National College of Ireland

Configuration Manual

MSc Research Project
MSc Cloud Computing

Bhavna Jasmine Maria Rathna Kumar

Student ID: 22185101

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Bhavna Jasmine Maria Rathna Kumar
Student ID:	22185101
Programme:	Masters in Cloud Computing **Year:** 2023
Module:	MSc Research project
Lecturer:	Rashid Mijumbi
Submission Due Date:	16-09-2024
Project Title:	Using Machine Learning in Edge Computing for Optimizing Resource Scheduling
Word Count:	315 **Page Count:** 7

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:	Bhavna Jasmine Maria
Date:	16-09-2024

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

Bhavna Jasmine Maria Rathna Kumar
Student ID: 22185101

1 Introduction

Below is the configuration manual for the research carried out under the topic "Using Machine Learning in Edge Computing for Optimizing Resource Scheduling". There are three areas covered i.e., Environmental Setup, Tools Setup and Information on the dataset used in this research.

2 Environmental Setup

Below is the Hardware Setup for finding the best model in the local environment
Processor: Intel i3 or above
Memory: 8 GB RAM
Below is the Programming Setup
Python Programming Version 3.10 and above

3 Tools Setup

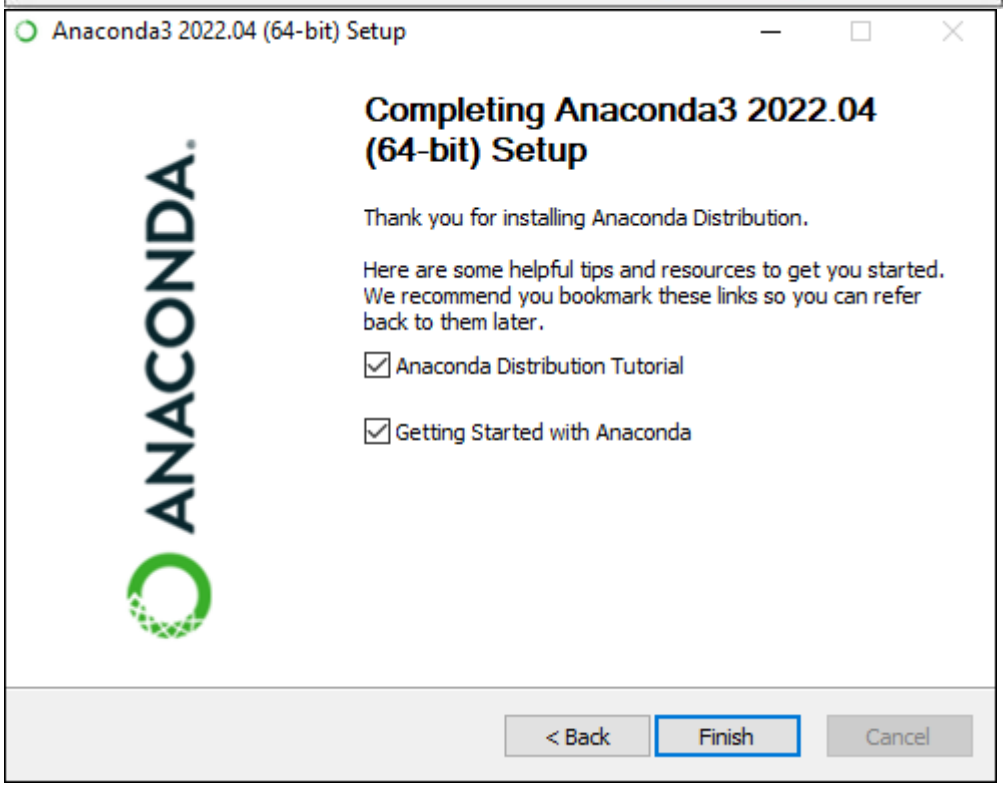
There are two tools required for this project and a dataset

- Anaconda Navigator: For creation of the predictive Model
- AWS Cloud – EC2 Instance
- Dataset (Link mentioned in section 3.3)

