National College of Ireland

Configuration Manual

MSc Research Project
MSc Data Analytics

Deeksha Chaudhry
Student ID: 20154267

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Deeksha Chaudhry……

Student ID: x20154267……

Programme: Masters …….. **Year:** ……2021……..

Module: Data Analytics……

Lecturer: Vladimir Milosavljevic ……

Submission Due Date: 16/12/2021……

Project Title: … Covid-19 Vaccine Understanding and Aftereffects: Twitter User Mental Health Analysis and Pfizer Stock Predictions in Major Regions

Word Count: ……565…… **Page Count:** ……8……

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: Deeksha Chaudhry……

Date: 16/12/2021……

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

Deeksha Chaudhry
Student ID: 20154267

1 Introduction

The program's setup procedures have been broken down into sub parts for easy execution. This manual contains a full explanation of the entire implementation. This has been arranged to make them simpler to comprehend:

- Python Environment Setup
- Downloading required toolkits, programs and libraries
- Save data files (csv and text) at required location
- Implementation of the python code
- Visualizations

2 Prerequisites

Our solution requires the installation of a Python environment. The following are the system requirements for the effective completion of this research study

- RAM: 8GB
- Hard Disk: 1TB
- Operating System: Windows 10
- Programming Language: Python3.8.8

After we have verified that all the prerequisites have been met and Python has been installed, we must save two files: emotions and the Twitter data file. These files should be saved somewhere where they may be accessed when executing the programme. Along with this, specific libraries must be imported, a screenshot of which is provided below –

In [1]:
```
import pandas as pd
import numpy as np
import yfinance as yf
import tensorflow as tf
import plotly.express as px
from tqdm import tqdm
from wordcloud import WordCloud,STOPWORDS
import matplotlib.pyplot as plt
from collections import Counter
import collections
from plotly.subplots import make_subplots
import plotly.graph_objects as go
from sklearn.preprocessing import StandardScaler
```

```
[2]: import nltk
     nltk.download('vader_lexicon')
     nltk.download('stopwords')
     nltk.download('punkt')
```

```
In [3]: from plotly.offline import init_notebook_mode
        init_notebook_mode(connected=True)
```

```
In [4]: import warnings
        warnings.filterwarnings(action="ignore")
```

```
In [5]: import re
        from nltk.tokenize import word_tokenize
        from string import punctuation
        from nltk.corpus import stopwords
        from nltk.sentiment.vader import SentimentIntensityAnalyzer
        from collections import Counter
        import string
```

We may begin implementation after all of the prerequisites have been satisfied.

3 Implementation

To read the data we need to import the twitter data file using below command.

Reading Data

```
tweets_data = pd.read_csv("last_3_month_vaccination_tweet_scrape.csv",infer_datetime_format=True,parse_dates=["date"])
```

To extract the yahoo finance data which would be used for Pfizer stock prices prediction we must download data from yahoo finance.

```
pzier_stock = yf.download('PFE',
                      start='2021-10-01',
                      end='2021-12-04',
                      progress=True,
)
pzier_stock.head()
```

We will now visualize the data to have a better grasp of it. After the data has been visualized, it is preprocessed and cleaned in preparation for future deployment. Once the data has been cleaned, the resulting words are processed and saved in final words.

```
final_words = []
for i in tqdm(range(len(data)),desc ="Processing Tweets"):
    final_words = final_words+ processTweet(data.content[i])
```

```
Processing Tweets: 100%|██████████| 146588/146588 [3:53:51<00:00, 10.45it/s]
```

Mental Health Analysis - Once we get a list of final words, we will use the emotion file for mental health analysis. All extracted words are compared to the emotion list, and each extracted word is assigned to one of the emotions described in the emotion list.

```
Counter({' cheated': 12,
         ' loved': 7,
         ' attracted': 24,
         ' sad': 46,
         ' fear': 41,
         ' happy': 45,
         ' angry': 34,
         ' bored': 5,
         ' esteemed': 7,
         ' lustful': 4,
         ' attached': 12,
         ' independent': 4,
         ' embarrassed': 3,
         ' surprise': 8,
         ' powerless': 20,
         ' fearless': 10,
         ' safe': 2,
         ' adequate': 9,
         ' alone': 18,
```

Stock Prediction –

Before training and testing stock data for machine learning models we visualized the data for more clarity about the stock prices.

```
# Create subplots and mention plot grid size
fig = make_subplots(rows=2, cols=1, shared_xaxes=True,
                vertical_spacing=0.03, subplot_titles=('OHLC', 'Volume'),
                row_width=[0.2, 0.7])
```

