Data Mining Project Report: Company Bankruptcy Prediction

Alber Abbas (21l-5623), Wara Imran (21l-6244), Ayesha Ulfat (21l-6216)

1 Motivation

The motivation driving this project lies in the profound economic impact of company bankruptcies, both locally and globally. Bankruptcies can disrupt markets, affect stakeholders, and have far-reaching consequences on employment, investment, and economic stability. Understanding the dynamics and predictors of bankruptcy is crucial for stakeholders, including investors, creditors, and policymakers, to make informed decisions and mitigate risks effectively. Leveraging data mining techniques to analyze comprehensive financial datasets can provide valuable insights into the factors influencing bankruptcy outcomes, thereby assisting in risk management, strategic planning, and regulatory compliance.

2 Introduction

Bankruptcy represents a critical aspect of corporate finance, warranting a thorough examination of its determinants, implications, and predictive factors. In this project, we delve into a comprehensive dataset on corporate bankruptcies, encompassing financial records, industry data, and other relevant variables. Our primary objectives include uncovering trends in bankruptcy occurrences, identifying key factors contributing to bankruptcy risk, assessing the efficacy of financial indicators in predicting bankruptcy, and constructing predictive models to aid stakeholders in proactive risk management and decision-making. By elucidating the intricacies of bankruptcy dynamics, we aim to empower stakeholders with actionable insights to navigate the complexities of the financial landscape and mitigate the adverse effects of corporate insolvency.

3 Dataset Description

The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange.

4 Attribute Information

Y - Bankrupt?: Class label

X1 - ROA(C) before interest and depreciation before interest: Return On Total Assets(C)

X2 - ROA(A) before interest and

X3 - ROA(B) before interest and depreciation after tax: Return On Total Assets(B)

X4 - Operating Gross Margin: Gross Profit/Net Sales

X5 - Realized Sales Gross Margin: Realized Gross Profit/Net Sales

X6 - Operating Profit Rate: Operating Income/Net Sales

X7 - Pre-tax net Interest Rate: Pre-Tax Income/Net Sales

X8 - After-tax net Interest Rate: Net Income/Net Sales

X9 - Non-industry income and expenditure/revenue: Net Non-operating Income Ratio

X10 - Continuous interest rate (after tax): Net Income-Exclude Disposal Gain or Loss/Net Sales

X11 - Operating Expense Rate: Operating Expenses/Net Sales

X12 - Research and development expense rate: (Research and Development Expenses)/Net Sales

X13 - Cash flow rate: Cash Flow from Operating/Current Liabilities

X14 - Interest-bearing debt interest rate: Interest-bearing Debt/Equity

X15 - Tax rate (A): Effective Tax Rate

X16 - Net Value Per Share (B): Book Value Per Share(B)

X17 - Net Value Per Share (A): Book Value Per Share(A)

X18 - Net Value Per Share (C): Book Value Per Share(C)

X19 - Persistent EPS in the Last Four Seasons: EPS-Net Income

X20 - Cash Flow Per Share

X21 - Revenue Per Share (Yuan ¥): Sales Per Share

X22 - Operating Profit Per Share (Yuan ¥): Operating Income Per Share

X23 - Per Share Net profit before tax (Yuan ¥): Pretax Income Per Share

X24 - Realized Sales Gross Profit Growth Rate

X25 - Operating Profit Growth Rate: Operating Income Growth

X26 - After-tax Net Profit Growth Rate: Net Income Growth

X27 - Regular Net Profit Growth Rate: Continuing Operating Income after Tax Growth

X28 - Continuous Net Profit Growth Rate: Net Income-Excluding Disposal Gain or Loss Growth

X29 - Total Asset Growth Rate: Total Asset Growth

X30 - Net Value Growth Rate: Total Equity Growth

X31 - Total Asset Return Growth Rate Ratio: Return on Total Asset Growth

X32 - Cash Reinvestment

X33 - Current Ratio

X34 - Quick Ratio: Acid Test

X35 - Interest Expense Ratio: Interest Expenses/Total Revenue

X36 - Total debt/Total net worth: Total Liability/Equity Ratio

X37 - Debt ratio

X38 - Net worth/Assets: Equity/Total Assets

X39 - Long-term fund suitability ratio (A): (Long-term Liability+Equity)/Fixed Assets

