Pandemics and the stock market

Time series analysis of the stock market during pandemic

Austin Wilson
Computer Science & Statistics
California State University of Sacramento
Sacramento, CA, US
austin1224@gmail.com

Roberto Campos
Computer Science
California State University of Sacramento
Sacramento, CA, US
campos.roberto100@gmail.com

Isaac Shah
Comp Sci & Astrology
California State University of Sacramento
Sacramento, CA, US
ishah@csu.edu

ABSTRACT

COVID-19 sent shockwaves throughout the globe. With stay-at-home orders, the global economy came to a screeching halt. In this report, we aim to analyze the effects of the pandemic on the US stock market. We use Linear Regression, Linear Support Vector Classifier, K Nearest Neighbors and Random Forest Classifier models to analyze patterns of the market before and after the SARS, H1N1, Ebola and COVID-19. To analyze the data, we use Python and Jupyter Notebooks. Libraries such as NumPy, Pandas, sklearn, matplotlib and statsmodels helped us read, manipulate and predict the data. We also wanted to predict how COVID-19 could pan out in the next couple of months based off of other outbreak data, such as SARS and H1N1. After looking into the data and trying to connect COVID-19 data to stock market data we decided to focus on financial markets and use dates of pandemics to train our models. Essentially we are using market data from the time of other pandemics to see how certain companies perform during pandemics. We will use classification techniques to classify stock from a handful of companies as buy, sell or hold.

1 Intro

Epidemics shake the very foundation of society. Mankind has been at war with this unseen enemy since the dawn of time. For example, the Black Plague killed nearly 1/3 of the population of Europe. That said, it is important how an epidemic propagates throughout society and how we can mitigate its effects. However, due to the nature of COVID-19, we are treading on unmapped grounds. It is important to note that although similarities are being made with past breakouts, we should take any conclusions from this report with a grain of salt. An important aspect of society is the economy. Disruption to the economy can lead to devastating results. An example of collateral damage is the plummeting of oil prices. Fewer people are traveling, therefore less fuel is being used for planes, cars and all forms of traditional transportation. The energy sector has been heavily impacted by this recession. Oil futures now have a negative value due to the surplus of oil. In fact, crude oil has been hit the hardest according to our results.

2 Data collection

We began our data collection by scouring the internet for financial data. We found that yahoo finance was one of the best sources.

2.1 Initial method

We first looked at financial data from some of our favorite companies on Yahoo Finance. We got data from Apple, American Express, Nike, Chevron, Johnson & Johnson, Ford and Alaska Airlines. We realized that this may not be the best way to go about getting the data. We found that the data manually downloaded from the internet was not robust enough. We only had 100 rows of data and this did not include some of the dates we wanted to investigate.

2.2 Pandas datareader

We did some research and found an API for collecting financial data built on pandas. This API is called pandas_datareader. We began to test this API by doing some exploratory data analysis. We requested data from the same stocks we had manually collected before to see if we could manipulate the amount of data returned. We were very pleased with the results. We decided that we should get data starting from January 1, 2000 until the current date. We were able to re-request the data several times and update the data to the current date. The last date we updated our training data was April 24th. We also used this API for testing the accuracy of some of our predictions.

2.3 S&P 500

As we moved on we realized that data from just a few stocks would not be enough to represent that entire US economy. We quickly turned to the S&P 500. Although we had the option to just download this data from Kaggle we decided to build out an entire data pipeline ourselves. We wanted to acquire data from all of the companies in the S&P 500, gold, crude oil as well as some data on treasury bonds. As a note to the reader, the S&P 500 is a stock index that contains information on 500 largest companies listed on stock exchanges in the US. Largest is an ambiguous term and will be defined as the companies with the largest market capitalization. Market capitalization or market cap is the number of outstanding shares multiplied by the stock price. This index is widely used as a metric for the economy as a whole. We had the option to collect financial data from this index but decided to go another route. We wanted to collect all the data from each company listed on the S&P 500.

2.3.1 Web scraping ticker symbols The first step in our data pipeline was to obtain the ticker symbols for all the companies in the S&P 500. We used Beautiful Soup, a web scraping library in python to gather this data from a table on Wikipedia. Although Wikipedia was not necessarily the best source for financial data we are confident that the ticker symbols we obtained are accurate. The companies listed on Wikipedia were last updated on April 19, 2020. Our web scraper stored this information in a pickle object. Pickle is a library that serializes Python objects into a character stream. Information stored as a pickle object can easily be reconstructed. Pickle objects preserve the underlying structure of the stored object. The next step was to gather data from the 500 largest companies using the ticker pickle object.

