Final Year Project

Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms

By

Tarun Aitha

TP058015

APD3F2202CS(DA)

A project submitted in partial fulfilment of the requirements of Asia Pacific University of Technology and Innovation for the degree of

B.Sc. (Hons) Computer Science with specialism in Data Analysis

Supervised by Mr. Raheem Mafas

2nd Marker: Mr. Mohammad Namazee Mohd Nizam

September 2022

Acknowledgment

First and foremost, I would like to offer my most sincere appreciation to Mr. Raheem Mafas, the supervisor of my final year project, for his patience in offering advice during the whole of my preparation of this inquiry report. I couldn't have done it without him. During each and every meeting, he was incredibly encouraging to me and provided the finest advice that he could think of. In addition to this, he questioned me to make sure that my development is proceeding as anticipated. It is a fact that without the assistance provided by Mr. Raheem Mafas, it would not have been possible to complete this report in the manner in which it was.

Next, I would want to express my gratitude to Mr. Dhason Padmakumar, who is our FYP professor. In his seminars, he has flawlessly provided an explanation of each activity and method that must be completed in order to compile an investigative report. He carefully responded to any uncertainties or queries raised by the kids and did so in a timely manner. Because he is such a loving educator, he will answer any question, no matter what time it is submitted, even if it comes in at midnight.

I would like to show my appreciation to all of the professors at APU who have instructed me over the course of the past three years, and I would like to make a commitment to incorporating as much of the information that I have gained into my capstone project or into my future professional life as I possibly can.

In closing, I would want to express my gratitude to all of my friends, siblings, and parents for the unending love and support they have shown me throughout my life. My self-assurance has been much enhanced, and I've been able to break through previously insurmountable barriers thanks, in large part, to the financial and emotional help I've received.

Table of Contents

Acknowledgment 2
1. Introduction to the Study 5
1.1 Background to the project 5
1.2 Problem statement 6
1.3 Rationale 6
1.4 Potential Benefits 6
1.4.1 Tangible Benefits 7
1.4.2 Intangible Benefits 7
1.5 Target users 7
1.6 Objective and Scope of the project 7
1.6.1 Aim 7
1.6.2 Objectives 7
1.6.3 Deliverables 7
1.7 Nature of Challenge 8
1.8 Project Plan 8
2. Literature Review 9
2.1 Introduction 9
2.2 Domain Research 9
2.2.1 Bitcoin 9
2.2.2 Machine Learning 10
2.2.3 Regression Analysis 10
2.2.4 Neural Networks 11
2.2.5 LSTM (Long term-short Memory) 12
2.2.6 Autoregressive Integrated Moving Average (ARIMA) 13
2.2.7 Support Vector Regression (SVR) 14
2.3 Similar Systems 15
2.3.1 Prediction of bitcoin by Anuradha Datta 15
2.3.2 Prediction of cryptocurrencies by J. Hamayel and Y. Owada 16
2.3.3 Bitcoin price prediction using linear regression by V. Ramesh and Alvin Ho. 18
2.3.4 Differentiation of similar systems. 18
3. Technical research 21
3.1 Introduction 21
3.2 Programming Language Chosen 21
3.2.1 Comparative analysis on Python, R programming language and SAS 21
3.2.2 Python 23
3.3 Integrated Development Environment (IDE) 24
3.3.1 Google Collab 25
3.3.2 Jupyter Notebook 26
3.3 Libraires Chosen 26
3.3.1 Numpy 26
3.3.2 Pandas 27
3.3.3 Scikit Learn 28
3.4 Operating System chosen for this project 29
3.5 Web Browser chosen for the project 29

3.6 Summary **30**

4. Methodology ***30***

4.1 Introduction **30**

4.2 Method: Crisp DM Methodology **30**

4.3 Conclusion **33**

5. Data Analysis ***34***

5.1 Introduction **34**

5.2 Initial data exploration **35**

5.3 Long short-term Memory (LSTM) **42**

5.3.1 Data preparation 42

5.3.2 Data normalization 43

5.3.3 Data split 44

5.3.4 Model building 45

5.3.5 Model Training 47

5.4 Auto Aggressive Moving Average (ARIMA) **47**

5.4.1 Data Preparation 47

5.4.2 Data Split 47

5.4.3 Model Building 48

6.0 Results and Discussions ***50***

6.1 Model Comparison **50**

7.0 Conclusion ***55***

8.0 References ***57***

Appendices ***62***

FYP Poster **62**

Project Log Sheets **63**

Project Log Sheet 1 63

Project Log sheet 2 64

Project Log Sheet 3 65

Project Log sheet 4 66

Project Log sheet 5 67

Project Log sheet 6 68

Project Log sheet 7 69

Project proposal Form (PPF) **70**

Project Specification Form (PSF) **71**

Fast Track Form **74**

Gannt Chart For Final Year Rpeort **78**

1. Introduction to the Study

1.1 Background to the project

In the realm of payment money, cryptocurrency is a digital payment money that does not need to be documented or maintained on file by any bank anywhere in the world. Because cryptocurrency transactions are not dependent on any bank anywhere in the world, they are completely immune to the reach of any government authority. When it comes to receiving or sending crypto, they depend on a peer-to-peer system that can be set up almost anywhere. Cryptocurrency operates on a distributed ledger, which is made accessible to the public under the name blockchain. It is produced by a process known as mining, which includes the use of a computer to perform basic mathematical problems that result in the production of coins. Individuals may also purchase it from third-party websites and brokers. If you hold a cryptocurrency, you don't own anything substantial; instead, you own a key that allows you to unlock and transfer a certain unit of measure from one location to another without the involvement of a third party. There are several cryptocurrencies available, some of which are Bitcoin, Ethereum, Litecoin, and Ripple.

The Bitcoin was invented in 2008 by Satoshi Nakamoto, and it was first made available to the public in 2009, according to the Bitcoin website. It will take around one day to generate a single bitcoin, depending on the hardware of the user's computer. It may also take approximately thirty days to mine a single bitcoin. Bitcoin is an open-source project, and its design is available to the public; no one owns or controls it (Bitcoin, 2009).

According to Reuben Grinberg, 50 bitcoins are released every 50 minutes, resulting in the creation of 10.5 million bitcoins over the next four years. Bitcoins have a decimal place count of eight decimal places. The development of a bitcoin eco-system will involve large-scale transactions, online merchants, market information, and many bitcoin miners. In today's world, many online retailers accept bitcoin as payment, which increases the popularity of bitcoin (Grinberg, 2012)

Machine Learning is a type of artificial intelligence that automatically improves the user experience via user data. It uses the historical data to as input and predict the output as new values. Machine learning is very important these days because it gives an enterprise a view of customers and the current trends (Burns, 2019).

Machine learning can be used to predict the price of bitcoin through many machine learning frameworks such as tree-based model, Long-Term-Short-Term (LSTM) and many more models which can be used. LSTM is form of a recurrent model and a neural network capable of learning a long team-dependency. It is used by update gates and forget gates to predict the future outcome. LSTM is a network which is capable of learning order dependencies in a sequential problem of prediction, these are commonly found in speech recognition software, machine learning software’s and many more similar systems (Brownlee, 2017).

1.2 Problem statement

Bitcoin is the most complex among all the cryptocurrency which the value change from time to time. Investing money in Bitcoin is risky and people make less profit from it. Bitcoin is more volatile because the supply and demand the Bitcoin prices increase and decrease depending on how many bitcoins are circulating and how many people are ready to pay. Bitcoin also depends on its investors; the bitcoin prices will increase based on the number of bitcoins circulating. According to the report by National Bureau of Economic growth, almost one third of the bitcoins were held by the around top 10,000 investors. Bitcoin is not stable always it will always be fluctuating, and it is always risky for the investors to invest if the prediction of bitcoins is not accurate.

Volatility is one of the markers of the digital financial market, and bitcoins are the most volatile of all the cryptocurrencies currently available on the market, according to CoinMarketCap. According to research by Keyede Erinfolami, Dogecoin's value increased by 20,000 percent in the first quarter of 2021, but it began to decline towards the end of the year. In the near future, the future of cryptocurrency might be either positive or negative. It is dependent on the investors and buyers' ability to quickly sell and acquire bitcoins, which causes the price of bitcoin to fluctuate between drops and increases from time to time (Erinfolami, 2021).

One of the major issues with the price predation is that the developer will not have enough analytical support to show the predicted output of the bitcoin. If the predicted price of the bitcoin shows an upward trend the investors will be tempted to invest in the bitcoin, but the predicted price is not 100% accurate (Reiff, 2019). Bitcoin Volatile Index (BVI) is a metric system that monitors how the bitcoin deviates from its mean as its price, it calculated by the standard deviation which is calculated by the square root of the variance determined by the relative to the mean.

1.3 Rationale

Making educated guesses about the price of bitcoin in the next years is a difficult endeavour. The most significant problem is that the developer will not have any documentation to back up the prices that have been forecasted. The price prediction will have an impact on a large number of investors and purchasers. Given the unpredictability of Bitcoins and their behaviour, developers can only forecast the correct value after a specific number of days, but not the actual price of the cryptocurrency.

1.4 Potential Benefits

Intangible benefits and tangible benefits are the two sorts of potential benefits that might be derived from an operation. Tangible benefits are those that can be measured and calculated, which are referred to as hard

benefits, while intangible benefits are those that cannot be measured or calculated, which are referred to as soft benefits.

1.4.1 Tangible Benefits

- Investors and buyers will have clear price of the bitcoin after certain number of arbitrary days.
- To increase the investors and buyers satisfaction in price prediction of bitcoin.

1.4.2 Intangible Benefits

- To increase the people's benefit in the field of cryptocurrency.
- To increase the people invest in the bitcoin investment and other cryptocurrency.

1.5 Target users

The Bitcoin Price Prediction tool, which makes use of machine learning, will be able to assist in predicting what the price of bitcoin will be after a given number of days have passed. Therefore, the people who invest in bitcoin and other cryptocurrencies are the ones who will benefit the most from this project.

1.6 Objective and Scope of the project

1.6.1 Aim

To predict the price of the Bitcoin cryptocurrency based on several days given using machine learning techniques to help investors, buyers and sellers better know the market trends.

1.6.2 Objectives

- To ensure proper data cleaning of the data set to get more accurate results.
- To extract a proper data set which is suitable for the project.
- To use appropriate Machine Learning Algorithms and build a suitable model for the Data set obtained.
- To predict the price of the bitcoin after an arbitrary number of days.

1.6.3 Deliverables

Deliver a clear observation of bitcoin prices after certain number of arbitrary days. LSTM and certain regression model will be applied to get the accurate results.

- A clear comparison of the predicted price compared with the previous price.
- A clear understanding for the investors and buyers of the predicted bitcoin price.

1.7 Nature of Challenge

The most difficult difficulty is determining the current market trend for cryptocurrencies, which is not always consistent from day to day. This is mostly dependent on the number of investors and purchasers, as well as the quantity of bitcoins that are freely circulating on the internet. Obtaining a sufficient and appropriate data collection for this project is also one of the most difficult tasks to do. Obtaining datasets is simple and can be done via any website like as Kaggle, although there are many different sorts of datasets to pick from, making it more difficult to narrow down your options. In order to choose the best data set, the developer must take into account a variety of parameters that are dependent on the data set, including the timestamp, the price when the data set is opened, the maximum and lowest bitcoin prices, and the price of bitcoin when the data set is closed.

The developer must also take into consideration the machine learning methods that are accessible, and there are several data models that may be used. Many machine learning approaches may be applied to a given data set in order to forecast the price, but the projected price will change depending on which machine learning technique is used to analyse the data set in the first place. Because bitcoins are so unpredictable, the developer is also unable to identify and maintain the documentation that has been generated. In addition, data cleansing for the data set is one of the accomplished tasks that must be taken into consideration. Because the price is not the region that should be updated, the bitcoin data that was collected initially from the source cannot be changed or amended after it has been obtained. This will have a negative impact on the estimated pricing, which will result in a number of difficulties for the developer after the price is predicted.

According to Hamayal and Owda they have stated in their journal that the prices predicted by the people with logistic regression and linear discriminant techniques are only 66% accurate (Hamayel & Owda, 2021). There will also be a hurdle in evaluating and finishing the documentation that was generated by the developer since there are many components in the documentation that are sophisticated and need more time and effort to be applied.

1.8 Project Plan

The time required to accomplish this project will be shown in this section of the chapter. The Gant chart will show the chronology and the evolution of the tasks, which will be set out in detail and shown in a straightforward manner.

Investigation Report (IR) Project Plan				
Task	**No. of Days**	**Start date**	**End date**	**Progress/Status**
Project Proposal Form (PPF)	4	14/02/2022	18/02/2022	Done
Project Specification Form (PSF)	9	19/02/2022	28/02/2022	Done
Chapter 1: Introduction to the study	24	01/03/2022	25/03/2022	Done
Chapter 2: Literature Review	35	26/03/2022	30/04/2022	Done
Chapter 3: Technical Research	6	01/05/2022	07/05/2022	Done
Chapter 4: Methodology	11	08/05/2022	19/05/2022	Done
Chapter 5: Conclusion	9	20/05/2022	29/05/2022	Done

2. Literature Review

2.1 Introduction

This area of the chapter will the details of the literature review that has been done for this project and the domain research conducted for this project. The algorithms and the similar systems will also be mentioned this section of the project.

2.2 Domain Research

2.2.1 Bitcoin

The bitcoin network is a group of computers (Nodes) that operate on code that was produced by the bitcoin's creator and is based on the blockchain technology. Blockchain is also referred to as a collection of blocks in certain circles. There is no way for anybody to cheat in the network since all of the machines in the system that are up and running have the same list of blocks, and all transactions are transparent because everyone can see the blocks that are being filled up, which means that nobody can cheat in the network.
According to Jake Frankenfield Bitcoins have 13,768 full nodes in the network as of mid-November 2021 which means any person who wants to attack the network needs to computer more than 51% of the bitcoin’s total computer power network to attack which is highly impossible now. Bitcoin miners, on the other hand, would most certainly branch out to construct a different blockchain if an attack were to take place, therefore making the bad actor's endeavour worthless (Frankenfield, 2021).
The subject of digital currencies has continued to be discussed in the media on a regular basis. Approximately one billion dollars' worth of stock has been bought by Warren Buffett's Berkshire Hathaway, which will be invested in a cryptocurrency-focused digital bank. Since December 2020, Bitcoin was market-disrupting by many financial experts and has warned to be dangerous. The issue with the bitcoin pricing is that it is not under printed in terms of inherent worth in any way. Mark Northway, an

investment manager at sparrow capital, has said that the price of bitcoin is controlled by one aspect of a person's character, namely, their level of faith in the market (Denham & Otte, 2022).

2.2.2 Machine Learning

Machine learning can be defined as using computational methods using the user experience of the performance to make accurate prediction. Here the experience means the previous or the past data from the user, which typically can be found and be available for data analysis. Machine learning is designed efficient and accurate algorithms. Machine learning application will make the user experience more easy and make many complex algorithms for the developer easy (Mohri et al., 2018).

Machine learning has a wide range of skills, and it is no longer believed to be science fiction. It is frequently employed in business and in all of the industries that are now accessible on the market. In order to enhance and solve numerous hard issues in the company, 41 percent of all organisations have adopted and accelerated the usage of Machine Learning and Artificial Intelligence (AI) in their organisation. Machine learning has already been used by 31 percent of the new companies (Selig, 2020).

Machine learning is used in many sectors like Data security, Finance, Healthcare, Fraud detection and retail. Machine learning offers many benefits in the application of AI, it provides machine learning methods such as supervised learning, unsupervised learning and semi-supervised learning.

2.2.3 Regression Analysis

It is a statical modelling, regression analysis is a set of statical values for estimating the relationships among the dependent variable and one more independent variable. Its formula is $\boldsymbol{Y_i = f(X_i, \beta) + e_i}$

where

- Y_i = $dependent\ variable$
- $f = function$
- $X_i = independentvariable$
- $\beta = unknown\ parameter$
- $e_i = errorterms$

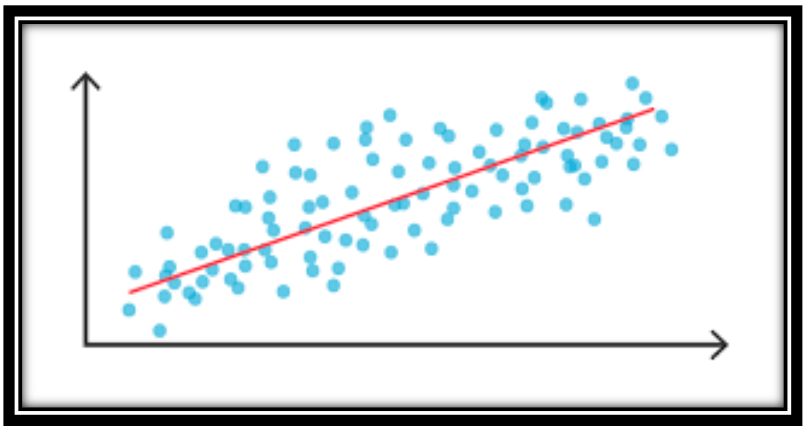
or to keep it in simple terms the formulae can be written as $\boldsymbol{Y = Mx + B}$ where

- $Y = dependent\ variable$
- $M = the\ slope\ of\ the\ equation$
- $x = dependent\ variable\ of\ the\ equation$
- $B = constant$

(Thakur, 2019)

While running the regression analysis the purpose of the developer is to find out the relationship between the dependent variable and the independent variable. So that the developer can predict the dependent variable using the one more independent variable which are chosen that can help to find the dependent variable. The regression analysis will the developer to validate the predictor values are good enough to help in predicting the dependent variables.

A regression analysis graph will look like

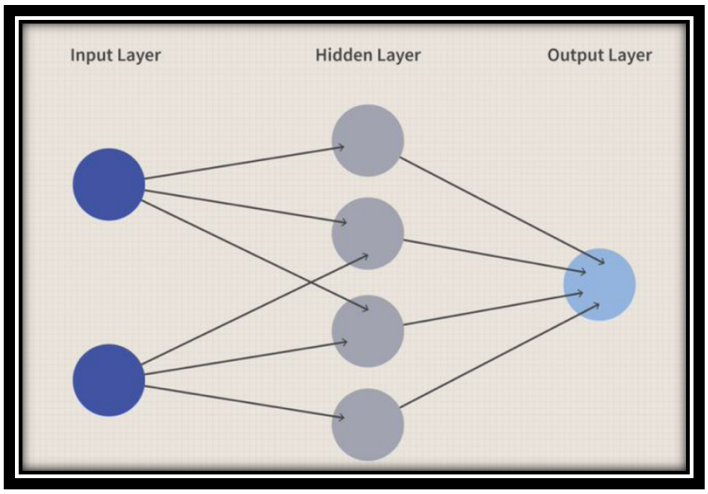


Regression Analysis

where the line is the dependent variable and the blue dots are the independent variables.

2.2.4 Neural Networks

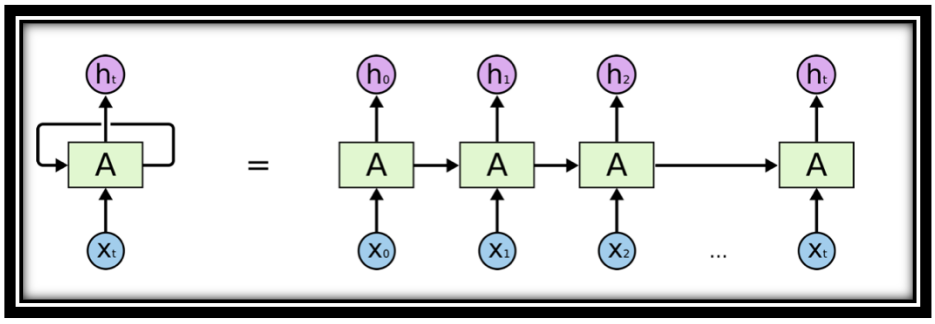
A neural network is a sequence of algorithms that attempts to detect and recognise the underlying connection in a given data set by using the same technique on how the human brain works in real life. This process is analogous to how the human brain operates in the actual world. When presented with a new input in order to derive a new result, neural networks are able to adjust and accommodate for the change. Neuronal networks are any system of neurons, whether they biological or artificial in origin. Neural networks may also refer to neural circuits.



