National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Sai Prasanna Gontyala
Student ID: X19233388

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sai Prasanna Gontyala
Student ID:	X19233388
Programme:	Data Analytics **Year:** 2021
Module:	MSc Research Project
Lecturer:	Dr. Catherine Mulwa
Submission Due Date:	16/08/2021
Project Title:	Prediction of Cryptocurrency Price based on Sentiment Analysis and Machine Learning Approach
Word Count:	1548 **Page Count:** 11

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: .....

Date: 16th August 2021

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

Sai Prasanna Gontyala
X19233388

1 Introduction

The system setup, software hardware specifications, and activities carried out for the implementation of the Research Project: Forecasting bitcoin values using Machine Learning are detailed in this configuration manual. Section 2 outlines the hardware and software requirements to implement the project. Section 3 illustrates the data collection. The process of merging the two datasets is covered in the section 4.

2 System Configuration

This section outlines the various specifications of the system and environment setup that had allowed the research to proceed uninterrupted.

2.1. Hardware Requirements

The project was conducted on the local system which had the below hardware specifications.

- System OS: Windows 10 Home 64-bit
- Processor: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
- RAM: 8.00 GB
- System Type: 64-bit operating system, x64-based processor.

2.2. Software Requirements

- Python 3.8 - This is a programming language that was utilized throughout the project from data preparation to evaluation of the predictive deep learning model.
- Jupyter Notebook - Jupyter Notebook is an open-source software that helps to execute the python code.
- Visual Studio Code - This is an open-source web application that helps in coding and execution of Python. It was used to execute Python 3.8 in this experiment.
- Microsoft Excel: The imported and extracted data is stored in python data frame initially and then converted into comma separated les after pre-processing of data and merging of twitter data with bitcoin time series data.
- Twitter Developer Account - Access for twitter developer is raised and got approved to generate the API token. The use case of the research, method of twitter data

extraction, the further use and storage process of the data extracted and the secured approach of maintaining the data were explained clearly to the twitter team to get approval for the developer access request. This helps in having a legal access to the twitter API to access the tweets related to bitcoin.

3 Data Collection

To achieve the objectives of the research project two datasets are required for the analysis- Tweets related to the bitcoin and Bitcoin price history which is in the form of time-series. The approach used for the extraction of the data are explained in detail.

3.1 Twitter

A Twitter development account would be required to generate API key and API secret key. Once a developer account is created, an APP should be created on the twitter developer portal[1]. Figure 1 helps in understanding the API key generation on the twitter portal.

