A Purposeful Walk Down Wallstreet

Exploring Advanced Data Analytics in Financial Markets



User Manual

Version 2.0

July 30, 2020

Frino Jais Sri Padmini Jayanti William Aman Minhajul Abadeen

**VERSION HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Reason** |
| 1.0 | Nabeel Asgha Michael Shields Shojib Miah  Michael Chen | 04/16/2020 |  |  | Original document submitted to Professor Seyed |
| 2.0 | Frino Jais  Sri Padmini Jayanti  Minhajul Abadeen  William Aman | 07/26/2020 | Frino Jais  Sri Padmini Jayanti  Minhajul Abadeen  William Aman | 07/26/2020 | Modified version that includes information about the new additions, changes, and updates from the previous team’s version. |

**TABLE OF CONTENTS**

[**1. Introduction** 4](bookmark://_Toc46954537#_Toc46954537)

[1.1 What You Will Learn 5](bookmark://_Toc46954538#_Toc46954538)

[1.2 Before You Begin 5](bookmark://_Toc46954539#_Toc46954539)

[1.3 What You Need to Get Started 5](bookmark://_Toc46954540#_Toc46954540)

[1.4 Help Resources 5](bookmark://_Toc46954541#_Toc46954541)

[**2. The Backend** 7](bookmark://_Toc46954542#_Toc46954542)

[2.1 The Database Schema 7](bookmark://_Toc46954543#_Toc46954543)

[The GM FinTech application’s backend is built using a MySQL database. A local instance is created with the username **root** and password **password**. 7](bookmark://_Toc46954544#_Toc46954544)

[2.2 Populating the Database 8](bookmark://_Toc46954545#_Toc46954545)

[**3. The Frontend** 8](bookmark://_Toc46954546#_Toc46954546)

[3.1 Running the Frontend 8](bookmark://_Toc46954547#_Toc46954547)

[3.2 Power BI Dashboards 9](bookmark://_Toc46954548#_Toc46954548)

[3.3 Radio Buttons 10](bookmark://_Toc46954549#_Toc46954549)

[3.4 Legends 10](bookmark://_Toc46954550#_Toc46954550)

[**Appendices** 11](bookmark://_Toc46954551#_Toc46954551)

[Appendix A: Financial Base Data 11](bookmark://_Toc46954552#_Toc46954552)

[Appendix B: Buy/Sell/Hold Strategy Formulas 12](bookmark://_Toc46954553#_Toc46954553)

[B1 CMA (Cross Moving Average) 12](bookmark://_Toc46954554#_Toc46954554)

[B2 FRL (Fibonacci Retracement Lines) 13](bookmark://_Toc46954555#_Toc46954555)

[B3 EMA (Exponential Moving Average) 14](bookmark://_Toc46954556#_Toc46954556)

[B4 MACD (Moving Average Convergence Divergence) 15](bookmark://_Toc46954557#_Toc46954557)

[B5 Algorithm Forecast 16](bookmark://_Toc46954558#_Toc46954558)

[B6 Custom Combined 17](bookmark://_Toc46954559#_Toc46954559)

[B7 Buy and Hold 18](bookmark://_Toc46954560#_Toc46954560)

[APPENDIX C: Engineered Features/Technical Indicators 19](bookmark://_Toc46954561#_Toc46954561)

[**Simple Moving Average (SMA)** 19](bookmark://_Toc46954562#_Toc46954562)

[**Exponential Moving Average (EMA)** 19](bookmark://_Toc46954563#_Toc46954563)

[**Relative Strength Index (RSI)** 19](bookmark://_Toc46954564#_Toc46954564)

[**Fibonacci Retracement Lines (FRL)** 19](bookmark://_Toc46954565#_Toc46954565)

[**Moving Average Convergence/Divergence (MACD)** 19](bookmark://_Toc46954566#_Toc46954566)

[**Bollinger Bands** 19](bookmark://_Toc46954567#_Toc46954567)

[APPENDIX D: Custom Trading Simulation Algorithm 20](bookmark://_Toc46954568#_Toc46954568)

[Signal Values 20](bookmark://_Toc46954569#_Toc46954569)

[APPENDIX E: Algorithms 21](bookmark://_Toc46954570#_Toc46954570)

[E1 Autoregressive Integrated Moving Average (ARIMA) 21](bookmark://_Toc46954571#_Toc46954571)

[E2 Extreme Gradient Boosting (XGB) 22](bookmark://_Toc46954572#_Toc46954572)

[E3 Random Forest 23](bookmark://_Toc46954573#_Toc46954573)

[E4 Support Vector Machine (SVM) 24](bookmark://_Toc46954574#_Toc46954574)

[E5 Custom Algorithm Buy Sell 25](bookmark://_Toc46954575#_Toc46954575)

[E6 Fibonacci Retracement Lines (FRL) 25](bookmark://_Toc46954576#_Toc46954576)

[E7 Michael Shields Special 1 (MSF1) 27](bookmark://_Toc46954577#_Toc46954577)

[E8 Michael Shields Special 2 (MSF2) 27](bookmark://_Toc46954578#_Toc46954578)

[E9 Michael Shields Special 3 (MSF3) 27](bookmark://_Toc46954579#_Toc46954579)

[E10 Frino Jais Function (FJF) 27](bookmark://_Toc46954580#_Toc46954580)

[E11 Aman Range Shift (ARS) 28](bookmark://_Toc46954581#_Toc46954581)

[E12 Michael Shields Function Final (MSF\_final) 29](bookmark://_Toc46954582#_Toc46954582)

[E13 Linear Regression (LR) 30](bookmark://_Toc46954583#_Toc46954583)

# **1. Introduction**

The GM Fintech Application is a program that leverages financial technologies to attempt to track and predict the positive and negative pricing movements within the stock market. In the current version of the application, thirteen different algorithms are used to forecast the close prices of ten predetermined stocks. Various strategies are implemented to perform forecasts including the manipulation of previous stock price data and the addition of several macroeconomic variables.

This is a legacy project that has been passed down from team to team as part of the Wayne State University Senior Capstone Project for undergraduate students in the field of Computer Science. Under the supervision of Dr. Seyed Ziae Mousavi, this project was completed to satisfy the requirements set by the client Joshua Feinstein, Financial Analytics Lead of Global Data, AI, & Analytics Services for General Motors.

The main goal of this application is to allow its’ users to make informed decisions when making investments for a particular stock by understanding the right time to buy, hold, or sell a stock, tracking the prices and trends that occur for any given stock over time, and visualize these concepts in a user friendly manner. This User Manual will present a walkthrough of the entire application, depicting the steps needed to run the program successfully. After understanding the information in this manual, one can expect to have a complete understanding of the different parts that make up the application and how they connect to one another to deliver the proper results.

