PREDICTIVE ANALYSIS OF CRYPTOCURRENCY PRICES

a project report submitted by

PRIYA SHAJI (REG. NO: UR14CS092)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Mrs. A. SHAMILA EBENEZER, M.E., (Ph.D.)

DEPARTMENT OF COMPUTER SCIENCES TECHNOLOGY

SCHOOL OF ENGINEERING AND TECHNOLOGY

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-University under Sec-3 of the UGC Act, 1956)
Karunya Nagar, Coimbatore - 641 114, India.

APRIL 2018

BONAFIDE CERTIFICATE

Certified that this project report **"PREDICTIVE ANALYSIS OF CRYPTOCURRENCY PRICES"** is the bonafide work of "PRIYA SHAJI (REG. NO: UR14CS092)**"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. E. Kirubakaran	**Mrs. A. Shamila Ebenezer**
Head of the Department Department of Computer Science and Technology	**Supervisor** Assistant Professor Department of Computer Sciences Technology

Submitted for the Project Viva Voce held on……………………….

Examiner

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(DEEMED TO BE UNIVERSITY)
A CHRISTIAN MINORITY INSTITUTION
ACCREDITED BY NAAC

SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCES TECHNOLOGY
PROGRAM OF COMPUTER SCIENCE AND ENGINEERING

KU/CST/CERTIFICATE/2018/047 April 17, 2018

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the following student carried out the project work titled **"PREDICTIVE ANALYSIS OF CRYPTOCURRENCY PRICES"** from November 30, 2017 to April 17, 2018 in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Computer Science and Engineering of Karunya Institute of Technology and Sciences.

Ms. PRIYA SHAJI [UR14CS092]

The department wishes her all the best.

Project Supervisor **Project Technical Coordinator** **HoD/CST**

Karunya Nagar, Coimbatore - 641 114, Tamil Nadu, India. +91-422-2614300 +91-422-2615615 ku@karunya.edu www.karunya.edu

ACKNOWLEDGEMENT

First and foremost, we praise and thank **ALMIGHTY GOD** for giving us the will power and confidence to carry out our project.

I am grateful to our beloved founders Late **Dr. D.G.S. Dhinakaran**, **C.A.I.I.B, Ph.D.,** and **Dr. Paul Dhinakaran**, **M.B.A., Ph.D.,** for their love and always remembering us in their prayers.

I extend my thanks to **Dr. P. Mannar Jawahar, Ph.D.,** our honorable vice chancellor, **Dr. Ridling Margaret Waller, Ph.D.,** our honorable Pro-Vice Chancellor and **Dr. R. Elijah Blessing, Ph.D.,** our respected registrar for giving us the opportunity to carry out this project.

I am thankful to **Dr. G. Prince Arulraj, M.E., Ph.D.,** Dean (Engineering & Technology) for his support and encouragement.

I would like to place my heart-felt thanks and gratitude to **Dr. E. Kirubakaran, Ph.D.,** HOD, Department of Computer Sciences Technology for his encouragement and guidance.

I extend my thanks to **Dr. J. Immanuel John Raja, M.E., Ph.D.,** Program Coordinator-UG, Computer Science and Engineering for his encouragement and guidance.

I am grateful to my guide, **Mrs. A. Shamila Ebenezer, M.E., (Ph.D.,)** Assistant Professor, Department of Computer Sciences Technology for his valuable support, advice and encouragement.

I also thank all the staff members of the Department for extending their helping hands to make this project work a success.

I would also like to thank all my friends and my parents who have prayed and helped me during the project work.

ABSTRACT

Today cryptocurrencies have become a global phenomenon known to most people. The main objective of the paper is to analyze the characteristics and features that affects the fluctuations i.e. highs and lows of cryptocurrency prices and to implement the predictive analysis of the same using machine learning techniques. Fifteen different cryptocurrency datasets have been analyzed using candlestick charts, correlation graphs and price versus date graph using matplotlib package. The output would be to analyze with what accuracy can the cryptocurrency price in USD can be predicted. The cryptocurrency price data is sourced from the kaggle dataset, coinmarketcap and blockchain info.

ARIMA package have been used to build the model and train on historical data. The popular ARIMA model for time series forecasting is implemented for fifteen different cryptocurrency datasets and also plotted future fluctuation graphs for them using the forecasted datasets which was generated using Facebook's package Prophet(). As expected the ARIMA forecast outperforms with the highest accuracy of 98% for bitcoin. Using these results a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ABBREVIATION	**MEANING**
BTC	Bitcoin
ETH	Ethereum
IBM	International Business Machines

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

Today cryptocurrencies have become a global phenomenon known to most people. The main objective of the paper is to analyze the characteristics and features that affects the fluctuations i.e. highs and lows of cryptocurrency prices and to implement the predictive analysis of the same using machine learning techniques. Fifteen different cryptocurrency datasets has to be analyzed using three different analysing techniques i.e. candlestick charts, correlation graphs and plotting price versus date graph using matplotlib package. The output would be to analyze with what accuracy can the cryptocurrency price in USD can be predicted. The cryptocurrency price data is sourced from the kaggle dataset, coinmarketcap and blockchain info.

ARIMA package will be used to build the model and train on historical data. The popular ARIMA model for time series forecasting will be implemented for fifteen different cryptocurrency datasets and will plot the future fluctuation graphs for them using the forecasted datasets which was generated using Facebook's package Prophet(). For the implementation of the results, a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

1.2 PROBLEM STATEMENT

Cryptocurrency is a highly volatile market. The main problem statement of predictive analysis of cryptocurrency prices is to analyze the factors on which the currency price fluctuations depends and how accurately the cryptocurrency price in USD can be predicted using time series model, also, to provide an interactive platform where the investors can know about the present, past and future prices of cryptocurrencies which would indeed help them to make better investment.

1.3 CHAPTER WISE SUMMARY

In chapter 1, the objective of the project and it's problem statement is discussed.

In chapter 2, the project "System Analysis" is explained which includes existing system, proposed system, usecase analysis and requirement specification which further includes functional requirements, non-functional requirements, hardware requirements, software requirements.

In chapter 3, project "System Design" is explained which includes detailed design, design of methodology, modules, database design which further includes entity relationship diagram , tables or entity.

In chapter 4. Project "System Implementation" is explained which includes module implementation and testing

In chapter 5, "Conclusion and Future Scope" of the project is discussed.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

An existing system of prediction of bitcoin price using machine learning techniques is implemented. The historical data for bitcoin is used for implementation .The method is intended to predict fluctuations in cryptocurrencies based on the attributes of online communities. It analyzes user comments on online cryptocurrency communities, and conducts an association analysis between these comments and fluctuations in the price and number of transactions of cryptocurrencies to extract significant factors and formulate a prediction model.

2.2 PROPOSED SYSTEM

Forecasting is implemented for fifteen different cryptocurrency datasets which are analyzed using candlestick charts, correlation graphs and price versus date graph using matplotlib package. ARIMA package is used to build a model and train the historical data. Facebook's package prophet() is used to generate the forecasted datasets. Using these results a cryptocurrency chatbot is to be implemented using IBM Watson Assistant, an interactive platform to know the present, past and future values which would indeed benefit investors in future.

2.3 USECASE ANALYSIS

Use cases are used to capture dynamic behavior of the system. It defines interaction between roles(actors and their tasks).In use case diagram, four actors are used namely data analyst, programmer, prediction system and investors. The actors perform the respective tasks as mentioned in the diagram.

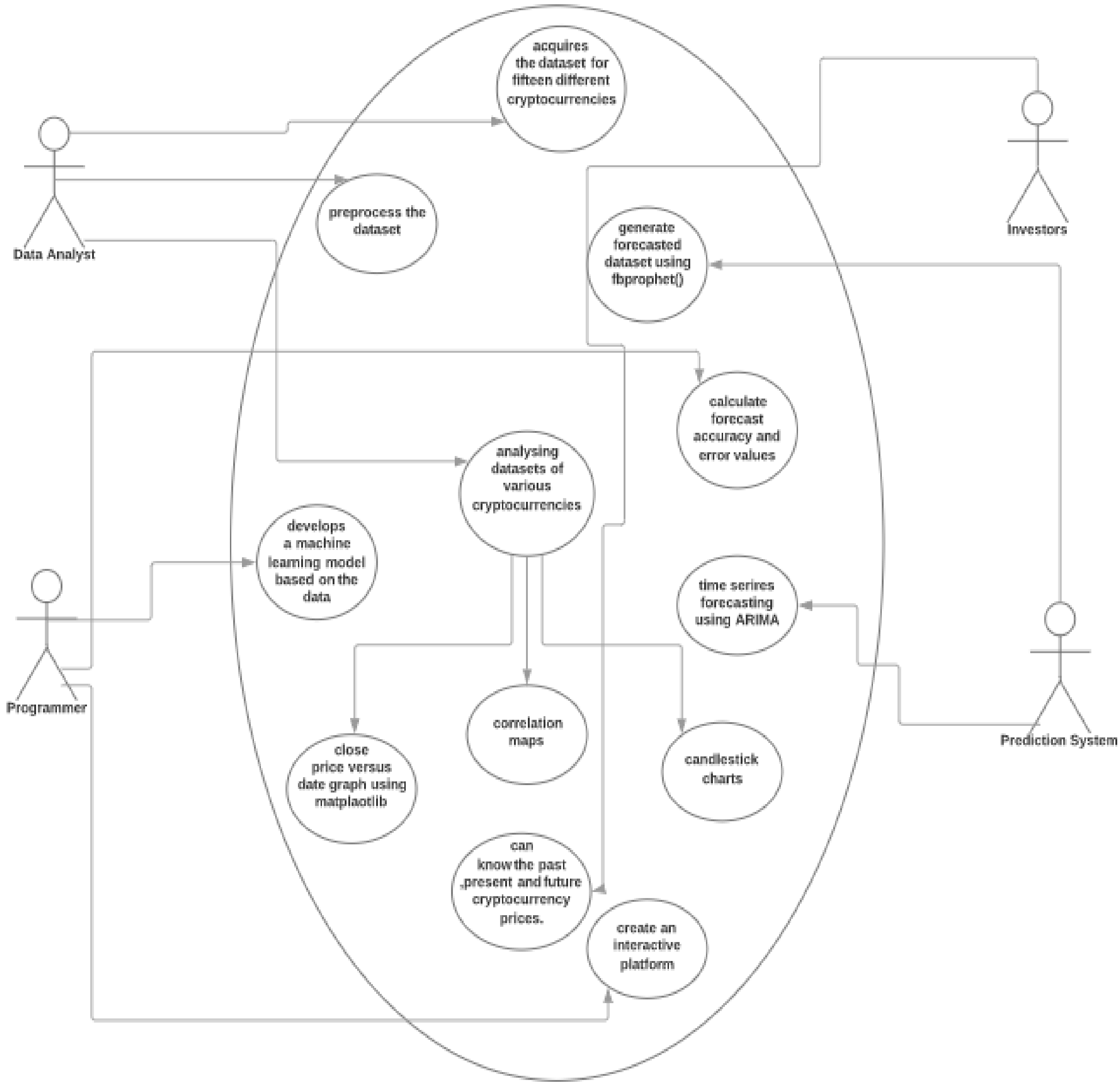


Fig 2.1 Use Case Diagram For Predictive Analysis Of Cryptocurrency Prices

2.4 REQUIREMENT SPECIFICATION

2.4.1 Functional Requirements

The software used is dynamic and can adapt to the different codes and functionalities given. The project is capable of taking user input for the constraints that are considered for cryptocurrency price prediction. ARIMA package have been used to build the model and train on historical data. The popular ARIMA model for time series forecasting is implemented for fifteen different cryptocurrency datasets .Time series model is used to define the accuracy of the forecasted values. Facebook's package prophet() is used to generate the forecasted dataset for fifteen different cryptocurrencies. Using IBM Watson Assistant , a cryptocurrency chatbot is implemented which is trained according to the predicted responses of the investors and common users. The chatbot is trained for the prices of fifteen different cryptocurrency datasets. The bot distinguishes between the past, present and future prices of the different cryptocurrencies by identifying the intents through which the bot is trained and responds to the inputs of investors according to their queries.

2.4.2 Non-Functional Requirements

Cryptocurrencies have become a global phenomenon known to most people. While still somehow not understood by most of them, example: banks, governments and many companies are unaware of its importance. The project focuses on analyzing the factors affecting the price fluctuations of various cryptocurrencies and using these factors to predict the future fluctuations of the prices and calculate forecast accuracy and error values. Using these results a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies which increases it's efficiency. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

2.4.3 Hardware Requirements

Anaconda3-5.0.1-Windows-x86_64 tool runs on all computers of following os i.e. Linux, Mac OS or Microsoft Windows. Hardware refers to physical components of a computer. The minimum hardware requirements needed are:

- RAM : 4GB RAM or more
- Processor :Intel Corei3 or faster

2.4.4 Software Requirements

The software used is dynamic and can adapt to the different codes and functionalities given.Anaconda 3-5.0.1 for windows-x86_64 is used in order to implement the coding part of the project , which is written in python language using the platform python-3.6.3-amd64. The software used for code implementation is jupyter notebook. In order to implement the cryptocurrency chatbot in the anaconda command line, cf_installer is installed along with softwares like git-2.16.2-64-bit and visualcppbuildtools_full..

