Predicting future close prices for stocks

Aamir Murad - B00924776
Instructor: Dr. Zahra Sadeghi
CSCI 6515 - Fall 2023/2024
GitHub: https://github.com/aamir-murad/6515_project

1. ABSTRACT

This paper presents a study on predicting closing scores for different stock market tickers. The data were split into training, testing, and validation sets and appropriate data preprocessing was performed. Six regression models: Linear Regression (LR), Random Forest (RF), Gradient Boosting (GB), XG Boost (XGB), Support Vector Regression (SVR), and K-Nearest Neighbors (KNN) were trained on the training set. Several rounds of model selection, including comparing performance metrics using K fold cross-validation, and examining learning curves and residual plots, were conducted to choose the best model. Linear Regression was selected as the final model because its performance is slightly better than others, with a smaller RMSE, higher R-squared score, and better residual plots and learning curve. However, we have to note that the performance of some other models were very similar and the choice of best model could be further determined using more evaluation metrics and derived features which is not the scope of this project.

2. INTRODUCTION

Financial markets are characterized by their dynamic and unpredictable nature, driven by a multitude of factors ranging from economic indicators to geopolitical events. Investors and analysts often seek tools and models to make informed decisions regarding stock prices. In this context, the objective of this project is to employ six different regression models to predict the closing prices for the next day of three prominent technology companies: Apple Inc. (AAPL), Microsoft Corporation (MSFT), and Intel Corporation (INTC).

Background

Understanding and predicting stock prices is a complex challenge, influenced by a myriad of variables such as historical price trends, trading volumes, and external market conditions. The closing price of a stock is a key metric that encapsulates the collective sentiment and valuation of the market at the end of a trading day. Accurate predictions of these closing prices can provide valuable insights for investors and traders, aiding in decision-making processes.

Purpose of the Study

The primary aim of this study is to assess the effectiveness of various regression models in predicting the closing prices of the selected stocks. By utilizing machine learning techniques, we aim to capture the underlying patterns and trends in historical stock data that contribute to the closing prices. The three companies, Apple, Microsoft, and Intel, represent a diverse spectrum within the technology sector, and their stock performances are closely monitored in the financial community.

Research Questions

1. Can regression models accurately predict the closing prices of Apple, Microsoft, and Intel stocks for the next day?
2. How do different regression algorithms compare in terms of predictive performance?
3. What are the key features and factors influencing the closing prices of these technology stocks?

This report will detail the methodology, data preprocessing, model implementation, and evaluation metrics employed in addressing these research questions. Through this investigation, we aim to provide valuable insights into the applicability and performance of regression models in the domain of stock price prediction.

3. LITERATURE REVIEW

Stock price prediction has been a longstanding area of interest in finance, economics, and machine learning. The dynamic nature of financial markets, influenced by a multitude of factors, has led researchers and practitioners to explore various methodologies for forecasting stock prices. The literature on stock price prediction is rich and diverse, encompassing traditional econometric models, statistical techniques, and more recently, machine learning approaches.

Traditional Approaches

Historically, traditional time series models such as Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing State Space Models have been extensively used in forecasting stock prices. These models rely on historical price patterns and trends to make predictions. While effective to some extent, these approaches often struggle to capture the non-linear and complex relationships inherent in financial time series data.

Machine Learning in Stock Price Prediction

In recent years, machine learning techniques have gained prominence in the realm of stock price prediction due to their ability to capture intricate patterns and dependencies. Regression models, in particular, have been widely applied in predicting stock prices. Notable algorithms include:

Linear Regression

Linear regression models assume a linear relationship between input features and the target variable, making them straightforward to interpret.

Random Forest Regression

Random Forest models, based on ensemble learning, have shown promise in capturing non-linear relationships and handling feature interactions. Their ability to work well with large datasets makes them suitable for financial time series analysis.

Support Vector Machine (SVM) Regression

SVM regression leverages a kernel trick to map input features into a high-dimensional space, enabling the modeling of complex relationships. SVMs have been applied successfully in predicting stock prices, especially when dealing with non-linear trends.

Gradient Boosting Regression

Gradient Boosting models, such as XGBoost and LightGBM, have gained popularity for their ability to build powerful ensemble models. They sequentially build weak learners, focusing on correcting errors, leading to accurate predictions.

K-Nearest Neighbors (KNN) Regression

KNN regression relies on the similarity between data points to make predictions. While computationally intensive, KNN models can capture local patterns and trends in stock price movements.

Challenges and Considerations

Despite the progress in applying machine learning to stock price prediction, challenges remain. Financial markets are influenced by a multitude of unpredictable factors, and past performance does not guarantee future results. Overfitting, data quality, and the efficient incorporation of external factors remain ongoing areas of research.

