USER MANUAL

A Purposeful Walk Down Wallstreet

Nabeel Asghar | Michael Shields | Shojib Miah | Michael Chen
Wayne State University –April 16th, 2020
Senior Capstone Project

Revision History

Date	Description	Author	Comments
4/16/2020	Version 1	Nabeel Asghar, Michael Shields, Shojib Miah, Michael Chen	Original document that we submitted to canvas

Table of Contents

1. Introduction

GM FinTech is a stock price analysis application that trends, forecasts and simulate stock prices using five algorithms and five strategies along with three newly added algorithms that use several macroeconomic variables on six different stock symbols. This application is designed as part of Capstone project by senior undergraduate students in the department of Computer Science at Wayne State University under the advising of Dr. Seyed Ziae Mousavi Mojab and Mr. Joshua Feinstein Lead in the Global Data, Artificial Intelligence & Analytics Services department at General Motors Corporation in Detroit, Michigan. With GM FinTech application an experienced stock data analyst can make decisions to buy, sell or hold stocks, monitor stock prices, observe the simulated trading strategies performance and view the stock forecasts based on different algorithms.

1.1 What You Will Learn

The following chapters will give you a tour of the GM FinTech application and a tutorial on how to update the database. You will learn to do the following:

- Run the data files
- Load data to database
- Use the Tableau front end
- Manage the instrument master table

1.2 Before You Begin

All software must be installed as described in the product design specification document. Following is the list for your reference:

- MySQL 8.0 or later (Workbench), Username: **root** – Password: **password**
- PyCharm Professional Edition 2019.2 or later (with all required packages)
- Python 3.7
- PIP 19.3.1 or later (PyPA recommended tool for installing Python packages)
- Tableau Desktop Professional Edition 2019.3.0 or later
- Microsoft Windows 10 PC with Intel 10th generation processor and 8GB memory, minimum screen resolution of 1366 x 768
- Download all the source code from https://github.com/mshields11/GM-Senior-Project

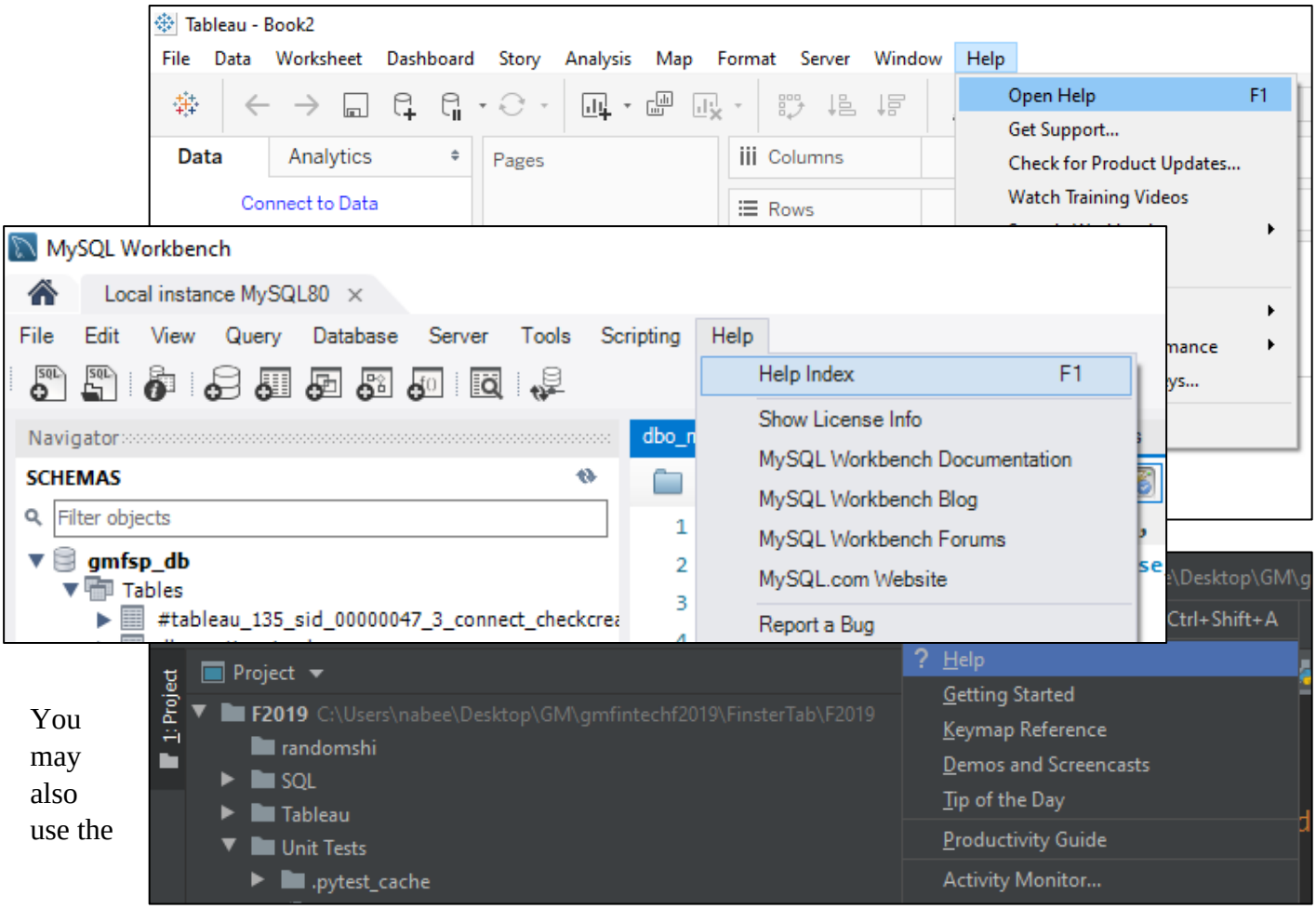
1.3 What You Need to Get Started

A Microsoft Windows computer with all the required software as listed in section 1.2 above and in the product design specification document, source code, and an internet connection. All software must be installed using the README file.

1.4 Help Resources

There is no custom onscreen help available in the GM FinTech application. Tooltips on each graph and measure will provide information regarding the calculations.