The software requirements needed for the implementation of this project are:

- Anaconda3-5.0.1-Windows-x86_64
- python-3.6.3-amd64
- cf_installer
- Git-2.16.2-64-bit
- visualcppbuildtools_full
- IBM Watson Assistant

CHAPTER 3

SYSTEM DESIGN

3.1 DETAILED DESIGN

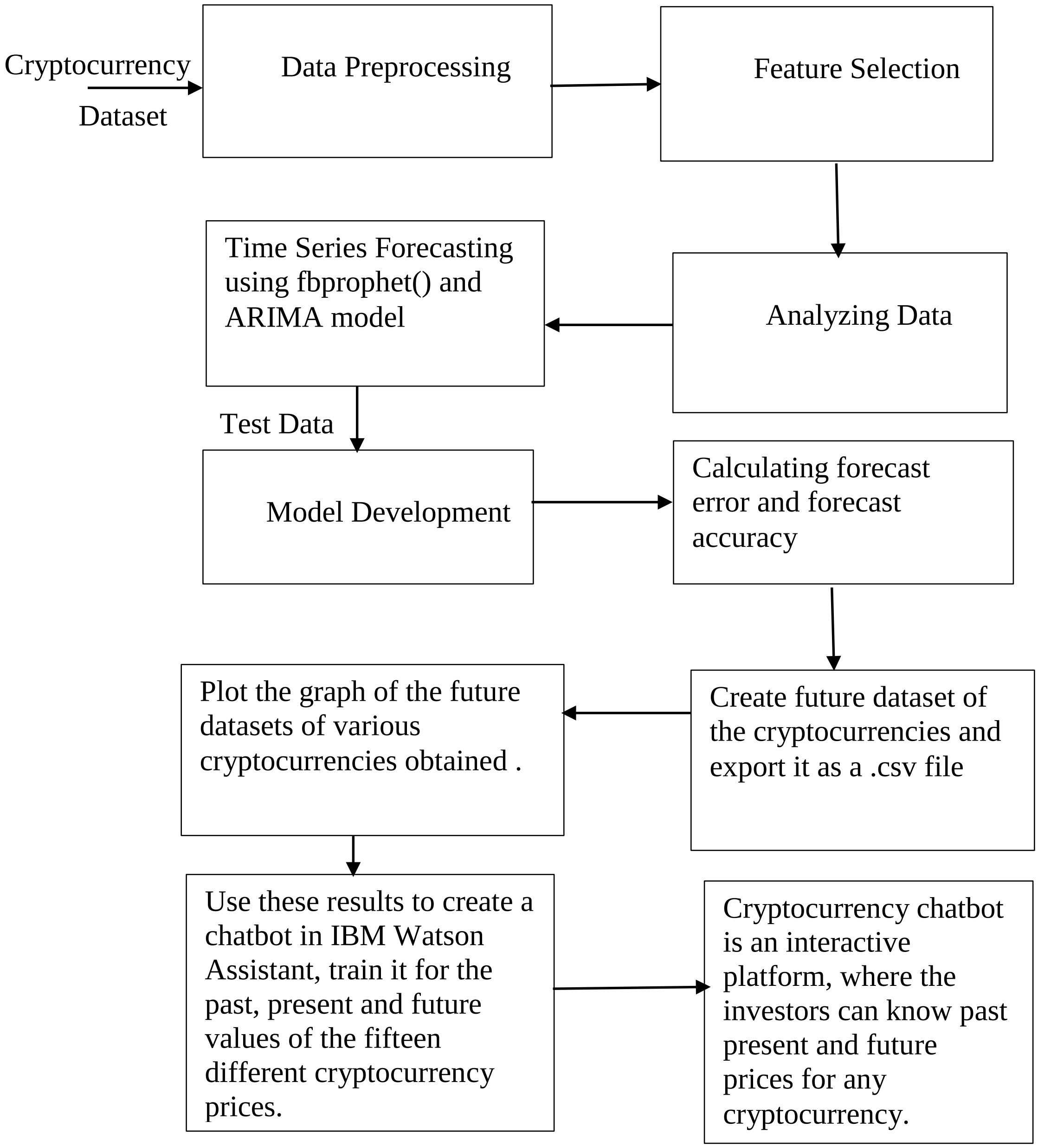


Fig 3.1 Architecture Diagram for Predictive Analysis Of Cryptocurrency Analysis

Description

According to the architecture, the dataset of fifteen different cryptocurrency prices is pre-processed into a more user understandable format. It includes steps like data cleaning, data integration, data reduction. The next step is feature selection refers to the process of extracting useful information or features from existing data. The datasets are then analyzed depending on the feature selected and the output results in the factor which fluctuates the values of cryptocurrency price i.e. close price. For the following steps the aim is to analyze with what accuracy can the cryptocurrency price in USD be predicted. ARIMA package have been used to build the model and train on historical data. The popular ARIMA model for time series forecasting is implemented for fifteen different cryptocurrency datasets and also plotted future fluctuation graphs for them using the forecasted datasets which was generated using Facebook's package Prophet().Forecast accuracy and error values are calculated. Using these results a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

3.2 DESIGN OF METHODOLOGY

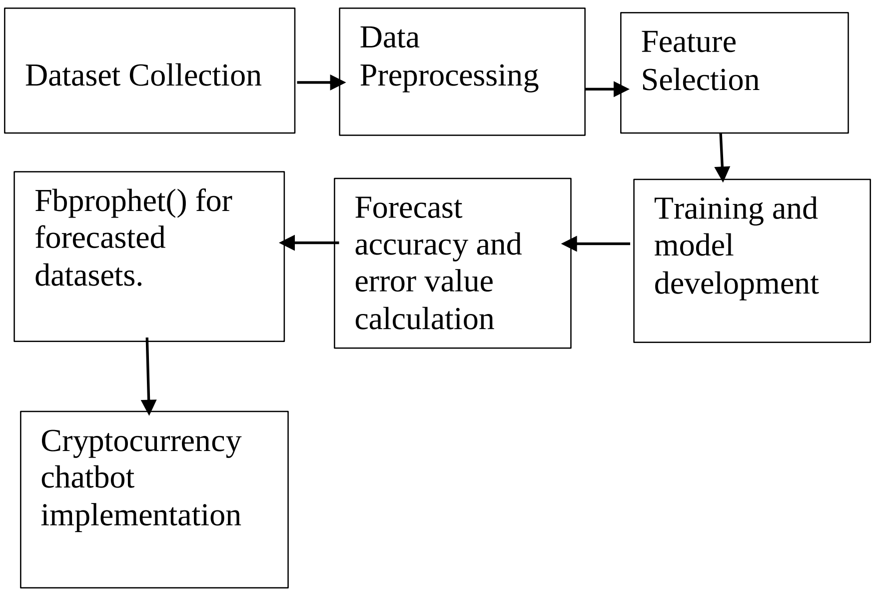


Fig 3.2 Design Of Methodology For Predictive Analysis Of Cryptocurrency Prices

- Dataset Collection

  Datasets for various cryptocurrencies were collected from three main sources

Kaggle Dataset, CoinmarketCap, Blockchaininfo

- Data Preprocessing

  The dataset considered is pre-processed into a more user understandable format. The null values were removed from the dataset .The redundant values were also removed from the dataset.It reduces the volume but produces the same or similar analytical results and increases the efficiency of the forecast

- Feature Selection

  Feature selection refers to the process of extracting useful information or features from existing data. The features which would prove useful for the forecast.

The features selected are as follows:

Date : date of observation

Open : Opening price on the given day

High : Highest price on the given day

Low : Lowest price on the given day

Close : Closing price on the given day

- Training and Model development

  The data is collected and a Time Series model is developed.

The Time Series models used are:

ARIMA Model

Call the ARIMA() function and pass the required p,q,d values: ARIMA(df_bitcoin, order=(5,1,0))

Fit the model: model.fit(disp=0)

Call the summary() function for obtaining the ARIMA model results

Divide the dataset into test and training sets: train, test = X[0:size], X[size:len(X)]

Calculate the forecast error and forecast accuracy

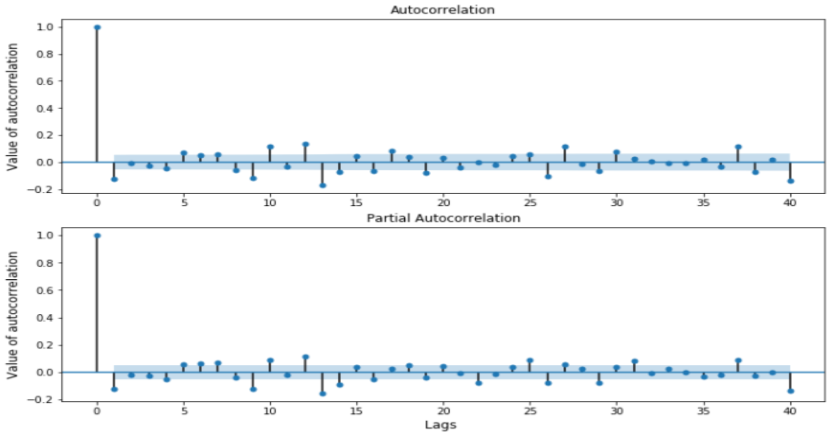


Fig 3.3 ACF And PACF Graph Of A Cryptocurrency To Calculate AR, I ,MA Values

- Forecast Error and Forecast Accuracy Calculation Equation

$$ForecastError = \left(|\,Actual - Forecast\,| \big/ Actual \right) * 100$$

$$ForecastAccuracy = Maximum(0, 100\% - ForecastError)$$

- Fbprophet()

Selecting ds and y: The prophet model requires only two columns in its dataframe.

ds: Datetime column and y

Calling the Prophet() function

Fitting the model: model = Prophet()

Making future dataframe: future model.make_future_dataframe(periods = 60)

Predicting future values by calling the predict() function: model.predict(future)

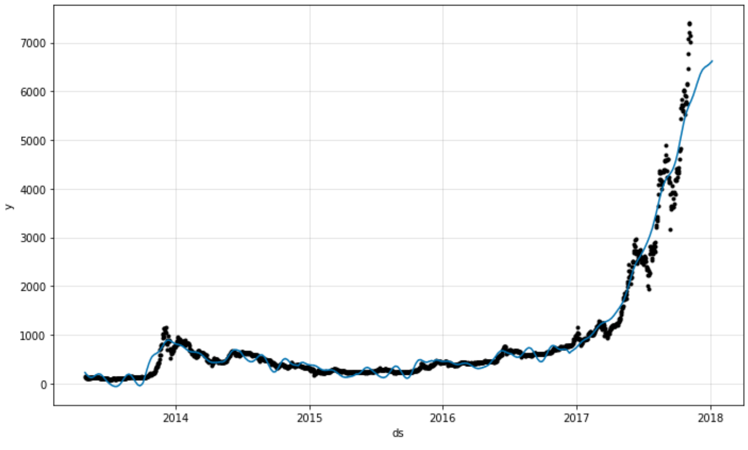


Fig 3.4 Ds(Date) Versus Y(Close Price) Graph Of A Cryptocurency Plotted Using Prophet() Package From Year 2014 To 2018

Implementation of cryptocurrency chatbot

Steps in implementing the cryptocurrency chatbot:

- Log in to IBM Watson Assistant account
- Create a workspace in the Watson Assistant
- Describe the workspace
- Create intents and entities related to the chatbot
- Create the dialog flow and train the bot according to the predicted user inputs
- Build a client application
- Getting the service information by knowing the credentials of the workspace.
- Communicating with the Watson Assitant service and processing user inputs to detect intents and entities by maintaining state.
- Implementing cryptocurrency app actions.

3.3 MODULES

3.3.1 Data Preprocessing

The dataset considered is pre-processed into a more user understandable format. The dataset consists of many unknown values. These were processed and removed. Other unwanted necessary data were also omitted. Hence program running time is reduced due to decreased dataset size. [5] Tasks in data preprocessing are:

- Data cleaning: It involves filling missing values, smooth noisy data and resolve inconsistencies. In this project, the data that having unknown values were removed. This reduced the dataset size and running time.
- Data integration: using multiple databases or files.

- Data reduction: reducing the volume but producing the same or similar analytical results. In this project, the redundant values were removed the dataset

3.3.2 Feature Selection

Feature selection refers to the process of extracting useful information or features from existing data. This process is for the selection of meaningful inputs that are used to find the appropriate output.

3.3.3 Analysing the datasets of various cryptocurrencies3.3.3.1. Closing price of various cryptocurrencies:

It denotes the value for which the crypotcurrency prices are closed at the given day where the prices are denoted in USD and date is denoted in the dd-mm-yy format.

3.3.3.2 CandleStick Charts for various cryptocurrencies:

CandleStick Charts are technical tool that pack data for multiple time frames into single price bars and very famous in financial world.Proper color coding adds depth to this colorful technical tool, which dates back to 18th century Japanese rice.

3.3.4 Correlation Map for various cryptocurrency:

3.3.4.1 Spearman Correlation Map:

Spearman's correlation assesses monotonic relationships (whether linear or not). If there are no repeated data values, a perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other.

3.3.4.2 Pearson Correlation Map:

The most familiar measure of dependence between two quantities is the Pearson product-moment correlation coefficient, or "Pearson's correlation coefficient", commonly called simply "the correlation coefficient". It is obtained by dividing the covariance of the two variables by the product of their standard deviations.

3.3.4.3 Kendall Correlation Map:

In statistics, the Kendall rank correlation coefficient**,** is a statistic used to measure the ordinal association between two measured quantities.

3.3.5 Time Series Models

3.3.5.1 Time Series Forecasting using fbprophet():

Prophet is a procedure for forecasting time series data. It is based on an additive model where non-linear trends are fit with yearly and weekly seasonality, plus holidays. It works best with daily periodicity data with at least one year of historical data. Prophet is robust to missing data, shifts in the trend, and large outliers.

the Core Data Science team at Facebook recently published a new method called Prophet, which enables data analysts and developers alike to perform forecasting at scale in Python3.

Steps in the fbprophet() model development

- Selecting ds and y: The prophet model requires only two columns in its dataframe. ds: Datetime column and y:The column whose future values needs to be predicted. Thus creating a new dataframe for forecasting values.

- Call the Prophet() function

- Fitting the model: model = Prophet()

- Making future dataframe: future = model.make_future_dataframe(periods = 60)

- Predicting future values by calling the predict() function: model.predict(future)

3.3.5.2 Time Series Forecasting using ARIMA model

ARIMA models are a popular and flexible class of forecasting model that utilize historical information to make predictions. This type of model is a basic forecasting technique that can be used as a foundation for more complex models.

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average.

Steps in the ARIMA model development:

- Plot the time series graph of the dataset to analyse the stationarity of the dataset.

- Plot ACF and PACF graphs to identify the correlation between the current values and the past values.

- Identify the p and q value from ACF and PACF plots respectively.

- Call the ARIMA() function and pass the required p,q,d values: ARIMA(df_bitcoin, order=(5,1,0))
- Fit the model: model.fit(disp=0)
- Call the summary() function for obtaining the ARIMA model results
- Divide the dataset into test and training sets: train, test = X[0:size], X[size:len(X)]
- Calculate the forecast error and accuracy error

3.3.6 Create the forecasted dataframe.

Create the future dataframes of all the forecasted datasets of the cryptocurrencies that consist of the future values.

Steps in creating a future dataframe:

- Create an object of prophet function
- Fit the dataframe as m.fit(df);
- Specify the time period of the future dataframe
- future = m.make_future_dataframe(periods=120)
- Pass the dataframe to the predict() function
- forecast = m.predict(future)
- Convert the dataframe to .csv file.
- forecast.to_csv('iota_forecast.csv'

3.3.6.1 Plot the graph of the future closing price of various cryptocurrencies.

Plot the graphs for the future closing price for cryptocurrencies using matplotlib to know about their future price fluctuations in order to make better investment.

3.3.7 Create the interactive cryptocurrency chatbot and building a client application

Using the results of the forecasted dataframe of fifteen different cryptocurrency prices, a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

Steps in implementing the cryptocurrency chatbot:

- Log in to IBM Watson Assistant account
- Create a workspace in the Watson Assistant
- Describe the workspace
- Create intents and entities related to the chatbot
- Create the dialog flow and train the bot according to the predicted user inputs
- Build a client application
- Getting the service information by knowing the credentials of the workspace.
- Communicating with the Watson Assitant service and processing user inputs to detect intents and entities by maintaining state.

3.4 DATABASE DESIGN

3.4.1 Entity Relationship Diagram

In entity relationship diagram, there are four entities namely data analyst, programmer, prediction system, investors who performs their respective work flows as mentioned above the arrows directed towards their respective entities.

