Markov Chain Versus Neural Network Asset Class Prediction

Henry Plante
University of Notre Dame
Notre Dame, IN 46556, USA

Instructor: Daniele E. Schiavazzi

Abstract

This paper used both Markov chains and neural network models to predict asset class movements. The models were trained on daily data from indices representing stocks, bonds, and real estate. These models achieved less than five percent error on all asset classes with the neural network performing slightly better.

Keywords: Markov, Neural Network, Assets

1. Background and Motivation

As long as markets have been around people have been trying to understand them in order to exploit them for profit. People are constantly attempting to find new indicators as to how a market will move. Finding something that gives the trader an edge in the marketplace allows them to reap massive rewards, think hedge funds and high frequency trading firms.

This paper will compare the capabilities of a Markov chain model and a neural network model in predicting asset movements. Indexes for stocks, bonds, and real estate will be used in an attempt to understand how the three main asset classes move. Asset classes are used instead of individual securities as asset classes are a bag of securities and by that nature tend to exhibit less volatility, and the hope is that the models will be more accurate with out as much volatility.

A Markov model and neural network were trained with data from 2013-2021 and then evaluated on data from 2022. The accuracy of both models was compared and analyzed for effectiveness.

Previous work by Lee (Lee and Komeiji, 2022) and Burnett (Burnett, 2021) is similar to this paper as they attempt to use Markov models to predict price movements, however, this paper differs in its use of a different dataset, different Markov model states, and by the comparison to a different type of model. Both of these papers used individual securities rather than asset classes. They showed a limited capacity of Markov chains to accurately model the stock market. The Burnett paper attempted to rank industries based on returns and was able to achieve just better than random chance predictions. The Lee paper attempted to create a portfolio of stocks based on a Markov model and PageRank. Their best portfolio returned 1.33x profit while the overall stock market more than doubled in value during the same time period.

2. Team

The team consisted of Henry Plante. Henry did all parts of the project.

3. Dataset Selection and Analysis

The dataset consisted of the daily closing prices of the index of each asset: stocks, bonds, and real estate. Indexes are ways to describe the worth of the entire asset class at once. The Standard and Poor 500 index was used as the data for stocks. This index tracks the worth of the 500 largest US stocks which is often used to describe the state of the entire market. The IShares Core U.S. Aggregate Bond ETF was used as the data for bonds.This ETF attempts to track an index of U.S. investment grade bonds. Investment grade meaning low risk and usable as fixed income sources rather than for speculation purposes (how bonds are normally used). The IShares ETF was chosen because of the availability of years of historical data. The Dow Jones U.S. real estate index was used as the data for real estate. This index tracks the performance of publicly traded real estate investment trusts in an attempt to provide an understanding of the entire U.S. real estate market. This index was chosen as it had accessible historical data and was provided by a reputable financial firm.

The real estate data goes back the shortest and is the limiting factor in the amount of data for this paper. The final dataset ranges from 1/1/2013 - 12/30/2022 giving 10 years of daily historical data or around 2300 data points. This is not an ideal amount of data, but due to the lack of publicly available asset pricing data is all that will be used in this paper. Because of the lack of data a 90/10 training testing split was used in an attempt to get as much training data as possible so that the Markov transition probability matrix would be as accurate as possible. This means the years 2013-2021 were used as training and 2022 was used as testing. It is important to note that the training data was almost entirely upward trending while the testing data was on a downward trend. The difference in market conditions between the datasets likely had an impact on the accuracy of the model. The training and testing sequences stocks can be seen in figures 1 and 2 to see this effect.

4. Model Formulation

4.1 Markov Model

The construction of the Markov model starts with state definition. A previous paper by Jimmy Burnet used the idea of an asset being in a bear or a bull state. Bear meaning having decreased in value the previous timestep and bull meaning having increased in value for the previous timestep. Drawing on this, this paper will define an asset to be in 6 different states based on how many days in a row the asset has been in a bear or bull state: 3 or more days bear, 2 days bear, 1 day bear, 1 day bull, 2 days bull, 3 or more days bull. The combination of 3 assets each with 6 possible states leads to $6^3 = 216$ distinct states.

