Predicting Long-Term Future Stock Prices Using Extrapolation

Austin Dobrusky, Jack Nolley

May 15th, 2023

Introduction

The objective of this project is to predict long-term future stock prices by using extrapolation with historical stock price data. Stock price prediction is important because it helps investors identify potential growth opportunities, allocate resources efficiently, and manage risks associated with their investment portfolios. In addition, stock price prediction enables investors to assess market risks and uncertainties and develop strategies to maximize returns and mitigate losses.

We created a software program that can predict long-term future stock prices based on polynomial and linear regression techniques. This report will provide an overview of the results of our software, the methodology behind our software, and how our results between linear and polynomial regression compare.

Background Information and Related Work

Some of the existing methods for stock price prediction include technical analysis, fundamental analysis, and machine learning models.

Technical analysis involves analyzing past market data, primarily price action and volume, to identify patterns and trends that may help predict future stock prices. Technical analysis is useful for short-term price predictions and is commonly used in short-term trading rather than long-term investing.

Fundamental analysis, on the other hand, focuses on analyzing the financial statements, earnings, and other factors that affect a company's performance, such as industry trends and macroeconomic factors (inflation data, GDP data, geopolitical news, etc.). This method aims to identify stocks that are undervalued or overvalued to predict their future price movements. Fundamental analysis is more suitable for long-term investing; however, it can still be tremendously useful for predicting short-term price movements.

Machine learning models are a relatively new approach to stock price prediction. Machine learning models use complex algorithms to analyze large amounts of historical data and identify patterns that can help predict future stock prices. Machine learning models can incorporate elements of both technical and fundamental analysis such as price action, volume, market conditions, company performance, and news.

While these methods, when used correctly, can provide investors and traders with high probabilities of predicting stock prices accurately, they also have limitations. The market is an extremely complex environment which makes it difficult to develop accurate models, and unpredictable events such as economic crises and political events can significantly impact stock prices.

Various models have been developed to address the challenges of stock price prediction. Our project uses linear and polynomial regression models. Linear regression models attempt to establish a linear relationship based on the stock price data. Polynomial regression models attempt to fit a polynomial function to the stock price data. Polynomial regression can capture non-linear relationships between the variables, making it more useful for predicting stock prices than linear regression. It is relatively straightforward to interpret the results of these models. However, they are not ideal for time series data and may not capture complex market dynamics.

More advanced models like ARIMA (Autoregressive Integrated Moving Average), LSTM (Long Short-Term Memory), and Prophet also exist. ARIMA is a statistical method that captures trends, seasonality, and noise in time series data. However, ARIMA requires proper parameter tuning to produce accurate predictions. LSTM is a type of Recurrent Neural Network (RNN) with memory cells that can capture long-term dependencies in time series data. LSTM has shown promising results in stock price prediction. However, it requires larger datasets and longer training times than other models. Prophet was created by Facebook as a time series forecasting tool. Prophet is robust to outliers and missing data, automatically detects seasonality and trend changes, and is tunable to specific business requirements. It is relatively easy to use and has shown impressive results in various time series forecasting tasks.

Problem Description

The problem that our project addresses is the prediction of long-term future stock prices using historical data. Accurate prediction of stock prices can aid investors in making informed decisions, identifying potential growth and investment opportunities, and enhancing portfolio management strategies. However, predicting stock prices is a complex task as stock prices are influenced by various factors such as market conditions, company performance, and news. Also, financial time series data is non-linear and non-stationary, and unexpected events like economic crises and political events can impact stock prices. These challenges make it rather difficult to accurately predict long-term future stock prices.

To address this problem, our software requires three inputs for each prediction: the stock's ticker symbol (unique identifier for a publicly-traded company's stock), the number of years in the future to predict the price, and the number of years of historical data to use for the calculation.

Methodology

Our methodology consists of the following steps:

Collecting historical stock price data: We use the yfinance library to download historical stock price data for the selected ticker symbol, based on the user's input for the number of years of historical data to use for the prediction. The data includes daily stock prices along with other financial information.