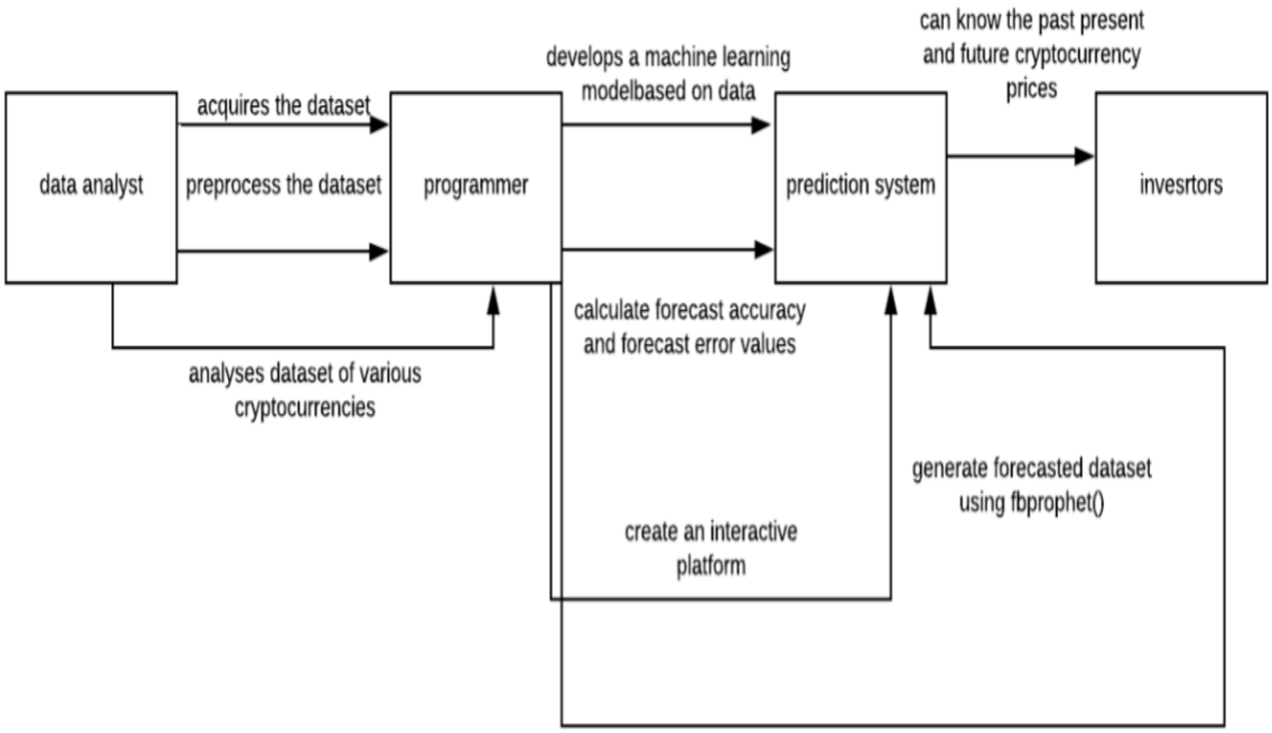


Fig 3.5 Entity Relationship Diagram For Predictive Analysis Of Cryptocurrency Prices

3.4.2 Tables Or Entities

Table 3.1 BitcoinCash dataset from 23-jul to 7-nov-17

Date	Open	High	Low	Close
4-Aug-13	0.005874	0.005927	0.005874	0.005882
5-Aug-13	0.005875	0.00598	0.005613	0.005613
6-Aug-13	0.005637	0.005661	0.004629	0.00468
7-Aug-13	0.004669	0.004682	0.004333	0.004417
8-Aug-13	0.004397	0.004424	0.004175	0.004254
9-Aug-13	0.004257	0.004367	0.004253	0.004291
10-Aug-13	0.004291	0.004366	0.004279	0.004314
11-Aug-13	0.004314	0.004421	0.0043	0.004415
12-Aug-13	0.004414	0.00452	0.004359	0.004449
13-Aug-13	0.004463	0.004463	0.004171	0.004245
14-Aug-13	0.004267	0.004267	0.003785	0.003788
15-Aug-13	0.003788	0.003804	0.003067	0.003092
16-Aug-13	0.003082	0.003162	0.002875	0.003134
17-Aug-13	0.003138	0.005684	0.003138	0.005671
18-Aug-13	0.005687	0.006263	0.005585	0.005617
19-Aug-13	0.005587	0.005839	0.005223	0.00526
20-Aug-13	0.005376	0.006158	0.005309	0.006098
21-Aug-13	0.006098	0.006349	0.00595	0.006131
22-Aug-13	0.006131	0.006138	0.005143	0.005143
23-Aug-13	0.005179	0.005232	0.004919	0.005139
24-Aug-13	0.005139	0.005603	0.005126	0.005546
25-Aug-13	0.005526	0.006148	0.005519	0.006104

Table 3.2 Bitcoin dataset from 28-apr-13 to 7-nov-17

Date	Open	High	Low	Close
12-Aug-16	0.013826	0.013897	0.009226	0.011175
13-Aug-16	0.011171	0.016318	0.008483	0.014405
14-Aug-16	0.014405	0.033588	0.013431	0.02396
15-Aug-16	0.024023	0.027224	0.019404	0.024511
16-Aug-16	0.023893	0.031045	0.022763	0.028699
17-Aug-16	0.02831	0.030283	0.024678	0.027411
18-Aug-16	0.02736	0.027938	0.02085	0.02399
19-Aug-16	0.02468	0.024977	0.019231	0.020717
20-Aug-16	0.020733	0.022651	0.019495	0.020357
21-Aug-16	0.02054	0.021267	0.019723	0.020555
22-Aug-16	0.020654	0.023463	0.020379	0.021984
23-Aug-16	0.021985	0.023075	0.019797	0.022228
24-Aug-16	0.022228	0.022234	0.020301	0.02095
25-Aug-16	0.02095	0.021053	0.016875	0.018488
26-Aug-16	0.018488	0.02432	0.018065	0.021864
27-Aug-16	0.022749	0.027015	0.021072	0.023835
28-Aug-16	0.02338	0.025276	0.020146	0.021103
29-Aug-16	0.021109	0.023414	0.020138	0.021121
30-Aug-16	0.021179	0.021537	0.020227	0.020241
31-Aug-16	0.020245	0.022638	0.020113	0.021287
1-Sep-16	0.021289	0.022687	0.020624	0.02209
2-Sep-16	0.022095	0.0248	0.021984	0.023228

Table 3.3 Dash dataset from 14-feb to 7-nov-17

Date	Open	High	Low	Close
23-Jun-17	35.05	48.58	17.67	45.63
24-Jun-17	44.48	116.4	42.04	101.83
25-Jun-17	103.21	168.49	72.49	86.55
26-Jun-17	88.17	98.78	49.7	59.9
27-Jun-17	59.82	62.31	36.29	53.05
28-Jun-17	51.92	58.38	46.21	53.07
29-Jun-17	53.26	53.26	38.82	42.89
30-Jun-17	42.26	43.96	29.13	34.2
1-Jul-17	34.19	68.93	33.29	45.26
2-Jul-17	43.12	54.01	39.95	42.87
3-Jul-17	43.17	46.92	41.41	41.69
4-Jul-17	41.6	43.46	37.38	39.42
5-Jul-17	39.39	41.59	37.64	39.03
6-Jul-17	38.6	39.21	32.2	35.79
7-Jul-17	36.08	36.88	25.44	29.39
8-Jul-17	28.99	30.77	23.66	27.51
9-Jul-17	27.54	30.14	26.83	28.29
10-Jul-17	28.65	31.54	20.99	24.04
11-Jul-17	23.53	26.49	18.33	20.73
12-Jul-17	20.32	32.9	18.75	30.71
13-Jul-17	30.83	31.24	25.64	26.82
14-Jul-17	26.8	27.67	21.18	23.24

Table 3.4 Ethereum dataset from 24-jul-16 to 7-nov-17

Date	Open	High	Low	Close
23-Jun-17	35.05	48.58	17.67	45.63
24-Jun-17	44.48	116.4	42.04	101.83
25-Jun-17	103.21	168.49	72.49	86.55
26-Jun-17	88.17	98.78	49.7	59.9
27-Jun-17	59.82	62.31	36.29	53.05
28-Jun-17	51.92	58.38	46.21	53.07
29-Jun-17	53.26	53.26	38.82	42.89
30-Jun-17	42.26	43.96	29.13	34.2
1-Jul-17	34.19	68.93	33.29	45.26
2-Jul-17	43.12	54.01	39.95	42.87
3-Jul-17	43.17	46.92	41.41	41.69
4-Jul-17	41.6	43.46	37.38	39.42
5-Jul-17	39.39	41.59	37.64	39.03
6-Jul-17	38.6	39.21	32.2	35.79
7-Jul-17	36.08	36.88	25.44	29.39
8-Jul-17	28.99	30.77	23.66	27.51
9-Jul-17	27.54	30.14	26.83	28.29
10-Jul-17	28.65	31.54	20.99	24.04
11-Jul-17	23.53	26.49	18.33	20.73
12-Jul-17	20.32	32.9	18.75	30.71
13-Jul-17	30.83	31.24	25.64	26.82
14-Jul-17	26.8	27.67	21.18	23.24

Table 3.5 Ethereum dataset from 7-aug-15 to 7-nov -17

Date	Open	High	Low	Close
9-Sep-16	0.181483	0.558951	0.181357	0.558478
10-Sep-16	0.558536	0.559143	0.37096	0.391001
11-Sep-16	0.390948	0.398459	0.37279	0.37615
12-Sep-16	0.376312	0.376671	0.360443	0.374598
13-Sep-16	0.374469	0.375092	0.301766	0.309509
14-Sep-16	0.309516	0.337779	0.306223	0.325127
15-Sep-16	0.325188	0.365124	0.325024	0.349832
16-Sep-16	0.349726	0.389464	0.348948	0.388858
17-Sep-16	0.388924	0.427314	0.388896	0.426568
18-Sep-16	0.426591	0.427137	0.384916	0.384916
19-Sep-16	0.384877	0.384881	0.331207	0.331296
20-Sep-16	0.331291	0.336507	0.309502	0.312621
21-Sep-16	0.312539	0.318675	0.303564	0.303794
22-Sep-16	0.303806	0.308768	0.252592	0.252723
23-Sep-16	0.252745	0.277371	0.241462	0.243237
24-Sep-16	0.243321	0.25485	0.213843	0.213843
25-Sep-16	0.213843	0.224547	0.192797	0.198994
26-Sep-16	0.199118	0.224506	0.193879	0.202491
27-Sep-16	0.202473	0.202473	0.179976	0.197913
28-Sep-16	0.197915	0.21275	0.184495	0.194798
29-Sep-16	0.194845	0.208155	0.175516	0.191738
30-Sep-16	0.191685	0.194601	0.180176	0.184939

Table 3.6 Iota dataset from 13-jul-17 to 7-nov-17

Date	Open	High	Low	Close
1-Apr-15	0.0004	0.000458	0.00017	0.000242
2-Apr-15	0.000242	0.000323	0.000227	0.000314
3-Apr-15	0.000309	0.00033	0.000291	0.00031
4-Apr-15	0.00031	0.000318	0.000251	0.000277
5-Apr-15	0.000272	0.000283	0.000218	0.000232
6-Apr-15	0.000232	0.000299	0.000229	0.000289
7-Apr-15	0.000288	0.000288	0.000234	0.000241
8-Apr-15	0.000245	0.000259	0.000223	0.000228
9-Apr-15	0.000228	0.000239	0.000213	0.000224
10-Apr-15	0.000224	0.000227	0.000167	0.000175
11-Apr-15	0.000175	0.000192	0.000143	0.000163
12-Apr-15	0.000163	0.000173	0.000144	0.000146
13-Apr-15	0.000146	0.000155	0.000123	0.000139
14-Apr-15	0.000139	0.000178	0.000137	0.000153
15-Apr-15	0.000156	0.000172	0.000146	0.000155
16-Apr-15	0.000155	0.000174	0.000155	0.000163
17-Apr-15	0.000168	0.000172	0.00016	0.000167
18-Apr-15	0.000167	0.00017	0.000141	0.000141
19-Apr-15	0.000141	0.00016	0.000137	0.000147
20-Apr-15	0.000147	0.00015	0.000128	0.000133
21-Apr-15	0.000133	0.000148	0.000129	0.000137
22-Apr-15	0.000137	0.000149	0.000129	0.000138

Table 3.7 Litecoin dataset from 28-apr-13 to 7-nov-17 Table 3.8 Monero dataset from 21-may-14 to 7-nov

Date	Open	High	Low	Close
21-May-14	2.47	2.65	1.23	1.6
22-May-14	1.59	2.19	1.36	2.1
23-May-14	2.05	3.43	2.05	2.96
24-May-14	2.92	4.01	2.62	3.7
25-May-14	4.04	4.04	2.8	3.14
26-May-14	3.22	3.76	2.31	3.02
27-May-14	3.03	3.03	2.07	2.21
28-May-14	2.24	2.85	1.91	2.27
29-May-14	2.21	2.49	1.68	1.68
30-May-14	1.68	1.87	1.08	1.57
31-May-14	1.58	2.28	1.41	1.94
1-Jun-14	1.94	2.57	1.54	1.74
2-Jun-14	1.76	1.94	1.6	1.73
3-Jun-14	1.73	2.1	1.65	1.93
4-Jun-14	1.93	1.99	1.71	1.81
6-Jun-14	1.46	1.51	1.17	1.24
7-Jun-14	1.24	1.69	1.21	1.36
8-Jun-14	1.36	1.44	1.24	1.34
9-Jun-14	1.33	1.64	1.27	1.52
10-Jun-14	1.48	2.39	1.31	1.73
11-Jun-14	1.72	2.17	1.53	1.62
12-Jun-14	1.62	2.17	1.53	1.74

Date	Open	High	Low	Close
28-Apr-13	4.3	4.4	4.18	4.35
29-Apr-13	4.37	4.57	4.23	4.38
30-Apr-13	4.4	4.57	4.17	4.3
1-May-13	4.29	4.36	3.52	3.8
2-May-13	3.78	4.04	3.01	3.37
3-May-13	3.39	3.45	2.4	3.04
4-May-13	3.03	3.64	2.9	3.48
5-May-13	3.49	3.69	3.35	3.59
6-May-13	3.59	3.78	3.12	3.37
7-May-13	3.37	3.41	2.94	3.33
8-May-13	3.28	3.49	3.28	3.41
9-May-13	3.4	3.44	3.29	3.42
10-May-13	3.41	3.63	3.37	3.44
11-May-13	3.46	3.48	3.3	3.35
12-May-13	3.36	3.42	3.22	3.27
13-May-13	3.26	3.37	3.2	3.28
14-May-13	3.28	3.33	2.78	2.82
15-May-13	2.82	3.04	2.64	2.94
16-May-13	2.93	2.97	2.69	2.92
17-May-13	2.89	3.34	2.87	3.19
18-May-13	3.2	3.36	3.13	3.29
19-May-13	3.3	3.41	3.19	3.3

Table 3.9 Nem dataset from 1-apr-15 to 7-nov-17

Date	Open	High	Low	Close
13-Jun-17	0.638503	0.652862	0.53391	0.590255
14-Jun-17	0.592347	0.606196	0.495745	0.528916
15-Jun-17	0.528284	0.543165	0.300365	0.363661
16-Jun-17	0.353285	0.448249	0.309852	0.410757
17-Jun-17	0.426762	0.444205	0.414139	0.419906
18-Jun-17	0.420597	0.426069	0.39379	0.405862
19-Jun-17	0.405456	0.42099	0.388231	0.412183
20-Jun-17	0.414299	0.422032	0.398649	0.418494
21-Jun-17	0.419439	0.43734	0.405037	0.413547
22-Jun-17	0.413371	0.427283	0.407261	0.417742
23-Jun-17	0.417769	0.48091	0.408405	0.48091
24-Jun-17	0.482769	0.635612	0.482769	0.51291
25-Jun-17	0.514185	0.525564	0.454951	0.470941
26-Jun-17	0.470192	0.473662	0.351156	0.402438
27-Jun-17	0.411989	0.420035	0.353549	0.392269
28-Jun-17	0.399874	0.399874	0.357338	0.379136
29-Jun-17	0.375767	0.459387	0.361568	0.417798
30-Jun-17	0.417167	0.453008	0.398195	0.401591
1-Jul-17	0.402295	0.414192	0.381489	0.38963
2-Jul-17	0.39188	0.39689	0.366798	0.382249
3-Jul-17	0.379055	0.384955	0.356024	0.376139
4-Jul-17	0.3749	0.410698	0.373865	0.39639

