Masters of Science in Mathematics Project: Stock Market Web Application

Jordan Wheeler

April 18, 2019

1 Overview

As partial fulfillment of the Master of Science degree in Mathematics (with a Statistics concentration) from the University of Nebraska - Omaha, a project with an academic and external advisor must be conducted. The project ought to use knowledge and skills learnt throughout the course of the program (in my case, statistics).

For this project, Jeff Anderson, the Director of Business Solution Strategies at TDAmeritrade, was my external advisor. For the project, we decided that I would create a time series forecasting web application for stock closing prices. Since I would be working with time series data, I decided that I would work with Dr. Andrew Swift as my academic advisor.

1.1 Project Description

As defined through the project proposal form, this project was aimed to accomplish the following:

> Create a web application to allow the user to choose various forecasting methods on daily stock market closing prices and volume. The user will also be able to choose 15 minutes and 60 minutes real-time intervals rather than daily.
>
> The web application will also allow the user to compare the relationship between multiple stocks. The application will use multiple visualizations to allow the user to conduct exploratory data analysis. Also multiple forecasting methods will be offered to the user so that the user can see which method best fits.
>
> The application will be built using R (RShiny) for the server and user-interface side, and a Yahoo API for streamlining stock data.

As the application was being built out, the requirements shifted and the application included the following:

- Traditional stock graphs (i.e. time series line graph and candlestick graph)
- Generalized and usable for any stock on the New York Stock Exchange (NYSE)
- Automated ARMA+GARCH modeling of any stock on the NYSE for any date range as long as the beginning date was before 2008
- Popup text box to share with the user the ARMA+GARCH model that was chosen
- Manual capabilities of an user-defined ARMA+GARCH model
- Easy interpretability of forecasts
- Error bars arround the projected forecast of prices
- Downloadable table (i.e. downloadable .csv file) of actual stock prices and forecasts

1.2 Motivation

The project was designed with my academic background in thought. Since majority of my courses during my Masters of Science program dealt with statistical modeling (e.g. time series), data visualization, web application, and data science, we decided that creating a web application would demostrate the knowledge and skills I had gaine.

Aside from my academic background, I have been interested in integrating statistics and mathematical models within a application to allow the user to gain insight from my knowledge of statistics, modeling, and visualizations. There is also an increasing desire from society to integrate mathematical modeling in technology and a desire to automate and streamline the processes for users of the technology.

Lastly, since the data we are working with is financial and stock data from the NYSE it is publicly available to everyone. This means that this project and application can we shared for future purposes.

2 Time Series

Time series is a special case of statistical modeling that models and understands data which is collected consecutively over time. Whether it is financial or stock prices, temperature or weather related values, sales of a business, biological and chemical processes, or even daily activity levels, we observe time series data everyday throughout life.

Generally speaking, there are two reasons we are concerned with time series modeling. The first is to understand sequential data. This means that we are interested in identifying trends, seasons, and cycles within sequential data. The second reason is to forecast future values given the history. This allows us to make current decisions based on what is projected for the future. For example, we have seen this with carbon emissions. Given the historical and current data of climate and weather, we can forecast future values and we see that the future is quite concerning. We can use this information from the forecasts to make present decisions on lowering carbon emissions to try and reduce future harm of the environment.

There are several indicators which would tell someone that a time series model is appropriate given certain data. The most obvious indicators are that there is a time element which is apart of the data and that consecutive points or values are correlated with eachother.

2.1 Stock and Financial Data

For this specific project we are working with stock data. The best way to model stock data is through time series modeling. Even though it is quite obvious that stock data is a time series, we can see that the obious indicator from the previous paragraph apply. Stock data is reported on a timely basis, whether it is daily or intraday. It is also obvious that current stock's price does not start from 0 and then go up, it starts from the previous value. This would indicate that the current price is correlated with the previous price values.

However, it was decided in the 1980s by Dr. Robert Engle and Dr. Tim Bollerslev (Bollerslev 1986) that financial data should not be model using current time series methods. At the time current methods only modeled the condition mean of the data being model. This means that it did not take into account the previous variances of the model. However, Dr. Engle and Dr. Bollerslev noticed that financial data involved volatility clustering. Volatility clustering can be thought of the phenomenon where large flucuations (or high volatility) of a stock are typically followed by more large flucuations and small flucuations (or low volatility) of a stock are typically followed by small flucuations (see Figure 1). In other words, if a stock is volatile today, then it is likely to be volatile tomorrow, and it will eventually return to a calm and stable state. And if a stock is calm and stable today, then it is likely to be calm and stable tomorrow, and it will eventually return to a volatile state.

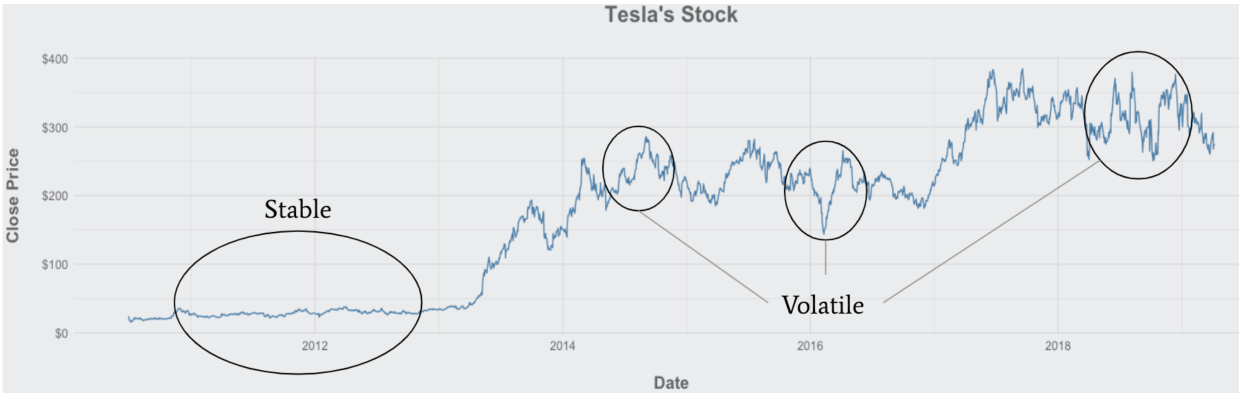


Figure 1: Shown here is Tesla (TSLA)'s stock from 2010 to 2018. Circle are areas which volatility clustering can be seen. Stable states are grouped together and volatile (big changes in prices) states are grouped together

This volatility clustering concept sparked the motivation to come up with a new time series method where the mechanism models the variance given the previous variances. This idea was drawn out by Dr. Engle and he created a new type of time series model known as the Autoregressive Condition Heteroskedasticity (ARCH), which models the conditional variance of a stock (Engle 1982). A few years later, Dr. Bollerslev developed a new model based off of the ARCH model. This model is known as the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model and it is used when the data (i.e. it's mean) follows an Autoregressive Moving Average (ARMA) model.

Since the development of the ARCH/GARCH models, financial data has been model using an ARMA+GARCH model. These models will be explained in the following subsections. It is important to note that the idea of the ARMA+GARCH model is that the ARMA part models the conditional mean of the stock (i.e. its value) and the GARCH part models the conditional variance of the stock.

2.2 ARMA Modeling

As it was stated previously, the ARMA model forecasts the conditional mean of an entity. We define the ARMA model as (Cryer and Chan 2011):

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \tag{1}$$

Breaking down the equation above, we have that Y_t is the value of the entity (or stock) at time t. Also, we have that e_t is the value of the error term at time t. Lastly, we have that ϕ's and θ's are constants of the value of the entity and the error of the entity. We then state that the ARMA model has two parameters p and q, where p is the number of lag conditional means and q is the number of error terms. We then define an ARMA model by $ARMA(p, q)$

2.3 GARCH Modeling

As it was stated previously, the GARCH model forecasts the conditional variance of an entity. We define the GARCH model as (Cryer and Chan 2011):

$$\sigma^2_{t|t-1} = \omega + \beta_1 \sigma^2_{t-1|t-2} + ... + \beta_p \sigma^2_{t-p|t-p-1} + \alpha_1 r^2_{t-1} + ... + \alpha_q r^2_{t-q} \tag{2}$$

In a similar fashion to the ARMA section, we will break down the above equation. First, we have that $\sigma^2_{t|t-1}$ is the conditional variance at time t given the information up until time $t-1$. Also, we have that r^2_{t-q} is the squared error term at time $t-q$ and $\sigma^2_{t-p|t-p-1}$ is lagged variance at time $t-p$ given the information up until time $t-p-1$. Lastly, we have that the ω, β's, and $, \alpha$'s are constants. We then state that the GARCH model has two parameters p and q, where p is the number of lag conditional variances and q is the number squared error terms. We then define a GARCH by $GARCH(p, q)$.

