Prophet Project

Authors: Connor Tomchin, Emmanuel Epau, Lakshmi Krishnamurthy, Chamroeun Chhay, Aziz Al Mezraani, Abhigyan Ghosh

Introduction

Through this project, we are attempting to build a predictive model to make forecasts using time-series data. Our primary goal is to leverage various machine learning models to make forecasts on IBM, Apple, Google, and Microsoft's stock prices. We explored different models including Linear Regression, Backward Selection, xGBoost, Random Forest, and the Prophet model and applied them on stock data sourced from Wharton Research Data Services.

Forecasting financial time series has been a challenging endeavor for both academia and businesses. Advances in forecasting methods have evolved from using traditional techniques to leveraging machine learning and deep learning models. So the problem at hand involves forecasting stock prices for top tier technology companies using time series data. This is an interesting problem because accurate forecasting of stock prices is important to investors as it allows them to make informed data-driven financial decisions. It helps investors and financial analysts anticipate future market movements, optimize investment strategies, and take necessary risk management measures, because of the volatility of financial data.

Related Work

ML models, including ARIMA, LSTM (Long Short-Term Memory), Artificial Neural Network (ANN), Logit model, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and K-Nearest Neighbor (KNN) have been widely used for stock price prediction, enabling investors and analysts to make informed decisions. However, many of these methods struggle with specific challenges such as capturing seasonality, sudden market shifts, and handling non-linear relationships in the data.

Recent research highlights the growing potential of Facebook's Prophet model, a time series forecasting tool, which has demonstrated promising results across various industries. Prophet is particularly effective because of its flexibility in handling missing data, its ability to incorporate seasonal components and external events, and its ease of use. For example, Prophet has been successfully applied to financial time series data from indices such as SP500, Dow Jones Industrial Average (DJIA), China Securities Index (CSI300), and others. Prophet bridges these gaps by providing a scalable and interpretable forecasting method that is accurate and usable, making it particularly suitable for stock price forecasting.

This project builds upon previous research by applying ML models like Linear Regression, Backward Selection, xGBoost, Random Forests, and Prophet to forecast future closing stock prices for companies such as IBM, Apple, Google, and Microsoft.

Data Description

The datasets we are using for this project consist of historical daily stock price data for IBM, Apple, Google, and Microsoft acquired via the Wharton Research Data Services which contains datasets useful for business research. The dataset is retrieved in CSV format, making it compatible with preprocessing in R and other analytical tools, and is vital for developing predictive models to forecast future stock performance. It encompasses daily stock prices over the past 10 years- from 2015 to 2024. Preprocessing steps included formatting the date formats and adding a dependent variable 'y' using a for loop to make predictions up to 30 days. We segmented our dataset into training (records from Jan 1 2015 - Dec 31 2019) and test data (Jan 1 2020 - Dec 31 2024). We also removed the features with the character data types, such as the millisecond data because they did not have any significant impact on the final output. We also undertook additional preprocessing steps for the models if they needed additional data manipulations. By leveraging Prophet's strengths, we were also able to make predictions for IBM, Apple, Google, and Microsoft. Prophet also helped automate the training and testing steps, handle missing data and capture recurring patterns, this approach aims to deliver interpretable, robust forecasts. We also used TipRanks data as a benchmark to compare the performance of our data.

The dataset includes columns capturing key metrics related to stock trading. The DATE column represents trading dates, while symbol columns identify stocks. The Buy/Sell metrics provide the number, volume, and dollar value of trades, and the Price metrics include average and volume-weighted prices. The Order execution details and trade timing details cover trade sizes, prices, and timing for opening and closing trades. Midpoint prices and market depth and spread measure bid-ask dynamics. The Realized spread and price impact analyze execution quality, while odd lot trades and ISO trades capture specific trade types. The Retail and institutional metrics differentiate trading behavior, and the remaining columns include volatility and market concentration measures.

