TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Title : Taiwanese Bankruptcy Prediction

Course Code: BMDS2003

Group Number: 5

Semester 202501

Student Name (Block Capital)	**Registration No.**	**Signature**
1.TEOH YU XIANG	2314622	Yuack

Lecturer/Tutor's Name: MS NURUL AKMAR BINTI AZMAN

Date of Submission: 24/12/2025

Table Of Content

Table Of Content 2
1. Executive Summary 4
2. Business Understanding 5
2.1. Business Objective 5
2.2. Key Stakeholders 5
2.3. Project Goals 6
2.4. Success Criteria 6
2.5. Significance and Potential Business Impact 7
2.6. Scope of the Project 8
2.7. Assumptions 8
2.8. Risks 8
4. Data Description 10
4.1. Attribute Information 10
4.1.1. Profitability Ratios 11
4.1.2. Liquidity Ratios 13
4.1.3. Leverage and Debt Ratios 14
4.1.4. Cash Flow Ratios 15
4.1.5. Operational Efficiency Ratios 16
4.1.6. Growth Indicators 16
4.1.7. Risk and Flag Indicators 17
4.2. Data Processing 18
4.3. Exploratory Data Analysis (EDA): 23
4.3.1. Target distribution 23
4.3.2. Box plots 27
4.3.3. Regression Plot 34
4.3.4. Correlation Heatmaps 36
4.3.5. Bar Chart 39
5. Data Preparation 41
5.1.1. Handling Missing Values 41
5.1.2. Duplicated Data Checking 41
5.1.3. Outlier Detection and Handling 42
5.1.4. Capping of outliers 70
5.1.5. Binning for Continuous Data 77
5.1.6. Drop Unnecessary Features 92
5.1.7. Splitting Data into Features and Attributes 96
5.1.8. Training Data and Testing Data Split 99
5.1.9. Data Balancing 100
5.1.10. Data Standardization 103
3. Modeling 104

5.2. Gaussian Naive Bayes 105
5.2.1. Data A 106
5.2.2. Data B 108
5.3. K-Nearest Neighbors (KNN) 110
5.3.1. Data A 112
5.3.2. Data B 114
5.4. Decision Tree 117
5.4.1. Data A 119
5.4.2. Data B 121
5.5. Random Forest 123
5.5.1. Data A 125
5.5.2. Data B 127
4. Model Evaluation 129
4.1. Accuracy 131
4.2. Precision 133
4.3. F1-Score 136
5. Deployment 140
5.1. Deployment Interface 141
6. Conclusion 151
7. Reference 152

1. Executive Summary

This project applies the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology to develop a predictive analytics solution using Python. The objective is to identify a real-world business problem involving classification or regression, implement appropriate data preprocessing and machine learning techniques, and generate insights that support data-driven decision-making. Specifically, the project focuses on bankruptcy prediction, a critical task in financial risk management that enables early identification of distressed firms and supports stakeholders in making informed strategic, investment, and regulatory decisions.

The dataset used for this analysis is the **Taiwanese Bankruptcy Prediction dataset**, obtained from the Taiwan Economic Journal and covering financial data from 1999 to 2009. It contains 6,819 company records, each described by 95 financial indicators, including liquidity ratios, profitability measures, leverage ratios, operational efficiency metrics, cash flow indicators, and growth variables. The target variable is a binary classification label indicating whether a company went bankrupt based on standards from the Taiwan Stock Exchange. The high dimensionality and mixed financial attributes of this dataset provide a comprehensive foundation for building a robust predictive model.

2. Business Understanding

2.1. Business Objective

The objective of this project is to develop a machine learning model that can accurately predict whether a company is at risk of bankruptcy using historical financial indicators. Early detection of financial distress is critical for investors, financial institutions, regulators, and company management, as it enables timely interventions, reduces financial losses, and supports more informed decision-making. By leveraging the Taiwanese Bankruptcy dataset, this project aims to uncover the financial patterns associated with corporate failure and build a predictive system that can function as an early-warning tool for business stakeholders.

2.2. Key Stakeholders

1. **Financial Institutions**

    1.1. Banks, lenders, and credit providers who rely on bankruptcy prediction systems to assess creditworthiness, manage lending risks, and determine loan approval terms.

2. **Investors and Shareholders**

    2.1. Individuals and institutions seeking to minimize financial losses by identifying companies with declining financial health.

3. **Corporate Management and Executives**

    3.1. Internal stakeholders who can use predictive insights to make strategic decisions, improve financial stability, and implement early corrective actions.

4. **Regulatory Bodies and Government Agencies**

    4.1. Organizations responsible for safeguarding market stability and enforcing financial transparency standards.

5. **Data Scientists and Risk Analysts**

    5.1. Professionals tasked with developing and maintaining predictive models that support risk assessment workflows.

2.3. Project Goals

- Build a classification model using the Taiwanese Bankruptcy dataset to predict whether a company is likely to go bankrupt.

- Identify key financial indicators which include liquidity ratios, profitability metrics, leverage ratios, and cash flow data that contribute most significantly to bankruptcy risk.

- Achieve strong model performance, prioritizing accuracy, precision, and recall due to the imbalanced nature of bankruptcy cases.

- Develop an interpretable and reliable system that stakeholders can trust and use for decision-making in real-world financial environments.

- Provide actionable insights that support financial risk assessment, early intervention, and strategic planning.

2.4. Success Criteria

1. **Model Performance:**

    1.1. The predictive model should achieve high accuracy and strong recall for bankruptcy cases, ensuring that high-risk companies are correctly identified.

    1.2. Performance benchmarks may include:

        1.2.1. Accuracy $\geq 80\%$

        1.2.2. AUC ≥ 0.90

        1.2.3. Balanced precision and recall

2. **Business Interpretability:**

    2.1. Stakeholders should be able to understand why the model makes certain predictions, supported by feature importance analysis or explainable AI methods.

3. **Practical Relevance:**

    3.1. The model should provide valuable insights into financial health indicators that influence bankruptcy outcomes.

4. **Scalability and Integration:**

    4.1. The system should be scalable for deployment in risk management departments and financial analytic platforms.

2.5. Significance and Potential Business Impact

Predicting bankruptcy has major implications for financial risk management. An early-warning predictive system can:

- **Reduce financial losses** for banks and investors by identifying at-risk firms before collapse.
- **Improve credit assessment processes**, leading to better loan decisions and interest rate adjustments.
- **Support regulators** in monitoring systemic risks within the financial market.
- **Enable companies** to respond proactively, improving financial restructuring, budgeting, and resource allocation.
- **Enhance market stability**, as early detection reduces the likelihood of cascading business failures.

In a highly competitive economic environment, the ability to anticipate financial distress offers organizations a significant strategic advantage.

2.6. Scope of the Project

- Data preprocessing, feature engineering, and handling of class imbalance.
- Exploration of multiple machine learning models (e.g., logistic regression, decision trees, random forest, boosting algorithms).
- Model evaluation using standard classification metrics.
- Interpretation of results, including identification of key financial risk indicators.
- Development of insights and recommendations for stakeholders.

2.7. Assumptions

- The dataset is representative of real-world financial conditions among Taiwanese companies.
- Financial indicators included in the dataset are reliable and accurately recorded.
- Stakeholders will be able to integrate predictive insights into credit assessment and risk management workflows.

2.8. Risks

1. **Data Limitation:**
   1.1. The dataset contains 96 attributes, but only 43 are commonly used in published studies. Careful feature selection will be needed to optimize model performance.
2. **Data Quality Risk:**

   2.1. Incomplete or inconsistent financial data may affect model accuracy.

3. **Model Performance Risk:**

   3.1. Due to severe class imbalance, the model may struggle to detect rare bankruptcy cases.

4. **Interpretability Risk:**

4.1. Complex models may achieve high performance but be difficult for financial stakeholders to interpret.

5. **Misuse Risk:**

5.1. Predictions could be used unfairly in credit decisions or investment practices if not regulated properly.

4. Data Description

The Taiwanese Bankruptcy Prediction dataset used in this project contains comprehensive financial information about companies listed in Taiwan between 1999 and 2009. The primary objective of this dataset is to determine whether a company is at risk of bankruptcy based on its financial ratios, performance indicators, and profitability metrics. This dataset is suitable for binary classification, where the target variable indicates whether a company is **bankrupt (1)** or **non-bankrupt (0)**.(UCI Machine Learning Repository, 2020)

The dataset consists of the following:

1. **Number of Instances (rows):** 6,819
2. **Number of Attributes (columns):** 96 (including the target variable)
3. **Target Variable:**
    - **Bankrupt?**
        - **1 = Bankrupt**
        - **0 = Not Bankrupt**

The remaining 39 attributes represent a wide range of financial ratios and business performance metrics. These features cover profitability, liquidity, leverage, efficiency, cash flow, operational performance, growth indicators, and risk-related financial ratios. This rich variety of attributes provides a strong foundation for building predictive models and understanding the financial patterns associated with corporate bankruptcy.

4.1. Attribute Information

The dataset includes the following major categories of financial features:

4.1.1. Profitability Ratios

These indicators measure a company's ability to generate profit relative to its assets, equity, revenue, and capital structure.

Examples include:

1. **ROA(A) before interest and % after tax**

   Measures operating efficiency using *after-tax income* but excludes interest expense to remove capital structure effects.

2. **ROA(B) before interest and depreciation after tax**

   Uses earnings before interest and tax (EBIT), sometimes adjusting for depreciation, to reflect core operations.

3. **ROA(C) before interest and depreciation, before tax**

   Uses EBITDA over total assets to capture operational cash-based profitability.

4. **Operating Gross Margin** measures how much profit a company retains after covering the direct costs of production (e.g., materials, labor), before accounting for operating expenses.It reflects a firm's ability to generate earnings to cover financing costs.

   Formula : Operating Gross Margin = (Sales -COGS) / Sales

5. **Operating Profit Rate** measures the proportion of revenue left after paying for operating expenses but before interest and tax.It evaluates how efficiently the firm manages its daily operations.

   Formula : Operating Profit Rate = Operating Income/ Sales

6. **Pre-tax Net Interest Rate** is an indicator that evaluates the return generated relative to interest-bearing liabilities before taxes.It reflects a firm's ability to generate earnings to cover financing costs.

Formula: Pre-tax Net Interest Rate = Pre-tax Net Income/ Interest- bearing Debt

7. **After-tax Net Interest Rate** uses after-tax net income, showing the real return from operations after tax obligations.

   Formula: After-tax Net Interest Rate = After-tax Net Income/ Interest- bearing Debt

8. **Persistent EPS in the Last Four Seasons** is average earnings per share over the last four quarters.It evaluates the sustainability and stability of profits.

   Formula : Persistent EPS = $\sum_{i=1}^{4} \frac{(EPS)i}{4}$

9. **Operating Profit Per Share (Yuan ¥)** measures how much operating income is earned for each outstanding share.It indicates operational profitability on a per-share basis.

   Formula : Operating Profit per share = Operating Income/ Number of share
   It Indicates operational profitability on a per-share basis.

10. **Per Share Net profit before tax (Yuan ¥)** shows the pre-tax profitability available to each shareholder. It assesses performance without tax effects.

    Formula : Net Profit Before Tax Per Share = Pre-tax Income/ Shares Outstanding

11. **Gross Profit to Sales** measures how much gross profit is generated per unit of sales. It reflects production efficiency and pricing strategy.

    Formula : Gross Profit to Sales = Gross Profit / sales

12. **Net Income to Stockholder's Equity.** Also called Return on Equity (ROE), it shows how effectively the company uses shareholders' capital to generate profit. Higher ROE indicates efficient profit generation relative to equity.

Formula: ROE = Net income / Shareholders' Equity

4.1.2. Liquidity Ratios

These metrics measure a firm's short-term financial stability and ability to meet immediate obligations. Examples include:

13. **Current Ratio** measures a firm's ability to pay short-term obligations using current assets.Higher values indicate better short-term liquidity.

    Formula : Current Ratio = Current Assets/ Current Liabilities

14. **Quick Ratio** Evaluates the ability to meet short-term liabilities using highly liquid assets (excludes inventory).

    Formula : Quick Ratio = (Current Assets−Inventory) / Current Liabilities

15. **Cash/Total Assets** Assesses how much of total assets exist in cash or cash equivalents. Indicates liquidity strength and cash buffer

    Formula : Cash to Total Assets = Cash / Total Assets

16. **Quick Assets/Current Liability** measures the coverage of current liabilities using liquid assets (cash, marketable securities, receivables).

    Formula : Quick Assets Ratio = Quick Assets/ Current Liabilities

17. **Cash/Current Liability** evaluates how much of short-term obligations can be paid immediately using cash.

    Formula : Cash Ratio = Cash / Current Liabilities

4.1.3. Leverage and Debt Ratios

These indicators capture long-term solvency and bankruptcy risk by examining the firm's dependency on borrowed funds. Higher ratios reflect higher financial risk.

Examples include:

18. **Total Debt/Total Net Worth** indicates the proportion of total debt relative to shareholders' equity.

    Formula : Debt to net worth = Total Debt / Equity

19. **Debt Ratio %** measures the percentage of assets financed by liabilities.

    Formula : Debt Ratio = Total Liabilities/ Total Assets

20. **Net Worth/Assets** shows the proportion of assets financed by equity (inverse of debt ratio).

    Formula = Equity Ratio = Equity/ Total Assets

21. **Liability to Equity** evaluates solvency and capital structure risk.

    Formula : Liability to Equity = Total Liability/ Equity

22. **Degree of Financial Leverage (DFL)** measures sensitivity of net income to changes in operating income due to fixed financial costs (interest).Higher DFL = higher financial risk.

    Formula : DFL = Percentage Change in Net Income/ Percentage Change in EBIT

23. **Long-term Fund Suitability Ratio (A)** evaluates whether long-term funds (equity + long-term debt) are sufficient to finance long-term assets. Ratio ≥ 1 indicates healthy long-term financing.

    Formula : Long-Term Fund Suitability Ratio = Long-term Funds/ Fixed Assets

4.1.4. Cash Flow Ratios

These features assess a company's cash management efficiency and its ability to generate cash relative to its operations and liabilities.

Examples include:

24. **Cash Flow Rate** measure showing the proportion of operating cash flow relative to total assets, used to assess the company's ability to generate cash from its asset base. It indicates financial flexibility and internal liquidity strength.

    Formula : Cash Flow Rate = Operating Cash Flow(CFO)/ Total Assets

25. **Cash Flow Per Share** represents the amount of operating cash flow generated for each outstanding share. This measure is often considered a more accurate indicator of earnings quality than earnings per share (EPS).

    Formula : Cash Flow per Share = Operating Cash Flow / Number Of Shares Outstanding

26. **Cash Flow to Sales** measures how much cash flow is generated from each unit of revenue.A higher ratio indicates strong cash-generation efficiency from sales operations.

    Formula : Cash Flow to sales = Operating cash flow/ net sales

27. **CFO to Assets** call operating cash flow to total assets, this ratio examines how effectively assets generate operating cash flow. It is a useful indicator of long-term financial health.

    Formula : CFO to assets = operating cash flow / total assets

28. **Cash Flow to Equity** Measures the cash available to equity holders after satisfying all expenses, taxes, and debt-related payments.Often used in valuation, especially in discounted cash flow (DCF) models.

    Formula : Cash flow to equity = cash flow available to equity/ equity

29. **Cash Flow to Liability** used in valuation, especially in discounted cash flow (DCF) models. Often used in valuation, especially in discounted cash flow (DCF) models.

    Formula : Cash Flow to liability = Operating cash flow/ Total Liabilities

4.1.5. Operational Efficiency Ratios

These metrics evaluate how effectively a company uses its assets and resources to generate revenue. Examples include:

30. **Accounts Receivable Turnover** measures how quickly receivables are collected.

    Formula : AR Turnover = Net Credit Sales / Accounts Receivable

31. **Inventory Turnover Rate (times)** shows how often cycles throughout the business.

    Formula : Inventory Turnover = Cost of Goods Sold(COGS)/ Average inventory

32. **Fixed Assets Turnover Frequency** measures how efficiently fixed assets generate revenue.

    Formula : Fixed asset turnover= Net Sales / Net Fixed Assets

33. **Cash Turnover Rate** shows how effectively cash is used to generate revenue.

    Formula : Cash TurnOver = Net Sales/ Cash

4.1.6. Growth Indicators

These attributes measure year-over-year improvement or decline in revenue, profit, equity, or assets. Examples include:

34. **Operating Profit Growth Rate** measures year-over-year growth in operating profit.

    Formula : Operating Profit Growth = (Operating $Profit_t$ - Operating $Profit_{t-1}$)/ Operating $Profit_{t-1}$

35. **After-tax Net Profit Growth Rate** measures growth in net income after taxes.

    Formula : Net Profit Growth = $(ROA_t - ROA_{t-1})/ _{t-1}$

36. **Net Value Growth Rate** assesses growth in shareholders' equity (net worth).

    Formula : Net Value Growth = $(Equity_t - Equity_{t-1})/ Equity_{t-1}$

37. **Total Asset Growth Rate** measures year-over-year change in total assets.

    Formula : Asset Growth= $(Total\ Assets_t - Total\ Assets_{t-1})/ Total\ Assets_{t-1}$

4.1.7. Risk and Flag Indicators

These binary variables indicate critical negative financial conditions.
Examples include:

38. **Liability-Assets Flag**

    A binary indicator signaling financial distress:

    - **1** = liabilities exceed assets (negative equity)
    - **0** = assets exceed liabilities

    This flag indicates potential insolvency.

39. **Net Income Flag**

    Identifies firms with consecutive years of losses:

    - **1** = net income is negative for two consecutive years
    - **0** = otherwise

    Used as an early-warning signal for financial instability.

4.2. Data Processing

Before analyzing and modeling the dataset, several preprocessing steps are necessary to ensure data quality and compatibility with machine learning algorithms:

1. Show a few rows of data from the head of data.

Bankrupt?	ROA(B) before interest and depreciation after tax	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	Operating Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Persistent EPS in the Last Four Seasons	Operating Profit Per Share (Yuan ¥)	Per Share Net profit before tax (Yuan ¥)	Gross Profit to Sales	Net Income to Stockholder's Equity	Current Ratio	Quick Ratio	Cash/Total Assets
1	0.405750	0.370594	0.424389	0.601457	0.998969	0.796887	0.808809	0.169141	0.095921	0.138736	0.601453	0.827890	0.002259	0.001208	0.004094
1	0.516730	0.464291	0.538214	0.610235	0.998946	0.797380	0.809301	0.208944	0.093722	0.169918	0.610237	0.839969	0.006016	0.004039	0.014948
1	0.472295	0.426071	0.499019	0.601450	0.998857	0.796403	0.808388	0.180581	0.092338	0.142803	0.601449	0.836774	0.011543	0.005348	0.000991
1	0.457733	0.399844	0.451265	0.583541	0.998700	0.796967	0.808966	0.193722	0.077762	0.148603	0.583538	0.834697	0.004194	0.002896	0.018851
1	0.522298	0.465022	0.538432	0.598783	0.998973	0.797366	0.809304	0.212537	0.096898	0.168412	0.598782	0.839973	0.006022	0.003727	0.014161

Figure 4.2.1 show head of data from selected features (1 of 3)

Quick Assets/Current Liability	Cash/Current Liability	Total debt/Total net worth	Debt ratio %	Net worth/Assets	Liability to Equity	Degree of Financial Leverage (DFL)	Long-term fund suitability ratio (A)	Cash flow rate	Cash Flow Per Share	Cash Flow to Sales	CFO to Assets	Cash Flow to Equity	Cash Flow to Liability	Accounts Receivable Turnover
0.001997	1.473360e-04	0.021266	0.207576	0.792424	0.290202	0.026601	0.005024	0.458143	0.311664	0.671568	0.520382	0.312905	0.458609	0.001814
0.004136	1.383910e-03	0.012502	0.171176	0.828824	0.283846	0.264577	0.005059	0.461867	0.318137	0.671570	0.567101	0.314163	0.459001	0.001286
0.006302	5.340000e+09	0.021248	0.207516	0.792484	0.290189	0.026555	0.005100	0.458521	0.307102	0.671571	0.538491	0.314515	0.459254	0.001495
0.002961	1.010646e-03	0.009572	0.151465	0.848535	0.281721	0.026697	0.005047	0.465705	0.321674	0.671519	0.604105	0.302382	0.448518	0.001966
0.004275	6.804636e-04	0.005150	0.106509	0.893491	0.278514	0.024752	0.005303	0.462746	0.319162	0.671563	0.578469	0.311567	0.454411	0.001449

Figure 4.2.2 show head of data from selected features (2 of 3)

Inventory Turnover Rate (times)	Fixed Assets Turnover Frequency	Cash Turnover Rate	Operating Profit Rate	After-tax Net Profit Growth Rate	Net Value Growth Rate	Total Asset Growth Rate	Liability-Assets Flag	Net Income Flag
1.820926e-04	1.165007e-04	4.580000e+08	0.998969	0.688979	0.000327	4.980000e+09	0	1
9.360000e+09	7.190000e+08	2.490000e+09	0.998946	0.689693	0.000443	6.110000e+09	0	1
6.500000e+07	2.650000e+09	7.610000e+08	0.998857	0.689463	0.000396	7.280000e+09	0	1
7.130000e+09	9.150000e+09	2.030000e+09	0.998700	0.689110	0.000382	4.880000e+09	0	1
1.633674e-04	2.935211e-04	8.240000e+08	0.998973	0.689697	0.000439	5.510000e+09	0	1

Figure 4.2.3 show head of data from selected features (3 of 3)

2. Description of data

	Bankrupt?	ROA(B) before interest and depreciation after tax	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	Operating Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Persistent EPS in the Last Four Seasons	Operating Profit Per Share (Yuan ¥)	Per Share Net profit before tax (Yuan ¥)	Gross Profit to Sales	Net Income to Stockholder's Equity
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000
mean	0.032263	0.553589	0.505180	0.558625	0.607948	0.998755	0.797190	0.809084	0.228813	0.109091	0.184361	0.607946	0.840402
std	0.176710	0.061595	0.060686	0.065620	0.016934	0.013010	0.012869	0.013601	0.033263	0.027942	0.033180	0.016934	0.014523
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.527277	0.476527	0.535543	0.600445	0.998969	0.797386	0.809312	0.214711	0.096083	0.170370	0.600443	0.840115
50%	0.000000	0.552278	0.502706	0.559802	0.605997	0.999022	0.797464	0.809375	0.224544	0.104226	0.179709	0.605998	0.841179
75%	0.000000	0.584105	0.535563	0.589157	0.613914	0.999095	0.797579	0.809469	0.238820	0.116155	0.193493	0.613913	0.842357
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 4.2.4 show overview of data(1 of 4)

Current Ratio	Quick Ratio	Cash/Total Assets	Quick Assets/Current Liability	Cash/Current Liability	Total debt/Total net worth	Debt ratio %	Net worth/Assets	Liability to Equity	Degree of Financial Leverage (DFL)	Long-term fund suitability ratio (A)	Cash flow rate
6.819000e+03	6.819000e+03	6819.000000	6.819000e+03	6.819000e+03	6.819000e+03	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000
4.032850e+05	8.376595e+06	0.124095	3.592902e+06	3.715999e+07	4.416337e+06	0.113177	0.886823	0.280365	0.027541	0.008783	0.467431
3.330216e+07	2.446847e+08	0.139251	1.716209e+08	5.103509e+08	1.684069e+08	0.053920	0.053920	0.014463	0.015668	0.028153	0.017036
0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7.555047e-03	4.725903e-03	0.033543	5.239776e-03	1.973008e-03	3.007049e-03	0.072891	0.851196	0.276944	0.026791	0.005244	0.461558
1.058717e-02	7.412472e-03	0.074887	7.908898e-03	4.903886e-03	5.546284e-03	0.111407	0.888593	0.278778	0.026808	0.005665	0.465080
1.626953e-02	1.224911e-02	0.161073	1.295091e-02	1.280557e-02	9.273293e-03	0.148804	0.927109	0.281449	0.026913	0.006847	0.471004
2.750000e+09	9.230000e+09	1.000000	8.820000e+09	9.650000e+09	9.940000e+09	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 4.2.5 show overview of data(2 of 4)

Cash Flow Per Share	Cash Flow to Sales	CFO to Assets	Cash Flow to Equity	Cash Flow to Liability	Accounts Receivable Turnover	Inventory Turnover Rate (times)	Fixed Assets Turnover Frequency	Cash Turnover Rate	Operating Profit Rate	After-tax Net Profit Growth Rate	Net Value Growth Rate	Total Asset Growth Rate
6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6.819000e+03	6.819000e+03	6.819000e+03	6.819000e+03	6819.000000	6819.000000	6.819000e+03	6.819000e+03
0.323482	0.671531	0.593415	0.315582	0.461849	1.278971e+07	2.149106e+09	1.008596e+09	2.471977e+09	0.998755	0.689146	1.566212e+06	5.508097e+09
0.017611	0.009341	0.058561	0.012961	0.029943	2.782598e+08	3.247967e+09	2.477557e+09	2.938623e+09	0.013010	0.013853	1.141594e+08	2.897718e+09
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000e+00	0.000000e+00
0.317748	0.671565	0.565987	0.312995	0.457116	7.101336e-04	1.728256e-04	2.330013e-04	2.735337e-04	0.998969	0.689270	4.409689e-04	4.860000e+09
0.322487	0.671574	0.593266	0.314953	0.459750	9.678107e-04	7.646743e-04	5.930942e-04	1.080000e+09	0.999022	0.689439	4.619555e-04	6.400000e+09
0.328623	0.671587	0.624769	0.317707	0.464236	1.454759e-03	4.620000e+09	3.652371e-03	4.510000e+09	0.999095	0.689647	4.993621e-04	7.390000e+09
1.000000	1.000000	1.000000	1.000000	1.000000	9.740000e+09	9.990000e+09	9.990000e+09	1.000000e+10	1.000000	1.000000	9.330000e+09	9.990000e+09

Figure 4.2.6 show overview of data(3 of 4)

Liability-Assets Flag	Net Income Flag
6819.000000	6819.0
0.001173	1.0
0.034234	0.0
0.000000	1.0
0.000000	1.0
0.000000	1.0
0.000000	1.0
1.000000	1.0

Figure 4.2.7 show overview of data(4 of 4)

The figure 4.2.4 to 4.2.7 above shows a quick overview of our dataset. We can see the

- Count (Number of non-null values in each attribute)
- Mean
- Standard deviation
- Minimum value
- First Quartile
- Median
- Third Quartile
- Maximum value

Of each column.

3. Information about data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6819 entries, 0 to 6818
Data columns (total 40 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   Bankrupt?                                               6819 non-null   int64
 1   ROA(B) before interest and depreciation after tax       6819 non-null   float64
 2   ROA(C) before interest and depreciation before interest 6819 non-null   float64
 3   ROA(A) before interest and % after tax                  6819 non-null   float64
 4   Operating Gross Margin                                  6819 non-null   float64
 5   Operating Profit Rate                                   6819 non-null   float64
 6   Pre-tax net Interest Rate                               6819 non-null   float64
 7   After-tax net Interest Rate                             6819 non-null   float64
 8   Persistent EPS in the Last Four Seasons                 6819 non-null   float64
 9   Operating Profit Per Share (Yuan ¥)                     6819 non-null   float64
 10  Per Share Net profit before tax (Yuan ¥)                6819 non-null   float64
 11  Gross Profit to Sales                                   6819 non-null   float64
 12  Net Income to Stockholder's Equity                      6819 non-null   float64
 13  Current Ratio                                           6819 non-null   float64
 14  Quick Ratio                                             6819 non-null   float64
 15  Cash/Total Assets                                       6819 non-null   float64
 16  Quick Assets/Current Liability                          6819 non-null   float64
 17  Cash/Current Liability                                  6819 non-null   float64
 18  Total debt/Total net worth                              6819 non-null   float64
 19  Debt ratio %                                            6819 non-null   float64
 20  Net worth/Assets                                        6819 non-null   float64
 21  Liability to Equity                                     6819 non-null   float64
 22  Degree of Financial Leverage (DFL)                      6819 non-null   float64
 23  Long-term fund suitability ratio (A)                    6819 non-null   float64
 24  Cash flow rate                                          6819 non-null   float64
 25  Cash Flow Per Share                                     6819 non-null   float64
 26  Cash Flow to Sales                                      6819 non-null   float64
 27  CFO to Assets                                           6819 non-null   float64
 28  Cash Flow to Equity                                     6819 non-null   float64
 29  Cash Flow to Liability                                  6819 non-null   float64
 30  Accounts Receivable Turnover                            6819 non-null   float64
 31  Inventory Turnover Rate (times)                         6819 non-null   float64
 32  Fixed Assets Turnover Frequency                         6819 non-null   float64
 33  Cash Turnover Rate                                      6819 non-null   float64
 34  Operating Profit Rate                                   6819 non-null   float64
 35  After-tax Net Profit Growth Rate                        6819 non-null   float64
 36  Net Value Growth Rate                                   6819 non-null   float64
 37  Total Asset Growth Rate                                 6819 non-null   float64
 38  Liability-Assets Flag                                   6819 non-null   int64
 39  Net Income Flag                                         6819 non-null   int64
dtypes: float64(37), int64(3)
memory usage: 2.1 MB
```

Figure 4.2.8 information of data type

The Figure 4.2.8 above shows some basic information about the data. Such as the data type of

each attribute and the non-null count.

- All the attributes in the dataset are of type integer or float.
- Attributes do not have missing values.

4.3. Exploratory Data Analysis (EDA):

EDA is a crucial step to gain insights from the dataset and understand the relationships between variables.

4.3.1. Target distribution

The distribution of the target variable, especially in classification tasks to ensure that the model

makes accurate predictions, generalizes well, and avoids bias toward certain classes.

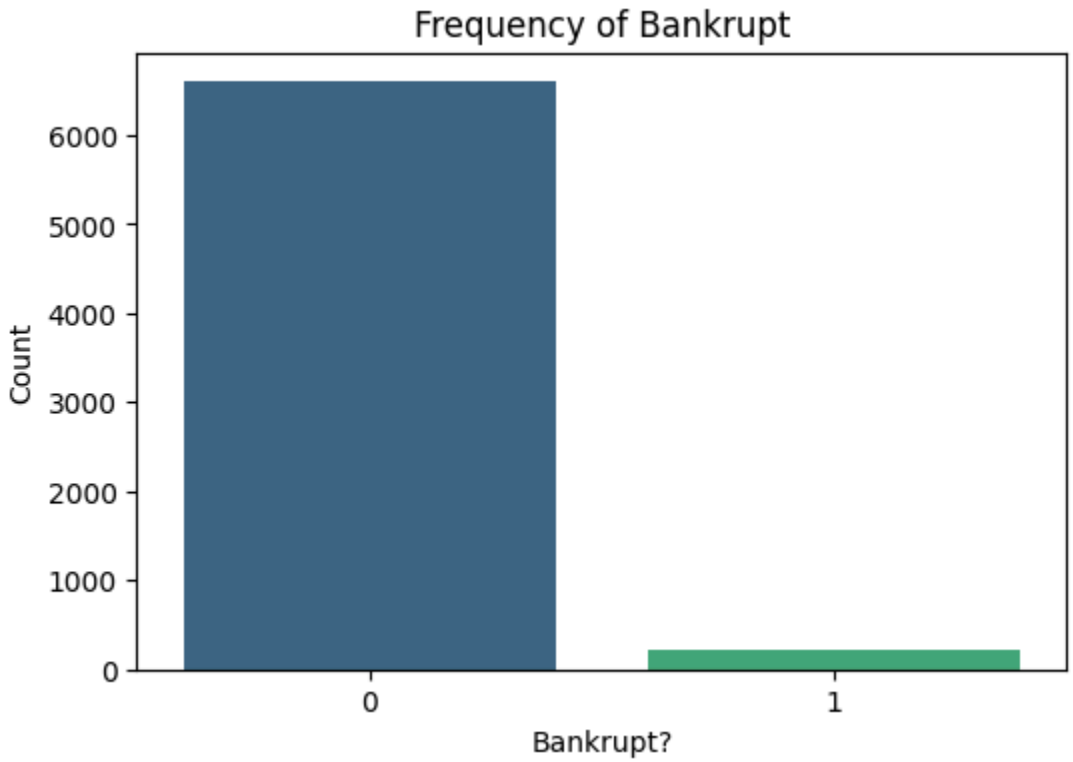


Figure 4.3.1.1 Bar Chart count number of bankrupt and non bankrupt company

```
Percentage of Bankrupt: Bankrupt?
0    0.967737
1    0.032263
Name: proportion, dtype: float64
```

Figure 4.3.1.2 Percentage of bankrupt and non bankrupt company

We observe that the target is not fairly balanced with ~96% company non bankrupt and ~3% company bankrupt.

Several imbalance mitigation strategies were explored at the exploratory stage to understand their impact on dataset structure. Random oversampling balances the dataset by duplicating minority class observations, increasing dataset size but introducing duplication risk. Random undersampling achieves balance by removing majority class observations, significantly reducing dataset size and potentially discarding valuable information. Class weighting preserves the original data distribution while accounting for asymmetric misclassification costs. A combined oversampling and undersampling approach was further evaluated to achieve class balance while controlling both duplication and information loss, offering a practical compromise for subsequent modeling stages.

1. Original Dataset (Baseline)

```
Original dataset shape: (6819, 40)
Original class distribution: Counter({0: 6599, 1: 220})
```

Figure 4.3.1.3 Output for original dataset shape

The figure 4.3.1.3 shows that Bankruptcy cases represent **only 3.23%** of the dataset. This Severe imbalance ratio ≈ **1 : 30.** Any analysis without mitigation will be dominated by non-bankrupt firms.The dataset is **highly imbalanced**, and mitigation strategies are required before classification.

2. Solution 1: Random Oversampling (Standalone)

```
--- Solution 1: Oversampling the Minority Class (standalone) ---
Resampled dataset shape (Oversampling): (13198, 39) (13198,)
Class distribution after oversampling: Counter({1: 6599, 0: 6599})
```

Figure 4.3.1.4 Oversamping the minority class output

The figure 4.3.1.4 Minority class (Bankrupt = 1) was **duplicated.** The dataset size **nearly doubled** from 6599 to 13198**.**Perfect class balance achieved

Advantages

- No loss of majority class information

- Minority patterns are equally represented

Limitations

- No new information added
- High risk of duplicated financial records
- Potential overfitting in later modeling

Oversampling improves balance but introduces redundancy and inflates dataset size.

3. Solution 2: Random Undersampling (Standalone)

