Predicting Stock Price Changes Using Machine Learning Models

Student: Alen Sahinpasic, Professor: Petra Kralj Novak, TA: Bojan Evkoski
Institutions
Central European University
Quellenstrasse 51, 1100 Vienna, Austria
e-mail: Sahinpasic_Alen@student.ceu.edu

ABSTRACT

This report explores the application of machine learning techniques to predict the percentage change in Microsoft's adjusted stock prices. Utilizing historical stock data from 2010 to 2024, I implemented Linear Regression, Random Forest, and Gradient Boosting models to capture trends and variations. Hyperparameter tuning was conducted to optimize model performance, and the results were evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2). The analysis highlighted the Close-Open Difference and lagged percentage change features (e.g., Lag_1) as significant predictors. Gradient Boosting emerged as the most accurate model, achieving an R^2 of 0.434. Visualizations, including residual analysis and feature importance plots, provided valuable insights into model performance and limitations. This study demonstrates the potential of machine learning in financial time series forecasting, with implications for more robust stock market predictions and portfolio management strategies.

1 INTRODUCTION

The stock market is a dynamic and complex system influenced by numerous factors, making it a challenging domain for predictive modeling. Accurate stock price prediction can provide significant advantages for investors, traders, and financial institutions by aiding in decision-making and risk management. This project focuses on predicting percentage changes in Microsoft Corporation (MSFT) stock prices using machine learning models trained on historical stock data.

The primary challenge in stock price prediction lies in capturing the underlying patterns and trends from historical data while accounting for market volatility and noise. Traditional methods, such as technical and fundamental analysis, often rely on human expertise and may fail to generalize across different market conditions. Machine learning offers a data-driven approach to address these limitations, enabling the discovery of complex patterns and relationships in large datasets.

The objectives of this project are following:

1. To preprocess and engineer features from historical MSFT stock data, including lagged values and price differences, ensuring no data leakage.
2. To implement and evaluate machine learning models like Linear Regression, Random Forest, and Gradient Boosting for predicting percentage changes in stock prices.
3. To compare model performance using key metrics such as MSE, MAE, and R^2, and to analyze feature importance for interpretability.
4. To establish a baseline model (Lag_1) and assess its performance relative to other machine learning models.

Stock price prediction has been extensively studied in the field of financial analytics. Traditional methods, such as autoregressive integrated moving average (ARIMA) models and exponential smoothing, focus on time-series modeling but often struggle with capturing non-linear relationships. More recent approaches have utilized machine learning algorithms, including support vector machines (SVMs) and neural networks, which excel at handling complex, high-dimensional data. This project adopts a combination of feature engineering and machine learning to balance interpretability and predictive power.

The dataset was obtained using the Yahoo Finance API, covering daily MSFT stock data from 2010 to 2024. Key features, such as high-low price differences, close-open price differences, and lagged percentage changes, were engineered to capture relevant market dynamics. The data was split into training (2010–2022) and testing (2023) periods to simulate real-world forecasting scenarios.

Price	Adj Close	Close	High	Low	Open	Volume
Ticker	MSFT	MSFT	MSFT	MSFT	MSFT	MSFT
Date						
2010-01-04	23.300684	30.950001	31.100000	30.590000	30.620001	38409100
2010-01-05	23.308220	30.959999	31.100000	30.639999	30.850000	49749600
2010-01-06	23.165163	30.770000	31.080000	30.520000	30.879999	58182400
2010-01-07	22.924259	30.450001	30.700001	30.190001	30.629999	50559700
2010-01-08	23.082355	30.660000	30.879999	30.240000	30.280001	51197400

Figure 1

RandomizedSearchCV was employed to optimize hyperparameters for Random Forest and Gradient Boosting models, ensuring robust performance. Linear Regression served as a baseline machine learning model for comparison. Additionally, a baseline model using the previous day's percentage change (Lag_1) was implemented to contextualize model performance. The results of this study are intended to:

- Demonstrate the feasibility of machine learning for stock price prediction.
- Identify the most significant features influencing stock price changes.
- Provide actionable insights for improving financial forecasting models.
- Highlight the strengths and limitations of each modeling approach, offering guidance for future research in the domain.

2 DATA

This study leverages historical stock data for Microsoft Corporation (MSFT), spanning from 2010 to 2023, obtained via the Yahoo Finance API. The dataset includes daily stock prices, with features engineered to capture market dynamics, such as price differences and lagged percentage changes.

2.1 Data description

- Number of Instances: 3,511 (after preprocessing).
- Number of Attributes: 19.
    - Numerical: High, Low, Open, Close, Volume, Adjusted Close, High-Low Difference, Close-Open Difference, Lagged Features (Lag_1 to Lag_10).
    - Target Variable: Adjusted Close Percentage Change (Numerical).
- Distribution of Target Variable (Figure 2.2): The target variable exhibits a roughly normal distribution, as visualized in the histogram of percentage changes.
- Missing Values: Rows with missing values caused by lagged features and percentage change computation were removed.

