**Evaluation**

Below you can see my objectives and if I met them. The numbered point refers to the objective number in my project analysis. Most objectives were specific so I explain if I met them or not.

|  |  |  |
| --- | --- | --- |
| Objective No. | Objective | Description |
| 1 | Pull Data from a .txt file. | This objective was met exactly as you can see throughout the tests. |
| 2 | Have a format for at least the stocks open, close, high, low, and volume of chosen stocks from the past year. | This objective was met exactly as you can see in the pseudocode for the algorithms, where the data is stored. |
| 3 | User chooses the maximum amount of stocks that they can have in one instance is. | This objective was met exactly as you can see throughout the tests. |
| 4 | Be able to display all the data of a stock that was pulled. | This objective was met exactly as you can see throughout the tests. |
| 5 | To be able to display specific lines of the data you want. | This objective was met exactly as you can see throughout the tests. |
| 6 | Find direct correlations coefficient. | This objective was met exactly as you can see throughout the tests. |
| 7 | To predict whether the stock is going to go up or down as a binary value. | This objective was met exactly as you can see throughout the tests. |
| 8 | An easy to use UI. | This objective was met exactly as you can see throughout the tests, this is due to fact that the input boxes are intuitively placed by labels or buttons so the user knows what they’re for and memos have appropriate titles. |
| 9 | Be able to delete stocks that are no longer needed. | This objective was met exactly as you can see throughout the tests. |
| 10 | Be able to restart the program on demand from the user, without having to exit the program and start again. | This objective was met exactly as you can see throughout the tests. |
| 11 | Draw a graph of a pulled stock. | This objective was met exactly as you can see throughout the tests. |
| 12 | Implement a minimum of 2 trading algorithms for the solution of my pic, 1 algorithm has to be the correlation algorithm and another algorithm, of my choice. | This objective was met exactly as you can see throughout the tests. Which was the correlation algorithm and the simple moving day average. |
| 13 | Make a suggested portfolio. | This objective was met exactly as you can see throughout the tests, which is based on the users desired size. |
| 14 | Pull Data from a .txt file without changing its content. | This objective was met exactly as you can see throughout the tests, an extension of objective 1. |
| 15 | To make the computer automatically calculate all the stocks pulled correlation. | This objective was met exactly as you can see throughout the tests. |
| 16 | To sort the correlated stock from highest positive to highest negative (1🡪-1). | This objective was met exactly as you can see throughout the tests. |
| 17 | To allow the user to choose either the start date or the end date for the correlation or none, and allow the computer to calculate the correlation with the dates given, or handle the error if it can’t find the date. | This objective was met exactly as you can see throughout the tests. |
| 18 | Find direct correlation coefficients for an array of stocks that don’t start or end at the same date, with no date specified for the correlation (Find a common start date and end date between the stocks) | This objective was met exactly as you can see throughout the tests. |
| 19 | To be able to pull data from an API on the internet on demand, from the user, by use of the stocks Ticker symbol. | This objective was met exactly as you can see throughout the tests. |
| 20 | To be able to pull stock data from an API of different lengths (E.g. 3 months 🡪 2 years). | This objective was met exactly as you can see throughout the tests, the user can pull stocks of time period 5 days, 1 months, 3 months, 6 months, 1 year, 2 years and 5 years. |
| 21 | Introduce some AI machine learning so the predictions get better over time. | This was not added to the program as they’re unfortunately was not enough time to effectively code a complex machine learning algorithm. Therefore this extension was not met, which the client said they expected. |
| 22 | To predict stock prices for the future days, weeks and months. | This was not added to the program as they’re unfortunately was not enough time to effectively code a complex stock price prediction algorithm. Therefore this extension was not met, which the client said they expected. |
| 23 | Actually execute purchases and sales on the stock market with real or fake money. | This was not added to the program as they’re unfortunately was not enough time to effectively get and learn the resources that would allow a Delphi program to connect to the stock market. This would have taken a considerable amount of time and hindered the completion of the other objectives. Therefore this extension was not met, which the client said they expected. |
| 24 | Error handle for the majority of inputs. | This objective was met exactly as you can see throughout the tests. |

