**AN INDUSTRY ORIENTED MINI PROJECT REPORT**

# On

# CRYPTO CURRENCY PRICE PREDICTION USING LINEAR REGRESSION AND TIME SERIES

Submitted in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

**By**

**Sabrina Shaik 17261A1251**

Under the guidance of

**Mrs. J.Aruna Shanthi**

Assistant Professor, IT.



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**MAHATMA GANDHI INSTITUTE OF TECHNOLOGY**

**(Affiliated to JNTUH, Hyderabad; Six UG Programs Accredited by NBA 3times Accredited by NAAC with ‘A’ Grade)**

**Kokapet (vill), Gandipet (Mandal), Ranga Reddy (Dist.), Chaitanya Bharathi P.O., Hyderabad-500 075.**

**2020-2021**



**CERTIFICATE**

This is to certify that the project work entitled **Crypto currency Price Prediction using Linear Regression and Time Series s**ubmitted by **Sabrina Shaik (17261A1251) Banu Sindhu Etikala (17261A1211)** and **Samreen Begum (17261A1254)** in partial fulfilment of requirements for the award of degree of Bachelor of Technology in Information Technology as specialization is a record of the bonafide work carried out under the supervision of Mrs. **J.Aruna Shanthi**, and this has not been submitted to any other university or institute for award of degree or diploma.

Project Guide Project Coordinators

### Mrs. J. Aruna Shanthi Mrs. J. Aruna Shanthi

Assistant Professor Assistant Professor

Dept. of IT Dept. of IT

**EXTERNAL EXAMINER DR. D. VIJAYA LAKSHMI**

Professor and HOD, Dept. of IT

# DECLARATION

We hereby declare that the research work entitled **CRYPTOCURRENCY PRICEPREDICTION USING LINEAR REGESSION AND TIME SERIES** is original and bonafide work carried out by us as a part of fulfilment for Bachelor of Technology in Information Technology, Mahatma Gandhi Institute of Technology, Gandipet, Hyderabad, under the guidance of **Mrs. J. Aruna Shanthi**, Asst.Prof., Department of IT, MGIT.

No part of the project work is copied from books/journals/internet and wherever the partition is taken, the same has been duly referred in the text. The reported is based on the project work done entirely by us and not copied from any other source.

**SABRINA SHAIK-17261A1251**

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

We would like to express our sincere gratitude and indebtedness to our project guide **Mrs. J. Aruna Shanthi,** Assistant Professor, Dept. of IT**,** who has supported us throughout our project with immense patience and expertise.

We are also thankful to the honorable Principal of MGIT **Prof. K. JAYA SANKAR** and **DR. D. VIJAYA LAKSHMI,** HOD and professor, Department of IT, for providing excellent infrastructure and a conducive atmosphere for completing this project successfully.

We are also extremely thankful to our Project Coordinator Mrs.CH.Lakshmi **Kumari**, Assistant Professor, Dept. of IT, for her valuable suggestions and interest throughout the course of this project.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our families for their support all through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

**SABRINA SHAIK-17261A1251**

# ABSTRACT

Bitcoin is a cryptocurrency which is utilized worldwide for computerized investments. Bitcoin and Ethereum both are decentralized currencies which are not possessed by any authority. Trades made by Bitcoins are simple as they are not fixing to any nation. Crypto-currency such as Bitcoin is more popular these days among investors. In the proposed work, it is studied to forecast the Bitcoin price precisely considering different parameters that influence the Bitcoin price. This study first handles, it is identified the price trend on day-by-day changes in the Bitcoin price while it gives knowledge about Bitcoin price trends. The dataset till current date is taken with open, high, low and close price details of Bitcoin value. Exploiting the dataset machine learning module is introduced for prediction of price values. The aim of this work is to conduct Technical Analysis on Crypto currency price fluctuations by plotting data graph and candlestick graph. The future predictions of cryptocurrency like bitcoin and Ethereum are predicted by applying machine learning algorithms like linear regression and time series analysis

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Figure name** | **Page Number** |
| 1 | 4.1: Architecture | 10 |
| 2 | 4.2: Flowchart | 11 |
| 3 | 5.1: Use Case Diagram | 15 |
| 4 | 5.2: Sequence Diagram | 16 |
| 5 | 5.3: Class Diagram | 17 |
| 6 | 5.4: Activity Diagram | 18 |
| 7 | 5.5: Collaboration Diagram | 19 |
| 8 | 5.6: Component diagram | 20 |
| 9 | 5.7: Deployment Diagram | 21 |
| 10 | 7.1: Bitcoin Dataset | 35 |
| 11 | 7.2: Ethereum Dataset | 35 |
| 12 | 7.3: Deploying on Local Host | 36 |
| 13 | 7.4: Running on Local Host | 36 |
| 14 | 7.5: Data Table: Bitcoin | 37 |
| 15 | 7.6: Dropdown | 37 |
| 16 | 7.7: Candlestick for Technical Analysis | 38 |
| 17 | 7.8: Linear Regression Prediction | 38 |
| 18 | 7.9: Time Series predicted values | 39 |
| 19 | 7.10: Time Series Predicted Curve | 39 |

# TABLE OF CONTENTS

Certificate ii

Declaration iii

Declaration iii

Acknowledgement iv

Abstract v

List of Figures vi

List of Tables vii

1. Introduction 1

1.1 Objective 2

1.2 Problem Definition 3

1.3 Existing System 3

1. 4. Proposed System 3

2. Literature Survey 4

3. System Requirements 7

3.1 Software Requirements 8

3.2 Hardware Requirements 8

4. System Analysis 9

4.1 Proposed Architecture/ block diagram 10

4.2 Flowchart 11

5. System Design 12

5.1 DFDs / E-R Diagrams 13

5.2 UML Diagrams 14

5.2.1 Use Case Diagram 15

5.2.2 Sequence Diagram 16

5.2.3 Class Diagram 17

5.2.4 Activity Diagram 18

5.2.5 Collaboration Diagram 19

5.2.6 Component Diagram 20

5.2.7 Deployment Diagram 21

5.3 Feasibility Study 22

5.3.1 Economic Feasibility 23

5.3.2 Operational Feasibility 23

6. Coding and Implementation 24

6.1 Sample Code 24

7. Results 34

7.1 Output Screens 35

8. Conclusion and Future work 40

9.1 Conclusion and Future Enhancement 41

9.2 References 42

# INTRODUCTION

1. **Introduction**

Bitcoin is a crypto-currency which is utilized worldwide for computerized investment. Bitcoin is decentralized for example it sn't possessed by anybody. Trades made by Bitcoins are simple as they are not fixing to any nation. Investment is possible through different commercial centers known as bitcoin trades. These enable individuals to trade on Bitcoins utilizing various currencies. Mt. Gox is the biggest Bitcoin exchange, where bitcoin is stored as a virtual digital bank. The record of the considerable number of exchanges, the timestamp information handled in this market is called Blockchain.

Each record of blockchain information is encrypted. Trades done by the client’s name are made private only wallet ID is made open. The Bitcoin's simply and similar like a stock but in different way. There are various algorithm using machine learning a reutilized-on price prediction on stock value. The features influencing Bitcoin are unique. For investors it is mandatory to predict Bitcoin prices. Bitcoin price do not affect by business announcements or government announcements and it is not at all like securities exchange.

Thus, we exploit machine learning techniques to foresee the cost of Bitcoin. Bitcoin is an effective crypto currency brought into the monetary market dependent on its one-of-a-kind convention and Nakamoto's structure. Bitcoin try to achieve decentralization in currency market. Investors in Bitcoin market establish trust connections through the development of Blockchain using cryptography strategies. Bitcoin now-a-days gaining more interest due to innovations in Blockchain and machine learning...

* 1. **Objective** 
     + The cryptocurrencies can be used to transfer the ownership of assets on one name to another name by paying the seller through bitcoin.
     + Enables Confidential transactions
     + Provides strong security
     + The best part of cryptocurrency is that you will be the sole owner of private and public encryption keys
  2. **Problem Definition**

Crypto-currency such as Bitcoin is more popular these days among investors. In the proposed work, it is studied to forecast the Bitcoin price precisely considering different parameters that influence the Bitcoin price. This study first handles, it is identified the price trend on day by day changes in the Bitcoin price while it gives knowledge about Bitcoin price trends. The dataset till current date is taken with open, high, low and close price details of Bitcoin value. Exploiting the dataset machine learning module is introduced for prediction of price values. The aim of this work is to derive the accuracy of Bitcoin prediction using different machine learning algorithm and compare their accuracy. Experiment results are compared for decision tree and regression model.

