University of Hertfordshire

School of Engineering and Computer Science

MSc Computer Science (Data Science and Analytics)

Advanced Computer Science Masters Project (7COM1039)

2023

Final Project Report (FPR)

**Bitcoin Price Prediction using Machine Learning**

Student Name: Ezugwu Thankgod Chukwuebuka

Student ID: 21084271

Supervisor: Liang Chen

**Abstract**

This research project explores the use of machine learning techniques to predict cryptocurrency prices, with a primary focus on Bitcoin. The study is motivated by the increasing popularity of cryptocurrencies and the need for accurate prediction tools in this volatile market. The project addresses challenges related to market dynamics, real-time data, and data quality. It emphasizes the importance of precise price predictions for risk management and returns optimization. The findings have practical implications for investors navigating the cryptocurrency market, offering insights into risk mitigation and market efficiency. Additionally, the research contributes to the academic discourse on cryptocurrency analysis and prediction, ultimately highlighting the suitability of LSTM as a robust predictive model and showcasing the potential of machine learning in this field.

**Table of Contents**

**Abstract……………………………………………………………………………………1**

**Table of contents………………………………………………………………………...2**

**CHAPTER 1: INTRODUCTION…………………………………………………………5**

1.0 Introduction……………………………………………………………………………5

1.1 Research question……………………………………………………………………5

1.2 Motivation……………………………………………………………………………...6

1.3 Problem Statement…………………………………………………………………...7

1.4 Aim and Objectives…………………………………………………………………...8

1.5 Significant of study……………………………………………………………………8

1.5.1 Why This Problem is Interesting…………………………………………………..9

**CHAPTER 2: LITRATURE SURVEY…………………………………………………..11**

2.0 Introduction…………………………………………………………………………….11

2.2 Litrature Review……………………………………………………………………….11

2.3 Research Similar to the Topic……………………………………………………….13

2.4 State of Art…………………………………………………………………………….14

2.5 Implemented Techniques on Bitcoin Prediction…………………………………..14

2.6 Gap Analysis………………………………………………………………………….15

2.7 Critical Analysis………………………………………………………………………16

**CHAPTER 3: RESEARCH METHODOLOGY…………………………………………18**

3.1 SEMMA DM Crisp Model……………………………………………………………18

3.1.1 Reason for using SEMMA methodology………………………………………..20

3.2 Data Set Information………………………………………………………………...20

3.3 Reason for Dataset Selection………………………………………………………21

3.4 Model Selection……………………………………………………………………...21

3.4.1 LONG AND SHORT TERM MEMORY (LSTM)………………………………..22

3.4.2 Advantages of LSTM(Wang, Y., et al., (2019) )………………………………..22

3.4.3 Disadvantages of LSTM(Burgueno, L., et al., (2019)…………………………22

3.4.4 WORKING OF LSTM…………………………………………………………….23

3.5 LIBRARIES………………………………………………………………………….24

3.6 Metrics……………………………………………………………………………….26

**CHAPTER 4: EXPERIMENTS AND IMPLEMENTATION………………………...30**

4.1 Libraries……………………………………………………………………………..30

4.2 Data Pre-processing……………………………………………………………….31

4.2.1 Import Libraries….……………………………………………………………….31

4.2.2 Load dataset…..……………………….………………………………………...32

4.2.3 Understand the structure of the dataset………………………………………32

4.2.4 Understand the statistical distribution of the data……………………………33

4.2.5 Check null values………………………………………………………………..34

4.2.6 Understand the correlation……………………………………………………..34

4.2.7 Understand the relationship using heatmap………………………………….35

4.2.8 Feature selection…………………………………………………………………35

4.2.9 Datetime Format………………………………………………………………….36

4.2.10 Lineplot before filtering…………………………………………………………36

4.2.11 Filter the data……………………………………………………………………37

4.2.12 Lineplot after filtering…………………………………………………………...37

4.2.13 Pre-processing…………………………………………………………………..38

4.3 Train and Test……………………………………………………………………….39

4.3.1 Training LSTM Model…………………………………………………………….40

4.3.2 Training and Validation…………………………………………………………..42

**CHAPTER 5: EVALUATION AND RECOMMENDATION………………………….45**

5.1 EVALUATION……………………………………………………………………….45

5.1.1 RMSE, or Root Mean Square Error…………………………………………….45

5.1.2 MSE, or Mean Squared Error…………………………………………………...45

5.1.3 MAE: Mean Absolute Error………………………………………………………45

5.1.4 R Squared - Coefficient of Determination (R2) Score………………………...46

5.1.5 Score from the Train Data Explained Variance Regression………………….47

5.1.5.1 Score for Explained Variance Regression Using Test Data………………..47

5.2 Comparison between original data and predicted data………………………….48

5.3 Recommendations…………………………………………………………………..48

**CHAPTER 6: CONCLUSION AND LIMITATIONS…………………………………...51**

6.1 Conclusion……………………………………………………………………………51

6.2 LIMITATIONS………………………………………………………………………….51

Bibliography……………………………………………………………………………….54

**CHAPTER 1**

**INTRODUCTION**

Bitcoin, created by an anonymous entity known as Satoshi Nakamoto in 2009, is a decentralized digital currency that operates on a technology called blockchain. Unlike traditional currencies, Bitcoin is not issued or regulated by any central authority, such as a government or central bank. Instead, it relies on a network of nodes to validate and record transactions.

The price of Bitcoin is notoriously volatile, making it a compelling subject for financial analysis and prediction. Investors and traders are keen to understand and anticipate price movements to make informed decisions. Various strategies have been used by academics and business professionals to forecast the price of Bitcoin. These methods include technical analysis, which looks at past price patterns and market indicators to spot prospective trends, and fundamental analysis, which determines the true worth of Bitcoin based on variables like adoption rates, transaction volume, and network activity (Mcafee, A. and Brynjolfsson, E., 2017).

It is crucial to remember that it is difficult and unpredictable to precisely anticipate the price of Bitcoin, or any other financial asset for that matter. Being extremely speculative and subject to sharp price fluctuations, the cryptocurrency market makes it challenging to reliably forecast short-term price changes. Long-term trends and patterns, however, might offer insights into possible future price orientations (Kristoufek, L., 2013).

Numerous academic studies, research papers, and market analyses have been published on this topic as interest in predicting the price of Bitcoin rises. These publications offer insightful approaches for comprehending and predicting the price of Bitcoin. Furthermore, trustworthy cryptocurrency exchanges, financial news sources, and research platforms frequently provide consistent analyses and forecasts from subject-matter specialists.

**1.1 RESEARCH QUESTION**

The research question aims to investigate the overall effectiveness of machine learning techniques specifically in the context of predicting cryptocurrency prices. The primary research question is: What is the effectiveness of machine learning techniques in predicting cryptocurrency prices?

**1.2 MOTIVATION**

This research was motivated by the rising popularity of cryptocurrencies and the demand for accurate tools to predict price fluctuations. Although cryptocurrency markets present significant opportunities for investors, they also carry a number of hazards. The goal is to assist investors in making informed decisions, reducing risks, and potentially boosting their profits by creating precise price prediction model. This initiative aligns with the overarching objective of exploring the potential of machine learning in financial analysis and forecasting. Below highlights several key points that justify the need for such this project:

**Market Dynamics and Volatility:** Due to Bitcoin's high level of volatility, it is difficult to correctly forecast its future price alterations. Predictive models that may exceed conventional forecasting techniques may be created using machine learning algorithms, which can identify complex patterns and correlations in market data (Zhang, Y., et al., 2018).

**Data Availability and Accessibility:** There is a plethora of data that can be used for predictive modelling due to the growth of cryptocurrency exchanges and the growing availability of historical price and transaction data. Large-scale data analysis and pattern recognition are areas in which machine learning algorithms thrive. These areas may not be immediately obvious to humans. These algorithms enable us to examine large datasets, analyse them for useful insights, and build reliable prediction models (Malhotra, B. et al., 2022).

**Support for Investment Decisions:** By giving investors, traders, and financial institutions useful information for their investment decision-making processes, accurate Bitcoin price predictions may be quite advantageous. We can provide probabilistic projections by utilising machine learning algorithms that take into account several variables, assisting investors in identifying possible buy or sell opportunities, optimising portfolio allocation, and successfully managing risk. Furthermore, being able to anticipate short-term price changes might help traders take advantage of market imperfections and close deals on time (Kazeminia, S., et al., 2023).

**1.3 PROBLEM STATEMENT**

The current challenge is to correctly estimate Bitcoin prices using machine learning methods. Investors, traders, and financial institutions that want to make educated decisions and reduce risks in the cryptocurrency market would greatly benefit from the capacity to anticipate Bitcoin prices with a high degree of accuracy.

The problem statement includes a number of significant difficulties:

**Price Volatility:** The cryptocurrency market experiences significant price volatility due to various factors, including investor attitude, market demand, governmental changes, and technological improvements. Finding trustworthy patterns and trends for precise price prediction is challenging due to this unpredictability.

**Continuous Market:** The cryptocurrency market operates around the clock, and global events can impact Bitcoin prices at any time. This necessitates real-time data collection, processing, and prediction capabilities to provide the most up-to-date forecasts.

**Complex Price Fluctuations:** Bitcoin's price fluctuations are highly complex and non-linear. Traditional statistical models and forecasting methods may struggle to account for the intricate connections and linkages within the data. Machine learning algorithms are better equipped to identify and utilize these complex patterns and connections.

**Data Quality:** The accuracy of predictions depends on the quantity and quality of data used to train and evaluate machine learning models. Gathering, cleaning, and preprocessing cryptocurrency data is essential, including historical price data, trade volumes, and other relevant attributes.

The overarching issue statement focuses on creating trustworthy forecasts using robust and accurate machine learning models that can effectively capture Bitcoin's price fluctuations. These models should address the challenges of market volatility, real-time data processing, complexity, and data quality to provide valuable insights and assist market participants in making informed decisions in the Bitcoin market.

**1.4 AIM AND OBJECTIVES**

The project aim is to investigate and develop a predictive model that forecasts the price movements of Bitcoin, a popular and volatile cryptocurrency, using machine learning algorithms. The project's ultimate goal is to provide valuable insights to investors, traders, and stakeholders interested in the cryptocurrency market. The project focuses on utilizing historical price data, along with relevant features, to build accurate and reliable predictive models. The objectives of the project can be broken down as follows:

* Collect and preprocess cryptocurrency data: The first objective involves gathering historical cryptocurrency data and relevant features from various sources. Ensuring data quality through cleaning, normalization, and preprocessing is a priority.
* Select appropriate machine learning algorithms: The project aims to explore and choose suitable machine learning techniques for predicting cryptocurrency prices, including ensemble methods, deep learning models, and regression models.
* Develop and train machine learning models: This objective focuses on creating and training machine learning models using the collected and processed data. Creating training, validation, and test sets is essential for optimizing model performance.
* Evaluate model performance: The project aims to assess the precision and efficacy of machine learning models in predicting cryptocurrency values using appropriate evaluation metrics such as MAE, RMSE, and directional correctness. And lastly, to identify potential areas of improvement and future research directions in the field of cryptocurrency price prediction using machine learning.