X40 - Borrowing dependency: Cost of Interest-bearing Debt

X41 - Contingent liabilities/Net worth: Contingent Liability/Equity

X42 - Operating profit/Paid-in capital: Operating Income/Capital

X43 - Net profit before tax/Paid-in capital: Pretax Income/Capital

X44 - Inventory and accounts receivable/Net value: (Inventory+Accounts Receivables)/Equity

X45 - Total Asset Turnover

X46 - Accounts Receivable Turnover

X47 - Average Collection Days: Days Receivable Outstanding

X48 - Inventory Turnover Rate (times)

X49 - Fixed Assets Turnover Frequency

X50 - Net Worth Turnover Rate (times): Equity Turnover

X51 - Revenue per person: Sales Per Employee

X52 - Operating profit per person: Operation Income Per Employee

X53 - Allocation rate per person: Fixed Assets Per Employee

X54 - Working Capital to Total Assets

X55 - Quick Assets/Total Assets

X56 - Current Assets/Total Assets

X57 - Cash/Total Assets

X58 - Quick Assets/Current Liability

X59 - Cash/Current Liability

X60 - Current Liability to Assets

X61 - Operating Funds to Liability

X62 - Inventory/Working Capital

X63 - Inventory/Current Liability

X64 - Current Liabilities/Liability

X65 - Working Capital/Equity

X66 - Current Liabilities/Equity

X67 - Long-term Liability to Current Assets

X68 - Retained Earnings to Total Assets

X69 - Total income/Total expense

X70 - Total expense/Assets

X71 - Current Asset Turnover Rate: Current Assets to Sales

X72 - Quick Asset Turnover Rate: Quick Assets to Sales

X73 - Working capital Turnover Rate: Working Capital to Sales

X74 - Cash Turnover Rate: Cash to Sales

X75 - Cash Flow to Sales

X76 - Fixed Assets to Assets

X77 - Current Liability to Liability

X78 - Current Liability to Equity

X79 - Equity to Long-term Liability

X80 - Cash Flow to Total Assets

X81 - Cash Flow to Liability

X82 - CFO to Assets

X83 - Cash Flow to Equity

X84 - Current Liability to Current Assets

X85 - Liability-Assets Flag: 1 if Total Liability exceeds Total Assets, 0 otherwise

X86 - Net Income to Total Assets

X87 - Total assets to GNP price

X88 - No-credit Interval

X89 - Gross Profit to Sales

X90 - Net Income to Stockholder's Equity

X91 - Liability to Equity

X92 - Degree of Financial Leverage (DFL)

X93 - Interest Coverage Ratio (Interest expense to EBIT)

X94 - Net Income Flag: 1 if Net Income is Negative for the last two years, 0 otherwise

X95 - Equity to Liability

5 Data Preprocessing

5.1 Number of features

96

5.2 Data Type

Features are of the type int or float.

5.3 Missing Values

Rows with null values have been dropped from the dataset.

5.4 Duplicate Values

The data has no duplicate values.

5.5 Outlier Detection and Removal

Outliers have been detected using Interquartile Range (IQR) method and have been replaced with the mean value of the column.

5.6 Data Visualization

Histograms are used to visualize the distribution of features, roc curves are used to understand the model's performance and bar plot was used to find the best accuracy.

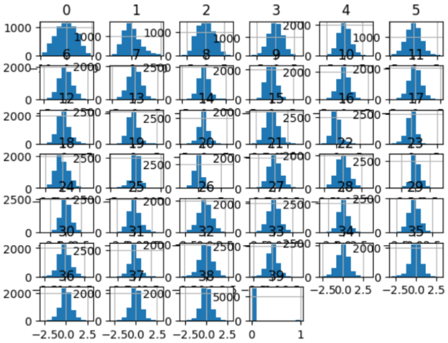


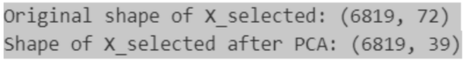
Figure 1: Histogram (PCA)

5.7 Feature Selection

Heatmap has been used to see the correlation between attributes. Feature selection is done using a backward elimination method with a Random Forest Regressor.

