Configuration Manual

MSc Research Project
MSc Data Analytics

Koushik Reddy Challa
x22173544

School of Computing
National College of Ireland

Supervisor: DR. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Koushik Reddy Challa

Student ID: X22173544

Programme: Data Analytics **Year:** 2023-2024

Module: Msc Research Project

Lecturer: DR. Catherine Mulwa

Submission Due Date: 31/01/2024

Project Title: Cryptocurrency Forecasting: Unveiling the Future of Bitcoin Prices through Deep Neural Networks

Word Count: 446 **Page Count:** 7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Koushik Reddy Challa

Date: 31th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Koushik Reddy Challa
x22173544

1. Introduction

This manual situates that, how to perform the developments to run the code which was implementation for prediction of bitcoin future prices . The code is written in Python language where we keeped the all pre-requestiques for running of the developed program.

2. System Specification

The development of the forecasting of the bitcoin future prices was developed throughout the following hardware configurations:

- Processor: Intel i7 10 generation
- Operating System: Windows 10
- Ram: 8 GB
- Solid State Drive (SSD): 512GB

3. Softwares Used:

The following tools are used which helps in the development of this project of bitcoin future prices prediction:

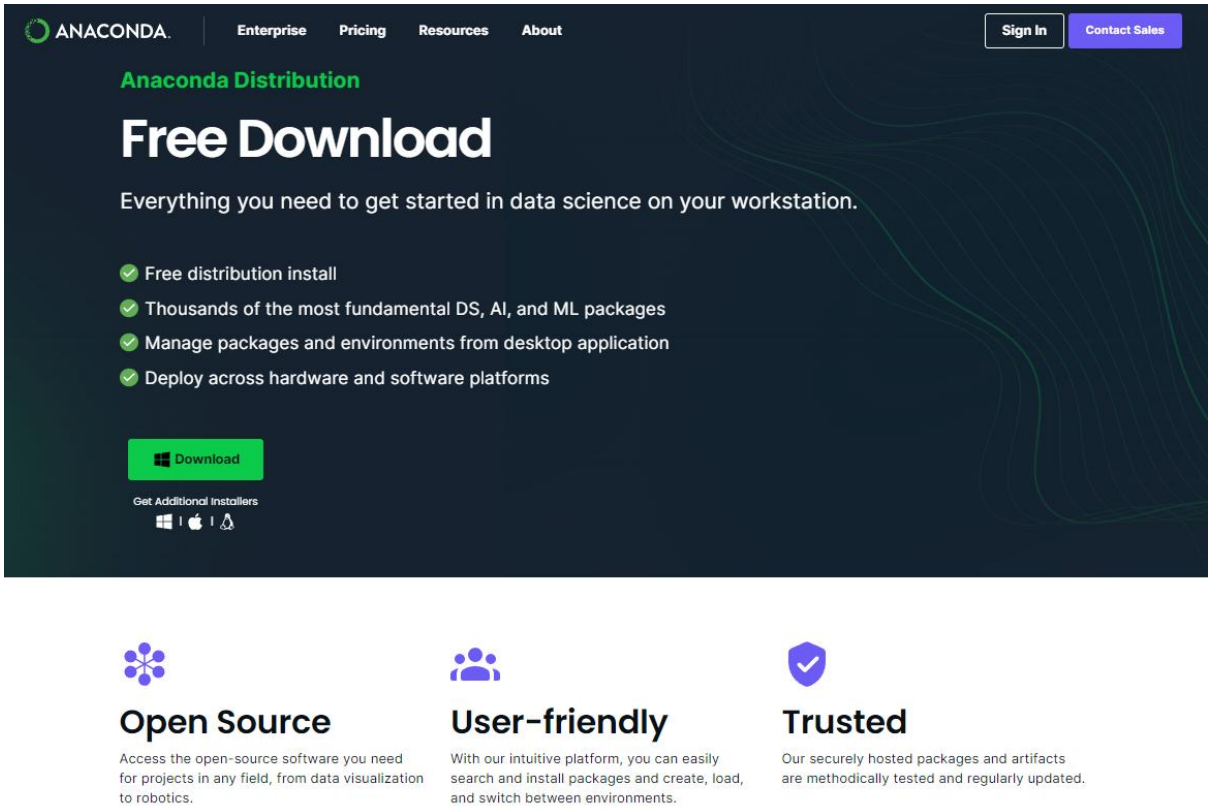
- Python
- Anaconda
- Jupyter

4. Steps to Download and Install the Software:

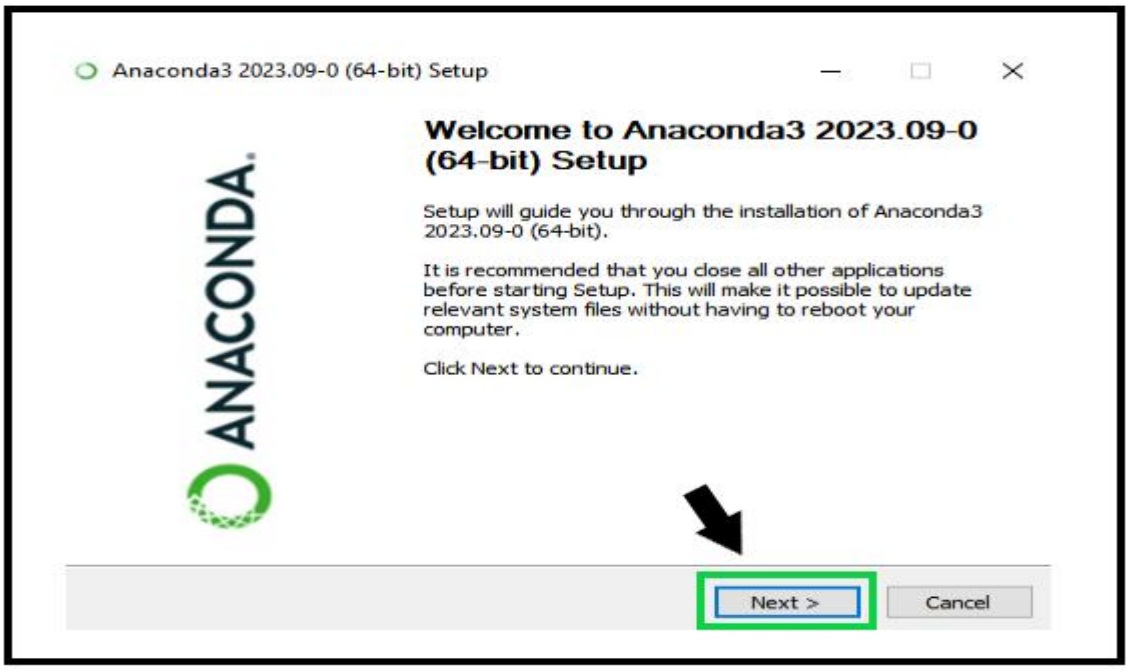
The sections describes how to install the anaconda

- Download and install the Anaconda from their official website: https://www.anaconda.com

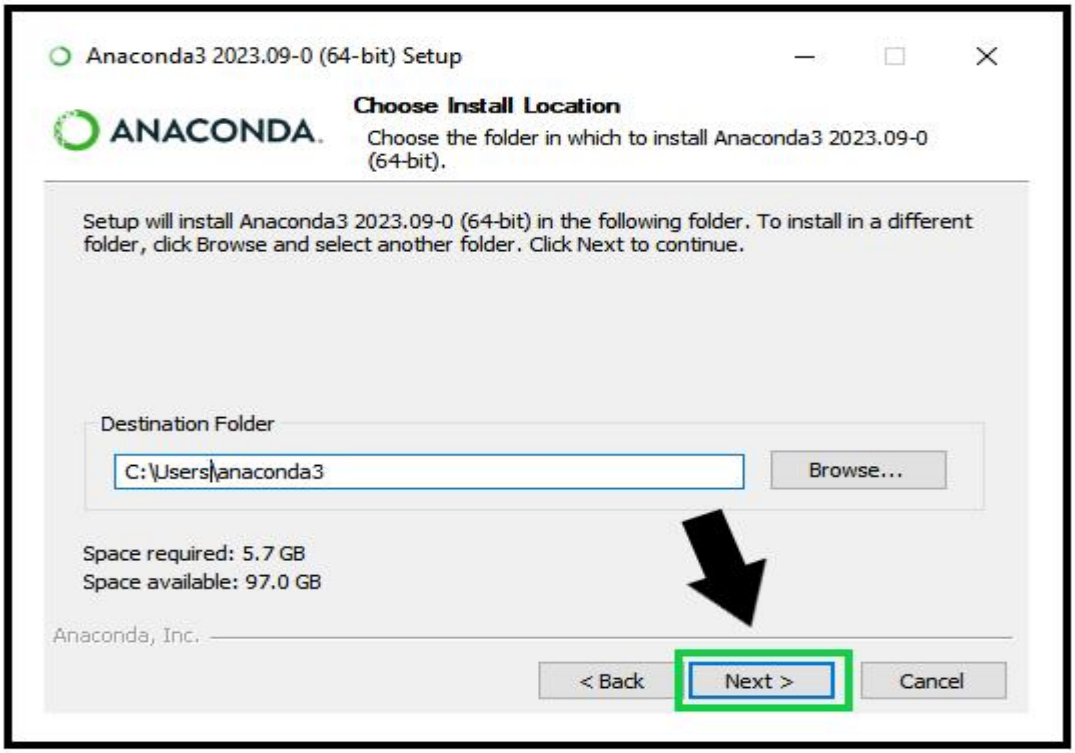
- Click on Download to download the anaconda to your operating system.