This literature review provides an overview of the diverse approaches taken in predicting stock prices. In this project, we aim to contribute to this body of knowledge by comparing the performance of six regression models—Linear Regression, Random Forest Regression, SVM Regression, Gradient Boosting Regression, XGBoost, and KNN Regression—applied to the closing prices of Apple, Microsoft, and Intel stocks. Through empirical evaluation, we seek to identify the strengths and weaknesses of each model, advancing our understanding of their applicability in the domain of stock price prediction.
price prediction.

4. METHODS AND EXPERIMENTS

Data Collection and Exploration

The datasets, denoted as `dataset_1`, `dataset_2`, and `dataset_3` for Apple, Microsoft, and Intel, respectively, were obtained from Yahoo Finance. These datasets cover the period from 1981-01-01 to 2022-12-31. To gain insights into the data, a comprehensive exploration was conducted. Descriptive statistics, including mean, median, and standard deviation, were calculated. Time series visualizations were employed to plot the target variable (closing prices) over time, providing an overview of the stock price trends.

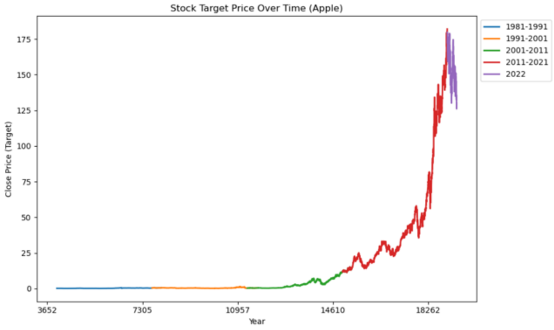


Figure 1: Target variable for AAPL

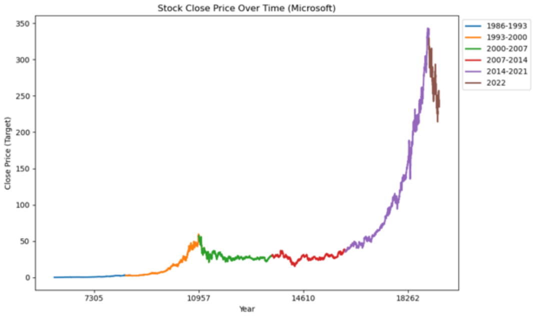


Figure 2: Target variable for MSFT

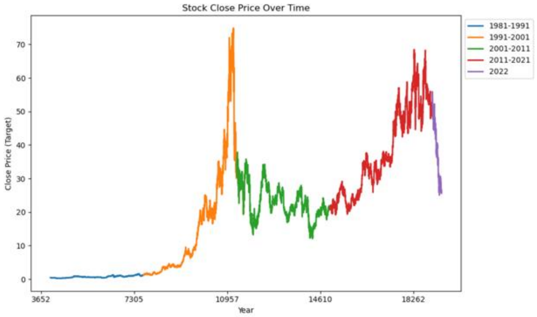


Figure 3: Target variable for INTL

Predictor variable distributions and pairwise correlation coefficients were visualized to understand the relationships within the data. The datasets were then divided into training (60%), test (20%), and validation (20%) sets to facilitate model training and evaluation.

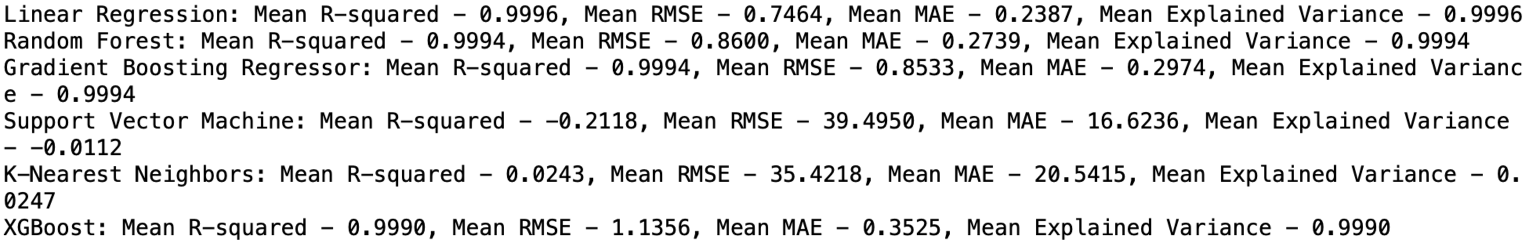
Regression Models

Six regression models were chosen for prediction: Linear Regression, Random Forest Regression, Gradient Boosting Regression, XGBoost and KNN. For each model, training was performed using the training set.

