Stock Price Predictor

Vilius Radavicius

Soft 050

Student Id: 10652211

Contents

Table of Contents

[Analysis 3](#_Toc35267885)

[About the project 3](#_Toc35267886)

[Problems 3](#_Toc35267887)

[Research 4](#_Toc35267888)

[Methods and algorithms 7](#_Toc35267889)

[User interface: 8](#_Toc35267890)

[Development 8](#_Toc35267891)

[References 10](#_Toc35267892)

# Analysis

## About the project

Stock prediction software are extremely useful and convenient way to ensure that the users are making a safe investment in certain types of stocks. Although, you can make investments without the special software but the risk is significantly higher and not recommended.  In the US, about 70 percent of overall trading volume is generated through algorithmic trading.[1] Thus the use of Stock prediction software is highly recommended and way more safer than basic ordinary stock trading.

The stock prediction software will be using complicated neural network algorithms, in order to actually be able to predict the fate of a stock with the given data. This is a very crucial part of the program as if without it, the program would just start predicting pure random numbers in which is highly inaccurate and would be essentially useless to be used in the stock market business.

This project will require an Api to be used in which will allow us to fetch real-world live data from the chosen stocks. Although the rest of the code will be hard coded as it will give a better customisability and will be able to have its own custom neural net architecture (more/less layers).

## Problems

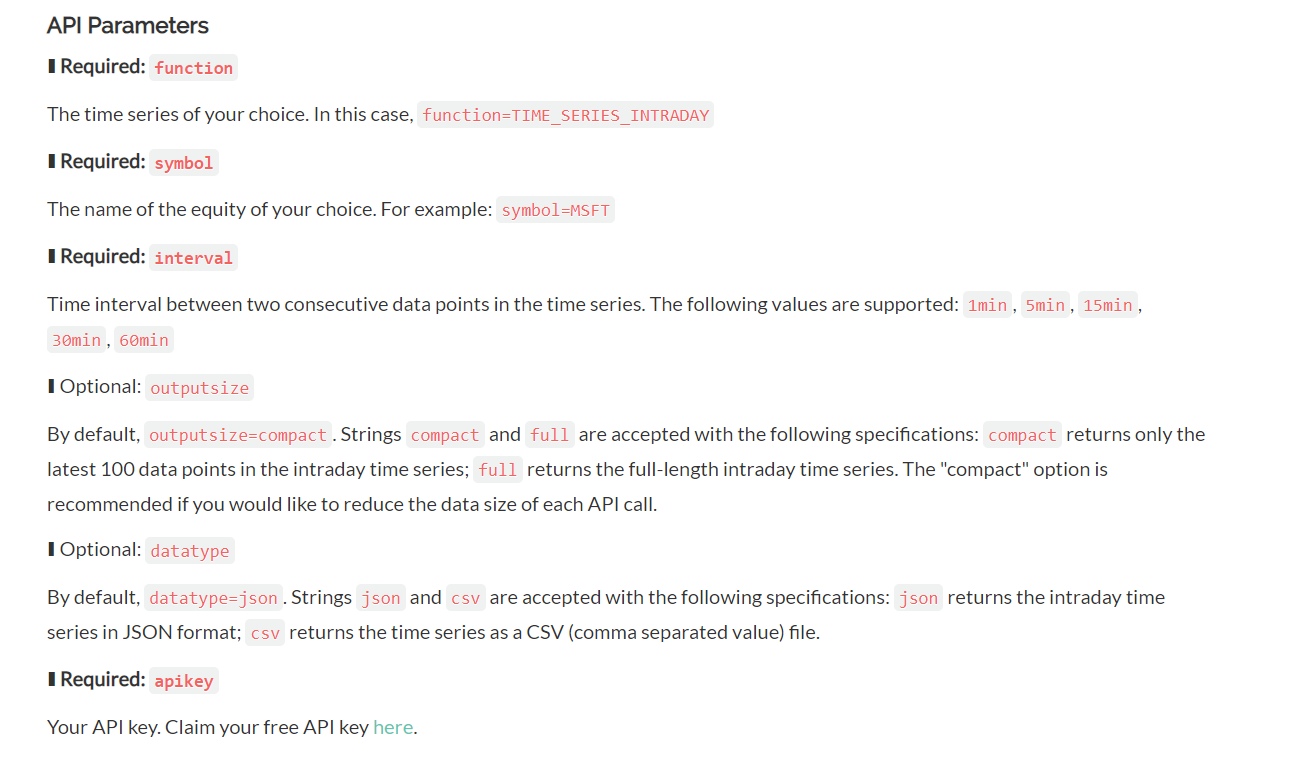
One of the problems on this project is the difficulty of making the neural net to predict the stock market values. Instead of using a simple deep neural network with just a couple of hidden layers, we will have use something called a recurrent neural network in which Is a bit more complicated and difficult to make rather than a traditional vanilla neural network.

