UNIVERSITY OF ECONOMICS AND LAW

BANKING AND FINANCE FACULTY

GRADUATION THESIS

PREDICTING PROFITABILITY OF NON-FINANCIAL ENTERPRISES IN VIETNAM USING HYBRID MACHINE LEARNING METHODOLOGY

Name: Trần Thanh Phúc

ID: K194141740

Class: K19414C

Instructor: PhD. Phạm Thị Thanh Xuân

Ho Chi Minh City, April 12, 2023

UNIVERSITY OF ECONOMICS AND LAW

BANKING AND FINANCE FACULTY

GRADUATION THESIS

PREDICTING PROFITABILITY OF NON-FINANCIAL ENTERPRISES IN VIETNAM USING HYBRID MACHINE LEARNING METHODOLOGY

Name: Trần Thanh Phúc

ID: K194141740

Class: K19414C

Instructor: PhD. Phạm Thị Thanh Xuân

Ho Chi Minh City, April 12, 2023

COMMENTS OF THE INSTRUCTOR

MARK	INSTRUCTOR (Signature and Full name)

COMMENTS

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

TABLE OF CONTENTS

LIST OF TABLES

LIST OF PICTURES

LIST OF ACRONYMS

Acronym	Write full
AUC	Area Under Curve
AUC-PR	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
CCC	Cash Conversion Cycle
CPI	Consumer Price Index
DSO	Days Sales Outstanding
EPS	Earnings Per Share
FA turnover ratio	Fix Assets Turnover Ratio
FA turnover ratio	Fix Assets Turnover Ratio
FPR	False Positive Rate
GDP	Gross Domestic Product
HNX	Hanoi Stock Exchange
HOSE	Hochiminh Stock Exchange
K-NN	K-Nearest Neighbors
ML	Machine Learning
NI	Net Income
OCF	Operating Cash Flow
PPE	Property, Plant, and Equipment
ROA	Return on Assets
ROC	Receiver Operating Characteristics
ROE	Return on Equity
SVM	Support Vector Machine
TA turnover ratio	Total Assets Turnover Ratio
TPR	True Positive Rate

SUMMARY

One of the important factors that help investors decide whether to invest in a business is the profitability of that business. Therefore, this study used data from non-financial companies listed on the HOSE and HNX exchanges in Vietnam for twelve years from 2010 to 2022 to predict the profitability of the enterprise for the next two years. In addition to basic financial factors, my research added some new variables such as macroeconomic variables, human resource variables, and industry variables. After processing the dataset to ensure the quality of the input data, six popular single machine learning algorithms were used to evaluate the predictive ability and find the model for the best performance. Then I develop more Hybrid Machine Learning models based on Ensemble Learning method. These models are developed by combining the single top performing Machine Learning model above with the rest. My method shows that Hybrid Machine Learning models actually increase performance compared to single Machine Learning models. Among them, the model that combines LightGBM and Neural Networks is the best performing model. Although the model does not give good results when forecasting unprofitable businesses, it does very well when forecasting how profitable businesses will be, which is also the main goal. The study also determines the level of impact of each variable on the profitability of non-financial enterprises in the future, thereby giving the variables that have the most impact and the least impact.

1. INTRODUCTION

1.1. Research motivation

Companies need an appropriate investment strategy to ensure sustainable growth in a competitive economic environment. Investors must analyze and evaluate a company's potential profitability to make accurate investment decisions and assess. Moreover, profit prediction helps companies to recognize risks early and make more effective strategies.

Machine Learning (ML) is an effective predictive tool for processing large and complex data to generate accurate and detailed prediction models, and comprehensively analyzing factors that impact a company's profitability. While traditional methods often rely on assumptions and explicit models, Machine Learning surpasses them in predictive ability due to its flexibility, ability to learn and develop over time, and efficient processing of large amounts of data, resulting in increased accuracy and reduced margin of error.

Research on the potential of using ML to predict a company's profitability shows great promise. However, most studies focus on specific models and lack generality, neglecting the advantages of other models. Additionally, previous studies have focused on markets in other countries, with little research on the Vietnamese market, limiting the practical applicability of research results.

Being aware of the above problems, I have carried out the topic "Forecasting the profitability of Vietnamese enterprises by Hybrid Machine Learning model".

1.2. Research objectives

The objective of this study is to predict whether Vietnamese companies can generate profits from their business activities in the future and identify the influential factors using a Hybrid Machine Learning model. Predicting a company's profitability is crucial as it helps investors mitigate investment risks by providing information about the company's likelihood of success. This information allows investors to assess the potential of the company and determine whether it is worth investing in. In addition, predicting profitability helps business managers to early detect risks and difficulties in the future and make more effective decisions and strategies to cope. Therefore, the aim of this study is to increase the success rate of investments for investors and support businesses in early understanding the current and future business situation, helping companies make more accurate decisions on organization, investment, and development. By applying ML to the research to handle the large, diverse, and unique Vietnamese market, this out-of-sample prediction study ensures its appropriateness and high applicability for both investors and businesses in Vietnam and beyond.

1.3. Research scope

Object: Non-financial enterprises listed on two stock exchanges in Ho Chi Minh City (HOSE) and Hanoi (HNX).

Time: From 2010 to 2022.

Factors: Financial factors of enterprises themselves, human factors and macro factors.

1.4. Research results

The research results will provide the ability to forecast the profitability of Vietnamese companies in the next two years, thereby supporting investors in making more informed and accurate investment decisions. Additionally, the study will provide a list of factors that influence the profitability of companies, helping both companies and investors to easily assess the future situation of businesses and reduce risks.

1.5. Research contributions

Update: This study was conducted on a very recent dataset up to 2022, very close to the current period, which is the first half of 2023.

New factors: In addition to the usual financial factors of companies, this study includes new factors such as the company's human resources, industry factors, and macroeconomic factors.

New model development: A new ML model has been developed by selecting the best-performing ML model and combining it with the remaining single ML models. This new model is called Hybrid Machine Learning, which has higher performance.

2. THEORETICAL OVERVIEW AND PREVIOUS STUDIES

2.1. Theoretical overview

2.1.1. Profitability

Profitability of a company refers to its ability to make a profit or financial gain from its business activities. It measures how well the company is doing financially by comparing the total money earned to the total expenses during a specific time, like a year or a quarter.

If a company's revenue is higher than its expenses, it is considered profitable. This means the company is managing its resources well, controlling costs, and earning enough money from its products or services. Profitability is essential for a company's overall financial health and sustainability.

Being profitable not only shows that a company is financially successful, but it also creates a strong foundation for future growth, reinvestment, and creating value for shareholders. It enables the company to meet its financial obligations, invest in research and development, expand its operations, reward stakeholders, and handle economic downturns or unexpected challenges.

2.1.2. Net income (NI)

Net income is the profit a company earns after subtracting all expenses, taxes, and other deductions from its total revenue. It represents the amount of money that remains for the company after covering all the costs related to its operations. Net income is a crucial measure of a company's financial performance and profitability. It shows how effectively the company manages its expenses and generates profit from its business activities.

$$\text{Net Income} = \text{Total Revenue} - \text{Total Expenses}$$

To calculate net income, you take the total revenue and subtract all the expenses. The result is the net income, which shows whether the company made a profit or a loss during that specific time. If the net income is positive, it means the company earned more money than it spent and made a profit. But if the net income is negative, it means the company spent more money than it earned, resulting in a loss.

2.1.3. Operating Cash Flow (OCF)

Operating cash flow is the cash generated or used by a company's main activities, excluding financing and investing. It includes the cash coming in and going out from the day-to-day business operations.

This measure is crucial for assessing a company's financial well-being and how well it manages its cash flow. It indicates how effectively the company generates cash from its core operations, such as sales, collecting payments, and paying expenses related to running the business.

$$\text{Operating Cash Flow} = \text{Net Income} + \text{Depreciation} + \text{Amortization} + \text{Non-cash Expenses} - \text{Changes in Working Capital}$$

A positive operating cash flow indicates that the company generates more cash than it spends on operations, reflecting sustainability and the ability to cover expenses and investments. Conversely, a negative operating cash flow raises concerns about cash flow management and overall profitability.

2.1.3. Reasons to use both Net Income and Oparating Cash Flow

Net income (NI) and Operating Cash Flow (OCF) are important metrics for predicting a company's profitability, as they provide a detailed picture of financial performance, stability, and potential. company growth. They are all important indicators but each has its own advantages and limitations.

Net Income measures a business's after-tax profit for a given period of time, including expenses and income unrelated to the company's main business. However, Net Income has some limitations as follows:

- Revenues and expenditures not related to the main business of the company such as fixed asset costs, interest expenses, income from investment and financing activities that may affect the profit of the company. company and lose the accuracy of the Net Income index.
- Net Income does not reflect changes in a company's cash flow and does not help assess a company's ability to pay off debt and finance.

Operating Cash Flow (OCF) measures cash flow from a company's operations, providing information about a company's ability to generate cash from its core business and its ability to pay off debt and finance and invest into new activities and pay dividends to shareholders. However, OCF also has some limitations as follows:

- OCF does not reflect expenses unrelated to the company's core business and does not measure the company's profitability.
- OCF can be affected by the company's long-term investments and does not fully reflect the company's long-term profitability.

Therefore, to evaluate the profitability of a business, both Net Income and OCF should be used to get a more comprehensive and accurate view of a company's financial position. By analyzing both metrics, investors can gauge a company's financial strength, stability, and growth potential. Companies with high net income and positive OCF may have a more stable and secure financial position than those with low net income and negative OCF. Furthermore, evaluating both metrics can help investors identify potential risks, such as high debt levels or poor operating performance, that could affect a company's future profitability.

2.1.4. Machine Learning

Machine learning is a subset of artificial intelligence (AI) that focuses on creating computer systems capable of learning and making predictions or decisions without explicit programming. It involves designing algorithms that can automatically learn and improve from data, enabling them to recognize patterns and make accurate predictions.

Machine learning is one way to use AI. It was defined in the 1950s by AI pioneer Arthur Samuel as “the field of study that gives computers the ability to learn without explicitly being programmed.” Machine learning algorithms are trained using large datasets to identify patterns, extract insights, and make predictions. The algorithms adjust their rules or parameters to optimize performance and achieve desired outcomes.

Machine learning is a versatile field with applications in image and speech recognition, natural language processing, recommendation systems, fraud detection, autonomous vehicles, and more. It excels in handling complex data, automating tasks, and delivering accurate predictions. Common machine learning algorithms like Linear Regression, Logistic Regression, Decision Trees, Random Forest, and Naive Bayes contribute to solving diverse problems by learning from data.

2.1.5. Hybrid Ensemble Machine Learning

Hybrid Machine Learning is combining two or more different machine learning techniques to solve a problem. In it, Hybrid Ensemble Machine Learning is combining different machine learning models to form a better final predictive model. Ensemble Learning can use many techniques such as Voting, Bagging, Boosting, Stacking, ... to combine different models. The purpose of Ensemble Learning is to leverage the power of various models to reduce errors and improve the accuracy of the predictive model.

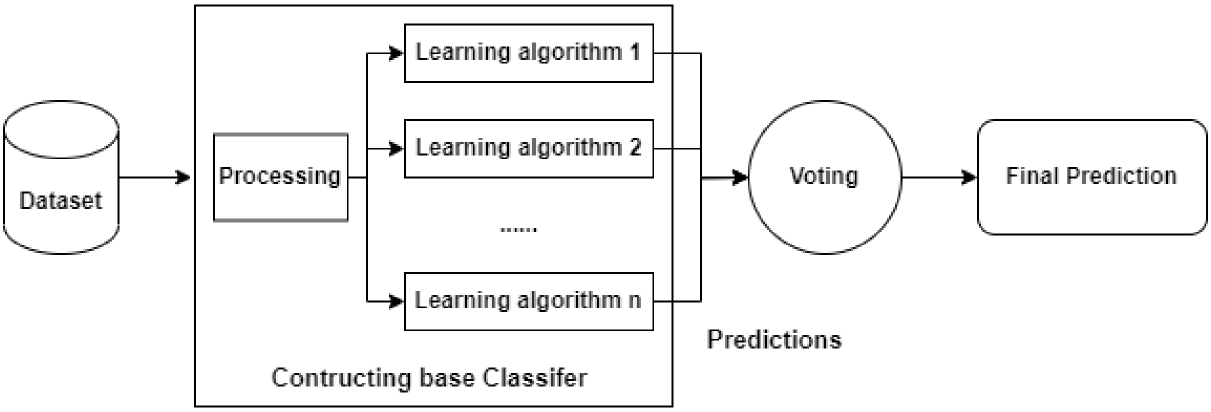


Figure 1: Working principle of Hybrid Ensemble Machine Learning using Voting method

Hybrid ensemble machine learning using the voting method is a technique that combines multiple machine learning models. Each model independently makes predictions, and the final prediction is determined by a voting scheme, where the most common prediction is selected. This approach improves accuracy and robustness by leveraging the collective decision-making of diverse models.

2.2. Previous studies

TS. Nguyen, H, A. TS. Nguyen, T, T. (2017) Factors affecting the profit of enterprises listed on HOSE. This study evaluated the factors affecting the quality of income of enterprises listed on the Ho Chi Minh Stock Exchange. Ho Chi Minh City (HOSE) in the period from 2012-2016, research shows that enterprises have high growth rate (Growth), high liquidity (Liquid), size (Size), financial leverage (Lev), capital asset investment ratio (PPE) or companies that have a high foreign ownership ratio tend to have higher quality earnings, whereas those with high financial leverage or a high ratio of asset investment may have lower quality income.

TS. Nguyen, A. P. (2021). Measuring business performance through market value index and book value index by machine learning method. The author used the variables Operating Cash Flow (OCF), Return on Equity (ROE), Return on Assets (ROA) and Age of the business (Age) to measure a company's operational efficiency using machine learning.