Prepared the tweets data and arranged them by date.

```
temp_df = tweets_data[["content","date"]]
temp_df.date = temp_df.date.dt.date
temp_df.sort_values(by = "date",ignore_index=True,inplace=True)
```

Processing all the tweets by sentiment analysis as numerical value so that it could help us in predicting the stock prices.

```
final_df = pd.DataFrame(columns=["negative_score","neutral_score","positive_score","compound_score","Price"])
```

4 Model Comparisons

Dividing the data into test and train data for machine learning algorithms

```
X_train = final_df.iloc[0:-7,:-1]
Y_train = final_df.iloc[0:-7,-1:]

X_test = final_df.iloc[-7:,:-1]
Y_test = final_df.iloc[-7:,4:]
```

4.1 Neural Network

Execution

```
model_neural = tf.keras.models.Sequential()
model_neural.add(tf.keras.layers.Dense(64,input_shape=(4,),activation="relu"))
model_neural.add(tf.keras.layers.Dense(128,activation="relu"))
model_neural.add(tf.keras.layers.Dense(512,activation="relu"))
model_neural.add(tf.keras.layers.Dense(128,activation="relu"))
model_neural.add(tf.keras.layers.Dense(1,activation="relu"))
```

```
model_neural.compile(optimizer="adam",loss="mean_squared_error",metrics=["mae"])
history_neural = model_neural.fit(X_train,Y_train,validation_data=(X_test,Y_test),epochs=50)
```

Result

```
2/2 [==============================] - 0s 17ms/step - loss: 137.0325 - mae: 9.4325 - val_loss: 7288.6919 - val_mae: 63.0878
```

4.2 LTSM

Execution

```
X_train = X_train.reshape((-1,1,4))
X_test = X_test.reshape((-1,1,4))
```

```
model_lstm = tf.keras.models.Sequential()
model_lstm.add(tf.keras.layers.LSTM(128,input_shape=(1,4),activation="tanh",return_sequences=True))
model_lstm.add(tf.keras.layers.LSTM(512,activation="tanh"))
model_lstm.add(tf.keras.layers.Dense(128,activation="relu"))
model_lstm.add(tf.keras.layers.Dense(512,activation="relu"))
model_lstm.add(tf.keras.layers.Dense(128,activation="relu"))
model_lstm.add(tf.keras.layers.Dense(1))
```

Result

```
Epoch 50/50
2/2 [==============================] - 0s 20ms/step - loss: 11.9095 - mae: 2.6813 - val_loss: 305.9074 - val_mae: 14.4660
```

4.3 Bi- LTSM

Execution

```
model_bilstm = tf.keras.models.Sequential()
model_bilstm.add(tf.keras.layers.LSTM(128,input_shape=(1,4),activation="tanh",return_sequences=True))
model_bilstm.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(512,activation="tanh",return_sequences=True)))

model_bilstm.add(tf.keras.layers.Dense(128,activation="relu"))
model_bilstm.add(tf.keras.layers.Dense(512,activation="relu"))
model_bilstm.add(tf.keras.layers.Dense(128,activation="relu"))
model_bilstm.add(tf.keras.layers.Dense(1))
```

```
model_bilstm.compile(optimizer="adam",loss="mean_squared_error",metrics=["mae"])
history_bilstm = model_bilstm.fit(X_train,Y_train,validation_data=(X_test,Y_test),epochs=50)
```

Result

```
Epoch 50/50
2/2 [==============================] - 0s 33ms/step - loss: 10.0559 - mae: 2.8135 - val_loss: 74.9255 - val_mae: 7.6853
```

Following that, we compared all three models, calculating the train loss and validation loss comparison for each. By comparing these models, it was evident that the neural network could not be employed as a prediction model, but the other two produced good results.

References

Glowacki, E., Wilcox, G. and Glowacki, J. (2020). Identifying addiction concerns on twitter during the covid-19 pandemic: A text mining analysis, Substance abuse 42: 1–8.

Khattar, A., Jain, P. R. and Quadri, S. M. K. (2020). Effects of the disastrous pandemic covid 19 on learning styles, activities and mental health of young indian students - a machine learning approach, 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1190–1195.

Tariq Soomro, Z., Waseem Ilyas, S. H. and Yaqub, U. (2020). Sentiment, count and cases: Analysis of twitter discussions during covid-19 pandemic, 2020 7th International Conference on Behavioural and Social Computing (BESC), pp. 1–4.

Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q. and Wang, J. (2021). A stock closing price prediction model based on cnn-bislstm. URL: https://www.hindawi.com/journals/complexity/2021/5360828/