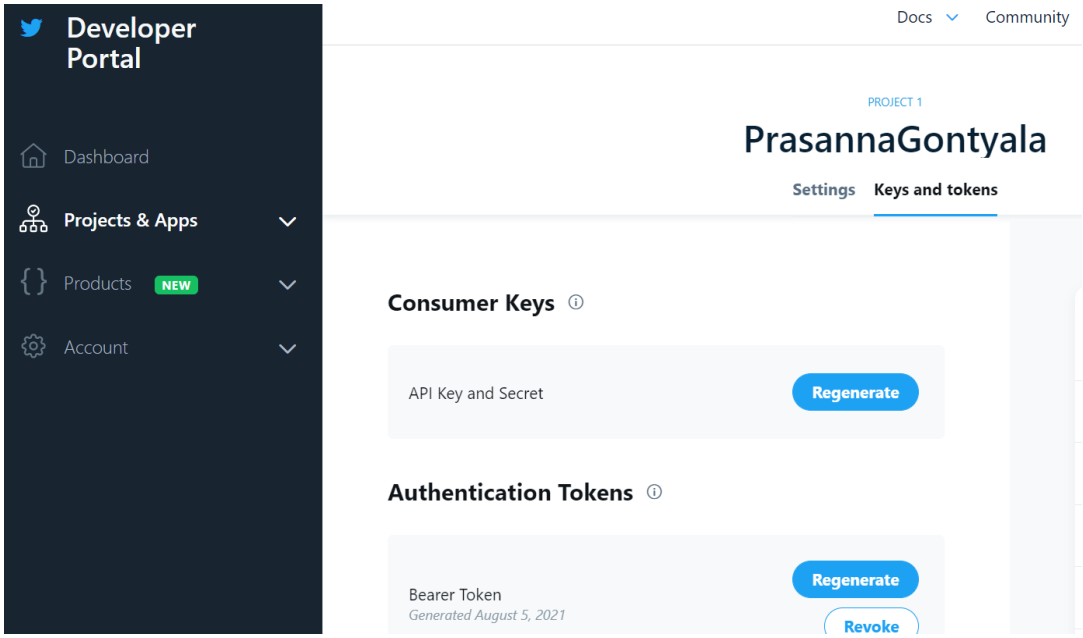

Figure 1: Twitter API key and secret key generation

The data is then read using the python libraries. Figure 1 helps in importing the python libraries required for the twitter data extraction.

[1] https://developer.twitter.com/en/portal/dashboard

```
import glob
import json
import os
import time
import urllib
import numpy as np
import pandas as pd
import datetime
from typing import List
from pandas.core.frame import DataFrame
```

Figure 2: Python libraries required to extract twitter data

The code for extracting the tweets is present in TwitterExtractor.py. While the python library "Twython" is used to extract the tweets. Figure 12 helps in understanding the methodology followed in research project.

```
def collect_tweets(self):
    twitter = Twython(self.AppKey, self.AppSecret, oauth_version=2)
    AccessToken = twitter.obtain_access_token()
    twitter = Twython(self.AppKey, access_token=AccessToken)
    twitter.get_application_rate_limit_status()["resources"]["search"]
```

Figure 3: Accessing twitter API using python libraries

The keywords "bitcoin" and "BTC" are searched for in the tweets to extract the tweets related to the bitcoin. As Twitter's access rate is limited and slow, and only a few queries per developer account are accepted, the dataset from an open-source is used[2]. The data from May 11th, 2018 at 9:00 a.m. to May 29th, 2018 at 12:00 p.m. is considered from the open-source datasets. Although the code is capable of retrieving most tweets but the data is less for deep learning analysis. Figure 4 helps in knowing the key words that are used to search tweets related to bitcoin. Figure 5 shows the code where tweets data is read into comma separated file.

[2] https://github.com/Drabble/TwitterSentimentAndCryptocurrencies/tree/master/data/ twitter/BTC

```
CRYPTO = "bitcoin"
CRYPTO_SYMBOL = "BTC"
APP_KEY = 'usOubgHZwj6bnewUAbB73GOVf'
APP_SECRET = 'jUnJoK7xoi3NeVmGd2b9Qb2lWgfTAPjn8dUtSh3HTqRwwS2ghY'

TWITTER_FOLDER = "data/twitter"
SEP_CHAR = "~"  # character seperating dates from and to in filename
MAX_ROW_PER_FILE = 20000  # Each file storing data has a maximum amount of rows
NUMBER_OF_QUERIES = 450
```

Figure 4: Key words and parameters used to extract bitcoins

```
with open(self._raw_file, "a+", encoding="utf-8") as f:
    #print(os.fstat(f.fileno()).st_size)
    if os.fstat(f.fileno()).st_size < 2:
        f.write('ID,Text,UserName,UserFollowerCount,RetweetCount,Likes,CreatedAt\n')
    while(True):
        twitter = Twython(self.AppKey, access_token=AccessToken)
        last_size = 0
        for _ in range(self.NumOfQueries):
            if not self.NextID:
                # Use since_id for tweets after id
                data = twitter.search(
                    q=query, lang='en', result_type='recent',
                    count="100")
```

Figure 5: Raw file structure defined

3.2 Bitcoin price time-series data

Bitcoin price data is available at many open sources in the time frame of yearly, monthly, weekly and hourly data. The data is extracted using an open-source API[3] which helps in extracting hourly data which is then converted into comma separated le. Since the analysis of the research project is done considering the old tweets from May 11th, 2018 at 9:00 a.m. to May 29th, 2018 at 12:00 p.m. the hourly bitcoin prices for the same time frame are extracted from the open source[4].

