Bitcoin Closing Price Predictions

Ethan Vanlerberghe

School of Professional Studies, Northwestern University

MSDS 422: Practical Machine Learning

Dr. Chaturvedi

8 December 2024

Business Understanding:

For my final project, I wanted to research something that would be relevant to me and what I do every day. While thinking about different ideas, I saw a news article discussing how Bitcoin prices are expected to rise to $100,000 per coin by the end of 2024[1]. I had remembered being introduced to Bitcoin in middle school, and being confused on how exactly it worked. Then, while I was in college, there was a boom in cryptocurrency trading, causing the price of bitcoin to skyrocket. Remembering all of these, and that I still held a minor amount in Bitcoin, I became interested in trying to predict the value of bitcoin, and to see when or if it would cross the $100,000 mark.

Bitcoin was created in 2008 by an anonymous developer (or developers) under the name 'Satoshi Nakamoto'[2]. Since its release in 2008, it has quickly become the most popular cryptocurrency in the world. Due to the nature of cryptocurrencies being loosely regulated, they are one of the fastest growing markets in the world, and thus, a very popular item to invest in[1]. By applying practical machine learning techniques to publicly available Bitcoin pricing data, an individual can predict what the closing price of Bitcoin will be on any given day. Using this technique, a user could more intelligently invest their money with the hope of a better return on their investment. A successful output of this project would be to predict the closing price of Bitcoin over the past five years, and use it to project the daily closing price of the coin, and when, if ever, it would hit $100,000 per coin in value.

I began by looking for a clean data source to pull data from. In my undergraduate work at Oakland University, we would often pull datasets from Yahoo Finance and work to analyze the trends being seen in various stock prices over the years. I decided to pull the five year rolling data from Yahoo

---

[1] **Fulton, RJ. "Prediction: Bitcoin Will Hit $100,000 before the End of the Year." The Motley Fool, August 29, 2024. https://www.fool.com/investing/2024/08/29/prediction-bitcoin-will-hit-100000-before-the-end/.**

[2] **History.com. "Satoshi Nakamoto Publishes a Paper Introducing Bitcoin | October 31, 2008." A&E Television Networks, October 29, 2009. https://www.history.com/this-day-in-history/satoshi-nakamoto-publishes-a-paper-introducing-bitcoin.**

Finance, and use it as my starting point for this project[3]. This dataset includes the date, open price, high price, low price, close price, adjusted close price, and volume of the coin every day from December 5, 2019 to December 5, 2024 (See Figure 1).

Exploratory Data Analysis

After importing the data, I first wanted to see if there were any null values or additional non-needed columns. All of my columns show 1827 non-null rows, meaning that every row has a value. All columns are of type 'Float64' except the 'Date' column which is of 'Date' type. All of the open, high, low, close, and adjusted close metrics are captured in US Dollars (See Figure 1). Then, in order to visualize the goal, I decided to plot the close price over time (See Figure 2). By predicting the close price, I can get a better understanding of when the overall price of the coin will cross over the $100,000 threshold. Looking at Figure 2, the closing price has a peak in 2021, another peak in 2022, a lull in 2023 and then rockets up towards the end of 2024. This climb to the top near the end of 2024 gives the impression that the article was correct, and that Bitcoin may hit the $100,000 mark by the end of the year. In order to see just how close the coin has come to the $100,000 mark, I used the describe feature to view the max and min of every value. As of December 4, 2024, the peak value of the coin was $98,997. To examine the closing prices closer, I decided to create a boxplot with the closing price over the past five years. The median closing price is $30,548.70, the mean closing price is at $34,970.53, the upper bound (75th percentile) is at $49,031.46, the lower bound (25th percentile) is at $19,244.45 (See Figure 10). These plots help develop a better understanding of where the historical price of Bitcoin has been, and how only at the end of 2024 is the coin's value per coin approaching the $100,000 mark.

