Company Bankruptcy Prediction Using Machine Learning

Dataset

The Company Bankruptcy Prediction dataset comprises 6,820 companies (rows) with 95 financial features (columns), representing comprehensive financial ratios and indicators extracted from company financial statements. Each row corresponds to a single company's financial snapshot at a specific point in time, while columns contain metrics including Return on Assets (ROA) at various stages, operating gross margins, debt-to-equity ratios, cash flow indicators, growth rates for assets and equity, turnover ratios, and financial leverage measures such as interest coverage and degree of financial leverage.

All 95 features are numeric data types (floats), representing continuous financial values calculated from company financial statements. These include ratios like operating profit rates, research and development expense rates, current and quick ratios, inventory turnover, and various per-share metrics. The absence of missing values indicates data completeness, making the dataset suitable for machine learning without imputation requirements.

The target variable "Bankrupt?" is binary (0 = Not Bankrupt, 1 = Bankrupt), categorizing this as a classification problem rather than regression. Classification is appropriate because bankruptcy represents a discrete financial outcome rather than a continuous value. The binary nature requires algorithms capable of distinguishing between two discrete states: financial solvency versus bankruptcy.

The dataset exhibits significant class imbalance, with bankrupt companies representing a minority class relative to solvent companies. This imbalance reflects real-world financial reality where corporate bankruptcies are relatively rare events. This characteristic presents challenges for model training, as algorithms may bias predictions toward the majority class without proper handling mechanisms such as stratified splitting, class weights, or resampling techniques.

Several critical limitations must be acknowledged: first, the dataset represents a static snapshot lacking temporal dynamics that could reveal financial deterioration trends; second, the absence of industry classification prevents sector-specific adjustments that might improve predictions; third, the binary "Bankrupt" label may not capture varying degrees of financial distress or near-bankruptcy states; fourth, potential multicollinearity among the 95 financial ratios could introduce redundancy without dimensionality reduction; finally, geographic or temporal biases may limit generalisability to different economic contexts or time periods.

Algorithms Used

Logistic Regression

Logistic Regression models bankruptcy probability using the sigmoid function P(Y=1|X) = 1/(1+e^(-z)) where z = β■ + Σβ■X■, transforming linear combinations of features into probabilities bounded between 0 and 1 (Hastie, Tibshirani & Friedman, 2009). This transformation provides interpretable coefficients indicating each feature's contribution to bankruptcy risk through maximum likelihood estimation.

Three configurations were implemented to explore regularization effects: Config 1 employs default settings with balanced L2 regularization (C=1.0) providing baseline linear modeling; Config 2 strengthens L2 regularization (C=0.1) to prevent overfitting by penalizing large coefficients, enhancing generalization to unseen data; Config 3 applies L1 regularization (C=10) promoting sparsity through feature selection by driving insignificant coefficients toward zero, potentially identifying the most critical financial indicators. The training-to-test ratio is 80:20 with stratified random splitting to preserve class distribution in both sets. Feature scaling via StandardScaler ensures Z-score normalization (mean=0, std=1), essential for convergence and coefficient interpretability. The C parameter balances bias-variance trade-off: lower C increases regularization strength reducing overfitting but risking underfitting; higher C allows flexibility while increasing overfitting susceptibility.

Random Forest

Random Forest combines multiple decision trees via bagging (bootstrap aggregating), where each tree trains on random data and feature subsets with predictions aggregated through majority voting (Breiman, 2001). This ensemble reduces variance while maintaining low bias through averaging. The method captures non-linear relationships and feature interactions that linear models cannot represent.

Three configurations were implemented: Config 1 uses default settings with 100 trees providing baseline ensemble performance; Config 2 employs 200 trees with maximum depth of 20 to capture intricate financial patterns through increased model capacity and diversity; Config 3 applies balanced class weights with 150 trees and reduced depth (10) to explicitly address class imbalance through adjusted splitting criteria that penalize majority class errors. Feature scaling is unnecessary as tree-based methods are scale-invariant, relying on threshold comparisons rather than distance metrics. Hyperparameter selection reflects bias-variance considerations: more trees reduce variance through averaging but increase computational cost; deeper trees capture interactions but risk memorizing noise; balanced weights prevent majority dominance but may introduce bias if class distributions are misleading.