As shown below, you can use Tableau, PyCharm, or MySQL help and support by clicking on the built-in **Help** menu.



You may also use the following resources to help you for any help:

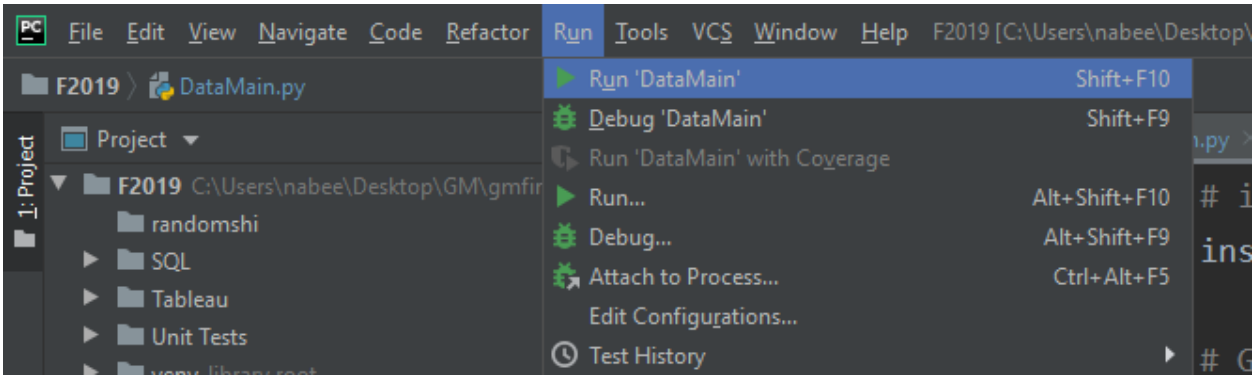
- https://www.tableau.com/support/help
- https://community.tableau.com/welcome
- https://www.python.org/doc/

2. The Backend

The next few chapters will provide to instructions to run this entire application from its back-end MySQL database to its front-end Tableau interface.

2.1 Running the Backend

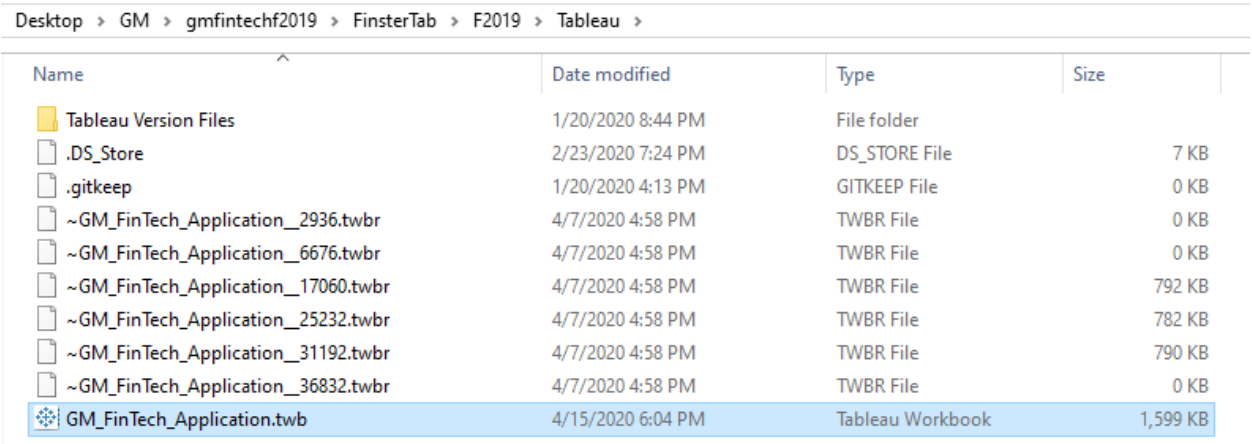
1 Open the project in PyCharm.
2 Make sure the local instance of MySQL is setup with the username and password listed before.
3 Run DataMain.py. As this is run, the fetching and calculation of data is being done. This may take a long time when running the first time.