**1.5 SIGNIFICANCE OF STUDY**

The study on Bitcoin price prediction using machine learning holds substantial significance in the realm of financial analysis and cryptocurrency markets. Bitcoin, as a decentralized digital asset, has not only gained widespread attention but also posed unique challenges and opportunities for investors, traders, and financial institutions. Understanding the importance of solving the problem of predicting Bitcoin prices using machine learning requires a comprehensive exploration of its broader implications.

**1.5.1 Why This Problem is Interesting:**

**Investor Decision-Making:** Bitcoin's volatile nature has made it a compelling investment option for many. Accurate price predictions can provide invaluable insights to investors, helping them make well-informed decisions regarding when to buy, sell, or hold Bitcoin. This is of particular interest given the potential for significant financial gains or losses in the cryptocurrency market.

**Risk Mitigation:** Cryptocurrency investments inherently carry risk due to the market's unpredictability. Machine learning-based price predictions offer a tool for risk management by enabling investors and financial institutions to assess potential price movements and optimize their portfolios accordingly. This is especially relevant in an era where cryptocurrencies are gaining recognition as a legitimate asset class.

**Market Efficiency:** Efficient financial markets require the availability of accurate and timely information. Machine learning models can process vast amounts of data in real-time, potentially contributing to market efficiency by incorporating a wide range of factors that influence Bitcoin prices. This may reduce information asymmetry and promote fairer trading environments.

**Technological Advancements:** The study delves into the intersection of finance and technology. Developing robust machine learning models for Bitcoin price prediction is not only a challenging intellectual endeavor but also contributes to the advancement of financial technology. It showcases the potential of modern AI techniques in addressing complex and dynamic financial challenges.

**Economic Implications:** Accurate Bitcoin price prediction models have implications beyond individual investors. They can impact broader economic decisions, such as central bank policies, regulatory measures, and investment strategies of institutional players. Understanding and forecasting cryptocurrency markets is becoming increasingly important as they become integrated into the global financial ecosystem.

**Research and Academic Contribution:** This study extends the boundaries of research in both machine learning and cryptocurrency analysis. It offers opportunities for scholars and researchers to explore novel methods, datasets, and evaluation metrics in the context of Bitcoin price prediction. Additionally, it encourages interdisciplinary collaboration between computer science and finance domains.

**Future Innovations:** As the cryptocurrency landscape continues to evolve, the ability to predict Bitcoin prices accurately becomes ever more crucial. Innovations in this field may lead to the development of novel financial products, trading strategies, and investment vehicles, fostering further growth in the cryptocurrency market.

The study on Bitcoin price prediction using machine learning is not only academically intriguing but also holds substantial practical significance. It addresses a pertinent issue in the cryptocurrency ecosystem and offers solutions that can potentially benefit individual investors, financial institutions, and the broader financial market. Moreover, it contributes to the ongoing dialogue about the role of machine learning in shaping the future of finance.

**CHAPTER 2**

**LITRATURE SURVEY**

**2.1: Introduction**

The rise of cryptocurrencies in recent years has fundamentally changed the financial landscape, with Bitcoin leading the charge as the innovator and most prominent player. Investors, speculators, and researchers have all been enthralled by its fast increase in value and decentralised nature. As a result, forecasting Bitcoin's price changes has gained critical significance since it may help investors make decisions, reduce risks, and gain a better knowledge of the dynamics of the cryptocurrency market.

This study of the literature intends to investigate the many strategies, approaches, and models that have been created to forecast the price of Bitcoin. Accurate prediction has proven to be a tough issue because of the market's intrinsic volatility and vulnerability to many external influences, such as regulatory changes and macroeconomic conditions. In their efforts to predict the price of Bitcoin, researchers have used a wide variety of instruments and methods, pulling from disciplines like economics, finance, statistics, and machine learning.

Review aims to offer a thorough overview of the state of the art in Bitcoin prediction, highlight strengths and limits of current approaches, and throw light on ongoing discussions and developments in this dynamic subject by examining the body of existing research. It also seeks to identify trends and shared characteristics among researchers' methods, enabling a critical evaluation of the validity and applicability of diverse prediction models.

**2.2 Literature Review**

Due to the cryptocurrency's quick uptake and unpredictable nature, forecasting Bitcoin price changes using machine learning has grown in popularity. An overview of the main approaches, difficulties, and conclusions in machine learning-based Bitcoin price prediction will be given in this study of the research.

The employment of supervised learning algorithms is a common strategy for forecasting Bitcoin values. Numerous regression models, including linear regression, support vector regression (SVR), and more recently deep learning methods like neural networks, have been studied by researchers. In order to predict future price patterns, these models often use historical Bitcoin price data, trade volume, and technical indicators as input characteristics (Roy, S., et al., (2018); Jang, H. and Lee, J. (2018)). SVR was used by Roy, S., et al., (2018) to forecast Bitcoin values based on a number of characteristics, such as trade volume and price history. Similar to this, Jang, H. and Lee, J. (2018) used a variety of machine learning models to capture the complex price patterns of Bitcoin.

Another popular method for predicting the price of bitcoin is time series analysis. The intrinsic time-dependent properties of Bitcoin have been modelled using techniques like Autoregressive Integrated Moving Average (ARIMA) and Generalised Autoregressive Conditional Heteroskedasticity (GARCH) (Qi, T. *et al.* (2020)). The serial correlation and volatility clustering that are frequently seen in bitcoin price data are taken into consideration by these models.

In addition, sentiment analysis has become more popular for forecasting Bitcoin values. To assess market mood and probable price fluctuations, this entails analysing information from social media, news articles, and other textual sources. According to Shah, D., et al., (2018), public opinion may affect trading choices and eventually affect Bitcoin prices. To extract sentiment-related information for price prediction, researchers have created sentiment analysis models that incorporate natural language processing and machine learning methods.

Although these techniques have produced encouraging results, predicting the price of Bitcoin is still difficult. Regulatory changes, macroeconomic developments, investor attitude, and other variables all have an impact on the cryptocurrency market, which can result in erratic and rapid price movements. Additionally, because to its speculative character, Bitcoin is particularly vulnerable to market manipulation and high volatility, which can throw off forecasting models.

Using machine learning to predict Bitcoin values is a complex task that incorporates supervised learning, time series analysis, and sentiment analysis, among other strategies. Although scientists have made great strides in creating models for predicting Bitcoin prices, the intrinsic complexity of the cryptocurrency market continues to pose difficulties. Future studies will probably concentrate on improving current models, adding new data sources, and adjusting to the shifting dynamics of the bitcoin ecosystem.

**2.3 Research Similar to the Topic**

Data gathering: From a variety of sources, researchers compile historical information on Bitcoin values and other characteristics. These sources might consist of news articles, financial platforms, blockchain data, social media sites, and cryptocurrency exchanges. The steps in the data gathering process entail finding and arranging the pertinent information for analysis(Kondor, D. *et al.* (2014) ).

Researchers find and develop pertinent aspects that could have an influence on Bitcoin values. Technical indicators, market mood indicators, on-chain measurements, trade volume, and volatility measures are a few examples of these elements. The process of feature engineering entails choosing the features to be transformed into an analysis-ready format(Zhang, H., Shen, D. and Li, X. (2020)).

Model Selection: Based on the study goal and the features of the data, researchers select relevant models that are suited for predicting the price of bitcoin. Time series models (like ARIMA, GARCH), machine learning models (like regression, SVM, ANN), deep learning models (like LSTM, GRU), and ensemble models are examples of frequently used models(Atsalakis, G. S. and Valavanis, K. P. (2015)).

Model Training and Evaluation: To train the chosen models, researchers divided the dataset into training and testing sets. Afterward, the models are trained using the historical data, and the performance is assessed using suitable evaluation metrics, such as mean absolute error (MAE), mean squared error (MSE), root mean square error (RMSE), or accuracy measures. The resilience of the model may also be evaluated using cross-validation methods(Fledelius, P., Santos, J. C. and Würtz, D. (2021)).

Ensemble approaches: To increase prediction accuracy, several research integrate many models or employ ensemble approaches. Averaging predictions from many models, implementing model stacking strategies, or employing ensemble algorithms like random forests or gradient boosting are all examples of ensemble approaches(Bouri, E., Azzi, G. and Dyhrberg, A. H. (2019)).

Evaluation Metrics and Statistical Tests: To determine the importance and correctness of their models, researchers use a variety of statistical tests and evaluation metrics. To validate the findings, these tests may include t-tests, F-tests, p-values, and statistical significance analyses.

**2.4 State of Art**

According to Singh, V. K. and Srivastava, G. (2021), Models for machine learning For predicting the price of bitcoin, machine learning techniques including artificial neural networks (ANN), support vector machines (SVM), random forests, and recurrent neural networks (RNN) have been widely used. These models use sentiment analysis, technical indicators, historical price data, and other pertinent information to create predictions. Machine learning techniques have demonstrated encouraging results in capturing complex patterns and dynamics in Bitcoin price movements.

Sentiment Analysis: In order to understand market sentiment and how it affects Bitcoin values, sentiment analysis techniques are used to examine social media, news articles, and other textual data. To collect sentiment signals for price prediction, natural language processing (NLP) techniques such as lexicon-based methods and machine learning algorithms have been used (Agrawal, N. and Bhardwaj, A. (2019)).

Hybrid Approaches: To capitalise on the advantages of various approaches, several researchers have suggested hybrid models that incorporate a number of distinct prediction methodologies. To improve forecast accuracy and resilience, machine learning models can be combined with technical indicators, sentiment analysis, or blockchain data (Shahzad, S. A., et al., (2020)).

**2.5 Implemented Techniques on Bitcoin Prediction**

Autoregressive Integrated Moving Average (ARIMA): Bitcoin price predictions have been made using Autoregressive Integrated Moving Average (ARIMA), a popular time series forecasting methodology. Moving average (MA), differencing (I), and autoregression (AR) are its three main constituents. On the basis of these patterns, ARIMA models anticipate future values by capturing the linear connections between historical observations. They have been successful in identifying the seasonality and pattern of Bitcoin pricing (Siami-Namini, S. and Trabelsi, M. M. (2019)).

Networks with Long Short-Term Memory (LSTM): Recurrent neural networks (RNNs) of the LSTM variety have demonstrated potential in predicting the price of bitcoin. By utilising specialised memory cells, LSTM networks can detect long-term relationships in time series data. They are perfect for collecting intricate patterns in Bitcoin price data since they have the capacity to preserve crucial information over long periods of time (Fischer, T., et al., (2019)).

Support Vector Regressor (SVM): The machine learning method known as Support Vector Regression (SVR) has been used to forecast the price of bitcoin. It is a member of the SVM family of support vector machines that builds a regression model using a subset of training data known as support vectors. SVR has been used to forecast Bitcoin values based on a variety of factors including trade volume, sentiment analysis, and technical indicators. SVR is capable of capturing non-linear correlations (Zhang, Y., et al., (2018)).

Recurrent Neural Network (RNN): By taking into account the sequential structure of the data, RNNs have been used to forecast Bitcoin values. RNNs are a good choice for modelling the dynamics of the Bitcoin price because they can capture the temporal relationships in time series data. RNNs can produce forecasts for price changes by using a sequence of historical Bitcoin prices as input (Zheng, Z. et al. (2018)).