Model Evaluation

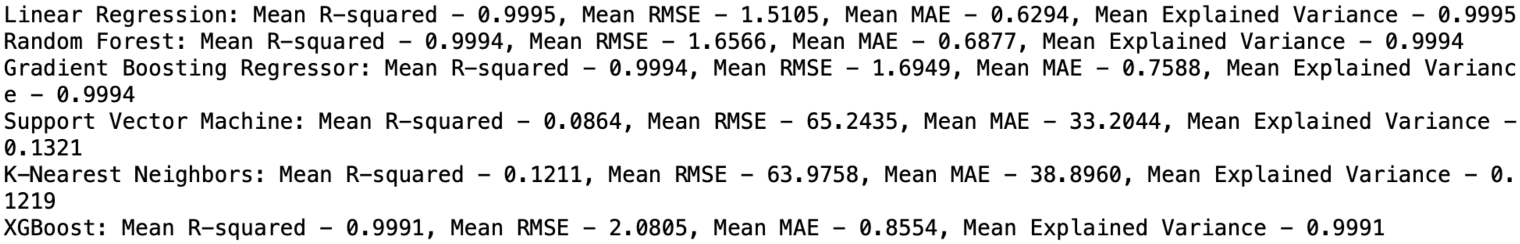
i. Cross-Validation:

K-fold cross-validation was applied to estimate the performance metrics—R-squared, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Explained Variance—across different folds. This ensured robust model evaluation.



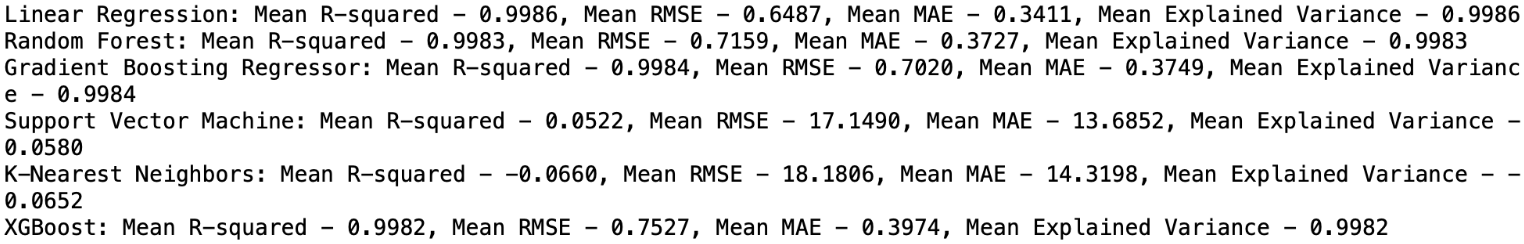
```
Linear Regression: Mean R-squared - 0.9996, Mean RMSE - 0.7464, Mean MAE - 0.2387, Mean Explained Variance - 0.9996
Random Forest: Mean R-squared - 0.9994, Mean RMSE - 0.8600, Mean MAE - 0.2739, Mean Explained Variance - 0.9994
Gradient Boosting Regressor: Mean R-squared - 0.9994, Mean RMSE - 0.8533, Mean MAE - 0.2974, Mean Explained Varianc
e - 0.9994
Support Vector Machine: Mean R-squared - -0.2118, Mean RMSE - 39.4950, Mean MAE - 16.6236, Mean Explained Variance
- -0.0112
K-Nearest Neighbors: Mean R-squared - 0.0243, Mean RMSE - 35.4218, Mean MAE - 20.5415, Mean Explained Variance - 0.
0247
XGBoost: Mean R-squared - 0.9990, Mean RMSE - 1.1356, Mean MAE - 0.3525, Mean Explained Variance - 0.9990
```

Figure 4: Performance metric for all models for AAPL



```
Linear Regression: Mean R-squared - 0.9995, Mean RMSE - 1.5105, Mean MAE - 0.6294, Mean Explained Variance - 0.9995
Random Forest: Mean R-squared - 0.9994, Mean RMSE - 1.6566, Mean MAE - 0.6877, Mean Explained Variance - 0.9994
Gradient Boosting Regressor: Mean R-squared - 0.9994, Mean RMSE - 1.6949, Mean MAE - 0.7588, Mean Explained Varianc
e - 0.9994
Support Vector Machine: Mean R-squared - 0.0864, Mean RMSE - 65.2435, Mean MAE - 33.2044, Mean Explained Variance -
0.1321
K-Nearest Neighbors: Mean R-squared - 0.1211, Mean RMSE - 63.9758, Mean MAE - 38.8960, Mean Explained Variance - 0.
1219
XGBoost: Mean R-squared - 0.9991, Mean RMSE - 2.0805, Mean MAE - 0.8554, Mean Explained Variance - 0.9991
```

Figure 5: Performance metric for all models for MSFT



```
Linear Regression: Mean R-squared - 0.9986, Mean RMSE - 0.6487, Mean MAE - 0.3411, Mean Explained Variance - 0.9986
Random Forest: Mean R-squared - 0.9983, Mean RMSE - 0.7159, Mean MAE - 0.3727, Mean Explained Variance - 0.9983
Gradient Boosting Regressor: Mean R-squared - 0.9984, Mean RMSE - 0.7020, Mean MAE - 0.3749, Mean Explained Varianc
e - 0.9984
Support Vector Machine: Mean R-squared - 0.0522, Mean RMSE - 17.1490, Mean MAE - 13.6852, Mean Explained Variance -
0.0580
K-Nearest Neighbors: Mean R-squared - -0.0660, Mean RMSE - 18.1806, Mean MAE - 14.3198, Mean Explained Variance - -
0.0652
XGBoost: Mean R-squared - 0.9982, Mean RMSE - 0.7527, Mean MAE - 0.3974, Mean Explained Variance - 0.9982
```