To create the transition probability matrix necessary for a Markov chain a 216x216 matrix was initialized. A forward walk on the training data was then done. At each timestep the entry corresponding to transitioning from the current state to the next was incremented. To make the model probabilistic each value was divided by the sum of its row.

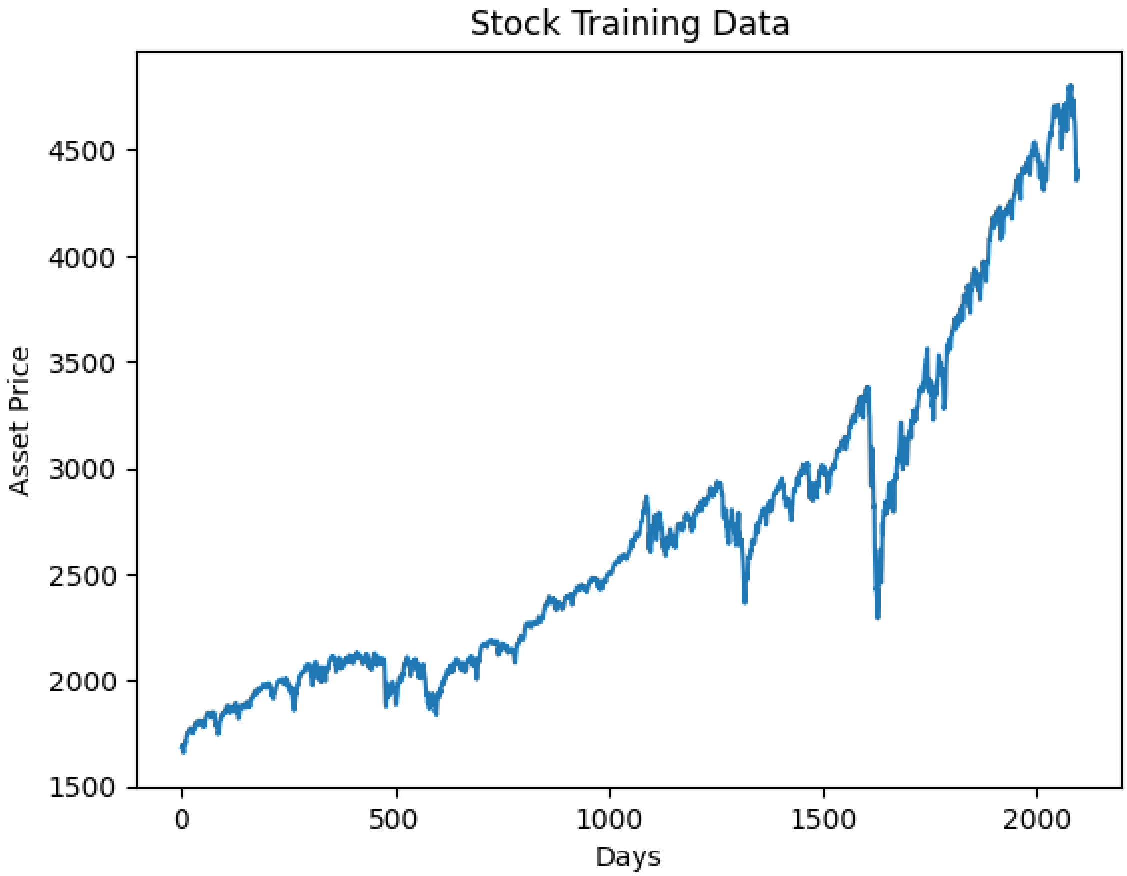

Figure 1: Stock Training Data

There was one row state that was never reached so the row corresponding to its transition probabilities was populated with the uniform distribution.

During the forward walk, the values of the percent price change for each asset at each state was saved, so that a mean and standard deviation of percent change could be calculated for each asset in each state. Price changes are assumed to be normally distributed as returns for assets usually have a normal distribution (Ozyasar, 2023). The state that was never reached was given the standard deviation and mean of the training dataset.