5.8 Principal Component Analysis

- Select a subset of features from the DataFrame df (assuming features contains the names of these features) and assign it to X_{selected}.
- Standardize these selected features using `StandardScaler()` to ensure each feature has a mean of 0 and a standard deviation of 1.
- Apply PCA to $X_{\text{selected\_scaled}}$ to transform the data into a lower-dimensional space while retaining 95% of the variance.
- Store the transformed data after PCA in $X_{\text{selected\_pca}}$.
- Finally, print the original shape of the selected features and the shape after PCA, indicating the reduction in dimensionality achieved by PCA. The reduced features are finally 39.



```
Original shape of X_selected: (6819, 72)
Shape of X_selected after PCA: (6819, 39)
```

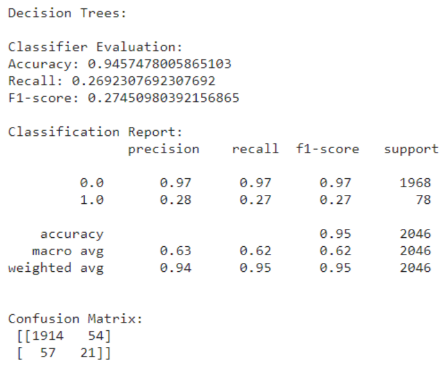
Figure 2: PCA

5.9 Class Imbalance Handling

- 3% of class imbalance in the data.

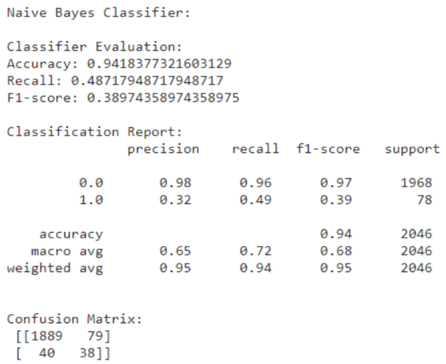
- Oversampling/Undersampling:
  - RandomOverSampler and RandomUnderSampler have been applied based on the class imbalance ratio to balance the classes in the dataset.
- Before resampling: Class Distribution: Counter({0.0: 6599, 1.0: 220})
- After resampling: Class Distribution: Counter({1.0: 6599, 0.0: 6599})

6 Performance of the Models initially



```
Decision Trees:

Classifier Evaluation:
Accuracy: 0.9457478005865103
Recall: 0.2692307692307692
F1-score: 0.27450980392156865

Classification Report:
              precision    recall  f1-score   support

         0.0       0.97      0.97      0.97      1968
         1.0       0.28      0.27      0.27        78

    accuracy                           0.95      2046
   macro avg       0.63      0.62      0.62      2046
weighted avg       0.94      0.95      0.95      2046


Confusion Matrix:
 [[1914   54]
 [  57   21]]
```

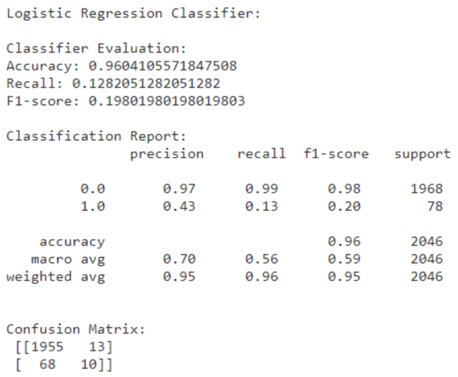
Figure 3: Decision Tree



```
Naive Bayes Classifier:

Classifier Evaluation:
Accuracy: 0.9418377321603129
Recall: 0.48717948717948717
F1-score: 0.38974358974358975

Classification Report:
              precision    recall  f1-score   support

         0.0       0.98      0.96      0.97      1968
         1.0       0.32      0.49      0.39        78

    accuracy                           0.94      2046
   macro avg       0.65      0.72      0.68      2046
weighted avg       0.95      0.94      0.95      2046


Confusion Matrix:
 [[1889   79]
 [  40   38]]
```