2.4 ARMA + GARCH Modeling

The ARMA+GARCH model uses the models from the above section meaning an ARMA+GARCH model is defined by $ARMA(p, q) + GARCH(p', q')$, where p and q are the parameters of the ARMA model and p' and q' are the parameters of the GARCH model. The ARMA+GARCH models the log returns of a stock or financial entity. This is an important distinction to make - we are not modeling the close price of a stock, rather, we are modeling the log of the return of a stock. The log return of a stock is defined by:

$$Ret_t = log\left(\frac{Close_t}{Close_{t-1}}\right)$$

Where, Ret_t is the log return at time t; $Close_t$ and $Close_{t-1}$ are the closing prices at time t and time $t-1$.

This means when we make a forecast using an ARMA+GARCH model, it is the log return not the close price. Somehow we need to convert the log return into a close price. We will show this by example. Suppose we train an ARMA + GARCH model on the log returns from time 1 to time t (i.e. we are modeling the data Ret_1, Ret_2,..., Ret_t). Our first forecast will be $\hat{Ret}_{t+1}$. We will need to convert $\hat{Ret}_{t+1}$ into a forecasted close price, $\hat{Close}_{t+1}$. Using the definiton of a log return, we will get the forecasted closed price at time $t+1$ through the following steps:

$$\begin{aligned}
\hat{Ret}_{t+1} \quad &= log\left(\frac{\hat{Close}_{t+1}}{Close_t}\right) \\
\text{iff. } e^{\hat{Ret}_{t+1}} \quad &= \left(\frac{\hat{Close}_{t+1}}{Close_t}\right) \\
\text{iff. } \hat{Close}_{t+1} \quad &= e^{\hat{Ret}_{t+1}} * Close_t
\end{aligned}$$

Therefore, the forecasted close price at time $t+1$ is the close price at time t multipled by the (natural) exponential of the forecasted log return at time $t+1$. For further information about the theory behind ARMA and GARCH modeling, refer to *Time Series Analysis with Applications in R* by Jonathan D. Cryer and Kung-Sik Chang.

3 Application

With the advancement in technology within the past 20 years, the use (and dare I say neccessity) of applications has increased. Applications allow technology and content experts to put their skills and knowledge in the hands of users who may not possess those certain skills.

Since forecasting heavily relies on a statistical and mathematical background, I can use my knowledge of these concepts to create models that will forecast the given data. This will streamline the statistical process for the user, which means the user will not need the knowledge of time series forecasting to create forecasts. However, the complication comes when I try to bridge the gap between my knowledge and a user who does not have the knowledge. The development of an application will act as this bridge, which will allow the user to easily interact with my statistical models and easily interpret the results.

3.1 R Shiny

Throughout my academic pursuit at the University of Nebraska - Omaha, I have learned to use a statistical programming language called R. The language R allows users to easily explore, clean, visualize, and model data inputs. Built within the R language is a package called *Shiny*. This package allows R users to create interactive web applications which can be shared with other users. It allows the R programmer to generalize its R code by accepting input values that interact with the outputs.

Majority of the application is built using R Shiny. However, I did use custom HTML (HyperText Markup Language), CSS (Cascading Style Sheets), and JS (JavaScript) to make style edits to my application. When building an application using R Shiny, you have two important parts, a User Interface (UI) side and a Server side. The UI side is used for the display, it is what the user of the application sees (visual component of the application). The Server side is used to take inputs from the interface (i.e. user), perform some sort of calculation, and provide an output, which is then displayed on the interface.

3.2 Automated Modeling

Since the main purpose of the application is to build a bridge between statistical knowledge and a user with no statistical background, it is important to automate the modeling process. This automation has to be generalized so that it can be done for any stock and any date range given.

There are packages in R that automate the modeling process for other statistical models, however, there is no package that automates the modeling process for a ARMA+GARCH model. After some digging on the internet, I found a function written by Ivan Popivanov, a Microsoft Software Engineer, that automates the ARMA+GARCH modeling (Popivanov 2013). Before blindly trusting Popivanov's automation process, I looked at the code to understand what was going on and determined that the process is sound.

It is important to understand the basics of the automation process by Popivanov. The basic idea is we will do a grid search method over the parameters for the ARMA and GARCH models. Focusing on the ARMA model, we have two parameters p and q. For the grid we will search over the values $(0, 1, 2, 3, 4, 5)$. This means we will have a 6 by 6 grid of potential parameters for the ARMA model. For the GARCH side, we will hold the parameters p' and q' at 1, therefore we do not perform a grid search for the GARCH model. This means we will fight a total of 36 models everytime we automate the modeling process. We will then take those 36 models and compare their AICs to determine which model is the best for the given data.

3.3 Prediction

This section is added due to an issue that occured while programming the application. With the ARMA+GARCH models, the standard prediction function outputs an error when trying to forecast more than 1 step out. To fix this issue, Dr. Swift provided me with his own custom prediction function that will not have an error when forecasting more than 1 step out for a ARMA+GARCH model. The prediction function not only provides the forecast, but it also provides the standard deviation of the forecast. This standard deviation can be used to show a confidence interval of the forecast, which is used to create error bars on the forecast in the output plots.

3.4 Application Guide

Getting familiar with the application and documenting its abilities creates a guide that is crucial for the user to successfully use the application. We will look at snapshots of different features within the application and example what it is and how it works. You will notice that the design of each individual page is similar to one another, this was done on purpose to make the application more user friendly and easy to use. The application itself has four total pages which consist of two different style of pages: Overview Pages and Forecast Pages. Each of these two styles have two separate pages, one for Daily returns and the other for

Intraday returns. In this guide section, we will focus on the Daily return pages for these two styles, however we will make note of how the Intraday pages differ (even though they are quite similar).

3.4.1 Overview Pages

The first thing we need to look at is the initial page when you open the application (Figure 2). When you first open the application you are taken to a page which is called the Daily Overview. This page is just a visualization page of the stock. When you open the application, it will automatically select the Apple Inc. (AAPL) stock. The date range will default to one month ago until current date. The graph which is shown will automatically default to the candlestick chart.

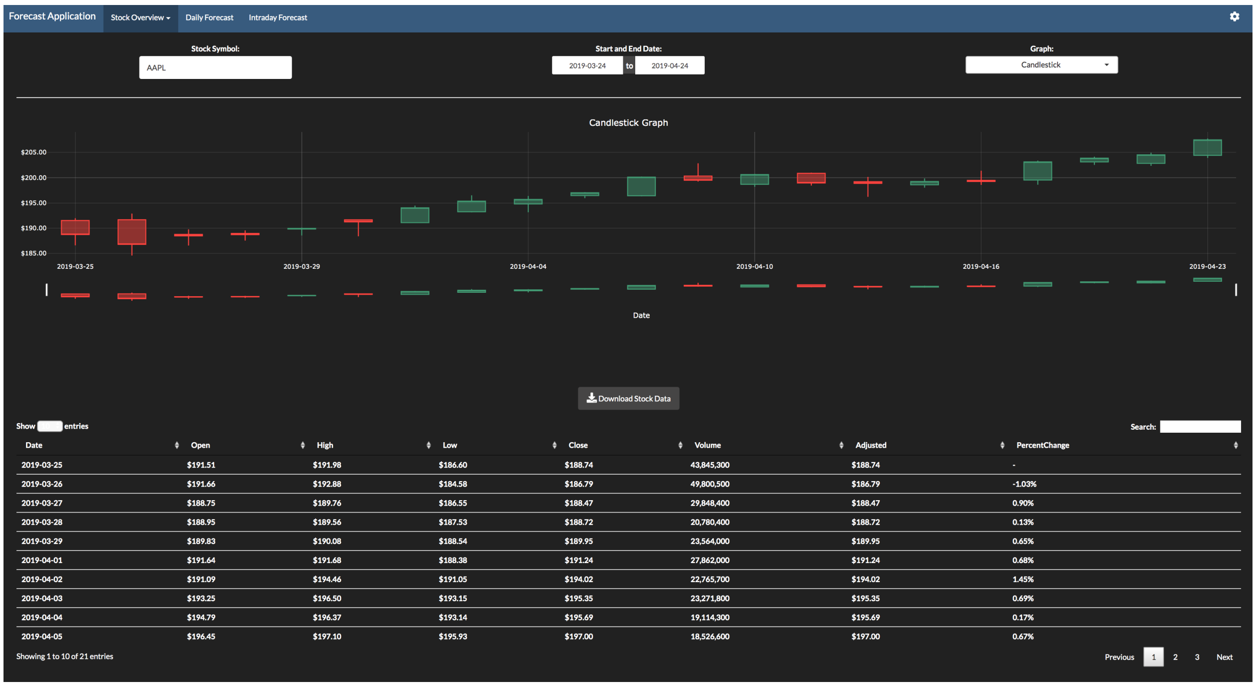


Figure 2: Shown here is Daily Overview. This is what the application will look like when launched.

