volume of data available on customers, financial institutions can leverage advanced machine learning techniques to predict whether an individual is likely to qualify for a personal loan. This not only enhances the efficiency of the loan approval process but also helps in tailoring financial products to better meet customer needs. In this study, we focus on developing a predictive model that analyzes customer data to determine the likelihood of an individual securing a personal loan.

By training our model on a comprehensive dataset that includes various customer attributes, we aim to provide banks with actionable insights into their clientele. This allows for more informed decision-making regarding loan approvals, ultimately improving customer satisfaction and financial outcomes. Our model demonstrates impressive performance metrics, achieving an accuracy of 0.95, an F1 score of 0.78, a recall of 0.89, and a precision of 0.94. These metrics highlight the model's ability to accurately classify customers who are likely to obtain a personal loan while minimizing both false positives and false negatives.

Manuscript received: 26 July 2024,
Revised: 27 July 2024,
Accepted: 30 July 2024.

[1]cemozkurt@subu.edu.tr (**Corresponding author**)

The high accuracy and precision indicate that the model effectively identifies creditworthy individuals, whereas the high recall suggests that it captures a significant proportion of potential loan candidates. By leveraging these insights, financial institutions can better assess the creditworthiness of their customers, streamline the loan approval process, and offer personalized financial solutions. This research not only contributes to the field of predictive analytics but also offers practical applications that can enhance the operational efficiency and customer service in the banking sector. In this study, we utilize various machine learning techniques to train and evaluate our model, including Gradient Boosting, XGBoost, and AdaBoost. Each technique is assessed for its effectiveness in predicting personal loan eligibility, providing a comprehensive understanding of the strengths and limitations of different approaches. Through this analysis, we aim to deliver a robust and reliable tool for predicting loan eligibility, ultimately supporting banks in making data-driven decisions and optimizing their lending strategies.

Machine learning techniques applied to financial market prediction have been extensively researched due to the complex nature of financial time series, which are non-linear, dynamic, and chaotic. Among the most studied models are support vector machines (SVMs) and neural networks, particularly for the North American market (Henrique *et al.* 2019). The ability to predict financial crises and business failures is crucial for financial institutions. Research has extensively explored bankruptcy prediction and credit scoring using machine learning techniques, such as neural networks and decision trees, revealing current achievements and limitations, and suggesting future research directions (Lin *et al.* 2011).

In financial risk management, machine learning has been increasingly adopted to avoid losses and maximize profits. Recent studies have provided a systematic survey of machine learning applications in financial risk management, highlighting significant publications, identifying major challenges, and pointing out emerging trends (Mashrur *et al.* 2020). Evaluating simple machine learning models for financial trading, particularly in the FOREX market, has shown that these models can achieve profitable trading. This research emphasizes the importance of attribute selection, periodic retraining, and appropriate training set size to enhance the classification capabilities of these models (Gerlein *et al.* 2016) .

Forecasting directional movement of stock prices using machine learning tools has been a significant research focus. Studies comparing stock price and return as input features have found that stock price is generally a more potent feature for predicting price movement, especially when combined with technical indicators (Kamalov *et al.* 2021). The integration of machine learning in financial planning has been limited but notable in areas such as high-frequency trading and credit scoring. Significant financial decisions often rely on formal decision models like the Markowitz portfolio model, which have been enhanced over time to incorporate machine learning advancements (Mulvey 2017).

Research on machine learning techniques in financial markets has covered various areas, including stock market forecasting, risk management, and debt management. Different algorithms and platforms have been evaluated for their accuracy, efficiency, speed, and usability, with particular attention to techniques involving neural networks and support vector machines (Vats and Samdani 2019).

