CPE 322 – Lab 8

1. First, I installed the following Python packages:
   a.
```
jstefan1@DESKTOP-6SV6DJ0:~$ sudo pip3 install numpy scipy scikit-learn matplotlib pandas tensorflow keras
[sudo] password for jstefan1:
Collecting numpy
  Downloading numpy-1.26.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 18.2/18.2 MB 39.7 MB/s eta 0:00:00
Collecting scipy
  Downloading scipy-1.11.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (36.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 36.4/36.4 MB 35.6 MB/s eta 0:00:00
Collecting scikit-learn
  Downloading scikit_learn-1.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (10.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.8/10.8 MB 51.6 MB/s eta 0:00:00
Collecting matplotlib
  Downloading matplotlib-3.8.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.6/11.6 MB 49.9 MB/s eta 0:00:00
Collecting pandas
  Downloading pandas-2.1.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.3/12.3 MB 55.0 MB/s eta 0:00:00
Collecting tensorflow
  Downloading tensorflow-2.14.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (489.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 489.8/489.8 MB 2.0 MB/s eta 0:00:00
Collecting keras
  Downloading keras-2.15.0-py3-none-any.whl (1.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 22.3 MB/s eta 0:00:00
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.2.0-py3-none-any.whl (15 kB)
Collecting joblib>=1.1.1
  Downloading joblib-1.3.2-py3-none-any.whl (302 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 302.2/302.2 KB 13.3 MB/s eta 0:00:00
Collecting contourpy>=1.0.1
  Downloading contourpy-1.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (310 kB)
```
2. Then, I installed docker desktop on windows.
3. I then ran the following commands:
   a.
```
jstefan1@DESKTOP-6SV6DJ0:~$ docker pull tensorflow/tensorflow
Using default tag: latest
latest: Pulling from tensorflow/tensorflow
445a6a12be2b: Pull complete
01d85bc616ac: Pull complete
7d32cc707b9d: Pull complete
f9de7320e6e5: Pull complete
d320baddd40b: Pull complete
7edaa76cd685: Pull complete
22d5103f16e8: Pull complete
deddc4db4620: Pull complete
46c9aa3bcb29: Pull complete
60f13e9272e1: Pull complete
eddee94293d6: Pull complete
c90b2848018a: Pull complete
Digest: sha256:e39bbda78194ea735df765d6f15396d9a08354319af156ffec4ad2c6eeba469b
Status: Downloaded newer image for tensorflow/tensorflow:latest
docker.io/tensorflow/tensorflow:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview tensorflow/tensorflow
```

b.

```
jstefan1@DESKTOP-6SV6DJ0:~$ docker run -it tensorflow/tensorflow bash

________                               _______________
___  __/__________________________________  ____/__  /________      __
__  /  _  _ \_  __ \_  ___/  __ \_  ___/_  /_   __  /_  __ \_ | /| / /
_  /   /  __/  / / /(__  )/ /_/ /  /   _  __/   _  / / /_/ /_ |/ |/ /
/_/    \___//_/ /_//____/ \____//_/    /_/      /_/  \____/____/|__/


WARNING: You are running this container as root, which can cause new files in
mounted volumes to be created as the root user on your host machine.

To avoid this, run the container by specifying your user's userid:

$ docker run -u $(id -u):$(id -g) args...
```

c.

```
root@397ca6b279a7:/# apt update
Get:1 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease [1581 B]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  Packages [591 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [44.0 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1194 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1392 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1008 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1466 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [49.8 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1274 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1420 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [32.6 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [78.3 kB]
Fetched 28.9 MB in 9s (3386 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
20 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

d. apt full-upgrade
e. apt install git-all
f. git clone https://github.com/kevinwlu/iot.git
g. pip install keras

h.

```
root@397ca6b279a7:/# cd iot/lesson8
root@397ca6b279a7:/iot/lesson8# python keras_diabetes.py
2023-11-13 03:02:22.813438: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to
use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler fl
ags.