A Simple Neural Network

A neural network is made up of several hidden layers of linked nodes, as illustrated in the image to the right. Each node in the network is referred to as a perceptron, and the network as a whole functions as a multiple linear regression. The signal is first transformed into a multiple linear regression using the perceptron, and then it is finally transformed into a nonlinear activation function.

The components of multi-layered perception (MLP) are placed systematically in layers that are related to one another. All of the information is gathered at the input layer, which also contains a great deal of classification and a great deal of signals that may map any pattern. The hidden layers will continue to fine-tune the input weights until the margin of error in the neural network is as small as possible. Hidden layers are supposed to identify important elements from the input data that have predictive potential for the outputs. This is done via the use of hidden regression models. This is how the process of feature extraction works, and it is analogous to the way that statistical methods such as principal component analysis function. The three fundamental building blocks consist of an input layer, a processing layer, and an output layer. It's possible to weight the inputs using a variety of different criteria. There are nodes and connections between these nodes that are supposed to be analogous to neurons and synapses in an animal brain within the processing layer, which is disguised from view. These nodes and connections may be found inside the layer. A kind of neural network known as a recurrent neural network is one that was developed specifically for the purpose of analysing time series data, event history, or temporal ordering.

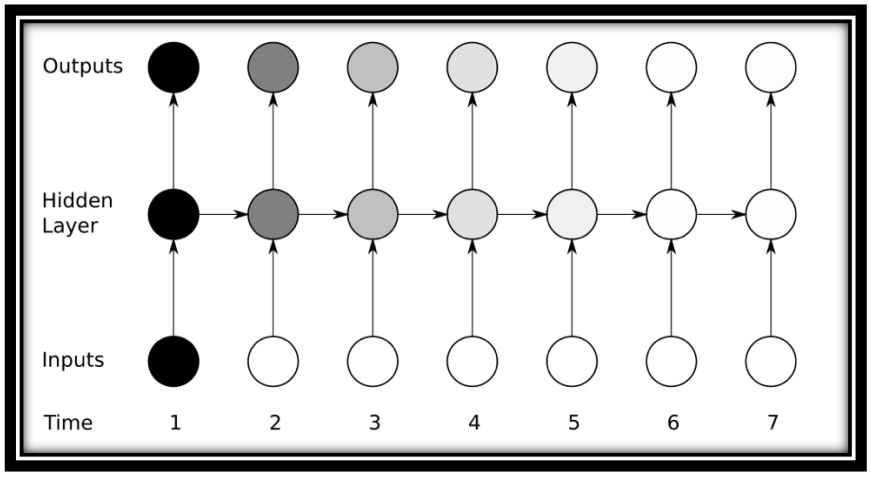


Recurrent Network

2.2.5 LSTM (Long term-short Memory)

Long term-short Memory (LSTM) is used in the area of deep learning; it has a variety of recurrent neural networks (RNN) that are able to carry out learning of long-term dependencies, such as sequence or number prediction. These sets of recurrent neural networks are also known as memory blocks, and they make up the architecture of the LSTM. These memory blocks may be thought of as a differentiated version of the memory chips seen in digital computers. Every block has at least one and maybe more than one self-connected memory cell as well as three multiplication units. These units are known as the input, the output,

and the forget gates, and they provide the cell blocks with constant analogues of the read, write, and reset operations.



RNN (LSMT)

The colouring of the nodes in the unfolded network illustrates the degree to which they are responsive to the inputs at time one (the darker the shade, the greater the sensitivity). The network 'forgets' the original inputs as time goes on, which causes the sensitivity to decline. This happens because new inputs are constantly overwriting the activations of the hidden layer (Graves, 2012).

It is possible to combine LSTM blocks with ordinary summation units, despite the fact that this is not often necessary. It is possible to use the same output layers for LSTM networks that are used for conventional RNNs. It is possible to combine LSTM blocks with ordinary summation units, despite the fact that this is not often necessary. It is possible to use the same output layers for LSTM networks that are used for conventional RNNs.

Analysis of time series and prediction have traditionally been useful methods in a wide range of practical applications, including weather forecasting, transportation planning, and traffic management. In the field of telecommunications, intelligent techniques have been developed in the past in order to monitor and analyse a huge range of time-dependent events. Some examples of these time-dependent events include data flow, user location, channel load, and service requests.

The prediction goal of the time series data like bitcoin $y1, y2, y3, \ldots . yN$ is to estimate the i based on the previous data $yi-1, yi-2, \ldots \ldots yi-n$ then we can denote $x = \{y(i-k), y(i-k+1) \ldots . y(i-1), i = \{k, \ldots . n\}$ here the goal is to find the function $f(x)$ so that the $\hat{Y}i = f(x)$ is same as the yi (Hua & Zhao, 2019, pp. 114–119).

2.2.6 Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a statical analysis model that uses time series data to either understand the data or predict the future trends. ARIMA models provide an alternative viewpoint on the process of time series forecasting.

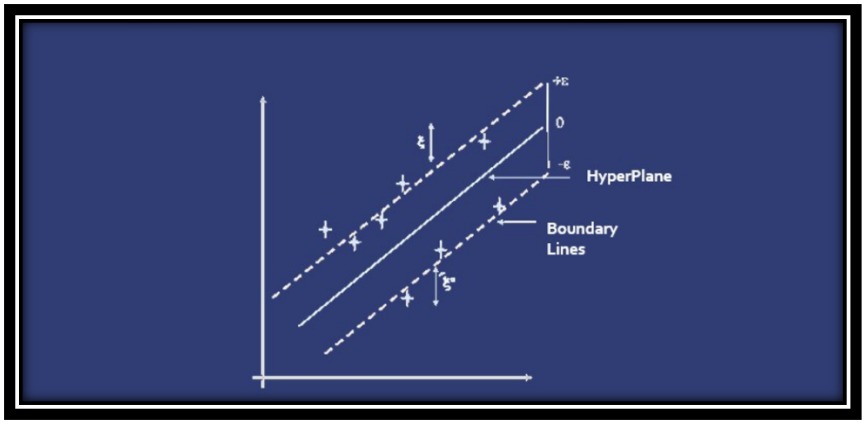
Exponential smoothing and ARIMA models, which take two very different approaches to the problem, are the two methods of time series forecasting that are used the most often. ARIMA models aim to characterise the data's autocorrelations, while exponential smoothing approaches attempt to explain the trend and seasonality of the data (J Hyndman & Athanasopoulos, 2018).

Predicting the price of a stock is a significant subject in the fields of finance and economics, and academics have been hard at work for years developing more accurate forecasting models. In the research that has been done, the autoregressive integrated moving average (ARIMA) models have been investigated for use in time series prediction. In order to construct an all-encompassing model for predicting stock prices, this research makes use of the ARIMA model. The New York Stock Exchange (NYSE) and the Nigeria Stock Exchange are both sources of public stock data that are used in the process of making stock price forecasts (NSE). The findings indicated that the ARIMA model has a lot of potential for short-term forecasting and is capable of competing with other approaches to predicting stock prices already in use (A. Ariyo, 2015).

2.2.7 Support Vector Regression (SVR)

Support Vector Machines are well-known for their capacity to solve a wide variety of classification problems, which has contributed to their widespread adoption. It is a potent algorithm that provides the developer with the ability to pick the tolerant of how we are the errors by both accepting error margin and by turning the tolerance of slipping outside that are acceptable error() rate. This is made possible by the fact that the algorithm has the ability to pick the tolerant of how we are the errors. Because of this, the developer has a great deal of flexibility (Sharp, 2020).

SVR is the strategy that is used whenever an issue is solved by applying a linear regression as the approach to take. In contrast to linear regression, structural vector regression (SVR) gives the developer the ability to use it not just to linear but also to non-linear connections. This is an advantage over linear regression. By making adjustments to the model's hyperparameters, the developer will have increased freedom to alter the robustness of the model. SVM is a popular algorithm for predicting and building trading systems. It is used mostly to predict stock, index price movement wheatear it will go up or down in future (Ślepaczuk & Zenkova, 2019)



Support Vector Regression (SVR)

2.3 Similar Systems

In this research the developer will discuss the past similar system to this project because this section focus the accuracy of the project. Here the developer will explain briefly on the similar system found and also compare them.

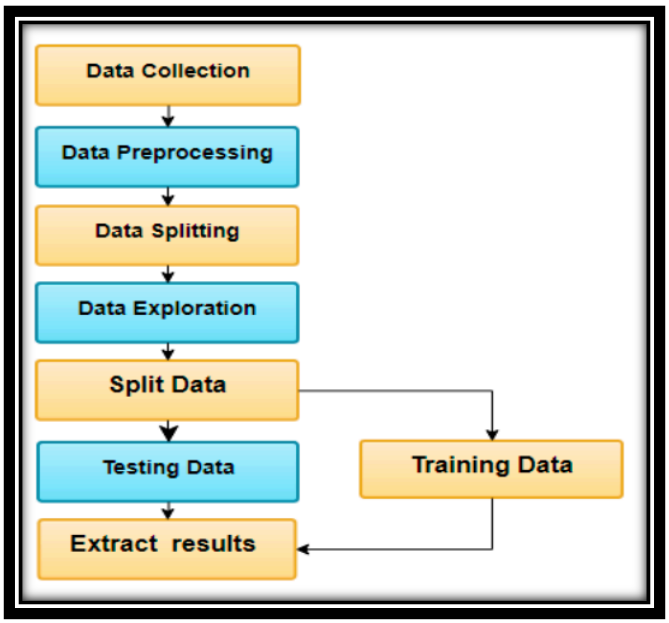
2.3.1 Prediction of bitcoin by Anuradha Datta

A study that was conducted by Anuradha Datta in the year 2020 regarding the price prediction of bitcoin reveals that she has compared the given types of algorithms, which are LSTM and AIRMA, both of which are used in any prediction model. The research was conducted in relation to the price prediction of bitcoin. She has arrived at the conclusion that the LSTM model is superior for use in her project. bitcoincharts.com was consulted in order for her to get the Bitcoin price time series data. In addition to this, she has highlighted that the LSTM model is a subtype of Recurrent Networks (RNN), which are networks that are purposely designed for long-term dependencies. A Gated recurrent unit (GRU) also was introduced in the paper which solves the problem of vanishing gradient of the RNN. She utilise two different trading strategies in order to evaluate the results of the portfolio management and precisely foresee the collapse. One of these strategies involves the trader selling Bitcoins in order to conserve cash before cryptocurrencies. For the purpose of clarity, we limited our analysis to the trading of Bitcoin and made the assumption that investment would pick back up once the price started to increase again in the middle of June. On the other hand, in order to engage in short selling, a trader must only make purchases and sales in accordance with the signals produced by quantitative algorithms. With very modest growth in portfolio value based on our test holding long positions simultaneously, the long-short strategy suffered throughout the same length of time set assessment. It is noticed that regarded the GRU one layer with recurrent dropout as our best model for with very modest growth in portfolio value. On the other hand, long-short systems have the potential to be more productive when it comes to putting trading strategies into action. Two distinct trading approaches were

used in the case of a portfolio that included a number of different cryptocurrencies and in which investors may participate in the activity.

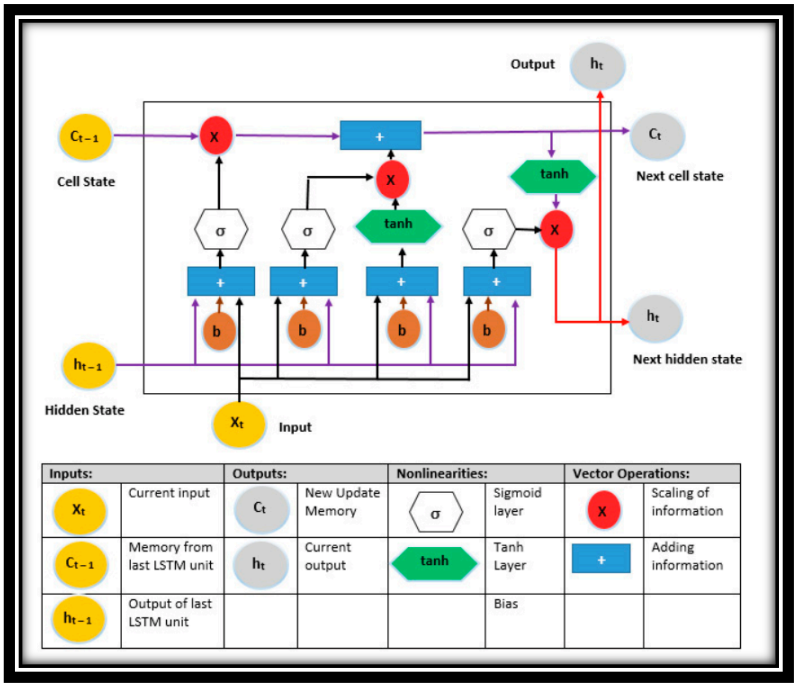
2.3.2 Prediction of cryptocurrencies by J. Hamayel and Y. Owada

In a different research study done by J.Hamayel and Y.Owada, the authors trained the data set into three different models, tested and trained those models to produce three distinct results, and then compared those results. This method and these materials are described in the paper by J.Hamayel and Y.Owada. In this paper, the authors collected information about cryptocurrencies first, and then performed data visualization and exploration.



Proposed system by J.Hamayel and Y.Owada

The LSTM algorithm was chosen because of involving various issue of sequential data, recurrent neural networks where the LSTM will act as effective and scalable approach. Due to the fact that they are unspecific and straightforward, they are helpful for capturing long-term temporal relationships. As a consequence of this, it is not apparent if the LSTM is the ideal design, and it is conceivable that more effective designs may be developed in the future. The RNN-style design of the LSTM has gates that are responsible for controlling the flow of superior ones that are available. The information that is carried along the cell state may be modified by the input and forget gate structures, and the final output is a filtered version of the cell state that is based on the context that was provided by the inputs. The design of the LSTM has been questioned on the grounds that it is ad hoc and that it has a large number of components the function of which is not immediately apparent. As a consequence of this, it is not possible to determine if the LSTM is the best design, and it is very probable that other designs are superior. The below figure shows the structure if the LSTM model.



Structure of LSTM

$$\boldsymbol{f_t = \sigma W_f \cdot [h_{t\_1}, x_t] + b_f (1)}$$

$$\boldsymbol{i_t = \sigma W_f \cdot [h_{t\_1}, x_t] + b_i (2)}$$

$$\boldsymbol{C_t = f_t * C_{t\_1} + i_t * tanh(W_c \cdot [h_{t\_1}, x_t] + b_c)(3) O_t = \sigma(W_0 \cdot [h_{t\_1}, x_t] + b_0)(4) h_t = o_t * tabh(C_t)}$$

The forward training process used by J.Hamayel and Y.Owada is formulated above where i_t , o_t , and f_t denotes the input output and the front gate respectively. The c_t and h_t denotes the activation vector of each single cell in the memory block, the W and b denotes the weight of the of the matrix and bias vector. $\sigma(°)$ denotes the sigmoid function.

The findings acquired by J.Hamayel and Y.Owada using long short-term memory (LSTM), gated recurrent unit (GRU), and bidirectional long short-term memory (bi-LSTM) algorithms employing three kinds of popular cryptocurrency: Bitcoin (BTC), Ethereum (ETH), and Litecoin (LTC). The findings of each model are presented using three prominent cryptocurrencies: bitcoin, ether, and Litecoin. It is generally agreed that the "best" model is the one that produces the lowest RMSE and MAPE values. On the basis of these requirements, any model that can be applied to one of the three kinds of currencies may be considered a model. According to these standards, all of the models that were used to analyse the three different kinds of currencies may be regarded as excellent models; nevertheless, the GRU was discovered to be the most successful of the three. The RMSE thought the models were excellent, but it was determined that the GRU was the most accurate of the three. The RMSE of the GRU model is the smallest of all the models. As a result, GRU has a greater capacity for long-term prediction, and its model has the lowest accuracy. As a result, when compared to LSTM and bi-LSTM, GRU is superior in its ability to forecast long-term

relationships and dependencies. The reason for this is because, in contrast to LSTM and bi-LSTM, it is dependent on previous envies. This is because current prices are dependent on previous pricing.

2.3.3 Bitcoin price prediction using linear regression by V. Ramesh and Alvin Ho.

In another paper by V. Ramesh, Alvin Ho, they have used liner regression and LSTM for the bitcoin price prediction. They have also given detailed explanations of the limitations, methodology chosen, in the project. The purpose of their research is to provide an explanation of how the linear regression model and the Long Short-Term Memory model work together to forecast the value of a Bitcoin. Bitcoin's ever-increasing popularity has transformed it into something like to an investment. Bitcoin also utilises the block chain technology, which paved the way for the development of other cryptocurrencies. Because of this, making an accurate prediction of its value is quite challenging; as a result, this predictor is evaluated with the use of a machine learning algorithm and an artificial neural network model. Methodology: In the research, they tested and trained the machine learning and artificial intelligence model using data sets pertaining to Bitcoin. The data filtration procedure was successfully completed with the assistance of python libraries. Python has the finest functionality for data analysis and visualisation thanks to what Python has supplied. After they have gained a knowledge of the data, they have prune the data and utilise the features or characteristics that are most appropriate for the model. The model's implementation has been completed, and the outcome has been documented. Finding: It was found that the linear regression model's accuracy rate is very high when compared to other Machine Learning models from related works; it was found to be 99.87 percent accurate. This was discovered after comparing the linear regression model's accuracy rate to those of other Machine Learning models. On the other hand, the LSTM model has a minor error rate that is just 0.08 percent. The conclusion that can be drawn from this is that the neural network model is more effectively optimised than the machine learning model. The novelty of this work consists on the creation of a simple graphical user interface (GUI) with the help of the *tkinter* library. This GUI will enable the user to enter the values for the High, Low, and Open features, and it will then forecast the next value for the coin. The results of predictions made by a machine learning model and an artificial neural network model are compared and contrasted in this research. We picked the linear regression model to compare it to the LSTM model since it delivered the best accuracy compared to the other machine learning models that we looked at.

