Module CS3AI
Student Number: 29021144
Date: 24/11/2023
Hours spent: 20

Abstract
This study explores the comparison of two distinct methodologies in time series forecasting [1], focusing on the highly volatile and nuanced asset that is bitcoin [2]. We adopt a Long Short Term Memory (LSTM) network [3], a recurrent neural network (RNN) based on advanced deep learning [4] techniques against Meta's Prophet [5], an accesible tool designed for time series forecasting that can handle trends, seasonality and holidays effectively.

Background
The landscape of cryptocurrency and bitcoin is an ever evolving field with prospects for increasing mainstream adoption and potential for significant regulatory change [6]. A notable development is the prospect of Bitcoin Exchange-Traded-Funds (ETFs) gaining approval [7]. The creation of such ETFs signifies progress toward integrating bitcoin and cryptocurrencies into traditional financial markets, likely increasing liquidity and accessibility for a larger range of market participants. Such a transition attracts attention from investors and underscores the need for accurate methods to predict future trends and volatility in Bitcoin prices.

Bitcoins inherent volatility, driven by factors such as market sentiment, regulatory news, technological advances and unique detachment from traditional economic factors presents a unique challenge in financial forecasting thus necessitating innovative approaches to predictive modelling [8]. Not only will this report determine the most effective modelling technique for predicting the price of Bitcoin but also to highlight the importance of data pre-processing and dimensionality reduction in achieving high prediction accuracy.

EDA
Exploratory Data Analysis (EDA) was conducted to gain insights into the data. In this study we used the daily historical values of the bitcoin price located in BTC-USD.csv [9] which is freely available from Yahoo Finance. This contained 3,352 entries representing daily Bitcoin price data marked in USD including seven columns: 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', and 'Volume'. The data types are mostly floating point numbers, with date (string) and volume (integer) as exceptions.

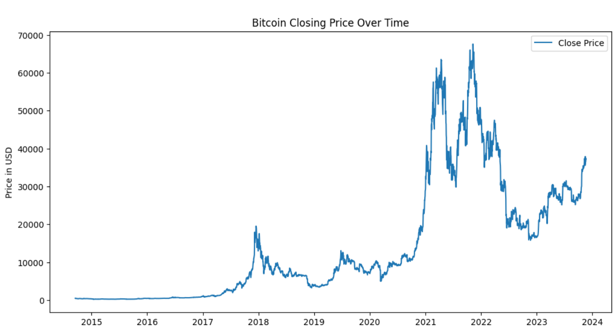


Figure 1, Bitcoin close price over time

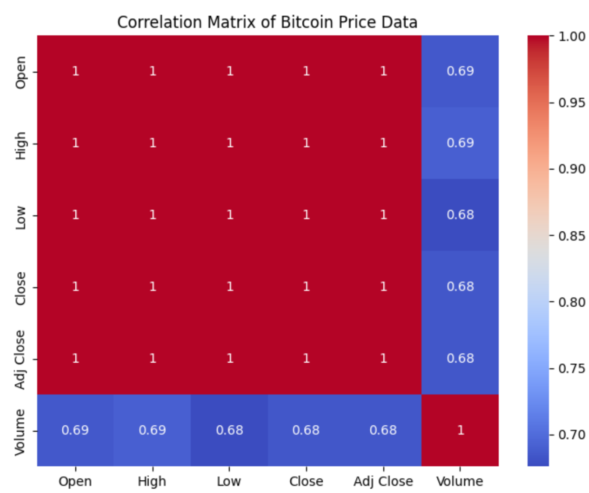


Figure 2, Correlation Matrix of the Dataset

Figure 1 shows how the value of bitcoin has swung wildly over time and Figure 3 shows the Annualized Volatility of BTC is significantly higher than other assets. The correlation matrix in Figure 2 shows how closely correlated all the columns are to eachother other than Volume, as these columns do not provide any useful information to our target value we will discard the majority of this data in the feature selection stage. The autocorrelation plot [10] (Figure 5) for bitcoin close prices (considering 50 lags) reveals how the price correlates with its past values. Significant autocorrelation is observed with peaks outside of the confidence band which seems to be the case for Bitcoin close prices. This indicates that past values have substantial influence on future values which is a common characteristic of time series data.

