Predicting Blue-Chip Financial Health Trajectory

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

Abstract

Predicting financial metrics using machine learning techniques is commonly performed in industry roles. In this project, machine learning models are used to analyze Blue-Chip company performance based on profitability and key performance indicators. Market Capitalization ("Market Cap") and status as a "profitable" or "non-profitable" company was investigated using data from the 2024 Fortune 1000 companies. Linear regression and neural networks were implemented to predict company Market Cap value for the last quarter of the fiscal year, and a collection of classifiers were trained to categorize companies by profitability.

1 Introduction

Examining a company's financial performance involves analyzing key performance indicators and market events. Market capitalization, or the equity value of a company, was chosen as the target variable in our regression analysis because it can give insights into what investors think about the company's financial health. Additionally, a company's status as "profitable" or "non-profitable" was the objective of our classification task. Therefore, our primary objective was to create data from 10-K reports and balance sheet statements to perform the following machine learning tasks.

(1) Regression: Predict 2024 Market Cap values using linear regression and a neural network. For both models, L2 regularization was implemented.
(2) Classification: Train a Decision Tree classifier and apply techniques like balanced model weights, random undersampling, random oversampling, and SMOTE to address data imbalance.

2 Data Source, Characteristics, and Preparation

A script for generating data was created to gather important financial information relevant to our regression and classification tasks. First, we found a Kaggle[1] dataset containing Fortune 1000 company financials reported in 2024. To collect the other KPIs, we used Yahoo! Finance's API to extract 2024 metrics, specifically focusing on EBITDA because it is (1) a measure of a company's profitability and (2) used to predict Market Cap. We then used the EDGAR[2] database on SEC.gov to manually impute values that were only accessible on their 10-K reports archive. These APIs were incorporated into a 'generate.py' script, which is available for users to run and view how our dataset was generated.

2.1 Data Preprocessing

After creating a script to generate data, we prepared the data for modeling using cleaning and processing methods. Financial data is oftentimes skewed, and our data specifically concerns blue-chip companies that tend to exhibit high profitability. The disparity between the majority and minority classes has implications for correctly classifying companies as profitable; therefore, we used multiple techniques to mitigate potential issues with the models correctly identifying companies, including:

ii. Stratified splitting: Given the imbalance in classes, we want to preserve the original dataset proportions when splitting the data into development and test sets.
iii. Synthetic Minority Oversampling Technique (SMOTE): Most Fortune 1000 companies are "profitable" (the majority class). Therefore, we used SMOTE to generate synthetic samples for the minority class ("non-profitable" companies) to address class imbalance and improve the model's ability to correctly classify non-profitable companies.

Financial data often has missing values, correlated metrics, and outliers. Therefore, we performed methods to clean, encode, and scale our data.

---

[1] Kaggle, "2024 Fortune 1000 Companies," accessed December 2024.
[2] U.S. Securities and Exchange Commission, "EDGAR Database," accessed December 2024.

i. **Missing Data:** For variables missing values—typically due to timeframes where companies have not reported earnings for a specific quarter—the median value of each column was used for imputation. We chose to use the median given the skewed nature of financial data and the need to preserve central tendency. Additionally, there were rows with missing values that represented a low proportion of the entire data; therefore, these were removed.

ii. **Standard Scaling and Encoding:** Categorical variables "Sector," "Industry," and "CompanyType" were transformed into binary columns using One Hot Encoding, and numerical features "Revenue," "Profits," and "Assets" were standardized.

iii. **Multicollinearity:** Highly correlated features were dropped prior to performing regression. The features that were correlated were:
   (1) "Number_of_Employees" and "Revenue": large businesses that generate a large amount of revenue typically have more workers.
   (2) "Profits_M" (gross profit) and "EBITDA": These metrics are *both* particular measures of a company's profitability.

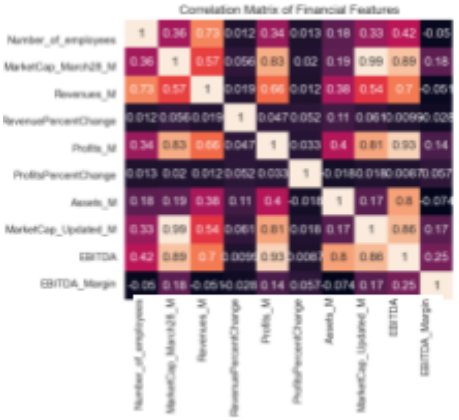

Figure 1: Correlation Matrix during Preprocessing

3. Regression

A linear regression model and a neural network was trained using *Scikit-learn*. Given the multicollinearity and imbalance in the data, we implemented L2 regularization for both the linear regression model and the neural network.

3.1 Linear Regression: In our first deliverable, we outlined that we wanted to potentially start with a baseline model like Linear Regression and incorporate other models from that point. Our performance from this model was moderate: $R^2 = 0.724$ for training, $R^2 = 0.689$ for validating, and $R^2 = 0.739$ for testing. The RMSE values for the model were: *43,457* (training), *40,949* (validation), and *47,800* (test).