```
--- Solution 2: Undersampling the Majority Class (standalone) ---
Resampled dataset shape (Undersampling): (440, 39) (440,)
Class distribution after undersampling: Counter({0: 220, 1: 220})
```

Figure 4.3.1.5 Undersampling the minority class output

The figure 4.3.1.5 shows the majority class (Non-bankrupt = 0) was **heavily reduced.** The dataset size dropped by **~94%.**

Advantages

- Perfect balance
- Faster computation

Limitations

- Significant loss of financial data
- Many healthy firms removed
- Risk of missing important trends

Undersampling achieves balance at the cost of substantial information loss.

4. Solution 3: Class Weights (No Resampling)

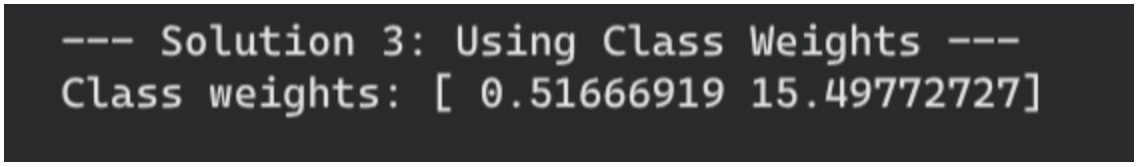


Figure 4.3.1.6 Using class weights

The figure 4.3.6 dataset remains **unchanged.** Bankruptcy misclassification is penalized **~15× more.**

Advantages

- Preserves full dataset
- Reflects asymmetric financial risk
- Aligns with business intelligence logic

Limitations

- Effectiveness depends on model choice
- No visual balance in raw data

This approach provides the best trade-off between balance, data integrity, and dataset size.
Random oversampling achieves class balance by duplicating minority samples but significantly increases dataset size and redundancy. Random undersampling balances classes by removing a large portion of majority samples, resulting in substantial information loss. Class weighting preserves the original data distribution while accounting for asymmetric misclassification costs. A combined oversampling and undersampling strategy offers a practical compromise by achieving balanced class representation while minimizing duplication and information loss, making it a suitable approach for subsequent classification modeling.

4.3.2. Box plots

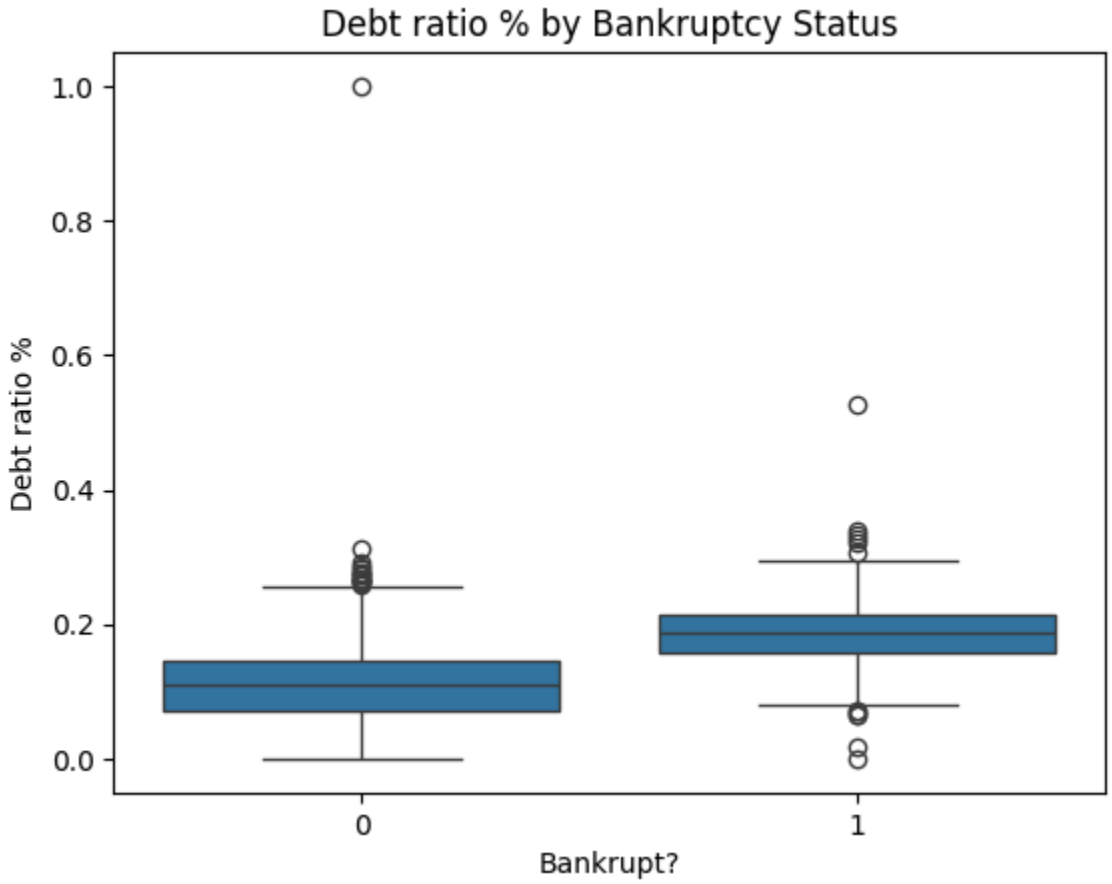


Figure 4.3.2.1 Boxplot debt ratio by bankruptcy status

The figure 4.3.2.1 shows that the bankrupt group (1) has a higher median debt ratio and its box is shifted upward compared to the non-bankrupt group (0), indicating that bankrupt companies generally carry more debt relative to assets. There is overlap between the two boxes, but the overall pattern suggests higher leverage is associated with a greater likelihood of bankruptcy.

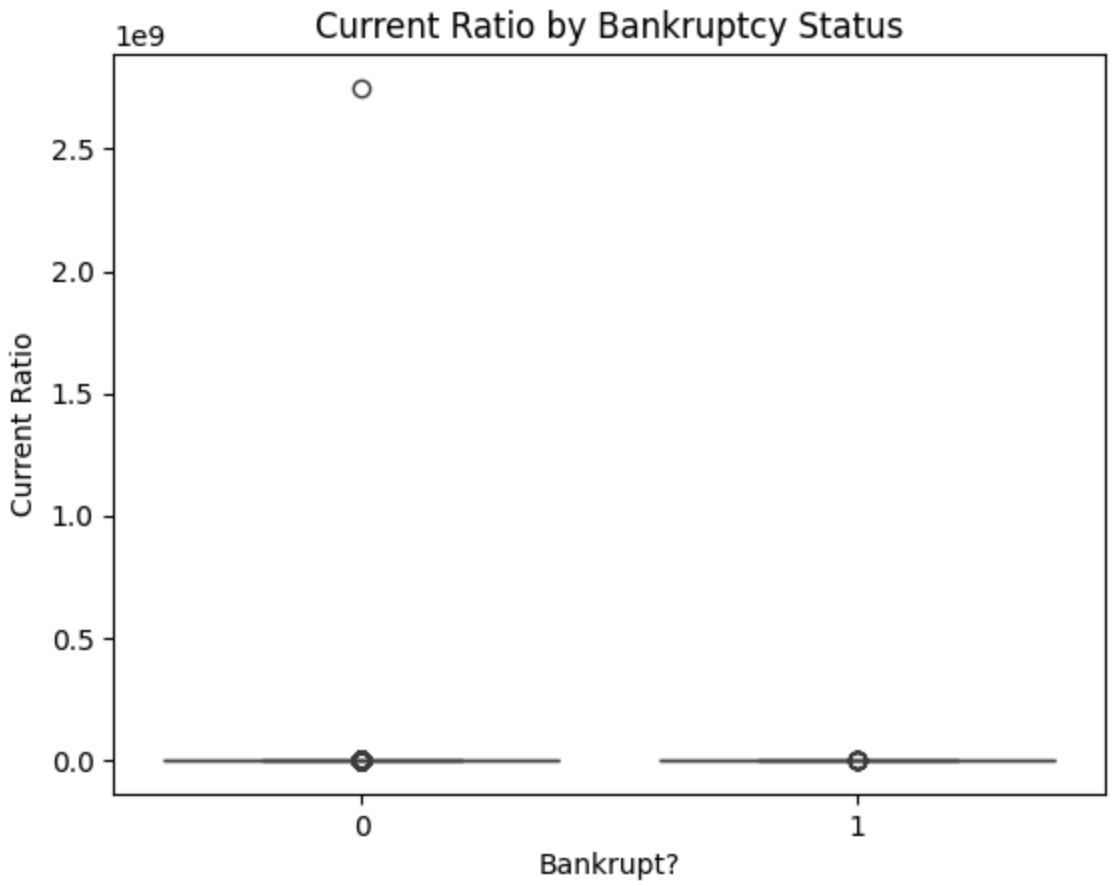


Figure 4.3.2.2 Boxplot Current ratio by bankruptcy status

Both groups (0 = non-bankrupt, 1 = bankrupt) have boxplots compressed near the bottom of the y-axis, indicating that for most firms, current ratios are in a similar and relatively low range.

The medians and interquartile ranges for both groups appear almost overlapping, so the current ratio alone does not clearly separate bankrupt from non-bankrupt firms.

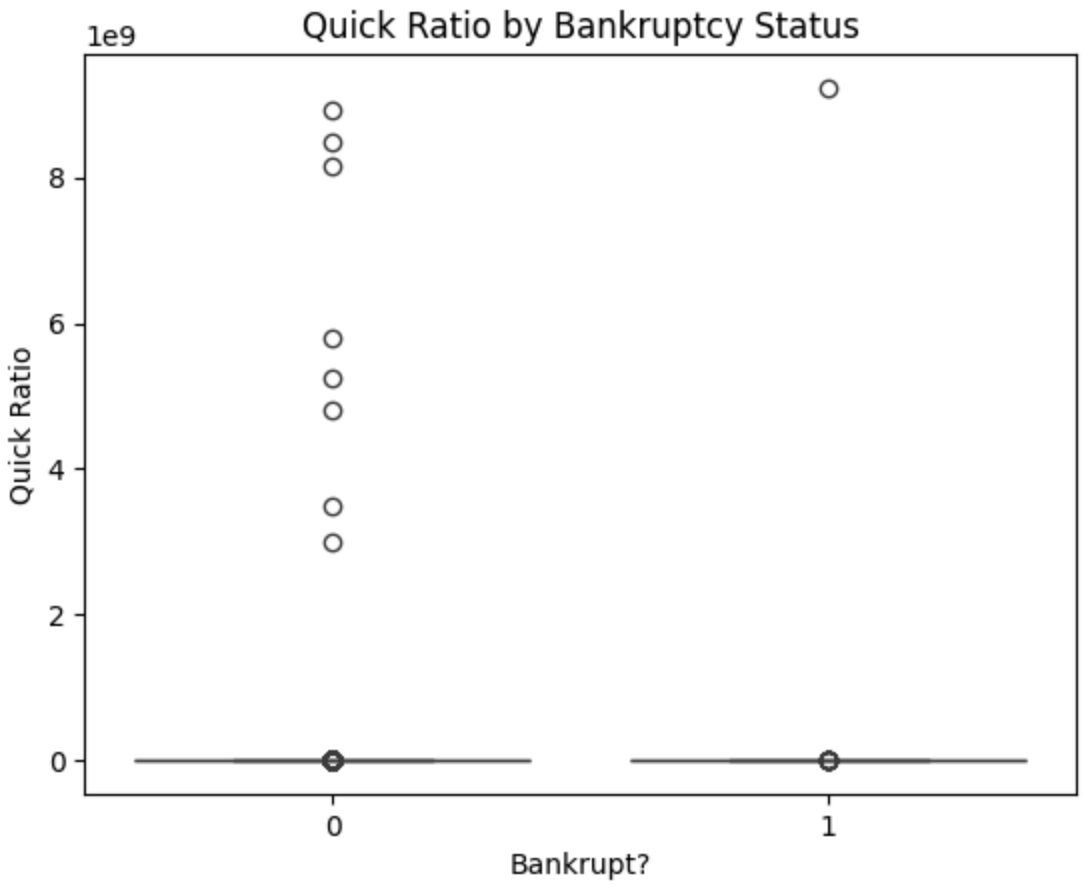


Figure 4.3.2.3 Boxplot Quick ratio by bankruptcy status

Both groups (0 = non-bankrupt, 1 = bankrupt) have their boxplots compressed near zero, indicating that the central quick-ratio values for most firms are low and very similar across bankruptcy status.

The medians and IQRs appear almost overlapping, so typical liquidity as measured by the quick ratio is not markedly different between the two groups in this dataset.

There are many extremely large outliers, especially in the non-bankrupt group and at least one in the bankrupt group, with quick ratios on the order of 10^9 , which stretch the y-axis and visually flatten the main data cluster.

Because of these extreme values, the chart implies data quality issues or very unusual firm cases, and suggests that quick ratio alone is a weak and noisy indicator of bankruptcy risk here.

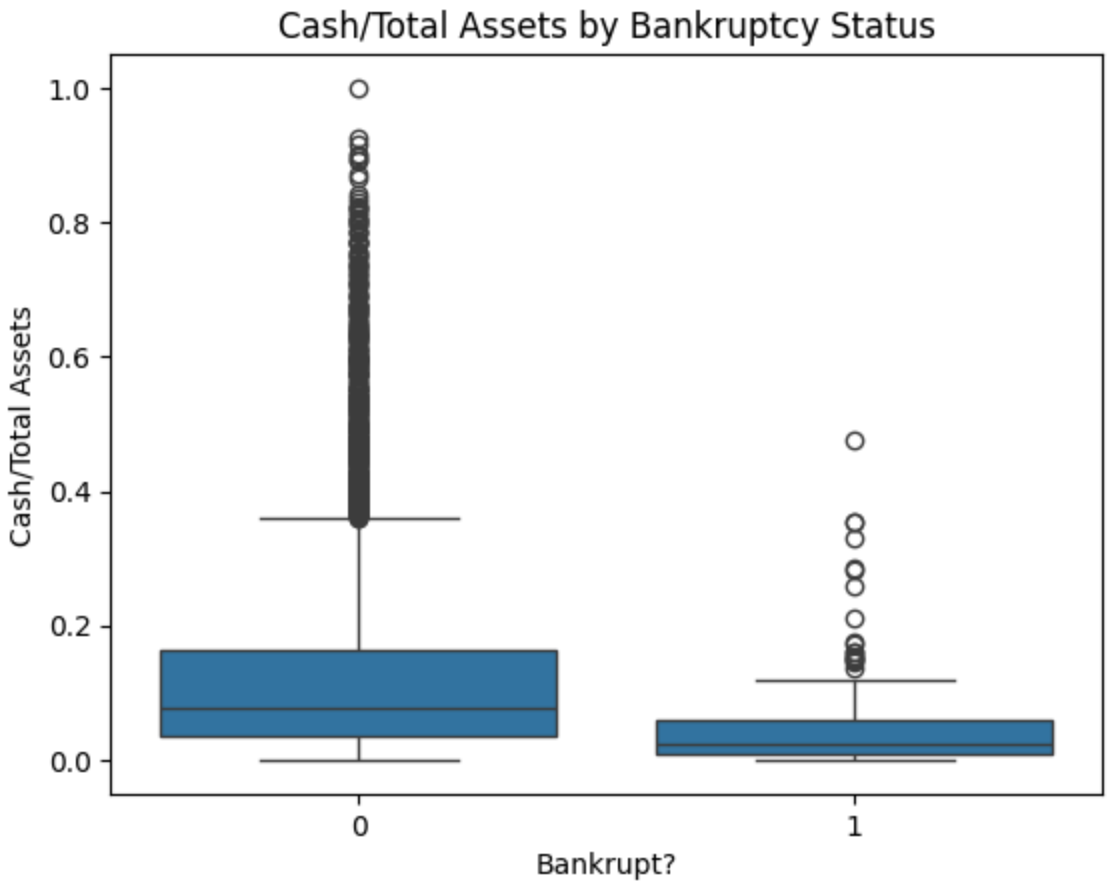


Figure 4.3.2.4 Boxplot cash/ total assets by bankruptcy status

The x-axis shows bankruptcy status (0 = non-bankrupt, 1 = bankrupt), and the y-axis shows the ratio of cash to total assets.

The box for non-bankrupt firms (0) is higher and wider, indicating a higher median cash/asset ratio and more dispersion compared with bankrupt firms (1), whose box is lower and tighter.

Non-bankrupt firms show many outliers with very high cash/asset ratios, some approaching 1, indicating a few firms are extremely cash-rich relative to their assets.

Bankrupt firms have fewer and generally lower outliers, suggesting that, in this dataset, maintaining higher cash buffers relative to total assets may be associated with a lower likelihood of bankruptcy.

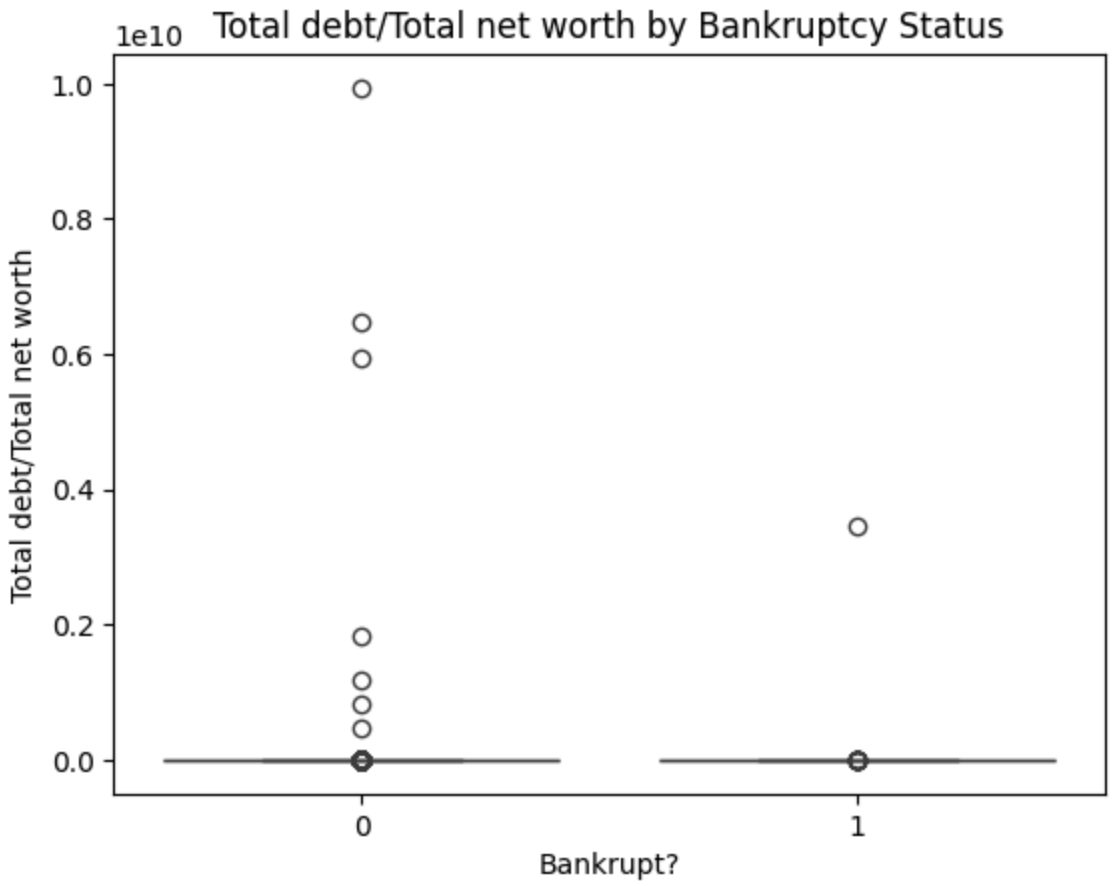


Figure 4.3.2.5 Boxplot Total debt/ total net worth by bankruptcy status

The x-axis shows bankruptcy status (0 = non-bankrupt, 1 = bankrupt), and the y-axis shows total debt divided by total net worth.

Both boxplots are compressed close to zero, with overlapping medians and interquartile ranges, indicating that typical leverage by this metric does not differ much between the two groups.

Outliers and implications

There are several extremely large outliers, especially among non-bankrupt firms (0) and at least one in the bankrupt group (1), with debt-to-net-worth ratios on the order of 10^{10} , which stretches the y-axis and flatten the rest of the data.

These extreme values may reflect data issues or very unusual firms and suggest that, in this dataset, total-debt-to-net-worth alone is not a clean discriminator of bankruptcy risk without handling outliers.

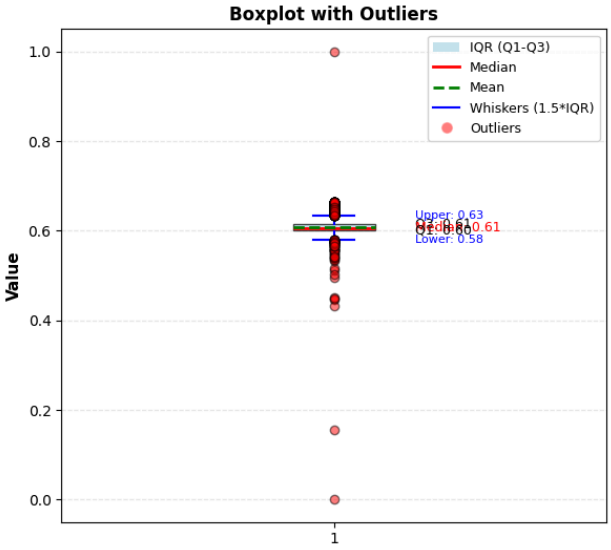


Figure 4.3.2.6 Boxplot Operating Profit Rate

Figure 4.3.2.6 values in Operating Profit Rate are being treated as outliers because the IQR band is extremely narrow and almost the entire distribution lies outside the whisker limits.Q1 and Q3 are both essentially 1 (0.9990 and 0.9991), so the IQR is only 0.0001. The whisker range is 0.9988–0.9993, but the variable actually spans from 0.0000 to 1.0000 with many values far away from that tiny band.

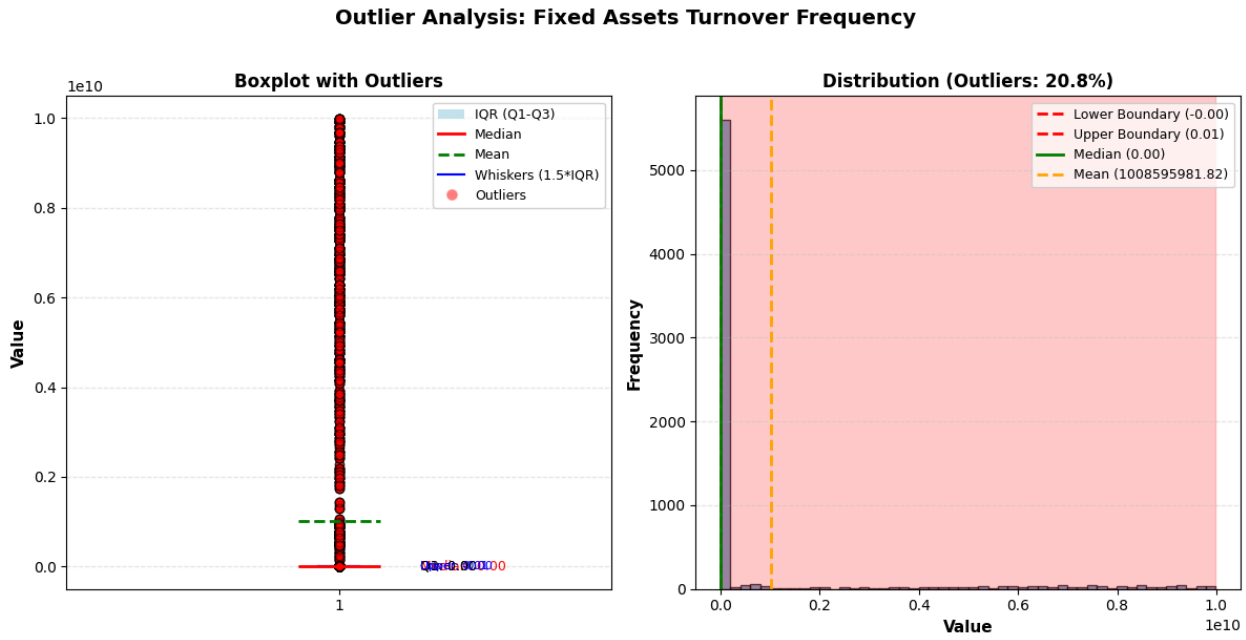


Figure 4.3.2.7 Boxplot and frequency distribution of Fixed Assets turnover

Boxplot view

The median is essentially 0, with a very small IQR near 0–0.01, meaning most firms have very low fixed-asset turnover.A large number of observations shoot up to around 10^{10} , all marked as red outliers, which is economically implausible and indicates calculation or scaling errors rather than genuine business behaviour.

Distribution view

The histogram is heavily concentrated near 0, but about 20.8% of the data lies in the shaded outlier region, and the mean is dragged up to an enormous value (around 1.0×10^{10}), completely misrepresenting the central tendency. Because the distribution is so distorted, this feature should either be heavily cleaned (e.g., capping/removing the huge spikes) or dropped from the model, since the current values will add noise and can destabilize training.

4.3.3. Regression Plot

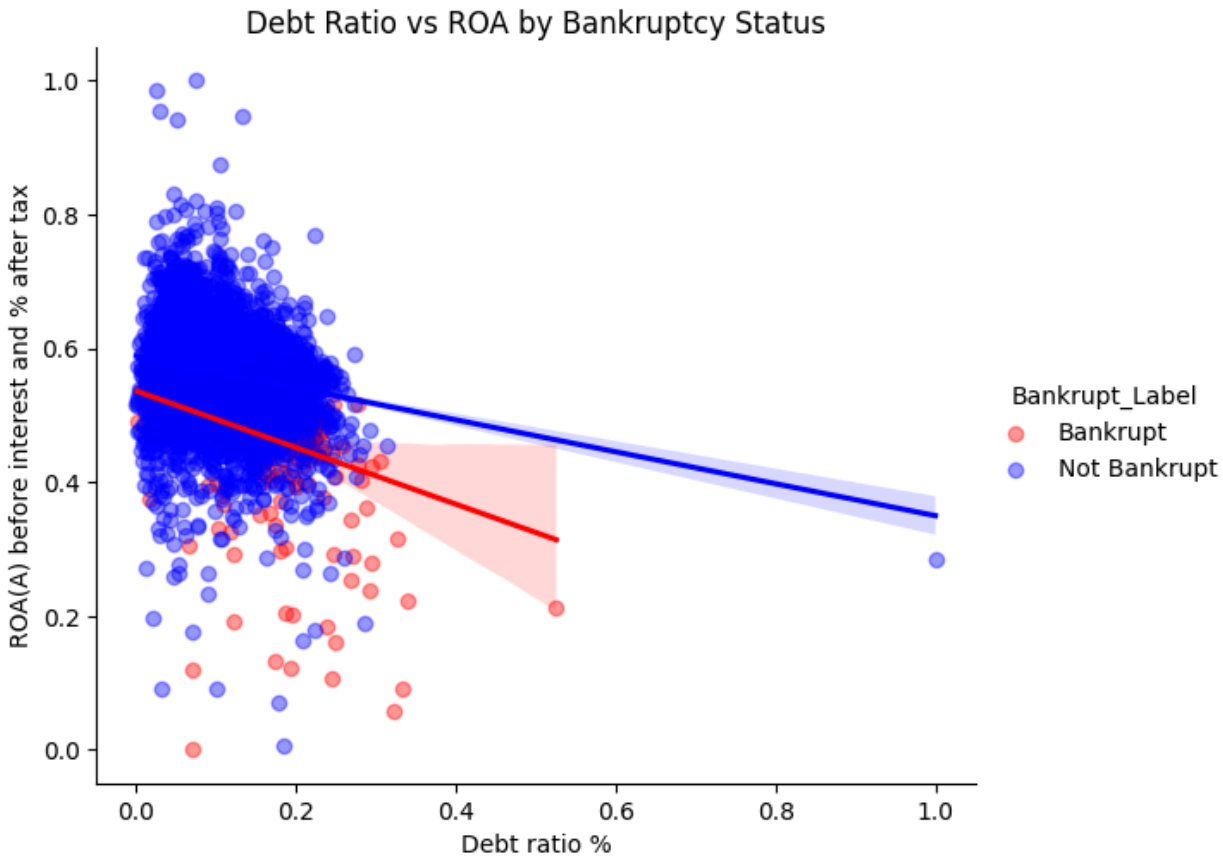


Figure 4.3.3.1 Relationship between ROA(A) before interest and after tax and Debt ratio% by bankruptcy status.

The x-axis is Debt ratio %, and the y-axis is ROA (before interest and after tax).Blue points/line are not-bankrupt firms, and red points/line are bankrupt firms, each with its own fitted regression line.

Patterns and implications

Both regression lines slope downward, indicating a negative correlation: as leverage increases, profitability (ROA) tends to decrease for both groups. The red line for bankrupt firms falls more steeply and sits below the blue line, suggesting that, at a given debt ratio, bankrupt firms generally exhibit lower ROA and are more adversely affected by higher leverage.

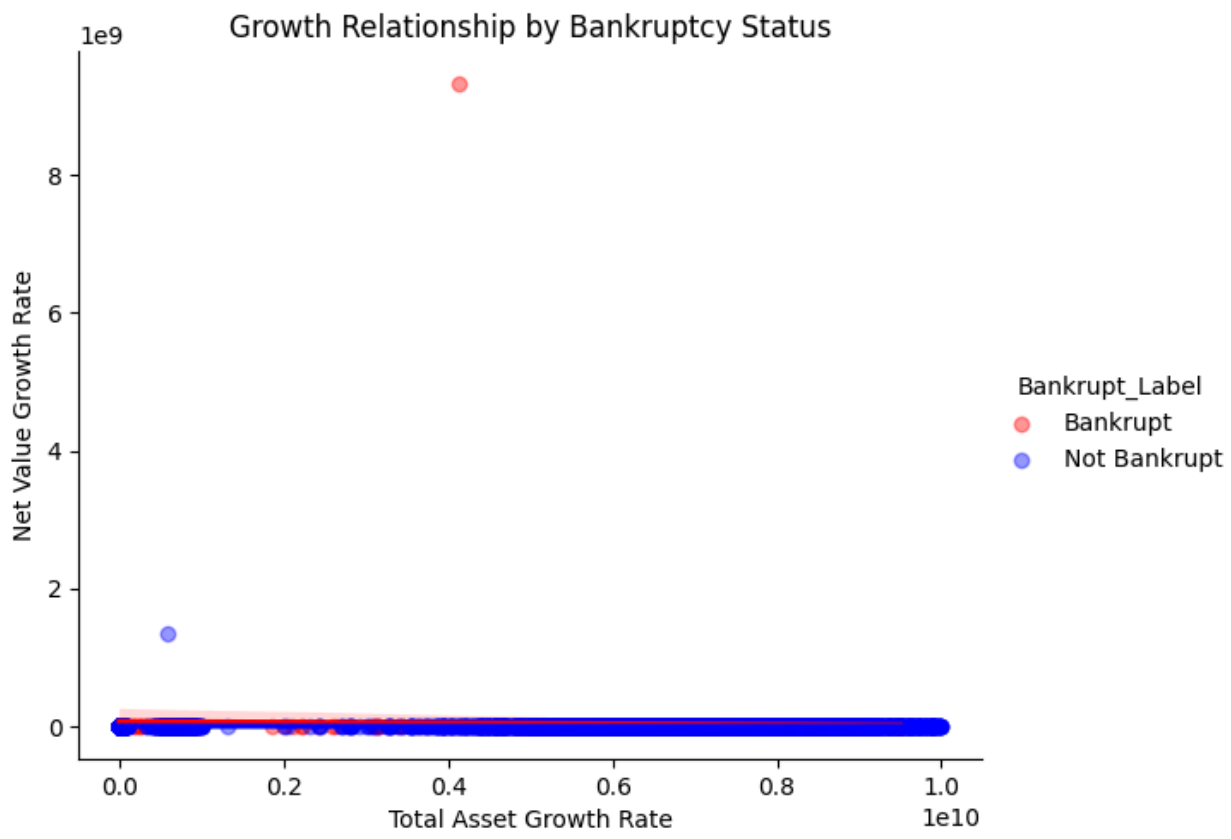


Figure 4.3.3.2 Relationship between and after tax

The x-axis is Total Asset Growth Rate and the y-axis is Net Value Growth Rate, with red points for bankrupt firms and blue points for non-bankrupt firms.

Almost all observations lie densely near the bottom of the plot with both growth measures close to zero, while one red and one blue point are extremely large, pushing the axis scales into the billions.

Interpretation and issues

Because those outliers set the scale, the bulk of the data is visually compressed into a thin band, making it impossible to see any meaningful pattern or difference between bankrupt and non-bankrupt firms.

This suggests that growth variables may contain severe outliers or data errors, and that, without rescaling or trimming these values, this plot cannot be used to draw substantive conclusions about how growth relates to bankruptcy risk.

4.3.4. Correlation Heatmaps

Correlation heatmap is a useful tool to graphically represent how two features are related to each other. Depending upon the data types of the features, we need to use the appropriate correlation coefficient calculation methods.

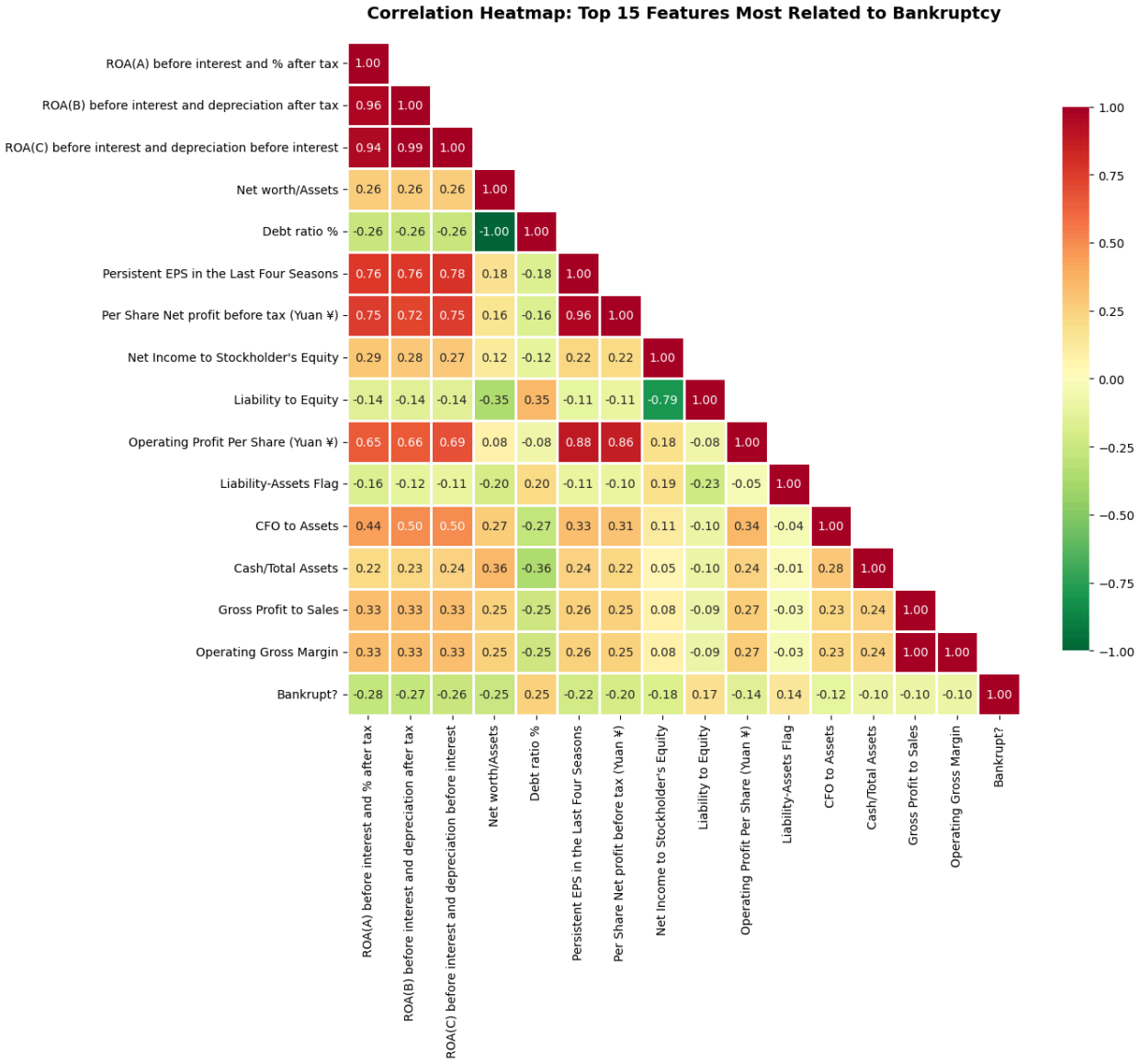


Figure 4.3.4.1 Correlation heatmap : top 15 features most related with to bankruptcy

The heatmap focuses on the 15 features most associated with the binary “Bankrupt?” flag. Colors close to red indicate strong positive correlation, green indicates strong negative correlation, and yellow/orange indicates weak to moderate relationships. The “Bankrupt?” row/column provides a direct view of which financial indicators move with or against bankruptcy, while the rest of the matrix reveals how those indicators interact with one another.

Profitability as a protective factor

All three profitability measures — ROA(A), ROA(B), and ROA(C) — show moderately negative correlations with bankruptcy (around −0.25 to−0.28). This means that as profitability improves, the likelihood of bankruptcy declines in a fairly consistent way. These same ROA measures are extremely highly correlated with one another (above 0.94), indicating they provide overlapping perspectives on the same underlying construct: core operating and asset profitability.

Profit-related per-share metrics (Persistent EPS in the Last Four Seasons, Per Share Net Profit before tax, Net Income to Stockholder’s Equity, Operating Profit Per Share) are all strongly positively correlated with ROA (often 0.7–0.9) and negatively correlated with bankruptcy. This forms a clear “profitability cluster,” suggesting that firms with strong and sustained earnings, healthy returns on equity, and robust per-share performance are structurally less likely to fail.

Leverage, capital structure, and solvency risk

Debt ratio % has a small positive correlation with bankruptcy, while Liability to Equity and the Liability:Assets flag also show weak positive correlations with the bankruptcy indicator. This pattern points to higher leverage being associated with a somewhat higher risk of default, but not as strongly as profitability factors.Net worth/Assets is negatively related to Liability to Equity (around −0.79), which is economically intuitive: firms with higher net worth relative to assets rely less on debt financing and therefore have lower gearing.Liability to Equity also has mild negative correlations with ROA and some earnings measures, indicating that firms with heavier debt loads tend to be less profitable in this dataset. That suggests leverage is not being used, on average, to amplify returns, but rather may be straining performance.

Capital structure contributes to bankruptcy risk, but mainly by eroding profitability and net worth. Risk mitigation strategies should focus less on arbitrary leverage limits and more on whether additional debt is accretive to ROA and equity returns.

Liquidity, cash generation, and operating robustness

Cash/Total Assets and CFO to Assets exhibit mild positive correlations with ROA and mild negative correlations with bankruptcy. Firms that hold more cash relative to their asset base and generate more operating cash flow per unit of assets tend to be more profitable and less likely to go bankrupt.These liquidity and cash-flow measures have weaker effects than earnings or ROA, but they still add explanatory power. For instance, a firm may be marginally profitable yet vulnerable if cash conversion is poor or if it operates with thin cash buffers.The combination of CFO to Assets with profitability metrics highlights a "quality of earnings" concept: profits backed by cash flow appear to be a more reliable signal of survivability than profits alone.

Margin strength and business model quality

Gross Profit to Sales and Operating Gross Margin correlate moderately with profitability measures (roughly 0.25–0.35) and negatively with bankruptcy. Stronger margins are associated with both higher ROA and lower default probability.These margin ratios sit at the link between the firm's competitive position (pricing power, cost control) and financial outcomes. They can deteriorate well before outright losses appear on the income statement.Since margins connect to both revenue quality and cost structure, trends in these ratios can serve as early leading indicators: narrowing margins will typically precede weakening ROA, which in turn raises bankruptcy risk.

4.3.5. Bar Chart

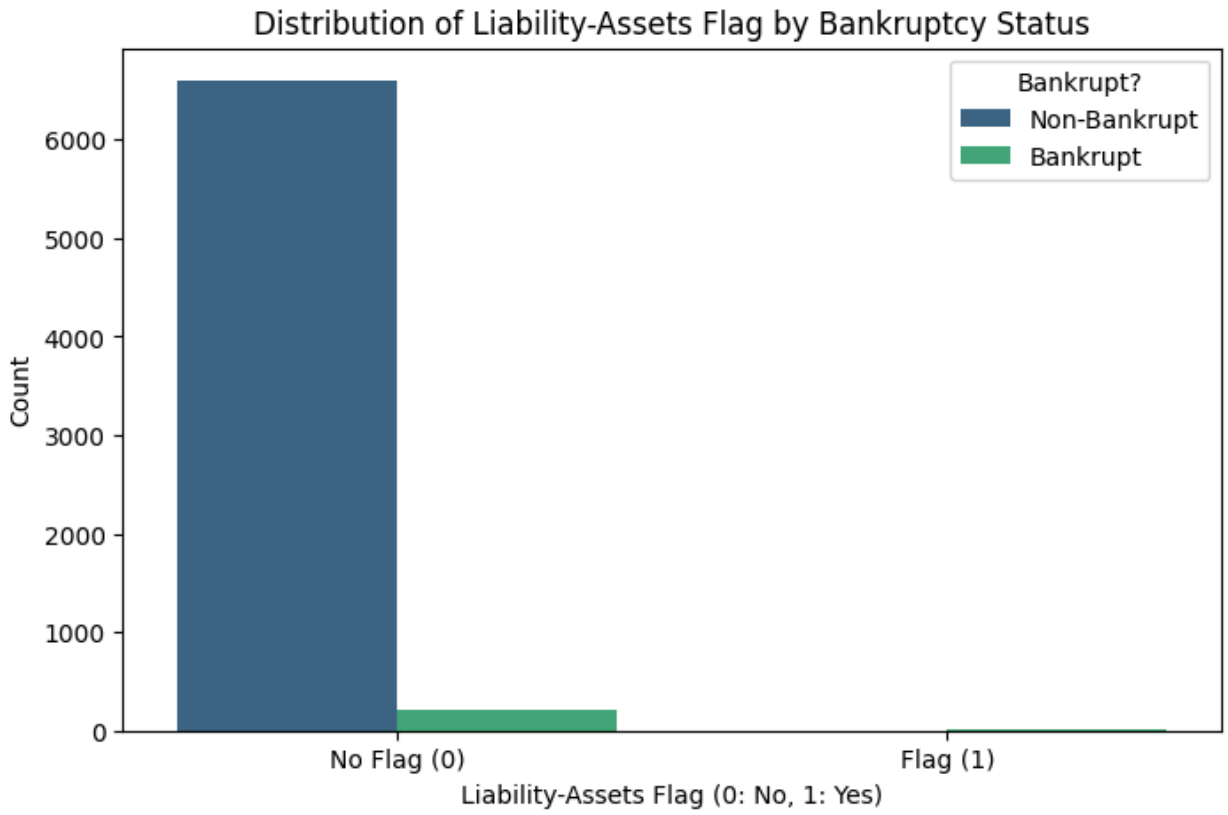


Figure 4.3.5.1 Distribution of liability-assests flag by bankruptcy status

Table 4.3.5.1 Counts for 'Liability-Assets Flag' by Bankrupt? Status:

Liability-Assets Flag	0	1
Bankrupt?		
0	6597	2
1	214	6

Among non-bankrupt firms, 6,597 have no flag (0) while only 2 have a flag (1).

Among bankrupt firms, 214 have no flag (0) and just 6 have a flag (1).

The “1” category is vanishingly rare in both classes, so the model will learn almost nothing from it and may even overfit those few cases.

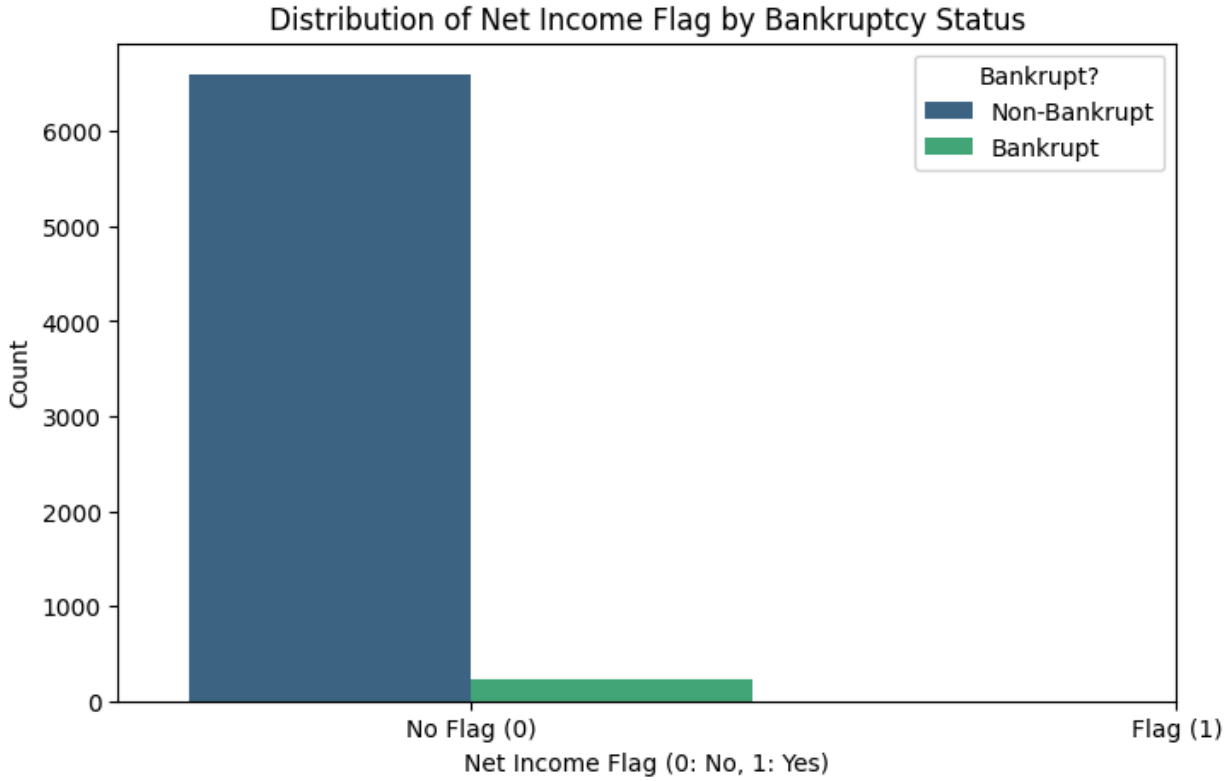


Figure 4.3.12 Distribution of net income flag by bankruptcy status

Table 4.3.1 Counts for 'net income Flag' by Bankrupt? Status:

Net income Flag Bankrupt?	0	1
0	6599	0
1	220	0

All firms, both non-bankrupt (6,599) and bankrupt (220), have Net Income Flag = 1; there are no observations with another value.

5. Data Preparation

Data preparation is the process of cleansing, transforming, and organizing raw data so that it is suitable for analysis and machine learning applications.

In this project we chose to use TWO datasets:

Data A: data without outliers

Data B: data without outliers and with binning

We chose this approach to analyze the impact of different data preparation techniques on the model and which approach will provide better results.

5.1.1. Handling Missing Values

```
1 print("Total missing values in df_filtered:", df_filtered.isna().sum().sum())
Total missing values in df_filtered: 0
```

Figure 5.4.1.1 handling missing values

The dataset was checked for nulls using a command like df_filtered.isna().sum().sum(), which returned 0, meaning there are no missing entries left in any column. Achieving zero missing values typically involves dropping unusable columns, imputing or removing rows with nulls, and verifying again so that models are not biased or broken by incomplete records.

5.1.2. Duplicated Data Checking