MATERIAL AND METHODS

This study was conducted using the "Bank Personal Loan Dataset" available on Kaggle. The dataset contains various demographic and banking data used by banks to evaluate personal loan offers to their customers. It comprises a total of 5000 customer records and 14 variables. The variables include age, years of experience, annual income, ZIP code, family size, education level, mortgage amount, securities account, CD account, online banking services, credit card usage, tenure with the bank, and loan default status. Initially, data cleaning and preprocessing steps were performed on the dataset, filling in missing values and converting categorical variables to numerical values using one-hot encoding. The features were standardized to enhance the model's performance. This dataset was analyzed using Logistic Regression, K-Nearest Neighbors (KNN), and Naive Bayes algorithms, and the performance of the models intended for evaluating personal loan offers was assessed.

Table 1 Dataset example

Age	Experience	Income	Family	Education	Credit Card
25	1	49	4	1	0
39	15	11	1	1	0
35	8	45	4	2	1

Machine Learning

Machine learning has become a pivotal technology across various domains due to its ability to process and analyze large datasets, enabling predictions and decision-making with minimal human intervention. It is defined as a subset of artificial intelligence that allows systems to learn from data, identify patterns, and make decisions without being explicitly programmed for specific tasks. Zhou (2021) provides an in-depth exploration of machine learning, discussing fundamental concepts and various methodologies used to train models effectively (Henrique *et al.* 2019).

The application of machine learning and deep learning techniques has been extensively reviewed across different industries, revealing their impact and versatility. According to Shinde and Shah (2018), these techniques are being applied in fields such as healthcare, automotive, and finance, providing enhanced capabilities in pattern recognition, natural language processing, and autonomous systems (Lin *et al.* 2011). The integration of these technologies has enabled systems to perform tasks such as diagnosing diseases, navigating vehicles, and personalizing marketing strategies.

In the financial sector, machine learning techniques are employed to predict market trends, optimize portfolios, and manage risks. Henrique, Sobreiro, and Kimura (2019) discuss various machine learning models used for financial market prediction, emphasizing their ability to analyze complex datasets and identify profitable trading opportunities (Mashrur *et al.* 2020). These models leverage historical data, news, and other relevant factors to forecast market movements, demonstrating the potential of machine learning to revolutionize financial analysis.

Gerlein *et al.* (2016) explores the broad range of machine learning applications, highlighting its role in educational technology and online learning. Machine learning algorithms are utilized to create adaptive learning environments, personalized content delivery, and automated assessment systems, thereby enhancing the educational experience. The research underscores the transformative impact of machine learning on traditional education methods, offering insights into its future potential in the academic sector.

Machine learning (ML) is a branch of artificial intelligence that enables computers to learn from and make decisions based on data. In the realm of finance, particularly in stock prediction, ML algorithms analyze historical market data to identify patterns and trends that can inform future predictions. Unlike traditional statistical models, ML models can handle large volumes of data and capture complex, non-linear relationships between variables, which are often present in financial markets. Stock prediction using ML involves various techniques such as regression, classification, and time series analysis. Advanced models like neural networks, decision trees, and ensemble methods (e.g., XGBoost) are frequently employed to enhance prediction accuracy. These models learn from vast amounts of historical stock prices, trading volumes, and other relevant financial indicators to predict future stock movements.

The integration of ML in stock prediction offers significant advantages. It improves the precision of predictions, adapts to changing market conditions, and uncovers hidden patterns that are not easily detectable by human analysts. Thus, ML has become an indispensable tool in modern finance, driving innovations in stock prediction and beyond.

Gradient Boosting

Gradient boosting is a powerful machine learning technique widely used for regression and classification tasks due to its high accuracy and flexibility. As a form of ensemble learning, gradient boosting combines the predictions of multiple weak models, typically decision trees, to produce a strong predictive model. Natekin and Knoll (2013) provide a comprehensive tutorial on gradient boosting machines, explaining the underlying principles and how they iteratively optimize a loss function to improve model performance.

One of the significant advantages of gradient boosting is its ability to handle complex datasets and uncover intricate patterns that might be missed by simpler models. This makes it particularly effective in scenarios where accurate predictions are crucial. For example, Zhang and Haghani (2015) applied a gradient boosting method to enhance travel time prediction, demonstrating its superior performance compared to traditional models. By leveraging the technique's capability to capture non-linear relationships and interactions between variables, they were able to achieve more precise travel time estimates, which are vital for transportation planning and management.