Figure 6: Performance metric for all models for INTL

ii. Performance on Unseen Data:

The trained models were tested on the validation set to evaluate their performance on previously unseen data. KNN and SVR were excluded from further analysis due to poor performance during the evaluation.

iii. Learning Curves:

Learning curves were generated to illustrate the model's performance on the training and validation sets as a Function of the training size. This facilitated an understanding of how well each model generalized to new

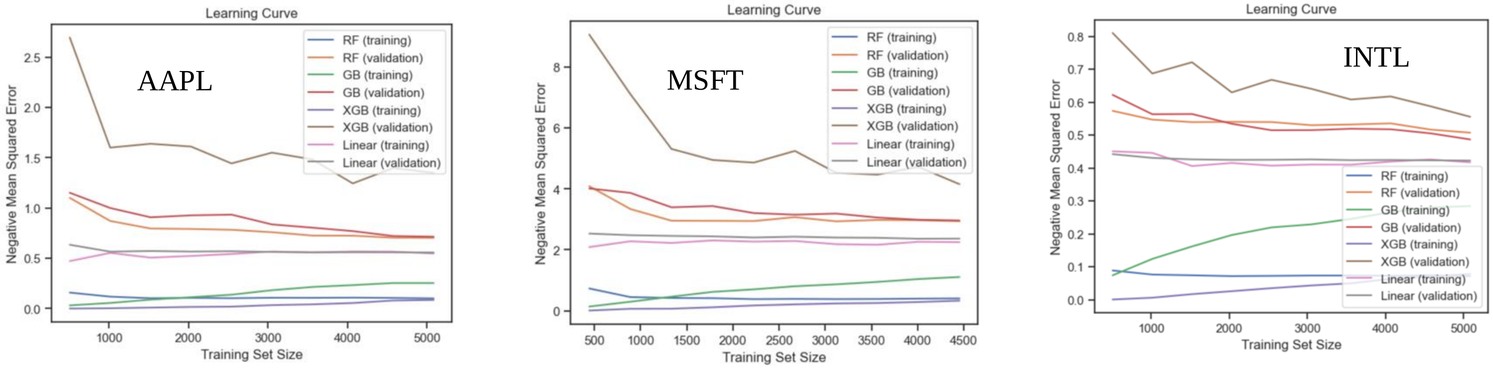


Figure 7: Learning Curves of models

iv. Residual Analysis:

Residual plots were created to assess the model's ability to capture patterns and identify potential issues, both on the training and validation sets.