4.2 Neural Network

To create a comparison to the Markov Model a Neural Network was trained. The input to this network was three sequences of prices for 20 days, a month worth of trading days. 63 days, a quarter of yearly trading days, was also tried and yielded similar accuracy. Each input sequence corresponds to a different asset class. The training output was the percent change in the three asset prices after 10 days from the last input datapoint. A convolution network was used as it is capable of learning from series data. A recurrent neural network

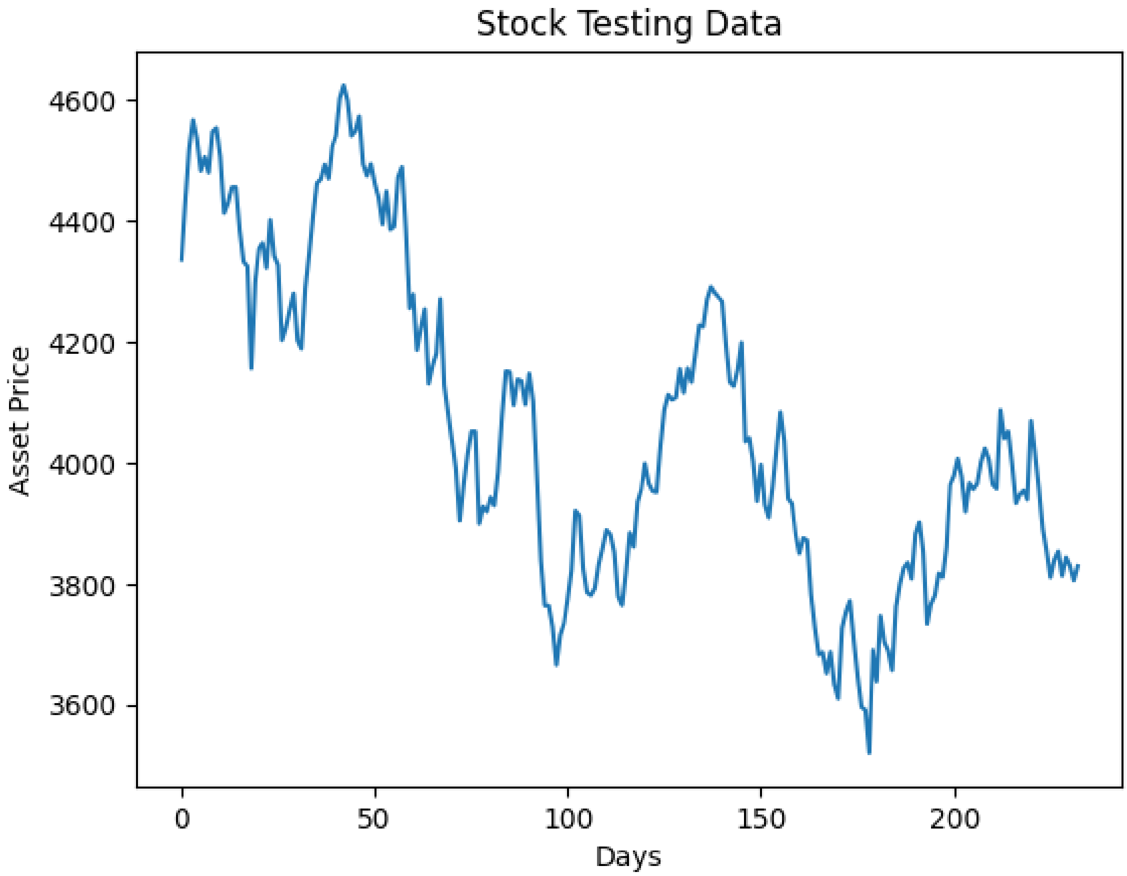

Figure 2: Stock Testing Data

could also have been used but the author had more familiarity with convolution networks. The network architecture is as follows. There are 5 convolution layers. The first three layers had kernel size of 3 and the last two had size 5. All layers used a stride of 1. The number of channels for each layer was 20, 25, 30, 35 , 40 respectively. Five layers were chosen in an attempt to learn high level features while not adding too many weights given the limited training size. Following the convolution layers are three fully connected layers of size 256, 128, 3 respectively. The activation function for each layer is ReLU to prevent vanishing gradients. Tanh activation was tried on the fully connected layers but the network failed to train at all. Dropout with a dropout rate of .3 was used after each fully connected layer in an attempt to improve training. Pooling was not used as the sequences were very short and there was no need to reduce dimensionality. The network was trained for 50 epochs using Adam optimizer and mean squared error loss.

