 VIETNAM NATIONAL UNIVERSITY HO CHI MINH

**UNIVERSITY OF ECONOMICS AND LAW**

**PROJECT FINAL REPORT**

**DATA SCIENCE FOR FINANCE**

**NĂM 2023**

**CREDIT SCORING BY MACHINE LEARNING**

**Course: K20414C\_ Fintech**

**Course code: 222CN10**

**Student: Tran Nguyen Bao Truc**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Name** | **Student ID** | **Class** | **Phone number** | **Email** |
|  | Tran Nguyen Bao Truc | K204141935 | K20414C | 0902384544 | [tructnb20414c@st.uel.edu.vn](mailto:tructnb20414c@st.uel.edu.vn) |

VIETNAM NATIONAL UNIVERSITY HO CHI MINH

**UNIVERSITY OF ECONOMICS AND LAW**

**PROJECT FINAL REPORT**

**DATA SCIENCE FOR FINANCE**

**NĂM 2023**

**CREDIT SCORING BY MACHINE LEARNING**

Table of Contents

[**1: TOPIC OVERVIEW** 4](#_Toc136286600)

[**2: THEORETICAL BASIS AND RESEARCH MODEL** 5](#_Toc136286601)

[2.1. Theoretical basis 5](#_Toc136286602)

[2.1.1. The concept of credit risk 5](#_Toc136286603)

[2.1.2. Previous studies 5](#_Toc136286604)

[2.2. Research Methods 7](#_Toc136286605)

[2.2.1. Rating Indicators 7](#_Toc136286606)

[2.2.2. Logistic regression model 10](#_Toc136286607)

[2.2.3. Decision tree 12](#_Toc136286608)

[2.2.4. Random Forest model 15](#_Toc136286609)

[2.2.5. XG Boost model 17](#_Toc136286610)

[2.2.6. SVM Model 20](#_Toc136286611)

[2.2.7. K-fold Cross-validation 22](#_Toc136286612)

[**3: DATA AND MODEL RESULTS** 24](#_Toc136286613)

[3.1. Data 24](#_Toc136286614)

[3.2. Description of variables used in the study 24](#_Toc136286615)

[3.3 Data processing 29](#_Toc136286616)

[3.3.1. Import data 30](#_Toc136286617)

[3.3.2. Data preprocessing 31](#_Toc136286618)

[3.3.3.Data descriptive: 32](#_Toc136286619)

[3.3.4.Data visualization 35](#_Toc136286620)

[3.4.Model Assesment 37](#_Toc136286621)

[1. Random Forest 39](#_Toc136286622)

[2. Logistic Regression 42](#_Toc136286623)

[3. Decision Tree 47](#_Toc136286624)

[4. XGBOOST 51](#_Toc136286625)

[5. Support Vector Machine (SVM) 54](#_Toc136286626)

[6. ROC curve comparison 56](#_Toc136286627)

[7. 10-fold Cross -Validation 58](#_Toc136286628)

[8. Predicting new data 61](#_Toc136286629)

[**4. CONCLUSION** 66](#_Toc136286630)

[REFERENCES 66](#_Toc136286631)

# **1: TOPIC OVERVIEW**

**The coronavirus pandemic is a humanitarian crisis** that continues to affect lives and livelihoods around the world. It has forced regional and national economies to close for weeks and months at a time, causing hardship—sometimes of existential gravity—for many populations. Countermeasures taken to contain the virus and save lives stopped the economy from functioning. With lockdowns now being lifted and businesses restarting, lending institutions are faced with a new and unfamiliar environment, in which they must evaluate and monitor credit risk with limited visibility and access to reliable data. The COVID-19 pandemic has had a significant impact on the global economy, and many borrowers have experienced financial hardship as a result. This has led to an increase in credit risk for lenders, as more borrowers may be struggling to make payments on time or may be at risk of defaulting on their loans. To manage this risk, many lenders have tightened their lending standards and become more selective about the borrowers they approve for loans. They may require higher credit scores, more income documentation, and additional collateral or guarantees to mitigate their risk.

Additionally, technology has played an increasingly important role in managing credit risk. Many lenders use sophisticated algorithms and machine learning models to analyze vast amounts of data and identify potential credit risks. This can help lenders make more informed lending decisions and manage their risk more effectively. Data and analytics capabilities are proving essential to the solution

However, it's important to note that credit risk is not solely the responsibility of lenders. Borrowers also play a critical role in managing their own credit risk. By maintaining a good credit history, paying bills on time, and managing debt responsibly, borrowers can improve their creditworthiness and reduce their risk of default.

# **2: THEORETICAL BASIS AND RESEARCH MODEL**

## 2.1. Theoretical basis

### 2.1.1. The concept of credit risk

Credit risk, also known as default risk or repayment risk, refers to the likelihood that a borrower will fail to repay a loan or credit obligation as agreed. It is a key consideration for lenders when evaluating a borrower's creditworthiness and determining whether to extend credit and at what terms. Credit risk is based on a number of factors, including the borrower's credit history, income, employment status, debt-to-income ratio, and other financial and personal factors. Lenders use this information to assess the borrower's ability to repay the loan and the likelihood of default. High credit risk can result in higher interest rates, lower credit limits, or even loan denial. This is because lenders want to manage their risk and ensure that they are lending to borrowers who are likely to repay the loan as agreed.

### 2.1.2. Previous studies

Scientific paper “Integration of unsupervised and supervised machine learning algorithms for credit risk assessment” *by* Bao, W., Lianju, N., & Yue, K. (2019). This study propose a combination strategy of integrating unsupervised learning with supervised learning for credit risk assessment. To asset the credit risk, the comparisons of model performance are performed based on three credit datasets in four groups: individual models, individual models + consensus model, clustering + individual models, clustering + individual models + consensus model.

“Machine learning in banking risk management” by Leo, M., Sharma, S., & Maddulety, K. (2019). This paper seeks to analyse and evaluate machine-learning techniques that have been researched in the context of banking risk management, and to identify areas or problems in risk management that have been inadequately explored and are potential areas for further research. The review has shown that the application of machine learning in the management of banking risks such as credit risk, market risk, operational risk and liquidity risk has been explored; however, it doesn’t appear commensurate with the current industry level of focus on both risk management and machine learning. A large number of areas remain in bank risk management that could significantly benefit from the study of how machine learning can be applied to address specific problems.  
Empirical studies have also been conducted in Vietnam to evaluate the repayment capacity of individual customers, such as:

Nguyen, T. D. D., Ha1-Nguyen, T. T., & Ngoc, B. (2021) research on prediction of Consumer Credit risk in Vietnamese Commercial Banks. The study seeks to test factors that can influence the credit risk of individual consumer borrowers of commercial banks in Vietnam through the use of discriminant analysis. Age, number of dependents, years in current job and salary are independent variables relating to demographic and socioeconomic condition of borrowers while loan amount is independent variable relating to characteristic of the loan. The results show that the estimation function is significant at the 1% level and can predict the financial position of the borrower (customer) with an average accuracy of 74.5%. Therefore, in this study, the demographic, socioeconomic and loan related variables can be used to classify individual borrowers of Vietnamese commercial banks into payment group and non-repayment group.

Van Sang, H., Nam, N. H., & Nhan, N. D. (2016) has conduct a study to evaluate credit scoring prediction model based on Feature Selection approach and parallel random forest. This paper proposed a credit scoring model based on parallel Random Forest classifier and feature selection method to evaluate the credit risks of applicants. By integration of Random Forest into feature selection process, the importance of features can be accurately evaluated to remove irrelevant and redundant features. An algorithm to select best features was developed by using the best average and median scores and the lowest standard deviation as the rules of feature scoring. Consequently, the dimension of features can be reduced to the smallest possible number that allows of a remarkable runtime reduction. The obtained results showed that an improved accuracy of the proposed model compared to other commonly used feature selection methods. In particular, our method can attain the average accuracy of 76.2% with a significantly reduced running time of 72 minutes on German credit dataset and the highest average accuracy of 89.4% with the running time of only 50 minutes on Australian credit dataset

Pham, Q. H. (2021) has researched on the credit rating system of firms in Vietnam. This research analyses the effect of the new accounting system on credit rating models of small, medium and large firms in Vietnam. To illustrate, the new accounting system significantly changes the measure of financial numbers in the financial reports of firms, including assets, liabilities, owner’s equity, revenues and expenses, so it also affects the credit rating model. This study investigates the new accounting system (Circular 200) implemented on 1 January 2015 in Vietnam which guides both local and foreign enterprises in accounting policies for financial years beginning 1 January 2015 (Phan, Joshi and Tran-Nam 2018). The results of this study show that the new local accounting system significantly affects the credit rating models in several ways: (1) changing the inputs of the models; (2) changing the model performance; and (3) changing the exact values of the weights and biases of the models.

To sum up, although the studies vary in terms of their focus, participants, and methodology, they generally suggest that income, loan purpose, loan amount, age, and collateral are the key variables in evaluating the likelihood of default or non-default of individual customers. These findings provide a scientific foundation for constructing research models for individual customer cases in the author's dataset.