Epoch 1/150
77/77 [==============================] - 1s 3ms/step - loss: 2.5711 - accuracy: 0.5742
Epoch 2/150
77/77 [==============================] - 0s 3ms/step - loss: 1.1789 - accuracy: 0.6302
Epoch 3/150
77/77 [==============================] - 0s 3ms/step - loss: 0.9096 - accuracy: 0.6432
Epoch 4/150
77/77 [==============================] - 0s 3ms/step - loss: 0.8041 - accuracy: 0.6445
Epoch 5/150
77/77 [==============================] - 0s 3ms/step - loss: 0.7424 - accuracy: 0.6523
Epoch 6/150
77/77 [==============================] - 0s 3ms/step - loss: 0.7023 - accuracy: 0.6536
Epoch 7/150
77/77 [==============================] - 0s 3ms/step - loss: 0.6775 - accuracy: 0.6576
Epoch 8/150
77/77 [==============================] - 0s 3ms/step - loss: 0.6549 - accuracy: 0.6706
Epoch 9/150
77/77 [==============================] - 0s 2ms/step - loss: 0.6646 - accuracy: 0.6576
Epoch 10/150
77/77 [==============================] - 0s 3ms/step - loss: 0.6429 - accuracy: 0.6745
Epoch 11/150
77/77 [==============================] - 0s 3ms/step - loss: 0.6384 - accuracy: 0.6693
Epoch 12/150
```

```
accuracy: 73.31%
24/24 [==============================] - 0s 2ms/step
[0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1,
 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1,
 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1,
 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 0, 0]
```

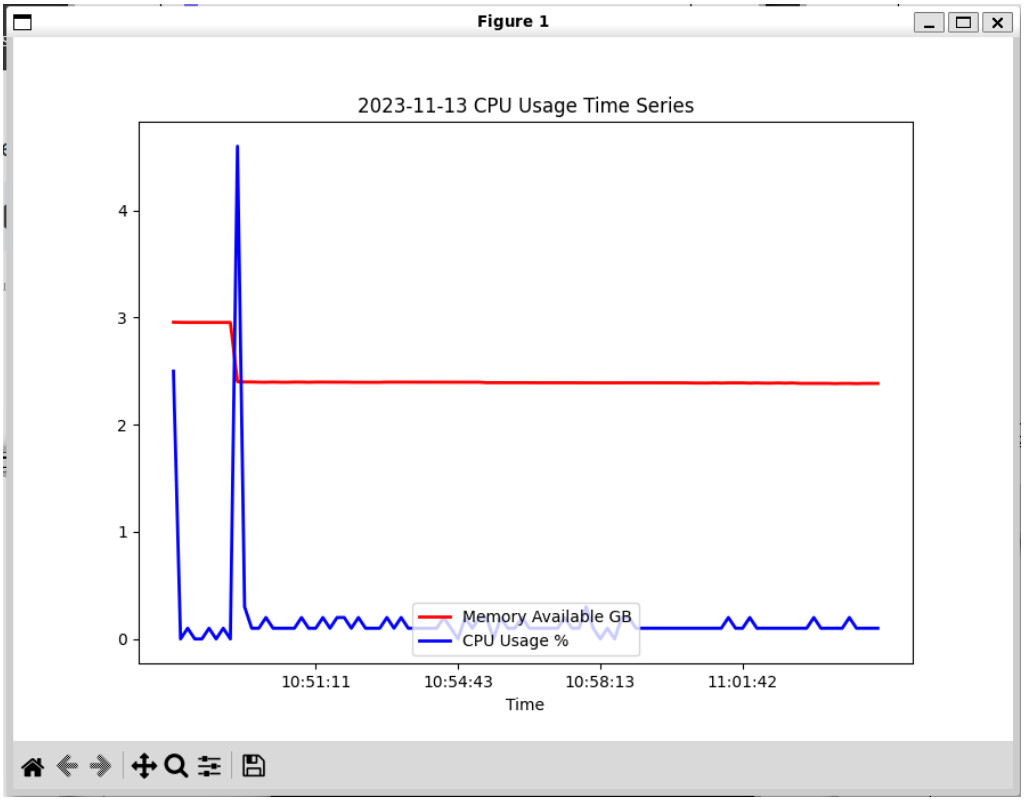
i.
```
root@397ca6b279a7:/iot/lesson8# python keras_first_network.py
2023-11-13 03:10:29.612562: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to
use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler fl
ags.
