sm-240340325074

June 17, 2024

[161]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import chi2_contingency
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, precision_score, recall_score,␣
↪accuracy_score, f1_score
```

[47]:
```python
df=pd.read_csv(r"D:\DBDA\plib\EDA\bankrupt\data.csv")
df.head()
```

[47]:
```
   Bankrupt?  ROA(C) before interest and depreciation before interest  \
0          1                                           0.370594
1          1                                           0.464291
2          1                                           0.426071
3          1                                           0.399844
4          1                                           0.465022

   ROA(A) before interest and % after tax  \
0                                0.424389
1                                0.538214
2                                0.499019
3                                0.451265
4                                0.538432

   ROA(B) before interest and depreciation after tax  Operating Gross Margin  \
0                                           0.405750                0.601457
1                                           0.516730                0.610235
2                                           0.472295                0.601450
3                                           0.457733                0.583541
4                                           0.522298                0.598783

   Realized Sales Gross Margin  Operating Profit Rate  \
0                     0.601457               0.998969
1                     0.610235               0.998946
```

```
2                    0.601364                 0.998857
3                    0.583541                 0.998700
4                    0.598783                 0.998973

   Pre-tax net Interest Rate  After-tax net Interest Rate  \
0                   0.796887                     0.808809
1                   0.797380                     0.809301
2                   0.796403                     0.808388
3                   0.796967                     0.808966
4                   0.797366                     0.809304

   Non-industry income and expenditure/revenue  …  \
0                                     0.302646  …
1                                     0.303556  …
2                                     0.302035  …
3                                     0.303350  …
4                                     0.303475  …

   Net Income to Total Assets  Total assets to GNP price  No-credit Interval  \
0                    0.716845                   0.009219            0.622879
1                    0.795297                   0.008323            0.623652
2                    0.774670                   0.040003            0.623841
3                    0.739555                   0.003252            0.622929
4                    0.795016                   0.003878            0.623521

   Gross Profit to Sales  Net Income to Stockholder's Equity  \
0               0.601453                            0.827890
1               0.610237                            0.839969
2               0.601449                            0.836774
3               0.583538                            0.834697
4               0.598782                            0.839973

   Liability to Equity  Degree of Financial Leverage (DFL)  \
0             0.290202                            0.026601
1             0.283846                            0.264577
2             0.290189                            0.026555
3             0.281721                            0.026697
4             0.278514                            0.024752

   Interest Coverage Ratio (Interest expense to EBIT)  Net Income Flag  \
0                                           0.564050                1
1                                           0.570175                1
2                                           0.563706                1
3                                           0.564663                1
4                                           0.575617                1

   Equity to Liability
```

```
0           0.016469
1           0.020794
2           0.016474
3           0.023982
4           0.035490

[5 rows x 96 columns]
```

[48]:
```
df.describe()
```

[48]:
```
         Bankrupt?  ROA(C) before interest and depreciation before interest  \
count  6819.000000                                        6819.000000
mean      0.032263                                           0.505180
std       0.176710                                           0.060686
min       0.000000                                           0.000000
25%       0.000000                                           0.476527
50%       0.000000                                           0.502706
75%       0.000000                                           0.535563
max       1.000000                                           1.000000

       ROA(A) before interest and % after tax  \
count                             6819.000000
mean                                 0.558625
std                                  0.065620
min                                  0.000000
25%                                  0.535543
50%                                  0.559802
75%                                  0.589157
max                                  1.000000

       ROA(B) before interest and depreciation after tax  \
count                                        6819.000000
mean                                            0.553589
std                                             0.061595
min                                             0.000000
25%                                             0.527277
50%                                             0.552278
75%                                             0.584105
max                                             1.000000

       Operating Gross Margin  Realized Sales Gross Margin  \
count             6819.000000                  6819.000000
mean                 0.607948                     0.607929
std                  0.016934                     0.016916
min                  0.000000                     0.000000
25%                  0.600445                     0.600434
50%                  0.605997                     0.605976
```

75%	0.613914	0.613842
max	1.000000	1.000000

	Operating Profit Rate	Pre-tax net Interest Rate \
count	6819.000000	6819.000000
mean	0.998755	0.797190
std	0.013010	0.012869
min	0.000000	0.000000
25%	0.998969	0.797386
50%	0.999022	0.797464
75%	0.999095	0.797579
max	1.000000	1.000000

	After-tax net Interest Rate \
count	6819.000000
mean	0.809084
std	0.013601
min	0.000000
25%	0.809312
50%	0.809375
75%	0.809469
max	1.000000

	Non-industry income and expenditure/revenue	… \
count	6819.000000	…
mean	0.303623	…
std	0.011163	…
min	0.000000	…
25%	0.303466	…
50%	0.303525	…
75%	0.303585	…
max	1.000000	…

	Net Income to Total Assets	Total assets to GNP price \
count	6819.000000	6.819000e+03
mean	0.807760	1.862942e+07
std	0.040332	3.764501e+08
min	0.000000	0.000000e+00
25%	0.796750	9.036205e-04
50%	0.810619	2.085213e-03
75%	0.826455	5.269777e-03
max	1.000000	9.820000e+09

	No-credit Interval	Gross Profit to Sales \
count	6819.000000	6819.000000
mean	0.623915	0.607946
std	0.012290	0.016934

```
min             0.000000              0.000000
25%             0.623636              0.600443
50%             0.623879              0.605998
75%             0.624168              0.613913
max             1.000000              1.000000

       Net Income to Stockholder's Equity  Liability to Equity  \
count                         6819.000000          6819.000000
mean                             0.840402             0.280365
std                              0.014523             0.014463
min                              0.000000             0.000000
25%                              0.840115             0.276944
50%                              0.841179             0.278778
75%                              0.842357             0.281449
max                              1.000000             1.000000

