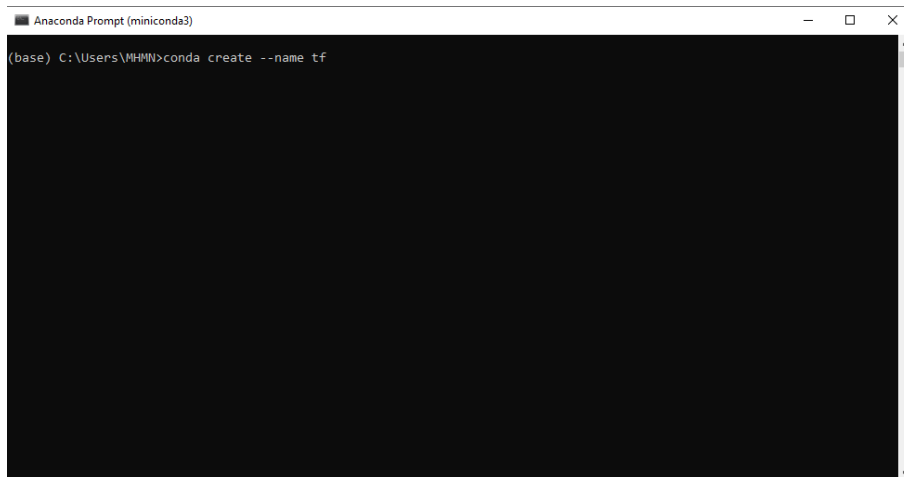


Creating Environment for TensorFlow

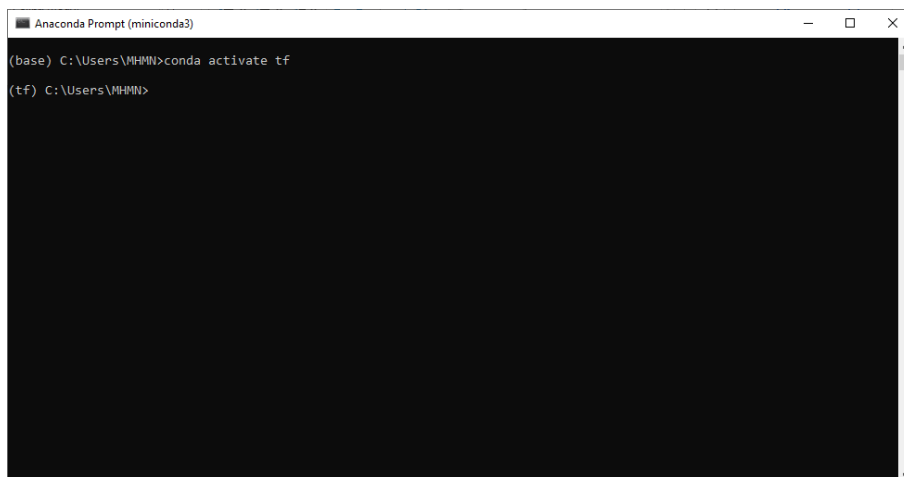
1. Open an Anaconda prompt via `Start | Anaconda3 | Anaconda Prompt`.
2. Create a dedicated environment for TensorFlow. Type the following command in the console. The name of the environment is "tf". With environment, all the packages to be installed will be in one environment (folder). Thus, it is easier to manage the packages and dependencies of the packages. If someday the environment is not needed, the environment can be removed together with all the packages.

```
conda create --name tf
```



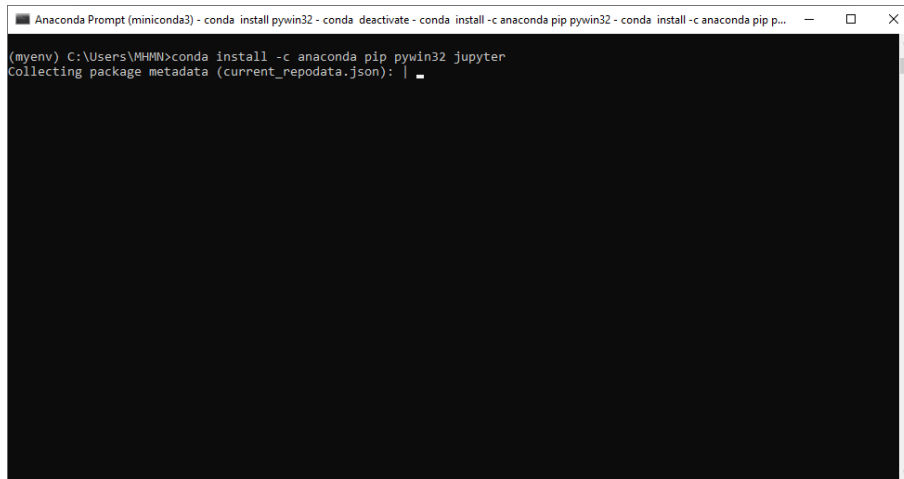
3. Activate the environment

```
conda activate tf
```



4. Now we can install the required packages in the created environment. First, we install pip and Jupyter Notebook as follow. Pip will be used to install the other packages including TensorFlow. This command will install the other dependencies such as Python.

```
conda install -c anaconda pip pywin32 jupyter
```



