



**“ARTIFICIAL INTELLIGENCE (AI) BASED
CRYPTOCURRENCY PRICE FORECASTING
SYSTEM UNDER CHANGING MARKET
CONDITION”**

**Kiran Babu Basnet
B00728243**

Thesis for the partial fulfillment
of the requirements for the Masters’s Degree
in M.S.c Advance Computing(BIG DATA)

University of the West of Scotland
School of Computing, Engineering and Physical Sciences
19th – March-2023

DECLARATION OF ORIGINALITY

I declare that this is an original study based on my own.
work and that I have not submitted it for any other
course or degree.

Signature Kiran

Library Form to Accompany MSc Project

TO BE COMPLETED IN FULL

Surname: Basnet	
First Name: Kiran Babu	Initials: Mr.
Banner No: B00728243	
Course Code: COMP11024	
Course Title: Master's Project 202122 T3	
Project Supervisor: Prof. Keshav Dahal Moderator: Dr. Ravi Koirala	
Project Title: Artificial Intelligence (AI) based cryptocurrency price forecasting system under changing market condition.	
Session: 2022/2023	Date of Submission: 19th March 2023
Signature: Kiran Babu Basnet	

Abstract

This research project aims to forecast the dominant cryptocurrency in the rapidly fluctuating market. Cryptocurrency is becoming increasingly popular in the world's financial landscape with a growing user base. In this age of information, every individual is a stakeholder in some way, directly or indirectly. The financial marketplace is also becoming more complex with technological advancements, providing better opportunities. Data-driven statistical modelling with artificial intelligence, especially machine learning approaches, has gained significant importance in the financial marketplace. Through an intensive literature review, the most discriminating factors have been identified, and their historical data will be used to feed advanced artificial intelligence frameworks such as recurrent neural network (RNN)based machine learning models viz, long short-term memory (LSTM), and regression-based Python programming machine learning library viz, PyCaret to predict the high-frequency time series of prices of highly impactful cryptocurrencies such as Bitcoin (BTC). The feature set includes technical(statistical) analysis, blockchain-based analysis, analysis, and asset-based analysis. This project will develop a complete data science process, using the most discriminating features identified, and appropriate machine learning models to forecast the future. It will explain the end-to-end process of forecasting, including data sourcing, data preparation, explanatory data analysis (EDA), feature engineering, feature selection, model training techniques, model performance comparison and interpretation and prediction. Significant recent downturns to the bitcoin, this project is able to follow the train and simulate the close price line graph with predicted line graph with very high correlation.

Keywords: Artificial Intelligence (AI), Machine Learning, Cryptocurrency, Financial Forecasting, Time series, Blockchain, Bitcoin

Contents

Abstract.....	i
Acknowledgement.....	v
List of Figure.....	vi
List of Abbreviation	ix
Chapter 1.....	1
Introduction.....	1
1.1 Overview	1
1.2 Problem Statement	2
1.3 Relevance of the Study.....	3
1.4 Aims	3
1.5 Objectives	4
1.6 Justification.....	4
1.7 Project Outline	5
Chapter 2.....	7
Literature Review	7
2.1 Blockchain Technology Background and its Development	7
2.2 Cryptocurrency	8
2.3 Understanding Bitcoin.....	10
2.4 The most Discriminating Features of Cryptocurrency Price Forecasting	12
2.5 Time Series	13
2.6 Financial Market Scenario.....	15
2.7 Forecast Ability.....	16
2.8 Artificial Intelligence (AI) and Machine Learning (ML).....	17
2.9 Can AI/ML Models Predict Short-term Movements of the Cryptos Market	18
2.10 Background Reading for Machine Learning Algorithm	18
2.10.1 Regression	18
2.11 Neural Network.....	19
2.12 Long Short-Term Memory (LSTM).....	21
2.12.1 LSTM Cycle	23
2.12.2 Sigmoid Activation Function	25
2.12.3 Tanh Activation Function	25
2.13 Bidirectional LSTM	27
2.14 PyCaret Model	28
2.14.1 PyCaret Regression Model.....	29
Chapter 3.....	30
Research Design and Methodology	30

3.1 Data Preprocessing	30
3.2 Data / Data Set	31
3.3 EDA (Exploratory Data Analysis) - Qualitative	31
3.4 ML (Machine Learning)-Qualitative	32
3.5 Core Solution	33
3.6 Forecasting Process	34
3.7 Feature Exploration	34
3.8 Big Data Storage and Computation	35
Chapter 4	36
Exploratory Data Analysis	36
4.1 Data Set	36
4.2 Data Preprocessing	37
4.3 Installing and Importing Necessary Dependencies	38
4.4 Data Cleaning	40
4.5 EDA Process	41
4.5.1 Statistical Analysis	41
4.5.2 Visualization Analysis	44
Chapter 5	51
Feature engineering, Training Testing and Model Building	51
5.1 Feature Engineering	51
5.1.1 Normalization	51
5.2 Training Testing and Model Building	52
5.2.1 For LSTM Model	52
5.2.2 For Pycaret Model	56
5.3 Prediction	64
5.3.1 By LSTM Model	64
5.3.2 PyCaret Model	65
Chapter 6	67
Result, conclusion and discussion	67
6.1 Result	67
6.1.1 PyCaret Model	67
6.1.2 By LSTM Model	68
6.2 Conclusion	68
6.3 Future Improvement	69
6.4 Critical Review	70
References	72
Appendix:1	75
Project Process Documentation	75

Project Process Documentation:1.....	75
Project Process Documentation:2.....	75
Project Process Documentation:3.....	76
Project Process Documentation:4.....	76
Project Process Documentation:5.....	77
Project Process Documentation:6.....	77
Project Process Documentation:7.....	78
Project Process Documentation:8.....	78
Appendix:2.....	79
Code File	79
Strep-by-step guide to run the project.....	79
References	81

Acknowledgement

This study entitled, “**ARTIFICIAL INTELLIGENCE (AI) BASED CRYPTOCURRENCY PRICE FORECASTING SYSTEM UNDER CHANGING MARKET CONDITION**” has been prepared for the partial fulfilment of the requirements for the Masters’ Degree in M.Sc. Advance Computing (BIG DATA). University of the West of Scotland School of Computing, Engineering and Physical Sciences. I take this opportunity to express my sincere gratitude to all my teachers at the Research Department University of the West of Scotland who have helped me through my research project. My limitless thanks go to our supervisor, **Prof. Keshav Dahal** and our moderator **Dr Ravi Koirala** for seeing each and every related work carefully during the time of research and also for continuous support. The constructive criticisms have helped me focus more from the beginning till the end of this research. I would also like to thank our module coordinator **Dr Daune West** for her useful feedback, and I do express my sincere gratitude to our module leader **Prof. Naeem Ramzan** for her support and guidance in the completion of this project. I would also like to thank my parents for continuously taking care of my health. during the research time. Without their support, it would not have been possible.

Finally, I would like to extend my sincere thanks to all those people who helped to execute the project directly and indirectly to complete this research project.

Thank you.

List of Figure

Figure 2.1: Crypto Operation.....	8
Figure 2.2: Coin Toss Condition.....	9
Figure 2.3: Process Flow Chart.....	10
Figure 2.4: Recurrent Neural Network.....	20
Figure 2.5: Different Gates in LSTM Model.....	22
Figure 2.6: Working of Gates in LSTM Model.....	23
Figure 2.7: Lstm Cycle.....	24
Figure 2.8: Sigmoid Activation Function.....	25
Figure 2.9: Tanh Activation Function.....	26
Figure 2.10: LSTM Model	26
Figure 3.1: Methodology and Model Selection	30
Figure 3.2: EDA Process	32
Figure 3.3: Core Solution for Crypto-currency Forecasting	33
Figure 3.3: Forecasting Process	34
Figure 4.1: Data Set for Bitcoin.....	37
Figure 4.2: Importing Required Library.....	38
Figure 4.3: Installing Package for PyCaret Model.....	39
Figure 4.4: Required Library for PyCaret Model	39
Figure 4.5: Reading the Data and Sorting According to Date.....	40
Figure 4.6: Checking for Null Values in Dataset.....	40
Figure 4.7: Checking for Duplicate Rows.....	40
Figure 4.8: Reading the Top 10 Rows of Data to Make Dataframe	41
Figure 4.9: Reading the Last 10 Rows of data to Make the Data frame	41
Figure 4.10: Counting the Number of Rows	41
Figure 4.11: Describing Data set	42
Figure 4.12: DataFrame Info and Data Type.....	42

Figure 4.13: Checking the NaN Values	43
Figure 4.14: Checking Duplicate Data.....	43
Figure 4.15: Counting the Entire Data	43
Figure 4.16: Checking Data Type Only	43
Figure 4.17: Looking for Max Values for Each Row	44
Figure 4.18: Pairplot / Correlation Diagram.....	45
Figure 4.19: Box Plot for Close Values	46
Figure 4.20: Box Plot Low , Hogh , Open, Close Combined.....	47
Figure 4.21: Box Plot for Low , High , Open , Close Combied In Horizontal Orientation	47
Figure 4.22: Box Plot for Low, High, Open, Close Volume, market Capitalization Combined.....	48
Figure 4.23: BoxPlot for Close Colum With Additional Information	49
Figure 4.24: HitMap Correlation	49
Figure 4.25: Plotting Close Values(As Training and Testing Data).....	50
Figure 5.1: Min-Max Scaler for Reshaping the Data.....	51
Figure 5.2: Scaling an Additional filter for Nan Values	52
Figure 5.3: Data Processing , Tuning and Testing.....	52
Figure 5.4: Training and Testing Data	53
Figure 5.5: Model Building with Necessary Layer.....	53
Figure 5.6: Training and Model Fitting.....	54
Figure 5.7: Evaluating the Model	55
Figure 5.8: Plotting Training and Testing Data	55
Figure 5.9: Creating New Dependent Variable	56
Figure 5.10: Splitting the Data into Training and Testing	56
Figure 5.11: Training Data set.....	57
Figure 5.12: Testing Data set.....	57
Figure 5.13: Setup Initialization	58
Figure 5.14: Train All Model	59
Figure 5.15: Production Matrix after Training	60

Figure 5.16: Model evaluation.....	61
Figure 5.17: Plotting Residuals.....	61
Figure 5.18: Plotting Prediction Error.....	62
Figure 5.19: Plotting Cooks Distance	62
Figure 5.20: Learning Curve	63
Figure 5.21: Validation Curve	63
Figure 5.22: Prediction Model.....	64
Figure 5.23: Prediction for LSTM Model	64
Figure 5.24: Predicted Price Visualization along with Close Price by LSTM Model	64
Figure 5.25: prediction For PyCaret Model	65
Figure 5.26: Visualization of Prediction with PyCaret Model	65
Figure 6.1: Result Predicted by PyCaret Model	67
Figure 6.2: Result Predicted by LSTM Model	68

List of Abbreviation

AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
CV	:	Computer Vision
LSTM	:	Long Short-term Memory
M2M	:	Machine to Machine
MI	:	Machine Intelligence
ML	:	Machine Learning
NI	:	Natural Intelligence
NLNN	:	Non-Learning Neural Networks
NLP	:	Natural Language Processing
LNN	:	Learning Neural Networks
RL	:	Reinforcement Learning
NLU	:	Natural Language Understanding
RNN	:	Recurrent Neural Network

Chapter 1

Introduction

1.1 Overview

In 2021, over 7,000 cryptocurrencies were being traded on more than 20,000 online exchanges, with a total market capitalization surpassing USD 300 billion (Intelligence, 2022). Despite not being backed by tangible assets, users trust these cryptocurrencies because they are publicly accessible. With the advancements in technology, the financial marketplace is undergoing a significant transformation. While technology has made things easier, it has also made them more complex. Cryptocurrency, powered by blockchain technology, is one of the biggest and most impactful deployments of this technology. Despite being only 12 years old, its market capitalization is massive. However, the rapid fluctuations in cryptocurrency prices make it difficult to predict the actual cost and make informed investment decisions.

In this project, we aim to forecast the most volatile currency in the cryptocurrency market by evaluating the statistical dependency of AI features using artificial intelligence (AI) and machine learning (ML) modelling. Our goal is to provide stakeholders in the crypto market with a quantitative estimate of AI tools and their corresponding feature categories for financial forecasting. We will perform a data science, analysis of the top-rated cryptos to gain insights and help investors make informed decisions that may lead to meaningful investments in the cryptocurrency market.

We aim to develop a reliable price forecasting model for the most popular cryptocurrencies by utilizing relevant data and employing artificial intelligence (AI) techniques. To achieve this, we build and train the model using historical and current cryptocurrency data obtained through yahoo finance and Kaggle crypto historical data set. So that we have both current data as well as historical data (a comprehensive data set). Both platforms are well-known and trusted financial sites and data driven community. We also create a real-time visual system that displays the latest cryptocurrency financial data.ad at the end make predicted graph as well.

Additionally, we strive to comprehend the functionality of the innovative technology known as Blockchain and its most significant application to date, cryptocurrency. We aim to assess its impact on the financial market and explore its potential benefits and limitations.

Our research project fundamentally focuses on developing a comprehensive understanding of the technological, financial, and interdependent aspects of cryptocurrency. We employ advanced fields of computer science and technologies such as artificial intelligence (AI) and machine learning (ML) to predict future cryptocurrency prices based on historical and current data. Our literature review identifies the most discriminating factors relevant to blockchain technology, cryptocurrency, and financial forecasting. Additionally, help to make understand about machine learning algorithm and it works. Using these factors, we develop an appropriate model to forecast cryptocurrency prices with high accuracy.

Cryptocurrencies are digital tokens that have the potential to replace conventional currencies in the future due to their accessibility. Their widespread adoption is fueled by the ease with which almost anyone can acquire and use them as payment, just like traditional currencies. These tokens are based on blockchain technology, which is decentralized and can have many more applications in creating secure and reliable organizational environments in the future. This technology has the potential to revolutionize how economies and industries operate and significantly reduce inefficiencies and human errors.

1.2 Problem Statement

Cryptocurrencies are becoming more prominent in transforming the financial system because they are gaining popularity among the general public and being accepted by more merchants. Despite the increasing number of people investing in cryptocurrencies, their dynamic features, unpredictability, volatility in nature and predictability variables are largely unknown, which poses a significant risk to crypto stakeholders.

The traditional financial services industry is undergoing a transformation due to the implementation of blockchain and AI technologies. These technologies facilitate trust and streamline multiparty transactions and speed up the transaction process. As blockchain incorporates three major fields of computing viz, Cryptography, peer-to-peer network and computing. Similarly, AI and the financial sector are huge. To know all about blockchain, AI and finance is tedious, and it is very complicated too. All three components combinedly determined the future value. The product of blockchain and tradable in the financial market is called cryptocurrency, and its future price forecasting by using AI/ ML tools is a challenging task.

1.3 Relevance of the Study

Predicting the value of cryptocurrencies is a commonly studied topic within the data science community. The value of stocks and cryptocurrencies is influenced by more than just the number of buyers and sellers. In addition to market forces, changes in government policies towards cryptocurrencies can also impact their value. The popularity of a particular cryptocurrency or the endorsement of it by influential individuals can also affect its value by driving demand and supply. As a result, the prices of cryptocurrencies are subject to change due to a variety of factors.

The world is in the phase of financial transformation due to the inception of digital currency and its value is influenced by the law of demand and supply (buying and selling), but these trends are impacted by various factors. While machine learning can be useful for predicting cryptocurrency prices by considering multiple factors, it is only effective in situations where price changes are influenced by past prices that individuals consider before buying or selling. Not only traditional stock or financial services, crypto is also the result of technology, computation, and distribution network. So, it is very much relevant to study its property and dependable factors and employ AI/ML approach to make reasonable forecasting to help the stakeholders.

1.4 Aims

- The main aim of this project is to make a reasonable price forecasting for the top-rated cryptocurrency price by considering most discriminating factor 's data built and train the model by employing artificial intelligence (AI) and machine learning (ML).
- Built a real time cryptocurrency data-driven visual system by retrieving the crypto financial data via web socket API and historical data.
- Understand the working of new technology called Blockchain and its biggest application till now called cryptocurrency and its influence in financial marketplace.

Evaluate the statistical dependency of AI features considered for crypto financial forecasting employing artificial intelligence (AI) and machine learning (ML) modelling. which provides the crypto stakeholders with quantitative estimate and features categories on AI tools. Perform a data science process to know insides of top rated cryptos so that people get benefited using meaningful investment on it.

1.5 Objectives

- To know all about the new technology is not really that easy, if such technology has multidisciplinary application, then it becomes more tedious. To make deep understanding about blockchain technology and its biggest application called cryptocurrency. Analyse it into financial world approach.
- Financial marketplace is a complicated one and have various dependable factor so understand the financial market and identify its dependable factor via intensive literature review which has already been published in different publication media.
- Analyse the crypto financial data and develop the AI model, examine the accuracy, and deploy the model for forecasting.
- Testing the output of the developed model to the real-world historic data to simulate the financial rise and fall scenario and determine variation.
- Use different machine learning (ML) techniques and find out the future price of crypto currency.
- Report the data processing, data analysis, feature engineering, exploratory data analysis, AI/ML model development, performance monitoring and interpretation.

1.6 Justification

With the advancement of technology, the financial marketplace is being complicated too. Technological development reshapes today's marketplace differently than the past. Cryptocurrency is one of the biggest and impactful deployment of blockchain technology with just the history of 12 years, but market capitalization is huge. Cryptocurrency is based on new technology and very rapid fluctuation in short time makes it hard to predict the actual cost to make a meaningful investment decision. This research project is helpful to make a good understanding form technological prospect, world financial aspect with related dependency and employ one of the advanced technologies in computer science called artificial intelligence (AI) and machine learning (ML) to predict future values and check correctness based on historical data. Find out the most discriminating factor by literatures review which help to make an understanding about blockchain technology, cryptocurrency, and financial forecasting. Consider these discriminating factors and develop as appropriate model to forecast the cryptos price.

1.7 Project Outline

This project aims to address the fundamental research questions that are essential for this project. The answers to these questions serve as inputs for model building and forecasting. By integrating these inputs with the model's output, we can successfully complete this research project. This project revolves around the following fundamental questions.

Fundamental research:

- What is blockchain technology, its background and development to cryptocurrency?
- What are the most discriminating features of cryptocurrency price forecasting?
- How financial market development, crypto adoption current situation and its performance?

To outline this project and report, we start to find the relevant data set through various open-source crypto historical financial data from actively trading site, we specifically focus to Yahoo finance crypto historical data, which is free for public use without any consent. we grab historical data and form Kaggle and also consider recent crypto data from yahoo finance, both are trusted source for data. By deploying data science process, we make a comprehensive data set for highest market share crypto data.

then introduce possible machine learning model such as PyCaret which is regression-based python programming library and Recurrent neural network-based model called long short-term memory (LSTM).

We organize the project report in distinct 6 chapters as follows,

Chapter 1. This chapter consist of the overview of project, aim, objective introduction of project, tentative framework, problem statement, aim, objective project outline and justification.

Chapter 2 In this chapter the literature of Blockchain technology crypto financial data, the most discriminating factor to affecting the crypto price, time series forecasting and prediction model along with the detail description of al the used machine learning algorithm such as regression-based model called PyCaret and Recurrent neural network-based model, long short-term Memory (LSTM) are explained form theoretical and practically implementation aspect. In addition to that financial market, AI /ML used over financial data set and its application has been user explained.

Chapter 3 Research Design and Methodology is all about research design methodology section, where we explained how data set has been generated and employ for the qualitative, time series and qualitative analysis for feature engineering has been deployed. we explained all data science process starting from data collection to final forecasting, qualitative aspect of exploratory data analysis (EDA), time series forecasting, develop Machine Learning Model and use that model to predict the future.

Chapter 4. Exploratory Data Analysis, This chapter is all about visualization and analysis report, qualitative approach with necessary step such as data exploration and visualization it is termed as exploratory data analysis which is helpful to find out the data insights.

Chapter 5.Feature engineering, Training Testing and Model Building is all about feature engineering, training the model, testing the model, and evaluating the model, which is all based on the data set collected and selected ML model. We use regression based PyCaret model algorithm and Recurrent neural network-based model called LSTM. The most important task of the project “prediction” model has been developed in this chapter.

Chapter 6.Result, conclusion and discussion, this chapter is all about result, conclusion, discussion question and future improvement. followed by extracting and comparing the result it is a comparative study model between different ML tools.

Chapter 2

Literature Review

Following the 2008 global financial crisis, an individual or entity named Satoshi Nakamoto created a protocol for a decentralized digital currency system, known as blockchain. This system operates without a central database and relies on volunteer computers around the world to maintain the ledger. The blockchain is publicly accessible and secure, utilizing encryption and public and private keys(cryptography). Transactions can be safely conducted without the need for financial institutions, allowing individuals to transfer funds directly to one another via its own shared lager technology (Weinstein, 2020).

With the rise of digital currency and over 50% of the world's population owning smartphones (Weinstein, 2020), some predict that blockchain technology may replace traditional banking technology, leading to the widespread availability of digital financial products. This potential partnership between the finance industry and new technology may open doors for innovative financial products (Weinstein, 2020).

2.1 Blockchain Technology Background and its Development

Blockchain is the growing list of datasets, that re linked together. It is a shared lager technology allow any user in its network to see the system record. It is the combination of three leading computing technology:

- Cryptography
- Peer-to peer network
- Computing

It is started in 2008, introduction of “**Bitcoin**” in 2009 is considered as blockchain 1.0 and after it gradual development introduction of “Ethereum” in 2015 is considered as blockchain 2.0 Inception of “EOS” in 2018 is considered as blockchain 3.0 and its rapid development and its area of application is growing significantly. The potential of blockchain technology is still uncertain as it is considered a new technology. However, some predictions suggest that by 2022, a pioneering company based on blockchain will have a value of \$10 billion. By 2026, the contribution of blockchain to business value will increase to approximately \$360 billion, and by 2030, it is expected to exceed \$3.1 trillion. (Intelligence, 2022).

Cryptocurrencies are the most advanced development of blockchain, and it is tradable assets and are directly affected by the world economy. The current transition from our conventional monetary system to a new digital, technology-based crypto-economic system is challenging for both the global economy and technology sector. Despite its short history, the impact of cryptocurrency is significant, equivalent to a thousand-year-old history.

2.2 Cryptocurrency

Cryptocurrency is the well-known application of blockchain-based on hash function for cryptographic algorithm. Hash function took any string as input and gives a fixed size output consist of 265 bits (Bitcoin deals), this one of the effective and computable formats. The calculated hash value consists of character like, collision-free, hiding and puzzle free. Basically, it is termed as collision free, but it is not collision free in fact it is very hard to find (almost impossible). Nobody can find the value of X and Y. Were,

$$X \neq Y \text{ \& } H(X) = H(Y)$$

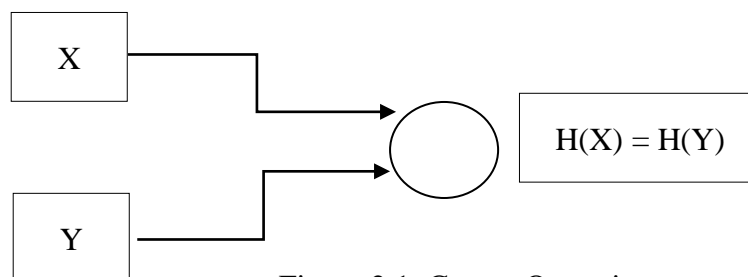


Figure 2.1: Crypto Operation

Collision do exist but very hard to find, very large number of possible inputs for appropriate output (QiangWang, 2020)Hash has mainly applicable for larger data comparison, that means just calculate the hash, the calculated value is only of 256 bits if another calculated hash value of same 256 bits, then we can say each file has same content.

Another important character is hiding property (Jhfree, 2022). It simply can understand in the following way,

Given $H(X)$ it is infeasible find X .

It can make it more clear via coin toss example.

While tossing a standard coin,

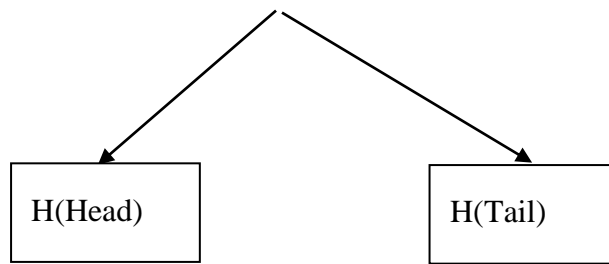


Figure 2.2 Coin Toss Condition

Then can find X.