The main disadvantage about this project is that, this stock market prediction software will not have features to actually buy/sell stocks for as this is a platform to inform the client about the predicted prices of a specific stock.

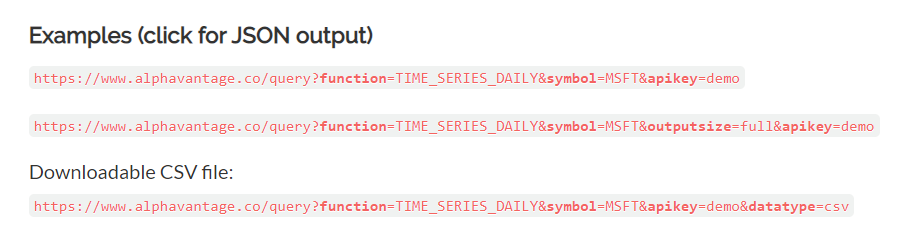
## 

## Research

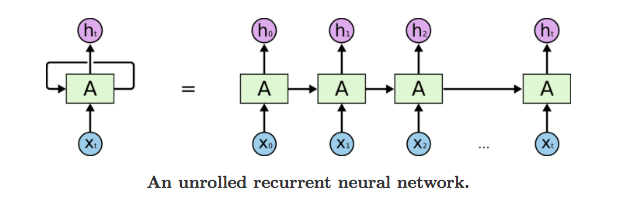
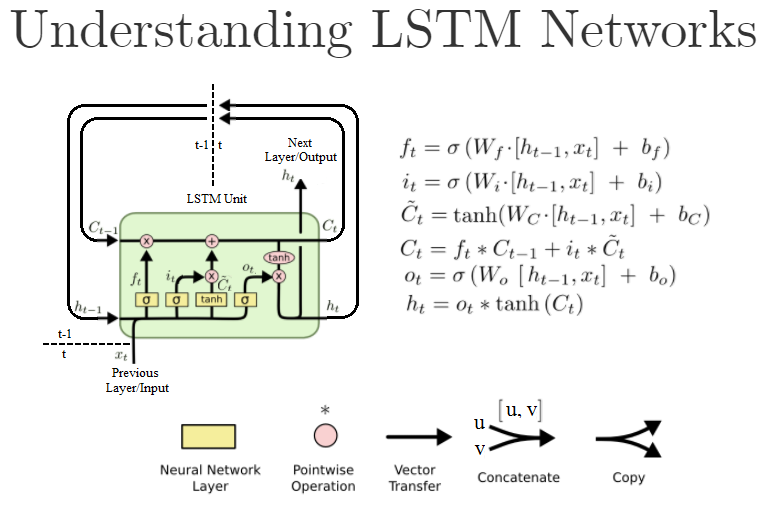
***Alpha vantage(API)***

This api is a very effective and the most popular option as it is easy to use and has really accurate data. Alpha vantage provides in both realtime data as well as some previous historical data. Alpha vantage includes a free version in which will only allow “(5 API requests per minute; 500 API requests per day)” Whilst as the premium versions go from 30 api requests per minute all the way up to 1200 and the prices are extremely high and should be used by professional profitable stock brokers.[2]

One key advantage of using Alpha vantage is that the data can be retrieved from a JSON output as well as a CSV file.

