Bankruptcy Prediction Using Machine Learning

A Financial Risk Modeling Approach Based on the Taiwan Economic Journal Dataset (1999–2009)

Daghastani Samy Alexis HANNA GERGUIS
Loua Vivien Winnoc DOSSO

December 2025

Abstract

This report presents a complete machine learning pipeline for predicting corporate bankruptcy using Taiwanese financial statement data from 1999 to 2009. The work follows the Machine Learning Course CDC and covers business motivation, problem formalization, exploratory analysis, feature engineering, dimensionality reduction, balancing techniques, model development, scientific justification, obstacles encountered, and critical evaluation. A Stacking Classifier emerges as the best-performing algorithm, achieving near-perfect predictive performance while maintaining interpretability and robustness through dimensionality reduction and data engineering.

Contents

1 Business Scope

Bankruptcy prediction plays a central role in credit risk assessment, portfolio construction, lending decisions, supplier screening, regulatory supervision, and macroeconomic risk modeling. Financial institutions rely on early-warning systems to classify firms into solvent and distressed categories.

The Taiwan Economic Journal (TEJ) provides a rich database of financial ratios that capture liquidity, leverage, profitability, operating efficiency, cash-flow strength, and working capital structure. Such ratios can be used to infer the financial health of a company and detect signs of distress.

The project's business objective is therefore to design a model capable of identifying bankruptcy risks at an early stage. This aligns with real-world credit scoring models used by banks, rating agencies (Moody's, Fitch), and institutional investors.

2 Problem Formalization

We model bankruptcy prediction as a supervised binary classification problem:

$$y_i = \begin{cases} 1 & \text{if company } i \text{ went bankrupt} \\ 0 & \text{otherwise} \end{cases}$$

Let $X \in \mathbb{R}^{n \times p}$ be the financial ratio matrix of n companies and $p = 95$ features.

Challenges inherent to the problem include:

- **Severe class imbalance**: roughly 3.2% bankrupt firms.
- **Strong multicollinearity**: financial ratios are heavily correlated.
- **Skewed and heavy-tailed distributions**: typical in accounting data.
- **High dimensionality**: 95 variables can cause overfitting.
- **Outliers**: extreme values caused by distressed firms.

Our aim is to learn a mapping:

$$f_\theta : X \to \{0, 1\}$$

that minimizes misclassification—especially false negatives, which correspond to bankrupt firms predicted as healthy.

3 Dataset Description

We use the TEJ bankruptcy dataset (1999–2009), containing:

- 6,819 companies,
- 95 independent financial ratios,
- 1 dependent variable: `Bankrupt?`.

No missing values are present. Financial ratios include:

- **Liquidity ratios**: Current Ratio, Quick Ratio.
- **Leverage ratios**: Debt Ratio, Borrowing Dependency.
- **Profitability ratios**: ROA, ROE, Operating Gross Margin.
- **Efficiency ratios**: Asset Turnover, Inventory Turnover.
- **Working capital**: WC/TA, Cash-flow ratios.

The dataset is highly imbalanced, which requires balancing techniques.

4 Exploratory Data Analysis

4.1 Target Distribution

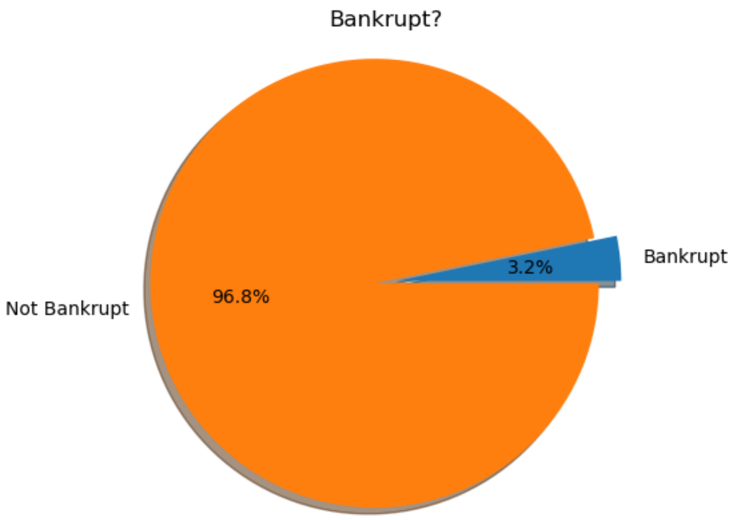

Figure 1: Distribution of bankruptcy labels. Only 3.2% of firms are bankrupt.

