CONTRIVER®, Mysore

Department of Computer Science and Engineering

INTERNSHIP TRAINING REPORT

BITCOIN PRICE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfilment of the Requirements for the Certification of

30 days Internship Training Program

SUBMITTED BY

SHAMSHER QUTBODDIN DESAI
(4CA18CS030)

Under the Guidance of

Ms. RAKSHITHA J
Design Engineer,
Contriver

Ms. SYEDA ARBEENA KAUSAR
Developer,
Contriver

Department of Programming and Development

M/S CONTRIVER®
#127/1, Chamalapura Street, Nanjangud,
Mysore 571301,
Karnataka, India
2021 - 2022

CONTRIVER®

#127/1, Chamalapura Street, Nanjangud, Mysore 571301.

Department of Programming and Development

TRAINING CERTIFICATE

*This is to certify that **MR.SHAMSHER QUTBODDIN DESAI(4CA18CS030).** Bonafide students of **College of Engineering** in partial fulfillment for the award of "**Training Certificate**" in **Department of Programming and Development*** of the ***CONTRIVER, Mysore** during the year 2021-2022. It is certified that she as undergone internship during the time period from 02/09/2021 to 10/10/2021 of all working days corrections/suggestions indicated for internal validation have been incorporated in the report deposited to the guide and trainer. The training report has been approved as it satisfies the organizational requirements in respect of Internship training prescribed for the said qualification.*

Ms.Syeda Arbeena Kausar
Developer,
Contriver

Ms.RAKSHITHA
Design Engineer,
Contriver

Mr. SANJAY B
DMT, B.E.
Sr. Production Head and Chief Executive Officer

ACKNOWLEDGEMENT

It is our privilege to express gratitude to all those inspired us and guided to complete the internship-training program. This work has remained incomplete without the direct and indirect help of many people who have guided us in the success of this internship. We are grateful to them.

Date:

Place: Mysore

-SHAMSHER QUTBODDIN DESAI

SHAMSHER QUTBODDIN DESAI

INDIAN +91-7892693230 sqdesai07@gmail.com

www.linkedin.com/in/shamsher-desai-07

OBJECTIVE

To work in a very challenging and competitive job environment with an appraisal and growth combination, where I would be able to significantly contribute to the organization's requirements while continuously enhancing my skill-set.

ACADEMIC INFORMATION

EDUCATION QUALIFICATIONS:

COURSE/EXAM	YEAR OF PASSING	MARKS OBTAINED IN %
B.E IN COMPUTER SCIENCE ENGINEERING	*(2022)*	*(7.85)*
DIPLOMA IN AGRICULTURE SCIENCE	*2018*	*7.75*
PUC	*2017*	*45%*
S.S.L.C	*2014*	*60.83%*

INTERNSHIP

IC SOLUTIONS	*WEB DEVELOPMENT*	01-10-2020 to 31-10-2020
DELITHE	*MACHINE LEARNING*	10-05-2021 to 10-06-2021

TECHNICAL SKILLS

➢ Operating System	:	WINDOWS
➢ Packages	:	MS Office, MS powerpoint, MS excel
➢ Engineering Tools	:	WEB DEVELOPER USING HTML,CSS AND JAVASCRIPT BACKEND USING PHP AND SQL PROGRAMMING LANGUAGES LIKE PYTHON, CORE JAVA,C,C++

PROJECT DETAILS

ENGINEERING MINI PROJECT:

1. TOP 10 SINGER'S IN INDIA
2. DATA BASE MANAGEMENT SYSTEM
3. TRAVEL BOOKING SITE , Etc

TECHNOLOGY USED: HTML,CSS,JAVA SCRIPT AND PHP

4. COVID-19 VACCINATION IN INDIA
5. CHAT-BOT AND TALK-BOT

TECHNOLOGY USED: MACHINE LEARNING USING PYTHON

PERSONAL STRENGTH

- Hardworking, dedicated,quick decision maker, responsible, self confident,Team Leader

PERSONAL PROFILE

Name	: SHAMSHER QUTBODDIN DESAI
Father's name	: QUTBODDIN S DESAI
Marital Status	: Single
Nationality	: Indian
Languages Known	: English, Kannada,Urdu,Hindi
Personal address	: S/O QUTBODDIN S DESAI Mehaboob Nagar Chikodi Belgaum

DECLARATION

I hereby declare that all the information's are correct and true to the best of my knowledge and belief.

DATE: Yours Sincerely

Place: (SHAMSHER QUTBODDIN DESAI)

TOPICS FROM TRAINING

HTML(Hypertext Markup Language)

HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see was written using one version of HTML.

HTML code ensures the proper formatting of text and images for your Internet browser. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the structure of a web page, and CSS as its appearance.

CSS(Cascading Style Sheet)

CSS stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML. CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

JAVA SCRIPT

JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for **Client-side** developments as well as **Server-side** developments. JavaScript contains a standard library of objects, like **Array, Date**, and **Math,** and a core set of language elements like **operators, control structures**, and **statements.**

JavaScript can be added to your HTML file in two ways:

- **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
- **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Applications of java script

- Web Development
- Web Applications
- Server Applications
- Games
- Smart watches
- Art
- Machine Learning.

PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including structured , object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Python is a must for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Python is Interpreted : Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to www browsers to games.

MACHINE LEARNING

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. Using the classic algorithms of machine learning, text is considered as a sequence of keywords instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

Some Machine Learning Methods

- Supervised machine learning algorithms
- Unsupervised machine learning algorithms
- Semi-supervised machine learning algorithms

TOPICS FROM GUEST LECTURER (WORD PRESS)

WordPress is a free, open-source website creation platform. On a more technical level, WordPress is a content management system (CMS) written in PHP that uses a MySQL database. WordPress is the easiest and most powerful blogging and website builder in existence today. WordPress is an excellent website platform for a variety of websites. From blogging to e-commerce to business and portfolio websites, WordPress is a versatile CMS. Designed with usability and flexibility in mind, WordPress is a great solution for both large and small websites.

A WordPress website is any website that uses WordPress as its content management system (CMS). WordPress powers both the *backend* of the website the interface where a user logs in to make changes or add new content and the *frontend* the visible part of the website that your visitors see on the web.

A WordPress plugin is a package of additional code that you can upload to your website to add new features, functionality or integrations.

- Plugins can be added via the Plugins menu in your WordPress dashboard.
- Using the built-in search function, you can find a free plugin available on the WordPress.org Plugin directory.
- Plugins can also be packaged as zip files that you upload through the Plugins menu uploader in your WordPress dashboard.