Data Preparation/Feature Engineering

Due to the relatively clean nature of the data produced by Yahoo Finance, there is relatively less to do in terms of data preparation. First, I split the data into an 80/20 test and train set based on date. By

---

[3] **Yahoo Finance. "Bitcoin USD (BTC-USD) Price History & Historical Data." Yahoo Finance. Accessed December 8, 2024. https://finance.yahoo.com/quote/BTC-USD/history/.**

first sorting on date, I can force the model to train itself on the first 80% of the data chronologically, then test itself on the last 20%, providing a better real-life scenario for the model. After the split, I used the minmax scaler to scale the data for time. Once the data had been split into the test group, the train group, and scaled, it was ready to begin modeling.

Methodology and Tools

I began by running model alpha, a simple recurrent neural network (RNN) using the adam optimizer and the mean square error measure for my loss scoring. I chose to start with an RNN due to their ability to analyze time-series datasets, and the price of bitcoin is a great example of a time-series dataset. RNN's are recurrent, meaning that previous information can impact future information, which is exactly the case when trying to predict stock price. I then trained it with the training dataset over 20 epochs with a batch size of 32. For this, I used Google Colab's cloud platform, running with a CPU accelerator. After model alpha, I did more research on the best type of model to use. I decided next to try a Long Short-Term Memory (LSTM) model. LSTMs are very similar to RNNs, as they are a type of RNN, however LSTMs have memory gates that allow them to better reflect past information and its impact on future predictions than an RNN[4]. In addition to converting to an LSTM model for model beta, this time I also added more epochs to try and reduce the loss from the first model. The second model used 50 epochs with a batch size of 64. For model gamma, I tried further increasing the epoch count to even further reduce the loss. I increased the number of epochs to 100, and left the batch size at 64. For my final model, model delta, I kept the epoch size the same at 100, but tried increasing the batch size to 128 instead of 64[5].

---

[4] **Idrees, Hassaan. "RNN vs. LSTM vs. GRU: A Comprehensive Guide to Sequential Data Modeling." Medium, July 5, 2024. https://medium.com/@hassaanidrees7/rnn-vs-lstm-vs-gru-a-comprehensive-guide-to-sequential-data-modeling-03aab16647bb.**
[5] **Chaturvedi, Anil. "MSDS 422 - Practical Machine Learning." 2024. Evanston, IL.**

Model Evaluation

Model Alpha, utilizing 20 epochs and a batch size of 32, resulted in a loss score of 0.0031, and 0.0025 for a value loss score (See Figures 3,4). These loss scores are measured in mean square error. To try and reduce these loss scores, I moved forward with hyper parameter tuning, where for model beta I increased the epoch count to 50, and increased the batch size to 64. Model Beta was also the point where I switched from using RNN models to LSTM models. This combination of changes resulted in a loss of 0.0015, and a value loss of 0.0049, a slight decrease in loss from model alpha (See Figures 3,5). After reviewing the performance from model beta, I revised model gamma to include 100 epochs, but the same batch size of 64. This new method took much longer, however delivered a loss score of 9.863e-4, and a value loss of 0.0017 (See Figures 3,6). To try and reduce the loss even further, for model delta, I kept the epoch size the same at 100, but increased the batch size to 128. This caused the loss to 0.0015 and a value of 0.0016 (See Figures 3,7). Although model delta was very accurate, due to the larger batch size and high epoch number, it took even longer to run than model gamma. Due to the advantages gained with the 100 epoch setting, as well as utilizing the abilities of an LSTM, I chose model gamma to use for the end modeling.

Findings and Conclusions

Reviewing the models that were created, model gamma, the LSTM model with 100 epochs and a batch size of 64 has the lowest loss values, and therefore performed the best of the developed models. In order to put the model to the test, I pulled in data from up to present day where the price of bitcoin had crossed the $100,000 threshold. When running model gamma against the actual closing data, I was disappointed to see that the model did not indicate the coin ever crossing the $100,000 mark. Instead, the model gets up to $95,438, but does not go beyond that (See Figures 8,9).

