2024 S.T. Yau High School Science Award (Asia)

Research Report

The Team

Registration Number: Econ-108

Name of team member: LI Yat Nok
School: Diocesan Girls' School

Name of team member: PAN Yam Yi
School: Diocesan Girls' School

Name of supervising teacher: CHEN Wei
Job Title: Teacher
School: Diocesan Girls' School

Title of Research Report

Modeling for Green Finance and Sustainable Investment Decision-Making

Date

12 August 2024

Modeling for Green Finance and Sustainable Investment Decision-Making

LI Yat Nok, PAN Yam Yi

Abstract

Green finance and sustainable investing have gained significant prominence in recent years as a means to facilitate sustainable development while promoting economic growth. This research aims to develop two decision models specifically tailored for green finance to enable investors and entrepreneurs to maximize returns while contributing to the development of a green society. The first model addresses sustainable investment decision-making, underscoring the critical need for robust decision-making tools in green finance to safeguard the interests of investors. Through an extensive literature review, this study examines existing models, frameworks, and regulatory landscapes governing sustainable investment and the green finance industry. Building upon this foundation, a comprehensive framework is established, incorporating key quantitative metrics such as price, risk, leverage, growth, and environmental, social, and corporate governance factors. The second model focuses on profit estimation for companies engaged in green finance initiatives, recognizing the intrinsic link between financial performance and sustainability efforts. This model integrates qualitative factors, notably reputation, with quantitative considerations encompassing costs, revenues, market environment, and government policies. By using machine learning, financial and environmental data are first collected and preprocessed using Python libraries like pandas and scikit-learn. An appropriate model, such as linear regression or random forests is selected. The model will be then trained on historical data. We will validate its performance and deploy it to continuously provide profit estimates, aligning the company's financial and sustainability goals. This comprehensive approach ensures the robustness and reliability of the models, enhancing their practical utility for decision-makers.

Furthermore, this research contributes actionable policy recommendations to catalyze the broader adoption of green finance and sustainable development practices. By providing decision-makers with powerful quantitative and qualitative tools, this study paves the way for informed and responsible investment decisions that harmonize financial returns with environmental and social imperatives, driving the global transition towards a more viable future.

Keyword: Green finance, sustainable investing, decision making, profit estimation, machine learning, economic growth, environmental, social and corporate governance (ESG)

References: Nil

Commitments on Academic Honesty and Integrity

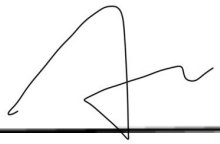
We hereby declare that we

1. are fully committed to the principle of honesty, integrity and fair play throughout the competition.
2. actually perform the research work ourselves and thus truly understand the content of the work.
3. observe the common standard of academic integrity adopted by most journals and degree theses.
4. have declared all the assistance and contribution we have received from any personnel, agency, institution, etc. for the research work.
5. undertake to avoid getting in touch with assessment panel members in a way that may lead to direct or indirect conflict of interest.
6. undertake to avoid any interaction with assessment panel members that would undermine the neutrality of the panel member and fairness of the assessment process.
7. observe the safety regulations of the laboratory(ies) where the we conduct the experiment(s), if applicable.
8. observe all rules and regulations of the competition.
9. agree that the decision of YHSA(Asia) is final in all matters related to the competition.

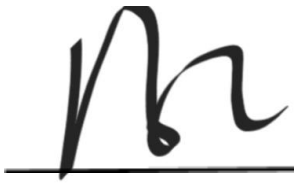
We understand and agree that failure to honour the above commitments may lead to disqualification from the competition and/or removal of reward, if applicable; that any unethical deeds, if found, will be disclosed to the school principal of team member(s) and relevant parties if deemed necessary; and that the decision of YHSA(Asia) is final and no appeal will be accepted.

(Signatures of full team below)

Name of team member: LI Yat Nok



Name of team member: PAN Yam Yi



Name of supervising teacher: CHEN Wei

Noted and endorsed by

(signature)

Mr. Joe Oddie, Deputy Head of Headmaster, Diocesan Girls' School

Table of Contents

1. Introduction

In the ever-changing global financial landscape, the rise of green finance signals a real move towards sustainable economic practices. The decision-making patterns of individual investors and firms are critical to capitalizing on this new investment opportunity, which not only provides financial returns but also contributes to environmental sustainability. This paper explores the decision-making frameworks used by investors and corporations to enter the green finance industry, in the hope that there will be a good cycle of investor-corporation relationships that will contribute to the sustainable development of green finance.

Individual investors need to make decisions based on an integrated understanding of risk, return and growth, among other things, under a combination of ESG (Environmental, Social and Governance) and traditional financial measurement metrics. Green products such as green bonds have proven to be a useful investment option. Since its inception in 2007, the number of green bonds has grown rapidly, supporting projects ranging from renewable energy to sustainable agriculture, and delivering strong returns for investors while having a clear environmental impact.

On the other hand, companies are increasingly recognizing the value of green finance in driving innovation, opening up new markets and enhancing reputation. For example, Google has committed to 24/7 carbon-free energy by 2030, and its investments in renewable energy technologies and green infrastructure not only reinforce its environmental credentials but also set the standard for corporate responsibility and innovation in green finance.

Given the enormous benefits of green finance for both investors and businesses. This paper will explore how a home-grown decision-making model can ensure that the benefits to both parties are maximized, and provide examples of how these investments have been successfully realized to produce economic, environmental, and social benefits.

2. Main reasons causing limited widespread adoption of Green Finance by companies

Green finance, which refers to financial investments flowing into sustainable development projects and initiatives, supports the achievement of long-term environmental goals, including those related to climate change mitigation and adaptation. Despite its importance, the widespread adoption of green finance has been somewhat limited. Here are two detailed reasons why this is the case:

2.1 Not Enough Investors

Green finance projects often require large-scale investment to get off the ground, which can be a barrier for many potential investors. Investment in renewable energy infrastructure, for example, needs substantial upfront capital that may not realize returns for several years. This long-term, capital-intensive nature may not align with the financial goals or risk profiles of many investors. Investing in green projects involves certain perceived and real risks that many traditional investors are hesitant to take on. These risks include technology risk, market risk, and regulatory risk. For instance, technologies used in green projects, such as carbon capture and storage or new renewable energy technologies, might not yet be proven at scale, which can dissuade investors who are risk-averse.

There is also a knowledge gap in understanding the intricacies of green projects. Many traditional investors lack the expertise to accurately assess the viability and potential returns of green investments. Without a clear understanding, the uncertainty makes it difficult to commit substantial resources to green finance.

2.2 Not Enough Promotions and Policy Implementation by Government

2.2.1 Role of Government

Governments around the world, from developing countries such as Kenya to developed countries such as Germany, have made significant contributions to the promotion of green finance, helping economies to grow sustainably. Kenya has set a regional standard by pioneering the issuance of green bonds to finance renewable projects (*First Green Bond From Kenya: Acorn USD40m - Climate Bonds Certified, Financing Green Buildings*, 2019), while Germany has incorporated green finance into national regulations and supported energy efficiency projects through subsidies and tax incentives (*Energy Efficiency Policy in Germany*, n.d.). Similarly, China is a leader in green bonds, guided by strict protocols to fund projects that reduce environmental impacts (*Schipke, n.d.*). These efforts highlight how important Governments are in shaping financial markets to achieve environmental goals.