5. Model Results

5.1 Testing Methodology

5.1.1 Markov Chain

To test the Markov chain, the testing data was split up into 23 distinct 10 day intervals. 10 day intervals correspond to two weeks of trading days (days that assets change price) which was the chosen time period to predict. 23 comes from how much validation data was available. For each time period and for each asset type a simulation was performed. The Markov model started off in the true state. Then by sampling from the transition probability matrix a new state was chosen for the next time step. Based on this new state another new state was chosen for the next timestep. This was done for the length of the time period, 10 times. At each timestep a sample was taken from the normal distribution created from the mean and standard deviation of the returns for transitioning to that state for that asset which was calculated during the formation of the Markov model. Sampling from the mean and standard deviation of the returns for transitioning away from that state was also tried with similar results. This gave a percentage change in the price of the asset for that timestep. These percentage changes were applied repeatedly to the actual starting price of the asset resulting in a prediction of the asset price at the end of the time period. This simulation was done 100 times for each time period for each asset type and the mean value of these simulations was taken as the Markov model prediction.

5.1.2 Neural Network

To test the neural network, sequences of data from the testing data were fed into the network. The data points chosen correspond to predicting the same periods as the Markov chain. Comparison of the predictions for each asset class seen below.

5.2 Simulation Results

Simulation results can be seen in figure 3, 4, 5 for stocks bonds and real estate respectively.

5.3 Comparison

The the mean relative error of the predictions abs((actual - predicted) / actual) was used to determine the accuracy of each model. The results are shown in table 1 and 2 for testing and training respectively.