## 2.2. Research Methods

## 2.2.1. Rating Indicators

**2.2.1.1. Accuacy**

Accuracy is a commonly used metric to evaluate the performance of predictive models. In the context of prediction, accuracy refers to the ability of a model to correctly predict the outcome of a given input.

For example, in a binary classification problem, where the goal is to predict whether an input belongs to one of two possible classes, the accuracy of a model is the percentage of times it correctly predicts the class of a given input.

The formula for accuracy is:

Accuracy =

In binary classification problems, where there are only two possible outcomes, we can define accuracy as:

Accuracy

where:

* True Positives (TP) are the number of correct predictions where the actual class is positive and the predicted class is also positive.
* True Negatives (TN) are the number of correct predictions where the actual class is negative and the predicted class is also negative.
* False Positives (FP) are the number of incorrect predictions where the actual class is negative but the predicted class is positive.
* False Negatives (FN) are the number of incorrect predictions where the actual class is positive but the predicted class is negative.

**2.2.1.2. ROC curve**

ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classifier as its discrimination threshold is varied. It is a plot of the true positive rate (TPR) against the false positive rate (FPR) at different threshold settings.

The TPR is the fraction of true positive predictions out of the total actual positive cases. It is also known as sensitivity or recall. The FPR is the fraction of false positive predictions out of the total actual negative cases. It is given by (1-specificity), where specificity is the fraction of true negative predictions out of the total actual negative cases

A perfect classifier would have a TPR of 1 and an FPR of 0, and would be represented by a point at the upper left corner of the ROC curve. A random classifier would have a diagonal ROC curve from (0,0) to (1,1), representing the line where TPR=FPR.

Important points in ROC Curve:  
**TPR = 0, FPR = 1** : The model predicts all cases to be negative type  
**TPR = 1, FPR = 1** : The model predicts all cases to be positive type  
**TPR = 1, FPR = 0:** Ideal model with 0 misclassification

**2.2.1.3. AUC**

The area under the ROC curve (AUC-ROC) is a commonly used metric to evaluate the overall performance of a binary classification model in prediction tasks. A higher AUC-ROC indicates a better overall performance of the model in distinguishing between the positive and negative classes across all possible thresholds.

AUC-ROC is a number between 0 and 1, where a value of 1 indicates a perfect classifier, while a value of 0.5 indicates a random classifier. The closer the AUC-ROC value is to 1, the better the overall performance of the model in distinguishing between the positive and negative classes.

**2.2.1.4. Precision**

Precision is a metric used to evaluate the performance of a binary classification model in predicting the positive class. In other words, precision measures the proportion of positive predictions made by the model that are actually correct. A higher precision value indicates that the model is making fewer false positive predictions and is more accurate in identifying the positive cases.

Precision can be calculated using the following formula:

Precision =

where True Positives are the number of positive cases that are correctly predicted by the model, and False Positives are the number of negative cases that are incorrectly predicted as positive by the model.

**2.2.1.5. Recall**

Recall is a metric used to evaluate the performance of a binary classification model in identifying the positive cases. In other words, recall measures the proportion of actual positive cases that are correctly identified by the model. A higher recall value indicates that the model is making fewer false negative predictions and is more accurate in identifying the positive cases.

Recall can be calculated using the following formula:

Recall =

where True Positives are the number of positive cases that are correctly predicted by the model, and False Negatives are the number of positive cases that are incorrectly predicted as negative by the model.

### 2.2.2. Logistic regression model

**2.2.2.1. Logistic regression model definition**

Logistic regression is a statistical model that is used to predict the probability of a binary outcome variable based on one or more independent variables. The model uses a logistic function to model the relationship between the independent variables and the probability of the binary outcome variable. Logistic regression is a type of generalized linear model (GLM), which is a flexible class of models that can handle a wide range of response distributions.

In logistic regression, the binary outcome variable is typically encoded as 0 or 1, where 0 represents the negative or non-event outcome, and 1 represents the positive or event outcome. The independent variables can be continuous or categorical, and their coefficients are estimated using maximum likelihood estimation. The logistic function is used to map the linear combination of the independent variables to the probability of the positive outcome, which is bounded between 0 and 1. The logistic function has an S-shaped curve that asymptotically approaches 0 as the linear combination approaches negative infinity, and approaches 1 as the linear combination approaches positive infinity

**2.2.2.2. Strengths and Weaknesses of Logistic Regression**

Logistic regression has several strengths as a classification model, including:

1. Interpretable: The coefficients of the logistic regression model can be interpreted as the effect of the independent variables on the log-odds of the positive outcome. This makes logistic regression a highly interpretable model that can provide insights into the relationship between the independent variables and the outcome.
2. Flexible: Logistic regression can handle both continuous and categorical independent variables, and can also be extended to handle multiple independent variables.
3. Efficient: Logistic regression is a computationally efficient model that can be trained on large datasets with relatively low computational resources.
4. Robust: Logistic regression is a robust model that can handle outliers and missing data.
5. Provides probabilistic predictions: Logistic regression produces probabilities of the positive outcome, which can be useful for making decisions based on the level of confidence in the predictions.

However, logistic regression also has some weaknesses:

1. Assumption of linearity: Logistic regression assumes a linear relationship between the independent variables and the log-odds of the positive outcome. This assumption may not hold in all cases, and nonlinear relationships may need to be modeled using other techniques.
2. Limited to binary classification: Logistic regression is designed for binary classification problems and may not be suitable for multi-class classification problems without modification.
3. Susceptible to overfitting: Logistic regression can be susceptible to overfitting if there are too many independent variables relative to the number of observations in the dataset.
4. Not always optimal: Logistic regression may not always provide the best performance compared to other more complex models such as decision trees or neural networks, especially in cases where the relationship between the independent variables and the outcome is highly nonlinear.

**2.2.2 .3. Application of logistic regression model**

Logistic regression is widely used in prediction problems where the outcome of interest is binary (i.e., it takes on only two possible values). Some examples of applications of logistic regression in prediction include:

1. Credit risk assessment: Logistic regression can be used to predict the probability of a borrower defaulting on a loan based on their credit history, income, and other financial indicators.
2. Fraud detection: Logistic regression can be used to predict the probability of a transaction being fraudulent based on various features of the transaction, such as the amount, location, and time.
3. Customer churn prediction: Logistic regression can be used to predict the probability of a customer leaving a company based on their usage patterns, demographics, and other factors.
4. Disease diagnosis: Logistic regression can be used to predict the probability of a patient having a particular disease based on their symptoms, medical history, and other factors.
5. Image recognition: Logistic regression can be used to predict the probability of an image containing a particular object or feature based on various features of the image, such as color, texture, and shape.

In all these applications, logistic regression is used to predict the probability of the positive outcome based on one or more independent variables. The model is trained on a dataset of examples where the outcome is known, and then applied to new examples where the outcome is unknown to make predictions.

### 2.2.3. Decision tree

**2.2.3.1. Definition of decision tree**

A decision tree is a graphical representation of a decision-making process that uses a tree-like structure to model various decision paths and their outcomes. It is a popular tool for decision analysis and predictive modeling in machine learning and data science.

A decision tree consists of nodes and branches. The nodes represent decision points or events in the decision-making process, while the branches represent possible outcomes or consequences of those decisions. The root node of the tree represents the initial decision, while the terminal nodes (also known as leaf nodes) represent the final outcomes or decisions.

Decision trees can be used for both classification and regression problems, depending on the type of outcome variable. In classification problems, the outcome variable is categorical (e.g., yes/no, high/medium/low), while in regression problems, the outcome variable is continuous (e.g., a numeric value).

Decision trees are easy to interpret and visualize, and can be used to identify the most important variables or features in a dataset.

**.2.3.2. Strengths and Weaknesses of Decision Trees**

Decision trees have several strengths and weaknesses that should be considered when using them for data analysis and modeling.

Strengths:

1. Easy to interpret: Decision trees are easy to understand and interpret, even for non-experts, because they represent the decision-making process in a visual and intuitive way.
2. Versatile: Decision trees can be used for both classification and regression problems, and can handle both categorical and continuous variables.
3. Feature selection: Decision trees can be used to identify the most important features or variables in a dataset, which can be useful for feature selection and dimensionality reduction.
4. Robust to noise: Decision trees are relatively robust to noise and outliers in the data, because they are based on binary splits that are not affected by small variations in the data.
5. Non-parametric: Decision trees are a non-parametric method, which means they do not make any assumptions about the distribution of the data or the relationships between variables.

Weaknesses:

1. Overfitting: Decision trees can easily overfit the training data, especially if the tree is too deep or complex. This can lead to poor generalization and performance on new data.
2. Instability: Small variations in the data can lead to different decision trees, which can make the model unstable and difficult to interpret.
3. Bias: Decision trees can be biased towards features with more levels or categories, because they can create more splits and potentially overfit to those features.
4. Greedy approach: The decision tree algorithm uses a greedy approach to make splits, which means it may not find the optimal or globally optimal tree.
5. Limited expressiveness: Decision trees have limited expressiveness compared to more complex models like neural networks or support vector machines, which can handle more complex relationships between variables.

