

Google Stock Price Analysis and Prediction using Machine Learning

Anjana Sebastian   
Faculty of Business *Humber Institute of Technology and Advanced Learning   
Toronto, Ontario*  
 n01510800@humbermail.ca

Rohit Kumar  
Faculty of Business *Humber Institute of Technology and Advanced Learning   
Toronto, Ontario*  
 n01512529@humbermail.ca

Chenguiling Feng  
Faculty of Business *Humber Institute of Technology and Advanced Learning   
Toronto, Ontario*  
 n04169402@humbermail.ca

Shoobham Hans  
Faculty of Business *Humber Institute of Technology and Advanced Learning   
Toronto, Ontario*  
n01480686@humbermail.ca

Jwaleena Jose  
Faculty of Business *Humber Institute of Technology and Advanced Learning   
Toronto, Ontario*  
 n01510696@humbermail.ca

*Abstract*— This paper presents a data product for stock price analysis and prediction using machine learning techniques. The goal of the data product is to analyze historical stock market data of Google and forecast its future prices. The paper outlines the problem statement, key requirements for the data product, and the choice of forecasting models, including Prophet and LSTM. The data collection process involves obtaining reliable and up-to-date historical stock market data of Google. Data preprocessing and cleaning techniques are applied to handle missing values and outliers. Time series analysis is performed to identify trends, seasonality, and patterns in the data. The forecasting models are trained and evaluated using performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). The data product provides visualizations to display historical stock prices, trends, and forecasted values, making it user-friendly for investors and businesses in the stock market domain. The success of the data product is measured based on the accuracy of the forecasting models and positive user feedback about its usability and effectiveness.

Keywords— Stock Price Analysis and Prediction, Machine Learning, Time Series Analysis, Forecasting Models, Prophet, LSTM, Data Preprocessing, Data Cleaning, Performance Metrics, Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Data Visualization, Stock Market Domain.

# Problem Statement

The analytics problem is to develop a data product that analyzes historical stock market data of Google and predicts its future prices using various forecasting models. The goal is to achieve accurate predictions and provide valuable insights to investors and businesses in the stock market domain.

The data product needs to have access to a reliable and up-to-date dataset of Google's historical stock market prices. This data will serve as the foundation for analysis and prediction. The data should be preprocessed and cleaned to handle missing values, outliers, and ensure consistency in the time series and it should be capable of performing time series analysis on the historical stock market data to identify trends, seasonality, and other patterns. Moreover, the solution should include forecasting models that can predict future stock prices based on historical patterns. These models can include methods like Prophet, LSTM. Then the forecasting models should be evaluated using appropriate performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Furthermore, the data product should provide visualizations to display historical stock prices, trends, and forecasted values, making it easier for users to interpret the results.

The proposed data product has the potential to offer significant benefits to both individual investors and businesses operating in the stock market domain. The data product will enable investors and traders to make more informed decisions about buying, selling, or holding Google's stock, leading to potentially higher returns on investments. By providing reliable predictions, the data product can also help investors manage risks and minimize potential losses. The automated forecasting models will save time for investors, eliminating the need for manual analysis and the visualization capabilities will provide valuable market insights to identify patterns and trends. Businesses in the stock market domain can gain a competitive advantage by leveraging the data product's accurate predictions to make strategic decisions. Improved decision-making and risk management can result in increased profitability for individual investors and businesses [1].

The success of the data product will be measured based on the accuracy of the forecasting models. The RMSE and MAPE metrics will be used to evaluate the models' performance. A successful outcome will be achieved if the data product can consistently provide accurate predictions, leading to informed decision-making and increased profitability for users. Additionally, positive feedback from users about the usability and effectiveness of the data product will also indicate its success.

# Summary Of Milestone 1 and 2

In milestone 1 and 2, the data science pipeline for predicting Google's future stock prices was outlined. The first step was data collection, where a dataset from Kaggle containing historical stock market data of Google's stock prices from 2014 to December 2022 was chosen. The dataset includes features such as the opening price, closing price, high, low, average volume, and change percentage.

Next, the data preprocessing and cleaning steps were performed. There are no missing values found in the dataset. Outliers were detected and handled. The outliers in the "Avg. Volume" column were not considered for further analysis as they were not used in forecasting.

Trends, seasonality, and other patterns that could influence the forecasting models were identified during the data exploration phase. Visualizations such as line plots, box plots, and scatter plots were created to better understand the data and observe trends in the closing price, opening price, high and low prices, and average volume over time. It was observed that the closing price of Google's stock showed an increasing trend from 2014 to 2021, with a slight decline between 2021 and 2022. The average volume fluctuated over time, with peak periods in 2014, 2016, and 2020.

The correlation matrix showed strong positive correlations between the open, high, low, and close prices, while the "Change %" had weak positive correlations with close and weak negative correlations with open, high, and low. The "Avg. Volume" had weak negative correlations with all other columns.