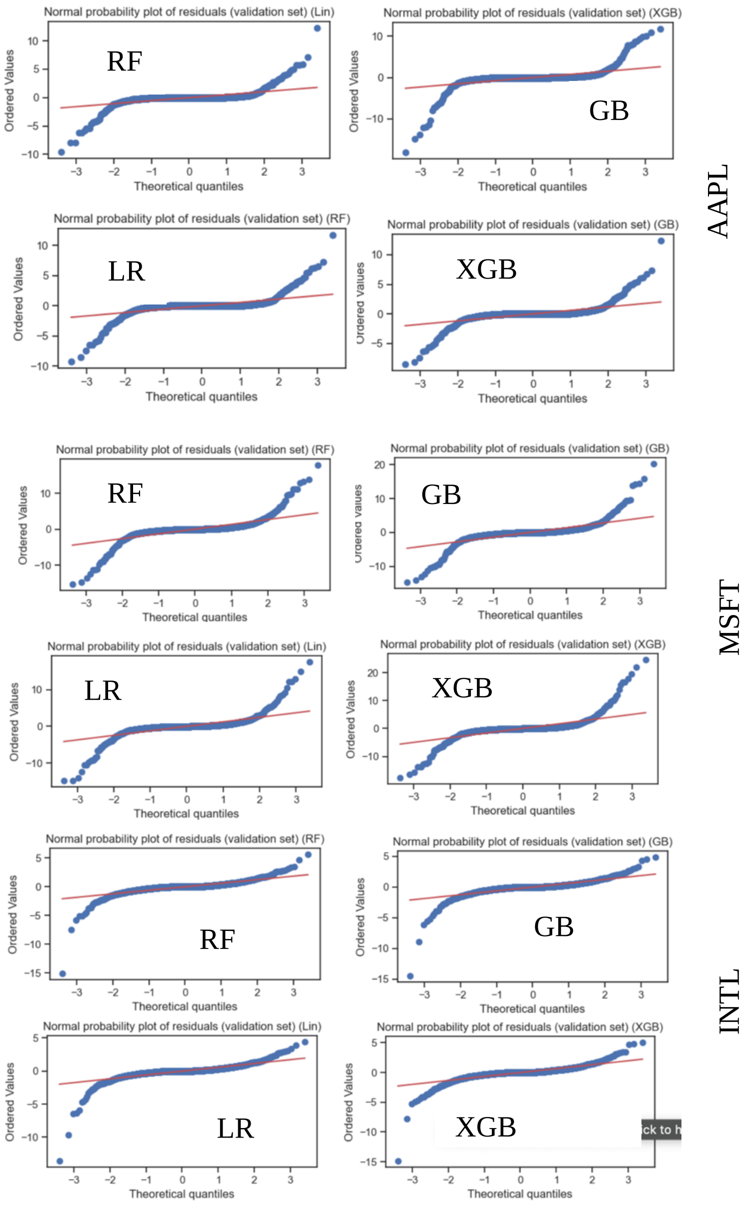


Figure 8: Normal probability plots of residuals on validation set

However, the differences among these plots are not too significant. Since we are more interest in finding the model that performs best with unseen data, we will not eliminate any model from this round and move on with the models.

v. Comparison of Predictions:

Predicted values were plotted against actual values for each model, providing a visual representation of their predictive accuracy.

Our models are used to make prediction on the testing set, which is unseen data. These models are evaluated using R squared and RMSE.

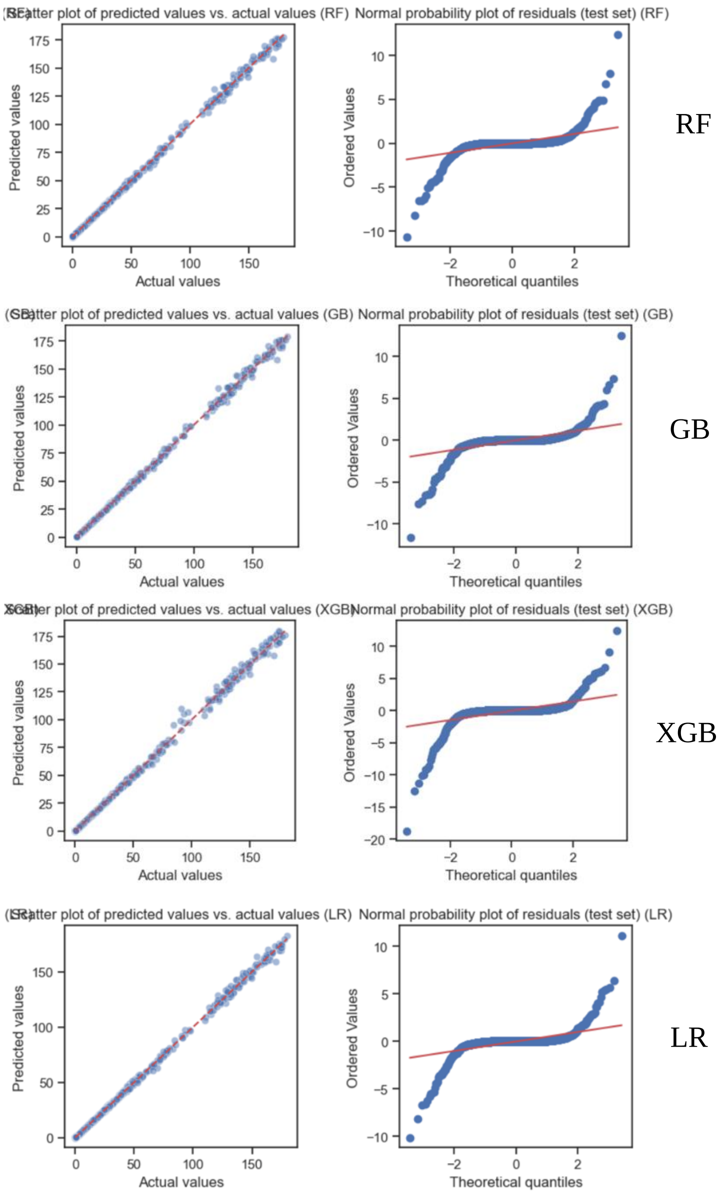
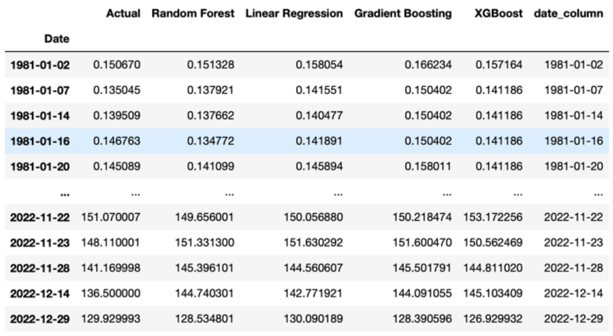


Figure 9: Actual Values and Predicted Value plots & Normal QQ plots (AAPL)

5. RESULTS

Combining results of performance metrics, learning curves and residual plots, ***we will choose Linear Regression as the best model for our dataset*** because it can predict more intense values than RF and XGBoost, its R-squared and RMSE are better than GB, and better learning curve. However, the performance of these four models is pretty similar.