```
1 df_filtered.duplicated().sum()
np.int64(0)
```

Figure 5.1.1.2 number of duplicated columns

5.1.3. Outlier Detection and Handling

We decided to use boxplot to determine the outliers of our dataset and capping the outliers in our dataset.

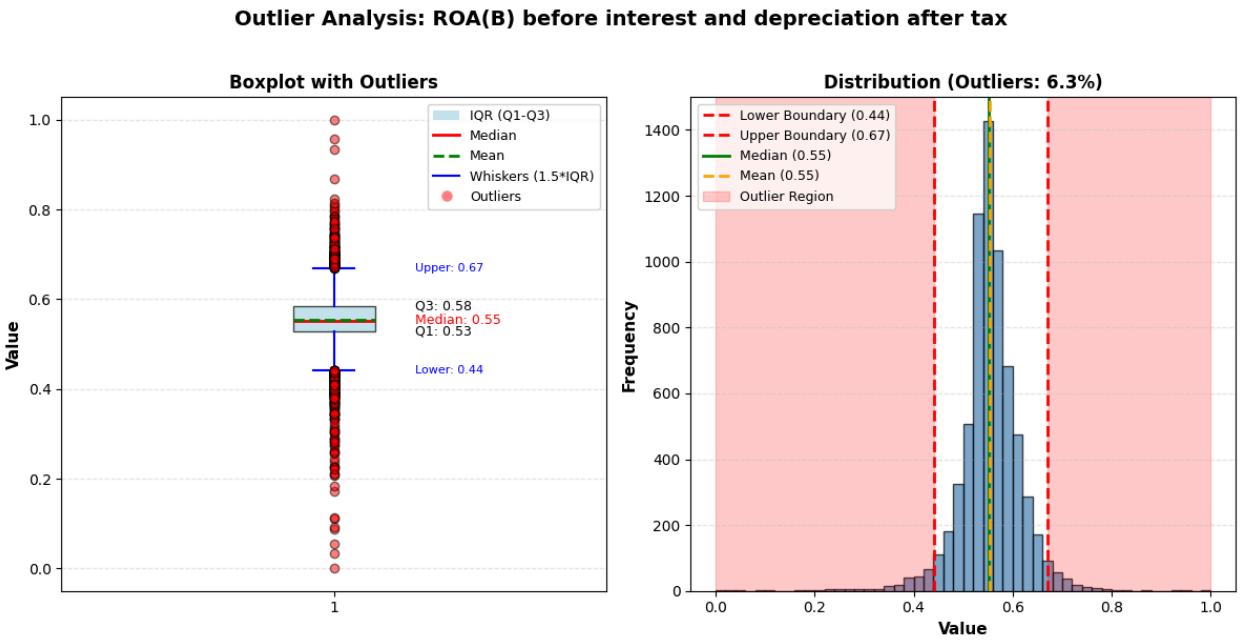


Figure 5.1.3.1 Boxplot and distribution view of ROA(B) before interest and depreciation after tax

Outlier Analysis for 'ROA(B) before interest and depreciation after tax'

Number of outliers: 432 (6.34%)

Boxplot view

Most firms have ROA(B) between about 0.53 and 0.58, with a median of 0.55 and mean of 0.55, so the central profitability level is stable and symmetric. The whiskers extend from roughly 0.44 to 0.67; points outside this range (a few very low ROAs near 0–0.2 and some very high ones up to around 1.0) are flagged as outliers, representing unusually poor or exceptional profitability.

Distribution view

The histogram is concentrated between the lower boundary (0.44) and upper boundary (0.67), and only about 6.3% of observations fall in the shaded outlier region. Because the bulk of the data forms a smooth, unimodal shape, we can safely trim or winsorise those 6.3% extreme values if we want to reduce model sensitivity to rare profitability extremes while preserving the main ROA pattern.

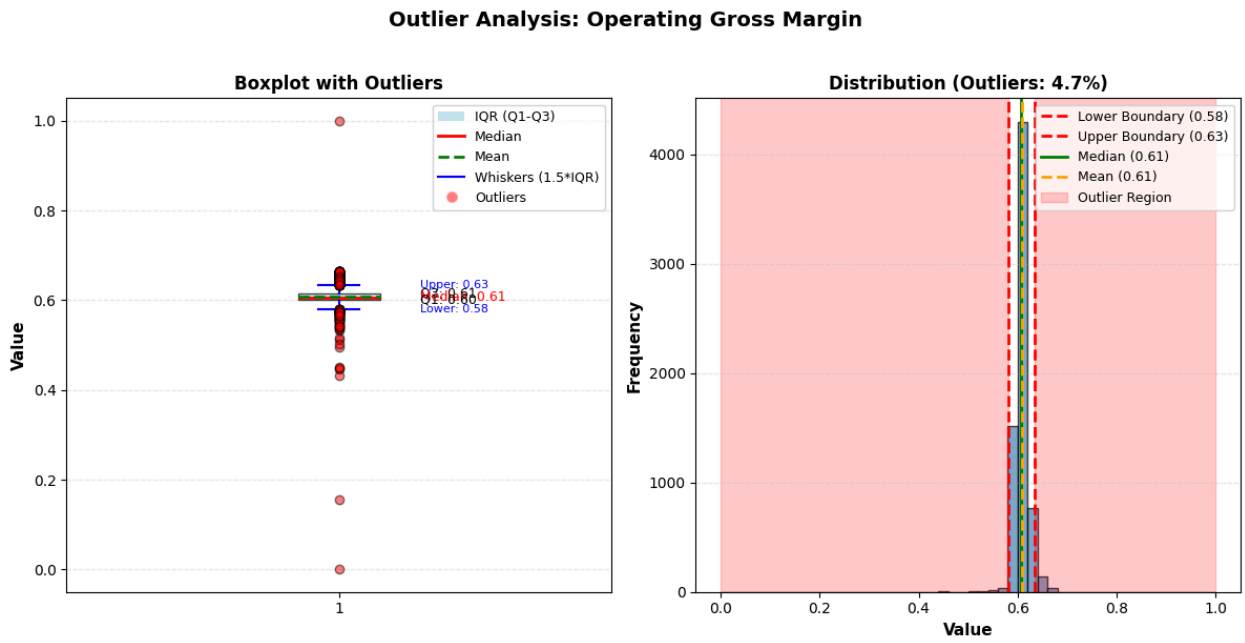


Figure 5.1.3.2 Boxplot and distribution view of Operating Gross Margin

Outlier Analysis for 'Operating Gross Margin'

Number of outliers: 320 (4.69%)

Most firms cluster very tightly between about 0.58 and 0.63, with median and mean both around 0.61, so the core distribution is narrow and stable. Only about 4.7% of observations fall outside the IQR-based bounds (a few low values down to 0 and some higher ones up to 1), which are flagged as outliers.

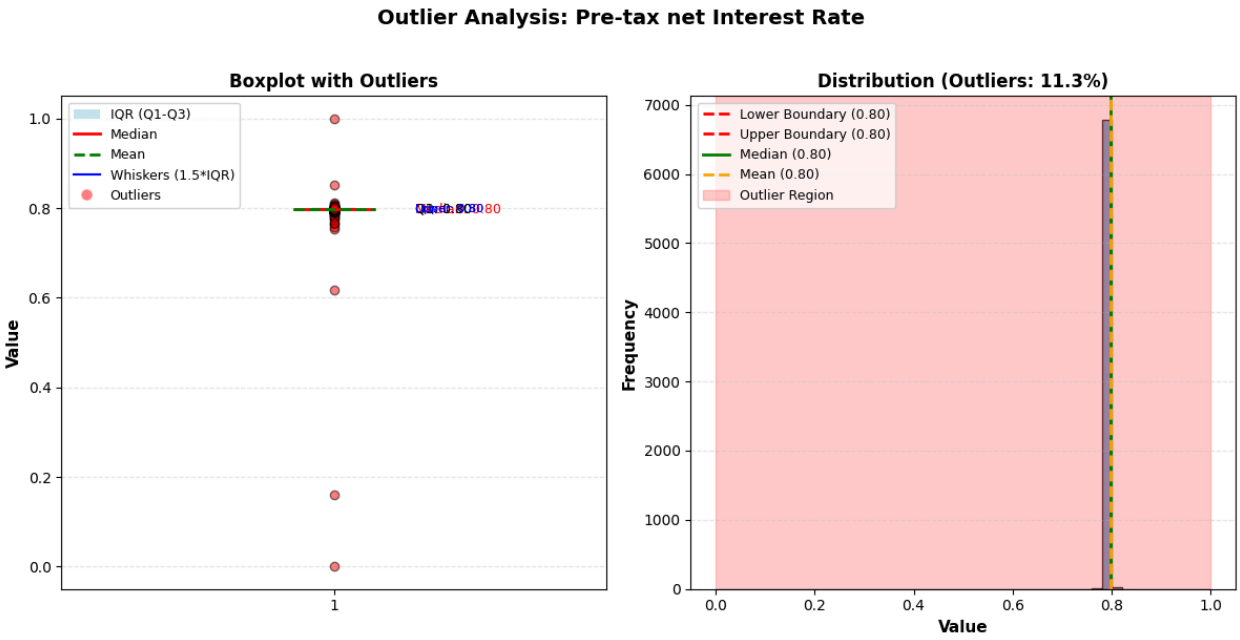


Figure 5.1.3.3 Boxplot and distribution view of Pre-tax net Interest Rate

Outlier Analysis for 'Pre-tax net Interest Rate'

Number of outliers: 773 (11.34%)

Boxplot view

Q1, median, and Q3 are all around 0.80, so the central distribution is extremely tight and almost flat at 0.8.

A handful of observations deviate noticeably (including some near 0 and some around 1), which appear as red outlier points outside the very short whiskers.

Distribution view

The histogram is a very tall spike at about 0.8, with the lower and upper IQR boundaries also at roughly 0.8, and about 11.3% of values falling in the shaded outlier region.Because nearly all information is that the rate ≈0.8, this feature contributes little variation to a model; the few outliers can be capped or removed, but overall this variable is a weak candidate and could be dropped if we need to reduce features.

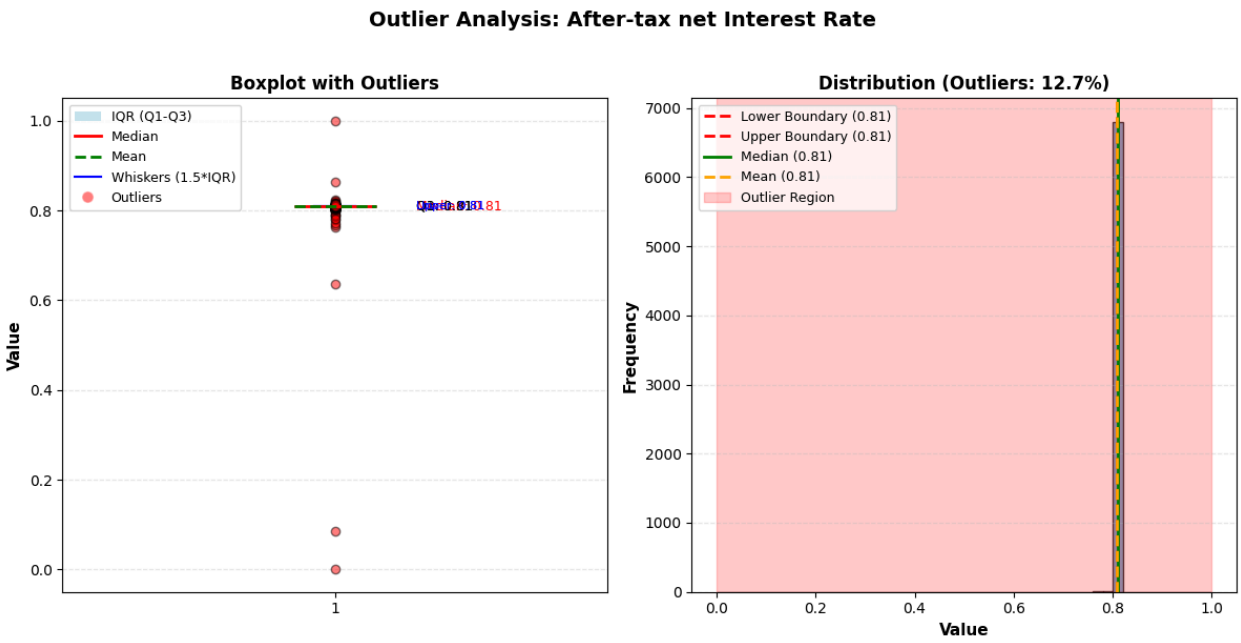


Figure 5.1.3.4 Boxplot and distribution view of After-tax net Interest Rate

Outlier Analysis for 'After-tax net Interest Rate'

Number of outliers: 867 (12.71%)

In the boxplot, Q1, median, and Q3 are all essentially 0.81, so the IQR is extremely narrow and the main data cloud is a flat line at that value; a few observations near 0, 0.1, or 1.0 appear as red outlier points outside the very short whiskers. In the histogram, nearly all observations form a tall spike at 0.81, with both IQR boundaries also at 0.81 and about 12.7% of values falling in the shaded “outlier” region, meaning the variable carries very little useful variation for a model and, like the pre-tax rate, is a good candidate to drop or to compress into a simple indicator if needed.

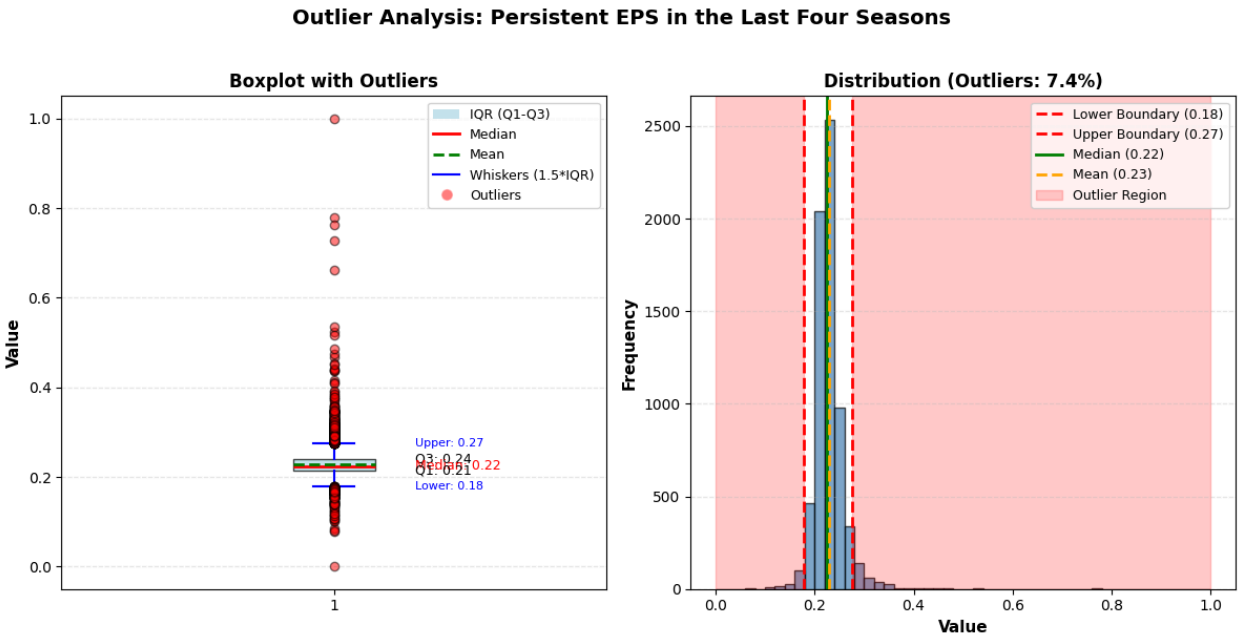


Figure 5.1.3.5 Boxplot and distribution view of Persistent EPS in the Last Four Seasons

Outlier Analysis for 'Persistent EPS in the Last Four Seasons'

Number of outliers: 508 (7.45%)

In the boxplot, most firms' persistent EPS lies between about 0.20 (Q1 ≈ 0.20) and 0.24 (Q3 ≈ 0.24), with a median around 0.22 and mean around 0.23, so the central cluster is tight and symmetric.The histogram confirms a strong peak between the lower and upper IQR-based bounds (≈0.18–0.27), while only about 7.4% of observations fall outside these limits as outliers (very low or very high EPS), meaning we can keep this variable for modelling and optionally cap or trim just those 7.4% extreme values.

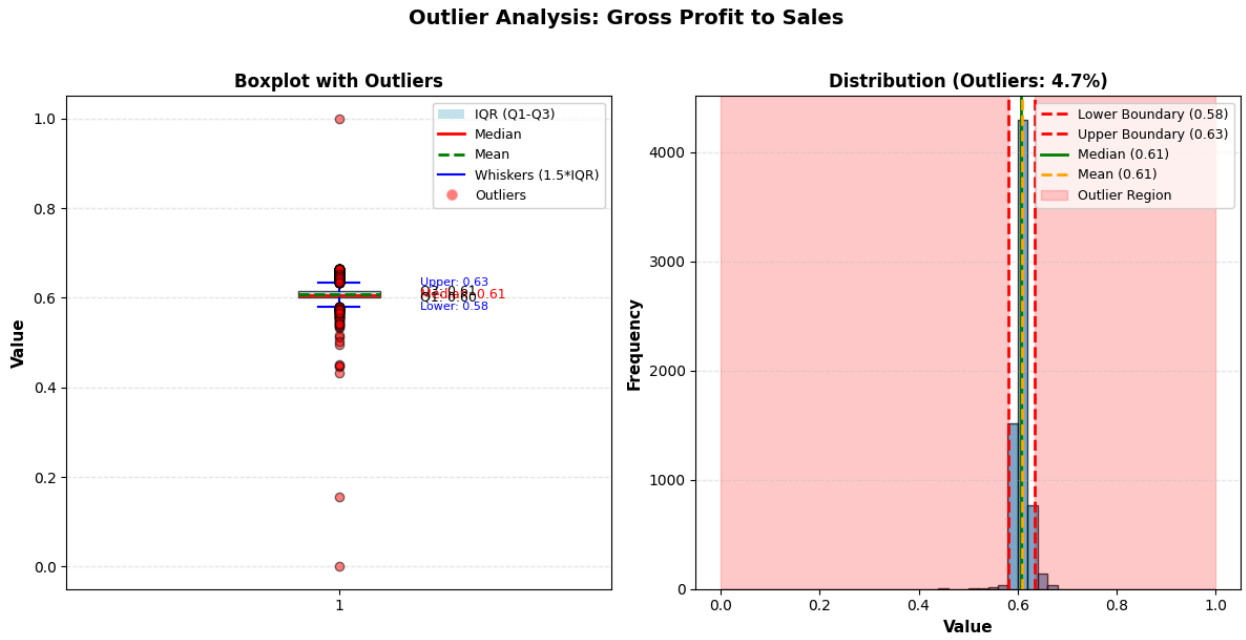


Figure 5.1.3.6 Boxplot and distribution view of Gross Profit to Sales

Outlier Analysis for 'Gross Profit to Sales'

Number of outliers: 320 (4.69%)

In the boxplot, the central 50% of firms lie between about 0.58 (Q1) and 0.63 (Q3), with mean and median both around 0.61, so most companies have gross margins near 60%. The histogram confirms a strong peak in that range, and only about 4.7% of observations fall outside the IQR-based bounds as outliers (a few very low or very high margins up to 1.0), meaning this variable is stable and informative for modelling; we can retain it and, if desired, cap or trim just those 4.7% extreme values rather than dropping the feature.

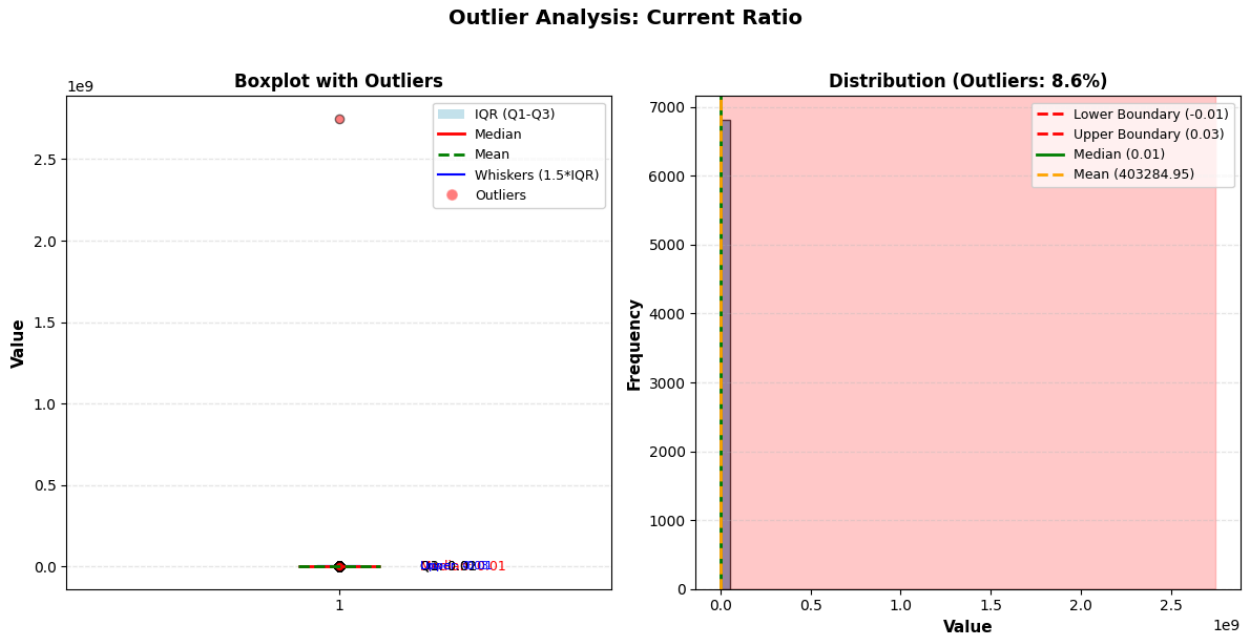


Figure 5.1.3.7 Boxplot and distribution view of Current Ratio

Outlier Analysis for 'Current Ratio'

Number of outliers: 589 (8.64%)

In the boxplot, almost all observations lie very close to zero, but a single point shoots up to about 2.7×10^{9} , which is clearly unrealistic for a current ratio and is flagged as a red outlier; Q1, median, and Q3 are all near 0.01, so the normal range is extremely tight.

In the histogram, the bulk of data is compressed between roughly −0.01 and 0.03 (the IQR-based bounds), while about 8.6% of values fall in the shaded outlier region and the mean is dragged up to more than 400,000, indicating that a small set of erroneous or mis-scaled records completely warps the distribution, so we should either cap/remove those extremes or consider dropping Current Ratio as a reliable feature.

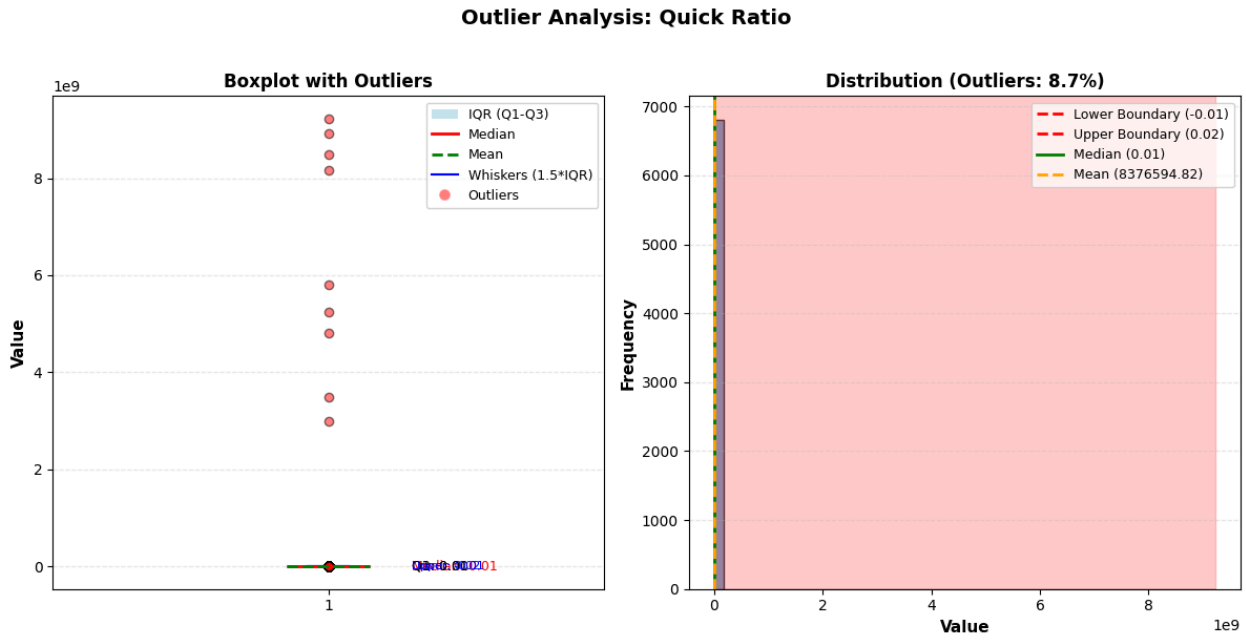


Figure 5.1.3.8 Boxplot and distribution view of Quick Ratio

Outlier Analysis for 'Quick Ratio'

Number of outliers: 591 (8.67%)

In the boxplot, almost all observations lie extremely close to zero (Q1, median, and Q3 are all around 0.01), but several points jump to about 3×10^9 –9×10^9 , which are flagged as red outliers and are not financially plausible for a quick ratio.

In the histogram, the main data mass is squeezed between roughly −0.01 and 0.02 (IQR-based bounds), while about 8.7% of values fall in the shaded outlier region and the mean is dragged up to roughly 8.4 million, indicating that these erroneous spikes dominate the statistics, so the Quick Ratio feature should either be heavily cleaned (capping/removing extreme records) or considered for removal from the model.

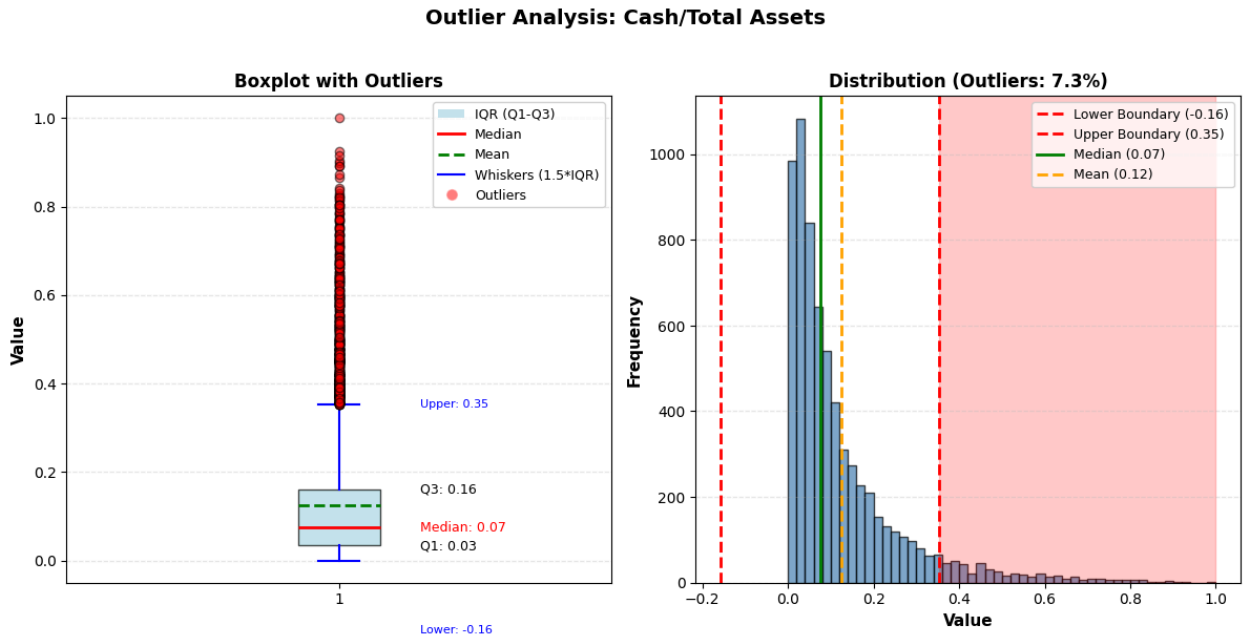


Figure 5.1.3.9 Boxplot and distribution view of Cash/Total Assets

Outlier Analysis for 'Cash/Total Assets'

Number of outliers: 496 (7.27%)

In the boxplot, the central 50% of firms have cash ratios between about 0.03 (Q1) and 0.16 (Q3), with a median around 0.07 and a mean around 0.12, indicating that most companies keep 3–16% of assets in cash while a few hold much more.

The histogram confirms a right-skewed distribution: most observations are below the upper IQR-based bound of 0.35, while about 7.3% of records lie in the shaded outlier region (cash ratios above 0.35 up to 1.0 and a few slightly negative values), so we can keep this variable for modelling and optionally cap or trim those high-cash extremes if they overly influence the model.

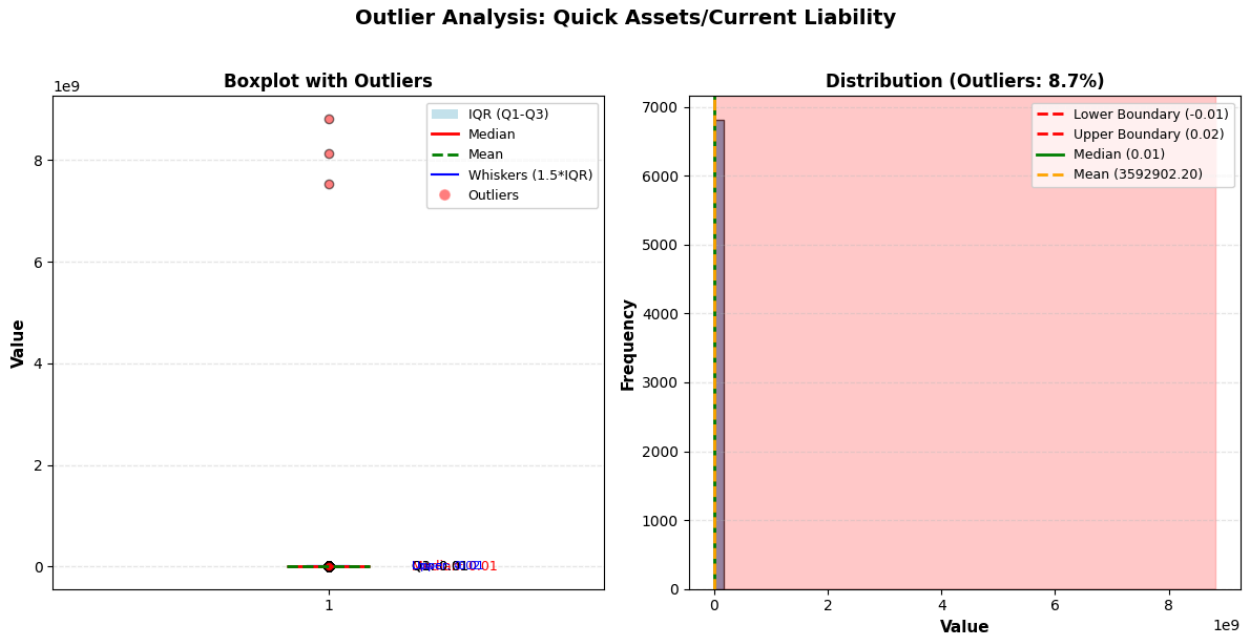


Figure 5.1.3.10 Boxplot and distribution view of Quick Assets/Current Liability

Outlier Analysis for 'Quick Assets/Current Liability'

Number of outliers: 596 (8.74%)

In the boxplot, Q1, median, and Q3 are all around 0.01, so almost all firms have quick-assets coverage very close to zero; however, several observations jump to about 7.5×10^9 –8.8×10^9 , which are unrealistically large for this ratio and appear as red outliers towering above the rest.

In the histogram, the genuine data are squeezed between roughly −0.01 and 0.02 (IQR-based bounds), but around 8.7% of values sit in the shaded outlier region and the mean is dragged up to about 3.6 million, indicating that these extreme records dominate the statistics, so this feature should either be heavily cleaned (capping or removing the huge spikes) or considered for removal from the modelling dataset.

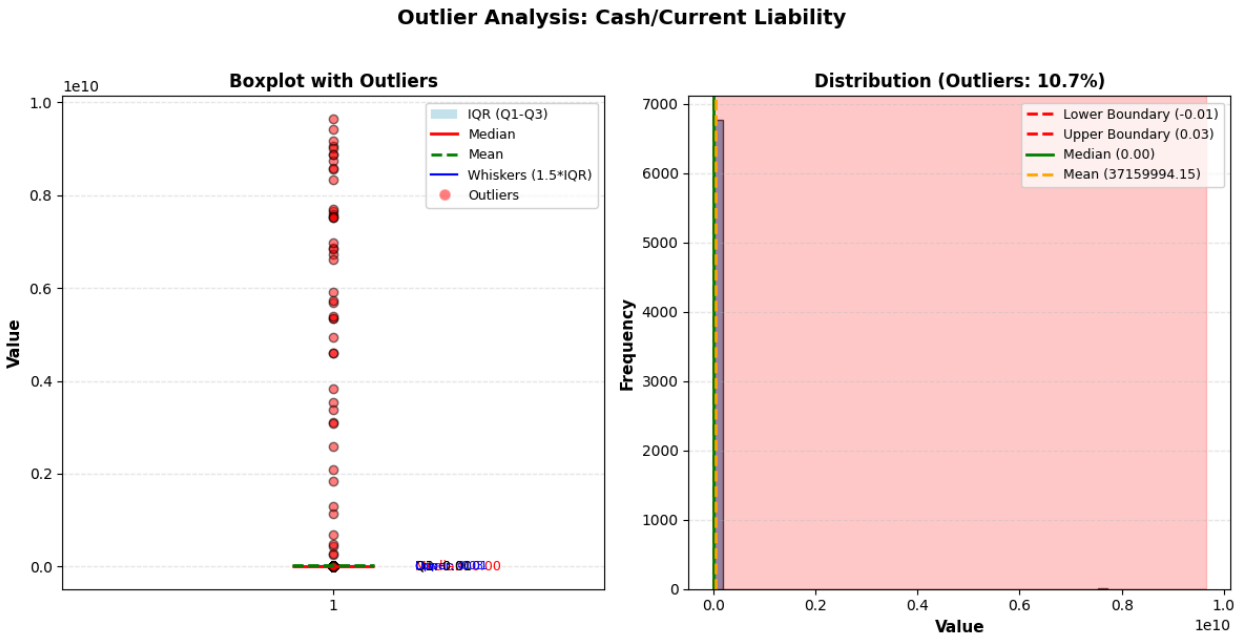


Figure 5.1.3.11 Boxplot and distribution view of Cash/Current Liability

Outlier Analysis for 'Cash/Current Liability'

Number of outliers: 728 (10.68%)

In the boxplot, almost all firms sit extremely close to zero (Q1, median, and Q3 are near 0.00–0.01), but many observations shoot up to around 4×10^9 –9.5×10^9 , all flagged as red outliers, which are not realistic levels for a liquidity ratio.

In the histogram, the genuine data are squeezed between about −0.01 and 0.03 (IQR bounds), while roughly 10.7% of records fall in the shaded outlier region and drag the mean up to around 3.7 million, so these extreme cases dominate the statistics and this feature should either be aggressively capped/filtered or considered for exclusion from the model.

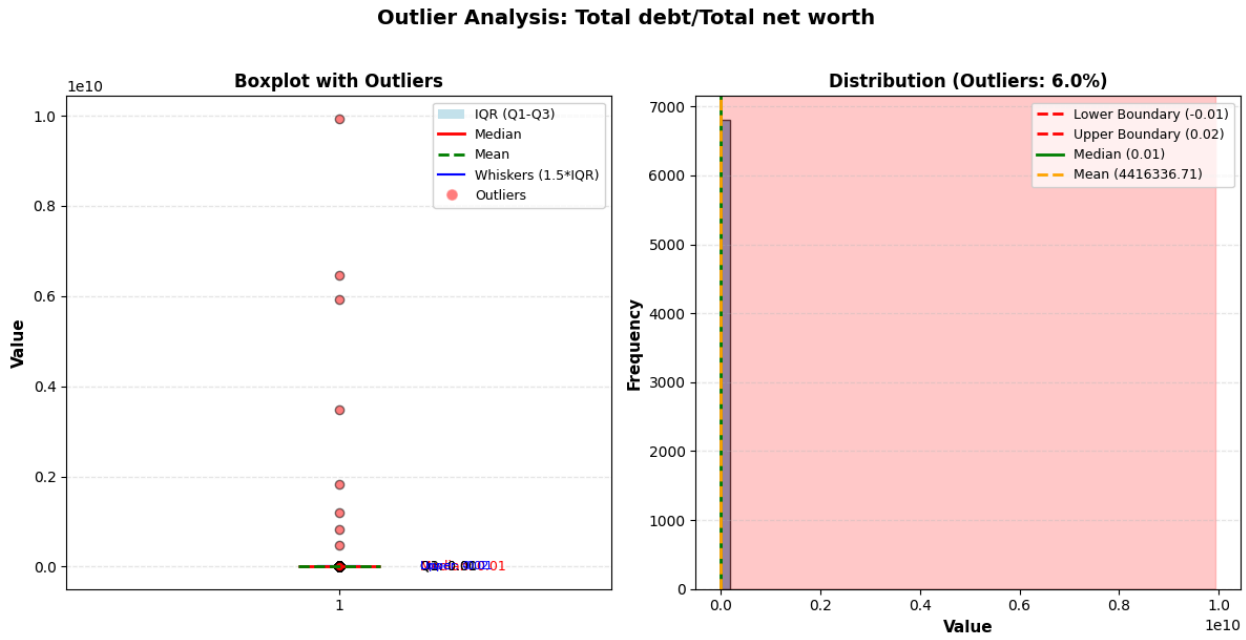


Figure 5.1.3.12 Boxplot and distribution view of Total debt/Total net worth

Outlier Analysis for 'Total debt/Total net worth'

Number of outliers: 407 (5.97%)

In the boxplot, Q1, median, and Q3 are all around 0.01, meaning most firms have very low values by this definition, while some records jump to roughly 1.0×10^{10} , clearly unrealistic for a debt-to-net-worth ratio and marked as red outliers.

In the histogram, almost all data sit between about −0.01 and 0.02 (the IQR bounds), but about 6.0% of observations fall in the shaded outlier region and push the mean up to roughly 4.4 million, so these extreme cases severely distort the distribution and this feature should either be aggressively capped/cleaned or considered for removal from our modelling dataset.

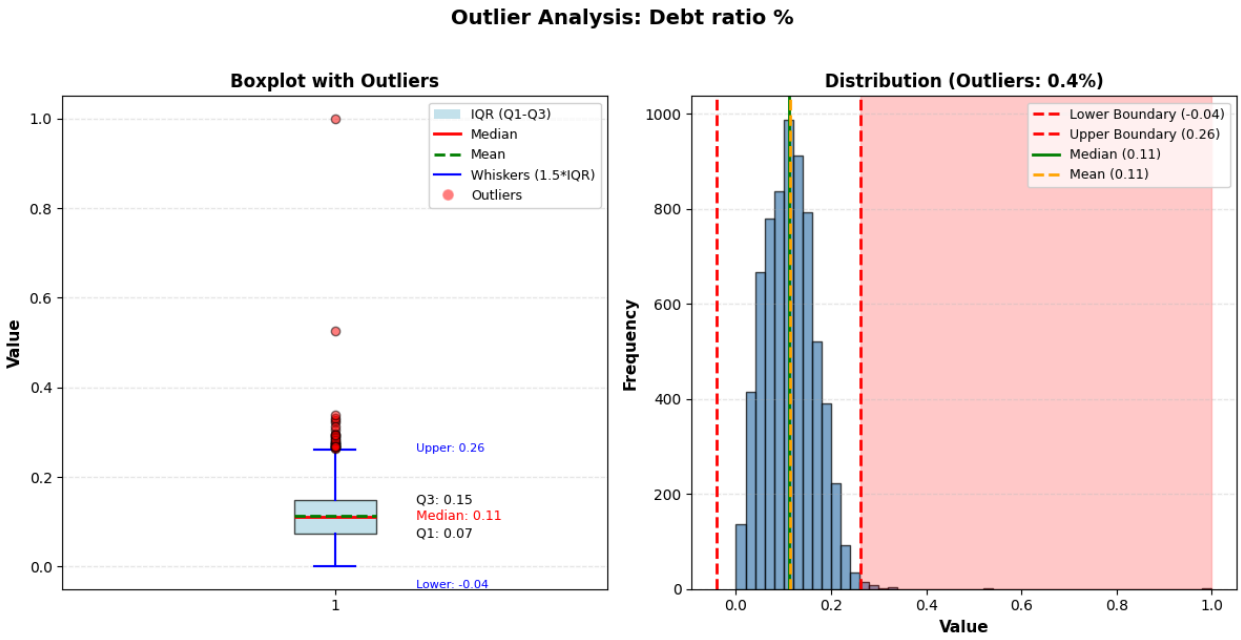


Figure 5.1.3.13 Boxplot and distribution view of Debt ratio %

Outlier Analysis for 'Debt ratio %'

Number of outliers: 30 (0.44%)

In the boxplot, most firms' debt ratios lie between about 0.07 (Q1) and 0.15 (Q3), with median and mean around 0.11, while the whiskers extend from roughly −0.04 to 0.26; only a tiny number of points beyond this range (up to 0.53 or 1.0) are flagged as red outliers.

The histogram confirms a well-shaped distribution concentrated within the IQR-based bounds and only about 0.4% of observations in the shaded outlier region, so we can safely retain Debt ratio % as a key feature and, if desired, lightly cap those few extreme leverage cases without materially affecting the data.

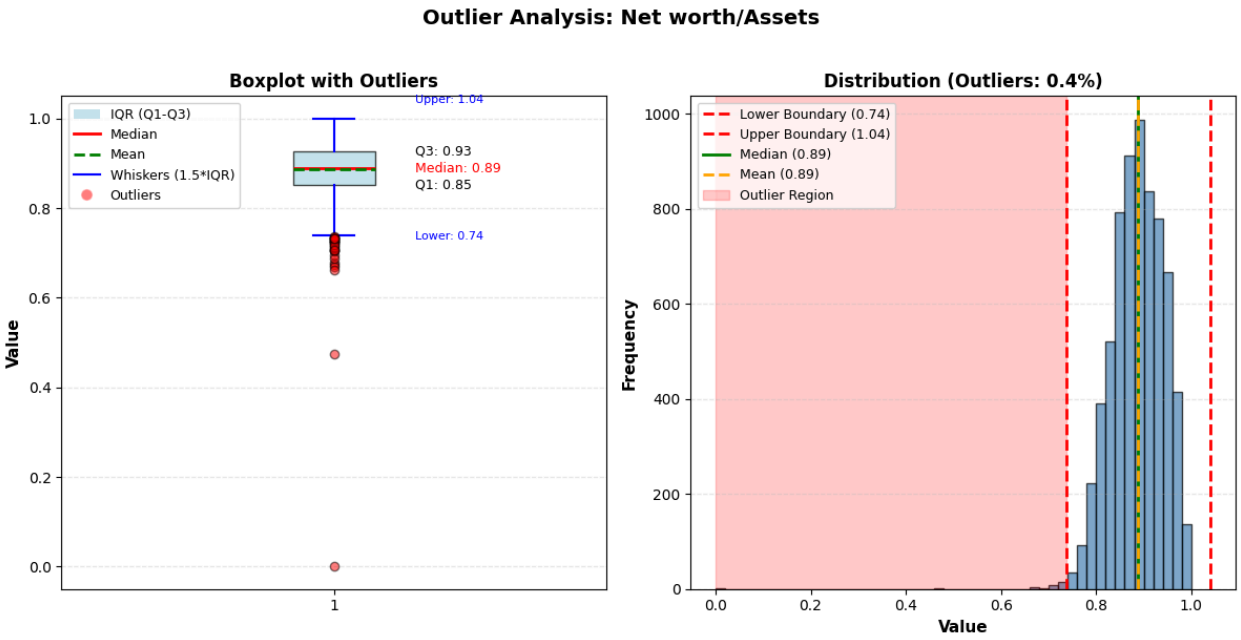


Figure 5.1.3.14 Boxplot and distribution view of Net worth/Assets

Outlier Analysis for 'Net worth/Assets'

Number of outliers: 30 (0.44%)

In the boxplot, most firms have net-worth ratios between about 0.85 (Q1) and 0.93 (Q3), with median around 0.89 and mean about 0.89, indicating a tight, high level of equity backing assets.

The histogram shows a strong peak in that range, with the IQR-based bounds at roughly 0.74 and 1.04 and only about 0.4% of records in the shaded outlier region (a few unusually low or slightly above-1 values), so we might optionally cap those extremes but do not need to drop the feature.

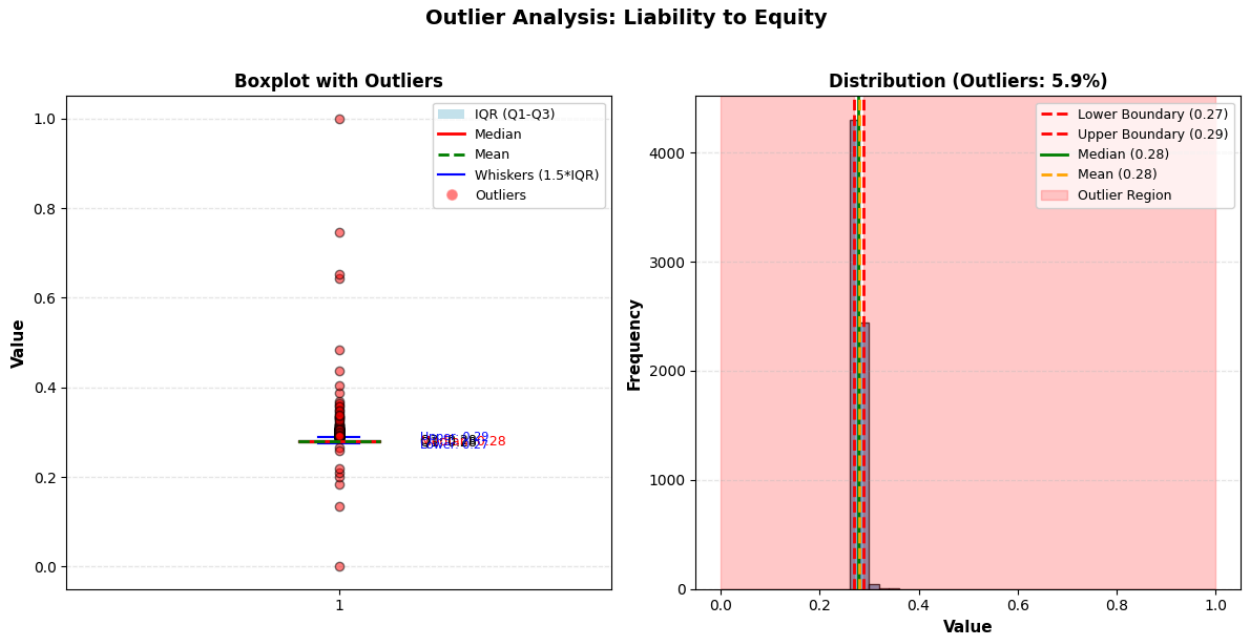


Figure 5.1.3.15 Boxplot and distribution view of Liability to Equity

Outlier Analysis for 'Liability to Equity'

Number of outliers: 404 (5.92%)

In the boxplot, most firms sit tightly between about 0.27 and 0.29, with median and mean around 0.28, meaning the typical liability-to-equity level is very stable and the central spread is narrow.

The histogram confirms a sharp peak in that band; only about 5.9% of observations fall outside the IQR-based bounds (values near 0, 0.2, 0.4–0.75, or 1.0) and are flagged as outliers, so trimming or winsorising those few high- or low-leverage cases is straightforward and preserves the main structure of the feature.

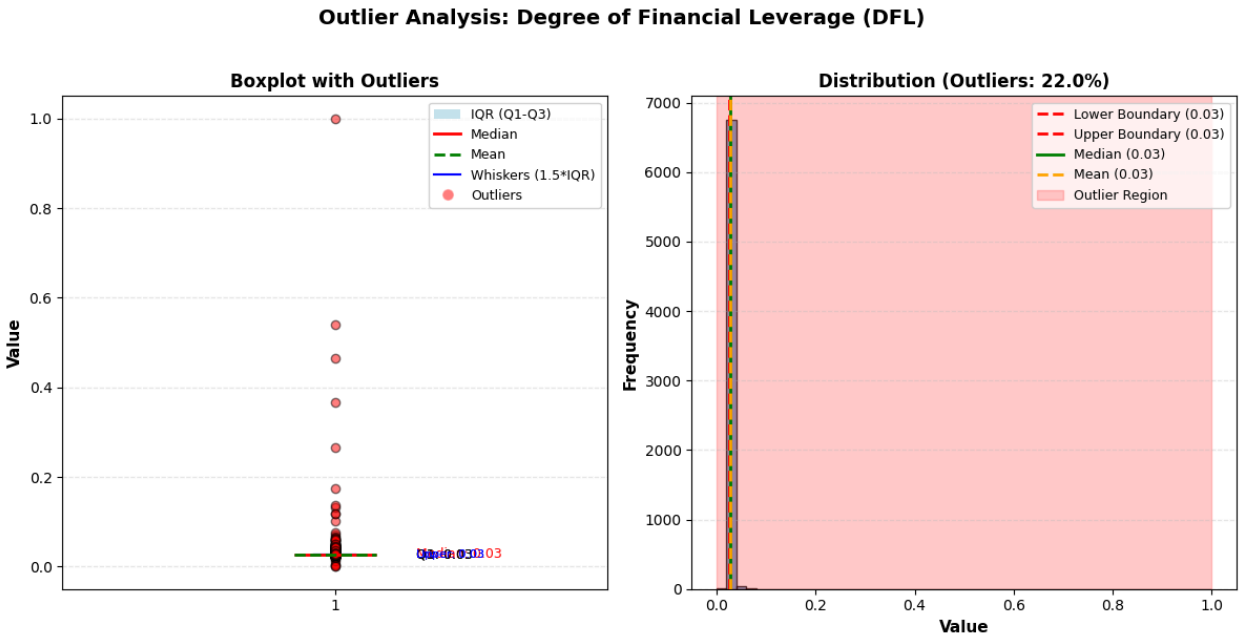


Figure 5.1.3.16 Boxplot and distribution view of Degree of Financial Leverage (DFL)

Outlier Analysis for 'Degree of Financial Leverage (DFL)'

Number of outliers: 1503 (22.04%)

In the boxplot, Q1, median, and Q3 are all around 0.03, meaning the main distribution is a very tight spike at 3%; a few records extend up to 0.1–1.0 and are flagged as outliers.

The histogram shows nearly all observations at 0.03 with both IQR bounds essentially at 0.03, while about 22.0% fall in the shaded outlier region; this pattern indicates strong imbalance and minimal variation in the core data, so DFL will contribute little predictive power compared with our other, richer leverage measures.

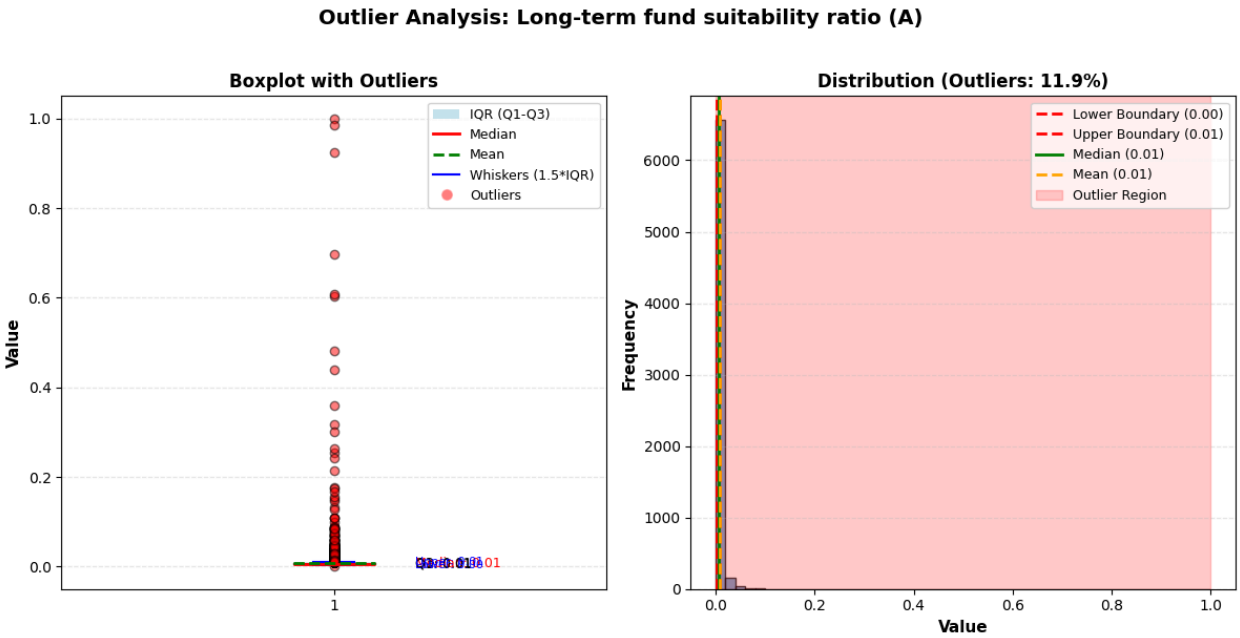


Figure 5.1.3.17 Boxplot and distribution view of Long-term fund suitability ratio (A)

Outlier Analysis for 'Long-term fund suitability ratio (A)'

Number of outliers: 810 (11.88%)

In the boxplot, Q1, median, and Q3 are all around 0.01, so almost all firms sit in a very tight band near zero; a stream of records climbs from about 0.02 up to 1.0 and is flagged as red outliers.

The histogram shows a tall spike at 0.01 with IQR bounds at roughly 0.00–0.01, while about 11.9% of observations lie in the shaded outlier region and stretch the x-axis to 1.0, indicating that a small, erratic set of large values distorts an otherwise flat, low-variance feature that is unlikely to help our model.

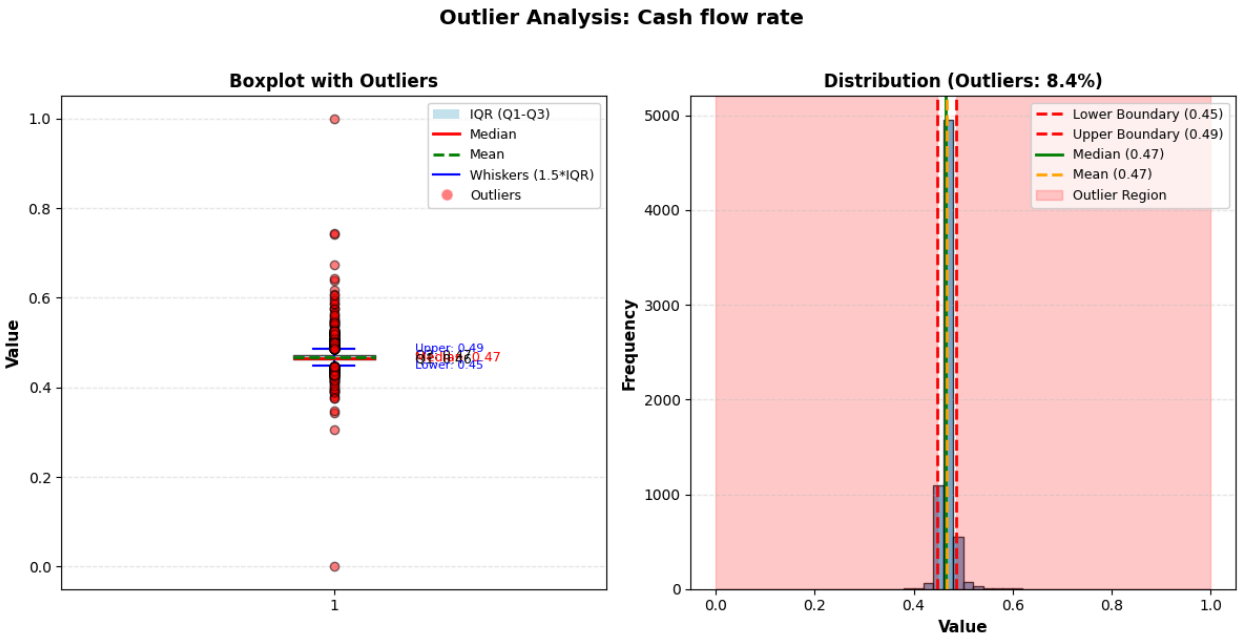


Figure 5.1.3.18 Boxplot and distribution view of Cash flow rate

Outlier Analysis for 'Cash flow rate'

Number of outliers: 576 (8.45%)

In the boxplot, the central 50% of firms have cash-flow rates between about 0.45 (Q1) and 0.49 (Q3), with median and mean around 0.47, and only a small set of points below 0.35 or above 0.6 flagged as outliers.

The histogram shows a clear, tight peak within the IQR-based bounds and about 8.4% of observations in the shaded outlier region, so we can keep this feature for modelling and, if needed, winsorise or trim those few low and high extremes to reduce their influence.

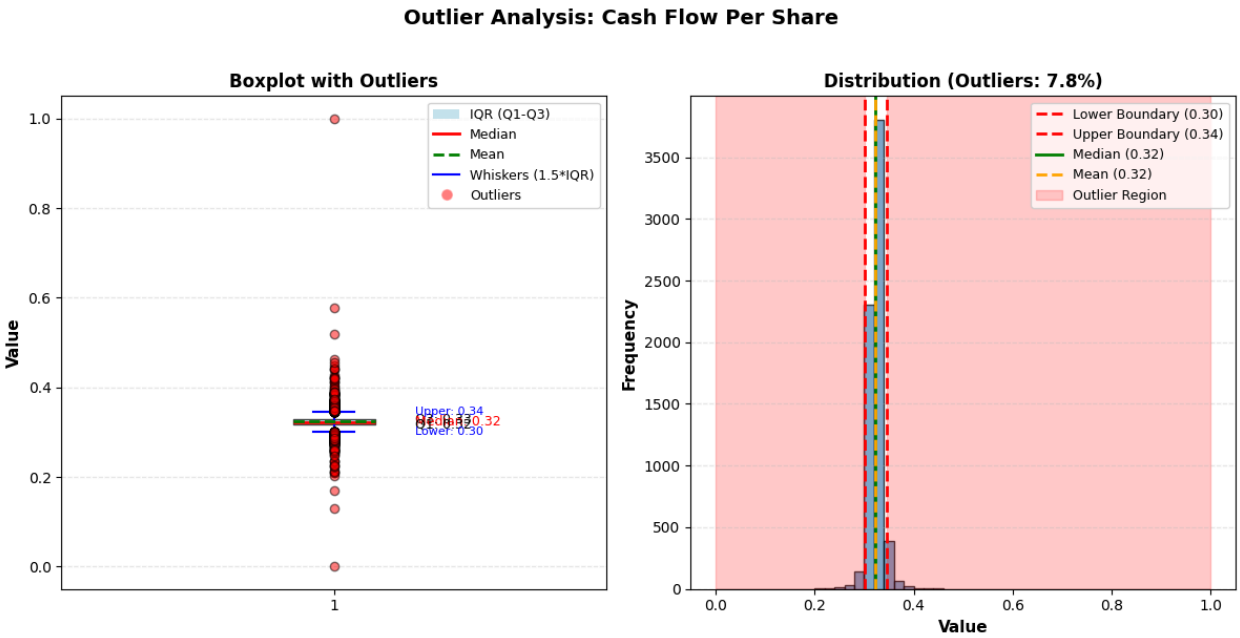


Figure 5.1.3.19 Boxplot and distribution view of Cash Flow Per Share

Outlier Analysis for 'Cash Flow Per Share'

Number of outliers: 532 (7.80%)

In the boxplot, the central 50% of firms have cash-flow rates between about 0.45 (Q1) and 0.49 (Q3), with median and mean around 0.47, and only a small set of points below 0.35 or above 0.6 flagged as outliers.

The histogram shows a clear, tight peak within the IQR-based bounds and about 8.4% of observations in the shaded outlier region, so we can keep this feature for modelling and, if needed, winsorise or trim those few low and high extremes to reduce their influence.

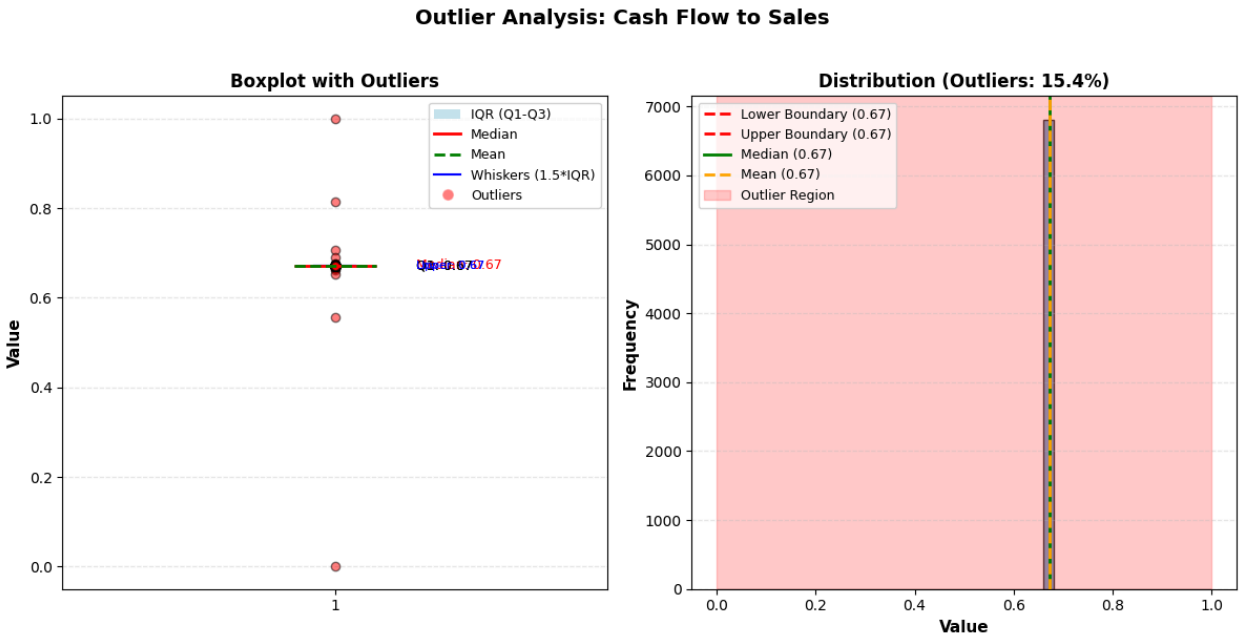


Figure 5.1.3.20 Boxplot and distribution view of Cash Flow to Sales

Outlier Analysis for 'Cash Flow to Sales'

Number of outliers: 1052 (15.43%)

In the boxplot, Q1, median, and Q3 are all essentially 0.67, so the core distribution is a flat spike at 67%; only a few points (near 0, 0.55, 0.82, 1.0, etc.) are flagged as red outliers.

The histogram shows nearly all observations at 0.67 with IQR bounds also at 0.67, while about 15.4% of records lie in the shaded outlier region, suggesting that the feature has very low variance in the main data and a tail of noisy extremes, which limits its usefulness relative to richer cash-flow and margin ratios already in our model.

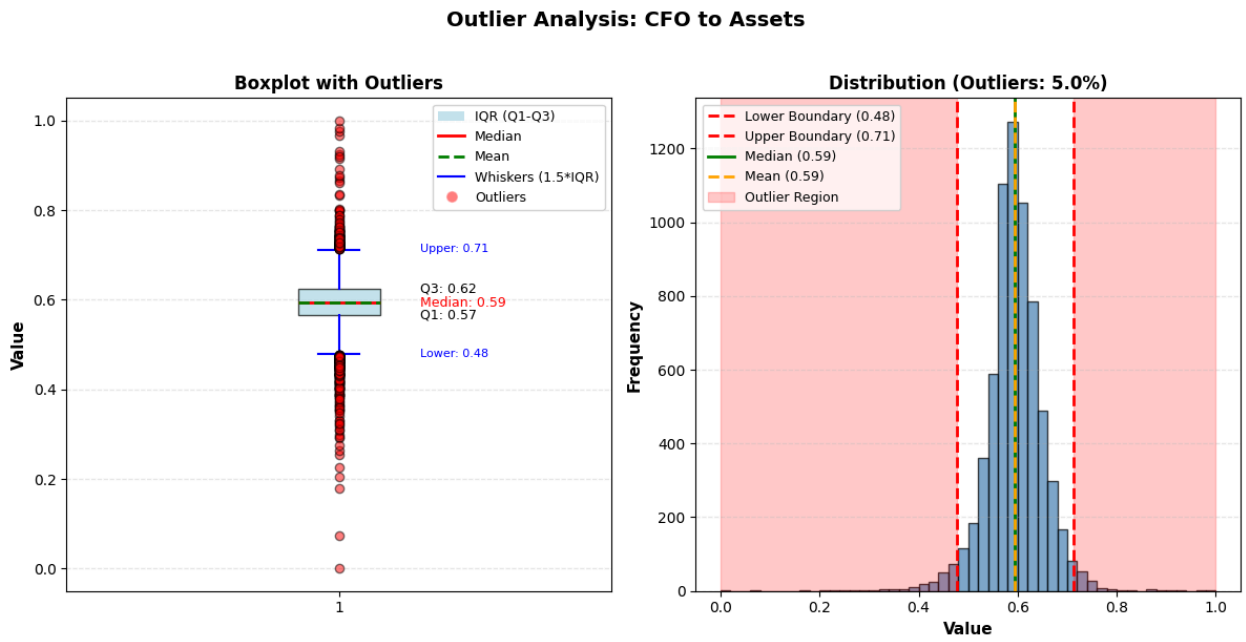


Figure 5.1.3.21 Boxplot and distribution view of CFO to Assets

Outlier Analysis for 'CFO to Assets'

Number of outliers: 342 (5.02%)

In the boxplot, most firms fall between about 0.57 (Q1) and 0.62 (Q3), with median and mean around 0.59; the whiskers extend to roughly 0.48 and 0.71, and points outside this range (down to near 0 and up to 1.0) are flagged as red outliers.

The histogram shows a smooth, bell-shaped peak within the IQR-based bounds and about 5.0% of observations in the shaded outlier region, indicating that cash flow generated per unit of assets has a meaningful, interpretable spread; we can retain this variable and optionally winsorise those few extreme low/high values to stabilise our model.

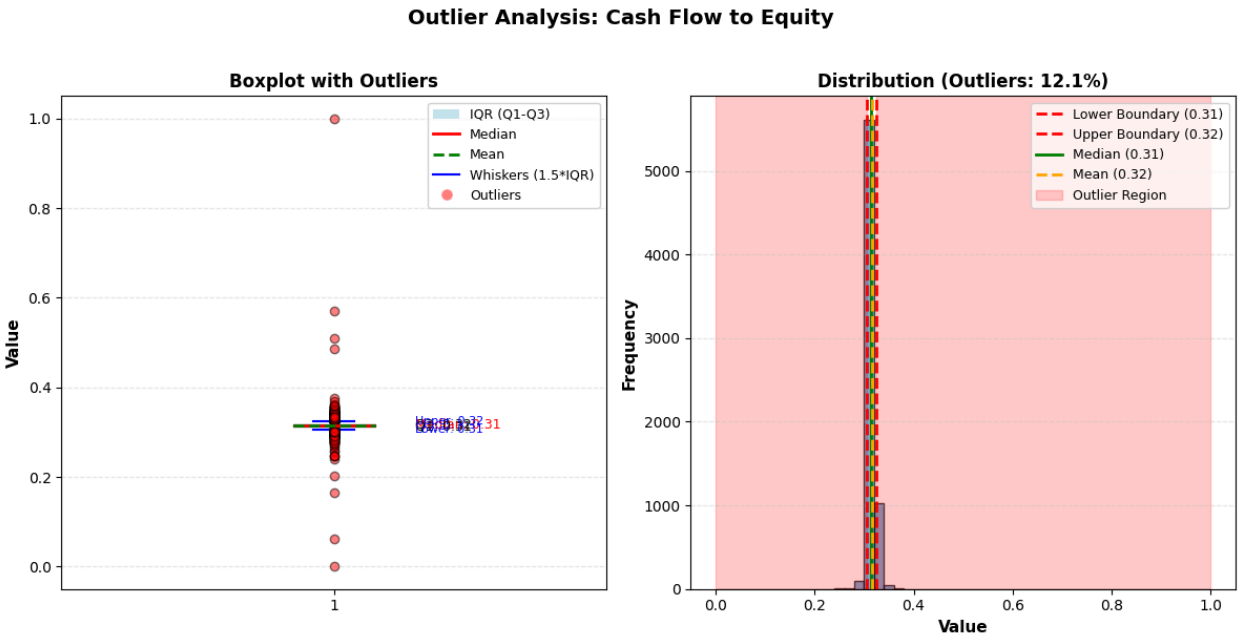


Figure 5.1.3.22 Boxplot and distribution view of Cash Flow to Equity

Outlier Analysis for 'Cash Flow to Equity'

Number of outliers: 827 (12.13%)

In the boxplot, most companies sit between about 0.31 and 0.32 (Q1–Q3), with median and mean around 0.31–0.32, while a series of points from near 0 up to 1.0 are flagged as outliers; the core spread is very narrow, but the upper tail is long.

The histogram shows a sharp peak at 0.31 within the IQR bounds and about 12.1% of observations in the shaded outlier region, indicating that typical cash-flow-to-equity levels are stable but a non-trivial set of firms have unusually low or high values, so trimming or winsorising those extremes would make this feature more robust for modelling.

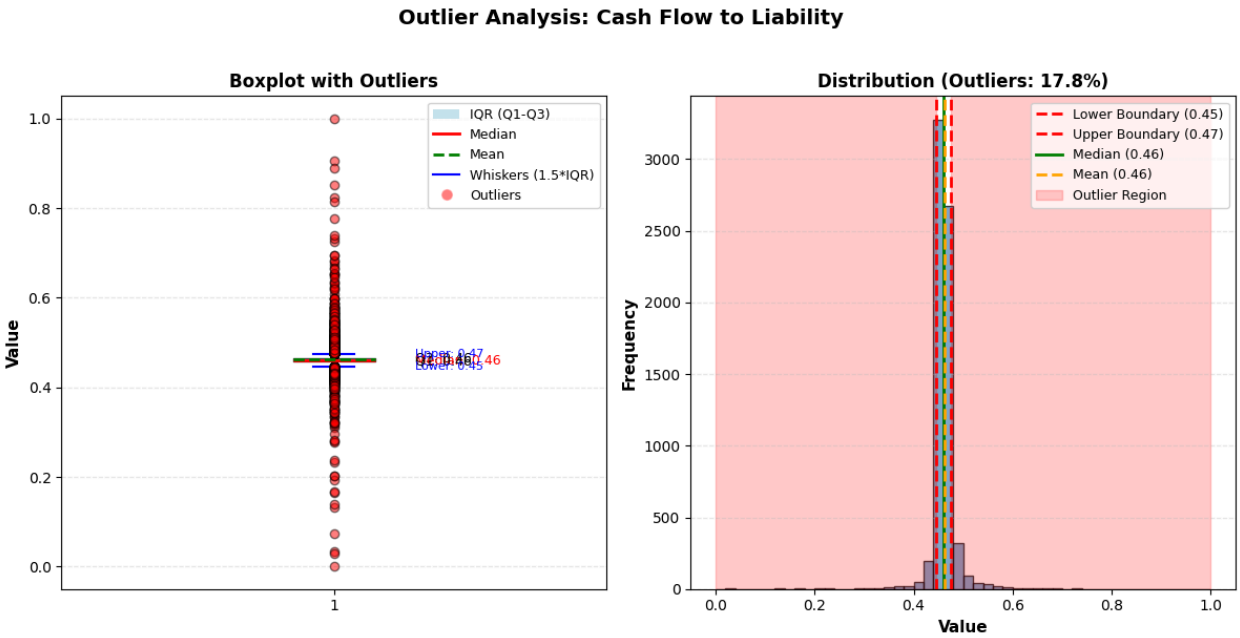


Figure 5.1.3.23 Boxplot and distribution view of Cash Flow to Liability

Outlier Analysis for 'Cash Flow to Liability'

Number of outliers: 1212 (17.77%)

In the boxplot, the central 50% of observations sit tightly between about 0.45 (Q1) and 0.47 (Q3), with median and mean around 0.46; a stream of points from roughly 0.1 up to 1.0 are flagged as red outliers.

The histogram shows a sharp spike around 0.46 within the IQR bounds and about 17.8% of records in the shaded outlier region, meaning typical firms cluster at a similar cash-flow-to-liability level while nearly one-fifth have unusually low or high values, so capping those extremes will make this feature more stable without losing its core information.