2.3.4 Differentiation of similar systems.

This section briefly compares all the similar system,

System	Anuradha's System	J. Hamayel and Y. Owada system.	V. Ramesh and Alvin Ho's system
Abstract	In this article, she analyse a framework for predicting daily Bitcoin values that includes a collection of sophisticated machine learning forecasting algorithms together with a predefined set of exogenous and endogenous parameters.	This article presents three different kinds of recurrent neural network (RNN) algorithms that may be used to forecast the values of three different cryptocurrencies. These cryptocurrencies are Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH). Taking into account the mean absolute percentage error, the models provide quite accurate forecasts (MAPE).	In this study, the linear regression and Long-Short-Term Memory models that are used to estimate the value of a Bitcoin are broken down and explained in detail.
Machine learning Algorithms used in the project.	The author of this research has made use of LSTM and also includes a comparison of the findings with ARIMA. Python was the computer programming language that was used for this study.	The sole technique that the developers here employed was the LSTM algorithm, and they validated the correctness of the outcome by separating the data set into training and validation halves.	In this section, the developers have attempted to forecast the price of bitcoin using linear regression, and then compared their findings to those obtained using the LSTM technique.
Results obtained	Her research led her to the conclusion that recurrent neural network models, such as LSTM and GRU, are	According to the findings that were acquired from these models, the performance of the	According to the findings of the research, Long-Term Short-Term Memory displays a higher accuracy rate

	superior to conventional machine learning models. Neural networks such as LSTM and GRU may efficiently learn from non-linear patterns despite having insufficient input because of their ability to control previously learned information. In contrast to more traditional methods of dealing with time series data, deep learning models need for precise training and tweaking of their hyperparameters before they can provide useful results. This might take a significant amount of processing resources when dealing with huge datasets. However, when it comes to predicting the price of stocks or cryptocurrencies, the market data are always restricted, and the computational complexity is not an issue; as a result,	gated recurrent unit (GRU) was superior to that of the long short-term memory (LSTM) and bidirectional LSTM (bi-LSTM) models when it came to the prediction of all forms of cryptocurrencies. As a result, it is possible to regard it as the most effective algorithm. The GRU model provides the most accurate forecast for LTC, with MAPE percentages of 0.2454 percent, 0.8267 percent, and 0.2116 percent, respectively, for BTC, ETH, and LTC. When compared with the other two algorithms, the bi-LSTM method produces the worst prediction result. This is seen by the MAPE percentages, which are as follows: 5.990 percent for Bitcoin, 6.85 percent for Ethereum, and 2.332 percent for Litecoin.	than Linear Regression does. Due to the fact that this research only compares four different characteristics—open, closed, high, and low—the findings may be different if we were to take into account a wider range of characteristics. Data sets cannot serve as the only basis for predicting since the cryptocurrency market is very volatile and is subject to the effect of social media as well as other external variables. The continued development of technology will allow for the collection, examination, and use of fresh data, which will ultimately lead to improved findings from this experiment.

	shallow learning models may be efficiently applied in reality. It is anticipated that these advantages will make a major contribution to quantitative finance during the next several years.		

3. Technical research

3.1 Introduction

The developer will detail the technical research that was done in order to complete this project in this portion of the chapter. This section will cover the programming languages that are available for the project, as well as which programming language should be used. The section will cover the libraries that are relevant to this project and how they may be used. It will also be explained in this part what sort of operating system has been selected, as well as which web browser has been used for this project. This section will go through in depth all of the materials that will be necessary for this project.

3.2 Programming Language Chosen

The selection of a programming language is also one of the most difficult difficulties for a developer to deal with. This is due to the fact that the developer must pick the most appropriate programming language for the project's requirements. A programming language will have a large number of built-in libraries to depend on. Because this project involves machine learning and massive data sets, Python, R and SAS will be the most popular programming languages on the market. In this project Python will be used due to many numbers of libraries are compatible, simplicity of the code syntax and many deep learning applications relays on this programming language.

3.2.1 Comparative analysis on Python, R programming language and SAS

In this section the developer will show the comparative analysis of Python, R and SAS programming language. The developer will highlight all the features and requirements of the programming languages.

Factors	Python	R	SAS
Outline	It is high level, general purpose and objected oriented programming language.	R is a lightweight, and statical programming language.	SAS stands for statical analysis system which is used analyse system data and visualize.
Simplicity	Python is one of the simple programming languages and easy to learn if put some hard work.	R is also one of the easy programming languages to master by it requires more time to understand and apply.	SAS is easy for the people who already have some experience in SQL. It will take around 6 to 8 months to master the programming language.
CPU and Ram Usage.	Python is an open source which will not consume more ram. But the ram usage will differ based on the libraries downloaded.	R requires a separate environment to run which automatically increase the Ram usage.	SAS is a heavy programming language and requires more Ram consumed compared to Python and R.
Minimum Requirements	4GB of Ram and 5 GB of disk space.	4GB of Ram and 5GB of disk space.	16GB of Ram and 256GB of disk space.
Deep Learning/Machine Learning.	Python has many deep leaning libraires which can be imported and be used for many deep learning techniques. TensorFlow, Numpy and many more.	R also has many packages that can do deep leaning. kerasR and keras are some of the packages that's needs to be downloaded.	SAS has the capability to apply deep learning and machine learning techniques that can be used, but not all the steps required be done on SAS.
Speed of executing	Python doesn't need any IDE or 3rd party software to write its code, it can be written in the system's Terminal and executed. It won't	R does require an 3rd party software to write and execute its code. It has some delays and require more time to	SAS requires more execution time and requires installing a .jnlp file every time we run or open application from SAS community.

	take much time to execute its code because it executes independently.	complete executing the code.	
Cost/Resource	All the programming languages are open source, which means they are accessible to anybody and any organization that wants to use them.		
Compatibility	All the programming languages are compatible with MacOS, Windows, Linux and all the available operating system in the market.		
Data Control/Handling	All the programming languages are capable of handling large data sets.		

According to the results of the preceding comparative research, Python is the programming language of choice since it is preferred by many developers in the data science sector and is extensively employed in the market in the deep learning and machine learning fields and industry.

3.2.2 Python

Python is a high-level, general-purpose, and object-oriented programming language that is utilized in practically every subject of education, including science, technology, engineering, and mathematics. Python is a free and open-source programming language that is freely given by python.org for any purpose, whether purely induvial or commercial in nature. Python is also one of the most straightforward programming languages to learn when compared to other commonly used programming languages in the business, such as C++, C, Java, and so on. It features a straightforward and appealing syntax that is simple to grasp for even the most inexperienced user. Python applications may also operate on a variety of platforms without the need to change their code; the compatibility of python is straightforward.

Python

Python was introduced by Guido van Rossum in 1990 as a Programming Language. In order to achieve his goals and objectives, he built a programming language that is simple to learn and powerful to use. It should be open source and accessible to everyone, with code that can be explained and understood as easily as English, and it should be utilized for daily activities. Python also as a capability to handle very large data sets. Python has risen to become the second most popular programming language in the world, and the first most popular programming language in the area of data science, 20 years after its first release (Python Institute, 2020)

Python enables developers to become more confident and productive while producing software, and it is also used to design, maintain, and upgrade numerous machine learning algorithms, among other things. Python is becoming increasingly acceptable in the area of data science due to the availability of sophisticated artificial intelligence and machine learning tools. In Python, a large variety of libraries are accessible, including Sckit-Learn, Pandas, Keras, and Tensor Flow. As a result, Python is the top programming language for machine learning domains in the industry. Aside from that, it provides a variety of visualizing choices and libraries that enable developers to represent human data into graphs using machine learning frameworks. It is also compatible with all major operating systems. When a developer writes code for a Windows operating system, that code may be utilized in other operating systems such as Mac OS, Linux and many more, allowing the developer to test and use the code on all the platforms now available in the industry (Kumar, 2021).

A number of emerging technologies such as artificial intelligence, deep learning, and machine learning are having a big impact on the current learning industry. Every day, new issues are encountered, and new solutions are generated via the use of Artificial Intelligence, Machine Learning, and Deep Learning approaches to address these challenges. Even if learning Python is straightforward, mastering it will take some dedication on the side of the developer. It is clear that Python is the programming language of choice for artificial intelligence, machine learning, deep learning applications, and data science given all of the advantages it offers.

3.3 Integrated Development Environment (IDE)

An integrated development environment (IDE) is software that allows developers to create code, debug it, compile it, and run it all at the same time, making it easier for them to interact with the code. IDEs (Integrated Development Environments) are created and developed to make the lives of developers who write code easier (Nyakundi, 2021).

Developers utilise an integrated development environment (IDE) because of its extensive capabilities and user-friendly User Interface (UI), which makes the code seem more standardised and clean for the developer. It also assists the developer in maintaining the flow of programming and navigating through any line of code if the code is lengthy. An integrated development environment (IDE) has various capabilities,

some of which will assist developers in organising their workflow and solving difficulties. Because IDEs are made up of Phrase codes, any errors will be due to human mistake exclusively. This is due to the fact that all of the functionality will be displayed visually on a single screen, which makes it easier for the developer to work with.

An IDE should typically consists of:

- **Code editor**

  A text editor which allows the developer to write and update the code, with the help of auto-complete, syntax highlighting and check for various big and small bugs in the code.
- **Build Automator**

  This is utility which comes with all the available IDE's in the market which automatically creates a local build, create binary file, packaging of codes and running refreshed codes.
- **Debugger**

  A feature that will help the developer to identify and graphically displays the exact location of the bug in the programme.

There are several well-known integrated development environments (IDEs) in the business, including Microsoft Visual Studio Code, Microsoft Visual Studio, Apache NetBeans, PyCharm, and others. Nonetheless, the IDE selected for this project is Google Collab, and it is a browser-based IDE that enables for the storage, compilation, execution, and debugging of code using could-based technology.

3.3.1 Google Collab

Collab is a free notebook environment which runs entirely on cloud and lets the developer to write and debug the code like google documents. As it is entirely stored in cloud the developer can access the code anywhere with his/her google account.

Collab

It is a total free of charge and also can be combined with Jupiter notebook. It gives the developer of 16GB ram and 250GB to build and develop code. All the code written in it will all get stored in the developer google dive account, and also it auto-saves every time the code is edited.

3.3.2 Jupyter Notebook

Google Collab is combined with Jupyter notebook and it acts as a notebook environment to perform many data analysis codes and visualization of graphs and data.

Jupyter

We don't have to download any packages or libraries into our PC every time we load or open a new notebook in Google Collab, which saves us a lot of storage space and RAM on our computer.

A computation notebook, Jupyter is a free and open source interactive notebook and online application that is used by many data scientists, researchers, and developers to mix software scripts and calculate them in one document. It is common for developers to utilise the Jupiter notebook to execute python code in various particular cells, which implies that it will assist the developer in running and checking any line of code in the application (Perkel, 2018).

Jupyter has also just released a new version of its software, JupyterLab, which enables free scientific computing from the Anaconda distribution. Developers may now upload .csv files, .Jpg files, and a variety of other file formats into it with relative ease. The developer has the option of either downloading JupyterLab or just running it directly in the browser.

3.3 Libraires Chosen

3.3.1 Numpy

Numpy is a library used in python which adds support to deal with large multidimension arrays and matrices, along with complex multiple mathematically functions.

NumPy

Jim Hugunin was the creator of Numeric, which is the forerunner to NumPy. Another package, Numarray, was created as well, which had some more features and functionality. Travis Oliphant developed the NumPy package in 2005 by merging the capabilities of the Numarray package into the Numeric package (Team, 2021).

Some of the most notable characteristics of NumPy include its capabilities of slicing and dicing multidimensional data sets, its processing speed for data transformation, it's very massive broadcasting potential, and several other characteristics (Mayer, 2021).

3.3.2 Pandas

It is a python package that offers quick, versatile, and expressive data structures that are designed to make working with labelled and relational data simple and straightforward. It is intended to aid the developer by serving as the core high-level building block and to do advanced analysis in Python. It is written in Python. Pandas is a programming language that works in both R and Python and has the same functionality in both programming languages (Pandas Org, 2020).

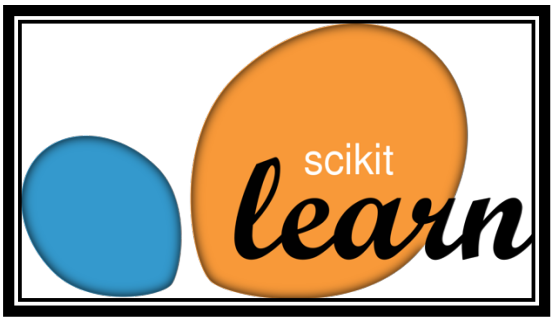
Pandas

Pandas is a tool for dealing with huge tabular data sets, such as those found in SQL or Excel, as well as sorted and unordered data, and any other kind of tabular data collections. Amongst its many features are the ability to handle missing data in very large data sets by representing them as NaN, the ability to insert columns and rows between very large data frames, automatic data alignment, merging and joining large

data sets, loading different types of data sets (.Xls,.Xlxs,.Csv, and so on) into the environment, data rage generation in time series data sets, and much more.

3.3.3 Scikit Learn

Scikit-learn is the name of a free machine learning library that may be used with Python. It is compatible with the NumPy and SciPy libraries for the Python programming language, in addition to a number of algorithms like as support vector machines, random forests, and k-neighbours. The Sklearn package contains a wide variety of powerful algorithms for machine learning and statistical modelling. Some examples of these algorithms include classification, regression, clustering, and dimensionality reduction (Jain, 2015).



Scikit Learn

3.3.4 Keras

The Keras Application Programming Interface was designed with users, not computers, in mind from the beginning. Keras adheres to best practises for reducing cognitive load, which includes delivering consistent and simple APIs, reducing the amount of user interaction required for typical use cases, and giving clear and responsive error signals. These features all contribute to Keras's ability to lower cognitive load (Keras, 2019).

The Keras library runs on both Python 2.7 and 3.5 with an heavy load on the CPU and GPU of the computer, it is a very powerful library developed and maintained by François Chollet a google engineer using four main principles: Modularity, Minimalism, Extensibility and python (Brownlee, 2016).

Keras Tenser flow

3.4 Operating System chosen for this project

Operating systems are pieces of software that manage hardware and software components, as well as perform a variety of other functions in a computer system. There are several operating systems available on the market, including Windows, MacOS, Linux, and others.

MacOS Monetary

The operating system used for this project is MacOS Monetary (Version 12), which is compatible with both the programming language and integrated development environment (IDE) selected for the project.

3.5 Web Browser chosen for the project

It is an application that enables the developer to access the world wide web or any local website anywhere in the globe using a web browser. When connecting to the internet, it may be done by either a computer or a mobile phone. There are several well-known browsers available on the market, including Chrome, Firefox, Brave, Opera, Safari, and others.

Safari

Web browsing software produced by Apple Inc. was selected for this project. It is called Safari, version 15.5. Due to the fact that Safari is highly recognised in the industry for its speed and ram consumption, it is a lightweight web browser that accomplishes all of the functions that other browsers can, but in a smoother and quicker manner than the competition.

3.6 Summary

Here, the technical tools that will be used in the project are thoroughly outlined, including the libraries that will be used, operating system that will be used, integrated development environment (IDE) that will be used, and web browser. The tools that have been chosen are well-known to many developers, are stable across many operating systems, and are extensively used throughout the globe, among other factors. The libraries selected, on the other hand, may differ depending on which browser and IDE are used and how they are utilised. The below are the summary of the selected tools and techniques that are going to be used in the project.

Elements	Component
Programming language	Python
IDE	Jupyter Notebook and Google Collab
Libraires chosen	Pandas and NumPy
Web Browser	Safari
Operating System	MacOS Monetary (Version 12)

4. Methodology

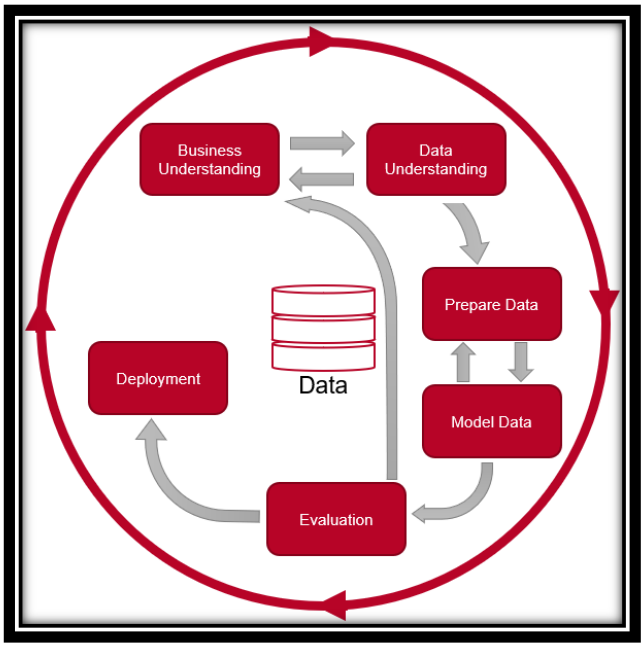
4.1 Introduction

A project management development methodology is a set of principles and practises that are used in the course of a project's development. It organises the process and maintains consistency throughout the project, from the beginning to the finish. An effective project management technique will be beneficial to the developer in a variety of specific ways. A project management methodology is a tool for increasing efficiency as it is used; it organises projects, provides tools and methods, estimates the tools and techniques appropriately, assists in managing and minimising project risk, and helps to manage and reduce project risk (Asana, 2021).

4.2 Method: Crisp DM Methodology

During the early 1990s, as data mining matured from a kid to a teenager, we expended a great deal of time and effort preparing data that was suited for the extremely limited tools and processing power available at the time. 'Data scientists,' as we were referred to at the time, were seldom seen in groups larger than one or two people, and we were much more likely to be referred to as 'predictive modellers,' since that kind of modelling was considered cutting-edge at the time. When it comes to data mining, CRISP-DM (Cross Industry Standard Process for Data Mining) is a method that was established in 1996 to aid in the shaping

of data mining projects. A Data Mining project is conceptualised in six stages, with cycle iterations depending on the needs of developers in each step.



Crisp DM Methodology

The events shown in this model are a hypothetical sequence of occurrences. However, many of the actions may be done in any order, and it will be required to go back and repeat several operations on a regular basis. During the data mining process, the model does not make an effort to capture all of the possible outcomes. Processes involved include Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment, to name a few.

Phase 1: Business Understanding

Aiming to put the objectives and data into context so that the developer or engineer can understand how data fits into the business model, the first level is Business Understanding. Their contributions include meetings, online meetings, documentation reading, specific field learning opportunities, and a plethora of additional ways they said the development team, such as asking crucial context questions.

In the wake of this phase, the development team has a greater awareness of the environment in which the project is being developed. Prior to starting a project, it is vital to define the goals and objectives of the project. Consider the following example: the development team should be aware that the aim is to increase sales before beginning work, and after that phase is complete, they should be familiar with what the customer is selling and how they are selling it. The data set is obtained from Kaggle.com (https://www.kaggle.com/datasets/varpit94/bitcoin-data-updated-till-26jun2021).

Phase 2: Data Understanding

The second step is Data Understanding, which is concerned with determining what can be anticipated and done with the data collected. It evaluates the quality of the data in a number of ways, including data completeness, value distributions, and compliance with data governance standards.

Considering that it defines how practical and dependable the final conclusions may be, this is an extremely crucial component of the project. During this stage, team members discuss how to extract the most amount of value from the various pieces of data. The development team may stop for a time to better understand the company and how it benefits from a specific piece of data if they are unclear of its value or relevance.

Phase 3: Data Preparation

The third step is Data Preparation, which includes the ETLs or ELTs process, which employs algorithms and procedures to convert data into something that can be manipulated and used. Data engineers and data scientists are responsible for standardising data when data governance rules are not followed or developed in an organisation. This ensures that the data has true meaning. The same is true for algorithms, with some doing better under certain situations than others. For example, some algorithms do not accept non-numerical numbers, while others do not perform well when dealing with a large range of values. The development team, on the other hand, is in charge of standardising the data collected. This was the area where the majority of the projects spent the most of their time. Given that these jobs are time-consuming and may become extremely sophisticated when dealing with large amounts of data, I believe that IT organisations would benefit from allocating resources only to them at this point in their development cycle.

Phase 4: Data Modelling

It is the fourth step that is at the core of every machine learning endeavour, and it is also the most time-consuming. This stage monitors the project's outputs, which should fulfil, or at the very least contribute to, the project's goals. Because if everything else is done well, there isn't much to modify, it takes the least amount of time, despite the fact that it is the most exciting component of the project. It is established that if the results can be improved, the procedure will be followed to return to the stage of data preparation and improve the data that has been made available.