One key disadvantage is the prices for the premium version and the fact that this will be limited to 5 api requests per minute.

***Recurrent Neural Network (RNN)***

[3]As a normal vanilla recurrent neural network has a good advantage of having a memory of what it did before in which is extremely useful especially in stock market prediction as neural net needs to know if the prices is slowly descending or ascending and here’s where RNN comes in, the idea behind it is that the previous hidden layer value is being merged into the current hidden layer thus giving the neural net access to the data that was the cycle before. Problem with this is that after a couple of cycles the gradient would usually explode and thus making the entire neural net not function. Fix for this is a unit called LSTM, essentially this is a very good method of to implement the whole remembering previous cycles function as well as making sure that the gradient doesn’t explode. See link [4]

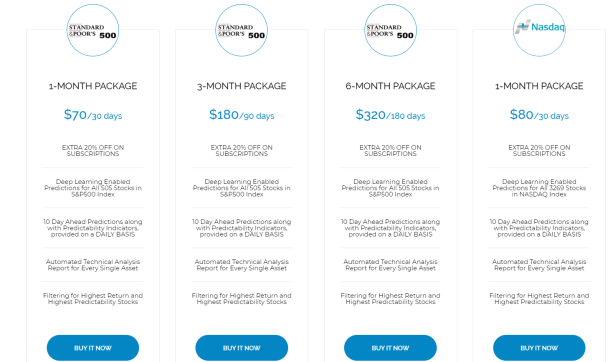
The feed forward of LSTM is fairly simple to implement, however back propagation on an LSTM layer is challenging, but luckily there’s a great example showing how to perform back-propagation on LSTM layer. See link [5]

***Already existing projects***

FinBrain Technologies:

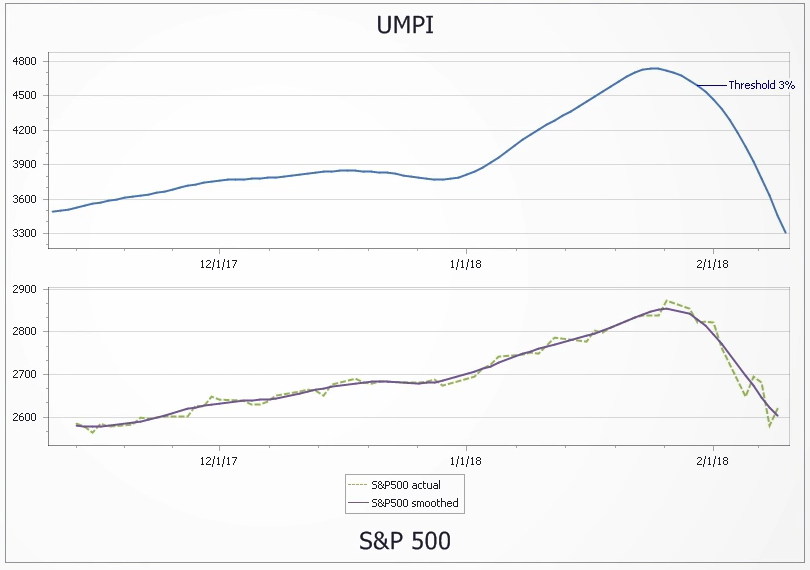
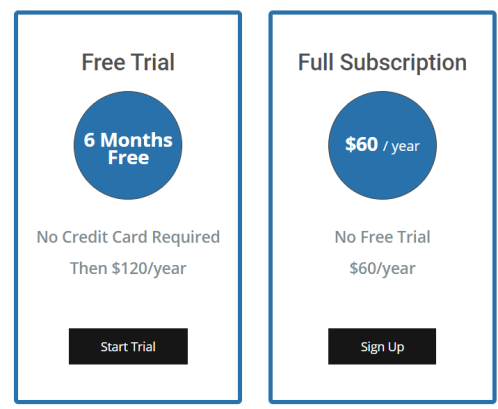
One very popular stock price prediction program is called FinBrain Technologies

As you may see, they produce a graph of historical data as well as the predicted data. This specific software offers a lot of different stocks, from crypto currencies to Forex and Nasdaq.