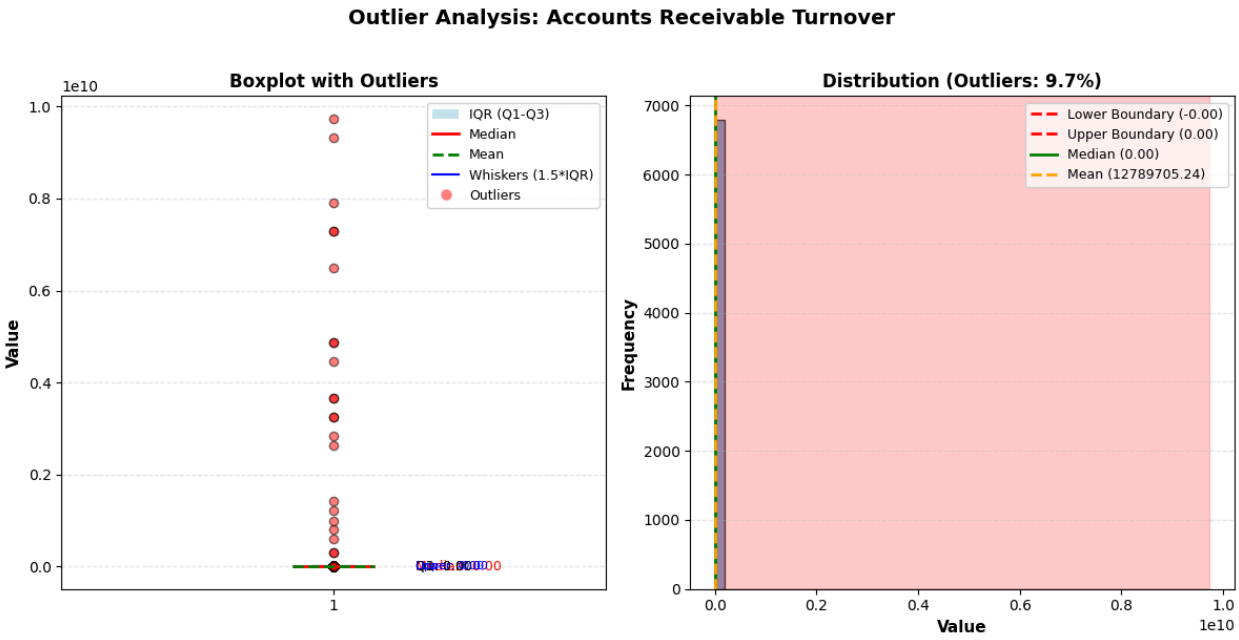


Figure 5.1.3.24 Boxplot and distribution view of Accounts Receivable Turnover

Outlier Analysis for 'Accounts Receivable Turnover'

Number of outliers: 659 (9.66%)

In the boxplot, Q1, median, and Q3 are all essentially 0, indicating that almost all firms have turnover very close to zero, while a series of observations shoot up to around 10^{10} , which are flagged as red outliers and are not realistic for a turnover ratio.

The histogram shows nearly all data crammed between the IQR bounds near 0, but about 9.7% of records lie in the shaded outlier region and drive the mean up to about 1.28×10^{7}, meaning these huge spikes completely distort the distribution; in practice, this feature should either be capped at a sensible upper limit or dropped from the model.

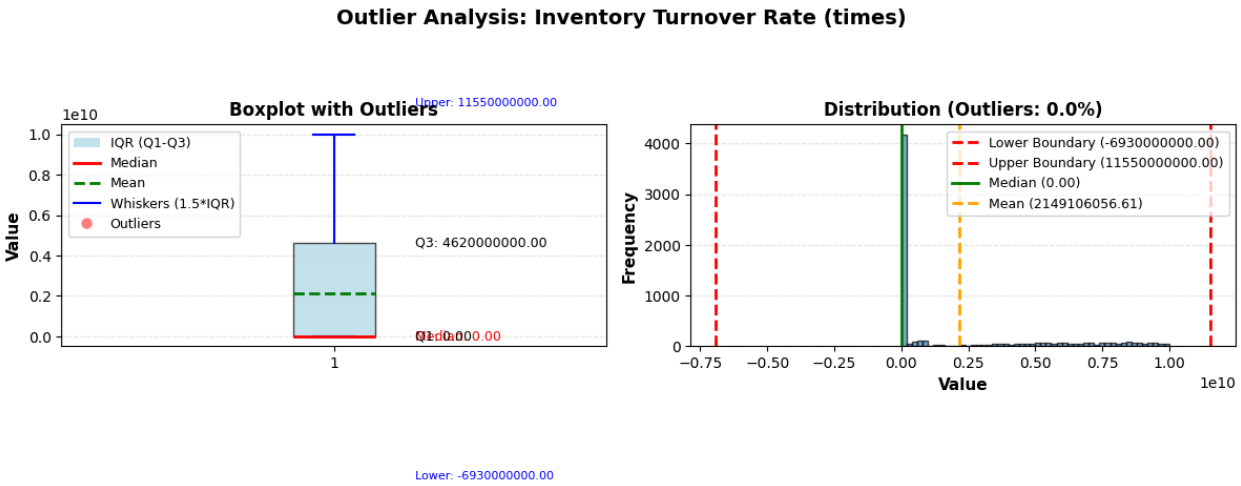


Figure 5.1.3.25 Boxplot and distribution view of Inventory Turnover Rate (times)

Outlier Analysis for 'Inventory Turnover Rate (times)'

Number of outliers: 0 (0.00%)

In the boxplot, the median is 0 while Q3 is around 4.62×10^{10} and the whisker reaches 1.155×10^{11} ; the IQR spans from 0 to an enormous number, which is not realistic for a turnover "times" ratio.

The histogram shows almost all observations counted as "non-outliers" despite this huge spread, with an upper boundary of 1.155×10^{11} , a lower boundary of -6.93×10^{10} , and a mean around 2.15×10^{10} , indicating that scaling or calculation errors dominate the feature rather than meaningful business variation.

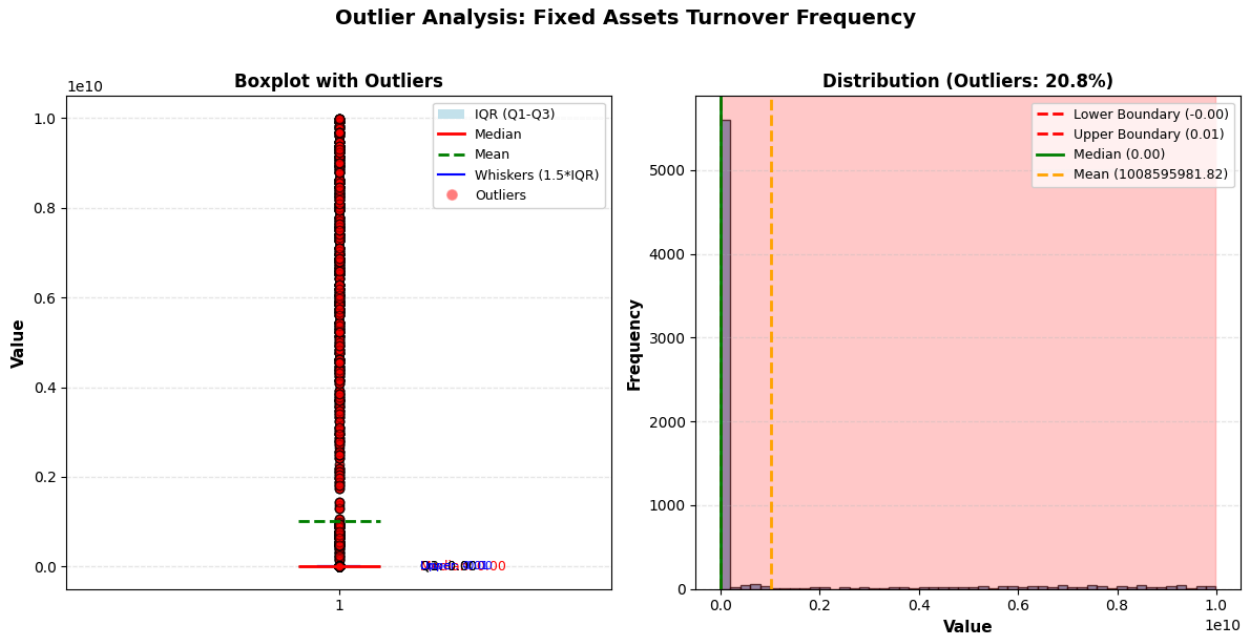


Figure 5.1.3.26 Boxplot and distribution view of Fixed Assets Turnover Frequency

Outlier Analysis for 'Fixed Assets Turnover Frequency'

Number of outliers: 1418 (20.79%)

In the boxplot, almost all non-extreme observations are essentially 0, while a vertical line of outliers rises up to about 1.0×10^{10} ; Q1, median, and Q3 are all near 0, but the mean is pulled up towards 0.1 due to those huge spikes.

The histogram shows the bulk of data crammed between the IQR bounds near 0–0.01, yet about 20.8% of records fall in the shaded outlier region and push the mean to roughly 1.0×10^{10} , indicating that calculation or scaling errors overwhelm any meaningful turnover information.

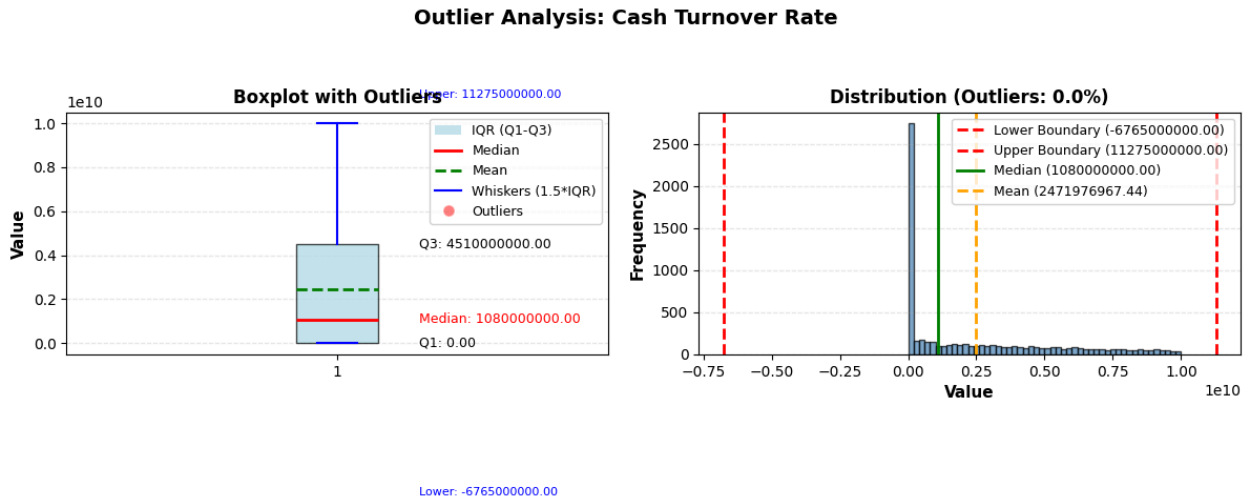


Figure 5.1.3.27 Boxplot and distribution view of Cash Turnover Rate

Outlier Analysis for 'Cash Turnover Rate'

Number of outliers: 0 (0.00%)

In the boxplot, Q1 is 0 and Q3 is about 4.51×10^{10} , with a median around 1.08×10^{10} and whiskers stretching between roughly -6.76×10^{10} and 1.13×10^{11} ; this enormous spread is implausible for a turnover ratio.

The histogram shows most observations distributed across this huge scale with IQR bounds identical to the extreme lower and upper limits and a mean around 2.47×10^{10} , indicating that the entire feature is skewed by extreme values rather than reflecting a realistic turnover behaviour.

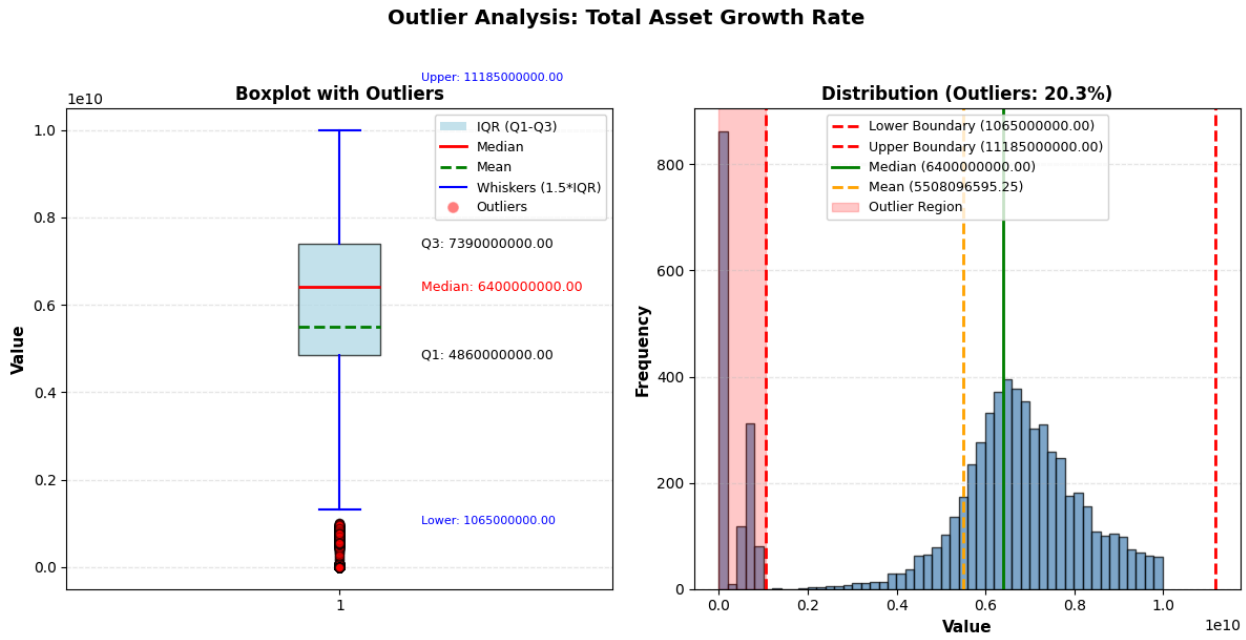


Figure 5.1.3.28 Boxplot and distribution view of Total Asset Growth Rate

Outlier Analysis for 'Total Asset Growth Rate'

Number of outliers: 1381 (20.25%)

In the boxplot, Q1 is around 4.86×10^{9} , Q3 about 7.39×10^{9} , and the median roughly 6.4×10^{9} , while the whiskers extend from roughly 1.06×10^{9} up to 1.1185×10^{10} ; a few smaller values near 0.1 billion are flagged as red outliers.

The histogram shows a concentration of firms in the mid-range (around $0.5–0.8\times10^{10}$), but 20.3% of records fall in the shaded outlier region outside the IQR-based bounds, indicating that asset growth is highly variable and that trimming or winsorising those extreme low and high growth rates would help stabilise this feature before modelling.

5.1.4. Capping of outliers

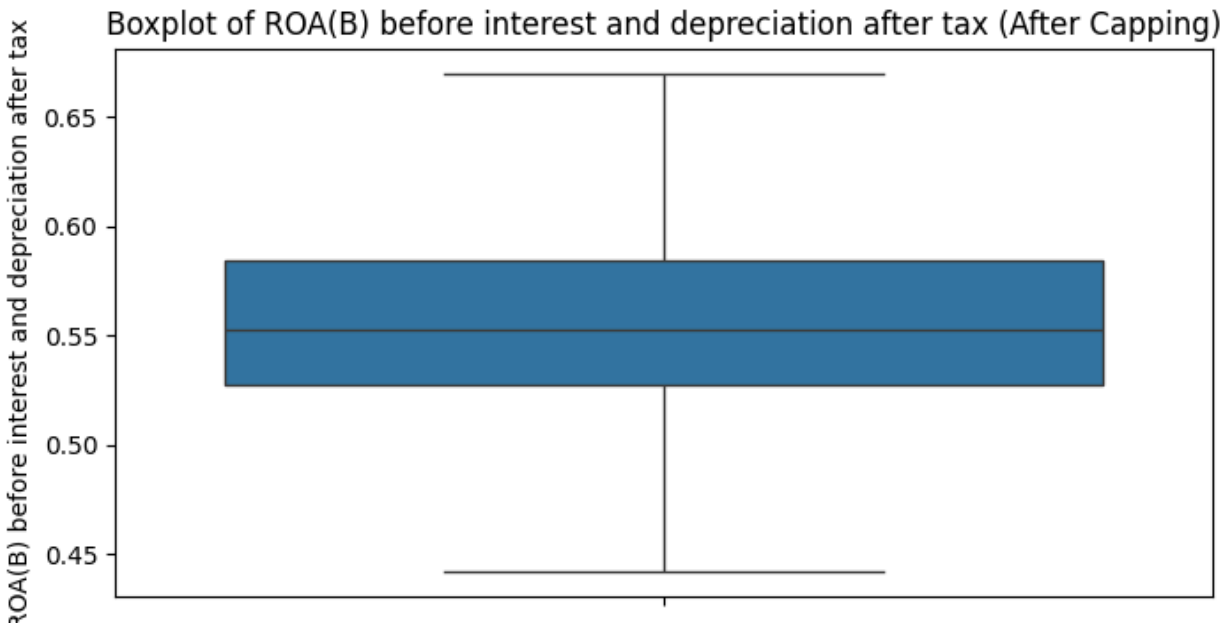


Figure 5.1.4.1 Boxplot of ROA(B) before interest and depreciation after tax(After Capping)

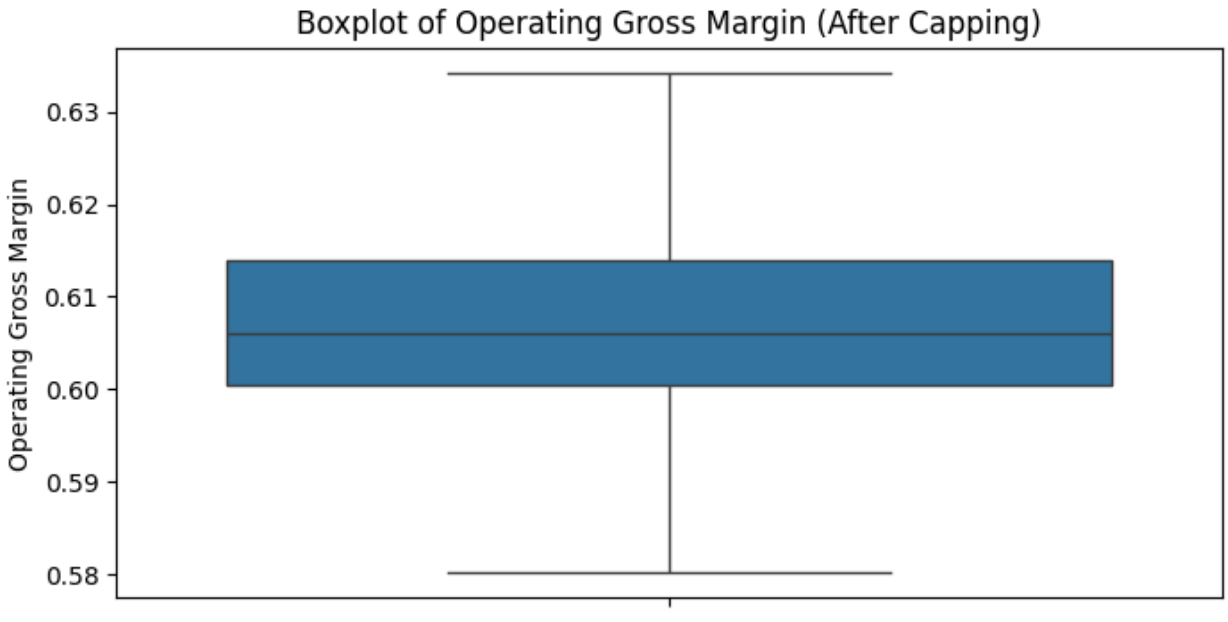


Figure 5.1.4.2 Boxplot of Operating Gross Margin (After Capping)

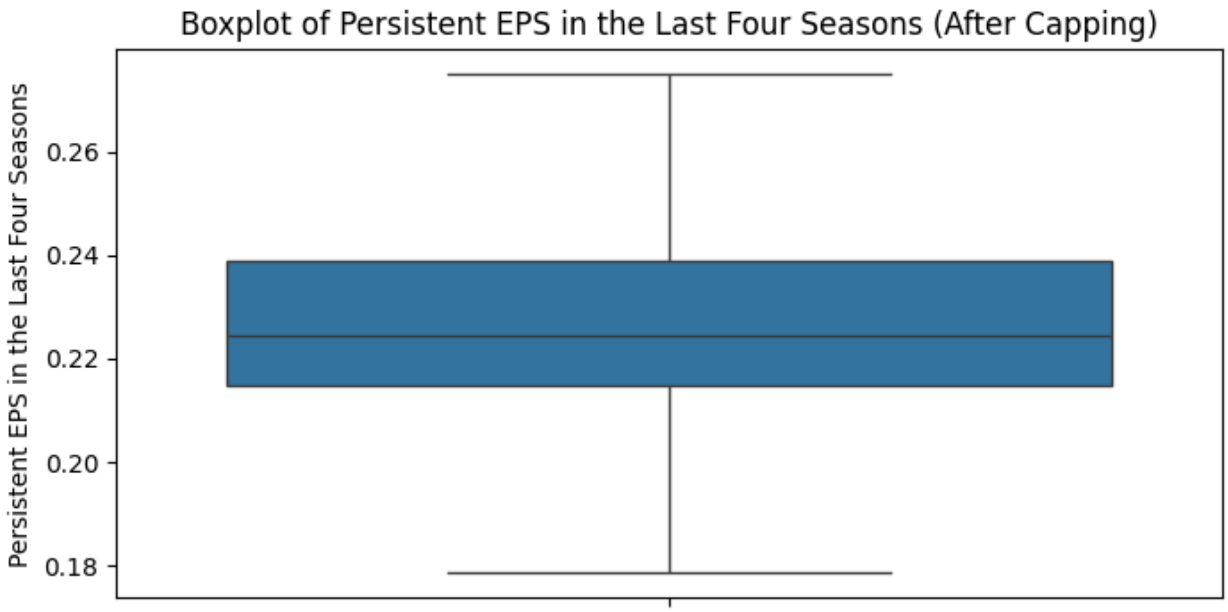


Figure 5.1.4.3 Boxplot of Persistent EPS in the Last four seasons (After Capping)

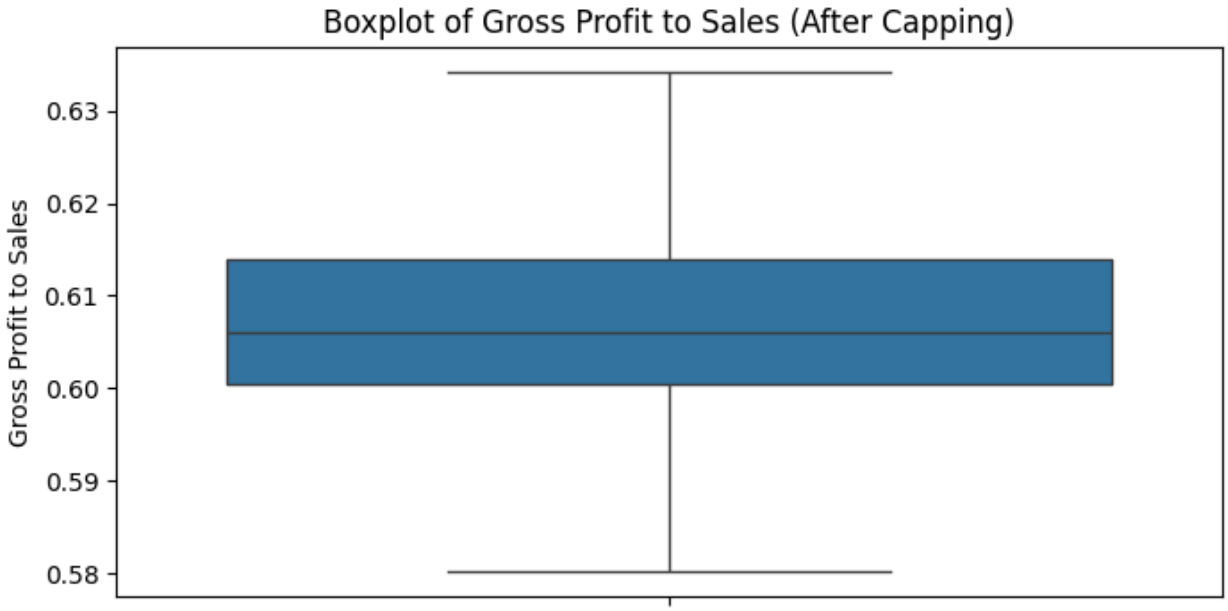


Figure 5.1.4.4 Boxplot of Persistent Gross Profit to sales (After Capping)

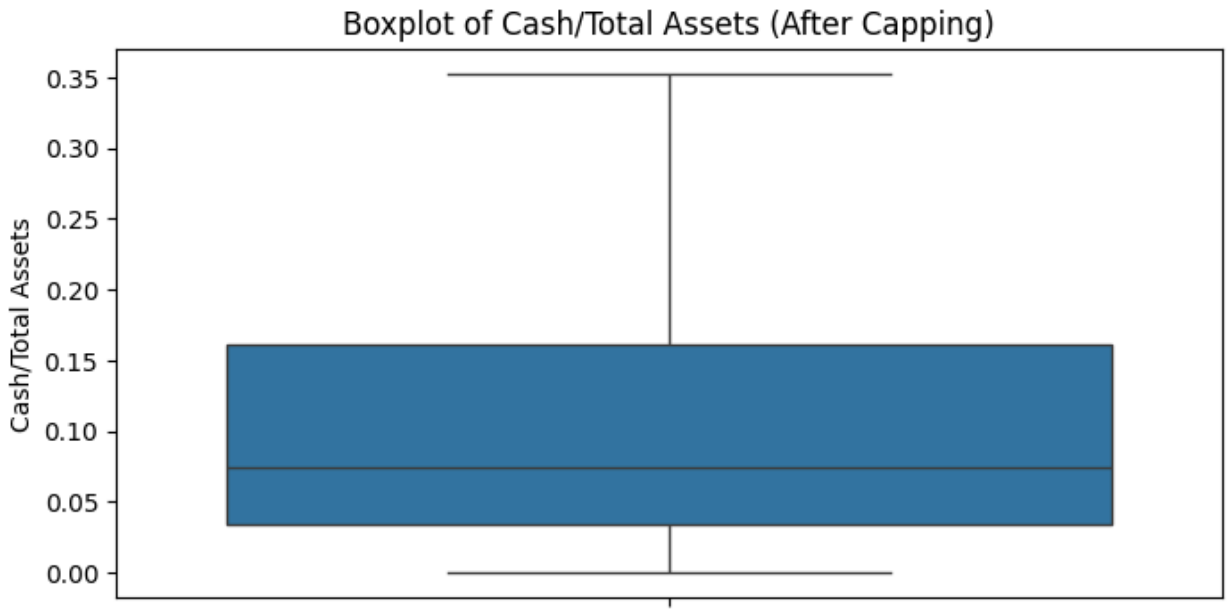


Figure 5.1.4.5 Boxplot of Cash/Total Assets (After Capping)

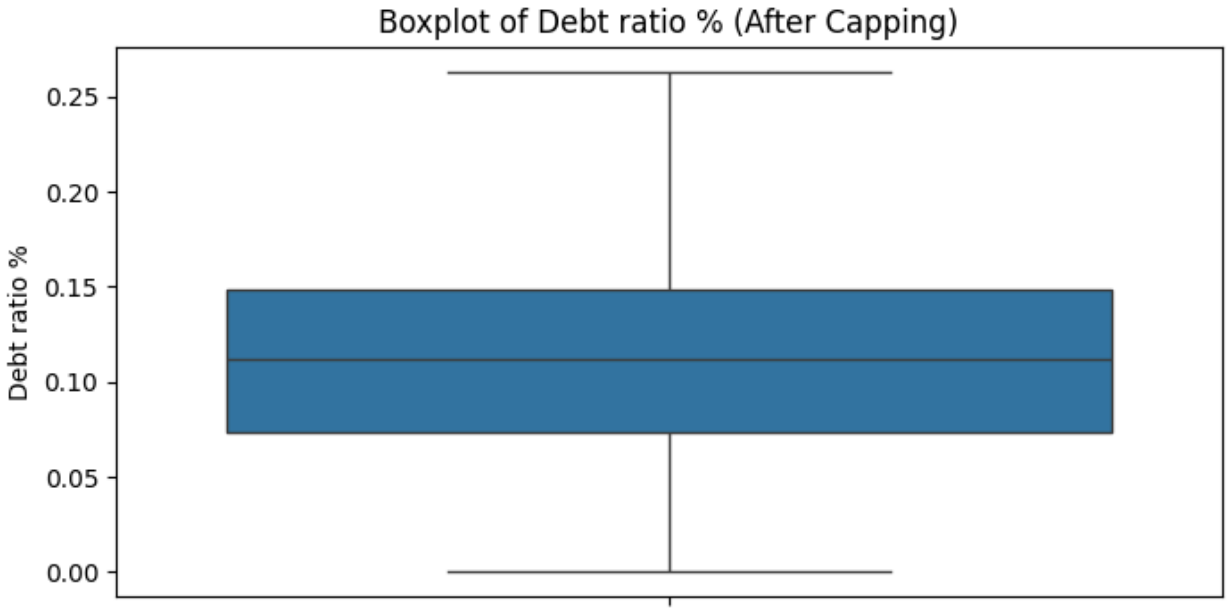


Figure 5.1.4.6 Boxplot of Debt Ratio (After Capping)

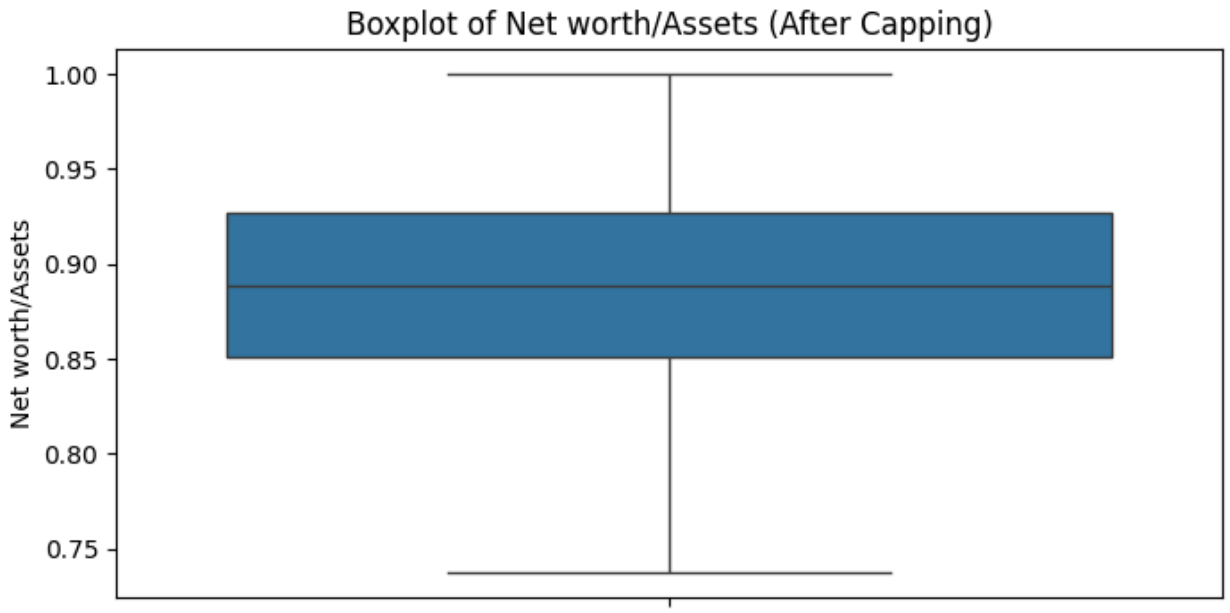


Figure 5.1.4.7 Boxplot of Net worth/Assets (After Capping)

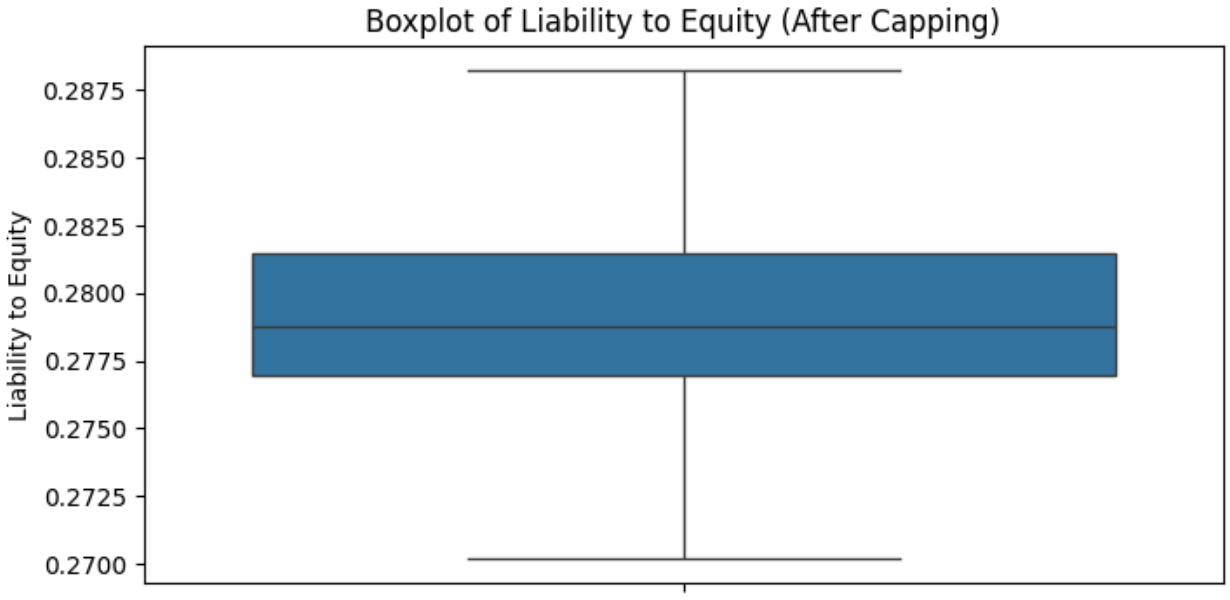


Figure 5.1.4.8 Boxplot of Liability to equity (After Capping)

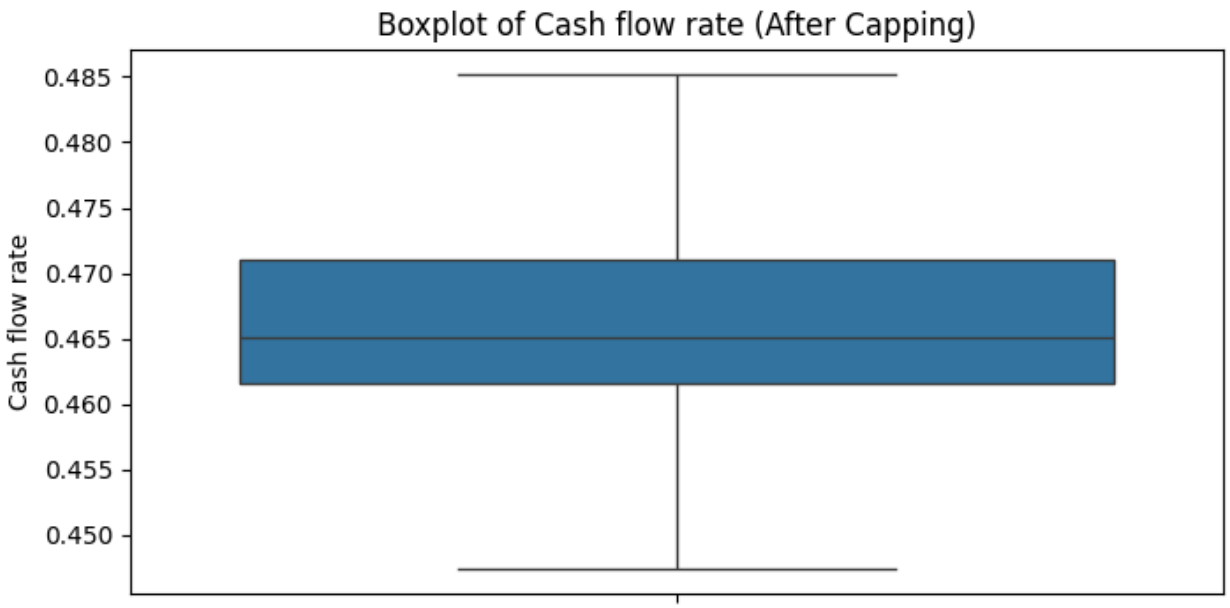


Figure 5.1.4.9 Boxplot of Cash Flow Rate (After Capping)

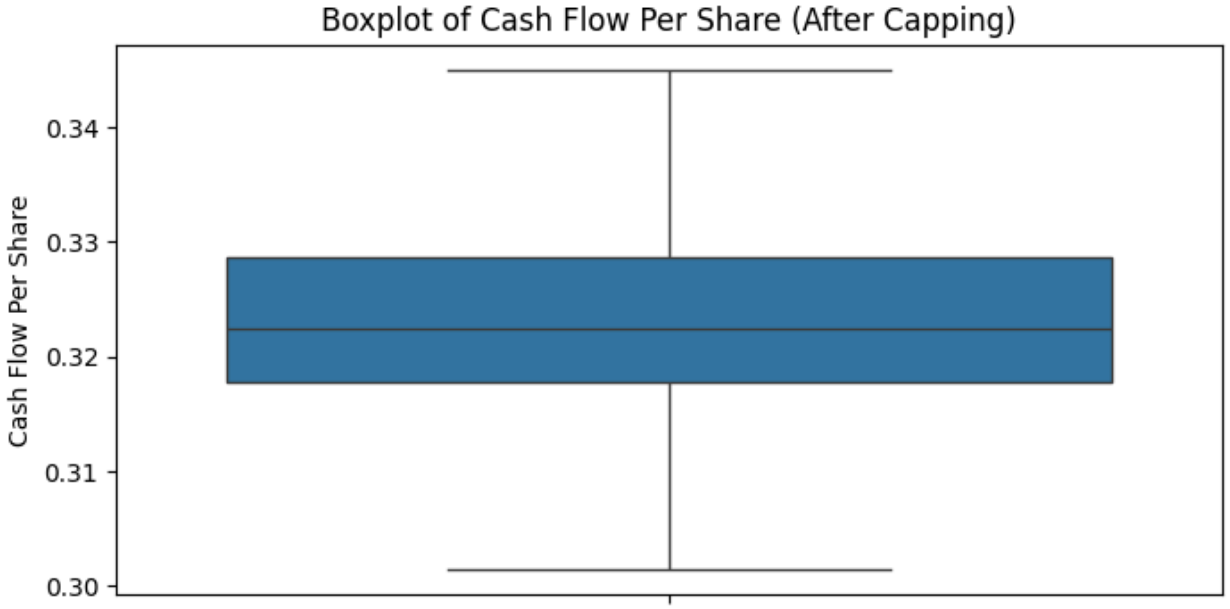


Figure 5.1.4.10 Boxplot of Cash Flow Per Share (After Capping)

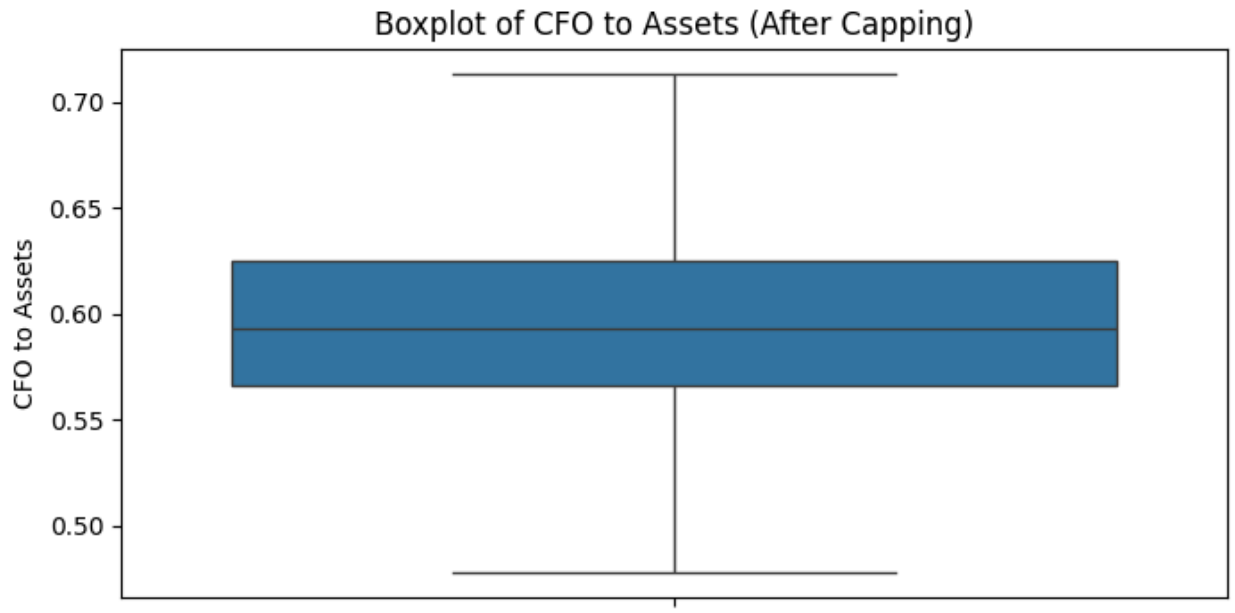


Figure 5.1.4.11 Boxplot of CFO to Assets (After Capping)

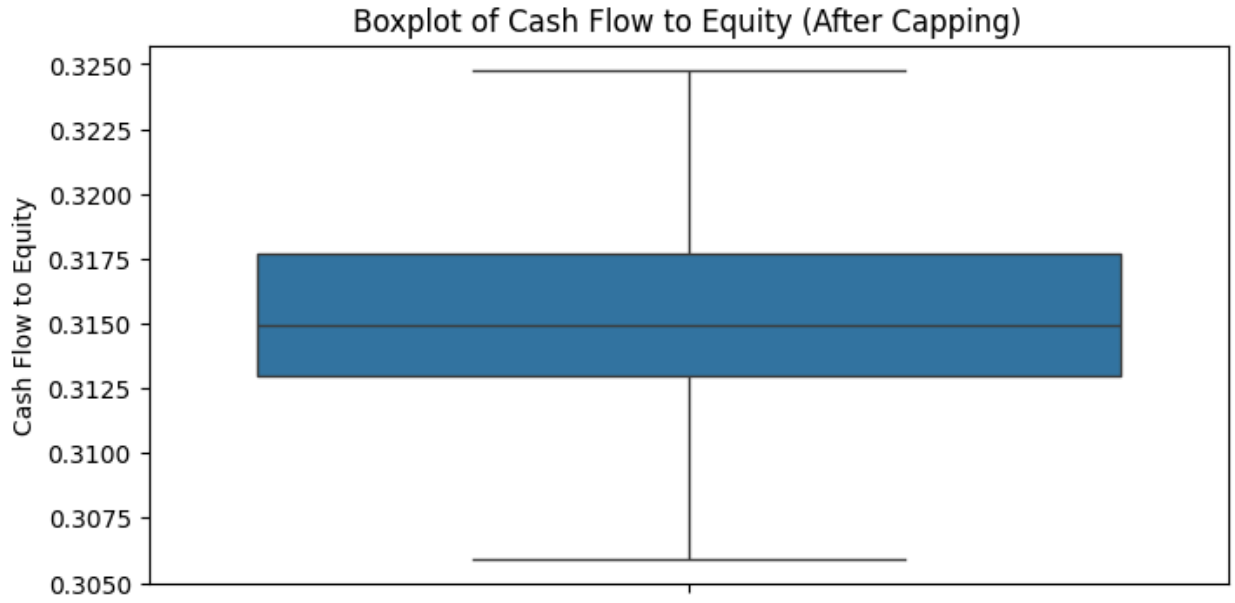


Figure 5.1.4.12 Boxplot of CFO to Assets (After Capping)

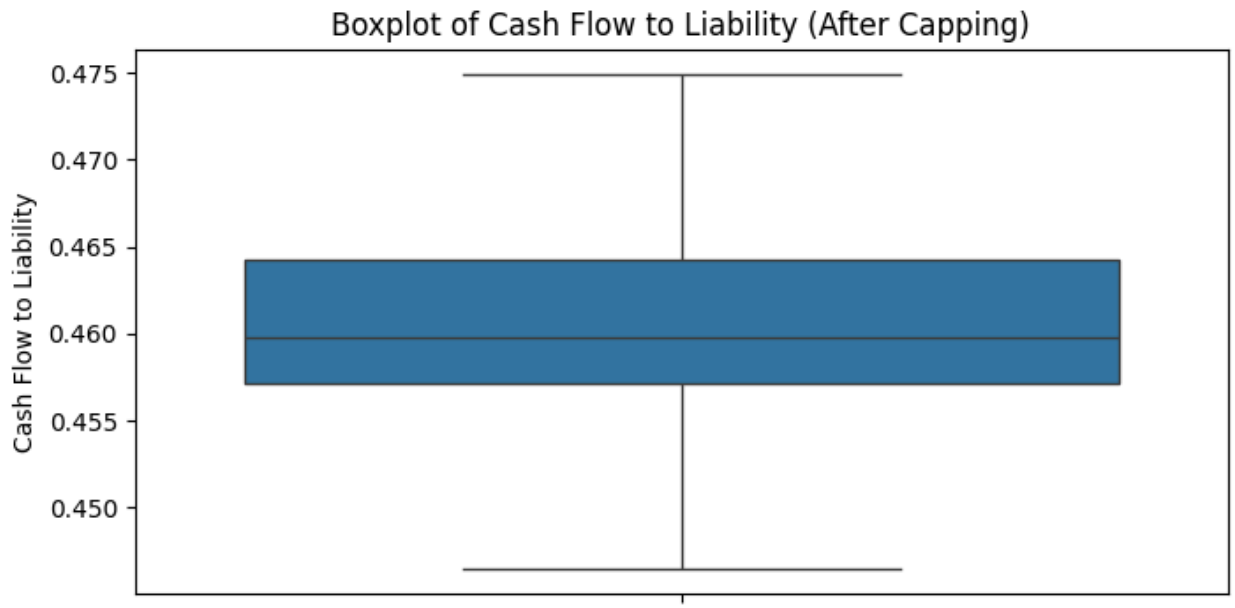


Figure 5.1.4.13 Boxplot of Cash Flow to Liability (After Capping)

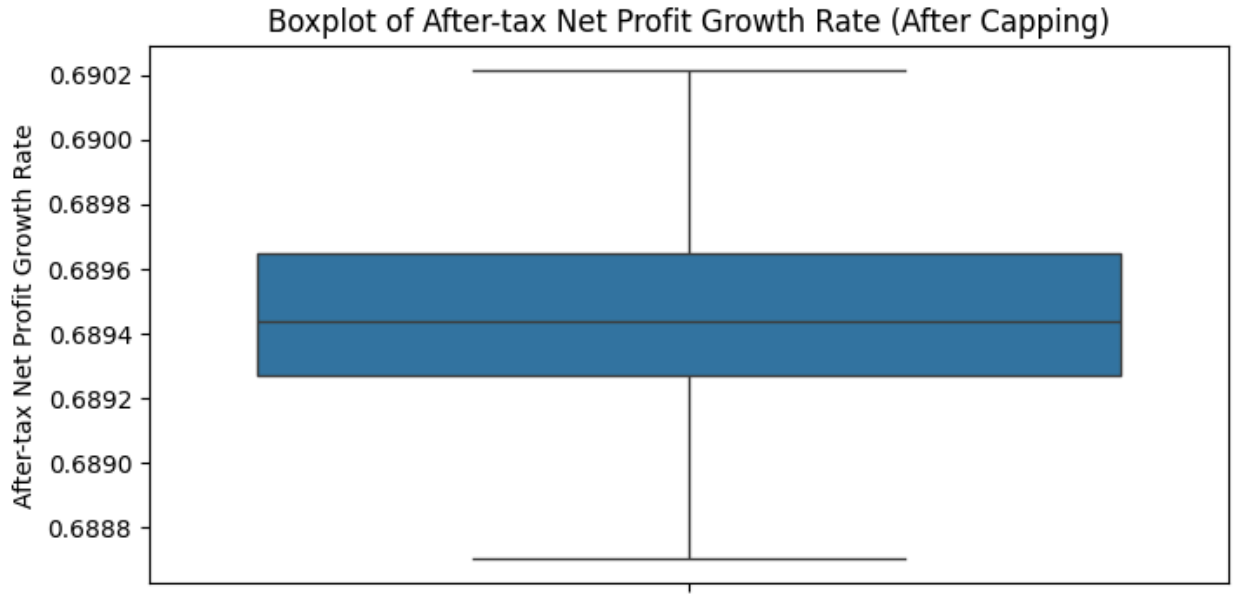


Figure 5.1.4.14 Boxplot of After-tax Net Profit Growth Rate (After Capping)

5.1.5. Binning for Continuous Data

Binning, also known as data binning or bucketing, is a data preprocessing technique used to group a set of continuous values into discrete intervals, called bins. This method simplifies data analysis by reducing the number of distinct values, making it easier to visualize and interpret the data.

● Equal width binning will be used to bin the continuous data.

● 10 bins will be created for each category to retain enough detail to avoid losing important information.

Key Advantages of Binning Continuous Data

Reduces the Effects of Noisy Data:

● Binning smooths out minor observation variances or noise in the data by grouping nearby data points into a bin. This can help reduce the impact of outliers or small fluctuations in the data that may not be meaningful, leading to more stable models.

Handles Non-linear Relationships:

● Continuous variables may have non-linear relationships with the target variable that can be difficult to model directly. Binning can help capture these relationships by dividing the data into intervals that reflect distinct behaviors or trends.

Improves Stability and Generalization:

● Binning continuous features can improve the stability of the model by reducing the sensitivity to small changes in the input values. This leads to more generalizable models that are less likely to overfit.

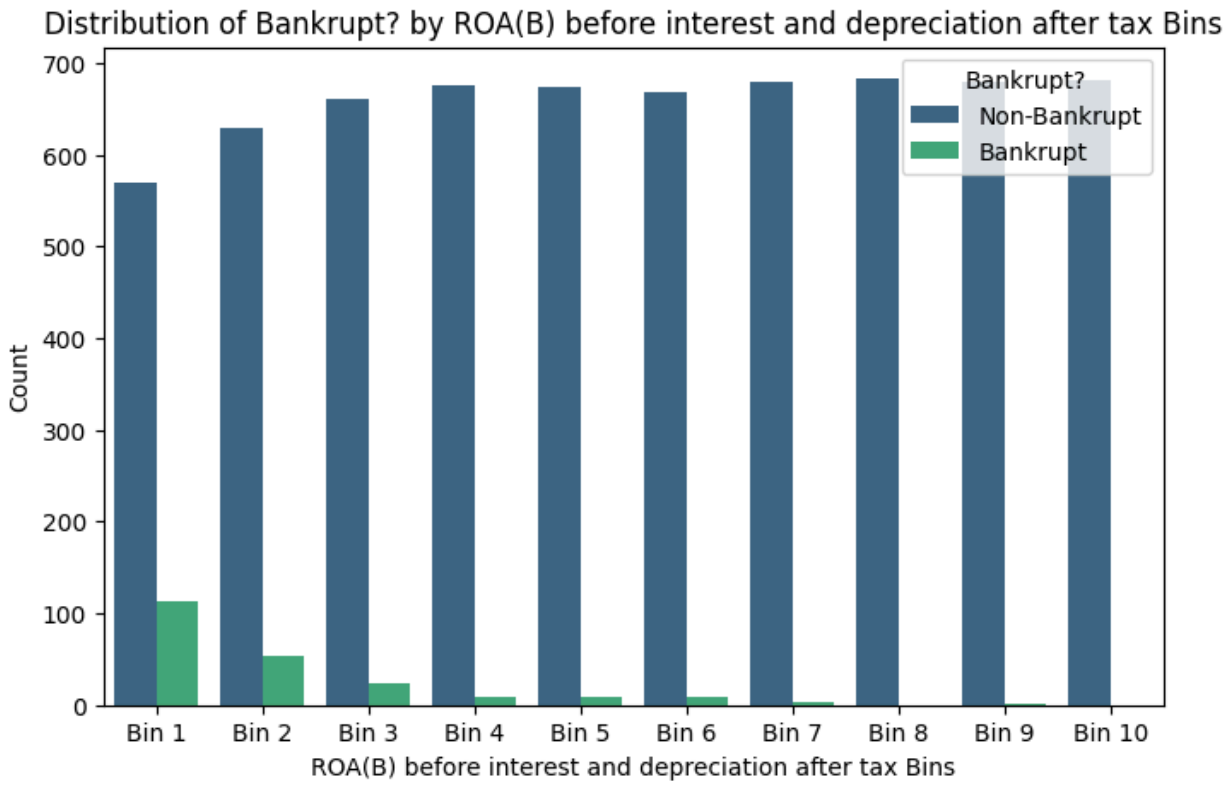


Figure 5.1.5.1 Bin edges for ROA(B) before interest and depreciation after tax:

Bin 1: 0.44203383 0.49178222

Bin 2: 0.49178222 0.52037047

Bin 3: 0.52037047 0.53300498

Bin 4: 0.53300498 0.54306976

Bin 5: 0.54306976 0.55227796

Bin 6: 0.55227796 0.56281386

Bin 7: 0.56281386 0.5759409

Bin 8: 0.5759409 0.59371487

Bin 9: 0.59371487 0.62058997

Bin 10: 0.62058997 0.66934793

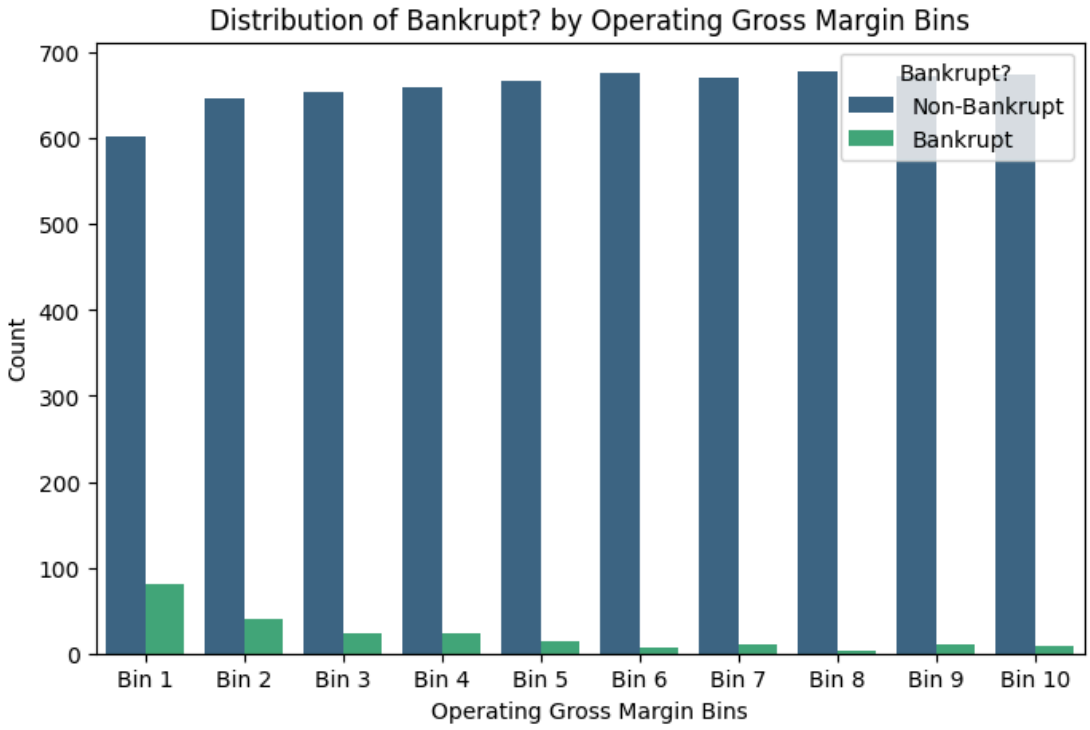


Figure 5.1.5.2 Bin edges for Operating Gross Margin:

Bin 1: 0.58024042 0.59651335

Bin 2: 0.59651335 0.59920869

Bin 3: 0.59920869 0.60149613

Bin 4: 0.60149613 0.6036697

Bin 5: 0.6036697 0.60599749

Bin 6: 0.60599749 0.60867842

Bin 7: 0.60867842 0.61191859

Bin 8: 0.61191859 0.61618934

Bin 9: 0.61618934 0.62315254

Bin 10: 0.62315254 0.63411839

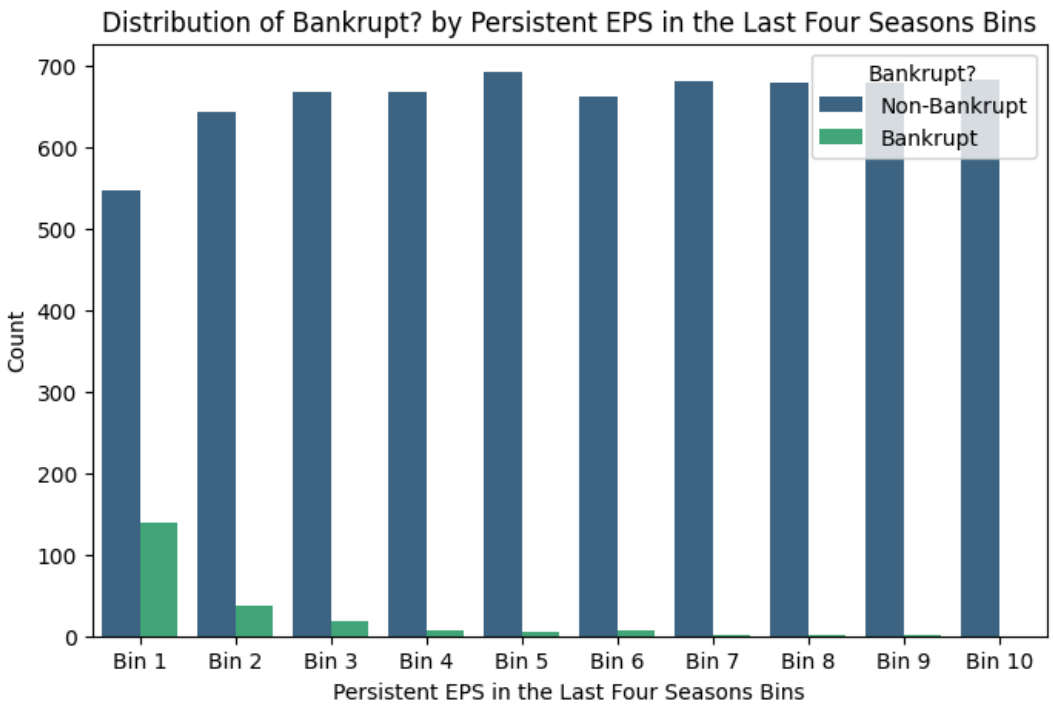


Figure 5.1.5.3 Bin edges for Persistent EPS in the Last Four Seasons:

Bin 1: 0.17854779 0.20109672

Bin 2: 0.20109672 0.21121301

Bin 3: 0.21121301 0.21650752

Bin 4: 0.21650752 0.22030822

Bin 5: 0.22030822 0.22454382

Bin 6: 0.22454382 0.22898743

Bin 7: 0.22898743 0.23466011

Bin 8: 0.23466011 0.24364186

Bin 9: 0.24364186 0.25878794

Bin 10: 0.25878794 0.27498345

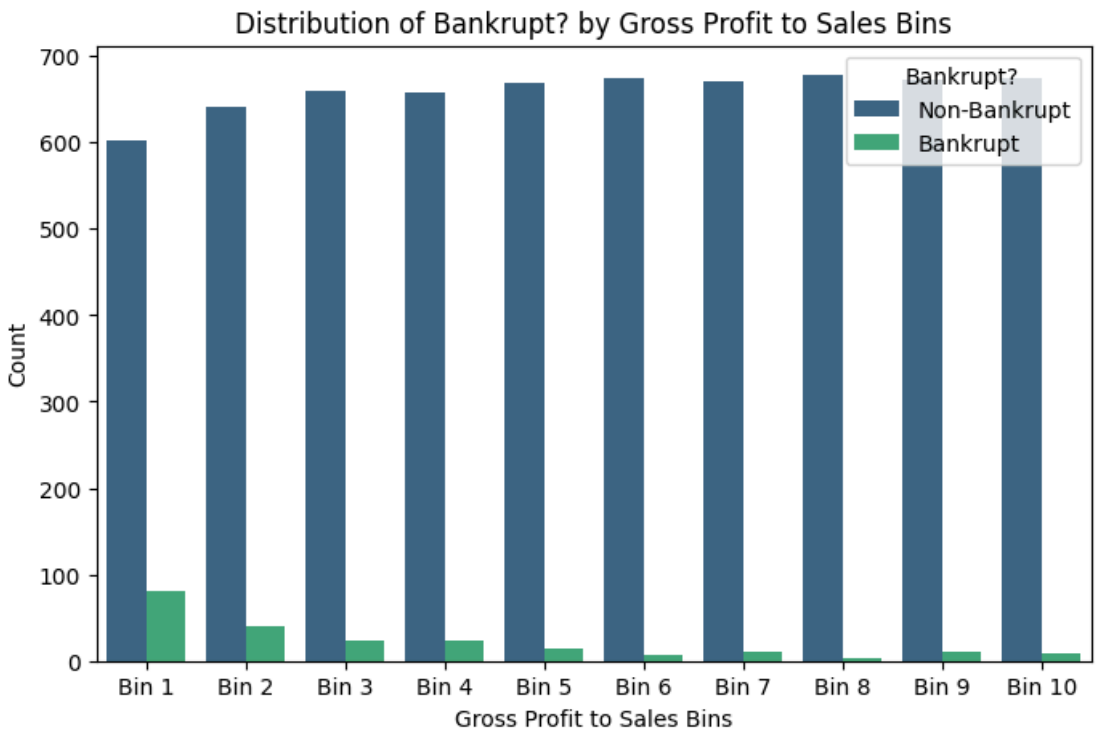


Figure 5.1.5.4 Bin edges for Gross Profit to Sales:

Bin 1:0.58023733 0.59651368

Bin 2: 0.59651368 0.59920487

Bin 3: 0.59920487 0.60149628

Bin 4: 0.60149628 0.60367024

Bin 5: 0.60367024 0.60599829

Bin 6: 0.60599829 0.608679

Bin 7: 0.608679 0.61191669

Bin 8: 0.61191669 0.61618595

Bin 9: 0.61618595 0.62315227

Bin 10: 0.62315227 0.63411883

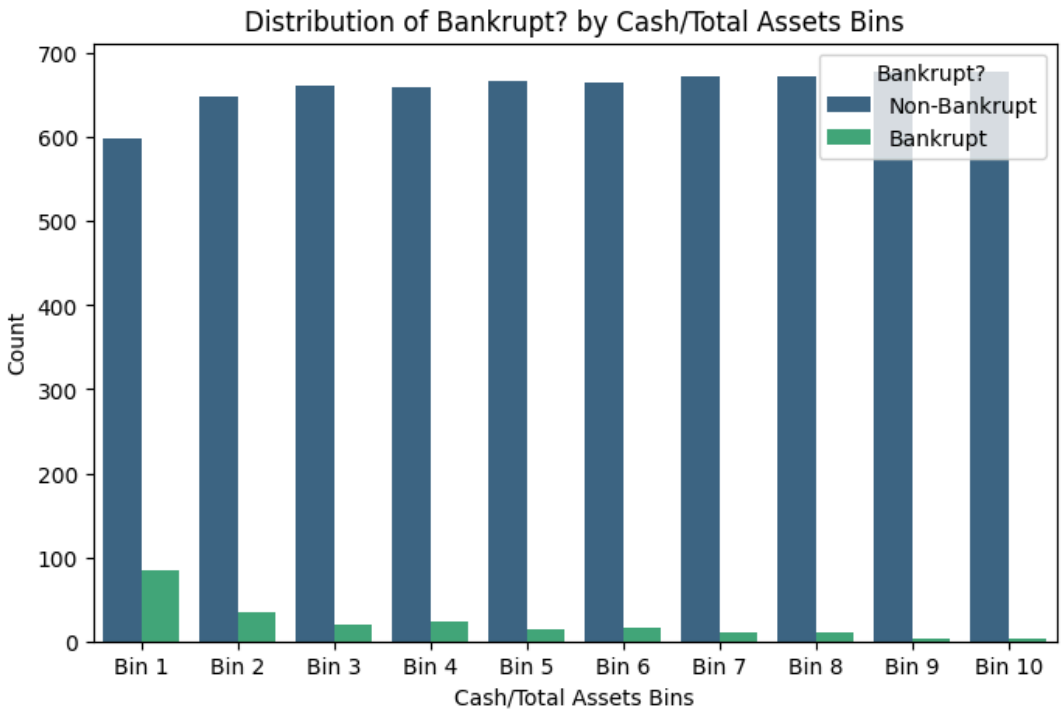


Figure 5.1.5.5 Bin edges for Cash/Total Assets:

Bin 1 : 0.0 0.0145802

Bin 2: 0 0.0145802 0.02702683

Bin 3: 0.02702683 0.0395681

Bin 4: 0.0395681 0.0557117

Bin 5: 0.0557117 0.07488746

Bin 6: 0.07488746 0.09996371

Bin 7: 0.09996371 0.13592449

Bin 8: 0.13592449 0.1929289

Bin 9: 0.1929289 0.29958556

Bin 10 : 0.29958556 0.35236805

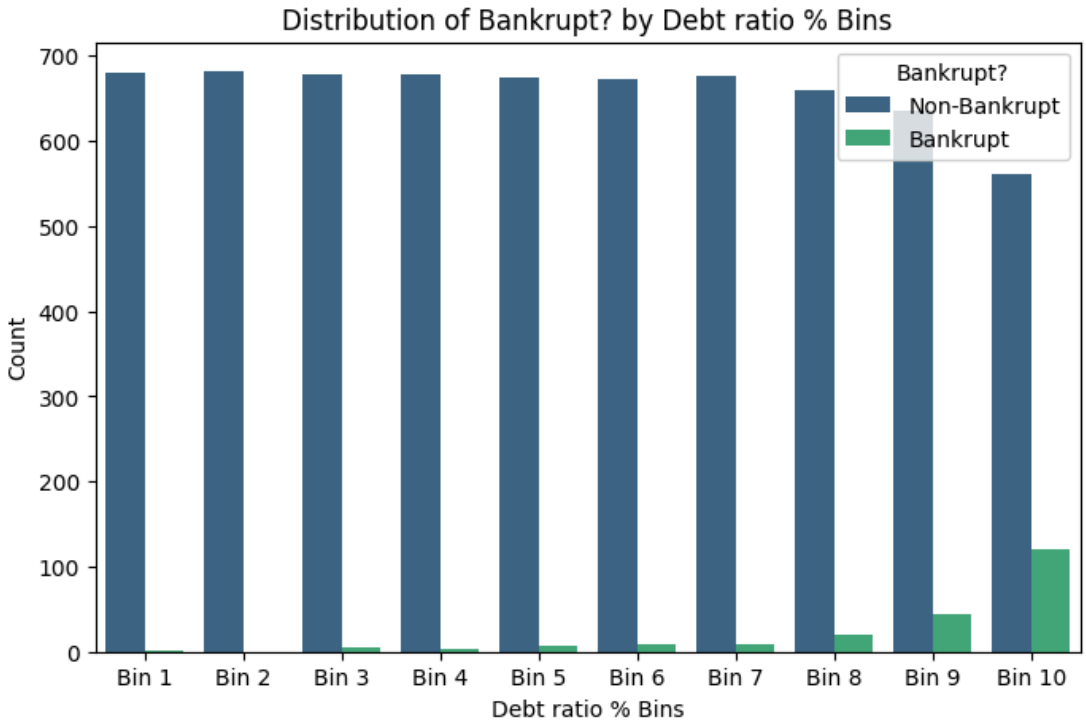


Figure 5.1.5.6 Bin edges for Debt ratio %:

Bin 1 : 0.0 0.04405478

Bin 2 : 0.04405478 0.06369381

Bin 3 : 0.06369381 0.08099283

Bin 4 : 0.08099283 0.09737884

Bin 5 : 0.09737884 0.11140672

Bin 6 : 0.11140672 0.12554343

Bin 7 : 0.12554343 0.14073223

Bin 8 : 0.14073223 0.15823684

Bin 9 : 0.15823684 0.18393446

Bin 10 : 0.18393446 0.26267497

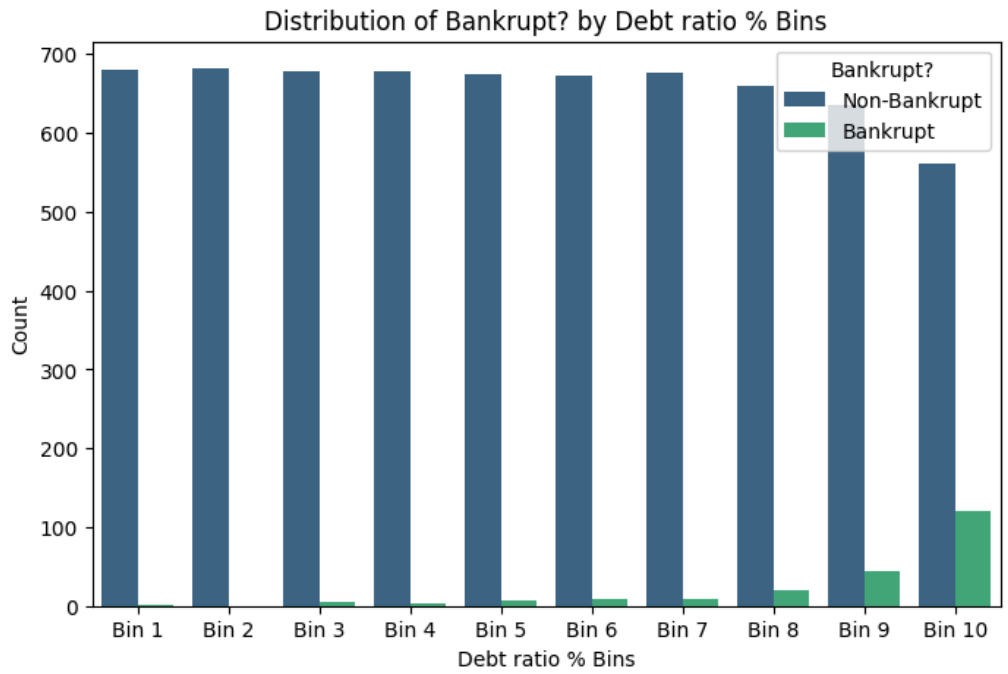


Figure 5.1.5.7 Bin edges for Net worth/Assets:

Bin 1 : 0.73732503 0.81606554

Bin 2 : 0.81606554 0.84176316

Bin 3 : 0.84176316 0.85926777

Bin 4 : 0.85926777 0.87445657

Bin 5 : 0.87445657 0.88859328

Bin 6 : 0.88859328 0.90262116

Bin 7 : 0.90262116 0.91900717

Bin 8 : 0.91900717 0.93630619

Bin 9 : 0.93630619 0.95594522

Bin 10 : 0.95594522 1.0

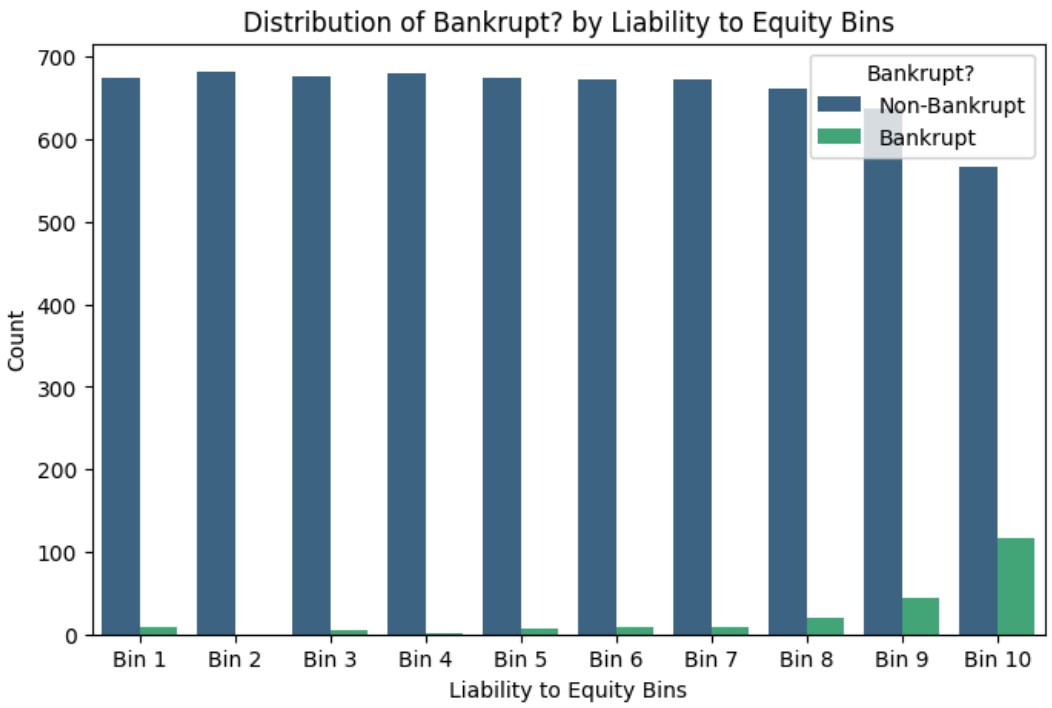


Figure 5.1.5.8 Bin edges for Liability to Equity:

Bin 1 : 0.27018683 0.27593864

Bin 2 : 0.27593864 0.2765891

Bin 3 : 0.2765891 0.27727823

Bin 4 : 0.27727823 0.27802844

Bin 5 : 0.27802844 0.27877758

Bin 6 : 0.27877758 0.27964924

Bin 7 : 0.27964924 0.28077578

Bin 8 : 0.28077578 0.28236489

Bin 9 : 0.28236489 0.28557324

Bin 10 : 0.28557324 0.2882066

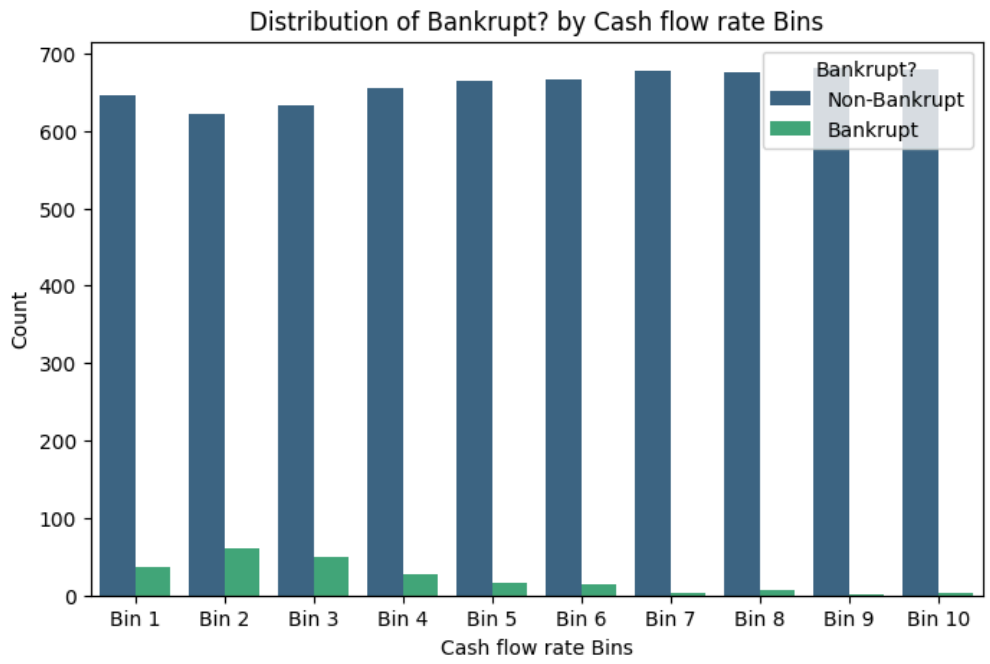


Figure 5.1.5.9 Bin edges for Cash flow rate:

Bin 1 : 0.44738851 0.45787507

Bin 2 : 0.45787507 0.46061649

Bin 3 : 0.46061649 0.46219773

Bin 4 : 0.46219773 0.46361021

Bin 5 : 0.46361021 0.46507972

Bin 6 : 0.46507972 0.46687636

Bin 7 : 0.46687636 0.46942486

Bin 8 : 0.46942486 0.47302316

Bin 9 : 0.47302316 0.48004798

Bin 10 : 0.48004798 0.48517316

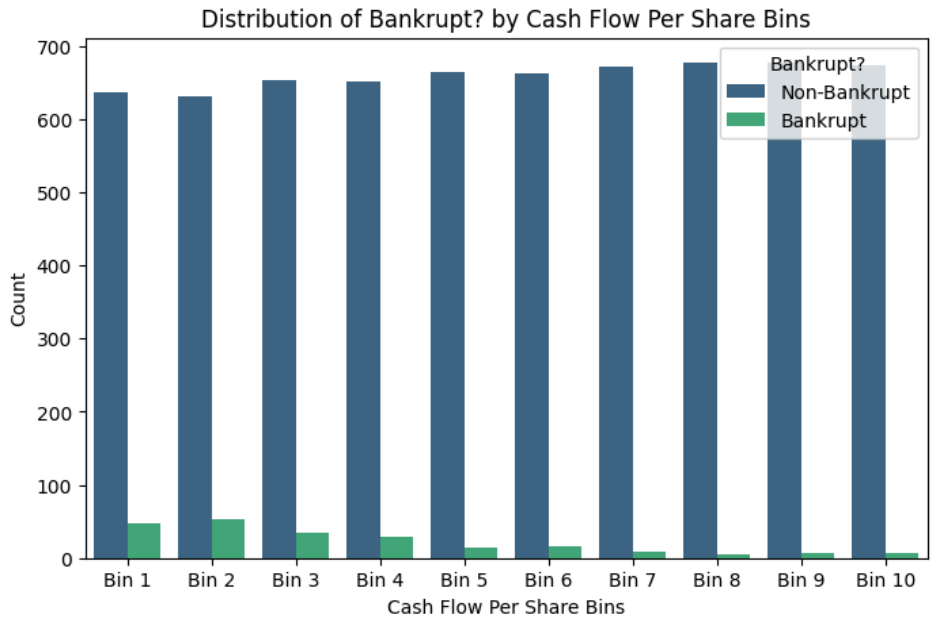


Figure 5.1.5.10 Bin edges for Cash Flow Per Share:

Bin 1 : 0.30143418 0.31159369

Bin 2 : 0.31159369 0.31629766

Bin 3 : 0.31629766 0.31877343

Bin 4 : 0.31877343 0.32057721

Bin 5 : 0.32057721 0.32248709

Bin 6 : 0.32248709 0.32453845

Bin 7 : 0.32453845 0.32714154

Bin 8 : 0.32714154 0.33052982

Bin 9 : 0.33052982 0.33702341

Bin 10 : 0.33702341 0.34493704

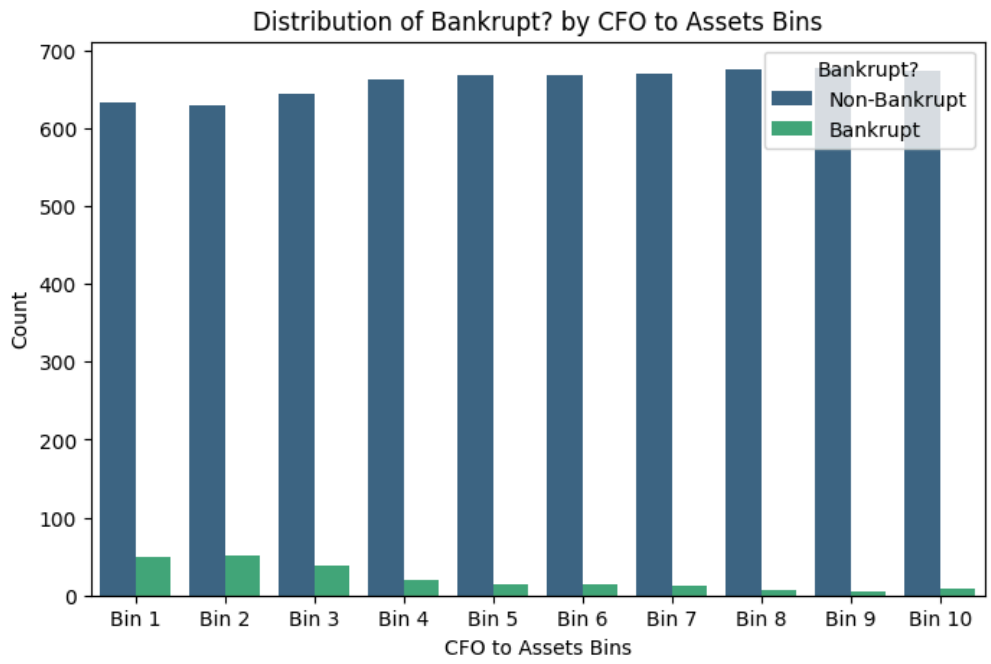


Figure 5.1.5.11 Bin edges for CFO to Assets:

Bin 1 : 0.47781404 0.53005064

Bin 2 : 0.53005064 0.55727966

Bin 3 : 0.55727966 0.57196395

Bin 4 : 0.57196395 0.58245729

Bin 5 : 0.58245729 0.59326627

Bin 6 : 0.59326627 0.60404164

Bin 7 : 0.60404164 0.61759178

Bin 8 : 0.61759178 0.63352483

Bin 9 : 0.63352483 0.65850202

Bin 10 : 0.65850202 0.71294178

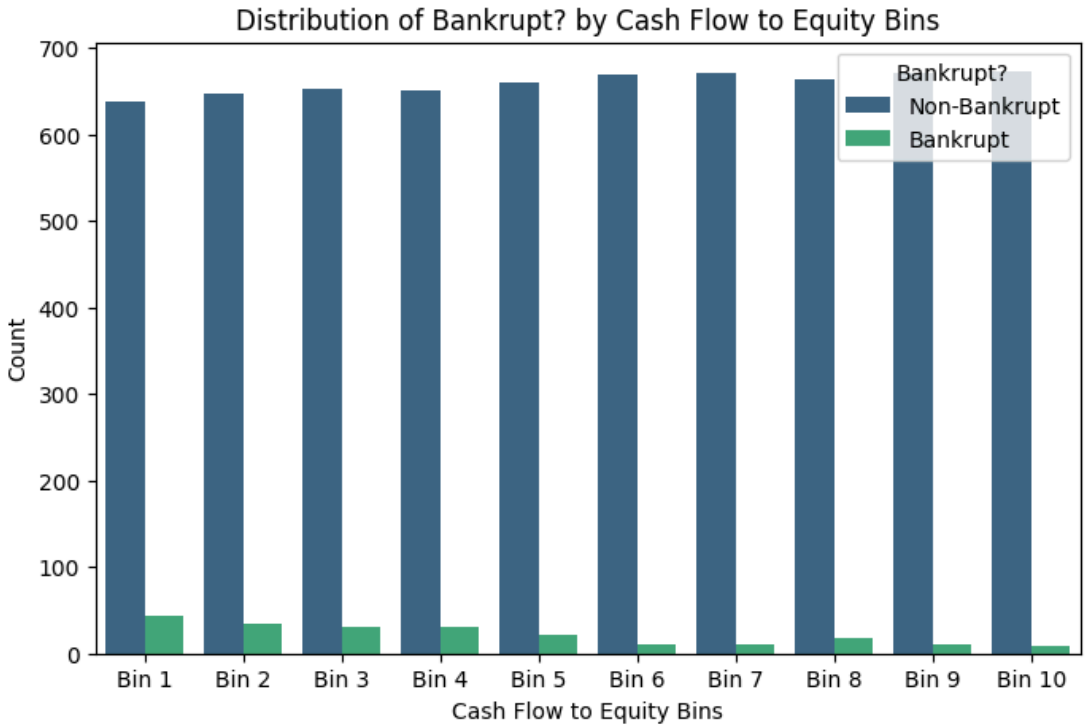


Figure 5.1.5.12 Bin edges for Cash Flow to Equity:

Bin 1 : 0.30592597 0.30941248

Bin 2 : 0.30941248 0.31217874

Bin 3 : 0.31217874 0.31356696

Bin 4 : 0.31356696 0.31437364

Bin 5 : 0.31437364 0.31495275

Bin 6 : 0.31495275 0.31574343

Bin 7 : 0.31574343 0.31684077

Bin 8 : 0.31684077 0.31876186

Bin 9 : 0.31876186 0.32277424

Bin 10 : 0.32277424 0.32477592

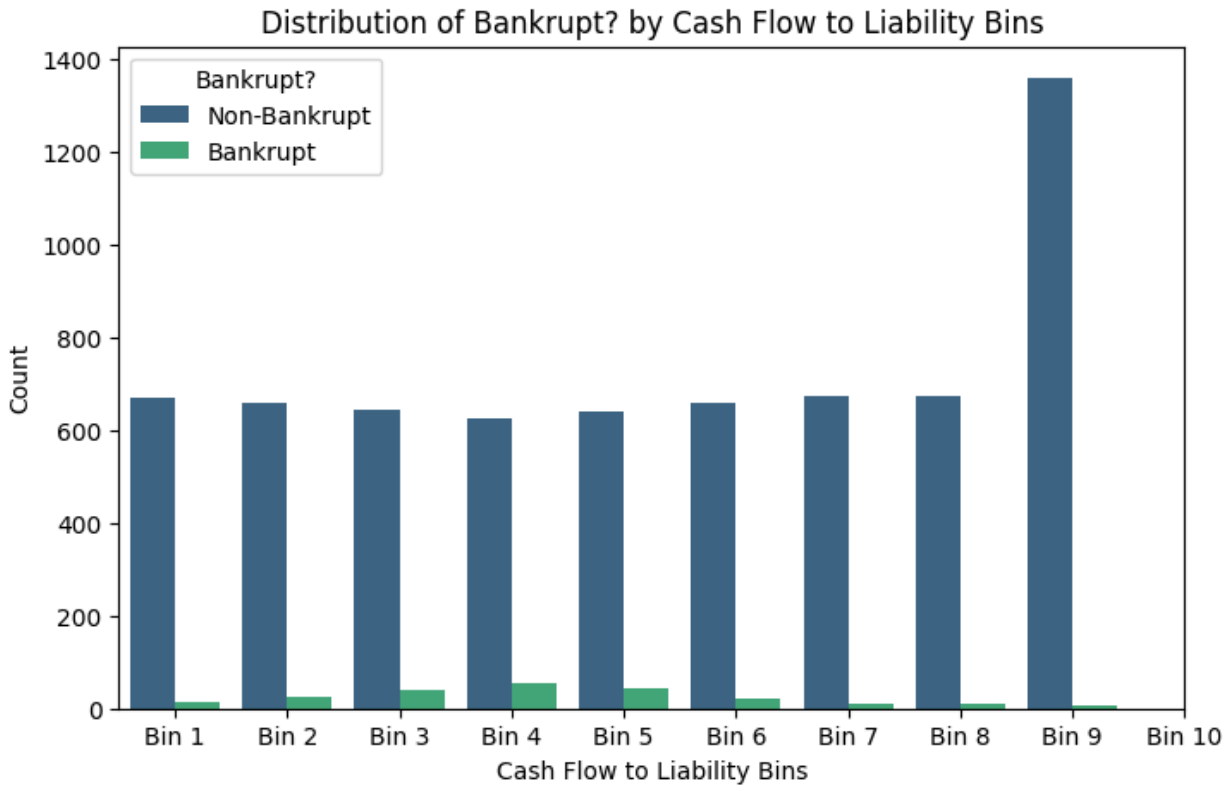


Figure 5.1..5.13 Bin edges for Cash Flow to Liability:

Bin 1 : 0.44643742 0.45023401

Bin 2 : 0.45023401 0.4556871

Bin 3 : 0.4556871 0.4580273

Bin 4 : 0.4580273 0.45904314

Bin 5 : 0.45904314 0.45975014

Bin 6 : 0.45975014 0.46087069

Bin 7 : 0.46087069 0.462796

Bin 8 : 0.462796 0.46630626

Bin 9 : 0.46630626 0.4749149

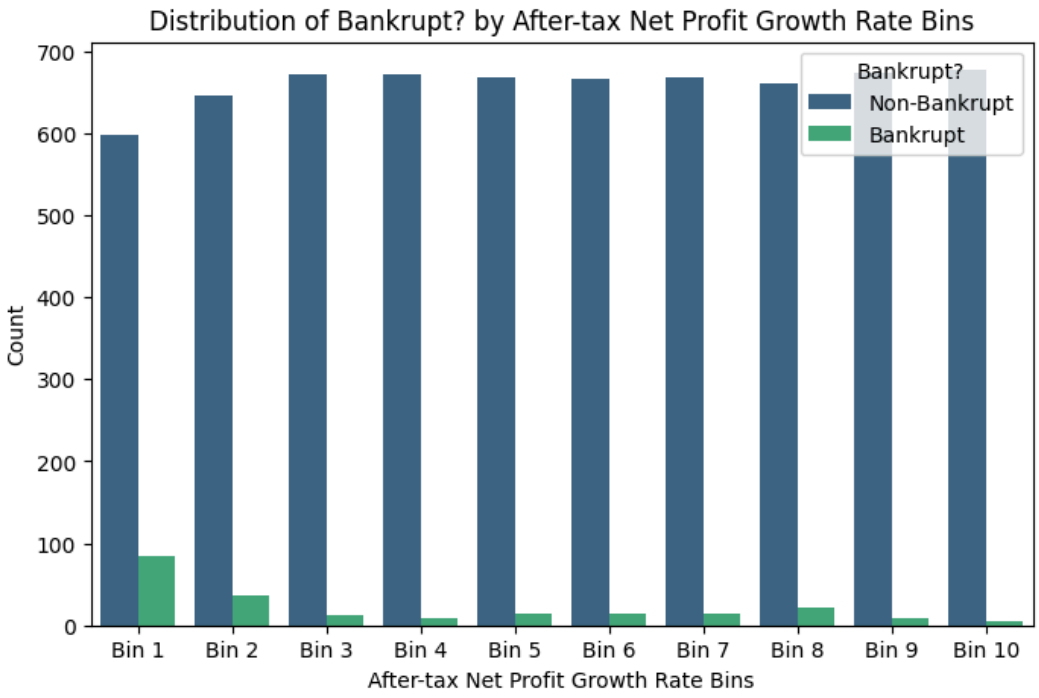


Figure 5.1.5.14 Bin edges for After-tax Net Profit Growth Rate:

Bin 1 : 0.68870408 0.68889187

Bin 2 : 0.68889187 0.6892156

Bin 3 : 0.6892156 0.68931319

Bin 4 : 0.68931319 0.68938179

Bin 5 : 0.68938179 0.68943853

Bin 6 : 0.68943853 0.68949403

Bin 7 : 0.68949403 0.68958595

Bin 8 : 0.68958595 0.68972297

Bin 9 : 0.68972297 0.69001404

Bin 10 : 0.69001404 0.69021302

5.1.6. Drop Unnecessary Features

Net Income Flag

Since all firms, both non-bankrupt (6,599) and bankrupt (220), have Net Income Flag = 1; there are no observations with another value hence value will drop.

Liability-Assets Flag

The "1" category is vanishingly rare in both classes, so the model will learn almost nothing from it and may even overfit those few cases. Hence it will drop.

ROA(A) before interest and % after tax

Almost perfectly correlated with other ROA measures and adds no new signal; one ROA is enough.

ROA(C) before interest and depreciation before interest

Same reason as ROA(A): near-duplicate of ROA(B), just increases multicollinearity.

Operating Profit Per Share (Yuan ¥)

Highly collinear with Persistent EPS and Per Share Net Profit, all measuring very similar profitability.

Per Share Net profit before tax (Yuan ¥)

Again a near-duplicate earnings-per-share metric; profitability is already well represented.

Net Income to Stockholder's Equity

Very similar information to ROA and EPS metrics; strong correlation cluster with them, so it is redundant.

Operating Profit Rate

Outlier analysis shows almost all values at ~0.999 with tiny IQR, so the variable is effectively constant and dominated by "outliers".

Pre-tax net Interest Rate

Central distribution collapsed around ~0.80 with very narrow IQR and ~11% outliers; almost no useful variation relative to other risk drivers.

After-tax net Interest Rate

Same pattern as pre-tax rate (almost constant at ~0.81), providing little incremental information.

Degree of Financial Leverage (DFL)

Core values almost all at ~0.03 with 22% outliers; very low information content compared with more interpretable leverage ratios.

Long-term fund suitability ratio (A)

Nearly all values around 0.01 with a noisy tail up to 1.0; behaves like a near-constant plus noisy extremes.

Current Ratio

Severe data issues: most values near 0, but a few in the order of 10^9 massively distort the mean and any regression; similar liquidity captured by cleaner variables.

Quick Ratio

Same problem as Current Ratio (tiny normal values plus unrealistically huge spikes); also redundant given other cash/liquidity ratios.

Quick Assets/Current Liability

Mirrors Quick Ratio's extreme outliers in the billions, with almost no variation in the normal range.

Cash/Current Liability

Same extreme-outlier pattern as the other current-liability ratios, with most values ~0 and some in the 10^9 range.

Total debt/Total net worth

Distribution heavily distorted by a small set of values around 10^{10} ; most observations near 0; cleaner leverage captured by Debt ratio % and Liability to Equity.

Cash Flow to Sales

Core distribution collapsed at ~0.67 with IQR essentially zero and 15% outliers; acts like a constant plus noise.

Accounts Receivable Turnover

Almost all values near 0, with ~10% extreme spikes up to 10^{10} ; clear data-quality/scaling issue.

Inventory Turnover Rate

Similar corruption: median at 0, but IQR and whiskers extend to around 10^{11} ; behaviour inconsistent with a realistic turnover ratio.

Fixed Assets Turnover Frequency

Most observations are exactly 0, but many spikes up to 10^{10} ; mean completely dominated by these errors.

Cash Turnover Rate

Same pattern as inventory and fixed-asset turnover: extremely large, implausible values and no stable central range.

Net Value Growth Rate

Q1, median, and Q3 at 0 with a set of huge positive spikes; growth information is essentially “zero vs extreme error”.

Total Asset Growth Rate

Very wide, skewed distribution with many extreme values and 20% outliers; growth signal is noisy and partly driven by apparent scaling issues.

After dropping attributes:

Data A

	Bankrupt?	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Persistent EPS in the Last Four Seasons	Gross Profit to Sales	Cash/Total Assets	Debt ratio %	Net worth/Assets	Liability to Equity	Cash flow rate	Cash Flow Per Share	CFO to Assets	Cash Flow to Equity	Cash Flow to Liability	After-tax Net Profit Growth Rate
0	1	0.442034	0.601457	0.178548	0.601453	0.004094	0.207576	0.792424	0.288207	0.458143	0.311664	0.520382	0.312905	0.458609	0.688979
1	1	0.516730	0.610235	0.208944	0.610237	0.014948	0.171176	0.828824	0.283846	0.461867	0.318137	0.567101	0.314163	0.459001	0.689693
2	1	0.472295	0.601450	0.180581	0.601449	0.000991	0.207516	0.792484	0.288207	0.458521	0.307102	0.538491	0.314515	0.459254	0.689463
3	1	0.457733	0.583541	0.193722	0.583538	0.018851	0.151465	0.848535	0.281721	0.465705	0.321674	0.604105	0.305926	0.448518	0.689110
4	1	0.522298	0.598783	0.212537	0.598782	0.014161	0.106509	0.893491	0.278514	0.462746	0.319162	0.578469	0.311567	0.454411	0.689697

Figure 5.1.6.1 Data A dropping attributes

Data B(Binning value)

	Bankrupt?	ROA(B) before interest and depreciation after tax_binned	Operating Gross Margin_binned	Persistent EPS in the Last Four Seasons_binned	Gross Profit to Sales_binned	Cash/Total Assets_binned	Debt ratio %_binned	Net worth/Assets_binned	Liability to Equity_binned	Cash flow rate_binned	Cash Flow Per Share_binned	A
0	1	0	2	0	2	0	9	0	9	1	1	
1	1	1	6	1	6	1	8	1	8	2	2	
2	1	0	2	0	2	0	9	0	9	1	0	
3	1	0	0	0	0	1	7	2	7	5	4	
4	1	2	1	2	1	0	4	5	4	3	3	

Figure 5.1.6.2 Data B dropping attributes (1 of 2)

CFO to Assets_binned	Cash Flow to Equity_binned	Cash Flow to Liability_binned	After-tax Net Profit Growth Rate_binned
0	2	3	1
2	3	3	7
1	4	4	5
6	0	0	1
3	1	1	7

Figure 5.1.6.2 Data B dropping attributes (2 of 2)

5.1.7. Splitting Data into Features and Attributes

Data A

```
1 X_a.head()
```

Figure 5.1..7.1 code to display table data without target variable

	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Persistent EPS in the Last Four Seasons	Gross Profit to Sales	Cash/Total Assets	Debt ratio %	Net worth/Assets	Liability to Equity	Cash flow rate	Cash Flow Per Share	CFO to Assets	Cash Flow to Equity	Cash Flow to Liability	After-tax Net Profit Growth Rate
0	0.442034	0.601457	0.178548	0.601453	0.004094	0.207576	0.792424	0.288207	0.458143	0.311664	0.520382	0.312905	0.458609	0.688979
1	0.516730	0.610235	0.208944	0.610237	0.014948	0.171176	0.828824	0.283846	0.461867	0.318137	0.567101	0.314163	0.459001	0.689693
2	0.472295	0.601450	0.180581	0.601449	0.000991	0.207516	0.792484	0.288207	0.458521	0.307102	0.538491	0.314515	0.459254	0.689463
3	0.457733	0.583541	0.193722	0.583538	0.018851	0.151465	0.848535	0.281721	0.465705	0.321674	0.604105	0.305926	0.448518	0.689110
4	0.522298	0.598783	0.212537	0.598782	0.014161	0.106509	0.893491	0.278514	0.462746	0.319162	0.578469	0.311567	0.454411	0.689697

Figure 5.1.7.2 Data A without target variable

```
1 y_a.head()
```

	Bankrupt?
0	1
1	1
2	1
3	1
4	1

dtype: int64

Figure 5.1.7.3 Data with only target variable

Data B

```
1 X_b.head()
```

Figure 5.1.7.4 code to display table data without target variable

	ROA(B) before interest and depreciation after tax_binned	Operating Gross Margin_binned	Persistent EPS in the Last Four Seasons_binned	Gross Profit to Sales_binned	Cash/Total Assets_binned	Debt ratio %_binned	Net worth/Assets_binned	Liability to Equity_binned	Cash flow rate_binned	Cash Flow Per Share_binned	CFO to Assets_binned	E
0	0	2	0	2	0	9	0	9	1	1	0	
1	1	6	1	6	1	8	1	8	2	2	2	
2	0	2	0	2	0	9	0	9	1	0	1	
3	0	0	0	0	1	7	2	7	5	4	6	
4	2	1	2	1	0	4	5	4	3	3	3	

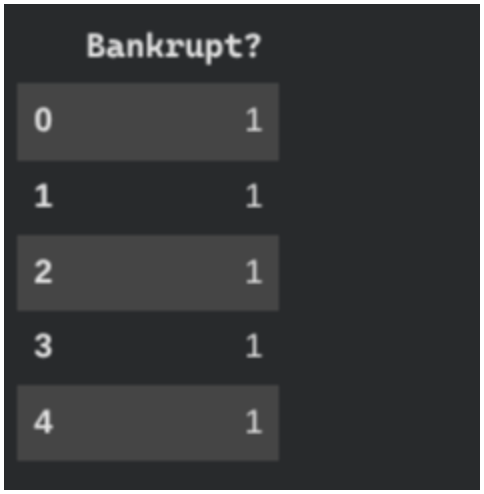
Figure 5.1.7.5 Data B without target variable (1 of 2)

CFO to Assets_binned	Cash Flow to Equity_binned	Cash Flow to Liability_binned	After-tax Net Profit Growth Rate_binned
0	2	3	1
2	3	3	7
1	4	4	5
6	0	0	1
3	1	1	7

Figure 5.1.7.5 Data B without target variable (2 of 2)

```
2 y_b.head()
```

Figure 5.1.7.6 code to display data B with only target variable



	Bankrupt?
0	1
1	1
2	1
3	1
4	1

Figure 5.1.7.3 Data with only target variable

5.1.8. Training Data and Testing Data Split

Data will be split into 80% training and 20% testing. The data will also be stratified to ensure that the proportions of classes in training and testing data are the same. This will help improve model generalization and ensure reliable evaluation metrics.

```
from sklearn.model_selection import train_test_split

X_train_a, X_test_a, y_train_a, y_test_a = train_test_split(X_a,y_a,test_size=0.2,random_state=3,shuffle=True, stratify=y_a)
X_train_b, X_test_b, y_train_b, y_test_b = train_test_split(X_b,y_b,test_size=0.2,random_state=3,shuffle=True, stratify=y_b)
```

Figure 5.1.8.1 code to split the data in 80 percentage for training and 20 percentage for testing

5.1.9. Data Balancing

After splitting the data into training and testing, we will perform oversampling on the data to deal

with the imbalance data which is our target.

The Synthetic Minority Over-sampling Technique (SMOTE) was employed to mitigate the inherent class imbalance within the dataset. In the context of bankruptcy prediction, SMOTE offers a robust alternative to traditional resampling methods by addressing the following critical factors:

Mitigation of Majority Class Bias

Financial datasets are characterized by extreme skewness, where bankrupt firms represent a negligible minority. Standard machine learning classifiers often optimize for global accuracy, leading to a predictive bias toward the majority class (non-bankrupt firms). By artificially balancing the class distribution, SMOTE ensures the model captures the subtle financial signals associated with distress rather than merely reflecting the frequency of healthy firms.

Synthetic Interpolation vs. Observation Duplication

Unlike random oversampling, which relies on the replication of existing minority instances, SMOTE generates synthetic observations by interpolating between nearest neighbors in the minority feature space. For continuous accounting variables and financial ratios, this method:

- Preserves Distributional Integrity: It maintains the underlying statistical properties of the minority class.
- Reduces Overfitting: By avoiding exact duplicates, it prevents the model from "memorizing" specific minority instances, thereby enhancing generalizability.
- Increases Feature Space Diversity: It populates previously empty regions of the minority class domain with realistic data points.

Preservation of Data Richness

While undersampling achieves balance by removing majority class observations, it risks the loss of valuable information regarding the financial stability of healthy firms. SMOTE avoids this trade-off by retaining the full dataset of the majority class while simultaneously enriching the minority class

representation. This is particularly vital in longitudinal financial studies where the nuances of non-distressed firms are essential for establishing a reliable baseline.

Optimization for Cost-Sensitive Recall

In credit risk and bankruptcy forecasting, the "Type II error" (failing to identify a bankrupt firm) is significantly more costly than a "Type I error" (false alarm). SMOTE is specifically designed to improve recall for the minority class. By shifting the decision boundary, the technique optimizes the F1-score and the Area Under the Precision-Recall Curve (PR-AUC), aligning the model with the objective of early risk detection and loss mitigation.

Algorithmic Versatility and Validation

As a model-agnostic preprocessing step, SMOTE facilitates the training of diverse architectures—ranging from traditional Logistic Regression to advanced ensemble methods like Random Forests and Support Vector Machines (SVM). Its widespread adoption in financial econometrics and its status as a benchmark technique for imbalanced classification further validate its application in this study.

```python
from imblearn.over_sampling import SMOTE
from collections import Counter

print("\n--- Solution 4: SMOTE (Synthetic Minority Over-sampling Technique) ---")
smote_a = SMOTE(
    sampling_strategy='auto',  # Balance all classes automatically
    random_state=3,
    k_neighbors=5              # Default = 5 nearest neighbors
)
smote_b = SMOTE(
    sampling_strategy='auto',  # Balance all classes automatically
    random_state=3,
    k_neighbors=5              # Default = 5 nearest neighbors
)
X_smote_a, y_smote_a = smote_a.fit_resample(X_train_a, y_train_a)
X_smote_b, y_smote_b = smote_b.fit_resample(X_train_b, y_train_b)

print("Resampled dataset shape a(SMOTE):", X_smote_a.shape, y_smote_a.shape)
print("Class distribution after a SMOTE:", Counter(y_smote_a))

print("Resampled dataset shape b (SMOTE):", X_smote_b.shape, y_smote_b.shape)
print("Class distribution after b SMOTE:", Counter(y_smote_b))
```

```
--- Solution 4: SMOTE (Synthetic Minority Over-sampling Technique) ---
Resampled dataset shape a(SMOTE): (10558, 14) (10558,)
Class distribution after a SMOTE: Counter({0: 5279, 1: 5279})
Resampled dataset shape b (SMOTE): (10558, 14) (10558,)
Class distribution after b SMOTE: Counter({0: 5279, 1: 5279})
```

Figure 5.1.9.1 for data balancing due to uneven distribution

5.1.10. Data Standardization

Standardization involves scaling the data so that it has a mean of 0 and a standard deviation of 1.

Many machine learning algorithms perform better when features are on a similar scale.The datasets

will be scaled using StandardScaler to ensure that all features have a similar scale

```python
from sklearn.preprocessing import StandardScaler

# Initialize a StandardScaler for dataset 'a' (without binned features)
scaler_a = StandardScaler()

# Fit on X_smote_a (SMOTE-resampled training data for 'a') and transform it
X_smote_scaled_a = scaler_a.fit_transform(X_smote_a)

# Transform the test data for 'a' using the *fitted* scaler from training data
X_test_scaled_a = scaler_a.transform(X_test_a)

# Initialize a StandardScaler for dataset 'b' (with binned features)
scaler_b = StandardScaler()

# Fit on X_smote_b (SMOTE-resampled training data for 'b') and transform it
X_smote_scaled_b = scaler_b.fit_transform(X_smote_b)

# Transform the test data for 'b' using the *fitted* scaler from training data
X_test_scaled_b = scaler_b.transform(X_test_b)

print("StandardScaler applied successfully to SMOTE-resampled training data and test data.")