**2.2.3.3. Application of the Decision Tree mode**

Decision trees have a wide range of applications in data science and machine learning. Some common applications of decision trees include:

1. Customer churn prediction: Decision trees can be used to predict whether a customer is likely to churn or leave a company based on their demographic and behavioral characteristics. This information can be used to develop targeted retention strategies and improve customer loyalty.
2. Credit risk assessment: Decision trees can be used to predict the likelihood of default or non-payment on a loan based on a borrower's credit history, income, and other factors. This information can be used to make more informed lending decisions and reduce credit risk.
3. Medical diagnosis: Decision trees can be used to predict the most likely diagnosis for a patient based on their symptoms, medical history, and test results. This information can be used to assist in clinical decision-making and improve patient outcomes.
4. Fraud detection: Decision trees can be used to detect fraudulent transactions or activities based on patterns in the data, such as unusual purchase amounts or frequencies. This information can be used to prevent financial losses and improve security.
5. Product recommendation: Decision trees can be used to recommend products or services to customers based on their past purchase history and preferences. This information can be used to improve customer satisfaction and increase sales.

Overall, decision trees are a powerful tool for prediction tasks, as they can handle both categorical and continuous variables and are easy to interpret. However, it is important to carefully validate and tune the decision tree model to avoid overfitting and ensure accurate predictions on new data.

### 2.2.4. Random Forest model

**2.2.4.1. Definition of the Random Forest Model**

Random Forest is a popular machine learning algorithm that belongs to the family of ensemble learning methods. It is a type of decision tree-based model that combines multiple decision trees to make more accurate predictions. In a random forest model, a large number of decision trees are built independently and then combined to make a final prediction.

The basic idea behind the random forest algorithm is to create a diverse set of decision trees that can capture different aspects of the data. Each decision tree is built using a random subset of the features and a random sample of the data, which helps to reduce overfitting and increase the model's generalization ability. The final prediction is made by aggregating the predictions of all the individual trees in the forest.

**2.2.4.2. Strengths and Weaknesses of Random Forest**

Random Forests have several strengths and weaknesses that make them suitable for different types of machine learning tasks. Here are some of the key strengths and weaknesses of Random Forests:

Strengths:

1. High Accuracy: Random Forests are known for their high accuracy and robustness, which makes them suitable for a wide range of classification and regression tasks.
2. Robustness: Random Forests are less prone to overfitting than individual decision trees, and can handle missing data and noisy features well.
3. Non-Parametric: Random Forests are non-parametric models, which means they do not make assumptions about the underlying distribution of the data.
4. Scalability: Random Forests can handle large datasets with high dimensionality, making them suitable for big data applications.
5. Feature Importance: Random Forests provide measures of feature importance, which can help in feature selection and interpretation of the model.

Weaknesses:

1. Computationally Expensive: Random Forests can be computationally expensive, especially for large datasets with many features. This can make training and testing the model time-consuming.
2. Black-Box Model: Random Forests can be difficult to interpret, especially when dealing with a large number of trees and features.
3. Bias-Variance Tradeoff: Random Forests can suffer from bias-variance tradeoff, where increasing the number of trees may lead to higher variance and lower bias, but may also increase the computational complexity of the model.
4. Data Imbalance: Random Forests may not perform well on imbalanced datasets, where one class is much more prevalent than the others.

Overall, Random Forests are a powerful and versatile machine learning algorithm that can be used for a wide range of classification and regression tasks. However, they have some limitations that should be taken into account when designing and evaluating the model.

**2.2.4.3. Random Forest application**

Random Forests can be used for a wide range of prediction tasks, including classification and regression. Here are some common applications of Random Forests in prediction:

1. Fraud Detection: Random Forests can be used to detect fraudulent transactions in banking and finance, by learning patterns from historical data and identifying anomalies in new transactions.
2. Healthcare: Random Forests can be used for diagnosis and prediction of diseases, by analyzing medical data and identifying risk factors for different conditions.
3. Marketing: Random Forests can be used for customer segmentation and prediction of customer behavior, by analyzing customer data and identifying patterns in their buying habits.
4. Image Recognition: Random Forests can be used for image recognition tasks, by analyzing image features and classifying images into different categories.
5. Natural Language Processing: Random Forests can be used for natural language processing tasks, such as sentiment analysis, by analyzing textual data and identifying patterns in language use.

Random Forests are particularly useful when dealing with large datasets with high dimensionality, as they can handle a large number of features and can be trained in parallel. Additionally, Random Forests provide measures of feature importance, which can help in feature selection and interpretation of the model.

### 2.2.5. XG Boost model

**2.2.5.1. Definition of the XG Boost model**

XGBoost (eXtreme Gradient Boosting) is a powerful machine learning algorithm used for regression, classification, and ranking tasks. It is an ensemble learning method that combines the predictions of multiple decision trees to produce a final prediction.

The algorithm works by building a series of decision trees sequentially, where each tree tries to correct the errors of the previous one. Each tree is built by recursively partitioning the data based on the features that best separate the data into different classes or reduce the mean squared error of the target variable in a regression task.

XGBoost uses gradient boosting to optimize the model by minimizing a cost function, such as mean squared error, using gradient descent. The gradient descent algorithm adjusts the weights of each feature to minimize the cost function, improving the accuracy of the model.

**2.2.5.2. Strengths and Weaknesses of XG Boost**

XGBoost (eXtreme Gradient Boosting) is a popular and powerful machine learning algorithm that has many strengths, but it also has some limitations. Here are some of the main strengths and weaknesses of XGBoost:

Strengths:

1. High accuracy: XGBoost is known for its high predictive accuracy, as it can handle complex non-linear relationships between features and target variables.
2. Speed: XGBoost is highly optimized for performance, and it can handle large datasets and high-dimensional feature spaces quickly.
3. Flexibility: XGBoost can handle a wide variety of data types, including categorical, numerical, and text data, and can be used for both regression and classification tasks.
4. Robustness: XGBoost can handle missing data and outliers well, and it includes built-in regularization techniques to prevent overfitting.
5. Interpretability: XGBoost provides feature importance scores, which can help users understand which features are the most important for making predictions.

Weaknesses:

1. Parameter tuning: XGBoost has many hyperparameters that need to be tuned carefully to achieve optimal performance, which can be time-consuming and requires a good understanding of the algorithm.
2. Memory usage: XGBoost can require a large amount of memory, especially for large datasets and high-dimensional feature spaces.
3. Black-box nature: XGBoost is an ensemble learning algorithm, which means that it can be difficult to interpret the individual decision trees and understand how the algorithm is making predictions.
4. Imbalanced data: XGBoost may have difficulty with imbalanced datasets, where one class is much more prevalent than the others. Additional techniques such as oversampling or undersampling may be required to address this issue.
5. Overfitting: Although XGBoost includes built-in regularization techniques, it is still possible to overfit the model if the hyperparameters are not tuned correctly.

**2.2.5.3. XG Boost application**

XGBoost (eXtreme Gradient Boosting) is a popular algorithm for prediction tasks, as it can handle complex non-linear relationships between features and target variables. Here are some examples of how XGBoost can be used for prediction:

1. Sales forecasting: XGBoost can be used to predict future sales for a business based on historical sales data, seasonal trends, and other factors such as promotions or changes in market conditions.
2. Healthcare outcomes: XGBoost can be used to predict healthcare outcomes such as hospital readmission rates, patient mortality, and disease diagnosis. This can help healthcare providers improve patient care and reduce costs.
3. Credit risk assessment: XGBoost can be used to predict credit risk, which can help banks and financial institutions assess the likelihood of a borrower defaulting on a loan. This can help lenders make more informed decisions about loan approvals and interest rates.
4. Fraud detection: XGBoost can be used to predict fraudulent transactions in real-time, helping to prevent financial losses for businesses and customers.
5. Marketing campaigns: XGBoost can be used to predict the success of marketing campaigns, such as email or social media campaigns, based on customer data, demographics, and other factors. This can help businesses optimize their marketing efforts and increase ROI.

Overall, XGBoost is a powerful tool for prediction tasks, and its ability to handle large datasets, complex relationships between features, and non-linear relationships make it well-suited for many types of prediction problems.

### 2.2.6. SVM Model

**2.2.6.1. Definition of the SVM Model**SVM (Support Vector Machine) is a supervised machine learning algorithm that can be used for classification or regression tasks. It is a powerful algorithm that is often used for complex problems where the data is non-linearly separable. The SVM algorithm works by finding a hyperplane that separates the data into different classes.

In the case of a binary classification problem, the hyperplane that is found by the SVM algorithm maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. These closest data points are known as support vectors. The SVM algorithm aims to find the hyperplane that not only separates the data but also has the largest margin, which leads to better generalization performance.

