Matthew Zilligen and Greg Cooper

Stochastic Modeling

Project Proposal: Stock Market Markov Chain Model

Background and Motivation - Discuss the background of your project, what you wanted to accomplish and why it is important. Also discuss previous work found in the literature on the same topic.

Overall, we chose to predict the future stock market patterns of a specific stock using real data. We wanted to create a model using a markov chain that outputs daily behavior of the stock, while taking into account past data. Our goal is to create a Markov Chain that is based on the counts of the days where the stock price's daily growth rate can be characterized by a great state, good state, bad state, etc. This is important because the model can be tested on a variety of different stocks in different industries to both test the accuracy of the model and to compare the stocks against each other.

A stochastic model that predicts future stock price behavior is very relevant in the real world since many large companies are investing in a wide variety of different stocks. The stock market is incredibly important because it is one of the primary ways that both large and small

companies can raise capital. By understanding the behavior of stocks, financial derivatives, or even currencies one can make a lot of money buying or selling these instruments. Our Markov Chain model is just one way we can predict the behavior of these different instruments. It can tell us whether the past behavior of a stock will lead to various scenarios of different returns and volatilities. It also can tell us projections regarding the future distribution of stocks in question.

Since a stock market model is a common application of a Markov Chain, there has been a wide variety of previous literature on a similar topic. For example, Samuel Okafor (2020) looks at a Markov Chain model that is also focused on transitions between states of the return of a stock over a period of time. Specifically, he discusses a model comparing returns for one week compared to a two week basis using three states: an upward state, a stable state, and a downward state. (Okafor, S. 2020) Our model will take a similar concept except instead of only using three states, we will use standard deviations to make it generalizable for n states. Therefore, instead of only looking at up, down, and stable states on a weekly basis, we will look at many up states and many down states on a daily basis. Additionally, Vyara Kostadinova, Ivan Georgiev, Vesela Mihva, et al (2021) also created a Markov Chain model to model stock prices. Much like the goal of our model, they also implement a Markov Chain with more than just three states. Specifically, they use 6 states, and they set their states based on prices that are chosen arbitrarily. Our model adds to this by instead of choosing arbitrary prices, we look at growth rates that are a certain number of standard deviations greater than or less than the mean.

Team - Matthew Zilligen and Greg Cooper

Overall, Matthew and Greg divided the work roughly 50/50 for the code, report, and powerpoint.

Matthew Zilligen contributed to the project by researching multiple stocks in different industries with a variety of returns and volatilities. He found the stock data and loaded it into the code. He spent time reading different Markov Chain models from past literature. He spent time creating the primary function that uses past growth rates, means, and standard deviations to output the Markov Chain Model. He focused on making the functions in the model generalizable, specifically for the number of states of the markov chain, the standard deviations, and the number of days of past data. Matthew also proofread and made changes to Greg's code. He also was responsible for half of the report and half of the powerpoint.

Greg Cooper was responsible for the initial idea and directionality of the project in terms of how to create the model and the general goal of maintaining mean and standard deviation through the trial.. He generated the first draft in how to create the matrix, how to normalize, and how to make the simulation. He proposed the use of poisson distribution for rare events to improve upon the standard counting method and the use of the random sum to generate stock prices. He was also responsible for creating most of the analysis graphs including final distribution histograms, final projection of data, and stock projections. He made the final data about the positive mass of the Markov chain and how to adjust for model overprediction. He worked to analyze the final data in terms of generating the plots and data for varying states, standard deviations, stocks, projection days. Lastly, made improvements to Matt's code and the model itself. The report and powerpoint was split evenly between the two members.

Dataset Selection and Analysis - Discuss the dataset you have selected and all the steps you have taken to process it in order to produce your model parameters. You should also verify that

your processing is correct, by estimating your model's parameters on disjoint dataset partitions (e.g., over different years, etc.) and critically analyze the resulting change.