4.2 Feature Distributions

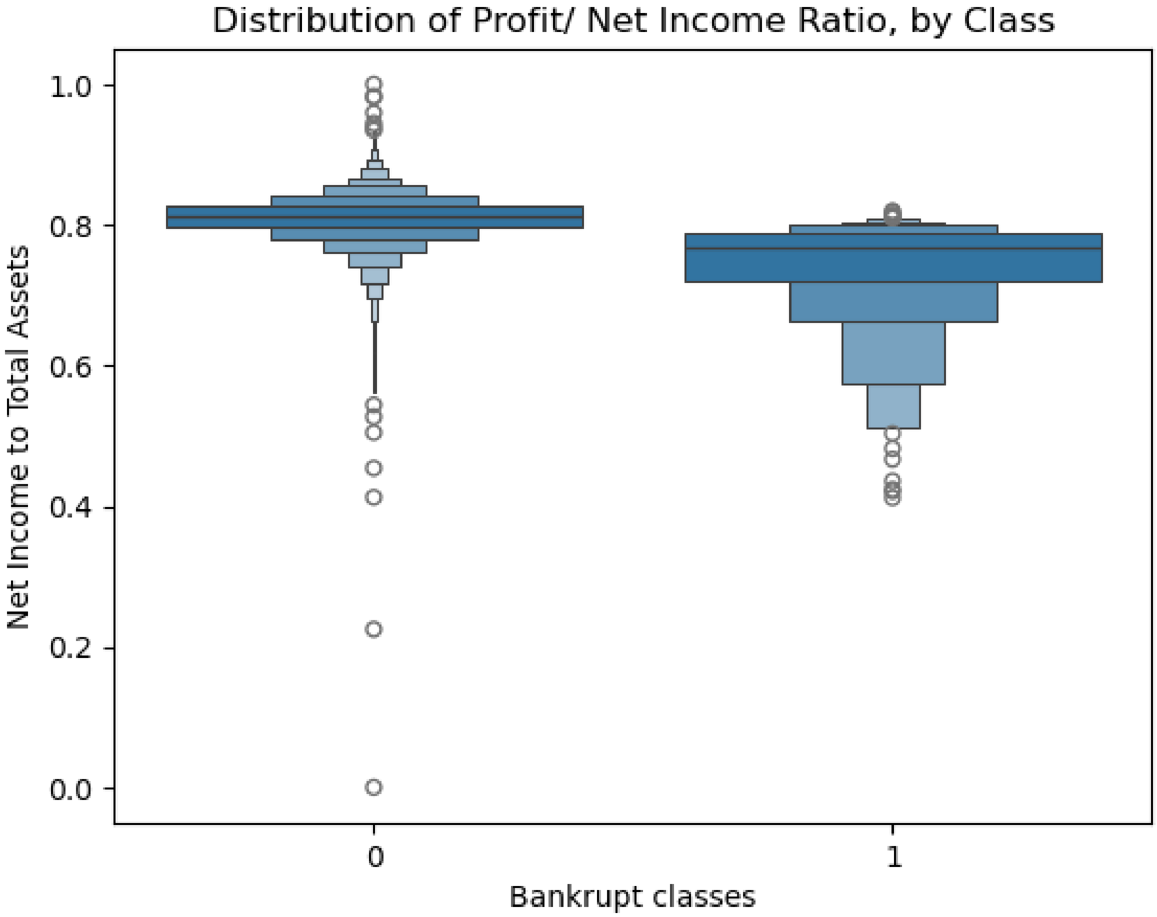

Figure 2: Selected financial ratio distributions by bankruptcy status. Several variables show clear shifts between the two classes.