**Overall Effectiveness:**

I believe that overall my code was very effective and met the clients objectives excepts the difficult ones that were consequently put in the extension section. I believe that it is efficient solution this is seen by the use of a merge sort algorithm to sort the coefficient array which is an array of records. The program can also pull stocks from an API which means that the user doesn’t have to have the file downloaded this is important as it majorly affects for the better the ease of use of the program. The program can also show the buy periods of a stock which takes advantage of a computers quick processing power to quickly calculate this. Additionally the program takes advantage the speed of a computer by calculating all the direct correlation coefficients of every combination of stock and sorting them, for the user to see instantly. Also the stock program cleverly finds the overlapping dates over where to take the correlation coefficient from, which saves the user time, and calculation.

Overall the program was mainly aimed to take advantage of how the computer can process big data a lot quicker than humans. This was reflected in the objectives which were vastly met, again except some of the extension objectives. This I believed was fully accomplished by the program that I have created, now allowing the user/client to focus on different sections when the client decides whether or not to buy or sell stock. You will see in the ‘Client Feedback’ that the client agrees with this statement as well.

Improvements that I would make to the solution of this program are few, however there are some. These include, on the display tab to give the user the option to display specific values, such as all the open values of the stock, or the close and volume values. Rather than all of the values in the line or in the stock. This is due to the user being normally interested in a couple values of the stock but not all, and additionally it makes the display cluttered.

Due to time constrictions at the end the code wasn’t broken off into as many functions as what I would normally have done. This doesn’t affect the output of the code, however it affects troubleshooting and developing clean code, which would be easily debugged.

The last thing I would change if time allowed is to add a feature that combines the simple moving day average and the correlation algorithm, so that the computer can get more data about the stock which would allow the computer to make a more informed choice about making portfolio, that is recommended.

**User Feedback:**

I first of all would like to say it has been wonderful working with Michael Borinuoluwa for the duration of this project.

After using the code that he has created, I can return some feedback to him, which can potentially make the code even better. However I would gladly say that I am more than happy with the outcome of this project. It has achieved everything that I wanted it to achieve, although not all the extension objectives were met, these were very ambitious and challenging to do under normal conditions; let alone the time constrictions. This is not even including the additional write up that had to be done, as well as balancing studying his other subjects at the same time. So for the objectives not met I would say I didn’t expect him to complete as many objectives as he did.

One thing that I was not keen on as the client was changing the input box to an outputted value from the program. This made it hard to track what you had entered and allowed the risk of the user running into a (handled) error. Additionally the outputs should be read only values where as the input isn’t so there was a bit of a discrepancy in that sense.

I liked the tab like structure of the program using the menu. This allowed one to multi-task very well on it, and was an intuitive way to get more out of one display, without making the display messy and over busy. The sections were also chosen very well, allowing me to work through the programs menu as I was using the program.

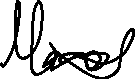
It would have been more ideal for me to be able to choose what parts of the stock value I wanted to see in the display. As the display becomes busy when too many lines or a whole stock is searched. Thus making it hard to manually look for patterns in the display. Additionally I would have like to display a line of data by date rather than line number as, this forced me to guess the index of the date that I desired, this would preferably be changed to being like the analysis tab where you can choose start date and end date rather than start and end index.

I liked how well everything was labelled and I knew which inputs did what. This made it easy to navigate the program which is important to me as time shouldn’t be used to decipher which parts of the program does what.

It was very useful to pull data from an API using its ticker symbol and its time frame. This was an extension objective however proved to be very useful as the mainly users and myself will predominantly use stock data pulled from the API, as is much easier than downloading the .txt.

I liked how well the user input was error handled. This is useful as accidental inputs occur and it is useful not to get some computer error, and the program not crashing is also not useful. I tried inputting invalid arguments into the inputs however the errors were very well handled, which marks the sign of good code.

The program is very good and even though there are a few tweaks that would make the program even better these are all minor suggestions. The whole programs is much more than useable, satisfies my requirements and therefore I would say this program was a huge success.



Marçin