Configuration Manual

MSc Research Project
MSc in Data Analytics

Sankalp Saoji
Student ID: x17154171

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sankalp Saoji		
Student ID:	X17154171		
Programme:	MSc in Data analytics	**Year:**	1st year
Module:	MSc Research project		
Supervisor:	Dr. Catherine Mulwa		
Submission Due Date:	18th April 2019		
Project Title:	Air pollutant concentration prediction using deep learning techniques for smart city: Beijing		
Word Count:	1070 **Page Count** 12		

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: 18th April 2019

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

Sankalp Saoji

X17154171

1 Windows Configuration

1.1 System Information

The table 1 below shows the system information of the device used. The system information consists data of the hardware and operating system configurations used for this research project.

Table 1: System information

Item	Value
OS Name	**Microsoft Windows 8.1 Single Language**
Version	**6.3.9600 Build 9600**
Other OS Description	**Not Available**
OS Manufacturer	**Microsoft Corporation**
System Name	**HP**
System Manufacturer	**Hewlett-Packard**
System Model	**HP Pavilion Notebook**
System Type	**x64-based PC**
System SKU	**M2W75PA#ACJ**
Processor	**Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz, 2200 Mhz, 2 Core(s), 4 Logical Processor(s)**
BIOS Version/Date	**Insyde F.52, 03-09-2015**
SMBIOS Version	**2.8**
Embedded Controller Version	**89.37**
BIOS Mode	**UEFI**
BaseBoard Manufacturer	**Hewlett-Packard**
BaseBoard Model	**Not Available**
BaseBoard Name	**Base Board**
Platform Role	**Mobile**

Secure Boot State	**Off**
PCR7 Configuration	**Binding Not Possible**
Windows Directory	**C:\Windows**
System Directory	**C:\Windows\system32**
Boot Device	**\Device\HarddiskVolume2**
Locale	**India**
Hardware Abstraction Layer	**Version = "6.3.9600.17196"**
User Name	**HP\HP-PC**
Time Zone	**India Standard Time**
Installed Physical Memory (RAM)	**16.0 GB**
Total Physical Memory	**15.9 GB**
Available Physical Memory	**11.9 GB**
Total Virtual Memory	**18.8 GB**
Available Virtual Memory	**14.4 GB**
Page File Space	**2.88 GB**
Page File	**C:\pagefile.sys**
Hyper-V - VM Monitor Mode Extensions	**Yes**
Hyper-V - Second Level Address Translation Extensions	**Yes**
Hyper-V - Virtualization Enabled in Firmware	**No**
Hyper-V - Data Execution Protection	**Yes**

The figure 1 below depicts the system configuration mentioned in the above table 1.

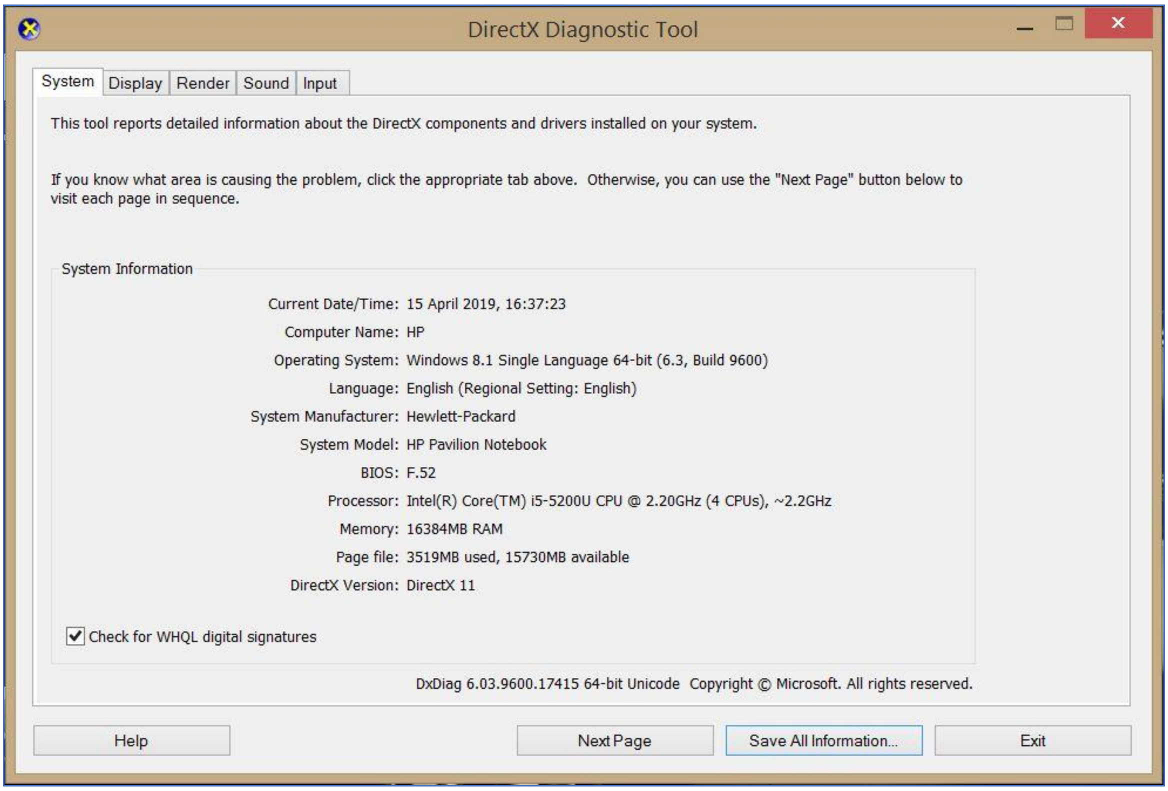


Figure 1: System configuration

1.2 Graphics and Display System Information

The table 2 below shows the Graphical and display information of the device used. This consists data of the high processing graphical hardware and display configurations used for this research project.