## 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 about the following aspects of the application:

* Creating the schema
* Populating the database
* Using Power BI
* Managing the instruments
* Explanations of techniques, algorithms, and signals

## 1.2 Before You Begin

All software must be installed as described in the README document. The following is a list for your convenience:

* MySQL Workbench 8.0
* PyCharm Professional Edition 2020.1.1 (with all required packages)
* Python 3.7
* PIP 19.3.1 (PyPA recommended tool for installing Python packages)
* Microsoft Power BI Desktop
* Microsoft Windows 10 PC
* Source Code: <https://github.com/frinojais/GM-Senior-Capstone-Project-SS2020>

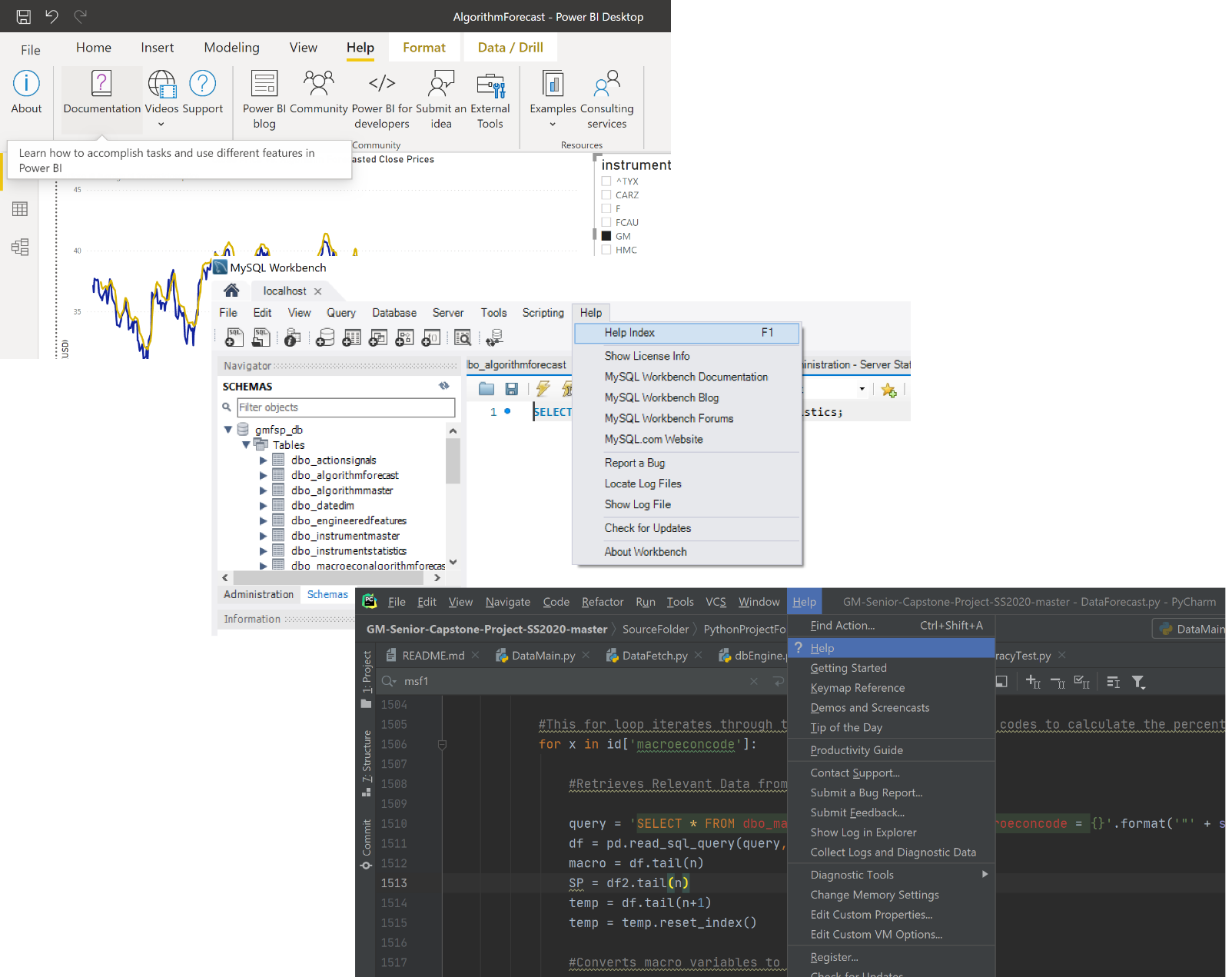
## 1.3 What You Need to Get Started

To get started with this project, you will need a Microsoft Windows computer with all the required software as listed in section 1.2 above. All software must be installed using the README file included in the source code folder.

## 1.4 Help Resources

The appendix section of this document will be useful for understanding the function and purpose of the various algorithms, signals, simulations, and other code in the project. However, the three applications used for this application all have their own “Help” sections built in.

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



You may also use any of the following resources for help:

* Power BI: <https://docs.microsoft.com/en-us/power-bi/>
* MySQL: <https://dev.mysql.com/doc/>
* Python: <https://www.python.org/doc/>
* PyCharm: https://www.jetbrains.com/help/pycharm/getting-help.html#learn

# **2. The Backend**

## 2.1 The Database Schema

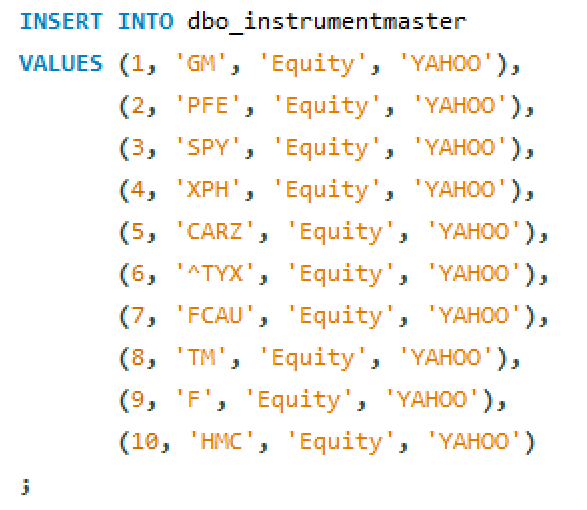
## The GM FinTech application’s backend is built using a MySQL database. A local instance is created with the username **root** and password **password**.