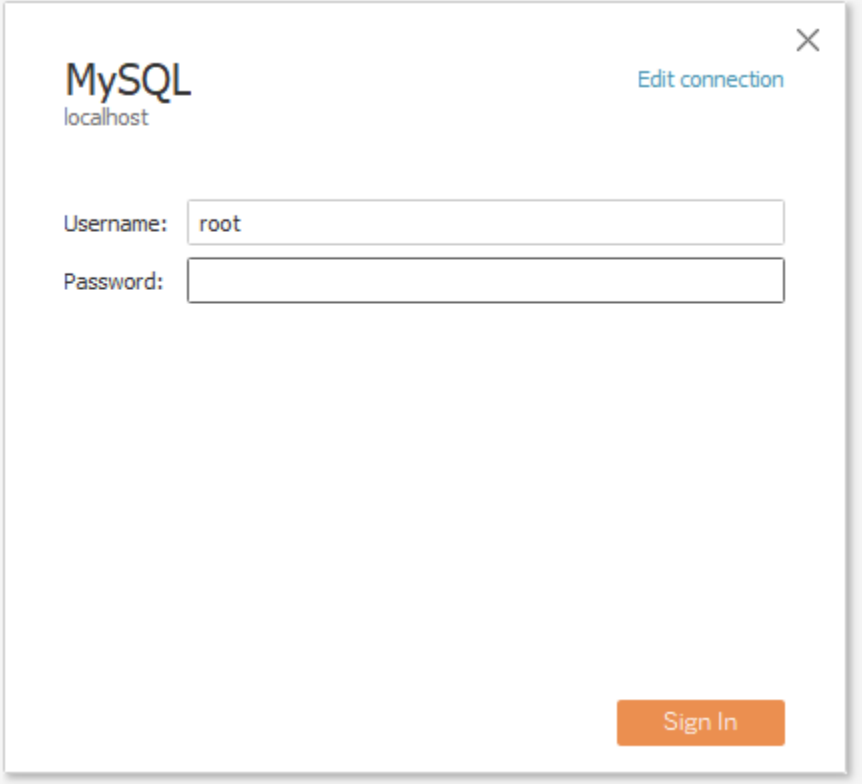
3. The Frontend

3.1 Running the Frontend

1. Navigate to the folder FinsterTab/F2019/Tableau
2. Run the file called GM_FinTech_Application.twb

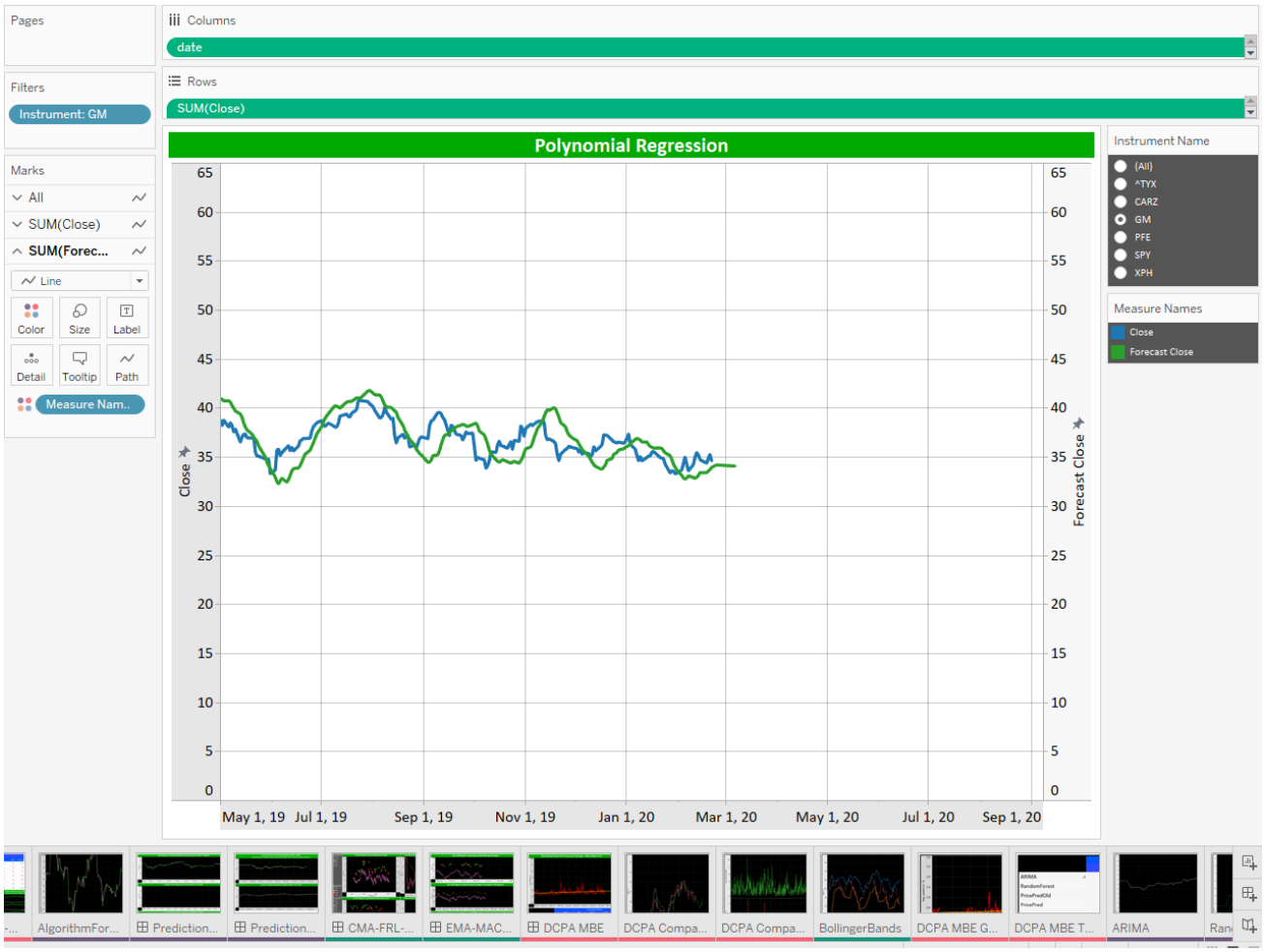


3. Login to the application with the username of **root** and the password of **password.**



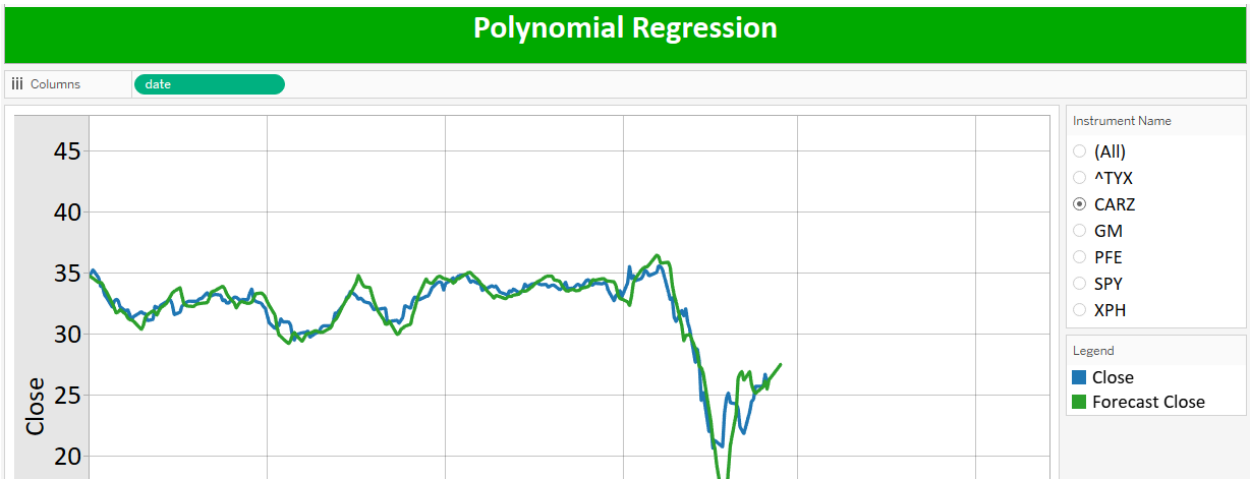
3.2 Tableau Sheets

As you can on the image below, you will see a sheet like this when you login. There are several tabs across the bottom of the application. Each tab is designed to show a different type of measure and view.