Table 2: Graphic and display system information

Item	Value
Card name	NVIDIA GeForce 940M
Manufacturer	NVIDIA
Chip type	GeForce 940M
DAC type	Integrated RAMDAC
Device Type	Render-Only Device
Device Key	Enum\PCI\VEN_10DE&DEV_1347&SUBSYS_8096103C&REV_A2
Display Memory	4020 MB
Dedicated Memory	1972 MB
Shared Memory	2048 MB
Current Mode	n/a

Driver Name	nvd3dumx,nvwgf2umx,nvwgf2umx,nvd3dum,nvwgf2um,nvwgf2um
Driver File Version	9.18.0013.4726 (English)
Driver Version	9.18.13.4726
DDI Version	11
Feature Levels	11.0,10.1,10.0,9.3,9.2,9.1
Driver Model	WDDM 1.3
Graphics Preemption	DMA
Compute Preemption	DMA
Miracast	Not Supported by Graphics driver
Hybrid Graphics GPU	Discrete
Power P-states	Not Supported
Driver Attributes	Final Retail
Driver Date/Size	2/9/2015 10: 22: 09, 17251976 bytes
WHQL Logo'd	Yes
Vendor ID	0x10DE
Device ID	0x1347
SubSys ID	0x8096103C
Revision ID	0x00A2
Driver Strong Name	oem8.inf
0f066de38b100cee	Section037
9.18.13.4726	pci\ven_10de&dev_1347&subsys_8096103c
Rank Of Driver	00DA0001
DDraw Status	Enabled
D3D Status	Enabled
AGP Status	Enabled

The figure 2 below depicts the graphical and display system configuration mentioned in the above table 2.

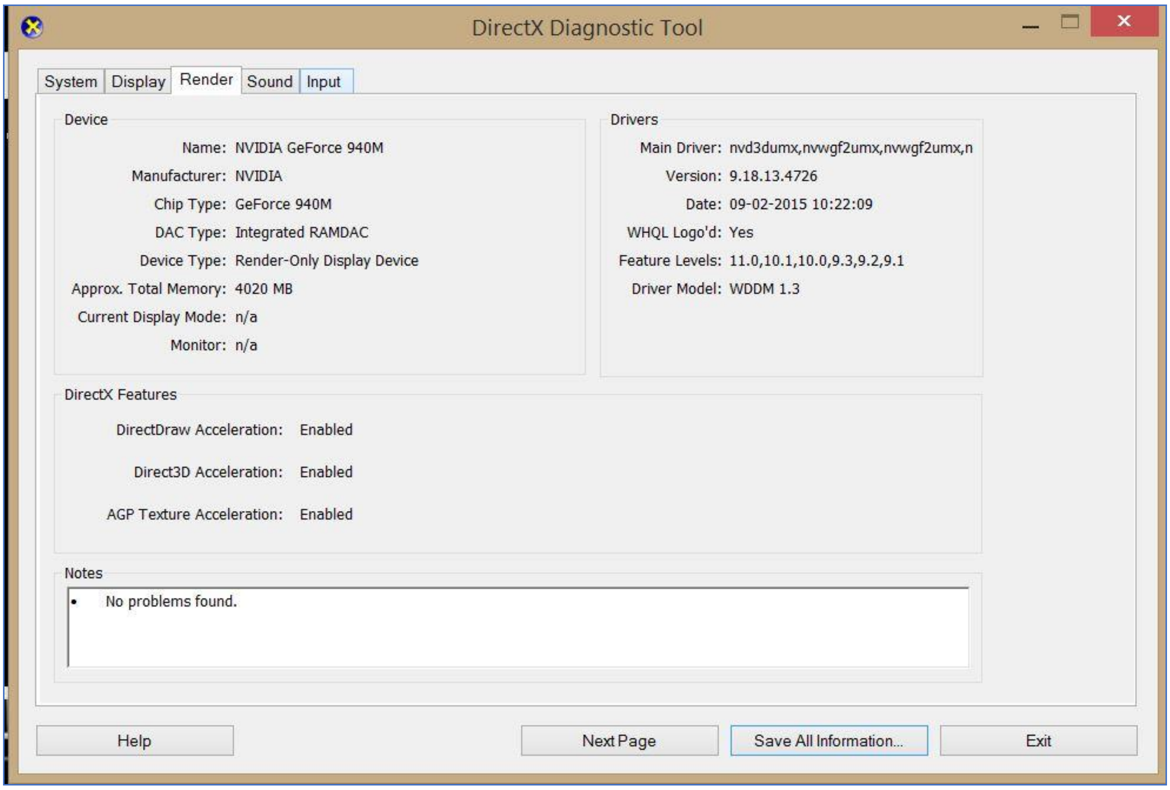


Figure 2: Display and graphic system information

2 Environment Setup

This section consists of the details of environment used for implementation of this research project. The figure 3 below shows from where we started our project, first we downloaded anaconda navigator for setting up the environment.