The data that was used in the model to test its accuracy and analyze the results was stock market price data downloaded from Yahoo Finance. The steps taken to process the data in order to produce the parameters necessary for the model was originally to simply load in each csv file into Python. Since the dataset taken from Yahoo Finance was already clean and in a table with the opening and closing stock prices, the first step was to create a numpy array of the daily opening prices of the stock for each day. Subsequently, the stock was plotted using matplotlib over all the years included in the dataset to verify the processing of the stock data was correct. To process the numpy array into an array that is suitable for the markov chain model, the array of stock prices was put into a function that outputs an array of daily growth rates of the stock.

When loading the other datasets to make sure the processing worked correctly, expected results for the past behavior of each stock were also observed. The obvious difference was that each stock had different prices, and clearly a different numpy array of growth rates. Some companies or currencies had stock data for much longer amounts of time than others, since some of the companies had only become publicly traded recently. As expected, the data was processed correctly as it only created an array and a plot with the number of days it had available. These different lengths of time were taken into account when formulating the Markov Model.

Markov Model Formulation - Discuss the formulation of your model, justify the selection of your state-space and the underlying assumptions. Critically analyze the expected accuracy of your Markov chain in the context of the phenomena you are trying to model.

The first step in formulating the Markov Chain Model is taking the array of stock prices and converting them into an array of daily growth rates. The primary model to create the Markov Chain is a function that requires the inputs of the array of stock prices, the number of days the user wants to look back at the stock price (400,800,1200 days, etc.), the maximum number of standard deviations from the mean for the worst and best states, and the number of states in the Markov Chain. The primary outputs of the function are the Markov Chain matrix, the mean daily growth rate of the past data, the standard deviation of the daily growth rates of the past data, an array of the mean daily growth rates of each state, and the initial state.

The states are defined by the inputs for the maximum number of standard deviations from the mean and the number of states. For example, a matrix that has a maximum standard deviation of 1 and 4 states would have cutoffs at -1, 0, and 1 standard deviation relative to the mean daily growth rate. This matrix would have the most extreme states at greater than 1 standard deviation below the mean daily growth rate and greater than 1 standard deviation above the mean daily growth rate. Likewise, a matrix that has a maximum standard deviation of 1 and 8 states would have cutoffs at -1, -⅔, -⅓, 0, ⅓, ⅔, and 1 standard deviation relative to the mean daily growth rate.

The function then loops through the numpy array of all of the daily growth rates of the past x defined days by the function. The loop adds a value of 1 for the state each growth rate is in based on the previous day's growth rate. Therefore, the counts in the matrix are dependent on the

growth rate of the previous day, as would be expected in a Markov Chain. Once the matrix of counts is created, the counts for most extreme states of the Markov Chain (the lowest daily growth rate and the highest daily growth) are adjusted based on a poisson distribution since they are considered rare events. For example, if the count of the daily growth rate that exceeds 1 standard deviation above the mean is only 1, the poisson distribution may lead to a value of 2. As a result this count of 1 is replaced by a count of 2 in the matrix. Other times this would yield a value of 0, but the expected poisson value was equal to the count which should make it valid over a monte carlo simulation. This was simply to make rare events behave similarly to real world events, and not be as deterministic as counting. After the extreme states are adjusted based on the poisson distribution, each row of the Markov Chain is normalized so the sum of each row is equal to 1.

The result of taking the counts of each state, adjusting based on the poisson distribution, and normalizing the values is the primary Markov Chain matrix. There are some cases where a user entered a very limited number of days of past data to look at or a stock may have only been publicly traded for a certain amount of time. As a result certain states may have never occurred in past data, so the values in that row of the Markov Chain would be 0 in each column. The model accounts for this by setting the probability in each column as 1 divided by the number of states. For example, in an 8x8 matrix this row would result in each probability being ⅛.

Once the primary function creates the output for the Markov Chain, the Markov Chain is put into a function that takes the most recent state and chooses the state for the next day based on the matrix. Then, based on the state chosen, a random sum model of a poisson distribution choosing a number of normal distributions which is run based on the mean daily growth rate of that specific state based on past data and the standard deviation of the daily growth rate based on

past data. The expected value of this was exactly the mean of the state, but the random sum better represented stock returns of a heavy tailed nature, and removed some deterministic attribute of the model. After choosing the growth rate for the next day based on this random sum, the next stock price based on this growth rate is appended to a separate list. This process is repeated for whatever number of days the user wants to track the future behavior of the stock. The result is a graph of future stock prices created using the Markov Chain created from past data.