SVM can be used for both linear and non-linear classification tasks, by using a kernel function that transforms the data into a higher-dimensional space where it can be separated by a hyperplane. Some common kernel functions used in SVM include linear, polynomial, radial basis function (RBF), and sigmoid.

SVM can also be used for regression tasks, where the goal is to predict a continuous target variable. In this case, SVM finds a hyperplane that minimizes the errors between the predicted values and the actual values.

**2.2.6.2. Strengths and Weaknesses of SVM**

SVM (Support Vector Machine) is a powerful machine learning algorithm that has several strengths and weaknesses. Here are some of the key strengths and weaknesses of SVM:

Strengths:

1. Effective in high-dimensional spaces: SVM is effective in high-dimensional spaces where the number of features is much larger than the number of samples. It can handle a large number of features and variables.
2. Robust to outliers: SVM is robust to outliers in the data, which means that it can still perform well even when there are data points that are far from the other data points.
3. Good generalization performance: SVM has good generalization performance, which means that it can accurately predict the target variable on new, unseen data.
4. Can handle non-linear relationships: SVM can handle non-linear relationships between the features and the target variable, by using different kernel functions to transform the data into a higher-dimensional space.

Weaknesses:

1. Computationally expensive: SVM can be computationally expensive, especially for large datasets. The training time increases rapidly as the number of samples increases.
2. Sensitivity to kernel choice: The performance of SVM is highly dependent on the choice of kernel function. Choosing the wrong kernel function can lead to poor performance.
3. Difficult to interpret: The output of SVM is not easily interpretable, which can make it difficult to understand how the algorithm is making predictions.
4. Can be prone to overfitting: SVM can be prone to overfitting the training data if the model is too complex or the kernel function is too specific to the training data.

**2.2.6.3.SVM application**

SVM (Support Vector Machine) can be applied to prediction tasks in various fields, including finance, healthcare, and engineering. Here are some examples of how SVM can be used for prediction:

1. Financial forecasting: SVM can be used to predict future trends in financial markets, such as stock prices and exchange rates. This can help traders and investors make informed decisions about buying and selling assets.
2. Healthcare outcomes: SVM can be used to predict healthcare outcomes such as patient mortality and disease diagnosis. This can help healthcare providers improve patient care and treatment.
3. Customer churn prediction: SVM can be used to predict which customers are likely to churn or cancel their subscriptions. This can help businesses develop strategies to retain customers and reduce customer churn.
4. Quality control: SVM can be used to predict product quality and detect defects in manufacturing processes. This can help manufacturers identify and correct issues before they become more serious.
5. Traffic flow prediction: SVM can be used to predict traffic flow and congestion on roads and highways. This can help transportation planners and engineers develop more efficient transportation systems.

Overall, SVM is a powerful tool for prediction tasks, and its ability to handle non-linear relationships between features and the target variable makes it well-suited for many types of prediction problems.

### 2.2.7. K-fold Cross-validation

**2.2.7.1. Definition of K-fold Cross-validation**

K-fold cross-validation is a technique used in machine learning to evaluate the performance of a model. It involves splitting the original dataset into K equal subsets, or "folds", where one fold is used as the validation set and the other K-1 folds are used for training the model. This process is repeated K times, with each fold used once as the validation set, and the remaining K-1 folds used for training the model.

At the end of the K iterations, the performance of the model is evaluated by calculating the average performance across all K folds. This technique helps to reduce the bias and variance of the model, and provides a more accurate estimate of the model's performance on new, unseen data.

**2.2.6.2. Strengths and Weaknesses of K-fold Cross-validation**

K-fold cross-validation is a widely used technique in machine learning for evaluating the performance of a model. It has several strengths and weaknesses, which are discussed below:

Strengths:

1. Reduces bias and variance: K-fold cross-validation helps to reduce the bias and variance of the model by training and testing the model on different subsets of the data.
2. More accurate estimate of performance: By averaging the performance of the model over K iterations, K-fold cross-validation provides a more accurate estimate of the model's performance on new, unseen data.
3. More efficient use of data: K-fold cross-validation allows us to use more of the data for both training and testing, which can lead to better model performance.
4. Helps to identify overfitting: K-fold cross-validation can help to identify whether a model is overfitting or underfitting the data, by comparing the training and validation performance.

Weaknesses:

1. Computationally expensive: K-fold cross-validation can be computationally expensive, especially for large datasets or complex models, as it requires training the model K times.
2. Can be time-consuming: K-fold cross-validation can be time-consuming to set up and execute, especially if the model requires significant preprocessing or feature engineering.
3. May not work well for imbalanced data: K-fold cross-validation may not work well for imbalanced datasets, where the number of samples in each class is significantly different.
4. May not generalize well to new data: K-fold cross-validation may not generalize well to new, unseen data, especially if the distribution of the data is significantly different from the training data.

# **3: DATA AND MODEL RESULTS**

## 3.1. Data

Using data from previous studies, a simulation dataset consisting of 1099 customers with credit relationships was created. The aim of this study is to predict the credit risk of individual customers by analyzing their ability to repay loans, which is influenced by a variety of factors such as gender, income, electricity bill, collateral, loan amount and period, marital status, number of years of service, loan purpose, and age of the customer.

## 3.2. Description of variables used in the study

Table 1. Description of variables used in the study

|  |  |
| --- | --- |
| **Variable name** | **The scale** |
| Target variable” | |
| Ability to repay debt | * 0: Debt payment not on time * 1: Paying debt on time |
| Character variable: | |
| 1: Gender | * 0: Female * 1: Male |
| 2: Electricity bill |  |
| 3: Loan amount |  |
| 4: Purpose of loan | * 1: Education * 2: Consumer spending * 3: Home purchase * 4: Car purchase * 5: Securities investment |
| 5: Family | * 1: Single * 2: Married * 3: Divorced |
| 6: Work time |  |
| 7: Age |  |
| 8: Loan term |  |
| 9: Monthly income |  |
| 10: Payroll statement | * 0: Have a payroll statement * 1: Do not have a payroll statement |
| 11: Collateral | * 0: Secured assets * 1: Unsecured assets |

Target variable: Ability to repay debt

* Economic significance: In the credit risk assessment process, banks or financial institutions need to determine whether the customers have the ability to repay their debt or not. If the customers do not have the ability to repay their debt, it is a credit risk for the financial institution, as it may lead to non-payment or require a large amount of money to recover the debt. 1 is for late payment, 0 is for on-time payment. Therefore, the ability to repay debt is an important factor in credit risk prediction models, and helps banks or financial institutions manage credit risk and protect their assets.
* 0: Paying debt on time
* 1: Debt payment not on time

Attribute 1: Gender

* Economic significance: The gender variable helps banks or financial institutions evaluate the separate aspects of different customer groups as a factor to decide on credit granting by providing information on different spending behaviors between gender groups.
* 0: Female
* 1: Male

Attribute 2: Electricity bill

* Economic significance: When evaluating the ability to pay, credit companies will assess the income and expenses of the customers, and they will consider many different factors, including electricity bills. When an individual or organization has a lot of fluctuations in electricity bills, it may indicate that they are experiencing financial difficulties or may have difficulty paying.

Attribute 3: Loan amount

* Economic significance: Analyzing the loan amount has important economic significance because it allows the evaluation of the borrower's ability to repay based on the loan amount. For example, a larger loan may create a greater financial burden for the borrower and thus increase credit risk. Conversely, a smaller loan may be easily repaid, thus reducing credit risk. In addition, the loan amount variable also affects the interest rate of the loan. Usually, larger loans may have lower interest rates than smaller loans. This can affect the borrower's ability to repay and thus increase or decrease credit risk.

Attribute 4: Purpose of loan

* Economic significance: Changing the purpose of borrowing allows credit institutions to evaluate the borrower's repayment ability based on the intended use of the loan and optimize credit risk management. The purpose of borrowing reflects the level of the borrower's intended use of the loan, meaning whether the loan will be used for a specific purpose or not. If the borrower has a specific use for the loan, the repayment rate will be higher than for unclear or unreliable purposes. For example, a loan for investing in a business project has a higher potential for profit than a loan for personal expenses. Therefore, a loan with a specific purpose of investing in a business will have lower credit risk than a loan for personal expenses.
* 1: Education
* 2: Consumer spending
* 3: Home purchase
* 4: Car purchase
* 5: Securities investment

Attribute 5: Family

* Economic significance: Family has an important economic significance in credit risk forecasting because it allows credit institutions to assess the financial stability of customers based on their family situation.
* 1: Single
* 2: Married
* 3: Divorced

Attribute 6: Work time

* Economic significance: In credit risk forecasting, analyzing the variable of work time at the job has an important economic significance because it allows evaluating the borrower's financial stability. Borrowers with longer work time at a job often have more stable incomes than those who are newly employed.

