**Project Analysis**

**The Background**

The project comes from the popular use of making financial gain from use of the stock market. The stock market is all the share prices in all the different financial sectors, such as consumable and mining, in all the different financial markets, such as FTSE100, SMALLCAP50 and S&P100. These are the different markets for different company sizes in different countries, every public company will have a share price, which is available to the public, and will be indexed in a financial market. The stock market also accommodates different financial markets that aren’t public companies, these include: commodities (raw materials) e.g. oil and gold; cryptocurrencies (digital currencies) e.g. Bitcoin and Ethereum; forex (foreign exchange) e.g. GBP/EUR which simply is to do with the exchange rate between the currencies; an indices this is a general value for a multitude of stocks included in the indices e.g. FTSE100.

Buying a share can be seen as a physical or more abstract depending mainly on the quantity of shares bought. You can visualise this as buying a share is buying ownership of a company. A share value is directly related to a percentage of owning the company. Therefore buying all the shares means that you own the whole company. Therefore one may buy the majority of the shares (51%) not for financial trading but to influence how the company is run, this is physical. But normally a share contributes to less than a tenth of a percentage of the company. So if not many shares are bought then the person is more than likely buying the share for trading later in the future; this is more abstract as one has bought something to sell later on at a different price but the percentage ownership is so low it’s as if nothing was bought.

These markets are very profitable, such as the most liquid market forex has an average daily trading volume of $5,000,000,000,000 (5 trillion dollars) and the total market capitalisation in 2017 was $76.4 trillion. This shows the scale of the volume of money traded and the potential profitability, which is why many people decide to get into trading stocks. However the stock market is very volatile and changes a lot daily, however even though you can make money when the stock market is generally going up, you can also make money when the stock market is generally going down, this is purely due to the volatility of the stocks.

The stock market is a volatile place where a lot of people decide to invest in, for the chance of making profit; however many would argue without a lot of practice and research it is much easier to make a loss than profit. The reason for this is because every share price changes by micro amounts in micro seconds, and is affected by an almost infinite number of factors including factors such as public doubt; factors like this makes it hard to predict as it’s impossible to quantify or predict how public doubt will affect it. This is also part of a number of factors so will affect a percentage of its change. This makes the stock market only really accessible to a certain group of people, because it can be argued that it’s easier to make a loss than a profit. However the key to success is you only have to win more than you lose to make a net profit. So if you guess/predict more stock prices right than wrong then you will overall make profit over time.

A lot of people take 2 main approaches to trading with the stock market and predicting a share price: the first is manually/physically, this is where the user looks at the history of the stock price, company news, and other pieces of information and make a calculated decision based on all the researched information. The second is automatically/mathematically, this method works by mainly looking at the graph and different values calculated from this and using these values to predict the new value, these include smaller moving day averages over taking bigger moving day averages. It can be done automatically by telling a computer to buy and sell when certain conditions are met, but mainly it is done with more user intervention, and the computer is mainly used to calculate important figures which the user uses to make a calculated decision.

This leads to the problem.

**Problem:**

Many factors affects the share price, this is all too much for most people to calculate and evaluate, thus reducing the reliability of the users decision to purchase or sell their shares. Additionally a user tends to have a portfolio of lots of different shares from lots of different sectors. This again for most users is just random guess work rather than calculated decision making. Ultimately the solution to this problem is to make the stock market more predictable and allow more people to invest with more confidence without the need to do a lot of research, without it being guess work.

Therefore my consultant *Marcin* assigned me a task to implement an algorithm that allows the stock market to be predicted to a reasonable degree of accuracy, specifically using stock market correlation as an important factor that affects a stocks share price.

The solution to this problem for me was to target a specific area of this rather than just generally. The creation of my project idea was the principal that the different financial stocks are linked such as; generally the basic material or mining stock sector is increasing then the energy stock may increase. For example the mining sector generally is increasing because they have found a lot coal, this in context means that natural fuel resources aren’t due to run out as quick. This then would increase the general energy stock as the contracts they have with the mining company can remain and the companies can remain, thus increasing the energy stock value. This is also true between companies such as companies that have contracts with other companies, even companies that have heavy rivals can have opposite impacts on their rival. This is known as stock correlation or stock sector correlation.

