Predicting Blue-Chip Financial Health Trajectory

Jessica Villanueva[1], Param Sejpal[2], Yu-Heng Chi[3], Yihan Yang[4], Zhiyi Zhang[5]
Applied Machine Learning (COMS W 4995), Columbia Engineering.
jss2326[1], pns2129[2], yc4548[3], yy3528[4], zz3274[5]

December 7, 2024

Abstract

This project examines machine learning approaches for predicting financial health among large publicly traded firms. Using data from the 2024 Fortune 1000 companies, we address two supervised learning tasks: regression to estimate Market Capitalization for the final fiscal quarter and classification to determine firm profitability status. Linear and ridge regression models, as well as a neural network, were implemented for the Market Capitalization task. A Decision Tree classifier with multiple sampling and weighting strategies was evaluated to address class imbalance in the profitability task. The results illustrate how financial indicators, preprocessing decisions, and sampling strategies influence predictive performance.

1 Introduction

This project examines two prediction tasks using data from the 2024 Fortune 1000 companies. The first is a regression task to estimate each firm's Market Capitalization (Market Cap) for the final fiscal quarter. Market Cap serves as a standard measure of firm size and investor valuation, making it an appropriate target for quantitative modeling. The second task is a binary classification problem to determine whether a company is profitable based on reported financial and operational metrics.

We evaluated several machine learning models under standard preprocessing steps, regularization methods, and sampling techniques designed to address feature correlations and class imbalance. The objective is to assess how these modeling choices influence predictive performance in large firm financial analysis.

1.1 Preprocessing

A data generation script was developed to collect variables and a Kaggle dataset of Fortune 1000 financials provided baseline attributes. Additional metrics (e.g., EBITDA) were extracted using the Yahoo! Finance API. Values unavailable through APIs were manually collected from SEC EDGAR 10-K filings. All steps were consolidated into `generate.py`.

Financial data is often skewed and blue-chip firms tend to exhibit strong profitability. This imbalance between profitable and non-profitable companies affects classification performance, so multiple techniques were incorporated to mitigate bias in the learning process.

To preserve class proportions across data splits, we used stratified sampling, ensuring that the distribution of profitable and non-profitable companies remained consistent in the development and test sets. Because the minority class ("non-profitable") was comparatively small, we also applied the Synthetic Minority Oversampling Technique (SMOTE) to generate synthetic minority examples. This procedure increases the representation of non-profitable firms and improves the classifier's ability to identify them.

In addition to imbalance considerations, the dataset exhibited typical financial domain issues such as missing values, correlated features, and outliers. Numeric variables with unavailable data were imputed using median values, which is appropriate for skewed distributions. Categorical attributes (Sector, Industry, CompanyType) were one hot encoded, and continuous variables were standardized to ensure consistent feature scaling.

1.2 Data Cleaning and Feature Preparation

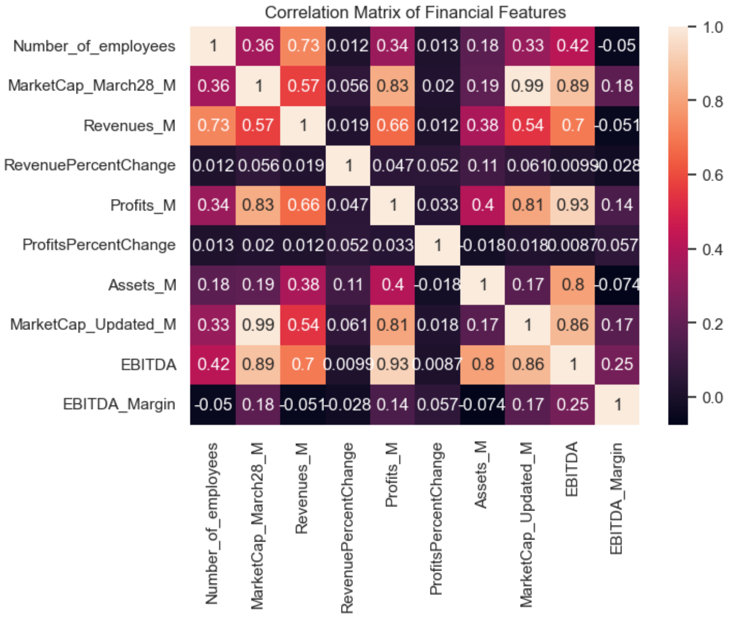
Financial datasets frequently contain missing values, correlated metrics, and outliers. To ensure consistency across models, we applied several procedures to clean, encode, and scale the variables.

Missing Data. Variables with minimal incomplete data, which typically comes from quarters in which firms had not yet reported earnings, were imputed using the median of each feature. Rows with extensive gaps in data represented a small fraction of the dataset and were removed.

Encoding and Standardization. Categorical variables (*Sector*, *Industry*, *CompanyType*) were transformed using one hot encoding. Continuous variables such as *Revenue*, *Profits*, and *Assets* were standardized to ensure consistent feature scaling across the regression and classification models.

