Predicting Stock Returns from Company Financials and Machine Learning

Aerjay Castañeda and Ligaya Leah Figueroa

Department of Computer Science
College of Engineering
University of the Philippines Diliman, Quezon City, Philippines
abcastaneda1@up.edu.ph, llfigueroa@up.edu.ph

Abstract. The accurate prediction of the performance of stocks in the stock market has been a longstanding problem in the field of finance and applied mathematics. We use financial statements data from the U.S. SEC and share price data from Kaggle to predict U.S. stock market returns using LightGBM. After training, we construct a daily portfolio from the predictions, which we backtested over the years 2015-2021, yielding annualized returns of 5.57% for the standard strategy, and 9.43% for the modified strategy, and Sharpe ratios of 0.855 and 0.956 respectively. Finally, we analyzed the relative importance of the features used, showing that momentum features are the most significant predictors, followed by `days_since_ddate` and Net Income-based features.

Keywords: stock market prediction, machine learning, financial ratios

1 INTRODUCTION

The accurate prediction of stock market performance has been a subject of intense scrutiny and research for many years. Financial analysts, economists, investors, and traders all stand to benefit immensely from accurate and timely forecasts of stock prices. In recent years, the advent of machine learning techniques has revolutionized the field, leading to the creation of sophisticated predictive models that leverage vast amounts of financial and economic data.

The goal of our research is twofold: First, to employ a machine learning model to predict the future performance of U.S. stocks using financial statement data; and second, to assess the relative importance of each feature to identify the most predictive variables. We focus on data items from the balance sheet, income statement, and cash flow statements, sourced from the U.S. SEC's financial statement dataset [13], and pricing data from a Kaggle stock market dataset [10].

We decided to employ gradient-boosted decision trees as our model, as such models have generally outperformed most other machine learning models, including neural networks, on tabular data [12]. As [12] also notes, they also require less tuning than other approaches. We used LightGBM, a gradient boosting library for Python, as our model for predicting stock returns primarily due to its efficiency [8].

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2023)*, Atlantis Highlights in Computer Sciences 20,
https://doi.org/10.2991/978-94-6463-388-7_22

2 BACKGROUND AND RELATED WORK

The academic field concerned with the analysis of stock market returns is an old but still active field of inquiry. In this section, we lay out some of the existing research on applying machine learning techniques on financial data for stock return prediction.

2.1 Financial Statement Data

There is a large literature on finding factors, which are usually ratios of financial statement values, that have predictive power. Jensen et al. analyses 153 factors from the literature, majority of which are based on items from financial statements, to assess whether most findings can be replicated both in-sample and out-of-sample [7]. They find that most such factors can be replicated, and can be clustered into various themes. Many of these clusters, including but not limited to *accruals*, *debt issuance*, *investment*, *value*, *profitability*, and *profit growth*, are based almost exclusively on financial statement items.

2.2 Machine Learning-based Asset Pricing Models

Traditional asset pricing models are used for determining the factors that influence the cross-section of stock returns. With the advances in machine learning and the ever-increasing amount of and access to data, the academic literature on such models have moved from linear models with a handful of factors to more complex machine learning models with dozens or hundreds of variables. In [5], the authors applied various models, spanning from linear models, to random forests, boosted regression trees, and neural networks. They find that tree-based and neural-network-based methods performed best in their study, which used 94 factors to predict stock returns.

2.3 Feature Importance in Stock Prediction

Despite the neural network resurgence, traditional machine learning retains its appeal due to simplicity, interpretability, and efficacy in low-data settings. Liu et al. utilized gradient boosting with financial ratios as features to predict financial distress in Chinese firms [9]. With TreeSHAP, they identified key predictive ratios, including *net asset value per share*, *net profit after deducting nonrecurring gains and losses*, *ratio of inventory to current liabilities*, *undistributed profit per share*, *ratio of operating profits to current liabilities*, *earnings per share (diluted operating profits)*, and *ROE (cut weighted)*. In [5], in addition to applying various machine learning models to stock prediction, Gu et al. also compared the variable importances of each model. The authors find that momentum-based features perform best across most models.

3 METHODOLOGY/DESIGN

This paper aims to construct a set of models for predicting stock returns using features derived from financial statement items and daily share prices. The following subsections detail the methodology employed, outlining the data gathering process, the feature extraction and engineering stages, the construction of the model, and the model validation process.

3.1 Data Acquisition and Management

The data for this study come from two sources. The quarterly financial statements data were downloaded from the U.S. Securities and Exchange Commission (SEC) website [13], while the daily share prices data are downloaded from Kaggle [10].