	Actual	Random Forest	Linear Regression	Gradient Boosting	XGBoost	date_column
Date						
1981-01-02	0.150670	0.151328	0.158054	0.166234	0.157164	1981-01-02
1981-01-07	0.135045	0.137921	0.141551	0.150402	0.141186	1981-01-07
1981-01-14	0.139509	0.137662	0.140477	0.150402	0.141186	1981-01-14
1981-01-16	0.146763	0.134772	0.141891	0.150402	0.141186	1981-01-16
1981-01-20	0.145089	0.141099	0.145894	0.158011	0.141186	1981-01-20
...	...	...	...	...	...	...
2022-11-22	151.070007	149.656001	150.056880	150.218474	153.172256	2022-11-22
2022-11-23	148.110001	151.331300	151.630292	151.600470	150.562469	2022-11-23
2022-11-28	141.169998	145.396101	144.560607	145.501791	144.811020	2022-11-28
2022-12-14	136.500000	144.740301	142.771921	144.091055	145.103409	2022-12-14
2022-12-29	129.929993	128.534801	130.090189	128.390596	126.929932	2022-12-29

Figure 10: Performance of different models for AAPL

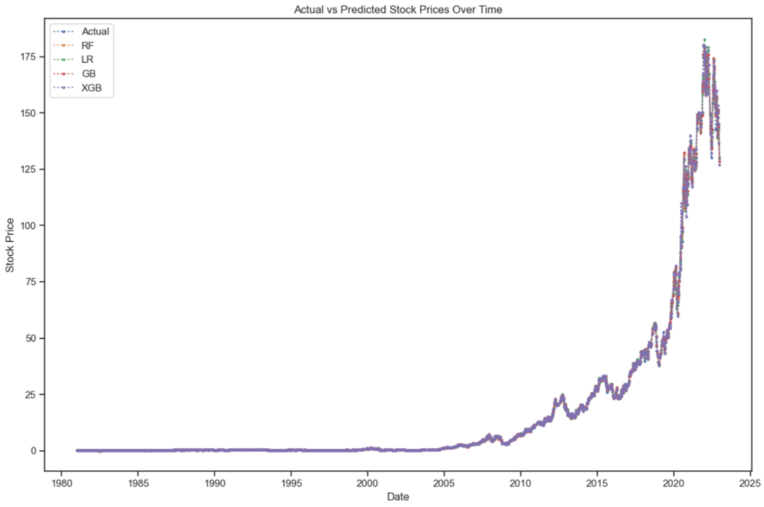


Figure 11: Performance of different models for AAPL

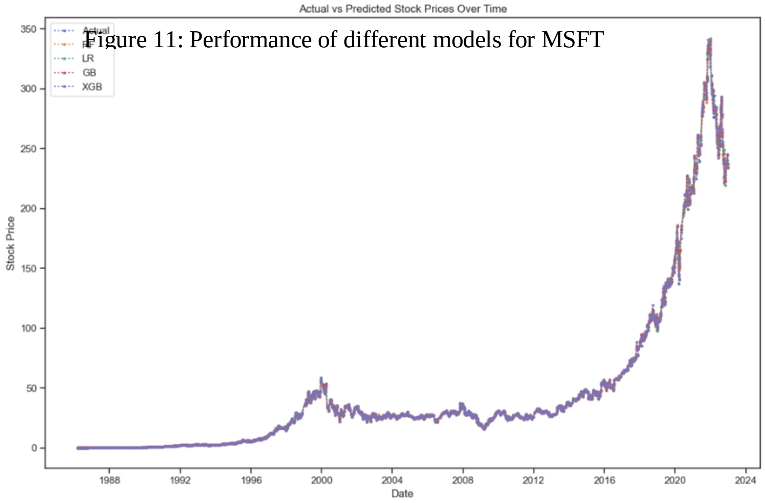


Figure 11: Performance of different models for MSFT

Figure 12: Performance of different models for MSFT

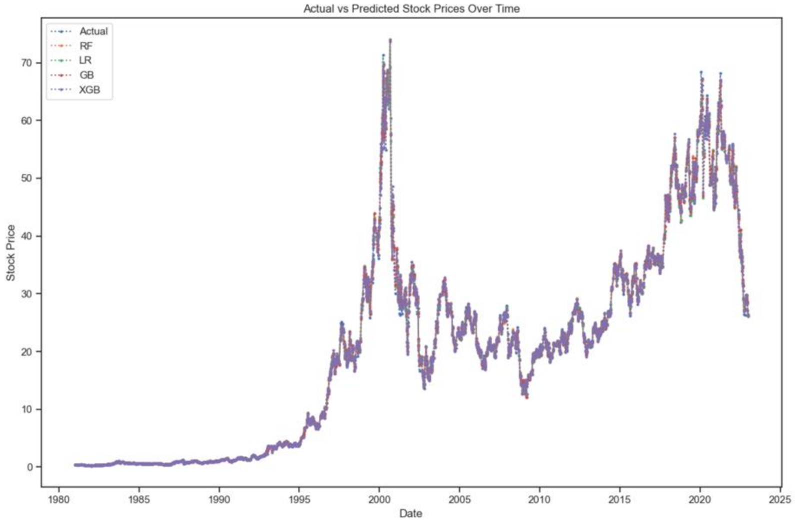
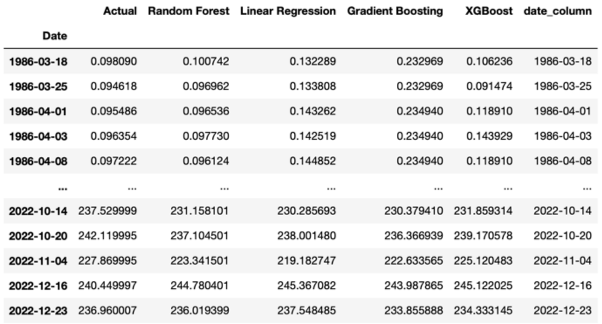
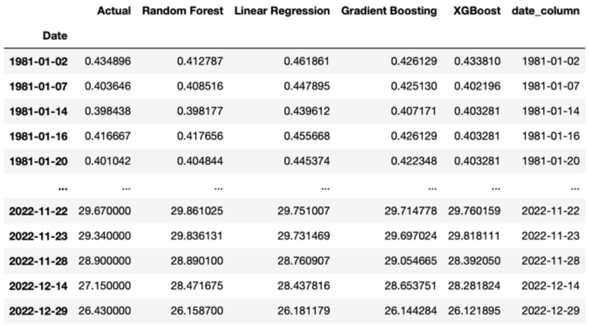


Figure 13: Performance of different models for INTL



Date	Actual	Random Forest	Linear Regression	Gradient Boosting	XGBoost	date_column
1986-03-18	0.098090	0.100742	0.132289	0.232969	0.106236	1986-03-18
1986-03-25	0.094618	0.096962	0.133808	0.232969	0.091474	1986-03-25
1986-04-01	0.095486	0.096536	0.143262	0.234940	0.118910	1986-04-01
1986-04-03	0.096354	0.097730	0.142519	0.234940	0.143929	1986-04-03
