Machine Learning Framework for Enterprise Sustainability Prediction

Kuber Budhija 2021260 | Palak Bhardwaj 2022344 | Rishi Pendyala 2022403 | Yashovardhan Singhal 2022591

Project Repository

You can find the GitHub repository for this project here.

1. Motivation

There has been a significant increase in the number of companies coming up. However, all of these companies do not manage to stay successful over time and burn up. Bankruptcy is a curse for the organization, investors, employees and all the stakeholders. It is expressed as the inability of a company to pay its debts to its creditors. Bankruptcy prediction is important for modern economies because early warnings of bankruptcy can help both investors and public policy makers to take effective decisions to minimize the impact. Despite the popularity of the standard statistical methods, application related problems of the current methods to corporate bankruptcy prediction still remain. We are using a high dimensional dataset with 96 features and machine learning models such as naive bayes, random forest classifier to give accurate predictions and effective results.

2. Introduction

Classification Algorithms have improved so much in recent years. Using these classification algorithms, we can classify a company and identify the factors which significantly affect the financial health of a particular company, this can be Operating Margin, Profit Rate, Liabilities etc. This is also significant for Better allocation of resources, Input to policy makers, Corrective action for business managers, Identification of sector wide problems, Signal to Investors and some of the major economic majors. Visualizing interesting patterns present in the data by different plots to identify the outliers and study the distribution of the data. Our target attribute as per the data is "Bankrupt?" which is a binary attribute, hence, we have used classification algorithms. We aimed to find the suitable features that affect bankruptcy using different feature learning techniques such as Naive Bayes, Logistic Regression and Random Forest Classifier; test and compare multiple machine learning models for our proposed task and analyse the results. This will help the real life application as this model can be used as an estimator of the financial health and help organizations to make wise business decisions.

3. Literature Survey

- Artifical Neural Network and other Machine Learning Techniques such as the Support Vector Machine were used by The School of Expertness and Valuation, Institute of Technology and Business in Czech Republic. They emphasised how development of companies going to bankrupt can be predicted by using artificial neural networks but at the same time the disadvantages of ANNs include requirement of high quality data and possible illogical behavior of networks [1].

- Machine learning models and bankruptcy prediction: In this study, the authors have used the data from 1985 to 2013 based on North American firms to test different machine learning models such as support vector machines, boosting and random forest classifier on this data and predict bankruptcy one year prior to the event, and then compare their performance with results from discriminant analysis, logistic regression, and neural networks. [2].

- Bankruptcy Prediction Using ML, by Nanxi Wang: In this paper, three relatively new methods have been proposed for predicting bankruptcy based on real-life data. The usage of the three models (support vector machine, neural network with dropout, autoencoder) in economics or finance is comparatively hard to find. So, the paper aims to find out if they work well in the economic field, by predicting company bankruptcy [3].

4. Dataset

4.1. Data Description

The dataset was collected from the Taiwan Economic Journal for the years 1999 to 2009 [4]. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. This dataset has around 96 parameters with mainly continuous features in 6819 rows. This would help us conduct a detailed analysis.

Few feature descriptions are below:

1. ROA(C) before interest and depreciation before interest : It refers to a company's Return on Assets (ROA) calculated before considering interest expenses and depreciation costs.

2. Total Asset Growth Rate: Total Asset Growth Rate defined as year-over-year percentage change in total assets.

3. Non-industry income and expenditure/revenue : It refers to financial activities and streams of income or costs that are outside of a company's primary business operations or industry-specific activities.

4. Tax Rate: A tax rate is the percentage at which an individual or corporation is taxed.

5. Revenue per share: Amount of revenue over common shares outstanding. Answers the question, what's the ownership of sales to each share? Increasing revenue per share (RPS) over time is a good sign, because it means each share now has claim to more revenues.

4.2. Exploratory Data Analysis

We performed EDA on the entire dataset to gain a better understanding of its nature. A significant imbalance was observed between the majority and minority classes, i.e., there are considerably more rows with class=0 compared to rows with class=1. This is illustrated in the plot below.