**1.3 Existing System**

Bitcoin is a booming crypto-currency market, and various researches have been studied in fields of economics and price prediction. Bitcoin dataset is considered from 2011 to till 2017 price and applied machine learning models such Linear regression and Decision Tree models.

* 1. **Proposed System**

Bitcoin dataset is considered from 2011 to till 2017 price and applied machine learning models such Linear regression and Time series network. The proposed learning method suggest the best algorithm to choose and adopt for crypto currency prediction problem.

# 2. LITERATURE SURVEY

* + Many existing techniques have been studied by the researchers on crypto-currency market like fluctuations on its prices, social media sentiments, etc.
  + Fluctuations on crypto-currency prices attracted many researchers, Tian et.al [1] discussed the fluctuation on bitcoin prices by its execution orders such as buy or sell. They handled regression techniques and moving average values. They derived model for time series, which also applied Gaussian time model to predict the bitcoin values. However, they proved their model is efficient on time series data, in our proposed model, we considered dataset of various years and we evaluated based on close price of bitcoin.
  + Connor et.al [2] studied the bitcoin price through sentiments of various users provided on news columns and social media. Apart from bitcoin, they handled two more crypto-currency for prediction study. They applied feature selection and classification algorithm on the collected dataset along with token weights with positive and negative values. They used three models namely, Naïve Bayes, regression models and SVM (Support Vector machine). For bitcoin, their experiments shown regression model outperform the others.
  + Young Bin Kim [3] studied the price fluctuation model in crypto-currency using comments given by users. Apart from bitcoin, they considered Litecoin and Ripple, which are the next two major currencies. User crawled data is taken for their study and they segregated sentiment opinions of five types such as very positive, positive, neutral, negative, and very negative.
  + HUISU JANG et.al [4] on his study discussed Bayesian neural networks considering block chain environment. They did machine learning with linear regression model. Their study identifies the bitcoin price and fluctuations.
  + Anshul et.al [5] used LSTM for bitcoin price forecast. LSTM, one of recurrent neural network algorithm will allow training bitcoin prices as time series data efficiently. In their study, though the time taken for compilation of LSTM model is high than existing ARIMA model, the accuracy is found to be high for LSTM.
  + Jethin et.al [6] handled Ethereum price and bitcoin price using Google Trends data and twitter texts. Twitter data considered main form of source for arriving decisions on price prediction. Itis necessary to understand the effect of tweets on price forecast, which can give quick assessment on buy or sell suggestion to bitcoin traders. The author collected dataset from twitter using
  + hashtag #btc and Ethereum dataset using hashtag #eth. They also extracted sales volume index data from Google. They pre-processed data and this data is given as input for linear regression model for bitcoin price prediction.
  + The author in [7], proposed a model to identify whether Bitcoin price is depends on volume of tweets or posts by various authors in social media the author used Google price trend data and positive sentiment tweets as the study is related on price increase. The author collected around two months data of tweets, which is around one million data along with timeline is extracted. Tweet Timeline is compared with price of bitcoin on the same timeline, which has given correlations on this work.
  + The work proposed by author [8] is finding the sentiment using multiple machine learning techniques. The author collected twitter dataset and performed the analysis, they handle various stages of pre-processing techniques such as removing URL, spelling corrections, and emoticons are replaced with their corresponding polarity values. They used classification techniques such as Naive Bayes algorithm and Support Vector Machine. Using NLTK package, they find the polarity of tweets, from which they arrived positive and negative sentiment.
  + The author [9] proposed finding anomalies on market. They observed that movement varies throughout the week, over the course of the year and after some time. Strikingly, request and supply increase/decline with the goal that costs are somewhat consistent after some time. They observed that costs are by and large lower on Sundays with the goal that imminent purchasers should move their interest right up 'til today of the week.
  + In the existing work [10], the authors considered daily price trend of crypto currency, particularly on bitcoin market, considering various features. They applied more than three normalization techniques on dataset, which is collected fromquandl.com. Next, they proposed the feature selection problem, in which they considered five features fed into machine learning algorithm such as Bayesian regression, random forest.

## 3. SYSTEM REQUIREMENTS

**3.1 Hardware Requirements:**

RAM : 4GB or 8GB

**3.2 Software Requirements:**

Frontend Language : HTML, CSS

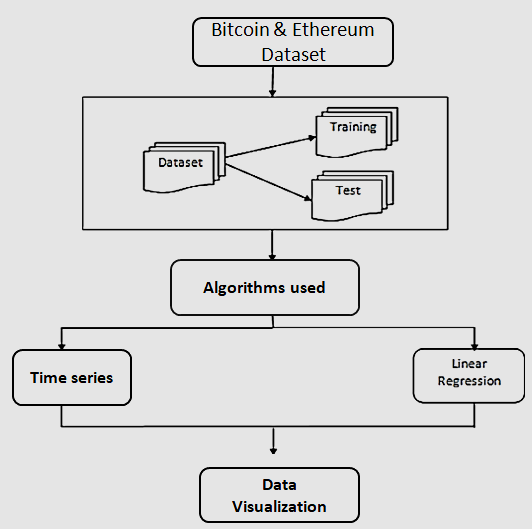
Backend Language : Python

Deployment Tool : Dash

Operating System : Windows 10

## 4. SYSTEM ANALYSIS

**4.1 Proposed Architecture**

Figure 4.1: Architecture

**4.2 Flowchart**

Converting attributers into NumPy array

Reshaping the rows & columns

Price Prediction with Linear Regression

CV

Calculating Epoch and Loss

Price Prediction with Time series

Start

Stop

Fix seed

Figure 4.2: Flowchart

1. **STSTEM DESIGN**

**5.1 System Design:**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage. During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

### 5.2 UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

* Actors
* business processes
* (logical) components
* Activities
* programming language statements
* database schemas, and
* Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (workflows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

**5.2.1Use Case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

The use cases are represented by either circles or ellipses.

While a [use](https://en.wikipedia.org/wiki/Use_Case) [case](https://en.wikipedia.org/wiki/Use_Case) itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system.

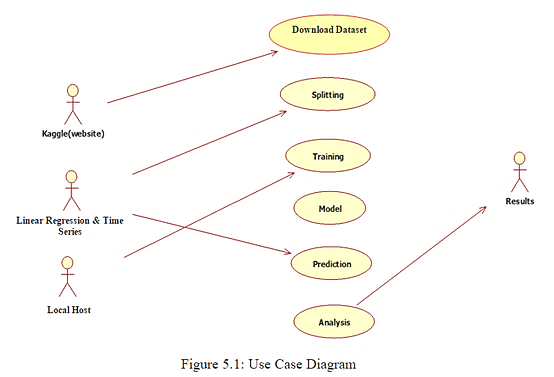
It has been said before that "Use case diagrams are the blueprints for your system".

They provide the simplified and graphical representation of what the system must actually do.

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders The drawings attempt to mimic the real world and provide a view for the [stake holder](https://en.wikipedia.org/wiki/Project_stakeholder) to understand how the system is going to be designed.

The purpose of the use case diagrams is simply to provide the high level view of the system and convey the requirements in laypeople's terms for the [stake holders.](https://en.wikipedia.org/wiki/Project_stakeholder)

Additional diagrams and documentation can be used to provide a complete functional and technical view of the system.