If a random value “A” is chosen from probability distribution, that has higher min-entropy then,

$$H(A|X),$$

Hash of “A” concatenate with X, it is infeasible to find X.

High mean- entropy distribution is “very spread out” so that no value is chosen with more than negligible probability.

This property has wide application while the user wants to seal a value in envelop and open that envelop later this property help to enhance security by hiding and binding.

Puzzle friendly is another important one. which can interpretate in terms of following.

For every possible output of Y, IF “B” is chosen from a distribution with high mean- entropy then it is infeasible to find X in such that,

$$H(B|X) == Y,$$

This property is highly applicable while given a puzzle id from high mean -entropy distribution and a target set Y.

Try to find X such that,

$$H(id|X) \in Y$$

Puzzle friendly property implies that no solving strategy is much better than trying random value of X. This is helpful in bitcoin mining (Jhfree, 2022).

SHA-256 hash function is used by Bitcoin.

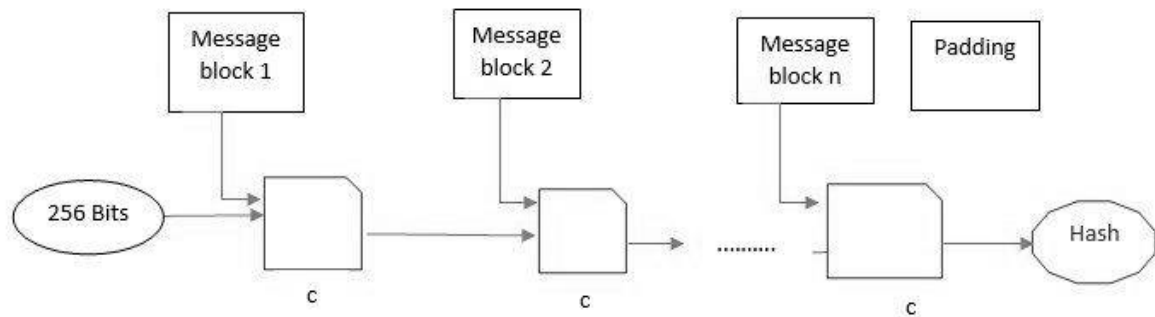


Figure 2.2 Process Flow Chart

Source : Coursera.org

If c is collision free, then

SHA-256 is collision free.

We will employ multidisciplinary aspect of data science process for crypto time series data to make a meaningful prediction to help the stakeholders. All the necessary data set accumulated from multiple source such as yfinance web open historical data and past data set from Kaggle. Before using machine learning technique, we perform exploratory data analysis, feature engineering from the data set. After successful feature engineering process the selected feature are employed to training process. The training process has performed some machine learning process such as regression, gradient boosting, random forest, artificial neural network, long short term memory, sequential neural network will be used to predict the future tick data. The trained model has been tested with the testing data set to make prediction as a result.

For outline the structure of this project report, we start from very scratch as how cryptocurrency work and its property this is basically the cryptocurrency overview its today's status and future potential Bitcoin.

2.3 Understanding Bitcoin

Bitcoin, a digital currency or digital token was established in January 2009 and is presently the most valuable cryptocurrency globally, traded on more than 40 exchanges worldwide, accepting over 30 different currencies (Kepios, 2022). Due to its high volatility, which is considerably greater than conventional currencies.

The bitcoin system comprises of a group of decentralized nodes that execute the bitcoin code and preserve its blockchain. Conceptually, the blockchain can be perceived as a compilation of blocks, where each block comprises of a set of transactions. As all the computers running

the blockchain possess an identical list of blocks and transactions, and can witness new blocks being filled with fresh bitcoin transactions, no one can deceive the system. Bitcoin employs peer-to-peer technology to enable prompt payments. The responsibility of handling transactions on the blockchain rests with miners, who are incentivized by transaction fees (B., 2009).

To comprehend the popularity of bitcoin, it is crucial to understand its mechanism. In contrast to other investments, cryptocurrency is not linked to tangible assets or the US dollar. The primary objective of bitcoin is to facilitate direct exchange of value between two individuals, irrespective of their location. This implies that the network does not have any centralized control, and there is no governmental or central bank authority capable of shutting down or manipulating the value of bitcoin. It will be intriguing to observe the extent to which central banks initiate the digitization of their respective currencies. The trend towards digitization of financial systems is augmenting the acceptance of bitcoin as a more conventional currency. However, the resurgence of the digital currency is also intricately linked to the state of the global financial system. Every transaction in bitcoin adds a unique encrypted signature to the ledger for authentication purposes.

The smooth operation of a business depends heavily on information, which needs to be accurate and delivered quickly. Blockchain technology is ideal for delivering information as it offers a shared, transparent, and immediate information storage system that can only be accessed by authorized network members. A blockchain network can track various aspects of a business, including orders, payments, accounts, and production. Since members share the same view of transactions, all details of a transaction can be seen from start to finish, resulting in greater confidence, increased efficiency, and new opportunities (IBM, 2023).

Cryptocurrencies function as a means of exchange, store of value, and unit of measure, despite their lack of inherent value. Bitcoin is the first and most popular cryptocurrency, serving both as a means of payment and a speculative commodity. Digital assets, or crypto assets, are representations of value enabled by blockchain and cryptography (Weinstein, 2020). Originally created to facilitate value transfer without a trusted intermediary, crypto assets are categorized into three types: cryptocurrencies, crypto commodities, and crypto tokens. One current topic of interest is stablecoins, cryptocurrencies pegged to stable assets such as the US dollar, which may play a vital role in decentralized finance (DeFi).

Despite the rapid fluctuation and technological change, bitcoin faces lots of ups and down in terms of cost. Stakeholders must look into multiple dependable factors to make meaningful decision. The features are now introduced to artificial intelligence and Machine Learning model for forecasting future price, so that stakeholder can have a better view if future in advance.

The fundamental research is now introduced into artificial intelligence /Machine learning algorithm to develop forecasting. The primary research questions of this project are,

- How can the AI/ML algorithm use over most discriminating factor to predict the future price of cryptocurrency?
- How the statistical dependencies help to predict the future price?
- How be the crypto stakeholders provided with quantitative estimate of price for future based on machine learning approach?
- What are the most discriminative, relying on factor of cryptocurrency price forecasting?

Time series forecasting is one of the highly applicable and popular approach in financial data analysis. Not only in trading, but it is also highly applicable on multidisciplinary approaches. The most popular is for future price prediction. Exploratory data analysis (EDA) process as a part of time series forecasting is helpful for getting insights and make it visible to make possible decision (Coinmarketcap. 2022).

2.4 The most Discriminating Features of Cryptocurrency Price Forecasting

Cryptocurrency is managed by cryptographic algorithms, allowing users on a blockchain network to safely own, store, trade, and exchange value. Similar to stock trading, cryptocurrency is also traded, and many firms have emerged in recent years that provide decentralized bitcoin exchanges. The use of blockchain for exchanges enables faster and more cost-effective transactions. Additionally, decentralized exchanges do not require investors to deposit their funds with a centralized authority, allowing them to retain greater control and security. While blockchain-based exchanges primarily trade cryptocurrencies, the concept can also be applied to more traditional assets (Intelligence, 2022).

Forecasting the price of any cryptocurrency is somehow like traditional prediction models, such as stock forecasting. Time series forecasting is the most effective method for cryptocurrency forecasting. The time series data of any cryptocurrency primarily consists of trading date and time, along with corresponding open, close, high, and low values, trade volume, and market capitalization (Wajde Baiod, Janet Light, Aniket Mahanti, 2021).

Forecasting has multiple dependable factors such as the supply of cryptos and market demand, The cost of production through the mining process, competing other cryptos, regulatory /governing developments its sales and use, and The News and media coverage that follows affects investor sentiment, one of the key elements influencing the price of cryptocurrencies (Sharma, 2022).

2.5 Time Series

Time series data is a set of information that is recorded and arranged chronologically, providing a record of changes over time. This data can be utilized to investigate how a specific variable fluctuates over time and make predictions about future patterns. Time series data is used in a variety of fields, including finance, economics, engineering, and social sciences. To extract meaningful insights from time series data, a range of statistical techniques are employed, such as trend and seasonal analysis and forecasting. Examples of time series data include stock prices, weather information, sales figures, and website traffic.

To predict the future of a variable based on its past behaviour, statistical models and techniques are used in time series forecasting. The goal of time series forecasting is to identify patterns and trends in historical data and utilize this information to generate future predictions (Time Series Forecasting in Python?, 2020).

There are variety of time series model has been taken into consideration based on need.

Some of the models are.

- Simple moving average
- ARIMA
- LSTM
- Prophet
- Exponential Smoothing etc..

Performance metrics are employed to evaluate the accuracy of the models used in time series forecasting. The selection of performance metrics is dependent on the specific application and requirements of the predicted variable. In time series forecasting, the following performance metrics are commonly used:

- The mean absolute error (MAE) is a performance metric commonly used in time series forecasting to measure the average absolute difference between the predicted and actual data. The MAE provides an indication of the magnitude of forecast errors.
- The mean squared error (MSE) is another performance metric used in time series forecasting, which calculates the average of the squared differences between the predicted and actual values. It provides a measure of error and indicates the typical magnitude of forecast errors.
- The Root Mean Squared Error (RMSE) is a performance metric frequently used in time series forecasting. It is calculated as the square root of the MSE and provides a measure of the average magnitude of prediction errors.
- The Mean Absolute Percentage Error (MAPE) is another performance metric used in time series forecasting. It calculates the average absolute percentage difference between the predicted and actual values. It is particularly useful when the magnitude of the predicted variable varies throughout the data set.
- Time series data is characterized by the fact that its data points are arranged chronologically with a consistent time interval between them, whether that interval is measured in minutes, hours, days, months, or years. This distinguishes time series data from other types of datasets that do not have a temporal ordering of the data points (Champaneria, 2023).
- The time-dependent structure present in a time series dataset allows for the analysis of patterns, trends, and seasonal fluctuations. In contrast, a normal dataset lacks this structure, and the data points can be seen as independent and uniformly distributed (Champaneria, 2023).

In financial forecasting, time series analysis is used to predict future values of various financial factors such as stock prices, exchange rates, and other financial indicators. This type of analysis is also used in financial risk management to predict the likelihood of financial crises and other events that may impact the financial market.

The world is currently going through a period of transition from the traditional monetary system to a new digital, technology-based crypto-economic landscape. Although it has only been around for a decade, the impact of cryptocurrency and its underlying technology, Blockchain, is akin to that of a thousand years of history. Bitcoin, introduced in 2009, is considered as Blockchain 1.0, and its gradual development and optimization have made it a valuable asset in today's world. By examining the history of cryptocurrencies from 2015 to 2021, we can understand the fluctuations in the crypto market and predict what the future holds for the world's digital economy.

2.6 Financial Market Scenario

The cryptocurrency market is directly linked to stock prices, which are affected by a variety of factors, including investor sentiment, economic conditions, monetary policy, geopolitics, regulatory changes, and developmental changes. Bitcoin and other cryptocurrency prices are influenced by factors such as perceived value, supply, and demand, like other currencies, goods, or services within an economy or nation (Ahmad, 2019).

The prices of cryptocurrencies are expected to rise if their popularity increases, and demand outstrips supply. Conversely, if popularity and demand wane, there will be an excess of supply and prices are likely to fall. In the case of Bitcoin, unless it holds its value for other reasons, its price is expected to decrease in such a scenario (Ahmad, 2019). The supply and demand of Bitcoin are significant factors that impact its price, as cryptocurrencies continue to gain popularity. If the demand for Bitcoin exceeds its supply, the price will rise, but if the popularity declines and demand decreases, the price will decrease unless it retains its value for other reasons. The emergence of new Bitcoin securities and the development of derivatives that are traded by investors and financial organizations also affect the price of Bitcoin. Additionally, the price of Bitcoin is subject to speculation, irrational exuberance, investor panic, and fear, as demand fluctuates based on investor emotions. The value of other cryptocurrencies may also potentially affect the price of Bitcoin, as more cryptocurrencies are being created and embraced by regulators, organizations, and businesses as legitimate means of payment and money (IBM,2023).

Cryptocurrencies are trading assets like stocks, commodities, and securities. Their prices are influenced by demand and supply, which are determined by the number of people interested in purchasing them and how many of them are available. The price is determined by the relationship between these two factors (G-Research Crypto Forecasting, n.d.).

One of the most noteworthy features of Bitcoin is its decentralization, which eliminates the control of traditional financial sectors and monetary authorities using blockchain technology. To forecast cryptocurrency values, the proposed system employs the ANN algorithm and LSTM. We utilized an ANN model with five different memory lengths to predict the price of Bitcoin one day ahead. Both ANN and LSTM are suitable for predicting cryptocurrency price time series since LSTM is designed to examine internal memory flow and its impact on future prediction (ARIMA vs Prophet vs LSTM for Time Series Prediction, 2022).

2.7 Forecast Ability

Cryptocurrencies have distinct characteristics, such as high volatility, nonlinearity, non-stationarity, unpredictable nature, lack of periodicity, noisy data, and nonlinear dynamics, which require special attention when forecasting their price. To predict the price of cryptocurrencies, various statistical, machine learning, and deep learning methods, including linear regression (LR), autoregressive integrated moving average (ARIMA), linear discriminant analysis (LDA), data transmission (DT), random forest (RF), XGBoost, quantitative descriptive analysis (QDA), support vector machine (SVM), and long short-term memory (LSTM) have been utilized by researchers worldwide. Hybrid models such as convolution neural network (CNN) and the new generation of recurrent neural network called GRU have also been employed to forecast the price of bitcoin effectively (Edwards, 2022).

The regression model is also has been classified into multiple variations on the basis of its applicability and area of indulgence, classical model such as auto regressive (AR) model, moving average (MA), auto-regressive moving average (ARMA) and auto regenerative integrated moving average (ARIMA) model. Among all the models it has been concluded that ARIMA model is best fit for financial time series analysis. Now a days python programming language and its different framework gives the flexibility to study and explore some complex data set well (How to Create an ARIMA Model for Time Series Forecasting in Python, 2020). PyCaret is python machine learning library which help to facilitate the data analysis and data science process efficiently (Guang Chen, 2018).

Similarly, hybrid model by combining ARIMA model and Neural network, basically it is a simple variation over ARIMA model only adding one neuron (One layer), which is applicable for both linear and nonlinear data set with enhanced accuracy. The proposed ARIMA -NN model is very useful for forecasting real world financial data, which is applicable for both linear and non-liner data sets (Champaneria, 2023).

Hybrid model by variation linear regression and deep neural network named LR-DBN. This model shows optimized result for predicting cyclical time series data. Every model have their own level of accuracy and dependency factor even though classical machine learning algorithmic method such as linear regression (LR), gradient boosting (GB), random forest (RF) and decision tree (DT) along with different hybrid model able to predict the market. However, this project also highlights feature engineering to improve the accuracy of prediction (Analytics Vidhya, 2018).

2.8 Artificial Intelligence (AI) and Machine Learning (ML)

Artificial intelligence (AI) refers to the process of simulating human intelligence using computers. This technology presents significant opportunities to enhance customer satisfaction, democratize financial services, ensure consumer protection, and greatly improve risk management. The ability of an AI system to be described in human language is reflected in the AI-based prediction models. Machine learning (ML) is a branch of AI that utilizes algorithms to learn from data and make predictions or judgments in the future. Various algorithms are available to assist traders in predicting cryptocurrency price patterns. Forecasting research is at the forefront of this area. An attentive LSTM network and an Embedding Network produce state-of-the-art performance compared to all baselines for the price prediction and fluctuation prediction problem (Bitcoin Price Prediction Using Recurrent Neural Networks and LSTM, 2021). They outperform traditional fully connected deep neural networks by accepting various cryptocurrency data as inputs and managing them individually to extract useful information from each cryptocurrency separately (Jhfree, 2022).

Cryptocurrency is considered to be one of the most challenging and volatile investment options. However, the recent surge in cryptocurrency development has enabled investors to diversify their portfolios, attracting new investors to the market. Additionally, the advancement of information technology has allowed consumers to access a plethora of websites and applications on a variety of portable devices, including smartphones and tablets, in addition to traditional desktop computers. The technique developed enables the forecasting of cryptocurrency prices using machine learning and data mining. To effectively train on bitcoin prices as time series data, LSTM, RNN, Decision tree, ANN, and Linear regression algorithms are used. The technique predicts the movement of a cryptocurrency over various time periods. The predictive accuracy of different algorithms varies, and the time required to build models varies. Despite the limited efforts on cryptocurrency price analysis and

prediction, there have been some research efforts to understand cryptocurrency time series and develop statistical models to recreate and anticipate price dynamics (Baheti, 2022).

2.9 Can AI/ML Models Predict Short-term Movements of the Cryptos Market

Machine learning is a field of artificial intelligence that has a wide range of applications. In the financial market, the deployment of machine learning has been highly successful. There are a variety of algorithms in machine learning that can be used to forecast the future based on historical data. A well-established machine learning model is trained and tested with relevant datasets to determine the correlation between the developed model and the data. The results demonstrate that the trained models significantly outperform random classification (Weinstein, 2020).

The initial step in cryptocurrency price forecasting is to systematically compare the predictive power of different prediction models, such as recurrent neural networks and gradient boosting classifiers, feature sets, including technical and blockchain-based, and prediction horizons. This process helps to establish a comprehensive benchmark for the predictive accuracy of short-term cryptocurrency market prediction models. Based on the analysis, it is clear that recurrent neural networks and gradient-boosting classifiers are suitable for this prediction problem, and technical features remain relevant. (B., 2009).

Another interesting finding is that, as the prediction horizon increases, the prediction accuracy tends to improve, and less recent features appear to be more relevant. Additionally, even though the models are capable of making accurate predictions about the bitcoin market, our results do not contradict the efficient market theory since the quantile-based long-short strategy used produces returns that are not enough to cover transaction costs. (B., 2009).

2.10 Background Reading for Machine Learning Algorithm

2.10.1 Regression

Regression is a statistical technique used to establish the connection between a dependent variable and one or more independent variables. The alteration in the independent variables is linked to the changes observed in the dependent variable. The widely applicable regression model is linear regression.

The linear regression model for financial data analysis has the following five assumptions. However, particularly for crypto currency data set, the assumption is likely to be violated (F Khan, 2020).

- **Linearity:** This assumption simply refers to the expected value of dependent variable is a linear function of independent variable, by considering other variable fixed. During our development process, we enrich features with their nonlinear transformation for the better model fit to linearity assumption.
- **Normality:** This assumption mainly deals with errors from the normal distribution which has unknown mean, standard deviation and variance. The normality property is mainly responsible for calculation of ‘confidence interval’ as the well-known analytical expressions.
- **Independence:** The residuals in the fitted model have no correlation or dependence on each other.
- **Homoscedasticity:** The residuals' variability remains consistent in relation to the dependent variables.
- **Multicollinearity:** In multiple linear regression, it is essential that there is little to no linear correlation between the predictor variables to fulfil this assumption.

Mathematically it can be written as,

$$Y = M * X + C$$

Where, X: Independent variable.

Y: Dependent variable

M: Slope

C: Y intercept

The Least Square Method is a widely used technique to determine the best fit line. It involves computing the regression line by minimizing the sum of squared errors between the data points and the line. Another method for determining this line is known as R Squared analysis.

2.11 Neural Network

Neural network is well suited for stock price prediction and financial analysis. Due to its complexity and its well-suited neuron and its layered approach it makes more approachable

for financial data analysis and make a meaningful decision. The variety of neural network like recurrent neural network is specialized recommendation for financial forecasting and natural language processing. Based on recurrent neural network a special class of recurrent neural network has been developed called Long Shot Term Memory (LSTM) by considering long term and short-term memory management which is best suited for financial forecasting and financial model development.

A recurrent neural network (RNN) is an advanced algorithm that is capable of retaining information from previous inputs in its memory, making it an effective tool for processing large sets of sequential data (A Brief Overview of Recurrent Neural Networks (RNN), 2022). Its primary objective is to process sequential data, which may include time-series data, text sequences, or audio data represented as a sequence of frequencies over time. Unlike other types of neural networks, RNNs can assume that inputs are not independent and are instead dependent on the previous inputs in the sequence.

This algorithm is designed with internal memory that allows it to remember past input, making it highly effective for solving machine learning tasks that involve sequential data. It is considered to be one of the most successful algorithms in the field of deep learning (A Brief Overview of Recurrent Neural Networks (RNN), 2022).

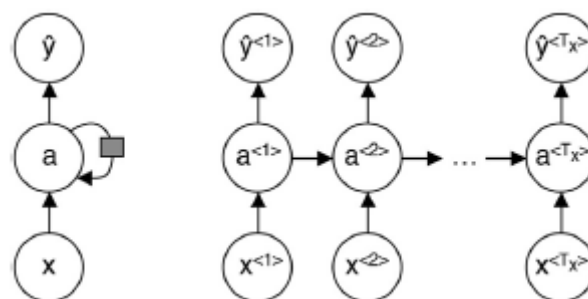


Figure 2.4: Recurrent Neural Network (RNN)

Source: Simplylearn.com

The major problem in recurrent neural network is exploding and vanishing, which can be clear with the following example.

If we have number of sequential data point (can say 50 days of crypto transaction data for financial analysis)

Now,

Input1 * 250 (Huge Number, Explode),

Alternatively, weight of feedback loop was less than 1 and now we have to set 0.5.

Input1 * 0.550 \approx 0 (Vanish)

In summary, Basic vanilla recurrent neural network are hard to train because of the gradient can explode or vanish as clearly explained above.

The above condition of explode and vanishing has avoided by deploying LSTM model (a special variant of recurrent neural network)

2.12 Long Short-Term Memory (LSTM)

Long shot term memory (LSTM) is a type of recurrent neural network which avoids exploding and vanishing by using two separate paths to make a prediction about tomorrow. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that is designed to handle the problem of vanishing gradients that occurs in traditional RNNs. The architecture of an LSTM includes memory cells that can store information for long periods of time, input and output gates that control the flow of information into and out of the cells and forget gates that allow the network to discard information that is no longer relevant.

LSTM networks are useful for processing sequential data, such as text, speech, and time series data, because they can selectively remember and forget information over long periods of time. They have been used in a variety of applications, including language modelling, speech recognition, and machine translation.

The Keras library provides a simple and easy-to-use interface for building and training LSTM networks in Python. With Keras, you can define the architecture of an LSTM network using a few lines of code, and then train it on your dataset using powerful optimization algorithms.

LSTMs were created with the intention of overcoming the issue of long-term dependency in neural networks, and they are naturally adept at retaining information for extended periods, rather than having to learn this ability. Recurrent neural networks share a common design of a chain of repeating modules of neural networks. In the case of traditional RNNs, this repeating module has a simple structure consisting of a single tanh layer (ARIMA vs Prophet vs LSTM for Time Series Prediction, 2022).

The LSTM is a neural network that is composed of a chain structure with memory blocks called cells and four sub-networks. The LSTM unit includes a cell, an input gate, an output

gate, and a forget gate. The flow of information in and out of the cell is regulated by the input, output, and forget gates, and the cell has the ability to retain information over any time period. The LSTM model is particularly suitable for analysing, categorizing, and forecasting time series data that have uncertain durations. These cells store the information, which can manipulate the memory and it is of three main entrances (Analytics, 2018).