Gradient boosting's versatility extends to various domains, including finance, healthcare, and marketing, where it is used for tasks such as credit scoring, disease diagnosis, and customer segmentation. Its ability to handle different types of data and provide interpretable results makes it a preferred choice among data scientists and analysts. Additionally, gradient boosting machines can be fine-tuned through hyperparameter optimization, allowing for further improvements in model accuracy and generalization to new data.

The process begins by fitting an initial model to the data. Then, a new model is trained to predict the residual errors of the initial model. These residuals represent the difference between the predicted and actual values. By iteratively adding models that focus on these residuals, the overall prediction accuracy is gradually improved. The "gradient" aspect refers to the optimization process, where each new model is trained to minimize a loss function, often using gradient descent techniques.

Gradient Boosting is highly effective due to its ability to handle various types of data and its robustness against overfitting when properly tuned. It can capture complex patterns and interactions within the data, making it particularly suitable for tasks like stock prediction, where intricate and non-linear relationships are common. Popular implementations of Gradient Boosting include XGBoost, LightGBM, and CatBoost, each offering enhancements in terms of speed and performance.

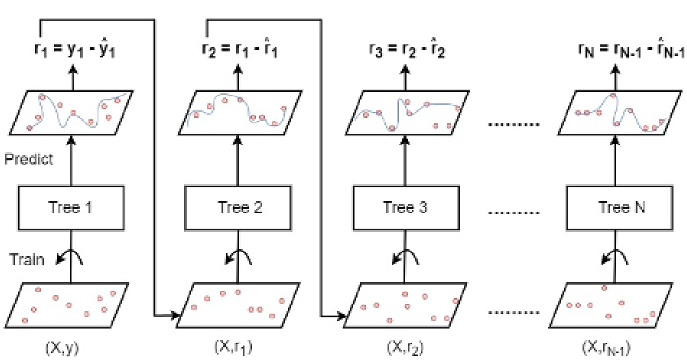

Figure 1 Gradient Boosting Algorithm

Figure 1, explains how Gradient Boosting algorithm works. By leveraging Gradient Boosting in stock prediction, we can achieve more accurate and reliable forecasts, enhancing the decision-making process in financial investments.

The initial model formula in Gradient Boosting is given by:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=0}^{n} L(Y_i, \gamma) \tag{1}$$

In this formula, we aim to find the initial prediction model $F_0(x)$ that minimizes the loss function L. Here's a breakdown of the components:

- $\arg\min_{\gamma}$: This notation means we are looking for the value of γ that minimizes the expression inside the sum. Essentially, it represents the argument (value) of γ that results in the smallest possible value of the loss function.
- $\sum_{i=0}^{n}$: This is a summation symbol indicating that we are summing over all n training examples. The index i runs from 1 to n, where n is the total number of training samples.
- $L(Y_i, \gamma)$: This is the loss function that measures the difference between the true value Y_i and the prediction γ. The specific form of the loss function γ depends on the problem at hand (e.g., mean squared error for regression, logistic loss for classification).
- Y_i: These are the true values or target values from the training data.
- γ: This is the initial constant prediction value we are trying to find that minimizes the loss function over all training examples.

In summary, this formula initializes the Gradient Boosting model by finding a constant prediction that best fits the training data according to the chosen loss function L. This initial step is crucial as it provides the starting point for subsequent iterations, where the model is incrementally improved by adding trees that correct the residual errors of previous models.

XGBoost

The XGBoost algorithm has gained prominence as a highly efficient and scalable machine learning technique, particularly suited for tasks involving large datasets and complex models. XGBoost stands for "Extreme Gradient Boosting" and is an implementation of gradient boosted decision trees designed for speed and performance. Li *et al.* (2019) demonstrated the effectiveness of XGBoost in the field of genomics by using it to predict gene expression values. Their study highlighted the algorithm's ability to handle high-dimensional data and its superior performance compared to traditional methods (Vats and Samdani 2019).

