National College of Ireland

Dynamic Virtual Machine Migration using Ensemble Regressor based controller to reduce the Green energy wastage and Optimal Utilization of Resources towards Green Computing - Configuration Manual

MSc Research Project
Cloud Computing

Lalit Rakesh Agrawal
Student ID: 19226331

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Lalit Rakesh Agrawal
Student ID:	19226331
Programme:	Cloud Computing **Year:** 2022
Module:	MSc Research Project
Lecturer:	Sean Heeney
Submission Due Date:	16/09/2022
Project Title:	Dynamic Virtual Machine Migration using Ensemble Regressor based controller to reduce the Green energy wastage and Optimal Utilization of Resources towards Green Computing.
Word Count:	900 **Page Count:** 10

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: Lalit Rakesh Agrawal

Date: 15/08/2022

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):	

Dynamic Virtual Machine Migration using Ensemble Regressor based controller to reduce the Green energy wastage and Optimal Utilization of Resources towards Green Computing - Configuration Manual

Lalit Rakesh Agrawal
19226331

1 Introduction

The Configuration manual outlines various steps that need to be performed by the user in order to successfully implement the project. The manual focuses on multiple aspects such as the System Configuration and libraries/ modules used in the project. Moreover, this manual includes step-by-step instructions to install all the software and libraries used and also focuses on commands needed to spawn key components of the architecture i.e. broker, cluster, and provisioner. The procedure of data generation and running notebooks for evaluation purposes are also covered in this manual.

2 System Configuration

2.1 Hardware Specification

• Processor: 11th Gen Intel(R) Core (TM) i5-11300H @ 3.10GHz, 2611 Mhz, 4 Core (s)
• Operating System: Ubuntu 20.04 (Linux) or above
• RAM: 8.00 GB
• Storage (SSD): 50 GB

3 Software Installation

3.1 Python

Implementation is been done using python as a coding language, Python is utilized so that the implementation can be carried out Downloads of the Python software are available at the following website: https://www.python.org/downloads/. Python 3.8 is the minimum required version.

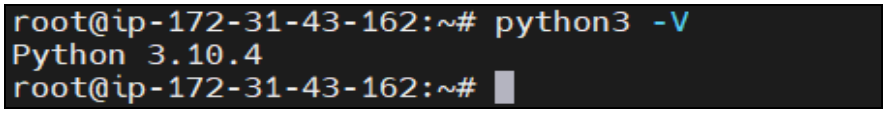

Figure 1: Python Version

3.2 Pip Installation

Pip stands for "Pip Installs Packages" which is a package manager. Python packages and libraries are installed using the "pip" command. Pip can be installed using the following command on ubuntu "**sudo apt install python3-pip**"

```
root@ip-172-31-45-67:~# pip --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

Figure 2: Pip Version

```
root@ip-172-31-45-67:~# apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools
  libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1
  liblsan0 libmpc3 libnsl-dev libpython3-dev libpython3.10 libpython3.10-dev libpython3.10-minimal
  libtirpc-dev libtirpc3 libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list make
  rpcsvc-proto zlib1g-dev
```

Figure 3: Pip installation

3.3 Python Libraries

Multiple Python libraries are been used in order to implement the AI Controller. Following is the list of Libraries that are been used.

- TensorFlow
- sklearn
- pandas
- matplotlib
- numpy

3.4 TensorFlow Installation

Tensor Flow version - v2.9

TensorFlow is a library used for machine learning and deep learning that facilitates the development of large-scale neural networks [1]. Ubuntu users can run "**pip install -U tensorflow"** to install and update the library.