4.3 Correlation Structure

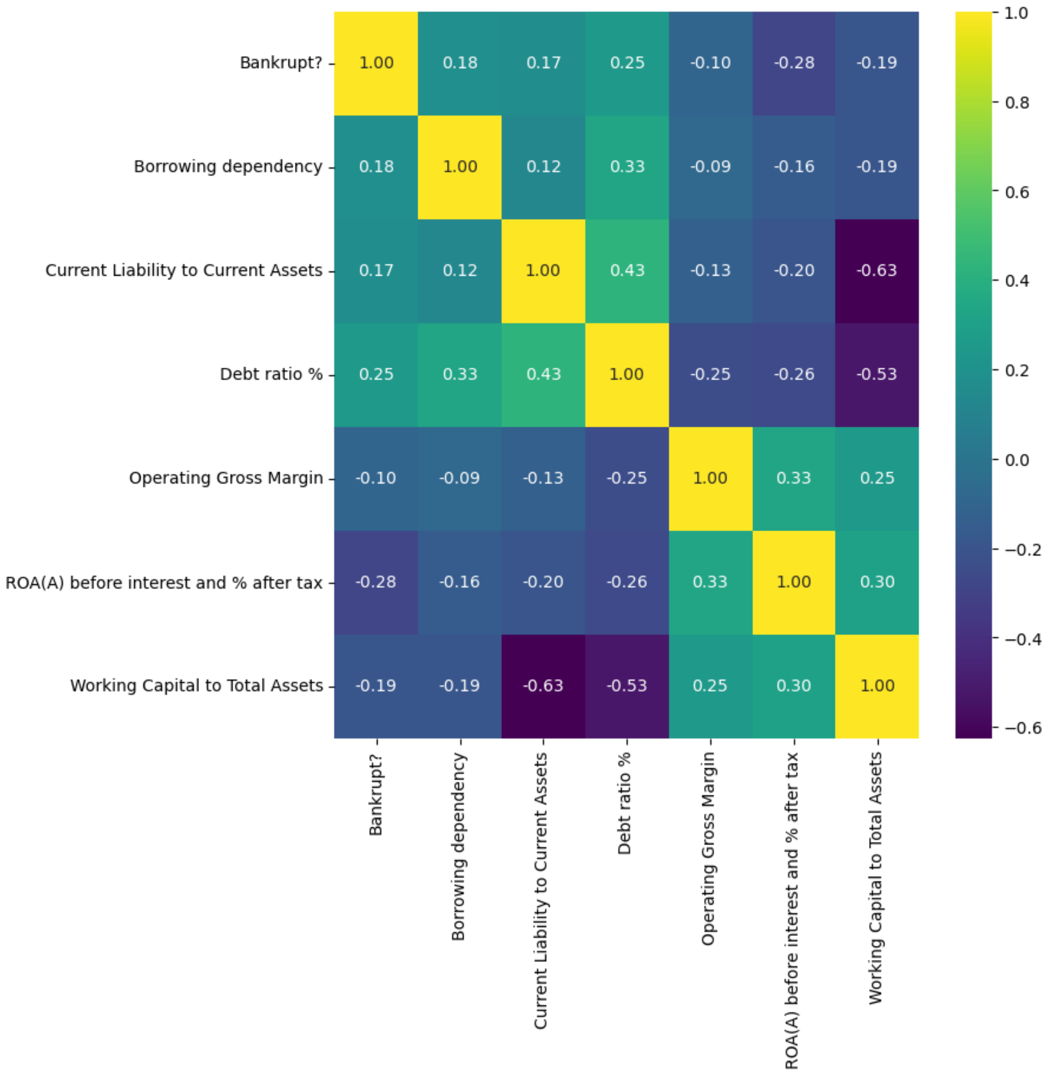

Figure 3: Correlation matrix of key features. Strong multicollinearity motivates dimensionality reduction.