- After downlaoding the anaconda from their offical website.
- Follow the instructions to install through steps provided on the website.
- Open downloaded anaconda's setup application to install the software



- Now Click on Next as depicted to the above image illustration to proceed with next step



- Specify the path where you wants to install the application and then click on "Next" till nex to install the application as depicted to the above image illustration

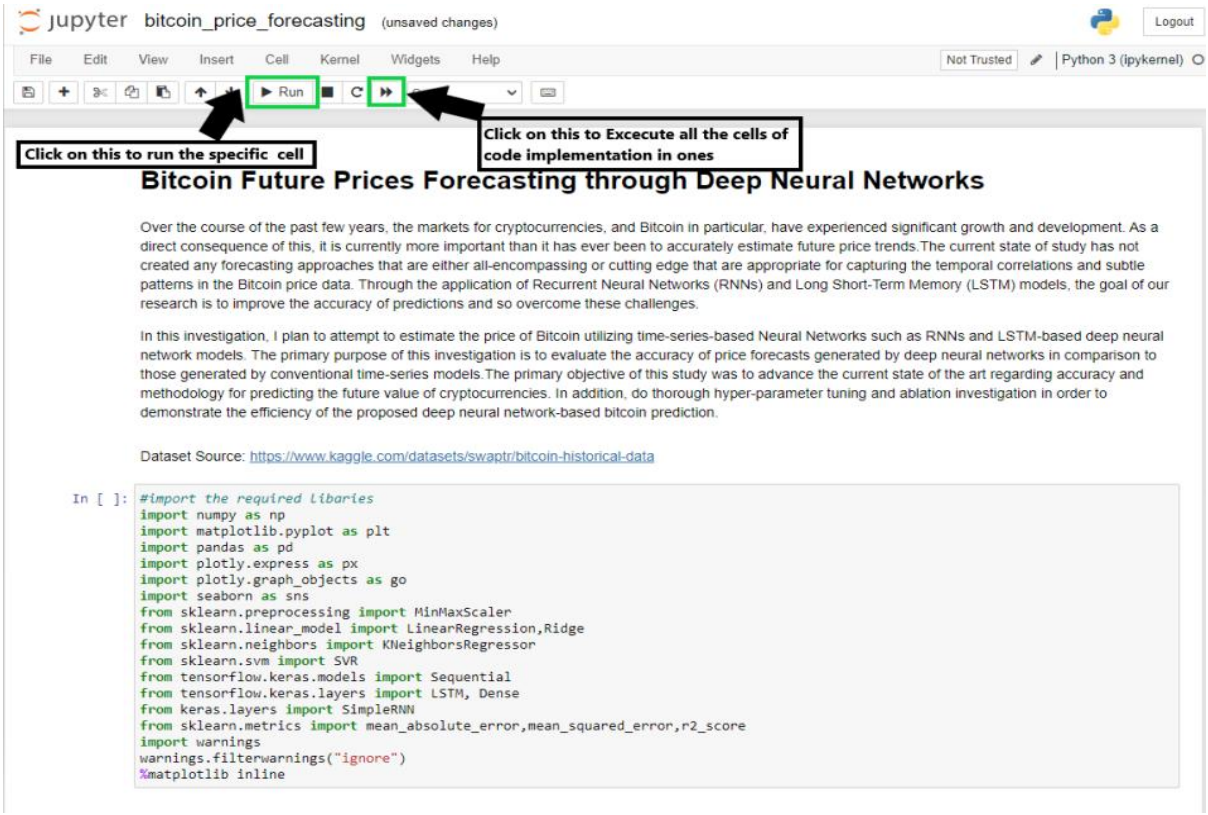
5. Dataset Source

The dataset for this study, I used the dataset from kaggle which is known for collaborates with other users and publishes the dataset. So I chosen the dataset for historical bitcoin dataset which is from 2015-2023

Dataset Source: https://www.kaggle.com/

6. Execution of the Code Implementation

Open the jupyter from the anaconda's navigator and then open the files to run that development of the bitcoin future prices prediction



A. Import the required libraries

```
#import the required libaries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression,Ridge
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from keras.layers import SimpleRNN
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

B. Load the fetched dataset from the kaggle to the jupyter environment/

Load & Explore the Dataset

```
#Load the Bitcoin previous data
dataFrame = pd.read_csv("data.csv")
```

```
#view the last 5 values of the attributes
dataFrame.tail()
```