The main downside of this program is the over expensive price for the software. There for I will make this project completely for free other than if the person would like to purchase the better api key from alpha vantage.

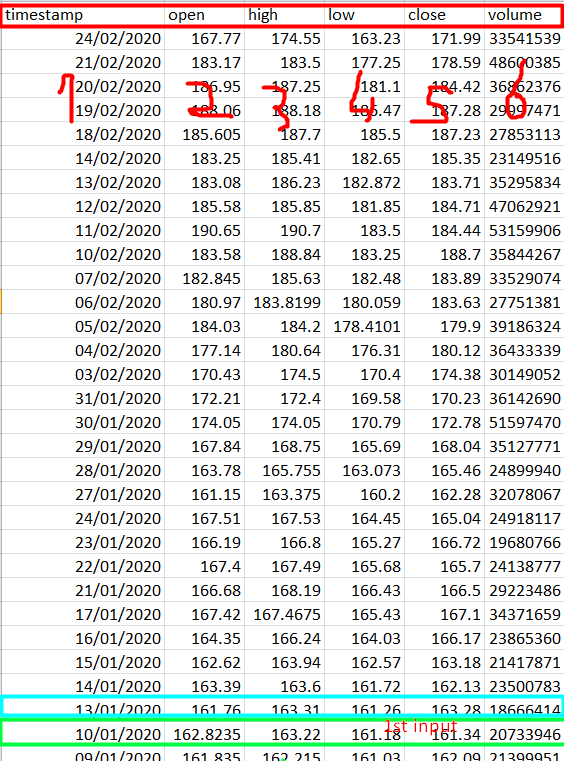
<https://finbrain.tech/#/home>

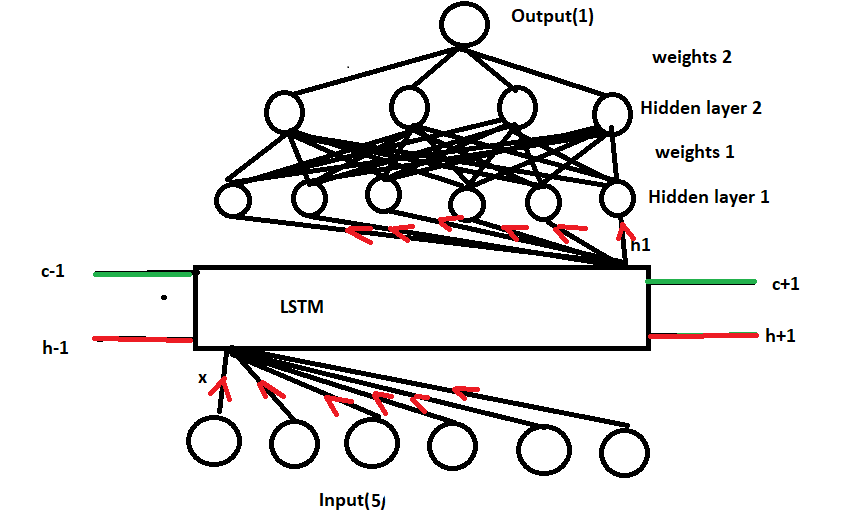
Umpindex:

This is yet another stock prediction software and as you may see it is mostly accurate but this one is giving out a free 6month membership but yet again, they will charge you a lot of money, although this is a much cheaper option than the previous one.