Ensemble Learning: To increase the reliability and accuracy of Bitcoin price forecasts, ensemble approaches combine numerous prediction models. An ensemble of many models may be built using strategies like bagging, boosting, and stacking. These models work together to provide predictions that are more accurate overall. The goal of ensemble techniques is to minimize bias and variation by combining the projections from many models, producing predictions that are more accurate (Moazami, H. et al. (2021)).

**2.6 Gap Analysis**

A thorough gap analysis of Bitcoin prediction using machine learning identifies several key areas where research and development can significantly advance the field. These areas encompass both technique and execution. Here are several significant gaps in the literature:

**Data Availability and Quality:** Predicting Bitcoin's price is hindered by the availability and quality of data. Cryptocurrency markets operate around the clock, leading to erratic and potentially manipulated data. Researchers need better historical data, real-time sources, and data cleaning techniques to ensure model correctness and reliability (Roy, S., et al., 2018).

**Feature Selection and Engineering:** There are gaps in feature selection and engineering. Researchers should explore novel variables that capture the unique dynamics of the Bitcoin market. Current models often rely on traditional data like past prices and trade volume, but more innovative features, such as network measurements and blockchain-based analytics, should be considered (Bhutta, M. N. M. et al., 2021).

**Sentiment Analysis Improvement:** Sentiment analysis methods for predicting Bitcoin values require enhancement. Future studies should focus on advanced natural language processing (NLP) techniques, sentiment lexicons designed for cryptocurrencies, and sentiment data from diverse sources (Otabek, S. and Choi, J., 2022).

**Model Interpretability:** Many machine learning models for Bitcoin price prediction lack interpretability. Researchers should work on creating models that provide explanations for their predictions, employing techniques like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) (Alamri, R. and Alharbi, B., 2021).

**2.7 Critical Analysis**

The critical examination in the table examines several aspects of machine learning-based Bitcoin prediction. Accurate and dependable prediction models are needed as cryptocurrencies become more prevalent on the financial markets. This paper explores important factors that developers and users of machine learning models for Bitcoin price prediction must take into account.

**Table 1: Critical Analysis**

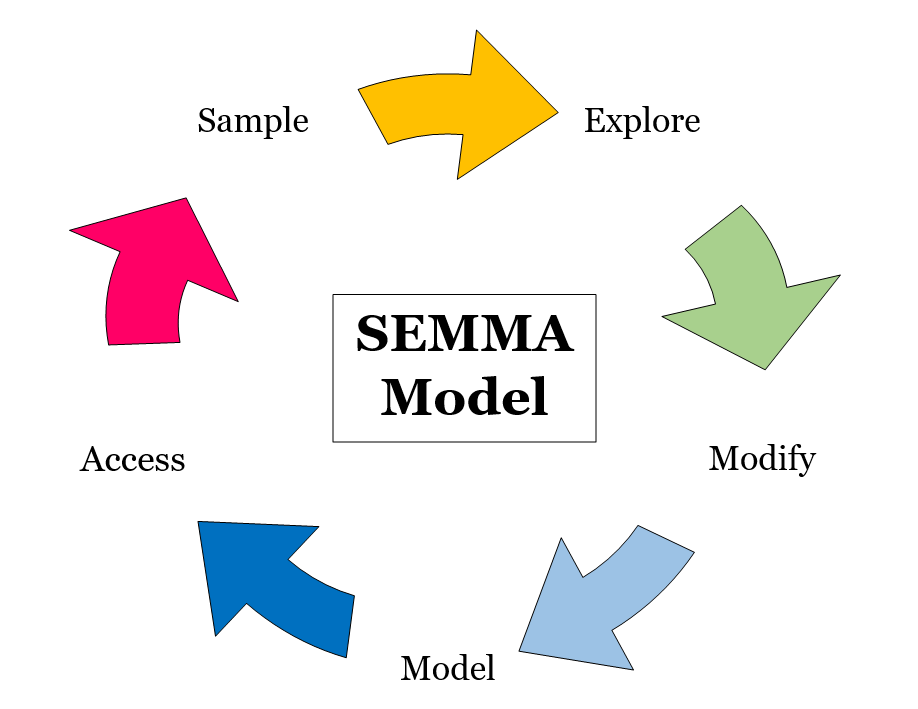
|  |  |  |
| --- | --- | --- |
| **Aspect of Critical Analysis** | **Discussion and Key Points** | **References** |
| Data Quality and Quantity | - The availability and quality of cryptocurrency data are crucial for model accuracy.  - Noisy and manipulated data pose challenges. | Roy, S., et al., (2018) |
| Feature Engineering | - Innovative feature engineering can improve model performance.  - Blockchain-based and on-chain data can be explored. | Bhutta, M. N. M. *et al.* (2021) |
| Sentiment Analysis | - Sentiment analysis is vital for market sentiment understanding.  - Advanced NLP techniques and diverse data sources are needed. | Otabek, S. and Choi, J. (2022) |
| Model Interpretability | - Interpretable models are necessary for trust and decision-making.  - LIME and SHAP methods can enhance model explain ability. | Alamri, R. and Alharbi, B. (2021) |
| Generalization and Robustness | - Models should not overfit to historical data.  - Techniques like transfer learning and reinforcement learning can help. | Naeem, M., et al., (2020) |
| Ethical and Regulatory Aspects | - Ethical considerations, fairness, and compliance with regulations are vital.  - Market manipulation and transparency issues need addressing. | ter Ellen, S. and Verschoor, W. F. C. (2017) |

**CHAPTER 3**

**RESEARCH METHODOLOGY**

The methodology used in this project is SEMMA. The SEMMA methodology is a framework used in the context of predictive modeling and data-driven decision-making. SEMMA stands for Sample, Explore, Modify, Model, and Assess. Here's how the SEMMA methodology can be applied to predict Bitcoin prices using machine learning:

**3.1 SEMMA DM Crisp Model**

****

**Figure 1: SEMMA DM Crisp Model**

**Sample:**

The first phase involves gathering historical information on Bitcoin prices, trading activity, market indicators, and other important elements. Typically, this information is gathered from a variety of sources, including blockchain data, financial platforms, news articles, social media, and cryptocurrency exchanges. The following step is to separate the obtained data into training, validation, and testing sets.

**Explore:**

Gain a deeper grasp of the gathered data during the exploration phase. To find patterns, trends, and correlations in the data, descriptive statistics, data visualisation methods, and exploratory data analysis are utilised. This process aids in locating prospective characteristics that could be helpful in forecasting Bitcoin values.

**Modify:**

The data is pre-processed and altered in this stage to get it ready for model training. Taking care of missing values, eliminating outliers, normalising or scaling the data, and doing feature engineering are all included in this. To enhance the prediction performance of the model, new features are developed or current ones are modified.

**Model:**

Following data modification, a variety of machine learning methods are used to create prediction models. For the purpose of predicting the price of bitcoin, regression models such as linear regression, support vector regression (SVR), artificial neural networks (ANN), decision trees, or ensemble approaches such as random forests or gradient boosting can be utilised. The models are optimised by modifying their parameters to minimize prediction errors after being trained using the training data.

**Assess:**

Using the testing and validation sets, the trained models are assessed during the assessment step. The effectiveness of the models in forecasting Bitcoin prices is measured using measures like mean absolute error (MAE), mean squared error (MSE), root mean square error (RMSE), or accuracy. The models that perform the best are chosen for additional examination.

**3.1.1 Reason for using SEMMA methodology**

This methodology SEMMA (Sample, Explore, Modify, Model, Assess) is a crucial data mining method for Bitcoin price prediction using machine learning because it provides a structured framework for the entire data mining process. First, it helps in efficiently sampling and exploring historical Bitcoin price data, enabling us to understand patterns and trends. Second, it facilitates data preprocessing and feature engineering, allowing us to enhance the quality of input data for machine learning models. Finally, SEMMA aids in model building and assessment, ensuring that our predictive models are robust and accurate, thus improving our ability to forecast Bitcoin prices with confidence.

**3.2 Data Set Information**

Obtaining a reliable and comprehensive dataset is paramount to building an accurate predictive model. As part of the accomplishment, after researching several different datasets, the one that matches the exact requirements is form the Kaggle Website, “The Bitcoin Stock Values”.

The historical data on Bitcoin stock values from September 17, 2014 to August 24, 2021 is included in the Bitcoin dataset that is chosen for implementation. This information will be used to estimate future prices, with the goal of estimating the price of Bitcoin based on previous trends and patterns.

Link to dataset: [[1]](#footnote-2)

The dataset acquisition from Kaggle significantly expedited the data collection process, providing a standardized and well-structured dataset suitable for analysis and model development. By leveraging this dataset, the project has been able to efficiently proceed with the subsequent data preprocessing and model training stages.

Terminologies:

Date: the day of the observation

Open: The day's starting price

High: The day's highest price.

Low: The day's lowest price.

Close: The day's closing price.

Volume: A day's worth of transactions.

Adj. Closing Price: Adjusted closing price of Bitcoin on a specific day.

**3.3 Reason for Dataset Selection**

The chosen dataset from Kaggle, "The Bitcoin Stock Values," offers several compelling reasons for selection. Firstly, it covers a substantial historical timeframe from September 17, 2014, to August 24, 2021, providing a rich dataset for studying Bitcoin's price trends and patterns over a long period. Secondly, the dataset's availability on Kaggle ensures data quality, reliability, and a well-structured format, expediting the data collection process and facilitating subsequent data preprocessing and model development. Lastly, the dataset aligns perfectly with the project's goal of estimating future Bitcoin prices based on historical data, making it an ideal choice for accurate predictive modeling.

**3.4 Model Selection**

Identifying the most suitable machine learning algorithms for Bitcoin price prediction can be complex due to the multitude of available models. To address this, comparative evaluations of some models such as ARIMA, LSTM and VAR were performed to assess the performance of different models on validation data, aiding in selecting the most appropriate one.

LSTM was selected as the primary model for this project following an extensive research and experimentation phase, during which various alternative models were rigorously evaluated and compared.

In the experimental analysis, ARIMA exhibits the lowest error metric; however, it's negative R-squared (R2) score indicates a notably poor performance, implying an inability to capture essential underlying patterns. Meanwhile, LSTM yields a slightly higher error metric than ARIMA but achieves a significantly higher R-squared score, affirming its capability to comprehensively capture the requisite information. Conversely, the VAR model demonstrates the weakest performance among the considered models, rendering it unsuitable for further consideration. Consequently, the selection of LSTM as the preferred model is substantiated by its superior ability to capture essential patterns and trends in the data.

The decision to employ LSTM was predicated on its superior performance, particularly in handling sequential data and capturing intricate temporal dependencies, aligning it effectively with the project's objectives and requirements. This meticulous model selection process ensures the project's robustness and adherence to best practices in data science and forecasting methodologies.

**3.4.1 LONG AND SHORT TERM MEMORY (LSTM)**

Sequence modeling and prediction problems frequently employ the recurrent neural network (RNN) architecture known as LSTM (Long Short-Term Memory). The shortcomings of conventional RNNs in collecting and keeping long-term dependencies in sequential data are addressed by LSTM networks (Selouani, S. A. and Sidi Yacoub, M. (2018)).

**3.4.2 Advantages of LSTM (Wang, Y., et al., 2019):**

* Exceptional at capturing long-term relationships in sequential data due to memory cells, enabling precise predictions.
* Reduces disappearing/expanding gradient issues seen in conventional RNNs, enhancing learning and training stability.
* Versatile and adaptive for various sequential data tasks, including time series forecasting, sentiment analysis, speech recognition, and machine translation.