Model Results - Please list all the results from your simulations. Compare the results with the state-of-the-art or a reference models/implementation, if available/appropriate. Critically analyze your results in light of your model's assumptions.

The results of the model were considered on a few different aspects and features. The first and most obvious was comparing the model to intuition. This is often done with stock data, as shown in "*The (mis)behavior of markets",* which shows an intuitive test for how well the model performed. As it turns out, the model generated data that was generally consistent with the previous returns. This is the shape and peaks seem rather similar to the real stock. However, it missed the linear growth periods and sharp retractions as the real stock data showed. The scale of the simulation varied largely as seen in figure one where the model predicted a 4 times return, whereas the real didn't even double. However for one simulation this kind of variance can be exhibited. Generally from the qualitative and basic look the model generates data rather alike to the real world.

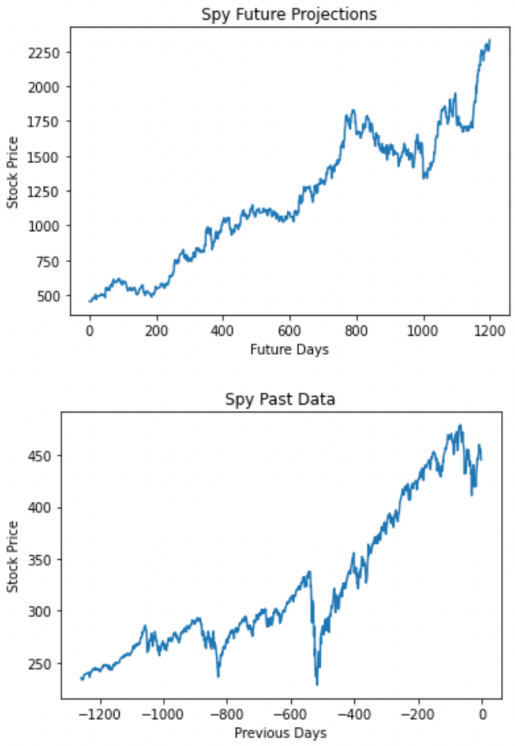

Figure 1. A comparison between a model generated simulation (above) and the real stock return over 1200 day period

Now that the model has some legitimacy of the eye test, now the real results of the model can be analyzed. The first is the main distributions of a large number of trials. Generally, any given simulation of the data is extremely variable. However, by using a monte carlo approach, a characteristic distribution and understanding can be established of the model. There are a few general graphs that can show this behavior. The first one of note is Figure 2. This figure shows that our model tended to over perform the market. The characteristic spread for the model standard deviation is also promising. This is because the deviation should rise exponentially as the model progresses, as the simulations tend to drift from each other. Now considering Figure 3, which shows the histogram of both returns and states. First, it once again shows an overpreformace of the real return. As for the returns, they seem to be log normally distributed,

which is a characteristic of not returns, but of stock prices. [4] The logic for this is that prices can come to 0, but cannot dip below 0, adjusting the probability into a skew. This is most likely due to the assumption built into calculating the return. The return was not the return for any day, rather it was the net return of the simulation divided by the days simulated. This means that it was representative of stock prices rather than the underlying daily return. This feature will be later discussed. As for the time in the states, it is as expected. There are generally more middle states because those are closest to the mean and because the states are selected by standard deviation, so it should happen more frequently. This just proves our model is behaving as expected.

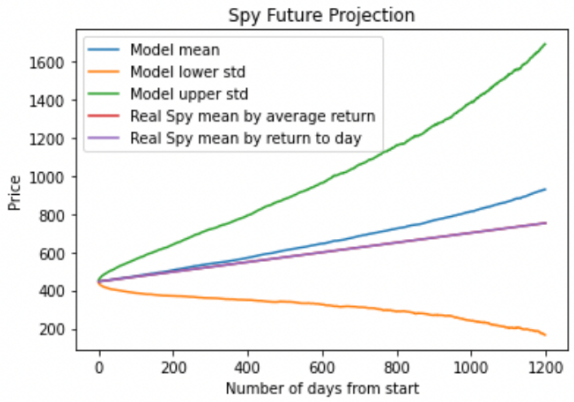

Figure 2. The average monte carlo results of the Markov Model. The model tended to over perform at this projection window.