**5.2.2 Sequence diagram:**

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

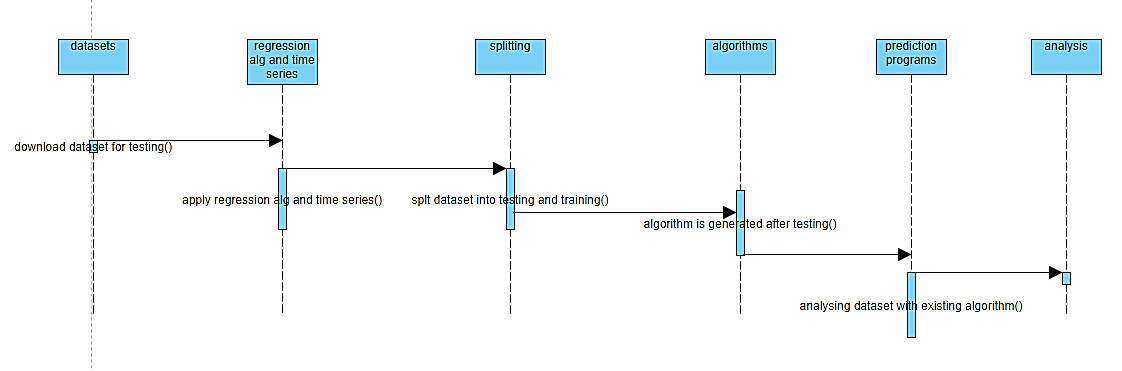


Figure 5.2: Sequence Diagram

**5.2.3 Class Diagram:**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

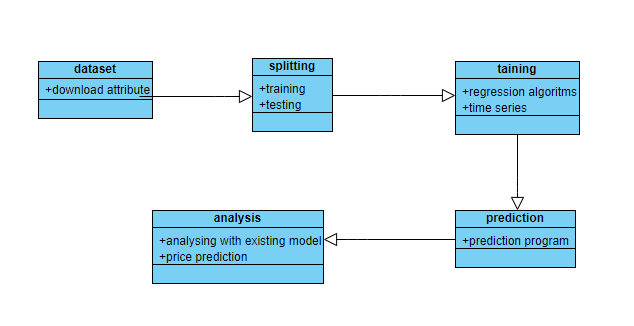


Figure 5.3: Class Diagram

**5.2.4 Activity diagram:**

Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. It is also called **object-oriented flowchart.** Activity diagrams consist of activities that are made up of actions which apply to behavioral modeling technology.

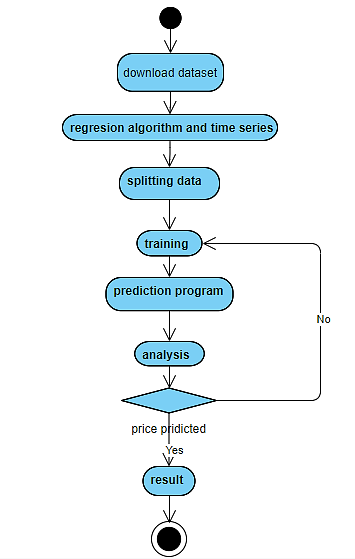


Figure 5.4: Activity Diagram

**5.2.5. Collaboration Diagram**

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language. These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

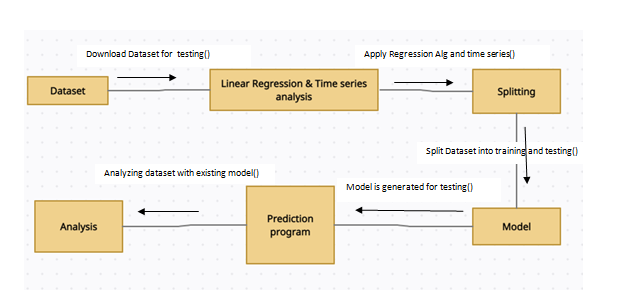


Figure 5.5 collaboration diagram

**5.2.6 Component Diagram:**

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that are often used to model the static implementation view of a system.

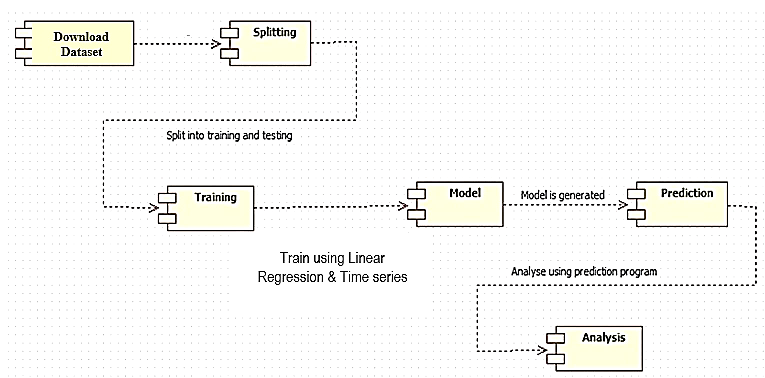


Figure 5.6: Component diagram

**5.2.7 Deployment Diagram:**

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

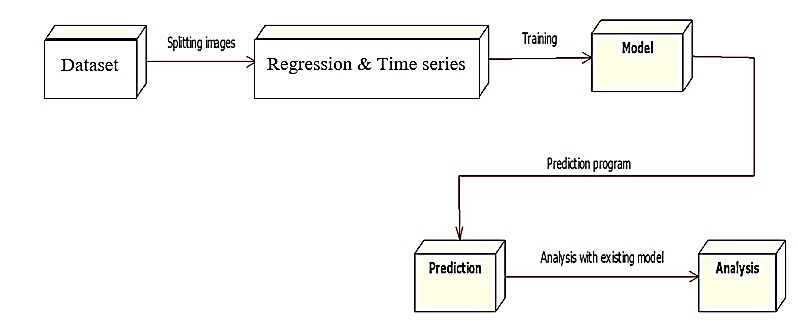


Figure 5.7: Deployment Diagram

**5.3 Feasibility Study**

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economic feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility

#### 5.3.1 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

#### 5.3.2 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

#### 5.3.3 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Does the proposed equipment have the technical capacity to hold the data required to use the new system?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security? Model is generated using Linear Regression algorithm and Time series neural network with the help of super computers. Accuracy of the model is nearly 70 percent. For any given crypto-currency dataset it predicts the price forecast of the cryptocurrency. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology.

# CODING AND IMPLEMENTATION

**SAMPLE CODE:**

import dash

import dash\_table

import datetime

import dash\_core\_components as dcc

import dash\_html\_components as html

from sklearn.linear\_model import LinearRegression

from dash.dependencies import Input, Output, State

import pandas as pd

import plotly.graph\_objs as go

import numpy

from keras.models import Sequential

from keras.layers import Dense

import math

df = pd.read\_csv('Gemini\_BTCUSD\_daily.csv')

dft = pd.read\_csv('Gemini\_ETHUSD\_daily.csv')

external\_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(\_\_name\_\_, external\_stylesheets=external\_stylesheets)

app.layout = html.Div([

html.Div([

# Title

html.Div([

html.H1(children='Crypto Currency Prediction')

], style={'textAlign': 'center'}),

# Drop down and data table

html.Div([

html.H3(children='Crypto Currency Type'),

dcc.Dropdown(

id='table-dropdown',

options=[

{'label': 'Bitcoin', 'value': 'BTC'},

{'label': 'Ethirium', 'value': 'ETH'}

],

value='BTC'

),

dcc.Markdown('''

#### Choose one of the crypto currency

Historical data of crypto currency daily price including it's highest, lowest, open, close, etc.

All data is taken from [Crypto Data](http://www.cryptodatadownload.com/data/northamerican/).

Data will be shown in both a table and graph.

'''),

html.H3(children='Data Table'),

dash\_table.DataTable(

id='table',

data=df.to\_dict('records'),

columns=[{'id': c, 'name': c} for c in df.columns],

fixed\_rows={'headers': True, 'data': 0},

style\_cell={'width': '150px'}

)],

),

html.Div([

html.H3(children='Data Graph'),

dcc.Graph(id='data-graph')

]),

html.Div([

html.H3(children='Candlestick'),

dcc.Graph(id='data-candlestick')

]),

html.Div([

html.H2(children='Prediction using linear regression'),

dcc.Markdown('''

####

'''),

dcc.Graph(id='graph')

]),

html.Div([

html.H2(children='Time series prediction'),

dcc.Markdown('''

#### Time series prediction with deep learning

'''),

dcc.Graph(id='graph-learn')

]),

]),

])

@app.callback(Output('table', 'data'), [Input('table-dropdown', 'value')])

def update\_rows(value):

# dff = df[df['some-column'] == value]

if value == 'BTC':

return df.to\_dict('records')

elif value == 'ETH':

return dft.to\_dict('records')