Attribute 7: Age

* Economic significance: The age of the borrower may affect their financial ability. Specifically, as customers age, their income potential may decrease, and they may have difficulty in repaying their debts. Therefore, age can be used as an independent variable in credit risk prediction models to help assess the borrower's ability to repay debts.

Attribute 8: Loan term

* Economic significance: Loan term is often seen as an important indicator of a customer's ability to manage and repay debt. The longer the borrowing time, the customer can allocate the debt repayment expenses over a longer period of time, so their ability to repay will be higher. However, the longer the borrowing time, the higher the interest cost will be, so customers also need to be able to pay interest over a long period of time.

Attribute 9: Monthly income

* Economic significance: Monthly income is usually considered an important factor in determining a customer's ability to repay, as it relates to the customer's ability to repay the loan. Typically, customers with higher monthly incomes will have a better ability to repay and be less risky when borrowing money, so monthly income becomes a factor to evaluate a customer's repayment ability in credit risk prediction models. In addition, monthly income can provide credit institutions with important information about customers' spending abilities. If customers have higher incomes, their spending and repayment capabilities can be predicted.

Attribute 10: Payroll statement

* Economic significance: A payroll statement is a document that shows the amount of money a customer receives in a specific period of time, usually a month or a quarter. It provides important information about a customer's income and expenses, and also provides credit institutions with information about the stability of a customer's income. If a customer's income is stable and sufficient, meaning enough to repay debts and meet daily expenses, the ability to repay debt will be higher. A payroll statement also provides credit institutions with information about the source of a customer's income, such as from formal jobs or from other income sources. This can help credit institutions assess the stability of a customer's income source and make decisions about lending money.
* 0: Have a payroll statement
* 1: Do not have a payroll statement

Attribute 11: Collateral

* Economic significance: Collateral is an asset provided as security to ensure the repayment of a customer's debt. It increases the reliability of the lender and reduces risk for the credit institution. When considering collateral, it can be real estate, cars, furniture, and other assets with relatively high commercial value. In credit risk prediction models, evaluating the value of collateral will help assess a customer's ability to repay the debt, and also help the lender make decisions about lending money.

## 3.3 Data processing

### 3.3.1. Import data

|  |
| --- |
| **import pandas as pd**  **import numpy as np**  **import seaborn as sns**  **import matplotlib.pyplot as plt**  **data = pd.read\_excel('credit.xlsx')**  **data** |

|  |
| --- |
|  |

|  |
| --- |
| **data.info()** |

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  RangeIndex: 2997 entries, 0 to 2996  Data columns (total 12 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Sex 2997 non-null int64  1 Electricity Bills 2997 non-null int64  2 Loan Amount 2997 non-null int64  3 Loan Purpose 2997 non-null object  4 Marital Status 2997 non-null object  5 Time of Working 2997 non-null int64  6 Age 2997 non-null int64  7 Loan Term 2997 non-null int64  8 Monthly Income 2997 non-null int64  9 Salary Statement 2997 non-null object  10 Collateral 2997 non-null object  11 Repayment Ability 2997 non-null object  dtypes: int64(7), object(5)  memory usage: 281.1+ KB |

The dataset contains 2999 observations and 12 attribute variables. There is no missing value in the dataset and the data types of columns are interger or object ( string or categorical) types.

### 3.3.2. Data preprocessing

**- Check if there are any duplicate value in the data**

|  |
| --- |
| **data.duplicated().sum()** |

|  |
| --- |
|  |

- The dataset has no duplicated values so we can continue to the next step.

**-Drop unnecssary columns in the data set: drop “ID” column**.

|  |
| --- |
| **data.drop(['ID'],axis=1, inplace = True)** |

In this step, the column “ID” is not revelant to the problem so be dropped out.

**-Rename columns’s name**

|  |
| --- |
| **data=data.rename(columns = {'Giới tính':'Sex','Hóa đơn tiền điện':'Electricity Bills',**  **'Số tiền vay':'Loan Amount','Mục đích vay ':'Loan Purpose',**  **'Gia đình':'Marital Status','Thời gian làm việc':'Time of Working',**  **'Tuổi':'Age','Thời hạn vay vốn':'Loan Term',**  **'Thu nhập hàng tháng':'Monthly Income','Sao kê bảng lương':'Salary Statement',**  **'Tài sản thế chấp':'Collateral','Khả năng trả nợ':'Repayment Ability'**  **})**  **data.head(5)** |

|  |
| --- |
|  |

### 3.3.3.Data descriptive:

**-Data descriptive of continous variable:**

|  |
| --- |
| **numerical\_features = ['Electricity Bills', 'Loan Amount','Monthly Income','Age','Time of Working','Loan Term']**  **print('Number of numerical features: ', len(numerical\_features))** |

|  |
| --- |
|  |

|  |
| --- |
| **data[numerical\_features].describe()** |

|  |
| --- |
|  |

- In the “ Electricity Bills” variable, the person who assume the most electricity must pay 30 millions per months, and the lowest paying is 0 miilions. However, looking at the table with more details, most of customers usually pay for their electricity bills in the range of 1.5 millions to 3 millions per months.

- In the “ Loan Amount” variable, the smallest amount is 15 millions and the highest amount is 300 millions and the most common amount is from 30 millions to 80 millions, which can include that the customer may be whoever such as students or adults.

- In the “Monthly Income” variable, the data has a very large range with the lowest value is 2.5 millions and the highest one is 150 miilions. But the majority of customers’ income is under 20 millions.

- In the “ Age” variable, because the loan need to be applied not for the person who is under 20 years old, so the youngest customer is 20 years old and the oldest one is 60 years old. On average, 22 to 50 years old persons take up the largest proportion in the data.

- In the “ Time of Working” variable, majority of customer have been working in their job for 4-12 years which is quite a high period, reflect their stable job. But there are alse someone has just worked for 1 years and someone has a sighnificant time of working with 15 years.

- In the “ Loan Term” variable, the company usually provides loan packages for a period of 2 to 4 years.

**-Data descriptive of categorical variable**

|  |
| --- |
| **categorical\_features = [x for x in data.columns if (x not in numerical\_features and x != 'Repayment Ability')]**  **print('Number of categorical features: ', len(categorical\_features))** |

|  |
| --- |
| Number of categorical features: 5 |

|  |
| --- |
| **for col in categorical\_features:**  **data[col] = data[col].astype('object')**  **data[categorical\_features].describe()** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | **Sex** | **Loan Purpose** | **Marital Status** | **Salary Statement** | **Collateral** | | --- | --- | --- | --- | --- | --- | | **count** | 2997 | 2997 | 2997 | 2997 | 2997 | | **unique** | 2 | 5 | 3 | 2 | 2 | | **top** | 0 | Đầu tư chứng khoán | Ly hôn | Có | Có | | **freq** | 1842 | 638 | 1039 | 1516 | 1510 | |

- In the “Sex” variable, there are two genders : female (0) and male (1) aand the figures for customers who are female tend to be the major of the data.

- In the “ Loan Purpose “ variable, it is clear that the propotion of peple using loan to invest in bonds with the number of 639 observations, approximately 22% of the data. While the other purposes are: Buying house, Buying vehicle, Education and Consumption.

- In the “ Marital Status” variable, the number of pepole who get divorced makes up the largest propotion in the data, nearly one-third of the data. The other marital status are Single and Married.

- In the “ Salary Statement” and “ Colleteral” variable ,over a half of customer have their evidence for their income and secured assets with the number of 1518 and 1511 obsercations respectively.

### 3.3.4.Data visualization

|  |
| --- |
| **plt.figure(figsize=(40,25))**  **def count(categorical\_features):**  **x=1**  **for i in categorical\_features:**  **plt.subplot(3,3,x)**  **ax = sns.countplot(x=i,data=data, palette='magma',hue='Repayment Ability')**  **for rect in ax.patches:**  **ax.text(rect.get\_x() + rect.get\_width() / 2,rect.get\_height()+ 0.75,rect.get\_height(),horizontalalignment='center', fontsize = 10)**  **plt.legend([],[], frameon=False)**  **x+=1**  **plt.legend(['default', 'undefault'])**  **plt.savefig('Data visualization.png')**  **count(['Sex', 'Loan Purpose','Collateral', 'Marital Status',**  **'Salary Statement'])** |

|  |
| --- |
|  |

The bar charts above give information about how each categorical affect the ability to repay debt in details.

- In the first chart represents for “ Sex” variable, female customer tend to have a much more credit worthness in repaying debt. The figures for female customer who repay debt in time takes up the highest observation, while the number for male customers who are not be able to repay debt in time is higher than that of female.

- According to the next graph, customers who borrow for the purpose of securities investment usually don’t repay their loan on time. Meanwhile, data for purpose of consumption have the highest ability to repay on time.

- Looking at the third graph representing the “Collateral”, it is clear that loans granted for customers with collateral are safer because they are more likely to repay on time. Besides, customers without collateral when borrowing are more risky.