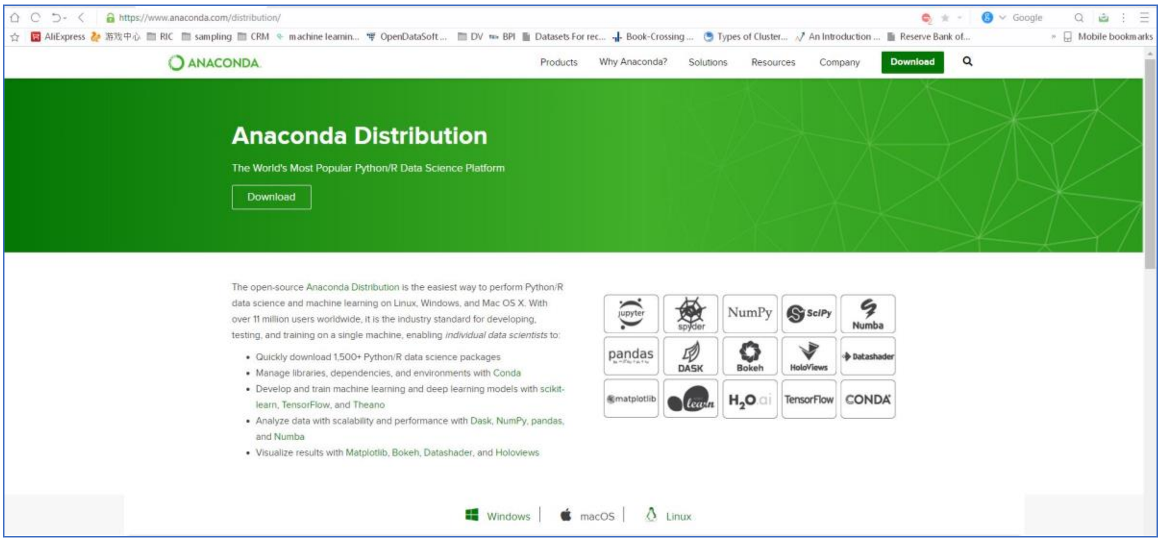


Figure 3: Anaconda setup download page

The figure 4 elaborates that Jupyter notebook from anaconda navigator was used throughout the research project.

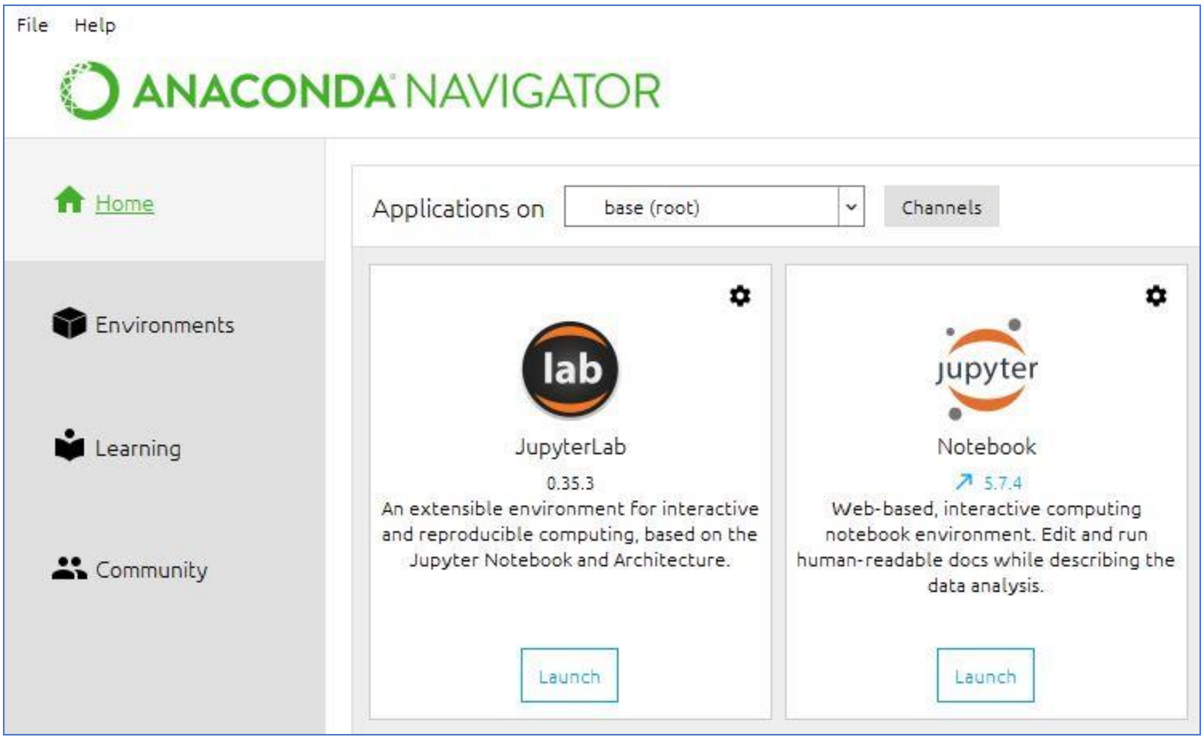


Figure 4: Jupyter notebook used from Anaconda

Figure 5 shows the code snippet used in Jupyter notebooks for this project.