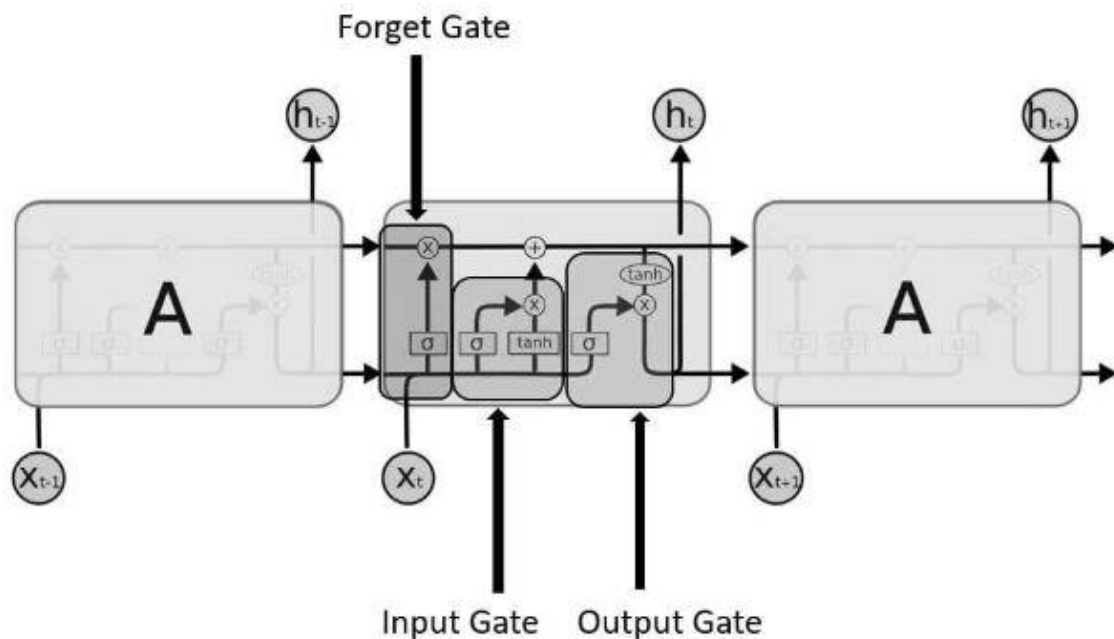


Figure 2.5: Different Gates in LSTM Model

Source: Javatpoint.com

- **Input Gate:** The input gate of a neural network decides which input values should be utilized to modify the memory. It employs the sigmoid function to determine whether to permit values of 0 or 1. Additionally, the gate utilizes the tanh function to assign significance to the input data, thereby establishing their importance on a scale ranging from -1 to 1.
- **Forget Gate:** The forget gate in a neural network is responsible for identifying the information that needs to be eliminated from the memory block. It employs a sigmoid function to make this determination. The gate analyses each value in the cell state (C_{t-1}) and evaluates the previous state (h_{t-1}) and input content (x_t) to produce a score ranging from 0 (discard) to 1 (retain) for each value in the cell state.
- **Output Gate:** The output gate of a neural network block uses the input and stored information to decide what output to produce. This decision is made using a sigmoid

function that either allows values of 0 or 1 to pass through. Additionally, a tanh function determines which values can pass through with a scale of 0 to 1. The tanh function also assigns weights to the input values, which determine their importance on a scale of -1 to 1. These weights are then multiplied by the output of the sigmoid function.

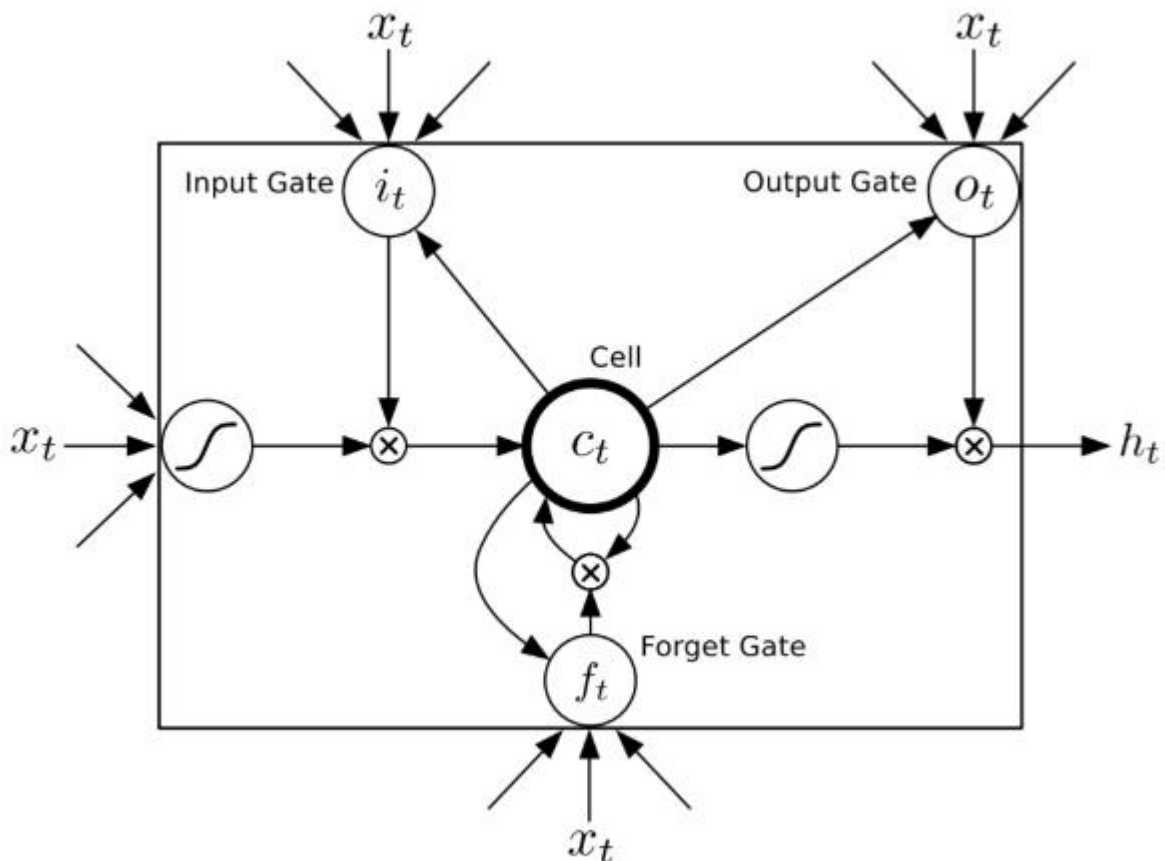


Figure 2.6: Working of Gates in LSTM Model

Source: blog.acolyer.org

2.12.1 LSTM Cycle

LSTM cycles are mainly divided into 4 steps(LSTM for Time Series Prediction, 2022).

- The forget gate is responsible for identifying the information that needs to be discarded from the previous time step.
- The input gate, in combination with the tanh function, is used to obtain new information that is then used to update the cell state.

- The cell state is updated using the information obtained from the input gate and the forget gate.
- The squashing operation and the output gate play a critical role in providing valuable information.

All the above 4 steps can be seen in the following picture.

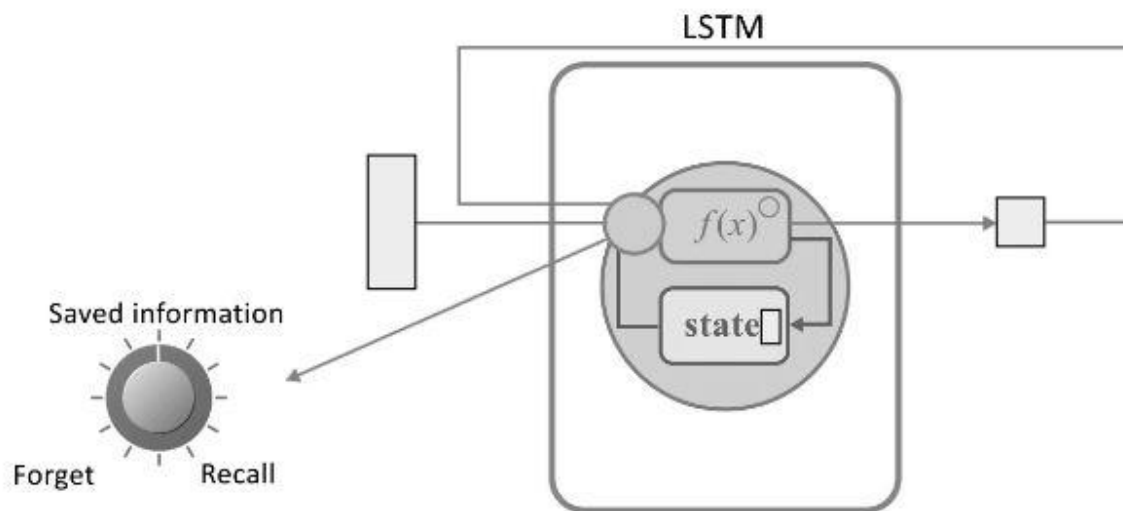


Figure 2.7: LSTM Cycle

Source: Youtube.com

Recurrent neural networks incorporate long short-term memory blocks that enable them to understand how inputs and outputs are related. These blocks employ short-term memory processes to build longer-term memory and hence are referred to as long short-term memory blocks. Such vanishing and exploding conditions have been avoided by considering one path from long-term memory and one path from short-term memory (LSTM for Time Series Prediction, 2022).

LSTMs are equipped with gates that can control the flow of information into and out of the cell state. These gates consist of a sigmoid neural network layer and a pointwise multiplication operation, enabling them to selectively allow information to pass through.

LSTM usage Sigmoid and Tanh activation function.

2.12.2 Sigmoid Activation Function

Sigmoid activation function is nonlinear activation function and always gives the Y co-ordinate of X between 1 & 0.

As depicted below, the output value approaches 1.0 when the input is greater (more positive) in magnitude, and it tends towards 0.0 when the input is smaller (more negative) in magnitude.

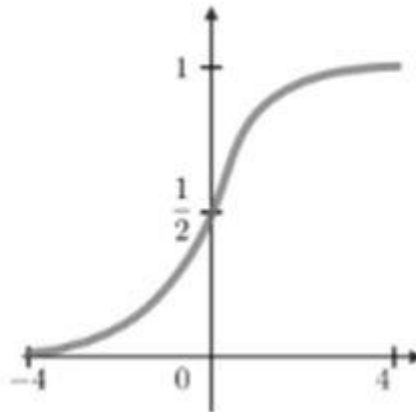


Figure 2.8: Sigmoid Activation Function

From the equation of the sigmoid activation function.

Mathematically it can be written as,

$$f(x) = \frac{1}{1 + e^{-x}}$$

Example,

$$f(10) = 0.99995 \text{ (as y axis)}$$

$$f(-5) = 0.01 \text{ (as of y axis).}$$

The sigmoid function is often used in models that involve predicting probabilities because probabilities only exist between 0 and 1, and the sigmoid function has a range that falls within this range. It has an S-shaped curve which ensures that the function is differentiable and has a smooth gradient, without any sudden changes in its output values.

2.12.3 Tanh Activation Function

Tanh stands for tangent hyperbolic function. Which takes X-axis as input and convert its Y-axis between 1 and -1.

For the Tanh activation function, as the input value increases and becomes more positive, the output value will approach 1.0. Conversely, as the input value decreases and becomes more negative, the output value will approach -1.0.

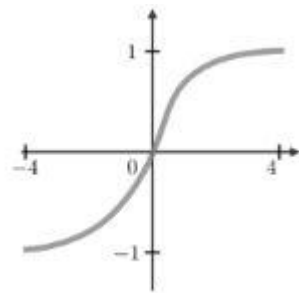


Figure 2.9: Tanh Activation Function

From the equation of Tanh activation function

Mathematically it can be written as,

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

Example,

$$f(2) = 0.96$$

$$f(-5) = -1$$

The Tanh activation function produces output values that are centred around zero, making it easy to categorize them as strongly negative, neutral, or strongly positive. It is commonly used in the hidden layers of neural networks because its output values range from -1 to 1, causing the mean of the hidden layer to be close to zero. This centering of the data makes it easier for the next layer to learn and process the data.

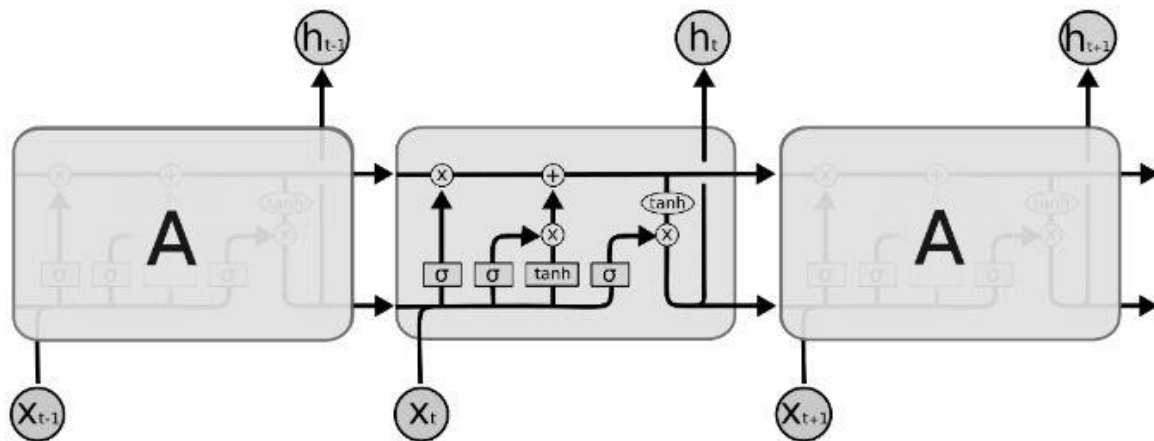


Figure 2.10: LSTM Model

The LSTMs rely heavily on the cell state, which is depicted as a horizontal line at the top of the diagram. The cell state can be compared to a conveyor belt in some respects. As it moves down the chain, there are minimal linear interactions, and data can effortlessly travel down it without undergoing significant modifications. The LSTM is capable of regulating the cell state by introducing or removing information through structures known as gates. Gates serve as a selective mechanism to permit the passage of specific information. They are created by combining a sigmoid neural network layer with a pointwise multiplication operation. The sigmoid layer generates values between zero and one that signify the extent to which each component should be allowed to pass through. A value of zero implies that no component should be allowed to pass through, while a value of one implies that all components should be allowed to pass through. The LSTM comprises three gates that function to regulate and safeguard the cell state (LSTM for Time Series Prediction, 2022).

2.13 Bidirectional LSTM

In Bidirectional Recurrent Neural Networks (BRNN), each training sequence is presented in both forward and backward directions to two separate recurrent neural networks that are connected to the same output layer. As a result, the BRNN can obtain complete sequential knowledge about all points that come before and after each point in a sequence. The use of a BRNN eliminates the need to determine a task-specific time window or target delay size since the network has the freedom to utilize as much or as little context as needed.

Standard RNNs are limited in their ability to utilize only the past contexts. In contrast, Bidirectional RNNs (BRNNs) overcome this limitation by processing data in both directions through two hidden layers that feed into the same output layer. By combining BRNNs with

LSTMs, a bidirectional LSTM is created, which can access long-range context in both input directions.

2.14 PyCaret Model

PyCaret is a Python-based machine learning library that utilizes low-code methodology and is available for free as an open-source tool. It streamlines machine learning workflows by automating various tasks and supports end-to-end machine learning and model management. This library accelerates the experimentation cycle and enhances productivity by providing a complete solution for machine learning tasks (Welcome to PyCaret - PyCaret Official, 2020). Comparatively PyCaret is a significantly low line of code used. PyCaret is a machine learning library in Python that facilitates fast and effective end-to-end experiments for data scientists. Unlike other open-source machine learning libraries, PyCaret is a low-code alternative that allows for the execution of intricate machine learning tasks using minimal lines of code. The library is straightforward and user-friendly, enabling a simplified user experience (Welcome to PyCaret - PyCaret Official, 2020). PyCaret is a deployment-ready Python library, which implies that all the procedures carried out in a machine learning experiment can be replicated using a reproducible pipeline that guarantees production readiness. This means that PyCaret regression user to create a deployable model that is production-ready, replicable, and can be utilized with ease. PyCaret can be use as regression , classification , timeseries , clustering , anomaly detection ,oop and many more (Welcome to PyCaret - PyCaret Official, 2020).

PyCaret was developed with the goal of simplifying the data analysis process for citizen data scientists. This term was coined by Gartner and refers to individuals who possess the ability to perform basic to moderately complex analytical tasks without extensive prior experience. Skilled data scientists are often in high demand and can be costly to employ, but citizen data scientists can help to bridge this gap and solve data science challenges in a business context. The design of PyCaret aims to make data analysis more accessible to a wider range of individuals, regardless of their technical background (Welcome to PyCaret - PyCaret Official, 2020).

PyCaret is a machine learning library that offers a more accessible and user-friendly alternative to the established scikit-learn library. It is particularly advantageous for individuals with limited experience in machine learning. This tutorial provides an introduction to the main features of PyCaret and includes a case study on regression. It is recommended that users install the latest version of Anaconda on Windows, macOS, or Linux

to follow along with the tutorial, but it is also compatible with Google Colab. The code can be executed in a Jupyter Notebook or any preferred IDE.

2.14.1 PyCaret Regression Model

Regression is a fundamental type of supervised machine learning task that involves determining the connection between an independent variable, referred to as the feature, and a dependent variable, which is called the target.

Regression is a machine learning technique that can be utilized for predicting continuous values, such as the value of a house, as opposed to classification, which is employed for discrete values identified as classes. The PyCaret regression module leverages the functionality of sklearn and enables users to design and evaluate regression models using just a few lines of code. The module encompasses numerous algorithms and features, including the ability to plot and conduct hyperparameter tuning (Welcome to PyCaret - PyCaret Official, 2020).

Chapter 3

Research Design and Methodology

We will employ a qualitative intensive literature review method to identify the most relevant and discriminating features of cryptocurrencies. Since cryptocurrencies are the result of both technology and tradable assets, there are many factors that can affect their value. We will conduct a thorough review of the literature, including articles, journals, publications, and books related to financial knowledge, to identify the most important features and their impact on cryptocurrency prices.

After identifying the most relevant discriminating features we will employ complete data science process to predict the future price of the cryptocurrency. The complete data science process is seen in **Figure 3.1**.

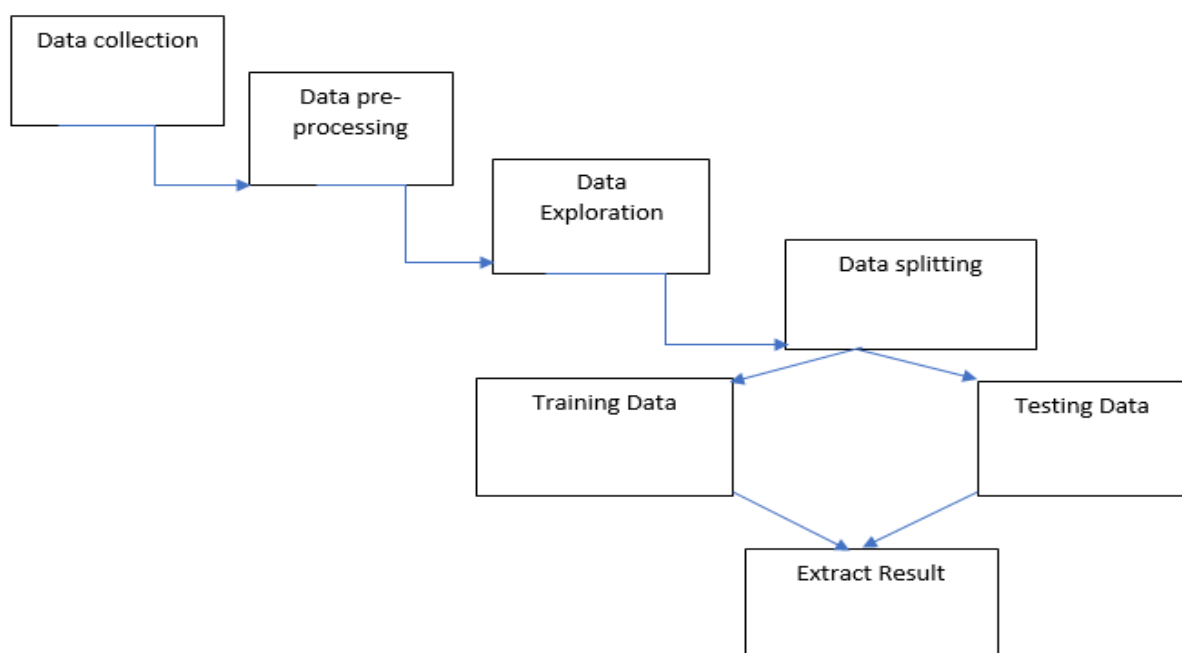


Figure 3.1 Methodology and Model Selection.

3.1 Data Preprocessing

The initial step in the data science process is acquiring raw data from a trusted source, for this specific project we will rely on Yahoo finance crypto data and the Kaggle crypto historical data set. The next step involves cleaning the data, which typically involves handling errors, incomplete data, and redundant data. In some cases, the data may contain decimal numbers

that are too large to be effectively calculated, so it may be necessary to apply a rounding function such as floor or ceiling to manage these values.

The next step in data pre-processing is classification. The collected data is already somewhat classified, with each coin having its own respective CSV file. However, there may be redundant rows of similar data that need to be reduced in order to streamline the data for analysis. This is considered a form of data transformation.

3.2 Data / Data Set

The data set is time series data. A time series is a set of data points that are regularly measured at uniform intervals. This implies that specific values are captured at consistent time intervals, our working data set had been retrieved every day at 11:59:59 PM starting date of 4/29/2013. The openness, legality, and ethics of the dataset are important. The dataset is somewhat clean, although there may be missing values. The data has already come from a trusted source (Yahoo & Kaggle) with a large community so that trustworthy ness issue can be avoided. The data formatted has already been formatted first level for public use specially for Yahoo Finance and Kaggle. These datasets are widely used to know more about cryptocurrency in multidisciplinary aspects, and its timely updated current and historical data.

The data set consists of five alphanumeric calculative values.

- Highest value
- Lowest value
- Opening value
- Closing value
- Volume of transaction

3.3 EDA (Exploratory Data Analysis) - Qualitative

The initial stage involves collecting relevant data, followed by implementing data gathering, pre-processing, and model building using the Python programming language and libraries such as TensorFlow, scikit-learn, XGBoost, NumPy, and Pandas.

The major activities include,

- Cryptocurrency data overview
- Time Series

- Data pre-processing
- Build and train model by using ML tools.
- Use the model to predict future cryptos price.

The complete EDA process is seen in **Figure 3.2**.

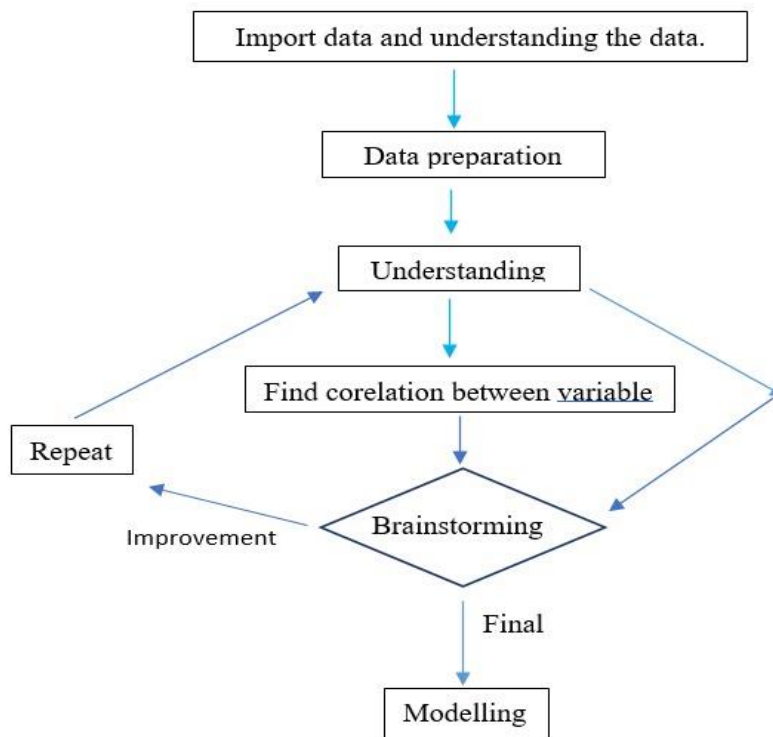


Figure 3.2: EDA Process

3.4 ML (Machine Learning)-Qualitative

After collecting and pre-processing the data, the next step is to train different models for predicting the prices of different cryptocurrencies based on their historical prices. The major steps are as follows.

- Data exploration and visualization.
- Training the models.
- Testing the models.
- Extracting and comparing the result

3.5 Core Solution

In order to develop a machine learning-based prediction model, several pre-processing steps must be performed. These include importing the necessary libraries and loading them, as well as importing the respective CSV files to analyse the data frames. Exploratory data analysis should be performed to better understand the relationships between variables. The data set must be split into training and testing sets to train and evaluate the model. Different machine learning algorithms, such as regression, recurrent neural network (RNN), LSTM, can then be applied to the data to create the prediction model.