You will notice that this Daily Overview page has three main components: Input Panel, Graph Section, and Table Section. The Input Panel component will control what the outputs of the Graph and Table Sections (the concept of the user interface and server side working together).

The Overview Input Panel (Figure 3) contains information about the stock. There are three main inputs: Stock Symbol, Start and End Date, and Graph. The Stock Symbol is where the user inputs any stock symbol which is on the NYSE. This input is not case sensitive and will automatically default to “AAPL”. The Start and End Date input is where the user puts the date range to view the history of a stock. The start date can go as far back as the user wants, however, most date is only goes back to the 2007s. If the user inputs a start date which is before the first date which has a value, then the output will automatically use the first date which has a value as the start. The end date can also go back however far the user wants (as long as the date is after the start date). The start date automatically defaults to a month from current date and the end date automatically defaults to current date. The Graph input is a selection input where the user can choose between a candlestick chart or time series line plot. The application automatically defaults to the candlestick chart.

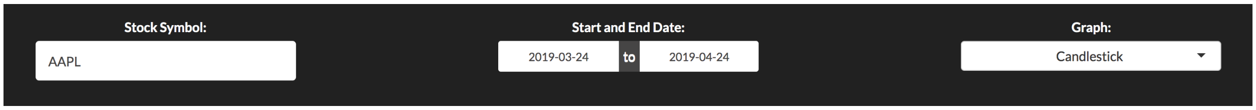


Figure 3: Shown here is the Overview Input Panel. The user will specify inputs for the application.

The Overview Graph Section (Figure 4) is where the graph of the stock is displayed. Since the Graph input automatically defaults to a candlestick chart, this is what is shown. It is important to note that this graph section is JavaScript, which mean that it is interactive. The user is able to hover over the different bars to learn information about that specific date. The user is also able to zoom into different sections of the plot. To reset any zooming, the user must click the house icon in the upper right of the chart.

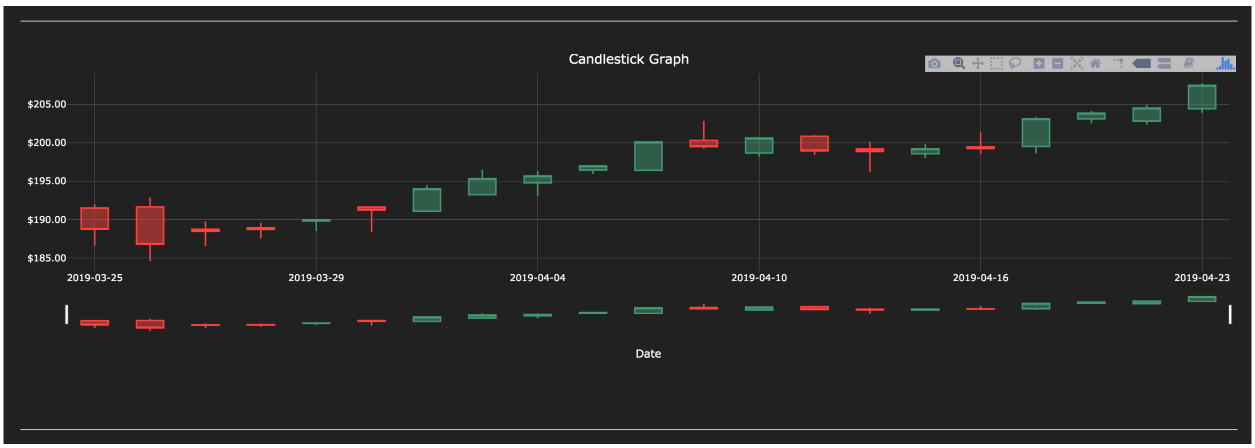
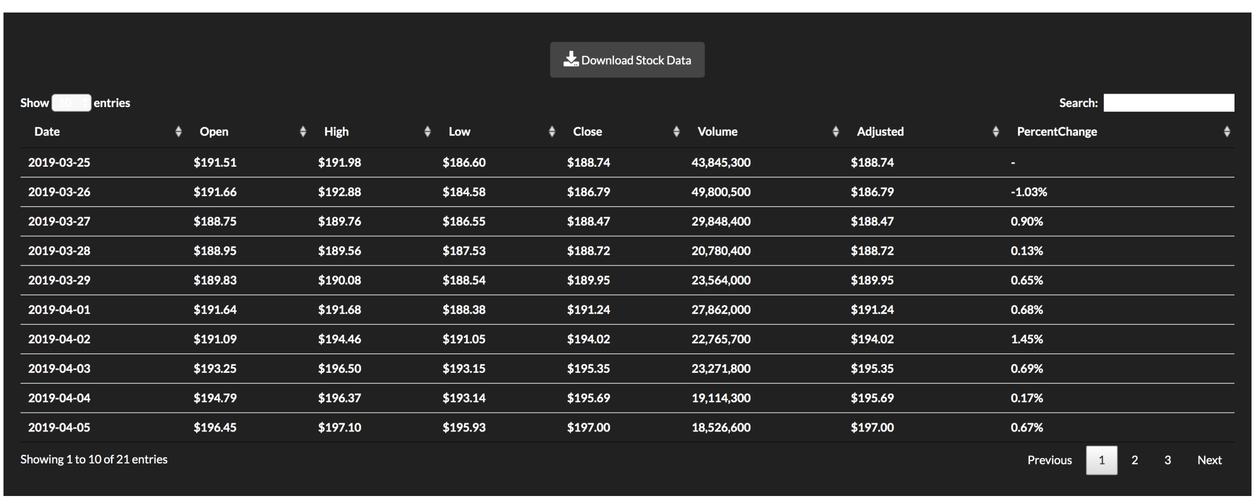


Figure 4: Shown here is the Overview Graph Section, which is interactive.

The Overview Table Section (Figure 5) is where a table of the data is displated. This table is for users to view the actual historical data easily. You will notice there is a download button. This will download the current stocks historical data given the date range (i.e. what is shown in the table). This table is also interactive, which means the user can do searches using the search bar, and the user can sort the table by different columns, which is defaulted to ascending order of date.


Download Stock Data

Show 10 entries

Search:

Date	Open	High	Low	Close	Volume	Adjusted	PercentChange
2019-03-25	$191.51	$191.98	$186.60	$188.74	43,845,300	$188.74	-
2019-03-26	$191.66	$192.88	$184.58	$186.79	49,800,500	$186.79	-1.03%
2019-03-27	$188.75	$189.76	$186.55	$188.47	29,848,400	$188.47	0.90%
2019-03-28	$188.95	$189.56	$187.53	$188.72	20,780,400	$188.72	0.13%
2019-03-29	$189.83	$190.08	$188.54	$189.95	23,564,000	$189.95	0.65%
2019-04-01	$191.64	$191.68	$188.38	$191.24	27,862,000	$191.24	0.68%
2019-04-02	$191.09	$194.46	$191.05	$194.02	22,765,700	$194.02	1.45%
2019-04-03	$193.25	$196.50	$193.15	$195.35	23,271,800	$195.35	0.69%
2019-04-04	$194.79	$196.37	$193.14	$195.69	19,114,300	$195.69	0.17%
2019-04-05	$196.45	$197.10	$195.93	$197.00	18,526,600	$197.00	0.67%

Showing 1 to 10 of 21 entries

Previous 1 2 3 Next

Figure 5: Shown here is the Overview Table Section, which is interactive.

Underneath the Stock Overview tab, you will have an option between Daily and Intraday Overview. The application automatically defaults to the Daily Overview. The Intraday Overview is similar to the Daily Overview, however, there is an additional input for step. This input allows the user to change between different intraday steps (i.e. 1 min., 5 min., 15 min., 30 min., and 60 min.). The date range will automatically default to just current date. The rest of the Intraday Overview similar to the Daily Overview, thus we will not go over it in detail.

3.4.2 Forecast Pages

The Forecast Pages (Daily Forecast and Intraday Forecast tabs) are the other important pages to explain. Since the Forecast Pages are similar, we will just discuss the Daily Forecast page. When you click the Daily Forecast tab, a page will be pulled up with just an input panel and some empty sections below the input panel (Figure 6). The user will need to interact with the input panel to get any outputs shown.