Ho Xuan Thuy, Dinh Le Minh Hieu, Dzoan Khoa Danh (2020). The factors affect financial performance of companies listed on Hanoi stock exchange. This study investigates the impact of factors on the financial performance of listed companies on Hanoi Stock Exchange (HNX) period 2013-2017. Factors considered include corporate income tax (TAX), company size, company growth rate (GROWTH), company age (AGE) and liquidity (LIQUID). The study found a significant negative relationship between corporate income tax, firm size and firm growth and financial performance. In contrast, there is a significant positive relationship between liquidity and financial performance.

Robert Brunner, Vic Anand, và Kelechi Ikegwu (2019). Predicting Profitability Using Machine Learning. This study utilized the Random Forest and Decision Tree machine learning methods to predict changes in profitability beyond the sample within the measured year. This approach achieved a classification accuracy of 57-64%, higher than random guessing. The results indicate that machine learning methods perform better than traditional regression-based approaches, especially in predicting cash flows and using aggregation to forecast future cash flows.

Alarussi, A.S., Alhaderi, S.M (2018). Factors affecting profitability in Malaysia. This study examined variables such as firm size (Size), liquidity (Liquid) and financial leverage (Lev). The findings show a significant positive correlation between total revenue and profitability, while financial leverage is negatively related to profitability. However, the study did not find any significant relationship between liquidity and profitability.

Coad, Alex et al (2016). Firm age and performance: A replication and extension considering new and continuing firms. This study has studied innovation and the age of firms. They show that businesses with an average age of 5 to 10 years tend to be more profitable than younger or older businesses. The authors suggest that this may be because businesses at this age have passed the initial start-up stage and are in the growth stage, have more stability and often have more experience in business operations management.

Syahbandar, N., & Lestari, N. (2023). Factors Affecting Profitability in Manufacturing Sector Companies Listed on BEI. This study investigates the factors affecting a company's profitability using quantitative methods. The research focuses on manufacturing companies listed on the Indonesia Stock Exchange from 2015 to 2019. Purposive sampling is employed to select the sample, and data is collected from the company's financial statements stored in a database. The results indicate that firm size (FS), leverage (DER), liquidity (CR), and sales growth (SG) do not significantly affect profitability. However, working capital and company efficiency show a significant impact on profitability. This study offers valuable insights into profitability factors in the manufacturing sector on the Indonesia Stock Exchange.

3. RESEARCH METHOD

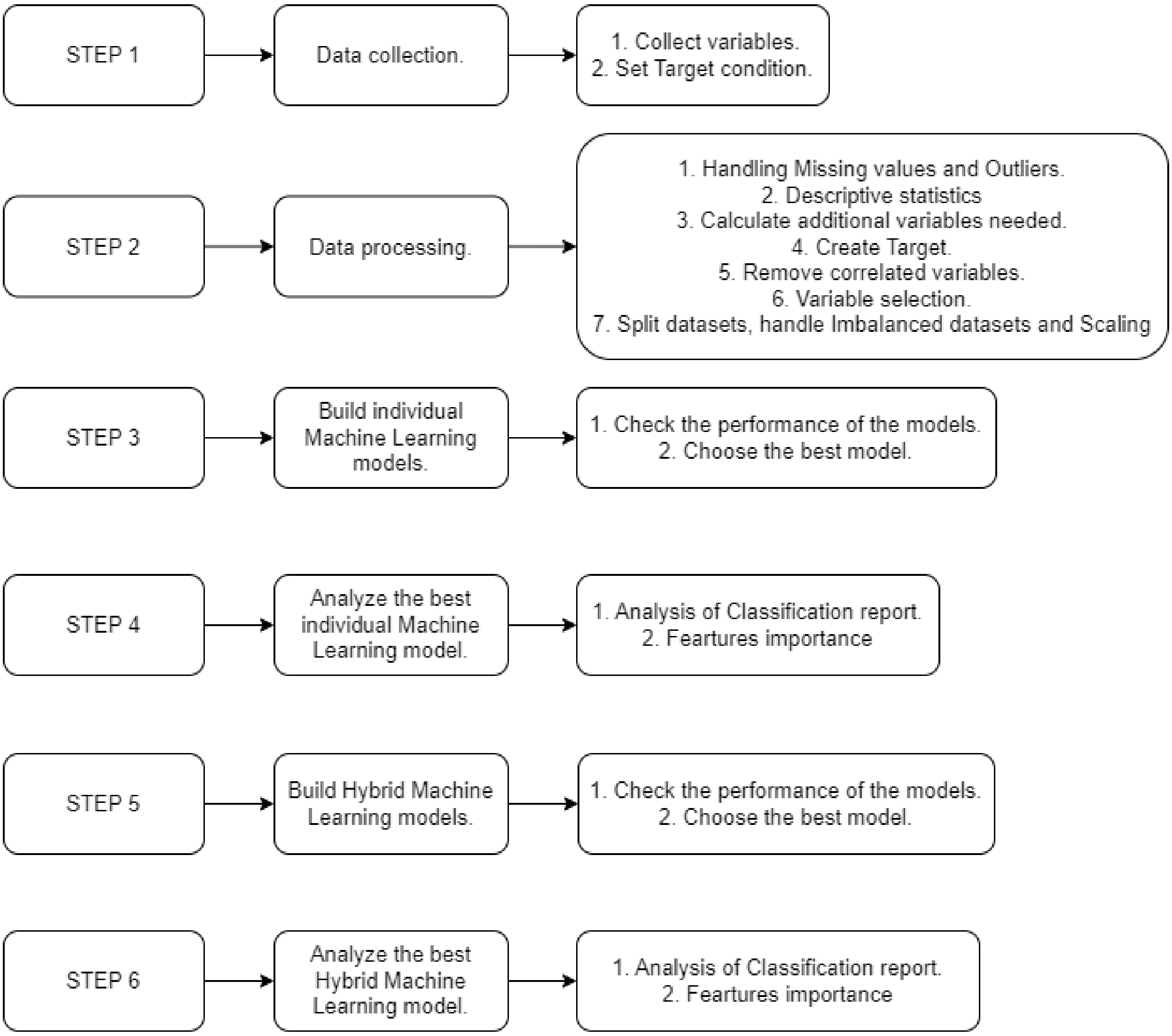

STEP 1
Data collection.
1. Collect variables.
2. Set Target condition.
STEP 2
Data processing.
1. Handling Missing values and Outliers.
2. Descriptive statistics
3. Calculate additional variables needed.
4. Create Target.
5. Remove correlated variables.
6. Variable selection.
7. Split datasets, handle Imbalanced datasets and Scaling
STEP 3
Build individual Machine Learning models.
1. Check the performance of the models.
2. Choose the best model.
STEP 4
Analyze the best individual Machine Learning model.
1. Analysis of Classification report.
2. Feartures importance
STEP 5
Build Hybrid Machine Learning models.
1. Check the performance of the models.
2. Choose the best model.
STEP 6
Analyze the best Hybrid Machine Learning model.
1. Analysis of Classification report.
2. Feartures importance

Figure 2: Analytics processing

3.1. STEP 1: Data collection.

The dataset used in this study is sourced from Thomson Reuter, the State Bank of Vietnam, and the keywords collected by me through the annual financial reports of companies. It includes all non-financial listed companies on the HOSE and HNX stock exchanges for thirteen years from 2009 to 2022, containing various types of variables such as micro, macro, industry, and company personnel variables.

In this step, I will explain the meaning and calculation formula of each variable. In addition, the Target variable is also explained in detail the conditions for labeling observations in the data set. In which, the label 0 is assigned to businesses with profit potential and 1 is the opposite.

3.2. STEP 2: Data processing.

In this step, I perform data processing on the data obtained from Step 1. First, I handle missing observations and process outliers in the dataset . Second, I conduct descriptive statistics for the dataset. Third, I calculate additional variables necessary for the collected data because they are only raw data and may not contain all the variables I need. Fourth, I rely on the conditions explained in step 1 on the Target variable to calculate and label each observation, from which the Target column is further created in the dataset. Then, I eliminate highly correlated variables and select important variables that have a strong impact on the dataset. Finally, I split the dataset into a training set and a testing set, handle imbalances in the training set, and scale the variables to have the same proportion.

3.3. STEP 3: Build single Machine Learning models.

In this step, I begin to feed the data in the training set into building models for single ML algorithms. I will find the best machine learning model among the trained models. GridSearch and K-fold cross-validation were used to build single ML models to search for optimal hyperparameters, which helps the model generalize better with new data and reduces overfitting by dividing the data into multiple parts for more accurate model evaluation. The single algorithms were compared in terms of prediction performance using three factors: Accuracy, AUROC, and AUC-PR. Then, I will select the model with the best performance to use for subsequent steps.

3.4. STEP 4: Analyze the best single Machine Learning model.

From the best-performing model in the previous step, I start to analyze its forecasting ability in more detail by using Classification report. Then, I present a chart showing the importance of each variable in the dataset for that model.

3.5. STEP 5: Build Hybrid Machine Learning models.

The objective of this phase is to integrate the best single ML model with other models to create the most effective Hybrid ML mode. Therefore, after finding the best-performing single ML model in the previous steps, that model will be used as a base model to combine with the remaining single ML models to create Hybrid ML models. Then, each model will be tested for its forecasting performance to find the best model.

3.6. STEP 6: Analyze the best Hybrid Machine Learning model.

Similar to step 4, after finding the best-performing Hybrid ML model, that model will be analyzed in more detail by using a Classification report and comparing it with the best-performing single machine learning model in step 4 to demonstrate the superior performance of the Hybrid ML model. Then, the graph showing the strength of each variable in the model will be analyzed to know how much influence they have on the model.

In addition, after having two charts measuring the level of importance of each variable for the two models, I will combine them to identify more accurately the strong variables and the variables that have less impact on the forecasting ability.

4. RESULTS

4.1. STEP 1: Data collection.

4.1.1. Variables.

The independent variables used in this study are taken first from the dataset that I have collected from Thomson Reuter, the State Bank of Vietnam, and the data collected by me through the annual financial reports of companies. Then, there are some variables are calculated indirectly from the datase. A total of 21 independent variables were identified and used in the study.

Table 1: List of variables

No	Full Name	Short Name	Variable Type
1	Age	Age	Micro
2	Board size	Board size	Micro
3	Cash Conversion Cycle	CCC	Micro
4	Earnings Per Share	EPS	Micro
5	Fix Assets turnover ratio	FA turnover ratio	Micro
6	Free cash flow	FCF	Micro
7	Gross Domestic Product	GDP	Macro
8	Gross profit	Gross profit	Micro
9	Growth	Growth	Micro
10	Industries	Industries	Micro
11	Inflation	Inflation	Macro
12	Inventory turnover ratio	Inventory turnover ratio	Micro

13	Leverage	Leverage	Micro
14	Liquidity	Liquidity	Micro
15	Management size	Management size	Micro
16	Property, Plant, and Equipment	PPE	Micro
17	Quick ratio	Quick ratio	Micro
18	Return on Assets	ROA	Micro
19	Return on Equity	ROE	Micro
20	Size	Size	Micro
21	Total Assets turnover ratio	TA turnover ratio	Micro

4.1.1.1. Age

The life expectancy of the business is calculated by the total number of years from the time the business was established to the present.

$$\text{Age} = \text{Current year} - \text{Year of establishment}$$

4.1.1.2. Board Size

Board size is the number of members on the Board of Directors of a business.

4.1.1.3. Cash Conversion Cycle (CCC)

Cash Conversion Cycle (CCC) is the time taken to convert a company's production costs into cash and is used to gauge financial management. If the CCC is too large, it may indicate that the company is having difficulty managing cash flow.

$$\text{CCC} = \text{Days Inventory Outstanding} + \text{Days Sales Outstanding} - \text{Days Payable Outstanding}$$

4.1.1.4. Earnings Per Share (EPS)

Earnings Per Share (EPS) is a metric to measure a company's earnings per share. The higher the EPS, the more attractive a company's stock is to investors.

EPS = (Net Income - Preferred Dividends) / Weighted Average Number of Common Shares Outstanding

4.1.1.5. Fix Assets Turnover Ratio

The Fixed Assets Turnover Ratio measures the efficiency of a company's fixed assets in generating revenue. It is used to compare the efficiency of using fixed assets.

Fixed Assets Turnover Ratio = Net Sales / Average Fixed Assets

4.1.1.6. Free Cash Flow (FCF)

Free Cash Flow (FCF) is a key financial metric that measures the cash generated by a company's operations after deducting necessary expenses and investments. It indicates the amount of cash available for debt repayment, dividends, asset acquisition, and new business opportunities. FCF serves as a gauge of a company's financial and business health, providing insights into its ability to generate surplus cash and make strategic decisions.

Free Cash Flow = Operating Cash Flow - Capital Expenditures

4.1.1.7. Gross Domestic Product (GDP)

Gross Domestic Product (GDP) measures the total value of production in a country in a year, which is an important indicator for measuring economic health, comparing with other countries, or tracking changes in the economy over time.

4.1.1.8. Gross Profit

Gross Profit is a company's profit after deducting production and service costs. This indicator measures the level of efficiency in managing production costs and pricing.

Gross Profit = Revenue - Cost of Goods Sold

4.1.1.9. Growth

Growth rate is a measurement of the change in an investment over a period of time, typically used to evaluate past growth and predict future trends.

$$\text{Growth} = (\text{Ending Value} / \text{Beginning Value})\ \text{^}\ (1/\text{Number of Years}) - 1$$

4.1.1.10. Industries

Industries categorize businesses based on common characteristics such as their products, services, technology, and production scale. This classification facilitates analysis and decision-making by grouping similar businesses together. The primary focus of a business plays a crucial role in determining its industry. Industries can be further divided into sectors and sub-industries, offering a framework to comprehend specific segments of the economy, and make well-informed assessments about market dynamics and business strategies.

4.1.1.11. Inflation

Inflation refers to the general increase in prices of goods and services over time, leading to a decrease in the purchasing power of money. Inflation can be caused by factors such as increased demand, rising production costs, or excessive money supply.