Overall, model gamma does provide a strong indication for the business case. The model was able to accurately predict Bitcoin's 5-year trend within $5,000 of its value. If an investor were able to use this model, they could get an indication, within $5,000, of where the value of the coin would go that day, and be able to make the decision if it is better to sell their position on the stock and/or coin or to hold their position.

Lessons Learned and Recommendations

If I were to repeat a project similar to this one, I would begin by straight away exploring the LSTM models. While running the first RNN model helped me to better understand the advantages of the LSTM, it would be better to go straight into using LSTM and focus more time on hyperparameter tuning. There are also various other models that could be used to compare to an LSTM to see if they would be better indicators of performance, such as a convolutional neural network (CNN).

In order to better improve the accuracy of the model, I would also bring in a dataset from a social media platform (Possibly X), where individuals can discuss the stock or cryptocurrency freely. By pulling in that data, I could analyze the sentiment of what everyone is saying about the stock or cryptocurrency, and use that information to help the model follow the trend even closer. For example, if there was a large discussion occurring on X about the excitement of Bitcoin’s value that day, with a lot of positive-sentiment messages being posted, the model could assume a stronger day than originally thought for the coin. I would also look into building my models using a GPU accelerator instead of a CPU accelerator, to see if there is any additional speed gained from using the different accelerator.

In order to correlate this project more directly with the investment world, I would build a model across more individualized stocks, not just the value of a single stock or cryptocurrency. While there is still earning potential in something like studying the price of bitcoin, the true value comes in being able to jump in on a brand new coin or stock that shows the same trend as those that are known to be extremely successful.

Figures

Figure 1: Exploratory Data Analysis on dataset from Yahoo Finance.

```
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Date        1827 non-null   datetime64[ns]
 1   Open        1827 non-null   float64
 2   High        1827 non-null   float64
 3   Low         1827 non-null   float64
 4   Close       1827 non-null   float64
 5   Adj Close   1827 non-null   float64
 6   Volume      1827 non-null   int64
```

Figure 2: Bitcoin closing price over the past five years.

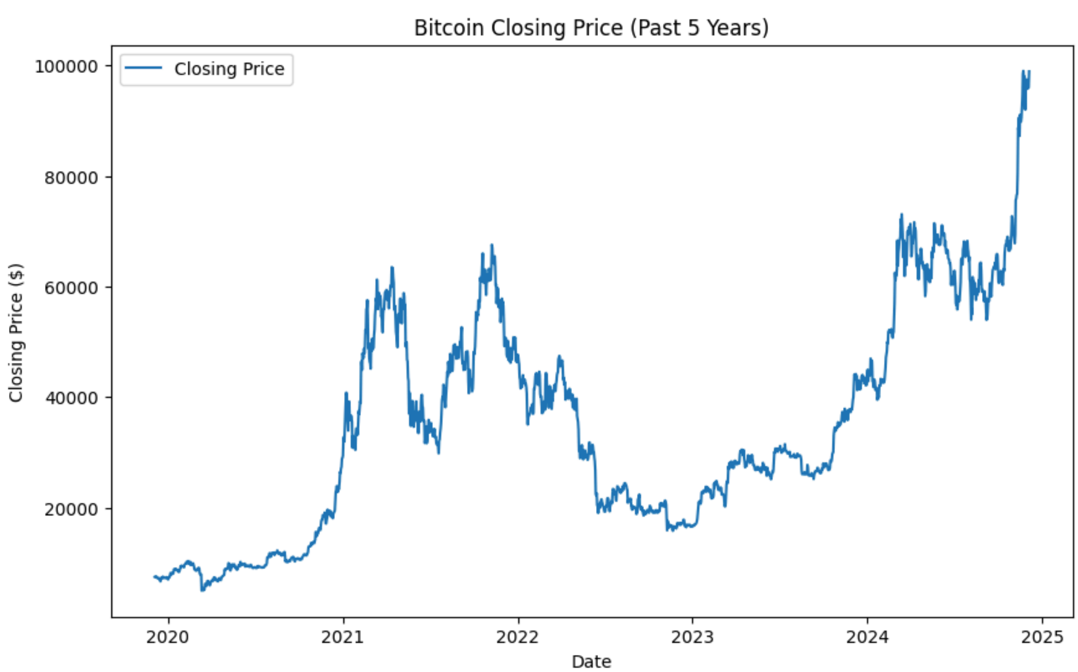

Figure 3: Chart of the performance metrics for the four tested models.