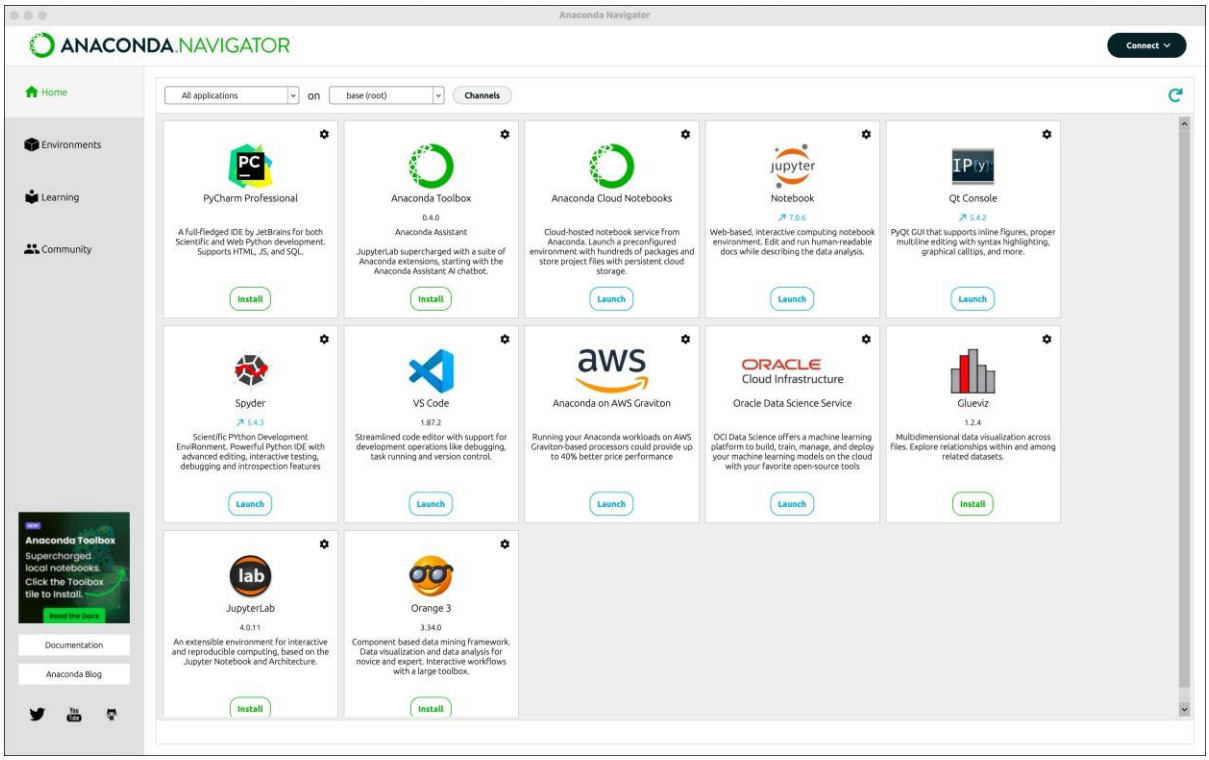
3.1 Installing Anaconda Navigator

Step 1: Download Anaconda Navigator from the below link
https://www.anaconda.com/download

