**Stock History and Forecasting Application**

**Introduction**

This is an application built for multiple purposes related to stock prices that are available on Yahoo Finance. It has been built entirely on Python with option to follow either a command line interface or a full-fledged Web Application interface. Following are the capabilities built into the application:

* Fetch and display stock prices in candlestick style
* Display statistical measures for the requested stock history interval
* Display trends and moving average indicators for user input window
* Display MACD and RSI lines inline to the stock history graph
* Make a prediction of stock closing price for a future date (Forecasting)

**Directions for Installation and Setup**

1. Extract the zip file and locate the directory named “Stock\_App” (root dir from now on)
2. Open Anaconda prompt and navigate to the above directory
3. If the user wants to maintain a separate python environment for using the App, create a new conda environment. Otherwise, skip this step. To create a new conda environment, execute in Prompt:

**> conda create -m env\_name**

**> conda activate env\_name**

1. All the required packages are listed in a file named “requirements.txt” in root directory. Execute the following line to install all the packages at once:

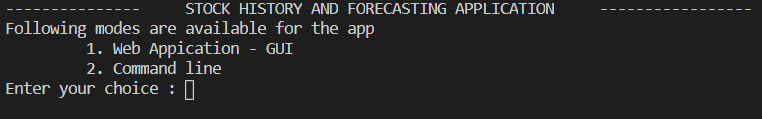
**> pip install -r requirements.txt**

**Directions for using the App command line mode**

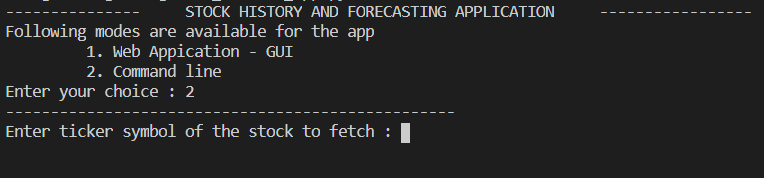
* Open terminal/command-prompt/anaconda-prompt and navigate to the root directory (“Stock\_App”)
* If you have created a new environment, make sure that it is activated.
* Execute the following command to launch the App

**> python command\_line.py**

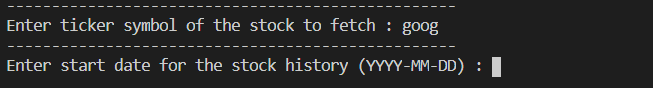
* Following header will appear with the option to select either command-line or web-app.



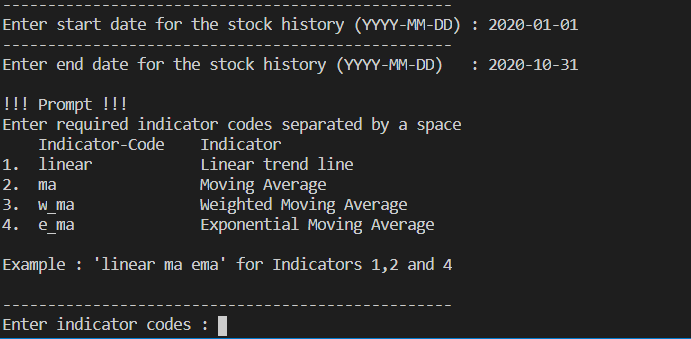
* Enter “2” as the choice. Command line app will immediately start :



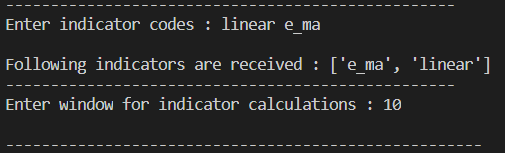
* Enter “goog” as the ticker symbol – not case sensitive



* Enter start and end dates for the stock history in the mentioned format. Enter “2020-01-01” (return) followed by “2020-10-31” (return)