- The next graph represents the data for “ Marital Status” variable. Divorced customers tend to have a higher likelihood of paying on time compared to both married and single customers. Those who are married pose the greatest risk among all customer groups.

- The final graph pertains to the “Salary Statement” variable, and the information suggests that customers who have salary statements are at a greater risk of making late payments compared to those without such statements.

## 3.4.Model Assesment

**-Train Test Split**

|  |
| --- |
| **from sklearn.model\_selection import train\_test\_split**  **n\_state = 42**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.1, random\_state = n\_state)**  **X\_train.shape, X\_test.shape** |

- The dataset is divided in to train and test set with a test size is 10%.

-**Scaling data**

|  |
| --- |
| **from sklearn.preprocessing import StandardScaler**  **scale = StandardScaler()**  **X\_train = scale.fit\_transform(X\_train)**  **X\_test = scale.fit\_transform(X\_test)** |

**- Feature Selection:**

|  |
| --- |
| **from sklearn.ensemble import RandomForestClassifier**  **import pandas as pd**  **X = data.drop('Repayment Ability', axis=1)**  **y = data['Repayment Ability']**  **rf = RandomForestClassifier(n\_estimators=100, random\_state=42)**  **rf.fit(X, y)**  **importances = pd.Series(rf.feature\_importances\_, index=X.columns)**  **importances\_sorted = importances.sort\_values()**  **importances\_sorted.plot(kind='barh', color='Crimson')**  **plt.title('Features Importances')**  **plt.show()** |

|  |
| --- |
|  |

- According to the result above, top the most important feauture is choosen are: 'Monthly Income','Electricity Bills','Age','Collateral','Loan Amount','Time of Working'

**- Evaluation with algorithms**

### 1. Random Forest

**- Train set:**

|  |
| --- |
| **from sklearn.ensemble import RandomForestClassifier**  **from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, roc\_auc\_score**  **RF\_classifier = RandomForestClassifier()**  **RF\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred = RF\_classifier.predict(X\_train)**  **print(confusion\_matrix(y\_train,y\_pred))**  **print(classification\_report(y\_train,y\_pred))**  **print('Random Forest accuracy: ', accuracy\_score(y\_train, y\_pred))** |

|  |
| --- |
| [[1309 18]  [ 18 1352]]  precision recall f1-score support  0 0.99 0.99 0.99 1327  1 0.99 0.99 0.99 1370  accuracy 0.99 2697  macro avg 0.99 0.99 0.99 2697  weighted avg 0.99 0.99 0.99 2697  Random Forest accuracy: 0.9866518353726362 |

After examining the classification report table, the following can be observed:

* Class 0 includes 1327 instances of on-time debt repayment, with an accuracy rate of 99% for the entire set of predictions. However, the accuracy rate based on the actual number of observations is 99%. The F1 score obtained is 99%.
* Class 1 consists of 1370 observations of late debt repayment, and the accuracy rate for the entire set of predictions is 99%. However, the accuracy rate based on the actual number of observations is 99%. The F1 score achieved is 99%.

**- Test set:**

|  |
| --- |
| **RF\_classifier = RandomForestClassifier()**  **RF\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred = RF\_classifier.predict(X\_test)**  **print(confusion\_matrix(y\_test,y\_pred))**  **print(classification\_report(y\_test,y\_pred))**  **print('Random Forest accuracy: ', accuracy\_score(y\_test, y\_pred))** |

|  |
| --- |
| [[139 13]  [ 25 123]]  precision recall f1-score support  0 0.85 0.91 0.88 152  1 0.90 0.83 0.87 148  accuracy 0.87 300  macro avg 0.88 0.87 0.87 300  weighted avg 0.88 0.87 0.87 300  Random Forest accuracy: 0.8733333333333333 |

**- ROC Curve:**

|  |
| --- |
|  |

- The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). From the graphical plot above, the Random Forest classifiers gives curves close to the top-left corner, which indicates a goood performance. With AUC = 0.93, there is a high chance that the classifier will be able to distinguish the positive class values from the negative ones. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.

### 2. Logistic Regression

- **Train set:**

|  |
| --- |
| **from sklearn.linear\_model import LogisticRegression**  **LR\_classifier = LogisticRegression()**  **LR\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred2 = LR\_classifier.predict(X\_train)**  **print(confusion\_matrix(y\_train,y\_pred2))**  **print(classification\_report(y\_train,y\_pred2))**  **print('Logistic Regression accuracy: ', accuracy\_score(y\_train, y\_pred2))** |

|  |
| --- |
| [[952 375]  [371 999]]  precision recall f1-score support  0 0.72 0.72 0.72 1327  1 0.73 0.73 0.73 1370  accuracy 0.72 2697  macro avg 0.72 0.72 0.72 2697  weighted avg 0.72 0.72 0.72 2697  Logistic Regression accuracy: 0.7233963663329626 |

Looking at the classification report table we can see:

- In class 0: Paying debt on time has 1327 observations and the correct prediction rate on the total number of predictions is 72%, but the probability of correct prediction compared to the actual number of observations is 72%. achieved an F1\_score of 72%.

- In the class 1: the rate of late repayment has 1370 observations and the ratio of correct predictions on the total number of predictions is 73%, but the probability of correct prediction compared to actual observations is 73%. F1\_score is 73%, showing that the model's results are not good enough , which is to miss many cases of late repayment. Therefore, it is necessary to change the model to get results consistent with the goal of the article

**- Test set**

|  |
| --- |
| **LR\_classifier = LogisticRegression()**  **LR\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred2 = LR\_classifier.predict(X\_test)**  **print(confusion\_matrix(y\_test,y\_pred2))**  **print(classification\_report(y\_test,y\_pred2))**  **print('Logistic Regression accuracy: ', accuracy\_score(y\_test, y\_pred2))** |

|  |
| --- |
| [[113 39]  [ 43 105]]  precision recall f1-score support  0 0.72 0.74 0.73 152  1 0.73 0.71 0.72 148  accuracy 0.73 300  macro avg 0.73 0.73 0.73 300  weighted avg 0.73 0.73 0.73 300  Logistic Regression accuracy: 0.7266666666666667 |

|  |
| --- |
| **import statsmodels.api as SM**  **model\_1 = SM.Logit(y\_train, X\_train).fit()**  **print(model\_1.summary())** |

|  |
| --- |
| Optimization terminated successfully.  Current function value: 0.582516  Iterations 7  Logit Regression Results  ==============================================================================  Dep. Variable: y No. Observations: 2697  Model: Logit Df Residuals: 2691  Method: MLE Df Model: 5  Date: Thu, 25 May 2023 Pseudo R-squ.: 0.1595  Time: 10:24:33 Log-Likelihood: -1571.0  converged: True LL-Null: -1869.1  Covariance Type: nonrobust LLR p-value: 1.438e-126  ==============================================================================  coef std err z P>|z| [0.025 0.975]  ------------------------------------------------------------------------------  x1 -0.9065 0.085 -10.657 0.000 -1.073 -0.740  x2 -0.3874 0.045 -8.625 0.000 -0.475 -0.299  x3 -0.1548 0.043 -3.595 0.000 -0.239 -0.070  x4 0.6234 0.043 14.387 0.000 0.538 0.708  x5 0.0230 0.043 0.534 0.594 -0.061 0.107  x6 0.0205 0.043 0.478 0.633 -0.064 0.105  ============================================================================== |

* Current function value: The value of 0.582516 represents the currentvalue of the objective function at the optimized solution. In logistic regression, the objective function is typically the log-likelihood, which measures the fit of the model to the data. A lower value indicates a better fit, so 0.582516 suggests a relatively good fit of the model.
* Iterations: The optimization process took 7 iterations to converge. During each iteration, the model's parameters are updated to find the optimal values that minimize the objective function.

Coefficient Summary:

* x1: The coefficient of -0.9065 suggests that a one-unit increase in x1 is associated with a decrease of approximately 0.9065 in the log-odds of the outcome, holding other variables constant. The coefficient is statistically significant (P < 0.001), indicating that x1 has a significant impact on the outcome.
* x2: The coefficient of -0.3874 indicates that a one-unit increase in x2 is associated with a decrease of about 0.3874 in the log-odds of the outcome, controlling for other variables. Similar to x1, this coefficient is statistically significant (P < 0.001), suggesting x2 has a significant influence on the outcome.
* x3: The coefficient of -0.1548 suggests that a one-unit increase in x3 is associated with a decrease of approximately 0.1548 in the log-odds of the outcome, while other variables are held constant. Like x1 and x2, this coefficient is statistically significant (P < 0.001), indicating that x3 has a significant effect on the outcome.
* x4: The coefficient of 0.6234 implies that a one-unit increase in x4 is associated with an increase of about 0.6234 in the log-odds of the outcome, controlling for other variables. This coefficient is statistically significant (P < 0.001), suggesting x4 has a significant positive impact on the outcome.
* x5: The coefficient of 0.0230 suggests that a one-unit increase in x5 is associated with a slight increase of approximately 0.0230 in the log-odds of the outcome, holding other variables constant. However, this coefficient is not statistically significant (P = 0.594), indicating that x5 may not have a significant effect on the outcome.
* x6: The coefficient of 0.0205 implies that a one-unit increase in x6 is associated with a small increase of about 0.0205 in the log-odds of the outcome, while other variables are held constant. This coefficient is also not statistically significant (P = 0.633), suggesting that x6 may not have a significant impact on the outcome.

|  |
| --- |
|  |

- With AUC = 0.79, there is a chance that the classifier will be able to distinguish the positive class values from the negative ones but not as good as the previous althgorim.