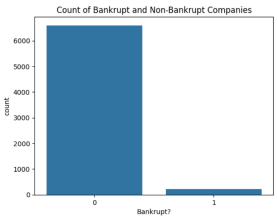

Figure 1. Class Imbalance between Bankrupt and Non-Bankrupt Companies

A large number of companies (6000+) are labeled as non-bankrupt (label 0) where a small number of companies (barely over 100) are labeled as bankrupt (Figure 1). The dataset is highly imbalanced toward non-bankrupt companies, with a very small proportion being bankrupt.

Missing and NULL data : No missing values were present in the dataset.

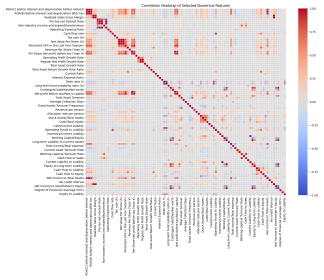

Figure 2. Correlation Matrix

Correlation between features : Several financial ratios, such as ROA(C) before interest and depreciation and ROA(A) before interest and % after tax, exhibit strong positive correlations. The presence of highly correlated features suggests multicollinearity.

Outlier Detection:

Outlier detection: The bar chart below shows the percentage of outliers present in each numerical feature in our dataset, with the value reaching 20% for some features. Features like **ROA(A)** and **Realized Sales Gross Margin** have significant outliers.

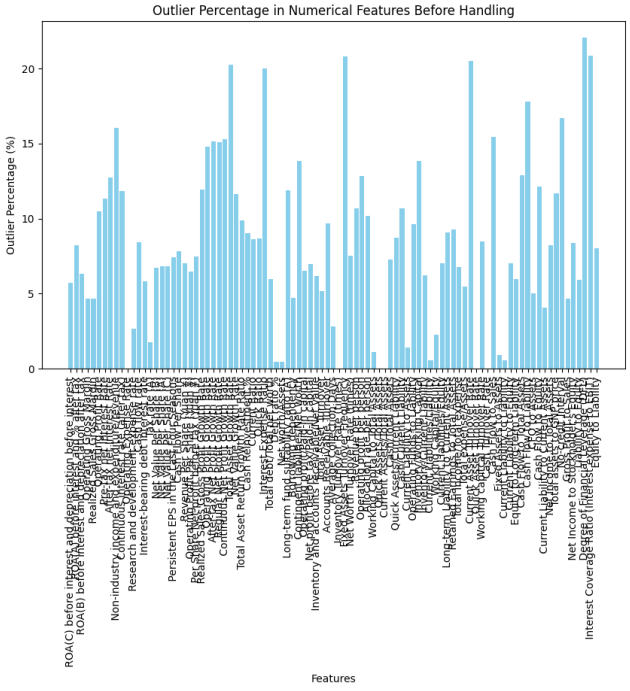

Figure 3. Outlier Detection

4.3. Data Preprocessing

Our Exploratory Data Analysis revealed some of the issues we needed to take care of before fitting, namely, data imbalance and existence of correlated features.

Handling Imbalance:

- **RandomOverSampler** is first applied to increase the number of samples in the minority class to match the majority class.
- **RandomUnderSampler** is applied afterward to reduce the number of samples in the majority class, balancing the dataset further.
- **SMOTE** generates synthetic minority samples by interpolating between existing ones to balance class distribution.
- **ADASYN** focuses on creating more synthetic samples for harder-to-classify minority instances.
- **CondensedNearestNeighbor** reduces majority class samples to maintain decision boundaries and improve efficiency.
- **Tomek links** remove noisy or ambiguous instances in imbalanced datasets, helping to improve classification performance.