Overall, the data collection and preprocessing steps provided a clean and organized dataset ready for model selection and training, paving the way for accurate forecasting of Google's future stock prices.

# Milestone 3:Data Modelling

In the context of predicting Google's future stock prices, data modeling refers to the process of preparing the dataset for training and using it to build forecasting models, such as Long Short-Term Memory (LSTM) and Prophet. Here our goal is to create a data model that captures the relevant patterns, trends, and dependencies within the historical stock market data to make accurate predictions about future stock prices.

We defined a close\_ts time series which includes the columns Month Starting which is already converted to date time format and the closing price to make the time series predictions. Here our main aim is to predict the closing price of Google stock over the years and for future years as well.

Next, we performed Centered and Trailing moving averages with a window size of 3. With a smaller window size, the model can capture short-term fluctuations and respond quickly to changes in the data. It can be useful in identifying short-term trends and turning points.

A computer screen shot of a program

Description automatically generated

A graph showing a line of growth

Description automatically generated with medium confidence

Fig 1

Here we plot two types of moving averages, the Centered Moving Average and the Trailing Moving Average, for Google's closing stock price time series. It also shows an Increasing trend from 2014 to 2021 and again slightly decreased between the year 2021 and 2022.

The plot provides valuable insights into the overall trend and short-term fluctuations in Google's closing stock prices over time. It helps investors and analysts in understanding the general direction of the stock price movement and identifying potential turning points in the stock's value.

We changed the y-axis limits depending on the supplied ylim and the x-axis limits to span the time period from 2014 to 2022. For the training and validation phases, it adds lines and text annotations to visually separate them. Then, to create the layout for two subplots, we defined the function graph Layout (axes, train\_df, valid\_df).

A graph with blue and orange lines

Description automatically generated

Fig 1.1

The layout of the two subplots is then set up by calling the single Graph Layout function twice, passing the training and validation data frames (train\_df and valid\_df) and the two subplots (axes). It divides the data into training and validation sets, with valid\_ts containing the final 36 data points of the close\_ts time series and train\_ts containing the first (len(close\_ts) - 36) data points. The moving average was then computed using a rolling window of size 3 on the training data and stored in the ma\_trailing variable. With a smaller window size, the model can capture short-term fluctuations and respond quickly to changes in the data. It can be useful in identifying short-term trends and turning points. By taking the most recent moving average value from the training data to construct the moving average prediction for the validation time range, it generates a forecast for the validation period. Additionally, the first subplot (axes[0]) depicts the time series, the moving average on the training data, and the forecast. By deducting the moving average prediction from the actual validation data, it determines the forecast errors (residuals), and then plots those residuals on the second subplot (axes[1]). Finally, we used the plt.tight\_layout() and plt.show() functions to display the plots after setting up the layout for the subplots and modifying labels and legends. The code's overall goal is to plot on two subplots the time series data, moving average, and forecast errors for the validation period. Knowing how well the moving average forecast performs in comparison to the actual validation data is useful.

A graph with orange line

Description automatically generated

Fig 1.2

Then, we sought to separate a time series into its trend, seasonality, and residual components. The decomposition is carried out using the seasonal\_decompose function from the statsmodels.tsa.seasonal library. To see how each component contributed to the original time series individually, the components are then plotted in distinct subplots. The seasonal component displays repeating patterns over a specific period, which in this series is 12 months, while the residual component comprises random fluctuations or noise. The trend component indicates the long-term pattern. Understanding the underlying patterns and fluctuations in the initial time series data is aided by this breakdown.

A graph of a growth

Description automatically generated

A graph of a graph

Description automatically generated

Fig 1.3

After that, since the data shows trend and seasonality we used exponential smoothing to forecast time series. It updates the index of the time series close\_ts with dates that are valid as of January 1, 2014. The time series is then divided into a training set (which contains 60% of the data) and a test set (40% of the data). With additive trend, seasonality with seasonal periods 5, the model is fitted to the training data. On the test set, it then offers predictions. Using matplotlib, the real, trained, and predicted values are each shown in a separate color. This image allows for a comparison of the forecast to the actual test data by demonstrating how effectively the Exponential Smoothing model predicts future values based on the training data.

A graph showing a line graph

Description automatically generated

Fig 1.4

We can see that the forecasted values varied widely from the test data values. This may be due to the sudden increase in the close price of Google during the years 2020 and 2021 which is not included in the training set. So, we cannot take this as a perfect prediction model. Also, we calculate the accuracy of the model which gives the following values.

A number on a white background

Description automatically generated

Fig 1.5

Using additive trend and additive seasonal components, we again used exponential smoothing to forecast time series for future periods. With a 12-month seasonal period, it fits the Exponential Smoothing model to the close\_ts time series data. From the final data point in 2022, the code then creates a forecast for the following 12 months. Using matplotlib, the predicted values are presented next to the initial time series data. This graphic offers insights into probable future price movements by demonstrating how well the Exponential Smoothing model forecasts the trend and seasonality in the time series.