4.1.1.12. Inventory turnover ratio

Inventory turnover ratio is a measure of the number of times inventory is sold during a period of time (usually a year) to assess the speed of sales and the ability of a business to manage inventory.

$$\text{Inventory turnover ratio} = \text{Cost of goods sold} / \text{Average inventory}$$

4.1.1.13. Leverage

Leverage involves using borrowed funds or financial instruments to increase potential investment returns. It allows businesses to control larger assets with a smaller initial investment. It can be achieved through borrowing money or using financial instruments like derivatives. Leverage has the potential for higher profits and increased risk.

$$\text{Leverage} = \text{Total debt} / \text{Equity}$$

4.1.1.14. Liquidity

Liquidity refers to a business's ability to convert assets into cash quickly and easily without significantly impacting their value. It is crucial for meeting short-term financial obligations and handling unexpected expenses. Cash, marketable securities, and accounts receivable are examples of liquid assets.

$$\text{Liquidity ratio} = \text{Current assets} / \text{Current liabilities}$$

4.1.1.15. Management size

Management size is the size of the management team, operating in a business or organization.

4.1.1.16. Property, Plant, and Equipment (PPE)

Property, Plant, and Equipment (PPE) is a fixed asset consisting of land, plant, machinery, equipment, vehicles, and other assets used by a business to produce goods or provide services.

$$\text{PPE} = \text{Cost of assets} + \text{Improvements} - \text{Accumulated depreciation}$$

4.1.1.17. Quick ratio

Quick ratio (also known as Acid Test ratio) measures the short-term solvency of a business or financial institution, measuring the ability to pay short-term debt without selling current assets for cash.

$$\text{Quick ratio} = (\text{Current assets} - \text{Inventory} - \text{Prepaid expenses}) / \text{Current liabilities}$$

4.1.1.18. Return on Assets (ROA)

Return on Assets (ROA) is a financial metric that measures how effectively a business or financial institution is using its assets. It measures a business's ability to generate profit from each unit of asset used.

$$\text{ROA} = \text{Net income} / \text{Average total assets}$$

4.1.1.19. Return on Equity (ROE)

Return on Equity (ROE) is a financial metric that measures how effectively a business or financial institution is using its equity. It measures a business's ability to generate a return from each unit of equity invested.

ROE = Net income / Average Shareholders' Equity

4.1.1.20. Size

The size of a company refers to its scale and magnitude, which can be measured using various indicators such as revenue, market capitalization, assets, and number of employees. It represents the extent of a company's operations and its influence in the market. The size of a company can change over time due to factors like growth, mergers, and market conditions.

4.1.1.21. Total Assets turnover ratio

Total Assets turnover ratio is a financial metric that measures how efficiently a business uses its assets in generating revenue. It measures a business's ability to generate revenue from each unit of asset used.

Total Assets turnover ratio = Net Sales / Average Total Assets

4.1.2. Target explanation

The purpose of this study is to determine whether a business is capable of generating profits from its operations in year 2 (T+2) from the data collection year (T). Therefore, the target variable of the dataset is classified into two labels: label 0 and label 1. Label 0 is applied to observations that indicate that those businesses have generated profits from their operations in year T+2. The condition for using label 0 is that the Net Income and OCF of those businesses in year T+2 are both positive. Conversely, label 1 will be used for businesses that did not make a profit in year T+2, as determined by their non-positive Net Income and OCF in year T+2.

4.2. STEP 2: Data processing.

4.2.1. Handling Missing values and Outliers

First, I checked the year of establishment of all businesses in the dataset and removed businesses established from 2016 to the present. Since these businesses will have no data for previous years, it will affect the dataset. Therefore, I decided to remove these businesses before checking missing values for each input variable.

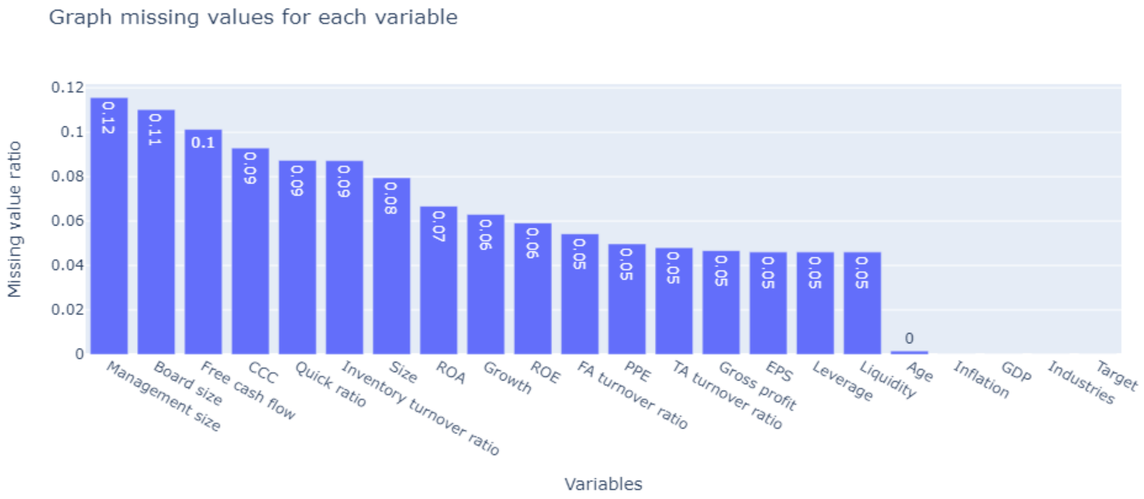


Figure 3: Number of missing values of each feature (%)

Next, I used interpolation to fill in the missing values, and the results showed that the number of missing values decreased significantly (less than 5%). However, there were still some variables in certain businesses that were completely missing, and interpolation could not fill in these gaps. Therefore, these observations with completely missing values were removed from the dataset.

Then, I used Histogram and describe function to check for outliers in the variables. One-Class SVM (Support Vector Machine) was used to detect and handle the outliers.

4.2.2. Descriptive statistics

Table 2: Descriptive statistics

	Gross profit	FCF	Growth	PPE	Liquidity	Size	Leverage
count	4669	4.67E+03	4669	4669	4669	4669	4669
mean	0.19558	-1.98E+10	0.285447	0.839825	2.223452	26.34007	0.511461
std	0.335712	1.31E+12	2.938673	7.12948	2.457197	1.716325	0.215406
min	-11.274236	-5.48E+13	-0.99294	0.000158	0.097129	21.71652	0.00572
25%	0.095122	-4.29E+10	-0.07863	0.076196	1.115955	25.15078	0.350921
50%	0.160072	2.53E+09	0.074607	0.197344	1.487024	26.14557	0.532662
75%	0.262616	4.65E+10	0.231005	0.489758	2.364783	27.33897	0.682985
max	11.221834	1.38E+13	127.4579	339.1962	44.42742	33.59044	0.969633

	Quick ratio	Inventory turnover	FA turnover	TA turnover	ROA	ROE	EPS
count	4669	4.67E+03	4669	4669	4669	4669	4669
mean	1.423526	3.63E+03	25.230358	1.218596	0.065524	0.115166	1790.779
std	2.235175	1.40E+05	181.29275	1.226152	0.078094	0.166489	2321.572
min	-16.068937	2.44E-03	0.002948	0.001044	-0.49186	-3.844775	-9001.25
25%	0.50836	2.50E+00	2.041824	0.486318	0.017913	0.047944	535.9111
50%	0.866193	4.95E+00	5.067294	0.931475	0.04948	0.109821	1358.91
75%	1.492295	1.16E+01	13.123973	1.542022	0.095429	0.178598	2493.563
max	40.205171	8.78E+06	6322.9966	12.733542	0.828778	0.953963	30069.25

	CCC	Age	Inflation	GDP	Board size	Management	Industries	Target
count	4669	4669	4669	4669	4669	4669	4669	4669
mean	479.42307	15.014564	0.054933	6.01399	5.641037	3.913258	3.073677	0.342686
std	4250.112	7.833899	0.046331	1.154455	1.399166	1.796876	2.093296	0.474658
min	-14883.616	2	0.0063	2.905836	2	0	0	0
25%	50.893351	10	0.028	5.421883	5	3	2	0
50%	115.48813	13	0.0354	6.240303	5	4	2	0
75%	226.26726	19	0.0659	6.812246	6	5	4	1
max	150425.23	60	0.1868	7.075789	18	19	8	1

I use the describe() method to summarize the basic parameters of a dataset. It provides users with quick access to key statistical parameters such as dataset size, mean, variance, minimum and maximum values of columns, aiding in detecting outliers, missing values, and uneven data distribution. It's a useful tool for summarizing and analyzing data, providing an overview and detecting unusual values.

4.2.3. Calculate additional variables needed

One thing to note is that the independent variables I mentioned earlier have been processed. In reality, the original dataset does not have many variables, and they have to go through the calculation stage here to have all the variables needed for the study. Some variables have been calculated at this stage from the original dataset, such as Size, Leverage, Liquidity, Growth, Free cash flow, etc.

4.2.4. Create Target

At this stage, the dataset is nearly complete, so I began creating the Target variable and assigning labels to each observation. As mentioned in the Purpose and Results section, this study predicts the future two years, so the dataset will have a lag of two years. The Target variable will be applied according to the conditions that I mentioned in the Target section of the study, which will be shifted back two years. Therefore, independent variables in the years 2021 and 2022 will be removed, and the dataset will only consider the time period from 2010 to 2020.

4.2.5. Remove correlated variables

I used the feature_engine library to identify highly correlated and unimportant variables in the dataset. After setting the condition of no correlation above 50%, only 17 variables were retained, including: Gross profit, Free cash flow, Growth, PPE, Liquidity, Size, Inventory turnover ratio, FA turnover ratio, TA turnover ratio, ROA, CCC, Age, Inflation, GDP, Board size, Management size, Industries. Below is the Correlation Matrix between the variables and the Correlation Chart between the attributes and the target variable.

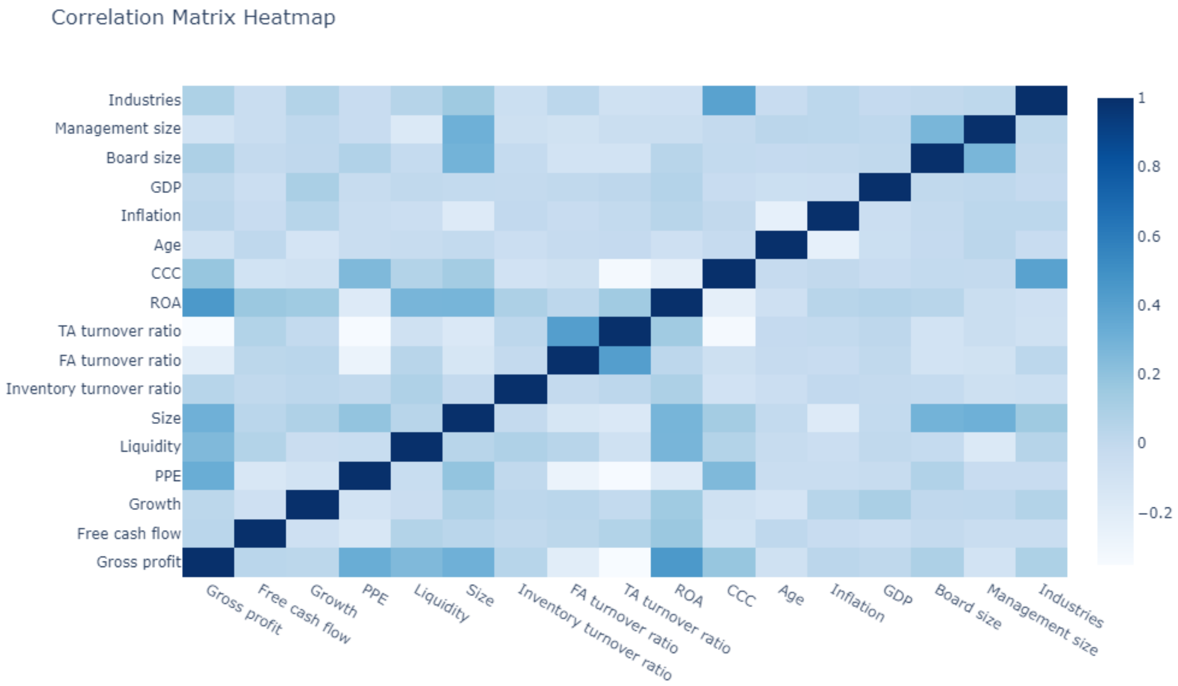


Figure 4: Correlation matrix heat map between features

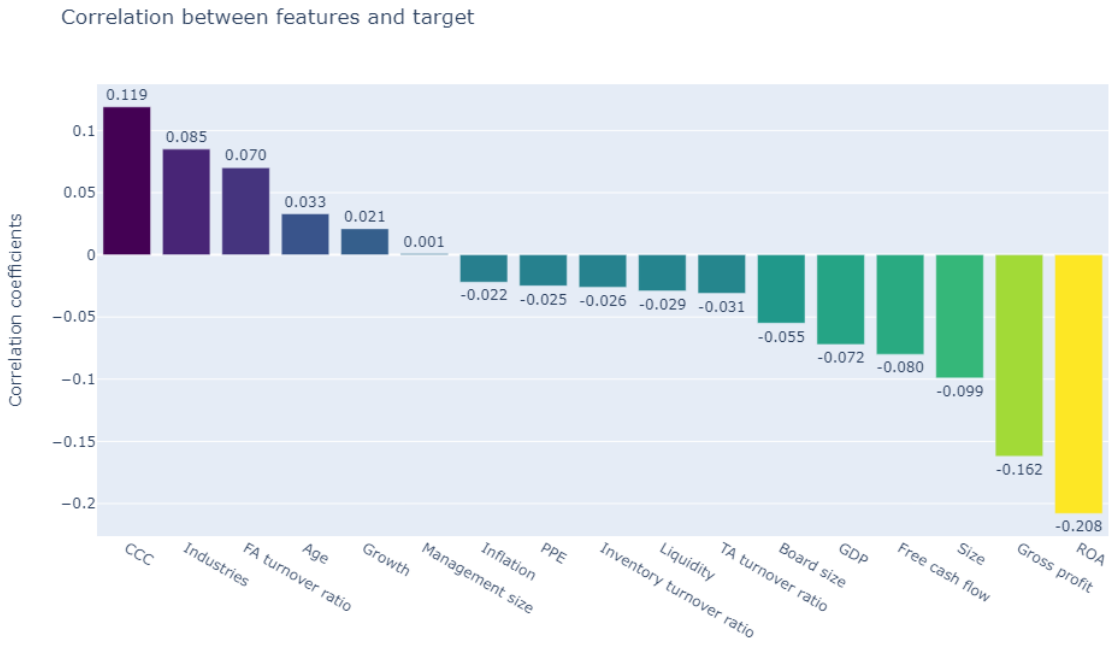


Figure 5: Correlation graph between features and target

Above is the correlation chart between categorical variables and Target variables. It is easy to see that none of the variables is highly correlated with the Target variable.