Description: df [6 × 184]

	DATE <date>	BuyNumTrades_LR <int>	SellNumTrades_LR <int>	total_trade <int>	BuyVol_LR <int>	SellVol_LR <int>	total_vol <int>
1	2015-01-02	18894	17671	36565	2365283	2114692	4479975
2	2015-01-05	18076	18896	36972	1992241	2243183	4235424
3	2015-01-06	25593	26063	51656	2851206	3033871	5885077
4	2015-01-07	18210	19007	37217	2115552	2213166	4328718
5	2015-01-08	16725	15545	32270	2012165	1876768	3888933
6	2015-01-09	16485	17557	34042	2000801	2112262	4113063

6 rows | 1-8 of 184 columns

```
'data.frame':   2490 obs. of  184 variables:
 $ DATE                       : Date, format: "2015-01-02" "2015-01-05" "2015-01-06" ...
 $ BuyNumTrades_LR            : int  18894 18076 25593 18210 16725 16485 15453 18412 17743 15391 ...
 $ SellNumTrades_LR           : int  17671 18896 26063 19007 15545 17557 17446 17315 20106 15391 ...
 $ total_trade                : int  36565 36972 51656 37217 32270 34042 32899 35727 37849 30782 ...
 $ BuyVol_LR                  : int  2365283 1992241 2851206 2115552 2012165 2000801 1784498 2104986
1965350 1682902 ...
 $ SellVol_LR                 : int  2114692 2243183 3033871 2213166 1876768 2112262 2139623 2032814
2421673 1931733 ...
 $ total_vol                  : int  4479975 4235424 5885077 4328718 3888933 4113063 3924121 4137800
4387023 3614635 ...
 $ avg_buy_price_LR           : num  162 160 157 155 158 ...
 $ avg_sell_price_LR          : num  162 160 157 155 158 ...
 $ buy_dv_LR                  : num  3.84e+08 3.18e+08 4.47e+08 3.28e+08 3.18e+08 ...
 $ sell_dv_LR                 : num  3.44e+08 3.58e+08 4.76e+08 3.43e+08 2.97e+08 ...
 $ total_dv_LR                : num  7.28e+08 6.77e+08 9.23e+08 6.71e+08 6.15e+08 ...
 $ vwavg_buy_price_LR         : num  162 160 157 155 158 ...
 $ vwavg_sell_price_LR        : num  162 160 157 155 158 ...
 $ CSize                      : int  934101 600258 209948 329524 212566 328635 205181 172318 224715 233916
...
```

Methods

To tackle the challenge of forecasting stock prices for IBM, Apple, Google, and Microsoft, we relied on the Prophet time-series forecasting model, comparing it to linear regression, backward selection, xGBoost and Random Forest to ensure a robust approach. Even though the Random forest model offered us the lowest RMSE value, Prophet stood out as our final choice because the difference in error was negligible as compared to Random forests. Prophet is also the better model due to its unique ability to model complex seasonal patterns, handle missing data seamlessly, and deliver highly interpretable results, which is an important feature for financial forecasting.

We acquired the stock prices through the Wharton Research Data Services which we carefully preprocessed. Then we used the Prophet model to capture the seasonality in the data and ultimately generated forecasts to predict future stock price movements. We validated these forecasts against historical data to ensure accuracy, refining the model iteratively as needed. Prophet's adaptability made it an ideal fit for this project, especially in capturing the cyclical nature of financial markets influenced by trends like quarterly earnings and broader economic conditions. Its balance between flexibility and ease of use allowed us to quickly implement changes while leveraging external variables to enrich the analysis. Using Prophet, we were able to go a step further and make predictions for 500 days out for IBM, Apple, Google, and Microsoft. The results of the Prophet mode were compared to price targets from TipRanks, a website that aggregates price targets from analysts and displays the highest average and lowest average price targets among them for companies. However, it has its limitations, especially because we are unable to see how the model actually works and what variables it takes into account or make any changes in variable selection.

Despite these challenges, the interpretability and reliability of the output gave us a clear advantage, enabling us to uncover actionable insights into stock price behavior while

maintaining a strong connection to real-world applications. Prophet was a great tool for forecasting and helped to better understand the dynamics of financial markets.