Epoch 1/150
77/77 [==============================] - 1s 3ms/step - loss: 5.9755 - accuracy: 0.4518
Epoch 2/150
77/77 [==============================] - 0s 3ms/step - loss: 2.4032 - accuracy: 0.5807
Epoch 3/150
77/77 [==============================] - 0s 2ms/step - loss: 1.7158 - accuracy: 0.5781
Epoch 4/150
77/77 [==============================] - 0s 3ms/step - loss: 1.4242 - accuracy: 0.5990
Epoch 5/150
77/77 [==============================] - 0s 3ms/step - loss: 1.2242 - accuracy: 0.6081
Epoch 6/150
77/77 [==============================] - 0s 2ms/step - loss: 1.1385 - accuracy: 0.6120
Epoch 7/150
77/77 [==============================] - 0s 2ms/step - loss: 1.0524 - accuracy: 0.6042
Epoch 8/150
77/77 [==============================] - 0s 2ms/step - loss: 0.9650 - accuracy: 0.6302
Epoch 9/150
77/77 [==============================] - 0s 3ms/step - loss: 0.9334 - accuracy: 0.6315
Epoch 10/150
77/77 [==============================] - 0s 2ms/step - loss: 0.8638 - accuracy: 0.6185
Epoch 11/150
77/77 [==============================] - 0s 2ms/step - loss: 0.8510 - accuracy: 0.6185
Epoch 12/150
77/77 [==============================] - 0s 2ms/step - loss: 0.8038 - accuracy: 0.6536
Epoch 13/150
```

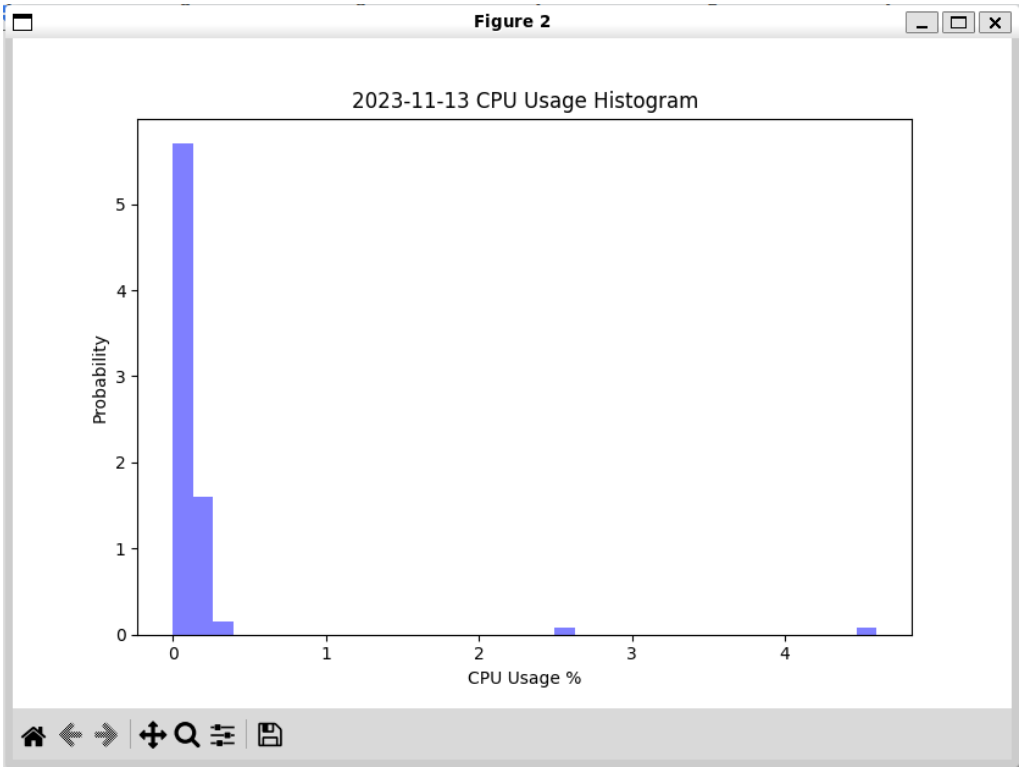
```
Accuracy: 75.52
```

j.
