Bankruptcy Prediction Using Financial Ratios and Machine Learning

April 18, 2025

Contents

1 Introduction **3**
- 1.1 Motivation 3
- 1.2 Objectives 3
- 1.3 Report Structure 3

2 Class Imbalance and Its Impact on Model Performance **4**

3 Data Description and Feature Engineering **5**
- 3.1 Raw Dataset 5
- 3.2 Target Label 5
- 3.3 Derived Features 6
- 3.4 Final Feature Set 6

4 Model Performance Challenges **6**

5 Exploratory Data Analysis **7**
- 5.1 Target Distribution 7

6 Class Imbalance and Resampling **7**
- 6.1 Challenges of Imbalance 7
- 6.2 Under-Sampling 7
- 6.3 SMOTE (Optional) 8

7 Modeling Methodology **8**
- 7.1 Train/Test Split 8
- 7.2 Models Evaluated 8
- 7.3 Evaluation Metrics 8

8 Experimental Results **9**
- 8.1 Decision Tree 9

8.2 Random Forest and XGBoost 10

9 Discussion and Future Work 11
9.1 Trade-Offs 11
9.2 Future Directions 11

10 Conclusion and Future Work 11
10.1 Future Work 12

A Appendix A: Full Feature List 13

B Appendix B: Hyperparameter Grids 13

1 Introduction

Predicting corporate bankruptcy is critical for investors, regulators, and credit agencies. A timely warning can prevent losses, allocate credit more safely, and guide regulatory action.

1.1 Motivation

The banking sector and capital markets rely heavily on accurate risk assessment. Traditional statistical models (e.g., logistic regression on Altman's Z-score) have shown promise but often fail in highly dynamic market conditions. Machine learning (ML) methods, with their ability to capture nonlinear patterns, offer a compelling alternative.

1.2 Objectives

- Engineer financial ratios over a five-year window as predictive features.
- Address severe class imbalance using resampling and class-weighting.
- Compare models (Logistic Regression, Decision Tree, Random Forest, XGBoost).
- Optimize recall for the minority (bankrupt) class while maintaining reasonable precision.

1.3 Report Structure

This report is organized as follows:

1. Data Description and Feature Engineering
2. Exploratory Data Analysis
3. Class Imbalance and Resampling
4. Modeling Methodology
5. Experimental Results
6. Discussion and Future Work
7. Conclusion
8. References
9. Appendix

2 Class Imbalance and Its Impact on Model Performance

In our bankruptcy prediction task, we manually labeled companies as "bankrupt" based on their suspension or delisting status, using this as a proxy for financial failure. All remaining companies were labeled as non-bankrupt. While this allowed us to create a labeled dataset, it introduced a significant class imbalance.

Observations

- **Original number of companies:** ∼11,000
- **After cleaning:** ∼7,400 companies
- **Companies labeled as bankrupt (suspended/delisted):** 140
- **Companies labeled as non-bankrupt:** ∼7,260
- **Class ratio (bankrupt : non-bankrupt):** ∼1:52

Such class imbalance is a common challenge in real-world fraud or risk detection problems, where the positive cases (in this case, bankruptcies) are naturally rare.

Consequences of Imbalanced Data

- **Misleading Accuracy:** A model that predicts all companies as non-bankrupt can still achieve over 98% accuracy while being completely ineffective at identifying bankruptcies.
- **Low Recall for Minority Class:** The model is likely to miss most bankrupt companies due to the overwhelming presence of the non-bankrupt class.
- **Skewed Metrics:** The F1-score for the bankrupt class remains low, indicating poor detection capability.

Addressing the Imbalance

To improve the model's ability to learn from the minority class, we applied various strategies:

- **Class Balancing Techniques:**
  - **Oversampling:** Using SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic examples of the minority class.
  - **Undersampling:** Randomly removing examples from the majority class to balance the dataset.
- **Model-Level Adjustments:**

  - Using models such as Logistic Regression, SVM, and Random Forest with `class_weight='balanced'` to automatically account for class imbalance.

