Gordon Lu

STAT 1223

Professor Nelson

25 April 2021

STAT 1223: Assignment 4

In the fast-paced, ever evolving world, buzzwords like “cryptocurrency”, “investing”, “money” are attractive. Typically, with 9-5 jobs occupying most days, finding a leisure that brings wealth with minimal effort. Thus, the notion of investing in the stock market. In high-frequency trading firms, many complex algorithms are designed to optimally buy and sell stocks. For an average Joe, time is a sparse resource and spending hours upon hours on end trying to make profit from investing in the stock market while working is stressful. However, blindly investing into big name stocks is also dangerous. Holistically, at the base level, the response variable that I wish to learn more about is “buy or sell stock”, and how this is correlated with a stock’s previous price and other factors. What this paper seeks to do is not to describe trends in the stock market, but rather, given a certain stock, what determines how well a stock performs, and when should someone sell a stock versus buying the stock.

The literature surrounding stock price prediction is quite dense, and typically do not follow a typical regression model. Rather, the general consensus regarding stock price prediction is to formulate a linear time series. In an article written by Salim Lahmiri (2018), Lahmiri’s objective was to combine two popular approaches, being Singular Spectrum Analysis forecasting, which is used heavily in finance and Support Vector Regression (SVR), which is commonly used in time series analysis (Lahmiri, 2018). With both being deeply rooted in

statistics, Lahmiri furthermore performed optimization techniques to find the optimal SVR initial parameters, and applied the hybrid forecasting model to a set of stocks, and compared the results to benchmark models, which included: “FFN trained with WT coefficients (WT-FFNN), polynomial regression (PolyReg), naïve model, and the classical ARMA process” (Lahmiri, p. 445). What Lahmiri found through comparing the four models to the hybrid model was that it “achieved the lowest MAE, MAPE, and RMSE for all time series used in the study” (Lahmiri, p. 450). This article is fascinating in the sense that it provides intuition that employing a typical time series model may yield sub-optimal results in comparison to using a hybrid approach. It is important to note that there may be problems with overfitting and potential points that may pull the regression up. However, in developing an appropriate model to use to forecast stock prices, this article is a good steppingstone, as it not only provides insight on how to compare and benchmark models, but also approaches and decisions to use in deciding a hybrid approach.

On the other hand, in an article jointly written by Seungwoo Jeon, Bonghee Hong, and Victor Chang, the approach is vastly different. In the article, the authors employ a Dynamic Time Warping algorithm to find patterns between two temporal sequences, then use Stepwise Regression to select the determinants most impacted by stock price and lastly generate an artificial neural network (Jeon, Hong, Chang, 2018). What Jeon, Hong, and Chang concluded was for short-term prediction to use a combination of dynamic time warping, stepwise regression and artificial neural network to help gather similar datasets for each stock and predict daily stock prices (Jeon, Hong, Chang, p.185). The article in itself is very dense, it dives deep into theory, and is ambitious enough to dive into the realm of deep learning. This article is useful in the sense that it employs common regression techniques such as Stepwise Regression, and also utilizes an algorithm to find similar datasets. With this, in developing an optimal model so as to yield

feasible VIF values and finding good predictors, using Stepwise Regression, and perhaps even using a neural network to train the model and generate good predictions aid in developing an optimal model for forecasting stocks.

With the articles written so far, there are already predictors in mind, however Bruce James Vanstone, Adrian Gepp and Geoff Harris jointly worked on the idea of considering a commonly overlooked confounding variable in determining stock prices, news. What Vanstone, Gepp and Harris proposed was to build two Neural Network Autoregressive (NNAR) models, one being the base line, and the other being built on the counts of news articles and twitter posts, and to compare the predictive accuracy of the two models (Vanstone, Gepp, Harris, p. 3819). What Vanstone, Gepp and Harris found is that news and twitter sentiment counts do indeed play a role in stock price modelling. While the article is deeply rooted in financial econometric theory and machine learning, there is a lot of useful information. The trend that news and twitter sentiment counts do have a significant relationship to stock pricing, add to the long list of predictors in the model to develop to determine when to buy or sell a given stock.