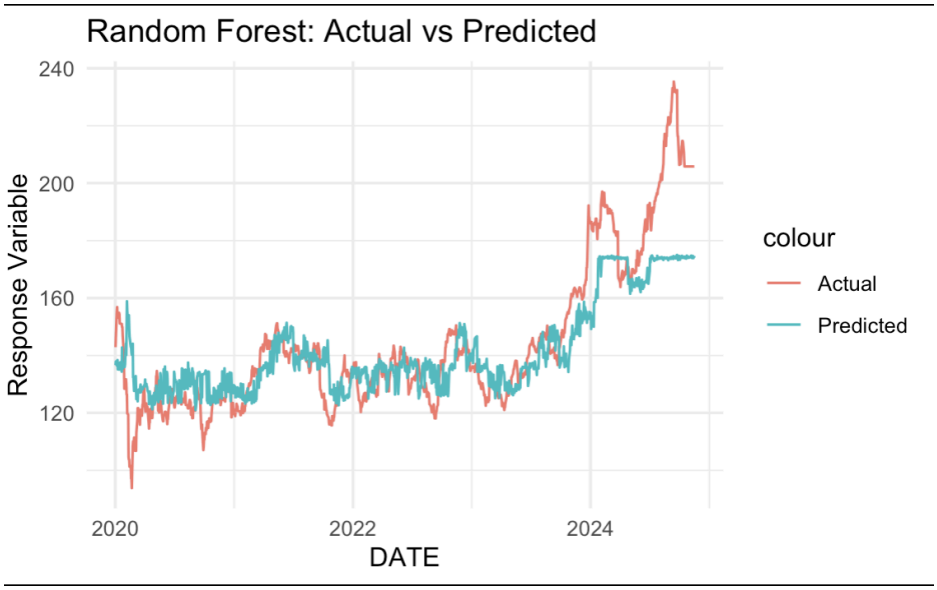
Results

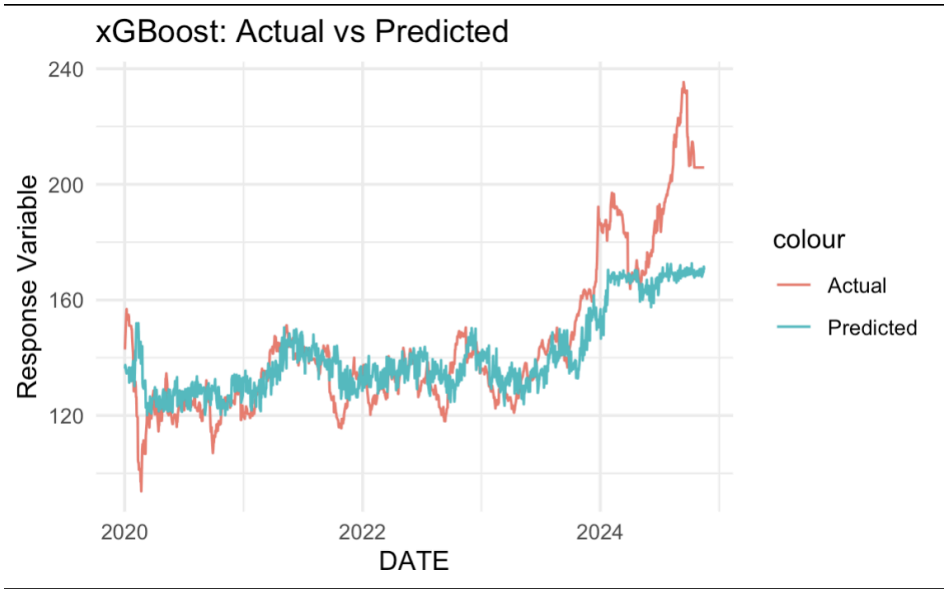
For the linear regression and backward selection, all we did was input the training data into the models and let them run. Then we predicted using the model with the test data and extracted the error. The RMSE that resulted was much higher than the other models as seen by the bar graph thus revealing it performed the worst out of the bunch.

