National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Tejveer Singh Goraya
Student ID: 19202687

School of Computing
National College of Ireland

Supervisor: Dr. Bharathi Chakravarthi

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Tejveer Singh Goraya
Student ID:	19202687
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Bharathi Chakravarthi
Submission Due Date:	16/12/2021
Project Title:	Configuration Manual
Word Count:	1,851
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:	Tejveer Singh Goraya
Date:	15th December 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

Tejveer Singh Goraya
19202687

1 Introduction:

This document lists the requirements and steps to configure and execute the files associated with this research. Apart from these steps, the document also contains information about the coding methodology applied in order to create **Modular Code** which can be taken apart and put together with ease in order to achieve the desired results.

2 System Specifications:

This section discusses the system specifications that are required to execute the code files related to this research. It is divided into two parts which discuss the Software and the Hardware requirements. **Later sections of this document also contain a demonstration of this research with the help of a Dashboard.**

2.1 Software Specifications:

This research project is Completely executed on the Google Colab Platform and it can be accessed simply by using a browser. The minimum requirements for a browser that supports Google Colab(All browsers which have the latest update of 2021 will support Google Colab) are listed below:

- OS: Mac OS, Windows, Linux.
- Browser: Chrome,Safari,Firefox.
- Microsoft Excel.
- Latex Plugins or Overleaf(Only required for report).

The primary reason for using Google Colab is its cloud computing capability as it reduces the load on personal machines and has enormous computing power which far exceeds the power of a Personal Computer(Carneiro et al. (2018)).

2.2 Hardware Specifications:

As this project is being executed on the Google Colab platform, it can be run on any machine with a browser and an internet connection. Google Colab provides the below Hardware configuration in terms of its computing capabilities:

Parameter	Google Colab
CPU Model Name	Intel(R) Xeon(R)
CPU Freq.	2.30GHz
No. CPU Cores	2
CPU Family	Haswell
Available RAM	12GB (upgradable to 26.75GB)
Disk Space	25GB

Figure 1: CPU specs for Google Colaboratory

Parameter	Google Colab
GPU	Nvidia K80 / T4
GPU Memory	12GB / 16GB
GPU Memory Clock	0.82GHz / 1.59GHz
Performance	4.1 TFLOPS / 8.1 TFLOPS
Support Mixed Precision	No / Yes
GPU Release Year	2014 / 2018
No. CPU Cores	2
Available RAM	12GB (upgradable to 26.75GB)
Disk Space	358GB

Figure 2: GPU specs for Google Colaboratory

Training and Testing large amounts of data can take long hours and during this the Online Code execution platforms need to have longer execution time limits and Idle times.

Parameter	Google Colab
Max execution time	12 hours
Max idle time	90 min

Figure 3: NIFTY 50 Index Value - Outliers

Google Colab provides a Max execution time of 12 hours and a Max idle time of 90 Minutes, which is far more than the time required to execute this research project files.

The figure below shows the landing page of Google Colab with the navigation buttons located on the **top left** corner of the page.