The primary target variable, Adjusted Close Percentage Change, measures the daily percentage change in the adjusted closing price. This variable was chosen for its relevance in short-term stock market predictions.

Figure 2.1

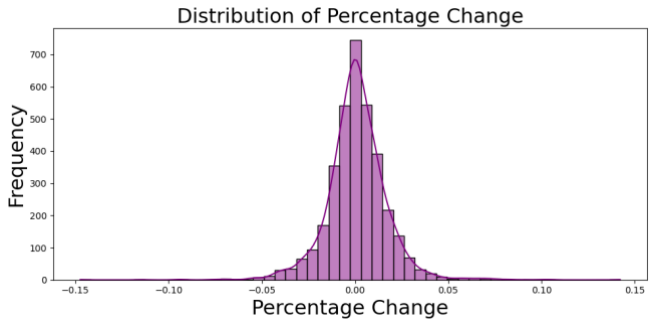

Figure 2.2

2.2 Data understanding

The target variable, Adj_Close_Pct_Change, plays a central role in this analysis as it captures the day-to-day percentage changes in adjusted close prices. Accurate predictions of this variable enable actionable financial insights and improved decision-making for traders and investors.

Among the predictors, Close_Open_Diff demonstrates the highest correlation (0.63) with the target variable, highlighting its significant influence on daily price changes. Lagged features (Lag_1 to Lag_10), while individually weaker in correlation, collectively contribute to capturing historical trends and dependencies essential for time-series modeling. High_Low_Diff, representing intraday price variability, also provides additional context, albeit with a weaker direct relationship to the target.

These relationships are visually captured in the correlation heatmap (Figure 2.3), which emphasizes the strong predictive role of Close_Open_Diff and the complementary value of lagged features in the model.

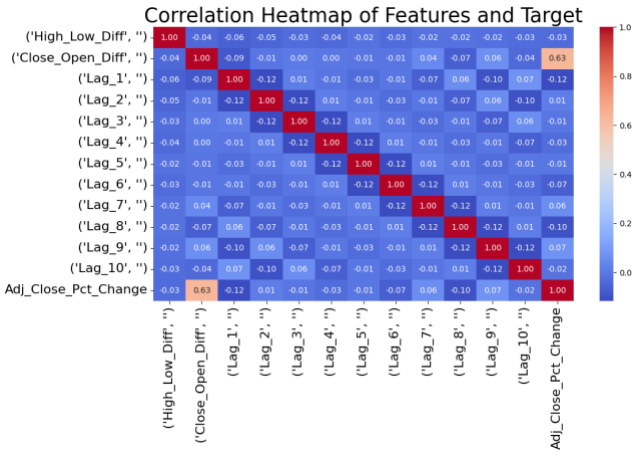

Figure 2.3

2.3 Data preprocessing

To ensure the dataset was suitable for machine learning, several preprocessing steps were undertaken. Feature engineering involved creating High_Low_Diff, representing the difference between the high and low prices, and Close_Open_Diff, capturing the difference between closing and opening prices. Additionally, lagged features (Lag_1 to Lag_10) were derived from the percentage change in adjusted close prices, providing temporal context.

Missing values, primarily caused by lagging and percentage change computations, were systematically removed to maintain data integrity. Normalization was applied to all numerical features, standardizing them to enhance model performance and prevent scale-induced bias.

These preprocessing steps were implemented using Python libraries such as Pandas and NumPy, ensuring efficiency and reproducibility. By carefully addressing data quality and preparing it for analysis, the dataset was optimized for predictive modeling while minimizing the risk of data leakage, thus enabling robust evaluation and reliable insights. I decided to manually split the training and test datasets to ensure the temporal order of the data is preserved and to prevent potential data leakage, which is a critical issue in time series forecasting. This approach helps maintain the integrity of the evaluation process by ensuring that future data does not influence model training

3. MACHINE LEARNING METHODS USED

In this project, I implemented three machine learning models to predict the percentage change in Microsoft stock prices: Linear Regression, Random Forest Regressor, and Gradient Boosting Regressor. These models were chosen because they represent a range of complexity, from simple linear models to more sophisticated ensemble approaches, enabling a comparative analysis of predictive performance.

3.1 Brief description of the methods used

Linear Regression - A simple and interpretable algorithm that assumes a linear relationship between features and the target variable. It uses coefficients to weigh each feature's contribution to the prediction. While it is less capable of capturing complex patterns, it serves as a strong baseline.