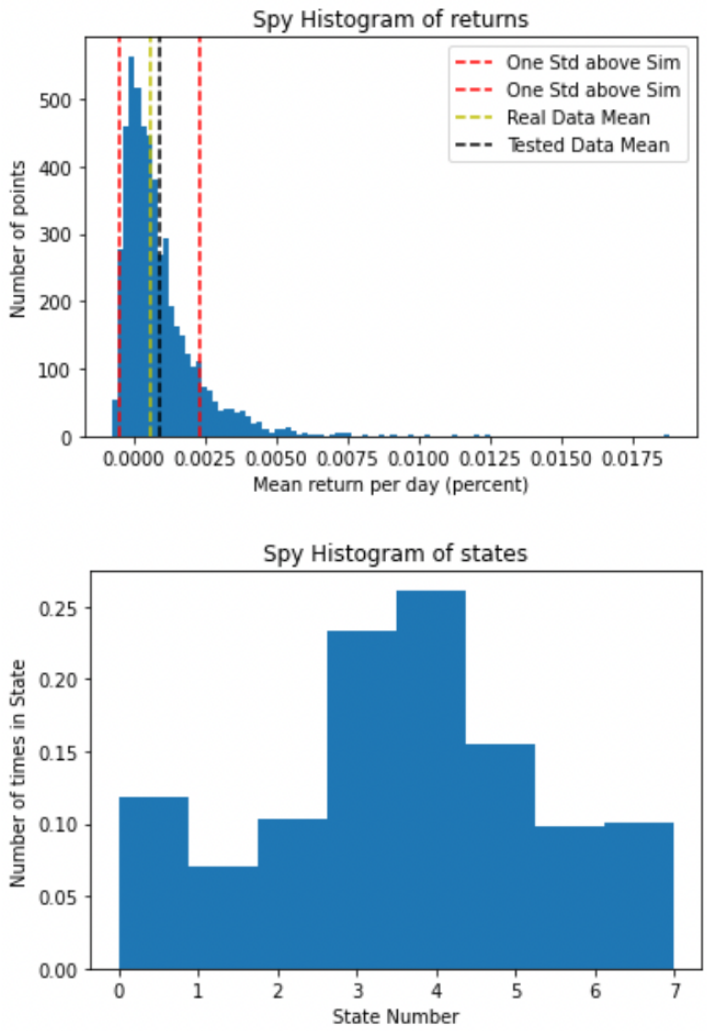

Figure 3. The histogram of the returns for the stock monte carlo (above) and the times in each state for all simulations (below). The model over performed and showed a characteristic log normal distribution.

The next aspect of analysis is the performance of the model over different standard deviations, states, and projection days. The results for these are summarized in table 1 below. The first big note is that these projections were run solely on the spy. The reason is that the model should not deviate significantly for these features stock to stock. This is once again because the states are chosen based on the standard deviation of the underlying. So the characteristic Markov chain should still be rather homogenous for features, such as the three mentioned above. This is not to

say they will be exactly the same, but in theory analysis on one stock should hold for all. The positive probability is graphed below. This positive probability was generated by taking the monte carlo distribution and shifting the return by the difference of the model and the real performance mean. This should account for the over estimation of the model. The distribution is then counted for all positive returns and puts that over the total number of returns. This generates the odds of a positive return. There are no major deviations for any of the trials as the largest maximum deviation was around 4% difference for the std varying trials in the model. This is to say that the model does not vary significantly in any of the three parameters. This suggests the model is rather robust to different conditions. Note that when varying a trial, the standard conditions was std is 1, states is 8, days projected is 1200. So trials 0 of "Number of States" corresponds to trials of std is 1, states is 4, days projected is 1200.