Model	Stocks	Bonds	Real Estate
NN:	.041	.011	.037
MC:	.044	.015	.039

Table 1: Mean Relative Error Testing

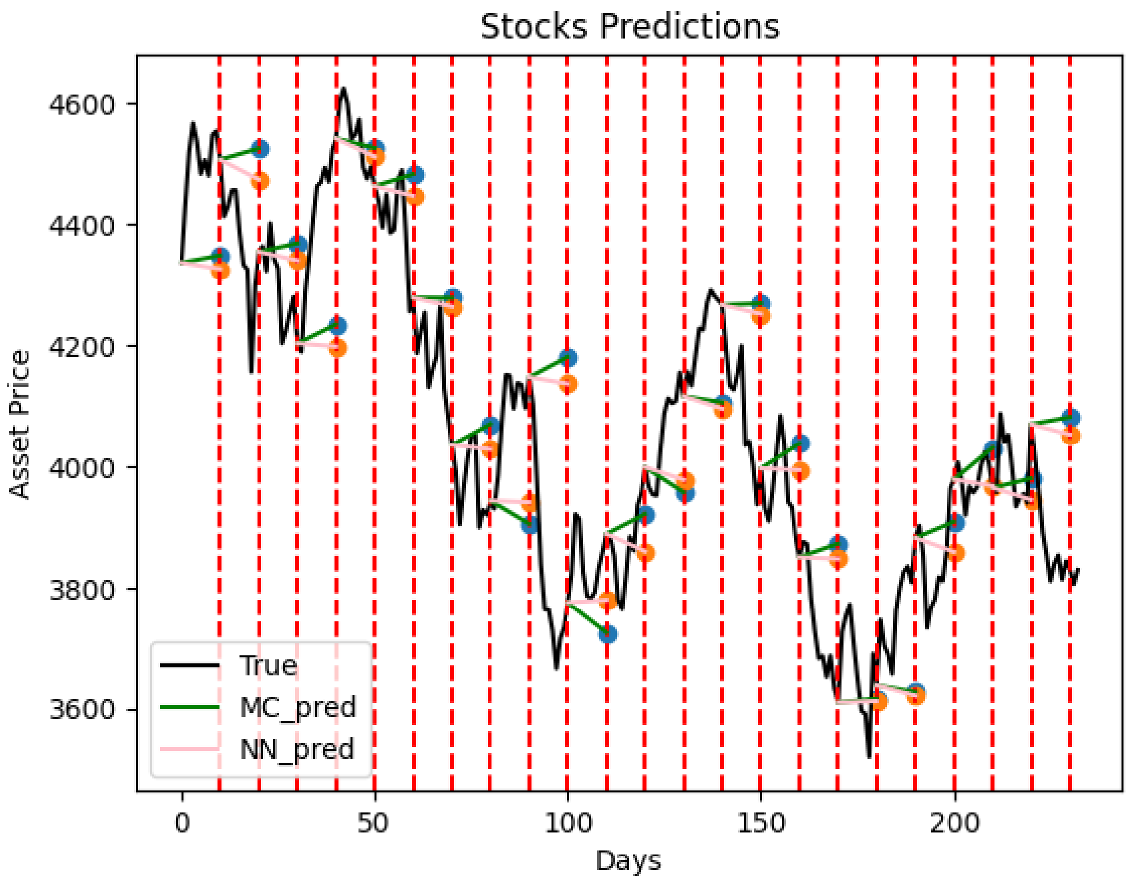

Figure 3: Results of Simulation for Stocks.

Model	Stocks	Bonds	Real Estate
NN:	.019	.005	.023
MC:	.020	.008	.024

Table 2: Mean Relative Error Training

5.4 Discussion

The testing results mean that both models' average only a 1 percent difference in their predicted price for bonds compared to the actual price two weeks into the future. For stocks and real estate it is around 4 percent.

For all assets, the neural network performed slightly better. This may be due to the ability of the neural network to utilize a sequence of states and not rely solely on the current state. The neural network is also able to understand high level features from the data that might inform the change in price better than a singular state like what is fed into the Markov chain would.

