Case Study: Predicting Company Bankruptcy Using Financial Data

DATA UNDERSTANDING AND PREPROCESSING

Importing Important Libraries for DataFrames and Data Loading

```
import pandas as pd
import numpy as np
```

Importing Matplotlib & Seaborn for Plotting Graphs

```
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing Statsmodels for Statistical Tests`

```
import statsmodels
import statsmodels.api as sm
from statsmodels.formula.api import ols
import statsmodels.stats.multicomp
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

UPDATING DATA ON GOOGLE COLAB

```
from google.colab import files
uploaded=files.upload() # data.csv

# Import OS to show current working directory.
import os
os.getcwd()
#os.chdir(r'Downloads\data.csv')
```

```
Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving data.csv to data.csv
'/content'
```

```
import os
os.getcwd()
```

```
'/content'
```

Loading Data of data.csv into DataFrame

```
df=pd.read_csv('data.csv') ### Loading Data of data.csv into DataFrame
```

Getting Important Detailing of data.csv

```
df.head(20)
### Getting first 20 rows along with all columns of df made.
```

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	...	
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	0.796887	0.808809	0.302646	...	0.
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	0.797380	0.809301	0.303556	...	0.
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	0.796403	0.808388	0.302035	...	0.
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	0.796967	0.808966	0.303350	...	0.
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	0.797366	0.809304	0.303475	...	0.
5	1	0.388680	0.415177	0.419134	0.590171	0.590251	0.998758	0.796903	0.808771	0.303116	...	0.
6	0	0.390923	0.445704	0.436158	0.619950	0.619950	0.998993	0.797012	0.808960	0.302814	...	0.
7	0	0.508361	0.570922	0.559077	0.601738	0.601717	0.999009	0.797449	0.809362	0.303545	...	0.
8	0	0.488519	0.545137	0.543284	0.603612	0.603612	0.998961	0.797414	0.809338	0.303584	...	0.
9	0	0.495686	0.550916	0.542963	0.599209	0.599209	0.999001	0.797404	0.809320	0.303483	...	0.
10	0	0.482475	0.567543	0.538198	0.614026	0.614026	0.998978	0.797535	0.809460	0.303759	...	0.
11	0	0.444401	0.549717	0.498956	0.623712	0.623712	0.998975	0.797443	0.809389	0.303605	...	0.
12	0	0.491152	0.551570	0.543391	0.608131	0.608138	0.999045	0.797429	0.809344	0.303435	...	0.
13	0	0.474041	0.533308	0.523690	0.600578	0.600578	0.998967	0.797368	0.809299	0.303492	...	0.
14	0	0.506703	0.575829	0.569838	0.604686	0.604686	0.999053	0.797514	0.809456	0.303566	...	0.
15	0	0.513821	0.571086	0.558756	0.621773	0.621773	0.999097	0.797507	0.809396	0.303462	...	0.
16	0	0.488909	0.560238	0.540286	0.606524	0.606524	0.998996	0.797877	0.809761	0.304319	...	0.
17	0	0.535953	0.590438	0.580920	0.618451	0.618451	0.999119	0.797588	0.809461	0.303559	...	0.
18	0	0.504071	0.559802	0.558649	0.598344	0.598344	0.998989	0.797412	0.809334	0.303523	...	0.
19	0	0.487398	0.543720	0.533647	0.636259	0.636252	0.999042	0.797444	0.809340	0.303467	...	0.

20 rows × 96 columns

```
df.shape
### Getting shape of df.
```

```
(6819, 96)
```

```
df.size
### Getting size of df.
```

```
654624
```

```
df.describe()
### Getting Statistical Details of df.
```

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-indus expenditu
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6
mean	0.032263	0.505667	0.560507	0.554691	0.607834	0.607812	0.999028	0.797467	0.809378	
std	0.176710	0.051483	0.050087	0.050277	0.010758	0.010725	0.000120	0.000189	0.000157	
min	0.000000	0.387973	0.455122	0.442034	0.580240	0.580321	0.998781	0.797096	0.809075	
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312	
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375	
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469	
max	1.000000	0.624116	0.669579	0.669348	0.634118	0.633954	0.999282	0.797868	0.809706	

8 rows × 96 columns

```
df.info()
## Summary Statistics.
### Getting info of df.
```

```
 40   Borrowing dependency                                    6819 non-null   float64
 41   Contingent liabilities/Net worth                        6819 non-null   float64
 42   Operating profit/Paid-in capital                        6819 non-null   float64
 43   Net profit before tax/Paid-in capital                   6819 non-null   float64
 44   Inventory and accounts receivable/Net value             6819 non-null   float64
 45   Total Asset Turnover                                    6819 non-null   float64
 46   Accounts Receivable Turnover                            6819 non-null   float64
 47   Average Collection Days                                 6819 non-null   float64
 48   Inventory Turnover Rate (times)                         6819 non-null   float64
 49   Fixed Assets Turnover Frequency                         6819 non-null   float64
 50   Net Worth Turnover Rate (times)                         6819 non-null   float64
 51   Revenue per person                                      6819 non-null   float64
 52   Operating profit per person                             6819 non-null   float64
 53   Allocation rate per person                              6819 non-null   float64
 54   Working Capital to Total Assets                         6819 non-null   float64
 55   Quick Assets/Total Assets                               6819 non-null   float64
 56   Current Assets/Total Assets                             6819 non-null   float64
 57   Cash/Total Assets                                       6819 non-null   float64
 58   Quick Assets/Current Liability                          6819 non-null   float64
 59   Cash/Current Liability                                  6819 non-null   float64
 60   Current Liability to Assets                             6819 non-null   float64
 61   Operating Funds to Liability                            6819 non-null   float64
 62   Inventory/Working Capital                               6819 non-null   float64
 63   Inventory/Current Liability                             6819 non-null   float64
 64   Current Liabilities/Liability                           6819 non-null   float64
 65   Working Capital/Equity                                  6819 non-null   float64
 66   Current Liabilities/Equity                              6819 non-null   float64
 67   Long-term Liability to Current Assets                   6819 non-null   float64
 68   Retained Earnings to Total Assets                       6819 non-null   float64
 69   Total income/Total expense                              6819 non-null   float64
 70   Total expense/Assets                                    6819 non-null   float64
 71   Current Asset Turnover Rate                             6819 non-null   float64
 72   Quick Asset Turnover Rate                               6819 non-null   float64
 73   Working capitcal Turnover Rate                          6819 non-null   float64
 74   Cash Turnover Rate                                      6819 non-null   float64
 75   Cash Flow to Sales                                      6819 non-null   float64
 76   Fixed Assets to Assets                                  6819 non-null   float64
 77   Current Liability to Liability                          6819 non-null   float64
 78   Current Liability to Equity                             6819 non-null   float64
 79   Equity to Long-term Liability                           6819 non-null   float64
 80   Cash Flow to Total Assets                               6819 non-null   float64
 81   Cash Flow to Liability                                  6819 non-null   float64
 82   CFO to Assets                                           6819 non-null   float64
 83   Cash Flow to Equity                                     6819 non-null   float64
 84   Current Liability to Current Assets                     6819 non-null   float64
 85   Liability-Assets Flag                                   6819 non-null   float64
 86   Net Income to Total Assets                              6819 non-null   float64
 87   Total assets to GNP price                               6819 non-null   float64
 88   No-credit Interval                                      6819 non-null   float64
 89   Gross Profit to Sales                                   6819 non-null   float64
 90   Net Income to Stockholder's Equity                      6819 non-null   float64
 91   Liability to Equity                                     6819 non-null   float64
 92   Degree of Financial Leverage (DFL)                      6819 non-null   float64
 93   Interest Coverage Ratio (Interest expense to EBIT)      6819 non-null   float64
 94   Net Income Flag                                         6819 non-null   float64
 95   Equity to Liability                                     6819 non-null   float64
dtypes: float64(95), int64(1)
memory usage: 5.0 MB
```

```
df.isnull().sum()
### Checking for null values.
```

```
Bankrupt?                                                 0
 ROA(C) before interest and depreciation before interest  0
 ROA(A) before interest and % after tax                   0
 ROA(B) before interest and depreciation after tax        0
 Operating Gross Margin                                   0
                                                         ..
 Liability to Equity                                      0
 Degree of Financial Leverage (DFL)                       0
 Interest Coverage Ratio (Interest expense to EBIT)       0
 Net Income Flag                                          0
 Equity to Liability                                      0
Length: 96, dtype: int64
```

It shows there are no null values in the data presesnt here.

```
df.duplicated().sum()
### Checking for duplicate values.
```

```
0
```

It shows there are no duplicate values in the data presesnt here.

```
# Check class frequency
df["Bankrupt?"].value_counts(normalize=True)
```

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

```
df.shape
```

```
(6819, 96)
```

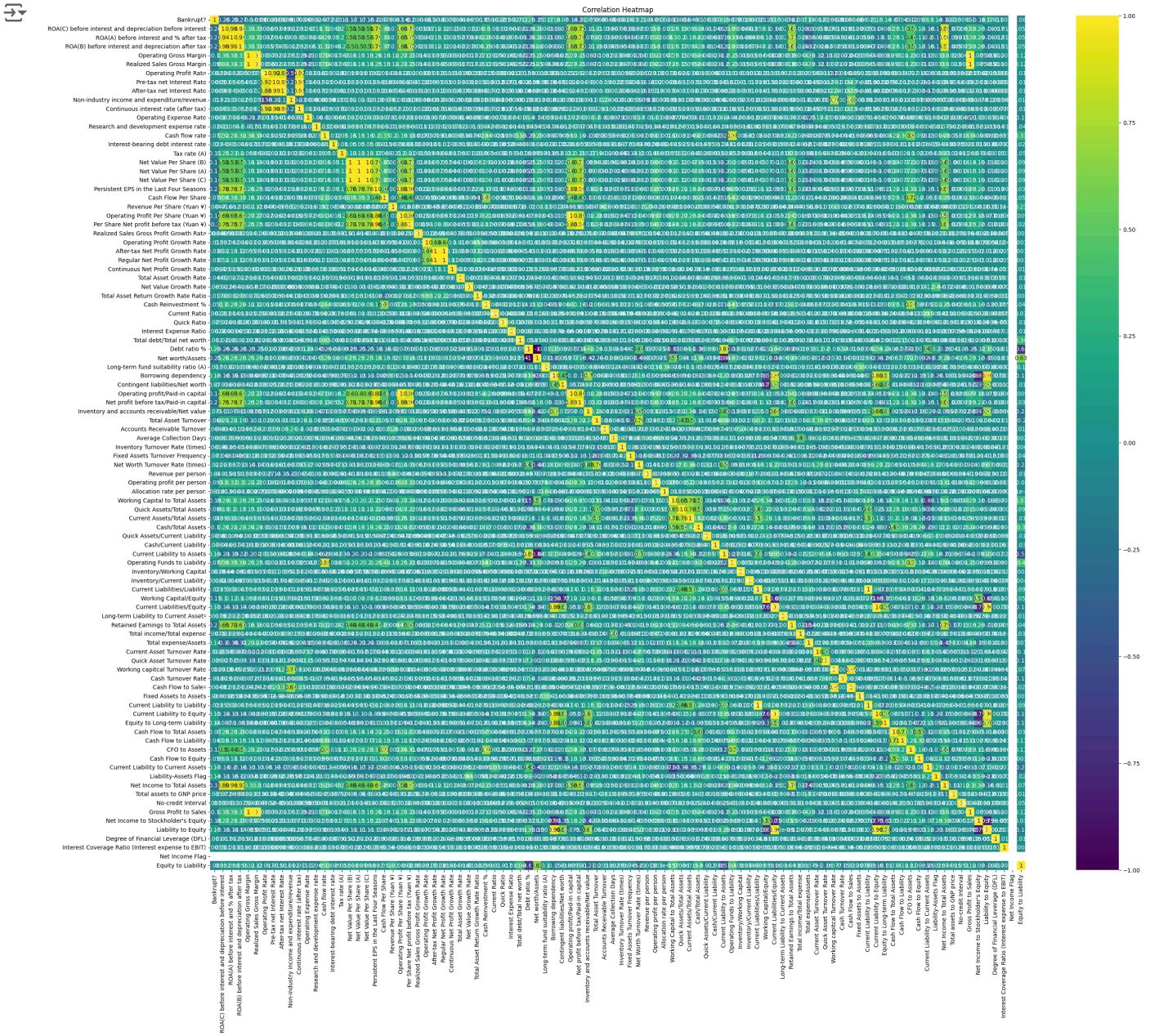
Observation from above Commands.

1. The dataset has 6819 records and 96 columns. Columns includes financial attributes and their bankruptcy status in Bankrupt?.
2. Datatype are float64 or int64.
3. There is no null or duplicated values.

The target column Bankrupt? is imbalanced, with only 3% of total values is 1, which indicates the bankrupt status.

EXPLORATORY DATA ANALYSIS

```
# Checking Correlations between Features using Heatmap
# Creating Correlation Matrix
corr_matrix = df.corr()
# Generating Heatmap
plt.figure(figsize=(32, 27))
sns.heatmap(corr_matrix, annot=True, cmap='viridis')
plt.title("Correlation Heatmap")
plt.show()
```