A WordPress theme provides the design “framework” of your website. Most WordPress themes provide

- The overall design (the look, feel and style) of your site
- Site-wide font styles
- Color scheme
- Styles for blog posts and blog archives
- Page layouts (or templates)
- Widget locations
- Additional stylistic details

FEEDBACK/OPINION OF THE INTERNSHIP

Innovative topics/Methods: Providing internship with real work is number one to ensuring your program's success. **Provide encouragement for thinking differently** having a team member show creativity-even if the end result was not positive is cause for praise. If they know you support their efforts, they will be motivated to try again in the future. In the spirit of failing forward, take the opportunity to discuss with them how to use any new insights learned to possibly improve other processes, systems, Sharing current organizational or team challenges provides an employee greater clarity on where to focus their innovative thoughts. Go so far as to quantify how many new ideas you'd like to see in a period of time.

Syllabus/Concepts that can be included/recommended in engineering curriculum: WordPress is the most popular content management system that powers 63% of all websites with a CMS, which equals 36% of all websites on internet. As the leading CMS, WordPress enables a whole ecosystem of business opportunities.

WordPress is web publishing software we can use to create a beautiful website or blog. It just may be the easiest and most flexible blogging and website content management system (CMS) for beginners.

Area of improvements/Drawbacks in the internship program: Stay connected with interns after they leave. If you felt that they were a good fit for the company, let them know about upcoming opportunities and new job openings. Keep relationships open and make sure they know there is always an open door. This could potentially be a future employee.

Opinion of the internship: The internship was extremely valuable for all us. It was a great experience the supervisors are so good their way of teaching and interacting. We learned that practical expirence in skills such as multitasking, communicating, learning to deal with real world, also dealing with deadliness are really good expirence and will help us in many ways.

The department was helpful and beneficial to encourage students to find their internship based on their future career goals. As a whole this internship was succesfulin furthering my knowledge of career in the field og computer science.

CONTENTS

CHAPTER 1:

INTRODUCTION

Overview

Bitcoin has recently received a lot of attention from the media and the public due to its recent price surge and crash. The Bitcoin daily prices from 29 November 2011 to 31 December 2018 on Bitstamp (https://www.bitstamp.net/), which is the longest-running cryptocurrency exchange. On Bitstamp, the Bitcoin price reached the highest price (19, 187.78 USD) on 16 December 2017 and has fallen up to 3179.54 USD (16.57% of the highest price) on 15 December 2018. Then, it has again increased with some fluctuations since April 2019. Although the Bitcoin price seems to follow a random walk, some recurring patterns seem to exist in the price fluctuations when considering the log value of the
Bitcoin price
Bitcoin, invented in 2009 by **Satoshi Nakamoto** to solve the inherent weakness of the trust-based model of transactions and initially defined as a purely peer-to-peer electronic cash system, has become an asset or commoditylike product traded in more than 16,000 markets around the world. Although proponents hold that one of Bitcoin's important application is to take the place of fiat currency, the true nature of Bitcoin remains a vexing problem. Investors do not treat Bitcoin as a currency according to the criteria used by economists; instead, they regard Bitcoin as a speculative investment similar to the Internet stocks of the last century. Before Bitcoin disrupted existing payment and monetary systems, its several-year trading and increasing popularity attracted attention from across society, including from policymakers, and the peak of Bitcoin's market capitalization in 2017 reached 300 billion US dollars, almost equal to that of Amazon in 2016.
Machine learning techniques to engineer sample dimensions for Bitcoin price prediction. Inspired by the principle of Occam's razor and the characteristics of our datasets, we tackle the problem as follows. First, the prediction sample is divided into daily intervals with small sample size and 5-minute intervals with a big sample size. Second, we conduct the features engineering: select high-dimension features for daily price and few features for 5-minute interval trading data respectively. Third, we conduct simple statistical models including Logistic Regression and Linear Discriminant Analysis and the more complicated machine learning models including Random Forest, XGBoost, Quadratic Discriminant Analysis, Support Vector Machine and Long Short-term Memory. Fourth, we adopt the simple statistical methods to predicting Bitcoin daily price with high-dimensional features to avoid overfitting.

Prediction:

The Bitcoin's value varies just like a stock albeit differently. There are a number of algorithms used on stock market data for price prediction. However, the parameters affecting Bitcoin are different. Therefore it is necessary to predict the value of Bitcoin so that correct investment decisions can be made. The price of Bitcoin does not depend on the business events or intervening government unlike stock market. Thus, to predict the value we feel it is necessary to leverage machine learning technology to predict the price of Bitcoin.

Existing System :

After the boom and bust of crypto currencies' prices in recent years, Bitcoin has been increasingly regarded as an investment asset. Because of its highly volatile nature, there is a need for good predictions on which to base investment decisions. Although existing studies have leveraged machine learning for more accurate Bitcoin price prediction, few have focused on the feasibility of applying different modeling techniques to samples with different data structures and dimensional features. To predict Bitcoin price at different frequencies using machine learning techniques, we first classify Bitcoin price by daily price and high- frequency price. A set of high-dimension features including property and network, trading and market, attention and gold spot price are used for Bitcoin daily price prediction, while the basic trading features acquired from a cryptocurrency exchange are used for 5-minute interval price prediction.

Proposed System :

Statistical methods including Logistic Regression and Linear Discriminant Analysis for Bitcoin daily price prediction with high-dimensional features achieve an accuracy of 66%, outperforming more complicated machine learning algorithms. Compared with benchmark results for daily price prediction, we achieve a better performance, with the highest accuracies of the statistical methods and machine learning algorithms of 66% and 65.3%, respectively. Machine learning models such as Long Short-term Memory for Bitcoin 5- minute interval price prediction are superior to statistical methods, with accuracy reaching 67.2%. Our investigation of Bitcoin price prediction can be considered a pilot study of the importance of the sample dimension in machine learning techniques.

Objective of the Project :

The objectives of the “Bitcoin Price Prediction” can be stated as follows: 1. Develop an application which will predict the bitcoin prices in future with decent accuracy. 2. Allow the investors to invest wisely in bitcoin trading as the prices of bitcoin have gone up to an exaggerating amount in the last ten years. 3. Make use of machine learning algorithms to increase the accuracy of Bitcoin price prediction

Chapter 2:

LITERATURE SURVEY

We have all considered where bitcoin costs will be one year, two years, five years or even 10 years from now. It's really difficult to anticipate however each and every one of us loves to do it. Tremendous measures of benefits can be made by purchasing and selling bitcoins, whenever done accurately.. It has been proven to be a fortune for many people in the past and is still making them a lot of money today. But this doesn’t come without its downside too. If not thought of and calculated properly, you can lose a lot of money too. You should have an incredible comprehension of how and precisely why bitcoin costs change (organic market, guidelines, news, and so forth), which implies you should realize how individuals make their bitcoin predictions. Considering these things (supply and demand, regulations, news, etc.), one must also think about the technology of bitcoin and its progress. This aside, we now have to deal with the technical parts using various algorithms and technologies which can predict precise bitcoin prices. Although we came across various models which are currently present like Biological neural networks. (BNN), Recurrent neural network (RNN), Long short-term memory (LSTM), Auto regressive integrated moving average (ARIMA), etc. with machine learning and deep neural network concepts. Normally a time series is a sequence of numbers along time. This is due to the fact that this being a time series data set, the overall data sets should be split into two parts: inputs and outputs.Moreover, LSTM is great in comparison with the classic statistics linear models, since it can very easily handle multiple input forecasting problems.