```
root@ip-172-31-45-67:~# pip install -U tensorflow
Collecting tensorflow
  Downloading tensorflow-2.9.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (511.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 511.7/511.7 MB 32.2 MB/s eta 0:00:01Killed
root@ip-172-31-45-67:~#
```

Figure 4: TensorFlow Installation

3.5 scikit-learn/sklearn Installation

Sckit-Learn Version - v1.1.2

Models for machine learning can be constructed with the help of the scikit library [2]. Ubuntu users can run "**pip install -U sklearn"** to install and update the library.

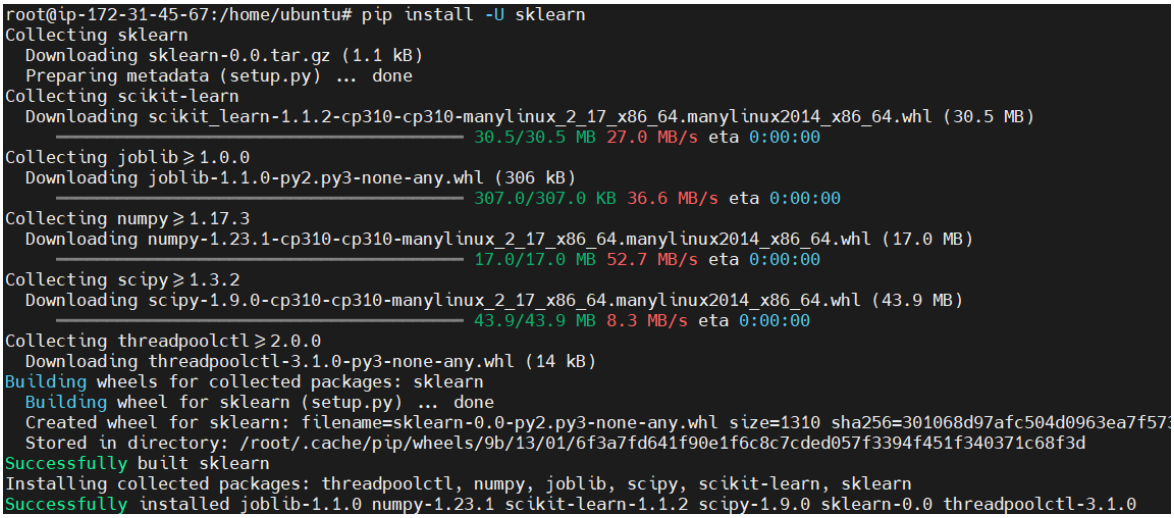

Figure 5: scikit-learn Installation

3.6 Pandas Installation

Pandas - v1.4.3

The Pandas library supports the import of data from a variety of file formats, including CSV, JSON, SQL, and Microsoft Excel, among others. Additionally, it enables operations such as data merging, cleaning, and other manipulations [3]. Ubuntu users can run "**pip install pandas"** to install pandas.

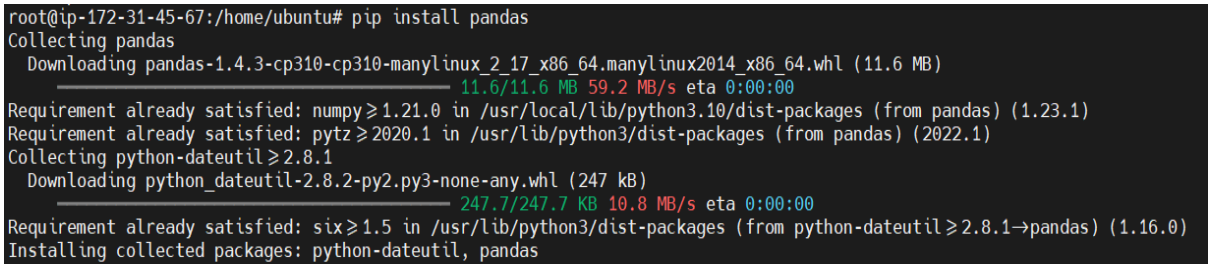

Figure 6: Pandas Installation

3.7 matplotlib Installation

matplotlib - v3.5.3

The matplotlib is used to create a visualization of the data generated into a graphical format [4]. This visualization is then used for evaluation purposes. Ubuntu users can run "**pip install matplotlib"** for installation.

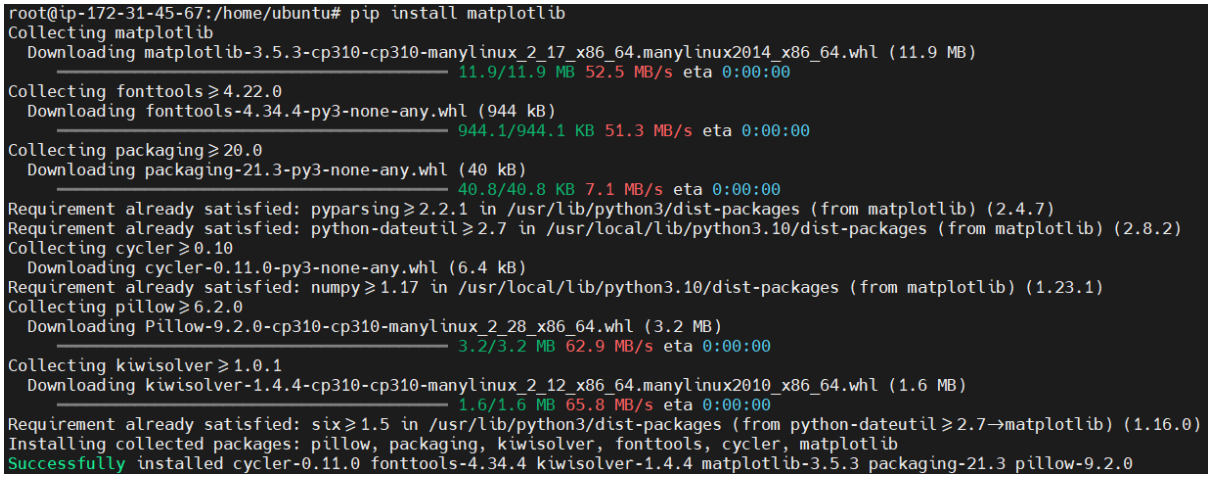

Figure 7: matplotlib Installation

3.8 Juypter Notebook Installation