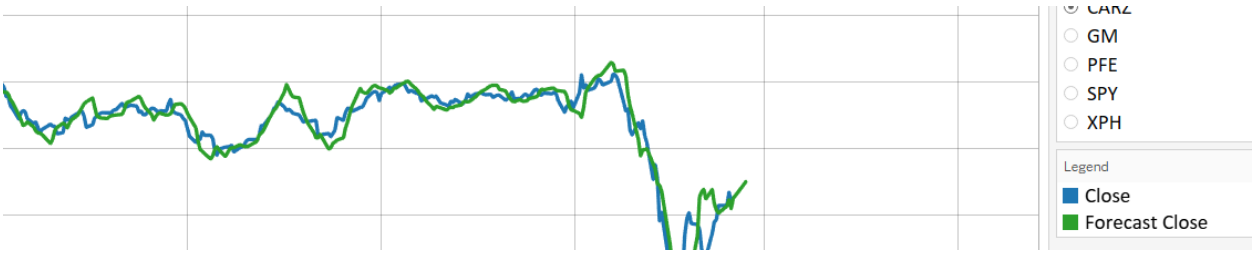
3.3 Radio Buttons

Graphs and metrics have radio buttons to the right or left of the screen. These buttons can be used to make selections according to your requirement.



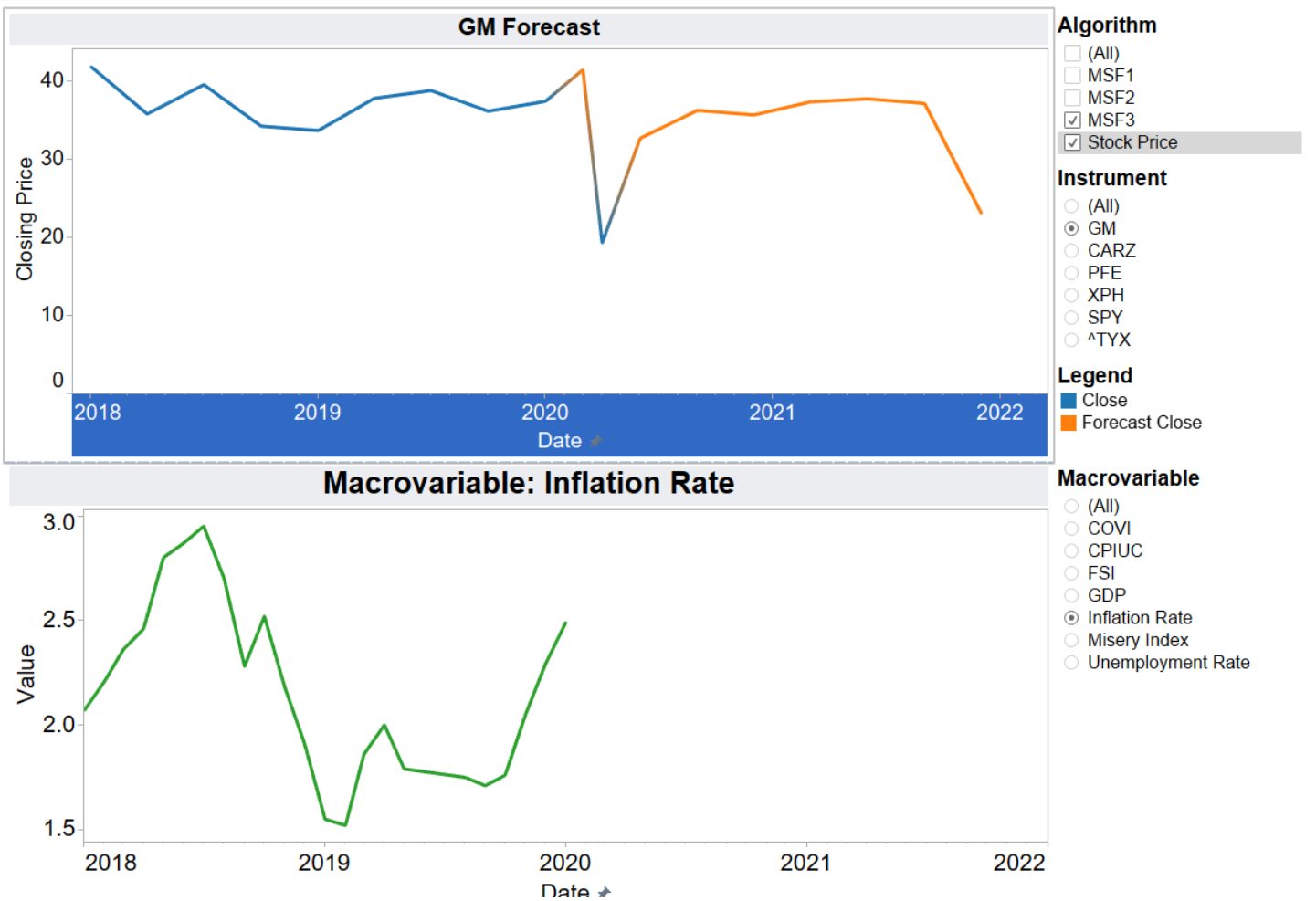
3.4 Legends

All graphs include color coded legends for each measure and line. All values are color coded throughout the application. For instance, ‘Close’ price is always in Pink color on all graphs as shown on the right side.



3.5 Dashboard

A dashboard in Tableau is created by importing tabs in the application on a single page. There are several dashboard tabs available in the application, independent of other instances, so it is possible to install the application on multiple user machines on the same network, or domain, and set them to run for separate financial instruments.



4. Database

GM FinTech application's backend is built using a MySQL database. A local instance is created with the username **'root'** and password **'password'**. MySQL is a free database solution.

4.1 MySQL Workbench

As mentioned in section 1.2, MySQL Workbench is needed to execute any SQL script.