In the approach which we are following, the LSTM will use the previous data to predict bitcoin prices 30 days ahead of it’s closing price. In the approach used by us, we implement Bayesian optimized Recurrent Neural Network (RNN) and a Long Short Term Memory (LSTM) network. The highest classification accuracy is achieved by LSTM with the accuracy of 52% and a RMSE of 8%. Presently we execute the famous Auto backward incorporated moving normal (ARIMA) model for time arrangement gauging as a correlation with the profound learning models. The ARIMA forecast is out performed by the nonlinear deep learning methods which performed much better. Finally both the profound learning models are benchmarked on both a GPU and CPU. The training time on the CPU is outflanked by the GPU execution by 67.7%. In the base papers selected by us, the author collected a data set of over 25 features relating to the bitcoin price and payment network over a period of five years, recorded on a daily basis were able to predict the sign of the daily bitcoin price change with an incredible accuracy of 98.7%.

In the second period of our examination we are just focusing in on the bitcoin price information alone and utilized information at 10 minutes and 10 seconds time frame.. This is due to the fact that we saw an incredible opportunity to precisely evaluate price predictions at various levels of granularity and noisiness are modelling. This resulted in incredible results which had 50 to 55% accuracy in precisely predicting the future bitcoin price changes using 10 minute time intervals.

Chapter 3

SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives a brief description of the services that the system should provide and also the constraints under which the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources. Bitcoin Price Prediction using Machine Learning

3.1 Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements: - Following are the functional requirements on the system:

1. The entire control model set must be translated to C output Code.
2. Inputs must be models designed using CLAW design components along with standard design components,
3. Multiple design models must be processed and the result must be combined to obtain a single output file.

3.2 Non-Functional Requirement

Nonfunctional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Nonfunctional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as: -

- ➢ Product Requirements
- ➢ Organizational Requirements
- ➢ User Requirements
- ➢ Basic Operational Requirements

3.2.1 Product Requirements

Platform Independency: Standalone executables for embedded systems can be created so the algorithm developed using available products could be downloaded on the actual hardware and executed without any dependency to the development and modeling platform. Bitcoin Price Prediction using Machine Learning.

Correctness:

It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data. Ease of Use: Model Coder provides an interface which allows the user to interact in an easy manner. Modularity: The complete product is broken up into many modules and well- defined interfaces are developed to explore the benefit of flexibility of the product. Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevance and correctness Nonfunctional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

3.2.2 Organizational Requirements

Process Standards: The standards defined by DRDO are used to develop the application which is the standard used by the developers inside the defense organization. Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs. Bitcoin Price Prediction using Machine Learning

3.2.3 User Requirements

The coder must request the name of the model file to be processed o In case of multiple files, the coder must ask the names of the files sequentially. o The output file must be a C code translated from the model. o Only a single output file must be created even if multiple input files are provided.

3.2.4 Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points: - Mission profile or scenario: It describes the procedures used to accomplish mission objectives. It also finds out the effectiveness or efficiency of the system. Performance and related parameters: It points out the critical system parameters to accomplish the mission Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation. Operational life cycle: It defines the system lifetime. Bitcoin Price Prediction using Machine Learning

3.2.5 System Configuration H/W System Configuration:

Processor :	Pentium –IV , Intel I5
Speed :	1.1 Ghz (min)
RAM :	8GB RAM
Hard Disk :	100 GB
Key Board :	Standard Windows Keyboard
Mouse :	Two or Three Button Mouse
Coding Language :	8.1/10

S/W Configuration:

Python:	3.7.0 / 3.86
Software :-	Anaconda Jupyter, Google Colab and Pichamp

Chapter 4:

ALGORITHM

Step 1: Start
Step 2: Install All Required Packages(numpy,pandas,matplotlib,tensorflow,scikit-learn, fbprophet)
Step 3: Import Installed Packages
Step 4: Get e Financial Data OR Get the Data from YahooFinancial and Download the Data in CSV formate Step 5: Prepare the data for prediction
Step 6: Train Your Current data with x_train and y_train
Step7: With use of numpy create array for both x_train and y_train
Step 8: Create a model for sequential data
Step 9: Add the model to the LSTM, Dropout and Dense with shape of x_train and y_train
Step 10: Compile the model with optimizer
Step 11: fit the model for x_train and y_train
Step 12: Test the model with start date and end date and read the data with web.Datareader
Step 13: Concatenate the model datta and trained data with pandas
Step 14: Create Model Inputs with the concatenated data
Step 15: Predict the Price with the model and scaler.inverse_transform
Step 16: Plot the Graph of Actual rate and the Prediction Prices with Time and Price
Step 17: Prediction of Next Day Data and Show the Output
Step 18: Stop

Chapter 5:

BLOCK DIAGRAM

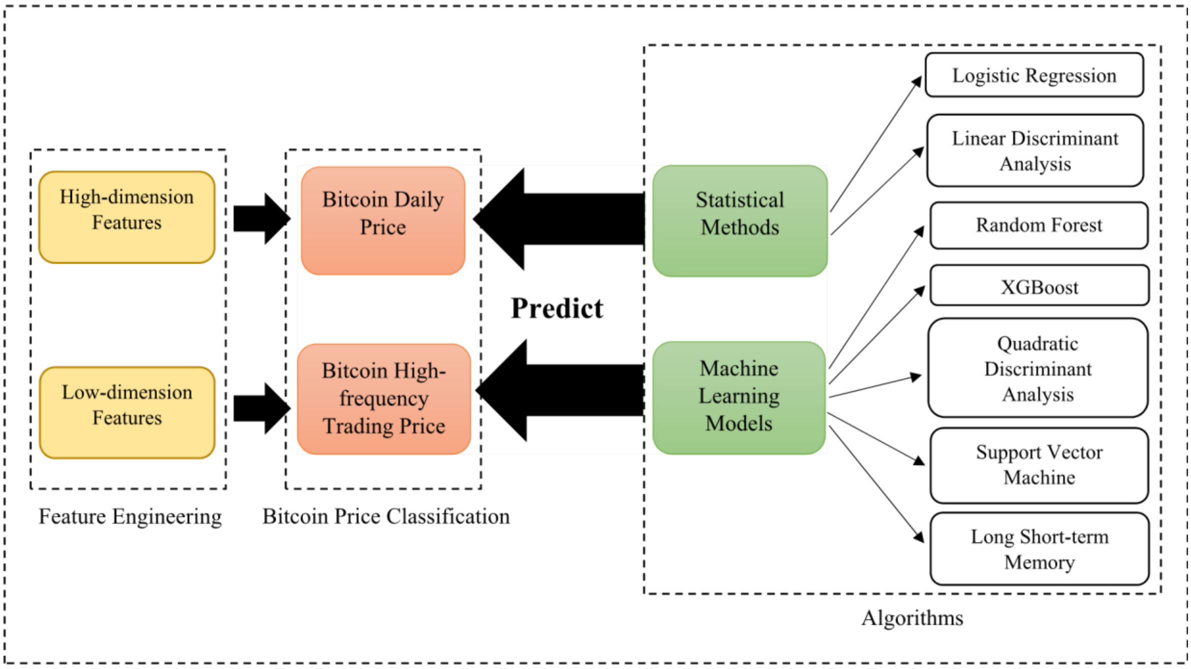
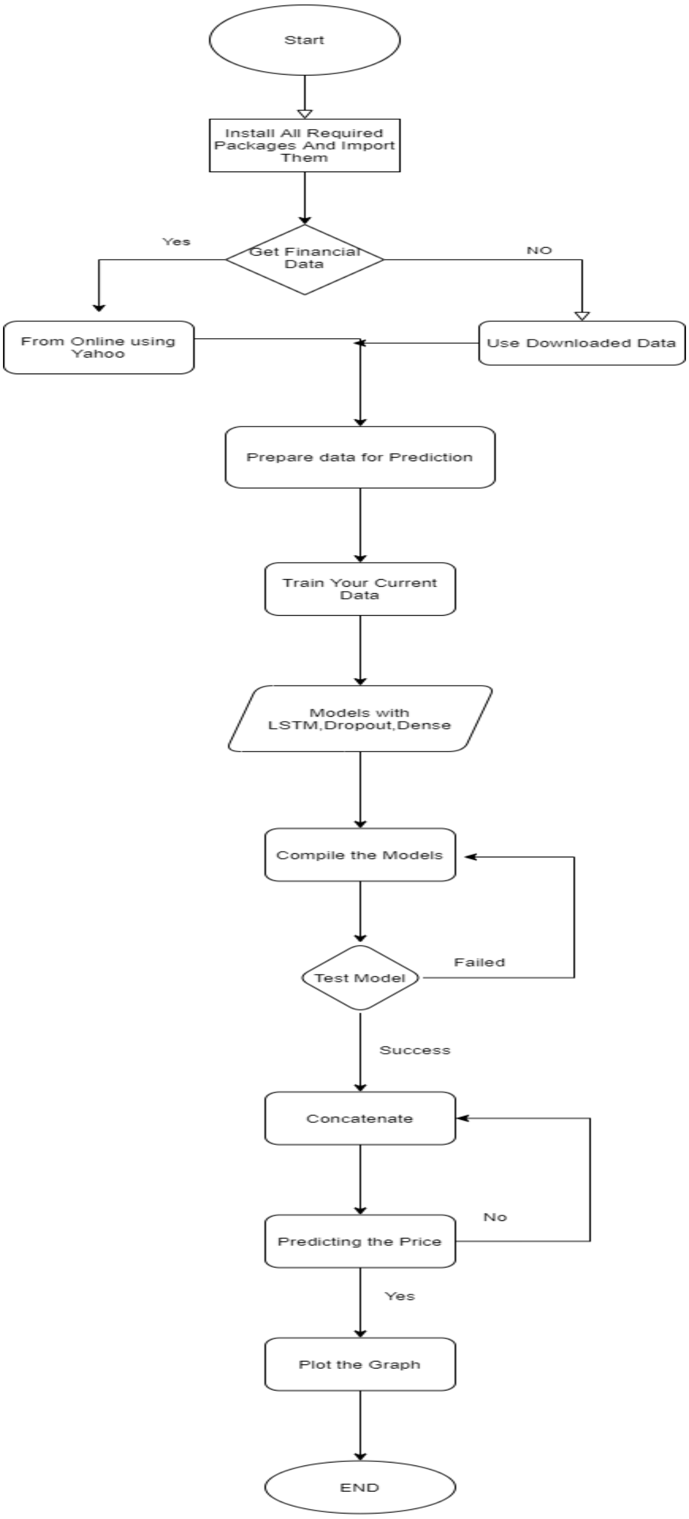


Fig. 1. Overview of the research framework.

Our project addresses leveraging appropriate machine learning techniques to engineer sample dimensions for Bitcoin price prediction. Inspired by the principle of Occam’s razor and the characteristics of our datasets, we tackle the problem as follows. First, the prediction sample is divided into daily intervals with small sample size and 5-minute intervals with a big sample size. Second, we conduct the features engineering: select high-dimension features for daily price and few features for 5-minute interval trading data respectively. Third, we conduct simple statistical models including Logistic Regression and Linear Discriminant Analysis and the more complicated machine learning models including Random Forest, Quadratic Discriminant Analysis, Support Vector Machine and Long Short-term Memory. Fourth, we adopt the simple statistical methods to predicting Bitcoin daily price with high- dimensional features to avoid overfitting. Meanwhile, the machine learning models are leveraged in high-frequency price few features. Fig shows the overview of our research framework. Our project makes observations in two ways. One is to extend the feature dimensions, and the other is to evaluate different machine learning techniques for solving problems of multiple frequency Bitcoin Price Prediction using Machine Learning Bitcoin prices. The study makes the following contributions. To the best of our knowledge, we are at the forefront of establishing higher dimensional features for problems of Bitcoin daily price prediction by integrating investor attention, media hype and gold spot features with common and traditional features such as network and market. We address the importance of the sample dimension by classifying Bitcoin price data by interval. The real-time 5-minute interval trading data acquired from the top cryptocurrency exchange is high-frequency and large scale, and the aggregated Bitcoin daily price obtained from CoinMarketCap is low-frequency and small scale. Hence, the problem of Bitcoin price prediction is addressed from a broad perspective.To find appropriately complex models and meet the requirement of accuracy, we evaluate different machine learning techniques using problems of multiple frequency Bitcoin price. Specifically, we lower the complexity of algorithms for low-frequency daily price prediction with higher-dimension features and apply more complicated models for high-frequency price prediction with a few features. The results show that simple statistical methods outperform machine learning models for daily Bitcoin price prediction while more complicated models should be adopted for high frequency Bitcoin price prediction. We envision this study as a pilot for dealing with datasets with different scales and intervals, which can shed light on other industrial prediction problems in the context of machine learning.