4.2.6. Fetures selection

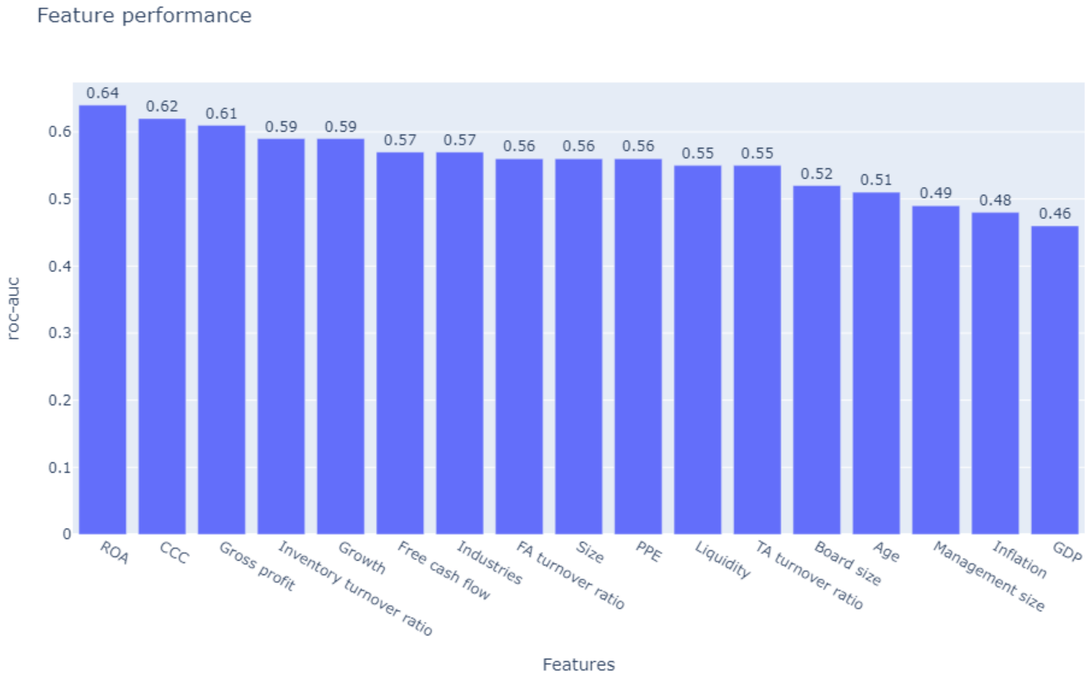


Figure 6: Important features

The feature_engine library was also used to identify important variables. However, because the results of the variables were not too distinguishable so no variables were removed.

4.2.7. Split datasets, handle Imbalanced datasets and Scaling

Before fitting the dataset into the model, the dataset needs to be divided into two parts, the training set and the test set with a ratio of 80:20. After checking, I found that the training set was imbalanced with 3069 observations labeled as 0 and 1600 observations labeled as 1. To address this issue, I used the Tomek Links function to undersample the dataset for balancing. Then, the dataset was scaled to bring the variables into the same value range to avoid the influence of variables with different units, ranges, or weights on the model's results.

4.3. STEP 3: Build single Machine Learning models.

After completing the data preprocessing stage, the dataset is complete and can be fed into the model for training. Six algorithms will be selected to develop the model in this study: Logistic Regression, Random Forest, Naive Bayes, K-Nearest Neighbors, Neural Networks, and LightGBM.

Table 3: List of Machine Learning algorithms

Name	Explantion	Advantages
Logistic Regression	Logistic Regression is a machine learning method for predicting output values based on input variables. It's used for discrete predictions, like grouping samples.	The advantages include its ease of understanding, implementation, and effectiveness for classification problems with large data sets. It's also good at interacting with input variables without overfitting.
Random Forest	Random Forest combines multiple decision trees to predict output values and outperforms single decision trees.	Its advantages include handling large data sets, minimizing overfitting risks, multi-class classification, and estimating input variable importance.
Naive Bayes	Naive Bayes is a classification model based on Bayes theorem. It is based on probability where the	It is simple, fast and efficient for large classification problems. It handles both discrete and continuous input

	model is determined by estimating the input variable probabilities.	variables and requires minimal training data to build the model.
K-Nearest Neighbors	K-Nearest Neighbors (K-NN) is a classification model that determines the class of a new data point based on the classes of the nearest data points. It calculates the distance between the new data point and known data points, selects the K nearest points, and uses a voting method.	K-NN has many advantages: high accuracy if K is chosen appropriately, no assumptions about data distribution, ease of use and understanding, noise and outlier handling, and multi-class classification capability.
Neural Networks	Neural Networks is a machine learning model inspired by the way the human brain works. It is built from multiple layers and each layer contains many neurons.	They learn automatically, process non-linear/multi-class data, handle large datasets, and solve a wide range of problems. In the study, they were chosen for flexibility with complex data and solving revenue forecasting.
LightGBM	LightGBM Classifier is one of the Ensemble Method models that uses Gradient Boosting Decision Tree (GBDT) to create a better predictive model. Ensemble Method is a machine learning technique that combines multiple models to create a better predictive model	It's faster and more efficient than other Ensemble Method models, can handle large datasets, and has high accuracy. In business profit forecasting, it was chosen over other models like XGBoost or AdaBoost due to its faster training speed and efficiency with large datasets, saving time and increasing profit prediction accuracy.

Table 4: List of model evaluation methods

Name	**Explanation**
Accuracy	Accuracy in Classification report is the ratio of correct predictions to total predictions, used to measure the model's ability to predict the correct class of new samples. It's a simple and commonly used metric, but not suitable for imbalanced datasets.
AUROC	AUROC is the area under the ROC curve, used to evaluate classification model performance. It ranges from 0 to 1, with values closer to 1 indicating

	better performance. It's commonly used in classification models, especially in medicine.
AUC-PR	AUC-PR is the area under the precision-recall curve, used to evaluate the performance of binary classification models. It ranges from 0 to 1, with values closer to 1 indicating better performance.

Based on three test factors Accuracy, AUROC, and AUC-PR, the algorithm that works best in this dataset and is most suitable to be used as the first algorithm for the proposed Hybrid ML models was identified.

Table 5: Machine Learning models' performance

Algorithm	**Time**	**Accuracy**
Logistic Regression	0.0130s	0.6692
Random Forest	13.7094s	0.6698
Naive Bayes	0.0040s	0.6242
K-Nearest Neighbors	0.0050s	0.6445
Neural Networks	3.0680s	0.6702
LightGBM	1.4525s	0.6809

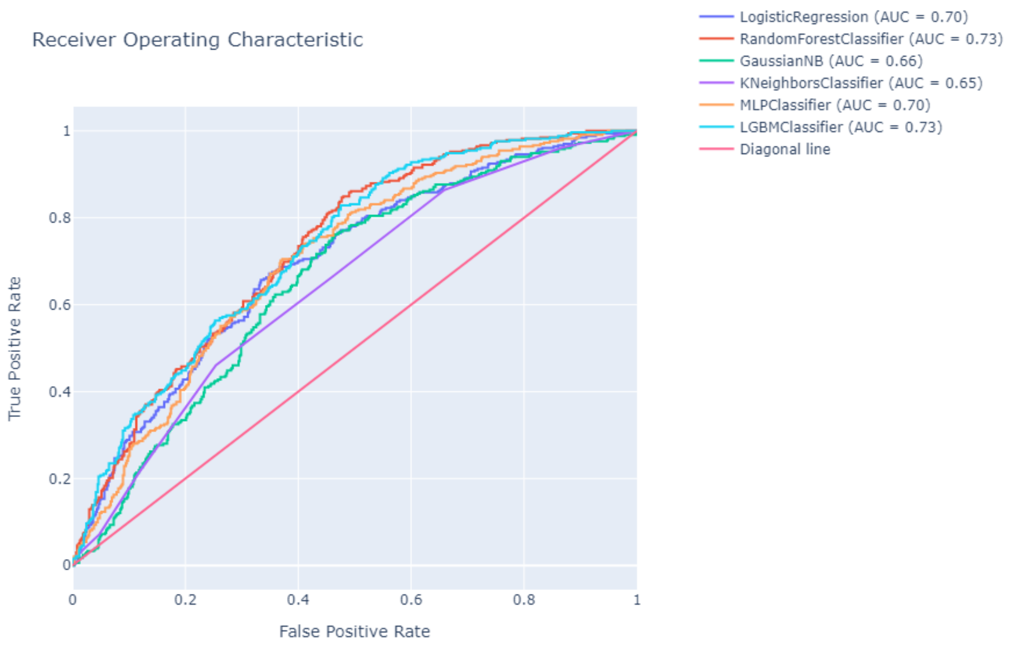


Figure 7: ROC curve chart of models

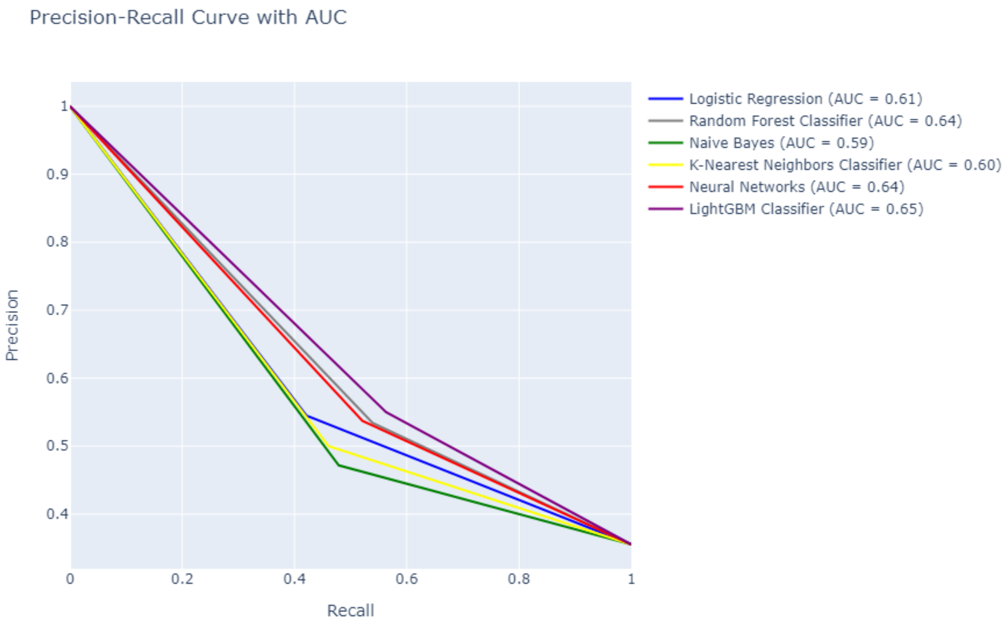


Figure 8: AUC-PR chart of models

Each factor will be considered for each single model. Starting with the Accuracy metric, it is easy to see that LightGBM is the algorithm that produces the highest accuracy of 68.09%, followed by Neural Networks with 67.02%, while Naive Bayes is the model with the lowest performance with only 62.42%. Next, I will use the AUROC metric to evaluate the models as in the ROC curve. The results show that LightGBM and Random Forest are the two algorithms with the best results, both with a score of 0.73, while K-Nearest Neighbors is the algorithm with the lowest score of 0.65. Finally, when considering AUC-PR, once again, LightGBM ranks the highest with a score of 0.65, tied for second place are the Random Forest and Neural Networks algorithms, and the poorest algorithm is Neural Networks with only 0.59. After measuring the performance of each single model, it is easy to see that the model using the LightGBM algorithm is the best performing model, and the model using the Naive Bayes algorithm is the lowest performing model. Therefore, LightGBM was chosen as the sole optimized baseline model to be combined with the other models to find the best performing Hybrid ML model.

4.4. STEP 4: Analyze the best single Machine Learning model.

Table 6: Classification report of the model using LightGBM algorithm

	Precision	Recall	F1-score	Support
0	0.76	0.75	0.75	602
1	0.55	0.56	0.56	332
Accuracy			0.68	934
Macro avg	0.65	0.65	0.65	934
Weighted avg	0.65	0.68	0.68	934

Sure, let's take a look at the Classification report of the selected model. Let's focus on label 0, which is the target label in this study as it represents the businesses that can generate profit after the next two years.

Based on the Classification report, the Precision, Recall, and F1-score metrics evaluate the model's ability to predict each specific class. Precision indicates the ratio of correct predictions to the total number of predictions made for a class, while Recall indicates the ratio of correct predictions for a class to the total number of actual instances of that class. For label 0, the model has a Precision of 0.76, meaning that 76% of the businesses predicted to have profit potential are correct. At this label, the Recall rate is 0.75, meaning that the model is correct with 75% of the businesses with profit potential. For label 1, the model has a Precision of 0.55, meaning that only 55% of the businesses predicted to have no profit potential are correct. The Recall rate is 0.56, meaning that the model is correct with 56% of the businesses with no profit potential. Additionally, the F1-score value, which combines Precision and Recall, is 0.75, indicating a relatively good balance between Precision and Recall.