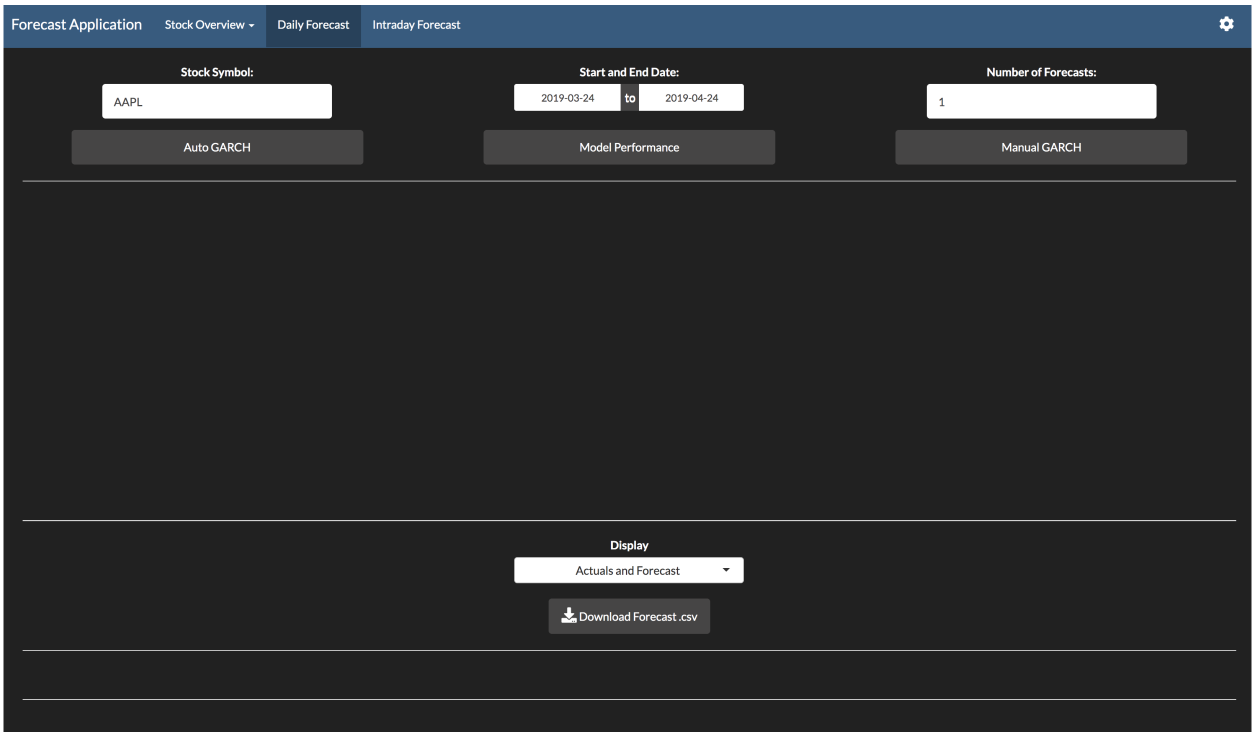


Figure 6: Shown here is the Daily Forecast Page when first clicked.

Similar to the Overview Pages, the Forecast Pages have three main componenets to its layout: Input Panel for forecasting information, a Graph Section to display the forecasts once a model is chosen, and a Table Section to display the date range historical information as well as the forecast information.

The Forecast Input Panel (Figure 7) have different inputs, some of which are similar to the Overview Pages inputs, and action buttons. The inputs are Stock Symbol, Start and End Date, and Number of Forecasts. The action buttons are Auto GARCH, Model Performance, and Manual GARCH. Looking at the different inputs, the first thing the user needs to do is select a stock which they would like to forecast on. The appliaction once again defaults to Apple Inc. (AAPL). The user is able to enter any stock from the NYSE into the Stock Symbol input. The next thing the user will need to do is choose a date range which will be the training data for the ARMA+GARCH model. The Start and End Date inputs automatically defaults to a month previous of current until current date. It is important for the user to know that for more accurate forecasts, multiple years (around 4 to 5) of data is necessary for the model to be trained on. I would suggest that the user goes back until 2010 when making a forecast. The last input that the user has to specify is Number of Forecasts. This tells the application how many steps out (i.e. number of trading days for daily forecasting). The application automatically defaults to 1 forecast, which means it will forecast tomorrows close (or the next trading date after the specified end date). The user can choose as many forecasts as they would like, however, the more steps the forecast goes out, the less accurate they will be.

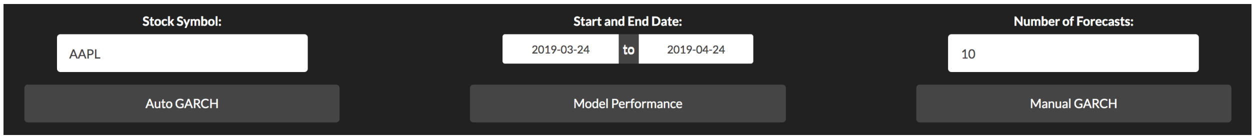


Figure 7: Shown here is the Forecast Input Panel. Users will interact with input panel to get desired forecasts and outputs

The Forecast Input Panel also consists of different action buttons. The first action button is Auto GARCH. Upon clicking this button, the application will do a grid search for the best ARMA+GARCH model (explained in the previous section). The Auto GARCH will model the specified stock symbol and use the date range as the training data and provide the specified number of forecasts. This process can take up to a minute or so but a spinning wheel and progress bar is provided to let the user know that the model is still training. Once, a model is chosen, the user will be able to click the Model Performance action button and this will open a popup with an R output of the model (Figure 8). This will tell the user which ARMA and GARCH parameters were chosen and the significance of each variable.

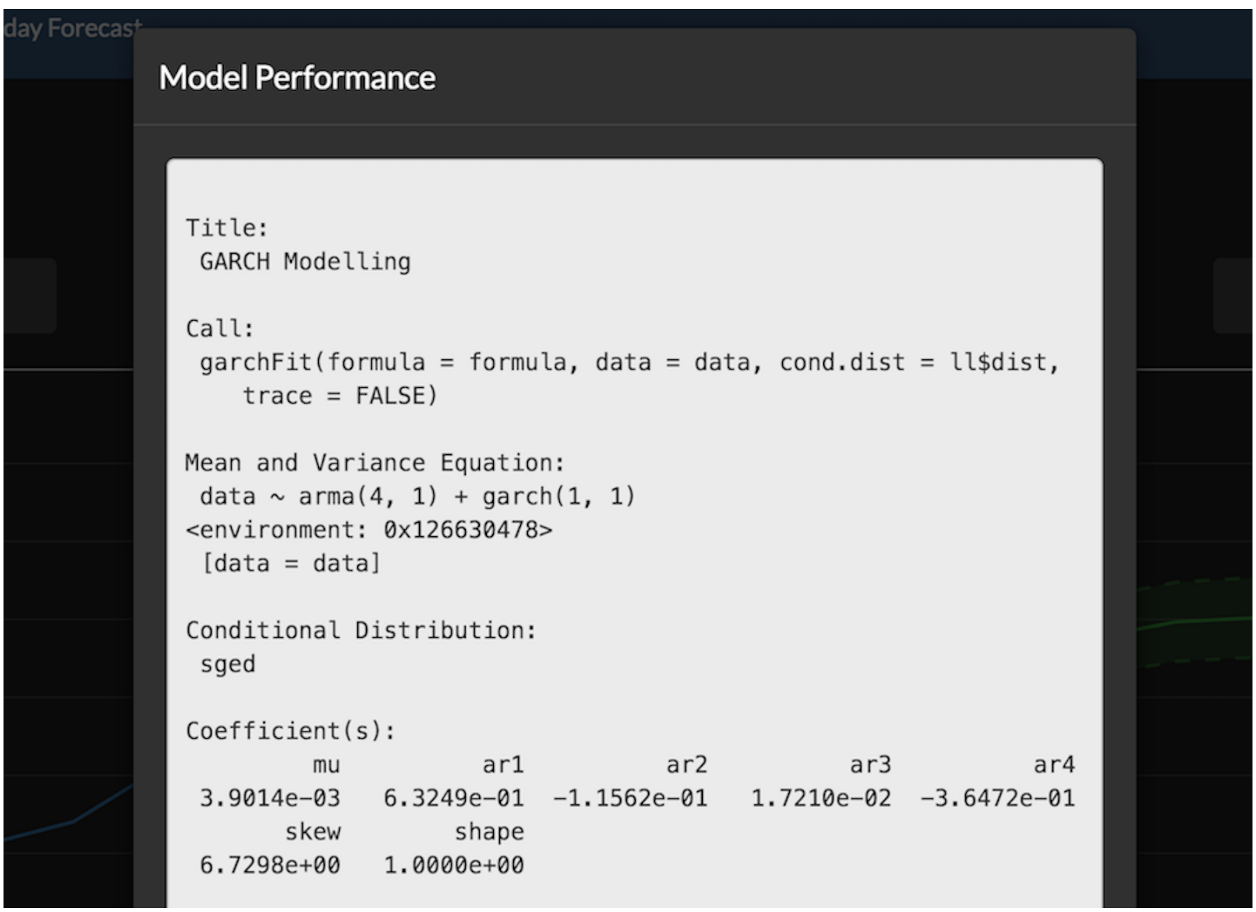


Figure 8: Shown here is the Forecast Model Output. Shows the model which was chosen as the best.