Multicollinearity. Highly correlated features were excluded prior to regression to reduce variance inflation and stabilize coefficient estimates. Notable correlations included:

- *Number_of_Employees* with *Revenue*, reflecting the tendency of larger firms to employ more workers.
- *Profits_M* (gross profit) with *EBITDA*, both capturing related dimensions of firm profitability.



	Number_of_employees	MarketCap_March28_M	Revenues_M	RevenuePercentChange	Profits_M	ProfitsPercentChange	Assets_M	MarketCap_Updated_M	EBITDA	EBITDA_Margin
Number_of_employees	1	0.36	0.73	0.012	0.34	0.013	0.18	0.33	0.42	-0.05
MarketCap_March28_M	0.36	1	0.57	0.056	0.83	0.02	0.19	0.99	0.89	0.18
Revenues_M	0.73	0.57	1	0.019	0.66	0.012	0.38	0.54	0.7	-0.051
RevenuePercentChange	0.012	0.056	0.019	1	0.047	0.052	0.11	0.061	0.0099	-0.028
Profits_M	0.34	0.83	0.66	0.047	1	0.033	0.4	0.81	0.93	0.14
ProfitsPercentChange	0.013	0.02	0.012	0.052	0.033	1	-0.018	0.018	0.0087	0.057
Assets_M	0.18	0.19	0.38	0.11	0.4	-0.018	1	0.17	0.8	-0.074
MarketCap_Updated_M	0.33	0.99	0.54	0.061	0.81	0.018	0.17	1	0.86	0.17
EBITDA	0.42	0.89	0.7	0.0099	0.93	0.0087	0.8	0.86	1	0.25
EBITDA_Margin	-0.05	0.18	-0.051	-0.028	0.14	0.057	-0.074	0.17	0.25	1

Figure 1: Correlation Matrix

2 Regression

A linear regression model and a neural network were trained using `scikit-learn`. Given the multicollinearity present in several financial variables and the imbalanced structure of the dataset, L2 regularization was implemented for both models to improve stability and reduce overfitting.

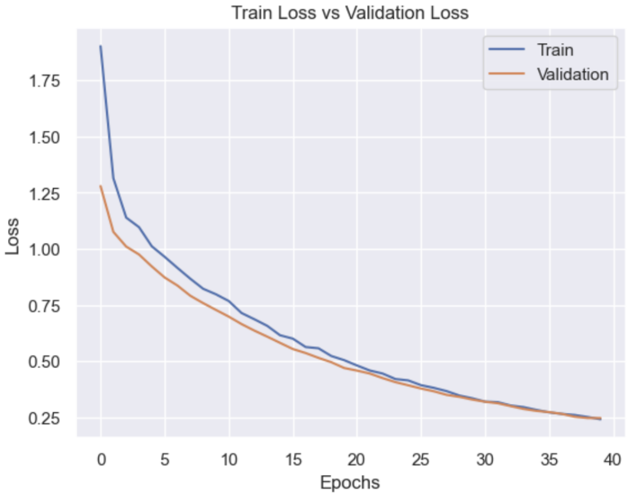


Figure 2: Train Loss vs. Validation Loss

2.1 Linear Regression

In our first deliverable, we noted that we would begin with a baseline model such as linear regression and incorporate additional models from that point forward. Performance from this baseline model was moderate: $R^2 = 0.724$ for training, $R^2 = 0.689$ for validation, and $R^2 = 0.739$ for testing. The RMSE values for the model were 43,457 (training), 40,949 (validation), and 47,800 (test). These metrics indicate that the baseline linear model captured general trends in the dataset but still exhibited notable variance across splits.

2.2 Ridge Regression

Given the nature of financial data and the level of feature correlation, we incorporated ridge regression as outlined in Deliverable 2. Hyperparameter tuning via grid search with cross-validation improved the model's generalization performance. The optimal regularization parameter was determined to be $\alpha = 10.0$. The final ridge regression model demonstrated solid results, with R^2 scores of 0.705 (training), 0.717 (validation), and 0.731 (test).

These values suggest an improved balance between bias and variance relative to the baseline model. The RMSE values were 44,983 (training), 39,159 (validation), and 48,489 (test), showing that the ridge penalty effectively reduced overfitting by discouraging large coefficient magnitudes and helped the model generalize more consistently across data splits.

2.3 Model Evaluation Metrics

Both regression and classification models were evaluated using standard supervised learning metrics.

Regression Metrics. Model performance for Market Cap prediction was assessed using:

- *Coefficient of Determination* (R^2): Measures the proportion of variance in the target explained by the model.
- *Root Mean Squared Error* (RMSE): Quantifies the average magnitude of prediction error in the same units as the target variable.

These metrics provide complementary views of model fit, with R^2 emphasizing variance explanation and RMSE reflecting absolute error.