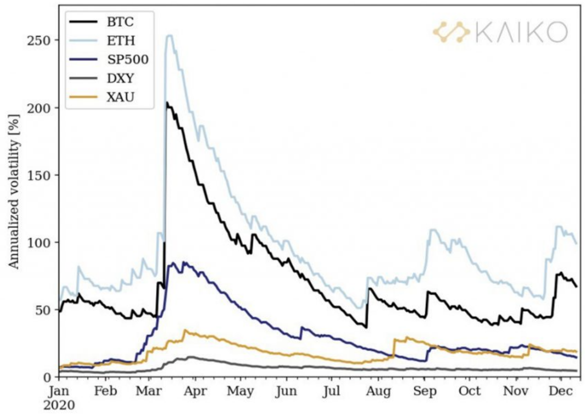


Figure 3, Volatility of Bitcoin from 2020 (in black) [11]

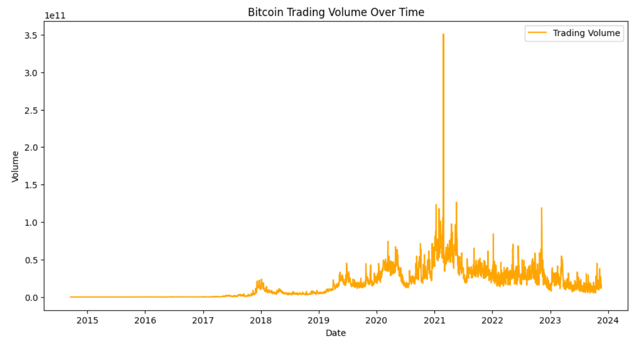


Figure 4, Trading Volume over time

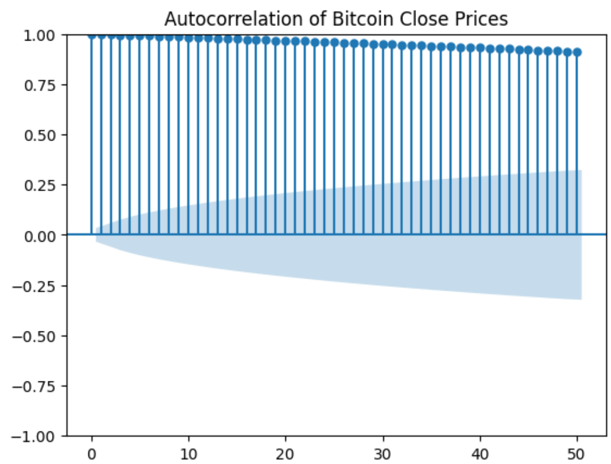


Figure 5, Autocorrelation of Bitcoin Close Prices

The dataset did not include any NaN or Null values hence no requirement to remove these values which suggests good data completeness. Observations from the EDA suggest Bitcoin's price has shown significant volatility which is evident from large standard deviations (seen in the appendix) in the Open, High, Low, Close columns.

Data pre-processing and Feature Selection

Data pre-processing and Feature Selection must be done to improve the quality of the data set and prepare it for modelling. In this study we chose to reduce the dimensionality of the data down to just Close price and Date. The decision to rule out volume in the modelling process is based on numerous research papers that suggest that a large majority of bitcoin trade volumes are manipulated through practices like wash trading[12]. This means that the volume data may not accurately reflect the true market activity which can be evidenced by the correlation score of ≈ 0.68 (Figure 2). The removal of this data would also reduce the noise on the input side of the models where including this potentially random information could obscure underlying patterns potentially leading to impaired accuracy. As the price of Bitcoin is influenced by a multitude of factors, by focusing solely on Close price, the models can concentrate on these direct indicators of market behaviour rather than potentially misleading volume data. The decision also aligns with empirical findings in existing literature [13] where models excluding volume data have shown superior performance in predicting asset prices suggesting that volume may not add significant predictive power in the context of Bitcoin price forecasting. Since the price of Bitcoin has drastically fluctuated from ≈ $400 in 2014 to ≈$67,000 in 2021 to around ≈$37500 as of the time of this report, we will consider only 1 year of data to avoid these large fluctuations in the data but we will different hyper parameters . After all the pre-processing, we are left with 365 rows and 2 columns.