The last action button is the Manual GARCH. This is for more advanced useres who know the theory behind ARMA and GARCH modeling and want to select parameters outside the grid search. Upon clicking this button, a popup will appear (Figure 9), which will have four addition inputs: AR Parameter, MA Parameter, GARCH Parameter, and ARCH Parameter (these were explained in the previous section). Once the parameters are chosen, then the user can click the Forecast button and the model will use those parameters and display the associated output. If the user specifies parameters which are negative, the application will display an error message and tell the users to choose a non negative value for the parameters.

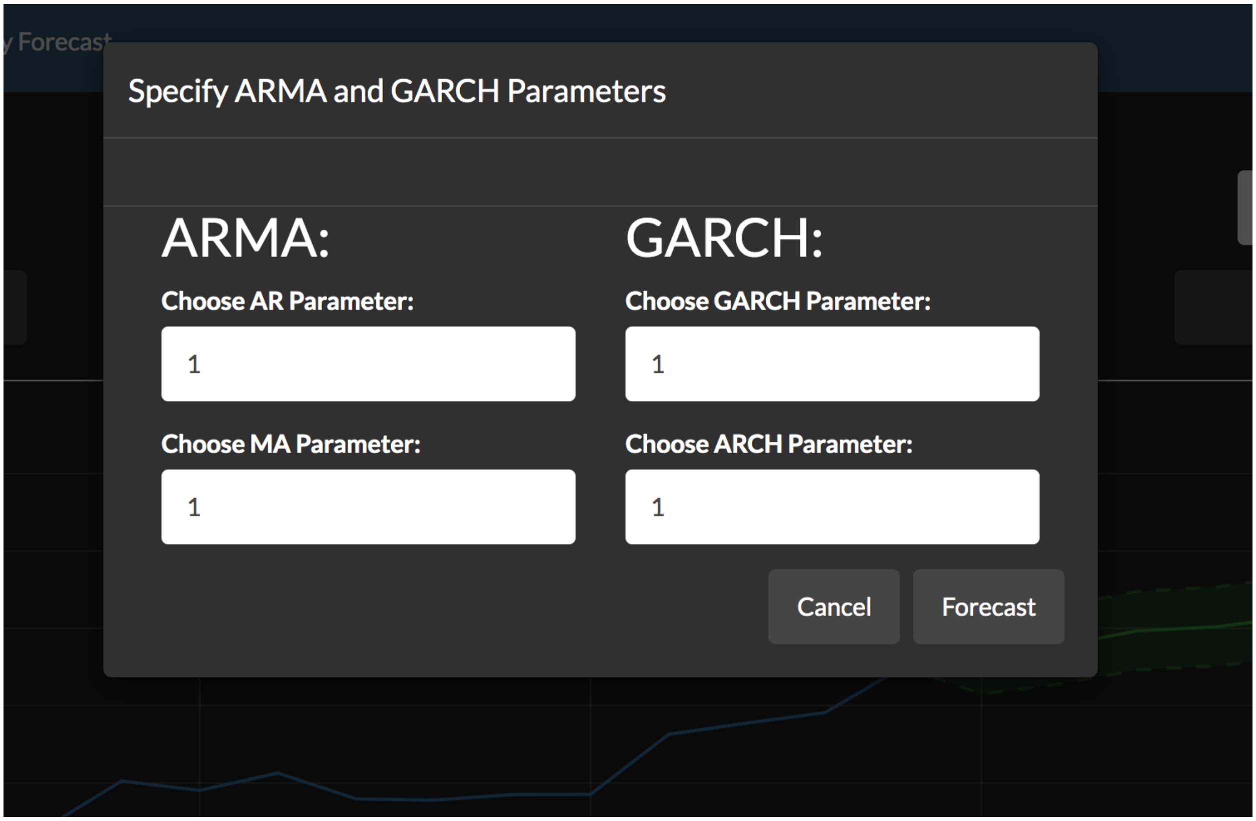


Figure 9: Shown here is the Forecast Model Output. Shows the model which was chosen as the best.

Once a model is chosen, from either the Auto GARCH or Manual GARCH methods, two outputs will appear below the Forecast Input Panel: Forecast Graph Output and Forecast Table Output. These outputs will clearly display the forecasted values for the user. The Forecast Graph Output will help visualize the forecasted values and the Forecast Table Output will show the exact values of the forecasts.

The Forecast Graph Output (Figure 10) shows the actual values from the given date range plus the forecasted values and their 95% confidence interval. The solid blue line in the graph shows the actual values (actual values for the given date range), the solid green line shows the forecasted values (number of forecasts shown depends on what the users put in the Number of Forecast input, in this case it shows 10 forecasts), and the dashed green line shows the 95% confidence interval upper and lower bounds of the closed price. It is important to note that this graph is interactive, meaning the user can hover over the lines to see the date and closing price and the user can zoom in on different sections of the graph. Any zooms can be reset by the house icon in the upper right.

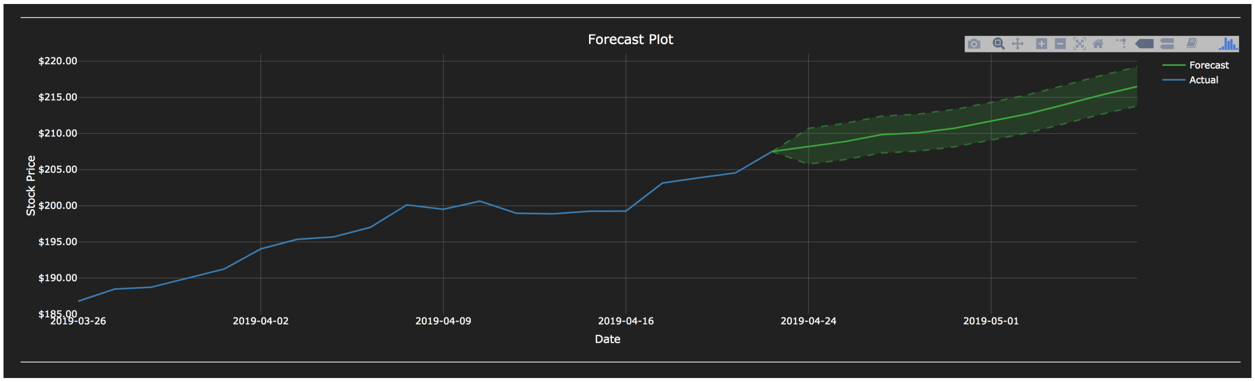
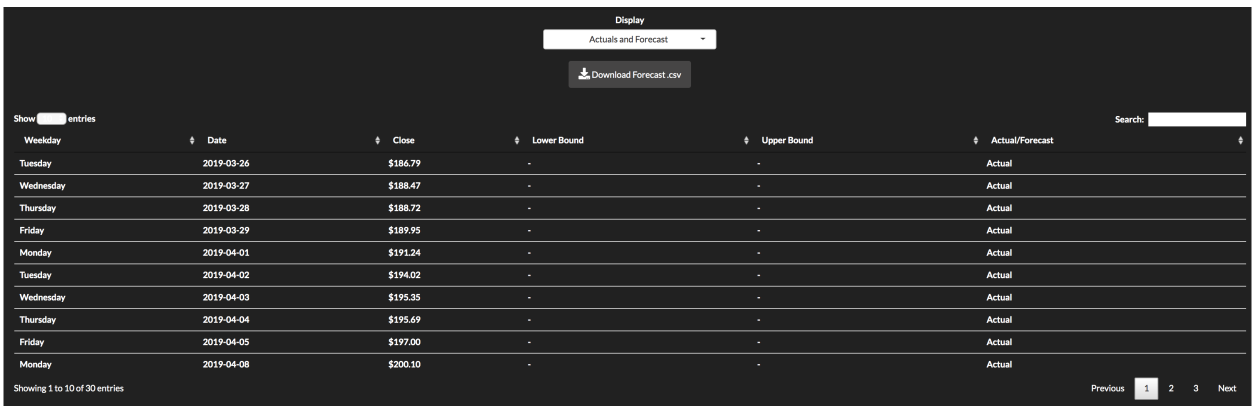


Figure 10: Shown here is the Forecast Graph Output. Shows the forecast and a 95 percent confidence interval.

The Forecast Table Output (Figure 11) shows the actual and forecasted values given the date range and number of forecasts from the Forecast Input Panel. You will notice there is a drop-down input which allows the user to choose whether the table will display the actual and forecasted values or just the forecasted values. Similarly to the Overview Pages, there is a download button. This will download a .csv file of the displayed table. The Forecast Table Output is also interactive, which means the user can do searches using the search bar, and the user can sort the table by different columns, which is defaulted to ascending order of date.