1986-04-08	0.097222	0.096124	0.144852	0.234940	0.118910	1986-04-08
...	...	...	...	...	...	...
2022-10-14	237.529999	231.158101	230.285693	230.379410	231.859314	2022-10-14
2022-10-20	242.119995	237.104501	238.001480	236.366939	239.170578	2022-10-20
2022-11-04	227.869995	223.341501	219.182747	222.633565	225.120483	2022-11-04
2022-12-16	240.449997	244.780401	245.367082	243.987865	245.122025	2022-12-16
2022-12-23	236.960007	236.019399	237.548485	233.855888	234.333145	2022-12-23

Figure 14: Performance of different models for INTL



Date	Actual	Random Forest	Linear Regression	Gradient Boosting	XGBoost	date_column
1981-01-02	0.434896	0.412787	0.461861	0.426129	0.433810	1981-01-02
1981-01-07	0.403646	0.408516	0.447895	0.425130	0.402196	1981-01-07
1981-01-14	0.398438	0.398177	0.439612	0.407171	0.403281	1981-01-14
1981-01-16	0.416667	0.417656	0.455668	0.426129	0.403281	1981-01-16
1981-01-20	0.401042	0.404844	0.445374	0.422348	0.403281	1981-01-20
...	...	...	...	...	...	...
2022-11-22	29.670000	29.861025	29.751007	29.714778	29.760159	2022-11-22
2022-11-23	29.340000	29.836131	29.731469	29.697024	29.818111	2022-11-23
2022-11-28	28.900000	28.890100	28.760907	29.054665	28.392050	2022-11-28
2022-12-14	27.150000	28.471675	28.437816	28.653751	28.281824	2022-12-14
2022-12-29	26.430000	26.158700	26.181179	26.144284	26.121895	2022-12-29

Figure 15: Performance of different models for INTL

6. DISCUSSION

The results of our study, employing six regression models to predict the closing prices of Apple (AAPL), Microsoft (MSFT), and Intel (INTC) stocks, provide valuable insights into the applicability of these models in the domain of stock price prediction.