Table 3.10 Neo dataset from 9-sep-16 to 7-nov-17

Date	Open	High	Low	Close
7-Aug-15	2.83	3.54	2.52	2.77
8-Aug-15	2.79	2.8	0.714725	0.753325
9-Aug-15	0.706136	0.87981	0.629191	0.701897
10-Aug-15	0.713989	0.729854	0.636546	0.708448
11-Aug-15	0.708087	1.13	0.663235	1.07
12-Aug-15	1.06	1.29	0.883608	1.22
13-Aug-15	1.22	1.97	1.17	1.83
14-Aug-15	1.81	2.26	1.75	1.83
15-Aug-15	1.8	1.88	1.57	1.69
16-Aug-15	1.68	1.7	1.09	1.57
17-Aug-15	1.58	1.58	1.19	1.2
18-Aug-15	1.22	1.33	1.09	1.09
19-Aug-15	1.17	1.32	1.17	1.26
20-Aug-15	1.25	1.53	1.25	1.46
21-Aug-15	1.48	1.56	1.35	1.4
22-Aug-15	1.4	1.48	1.35	1.38
23-Aug-15	1.38	1.41	1.3	1.35
24-Aug-15	1.35	1.36	1.23	1.23
25-Aug-15	1.23	1.24	1.13	1.14
26-Aug-15	1.13	1.2	1.06	1.16
27-Aug-15	1.17	1.19	1.14	1.15
28-Aug-15	1.15	1.21	1.12	1.19

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULE IMPLEMENTATION

4.1.1 Data Preprocessing

- Data cleaning: In this project, which consist of 18 datasets, required filling of the missing values and resolving inconsistencies. Example: The dataset of Bitcoin_cash_price consists of two columns i.e. MarketCap and Volume, whose values are unknown .Therefore this reduced the dataset size and running time.
- Data integration: Cryptocurrency price analysis consist of 18 datasets of different cryptocurrencies, Which has been integrated in one unit.
- Data reduction: Some of the values of the cryptocurrency dataset is not required ,example: in the ethereum data set ,features like eth_gasprice, eth_gasused is not needed during the model development. Therefore it reduces the volume but produces the same or similar analytical results.

4.1.2 Feature selection

In this project some attributes were selected and some were discarded based on the features that are useful for analysis and model development.

The features selected are as follows:

- Date : date of observation
- Open : Opening price on the given day
- High : Highest price on the given day
- Low : Lowest price on the given day
- Close : Closing price on the given day

4.1.3. Analysing the datasets of various cryptocurrencies

4.1.3.1Closing price of various cryptocurrencies:

Fig 4.1 Stratis Closing Price from 2016-17

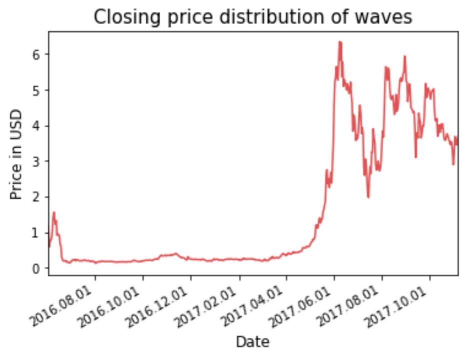


Fig 4.2 Waves Closing Price from 2016-17

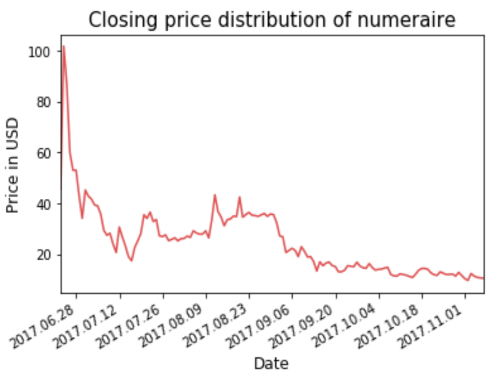


Fig 4.3 Numeraire Closing Price From 2016-17

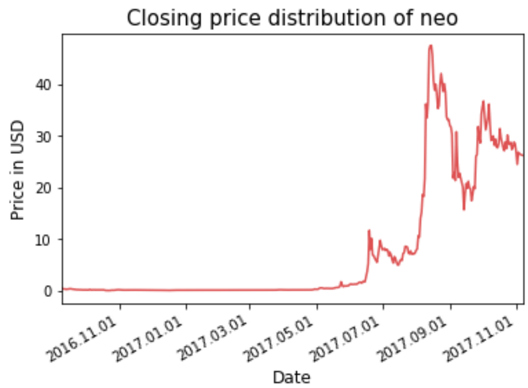


Fig 4.4 Neo Closing Price from 2016-17

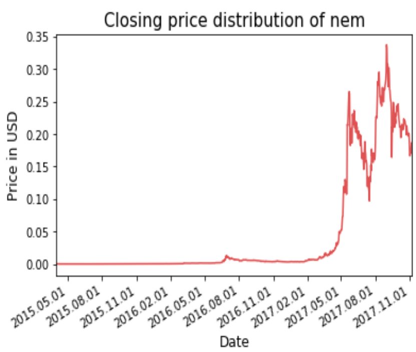


Fig 4.5 Nem Closing Price from 2015-17

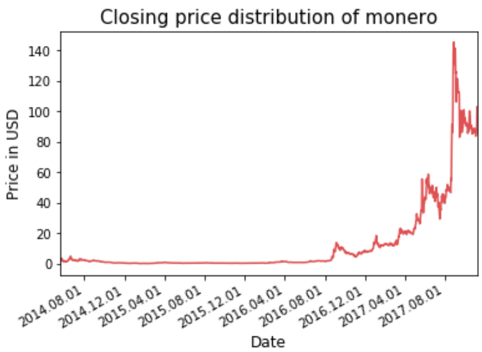


Fig 4.6 Monero Closing Price from 2014-17

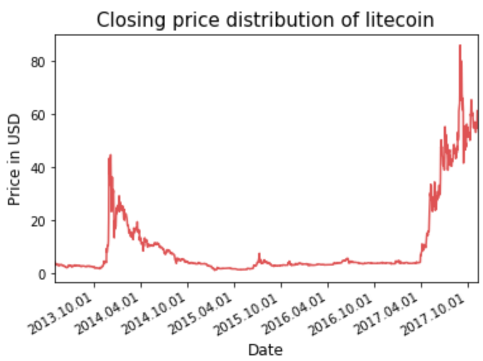


Fig 4.7 Litecoin Closing Price from 2013-17

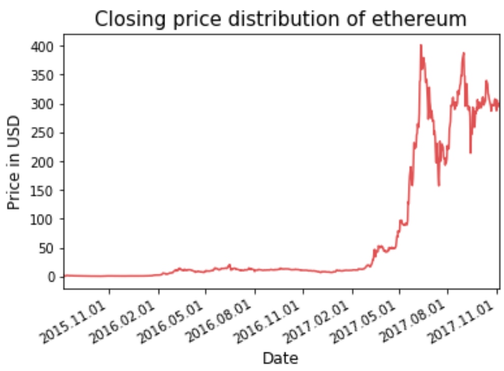


Fig 4.8 Ethereum Closing Price from 2015-17

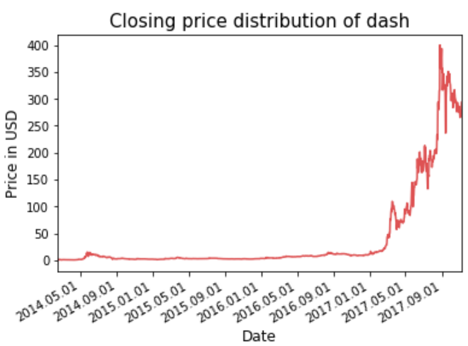


Fig 4.9 Dash Closing Price from 2014-17

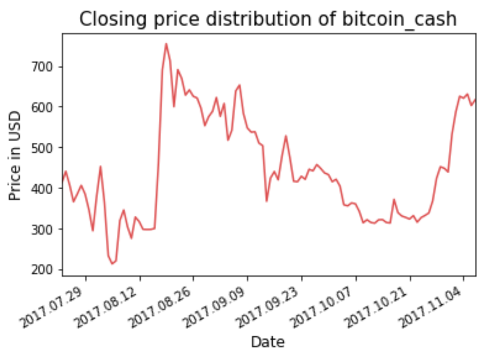


Fig 4.10 Bitcoin_cash Closing Price from 2016-17

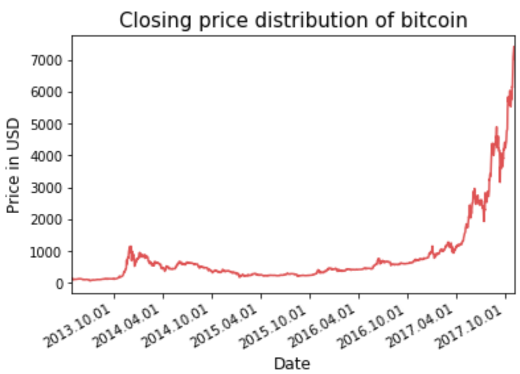


Fig 4.11 Bitcoin Closing Price from 2013-17

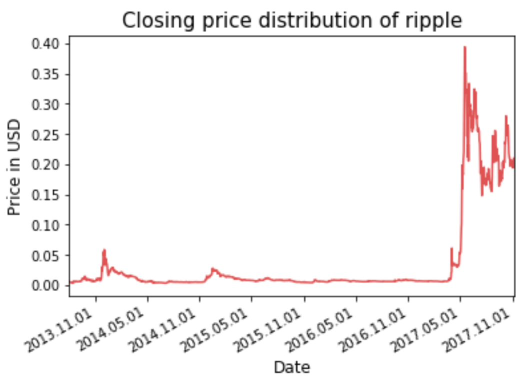


Fig 4.12 Ripple Closing Price from 2013-17

4.1.3.2 Analysing of CandleStick chart for various cryptocurrency

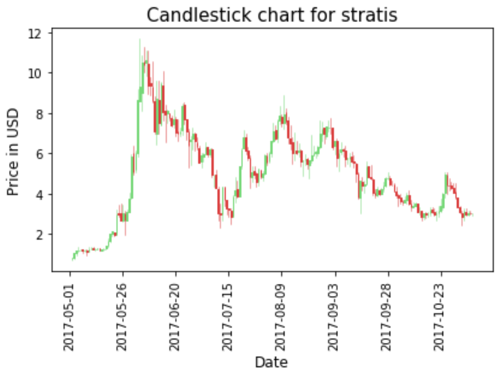


Fig 4.13 Stratis CandleStick chart

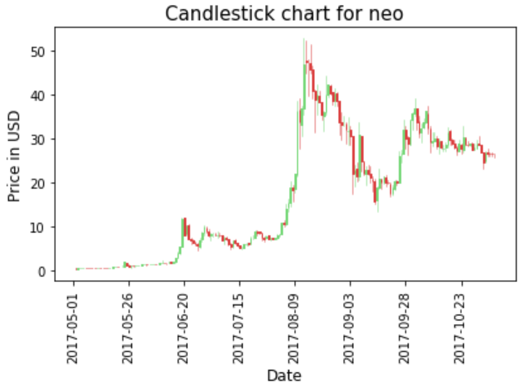


Fig 4.14 Neo CandleStick chart

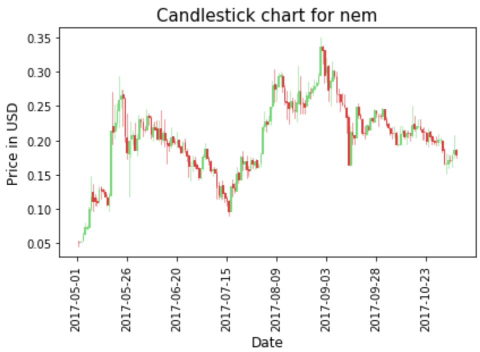


Fig 4.15 Nem CandleStick chart

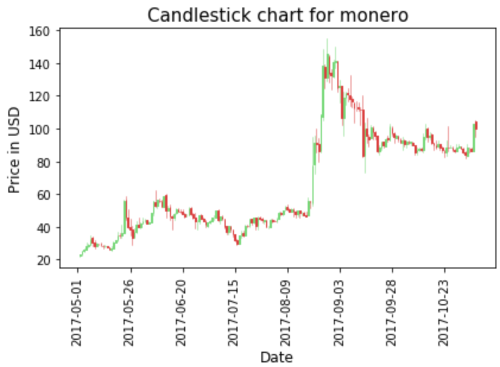


Fig 4.16 Monero CandleStick chart

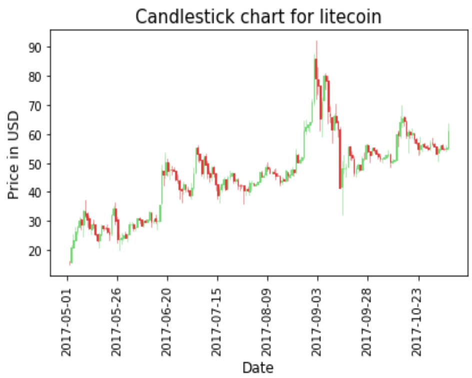


Fig 4.17 Litecoin CandleStick chart

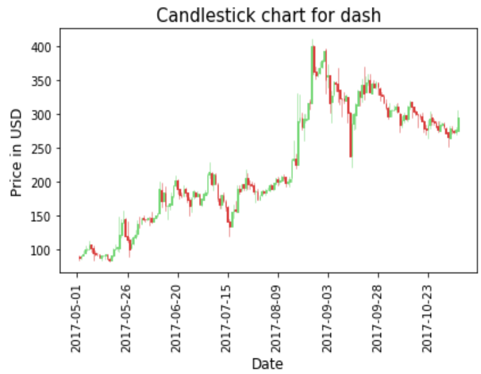


Fig 4.18 Dash CandleStick chart

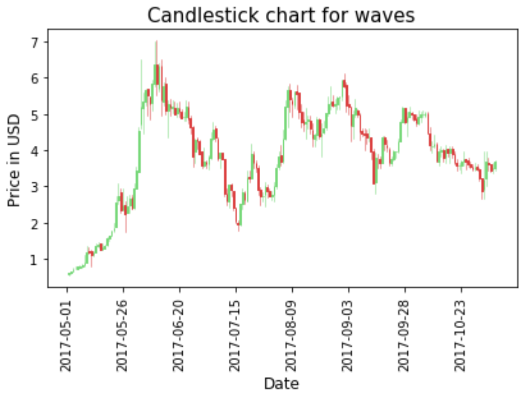


Fig 4.19 Waves CandleStick chart

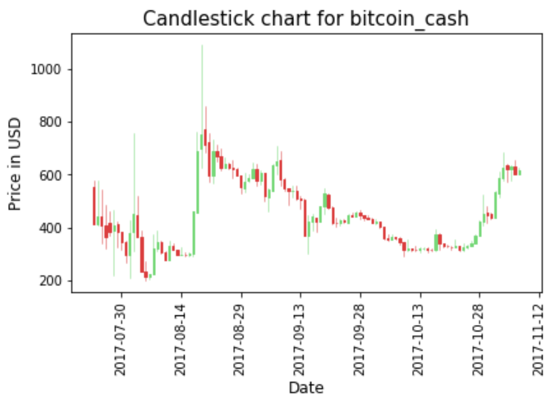


Fig 4.20 Bitcoin_cash CandleStick chart

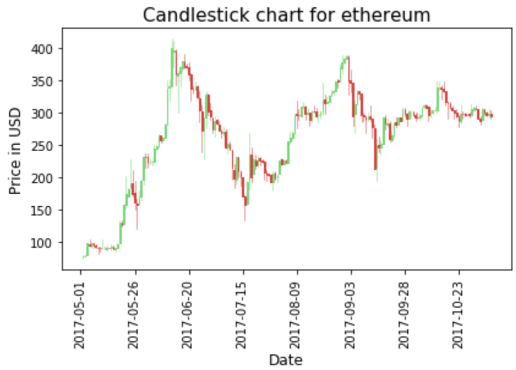


Fig 4.21 Ethereum CandleStick chart

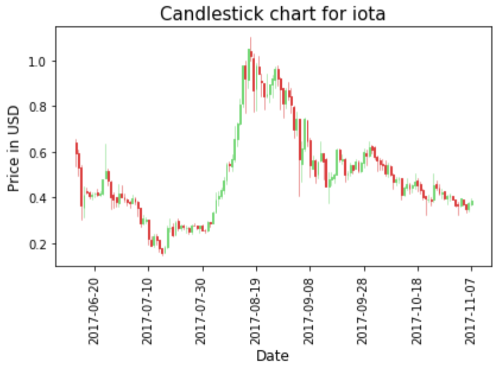


Fig 4.22 Iota CandleStick chart

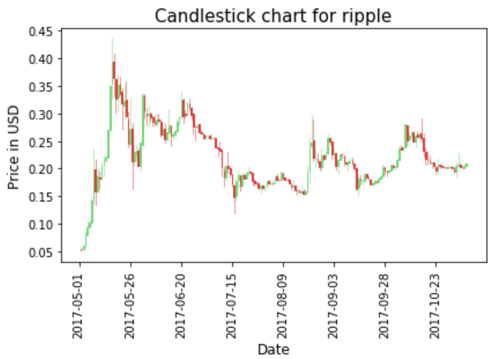


Fig 4.23 Ripple CandleStick chart

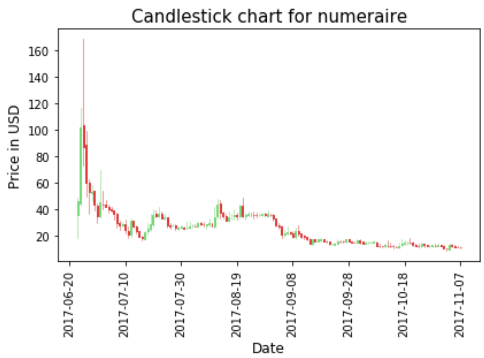


Fig 4.24 Numeraire CandleStick chart

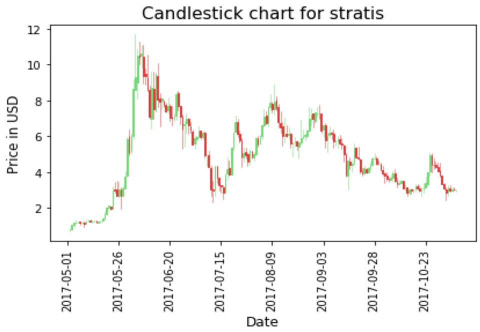


Fig 4.25 Stratis CandleStick chart

4.1.3.3 Correlation Map for various cryptocurrency

4.1.3.3.1 Spearman Correlation Map

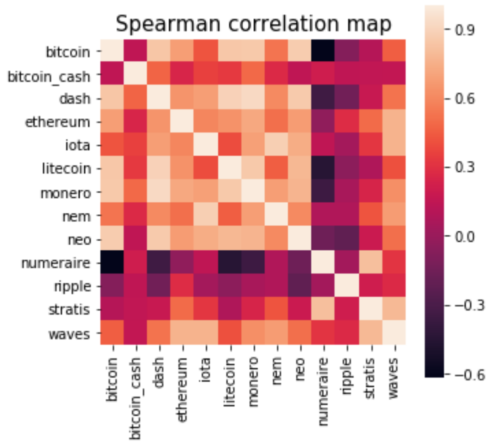


Fig 4.26 Spearman Correlation Map between different cryptocurrencies

4.1.3.3.2 Pearson Correlation Map

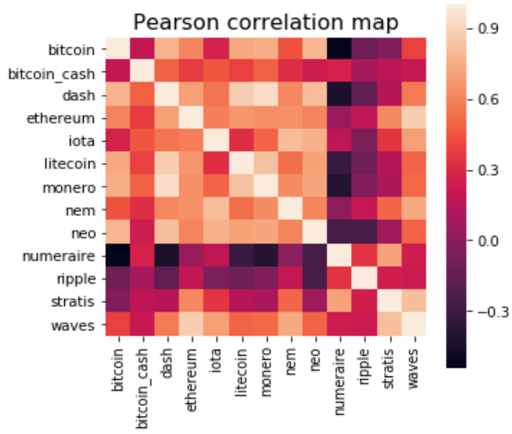


Fig 4.27 Pearson Correlation Map between different cryptocurrencies

4.1.3.3.3 Kendall Correlation Map

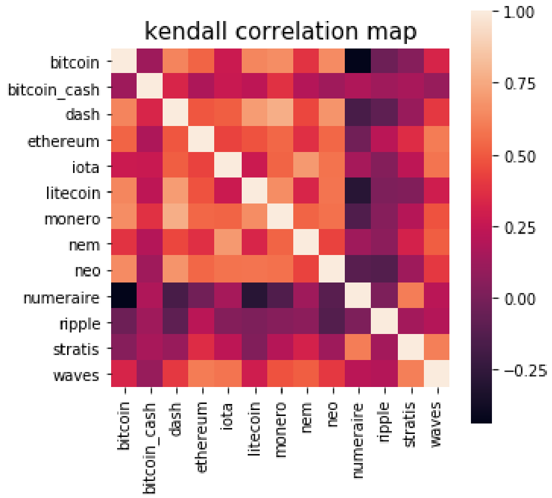


Fig 4.28 Kendall Correlation Map between different cryptocurrencies

4.1.4 Time Series Forecasting using fbprophet()

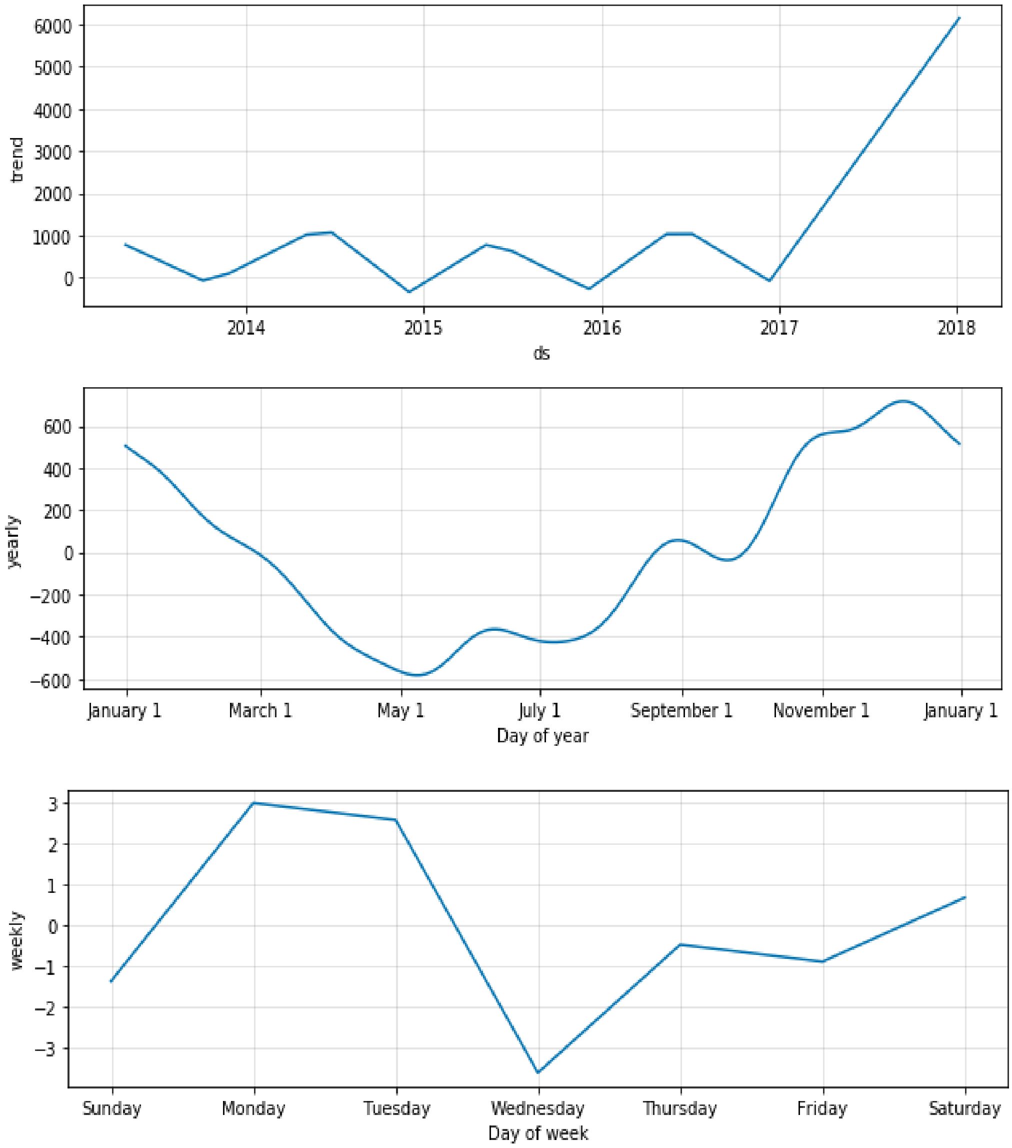
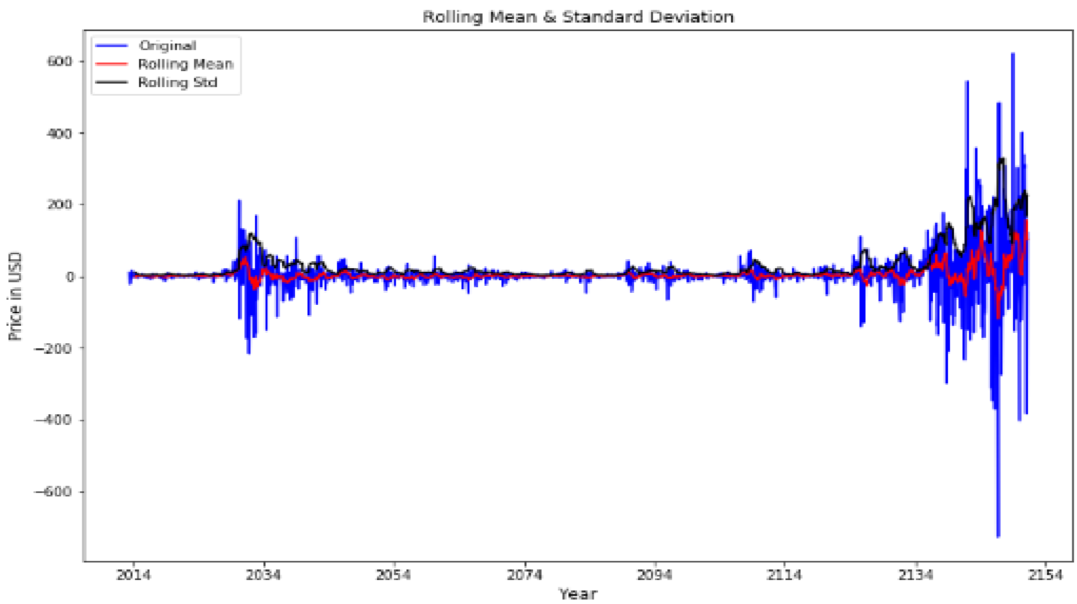


Fig 4.29 Weekly And Yearly Trends Of A Cryptocurrency from Year 2014-18

Fig 4.30 Fbprophet() Graph of cryptocurrency from year 2014-18 showing the accuracy rate

4.1.4.1 Time series Forecasting using ARIMA model




```
Results of Dickey-Fuller Test:
Test Statistic                   -4.790943
p-value                           0.000057
#Lags Used                       23.000000
Number of Observations Used    1630.000000
Critical Value (1%)              -3.434368
Critical Value (5%)              -2.863315
Critical Value (10%)             -2.567715
dtype: float64
```

Fig 4.31 Stationarity Graph Of A Cryptocurrency Where Mean, Std And Original Values Are Taken

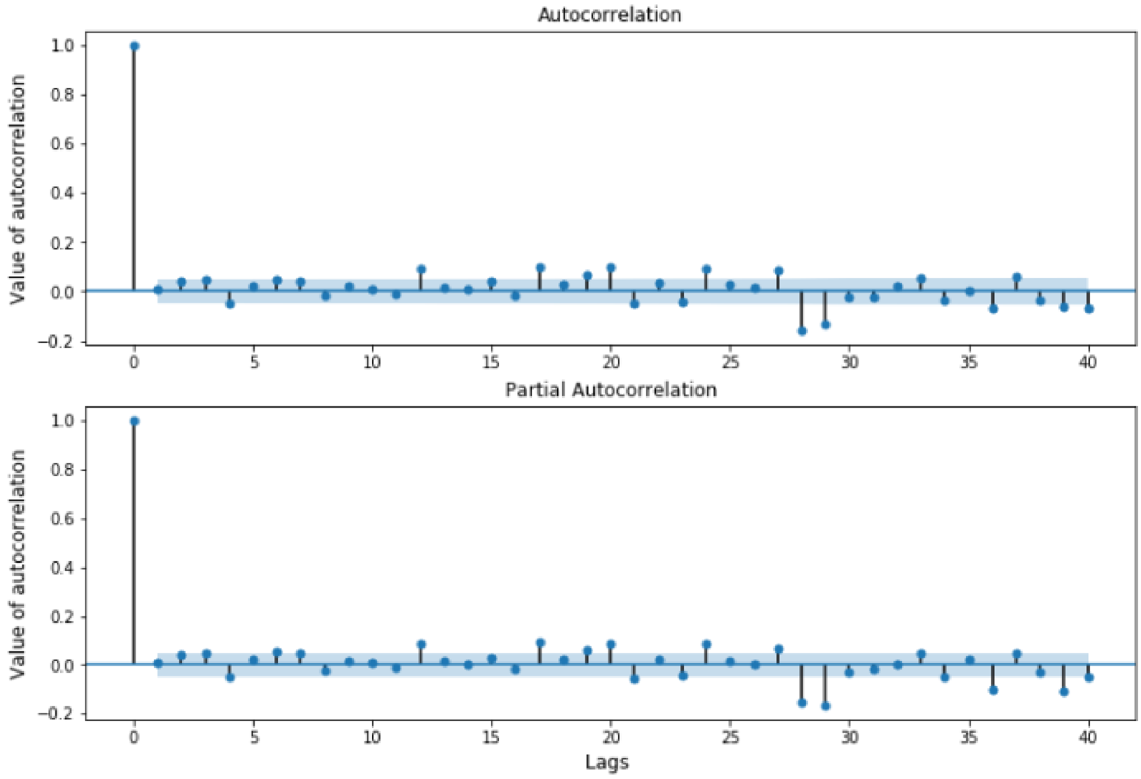


Fig 4.32 ACF And PACF Graph of a cryptocurrency to calculate the AR, I, MA values