To generate the schema that will be used as the database for the whole project, follow the steps below:

1. Launch MySQL Workbench
2. Create a new connection with the following credentials
   1. Connection Name: localhost
   2. User Name: root
   3. Password: password
3. Create a new schema named “gmfsp\_db”
4. Open the SQL script named “CREATE\_DATABASE\_TABLES\_MYSQL.sql” from the project folder
5. Execute the script to generate the schema

If you want to modify the instruments being tracked throughout the project, follow the steps below:

1. Start MySQL Workbench and go to the script file “CREATE\_DATABASE\_TABLES\_MYSQL.sql”
   1. On line 33 of the script, you will see the following:

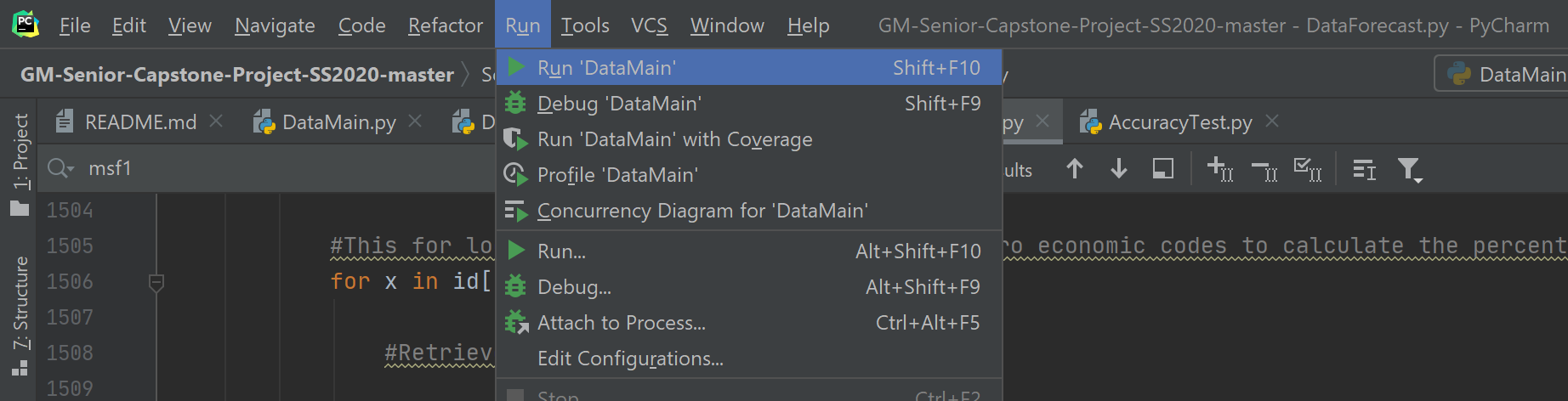


1. To insert a new instrument, add a new row with the instrument name, ID, and source.
2. To remove an instrument, just remove the row with the instrument.
3. More information on the database tables and relationships can be found in the following document:

[GM-Senior-Capstone-Project-SS2020](https://github.com/frinojais/GM-Senior-Capstone-Project-SS2020) -> SourceFolder -> SupplementalMaterial -> Database schema improvements.docx

## 2.2 Populating the Database

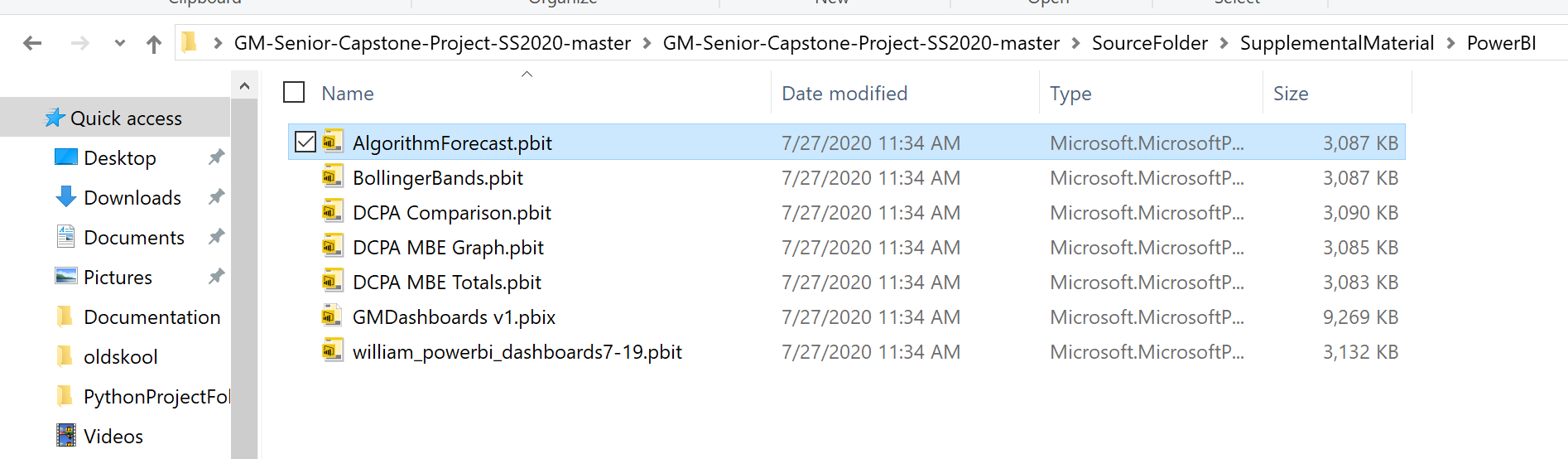
1. Open the project in PyCharm.
2. Make sure the local instance of MySQL is setup with the username **root** and password **password**.
3. Toggle all of the Boolean values at the top of DataMain.py to True to enable them.
4. Run DataMain.py. As this is run, the fetching and calculation of data is being done. This may take a long time when running the first time. (In the magnitude of hours)