In summary, the model has relatively good performance on label 0, which is the target label. The model has a fairly decent coverage and accuracy in predicting which businesses can generate profit after the next two years. However, the weakness of the model is the classification performance on label 1, which is quite low, and with such a low rate, the model cannot be used to predict businesses that are unlikely to generate profit in the next two years. However, as I mentioned earlier, the goal is label 0, which corresponds to businesses with profit potential, so this is entirely acceptable.

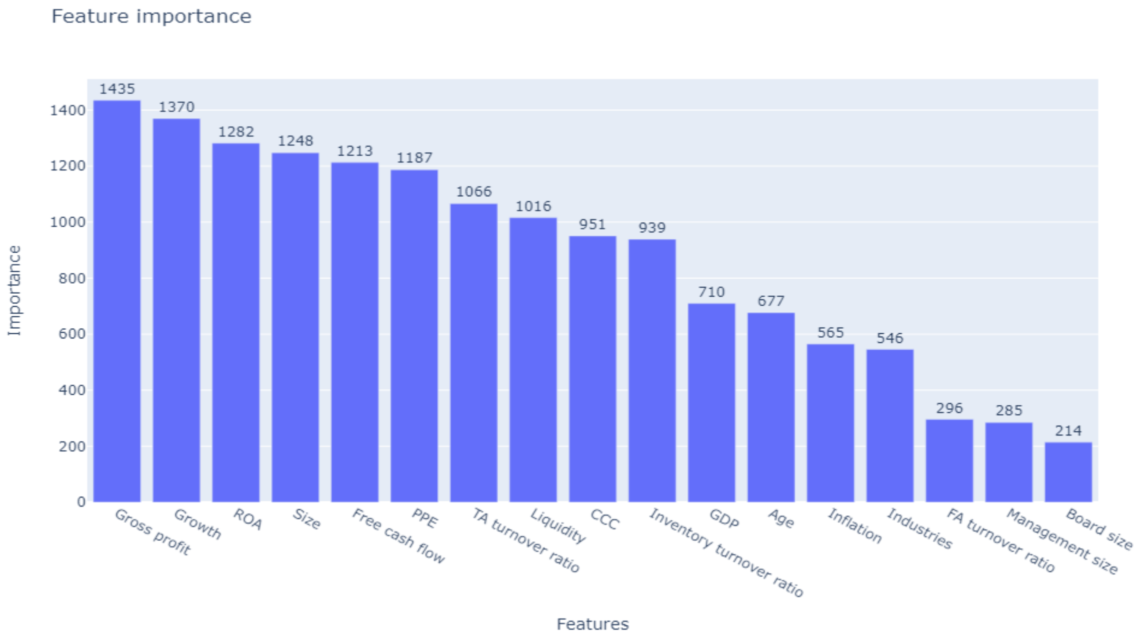


Figure 9: The features importance of the model using LightGBM algorithm

The Feature Importance chart reveals that the six most important factors in the LightGBM model for predicting a business's profit potential in Vietnam are Gross profit, Growth, ROA, Size, Free cash flow, and PPE. Three factors, FA turnover ratio, Management size, and Board size, have a low impact on the model, indicating that the number of board members or management members has very little effect on a business's profit potential. Meanwhile, four factors, GDP, Age, Inflation, and Industries, although not having as much impact as usual financial indicators, still have a quiet impact on the model, suggesting that factors such as establishment time, macroeconomic indicators, and industry can significantly affect a business's profit potential in the future.

4.5. STEP 5: Build Hybrid Machine Learning models.

Hybrid Machine Learning Ensemble Learning combines multiple models to create a more accurate and stable predictive model. In Hybrid Machine Learning Ensemble Learning there are many operating mechanisms that combine models such as voting, stacking or blending. In this study, the voting method is used. This method works by using the majority of votes to decide the final prediction. In the voting method, there are two sub-methods, hard-voting and soft-voting. Hard-voting is used in this study and how it works is to make the final prediction by counting the number of predictions of each model and choosing the most predicted prediction. It eliminates biases and noise, increases generalizability, reduces overfitting, and improves reliability.

In this step, when the best algorithm for the model has been determined to be LightGBM, five combined machine learning models will be developed by combining the LightGBM algorithm with other single machine learning models.

Table 7: Hybrid Machine Learning models' performance

Model	Accuracy
LightGBM Classifier + LogisticRegression	0.676
LightGBM Classifier + Random Forest Classifier	0.678
LightGBM Classifier + Naive Bayes	0.678
LightGBM Classifier + K-Nearest Neighbors Classifier	0.682
LightGBM Classifier + Neural Networks	0.694

From the table, we can see that the model combining the LightGBM algorithm and Neural Networks has the best performance with an accuracy of 69.4%. Compared to the initial performance of the LightGBM algorithm, the accuracy of the combined model has been improved from 68.1% to 69.4%. Although the increase is not significant, this combined model has indeed helped to improve the predictive performance.

4.6. STEP 6: Analyze the best Hybrid Machine Learning model.

Table 8: Classification report of selected Hybrid Machine Learning model

	Precision	Recall	F1-score	Support
0	0.73	0.84	0.78	602
1	0.60	0.43	0.50	332
Accuracy			0.69	934
Macro avg	0.66	0.63	0.64	934
Weighted avg	0.68	0.69	0.68	934

Looking at the Classification report of the hybrid model, the Precision for label 0 has decreased slightly, but there is a significant increase in Recall for label 0 from 0.75 to 0.84. This means that this hybrid model has significantly increased the percentage of identifying businesses with profit potential after the next two years by 9%, increasing the coverage of the model to 84% for businesses in label 0. However, the Recall for label 1 has decreased significantly, but the Precision has increased. Since my goal is label 0, this trade-off can be acceptable to improve the performance of predicting businesses in label 0. Overall, the model has a relatively good quality when considering the prediction purpose.

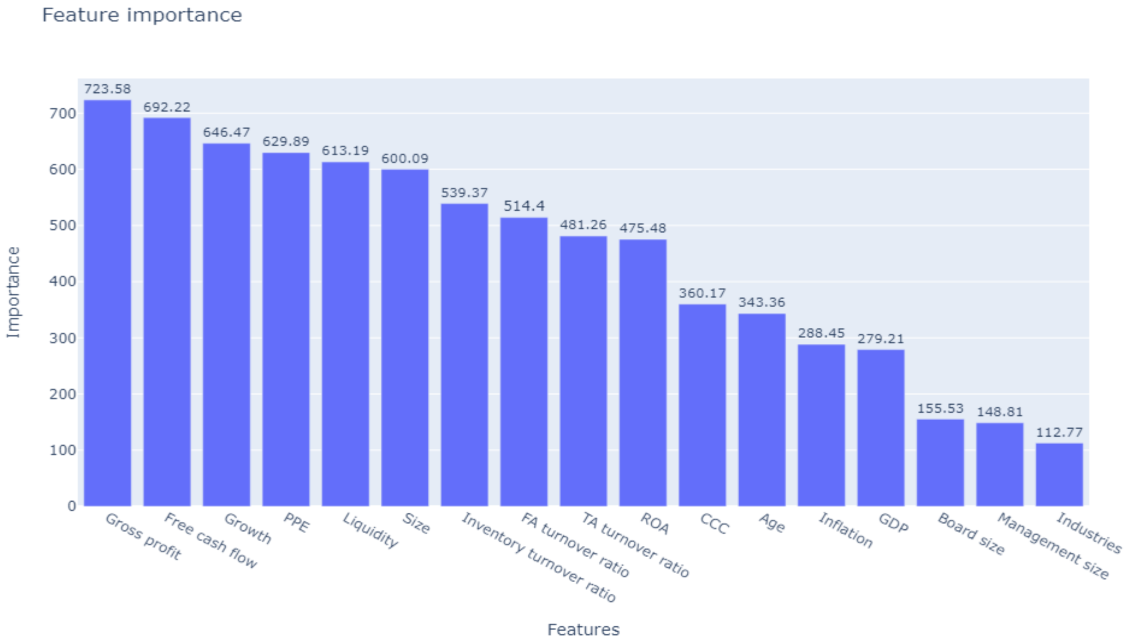


Figure 10: The features importance of selected Hybrid Machine Learning model

Based on the Feature Importance chart, the 6 most important factors in the model can be identified: Gross Profit, Free Cash Flow, Growth, PPE, Liquidity, and Firm Size. These factors strongly affect the profitability of enterprises in Vietnam in the next 2 years. Board size, Management size and Industry do not affect profitability much, indicating that the number of these members in businesses does not have much impact on profitability. Age, Inflation and GDP, although not as influential as basic financial indicators, still have a certain influence. This shows that the time of establishment and macroeconomic indicators are important to assess the profit potential of the business in the future.

Combining the two Feature Importance charts, it can be concluded that the five factors that have the most impact on profit potential are Gross profit, Growth, Free Cash Flow, PPE, and Size. On the other hand, Board size and Management size have little to no significant impact on future profit potential. It should be noted that there is a considerable difference in the ranking position of the FA turnover ratio and Industries variables, but based on their ranking scores, I would evaluate the FA turnover ratio as having a moderate impact and Industries as having a low impact. The remaining factors are classified into the moderate impact group.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

Predicting profitability of non-financial enterprises in Vietnam using Hybrid Machine learning Methodology is a new topic that I made due to current demand. The Vietnamese enterprises that I chose are those listed on two stock exchanges, HOSE and HNX. A dataset of Vietnamese enterprises on two stock exchanges from 2010 to 2022 was used to measure profitability using 21 variables, which were reduced to 17 after processing. First, compared to traditional methods, ML has many strengths in forecasting including the ability to handle big data, flexibility and diversity, automation, and more accurate forecasting capabilities. With such advantages and diversity, I selected six ML algorithms based on their performance difference to participate in this study. Six single ML models were developed using different algorithms, including Logistic Regression, Random Forest, Naive Bayes, K-Nearest Neighbors, Neural Networks, and LightGBM. The LightGBM algorithm produced the best results, with an accuracy of 68.09% and an F1-score of 75% for profitable businesses. It also achieved the highest AUROC and AUC-PR scores among the models. Overall, this model had good predictive performance for profitable businesses.

The Hybrid ML model was developed based on a combination of the best performing model, which used the LightGBM algorithm, and the other single ML models. Five Hybrid ML models were developed, including LightGBM + Logistic Regression, LightGBM + Random Forest, LightGBM + Naive Bayes, LightGBM + K-Nearest Neighbors, and LightGBM + Neural Networks. The results showed that the Hybrid ML model, which combined the LightGBM and Neural Networks models, had the best performance with an accuracy of 69.4%. In terms of predicting businesses with profit potential, this Hybrid model increased the Recall score by 9% to 0.84. This indicates that the model is capable of identifying a large majority of businesses that will generate profits in the future, but there is a trade-off in terms of accuracy, which decreased by 3% compared to the model using the LightGBM algorithm. However, overall, the F1-score of the Hybrid ML model for predicting businesses with profit potential was still increased compared to the original model. Therefore, this trade-off can be accepted.

In addition, my method also drew conclusions about the important factors that influence the prediction results. The five most influential attributes for profit potential were Gross profit, Growth, Free Cash Flow, PPE, and Size. On the other hand, Industries, Board size, and Management size had little to no significant impact on future profit potential. The remaining attributes were classified as having a moderate impact.

5.2. Recommendations

5.2.1. Realistic application

As stated at the beginning of the research article, the practical application of the results will bring many benefits to both investors and business managers.

The information from the research model supports investors in evaluating a company's potential and determining whether it is worth investing in. By increasing the accuracy of predicting which businesses have profit potential, investors can invest their money wisely, increase their chances of earning profits from their investments, and reduce risks. Additionally, this method significantly reduces the time compared to using traditional analysis methods.

Predicting a company's profitability in the future provides many benefits to business managers. The information from the predictive model helps managers make smarter investment decisions, reduce financial risks, and develop appropriate business strategies based on their company's operating conditions. Moreover, using the predictive model increases transparency and reliability for the business since decisions made based on the model's information can be explained and supported by statistics and data. Therefore, predicting a company's profitability will help managers make more informed and effective decisions in managing their business.

5.2.2. Weaknesses of the research

Although the research has yielded positive results in identifying businesses that will generate profits in the future, there are still some weaknesses.

The first weakness is that my model contains some variables that do not contribute much to enhancing the ability to predict which businesses will be profitable in the next two years.

When an ML model contains too many irrelevant variables, it can cause issues such as increasing the complexity of the model, making it difficult to understand and explain. In addition, data processing and model training can take more time and resources. Moreover, irrelevant variables can also cause noise and reduce the accuracy of the forecast model. However, this is necessary because we need to keep them in the dataset to determine which factors are important and which are not after developing the model.

The second weakness is that my models (including both single ML models and Hybrid ML models) have issues and are unable to predict businesses that are not profitable or are incurring losses after two years. Both accuracy and coverage are not sufficient. This may be due to the input variables not having a significant impact on a business's ability to generate profits.

5.2.3. Propose solutions for future researches.

To overcome the first weakness for future researches, less important variables need to be removed from the dataset. In addition, feature selection is more rigorous, regularization, consulting with domain experts, and using an iterative approach to identify and remove irrelevant variables over time. By implementing these recommendations, the model can be improved by reducing its complexity, increasing its accuracy and interpretability, and making better use of resources.

To overcome the second weakness for future researches, it is recommended to find additional variables that have a strong impact on a business's ability to be profitable or incur losses. Future studies need to collect more data, use feature engineering techniques, consult with field experts to find the most relevant variables, evaluate different models and regularly update the model. By implementing these recommendations, the model's accuracy and coverage can be improved, leading to better predictions of unprofitable or loss-incurring businesses.