The SEC updates the financial statements dataset on a quarterly basis, starting from 2009Q1, and with the latest release being 2022Q4 (as of March 2023). Each release is in the form of a ZIP file, containing four text files (in tab-separated values format) and one `readme.htm`, which contains documentation for the dataset. The description and datatypes of the fields, and the mappings of the keys between the four files are all documented in this `readme.htm` file, which we relied on when constructing the data to be merged with the Kaggle dataset.

The Kaggle prices dataset contains the daily open, high, low, and closing prices for all stocks listed in the various main exchanges in the U.S. (NASDAQ, NYSE), and also some indices (S&P 500, Forbes 2000) from the 1980's to late 2022. It also includes corporate events data, including splits and dividends, and most importantly, the split-adjusted closing prices for all stocks. This is important because not adjusting for stock splits would yield erroneous target data, since we will derive the target values (in the form of log returns) from the adjusted closing prices.

The SEC financial statements dataset and the Kaggle prices dataset do not share a key, we use the CIK-Ticker mappings provided in [3] in order to merge both datasets.

3.2 Feature Construction

Most features used to train the model are derived from the SEC Financial Statements dataset, with the exceptions of four features deriving from the Kaggle Stock Market dataset. These four features are all *momentum* features, based on one of the factors from Carhart's pricing model [1]. These four are calculated as follows:

- `mom_21`: rolling 21-day (i.e. 1 month) log return
- `mom_63`: rolling 63-day (i.e. 3 month) log return
- `mom_126`: rolling 126-day (i.e. 6 month) log return
- `mom_252`: rolling 252-day (i.e. 12 month) log return

where n-day log return is defined as follows, in terms of the adjusted closing price P:

$$n\text{-day logarithmic return}_t = \ln \frac{P_t}{P_{t-n}}$$

The rest of the features are constructed from a number of well-known financial ratios, based on [2], and transformations thereof.

We also added a feature defined as the number of days since the period end date. Many of the features defined so far are updated only quarterly. We convert these to daily features by propagating the latest observation forward in time until the next release.

Finally, we defined the target variable as the next 5 trading days' logarithmic return, Winsorized at the 5^{th} and 95^{th} percentiles. We applied Winsorization to reduce the effect of outlier returns on our models, as stock returns are known to have heavy tails [4].

3.3 Model Construction

To construct our models, we used LightGBM, gradient boosting library [8]. The structure of the model training setup follows the conventional time series cross-validation methodology [6]. We refit the model at the end of every calendar quarter. Each model is given 1,200 trading days of lookback for its training window. Note that this is clipped for earlier years, as the earliest data was available around 2007.

Since the target variable used is the future 5-day cumulative logarithmic return, we always exclude the last 5 trading days of the training period to avoid lookahead bias.

3.4 Portfolio Construction

We follow the following steps in order to construct our daily portfolio:

1. Train the model for the the current period (quarter).
2. Run the model given the data for this period, and save its predictions.
3. Construct a long-short portfolio of stocks from the predictions.
4. Calculate the Net Profit / Loss of the portfolio using logarithmic returns.

To construct a portfolio given predictions of the returns of each stock for a day, we use the following equation:

$$w_i = \begin{cases} \frac{\hat{y}_i}{2\sum_{\hat{y}>0}\hat{y}} & \hat{y}_i \geq 0 \\ -\frac{\hat{y}_i}{2\sum_{\hat{y}<0}\hat{y}} & \hat{y}_i < 0 \end{cases}$$

where $\hat{y}_i = y_i - y_{\text{median}}$ is the median prediction for stock i minus the median prediction on the same day, and w_i is the resulting portfolio weight for stock i.

The subtraction by the median prediction centers the values on zero, so that it can be scaled as above to satisfy the following constraints:

$$\sum |w_i| = 1$$
$$\sum w_i = 0$$

Note that a positive w_i means that the model is taking a long position on the stock, while a negative value means that it is taking a short position.

A portfolio derived using the equations above would take long positions on stocks that the model predicts would have a higher return, and short positions on those that it predicts would have a lower return. Moreover, the higher the prediction is, the bigger the position will be. In this paper, we call this portfolio construction method a *weighted long-short* strategy.

Finally, we add a slight variation on this strategy, which is to only consider stocks that are in the top and bottom deciles of the distribution of the predictions. In other words, this strategy will only take positions on stocks where the models are most confident—the top 10% and the bottom 10% in terms of the predictions for the day.

4 EVALUATION

The models were trained over a span of 10 years of quarterly financial statements data and daily pricing data, with a quarterly refit frequency. The first model was set to predict on the first trading day of 2015, and the last model was set to predict on the last trading day of 2021.