# **3. The Frontend**

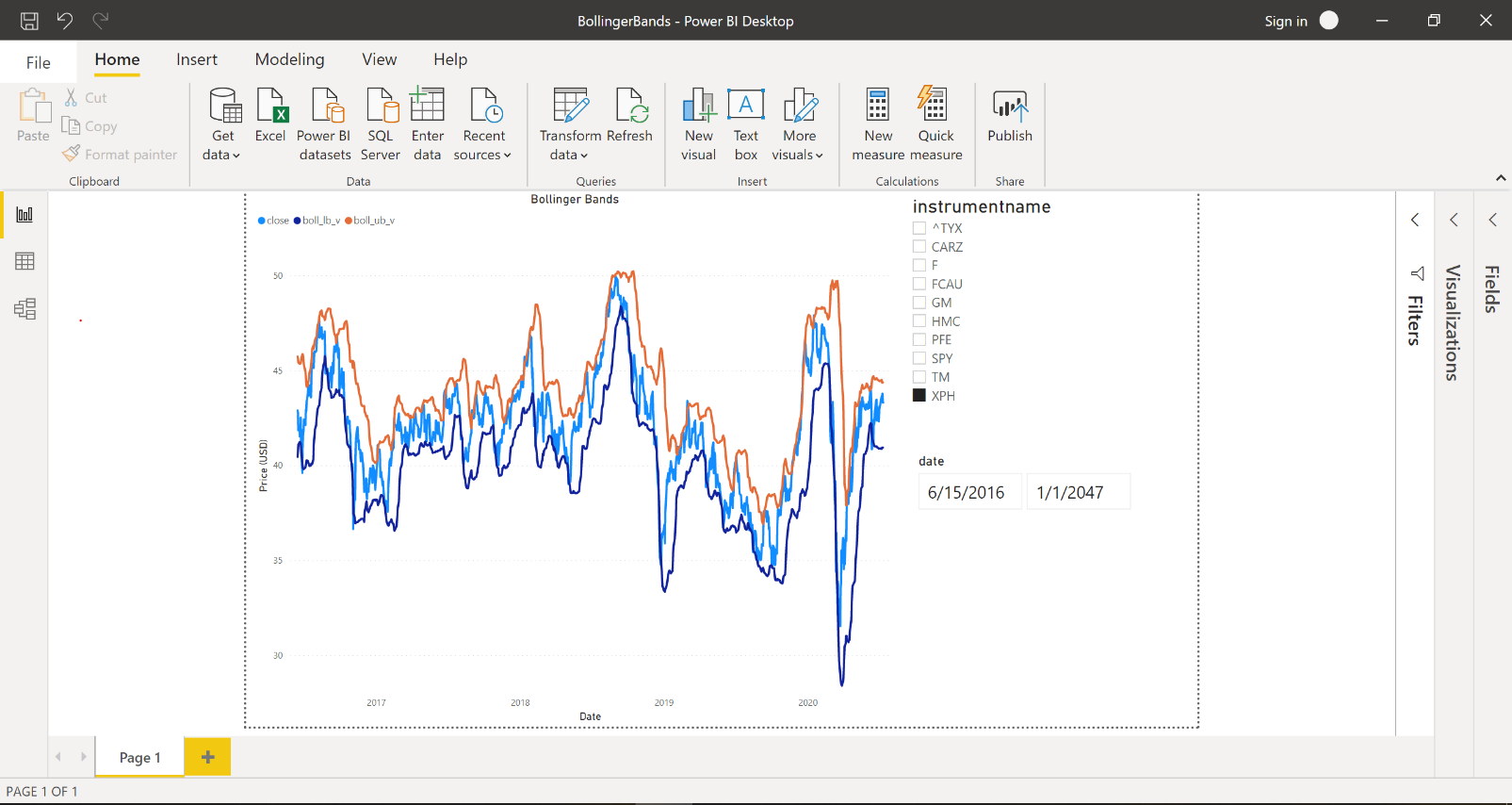
## 3.1 Running the Frontend

1. Navigate to the folder SourceFolder/SupplementalMaterial/PowerBI
2. Select from any of the given dashboards shown in the folder.
3. This will launch Power BI and the corresponding views from the given Power BI template file.

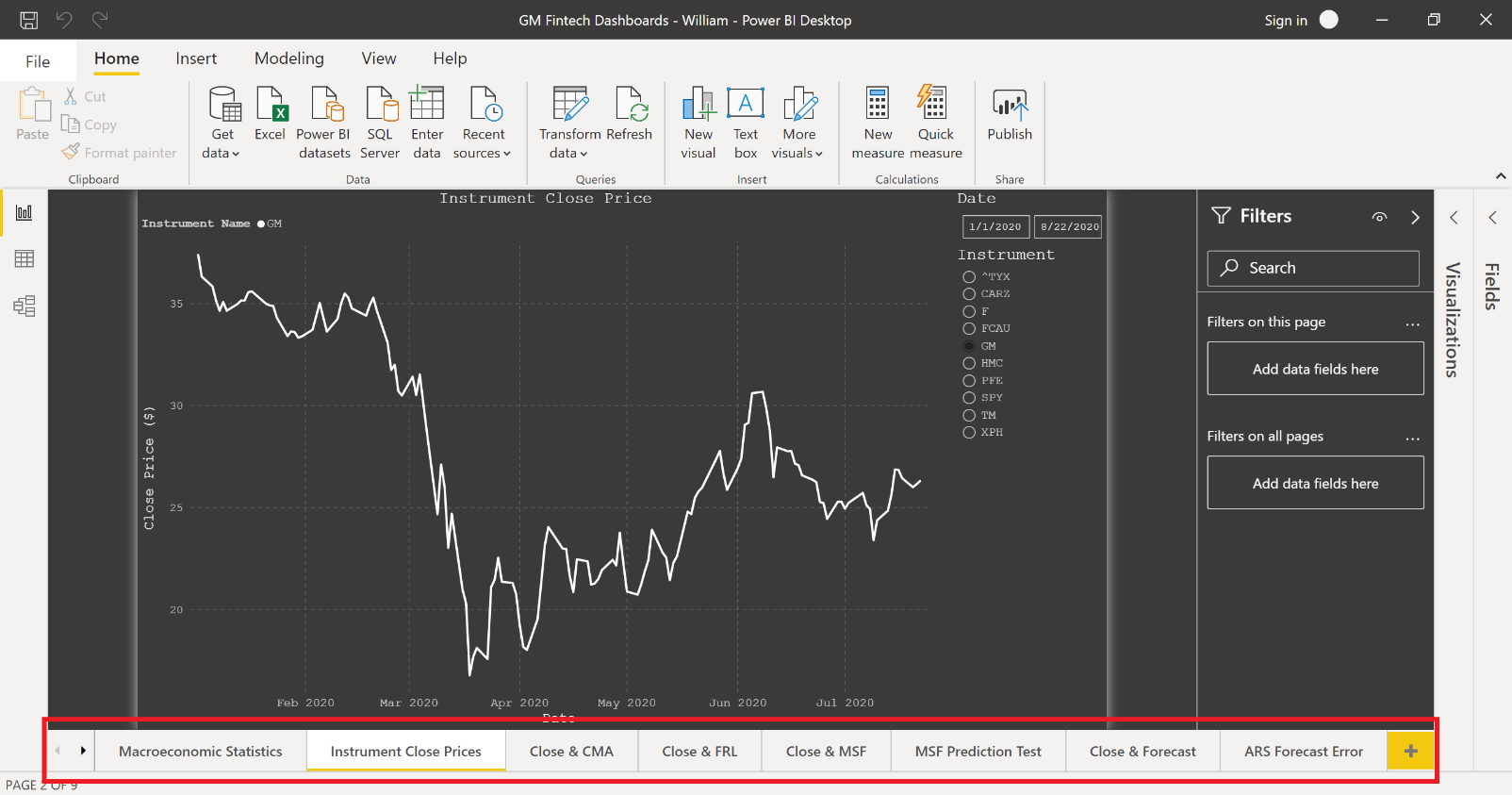


## 3.2 Power BI Dashboards

Upon opening one of the dashboards, the user will be presented with something that looks similar to this:

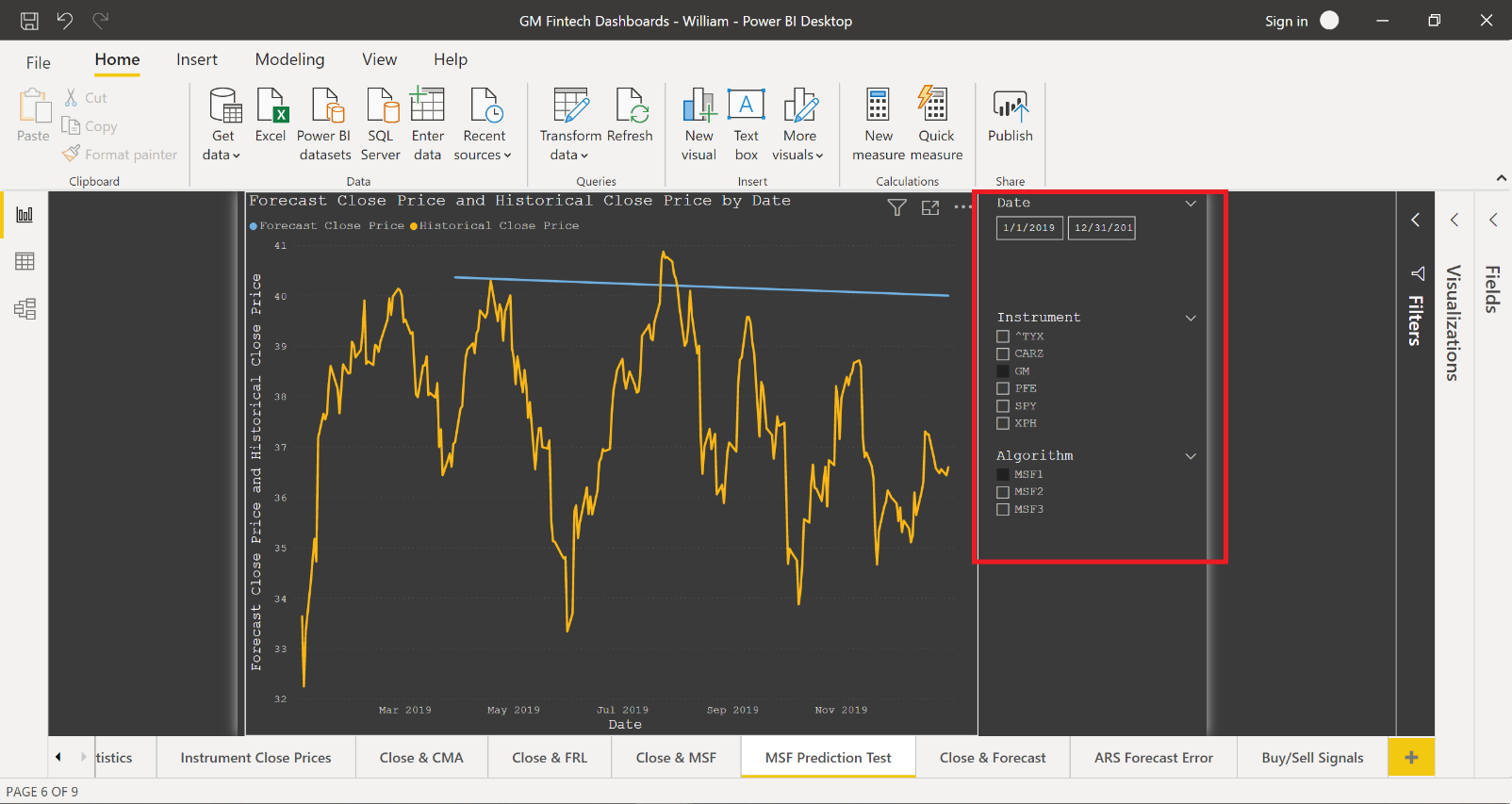


Some reports contain several dashboards, which can be toggled through by using the tabs at the bottom of the application which is shown in red below:



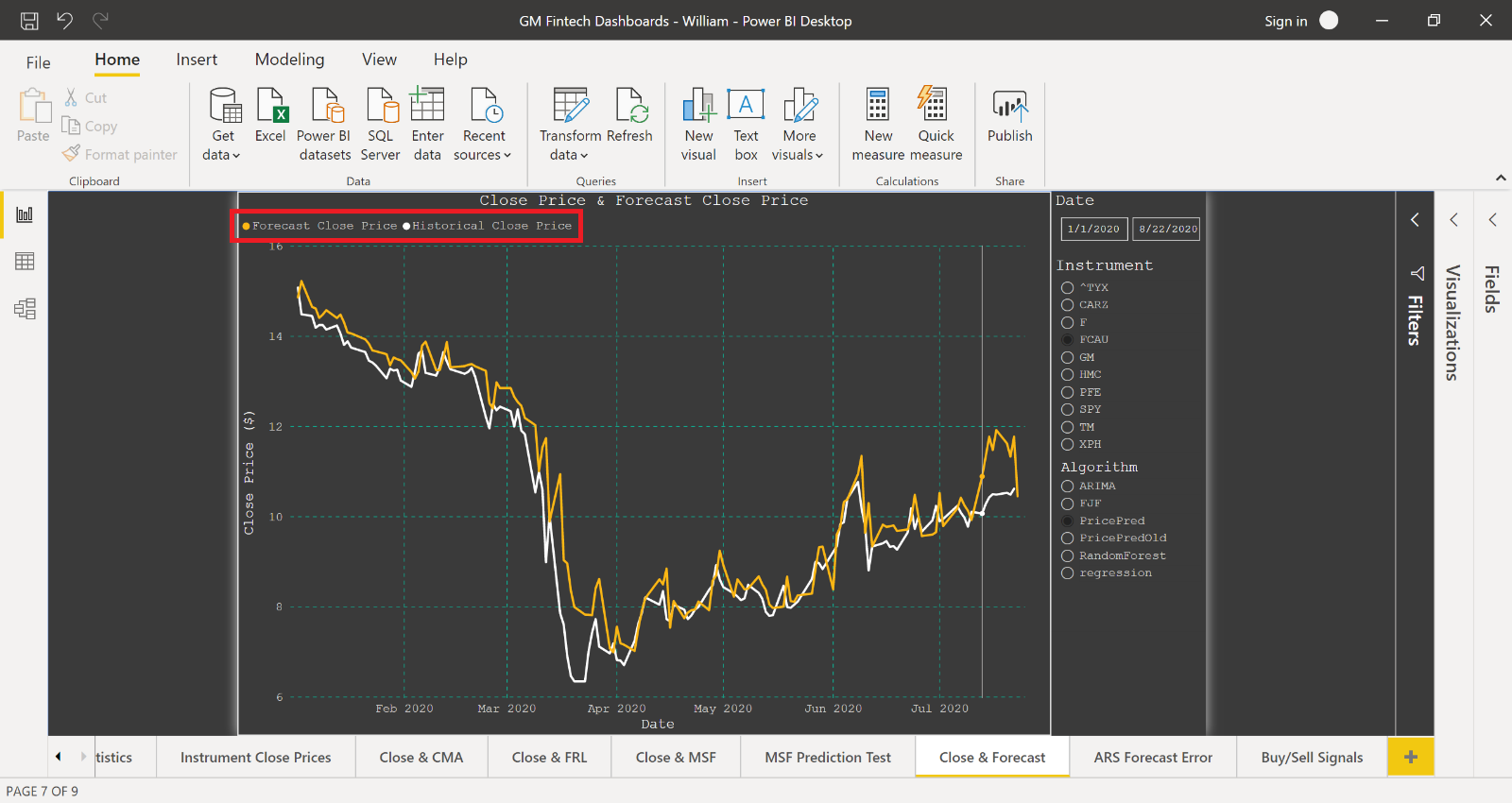
## 3.3 Radio Buttons

Any given dashboard will have radio buttons and/or date filters that the user may interact with to customize and filter the visualization of their choosing. These buttons and filters are shown in red below:



## 3.4 Legends

Any given graph in a dashboard will include a color-coded legend for each measure and line shown. The legend is typically in the same spot for each view and is shown in red below:



# **Appendices**

## Appendix A: Financial Base Data

These are the data elements extracted from the YAHOO! Finance data exchange API. This data is loaded into the **dbo\_instrumentstatistics** table in the database. Below is an explanation of what each of the fields of this table represent.

|  |  |
| --- | --- |
| **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 Standard 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: Buy/Sell/Hold 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 WHEN:

- 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 WHEN:

- In all other situation

### 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 WHEN:

- 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 WHEN:

- 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 WHEN:

- 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 WHEN:

- 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

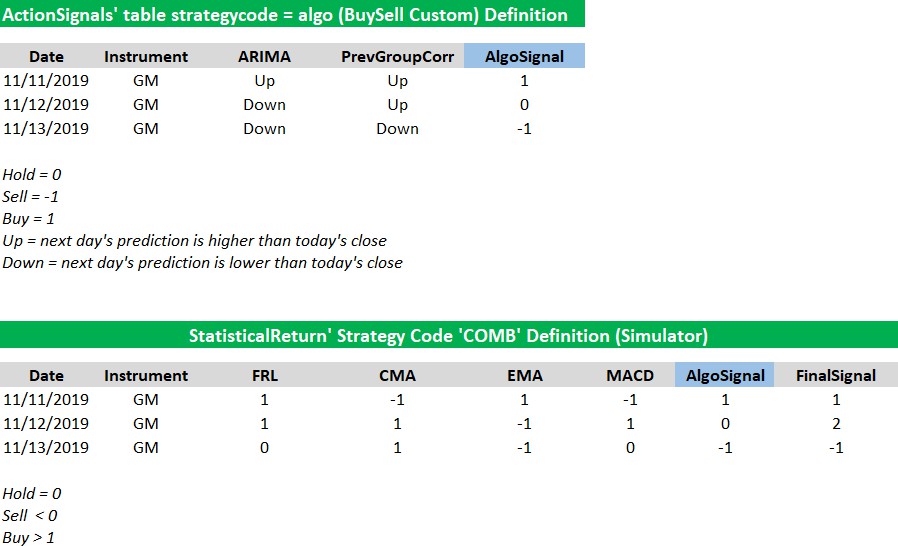
The custom trading simulation strategy uses ARIMA + one of the previous group’s improved prediction + FRL + CMA + EMA + MACD predictions for next day.

### 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



## APPENDIX E: Algorithms

### E1 Autoregressive Integrated Moving Average (ARIMA)

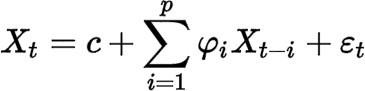
• What is ARIMA:

• ARIMA Stands for Auto Regressive Integrated Moving Average

• It can be broken down into its various representative parts:

o AR: Stands for Auto-Regression, it means that it’s modeling a dependent

relationship between an observation and previous observations



o I: Stands for Integrated. Simply calculating the differences of observations to create a stationary time series

o 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

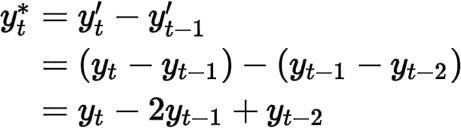
• d = the number of times that the observations are differenced, this is known as the degree of differencing

• 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 intime series data.