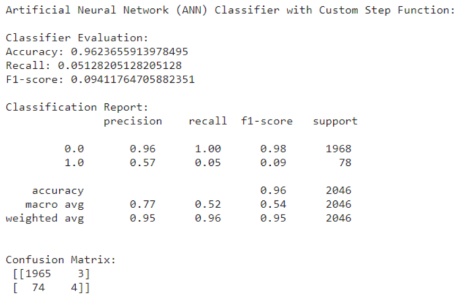
Figure 4: Naive Bayes

While accuracy is a commonly used metric to evaluate the overall performance of a classification model, precision and recall provide more nuanced insights, especially in scenarios where class imbalance exists or when different



```
Logistic Regression Classifier:

Classifier Evaluation:
Accuracy: 0.9604105571847508
Recall: 0.1282051282051282
F1-score: 0.19801980198019803

Classification Report:
               precision    recall  f1-score   support

         0.0       0.97      0.99      0.98      1968
         1.0       0.43      0.13      0.20        78

    accuracy                           0.96      2046
   macro avg       0.70      0.56      0.59      2046
weighted avg       0.95      0.96      0.95      2046


Confusion Matrix:
 [[1955   13]
 [  68   10]]
```

Figure 5: Logistic Regression



```
Artificial Neural Network (ANN) Classifier with Custom Step Function:

Classifier Evaluation:
Accuracy: 0.9623655913978495
Recall: 0.05128205128205128
F1-score: 0.09411764705882351

Classification Report:
               precision    recall  f1-score   support

         0.0       0.96      1.00      0.98      1968
         1.0       0.57      0.05      0.09        78

    accuracy                           0.96      2046
   macro avg       0.77      0.52      0.54      2046
weighted avg       0.95      0.96      0.95      2046


Confusion Matrix:
 [[1965    3]
 [  74    4]]
```

Figure 6: ANN

types of errors have varying costs. Accuracies can be misleading. So precision and recall complement accuracy by providing insights into the model's performance with respect to specific aspects of classification, particularly in scenarios with class imbalance or asymmetric costs associated with different types of errors.

7 Performance of the model:

Since the models were not performing well despite having good accuracies, their recall and precision were very low. To enhance their performance, several techniques were applied:

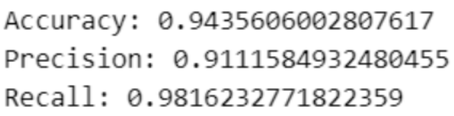
- Increasing hidden layers and other architectural changes: The models were modified by increasing the number of hidden layers or making other architectural changes to improve their capacity to capture complex patterns in the data.

- Hyperparameter tuning: Hyperparameters, such as learning rate, regularization strength, batch size, and others, were tuned to find the optimal configuration for the models, leading to better performance.

- Ensembles: Ensemble methods, such as bagging, boosting, or stacking, were employed to combine the predictions of multiple base models, leveraging the diversity of individual models to improve overall performance.
- Other techniques: Various other techniques, such as feature engineering, data preprocessing, and model selection, were also applied to further enhance the performance of the models.

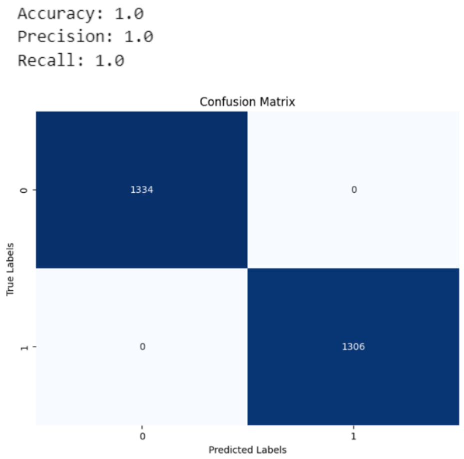
7.0.1 ANN

A Sequential model is initialized, allowing the creation of a linear stack of layers. One hidden layer with 64 units and ReLU activation function is added. An output layer with 1 unit and a sigmoid activation function is added for binary classification. The model is compiled using the Adam optimizer and binary cross-entropy loss function. Accuracy is chosen as the metric for model evaluation. The model is trained on the training data with 3 epochs and a batch size of 32. A validation split of 20% is used for monitoring the model's performance during training.



