International Journal of Scientific Engineering and Science
Volume 9, Issue 5, pp. 109-114, 2025. ISSN (Online): 2456-7361

The Application of Several Machine Learning Models to Loan Approval Prediction

Le Thi Thu Giang[1]
[1]Faculty of Mathematical Economics, Thuongmai University, Ha noi, Vietnam

Abstract— *This paper investigates the application of several machine learning models, including Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting Machines, for loan approval prediction. Experimental results on a dataset obtained from https://www.kaggle.com show that the proposed machine learning models are suitable.*

Keywords— *Loan approval, Machine learning, Naive Bayes, Decision Tree, Random Forest, Gradient Boosting Machines.*

I. INTRODUCTION

The banking and financial industry's operational efficiency and are deeply intertwined with the effective management of loan portfolios. A core challenge in this sector is the accurate prediction of loan approval. Traditional loan approval methods are often characterized by inefficiencies, high operational costs, and an increased risk of human error. These shortcomings highlight the pressing need for automated and data-driven solutions to streamline and enhance the loan approval process.

The traditional loan approval process used to be manually assessed based on certain parameters such as the applicant's credit history, income level, employment status, and similar financial metrics. According to Livingstone and Lunt in [1], credit history reveals a person's previous loan and debt repayments, and payment history is a crucial indicator of a borrower's creditworthiness. The income level determines whether a person has the financial strength to afford the loan payments, while the employment status reflects the person's ability to earn a regular income [2]. In addition, financial metrics such as the debt-to-income ratio are among other important parameters considered during the loan approval process. These parameters help the lender assess the applicant's eligibility for the loan and set the loan terms. However, since this process is manual, it is time-consuming and carries a high risk of error.

Machine learning (ML) has emerged as a transformative technology with the potential to revolutionize various aspects of the financial sector. The selection of machine learning models is an important step for making the right decisions in loan approval systems. ML algorithms can analyze large datasets of historical loan information, identifying intricate patterns and correlations between applicant attributes and loan repayment behavior. By leveraging these learned patterns, ML models can predict the likelihood of loan default, enabling financial institutions to make more informed and objective lending decisions. A number of studies have examined various machine learning techniques and their efficiency in this domain.

The early foundation of credit scoring models lies in statistical techniques such as Logistic Regression (LR) which was discussed by Thomas et al. [3]. While interpretable, these models struggle to capture nonlinear relationships and complex interactions between variables. Brown and Mues [4] compared LR with modern ML techniques and found traditional models lacking in handling imbalanced data common in loan datasets.

Due to their simplicity and effectiveness, Random Forest (RF) and Decision Trees (DT) are widely used. Khandani et al. [5] demonstrated that RF models outperformed traditional credit-scoring techniques, especially in predicting default risk. Similarly, Malekipirbazari and Aksakalli [6] applied Random Forest to peer-to-peer lending data and achieved high accuracy, demonstrating the model's robustness and adaptability to different lending platforms.

Gradient Boosting Machines (GBM), including XGBoost and LightGBM, have emerged as top-performing models in structured datasets. Zhou et al. [7] employed XGBoost to predict loan defaults and highlighted its superior performance in comparison to traditional models. Similarly, Lessmann et al. [8] conducted a benchmarking study across 15 ML algorithms and found that boosting-based methods consistently ranked among the most accurate.

Neural Networks (NNs), especially deep learning models, are gaining traction. Baesens et al. [9] discussed the potential of neural networks for financial analytics but noted concerns over interpretability. Chen et al. [10] proposed a hybrid deep learning model combining rule-based filtering with neural networks to improve performance while addressing fairness concerns. Suto and Takeuchi [11] used Long Short-Term Memory (LSTM) networks to analyze time-series credit data, showing that sequential information can enhance creditworthiness assessment.

This paper focuses on the application of four distinct and widely used machine learning algorithms to the problem of loan approval prediction: Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting Machines. These algorithms represent a range of methodological approaches, offering different strengths and weaknesses in terms of interpretability, computational efficiency, and predictive accuracy. By comparing their performance and analyzing their characteristics, this study aims to provide valuable insights into their suitability for enhancing loan approval processes within modern financial institutions.