2.2.2 Result of Lack of Policies

Governments promote green finance through incentives such as tax breaks and subsidies, which can go a long way towards encouraging investment in green projects. Without these incentives, investment becomes less attractive. Stable and consistent policies are essential to ensure long-term investment in the green sector. In addition, Governments must actively engage in

educational campaigns to raise awareness of the benefits of green finance in order to attract potential investors. Without global coordination in these efforts, opportunities for international investment and cooperation could be missed. Good government practices and policies, as well as strong support for green finance, are therefore indispensable elements in promoting the globalization of green finance.

2.3 Macro Analysis

2.3.1. Government Policy and Regulations

2.3.1.1 The European Union (EU)

The European Commission intends to issue EUR 250 billion green bonds which would be 30% of NextGenerationEU funds in 2022 (*Browse Policies and Regulations, n.d.)*.

2.3.1.2 The United State of America (USA)

In 2014, California issued a USD 300 million green bond. The California Lending for Energy and Environmental Needs (CLEEN) Center provides direct public financing to municipalities, universities, schools, and hospitals to help meet the State's goals for greenhouse gas (GHG) emissions reduction, water conservation, and environmental conservation.

2.3.1.3 China

In 2019, Hong Kong sold a five-year U.S. dollar-denominated green bond with a coupon of 2.5%, attracting over $4 billion in orders from investors globally. The People's Bank of China ("PBOC") issued Guidelines defining criteria and categories for green bond projects, which cover green financial bonds within the interbank market in 2020. In July 2021, the Shenzhen Stock Exchange issued an update of its guidelines for green corporate bonds to include carbon neutrality green corporate bonds and blue bonds. In January 2021, Bank of China issued the first transition bond from China.

2.3.2 Financial Environment

The annual inflation rate in the US is expected to fall for a third straight month to 3.1% in June 2024, the lowest since January, compared to 3.3% in May *("United States Inflation Rate")*.The business confidence will become stronger under the low inflation rate which will boost investment.

2.3.3 Conclusion

In summary, the limited spread of green finance can be largely attributed to a lack of sufficient investors, driven by a combination of risk aversion, long-term capital requirements, and a

knowledge gap, as well as insufficient governmental promotion and policy implementation, characterized by inadequate incentives, policy uncertainty, and lack of awareness. Addressing these issues could help increase the adoption and effectiveness of green finance initiatives globally.

Although there are not many existing policies, we can still see the development trend of green finance. More policies are believed to appear in the near future. In general, countries around the world agree on the urgency of developing green finance and are actively promoting policies to encourage sustainable development of finance. Therefore, green finance has great room for development in the international community.

3. Model for investors in making decision whether to invest in Green Finance

3.1 Literature review

The literature review shows that environment protection, social responsibility and governance standards (ESG) is an efficient and useful tool of sustainable investment. ESG integration acts as the second largest global investment strategy. Meanwhile, previous research showed that other than ESG, positive and negative screening are also important approaches *(Talan)*. In this study, a complete model which can guide investors' decision-making in order to maximize their return will be built, combined with the above techniques and other quantitative methods. Below shows a simple existing investment decision-making model *("Theoretical framework of the study | Download Scientific Diagram")*, other variables and metrics are needed as a more accurate model will be setted up.

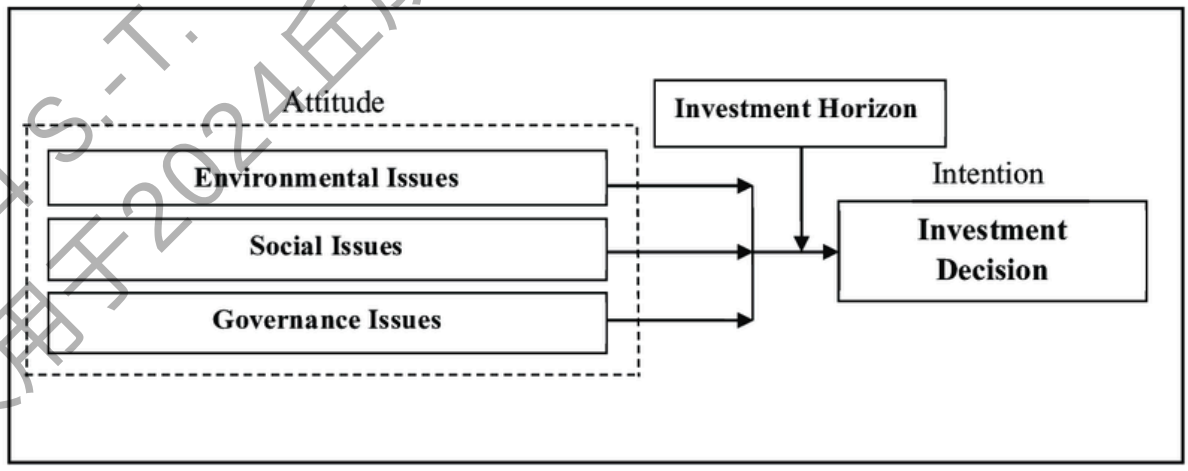


Figure 3a Existing Model Sample

3.2 Conceptual framework

This study aims to set up a model that can help users avoid risks. The original concept of this model is to encourage investment in green finance and avoid possible losses. After refinement and improvement, the framework of the final model is:

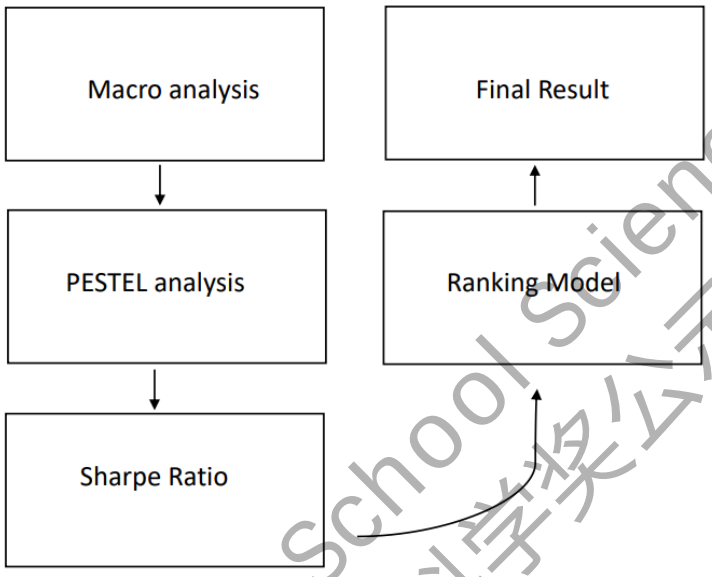


Figure 3b Framework of our Model

After macro analysis, the next step is to use qualitative methods to filter industry to see whether this industry is qualified to be a "potential industry". The criteria of Political, Economic, Social, Technology, Environmental and Legal (PESTEL analysis). After screening the industry, quantitative methods will be used to measure the stability of a company after it issued green bonds or shares by considering

1. EV/EBITDA ratio: whether the company is overvalued or not.
2. Stock volatility: the speed of change of the price of a stock, hence evaluate risk
3. Debt/Equity Ratio: reflect the use of debt of a company
4. Revenue Growth Rate: to see how fast the revenue increase
5. ESG: non-financial factors impacting a company's long-term success, especially for green finance

The result can reflect the stability of a company. More fluctuations, less stable return, vice versa. A programme will be written to automate the above procedure. Then it will show what percent

the company ranks in the industry and the suggested rates for ‘yes’ and ‘no’ option. Users can refer to the result to decide whether to invest or not.