2.3.2 Using ticker symbols for API requests We first created a directory to store all of the different company's data separately. We wanted to have access to all each company's data individually to have the option to inspect at a micro-level. Then we defined that start date as of January 1, 2000. We updated the data with different end dates periodically. Then we loaded the tickers from the pickle object. We looped through each ticker and loaded all the corresponding data into a single CSV file. We had some issues with a KeyError. This is likely something to do with changes in the S&P 500 not reflected in the data we scraped from Wikipedia. We put everything inside a try-except expression to make sure there were no

errors thrown. Inside the try-except statement we created a new CSV file using the name from the pickle object. Then we requested the data through the pandas_datareader API via Yahoo Finance. We did this for each company in the S&P 500. In order to check for updated data, we added a conditional to see if the file exists. When we came back later to update the data we deleted the data uploaded from the API again. The next step was to combine all this data into one data frame.

2.3.1 Single data file In order to create a single data frame for all the S&P 500 data, we decided to slice the data. The only columns we are interested in are date and adjusted close. Adjusted close is very similar to the closing price of the stock but accounts for stock splits. If a company thinks that the stock price is too high they will split it. This is important for machine learning because a stock split should not reflect the underlying value of the assets. We went through the directory containing all of the individual stock price data and dropped all the columns other than adjusted close. We set the date as the index and combined all these data frames into a single data frame. The final step was to save this data as an individual CSV for later use. The purpose of doing all the data acquisition was to have options for what we analyzed. We can pick whether we want to analyze the S&P 500 or we can look into data from any individual company. As it turns out we later decided to just look at individual companies.

3 Data preprocessing

After collecting this abundant amount of data we had to figure out what to do with it. Since our main goal is to see what happens to financial markets during pandemics we decided to look at financial data from times when a pandemic is going on. We choose a few different pandemics to look at. The data we have access to is from 2000 until 2020 so the pandemics we choose to look at were SARS, H1N1, Ebola and of course, Covid-19.

3.1 SARS

According to the CDC, the first case of SARS was reported on November 16, 2002. This pandemic went on for a little over a year and died down in late 2003. We subsetted our data for SARS between October 10, 2002 and February 1, 2004. We wanted to include a few months before and after the pandemic to get an idea of what was going on financially before and after the pandemic. During this pandemic 8,098 people were infected and 774 died. SARS was relatively short and harmless when compared to something like H1N1 or covid-19. Nevertheless we still investigated this time period because ultimately there must have been some reflection in the market.

3.2 H1N1

The next pandemic we analyzed was H1N1. The rest of the world was extremely negatively impacted but the US was somewhat unscathed. The first infection was recorded on April 12, 2009. This pandemic was much more destructive, although mostly abroad. There were 60 million cases and 12,469 deaths. It ended April 10, 2010. This pandemic is of particular interest because of how widespread it was. One other issue will be that this pandemic occurred just as markets were on the rise after the 2008 financial crisis. We subsetted our data accordingly and moved on to the next pandemic.

3.4 Ebola

The last pandemic we looked at, besides covid-19 itself was ebola. According to the CDC this pandemic lasted a little over a year from March 2014 until July 2015. Ebola is somewhat of an ongoing problem. It was hard to find a lot of information on what I was looking for. This disease has a lot of outbreaks in Africa. I could not find an exact number for infections and deaths but all we really need to subset the date is a start and end date.

3.4 **Selecting data**

We wanted some companies that were heavily impacted as well as some companies that were not as heavily impacted. The companies we chose were Apple, American Express, Nike, Chevron, Johnson & Johnson, Ford and Alaskan Airlines. We felt that this was a good mix of companies from different sectors that may be affected differently. The next step was to put the data from all these companies into one data frame.

3.5 **Subsetting selected data**

We put all the data from the chosen companies into one data frame and dropped all columns except adjusted close and date. What we ended up with is a time series with all the companies we are interested in for the period of time that we are interested. This is the data we want to send into the model. We also built a data set for all the data from the S&P 500 from 2000 until 2020 to test against. We stored the data from each time period corresponding to different pandemics in separate files.

3.6 **Processing data for classification**

The next step in our model was to process the data for labeling. This was a key step in the logic for our model. We wrote a function to return the percent change over seven days and store it in a new column. This function will calculate the percent change over seven days for each stock. The percent change as we define it in the function is the price i (a variable taking on values from 1 to 7) days later minus the price of the first day divided by the price on the first day. We then store these values in a new data frame and replace any missing values with 0 just in case. The essence of this function is to see how the stock prices behave in the following 7 days.