Model	Model Type	# of Epochs	Batch size	Loss (MSE)	Value Loss (MSE)
Alpha	RNN	20	32	0.0031	0.0025
Beta	LSTM	50	64	0.0015	0.0049
Gamma	LSTM	100	64	9.863e-04	0.0017
Delta	LSTM	100	128	0.0015	0.0016

Figure 4: Performance of model alpha vs. the actual closing price.

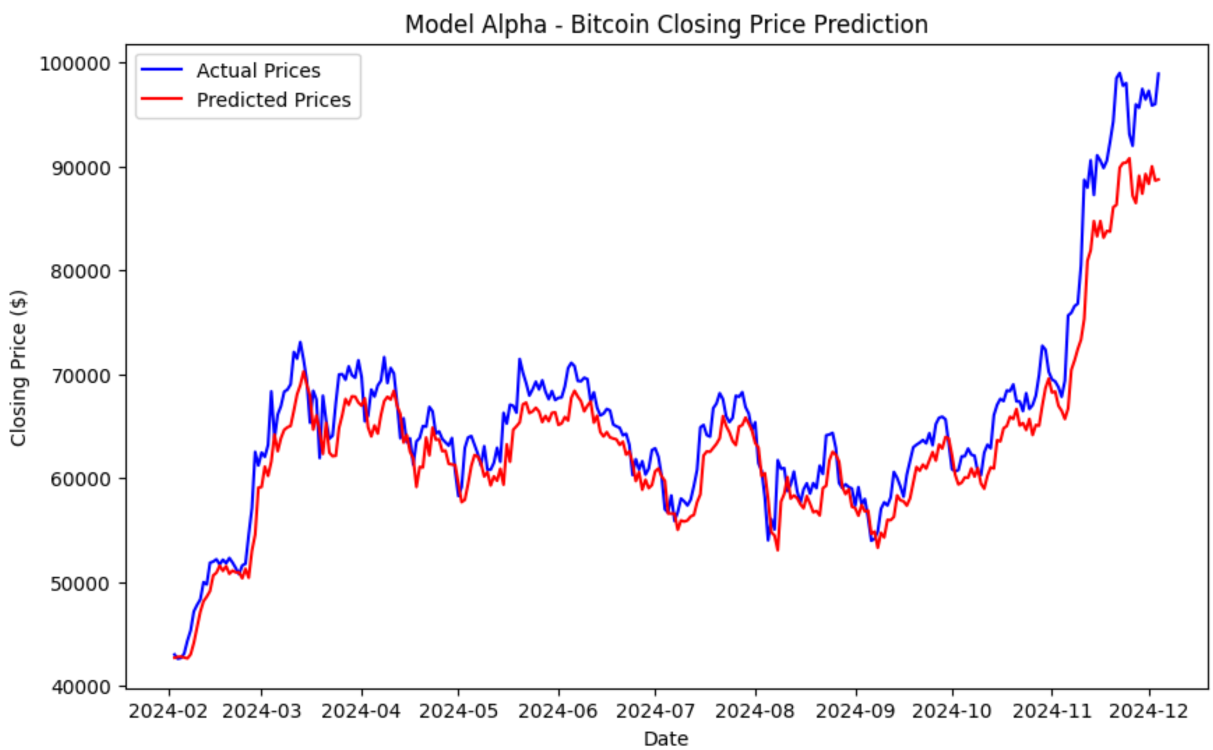

Figure 5: Performance of model beta vs. the actual closing price.