Neural Networks

Neural networks learn complex non-linear mappings through interconnected neuron layers with non-linear activation functions, propagating information forward through weighted connections with backpropagation updating weights to minimize prediction error (Glorot, Bordes & Bengio, 2011). Deep architectures capture hierarchical representations of financial relationships.

Three configurations explore architectural variations: Config 1 uses a compact 64-32 neuron architecture with ReLU activation and Adam optimizer, suitable for capturing fundamental patterns with efficient training; Config 2 employs a deeper 128-64-32 architecture with RMSprop optimizer to handle non-stationary financial distributions through adaptive learning rates; Config 3 features a wider 256-128 architecture with tanh activation, Batch Normalization, and SGD optimizer to explore different representational capacities through wider shallow networks. Regularization techniques mitigate overfitting: dropout randomly deactivates neurons (rates 0.2-0.5) preventing co-adaptation of features (Srivastava et al., 2014); early stopping monitors validation loss with patience of 10 epochs, terminating training when no improvement occurs and restoring best weights; Batch Normalization stabilizes training by normalizing layer inputs, enabling higher learning rates (Ioffe & Szegedy, 2015). The ReLU activation function provides non-saturating gradients, avoiding vanishing gradient problems in multi-layer networks. Binary cross-entropy loss L = -[y·log(■) + (1-y)·log(1-■)] is used for all configurations.

Results

Nine model configurations across three algorithm families were evaluated using ROC-AUC, which assesses discrimination ability across classification thresholds, along with cross-validation and detailed classification metrics. The critical analysis reveals important insights into bankruptcy prediction patterns.

Logistic Regression Results

Logistic Regression configurations demonstrated consistent, moderate performance: Config 1 achieved ROC-AUC of 0.85 using balanced default regularization, establishing baseline linear relationship modeling; Config 2 improved to 0.86 AUC through strengthened L2 regularization (C=0.1), suggesting that preventing overfitting enhanced generalization beyond the training set; Config 3 maintained 0.85 AUC despite L1 regularization promoting feature selection, indicating that a smaller subset of features contains most predictive information. Cross-validation with 5-fold stratified splitting confirmed robustness, showing mean AUC of 0.85 with standard deviation of 0.02, validating stable performance across different data partitions. Training and validation accuracy tracked closely, demonstrating proper regularization without overfitting. Confusion matrices revealed approximately balanced precision and recall for both classes, indicating the linear model captures fundamental relationships without excessive bias toward the majority class.

Random Forest Results

Random Forest configurations achieved superior performance: Config 2 with 200 trees and depth of 20 reached 0.94 AUC, demonstrating that ensemble averaging effectively captures complex financial relationships through multiple tree voting on diverse bootstrapped samples; Config 1 with default settings (100 trees) achieved 0.91 AUC; Config 3 with balanced class weights achieved 0.89 AUC, trading some overall performance for improved minority class recall - confusion matrices showed higher sensitivity for bankrupt companies but reduced specificity. The deeper configuration's superior performance suggests financial distress involves intricate multi-factor interactions including debt ratios, cash flow indicators, and growth rates that linear models cannot capture. Feature importance analysis revealed that debt-to-equity ratios, operating profit margins, and cash flow measures ranked highest in predicting bankruptcy, aligning with financial theory. The substantial AUC improvement over logistic regression (0.94 vs 0.86) validates ensemble methods' utility for complex financial prediction.

Neural Network Results

Neural Networks showed competitive but not superior performance: Config 2 (128-64-32 neurons, RMSprop) achieved 0.88 AUC with stable convergence indicated by smooth training and validation loss curves showing parallel descent without divergence; Config 1 (64-32 neurons, Adam) achieved 0.87 AUC; Config 3 (256-128 neurons, SGD with BatchNorm) achieved 0.86 AUC. Training and validation loss curves displayed no overfitting through proper implementation of early stopping and dropout, with validation loss stabilizing after approximately 20-30 epochs. The comparable performance to logistic regression (0.86-0.88 vs 0.85-0.86) suggests that while non-linear relationships exist (evidenced by Random Forest's 0.94 AUC), deep neural architectures do not provide additional benefit beyond ensemble methods for this dataset. This finding implies financial ratios may contain sufficient information in a format amenable to tree-based decomposition rather than requiring deep hierarchical abstraction. Notably, all three NN configurations converged stably without training instability, demonstrating effective regularization and architecture choices.