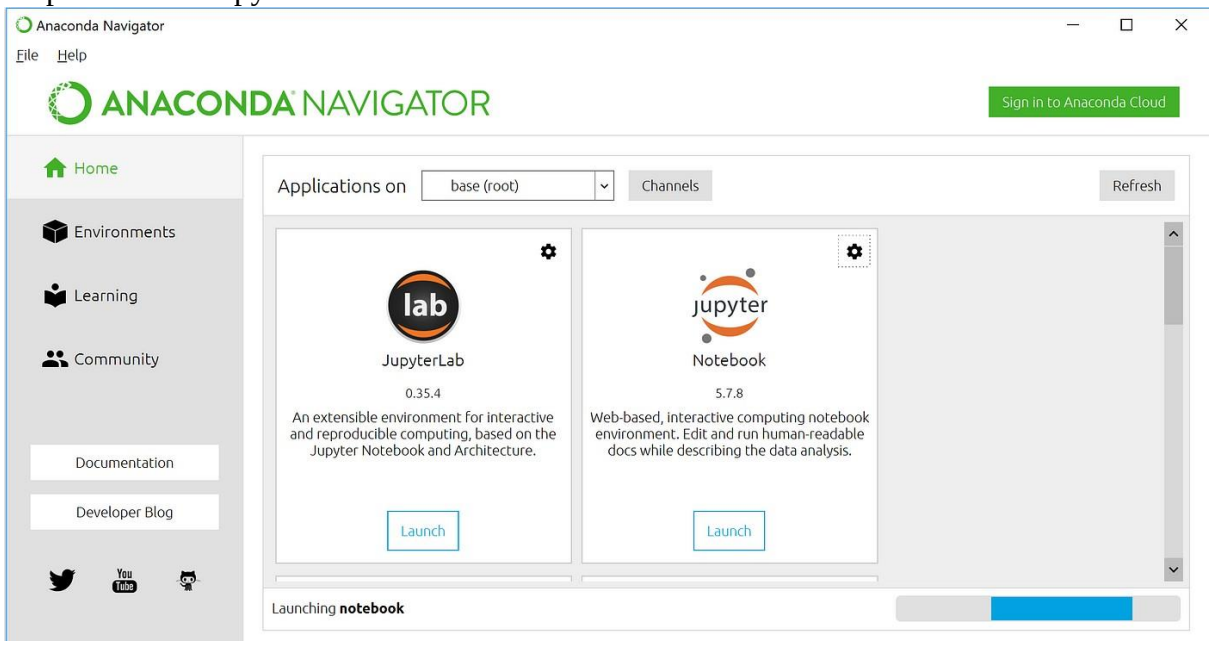
Step 2: Install Anaconda Navigator by following the below instructions



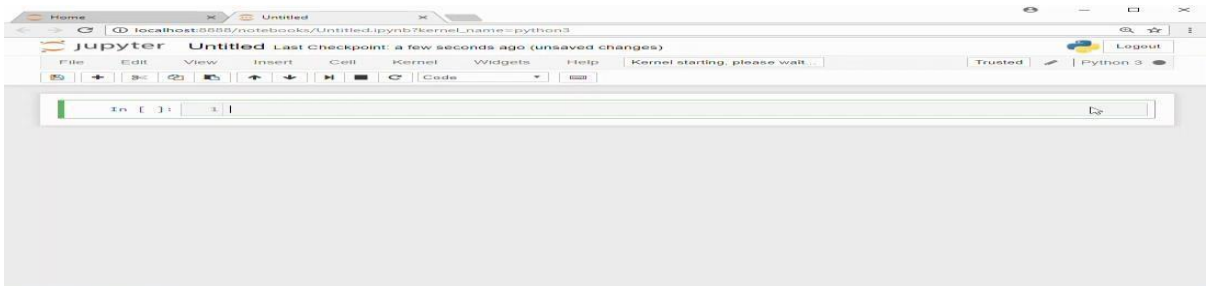
Step 3: Open Anaconda Navigator



Step 4: Launch Jupyter Notebook



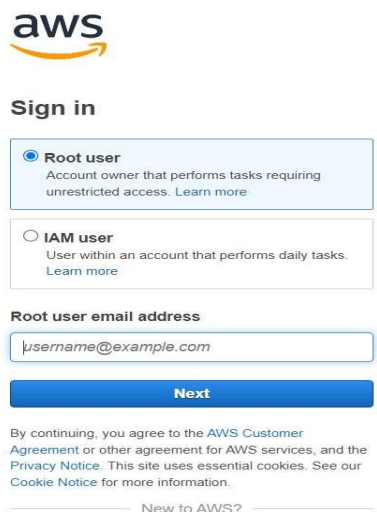
Step 5: Use any browser to launch Jupyter Notebook