6. APPENDIX

```
# Import libraries
import pandas as pd
pd.set_option('display.max_columns', None)
import numpy as np
import matplotlib.pyplot as plt
import time
import plotly.graph_objs as go
import plotly.subplots as sp
from plotly.subplots import make_subplots
import plotly.graph_objs as go
from plotly.offline import iplot
import plotly.offline as plot
import plotly.io as pio
import plotly.figure_factory as ff
import seaborn as sns
sns.set_style("whitegrid") # Set the style
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
import lightgbm as ltb
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import roc_curve, auc, precision_recall_curve, roc_auc_score
from sklearn.metrics import mean_squared_error, accuracy_score
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import KFold, GridSearchCV
from feature_engine.selection import DropCorrelatedFeatures, SelectBySingleFeaturePerformance
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.ensemble import VotingClassifier
from sklearn import model_selection
from imblearn.under_sampling import TomekLinks
from sklearn.svm import OneClassSVM
import warnings
warnings.filterwarnings('ignore')


# Read data
# Corporate finance variables
name = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Name", index_col=0)
public_date = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Public Date", index_col=0,
parse_dates=True)
ni_22 = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="NI", index_col=0)
ocf_22 = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="OCF", index_col=0)
roa = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="ROA", index_col=0)
ta = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="TA", index_col=0)
mv = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="MV", index_col=0)
total_lia = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Total Lia", index_col=0)
sales = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Sales", index_col=0)
tangible_FA = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Tangible FA", index_col=0)
total_ca = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Total CA", index_col=0)
current_lia = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Total Current Lia", index_col=0)
fixed_assets = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Fixed assets", index_col=0)
accounts_receivable = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Accounts Receivable",
index_col=0)
costs_of_revenue = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Costs of revenue", index_col=0)
ebit = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="EBIT", index_col=0)
eps = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="EPS", index_col=0)
equity = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Equity", index_col=0)
avg_receivable_days = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Avg. Receivable days",
index_col=0)
avg_payable_days = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Avg. Payable days", index_col=0)
```

```
avg_inventory_days = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Avg. Inventory days",
index_col=0)
total_revenue = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Total Revenue", index_col=0)
cash_dividend_paid = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Cash Dividend Paid",
index_col=0)
capital_expenditures = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Capital Expenditures",
index_col=0)
# Macro variables
inflation = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Inflation", index_col=0)
gdp = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="GDP", index_col=0)
# Personnel variables
board_size = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Board Size", index_col=0)
management_size = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Management Size", index_col=0)
# Industry variable
industries = pd.read_excel(r'D:\Courses_School\4th_Year_Se2\KLTN\Data\dataset.xlsx', sheet_name="Industries", index_col=0)


# Delete businesses established after 2016
new_found_companies = [i for i in name.index if int(public_date.loc[i,'Date Became Public'].strftime('%Y')) > 2016]
print(len(new_found_companies), 'companies are established under 4 years')
name.drop(new_found_companies,axis='index', inplace=True)
ni_22.drop(new_found_companies,axis='columns', inplace=True)
ocf_22.drop(new_found_companies,axis='columns', inplace=True)
roa.drop(new_found_companies,axis='columns', inplace=True)
ta.drop(new_found_companies,axis='columns', inplace=True)
mv.drop(new_found_companies,axis='columns', inplace=True)
total_lia.drop(new_found_companies,axis='columns', inplace=True)
sales.drop(new_found_companies,axis='columns', inplace=True)
tangible_FA.drop(new_found_companies,axis='columns', inplace=True)
total_ca.drop(new_found_companies,axis='columns', inplace=True)
current_lia.drop(new_found_companies,axis='columns', inplace=True)
fixed_assets.drop(new_found_companies,axis='columns', inplace=True)
accounts_receivable.drop(new_found_companies,axis='columns', inplace=True)
costs_of_revenue.drop(new_found_companies,axis='columns', inplace=True)
ebit.drop(new_found_companies,axis='columns', inplace=True)
eps.drop(new_found_companies,axis='columns', inplace=True)
equity.drop(new_found_companies,axis='columns', inplace=True)
avg_receivable_days.drop(new_found_companies,axis='columns', inplace=True)
avg_payable_days.drop(new_found_companies,axis='columns', inplace=True)
avg_inventory_days.drop(new_found_companies,axis='columns', inplace=True)
total_revenue.drop(new_found_companies,axis='columns', inplace=True)
cash_dividend_paid.drop(new_found_companies,axis='columns', inplace=True)
capital_expenditures.drop(new_found_companies,axis='columns', inplace=True)
inflation.drop(new_found_companies,axis='columns', inplace=True)
gdp.drop(new_found_companies,axis='columns', inplace=True)
board_size.drop(new_found_companies,axis='columns', inplace=True)
management_size.drop(new_found_companies,axis='columns', inplace=True)
industries.drop(new_found_companies,axis='columns', inplace=True)
public_date = public_date.loc[name.index, :]
print(len(name), 'companies remaining')


# Check % missing value
list_inputs = [ni_22, ocf_22, roa, ta, mv, total_lia, sales, tangible_FA, total_ca, current_lia, fixed_assets, accounts_receivable, costs_of_revenue,
ebit, eps, equity, \
                avg_receivable_days, avg_payable_days, avg_inventory_days, total_revenue, cash_dividend_paid, capital_expenditures, inflation, gdp,
board_size, management_size, industries]
for i in list_inputs:
    missing_values = i.isnull().sum().sum()
    num_rows, num_cols = i.shape
    percentage_missing = missing_values / (num_rows * num_cols)
    print(round(percentage_missing,2))
print('--------------')
for i in list_inputs:
    i = i.interpolate(method='linear', limit_direction='both', axis=0)
    missing_values = i.isnull().sum().sum()
    num_rows, num_cols = i.shape
    percentage_missing = missing_values / (num_rows * num_cols)
    print(round(percentage_missing,2))


# Transform the values in the following columns to positive: Cost of revenue, A.R, Avg.Inv days, Avg. Receivable days, and Avg. Payable days.
```

```
    print(round(percentage_missing,2))


# Transform the values in the following columns to positive: Cost of revenue, A.R, Avg.Inv days, Avg. Receivable days, and Avg. Payable days.
def fix_negative(data):
  data[data < 0] = data.abs()
fix_negative(costs_of_revenue)
fix_negative(accounts_receivable)
fix_negative(avg_inventory_days)
fix_negative(avg_receivable_days)
fix_negative(avg_payable_days)


# Calculate variables
# Calculate NI & OCF and drop 2 year 2021, 2022 in NI & OCF
ni = ni_22.iloc[:-2]
ocf = ocf_22.iloc[:-2]
# Calculate Gross Profit Margin
gross_profit = (total_revenue - costs_of_revenue) / total_revenue
# Calculate Free cash flow
free_cash_flow = ocf - capital_expenditures
# Calculate Liquidity
liq = total_ca / current_lia
# Calculate Growth
growth = sales.pct_change(periods=1)
growth.drop(growth.index[0], axis='index', inplace=True)
# Calculate Inv turnover ratio
inv_turnover = 365 / avg_inventory_days
# Calculate Inv
inv = sales/inv_turnover
# Calculate Quick ratio
quick = (total_ca - inv) / current_lia
# Calculate FA turnover ratio
fa_turnover = sales / fixed_assets
# Calculate TA turnover ratio
ta_turnover = sales / ta
# Calculate Size
size = np.log(mv)
# Calculate Leverage
lev = total_lia / ta
# Calculate PPE
ppe = tangible_FA / sales
# Calculate DSO
dso = accounts_receivable / (sales / 365)
# Calculate Capital intensity
capital_intensity = ta / sales
# Calculate Expense revenue ratio
expense_revenue = costs_of_revenue / sales
# Calculate ROE
roe = ni / equity
# Calculate growth of CCC
ccc = avg_inventory_days + avg_receivable_days - avg_payable_days
# Calculate Operating margin
operating_margin = ebit / sales
# Calculate Net profit margin
net_profit_margin = ni / sales


# Drop year 2009
df_list = [ni, ni_22, ocf, ocf_22, size, lev, ppe, liq, inv_turnover, quick, fa_turnover, ta_turnover, dso,
capital_intensity, expense_revenue, operating_margin, net_profit_margin, roe, roa,
eps, ccc, gross_profit, free_cash_flow, inflation, gdp, cpi, interest_rates, board_size, management_size, industries]
for i in df_list:
    i.drop(i.index[0], axis='index', inplace=True)


# Calculate Age
ind = list(range(2010, 2021))
col = size.columns
age = pd.DataFrame(columns=size.columns)
for i in ind:
  list_year = []
```

```
 for j in size.columns:
  num = i - public_date.loc[j,'Organization Founded Year']
  list_year.append(num)
 age_length = len(age)
 age.loc[age_length] = list_year
age.index = size.index
age = age.replace(list(range(-5,0)), np.NaN)


# Calculate Target
target = pd.DataFrame(np.random.randn(13, len(name)), columns=size.columns, index = ni_22.index)
for i in range(len(ni_22.index)):
 for j in range(len(ni_22.columns)):
  if ni_22.iloc[i,j] > 0 and ocf_22.iloc[i,j] > 0:
   target.iloc[i,j] = 0
  else:
   target.iloc[i,j] = 1
print(target.to_string())


# Create a 2 year delay for Target
target = target.shift(-2)
target = target.dropna()
print(target.to_string())


# Create complete dataset
column_names = ['Gross profit', 'Free cash flow', 'Growth', 'PPE', 'Liquidity', 'Size', 'Leverage', 'Operating margin',
        'Net profit margin', 'Quick ratio', 'Inventory turnover ratio', 'FA turnover ratio', 'TA turnover ratio',
        'ROA', 'ROE', 'EPS', 'CCC', 'DSO', 'Capital intensity', 'Expense of revenue ratio', 'Age', 'Inflation', 'GDP',
        'CPI', 'Interest rates', 'Board size', 'Management size', 'Industries', 'Target']
variables = [gross_profit, free_cash_flow, growth, ppe, liq, size, lev, operating_margin, net_profit_margin, quick,
      inv_turnover, fa_turnover, ta_turnover, roa, roe, eps, ccc, dso, capital_intensity, expense_revenue,
      age, inflation, gdp, cpi, interest_rates, board_size, management_size, industries, target]
data = pd.DataFrame({col_name: var.values.flatten() for col_name, var in zip(column_names, variables)})
# Set up multi index
year = np.arange(2010,2021)
com = list(size.columns)
index = pd.MultiIndex.from_product([year, com],
             names=['Year', 'Company'])
data.index = index
data = data.reset_index(drop=False)
data['Index data'] = data[['Year', 'Company']].apply(lambda x: f"{x['Year']} : {x['Company']}", axis=1)
# Đặt cột 'new_col' làm index
data.set_index('Index data', inplace=True)
# Delete col1 và col2
data.drop(['Year', 'Company'], axis=1, inplace=True)
data


# Check missing values for each column
# Calculate the percentage of missing values for each column
missing_values = data.isnull().sum() / len(data)
missing_values = missing_values.sort_values(ascending=False)

# Create trace for bar chart
trace = go.Bar(
  x=missing_values.index,
  y=missing_values.values,
  text=missing_values.values.round(2),
  textposition='auto',
  hovertext=missing_values.values.round(2),
  hoverinfo='text'
)
# Create layout for the chart
layout = go.Layout(
  title='Graph missing values for each variable',
  xaxis=dict(title='Variables'),
  yaxis=dict(title='Missing value ratio'),
  width=1000
)
# Make figures and draw charts
```

```
fig = go.Figure(data=[trace], layout=layout)
pio.show(fig)


# Drop rows that have NaN
new_data = data.dropna(axis=0)
new_data = new_data.astype({'Age': int})
print(new_data.isnull().sum().sum())


# Create Histograms for each column
list_variables = new_data.columns.to_list()
list_variables_no_target = list_variables[:-1]
def histogram(data, variables):
    n_rows = len(variables) // 5 + (len(variables) % 5 > 0)
    fig = make_subplots(rows=n_rows, cols=5, subplot_titles=tuple(variables), vertical_spacing=0.05, horizontal_spacing=0.05)
    row, col = 1, 1
    for i in variables:
        fig.add_trace(go.Histogram(x=data[i], name=i), row, col)
        col += 1
        if col > 5:
            col = 1
            row += 1
    fig.update_layout(height=n_rows*250, width=1500, title='Density Plots', showlegend=False)
    # Edit chart names
    for i, name in enumerate(variables):
        fig['layout']['annotations'][i]['text'] = name
    pio.show(fig)
histogram(new_data, list_variables_no_target)

# Handling Outliers
list_variables_outlier = [
    'Growth',
    'PPE',
    'Liquidity',
    'Quick ratio',
    'Inventory turnover ratio',
    'FA turnover ratio',
    'DSO',
    'Capital intensity',
    'Expense of revenue ratio',
    'Operating margin',
    'Net profit margin',
    'ROE',
    'CCC',
    'Gross profit',
    'Free cash flow'
]
def replace_with_ocsvm(df):
    for col in list_variables_outlier:
        clf = OneClassSVM(nu=0.04)
        clf.fit(df[col].values.reshape(-1, 1))
        y_pred = clf.predict(df[col].values.reshape(-1, 1))
        upper_limit = df.loc[y_pred == 1, col].max()
        lower_limit = df.loc[y_pred == 1, col].min()
        df[col] = np.where(df[col] > upper_limit, upper_limit, df[col])
        df[col] = np.where(df[col] < lower_limit, lower_limit, df[col])
    return df
replace_with_ocsvm(new_data)
# Descriptive statistics
new_data.describe()


# Re-visualize Histogram in each column.
histogram(new_data, list_variables_no_target)


# Initialize DropCorrelatedFeatures with threshold of 0.5
correlated_selector = DropCorrelatedFeatures(threshold=0.6)

# Run fit_transform method to remove highly correlated features
new_data = correlated_selector.fit_transform(new_data)
```

```
# Create Corralation Matrix
corr_matrix = new_data.iloc[:, :-1].corr()
# Create heatmap object
heatmap = go.Figure(data=go.Heatmap(z=corr_matrix.values,
                                    x=corr_matrix.columns.values,
                                    y=corr_matrix.index.values,
                                    colorscale='Blues'))
# Configure the layout for the chart
heatmap.update_layout(title='Correlation Matrix Heatmap', width=1000, height=600)
pio.show(heatmap)


# Corralation between features and target
corr_matrix_target = new_data.corr()
corr_with_target = corr_matrix_target['Target'].drop('Target')
# Reduce the number to 3 decimal places
corr_with_target = corr_with_target.round(3)
# Sort correlation with target in descending order
corr_with_target_sorted = corr_with_target.sort_values(ascending=False)
# Create a Bar object
bar = go.Bar(
    x=corr_with_target_sorted.index.values,
    y=corr_with_target_sorted.values,
    marker=dict(
        color=corr_with_target_sorted.values,
        colorscale='Viridis',
        reversescale=True
    ),
    text=corr_with_target_sorted.values,
    textposition='outside',
    texttemplate='%{text:.3f}'
)
# Create a layout
layout = go.Layout(
    title='Correlation between features and target',
    xaxis_title='Features',
    yaxis_title='Correlation coefficients',
    width=1000,
    height=600
)

# Create a Figure object
fig = go.Figure(data=[bar], layout=layout)
# Show the plot
pio.show(fig)

# Create a Features importance chart
# Initialize SelectBySingleFeaturePerformance with model RandomForestRegressor
selector = SelectBySingleFeaturePerformance(
    estimator = RandomForestClassifier(n_estimators=10, max_depth=2, random_state=1), # the model
    scoring="roc_auc", # the metric to determine model performance
    cv=3, # the cross-validation fold,
    threshold=None, # the performance threshold
)
# Run the fit method to calculate the performance of each feature
selector.fit(new_data.iloc[:, :-1], new_data['Target'])
# the univariate performance of the features
pfm = pd.Series(selector.feature_performance_)