II. METHODOLOGY

2.1 Naive Bayes

International Journal of Scientific Engineering and Science
Volume 9, Issue 5, pp. 109-114, 2025. ISSN (Online): 2456-7361

The Naive Bayes model is a probabilistic algorithm based on Bayes' Theorem with following fomular

$$\mathrm{P(Target\ label/Features)} = \frac{\mathrm{P(Features)P(Features/Target\ label)}}{\mathrm{P(Features)}}$$

to simplify the calculation of this probability, all features are assumed to be independent in their classes. Therefore, instead of calculating the joint probability of all features, we only need to multiply the probabilities of individual features given the class:

$$\mathrm{P(Features/Class)} = \mathrm{P(Feature\ 1/Class)P(Feature\ 2/Class)\ldots P(Feature\ n/Class).}$$

In practice, this assumption is often violated because input variables often have hidden relationships with each other. That is the reason why it is called the "naive" assumption. However, this algorithm still proves to be quite effective in classification problems, such as the loan approval problem due to its computationally and speed efficient, especially with high-dimensional data (many features) such as text data.

In the prediction problem for loan approval, we use Naive Bayes to calculate the probability of approving or rejecting a loan based on the applicant's information. Here, the "Target Label" represents the loan status, including "Approved" and "Rejected," and "Features" is a feature vector describing the parameters and characteristics of the applicant requesting loan approval (may include: gender, education level, income, etc.). If $\mathrm{P(Approved/}Features) > \mathrm{P(}Rejected/\mathrm{Features})$ the loan is suggested to be accepted.

In this problem, $Y \in \{Approved, Rejected\}$ is the target label (loan status) and $X = (x_1, x_2, \ldots, x_n)$ is the feature vector describing the applicant (e.g., gender, education, income, etc.)

If $P(Approved/X) > P(Rejected/X)$ then the loan is approved.

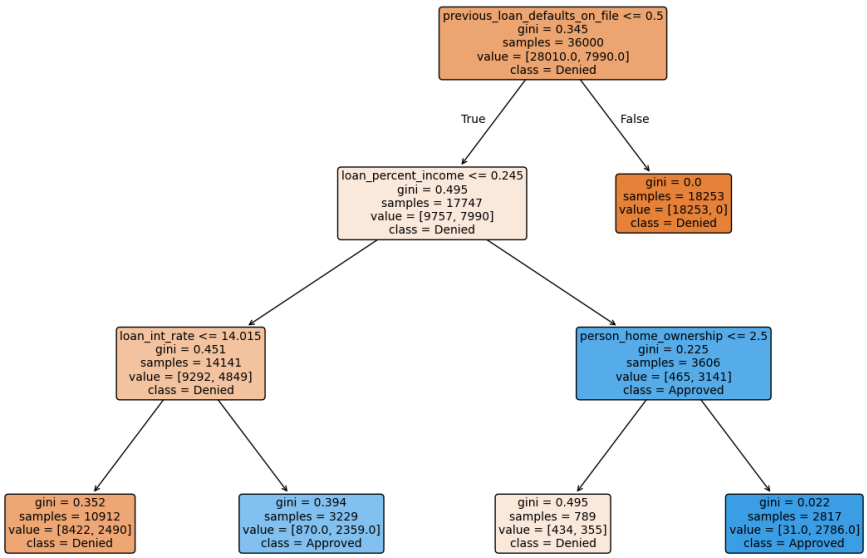
2.2 Decision Tree

The decision tree algorithm is widely used in classification problems. Due to its intuitive structure, the algorithm has the advantage of being easy to understand and. Furthermore, the algorithm is less sensitive to noisy data and can use for both qualitative and quantitative data types.

A decision tree includes the following main components:

- Root Node: This is the starting point of the tree, representing the entire dataset. The root node will be split into branches (child nodes) based on a certain attribute (feature).
- Internal Node: These nodes represent a question or a condition based on the value of an attribute. From each decision node, there will be branches leading to subsequent child nodes, corresponding to different cases (values) of that attribute.
- Branch: Each branch connects two nodes and represents an outcome or a choice based on the condition at the splitting node.
- Leaf Node/Terminal Node: These are the tree's last nodes, and they don't divide anymore. A final prediction result is represented by each leaf node.