Long Short-Term Memory Model (LSTM)

Using the pre-processed time series data, numerous variations of the LSTM were experimented with, a discovery made was that a 2 layer LSTM model where the first layer has the return_sequences=True parameter, which means it returns the full sequence to the next layer instead of just the output of the last timestep. This is crucial for stacking LSTM layers. The second LSTM layer then processes this sequence further. This additional layer allows the model to learn more complex features and relationships in the data, leading to better performance in the models prediction ability. The LSTM model that performed the best had two layers with 128 units each processed with a batch size of either 8 or 16 on a test train split of 60-40. This model had an R2 score of 0.9758 on the training data set and 0.9523 on the test data set with an RMSE of 667. It is to be expected that the model performs better on 'seen' data than unseen data but as the validation loss is decreasing, this is a positive sign that the model is generalising well and not just memorising the training data.

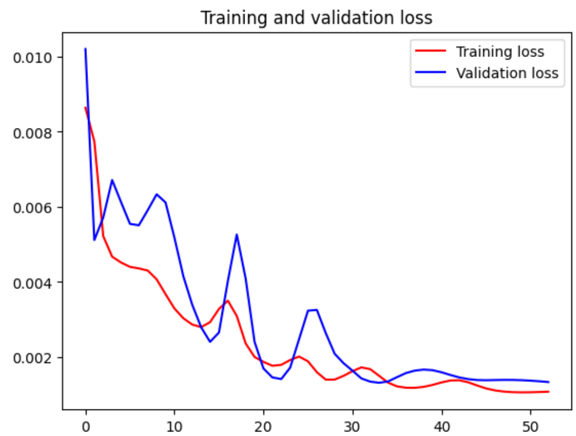


Figure 6, 2 layer LSTM Loss curve

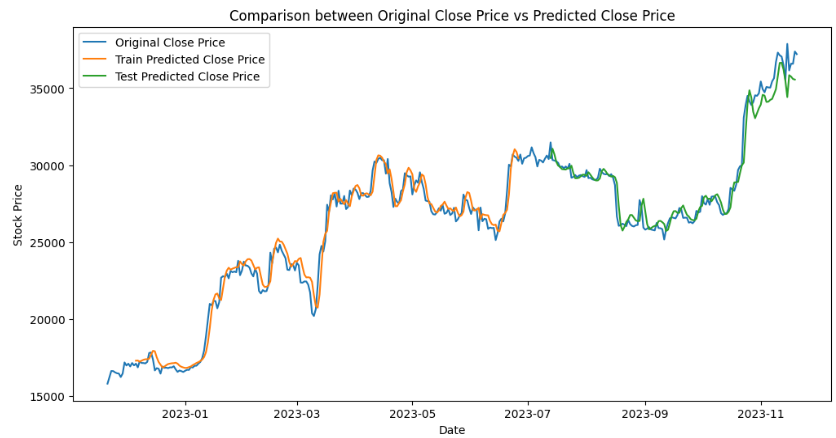


Figure 7, Comparing the train and test predicted price

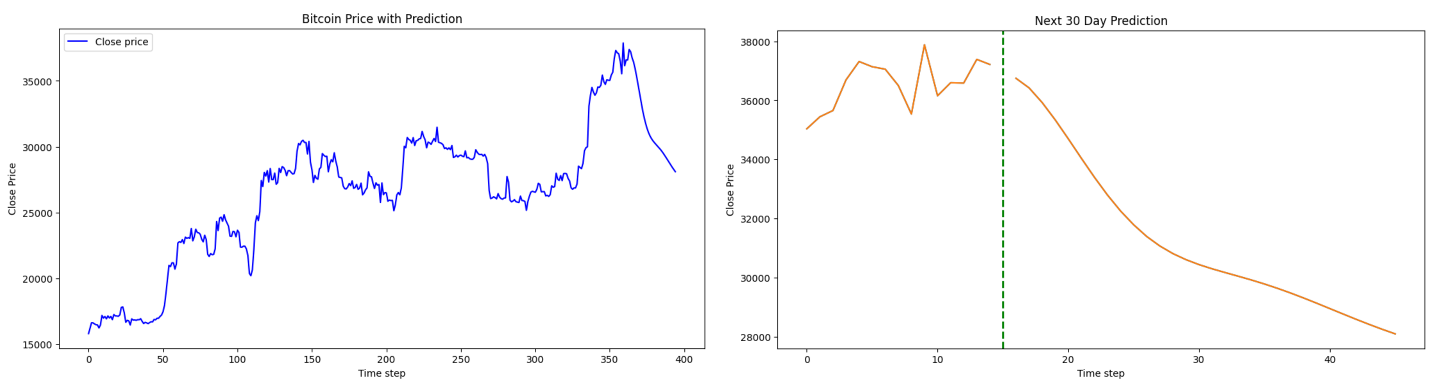


Figure 8 & 9, Using the model to predict the next 30 days of Bitcoin price movement.

Prophet