Weekday	Date	Close	Lower Bound	Upper Bound	Actual/Forecast
Tuesday	2019-03-26	$186.79	-	-	Actual
Wednesday	2019-03-27	$188.47	-	-	Actual
Thursday	2019-03-28	$188.72	-	-	Actual
Friday	2019-03-29	$189.95	-	-	Actual
Monday	2019-04-01	$191.24	-	-	Actual
Tuesday	2019-04-02	$194.02	-	-	Actual
Wednesday	2019-04-03	$195.35	-	-	Actual
Thursday	2019-04-04	$195.69	-	-	Actual
Friday	2019-04-05	$197.00	-	-	Actual
Monday	2019-04-08	$200.10	-	-	Actual

Figure 11: Shown here is the Forecast Table Output. The table is interactive.

The Intraday Forecast page is similar to the above Daily Forecast page. The only difference is there an additional input for step. This input allows the user to change between different intraday steps (i.e. 5 min., 15 min., 30 min., and 60 min.), which is defualted to 5 min. The date range default changes to just current date. The rest of the Intraday Overview similar to the Daily Overview, thus we will not go over it in detail.

The Overview and Forecast Pages consitute the entire application. The Overview Pages allow the users to do exploratory data analysis on different stocks in the NYSE. The Forecast Pages allow the users to easily model the data and make forecasts for future date values.

4 Performance

Although the application will automatically create the "best" ARMA+GARCH model and provide forecasts, it does not necessarily mean that those forecasts are accurate. One way to measure our system's accuracy is to look at the forecasted price and compare it to the actual. However, since prices between steps of the data are generally closely related, the "accuracy" of each forecast will be relatively high. Therefore, we should also be interested if the forecast is predicting an increase or decrease.

To evaluate the model's accuracy, we will calculate two metrics: Accuracy of Forecast and Probability of Predicting True Increases/Decreases.

4.1 Accuracy of Forecast

For the Accuracy of Forecast metric, we will train an ARMA+GARCH model using a window and then generate a forecast for the next day. Once we generate a forecast we will compare that forecast to the actual on that day by taking the absolute value of the difference between the actual and forecasted value (absolute percent error). We will then divide that absolute difference by the actual and subtract it from 1, which will give us the accuracy.

$$\text{Absolute Percent Error} = \frac{|(\hat{y}_t - y_t)|}{y_t}$$
$$\text{Accuracy} = 1 - \text{Absolute Percent Error}$$

Since doing this just one is not a good representation of how the models are performing, we will do this in multiple steps and for multiple stocks, so that we have a good sample of how the model is performing. This means that we will collect multiple absolute percent errors for a set of stocks, and take the average absolute percent error (Mean Absolute Percent Error, MAPE), and subtract that from 1 to get an overall accuracy of the model (for each stock).

$$\text{Mean Absolute Percent Error (MAPE)} = \frac{1}{n}\sum_{i=1}^{n}\frac{|(\hat{y}_i - y_i)|}{y_i}$$
$$\text{Overall Accuracy} = 1 - \text{MAPE}$$

4.2 Probability of Predicting a True Increase/Decrease

For the Probability of Predicting a True Increase/Decrease metric, we are concerned if our model is forecasting an increase or decrease in stock price accurately. This means rather than looking at the price, we will look at the forecasted increase/decrease. Similar to the accuracy metric, we will train an ARMA+GARCH model using a window and then generate a forecast for the next day. We determine a forecasted increase or descrease by comparing the forecast at time t (the first forecast) to the previous step at time $t-1$ (the last known value). An increase will be determined if the forecast is greater than the previous step, and a decrease will be determined if the forecast is less than the previous step:

$$\hat{y}_t > y_{t-1} \Rightarrow \text{Forecasted Increase}$$
$$\hat{y}_t < y_{t-1} \Rightarrow \text{Forecasted Decrease}$$

After determining the forecasted increase/descrease, we will look at the actual increase/descrease. To determine this, we will compare the actual at time t to the actual at time $t-1$. An increase will be determined if the actual at time t is greater than the actual at time $t-1$, and a decrease will be determined if the actual at time t is less than the actual at time $t-1$:

$$y_t > y_{t-1} \Rightarrow \text{Actual Increase}$$
$$y_t < y_{t-1} \Rightarrow \text{Actual Decrease}$$

Once again, since doing this just once is not a good representation of the models, we will do this in multiple steps and for multiple stocks, which will give us a good sample of how the model is performing. After completing this in multiple steps we will compare all the forecasted directions with their actual direction and calculate how many we got correct. The accuracy will be calculated by taking the sum of the number of correct forecasted directions divided by the total number of forecasts:

$$\text{Direction Accuracy} = \frac{\text{Number of Correct Forecasts}}{n}$$

$$\text{Number of Correct Forecasts} = \sum_{i=1}^{n} d_i$$

$$d_i = \begin{cases} 0 & \text{Forecasted Direction} \neq \text{Actual Direction} \\ 1 & \text{Forecasted Direction} = \text{Actual Direction} \end{cases} \forall i \in 1, 2, 3, ..., n$$

4.3 Methodology

The methodoly for evaluating performance is quite simple: determine a date range (window size) to train the model, make forecasts for days which have known actuals, compare the forecasts to actuals, shift the date range, make new forecasts for days which have known actuals, compare the forecasts to actuals, and so on. For our window size, we determined to use data beginning in 2010. This would eliminate most issues due to the recession that occured in 2008 and 2009. Also, data beginning in 2010 would make sure we had sufficient amount of history, which would allow for more accurate forecasts. We decided that our date range for the training would be from January 1, 2010 to December 31, 2018. We also decided to forecast the next five trading days, which would allow us to not only see how well the model does on one day forecasts but an entire weeks worth of forecasts.

For our first interation of forecasts we will use the date range from January 1, 2010 to December 31, 2018 as our training window for the model. We will determine the best ARMA+GARCH model through the automation process and then forecast the next five trading days. This means that we will have forecasts for January 2 2019, January 3 2019, January 4 2019, January 7 2019, and January 8 2019 (Note that January 1 is New Years, thus the stock market is close so we did not have a forecast for that day). We will then compare those forecasts to the last known value of December 31 2018 and determine if that stock is increasing or decreasing. We will then compare the actual prices of those five trading days with December 31 2018 to determine if they were actually increasing or decreasing. Finally, we will compare the forecasted increase or decrease to the actual increase or decrease to determine if the model accurately forecasted in the correct direction. Lastly, we will calculate the MAPE (shown in section 4.1) of the five forecasted values.

For our second interation of forecasts we will use the date range from January 1, 2010 to January 02, 2019 as our training window for the model (Note that we skipped January 1 2019 since that was not a trading day). Once again, we will determine the best ARMA+GARCH model through the automation process and then forecast the next five trading days. This means that we will have forecasts for January 3 2019, January 4 2019, January 7 2019, January 8 2019, and January 9 2019. Similarly to the first iteration, we will compare those forecasts to the last known value of January 02 2019 and determine if that stock is increasing or decreasing, and we will compare the actual prices of those five trading days with January 02 2019 to determine if they were actually increasing or decreasing. We will compare the forecasted increase or decrease to the actual increase or decrease and determine if the model accurately forecasted in the correct direction. We will also calculate the MAPE of the five forecasted values.

This process will continue for a total of 56 iterations, which means we will have a total of 280 forecasts to compare to the actuals. This should be a sufficient amount of history to determine how the automated modeling is performing.

4.4 Evaluation

Since we there are thousands of stocks traded, we are unable to evaluate the model for each stock. Jeff Anderson provided a list of stocks that are either commonly traded, have had recent volatility, or have low prices (less than a dollar). The stocks which were provided for evaluation are:

- Apple (AAPL)
- Amazon (AMZN)
- AT&T (T)
- Dow Jones Industry Average (DIA)
- Facebook (FB)
- General Electric (GE)
- Hemp (HEMP)
- Micron Technology (MU)
- S&P 500 (SPY)
- Tesla (TSLA)

In the following sections, we will review how the automated modeling process performed in the above stocks. It is important to note that we are mainly interested in how well our model predicts the directional change. Our second concern the MAPE performance.