Phase 5: Data Evaluation

The fifth step is evaluation, which comprises confirming that the results are right and that the interpretations are accurate and authentic. Alternatively, if the findings are incorrect, the technique allows for a review of the first stage in order to determine why the findings were incorrect. The developer divides the data into two categories while working on a data science project, which is often training data and testing data. It is at this step that the testing data is used, with the goal being to ensure that the model is as accurate as feasible. Every instance in which error measurement is used must include a reference to the source of the error measurement. This error metric conveys a developer level of confidence in the outcomes, which varies from "for certain this will work" to "for certain this will not." Whenever the error measure for all examples is

zero or none, the model is considered overfit, and reality may behave in a manner that differs from what was anticipated by the model. There must be some explanation of the error measurement's source provided in every occasion. As a result of this error measure, the user is informed of their degree of confidence in the results, which might range from "for certain this will work" to "certainly this will not work." Overfitting occurs when the error measure for all scenarios is zero or none, regardless of whether reality behaves in a similar manner to the model.

For the LSTM model the activation function will be sigmoid function, primary reason is the developer utilise the sigmoid function is that it may be found between and (0 to 1). As a consequence of this, it is utilised in particular for models the output of which is a probability prediction. The sigmoid distribution is the optimal choice due to the fact that the likelihood of anything occurring is always between 0 and 1.

The developer has set the number of epochs to 250 and the batch size as 5. The reason why the number if epochs is set as high because as the number of epochs increase, more number of times the weight is changed in the neural network and the curve goes from underfitting to optima to overfitting curve. The data split is also set to be 80/20 for the LSTM algorithm and 90/10 for the ARIMA model.

The developer will envaulted many times with many different number of epochs, different partition of data and batch size, and the developer will give the best one as the result.

Phase 6: Deployment

Deployment is the sixth and last step, and it comprises presenting the results in an useful and understandable form in order to achieve the project's goals as efficiently as possible. It is the only phase that does not occur as a part of a larger cycle. The technique that is beneficial and understandable will vary based on who will be the final user. In this project the deployment stage will not be applicable for the project, because the price of bitcoin is very volatile and unstable so it might affect the final result if the data keeps updating. The results obtained below will be considered as the accurate values only, the developer cannot assure the exact price of bitcoin because of the fluctuations of the price in the real world.

4.3 Conclusion

The technique that was decided to be used is called Crisp-DM, and it takes an iterative approach to the flow of the project. One of the benefits of using Crisp DM is that it will make it possible for the developer to reverse the flow of the project at any step of the process.

The business understanding stage of the project is where the developer will learn more about the cryptocurrency and bitcoin in the context of the market trend during the beginning stage of the project. In addition to this, the developer will have an understanding of how machine learning algorithms and deep learning principles will contribute to the smooth flow of processes inside the project. After gaining an awareness of the patterns and activities shown by bitcoin in the market, the next phase of the project may

be chosen. The following step for the developer is to locate the dataset that is appropriate for the project. The developer will now locate an appropriate data set and get an understanding of the data included inside the data set. The following phase is data preparation, during which the developer will organise the data set in preparation for the further steps of data cleaning, data visualisation, and data transformation, if any of those steps are necessary. The following phase in the process is for the developer to produce an adequate data model, during which the data set will be put through a number of machine learning and deep learning algorithms, which will ultimately result in an appropriate and suitable data model. Following the completion of the evaluation, the model will be deployed so that the developer may discover it and comprehend its findings.

5. Data Analysis

In this chapter, discussion and illustration of the data that the developer is utilising in order to complete this project. The data set's developer will explain the source of the data, the method that was really carried out, as well as the requirements for data exploration, data pre-processing, and data modelling.

5.1 Introduction

The developer has obtained the data set from Kaggle.com where it is used to predict and model the data. The data set is obtained in .csv format and imported directly to Google Collab environment where the data pre-processing and data visualization. The data set downloaded is named as BTC-USD.csv. The file has a total 7 rows and 2714 columns of bitcoins and its price.
The below table is the listed attributes for the data set.

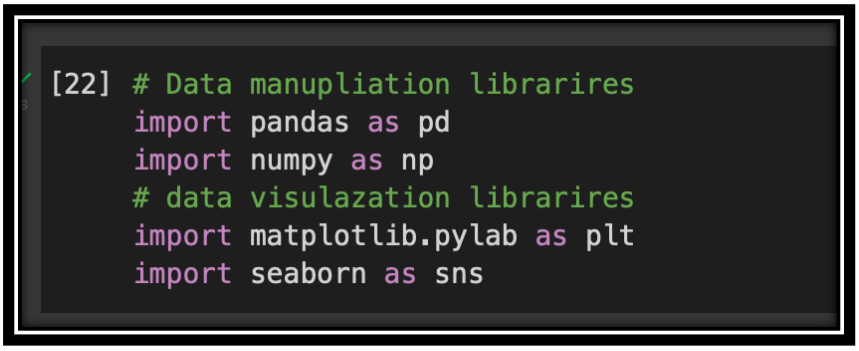
No	Variable	Type of the variable	Description of the variable
1.	Date	Object	The date of the bitcoin when it was released.
2.	Open	Numerical	The price of the bitcoin when it is open. (E.g., Start of the day)
3.	High	Numerical	The highest price of all time of the bitcoin.
4.	Low	Numerical	The lowest price of all time of the bitcoin.
5.	Close	Numerical	The price of the bitcoin when it is closed. (E.g., End of the day)

6.	Adj.Close	Numerical	The closing price of the bitcoin after all the adjustments for all applicable splits.
7.	Volume	Numerical	The sum of total of actual trades of the bitcoin.

The next step of this phase is exploring the data set which the developer will conduct the exploratory data analysis.

5.2 Initial data exploration

First, we import the required libraires for data manipulation and data visualization for the data set into the environment.



Required libraires

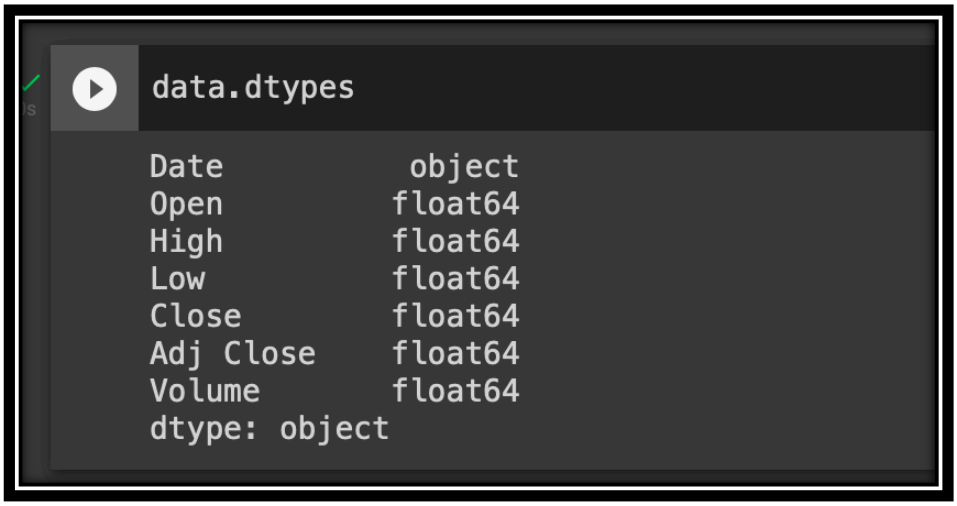
The initial data set is small enough to directly import into the collab environment. The data set is imported in the format of .CSV file.

Importing the data set

The data set is successfully imported into the collab environment.

Shape of the data

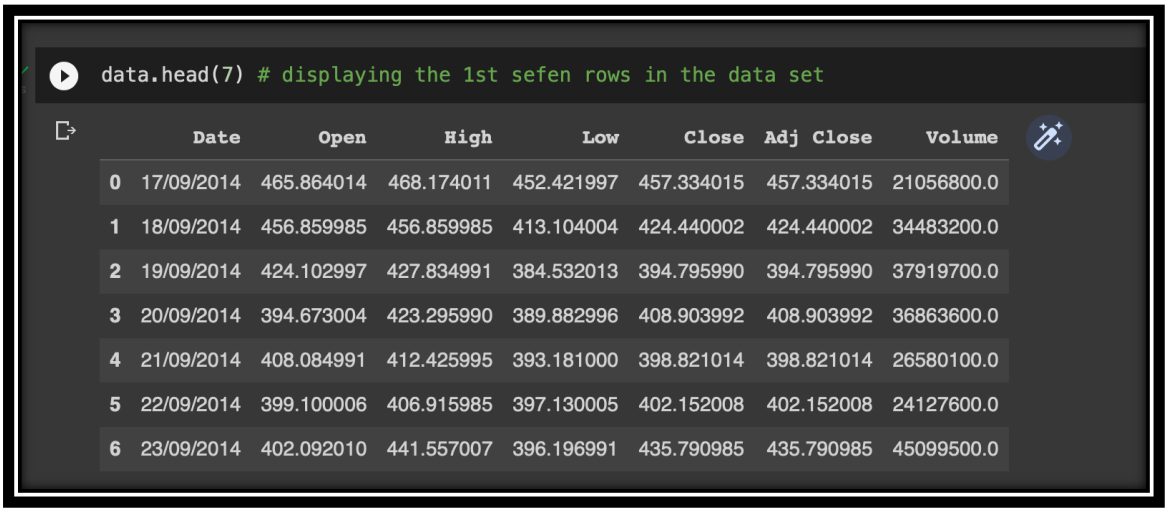
The data set has a total of 2713 columns and a total of 7 rows.



```
data.dtypes

Date          object
Open         float64
High         float64
Low          float64
Close        float64
Adj Close    float64
Volume       float64
dtype: object
```

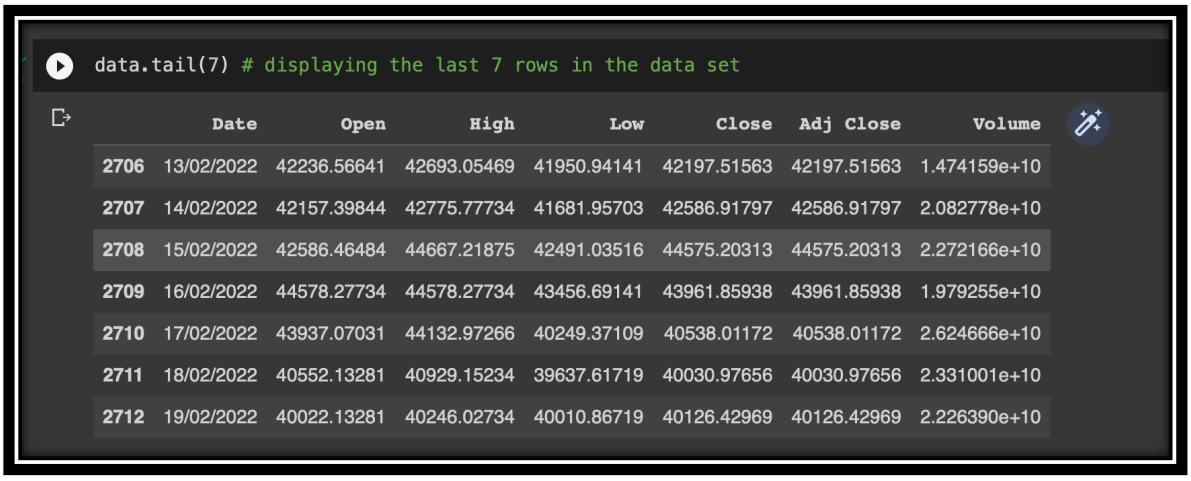
Data types in the data set



```
data.head(7) # displaying the 1st sefen rows in the data set
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	17/09/2014	465.864014	468.174011	452.421997	457.334015	457.334015	21056800.0
1	18/09/2014	456.859985	456.859985	413.104004	424.440002	424.440002	34483200.0
2	19/09/2014	424.102997	427.834991	384.532013	394.795990	394.795990	37919700.0
3	20/09/2014	394.673004	423.295990	389.882996	408.903992	408.903992	36863600.0
4	21/09/2014	408.084991	412.425995	393.181000	398.821014	398.821014	26580100.0
5	22/09/2014	399.100006	406.915985	397.130005	402.152008	402.152008	24127600.0
6	23/09/2014	402.092010	441.557007	396.196991	435.790985	435.790985	45099500.0

The first seven rows in the data set.



```
data.tail(7) # displaying the last 7 rows in the data set
```

	Date	Open	High	Low	Close	Adj Close	Volume
2706	13/02/2022	42236.56641	42693.05469	41950.94141	42197.51563	42197.51563	1.474159e+10
2707	14/02/2022	42157.39844	42775.77734	41681.95703	42586.91797	42586.91797	2.082778e+10
2708	15/02/2022	42586.46484	44667.21875	42491.03516	44575.20313	44575.20313	2.272166e+10
2709	16/02/2022	44578.27734	44578.27734	43456.69141	43961.85938	43961.85938	1.979255e+10
2710	17/02/2022	43937.07031	44132.97266	40249.37109	40538.01172	40538.01172	2.624666e+10
2711	18/02/2022	40552.13281	40929.15234	39637.61719	40030.97656	40030.97656	2.331001e+10
2712	19/02/2022	40022.13281	40246.02734	40010.86719	40126.42969	40126.42969	2.226390e+10

The last seven rows in the data set

```
[15] data.isnull().sum() # to check the data set if has any null values

Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

Missing values/Null values in the data set.

The above picture shows that there are no missing values or null values in the data set. All the variables in the data set are float64, which are all numerical variables.

Using the library *matplotlib* with the functions *plt.plot()* detailed description of the data is given below.

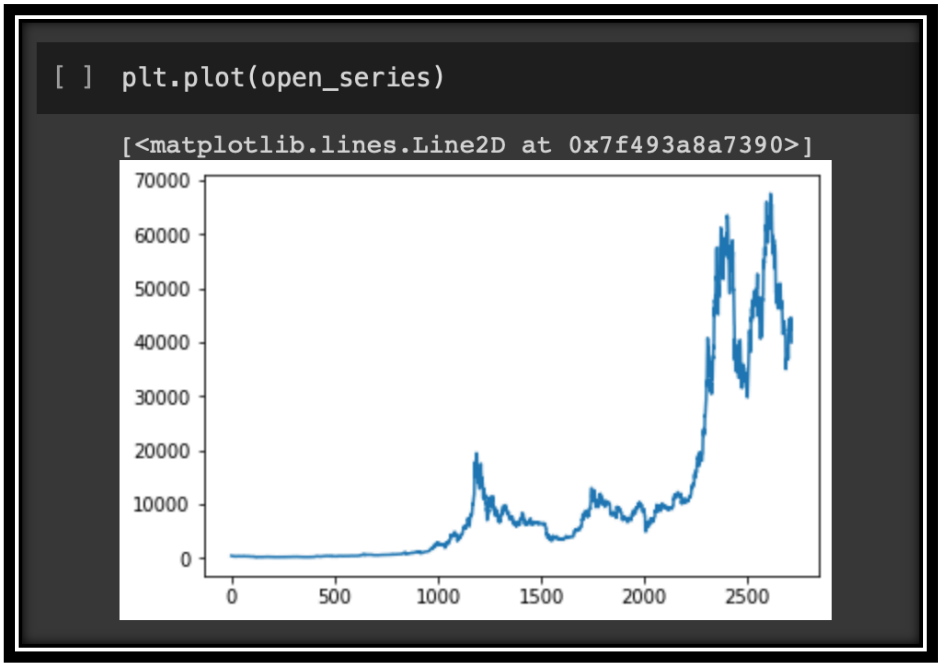
```
data.describe() # Data Exploration
```

	Open	High	Low	Close	Adj Close	Volume
count	2713.000000	2713.000000	2713.000000	2713.000000	2713.000000	2.713000e+03
mean	11311.041069	11614.292482	10975.555058	11323.914637	11323.914637	1.470421e+10
std	16106.428892	16537.390649	15608.572561	16110.365010	16110.365010	2.001437e+10
min	176.897003	211.731003	171.509995	178.102997	178.102997	5.914570e+06
25%	606.396973	609.260986	604.109985	606.718994	606.718994	7.991080e+07
50%	6301.569824	6434.617676	6214.220215	6317.609863	6317.609863	5.098183e+09
75%	10452.399410	10762.644530	10202.387700	10462.259770	10462.259770	2.456992e+10
max	67549.734380	68789.625000	66382.062500	67566.828130	67566.828130	3.510000e+11

Data Description using .describe()

- **Open**

  It is the most important time to check out for the price of the bitcoin at this moment because it is starting for the day when the open price is known. This is because the open variable in the data set means it the price of the bitcoin at the starting point of the bitcoin when it is created or when it is mined. In other words, it is the price of the bitcoin when it is created. There are a total of 2713 values included in the data set, with a mean value of 11311.04 and a standard deviation of open price that equals 16106.42.

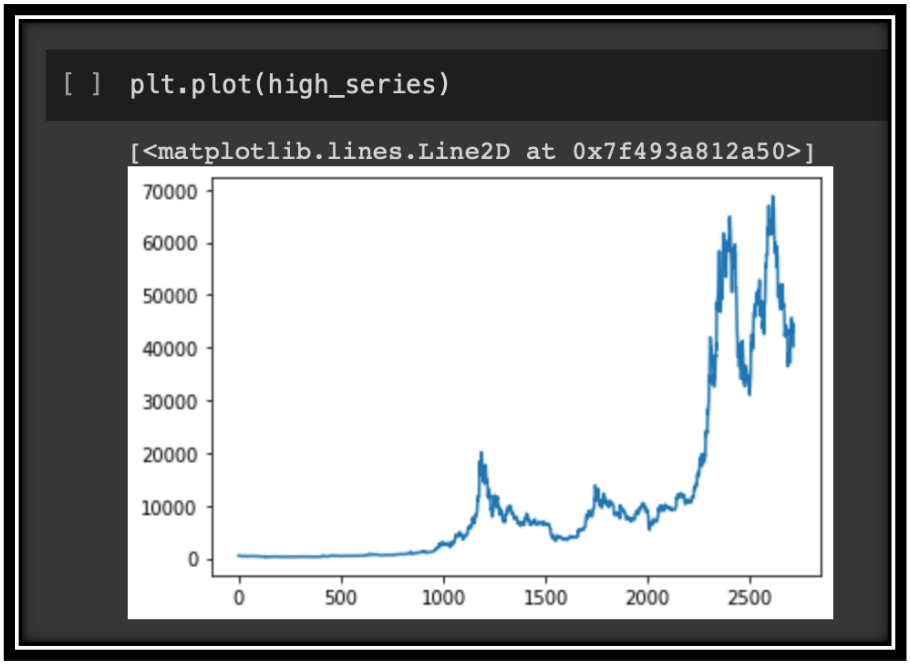


Open series

- **High**

  The term "High" or "All Time High" (ATH) refers to the point in the day when the price of bitcoin is at its highest. Which indicates that the price of bitcoin has reached an all-time high. The High value not only indicates the potential highest price that one could have sold the given item for, but it also represents the maximum price that another trader was willing to pay for that asset during that time period. In other words, the High value indicates the highest price that another trader was willing to pay for that asset. To put it another way, the High value indicates the greatest possible price for which the object in question could have been sold. It is possible that the high price was acquired through the transaction of a fraction of an asset rather than a coin or token in its entirety due to the fact that the vast majority of digital assets are structured in the form of fractions. This is because the vast majority of digital assets are structured in the form of fractions (Binance Academy, 2021).