5. Run Jupyter Notebook to check if it is working. Click Jupyter Notebook icon from main menu or type the following command

```
jupyter notebook
```

or the following command to specify the working folder. Replace "username" with your username.

```
jupyter notebook "C:\users\username"
```

6. Now use pip to install the following packages.

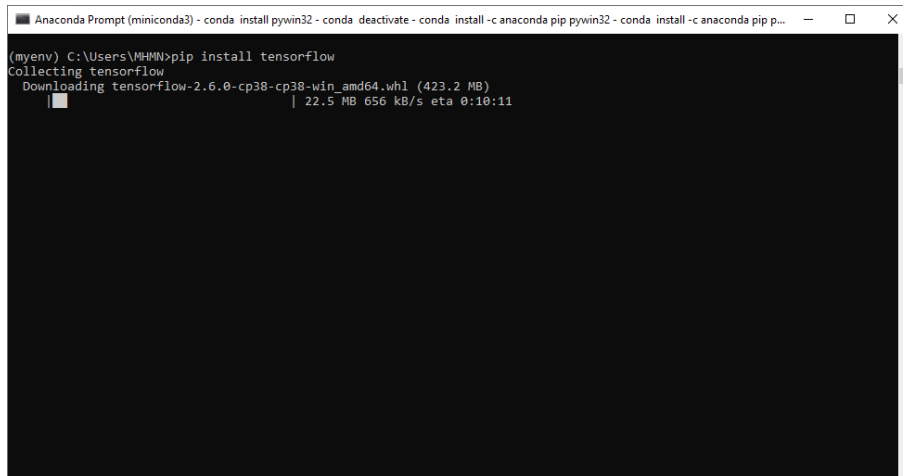
```
pip install numpy pandas scikit-learn matplotlib
```



Installing Tensorflow (No GPU)

7. Now install tensorflow

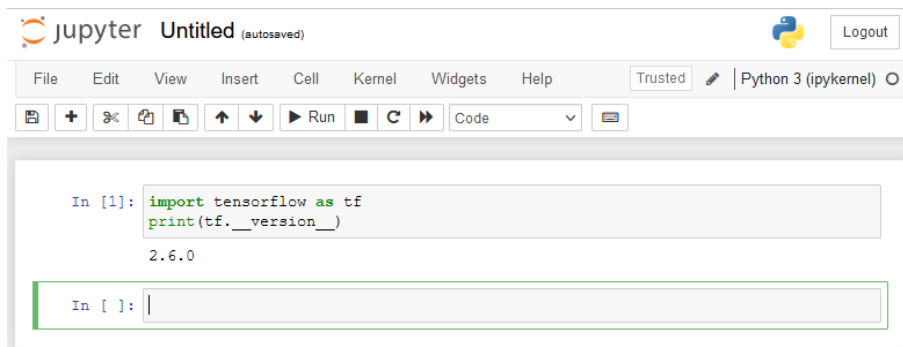
```
pip install tensorflow
```



```
Anaconda Prompt (miniconda3) - conda install pywin32 - conda deactivate - conda install -c anaconda pip pywin32 - conda install -c anaconda pip p...  
(myenv) C:\Users\WHMN>pip install tensorflow  
Collecting tensorflow  
  Downloading tensorflow-2.6.0-cp38-cp38-win_amd64.whl (423.2 MB)  
    | 22.5 MB 656 kB/s eta 0:10:11
```

6. Run the following command in Jupyter Notebook to ensure TensorFlow was installed correctly.

```
import tensorflow as tf  
print(tf.__version__)
```



```
Jupyter Untitled (autosaved) Python 3 (ipykernel)  
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)  
In [1]: import tensorflow as tf  
        print(tf.__version__)  
        2.6.0  
In [ ]: |
```

Installing TensorFlow (with GPU)

If the guide is not clear, please refer to this link:

<https://medium.com/@Gunter-Pearson/installing-latest-tensorflow-version-with-cuda-cudnn-and-gpu-support-on-windows-11-pc-e41fac5c5795>

7. Download CUDA Toolkit for Windows

The CUDA Toolkit (free) can be downloaded from the Nvidia website. Download the installer and install it.

<https://developer.nvidia.com/cuda-downloads>

8. Download and Install cuDNN

In order to download CuDNN, you have to register to become a member of the NVIDIA Developer Program (which is free). When you create an account, login and fill out some other required details about why you are using the account, you get the download page.

<https://developer.nvidia.com/cudnn>

9. Unzipping cuDNN files and copying to CUDA folders. There are three files in the unzipped cuDNN folder subdirectories which are to be copied into the CUDA Toolkit directories. These are cudnn64_7.dll, cudnn.h and cudnn.lib.

Copy cudnn.lib directly into the CUDA folder with the following path:

Copy cudnn64_7.dll into

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin\

Copy cudnn.h into

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\include\

Copy cudnn.lib into

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\lib\x64\

10. Add CUDA Installation Path

In Windows 10, the Environment Variables can be found by choosing:

Control Panel ->System and Security->System->Advanced System settings.

Add following path, assuming the CUDA version is 9.0

CUDA\v9.0\lib

CUDA\v9.0\libnvvp

11. Now install TensorFlow

`pip install tensorflow`

12. Run the following command in Jupyter Notebook to ensure TensorFlow was installed correctly.

```
import tensorflow as tf
print(tf.__version__)
```