First, we evaluate the predictions using the traditional metrics used in the field of machine learning. One of the standard metrics used for regression models is the Root Mean Squared Error (RMSE), defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_i^N (\hat{y}_i - y_i)^2}{N}}$$

Over the entire training period, the models achieved an RMSE of 0.033. We can also evaluate the predictions by converting the problem from regression to classification. We can do so by setting the target variable and the prediction to 0 or 1 depending on their sign, which is equivalent to the problem of predicting whether a stock's price will go up or down. Thus, we can now leverage the traditional classification metrics to evaluate our models, such as accuracy, precision, recall, and F1 score, defined as follows [11]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP, FP, TN, FN are the number of true positives, false positives, true negatives, and false negatives, respectively. Table 1 shows the values achieved for each of these metrics.

While these metrics are heavily used in traditional machine learning problems, it has some issues when used in the field of stock trading. The issues arise from the difference between the goal of stock *prediction* and stock *trading*. The former has the intrinsic goal of predicting the prices (or equivalently, returns) of *all* stocks accurately, while the latter has the goal of maximizing profit. To illustrate, if one's calibrated predictions is composed of a few high- and many low-confidence predictions, these will be severely penalized under the conventional ML metrics, but will not be as severely penalized under more trading-native metrics. This is because in trading, the lower confidence predictions can be discounted. This is done explicitly in the construction of the long-short portfolios, which weight the predictions by their distance to the median.

Method	**Metric**	**Value**
Regression	RMSE	0.033
Classification	Accuracy	0.503
	Precision	0.512
	Recall	0.545
	F1 Score	0.528
Long/Short	Sharpe Ratio	0.855
Decile Long/Short		0.956

Table 1. Metrics for evaluating the resulting models.

Over the entire training period of 7 years, the *weighted long-short* portfolio had a compounded return of 39% (5.57% annualized), while the *top/bottom decile weighted long-short* portfolio had 66% (9.43% annualized). In comparison, a strategy of taking a long position on all the stocks for which we have data (equally weighted) would yield a return of 48.7% (6.96% annualized). This has a higher return than the first strategy, but has a much lower return than the second.

Figure 1 graphs the cumulative net profit (loss) across time for the 3 given strategies. Note that both proposed strategies from our models are much more

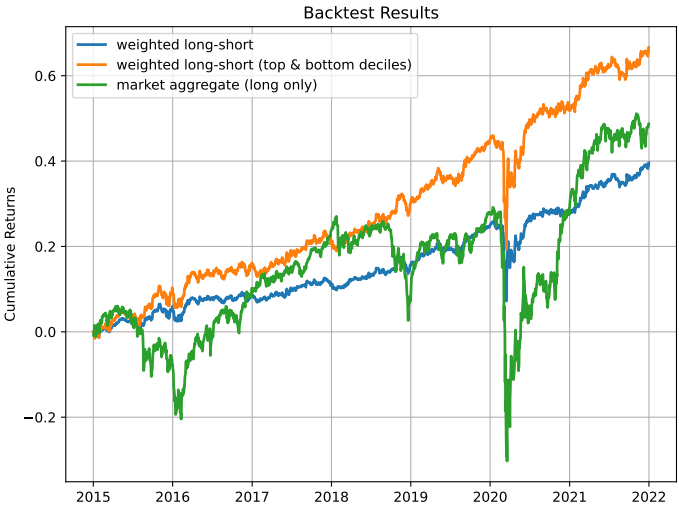

Fig. 1. The results of the backtest for the two proposed strategies and the market aggregate.

stable compared to the market aggregate. Ideally, we want a performance measure that takes into account both the risks and reward that a strategy takes. There is such a performance measure, and in fact is a standard in the field of trading, called the Sharpe ratio. It is defined as follows:

$$\text{Sharpe Ratio} = \frac{\mathbb{E}[R_a - R_b]}{\sigma_a}$$

where R_a is the asset returns (which in our case is the portfolio return series), R_b is the risk free rate of return, and σ_a is the standard deviation of the asset's excess return (over the benchmark). By convention, risk free rate is generally set to the return of U.S. Treasury bonds. As we do not have access to these data, we set this to 0. This ratio measures the expected return we get per unit of risk that we take. Over the training period, the *weighted long-short* portfolio has achieved a Sharpe ratio of 0.855, while the *top/bottom decile weighted long-short* portfolio achieved 0.956. In contrast, the market aggregate had a Sharpe ratio of 0.368. This difference can be attributed to the relative stability of the returns from the *long-short* strategies proposed, compared to the market's volatility.