In a decision tree, the root and branch nodes are chosen based on how well a feature separates the target classes. The root is chosen to be the feature with the greatest information gain or least Gini impurity. The tree then grows by selecting the best features at each level to form branch nodes. When no further useful splits can be made, the process stops at leaf nodes, where final predictions (approved/rejected, for this problem) are assigned.



Source: compiled by the authors

2.3 Random Forest

Random Forest is a popular and effective machine learning algorithm. It is an ensemble learning method. Random Forest builds a "forest" of multiple random, individual decision trees and then combines the results by aggregating their votes to make the final prediction. This method was developed to overcome the limitations of Decision, especialy the overfitting problem.

The working principle of SVM includes the following steps
Step 1: From the original training dataset, create multiple random data subsets by sampling randomly with replacement from the original dataset.
Step 2: For each data subset, build a decision tree. Unlike the single decision tree algorithm, this algorithm only considers a random subset of features selected from the initial features to find the best split point.
Step 3: Train each decision tree independently on the corresponding data subset and feature subset.
Step 4: The ultimate outcome of Random Forest is the class that the majority of the trees predicted after all of the forest's trees have been trained (majority voting).
Building trees on random data and feature subsets helps reduce the correlation between trees. By making the entire model less susceptible to noise in the training data and better able to generalize to fresh data, this lowers overfitting and improves the model's stability. Additionally, Random Forest's capacity to choose features at random helps it function well even when dealing with datasets with a high number of features.

2.4 Gradient Boosting Machines

Gradient Boosting Machines (GBM) are another ensemble learning method that builds trees in a stage-wise fashion. The main idea of GBM is to build a model by sequentially combining multiple "weak" models (often small decision trees). Each subsequent weak model is built to correct the errors or residuals that the previous model did not handle well. Popular algorithms based on the Gradient Boosting principle include XGBoost, LightGBM, and CatBoost, which have demonstrated superior performance in many different problems.

The operating principle of GBM can be explained through the following main steps:
Step 1: Model Initialization: Start with a simple initial model which is usually a model to predict the average value of the target variable (for regression problems) or the probability distribution of classes (for classification problems).
Step 2: Calculate Residuals (Errors): After having the initial model, GBM calculates the residuals between the actual values of the target variable and the predicted values of the current model. These residuals are the "errors" that subsequent models need to learn to minimize.
Step 3: Train a new weak model on Residuals: A new weak is trained. However, instead of directly predicting the target variable, this weak model is trained to predict the residuals calculated in the previous step. The goal is for this weak model to "learn" the relationship between the input features and the errors of the current model.
Step 4: Update the Overall Model: The predictions of the new weak model (after being trained on residuals) are not added directly to the overall model in their entirety. Instead, they are multiplied by a small learning rate before being added to the overall model. This learning rate helps control the learning speed and prevents the model from learning too quickly, which can lead to overfitting.
Step 5: Repeat the Process: Steps 2, 3, and 4 are repeated sequentially for a predetermined number of iterations. A new weak model is added during each iteration to account for the remaining residuals after the contribution of the prior weak models.
Final Model: The final GBM model is the weighted sum of the initial model and all the weak models trained during the iterative process.

The advantages of the Gradient Boosting Machine algorithm is its ability to achieve high prediction accuracy and effectively handle complex and non-linear relationships in the data. However, its significant disadvantages are it may be overfitting if not carefully tuned; the training process is resource-intensive and time-consuming, it requires tuning many complex hyperparameters, and the final model is often difficult to interpret.