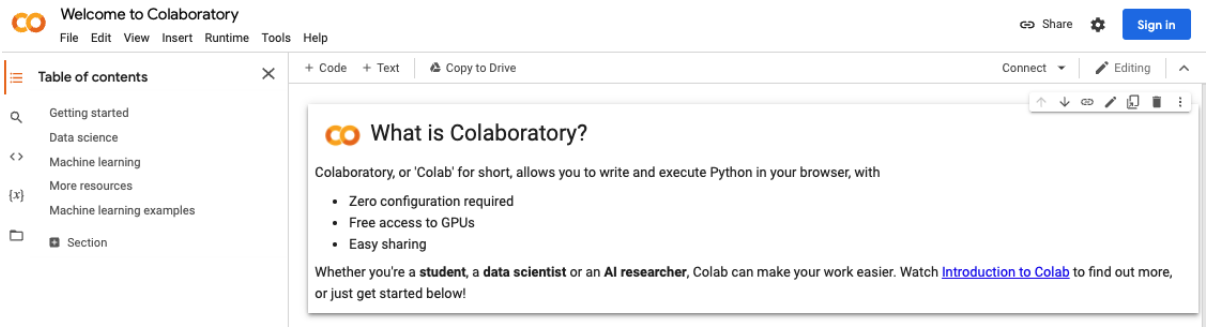

Figure 4: Google Colaboratory Landing page

2.3 Loading any of the Code Books on Google Colab:

The First Step to upload any of the code books from the research artefacts folder is to **Click on the File - Upload Notebook** as shown below.

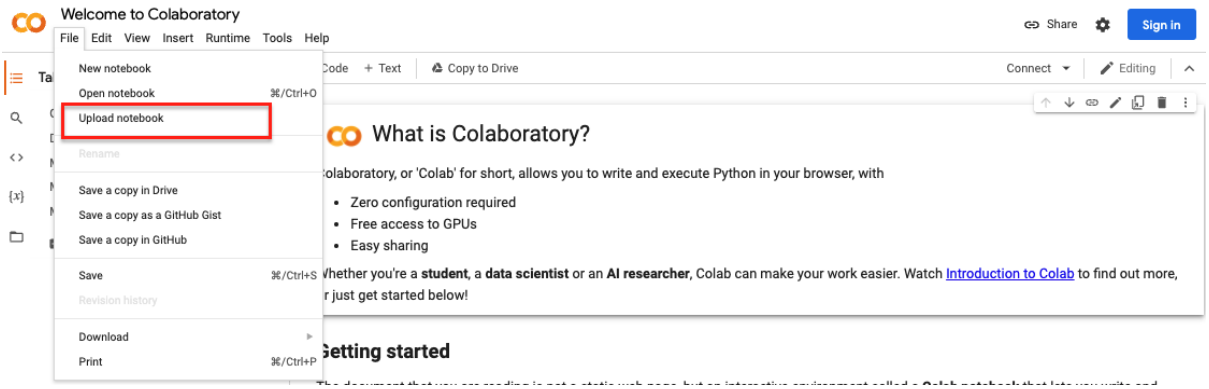

Figure 5: Uploading Notebooks to Google Colaboratory

This can be done to upload any Jupyter notebook onto the Google Colab platform as all the notebooks in the folder are executed primarily on this platform.

3 Programming Methodology and Code Structure:

This section summarizes the flow of the program and the modular approach taken to write the code for executing the machine learning models in this research:

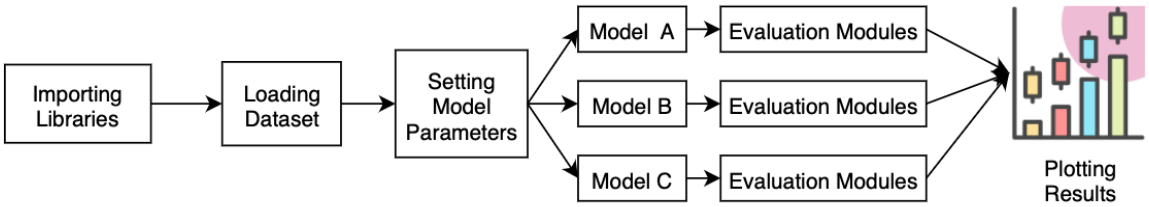

Figure 6: Flow of program

The code books/programs are mainly differentiated in 6 majors steps. **The code in this project is developed in a way that any of the pieces on the above flow diagram can be interchanged for any of the applied models.** *All the Variables names used in the code files are uniform and easy to debug.*