- **Metric Focus:** We shifted the evaluation from overall accuracy to more meaningful metrics such as:
  - Recall (Sensitivity) for the bankrupt class
  - Precision and F1-Score
  - ROC-AUC Score

Key Takeaways

- **Accuracy is not sufficient** when the data is highly imbalanced.
- **The minority class is more important**: In bankruptcy prediction, detecting those 140 at-risk companies is more critical than correctly identifying the rest.
- **Better target labels** (such as actual bankruptcy filings) could improve model performance and reliability.
- **Feature engineering** that highlights financial distress indicators (e.g., declining profit, increasing liabilities) plays a crucial role.

3 Data Description and Feature Engineering

3.1 Raw Dataset

Our raw dataset comprises annual financial statements for 7,412 companies, spanning 10 years. Key fields include:

- `Company Name`, `Year`
- Total Income, Total Expenses, Profit After Tax, Total Liabilities, Total Assets
- Suspension Date, Delisting Date

3.2 Target Label

We define:

$$\text{Bankrupt} = \begin{cases} 1 & \text{if Suspended Date or Delisted Date is not missing,} \\ 0 & \text{otherwise.} \end{cases}$$

This yields 7,282 non-bankrupt and 130 bankrupt observations, a ratio of 56:1.

3.3 Derived Features

We compute 5-year averages and financial ratios:

$$\begin{aligned}
\text{Profit_margin} &= \frac{\text{Profit after tax}}{\text{Total income}}, \\
\text{Expense_ratio} &= \frac{\text{Total expenses}}{\text{Total income}}, \\
\text{ROA} &= \frac{\text{Profit after tax}}{\text{Total assets}}, \\
\text{Asset_turnover} &= \frac{\text{Total income}}{\text{Total assets}}, \ldots
\end{aligned}$$

Binary flags such as `is_negative_profit` capture extreme cases.

3.4 Final Feature Set

Feature	Description
Profit_margin	Profit after tax / Total income
ROA	Return on assets
Liability_to_income	Total liabilities / Total income
is_negative_profit	Indicator (profit ¡ 0)
...	...

Table 1: Engineered financial features (total 14).

4 Model Performance Challenges

In our attempt to predict corporate bankruptcy, we experimented with various machine learning models including Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Random Forest. However, the performance of most models remained unsatisfactory due to the extreme class imbalance present in our dataset. Out of approximately 7400 companies, only around 140 were identified as bankrupt. This imbalance, coupled with missing financial values and limited indicators of financial distress, significantly hindered model learning and generalization.

Traditional classifiers such as Logistic Regression and Naive Bayes failed to capture the minority class patterns, resulting in high accuracy but extremely low recall and precision for the bankrupt class. This demonstrates the challenge of relying solely on accuracy in imbalanced datasets.

Among the models tried, Decision Tree and Random Forest classifiers showed comparatively better performance. These models are capable of handling non-linear feature interactions and tend to perform well on imbalanced datasets without requiring heavy scaling or transformation. They achieved slightly improved recall and F1-scores for the bankrupt class, although the overall performance still leaves room for improvement.

We acknowledge that further improvements might be possible through advanced imbalance handling techniques (e.g., SMOTE, class weighting, anomaly detection) or by incorporating more informative features related to market behavior, governance, or macroeconomic indicators.