One of the key strengths of XGBoost is its ability to leverage hardware advancements, such as GPU computing, to accelerate training processes. Mitchell and Frank (2017) explored the acceleration of the XGBoost algorithm using GPUs, which significantly reduced training times while maintaining model accuracy. This enhancement makes XGBoost particularly appealing for real-time applications and scenarios where computational resources are limited (Zhou 2021). By utilizing GPUs, data scientists can efficiently train complex models and iterate faster, leading to quicker insights and decision-making.

XGBoost's robust performance and adaptability have made it a popular choice across various domains, including finance, healthcare, and marketing. Its ability to handle missing values, support parallel and distributed computing, and provide built-in regularization to prevent overfitting sets it apart from other machine learning algorithms. Additionally, XGBoost offers a range of hyperparameters that can be fine-tuned to optimize model performance, allowing practitioners to tailor the algorithm to their specific needs.

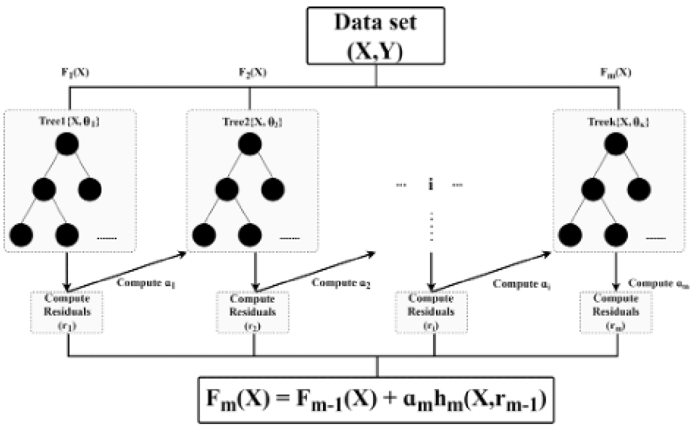

Figure 2 XGBoost Algorithm

One of the key features of XGBoost is its regularization technique, which helps prevent overfitting. Regularization adds a penalty term to the loss function, ensuring that the model remains generalizable and does not overly fit the training data. Additionally, XGBoost uses a sophisticated tree pruning algorithm to optimize the tree structure, further improving model performance.

The objective function of XGBoost is defined as:

$$L(\theta) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) \tag{2}$$

where:

- $L(\theta)$ is the overall objective function.
- $L\left(y_i, \hat{y}_i^{(t)}\right)$ is the loss function, measuring the difference between the true value y_i and the predicted value $\hat{y}_i^{(t)}$ at iteration t.
- $\Omega(f_k)$ is the regularization term for the k-th tree f_k.
- θ represents the model parameters.

The regularization term $\Omega(f_k)$ can be further defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{3}$$

where:

- γT penalizes the complexity of the tree with T leaves.
- $\lambda \sum_{j=1}^{T} w_j^2$ penalizes the leaf weights w_j, helping to reduce overfitting.

By combining these components, XGBoost aims to build a model that not only fits the training data well but also generalizes effectively to new, unseen data. The inclusion of the regularization term is particularly important in preventing overfitting, making XGBoost a robust and reliable choice for a wide range of machine learning tasks, including stock prediction.

AdaBoost

The AdaBoost algorithm, short for Adaptive Boosting, is a powerful ensemble learning method that has significantly influenced the development of machine learning models. Originally introduced to improve the performance of weak classifiers, AdaBoost combines multiple weak learners to create a strong classifier with enhanced accuracy. Ying *et al.* (2013) provide a comprehensive overview of the advancements and future prospects of the AdaBoost algorithm, highlighting its theoretical foundations and various applications across different fields (Shinde and Shah 2018).