	Timestamp	Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD
3766758	1444311840	2015-10-08 13:44:00	BTC/USD	242.96	242.96	242.96	242.96	0.033491	8.137003
3766759	1444311780	2015-10-08 13:43:00	BTC/USD	242.95	242.96	242.95	242.96	0.010000	2.429600
3766760	1444311720	2015-10-08 13:42:00	BTC/USD	242.95	242.95	242.95	242.95	0.000000	0.000000
3766761	1444311660	2015-10-08 13:41:00	BTC/USD	242.50	242.95	242.50	242.95	0.001000	0.242950
3766762	1444311600	2015-10-08 13:40:00	BTC/USD	0.00	242.50	0.00	242.50	0.050000	12.125000

```
#basic information about the dataset
dataFrame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3766763 entries, 0 to 3766762
Data columns (total 9 columns):
 #   Column      Dtype
---  ------      -----
 0   Timestamp   int64
 1   Date        object
 2   Symbol      object
 3   Open        float64
 4   High        float64
 5   Low         float64
 6   Close       float64
 7   Volume BTC  float64
 8   Volume USD  float64
dtypes: float64(6), int64(1), object(2)
memory usage: 258.6+ MB
```

The dataset contains the 3766763 rows with 9 different columns

C. Preprocess the dataset

```
In [13]: # Convert the 'Date' column to a datetime format
         dataFrame['Date'] = pd.to_datetime(dataFrame['Date'])

In [14]: # Sort the DataFrame by the 'Timestamp' column in ascending order
         dataFrame = dataFrame.sort_values(by='Timestamp')

In [15]: # Reset the index of the DataFrame
         dataFrame = dataFrame.reset_index(drop=True)
```

Now view the preprocessed dataset