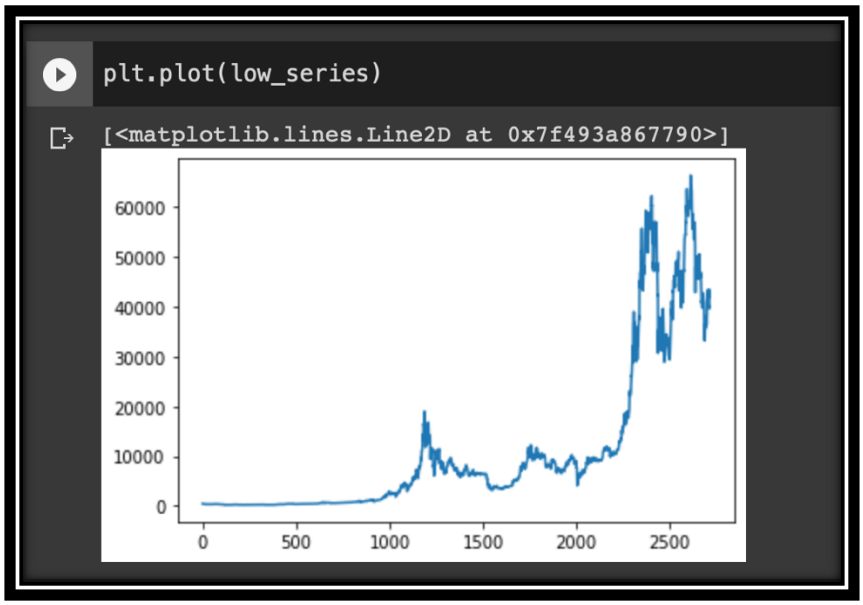
  There is a total count is 2713 variables in the data set, which has a mean of 11614.29 with a standard deviation of 16537.39.



High series

- **Low**

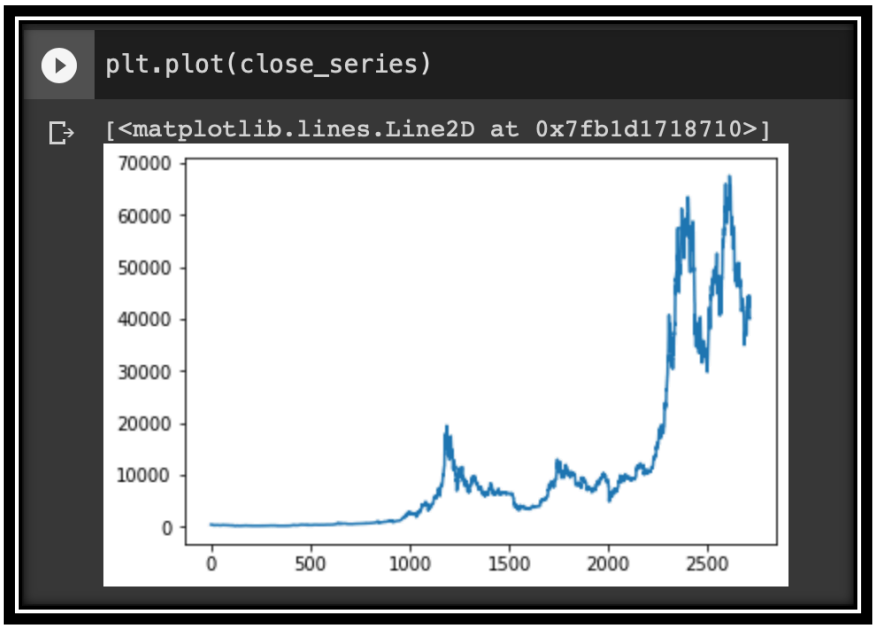
  Low or All Time Low (ATL) is the term for when the bitcoin is the lowest price of its all-time in the day. This column also has a total count of 2713 and a standard deviation of 15608.57 with a mean of 10975.55.



Low series

- **Close**

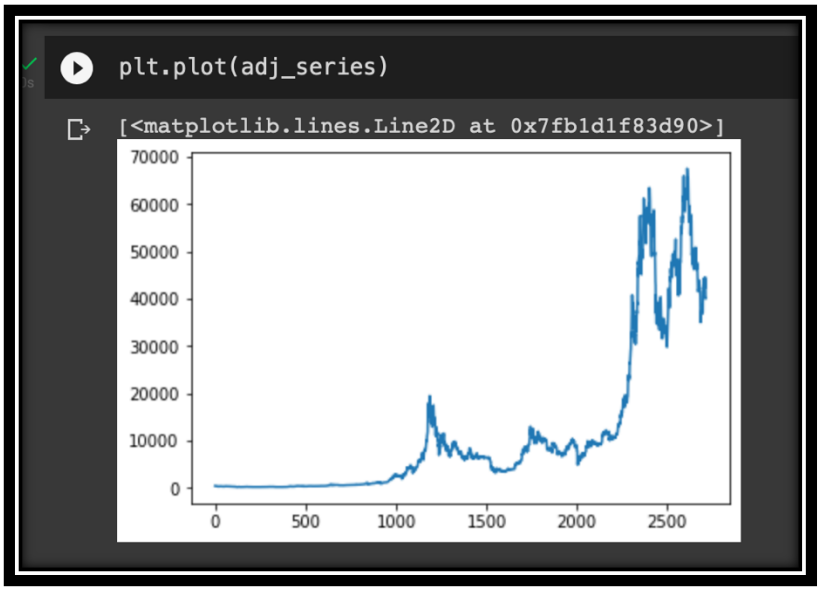
  Close is a reference to the end of a trading session in the crypto currency. In simple terms it means the price of the bitcoin at the end of the day. It has a total count of 2713 in the dataset and the with a standard deviation of 16110.36 and with a mean of 11323.91.



Close series

- **Adj Close**

  The "adjusted close" refers to the closing price of a security after all applicable splits and dividend pay-outs have been factored into the calculation. The data is adjusted so that it is compliant with the rules established by the Centre for Research in Security Prices (CRSP) by applying the appropriate split and dividend multipliers (Hartill, 2021).

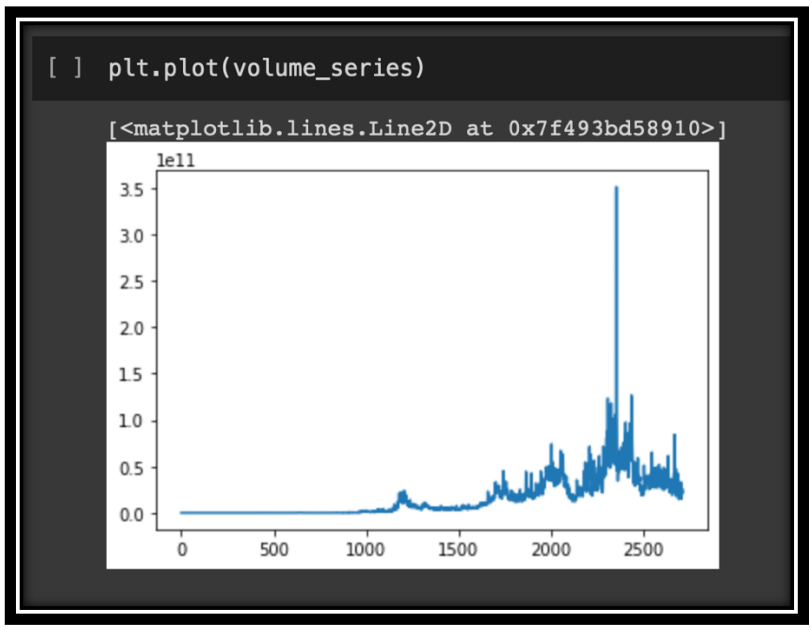


Adjacent series

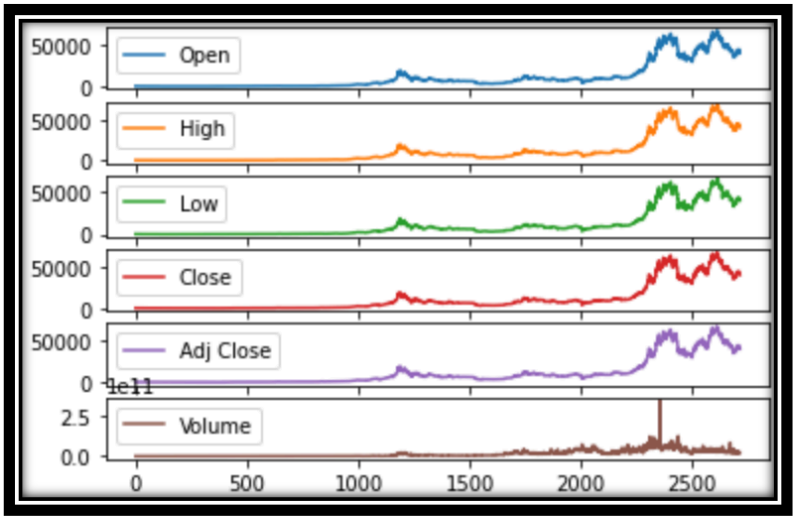
The total count is 2713 and has mean of 11323.91 with a standard deviation of 16110.36.

- **Volume**

  The term volume means how many times the bitcoin has been changed over hand in certain period of time in a given time frame. It is an important factor in the cryptocurrency world where the traders use it to project the projectile of a bitcoin (Nibley, 2021).



Volume series



Overview of the columns in the dataset

The above figure shows the overview of all the columns is the dataset. Here we will be focusing on the volume because the total volume will impact on the information of the bitcoin where how much is affected in the market. There is a clear correlation between the total volume of transactions for a certain cryptocurrency and its level of volatility. Remember that the price is a reflection of the relative perceptions of buyers and sellers in the market. When you look into trading a certain cryptocurrency, the aggregate

volume traded will quickly inform you how volatile the cryptocurrency is going to be. The developer's interest will be piqued further by the fact that the volume of the project changed over time (LearnCrypto, 2021).

5.3 Long short-term Memory (LSTM)

5.3.1 Data preparation

Here the *Closing* price is chosen as the target variable and stored in a new data frame. With the help of *Mathlab* library the developer as plotted the graph of the Closing price of the data set.

```
price = data[['Close']] # dataframe
# ploting the graph
plt.figure(figsize=(20,9)) # figure size
plt.plot(price) # ploting the graph
plt.xticks(range(0,data.shape[0],50),data['Date'].loc[::50],rotation = 45) # settig up the ticks
plt.title('Closing Price', fontsize = 18, fontweight='bold') # setting the title of the grpah
plt.xlabel('Date',fontsize=18) # title for the x axis
plt.ylabel('Price', fontsize = 18) # title for the y axis
plt.show()
```

Plotting the Closing price variable

First we create a new variable *Price* and put the Close price as a data frame in it. With the help of plt() the developer will plot the price variable.

Xticks() functions helps the developer to plot the values according to the range set by the developer, this function controls of the plot range, tick locations, and the tick labels. *plt.title()* is a function where it helps to the set the title of the graph by the developer, *plt.xlabel()* and *plt.ylabel()* allows to set the names for the x-axis and y-axis respectively for the developer (Vanderplas, 2017).

Closing price

From the above graph we can see the bitcoin values are from 2014 up to 2022. Where we can see a sudden fluctuation in the year 2017 and a sudden increase of the price from 2021 and it got low at the mid of 2021 and increased again by the end of 2021.

```
price.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2713 entries, 0 to 2712
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Close   2713 non-null   float64
dtypes: float64(1)
memory usage: 21.3 KB
```

Brief information of the closing price

The Closing price is transformed into a data frame and stored in the variable price. With the help of info() function a brief exploration of the column is shown above the picture. The price column has a total count of 2713 non-null values and as a data type of float64.

5.3.2 Data normalization

Before beginning to construct a data model, the most critical step is to first normalise the data. If the process is skipped, it will have an effect on the correctness of the data that is used to construct the model; also, if data normalisation is not performed, it will be impossible to locate a good data model.

The *Close* price consist of many large values in the data set, the data normalization is done which will help to transform the data from large values to 0 to 10 values.

To do data normalization first we import the required library from Sklearn.

```
from sklearn.preprocessing import MinMaxScaler # import the libraries for normalzation
min_max_scaler  = MinMaxScaler() # this function returns the value bertween 0 to 10
norm_data = min_max_scaler.fit_transform(price.values) # fitting of the values (Normalazation)
```

MinMaxScalar function

The *min_max_scalar* holds the MinMaxScaler function and it is called in the next line, and the normalized data is stored in the variable *norm_data*.

```
norm_data # values after data normalization

array([[0.00414359],
       [0.00365546],
       [0.00321557],
       ...,
       [0.59891189],
       [0.59138785],
       [0.59280431]])
```

Values after Normalization

5.3.3 Data split

The process of dividing a set of data into two or more subgroups is referred to as "data splitting." In a two-part split, one of the parts is often dedicated to testing or assessing the data, and the other half is dedicated to training the model. The process of data splitting is an essential part of data science, particularly when developing models based on the data. The building of data models and the processes that make use of data models, such as machine learning, can both benefit from using this strategy to help ensure correctness (S.Gillis, 2021).

In this data model the developer will split the data into two parts which is training set and testing set data. After the data split is done the developer will validate the two variables. The developer will use the training set to train the data, which is 80% of the data. The testing set data ideally consist of the rest 20% of the data. Here the testing data is used as a base line comparison for testing purpose.

The developer has created a *univerate_data()* function which the function accepts the dataset, starting index of the data, the ending index of the data, history size and the target size.

start_index helps the developer to get the starting index of the function, where the end_index will get the ending index of the function by subtracting the length of the data size and the target size, the end_index will be placed in a loop which will run till the end of the data set where the arguments are given the length of the data set. The *past_data* is set to 5 because we will be using 5 days of data to learn and point the data in the time series, the next point in the time series is the *future_target* which is now assigned as 0. After the variables are set the function (*univerate_data()*) is called by passing the arguments to the function.

```
# importing the librarys
import keras
from keras.models import Sequential # model import
from tensorflow.keras.optimizers import Adam # optimizer
from keras.layers import Dense, LSTM, LeakyReLU,Dropout # layers

num_units = 64 # the number of nurons

learning_rate = 0.0001
activation_funtion = 'sigmoid' # activation function
adam = Adam(lr = learning_rate)
loss_function = 'mse' # loss function = mean squared error
batch_size = 8
num_epochs = 250 # number of epochs (trial/error)

# starting the RNN
model = Sequential()
model.add(LSTM(units = num_units,activation=activation_funtion,input_shape=(None,1)))
model.add(LeakyReLU(alpha=0.5))
model.add(Dropout(0.1))
# this layer will now stop from overfittig
model.add(Dense(units=1))

model.compile(optimizer=adam,loss = loss_function) # model buliding function
```

Data split (80/20)

5.3.4 Model building

Steps involved in the process of building a model include establishing methods for data collection, comprehending and paying attention to what is significant in the data in order to answer the questions that you are posing, and creating a statistical, mathematical, or simulation model in order to acquire understanding and make predictions.

Here the developer will use the LSTM model with the model layer of 50 neurons and the *relu* activation function. With the help of *keras.model* the developer will install the *sequential* model, as for the activation function the developer will be using *adam* , *Dense, LSTM, LeakyReLU,Dropout* are the layers imported from Kears library.

The number of units which the number neurons are set as 64, the learning rate is set as 0.0001, the activation function as *sigmoid* and loss function as *mse* (mean-squared error), the batch size as 5 and the number of epochs as 250.

```
# importing the librarys
import keras
from keras.models import Sequential # model import
from tensorflow.keras.optimizers import Adam # optimizer
from keras.layers import Dense, LSTM, LeakyReLU,Dropout # layers

num_units = 64 # the number of nurons

learning_rate = 0.0001
activation_funtion = 'sigmoid' # activation function
adam = Adam(lr = learning_rate)
loss_function = 'mse' # loss function = mean squared error
batch_size = 5
num_epochs = 250 # number of epochs (trial/error)

# starting the RNN
model = Sequential()
model.add(LSTM(units = num_units,activation=activation_funtion,input_shape=(None,1)))
model.add(LeakyReLU(alpha=0.5))
model.add(Dropout(0.1))
# this layer will now stop from overfittig
model.add(Dense(units=1))

model.compile(optimizer=adam,loss = loss_function) # model buliding function
```

Model Building

Here the epochs is 250 because the more number of times the model is trained the less overfitting we get which leads us to a better accuracy result. Even if the epochs is set to higher value also then the model will have a chance to overfit. To get the more accuracy of for the model the developer will keep changing the number of epochs with many different values and take the best one for the result. The developer will now initialize the RNN model in a sequential method by calling the *sequential()* function and adding the layers to it, so the function executes in a sequential way.

The *model.dense()* function is only true network layer in which each neuron in a dense layer receives one output from the layer that comes before it, and this output is then supplied to all of the neurons in the layer. It is the layer that is considered to be the most fundamental in neural networks.

Now after the model building is done the developer will use the *model.summay()* function to get the summary of the model.

```
model.summary() # summary of the model

Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 64)                16896

 leaky_re_lu_2 (LeakyReLU)   (None, 64)                0

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 16,961
Trainable params: 16,961
Non-trainable params: 0
_________________________________________________________________
```

Summary of the model

The total parameters are 16,961 and trainable parameters are 16,961.

5.3.5 Model Training

```
#Training the model.
history = model.fit(
    x_train,
    y_train,
    validation_split = 0.1,
    batch_size = batch_size,
    epochs = num_epochs,
    shuffle=False
)
```

Training the model

The history variable is used for getting the training data, the hyper parameters are stated above in the code and the shuffle is set to false. The data which the developer is working on is sequential model. If the shuffle is set true then the date in the data will get messed up making the values go wrong as this is not a chorological data.

5.4 Auto Aggressive Moving Average (ARIMA)

First the developer will import the required libraries for building the ARIMA model.

```
# ARIMA Model
from statsmodels.tsa.arima_model import ARIMA
# mean squared error and mean absolute error libraries
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

Required libraries

5.4.1 Data Preparation

The developer will now again import the data into the variable *data2* for ARIMA model.

```
dateparse = lambda dates: pd.datetime.strptime(dates,'%d/%m/%Y') # date phaease
data2 = pd.read_csv('/content/BTC-USD.csv',index_col = 'Date',parse_dates = ['Date'],date_parser = dateparse)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: The pandas.datetime class is depr
  """Entry point for launching an IPython kernel.
```

Importing the Dataset into the Environment

5.4.2 Data Split

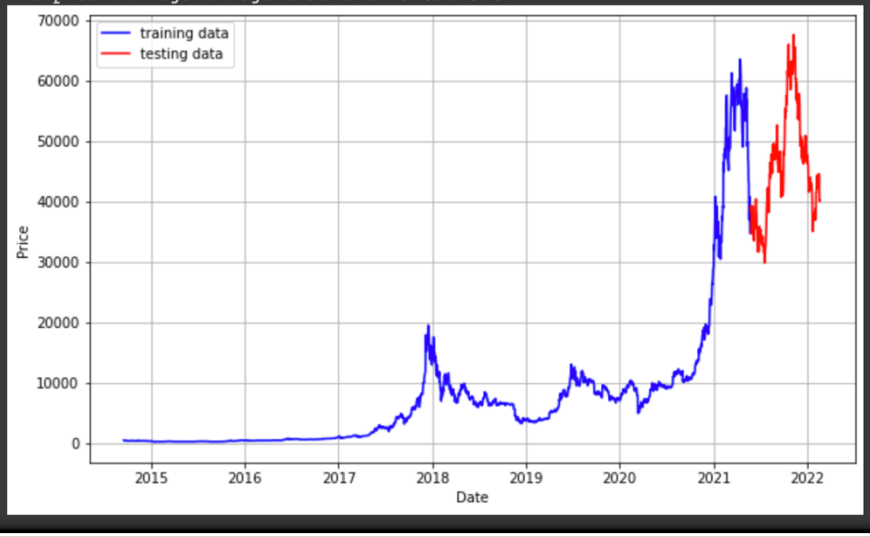
The developer will now split the data into 90/10.