Handling multicollinearity : We removed highly correlated features to prevent multicollinearity in our dataset. We iterated through the correlation matrix and identified the feature pairs with a correlation above a threshold of 0.70. For each pair, we retained the first feature and marked the second for removal. After excluding these highly correlated features, we stored the remaining ones. We generated a heatmap to visualize the correlation among the reduced feature set which confirmed the reduction of multicollinearity as now the coloured blocks are mostly along a single line.

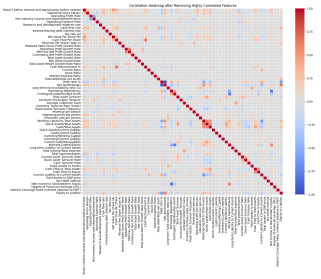

Figure 4. Heatmap of Reduced Feature Correlation

4.4. Data visualisation

Box-plots of some features of raw data

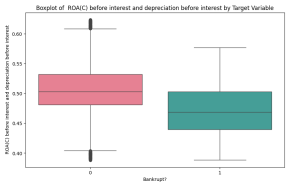

Figure 5.

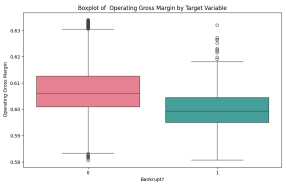

Figure 6. .

Figure 7.

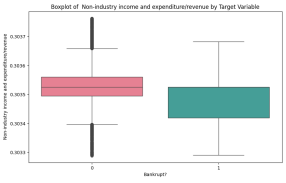

Figure 8.

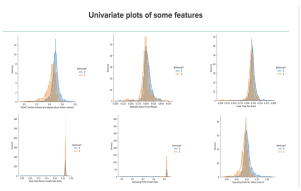

Figure 9. Univariate Plots

4.5. Feature Extraction

We used **ANOVA** and **Chi-square tests** to select the most relevant features which helped in dimentionality reduction:

- **ANOVA for Numerical Features:** We applied the ANOVA test to assess the variance of numerical features in relation to the target variable (Bankrupt?), helping us identify those that significantly influence the target. It identifies whether the means of numerical features are significantly different across the target categories helping to indicate if the feature is statistically significant and impacts the target variable. This ensures that we retain only the numerical features that have a meaningful impact on the target variable.

- **Chi-square Test for Categorical Features:** We applied the Chi-square test to evaluate the dependence of each categorical feature on the target variable. This enabled us to extract features that had the most impact on the target, further refining our feature set by excluding irrelevant or redundant features.

By combining the above two, we reduced dimensionality by retaining only the most informative features for our model.

5. Methodology

Machine Learning Design Cycle

We followed the procedure of Exploratory Data Analysis (EDA), Preprocessing, Feature Selection, and Model Training, revisiting earlier stages as needed based on the iterative Machine Learning Design Cycle.

Model Descriptions

Logistic Regression: We used Logistic Regression as a baseline model because of its simplicity. It calculates the probability of a company being bankrupt (1) or not bankrupt (0), assuming a linear relationship between the input features and the target.

Random Forest Classifier: Random Forest was selected for its robustness and ability to handle mixed data types, missing values, and outliers effectively, making it well-suited for our diverse dataset.

Gaussian Naive Bayes: We employed Naive Bayes to explore the assumption of feature independence and test its performance on normally distributed data.

Support Vector Machine (SVM): SVM was chosen for its effectiveness in high-dimensional spaces and its ability to model complex decision boundaries.

Multi-Layer Perceptron (MLP): MLP was utilized to capture non-linear relationships in the data through its deep learning architecture.

XGBoost: XGBoost was included for its efficiency and scalability, leveraging gradient boosting to optimize predictive accuracy while preventing overfitting.