Another goal of this paper is to assess which financial data items had the greatest predictive power in our dataset. We use LightGBM's feature importance metrics in order to extract the mean *gain* and *split* values per feature, as shown in Figure 2. Two sets of features seem to dominate: the momentum-based features, and `days_since_ddate` (i.e. the number of days since the end of the quarter corresponding to the current release). The fact that momentum is a strong signal is expected: this is a well-known factor that influences the cross-section of returns, as discussed in [1]. This finding also supports Gu et

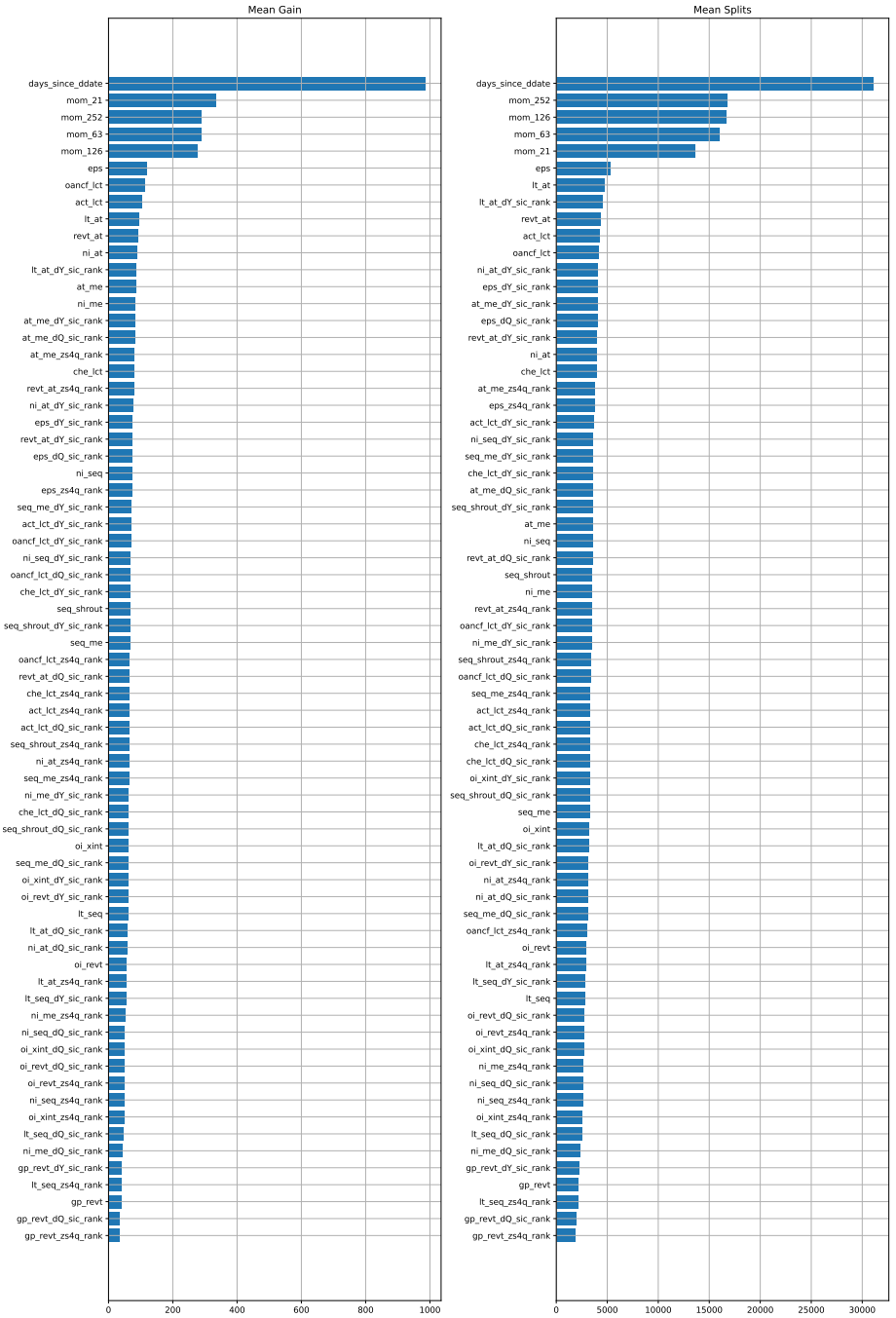

Fig. 2. Feature importance values averaged over all models from 2015 to 2021.