```
In [16]: #view the processed dataframe first five attributes
         dataFrame.head()
```

Out[16]:

	Timestamp	Date	Symbol	Open	High	Low	Close	Volume BTC	Volume USD
0	1444311600	2015-10-08 13:40:00	BTC/USD	0.00	242.50	0.00	242.50	0.050000	12.125000
1	1444311660	2015-10-08 13:41:00	BTC/USD	242.50	242.95	242.50	242.95	0.001000	0.242950
2	1444311720	2015-10-08 13:42:00	BTC/USD	242.95	242.95	242.95	242.95	0.000000	0.000000
3	1444311780	2015-10-08 13:43:00	BTC/USD	242.95	242.96	242.95	242.96	0.010000	2.429600
4	1444311840	2015-10-08 13:44:00	BTC/USD	242.96	242.96	242.96	242.96	0.033491	8.137003

D. **After the dataset is get preprocessed it is now ready for the further steps includes splitting of dataset**

Splitting of Data into Training Testing

In [36]:
```
# Split the data into training and testing sets
train_size = int(len(prices_scaled) * 0.8)
train, test = prices_scaled[0:train_size, :], prices_scaled[train_size:len(prices_scaled), :]
```

In [37]:
```
# Function to create dataset with look back
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back):
        a = dataset[i:(i + look_back), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return np.array(dataX), np.array(dataY)
```

In [38]:
```
# Create the dataset with look back
look_back = 400
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
```

E. Now initiailize and train the machine learning and deep learning models.

Machine Learning Model
1. Linear Regression
2. Ridge
3. K-Nearest Neighbors (KNN)
4. Support Vector Machine (SVM)

Deep Learning Model
1. Long Short-Term Memory (LSTM)
2. Recurrent Neural Network (RNN)

As the information representated on the above image is the proces or step to exceute the code implementation for bitcoin prediction prices.

The below image depicts the results for both Machine learning and deep learning models:

Comparison of Machine Learning & Deep Learning Models

In [74]:
```
#prints the machine learning metrics
machine_learning_metrics_df
```

Out[74]:

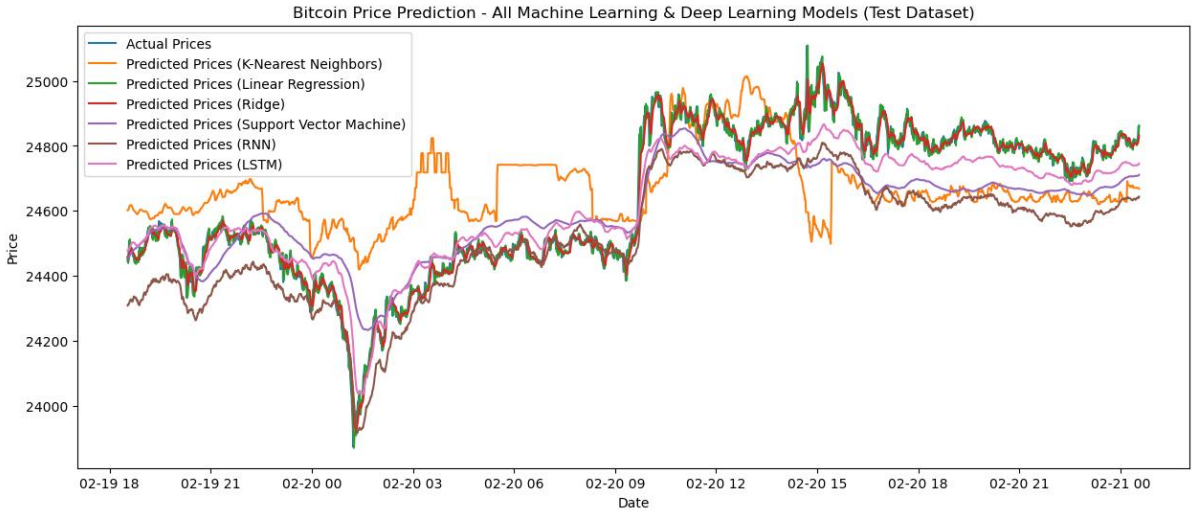
	Model	MAE	MSE	RMSE	R-squared
0	Linear Regression	24631.873895	6.067815e+08	24632.936107	-1.676393e+11
1	Ridge	24631.484772	6.067621e+08	24632.540525	-1.676339e+11
2	K-Nearest Neighbors	24667.818777	6.085138e+08	24668.073016	-1.681179e+11
3	Support Vector Machine (SVM)	24598.328314	6.050957e+08	24598.692410	-1.671735e+11

In [75]:
```
#prints the deep learning metrics
deep_learning_metrics_df
```

Out[75]:

	Model	MAE	MSE	RMSE	R-squared
0	Long Short-Term Memory (LSTM)	67.301227	6287.818019	79.295763	0.879798
1	Recurrent Neural Netwok (RNN)	118.905744	18991.219221	137.808633	0.636951

This plot demonstrates the predictions of all machine learning and deep learning models.



This configure manual provides the a comprehensive guide for configuring, execution of code , and understanding the Bitcoin future price forecasting implementation.

References

Anaconda:Kaggle Dataset Source: https://www.kaggle.com/