# Display the shapes of the scaled data to confirm
print(f"X_smote_scaled_a shape: {X_smote_scaled_a.shape}")
print(f"X_test_scaled_a shape: {X_test_scaled_a.shape}")
print(f"X_smote_scaled_b shape: {X_smote_scaled_b.shape}")
print(f"X_test_scaled_b shape: {X_test_scaled_b.shape}")
```

Figure 5.1.10.1 Code to Standardize data

```
StandardScaler applied successfully to SMOTE-resampled training data and test data.
X_smote_scaled_a shape: (10558, 14)
X_test_scaled_a shape: (1364, 14)
X_smote_scaled_b shape: (10558, 14)
X_test_scaled_b shape: (1364, 14)
```

Figure 5.1.10.2 Output Code to Standardize data

3. Modeling

Modeling refers to the process of developing computational models that learn patterns from historical financial data in order to predict future outcomes. In this study, modeling is used to identify whether a company is likely to become bankrupt based on its financial ratios and performance indicators.

Four machine learning models are employed to predict the target variable **"Bankrupt?"**, which indicates whether a company is bankrupt (1) or non-bankrupt (0):

- Gaussian Naive Bayes
- K-Nearest Neighbors (KNN)
- Decision Tree
- Random Forest

Each model is trained and evaluated using two datasets (Data A and Data B) to examine how different data representations affect predictive performance.

To ensure optimal performance, **hyperparameter tuning is conducted using GridSearchCV**, allowing each model to be trained with the most suitable parameter configuration. Due to the imbalanced nature of bankruptcy data, multiple evaluation metrics are used rather than relying solely on accuracy.

The models are evaluated using the following performance measures:

- **Accuracy**: Measures the overall proportion of correctly classified companies
- **Recall (Bankrupt class)**: Measures the model's ability to correctly identify bankrupt companies. This metric is particularly important because failing to detect a bankrupt firm (false negative) can lead to significant financial risk.
- **F1-score**: Represents the harmonic mean of precision and recall, providing a balanced evaluation of model performance when class distributions are uneven.

The inclusion of recall and F1-score ensures that the models are assessed based on their effectiveness in identifying bankrupt companies, rather than being biased toward the majority non-bankrupt class.

3.1. Gaussian Naive Bayes

Characteristics

1. **Assumption of Feature Independence**
   Gaussian Naive Bayes assumes that all financial indicators are conditionally independent given the bankruptcy status. In practice, many financial ratios such as profitability, liquidity, and leverage measures are often correlated. Although this assumption is rarely fully satisfied, the model can still provide reasonable performance and serves as a useful benchmark.
2. **Gaussian Distribution Assumption**
   The model assumes that continuous financial features follow a Gaussian (normal) distribution within each class. If financial ratios approximately follow a normal distribution, Gaussian Naive Bayes can perform effectively. However, significant deviations from normality, such as skewed or heavy-tailed distributions commonly found in financial data, may negatively impact performance.

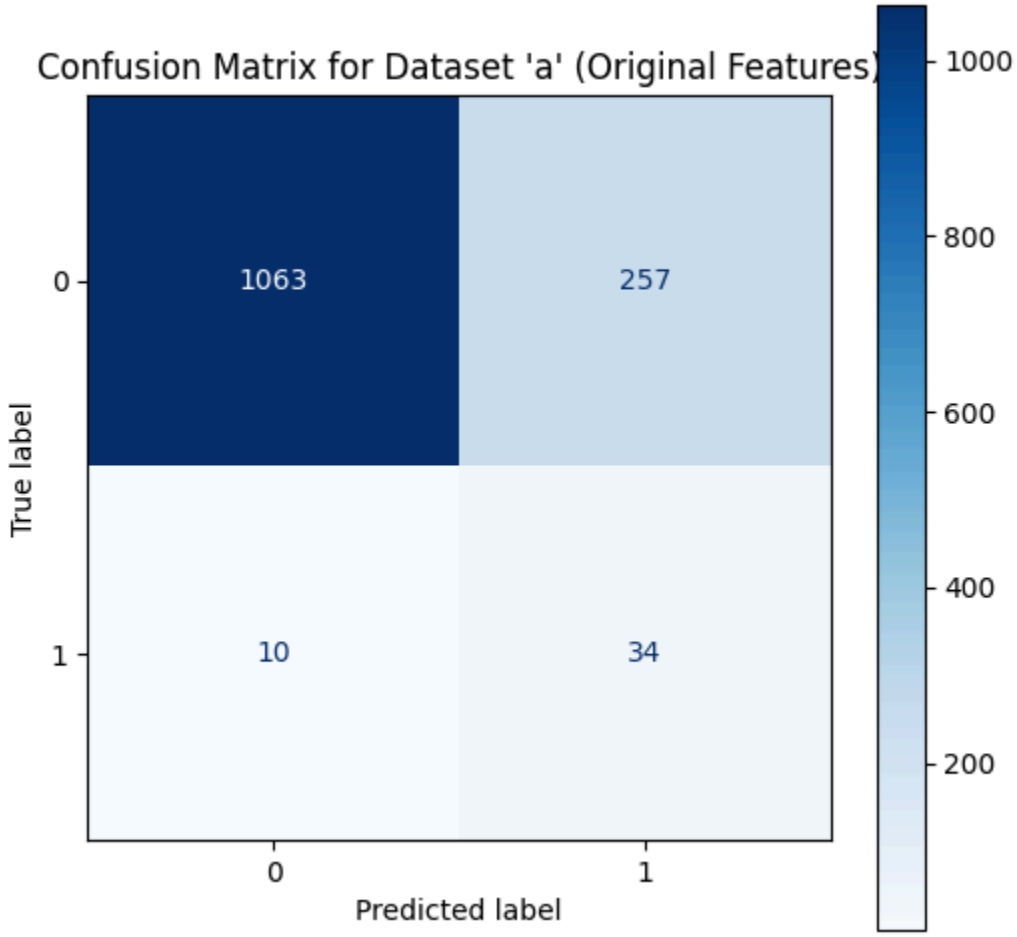
Expected Performance

- **Accuracy**: Expected to be moderate, as the simplicity of the model limits its ability to capture complex relationships among financial variables.
- **Recall (Bankrupt Class)**: The model may struggle to consistently identify bankrupt firms, particularly when class imbalance and feature correlations are present.
- **F1-score**: Since both precision and recall can be affected by the model's assumptions, the F1-score is expected to be moderate, reflecting a trade-off between false positives and false negatives.

Summary

Gaussian Naive Bayes serves as a strong baseline model for bankruptcy prediction due to its computational efficiency and ease of implementation. However, the assumptions of feature independence and normal distribution limit its ability to model the complex interactions inherent in financial data. As a result, more advanced models such as Decision Trees or Random Forests are expected to outperform Gaussian Naive Bayes in terms of recall and F1-score. Despite these limitations, Gaussian Naive Bayes remains valuable for initial analysis and comparative evaluation.

3.1.1. Data A



Effect of Hyperparameter Tuning:

- Gaussian Naive Bayes was fine-tuned using GridSearchCV, with var_smoothing = 0.00534 selected as the optimal parameter.
- The **best cross-validation F1-score of 0.8640** indicates that, during training, the tuned model achieved strong overall balance between precision and recall across folds.
- This suggests that variance smoothing helped stabilize probability estimates and reduced numerical noise in the original financial features.

Performance on Test Data:

- The tuned model achieves an **accuracy of 80.43%**, which is identical to the untuned version, indicating that hyperparameter tuning did not significantly improve overall accuracy on unseen data.
- For non-bankrupt companies (Class 0), the model performs strongly, with **high precision (99%)** and an **F1-score of 0.89**, showing reliable identification of financially healthy firms.

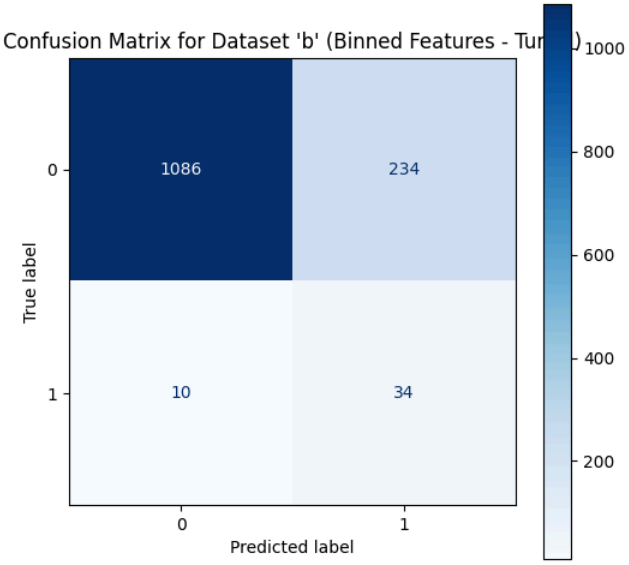
Minority Class (Bankrupt Companies) Performance:

- The model maintains a **high recall of 77% for bankrupt companies (Class 1)**, demonstrating good sensitivity in identifying financially distressed firms.
- However, **precision remains very low at 12%**, leading to a **poor F1-score of 0.20** for the minority class.
- This indicates that while the tuned model detects most bankrupt firms, it also produces a large number of false positive bankruptcy predictions.

Impact of Class Imbalance:

- The discrepancy between the strong cross-validation F1-score and the weak minority-class F1-score on the test set highlights the effect of **severe class imbalance**.
- The high weighted F1-score is dominated by the majority non-bankrupt class, masking the limited reliability of predictions for bankrupt firms.

3.1.2. Data B



Effect of Hyperparameter Tuning:

- Gaussian Naive Bayes was optimized using GridSearchCV, with var_smoothing = 0.0811 identified as the best-performing parameter.
- The **best cross-validation F1-score of 0.8785** indicates strong model stability and balanced learning during training, suggesting that variance smoothing is particularly effective when applied to binned financial features.
- Compared to Dataset ‘a’, the higher cross-validation F1-score implies improved robustness of the model under the binned feature representation.

Performance on Test Data:

- The tuned model achieves an **accuracy of 82.11%**, which is slightly higher than the tuned model on Dataset ‘a’, indicating a modest benefit from feature binning.
- For non-bankrupt companies (Class 0), the model maintains excellent performance, with **99% precision** and an **F1-score of 0.90**, reflecting reliable identification of financially healthy firms.

Minority Class (Bankrupt Companies) Performance:

- The recall for bankrupt companies (Class 1) remains **high at 77%**, confirming the model's strong sensitivity toward detecting financially distressed firms.
- However, **precision remains low at 13%**, resulting in an **F1-score of 0.22** for the minority class.
- This indicates that although most bankrupt firms are detected, the model continues to generate a high number of false positive predictions.

3.2. K-Nearest Neighbors (KNN)

Characteristics

1. **Instance-Based Learning**
   K-Nearest Neighbors is an instance-based (lazy learning) algorithm that does not build an explicit predictive model during training. Instead, it stores all training observations and classifies a company based on the majority class among its nearest financial neighbors, determined using a distance metric in the feature space.

2. **Non-Parametric Nature**
   KNN is a non-parametric model and does not assume any specific underlying data distribution. This characteristic makes it suitable for financial datasets where relationships between ratios and indicators may be complex and non-linear, which is common in bankruptcy prediction tasks.

Expected Performance

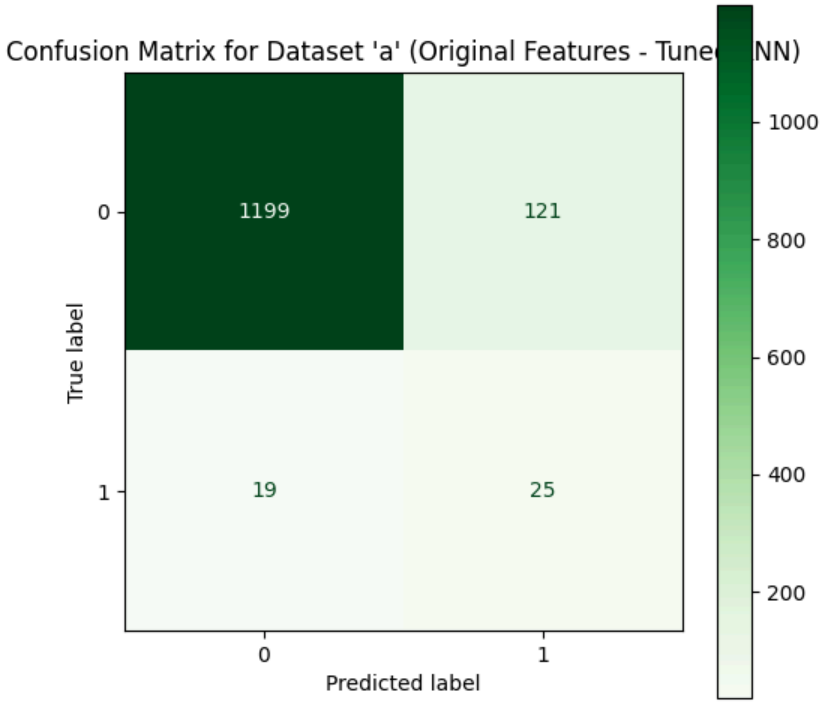
- **Accuracy**: KNN is expected to achieve good accuracy when financial features are properly scaled and the number of neighbors (K) is optimally selected through cross-validation. Its flexibility allows it to capture local patterns that simpler models may overlook.
- **Recall (Bankrupt Class)**: With an appropriate choice of K and distance metric, KNN can achieve relatively strong recall for bankrupt companies. Its sensitivity to local neighborhoods enables it to identify firms with financial characteristics similar to previously bankrupt cases.

- **Precision and F1-score**: Precision and F1-score depend heavily on the selected value of K. A small K may lead to overfitting and unstable predictions, while a large K may oversmooth class boundaries and reduce sensitivity to the minority class. Cross-validation helps balance this trade-off to achieve more stable F1-score performance.

Summary

K-Nearest Neighbors is a flexible and intuitive model for bankruptcy prediction, particularly when financial features are carefully preprocessed and scaled. Compared to simpler probabilistic models such as Gaussian Naive Bayes, KNN has the potential to capture more complex interactions among financial indicators. However, its performance is highly dependent on preprocessing steps, the choice of distance metric, and the optimal selection of K. Additionally, KNN can be computationally expensive for large datasets, which should be considered when scaling the model.

3.2.1. Data A



Effect of Hyperparameter Tuning:

- KNN was optimized using GridSearchCV, with **3 neighbors**, **Euclidean distance**, and **distance-based weighting** selected as the best parameter combination.
- The **high cross-validation F1-score of 0.9565** indicates that the tuned KNN model learned strong local patterns during training and achieved well-balanced performance across folds.
- Using distance-based weights allows closer neighbors to have greater influence, improving sensitivity to financially distressed firms.

Performance on Test Data:

- The tuned KNN model achieves an **accuracy of 89.74%**, which is noticeably higher than that of Gaussian Naive Bayes on the same dataset.
- For non-bankrupt companies (Class 0), the model performs very well, with **98% precision**, **91% recall**, and an **F1-score of 0.94**, indicating strong reliability in identifying financially stable firms.

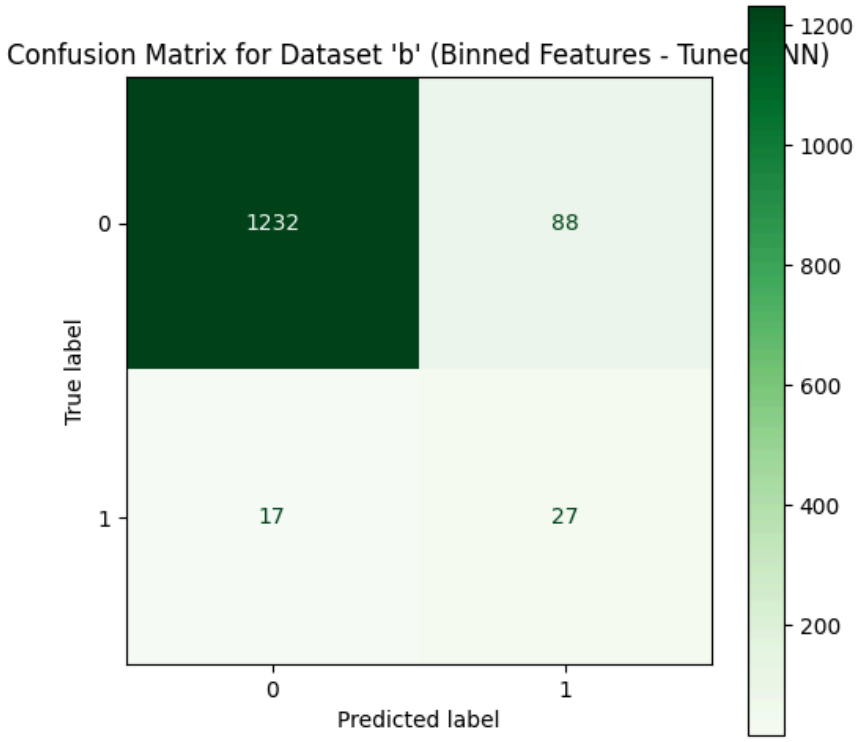
Minority Class (Bankrupt Companies) Performance:

- The model achieves a **recall of 57% for bankrupt firms (Class 1)**, showing a moderate ability to detect companies that eventually become bankrupt.
- However, **precision remains low at 17%**, resulting in an **F1-score of 0.26** for the minority class.
- This suggests that while KNN identifies over half of the bankrupt firms, it still produces a considerable number of false positive bankruptcy predictions.

Impact of Class Imbalance:

- The strong weighted F1-score is largely driven by the majority non-bankrupt class, masking the weaker performance for bankrupt firms.
- The gap between the very high cross-validation F1-score and the lower minority-class F1-score on the test set suggests possible sensitivity to class imbalance and local noise.

3.2.2. Data B



Effect of Hyperparameter Tuning:

- For both datasets, GridSearchCV selected the same optimal configuration: Euclidean distance, 3 nearest neighbors, and distance-based weighting.
- The high cross-validation F1-scores (0.9565 for Dataset ‘a’ and 0.9592 for Dataset ‘b’) indicate that the tuned KNN model performs very well during training, suggesting strong capability in capturing local patterns within the financial data.

Performance on Dataset ‘a’ (Original Features)

Overall Performance:

- The model achieves an accuracy of 89.74%, indicating strong overall classification performance.
- For non-bankrupt companies (Class 0), the model performs well with a high F1-score of 0.94, showing reliable identification of financially healthy firms.

Minority Class (Bankrupt Companies) Performance:

- The recall for bankrupt firms is 57%, meaning the model correctly identifies more than half of the bankrupt companies.
- However, precision remains low at 17%, resulting in a modest F1-score of 0.26 for Class 1.
- This indicates that while KNN is able to detect some bankrupt firms, it still produces a considerable number of false positive bankruptcy predictions.

Summary for Dataset ‘a’:

- KNN with original features shows a notable improvement over Gaussian Naive Bayes in terms of accuracy and minority-class F1-score.
- Nevertheless, the imbalance between precision and recall suggests that predictions for bankrupt companies remain unreliable without additional imbalance-handling techniques.

Performance on Dataset ‘b’ (Binned Features)

Overall Performance:

- The accuracy improves further to 92.30%, indicating that feature binning enhances KNN’s ability to separate classes.
- The non-bankrupt class continues to show excellent performance, with an F1-score of 0.96.

Minority Class (Bankrupt Companies) Performance:

- Recall for bankrupt firms increases to 61%, showing improved sensitivity compared to Dataset ‘a’.

- Precision also improves to 23%, leading to a higher F1-score of 0.34 for Class 1.

- This improvement suggests that binned features help KNN form clearer neighborhood boundaries for financially distressed firms.

Summary for Dataset ‘b’:

- KNN with binned features outperforms the original feature set across all key metrics, particularly in terms of accuracy, recall, and F1-score for the bankrupt class.
- Despite these improvements, precision for bankrupt firms remains relatively low, highlighting the ongoing impact of class imbalance.

3.3. K-Nearest Neighbors (KNN)

Characteristics

1. **Tree-Based Structure**

   A Decision Tree model works by recursively splitting the dataset into smaller subsets based on the values of financial features. At each split, the algorithm selects the feature and threshold that maximize information gain or minimize Gini impurity.Each internal node represents a decision rule based on a financial indicator, while each leaf node corresponds to a predicted bankruptcy outcome (bankrupt or non-bankrupt).

2. **Non-Parametric Model**

   Decision Trees are non-parametric and do not assume any specific distribution of the data. This makes them well-suited for bankruptcy prediction, where relationships between financial ratios can be non-linear and complex. The model is capable of capturing interactions between multiple financial indicators without requiring prior assumptions about their distributions.

Expected Performance

- **Accuracy**: Decision Trees can achieve good accuracy in bankruptcy prediction, particularly when hyperparameters such as tree depth and minimum samples per leaf are properly tuned. However, overly deep trees may overfit the training data and lead to reduced performance on unseen companies.

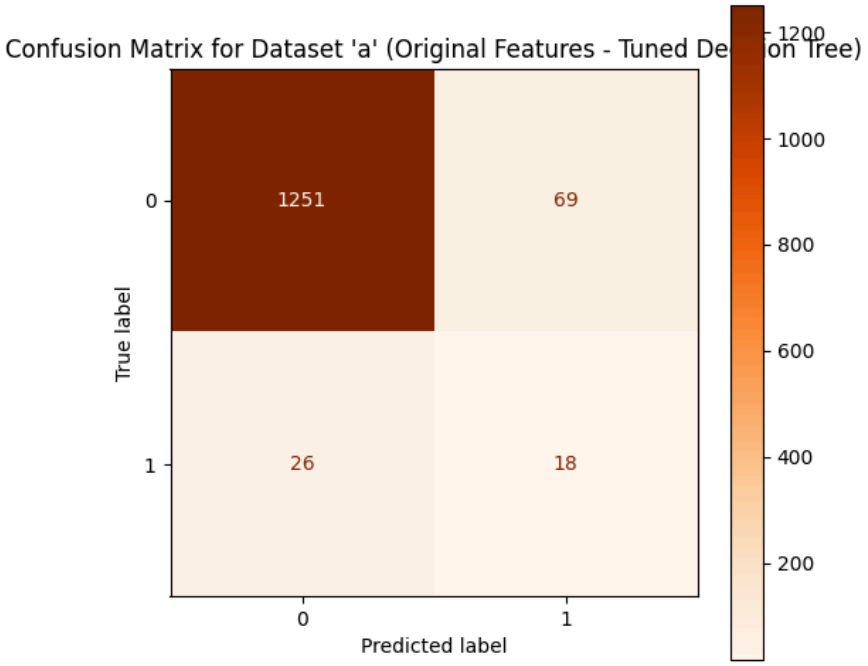
- **Precision and Recall**: Decision Trees are capable of achieving reasonable precision and recall for both bankrupt and non-bankrupt firms. Nevertheless, without proper regularization, the model may overfit, resulting in unstable precision and recall when applied to test data.

- **F1-score**: The F1-score is expected to be moderate, reflecting a balance between precision and recall. If overfitting occurs, the model may show strong performance on training data but weaker generalization on the test set, particularly for the minority bankrupt class.

Summary

Decision Trees are a useful and interpretable model for bankruptcy prediction, as they can capture both simple decision rules and complex interactions among financial features. Their ability to handle non-linear relationships makes them suitable for financial datasets with diverse indicators. However, Decision Trees are prone to overfitting, which necessitates careful hyperparameter tuning to ensure robust performance on unseen data. While they provide valuable insights into decision-making patterns, their predictive stability may be lower compared to ensemble methods such as Random Forest.

3.3.1. Data A



Moderate Recall for Class 1 (Bankrupt Companies):

- The recall for bankrupt firms (Class 1) is **41%**, indicating that the Decision Tree is able to identify some financially distressed companies but still misses a considerable portion of actual bankrupt cases.

- This level of recall suggests a **moderate ability to detect bankruptcy**, with a notable number of false negatives, which is a critical concern in financial risk assessment.

Imbalanced Class Performance:

- The model performs very well for non-bankrupt companies (Class 0), achieving a **high F1-score of 0.96**, reflecting strong precision and recall for financially healthy firms.

- In contrast, the **F1-score for bankrupt companies is relatively low at 0.27**, indicating that the model struggles to balance precision and recall for the minority class.

- This imbalance highlights the impact of class skew, where the model favors the majority non-bankrupt class.

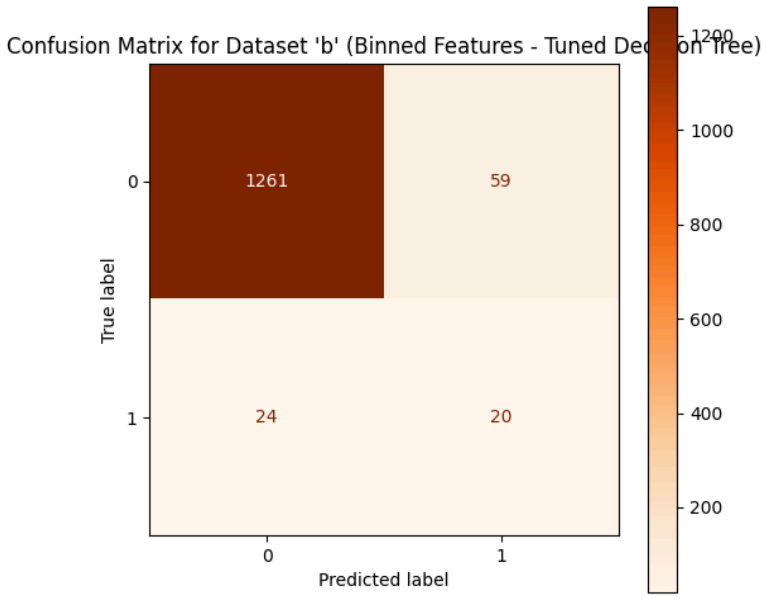
Overfitting Consideration:

- The Decision Tree achieves a strong **cross-validation F1-score of 0.9493**, while the test accuracy is **93.04%**, suggesting good overall learning capacity.

- However, the weaker minority-class performance on the test set indicates that the model may be **overfitting to dominant patterns in the training data**, particularly those associated with non-bankrupt firms.

- Further regularization, such as limiting tree depth or increasing the minimum number of samples per leaf, could help improve generalization for bankruptcy company detection.

3.3.2. Data B



Moderate Recall for Class 1 (Bankrupt Companies):

- The recall for bankrupt firms (Class 1) is **45%**, indicating that the Decision Tree is able to identify nearly half of the companies that eventually become bankrupt.

- While this shows an improvement compared to simpler models, a considerable number of bankrupt firms are still misclassified as non-bankrupt, leading to false negatives, which remain a concern in financial risk assessment.

Balanced but Class-Skewed Performance:

- The model performs strongly for non-bankrupt companies (Class 0), achieving an **F1-score of 0.97**, reflecting reliable classification of financially healthy firms.

- For bankrupt companies, the **F1-score is 0.33**, which highlights an imbalance between precision and recall for the minority class.

- The macro-average F1-score of **0.65** suggests that overall performance across both classes is moderate, with better results driven primarily by the majority class.

Generalization and Model Stability:

- The **accuracy of 93.91%** indicates strong overall predictive performance; however, this metric is influenced by the dominance of non-bankrupt firms in the dataset.

- The relatively high cross-validation F1-score (**0.9654**) compared to the test-set minority-class performance suggests that the model may still be sensitive to data imbalance and potential overfitting to training patterns.

3.4. Random Forest

Characteristics

1. **Ensemble Learning**

   Random Forest is an ensemble learning algorithm that constructs multiple decision trees using different subsets of the training data and features. Each tree independently predicts the bankruptcy status of a company, and the final prediction is determined through majority voting. By aggregating multiple trees, Random Forest improves predictive performance and reduces the variance associated with single decision tree models.

2. **Overfitting Prevention**

   Compared to a single decision tree, Random Forest is more resistant to overfitting. Although individual trees may overfit to specific patterns in the financial data, combining their predictions results in a more stable and generalized model. This is particularly beneficial in bankruptcy prediction, where financial indicators may contain noise and complex interactions.

Expected Performance

- Accuracy: Random Forest is expected to achieve high accuracy in predicting bankruptcy status. Its ability to model complex, non-linear relationships among financial ratios allows it to outperform simpler models, especially when class imbalance is properly addressed.

- Precision and Recall: The model is generally effective at balancing precision and recall. With appropriate hyperparameter tuning, Random Forest can reliably identify bankrupt firms (high recall) while reducing false bankruptcy predictions (maintaining reasonable precision).

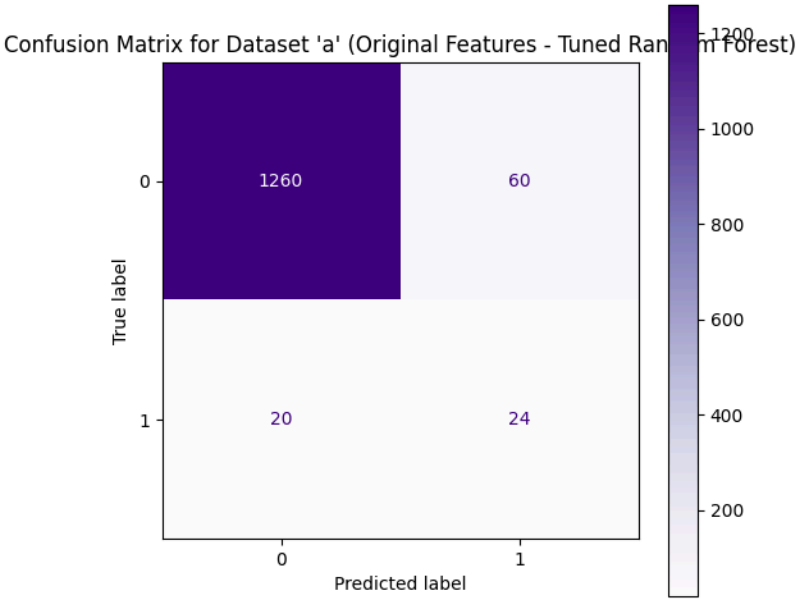
- F1-score: Due to its balanced treatment of precision and recall, Random Forest is expected to achieve a strong F1-score, making it suitable for imbalanced classification problems such as bankruptcy prediction.

Summary

Random Forest is a robust and reliable model for bankruptcy prediction. It delivers high predictive accuracy, mitigates overfitting, and is capable of capturing both linear and non-linear relationships among

financial indicators. Additionally, Random Forest provides feature importance measures, offering valuable insights into the factors contributing to financial distress. With proper hyperparameter tuning and imbalance-handling techniques, Random Forest is well-suited for practical bankruptcy risk assessment and comparative model evaluation.

3.4.1. Data A



Expected Performance

- Accuracy: Random Forest is expected to achieve high accuracy in bankruptcy prediction due to its ability to model complex, non-linear relationships among financial ratios. It generally outperforms simpler models by capturing interactions between profitability, liquidity, and leverage indicators.

- Recall (Bankrupt Class): With appropriate hyperparameter tuning, Random Forest can achieve improved recall for bankrupt companies by learning diverse decision boundaries. This is crucial in bankruptcy prediction, where missing a financially distressed firm carries significant risk.

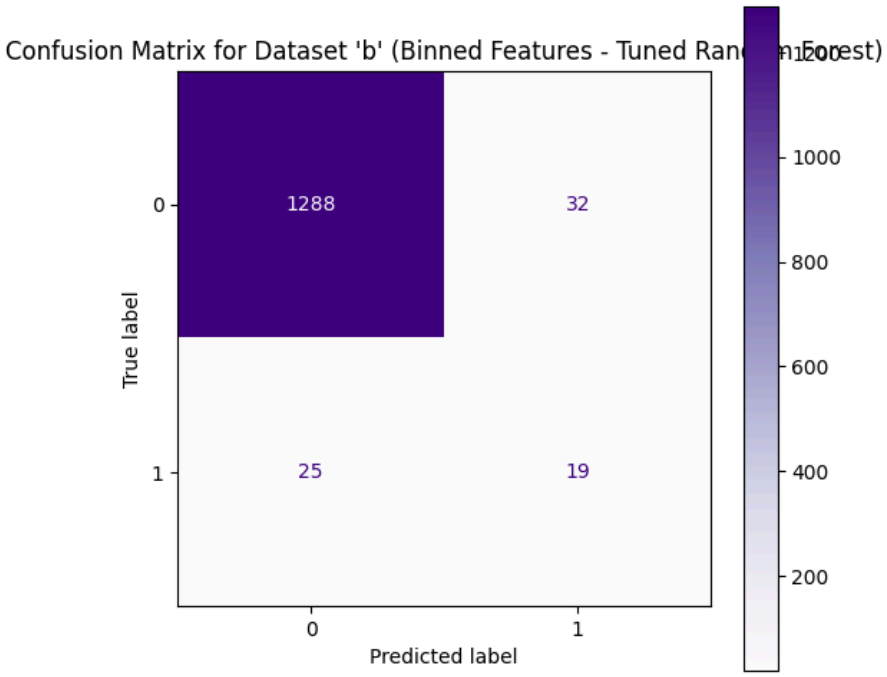
- Precision and F1-score: Random Forest typically provides a better balance between precision and recall compared to simpler models. As a result, the F1-score is expected to be higher, indicating more reliable performance in identifying bankrupt firms while controlling false positives.

Summary

Random Forest is a strong and reliable model for bankruptcy prediction. It offers high predictive accuracy, effectively reduces overfitting, and is capable of capturing both linear and non-linear

relationships in financial data. Additionally, Random Forest provides feature importance measures, which can offer valuable insights into the key financial indicators associated with corporate bankruptcy. With proper hyperparameter tuning, Random Forest is well-suited for handling complex financial datasets and serves as a powerful benchmark for comparison with other models.

3.4.2. Data B



Expected and Observed Performance

- Accuracy:
  Random Forest achieves high accuracy on both datasets, with 94.13% on Dataset ‘a’ and 95.82% on Dataset ‘b’. The higher accuracy on the binned dataset suggests that feature binning improves the model’s ability to separate bankrupt and non-bankrupt firms.

- Recall (Bankrupt Class):
  The model demonstrates moderate recall for bankrupt companies, with 55% on Dataset ‘a’ and 43% on Dataset ‘b’. While Random Forest is effective at identifying most non-bankrupt firms, detecting the minority bankrupt class remains challenging due to class imbalance.

- Precision and F1-score (Bankrupt Class):
  Precision for bankrupt firms improves from 29% (Dataset ‘a’) to 37% (Dataset ‘b’), resulting in an increase in F1-score from 0.38 to 0.40. This indicates that the binned feature representation helps reduce false positive bankruptcy predictions and improves classification balance.

- Overall Model Balance:
  The strong weighted F1-scores (0.95–0.96) show excellent overall performance, while the macro-average scores highlight the remaining difficulty in predicting the minority class fairly.

Summary

Random Forest is one of the strongest-performing models in this bankruptcy prediction study. It consistently achieves high accuracy and stable performance across both original and binned feature sets. Feature binning further enhances the model's precision and overall robustness.

However, despite its strengths, Random Forest still exhibits moderate recall and F1-score for bankrupt firms, reflecting the inherent difficulty of minority-class prediction in highly imbalanced financial datasets. These results suggest that combining Random Forest with imbalance-handling techniques such as SMOTE may further improve bankruptcy detection performance.

4. Model Evaluation

Model evaluation is the process of assessing how well a machine learning model performs on a given dataset. It involves using various metrics and techniques to measure the model's performance, robustness, and generalization ability, typically using unseen data (like a test set or validation set) that the model has not been trained on. Model evaluation helps to determine if the model meets the desired accuracy and whether it can generalize well to new, unseen data.

Comparison Table

--- Model Comparison Table ---

	Model	Dataset	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)
0	Gaussian Naive Bayes	Original Features (a)	0.804300	0.116800	0.772700	0.203000
1	Gaussian Naive Bayes	Binned Features (b)	0.821100	0.126900	0.772700	0.217900
2	K-Nearest Neighbors	Original Features (a)	0.897400	0.171200	0.568200	0.263200
3	K-Nearest Neighbors	Binned Features (b)	0.923000	0.234800	0.613600	0.339600
4	Random Forest	Original Features (a)	0.941300	0.285700	0.545500	0.375000
5	Random Forest	Binned Features (b)	0.958200	0.372500	0.431800	0.400000
6	Decision Tree	Original Features (a)	0.930400	0.206900	0.409100	0.274800
7	Decision Tree	Binned Features (b)	0.939100	0.253200	0.454500	0.325200

Note: Precision, Recall, and F1-Score are reported for the minority class (Bankrupt? = 1).

Figure 5.1 show the comparison between all model

1. Model names
    - gnb_A, gnb_B: Gaussian Naive Bayes trained on original and binned features respectively.
    - knn_A, knn_B: K-Nearest Neighbors with original and binned features.
    - dt_A, dt_B: Decision Tree models using original and binned features.
    - rf_A, rf_B: Random Forest with original and binned feature sets.
2. Metrics
    - Accuracy: Overall proportion of correctly classified companies on the test set.

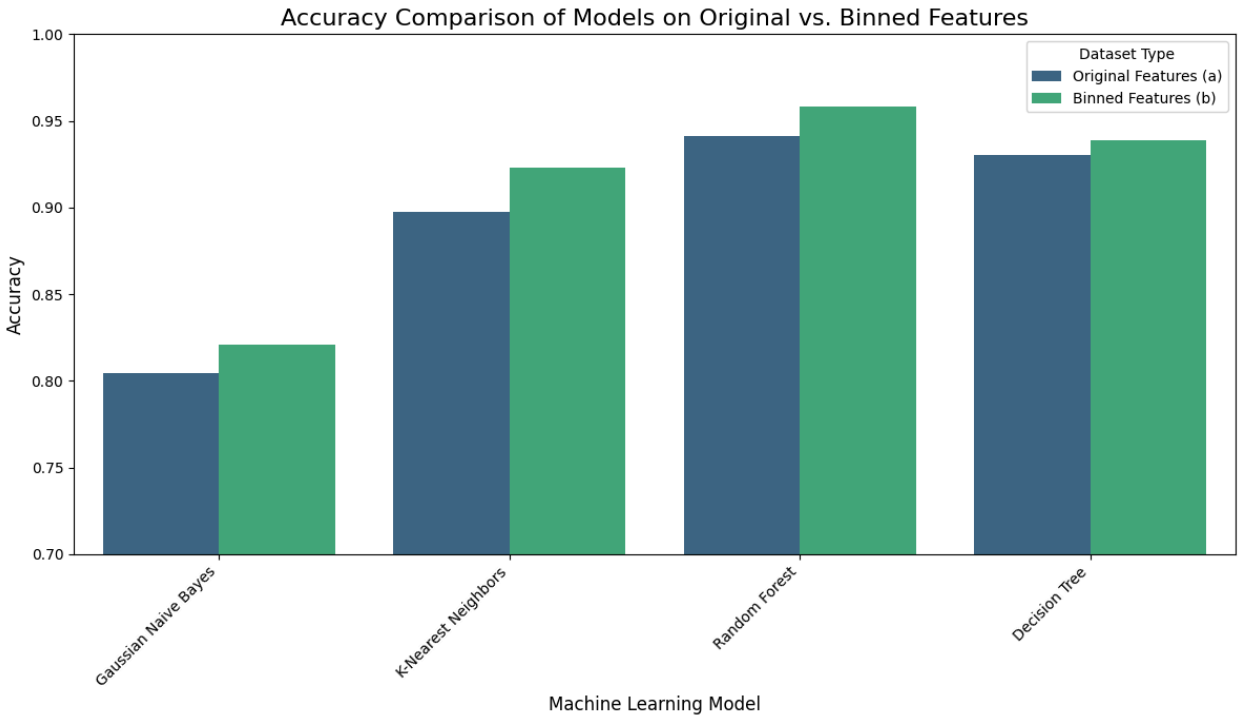
- Precision (Class 1): Among companies predicted as bankrupt, the proportion that are truly bankrupt.
- Recall (Class 1): Among truly bankrupt companies, the proportion correctly identified as bankrupt.
- F1-Score (Class 1): Harmonic mean of precision and recall for the bankrupt class, balancing both metrics.

3. Color coding
    - Green cells highlight the highest value within each metric column, indicating the strongest performance for that metric.
    - Red cells indicate the lowest value for a given metric, marking comparatively weaker performance.

4.1. Accuracy

Accuracy is the percentage of correct predictions out of all predictions.

Accuracy = (*True Positive* (*TP*) + *True Negative* (*TN*))/*Total Instances*



This image shows a bar chart comparing the test accuracy of four machine learning models trained on original features (a) versus binned features (b). Each pair of bars represents how binning affects model performance.

1. Models displayed
    - Gaussian Naive Bayes: Accuracy increases slightly when using binned features compared to original features.
    - K-Nearest Neighbors: Binned features yield a noticeable accuracy improvement over original features.
    - Random Forest: Achieves the highest accuracy among all models, with binned features performing better than original features.
    - Decision Tree: Also shows a small accuracy gain when using binned features.
2. Axes and legend

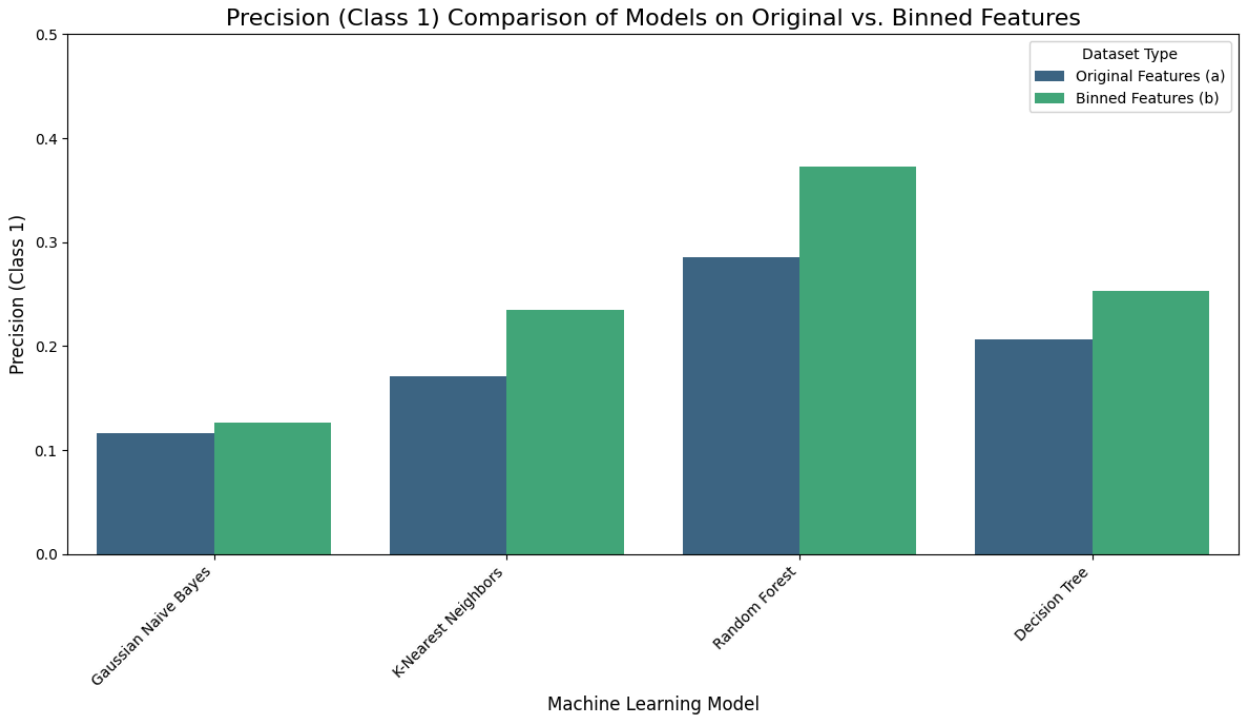
- The x-axis lists the machine learning models, while the y-axis shows accuracy values between 0.70 and 1.00.
- Blue bars correspond to models trained on original features (a), and green bars correspond to models trained on binned features (b), as indicated in the legend.

3. Overall interpretation
    - For every model, using binned features leads to equal or higher accuracy than using original features.
    - Random Forest with binned features attains the best overall accuracy, suggesting that feature binning can enhance predictive performance for this bankruptcy classification task.

4.2. Precision

Precision is the ratio of true positives to the total number of predicted positives. It answers: "Of all the instances predicted as positive, how many are actually positive?"

Precision = *True Positive* (*TP*) / (*True Positive* (*TP*) + *False Positive* (*FP*))



This image shows a bar chart comparing the precision for the minority class (Class 1: bankrupt) across four machine learning models, using original features (a) and binned features (b). Precision here measures how many of the companies predicted as bankrupt are actually bankrupt.

Models displayed

Gaussian Naive Bayes: Slight increase in precision when using binned features compared to original features.

K-Nearest Neighbors: Precision improves noticeably with binned features.

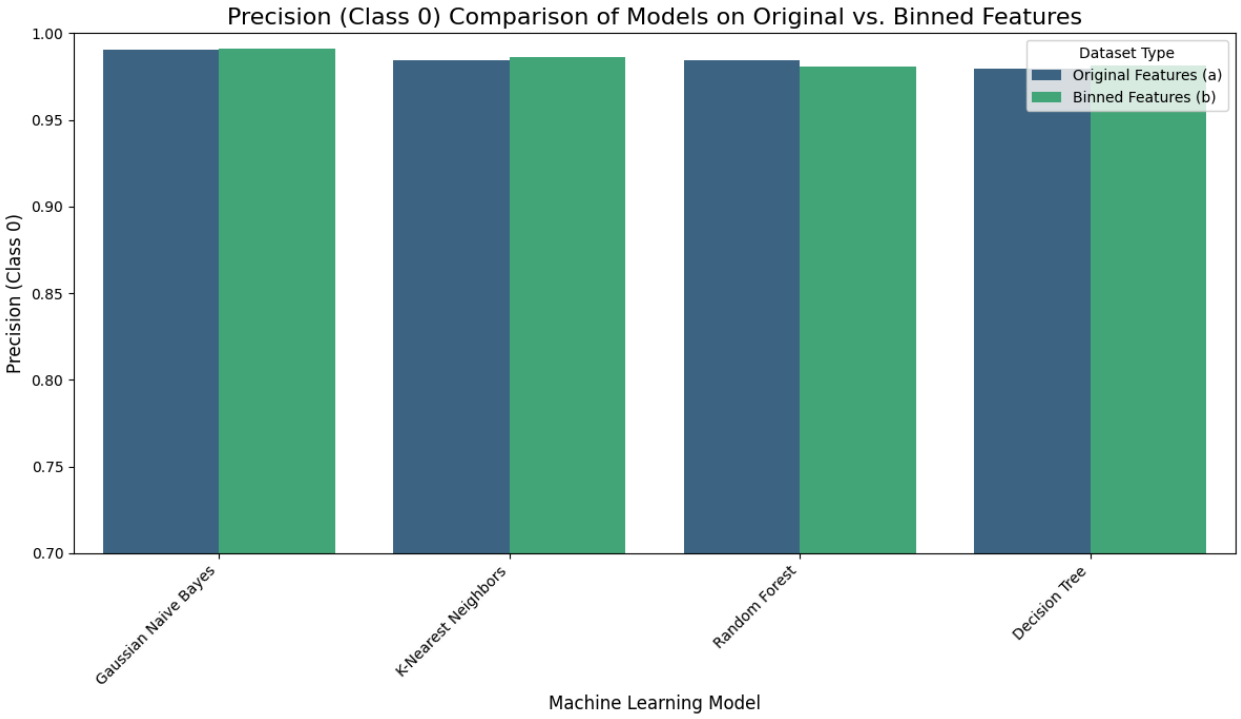
Random Forest: Achieves the highest precision overall, with a substantial gain from binning.

Decision Tree: Also shows higher precision when trained on binned features.

Axes and legend

The x-axis lists the machine learning models, while the y-axis shows precision values from 0.0 to 0.5.

Blue bars represent models trained on original features (a), and green bars represent models trained on binned features (b), as indicated by the legend.



Overall interpretation

For all four models, binning the features leads to higher precision for the bankrupt class.

Random Forest with binned features offers the best precision, indicating it makes the most reliable positive (bankrupt) predictions among the compared models.

This image presents a bar chart comparing the precision for the majority class (Class 0: non-bankrupt) across four machine learning models using original features (a) and binned features (b). Precision here indicates how many companies predicted as non-bankrupt are actually non-bankrupt.

Models displayed

Gaussian Naive Bayes: Shows very high precision for Class 0 on both original and binned features, with almost no visible difference.

K-Nearest Neighbors: Maintains similarly high precision for Class 0, with a slight increase when using binned features.

Random Forest: Precision for Class 0 remains high, with a marginal decrease for binned features compared to original.

Decision Tree: Also exhibits consistently high precision, with almost overlapping bars for original and binned features.

Axes and legend

The x-axis lists the machine learning models, while the y-axis shows precision values, ranging from 0.70 to 1.00 but concentrated near the upper end.

Blue bars represent models trained on original features (a), and green bars represent models trained on binned features (b), as indicated in the legend.

Overall interpretation

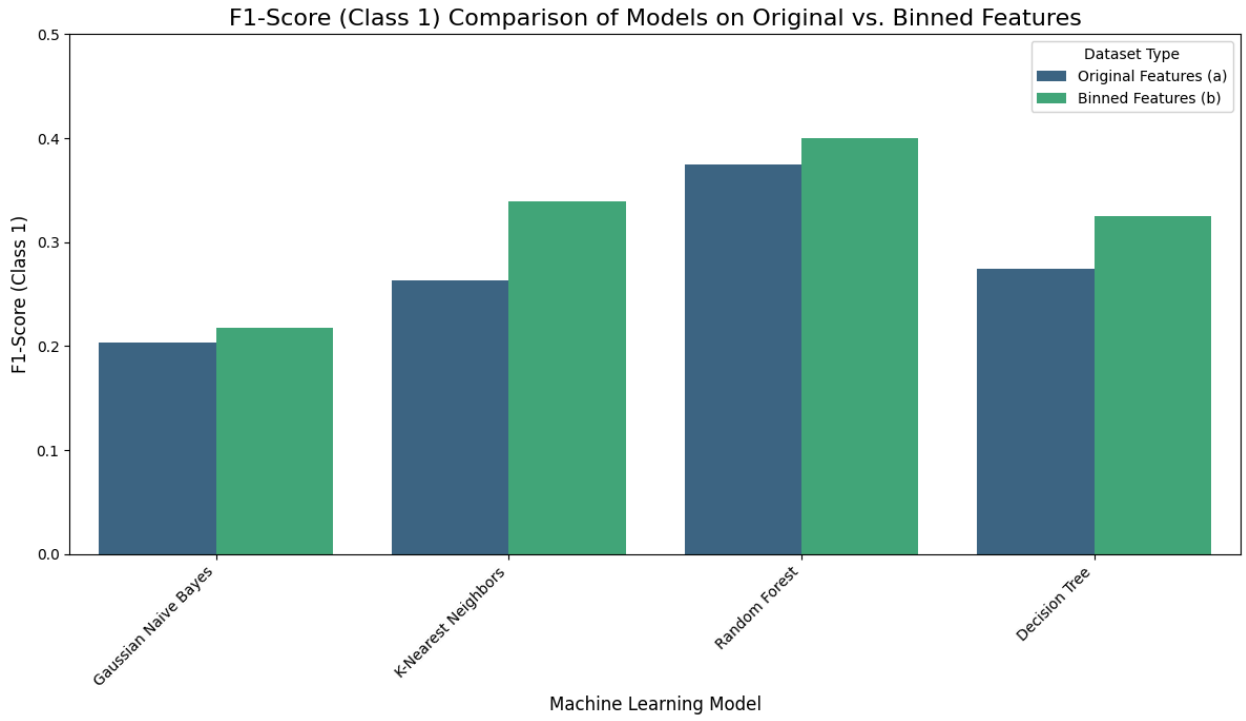
All models achieve very high precision for the non-bankrupt class regardless of whether original or binned features are used.

Feature binning produces only minor changes in Class 0 precision, suggesting that it mainly affects performance on the minority (bankrupt) class rather than the majority class.

4.3. F1-Score

F1-Score is the harmonic mean of precision and recall. It's useful when we want to balance precision and recall.

$F1 = 2 \times Precision \times Recall / (Precision + Recall)$



This image presents a bar chart comparing the F1-Score for the minority class (Class 1: bankrupt) across four machine learning models, using both original features (a) and binned features (b). The F1-Score combines precision and recall to reflect the overall effectiveness of each model at detecting bankrupt companies.

Models displayed

Gaussian Naive Bayes: Shows a small improvement in Class 1 F1-Score when using binned features.

K-Nearest Neighbors: Exhibits a noticeable increase in F1-Score with binned features compared to original features.

Random Forest: Achieves the highest F1‑Scores overall, with binned features giving a further boost over original features.

Decision Tree: Also benefits from binning, with a higher F1‑Score on binned features.

Axes and legend

The x‑axis lists the machine learning models, while the y‑axis shows F1‑Score values from 0.0 to 0.5.

Blue bars represent models trained on original features (a), and green bars represent models trained on binned features (b), as indicated by the legend.

Overall interpretation

For all four models, binning features leads to equal or higher F1‑Scores for the bankrupt class.

Random Forest with binned features provides the strongest balance of precision and recall for detecting bankrupt firms, making it the most effective model among those compared in this chart.

Best model

After comparing all models, the Random Forest classifier trained on binned features (b) is selected as the best-performing model for the bankruptcy prediction task. It achieves the highest overall accuracy, strong precision, and the best F1‑Score for the minority class (Bankrupt = 1) among the evaluated models, making it suitable when correctly identifying bankrupt firms is a priority.

Choosing the Random Forest with binned features

The Random Forest model on binned features consistently outperforms Gaussian Naive Bayes, K‑Nearest Neighbors, and Decision Tree on key minority‑class metrics. Its F1‑Score for bankrupt companies is the highest, indicating a good balance between precision and recall for this critical class, while its accuracy is also the best among all configurations.

Limitation: class‑imbalance and minority errors

Despite being the best among the tested models, the Random Forest with binned features still has limitations for the minority (bankrupt) class. Precision and recall for bankrupt firms remain considerably lower than for non‑bankrupt firms, reflecting the strong class imbalance in the dataset and leaving room for both false negatives and false positives.

SMOTE and further improvements

To address class imbalance, Synthetic Minority Oversampling Technique (SMOTE) was applied to generate synthetic bankrupt samples instead of simple random resampling, helping the model learn a richer decision boundary for the minority class. Beyond SMOTE, further gains may come from tuning Random Forest hyperparameters, adjusting decision thresholds, and exploring cost‑sensitive learning or additional ensemble and boosting methods.

Summary

The Random Forest classifier trained on binned features (b), enhanced with SMOTE for handling the minority bankrupt class, is chosen as the best model because it offers the strongest combination of accuracy and minority-class F1-Score among the evaluated models. Although challenges from class imbalance remain, this setup provides a solid baseline for bankruptcy prediction and a foundation for future refinement.

5. Deployment

Model deployment is the process of taking a trained machine learning model and making it available for use in a production environment, where it can provide predictions or insights based on real-world data. Deployment allows the model to be integrated into software systems, applications, or business workflows so that end users or automated systems can leverage the model's predictions or decisions.

Save Scaler and Model

We save the scaler used to scale the data and the model into a ".joblib" file which is the preferred way to save sklearn models.