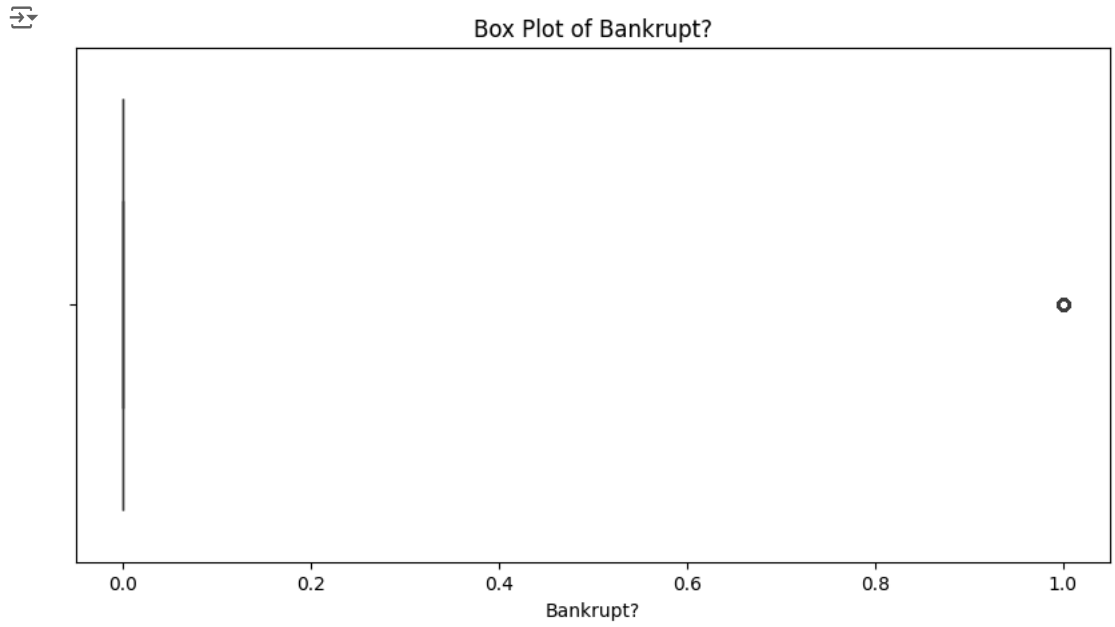

Correlation Heatmap
Bankrupt?
ROA(C) before interest and depreciation before interest
ROA(A) before interest and % after tax
ROA(B) before interest and depreciation after tax
Operating Gross Margin
Realized Sales Gross Margin
Operating Profit Rate
Pre-tax net Interest Rate
After-tax net Interest Rate
Non-industry income and expenditure/revenue
Continuous interest rate (after tax)
Operating Expense Rate
Research and development expense rate
Cash flow rate
Interest-bearing debt interest rate
Tax rate (A)
Net Value Per Share (B)
Net Value Per Share (A)
Net Value Per Share (C)
Persistent EPS in the Last Four Seasons
Cash Flow Per Share
Revenue Per Share (Yuan ¥)
Operating Profit Per Share (Yuan ¥)
Per Share Net profit before tax (Yuan ¥)
Realized Sales Gross Profit Growth Rate
Operating Profit Growth Rate
After-tax Net Profit Growth Rate
Regular Net Profit Growth Rate
Continuous Net Profit Growth Rate
Total Asset Growth Rate
Net Value Growth Rate
Total Asset Return Growth Rate Ratio
Cash Reinvestment %
Current Ratio
Quick Ratio
Interest Expense Ratio
Total debt/Total net worth
Debt ratio %
Net worth/Assets
Long-term fund suitability ratio (A)
Borrowing dependency
Contingent liabilities/Net worth
Operating profit/Paid-in capital
Net profit before tax/Paid-in capital
Inventory and accounts receivable/Net value
Total Asset Turnover
Accounts Receivable Turnover
Average Collection Days
Inventory Turnover Rate (times)
Fixed Assets Turnover Frequency
Net Worth Turnover Rate (times)
Revenue per person
Operating profit per person
Allocation rate per person
Working Capital to Total Assets
Quick Assets/Total Assets
Current Assets/Total Assets
Cash/Total Assets
Quick Assets/Current Liability
Cash/Current Liability
Current Liability to Assets
Operating Funds to Liability
Inventory/Working Capital
Inventory/Current Liability
Current Liabilities/Liability
Working Capital/Equity
Current Liabilities/Equity
Long-term Liability to Current Assets
Retained Earnings to Total Assets
Total income/Total expense
Total expense/Assets
Current Asset Turnover Rate
Quick Asset Turnover Rate
Working capital Turnover Rate
Cash Turnover Rate
Cash Flow to Sales
Fixed Assets to Assets
Current Liability to Liability
Current Liability to Equity
Equity to Long-term Liability
Cash Flow to Total Assets
Cash Flow to Liability
CFO to Assets
Cash Flow to Equity
Current Liability to Current Assets
Liability-Assets Flag
Net Income to Total Assets
Total assets to GNP price
No-credit Interval
Gross Profit to Sales
Net Income to Stockholder's Equity
Liability to Equity
Degree of Financial Leverage (DFL)
Interest Coverage Ratio (Interest expense to EBIT)
Net Income Flag
Equity to Liability
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

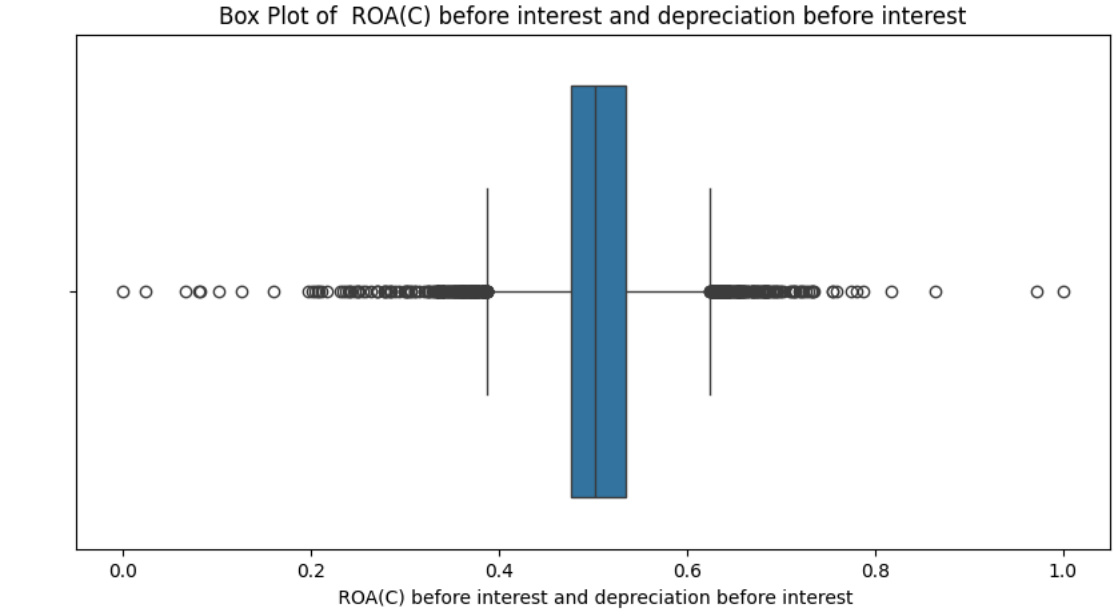
```
'''
 *Analysis:*

- There are several features that have a strong positive correlation with the target variable "Bankrupt?". These include:
    - Net Income to Total Assets
    - Total Debt to Total Assets
    - Debt to Equity
    - Interest Expense to Total Assets
    - Sales to Total Assets
    - Current Assets to Total Assets
    - Inventory to Total Assets
- There are also several features that have a strong negative correlation with the target variable "Bankrupt?". These include:
    - Return on Assets
    - Return on Equity
    - Gross Profit Margin
    - Operating Profit Margin
    - Net Profit Margin
    - Quick Ratio
    - Current Ratio
- The heatmap also reveals that there are several features that are highly correlated with each other. For example:
    - Total Debt to Total Assets and Debt to Equity
    - Sales to Total Assets and Current Assets to Total Assets
    - Inventory to Total Assets and Current Assets to Total Assets

*Recommendations:*

- Given the strong correlations between several features and the target variable, it is important to carefully select the features that
- It may also be beneficial to consider using dimensionality reduction techniques, such as principal component analysis (PCA), to reduce

*Insights:*

- The heatmap provides valuable insights into the relationships between the different features in the dataset. This information can be u
    - Identify potential risk factors for bankruptcy.
    - Develop early warning systems for bankruptcy.
    - Improve the accuracy of predictive models for bankruptcy.
'''
```

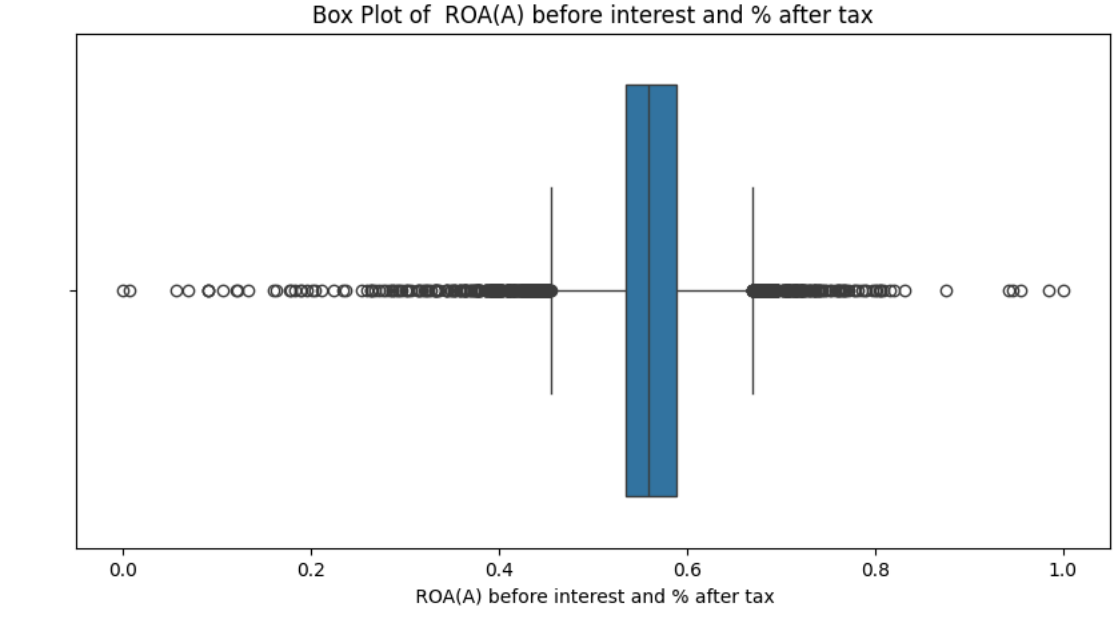
```
'\n *Analysis:*\n\n- There are several features that have a strong positive correlation with the target variable "Bankrupt?". These
include:\n    - Net Income to Total Assets\n    - Total Debt to Total Assets\n    - Debt to Equity\n    - Interest Expense to Total
Assets\n    - Sales to Total Assets\n    - Current Assets to Total Assets\n    - Inventory to Total Assets\n- There are also severa
l features that have a strong negative correlation with the target variable "Bankrupt?". These include:\n    - Return on Assets\n
- Return on Equity\n    - Gross Profit Margin\n    - Operating Profit Margin\n    - Net Profit Margin\n    - Quick Ratio\n    - Cur
rent Ratio\n- The heatmap also reveals that there are several features that are highly correlated with each other. For example:\n
- Total Debt to Total Assets and Debt to Equity\n    - Sales to Total Assets and Current Assets to Total Assets\n    - Inventory to
Total Assets and Current Assets to Total Assets\n\n*Recommendations:*\n\n- Gi  '
```

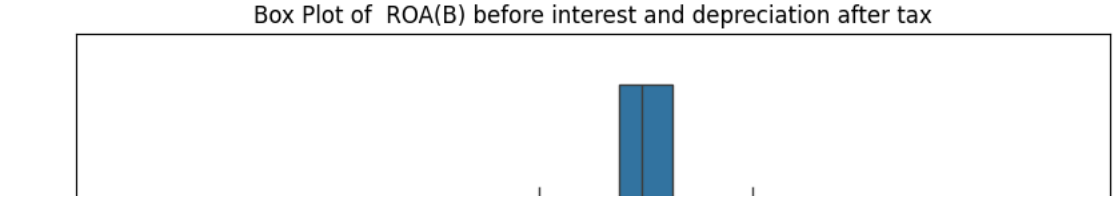
DETECTING OUTLIERS FROM df(DATAFRAME)

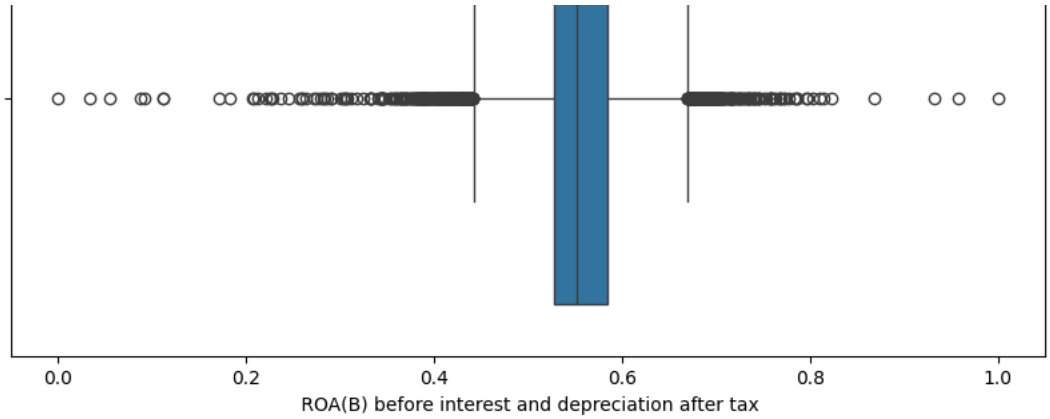
```
# Detect outliers using IQR
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

# Define lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Visualize outliers for a few selected features using box plots
selected_features = df.columns[:5]  # Replace with actual feature names or a subset of features
for feature in selected_features:
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df[feature])
    plt.title(f'Box Plot of {feature}')
    plt.show()

# Cap and floor outliers
for column in df.columns:
    if df[column].dtype != 'object' and column != 'Bankrupt?':
        df[column] = np.where(df[column] > upper_bound[column], upper_bound[column], df[column])
        df[column] = np.where(df[column] < lower_bound[column], lower_bound[column], df[column])

# Verify that outliers have been capped and floored
for feature in selected_features:
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df[feature])
    plt.title(f'Box Plot of {feature} After Capping and Flooring')
    plt.show()

# Check for remaining outliers
outliers_after = ((df < lower_bound) | (df > upper_bound)).sum()
print("Number of outliers in each feature after capping and flooring:\n", outliers_after)

# Generate descriptive statistics
print(df.describe())
```

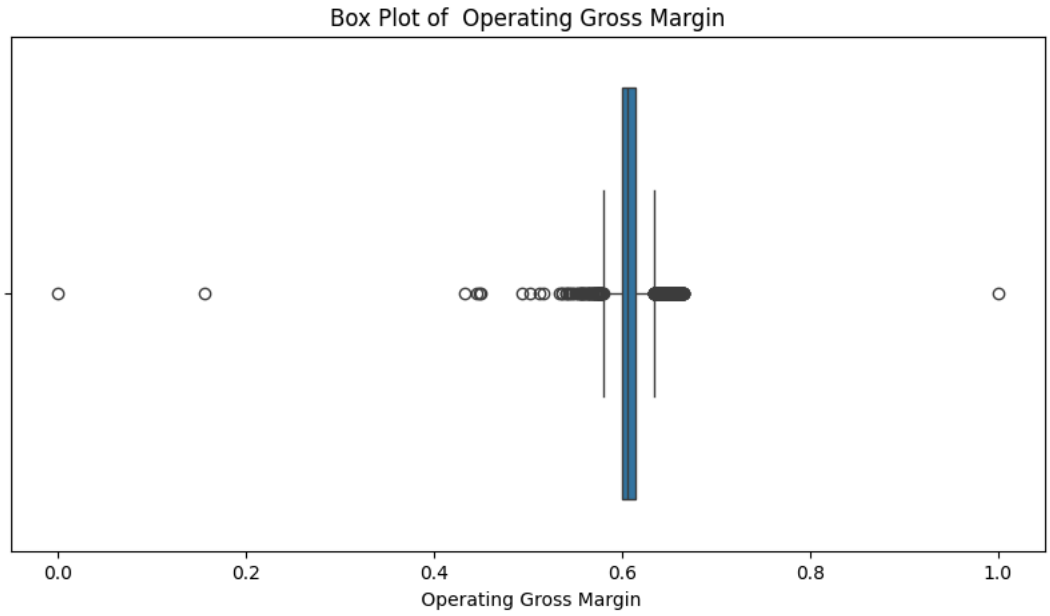

Box Plot of Bankrupt?
0.0
0.2
0.4
0.6
0.8
1.0
Bankrupt?