       Degree of Financial Leverage (DFL)  \
count                         6819.000000
mean                             0.027541
std                              0.015668
min                              0.000000
25%                              0.026791
50%                              0.026808
75%                              0.026913
max                              1.000000

       Interest Coverage Ratio (Interest expense to EBIT)  Net Income Flag  \
count                                        6819.000000           6819.0
mean                                            0.565358              1.0
std                                             0.013214              0.0
min                                             0.000000              1.0
25%                                             0.565158              1.0
50%                                             0.565252              1.0
75%                                             0.565725              1.0
max                                             1.000000              1.0

       Equity to Liability
count          6819.000000
mean              0.047578
std               0.050014
min               0.000000
25%               0.024477
50%               0.033798
75%               0.052838
max               1.000000

[8 rows x 96 columns]
```

```
[49]: df.isnull().sum()
```

```
[49]: 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
```

```
[50]: def identify_and_impute_outliers(df, response_column):
          for column in df.columns:
              if column == response_column:
                  continue
              if df[column].dtype in ['int64', 'float64']:  # Only consider numeric␣
       ↪columns
                  Q1 = df[column].quantile(0.25)
                  Q3 = df[column].quantile(0.75)
                  IQR = Q3 - Q1

                  lower_bound = Q1 - 1.5 * IQR
                  upper_bound = Q3 + 1.5 * IQR
                  outliers = df[(df[column] < lower_bound) | (df[column] >␣
       ↪upper_bound)]
                  print(f"Column: {column}, Outliers Count: {len(outliers)}")
                  median = df[column].median()
                  df.loc[(df[column] < lower_bound) | (df[column] > upper_bound),␣
       ↪column] = median

          return df
      response_column = 'Bankrupt?'
      df = identify_and_impute_outliers(df, response_column)
```

```
Column: ROA(C) before interest and depreciation before interest, Outliers Count:
391
Column: ROA(A) before interest and % after tax, Outliers Count: 561
Column: ROA(B) before interest and depreciation after tax, Outliers Count: 432
Column: Operating Gross Margin, Outliers Count: 320
Column: Realized Sales Gross Margin, Outliers Count: 318
Column: Operating Profit Rate, Outliers Count: 716
Column: Pre-tax net Interest Rate, Outliers Count: 773
Column: After-tax net Interest Rate, Outliers Count: 867
```

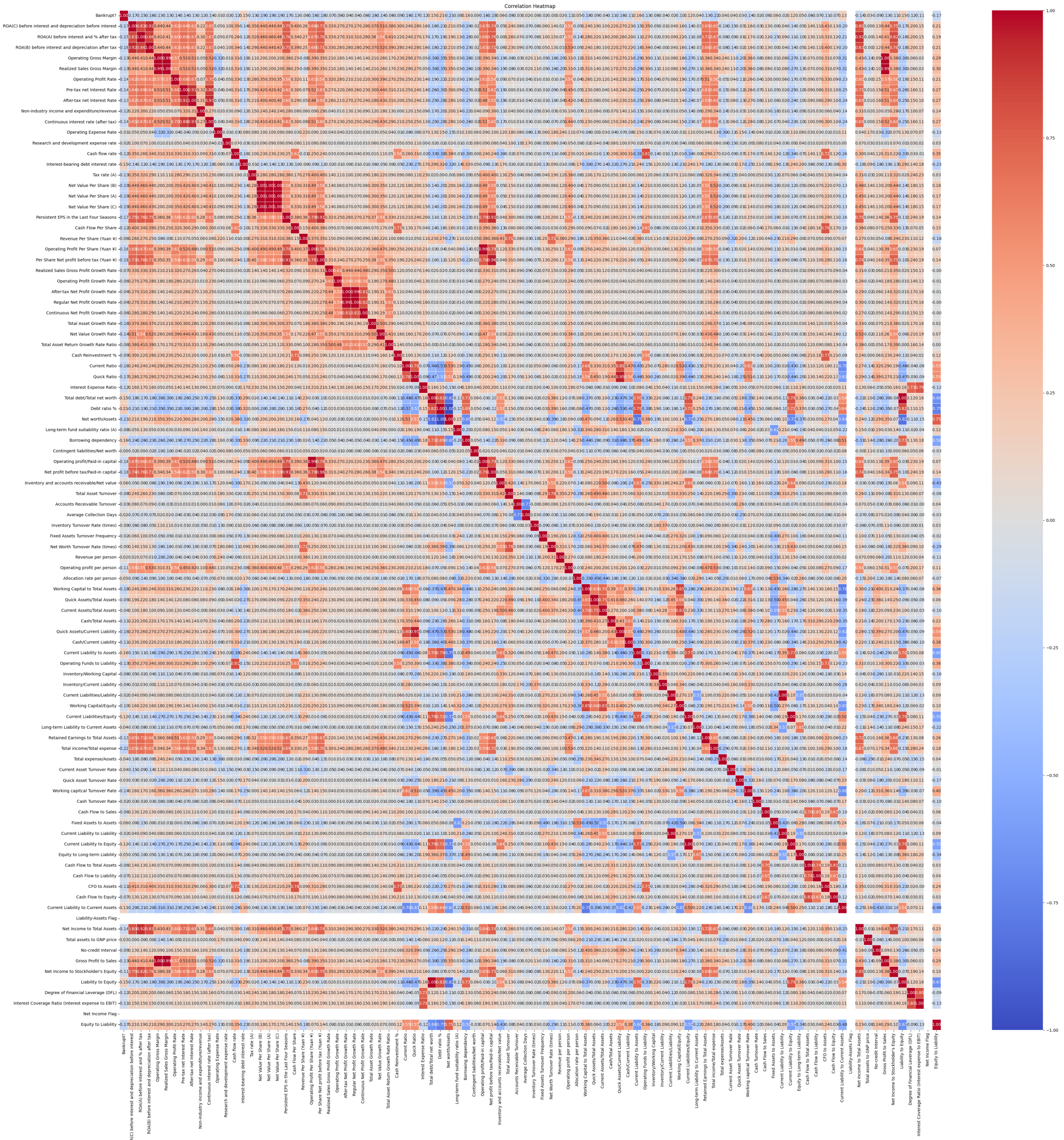
```
Column: Non-industry income and expenditure/revenue, Outliers Count: 1094
Column: Continuous interest rate (after tax), Outliers Count: 806
Column: Operating Expense Rate, Outliers Count: 0
Column: Research and development expense rate, Outliers Count: 182
Column: Cash flow rate, Outliers Count: 576
Column: Interest-bearing debt interest rate, Outliers Count: 396
Column: Tax rate (A), Outliers Count: 120
Column: Net Value Per Share (B), Outliers Count: 457
Column: Net Value Per Share (A), Outliers Count: 464
Column: Net Value Per Share (C), Outliers Count: 465
Column: Persistent EPS in the Last Four Seasons, Outliers Count: 508
Column: Cash Flow Per Share, Outliers Count: 532
Column: Revenue Per Share (Yuan ¥), Outliers Count: 478
Column: Operating Profit Per Share (Yuan ¥), Outliers Count: 442
Column: Per Share Net profit before tax (Yuan ¥), Outliers Count: 511
Column: Realized Sales Gross Profit Growth Rate, Outliers Count: 814
Column: Operating Profit Growth Rate, Outliers Count: 1008
Column: After-tax Net Profit Growth Rate, Outliers Count: 1033
Column: Regular Net Profit Growth Rate, Outliers Count: 1030
Column: Continuous Net Profit Growth Rate, Outliers Count: 1042
Column: Total Asset Growth Rate, Outliers Count: 1381
Column: Net Value Growth Rate, Outliers Count: 792
Column: Total Asset Return Growth Rate Ratio, Outliers Count: 674
Column: Cash Reinvestment %, Outliers Count: 617
Column: Current Ratio, Outliers Count: 589
Column: Quick Ratio, Outliers Count: 591
Column: Interest Expense Ratio, Outliers Count: 1362
Column: Total debt/Total net worth, Outliers Count: 407
Column: Debt ratio %, Outliers Count: 30
Column: Net worth/Assets, Outliers Count: 30
Column: Long-term fund suitability ratio (A), Outliers Count: 810
Column: Borrowing dependency, Outliers Count: 321
Column: Contingent liabilities/Net worth, Outliers Count: 942
Column: Operating profit/Paid-in capital, Outliers Count: 446
Column: Net profit before tax/Paid-in capital, Outliers Count: 476
Column: Inventory and accounts receivable/Net value, Outliers Count: 421
Column: Total Asset Turnover, Outliers Count: 351
Column: Accounts Receivable Turnover, Outliers Count: 659
Column: Average Collection Days, Outliers Count: 193
Column: Inventory Turnover Rate (times), Outliers Count: 0
Column: Fixed Assets Turnover Frequency, Outliers Count: 1418
Column: Net Worth Turnover Rate (times), Outliers Count: 513
Column: Revenue per person, Outliers Count: 729
Column: Operating profit per person, Outliers Count: 876
Column: Allocation rate per person, Outliers Count: 693
Column: Working Capital to Total Assets, Outliers Count: 75
Column: Quick Assets/Total Assets, Outliers Count: 2
Column: Current Assets/Total Assets, Outliers Count: 0
```