@app.callback(Output('data-graph', 'figure'), [Input('table-dropdown', 'value')])

def update\_data\_graph(value\_dropdown):

if value\_dropdown == 'BTC':

df\_graph = df

text = 'Bitcoin'

elifvalue\_dropdown == 'ETH':

df\_graph = dft

text = 'Ethirium'

return {

'data': [go.Scatter(

mode='lines',

x=df\_graph['Date'],

y=df\_graph['Open'],

text=text,

marker={

'size': 15,

'opacity': 0.5,

'line': {'width': 0.5, 'color': 'blue'}

}

)],

'layout': go.Layout(

xaxis={

'title': 'Date',

'tickvals': [df\_graph["Date"].iloc[0], df\_graph["Date"].iloc[250], df\_graph["Date"].iloc[500],

df\_graph["Date"].iloc[750], df\_graph["Date"].iloc[1000], df\_graph["Date"].iloc[-1]],

# 'tickformat': '%m/%y'

},

yaxis={

'title': 'Price : USD',

},

margin={'l': 40, 'b': 40, 't': 10, 'r': 0},

hovermode='closest'

)

}

@app.callback(Output('data-candlestick', 'figure'), [Input('table-dropdown', 'value')])

def update\_graph(value\_dropdown):

if value\_dropdown == 'BTC':

df\_to\_candlestick = df

elifvalue\_dropdown == 'ETH':

df\_to\_candlestick = dft

return {

'data': [go.Candlestick(

x=df\_to\_candlestick['Date'],

open=df\_to\_candlestick['Open'],

high=df\_to\_candlestick['High'],

low=df\_to\_candlestick['Low'],

close=df\_to\_candlestick['Close'],

)],

}

def datetime\_to\_float(d):

epoch = datetime.datetime.utcfromtimestamp(0)

total\_seconds= (d - epoch).total\_seconds()

# total\_seconds will be in decimals (millisecond precision)

return total\_seconds

@app.callback(Output('graph', 'figure'), [Input('table-dropdown', 'value')])

def update\_graph(value\_dropdown):

if value\_dropdown == 'BTC':

df\_to\_graph = df

df\_graph = df

text = 'Bitcoin'

X = df.iloc[:, 0].values.reshape(-1, 1) # values converts it into a numpy array

Y = df.iloc[:, 4].values.reshape(-1, 1) # -1 means that calculate the dimension of rows, but have 1 column

elifvalue\_dropdown == 'ETH':

df\_to\_graph = dft

df\_graph = dft

text = 'Ethirium'

X = dft.iloc[:, 0].values.reshape(-1, 1) # values converts it into a numpy array

Y = dft.iloc[:, 4].values.reshape(-1, 1) # -1 means that calculate the dimension of rows, but have 1 column

linear\_regressor = LinearRegression() # create object for the class

linear\_regressor.fit(X, Y) # perform linear regression

Y\_pred = linear\_regressor.predict(X) # make predictions

df\_for\_graph = pd.DataFrame(Y\_pred, columns=['Prediction'])

return {

'data': [go.Scatter(

name='Linear Regression Prediction',

mode='lines',

x=df\_to\_graph['Date'],

y=df\_for\_graph['Prediction'],

text='Linear Regression Line',

marker={

'size': 15,

'opacity': 0.5,

'line': {'width': 0.5, 'color': 'blue'}

}

),

go.Scatter(

name='Actual Data',

mode='lines',

x=df\_graph['Date'],

y=df\_graph['Open'],

text=text,

marker={

'size': 15,

'opacity': 0.5,

'line': {'width': 0.5, 'color': 'red'}

})

],

'layout': go.Layout(

xaxis={

'title': 'Date',

'tickvals': [df\_to\_graph['Date'].iloc[0], df\_to\_graph["Date"].iloc[250], df\_to\_graph["Date"].iloc[500],

df\_to\_graph["Date"].iloc[750], df\_to\_graph["Date"].iloc[1000], df\_to\_graph["Date"].iloc[-1]

],

},

yaxis={

'title': 'Price : USD',

},

margin={'l': 40, 'b': 40, 't': 10, 'r': 0},

hovermode='closest'

)

}

# convert an array of values into a dataset matrix

def create\_dataset(dataset, look\_back=1):

dataX, dataY = [], []

for i in range(len(dataset) - look\_back - 1):

a = dataset[i:(i + look\_back), 0]

dataX.append(a)

dataY.append(dataset[i + look\_back, 0])

return numpy.array(dataX), numpy.array(dataY)

@app.callback(Output('graph-learn', 'figure'), [Input('table-dropdown', 'value')])

def update\_graph\_learn(value\_dropdown):

if value\_dropdown == 'BTC':

csv\_name = 'Gemini\_BTCUSD\_daily.csv'

df\_to\_learn = df

elifvalue\_dropdown == 'ETH':

csv\_name = 'Gemini\_ETHUSD\_daily.csv'

df\_to\_learn = dft

# fix random seed for reproducibility

numpy.random.seed(7)

# load the dataset

dataframe = pd.read\_csv(csv\_name, usecols=[6], engine='python')

dataset = dataframe.values

dataset = dataset.astype('float32')

# split into train and test sets

train\_size = int(len(dataset) \* 0.67)

test\_size = len(dataset) - train\_size

train, test = dataset[0:train\_size, :], dataset[train\_size:len(dataset), :]

# reshape dataset

look\_back = 3

trainX, trainY = create\_dataset(train, look\_back)

testX, testY = create\_dataset(test, look\_back)

# create and fit Multilayer Perceptron model

model = Sequential()

model.add(Dense(12, input\_dim=look\_back, activation='relu'))

model.add(Dense(8, activation='relu'))

model.add(Dense(1))

model.compile(loss='mean\_squared\_error', optimizer='adam')

model.fit(trainX, trainY, epochs=60, batch\_size=2, verbose=2)

# Estimate model performance

trainScore = model.evaluate(trainX, trainY, verbose=0)

print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))

testScore = model.evaluate(testX, testY, verbose=0)

print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))

# generate predictions for training

trainPredict = model.predict(trainX)

testPredict = model.predict(testX)

# shift train predictions for plotting

trainPredictPlot = numpy.empty\_like(dataset)

trainPredictPlot[:, :] = numpy.nan

trainPredictPlot[look\_back:len(trainPredict) + look\_back, :] = trainPredict

# shift test predictions for plotting

testPredictPlot = numpy.empty\_like(dataset)

testPredictPlot[:, :] = numpy.nan

testPredictPlot[len(trainPredict) + (look\_back \* 2) + 1:len(dataset) - 1, :] = testPredict

dataset = pd.DataFrame(dataset, columns=['Data'])

trainPredictPlot = pd.DataFrame(trainPredictPlot, columns=['Train'])

testPredictPlot = pd.DataFrame(testPredictPlot, columns=['Test'])

return {

'data': [go.Scatter(

mode='lines',

x=df\_graph['Date'],

y=df\_graph['Open'],

text=text,

marker={

'size': 15,

'opacity': 0.5,

'line': {'width': 0.5, 'color': 'blue'}

),

go.Scatter(

name='Training',

mode='lines',

x=df\_to\_learn['Date'],

y=dataset['Train'],

text='Training',

marker={

'size': 15,

'opacity': 0.5,

'line': {'width': 0.5, 'color': 'green'}

}),

go.Scatter(

name='Test',

mode='lines',

x=df\_to\_learn['Date'],

y=dataset['Test'],

text='Test',

marker={

'size': 15,

'opacity': 0.5,

'line': {'width': 0.5, 'color': 'red'}

})

],

'layout': go.Layout(

xaxis={

'title': 'Date',

'tickvals': [df\_to\_learn['Date'].iloc[0], df\_to\_learn["Date"].iloc[250], df\_to\_learn["Date"].iloc[500],

df\_to\_learn["Date"].iloc[750], df\_to\_learn["Date"].iloc[1000], df\_to\_learn["Date"].iloc[-1]

],

},

yaxis={

'title': 'Price : USD',

},

margin={'l': 40, 'b': 40, 't': 10, 'r': 0},

hovermode='closest'

)

}

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True)