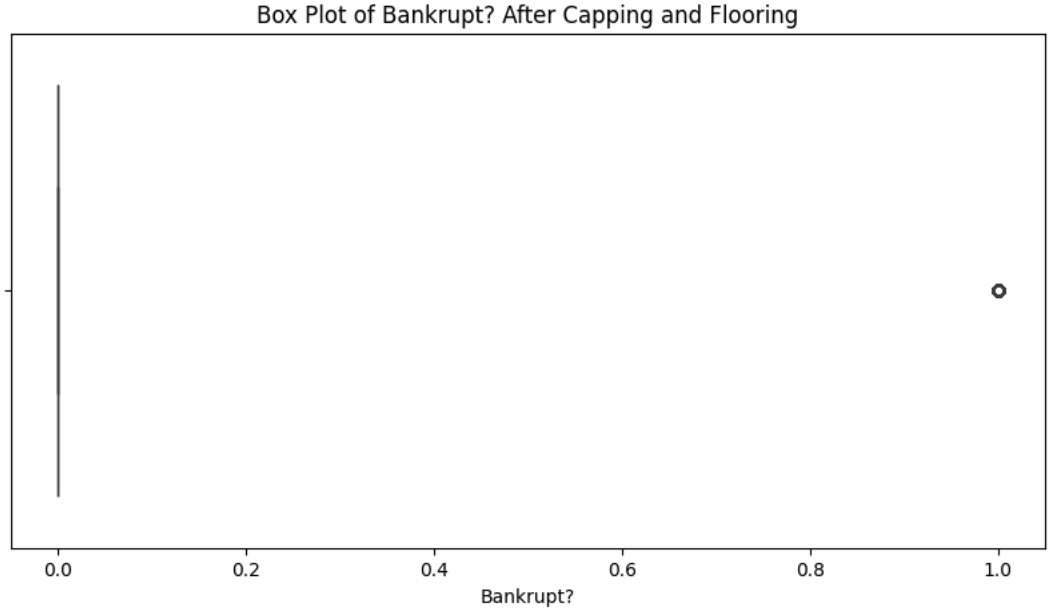

Box Plot of ROA(C) before interest and depreciation before interest
0.0
0.2
0.4
0.6
0.8
1.0
ROA(C) before interest and depreciation before interest

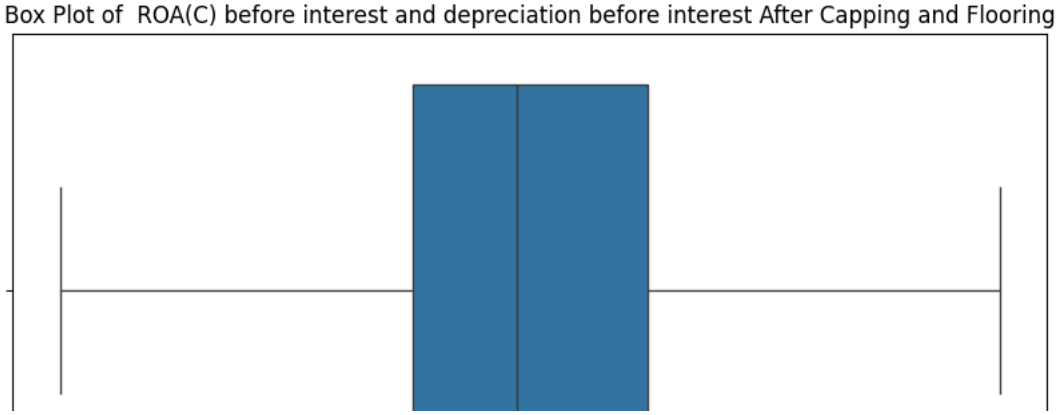

Box Plot of ROA(A) before interest and % after tax
0.0
0.2
0.4
0.6
0.8
1.0
ROA(A) before interest and % after tax


Box Plot of ROA(B) before interest and depreciation after tax

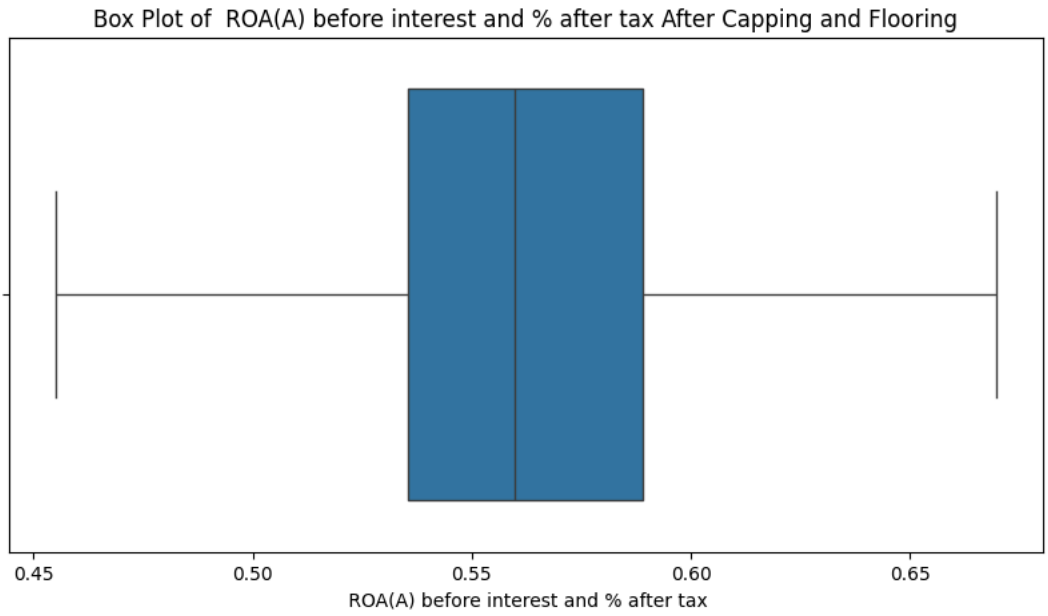

0.0
0.2
0.4
0.6
0.8
1.0
ROA(B) before interest and depreciation after tax

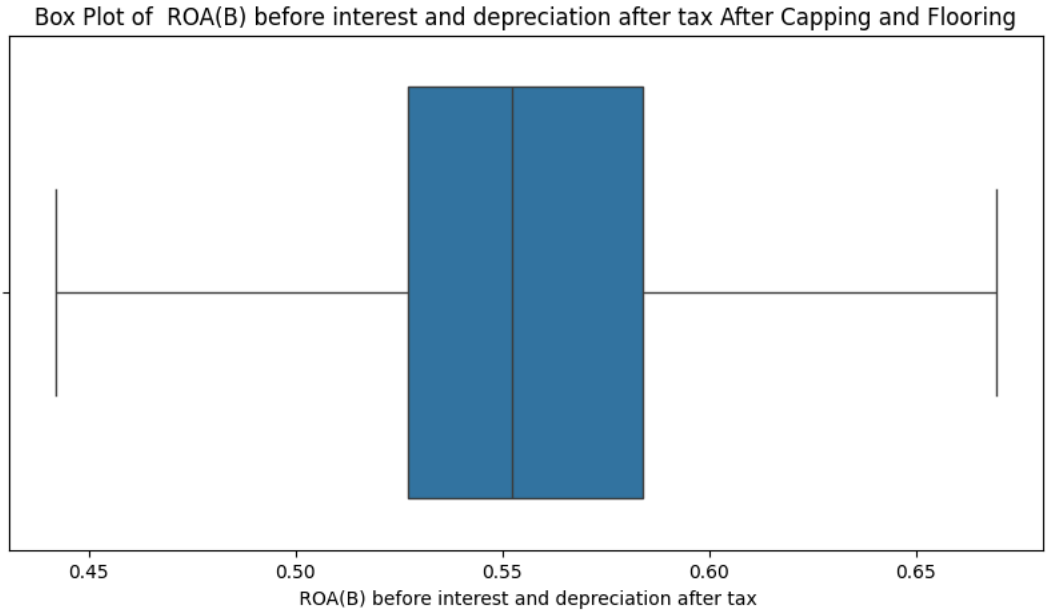

Box Plot of Operating Gross Margin
0.0
0.2
0.4
0.6
0.8
1.0
Operating Gross Margin

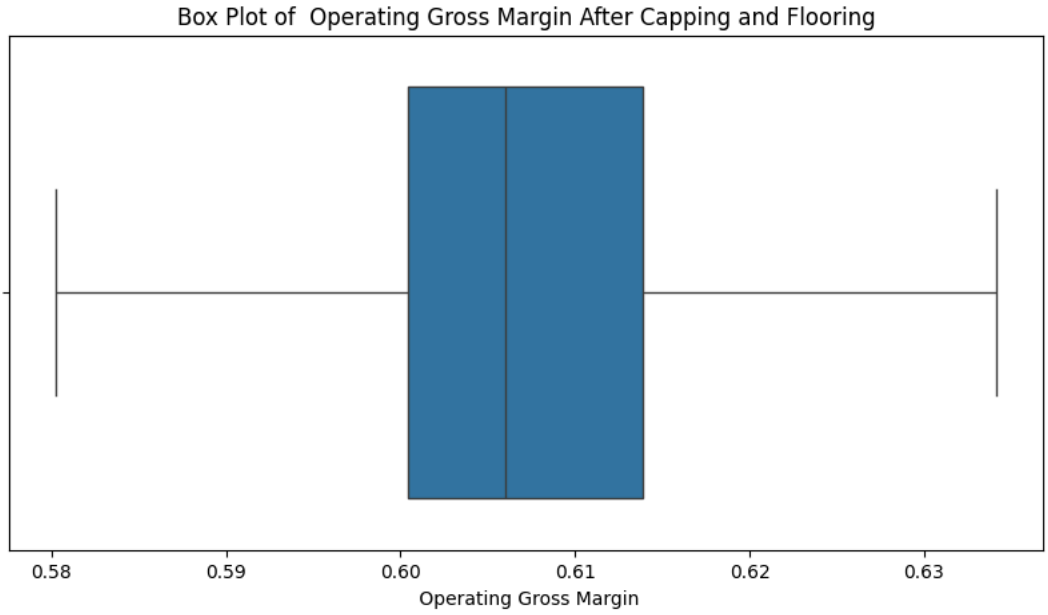

Box Plot of Bankrupt? After Capping and Flooring
0.0
0.2
0.4
0.6
0.8
1.0
Bankrupt?


Box Plot of ROA(C) before interest and depreciation before interest After Capping and Flooring


0.40
0.45
0.50
0.55
0.60
ROA(C) before interest and depreciation before interest


Box Plot of ROA(A) before interest and % after tax After Capping and Flooring
0.45
0.50
0.55
0.60
0.65
ROA(A) before interest and % after tax


Box Plot of ROA(B) before interest and depreciation after tax After Capping and Flooring
0.45
0.50
0.55
0.60
0.65
ROA(B) before interest and depreciation after tax


Box Plot of Operating Gross Margin After Capping and Flooring
0.58
0.59
0.60
0.61
0.62
0.63
Operating Gross Margin

```
Number of outliers in each feature after capping and flooring:
 Bankrupt?                                                   220
 ROA(C) before interest and depreciation before interest      0
 ROA(A) before interest and % after tax                       0
 ROA(B) before interest and depreciation after tax            0
 Operating Gross Margin                                       0
                                                           ...
 Liability to Equity                                          0
 Degree of Financial Leverage (DFL)                           0
 Interest Coverage Ratio (Interest expense to EBIT)           0
 Net Income Flag                                              0
 Equity to Liability                                          0
Length: 96, dtype: int64
         Bankrupt?   ROA(C) before interest and depreciation before interest  \
count  6819.000000                                          6819.000000
mean      0.032263                                             0.505667
std       0.176710                                             0.051483
min       0.000000                                             0.387973
25%       0.000000                                             0.476527
50%       0.000000                                             0.502706
75%       0.000000                                             0.535563
max       1.000000                                             0.624116

        ROA(A) before interest and % after tax  \
count                              6819.000000
mean                                  0.560507
std                                   0.050087
min                                   0.455122
25%                                   0.535543
50%                                   0.559802
75%                                   0.589157
max                                   0.669579

        ROA(B) before interest and depreciation after tax  \
count                                        6819.000000
mean                                            0.554691
std                                             0.050277
min                                             0.442034
25%                                             0.527277
50%                                             0.552278
75%                                             0.584105
max                                             0.669348

        Operating Gross Margin   Realized Sales Gross Margin  \
count              6819.000000                  6819.000000
mean                  0.607834                     0.607812
std                   0.010758                     0.010725
min                   0.580240                     0.580321
25%                   0.600445                     0.600434
50%                   0.605997                     0.605976
75%                   0.613914                     0.613842
max                   0.634118                     0.633954

        Operating Profit Rate   Pre-tax net Interest Rate  \
count              6819.000000                 6819.000000
mean                  0.999028                    0.797467
std                   0.000120                    0.000189
min                   0.998781                    0.797096
25%                   0.998969                    0.797386
50%                   0.999022                    0.797464
75%                   0.999095                    0.797579
max                   0.999282                    0.797868

        After-tax net Interest Rate  \
count                   6819.000000
mean                       0.809378
std                        0.000157
min                        0.809075
25%                        0.809312
50%                        0.809375
75%                        0.809469
max                        0.809706

        Non-industry income and expenditure/revenue  ...  \
count                                   6819.000000  ...
mean                                       0.303526  ...
std                                        0.000124  ...
min                                        0.303288  ...
25%                                        0.303466  ...
50%                                        0.303525  ...
75%                                        0.303585  ...
max                                        0.303764  ...

        Net Income to Total Assets   Total assets to GNP price  \
count                 6819.000000                 6819.000000
mean                     0.810070                    0.003750
std                      0.027735                    0.003805
min                      0.752193                    0.000000
25%                      0.796750                    0.000904
50%                      0.810619                    0.002085
75%                      0.826455                    0.005270
```

```
max                     0.871012                  0.011819

       No-credit Interval   Gross Profit to Sales  \
count         6819.000000            6819.000000
mean             0.623890               0.607833
std              0.000553               0.010758
min              0.622838               0.580237
25%              0.623636               0.600443
50%              0.623879               0.605998
75%              0.624168               0.613913
max              0.624966               0.634119

       Net Income to Stockholder's Equity   Liability to Equity  \
count                         6819.000000          6819.000000
mean                             0.841060             0.279689
std                              0.002057             0.003566
min                              0.836752             0.270187
25%                              0.840115             0.276944
50%                              0.841179             0.278778
75%                              0.842357             0.281449
max                              0.845720             0.288207

       Degree of Financial Leverage (DFL)  \
count                         6819.000000
mean                             0.026846
std                              0.000138
min                              0.026608
25%                              0.026791
50%                              0.026808
75%                              0.026913
max                              0.027096

       Interest Coverage Ratio (Interest expense to EBIT)   Net Income Flag  \
count                                         6819.000000              6819.0
mean                                             0.565422                 1.0
std                                              0.000627                 0.0
min                                              0.564309                 1.0
25%                                              0.565158                 1.0
50%                                              0.565252                 1.0
75%                                              0.565725                 1.0
max                                              0.566574                 1.0

       Equity to Liability
count          6819.000000
mean              0.042019
std               0.023671
min               0.000000
25%               0.024477
50%               0.033798
75%               0.052838
max               0.095380

[8 rows x 96 columns]
```

```
df.describe()
```

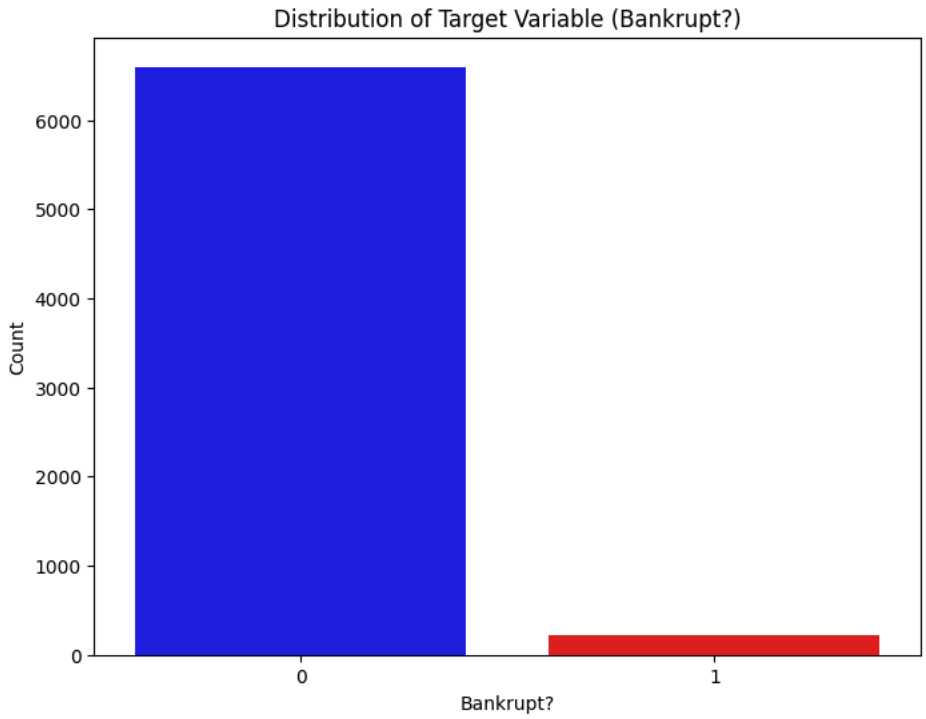
	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-indus expenditu
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6
mean	0.032263	0.505667	0.560507	0.554691	0.607834	0.607812	0.999028	0.797467	0.809378	
std	0.176710	0.051483	0.050087	0.050277	0.010758	0.010725	0.000120	0.000189	0.000157	
min	0.000000	0.387973	0.455122	0.442034	0.580240	0.580321	0.998781	0.797096	0.809075	
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312	
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375	
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469	
max	1.000000	0.624116	0.669579	0.669348	0.634118	0.633954	0.999282	0.797868	0.809706	