One notable application of the AdaBoost algorithm is in network intrusion detection, where it has proven effective in identifying and mitigating cyber threats. Hu *et al.* (2008) developed an AdaBoost-based algorithm for network intrusion detection, demonstrating its ability to detect a wide range of network anomalies with high accuracy. Their study underscores AdaBoost's strength in handling imbalanced datasets, a common challenge in cybersecurity, and its capacity to improve detection rates while minimizing false positives.

AdaBoost's versatility and adaptability make it suitable for a variety of domains beyond cybersecurity, including finance, healthcare, and image recognition. Its ability to focus on hard-to-classify instances and iteratively refine the model by adjusting weights for misclassified data points allows it to achieve high accuracy and robustness. Moreover, AdaBoost is compatible with various base classifiers, providing flexibility for practitioners to tailor the algorithm to specific tasks.

Despite its advantages, AdaBoost also has some limitations, such as sensitivity to noisy data and the potential for overfitting, particularly when dealing with complex datasets. However, ongoing research and advancements continue to enhance its capabilities and address these challenges, ensuring its relevance and utility in the evolving landscape of machine learning.

The key steps of the AdaBoost algorithm are as follows:

1. Initialize the weights of all training instances equally.
2. For each iteration:
   - Train a weak learner on the weighted training data.
   - Compute the weighted error of the weak learner.
   - Calculate the importance (weight) of the weak learner based on its error.
   - Update the weights of the training instances: increase the weights of incorrectly classified instances and decrease the weights of correctly classified instances.
3. Combine the weak learners into a final strong classifier by weighted voting.

Mathematically, the weight update and final prediction in AdaBoost can be described as follows:

Initialization

$$w_i^{(i)} = \frac{1}{n}, \forall i = 1, \ldots, n \tag{4}$$

where $w_i^{(i)}$ is the initial weight for each instance, and n is the total number of training instances.

Weight Update

$$w_i^{(i1)} = w_i^{(i)} \cdot \exp\left(a_t \cdot 1\{h_t(x_i) \neq y_i\}\right) \tag{5}$$

where:

- a_t is the weight of the weak learner h_t at iteration t, calculated as:

$$a_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{6}$$

with ϵ_t being the weighted error rate of h_t.

- $1\{h_t(x_i) \neq y_i\}$ is an indicator function that equals 1 if $h_t(x_i) \neq y_i$ and 0 otherwise.
- $w_i^{(t)}$ is the weight of instance i at iteration t.

Final Prediction

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} a_t \cdot h_t(x)\right) \tag{7}$$

where $H(x)$ is the final strong classifier, and T is the total number of iterations.

AdaBoost's ability to adaptively focus on challenging instances and combine multiple weak learners into a strong classifier makes it a robust method for various machine learning tasks, including stock prediction. By leveraging AdaBoost, we can enhance the predictive power and accuracy of our models, contributing to more informed financial decision-making.

CatBoost

CatBoost is a Gradient Boosting algorithm designed to deal with categorical variables. CatBoost is basically an ensemble learning model created by combining many decision trees.

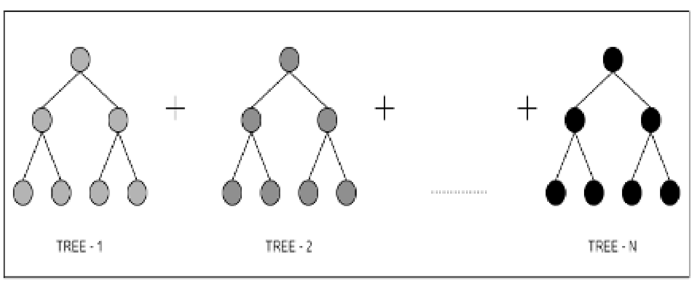

Figure 3 CatBoost Structure

The main components of CatBoost are the Objective Function, Decision Trees and Gradient Boosting Algorithm. In addition, it has a special structure that processes categorical variables. Thanks to this structure, more effective processing is achieved by using the internal order of categorical variables.