```
# train test split
train = int(len(data2)*0.9) # Split the data
train_data = list(data2[0:train]['Close']) # 90% of the data
test_data = list(data2[train:]['Close']) # 10% of the data
```

Splitting the data

```
# Visualizing the testing the training data
plt.figure(figsize=(10,6)) # figure size
plt.grid(True) # enabling the grid
plt.xlabel('Date') # title for the x axis
plt.ylabel("Price") # title for the y axis
plt.plot(data2[0:train]['Close'],'blue', label = 'training data') # ploting the training data
plt.plot(data2[train:]['Close'],'red',label = "testing data") # ploting the testing data
plt.legend() # showing the legend
```

```
<matplotlib.legend.Legend at 0x7f1328ab2810>
```



Training and Testing data

5.4.3 Model Building

The developer will now build the model. Before building the model the developer will create an empty list and store the length of the test_data which will be used while building the model.

```
[8] model_predictions = [] # Creating an empty list
    n_test_obser = len(test_data) # saving the lenght of the test data
```

Creating the empty list and storing the length of the test data

```
[9]  for i in range(n_test_obser): # loop for the building the model
       model = ARIMA(train_data, order =(4,1,0))  # ARIMA Model
       modelfit = model.fit() # model Fitting
       result = modelfit.forecast() # model Forecasting
       yhat = list(result[0])[0]
       model_predictions.append(yhat) # saving the test data after running the model
       actual_test_data = test_data[i] # saving the test data
       train_data.append(actual_test_data) # Apending the data
```

Building the model

The developer will run a for loop for the length of rage of the test data where the model should be build. The developer will initialize the *p*=4, *d* = 1, *q*= 0, where p is Lag observation, d is Degree of differencing, q= moving average. After the model is build it will be fitted by *model.fit()*. The model forecasting in ARIMA model makes use of differencing to convert a non-stationary time series into a stationary time series, and then it uses the historical data in the data set to forecast future values based on those historical values. They are calculated by applying autocorrelations and moving averages across the lingering flaws in the data in order to make projections about the values of the future. The variable *model_predection* will contain the values after training the data with the help of ARIMA model, and the *test_data* will contain the values before the training of the model.

After the model is training is done the model summary as shown below:

```
print(modelfit.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.y   No. Observations:                 2711
Model:                 ARIMA(4, 1, 0)   Log Likelihood              -21873.605
Method:                       css-mle   S.D. of innovations            772.416
Date:                Thu, 15 Sep 2022   AIC                          43759.209
Time:                        02:17:52   BIC                          43794.640
Sample:                             1   HQIC                         43772.019