The libraries mentioned in Figure 5 are used to extract bitcoin time-series price data.

- **Numpy:** NumPy is a Python library for working with arrays. It provides a variety of resources for interacting with arrays at the highest possible speed.

[3] https://min-api.cryptocompare.com/
[4] https://github.com/Drabble/TwitterSentimentAndCryptocurrencies/tree/master/data/ crypto/BTC

- **Urllib:** The Urllib module is a Python module for managing URLs. It's used to get URLs from the internet (Uniform Resource Locators). It makes use of the urlopen function and may retrieve URLs using a variety of protocols.
- **Glob:** By combining the glob () method with the getmtime () method in the os module, we can order the files based on the date and time of change. We may use the glob () function to delete files from directories by iterating through the list and then executing os.remove () for each file.

```
import glob
import json
import os
import time
import urllib
import numpy as np
import pandas as pd
import datetime
from typing import List
from pandas.core.frame import DataFrame
```

Figure 5: Python libraries required to extract bitcoin data

```
CRYPTO_FOLDER = "data/crypto"
CRYPTO_BASE_URL = "https://min-api.cryptocompare.com/data/histominute"
CURRENCY = "USD"  # Available currency in the API
# API advises us to give an app name in requests
CRYPTO_APP_NAME = "HES_SO_master_crypto_analysis"
_datatype = {"LINE_COUNT": np.int32}
```

Figure 6: Bitcoin API used in the research to extract price data

```
self.url = f"{self.BaseURL}?fsym={self.Crypto}&tsym={self.Currency}" + \
    f"&limit={self.Limit}&toTs={mytoTs}&extraParams={self.AppName}"
contents = urllib.request.urlopen(self.url).read()
#print(contents)
json_string = contents.decode("utf-8")
obj = json.loads(json_string)
df = pd.DataFrame.from_dict(obj["Data"])
if not df.empty:
    return df.drop(["volumefrom", "volumeto"], axis=1)
return df
```

Figure 7: Bitcoin API used in the research to extract price data

```
# Retrieve data for the 7 past days until last_ts
while(True):
    #print("last_ts")
    #print(last_ts)
    df = self.fetch_data(last_ts)
    if(df.empty):
        break
    df_historical = df_historical.append(df)
    min_time = df["time"].iloc[0]
    max_time = df["time"].iloc[-1]
    last_ts = min_time
    #total_wished = total_wished - df.shape[0]
    self.update_var(self.DFHeaders[3], max_time)
```

Figure 8: Understanding the bitcoin data retrieval for recent 7 days

Along with the static data considered from the open source for analysis. The research project is also capable of retrieving the most recent tweets and most recent seven days bitcoin price data and predict the bitcoin price for the next hour which is illustrated through Figure 8. While the structure of the bitcoin price dataset is explained in Figure 9.

1	close	high	low	open	time
2	8737.87	8747.28	8737.63	8745.27	1526032500
3	8745.14	8745.14	8737.76	8738.79	1526032560
4	8748.48	8753.09	8744.79	8745.14	1526032620
5	8746.25	8748.76	8742.89	8748.48	1526032680
6	8753.83	8753.86	8743.8	8746.25	1526032740
7	8785.81	8786.22	8755.28	8755.45	1526032800
8	8786.51	8793.96	8785.81	8785.81	1526032860
9	8774.75	8788.85	8774.75	8786.51	1526032920
10	8770.64	8778.77	8768.95	8774.18	1526032980
11	8771.9	8777.47	8770.64	8770.64	1526033040
12	8778.29	8779.21	8769.54	8771.11	1526033100
13	8778.2	8779.74	8777.13	8778.33	1526033160
14	8777.14	8778.2	8773.91	8778.2	1526033220
15	8775.42	8778.09	8774.6	8776.98	1526033280
16	8772.17	8775.17	8770.67	8775.17	1526033340
17	8751.44	8772.21	8751.44	8772.17	1526033400
18	8753.78	8759.3	8747.99	8749.43	1526033460