5 Exploratory Data Analysis

5.1 Target Distribution

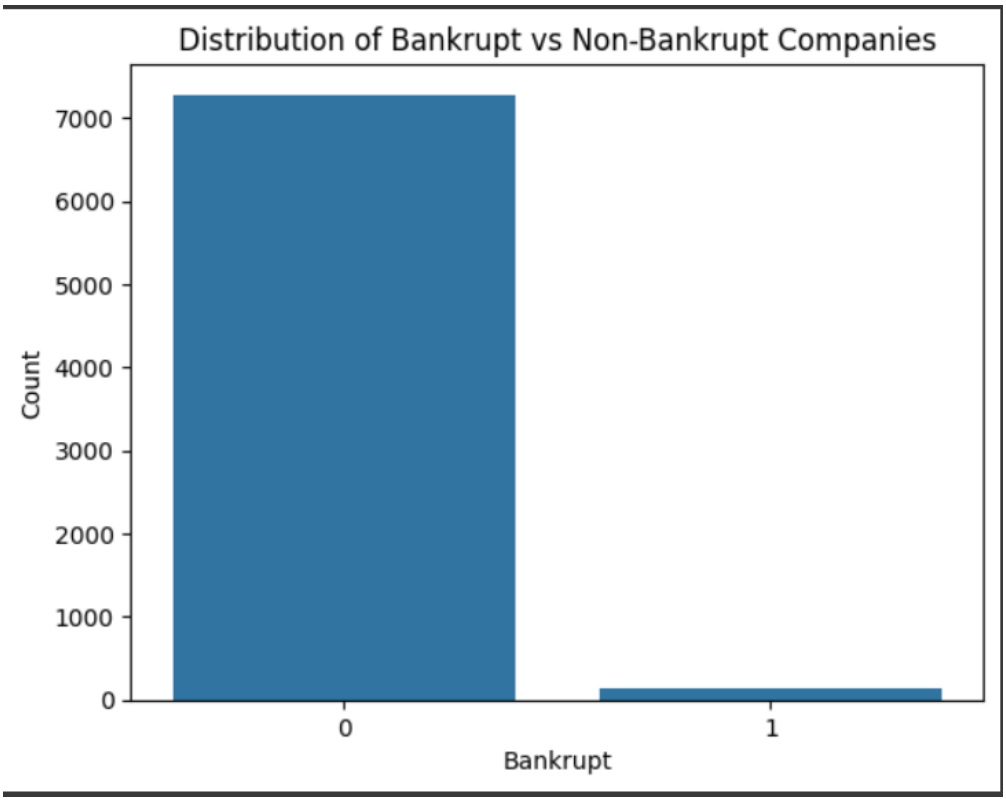

Figure 1: Imbalanced target: 7,282 non-bankrupt vs. 130 bankrupt.

6 Class Imbalance and Resampling

6.1 Challenges of Imbalance

Severe skew (ratio $\approx 56:1$) can lead to models always predicting the majority class.

6.2 Under-Sampling

We randomly sample 130 non-bankrupt cases:

```
legit = df_feat[df_feat.Bankrupt==0]
bankrupt = df_feat[df_feat.Bankrupt==1]
legit_sample = legit.sample(n=130, random_state=42)
new_dataset = pd.concat([legit_sample, bankrupt], axis=0).sample(
   frac=1, random_state=42)
```

Resulting in 260 balanced observations.

6.3 SMOTE (Optional)

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state=42)
X_res, y_res = sm.fit_resample(X, y)
```

7 Modeling Methodology

7.1 Train/Test Split

```
from sklearn.model_selection import train_test_split
X = new_dataset.drop(columns=['Bankrupt','Suspended␣Date','Delisted␣
   Date'])
y = new_dataset.Bankrupt
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=2)
```

7.2 Models Evaluated

- **Logistic Regression** (baseline)
- **Decision Tree** (tuned with class weights)
- **Random Forest** (balanced)
- **XGBoost** (with scale_pos_weight)
- **Stacking Ensemble**

7.3 Evaluation Metrics

We report:

$$\text{Accuracy}, \quad \text{Precision}, \quad \text{Recall}, \quad F_1\text{-score},$$

8 Experimental Results

8.1 Decision Tree