III. Data Preparation

This dataset contains 45,000 records of loan applicants which is taken from https://www.kaggle.com. The data includes 14 columns with 13 columes represent different factors influencing loan approvals:
person_age: Age of the applicant (in years),
person_gender: Gender of the applicant (Male, Female),
person_education: Educational background (High School, Bachelor, Master, Doctorate, Associate),
person_income: Annual income of the applicant (in USD),
person_emp_exp: Years of employment experience,
person_home_ownership: Type of home ownership (Rent, Own, Mortgage),
loan_amnt: Loan amount requested (in USD),
loan_intent: Purpose of the loan (Personal, Education, Medical, ventural, homeimprovement, deptconsolation),
loan_int_rate: Interest rate on the loan (percentage),
loan_percent_income: Ratio of loan amount to income,
cb_person_cred_hist_length: Length of the applicant's credit history (in years),
credit_score: Credit score of the applicant,
previous_loan_defaults_on_file: Whether the applicant has previous loan defaults (Yes or No),
and one column to describe the target variable:
loan_status: 1 if the loan was repaid successfully, 0 if the applicant defaulted.
Categorical qualitative data such as loan_intent, person_education, person_gender, person_home_ownership, previous_loan_defaults_on_file were converted into numerical form using the LabelEncoder algorithm available from the sklearn.preprocessing module in the scikit-learn library.
The data was processed and tested using Python software.

IV. Experimental Results

After being processed, the dataset was splidt into two parts: a training set comprising 80% of the original data and the

remaining 20% for the test set. The results from the Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting Machine (in this paper, we use XGBoost algorithms) models indicate that all four methods have quite high accuracy. These results are presented in confusion matrices, accuracy scores, and their ROC curves.

When running the Naive Bayes, Decision Tree, XGBoost, and Random Forest algorithms, the confusion matrices are displayed in the diagrams above. It can be seen that the Gradient Boosting Machine and Random Forest algorithms presents the very good classification results with the highest number of correct classifications, whereas the classification results of the Naive Bayes algorithm show the lowest accuracy. However, despite its ability to distinguish between classes is poorer, however, this classification performance is acceptable.