In order to model financial data, time-series methods are typically more appropriate. For this purpose, various time-series methods such as the Generalized Autoregressive Conditional Heteroskedastic Model (GARCH), Exponential Weighted Averages (EWMA), and Structural Time Series models were used to forecast the future of coins. However, for this project, the time series data has been executed in regression based PyCaret model and Recurrent neural network-based LSTM model. The accuracy of the predictions was determined using uncertainty quantification methods such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

The core solution of this project is figure out in **Figure 3.3**

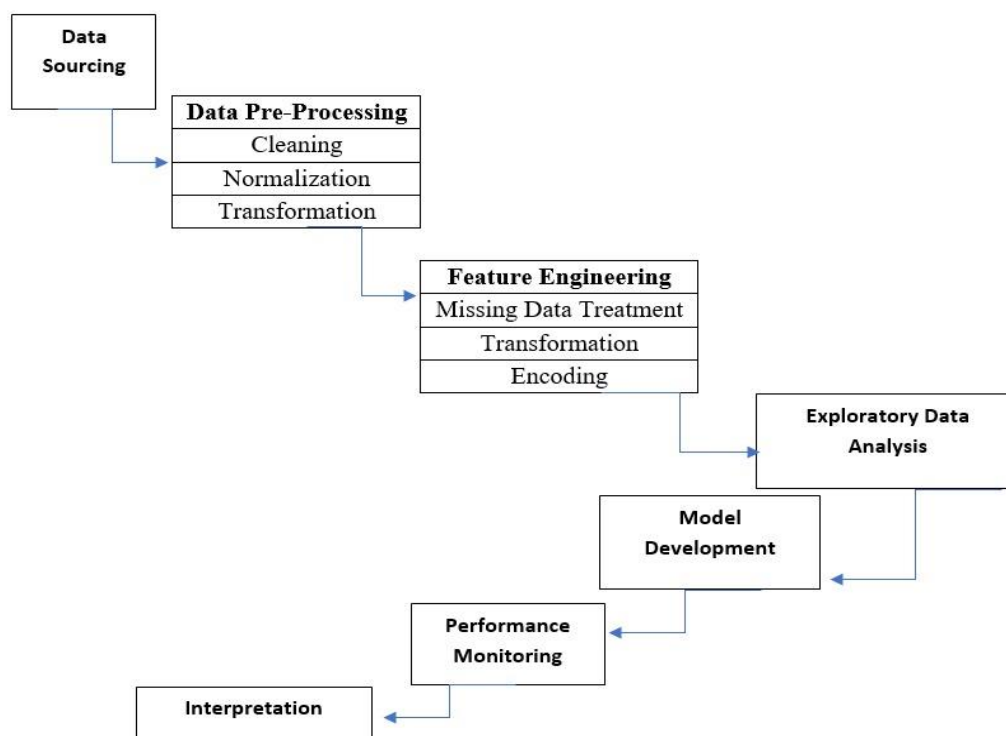


Figure:3.3: Core Solution for Crypto Price Forecasting

3.6 Forecasting Process

The forecasting process (make future price prediction) based on the Regression model called PyCaret model and recurrent neural network-based model called LSTM model is shown in **Figure 3.4.** along with testing data set.

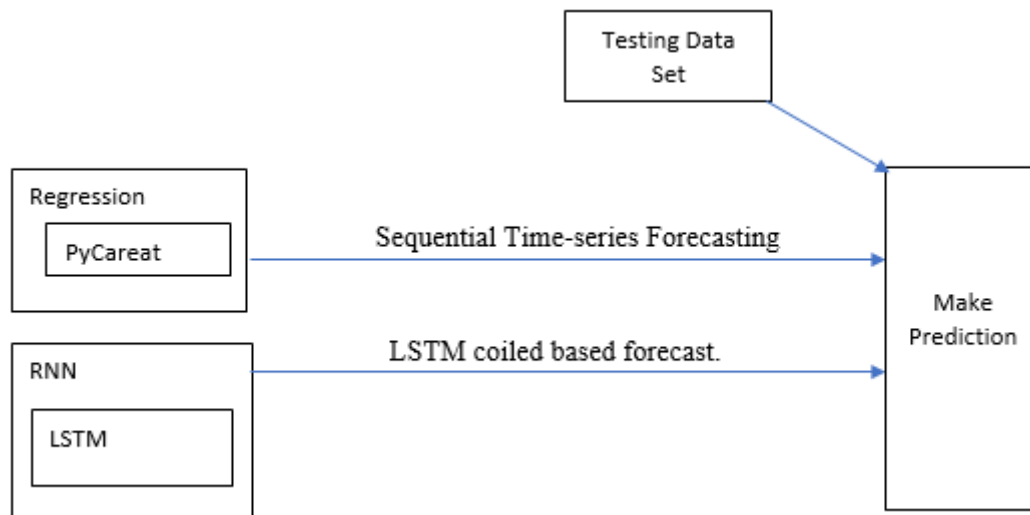


Figure 3.4: Forecasting Process

3.7 Feature Exploration

The objective here is to identify traits that have a high connection to Bitcoin, but ideally live outside the Bitcoin ecosystem. An example of this would be cryptocurrency universe market capitalization, of which the Bitcoin market capitalization constitutes roughly 35% according to **CoinMarketCap.com** at the time of this writing (Coinmarketcap, 2022). Features evaluated include:

- Cryptocurrency market capitalization
- Current price
- Volume
- Number of Transaction
- Average Block size
- Transaction Fees

- Unique Address
- Hash Rate

3.8 Big Data Storage and Computation

The collected data and compiled data set has been stored in local machine. As it needs enormous computational power so we use google co-laboratory platform so that the computation can be made fast. To reduce data complexity and dimension of the model, we eliminate unnecessary columns so that ETL (extraction, transformation, loading) becomes efficient. To prevent over-fitting, we evaluate combinations of the features by using training and validation sets.

Chapter 4

Exploratory Data Analysis

4.1 Data Set

The raw data set has been retrieved via yahoo finance web API, only for cryptos data. The data set has also been supported by Kaggle with its historical data set. The Yahoo finance data set link is as follows.

<https://uk.finance.yahoo.com/quote/BTC-GBP/history?period1=1410912000&period2=1680393600&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true>

The real-time data from a trusted financial institution and the data range covered by the data set is 29 April 2013(Inclusive) to 6 July 2021(inclusive). The training data set and testing data set has developed as per need from the same set.

The dataset is somewhat clean, although there may be missing values. The data has already come from a trusted source with a large community, so it has been gathered via multiple sources and formatted for public use especially for Yahoo Finance and Kaggle. These datasets are widely used to know more about cryptocurrency in multidisciplinary aspects, and its timely updated historical data.

The data set consists of five alphanumeric calculative values.

- Highest value
- Lowest value
- Opening value
- Closing value
- Volume of transaction

We merge the Kaggle historical data set and Yahoo Finance data to make a comprehensive data set. The merged data set has been used for further analysis. we explore some time series, and statistical approach for the data. **Figure 4.1** shows sample data.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Volume	
2	9/17/2014	286.456	287.523	278.037	281.177	1.3E+07	
3	9/18/2014	280.885	281.075	252.262	258.427	2.1E+07	
4	9/19/2014	258.221	259.373	235.116	242.429	2.3E+07	
5	9/20/2014	242.354	259.93	239.412	251.092	2.3E+07	
6	9/21/2014	250.589	253.255	241.438	244.57	1.6E+07	
7	9/22/2014	244.742	249.061	243.086	245.679	1.5E+07	
8	9/23/2014	245.642	269.472	241.701	265.872	2.8E+07	
9	9/24/2014	265.848	265.922	256.819	259.269	1.9E+07	
10	9/25/2014	259.239	259.367	251.007	252.036	1.6E+07	
11	9/26/2014	251.947	254.173	245.299	248.908	1.3E+07	
12	9/27/2014	248.373	250.26	244.567	245.889	9249938	
13	9/28/2014	245.859	246.81	230.514	232.283	1.5E+07	
14	9/29/2014	232.127	236.805	229.198	231.213	2E+07	
15	9/30/2014	231.596	241.165	229.924	238.692	2.1E+07	
16	10/1/2014	238.99	241.786	234.919	236.99	1.6E+07	
17	10/2/2014	237.22	237.565	230.941	232.156	1.3E+07	
18	10/3/2014	232.224	235.046	224.18	225.216	1.9E+07	

Figure 4.1: Data Set for Bitcoin

4.2 Data Preprocessing

Data pre-processing often involves dealing with issues like errors, missing information, and duplicated data. If the data contains decimal numbers that are too big to work with, a rounding function such as floor or ceiling can be used to manage these values.

It's been said that our minds are best suited for visualization rather than working with theoretical or descriptive data. During the process of Data Pre-processing, we import necessary libraries, load CSV files for data analysis, handle missing data, visualize the data, identify relationships between variables, and split it into training and testing data for model development. The act of visually representing information and data is referred to as data visualization. By using visual components like charts, graphs, and maps, data visualization tools enable us to easily examine and understand trends, patterns, and outliers within data. In a Big Data environment, data visualization tools and technologies are crucial for analysing large volumes of data and making data-driven decisions. Data pre-processing and cleaning mainly include the following work.

- Load dataset into data frame
- Explore insights (row, column, range)
- Handel missing values and data

4.3 Installing and Importing Necessary Dependencies

Exploratory data analysis starts with importing the necessary Python's library. The base library are NumPy, Panda, TensorFlow, OS, Matplotlib, Seaborn. These libraries have their particular task. The baseline library import is seen in the following figure and has been explained in the respective task below.



```
import os
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras

import seaborn as sns
from pylab import rcParams
import matplotlib.pyplot as plt
from matplotlib import rc
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.layers import Bidirectional, Dropout, Activation, Dense, LSTM
from tensorflow.python.keras.layers import CuDNNLSTM
from tensorflow.keras.models import Sequential

%matplotlib inline
```

Figure 4.2: Importing Required Library.

The OS module in Python is used for creating and managing directory structures, fetching the content, and identifying the current directory.

NumPy is a library in Python for large multi-dimensional arrays and matrices, as well as for mathematical functions.

Pandas is a library in Python for manipulating and analysing data, mainly for numerical calculations.

Matplotlib is a plotting library in Python for visualization.

Seaborn is a data visualization library based on matplotlib.

Scikit-learn is an ML library, especially for classification, regression, and clustering.

TensorFlow is an ML library for the training and inference of neural networks.

Keras is a Python library for AI/ML activities and inference for TensorFlow.

Similarly, package installation is similar approach for installing necessary package like PyCaret,

```
[ ] pip install pycaret # package pycaret installation
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pycaret in /usr/local/lib/python3.8/dist-packages (2.3.10)
Requirement already satisfied: kmodes>=0.10.1 in /usr/local/lib/python3.8/dist-packages (from pycaret) (0.12.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (from pycaret) (3.2.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages (from pycaret) (1.2.0)
Requirement already satisfied: yellowbrick>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from pycaret) (1.3.post1)
Requirement already satisfied: scipy<=1.5.4 in /usr/local/lib/python3.8/dist-packages (from pycaret) (1.5.4)
Requirement already satisfied: imbalanced-learn==0.7.0 in /usr/local/lib/python3.8/dist-packages (from pycaret) (0.7.0)
Requirement already satisfied: textblob in /usr/local/lib/python3.8/dist-packages (from pycaret) (0.15.3)
Requirement already satisfied: mlflow in /usr/local/lib/python3.8/dist-packages (from pycaret) (2.0.1)
Requirement already satisfied: pyod in /usr/local/lib/python3.8/dist-packages (from pycaret) (1.0.6)
Requirement already satisfied: pyyaml<6.0.0 in /usr/local/lib/python3.8/dist-packages (from pycaret) (5.4.1)
Requirement already satisfied: scikit-learn==0.23.2 in /usr/local/lib/python3.8/dist-packages (from pycaret) (0.23.2)
Requirement already satisfied: cycler>=2.4.0 in /usr/local/lib/python3.8/dist-packages (from pycaret) (2.3.0)
```

```
[ ] pip install markupsafe==2.0.1
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: markupsafe==2.0.1 in /usr/local/lib/python3.8/dist-packages (2.0.1)
```

```
[ ] pip install jinja2
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: jinja2 in /usr/local/lib/python3.8/dist-packages (2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.8/dist-packages (from jinja2) (2.0.1)
```

Figure 4.3: Installing Package for PyCaret Model

To access the functionality of PyCaret, you need to install the PyCaret package along with Markupsafe and Jinja2. Pycaret is a low-code ML library for EDA, pre-processing, modelling, training, and MLOPS. Markupsafe is for dealing with text and special characters to wrap in markup. Jinja2 is a fast-templating engine. All other common model as prerequisite has already been installed.

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
import jinja2
from pycaret.regression import *

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Figure 4.4: Required library for PyCaret Model.

4.4 Data Cleaning

The data cleaning process is the way of making data ready for analysis through visualization. The very first step is making Data Frame by using pandas. Data Frame is a 2-dimensional labelled data structure. Its primary activities are data sorting, removing unwanted columns, avoiding null values and make the data complete as possible. We clean the data in the following way with the help of Python programming and R programming tools.

```
[ ] df = pd.read_csv('coin_Bitcoin.csv', parse_dates=['Date'])
```

```
[ ] df = df.sort_values('Date')
```

Figure 4.5: Reading the Data and Sorting According to Date.

After this process, the data frame has been sorted according to date (ascending in structure).

```
[ ] coinbit.isnull().sum()# checking null values
```

```
SNo      0
Name      0
Symbol    0
Date      0
High      0
Low       0
Open      0
Close     0
Volume    0
Marketcap 0
dtype: int64
```

Figure 4.6: Checking for null values in the dataset.

The above figure clearly signifies that there are not any null values in the working data set. It signifies that the working data set is free from null values, which is the preliminary stage of data pre-processing.

```
[ ] df.duplicated().sum()
```

```
0
```

Figure 4.7: Checking for Duplicate Rows.

We check the data duplicate; the above expression clearly shows that there is not any duplicate data

that exist in the data set. From all the above activities, we can ensure that the data set is clean. Now the data set is ready for exploratory data analysis.

4.5 EDA Process

4.5.1 Statistical Analysis

df.head(10)

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	1	Bitcoin	BTC	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.0	1.603769e+09
1	2	Bitcoin	BTC	2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.0	1.542813e+09
2	3	Bitcoin	BTC	2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.0	1.298955e+09
3	4	Bitcoin	BTC	2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.0	1.168517e+09
4	5	Bitcoin	BTC	2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.0	1.085995e+09
5	6	Bitcoin	BTC	2013-05-04 23:59:59	115.000000	92.500000	98.099998	112.500000	0.0	1.250317e+09
6	7	Bitcoin	BTC	2013-05-05 23:59:59	118.800003	107.142998	112.900002	115.910004	0.0	1.288693e+09
7	8	Bitcoin	BTC	2013-05-06 23:59:59	124.663002	106.639999	115.980003	112.300003	0.0	1.249023e+09
8	9	Bitcoin	BTC	2013-05-07 23:59:59	113.444000	97.699997	112.250000	111.500000	0.0	1.240594e+09
9	10	Bitcoin	BTC	2013-05-08 23:59:59	115.779999	109.599998	109.599998	113.566002	0.0	1.264049e+09

Figure 4.8: Reading the top 10 rows of data to make data frame.

df.tail(10)

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
2981	2982	Bitcoin	BTC	2021-06-27 23:59:59	34656.127356	32071.757148	32287.523211	34649.644588	3.551164e+10	6.494617e+11
2982	2983	Bitcoin	BTC	2021-06-28 23:59:59	35219.891791	33902.075892	34679.122222	34434.335314	3.389252e+10	6.454428e+11
2983	2984	Bitcoin	BTC	2021-06-29 23:59:59	36542.111018	34252.484892	34475.559697	35867.777735	3.790146e+10	6.723334e+11
2984	2985	Bitcoin	BTC	2021-06-30 23:59:59	36074.759757	34086.151878	35908.388054	35040.837249	3.405904e+10	6.568525e+11
2985	2986	Bitcoin	BTC	2021-07-01 23:59:59	35035.982712	32883.781226	35035.982712	33572.117653	3.783896e+10	6.293393e+11
2986	2987	Bitcoin	BTC	2021-07-02 23:59:59	33939.588699	32770.680780	33549.600177	33897.048590	3.872897e+10	6.354508e+11
2987	2988	Bitcoin	BTC	2021-07-03 23:59:59	34909.259899	33402.696536	33854.421362	34668.548402	2.438396e+10	6.499397e+11
2988	2989	Bitcoin	BTC	2021-07-04 23:59:59	35937.567147	34396.477458	34665.564866	35287.779766	2.492431e+10	6.615748e+11
2989	2990	Bitcoin	BTC	2021-07-05 23:59:59	35284.344430	33213.661034	35284.344430	33746.002456	2.672155e+10	6.326962e+11

Figure 4.9: Reading the last 10 rows of data to make data frame.

Statistical analysis always starts with looking into the data. **Figure 4.8** and **Figure 4.9** (above two tables) clearly show that the data set is clear and fair to proceed with the data analysis process.

```
[ ] df.shape
```

```
(2991, 10)
```

Figure 4.10: Counting the number of rows.

The shape function gives us the number of rows and columns of data that exist in the data set. Our data set consists of 2991 rows and 10 columns. NOTE: This data set only consists of Bitcoin data set.

```
[ ] df.describe
```

```
<bound method NDFrame.describe of
0      1  Bitcoin  BTC 2013-04-29 23:59:59  147.488007  134.000000
1      2  Bitcoin  BTC 2013-04-30 23:59:59  146.929993  134.050003
2      3  Bitcoin  BTC 2013-05-01 23:59:59  139.889999  107.720001
3      4  Bitcoin  BTC 2013-05-02 23:59:59  125.599998  92.281898
4      5  Bitcoin  BTC 2013-05-03 23:59:59  108.127998  79.099998
...    ...    ...    ...    ...    ...    ...
2986  2987 Bitcoin  BTC 2021-07-02 23:59:59  33939.588699  32770.680780
2987  2988 Bitcoin  BTC 2021-07-03 23:59:59  34909.259899  33402.696536
2988  2989 Bitcoin  BTC 2021-07-04 23:59:59  35937.567147  34396.477458
2989  2990 Bitcoin  BTC 2021-07-05 23:59:59  35284.344430  33213.661034
2990  2991 Bitcoin  BTC 2021-07-06 23:59:59  35038.536363  33599.916169

      Open      Close      Volume      Marketcap
0      134.444000  144.539993  0.000000e+00  1.603769e+09
1      144.000000  139.000000  0.000000e+00  1.542813e+09
2      139.000000  116.989998  0.000000e+00  1.298955e+09
3      116.379997  105.209999  0.000000e+00  1.168517e+09
4      106.250000   97.750000  0.000000e+00  1.085995e+09
...    ...    ...    ...    ...
2986  33549.600177  33897.048590  3.872897e+10  6.354508e+11
```

✓ 1s completed at 11:01 PM

Figure 4.11: Describing the data set.

```
[ ] df.info()
```


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2991 entries, 0 to 2990
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   SNo          2991 non-null   int64
1   Name         2991 non-null   object
2   Symbol       2991 non-null   object
3   Date         2991 non-null   datetime64[ns]
4   High         2991 non-null   float64
5   Low          2991 non-null   float64
6   Open         2991 non-null   float64
7   Close        2991 non-null   float64
8   Volume       2991 non-null   float64
9   Marketcap    2991 non-null   float64
dtypes: datetime64[ns](1), float64(6), int64(1), object(2)
memory usage: 257.0+ KB
```

```
[ ] df.dtypes
```

```
SNo          int64
Name         object
Symbol       object
Date         datetime64[ns]
High         float64
Low          float64
Open         float64
Close        float64
Volume       float64
Marketcap    float64
dtype: object
```

Figure 4.12: Data frame info and data type.

The data frame consists of no null values in every cell. and the data frame consists of object, datetime64 and float64 data types. The high value, low value, open value, close value, volume and market cap are float64 data type. which is clear from **Figure 4.12**.

 `df.isna()`

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...
2986	False	False	False	False	False	False	False	False	False	False
2987	False	False	False	False	False	False	False	False	False	False
2988	False	False	False	False	False	False	False	False	False	False
2989	False	False	False	False	False	False	False	False	False	False
2990	False	False	False	False	False	False	False	False	False	False

✓ 1s completed at 11:01 PM

Figure 4.13: Checking for Nan values.

.”isna()” function is responsible for checking the existence of Nan values in the data frame .**Figure 4.13** can see in the output table there is no any Nan values.

```
[ ] df.duplicated().sum()

0
```

Figure 4.14: Checking for duplicate data.

It signifies that the data frame contains no duplicate data. All data are distinct.

df.value_counts()

SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap	
1	Bitcoin	BTC	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.000000e+00	1.603769e+09	1
1998	Bitcoin	BTC	2018-10-17 23:59:59	6601.210000	6517.450000	6590.520000	6544.430000	4.088420e+09	1.133993e+11	1
1989	Bitcoin	BTC	2018-10-08 23:59:59	6675.060000	6576.040000	6600.190000	6652.230000	3.979460e+09	1.151629e+11	1
1990	Bitcoin	BTC	2018-10-09 23:59:59	6661.410000	6606.940000	6653.080000	6642.640000	3.580810e+09	1.150078e+11	1
1991	Bitcoin	BTC	2018-10-10 23:59:59	6640.290000	6538.960000	6640.290000	6585.530000	3.787650e+09	1.140308e+11	1
...										
1000	Bitcoin	BTC	2016-01-23 23:59:59	394.542999	381.980988	382.433990	387.490997	5.624740e+07	5.858060e+09	1
1001	Bitcoin	BTC	2016-01-24 23:59:59	405.484985	387.510010	388.101990	402.971008	5.482480e+07	6.093788e+09	1
1002	Bitcoin	BTC	2016-01-25 23:59:59	402.316986	388.553986	402.316986	391.726013	5.906240e+07	5.925345e+09	1
1003	Bitcoin	BTC	2016-01-26 23:59:59	397.765991	390.575012	392.002014	392.153015	5.814700e+07	5.933373e+09	1
2991	Bitcoin	BTC	2021-07-06 23:59:59	35038.536363	33599.916169	33723.509655	34235.193451	2.650126e+10	6.418992e+11	1

Length: 2991, dtype: int64

Figure 4.15: Counting the entire data.

```
df.max()

SNo                2991
Name              Bitcoin
Symbol            BTC
Date      2021-07-06 23:59:59
High      64863.098908
Low       62208.964366
Open      63523.754869
Close     63503.45793
Volume    350967941479.059998
Marketcap 1186364044140.27002
dtype: object
```

Figure 4.17: Looking for Max values for each row.

“max ()” gives the maximum value in the data frame as output. The above table, **Figure 4.17** clearly depicts the max value for each column. As the data frame consists of 2991 columns which is printed from SNo column. Similarly latest date has been printed from date column. From high value column 64863.098908 has been printed. Similarly highest value of the respective column has been printed as output form the data frame.

4.5.2 Visualization Analysis

Data visualization is an essential aspect of cryptocurrency applications, enabling stakeholders to monitor real-time indicators. This is unlike traditional market reports and news outlets. The use of visualization makes it easier for customers to conduct both fundamental and technical analysis, leading to the development of successful trading strategies and profitability.