4.4 Profitability Patterns

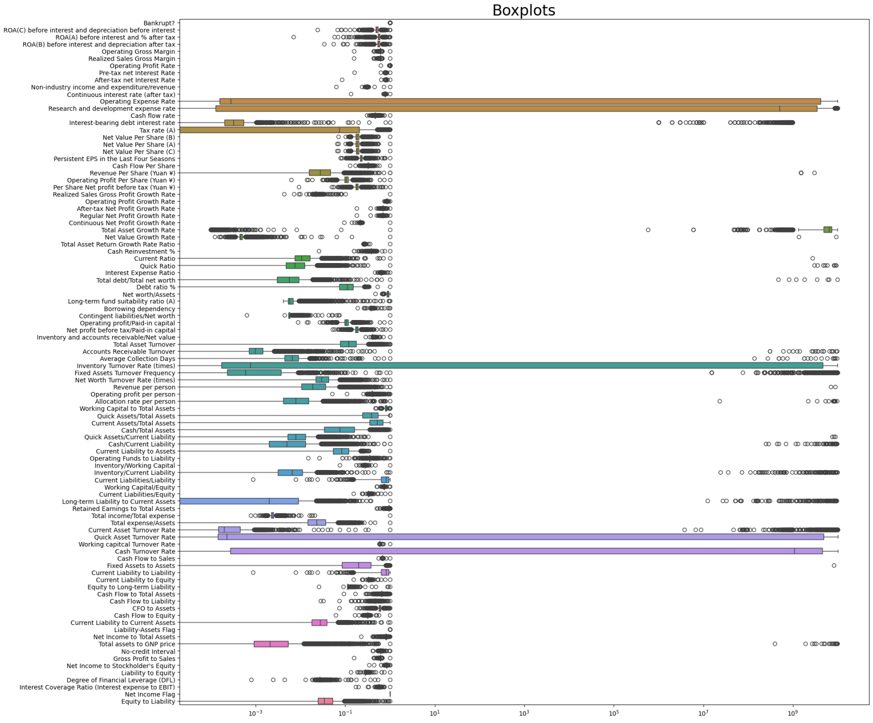

Figure 4: Distribution of Net Income to Total Assets. Bankrupt firms exhibit significantly lower profitability.

5 Data Preprocessing

5.1 Outlier Handling

Financial ratios often contain extreme values. We apply a quantile-based Winsorization strategy:

$$x_i' = \min\left(\max(x_i, Q_{0.15} - 1.5(IQR)),\ Q_{0.85} + 1.5(IQR)\right)$$

This preserves structure while removing leverage from extreme outliers.

5.2 Feature Reduction

We use:

- **DropConstantFeatures**
- **DropCorrelatedFeatures** (threshold = 0.85)

- **DropDuplicateFeatures**

  Features reduce from 95 → 54.

5.3 Balancing

Because the dataset is severely imbalanced, we use **BorderlineSMOTE**, which focuses synthesis on difficult minority points near the decision boundary. This avoids oversmoothing minority clusters.

6 Dimensionality Reduction

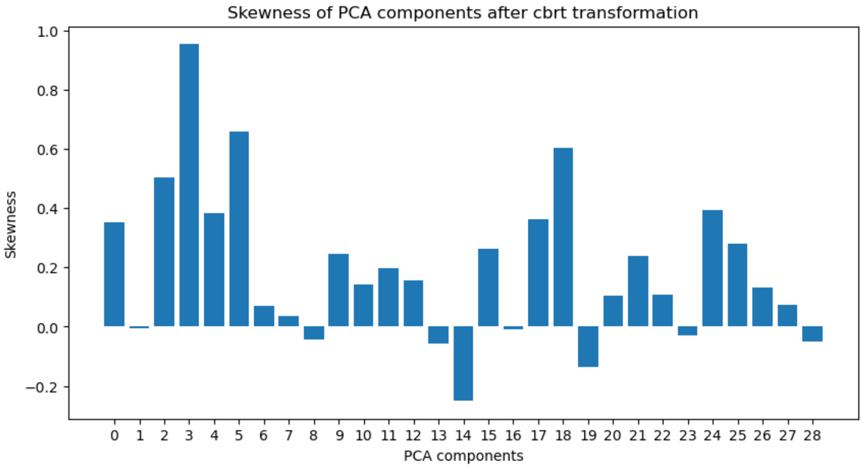

Figure 5: Cumulative explained variance of PCA. 29 components retain 90% variance.

PCA is essential because:

- It removes multicollinearity,
- Stabilizes training,
- Improves generalization,
- Reduces noise from redundant ratios.

7 Models and Hyperparameter Tuning (A1 Long Version)

We evaluate the following models:

- Logistic Regression,
- K-Nearest Neighbors,
- Support Vector Machine,
- Decision Tree,
- Random Forest (tuned),
- XGBoost (tuned),
- LightGBM,
- CatBoost,
- **Stacking Classifier (final model)**.

7.1 Logistic Regression

LR provides a linear decision boundary. **Pros**: fast, interpretable. **Cons**: insufficient for nonlinear financial patterns.

Hyperparameters:

- penalty = {L1, L2}
- C = [0.01, 0.1, 1, 10]

7.2 K-Nearest Neighbors

KNN is sensitive to scaling and dimensionality. Not competitive in high dimensions.

Hyperparameters:

- k = [3,5,7,11]
- distance metric = {Euclidean, Manhattan}

7.3 Support Vector Machine

SVM with RBF kernel handles nonlinear structure but scales poorly with n.

Hyperparameters:

- C = [0.1,1,10]
- gamma = ['scale','auto']

7.4 Decision Tree

Interpretable but prone to overfitting. We mainly use DT for feature importance.