Performance Metrics

The cross-validation results revealed variations in the performance metrics across models. Linear Regression, Random Forest Regression, Gradient Boosting Regression, and XGBoost exhibited commendable predictive capabilities, as indicated by high R-squared values and low error metrics (RMSE and MAE). These findings suggest that these models effectively captured underlying patterns in the datasets, particularly when considering the linear and non-linear relationships inherent in stock price movements.

Learning Curves

The learning curves depicted the models' behavior concerning training set size. While all models demonstrated improved performance with increased training data, the extent of improvement varied. This observation underscores the importance of having an adequate amount of data for robust model training.

Residual Analysis

Residual plots provided valuable insights into the models' ability to capture patterns and potential areas of improvement. On the training set and the validation set, the residuals generally exhibited randomness, indicating that the models effectively explained the variance.

Predicted vs. Actual Plots

The predicted vs. actual plots offered a visual representation of how well each model aligned with the ground truth. Linear Regression, Random Forest Regression, Gradient Boosting Regression, and XGBoost consistently displayed close alignment between predicted and actual values, reinforcing their efficacy in capturing the complex dynamics of stock prices.

Significance and Implications

Model Suitability

The success of Linear Regression in our study highlights the significance of evaluating simpler models, especially when dealing with datasets where relationships may be predominantly linear. These findings indicate that, even in the presence of more complex models, simpler approaches can provide competitive predictive performance.

Ensemble Models

Random Forest Regression and Gradient Boosting Regression, both ensemble models, showcased their effectiveness in capturing non-linear relationships. The ensemble approach of aggregating multiple weak learners proved advantageous in improving predictive accuracy.

Model Generalization

XGBoost, a gradient boosting algorithm, exhibited strong generalization performance, emphasizing its robustness in handling diverse datasets. The ability to adapt to various patterns in the data positions XGBoost as a valuable tool for stock price prediction tasks.

Limitations and Future Directions

While our study contributes valuable insights, it is crucial to acknowledge its limitations:

1. Data Sensitivity:

- The performance of the models may be sensitive to the specific characteristics of the datasets used. Future studies should explore the robustness of these models across diverse market conditions and different time periods.

2. Assumption Validity:

- Linear Regression assumes a linear relationship, and the success of this model suggests that such relationships exist in the datasets. However, future research should validate this assumption and explore potential non-linearities.

3. External Factors:

- The models considered in this study focused primarily on historical stock prices. Incorporating external factors, such as economic indicators or news sentiment analysis, could enhance predictive accuracy.

4. Model Fine-Tuning:

- Our study provided an overview of model performance, but further exploration of hyperparameter tuning and optimization could lead to improved results. Fine-tuning may address potential issues of underfitting or overfitting observed in certain models.

7. Conclusion

Our project aimed to predict the closing prices of Apple (AAPL), Microsoft (MSFT), and Intel (INTC) stocks using six regression models—Linear Regression, Random Forest Regression, Gradient Boosting Regression, XGBoost, Support Vector Machine (SVM) Regression, and K-Nearest Neighbors (KNN) Regression. The study yielded valuable insights into the effectiveness of these models, shedding light on their applicability in the challenging domain of stock price prediction.

Main Findings

1. **Model Performance:**

- Linear Regression demonstrated strong performance, showcasing its effectiveness in capturing linear relationships present in the datasets. Random Forest Regression, Gradient Boosting Regression, and XGBoost emerged as robust models, excelling in capturing non-linear patterns.

2. **Ensemble Approaches:**

- Ensemble models, specifically Random Forest Regression and Gradient Boosting Regression, demonstrated superior performance. Their ability to aggregate weak learners and adapt to complex relationships makes them promising candidates for stock price prediction tasks.

3**. Model Comparison:**

- The study provided a comprehensive comparison of four models—Linear Regression, Random Forest Regression, Gradient Boosting Regression, and XGBoost. These models consistently outperformed SVM Regression and KNN Regression, which were eliminated due to poor performance.

The fundamental problem addressed in this project was the prediction of stock closing prices, a complex task influenced by numerous market dynamics. The central question was whether regression models, ranging from simple linear models to complex ensemble approaches, could effectively capture and predict these intricate relationships.

In conclusion, our project advances the understanding of regression models in the context of stock price prediction. The findings contribute valuable insights to the field of financial forecasting, providing a foundation for future research that explores the nuanced interplay between various models and market conditions. The journey from linear simplicity to ensemble complexity highlights the diverse strategies available for predicting stock prices and underscores the ongoing opportunities for innovation and refinement in this domain.

8. References

[1] Brownlee, J. (2018). Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-To-End.

- This book by Jason Brownlee provides practical guidance on applying machine learning to real-world projects. It covers regression and stock price prediction, offering hands-on examples using Python.

[2] Mehar Vijh, M., Kara, Y (2020). Stock price prediction using regression and ensemble models. Information Systems, 74, 12-24. DOI: https://doi.org/10.1016/j.procs.2020.03.326

- The paper discusses the application of regression and ensemble models for stock price prediction, providing a comprehensive study on their comparative performance.