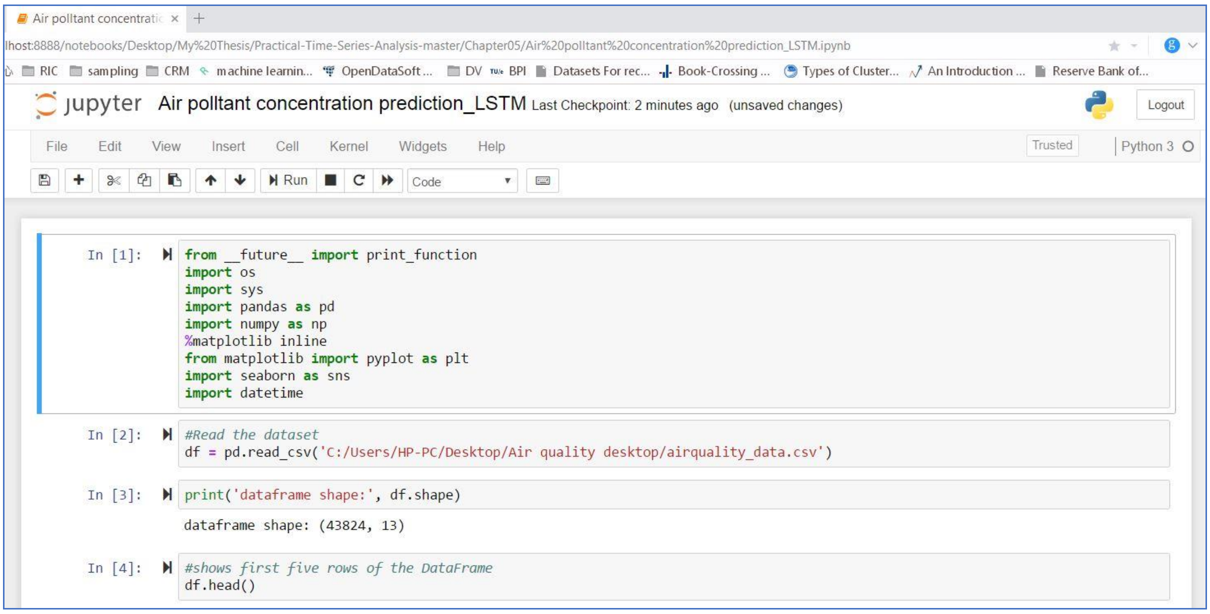


Figure 5: Example of code in Jupyter notebook

3 Python Packages and Libraries Used

This section lists all the libraries used in python for the implementation of this research project in brief. Table 3 lists all the python libraries used in detail.

Table 3: Python libraries in detail

Python Libraries	Description	Version
Pandas	Data analysis and high performance data structures	**0.24.1**
NumPy	Used for processing arrays for number, objects, strings	**1.15.4**
MatplotLib	Used for publishing quality plots and graphs	**3.0.2**
Scikit-learn	Used for data mining and machine learning models	**0.20.2**
Seaborn	Library for statistical data visualization	**0.9.0**
Keras	Deep learning library for tensorflow	**2.2.4**
statsmodels	Library used for statistical computations and model	**0.9.0**

The figure 6 below shows how we access, install and updates libraries for python in anaconda environment.