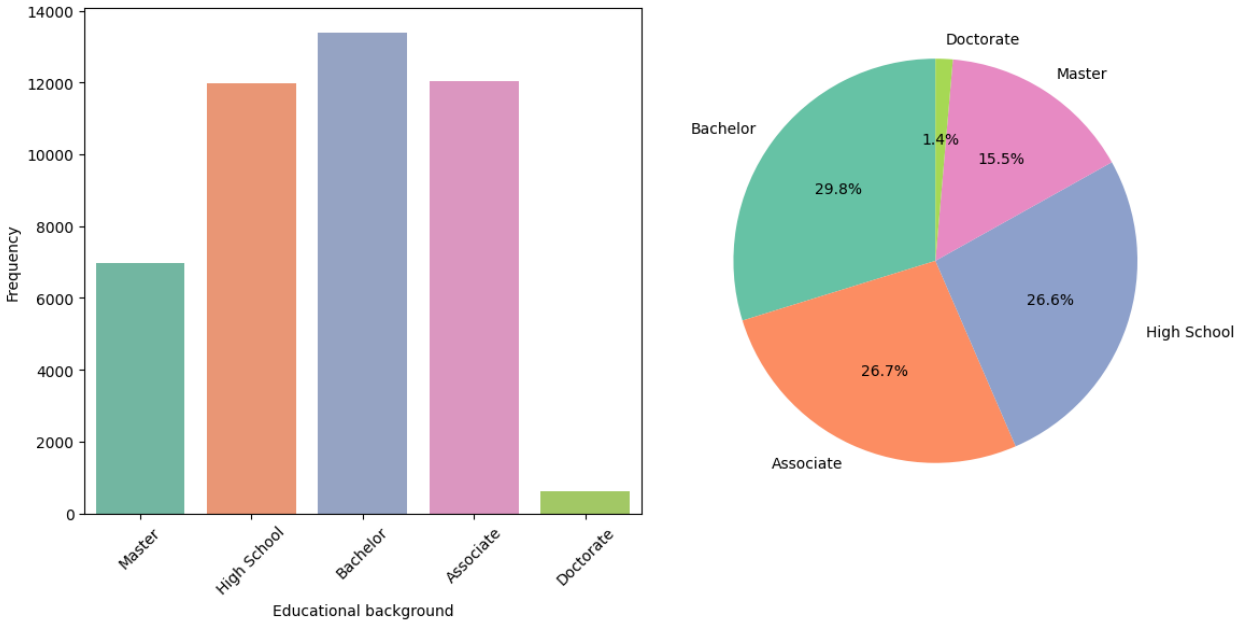


Fig. 1. Educational background distribution

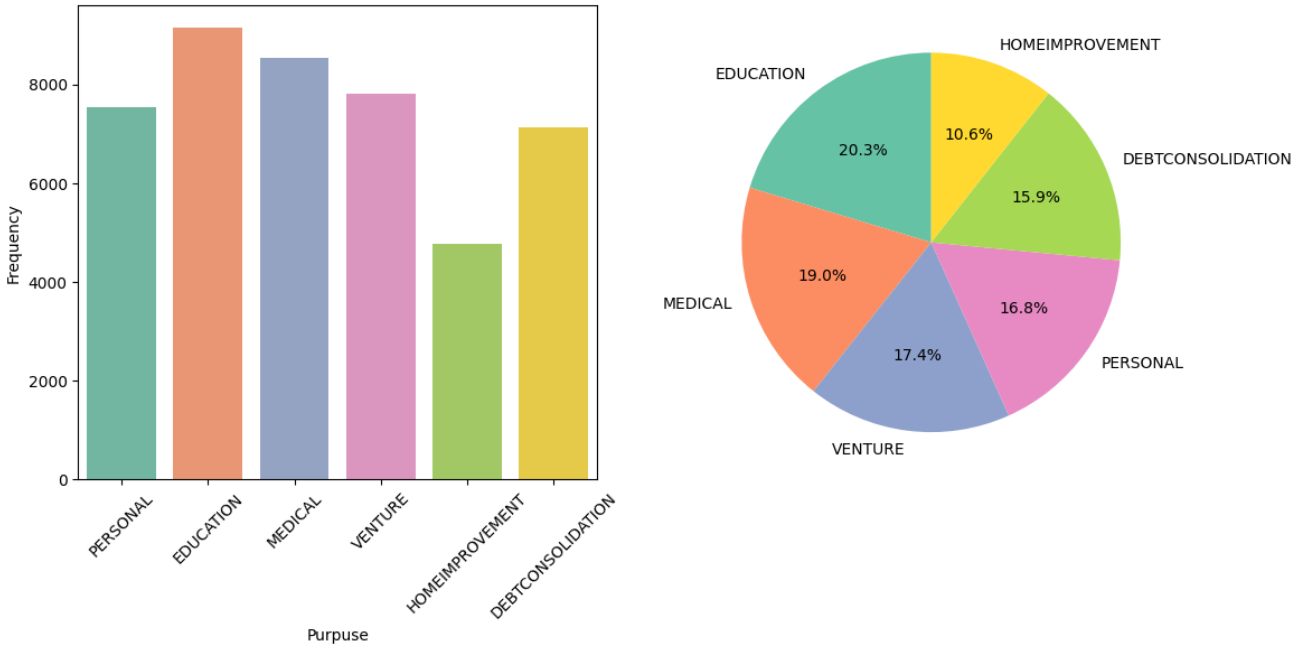


Fig. 2. Loan Purpose Distribution

International Journal of Scientific Engineering and Science
Volume 9, Issue 5, pp. 109-114, 2025. ISSN (Online): 2456-7361