Jupyter Notebook is the most recent iteration of an interactive development environment that is web-based and designed for notebooks, code, and data. Because of its adaptable user interface, users are able to configure and organize workflows in a variety of fields, including data science, scientific computing, computational journalism, and machine learning [5]. Ubuntu users can run "**sudo apt-get -y install jupyter"** for installation.

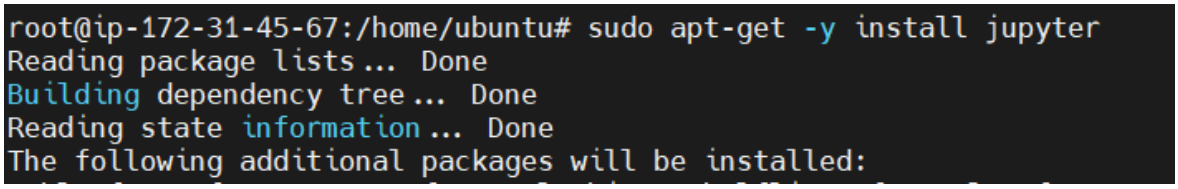

Figure 8: Juypter Notebook Installation

4 Implementation

4.1 Data Generation

The data were generated taking into account the user cycle and the energy cycle, both of which are periodic and have an element of randomness. As a result, a sin wave function was used to generate the dataset, with the load and energy generation increasing and decreasing over time. The Generated data is stored in a PKL file which is later used by the Machine Learning model to train the data. The user should run the following command to data generation "python3 data_gen.py". The user cycle data and the energy cycle data will be generated at "./basic/user_data.pkl" and "./basic/energy_data.pkl" locations respectively.

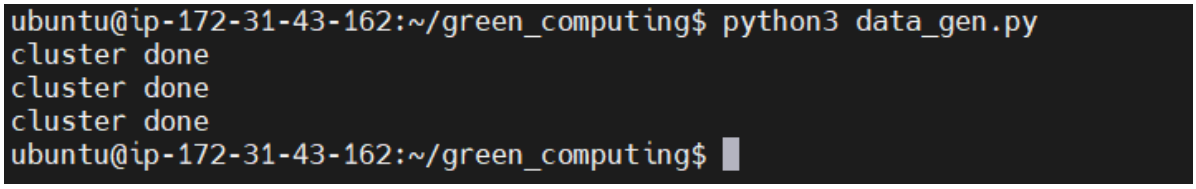

Figure 9: Data Generation Output

4.2 Model Training and Prediction.

Once all the libraries and software users can implement the project on the system. To implement the project user needs to run all the algorithms used in the project i.e Linear regressor, Support vector regressor, Bayesian regression, and Ensemble regressor using the following commands. These are the following commands: -

- python3 lr_demo.py > lr_demo.txt
- python3 svm_demo.py > svm_demo.txt
- python3 bay_demo.py > bay_demo.txt
- python3 em_demo.py > em_demo.txt