Using my data set on the prophet model generated an R2 score of 0.9015 and RMSE of 1579. This machine learning model is based on an additive model where non-linear trends are fit with yearly, weekly and daily seasonality which can be described statistically as $y(t)=g(t)+s(t)+h(t)+\epsilon t$. Where $g(t)$ represents the trend function which models non-periodic changes. $s(t)$ represents periodic changes (e.g., weekly, yearly seasonality). ϵt represents the error term accounts for any idiosyncratic changes which are not accommodated by the model. $h(t)$ represents the effects of holidays which occur on potentially irregular schedules [14]. The model fits to historical data by decomposing a time series into several componoents and uses a Bayesian approach [15] to estimating parameters of the trend, seasonality and holiday components. One key feature is changepoint detection, this gives the model the ability to adapt to shifts in trends when the time series data shows abrupt changes in trajectory, due to the volatile nature of bitcoin, this feature proves to be useful. Figure 10 shows the output of the Prophet model with the changepoints marked in blue.

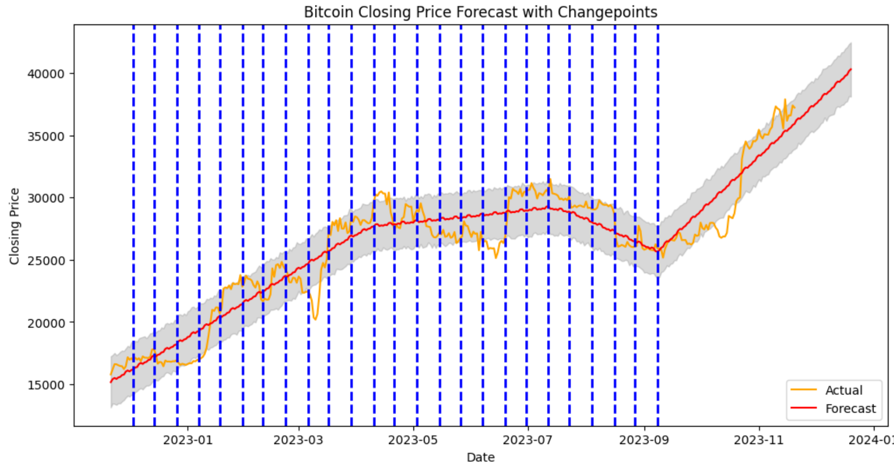


Figure 10, Prophet model output with 30 day forecast and changepoints marked

Evaluation

The evaluation metrics used in this study are Root Mean Square Error (RMSE) and R^2 are the main scores used for this study, RMSE is a measure of how close the regression line is to a set of points [16]. The scale of the RMSE in this study should be considered relative to the scale of the dependant variable which is Bitcoin price, which trades in the tens of thousands of dollars and can have mutli thousand dollar moves in a short time frame. Before we get to the evaluating, we conducted hyper parameter testing as models can perform poorly if these values are not configured correctly.