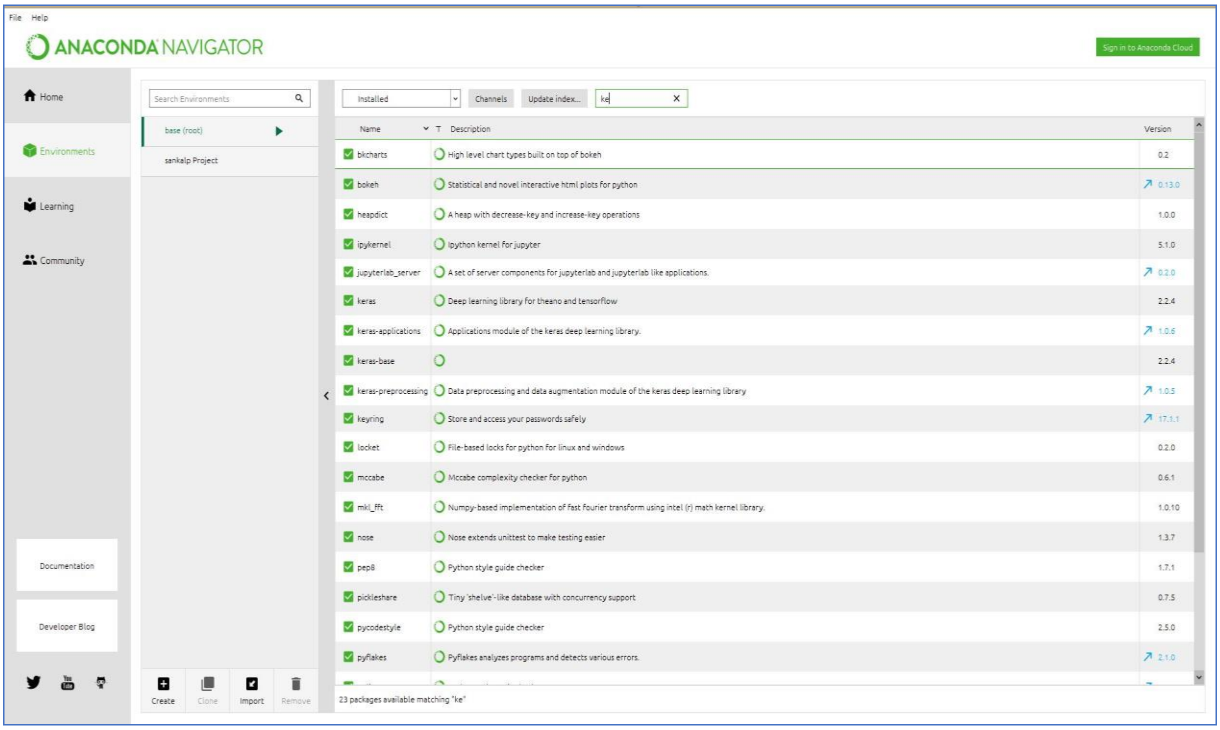
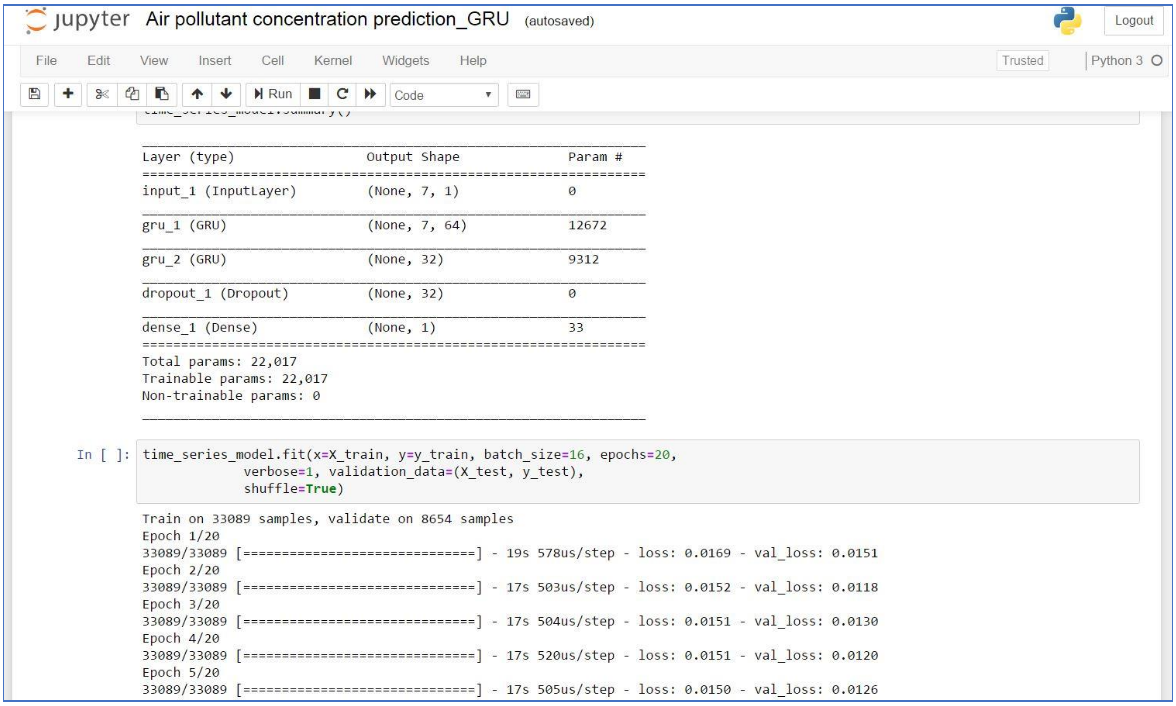


Figure 6: Python libraries in anaconda navigator

3.1 Python Code Execution with Anaconda Juypter Notebook

The figure 7(a), 7(b) and 7(c) shows the example of code which has been implemented for this research project using Jupyter notebook.



jupyter Air pollutant concentration prediction_GRU (autosaved)