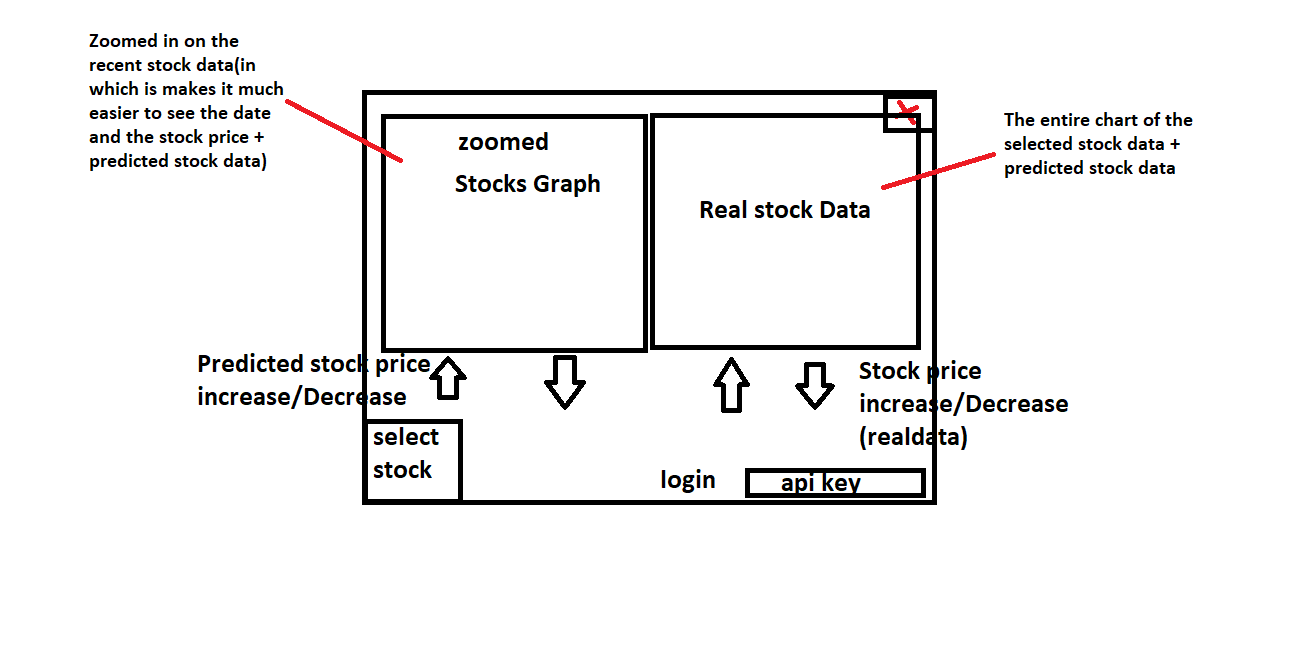
## Methods and algorithms

The entire project will be programmed in c# language and with the use of visual studio 2019 as the ide, the reason behind this is because, visual studio offers a great debugging tool for discovering problems within the code and c# has a lot of potential to be used in a large variety of projects.

As stated on page above, I will be using a specific type of neural net (rnn) and I will be recreating the lstm layer for the rnn to prevent the gradient from exploding and breaking the entire program. The project however will contain a custom model design.

The best way to figure out what the model will look like, it’s to first take a look at the inputs and the way that this AI will work is that it will be trained on 5 different inputs as you can see in the screenshot of real data from Alpha vantage, this should give the neural net enough information to be trained from. Although the original plan was to use 6 inputs but after some realisation it would complicate the project and there isn’t much use for feeding the neural net with the date. You may see in the example of how the data will be selected from the file, it will start by taking the last element (oldest stock) and feeding it through the neural net then later the pointer will move up by one and work out the cost using the newer results and then later use the newer results for feed forwarding and repeat till the pointer reaches the top in which the neural net should be updated with the pattern of the stock.