3.3 Decision-making model

3.3.1 Screening

sector industry	Sharpe Ratio
energy	0.2
materials	0.42
industrials	0.55
utilities	0.47
healthcare	0.7
financials	0.53
consumer discretionary	0.59
consumer staples	0.59
information technology	1.02
communication services	0
real estate	0.28

Figure 3c Table of data (*ALL data collected from Yahoo Finance*)

The Sharpe ratio is known as an useful indicator to compare the return of an investment with its risk. A Sharpe ratio less than 0.5 is considered unacceptable or bad. From 0.5-0.99 is considered normal. From 1 to 1.99 is considered good, from 2 to 2.99 is considered very good, and greater than 3 is considered excellent. From the table on the left, it shows that ‘information technology’ is the only sector where its Sharpe ratio is larger than 1 which is considered good and four sectors are considered ‘normal’. Note that the Sharpe ratio will change with time, so investors are recommended to check the Sharpe ratio before they make their investment decision. Meanwhile, investors are suggested to use the PESTEL analysis to further analyze and monitor the macro-environmental factors that are affecting the industry. For ease of understanding, the PESTEL analysis will be presented as a scoring system, scores range from 0 to 5.

For Example: PESTEL for Information Technology

Political: 5 It is perfectly supported by the government.

Economic: 5 It shows remarkable valuation in the total contribution of the global GDP.

Social: 4 The age group of costumes are very broad, except for some of the elderly.

Technology: 5 Promoting innovation and development of science and technology.

Environment: 4 Not very harmful to the environment but the carbon footprint is not low enough.

Legal: 5 It does not violate any law.

Mean: 4.67 >4

As a result, the mean score of the information technology sector is higher than 4 which is considered as good. For an industry which satisfies the above 2 criteria, it is regarded as a "potential industry". After selecting the sector, users can select companies that are issuing green stocks or securities and input them into the code and conduct risk assessment and review.

3.3.2 Data Collection

A Python Crawler Programme is used to collect data, including the EV/EBITDA ratio, stock volatility(BETA), Debt/Equity Ratio, Revenue Growth Rate, and ESG, all of these can be found in the Yahoo Finance website. However, stock code is required as the stock code corresponding to each company is the suffix of its webpage. Thus, a complete webpage link is formed and data can be collected. Therefore, another Python Crawler Programme is written to extract the stock code list so as to facilitate the process.

3.3.3 Z-score

After all information needed is collected, they are transformed into a z-score system. The z-score of each value of the company selected will be calculated by using the following formula.

$$Z = \frac{(x - \mu)}{\sigma}$$

The larger the Z-score, the further from the norm the data set. A Python program will be written to facilitate the process. The z-scores are weighted as EV/EBITDA ratio =0.2, stock volatility (BETA) = 0.2, Debt/Equity Ratio =0.1, Revenue Growth Rate =0.2, and ESG =0.3 and will be added up. The final ranking of that company among all companies will be generated according to their final score and the recommended rate to invest in the selected company will be shown. Users are suggested to make their decision based on the result. After inputting the name of the selected company, a result like below will be generated.

```
Ranking percentage: 6.68%
Yes value: 74.7510067114094%, No value: 25.248993288590597%
```

3.4 Empirical Analysis

3.4.1 Case study - Meituan

In order to test the accuracy of the model, data from June 2023 will be input into the model, and then the model results will be compared with reality to calculate the model deviation. In order to

calculate past results, the code was slightly changed, and the crawled data was changed to the data in June 2023.

sector industry	Sharpe Ratio
energy	1.33
materials	0.42
industrials	0.43
utilities	0.12
healthcare	0.42
financials	0.48
consumer discretionary	0.58
consumer staples	0.26
information technology	0.55
communication services	0.2
real estate	0.11

Figure 3e Table of Past Data

The table above shows the Sharpe ratio of each sector in June 2023. Energy sector is the only sector that is considered 'good' while the consumer discretionary and information technology sector are considered 'normal'.

	P	E	S	T	E	L	AVG
energy	3	4	4	4	5	3	3.83
consumer discretionary	4	3	4	5	4	5	4.17
information technology	5	5	4	5	5	5	4.67

Figure 3f Result of PESTEL analysis

According to the above results, only the consumer discretionary and information technology sector passed the screening, and consumer discretionary was selected as the test object this time. Meituan is one of the companies that belong to the consumer discretionary sector.

	Value
EV/EBITDA ratio	34.81
BETA	0.49
Debt/Equity ratio	0.22
Revenue Growth Rate	0.30

ESG	21.00

Table 3g Data of Meituan

After all information is collected, the result is generated as below:

```
Ranking percentage: 1.66%
Yes value: 78.7610678%, No value: 21.2389322%
```

Figure 3h Result of the programme

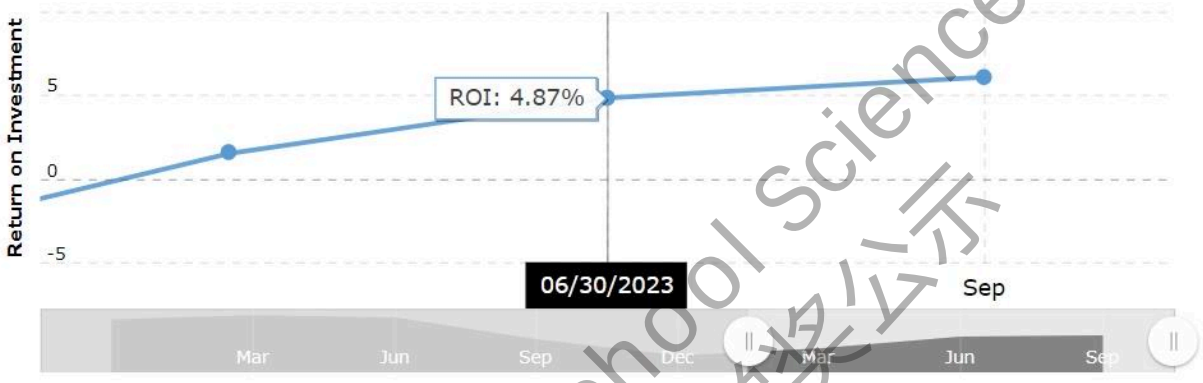


Figure 3i Return on Investment of Meituan from 06/2023 to 09/2023

(*The Long Term Perspective on Markets*, n.d.)

The result suggests that the 'yes' rate is much higher than the 'no' rate. And from the investment return trend chart above, it can be concluded that Meituan's investment report has maintained a steady increase during this period, rising from 4.87% to 6.12%. As a result, this model can indeed predict the approximate investment return trend.

3.4.2 Evaluation of Performance

This model is effective in predicting the direction of investment returns. In calculating the final ranking, the ESG ratio is the highest to increase the influence of environmental impact in the results. Users who want to invest in green finance can use this model to detect whether the investment risk of the target company they want to invest in is high. This model can help users avoid short-term investment losses, but this program is not one-off, because each company's information will change over time, and users also need to check the market regularly.