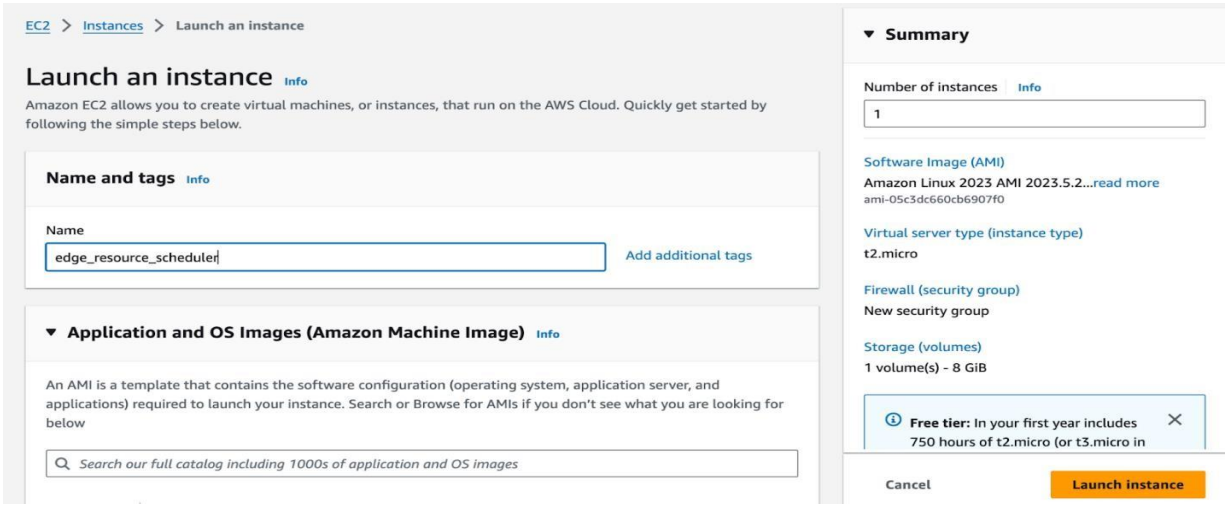


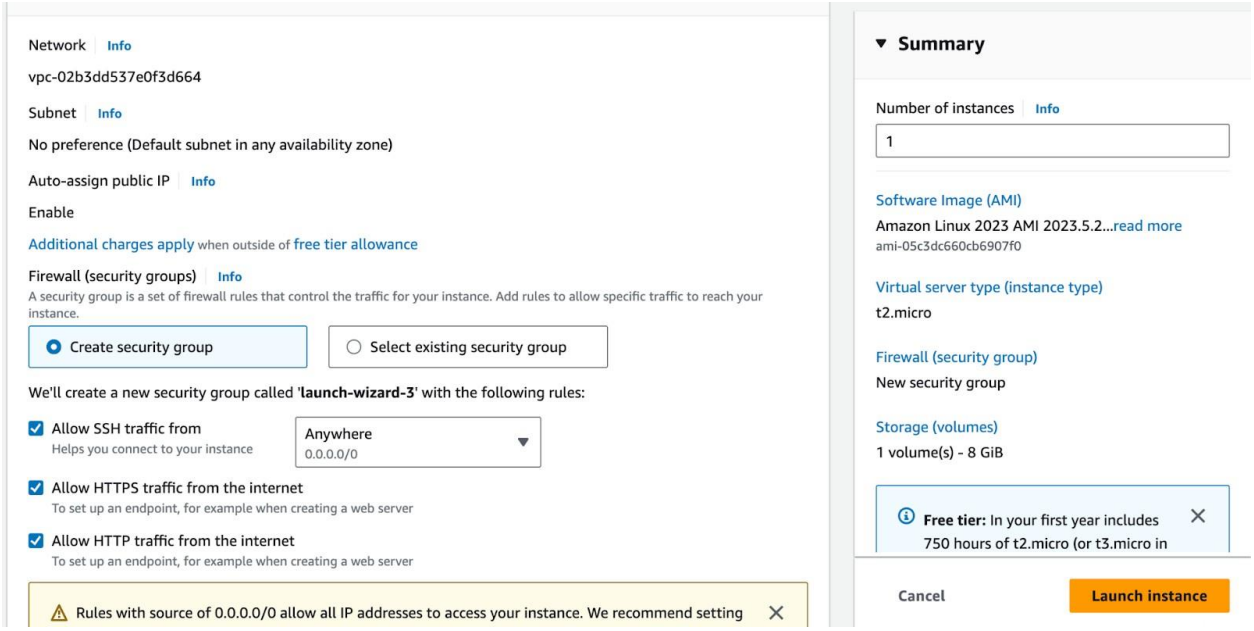
Step 6: Open the jupyter notebooks from the source code and run the cells one by one. Note first run the notebook named ‘1-Network Load Balancing Data Preparation.ipynb’ and then run ‘2-Building ML Models.ipynb’ and finally run ‘3-fusion models.ipynb’.