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         14.5974     15.305      0.954      0.340     -15.400      44.595
ar.L1.D.y     -0.0223      0.019     -1.165      0.244      -0.060       0.015
ar.L2.D.y     -0.0027      0.019     -0.139      0.889      -0.040       0.035
ar.L3.D.y      0.0100      0.019      0.519      0.604      -0.028       0.048
ar.L4.D.y      0.0458      0.019      2.378      0.017       0.008       0.084
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            2.1406           -0.0000j            2.1406           -0.0000
AR.2           -0.0802           -2.1520j            2.1535           -0.2559
AR.3           -0.0802           +2.1520j            2.1535            0.2559
AR.4           -2.1982           -0.0000j            2.1982           -0.5000
-----------------------------------------------------------------------------
```

Model Summary

The *p* value is 4 we get the lag observation of 4 models, where the co-efficient, standard deviation and the *z* value is shown in the figure above. The *z* values tells us the how many standard deviations are we away from the mean.

The developer will also do the same for the column *Open* and the results and discussions will be done in the next chapter.

6.0 Results and Discussions

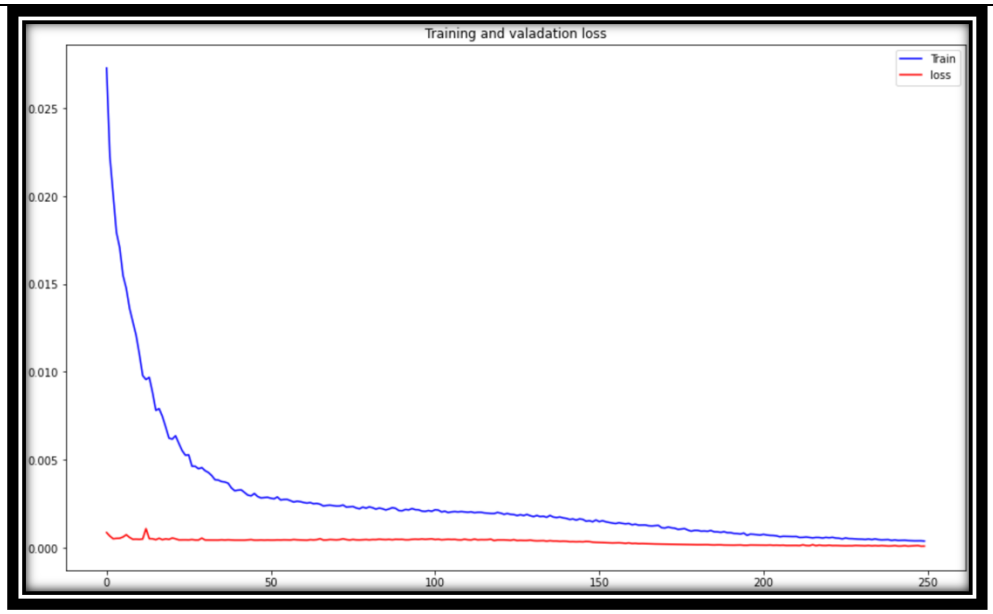
The developer has used two different models to predict the values from the data set. *Open* and *Close* are the two column used by the developer with the help of LSTM and ARIMA model.

6.1 Model Comparison

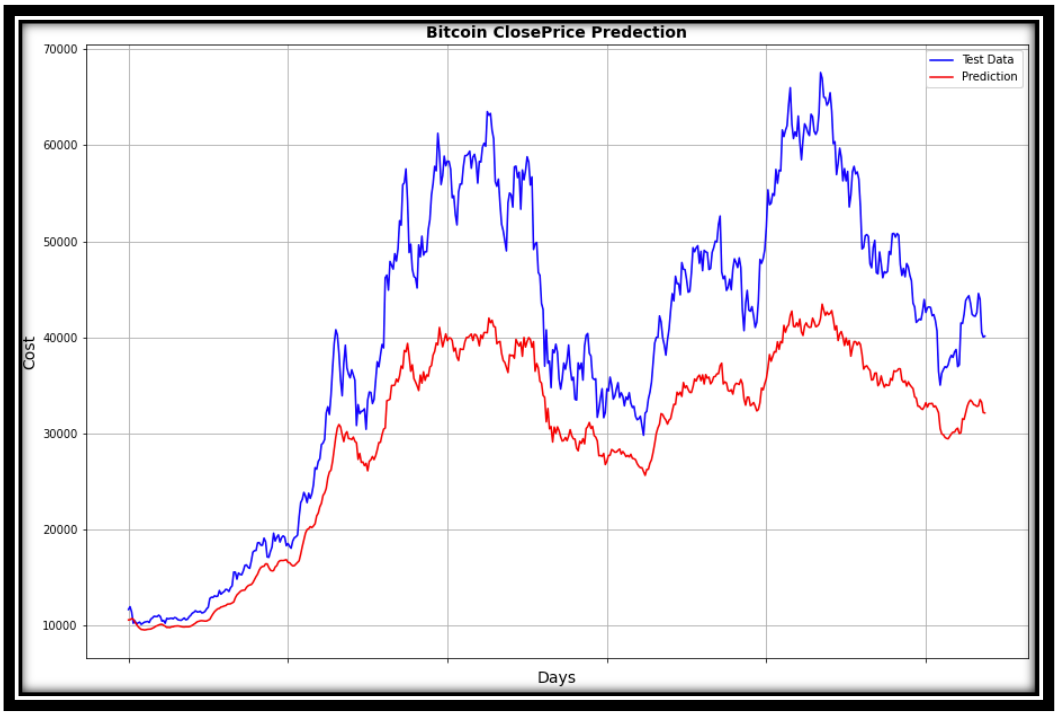
This section will show the clear comparison of the column and show the results between the two different models and recommend the best the model.

For the LSTM algorithm the data split is done for 80/20, whereas for the ARIMA algorithm the data split is done for 90/10.

Model = LSTM Column = Close Number of Epochs = 250 Batch Size = 5 Loss function = Mean Squared Error (MSE) Activation Function =Sigmoid Number of neurons = 64 Data Split = 80/20 MAPE value = 67.81
Training and Validation Loss

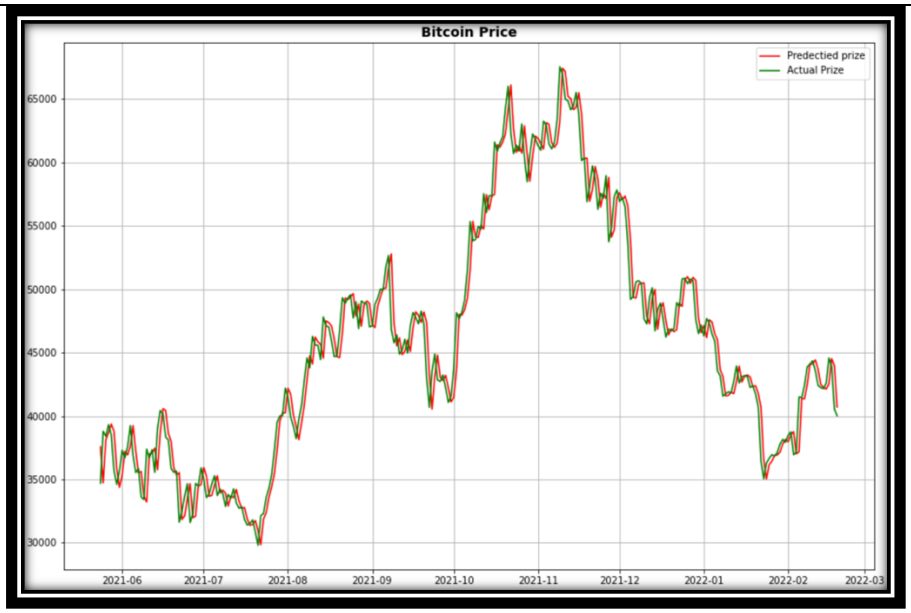


Predicted Price



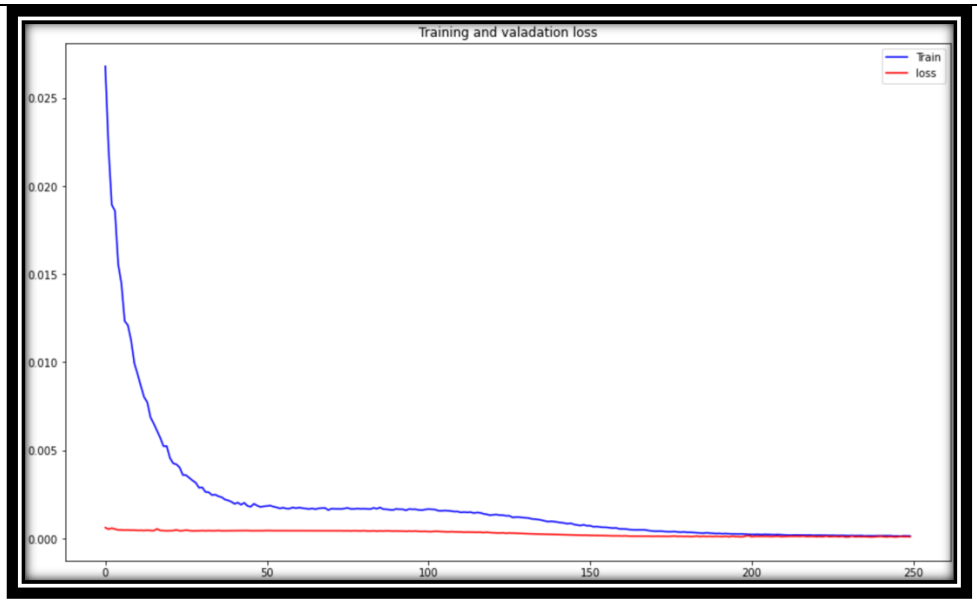
The above table explains the bitcoin price which the data has been processed with the LSTM algorithm with the number of epochs of 250 with a batch size of 5. The predicted price as shown above in the figure.

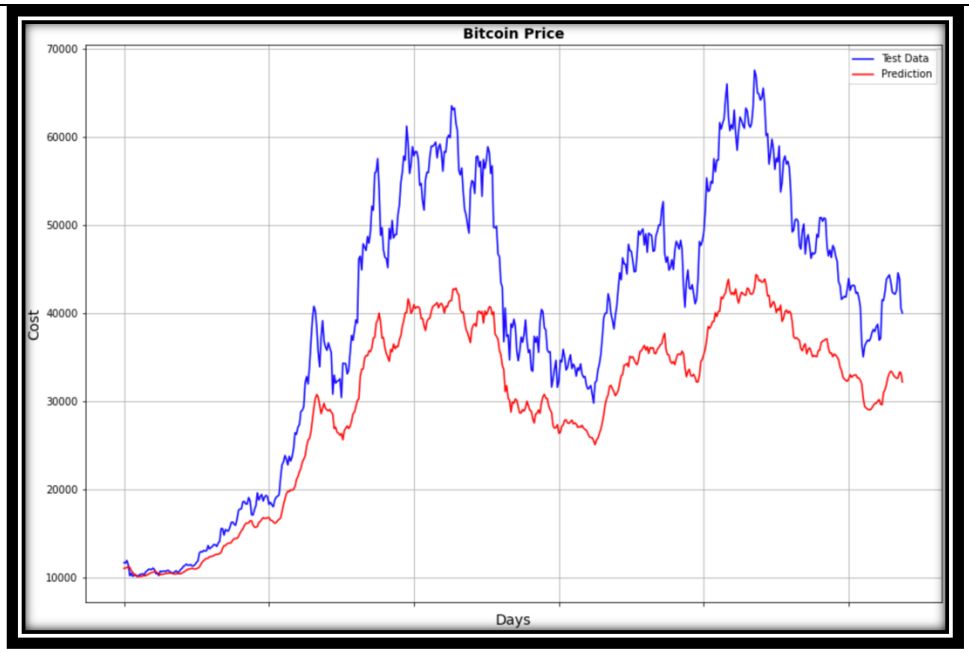
Model = ARIMA
Column = Close
Data Split = 90/10
p = 4
q = 1

d = 0 MAPE Value = 2.74


The above is the figure of the same column but processed in ARIMA model with test data as 90% and train data as 10%.

Model = LSTM Column = Open Number of epochs = 250 Batch size = 5 Loss function = Mean Squared Error (MSE) Activation Function =Sigmoid Number of neurons = 64 Data Split = 80/20 MAPE Value = 66.49
Training and Validation loss





The above figure shows the graph predicted for the column Open with the LSTM algorithm and the test data as 80%and train data as 20%. The epochs is set as 250 and the batch size as 5.

Model = ARIMA

Column = Open

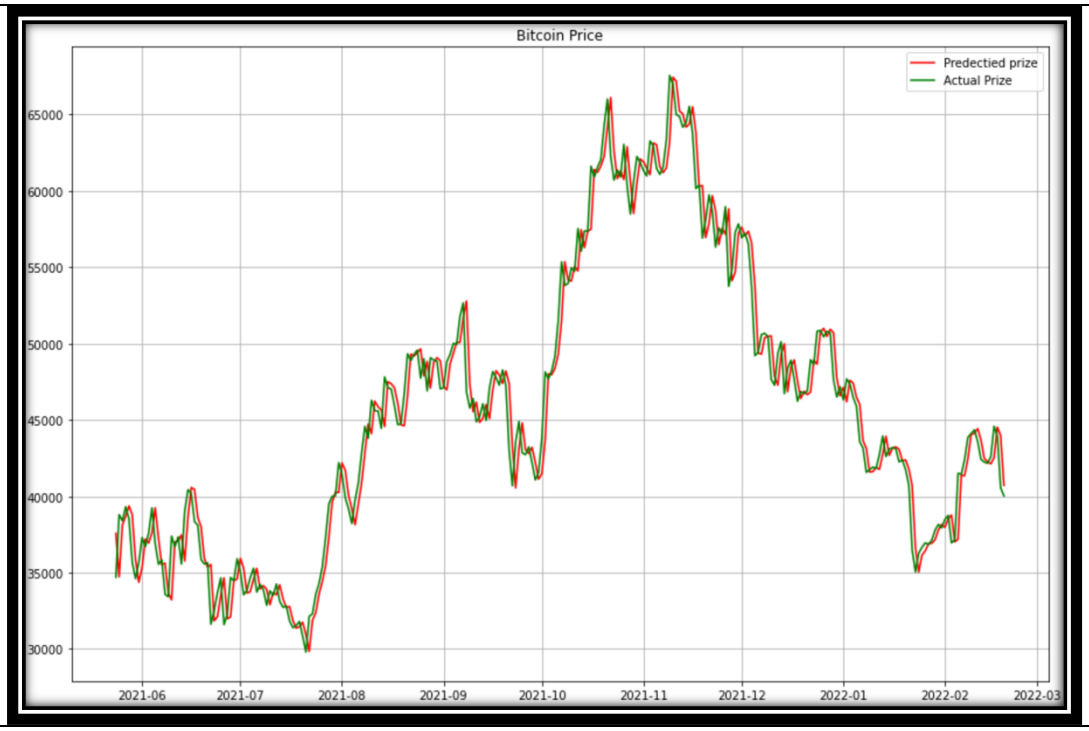
Data Split = 90/10

p = 4

q = 1

d = 0

MAPE Value = 2.87



The Open column in the data set is processed through ARIMA model and the above figure shows the price predicted.

Comparing the results obtained from the above models and MAPE value compared the best model is ARIMA for bitcoin price prediction. The reason is it is a common tool in demand forecasting, such as when anticipating the future demand for food supply or stocks prediction and bitcoin price prediction, and one of the common applications of this tool is as follows: When it comes to decisions regarding the supply chain, the developer will now have precise rules to follow. ARIMA models can also be used to forecast the future price of the bitcoin by analysing historical data. This is done using the data from the past (Bora, 2021).

The layers that are present between the input and output layers are considered to be hidden layers. Because of this fundamental concept, deep learning networks are frequently referred to as "black boxes," and they have been the subject of criticism for their lack of transparency and the inability of anyone to validate the accuracy of their predictions. Believe it or not, a strategy that relies on trial and error will offer the best results depending on the specific circumstances, given that there is no universally accepted standard for the number of nodes (hidden neurons) or hidden layers that should be employed.

For improving the LSTM model in this project, the data split should stay 80/20 and increase the number of epochs with batch size as less than the usual which is 8.

7.0 Conclusion

There are a lot of individuals across the globe who are interested in learning about and investing in the cryptocurrency sector. But as we all know, cryptocurrency is not stable, and it will never be stable since its value will continue to fluctuate on a regular basis. In one sense, cryptocurrency is beyond the comprehension of any one person altogether.

The developer will be able to forecast the price of bitcoin over the next several hours or minutes with the use of machine learning and deep learning techniques. There has been a significant amount of research carried out, some of which was discussed in more depth in chapter 2 of this study. There have been a lot of projects much like this one done, and the author has both described the similarities and differences between them. All of the research that was identified uses machine learning and deep learning methods, such as LSTM, ARIMA, and Linear Regression principles. These algorithms are applied in the study. The majority of those already involved in the cryptocurrency market as purchasers or investors in bitcoin will benefit the most from this endeavour.

The chapter 3 of this paper has explained in detailed of what programming language is been used. Python is the chosen language and also the developer has listed out and compared between R, Python and SAS. The developer has also listed out the libraries that will be implemented in the project.

Although a great deal of research and development work has been done over the course of the past several years to enhance the effectiveness, dependability, and precision of prediction of bitcoin using a wide variety of machine learning and deep learning techniques, there is still room for additional development for a model that is more effective. The currently available models are trained using a small number of classes, making it for them to predict the price of bitcoin.

The purpose of the data, first data exploration, data visualization, and data cleansing, if necessary, are all covered in this chapter of the article, which is chapter 4. It has provided a full explanation of what the data is about. Following the completion of the data exploration, the developer will then discuss the model development process for both the LSTM model and the ARIMA model. In this chapter, we split up the data and develop the models, which results in the production of two graphs for each model: one depicting the open price, and the other depicting the close price.

In chapter 5, the results that were gained from the previous chapter are explained, and a comparison is made between the two distinct models and the results, which explain the MAPE value of each model and each graph that was formed. The developer has also included an explanation of the best model for predicting the price of bitcoin as well as future recommendations for the model to ensure it achieves optimal results.

So, far this project explains the deep concepts of LSTM and ARIMA model, and impact on time-series data. The developer also explains the prices in the world of cryptocurrency the advantages of the cryptocurrency in the world and the down sides of the cryptocurrency. This project first starts with the data import from

Kaggle which then the developer explores the data with basic python libraries such as mathlab and pandas. The developer has fullfed the main objective which is the prediction of the price and proper data visualization and data cleaning, if necessary, the developer also understands the data and splits the data according to the model to get better results from it. After data split, data normalization and model building the developer then fits the data into the model and to get the predicted graph. The developer then validates the model with mean squared error *(mse)* and then explains the best model for predicting the price of bitcoin. The developer has one limitation which is the deployment of the model. The model building is done with fixed conditions for this data set and data in it. If the data set keeps changing the conditions of each data set might be different, this is because the data of the bitcoin is volatile in nature as explained in the chapter 1.7. As bitcoin price fluctuates in nature every second the price I always unstable and the predicted value from the developer is not accurate and the developer also has no proof or evidence that the price predicted will be accurate after 30 days of prediction.

8.0 References

A. Ariyo, A. (2015). *Stock price prediction using the ARIMA model*. IEEE.

Asana. (2021, July 28). *Project Management Methodologies: 12 Popular Frameworks • Asana*. Asana. https://asana.com/resources/project-management-methodologies

Batey, N. (2021, August 10). *A Brief History of Bitcoin*. James Moore. https://www.jmco.com/brief-history-bitcoin/

Binance Academy. (2021). *All-Time High (ATH)*. Binance Academy. https://academy.binance.com/en/glossary/all-time-high

Bitcoin. (2009). *Bitcoin - Open source P2P money*. Bitcoin.org. https://bitcoin.org/en/

Bora, N. (2021, November 9). *Understanding ARIMA Models for Machine Learning*. Capital One. https://www.capitalone.com/tech/machine-learning/understanding-arima-models/

Brownlee, J. (2016, May 9). *Introduction to Python Deep Learning with Keras*. Machine Learning Mastery. https://machinelearningmastery.com/introduction-python-deep-learning-library-keras/

Brownlee, J. (2017, July 19). *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*. Machine Learning Mastery. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

Burns, E. (2019). *What Is Machine Learning and Why Is It Important?* SearchEnterpriseAI. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, *91*(4). https://doi.org/10.1103/revmodphys.91.045002

Chen, J. (2019). *Neural Network Definition*. Investopedia. https://www.investopedia.com/terms/n/neuralnetwork.asp

Denham, K., & Otte, J. (2022, March 14). Should I Invest In Bitcoin? *Www.thetimes.co.uk*. https://www.thetimes.co.uk/money-mentor/article/invest-bitcoin-cryptocurrencies/

El Naqa, I., & Murphy, M. J. (2015). What is machine learning? *Machine Learning in Radiation Oncology*, 3–11. https://doi.org/10.1007/978-3-319-18305-3_1

Erinfolami, K. (2021, November 28). *The 9 Biggest Risks for Crypto Investors (Both Beginners and Veterans)*. MUO. https://www.makeuseof.com/biggest-risks-crypto-investors/

Eyal, I., & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. *Financial Cryptography and Data Security*, 436–454. https://doi.org/10.1007/978-3-662-45472-5_28

Frankenfield, J. (2021, February 18). *Bitcoin*. Investopedia. https://www.investopedia.com/terms/b/bitcoin.asp

GeekforGeeks. (2019, January 18). *Best Python libraries for Machine Learning*. GeeksforGeeks. https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/

Gil, M. (2016, April 26). *Why use a project management methodology?* Nae.global. https://nae.global/en/why-use-a-project-management-methodology/

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*. Berlin, Heidelberg Springer Berlin Heidelberg.

Grinberg, R. (2012). Bitcoin: An innovative alternative digital currency. *Hastings Science and Technology Law Journal*, *4*(1).

GUPTA, M. (2021, April 19). *Bitcoin Price Prediction using RNN LSTM*. Www.youtube.com. https://www.youtube.com/watch?v=qpZ-dKPaj8M&list=LL&index=5&t=997s

Hamayel, M. J., & Owda, A. Y. (2021). A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. *AI*, *2*(4), 477–496. https://doi.org/10.3390/ai2040030

Hartill, R. (2021, November 10). *Adjusted closing price*. The Balance. https://www.thebalance.com/what-is-adjusted-closing-price-5190242

Ho, A., Vatambeti, R., & Ravichandran, S. K. (2021). Bitcoin Price Prediction Using Machine Learning and Artificial Neural Network Model. *Indian Journal of Science and Technology*, *14*(27), 2300–2308. https://doi.org/10.17485/ijst/v14i27.878

Hochreiter, S., & Schmidhuber, J. (1997). *Neural computation.* MIT Press.

Hua, Y., & Zhao, Z. (2019). *Deep learning with long short-term memory for time series prediction* (Vol. 57, pp. 114–119). IEEE Communications Magazine.

IBM. (2021, August 17). *CRISP-DM help overview*. Www.ibm.com. https://www.ibm.com/docs/en/spss-modeler/SaaS?topic=dm-crisp-help-overview

Intellipaat. (2022, February 5). *What is LSTM? Introduction to long short term memory*. https://intellipaat.com/blog/what-is-lstm/

ISIXSIGMA. (2020). *Tangible Benefits Definition*. ISixSigma. https://www.isixsigma.com/dictionary/tangible-benefits/

J Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.

Jain, K. (2015, January 5). *Scikit-Learn In Python - Important Machine Learning Tool*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/

Ji, S., Kim, J., & Im, H. (2019). A Comparative Study of Bitcoin Price Prediction Using Deep Learning. *Mathematics*, *7*(10), 898. https://doi.org/10.3390/math7100898

Jiang, X. (2020). Bitcoin Price Prediction Based on Deep Learning Methods. *Journal of Mathematical Finance*, *10*(01), 132–139. https://doi.org/10.4236/jmf.2020.101009

Karlijn Willems. (2019). *keras: Deep Learning in R*. DataCamp Community. https://www.datacamp.com/community/tutorials/keras-r-deep-learning

Kaspersky. (2019, January 11). *What is Cryptocurrency? Cryptocurrency Security: 4 Tips to Safely Invest in Cryptocurrency*. Www.kaspersky.com. https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency

Keras. (2019). *Home - Keras Documentation*. Keras.io. https://keras.io

Kuadli, J. (2021, February 1). *20 Most Fascinating Credit Card Fraud Statistics*. Legaljobs.io. https://legaljobs.io/blog/credit-card-fraud-statistics/

Kumar, S. (2021, June 23). *Why Python is Best for AI, ML, and Deep Learning*. RTInsights. https://www.rtinsights.com/why-python-is-best-for-ai-ml-and-deep-learning/

LearnCrypto. (2021). *Understanding Crypto Trading Volume & Its Significance*. Learncrypto.com. https://learncrypto.com/knowledge-base/how-to-trade-crypto/understanding-crypto-trading-volume

Mayer, Dr. C. (2021). *[NumPy vs Python] What are Advantages of NumPy Arrays over Regular Python Lists? – Finxter*. Finxter. https://blog.finxter.com/what-are-advantages-of-numpy-over-regular-python-lists/

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of Machine Learning, second edition. In *Google Books*. MIT Press. https://books.google.com.my/books?hl=en&lr=&id=dWB9DwAAQBAJ&oi=fnd&pg=PR5&dq=

what+is+machine+learning&ots=AysQZSx3l2&sig=EgM2ilBo0Oa_9-KXZb7n1ivA9OY&redir_esc=y#v=onepage&q=what%20is%20machine%20learning&f=false

Nibley, B. (2021, December 18). *What Is Volume in Cryptocurrency? An Overview*. SoFi. https://www.sofi.com/learn/content/what-is-volume-in-cryptocurrency/

Nyakundi, H. (2021, June 28). *What is an IDE in Programming? An IDE Definition for Developers*. FreeCodeCamp.org. https://www.freecodecamp.org/news/what-is-an-ide-in-programming-an-ide-definition-for-developers/

Pandas Org. (2020). *Package overview — pandas 1.1.5 documentation*. Pandas.pydata.org. https://pandas.pydata.org/docs/getting_started/overview.html

Perkel, J. M. (2018). Why Jupyter is data scientists' computational notebook of choice. *Nature*, *563*(7729), 145–146. https://doi.org/10.1038/d41586-018-07196-1

Polasik, M., Piotrowska, A., Wisniewski, T. P., Kotkowski, R., & Lightfoot, G. (2014). Price Fluctuations and the Use of Bitcoin: An Empirical Inquiry. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2516754

Python Institute. (2020). *About Python | Python Institute*. Python Institute. https://pythoninstitute.org/what-is-python/

R. (2019). *R: What is R?* R-Project.org. https://www.r-project.org/about.html

Reiff, N. (2019). *Why Bitcoin Price Predictions Are Unreliable*. Investopedia. https://www.investopedia.com/tech/why-bitcoin-price-predictions-are-unreliable/

Ritvik. (2021, December 26). *Using ARIMA to predict bitcoin prices in python in 2022*. Www.youtube.com. https://www.youtube.com/watch?v=pryXhOgDY9A&list=LL&index=1&t=868s

Rodrigues, I. (2020, February 20). *CRISP-DM methodology leader in data mining and big data*. Medium. https://towardsdatascience.com/crisp-dm-methodology-leader-in-data-mining-and-big-data-467efd3d3781

S.Gillis, A. (2021). *What is data splitting and why is it important?* SearchEnterpriseAI. https://www.techtarget.com/searchenterpriseai/definition/data-splitting

SAS. (2020). *System Requirements--SAS® Visual Analytics 7.3*. Support.sas.com. http://support.sas.com/documentation/installcenter/en/ikvisanlytofrsr/68904/HTML/default/index.html

Selig, J. (2020, May 6). *What is machine learning? A definition - expert system*. Expert.ai. https://www.expert.ai/blog/machine-learning-definition/

Sharp, T. (2020, May 6). *An Introduction to Support Vector Regression (SVR)*. Medium. https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

Ślepaczuk, R., & Zenkova, M. (2019). Robustness of Support Vector Machines in Algorithmic Trading on Cryptocurrency Market. *Central European Economic Journal*, *5*(1), 186–205. https://doi.org/10.1515/ceej-2018-0022

Sparkes, M. (2021). *What is bitcoin and how does it work?* New Scientist. https://www.newscientist.com/definition/bitcoin/

Spilka, D. (2021, August 12). *Why Are Bitcoin Prices So Volatile?* Kiplinger. https://www.kiplinger.com/investing/cryptocurrency/603280/why-are-bitcoin-prices-so-volatile

Team, G. L. (2021, May 28). *What is Numpy in Python - Everything you Need to Know About*. GreatLearning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/python-numpy-tutorial/

Thakur, M. (2019, May 20). Regression analysis formula | step by step calculation with examples. *Wallstreet Mojo*. https://www.wallstreetmojo.com/regression-analysis-formula/

Vanderplas, J. T. (2017). *Python data science handbook : essential tools for working with data*. O'reilly, Cop.

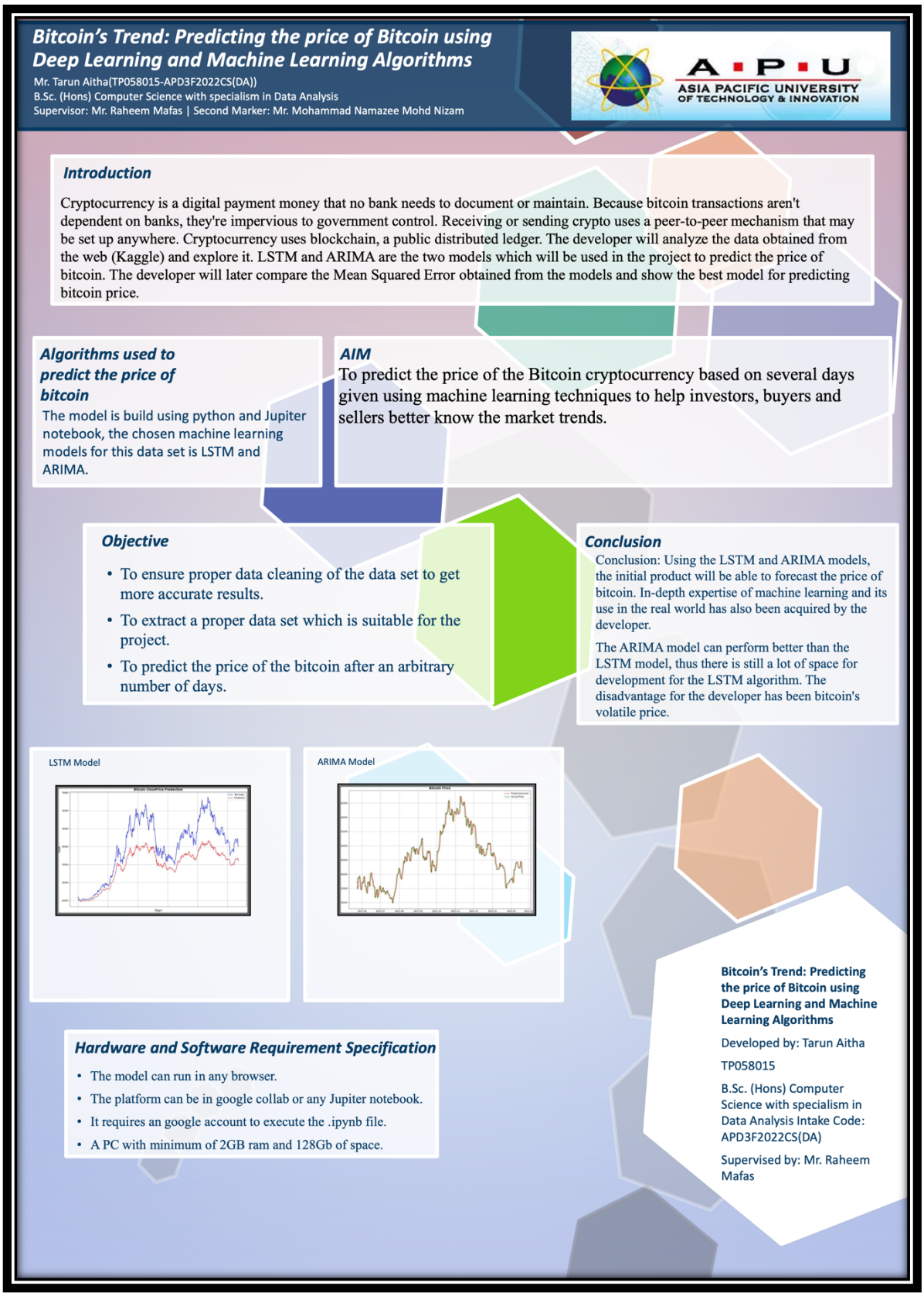
Velankar, S., Valecha, S., & Maji, S. (2018, February 1). *Bitcoin price prediction using machine learning*. IEEE Xplore. https://doi.org/10.23919/ICACT.2018.8323676

Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*. http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

Writers, S. (2021, November 2). *The 6 Easiest Programming Languages to Learn | BestColleges*. Www.bestcolleges.com. https://www.bestcolleges.com/bootcamps/guides/6-easiest-programming-languages-to-learn/

Appendices

FYP Poster



Project Log Sheets

Project Log Sheet 1

(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha…….... **Date: …16/02/2022…Meeting No: …1…...**

Project title: Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms. Intake: APD3F3022CS(DA)

Supervisor's name: ……Mr. Raheem Mafas **Supervisor's signature: ……..…**

Items for discussion (noted by student before mandatory supervisory meeting):

1. **FYP tittle**
2. **PSF initiation**

Record of discussion (noted by student during mandatory supervisory meeting):

1. **Plan a good title**
2. **Do more research**

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Plan a good title**
2. **Need to research more information**

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log sheet 2

(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha .... **Date: ……24/02/2022……Meeting No: …2…...**

Project title: Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms. Intake: APD3F2202CS(DA)

Supervisor's name: …Mr. Raheem Mafas ……. ….. **Supervisor's signature: ……...…**

Items for discussion (noted by student before mandatory supervisory meeting):

1. **PSF update**
2. **Initiation on PPF**
3. **Initiation on IR**

Record of discussion (noted by student during mandatory supervisory meeting):

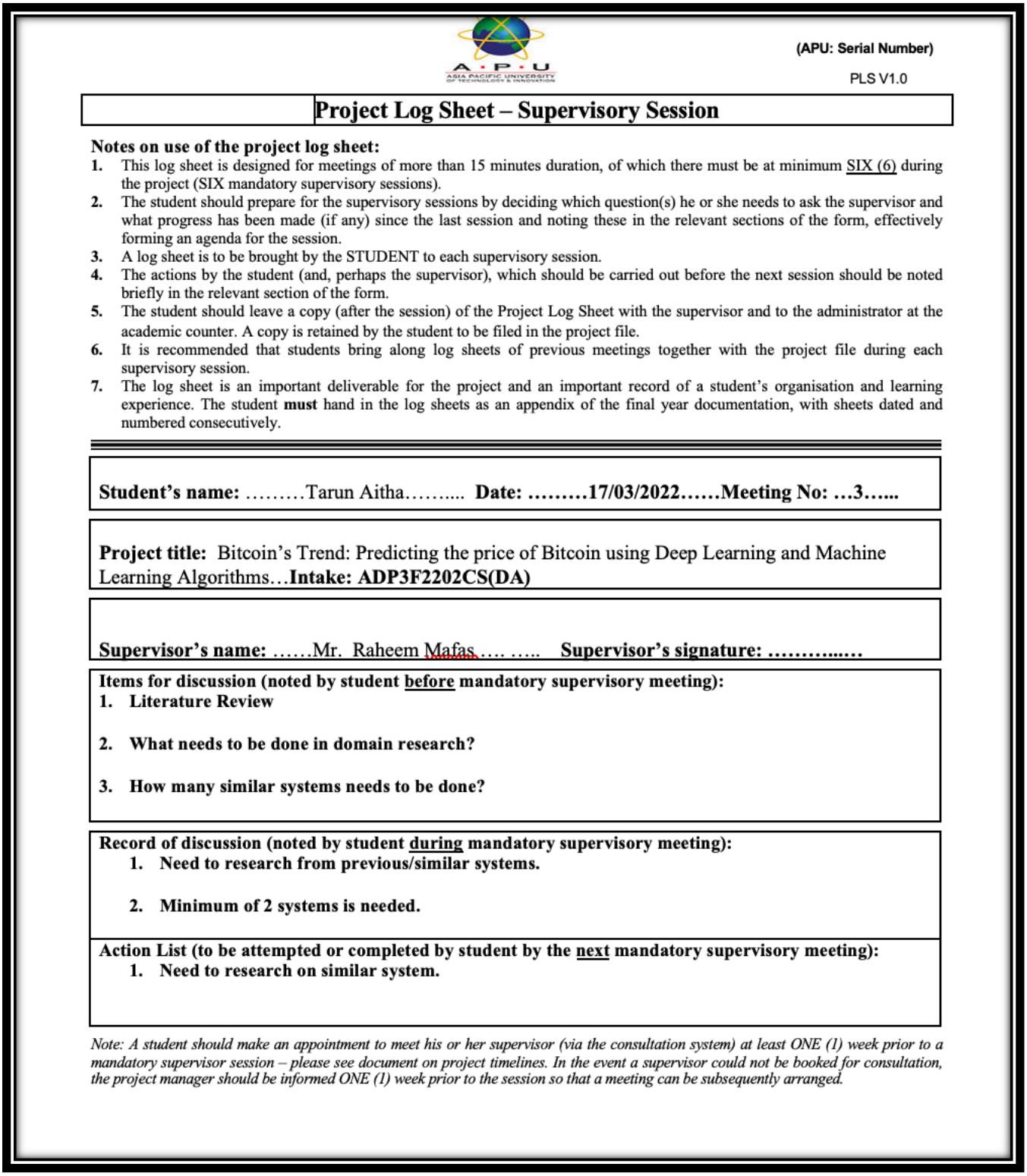
1. **Can improvise the Aim and Objective**
2. **A lot of research needs to be done**

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Need to improvise the PSF to submit the PPF**
2. **A lot of research needs to be done.**

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log Sheet 3



(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha…….... **Date: ……17/03/2022……Meeting No: …3…...**

Project title: Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms…**Intake: ADP3F2202CS(DA)**

Supervisor's name: ……Mr. Raheem Mafas…. ….. **Supervisor's signature: ……...…**