Among all three articles presented, it is apparent that some degree of time-series was applied. In particular, in Vanstone, Gepp and Harris' article, a NNAR model, based on the marriage of a neural network and autoregressive model from time series analysis was used. With Jeon, Hong and Chang's article, a Dynamic Time Warping algorithm, which involved searching and identifying similar temporal datasets, and is commonly used in time series analysis. With the article written by Lahmiri, a hybrid model of SSA and SVR was used, both of which are deeply rooted in time series analysis. However, where the articles diverge is the models. It is clear that each of the goals of the models presented is to determine the efficiency and accuracy of predicting the prices of stocks. With each model presented the overall consensus is that the

individual models do provide some degree of leverage over typical rudimentary regression models. What can be conclusively said is that in formulating a model that utilizes a binary response, consolidating the predictors such as a variable reflecting how the price of a given stock changes over time, and considering exogenous variables such as the news and twitter sentiment counts are necessary as to avoid introducing confounding variables into the models. All the articles jointly provide evidence that perhaps forming a regression by considering models rooted in time series and utilizing methods such as Stepwise Selection to decide on relevant predictors are necessary. Additionally, a step to further elevate the regression would be to utilize machine learning and generate a neural network to predict whether a stock would be worth buying or selling.

Contrary to centuries past, the world in the 21st century is growing steadily whether it be in terms of technological innovations, or developing new ways to fight viruses like COVID-19, it is beyond a shadow of a doubt that the world is competitive. With so much going on in the world, buzzwords like, “cryptocurrency”, “money”, “wealth” meet the eye well. Typically, with 9-5 jobs occupying most days, finding some leisure time that brings in a decent amount of money is hard to come by. Thus, the idea of “making easy money” by investing in the stock market becomes an attraction option. . In high-frequency trading firms, many complex algorithms are designed to optimally buy and sell stocks. For an average Joe, time is a sparse resource and spending hours upon hours on end trying to make profit from investing in the stock market while working is stressful. However, blindly investing into big name stocks is also dangerous.

Recent studies have shown that predicting stock prices is no stroll in the park to get done. With the number of confounding variables could exist, studies such as one written by Salm

Lahmiri, looks to use a linear time series paired with Support Vector Regression and Singular Spectrum to predict the price of a stock. On the other hand, other approaches such as articles written by Seungwoo Jeon, Bonghee Hong and Victor Change as well as an article written by James Vanstone, Adrian Gepp and Geoff Harris involve stock prediction hinging upon using a neural network paired along with a linear times series. What is evident about each of the respective studies is the overarching goal to predict how well a model can predict stock prices. Holistically, at the base level, the response variable that I wish to learn more about is "buy or sell stock", and how this is correlated with a stock's previous price and other factors. What this paper seeks to do is not to describe trends in the stock market, but rather, given a certain stock, what determines how well a stock performs, and when should someone sell a stock versus buying the stock. One might wonder, "Why embark on trying to do something that seems to be a challenge to those who have tried?" I would refute and note that, I am doing this for the betterment of those, like me, who do not have the luxury of time, and have a genuine interest in investing and finance and learn how to predict time series data and potentially make a couple bucks. One way in which this paper may be appealing is the idea of having a computer decide whether to buy a stock or not, rather than spending hours on end, researching then deciding to buy or sell a stock. What I seek to learn is what variables influence the price of a stock, and given such variables, put together in a regression model, how well do said variables influence buying or selling decisions. If the result of overall regression turns out to be successful in predicting buying or selling, people would be able to boot up their computers, run a program, and watch their pockets grow (at some rate) and enjoy the show. It's a win-win, both the general population would get to enjoy their wealth increase, and I would learn a great deal more about how stocks and time series work more closely.

I seek to use the Yahoo Finance API in order to collect data regarding stocks. In the Yahoo Finance API, each stock records the current listing for its price, as well as its historic data and other logistics such as the PE ratio, beta value and volume. The amount of data collected for a given stock from the Yahoo Finance API will vary depending on how frequent the price is updated as well as the date it was listed on the stock market. For example, a stock like Ford would have more observations than a stock like Apple, simply due to the fact that Ford has been around longer. For a given stock, I intend to use a small slice of the data, as with so many potential observations from when a stock is first listed, selecting data from a single year to the current date will be sufficient. As noted earlier, the observations will differ from stock to stock. I anticipate around 300-400 observations, which will be more than enough to conduct statistical tests.