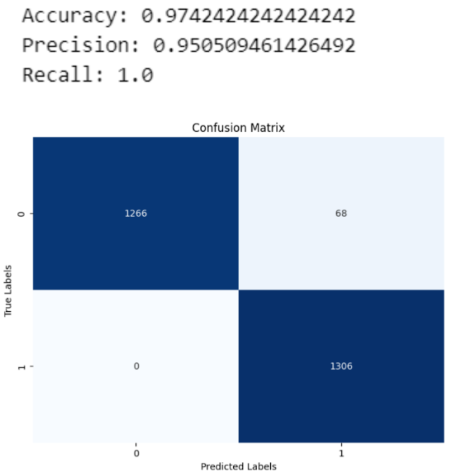
```
Accuracy: 0.9435606002807617
Precision: 0.9111584932480455
Recall: 0.9816232771822359
```

Figure 7: ANN



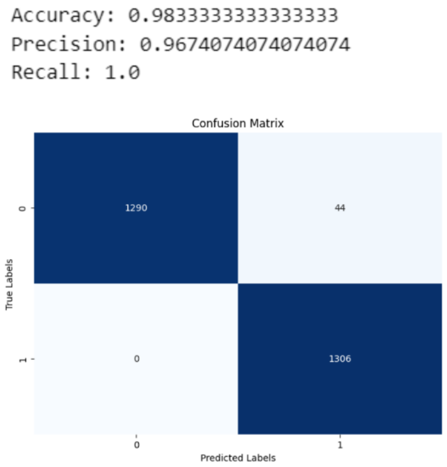
```
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
```

Figure 8: Random Forest



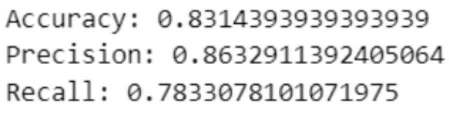
```
Accuracy: 0.9742424242424242
Precision: 0.950509461426492
Recall: 1.0
```

Figure 9: KNN



```
Accuracy: 0.9833333333333333
Precision: 0.9674074074074074
Recall: 1.0
```

Figure 10: Decision Trees



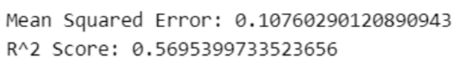
```
Accuracy: 0.8314393939393939
Precision: 0.8632911392405064
Recall: 0.7833078101071975
```

Figure 11: Naïve Bayes

8 Hyperparameter tuning

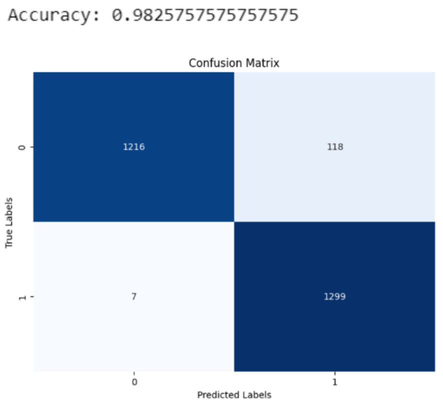
8.0.1 Logistic Regression

Grid Search: Hyperparameters for Logistic Regression are tuned using GridSearchCV to optimize model performance.



```
Mean Squared Error: 0.10760290120890943
R^2 Score: 0.5695399733523656
```

Figure 12: Linear Regression



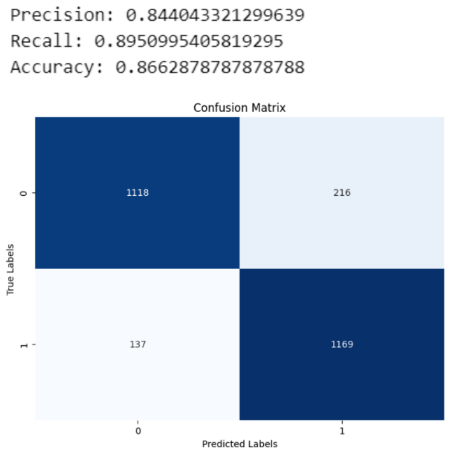
```
Accuracy: 0.9825757575757575
```

Confusion Matrix

True Labels	0	1
0	1216	118
1	7	1299

Figure 13: Gradient Boosting

Best Estimator: The best estimator with optimized hyperparameters is identified for improved predictive accuracy.



```
Precision: 0.844043321299639
Recall: 0.8950995405819295
Accuracy: 0.8662878787878788
```