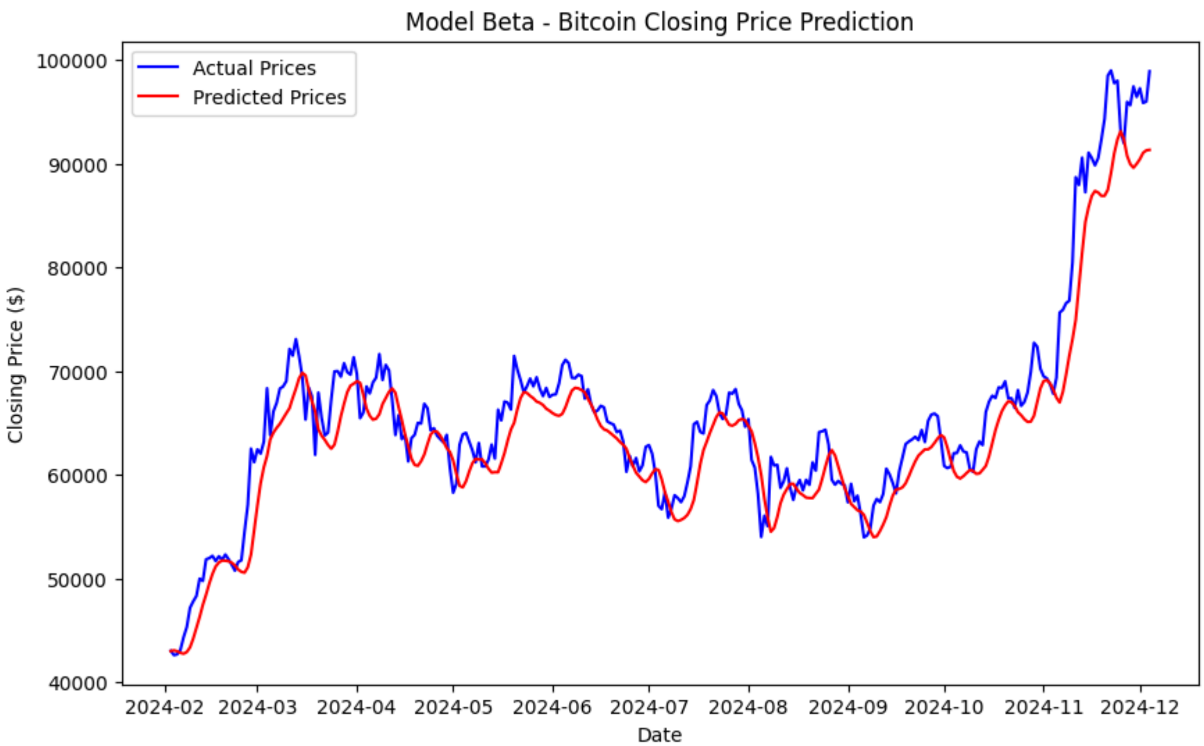

Figure 6: Performance of model gamma vs. the actual closing price.