```
Column: Cash/Total Assets, Outliers Count: 496
Column: Quick Assets/Current Liability, Outliers Count: 596
Column: Cash/Current Liability, Outliers Count: 728
Column: Current Liability to Assets, Outliers Count: 95
Column: Operating Funds to Liability, Outliers Count: 657
Column: Inventory/Working Capital, Outliers Count: 944
Column: Inventory/Current Liability, Outliers Count: 426
Column: Current Liabilities/Liability, Outliers Count: 40
Column: Working Capital/Equity, Outliers Count: 153
Column: Current Liabilities/Equity, Outliers Count: 480
Column: Long-term Liability to Current Assets, Outliers Count: 620
Column: Retained Earnings to Total Assets, Outliers Count: 633
Column: Total income/Total expense, Outliers Count: 463
Column: Total expense/Assets, Outliers Count: 372
Column: Current Asset Turnover Rate, Outliers Count: 1399
Column: Quick Asset Turnover Rate, Outliers Count: 0
Column: Working capitcal Turnover Rate, Outliers Count: 578
Column: Cash Turnover Rate, Outliers Count: 0
Column: Cash Flow to Sales, Outliers Count: 1052
Column: Fixed Assets to Assets, Outliers Count: 62
Column: Current Liability to Liability, Outliers Count: 40
Column: Current Liability to Equity, Outliers Count: 480
Column: Equity to Long-term Liability, Outliers Count: 406
Column: Cash Flow to Total Assets, Outliers Count: 878
Column: Cash Flow to Liability, Outliers Count: 1212
Column: CFO to Assets, Outliers Count: 342
Column: Cash Flow to Equity, Outliers Count: 827
Column: Current Liability to Current Assets, Outliers Count: 276
Column: Liability-Assets Flag, Outliers Count: 8
Column: Net Income to Total Assets, Outliers Count: 561
Column: Total assets to GNP price, Outliers Count: 797
Column: No-credit Interval, Outliers Count: 1139
Column: Gross Profit to Sales, Outliers Count: 320
Column: Net Income to Stockholder's Equity, Outliers Count: 571
Column: Liability to Equity, Outliers Count: 404
Column: Degree of Financial Leverage (DFL), Outliers Count: 1503
Column: Interest Coverage Ratio (Interest expense to EBIT), Outliers Count: 1421
Column: Net Income Flag, Outliers Count: 0
Column: Equity to Liability, Outliers Count: 549
```

[51]:
```
numeric_columns = df.select_dtypes(include=['int64', 'float64'])

correlation_matrix = numeric_columns.corr()
plt.figure(figsize=(45, 45))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".
↪2f",linewidths=0.5)
plt.title('Correlation Heatmap')
```