Confusion Matrix

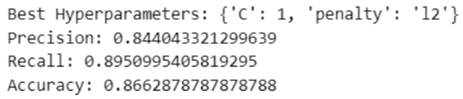
True Labels	0	1
0	1118	216
1	137	1169

Figure 14: Logistic Regression

In order to improve the performance of the model, the hyperparameters were tuned. The following results were obtained:

- A parameter grid (`param_grid`) is defined, specifying the hyperparameters to be tuned (penalty and C for regularization strength).

- `LogisticRegression()` model is initialized.
- Grid search (`GridSearchCV`) is performed with cross-validation (`cv=5`) using accuracy as the scoring metric. This process identifies the best combination of hyperparameters from the grid.
- The best estimator and its hyperparameters are obtained.
- Predictions are made on the test set using the best estimator.

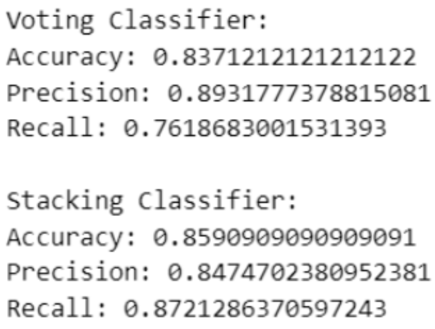


```
Best Hyperparameters: {'C': 1, 'penalty': 'l2'}
Precision: 0.844043321299639
Recall: 0.8950995405819295
Accuracy: 0.8662878787878788
```

Figure 15: Hyperparameter Tuning

9 Ensembles

Naïve Bayes and Logistic Regression have been combined to achieve an improved performance through the ensemble's technique. Voting and Stacking Classifiers were used and performances were compared.



```
Voting Classifier:
Accuracy: 0.8371212121212122
Precision: 0.8931777378815081
Recall: 0.7618683001531393

Stacking Classifier:
Accuracy: 0.8590909090909091
Precision: 0.8474702380952381
Recall: 0.8721286370597243
```

Figure 16: Evaluating Metrics

10 Performance Comparisons

Accuracies of all the models have been compared to see which performed the best among all.

The results suggest that Random Forest has shown the best performance among all the models.

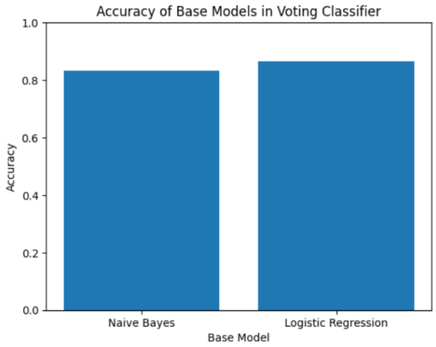


Figure 17: Bar Chart

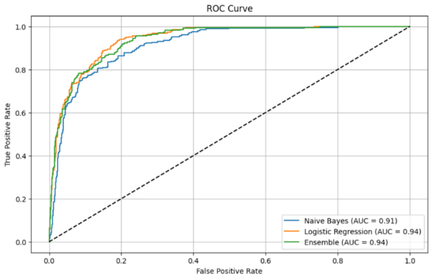


Figure 18: ROC curve

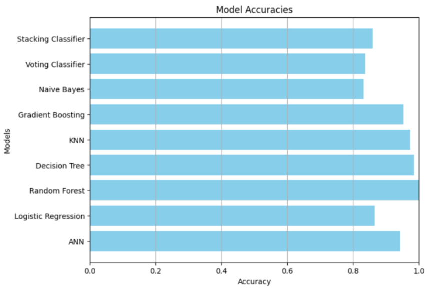


Figure 19: Comparing Accuracies