Data preprocessing: We resample the collected data to monthly intervals and extract the closing prices for each month. We then create an array of time indices and reshape both the time indices and the closing prices into appropriate 2D arrays for model input.

Train-test split: To evaluate the performance of our models, we split the data into training and testing sets using the train_test_split function from sklearn. We use an 80-20 split, meaning 80 percent of the data is used for training the model, and 20% is reserved for testing.

Model selection: We provide the user with two options for stock price prediction: linear regression and polynomial regression. Based on the user's choice, the corresponding model is selected.

Model training and prediction:
a. For linear regression, we create a LinearRegression model using the sklearn library. We train this model on the historical data and use it to predict the stock prices for both the test data and the future time period specified by the user.
b. For polynomial regression, we create a pipeline with the PolynomialFeatures and LinearRegression classes from sklearn. The user is prompted to provide the degree of the polynomial. We train the model on the historical data and use it to predict the stock prices for both the test data and the future time period specified by the user.

Model evaluation: We evaluate the performance of the models by calculating the Mean Squared Error (MSE) and R-squared (R2) scores using the test data. These metrics give us an indication of how well the models fit the data and help us assess their performance.

Displaying results: We present the calculated MSE, R2 scores, and the predicted future stock price to the user. The user can then use this information to make informed decisions about their investments.

This methodology provides a systematic approach to predicting long-term future stock prices using linear and polynomial regression techniques. It allows for user interaction, enabling the user to choose the appropriate method and parameters for their specific use case.

Result Analysis

In this section, we analyze the results of our software when applied to Tesla Inc.'s stock (ticker symbol: TSLA) using both linear and polynomial regression techniques for stock price prediction.

Ticker	Years of Data	Years Predicted	Model Type	MSE	R-Squared
TSLA	20	5	Linear	3730.10	0.48
TSLA	20	5	Polynomial (Degree 3)	1243.59	0.83
TSLA	20	5	Polynomial (Degree 2)	1431.26	0.80

Table 1: Performance of different models on TSLA stock price prediction

We utilized 20 years of historical data to predict the price of TSLA stock 5 years into the future. The model's performance was evaluated using Mean Squared Error (MSE) and R-squared (R2) metrics.

Linear Regression: The linear regression model predicted that the price of TSLA stock will be 284.82 dollars in 5 years. The MSE for the linear regression model on the test data was 3730.10, indicating a considerable difference between the actual and predicted values. The R2 score was 0.48, which means that 48 percent of the variation in the test data could be explained by our model. These values suggest that the linear regression model's prediction may not be entirely reliable for TSLA's stock.

Polynomial Regression: We used the polynomial regression model twice, each time with a different degree of the polynomial.

a. Using a cubic polynomial (degree 3), the polynomial regression model predicted that the price of TSLA stock will be 988.35 dollars in 5 years. The MSE for the polynomial regression model on the test data was 1243.59, and the R2 score was 0.83, meaning that 83 percent of the variation in the test data could be explained by our model. This indicates a better performance than the linear regression model, suggesting that the cubic polynomial regression model may be more suitable for predicting TSLA's stock price.

b. Using a quadratic polynomial (degree 2), the polynomial regression model predicted that the price of TSLA stock will be 690.00 dollars in 5 years. The MSE for the polynomial regression model on the test data was 1431.26, and the R2 score was 0.80, meaning that 80 percent of the variation in the test data could be explained by our model. This performance is slightly lower than the cubic polynomial regression model but still significantly better than the linear regression model.

These results illustrate the potential of polynomial regression over linear regression for predicting long-term future stock prices. However, the choice of polynomial degree is crucial as it impacts the model's performance. In this case, the cubic polynomial regression model showed the best performance, but this may not always be the case.

Initially, our project was designed to utilize linear regression for predicting long-term future stock prices. The rationale was that linear regression, being a simple and interpretable model, would provide a good starting point. However, upon testing our software, we observed that the linear regression model was not providing reliable predictions. The Mean Squared Error (MSE) was quite high, indicating a considerable difference between the actual and predicted values. Also, the R-squared value was less than 0.5, suggesting that our model was unable to explain a significant portion of the variation in the stock price data.