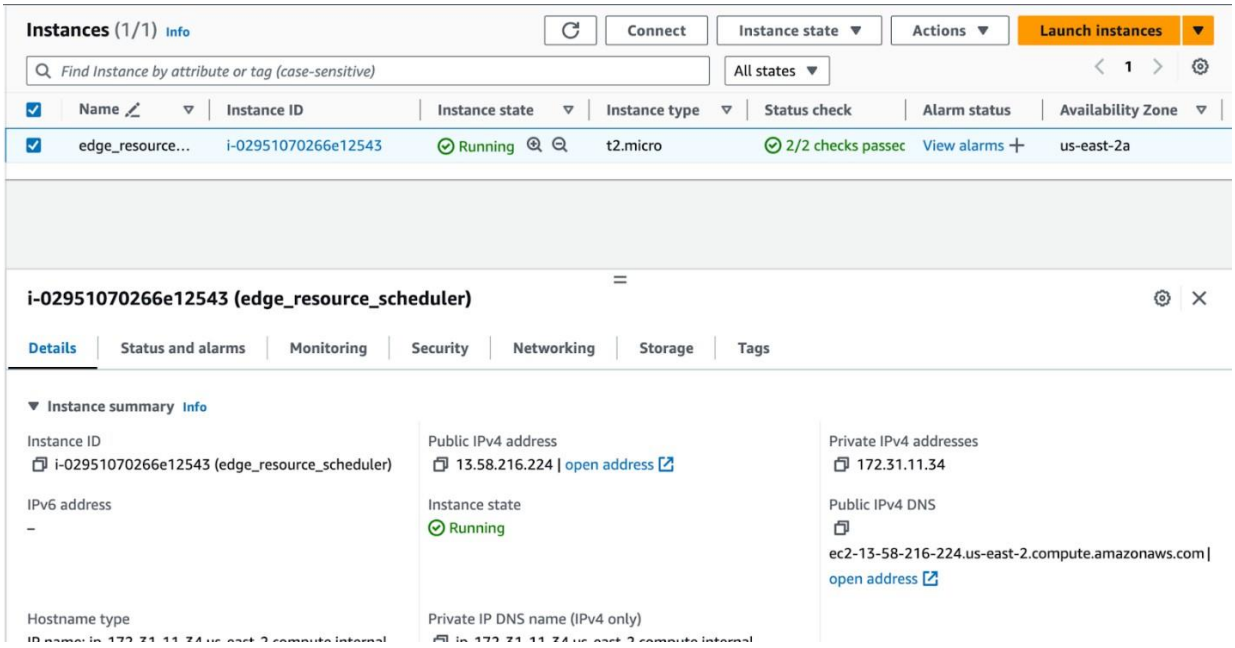
3.2 AWS Cloud – EC2 Instance

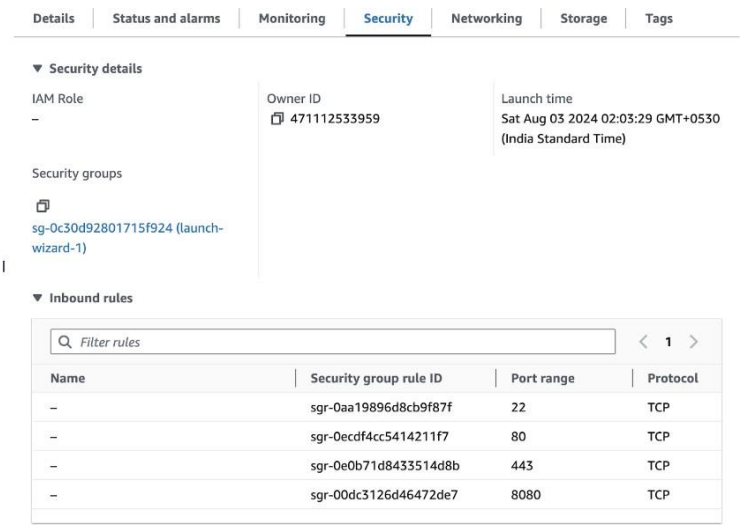
For creation of the EC2 Instance, Login to the AWS Cloud, Narrow down to EC2 Instance and Setup accordingly. Create security group and ports 22, 80, 443 and 8080. Remember to save your key pair.