```python
import joblib
import os

# Define the filename for the best model to be saved locally in Colab
model_filename = 'best_random_forest_binned_model.joblib'

# Save the best Random Forest model trained on binned features
joblib.dump(best_rf_b, model_filename)

print(f"Best model (Random Forest with Binned Features) exported successfully as '{model_filename}'")
print(f"It is saved in the current Colab working directory: {os.getcwd()}")


# Save the StandardScaler for binned features locally in Colab
scaler_b_filename = 'scaler_b.joblib'
joblib.dump(scaler_b, scaler_b_filename)
print(f"Scaler for binned features exported successfully as '{scaler_b_filename}'")
```

```
est model (Random Forest with Binned Features) exported successfully as 'best_random_forest_binned_model.joblib'
t is saved in the current Colab working directory: /content
caler for original features exported successfully as 'scaler_a.joblib'
caler for binned features exported successfully as 'scaler_b.joblib'
```

Figure 6.1 export model and scaler

5.1. Deployment Interface

We decided to use a web application to deploy our model. It is created using the python library

Gradio as it is simple to use and set up.

Load Scaler and Model

```
12      loaded_scaler_b = joblib.load(scaler_b_path)
13      loaded_model = joblib.load(model_path)
```

Figure 6.1.1 Load the mode in UI.py

Data Input

```
with gr.Blocks() as demo:
    gr.Markdown("Company Bankrupt prediction")
    with gr.Row():
        roa_b = gr.Number(value=0.01, label="ROA (B) before interest and depreciation after tax", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Operating_Gross_Margin = gr.Number(value=0.01, label="Operating Gross Margin", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Persistent_EPS = gr.Number(value=0.01, label="Persistent EPS in the Last Four Seasons", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Gross_Profit_to_Sales = gr.Number(value=0.01, label="Gross Profit to Sales", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Cash_Total_Assets = gr.Number(value=0.001, label="Cash / Total Assets", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Debt_Ratio = gr.Number(value=0.01, label="Debt Ratio %" ,minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Net_Worth_Assets = gr.Number(value=0.001, label="Net Worth / Assets" ,minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Liability_to_Equity = gr.Number(value=0.001, label="Liability to Equity", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Cash_Flow_Rate = gr.Number(value=0.001, label="Cash Flow Rate", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Cash_Flow_Per_Share = gr.Number(value=0.001, label="Cash Flow Per Share", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        CFO_to_Assets = gr.Number(value=0.001, label="CFO to Assets", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Cash_Flow_To_Equity = gr.Number(value=0.001, label="Cash Flow to Equity", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        Cash_Flow_To_Liabilities = gr.Number(value=0.001, label="Cash Flow to Liabilities", minimum=0.0, maximum=1.0, min_width=100)
    with gr.Row():
        After_tax_Net_Profit_Growth_Rate = gr.Number(value=0.001, label="After-tax Net Profit Growth Rate", minimum=0.0, maximum=1.0, min_width=100)
```

Figure 6.1.2 fields to enter in UI.py

Data Binning

Set the value for each feature binning value

```
53      # binning each value to specific bin
54      roa_bins = [-float('inf'),
55                  0.491782215322019, 0.520370469511216,
56                  0.533004978853258, 0.543069757481664,
57                  0.552277959205525, 0.5628138551314312,
58                  0.575940896193586, 0.5937148669629,
59                  0.620589967343005,
60                  float('inf')]
61      ogm_bins = [
62          -float('inf'),
63          0.596513354184984,
64          0.599208694273483,
65          0.6014961299528674,
66          0.6036696983237,
67          0.605997492036495,
68          0.608678418541634,
69          0.6119185920811776,
70          0.6161893368310298,
71          0.6231525389527092,
72          +float('inf')
73      ]
74      p_eps_bins = [
75          -float('inf'),
76          0.201096719296587, 0.211213009359932,
77          0.216507516308972, 0.22030821594024744,
78          0.224543821499948, 0.228987425545996,
79          0.234660111562825, 0.2436418644228O4,
80          0.25878793608773737,
81          +float('inf')
82      ]
```

Figure 6.1.3 per-set the binning value to specific bin in UI.py

Bin the data value to their specific bin

```
231     # binning the data
        Windsurf: Refactor | Explain | Generate Docstring | X
232     def bin_data(data, bins):
233         for j in range(len(bins)-1):
234             if data >= bins[j] and data < bins[j+1]:
235                 return j
236         return len(bins)-1 # return the last bin if data is larger than the last bin
237
238     roa_b = bin_data(roa_b, roa_bins)
239     Operating_Gross_Margin = bin_data(Operating_Gross_Margin, ogm_bins)
240     Persistent_EPS = bin_data(Persistent_EPS, p_eps_bins)
241     Gross_Profit_to_Sales = bin_data(Gross_Profit_to_Sales, g_p_t_s)
242     Cash_Total_Assets = bin_data(Cash_Total_Assets, c_t_a)
243     Debt_Ratio = bin_data(Debt_Ratio, d_r)
244     Net_Worth_Assets = bin_data(Net_Worth_Assets, n_w_a)
245     liability_to_Equity = bin_data(Liability_to_Equity, l_t_e)
246     cash_Flow_Rate = bin_data(Cash_Flow_Rate, c_f_r)
247     cash_Flow_Per_Share = bin_data(Cash_Flow_Per_Share, c_f_p_s)
248     CFO_to_Assets = bin_data(CFO_to_Assets, c_f_o_a)
249     cash_Flow_To_Equity = bin_data(Cash_Flow_To_Equity, cfo_t_a)
250     cash_Flow_To_Liabilities = bin_data(Cash_Flow_To_Liabilities, c_f_t_l)
251     After_tax_Net_Profit_Growth_Rate = bin_data(After_tax_Net_Profit_Growth_Rate, a_t_n_p_g_r)
```

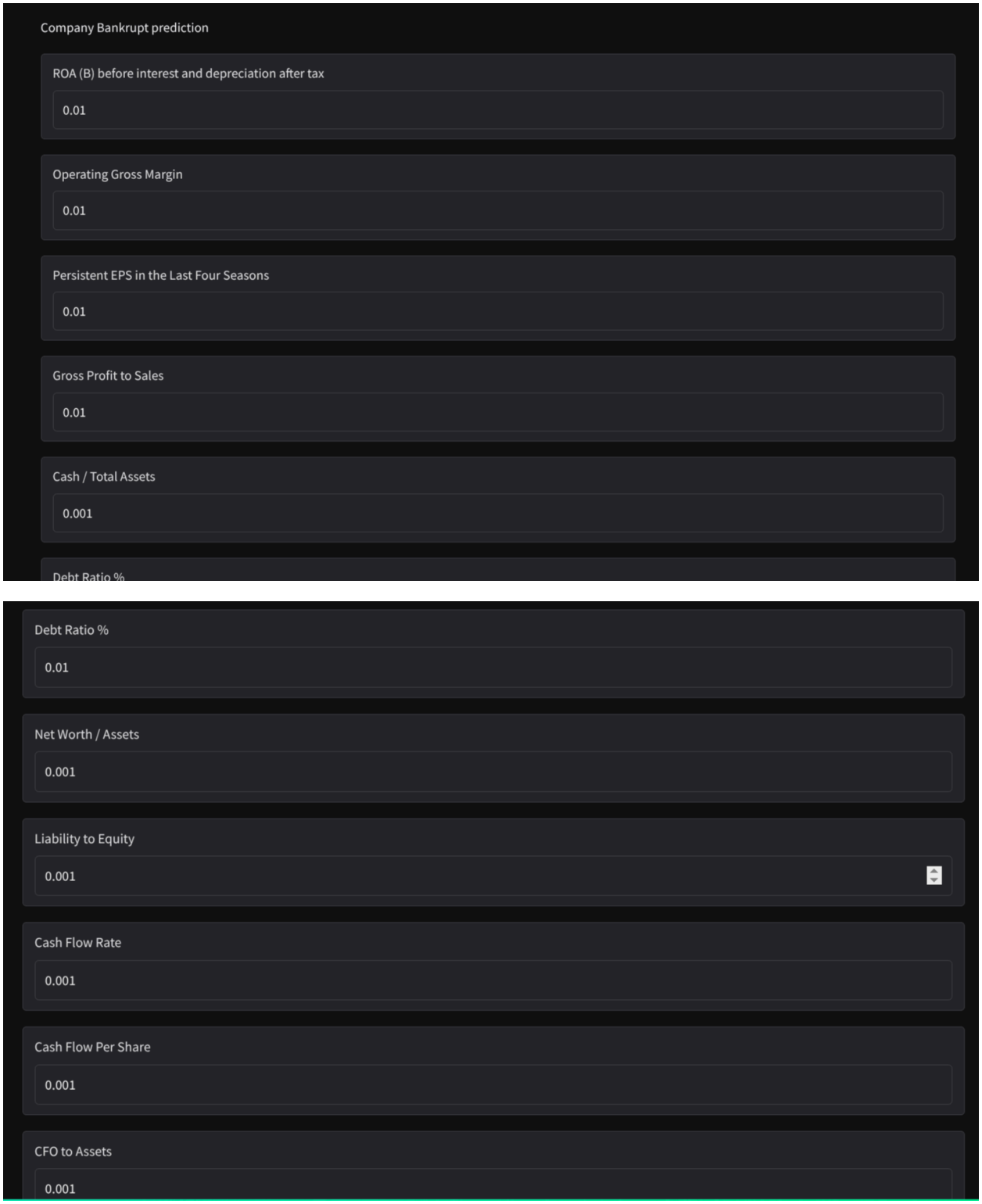
Figure 6.1.4 map the value to the bin value in UI.py

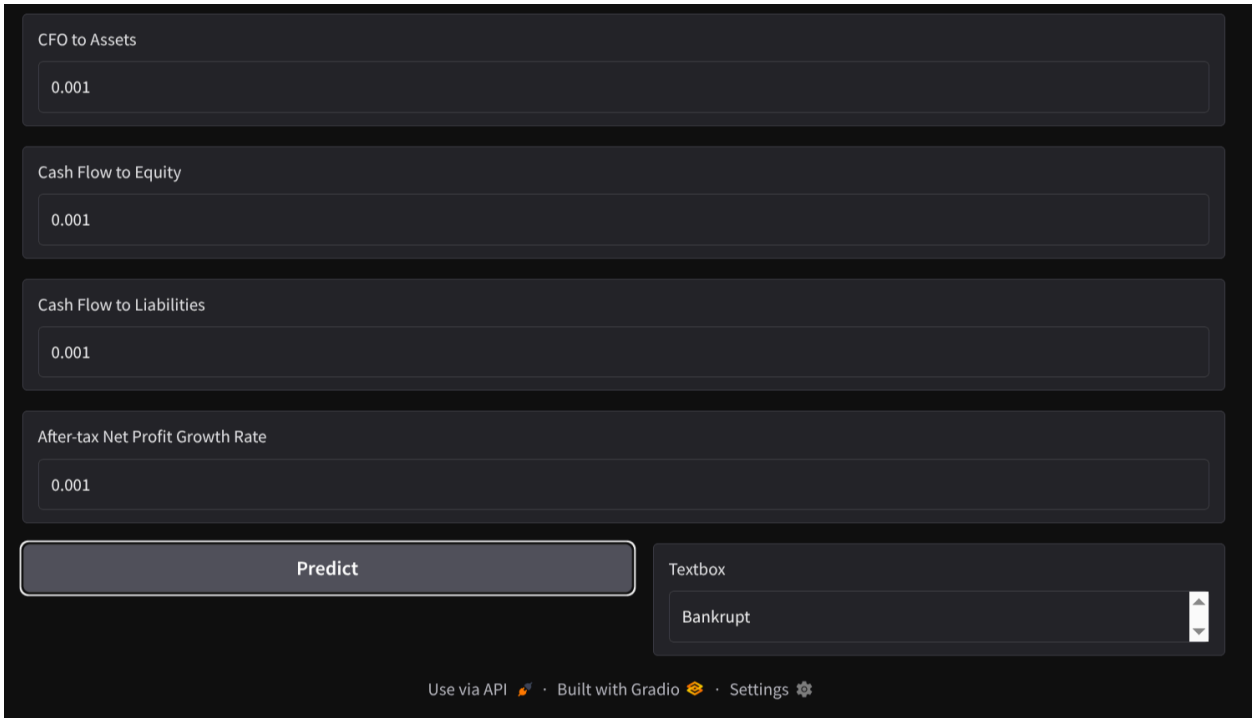
Model Prediction

```
286     input_data = loaded_scaler_b.transform(input_data)
287     prediction = loaded_model.predict(input_data)
288     if prediction[0] == 1:
289         prediction = "Bankrupt"
290     else:
291         prediction = "Not Bankrupt"
292     return prediction
```

Figure 6.1.5 Model prediction in UI.py

Web application Interface





The web application interface is titled “Bankruptcy Prediction.” It provides an input form where a user can enter several financial ratios to predict whether a company is likely to become bankrupt. The form elements are as follows:

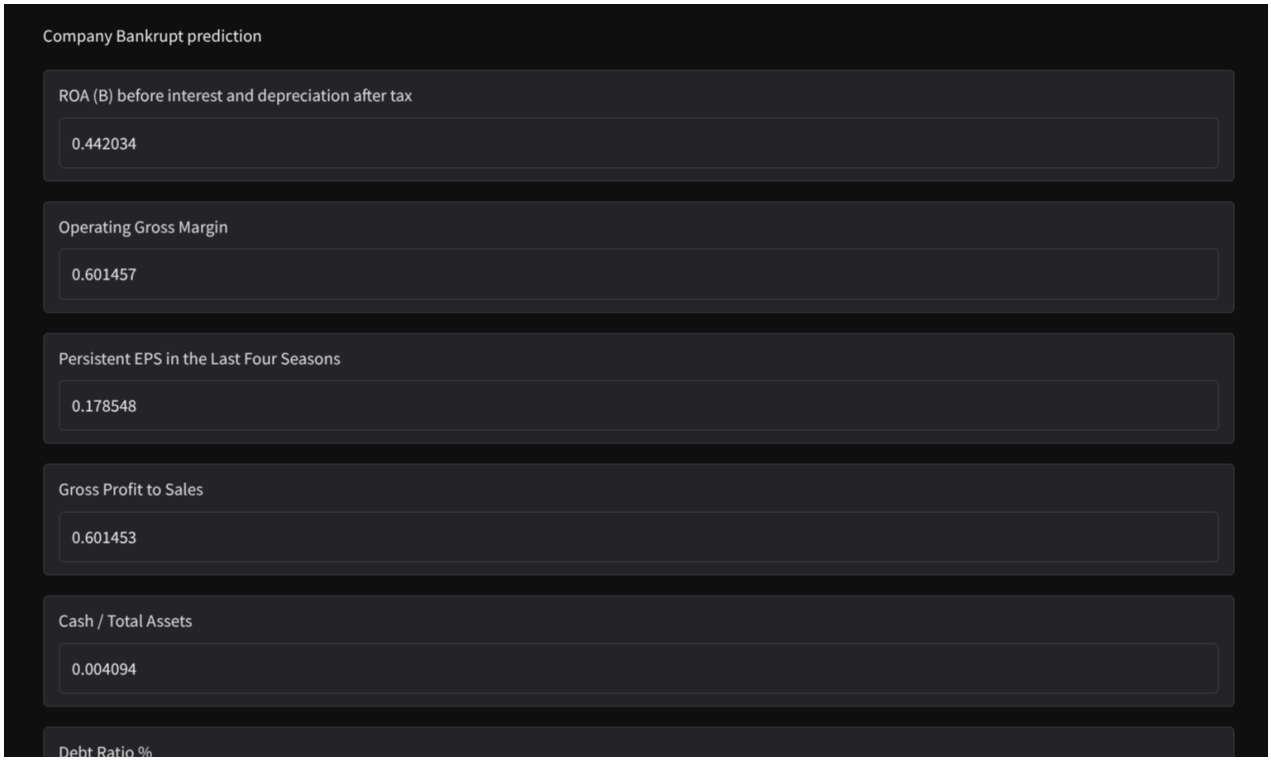
The web application interface is titled “Company Bankrupt Prediction.” It provides an input form where a user can enter several key financial indicators to estimate the likelihood that a company will go bankrupt. The visible form elements are as follows:

1. ROA (B) before interest and depreciation after tax: A field where the user inputs the company’s return on assets before interest and depreciation, after tax.
2. Operating Gross Margin: A field for entering the operating gross margin, reflecting core profitability.
3. Persistent EPS in the Last Four Seasons: A numeric field where the user specifies the company’s earnings per share persistence over the last four seasons.
4. Gross Profit to Sales: A field for inputting the ratio of gross profit to total sales.
5. Cash / Total Assets: A field where the user enters the proportion of cash relative to total assets.
6. Debt Ratio %: A field where the user enters the company’s debt ratio as a percentage, indicating the proportion of total assets financed by debt.

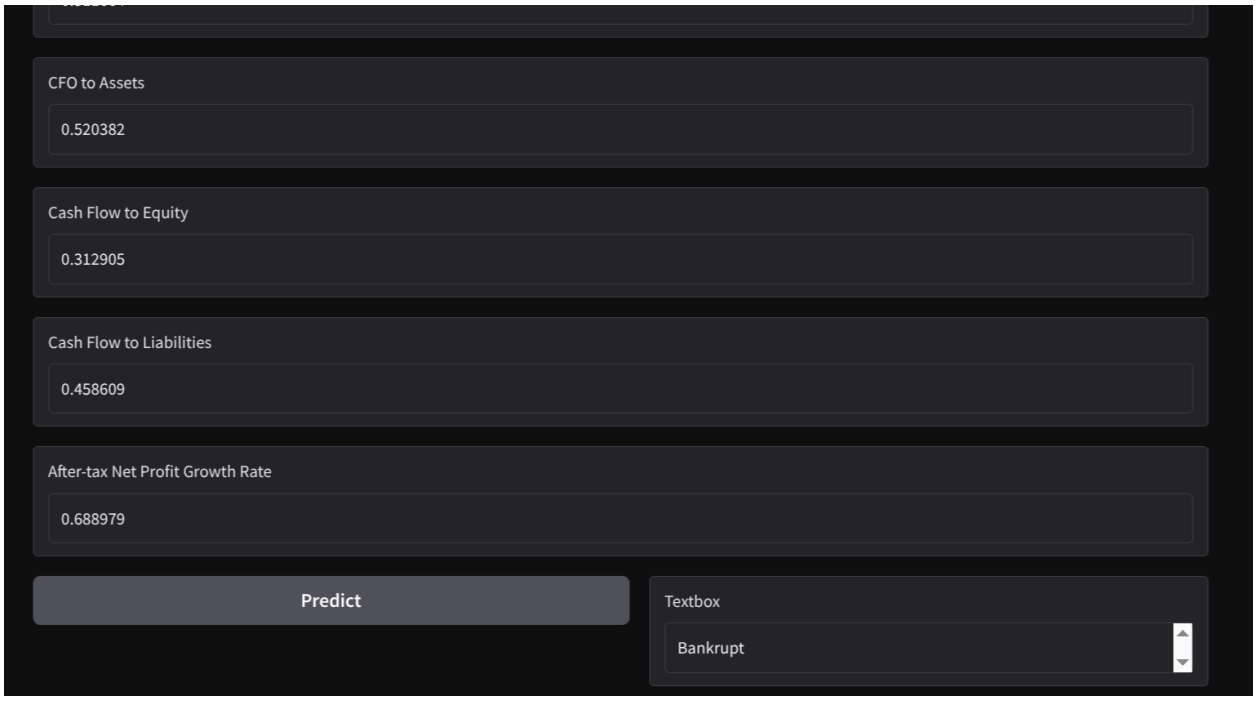
7. Net Worth / Assets: A field for specifying the ratio of shareholders' equity (net worth) to total assets.
8. Liability to Equity: A numeric field where the user inputs the ratio of total liabilities to shareholders' equity.
9. Cash Flow Rate: A field for entering the company's cash flow rate, reflecting the speed at which cash is generated.
10. Cash Flow Per Share: A field where the user provides the cash flow available per outstanding share.
11. CFO to Assets: A field where the user enters the ratio of cash flow from operations to total assets.
12. Cash Flow to Equity: A field for inputting the cash flow available to equity holders relative to equity.
13. Cash Flow to Liabilities: A field for entering the cash flow relative to total liabilities.
14. After-tax Net Profit Growth Rate: A field where the user specifies the company's after-tax net profit growth rate.

At the bottom of the form, there is a "Predict" button. When the user clicks this button, the application processes the input values and displays a textual prediction result (for example, "Bankrupt" or "Non-Bankrupt") in the output textbox.

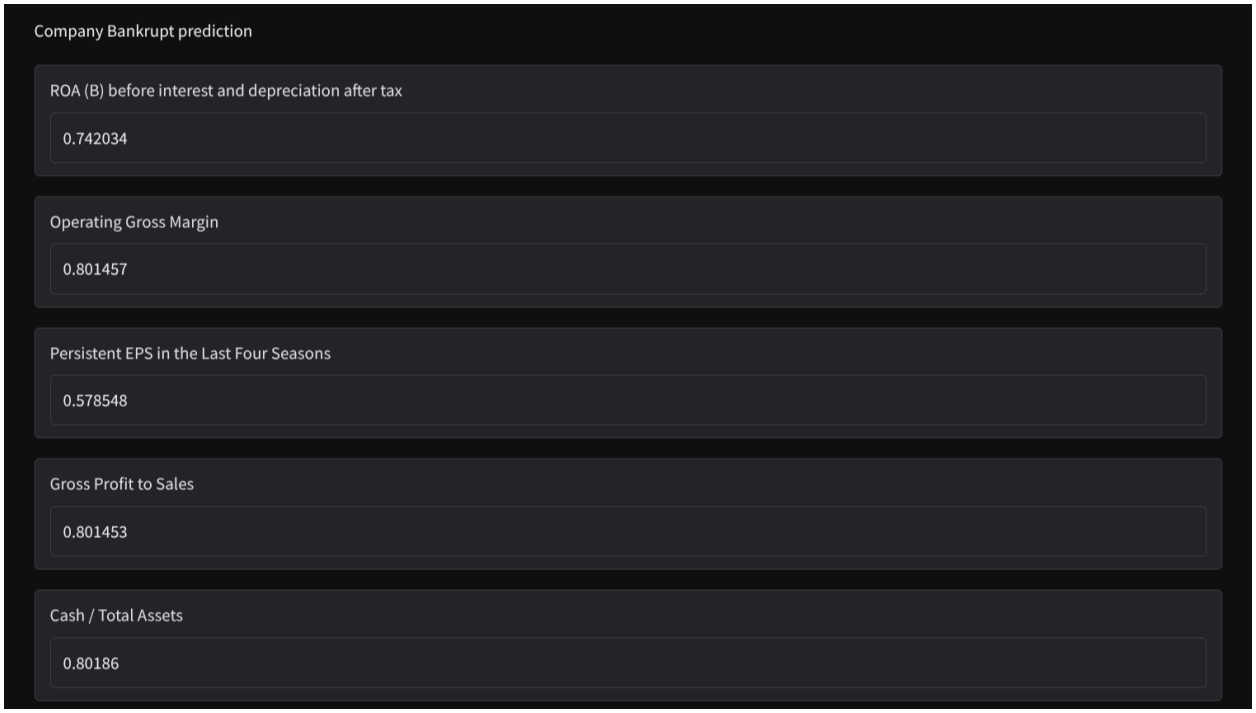
Output (Bankrupt)

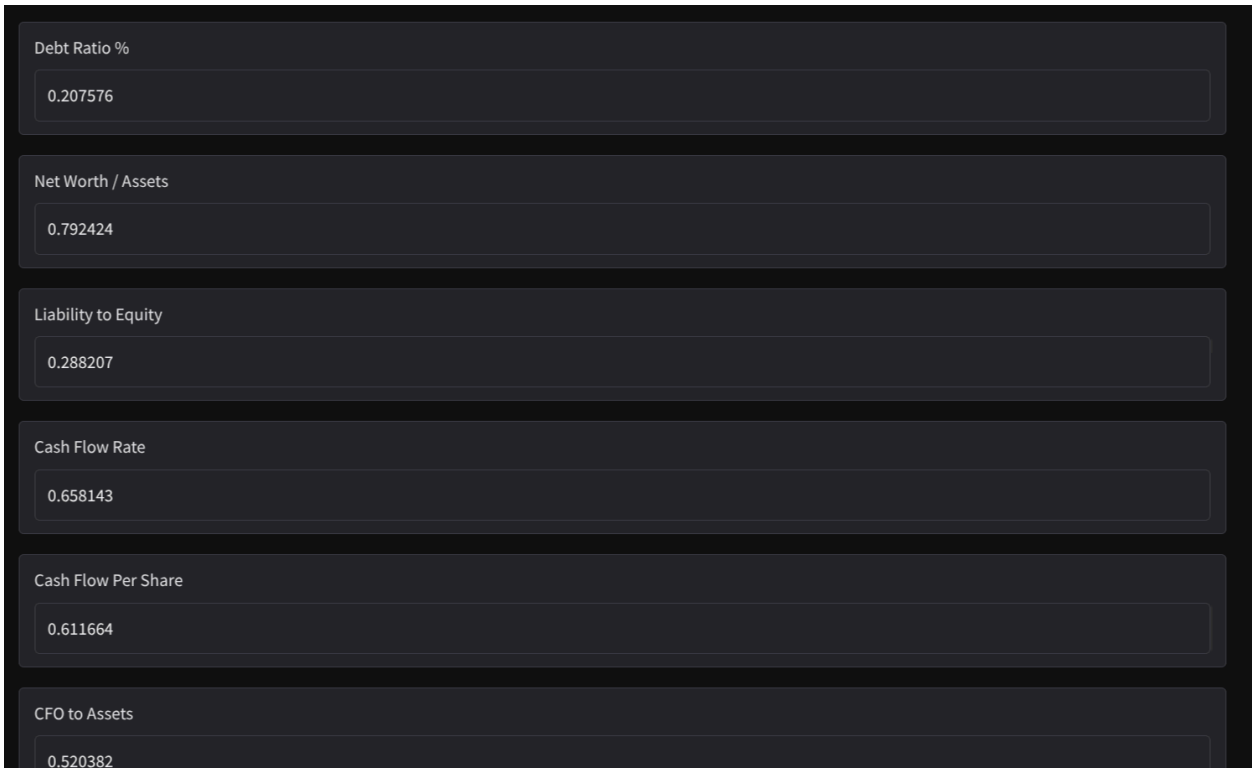


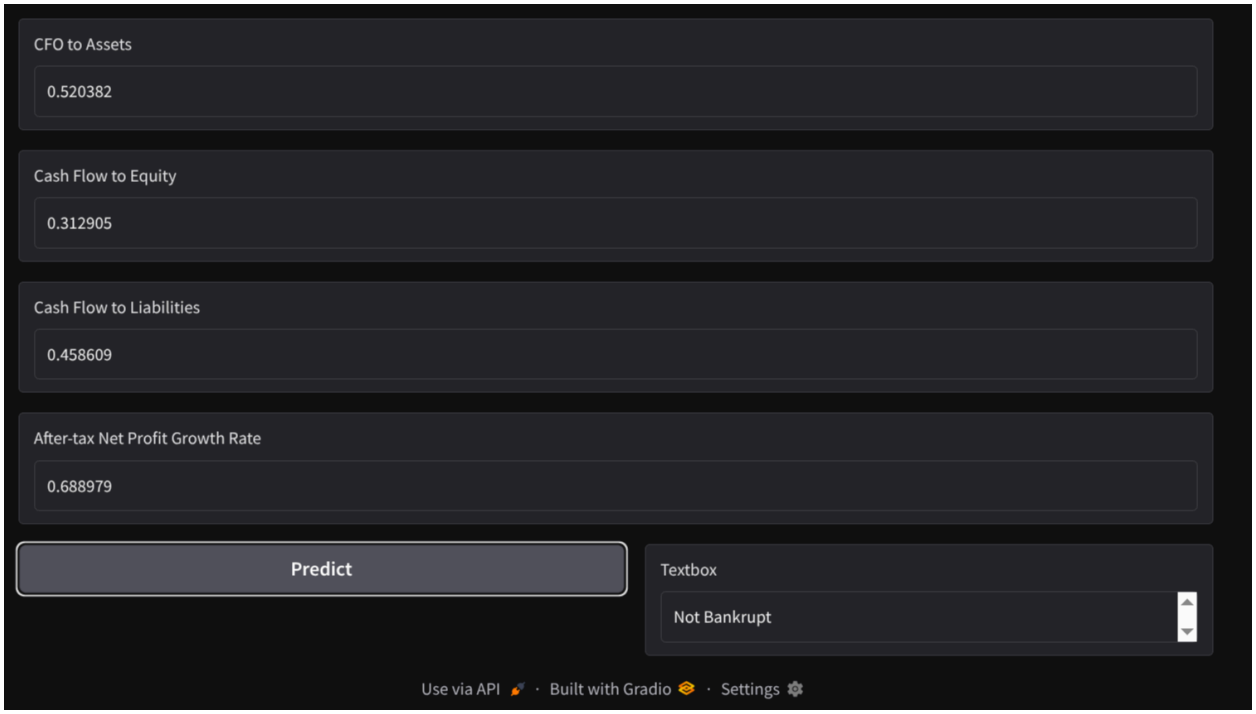

Debt Ratio %
0.207576
Net Worth / Assets
0.792424
Liability to Equity
0.288207
Cash Flow Rate
0.458143
Cash Flow Per Share
0.311664
CFO to Assets
0.520382


CFO to Assets
0.520382
Cash Flow to Equity
0.312905
Cash Flow to Liabilities
0.458609
After-tax Net Profit Growth Rate
0.688979
Predict
Textbox
Bankrupt

Output (Not Bankrupt)






CFO to Assets
0.520382
Cash Flow to Equity
0.312905
Cash Flow to Liabilities
0.458609
After-tax Net Profit Growth Rate
0.688979
Predict
Textbox
Not Bankrupt
Use via API · Built with Gradio · Settings

6. Conclusion

In this project, a machine learning system was developed to predict company bankruptcy using financial ratios derived from the Taiwanese bankruptcy dataset. The workflow covered data preprocessing, exploratory data analysis, feature binning, handling class imbalance with SMOTE, model training, evaluation, and deployment through a web-based interface. The deployed application allows users to input key financial indicators and immediately receive a “Bankrupt” or “Non-Bankrupt” prediction, making the model practically usable for decision support.

Several algorithms were explored, including Gaussian Naive Bayes, K-Nearest Neighbors, Decision Tree, and Random Forest, with each model evaluated using accuracy, precision, recall, and F1-Score for the minority class (Bankrupt = 1). Among these, the Random Forest classifier trained on binned features and enhanced with SMOTE achieved the strongest balance across metrics, obtaining the highest overall accuracy and the best F1-Score for the bankrupt class, while maintaining strong performance on non-bankrupt firms.

The results indicate that machine learning can be an effective tool for supporting financial risk assessment, helping stakeholders identify at-risk companies earlier and potentially reduce credit and investment losses. However, the study also exposes limitations: performance on the minority bankrupt class still lags behind that of the majority class, and predictions may be affected by data quality, feature noise, and residual class imbalance. Future work could incorporate additional financial or macroeconomic variables, more sophisticated cost-sensitive or ensemble methods, and deeper analysis of model explanations to improve transparency and trust.

Overall, this project demonstrates the potential of data-driven bankruptcy prediction and provides a solid foundation for further research and enhancement. By integrating the model into a user-friendly web interface, analysts, lenders, and managers can leverage quantitative insights to complement traditional financial analysis and make more informed, proactive decisions about corporate solvency.

7. Additional Adds On

Improved neural network setup

A second series of experiments was conducted using a simplified neural network trained on a properly stratified train–validation–test split without SMOTE. The input data were divided into 70% training, 15% validation, and 15% testing, while preserving the original bankruptcy ratio in each subset. All numeric predictors were standardized using a StandardScaler fitted on the training set and then applied to the validation and test sets to avoid information leakage.

The final model is a compact feedforward network with two hidden layers of 16 and 8 ReLU units, each followed by dropout (0.4 and 0.3 respectively), and a single sigmoid output neuron that estimates the probability of bankruptcy. The model was trained with the Adam optimizer, binary cross-entropy loss, and accuracy, precision, and recall as monitoring metrics, using early stopping with a patience of 8 epochs to prevent overfitting. To address class imbalance directly, class weights of 1.0 for non-bankrupt firms and 15.0 for bankrupt firms were applied so that misclassifying a bankrupt company is penalized much more heavily than misclassifying a healthy company.

Threshold tuning on validation set

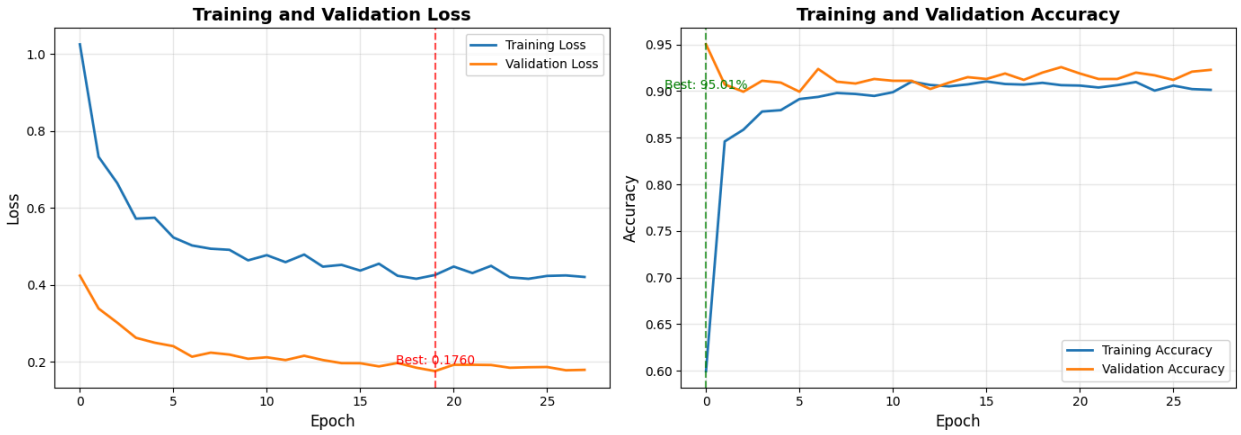
Instead of using the default probability threshold of 0.50, the decision threshold for converting predicted probabilities into class labels was tuned on the validation set. Model predictions on the validation data were scanned over thresholds between 0.10 and 0.55 in steps of 0.05, and for each threshold the recall of the bankrupt (positive) class was computed. The smallest threshold that achieved at least 60% recall on the validation set was selected as the operating point for deployment, ensuring that the model prioritises catching a majority of bankrupt firms even at the cost of additional false alarms.

Test-set results and interpretation

When evaluated on the held-out test set using the tuned threshold of 0.10, the neural network correctly identified 29 out of 33 bankrupt firms, corresponding to a bankrupt recall of 87.9%. Over the same test set, 309 out of 990 healthy firms were incorrectly flagged as bankrupt, which corresponds to a false-alarm rate of 31.2% among non-bankrupt companies. These results illustrate a deliberate shift from the earlier high-accuracy but useless model (0% bankrupt recall) toward a more practical classifier that captures the vast majority of truly bankrupt firms, at the acceptable cost of reviewing a moderate number of false alarms.

Training behaviour

Across 28 epochs, both training and validation curves show stable learning, with loss decreasing and accuracy increasing on both sets. The best validation loss of 0.1760 is reached around epoch 19, after which the validation curve remains flat, indicating that early stopping at this point prevents unnecessary training without any sign of divergence or instability.



Generalization assessment

At the final epoch, the model attains a training loss of 0.4205 and validation loss of 0.1793, producing a negative loss gap of −0.2412, which suggests that the model does not overfit the training data. Similarly, training accuracy (90.13%) and validation accuracy (92.28%) are close, with a small gap of −2.15%, showing that the network generalizes well and performs slightly better on unseen validation data than on the training set.

Conclusion

The overall results show that traditional machine learning models and the neural network each capture different strengths for bankruptcy prediction, and that careful handling of class imbalance and decision thresholds is more important than maximising raw accuracy.

Comparative performance overview

Among the classical models, Random Forest and K-Nearest Neighbors consistently achieve the highest overall accuracy and minority-class F1-scores, especially when trained on binned features (Dataset ‘b’), while Gaussian Naive Bayes and single Decision Trees serve mainly as interpretable baselines. Random Forest with binned features reaches about 95–96% accuracy and improves bankrupt-class precision and F1 compared with the same model on original features, confirming that feature binning helps complex, non-linear models separate the classes more effectively.

Strengths and limits of classical models

Gaussian Naive Bayes offers fast, simple baselines but its independence and Gaussian assumptions limit minority-class precision, leading to many false bankruptcy alarms despite reasonably high recall and accuracy. KNN benefits strongly from scaling and binning, achieving higher accuracy and a better bankrupt-class F1 than Naive Bayes, yet still struggles with low precision because of the imbalanced data and sensitivity to local noise. Decision Trees provide interpretable rules and strong performance for non-bankrupt firms, but they show only moderate recall and low F1 for bankrupt companies and are prone to overfitting dominant majority patterns without careful regularisation.

Neural network behaviour and improvements

The initial neural network configuration, trained on oversampled data and evaluated with a fixed 0.5 threshold, achieved very high test accuracy but completely failed to detect any bankrupt firms, illustrating how misleading accuracy can be on heavily imbalanced datasets. After redesigning the pipeline with a stratified train–validation–test split, standardisation, class-weighted training, early stopping, and validation-based threshold tuning, the simplified neural network reaches around 88% recall for bankrupt firms at the cost of roughly 31% false alarms among healthy companies, while maintaining good validation accuracy and no clear signs of overfitting.

Taken together, the experiments show that ensemble tree methods such as Random Forest offer the best balance between accuracy, robustness, and interpretability for this bankruptcy dataset, especially with binned features, while KNN provides a strong but more sensitive alternative. The improved neural network does not maximise overall accuracy, but it delivers the highest sensitivity to bankrupt firms once class weights and threshold tuning are applied, making it attractive for risk-averse settings where missing a true bankruptcy is far more costly than investigating additional false positives. For practical deployment, a Random Forest or class-weighted neural network operating at an explicitly chosen recall

level for the bankrupt class, combined with further imbalance-handling techniques (e.g., resampling or cost-sensitive learning), would provide a strong foundation for data-driven bankruptcy risk assessment.

8. Reference

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. Y. (2019). Gradio: Hassle-Free sharing and testing of ML models in the Wild. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1906.02569

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *SciKit-Learn: Machine Learning in Python*. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

UCI Machine Learning Repository. (2020). Taiwanese Bankruptcy Prediction [Dataset]. In *UC Irvine*. https://doi.org/10.24432/c5004d

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Pham, H. V., Chu, T., Le, T. M., Tran, H. M., Tran, H. T., Yen, K. N., & Dao, S. V. T. (2025). Comprehensive evaluation of bankruptcy prediction in Taiwanese firms using multiple machine learning models. *International Journal of Technology*, *16*(1), 289. https://doi.org/10.14716/ijtech.v16i1.7227

Raymond, F. E. (2017). A modern validation of hotelling's rule. *Theoretical Economics Letters*, *07*(07), 2070–2080. https://doi.org/10.4236/tel.2017.77140