My program will be act as a solution to this problem, as it aims to pull data over 2 different financial markets FTSE100 and S&P100, it will pull data about the past year, and the data pulled will be the opening and closing price, the average price, the high price for the day and low price for the day. So it will pull 365 bits of those data and then use algorithms to see correlation and predict a certain increase or decrease for the 365th day and a few latter days after that.

The program will then analyse all the stocks selected and then see stocks are directly proportional and which are inversely proportional. Then it will give it a score between -1 and 1. 1 being they are perfectly proportional and -1 being they are perfectly inversely proportional. It will also give an uncertainty for the prediction and a likely hood number that will be associated with each stock.

This program will help people wisely choose their portfolio of stocks as you wouldn’t want to have 2 stocks that are inversely proportional, as the money you would make on one you would lose on the other. On the other hand it easily enables you to increase your profit as now you can get it with two companies rather than one. It will also allow a user to see a figure which is likely to be the stocks next day price.

My consultant Marcin who requested this project is trying to find relationships that will enable a more accurate prediction of the stock market this inter-stock correlation will allow one to more definitely map a company’s share value price and furthermore extrapolate that graph to predict whether a company will be profitable in the future. The project will also make the stock market more accessible to a wider audience and could potentially have a very wide user base, the program itself has the potential for profitability as it allow more people to interact with the stock market.

**Initial Research:**

**Extras that need to be considered:**

After some initial research and some advising it became apparent to me that the projects algorithms may be seen as a bit too basic, so extra operations and actions need to be done with the data created and received.

Such as making a portfolio where the computer chooses the stocks, additionally could use this data to help the algorithm trading helping the computer choose and decided which stocks to buy and sell. Also need to decide how the information needs to be presented, as it maybe python or Delphi the GUI will have to be more basic. Ideas to make the algorithm more complex are listed below:

* GUI
  + This entails having a clean graphical user interface which would allow users to interact with the project and algorithms more easily, thus allowing more user control as well. However if the project is to be implemented with a GUI the project may not be implemented in python as python is notoriously known for being excessively difficult for making GUI, however the GUI could be in a simpler form such as displaying a sorted list from highest to lowest, and could include a filtering option as well.
* Winning portfolio
  + This addition involves using the data created and predictions made to make a variety of winning portfolios that the computer believes will make you the most money in the future. It will give you a minimum of 5 portfolio options with at least 5 different stocks in each portfolio. This will allow a user to see how closely there stocks match these portfolios. This will require the need for an extra algorithm to make multiple instances of its perfect portfolio taking into account factors such as stock correlation, previous years stock prices, how the stock did in this quarter(e.g. Q2), and other factors
* Use it for algorithm trading
  + This entails the program and algorithm not only making predictions of different stock shares but actually executing these predictions on the stock market. Also the program will adapt to different methods of algorithmic trading when different conditions are met, while taking into account the correlation factor between two stocks.
* Use it to influence potential purchases and sales
  + This addition is more like a suggestive feature that suggest when to buy stocks and suggests when to sell stocks that are in your portfolio. These suggestions naturally will come with reasons behind them which can be given on request. Moreover it will have a projected price to sell, and a projected day to sell at, which it will alert you when approaching and change daily depending on the new data for the day that past.

**Websites Visited:**

<https://www.investopedia.com/articles/active-trading/121014/stock-correlation-strategy-effective.asp>. This website is titled “Is the Stock Correlation Strategy Effective?” and this website describes the basic principal of this and whether or not it is an effective strategy to implement or whether one’s time would be better spent elsewhere. It talks firstly about diversification (having a portfolio of different stocks from different sectors) and how it limits risks and shields overall returns, this is dependent on the degree of correlation between stocks which in turn make it difficult to achieve diversification effectively. Correlation coefficient can be calculated using the formula in *figure 1*. Where cov= covariance, and σ is standard deviation. In *figure 2* you can see a list of values in an excel document between 2 different stocks. The 2 stocks had a correlation coefficient -0.11 this means that stocks are not very inversely correlated. The website was also saying that stock correlation can be used in 2 ways. 1: to maximise profits and potential losses by having a highly correlated portfolio, a high correlation coefficient. 2: to have an overall null correlation coefficient this will protect your portfolio from big losses due this correlation. Ultimately the correlation coefficient changes over time, or when certain conditions are met and needs to be repeatedly calculated so that the portfolio accounts for this.

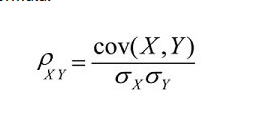
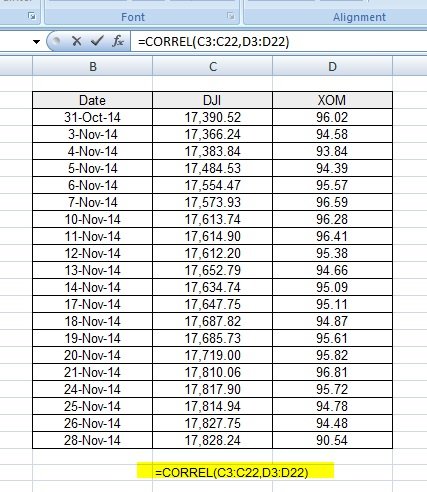


Figure 1

Figure 2

<http://francescopochetti.com/stock-market-prediction-part-introduction/>. This website showed mainly how to implement stock market prediction in python, which I would be able to translate into Delphi, given some time. Showing what libraries are needed to make this work. The actual prediction algorithm was quite a simple algorithm which took a few values and returned a binary value which was dependent on whether the prediction was for it to go up or down, as you can see in *figure 3.* This website told me the data where most of the predictions are made off which are: Date, open, close, high, low, volume, and adjusted close. This website showed me that the data can be pulled of the internet and the important figures that are needed for the day.

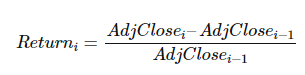
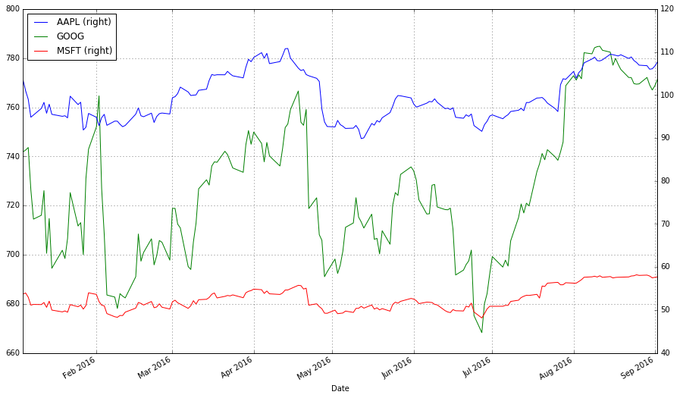
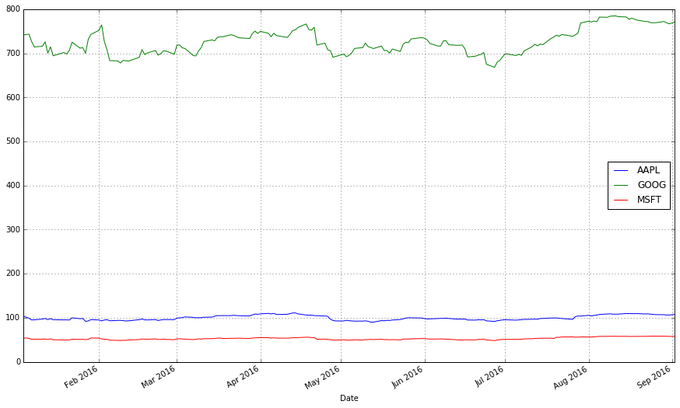
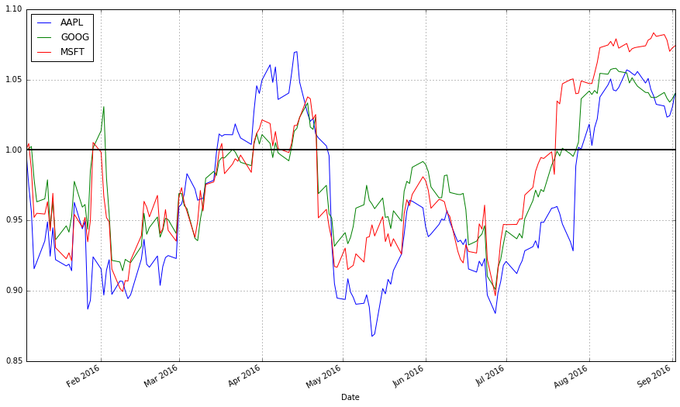


Figure 3

<http://blog.yhat.com/posts/stock-data-python.html>. This website showed me that you can use the “pandas” library to easily pull the data from an online finance site such as Yahoo finance or Google finance. This site also said how to visualise the data by graphing the data using “matplot” library, if graphing was needed in the end. It also said the tips of plotting the stocks, for example you shouldn’t plot the prices as if you want to plot more than one on the same plot as more expensive stocks ruins usability of the graph plotting. The site also enlightened me about moving day averages, and how that algorithm can be implemented to see whether a stock should be bought.