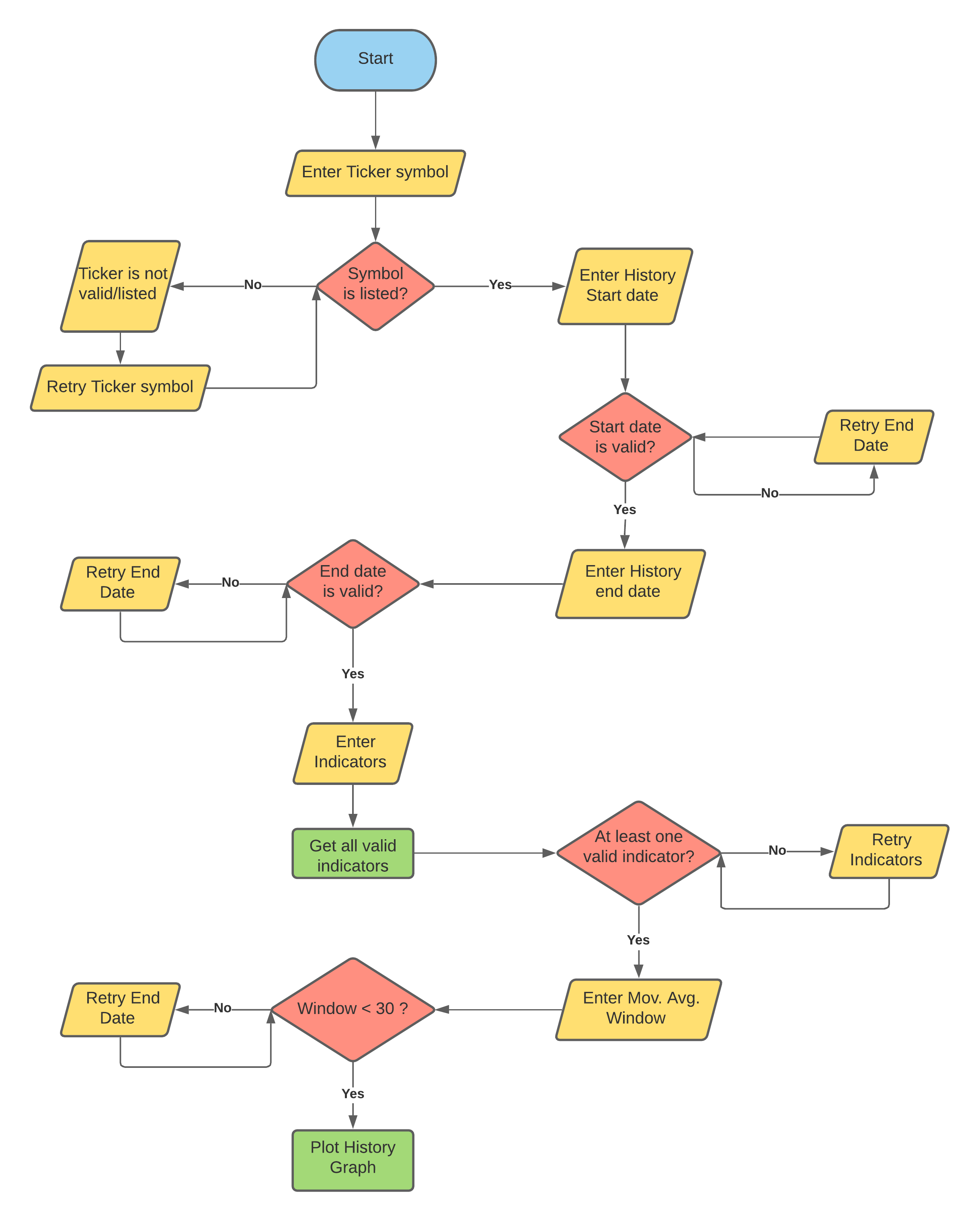
* Next, we enter the required stock price indicators to be displayed on the history plot. Here, we choose linear trend line and exponential moving average with window as 10. First enter “linear e\_ma” and return. In the next prompt, enter “10” as the window and return again.