```
root@397ca6b279a7:/iot/lesson8# exit
exit
```

4. Next, I saved my csv file from Lab 7 to ~/demo in my linux terminal using the following command:
   a.
   ```
   jstefan1@DESKTOP-6SV6DJ0:~/demo$ sudo cp /mnt/c/Users/joeys/Downloads/cpudata3.csv .
   ```
5. Then I edited the following two Python files to use my csv file. I also edited the graph elements.
   a.
   ```
   jstefan1@DESKTOP-6SV6DJ0:~/demo$ nano plt_final.py
   jstefan1@DESKTOP-6SV6DJ0:~/demo$ nano plt_cv2.py
   ```

b. This is the output of the first file:





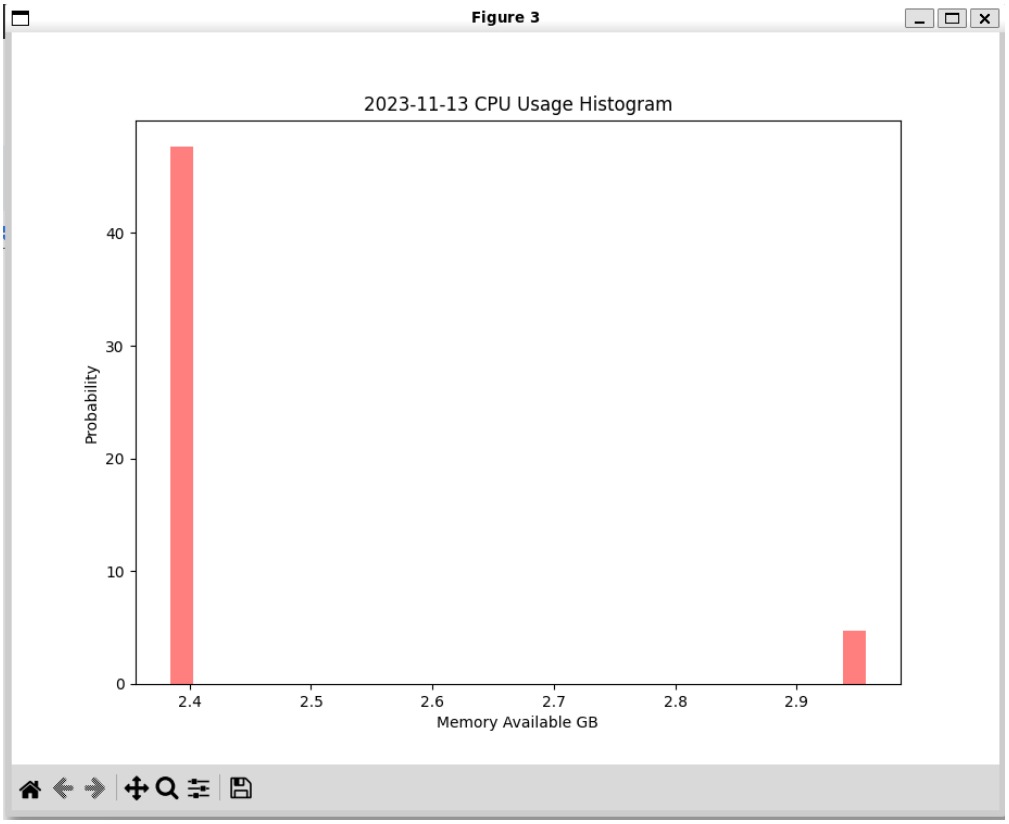


Figure 3
2023-11-13 CPU Usage Histogram
Probability
0
10
20
30
40
2.4
2.5
2.6
2.7
2.8
2.9
Memory Available GB

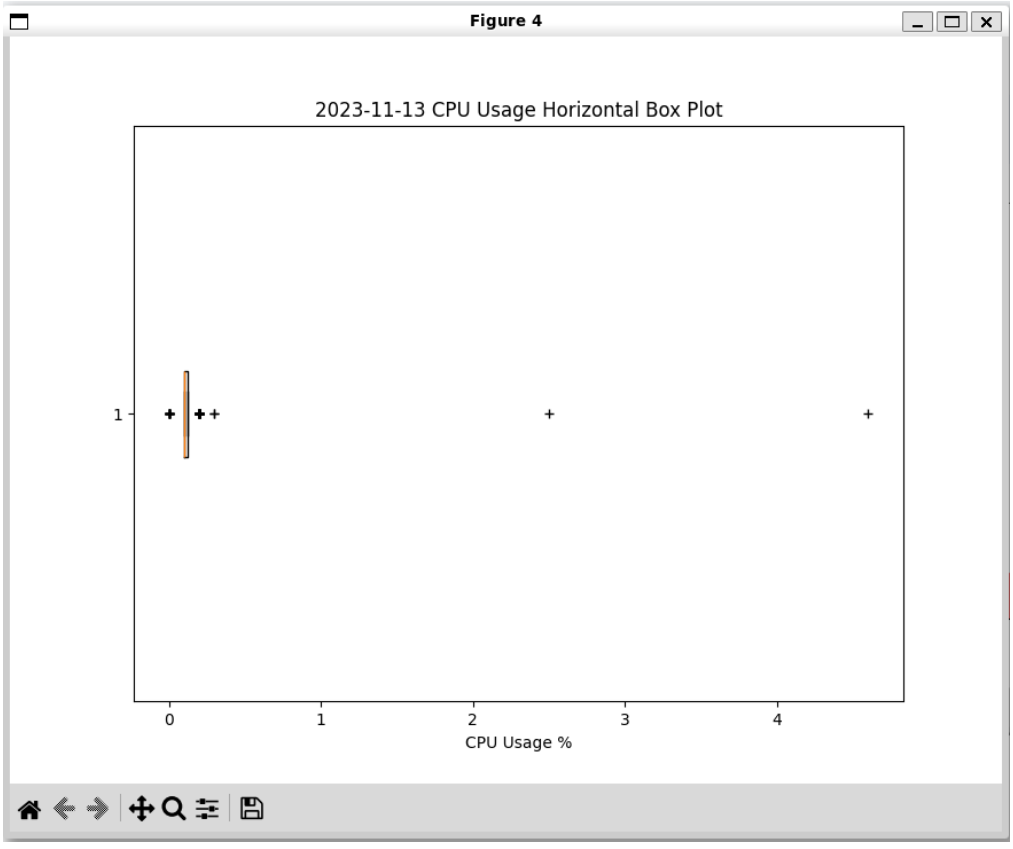


Figure 4