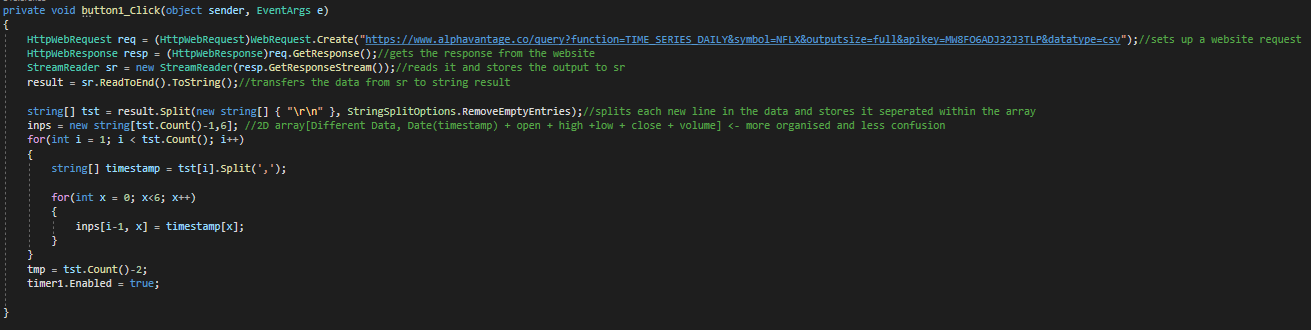
## User interface:

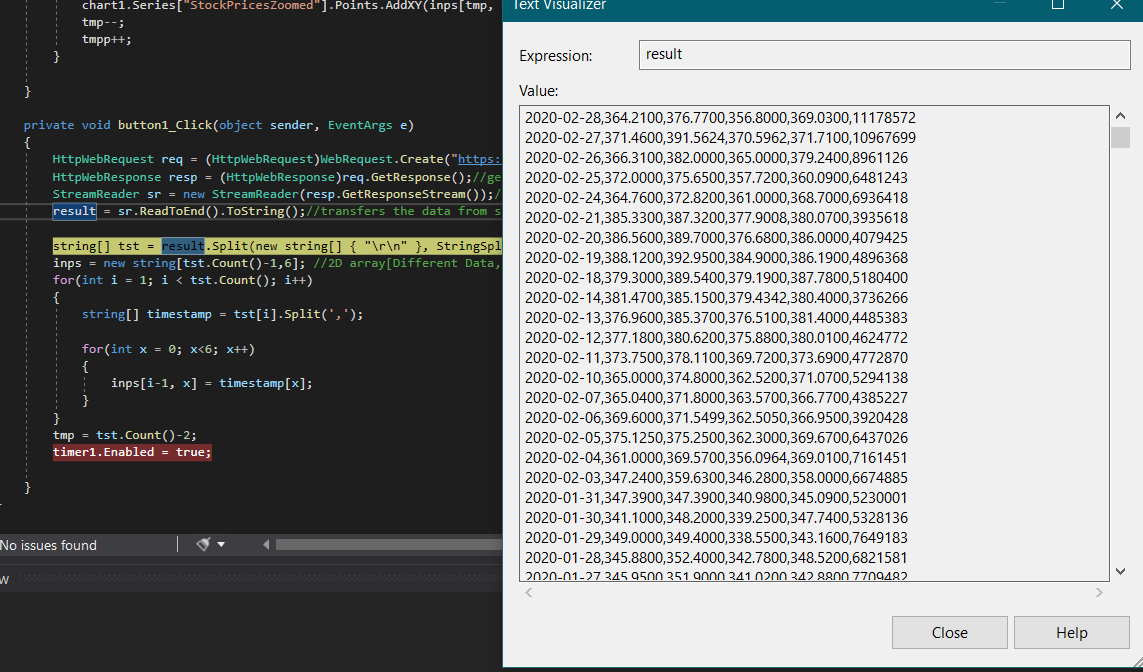


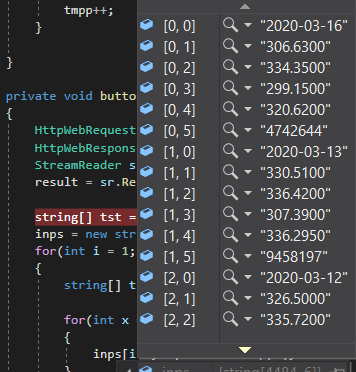
# 

# Development

The way that I wanted to obtain the data is through the use of alpha vantage api. As the user initialises the stocks by clicking a button, a http request is sent to alphavantage to retrieve a csv file and then to be stored into a long string to be processed into a manageable array.