```
sns.pairplot(df)
```

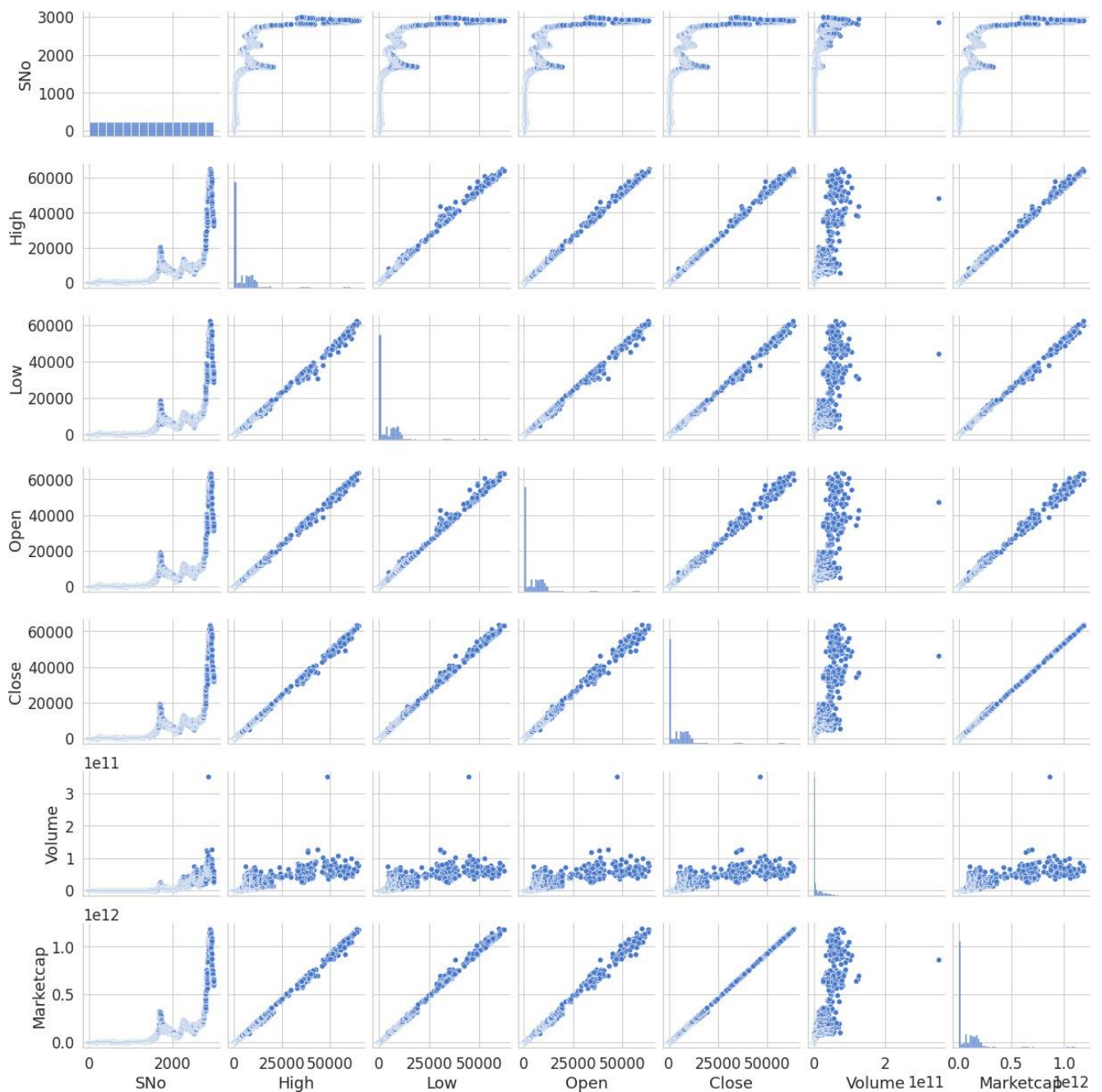


Figure 4.18: Pair plot /Correlation

A pair plot is a type of visualization tool that helps identify relationships between different variables in a given dataset. Basically, pair plot visualizes one column of data against another. It can be used for both continuous and categorical data. We want to examine the correlation between the closing

values of Bitcoin currencies by using pair plots for visualization. Initially, it is observed positive linear relationships for certain features such as Open and Close. Form the above pair plot **Figure 4.18** it is clear that most of the parameter has linear correlation among each other, though the correlation matrix will calculate the numeric value, but it is clearly visible that the pair plot also supports the linearity character to each other. The correlation coefficient will be calculated later in the correlation matrix.

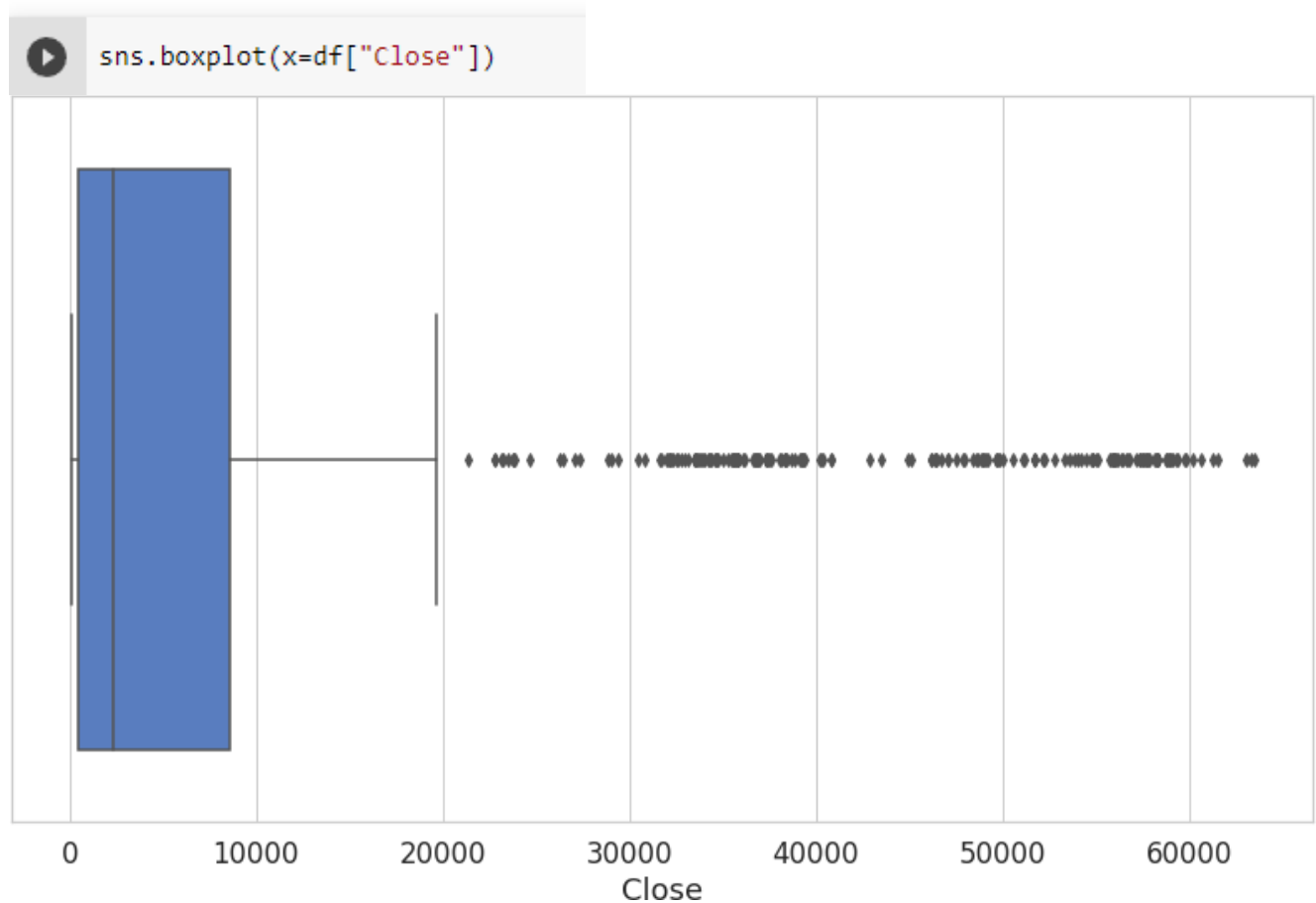


Figure 4.19: Boxplot for Close values

Closing price drives the leading factor for price prediction so that boxplot of “close” has drawn separately. Draw a single horizontal boxplot, assigning the data directly to the coordinate variable:

```
[ ] boxplot = df.boxplot(column=['Low', 'High', 'Open', 'Close'])
```

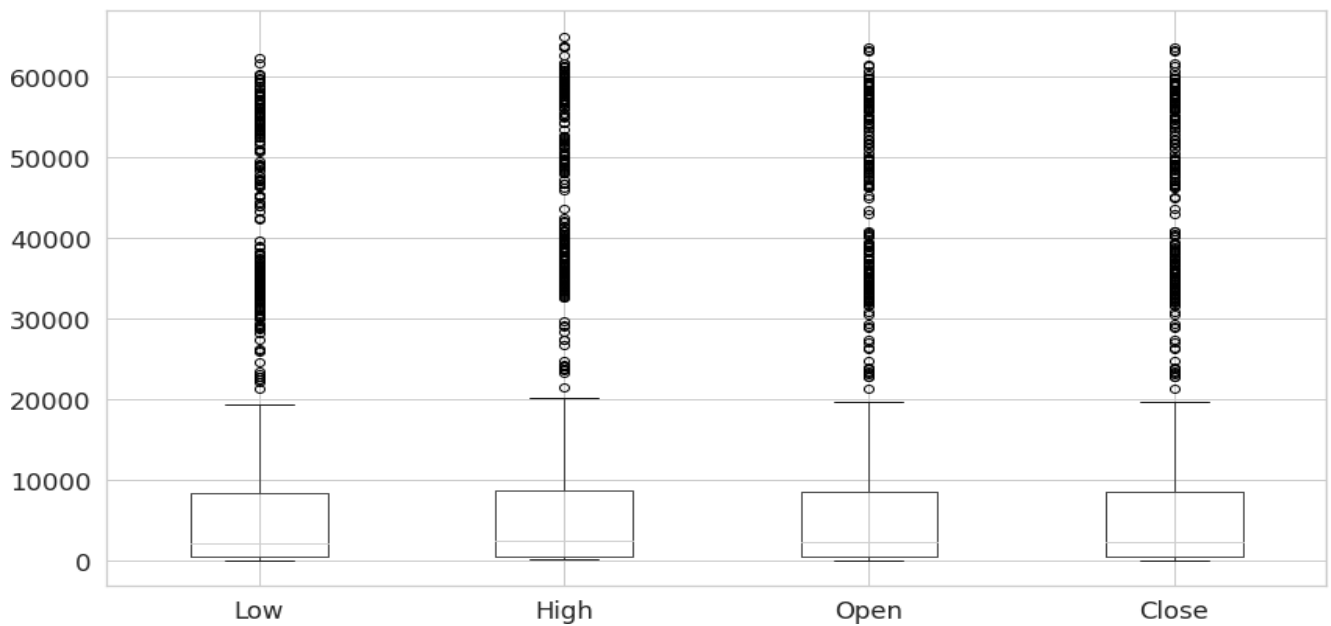


Figure 4.20: Boxplot Low, High, Open and Close combined.

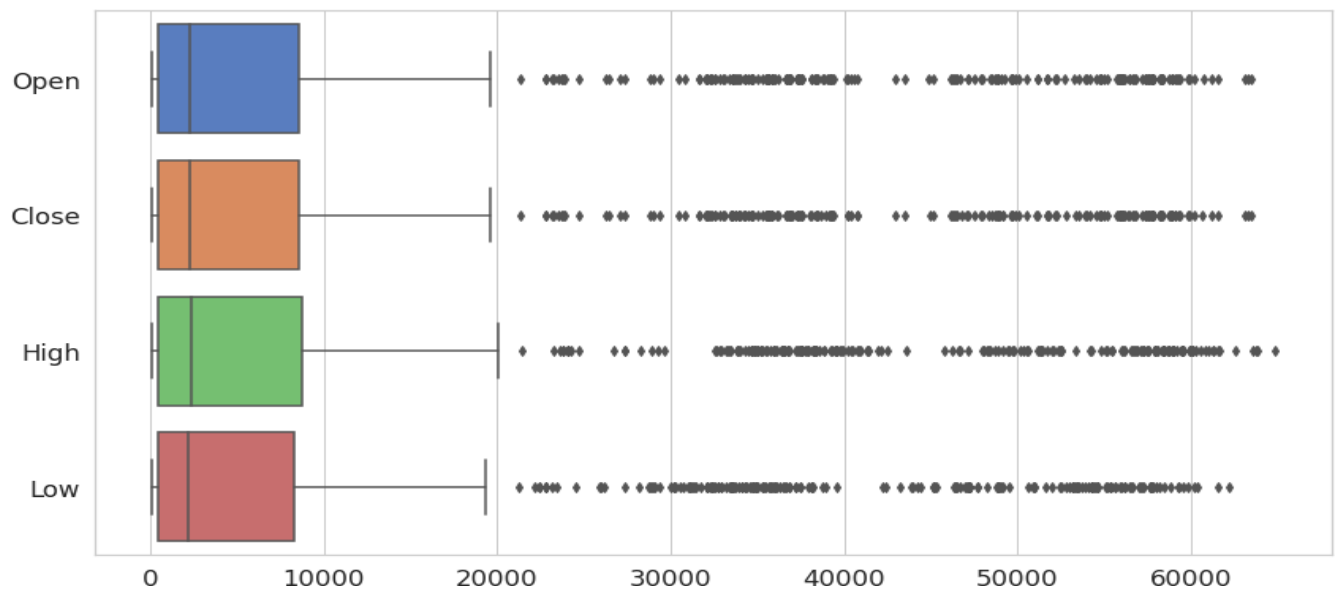


Figure 4.21: Boxplot for Low, High, Open and Close combined in horizontal orientation.

Open, close, High, and low have not much difference in numerical value so all four columns have plotted in a single diagram, so it is helpful for visual analysis, as in boxplot there is not much significant difference can be visible in both horizontal and vertical approach.

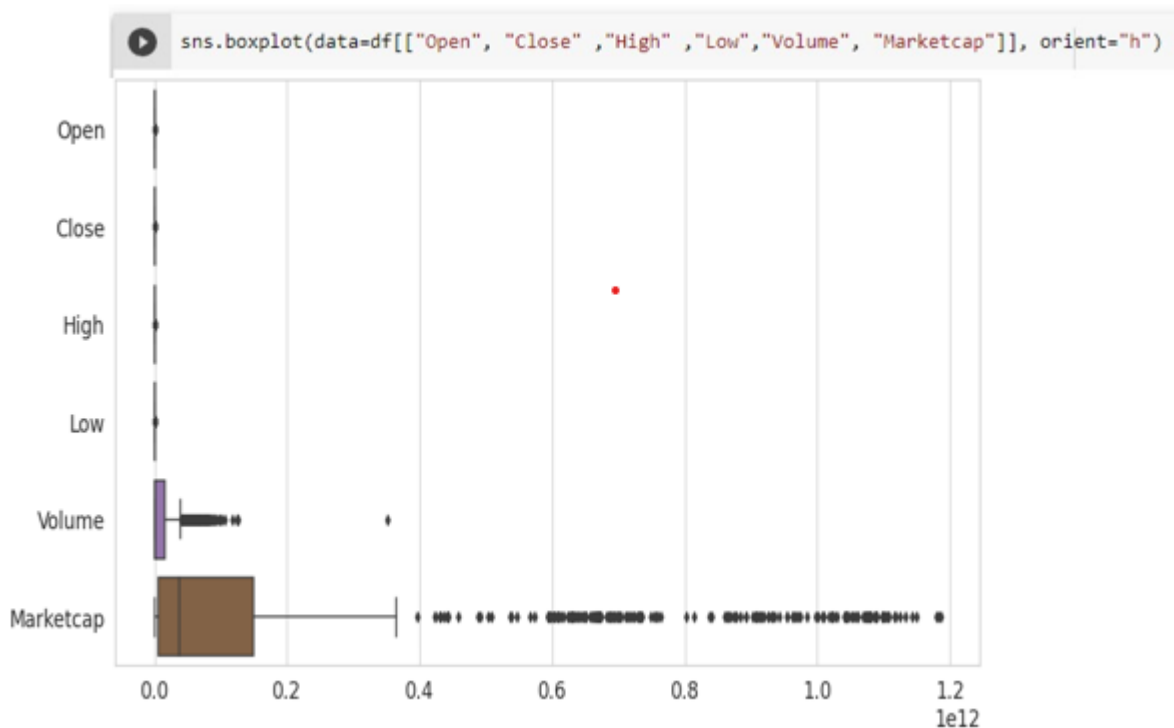


Figure 4.22: Boxplot for Low, High, Open, Close, Volume and Market Capitalization combined.

The volume and market cap have significantly higher values, so in the above boxplot diagram, all other column data seems very low. This is taken as a drawback for linear representation. such problem can be avoided by using logarithmic expression.

```
sns.boxplot(
    data=df, x="Close",
    notch=True, showcaps=False,
    flierprops={"marker": "x"},
    boxprops={"facecolor": (.4, .6, .8, .5)},
    medianprops={"color": "coral"},
)
```

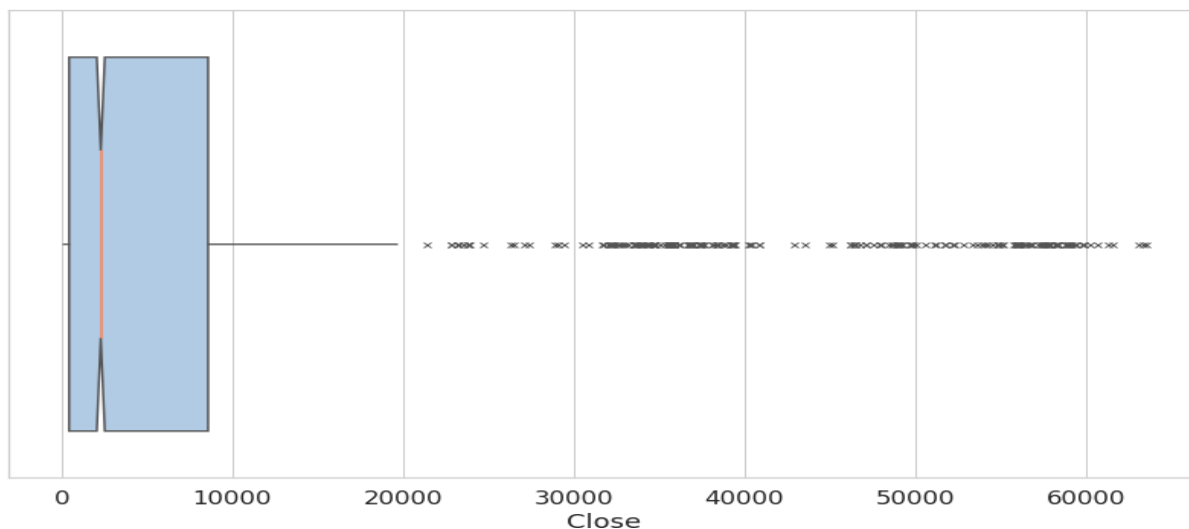


Figure 4.23: Boxplot of close column with additional information.

Passing additional value to the box plot for closed column.

```
sns.heatmap(df.corr(), annot= True ,cmap = "coolwarm")# heatmap for correlation
```



Figure 4.24: Hit map correlation.

The prices of BTC, which include the opening, closing, highest, lowest, and weighted prices, exhibited a strong correlation with each other. However, the correlation between the volume of BTC, the volume of currency traded, and the price was weak. The correlation heatmap displays a positive correlation among the variables Open, High, Low, Close, and Weighted Price, indicating that these

variables move in the same direction, either increasing or decreasing together. Which is clearly displayed in **Figure 4.24**.

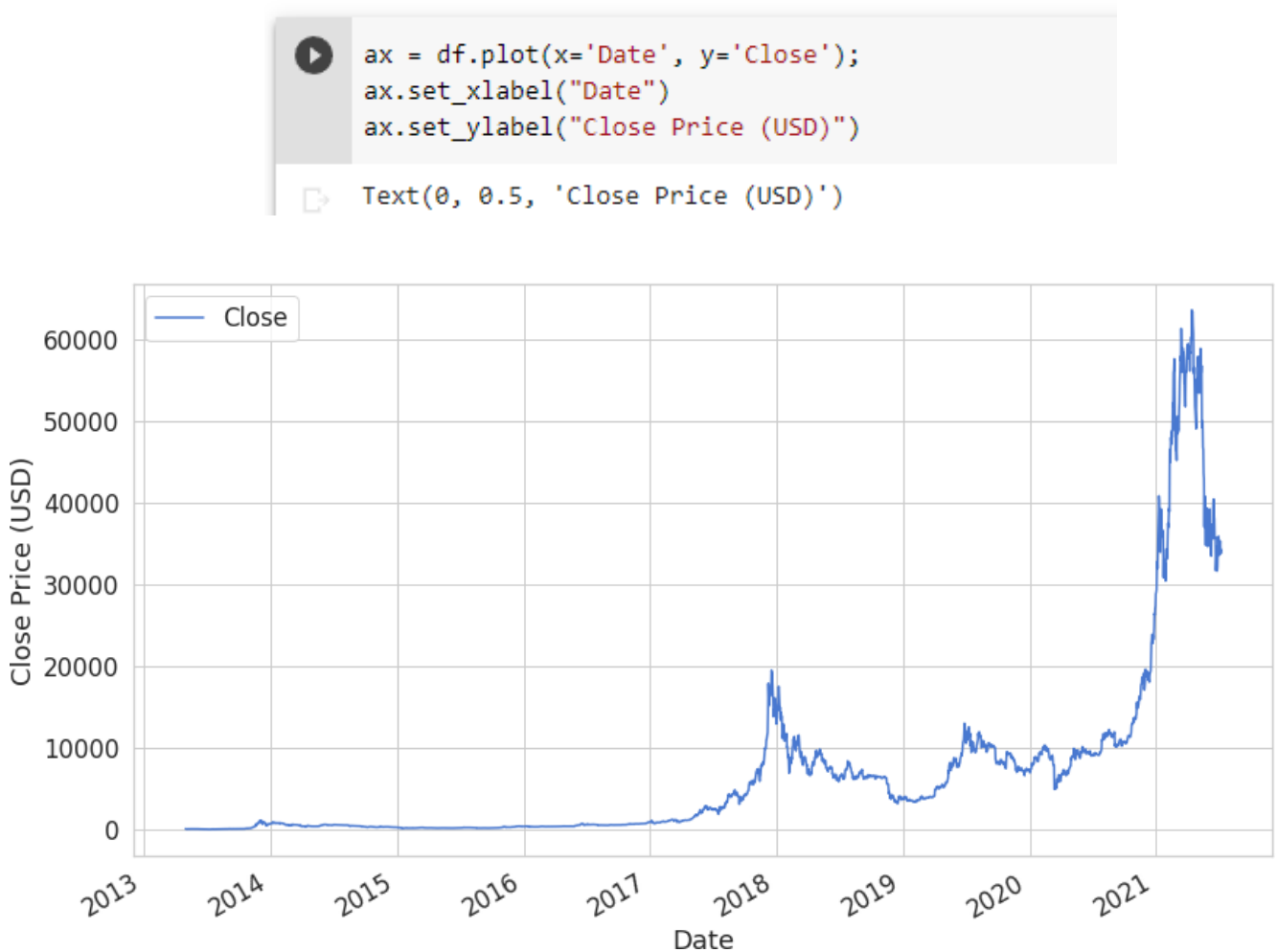


Figure 4.25: Plotting Close value (As Training and testing set)

We have plot line graph for close price. As close price is the result of previous all the prices. It can be termed as cumulative result of the past. For our prediction model it can be said a complete train and test model. From the **Figure 4.25** it is clearly visible the fluctuation of the closing price and rapid inclination from 2021.

Chapter 5

Feature engineering, Training Testing and Model Building

5.1 Feature Engineering

Feature engineering helps to significantly optimize the performance of machine learning algorithms. Machine learning algorithms are sensitive to the feature and normalization is one of the widely used scaling approaches of feature engineering.

5.1.1 Normalization

Normalization refers to the process of transforming data in a way that makes all values conform to a common scale or distribution. This can involve scaling values to a comparable metric or adjusting time scales to facilitate comparisons between similar time periods. The ultimate goal of normalization is to ensure that data can be effectively analysed and compared despite differences in their original scales or distributions. Normalization prevents variables with large values from having disproportionate effects on variables with smaller values and enables comparisons to be made between similar time periods. This is achieved by adjusting the scale or distribution of the data so that all values are on a more comparable level. The primary objective of normalization is to facilitate the analysis and comparison of data by eliminating disparities in the original scales or distributions of the variables. There are various types of normalization techniques among them we employ min-max scalar technique.

