Forecasting Stock Market Dynamics in the Tech Sector: A Multi-Stock Gradient Boosting Approach*

Jamie Lee

November 21, 2024

First sentence. Second sentence. Third sentence. Fourth sentence.

1 Introduction

Overview paragraph

Estimand paragraph

Results paragraph

Why it matters paragraph

Telegraphing paragraph: The remainder of this paper is structured as follows. Section 2….

2 Data

2.1 Overview

The data for this paper, obtained from the Yahoo Finance API via the `tidyquant` library in R on November 19, 2024, encompasses historical stock price information for four leading technology companies: Google (GOOG), Apple (AAPL), Amazon (AMZN), and Microsoft (MSFT). This dataset spans from January 1, 2018, to December 31, 2024, providing daily stock market data for each company. Key variables include **`symbol`**, **`date`**, **`open`**, **`high`**, **`low`**, **`close`**, **`volume`**, and **`adjusted`** prices. Our analysis focuses on predicting daily price differences

*Code and data are available at: https://github.com/jamiejiminlee/Tech-Stock-Forecast.git

and percentage changes across this basket of stocks, leveraging additional features constructed from the raw dataset. These features include `Price_Lag1` (lagged closing prices), `Price_Diff` (daily price differences), and `Price_Change_Percent` (percentage changes). By incorporating these derived variables, we aim to capture sequential trends and normalize differences across stocks with varying price ranges.

To ensure reliability, the dataset was preprocessed to remove missing values, align timeframes across stocks, and compute the constructed variables. Observations with incomplete data were excluded to maintain consistency in analysis. Details on the data cleaning and feature construction process are provided in **?@sec-data-cleaning**.

Data cleaning and analysis were conducted using the following packages: the `tidyverse` package of (**Tidyverse?**), `lubridate` package of (**Lubridate?**), and `arrow` package of (**arrow?**). These tools facilitated efficient data manipulation, feature construction, and storage in a compact Parquet format for further analysis.

2.2 Measurement

Stock prices are influenced by numerous factors, including macroeconomic conditions, industry trends, corporate actions, and investor sentiment. These dynamics fluctuate frequently, often in response to earnings reports, geopolitical events, or shifts in market expectations. Financial data simplifies these complex market behaviors into numerical indicators such as `open`, `high`, `low`, and `close` prices, which, while precise, may not fully encapsulate underlying investor motivations or market sentiment. For instance, the closing price represents only the last traded price of the day, smoothing out intraday volatility and obscuring more nuanced trading patterns.

Constructed variables like `Price_Diff` and `Price_Change_Percent` aim to quantify daily stock price movements by measuring differences or proportional changes relative to the previous day. However, these metrics assume that price movements reflect consistent investor sentiment across all stocks, potentially overlooking idiosyncratic factors specific to individual companies, such as leadership changes or product launches. Additionally, these metrics equally weight all days, despite certain trading days (e.g., earnings announcements or market holidays) being more influential.

The `volume` variable, representing the total shares traded, is used as a proxy for market activity and investor interest. This assumes that higher trading volumes uniformly indicate significant market events, despite possible differences in trading motives, such as speculative activity or algorithmic trading. Similarly, the temporal predictors, including `date`, `day of the week`, and `month`, aim to capture seasonal trends or trading patterns. However, these features simplify complex market behaviors into discrete time intervals, potentially overlooking shorter-term dynamics like market reactions to breaking news or macroeconomic data releases.

By aggregating these variables across a basket of stocks, the model assumes comparability among them, despite differences in market capitalization, industry focus, and investor bases. This simplification may obscure unique drivers of price changes for each stock, potentially limiting the model's ability to capture more granular patterns. Nonetheless, these variables collectively provide a structured framework for analyzing and predicting daily price movements within the selected basket of stocks.

2.3 Variables

The collected data from Yahoo Finance includes several key variables relevant to the analysis of price changes across a basket of tech stocks (**Google**, **Apple**, **Amazon**, and **Microsoft**). The original dataset contains the following columns:

- `symbol`: Represents the stock ticker symbol, identifying the company (e.g., GOOG for Google, AAPL for Apple).
- `date`: The trading date, essential for tracking and analyzing temporal patterns.
- `open`: The stock's opening price on a given day, providing context for daily price movements.
- `high`: The highest price of the stock during the trading day, useful for understanding intraday volatility.
- `low`: The lowest price of the stock during the trading day, another measure of volatility.
- `close`: The stock's closing price on a given day, which is a standard benchmark for daily performance.
- `volume`: The total number of shares traded during the day, reflecting market activity and investor interest.
- `adjusted`: The adjusted closing price, accounting for corporate actions like stock splits and dividends to provide a standardized measure of value.