Hyperparameters:

- max_depth = [3,5,7,9]

7.5 Random Forest (Tuned)

RF reduces variance by aggregating multiple trees.

Hyperparameters (Grid Search):

- n_estimators = [200,300,500]
- max_depth = [10,20,None]
- max_features = ['sqrt','log2']

7.6 XGBoost (Tuned)

XGBoost outperforms most classical ML models.

Grid search spans:

- learning_rate = [0.05,0.1]
- max_depth = [4,6]
- subsample = [0.7,0.8]
- n_estimators = [200,300]

7.7 Stacking Classifier (Final Model)

We stack three diverse base learners:

- Random Forest
- SVC (RBF)
- KNN

The meta-learner is Logistic Regression.

Stacking reduces generalization error by exploiting different learning biases.

8 Scientific Justification

8.1 PCA (Hotelling, 1933)

PCA provides an orthogonal transformation minimizing reconstruction error.

8.2 Gradient Boosting (Friedman, 2001)

Boosting iteratively fits weak learners on residuals.

8.3 Stacked Generalization (Wolpert, 1992)

A meta-learner captures patterns missed by individual models.

9 Obstacles Encountered

- Heavy imbalance → fixed with BorderlineSMOTE.
- Multicollinearity → addressed through PCA.
- Outliers → mitigated using quantile Winsorization.
- Lack of interpretability → solved via feature importance.
- Unstable training on raw data → improved with StandardScaler.

10 Results

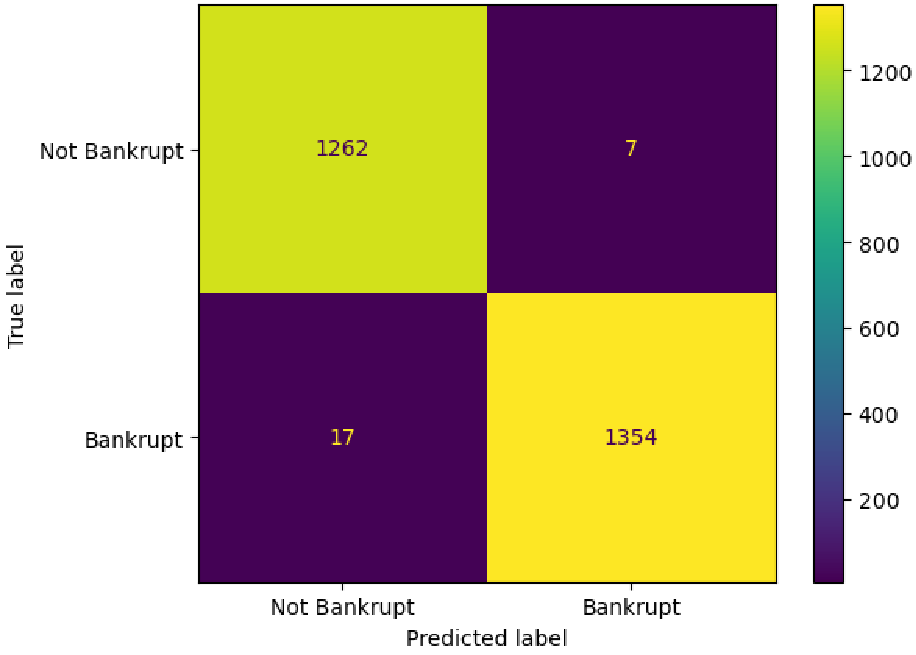

Figure 6: Confusion matrix of the Stacking Classifier. Extremely low false negatives.

Final performance:

$$F_1 \approx 0.99, \quad \text{Accuracy} \approx 0.992$$

11 Limitations

- PCA sacrifices interpretability.
- Data is Taiwan-specific.
- SMOTE introduces synthetic points.

12 Conclusion

The project successfully produces a high-performance bankruptcy prediction system. A Stacking Classifier combined with PCA and SMOTE offers near-perfect accuracy and robustness. The full pipeline aligns with the CDC guidelines and demonstrates solid methodological and analytical rigor.

13 References

- Chen, T. & Guestrin, C. (2016). XGBoost: A scalable tree boosting system.
- Hotelling, H. (1933). Analysis of a complex of statistical variables.
- Friedman, J. (2001). Greedy function approximation.
- Wolpert, D. (1992). Stacked Generalization.