**3.4.3 Disadvantages of LSTM (Burgueno, L., et al., 2019):**

* High computing complexity and resource-intensive training due to memory cells and gating mechanisms.
* Limited interpretability, often considered as "black box" models, making it challenging to understand inner workings and interpretations.
* Prone to overfitting, especially with limited training data, requiring careful use of regularization techniques like dropout or L2 regularization for balance.

**3.4.4 WORKING OF LSTM**

* Memory Cells: The information in LSTM networks is stored in memory cells over time. These memory cells enable the network to grasp long-term dependencies by preserving pertinent context from previous time steps.
* Gates: To manage the flow of information across LSTM networks, gating methods are used. An LSTM has three different types of gates:
* Forget Gate: The information from the previous time step that is lost or forgotten is decided by the forget gate. It chooses which data to keep in the memory cell based on the input from the most recent time step and the current input.
* The input gate chooses which fresh data from the most recent time step should be kept in the memory cell. After analysing the input, it determines how much of it should be stored in the memory cell.
* Output Gate: The output gate regulates the information output from the memory cell to the forecast for the current time step. It chooses which memory cell components should be displayed as the output.
* Activation Functions: To incorporate non-linearity and enable the network to simulate complicated connections within the data, LSTM networks utilise activation functions. Both the sigmoid function and the hyperbolic tangent (tanh) function are frequently used activation functions in LSTM networks.
* Backpropagation Through Time (BPTT): LSTM networks use the BPTT technique to update the weights and biases of the network throughout the training phase. By backpropagating the mistakes over time to determine the gradients of the network's parameters, BPTT allows the network to develop and modify its internal representations.

When using LSTM to forecast Bitcoin prices, the network learns to map the input sequence to the associated future price using past Bitcoin price data as input. To reduce the prediction error, the network is trained using an optimisation approach such stochastic gradient descent (SGD).

It's crucial to remember that the effectiveness of LSTM for predicting Bitcoin prices depends on a variety of elements, including the calibre and amount of input data, the network architecture used, hyperparameter tweaking, and the intricacies of the implementation(Andrew Pulver, S. L. (2017)).

**3.5 LIBRARIES**

Predicting Bitcoin prices using machine learning involves a combination of data preprocessing, feature engineering, model selection, and evaluation techniques. There are several libraries in Python that can be used to implement various aspects of this process. Here's a list of libraries commonly used in this project:

**Pandas:**

The pandas library is an effective tool for analysing and manipulating data. It offers data structures and operations for working effectively with time series data and other structured data. Pandas may be used to load, preprocess, and analyse historical price data, execute data transformations, and provide datasets for training and evaluating models in the context of predicting the price of Bitcoin(Satori, B. (2023)).

**Numpy:**

Python's numpy library is a key component of the language's scientific computing environment. It offers multidimensional array objects and effective numerical computations. Numpy is frequently used for numerical computations, managing mathematical operations on arrays, and manipulating data structures in the context of forecasting the price of Bitcoin(*NumPy: the absolute basics for beginners — NumPy v1.25 Manual*).

**Datetime:**

Python's datetime module offers classes and methods for dealing with dates and times. It can manage timestamps, translate between various date and time formats, and carry out computations involving periods of time. The datetime library may be used to modify the dataset's time-related characteristics or provide unique time-based features for Bitcoin price prediction (*datetime — Basic date and time types*).

**Sklearn.preprocessing():**

The scikit-learn library's MinMaxScaler class offers a straightforward and widely used method for scaling numerical input. By linearly transforming the original numbers, it converts the data to a certain range, frequently between 0 and 1. Prior to putting them into a machine learning model for Bitcoin price prediction, MinMaxScaler may be used to normalise the input characteristics.

**Sklearn.linear\_model:**

A linear regression model is represented by the scikit-learn class called LinearRegression. It is utilised to fit a linear connection between the target variable and the input characteristics. LinearRegression may be used as a foundation model to build a linear relationship between input characteristics (such as time) and Bitcoin prices in the context of Bitcoin price prediction.

**Sklearn.metrics.mean\_absolute\_error and mean\_squared\_error:**

The mean squared error (MSE) and mean absolute error (MAE) between the predicted and actual values are computed using these scikit-learn functions. These measurements are frequently used to assess how well regression models perform in terms of the precision of predictions and the size of prediction errors.

**Sklearn.neural\_network:**

A multi-layer perception (MLP) regression model is represented by the scikit-learn class MLPRegressor. It is an instance of an artificial neural network model that can discover non-linear correlations between the properties of the input and the variable under study. MLPRegressor may be used to create and train a neural network model for forecasting Bitcoin prices based on historical data in order to predict Bitcoin values.

**Tensorflow.keras.models:**

Sequential is a class found in the TensorFlow's Keras API. For the purpose of creating deep learning models, it represents a linear stack of layers. Sequential is used to characterise the architecture of a neural network model, including the layers and their connections, in the context of predicting the price of Bitcoin.

**Tensorflow.keras.layers Dense and Dropout**

A completely connected layer in a neural network is represented by the Keras class called Dense. It links every neuron in the layer below to every neuron in the layer above. Dropout is a regularisation strategy that, in order to avoid overfitting, randomly eliminates a portion of the neurons during training. These layers are frequently used to specify the structure and behaviour of the neural network in deep learning models for predicting the price of bitcoin.

**Tensorflow.keras.layers.LSTM**

The LSTM layer of a recurrent neural network is represented by the LSTM class from Keras. It was created primarily to handle sequential data and record enduring dependencies. The temporal relationships in the historical Bitcoin price data may be modelled using LSTM layers in Bitcoin price prediction.

**3.6 Metrics**

Evaluating the performance of a Bitcoin price prediction model involves using various metrics to assess how well the model's predictions align with the actual price movements. Below are the evaluation metrics used in this project:

**Root mean square error:**

A frequently used statistic to assess the precision of a prediction or an estimate is root mean square error (RMSE). It is very helpful for assessing the effectiveness of regression models. The ultimate result of RMSE is calculated by taking the square root of the average of the squared discrepancies between the projected values and the actual values (Glen., S. (2023)).

The formula for calculating RMSE is as follows:

Where:

* n is the total number of data points.
* is the actual value (observed or true value) of the target variable for the ith data point.
* ​is the predicted value for the ith data point.

Here's an explanation of the components of the formula:

**Squared Differences:** For each data point, you calculate the squared difference between the actual value () and the predicted value ()​. Squaring the differences ensures that negative differences don't cancel out positive differences, and it gives more weight to larger errors.

**Summation:** You sum up all the squared differences for all data points.

**Average:** You divide the sum of squared differences by the total number of data points (n) to get the average squared difference.

**Square Root:** Finally, you take the square root of the average squared difference to bring the metric back to the same scale as the original data. This is the RMSE, which represents the typical magnitude of errors in your predictions.

The RMSE provides a measure of how well the prediction model's predictions align with the actual data. Smaller values of RMSE indicate that the predictions are closer to the actual values, while larger values indicate greater prediction errors.

**R-Squared:**

R-squared, sometimes referred to as the coefficient of determination, is a statistical metric used to evaluate a regression model's goodness-of-fit. It shows how much of the variance in the dependent variable can be accounted for by the model's independent variables. R-squared has a range of 0 to 1, with 1 denoting a perfect fit and 0 denoting that the model is unable to account for any variation in the dependent variable (GeeksforGeeks (2019)).

The formula for calculating R-squared is as follows:

Where:

* : R-squared coefficient of determination
* SSt: Total Sum of Squares (SSt = Σ(yᵢ - ȳ)²)
* SSe: Sum of Squares of Errors or Residual Sum of Squares² (SSe = Σ(yᵢ - ŷᵢ)²)
* yi: The actual value (observed or true value) of the dependent variable for the ith data point.
* ŷᵢ: The predicted value of the dependent variable for the ith data point.
* ȳ: The mean of the observed values of the dependent variable.

Here's an explanation of the components of the formula:

Numerator: The numerator calculates the sum of squared differences between the actual values (yᵢ) and the predicted values (ŷᵢ). This represents the total variability that is not explained by the model.

Denominator: The denominator calculates the sum of squared differences between the actual values (yᵢ) and the mean of the actual values (ȳ). This represents the total variability in the data.

Difference in Variability: By subtracting the ratio of these two sums from 1, you get a value that indicates the proportion of the total variability in the data that is explained by the model. This is the value.

provides an indication of how well the regression model fits the data. A higher value suggests that a larger proportion of the variability in the dependent variable is explained by the model, indicating a better fit. The was complemented with other evaluation metrics and visualizations to fully understand the performance of the model.

**CHAPTER 4**

**EXPERIMENTS AND IMPLEMENTATION**

**4.1 Libraries**

The pandas library is a powerful tool for analyzing and manipulating data, including time series data and structured data. It can be used to load, preprocess, and analyze historical price data, execute data transformations, and provide datasets for training and evaluating models in the context of predicting the price of Bitcoin.

The numpy library is a key component of the language's scientific computing environment, offering multidimensional array objects and effective numerical computations. It is frequently used for numerical computations, managing mathematical operations on arrays, and manipulating data structures in the context of forecasting the price of bitcoin.

The datetime module in Python offers classes and methods for dealing with dates and times, managing timestamps, translating between various date and time formats, and carrying out computations involving periods of time. It can modify the dataset's time-related characteristics or provide unique time-based features for Bitcoin price prediction.

The MinMaxScaler class in the scikit-learn library is a straightforward method for scaling numerical input, normalizing the input characteristics before putting them into a machine learning model for Bitcoin price prediction. Linear regression models, such as LinearRegression, are used to fit a linear connection between the target variable and input characteristics. The mean squared error (MSE) and mean absolute error (MAE) between predicted and actual values are computed using these scikit-learn functions.

The multi-layer perceptron (MLP) regression model, represented by the MLPRegressor class, is an artificial neural network model that can discover non-linear correlations between the properties of the input and the variable under study. It can be used to create and train a neural network model for forecasting Bitcoin prices based on historical data to predict Bitcoin values.

The TensorFlow Keras API provides a linear stack of layers for creating deep learning models, including Sequential, Dense, and Dropout. These layers are used to specify the structure and behavior of the neural network in deep learning models for predicting the price of bitcoin.

Lastly, the LSTM layer of a recurrent neural network is represented by the LSTM class from Keras, created primarily to handle sequential data and record enduring dependencies. These layers can be used to model temporal relationships in historical Bitcoin price data for Bitcoin price prediction.

**4.2 Data Pre-processing:**

Data preprocessing is a critical phase that involves cleaning, transforming, and preparing the raw data to ensure its quality, consistency, and suitability for model training. The successful completion of this stage sets the groundwork for developing an accurate and reliable predictive model for forecasting cryptocurrency prices.

**4.2.1** **Import Libraries**

The first step in the coding includes importing the libraries that are necessary for the implementation of the code.