Table 4.1 ARIMA Model Results

Model:	ARIMA(5, 1, 0)	Log Likelihood	-9154.460
Method:	css-mle	S.D. of innovati -ons	61.301
Date:	Mon, 29 Jan 2018	AIC	18322.919
Time:	16:18:32	BIC	18360.796
Sample:	11-06-2017 04-28-2013	HQIC	18336.961

Table 4.2 Dependant Factor Table Which Shows Coefficient,Std Error As The Main Factors

	coef	std err	z	P>|z|	[0.025]	[0.975]
Const	-4.2269	1.616	-2.61	0.009	-7.395	-1.059
ar.L1.D.Close	0.0086	0.025	0.35	0.726	-0.040	0.057
ar.L2.D.Close	0.0457	0.025	1.83	0.066	-0.003	0.095
ar.L3.D.Close	0.0469	0.025	1.88	0.060	-0.002	0.096
ar.L4.D.Close	-0.0539	0.025	-2.16	0.031	-0.103	-0.005
ar.L5.D.Close	0.0202	0.025	0.80	0.419	-0.029	0.069

Table 4.3 ARIMA Model Roots Table Which Shows Real,Imaginery,Modulus And Frequency

	Real	Imaginary	Modulus	Frequency
AR.1	-1.319	-1.1946j	1.7797	-0.3829
AR.2	-1.319	+1.1946j	1.7797	0.3829
AR.3	2.552	-0.0000j	2.5523	-0.0000
AR.4	1.376	-2.0559j	2.4743	-0.1561
AR.5	1.376	+2.0559j	2.4743	0.1561

4.1.5 Create the forecasted dataframe.

- forecast.to_csv('waves_forecast.csv')
- forecast.to_csv('stratis_forecast.csv')
- forecast.to_csv('ripple_forecast.csv')
- forecast.to_csv('numeraire_forecast.csv')
- forecast.to_csv('neo_forecast.csv')
- forecast.to_csv('nem_forecast.csv')
- forecast.to_csv('monero_forecast.csv')
- forecast.to_csv('litecoin_forecast.csv')
- forecast.to_csv('ethereum_forecast.csv')
- forecast.to_csv('dash_forecast.csv')
- forecast.to_csv('omisego_forecast.csv')
- forecast.to_csv('iota_forecast.csv')
- forecast.to_csv('ethclassic_forecast.csv')
- forecast.to_csv('bitcoin_cash_forecast.csv')
- forecast.to_csv('bitcoin_forecast.csv