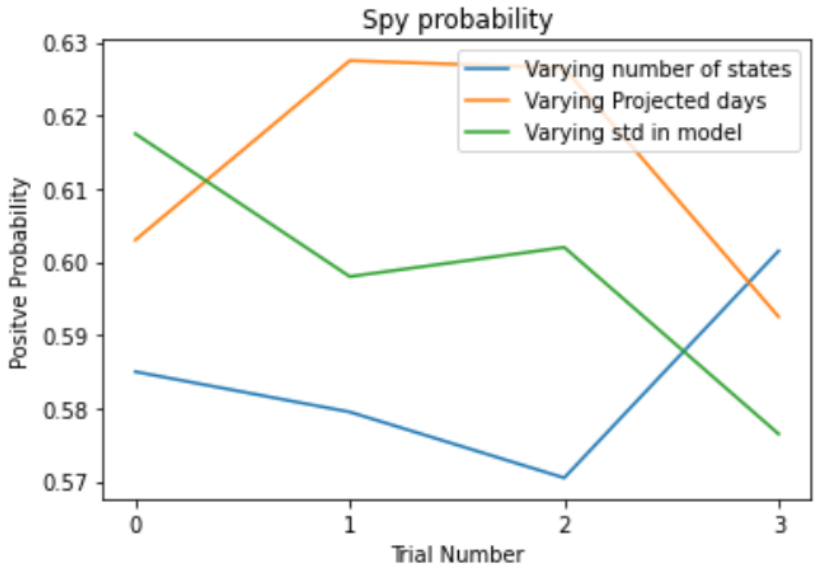

Figure 4. The positive probability pertains to the trial numbers listed in table 2 below. Note these are all independent trials, just graphed together to be concise.

Table 2. Trials for figure 4 above. Standard conditions in red.

Trial Number	Standard Deviation	Number of States	Number of Days
0	.25	4	200
1	.5	6	400
2	1	8	800
3	2	10	1200

An interesting result was also observed in days projected trials. That is the model would still overperform based on the net return of the stock, but would perform below the stock at a given point. This is shown in the 400 day trials in figure 5. It shows that our model may be over performing average, but that may be due to the fact that on the end date of the selection, referring to figure 1, the spy was under performing. This suggests our model may rather quantify the trend of the graph, but not account for sharper drops. This is also because the spy is a reference for the market at large, it would also make sense that other stocks are correlated so drop at a similar time frame.

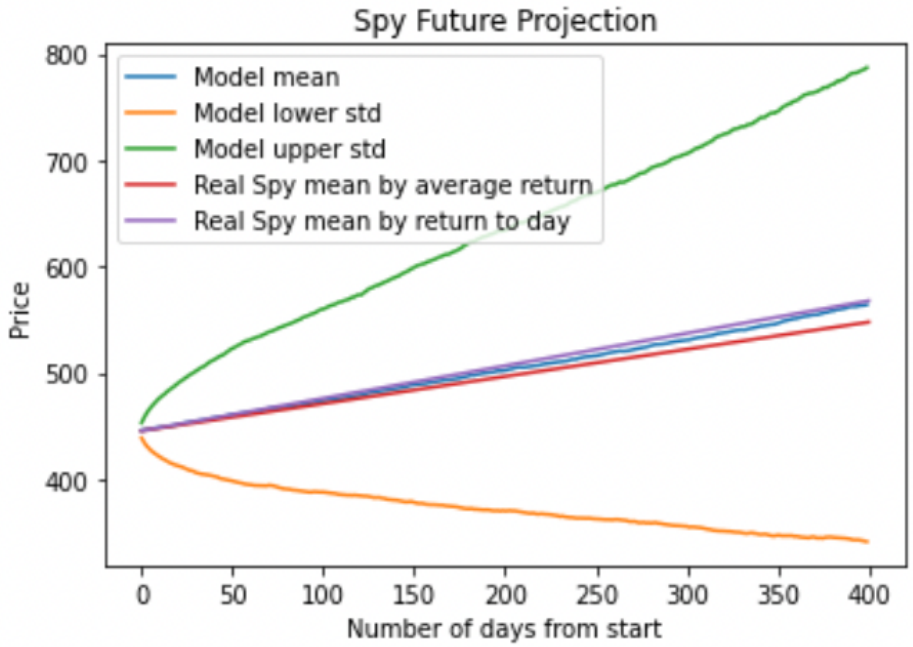

Figure 5. An example of the 400 projected return. Note that the model mean is in between return to day and average return.