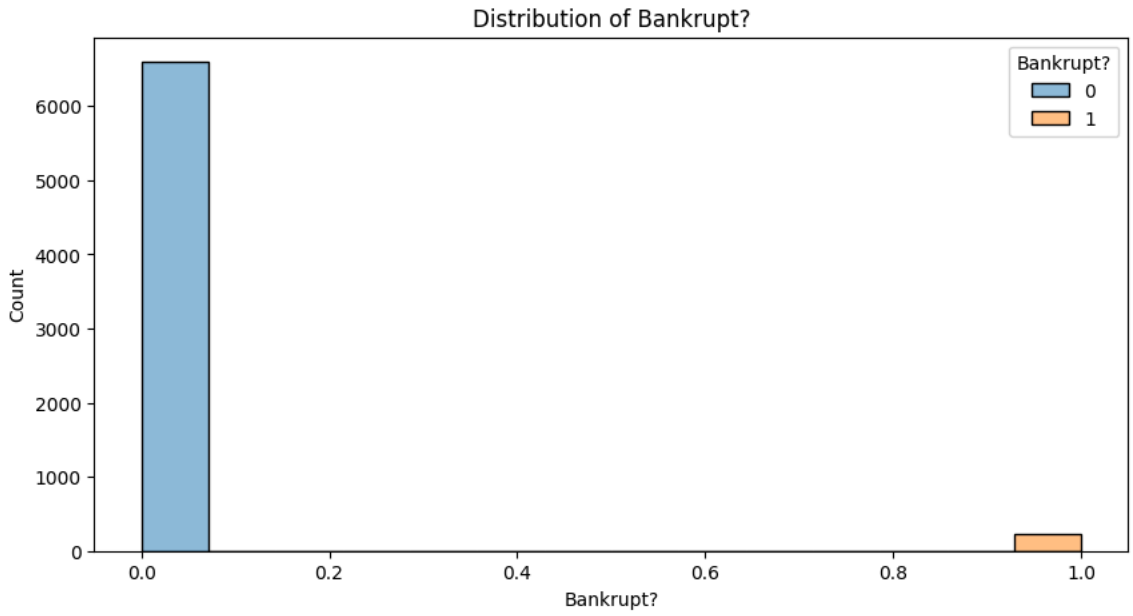
8 rows × 96 columns

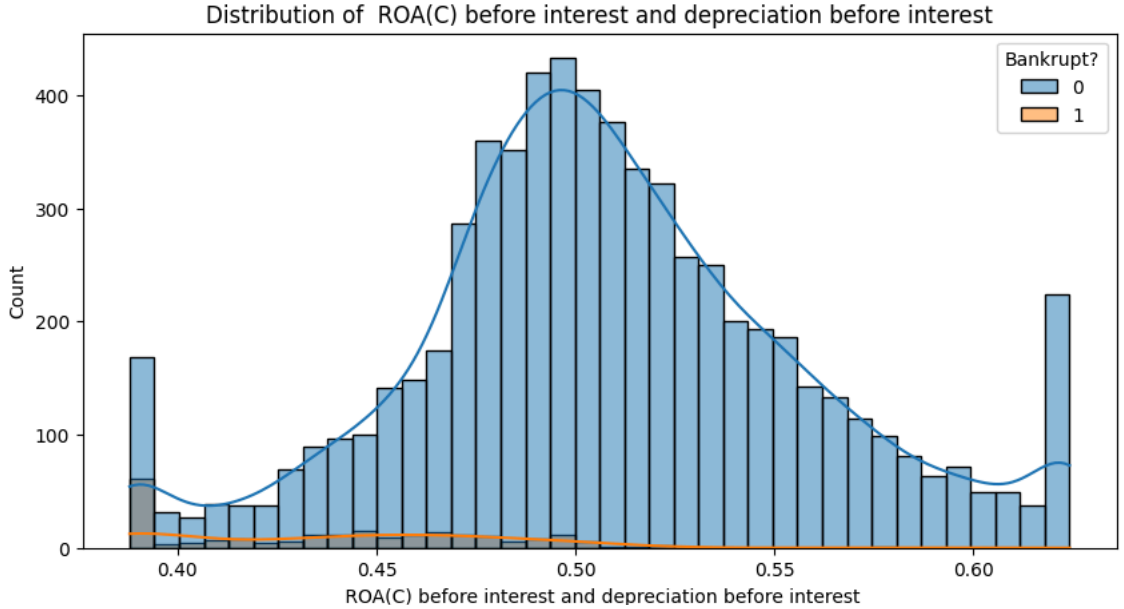
```
# Visualizing the distribution of the Target Variable
plt.figure(figsize=(8, 6))
sns.countplot(x='Bankrupt?', data=df, palette=palette)
palette = ["blue", "red"]
plt.xlabel('Bankrupt?')
plt.ylabel('Count')
plt.title('Distribution of Target Variable (Bankrupt?)')
```

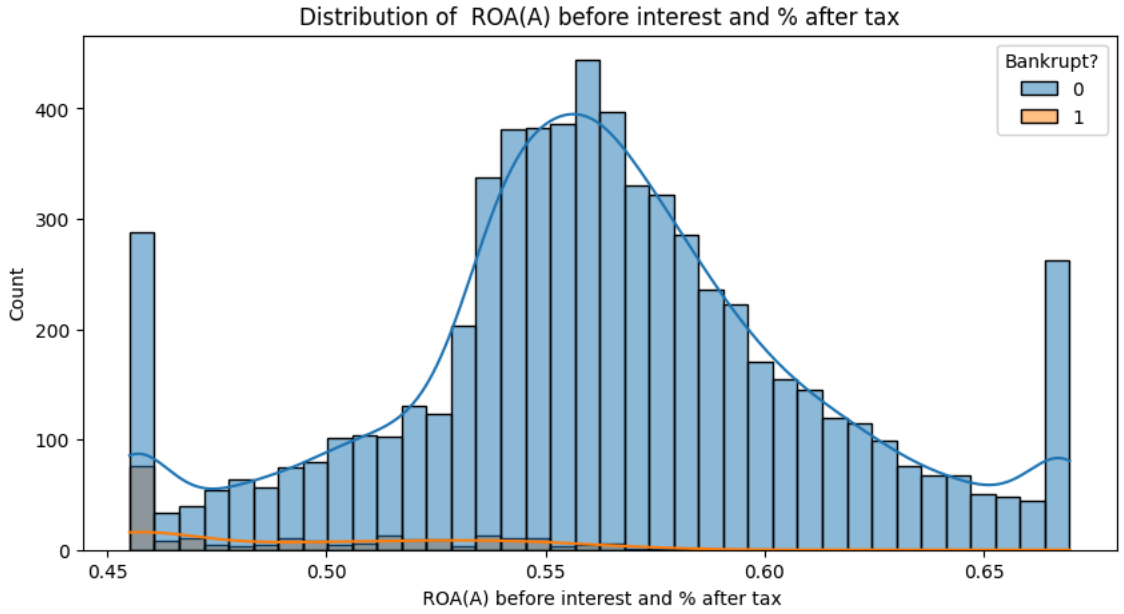
```
plt.show()

# Visualizing the distributions of a few selected features
selected_features = df.columns[:5]  # Replace with actual feature names or a subset of features
for feature in selected_features:
    plt.figure(figsize=(10, 5))
    sns.histplot(df, x=feature, hue='Bankrupt?', kde=True)
    plt.title(f'Distribution of {feature}')
    plt.show()

# Calculating the Correlation Matrix
corr_matrix = df.corr()

# Visualizing the Correlation Matrix
plt.figure(figsize=(25, 20))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='viridis')
plt.title('Correlation Matrix')
plt.xticks.top = False
plt.yticks.right = False
plt.show()

# Focus on correlations with the target variable
target_corr = corr_matrix['Bankrupt?'].sort_values(ascending=False)
print(target_corr)
```

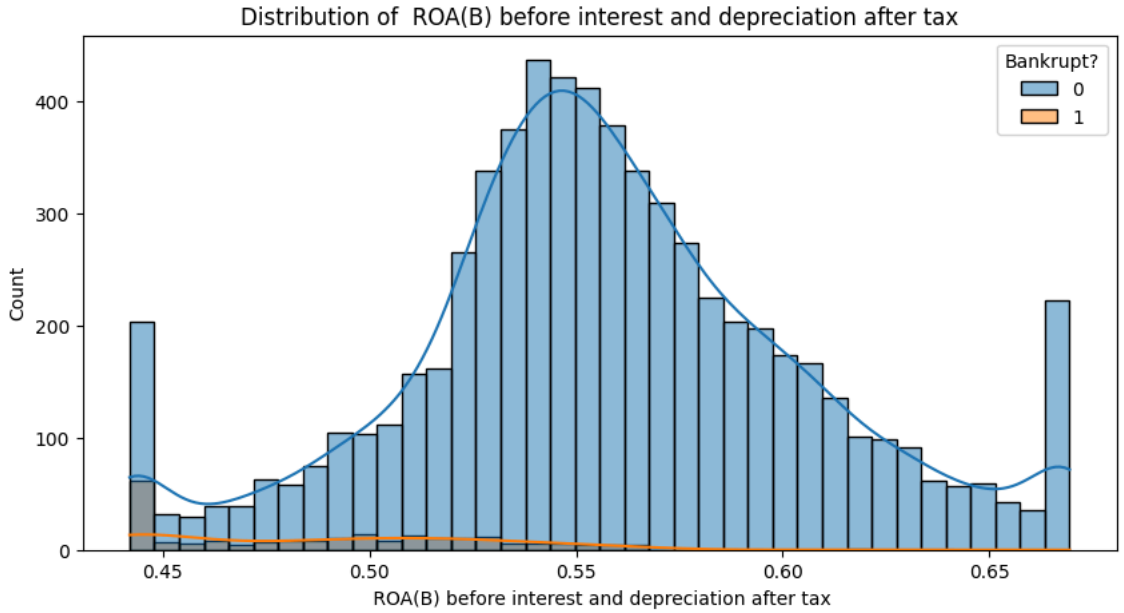
```
<ipython-input-76-2c8b1bb1e42e>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x='Bankrupt?', data=df, palette=palette)
```