LightGBM

LightGBM is an implementation of the Gradient Boosting framework, a machine learning framework. Therefore, the mathematical formula of LightGBM is generally similar to the formula of Gradien Boosting algorithms. LightGBM stands out with features such as histogram-based learning and scaled gradient descent. Basically the mathematical equation is:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \tag{7}$$

In this equation, $F_m(x)$ is the sum of the prediction when m trees are added. $F_{m-1}(x)$, $m-1$ is the estimate with trees added. η represents the learning rate and $h_m(x)$ is the contribution of the m^{th} tree. LightGBM specifically uses histogram-based learning. In this way, the learning process accelerates and allows lower memory usage.

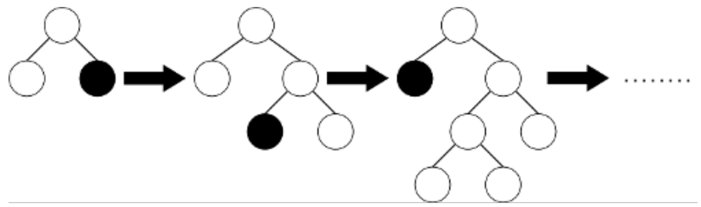

Figure 4 LightBM Structure

Confusion Matrix and Performance Metrics

A complexity matrix is used to interpret the results of an established classification model and to cross-examine the errors in the relationship between real and predicted values.

Table 2 Confusion Matrix

Confusion Matrix		Actual Values	
		Positive (1)	**Negative (0)**
2*Predicted Results	**Positive (1)**	TP [1,1]	FP [1,0]
	Negative (0)	FN [0,1]	TN [0,0]

- **True Positive**: Correctly predicting the positive situation.
- **True Negative**: Correctly predicting the negative situation.
- **False Positive**: Incorrectly predicting the positive situation.
- **False Negative**: Predicting the negative situation incorrectly.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

RESULTS

In our study, we evaluated the performance of three machine learning algorithms—Gradient Boosting, XGBoost, and AdaBoost—for predicting personal loan eligibility. Our results showed that XGBoost outperformed the other models, achieving an accuracy of 0.95, precision of 0.95, recall of 0.95, and an F1 score of 0.95. Gradient Boosting also performed well, with an accuracy of 0.91, precision of 0.92, recall of 0.86, and an F1 score of 0.89. AdaBoost, while still effective, had slightly lower performance metrics, with an accuracy of 0.89, precision of 0.89, recall of 0.85, and an F1 score of 0.87, as illustrated in Figure5 and Table 3 . These results indicate that XGBoost is the most effective model for predicting personal loan eligibility in our dataset, providing high accuracy and reliability in identifying creditworthy individuals while minimizing false positives and false negatives.

Table 3 Success Metrics

	SUCCESS METRICS (%)			
Model	**Accuracy**	**Precision**	**Recall**	**F1 - Score**
Adaboost	89	89	85	87
Gradient Boosting	91	92	86	89
XGBoost	95	95	95	95
LightGBM	90	93	92	89