In addition to these original variables, new features were constructed to enhance the analysis:

- `Price_Lag1`: Derived as the closing price from the previous trading day, enabling the analysis of sequential price changes.
- `Price_Diff`: Calculated as the difference between the current day's closing price and the previous day's closing price, capturing day-to-day fluctuations.
- `Price_Change_Percent`: Derived as the percentage change in stock price relative to the previous day's closing price, normalized to enable comparisons across stocks.

These constructed variables allow for a more detailed examination of stock price movements. To ensure high-quality analysis, the data was cleaned and preprocessed to remove missing values, align timeframes across stocks, and compute these derived features. For further details on the data cleaning and preparation process, refer to **?@sec-data-cleaning**.

2.3.1 Outcome Variables

The outcome variable for this paper is the **Daily Percentage Change (`Price_Change_Percent`)**, which measures the proportional change in a stock's closing price relative to its previous day's closing price. This variable is calculated as:

$$\text{Price\_Change\_Percent} = \frac{\text{close} - \text{Price\_Lag1}}{\text{Price\_Lag1}} \times 100$$

This metric normalizes price movements across the basket of stocks, enabling meaningful comparisons despite differences in price ranges between stocks such as Google (**GOOG**), Apple (**AAPL**), Amazon (**AMZN**), and Microsoft (**MSFT**). By focusing on percentage changes, the analysis captures the relative daily performance of each stock, providing insights into trends and volatility. This variable is widely used in financial analysis to track and compare stock performance over time, making it a suitable choice for this study.

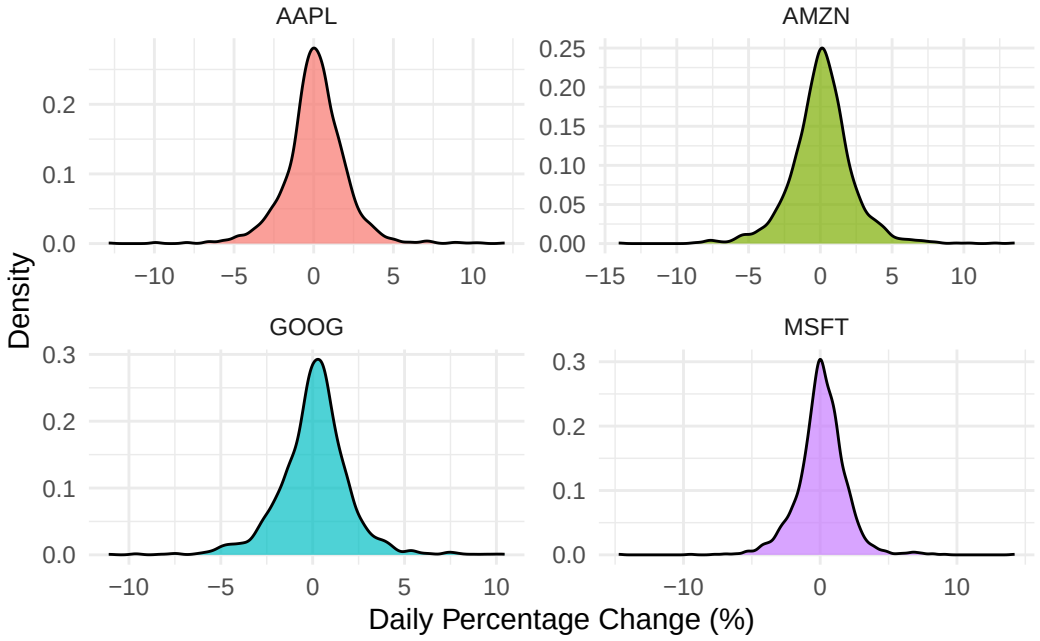

Figure 1: Distribution of Daily Percentage Changes by Stock

Figure 1 above illustrates the distribution of **Daily Percentage Changes (`Price_Change_Percent`)** for each stock in the basket: Apple (**AAPL**), Amazon (**AMZN**), Google (**GOOG**), and Microsoft (**MSFT**). Each panel represents a stock, providing a separate density curve that highlights its unique distribution pattern. Across all stocks, the distributions exhibit a sharp central peak around 0%, indicating that most daily percentage changes are minor. However, the tails of the distributions reveal occasional larger price movements, both positive and

negative, reflecting the volatility inherent in stock markets. Notably, the widths of the distributions differ slightly among the stocks, with Amazon (AMZN) displaying a broader tail, suggesting higher variability in its daily percentage changes compared to others.