Chapter 6:

FLOWCHART



CHAPTER 7:

IMPLEMENTATION

Packages Required for Machine Learning and Prediction

pip install numpy
pip install pandas
pip install matplotlib
pip install seaborn

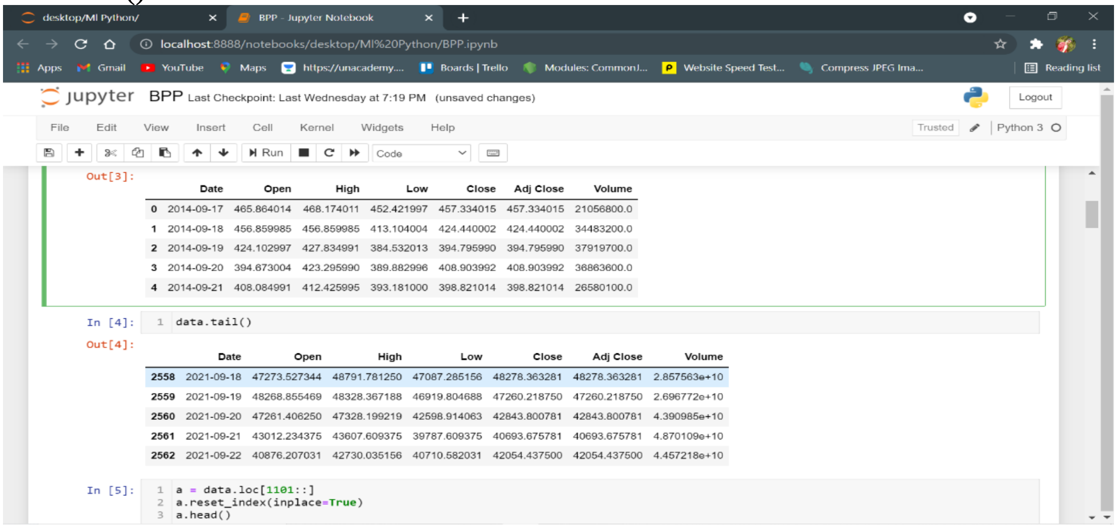
#Data Manupulation Libraries

import numpy as np
import pandas as pd

#Data Visualization Libararies

from matplotlib import pyplot as plt
import seaborn as sns

#Reading the CSV file
data = pd.read_csv("BTC-USD.csv")
#Sorting the Given file's Data
data = data.sort_values('Date')
#Lets See the peal of the data set
data.head()



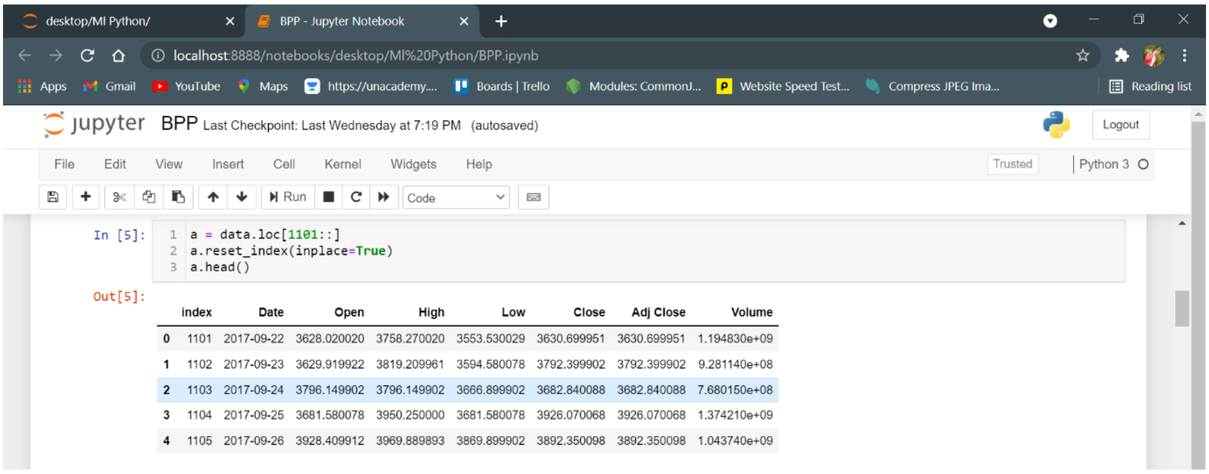
```
Out[3]:
        Date        Open        High         Low       Close   Adj Close       Volume
0 2014-09-17  465.864014  468.174011  452.421997  457.334015  457.334015   21056800.0
1 2014-09-18  456.859985  456.859985  413.104004  424.440002  424.440002   34483200.0
2 2014-09-19  424.102997  427.834991  384.532013  394.795990  394.795990   37919700.0
3 2014-09-20  394.673004  423.295990  389.882996  408.903992  408.903992   36863600.0
4 2014-09-21  408.084991  412.425995  393.181000  398.821014  398.821014   26580100.0
```

```
In [4]:  data.tail()
```

```
Out[4]:
           Date          Open          High           Low         Close     Adj Close        Volume
2558 2021-09-18  47273.527344  48791.781250  47087.285156  48278.363281  48278.363281  2.857563e+10
2559 2021-09-19  48268.855469  48328.367188  46919.804688  47260.218750  47260.218750  2.696772e+10
2560 2021-09-20  47261.406250  47328.199219  42598.914063  42843.800781  42843.800781  4.390985e+10
2561 2021-09-21  43012.234375  43607.609375  39787.609375  40693.675781  40693.675781  4.870109e+10
2562 2021-09-22  40876.207031  42730.035156  40710.582031  42054.437500  42054.437500  4.457218e+10
```

```
In [5]:  a = data.loc[1101::]
         a.reset_index(inplace=True)
         a.head()
```

Fig:7.1



BPP Last Checkpoint: Last Wednesday at 7:19 PM (autosaved)

In [5]:

```
a = data.loc[1101::]
a.reset_index(inplace=True)
a.head()
```

Out[5]:

	index	Date	Open	High	Low	Close	Adj Close	Volume
0	1101	2017-09-22	3628.020020	3758.270020	3553.530029	3630.699951	3630.699951	1.194830e+09
1	1102	2017-09-23	3629.919922	3819.209961	3594.580078	3792.399902	3792.399902	9.281140e+08
2	1103	2017-09-24	3796.149902	3796.149902	3666.899902	3682.840088	3682.840088	7.680150e+08
3	1104	2017-09-25	3681.580078	3950.250000	3681.580078	3926.070068	3926.070068	1.374210e+09
4	1105	2017-09-26	3928.409912	3969.889893	3869.899902	3892.350098	3892.350098	1.043740e+09

Fig:7.2

```
#Creating a dataframe for closing price (USD) column as it's our target variable to predict.
price = data[['Close']]
#Ploting the price for the historical data of bitcoin
plt.figure(figsize = (15,10))
plt.plot(price)
plt.xticks(range(0,data.shape[0],50),data['Date'].loc[::50],rotation=45)
plt.title("Bitcoin Price",fontsize=18,fontweight='bold')
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close (USD)',fontsize=18)
plt.show()
```