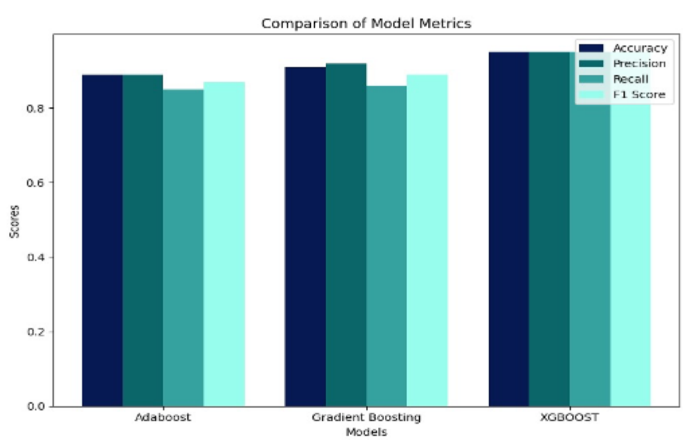

Figure 5 Gradient Boost, XGBoost, AdaBoost Comparison of Model Matrix

CONCLUSION

The superior performance of XGBoost can be attributed to its advanced implementation of the Gradient Boosting algorithm, which includes enhancements such as regularization, parallel processing, and efficient handling of missing values. By using XGBoost, financial institutions can enhance their decision-making process regarding personal loan approvals, confidently approving loans for individuals who are likely to repay and reducing the risk of defaults. Additionally, the high recall value ensures that the model captures a significant proportion of potential loan candidates, providing a comprehensive assessment of creditworthiness.

Beyond improving loan approval processes, this study's predictive model can have broader implications. Financial institutions can leverage the model's insights to tailor financial products to better meet customer needs, improving overall customer satisfaction and retention. The model's ability to accurately predict creditworthiness can also help banks optimize their lending strategies, allocate resources more efficiently, and reduce operational costs associated with loan defaults. Furthermore, by integrating such advanced machine learning techniques into their operations, banks can stay competitive in a rapidly evolving financial landscape, ultimately enhancing their overall operational efficiency and customer service. In conclusion, our study demonstrates that XGBoost is a powerful tool for predicting personal loan eligibility, offering significant benefits for both financial institutions and customers. By leveraging these advanced machine learning models, banks can streamline their loan approval process, improve customer satisfaction, and optimize their lending strategies, leading to better financial outcomes for all stakeholders involved.

Availability of data and material

Not applicable.

Conflicts of interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The author has no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Gerlein, E., T. McGinnity, A. Belatreche, and S. Coleman, 2016 Evaluating machine learning classification for financial trading: An empirical approach. Expert Systems with Applications **54**: 193–207.

Henrique, B. M., V. A. Sobreiro, and H. Kimura, 2019 Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications **124**: 226–251.

Hu, W., W. Hu, and S. Maybank, 2008 Adaboost-based algorithm for network intrusion detection. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **38**: 577–583.

Kamalov, F., I. Gurrib, and K. Rajab, 2021 Financial forecasting with machine learning: price vs return. Journal of Computer Science **17**: 251–264.

Li, W., Y. Yin, X. Quan, and H. Zhang, 2019 Gene expression value prediction based on xgboost algorithm. Frontiers in genetics **10**: 1077.

Lin, W.-Y., Y.-H. Hu, and C.-F. Tsai, 2011 Machine learning in financial crisis prediction: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**: 421–436.

Mashrur, A., W. Luo, N. A. Zaidi, and A. Robles-Kelly, 2020 Machine learning for financial risk management: a survey. IEEE Access **8**: 203203–203223.

Mitchell, R. and E. Frank, 2017 Accelerating the xgboost algorithm using gpu computing. PeerJ Computer Science **3**: e127.

Mulvey, J. M., 2017 Machine learning and financial planning. IEEE Potentials **36**: 8–13.

Natekin, A. and A. Knoll, 2013 Gradient boosting machines, a tutorial. Frontiers in neurorobotics **7**: 21.

Shinde, P. P. and S. Shah, 2018 A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pp. 1–6, IEEE.

Vats, P. and K. Samdani, 2019 Study on machine learning techniques in financial markets. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–5, IEEE.

Ying, C., M. Qi-Guang, L. Jia-Chen, and G. Lin, 2013 Advance and prospects of adaboost algorithm. Acta Automatica Sinica **39**: 745–758.

Zhang, Y. and A. Haghani, 2015 A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies **58**: 308–324.

Zhou, Z.-H., 2021 *Machine learning*. Springer Nature.

How to cite this article: Özkurt, C. Enhancing Financial Decision-Making: Predictive Modeling for Personal Loan Eligibility with Gradient Boosting, XGBoost, and AdaBoost. *Information Technologies in Economics and Business*, 1(1), 7-13, 2024.