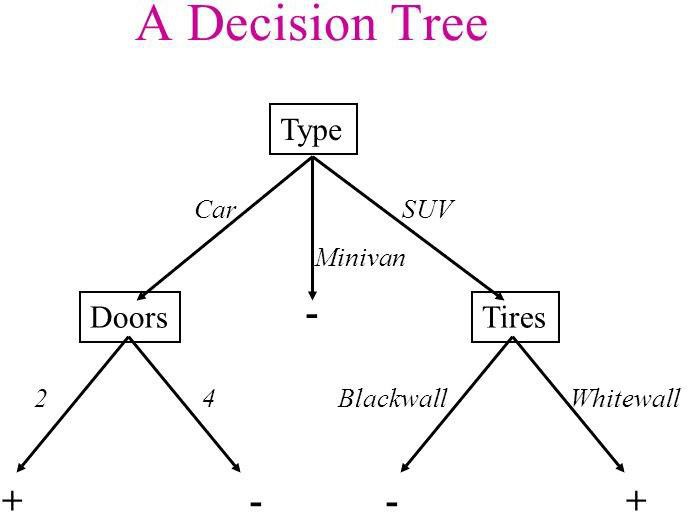
### E2 Extreme Gradient Boosting (XGB)

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:

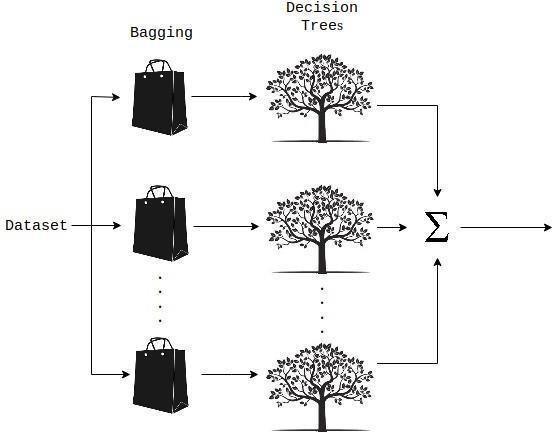
* + - The Gradient Boosting Algorithm Follows 3 main Steps:
  + Optimize the loss function
  + Use a weak learner to make predictions
  + Implement an additive model to add weak learners to minimize loss function
    - The loss function must be differentiable
    - Decision tress are used as the weak 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



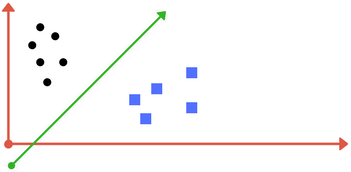
Understanding the Programming Methodology:

* + - 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 (SVM)

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.



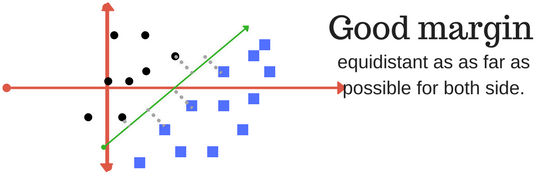
* + - For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:
    - f(x) = B(0) + sum(ai \* (x,xi))

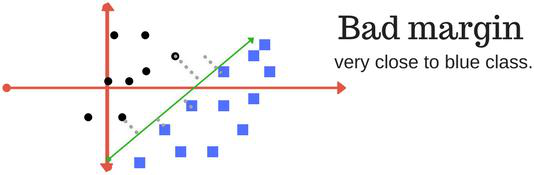
The Regularization parameter (often termed as C parameter in python’s sklearn library) tells the SVM

optimization how much you want to avoid misclassifying each training example.

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

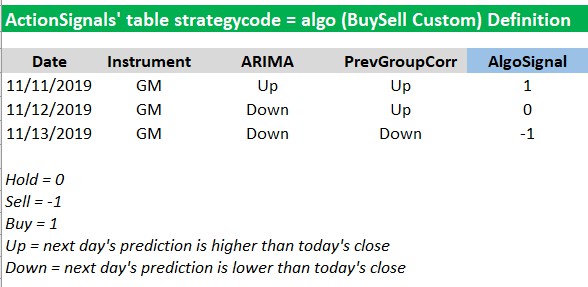
A good margin is one where this separation is larger for both the classes.





### E5 Custom Algorithm Buy Sell

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



### 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

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.

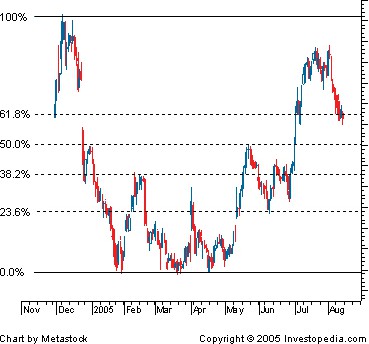
This common relationship between every number in the series is the foundation of the common ratios used in retracement studies. The key Fibonacci ratio of 61.8% is found by dividing one number in the series by the number that follows it. For example, 21 divided by 34 equals 0.6176 and 55 divided by 89 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.

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) + (MI2))

### 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) + (CPIUC2))

### E10 Frino Jais Function (FJF)

FJF, also known as the Frino Jais Function, is an algorithm that makes use of an artificial recurrent neural network called Long Short-Term Memory(LSTM) to forecast stock prices. A popular tool used in the tech industry in today’s generation is Artificial Intelligence(AI), which is a wide-ranging branch of computer science concerned with building smart machines that are capable of performing tasks that would typically require human intelligence. A subfield of AI is Machine Learning, which deals with the development of algorithms that improve automatically through experience. A subfield of Machine learning is Deep Leaning, which mimics the workings of the human brain in processing data and making decisions. Deep Learning is directly related to LSTM and its specificity in terms of the field of AI is what helps FJF deliver strong forecasting results.

The model of FJF is set up in a way where the algorithm will make its forecasts one day into the future using the past 90 days of close price values for a given stock. The “brain power” of FJF comes from its use of LSTM. By using this concept, FJF will be trained to understand the past 10 years of close price data for any given stock, understanding the historical trends and fluctuations that have occurred. After the training, the algorithm will understand how a stock has performed in the past and will make future decisions based on this knowledge. FJF currently forecasts close prices 30 days into the future, using its predictions to forecast each additional day.

The margin of error for this algorithm can vary between 1% - 18% depending of the stock on any given day. The most accurate prediction price will be the price shown for the next day and each additional prediction after than will decrease in accuracy.

A deeper understanding of how FJF works can be found under the project file path:

GM-Senior-Capstone-Project-SS2020>SourceFolder>SupplementalMaterial>

FJF>FJF LSTM Model.pdf.

### E11 Aman Range Shift (ARS)

Aman Range Shirt, or ARS for short, is an algorithm that forecasts future instrument closing prices based on its own past closing prices. This means it is an autoregressive function. The algorithm focuses on maximum close price, minimum close price, average close price, and average deviation from the close price as the primary indicators for generating a forecast. The thought process behind creating the algorithm is that, “If it has happened in the past, it will probably happen again”. The function is called “Aman Range Shift” because it generates a range of probable close prices using the average deviation, and then shifts that range either in favor of an increase or decrease depending on where the last close price is compared to the average close price. This algorithm also incorporates retracement techniques to prevent the forecasts from eventually reaching either infinity or zero, if the forecast range is very far into the future. This means, if the forecast trend starts to escape the bounds of the maximum or minimum close prices from the past, a forced retracement will be used on the next forecasted value.