3.2 Ridge Regression: Given how imbalanced financial data typically is, we wanted to incorporate ridge regression (outlined in Deliverable 2). Hyperparameter tuning definitely improved the model and helped us achieve slightly higher R^2 scores on validation and test sets. The **Ridge regression** model was optimized using grid search with cross-validation to find the best alpha value, which was determined to be $\alpha = 10.0$. The final Ridge regression model demonstrated solid performance, with R^2 scores of *0.705*, *0.717*, and *0.731* for the training, validation, and test sets, respectively. These values indicate that the model is balancing bias and variance and has an improved test performance compared to the linear regression model. The RMSE values for this model were: *44,983* (training), *39,159* (validation), and *48,489* (test), showing that the model generalizes well across the datasets and has successfully helped reduce overfitting by penalizing the large coefficients in our data.

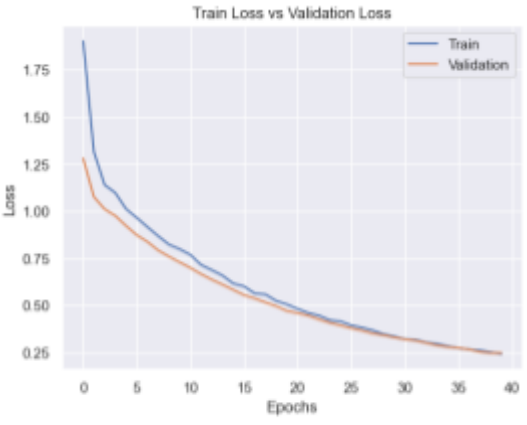

Figure 2: Train Loss vs. Validation Loss for the Neural Network with L2 Regularization

3.3 Neural Networks: Similar to linear regression, we developed a baseline model and then added a penalty due to the nature of our dataset. The training and validation loss for the baseline NN decreased, showing that it was properly learning. To improve performance, we then trained a model with L2 regularization and experimented with different numbers for learning rate, batch size, and epochs. We found that performance was improved with an L2 penalty of *0.001* and a dropout rate of *0.1*. This model had significantly better performance than the baseline NN because it prevented overfitting with the L2 penalty.

3.4 Feature importance analysis suggests that "Industry" (e.g., healthcare) and "Sector" (e.g., telecommunications) seem to be a significant indication of a company's Market Cap value. A real-world explanation for this result is that the healthcare industry tends to grow significantly faster than others, and a company's pending launches of medical treatments can spike its Market Cap. For instance, during the pandemic, Pfizer's Market Cap values shot up exponentially alongside the release of COVID-19 vaccines.

4 Classification

The classification task was to identify companies as *profitable* or *non-profitable*. We trained a Decision Tree Classifier and used sampling techniques to handle imbalance.

4.1 Classification Results

We can see from our confusion matrices that the balanced weight model achieved an overall ideal balance of true positives and negatives but misclassified a small amount of profitable companies.

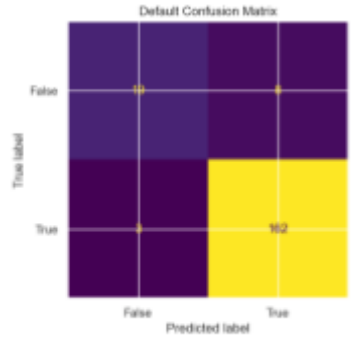

Figure 3: Confusion Matrix for Balanced Weight Model (the Model with the most Optimal Results)

i. **Balanced Weights:** This model prioritized precision and recall but prioritizes precision for the minority class (non-profitable). This model does well at identifying profitable companies (157 true positives) but is less accurate with non-profitable ones (23 true negatives). The model has few non-profitable companies that are incorrectly classified as profitable (4 false positives) and does not misclassify any profitable companies as non-profitable (0 false negatives).

ii. **Random Oversampling:** There was higher recall for profitable companies and slightly reduced precision. This model was the best at identifying profitable companies (165 true positives) and had no profitable companies misclassified. This model's performance is moderate; there are 17 non-profitable companies (true negatives) and a higher number of misclassifications (10 false positives).

iii. **Random Undersampling:** Similar to oversampling, this model has more recall for profitable companies (162 true positives and 3 false negatives). However, it may result in non-profitable companies being misclassified as profitable (8 false positives with 19 true negatives).

iv. **SMOTE:** Resulted in very similar performance to the Balanced Weights model; the model correctly identified 158 true positives and misclassified 22 non-profitable companies.

Lastly, the ROC curve showed that the models achieved AUC scores ranging from 0.84 to 0.94. The default model had the lowest AUC at 0.84, and the Random Oversampling and SMOTE models achieved the highest AUC scores of 0.94. The balanced weight model had an AUC of 0.90, which shows that the Random Oversampling and SMOTE models may be the most appropriate choices for our task.

5 Conclusion

Observations about company financial health and performance can be made based on the results of the models we used throughout this project. We used regression and classification to predict Market Cap, which represents company stock price and shares outstanding, and company profitability, which is an indicator of growth, company health, and potential growth. Our regression models showed that industry metrics, such as the ones in our dataset, can help us predict financial metrics like Market Cap. Additionally, our classification models provided insights into the financial characteristics of companies, particularly in distinguishing between "profitable" and "non-profitable" (which also gives us insight into Market Cap values). Through this project, we saw how to handle certain data (e.g., imbalanced classes or correlated features), and how model generalizability varies across different techniques.

Acknowledgements

We would like to thank Dr. Pappu for the opportunity to conduct this project and for providing us with the lecture and assignment materials to prepare us for this project. The content and assignments in this course prepared us not only for future machine learning courses at Columbia University, but also for numerous roles in the industry. We would also like to thank the teaching assistants in this course for giving us their time in advising and helping us with course material and project inquiries.