3.4.3 Comparing tradition model & Ours

The investment decision making models that exist today tend to be more conceptual and theoretical models, providing a rough model framework. The rest rely on users to collect a large

amount of data. This new model is more suitable for novice investors and saves a lot of money. The steps of collecting information are more automated, which simplifies the entire decision-making process, but at the same time, this is also its shortcoming. This may cause users to rely too much on using the program and ignore that the program cannot 100% predict the results, thus investing blindly.

3.5 Implication and Recommendations

3.5.1 Limitations

One key limitation of predictive models is their reliance on historical data. The quantity and quality of data are very important factors which affect the accuracy of the entire model. The data collected by this model may change a lot due to some force majeure, which affects the data quality and consistency. Predictive models are trained on past data, which may not accurately represent future trends or account for unforeseen events and changes in the underlying system. Finally, it causes the short-term effect of this model to be better than the long-term effect.

3.5.2 Policies Suggestion

3.5.2.1 The Establishment of a Comprehensive Framework

A clear and coherent policy framework that defines the government's vision, objectives, and strategies for promoting green finance is suggested to be developed. This framework should align with national sustainable development goals and international commitments, such as the Paris Agreement. Meanwhile, the government should engage with stakeholders, including financial institutions, investors, civil society, and academia, to Gather input and build consensus on the green finance agenda.

3.5.2.2 Strengthen Regulatory and Supervisory Measures

Robust regulations and supervisory mechanisms to ensure the integrity and transparency of green financial products and services should be implemented.This could include mandatory sustainability reporting, green asset classification and enhanced disclosure requirements for financial institutions. The government could also empower financial regulators to monitor the green credentials of financial institutions and products, and enforce compliance with regulations.

3.5.2.3 The Development of Green Financial Instruments

The government is recommended to facilitate the issuance and trading of green bonds, green loans, green mortgages, and other innovative green financial instruments to mobilize private capital for sustainable projects. Additionally, the government could explore the potential of

hybrid financing mechanisms such as public-private partnerships to reduce risks and leverage private capital for green investments.

4. Profit estimation model for companies adopting Green Finance

4.1 Brief Framework

Our second mod is built for companies to help them determine the benefits of executing green finance. And this mod requires the use of machine learning methods as well as a large number of cases and data to support it. As just mentioned, green finance has not been adopted by the general public, and after using model 1 to help investors make decisions easily and call on the government to enact more measures to support green finance, we will get enough cases, both successful and unsuccessful, as well as their data on each of the below metrics. Then we can start the second model, the profit estimation model, and we will build the second half of the model, the machine learning, before green finance is popularized. The second model can be started as soon as enough data is available.

4.2 Important metrics

When developing a model to estimate the profitability and impact of green finance initiatives, incorporating both quantitative and qualitative factors is crucial. These factors help in capturing the broad spectrum of influences on financial performance and the sustainability impact of these initiatives.

4.2.1 Quantitative metrics

1. Financial Metrics
- Return on Investment (ROI): Measures the efficiency of an investment.
- Net Present Value (NPV): Calculates the present value of net cash flows over the life of the project.
- Internal Rate of Return (IRR): Estimates the profitability of potential investments.
- Payback Period: Time it takes for an investment to recover its initial outlays.

2. Energy Production and Savings
- Total Energy Produced (kWh): For energy projects, the total output in kilowatt-hours.
- Energy Cost Savings: Reduction in energy costs due to improved efficiencies or renewable generation.

3. Emission Reductions

- CO_2 Emissions Avoided: Quantities of reduced carbon emissions, often critical for projects aimed at environmental sustainability.
- Other Pollutants Reduced: Measures reductions in other emissions like SOx, NOx, particulate matter, etc.

4. Market Factors

- Market Penetration Rates: Reflects the adoption rate of the technology or initiative.
- Commodity Prices: Prices of essential inputs or outputs, such as solar panel costs or electricity rates.

5. Governmental and Regulatory Impact

- Subsidies Received: Monetary value of governmental subsidies.
- Tax Credits and Incentives: Quantitative impact of tax breaks and other incentives.

6. Quantifying reputation

Quantifying reputation involves assessing how much these benefits contribute to the financial metrics such as cash flows or cost reductions. This could be approached by:

- Estimating increased sales from customer loyalty or the ability to charge premium prices.
- Calculating cost savings from favorable credit terms or supplier discounts.
- Assessing the value of reduced employee turnover and increased productivity.

4.2.2 Qualitative metrics

1. Regulatory Environment

- Stability of Policies: Stability in government policies regarding business operations can affect long-term strategic decisions.
- Supportiveness of Regulations: Favorable regulations (e.g., subsidies for green technologies) can enhance project viability.

2. Operational Capabilities

- Management Expertise: The experience and knowledge of the management team can greatly influence project success.
- Operational Efficiency: The ability of the company to execute projects efficiently, affecting cost and time overruns.

3. Technological Advancements

- Rate of Innovation: Industries with rapid technological advancements might require more frequent investments and updates.

- Access to Technology: Availability of advanced technologies that can improve efficiency or product quality.

4. Social Factors

- Public Perception: Consumer attitudes towards environmental sustainability or corporate responsibility can drive sales.
- Social Impact: Projects that significantly benefit the community or environment might receive more favorable internal and external support.

4.3 Key Strategy - machine learning

To write a machine learning program to estimate the profitability of a company that adopts green finance principles, we first need to collect and preprocess relevant data, including financial metrics, as well as the aforementioned metrics referring to costs, environmental impacts, and so on. We can use libraries such as Python and pandas for data processing and scikit-learn for machine learning. First, we want to remove inconsistencies and missing values from the data. Next, comes the all-important feature engineering. We want to create variables that capture the nature of green investments (e.g., renewable energy use and carbon credits) and their financial impact. It's important to choose the right machine learning model based on the data features; regression models such as linear regression or more sophisticated models such as random forests are appropriate. We trained the model using historical data, making sure to divide the data into a training set and a test set to validate the model's performance. Once training is complete, we evaluate the model using metrics such as R-squared and mean squared error to understand its accuracy. Finally, we put the model into a production environment where it receives new data and provides ongoing profit estimates, helping the company align its financial goals with its sustainability goals. This approach not only helps with financial forecasting, but also integrates sustainability into the company's core financial strategy.

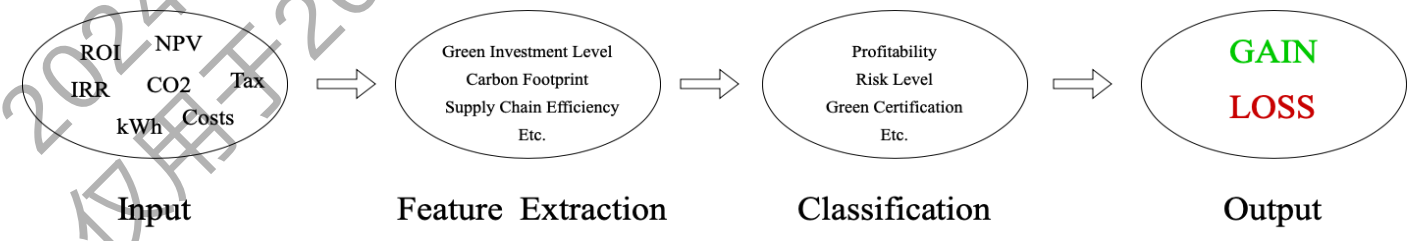


Figure 4.3a Process of Machine Learning

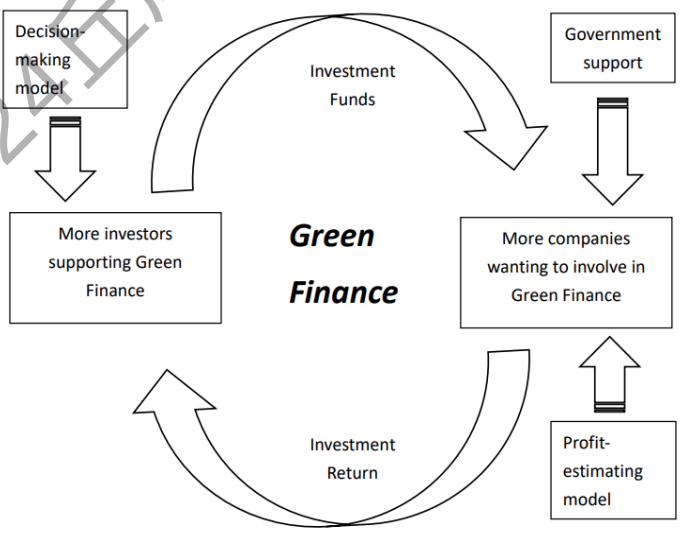
5. Overall Conclusion

The decision model presented in this paper can effectively help users reduce the risk of investment loss in green finance investments. This model, which provides potential risks and returns, can help investors to be able to make more informed and prudent decisions. On the other hand, the Profit Estimation Model helps companies assess the viability and profitability of their green finance programs. This enables them to refine their strategies and decisions to ensure that their green finance activities are not only environmentally responsible but also financially sustainable.

In addition, the widespread use of these models is not without more supportive policies and regulations from governments. Having effectively demonstrated the practical benefits and viability of green finance, policymakers will be better able to develop incentives and frameworks that encourage businesses to prioritize sustainable practices and investments.

The continued development and refinement of these models holds great promise for the growth and widespread adoption of green finance. As more investors and companies utilize these tools, we hope that they will increase confidence and investment in the green finance sector. This, in turn, will lead to a virtuous cycle whereby more capital and resources will flow towards environmentally friendly financial products and services.

Ultimately, the successful implementation of these decision-making and profit estimation models will play a critical role in driving green finance into the mainstream. As more companies and investors embrace this paradigm shift, it will contribute significantly to environmental protection and the transition to a more sustainable future.



6. A letter to government

Dear Sir/Madam,

I am writing on behalf of individuals wishing to contribute to environmental sustainability and economic development entities. As we address such challenges posed by climate change and economic transformation, it is imperative that we implement strong policies that encourage and facilitate green finance initiatives across all sectors.

Green finance, which supports projects and initiatives aimed at environmental sustainability, has proved to be not only a catalyst for innovation but also an important driver of economic stability and growth. However, the full potential of green finance has yet to be unlocked, largely due to a lack of adequate policies and incentives to promote its wider adoption by companies and financial institutions.

Countries such as Germany and Denmark have made commendable progress in integrating green finance into their economies, largely thanks to supportive government policies. For example, Germany's comprehensive incentive program for renewable energy not only reduces vehicle emissions but also increases energy efficiency.

There are lots of benefits of policy support, one of which is the increased market penetration. More companies are encouraged to invest in sustainable technologies through the implementation of policies that reduce financial risk, such as loan guarantees or lower interest rates for green projects. This is evident in Denmark, where government-supported incentives have resulted in one of the highest penetration rates of wind energy use in the world. The second benefit is rewarding the economy and creating jobs. Investments in green technologies promote new industries and jobs. In the U.S., employment growth in the solar industry is 17 times the national average, largely due to federal tax incentives and grants.Finally, public and environmental health can be enhanced, and by reducing pollutants and improving quality of life, support for green finance is also consistent with public health goals. The shift to renewable energy in many cities has demonstrated significant reductions in health problems associated with air pollution, resulting in lower health care costs and increased community productivity.

In light of these solid examples, there is an urgent need for us to strengthen our policies in support of green finance. This can be accomplished in the following ways. Firstly, we can implement tax credits, subsidies, and lower interest rates on loans to companies involved in green projects for fiscal incentives. Secondly, we can enact legislation that mandates or strongly encourages the use of renewable resources for regulatory support. Last but not least, we may educate the public and businesses about the benefits and opportunities of green finance, encouraging more and more entrepreneurs to involve in such meaningful business.
We urge your esteemed office to consider these points and initiate a comprehensive dialogue among stakeholders to craft policies that will position our country as a leader in green finance. By doing so, we not only safeguard our environment but also stimulate our economy in a sustainable manner.

Thank you for considering this vital aspect of our nation's future. We are eager to assist and participate in this endeavor and look forward to your support.

Yours sincerely.

Complete coding(decision-making model):

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 31 22:25:47 2024

@author: MUDUO
"""

# Regular Expression Library
import re
# HTTP Request Library
import requests
# HTML Parsing Library
from bs4 import BeautifulSoup
# MySQL Database Library
import pymysql
import pandas as pd
from sqlalchemy import create_engine
import mysql.connector
from mysql.connector import Error

# Constants
url_stock_code_HK = 'https://www.etnet.com.hk/www/eng/stocks/cas_list.php'
# url_stock_code_US =
'https://www.futunn.com/en/quote/us/stock-list/all-us-stocks/top-market-cap'
base_url_statistics = "https://finance.yahoo.com/quote/{}.HK/key-statistics/"
base_url_sustainability = "https://finance.yahoo.com/quote/{}.HK/sustainability/"
# Statistics Page Type
type_statistics = 0
# Sustainability Page Type
type_sustainability = 1

# Database Connection Parameters
connection_params = {
        'host': 'localhost',
        'user': 'root',
        'password': '123456',
        'database': 'styau',
        'charset': 'utf8mb4',
        'cursorclass': pymysql.cursors.DictCursor  # Optional, using DictCursor makes accessing
columns in result sets easier
    }

def get_raw(url):
    """
    Fetch the latest source code of the given URL and parse it using BeautifulSoup.
```

```
    Args:
        url (str): The URL to fetch and parse.

    Returns:
        soup (BeautifulSoup): Parsed source code data.
    """

    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'
    }
    response = requests.get(url, headers=headers)

    soup = BeautifulSoup(response.text, 'html.parser')

    # Check if the request was successful
    if response.status_code == 200:
        # HTML content is parsed
        return soup
    else:
        print('Request failed, status code:', response.status_code)
    return soup

def get_db_connection(connection_params):
    """
    Establish a connection to the database using the provided connection parameters.

    Args:
        connection_params (dict): Database connection parameters including host, user, password,
etc.

    Returns:
        connection (pymysql.Connection): Database connection object.
    """
    connection = pymysql.connect(**connection_params)
    return connection


def insert_code_hk_to_mysql(company_list):
    """
    Insert a list of Hong Kong stock codes and company names into the MySQL database.