The last area of note is the analysis of the positive probability in regards to different stocks. This will be explored in table 3. This suggests that of all the stocks to invest in, the Spy and Coke have the best chance of returning gains, with stocks like Tesla and assets like crypto bode the worst. The other takeaways for this is that related stocks, such as airlines, don't have specifically similar positive returns. Delta has 17% compared to American 9.85%, and Bitcoin nearly doubles Ethereum's chance of positive returns at 12.8% compared to 6.3%. This is not an expected result, but rather may be related to the characteristics intrinsic to the assets themselves. The other note is that stocks that had a high standard deviation had a lower return probability. This can be explained by the fact that our model over estimated return. The higher the standard deviation, the more the model drifts above the mean return, and the more it must be corrected. This will lead to larger corrections and consequently lower positive return probabilities.

Table 3. The positive probability after varying stocks

Stock	Positive return probability
Spy	59.25%
Chipotle	28.9%
Tesla	9.05%
Delta	17.05%
American	9.85%
Coke	49.65%
Bitcoin	12.8%
Etherium	6.3%
Apple	35.9%

Biocryst	8.65%

Possible improvements - Discuss any possible improvements to your model that could improve its accuracy.

There are a few areas where improvement can be made. The first of these is to find a better method to generate returns. As mentioned earlier, the return was based upon the price of the simulations. Instead, the total distributions of returns can be analyzed for all of the simulations. This was considered, but due to memory it was scrapped as an idea. This is because the simulation would generate n trials * projection days which would result in around 2.4 million points of data for most trials. This would have to be generated for every single experiment which would have been rather extreme. However, doing this should give an exactly similar distribution to the original stock return distribution. This would allow more model analysis.

The next area of analysis is to use different distributions to generate the matrix. For example, if one uses a uniform distribution then the matrix would drastically change. This would result in different stages and allow a better analysis of the underlying stock. That is to say there can be an analysis of the long run behavior by use of limiting distributions. One could then compare the distribution for each state and for each distribution used to create the markov chain. Since the model is one that is chosen based on the number of standard deviations from the mean, calculating the eigenvalues to determine the long run behavior or conducting a page rank does not make much sense in context. By comparing the distributions in a different model it would allow us to see what distributions work best and analyze further different stocks from both a model return and perhaps an eigenvalue analysis.

Another area of change could be finding a better metric to account for the over prediction of our model. This was done in practice by shifting the mean of all returns by the over correction. This as mentioned above hurt some stocks more than others. This means that finding another way that scales with standard deviation would be helpful for analysis. One idea is to scale the mean and then normalize by the standard deviation or add a weighting function to help high variance stocks. This should help analysis and help the model to be fair in terms of predicting positive probability for each stock.

The last two areas are just areas for further analysis beyond the scope of what was done in this paper. The first of these is to look into why there are differences in stocks that are very similar. As mentioned above, airlines and crypto behave differently, which was not expected in the slightest. This calls for a further look into why this behavior deviated for the similar stocks. The second area is to find a way to quantify the deviation from the model and why it was deviating. As mentioned, the theory is that the stock dipped at the end of the period, which makes the average return lower. Our model overpredicts because it cannot account for this sudden dip. However, a further and more quantitative method should be explored.

Works cited:

[1] - Kostadinova, Vyara, & Georgiev Ivan (2021). An application of Markov chains in stock price prediction and risk portfolio optimization. *AIP Conference Proceedings*. https://aip.scitation.org/doi/pdf/10.1063/5.0041119

[2] - Mandelbrot Benoît, & Hudson, R. L. (2008). *The (mis)behavior of markets: A fractal view of financial turbulence*. Basic Books.

[3] - Okafor, S. (2020). Markov Chain Applied to Returns on Stock Prices. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3645482

[4]- Zucchi, Kristina. “Lognormal and Normal Distribution.” *Investopedia*, Investopedia, 8 Feb. 2022, https://www.investopedia.com/articles/investing/102014/lognormal-and-normal-distribution.asp.

Image: https://www.nasdaq.com/market-activity/quotes/real-time