In this study the hyper parameters investigated for the 2 layer LSTM model were, batchsize and LSTM units. [17][18]

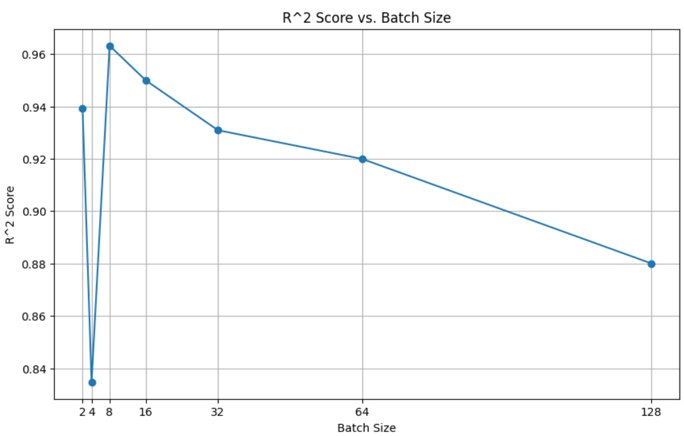


Figure 11, R^2 Score plotted against Batch size

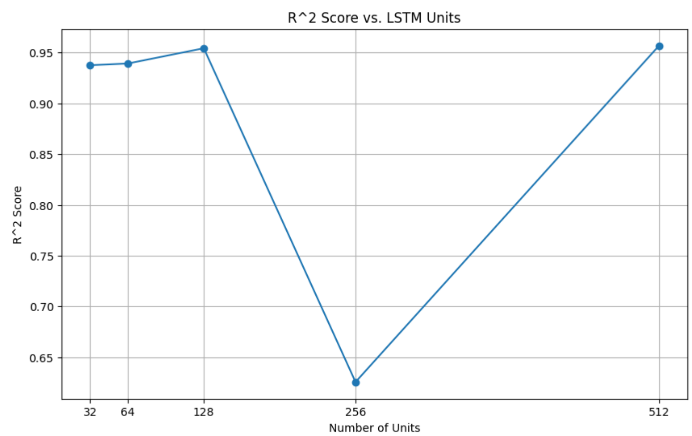


Figure 12, R^2 Score plotted against Number of Units

Smaller batch sizes lead to slower training times but can help improve generalisation as each update is less representative of the whole data set., larger batch sizes speed up training speed but can lead to a poorer estimation. Figure 11 shows the result of different batch size parameter testing, where for this model it is observed that batchsize equal to 8 leads to the best R2 score on this dataset.

LSTM units were also tested, these units refer to the cells or neurons in an LSTM layer, each LSTM unit has a cell state and three gates (input, output and forget), these cells give LSTM models the ability to capture long term trends and the ability to retain information. For this dataset it was observed that 128 units in both of the hidden layers gave the best R2 score which is evident in Figure 12. Both models demonstrate strong performance on the Bitcoin price data set, where LSTM showed a greater accuracy with the higher R2 score but was comparatively more computationally intensive and complex to test and train than Prophet.

Limitations:
One limitation of using LSTM Recurrent Neural Network is the computational complexity involved in testing and training the model. Even with only 365 daily observations after pre-processing, the model took significantly longer to train then that of Prophet. Another limitation of using LSTM is that due to the complexity and capacity of the network, these models have a higher risk of overfitting the data, especially with smaller datasets.
The limitations of Meta's Prophet include that due to its simplicity which can be advantageous, Prophet may not capture the complex relationships in the data as effectively as a more sophisticated LSTM model.
It is also possible that with only one year of data, Prophet might not capture the multi-year seasonality patterns or longer term trends effectively.
Observing the 30 day future predictions of both models (Figure 8 & Figure 10) gives curious results, the LSTM model predicts a steep decline in the Bitcoin price which seems improbable but could guide investors on the future price trajectory of Bitcoin. The 30 day prediction of Prophet just assumes the continued increase and trend continuation over the next 30 days. As these models give completely opposite 30 day predictions it is not recommended to make any financial decisions based on any implied predictions that these models give.