In utilizing the data from the Yahoo Finance API, it is incredibly difficult to influence the price of a stock, and with a variety of other sources such as the TD Ameritrade API reporting similar numbers, to say that a stock price on Yahoo Finance does not reflect what it actually is would be similar to saying that billions of users are being lied to. With this, I am confident in saying that if Yahoo Finance API did use data that is influenced by external sources such as the government inflating the prices, then Yahoo Finance would not be as popular as it is now. With billions of users investing in the stock markets, and utilizing Yahoo Finance, along with other tools, it is incredibly difficult to say that the data that Yahoo Finance provides is incorrect, and biased. On the issue on whether the dataset is clean, it is beyond a shadow of a doubt that the data that the Yahoo Finance API presents regarding a stock is clean. It consistently updates the price of a stock with up-to-date information, and the only case in which a stock would be missing

values would be if the stock were on the market, but then taken down, or simply isn't on the stock market.

Overall, consolidating whether or not to buy a stock into a single response is the goal. The response, rather than being what the price of a stock could be, I seek to instead use a binary variable that will determine whether buying a stock at a certain day is worth the investment. I will call the response, "buy_stock".

In an attempt to account for the potential influences on "buy_stock", I will consider the following predictor variables:

- stock_price: Quantitative variable determined by running the ARIMA model on the data, and will yield the optimal times to include for the price that will minimize volatility. The times will be based on days, so stock_price_1 would be yesterday's price.

I seek to employ a multiple regression, specifically, I will run a generalized ARMA(p,q) model on the data. Eventually, this model will be upgraded to a logistic regression model based on a modified ARMA(p,q) model. The model will be of the following form:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

It is important to note that each of the X_{t-i}'s represent the price of a given stock at a certain period of time. With this model in mind, I seek to find the optimal periods in time to predict the price of a stock. Furthermore, once the parameters for the ARMA(p,q) model are found, this will be combined with other exogenous variables such as the volume of the stock, and the market

beta. I seek to explore the relationships between the current price of a stock and factors like current news, and the sector the stock is in.

The model is in no way set in stone. The list of improvements to the regression model in mind are endless. One notable change to be implemented is the type of regression, as there are drawbacks in using ARMA(p,q) models, and other candidates such as LSTM regression, SVM regression, and panel data regression are candidate models. One option to explore in the future is to use the ARMA(p,q) model as a benchmark, and form regression models using other choices, as well as incorporate other exogenous variables into each respective model. Perhaps performing something akin to cross-validation, treating different models as hyperparameters, and tuning which model would be the best to use at each period of time would be optimal. What is notable is that once the regressors are selected through the respective models, I plan to use stepwise regression to determine which regressors are the most important. I will use adjusted R^2 and BIC, so to penalize models with more parameters, and to prefer more significant terms in the respective regression models.

I used R to analyze the data. Beyond metrics, I looked at residual plots, ACF plots, time series plots, and diagrams of the roots of the lag polynomials generated from the ARMA(p,q) model. I used a slice of a stock from January 2020 to April 2021. I will measure the predictive accuracy of the regression by testing the labels that the regression model generates at each point in time to the actual results. I will accordingly use a test set, training set, and a validation set, and across different regression models determined by cross-validation I will compare the predictive accuracies of each model, and report which of the models yielded the highest accuracy and lowest error.