Logout

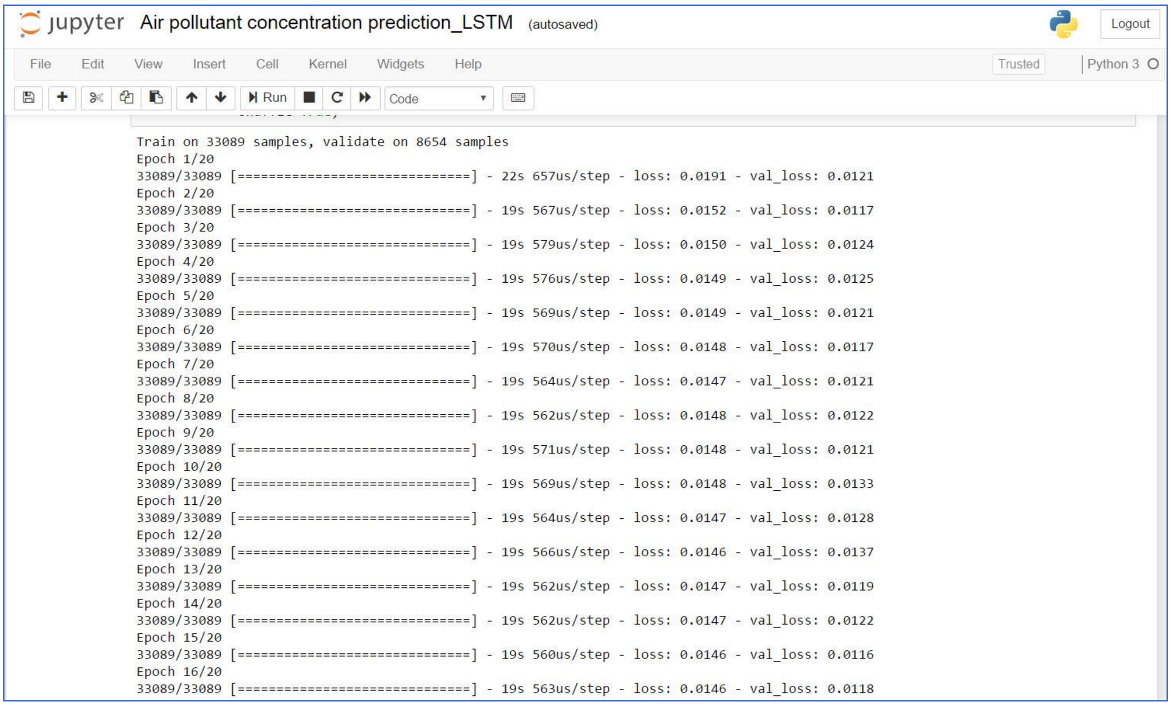
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 7, 1)              0
_________________________________________________________________
gru_1 (GRU)                  (None, 7, 64)             12672
_________________________________________________________________
gru_2 (GRU)                  (None, 32)                9312
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 22,017
Trainable params: 22,017
Non-trainable params: 0
_________________________________________________________________

In [ ]: time_series_model.fit(x=X_train, y=y_train, batch_size=16, epochs=20,
                      verbose=1, validation_data=(X_test, y_test),
                      shuffle=True)