Conclusion:
After hyper parameter testing, the 2-layer LSTM performed marginally better than the simpler Prophet model but comes at the cost of being more computationally intensive. Although Prophet performed well with an R2 score of 0.9 for its relative simplicity.
Further work could involve using a dataset that takes a higher frequency of observations for example, minutely or hourly, this would increase the computational cost of training but could lead to differing results. It may also be relevant to scrape the internet with any mentions or articles of bitcoin and conduct sentiment analysis [19] to then augment the models which could give improved results as our model only takes price into consideration.

References:
[1] Tableau. (n.d.). Time Series Forecasting. Available at: https://www.tableau.com/learn/articles/time-series-forecasting.

[2] Investopedia. (n.d.). Bitcoin. Available at: https://www.investopedia.com/terms/b/bitcoin.asp.

[3] Analytics Vidhya. (2021). Introduction to Long Short Term Memory (LSTM). Available at: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/.

[4] Wikipedia. (n.d.). Deep learning. Available at: https://en.wikipedia.org/wiki/Deep_learning.

[5] Facebook. (n.d.). Prophet. Available at: https://facebook.github.io/prophet/.

[6] Thomson Reuters. (n.d.). Cryptos on the Rise 2022. Available at: https://www.thomsonreuters.com/en/reports/cryptos-on-the-rise-2022.html.

[7] BSC News. (n.d.). How Bitcoin ETF Approval Can Impact the Crypto Market. Available at: https://www.bsc.news/post/how-bitcoin-etf-approval-can-impact-the-crypto-market.

[8] JFin - SWUFE. (2020). Available at: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-020-00217-x.

[9] Yahoo Finance. (n.d.). BTC-USD Historical Data. Available at: https://finance.yahoo.com/quote/BTC-USD/history?p=BTC-USD.

[10] Wikipedia. (n.d.). Autocorrelation. Available at: https://en.wikipedia.org/wiki/Autocorrelation.

[11] Deribit Insights. (n.d.). Bitcoin Volatility and Correlation to Major Asset Classes. Available at: https://insights.deribit.com/industry/bitcoin-volatility-and-correlation-to-major-asset-classes/.

[12] Forbes. (2022). More Than Half of All Bitcoin Trades Are Fake. Available at: https://www.forbes.com/sites/javierpaz/2022/08/26/more-than-half-of-all-bitcoin-trades-are-fake/?sh=5c9f87b36681.

[13] Journal of Big Data. (2020). [Short-term stock market price trend prediction using a comprehensive deep learning system]. Available at: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6.

[14] Medium - Aditya Roc. (n.d.). A Noob's Guide to Facebook's Prophet. Available at: https://adityaroc.medium.com/a-noobs-guide-to-facebook-s-prophet-d9655cff7583.

[15] ScienceDirect. (n.d.). Bayesian Approach. Available at: https://www.sciencedirect.com/topics/engineering/bayesian-approach#:~:text=The%20Bayesian%20approach%20begins%20by,out%20of%20sample%20historical%20data.

[16] Wikipedia. (n.d.). Root-mean-square deviation. Available at: https://en.wikipedia.org/wiki/Root-mean-square_deviation.

[17] Stack Exchange - Statistics. (n.d.). What is batch size in neural network? Available at: https://stats.stackexchange.com/questions/153531/what-is-batch-size-in-neural-network.

[18] Pluralsight. (n.d.). Introduction to LSTM Units in RNN. Available at: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn.

[19] National Center for Biotechnology Information. (n.d.). [Stock trend prediction using sentiment analysis]. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10403218/#:~:text=They%20can%20utilize%20text%20mining,to%20predict%20various%20stock%20trends.