```
dt = DecisionTreeClassifier(
    max_depth=5, min_samples_leaf=5,
    class_weight='balanced', random_state=42)
```

	Precision	Recall	F1-score	Accuracy
Non-Bankrupt	0.73	0.69	0.70	0.60
Bankrupt	0.37	0.42	0.39	

Table 2: Decision Tree performance

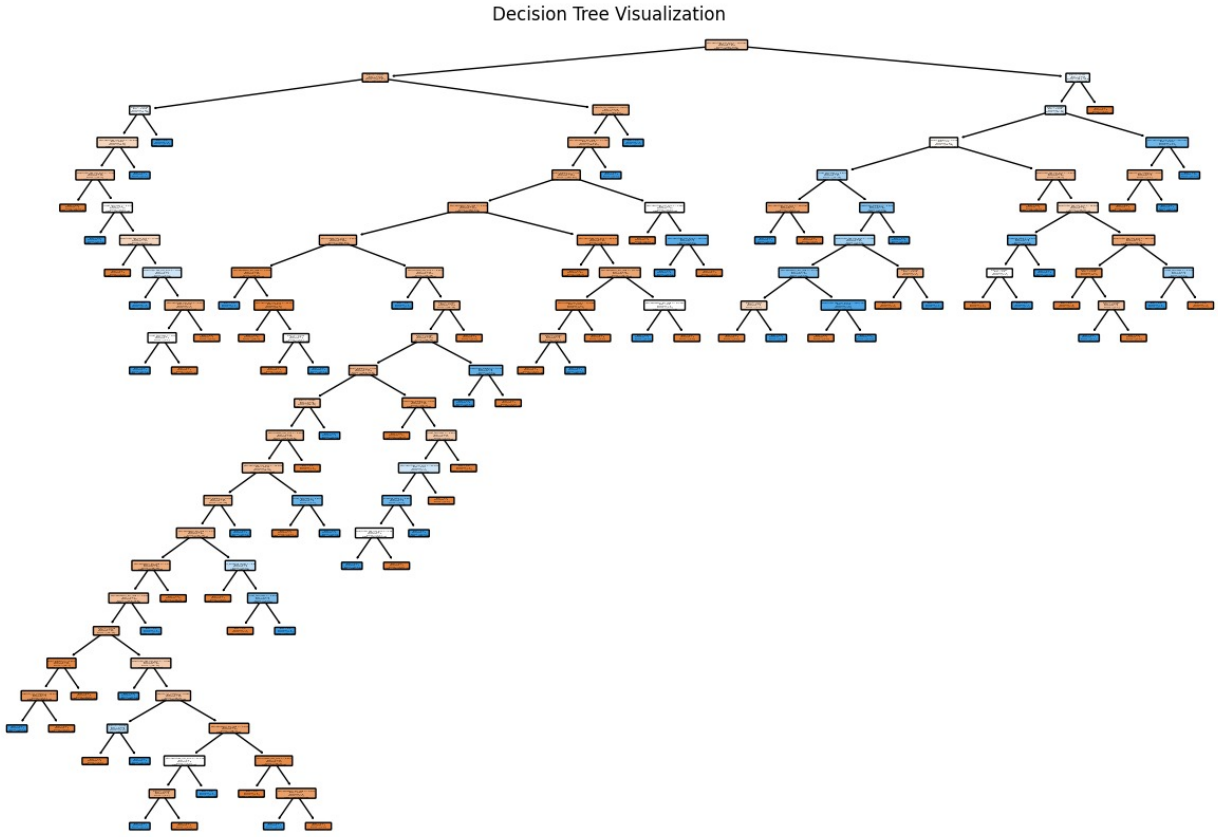

Figure 2: Decision Tree Visualization.

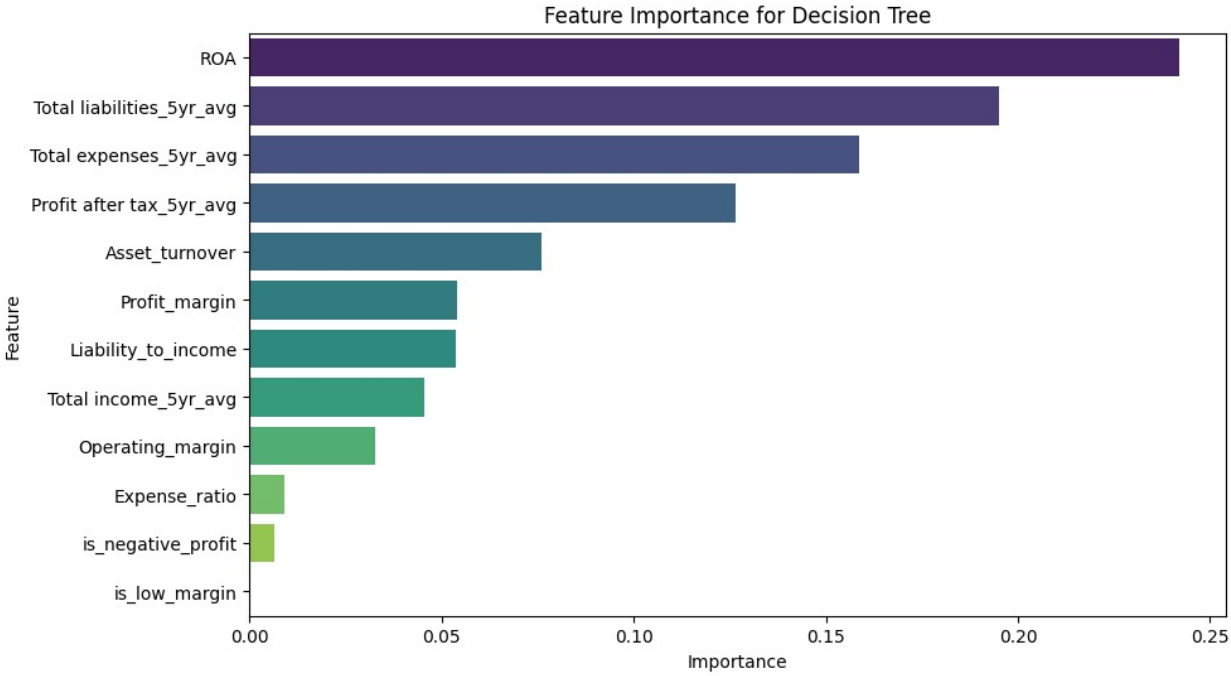

Figure 3: Features Importance.

8.2 Random Forest and XGBoost

- Random Forest (100 trees, class_weight='balanced'): Recall 0.45, Precision 0.36
- XGBoost (scale_pos_weight=56): Recall 0.25, Precision 0.30

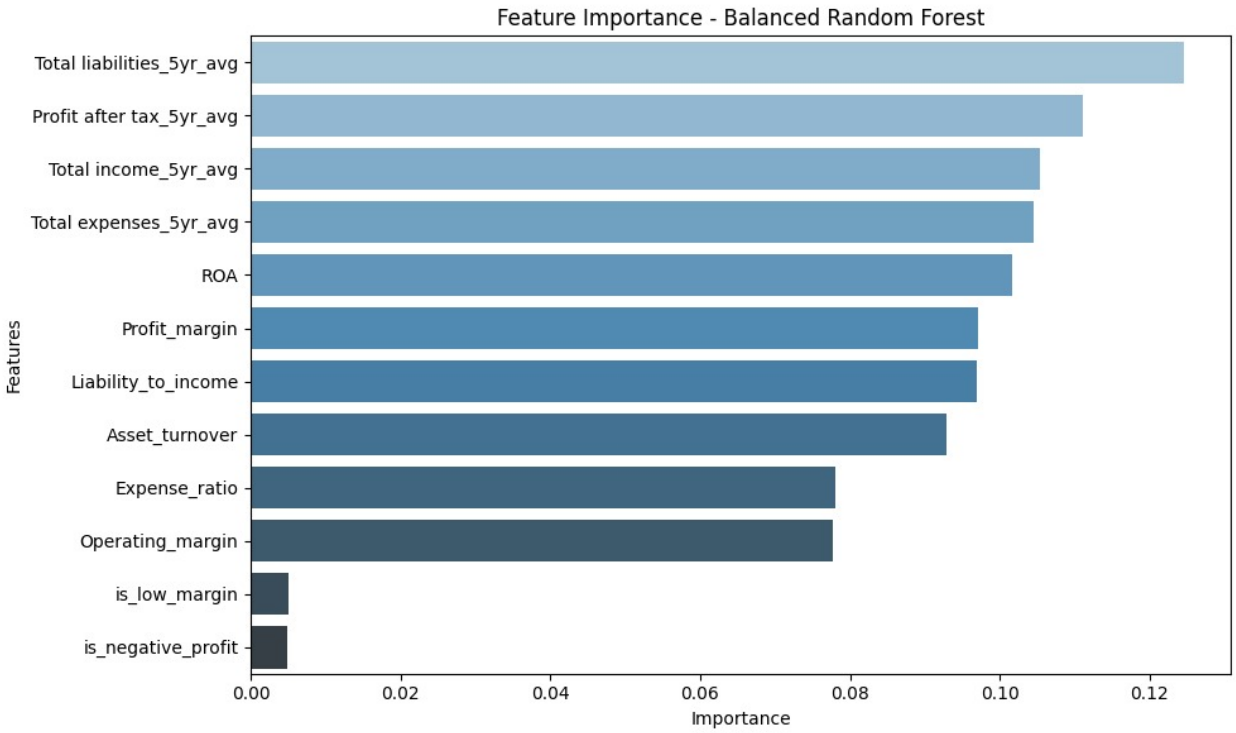

Figure 4: Feature Importance of Random Forest.

9 Discussion and Future Work

9.1 Trade-Offs

- Lower decision thresholds and oversampling improve recall but can reduce precision. - Tree-based models outperform linear ones in capturing nonlinear interactions.

9.2 Future Directions

- Hyperparameter tuning via `GridSearchCV` or Bayesian optimization.
- Ensemble stacking with meta-learner.
- Incorporate text or macroeconomic features.
- Evaluate on a rolling-window out-of-time test.

10 Conclusion and Future Work