Train on 33089 samples, validate on 8654 samples
Epoch 1/20
33089/33089 [==============================] - 19s 578us/step - loss: 0.0169 - val_loss: 0.0151
Epoch 2/20
33089/33089 [==============================] - 17s 503us/step - loss: 0.0152 - val_loss: 0.0118
Epoch 3/20
33089/33089 [==============================] - 17s 504us/step - loss: 0.0151 - val_loss: 0.0130
Epoch 4/20
33089/33089 [==============================] - 17s 520us/step - loss: 0.0151 - val_loss: 0.0120
Epoch 5/20
33089/33089 [==============================] - 17s 505us/step - loss: 0.0150 - val_loss: 0.0126
```

Figure 7(a): Implementation of code in Jupyter notebook



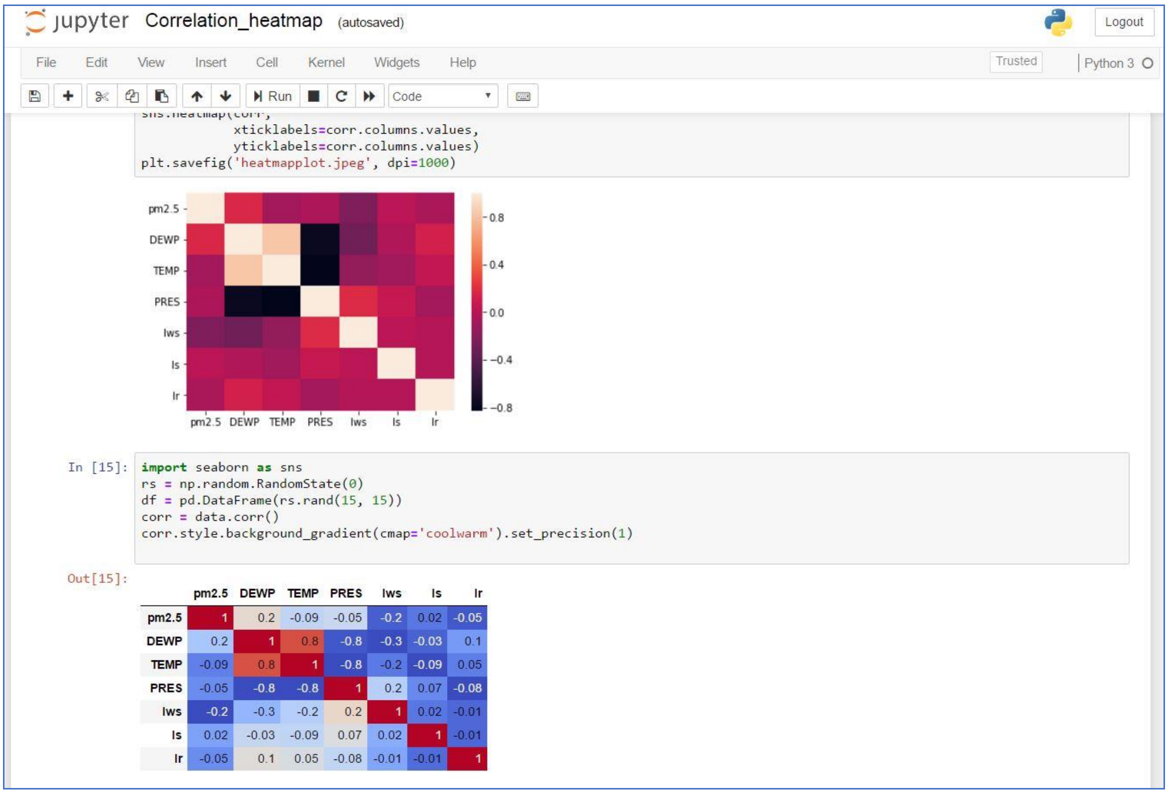
jupyter Air pollutant concentration prediction_LSTM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
Train on 33089 samples, validate on 8654 samples
Epoch 1/20
33089/33089 [==============================] - 22s 657us/step - loss: 0.0191 - val_loss: 0.0121
Epoch 2/20
33089/33089 [==============================] - 19s 567us/step - loss: 0.0152 - val_loss: 0.0117
Epoch 3/20
33089/33089 [==============================] - 19s 579us/step - loss: 0.0150 - val_loss: 0.0124
Epoch 4/20
33089/33089 [==============================] - 19s 576us/step - loss: 0.0149 - val_loss: 0.0125
Epoch 5/20
33089/33089 [==============================] - 19s 569us/step - loss: 0.0149 - val_loss: 0.0121
Epoch 6/20
33089/33089 [==============================] - 19s 570us/step - loss: 0.0148 - val_loss: 0.0117
Epoch 7/20
33089/33089 [==============================] - 19s 564us/step - loss: 0.0147 - val_loss: 0.0121
Epoch 8/20
33089/33089 [==============================] - 19s 562us/step - loss: 0.0148 - val_loss: 0.0122
Epoch 9/20
33089/33089 [==============================] - 19s 571us/step - loss: 0.0148 - val_loss: 0.0121
Epoch 10/20
33089/33089 [==============================] - 19s 569us/step - loss: 0.0148 - val_loss: 0.0133
Epoch 11/20
33089/33089 [==============================] - 19s 564us/step - loss: 0.0147 - val_loss: 0.0128
Epoch 12/20
33089/33089 [==============================] - 19s 566us/step - loss: 0.0146 - val_loss: 0.0137
Epoch 13/20
33089/33089 [==============================] - 19s 562us/step - loss: 0.0147 - val_loss: 0.0119
Epoch 14/20
33089/33089 [==============================] - 19s 562us/step - loss: 0.0147 - val_loss: 0.0122
Epoch 15/20
33089/33089 [==============================] - 19s 560us/step - loss: 0.0146 - val_loss: 0.0116
Epoch 16/20
33089/33089 [==============================] - 19s 563us/step - loss: 0.0146 - val_loss: 0.0118
```

Figure 7(b): Implementation of code in Jupyter notebook



	pm2.5	DEWP	TEMP	PRES	Iws	Is	Ir
pm2.5	1	0.2	-0.09	-0.05	-0.2	0.02	-0.05
DEWP	0.2	1	0.8	-0.8	-0.3	-0.03	0.1
TEMP	-0.09	0.8	1	-0.8	-0.2	-0.09	0.05
PRES	-0.05	-0.8	-0.8	1	0.2	0.07	-0.08
Iws	-0.2	-0.3	-0.2	0.2	1	0.02	-0.01
Is	0.02	-0.03	-0.09	0.07	0.02	1	-0.01
Ir	-0.05	0.1	0.05	-0.08	-0.01	-0.01	1

Figure 7(c): Implementation of code in Jupyter notebook

4 Software Used

4.1 Tableau and Excel Used for Data Visualization

Tableau and excel were used for visualizing the air pollution data for this research project. Figure 8 shows how tableau is used for finding the hidden patterns in PM2.5 concentrations for this project.