* A tab will open in the default browser displaying a full-screen graph of the requested stock history which shows all the stock prices in candlestick mode along with all the requested indicators on top. With this example, following plot will show up:



**UML Diagram**



**Web App (DASH)**

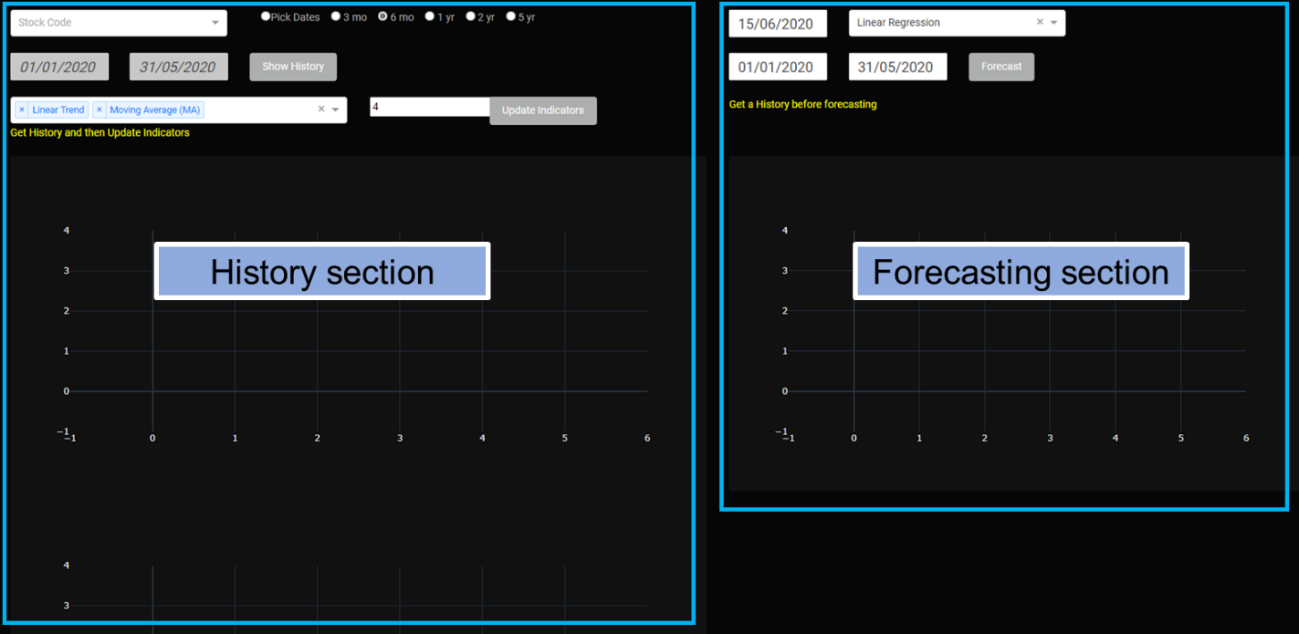
**Running the App in Web-App mode**

* Open terminal/command-prompt/anaconda-prompt and navigate to the root directory (“Stock\_App”)
* If you have created a new environment, make sure that it is activated.
* Execute the following command to launch the App