6. Results and Analysis

Training Data Metrics

Table 1. Training Metrics for Different Models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Log Loss
Logistic Regression	0.7763	0.4620	0.8202	0.5911	0.8613	0.4994
Random Forest	1.0000	1.0000	1.0000	1.0000	1.0000	0.0839
Gaussian Naive Bayes	0.7663	0.4489	0.8146	0.5788	0.8711	2.3367
Decision Tree	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000
SVM (Linear Kernel)	0.7741	0.4601	0.8427	0.5952	0.8619	0.3468
SVM (RBF Kernel)	0.7785	0.4656	0.8371	0.5984	0.8629	0.3454
SVM (Poly Kernel)	0.7885	0.4792	0.8427	0.6110	0.8728	0.3368
MLP Classifier	0.8472	0.6493	0.4888	0.5577	0.8712	0.3379
ANN (Custom)	0.8427	0.6475	0.4438	0.5267	0.8722	0.3365
XGBoost	1.0000	1.0000	1.0000	1.0000	1.0000	0.0107

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Log Loss
Logistic Regression	0.8053	0.4868	0.8810	0.6271	0.8763	0.4893
Random Forest	0.8938	0.7368	0.6667	0.7000	0.9109	0.2816
Gaussian Naive Bayes	0.7920	0.4684	0.8810	0.6116	0.8480	2.3010
Decision Tree	0.8097	0.4909	0.6429	0.5567	0.7453	6.8579
SVM (Linear Kernel)	0.8009	0.4810	0.9048	0.6281	0.8813	0.3250
SVM (RBF Kernel)	0.8009	0.4810	0.9048	0.6281	0.8815	0.3239
SVM (Poly Kernel)	0.8097	0.4935	0.9048	0.6387	0.8863	0.3179
MLP Classifier	0.8540	0.6364	0.5000	0.5600	0.8888	0.3177
ANN (Custom)	0.8407	0.6250	0.3571	0.4545	0.8868	0.3186
XGBoost	0.8850	0.6905	0.6905	0.6905	0.9076	0.3853