```
plt.show()
```



[52]:
```
selected_predictors
```

[52]:
```
Bankrupt?                   1.000000
Debt ratio %                0.214388
Net worth/Assets           -0.214388
Total income/Total expense -0.203941
dtype: float64
```

[76]:
```
correlation_matrix = df.corr()
get = (correlation_matrix['Bankrupt?']>0.2) | (correlation_matrix['Bankrupt?
↪']<-0.2)
get.sum()
pred_colms = (get[get].index.to_list())
pred_colms

new_df = df[pred_colms]
new_df.shape
```

[76]: (6819, 4)

[77]:
```
new_df
```

[77]:

	Bankrupt?	Debt ratio %	Net worth/Assets	Total income/Total expense
0	1	0.207576	0.792424	0.002022
1	1	0.171176	0.828824	0.002226
2	1	0.207516	0.792484	0.002060
3	1	0.151465	0.848535	0.002336
4	1	0.106509	0.893491	0.002224
…	…	…	…	…
6814	0	0.124618	0.875382	0.002266
6815	0	0.099253	0.900747	0.002288
6816	0	0.038939	0.961061	0.002239
6817	0	0.086979	0.913021	0.002395
6818	0	0.014149	0.985851	0.002791

[6819 rows x 4 columns]

[60]:
```
numeric_categorical_df = df[['Bankrupt?', 'Liability-Assets Flag', 'Net Income␣
↪Flag']]
```

[82]:
```
numeric_categorical_df.head()
```

[82]:

	Bankrupt?	Liability-Assets Flag	Net Income Flag
0	1	0	1
1	1	0	1
2	1	0	1
3	1	0	1
4	1	0	1

[67]:
```
#H0bl There is no association between the 'Bankrupt?' status of a company and␣
↪its 'Liability-Assets Flag'
contingency_table_liability = pd.crosstab(df['Bankrupt?'], df['Liability-Assets␣
↪Flag'])
chi2_stat_la, p_value_la, dof_la, expected_freq_la =␣
↪chi2_contingency(contingency_table_liability)
```

```
#H0bn  There is no association between the 'Bankrupt?' status of a company and␣
↪its 'Net Income Flag'.
contingency_table_ni = pd.crosstab(df['Bankrupt?'], df['Net Income Flag'])
chi2_stat_ni, p_value_ni, dof_ni, expected_freq_ni =␣
↪chi2_contingency(contingency_table_ni)
```

[69]:
```
print(str(p_value_la)+" "+str(p_value_ni))
```

```
1.0 1.0
```

[81]:
```
# based on above chi-square test, it has been found that numeric_categorical␣
↪data has no effect on the the response variable
# so for analysis part i am not using those 2 columns

# Based on corretion matrix method we have found only 3 columns which are␣
↪effective immpact on response variable
#although more threshold value less data set columns will lead to strong␣
↪prediction model i will use only 0.2 as threshold value

# In one way i am doing feature selection for numeric and catgorical data which␣
↪is not as per proceess
# But instead of doing EDA/ visualization on large amount of predictors, it␣
↪will be good to perform visualization and modeling on selected columns

#But i am sure that remaining columns also affect respose variable.
# So i am conclude that i am doing this analysis based on very little columns␣
↪and it may be strong prediction
#in coming future i suggest do analysis based on domain knowledge
```

[90]:
```
#X = df.drop(['Bankrupt?', 'Liability-Assets Flag', 'Net Income Flag'], axis=1)␣
↪ # Independent variables
#y = df['Bankrupt?']  # Dependent variable
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=4)
#model = sm.MNLogit(y_train, sm.add_constant(X_train))  # Add constant for the␣
↪intercept
#result = model.fit()
```

[89]:
```
len(df.columns)
```

[89]: 96

[91]:
```
new_df.head()
```

[91]:
```
   Bankrupt?  Debt ratio %  Net worth/Assets  Total income/Total expense
0          1      0.207576          0.792424                    0.002022
```

```
1          1    0.171176        0.828824                  0.002226
2          1    0.207516        0.792484                  0.002060
3          1    0.151465        0.848535                  0.002336
4          1    0.106509        0.893491                  0.002224
```

[92]:

```
new_df.columns = ['Bankrupt', 'Debt_Ratio', 'Net_Worth_Assets',␣
↪'Income_Expense_Ratio']
```

[96]:

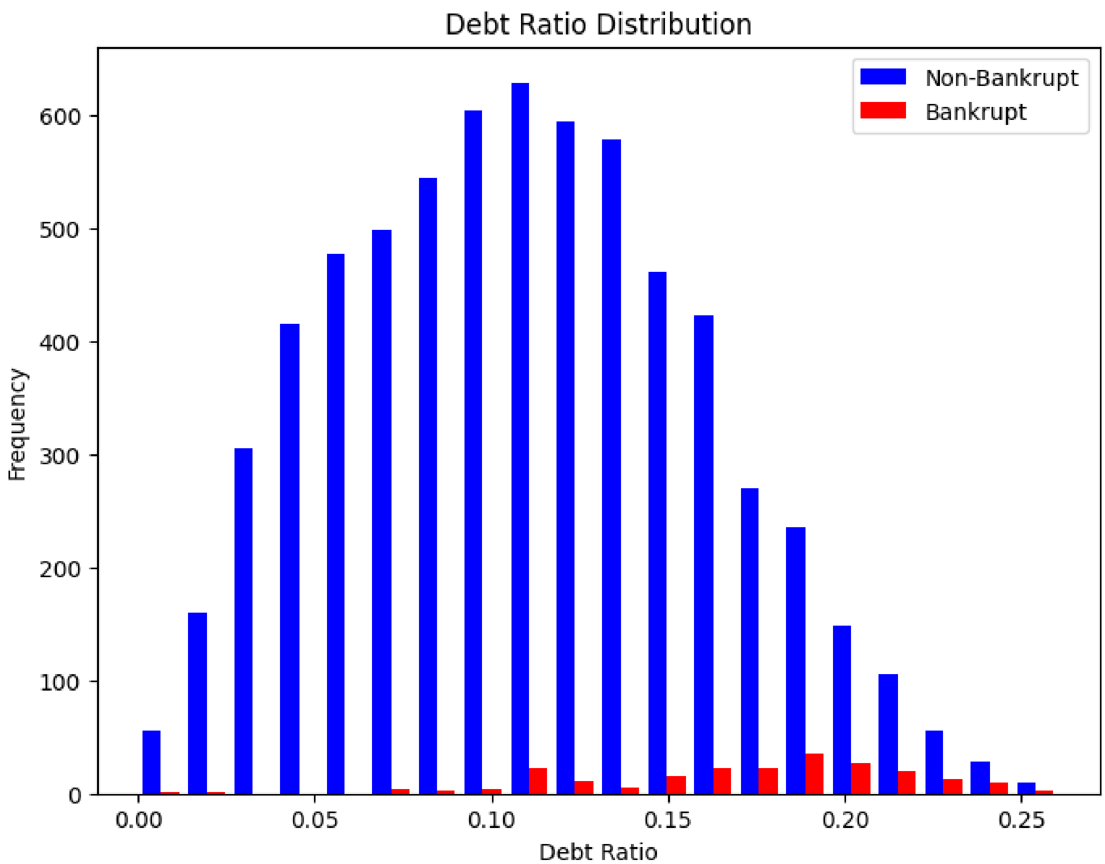
```
new_df.head()
```

[96]:

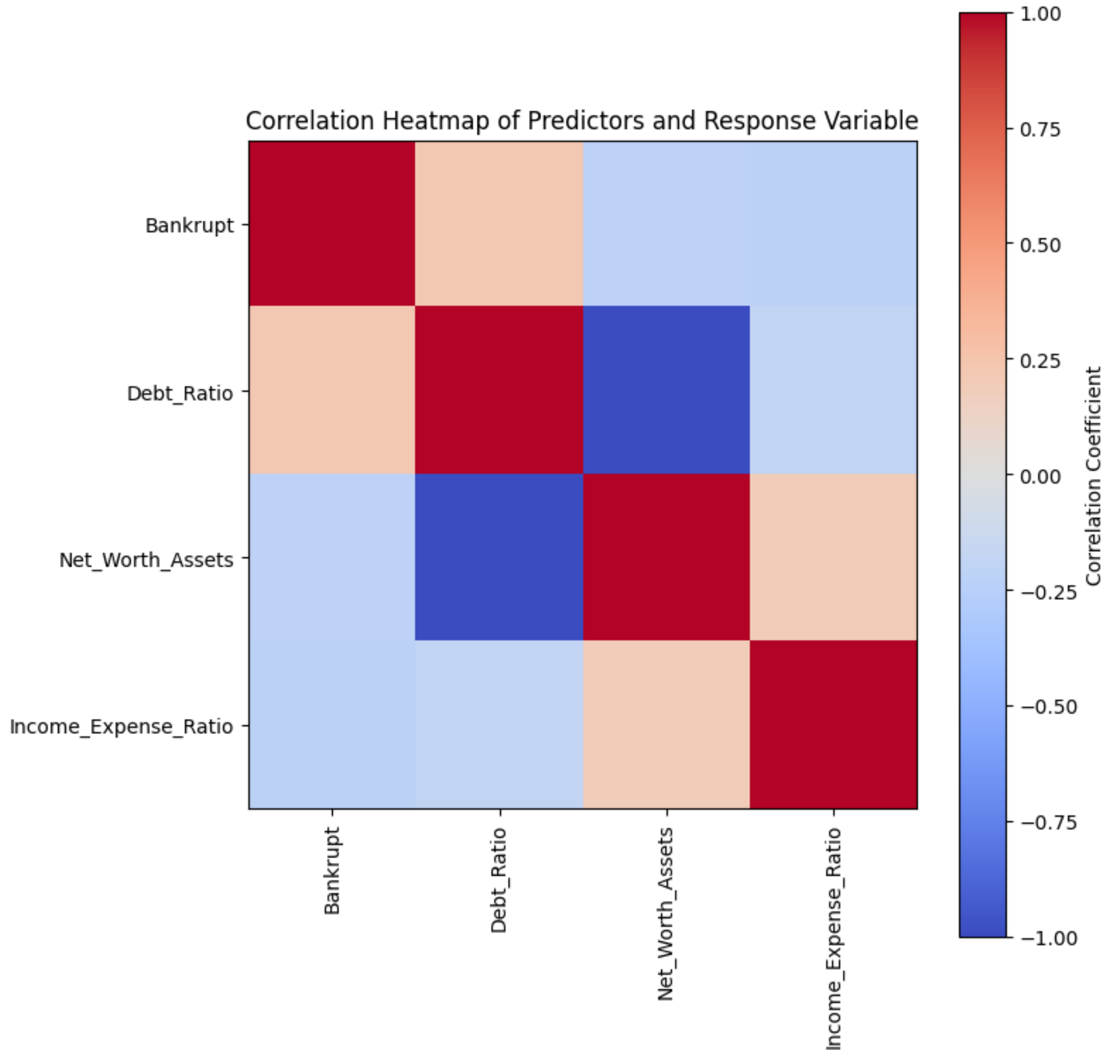
	Bankrupt	Debt_Ratio	Net_Worth_Assets	Income_Expense_Ratio
0	1	0.207576	0.792424	0.002022
1	1	0.171176	0.828824	0.002226
2	1	0.207516	0.792484	0.002060
3	1	0.151465	0.848535	0.002336
4	1	0.106509	0.893491	0.002224

[97]:

```
plt.figure(figsize=(8, 6))
plt.hist([new_df[new_df['Bankrupt'] == 0]['Debt_Ratio'],␣
↪new_df[new_df['Bankrupt'] == 1]['Debt_Ratio']], bins=20, color=['blue',␣
↪'red'], label=['Non-Bankrupt', 'Bankrupt'])
plt.title('Debt Ratio Distribution')
plt.xlabel('Debt Ratio')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

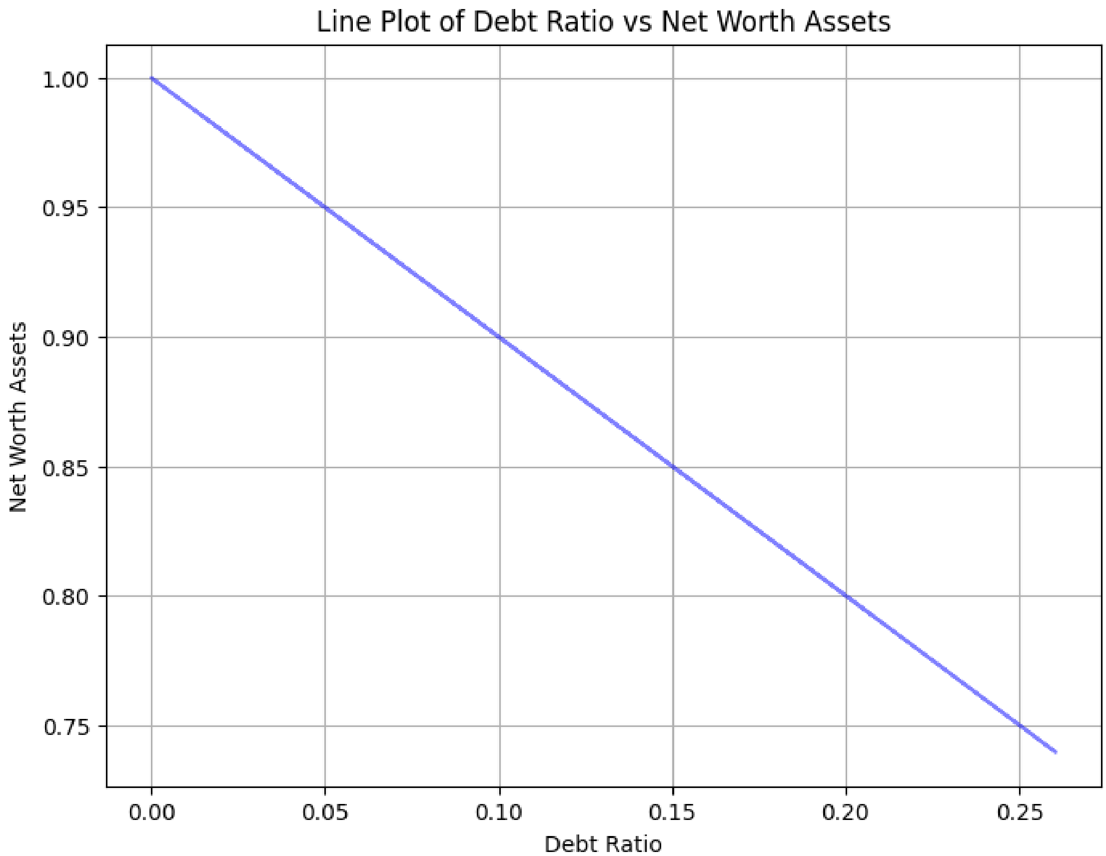


[99]:
```
corr_matrix = new_df.corr()
plt.figure(figsize=(8, 8))
plt.imshow(corr_matrix, cmap='coolwarm', interpolation='nearest')
plt.colorbar(label='Correlation Coefficient')
plt.title('Correlation Heatmap of Predictors and Response Variable')
plt.xticks(range(len(corr_matrix)), corr_matrix.columns, rotation=90)
plt.yticks(range(len(corr_matrix)), corr_matrix.columns)
plt.tight_layout()
plt.show()
```