pfm = pfm.round(2)
# Sort columns in descending order
pfm_sorted = pfm.sort_values(ascending=False)
# Create Plotly's Bar object
trace = go.Bar(x=pfm_sorted.index, y=pfm_sorted.values, text=pfm_sorted.values, textposition='outside')
# Create layout for chart
layout = go.Layout(
    title='Feature performance',
    xaxis=dict(title='Features'),
    yaxis=dict(title='roc-auc'),
    width=1000, height=600
)
```

```
fig = go.Figure(data=[trace], layout=layout)
fig.show()


# Create a chart that checks the number of labels in the Target variable
value_counts = new_data['Target'].value_counts()
trace = go.Bar(x=value_counts.index, y=value_counts.values)
layout = go.Layout(title=f"Value Counts of Target", width=1000)
fig = go.Figure(data=[trace], layout=layout)
# Add the number of occurrences of each value to the chart
for i, val in enumerate(value_counts.values):
    fig.add_annotation(
        x=value_counts.index[i],
        y=val,
        text=str(val),
        font=dict(size=14),
        showarrow=False,
        yshift=10
    )
pio.show(fig)
# We can easily see that the Dataset is imbalanced


#BUILD MODEL

# Split train-test sets
X = new_data.drop('Target', axis=1)
y = new_data['Target']
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=6)


# Handle imbalance dataset
# Check the number of 0 and 1 observations in the training df
print("Before UnderSampling, counts of label '1': {}".format(sum(y_train == 1)))
print("Before UnderSampling, counts of label '0': {} \n".format(sum(y_train == 0)))

# Tomek Links to deal with data imbalances
tl = TomekLinks()
X_train, y_train = tl.fit_resample(X_train, y_train)
print('After UnderSampling, the shape of train_X: {}'.format(X_train.shape))
print('After UnderSampling, the shape of train_y: {} \n'.format(y_train.shape))
print("After UnderSampling, counts of label '1': {}".format(sum(y_train == 1)))
print("After UnderSampling, counts of label '0': {}".format(sum(y_train == 0)))


# Scale X
scaler = StandardScaler()
scaler.fit(X)
X_train = pd.DataFrame(scaler.transform(X_train), index=X_train.index, columns=X_train.columns)
X_test = pd.DataFrame(scaler.transform(X_test), index=X_test.index, columns=X_test.columns)


# Since this step uses GridSearch and K-ford to find the best parameters for the models, it will be very time consuming.
# The results have been printed by me below this cell and the next cell is also calculated by me using the results. So it is not necessary to run this cell again
# -----------------------------------------------------------------------------------------------------------------------------
# # Initialize models
# models = {
#     'Logistic Regression': LogisticRegression(random_state=42),
#     'Random Forest': RandomForestClassifier(random_state=42),
#     'Naive Bayes': GaussianNB(),
#     'K-Nearest Neighbors': KNeighborsClassifier(),
#     'Neural Network': MLPClassifier(random_state=42),
#     'LightGBM Classifier': LGBMClassifier(random_state=42)
# }
# # Use K-fold cross validation to evaluate the model and reduce overfitting
# kf = KFold(n_splits=5, shuffle=True, random_state=42)
# for model_name, model in models.items():
#     # Tinh chỉnh hyperparameters
#     param_grid = {}
#     if model_name == 'Logistic Regression':
```

```
#         param_grid = {'C': [0.1, 1, 10]}
#     elif model_name == 'Random Forest':
#         param_grid = {'n_estimators': [100, 500, 1000], 'max_depth': [None, 5, 10], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4]}
#     elif model_name == 'Naive Bayes':
#         param_grid = {'var_smoothing': [1e-09, 1e-08, 1e-07]}
#     elif model_name == 'K-Nearest Neighbors':
#         param_grid = {'n_neighbors': [3, 5, 7], 'weights': ['uniform', 'distance'], 'algorithm': ['ball_tree', 'kd_tree', 'brute']}
#     elif model_name == 'Neural Network':
#         param_grid = {'hidden_layer_sizes': [(10,), (50,), (100,)], 'activation': ['relu', 'tanh', 'logistic'], 'alpha': [0.0001, 0.001, 0.01]}
#     elif model_name == 'LightGBM Classifier':
#         param_grid = {'n_estimators': [100, 500, 1000], 'max_depth': [None, 5, 10], 'learning_rate': [0.1, 0.01, 0.001]}
# # Refine the model
#     grid_search = GridSearchCV(model, param_grid, cv=kf)
#     grid_search.fit(X_train, y_train)
#     # Print out the best hyperparameters
#     print(model_name, "Best parameters:", grid_search.best_params_)
#     # Evaluate model on test set
#     y_pred = grid_search.predict(X_test)
#     acc = accuracy_score(y_test, y_pred)
#     print(model_name + ' accuracy:', acc)
# ------------------------------------------------------------------------------------------------------------------------------------
# RESULTS
# Logistic Regression Best parameters: {'C': 10}
# Logistic Regression accuracy: 0.6691648822269807
# Random Forest Best parameters: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 1000}
# Random Forest accuracy: 0.6691648822269807
# Naive Bayes Best parameters: {'var_smoothing': 1e-09}
# Naive Bayes accuracy: 0.6241970021413277
# K-Nearest Neighbors Best parameters: {'algorithm': 'ball_tree', 'n_neighbors': 7, 'weights': 'uniform'}
# K-Nearest Neighbors accuracy: 0.6445396145610278
# Neural Network Best parameters: {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50,)}
# Neural Network accuracy: 0.6702355460385439
# LightGBM Classifier Best parameters: {'learning_rate': 0.01, 'max_depth': None, 'n_estimators': 500}
# LightGBM Classifier accuracy: 0.6809421841541756


# Linear Model
model_LogisticRegression = LogisticRegression(C=10, random_state=42)
# Tree-Based Model
model_RandomForestClassifier = RandomForestClassifier(max_depth=None, min_samples_leaf=2, min_samples_split=10, n_estimators=1000, random_state=42)
# Naive Bayes:
model_NaiveBayes = GaussianNB(var_smoothing=1e-09)
# K-Nearest Neighbors
model_KNeighborsClassifier = KNeighborsClassifier(algorithm='ball_tree', n_neighbors=7, weights='uniform')
# Neural Networks
model_NeuralNetworks = MLPClassifier(activation='tanh', alpha=0.0001, hidden_layer_sizes=(50,), random_state=42)
# Ensemble Method
model_LGBMClassifier = ltb.LGBMClassifier(learning_rate=0.01, max_depth=None, n_estimators=500, random_state=42)


# Training the Machine Learning models
list_model_names = ['LogisticRegression', 'Random Forest Classifier', 'Naive Bayes', 'K-Nearest Neighbors Classifier', 'Neural Networks', 'LightGBM Classifier']
list_models = [model_LogisticRegression, model_RandomForestClassifier, model_NaiveBayes, model_KNeighborsClassifier, model_NeuralNetworks, model_LGBMClassifier]
scores = []
train_times = []
for name, model in zip(list_model_names, list_models):
    start_time = time.time()
    model.fit(X_train, y_train)
    end_time = time.time()
    train_time = end_time - start_time
    train_times.append(train_time)
    score_model = model.score(X_test, y_test)
    score_model = round(score_model, 4)
    scores.append(score_model)
    print(f"{name}: Train time = {train_time:.4f}s, Test score = {score_model:.4f}")


#Making the prediction
y_pred_LogisticRegression = model_LogisticRegression.predict(X_test)
```

```
y_pred_RandomForestClassifier = model_RandomForestClassifier.predict(X_test)
y_pred_NaiveBayes = model_NaiveBayes.predict(X_test)
y_pred_KNeighborsClassifier = model_KNeighborsClassifier.predict(X_test)
y_pred_NeuralNetworks = model_NeuralNetworks.predict(X_test)
y_pred_LGBMClassifier = model_LGBMClassifier.predict(X_test)


# Initialize a new figure
fig = go.Figure()
# Loop through each model in list_models
for model in list_models:
    y_test_scores = model.predict_proba(X_test)[:, 1]
    fpr, tpr, thresholds = roc_curve(y_test, y_test_scores)
    # Draw ROC and calculate AUC
    roc_auc = auc(fpr, tpr)
    fig.add_trace(go.Scatter(x=fpr, y=tpr, mode='lines', name='{} (AUC = {:.2f})'.format(type(model).__name__, roc_auc)))
# Set title and axis
fig.update_layout(title='Receiver Operating Characteristic', xaxis_title='False Positive Rate', yaxis_title='True Positive Rate')
fig.add_trace(go.Scatter(x=[0, 1], y=[0, 1], mode='lines', name='Diagonal line'))
fig.update_layout(legend=dict(yanchor="top", y=0.99, xanchor="left", x=0.01))
fig.update_layout(
    title='Receiver Operating Characteristic',
    xaxis_title='False Positive Rate',
    yaxis_title='True Positive Rate',
    legend=dict(x=1.1, y=1.05, xanchor='left', yanchor='middle'),
    width=900, height=600
)
fig.show()


# Calculate precision, recall and threshold values for each model
precision_LogisticRegression, recall_LogisticRegression, thresholds_LogisticRegression = precision_recall_curve(y_test,
y_pred_LogisticRegression)
precision_RandomForestClassifier, recall_RandomForestClassifier, thresholds_RandomForestClassifier = precision_recall_curve(y_test,
y_pred_RandomForestClassifier)
precision_NaiveBayes, recall_NaiveBayes, thresholds_NaiveBayes = precision_recall_curve(y_test, y_pred_NaiveBayes)
precision_KNeighborsClassifier, recall_KNeighborsClassifier, thresholds_KNeighborsClassifier = precision_recall_curve(y_test,
y_pred_KNeighborsClassifier)
precision_NeuralNetworks, recall_NeuralNetworks, thresholds_NeuralNetworks = precision_recall_curve(y_test, y_pred_NeuralNetworks)
precision_LGBMClassifier, recall_LGBMClassifier, thresholds_LGBMClassifier = precision_recall_curve(y_test, y_pred_LGBMClassifier)
# Calculate AUC for each model
auc_LogisticRegression = roc_auc_score(y_test, y_pred_LogisticRegression)
auc_RandomForestClassifier = roc_auc_score(y_test, y_pred_RandomForestClassifier)
auc_NaiveBayes = roc_auc_score(y_test, y_pred_NaiveBayes)
auc_KNeighborsClassifier = roc_auc_score(y_test, y_pred_KNeighborsClassifier)
auc_NeuralNetworks = roc_auc_score(y_test, y_pred_NeuralNetworks)
auc_LGBMClassifier = roc_auc_score(y_test, y_pred_LGBMClassifier)
# Create a precision-recall curve by plotly for each model and add AUC information to the title of each line
fig = go.Figure()
fig.add_trace(go.Scatter(x=recall_LogisticRegression, y=precision_LogisticRegression, mode='lines', name='Logistic Regression (AUC =
{:.2f})'.format(auc_LogisticRegression), line=dict(color='blue')))
fig.add_trace(go.Scatter(x=recall_RandomForestClassifier, y=precision_RandomForestClassifier, mode='lines', name='Random Forest Classifier
(AUC = {:.2f})'.format(auc_RandomForestClassifier), line=dict(color='gray')))
fig.add_trace(go.Scatter(x=recall_NaiveBayes, y=precision_NaiveBayes, mode='lines', name='Naive Bayes (AUC = {:.2f})'.format(auc_NaiveBayes),
line=dict(color='green')))
fig.add_trace(go.Scatter(x=recall_KNeighborsClassifier, y=precision_KNeighborsClassifier, mode='lines', name='K-Nearest Neighbors Classifier
(AUC = {:.2f})'.format(auc_KNeighborsClassifier), line=dict(color='yellow')))
fig.add_trace(go.Scatter(x=recall_NeuralNetworks, y=precision_NeuralNetworks, mode='lines', name='Neural Networks (AUC =
{:.2f})'.format(auc_NeuralNetworks), line=dict(color='red')))
fig.add_trace(go.Scatter(x=recall_LGBMClassifier, y=precision_LGBMClassifier, mode='lines', name='LightGBM Classifier (AUC =
{:.2f})'.format(auc_LGBMClassifier), line=dict(color='purple')))
# Set axes and titles for the graph
fig.update_layout(
title='Precision-Recall Curve with AUC',
xaxis_title='Recall',
yaxis_title='Precision',
width=900, height=600
)
fig.show()


# Calculate predicted value on test set
```

```
y_pred_LGBMClassifier = model_LGBMClassifier.predict(X_test)
# Calculate report
report_LGBMClassifier = classification_report(y_test, y_pred_LGBMClassifier)
print(report_LGBMClassifier)


# Get feature importances from the LGBMClassifier model
feature_importances_LGBMClassifier = model_LGBMClassifier.feature_importances_
# Get the list of features in descending order of feature importances
sorted_features = new_data.columns[np.argsort(feature_importances_LGBMClassifier)[::-1]]
# Create Plotly's Bar object and sort it in descending order
trace = go.Bar(x=sorted_features, y=feature_importances_LGBMClassifier[np.argsort(feature_importances_LGBMClassifier)[::-1]], text=feature_importances_LGBMClassifier[np.argsort(feature_importances_LGBMClassifier)[::-1]], textposition='outside')
# Create layout for the chart
layout = go.Layout(
title='Feature importance',
xaxis=dict(title='Features'),
yaxis=dict(title='Importance'),
width=1000, height=600
)
# Create Plotly's Figure object
fig = go.Figure(data=[trace], layout=layout)
fig.show()


# Create Hybrid Machine Learning models
# combine models using the Majority Voting method
ensemble_model_LogisticRegression = VotingClassifier(estimators=[('LightGBM Classifier', model_LGBMClassifier), ('LogisticRegression', model_LogisticRegression)], voting='hard')
# combination model evaluation
ensemble_model_LogisticRegression.fit(X_train, y_train)
y_pred_ensemble_LogisticRegression = ensemble_model_LogisticRegression.predict(X_test)
accuracy_ensemble_LogisticRegression = round(accuracy_score(y_test, y_pred_ensemble_LogisticRegression), 5)
print('LightGBM Classifier + LogisticRegression:', accuracy_ensemble_LogisticRegression)
ensemble_model_RandomForestClassifier = VotingClassifier(estimators=[('LightGBM Classifier', model_LGBMClassifier), ('Random Forest Classifier', model_RandomForestClassifier)], voting='hard')
# combination model evaluation
ensemble_model_RandomForestClassifier.fit(X_train, y_train)
y_pred_ensemble_RandomForestClassifier = ensemble_model_RandomForestClassifier.predict(X_test)
accuracy_ensemble_RandomForestClassifier = round(accuracy_score(y_test, y_pred_ensemble_RandomForestClassifier), 5)
print('LightGBM Classifier + Random Forest Classifier:', accuracy_ensemble_RandomForestClassifier)
ensemble_model_NaiveBayes = VotingClassifier(estimators=[('LightGBM Classifier', model_LGBMClassifier), ('Naive Bayes', model_NaiveBayes)], voting='hard')
# combination model evaluation
ensemble_model_NaiveBayes.fit(X_train, y_train)
y_pred_ensemble_NaiveBayes = ensemble_model_NaiveBayes.predict(X_test)
accuracy_ensemble_NaiveBayes = round(accuracy_score(y_test, y_pred_ensemble_NaiveBayes), 5)
print('LightGBM Classifier + Naive Bayes:', accuracy_ensemble_NaiveBayes)
ensemble_model_KNeighborsClassifier = VotingClassifier(estimators=[('LightGBM Classifier', model_LGBMClassifier), ('K-Nearest Neighbors Classifier', model_KNeighborsClassifier)], voting='hard')
# combination model evaluation
ensemble_model_KNeighborsClassifier.fit(X_train, y_train)
y_pred_ensemble_KNeighborsClassifier = ensemble_model_KNeighborsClassifier.predict(X_test)
accuracy_ensemble_KNeighborsClassifier = round(accuracy_score(y_test, y_pred_ensemble_KNeighborsClassifier), 5)
print('LightGBM Classifier + K-Nearest Neighbors Classifier:', accuracy_ensemble_KNeighborsClassifier)
ensemble_model_NeuralNetworks = VotingClassifier(estimators=[('Random Forest Classifier', model_LGBMClassifier), ('Neural Networks', model_NeuralNetworks)], voting='hard')
# combination model evaluation
ensemble_model_NeuralNetworks.fit(X_train, y_train)
y_pred_ensemble_NeuralNetworks = ensemble_model_NeuralNetworks.predict(X_test)
accuracy_ensemble_NeuralNetworks = round(accuracy_score(y_test, y_pred_ensemble_NeuralNetworks), 5)
print('LightGBM Classifier + Neural Networks:', accuracy_ensemble_NeuralNetworks)

# Generate Classification report for the best Hybrid model
report_LGBMClassifier_NeuralNetworks = classification_report(y_test, y_pred_ensemble_NeuralNetworks)
print(report_LGBMClassifier_NeuralNetworks)

# Calculate feature importance for each submodule
LGBMClassifier_feature_importance = model_LGBMClassifier.feature_importances_
NeuralNetworks_feature_importance = np.abs(model_NeuralNetworks.coefs_[0]).sum(axis=1)
# combine feature importance of each model
ensemble_feature_importance = (LGBMClassifier_feature_importance + NeuralNetworks_feature_importance) / 2
ensemble_feature_importance = np.round(ensemble_feature_importance, 2)
# show feature importance of each feature
feature_importances_LGBMClassifier_NeuralNetworks = pd.Series(ensemble_feature_importance,
index=X.columns).sort_values(ascending=False)
# Create Plotly's Bar object
trace = go.Bar(x=new_data.columns, y=feature_importances_LGBMClassifier_NeuralNetworks,
text=feature_importances_LGBMClassifier_NeuralNetworks.round(2), textposition='outside')
# Create layout for the chart
layout = go.Layout(
   title='Feature importance',
   xaxis=dict(title='Features'),
   yaxis=dict(title='Importance'),
   width=1000, height=600
)

# Create Plotly's Figure object
fig = go.Figure(data=[trace], layout=layout)
fig.show()
```