3.1 Step 1: Importing Libraries

The figure below shows the list of of all imported libraries and the modules under those libraries in order to achieve the results.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import yfinance as yf
from plotly.offline import iplot, init_notebook_mode
init_notebook_mode()
from matplotlib import *
import sys
from pylab import *
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
plt.style.use("fivethirtyeight")
%matplotlib inline
import matplotlib.dates as mdates
import matplotlib.ticker as mticker
import plotly.graph_objs as go
from  plotly.offline import plot
import chart_studio.plotly as py
import cufflinks as cf
cf.go_offline()
from plotly.offline import download_plotlyjs, init_notebook_mode, plot,iplot
init_notebook_mode(connected='true')
```

Figure 7: Step 1: Importing Libraries

Below table summarizes the libraries imported in the project in order to ensure with the systems current compatibility(Gevorkyan et al. (2019)).

pandas	tensorflow
numpy	tensorflow.keras.models
matplotlib	tensorflow.keras.preprocessing
datetime	sklearn
warnings	seaborn
skimage	tqdm
yfinance	sklearn.neighbors
Sys	sklearn.ensemble
sklearn.preprocessing	sklearn.grid_search
sklearn.metrics	sklearn.model

Figure 8: Summary Table of Libraries

3.2 Step 2: Loading Datasets

The initial approach for obtaining the datasets in this project was to download the datasets from any of the available open Stock Exchanges such as the National Stock Exchange Of India(NSE) or Google Finance. But downloading the dataset and uploading it to the code file adds an additional step to the process and increases manual work. In order to reduce redundancy, this research makes use of an open source finance API

provided by Yahoo Finance (Data (2021)). The code to obtain this data is as shown below:

```
df = yf.download('^NSEI',start='2021-01-01',end='2021-11-30')
df['Date'] = pd.to_datetime(df.index)
df = df.reset_index(drop=True)
df = df[['Date','Open','High','Low','Close','Adj Close','Volume']]
df
```

Figure 9: Dataset Download

As seen in the above figure, using the finance library by yahoo is a robust way of obtaining stock market data. The start date indicated the starting date range for which data is required and the end date indicates the last date range of the data. The module takes the Ticker symbol as the input in order to provide data of any stock listed on the stock market.

3.3 Step 3: Setting the Model Parameters

This step is a necessary step in order to set some model as well as environment parameters for the research.**In order to run the models, it is important that the Google Colab notebook has Tensorflow Version = 1.14.0.**. To install this version of tensorflow, it is important to run **!pip install tensorflow==1.14.0** command in the notebook cell. Once this step is completed the below piece of code is used to set the parameters for the models:

```
test_size = 30
simulation_size = 10

df_train = df_log.iloc[:-test_size]
df_test = df_log.iloc[-test_size:]
df.shape, df_train.shape, df_test.shape
```

Figure 10: Settings for Epoch tuning.

Here, **simulation_size is the number of epochs and it can be adjusted as per requirement and test_size is the number of days the model is forecasting in future.**The above figure shows the section of code which is used to adjust the epochs of the model.Each code book in the repository in standardized in a way that this part of the code is like a dashboard tool which can be adjusted in order to improve or test the outcome.

3.4 Step 4: Applying Models

The below figure shows the list of code books in the artefacts folder which contains all the models that are applied to two separate datasets based on time periods.

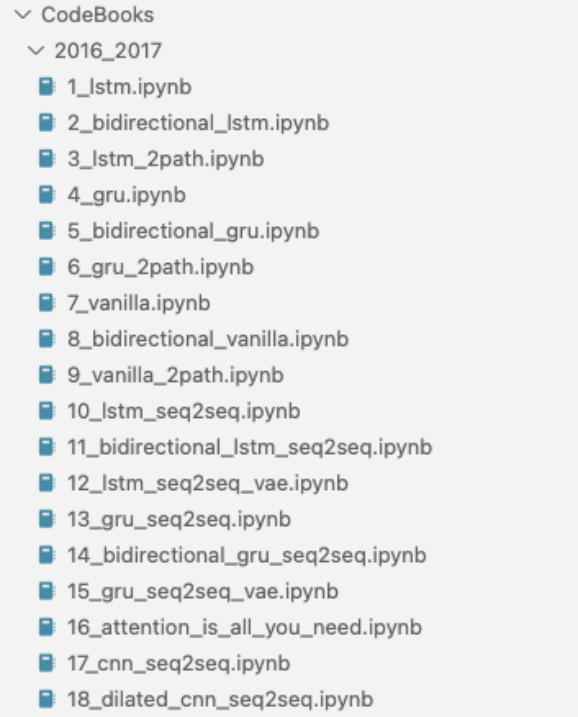

Figure 11: Models for Dataset 2016-2017

Figure 12: Models for Dataset 2021

All the above files have been structured as per the methodology in *figure 6.*

3.5 Step 5: Evaluation of the models

This section discusses the evaluation modules which are developed with an agile approach.

3.5.1 Accuracy Measure

The below code is a generalised part of code which can be entered in any of the Code Books in order to evaluate Accuracy of the Model.This set of code can be inserted in the last cell of any of the Code Books in order to get Accuracy