Appendix:

2 layer LSTM train and test R2 scores:

```
Train data R2 score: 0.9758360835278365
Test data R2 score: 0.9523352190268198
```

Testing LSTM with volume leads to a significant reduction in r2 score

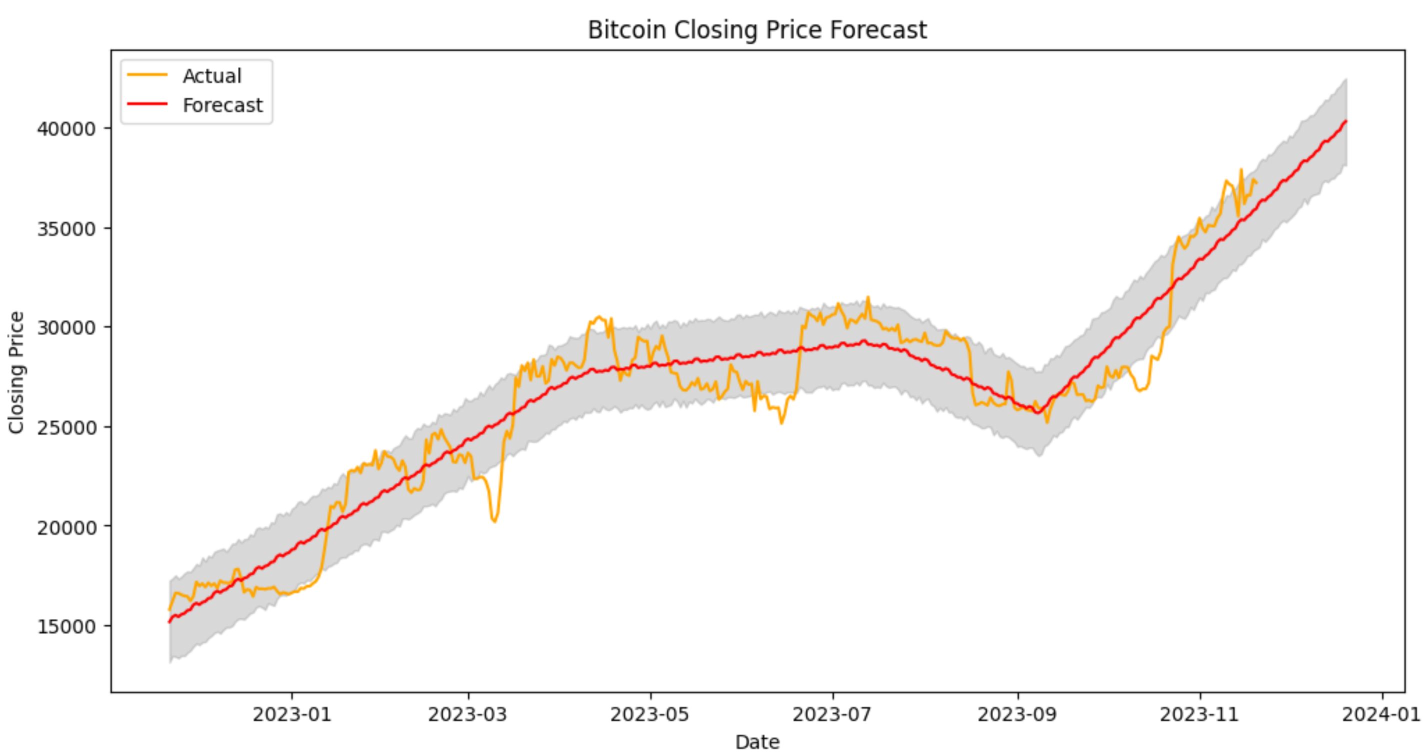
```
[44] print("Train data explained variance regression score:",
           explained_variance_score(original_ytrain, train_predict))
     print("Test data explained variance regression score:",
           explained_variance_score(original_ytest, test_predict))

     Train data explained variance regression score: 0.8869211022015433
     Test data explained variance regression score: 0.7726789520861262

[45] print("Train data R2 score:", r2_score(original_ytrain, train_predict))
     print("Test data R2 score:", r2_score(original_ytest, test_predict))

     Train data R2 score: 0.8868374589138299
     Test data R2 score: 0.7707279506476561
```

Prophet graph forecast



Prophet model scores

```
Root Mean Squared Error:  1579.6653159785853
Mean Squared Error:  2495342.5105057238
R2 Score:  0.9015418767626449
```

Code available in the .ipynb notebooks