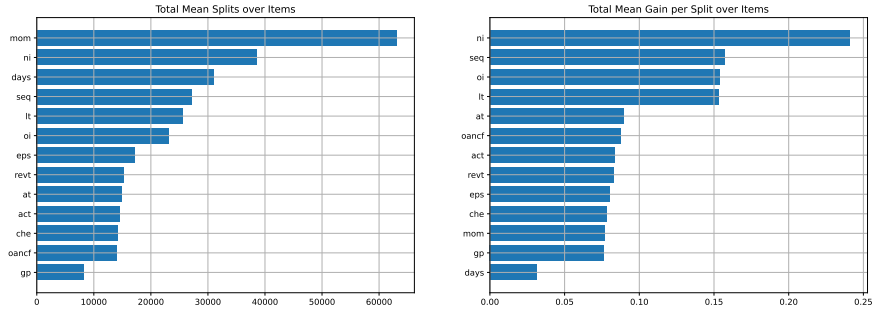

Fig. 3. Mean gain (left) and mean splits (right) over 2015-2021, summed over the financial items.

al.'s findings, whose models found that momentum-based variables perform best across a wide variety of machine learning models [5]. As for `days_since_ddate`, we suspect that this is most likely due to the fact that most of the financial features we have have a very low frequency, and so it is a useful indicator for the model as to the "staleness" of the current datapoint.

One potential issue with Figure 2 is that the contribution of each of the (non-momentum) financial items are spread out over multiple features, and so it is not a fair comparison. To deal with this, we sum each of the mean gain and split values over the raw items, and create an aggregated plot on Figure 3. As an example, the importance value for net income (`ni`) would then be the sum of the importance values of `ni_at`, `ni_me`, and so on. While the momentum features still dominate, we also see that Net Income is the most important financial item to our models.

5 CONCLUSION

This study sought to predict stock returns using historical quarterly financial statements data and daily stock pricing data, and to analyze which features were most significant to the model. Using the LightGBM model, we were able to construct a simple but effective model, and to determine which financial items had the most predictive power.

Our results demonstrate the potential of combining machine learning and financial analysis to improve stock return predictions. The constructed models, based on derived features from both financial statements and pricing data, showed promising performance, with our best strategy (top / bottom decile weighted long-short) outperforming the market aggregate both in terms of higher returns and lower volatility. This can be attributed to the feature engineering process and the flexibility of LightGBM, which was able to handle the relatively sparse time-series dataset. Also, the choice of the portfolio construction method had a significant impact on the overall returns that the strategy was able to

achieve. It is important to note, however, that stock market prediction is inherently complex due to numerous unpredictable factors such as market sentiment and global economic events.

Future research in this area could explore the integration of additional data sources, such as social media sentiment analysis or macroeconomic indicators, to further enhance the predictive power of the model. We suspect this would be particularly useful as almost all of the features used in this paper have a very low frequency, so adding higher frequency data may give a significant boost in the model's performance. Another worthwhile endeavor would be to explore the application of newer or alternative machine learning techniques, and better interpretability tools and techniques. And finally, the backtesting methodology was simplified due to resource constraints. Ideally, this should factor in trading costs, market impact, risk-free rate, and other more realistic assumptions.

In conclusion, this study contributes to the growing body of literature on the intersection of machine learning and finance. It demonstrates the feasibility and potential of using machine learning models for stock returns prediction, offering valuable insights for both academics and practitioners in the field.

References

1. Mark M Carhart. On persistence in mutual fund performance. *The Journal of finance*, 52(1):57–82, 1997.
2. CFI Team. Financial Ratios, 2023. Accessed: May 6, 2023.
3. Jad Chaar. Sec cik-ticker mapping. https://github.com/jadchaar/sec-cik-mapper, 2022. GitHub repository.
4. Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance*, 1(2):223, 2001.
5. Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.
6. Hyndman, R.J. and Athanasopoulos, G. *Forecasting: principles and practice*. OTexts, Melbourne, Australia, 3 edition, 2021. Accessed: May 20, 2023.
7. Theis Ingerslev Jensen, Bryan Kelly, and Lasse Heje Pedersen. Is there a replication crisis in finance? *The Journal of Finance*, 2022.
8. Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
9. Jiaming Liu, Chengzhang Li, Peng Ouyang, Jiajia Liu, and Chong Wu. Interpreting the prediction results of the tree-based gradient boosting models for financial distress prediction with an explainable machine learning approach. *Journal of Forecasting*, 2022.
10. Paul Mooney. Stock Market Data (NASDAQ, NYSE, S&P500), 2023. Accessed: April 8, 2023.
11. David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, 2(1):37–63, 2011.
12. Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.
13. U.S. Securities and Exchange Commission. Financial Statement Data Sets, 2023. Accessed: April 8, 2023.