4.4.1 Apple (AAPL)

Apple is a frequently traded stock (high volume) on the NYSE, with relatively low volatility (Figure 12). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 12: Apple (AAPL) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance history, we get the results below (Table 1). We see that the overall percent accuracy for predicting directional change is 70.36%. This indicates that we are doing better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our first forecast (meaning the next trading day forecast) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our fifth forecast (meaning five forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 2.10%. Contrary to the directional change predictions, we see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 1: Apple Inc. (AAPL) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	35	56	62.50%	1.29%
Second Forecast	39	56	69.64%	1.93%
Third Forecast	37	56	66.07%	2.24%
Fourth Forecast	41	56	73.21%	2.36%
Fifth Forecast	45	56	80.36%	2.69%
Overall	**197**	**280**	**70.36%**	**2.10%**

4.4.2 Amazon (AMZN)

Amazon is a frequently traded stock (high volume) on the NYSE, with recent high volatility (Figure 13). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 13: Amazon (AMZN) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 2). We see that the overall percent accuracy for predicting directional change is 60.36%. This indicates that we are doing better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our first forecast (meaning the next trading day forecast) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our fifth forecast (meaning five forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 2.20%. Contrary to the directional change predictions, we see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 2: Amazon (AMZN) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	30	56	53.57%	1.45%
Second Forecast	35	56	62.50%	2.16%
Third Forecast	33	56	58.93%	2.35%
Fourth Forecast	34	56	60.71%	2.45%
Fifth Forecast	37	56	66.07%	2.60%
Overall	**169**	**280**	**60.36%**	**2.20%**

4.4.3 AT&T (T)

AT&T is a frequently traded stock (high volume) on the NYSE, with relatively constant high volatility (Figure 14). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 14: ATT (T) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 3). We see that the overall percent accuracy for predicting directional change is 59.28%. This indicates that we are doing slightly better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our first forecast (meaning the next trading day forecast) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our fifth forecast (meaning five forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 1.41%. Contrary to the directional change predictions, we see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 3: ATT (T) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	30	56	53.57%	0.92%
Second Forecast	30	56	53.57%	1.26%
Third Forecast	35	56	62.50%	1.45%
Fourth Forecast	34	56	60.71%	1.66%
Fifth Forecast	37	56	66.07%	1.77%
Overall	**166**	**280**	**59.28%**	**1.41%**

4.4.4 Dow Jones Industry Average (DIA)

Dow Jones Industry Average is a frequently traded stock (high volume) on the NYSE (and a good representation of the overall market), with relatively low volatility (Figure 15). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 15: Dow Jones Indusry Average (DIA) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 4). We see that the overall percent accuracy for predicting directional change is 63.93%. This indicates that we are doing better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our first forecast (meaning the next trading day forecast) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our fifth forecast (meaning five forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 0.95%. Contrary to the directional change predictions, we see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 4: Dow Jone Industrial Average (DIA) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	28	56	50.00%	0.65%
Second Forecast	35	56	62.50%	0.83%
Third Forecast	38	56	67.86%	0.95%
Fourth Forecast	39	56	69.64%	1.08%
Fifth Forecast	39	56	69.64%	1.22%
Overall	**179**	**280**	**63.93%**	**0.95%**

4.4.5 Facebook (FB)

Facebook is a frequently traded stock (high volume) on the NYSE, with recent dip in prices (Figure 16). We will use the the performance methodology from the previous section to compute performance metrics.

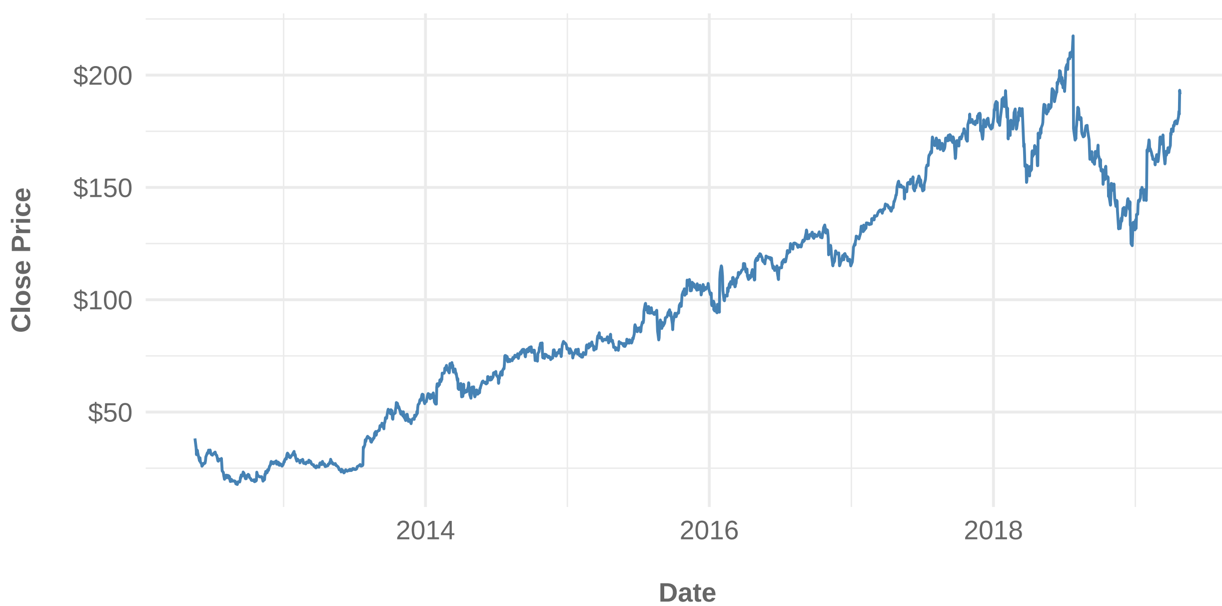


Figure 16: Facebook (FB) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 5). We see that the overall percent accuracy for predicting directional change is 57.50%. This indicates that we are doing slightly better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our second forecast (meaning two forecasted trading days from the last known day) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our fifth forecast (meaning five forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 1.82%. We see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 5: Facebook (FB) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	30	56	53.57%	1.00%
Second Forecast	27	56	48.21%	1.60%
Third Forecast	30	56	53.57%	2.04%
Fourth Forecast	27	56	48.21%	2.14%
Fifth Forecast	47	56	83.93%	2.34%
Overall	**161**	**280**	**57.50%**	**1.82%**

4.4.6 GE (General Electric)

General Electric is a frequently traded stock (high volume) on the NYSE, with dramatic drop in prices over the past two years (Figure 17). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 17: General Electric (GE) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 6). We see that the overall percent accuracy for predicting directional change is 47.50%. This indicates that we are doing worse than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our second forecast (meaning two forecasted trading days from the last known day) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our first forecast (meaning the next trading day forecast) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 4.04%. We see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 6: General Electric (GE) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	29	56	51.79%	2.37%
Second Forecast	25	56	44.64%	3.55%
Third Forecast	28	56	50.00%	4.17%
Fourth Forecast	24	56	42.86%	4.56%
Fifth Forecast	27	56	48.21%	5.55%
Overall	**133**	**280**	**47.50%**	**4.04%**

4.4.7 Hemp (HEMP)

Hemp is a frequently traded stock (high volume) on the NYSE, with extremely low volatility over the past few years (Figure 18). We will use the the performance methodology from the previous section to compute performance metrics.

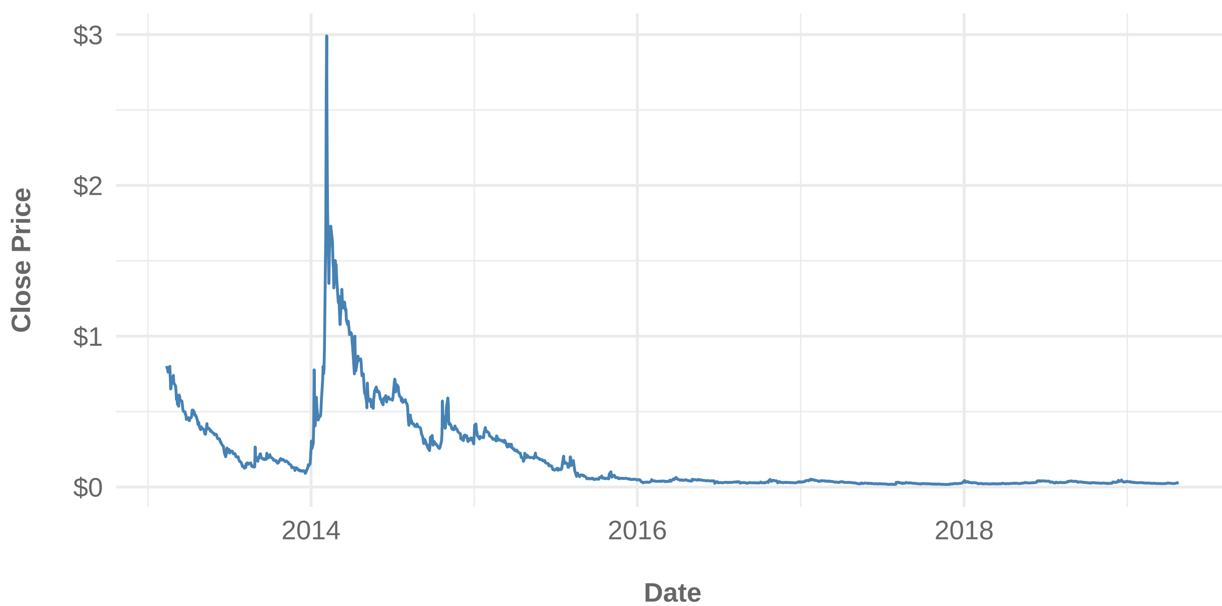


Figure 18: Hemp (HEMP) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 7). We see that the overall percent accuracy for predicting directional change is 72.14%. This indicates that we are doing better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our first forecast (meaning the next trading day forecast) has the lowest percent accuracy on predicting the directional change, meaning it is performing the worst out of the five forecasts. Our second, third, and fourth forecasts (meaning two, three, and four forecasted trading days from the last known day) have the highest percent accuracy on predicting the direction change, meaning they are equally performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 3.13%. We see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 7: Hemp (HEMP) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	36	56	64.29%	1.92%
Second Forecast	42	56	75.00%	2.61%
Third Forecast	42	56	75.00%	3.46%
Fourth Forecast	42	56	75.00%	3.51%
Fifth Forecast	40	56	71.43%	4.16%
Overall	**202**	**280**	**72.14%**	**3.13%**