```
[ ] scaler = MinMaxScaler()

close_price = df.Close.values.reshape(-1, 1)

scaled_close = scaler.fit_transform(close_price)
```

Figure 5.1: Min Max scaler for reshaping the data.

Min-Max Scaling is a normalization technique that involves subtracting the minimum value of each column from its maximum value and then dividing the result by the range. This operation results in a new column where the minimum value is zero and the maximum value is one. The purpose of this method is to rescale the data values so that they fall within a common range, which makes it easier to compare and analyse them. By using this approach, the data is normalized between a minimum and maximum value, regardless of the original distribution of the data.

Despite the standard range (0 to -1), we altered the range to -1 to 1 to make our calculation easy and distinctive for the close price column only.

```
[ ] scaled_close = scaled_close.reshape(-1, 1)

[ ] np.isnan(scaled_close).any()

False
```

Figure 5.2: scaling an additional filter for NaN values.

We are utilizing sklearn's MinMaxScaler and it is important for us to fit the scaler on the entire data range. This is to avoid any discrepancies between the scaling of our test and training data. Once we fit the scaler, we will then apply it to both our test and training data. Our dataset has already been cleaned so that there are not any Nan values existing in our normalized table as well.

5.2 Training Testing and Model Building

5.2.1 For LSTM Model

```
SEQ_LEN = 100

def to_sequences(data, seq_len):
    d = []

    for index in range(len(data) - seq_len):
        d.append(data[index: index + seq_len])

    return np.array(d)

def preprocess(data_raw, seq_len, train_split):

    data = to_sequences(data_raw, seq_len)

    num_train = int(train_split * data.shape[0])

    x_train = data[:num_train, :-1, :]
    y_train = data[:num_train, -1, :]

    x_test = data[num_train:, :-1, :]
    y_test = data[num_train:, -1, :]

    return x_train, y_train, x_test, y_test

x_train, y_train, x_test, y_test = preprocess(scaled_close, SEQ_LEN, train_split = 0.80)
```

Figure 5.3: Data processing, Training and Testing

We will now split our training data into inputs and outputs at regular intervals of timesteps, where we look at a timesteps amount of data before making a prediction for the output value. We will also split our training data into training and validation sets.

For our model implementation, we will begin with an LSTM input layer that has 100 hidden units. We will add a dropout of 0.2 before our Dense output layer, which will have a linear activation and a shape of 1 since we are outputting the expected price. We will use mean squared error as our loss calculation and Adam as our optimizer as seen in **Figure 5.3**.

```
[ ] X_train.shape # training data
    (2312, 99, 1)

[ ] X_test.shape # testing data
    (579, 99, 1)
```

Figure 5.4: Training and testing data.

By following standard practice, we split our data set into training model and testing model. we separate 80% data for training purpose. i.e. 2312 rows of data from data set and 20% data for testing purpose, i.e. 579 rows. which is clear from the **Figure 5.4** (above picture) as well.

5.2.1.1 Model Building for LSTM

```
[ ] from keras.layers import Input, LSTM, Dense, TimeDistributed, Activation, BatchNormalization, Dropout, Bidirectional
    from keras.models import Sequential
    from keras.utils import Sequence
    from keras.layers import CuDNNLSTM
    DROPOUT = 0.2
    WINDOW_SIZE = SEQ_LEN - 1

    model = keras.Sequential()

    model.add(Bidirectional(CuDNNLSTM(WINDOW_SIZE, return_sequences=True),
                             input_shape=(WINDOW_SIZE, X_train.shape[-1])))
    model.add(Dropout(rate=DROPOUT))

    model.add(Bidirectional(CuDNNLSTM((WINDOW_SIZE * 2), return_sequences=True)))
    model.add(Dropout(rate=DROPOUT))

    model.add(Bidirectional(CuDNNLSTM(WINDOW_SIZE, return_sequences=False)))

    model.add(Dense(units=1))

    model.add(Activation('linear'))
```

Figure 5.5: Model building with necessary layers.

So far, the models I have built do not account for operating on sequential data. However, we can use a specialized class of neural network models known as Recurrent Neural Networks (RNNs) for this purpose. RNNs allow for the use of the output from the model as a new input for the same model. This process can be repeated indefinitely. One significant limitation of RNNs is their inability to capture long-term dependencies in a sequence. One way to address this issue is by using a variant of RNN called Long Short-Term Memory (LSTM), which is designed to remember information for prolonged periods of time. In the following section, we will explore how to use LSTM in Keras.

By using Bidirectional RNNs, we can train sequence data in both forward and backward (reversed) directions. In practice, this approach works well with LSTMs.

CuDNNLSTM is a fast LSTM implementation backed by CuDNN. While we think it is a good example of a leaky abstraction, it is undeniably very fast. Our output layer has a single neuron that predicts the Bitcoin price. We use a linear activation function and “adam” as optimizer. Loss will be handled by root mean square function. Which is proportional to the input.

```
[ ] import tensorflow as tf
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense, Dropout, LSTM, CuDNNLSTM
    BATCH_SIZE = 64

    history = model.fit(
        X_train,
        y_train,
        epochs=20,
        batch_size=BATCH_SIZE,
        shuffle=False,
        validation_split=0.1
    )
```

```
Epoch 1/20
33/33 [=====] - 2s 46ms/step - loss: 3.5198e-04 - val_loss: 3.6522e-04
Epoch 2/20
33/33 [=====] - 1s 41ms/step - loss: 3.9730e-04 - val_loss: 1.0688e-04
Epoch 3/20
33/33 [=====] - 1s 41ms/step - loss: 1.8247e-04 - val_loss: 2.4138e-04
Epoch 4/20
33/33 [=====] - 1s 42ms/step - loss: 1.6361e-04 - val_loss: 0.0015
Epoch 5/20
33/33 [=====] - 1s 42ms/step - loss: 3.0279e-04 - val_loss: 1.8994e-04
Epoch 6/20
33/33 [=====] - 1s 42ms/step - loss: 4.1041e-04 - val_loss: 2.0402e-04
Epoch 7/20
33/33 [=====] - 1s 42ms/step - loss: 1.1111e-04 - val_loss: 6.4000e-04
0s completed at 12:07 AM
```

Figure 5.6: Training and model fitting

Now we can fit our model on our X and Y training/validation data. We can take advantage of Keras' Early Stopping class, which will use the best weights of the model once it is no longer receiving improvements and stop the training. Next, we can break up our testing data into timesteps and split it again into inputs and expected outputs. We can then make predictions on the inputs and scale both the inputs and outputs back up. Now, we are ready to see how well our model performed.

5.2.1.2 Model Evaluation

```
[ ] model.evaluate(X_test, y_test)
```

```
19/19 [=====] - 0s 21ms/step - loss: 8.4954e-04  
0.0008495371439494193
```

Figure 5.7: Evaluating the Model.

```
[ ] plt.plot(history.history['loss'])  
plt.plot(history.history['val_loss'])  
plt.title('model loss')  
plt.ylabel('loss')  
plt.xlabel('epoch')  
plt.legend(['train', 'test'], loc='upper left')  
plt.show()
```

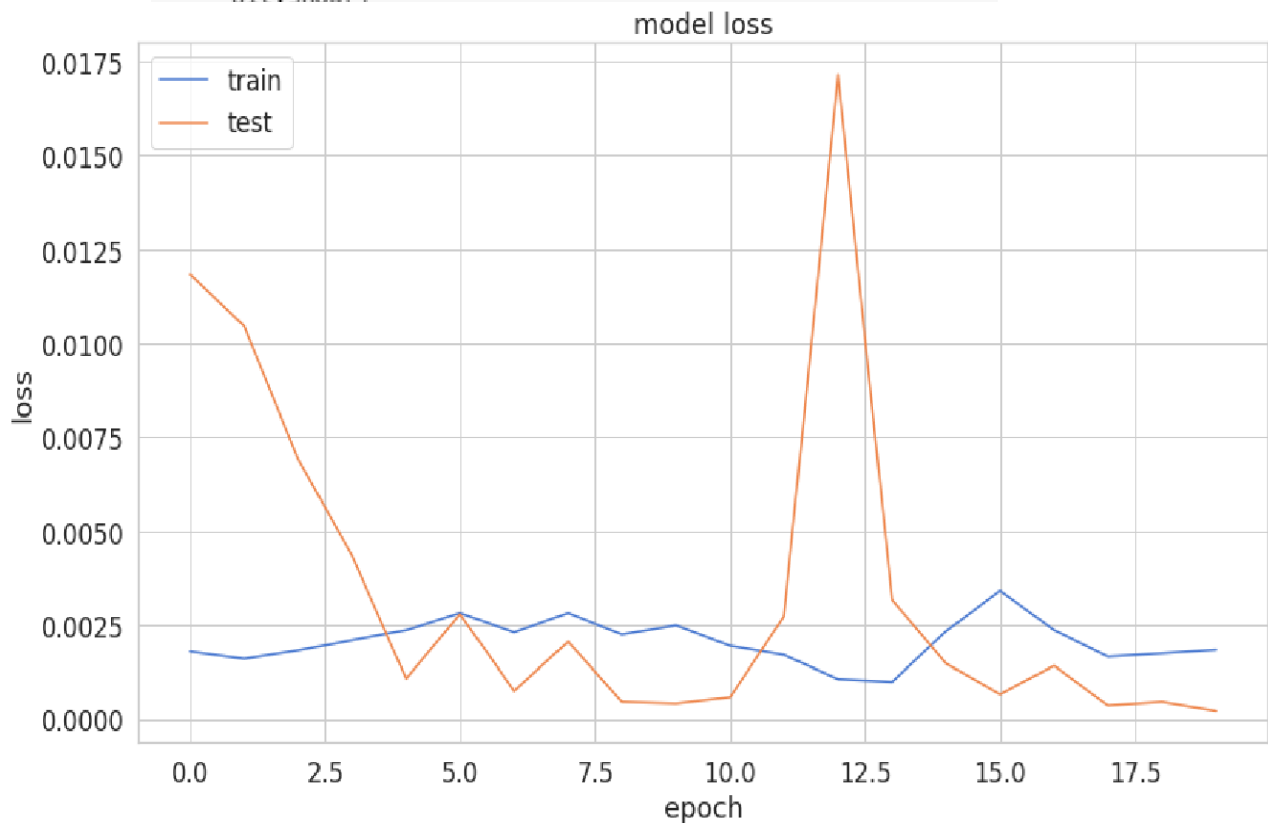


Figure 5.8: Plotting training and testing data.

When evaluating a model, we assess its performance on a specific evaluation setup by calculating performance metrics such as F1 score, MSE or RMSE. The choice of machine learning evaluation metrics should align with the time series data set that we aim to optimize with the machine learning solution. F1 score is simple based on harmonic mean. MSE is based on mean square and RMSE is

based on root mean square. The **Figure 5.8**-line graph shows training and testing modal. As blue line is for training model and orange line is for testing model.

5.2.2 For Pycret Model

```
[ ] coming_day = 10 # variable for predicting for coming 10 day
coinbit['New_Price'] = coinbit[['Close']].shift(-coming_day)#creating the new coumns for dependent variable
coinbit = coinbit[['Close' , 'New_Price']] # choose new column
coinbit # displayes close value and new dependent variables data.
```

	Close	New_Price
0	144.539993	112.669998
1	139.000000	117.199997
2	116.989998	115.242996
3	105.209999	115.000000
4	97.750000	117.980003
...

Figure 5.9: Creating new dependent variable.

A new dependent variable New_Price has been derived based on close price to predict the future value by shifting the predict day (Future) over the close price. This process is helpful for future prediction based on current closing values as shown in **Figure 5.9**.

```
[ ] df = coinbit.copy()# making a copy of data set as data frame.
x = np.array(df[df.columns])#creating independent data set
x = x[:len(coinbit)-coming_day] # remov last n row from the data set now n= coming_day=10
y = np.array(df['New_Price']) # creating dependent data set
y = y[:-coming_day] # getting all y values except last 10 rows
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state = 0, shuffle = False)#splitting training and testing data set Tra
```

Figure 5.10: Splitting the data in training and testing.

The complete data set has been divided into training and testing set. We follow the standard practice of splitting into training and testing. 80 % of data is for training the model and 20 % is for testing the model. We have created an independent dataset from the data frame as NumPy array. We remove the last prediction day so that new dependent data set has been derived. We initialize the random state value as 0 and do not allow to shuffle the values.

```
train_data = pd.DataFrame(X_train , columns = df.columns) # getting train data and transform into data frame.
train_data.head(10)# show first 10 rows of data
```

	Close	New_Price
0	144.539993	112.669998
1	139.000000	117.199997
2	116.989998	115.242996
3	105.209999	115.000000
4	97.750000	117.980003
5	112.500000	111.500000
6	115.910004	114.220001
7	112.300003	118.760002

Figure 5.11: Training data sets

The **Figure 5.11** code show that formulation of the training set and transform that Data Frame and displayed 10 columns from top of the data set. This data frame consists of 80% of total data from the original data set.

```
[ ] test_data = pd.DataFrame(X_train , columns= df.columns) # getting test data and transform into dataframe
test_data.head(10)# Show first 10 rows of data
```

	Close	New_Price
0	144.539993	112.669998
1	139.000000	117.199997
2	116.989998	115.242996
3	105.209999	115.000000
4	97.750000	117.980003
5	112.500000	111.500000
6	115.910004	114.220001
7	112.300003	118.760002
8	111.500000	123.014999

Figure 5.12: Testing dataset.

In the same fashion, the testing data set has been transformed to Data Frame and displayed 10 columns. The testing data frame consists of 20% of data from the original set.

```

regression_setup = setup(data = train_data, target = 'New_Price' , session_id =123 , use_gpu = True)# setup initialization

```

	Description	Value
0	session_id	123
1	Target	New_Price
2	Original Data	(2384, 2)
3	Missing Values	False
4	Numeric Features	1
5	Categorical Features	0
6	Ordinal Features	False
7	High Cardinality Features	False
8	High Cardinality Method	None
9	Transformed Train Set	(1668, 1)
10	Transformed Test Set	(716, 1)

Figure 5.13: Setup Initialization

The PyCaret function called "setup ()" is used to prepare the environment and data for machine learning modelling and deployment. This function requires two parameters: a dataset and a target variable. Once executed, the function automatically detects the data types of each feature and performs several pre-processing tasks on the data.

As explained in **Chapter 2** ,literature review section, PyCaret package is able to handle regression, classification, timeseries, anomaly detection, OOP. We chose to initialize the regression model, which is comparatively good for interpolation and extrapolation both. We started to use GPU (Graphical Processing Unit) to improve efficiency of the model to reduce the calculation time. The system uses “KFold” as fold generator by default.

“setup () “initiate pre-processing pipeline to the data set. Some highlights are as follows,

- Inferred data type.: It can be observed that the four features have been accurately recognized as numerical variables, while the remaining features have been identified as categorical variables.
- Train/Test Split: The dataset has been divided into two sets, namely the train set and test set, which is a common practice in machine learning. The size of the train set has been specified as 80% of the original dataset, indicating that 80% of the data will be utilized for training the machine learning model, and the remaining 20% will be employed for assessing its accuracy.
- Numeric feature normalization: The standard method is to use z-score to normalize the values.

- Target transformation: Target data (Close) values will be transformed to normal distribution if needed.

```
#Train all the model and sort it by R -square matrix(r2) and store the model.
best_model = compare_models(sort = 'r2')
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	317.4969	5.284034e+05	713.1998	0.9622	0.1346	0.0964	0.066
llar	Lasso Least Angle Regression	370.5780	5.529972e+05	732.1215	0.9602	0.2919	0.2675	0.009
lasso	Lasso Regression	357.3850	5.514736e+05	731.3831	0.9602	0.2456	0.2173	0.012
br	Bayesian Ridge	357.4169	5.514734e+05	731.3823	0.9602	0.2458	0.2175	0.009
omp	Orthogonal Matching Pursuit	357.3848	5.514735e+05	731.3831	0.9602	0.2456	0.2173	0.008
lr	Linear Regression	357.3848	5.514735e+05	731.3831	0.9602	0.2456	0.2173	0.009
lar	Least Angle Regression	357.3848	5.514736e+05	731.3831	0.9602	0.2456	0.2173	0.012
en	Elastic Net	357.3850	5.514736e+05	731.3831	0.9602	0.2456	0.2173	0.012
ridge	Ridge Regression	357.3849	5.514735e+05	731.3831	0.9602	0.2456	0.2173	0.008
huber	Huber Regressor	332.2659	5.528015e+05	732.3528	0.9601	0.1433	0.1059	0.020
knn	K Neighbors Regressor	328.7909	5.655004e+05	738.4657	0.9594	0.1386	0.0996	0.182
gbr	Gradient Boosting Regressor	329.0785	5.995877e+05	764.1338	0.9561	0.1457	0.1091	0.111

Figure 5.14: Train all model.

There are many regression algorithms to choose from and it can be difficult to determine the best one for a given dataset. The most effective way to find the best model is to try several and compare the results. Fortunately, PyCaret offers the “compare_models ()” function, which simplifies the process of comparing various models.

This is the actual beauty of using PyCaret model, as this model is termed as low-code model. It trains all the possible regression models by using root mean square matrix (also termed as R-2 matrix) and gives us the optimum output. For our particular model, Light Gradient Boosting Machine(lightgbm) model is evaluated as best with highest R2 value i.e., 0.9622 with respective MAE 317.4969 and RMSE 713.1998, From **Figure 5.14**.

Similarly, our second-best regression model is Lasso Least Angle Regression(llar)with R2 value 0.9602. In the same manner Lasso Regression(lasso) is trained as third best model with R2 0.9602 and Bayesian Ridge(br) as fourth and Orthogonal Matching Pursuit(omp) as fifth and Linear Regression(lr) as sixth model. All other trained models can be seen in the table above with their respective all possible calculative values.

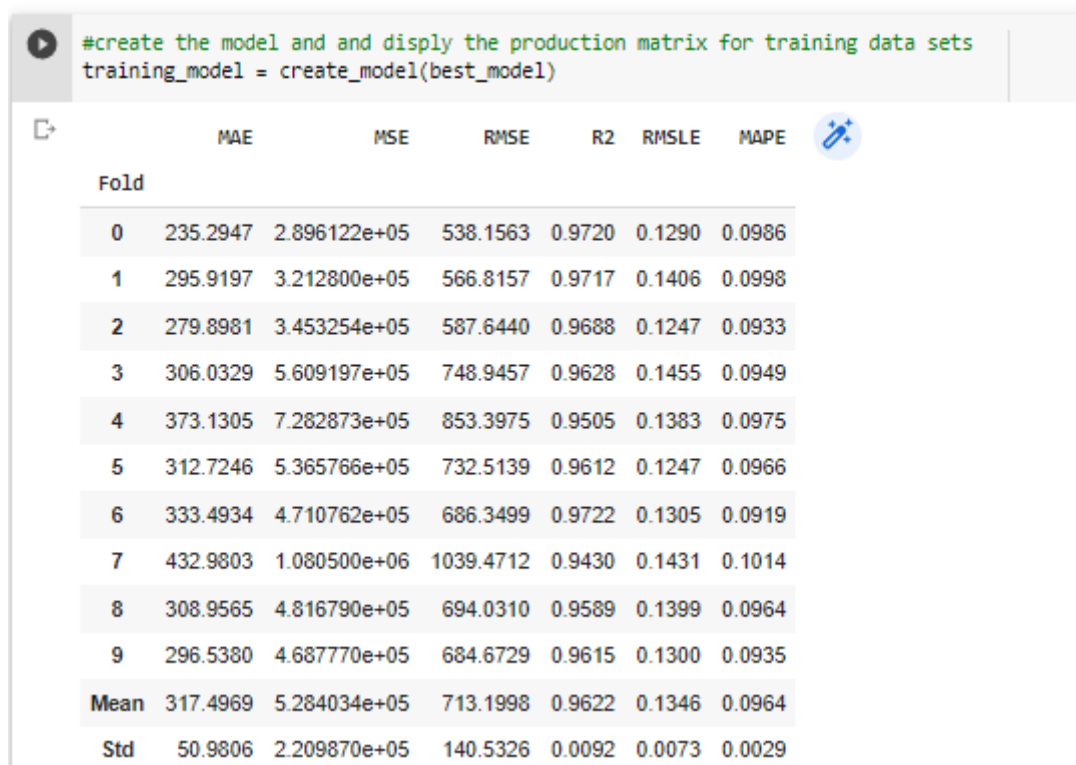


Figure 5.15: Production matrix after training

The production matrix is the summary of different model comparisons after training the data set. It is clear that men and standard deviation of the trained model. It uses **K-fold** cross validation to evaluate the model accuracy. The mean of R2 value is 0.9622 which is exactly calculated by light gradient bosting machine and its respective standard deviation is 0.0092 which is acceptable to the state

```
[ ] #model evaluation
evaluate_model(training_model)
```

INFO:logs:Initializing evaluate_model()
INFO:logs:evaluate_model(estimator=LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0, importance_type='split', learning_rate=0.1, max_depth=-1, min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31, objective=None, random_state=123, reg_alpha=0.0, reg_lambda=0.0, silent='warn', subsample=1.0, subsample_for_bin=200000, subsample_freq=0), fold=None, fit_kwargs=None, plot_kwargs=None, feature_name=None, groups=None, use_tr

Plot Type:	Hyperparameters	Residuals	Prediction Error	Cooks Distance	Feature Selection	Learning Curve	Manifold Learning
	Validation Curve	Feature Importance	Feature Importance...	Decision Tree	Interactive Residuals		

INFO:logs:Initializing plot_model()
INFO:logs:plot_model(fold=KFold(n_splits=10, random_state=None, shuffle=False), use_train_data=False, verbose=True, is_in_evaluate=True, display=None, display_format=None, estimator=LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0, importance_type='split', learning_rate=0.1, max_depth=-1, min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,

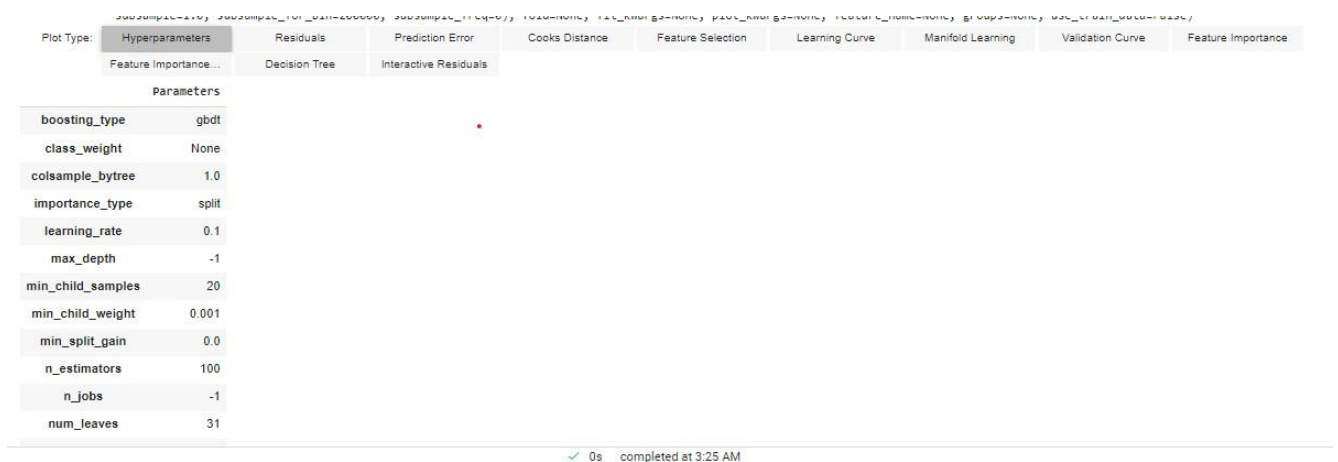


Figure 5.16: Model Evaluation

This is the best part of PyCaret model, where all the complexity has been hiding so that it is called low code. Model evaluation execute hyperparameter, residual, prediction error, cooks' distance, feature selection learning curve, manifold learning, validation curve, feature improvement, decision tree interactive residual etc...