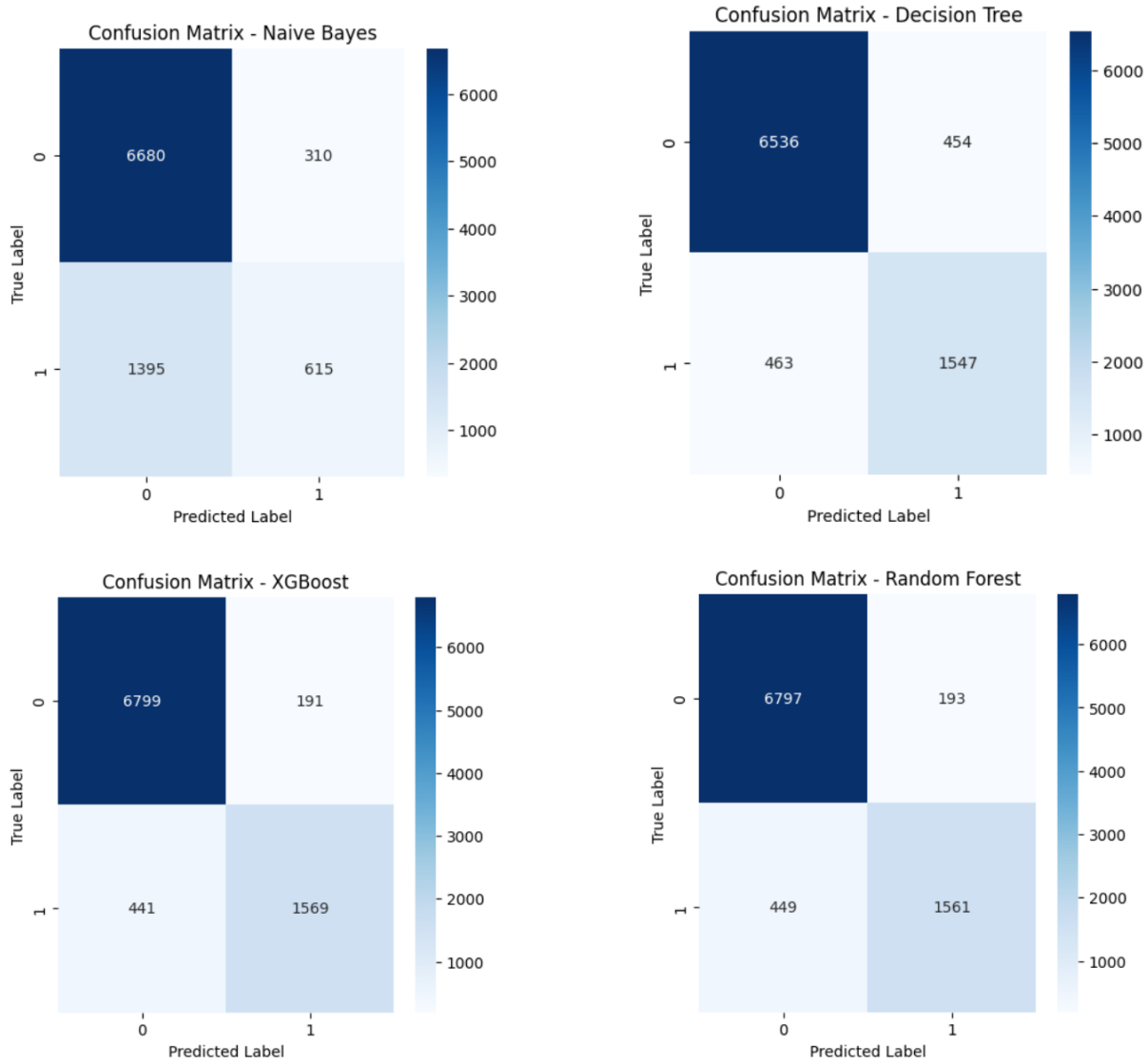


Fig. 3. Confusion matrices

TABLE 1: The accuracy of algorithms

	Algorithms	Accuracy
1	Naive Bayes	0.810556
2	Decision Tree	0.898111
3	Random Forest	0.928667
4	XGBoost	0.929778

Comparison results from Table1 show that all methods yield quite high accuracy, which are all above 80%. The two algorithms Random Forest and Gradient Boosting Machine achieved the highest correct classification rates, with over 92%. Those of Naïve Bayes and Decision Tree are also high with 81% and approximate 90%, respectively. This demonstrates the potential of applying the proposed machine learning models for the loan approval prediction problem.

From the comparison results of the ROC curves between the methods, the superior predictive performance of the two methods, Gradient Boosting Machine and Random Forest, is also evident. Both of these algorithms show very good performance with AUC values of 0.98 and 0.97, respectively. Their ROC curves lie almost directly against the top-left corner and are nearly identical over most of the FPR range. This indicates that both XGBoost and Random Forest have a very high ability to classify and distinguish between classes. The Decision Tree algorithm has an AUC of 0.85. Although not as good as the two aforementioned algorithms, the performance of the Decision Tree is still considered quite good, as shown by its ROC curve lying significantly above the random diagonal line. The Naive Bayes algorithm has the lowest performance among the four models with an AUC of 0.78. The ROC curve of Naive Bayes lies closer to the diagonal line compared to the other algorithms, indicating its poorer ability to distinguish between classes; however, this classification performance is still acceptable.