4.4.8 Micron Technology (MU)

Micron Technology Apple is a frequently traded stock (high volume) on the NYSE, with recent high volatility (Figure 19). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 19: Micron Technology (MU) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 8). We see that the overall percent accuracy for predicting directional change is 53.93%. This indicates that we are doing slightly better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our second forecast (meaning two forecasted trading days from the last known day) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our third forecast (meaning three forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 3.76%. We see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 8: Micron Technology (MU) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	31	56	55.36%	2.48%
Second Forecast	27	56	48.21%	3.32%
Third Forecast	35	56	62.50%	3.78%
Fourth Forecast	30	56	53.57%	4.45%
Fifth Forecast	28	56	50.00%	4.75%
Overall	**151**	**280**	**53.93%**	**3.76%**

4.4.9 S&P 500 (SPY)

S&P 500 is a frequently traded stock (high volume) on the NYSE (and a good representation of the overall market), with relatively low volatility and steady increases (Figure 20). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 20: SP 500 (SPY) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 9). We see that the overall percent accuracy for predicting directional change is 65.00%. This indicates that we are doing better than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our first forecast (meaning the next trading day forecast) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our third forecast (meaning three forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 0.94%. We see that our first forecast has the smallest MAPE, meaning it is performing best, and the fifth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 9: SP 500 (SPY) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	29	56	51.79%	0.63%
Second Forecast	35	56	62.50%	0.86%
Third Forecast	41	56	73.21%	0.95%
Fourth Forecast	38	56	67.86%	1.05%
Fifth Forecast	39	56	69.64%	1.20%
Overall	**182**	**280**	**65.00%**	**0.94%**

4.4.10 Tesla (TSLA)

Tesla is a frequently traded stock (high volume) on the NYSE, with high volatility (Figure 21). We will use the the performance methodology from the previous section to compute performance metrics.

Figure 21: Tesla (TSLA) stock daily closing price from January 1, 2010 to April 18, 2019

Running the performance methodology, we get the results below (Table 10). We see that the overall percent accuracy for predicting directional change is 47.14%. This indicates that we are doing worse than a coin flip at predicting whether or not the stock will increase or decrease in price. We can see that our fourth forecast (meaning four forecasted trading days from the last known day) has the lowest percent accuracy on predicting the directional change, thus it is performing the worst out of the five forecasts. Our fifth forecast (meaning five forecasted trading days from the last known day) has the highest percent accuracy on predicting the direction change, thus it is performing the best out of the five forecasts.

When looking at the accuracy of the actual price versus forecasted price, we see that the overall MAPE is 4.12%. We see that our first forecast has the smallest MAPE, meaning it is performing best, and the fourth forecast has the largest MAPE, meaning it is performing the worst. Heuristically this makes sense because the more your forecast out, the less accurate the price predictions are since the forecasts build upon eachother.

Table 10: Tesla (TSLA) Automated Stock Performance

Step	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
First Forecast	25	56	44.64%	2.57%
Second Forecast	28	56	50.00%	3.91%
Third Forecast	25	56	44.64%	4.54%
Fourth Forecast	24	56	42.86%	5.03%
Fifth Forecast	30	56	53.57%	4.56%
Overall	**132**	**280**	**47.14%**	**4.12%**

4.4.11 Overall

We are concerned with how the automated modeling performs overall. To get a sense of how well (or poor) the automated modeling is doing, we combined the results from each stock and calculated an overall accuracy and MAPE (Table 11). From the overall table, we see that Apple (AAPL) and Hemp (HEMP) performed the best when it came to forecasting the directional change. When considering the MAPE, the Dow Jone Industry Average (DIA) and the S&P 500 (SPY) had the lowest percents, meaning they were most accurate at forecasting the actual price. On the other hand, we have that General Electric (GE) and Tesla (TSLA) performed the worst when it came to forecsting the directional change. We also see that General Electric and Tesla had the highest MAPE, meaning they were the least accurate at forecasting the actual price.

Table 11: Combined Automated Stock Performance

Stock	Correct Direction	Total Forecasts	Percent Accuracy	MAPE
AAPL	197	280	70.36%	2.10%
AMZN	169	280	60.36%	2.20%
DIA	179	280	63.93%	0.95%
FB	161	280	57.50%	1.82%
GE	133	280	47.50%	4.04%
HEMP	202	280	72.14%	3.13%
MU	151	280	53.93%	3.76%
SPY	182	280	65.00%	0.94%
T	166	280	59.28%	1.41%
TSLA	132	280	47.14%	4.12%
Overall	**1672**	**2800**	**59.71%**	**2.45%**

Another thing to know is how the different stocks are performing given both the directional change accuracy and the MAPE, since some stocks may have higher directional change accuracy and a higher MAPE, others may have a lower directional change accuracy and a lower MAPE. We would like a way to combine these two measurements into one and see which stocks are the best to use on the application. Since a larger directional change accuracy is desired and a smaller MAPE is desired, multiplying the two will not give us a useful (relative score). Instead, to do this combination, we will divide the MAPE (as a percent) by the direction change accuracy (as a decimal), which will give use an indicator/score of the relative performance on a stock across both measurements. We will then sort the list of stocks in ascending order and the stocks with the lowest combined score will be the best performing stocks on the application, relative to the other stocks (Table 12).

Table 12: Relative Automated Stock Performance

Stock	Correct Direction	Total Forecasts	Percent Accuracy	MAPE	Relative Performance Score
SPY	182	280	65.00%	0.94%	1.45
DIA	179	280	63.93%	0.95%	1.49
T	166	280	59.28%	1.41%	2.38
AAPL	197	280	70.36%	2.10%	2.98
FB	161	280	57.50%	1.82%	3.17
AMZN	169	280	60.36%	2.20%	3.64
HEMP	202	280	72.14%	3.13%	4.34
MU	151	280	53.93%	3.76%	6.97
GE	133	280	47.50%	4.04%	8.51
TSLA	132	280	47.14%	4.12%	8.74

From the above table, we can see that the S&P 500 (SPY) and the Dow Jones Industry Average (DIA) are the best performing stocks on the application, given the two measurements of performance. We can also see that Apple (AAPL) and Hemp (HEMP) are fourth and seventh, respectively, meaning that a high directional change accuracy does not indicate that the stock is performing the best.

5 Conclusion

This project combined the statistical modeling knowledge and data science topics learnt throughout the Master of Science in Mathematics with a concentration in Statistics program at the University of Nebraska - Omaha. I used these skills to create a web application that achieved multiple goals: I used a Yahoo API to streamline data from the NYSE, I then automated and generalized statistical models (ARMA+GARCH time series) to model any stock and provide forecasts. The application made it easy for the user to interact and get the desired outputs.

The automated modeling did not perform as well as I hoped, however, it was performing better than guessing. To evaluate the performance of the automated modeling, we looked at ten different stocks. For these ten stocks, we ran historical models and compared the forecasted values to the (now) known values from those days. The overall performance across all of the ten stocks was 59.71% on forecasting the correct direction change and 2.45% on the MAPE.

Since the data used for this project is publicly available, one can request to view the application and the associated .R files by contacting Dr. Swift from the Mathematics Department at the University of Nebraska - Omaha or by contacting Jordan Wheeler (via: jordanwheeler@outlook.com). One may also view all materials related to this project (including the .R files for the application and this write up) at https://github.com/jordanmwheeler/UNO-MasterProject.

References

Bollerslev, Tim. 1986. “Generalized Autoregressive Conditional Heteroskedasticity.” *Journal of Econometrics* 31 (3): 307–27. doi:10.1016/0304-4076(86)90063-1.

Cryer, Jonathan D., and Kung-sik Chan. 2011. *Time Series Analysis with Applications in R*. Springer.

Engle, Robert F. 1982. “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation.” *Econometrica* 50 (4): 987–1007. doi:10.2307/1912773.

Popivanov, Ivan. 2013. “Automatic Arma/Garch Selection in Parallel.” *R*. https://www.r-bloggers.com/author/ivannp/.