There are two versions of this algorithm that can be switched between by changing the value of the “average” parameter. If this parameter is set to True, the average version will be used. If False, the random version will be used. The difference between the versions is how the forecasted value is chosen. Both versions follow the same general structure. However, when the final shifted range is calculated, the average version takes the mean in the range, and the random version takes a random value in the range. The effects of these versions causes the average version to generate more of a trend line, and the random version to generate a forecast more consistent with how the market close prices actually look. Their respective accuracies can vary, meaning neither is always more accurate than the other.

Further details can be seen by running the algorithm with the “show\_output” parameter toggled to True in the code. There is also many comments written in the function itself to help explain the process. The algorithm has a lot of customizability built into the function signature, and all of the explanations of the parameters can be found in the comment section at the top of the algorithm implementation.

Since the function is autoregressive, as the forecast range increases, so does the error. The mean absolute percentage error for this algorithm for a thirty-day forecast can range from around 1% to 15% depending upon the version being used and the volatility of the instrument being tested. This error test can be ran by toggling the “is\_test” parameter to True when using a date range that already has market closing values.

### E12 Michael Shields Function Final (MSF\_final)

MSF\_final is a statistical analysis algorithm which makes use of the close prices of the 10 financial instruments used in our application along with 5 of the macro economic variables being pulled by Quandl and Fred API’s. The idea overall in this algorithm is to make use of the quarterly macro-economic statistics given by Quandl and Fred for GDP, UR, IR, MI and COVI and append it into the monthly moving average closing price calculated as an extra step.

MSF\_final is close to MSF2 in a way that it uses the macro-economic variables with weights in the calculation, which can be turned off now after this improvement. The additional moving average infusion into the MSF concept is what is new to the algorithm. It has made the calculation more relevant and up-to-date with the closing prices. Unlike earlier, the quarterly predictions were more far off to reality. The weight function can be toggled True/False as specified in the comments written in the MSF\_final code lines.

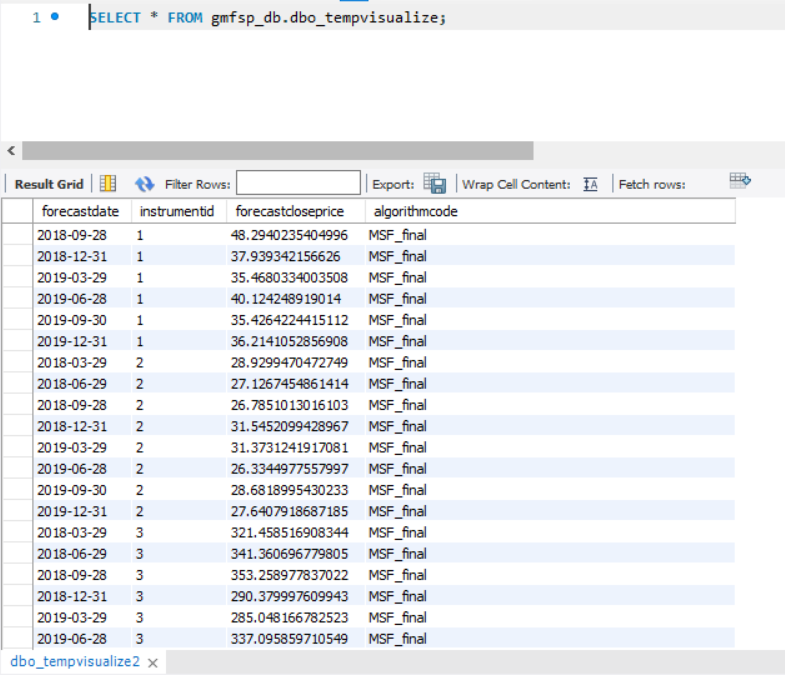
Due to the nature of the input data for the macro-economic infused algorithms, a quarterly close price prediction is made for each financial instrument separately for upto the next 2 years. These predictions are stored in “dbo\_macroeconalgorithmforecast” table in our database. For comparison purposes, a past 1 year time frame has been saved in “dbo\_tempvisualize” table to measure the accuracy of the algorithm. The same tables mentioned above have been used to plot the Power BI graphs related to MSF\_final.

There are other toggles in the “DataForecast.py” file related to outputs in “dbo\_macroeconalgorithm” that can be toggled to see some future predictions. This helps the user to just run the functionalities necessary for that particular run. The algorithm needs to be run a bunch of times if it is the initial run, due to the indices generated by several lists. So, there may be some warnings and errors given out by it initially for about 3-5 times of the run. I have also added comments that could help the user to change a few parameters that may help aid the run smoothly.

There is a relative improvement for all the 10 financial instruments compared to MSF1, which is from 50% to 70% being in the best in most of the cases. The depicted accuracy is a trend line accuracy (mentioned in code comments) for the entire quarterly term period. All the MSF functions including MSF\_final have a higher error percentage due to:

* + Quarterly term price prediction
  + Abnormal anomalies in macro-economic data

Once the initial run is successful, the following data will be shown in “dbo\_tempvisualize”:



### E13 Linear Regression (LR)

Linear Regression is a statistical modeling approach that is used to compare the relationship between an independent variable and a dependent variable. A dependent variable is a variable the depends on the value of another variable, in statistics this is denoted as “y”. An independent variable is a variable that depends on the value of other variables, in statistics this is denoted as “x”. Linear regression can be used to calculate predictions where the independent variable solely depends on the dependent variable. So, the variable we will have to predict is the dependent variable.

Linear regression can be a good forecasting algorithm in the sense linear regression consists of using the “line of best fit”. The line of best fit can also be referred to as the regression line. The line of best fit gives a good approximation of the data because the line of best fit tries to reduce the sum of squared errors automatically in a prediction. The best fitted line sits on the graph, to which the line has a linear relationship to the data set that is getting predicted.

The linear regression algorithm that is integrated in the Gm Fintech application uses such strategies to predict future forecast prices. In this case, the independent variables are the past closing values, and the dates. The dependent variable is the variable we will predict, which is the future closing price. The future closing price depends on past close values to do the forecast for a 30-day prediction. In this scenario, the line of best fit is the forecast future closing price. The closing price will build an accurate approximation for the forecasts since the sum of squared error is reduced. Since the algorithm reduces the sum of squared errors, the prediction error is between 2 - 2.8