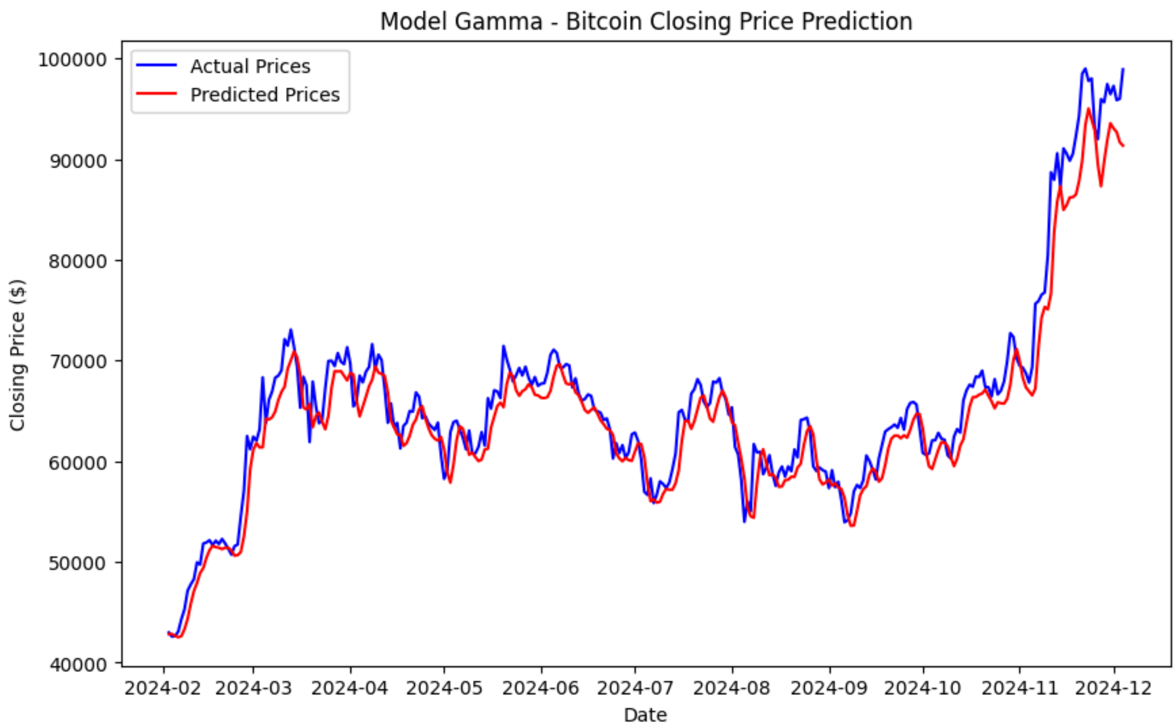

Figure 7: Performance of model delta vs. the actual closing price.

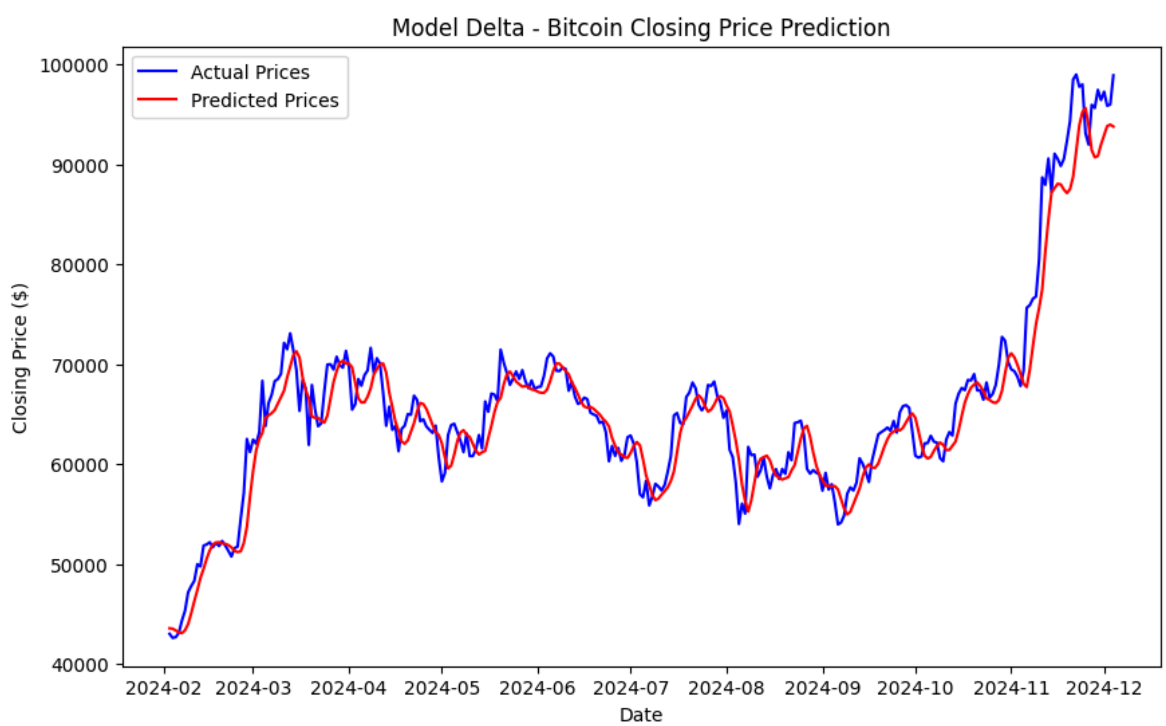

Figure 8: Performance of updated model gamma vs. the actual closing price.