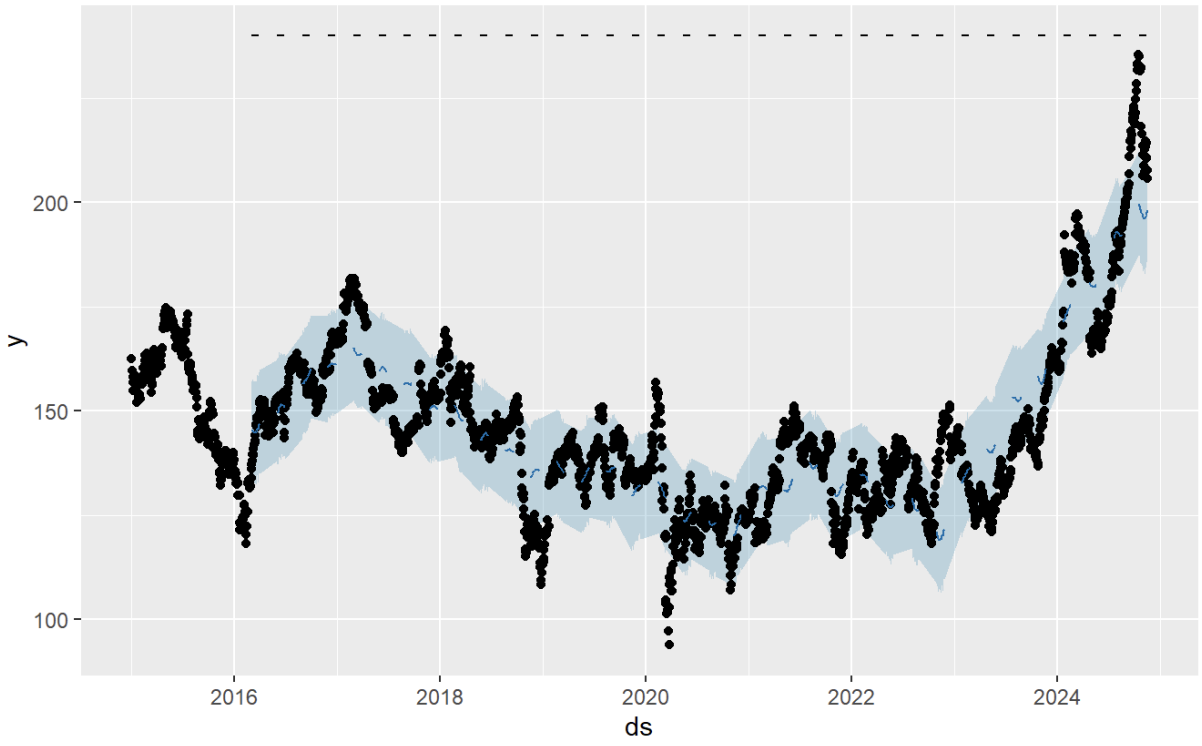
In the Random Forest and the xGBoost, we did some tuning and cross validation through different iterations to find the best models. For xGBoost, the parameters for the tuning process are eta, max depth, minimum_child_weight, gamma, subsample, colsample_bytree and nrounds. For the Random Forest, the parameters are ntree, nodesize and mtry. Afterward, we used the models to predict using the test data. The RMSE for the Random Forest and xGBoost are 14.82 and 15.93, respectively.

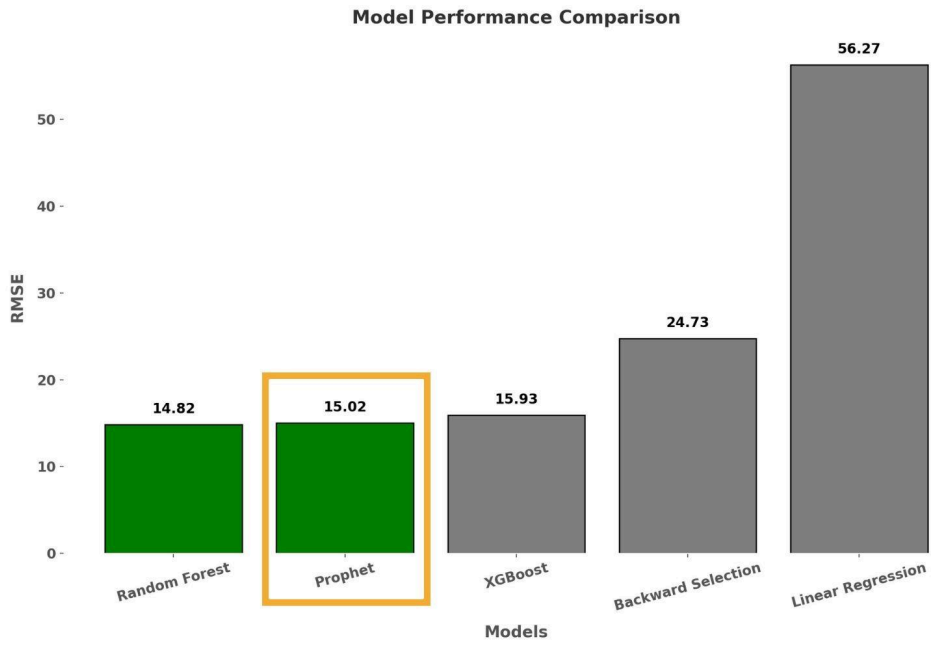
The prophet model backtests automatically so there is no split training and test data for it. The outcomes, or y, are the closing prices for the stock prices. It had an RMSE of 15.02, which was the second lowest behind the Random forest model. RMSE, or root mean square error, is a measure of predicted values and actual output values for a model. The prophet model performed well because it takes into account trends, which it classifies as non-periodic changes, as well as seasonality, which are periodic changes such as daily, weekly, or yearly patterns.