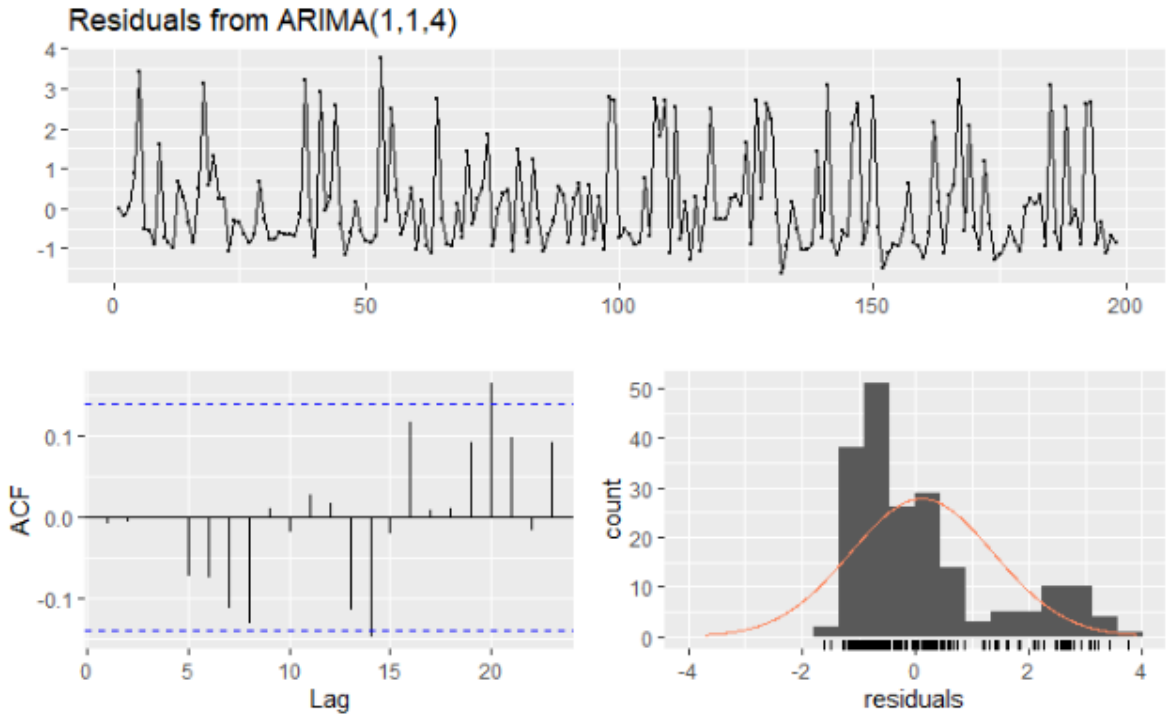
What I seek to learn is what variables influence the price of a stock, and given such variables, put together in a regression model, how well do said variables influence buying or selling decisions. If the result of overall regression turns out to be successful in predicting buying or selling, people would be able to boot up their computers, run a program, and watch their pockets grow (at some rate) and enjoy the show. It's a win-win, both the general population would get to enjoy their wealth increase, and I would learn a great deal more about how stocks and time series work more closely. I seek to use the quantmod package in R in order to collect data regarding stocks. In the quantmod package, it retrieves data from a variety of online sources such as Yahoo Finance, Google Finance and OandA, each stock records the current listing for its price, as well as its historic data and other logistics such as the PE ratio, beta value and volume. The amount of data collected for a given stock from the quantmod package will vary depending on how frequent the price is updated as well as the date it was listed on the stock market.

Overall, consolidating whether or not to buy a stock into a single response is the goal. The response, rather than being what the price of a stock could be, I seek to instead use a binary variable that will determine whether buying a stock at a certain day is worth the investment. I will call the response, "buy_stock". In an attempt to account for the potential influences on "buy_stock", I will consider the following predictor variables:

- stock_price: Quantitative variable determined by running the ARIMA model on the data, and will yield the optimal times to include for the price that will minimize volatility. The times will be based on days, so stock_price_1 would be yesterday's price.

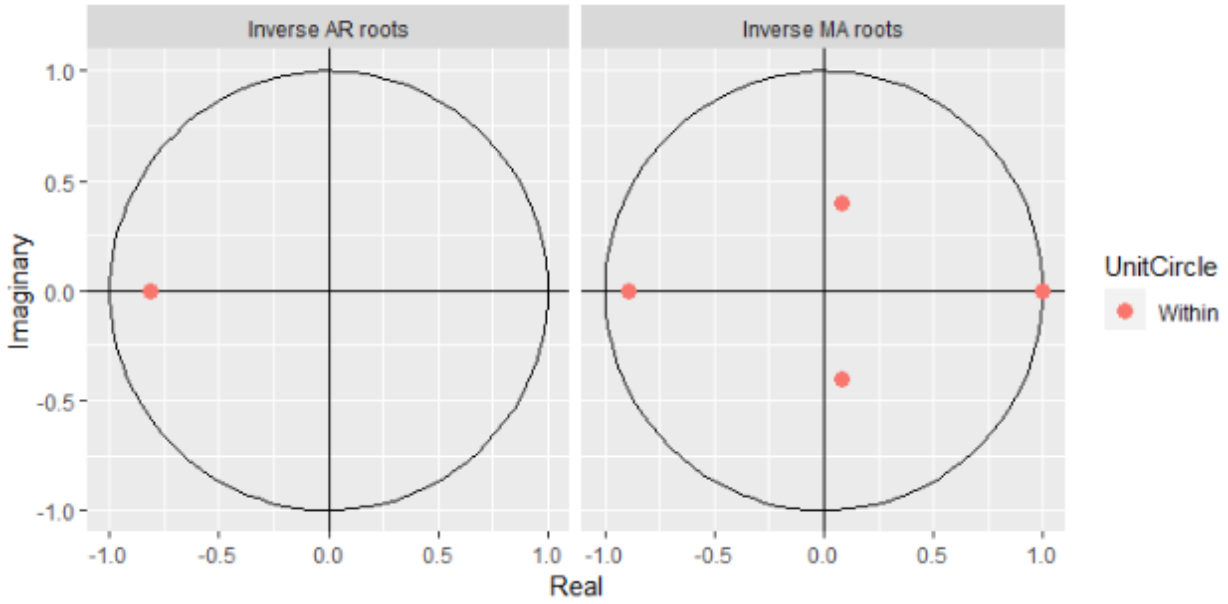
Currently, the big struggle is incorporating exogenous variables while maintaining the nice properties of the ARMA(p,q) model. Using a 60-40 training, test split, I trained a ARMA model on the logarithm of the stock prices for GameStop.