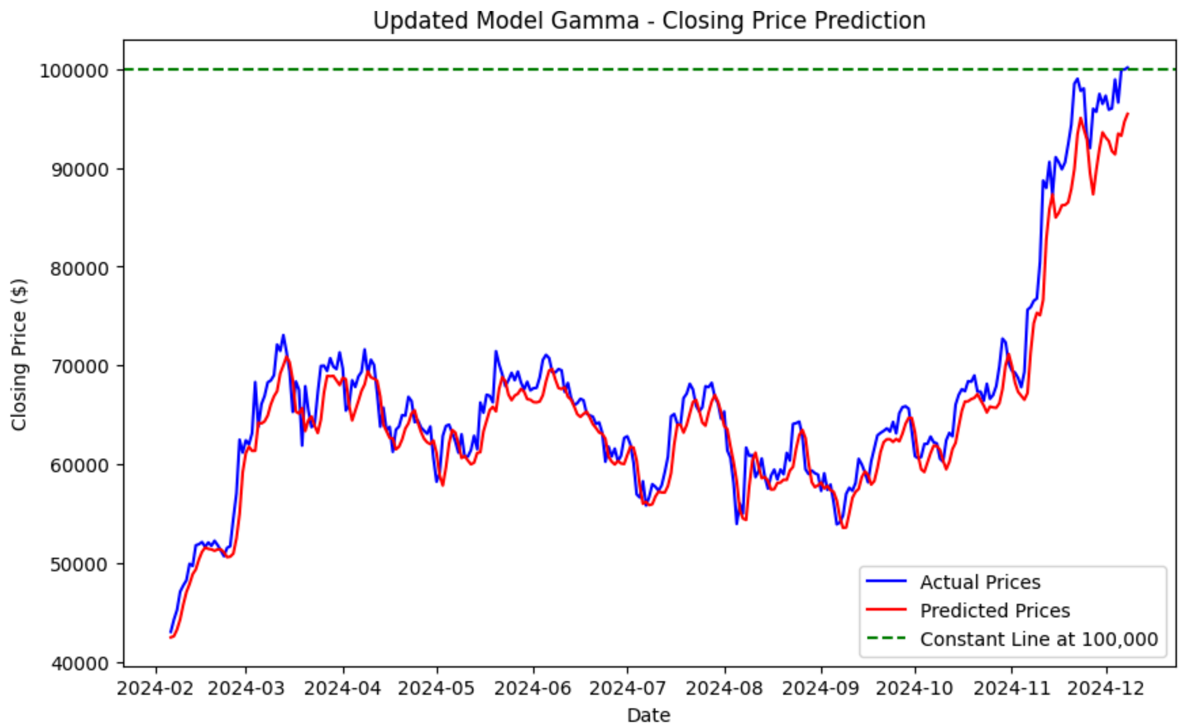

Figure 9: Updated model gamma predictions for closest dates to actual $100,000 mark.

	Date	Actual Close Price ($)	Predicted Close Price ($)
6	2024-12-02	95865.30	92695.695312
5	2024-12-03	96002.16	91682.429688
4	2024-12-04	98918.27	91355.101562
3	2024-12-05	96593.57	93442.828125
2	2024-12-06	99920.71	93236.210938
1	2024-12-07	99923.34	94656.742188
0	2024-12-08	100121.52	95438.164062

Figure 10: Exploratory boxplot of Bitcoin closing price.