```
ubuntu@ip-172-31-43-162:~/green_computing$ python3 lr_demo.py > lr_demo.txt
ubuntu@ip-172-31-43-162:~/green_computing$ cat lr_demo.txt
(985,)
(985,)
[ 6.08865804  7.81642744  5.00125057 10.21750544  7.44408701]
(985,)
(985,)
[6.02432279 4.83571498 4.95816987 7.81020121 3.11631053]
(985,)
(985,)
[6.13725579 7.77114197 4.06542949 6.19450373 5.56961982]
0 on cluster_1 processed 1.0638294199067728
1 on cluster_1 processed 1.0638294199067728
2 on cluster_1 processed 1.0638294199067728
3 on cluster_1 processed 1.0638294199067728
4 on cluster_1 processed 1.0638294199067728
5 on cluster_1 processed 1.0638294199067728
6 on cluster_1 processed 1.0638294199067728
7 on cluster_1 processed 1.0638294199067728
8 on cluster_1 processed 1.0638294199067728
9 on cluster_1 processed 1.0638294199067728
10 on cluster_1 processed 1.0638294199067728
11 on cluster_1 processed 1.0638294199067728
```

Figure 10: Logical Regression Model Traning and Prediction output.

```
ubuntu@ip-172-31-43-162:~/green_computing$ python3 svm_demo.py > svm_demo.txt
ubuntu@ip-172-31-43-162:~/green_computing$ cat svm_demo.txt
(985,)
(985,)
[7.22959749 5.11266266 5.72898506 1.21350966 0.05639016]
(985,)
(985,)
[7.50657954 6.68668026 6.66611966 3.40321593 4.01951968]
(985,)
(985,)
[1.98882334 3.84697726 0.43873097 3.65729988 4.69640667]
0 on cluster_1 processed 1.0638294199067728
1 on cluster_1 processed 1.0638294199067728
2 on cluster_1 processed 1.0638294199067728
3 on cluster_1 processed 1.0638294199067728
4 on cluster_1 processed 1.0638294199067728
5 on cluster_1 processed 1.0638294199067728
6 on cluster_1 processed 1.0638294199067728
7 on cluster_1 processed 1.0638294199067728
8 on cluster_1 processed 1.0638294199067728
9 on cluster_1 processed 1.0638294199067728
10 on cluster_1 processed 1.0638294199067728
```

Figure 11: SVR Model Traning and Prediction output.

```
ubuntu@ip-172-31-43-162:~/green_computing$ python3 bay_demo.py > bay_demo.txt
ubuntu@ip-172-31-43-162:~/green_computing$ cat bay_demo.txt
(985,)
(985,)
[5.90256053 7.59456263 4.75894555 9.94017157 6.88924431]
(985,)
(985,)
[5.81910949 4.64260802 4.72713088 7.52747563 2.75666391]
(985,)
(985,)
[5.65700115 7.02549002 3.62178414 7.16636733 6.13942056]
0 on cluster_1 processed 1.0638294199067728
1 on cluster_1 processed 1.0638294199067728
2 on cluster_1 processed 1.0638294199067728
3 on cluster_1 processed 1.0638294199067728
4 on cluster_1 processed 1.0638294199067728
5 on cluster_1 processed 1.0638294199067728
6 on cluster_1 processed 1.0638294199067728
7 on cluster_1 processed 1.0638294199067728
8 on cluster_1 processed 1.0638294199067728
9 on cluster_1 processed 1.0638294199067728
10 on cluster_1 processed 1.0638294199067728
```

Figure 12: Bayesian Model Traning and Prediction output.