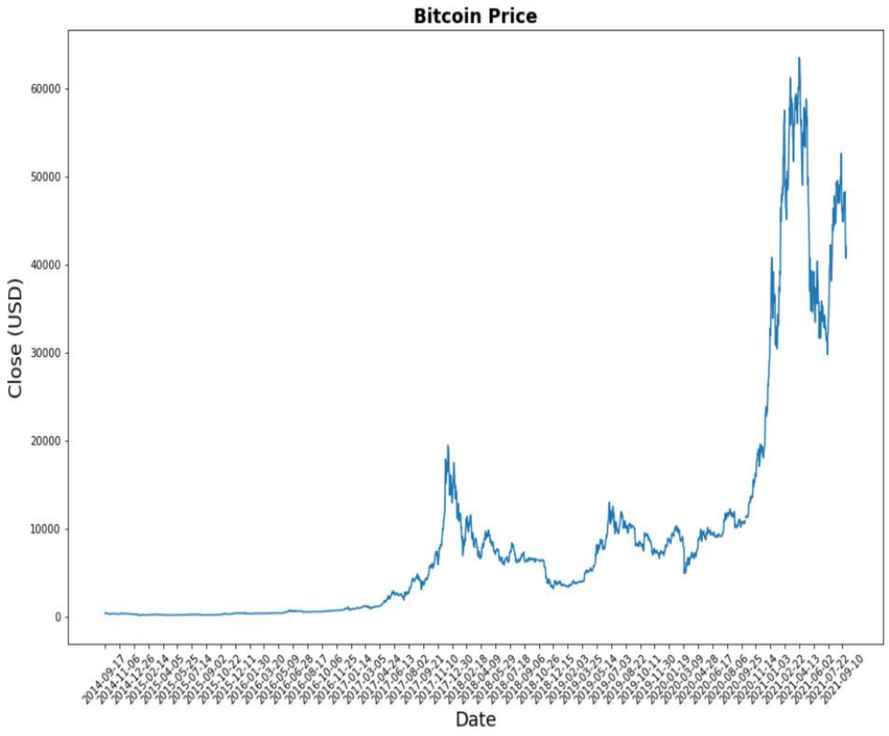
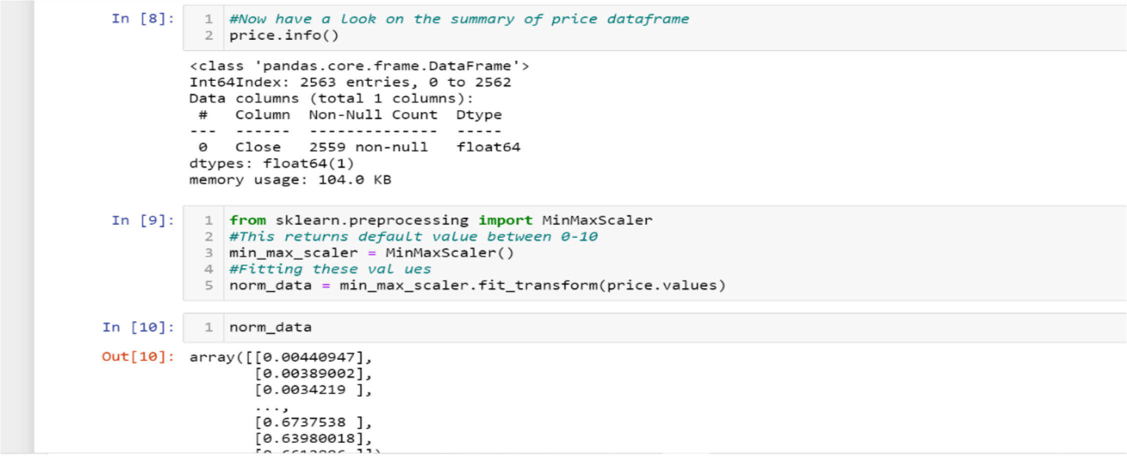
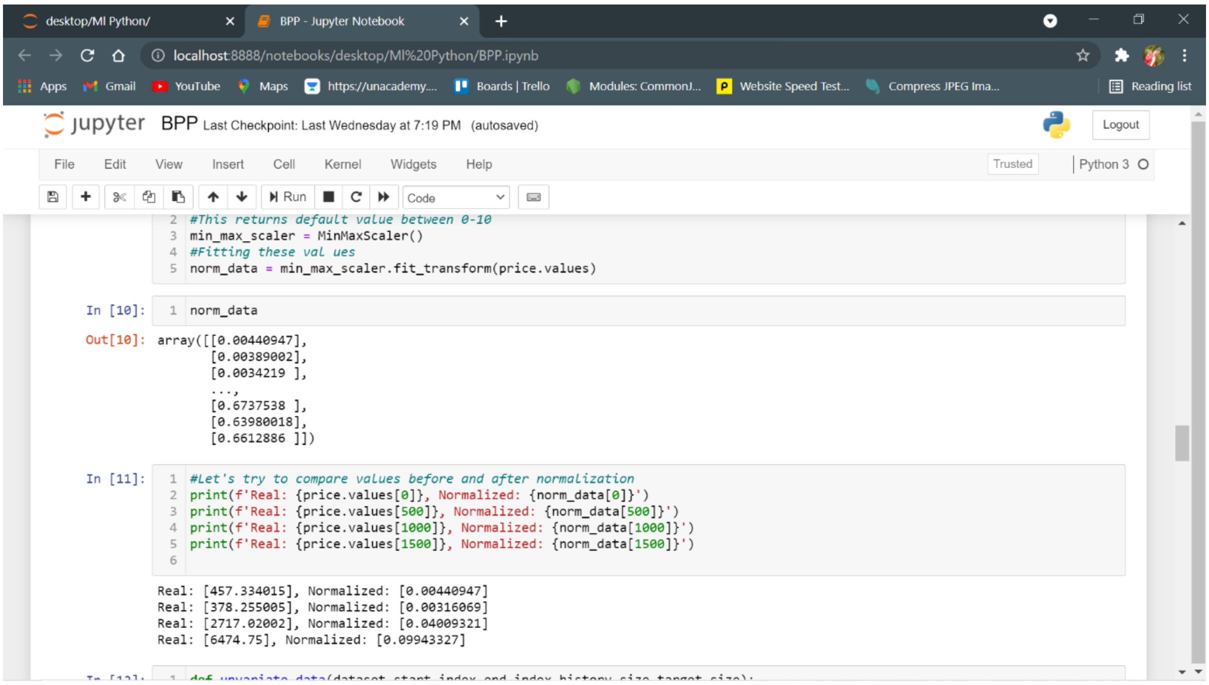