    Args:
        company_list (list of tuples): A list of tuples, where each tuple contains a stock code and a
company name.
```

```
    Raises:
        Exception: If an error occurs during the insertion process.
    """
    connection = get_db_connection(connection_params)  # Assuming connection_params is
globally defined or passed as an argument

    try:
        with connection.cursor() as cursor:
            # Iterate through the company list, adding a stock_type value (0 in this case) for each
record
            for stock_code, company_name in company_list:
                # Prepare data to insert (including stock_type)
                data_to_insert = (stock_code, company_name, 0)
                if (stock_code != 'Code' and stock_code != ''):
                    # Prepare SQL insert statement
                    sql = "INSERT INTO companies (stock_code, company_name, stock_type)
VALUES (%s, %s, %s)"

                    # Execute SQL statement
                    cursor.execute(sql, data_to_insert)

            # Commit the transaction
            connection.commit()
            print("All records have been successfully inserted into the database.")

    except Exception as e:
        print(f"An error occurred while inserting data: {e}")
        if connection:
            connection.rollback()  # Rollback the transaction if an error occurs

    finally:
        # Close the database connection
        connection.close()

# Fetch Hong Kong stock codes from a web page and persist them to a database
def fetch_code_hk_from_page(soup):
    """
    Extract Hong Kong stock codes and related information from a web page and prepare them for
database insertion.

    Args:
        soup (BeautifulSoup): A BeautifulSoup object representing the parsed HTML content of the
web page.

    Raises:
```

```
    Exception: If an error occurs during the extraction or insertion process.
    """
    # Assume the HTML tag for the table is <table>, and we want to find the first table
    table = soup.find('table')

    # Initialize an empty list to store the results
    formatted_cols = []

    # Iterate through each row of the table
    rows = table.find_all('tr')
    for row in rows:
        # Iterate through each cell of the row
        cols = row.find_all(['td', 'th'])  # Sometimes headers are in <th> tags, data in <td> tags
        cols = [ele.text.strip() for ele in cols]  # Extract text and strip whitespace

        # Iterate through the cols list, skipping one element on each iteration
        for i in range(0, len(cols), 2):
            # Combine the current element and the next element into a list and append to
formatted_cols
            formatted_cols.append([cols[i], cols[i+1]])

        # At this point, formatted_cols contains lists of paired elements

    # Insert the extracted data into the MySQL database
    insert_code_hk_to_mysql(formatted_cols)


def select_code_hk_from_db():
    """
    Select Hong Kong stock codes from the database.

    Returns:
        list: A list containing the stock codes extracted from the database.
    """
    # Connect to the database
    connection = get_db_connection()

    try:
        with connection.cursor() as cursor:
            # Compose the SQL query to select the stock codes
            sql = "SELECT stock_code FROM companies"
            # Execute the SQL query
            cursor.execute(sql)
            # Fetch all results and extract the stock codes
            stock_codes = [row['stock_code'] for row in cursor.fetchall()]
```

```
        # Optionally, you can uncomment the following line to print the results for debugging purposes
        # print(stock_codes)
        return stock_codes

    finally:
        # Close the database connection
        connection.close()


# Retrieve stock codes from the database and concatenate them into URLs
def concatenate_url(url_type):
    """
    Concatenate stock codes retrieved from the database with a base URL to form complete URLs.

    Args:
        url_type (str or enum): The type of URL to construct, used to determine the base URL format.

    Returns:
        list: A list of fully constructed URLs based on the stock codes and the specified URL type.
    """
    # Assume select_code_hk_from_db is a function that retrieves stock codes from the database
    stock_codes = select_code_hk_from_db()

    # Use list comprehension and string slicing to remove leading '0's from each stock code string
    stock_codes = [x[1:] if x.startswith('0') else x for x in stock_codes]

    # Define the base URLs based on the URL type (Assuming type_statistics, base_url_statistics, and base_url_sustainability are defined)
    # Note: Ensure type_statistics is a valid identifier or consider using a string or enum for url_type
    if url_type == 'statistics':  # Assuming 'statistics' as an example, adjust based on actual usage
        # Use list comprehension and f-string to generate a list of complete URLs
        urls = [f"{base_url_statistics.format(stock_code)}" for stock_code in stock_codes]
    else:
        urls = [f"{base_url_sustainability.format(stock_code)}" for stock_code in stock_codes]

    return urls


# Get the 5 data of each company
def fetch_data_from_url(url1, url2):
    # Store all crawled data
```

```
    data = []

    # Parse url to get stock_code
    stock_code = format_number_from_url(url1)
    # Obtain the company's stock_code, company_name, stock_type information through
stock_code
    company_info = get_company_info_by_stock_code(stock_code)

    data.extend(company_info)

    # Initialize result variables
    # variable1
    enterprise_value_eBITDA_result = ""
    # variable2
    TotalDebt_Equity_mrq_result = ""
    # variable3
    Quarterly_Revenue_Growth_yoy_result = ""
    # variable4
    Beta_5Y_Monthly_result = ""
    # variable5
    Total_ESG_Risk_Score_result = ""


    # Single link for testing
    # url1 = 'https://finance.yahoo.com/quote/6078.HK/key-statistics/'
    soup = get_raw(url1)

    # Find the target element using the selector you provide
    enterprise_value_eBITDA = soup.select_one('#nimbus-app > section > section > section >
article > section.yf-104jbnt > div.table-container.yf-104jbnt > table:nth-child(1) > tbody >
tr:nth-child(9) > td:nth-child(2)')

    # Check if the element is found
    if enterprise_value_eBITDA:
        # Extract and return the text in an element
        enterprise_value_eBITDA_result =  enterprise_value_eBITDA.text.strip()
    data.append(enterprise_value_eBITDA_result)


    TotalDebt_Equity_mrq = soup.select_one('#nimbus-app > section > section > section > article
> article > div > section:nth-child(1) > div > section:nth-child(5) > table > tbody > tr:nth-child(4)
> td.value.yf-vaowmx')
    # Check if the element is found
    if TotalDebt_Equity_mrq:
        # Extract and return the text in an element
        TotalDebt_Equity_mrq_result =  TotalDebt_Equity_mrq.text.strip()
```

```
    data.append(TotalDebt_Equity_mrq_result)


    Quarterly_Revenue_Growth_yoy = soup.select_one('#nimbus-app > section > section > section > article > article > div > section:nth-child(1) > div > section:nth-child(4) > table > tbody > tr:nth-child(3) > td.value.yf-vaowmx')
    if Quarterly_Revenue_Growth_yoy:
        # Extract and return the text in an element
        Quarterly_Revenue_Growth_yoy_result = Quarterly_Revenue_Growth_yoy.text.strip()
    data.append(Quarterly_Revenue_Growth_yoy_result)


    Beta_5Y_Monthly = soup.select_one('#nimbus-app > section > section > section > article > article > div > section:nth-child(2) > div > section:nth-child(1) > table > tbody > tr:nth-child(1) > td.value.yf-vaowmx')
    if Beta_5Y_Monthly:
        # Extract and return the text in an element
        Beta_5Y_Monthly_result = Beta_5Y_Monthly.text.strip()
    data.append(Beta_5Y_Monthly_result)


    # Single link for testing
    #url2 = "https://finance.yahoo.com/quote/6078.HK/sustainability/"

    soup2 = get_raw(url2)

    Total_ESG_Risk_Score = soup2.select_one('#nimbus-app > section > section > section > article > section:nth-child(3) > section:nth-child(1) > div > section:nth-child(1) > div > div > h4')
    if Total_ESG_Risk_Score:
        # Extract and return the text in an element
        Total_ESG_Risk_Score_result = Total_ESG_Risk_Score.text.strip()
    data.append(Total_ESG_Risk_Score_result)

    print(data)

    insert_company_info(data)


# Fetch data from all URLs obtained from the concatenation process
def fetch_all_data_from_urls():
    """
    Retrieves data from all URLs generated by concatenating specific types of URLs.