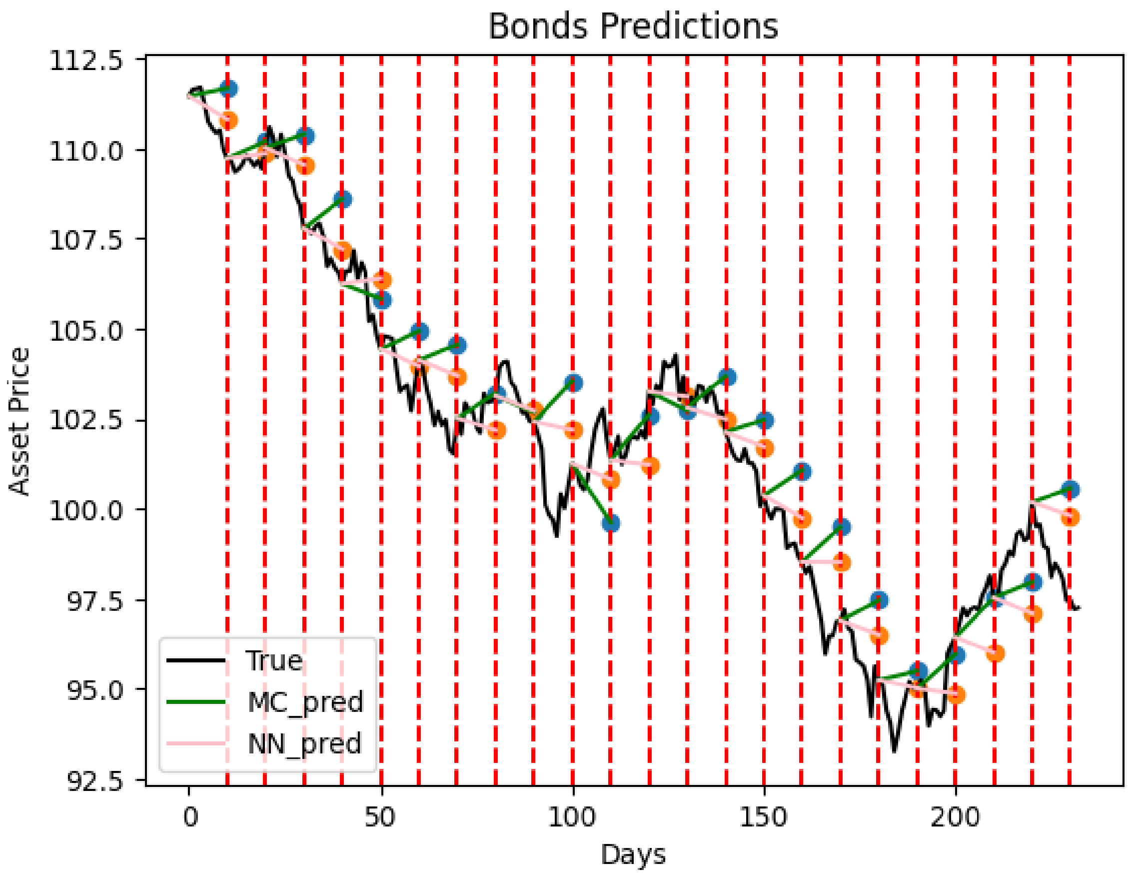

Figure 4: Results of Simulation for Bonds.

Both models performed better on the bond data than on real estate or stock data. This may be for two reasons. Bonds are much less volatile than the other two assets. Over the course of the testing period the bonds changed by 12% while the real estate and stock assets changed by 20%. Looking at the graphs you can also see the larger swings in price from the stocks and real estate compared to the bonds. Another reason why bonds may have performed better is that they tend to be speculated on less than the other assets. The other assets may not respond to overall market conditions (such as those described in the state of the Markov chain) as well as bonds as there may be hype surrounding the asset that doesn't take into account the market conditions. The bonds, however, which don't tend to primarily be an investment tool rather than a speculation tool, as explained previously, may react in a more predictable way to market conditions.

While relative error is only a few percent, these models tend not to predict large price changes accurately. For the Markov chain, this is partly due to how each prediction is the mean value of 100 simulations. If one simulation was used the predictions for some timesteps would have much larger changes. The neural network likely predicts small changes because the limited data it was given doesn't give it a signal of when a large price change will