Fig:7.3



```
In [8]:  #Now have a look on the summary of price dataframe
         price.info()

         <class 'pandas.core.frame.DataFrame'>
         Int64Index: 2563 entries, 0 to 2562
         Data columns (total 1 columns):
          #   Column  Non-Null Count  Dtype
         ---  ------  --------------  -----
          0   Close   2559 non-null   float64
         dtypes: float64(1)
         memory usage: 104.0 KB

In [9]:  from sklearn.preprocessing import MinMaxScaler
         #This returns default value between 0-10
         min_max_scaler = MinMaxScaler()
         #Fitting these val ues
         norm_data = min_max_scaler.fit_transform(price.values)

In [10]: norm_data

Out[10]: array([[0.00440947],
                [0.00389002],
                [0.0034219 ],
                ...,
                [0.6737538 ],
                [0.63980018],
                [0.6612886 ]])
```

Fig:7.4



```
         #This returns default value between 0-10
         min_max_scaler = MinMaxScaler()
         #Fitting these val ues
         norm_data = min_max_scaler.fit_transform(price.values)

In [10]: norm_data

Out[10]: array([[0.00440947],
                [0.00389002],
                [0.0034219 ],
                ...,
                [0.6737538 ],
                [0.63980018],
                [0.6612886 ]])

In [11]: #Let's try to compare values before and after normalization
         print(f'Real: {price.values[0]}, Normalized: {norm_data[0]}')
         print(f'Real: {price.values[500]}, Normalized: {norm_data[500]}')
         print(f'Real: {price.values[1000]}, Normalized: {norm_data[1000]}')
         print(f'Real: {price.values[1500]}, Normalized: {norm_data[1500]}')


         Real: [457.334015], Normalized: [0.00440947]
         Real: [378.255005], Normalized: [0.00316069]
         Real: [2717.02002], Normalized: [0.04009321]
         Real: [6474.75], Normalized: [0.09943327]

In [12]: def unvariate_data(dataset,start_index,end_index,history_size,target_size):
```

Fig:7.5

```
def unvariate_data(dataset,start_index,end_index,history_size,target_size):
    data2 = []
    labels = []
    start_index = start_index + history_size
    if end_index is None:
        end_index = len(dataset) - target_size
    for i in range(start_index, end_index):
        indices = range(i-history_size, i)
        #Reshape  data from (history_size,) to (history_size, 1)
        data2.append(np.reshape(dataset[indices], (history_size, 1)))
        labels.append(dataset[i+target_size])
    return np.array(data), np.array(labels)
past_history = 5
future_target = 0
#Here we are using last 5 days data to learn and predict the next point in the time series
#last indexed 80% data using for training
TRAIN = int(len(norm_data)*0.8)
x_train, y_train = unvariate_data(norm_data,0,TRAIN,past_history,future_target)
x_test,y_test = unvariate_data(norm_data,TRAIN,None,past_history,future_target)

import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Dense,LSTM,leakyReLU, Dropout

num_units = 64 #Numbers of neurons
learning_rate = 0.0001
activation_function = 'sigmoid'
adam = Adam(lr=learning_rate)
loss_function = 'mse' #mse = Mean Squar error
batch_size = 5
num_epochs = 250
#initialize the RNN
model = Sequntial()
#keras has simply the stack multiple layers on top of each other, for this we need to initialize
the model as Sequential()
model.add(LSTM(units = num_units, activation = activation_function, input_shape=(None,
1)))
model.add(LeakyReLU(alpha=0.5))
model.add(Dropout(0.1))
#This layer helps to prevent overfitting by ignoring randomly selected neurons during training,
and reduces the sensitivity of specific neurons..
model.add(Dense(units=1))  #fully connected layer
#compiling the RNN
model.compile(optimizer=adam, loss=loss_function)
model.summary()
```

#Train the Model using Training set

```
history = model.fit(x_train, y_train, validation_split=0.1, batch_size=batch_size,
epochs=num_epochs, shuffle=False)
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
plt.figure(figsize=(15,10))
plt.plot(epochs, loss, 'b', label ='Training Loss')
plt.plot(epochs, val_loss, 'r', label = 'Validation Loss')
plt.title("Training Loss VS Validation Loss")
plt.legend()
plt.show()

original = pd.DataFrame(min_max_scaler.inverse_transform(y_test))
predictions = pd.DataFrame(min_max_scaler.inverse_transform(model.predict(x_test)))
sns.set(rc={'figure.figsize':(11.7+2,8.27+2)})
ax = sns.lineplot(x=original.index, y=original[0], label="Test Data", color = 'royalblue')
ax = sns.lineplot(x=predictions.index, y=predictions[0], label="Prediction", color="tomato")
ax.set_title("Bitcoin Price", size = 14, fontweight='bold')
ax.set_xlabel("Days", size=14)
ax.set_ylabel("Cost (USD)", size=14)
ax.set_xticklabels(", size=10)
```

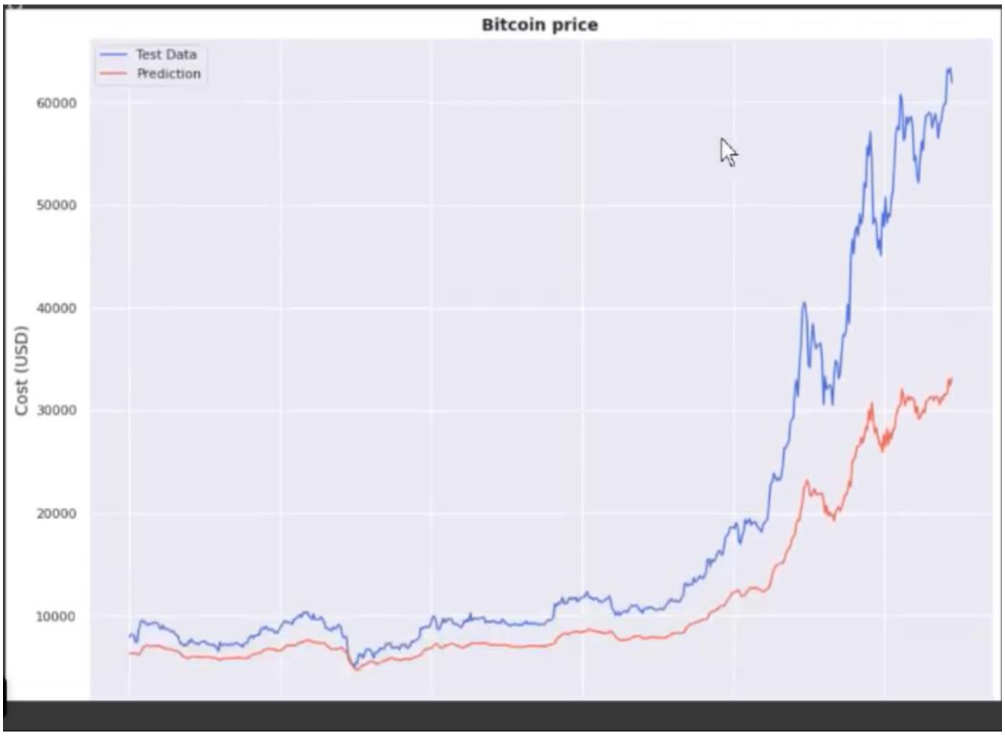


Fig:3.6

FUTURE PREDICTION

```
# IMPORTING LIBRARIES
 import pandas as pd
 from fbprophet import Prophet
 import io

# UPLOADING CSV FILE TO COLAB
from google.colab import files
 uploaded = files.upload()
```