Network Info
vpc-02b3dd537e0f3d664
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-3' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-05c3dc660cb6907f0
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes
750 hours of t2.micro (or t3.micro in
Cancel
Launch instance


Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
edge_resource...
i-02951070266e12543
Running
t2.micro
2/2 checks passed
View alarms
us-east-2a
i-02951070266e12543 (edge_resource_scheduler)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-02951070266e12543 (edge_resource_scheduler)
Public IPv4 address
13.58.216.224 | open address
Private IPv4 addresses
172.31.11.34
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-13-58-216-224.us-east-2.compute.amazonaws.com |
open address
Hostname type
Private IP DNS name (IPv4 only)



In your terminal change the permissions of the pem file

```
sudo chmod 600 ./ key_pair.pem
```

Secure Copy the edge_computing_predictor.py file and the '.sav' file into the instance

```
scp -i "key_pair.pem" edge_computing_predictor.py instance_dns_url:~
scp -i "key_pair.pem" AdaBoostClassifier.sav instance_dns_url:~
```

SSH into the instance and install the required packages

```
-i "edge_resource_scheduler.pem" ec2-user@ec2-18-118-1-44.us-east-2.compute.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-14-104 ~]$ sudo yum install python-pip
Last metadata expiration check: 0:03:08 ago on Fri Aug  2 20:34:17 2024.
Dependencies resolved.
================================================================================
 Package             Arch      Version                   Repository      Size
================================================================================
Installing:
 python3-pip         noarch    21.3.1-2.amzn2023.0.7     amazonlinux    1.8 M
Installing weak dependencies:
 libxcrypt-compat    x86_64    4.4.33-7.amzn2023         amazonlinux     92 k

Transaction Summary
================================================================================
Install  2 Packages

Total download size: 1.9 M
Installed size: 11 M
Complete!
[ec2-user@ip-172-31-14-104 ~]$ pip install bottle joblib pandas scikit-learn
Defaulting to user installation because normal site-packages is not writeable
Collecting bottle
  Downloading bottle-0.12.25-py3-none-any.whl (90 kB)
     |████████████████████████████████| 90 kB 3.4 MB/s
Collecting joblib
  Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
     |████████████████████████████████| 301 kB 5.4 MB/s
Collecting pandas
  Downloading pandas-2.2.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.1 MB)
     |████████████████████████████████| 13.1 MB 7.4 MB/s
Collecting scikit-learn
  Downloading scikit_learn-1.5.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.4 MB)
     |████████████████████████████████| 13.4 MB 37.5 MB/s
Collecting numpy>=1.22.4
  Downloading numpy-2.0.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (19.5 MB)
     |████████████████████████████████| 19.5 MB 29.2 MB/s
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3.9/site-packages (from pandas) (2022.7.1)
Collecting tzdata>=2022.7
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
     |████████████████████████████████| 345 kB 45.3 MB/s
Collecting python-dateutil>=2.8.2
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     |████████████████████████████████| 229 kB 51.1 MB/s
```

Run the API file

```
python3 edge_computing_predictor.py
```

Open a new terminal to build the UI but before that change the instance url in the app.py file

```
    # Define the URL of the API endpoint
    api_url = 'http://ec2-13-58-216-224.us-east-2.compute.amazonaws.com:8080/resource-scheduling'

    # Send POST request
    response = requests.post(api_url, headers={'Content-Type': 'application/json'}, json=data)
    response_data = response.json()  # Assuming the response is JSON

    # Extract server ID from response
    server_id = response_data.get('server_id', 'Not available')

    return render_template('output.html', server_id=server_id, data=response_data)

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=8080)
```

```
pip install flask
```

```
python app.py
```

3.3 Dataset

The dataset is downloaded from the below link:
https://www.kaggle.com/datasets/omarsobhy14/5g-quality-of-service?select=Quality+of+Service+5G.csv

4 References

https://aws.amazon.com/
https://www.anaconda.com/download
https://scikit-learn.org/stable/user_guide.html