2023-11-13 CPU Usage Horizontal Box Plot
1
0
1
2
3
4
CPU Usage %

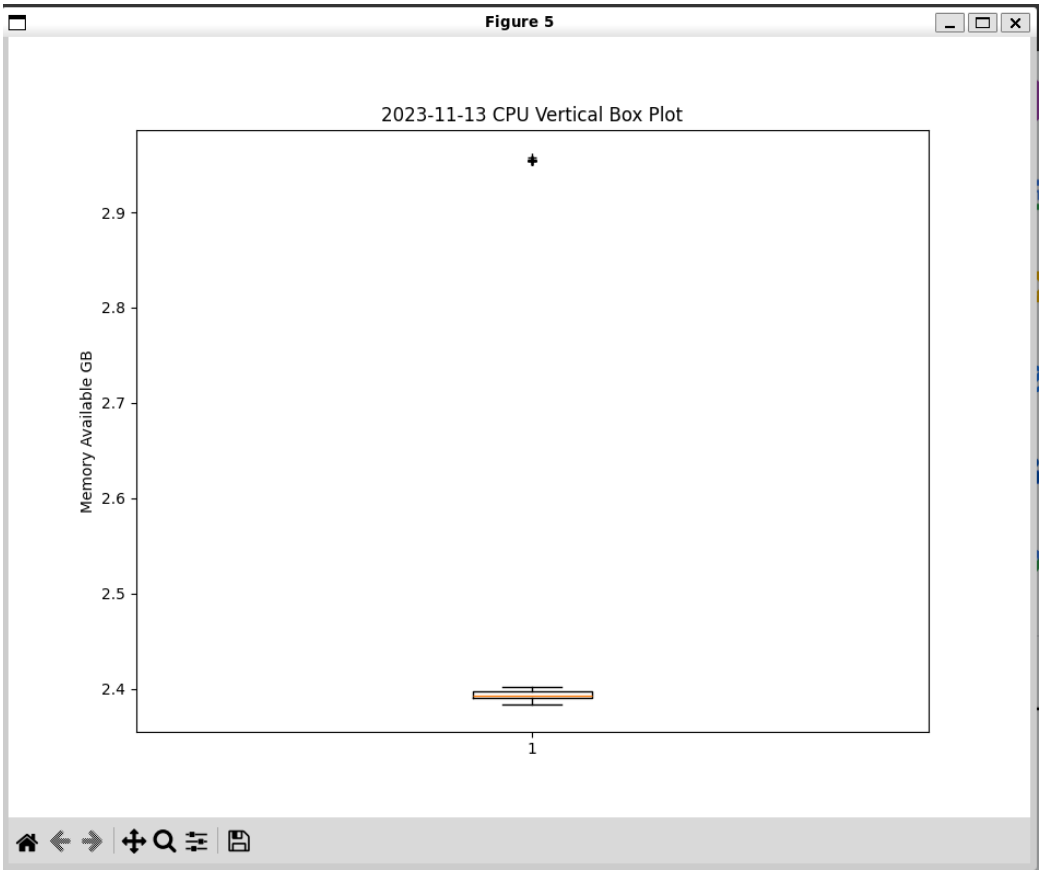


Figure 5
2023-11-13 CPU Vertical Box Plot
Memory Available GB
2.9
2.8
2.7
2.6
2.5
2.4
1

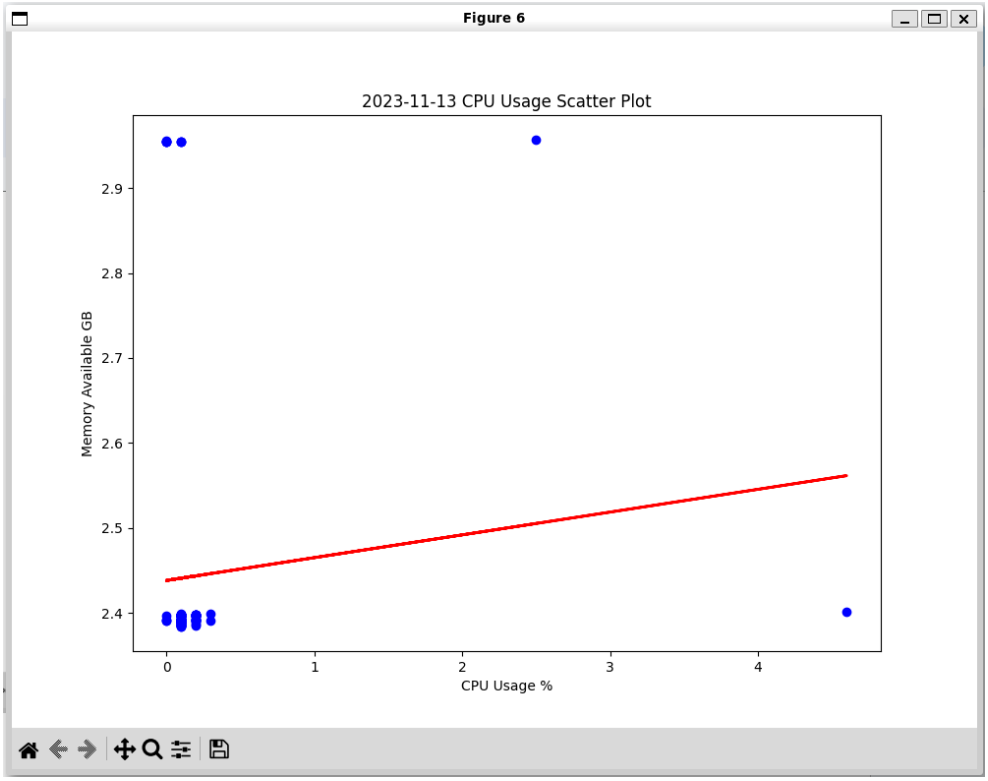


Figure 6
2023-11-13 CPU Usage Scatter Plot
Memory Available GB
2.9
2.8
2.7
2.6
2.5
2.4
0
1
2
3
4
CPU Usage %

c. This is the output of the second file: