Capstone Spring 2021

Dr.Acken & Prof.Jeske

User Manual for Multi-Factor Authentication Program

Team 11:

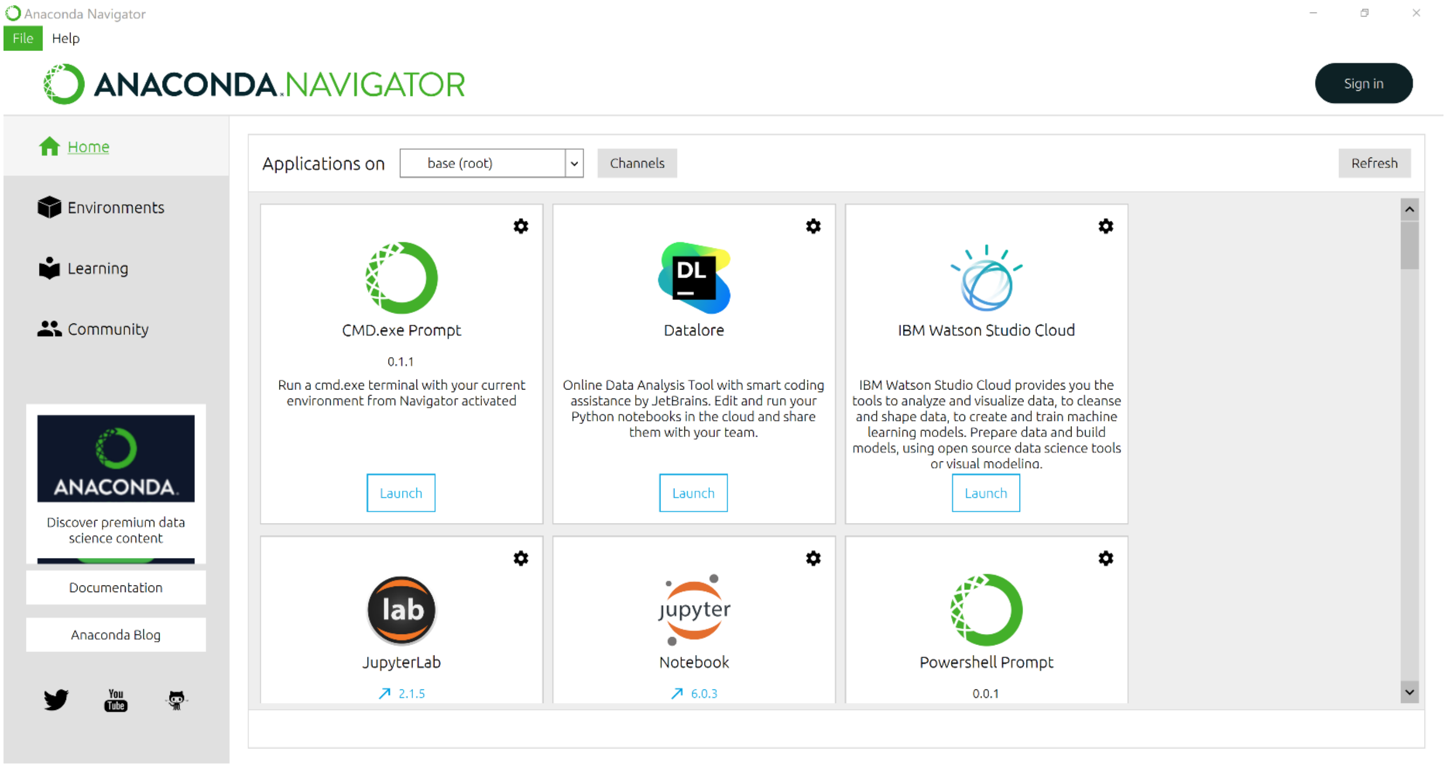
- Danny Wu
- Navid Karami-chamgordani
- Tamarr Stigler
- Uzias Cruz Asuncion

Portland State University

Part 1:

The program is written in Python. We need an IDE capable of handling Python. We recommend Anaconda Navigator - download it from this Link. Note, there are several versions available. For example, my laptop runs on Windows 10 64-bit, so I downloaded the 64-bit version. After downloading it, run the installer and install it at the recommended location. Leave all the installation configurations in defaults. If any issues, a detailed installation guide can be found here.

After installing the navigator, launch it. A window opens – see figure 1.



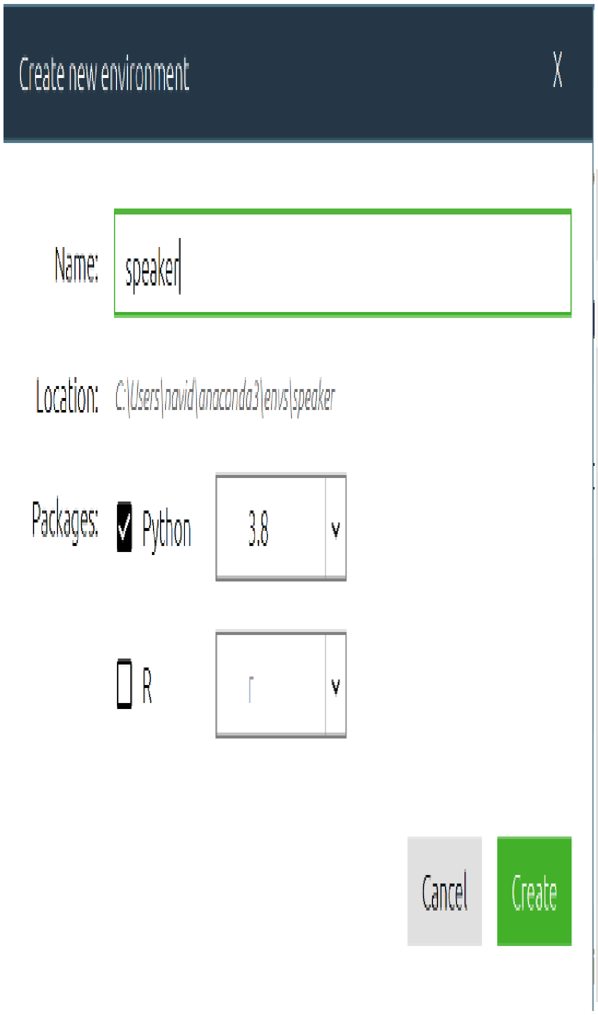
Run a cmd.exe terminal with your current environment from Navigator activated

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Figure 1

From the list, select Environments and then select create (left bottom corner), a window opens (figure 2) – keep everything the same as figure 2 and select create – environment creating would take a minute or less. Note, you could select a different name for your environment but in this document, we assume that the environment name is “speaker”. After creating the environment, you should see the environment listed – see figure 3.



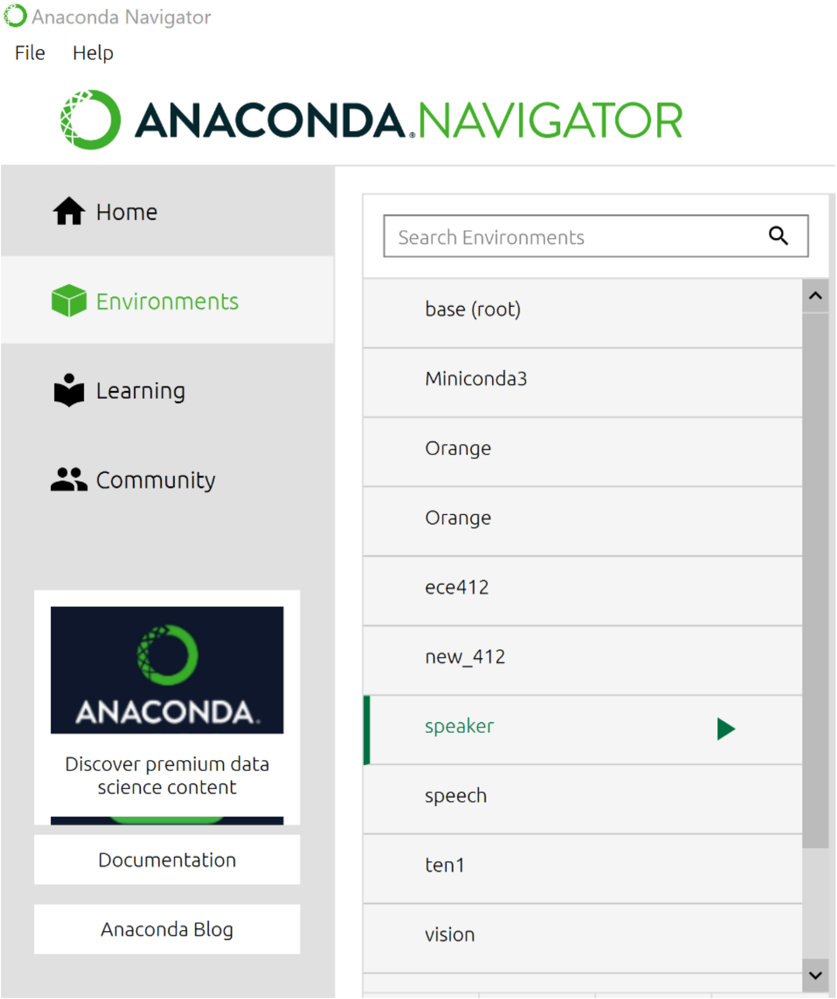
Name: speaker

Location: C:\Users\navid\anaconda3\envs\speaker

Packages: ☑ Python 3.8

□ R r

Figure 2



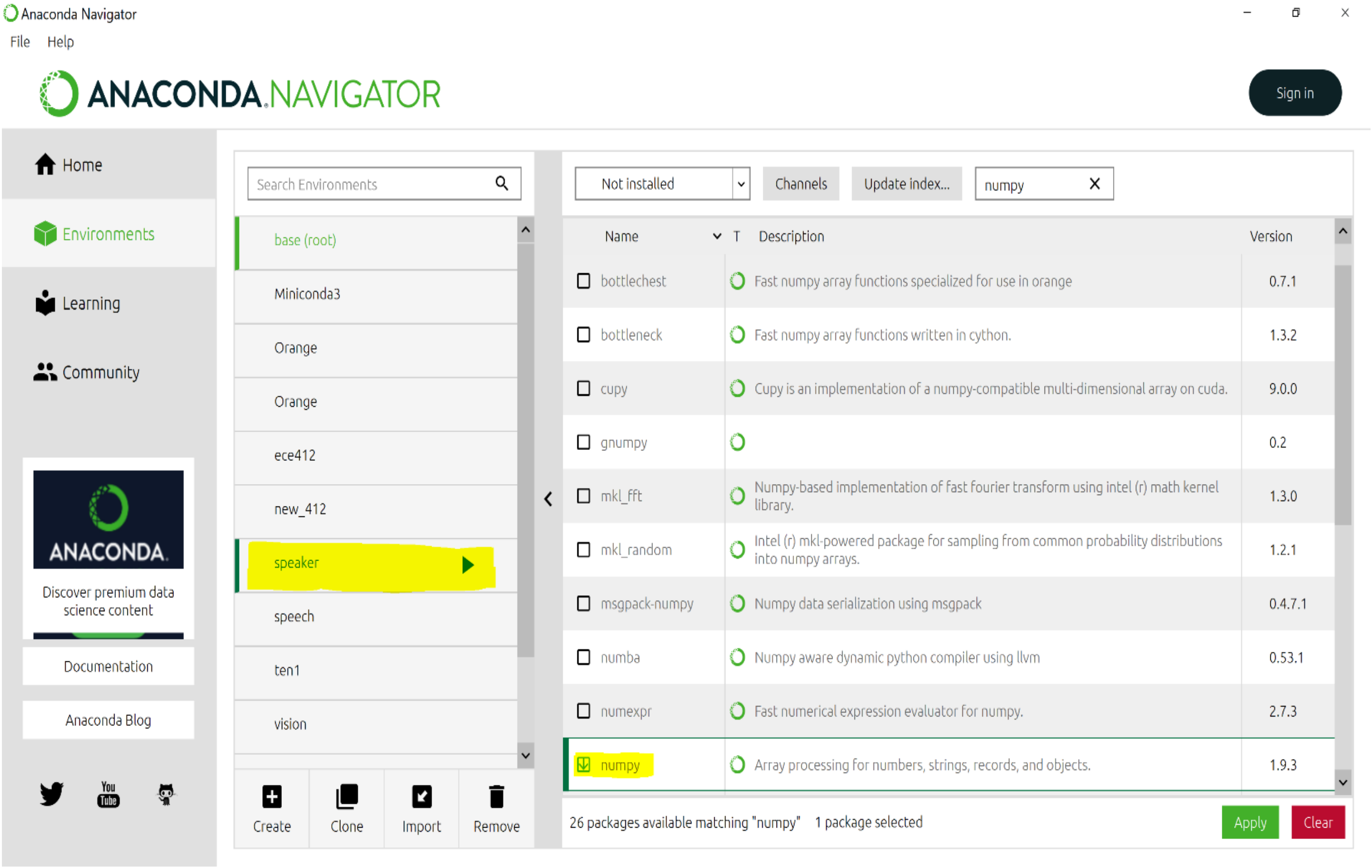
- base (root)
- Miniconda3
- Orange
- Orange
- ece412
- new_412
- speaker
- speech
- ten1
- vision

Figure 3

Now, it is time to install the required libraries. Below I listed them. I will explain how to install one of them – the rest can be installed by following the same steps.

- numpy
- scipy
- pyaudio
- scikit-learn
- pyserial

See figure 4 - in Environment tab, select “speaker” environment, open the drop-down menu, and select “Not installed” then search for “numpy”. Select the “numpy” from the list and select “apply”. The navigator will get the dependencies ready, and a small window opens, select “apply” and wait until the process is done. Repeat the same for the rest of the above libraries.



	Name	T	Description	Version
□	bottlechest	○	Fast numpy array functions specialized for use in orange	0.7.1
□	bottleneck	○	Fast numpy array functions written in cython.	1.3.2
□	cupy	○	Cupy is an implementation of a numpy-compatible multi-dimensional array on cuda.	9.0.0
□	gnumpy	○		0.2
□	mkl_fft	○	Numpy-based implementation of fast fourier transform using intel (r) math kernel library.	1.3.0
□	mkl_random	○	Intel (r) mkl-powered package for sampling from common probability distributions into numpy arrays.	1.2.1
□	msgpack-numpy	○	Numpy data serialization using msgpack	0.4.7.1
□	numba	○	Numpy aware dynamic python compiler using llvm	0.53.1
□	numexpr	○	Fast numerical expression evaluator for numpy.	2.7.3
☑	numpy	○	Array processing for numbers, strings, records, and objects.	1.9.3

26 packages available matching "numpy"   1 package selected

Figure 4

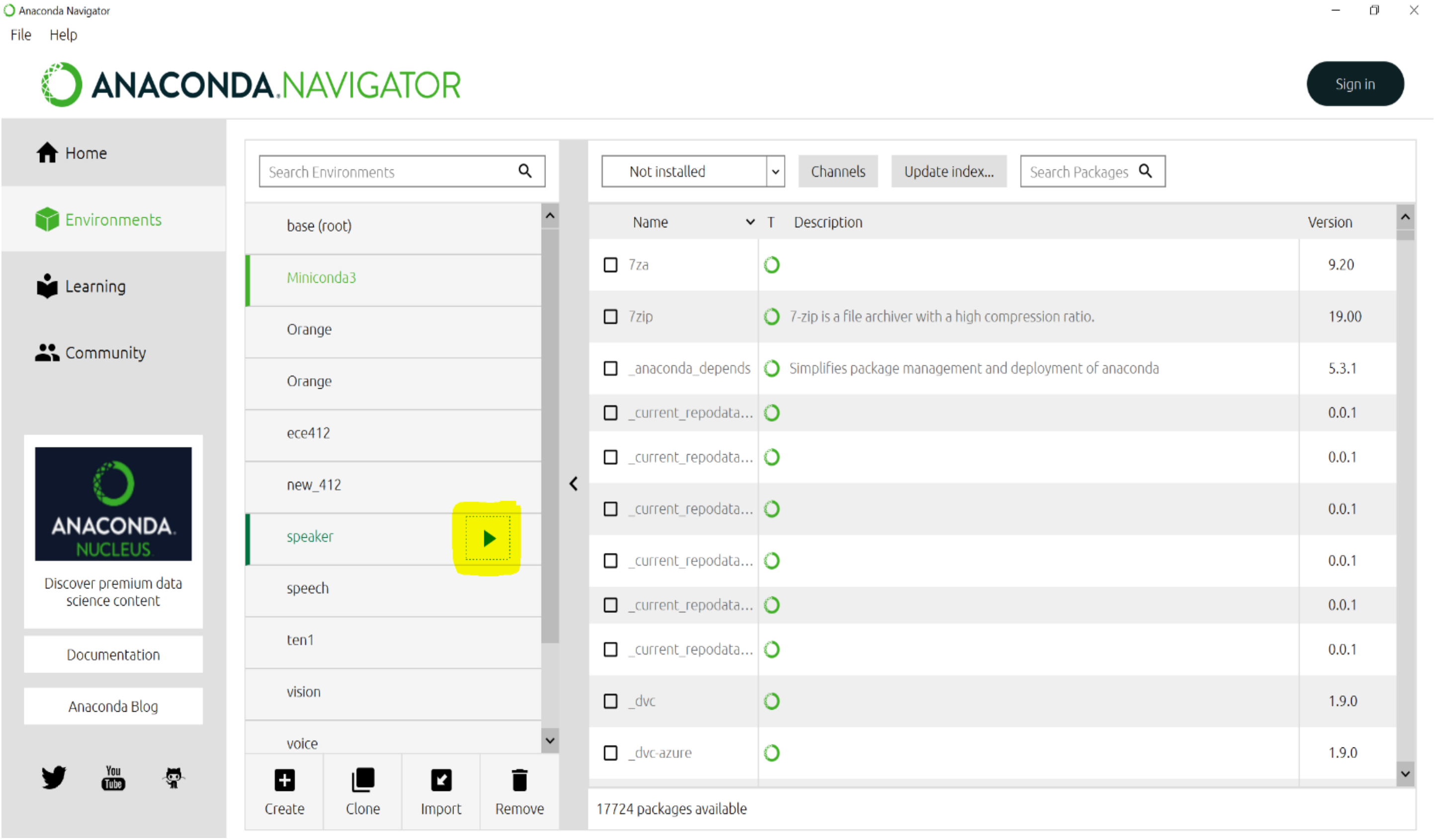
The last required library is installed by pip command, see the details below.

- python_speech_features (version 0.6)

See figure 5: ensure “speaker” environment is selected. Click on the green play icon. A drop-down menu opens. Select “Open Terminal”. In the terminal type the below command and hit enter (figure 6).

- pip install python_speech_features

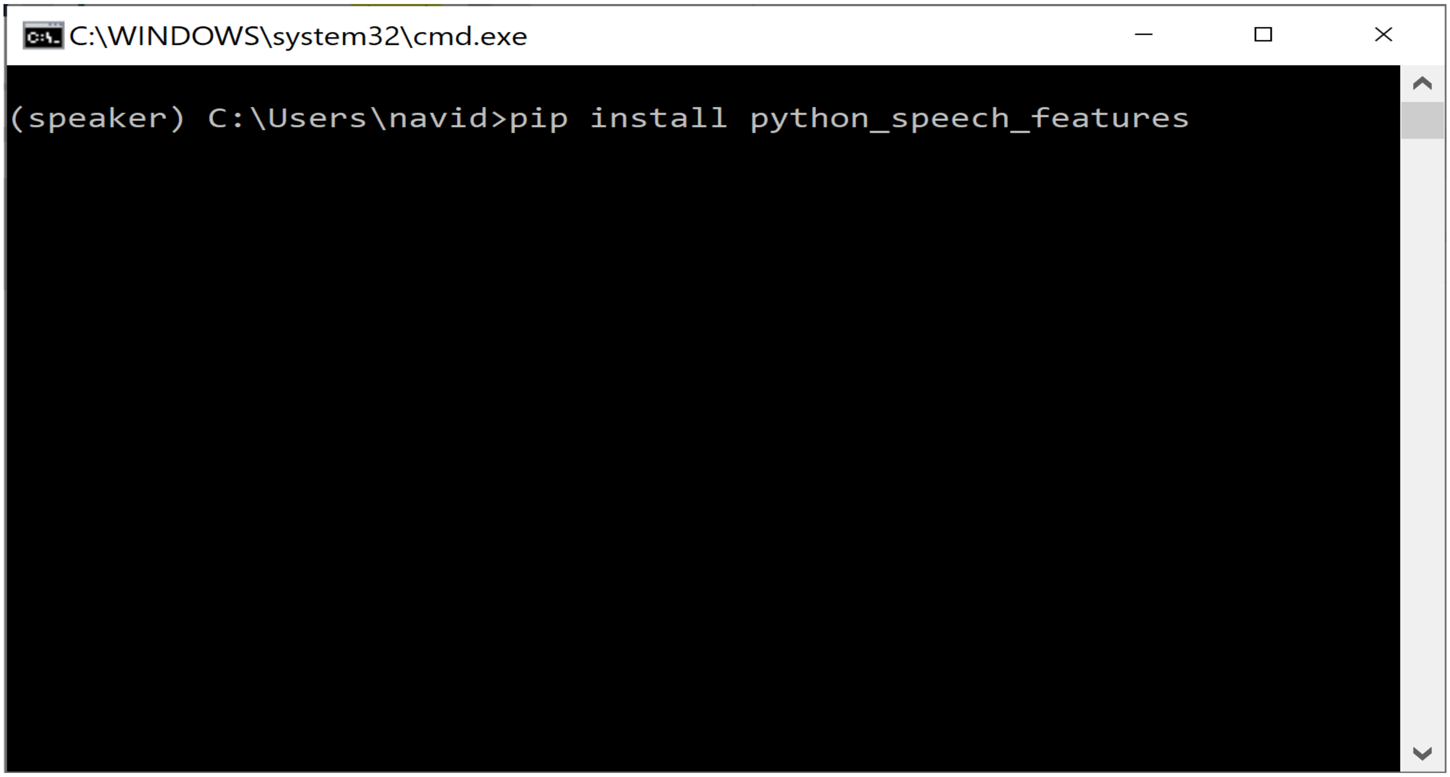
The installation should be quick, and the terminal will inform you when it is complete. Close the terminal after the installation.



Name	T	Description	Version
□ 7za	○		9.20
□ 7zip	○	7-zip is a file archiver with a high compression ratio.	19.00
□ _anaconda_depends	○	Simplifies package management and deployment of anaconda	5.3.1
□ _current_repodata...	○		0.0.1
□ _current_repodata...	○		0.0.1
□ _current_repodata...	○		0.0.1
□ _current_repodata...	○		0.0.1
□ _current_repodata...	○		0.0.1
□ _current_repodata...	○		0.0.1
□ _dvc	○		1.9.0
□ _dvc-azure	○		1.9.0

17724 packages available

Figure 5



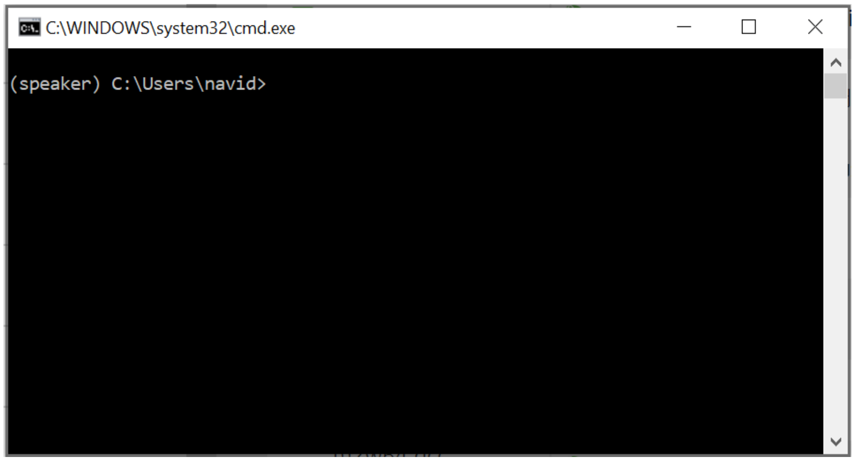
```
(speaker) C:\Users\navid>pip install python_speech_features
```

Figure 6

At this point, all the required libraries are installed. To run the code, you need to download it and place it in a directory. So, create a folder called “speaker” in your PC Documents directory. After creating the folder, download all the code(s)/file(s) from this link and copy paste all the downloaded items in the “speaker” folder. Before going any further, visit Appendix A and configure the GPS module connection and then return and continue.

Launch Anaconda navigator, select the Environments tab, select the “speaker” environment, click on the green “play” icon, and open a terminal (figure 7).

We should run the code from the folder where the code resides on – remember we created a folder called “speaker” in Document’s directory. You should then navigate through directories until you are in the “speaker” folder. Here is a link to help you through this process. Below I explain how to get to our speaker directory.



```
(speaker) C:\Users\navid>
```

Figure 7

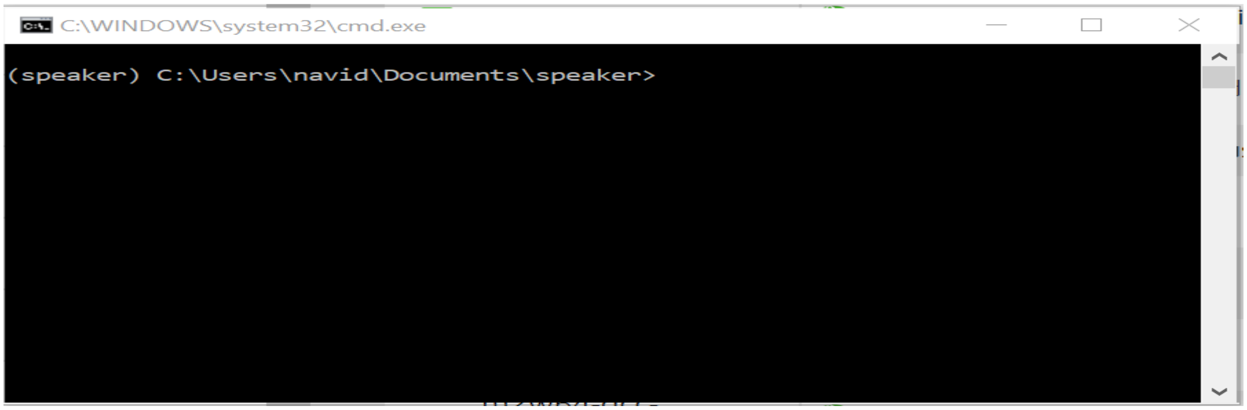
Your terminal should look the same, except “navid” is replaced by your username. Note, in the terminal, “(speaker)” indicates that you are in “speaker” environment. If the speaker environment is not showing, activate it by typing the following command in the terminal,

- conda activate speaker

After ensuring that the environment is active, type the following commands,

- cd documents\speaker

At this point you are in the speaker folder and ready to compile/launch the code (figure 8)



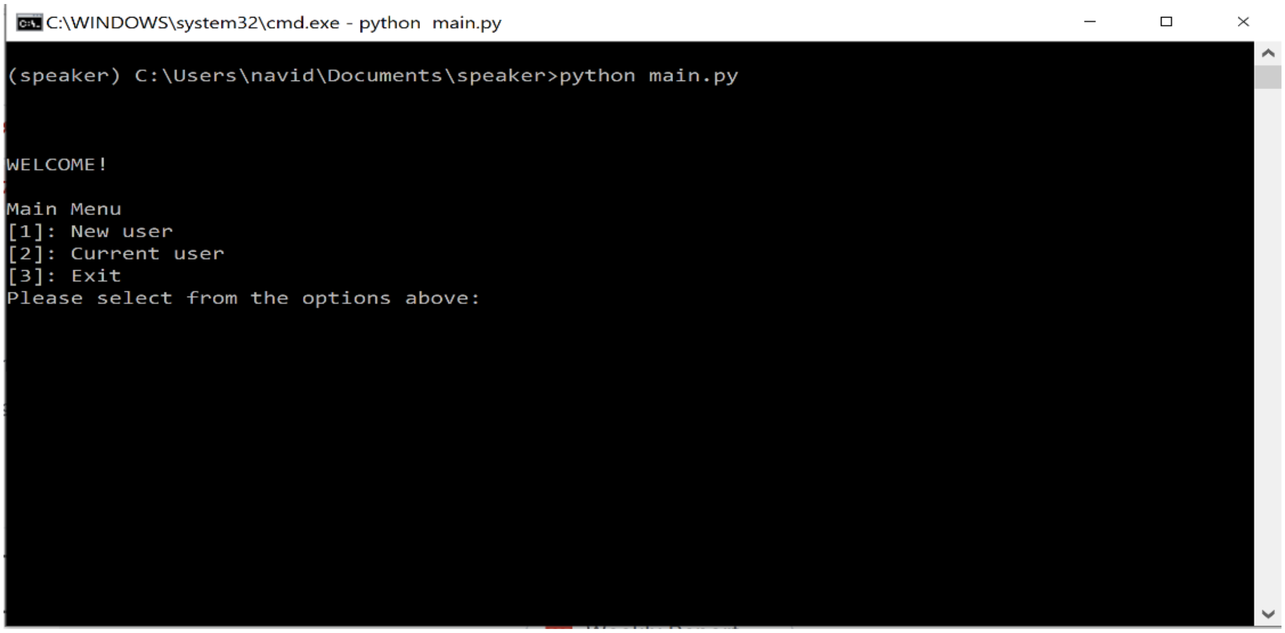
```
(speaker) C:\Users\navid\Documents\speaker>
```

Figure 8

Type the command below in the terminal and hit enter,

- python main.py

The program runs and the main menu is displayed (figure 9)



```
(speaker) C:\Users\navid\Documents\speaker>python main.py


WELCOME!

Main Menu
[1]: New user
[2]: Current user
[3]: Exit
Please select from the options above:
```

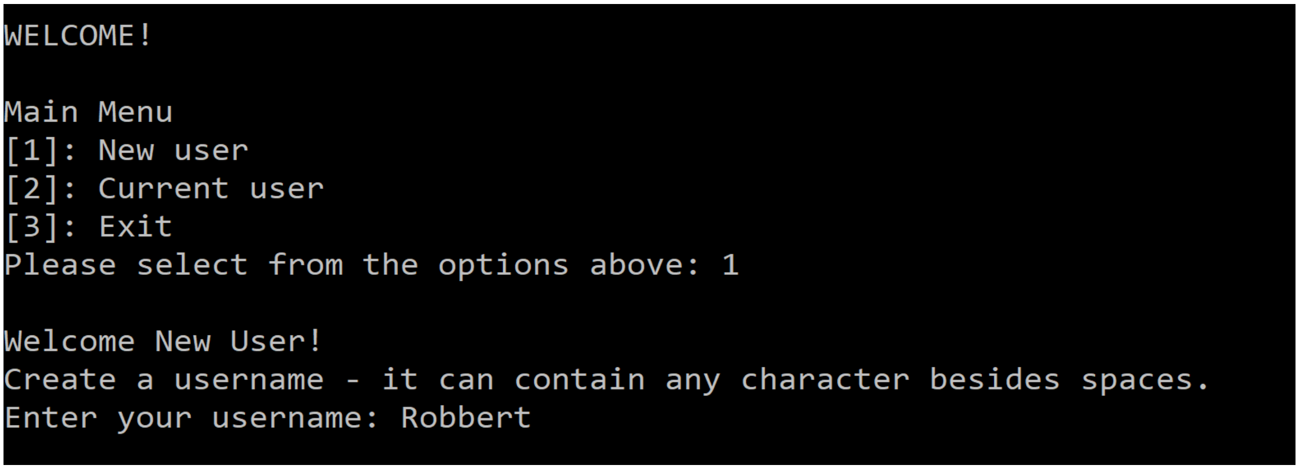
Figure 9

Part 2:

In this part, we will discuss the three options,

1. New user
2. Current user
3. Exit

New user: It allows new users to register and be saved as part of current users. When option #1 is selected (figure 10), the program asks the user to choose a username. The username can be a combination of letters, numbers, and special characters and it can have any length. However, it must not contain any space(s) as the program will not accept it. Also, you can't use the same username as someone else. If either of these rules are violated, the program will give you an error message. But you have unlimited amount of attempts to create a valid username. We typed "Robbert" and hit enter.



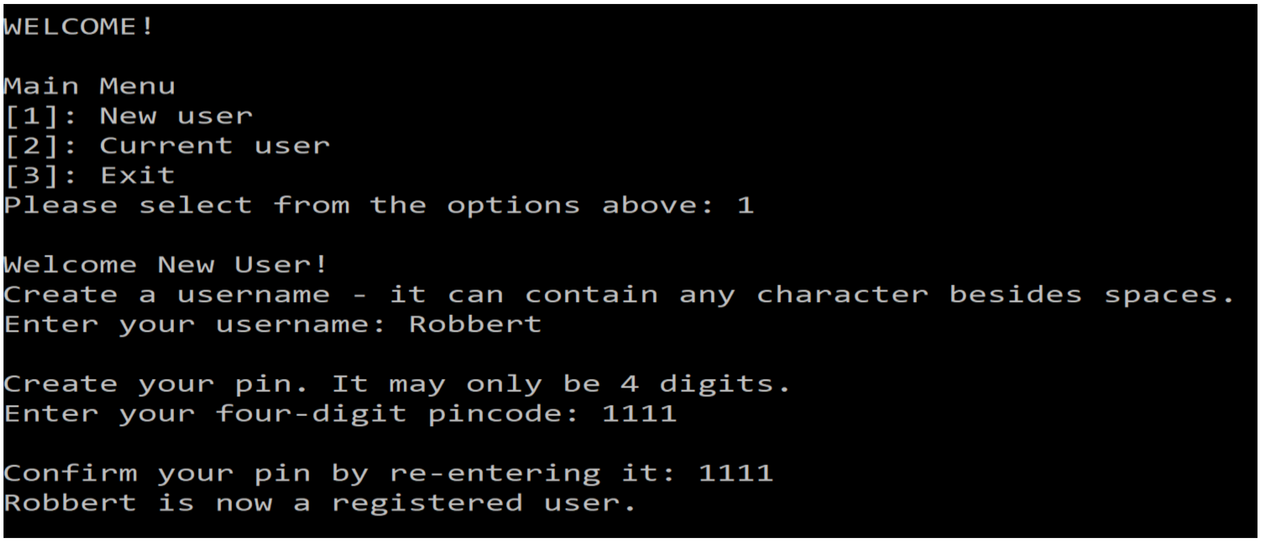
```
WELCOME!

Main Menu
[1]: New user
[2]: Current user
[3]: Exit
Please select from the options above: 1

Welcome New User!
Create a username - it can contain any character besides spaces.
Enter your username: Robbert
```

Figure 10

After selecting username, the program asks the user to create a pin. The pin must be a four-digit number. If anything other than 4 numbers are entered (for example letters or special characters or combination of such), the program displays an error message. If you enter more than 4 number or less than 4 numbers, you will get and error message as well. The program asks the user to re-enter the selected pin for confirmation purposes and at the end acknowledges that the user was successfully registered (figure 11). Again, the user gets an unlimited number of attempts to create a valid pin as well as confirming it. We typed "1111" and hit enter.



```
WELCOME!

Main Menu
[1]: New user
[2]: Current user
[3]: Exit
Please select from the options above: 1

Welcome New User!
Create a username - it can contain any character besides spaces.
Enter your username: Robbert

Create your pin. It may only be 4 digits.
Enter your four-digit pincode: 1111

Confirm your pin by re-entering it: 1111
Robbert is now a registered user.
```

Figure 11

After creating username and password, the program asks the user to say your phrase 15 times. The phrase is "this is (your name)". In this example, we said "this is Robbert". Be prepared as the program only gives you a 2 second delay between each try – you need to say your phrase in a three second time window (keep your tone and speed the same for consistency). Also, record your phrase in a quiet environment. Below we give a high-level explanation of what happens to the 15 recorded audio files and how a model for the new user is created.

- The 15 audio files get saved in the “speaker” folder
- We extract the features of the audio files with MFCC + Delta approach
- We save all the extracted features in one array/vector
- We calculate the Gaussian Matrix Distribution of the data points
- We save the calculated result as a .gmm file and place it in the “trained_models” folder
- We remove all the recorded audio files

At this point, the registration is fully completed - the program returns to the main menu. To verify your registration, you can visit the “speaker” folder and check the “username.txt” and “password.txt” files – “Robbert” should be added to the username text file and “1111” should be added to the password text file. Also, a file called “Robbert_15.wav.gmm” should be added to the “trained_models” folder.

From the main menu, select option 2 for “current user”.

- Enter your username and hit enter – “Robbert” in this example.
- Enter your password and hit enter – “1111” in this example.
- Press “r” when ready to say your phrase – “This is Robbert” in this example.
- The program welcomes Robbert and displays Robbert’s GPS coordinates

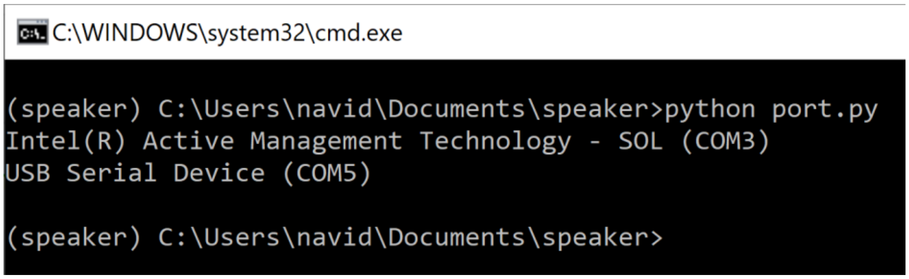
If you input a username that isn’t registered already, you will get an error message telling you that it is an invalid username, and you may attempt again. You get a total of 3 attempts to enter a registered username. If you fail all 3 attempts, the program locks itself for 10 sec and returns to the main menu. The same occurs with the pin. If you enter an incorrect pin 3 times, you get locked out for the 10 sec. You can add as many users as you wish by following the above steps. If you wish to remove a user from the system follow the below steps,

- Open “username.txt” and remove the username
- Open “password.txt” and remove the password
- Note, there is a one-to-one relationship between the username.txt and password.txt. For example, the password at line #3 belongs to username at line #3. Pay attention to this as users can have identical pin codes.
- Open “trained_models” folder and remove the user’s .gmm file

Note: The program comes with 4 pre-built models, usernames and passwords. Please visit “username.txt”, “password.txt” and “trained_models” folder to learn more about them.

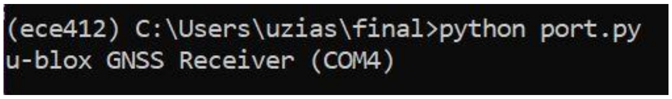
Appendix A:

- Plug the GPS module to one of your laptop/PC USB ports
- Open the anaconda prompt as stated in part 1
- Once you are in the right directory, you will want to run the “port.py” code:
  - python port.py
- The code will list out the USB ports that are being used
  - Look for the “USB Serial Device”
- The picture below shows what it will look like on your end. It might vary slightly than what is pictured but it should say something about USB Serial Device or GNSS Receiver.



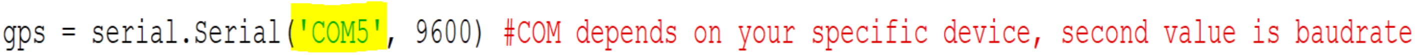
```
(speaker) C:\Users\navid\Documents\speaker>python port.py
Intel(R) Active Management Technology - SOL (COM3)
USB Serial Device (COM5)

(speaker) C:\Users\navid\Documents\speaker>
```



```
(ece412) C:\Users\uzias\final>python port.py
u-blox GNSS Receiver (COM4)
```

- In this example, COM5 is the port where the GPS is connected
- Now, head over to the folder where you have all the codes/text files
- You will need to open and edit the GPS.py code
- In line 7, there is the line of code where we set the USB port of the GPS device
- You will need to edit this to the COM# you got previously
- It will look like the figure below. Once you change that port, the GPS is ready to go



```
gps = serial.Serial('COM5', 9600) #COM depends on your specific device, second value is baudrate
```