```
#Accuracy
accuracies = [calculate_accuracy(df['Close'].iloc[-test_size:].values, r) for r in results]
```

Figure 13: Accuracy Measure

3.5.2 Root Mean Square Error

The below code is a generalised part of code which can be entered in any of the Code Books in order to evaluate Root Mean Square Error of the Model.This set of code can be inserted in the last cell of any of the Code Books in order to get Root Mean Square Error.

```
#Root Mean Square Error

import math
import sklearn.metrics
mse = sklearn.metrics.mean_squared_error(df['Close'].iloc[-test_size:].values,r)
rmse = math.sqrt(mse)
print(rmse)
```

Figure 14: Root Mean Square Error

3.5.3 Mean Absolute Error

The below code is a generalised part of code which can be entered in any of the Code Books in order to evaluate Mean Absolute Error of the Model.This set of code can be inserted in the last cell of any of the Code Books in order to get Mean Absolute Error.

```
#Mean Absolute Error

from sklearn.metrics import mean_absolute_error

mean_absolute_error(df['Close'].iloc[-test_size:].values, r)
```

Figure 15: Mean Absolute Error

3.6 Step 6: Plotting the results

After achieving the results of the above models, the below piece of code is used in order to plot the graphs used in the research paper.

```
#Accuracy
accuracies = [calculate_accuracy(df['Close'].iloc[-test_size:].values, r) for r in results]