4.1.5.1 Plot the graph of the future closing price of various cryptocurrencies.

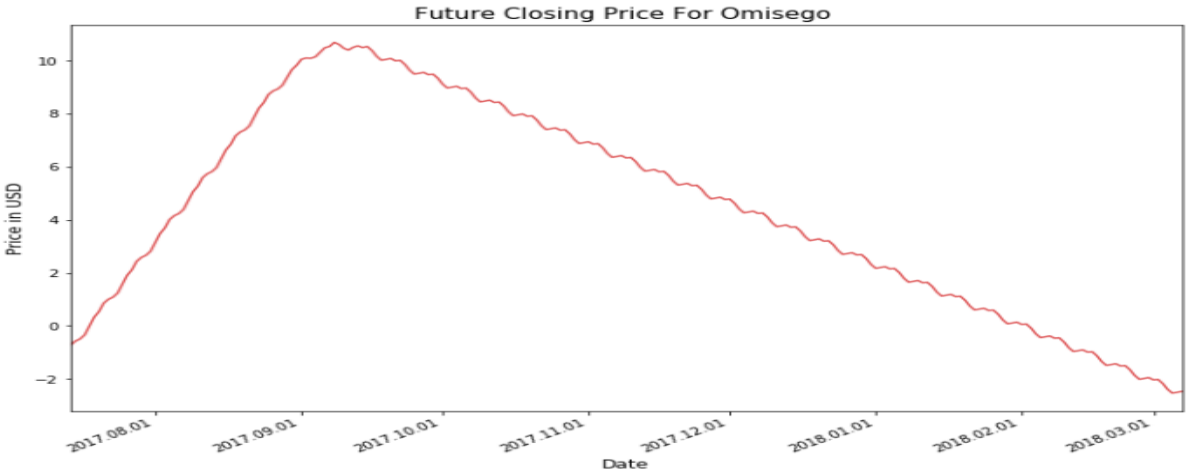


Fig4.33FutureClosingPriceForOmisegofrom2017-18

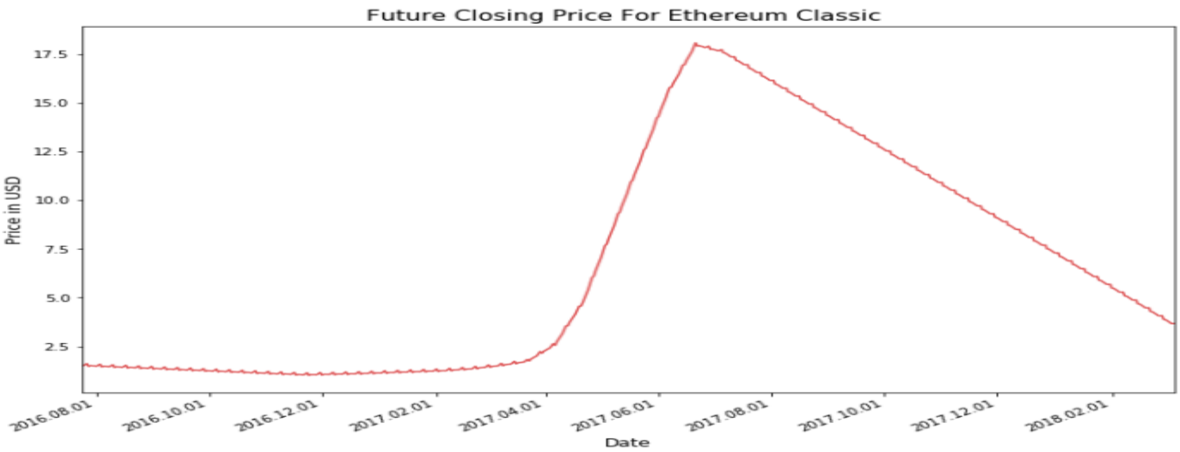


Fig4.34 Future Closing Price For Ethereum Classic From 2016-18

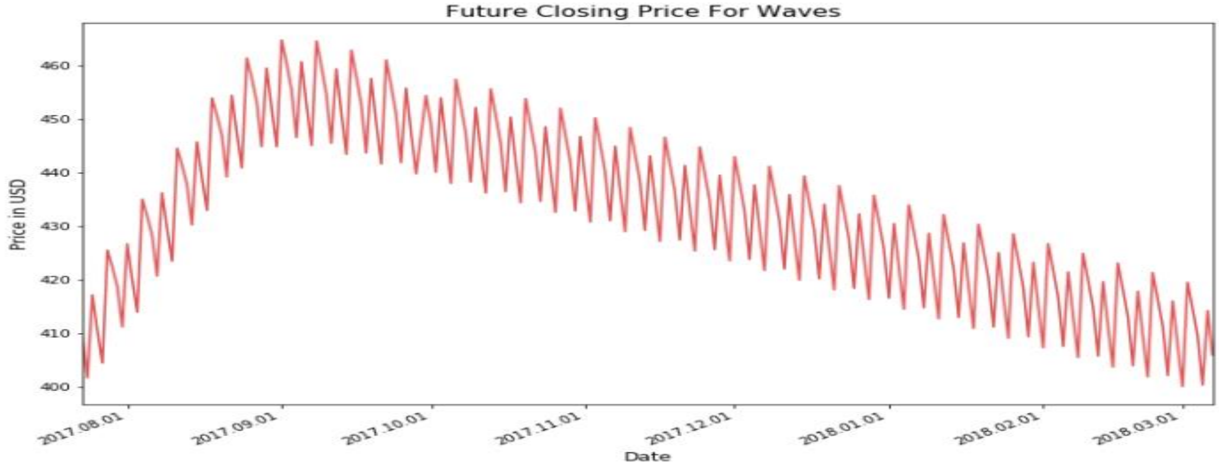


Fig4.35 Future Closing Price For Waves from 2017-18

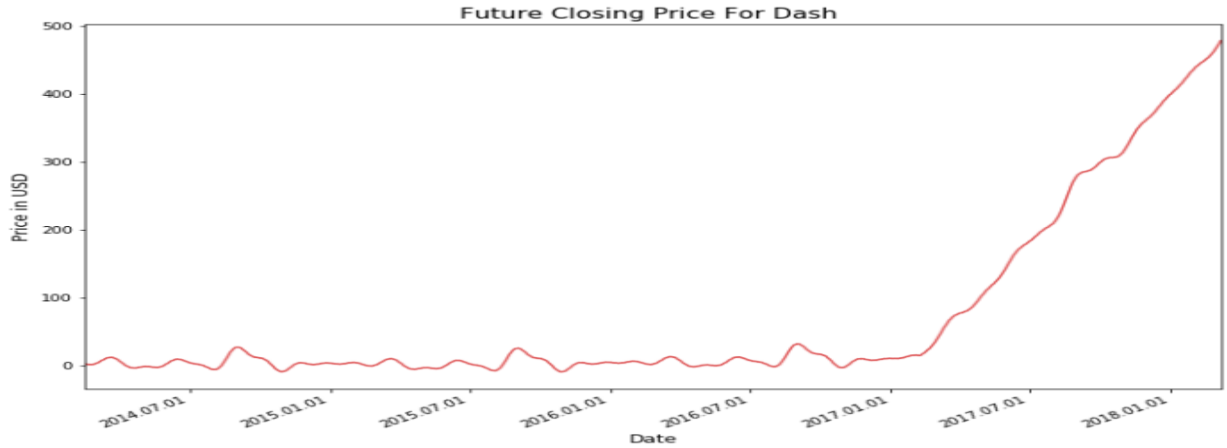


Fig4.36 Future Closing Price For Dash from 2014-18

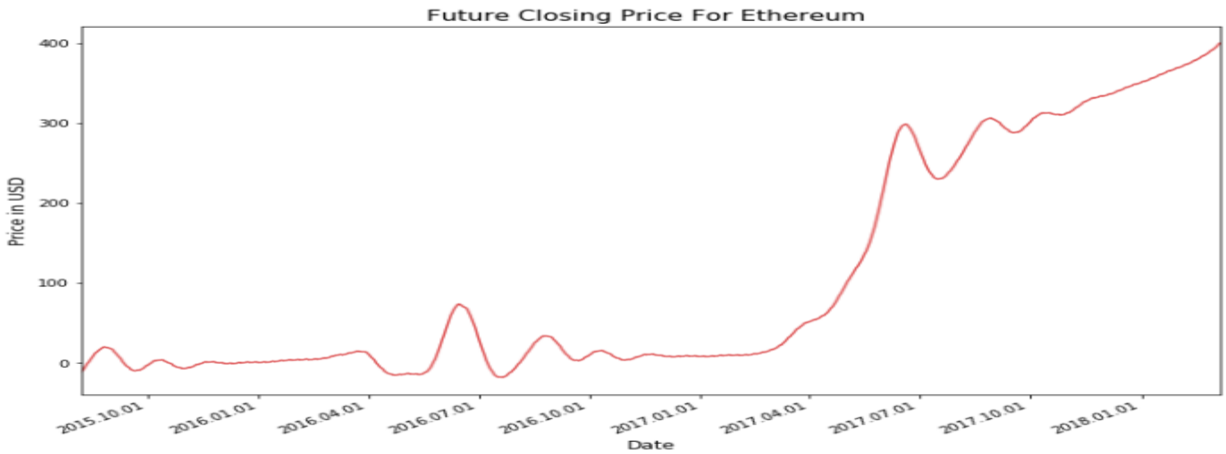


Fig4.37 Future Closing Price For Ethereum from 2015-2018

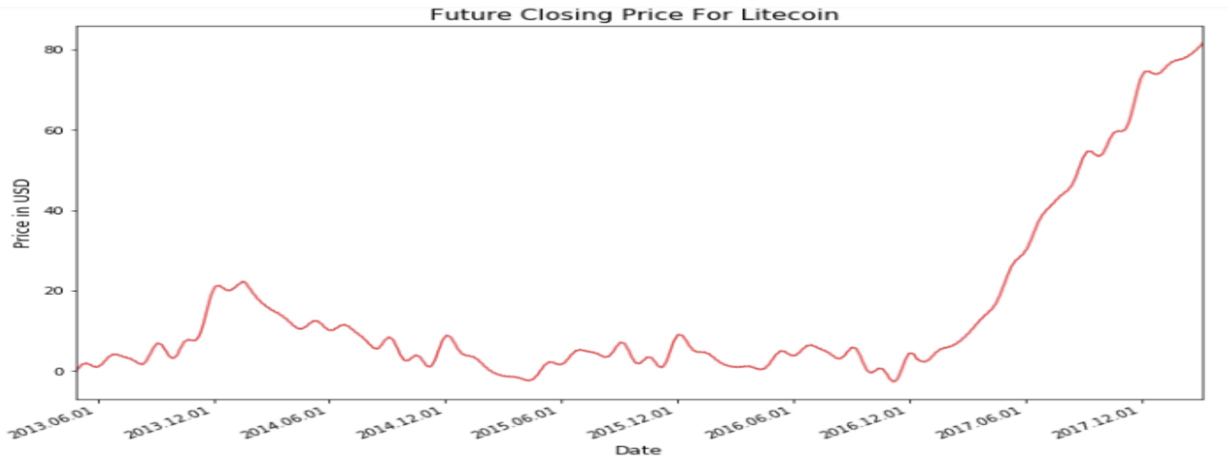


Fig4.38 Future Closing Price For Litecoin from 2013-2017

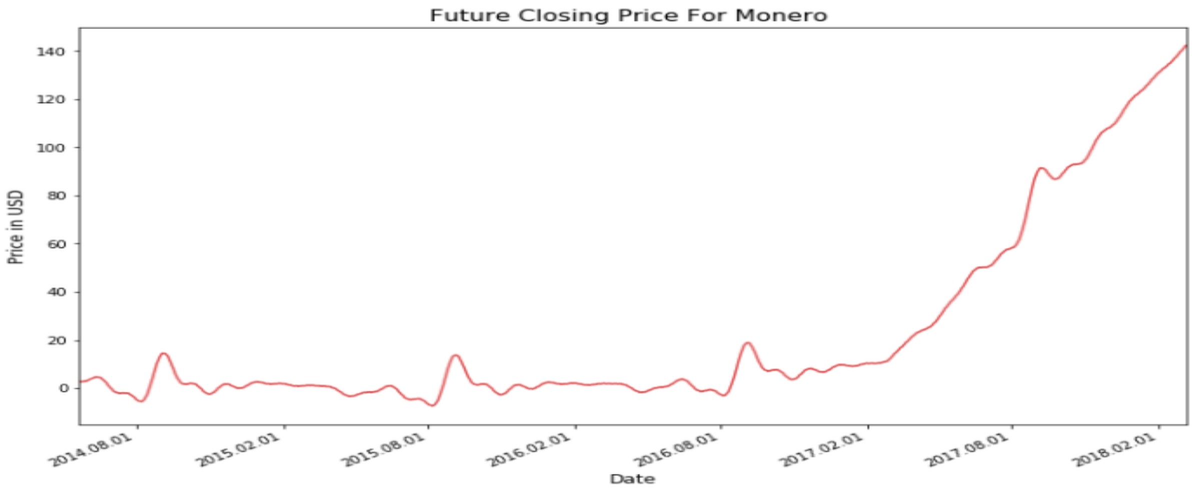


Fig4.39 Future Closing Price For Monero from 2014-2018

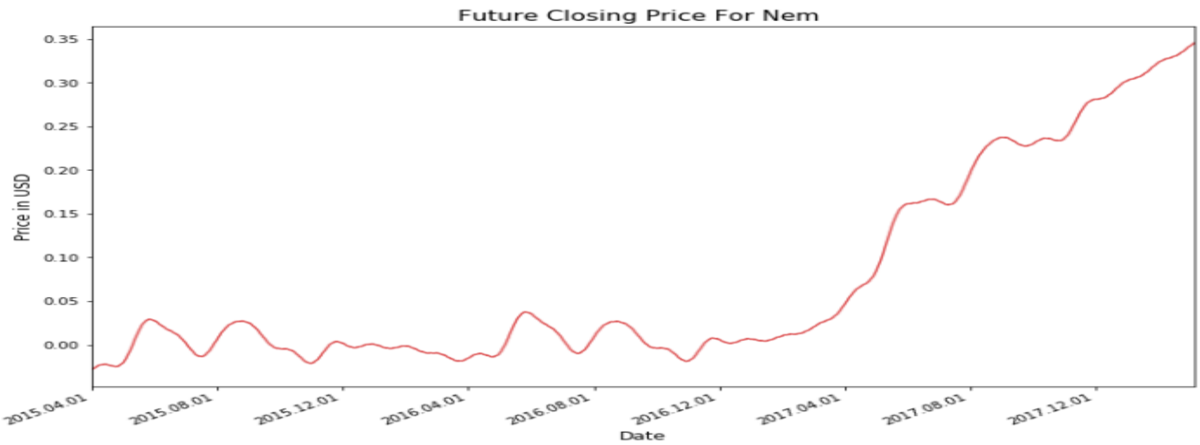


Fig4.40 Future Closing Price For Nem from 2015-17

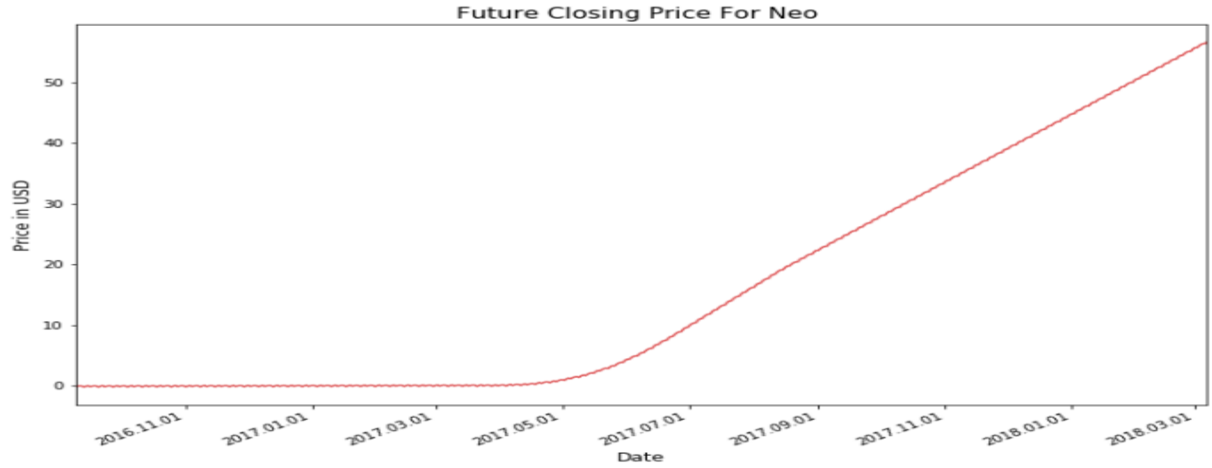


Fig4.41 Future Closing Price For Neo from 2016-18

Fig4.42 Future Closing Price For Numeraire from 2017-18

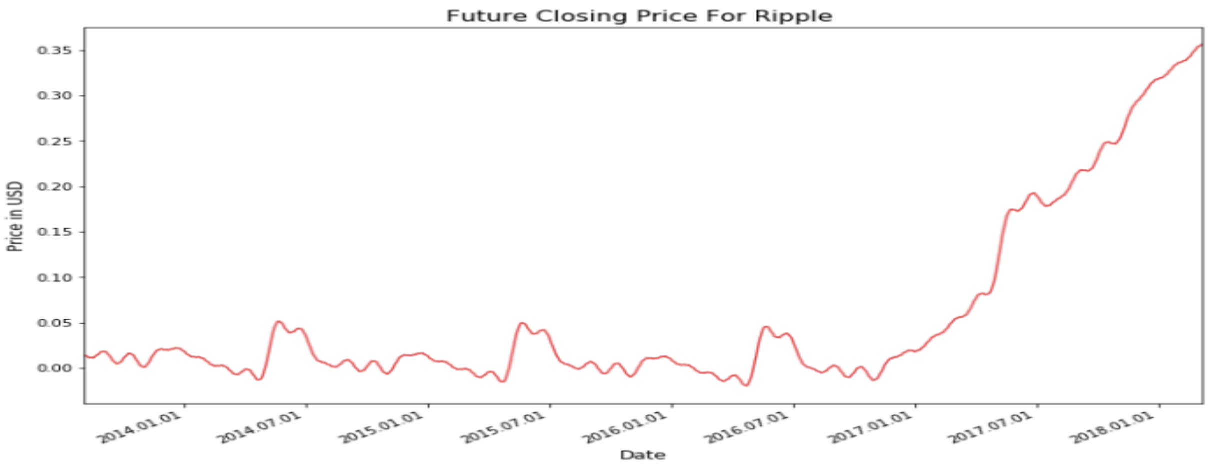


Fig4.43 Future Closing Price For Ripple from 2014-18

Fig4.44 Future Closing Price For Stratis from 2016-18

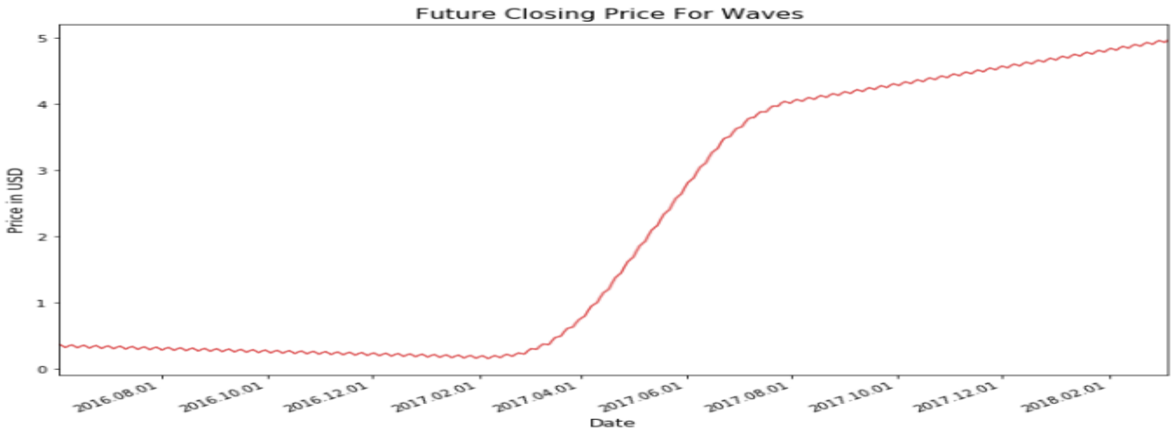


Fig4.45 Future Closing Price For Waves from 2016-18

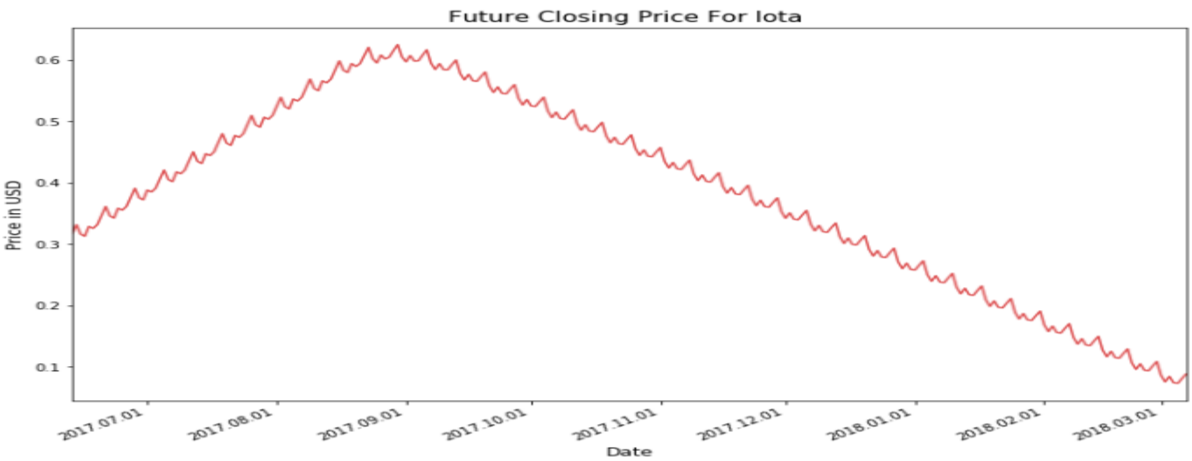


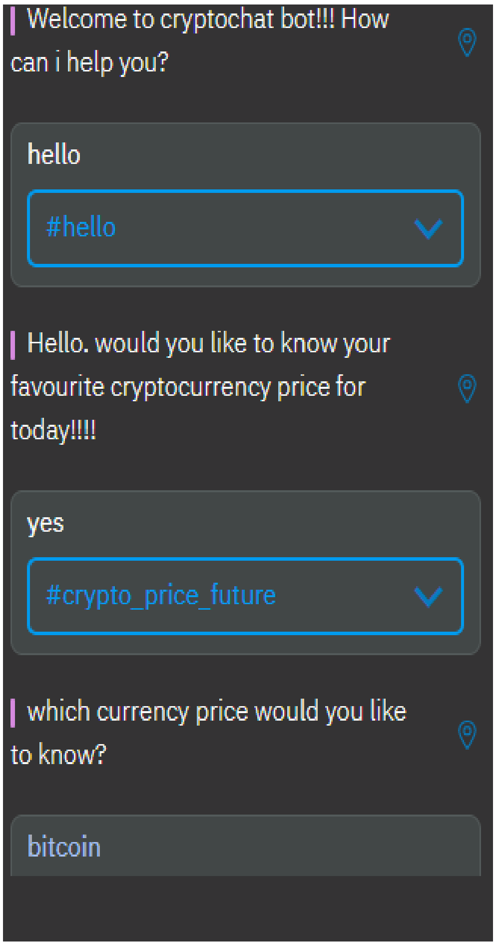
Fig4.46 Future Closing Price For Iota from2017-18

4.1.6 Create the interactive cryptocurrency chatbot and build a client application

Using the results of the forecasted dataframe of fifteen different cryptocurrency prices, a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

Steps in implementing the cryptocurrency chatbot:

- Log in to IBM Watson Assistant account
- Create a workspace in the Watson Assistant
- Describe the workspace
- Create intents and entities related to the chatbot
- Create the dialog flow and train the bot according to the predicted user inputs
- Build a client application
- Getting the service information by knowing the credentials of the workspace.
- Communicating with the Watson Assitant service and processing user inputs to detect intents and entities by maintaining state.
- Implementing cryptocurrency app actions.



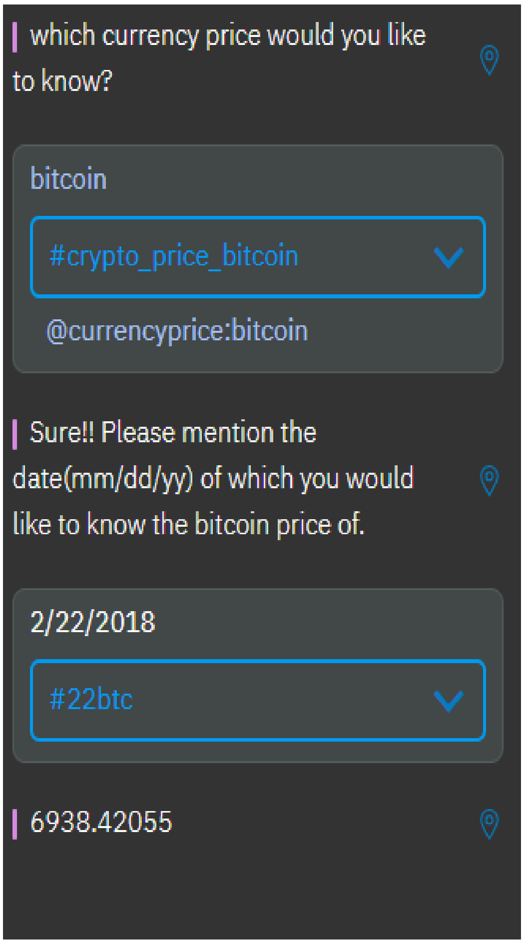


Fig4.46 Interactive Cryptocurrency ChatBot trained for cryptocurrency prices of past ,present and future values of fifteen different cryptocurrencies

TESTING

For testing , all the respective cryptocurrency dataset is divided into training and testing datasets. ARIMA package have been used to build the model and train on historical data.The fitted model is used to forecast the future values of the prices and generate the forecasted dataset.The mean square error value is calculated along with the root mean square error values. This is implemented for fifteen different cryptocurrency dataset. Forecast accuracy and error values are also calculated for the same. The following table summarises the test results of the analysis done:

Table 4.4 Test Results For Forecast Accuracy And Forecast Error

Sno.	Cryptocurrency dataset	RMSE	Forecast Accuracy	Forecast error
1.	Dash	0.709	87.355	12.644
2.	Ethereum	0.488	51.580	48.419
3.	Iota	0.047	97.504	2.495
4.	Waves	0.093	95.790	4.209
5.	Stratis	0.008	77.944	22.055
6.	Bitcoin_cash	40.297	95.323	4.676
7.	Bitcoin	0.503	98.496	1.504

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

CONCLUSION

Predictive analysis of cryptocurrency prices is a boon for investors for future predictions of the price fluctuations. ARIMA package have been used to build the model and train on historical data. The popular ARIMA model for time series forecasting is implemented for fifteen different cryptocurrency datasets and also plotted future fluctuation graphs for them using the forecasted datasets which was generated using Facebook's package Prophet().Forecast accuracy and error values are calculated for the fifteen different cryptocurrency datasets. Using these results a cryptochat bot has been implemented using IBM Watson Assistant, an interactive platform where the users can get to know the present, past and future prices of cryptocurrencies. This platform would help the investors know better about the ups and downs of the financial world which indeed would help them to make better investment.