Items for discussion (noted by student before mandatory supervisory meeting):

1. **Literature Review**
2. **What needs to be done in domain research?**
3. **How many similar systems needs to be done?**

Record of discussion (noted by student during mandatory supervisory meeting):

1. **Need to research from previous/similar systems.**
2. **Minimum of 2 systems is needed.**

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Need to research on similar system.**

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log sheet 4

(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha ……..... **Date: ……10/04/2022……Meeting No: ……4.....**

Project title: Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms. Intake: APD3F2202CS(DA)

Supervisor's name: Mr. Raheem Mafas Supervisor's signature: ……..…

Items for discussion (noted by student before mandatory supervisory meeting):

1. **Technical research**
2. **Methodology**
3. **Survey**

Record of discussion (noted by student during mandatory supervisory meeting):

1. **What are the tools and software you are going to use in the project?**
2. **How and what methodology are you going to follow.**
3. **Survey is optional**

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Start working on technical research**
2. **Gather information methodology and technical research**

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log sheet 5

(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the course of the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session, and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha…..... **Date: ……6th July 2022Meeting No: 1/Semester 2……**

Project title: …… Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms……**Intake:** APD3F2022CS(DA)……

Supervisor's name: Raheem Mafas……..... **Supervisor's signature:** RMF

Items for discussion (noted by student before mandatory supervisory meeting):

1. **Data Analysis**
2. **Data exploration**
3. **Data cleaning**
4.

Record of discussion (noted by student during mandatory supervisory meeting):

1. **Needs to be done if necessary**
2.
3.
4.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Need to explore more on the data set.**
2.
3.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log Sheet

Project Log sheet 6

(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the course of the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session, and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha…..... **Date: ……24th August 2022Meeting No: 2/Semester 2……**

Project title: …… Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms……**Intake:** APD3F2022CS(DA)……

Supervisor's name: Raheem Mafas……..... **Supervisor's signature:** RMF

Items for discussion (noted by student before mandatory supervisory meeting):

1. **Data evaluation**
2. **Algorithms**

Record of discussion (noted by student during mandatory supervisory meeting):

1. **LSTM and ARIMA is most used one**
2.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Need to implement ARIMA and LSTM**

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log sheet 7

(APU: Serial Number)

PLS V1.0

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the course of the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session, and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: ……Tarun Aitha…..... **Date: ……14th September 2022Meeting No: 3/Semester 2……**

Project title: …… Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms……**Intake:** APD3F2022CS(DA)……

Supervisor's name: Raheem Mafas……..... **Supervisor's signature:** RMF

Items for discussion (noted by student before mandatory supervisory meeting):

1. **Conclusion**
2. **Any more changed or improvements in the code**
3. **Deployment**

Record of discussion (noted by student during mandatory supervisory meeting):

1. **Further improves can be done**
2. **Need to give reason why deployment is not necessary for the model.**

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. **Deployment of the model is not necessary**

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project proposal Form (PPF)

PPF:

Aim:

Bitcoin Price Prediction Using Machine Learning.

Introduction:

Bitcoin, also known as Cryptocurrency, is a digital currency that does not have a central bank or administration and may be exchanged digitally between individuals. When fresh bitcoins are produced and given to the individuals who administer the computer system and assist verify transactions, the process is referred to as mining. Satoshi Nakamoto invented Bitcoin in 2008 and made it available to the for the first time in 2009. According to the users computers hardware, it may take anywhere from one to 30 days to mine a single bitcoin. To put it simply, its made by solving basic arithmetic and computer problems in a process known as mining. Bitcoins price will be accurately predicted using a neural network.

Problem Statement:

Bitcoin is the most complicated cryptocurrency, with its value fluctuating from time to time. Investing in Bitcoin is hazardous, and individuals earn less money. Bitcoin is more volatile because of supply and demand. Bitcoin prices rise and fall based on how many bitcoins are in circulation and how many people are willing to pay. Bitcoin is also dependent on its investor's bitcoin values rise in proportion to the number of bitcoins in circulation. According to a National Bureau of Economic Growth survey, the top 10,000 investors owned about one-third of all bitcoins. Bitcoin is never steady it is always volatile, and it is always perilous for investors to invest if the forecast of bitcoins is inaccurate.

Project Aim:

To predict the price of the Bitcoin cryptocurrency based on several days given using machine learning techniques to help investors, buyers and sellers better know the market trends.

Project Objectives:

To predict the price of the bitcoin after an arbitrary number of days.
To visualize the predicted price and compare it with the previous price.
To ensure proper data cleaning of the data set to get more accurate results.

Literature Review (Optional):

Deliverables:

A proper price prediction of the bitcoin after an arbitrary number of days.
A clear comparison of the predicting prize for the bitcoin by visualization.

Propose at least 5 to 7 names of supervisors:

Mr. RAHEEM MAFAS
Mr. MOHAMMAD NAMAZEE
DR. PREETHI SUBRAMANIAN
DR. BEHRANG SAMADI
DHASON PADMA KUMAR

Project Specification Form (PSF)

PSF:

Title: Using machine learning to forecast the price of Bitcoin

Project background.

Bitcoin, often known as cryptocurrency, is a digital currency that does not have a central bank or an administration. It may be transmitted digitally between users over a peer-to-peer link. Bitcoin mining is a method in which new bitcoins are mined and delivered to those who are rewarded, lucrative, and have a high income and manage the computer system and assist to verify transactions. Satoshi Nakamoto designed Bitcoin in 2008, and it was first made available to the public in 2009. It will take roughly a day to manufacture a single bitcoin, depending on the user's computer hardware. It may potentially take around 30 days to mine a single bitcoin. It is generated via a decentralized and competitive process known as mining, which is accomplished by solving basic mathematics and computing challenges. An appropriate neural network will be used, and an accurate price forecast for bitcoin will be provided.

Problem Statement:

Bitcoin is the most sophisticated cryptocurrency, and its value fluctuates. Investing in Bitcoin is riskier, and individuals earn less money from it. Bitcoin is more volatile because of supply and demand. Bitcoin prices rise and fall based on how many bitcoins are in circulation and how many people are willing to pay. Bitcoin is also influenced by its investors; the price of bitcoin rises in proportion to the quantity of bitcoins in circulation. According to research by the National Bureau of Economic Growth, the top 10,000 investors owned about one-third of all bitcoins. Bitcoin is not stable; it will constantly fluctuate, and it is always dangerous for investors to invest if the forecast of bitcoins is not right.

Description of problem area:

Bitcoin is a decentralized digital currency that is the most popular means to earn and invest in current age. Cryptocurrency is an interest in the need for dependable and long-term investment. Cryptocurrency, unlike flat money, has a finite quantity and is created by solving basic mathematical procedures, making the cryptocurrency untouchable by any authority. When you purchase a bitcoin, you may trade, invest, buy, and sell it at any moment. Bitcoin values are volatile, and they are determined by supply and demand.

Nature of challenge

Cryptocurrency is a great location to earn money and invest, but it is also unsafe for many individuals since bitcoin values are always fluctuating. It is impossible to anticipate how much it will be in the next days.

The biggest issue encountered throughout this research is knowing bitcoin pricing and current market trends. Bitcoin is very volatile, and the precise price cannot be anticipated after a certain number of random days; it is just forecasted without any proof. It will also be more difficult to locate an appropriate data set.

There are several machine learning approaches, and not all of them are ideal for price prediction. Choosing a suitable machine learning approach is also one of the most difficult aspects of this endeavor.

Rationale:

As mentioned in the issue statement, investors, buyers, and sellers are continually at a loss owing to the volatility of Bitcoin's price. Predictions like this one will give investors and buyers a little more confidence since the system will predict the price of bitcoin after a specific amount of time has passed.

Description of project objectives:

Aim:

To predict the price of the Bitcoin cryptocurrency based on several days given using machine learning techniques to help investors, buyers and sellers better know the market trends.

Objectives:

To predict the price of the bitcoin after an arbitrary number of days.

To visualize the predicted price and compare it with the previous price.

To ensure proper data cleaning of the data set to get more accurate results.

Resources needed by the proposal

8Gb of RAM and a minimum of 128GB of free space of any Laptop or desktop will do for this project along with a mouse and a keyboard.

Python, Google Collab, SAS Enterprise miner, Visual Studio Code, Jupiter Notebook, and Pandas.

Academic research being carried out:

The data set which is going to be used is the Bicoin.csv obtained from Kaggle.com.,

This data set contains

- The initial release date of the individual bitcoin.
- The price of the individual bitcoin when it was released.
- The highest price of the individual bitcoin
- The lowest price of the individual bitcoin.
- The closed price of the individual bitcoin.

Development plan for the proposed project

Crisp-DM Methodology

The CRISP-DM (Cross-Industry Standard Process for Data Mining) technique is a tried-and-true way for leading your data mining operations. It includes definitions of common project stages, activities connected with each phase, and a methodology for explaining the links between these tasks.

The data mining process is separated into smaller tasks at the first level. Each job is made up of several second level generic tasks. It is named generic tasks because it is designed to cover all the processes in data mining; they are produced for the model's stability and completion.

Data mining has six stages in all.

Business Understanding

The first process is to understand the project's requirements. The requirements are then turned into data mining challenges, and a preliminary project is built to meet the project objectives.

Data Understanding

This phase begins with data collecting and gathering. Once the data is collected, we assess the data quality and uncover data insights.

Data Preparation

In this step, all data cleaning for modelling is done based on the data insights gleaned through data comprehension. The data is prepared for the following step so that correct findings may be obtained.

Modeling

In this phase various modeling techniques are applied to the data and the best data model is chosen for the data.

Evaluation

From the perspective of a data analyst, the models are evaluated at this step to determine which are of the greatest quality. Using the business goals and objectives as a guideline, the models are thoroughly examined before moving on to the next phase.

Deployment

The knowledge will be retrieved and arranged by the presenter so that it can be used by the client after the model has been deployed. Depending on the needs, the model may be published, or the same data mining process can be performed several times.

Fast Track Form

Office Record	Receipt – Fast-Track Ethical Approval
Date Received:	Student name: Tarun Aitha Student number: TP058015
Received by whom:	Received by: Date:

APU / APIIT FAST-TRACK ETHICAL APPROVAL FORM (STUDENTS)

Tick one box (level of study):

☐ POSTGRADUATE (PhD / MPhil / Masters)
☐ UNDERGRADUATE (Bachelor's degree)
☐ FOUNDATION / DIPLOMA / Other categories

Tick one box (purpose of approval):

☐ ~~Thesis / Dissertation~~ / FYP project
☐ Module assignment
☐ Other: _____

Title of Programme on which enrolled …

Tick one box: ☐✓ Full-Time Study or ☐ Part-Time Study

Title of project / assignment: Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms

Name of student researcher: Mr. Raheem Mafas

Student Researchers- please note that certain professional organisations have ethical guidelines that you may need to consult when completing this form.

Supervisors/Module Lecturers - please seek guidance from the Chair of the APU Research Ethics Committee if you are uncertain about any ethical issue arising from this application.

		YES	NO	N/A
1	Will you describe the main procedures to participants in advance, so that they are informed about what to expect?			✓
2	Will you tell participants that their participation is voluntary?			✓
3	Will you obtain written consent for participation?			✓
4	If the research is observational, will you ask participants for their consent to being observed?			✓
5	Will you tell participants that they may withdraw from the research at any time and for any reason?			✓
6	With questionnaires and interviews will you give participants the option of omitting questions they do not want to answer?			✓
7	Will you tell participants that their data will be treated with full confidentiality and that, if published, it will not be identifiable as theirs?			✓
8	Will you give participants the opportunity to be debriefed i.e. to find out more about the study and its results?			✓

If you have ticked **No** to any of Q1-8 you should complete the full Ethics Approval Form.

		YES	NO	N/A
9	Will your project/assignment deliberately mislead participants in any way?			✓
10	Is there any realistic risk of any participants experiencing either physical or psychological distress or discomfort?			✓
11	Is the nature of the research such that contentious or sensitive issues might be involved?		✓	

If you have ticked **Yes** to 9, 10 or 11 you should complete the full Ethics Approval Form. In relation to question 10 this should include details of what you will tell participants to do if they should experience any problems (e.g. who they can contact for help). You may also need to consider risk assessment issues.

			YES	NO	N/A
12	Does your project/assignment involve work with animals?			✓	
13	Do participants fall into any of the following special groups? **Note that you may also need to obtain satisfactory clearance from the relevant authorities**	Children (under 18 years of age)			✓
		People with communication or learning difficulties			
		Patients			
		People in custody			
		People who could be regarded as vulnerable			
		People engaged in illegal activities (eg drug taking)			
14	Does the project/assignment involve external funding or external collaboration where the funding body or external collaborative partner requires the University to provide evidence that the project/assignment had been subject to ethical scrutiny?			✓	

If you have ticked **Yes** to 12, 13 or 14 you should complete the full Ethics Approval Form. There is an obligation on student and supervisor to bring to the attention of the APU Research Ethics Committee any issues with ethical implications not clearly covered by the above checklist.

STUDENT RESEARCHER

Provide in the boxes below (plus any other appended details) information required in support of your application, THEN SIGN THE FORM.

	Please Tick Boxes
I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU Research Ethics Committee.	
Give a brief description of participants and procedure (methods, tests used etc) in up to 150 words. The data set is obtained from Kaggle. Link for the dataset(https://www.kaggle.com/datasets/varpit94/bitcoin-data-updated-till-26jun2021) All the images are taken from open source (Google.com).	
I also confirm that: i) All key documents e.g. consent form, information sheet, questionnaire/interview are appended to this application. Or	
ii) Any key documents e.g. consent form, information sheet, questionnaire/interview schedules which need to be finalised following initial investigations will be submitted for approval by the project/assignment supervisor/module lecturer before they are used in primary data collection.	✓

E-signature… … … … … Tarun Aitha… … … … … … Date… … 27/05/2022 …
(Student Researcher)

Please note that any variation to that contained within this document that in any way affects ethical issues of the stated research requires the appending of new ethical details. New ethical consent may need to be sought.
The completed form (and any attachments) should be submitted for consideration by your Supervisor/Module Lecturer

SUPERVISOR/MODULE LECTURER
PLEASE CONFIRM THE FOLLOWING:

	Please Tick Box
I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU Research Ethics Committee	✓
i) I have checked and approved the key documents required for this proposal (e.g. consent form, information sheet, questionnaire, interview schedule) Or	
ii) I have checked and approved draft documents required for this proposal which provide a basis for the preliminary investigations which will inform the main research study. I have informed the student researcher that finalised and additional documents (e.g. consent form, information sheet, questionnaire, interview schedule) must be submitted for approval by me before they are used for primary data collection.	

SUPERVISOR AND SECOND ACADEMIC SIGNATORY

STATEMENT OF ETHICAL APPROVAL (please delete as appropriate)

1) **THIS PROJECT/ASSIGNMENT HAS BEEN CONSIDERED USING AGREED APU/SU PROCEDURES AND IS NOW APPROVED**

2) **THIS PROJECT/ASSIGNMENT HAS BEEN APPROVED IN PRINCIPLE AS INVOLVING NO SIGNIFICANT ETHICAL IMPLICATIONS, BUT FINAL APPROVAL FOR DATA COLLECTION IS SUBJECT TO THE SUBMISSION OF KEY DOCUMENTS FOR APPROVAL BY SUPERVISOR (see Appendix A)**

RMF Mafas Raheem 28.05.2022
E-signature... ... ... Print Name... ... ... Date... ... ...
(Supervisor/Lecturer)

E-signature... ... ... Print Name... ... ... Date... ... ...
(Second Academic Signatory)

Office Record	Receipt – Appendix A (Fast-Track Ethics Form)
Date Received:	Student name: Student number:
Received by whom:	Received by: Date:

APPENDIX A
AUTHORISATION FOR USE OF KEY DOCUMENTS

Completion of Appendix A is required when for good reasons key documents are not available when a fast track application is approved by the supervisor/module lecturer and second academic signatory.

I have now checked and approved all the key documents associated with this proposal e.g. consent form, information sheet, questionnaire, interview schedule

Title of project/assignment: Bitcoin's Trend: Predicting the price of Bitcoin using Deep Learning and Machine Learning Algorithms

Name of student researcher … … … … … … … … …Tarun Aitha … … … … … … … … … …

Student ID: ……TP058015…… Intake: … … … APD3F2202CS(DA)… … …

E-signature… … … … … … … … Print Name… Mafas Raheem … Date… … …… …
(Supervisor/Lecturer)

Gannt Chart For Final Year Rpeort

Title	Start Day	End Day	Status
Project Proposal Form (PPF)	14/02/2022	18/02/2022	Done
Project Specification Form (PSF)	19/02/2022	28/02/2022	Done
Chapter 1: Introduction to the study	01/03/2022	25/03/2022	Done
Chapter 2: Literature Review	26/03/2022	30/04/2022	Done
Chapter 3: Technical Research	01/05/2022	07/05/2022	Done
Chapter 4: Methodology	08/05/2022	19/06/2022	Done
Chapter 5: Data Analysis	20/06/2022	09/08/2022	Done
Chapter 6: Results and Discussion	10/08/2022	02/09/2022	Done
Chapter 7: Conclusions and Reflections	03/09/2022	20/09/2022	Done