```
ubuntu@ip-172-31-43-162:~/green_computing$ python3 em_demo.py > em_demo.txt
ubuntu@ip-172-31-43-162:~/green_computing$ cat em_demo.txt
(985,)
(985,)
[1.80029216 3.55034161 1.26926736 7.00970239 4.12613822]
(985,)
(985,)
[1.6824564  1.35529129 1.48142686 4.79484802 0.73188866]
(985,)
(985,)
[4.07515695 5.33109798 2.38962851 5.00338259 5.03763746]
0 on cluster_1 processed 1.0638294199067728
1 on cluster_1 processed 1.0638294199067728
2 on cluster_1 processed 1.0638294199067728
3 on cluster_1 processed 1.0638294199067728
4 on cluster_1 processed 1.0638294199067728
5 on cluster_1 processed 1.0638294199067728
6 on cluster_1 processed 1.0638294199067728
7 on cluster_1 processed 1.0638294199067728
8 on cluster_1 processed 1.0638294199067728
9 on cluster_1 processed 1.0638294199067728
10 on cluster_1 processed 1.0638294199067728
```

Figure 13: Ensemble Model Traning and Prediction output.

Once all the algorithm files are executed. The prediction Accuracy graph can be plotted by running the following command "**python3 provisioner.py**". Users can see the prediction Accuracy graphs for all the models as shown below.

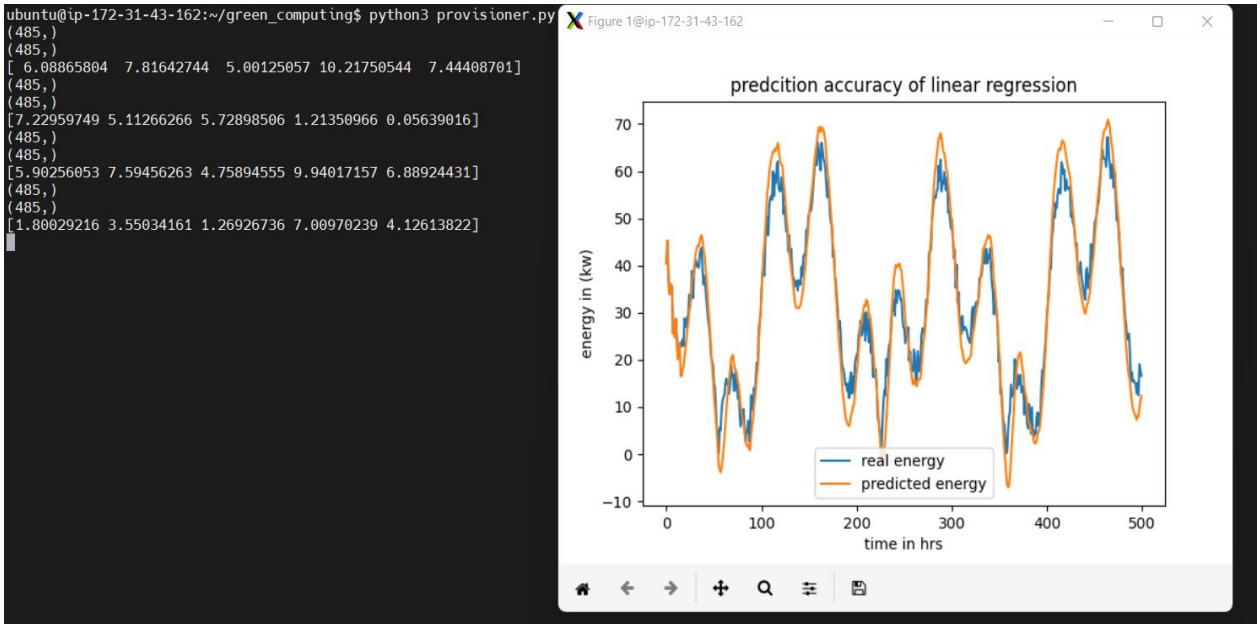

Figure 14: Prediction Accuracy for Linear Regression

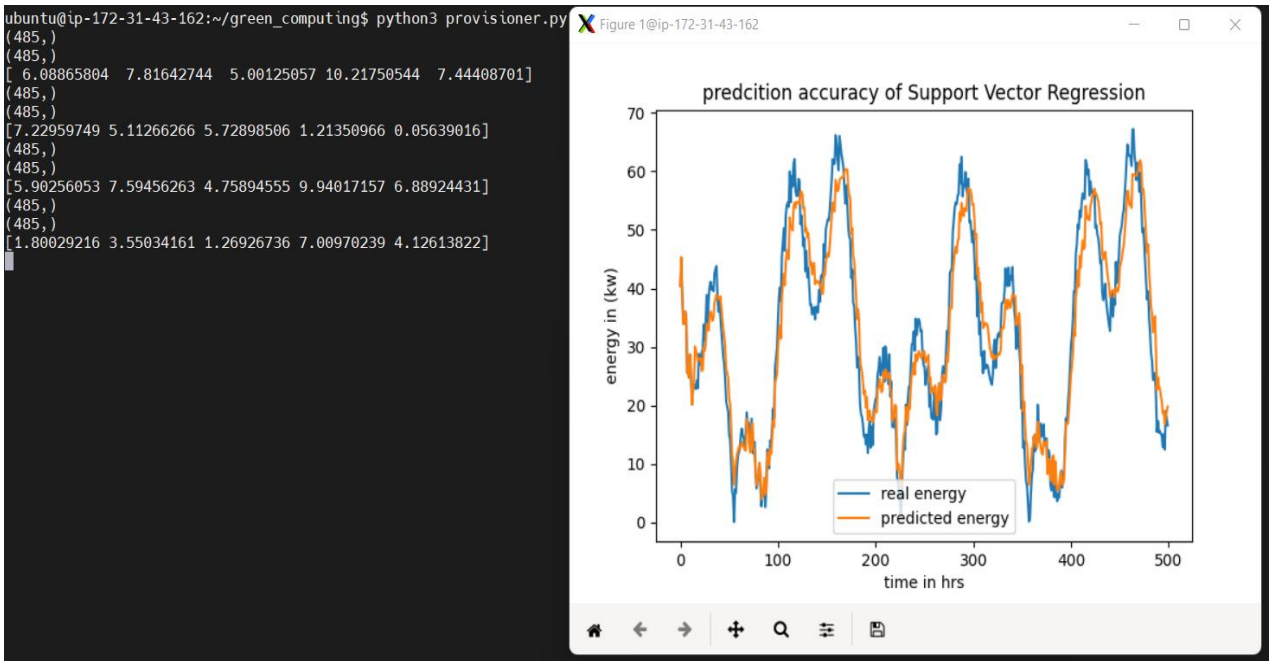