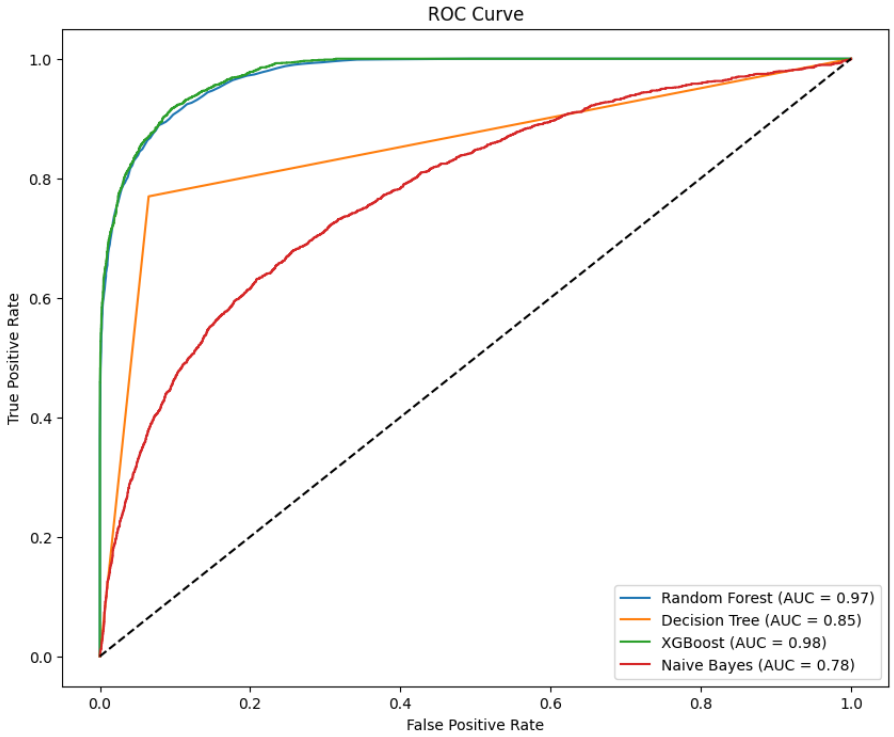


Fig. 4. ROC curves

REFERENCES

[1] S. M. Livingstone and P. K. Lunt, "Predicting personal debt and debt repayment: Psychological, social and economic determinants," *J. Econ. Psychol.*, vol. 13, no. 1, pp. 111-134, 1992.

[2] N. W. Hillman, "College on credit: A multilevel analysis of student loan default," Rev. High. Educ., vol. 37, no. 2, pp. 169-195, 2014.

[3] L. C. Thomas, D. B. Edelman, and J. N. Crook, *Credit Scoring and its Applications*. SIAM, 2002.

[4] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3446-3453, 2012.

[5] A. E. Khandani, A. J. Kim, and A. W. Lo, "Consumer credit-risk models via machine-learning algorithms," *Journal of Banking & Finance*, vol. 34, no. 11, pp. 2767-2787, 2010.

[6] M. Malekipirbazari and V. Aksakalli, "Risk assessment in social lending via random forests," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4621-4631, 2015.

[7] L. Zhou, H. Luo, and X. Zhang, "Loan default prediction on peer-to-peer lending data using artificial intelligence," *Financial Innovation*, vol. 5, no. 1, pp. 1-20, 2019.

[8] S. Lessmann, B. Baesens, H. V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124-136, 2015.

[9] B. Baesens, V. Van Vlasselaer, and W. Verbeke, *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*. Wiley, 2014.

[10] M. Chen, Y. Zhang, and D. Wu, "A hybrid ensemble deep learning model for credit risk evaluation," *Knowledge-Based Systems*, vol. 217, 106800, 2021.

[11] Y. Suto and K. Takeuchi, "Loan default prediction using LSTM model and its explanation," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020, pp. 735-740.