3.7 **Encoding the data for classification**

Next we wrote a function to encode buy or sell or hold a stock. We coded our requirement to be .02. This means that if there is more than a 2% increase in the value of the stock each day then we will count that as a buy. If there is less than a 2% increase in the value of the stock each day then we will count that as a sell. Otherwise the stock will be default as a hold. If the stock has a 2% gain for 7 days it should be bought. If the stock has a 2% loss of 7 days it should be sold. The assumption here is that if a stock is decreasing in value it will keep decreasing in value and vice versa.

4 **Model design**

Initially we had the idea of implementing a regression model to predict the stock price. We also used ensemble learning with Linear SVC, K Nearest Neighbors and Random Forest Classifier. This decision was based on the fact that predicting a stock price is very difficult. Classifying a stock as buy, sell or hold seems a little more practical.

4.1 **Linear Regression**

We started with a simple linear model. We converted the dates to ordinal format and trained a model from the statsmodels library in python. Our x value was the converted dates. Our y value was adjusted close stock price. We fit the model on a subset of the converted dates and adjusted closing price and made some predictions. We found this method to be inaccurate. The assumption that there is a linear relationship between date and stock price is hard to justify. This over simplifying assumption will undoubtedly result in a large bias. Stocks tend to fluctuate up and down over time. Some of them trend upwards in the long term. This assumption could be used for a very long term investment strategy.

4.2 **Classifier**

We looked into several classifiers. We started with the simplest and slowly moved forward. Eventually we combined all the classifiers and use ensemble learning with a voting classifier.

4.2.1 KNN We had good results with the KNN classifier. We did have some issues with uniform support of our classes. We thought we could improve by using a more sophisticated classifier.

4.2.2 Linear SVC The next classifier we tried was Linear SVC. We had no experience with this classifier and hoped for better results as a tradeoff for less interpretability.

4.2.3 Random Forest Classifier Similar to our decision to use Linear SVC we randomly chose to use LinearSVC for our classification and compare the results. Another big part of our decision to use the classifiers was that they were used in the tutorials that helped us put this project together.

4.2.4 Voting classifier In order to increase our performance we used all three classifiers and as a voting classifier to maximize performance

4.3 ARIMA

ARIMA stands for autoregressive integrated moving average. This is a combination of moving average, taking the average over a period of time and autoregressive which is regression for time series. The first step of creating the ARIMA model was to take the log of all the stock prices. This is done to scale the data down and make it more manageable to work with.

4.3.1 Autocorrelation The first step in building an ARIMA model is to determine the parameters of the model: p,q,d. P represents the number of lag observations or the number of units of time in the past you want to look at. Q represents the number of times the observations are differenced. D represents the size of the moving average window. When we plot our autocorrelation we find that the autocorrelation is constant so we use P = 0. This was the case for all of our companies of interest so we use P = 0 across the board for all our models. The autocorrelation represents the similarity between observations.

4.3.2 Partial autocorrelation Partial autocorrelation is similar to autocorrelation if you remove all the events between two events that are not next to each other. So it is how similar two observations are that are not next to each other. After calculating this for each stock we see the same trend for all stocks. It starts at 1 and then drops and fluctuates between a very small negative and positive number. This means that the parameter P = 0.

4.3.3 Graph The last step before building a model was to look at the graph. We graphed Adjusted stock price and saw an overall upward trend. This means that we should set the parameter D = 1 so that we just look at the day to day difference.

4.3.5 Models We build these models on all the data. We did not just look at subsets of the data when there was a pandemic. We looked at all the data and build models for predicting the price of each company.

5 Data visualization

Again, the companies we chose were Apple, American Express, Nike, Chevron, Johnson & Johnson, Ford and Alaskan Airlines. For each of these companies data was subsetted from each pandemic timeline. The subset begins at the first recorded case, declared pandemic ends at the end date according to the CDC. The subsetted data was standardized then turned into a simple but illustrative graph. The idea of standardizing the data was to scale and compare each pandemic visually through the graphs. The graph marks exactly where the first recorded case, pandemic declaration and end date happened. This allows the reader to interpret the graph in a much easier way usually observing the movement of the line after each point. Moving averages were also included in the graph for further understanding. Moving averages allows analysts to filter out the noise of random fluctuations to see where the trend is headed. Allowing the analyst to make better predictions about the movement of the stock. In many of the cases, after the pandemic was declared many stocks plummeted down due to the pandemic declaration, then rising after the pandemic ended. Our model predicts whether

we should buy, hold or sell, the graphs allowed further analysis to verify if the prediction is accurate. The library used to visualize the trends was matplotlib, along with the use of a simple script to automate the making of each visualization.