Several challenges were faced during the process. For example, the tunings of xGBoost and Random Forest took a long time, especially Random Forest. Additionally, we were unable to tune Prophet effectively due to our lack of knowledge of the model and its black box characteristics. Moreover, we weren’t able to extract the important features of the prophet again due to the model's complexity. Lastly, the lack of data (macroeconomic data, financial statements, etc.) did not allow us to predict yearly effectively for the models other than Prophet.

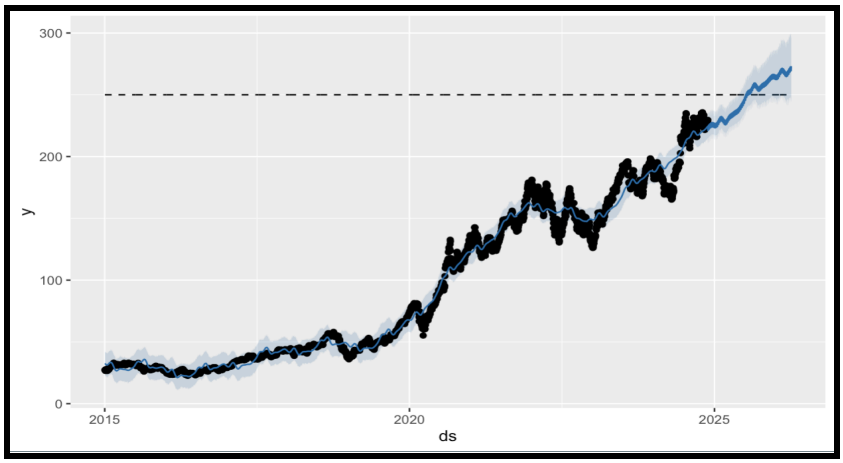

Random Forest: Actual vs Predicted
240
200
160
120
Response Variable
2020
2022
2024
DATE
colour
Actual
Predicted


xGBoost: Actual vs Predicted
240
200
160
120
Response Variable
2020
2022
2024
DATE
colour
Actual
Predicted