Recognizing these limitations, we decided to enhance our software by introducing polynomial regression. Polynomial regression can capture non-linear relationships in the data, making it more suitable for predicting stock prices, which are known to exhibit non-linear behavior. We also implemented functionality to allow the user to specify the degree of the polynomial, providing additional flexibility. Upon integrating polynomial regression, we observed an improvement in our model's performance, as evidenced by lower MSE values and higher R-squared scores. This adaptation demonstrated the importance of iterative development and flexible methodologies when dealing with complex and unpredictable data such as stock prices.

The extrapolated stock prices should be taken with caution as these models are based on historical data and do not consider other external factors that could significantly influence future stock prices. The models serve as a guide rather than a definitive prediction of future stock prices.

The results also highlight the importance of evaluating model performance using appropriate metrics. The MSE provides an indication of the model's accuracy, while the R2 score helps us understand how well our model explains the variation in the data. By considering both these metrics, we can make more informed decisions about our model's performance.

Conclusion

This project aimed to predict long-term future stock prices using linear and polynomial regression models on historical stock price data. Our original approach of using a linear regression model did not yield satisfactory results, as indicated by a high mean squared error (MSE) and a low R-squared value. This led us to introduce a polynomial regression model to better capture the non-linear nature of stock prices. The improved performance of the polynomial regression model, particularly with a higher degree, underlines the importance of non-linear models in predicting stock prices.

However, it's important to acknowledge that predicting stock prices remains a challenging task due to the complex nature of financial markets. Factors such as global economic trends, company-specific news, and investor sentiment can all significantly influence stock prices. While our models were able to provide reasonable predictions based on past price data, they may not account for these other factors that could impact future prices.

The results of this project underscore the potential of machine learning techniques in financial forecasting. However, they also serve as a reminder of the limitations of these models. While we have seen improvements in our predictive performance with polynomial regression, more sophisticated models that incorporate additional information beyond historical prices could potentially deliver more accurate predictions.

Our future work will focus on incorporating other factors such as company fundamentals, news sentiment, and macroeconomic indicators to improve the accuracy of our predictions. Furthermore, we will explore more sophisticated machine-learning techniques that can better capture the complexities of financial markets.

In conclusion, this project represents a significant step in our ongoing exploration of stock price prediction. We believe the insights gained will be valuable in further improving our models and better understanding the dynamics of financial markets.

Contributions

Austin: Research on project topic, created the linear regression software, introduction section (paper), background information section (paper), problem description (paper), and results section (presentation).

Jack: Research on project topic, created the polynomial regression software, introduction section (presentation), background information section (presentation), problem description (presentation), methodology (presentation and paper), results analysis (presentation and paper), and conclusion (presentation and paper).

References

[1] Ng, Y. (2021). Forecasting Stock Prices Using Prophet. *Medium.* Available at https://towardsdatascience.com/forecasting-stock-prices-using-prophet-652b31fb564e.

[2] Serafeim Loukas, P. (2023a). LSTM Time-Series Forecasting: Predicting Stock Prices Using an LSTM Model. *Medium.* Available at https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-lstm-model-6223e964

[3] Serafeim Loukas, P. (2023b). Time-Series Forecasting: Predicting Stock Prices Using an ARIMA Model. *Medium.* Available at https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-an-arima-model-2e3b3080bd70

[4] Python Software Foundation. Python Language Reference, version 3.7. Available at https://www.python.org

[5] van der Walt, S., Colbert, S.C. & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13, 22-30. Available at https://numpy.org/doc/stable/

[6] Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9, 90-95. Available at https://matplotlib.org/stable/users/index.html

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Available at https://scikit-learn.org/stable/

[8] Szèakely, G. (2019). yfinance. Available at https://pypi.org/project/yfinance/

[9] Niemeyer, G. (2020). python-dateutil. Available at https://dateutil.readthedocs.io/en/stable/