2018-05-11 11-55-00~2018-05-18

Figure 9: Bitcoin sample hourly price data

4 Data Merge

Sentiment score against each extracted tweet is calculated using the formula mentioned in Figure 10. The twitter data with the score is then combined with the bitcoin price dataset based on the timestamp.

```
for i, s in enumerate(df_clean["Text"]):
    vs = analyzer.polarity_scores(s)
    compound.append(vs["compound"])
df_clean["compound"] = compound
scores = []
for _, s in df_clean.iterrows():
    scores.append(s["compound"] *
                  ((s["UserFollowerCount"] + 1)) * ((s["Likes"] + 1)))
df_clean["score"] = scores
```

Figure 10: Sentiment score calculation

Below figure explains the structure of the merged data which has the aggregated sentiment score for every hour and the corresponding close price of bitcoin.

	time	score	close
0	0	4964373.047	8746.314
1	1	-17299015.75	8759.485
2	2	7207900.386	8713
3	3	9280097.309	8555.751
4	4	28492620.45	8570.464
5	5	-7964296.984	8610.821
6	6	-75865889.73	8612.679
7	7	25878413.5	8619.968
8	8	-848532.6407	8626.546
9	9	18599446.39	8614.144
10	10	2652400.801	8649.972
11	11	17261199.43	8552.144
12	12	67920764.8	8419.266
13	13	12943778.21	8440.714
14	14	-12704873.96	8445.545
15	15	2788210.653	8459.46
16	16	3900609.016	8450.541

merged_data

Figure 11: Merged data post sentiment analysis

5 Predictive Model and Evaluation Requirements

Below mentioned python libraries were used as a part of the prediction Model build and to evaluate the results obtained from the model.

- Sklearn: Sklearn also widely known as Scikit-learn. Rather than importing, editing, and summarizing data, the Scikit-learn library concentrates on data modelling. It is Python's most useful and stable machine learning library. It uses a Python consistency

interface to give a set of efficient tools for machine learning and statistical modelling, such as classification, regression, clustering, and dimensionality reduction.

- Keras: Keras is an open-source high-level Neural Network library written in Python that can be used with Theano, TensorFlow, and CNTK. It seamlessly runs on CPU as well as GPU.
- Matplotlib: Matplotlib is an amazing Python visualization package for 2D array charts. It is a multi-platform data visualization package based on NumPy arrays and intended to operate with the SciPy stack as a whole.
- Dash html components: Dash Html Components is a web application framework that wraps HTML, CSS, and JavaScript in a pure Python abstraction. The dash-html-components library allows you to create your layout using Python structures rather than writing HTML or using an HTML templating engine. The results of the project will be displayed on a dash html component.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from pandas import datetime
import math, time
import itertools
from sklearn import preprocessing
import datetime
from operator import itemgetter
from sklearn.metrics import mean_squared_error
from math import sqrt
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.layers.recurrent import LSTM
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_poisson_deviance
from sklearn.metrics import mean_gamma_deviance
from sklearn.metrics import mean_tweedie_deviance
```

Figure 12: Python libraries required for LSTM model build and evaluation

```
import dash
import dash_html_components as html

from dotenv import load_dotenv
from torch import nn
from pytorch_lightning import Trainer, seed_everything
from pytorch_lightning.loggers.csv_logs import CSVLogger
from predictor.Interface import DataVizInterface
from predictor.TwitterExtractor import TwitterExtractor
from predictor.CryptoCurrencyExtractor import CryptoCurrencyExtractor
from predictor.Model import TweetAndCryptoDataModule, LSTMRegressor
import time
import numpy as np
import pandas as pd
import datetime
from RNN import RNN_Model
```

Figure 11: Merged data post sentiment analysis

References

Anon, (n.d.). [online] Available at: https://towardsdatascience.com/ predicting-bitcoin-prices-with-deep-learning-438bc3cf9a6f.

Anon, (n.d.). [online] Available at: https://towardsdatascience.com/ apple-stock-and-bitcoin-price-predictions-using-fbs-prophet-for-beginners-python-96d5ec404b77.