# The Facebook Prophet model only works with data that contains a string time-series format in a column called “ds” and continuous values in a column called “y”. So we need to create the data accordingly:

```
 df = pd.read_csv(io.BytesIO(uploaded['BTC-USD.csv']))
 df = df[["Date", "Close"]]
df.columns = ["ds", "y"]
print(df)
```

# Now let’s fit the data into our model:

```
 prophet = Prophet()
prophet.fit(df)
```

# Now let’s make predictions. The make_future_dataframe method in Prophet model has a parameter named as ‘periods’, we can use it to set the amount of time we need to make predictions. Now let’s make predictions for the next 365 days:

```
 future = prophet.make_future_dataframe(periods=365)
print(future)
forecast = prophet.predict(future)
 forecast[["ds", "yhat", "yhat_lower", "yhat_upper"]].tail(200)
#Now let’s plot our predictions:
from fbprophet.plot import plot
 prophet.plot(forecast, figsize=(20, 10)
```

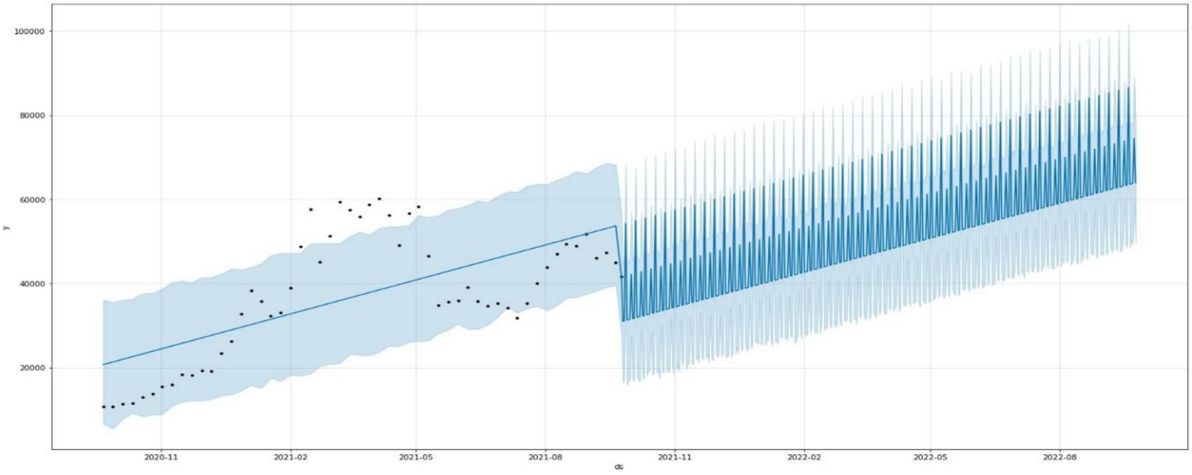


Fig:7.7

Chapter 8:

Results and discussion

Summarizes the performance of all of the machine learning models concerning for the Bitcoin daily price. From the results, we can make the following observations. As expected, the results of the two statistical methods are better overall. The average accuracy of the statistical methods is 65.0%, higher than the average accuracy of the machine learning models (55.3%). The LR model achieved the best results, with an accuracy of 66.0%. Among the machine learning models, XGB performed the worst, with an accuracy of 48.3%, and SVM was the best, with an accuracy of 65.3%, competitive with the statistical methods. In general, LR and LDA outperformed the other machine learning models on the daily price dataset, indicating that properly selected highdimensional feature sets can compensate for the simplicity of models in Bitcoin daily price prediction.

Summarizes the performance of all the machine learning models concerning for the Bitcoin 5minute interval price. As shown, the machine learning models achieved better accuracy than the statistical methods, with LSTM achieving the best result (67.2% accuracy). The average accuracy of the statistical methods was only 53.0%, worse than that of the machine learning models (62.2%). The prediction accuracies of the LR and LDA statistical methods were 54.5% and 51.5%, respectively; both results were inferior to the accuracies of the machine learning models. The machine learning models generally outperformed the two statistical methods due to the large scale of the Bitcoin 5-minute interval dataset. This result is consistent with the paradigm for the main resolution of the practical prediction of complicated models with few features.

Chapter 9:

Conclusion and Future Work

In this study, we investigated machine learning techniques based upon sample characteristics of sample and dimension to predict Bitcoin price. While most previous works simply leverage machine learning algorithms in Bitcoin price prediction, we show that the sample's granularity and feature dimensions should be considered. The Bitcoin aggregated daily price, acquired from CoinMarketCap, facilitates the inclusion of high-dimensional features, including property In this work, we evaluated advanced DL models for predicting cryptocurrency prices and also investigated three research question concerning this problem in a review and discussion approach. Our results revealed that the presented models are inefficient and unreliable cryptocurrency price predictors, probably due to the fact that this problem is a very complicated one, that even advanced deep learning techniques such as LSTM and CNNs are not able to solve efficiently. Also, based on our experimental results and investigation regarding to our research questions about cryptocurrency price problem, we conclude that cryptocurrency prices follow almost a random walk process while few hidden patterns may probably exist in, where an intelligent framework has to identify them in order for a prediction model to make accurate and reliable forecasts. Therefore, new sophisticated algorithmic methods, alternative approaches, new validation metrics should be explored.

Finally, since cryptocurrency datasets follow typical time-series patterns, one may logically conclude that the research questions posed in this work and our concluding remarks and proposals apply to all application domains in which the datasets demonstrate time-series behavior.

References

1. Nakamoto S et al. (2008) Bitcoin: a peer-to-peer electronic cash system, 2008. Working Paper from www.bitcoin.org
2. Urquhart A (2016) The inefficiency of Bitcoin. Elsevier, pp 80–82
3. Jang H, Lee J (2017) An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. 5427—5437
4. Dennys CA, Mallqui RAF (2018) Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. Int J Soft Comput (IJSC) 596–606
5. McNally S, Roche J, Caton S (2018) Cryptocurrency forecasting with deep learning chaotic neural networks. IEEE, pp 339–343
6. Atsalakis GS, Atsalaki IG, Pasiouras, F, Zopounidis C (2019) Bitcoin price forecasting with neuro-fuzzy techniques. Elsevier, pp 770–780
7. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press