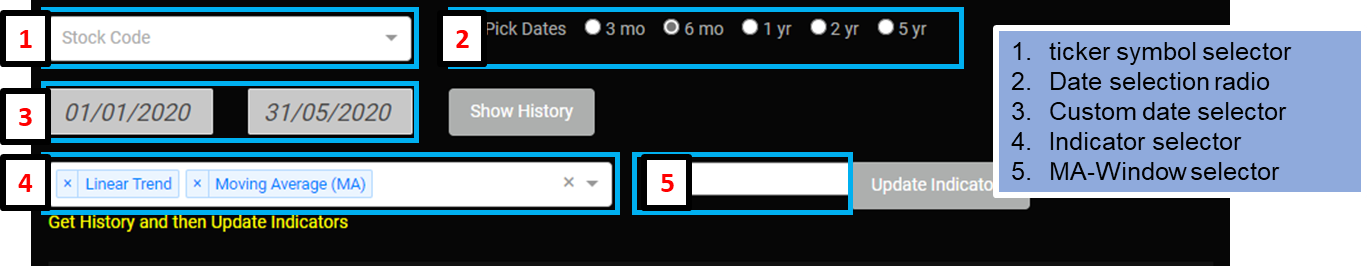
**> python dash\_app.py**

**Web-App sections**

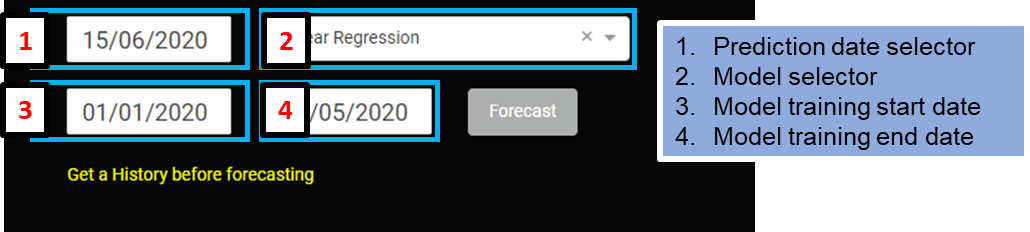
The starting screen of the App will be as shown below.



Input fields in the history section are as follows:

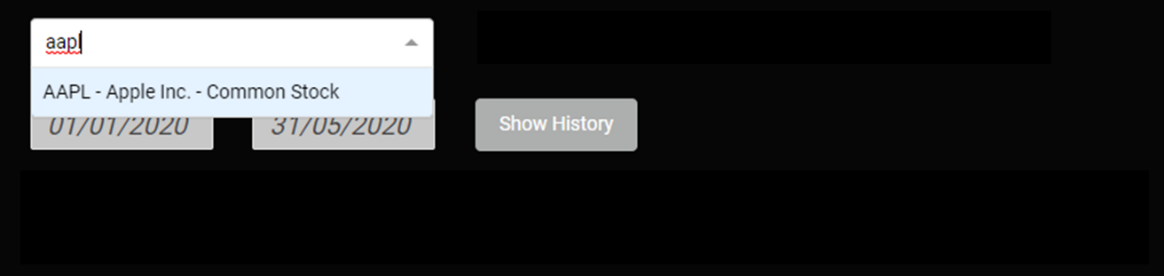


Inputs in the forecasting section are as follows:

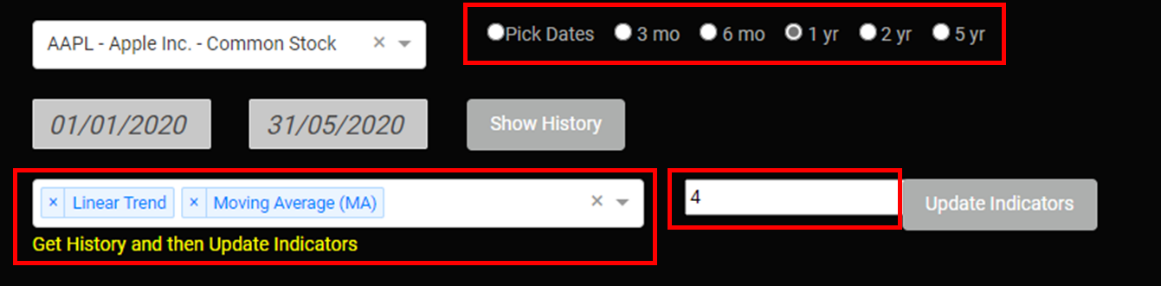


**Web App : Fetching stock history and displaying the graph**

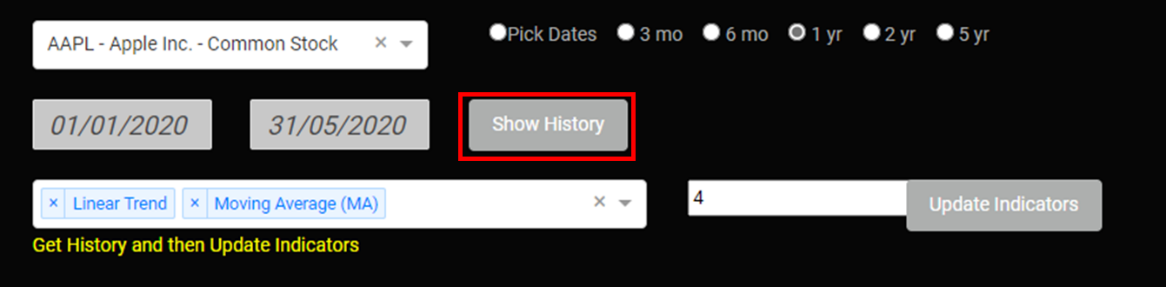
* Search and select a ticker from the ticker symbol selector



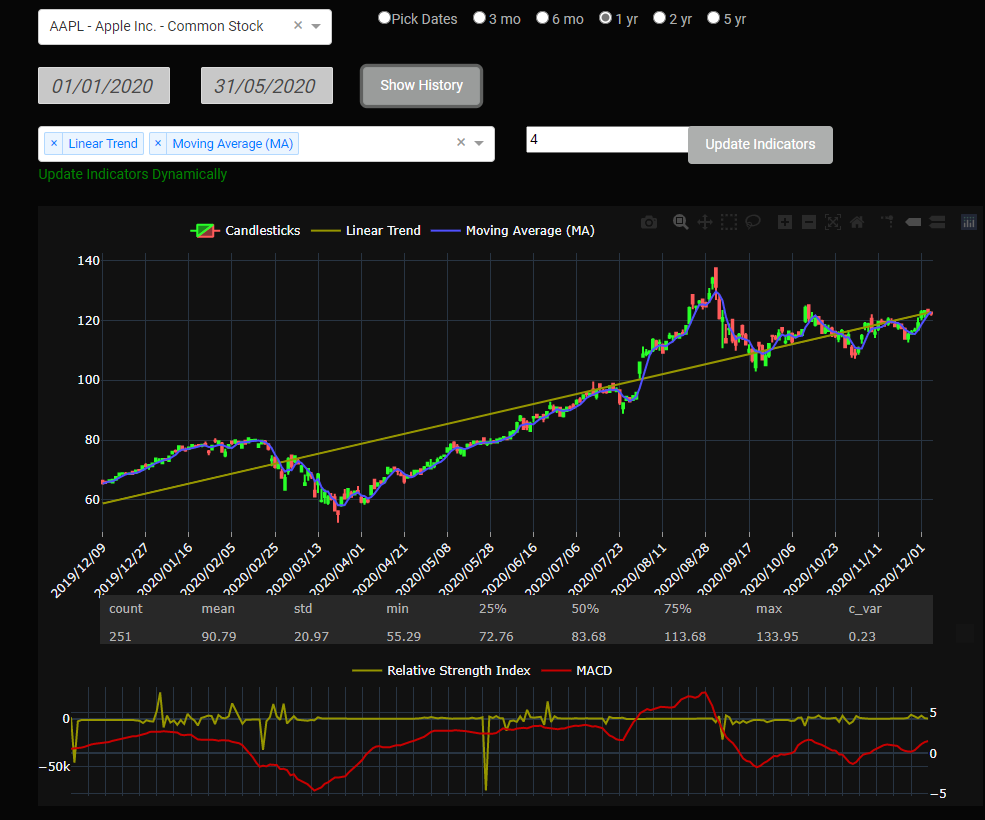
* Select a date span from the date selection radio. Also select required indicators from the dropdown box and the window for moving averages (4 is default).