6 **Results**

We have results for two models. We recorded the predictions and accuracy for the ARIMA models and classifier for all the companies we chose.

6.1 **Apple**

Overall the sentiment on Apple is that it does well during pandemics and would be a good investment in the current economic climate.

6.1.1 Classification For Apple we found that the train data had 160 buy labels, 146 sell labels and 10 hold labels for the SARS data. The test data had 52 buy labels and 27 sell labels. For the H1N1 train data we had 183 buy labels, 122 sell labels and 30 hold labels. For Ebola the train data had 122 buy labels, 211 sell labels and 102 hold labels. The corona test data had 62 buy, 39 sell and 8 hold. The test data had 25 buy, 2 sell and 1 hold.

6.1.2 ARIMA The coefficient was .0009. Our three predictions were as follows:
4-27 prediction is $283.21 actual value is $283.17
4-28 prediction is $283.45 actual value is $278.58
4-29 prediction is $283.70 actual value is $287.73
The resulting root mean square error was 3.65.

6.2 **American Express**

American Express was somewhat ambiguous based on the classification data alone. We found that it does well in some pandemics and does not in others. In general the financial sector does not do well during pandemics.

6.2.1 Classification The SARS training data had 147 buy, 131 sell and 38 hold. The test data had 45 buy, 33 sell and 1 hold. The H1N1 training data 191 buy, 122 sell and 22 hold. The test data had 71 sell and 13 hold. The Ebola training data had 205 sell, 173 hold and 156 buy. The test data had 111 sell, 7 buy and 16 hold. Overall this is hard to conclude whether to buy, sell or hold this stock. The Corona train data had 46 buy, 40 sell and 23 hold. The test data had 19 buy, 7 sell and 2 hold.

6.2.2 ARIMA The coefficient was .0002.
4-27 prediction is $83.18 actual value is $85.06
4-28 prediction is $83.19 actual value is $88.19
4-29 prediction is $83.21 actual value is $96.12
The resulting root mean square error was 8.06.

6.3 **Nike**

Overall the sentiment on Apple is that it does well during pandemics and would be a good investment in the current economic climate.

6.3.1 Classification The SARS train data had 160 buy, 112 sell and 0 hold. The test data had 52 buy, 26 sell and 1 hold. The H1N1 train data had161 buy, 121 sell and 53 hold. The Ebola train data had 208 buy, 172 sell and 53 hold. The Corona train data had 51 buy, 39 sell and 19 hold. The test data had 21 buy, 6 sell and 1 hold.

6.3.2 ARIMA The coefficient was .0008. The prediction we made were the following:
4-27 prediction is $88.44 actual value is $89.37
4-28 prediction is $88.50 actual value is $88.8
4-29 prediction is $88.57 actual value is $88.07
The resulting root mean squared error was 8.06.

6.4 **Chevron**

Chevron was expected to be a sell because the energy sector was hit hard by social distancing. We ended up with a model predicting it as a buy. This is possibly because the stock is underpriced.

6.4.1 Classification The SARS training data had 128 buy, 112 hold and 76 sell. The test data had 32 buy, 31 hold and 16 sell. The H1N1 train data had 149 buy, 127 sell and 59 nold. The test data had 56 buy, 25 sell and 3 hold. The Ebola train data had 217 sell, 163 buy and 154 hold. The test data had 114 sell, 15 hold and 5 buy. The Corona train data had 54 sell, 30

buy and 25 sell. The test data had 22 sell, 4 buy and 2 hold.

6.4.2 ARIMA The coefficient was .0003. The predictions we made were the following:
4-27 prediction is $87.04 actual value is $89.37
4-28 prediction is $87.06 actual value is $88.8
4-29 prediction is $87.09 actual value is $88.07
The root mean square error that resulted was 4.89.

6.5 **Johnson & Johnson**

We expectd Johnson & Johnson to be a buy because they make hand sanitizer! People will buy more hand sanitizer and other products form this company when a virus is ravaging the whole world.