plt.figure(figsize = (9, 6))
for no, r in enumerate(results):
    plt.plot(r, label = 'forecast %d'%(no + 1))
plt.plot(df['Close'].iloc[-test_size:].values, label = 'true trend', c = 'black')
plt.legend()
plt.title('average accuracy: %.4f'%(np.mean(accuracies)))
plt.show()
```

Figure 16: Plotting the results

4 Demonstration of the Research Outcome with a Dashboard:

This section discusses a real world application of this thesis and an interactive demonstration as an additional feature(Brath and Peters (2004)).

To demonstrate the output this research and showcase a real world application, a dashboard has been developed as shown below:

Figure 17: Main Dashboard

This is an interactive dashboard with multiple functionalities:

4.1 Step 1 : Pick a CSV File

As shown in the figure above, the user can select any CSV file which contains information about any stock with Date,Open,Close,High,Low,Volume Data. This functionality comes in handy as the user can apply the models in the next step for any of the listed stocks.

4.2 Step 2 : Set the Parameters

The user can set the parameters for the models by giving an input for the below mentioned values:

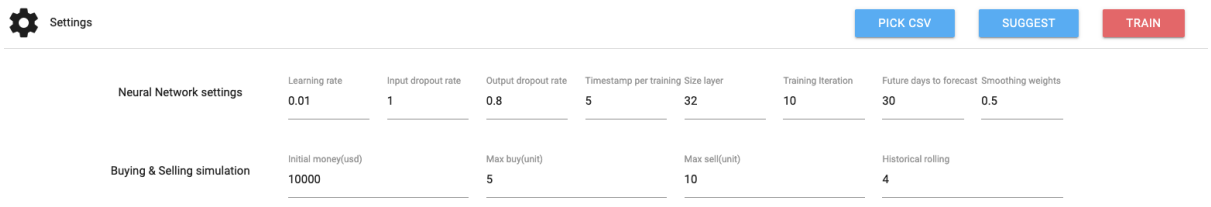

Figure 18: Model Parameters

4.3 Step 3 : Training the Model

After uploading the CSV and setting the training parameters, the models are trained in the backend and they produce outputs as shown below:

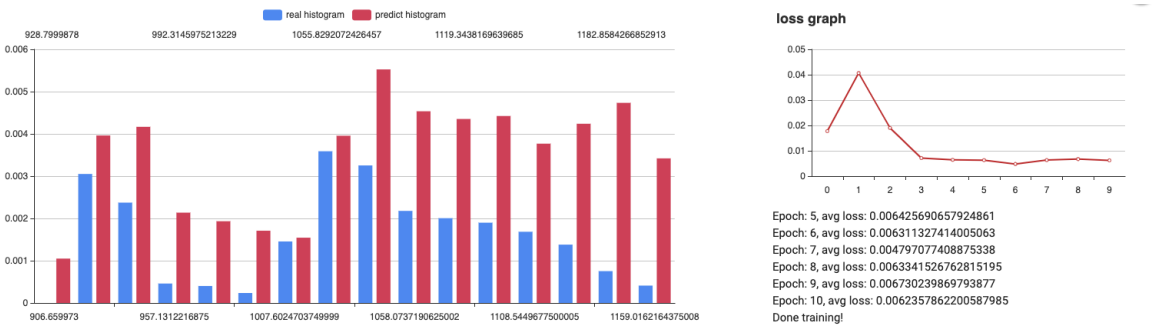

Figure 19: Model Output

4.4 Step 4 : Buying/Selling Signals

Based upon the models that are developed, the dashboard will provide Buying/Selling signals in form of log files and represent these signals on the price graph as shown below:

Figure 20: Buying/Selling Chart

The above figure represents when the simulation gives a buy/sell signal on the chart:

2018-06-06	sell 5 units	5684.400025	5.296896991281674%	10597.250974999999
2018-06-21	buy 5 units	5788.30017	NULL	4808.950804999999
2018-7-8	sell 5 units	5871.461626099467	1.3830264652054949%	10680.412431099467
2018-7-13	buy 5 units	5653.704065909359	NULL	5026.7083651901075
2018-7-20	sell 5 units	5797.814263053979	2.54895189887235%	10824.522628244085
2018-7-28	buy 5 units	5719.202544498257	NULL	5105.320083745828
2018-8-3	sell 5 units	5892.666201156224	3.033004257295175%	10997.986284902052

Overall gain: 997.9862849020519, Overall investment: 9.97986284902052%

Figure 21: Buying/Selling Logs

This dashboard also calculates the starting capital as well as the capital gained after the simulations have ended.

5 Other Important Mentions:

The below set of notebooks can be run using the same steps as mentioned above:

- Forecasting Models.
- Monte Carlo Simulations.
- Outlier Detection Notebooks
- Other Agent works used for demonstration purposes.

These notebooks are used for either deducing the results or get output for demonstration purposes.

References

Brath, R. and Peters, M. (2004). Dashboard design: Why design is important, *DM Direct* **85**: 1011285–1.

Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C. and Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications, *IEEE Access* **6**: 61677–61685.

Data, U. (2021). Yahoo! finance.

Gevorkyan, M. N., Demidova, A. V., Demidova, T. S. and Sobolev, A. A. (2019). Review and comparative analysis of machine learning libraries for machine learning, *Discrete and Continuous Models and Applied Computational Science* **27**(4): 305–315.