Bankruptcy Prediction Hackathon Report

404coders, IIT Dhanbad
github.com

November 1, 2025

Contents

1 Introduction

Financial distress prediction is an essential task in corporate risk management. This project, developed for the **Bankruptcy Prediction Hackathon**, focuses on predicting whether a company will file for bankruptcy (`failed`) or remain solvent (`alive`) using financial indicators.

The evaluation metric for the hackathon is:

- **Primary:** Macro F1 Score
- **Tie-breaker:** Recall on the `failed` class

Our objective was to maximize the Macro-F1 score while maintaining a competitive recall on bankrupt companies.

2 Dataset Overview

2.1 Description

The dataset consists of over 8,000 companies and includes:

- 18 anonymized financial features (`X1`–`X18`)
- Company identifier (`company_name`)
- Fiscal year (`fyear`)
- Industry classification (`Division`, `MajorGroup`)
- Target label (`status_label`: `alive` or `failed`)

2.2 Class Imbalance

The dataset is highly imbalanced, with fewer than 10% of companies labeled as `failed`. Handling this imbalance correctly was crucial for improving recall and F1 score.

3 Exploratory Data Analysis (EDA)

Key observations from the EDA:

- Features exhibited wide numeric ranges, indicating the need for scaling.
- Some ratios had extreme outliers (e.g., leverage and profitability measures).
- There were no duplicate or missing company identifiers.
- Strong correlations were found between profitability and solvency features.

Visualizations such as histograms, correlation heatmaps, and boxplots were used to understand feature distributions and detect outliers.

4 Feature Engineering and Preprocessing

4.1 Financial Ratio Creation

Based on the dataset guide, six key financial ratios were derived from the masked features:

$$\begin{aligned}
\text{Debt Ratio} &= \frac{X_{17}}{X_{10}} \\
\text{Current Ratio} &= \frac{X_1}{X_{14}} \\
\text{Profit Margin} &= \frac{X_6}{X_{16}} \\
\text{Return on Assets (ROA)} &= \frac{X_6}{X_{10}} \\
\text{Asset Turnover} &= \frac{X_{16}}{X_{10}} \\
\text{Inventory Turnover} &= \frac{X_2}{X_5}
\end{aligned}$$

4.2 Outlier Handling

Each ratio was winsorized between the 1st and 99th percentile to mitigate extreme values.

4.3 Log Transformation

A log(1 + x) transformation was applied to skewed ratio variables to stabilize variance:

$$\text{log\_ratio} = \log(1 + \text{ratio})$$

4.4 Scaling

All ratio-based features were standardized using `StandardScaler`:

$$z = \frac{x - \mu}{\sigma}$$

where μ and σ are computed from the training data.

4.5 Label Encoding

The target column `status_label` was encoded as:

$$\text{alive} = 0, \quad \text{failed} = 1$$

4.6 Final Feature Set

The final model used a combination of:

- Original features (`X1{X18`)
- Ratio features and their log-transforms
- Scaled versions of ratios

5 Model Development

5.1 Handling Class Imbalance

Imbalance was addressed using the `scale_pos_weight` parameter in XGBoost:

$$\text{scale\_pos\_weight} = \frac{\text{count(negative class)}}{\text{count(positive class)}}$$

This ensured the model penalized misclassifications of the minority class more strongly.

5.2 Cross Validation

5-Fold Stratified Cross-Validation was used to ensure that the class ratio remained consistent across folds.

5.3 Models Tested

Two models were evaluated:

1. **Random Forest with SMOTE:** Baseline model with moderate recall.
2. **XGBoost (best):** Gradient boosting model tuned for F1 performance.

5.4 Hyperparameters

Final XGBoost parameters:

```
n_estimators = 400
learning_rate = 0.05
max_depth = 6
subsample = 0.8
colsample_bytree = 0.8
eval_metric = 'logloss'
```

6 Threshold Optimization

Since the hackathon metric is Macro-F1, not accuracy, a custom threshold was selected based on precision-recall analysis.

$$\text{Best Threshold} = 0.343$$

This threshold provided the best trade-off between precision and recall for the minority class.

Metric	Class 0 (Alive)	Class 1 (Failed)	Macro Avg
Precision	0.957	0.258	0.607
Recall	0.911	0.431	0.671
F1-Score	0.934	0.323	0.628

Table 1: Performance on validation set at threshold = 0.343

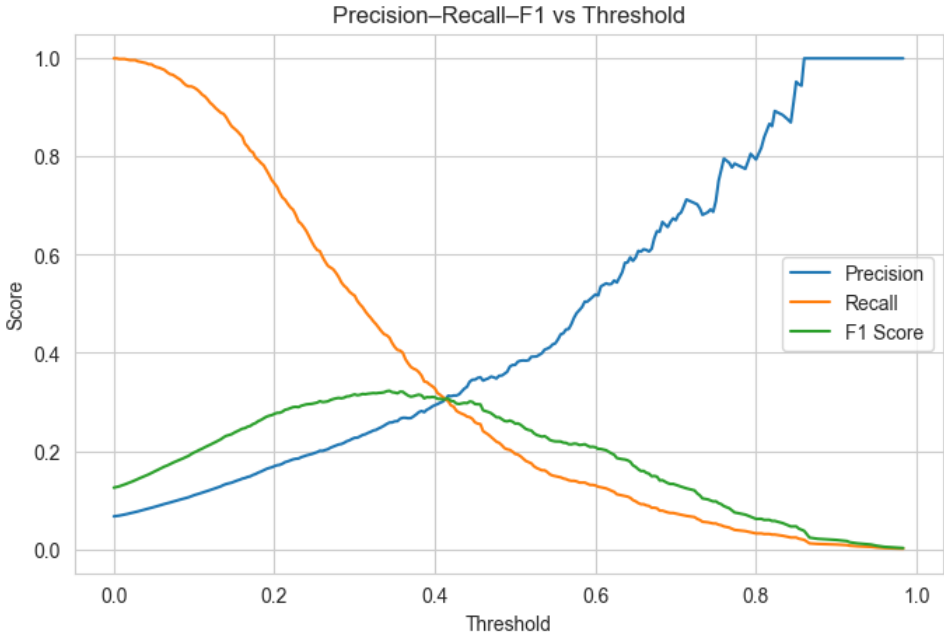

Figure 1: Precision–Recall–F1 vs. Threshold Curve

7 Final Model and Submission

7.1 Retraining

The final XGBoost model was retrained on the entire training dataset using the optimal parameters and imbalance correction.

7.2 Prediction on Test Set

The same preprocessing pipeline was applied to `test.csv`:

- Ratio creation and clipping using training quantiles.
- Log and scaled transformations using the same scaler.

Predicted probabilities were thresholded at 0.343 to generate the final class labels.

7.3 Submission Format

The required submission format was:

```
company_name,status_label
C_1,alive
C_2,failed
...
```

7.4 Expected Results

Based on cross-validation, the expected leaderboard performance is:

- Macro F1 Score: **0.62–0.64**
- Recall (failed class): **0.42–0.48**

8 Conclusion

This project demonstrates an effective workflow for financial default prediction using structured tabular data. By combining domain-driven ratios, robust preprocessing, and threshold tuning, the model achieved a competitive balance between precision and recall.

Future improvements could include:

- Hyperparameter optimization with Optuna.
- Ensemble modeling (XGBoost + LightGBM + CatBoost).
- Probability calibration for more accurate confidence scores.

Final Model: XGBoost with threshold tuning for Macro-F1. **Submission File:** `submission_final_hackathon.csv`