4.2 Add or Remove Instruments

Start MySQL Workbench and go to the script file
"INSERT_INTO_INSTRUMENT_MASTER_MYSQL.sql"

You will get the following:

```
insert into dbo_instrumentmaster
values (1 , 'GM'   , 'Equity' , 'YAHOO'),
       (2 , 'PFE'  , 'Equity' , 'YAHOO'),
       (3 , 'SPY'  , 'Equity' , 'YAHOO'),
       (4 , 'XPH'  , 'Equity' , 'YAHOO'),
       (5 , 'CARZ' , 'Equity' , 'YAHOO'),
       (6 , '^TYX' , 'Equity' , 'YAHOO')
;
```

To insert a new instrument, add a new line with the instrument name and ID and source.

To remove an instrument, just remove the line with the instrument.

Appendix

Appendix A: Financial Base Data

These are the data elements extracted from the YAHOO! Finance data exchange API. This data is loaded into '**dbo_instrumentstatistics'** table in the database.

Date	Trading date
High	Highest price reached on the trading date
Low	Lowest price reached on the trading date
Open	Price at which the stock opened trading on the trading date, please note that this is not necessarily the closing price of yesterday.
Close	Closing price of a single unit/share of stock at the end of the trading day. This is the most talked about and important number in this application. Trading closes at 4:00pm Eastern Time.
Volume	Number of shared bought and sold during the trading day.
Adj Close	The adjusted close price analyzes the stock's dividends, stock splits and new stock offerings to determine an adjusted value. The adjusted closing price reflects the change in stock value caused by new offerings from the corporation

Appendix B: Strategy Formulas

B1 CMA (Cross Moving Average)

The CMA strategy is based on the comparison of moving averages of varying lengths. In short, when the averages are close together a reversal is more likely to be occurring. When the averages are farther apart, an upward or downward trend is more likely to be developing.

wcma = 7-day moving average
scma = 20-day moving average
lcma = 100-day moving average

5-day average = 5-day simple moving average
BuyWeekApproach = previous day's wcma /
lcma week_long = wcma / lcma

SellWeekApproach = previous day's wcma / scma

week_short = wcma / scma momentumA
= price / price 5 days ago

BUY WHEN:

-*BuyWeekApproach* is between 0.977 and 1.025

-*week_long* is greater than 1.018

-*momentumA* is greater than 1

SELL WHEN:

-S*ellWeekApproach* is greater than 1

-*week_short* is greater than 0.93

-*wmca* from 3 days ago is greater than *wmca* today

HOLD: In all other situations

B2 FRL (Fibonacci Retracement Lines)

Fibonacci Retracement Lines are common trading indicators that can be used in a variety of ways. Signals are generated based on the recent price behavior when the price is close to one of the retracement levels. In this strategy five different levels are used in addition to other momentum calculations.

lpeak = maximum price in the past 100 days
*highfrllinelong = 0.764 * lpeak*
*medfrllinelong = 0.618 * lpeak*
*lowfrllinelong = 0.382 * lpeak*

ltrough = minimum price in the past 100 days

ActualChange = today's price / yesterday's price

momentumA = price / price 5 days ago

BUY WHEN:

-close price is between 1.25% and 2.25% above *highfrllinelong, medfrllinelong,* or *lowfrllinelong*

-*momentumA* is less than 0.99

SELL WHEN:

-close price is between 2.5% and 1.5% below *highfrllinelong, medfrllinelong,* or *lowfrllinelong*

-*momentumA* is greater than 0.99

-*ActualChange* yesterday minus *ActualChange* today is less than 0.1

HOLD: In all other situations

B3 EMA (Exponential Moving Average)

The EMA strategy is similar to the CMA strategy but replaces the simple moving average calculation with a weighted average that places exponentially more weight on prices as the they move closer and closer to today.

sema = 20-day moving average
mema = 50-day moving average
lema = 100-day moving average

5-day avg = 5-day simple moving average sigMid = sema / mema

sigLong = sema / lema
momentumA = price / 5-day avg

BUY WHEN:

-*sigLong* is less than 1

-*momentumA* is greater than 0.97

SELL WHEN:

-*sigMid* is between 0.983 and 1.004

-*momentumA* is less than 1.012

HOLD: In all other situations

B4 MACD (Moving Average Convergence Divergence)

The moving average convergence/divergence (MACD) is another commonly used technical indicator.

macd = difference between the 12-day exponential moving average and the 26-day exponential moving average

macds = 9-day exponential moving average of macd

BUY WHEN:

-yesterday's *macd* < yesterday's *macds*

-today's *macd* > today's *macds*

SELL WHEN:

-yesterday's *macd* > yesterday's *macds*

-today's *macd* < today's *macds*

HOLD: In all other situations

B5 Algorithm Forecast

This signaling strategy is based on the next day directional prices forecasts generated by the program. It uses the price prediction model and the ARIMA model to generate signals.

BUY WHEN:

-price prediction model generates a 'buy' signal
-ARIMA model generates a 'buy' signal

SELL WHEN:

-price prediction model generates a 'sell' signal
-ARIMA model generates a 'sell' signal

HOLD: In all other situations

B6 Custom Combined

The combination strategy takes into consideration each of the signals generated by the previous five strategies. A 'buy' signal is assigned a value of 1, a 'sell' signal is assigned a value of -1, and a 'hold'

is assigned a value of 0. The five individual signals generated for each day are then summed together to produce a *signalsum* value. This value will be greater than or equal to -5 but less than or equal to 5.

BUY WHEN:

-*signalsum* is greater than 0

SELL WHEN:

-*signalsum* is less than 0

HOLD WHEN:

-*signalsum* equals 0

B7 Buy and Hold

The buy and hold strategy can be considered a baseline against which the performance of all other algorithms can be measured. Beginning on the first day pricing data is available as many shares as possible are bought. No further action is taken after this. The change in the value of the asset will be reflected directly in the change in the value of the portfolio.

BUY WHEN:

-first day pricing data is available, or when you initiate your portfolio

SELL WHEN:

-never

HOLD: Always hold, no trading activity should occur

APPENDIX C: Engineered Features/Technical Indicators

Simple Moving Average (SMA): An unweighted mean of each of the previous ***'n'*** data points.

Exponential Moving Average (EMA): A weighted mean of each of the previous *n* data points. Weighting factors that decrease exponentially are applied to each older data point.