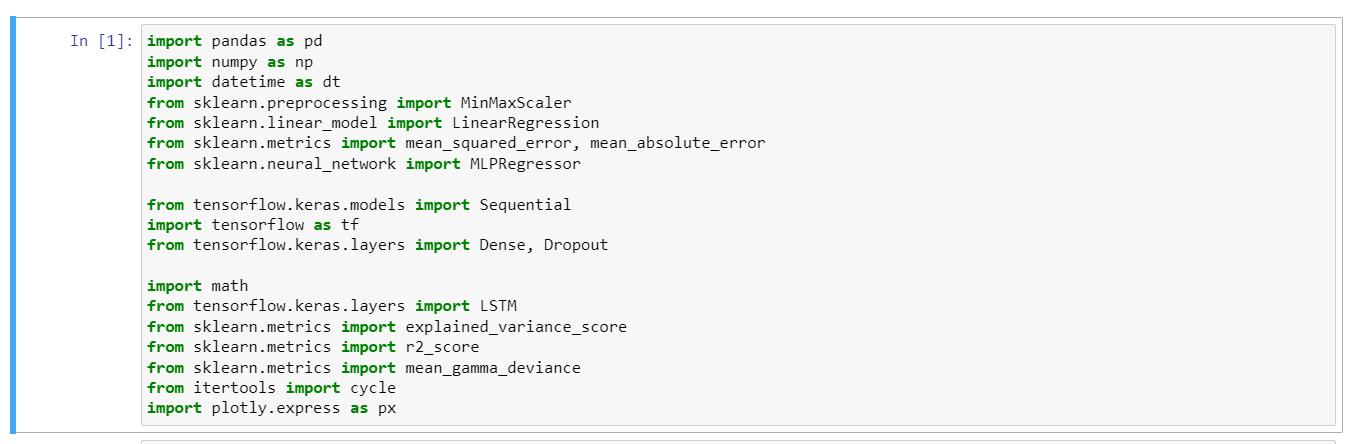


Figure 2: Importing Libraries

**4.2.2** **Load dataset**

The second step is to load the data. The data is loaded into the system using pandas library. The dataset is in the .csv format. The data is then checked if loaded correctly into the system or not.

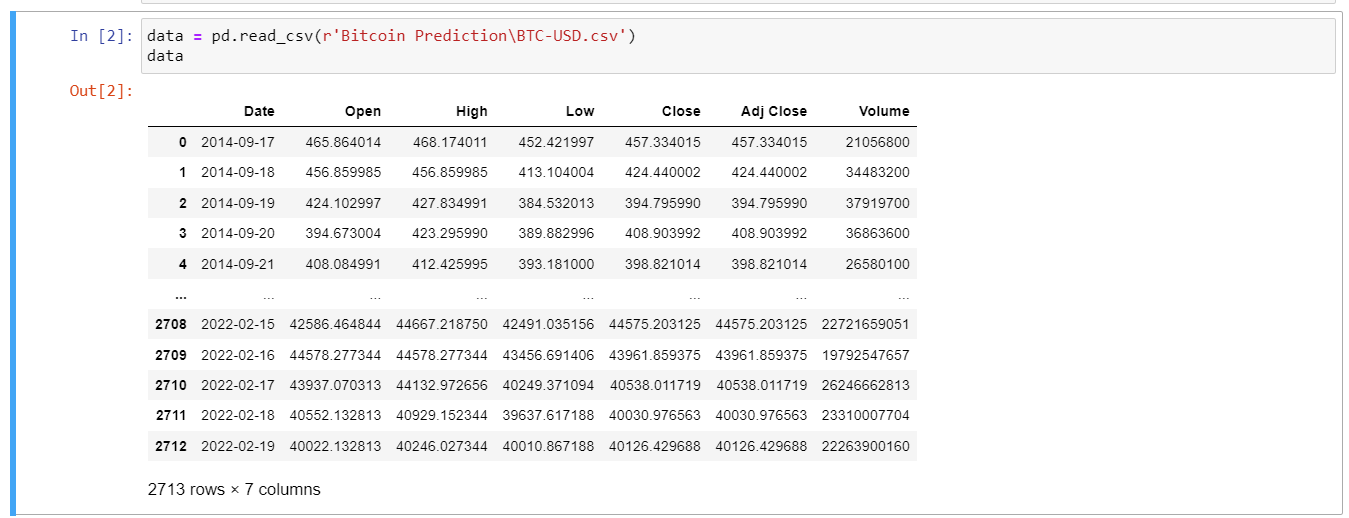


Figure 3: Loading Dataset

**4.2.3 Understand the structure of the dataset**

We use the info() method provided by the Pandas library to get the structure of the data in order to gain more reasonable insights on the structure of a dataset.



Figure 4: Data information

From Figure 4 above, we can observe that the dataset has no null values in all the columns. Five of the columns are of float data type, one is of object data type, and the last one, the volume, is of integer data type.

**4.2.4 Understand the statistical distribution of the data**

We use describe() method to generate summary statistics for a dataset, to get an overview of the central tendencies and distribution of numeric columns within the dataset.

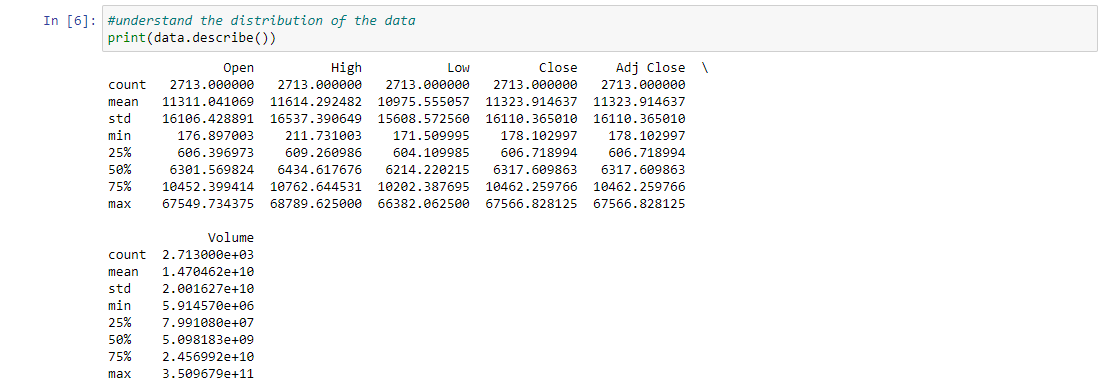


Figure 5: Distribution of the data

From Figure 5 above, we can observe the distribution of the numeric columns. It displays counts, means, standard deviations (std), minimum values (min), 25th percentiles, medians (50%), 75th percentiles, and maximum values (max) for each of the columns. This allows us to gain a deeper understanding of the statistical distribution of the dataset we are working with.

**4.2.5** **Check null values:**

We use isnull() to check if some null or nan values exist in dataset.

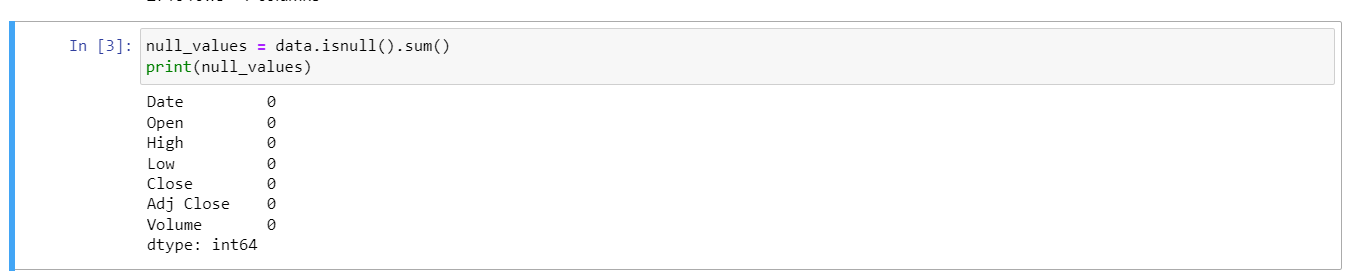


Figure 6: Null Values

Figure 6 above shows that all the columns have no empty or null values, which helps us gain more insight into the dataset. The presence of zero (0) values indicates that there are zero null values.

**4.2.6 Understand the correlation**

The corr() method is used to compute the pairwise correlation coefficients between all pairs of numeric columns in the dataset.

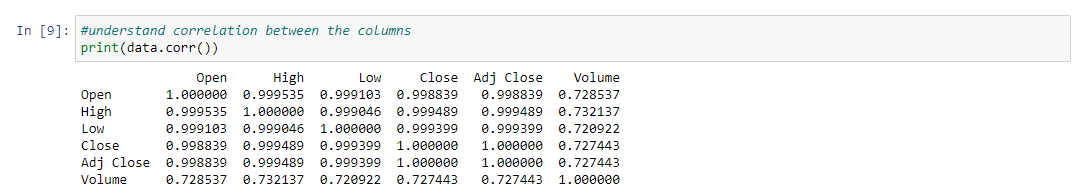


Figure 7: Correlation of the data

In Figure 7 above, there is a perfect positive correlation among all the numeric columns in the dataset, except for the "volume" column, which also exhibits a strong positive relationship. This implies that as one variable increases, the others tend to increase as well. The closer the correlation coefficient is to 1, the stronger the positive relationship. A correlation coefficient of 1 signifies a perfect positive correlation, while a coefficient of 0.7 suggests a strong but not perfect positive relationship.

**4.2.7 Understand the relationship using heatmap**

We used heatmap to visualise the pairwise correlations between variables in a dataset.



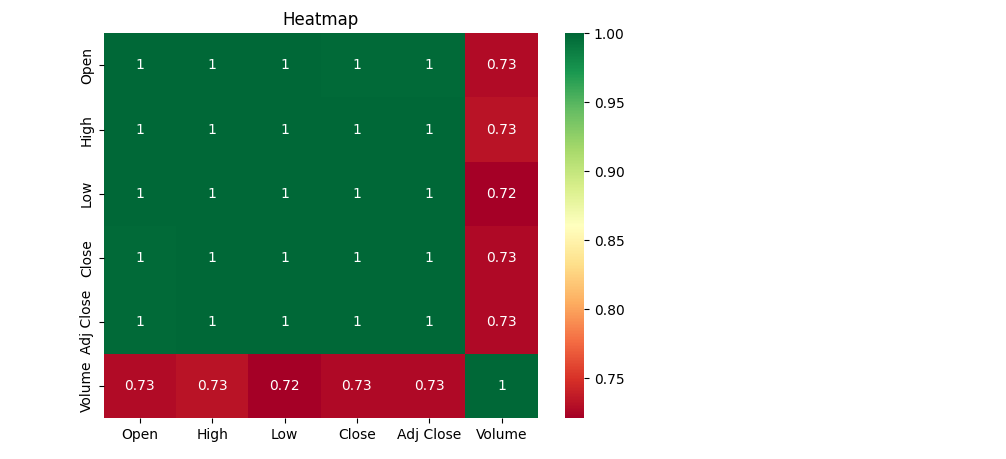


Figure 8: Heatmap

In Figure 8's heatmap above, the green color represents a perfect positive correlation, while the dark red color signifies a strong positive correlation. This aids us in gaining deeper visual insights into the dataset.

**4.2.8 Feature selection:**

We selected the Close price because the Close price is the target feature. As the decisions are made based on the close price we have chosen it as our target variable.



Figure 9: Close column selected

**4.2.9** **Date Format:**

The date given in the dataset is not in the proper format. So, using pandas datetime library we convert date into year, month and date format.