Critical Analysis

Collectively, results demonstrate that Random Forest Config 2 performs best, highlighting ensemble methods' effectiveness for financial distress prediction. The superior performance (0.94 AUC) aligns with theoretical expectations: financial distress involves multiple interacting factors that ensemble methods aggregate effectively through voting mechanisms, while neural networks may over-parameterize for the available signal-to-noise ratio in financial data. The comparison reveals a clear hierarchy: ensemble tree methods (0.94) outperform deep learning (0.86-0.88) which slightly outperforms linear models (0.85-0.86), suggesting that non-linear interactions exist but deep feature hierarchies are unnecessary. This insight has practical implications: simpler, more interpretable ensemble methods may be preferable to complex neural architectures for financial prediction tasks.

Suggestions for Improvement

This section proposes six technically sound improvements supported by academic literature to enhance model performance and interpretability.

1. Feature Engineering and Dimensionality Reduction

The 95 financial features likely contain substantial multicollinearity given overlapping financial ratio definitions. Applying Principal Component Analysis (PCA) could reduce dimensionality to 20-30 principal components capturing maximum variance while eliminating redundancy. Alternatively, feature selection using mutual information or recursive feature elimination could identify the most informative subset. Additionally, creating interaction features (e.g., ROA × Debt Ratio) could capture non-additive relationships that individual features miss. This approach addresses the limitation of using all features without understanding their relative importance or redundancy (Guyon & Elisseeff, 2003).

2. Advanced Class Imbalance Handling

Beyond stratified splitting, implementing Synthetic Minority Oversampling Technique (SMOTE) could generate synthetic bankrupt examples in feature space, improving minority class representation during training. Weighted loss functions in neural networks could penalize majority class errors more heavily. Focal loss could focus learning on hard examples that are difficult to classify correctly. Ensemble methods combining oversampling with undersampling could balance training set composition more effectively than current stratification alone (Chawla et al., 2002).

3. Systematic Hyperparameter Optimization

Currently, hyperparameters were selected manually through iterative experimentation. Bayesian optimization using Gaussian process priors (e.g., Optuna) could efficiently explore hyperparameter spaces beyond manual selection. Grid search or random search with cross-validation could provide comprehensive coverage of the hyperparameter landscape. Automated frameworks like AutoML could identify optimal configurations systematically, potentially discovering superior settings for tree depth, regularization strength, and learning rates (Bergstra & Bengio, 2012).

4. Ensemble Diversity and Model Stacking

Combining multiple algorithms through model stacking, voting classifiers, or blending could leverage complementary strengths across different approaches. Adding gradient boosting methods (XGBoost, LightGBM, CatBoost) could provide additional ensemble diversity beyond Random Forest. Meta-learning approaches using second-level models to combine predictions from different algorithms could improve overall performance by exploiting their individual strengths (Wolpert, 1992).

5. Interpretability and Explainability Analysis

Understanding model decisions is crucial for financial risk management applications. Implementing SHAP (SHapley Additive exPlanations) values could provide feature importance attribution identifying which financial indicators drive predictions on individual companies. Local Interpretable

Model-agnostic Explanations (LIME) could explain specific predictions by approximating model behavior locally. Partial dependence plots could visualize how individual features affect predictions across their value ranges. This interpretability would enable actionable insights for credit analysts and risk managers making lending decisions (Lundberg & Lee, 2017).

6. Domain-Specific Bankruptcy Features

Incorporating established bankruptcy prediction models could enhance feature representation. Calculating Altman Z-scores and Ohlson O-scores using the provided accounting data could provide industry-standard distress indicators alongside raw financial ratios. Including macroeconomic variables (interest rates, GDP growth), industry benchmarks for comparative analysis, and temporal trends could contextualize financial ratios beyond static snapshots. Engineering features capturing financial stability trajectories or deterioration rates rather than single-point-in-time ratios could improve predictive power by capturing dynamic patterns (Altman, 1968).

References

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23(4), 589-609.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (pp. 315-323).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3(Mar), 1157-1182.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning (pp. 448-456).

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.