    This function combines two sets of URLs (one for statistics and one for sustainability)
```

```
    and fetches data from each pair of URLs using the fetch_data_from_url function.
    """
    urls1 = concatenate_url('statistics')  # Assuming 'statistics' is the correct type for the first set of URLs
    urls2 = concatenate_url('sustainability')  # Assuming 'sustainability' is the correct type for the second set of URLs

    # Use zip to iterate over both lists in parallel
    for url1, url2 in zip(urls1, urls2):
        print(url1)  # Optionally print the URLs for debugging purposes
        print(url2)
        fetch_data_from_url(url1, url2)  # Fetch data from the current pair of URLs


def format_number_from_url(url):
    """
    Extracts and formats a number found in the given URL.

    Args:
        url (str): The URL string to search for a number.

    Returns:
        str: The formatted number if found, formatted as a string with leading zeros to a total of 5 digits.
             If no number is found, returns a message indicating that no number was found.
    """
    # Use regular expression to find numbers in the URL
    match = re.search(r'\d+', url)
    if match:
        # Convert the matched number to an integer and format it
        number = int(match.group(0))
        formatted_number = f"{number:05d}"
        return formatted_number  # Return the formatted number
    else:
        # If no number is found, return an error message or None
        return "No number found"


def get_company_info_by_stock_code(stock_code):
    """
    Retrieves company name and stock type based on the provided stock code, returning the results (including the stock code) as a list.

    :param stock_code: The stock code to query for company information.
```

```
    :return: A list containing the query results, with the first element being the stock_code,
followed by company_name and stock_type.
             If no record is found, an empty list is returned.
    """
    try:
        connection = get_db_connection()  # Ensure this function correctly returns a database
connection

        with connection.cursor() as cursor:
            # SQL query statement
            sql = "SELECT stock_code, company_name, stock_type FROM companies WHERE
stock_code = %s"
            cursor.execute(sql, (stock_code,))

            # Fetch the query result
            result = cursor.fetchone()  # Assuming each stock_code corresponds to a single record

            if result:
                # If result is a tuple (assuming cursor.fetchone() returns a tuple), access elements by
index
                return [result[0], result[1], result[2]]  # Assuming the order is stock_code,
company_name, stock_type
            else:
                # If no record is found, return an empty list
                return []
    except Exception as e:
        # Catch and handle any exceptions, such as database connection failures, query errors, etc.
        print(f"An error occurred: {e}")
        return []  # Alternatively, raise an exception based on the application's requirements



def insert_company_info(data):
    """
    Inserts company information into the database.

    :param data: A list containing company information, where the order should match the column
order of the database table (excluding the id, assuming it's auto-incremented).
    """
    try:
        # Connect to the database
        connection = get_db_connection()

        with connection.cursor() as cursor:
            # Construct the SQL insert statement, assuming the id is auto-incremented and thus not
included
```

```
            sql = """
                INSERT INTO company_info (stock_code, company_name, stock_type,
Enterprise_value_EBITDA,
                    TotalDebt_Equity_mrq, Quarterly_Revenue_Growth_yoy, Beta_5Y_Monthly,
Total_ESG_Risk_Score)
                VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
            """

            # Convert data, replacing None and empty strings with "
            values = tuple('' if (value is None or value == '') else value for value in data)

            # Execute the insert operation
            cursor.execute(sql, values)

            # Commit the transaction
            connection.commit()

            # Assuming stock_code is the first field, we can retrieve it from data to print
            print(f"Data inserted for stock_code: {data[0]}")

    except pymysql.MySQLError as e:
        # Handle database connection or query errors
        print(f"Database error: {e}")
        connection.rollback()  # Rollback the transaction if there's an error

    finally:
        # Close the database connection
        connection.close()



def calculate_and_store_company_rankings(connection_params):
    """
    Calculate company rankings based on financial metrics and store them in a database.

    Parameters:
    - connection_params: A dictionary containing the database connection parameters.
    """
    # Connect to the database
    connection = pymysql.connect(**connection_params)

    try:
        # Read data from the database
        with connection.cursor() as cursor:
            sql = """
```

```
        SELECT stock_code, company_name, Enterprise_value_EBITDA,
TotalDebt_Equity_mrq,
            Quarterly_Revenue_Growth_yoy, Beta_5Y_Monthly, Total_ESG_Risk_Score
        FROM company_info
        """
        cursor.execute(sql)
        data = cursor.fetchall()

    # Convert data to DataFrame
    df = pd.DataFrame(data)
    df.columns = [
        'stock_code', 'company_name', 'Enterprise_value_EBITDA', 'TotalDebt_Equity_mrq',
        'Quarterly_Revenue_Growth_yoy', 'Beta_5Y_Monthly', 'Total_ESG_Risk_Score'
    ]

    # Convert strings to floats (assuming all values can be converted)
    for col in df.columns[2:]:
        df[col] = df[col].astype(float)

    # Calculate means and standard deviations
    stats = df.iloc[:, 2:].describe().loc[['mean', 'std'], :].T

    # Initialize results DataFrame
    results = pd.DataFrame(columns=['stock_code', 'company_name', 'weighted_sum'])

    # Weights for calculating the weighted sum
    weights = {
        'Enterprise_value_EBITDA': 0.2,
        'TotalDebt_Equity_mrq': 0.1,
        'Quarterly_Revenue_Growth_yoy': 0.2,
        'Beta_5Y_Monthly': 0.2,
        'Total_ESG_Risk_Score': 0.3
    }

    # Calculate z-scores and weighted sums for each company
    for index, row in df.iterrows():
        z_scores = []
        for col in df.columns[2:]:
            u = stats.loc[col, 'mean']
            r = stats.loc[col, 'std']
            x = row[col]
            z = (x - u) / r
            z_scores.append(z * weights[col])

        weighted_sum = sum(z_scores)
        results = results.append({
```

```
            'stock_code': row['stock_code'],
            'company_name': row['company_name'],
            'weighted_sum': weighted_sum
        }, ignore_index=True)

    # Rank the companies based on their weighted sums
    results['rank'] = results['weighted_sum'].rank(ascending=False, method='min').astype(int)

    # Prepare database connection string
    database_uri = 'mysql+pymysql://root:123456@localhost/styau'
    engine = create_engine(database_uri)

    # Remove 'id' column if it exists (assuming 'id' is auto-incremented in the database)
    if 'id' in results.columns:
        results = results.drop('id', axis=1)

    # Insert results into the company_rankings table
    results.to_sql('company_rankings', con=engine, if_exists='append', index=False)

    # Print the results
    print(results[['stock_code', 'company_name', 'rank']])

finally:
    # Close the database connection
    connection.close()


def get_company_ranking_percentage_and_yn(stock_code, db_config):
    """
    Retrieve the company's ranking percentage and calculate yes/no values based on the ranking.