### 3. Decision Tree

|  |
| --- |
| **from sklearn.tree import DecisionTreeClassifier**  **DT\_classifier = DecisionTreeClassifier(max\_depth=5,min\_samples\_leaf=100)**  **DT\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred3 = DT\_classifier.predict(X\_train)**  **print(confusion\_matrix(y\_train,y\_pred3))**  **print(classification\_report(y\_train,y\_pred3))**  **print('Decision Tree accuracy: ', accuracy\_score(y\_train, y\_pred3))** |

|  |
| --- |
| [[1215 112]  [ 280 1090]]  precision recall f1-score support  0 0.81 0.92 0.86 1327  1 0.91 0.80 0.85 1370  accuracy 0.85 2697  macro avg 0.86 0.86 0.85 2697  weighted avg 0.86 0.85 0.85 2697  Decision Tree accuracy: 0.8546533185020393 |

Looking at the classification report table we can see:

- Class 0 consists of 1327 observations of timely debt repayment, and the accuracy rate for the entire set of predictions is 81%. However, the accuracy rate based on the actual number of observations is 92%. The F1 score achieved is 86%.

- In class 1, there are 1370 observations of late debt repayment, and the accuracy rate for the entire set of predictions is 91%. However, the accuracy rate based on the actual number of observations is 80%. The F1 score is 85%, indicating that the model's results are unsatisfactory and it misses many cases of late repayment. Therefore, it is necessary to modify the model to obtain results that align with the objective of the article.

|  |
| --- |
| **DT\_classifier = DecisionTreeClassifier(max\_depth=5,min\_samples\_leaf=100)**  **DT\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred3 = DT\_classifier.predict(X\_test)**  **print(confusion\_matrix(y\_test,y\_pred3))**  **print(classification\_report(y\_test,y\_pred3))**  **print('Decision Tree accuracy: ', accuracy\_score(y\_test, y\_pred3))** |

|  |
| --- |
| [[141 11]  [ 25 123]]  precision recall f1-score support  0 0.85 0.93 0.89 152  1 0.92 0.83 0.87 148  accuracy 0.88 300  macro avg 0.88 0.88 0.88 300  weighted avg 0.88 0.88 0.88 300  Decision Tree accuracy: 0.88 |

**- Roc Curve:**

|  |
| --- |
|  |

**-** Based on the graph, the Decision Tree classifier exhibits curves that closely approach the top-left corner, indicating a strong performance. Its AUC value of 0.92 suggests a high likelihood of effectively discerning positive class values from negative ones. This capability stems from the classifier's proficiency in detecting a larger number of True positives and True negatives while minimizing occurrences of False negatives and False positives.

|  |
| --- |
| **from six import StringIO**  **from IPython.display import Image**  **from sklearn.tree import export\_graphviz**  **import pydotplus**  **dot\_data = StringIO()**  **clf = DT\_classifier2**  **clf.fit(X\_plot\_tree.values, y.ravel())**  **export\_graphviz(clf, out\_file=dot\_data, feature\_names = features, filled=True, rounded=True, special\_characters=True)**  **graph = pydotplus.graph\_from\_dot\_data(dot\_data.getvalue())**  **Image(graph.create\_png())** |

|  |
| --- |
|  |

The algorithm selected the attribute of customers with "Monthly Income" of less than VND 5,5 million, which is 2997 people to separate them into 2 main cases:

- Case 01 – not be able to repay debt on time:

+ Customer have monthly income undder 5,5 miliion per months.

+ Customer have monthy income from 5,5 million per months, electricity bill is less than 0,35 million and they are under 34 years old as well as don’t have collateral.

- Case 02 – repay debt on time:

+ Customers have high salary ( from 16,5 miliions to 32,5 millions) and their electricity bill is more than 0,55 millions.

+ Customer have monthlt income is less than 16,5 millions but they have collateral and have to pay for their electricity bill only under 1,25 millions.

### 4. XGBOOST

**-Train set**

|  |
| --- |
| **import xgboost as xgb**  **from xgboost.sklearn import XGBClassifier**  **XGB\_classifier = XGBClassifier()**  **XGB\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred4 = XGB\_classifier.predict(X\_train)**  **print(confusion\_matrix(y\_train,y\_pred4))**  **print(classification\_report(y\_train,y\_pred4))**  **print('XGBoost accuracy: ', accuracy\_score(y\_train, y\_pred4))** |

|  |
| --- |
| [[1291 36]  [ 52 1318]]  precision recall f1-score support  0 0.96 0.97 0.97 1327  1 0.97 0.96 0.97 1370  accuracy 0.97 2697  macro avg 0.97 0.97 0.97 2697  weighted avg 0.97 0.97 0.97 2697  XGBoost accuracy: 0.9673711531331108 |

- Class 0 comprises 1327 observations of debt repayment that is timely, and the accuracy rate for the entire set of predictions is 96%. However, the accuracy rate based on the actual number of observations is 97%. The F1 score obtained is 97%.

- Class 1 represents 1370 observations of debt repayment that is late, and the accuracy rate for the entire set of predictions is 97%. However, the accuracy rate based on the actual number of observations is 97%. The F1 score achieved is 97%.

**-Test set**

|  |
| --- |
| **XGB\_classifier = XGBClassifier()**  **XGB\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred4 = XGB\_classifier.predict(X\_test)**  **print(confusion\_matrix(y\_test,y\_pred4))**  **print(classification\_report(y\_test,y\_pred4))**  **print('XGBoost accuracy: ', accuracy\_score(y\_test, y\_pred4))** |

|  |
| --- |
| [[139 13]  [ 33 115]]  precision recall f1-score support  0 0.81 0.91 0.86 152  1 0.90 0.78 0.83 148  accuracy 0.85 300  macro avg 0.85 0.85 0.85 300  weighted avg 0.85 0.85 0.85 300  XGBoost accuracy: 0.8466666666666667 |

**Roc Curve:**

|  |
| --- |
|  |

- The graph demonstrates that the XG Boost Classifier’s curves closely align with the top-left corner, indicating a strong performance. The high AUC value of 0.92 suggests a high probability of accurately distinguishing positive class values from negative ones.

### 5. Support Vector Machine (SVM)

|  |
| --- |
| **from sklearn.svm import SVC**  **SVM\_classifier = SVC(kernel = 'linear', random\_state = 0)**  **SVM\_classifier.fit(X\_train, y\_train.ravel())**  **y\_pred = SVM\_classifier.predict(X\_train)**  **print('Confusion matrix:')**  **print(pd.DataFrame(confusion\_matrix(y\_train,y\_pred)),'\n')**  **print('Classification report:')**  **print(classification\_report(y\_train,y\_pred))**  **print('SVM accuracy: ', accuracy\_score(y\_train, y\_pred))** |

* **Train set:**

|  |
| --- |
| Confusion matrix:  0 1  0 1006 321  1 444 926  Classification report:  precision recall f1-score support  0 0.69 0.76 0.72 1327  1 0.74 0.68 0.71 1370  accuracy 0.72 2697  macro avg 0.72 0.72 0.72 2697  weighted avg 0.72 0.72 0.72 2697  SVM accuracy: 0.7163515016685206 |

- In class 0: On-time debt repayment has 1327 observations and the correct prediction rate on the total number of predictions is 69%, but the probability of correct prediction compared to the actual number of observations is 76%. achieve a F1\_score of 72%.

- In the target class 1: the rate of late repayment has 1370 observations and the percentage of correct predictions on the total number of predictions is 74%, but the probability of correct prediction compared to actual observations is 68%. F1\_score level is 71%.