<https://www.oreilly.com/learning/algorithmic-trading-in-less-than-100-lines-of-python-code>. This site showed me the importance and value of back testing. Back testing is where you pull the data from 2 years ago and use that data to predict the data for last year, this is useful as you can compare the predicted data of last year to the real data of last year and see how right or wrong the algorithm implemented was. This eliminates the waste of time waiting for data to come in for the prediction model to be wrong. Additionally if I have the time can adjust the prediction model based on how the model is changing over time. To match the current pattern.

<https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp>. This website showed me the different types of algorithmic trading that are normally used to make profitability from stocks. The website also enlightened me to many of the advantages of algorithmic trading. The different types of algorithmic trading strategies were: Trend following, Arbitrage opportunities, Index Fund Rebalancing, Mathematical Model Based strategies, Trading Range (Mean Reversion), Volume Weighted Average Price (VWAP), Time Weighted Average Price (TWAP), Percentage of Volume (POV), Implementation Shortfall and many more strategies can be used to predict the future of a share price. To explain one example Arbitrage opportunities is an algorithm that takes advantage of slight differences between different stocks in different countries, for example a share may be listed in 2 different international stocks indices. The algorithm will check the price of the 2 in the different stocks markets converted using the exchange rate, and if there is a profit to be made it will execute this and make a profit. The strategy aims to make lot of high frequency trades to make a lot of money. However this strategy is not perfect because even though a computer can do this in micro-seconds the stock market also changes in micro-seconds and causes the problem of if a computer buys a share then the stock on the other side goes down. Does it get stuck with this stock which it doesn’t know what to do with it?

<https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/what-are-algorithms-algos/>. This website is similar to the one above, where it explains algorithmic trading and give more different examples, whilst explaining their positives and negatives, and how the algorithms actually work, and are implemented in the world of today. This website also told me that the algorithms are used by investments banks and hedge funds.

<https://www.quantinsti.com/blog/algorithmic-trading-strategies/>. Again this website talks about different algorithmic trading, and how to implement them such as simple moving day averages. It also talked about going from models to actual algorithm and how to go about making this. Again back testing was discussed, also optimisations was discussed and a numerous other configurable settings that can all be adjusted.

This essentially showed me what was needed for me to complete this project as many topics were began to reoccur, this shows me that I will soon be able to start prototyping what I need to make.

**Ideas that should be considered**

There are 2 things the algorithm could do predict whether it will go up or down, and if given enough time can estimate by how much. With this data it should also give its own confidence rating about how likely it will be to achieve this prediction. This could be based on previous predictions accuracy and how the new prediction has now changed from its reliability. This AI feature will be done by recording all the readings it makes storing them then recalling them to make a better decision, or to give it guidance on its decision. The hope of this is that over time the program will refine its prediction until ultimately it makes very accurate predictions.

Also can either include predictions for the day, week, month, quarter, and/or year, as an up or down binary figure and if time allows as a distinct numerical value.

There are many different algorithms which can be implanted to predict the stock price I ideally would like to incorporate as many as possible of these algorithms as possible. And hopefully use an AI feature to weight each algorithm accordingly in predicting the stock price.

An additional feature that I wish to be implemented is a predicted sell date time and a predicted buy sell date time for stocks that are already bought or are maybe waiting to be bought.

The AI feature I want ideally to be prominent in the solution to this problem. I intend to tackle this by storing the data that it predicted and the actual data that occurred seeing how off it was and by changing the weighting of the algorithms or by increasing the uncertainty or by adding a value to the data to accommodate for this in the future. But if the stock is within let’s say a 1% of the actual price then it doesn’t affect the algorithm as it would be down to human sway. Consider including algorithms to do with ‘hypothesis testing’ in the solution.