FUTURE SCOPE

- To increase the time period of the forecasted datasets and create a dataframe which consis t of increased number of future closing price values. It would enhance the future predictio n of the fluctuating cryptocurrency prices and increase the plotting efficiency of the foreca sted values of the closing price .
- To train the bot with more forecasted values and predicted user inputs in order to make it efficient to serve the investors with better investing ideas.
- To consider fundamental factors which affects the fluctuations of cryptocurrency prices.

APPENDIX A- SOURCE CODE

#Closing Price Of Various Cryptocurrencies

```
for coin in crypto_data:
df = pd.DataFrame(crypto_data[coin])
df = df[['Date' , 'Close']]
df['Date_mpl'] = df['Date'].apply(lambda x: mdates.date2num(x)) # making new column 'Date_mpl' by using date2num lamba function
fig, ax = plt.subplots(figsize=(6,4))
sns.tsplot(df.Close.values, time=df.Date_mpl.values, alpha=0.8, color=color[3], ax=ax)
ax.xaxis.set_major_locator(mdates.AutoDateLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y.%m.%d'))
fig.autofmt_xdate()
plt.xlabel('Date', fontsize=12)
plt.ylabel('Price in USD', fontsize=12)
title_str = "Closing price distribution of " + coin
plt.title(title_str, fontsize=15)
plt.show()
```

#Analysing Of Candlestick Chart For Various Cryptocurrency

```
for coin in crypto_data:
df = pd.DataFrame(crypto_data[coin])
fig = plt.figure(figsize=(6,4))
ax1 = plt.subplot2grid((1,1), (0,0))
df['Date_mpl'] = df['Date'].apply(lambda x: mdates.date2num(x))
temp_df = df[df['Date']>'2017-05-01']
ohlc = []
for ind, row in temp_df.iterrows():
ol=[row['Date_mpl'],row['Open'],row['High'],row['Low'],row['Close'], row['Volume']]
ohlc.append(ol)
```

candlestick_ohlc(ax1, ohlc, width=0.4, colorup='#77d879', colordown='#db3f3f')

ax1.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))

ax1.xaxis.set_major_locator(mticker.MaxNLocator(10))

plt.xlabel("Date", fontsize=12)

plt.xticks(rotation='vertical')

plt.ylabel("Price in USD", fontsize=12)

title_str = "Candlestick chart for " + coin

plt.title(title_str, fontsize=15)

plt.subplots_adjust(left=0.09, bottom=0.20, right=0.94, top=0.90, wspace=0.2, hspace=0)

plt.show()

#**Correlation Map For Various Cryptocurrency**

#Spearman Correlation Map

df = pd.DataFrame()

currency_name = []

df['Date'] = crypto_data['bitcoin'].Date

df = df[df['Date']>'2017-05-01']

for coin in crypto_data:

currency_name.append(coin)

temp_df = crypto_data[coin]

df[coin] = temp_df[temp_df['Date']>'2017-05-01'].Close

temp_df = df[currency_name]

corrmat = temp_df.corr(method='spearman')

fig, ax = plt.subplots(figsize=(5, 5))

sns.heatmap(corrmat, vmax=1., square=True)

plt.title("Spearman correlation map", fontsize=15)

plt.show()

temp_df.corr(method='spearman')

#**Pearson Correlation Map**

df = pd.DataFrame()

```
currency_name = []
df['Date'] = crypto_data['bitcoin'].Date
df = df[df['Date']>'2017-05-01']
for coin in crypto_data:
currency_name.append(coin)
temp_df = crypto_data[coin]
df[coin] = temp_df[temp_df['Date']>'2017-05-01'].Close
temp_df = df[currency_name]
corrmat = temp_df.corr(method='pearson')
fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(corrmat, vmax=1., square=True)
plt.title("Pearson correlation map", fontsize=15)
plt.show()
temp_df.corr(method='pearson')
```

#Kendall Correlation Map

```
df = pd.DataFrame()
currency_name = []
df['Date'] = crypto_data['bitcoin'].Date
df = df[df['Date']>'2017-05-01']
for coin in crypto_data:
currency_name.append(coin)
temp_df = crypto_data[coin]
df[coin] = temp_df[temp_df['Date']>'2017-05-01'].Close
temp_df = df[currency_name]
corrmat = temp_df.corr(method='kendall')
fig, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(corrmat, vmax=1., square=True)
```

#Time Series Forecasting using fbprophet()

```
for coin in crypto_data:
df = pd.DataFrame(crypto_data[coin])
temp_df = pd.DataFrame()
```

```
temp_df['ds'] = df['Date']
temp_df['y'] = df['Close']
temp_df['ds'] = temp_df['ds'].dt.to_pydatetime()
model = Prophet()
model.fit(temp_df)
future = model.make_future_dataframe(periods = 60)
forecast = model.predict(future)
title_str = "predicted value of "+ coin
model.plot(forecast, uncertainty=False)
model.plot_components(forecast, uncertainty=False)
```

#**Time series Forecasting using ARIMA model**

```
# fit model
model = ARIMA(df_bitcoin, order=(5,1,0))
model_fit = model.fit(disp=0)
print(model_fit.summary())
# plot residual errors
residuals = DataFrame(model_fit.resid)
residuals.plot()
plt.show()
residuals.plot(kind='kde')
plt.show()
print(residuals.describe())
  X = df_bitcoin.values
  size = int(len(X) * 0.80)
  train, test = X[0:size], X[size:len(X)]
  history = [x for x in train]
  predictions = list()
  for t in range(len(test)):
      model = ARIMA(history, order=(5,1,0))
      model_fit = model.fit(disp=0)
      output = model_fit.forecast()
      yhat = output[0]
      predictions.append(yhat)
      obs = test[t]
```

```
    history.append(obs)
    error = mean_squared_error(test, predictions)
    RMSE=error**0.5
    print("Root means sqauare error is ",RMSE)
#calcualting error
ar=test-predictions
np.absolute(ar)
ar=ar/test
ar=ar*100
forecast_error=(sum(ar))/(len(ar))
print("forecast error is ")
print (forecast_error)
arr=[100]
print ("forecast accuracy is")
print(arr-forecast_error)
#calculation of error done
print('Test MSE: %f' % error)
print('Test RMSE: %f' % RMSE)
# plot
plt.plot(test)
plt.plot(predictions, color='red')
plt.show()
```

#Future Graphs Of Fifteen Different Cryptocurrencies

```
forecast.to_csv('numeraire_forecast.csv')
import matplotlib.dates as mdates
#matplotlib.dates.datestr2num
df = pd.read_csv('./numeraire_forecast.csv')
#plt.plot(data.yhat, data.ds)
import matplotlib.dates as mdates
df['Date_mpl'] = df['ds'].apply(lambda x: mdates.datestr2num(x))
fig, ax = plt.subplots(figsize=(12,8))
ax.xaxis.set_major_locator(mdates.AutoDateLocator())
```

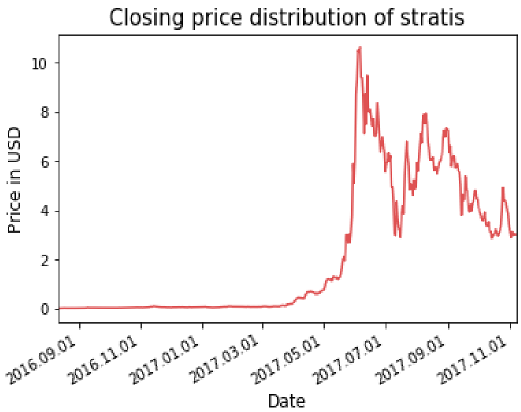
```
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y.%m.%d'))
fig.autofmt_xdate()
plt.xlabel('Date', fontsize=12)
plt.ylabel('Price in USD', fontsize=12)
plt.title("Future Closing Price For Numeraire ", fontsize=15)
plt.show()
```

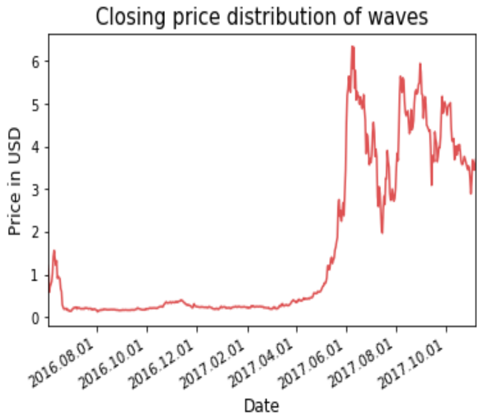
#ChatBot Implementation

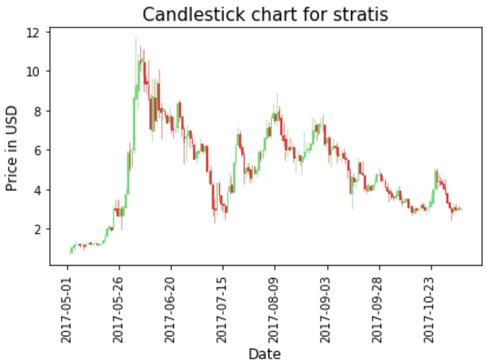
```
# sets up service wrapper, sends initial message, and
# receives response.
import watson_developer_cloud
conversation = watson_developer_cloud.ConversationV1(
username = '531b64f4-1023-4a46-b81a-f289c6e334b4', # replace with
username from service key
password = 'EeL8NB8f2wZR', # replace with password from service
key
version = '2017-05-26')
workspace_id = '3b77dedf-de9f-41b4-a46d-45a9706ed976' # replace
with workspace ID
# Start conversation with empty message.
response = conversation.message(
workspace_id = workspace_id,
input = {
'text': ''})
# Print the output from dialog, if any.
if response['output']['text']:
print(response['output']['text'][0])
# maintains state.
import watson_developer_cloud
# Set up Conversation service.
conversation = watson_developer_cloud.ConversationV1(
username = '531b64f4-1023-4a46-b81a-f289c6e334b4',
# replace with username from service key
password = 'EeL8NB8f2wZR', # replace with password from service
key
```

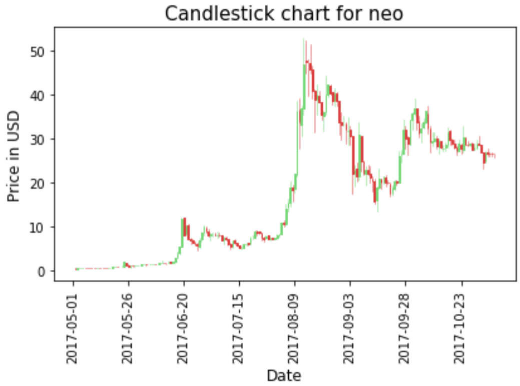
```
version = '2017-05-26')
workspace_id = '3b77dedf-de9f-41b4-a46d-45a9706ed976' # replace
with workspace ID
# Initialize with empty value to start the conversation.
user_input = ''
context = {}
# Main input/output loop
while True:
# Send message to Conversation service.
response = conversation.message(
workspace_id = workspace_id,
input = {
'text': user_input},
context = context)
# If an intent was detected, print it to the console.
if response['intents']:
print('Detected intent: #' + response['intents'][0]['intent'])
```

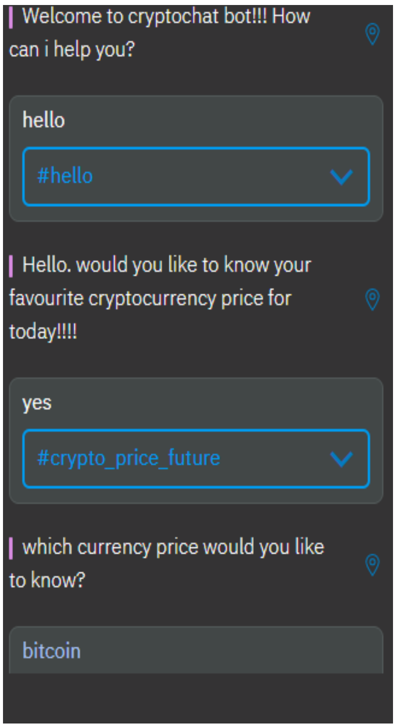
APPENDIX B- SCREENSHOTS

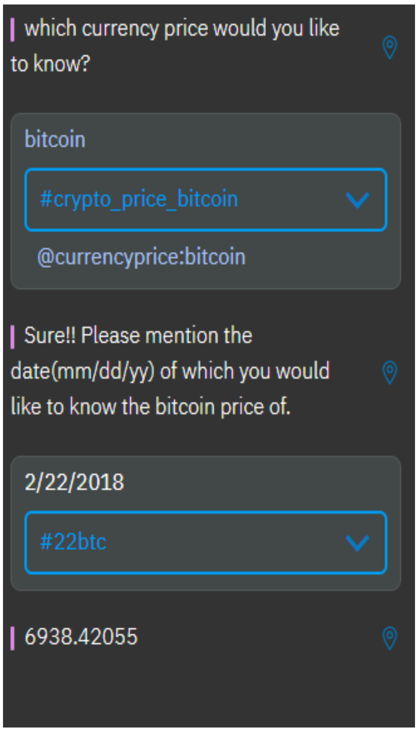












REFERENCES

[1] Kim, Y.B., Kim, J.G., Kim, W., Im, J.H., Kim, T.H., Kang, S.J. and Kim, C.H.,(2016). Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS one*, *11*(8), p.e0161197.

[2] Mai, F., Bai, Q., Shan, Z., Wang, X. and Chiang, R.,(2015). From bitcoin to big coin: The impacts of social media on bitcoin performance. *SSRN Electronic Journal*.

[3] Indera, N.I., Yassin, I.M., Zabidi, A. and Rizman, Z.I.,(2017). Non-linear autoregressive with exogeneous input (NARX) Bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators. *Journal of Fundamental and Applied Sciences*, *9*(3S), pp.791-808.

[4] Guo, L. and Li, X.J.,(2017). Risk Analysis of Cryptocurrency as an Alternative Asset Class. In *Applied Quantitative Finance*(pp. 309-329). Springer, Berlin, Heidelberg.

[5] Catania, L., Grassi, S. and Ravazzolo, F.,(2018). Predicting the Volatility of Cryptocurrency Time–Series.

[6] McNally, S.,(2016). *Predicting the price of Bitcoin using Machine Learning* (Doctoral dissertation, Dublin, National College of Ireland).

[7] McNally, Sean. "Predicting the price of Bitcoin using Machine Learning." PhD diss., Dublin, National College of Ireland, (2016).