**- Test set:**

|  |
| --- |
| **SVM\_classifier = SVC(kernel = 'linear', random\_state = 0)**  **SVM\_classifier.fsit(X\_train, y\_train.ravel())**  **y\_pred = SVM\_classifier.predict(X\_test)**  **print('Confusion matrix:')**  **print(pd.DataFrame(confusion\_matrix(y\_test,y\_pred)),'\n')**  **print('Classification report:')**  **print(classification\_report(y\_test,y\_pred))**  **print('SVM accuracy: ', accuracy\_score(y\_test, y\_pred))** |

|  |
| --- |
| Confusion matrix:  0 1  0 119 33  1 51 97  Classification report:  precision recall f1-score support  0 0.70 0.78 0.74 152  1 0.75 0.66 0.70 148  accuracy 0.72 300  macro avg 0.72 0.72 0.72 300  weighted avg 0.72 0.72 0.72 300  SVM accuracy: 0.72 |

### 6. ROC curve comparison

|  |
| --- |
| **from sklearn import metrics**  **from sklearn.metrics import RocCurveDisplay**  **disp = RocCurveDisplay.from\_estimator(RF\_classifier, X\_test, y\_test)**  **RocCurveDisplay.from\_estimator(LR\_classifier, X\_test, y\_test, ax = disp.ax\_);**  **RocCurveDisplay.from\_estimator(DT\_classifier, X\_test, y\_test, ax = disp.ax\_);**  **RocCurveDisplay.from\_estimator(XGB\_classifier, X\_test, y\_test, ax = disp.ax\_);**  **RocCurveDisplay.from\_estimator(SVM\_classifier, X\_test, y\_test, ax = disp.ax\_);** |

|  |
| --- |
|  |

- Even though the accuracies for the three models ( Random Forest, Decision Tree, XG Boost) are equivilent, the model with the higher AUC score will be more reliable because it takes into account the predicted probability. It is more likely to give you higher accuracy when predicting future data. It is evident from the plot that the AUC for the Random Forest ROC curve is higher than that for the other curves, with the AUC of 0.93. Therefore, we can say that Random Forest did a better job of classifying the positive class in the dataset.

### 7. 10-fold Cross -Validation

|  |
| --- |
| **models = [**  **('Logistic Regression', LogisticRegression()),**  **('Decision Tree', DecisionTreeClassifier()),**  **('Random Forest', RandomForestClassifier()),**  **('XGBOOST', XGBClassifier()),**  **('Support Vector Machine', SVC())**  **]**  **results = []**  **names = []**  **for name, model in models:**  **cv\_results = cross\_val\_score(model, X\_train, y\_train, cv=10, scoring='accuracy')**  **results.append(cv\_results)**  **names.append(name)**  **fig = plt.figure()**  **fig.suptitle('Machine Learning Algorithm Comparison')**  **ax = fig.add\_subplot(111)**  **plt.boxplot(results)**  **ax.set\_xticklabels(names)**  **plt.xlabel('Algorithm')**  **plt.ylabel('Accuracy')**  **plt.show()** |

|  |
| --- |
|  |

As depicted in picture above, a comparison can be understood about the model performance where the chosen accuracy measure is AUC. 10 Fold - Cross validation which is a form stratiﬁed sampling was used for evaluating our model performance overvarying splits of data. On observing picture, we can compare all of the initial AUC for 10 cross-valiation models the box plot displays the mean and the variance of the scores over the 10 models resulting from 10 fold - cross validation. Later thecross validation scores were compared against each other. Box-plots above show accuracies for the classifiers tested. The XG Boost model obtained the best forecasting resultcompared to the other techniques. This indicates that the XG Boost was able to use the raw data and temporal information for better classification results.

### 8. Predicting new data

**8.1. Predict new borrower with a Random sample test**

|  |
| --- |
| **target = ['Repayment Ability']**  **features = ['Monthly Income','Electricity Bills','Age','Collateral','Loan Amount','Time of Working']**  **X = data[features]**  **y = data[target]**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.1, random\_state=42)**  **XGB\_classifier.fit(X\_train.values, y\_train.values.ravel())**  **import random**  **num\_datasets = 5**  **datasets = []**  **for \_ in range(num\_datasets):**  **dataset = [**  **round(random.randint(2500000, 150000000), -5),**  **round(random.randint(0, 3000000), -5),**  **random.randint(20, 60),**  **random.choice([0, 1]),**  **random.randint(15000000, 300000000**  **round(random.randint(1, 15), -5)**  **]**  **datasets.append(dataset)** |

|  |
| --- |
| [[119100000, 2400000, 26, 1, 46062683, 0],  [30200000, 1000000, 40, 1, 225790612, 0],  [76900000, 1500000, 23, 1, 144835884, 0],  [71300000, 1600000, 31, 1, 221133247, 0],  [46300000, 2100000, 20, 0, 228291442, 0]] |

**-** Adding some random observations in order to be the input of predictng dafault.

|  |
| --- |
| **predictions = []**  **# Predict the "Repayment Ability" for each data set in datasets**  **for dataset in datasets:**  **prediction = XGB\_classifier.predict([dataset])**  **predictions.append(prediction)**  **for prediction in predictions:**  **print(prediction)** |

|  |
| --- |
| [0]  [0]  [0]  [0]  [0] |

- The ouput of this show us whether the customer will be able to repay debt on time.

**8.2. Predict new borrower with an available dataset**

|  |
| --- |
| **target = ['Repayment Ability']**  **features = ['Monthly Income','Electricity Bills','Age','Collateral','Loan Amount','Time of Working']**  **import pandas as pd**  **classifier = DecisionTreeClassifier**  **classifier.fit(X\_train, y\_train**  **dataset = pd.read\_excel('credit-predict-data.xlsx')**  **dataset.head()** |

|  |
| --- |
|  |

- Putting the new dataset to predict default base on th result of choosen model. This is a new raw data of new customers, without any preprocessing and will be used to test the result of model.

|  |
| --- |
| **dataset =dataset.rename(columns = {'Giới tính':'Sex','Hóa đơn tiền điện':'Electricity Bills',**  **'Số tiền vay':'Loan Amount','Mục đích vay ':'Loan Purpose',**  **'Gia đình':'Marital Status','Thời gian làm việc':'Time of Working',**  **'Tuổi':'Age','Thời hạn vay vốn':'Loan Term',**  **'Thu nhập hàng tháng':'Monthly Income','Sao kê bảng lương':'Salary Statement',**  **'Tài sản thế chấp':'Collateral','Khả năng trả nợ':'Repayment Ability'**  **})**  **dataset = dataset.replace({"Repayment Ability":{"Trả nợ đúng hạn":0,**  **"Trả nợ trễ hạn":1},**  **"Collateral": {"Có":0,**  **"Không":1},**  **"Salary Statement":{"Có":0,**  **"Không":1},**  **"Marital Status":{"Độc thân":1,**  **"Có gia đình":2,**  **"Ly hôn":3},**  **"Loan Purpose":{"Học tập":1,**  **"Tiêu dùng":2,**  **"Mua xe":3,**  **"Mua nhà":4,**  **"Đầu tư chứng khoán":5}})**  **dataset = dataset.drop(['ID'],axis=1)**  **dataset.head()** |

**-** In this step, we transform type of data from strring to interger as well as change the columns’s name.

|  |
| --- |
|  |

|  |
| --- |
| **X = dataset[features] # Extract the required features**  **predictions = classifier.predict(X)**  **dataset['Repayment Ability'] = predictions**  **output\_path = 'path\_to\_save\_updated\_data.csv'**  **dataset.to\_csv(output\_path, index=False)** |

|  |
| --- |
|  |

- The column “ Repayment Ability” of the dataset is now returned with result of “0” or “1” ( 0: On time repayment, 1: Late repayment). This is the outcome of thr prediting model using XGB classification.

# **4. CONCLUSION**

In the end, we employed various models to predict credit risk and utilized cross-validation to determine the most optimal approach. While the dataset comprises numerous variables, only a select few significantly impact the target variable, which is the ability to repay. These influential variables were utilized in conjunction with the chosen optimal model, XG Boost, to make predictions with new data. However, it is worth noting that another algorithm, Random Forest, also exerted a substantial influence on the results and can be employed to estimate or predict new observations as well.

**5. FUTURE WORK**

Upon comparing various algorithms on the train and test sets, we observe the presence of an overfitting issue in our model. This is evident as the model excels on the training data but performs inadequately on the test data. Multiple factors can account for this phenomenon. To enhance the model's practicality, several measures can be suggested. Firstly, employing regularization techniques may effectively mitigate overfitting in random forests. Secondly, refining parameters can enhance the performance of logistic regression and SVM. Lastly, exploring alternative models could aid in identifying the optimal approach for the credit classification problem.

## REFERENCES

García-Teruel, P. J., & Martinez-Solano, P. (2010). Determinants of trade credit: A comparative study of European SMEs. International Small Business Journal.

Ahmed, J., Xiaofeng, H., & Khalid, J. (2014). Determinants of trade credit: The case of a developing economy. European Researcher.

Van Sang, H., Nam, N. H., & Nhan, N. D. (2016). A novel credit scoring prediction model based on Feature Selection approach and parallel random forest. Indian Journal of Science and Technology.

Nguyen, T. D. D., Ha1-Nguyen, T. T., & Ngoc, B. (2021). Prediction of Consumer Credit risk in Vietnamese Commercial Banks. Journal of Economic and Banking Studies-Volume.

Pham, Q. H. (2021). An analysis of the credit rating system of firms in Vietnam.

Bao, W., Lianju, N., & Yue, K. (2019). Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. Expert Systems with Applications.

Leo, M., Sharma, S., & Maddulety, K. (2019). Machine learning in banking risk management: A literature review. Risks.