Relative Strength Index (RSI): An oscillating indicator that measures the degree to which a price is overbought or oversold compared to recent prices. The average of 'up' price closes is divided by the average of 'down' price closes. This ratio is then converted to a value between 0 and 100. Traditionally, a measure above 70 is considered over bought and a measure below 30 is considered oversold.

Fibonacci Retracement Lines (FRL): A tool used to calculate potential areas of support and resistance in the price range of a stock. The highest and lowest price in the past *n* days are taken as the peak and trough, respectively. Additional levels are then calculated at 23.6%, 38.2%, and 61.8% between the trough and peak. In combination with another indicator that signals direction, these levels can be used to initiate trades when prices approach one of these levels

Moving Average Convergence/Divergence (MACD): A momentum indicator that shows the difference between a short period exponential moving average and a long period moving average. This is known as the *MACD line*. Another exponential moving average is then calculated from the MACD line and is called the *signal line*. Once these two measures are calculated the daily difference is then plotted as a histogram. A positive and increasing histogram is a sign of a trend moving upwards. A negative and decreasing histogram is a sign of a trend moving downwards.

Bollinger Bands: A range based around an ***'n'*** period simple moving average. A certain number of standard deviations – usually two – is then calculated above and below the average to form the outer 'bands' of the range. The closer it moves to the upper or lower band is a signal of how over-bought or over-sold the market is.

APPENDIX D: Custom Trading Simulation Algorithm

Our custom trading simulation strategy uses ARIMA + Previous group's improved prediction + FRL + CMA + EMA + MACD predictions for next day.

Signal Values

- If the price is predicted to go up each of the above algorithms or strategies will generate a value of 1
- if the price is predicted to go down each of the above algorithms or strategies will generate a value of -1
- If the price is predicated to stay the same, each of the above algorithms or strategies will generate a value of 0

ActionSignals' table strategycode = algo (BuySell Custom) Definition

Date	Instrument	ARIMA	PrevGroupCorr	AlgoSignal
11/11/2019	GM	Up	Up	1
11/12/2019	GM	Down	Up	0
11/13/2019	GM	Down	Down	-1

Hold = 0
Sell = -1
Buy = 1
Up = next day's prediction is higher than today's close
Down = next day's prediction is lower than today's close

StatisticalReturn' Strategy Code 'COMB' Definition (Simulator)

Date	Instrument	FRL	CMA	EMA	MACD	AlgoSignal	FinalSignal
11/11/2019	GM	1	-1	1	-1	1	1
11/12/2019	GM	1	1	-1	1	0	2
11/13/2019	GM	0	1	-1	0	-1	-1

Hold = 0
Sell < 0
Buy > 1

APPENDIX E: Algorithms

E1 ARIMA

- What is ARIMA:
  - ARIMA Stands for Auto Regressive Integrated Moving Average
  - It can be broken down into its various representative parts:
    - AR: Stands for Auto-Regression, it means that it's modeling a dependent relationship between an observation and previous observations

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

    - I: Stands for Integrated. Simply calculating the differences of observations to create a stationary time series
    - MA: Stands for Moving Average. This part of the model uses a dependency between observations, and the residual error from this moving average model applied to previous observations

- There are 3 main parameters included in ARIMA:
  - P = the number of lag observations, also known as the lag order
  - d = the number of times that the observations are differenced, this is known as the degree of differencing
  - q = the size of the window calculating the moving average

- What is differencing:
  - As mentioned previously, d is known as the degree of differencing.
  - Differencing is a statistics principle that exists in time series data, where transformation is applied to the time series data in order to make it stationary. This makes it so that properties of the observations to not depend on the TIME of the observation. In other words, this eliminates factors like trend and seasonality in time series data.

$$\begin{aligned} y_t^* &= y_t' - y_{t-1}' \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned}$$

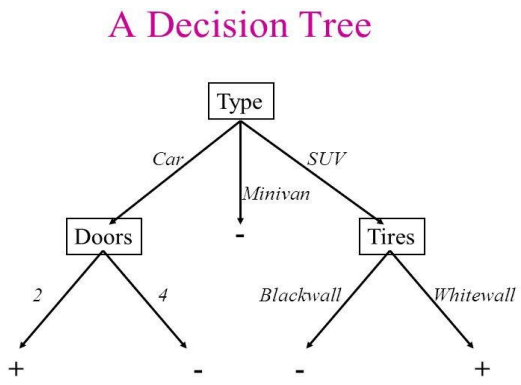
E2 Extreme Gradient Boosting XGB

What is XGBoost:

- XGBoost stands for extreme gradient boosting.
- XGBoost is a progression of the Gradient Boosting algorithm, which will be explained in just a moment.
- The engineering of this algorithm was designed for overall efficiency in computational time, and usage of memory resource
- XGBoost uses the gradient boosting decision tree algorithm
- The idea this follows is that new models are created in order to correct the errors made by already existing models
- These new models are then added together to create an informed prediction

What is the Gradient Boosting Algorithm:

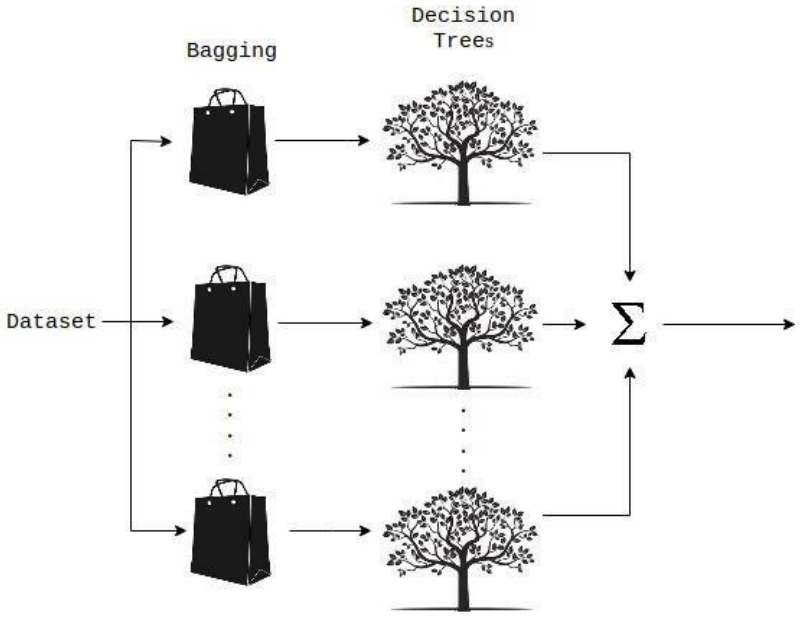
- The Gradient Boosting Algorithm Follows 3 main Steps:
  - Optimize the loss function
  - Use a weak learner to make predictions
  - Implement an additive model to add weak learners to minimize loss function
- The loss function must be differentiable
- Decision tress are used as the weak learner in the gradient boosting algorithm
- For the additive model, trees are added one by one, and the existing trees remain unmodified