Using a thorough process of determining the optimal values, it was deduced that using an ARIMA(1,1,4) was optimal in prediction for a given stock, GameStop from January 1st, 2020 to the current date, April 25th 2021. Furthermore, a Ljung-Box test was conducted to determine whether regressors were autocorrelated. The Ljung-Box test yielded a p-value of 0.00736, which allows us to conclude that the data is not independently distributed, in other words, they exhibit serial correlation, in other words, the data can be expressed using a time series, as the errors will be heteroscedastic rather than the typical assumption of OLS regression of the errors to be homoscedastic. Another important result is seen from looking at the below residual plots.



Based on the ARIMA errors, it is apparent that the errors represent a white noise series, and the residuals are not significantly different from white noise, thus this adds more to the point that a

time series is feasible to use. This implies that the ARIMA(1,1,4) is a stationary process. Additionally, looking at the roots of the lag polynomial from the ARIMA(1,1,4)



it is apparent that all the roots lie within the unit circle, thus the process is stationary. Upon using an infinite geometric series to estimate the coefficients of the lag polynomial, the ratio will converge, and yield a number in the unit circle from the MA and AR parts of the ARIMA model. Thus, the process is stationary and can be approximated using a linear time series. One significant result was that in determining the accuracy of the ARIMA(1,1,4) model, it yielded the following:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.794616	1.321718	0.8691971	19.33185	24.95099	6.78169	0.9498892

which implies that the RMSE yielded $1.32, which is not bad, given how much the price of a stock can vary. The results of running the ARIMA(1,1,4) model imply that a spare amount of autoregressive terms and a rather dense amount of lagged forecast errors aid itself well in predicting GameStop stocks. The most significant test had to be the Ljung-Box test to determine whether it was feasible to build a linear time series out of the given data.

The results of this model are skewed and certainly prone to lots of error. For one thing, it does not consider exogenous variables. Which was really something that I had planned to implement, but due to a struggle with how to combine ARIMA and linear regression, I struggled a lot with making such a model in R. The model implies that looking relatively recent in the past is better than peeking too far in the past. This aids itself well with the theory of time series, after some point, peeking around 100 years in the past for example won't be as useful as looking a week or two in the past. One limiting factor is that I had initially sought to formulate a binary regression to "buy or sell" stocks, but one difficulty is formulating a rule to classify to buy or sell. Such a task could be done with a recurrent neural network. Additionally, the time was rather sparse for this project, and with more time, more variables could be discovered and the encoding in R could probably be figured out. Building upon this paper, further research should be focused on considering exogenous factors as well as considering more machine learning models such as recurrent neural networks, and even implementing more time series models such as GARCH models. The sparsity of available resources on how to rationally implement exogenous variables was a huge drawback and explains the rather high RMSE. Furthermore, rationally coming up with a rule for whether to buy or sell stocks at a given time was tough. It would seem arbitrary, and perhaps consulting some financial experts would aid itself well to improving this rather simplistic model.

References

Jeon, S., Hong, B., & Chang, V. (2018). Pattern graph tracking-based stock price prediction using big data. *Future Generation Computer Systems*, 80, 171–187. https://doi.org/10.1016/j.future.2017.02.010

Lahmiri, S. (2018). Minute-ahead stock price forecasting based on singular spectrum analysis and support vector regression. *Applied Mathematics and Computation*, 320, 444–451. https://doi.org/10.1016/j.amc.2017.09.049

Vanstone, B., Gepp, A., & Harris, G. (2019). Do news and sentiment play a role in stock price prediction*? Applied Intelligence (Dordrecht, Netherlands)*, 49(11), 3815–3820. https://doi.org/10.1007/s10489-019-01458-9