7. REFERENCES

TS. Nguyen, H, A. TS. Nguyen, T, T. (2017). Các Yếu Tố ảnh hưởng đến Lợi Nhuận doanh nghiệp niêm yết trên sàn HOSE. Tạp chí Tài chính. Retrieved April 14, 2023, from https://tapchitaichinh.vn/cac-yeu-to-anh-huong-den-loi-nhuan-doanh-nghiep-niem-yet-tren-hose.html

TS. Nguyen, A. P. (2021). Do luong hieu qua hoat dong doanh nghiep qua chi so gia tri thi truong va chi so gia tri so sach bang phuong phap may hoc. Tap chi Kinh te Chau A - Thai Binh Duong.

Thủy, H. T. T., Hiếu, Đ. V., Danh, D. K., Đạt, P. Q., Ngọc, N. B., & Ngọc, P. B. (2020). The factors affect financial performance of companies listed on Hanoi stock exchange. Tạp Chí Khoa Học Và Công Nghệ: Chuyên San Kinh Tế - Luật - Khoa Học Quản Lý. https://doi.org/10.32508/stdjelm.v4i3.663

English

Xinyue, C., Zhaoyu, X., & Yue, Z. (2020). Using Machine Learning to Forecast Future Earnings. Atlantic Economic Journal, 48(4), 543–545. https://doi.org/10.1007/s11293-020-09691-1

Anand, V., Brunner, R. J., Ikegwu, K. M., & Sougiannis, T. (2019). Predicting Profitability Using Machine Learning. Social Science Research Network. https://doi.org/10.2139/ssrn.3466478

Alarussi, A., &amp; Alhaderi, S. M. (2020). Factors affecting profitability in Malaysia. Journal of Economic Studies. Retrieved April 15, 2023, from https://www.academia.edu/43218852/Factors_affecting_profitability_in_Malaysia

Malik, E. F., Khaw, K. W., Belaton, B., Wong, W. Y., & Chew, X. (2022). Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture. Mathematics, 10(9), 1480. https://doi.org/10.3390/math10091480

Guan, J., Leung, E., Kwok, K., & Chen, F. Y. (2023). A hybrid machine learning framework to improve prediction of all-cause rehospitalization among elderly patients in Hong Kong. BMC Medical Research Methodology, 23(1). https://doi.org/10.1186/s12874-022-01824-1

Coad, A., Krafft, J., &amp; Quatraro, F. (2018). Firm age and performance. Retrieved April 19, 2023, from https://www.researchgate.net/profile/Alexander-Coad/publication/290182119_Firm_age_and_performance/links/59a0813caca2726b901150c5/Firm-age-and-performance.pdf

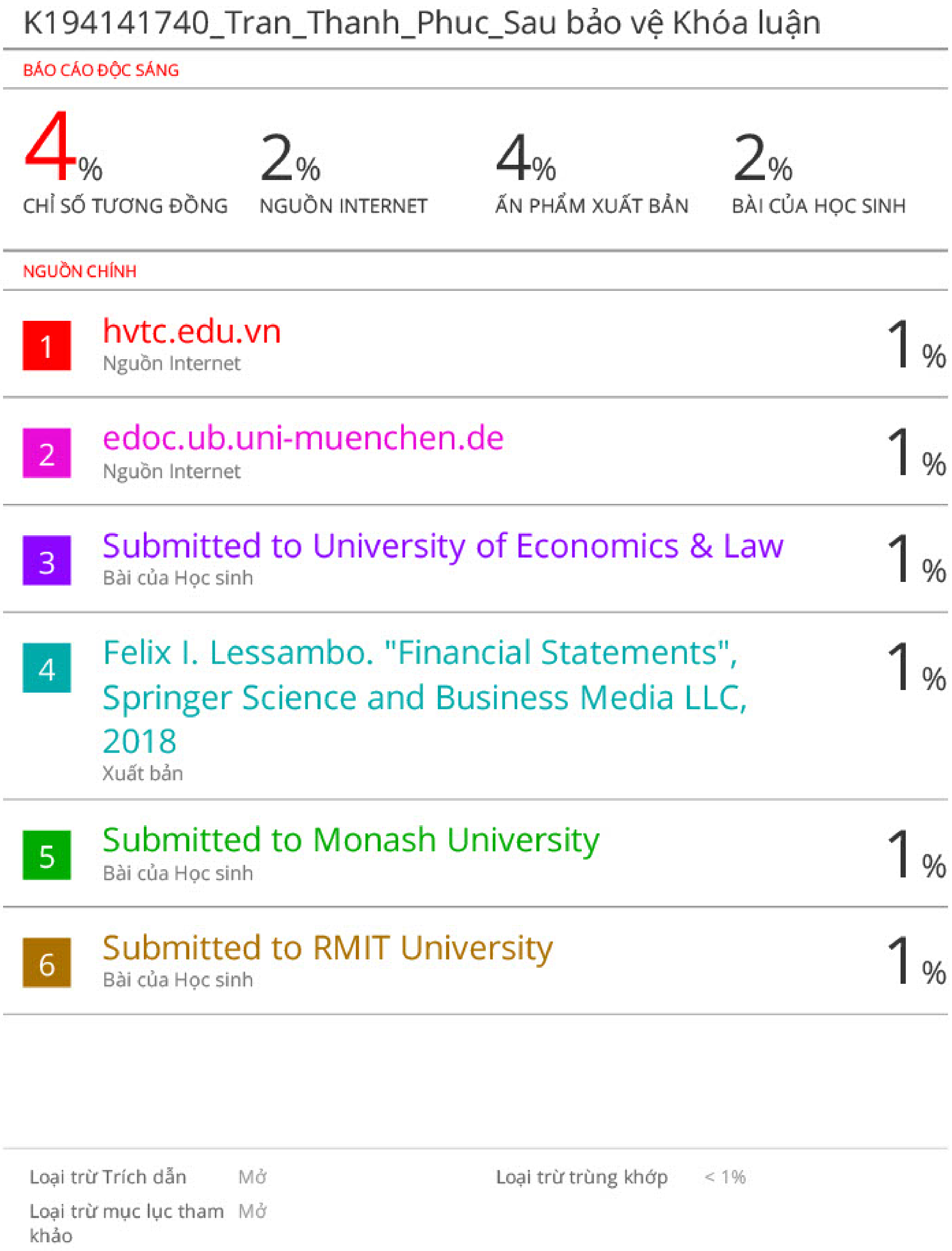
Syahbandar, N., & Lestari, N. (2023). Factors Affecting Profitability in Manufacturing Sector Companies Listed on BEI. https://doi.org/10.4108/eai.5-10-2022.2325855

Abebaw, K. (2018). Factors Affecting the Profitability of Small and Medium Enterprises in South Gondar: Evidence from Woreta Town. http://etd.aau.edu.et/handle/123456789/13568

Solegalli, S. (2022). Feature selection with feature-engine. Kaggle. Retrieved April 15, 2023, from https://www.kaggle.com/code/solegalli/feature-selection-with-feature-engine

Brownlee, J. (2021). No free lunch theorem for machine learning. MachineLearningMastery.com. Retrieved April 15, 2023, from https://machinelearningmastery.com/no-free-lunch-theorem-for-machine-learning/

Alnaim, M., & Kouaib, A. (2023). Inventory Turnover and Firm Profitability: A Saudi Arabian Investigation. Processes, 11(3), 716. https://doi.org/10.3390/pr11030716



K194141740_Tran_Thanh_Phuc_Sau bảo vệ Khóa luận

BÁO CÁO ĐỘC SÁNG

4%	2%	4%	2%
CHỈ SỐ TƯƠNG ĐỒNG	NGUỒN INTERNET	ẤN PHẨM XUẤT BẢN	BÀI CỦA HỌC SINH

NGUỒN CHÍNH

1	hvtc.edu.vn Nguồn Internet	1 %
2	edoc.ub.uni-muenchen.de Nguồn Internet	1 %
3	Submitted to University of Economics & Law Bài của Học sinh	1 %
4	Felix I. Lessambo. "Financial Statements", Springer Science and Business Media LLC, 2018 Xuất bản	1 %
5	Submitted to Monash University Bài của Học sinh	1 %
6	Submitted to RMIT University Bài của Học sinh	1 %

Loại trừ Trích dẫn	Mở	Loại trừ trùng khớp	< 1%
Loại trừ mục lục tham khảo	Mở		