Figure 15: Prediction Accuracy for Support Vector Regression.

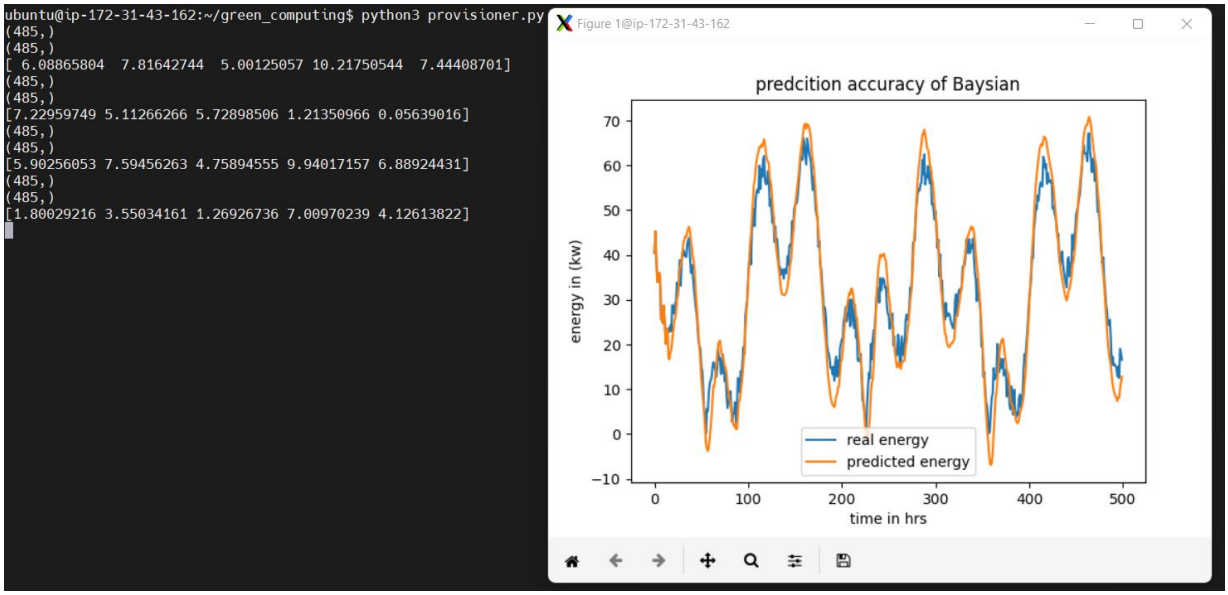

Figure 16: Prediction Accuracy for Bayesian.

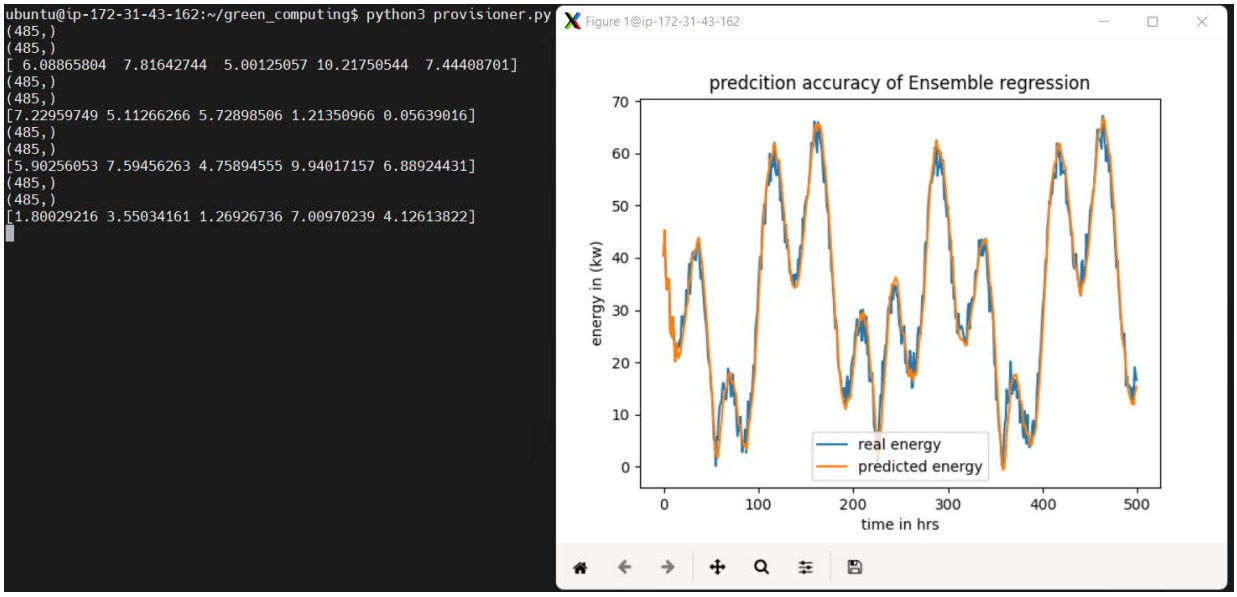

Figure 17: Prediction Accuracy for Ensemble.

From the above Prediction Accuracy graphs, it can be derived that the proposed Ensemble regression has a minimal between the predicted energy and actual energy.

4.3 Evaluation Based on Energy Wasted

For a comparative analysis, the total green energy wastage by each algorithm is combined. For which user need to run following jupyter notebooks.

- lr_demo_eda.ipynb
- svm_demo_eda.ipynb
- baysian_demo_eda.ipynb
- em_demo_eda.ipynb