- Parameters: Linear Regression has no hyperparameters, but its coefficients highlight the importance of features. For example, lagged features (e.g., Lag_1, Lag_6) demonstrated significant influence in the predictions.

The feature importance analysis revealed that Lag_6 and Lag_1 were the most impactful coefficients. (Figure 3.1)

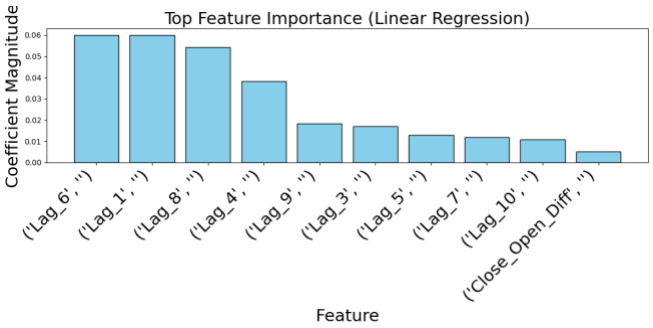

Figure 3.1

Random Forest - An ensemble method that constructs multiple decision trees and combines their outputs to improve accuracy and reduce overfitting. It is well-suited for capturing non-linear relationships in the data. Key Parameters:

- n_estimators: The number of trees in the forest. Increasing this generally improves accuracy but raises computational cost. Optimal value found: 200.
- max_depth: Limits the depth of each tree to prevent overfitting. Optimal value found: 20.
- min_samples_split: Controls the minimum samples required to split an internal node, influencing the model's complexity. Optimal value found: 10.

Feature importance analysis revealed that Close-Open Difference and Lag_1 contributed significantly to the model's predictive performance. (Figure 3.2)

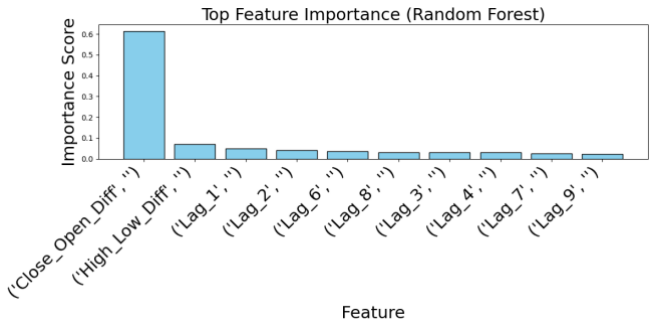

Figure 3.2

Gradient Boosting - Another ensemble technique that builds trees sequentially, with each tree correcting the errors of the previous ones. It often achieves high accuracy on structured data. Key Parameters:

- learning_rate: Controls the contribution of each tree to the final prediction. Lower values require more trees but improve generalization. Optimal value found: 0.1.
- n_estimators: Determines the number of trees in the sequence. Optimal value found: 200.
- max_depth: Limits tree depth to prevent overfitting. Optimal value found: 5.

Feature importance analysis similarly highlighted Close-Open Difference and Lag_1 as dominant predictor (Figure 3.3).

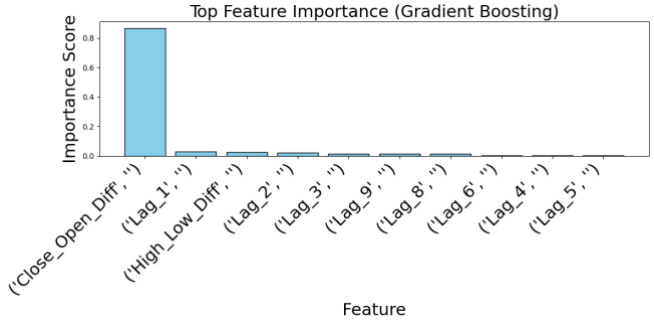

Figure 3.3

Hyperparameter tuning for Random Forest and Gradient Boosting was performed using RandomizedSearchCV. A time-series split was used to ensure the temporal order of data was respected during cross-validation.

3.2 Brief description of the evaluation criteria

To evaluate model performance, the following metrics were used:

1. Mean Squared Error (MSE): Measures the average squared difference between predictions and actual values. Lower values indicate better performance.
2. Mean Absolute Error (MAE): Captures the average absolute difference between predictions and actual values, providing an interpretable measure of error.
3. R-squared (R^2): Represents the proportion of variance in the target variable explained by the model. Higher values indicate better explanatory power.
4. A baseline model using a simple lag feature (Lag_1) was also included for comparison. This ensured that the machine learning models were evaluated relative to a naïve approach.

By using these metrics, the evaluation process balances interpretability, sensitivity to large errors, and a comparison against a meaningful benchmark, providing a robust and multidimensional understanding of model performance.

4. EXPERIMENTS

The experimental setup involved designing and evaluating machine learning models to predict the percentage change in Microsoft's adjusted stock prices. The goal was to determine how well various models could capture the trends in the data and outperform a simple baseline approach.