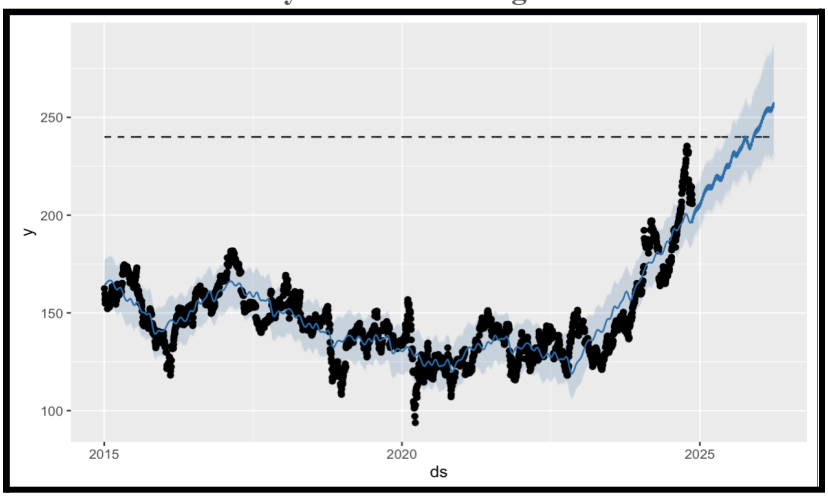




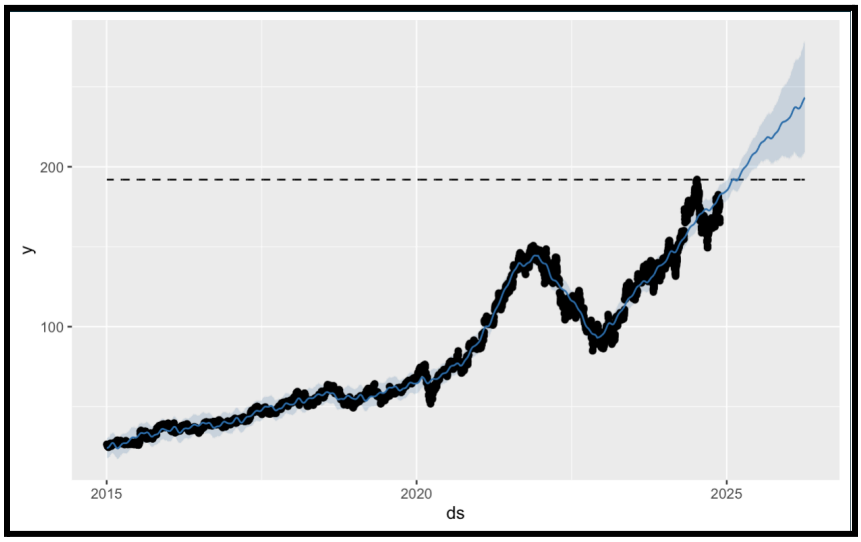
Apple- 500 days out- Price Target of 274.9434



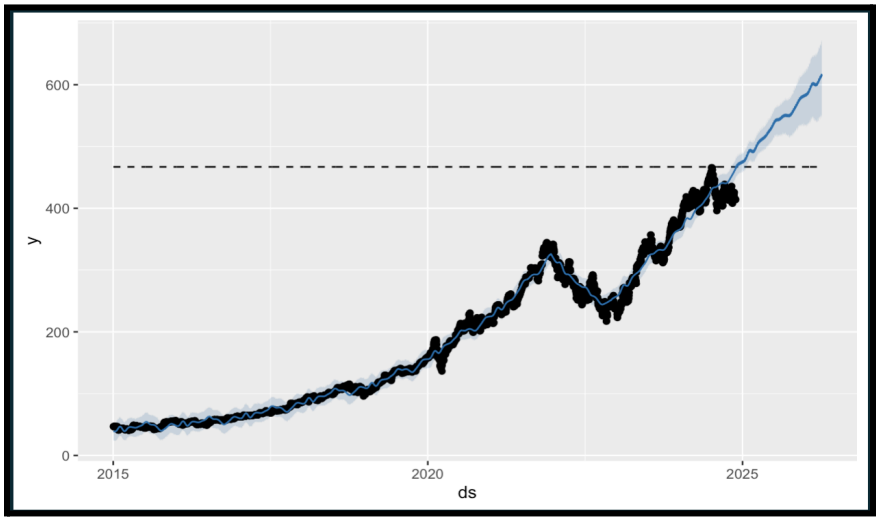
IBM- 500 days out- Price Target of 255.5633



Google- 500 days out- Price Target of 242.89



Microsoft- 500 days out- Price Target of 611.4129



Discussion

Our comparison of the models helps highlight a few things. Firstly, it takes a long time to improve some models, while others (like Prophet) are quick and perform well without any adjustments. Secondly, for the prediction of stock price with as many variables as we had, the more complex models did better. However, after reaching a specific complexity level, they all performed similarly. Thirdly, Prophet has much less limitations compared to the others, such as data needed, preprocessing needed, and the ability to predict further into the future that makes it superior to the other models for our use case even though it does not have the lowest error.

Our analysis of the results of the different forecasting uncovered some great insights into stock price movements and the role of seasonality, which demonstrated the potential of forecasting models like Prophet to inform individual and institutional decision-making. Accurate predictions are a game-changer for investors, enabling data-driven strategies for portfolio optimization and effective risk management. While thinking about individual use, these insights carry broader implications for economic planning and policy-making by revealing underlying market trends that could guide critical decisions at a macroeconomic level.