As I did some testing, it clearly stored all the data needed into a single string, so the next step after that was to split up the information so accessing different sections of the stock data would be more convenient.



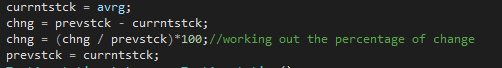
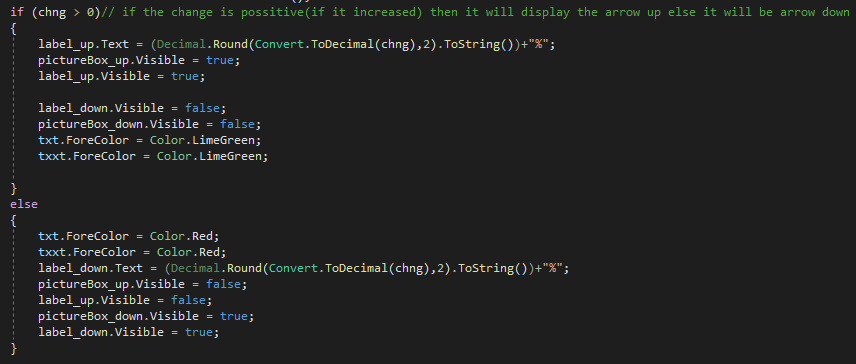
After the data is split up, it looked like this, each different information about the stocks are now able to be directly accessed in which will definitely make it much easier to use the data and to feed the neural net with.

The next stage is to make the program to be more customizable so the api input and the selection of which stock is the next function that needs to be added in.

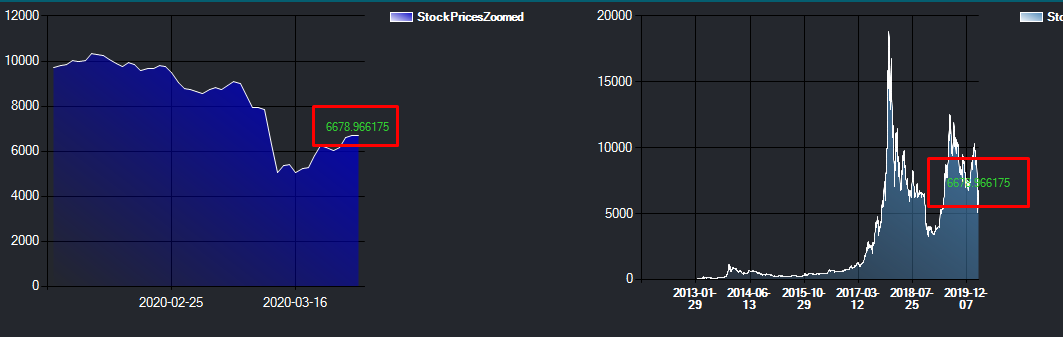
## Version 1.0

As Most of the basic features have been added in, I decided to make my program look a bit neater and added in a couple of useful features, in which they’re the following:

The main two arrows, are the percentage of change that I decided to add, this will give the user an idea of the current situation of the stock market.

To calculate it, was pretty simple, all you do is get the current stock price and you minus the previous stock price by the current one, then you divide the change by the previous stock and times it by a 100 and this should give you the percentage of change.

After that you just simply check if the change is greater than 0 and if so then it will display the green arrow and if not it will display the red arrow.

Another useful function is this label in which shows the actual value of the current sale and it stays on the same x-axis but it moves up and down on the same size as the current stock. Depending if the change is positive or negative the label will change its colour as well; this is for helping out the user and making it as comfortable and possible.

# 

# References

[1] <https://www.experfy.com/blog/the-future-of-algorithmic-trading>

[2] <https://www.alphavantage.co/premium/>

[3] <https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e>

[4] <https://stackoverflow.com/questions/44273249/in-keras-what-exactly-am-i-configuring-when-i-create-a-stateful-lstm-layer-wi>

[5] <https://towardsdatascience.com/back-to-basics-deriving-back-propagation-on-simple-rnn-lstm-feat-aidan-gomez-c7f286ba973d>