Classification Metrics. Profitability classification was evaluated using confusion matrix components (true positives, true negatives, false positives, and false negatives), along with:

- *Precision and Recall*: Used to assess model behavior on the minority (non-profitable) class.
- *Area Under the ROC Curve* (AUC): Summarizes the tradeoff between true-positive and false positive rates across thresholds.

These metrics capture the model's effectiveness on imbalanced label distributions.

Table 1: Regression Model Performance

Model	**Train R^2**	**Val/Test R^2**	**RMSE**
Linear Regression	0.724	0.689 / 0.739	43,457 / 40,949 / 47,800
Ridge Regression ($\alpha = 10$)	0.705	0.717 / 0.731	44,983 / 39,159 / 48,489

2.4 Neural Networks

Similar to linear regression, we first developed a baseline neural network model and subsequently added regularization given the nature of the dataset. The training and validation loss curves for the baseline model decreased, indicating appropriate learning behavior. To improve performance and reduce overfitting, we trained a second neural network with L2 regularization and experimented with learning rates, batch sizes, and epoch counts.

We found that an L2 penalty of 0.001 combined with a dropout rate of 0.1 substantially improved model performance. This regularized network outperformed the baseline neural network by controlling weight growth and stabilizing the learning process.

2.5 Feature Importance Analysis

Feature importance results indicate that "Industry" (e.g., healthcare) and "Sector" (e.g., telecommunications) are significant predictors of Market Capitalization. This aligns with real world behavior. For example, companies in the healthcare sector often experience accelerated growth, and pending medical treatment approvals or product launches can sharply increase Market Cap. During the COVID-19 pandemic, Pfizer's market valuation rose considerably following the release of its vaccine, illustrating how sector-specific developments influence model outcomes.

3 Classification

The classification objective was to identify companies as either profitable or non-profitable. Initially, we trained a Decision Tree Classifier and applied several sampling techniques to mitigate the effects of class imbalance. These techniques included balanced class weights, random undersampling, random oversampling, and SMOTE.

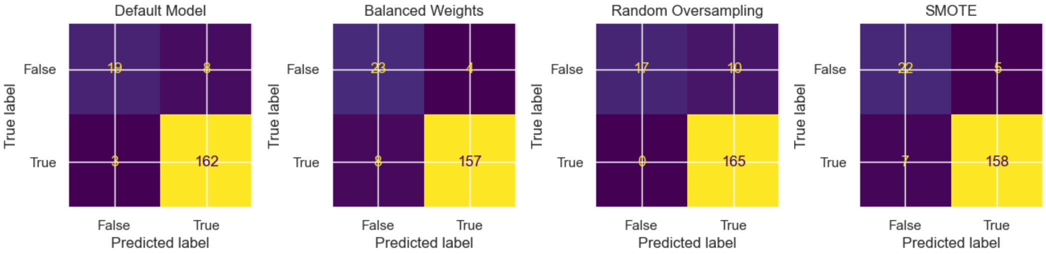


Figure 3: Confusion matrices for each classification model configuration.

3.1 Classification Results

The balanced weight model achieved an overall strong balance between true positives and true negatives and misclassified a small number of profitable companies.

Balanced Weights: The model prioritized precision and recall for the minority class (non-profitable). It performed well in identifying profitable companies, yielding 157 true positives. However, it was less accurate with non-profitable firms, identifying only 23 true negatives. The model incorrectly classified 4 non-profitable companies as profitable (false positives) and did not misclassify any profitable companies as non-profitable (0 false negatives).

Random Oversampling: Random oversampling improved recall for profitable companies but slightly reduced precision. This configuration produced the highest number of true positives (165) and misclassified none of the profitable firms. It also produced 17 true negatives but a higher number of false positives (10). While oversampling enhanced sensitivity, it introduced more false classifications among the non-profitable class.

Random Undersampling: Similar to oversampling, this model produced higher recall for profitable companies, correctly identifying 162 true positives. However, it introduced 3 false negatives and 8 false positives, with 19 true negatives. The reduction in data from undersampling may have contributed to decreased precision for the minority class.

SMOTE: The SMOTE-based model closely resembled the Balanced Weights configuration in behavior. It correctly identified 158 profitable companies (true positives) but misclassified 22 non-profitable companies as profitable. The similarity in performance suggests that synthetic oversampling provided only modest improvements under this model architecture.

The ROC analysis showed that the models achieved AUC scores ranging from 0.84 to 0.94. The default model achieved the lowest AUC at 0.84, while both the Random Oversampling and SMOTE models achieved the highest AUC scores of 0.94. The Balanced Weights model attained an AUC of 0.90. These results suggest that random oversampling and SMOTE may be the most suitable approaches for this task.

4 Acknowledgements

We would like to thank Dr. Pappu for providing the instruction and learning materials to help us create this project. We also thank the teaching assistants for their guidance and assistance with course content and project-related questions. The material and assignments in this course have contributed to our preparation for further advanced machine learning coursework at Columbia University and for future roles in industry.