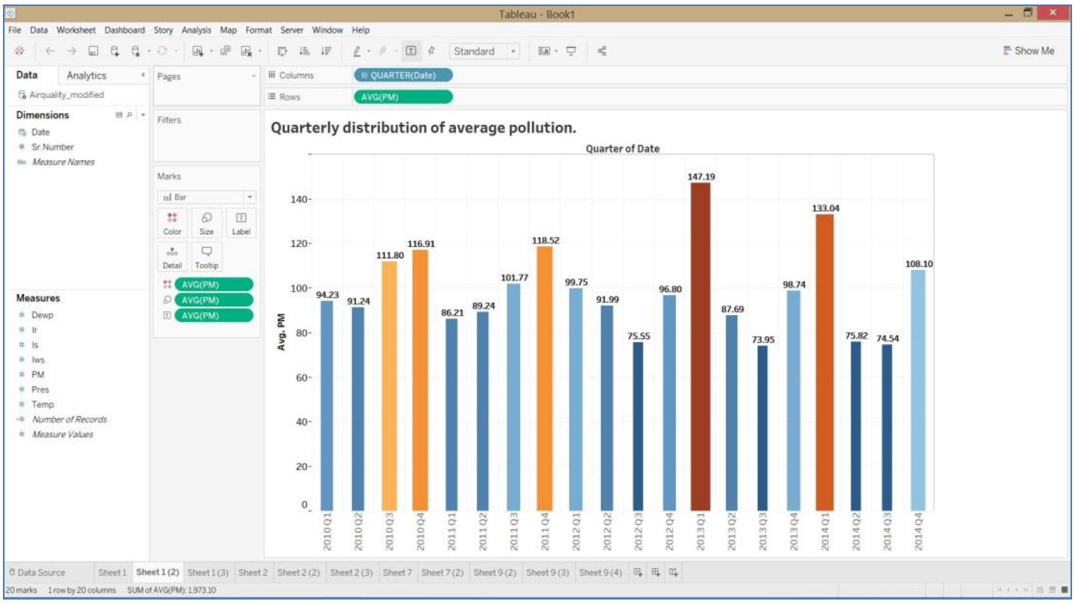


Figure 8: Tableau used for visualization of PM2.5 concentration

The figure 9 shows how Excel was used for comparing the mean absolute error of the implemented models for this research.

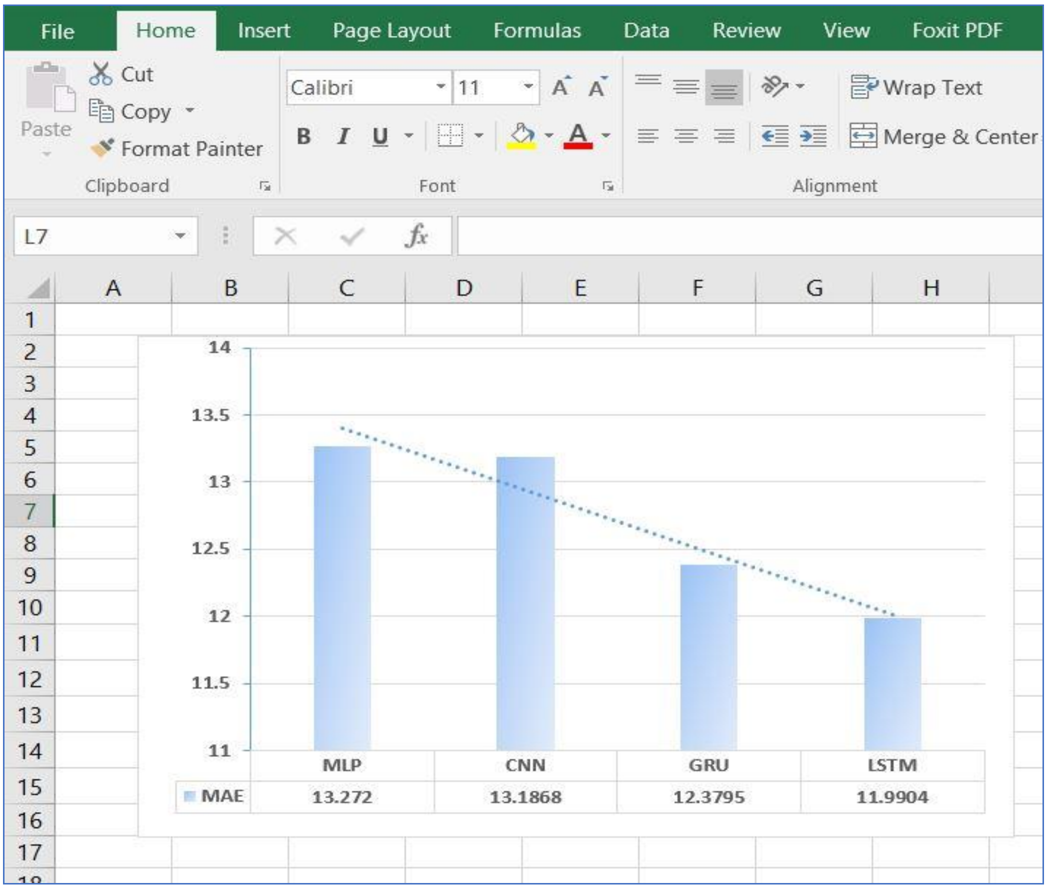


Figure 9: Excel used for comparing MAE