# 7. RESULTS

**7.1 Datasets used**

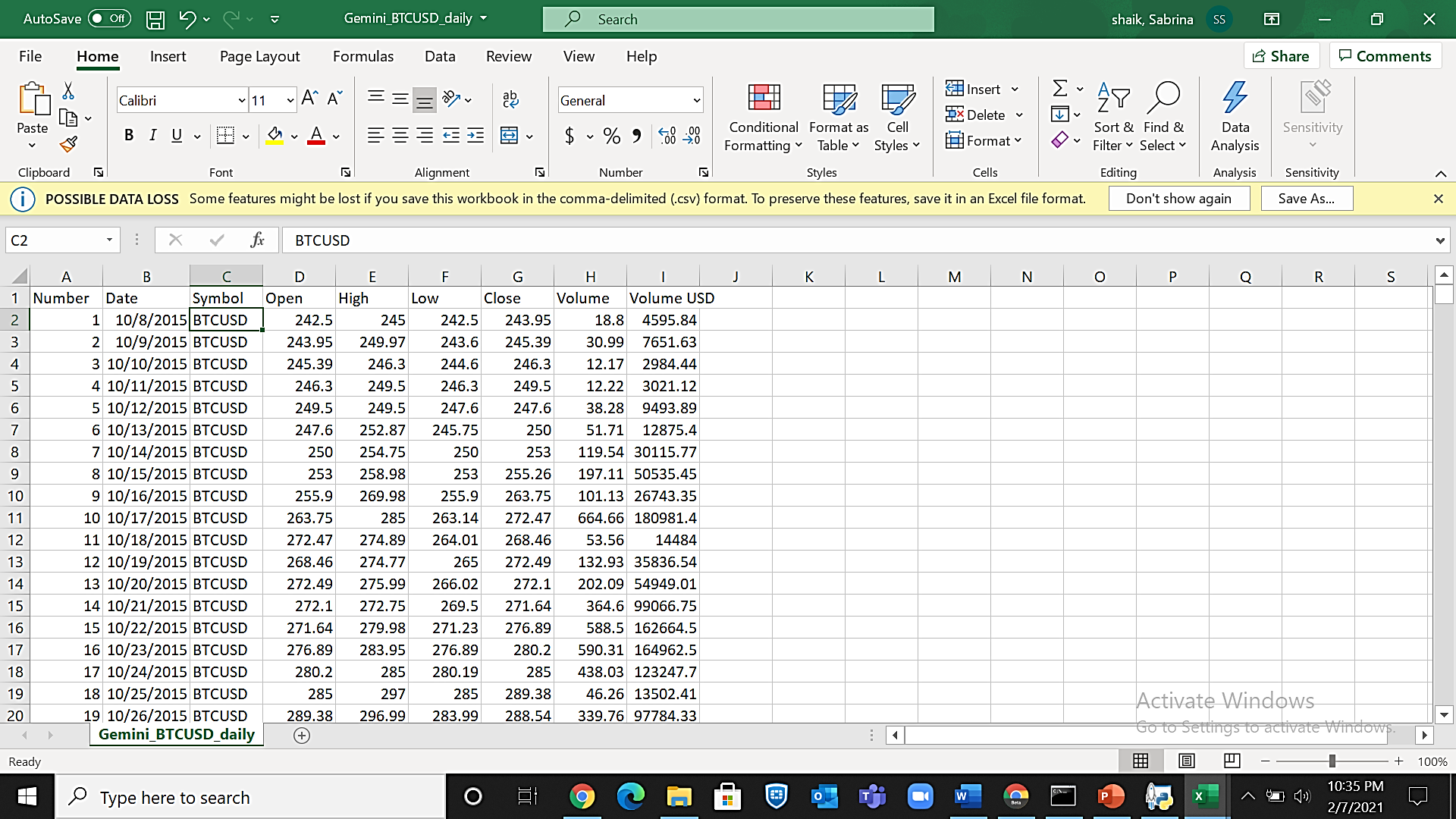


Figure7.1: Bitcoin Dataset

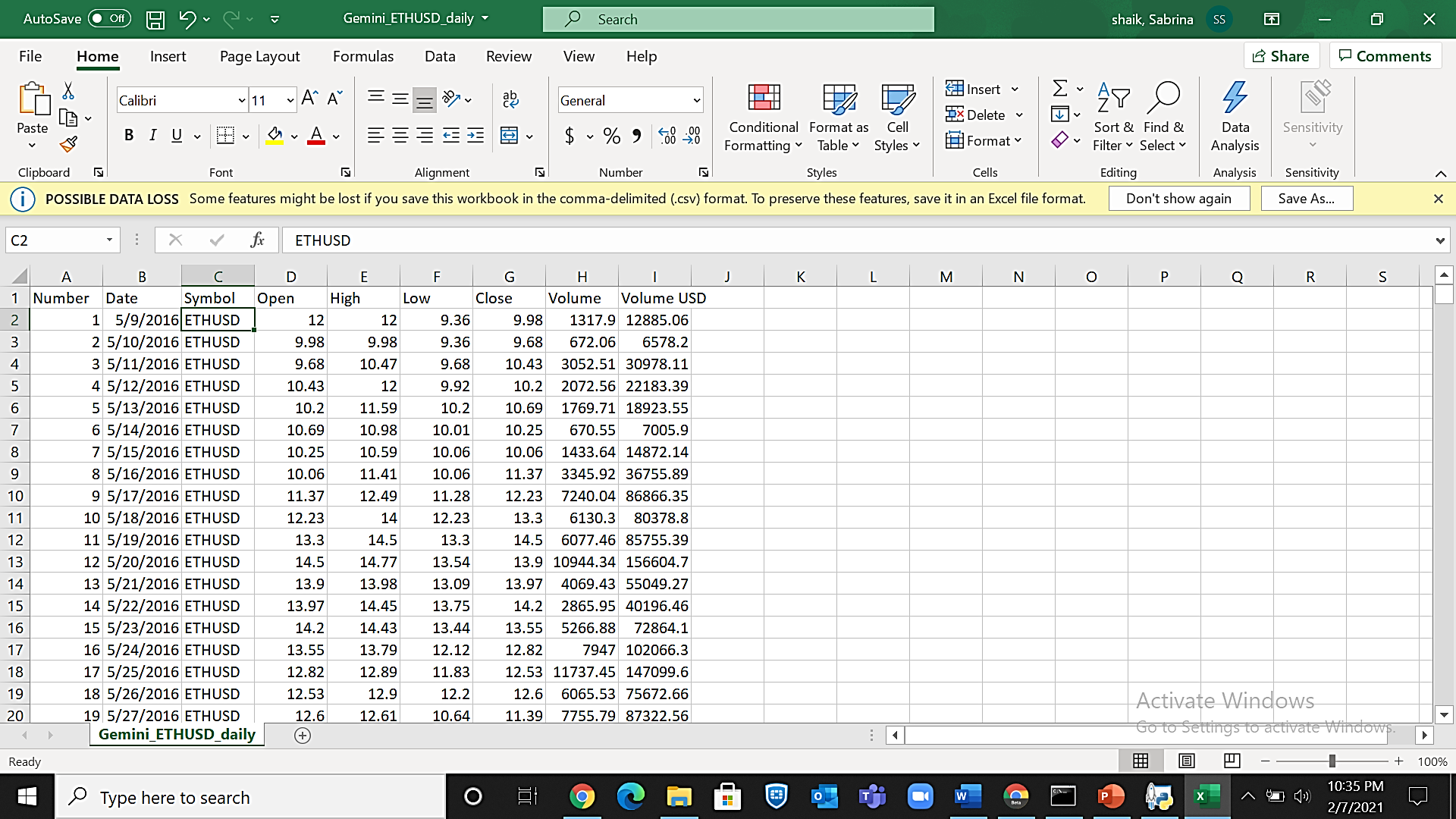


Figure 7.2: Ethereum Dataset

**8.1 Output Screen**

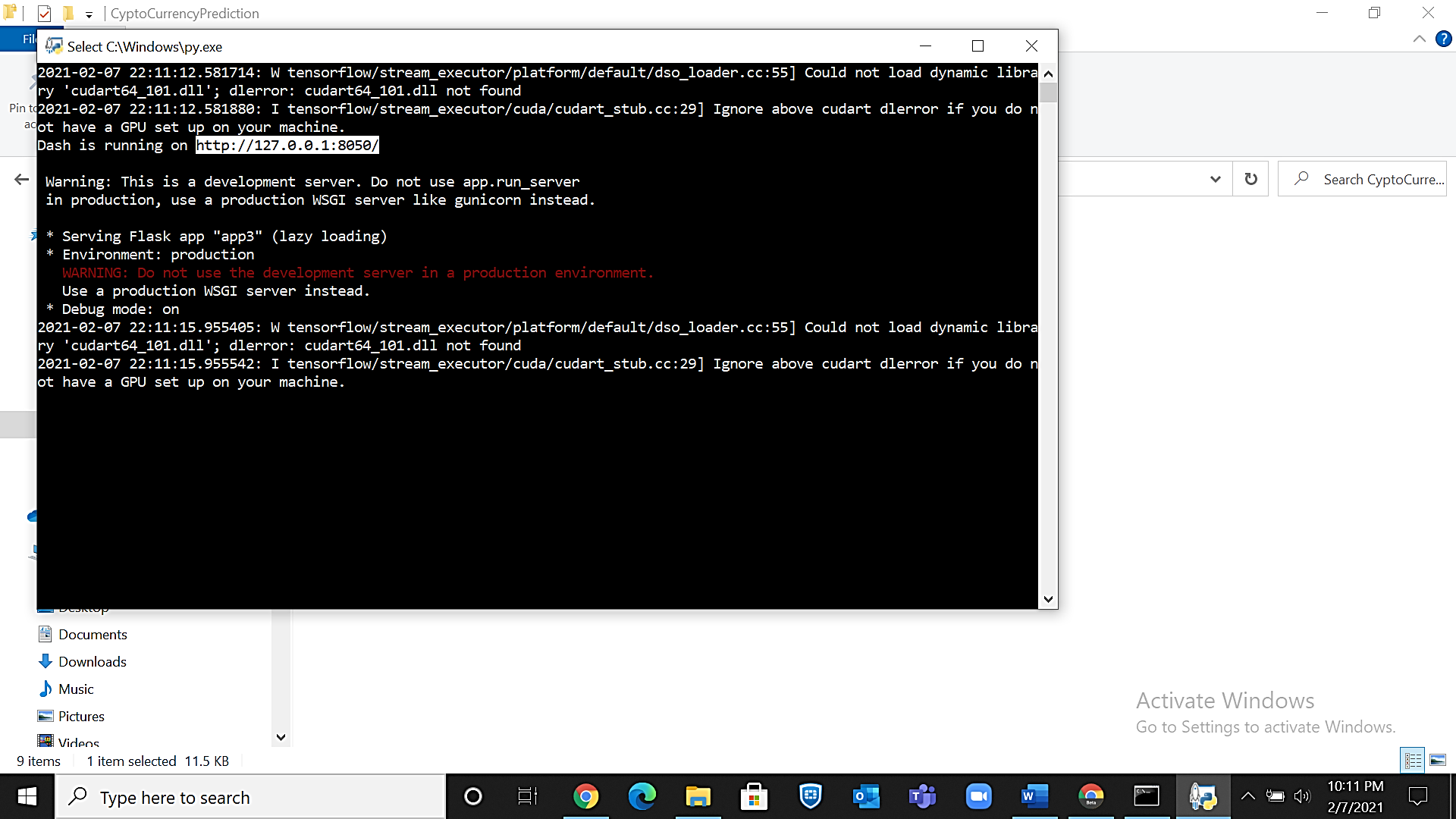


Figure 7.3: Deploying on Local Host

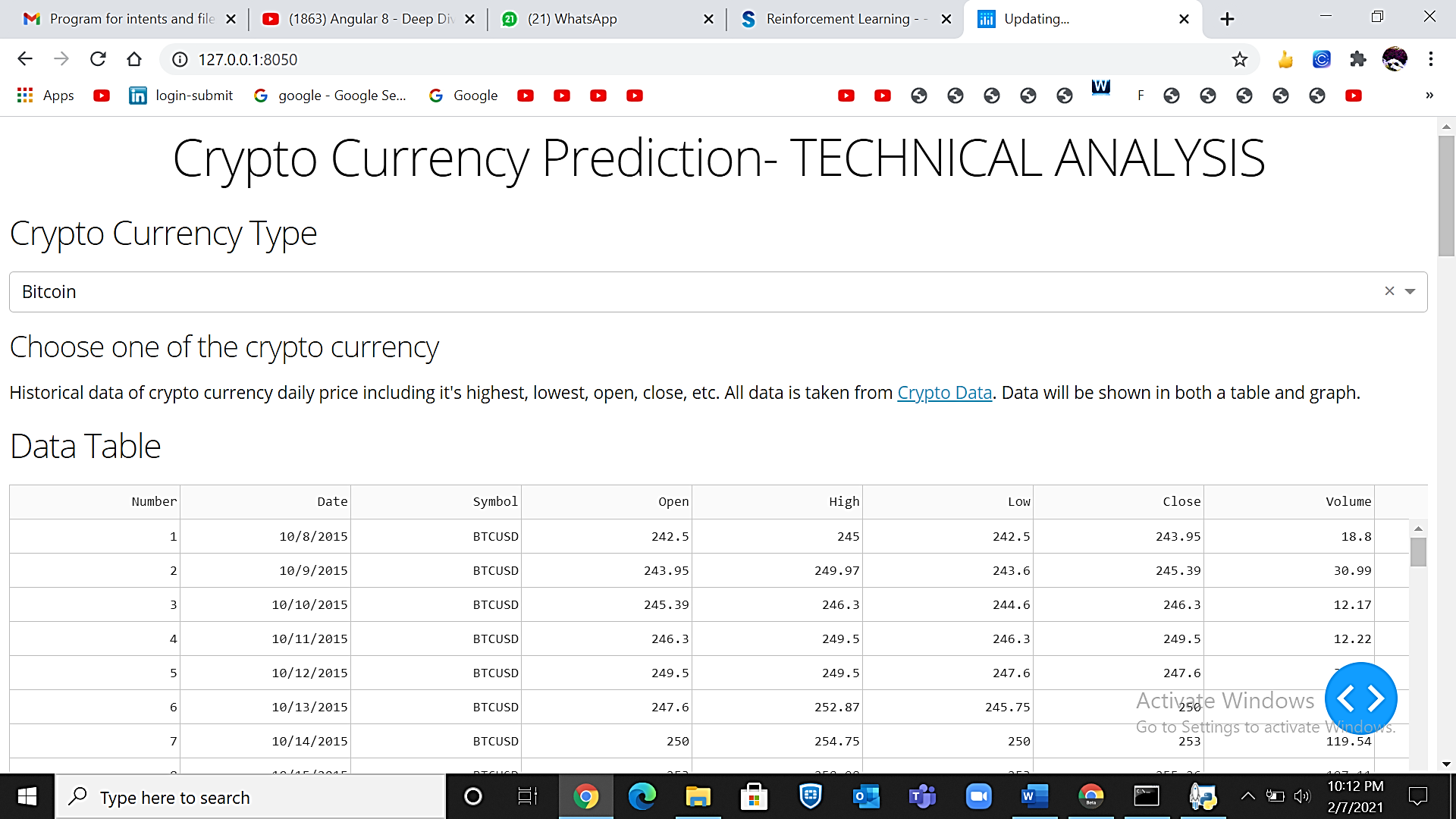


Figure 7.4: Running on Local Host

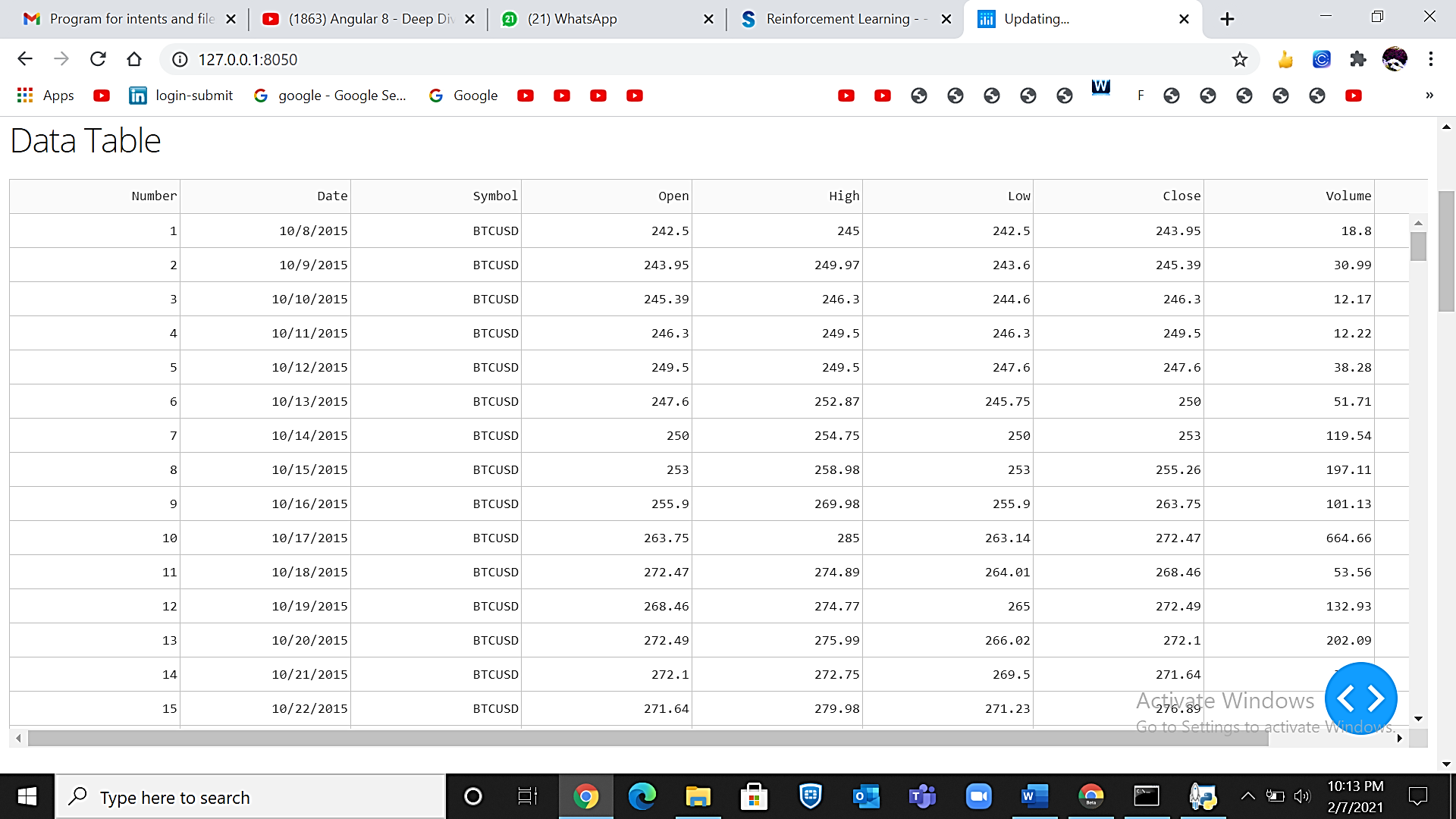


Figure 7.5: Data Table: Bitcoin

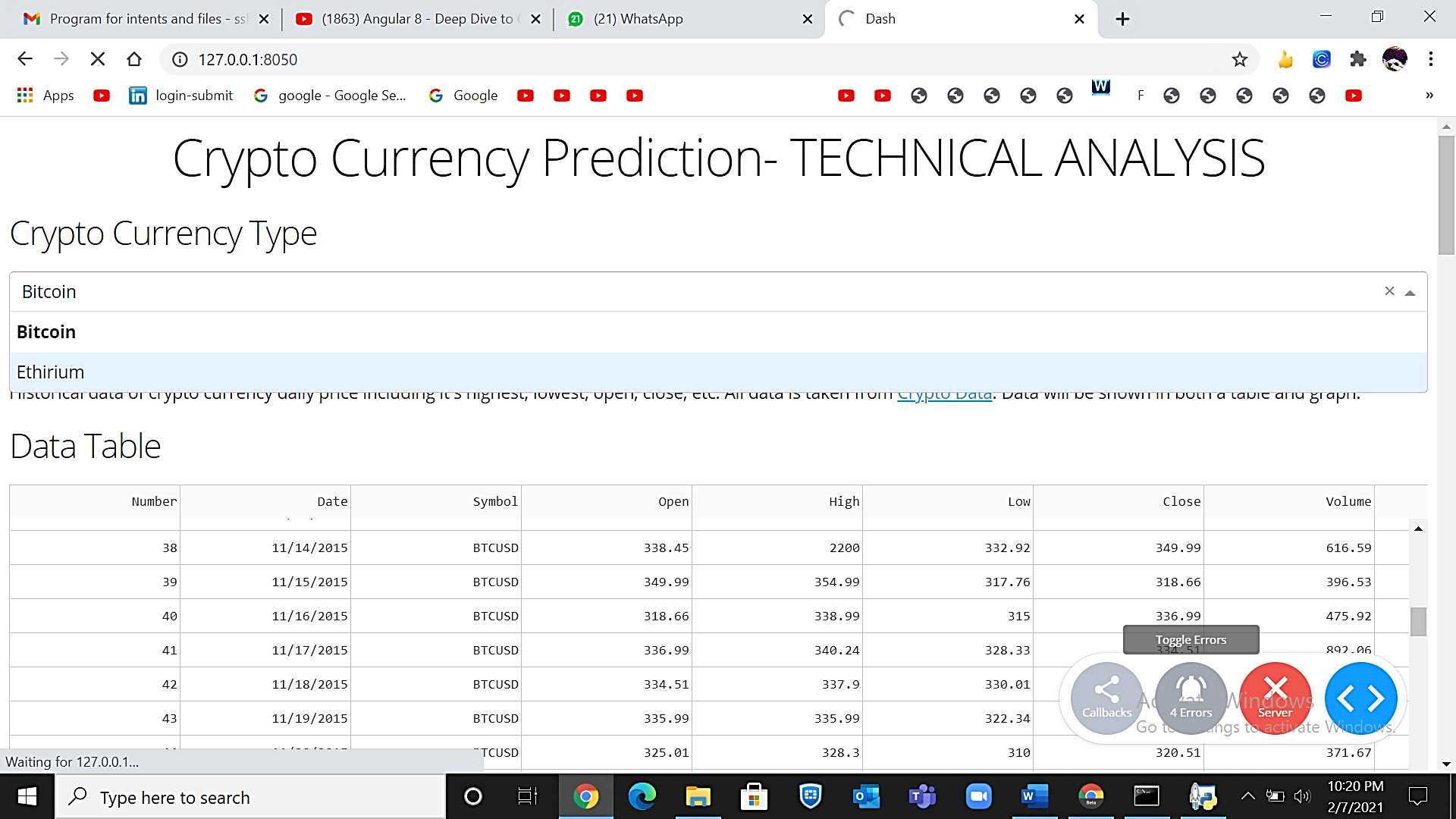


Figure 8.4: Dropdown



Figure 7.6: Data Graph of original dataset



Figure 7.7: Candlestick for Technical Analysis

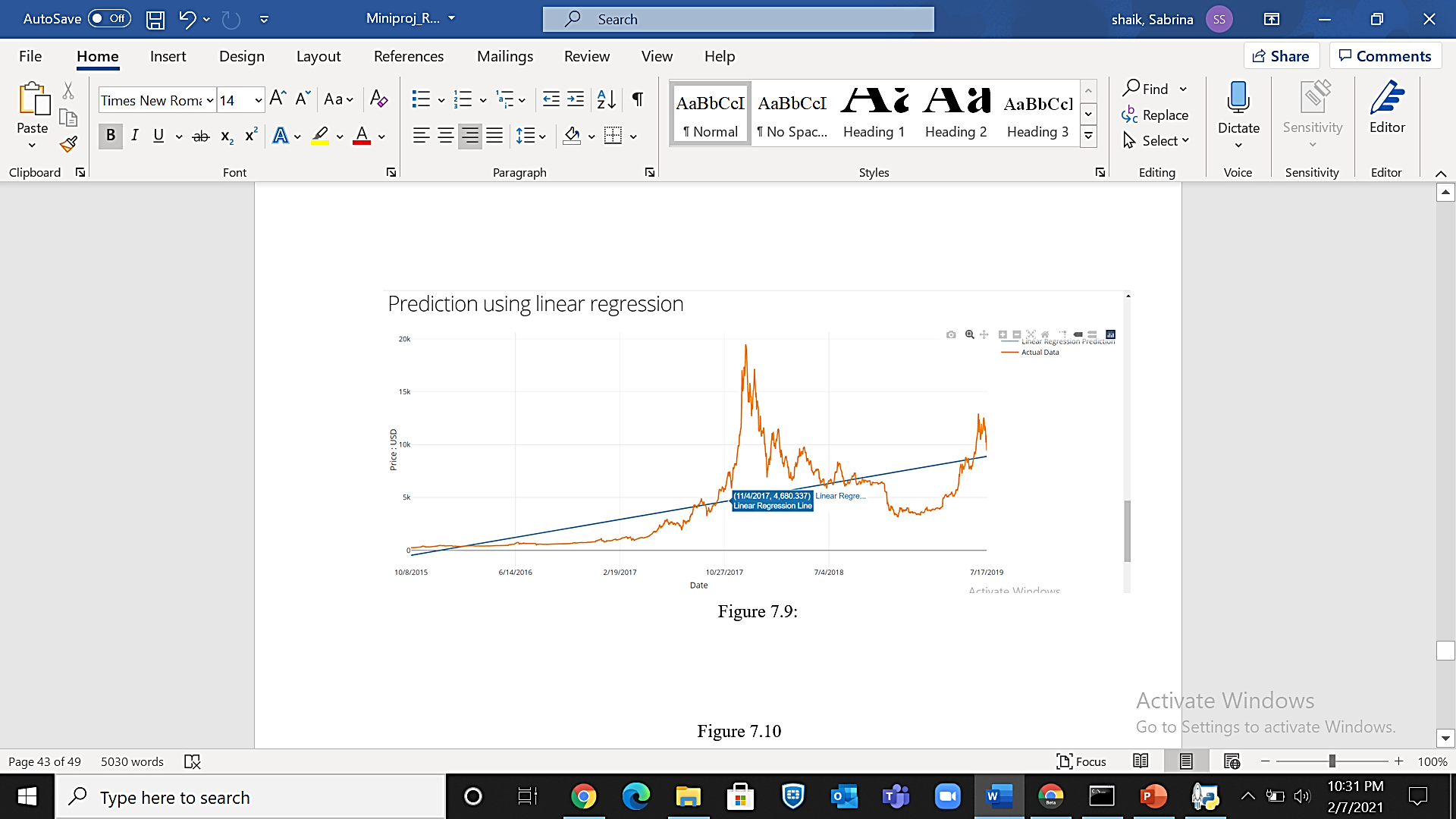


Figure 7.8: Linear Regression Prediction

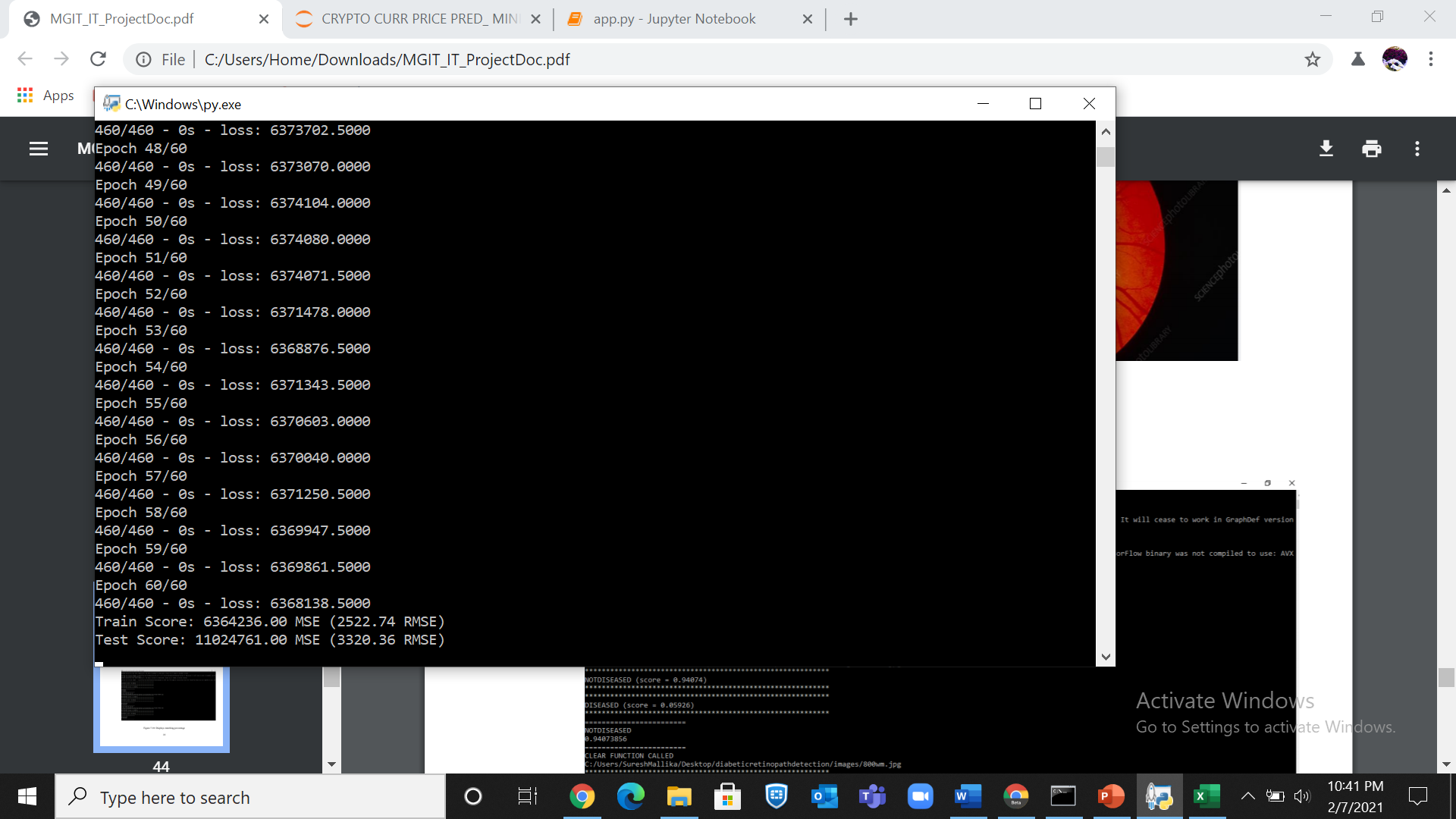


Figure: 7.9: Time Series predicted values



Figure 7.10: Time Series Predicted Curve