In this project, we attempted to build a bankruptcy prediction system based on financial ratios and metrics derived from ProwessIQ data. After extensive data cleaning and processing, we generated a dataset of approximately 7400 companies, out of which only around 140 were

labeled as bankrupt. This significant class imbalance heavily impacted the performance of standard machine learning models.

Although various algorithms were explored—including K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Naive Bayes, Decision Tree, and Random Forest—most models struggled to effectively identify bankrupt companies. Among them, Decision Tree and Random Forest showed slightly better performance in terms of recall and balanced accuracy, but the results were still far from satisfactory.

10.1 Future Work

Future efforts could focus on the following areas to improve predictive accuracy:

- Implement resampling techniques such as SMOTE or ADASYN to address class imbalance.
- Explore advanced algorithms like XGBoost, LightGBM, or neural networks.
- Enhance feature engineering by incorporating domain knowledge and time-series trends.
- Apply cost-sensitive learning to prioritize the correct prediction of bankrupt companies.
- Expand the dataset to include more instances of bankrupt companies for better representation.

By addressing these areas, we expect the performance of bankruptcy prediction models to improve in both recall and overall robustness.

References

1. Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, 23(4), 589–609.
2. Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
3. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD.*
4. Research papers given by you.

A Appendix A: Full Feature List

Feature	Description
Profit_margin	Profit after tax / Total income
Expense_ratio	Total expenses / Total income
Asset_turnover	Total income / Total assets
ROA	Profit after tax / Total assets
Operating_margin	(Income – Expenses) / Income
Liability_to_income	Total liabilities / Total income
is_negative_profit	Profit ¡ 0 flag
is_low_margin	Profit_margin ¡ 0.05
...	...

B Appendix B: Hyperparameter Grids

```
param_grid = {
  'max_depth': [3,5,7],
  'min_samples_leaf': [1,5,10],
  'class_weight': ['balanced', {0:1,1:3}]
}
```