* Submit the request by clicking “Show History” button

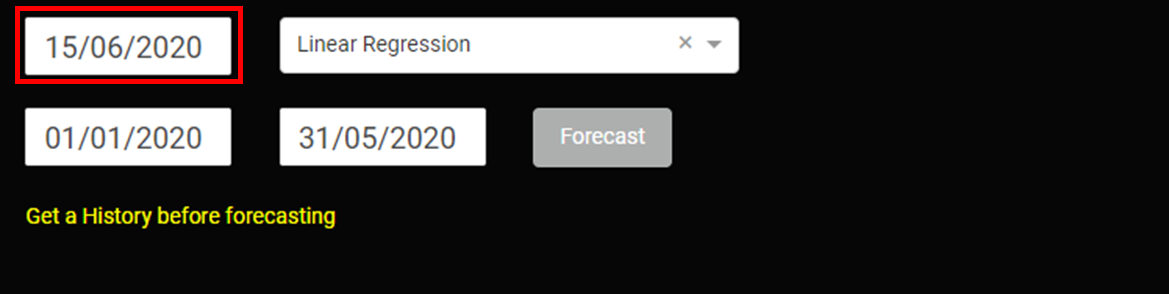


* Following plot appears for Apple Inc. stock (AAPL) with the above-mentioned inputs