Prophet's adaptability was a standout feature, as it seamlessly handled diverse financial datasets and captured the cyclical nature of stock prices. This versatility highlights its potential for applications beyond individual stock forecasts, making it a robust tool for financial analysis across industries. The practical applications of this work are equally compelling: companies can incorporate these forecasts into algorithmic trading strategies, while analysts can leverage the

insights to design more effective hedging strategies and implement stop-loss orders to mitigate risk. This project showcases the transformative impact of predictive modeling in the financial sector by bridging technical rigor with real-world utility.

This project successfully highlighted Prophet's potential for stock price forecasting, delivering accurate and interpretable predictions while underscoring the importance of thoughtful data integration. Prophet's flexibility and ability to model seasonality made it a strong choice for this task, particularly when combined with additional variables like quarterly financials. Our analysis reinforced the importance of incorporating external factors to enhance model robustness and its utility for real-world financial decision-making.

Conclusion and Future work

One key takeaway is the need for richer, more diverse datasets to improve feature attribution and capture the significant drivers of stock price movements. Future enhancements should focus on:

- **Integrating macroeconomic indicators**, such as interest rates, GDP growth, and inflation, to provide a broader economic context.
- **Incorporating sentiment analysis** from news and social media to account for market perception and investor sentiment.
- **Expanding the range of external covariates** to improve predictive accuracy and offer deeper insights into the factors influencing stock prices.

By broadening the scope of data and leveraging advanced analytics, we aim to redefine what's possible in financial forecasting.

The real-world significance of this project extends beyond individual investment strategies. Reliable stock price predictions have implications for optimizing portfolios, identifying trading opportunities, and implementing effective risk management. On a larger scale, these forecasts assist institutions and governments in economic planning, policy-making, and mitigating financial crises. Prophet's ability to handle financial market complexities underscores its value as a flexible and efficient solution, contributing to the broader field of financial forecasting.

Contribution

Connor Tomchin led the development by undertaking the coding aspects, ensuring the technical foundation was strong. Emmanuel Epau and Lakshmi Krishnamurthy contributed to the research, slides and the reports, contributing to analysis and documentation. Chamroeun Chhay and Aziz Al Mezraani supported the project by contributing significantly to the preprocessing, coding, and adding to the report where needed. Abhigyan Ghosh focused on data cleaning and preprocessing and laying the groundwork for the analysis, and played a key role in drafting the final report and making visualizations. Together, the team combined their skills to deliver a comprehensive and well-rounded project.

Bibliography

WRDS. (2024). *Wharton Research Data Services*. Retrieved November 18, 2024, from https://wrds.wharton.upenn.edu

Taylor, S. J., & Letham, B. (2017). *Prophet: Forecasting at scale*. Retrieved from https://facebook.github.io/prophet/

Alpha Vantage. (n.d.). *Alpha Vantage API documentation*. Retrieved from https://www.alphavantage.co/

Macrotrends. (n.d.). *IBM Income Statement (Quarterly). Retrieved from* https://www.macrotrends.net/stocks/charts/IBM/ibm/income-statement?freq=Q

Addai, S. (2016). *Financial forecasting using machine learning* (Master's thesis, African Institute for Mathematical Sciences). African Institute for Mathematical Sciences, South Africa. *Retrieved from* (Financial Forecasting Using Machine Learning)

Menculini, L., Marini, A., Proietti, M., Garinei, A., Bozza, A., Moretti, C., & Marconi, M. (2021). Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices. Forecasting, 3(3), 644-662. https://doi.org/10.3390/forecast3030040

Deng, T., Bi, S., & Xiao, J. (2023). Comparative Analysis of Advanced Time Series Forecasting Techniques: Evaluating the Accuracy of ARIMA, Prophet, and Deep Learning Models for Predicting Inflation Rates, Exchange Rates, and Key Financial Indicators. *Advances in Deep Learning Techniques*, *3*(1), 52–98. Retrieved from https://thesciencebrigade.com/adlt/article/view/300

Yusof, U.K., Khalid, M.N.A., Hussain, A., Shamsudin, H. (2021). Financial Time Series Forecasting Using Prophet. In: Saeed, F., Mohammed, F., Al-Nahari, A. (eds) Innovative Systems for Intelligent Health Informatics. IRICT 2020. Lecture Notes on Data Engineering and

Communications Technologies, vol 72. Springer, Cham.
https://doi.org/10.1007/978-3-030-70713-2_45