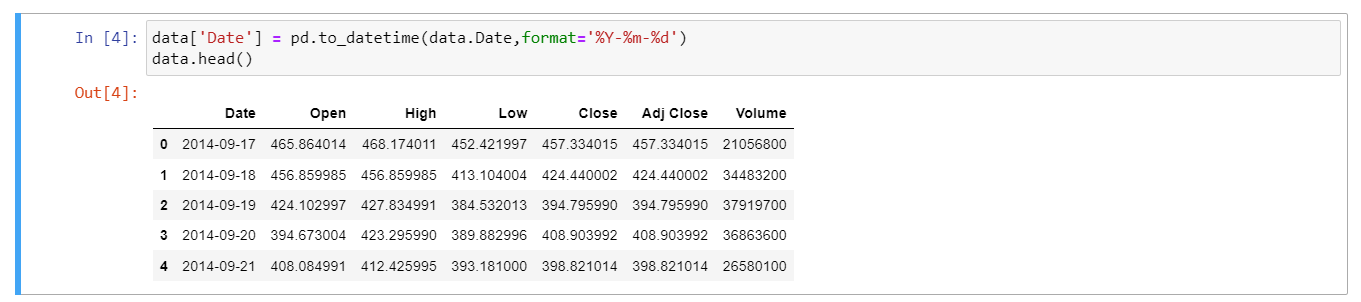


Figure 10: Date Format

The figure displayed above (Figure 10) represents dates in datetime format. In our project, we convert dates to datetime to establish a chronological order, thereby enabling the model to identify temporal relationships and patterns crucial for precise forecasting.

**4.2.10 Lineplot before filtering**

We used the Matplotlib and Seaborn libraries to create a line plot (time series plot) of the "Date" and "Close" price data from the dataset before filtering out the sample data we will use for our project.

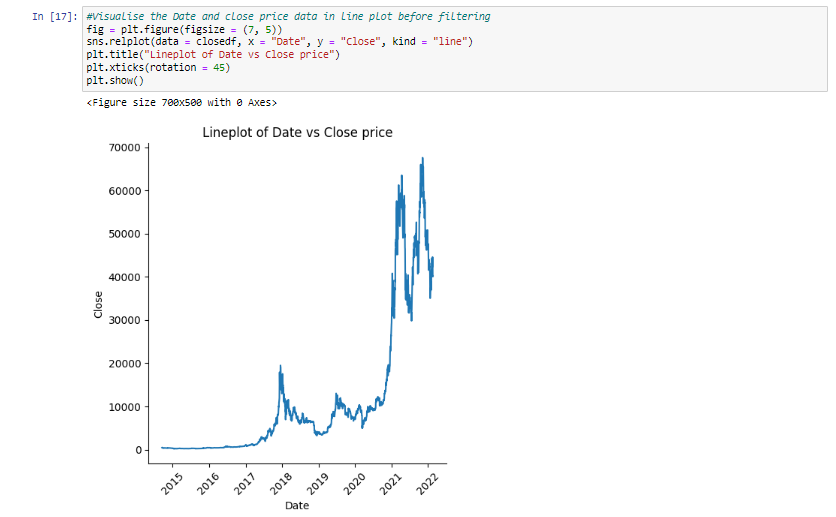
****

Figure 11: Lineplot before filtering

Figure 11, displayed above, illustrates the dataset's trend through a line plot, facilitating our comprehension of the general trajectory and amplitude of changes across time, thus simplifying the identification of patterns, seasonality, and potential anomalies critical for analysis and forecasting. It is evident from the figure that the price of Bitcoin experienced an increase from 2020 to 2022.

**4.2.11 Filter the data**

We filtered the data in the dataset to select the sample of data we will use for our prediction.

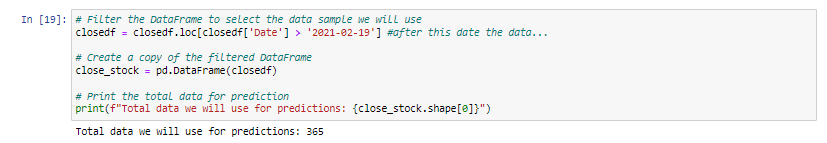
****

Figure 12: Filtered data sample

**4.2.12 Lineplot after filtering**

We used the Matplotlib and Seaborn libraries to create a line plot (time series plot) of the "Date" and "Close" price data from the dataset after filtering the dates we will use to understand the visual aspect of the sample data.



Figure 13: Line plot after filtering

Figure 13, displayed above, illustrates the dataset's trend of the selected sample dataset through a line plot. The plot helps to gain more insights about the price movement of the bitcoin with respect to time.

**4.2.13** **Pre-processing:**

The dataset contains a greater number of columns which are not actually needed in implementation part. So, we create a new data frame which only contains the values of “Date” and “Close” column.



Figure 14: Pre-processing

**4.3** **Train and Test**

The train and test sets refer to the splitting of the available data into two different subsets, the training set and the testing set, while discussing Bitcoin prediction. To properly create and evaluate prediction models, this divide is essential.

The prediction model is created using the training set. It is the piece of the data from which the model infers the underlying trends and connections between the input characteristics (such as past Bitcoin prices, trading volumes, and market indicators) and the target variable (such as projected Bitcoin prices). In order to recognise and capture these patterns, the model is trained using a variety of methods and methodologies.

After the model has been trained, it must be assessed on new data to determine its effectiveness and generalizability. The testing set is useful in this situation. The testing set is a unique subset of data that the model hasn't seen before during training. It is used to test how successfully the model can forecast future Bitcoin values by simulating real-world circumstances.

We may evaluate how correctly the model operates on fresh, untested data by utilising a different testing set. This assessment aids in determining the model's predictive power and its capacity to extrapolate beyond the training set of data.

The train-test split is often carried out at random to make sure that both sets accurately reflect the distribution of the data as a whole. Due to the need for additional data to adequately train the model, the training set is often bigger than the testing set. Typical splits, when the training set constitutes a bigger amount, are 60-40, 70-30, 80-20, or 90-10.

It's crucial to keep in mind that the train-test split should be carried out chronologically, particularly in time-series forecasting jobs like predicting the price of bitcoin. As a result, the model will more precisely represent real-world prediction situations since it will have been trained on historical data and evaluated on future data.

For the chosen dataset the data is split in the ratio of 60:40 for train and test respectively.

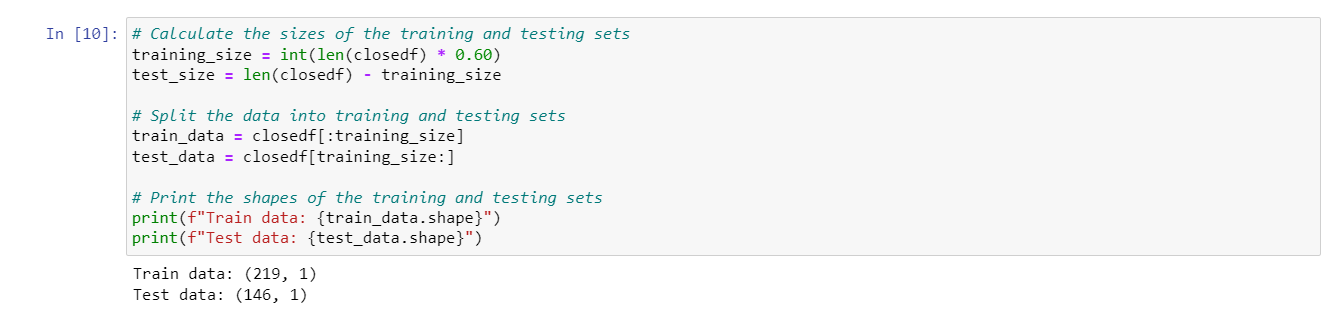


Figure 15: Train Test Split

**4.3.1** **Training LSTM Model**

The LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) that is well-suited for sequence prediction tasks, such as time series forecasting. Below are the approach used for training an LSTM (Long Short-Term Memory) model for this project:

**Initialising the model:**

* A linear stack of layers is built using the Sequential model from Keras.
* An LSTM layer of 10 units, commonly known as memory cells, makes up the top layer.
* The model may take input sequences of variable length (None), with one feature (1), because the input\_shape parameter is set to (None, 1).
* Rectified Linear Unit ("relu") is the activation function employed for the LSTM layer.
* The output layer is represented by the second layer, which is a Dense layer with 1 unit.

**Compilation of models:**

* The "mean\_squared\_error" loss function, which calculates the model, calculates the mean squared difference between the predicted and actual values.
* The model's weights are optimised, and the training loss is reduced, using the optimizer "adam".

**Model Training:**

* The fit() method is used to train the model.
* The training set's input sequences and corresponding target values are represented, respectively, by the variables X\_train and y\_train.
* When training a model, the validation\_data parameter is set to (X\_test, y\_test) to assess how well it performed on the testing set.
* The model will run through the full training dataset 200 times, which represents the number of training epochs.
* The number of samples required to update the model's weights after each iteration is indicated by the batch\_size parameter, which is set to 32.
* To prevent reporting of intermediate training progress during training, the verbose option is set to 0.

**Final Loss:**

The "final loss" is the value of the loss function after the training process is complete, or after a certain number of training iterations (epochs). This value indicates how well the trained model fits the training data. A lower final loss generally indicates that the model has learned to make better predictions and has successfully captured the patterns and relationships present in the training data.

* The history object, which holds the loss values registered throughout training, is used to determine the ultimate loss value.
* The loss values for each training period are kept in the history - history dictionary.
* Indexing into the 'loss' key and choosing the last element with [-1] will give you access to the last loss.

**Printing the Total Loss:**

* The print() method is used to print the final loss value.



Figure 16: Training LSTM Model

**4.3.2 Training and Validation**

Two crucial parameters, training loss and validation loss, are utilised to evaluate the effectiveness and development of a machine learning model during training. These metrics offer information on how successfully the model generalises to new data and learns from the training data.

**Training Loss:**

The mistake or difference between the values of the training dataset's projected values and actual values is measured as training loss, also known as training error or training goal.

To enhance its predicting capacity on the training data, the model iteratively adjusts its parameters during training to minimise the training loss.

The objective is for the training loss to decrease over time, showing that the model is adapting and getting better at fitting the training data.

The model's performance on unobserved data cannot be determined just by the training loss.

**Validation Loss:**

The validation loss, sometimes referred to as the validation error, is a measurement of the inaccuracy or disparity between the expected values and the actual values of a different validation dataset that the model has not encountered before.

The model's performance on unobserved data is evaluated using the validation dataset, which is used to imitate real-world circumstances.

We can assess how effectively the model generalises outside of the training set by keeping an eye on the validation loss.

The ideal situation would be for the training loss and the validation loss to both decreases. It may be a sign of overfitting, when the model gets overly specialised to the training data and struggles to generalise, if the validation loss starts to rise while the training loss keeps falling.