2.3.2 Predictor variables

To forecast the daily percentage changes in stock prices (`Price_Change_Percent`), several predictor variables are utilized, derived from the cleaned dataset. These predictors encompass historical price trends, market activity metrics, and temporal features, offering a comprehensive framework for modeling stock price behavior:

- `Price_Lag1`: The closing price of the stock on the previous trading day. This variable captures the most recent historical price trend, which is often a strong predictor of subsequent price movements.
- `volume`: The total number of shares traded on a given day. This variable reflects market activity and investor interest, with higher volumes often indicating significant events or changes in sentiment.
- `open`: The stock's opening price on a given day. This metric provides context for intra-day price dynamics and can indicate whether the day begins with positive or negative momentum.
- `high`: The highest price the stock reached during the trading day. This variable highlights the potential upward volatility within the day, offering insights into optimistic trading behavior.
- `low`: The lowest price the stock reached during the trading day. This variable captures the potential downward volatility, reflecting pessimistic market behavior.
- `adjusted`: The adjusted closing price, which accounts for corporate actions like stock splits and dividends. This ensures a standardized measure of value over time, enabling better comparisons across stocks.
- `date`: The trading date, used to derive temporal features such as the day of the week and the month. Temporal variables capture seasonal trends and trading patterns that can influence stock prices.

These predictor variables enables the model to incorporate historical price trends, market dynamics, and seasonal influences, providing a robust basis for forecasting daily percentage changes in stock prices. By leveraging this diverse set of predictors, the analysis aims to identify patterns and relationships that drive stock price fluctuations across the basket of tech stocks.

Table 1: Sample of Predictor Variables

Stock Symbol	Date	Previous Close (Price_Lag1)	Volume	Opening Price	Highest Price	Lowest Price	Adjusted Close
AAPL	2024-01-11	186.19	49128400	186.54	187.05	183.62	184.69
MSFT	2020-07-15	208.35	32179400	209.56	211.33	205.03	200.49
MSFT	2023-09-20	328.65	21436500	329.51	329.59	320.51	318.38
AMZN	2019-06-20	95.44	64344000	96.67	96.76	95.29	95.91
GOOG	2020-12-29	88.80	25988000	89.39	89.62	87.80	87.72

Table 1 displays a random sample of five rows from the dataset, showcasing the predictor variables used in the analysis. These variables include **stock symbol**, **date**, **previous day's closing price (Price_Lag1)**, **volume**, **opening price**, **highest price**, **lowest price**, and **adjusted closing price**. The data highlights the variability across stocks and trading days, with differences in prices and trading volumes reflecting unique market conditions. For instance, Google (GOOG) on 2024-07-18 had a closing price of $182.62, while Apple (AAPL) on 2023-03-13 exhibited significant movement, with a high of $153.14 and a low of $147.70. These variables collectively provide a robust foundation for modeling and forecasting daily percentage changes in stock prices.

3 Model

3.1 Model Overview

This analysis employs an **XGBoost (R Core Team 2023) regression model** to predict daily stock price changes, specifically focusing on the variable Price_Diff, which measures the difference in stock prices from one trading day to the next. By leveraging key predictors derived from historical data and stock market activity, the model identifies patterns and trends that inform short-term price movements. The XGBoost algorithm was selected for its efficiency and ability to handle large datasets, as well as its effectiveness in capturing nonlinear relationships and interactions among variables. Background details and diagnostics are included in Appendix B.

3.2 Model Assumptions

The XGBoost regression model used in this analysis relies on several key assumptions:

- **Independence of Observations**: The daily percentage changes in stock prices are assumed to be independent of each other. This implies that the prediction for one day does not influence or depend on the prediction for other days.
- **Relevance of Predictors**: The chosen predictors (e.g., `Price_Lag1`, `Price_Change_Percent`, `volume`, `open`, `high`, `low`, and `adjusted`) are assumed to adequately capture the key factors driving stock price changes. The model assumes no critical predictive variable is missing.
- **Stationarity of Features**: The statistical properties of the predictors, such as their mean and variance, are assumed to remain stable over time within the training and testing periods. This is crucial for the model to generalize well to future data.
- **No Multicollinearity**: The predictors are assumed to not be highly correlated with each other. Although gradient boosting models are robust to some multicollinearity, extreme collinearity could still negatively affect feature importance and model interpretation.
- **Consistency of Temporal Effects**: Temporal variables, such as `date` and derived features like day of the week, are assumed to capture stable patterns in stock price movements across the training and testing periods.
- **Additivity of Effects**: The model assumes that the combined effects of the predictors on the outcome variable (`Price_Diff`) can be captured additively through the boosting algorithm. Non-linear interactions are handled automatically by XGBoost, but this assumption helps guide the model design.
- **Complete Data**: The model assumes that missing data has been appropriately handled during preprocessing. This includes imputation, removal, or encoding of missing values.

These assumptions form the foundation of the modeling approach. Addressing potential violations, such as temporal dependencies or changes in feature distributions, will be discussed in the limitations section.

3.3 Model set-up

Define y_i as the daily price difference for a given stock on day i. The predictors include Price_Lag1_i, the lagged closing price, and other market and temporal features such as volume_i, open_i, high_i, low_i, and adjusted_i. The relationship is modeled as:

$$y_i = f(X_i) + \epsilon_i$$

where X_i is the set of predictors for observation i, $f(X_i)$ is the function learned by the XGBoost model, and ϵ_i represents the residual error.

The model parameters are optimized to minimize the following objective function:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} (y_i - f(X_i))^2 + \Omega(f)$$

where $\mathcal{L}(\Theta)$ is the loss function, Θ represents the model parameters, and $\Omega(f)$ is a regularization term to prevent overfitting.

3.4 Model justification

The XGBoost regression model is the most suitable approach for predicting daily price differences (**'Price_Diff'**) across a basket of tech stocks, including Google (**GOOG**), Apple (**AAPL**), Amazon (**AMZN**), and Microsoft (**MSFT**). By employing gradient boosting to iteratively minimize prediction errors, XGBoost effectively handles the complex, non-linear relationships inherent in stock price movements. It is particularly well-suited for continuous outcomes like **'Price_Diff'**, which exhibit significant variability driven by historical trends, market activity, and temporal effects. The model's ability to optimize squared error loss while incorporating regularization prevents overfitting and enhances generalizability. Additionally, XGBoost is robust to missing data, automatically manages variable interactions, and is computationally efficient, making it ideal for large datasets in financial analysis.

While alternative approaches, such as traditional linear regression models, could be used, they may not capture the complex interactions between predictors like `Price_Lag1`, `Price_Change_Percent`, and `volume`. Machine learning models like random forests or neural networks offer high flexibility but may sacrifice computational efficiency and interpretability compared to XGBoost.

3.5 Model Summary

The model's hyperparameters, set to optimize predictive accuracy and computational efficiency, are displayed in **?@tbl-model** results. It reveals 100 boosting rounds with a learning rate (η) of 0.1, balancing fast convergence and overfitting control. A maximum tree depth of 6 captures complex interactions among predictors without overfitting. Column and row sampling rates of 0.8 each ensure robustness by introducing randomness into feature and observation selection during tree construction.

Table 2: Summary of XGBoost Model Hyperparameters

Parameter	Value
Number of Trees (Rounds)	100
Learning Rate (eta)	0.1
Max Tree Depth	6
Column Sampling Rate	0.8
Row Sampling Rate	0.8
Objective Function	Squared Error Regression (reg:squarederror)

Furthermore, the model employs the squared error regression objective function (`reg:squarederror`), suitable for continuous outcome variables like `Price_Diff`. These hyperparameters enable the model to handle the non-linear relationships and variability inherent in financial data effectively.

4 Results

Our results are summarized in **?@tbl-modelresults**.

5 Discussion

5.1 First discussion point

If my paper were 10 pages, then should be be at least 2.5 pages. The discussion is a chance to show off what you know and what you learnt from all this.

5.2 Second discussion point

Please don't use these as sub-heading labels - change them to be what your point actually is.

5.3 Third discussion point

5.4 Weaknesses and next steps

Weaknesses and next steps should also be included.

Appendix

A Additional data details

B Model details

B.1 Posterior predictive check

In **?@fig-ppcheckandposteriorvsprior-1** we implement a posterior predictive check. This shows...

In **?@fig-ppcheckandposteriorvsprior-2** we compare the posterior with the prior. This shows...

B.2 Diagnostics

References

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.