Table 2. Test Metrics for Different Models

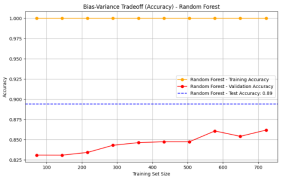

Figure 10. Bias-Variance Tradeoff (Accuracy) -Random Forest

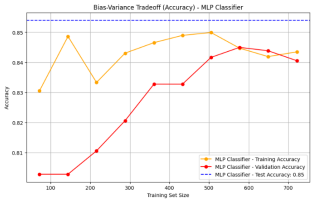

Figure 11. Bias-Variance Tradeoff (Accuracy)- Multi Layer Perceptron

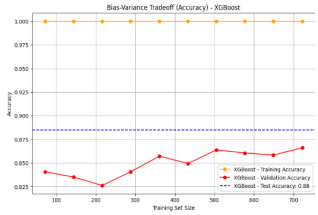

Figure 12. Bias-Variance Tradeoff (Accuracy)- XGboost

Observations

- **Preprocessing:**
  - Condensed Nearest Neighbors led to better results than other sampling methods due to extreme imbalance towards the minority class.
  - Principal Component Analysis did not prove to be very effective, rather SelectKBest was a better feature selection technique.
- **Logistic Regression:**
  - The model achieved a training accuracy of 77.63% and a test accuracy of 80.97%,
  - The recall score (0.8809) suggests that it is effective in identifying positive samples, though the precision (0.4933) indicates a higher false positive rate.
  - The F1 score on the test set (0.6325) indicates a balance between precision and recall.
- **Random Forest:**
  - The model exhibits perfect training accuracy (100%), but there was a drop to 89.82% on the test set, which suggests slight overfitting
  - It achieved a precision score of 0.7879 on the test set, indicating a strong ability to classify positive samples correctly.
  - The F1 score (0.6933) indicates a reasonable balance between precision and recall
- **Gaussian Naive Bayes:**
  - This model showed a training accuracy of 76.19% and a matching test accuracy of 80.97%, indicating good generalization.
  - The recall score (0.8809) is notably high, while the precision (0.4933) points to potential issues with false positive
  - The consistent F1 score (0.6325) across training and testing suggests a stable performance.
  - **SVM (Linear Kernel):**
    - Test accuracy of 80.09% demonstrates moderate generalization capability.
    - Recall (0.9048) is high, showing strong ability to capture positive cases, but precision (0.4810) highlights issues with false positives.
    - The F1 score (0.6281) indicates a fair trade-off between precision and recall.
  - **SVM (RBF Kernel):**
    - The model achieved a test accuracy of 80.09%, comparable to the Linear Kernel.
    - A high recall (0.9048) indicates good identification of positive samples, while precision (0.4810) shows false-positive concerns.
    - The F1 score (0.6281) reflects balanced yet moderate performance.
  - **SVM (Poly Kernel):**
    - The test accuracy of 80.97% is slightly better than other SVM models.
    - Precision (0.4935) and recall (0.9048) are improved compared to the Linear and RBF Kernels, reducing false positives slightly.
    - The F1 score (0.6387) shows better overall balance between precision and recall among SVM variants.
  - **MLP Classifier:**
    - The model achieved a test accuracy of 85.40%, higher than many other models.
    - Precision (0.5000) and recall (0.5600) are moderate
  - **XGBoost:**
    - The test accuracy of 88.50% is among the highest across models.
    - Balanced precision and recall scores (0.6905 each) reflect consistent identification of positive cases and low false positives.
    - The F1 score (0.6905) shows good performance

Since our use case was predicting whether a company will go bankrupt or not, false negatives (failing to predict a bankruptcy) are more critical than false positives (incorrectly predicting a company will go bankrupt). Thus, it is acceptable to have some false positives as long as the false negatives are minimized.

Based on these criteria, the Random Forest, XGBoost, and MLP Classifier models were determined to be the best-performing models due to their higher recall and balanced metrics for our use case.

Confusion Matrices for Selected Models

MLP Classifier	Predicted: No	Predicted: Yes
Actual: No	172	12
Actual: Yes	21	21

Table 3. Confusion Matrix for MLP Classifier

XGBoost	Predicted: No	Predicted: Yes
Actual: No	171	13
Actual: Yes	13	29

Table 4. Confusion Matrix for XGBoost

Random Forest	Predicted: No	Predicted: Yes
Actual: No	174	10
Actual: Yes	14	28

Table 5. Confusion Matrix for Random Forest

7. Conclusion

7.1. Learning from Project

- **Important Pre-processing steps** - Realized the effect of outliers on modeling, handling imbalance in dataset and reducing multicollinearity.
- Importance of different sampling methods and their effects on the results such as simple methods compares to method like smote and adasyn.
- Learnt about the ANOVA and chi-square test for feature selection.
- **Classification Models** - Gained practical skills in classification-based machine learning tasks.
- The importance of dimensionality reduction by careful feature selection.
- The motivation to explore additional models for achieving better results.

7.2. Timeline

We were able to adhere to the previously mentioned timeline without any major deviance.

7.3. Member Contributions

- **Rishi:** EDA, Preprocessing, Feature Selection, Model Training, Documentation
- **Palak:** EDA, Feature Selection, Documentation
- **Yashovardhan:** Literature Survey, Model Training, Presentation.
- **Kuber:** EDA, Preprocessing, Model Training, Presentation

References

[1] Nanxi Wang, "Selecting bankruptcy predictors using a support vector machine approach," *Journal of Mathematical Finance*, Vol.7 No.4, November 17, 2017.

[2] Flavio Barboza, Herbert Kimura, Edward Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 405-417, 2017.

[3] Nanxi Wang, "Bankruptcy Prediction Using Machine Learning," *Journal of Mathematical Finance*, Vol.7 No.4, November 17, 2017.

[4] Konstantin A. Danilov, "Corporate Bankruptcy: Assessment, Analysis and Prediction of Financial Distress, Insolvency, and Failure," MIT Sloan School of Management, May 9, 2014.

[5] *Company Bankruptcy Dataset*