E3 Random Forest

What is Random Forest Regressor:

- Random forest is an algorithm that uses two techniques: classification and regression. It performs these using a technique called Bootstrap Aggregation, also known as bagging
- Bagging: Uses a technique of training each “decision tree”, on different data sample, using a subset from the original data sample



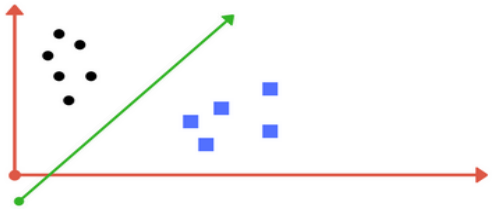
Understanding the Programming Methodology:

- First, we pass in ‘X’ and ‘Y’ variables, in the case of stock data, we are using **close** data. The Sklearn library that we see in the code uses these parameters to make determinations on how to split the data
- Next, we split our data into training and test data
- Then we specify the various parameters for our trees. In the case of our data, we are only specifying the number of estimators per bag, or the subsets of training data
- Finally, we calculate the accuracy of our model. This is done by creating a directional score and absolute mean base error based on the predicted data from the algorithm.
- Accuracy calculations details are available in the ‘Analyses and Support Files’ directory on the Git repository.

E4 Support Vector Machine

What is SVM:

- Support Vector Machine, or SVM, is an algorithm that works using the concept of separation of classes. In other words, it classifies training data based on its characteristics.

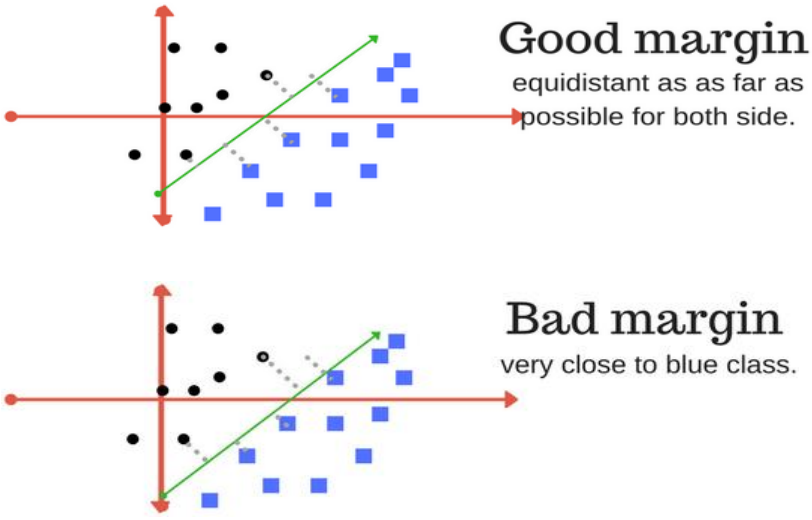


- For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:
- f(x) = B(0) + sum(ai * (x,xi))

The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.

A margin is a separation of line to the closest class points.

A good margin is one where this separation is larger for both the classes.



E5 Custom Algorithm BuySell

ARIMA + Previous group's improved predictions to generate a composite BuySell signal.

ActionSignals' table strategycode = algo (BuySell Custom) Definition

Date	Instrument	ARIMA	PrevGroupCorr	AlgoSignal
11/11/2019	GM	Up	Up	1
11/12/2019	GM	Down	Up	0
11/13/2019	GM	Down	Down	-1

Hold = 0
Sell = -1
Buy = 1
Up = next day's prediction is higher than today's close
Down = next day's prediction is lower than today's close

E6 Fibonacci Retracement Lines (FRL)

FRL is based on the key numbers identified by mathematician Leonardo Fibonacci in the 13thcentury. Fibonacci's sequence of numbers is not as important as the mathematical relationships, expressed as ratios, between the numbers in the series.
In technical analysis, a Fibonacci retracement is created by taking two extreme points (usually a major peak and trough) on a stock chart and dividing the vertical distance by the key Fibonacci ratios of 23.6%, 38.2%, 50%, 61.8%, and 100%. Once these levels are identified, horizontal lines are drawn and used to identify possible support and resistance levels.

- A Fibonacci retracement is a popular tool that traders can use to identify support and resistance levels, and place stop-loss orders or target prices
- A Fibonacci retracement is created by taking two extreme points on a stock chart and dividing the vertical distance by the key Fibonacci ratios of 23.6%, 38.2%, 50%, 61.8%, and 100%
- Fibonacci retracements suffer from the same drawbacks as other universal trading tools, so they are best used in conjunction with other indicators

Fibonacci Retracement

Before we can understand why these ratios were chosen, let's review the Fibonacci number series. The Fibonacci sequence of numbers is as follows: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, etc. Each term in this sequence is simply the sum of the two preceding terms, and the sequence continues infinitely. One of the remarkable characteristics of this numerical sequence is that each number is approximately 1.618 times greater than the preceding number. This common relationship between every number in the series is the foundation of the common ratios used in retracement studies. The key Fibonacci ratio of 61.8% is found by dividing one number in the series by the number that follows