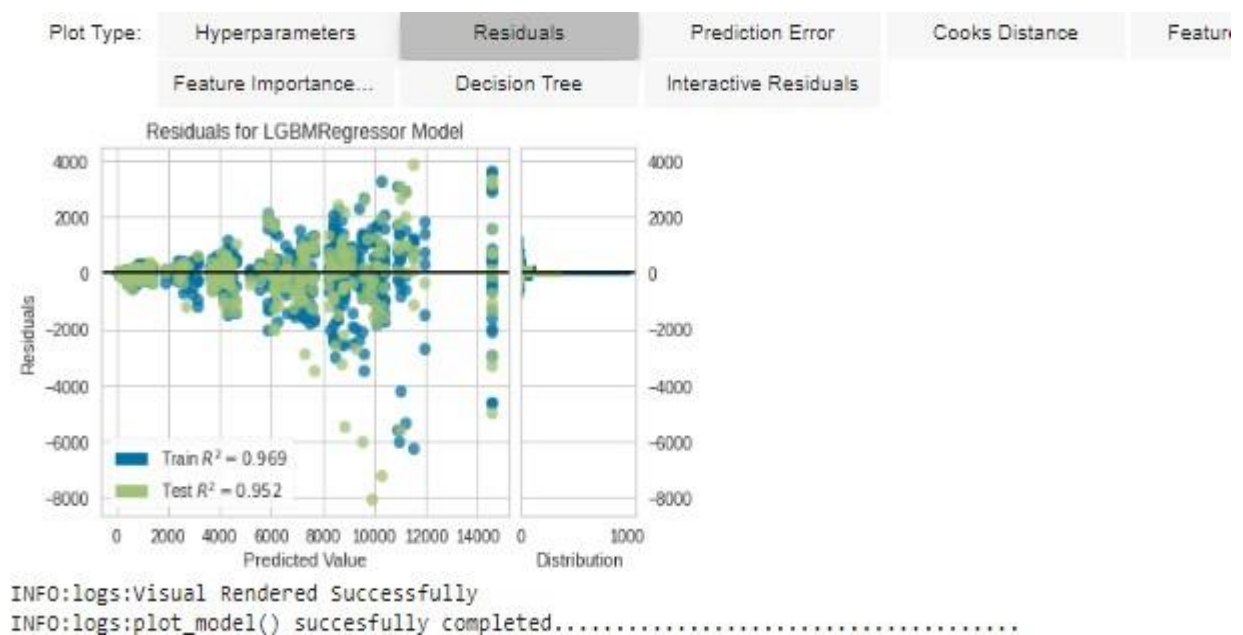


Figure 5.17: Plotting Residuals.

Residual simply means the difference between the actual value and predicted value. If the residuals of a model are all zero, then the model's predictions are perfect. Conversely, if the residuals are further away from zero, the model's accuracy is reduced. In linear regression, if the sum of squared residuals is higher, and everything else is held constant, then the R-squared statistic will be lower. The above figure clearly depicts that R-square is 0.969 and 0.952 for train and test data respectively.

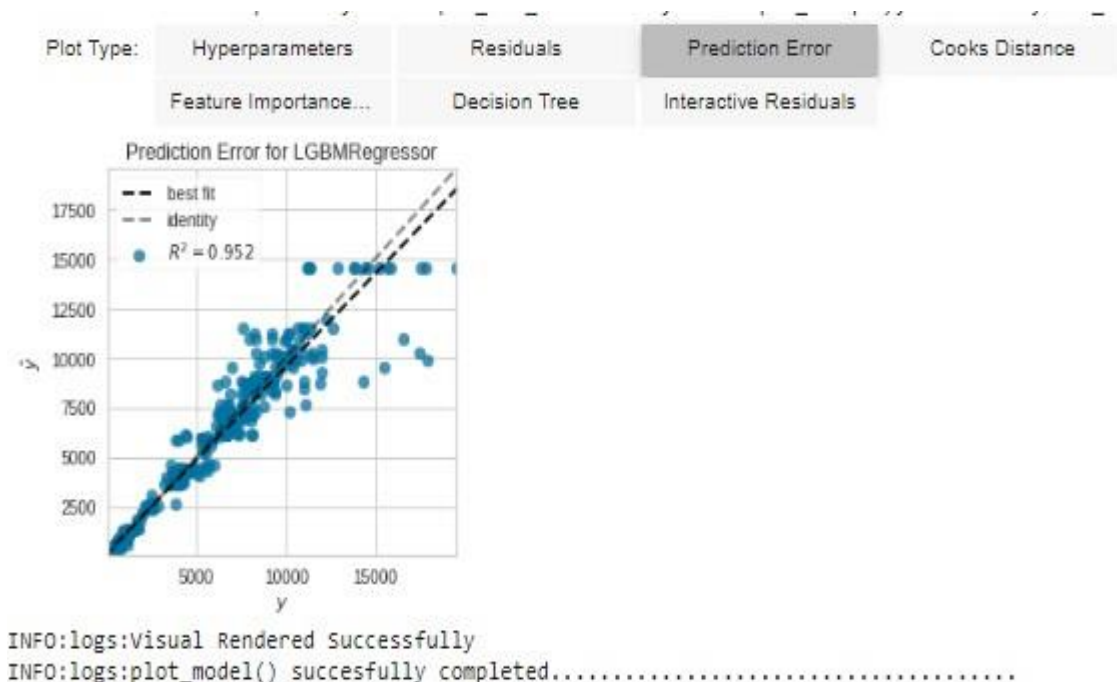


Figure 5.18: Plotting Prediction Error

Prediction error plot is slightly inclined towards X axis from the identity line. The plot also showing few outliers. The R-square value is 0.952.

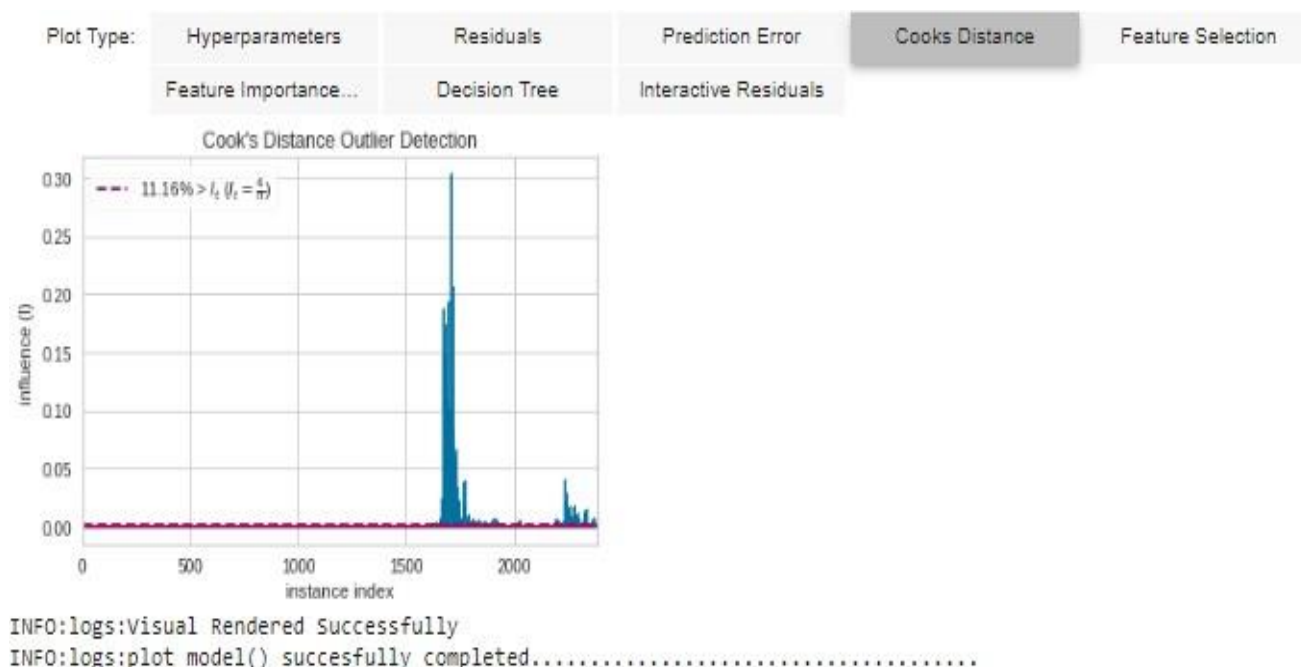


Figure 5.19: Plotting Cooks Distance.

Cooks distance in regression helps to find influential outlier from the set of predictor variable. Cooks distance is proportional to the residual value. The plot shoe that outlier and cut-offs reason exist at the highest value.

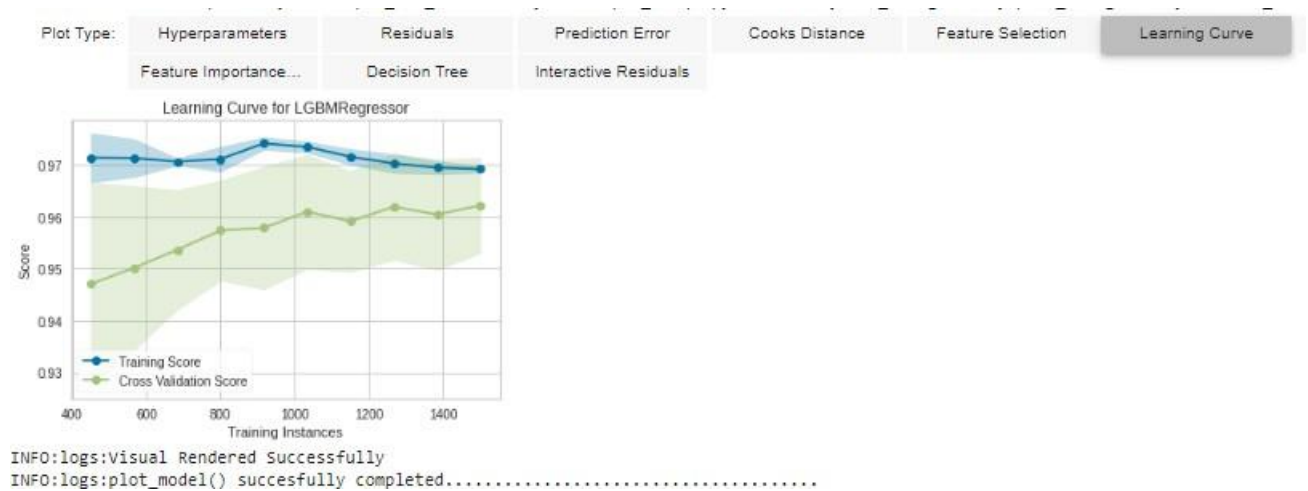


Figure 5.20: Learning Curve

The learning curve is also called training curve. It plots the optimal values of the model's loss function to validate the data set. The training score is revolving around 97%, which is perfectly okay and cross-validation score is increasingly inclined to 95-96%.

The validation curve measures the influence of close values. It gives training score which is about 97% and cross-validation score about 96%.



Figure 5.21: Validation Curve

5.3 Prediction

5.3.1 By LSTM Model

```
▶ y_hat = model.predict(X_test)

y_test_inverse = scaler.inverse_transform(y_test)
y_hat_inverse = scaler.inverse_transform(y_hat)

plt.plot(y_test_inverse, label="Actual Price", color='green')
plt.plot(y_hat_inverse, label="Predicted Price", color='red')

plt.title('Bitcoin price prediction')
plt.xlabel('Time [days]')
plt.ylabel('Price')
plt.legend(loc='best')

plt.show();
```

Figure 5.23: Prediction for LSTM Model

We use “scaler.inverse_transform()” to predict the model. After prediction, visual plotting the output in below graph. where actual price has been represented by green line and predicted price has been represented by red colour. Refer to **Figure 5.24**.

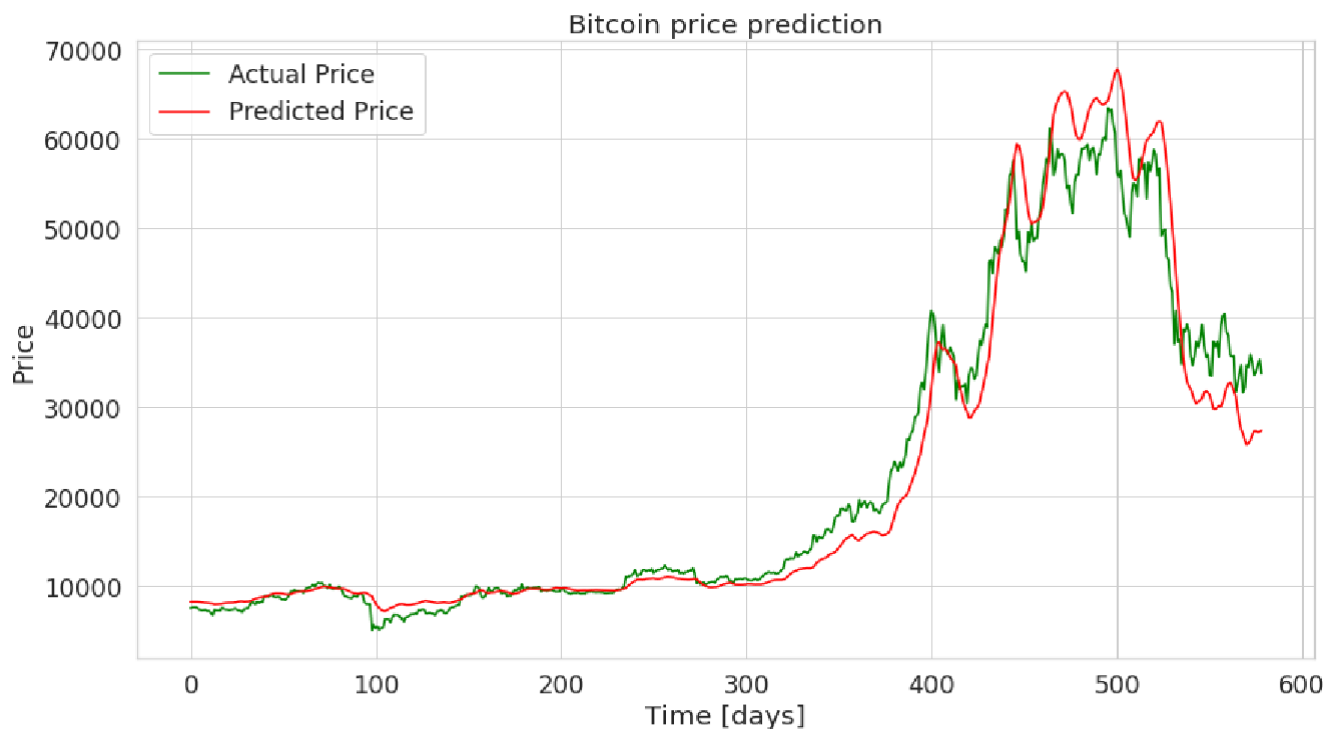


Figure 5.24: Predicted Price Visualization along with the Close Price by LSTM Model.

5.3.2 PyCaret Model

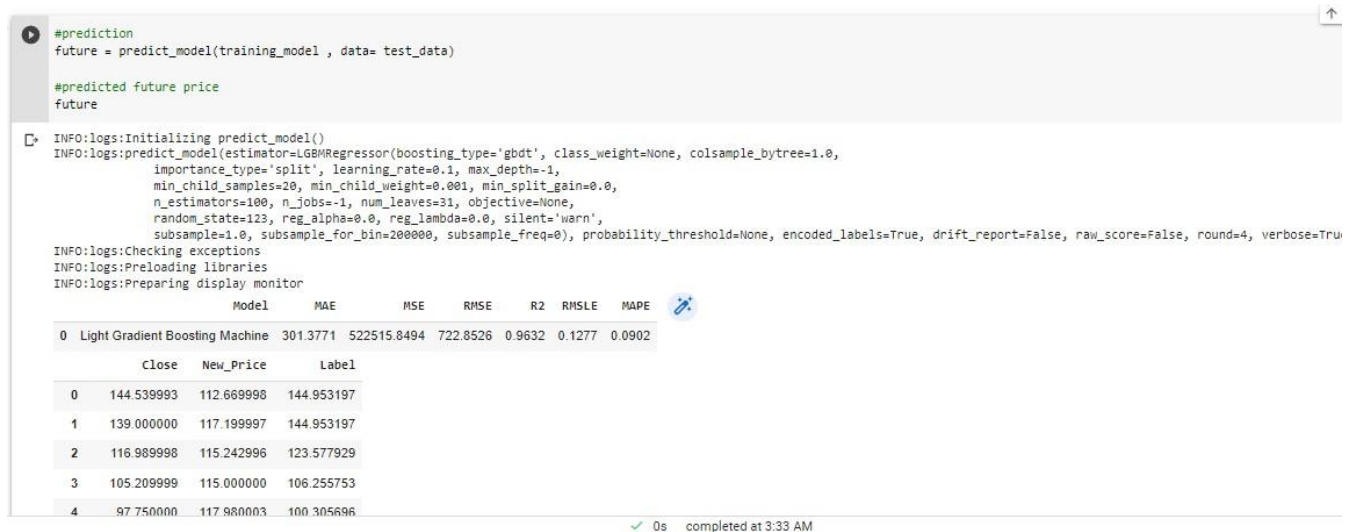


Figure 5.25: Prediction

Due to low code feature of PyCaret model, function `predict_model ()` with necessary parameter initializes it and develops the prediction model. As light gradient boosting machine is considered as taken as best model by PyCaret model so all other necessary parameters have been developed on the basis of light gradient boosting machine model. In **Figure 5.25** (above table) **New_Price** is a dependent variable and **Label** is our predicted price.

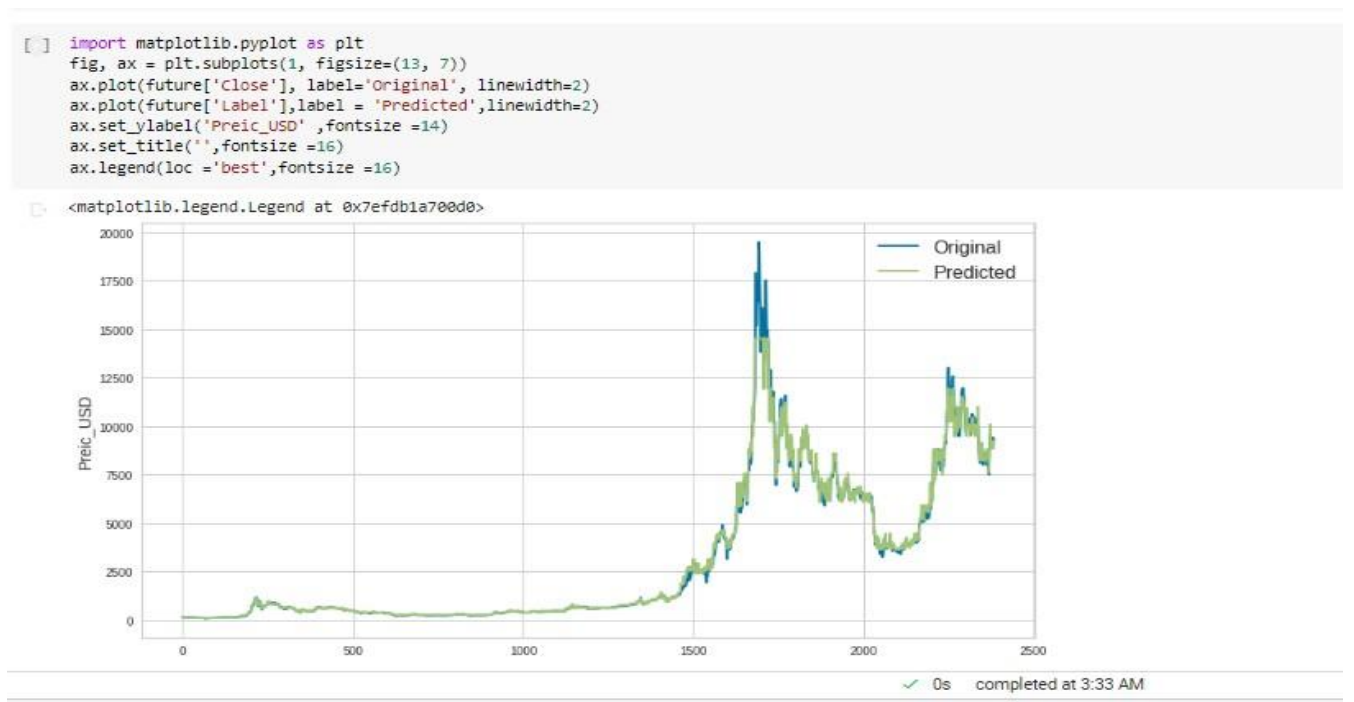


Figure 5.26: Visualization of Prediction with PyCaret Model.

The **Figure 5.26** is the plotting of actual close price and predicted price. Original close price has been represented by blue line and our predicted price has been plotted by grey line.

Chapter 6

Result, conclusion and discussion

6.1 Result

A Python programming library called PyCaret and a recurrent neural network-based model LSTM has been used to make predictions over the same dataset. This model predicts the future price so that stakeholders can directly benefited.

6.1.1 PyCaret Model

The graph in **Figure 6.1** compares the actual BTC price with the predicted price using the Pycaret library. It shows that the predicted and actual prices are very close to each other throughout the time interval, which indicates that the predictions are accurate. The Light Gradient Boosting model is considered the best because it has the lowest mean absolute percentage error (MAPE) for the BTC prediction model, which is 0.2454. The root mean squared error (RMSE) is 713.1998, and the R2 value is 0.9622, which means that the model explains 96.22% of the variability in the data. The RMSLE is 0.1346, and the MAE is 317.4969, as shown in Figure 5.14. The mean of R2 is 0.9622, and the standard deviation of R2 is 0.0092. Similarly, the mean of MAE is 317.4969, and the standard deviation of MAE is 50.9806, as seen in **Figure 5.15**. The statistical analysis of the data indicates that the predicted closing price closely matches the actual price. **Figure 5.26** shows that the difference between the predicted and actual BTC price is very small, especially during the testing set. There are only minor differences in the top few peaks of the time series.

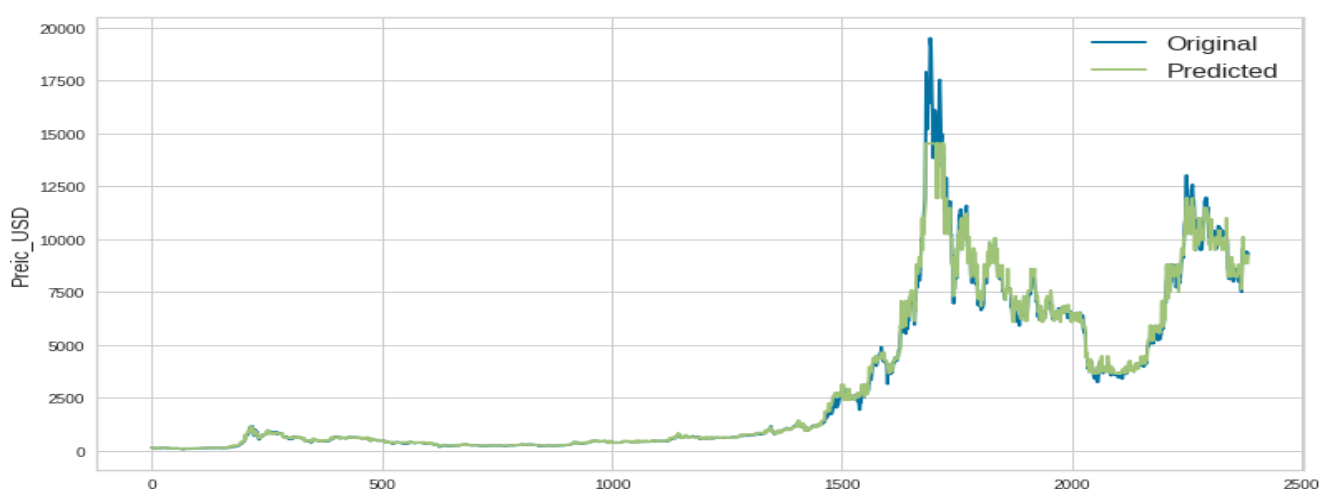


Figure 6.1: Result Predicted by PyCaret Model.

6.1.2 By LSTM Model

The line graph is seen in **Figure 6.2.** depict the forecasting by the LSTM model along with actual Price.