**Reason for the user to want this:**

The user decided to want a solution (algo-trading) like this because it eliminates the human factors that influence trading such as emotion, doubt and any psychological factor . Additionally the computer can check all the factors such as moving day averages a lot quicker than a human can check them therefore they can make informed decisions a lot quicker than humans. They also can execute trades instantly so they trade at the price they wanted it. It can also make simultaneous transactions on multiple platforms instantaneously which would be impossible for a single human. It gives reduced risk in manual errors made in placing trades. The client would like to exploit the speed of a computer which is much greater than a human.

Greatest proportion are HFT’s which are high frequency trading this involves making many trades on many platforms very quickly repeatedly based on a specific algorithm set.

**Consultant Questionnaire:**

A questionnaire was made for my consultant Marcin for purpose of allowing me to know which bits of the project were important and which bits were less important, allowing me to prioritise certain aspects of the project over other. This was very useful for allowing me to create my objectives for the project that I planned, as I knew which parts of it were a must and which parts were not necessary but would be great to add if time allowed. Find the questions asked and answers below:

1. **What is the application of the project?**
   1. For making money off the stock market, with the least amount of risk, and guess work.
2. **What are the favoured features of the project?**
   1. Correlation algorithms and the flexibility, configurable
3. **Is a GUI necessary?**
   1. GUI isn’t a necessary part for the solution of this project. As the main part is the actual numbers output, but an easy to use GUI is helpful and appreciated.
4. **How important is user control?**
   1. Needs to be configurable, however it’s too early to see how much it will be configurable. Will decide later in the future.
5. **What stock markets and shares particularly would you like to look at?**
   1. Focus on the UK or/and US market. Easiest one’s to obtain such as (FTSE 100, S&P100, DJ30)
6. **Is there a need for manual and automatic trading methods?**
   1. Not decided yet, however a lot of back testing will be done to create hypothetical profit/loss.
7. **Would you want program to suggest particularly profitable trades?**
   1. Yes, would also be helpful to display them in a sorted list, from most profitable to least or in reverse.
8. **How often would you want the program to refresh its predictions?** 
   1. Every run time program should refresh should compare against previous run time predictions and data.
9. **Would you like a suggested ranking option of profitable stocks, which could also be filtered by sector?**
   1. Could all be done in excel But it’s not a priority to worry about, not very important.
10. **An option for the user to see the reasons the program decided to buy or sell a stock? (To see data such as its 20 day moving average overtook the 100 day moving average)**
    1. Based on profit/loss and back model testing is one that would be favourable.

Also after much consideration the programming language of this project would be in Delphi. This is because Delphi has a much easier implementation of object oriented language, and creation of a GUI. Which would allow the solution of this project to be done easier, also even though I have a background in python recently I have been using Delphi so I believe I am more competent in Delphi allowing me to integrate more complex solutions in this language.

**Objectives:**

All this preparation ultimately lead me to come up with a series of objectives that I hope to achieve during the completion of this project. They are grouped between ones that I *must, should* and *could* achieve.

The objectives were made over many iterations and were added to and amended during the analysis section of the project. The objectives were also completely flexible and were discussed and re-discussed many times between, myself and the client.

**Must:**

1. Pull Data from a .txt file.
2. Have a format for at least the stocks open, close, high, low, and volume of chosen stocks from the past year.
3. User chooses the maximum amount of stocks that they can have in one instance is.
4. Be able to display all the data of a stock that was pulled.
5. To be able to display specific lines of the data you want.
6. Find direct correlations coefficient.
7. To predict whether the stock is going to go up or down as a binary value.
8. An easy to use UI.
9. Be able to delete stocks that are no longer needed.
10. Be able to restart the program on demand from the user. Without having to exit the program and start again.
11. Draw a graph of a pulled stock.

**Could:**

1. 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.
2. Make a suggested portfolio.
3. Pull Data from a .txt file without changing its content.
4. Include a value which tells the user how likely the computer thinks that its prediction will be correct.
5. To make the computer automatically calculate all the stocks pulled correlation.
6. To sort the correlated stock from highest positive to highest negative (1🡪-1).
7. 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.
8. 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)

**Extensions:**

1. To be able to pull data from an API on the internet on demand, from the user, by use of the stocks Ticker symbol.
2. To be able to pull stock data from an API of different lengths (E.g. 3 months 🡪 2 years).
3. Introduce some AI machine learning so the predictions get better over time.
4. Actually execute purchases and sales on the stock market with real or fake money.
5. Error handle for the majority of inputs.