6.5.1 Classification The SARS training data had 143 sell, 110 buy and 63 hold. The test data had 64 sell, 11 buy and 4 hold. The H1N1 train data had 174 hold, 97 buy and 64 sell. The test data had 74 hold, 6 sell and 4 buy. The Ebola train data had 237 hold, 153 buy and 144 sell. The test data had 99 hold, 31 sell and 4 buy. The Corona train data had 52 buy, 29 sell and 28 hold. The test data had 23 buy and 5 hold.

6.5.2 ARIMA The coefficient was .0003. Our predictions were as follows:
4-27 prediction is $154.91 actual value is $154.29
4-28 prediction is $154.96 actual value is $151.39
4-29 prediction is $155.02 actual value is $150.24
The root mean square error was 3.46.

6.6 **Ford**

As expected Ford took a huge hit from the recession caused by COVID-19. Overall this company performed very badly and both our models reflect this.

6.6.1 Classification The SARS train data had 164 buy, 131 sell and 21 hold. The test data had 57 buy and 22 sell. The H1N1 train data had 185 buy, 139 sell and 11 hold. The test had 61 buy and 23 sell. The Ebola train data had 212 sell, 191 buy and 131 hold. The test data had 110 sell, 14 hold and 10 buy. The Corona train data had 56 sell, 34 buy and 19 hold. The test data had 23 sell, 3 hold and 2 buy.

6.6.2 ARIMA The coefficient was -.0002. Our predictions were as follows:
4-27 prediction is $4.868 actual value is $5.17
4-28 prediction is $4.867 actual value is $5.38
4-29 prediction is $4.866 actual value is $5.26
The root mean square error was .41.

6.7 **Alaska Airlines**

We also expected Alaska Airlines to perform poorly during a pandemic because it is a travel company and people don't travel when there is a pandemic.

6.6.1 Classification The SARS train data had 153 buy, 151 sell and 12 hold. The test data had 43 buy and 36 sell. The H1N1 train data had 211 buy, 116 sell and 8 hold. The test data had 65 buy and 19 sell. The Ebola train data had 294 buy , 204 sell and 36 hold. The test data had 104 buy and 30 sell. The Corona train data had 62 sell, 35 buy and 12 hold. The test data had 25 sell, 2 buy and 1 hold.

6.6.2 ARIMA The coefficient was .0003. Our predictions were as follows:
4-27 prediction is $4.868 actual value is $27.85
4-28 prediction is $4.867 actual value is $31.58
4-29 prediction is $4.866 actual value is $34.00
The root mean square error was 4.01.

ACKNOWLEDGMENTS

Thank you to Professor Meilei Lu and Victor Chen. Both are wonderful professors that helped us learn data science. This is the beginning of what will hopefully be a prosperous career in technology!

REFERENCES

[1] Angelos Delivorias. Nicole Scholz. 2020. Economic impact of epidemics and pandemics. (2020). Retrieved April 30, 2020

[2] Michael Grogan ARIMA and Python: Stock Price Forecasting using statsmodels. Retrieved April 30, 2020 from https://youtu.be/o7Ux5jKEbcw

[3] Harrison Kinsley. ARIMA and Python: Stock Price Forecasting using statsmodels. Retrieved April 30, 2020 from https://www.youtube.com/watch?v=2BrpKpWwT2A&list=PLQVvvaa0QuDcOdF96TBtRtuQksErCEBYZ

[4] Ishan Shah. How Pandemics Impact Financial Markets. (April 2020). Retrieved https://towardsdatascience.com/pandemics-impact-financial-markets-9a4feb6951f5

[5] Carson Kai-Sang Leung, Richard Kyle MacKinnon, and Yang Wang. 2014. A machine learning approach for stock price prediction. In Proceedings of the 18th International Database Engineering & Applications Symposium(IDEAS '14). Association for Computing Machinery, New York, NY, USA, 274–277.

[6] Pratik Patil, Ching-Seh Mike Wu, Katerina Potika, and Marjan Orang. 2020. Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models. In Proceedings of the 3rd International Conference on Software Engineering and Information Management (ICSIM '20). Association for Computing Machinery, New York, NY, USA, 85–92. DOI:https://doi.org/10.1145/3378936.3378972

[7] Biao Huang, Qiao Ding, Guozi Sun, and Huakang Li. 2018. Stock Prediction based on Bayesian-LSTM. In Proceedings of the 2018 10th International Conference on Machine Learning and Computing (ICMLC 2018). Association for Computing Machinery, New York, NY, USA, 128–133.