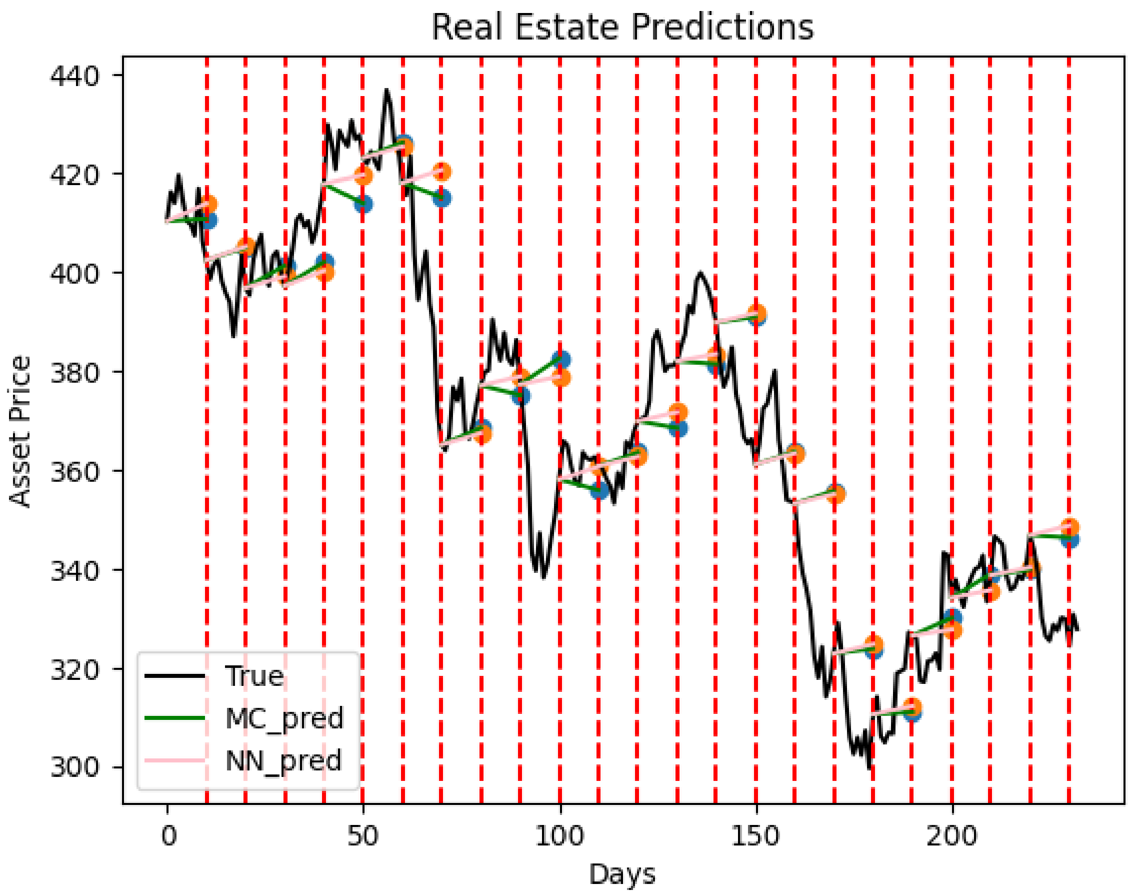

Figure 5: Results of Simulation for Real Estate.

happen. While the models have some ability to predict the way an asset is moving, they shouldn't be expected to predict exact price values.

Notice the large discrepancy between the training and testing errors. The errors for the training data are roughly half that of the testing. This is likely because the majority of the training data was on an upward trend while the testing data was during an unusual downward trend. This shows the lack of generalization abilities of these models. They work much better for a trending upward market like they were trained on but not as well for a downward trending market.

These results continue show the limited, yet still possibly useful, capacity of Markov chains to model the stock market from the previous papers mentioned.

6. Possible improvements

As discussed before, one reason why these results might not be as good as possible is because of the training and testing data selection. The majority of the training data between 2013-

2022 was during a time that the market was generally increasing in value. However the testing data during 2022 was an unusual case where the market was actually trending downward. This may have caused the models to perform worse than if they were tested during more similar to training market conditions. Breaking the training and testing data to make sure they both include downward trending markets and upward trending markets may be a way to increase the ability of these models.

Attempts like the two models discussed, that try to use only pricing data to predict price movements, are called technical analysis. While new technical analysis strategies do have merit in predicting price movements, strategies known to the public are understood to not be very capable indicators (Malkiel, 1973). This is because if everyone knows about it everyone will start to trade based on it and it will stop working. The simple neural network and Markov chain approach are very well known ways to attempt to model things like the stock market, so it is not unexpected that the models were not perfect.

To take these models beyond technical analysis (or into the realm of not yet discovered technical analysis), adding additional information to these models outside of asset price may be a way to increase their predictive capabilities. Things describing the state of the economy such as exchange rates and inflation rates could be added to these models. However, adding this additional data to the Markov model would result in an extremely large transition probability matrix. Given the small amount of data it may not be feasible to accurately populate the matrix. A neural network will likely be able to incorporate this additional data better, however it will also suffer given the limited amount of data. Models like the ones in this paper need lots of data to be useful. The problem this paper was solving only made use of one sequence of daily data stretching back 10 years. To be able to make use of additional data one could attempt to predict the price of individual assets, as opposed to trying to predict the prices of indices. This could increase the dataset 1000 times. Essentially, you would be able to break the indices into their individual components and using each of them independently.

To improve the neural network model, a different architecture could be used. Professor Schiavazzi recommended a sequence to sequence LSTM network. Using a recurrent neural network architecture like this might lead to better accuracy on the kind of data used in this project do to its sequential nature. However, I had tried using an RNN for essentially the same project for a different class and the network would always train to output the same value no matter the input and no matter how I trained it, so it is unclear if this would end up making a big difference.

References

Jimmy Burnett. Using markov chain modeling for stock market sector performance. 2021.

John Lee and Trevor Komeiji. Investment portfolio modeling using markov chains and pagerank. 2022.

Burton. G. Malkiel. *A Random Walk Down Wall Street*. Norton, New York, 1973.

Hunkar Ozyasar. Why are stock prices considered log-normal? 2023.