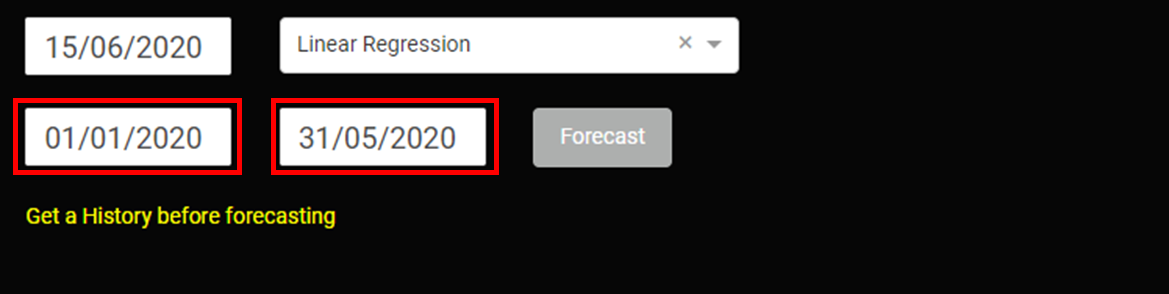


**Web App : Predicting Stock Closing Price**

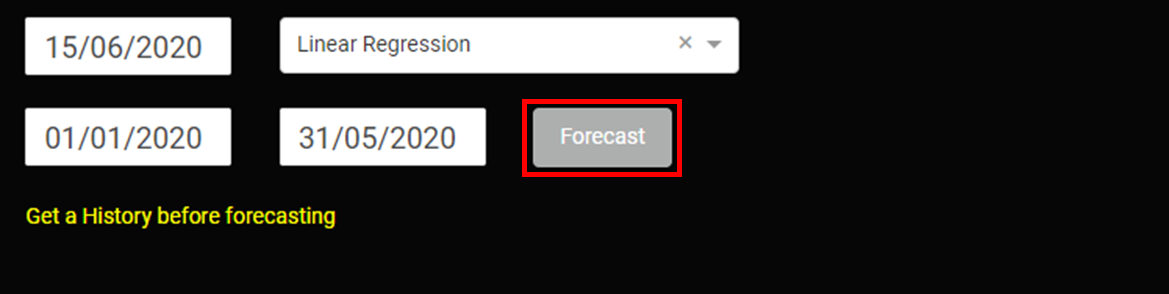
* Enter the Stock Prediction date



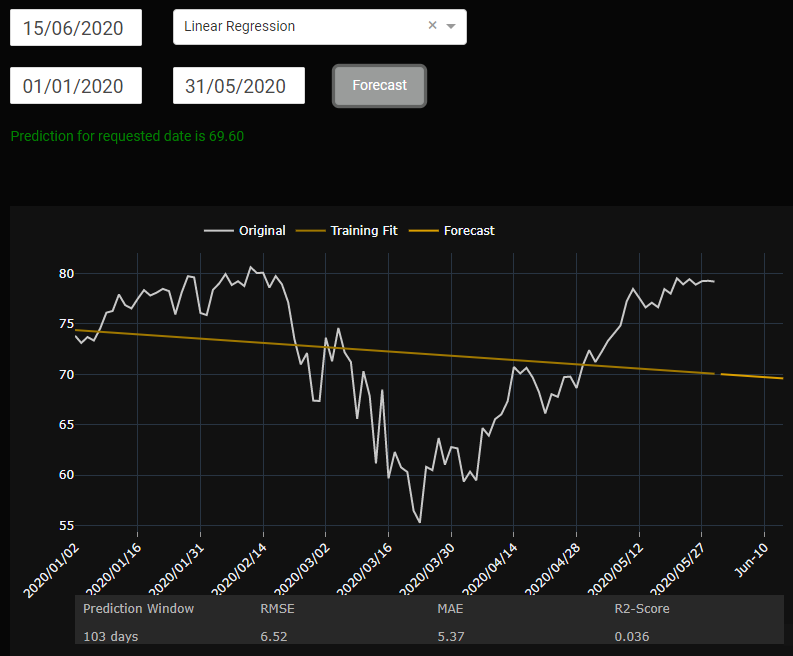
* Enter the start and end dates for training data – Training data should be before prediction date



* Select the forecasting model and press “Forecast” button



* The forecast section will now display the graph showing actual and forecast prices in training window along with the prediction price range.

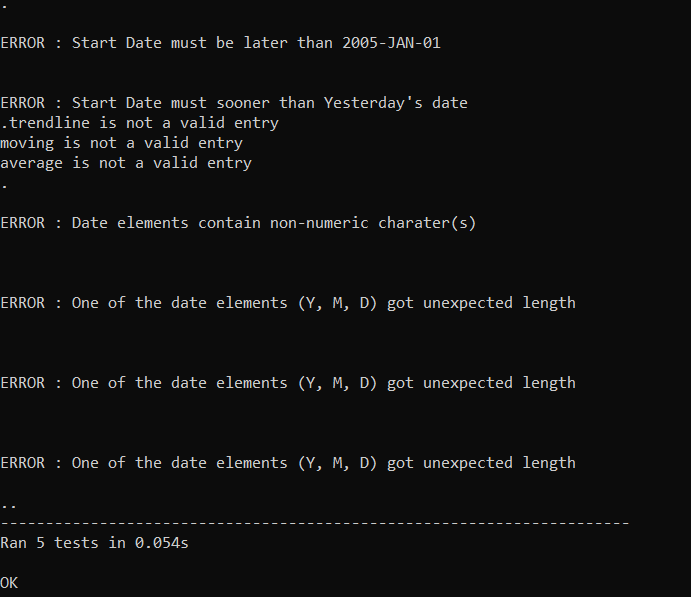


**How to perform unit testing**

* Open command-prompt/terminal and navigate to the root directory. Execute the following command\

**> python test\_cmd.py**

* If it outputs no error messages, then it is a success. Otherwise, diagnosis will be required.
* In the latest production test run, following message was received. All tests were executed with no errors.

****

The error messages are only runtime warnings when testing with failure cases.

**Updating Ticker Symbols List**

* Open terminal/command-prompt/anaconda-prompt and navigate to the root directory (“Stock\_App”)
* If you have created a new environment, make sure that it is activated.
* Execute the following command to launch the App

**> python fetch\_stocks.py**

* The symbol lists will be fetched from : [ftp.nasdaqtrader.com/SymbolDirectory/](ftp://ftp.nasdaqtrader.com/SymbolDirectory/)

**REFERENCES**

[1] https://stackoverflow.com/

[2] https://towardsdatascience.com/

[3] https://docs.python.org/

[4] https://dash.plotly.com/

**Submitted by**

**Unnikrishnan Thekke Puthumana**

**Student Number: 20200006**

**unnikrishnan.thekkeputhuma@ucdconnect.ie**