A graph with a line

Description automatically generated

Fig 1.6

From the above plot, we can observe a decreasing trend in the closing price in the coming year of 2023.

Additionally, we used a Long Short-Term Memory (LSTM) neural network to achieve time series forecasting. For training the LSTM, input data and corresponding output data sequences are constructed with a predetermined sequence length (seq\_length = 3). The LSTM model with one LSTM layer and one dense layer is built and trained using mean squared error loss on the training set of data. Based on the test data, predictions are made using the trained model. Inverse scaling is then used to change the anticipated values back to the original scale. Finally, a comparison between the forecast and the actual test data is possible thanks to the plotting of the actual and projected values, which shows how well the LSTM performs forecasting.

A screenshot of a computer program

Description automatically generated

A graph showing a line graph

Description automatically generated

Fig 1.7

We can observe that the test data values, and the forecasted values almost follow a similar pattern.

For time series forecasting, we explored creating and training an LSTM (Long Short-Term Memory) model using TensorFlow/Keras. One LSTM layer with 50 units and a ReLU activation function precedes a dense layer with one unit in the LSTM model. The activation function used in the LSTM model is the rectified linear unit (ReLU) activation function. This activation function is commonly used in deep learning models because of its simplicity and effectiveness in dealing with the vanishing gradient problem. In the LSTM model, the ReLU activation function is applied to the output of each LSTM unit to introduce non-linearity in the model and allow it to learn complex patterns in the data. The ReLU function has a simple thresholding behavior where any negative input is mapped to zero and any positive input is passed through unchanged, making it computationally efficient[3]. The Adam optimizer and mean squared error loss function are used in the model's construction. Then, with a batch size of 16, it is trained for 100 epochs using the training data (X\_train and y\_train). Following training, the model is applied to predict values from the test data (X\_test) using the model. Inverse scaling is used to return the predictions to their original scale. The predicted values are then printed for review.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Fig 1.8

The LSTM forecasting model was then given two evaluation metrics. The required sklearn. metrics functions mean\_absolute\_error and mean\_squared\_error are imported. It then compares the test predictions (test\_predictions) with the actual test results (y\_test) to get the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). The outcomes are displayed to evaluate how well the LSTM model predicted the test data.

A close up of text

Description automatically generated

Fig 1.9

Here we observed less accuracy of the model as the errors are very high.

Furthermore, we installed Prophet. The 'Prophet' library was then employed to forecast time series. It creates a DataFrame from the close\_ts data with the column 'ds' for dates and 'y' for time series values. The Prophet model is then set up, trained on the data, and applied to generate forecasts for the following 365 days. Plotting the expected values reveals the anticipated trend and uncertainty ranges. The anticipated close prices are displayed throughout time in the visualization, giving users a glimpse into the time series' possible future behavior.

A graph with a line

Description automatically generated

Fig 1.10

The 'Prophet' library was to be used for time series forecasting. It first prepares the data by turning the close\_ts time series into a DataFrame containing columns for dates and time series values in the form of 'ds' and 'y', respectively. After that, the data is divided into training and test sets. The 'Prophet' model is developed, trained on the training set of data, then applied to forecast the test set. The Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) are then computed as evaluation metrics using the projected values and the actual values from the test set. According to these criteria, the 'Prophet' model's predictions were more accurate than the actual test data.

A black text on a white background

Description automatically generated

Fig 1.11

Therefore, given that the Prophet technique has a low RMSE and MAPE, we may say that it is better.

# Conclusion

This paper successfully developed a data product for stock price analysis and prediction using machine learning techniques. The data product analyzed historical stock market data of Google and applied forecasting models like Prophet and LSTM to predict future stock prices. The data preprocessing and cleaning ensured the dataset's reliability and consistency, while time series analysis identified trends and seasonality patterns. The forecasting models were evaluated using performance metrics like RMSE and MAPE to assess their accuracy.

We think continuous monitoring and updating of the forecasting models will be essential to keep the data product relevant and reliable in changing market conditions.

We think the data product has the potential to offer valuable insights to investors and businesses in the stock market domain. By providing accurate predictions, it can enable investors to make informed decisions, manage risks, and potentially increase returns on investments. For businesses, the data product can serve as a strategic tool to identify market trends and make decisions that lead to increased profitability.

References:

[1] Advantages of Stock Market Prediction | Benefits You Must Know. (2022, June 3). Stock Pathshala. https://www.stockpathshala.com/advantages-of-stock-market-prediction/

[2] Google Stock Data 2014-2022. (2022, December 23). Kaggle. https://www.kaggle.com/datasets/malayvyas/google-stock-data

[3] Poudel, U. (2023, May 23). Time and Series Forecasting with LSTM- Recurrent Neural Networks. Medium.

https://levelup.gitconnected.com/time-and-series-forecasting-6a96bd89dd07