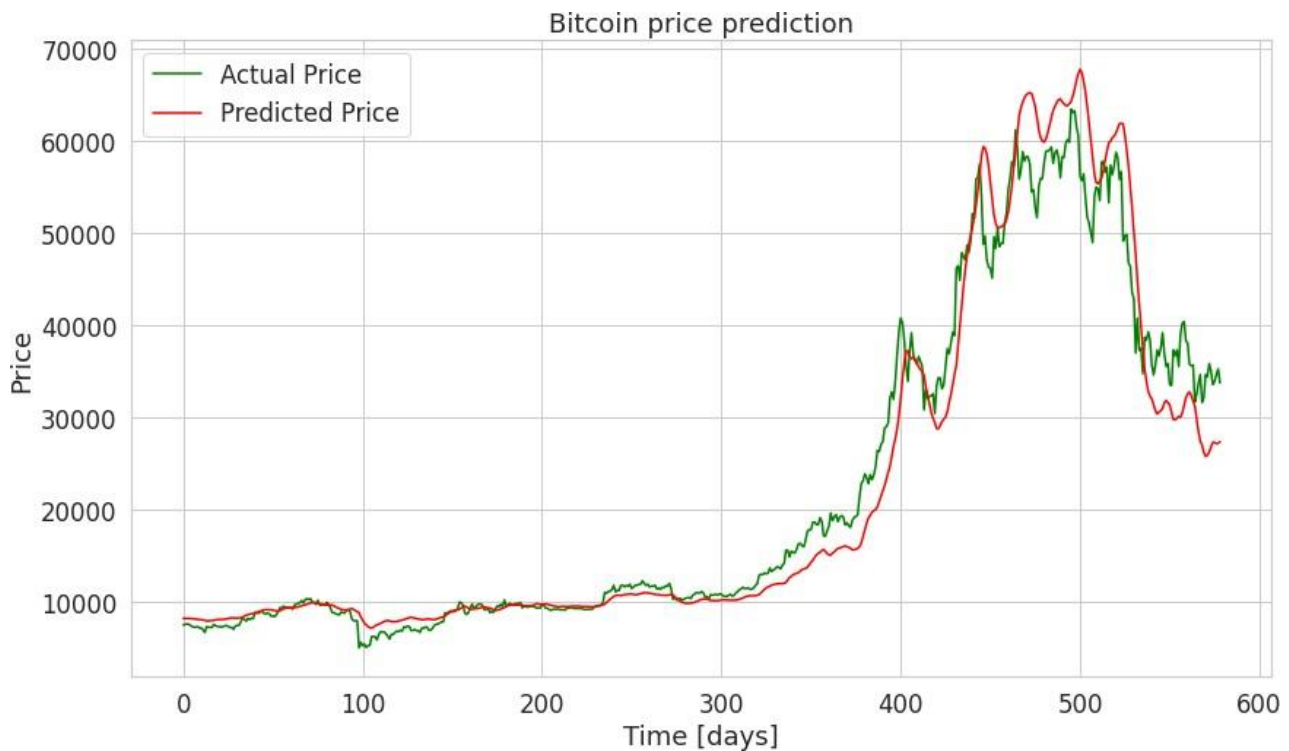


Figure 6.2: Result Predicted by LSTM Model

A special variation of Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM) model was used to predict the Bitcoin price. The machine learning model was used to predict the price based on the accuracy scores, which are presented in **Figure 6.2.** The predictive accuracy of all tested models was found to be 84%. Furthermore, the models have a probability of less than $3.08E-09$ for a true accuracy of 84%. The prediction presented in **Figure 5.24** compare the actual and LSTM-predicted price of BTC. The graph shows that the predicted price follows the trend of actual price is approximately the same over the entire interval. The mean absolute percentage error for the prediction model of BTC for LSTM is 1.1234%, and the root mean square error is 410.3.

6.2 Conclusion

Since its inception in 2008, Bitcoin has gained widespread popularity among millions of users around the world. As a result, academics have created a huge body of work on Bitcoin transactions, making it one of the most well-studied cryptocurrencies. This study focuses on conducting statistical analysis on the BTC time-series data from 2013 to 2021, applying various tools of data science and

presenting it visually, mainly applying the LSTM Model based on RNN, PyCaret library of python programming language. The future price of BTC can be predicted using the historical dataset, and Bitcoin has been increasing due to its closeness values. We built a Bidirectional LSTM Recurrent Neural Network in TensorFlow 2, and the developed model (and pre-processing "pipeline") is specific to the dataset developed by combining historical and recent data for crypto. One interesting direction of future investigation might be analysing the correlation between different cryptocurrencies and how that would affect the performance of our model.

In this project, two different machine learning approach were used to predict the prices of cryptocurrencies, with a focus on Bitcoin. Based on the results and calculations presented, the comparison of actual and predicted prices using both models can be considered efficient and reliable. However, the Pycaret model showed more accuracy than the LSTM model with substantial low differences between the actual and predicted prices for Bitcoin, as depicted in **Figure 6.2**. Therefore, it can be concluded that the Pycaret model is more suitable, reliable for predicting the prices of cryptocurrencies, particularly Bitcoin, in this case. PyCaret is also good due to its low code feature and multi-dimensional approach of features exploration.

- The python library Pycaret is reliable and acceptable for cryptocurrency prediction.
- Light gradient boosting in PyCaret model can predict cryptocurrency price comparatively better than LSTM but overall, all both algorithms represent excellent predictive result.

LSTM model is considerable model for crypto currency price forecasting with comparable accuracy.

6.3 Future Improvement

Because of limitations in resources, there are still numerous plans that we have not been able to finish or enhance. The list of potential future improvements is as follows:

- Data set optimization: we plan to use more comprehensive and homogeneous data set use by considering multiple secondary sources. which will help to enhance accuracy and reliability to the system.
- More advanced feature selection: Our future plans involve implementing a more sophisticated feature selection technique called auto-encoder (AE). AE has the ability to effectively capture the most critical features through non-linear transformation, while minimizing information loss. This makes it an effective tool for selecting useful features with reduced dimensions.

- **Involve economic prospect:** Our intention is to incorporate economic events into our features for forecasting purposes. Currently, we extract most of our features from the order book and trade data. However, we anticipate that by including economic events, we will gain new insights and alternative perspectives that will enhance our forecasting capabilities.
- **More ML tools to employ:** As financial data always been non-stationary and nonlinear, so we planned to employ more sophisticated ML tool such as gradient boosting tree (DBT), Generative Adversarial Network (GAN), ARIMA model, Support vector machine (SVM).
- **Sentiment analysis:** we will use sentiment analysis as well, which is another important factor for price variance, we will employ NLP to perform this in future.
- **Model Comparison:** Some other model comparison algorithm could be used to compare the performance between the models. we could also evaluate each model and make the comparison to find the best predicting model.

6.4 Critical Review

In this project we have developed two models to forecast the cryptocurrency using machine learning approach, correlated with financial world so that stakeholder get to know in advance to make a meaningful decision. Blockchain and its well-known application called bitcoin is considered as secure medium because of inheriting feature of trusted technology. The two machine learning models used to forecast are also very well known, transparent with high accuracy in the field of AI.

Though two well-known and today's world cohort, nascent technology used to facilitate the financial forecasting model, the end user has very less knowledge, understanding, experience and expertise. As a result, it is difficult to take the research and development fully implementable phase to the public user directly. It is needless to say the world is reshaping technologically as well as financially. In this transitional phase, the general user needs to make understood to fill flagged deployment of the whole system.

We have employed complete data science process starting from data acquisition, cleaning, transformation, exploratory data analysis in the preliminary phase and followed machine learning approach to forecast the future price. we develop 2 models' the recurrent neural network-based LSTM model and the regression-based PyCaret model. Both models forecast with higher accuracy in terms of time elapsed along with close price, R-square values and MAPE values.

Cryptocurrency, being a novel nature and technology there exist many aspects of price forecasting. Considering as many factors as possible definitely increases forecasting accuracy and enhances

prediction efficiency as well. This project particularly follows the data science process and sticks to the data science project development principals and methodology. So that the project primarily focusses into the data regardless of sentiment analysis and some other theoretical assumption. We use open value, close value, trade volume on particular day and market capitalization. All the considered data are numeric values so that the complete process become quantitative approach. However, the result is quite impressive, higher accuracy and more or less simulating the market, but another important aspect of the cryptocurrency called volatility, trustworthiness and some other theoretical qualitative hypothesis may not be fully considered in the developed ML models.

If world economic landscape, market hypothesis, law of demand and supply, technological merits along with financial co-factor would consider then better and market simulated, and more practical forecasting would be possible.

There are numerous machine learning models for forecasting, but regression and neural networks are well known. Dew to resource constraint, in this project, we only develop 2 models.

Regression based PyCaret model and recurrent neural network-based LSTM model. PyCaret model performs comparatively more accurate than the LSTM model however this model is unable to simulate the peak values. In future we will be able to employ and compare PyCaret model along with some other model such as gradient boosting, support vector machine, decision tree and make a data driven, result oriented more practical result. Which is missing in the project.

The data-driven cryptocurrency price forecasting project not only relies on data but also relies on effective project management skills, effective time management skill and previous research activities as well. Dew to resource constraints this project only limited to two techniques focuses on Bitcoin. But in future more robust machine learning technique and model can employ and develop. Also, can consider more accurate data optimization, more advanced feature selectin process, integrate recent economic landscape, sentiment analysis and fundamental theoretical framework as well.

References

- A Brief Overview of Recurrent Neural Networks (RNN). (2022, March 11). Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent/> (Accessed: 10 March 2023).
- Ahmad, H. (2019). Forecasting Future Prices of Cryptocurrency using Historical Data. Available at: <https://towardsdatascience.com/forecasting-future-prices-of-cryptocurrency-using-historical-data-83604e72bc68/> (Accessed: 23 November 2022).
- Analytics Vidhya. (2018, August 30). Build High-Performance Time Series Models using Auto. ARIMA in Python and R. Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/> (Accessed: 10 March 2023).
- Analytics Vidhya. (2018, February 8). Methods to improve Time series forecast (including ARIMA, Holt's winter). Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2018/02/time-series-forecasting-methods/> (Accessed: 10 March 2023).
- ARIMA vs Prophet vs LSTM for Time Series Prediction. (2022, January 4). Neptune.ai. Available at: <https://neptune.ai/blog/arima-vs-prophet-vs-lstm/> (Accessed: 10 March 2023).
- Baheti, P. (2022, March 8). 12 Types of Neural Networks Activation Functions: How to Choose? Available at: <https://www.v7labs.com/blog/neural-networks-activation-functions/> (Accessed: 10 March 2023).
- Bitcoin Price Prediction Using Recurrent Neural Networks and LSTM. Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/05/bitcoin-price-prediction-using-recurrent-neural-networks-and-lstm/> (Accessed: 10 March 2023).
- Bitcoin USD (btc-usd) STOCK historical prices & data. (2021, March 28). Retrieved March 29, 2021, from <https://finance.yahoo.com/quote/BTC-USD/history/> (Accessed: 10 March 2023).
- Champaneria, M. (2023, February 27). Introduction to Time Series Data Forecasting. Analytics Vidhya. Available at: [https://www.analyticsvidhya.com/blog/2023/02/introduction-to-time-series-data- /](https://www.analyticsvidhya.com/blog/2023/02/introduction-to-time-series-data-/) (Accessed: 10 March 2023).
- Conway, L. (2020, December 09). Why is bitcoin's price rising? from <https://www.investopedia.com/tech/cryptocurrency-this-week/> (Accessed: 10 March 2023).

European Commission (2020), The Digital Services Act package | Shaping Europe's digital Future Available at: <https://ec.europa.eu/digital-single-market/en/digital-services-act-package> (Accessed: 24 December 2022).

Fang, F.; Ventre, C.; Basios, M.; Kong, H.; Kanthan, L.; Li, L.; Martínez-Rego, D.; Wu, F. Cryptocurrency Trading: A Comprehensive Survey. arXiv 2020, arXiv:2003.11352. Available at: <https://arxiv.org/abs/2003.11352> (Accessed: 1 June 2022).

Financial Stability Board (2017), Artificial intelligence and machine learning in financial services Market developments and financial stability implications, <http://www.fsb.org/emailalert> (Accessed: 27 August 2022).

How to Create an ARIMA Model for Time Series Forecasting in Python? (2020, October 29). Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/10/how-to-create-an-arima-model-for-time-series-forecasting-in-python/> (Accessed: 10 March 2023).

IBM (2020), The Four Vs of Big Data | IBM Big Data & Analytics Hub, Available at: <https://www.ibmbigdatahub.com/infographic/four-vs-big-data> (Accessed: 1 December 2020)

IBM. (2023). What is Blockchain Technology - IBM Blockchain | IBM. Wwww.ibm.com. <https://www.ibm.com/topics/blockchain/> (Accessed: 10 March 2023).

JPMorgan (2019), Machine Learning in FX, Available at: <https://www.jpmorgan.com/solutions/cib/markets/machine-learning-fx> (Accessed :14 December 2022).

Liu, Y.; Zhang, L. Cryptocurrency Valuation and Machine Learning Model. SSRN Electron. J. 2020, Academic Press, pp 1–13.

Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available at: <https://bitcoin.org/bitcoin.pdf> (Accessed: 1 June 2022).

Research Crypto Forecasting. (n.d.). Kaggle.com, from <https://www.kaggle.com/c/g-research-crypto-forecasting/> (Accessed :24 November 2022)

Sharma R. (2022) Is There a Cryptocurrency Price Correlation to Equity Markets? Investopedia. The Technologist (2020), Does the future of smart contracts depend on artificial intelligence? Available at: <https://www.thetechnolawgist.com/2020/12/07/does-the-future-of-smartcontracts-depend-on-artificial-intelligence/> (Accessed: 12 December 2022).

Weinstein, L. (2020). Research Guides: Fintech: Financial Technology Research Guide: Cryptocurrency & Blockchain Technology. Guides.loc.gov. <https://guides.loc.gov/fintech/21st-century/cryptocurrency-blockchain/> (Accessed: 10 March 2023).

Welcome to PyCaret - PyCaret Official. (2020). Gitbook.io. <https://pycaret.gitbook.io/docs/> (Accessed: 10 March 2023).

Appendix:1

Project Process Documentation

Project Process Documentation:1

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:1	Date /Time: 5, January 2023/ 11:00 AM
Agenda of the meeting: Idea present/ share to supervisor	
Discussion of agenda items: Finalize my area of interest and speak to supervisor regarding my preliminary research and the tentative idea of finalizing to work related to blockchain technology and cryptocurrency.	
Summary of the agreed action plan: Professor Keshav suggest me few papers to so that I can get solid idea of what can I do, what is the best for me.	
Notes: It is the first meeting and I finalize the area of my project.	

Project Process Documentation:2

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:2	Date /Time: 20, January 2023 10:00 AM
Agenda of the meeting: The topic finalize, project specification	
Discussion of agenda items: Finalize my topic according to my interest and reviewed the paper as given to me in the first meeting. And wide discussion about am and objectives.	

Summary of the agreed action plan: I decide to cryptocurrency price forecasting by employing AI/ ML tools. Also develop tentative aim and objectives.

Notes: I finalize my topic "AI/ML based cryptocurrency price forecasting under changing market condition" and also start to develop specification.

Project Process Documentation:3

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:3	Date /Time: 15, February 2023 12:00 PM
Agenda of the meeting: Relevant literature review and fundamental understanding	
Discussion of agenda items: Discuss about blockchain technology, cryptocurrency and financial market from crypto prospect. Possible data source, possible technique and development strategy and approval of specification	
Summary of the agreed action plan: ML algorithm (PyCaret and LSTM) will use to forecast the future price. Start to develop the programme.	
Notes: This meeting approves my project specification.	

Project Process Documentation:4

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:4	Date /Time: 15, February 2023/10:00 AM
Agenda of the meeting: ML model development, Interim report	

Discussion of agenda items: Discuss about cryptocurrency and financial price forecasting algorithms. I have presented my developed model and also discuss regarding errors and accuracy of output.
Summary of the agreed action plan: Try to solve the error and also try another model so find co-relation. I also submitted interim report.
Notes: This is more specific towards programming approach.

Project Process Documentation:5

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:5	Date /Time: 2, March 2023 /10:00 AM
Agenda of the meeting: ML model development, Interim report feedback	
Discussion of agenda items: Discuss about cryptocurrency and financial price forecasting algorithms. I have presented my developed model and also discuss regarding errors and accuracy of output. And lacking point in interim report.	
Summary of the agreed action plan: Try to solve the error and also try another model so find co-relation.	
Notes: This is more specific towards programming approach. And leads me to start final reporting.	

Project Process Documentation:6

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:6	Date /Time: 14, March 2023/ 11:30 AM

Agenda of the meeting: Result interpretation and discussion	
Discussion of agenda items: small demo presentation the result and finding and outliers.	
Summary of the agreed action plan: need to develop comparative model and find best one	
Notes: This is all about result interpretation and methodology execution.	

Project Process Documentation:7

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:7	Date /Time: 12, April 2023/11:00 AM
Agenda of the meeting: Final report Draft discussion	
Discussion of agenda items: how to format the final report,	
Summary of the agreed action plan: Need to follow the standard practice, UWS report standard, figure and caption, citation should be in standard format.	
Notes: I have presented draft final report, supervisor comment on that to make it standard.	

Project Process Documentation:8

Student Name: Kiran Babu Basnet	Supervisor Name: Pro. Keshav Dahal
Meeting Number:8	Date /Time: 18, April 2023/1:30 PM
Agenda of the meeting: Final report improvement discussion	

Discussion of agenda items: how to standardize the final report,
Summary of the agreed action plan: Need to follow the standard practice, uws report guidelines, figure and caption, citation should be in standard format and need to standardize result section.
Notes: I have presented final draft final report, supervisor comment on that to make it standard and ready for final submission.

Appendix:2

Code File

We have uploaded my complete project in Google cloud. Complete project can access via following link.

https://drive.google.com/drive/folders/1QdowxYQ-dyh6PkMjZ-a1UMRRgADc3I_z?usp=share_link

Strep-by-step guide to run the project.

Please download the complete project and follow the steps to run the code.

- Follow **Chapter 4** to run step by step code for both PyCaret model and LSTM model for exploratory data analysis.
- Follow **Chapter 5** for feature engineering training testing and prediction for both models.

University of the West of Scotland
School of Computing, Engineering and Physical Sciences

MSc Project Specification

Student name: Kiran Babu Basnet

Banner ID: B00728243

Email: B00728243@studentmail.uws.ac.uk, kiransbanset@gmail.com

Project being undertaken on full-time basis:

MSc Programme/stream: MSc Advance Computing (BIG DATA)

MSc Programme Leader: Prof Dr. Naeem Ramzan

Project Title:

Artificial Intelligence (AI) based cryptocurrency price forecasting system under changing market condition.

Research Question to be answered:

- Q1. What are the most discriminative, relying factor of cryptocurrency price forecasting?
Q2. How does the crypto stakeholder be provided with quantitative estimate of specific feature fluctuation in changing market condition?
Q3. How to use the most discriminating, relying on factor for predictive model development in machine learning for financial time series, real time crypto financial data?

Outline (overview) and overall aim of project:

Evaluate the statistical dependency of AI features considered for crypto financial forecasting employing artificial intelligence (AI) and machine learning (ML) modelling, which provides the crypto stakeholders with quantitative estimate and features categories on AI tools. Perform a complete data science process to know insides of top rated cryptos so that people get benefited using meaningful investment on it.

- The main aim of this project is to make a reasonable price forecasting for the top-rated cryptocurrency price by considering most discriminating factor 's data built and train the model, test the model by employing well known machine learning approach.
- Built a real time cryptocurrency data driven visual system by retrieving the live crypto financial data via web socket API and historical data.

- Understand the working of new technology called Blockchain and its biggest application till now called cryptocurrency and its influence in financial marketplace.

Objectives (list of tasks to be undertaken to achieve overall aim of the project and to answer the research question posed):

- To make deep understanding about blockchain technology and its biggest application called cryptocurrency and analyse it into financial world approach. To understand the crypto financial market and identify its dependable factor via intensive literature review which has already been published in different publication media.
- Analyse the crypto financial data and develop the AI/ML model, examine the accuracy, and deploy the model for forecasting.
- Testing the output of developed model on the basis of different machine learning (ML) model to the real-world historic data to simulate the financial rise and fall scenario and determine variation so that to finalize the best forecasting.

Relationship of proposed project to MSc programme/stream:

This project requires knowledge in Data Science (especially financial data) and this fits well with courses taught during the MSc Advanced Computing (Big Data) such as Data Mining and Visualisation, Advanced Data Science). It is an opportunity to apply and develop skills that I learnt during the classes.

Indicative reading list (references to be correctly presented) and resources:

References

Ahmad, H. (2019). Forecasting Future Prices of Cryptocurrency using Historical Data. Available at <https://towardsdatascience.com/forecasting-future-prices-of-cryptocurrency-using-historical-data-83604e72bc68/> (Accessed : 23 November 2022).

European Commission (2020), The Digital Services Act package | Shaping Europe's digital Future Available at: <https://ec.europa.eu/digital-single-market/en/digital-services-act-package> (Accessed: 24 December 2022).

Fang, F.; Ventre, C.; Basios, M.; Kong, H.; Kanthan, L.; Li, L.; Martínez-Rego, D.; Wu, F. Cryptocurrency

Trading: A Comprehensive Survey. arXiv **2020**, arXiv:2003.11352. Available at: <https://arxiv.org/abs/2003.11352> (Accessed: 1 June 2022).

Financial Stability Board (2017), Artificial intelligence and machine learning in financial services Market developments and financial stability implications, <http://www.fsb.org/emailalert> (Accessed : 27 August 2020).

IBM (2020), The Four V's of Big Data | IBM Big Data & Analytics Hub, Available at : <https://www.ibmbigdatahub.com/infographic/four-vs-big-data> (Accessed:1 December 2020).

JPMorgan (2019), Machine Learning in FX, Available at : <https://www.jpmorgan.com/solutions/cib/markets/machine-learning-fx> (Accessed :14 December 2020).

Liu, Y.; Zhang, L. Cryptocurrency Valuation and Machine Learning Model. SSRN Electron. J. **2020**, Academic Press , pp 1–13.

Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available at: <https://bitcoin.org/bitcoin.pdf> (Accessed : 1 June 2022).

Research Crypto Forecasting. (n.d.). Kaggle.com. Retrieved November 30, 2022, from <https://www.kaggle.com/c/g-research-crypto-forecasting/> (Accessed :24 November 2022)

Sharma R. (2022) Is There a Cryptocurrency Price Correlation to Equity Markets? Investoedia. Available at : <https://www.investopedia.com/news/are-bitcoin-price-and-equity-markets-returns-correlated/> (Accessed: 23 November 2022).

The Technolawgist (2020), Does the future of smart contracts depend on artificial intelligence? Available at : <https://www.thetechnolawgist.com/2020/12/07/does-the-future-of-smartcontracts-depend-on-artificial-intelligence/> (Accessed :12 December 2022).

Marking scheme:

Introduction 5%
Literature review 15%
Explanatory Data Analysis 15%
Pipelines (data pre-processing) 10%
Implementation of Different Machine Learning models 30%
Evaluation 10%
Conclusion and Recommendation 5%
Critical self-evaluation 10%

Supervisor:

Professor Keshav Dahal

Moderator:

Dr. Ravi Koirala

Programme Leader:

Professor Dr. Naeem Ramzan

Date specification submitted:

1/14/2023

Please complete the 'ethics' form below for all projects.

MSc PROJECT – REQUIREMENT FOR ETHICAL APPROVAL

SECTION 1: TO BE COMPLETED BY THE STUDENT

Does your proposed research involve: research with human subjects (including requirements gathering and product/software testing), access to company documents/records, questionnaires, surveys, focus groups and/or other interview techniques? Does your research entail any process which requires ethical approval? (please enter √ in the appropriate box)

YES		You must submit an application for approval to the Ethics Review Manager
NO	√	You do not need to submit an application to the Ethics Review Manager

Name of Student (Print name): Kiran babu Basnet

Signature: Kiran

Date:1/14/2023

SECTION 2: TO BE COMPLETED BY THE PROJECT SUPERVISOR

I understand that the above project does not require ethical approval.

Supervisor (print name): Prof Keshav Dahal

Signature:

Date:

IMPORTANT: please note that by signing this form all signatories are confirming that any potential ethical issues have been considered and, where necessary, an application for ethical approval has been/will be made via the Ethical Review Manager software.

Any project requiring ethical approval, but which has not been given approval will not be accepted for marking.

Ethical approval cannot be sought in retrospect.