Correlation Between Stock Values of Various Companies Using Big Data

Problem Formulation/ Introduction:

Stock market was introduced in the 1790's. With every passing year the amount of capital in trading is increasing, the number of major companies in the market is ever increasing and the people involved is also increasing. This creates many variables that have an impact on the stock prices of a company. Some of the variables have more impact than others, a more accurate prediction could be achieved if most of these variables are considered for the predictions. Stock market is a highly sort after field because of this volatility and the uncertainty. There are many papers published on this topic but the field is so vast and the prediction capabilities of these papers is so limited that there is still a considerable expanse to explore. One part of stock markets that doesn't seem to attract a lot of researchers is the impact of other stock prices on a specific company's stock price. In this project, I will try to generate a list of correlation coefficients for the stock prices of all the companies based in USA. In this project, I will consider the high correlation values in both positive and negative side. A positive correlation indicates that the increase the stock price of this company can cause the stock price of the company under consider consideration to increase. A negative correlation indicates that the increase the stock price of this company can cause the stock price of the company under consider consideration to decrease. A high value on either side (> 0.5 or <-0.5) can have an impact on the stock prices.

This is interesting Big Data problem because there are almost 12,000 companies registered on the market. Some correlations can seem obvious but there might be some that we may not relate. Only way to be certain that there is no correlation or less correlation, is to find the correlation coefficient for all the companies with each other. So many companies with the data from the last 5 years, makes this an interesting Big Data problem.

This project will focus primarily on generating the relations. Making predictions of stock prices using correlations alone will not yield a good result as there are many other, more important factors for stock prices. The correlation can be used on top of the existing stock market prediction algorithms. The idea with this approach is to improve the accuracy of the prediction algorithms by 5-10% because in many occasions the prediction algorithms will not foresee a change in stock price as the impact of a related stock price is not part of the prediction algorithm.

Broadly, there are two main parameters that affect the stock price of a company [1]: Stock price trends, major events that influence stocks of the company. Average traders mainly use stock trends for their predictions but given the various factors that influence the stock prices, using trends alone will only give around 70-75% accuracy. We can increase that by considering the events that impact a company. For instance, if a company releases a new product which is well received then its stock price is likely to increase as the company is about to make profit from the sales. Incorporating this approach to the trends can increase the stock price predictions to close to 85%. This leaves the question of what else is influencing the stock price of a company.

This is where the correlation factor can help increase the accuracy. If we can use the top 3-5 companies that have an impact on the stock price of the company under consideration, then it is possible that we can increase the accuracy. As mentioned above, I would like to consider the companies that have the most correlation in both positive and negative way. If we know how much the correlation is, then we can use that data to predict the stock prices of the company. We can run the event based analysis on the

correlated companies and the events that influence the stock prices of those companies will have an impact on the company under consideration.

For an average trader, correlation is primarily used for portfolio diversification. Their target is to purchase stocks in companies that have very less to no correlation whatsoever. This is a safe way to invest in stock market as it is unlikely that a bunch of unrelated companies can all lose value at the same time. This can be another application for this project but this is not the primary consideration for me.

There is some research based on the correlation of stock values and the trading volume [4], this is an interesting approach but it won't be considered for this project. The volume of the trades won't be considered.

Strategy:

The main issue with this topic is getting the data required. There are over 12000 companies registered with NASDAQ in the USA. There are some methods that allow us to mass download historic data but even those are not built to get data of this magnitude. This needs to be done semi-manually. The historical prices usually include open, close, high, low and adjusted close of stock price on a given trade day. Fortunately, I only need the adjusted close price to get a correlation.

Another issue that I faced while getting the data was that quite a few companies in that list where created in the last 5 years, so the stock values for these companies was "nan". To overcome this, I took an average of that column or the average stock values of each of those companies and put that values in their respective columns where ever a "nan" is found.

I plan on using Pearson's Correlation Coefficient to get the correlation between the companies. Using **Apache Spark** [2], it is possible to generate a matrix that contains the coefficients. Ones the matrix is generated, we can use it to understand the companies that have most impact on the stock prices of any company.

The initial idea was to use the correlation matrix generation function which is available in Apache Spark but given the size of the input matrix, this approach is not a realistic option as we end up with an error in reading the data. I had to use a modified approach that made the code not very efficient but it works. This approach will be explained in detail in the next section.

Software Design:

The idea in the proposal was to create a coefficient matrix but given the size of the dataset, this is not possible. To overcome this, I had to individually compare and calculate the correlation coefficient for each of the columns with respect to every other column. This is not the most efficient approach but given the size of the dataset, I couldn't think of a better approach. Due to this the run time for the code is a lot more than it would've been for a built-in function.

With this approach, after every iteration I get a row of coefficient values. After a complete run, the final output will be a matrix with all the correlation coefficient values. Given the size of this matrix a graphical representation is very difficult. I plan of showing the top 20 correlations, the companies involved and the correlation coefficient.

Correlation Between Stock Values of Various Companies Using Big Data

In this algorithm, the correlation matrix itself was fairly straightforward. The main issue was with computing a delayed analysis. If we are to consider the correlation of a company's stock price with other companies after a certain period, then the dataset has to be remodeled with every iteration. This causes a major increase in the computation time.

I was able to do a 1-day delay in the stock prices but didn't have the time for a longer time delay calculation.

Testing:

Since this a correlation matrix, there is no real way to test the validity of the results. This uses a tried and tested approach of Pearson's coefficient to calculate the correlation and the results should be fairly conclusive. Only way to confirm would to be manually test the outputs.

I actually did go through the final result and tried to calculate the Pearson's coefficient for the top few results and got a similar result.

Results and Evaluation:

There was not much of a difference in the correlations for no delay and 1-day delay. I was not able to calculate 1-month delay in time for the submission so I am not considering it.

The top 20 correlations for No delay:

I	J	Coef
MJN	MITSY	9.99
MNRO	MNPP	9.98
GOL	GOLD	9.98
RTN	RHHVF	9.94
SYK	SPN	9.93
HE	HII	9.93
MCY	BSEFY	9.91
EGN	DRQ	9.91
SCHL	NVS	9.90
MSA	ELS	9.88
HMC	ADS	9.88
JIXAY	SPN	9.88
SCMWY	GJV	9.87
SCMWY	RGR	9.87
SPG	ADS	9.87
RJF	JIXAY	9.86
GPDNF	SHW	9.86
KCRPY	HAYN	9.86
EPC	PRU	9.85
EQIX	PRU	9.85

Correlation Between Stock Values of Various Companies Using Big Data

The top 20 correlations for 1-day delay:

I	J	Coef
GOL	GOLD	9.99
SYK	SPN	9.99
MNRO	MNPP	9.99
RTN	RHHVF	9.98
MJN	MITSY	9.98
SCHL	NVS	9.97
MCY	BSEFY	9.96
EGN	DRQ	9.96
SCMWY	RGR	9.95
MSA	ELS	9.93
HMC	ADS	9.91
JIXAY	SPN	9.91
SCMWY	GJV	9.90
EPC	PRU	9.88
HE	HII	9.87
SPG	ADS	9.87
RJF	JIXAY	9.86
GPDNF	SHW	9.86
CDNAF	DST	9.85
KCRPY	HAYN	9.85

We can see that the top 20 correlations for No-delay and 1-day delay are almost the same except that the correlation coefficient was slightly different.

This is quite understandable as the delay is only 1 day. I might have seen a bigger difference if I had tried a longer delay of about 1 month. Hopefully I can get that output before the demo.

This is not a very unique approach, although it is relatively new in the field of computer science. There has been some research done in the field of finance to understand the correlation in stock market. There hasn't been any research done on this topic using Big Data as most of the papers that I was able to find involved comparing the stock values of companies preselected by the researcher. The target for this project was to create a correlation between the stock prices. The top 20 is just for the visual representation. The complete correlation matrix is stored and can be used to find the top correlations for each company if needed.

Correlation Between Stock Values of Various Companies Using Big Data

Biblography:

1. Girija V Attigeri , Manohara Pai M M, Radhika M Pai, Aparna Nayak, "Stock Market Prediction: A Big Data Approach", *TENCON 2015 - 2015 IEEE Region 10 Conference*, 07 January 2016.
2. Matei Zaharia et al, "Spark: Cluster Computing with Working Sets", *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, p.10-10, June 22-25, 2010, Boston, MA.
3. J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. Commun. ACM, 51(1):107–113, 2008.
4. J. Cambell, S. Grossman, J. Wang, "Trading Volume and Serial Correlation in Stock Returns", *National Bureau of Economic Research*, October 2016.
5. http://www.investopedia.com/, last visited November 30, 2016.
6. https://finance.yahoo.com/, last visited November 30, 2016
7. http://www.nasdaq.com/, last visited November 30, 2016