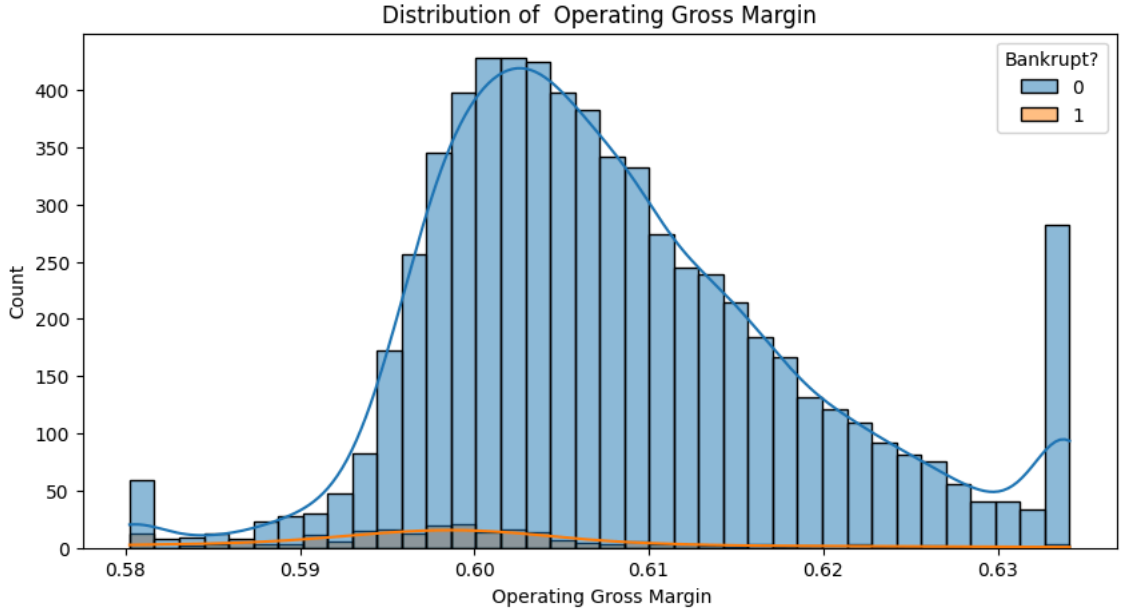


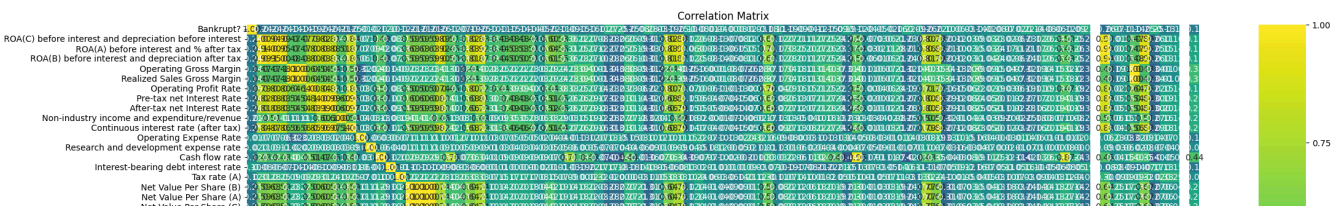


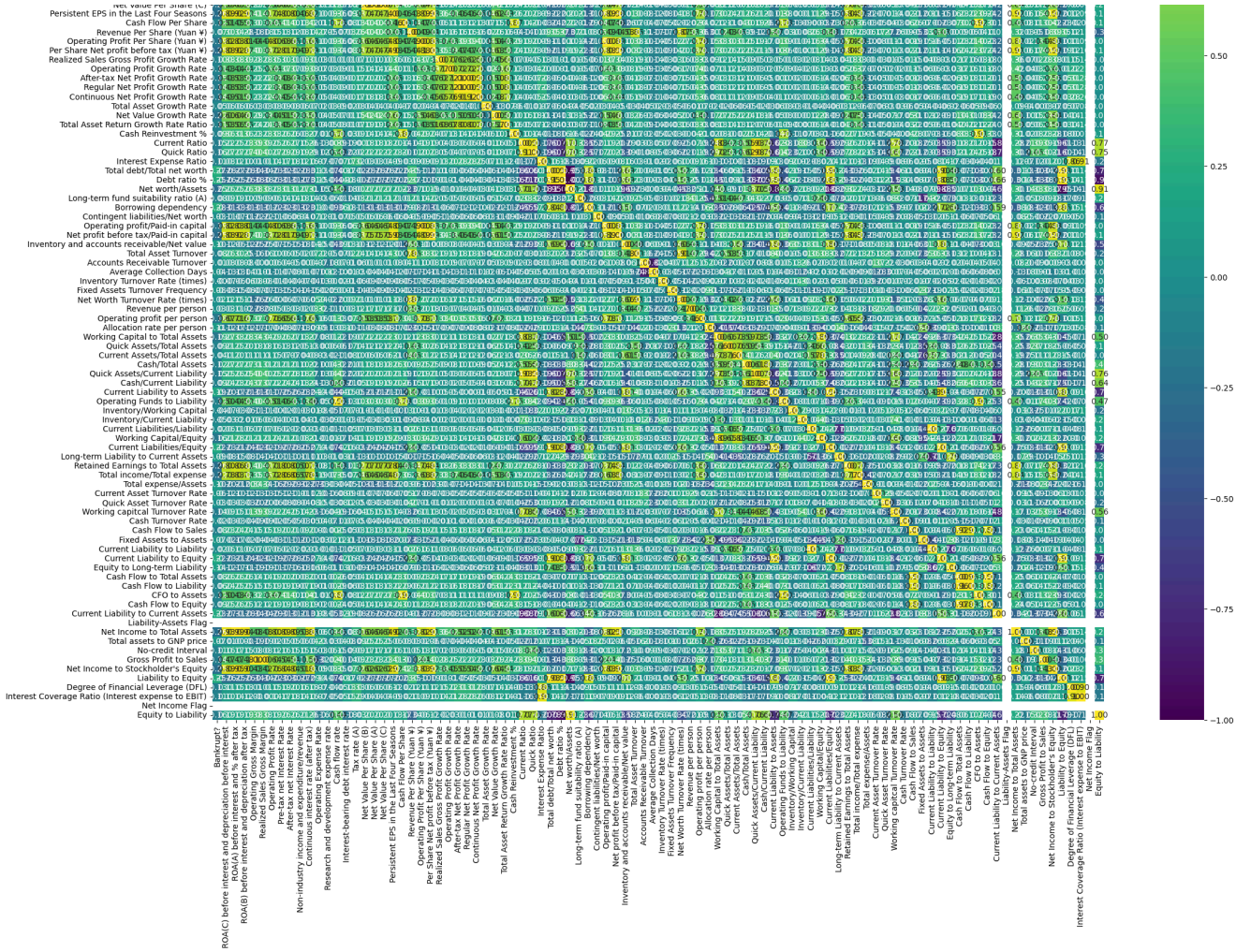



Distribution of ROA(A) before interest and % after tax
Bankrupt?
0
1
Count
ROA(A) before interest and % after tax


Distribution of ROA(B) before interest and depreciation after tax
Bankrupt?
0
1
Count
ROA(B) before interest and depreciation after tax


Distribution of Operating Gross Margin
Bankrupt?
0
1
Count
Operating Gross Margin


Correlation Matrix
Bankrupt?
ROA(C) before interest and depreciation before interest
ROA(A) before interest and % after tax
ROA(B) before interest and depreciation after tax
Operating Gross Margin
Realized Sales Gross Margin
Operating Profit Rate
Pre-tax net Interest Rate
After-tax net Interest Rate
Non-industry income and expenditure/revenue
Continuous interest rate (after tax)
Operating Expense Rate
Research and development expense rate
Cash flow rate
Interest-bearing debt interest rate
Tax rate (A)
Net Value Per Share (B)
Net Value Per Share (A)
1.00
0.75



```
Bankrupt?                                   1.000000
 Borrowing dependency                       0.278367
 Total debt/Total net worth                 0.272914
 Debt ratio %                               0.246535
 Liability to Equity                        0.246176
                                                ...
 Retained Earnings to Total Assets         -0.255218
 Net Income to Total Assets                -0.255797
 Persistent EPS in the Last Four Seasons   -0.256159
 Liability-Assets Flag                           NaN
 Net Income Flag                                 NaN
Name: Bankrupt?, Length: 96, dtype: float64
```

'''
Analysis:

- The heatmap of the correlation matrix shows that there are several features that have a strong positive or negative correlation with t
- The box plots show that there are outliers in some of the features. These outliers could potentially affect the results of any predict
- The distribution of the target variable shows that the data is imbalanced, with only a small percentage of companies being bankrupt. T
- The histograms show the distribution of the features for both bankrupt and non-bankrupt companies. These histograms can be used to ide

Recommendations:

- Further investigate the features that have a strong correlation with the target variable. These features could be used to develop a pr
- Consider using outlier detection and removal techniques to improve the accuracy of any predictive model that is trained on the data.
- Use oversampling or undersampling techniques to address the imbalance in the target variable.
- Use a variety of different feature selection and machine learning algorithms to find the best model for predicting bankruptcy.

Insights:

- The data provides valuable insights into the financial characteristics of bankrupt and non-bankrupt companies.
- The analysis of the data can be used to develop early warning systems for bankruptcy and to improve the accuracy of predictive models
'''

```
'\n*Analysis:*\n\n- The heatmap of the correlation matrix shows that there are several features that have a strong positive or nega
tive correlation with the target variable "Bankrupt?". These features could be potentially useful for predicting bankruptcy.\n- The
box plots show that there are outliers in some of the features. These outliers could potentially affect the results of any predicti
ve model that is trained on the data.\n- The distribution of the target variable shows that the data is imbalanced, with only a sma
ll percentage of companies being bankrupt. This could make it difficult to train a predictive model that is accurate for both bankr
upt and non-bankrupt companies.\n- The histograms show the distribution of the features for both bankrupt and non-bankrupt companie
s. These histograms can be used to identify features that are different between the two groups.\n\n*Recommendations:*\n\n- Further
investigate the features that have a strong correlation with the target variable. T '
```

```
# Setting benchmark for features correlation significance with target
corr_features = corr_matrix["Bankrupt?"].abs() >= 0.15
corr_features.value_counts()
```

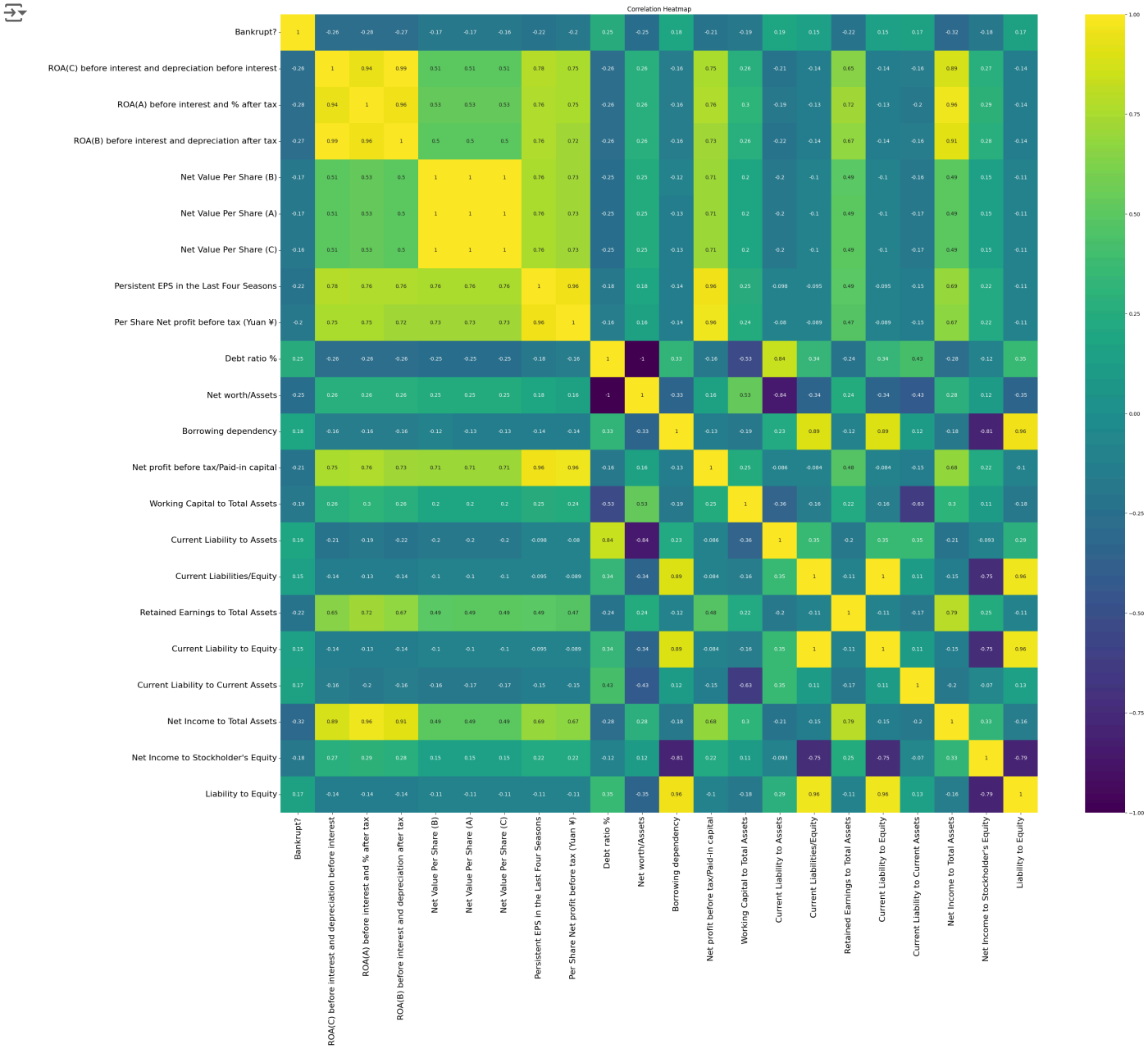
```
Bankrupt?
False    74
True     22
Name: count, dtype: int64
```

#Identification of features (variables) in the dataset have a significant correlation with the target variable Bankrupt?.

```
# Get List of Names for Features passing the benchmark
feature_names = list(corr_features[corr_features].index)
feature_names
```

```
['Bankrupt?',
 ' ROA(C) before interest and depreciation before interest',
 ' ROA(A) before interest and % after tax',
 ' ROA(B) before interest and depreciation after tax',
 ' Net Value Per Share (B)',
 ' Net Value Per Share (A)',
 ' Net Value Per Share (C)',
 ' Persistent EPS in the Last Four Seasons',
 ' Per Share Net profit before tax (Yuan ¥)',
 ' Debt ratio %',
 ' Net worth/Assets',
 ' Borrowing dependency',
 ' Net profit before tax/Paid-in capital',
 ' Working Capital to Total Assets',
 ' Current Liability to Assets',
 ' Current Liabilities/Equity',
 ' Retained Earnings to Total Assets',
 ' Current Liability to Equity',
 ' Current Liability to Current Assets',
 ' Net Income to Total Assets',
 " Net Income to Stockholder's Equity",
 ' Liability to Equity']
```

```
# Checking Correlations Between Selected Features

# Creating Correlation Matrix of Selected Features
mini_corr_matrix = df[feature_names].corr()

# Generating Heatmap
plt.figure(figsize=(33, 28))
sns.heatmap(mini_corr_matrix, annot = True, cmap="viridis")
plt.title("Correlation Heatmap")
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xticks.top = False
plt.yticks.right = False
plt.show()
```



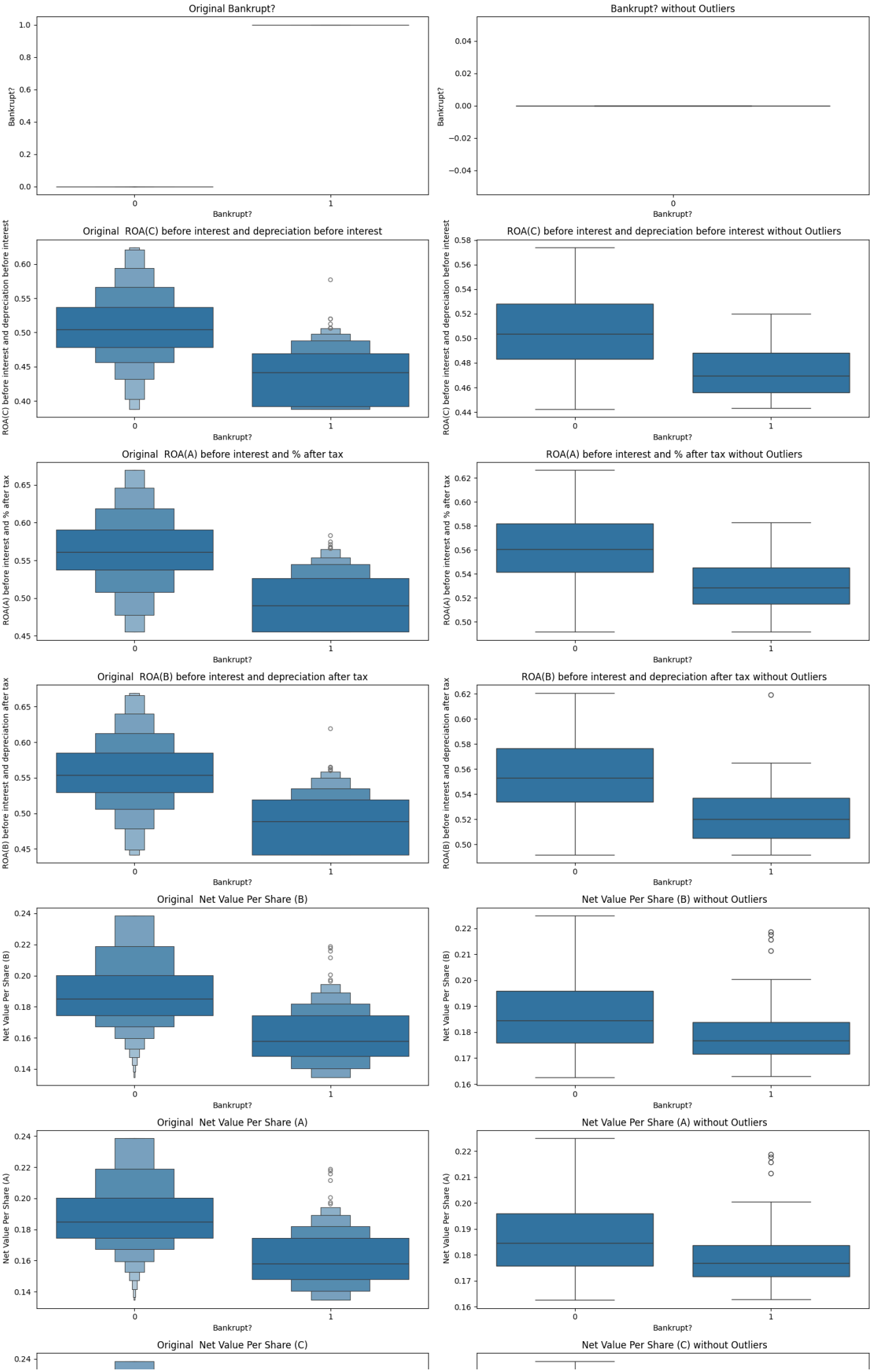
As, 0.15 is the benchmark for correlation significance, so only 22 attributes have at least weak correlation with the target. The rest attributes have correlation with target less than the benchmark, which means they have too weak or even no relation with the target.

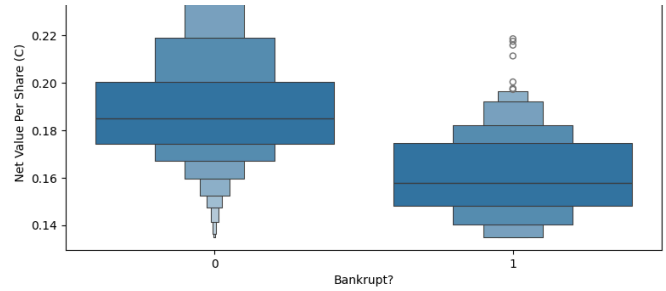
The choice based on correlation significance with the target only might not be the best choice. As correlation assumes linear relationship between features.

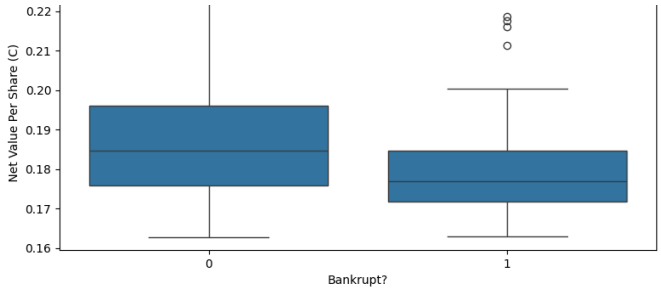
Start coding or generate with AI.

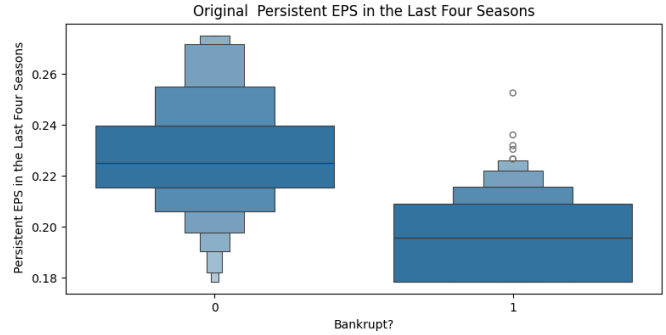
DETECTING OUTLIERS FROM df(DATAFRAME)

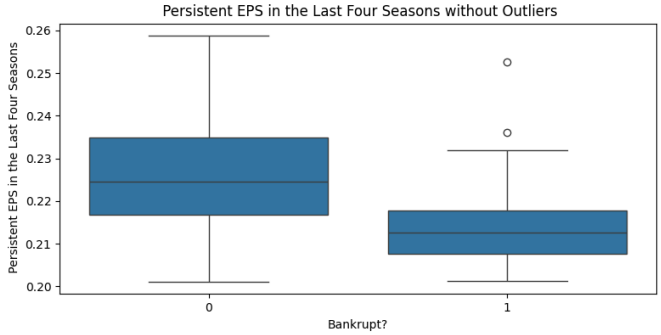
```
# Create a Figure with Subplots
fig, axes = plt.subplots(len(feature_names), 2, figsize=(16, len(feature_names) * 4))

for i, feature in enumerate(feature_names):
    # Plot with original feature
    sns.boxenplot(x="Bankrupt?", y=feature, data=df, ax=axes[i, 0])
    axes[i, 0].set_title(f'Original {feature}')

    # Plot after removing outliers in feature
    q1, q9 = df[feature].quantile([0.1, 0.9])
    mask = df[feature].between(q1, q9)
    sns.boxplot(x="Bankrupt?", y=feature, data=df[mask], ax=axes[i, 1])
    axes[i, 1].set_title(f'{feature} without Outliers')

# Adjust layout
plt.tight_layout()
plt.show()
```

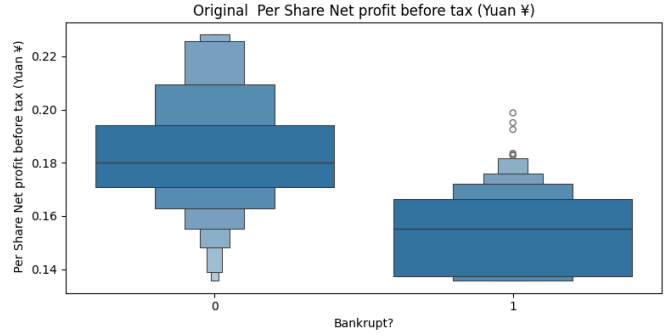

Original Bankrupt?
Bankrupt? without Outliers
Original ROA(C) before interest and depreciation before interest
ROA(C) before interest and depreciation before interest without Outliers
Original ROA(A) before interest and % after tax
ROA(A) before interest and % after tax without Outliers
Original ROA(B) before interest and depreciation after tax
ROA(B) before interest and depreciation after tax without Outliers
Original Net Value Per Share (B)
Net Value Per Share (B) without Outliers
Original Net Value Per Share (A)
Net Value Per Share (A) without Outliers
Original Net Value Per Share (C)
Net Value Per Share (C) without Outliers
Bankrupt?
ROA(C) before interest and depreciation before interest
ROA(A) before interest and % after tax
ROA(B) before interest and depreciation after tax
Net Value Per Share (B)
Net Value Per Share (A)

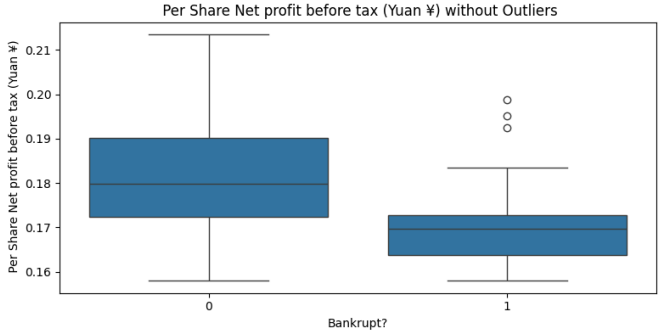

Net Value Per Share (C)
Bankrupt?

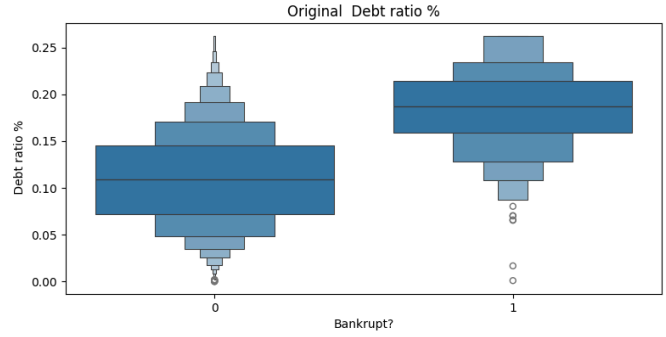

Net Value Per Share (C)
Bankrupt?

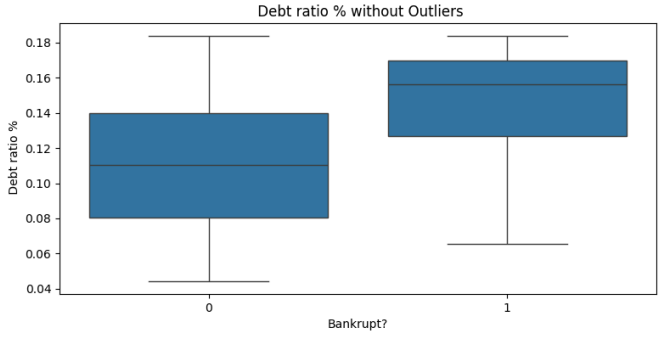

Original Persistent EPS in the Last Four Seasons
Persistent EPS in the Last Four Seasons
Bankrupt?

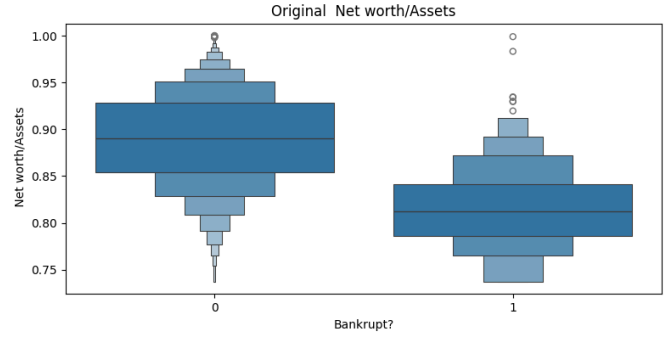

Persistent EPS in the Last Four Seasons without Outliers
Persistent EPS in the Last Four Seasons
Bankrupt?

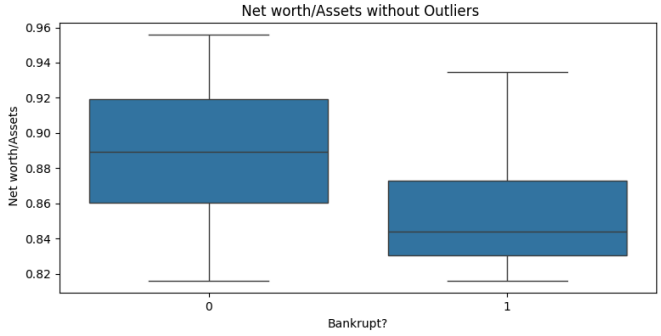

Original Per Share Net profit before tax (Yuan ¥)
Per Share Net profit before tax (Yuan ¥)
Bankrupt?

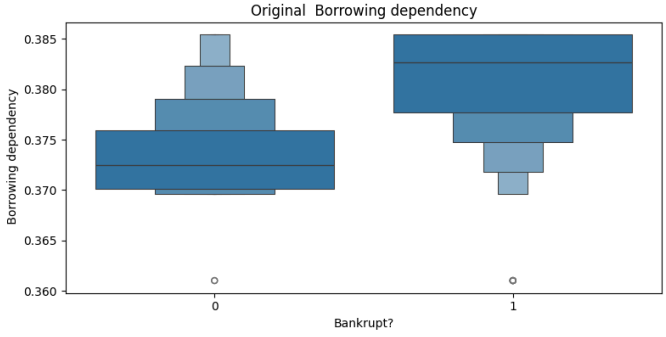

Per Share Net profit before tax (Yuan ¥) without Outliers
Per Share Net profit before tax (Yuan ¥)
Bankrupt?

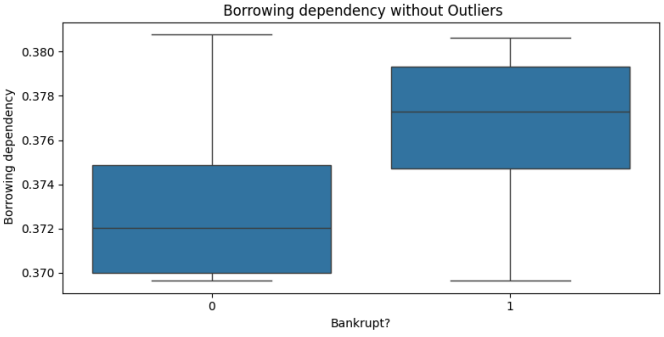

Original Debt ratio %
Debt ratio %
Bankrupt?

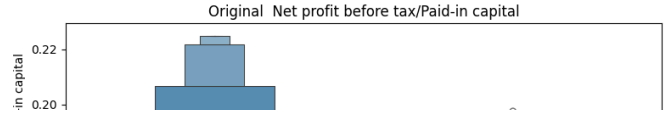

Debt ratio % without Outliers
Debt ratio %
Bankrupt?

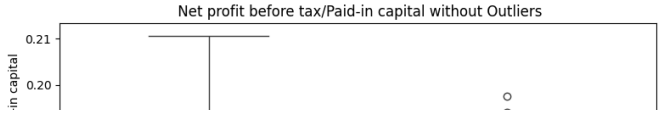

Original Net worth/Assets
Net worth/Assets
Bankrupt?

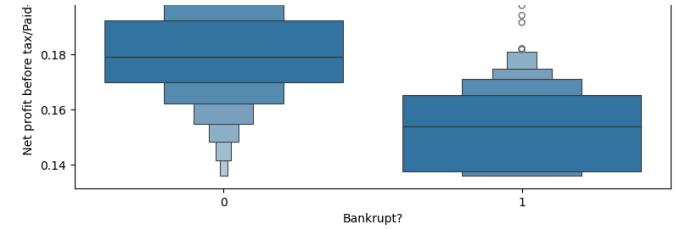

Net worth/Assets without Outliers
Net worth/Assets
Bankrupt?

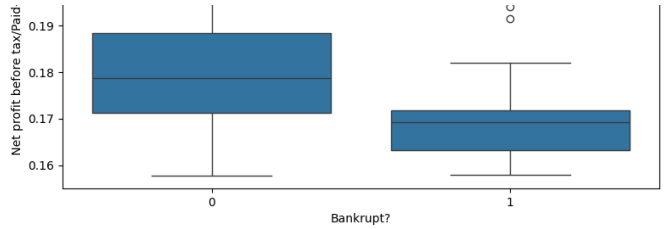

Original Borrowing dependency
Borrowing dependency
Bankrupt?

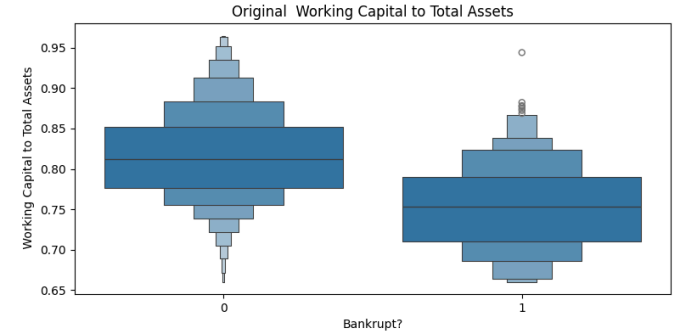

Borrowing dependency without Outliers
Borrowing dependency
Bankrupt?

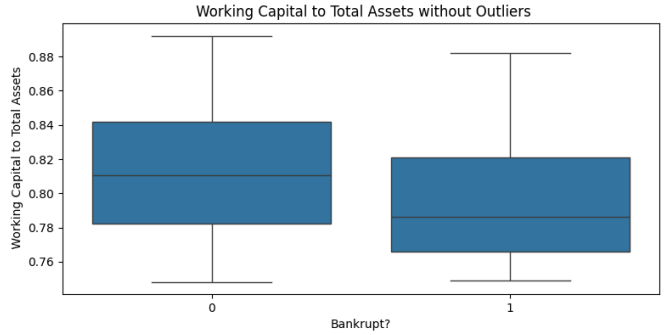

Original Net profit before tax/Paid-in capital

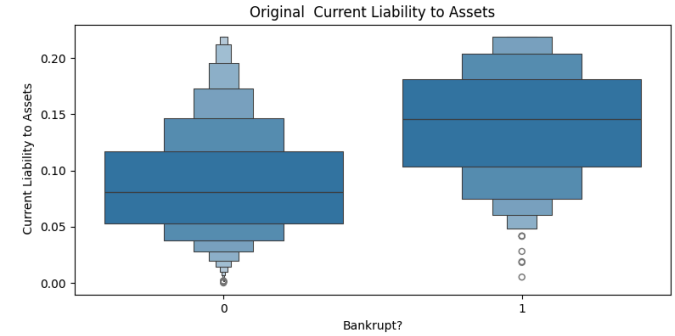

Net profit before tax/Paid-in capital without Outliers

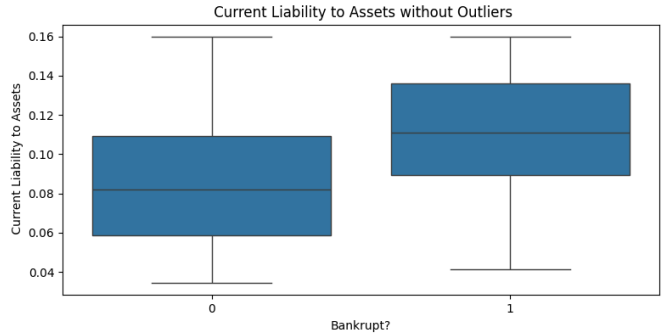

Net profit before tax/Paid
Bankrupt?

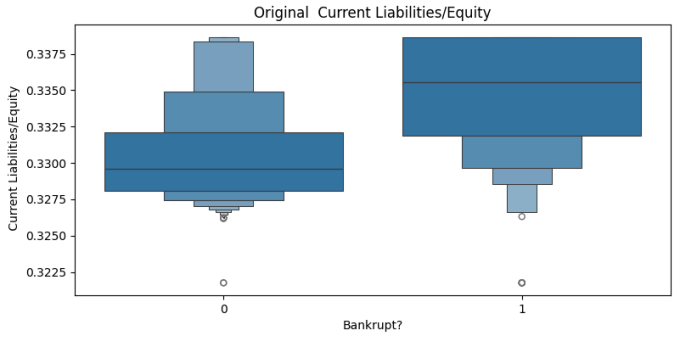

Net profit before tax/Paid
Bankrupt?

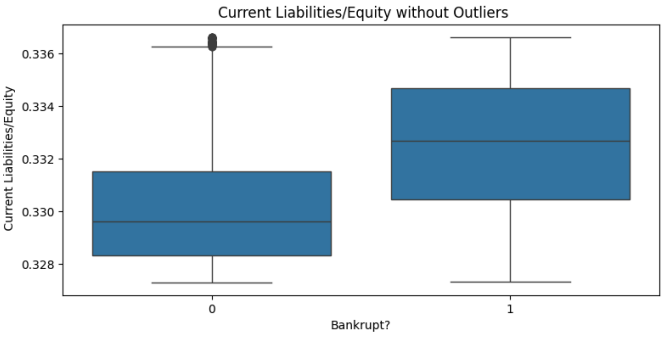

Original Working Capital to Total Assets
Working Capital to Total Assets
Bankrupt?

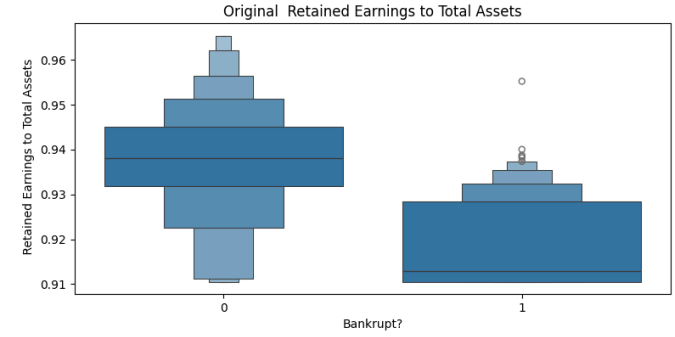

Working Capital to Total Assets without Outliers
Working Capital to Total Assets
Bankrupt?

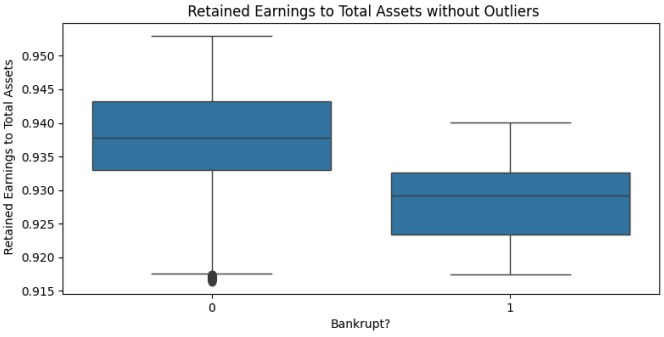

Original Current Liability to Assets
Current Liability to Assets
Bankrupt?

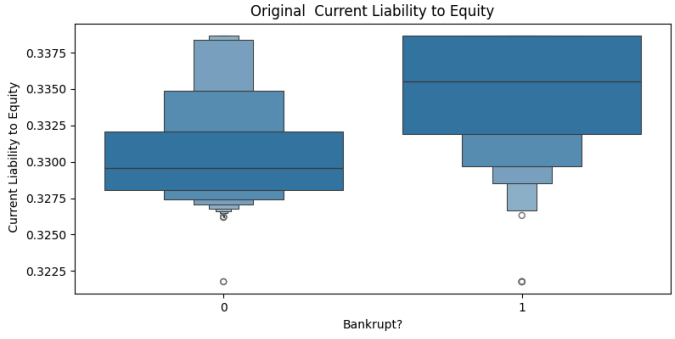

Current Liability to Assets without Outliers
Current Liability to Assets
Bankrupt?

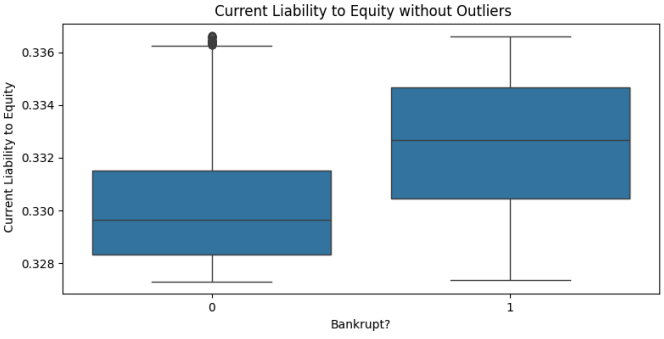

Original Current Liabilities/Equity
Current Liabilities/Equity
Bankrupt?

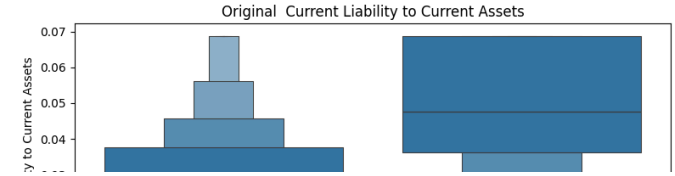

Current Liabilities/Equity without Outliers
Current Liabilities/Equity
Bankrupt?

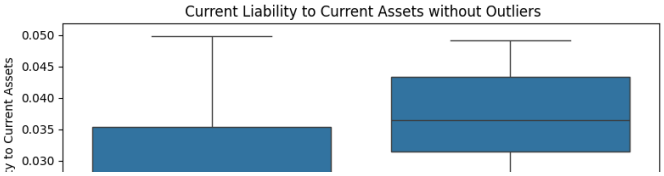

Original Retained Earnings to Total Assets
Retained Earnings to Total Assets
Bankrupt?

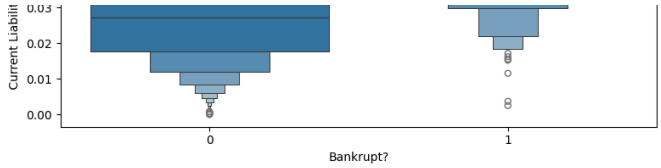

Retained Earnings to Total Assets without Outliers
Retained Earnings to Total Assets
Bankrupt?

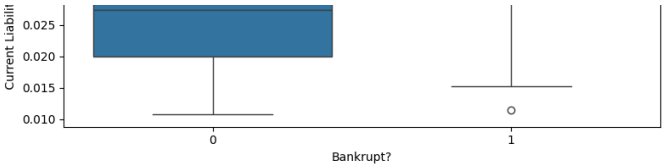

Original Current Liability to Equity
Current Liability to Equity
Bankrupt?

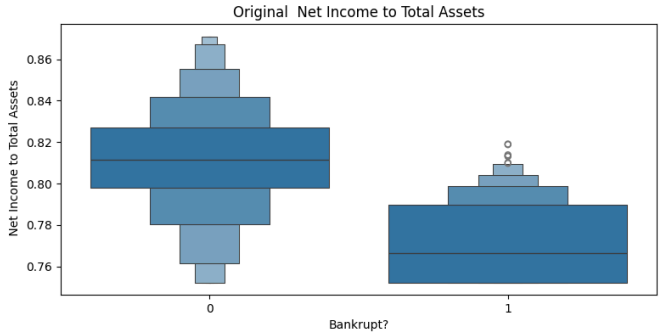

Current Liability to Equity without Outliers
Current Liability to Equity
Bankrupt?

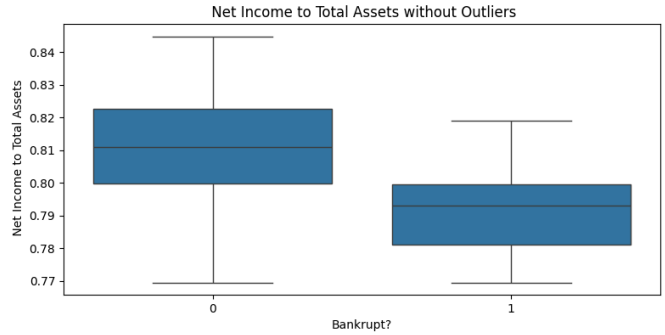

Original Current Liability to Current Assets

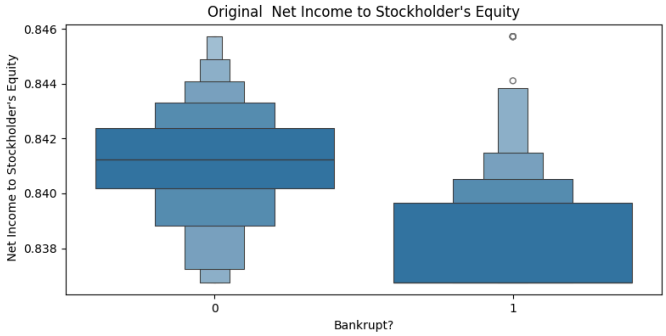

Current Liability to Current Assets without Outliers

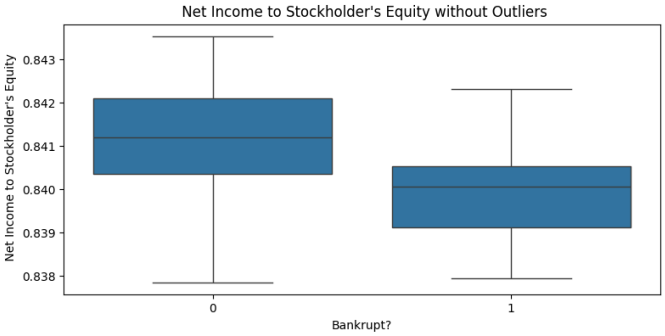

Current Liabilit
0.03
0.02
0.01
0.00
0
1
Bankrupt?

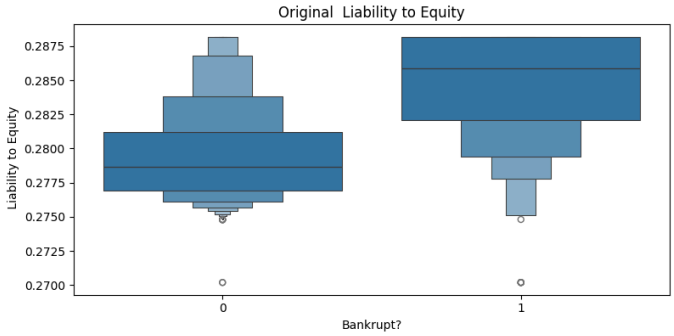

Current Liabilit
0.025
0.020
0.015
0.010
0
1
Bankrupt?

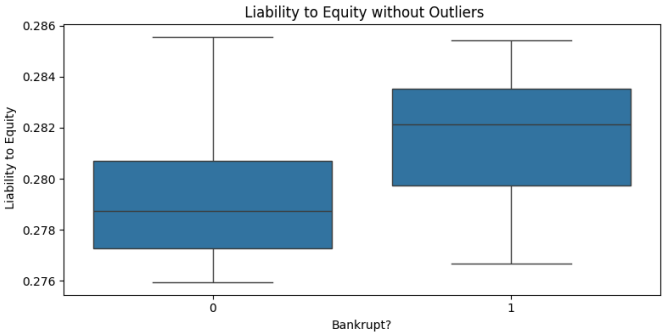

Original Net Income to Total Assets
Net Income to Total Assets
0.86
0.84
0.82
0.80
0.78
0.76
0
1
Bankrupt?


Net Income to Total Assets without Outliers
Net Income to Total Assets
0.84
0.83
0.82
0.81
0.80
0.79
0.78
0.77
0
1
Bankrupt?


Original Net Income to Stockholder's Equity
Net Income to Stockholder's Equity
0.846
0.844
0.842
0.840
0.838
0
1
Bankrupt?


Net Income to Stockholder's Equity without Outliers
Net Income to Stockholder's Equity
0.843
0.842
0.841
0.840
0.839
0.838
0
1
Bankrupt?


Original Liability to Equity
Liability to Equity
0.2875
0.2850
0.2825
0.2800
0.2775
0.2750
0.2725
0.2700
0
1
Bankrupt?

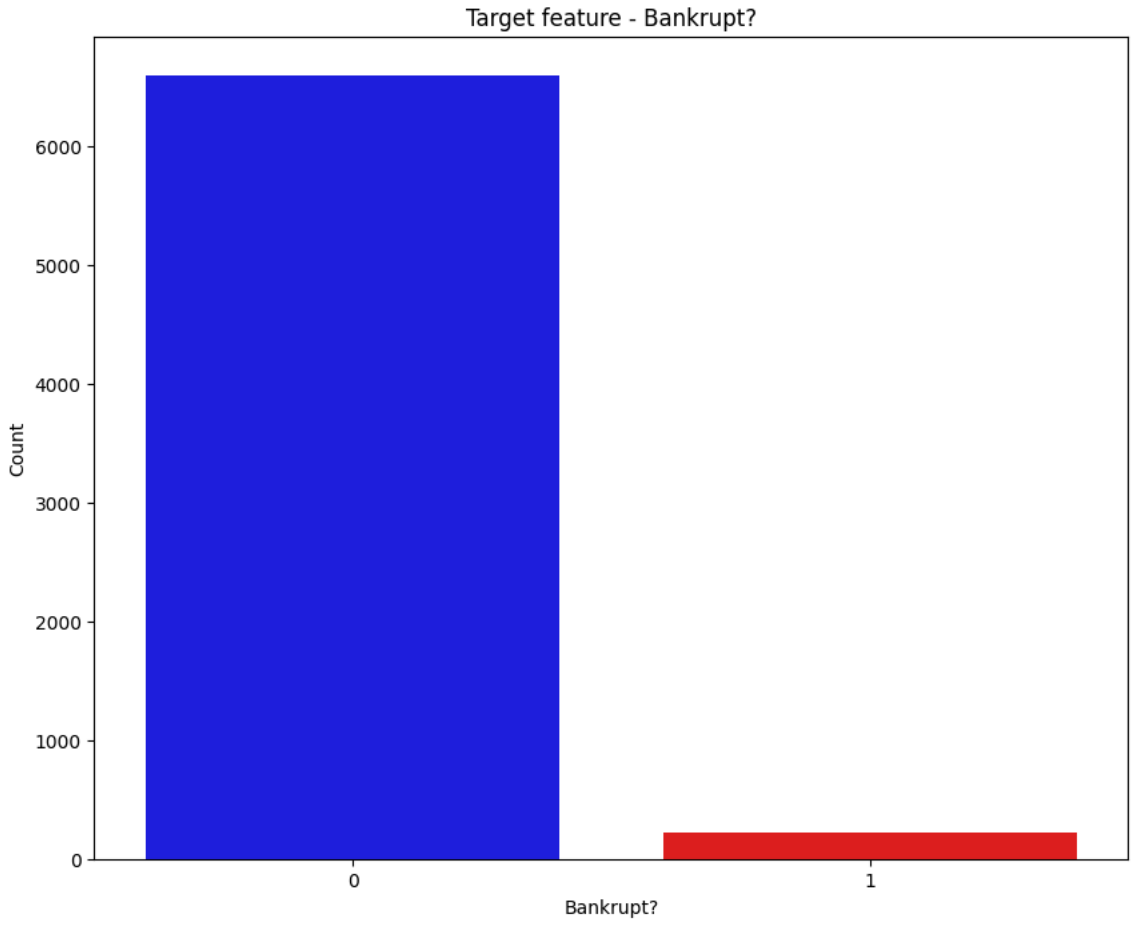

Liability to Equity without Outliers
Liability to Equity
0.286
0.284
0.282
0.280
0.278
0.276
0
1
Bankrupt?

```
'''
*Analysis:*

- The boxplots show that there are outliers in some of the features for both bankrupt and non-bankrupt companies.
- The outliers could potentially affect the results of any predictive model that is trained on the data.
- After removing the outliers, the boxplots show that the distributions of the features are more similar between bankrupt and non-bankrup

*Recommendations:*

- Consider using outlier detection and removal techniques to improve the accuracy of any predictive model that is trained on the data.
- Use a variety of different feature selection and machine learning algorithms to find the best model for predicting bankruptcy.

*Insights:*

- The data provides valuable insights into the financial characteristics of bankrupt and non-bankrupt companies.
- The analysis of the data can be used to develop early warning systems for bankruptcy and to improve the accuracy of predictive models f
'''
```

```
'\n*Analysis:*\n\n- The boxplots show that there are outliers in some of the features for both bankrupt and non-bankrupt companie
s.\n- The outliers could potentially affect the results of any predictive model that is trained on the data.\n- After removing the
outliers, the boxplots show that the distributions of the features are more similar between bankrupt and non-bankrupt companies.\n
\n*Recommendations:*\n\n- Consider using outlier detection and removal techniques to improve the accuracy of any predictive model t
hat is trained on the data.\n- Use a variety of different feature selection and machine learning algorithms to find the best model
for predicting bankruptcy.\n\n*Insights:*\n\n- The data provides valuable insights into the financial characteristics of bankrupt a
nd non-bankrupt companies.\n- The analysis of the data can be used to develop early warning systems for bankruptcy and to improve t
he accuracy of predictive models for bankruptcy.\n'
```

```
# Visualize the distribution of the Target Variable via Count Plot.
plt.figure(figsize=(10, 8))
sns.countplot(x='Bankrupt?', data=df, palette=palette)
palette = ["blue", "red"]
plt.xlabel('Bankrupt?')
plt.ylabel('Count')
plt.title('Target feature - Bankrupt?')
plt.show()
```

```
<ipython-input-74-41e1fcf5a3f5>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x='Bankrupt?', data=df, palette=palette)
```



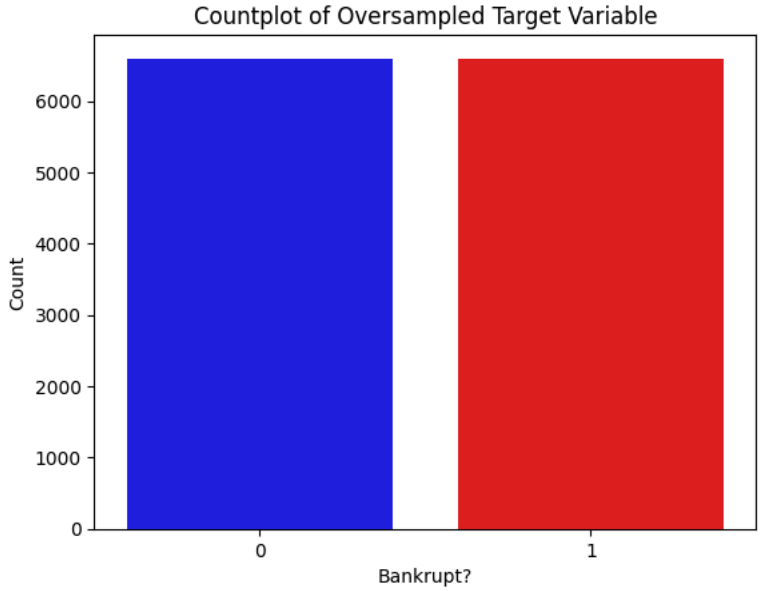
There is Imbalance in the Target Value, we need to oversample it....

```
##OVERSAMPLING THE TARGET VALUE##

from imblearn.over_sampling import SMOTE

X=df.drop(labels=['Bankrupt?'], axis=1)
y=df['Bankrupt?']
```

Oversampling the Target Variable using SMOTE

```
oversample = SMOTE()
X,y=oversample.fit_resample(X,y)
# Create the countplot with a custom color palette
palette = ["blue", "red"]
sns.countplot(x=y, palette=palette)  # Countplot based on the oversampled target variable (y)
plt.title("Countplot of Oversampled Target Variable")
plt.xlabel("Bankrupt?")
plt.ylabel("Count")
plt.show()
```

```
<ipython-input-75-e0dcb882f6f6>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x=y, palette=palette)  # Countplot based on the oversampled target variable (y)
```




```
# Separate the data into bankrupt and non-bankrupt
bankrupt = df[df['Bankrupt?'] == 1]
non_bankrupt = df[df['Bankrupt?'] == 0]
```

Hypothesis Testing:

```
# Perform t-tests
p_values = {}
for column in df.columns:
    if column != 'Bankrupt?':
        _, p_value = ttest_ind(bankrupt[column], non_bankrupt[column], equal_var=False)
        p_values[column] = p_value
```

```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_axis_nan_policy.py:523: RuntimeWarning: Precision loss occurred in moment calcu
  res = hypotest_fun_out(*samples, **kwds)
```

```
# Import the stats module from scipy
from scipy import stats

# Create a Contingency Table for Each Feature
contingency_tables = {}
for column in df.columns:
    if column != 'Bankrupt?':
        contingency_tables[column] = pd.crosstab(df['Bankrupt?'], df[column])

# Perform chi-square Tests
chi_square_results = {}
for column, table in contingency_tables.items():
    chi_square_results[column] = stats.chi2_contingency(table)

# Print the chi-square Statistics and p-values
for column, result in chi_square_results.items():
    print(f"*Feature:* {column}")
    print(f"Chi-square statistic: {result[0]:.4f}")
    print(f"P-value: {result[1]:.4f}")
    print()
```

```
P-value: 0.0000

*Feature:*  CFO to Assets
Chi-square statistic: 6257.4129
P-value: 0.9747

*Feature:*  Cash Flow to Equity
Chi-square statistic: 5855.8467
P-value: 0.8955

*Feature:*  Current Liability to Current Assets
Chi-square statistic: 5297.0521
P-value: 1.0000

*Feature:*  Liability-Assets Flag
Chi-square statistic: 0.0000
P-value: 1.0000

*Feature:*  Net Income to Total Assets
Chi-square statistic: 4601.6220
P-value: 1.0000

*Feature:*  Total assets to GNP price
Chi-square statistic: 5346.0774
P-value: 1.0000

*Feature:*  No-credit Interval
Chi-square statistic: 4495.2055
P-value: 1.0000

*Feature:*  Gross Profit to Sales
Chi-square statistic: 6451.5692
P-value: 0.6627

*Feature:*  Net Income to Stockholder's Equity
Chi-square statistic: 3925.2115
P-value: 1.0000

*Feature:*  Liability to Equity
Chi-square statistic: 4803.9360
P-value: 1.0000

*Feature:*  Degree of Financial Leverage (DFL)
Chi-square statistic: 3459.5523
P-value: 1.0000

*Feature:*  Interest Coverage Ratio (Interest expense to EBIT)
Chi-square statistic: 3113.3155
P-value: 1.0000

*Feature:*  Net Income Flag
Chi-square statistic: 0.0000
P-value: 1.0000

*Feature:*  Equity to Liability
Chi-square statistic: 6755.1758
P-value: 0.0000
```

ANALYSING THE RESULT OF t-TEST & chi-SQUARE TEST

```
# T-tests
significant_t_tests = [column for column, p_value in p_values.items() if p_value < 0.05]
print(f"There are {len(significant_t_tests)} features with significant differences between bankrupt and non-bankrupt companies based on
for column in significant_t_tests:
    print(f"\t- {column}")

# Chi-square tests
significant_chi_square_tests = [column for column, result in chi_square_results.items() if result[1] < 0.05]
print(f"\nThere are {len(significant_chi_square_tests)} features with significant differences between bankrupt and non-bankrupt companie
for column in significant_chi_square_tests:
    print(f"\t- {column}")

# Compare the results of the two tests
common_features = set(significant_t_tests) & set(significant_chi_square_tests)
print(f"\nThere are {len(common_features)} features that are significant in both t-tests and chi-square tests:")
for column in common_features:
    print(f"\t- {column}")
```

```
        -  Borrowing dependency
        -  Contingent liabilities/Net worth
        -  Operating profit/Paid-in capital
        -  Net profit before tax/Paid-in capital
        -  Total Asset Turnover
        -  Operating profit per person
        -  Cash/Total Assets
        -  Quick Assets/Current Liability
        -  Cash/Current Liability
        -  Operating Funds to Liability
        -  Inventory/Current Liability
        -  Long-term Liability to Current Assets
        -  Equity to Long-term Liability
        -  Cash Flow to Total Assets
        -  Cash Flow to Liability
        -  Equity to Liability

There are 40 features that are significant in both t-tests and chi-square tests:
        -  Equity to Long-term Liability
        -  Cash Flow to Liability
        -  Cash Flow Per Share
        -  Net Value Per Share (C)
        -  Operating Funds to Liability
        -  Total Asset Return Growth Rate Ratio
        -  Cash/Current Liability
        -  Realized Sales Gross Margin
        -  Cash/Total Assets
        -  Net Value Per Share (A)
        -  Net profit before tax/Paid-in capital
        -  Long-term Liability to Current Assets
        -  Per Share Net profit before tax (Yuan ¥)
        -  Current Ratio
        -  Operating profit per person
        -  Net Value Per Share (B)
        -  Total Asset Growth Rate
        -  Debt ratio %
        -  ROA(A) before interest and % after tax
        -  Net worth/Assets
        -  Quick Assets/Current Liability
        -  Interest-bearing debt interest rate
        -  Operating Profit Per Share (Yuan ¥)
        -  Inventory/Current Liability
        -  Operating Gross Margin
        -  Net Value Growth Rate
        -  Operating profit/Paid-in capital
        -  Borrowing dependency
        -  Contingent liabilities/Net worth
        -  Operating Profit Rate
        -  Total Asset Turnover
        -  ROA(C) before interest and depreciation before interest
        -  Quick Ratio
        -  Non-industry income and expenditure/revenue
        -  Long-term fund suitability ratio (A)
        -  Equity to Liability
        -  Interest Expense Ratio
        -  ROA(B) before interest and depreciation after tax
        -  Cash Flow to Total Assets
        -  Persistent EPS in the Last Four Seasons
```

```
# T-tests
for column, p_value in p_values.items():
    if p_value < 0.05:
        print("----------------------------------   As Per T - Tests.")
        print(f"*Feature:* {column}")
        print("Null hypothesis: There is no significant difference between the means of bankrupt and non-bankrupt companies.")
        print("Alternative hypothesis: There is a significant difference between the means of bankrupt and non-bankrupt companies.")
        print("Result: Reject the null hypothesis.")
        print("Conclusion: There is a significant difference between the means of bankrupt and non-bankrupt companies for the feature {c
        print()
    else:
        print("----------------------------------   As Per T - Tests.")
        print(f"*Feature:* {column}")
        print("Null hypothesis: There is no significant difference between the means of bankrupt and non-bankrupt companies.")
        print("Alternative hypothesis: There is a significant difference between the means of bankrupt and non-bankrupt companies.")
        print("Result: Do not reject the null hypothesis.")
        print("Conclusion: There is no significant difference between the means of bankrupt and non-bankrupt companies for the feature {
        print()

# Chi-square tests
for column, result in chi_square_results.items():
    if result[1] < 0.05:
        print("----------------------------------   As Per Chi Square Tests.")
        print(f"*Feature:* {column}")
        print("Null hypothesis: There is no association between the feature and the target variable.")
        print("Alternative hypothesis: There is an association between the feature and the target variable.")
        print("Result: Reject the null hypothesis.")
        print("Conclusion: There is an association between the feature {column} and the target variable.")
        print()
    else:
        print("----------------------------------   As Per Chi Square Tests.")
        print(f"*Feature:* {column}")
        print("Null hypothesis: There is no association between the feature and the target variable.")
        print("Alternative hypothesis: There is an association between the feature and the target variable.")
        print("Result: Do not reject the null hypothesis.")
        print("Conclusion: There is no association between the feature {column} and the target variable.")
        print()
```

```
---------------------------------   As Per Chi Square Tests.
*Feature:*  Equity to Liability
Null hypothesis: There is no association between the feature and the target variable.
Alternative hypothesis: There is an association between the feature and the target variable.
Result: Reject the null hypothesis.
Conclusion: There is an association between the feature {column} and the target variable.
```

FEATURE ENGINEERING AND SELECTION:

```
# Select significant features
significant_features = [k for k, v in p_values.items() if v < 0.05]
print("Significant Features:")
significant_features
```

```
' Net Value Growth Rate',
' Total Asset Return Growth Rate Ratio',
' Cash Reinvestment %',
' Current Ratio',
' Quick Ratio',
' Interest Expense Ratio',
' Total debt/Total net worth',
' Debt ratio %',
' Net worth/Assets',
' Long-term fund suitability ratio (A)',
' Borrowing dependency',
' Contingent liabilities/Net worth',
' Operating profit/Paid-in capital',
' Net profit before tax/Paid-in capital',
' Inventory and accounts receivable/Net value',
' Total Asset Turnover',
' Average Collection Days',
' Fixed Assets Turnover Frequency',
' Revenue per person',
' Operating profit per person',
' Allocation rate per person',
' Working Capital to Total Assets',
' Quick Assets/Total Assets',
' Current Assets/Total Assets',
' Cash/Total Assets',
' Quick Assets/Current Liability',
' Cash/Current Liability',
' Current Liability to Assets',
' Operating Funds to Liability',
' Inventory/Working Capital',
' Inventory/Current Liability',
' Working Capital/Equity',
' Current Liabilities/Equity',
' Long-term Liability to Current Assets',
' Retained Earnings to Total Assets',
' Total income/Total expense',
' Total expense/Assets',
' Current Asset Turnover Rate',
' Quick Asset Turnover Rate',
' Working capitcal Turnover Rate',
' Cash Flow to Sales',
' Fixed Assets to Assets',
' Current Liability to Equity',
' Equity to Long-term Liability',
' Cash Flow to Total Assets',
' Cash Flow to Liability',
' CFO to Assets',
' Cash Flow to Equity',
' Current Liability to Current Assets',
' Net Income to Total Assets',
' Total assets to GNP price',
' No-credit Interval',
' Gross Profit to Sales',
" Net Income to Stockholder's Equity",
' Liability to Equity',
' Degree of Financial Leverage (DFL)',
' Interest Coverage Ratio (Interest expense to EBIT)',
' Equity to Liability']
```

```
# Create polynomial features
poly = PolynomialFeatures(degree=2, interaction_only=True, include_bias=False)
X_poly = poly.fit_transform(df[significant_features])
X_poly_df = pd.DataFrame(X_poly, columns=poly.get_feature_names_out(significant_features))
```

```
# Import the StandardScaler class
from sklearn.preprocessing import StandardScaler

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_poly_df)


# Dimensionality reduction with PCA
pca = PCA(n_components=30)  # Adjust the number of components as needed
X_pca = pca.fit_transform(X_scaled)


# Feature selection with RFE
log_reg = LogisticRegression(max_iter=1000)
rfe = RFE(log_reg, n_features_to_select=30)  # Adjust the number of features as needed
X_rfe = rfe.fit_transform(X_pca, df['Bankrupt?'])
```

MODELLING:

```
# Handle class imbalance using SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_rfe, df['Bankrupt?'])


# Split the data
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42)


# Hyperparameter tuning for Logistic Regression
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2', 'elasticnet', 'none'],
    'solver': ['lbfgs', 'liblinear', 'saga']
}


!pip install scikit-learn
import sklearn.model_selection
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
grid_search = GridSearchCV(LogisticRegression(max_iter=1000), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
best_log_reg = grid_search.best_estimator_
```