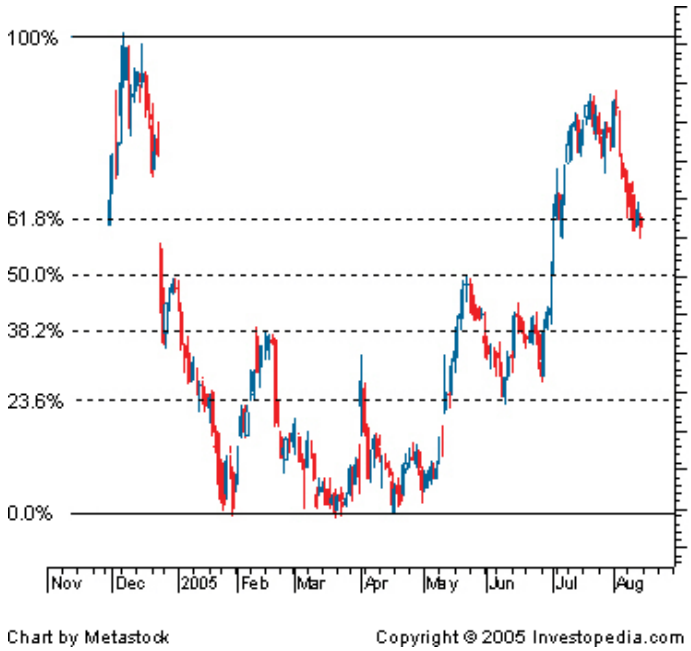
it. For example, 21 divided by 34 equals 0.6176 and 55 divided by 89 equals 0.6179. The 38.2% ratio is found by dividing one number in the series by the number that is found two places to the right. For example, 55 divided by 144 equals 0.3819. The 23.6% ratio is found by dividing one number in the series by the number that is three places to the right. For example, 8 divided by 34 equals 0.2352.

Fibonacci Retracement and Predicting Stock Prices

For reasons that are unclear, these Fibonacci ratios seem to play an important role in the stock market, just as they do in nature, and can be used to determine critical points that cause an asset's price to reverse. Fibonacci retracements are the most widely used of all the Fibonacci trading tools. This is partially due to their relative simplicity and partially due to their applicability to almost any trading instrument. They can be used to identify and confirm support and resistance levels, place stop-loss orders or target prices, and even act as a primary mechanism in a countertrend trading strategy.

Fibonacci retracement levels use horizontal lines to indicate where *possible* support and resistance levels are. Each level is associated with one of the above ratios or percentages, indicating the percentage is how much of a prior move the price has retraced. The direction of the prior trend is likely to continue once the price of the asset has retraced to one of the ratios listed above.

The following chart illustrates how a Fibonacci retracement appears. Most modern trading platforms contain a tool that automatically draws in the horizontal lines. Notice how the price changes direction as it approaches the support/resistance levels.



In addition to the ratios described above, many traders also like using the 50% level.

The 50% retracement level is not really a Fibonacci ratio, but traders often like it because of the overwhelming tendency for an asset to continue in a certain direction once it completes a 50% retracement.

https://www.investopedia.com/ask/answers/05/fibonacciretracement.asp

E7 Michael Shields Special 1 (MSF1)
MSF1 is an algorithm that takes macroeconomics variables, converts the variables to percent change, and applies it to the quarterly closing price to make the next quarter's prediction. The current variables that are being used are Gross Domestic Product (GDP), Inflation Rate(IR), Unemployment Rate (UR), Misery Index (MI), CBOE Oil Volatility Index (COVI), Financial Stress Index (FSI), and Consumer Price Index for Urban Consumption (CPIUC).

Next Price Prediction = Most Recent Quarterly Close Price + (Most Recent Quarterly Close Price * Macroeconomic Variable Percent Change)

This formula is done for each macroeconomic variable. Then each price is put into a sorted list and the median price is taken as the next prediction price.

E8 Michael Shields Special 2 (MSF2)
MSF2 is an algorithm that takes macroeconomics variables, converts the variables to percent change, and applies it to the quarterly closing price to make the next quarter's prediction. The current variables that are being used are Gross Domestic Product (GDP), Inflation Rate (IR), Unemployment Rate (UR), and Misery Index (MI). Each prediction builds off the most recent prediction.

Next Price Prediction = Most Recent Quarterly Close Price + (Close Price * (GDP * weight1 – (UR * weight2 + IR * weight3) + (MI^2))

E9 Michael Shields Special 3 (MSF3)
MSF3 is an algorithm that takes macroeconomics variables, converts the variables to percent change, and applies it to the quarterly closing price to make the next quarter's prediction. The current variables that are being used are Gross Domestic Product (GDP), CBOE Oil Volatility Index (COVI), Financial Stress Index (FSI), and Consumer Price Index for Urban Consumption (CPIUC). Each prediction builds off the most recent prediction.

Next Price Prediction = Most Recent Quarterly Close Price + (Close Price * (GDP * weight1 – (COVI * weight2 + FSI * weight3) + ($CPIUC^2$))

Appendix F: Results and Discussion

At the moment the application leverages the use of Macro Economic Variables for future forecasting. We were faced with several limitations regarding data repositories during the implementation of our functions. For one most of the Macro Economic variables only store statistics as far back as 2001, so any testing that you would like to do past that date will be severely limited. To overcome this obstacle, it is recommended that you begin searching for new data sources that will allow for the retrieval of data over longer periods of time.

We also are faced with the fact that our accuracy numbers for these Macro Economic functions have a hard time competing with the tried-and-true machine learning algorithms such as ARIMA, XGB, and Random Forest. To further expand our functions and hopefully improve our accuracy numbers it is recommended to expand the back-testing functions found in AccuracyTest.py to test data across larger timeframes to develop the weightings. This will require, as mentioned above, a new data source for Macro Economic variables that go back further in time.

Finally, the last important piece of information involves ARIMA. At the moment we have accuracy functions to test the accuracy of ARIMA, Price Prediction, Random Forest, SVM, and XGB along with the 3 Macro Economic functions MSF1, MSF2, and MSF3. The only standout result that we have noticed is with ARIMA which boasts an average trend accuracy of 92.74% along with an average percent error of .37%. This is too good to be true in our eyes, but through investigation we could not determine where we want wrong with our accuracy testing. Therefore, I believe it is important to mention that the accuracy results for the ARIMA function specifically should be read with skepticism and if possible, further investigated to discover the reason for such high accuracy numbers.