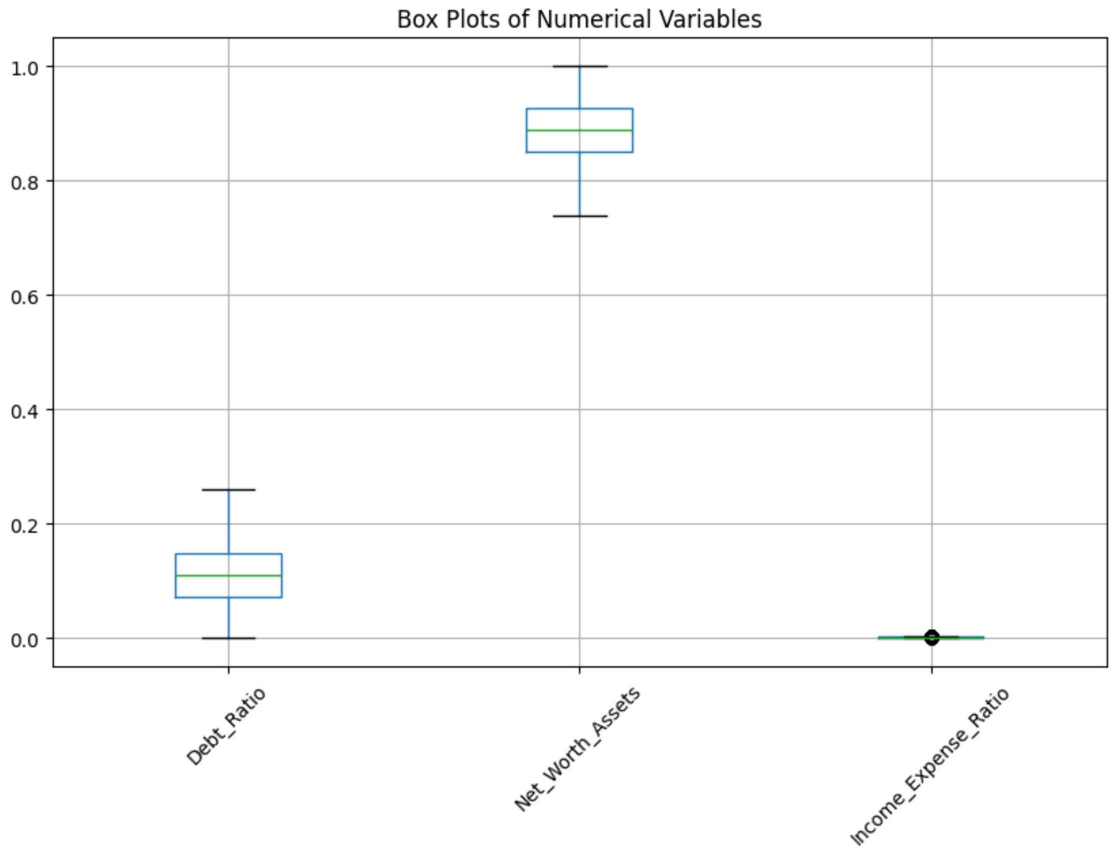


[101]:

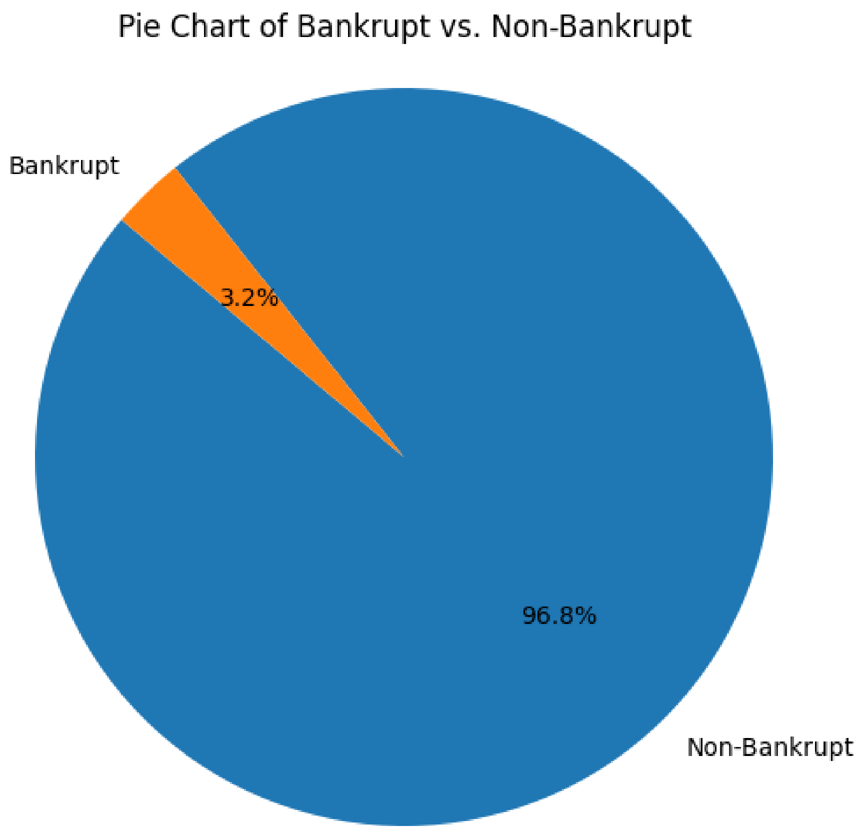
```
plt.figure(figsize=(8, 6))
plt.plot(new_df['Debt_Ratio'], new_df['Net_Worth_Assets'], color='blue',␣
↪alpha=0.5)
plt.title('Line Plot of Debt Ratio vs Net Worth Assets')
plt.xlabel('Debt Ratio')
plt.ylabel('Net Worth Assets')
plt.grid(True)
plt.show()
```



[102]:
```
plt.figure(figsize=(10, 6))
new_df.iloc[:, 1:].boxplot()
plt.title('Box Plots of Numerical Variables')
plt.xticks(rotation=45)
plt.show()
```



[115]:
```
plt.figure(figsize=(8, 6))
plt.pie(bankrupt_counts, labels=['Non-Bankrupt', 'Bankrupt'], autopct='%1.
  ↪1f%%', startangle=140)
plt.title('Pie Chart of Bankrupt vs. Non-Bankrupt')
plt.axis('equal')
plt.show()
```



[154]:
```
x = new_df[['Debt_Ratio', 'Net_Worth_Assets', 'Income_Expense_Ratio']]
y = new_df['Bankrupt']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)
x_train=sm.add_constant(x_train,prepend=False)
x_test=sm.add_constant(x_test,prepend=False)
model = sm.MNLogit(y_train, x_train).fit()
y_pred = model.predict(x_test)
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.094378
         Iterations: 35
```

```
C:\Users\ADMIN\AppData\Local\Programs\Python\Python312\Lib\site-
packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

[155]:
```
print(model.summary())
```

