Closing Price Prediction Using Time Series Models

Siqing Guan,Youjia Li, Dhruvi Shah, Yue Wu, Linxuan Zheng

1. Introduction

Stock markets are known for their intensity where prices fluctuate in a matter of seconds. In particular, the closing prices of stocks increase volatility and hold significant influence over the day's global economic narrative. In this project, we aim to build a strategy to predict the adjusted closing price for Google stock on the next day by implementing and evaluating three Time Series Models(ARIMA, ARIMAX, LSTM), and finding out which will be the most accurate in prediction. Our models will contribute to consolidating signals from past data and help investors with decision making. We will also incorporate a stock market performance index, S&P500, in order to better understand the pattern of Google stock in relation to market fluctuation.

2. Data Summary

There are 1258 trading days recorded from Oct 9, 2018 to Oct 9, 2023 and no missing values in the dataset. From the descriptive statistics table, our target Adjusted Close price ranges from 48.81 to 150.71, with the mean price of 94.95 and standard deviation of 30.42. We visualized the overall trend of adjusted closing price, along with Open, high, low price and volume. After calculating their correlations to each other, volume has a weak correlation of -0.3 with the 'Adjusted Close Price,' suggesting it may not significantly impact stock price prediction.

3. Methodology

In this project, we are choosing three time series models to implement: ARIMA, ARIMAX, and LSTM. We are choosing ARIMA because it captures the autocorrelation in time series by considering the past values and differences to achieve stationarity. ARIMAX is the extension from ARIMAX, and it allows the incorporation of exogenous variables, which will help us explore the relationship between Google stock price and that of other companies. LSTM is a type of recurrent neural network that captures complex patterns in both long term and short term, which is perfect for the non-linear stock price analysis. For evaluation metrics, we are using MSE, MAE, and MAPE, which are commonly used metrics for time series models that evaluate the difference between predicted and actual values in terms of the squared, absolute, and percentage. For the model training process, we are using 80-20 percent train test split for all three, and the implementation details will be discussed in the section below.

4. Model Results

4.1.1 ARIMA with Fama-French factors :

In our stock price prediction project, we combined the time series ARIMA model with the multifactor Fama-French three-factor model for a more robust analysis. We integrated collected stock factors with Fama-French factors and applied the least squares method to estimate the expected stock price. Subsequently, we derived the residual alpha and all relevant parameters to evaluate the predictive

strength and investment potential of the stocks. This linear regression model, with an R-squared of 0.635, indicates a moderately effective fit for predicting stock prices, explaining nearly 60% of the variance in the total observations. Key factors like MKT-RF, SMB, and HML are statistically significant, highlighting their influence on stock price movements. The Durbin-Watson statistic close to 2 suggests limited autocorrelation in residuals. However, the high Jarque-Bera value and the corresponding low p-value raise concerns about the normality of residuals, which is crucial for the reliability of predictions in financial models.

Upon examining the residuals, we applied them to the ARIMA time series model for stock price forecasting. Given the processed data was stationary, we set the 'd' parameter to zero. The ARIMA(1,0,1) model was identified as the best fit for stock price forecasting through a meticulous stepwise search aimed at minimizing the AIC. This model emerged as the most suitable from a range of alternatives, ultimately Achieving the lowest AIC= -6166.37. The analysis involved 1000 observations, and the model demonstrated a strong log-likelihood of 3086.18. Key performance metrics for the model included MSE = 0.0004, RMSE= 0.0193, and MAPE= 1.15. These metrics indicate the model's moderate accuracy in capturing the complexities and patterns in the stock price data, underlining its effectiveness in the realm of financial forecasting.