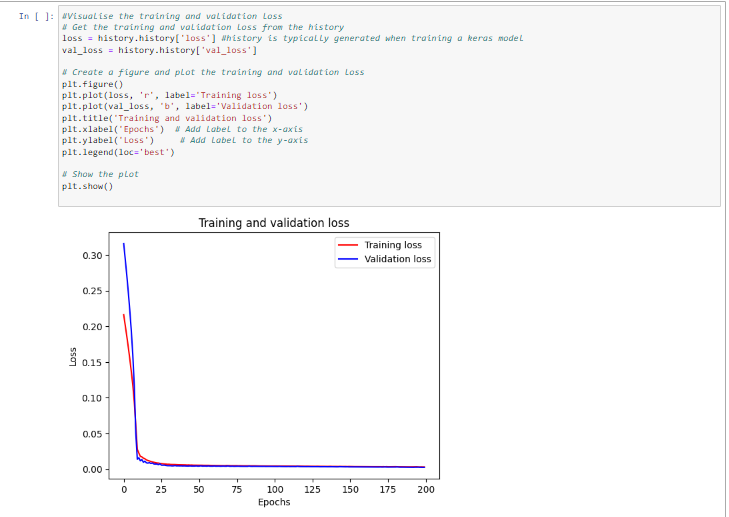


Figure 17: Training and Validation Loss

The plot displayed in Figure 17 above illustrates the evolution of training and validation loss across epochs, a consistent decrease in both training and validation loss suggests that the model is effectively learning and generalizing, as demonstrated in the graph.

**CHAPTER 5**

**EVALUATION AND RECOMMENDATION**

In this chapter we will evaluate the performance of the trained LSTM model for Bitcoin price prediction and provide recommendations based on the results.

**5.1 EVALUATION**

**5.1.1 RMSE, or Root Mean Square Error**

The average difference in magnitude between the predicted and actual values in the training set is around 1906.79 units, according to the train data RMSE of 1906.79.

The test set's RMSE of 1960.54 shows that the average difference in magnitude between predicted and actual values is around 1960.54 units.

We can tell that the test RMSE is somewhat greater than the train RMSE by comparing the two RMSEs. This shows that the model may have a little poorer prediction accuracy on test data that hasn't been seen compared to the training data.

**5.1.2** **MSE, or Mean Squared Error**

The average squared difference between the predicted and actual values in the training set is represented by the train data MSE, which is 3635830.21.

The average squared difference between the test set's actual and anticipated values is represented by the MSE of 3843722.09 for the test data.

The test MSE is marginally greater than the train MSE, similar to RMSE, suggesting a marginally larger prediction error on the test data.

**5.1.3** **MAE: Mean Absolute Error**

The average absolute difference between the predicted and actual values in the training set is represented by the train data MAE of 1454.77.

The average absolute difference between the test set's actual and projected values is represented by the MAE of 1533.83.

The test MAE is somewhat greater than the train MAE, which indicates slightly bigger absolute errors in predictions on the test data, when the two MAEs are compared.



Figure 18: MSE, MAE, RMSE

**5.1.4** **R Squared -** **Coefficient of Determination (R2) Score**

When the model is tested on the training set, the train data R2 score of 0.958 shows that around 95.8% of the variation in the target variable is explained by the independent variables in the model.

When the model is tested on the test set, the test data R2 score of 0.953 shows that the independent variables in the model account for around 95.3% of the variation in the target variable.

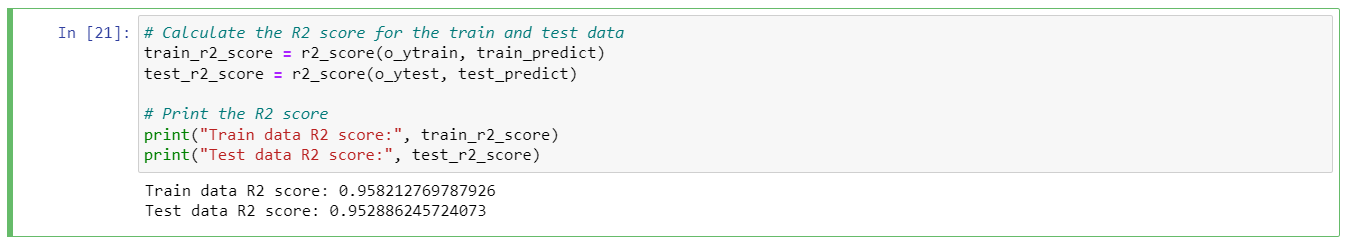


Figure 19: R Square Error

**5.1.5 Score from the Train Data Explained Variance Regression:**

When the model is tested on the training set, the train data explained variance regression score of 0.958 shows that around 95.8% of the variation in the target variable is explained by the independent variables in the model.

The model may be able to successfully capture and explain the underlying patterns and connections in the training data if it receives a score that is close to 1.0.

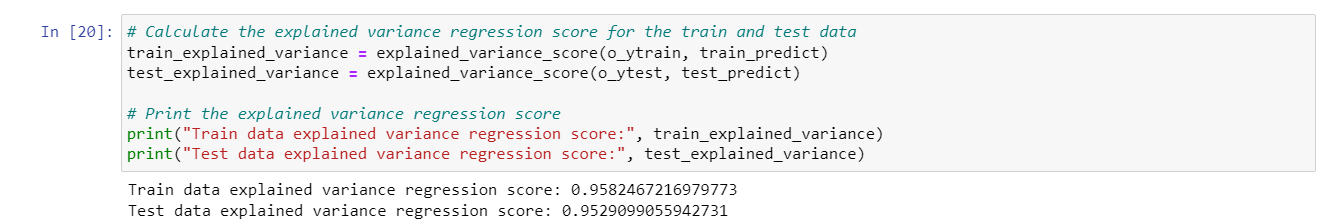


Figure 20: Explained Variance Ratio

**5.1.5.1 Score for Explained Variance Regression Using Test Data:**

When the model is assessed using the test set, the test data explained variance regression score of 0.953 shows that around 95.3% of the variation in the target variable is explained by the independent variables in the model.

The somewhat lower test data score indicates that, when used with unknown test data, the model's ability to explain the variance in the target variable is significantly less effective. The score is still rather high, demonstrating the model's strong generalisation capabilities.

**5.2 Comparison between original data and predicted data**

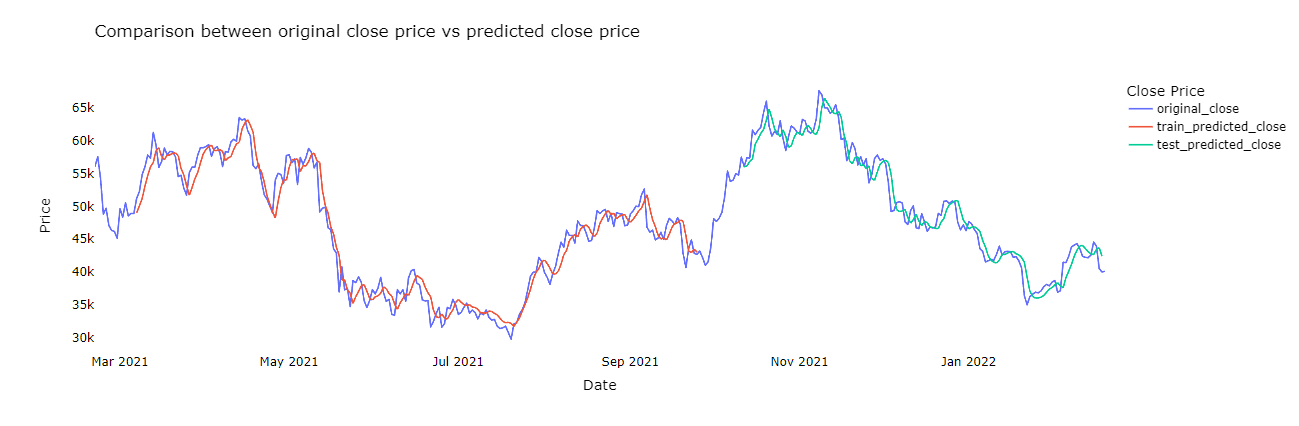
The subjoined visualization illustrates a comparative analysis between the raw dataset and the predictions generated by the trained model for both the training and testing datasets. This analysis underscores the pronounced similarity between the model's predictions and the actual data, affirming the model's capacity for accurate forecasting.****

Figure : comparison between the original data and the predicted train and test data

**5.3 Recommendations**

The evaluation of the LSTM model's performance in Bitcoin price prediction has provided valuable insights. To further enhance the accuracy and reliability of the model, the following recommendations are put forward:

**Hyperparameter Optimization:**

Conduct a comprehensive hyperparameter tuning process to identify the optimal configuration for the LSTM model. Experiment with different numbers of LSTM units, layers, batch sizes, and learning rates to fine-tune the model's performance.

**Feature Engineering and Selection:**

Refine the feature set by exploring additional relevant indicators and incorporating external factors. Experiment with a combination of technical indicators, sentiment analysis, trading volumes, and macroeconomic indicators to capture a more comprehensive view of market dynamics.

**Temporal Feature Exploration:**

Investigate the impact of various time intervals for features. Experiment with different time horizons for technical indicators and lagged variables to identify the timeframes that yield the most accurate predictions.

**Handling Market Volatility:**

Develop strategies to improve the model's ability to handle sudden price fluctuations and market irregularities. Investigate advanced techniques such as attention mechanisms within LSTM to focus on critical periods of volatility.

**Ensemble Strategies:**

Explore ensemble methods that combine the predictions of multiple models. Consider blending LSTM predictions with those of other algorithms, such as ARIMA or XGBoost, to leverage their complementary strengths.

**Cross-Validation Techniques:**

Implement cross-validation techniques to assess the model's robustness across different data subsets. This will provide a more accurate estimate of the model's performance on unseen data and its ability to generalize.

**Continuous Learning and Updating:**

Establish a regular update schedule for retraining the model with fresh data. Implement an incremental learning approach that adapts to evolving market conditions and captures the most recent trends.

**Risk Management Protocols:**

Integrate rigorous risk management protocols when applying the model's predictions for trading decisions. Implement stop-loss mechanisms, position sizing strategies, and backtesting frameworks to minimize potential losses.

**Interpretability and Explanation:**

Invest in methods that enhance the interpretability of the LSTM model. Consider techniques like SHAP (SHapley Additive exPlanations) to explain individual predictions and feature contributions.

In conclusion, these recommendations aim to build upon the strengths of the LSTM model while addressing its limitations. By refining hyperparameters, diversifying features, and adapting to market dynamics, the model can provide more accurate predictions and contribute to informed decision-making in the complex domain of Bitcoin price forecasting.

**CHAPTER 6**

**CONCLUSION AND LIMITATIONS**

**6.1 Conclusion**

This research project set out to develop a predictive model for Bitcoin price movements using machine learning algorithms. The primary aim was to provide valuable insights to cryptocurrency market stakeholders, including investors and traders. The project successfully achieved its objectives by collecting and preprocessing cryptocurrency data, selecting suitable machine learning algorithms, developing and training predictive models, and evaluating their performance using various metrics, including RMSE, MSE, MAE, R-squared, and explained variance regression.

Throughout the project, meticulous attention was paid to every stage of the process, ensuring the data used for training and testing the machine learning models was of the highest quality. The selection of suitable machine learning algorithms was a crucial decision, and a comprehensive comparative evaluation of models like ARIMA, LSTM, and VAR was conducted. This evaluation led to the informed choice of LSTM as the primary model, due to its superior performance in capturing essential patterns and trends in the Bitcoin price data.

Furthermore, the visual comparison between the model's predictions and actual data underscored its accuracy in forecasting Bitcoin prices. These findings validate the effectiveness of machine learning techniques in predicting cryptocurrency prices, aligning with the primary research question.

As a result, this research contributes valuable insights to the field of cryptocurrency analysis and prediction, offering a robust methodology and model selection process that can be applied to similar research in the future.

**6.2 LIMITATIONS**

There are several limitations that need to be acknowledged. Here are some common limitations that you might want to consider:

Market Volatility: The cryptocurrency market, including Bitcoin, is highly volatile. Sudden and unexpected price fluctuations can lead to challenges in accurately predicting prices, even with sophisticated machine learning models.

Non-Stationarity: Financial time series data, including Bitcoin prices, often exhibit non-stationarity, meaning that statistical properties can change over time. This can make it challenging to build models that can capture long-term trends and patterns effectively.

Limited Historical Data: Although historical data is essential for training models, the relatively short history of Bitcoin might limit the effectiveness of long-term predictions. Predictions for longer time horizons might be less reliable due to the limited amount of historical data available.

External Factors: Cryptocurrency prices can be influenced by external factors such as regulatory changes, news sentiment, macroeconomic events, and technological advancements. These factors are challenging to quantify and incorporate into predictive models.

Data Quality and Integrity: Cryptocurrency markets are also susceptible to data quality issues such as missing data, outliers, and irregularities. Low-quality data can negatively impact the performance of predictive models.

Overfitting and Generalization: Overfitting is a concern when training complex machine learning models on limited data. A model that fits the training data well might not generalize to unseen data, leading to poor predictive performance.

Model Assumptions: Many machine learning algorithms assume that data is generated from a certain distribution or follows specific patterns. These assumptions might not hold true in the case of cryptocurrency markets.

Risk Management and Financial Decisions: It's important to emphasize that even accurate predictions do not guarantee successful trading or investment outcomes. Implementing predictive models for real trading decisions involves significant financial risk and requires effective risk management strategies.

Model Complexity: Building sophisticated models like LSTM requires a good understanding of deep learning concepts. Complex models might be challenging to interpret and explain, which can be important for decision-making in real-world applications.

Dynamic Nature of Technology: The field of cryptocurrencies and blockchain technology is rapidly evolving. The introduction of new technologies, protocols, and market dynamics can render historical patterns less relevant over time.

It's crucial to address these limitations in your project to provide a balanced and realistic view of the potential challenges and uncertainties associated with Bitcoin price prediction using machine learning. Additionally, you can discuss strategies to mitigate these limitations and the broader implications of your findings for the cryptocurrency market and beyond.

# Bibliography

Agarwal, A. *et al.* (2021) ‘Prediction of Bitcoin, Litecoin and Ethereum trends using State-of-Art Algorithms’, in *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*. IEEE, pp. 538–545.

Agrawal, N. and Bhardwaj, A. (2019) ‘Bitcoin price prediction using news sentiment analysis with machine learning’, in *Proceedings of the International Conference on Advances in Computing, Communication & Automation*, pp. 1–6.

Andrew Pulver, S. L. (2017) *LSTM with working memory*, *Ieee.org*. Available at: https://ieeexplore.ieee.org/document/9200054 (Accessed: 5 July 2023).

Atsalakis, G. S. and Valavanis, K. P. (2015) ‘Forecasting exchange rates and Bitcoin prices with a composite nonlinear time series model’, *Journal of International Financial Markets*, 35, pp. 143–162.

Bouri, E., Azzi, G. and Dyhrberg, A. H. (2017) ‘On the return-volatility relationship in the Bitcoin market around the price crash of 2013’, *Economics The Open-Access Open-Assessment E-Journal*, 11(1). doi: 10.5018/economics-ejournal.ja.2017-2.

Bouri, E., Azzi, G. and Dyhrberg, A. H. (2019) ‘On the return-volatility relationship in the Bitcoin market around the price crash of 2013: A note’, *Finance Research Letters*, 31, pp. 91–96.

Burgueno, L., Cabot, J. and Gerard, S. (2019) ‘An LSTM-based neural network architecture for model transformations’, in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*. IEEE, pp. 294–299.

Chen, W.; Xu, H.; Jia, L.; Gao, Y. "Machine Learning Model for Bitcoin Exchange Rate Prediction Using Economic and Technology Determinants." International Journal of Forecasting, 2021, 37, 28–43. [Google Scholar] [CrossRef]

Cheah, E. T., Mishra, T. and Parhi, M. (2021) ‘Bitcoin price prediction using machine learning: An approach with consensus of multiple classifiers’, *Applied Soft Computing*, 108.

*datetime — Basic date and time types* (no date) *Python documentation*. Available at: https://docs.python.org/3/library/datetime.html (Accessed: 5 July 2023).

Fernandes, M. *et al.* (2021) ‘Bitcoin Price Prediction’, in *2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)*. IEEE, pp. 1–4.

Fischer, T., Krauss, C. and Treffenstädt, D. (2019) ‘Deep learning with long short-term memory networks for financial market predictions’, *European Journal of Operational Research*, 276(2), pp. 770–785.

Fledelius, P., Santos, J. C. and Würtz, D. (2021) ‘Modeling cryptocurrency returns with multiple machine learning algorithms’, *Journal of Financial Markets*.

Follow, V. (2020) *SEMMA model*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/semma-model/ (Accessed: 5 July 2023).

Garcia, D. *et al.* (2014) ‘The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy’, *Journal of the Royal Society, Interface*, 11(99), p. 20140623. doi: 10.1098/rsif.2014.0623.

Garcia, D., Tessone, C. J. and Mavrodiev, P. (2014) ‘Measuring the efficiency of the Bitcoin market’, *Scientific Reports*.

GeeksforGeeks (2019) *R-squared in regression analysis in machine learning*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/ml-r-squared-in-regression-analysis/ (Accessed: 6 July 2023).

Glen., S. (2023) *RMSE: Root Mean Square Error*, *Statisticshowto.com*. Available at: https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/ (Accessed: 6 July 2023).

Hitam, N.A.; Ismail, A.R.; Samsudin, R.; Alkhammash, E.H. "The Effect of Kernel Functions on Cryptocurrency Prediction Using Support Vector Machines." In Proceedings of the International Conference of Reliable Information and Communication Technology; Springer: Cham, Switzerland, 2022; pp. 319–332. [Google Scholar]

Katsiampa, P. (2017) ‘Volatility estimation for Bitcoin: A comparison of GARCH models’, *Economics letters*, 158, pp. 3–6. doi: 10.1016/j.econlet.2017.06.023.

Kazeminia, S., Sajedi, H. and Arjmand, M. (2023) ‘Real-time bitcoin price prediction using hybrid 2D-CNN LSTM model’, in *2023 9th International Conference on Web Research (ICWR)*. IEEE, pp. 173–178.

Kondor, D. *et al.* (2014) ‘Do the rich get richer? An empirical analysis of the bitcoin transaction network’, *PloS one*, 9(2), p. e86197. doi: 10.1371/journal.pone.0086197.

Kristoufek, L. (2013) ‘BitCoin meets Google Trends and Wikipedia: quantifying the relationship between phenomena of the Internet era’, *Scientific reports*, 3(1), p. 3415. doi: 10.1038/srep03415.

Kristoufek, L. (2015) ‘What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis’, *PloS one*, 10(4), p. e0123923. doi: 10.1371/journal.pone.0123923.

Li, J. (2021) ‘A hybrid ARIMA model with sentiment analysis for Bitcoin price forecasting’, *International Journal of Finance & Economics*, pp. 1–18.

Malhotra, B. *et al.* (2022) ‘Bitcoin price prediction using machine learning and deep learning algorithms’, in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. IEEE, pp. 1–6.

Mcafee, A. and Brynjolfsson, E. (2017) *Machine, platform, crowd: Harnessing our digital future*. W. W. Norton & Company.

Meet N. (2021) 'Bitcoin Stock Data (Sept 17, 2014 - August 24, 2021)'. Available at <https://www.kaggle.com/datasets/meetnagadia/bitcoin-stock-data-sept-17-2014-august-24-2021> (Accessed 16 June 2023).

Mittal, M. and Geetha, G. (2022) ‘Predicting Bitcoin Price using Machine Learning’, in *2022 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, pp. 1–7.

Moazami, H. *et al.* (2021) *Forecasting the price of Bitcoin using machine learning techniques. Scientific Programming*.

*NumPy: the absolute basics for beginners — NumPy v1.25 Manual* (no date) *Numpy.org*. Available at: https://numpy.org/doc/stable/user/absolute\_beginners.html (Accessed: 5 July 2023).

Oikonomopoulos, S. *et al.* (2022) ‘Cryptocurrency price prediction using social media sentiment analysis’, in *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, pp. 1–8.

Perianez, A. (2018) ‘Forecasting Bitcoin volatility: a comparative analysis of GARCH, SVR, and LSTM-ARMA models’, *Journal of Risk and Financial Management*, 11(3).

Satori, B. (2023) *Pandas*, *https://pypi.org/*. North Charleston, SC: Createspace Independent Publishing Platform.

Saad, M.; Choi, J.; Nyang, D.; Kim, J.; Mohaisen, A. "Toward Characterizing Blockchain-based Cryptocurrencies for Highly Accurate Predictions." IEEE Systems Journal, 2019, 14, 321–332. [Google Scholar]

Selouani, S. A. and Sidi Yacoub, M. (2018) ‘Long short-term memory neural networks for artificial dialogue generation’, in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, pp. 761–768.

Shahzad, S. A., Zaho, F. and Qazi, W. (2020) ‘Hybrid deep learning models for bitcoin price prediction’, in *Proceedings of the International Conference on Information Science and Communications Technologies*, pp. 1–6.

Siami-Namini, S. and Trabelsi, M. M. (2019) ‘Modeling and forecasting Bitcoin price returns: A combination of GARCH, ARIMA and NAR Neural Network models’, *Journal of Risk and Financial Management*, 12(1).

Silva, D. *et al.* (2019) ‘Bitcoin price forecasting with sentiment analysis: An empirical investigation’, *International Review of Financial Analysis*, 63, pp. 49–57.

Singh, V. K. and Srivastava, G. (2021) ‘A comprehensive review of bitcoin price prediction using machine learning approaches’, *Journal of Ambient Intelligence and Humanized Computing*, 12(8), pp. 10765–10783.

Wang, Y., Zhu, S. and Li, C. (2019) ‘Research on multistep time series prediction based on LSTM’, in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*. IEEE, pp. 1155–1159.

Xu, K. (2021) ‘Forecasting Bitcoin price with artificial neural network’, *The North American Journal of Economics and Finance*.

Zhang, H., Shen, D. and Li, X. (2020) ‘Bitcoin price prediction based on support vector machine’, *Journal of Computational Science*.

Zhang, Y., Fuehres, H. and Gloor, P. A. (2018) ‘Predicting cryptocurrency prices with machine learning’, in *International conference on collaboration and internet computing*. IEEE, pp. 81–88.

Zheng, Z. *et al.* (2018) ‘LSTM network: A deep learning approach for Bitcoin price prediction’, *Journal of Engineering Research*, 6(1), pp. 38–44.

1. <https://www.kaggle.com/datasets/meetnagadia/bitcoin-stock-data-sept-17-2014-august-24-2021> [↑](#footnote-ref-2)