    :param stock_code: The stock code of the company to query.
    :param db_config: The database connection configuration as a dictionary.
    :return: A tuple containing the ranking percentage and a tuple of (yes value, no value), or
None with error messages if an issue occurs.
    """
    try:
        # Connect to the database
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            cursor = connection.cursor()

            # Query the company's rank
            query = "SELECT `rank` FROM company_rankings WHERE stock_code = %s"
            cursor.execute(query, (stock_code,))
            result = cursor.fetchone()
```

```
            if result:
                rank = result[0]

                # Query the total number of companies
                query_total = "SELECT COUNT(*) FROM company_rankings"
                cursor.execute(query_total)
                total_companies = cursor.fetchone()[0]

                # Calculate the ranking percentage
                ranking_percentage = (rank / total_companies) * 100

                # Calculate the yes and no values based on the rank
                if rank == 1:
                    yes_value = 80
                    no_value = 20
                else:
                    # Calculate the decrease per rank for yes value
                    decrease_per_rank = (80 - 1) / (total_companies - 1)
                    # Calculate the yes value for the current rank
                    yes_value = 80 - (rank - 1) * decrease_per_rank
                    # Calculate the no value
                    no_value = 100 - yes_value

                return ranking_percentage, (yes_value, no_value)
            else:
                return None, ("Company ranking not found", "Company ranking not found")

        else:
            return None, ("Database connection failed", "Database connection failed")

    except Error as e:
        # Handle any database-related errors
        return None, (f"Database error: {e}", "Database error")

    finally:
        # Ensure resources are cleaned up
        if connection.is_connected():
            cursor.close()
            connection.close()


# Retrieves the 'Yes' and 'No' values for a company based on its stock code
def get_yes_no_by_stock_code(stock_code):

    # Database connection parameters
```

```
    connection_params = {
        'host': 'localhost',
        'user': 'root',
        'password': '123456',
        'database': 'styau',
        'charset': 'utf8mb4',
    }

    # Retrieve the ranking percentage and 'Yes'/'No' values for the company
    percentage, (yes_value, no_value) = get_company_ranking_percentage_and_yn(stock_code,
connection_params)

    if percentage is not None:
        # Print the ranking percentage and 'Yes'/'No' values if available
        print(f"Ranking percentage: {percentage:.2f}%")
        print(f"Yes value: {yes_value}%, No value: {no_value}%")
    else:
        # Print an alternative message if the percentage is not available
        print(f"No ranking percentage available, 'Yes' value: {yes_value}")



# Main function to fetch data
def fetch_data_main():

    # Retrieve Hong Kong stock codes from the web page
    soup = get_raw(url_stock_code_HK)  # Assuming get_raw() is a function to fetch and parse
the HTML

    # Extract Hong Kong stock codes from the web page and persist them into the database
    fetch_code_hk_from_page(soup)  # Assuming this function handles the extraction and
database persistence

    # Fetch all data based on the obtained URLs and persist it into the database tables
    fetch_all_data_from_urls()  # Assuming this function fetches and persists data from multiple
URLs


# Main function for data processing
def data_processing_main():
    """
    Performs the main data processing tasks.

    This function calculates the company rankings based on the data and persists the results to the
database.
    """
```

```
    # Invoke the function to calculate company rankings based on the data and persist the results
to the database
    calculate_and_store_company_rankings(connection_params)


# ----------------------------------Entry Point for Processing Functions-----------------------------

# Uncomment to call the data fetching function to retrieve all required data.
# (Note: Device might be banned by service providers during data fetching. Data has already
been stored in the database, no need to fetch again.)
# fetch_data_main()

# Uncomment to call the main data processing function, which calculates company rankings and
persists the data to the database.
# Data is already prepared in the database.
# data_processing_main()

# Input the stock_code to test
stock_code = '82331'
# Invocation
# Assuming get_yes_no_by_stock_code is a function that performs some operation based on the
stock code
get_yes_no_by_stock_code(stock_code)

# ----------------------------Entry Point for Processing Functions-----------------------------
```

Coding for Companies (Profit-Estimation Model)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Load the dataset
data = pd.read_csv('data.csv')

# Assume the target variable 'Profit' is already in the dataset
X = data.drop('Profit', axis=1)
y = data['Profit']

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Identifying numerical and categorical columns
```

```
numeric_features = [
    'ROI', 'NPV', 'IRR', 'Payback Period', 'Total Energy Produced',
    'Energy Cost Savings', 'CO2 Emissions Avoided', 'Other Pollutants Reduced',
    'Market Penetration Rates', 'Commodity Prices', 'Subsidies Received',
    'Tax Credits and Incentives'
]

categorical_features = [
    'Regulatory Environment', 'Supportiveness of Regulations', 'Management Expertise',
    'Operational Efficiency', 'Rate of Innovation', 'Access to Technology',
    'Public Perception', 'Social Impact'
]

# Creating a column transformer to handle scaling and encoding
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features),
        ('cat', OneHotEncoder(), categorical_features)
    ])

# Creating a pipeline that first transforms the data and then fits a model
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(n_estimators=100, random_state=42))
])

# Fitting the pipeline on the training data
pipeline.fit(X_train, y_train)

# Predicting the test set results
y_pred = pipeline.predict(X_test)

# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-Squared: {r2}")

# Optional: Hyperparameter tuning using GridSearchCV
param_grid = {
    'regressor__n_estimators': [100, 200],
    'regressor__max_depth': [None, 10, 20],
}

grid_search = GridSearchCV(pipeline, param_grid, cv=3)
```

```
grid_search.fit(X_train, y_train)
print("Best parameters:", grid_search.best_params_)

# Best model evaluation
best_model = grid_search.best_estimator_
best_y_pred = best_model.predict(X_test)
best_mse = mean_squared_error(y_test, best_y_pred)
best_r2 = r2_score(y_test, best_y_pred)

print(f"Best Model Mean Squared Error: {best_mse}")
print(f"Best Model R-Squared: {best_r2}")
```

References

(n.d.). Yahoo Finance - Stock Market Live, Quotes, Business & Finance News. Retrieved August 14, 2024, from https://finance.yahoo.com

Browse Policies and Regulations. (n.d.). Green Finance Platform. Retrieved August 14, 2024, from https://www.greenfinanceplatform.org/financial-measures/browse

Energy Efficiency Policy in Germany. (n.d.). sustainable transition china. Retrieved August 14, 2024, from https://transition-china.org/wp-content/uploads/2022/04/Energy-Efficiency-Policy-in-Germany-EN.pdf

First Green Bond from Kenya: Acorn USD40m - Climate Bonds Certified, financing green buildings. (2019, October 2). Climate Bonds Initiative. Retrieved August 14, 2024, from https://www.climatebonds.net/2019/10/first-green-bond-kenya-acorn-usd40m-climate-bonds-certified-financing-green-buildings

The Long Term Perspective on Markets. (n.d.). Macrotrends | The Long Term Perspective on Markets. Retrieved August 14, 2024, from https://www.macrotrends.net

Schipke, A. (n.d.). *China's Green Bond Market*. Hong Kong Green Finance Association. Retrieved August 14, 2024, from https://www.hkgreenfinance.org/wp-content/uploads/2019/12/China%E2%80%99s-Green-Bond-Market.pdf

Talan, G. (2019, January 11). *Doing Well by Doing Good: A Systematic Review and Research Agenda for Sustainable Investment*. MDPI. Retrieved August 14, 2024, from https://www.mdpi.com/2071-1050/11/2/353#B4-sustainability-11-00353

Theoretical framework of the study | Download Scientific Diagram. (n.d.). ResearchGate. Retrieved August 14, 2024, from https://www.researchgate.net/figure/Theoretical-framework-of-the-study_fig1_324994619

United States Inflation Rate. (n.d.). Trading Economics. Retrieved August 14, 2024, from https://tradingeconomics.com/united-states/inflation-cpi