Note: -User should Run all the cells of the notebooks mentioned above.

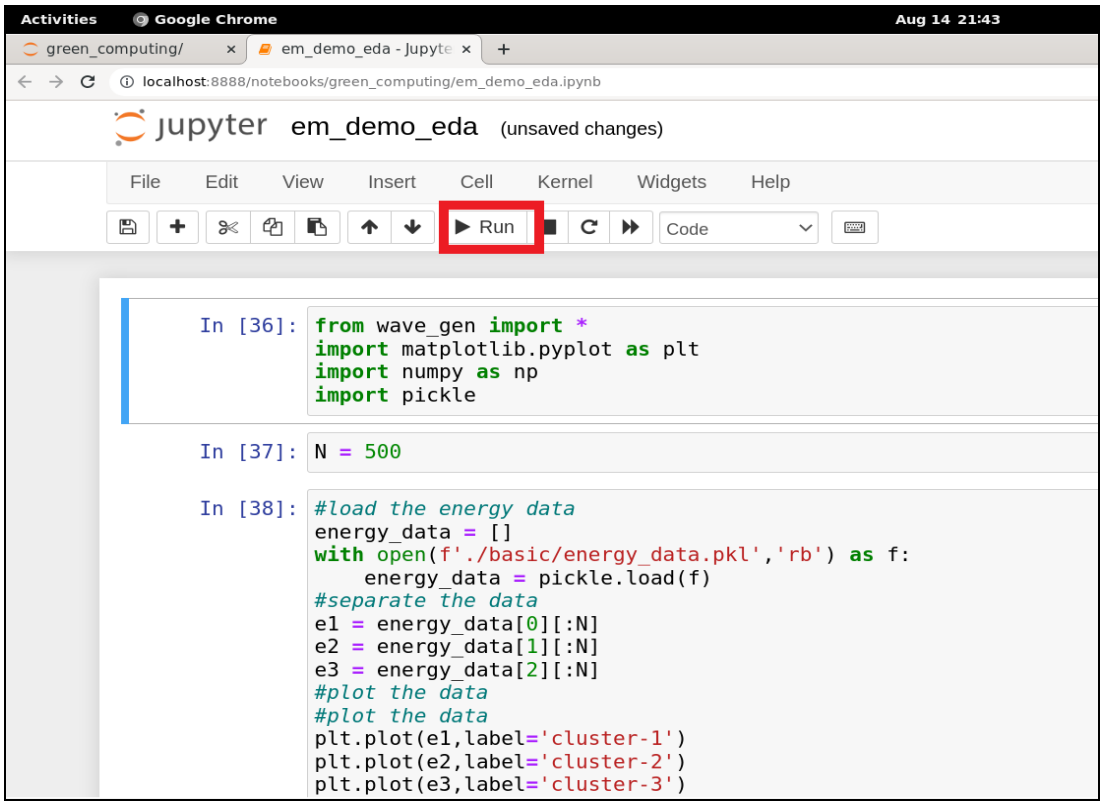

Figure 18: Execution of Ensemble Notebook

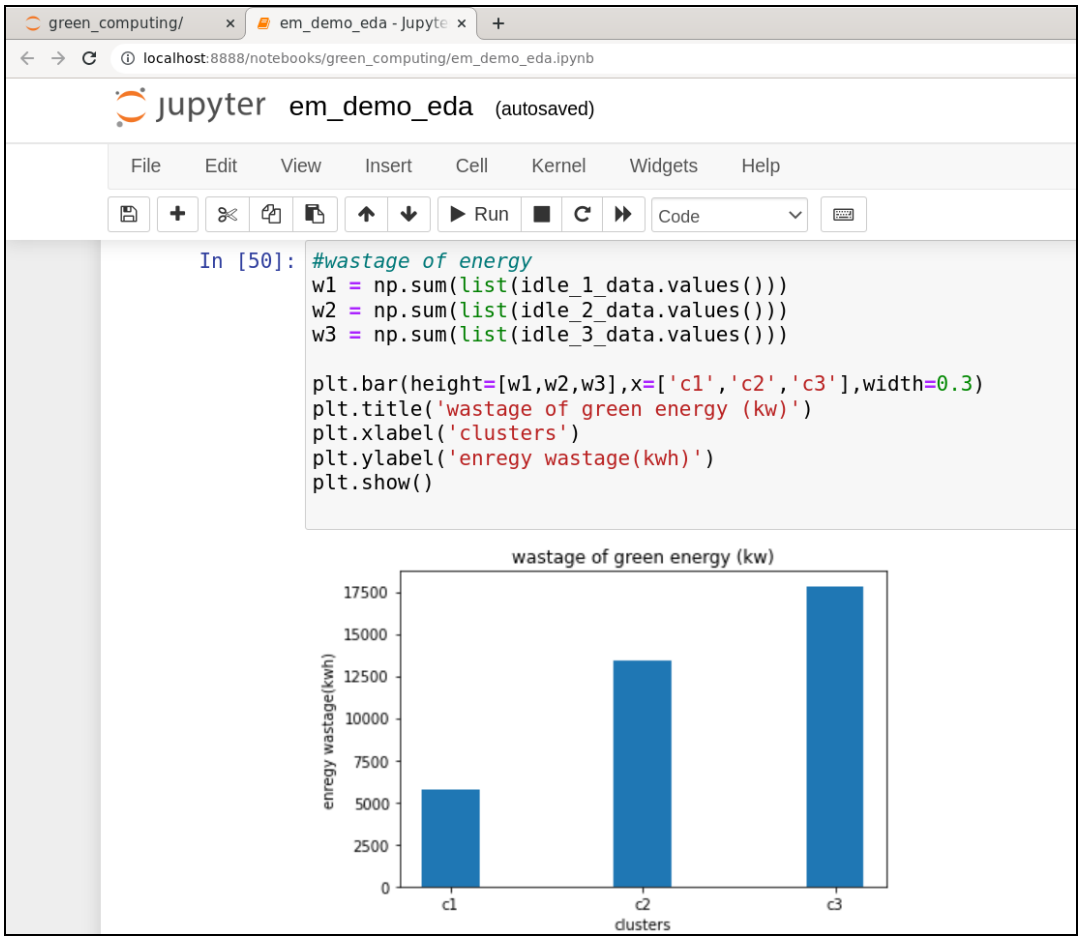

Figure 19: Wastage of Energy based on Ensemble Regressor

5 References

[1] TensorFlow, "TensorFlow," [Online]. Available: https://www.tensorflow.org/.
[2] scikit-learn, "scikit-learn," [Online]. Available: https://scikit-learn.org/stable/.
[3] pandas, "pandas," [Online]. Available: https://pandas.pydata.org/.
[4] matplotlib, "matplotlib," [Online]. Available: https://matplotlib.org/.
[5] jupyter, "jupyter," [Online]. Available: https://jupyter.org/.