```
                           MNLogit Regression Results
==============================================================================
Dep. Variable:               Bankrupt   No. Observations:                 5455
Model:                        MNLogit   Df Residuals:                     5451
Method:                           MLE   Df Model:                            3
Date:                Mon, 17 Jun 2024   Pseudo R-squ.:                  0.3136
Time:                        19:29:14   Log-Likelihood:                -514.83
converged:                      False   LL-Null:                       -750.08
Covariance Type:            nonrobust   LLR p-value:                1.179e-101
================================================================================
========
          Bankrupt=1       coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
--------
Debt_Ratio              17.9178   8.39e+06   2.13e-06      1.000   -1.65e+07
1.65e+07
Net_Worth_Assets        -3.4177   8.39e+06  -4.07e-07      1.000   -1.65e+07
1.65e+07
Income_Expense_Ratio -7924.3796    593.200    -13.359      0.000   -9087.030
-6761.729
const                   14.4973   8.39e+06   1.73e-06      1.000   -1.65e+07
1.65e+07
================================================================================
========
```

[ ]:
```
1. Debt Ratio: The coefficient is very high (17.9178), but the p-value is very
  ↪high (1.000),
indicating that the variable is not statistically significant in predicting
  ↪bankruptcy. Therefore,
we fail to reject the null hypothesis that the debt ratio has no effect on
  ↪bankruptcy.

2. Net Worth Assets: The coefficient is negative (-3.4177), but like the debt
  ↪ratio, the p-value is very high (1.000),
indicating that this variable is also not statistically significant in
  ↪predicting bankruptcy.

3. Income Expense Ratio: The coefficient is quite large in magnitude (-7924.
  ↪3796),
and the p-value is very low (close to 0), indicating that this variable is
  ↪statistically significant in predicting bankruptcy.
Therefore, we reject the null hypothesis that the income expense ratio has no
  ↪effect on bankruptcy.

Based on this analysis, the income expense ratio seems to be the most
  ↪influential predictor of bankruptcy among the variables considered.
```

```
A higher income expense ratio means a company is spending more compared to its␣
↪income, which could indicate financial instability and
increase the likelihood of bankruptcy.
This is evident from its substantial coefficient value (-7924.3796), indicating␣
↪a strong impact on the likelihood of bankruptcy
Therefore, understanding and managing this ratio is important for avoiding␣
↪financial distress.
```

[ ]:
```
1. Debt Ratio (Positive Coefficient): With a coefficient of approximately 17.
↪92, an increase in the debt ratio suggests a higher risk of
 bankruptcy. For each unit increase in the debt ratio, the probability of␣
↪bankruptcy rises.

2. Net Worth Assets (Negative Coefficient): The coefficient of around -3.42␣
↪implies that a decrease in net worth assets elevates the likelihood of
 bankruptcy. Each unit decrease in net worth assets corresponds to a higher␣
↪probability of facing bankruptcy.

3. Income Expense Ratio (Negative Coefficient): With a coefficient of about␣
↪-7924.38, an increase in the income
  expense ratio indicates a reduced risk of bankruptcy. For every unit␣
↪increase in this ratio, the probability of bankruptcy decreases␣
↪significantly.
```

[156]:
```
y_pred = model.predict(x_test)
```

[158]:
```
y_pred.head()
```

[158]:

	0	1
1581	0.909649	0.090351
6564	0.999176	0.000824
3465	0.987755	0.012245
3382	0.992748	0.007252
5011	0.999506	0.000494

[160]:
```
y_test.head()
```

[160]:
```
1581    0
6564    0
3465    0
3382    0
5011    0
Name: Bankrupt, dtype: int64
```

[164]:
```
y_pred_class = y_pred.idxmax(axis=1)
```

[165]:
```
precision = precision_score(y_test, y_pred_class)
recall = recall_score(y_test, y_pred_class)
accuracy = accuracy_score(y_test, y_pred_class)
f1 = f1_score(y_test, y_pred_class)
conf_matrix = confusion_matrix(y_test, y_pred_class)
```

[166]:
```
accuracy
```

[166]: 0.9596774193548387

[168]:
```
conf_matrix
```

[168]:
```
array([[1306,    6],
       [  49,    3]], dtype=int64)
```

[169]:
```
precision
```

[169]: 0.3333333333333333

[170]:
```
recall
```

[170]: 0.057692307692307696

[171]:
```
f1
```

[171]: 0.09836065573770492

[ ]:
```
The precision score of 0.333 indicates that when the model predicts bankruptcy,␣
↪it is correct approximately 33.3% of the time.
The recall score of 0.058 indicates that the model correctly identifies␣
↪approximately 5.8% of the actual bankruptcies.

The high accuracy score of 0.960 suggests that the model performs well overall,␣
↪but the low precision and recall scores
indicate that it may struggle to correctly identify bankruptcies, which could␣
↪be due to class imbalance or other factors.

Based on the confusion matrix:

True Negatives (TN): 1306
False Positives (FP): 6
False Negatives (FN): 49
True Positives (TP): 3
This matrix illustrates the model's performance in classifying instances as␣
↪bankrupt or non-bankrupt.
It shows that the model correctly identified 1306 instances as non-bankrupt␣
↪(TN) and 3 instances as bankrupt (TP).
```

```
However, it incorrectly classified 6 instances as bankrupt when they were␣
↪actually non-bankrupt (FP), and 49 instances as non-bankrupt when they
were actually bankrupt (FN).
```

[ ]:
```
so in summary

The model achieved a high accuracy of 95.97%, indicating that it correctly␣
↪classified the majority of instances.

However, the precision (33.33%) and recall (5.77%) for bankrupt instances are␣
↪low, suggesting that the model's ability
to correctly identify bankrupt cases is limited.

The F1 score, which considers both precision and recall, is also relatively low␣
↪at 9.84%, indicating the model's
overall performance considering both false positives and false negatives.

The confusion matrix reveals that while the model correctly identified a large␣
↪number of non-bankrupt instances,
it struggled with correctly classifying bankrupt instances, resulting in a␣
↪higher number of
false negatives and false positives.
```