Experimental Setup - We employed three machine learning models: Linear Regression, Random Forest, and Gradient Boosting. Each model was trained using a combination of lagged percentage change features, alongside high-low and close-open differences. The data was split into training (2010-2022) and testing (2023) periods to ensure robust evaluation and avoid data leakage. Hyperparameter tuning was conducted for Random Forest and Gradient Boosting using RandomizedSearchCV with a time-series split cross-validation strategy.

Results - The performance of the models was evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) scores. The table below summarizes these metrics (Figure 4.1):

Model Performance Summary:

	Model	MSE	MAE	R^2
0	Linear Regression	0.000185	0.009933	0.292999
1	Random Forest	0.000149	0.008620	0.433218
2	Gradient Boosting	0.000148	0.008612	0.433846
3	Baseline (Lag_1)	0.000525	0.017955	-1.001218

Figure 4.1

The results indicate that both Random Forest and Gradient Boosting significantly outperformed the baseline, with Gradient Boosting achieving the lowest error and highest R^2 score.

Interpretation of Results - The findings demonstrate that ensemble models like Random Forest and Gradient Boosting are better suited for capturing complex relationships in stock price changes compared to Linear Regression. However, the Linear Regression model also performed reasonably well, showing its utility as a straightforward and interpretable baseline. The baseline model using the lagged feature (Lag_1) was the weakest, highlighting the added value of more sophisticated models.

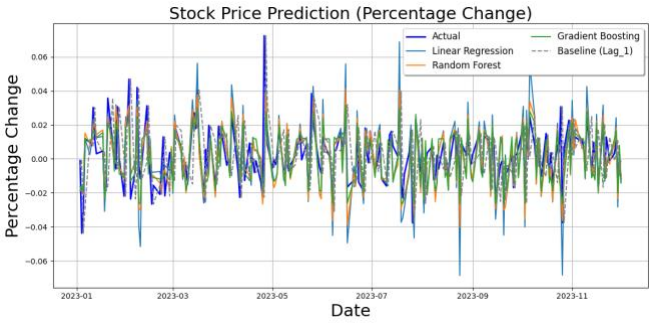

Figure 4.2

The bellow residual plot (Figure 4.3) provides a clear diagnostic of model performance, highlighting areas where predictions deviate from actual values and emphasizing the strengths of ensemble methods over simpler models like Linear Regression.

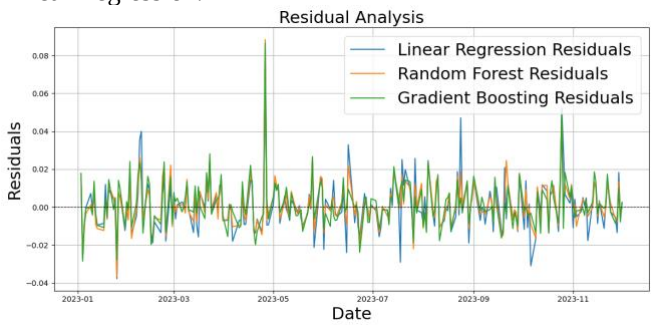

Figure 4.3

5. CONCLUSION

This project explored the use of machine learning models to predict percentage changes in Microsoft's adjusted stock prices, utilizing features derived from historical trends and market indicators. By employing Linear Regression, Random Forest, and Gradient Boosting, the study showcased the strengths and weaknesses of these approaches in capturing stock market dynamics.

The ensemble models Random Forest and Gradient Boosting demonstrated superior performance, significantly outperforming the baseline and Linear Regression. Feature importance analysis revealed that Close-Open Difference and lagged features, particularly Lag_1, were the most influential predictors across all models. Visualizations such as the Adjusted Close Price trend, correlation heatmap, and prediction comparisons provided clarity and contextual insights into the models' outputs.

The findings suggest that machine learning offers a valuable toolkit for stock market analysis, providing predictive accuracy and insights that go beyond simple heuristic methods. However, limitations such as the inability to capture unforeseen market shocks or incorporate external factors like news sentiment highlight areas for improvement.

Future work could explore integrating external datasets, such as economic indicators or sentiment analysis, to enhance the models' robustness. Additionally, experimenting with advanced techniques like deep learning or hybrid models may yield further improvements. Overall, this study underscores the potential and challenges of applying machine learning to financial time series prediction.

Evaluation criteria:

- The analyzed dataset needs to be non-trivial and large enough (thousands of examples, tens of attributes.)
- Use of a proper evaluation metric and method for the problem at hand
- Proper hyper-parameter tuning.
- Choice of baseline
- Interpretation of results
- Presentation

Paper template: https://is.ijs.si/?page_id=14415