**8. CONCLUSION AND FUTURE WORK**

**8.1 Conclusion and Future Enhancement:**

Bitcoin is a booming crypto-currency market, and various research have been studied in fields of economics and price prediction. In our proposed work, Bitcoin dataset is considered from 2011 to till date price and applied machine learning models such as Time Series and Linear regression models. Also, the price forecast for five days is done using Time Series and Linear regression models. Technical analysis involves using real-world data to try to predict the future of the market. It involves looking at past statistics of the cryptocurrencies in question, including factors like volume and movement. The proposed system focuses on creating an interactable dashboard consisting of charts, to predict the future market of cryptocurrencies using real-world data of Bitcoin and Ethereum. The proposed system results as Time series deep learning model to predict the most accurate forecast of cryptocurrency. This system can be further enhanced by making the dashboard more interactive and adding more charts.

**8.2 References:**

* [1] Tian Guo, Nino Antulov-Fantulin, "Predicting short-term Bitcoin price fluctuations from buy and sell orders" Machine Learning , arXiv:1802.04065, Feb 2018
* [2]Connor Lamon, Eric Nielsen, Eric Redondo, "Cryptocurrency Price Prediction Using News and Social Media Sentiment" semantic scholar, 2017
* [3] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, Chang Hun Kim, "Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies" Interdisciplinary Program in Visual Information Processing, Korea University, Seoul, Korea Aug 2016
* [4] HUISU JANG AND JAEWOOK LEE “An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information," IEEE access, December 4, 2017,
* [5] Anshul Saxena, T.R. Sukumar, "Predicting bitcoin price using lstm And Compare its predictability with arima model", International Journal of Pure and Applied Mathematics Volume 119 No. 17 2018
* [6] Jethin Abraham, Daniel Higdon, John Nelson, Juan Ibarra, "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis", SMU Data Science Review: Vol. 1 : No. 3 , Article 1.
* Matta, Martina & Lunesu, Maria Ilaria & Marchesi, Michele, "Bitcoin Spread Prediction Using Social And Web Search Media, 2015.
* [8] A, Vishal & Sonawane, Sheetal, "Sentiment Analysis of Twitter Data: A Survey of Techniques", International Journal of Computer Applications, 2016, 139. 5-15. 10.5120/ijca2016908625.
* [9] Nofer, M, “The value of social media for predicting stock returns: Preconditions, instruments and performance analysis”, 2015,10.1007/978-3-658-09508-6.
* [10] Siddhi Velankar, Sakshi Valecha, Shreya Maji, "Bitcoin Price Prediction using Machine Learning", International Conference on Advanced Communications Technology (ICACT), 2018