4.1.2 ARIMAX:

For ARIMAX, we incorporate an exogenous variable, daily S&P 500 index, into the ARIMA model. Same as for ARIMA, we require data to be stationary. Both Google's adjusted close price and S&P 500 index are non-stationary without any preprocessing according to the result from the ADF test. With first order differencing both series appear stationary and we use this as a baseline model. ARIMAX(1, 1, 0) on adjusted close price leads to an almost flat prediction line, which potentially suggests that the data is completely white-noise, though almost all real price lies within the confidence interval. This baseline model yields a RMSE of 17.73, a MAE of 14.36, and a MAPE of 12.29% on the test set. Instead of using differencing, we then calculate daily return. ADF test shows that both daily return of Google's adjusted close price and S&P 500 index are stationary. According to ACF and PACF graphs, we decide to use p=1 and q=1 since there's a significant spike on both graphs for period 1. Results on return show a MAPE of 1.14% on the test set; comparing MAE and MAPE with the baseline model in this case is not applicable. We then use the fitted returns and predicted returns to compute fitted and predicted prices for the whole time frame, which gives us a RMSE of 24.28, a MAE of 23.14, and a MAPE of 22.16% on the test set. However, we can tell from the plot that this model captures trend better than the baseline model which simply produces an almost flat prediction. We modify our computation so that the predicted adjusted close price is computed given last day's price in the train set. This modified prediction yields a RMSE of 12.19, a MAE of 10.10, and a MAPE of 10.08% which shows an improvement from the baseline model. However, one problem with ARIMAX regarding application is that we need to input values of the exogenous variable when trying to predict the target value. Thus, we need a reliable source of estimation for the exogenous variable, S&P 500 Index, to get a fairly good prediction on Google's adjusted close price, which is hard to achieve.

4.2 LSTM

We will be comparing the performance of two LSTM models: one utilizing a univariate approach with a single variable, and the other adopting a multivariate strategy with exogenous variables. The objective is to assess how the inclusion of additional features impacts the model's ability to predict closing

prices. The baseline model contains two LSTM layers (one for 128 units and another one with 64 units) and two dense layers with 25 and 1 units. Then compile with the Adam optimizer and use mean squared error as the loss function.

4.2.1 : Univariate Analysis

First, we create sequences of 60 time steps each from the training data as features for the LSTM model. This means that the model uses the past 60 days of adj close prices to predict the next day's price. Using the baseline LSTM model with batch_size = 32, epochs = 20, it has RMSE = 3.44. The reported train loss gets reduced to about 5e-04 but the test loss is higher, indicating possible overfitting problem. Plotting the predicted prices and the actual price together, we can see that the model's predictions capture the overall trend but higher in general. Therefore, we change the batch_size and epochs to improve model's performance since smaller batch sizes lead to improved generalization effect and increased epochs ensure sufficient learning from the training set. The new model of using batch_szie = 8 and epochs = 40 has RMSE = 2.04, MAE = 1.55, MAPE = 1.18, which successfully narrows the gap between predictions and the actual data. We also tried adding dropout after each LSTM layer with dropout_rate = 0.5, another dropout layer before the final dense layer with rate = 0.2, and with batch_size = 1. The performance of the model also improves with RMSE = 2.51. Though using batch_size = 1 improves the generalization, it requires considerably more time.

4.2.2 : Multivariate Analysis

For the multivariate LSTM model, we are adding in the following features to our independent variable: the open price, highest price, lowest price, and the adjusted closing price of S&P 500. The target variable stays the same, which is the adjusted closing price of the next day. We start off with the same baseline model architecture as in the univariate model, which resulted in a RMSE of 2.77, MAE of 2.22, and MAPE of 2.09%. This is showing an improvement compared to the univariate baseline model, and this result is expected because adding multiple variables increases the model complexity and that improves the prediction accuracy. In order to further optimize the model, we first implemented drop out layers with a rate of 0.2 to prevent overfitting, but it didn't lead to substantial improvement. This is possible because the LSTM already has mechanisms that can help control the flow of information and prevent the model from overfitting, and adding drop out can hinder the learning. We then proceed to further optimize the model by tuning different hyperparameters. The best result we get is the model with learning rate of 0.5, epochs of 50, and batch size of 32, with RMSE of 2.51, MAE of 1.99, and MAPE of 1.85%, indicating its improved accuracy in prediction. Using a multivariate model could improve the accuracy but it also introduces noise into the model as some factors that affect the movement of S&P 500 might not be directly related to Google's price.

5. Analysis & Conclusion

Overall, the univariable models tend to perform better than the multivariable models when comparing ARIMA to ARIMAX and univariate LSTM to multivariate LSTM. This might be due to the problem of noise and overfitting in this case, considering the gap between values for SP500 and Google closing price is substantial. The best performing model is ARIMA with the Fama-French three-factor model, which has an RMSE of only 0.0193 as it brings in economic and risk-related dimensions. It is expected to generate a better result than plain time series models. Therefore, we can conclude that the best strategy to predict google stock closing price is to use ARIMA with the Fama-French model, however, if only using our original data, the univariate LSTM will perform the best among all.