## **Computer preparation**

Dear All,

**Welcome to IBS7140!**

We begin by setting up our computers for all the Machine Learning (ML) applications that we will use with the following steps.

**1. Install the Anaconda distribution**, free for Mac, Windows, and Linux, from <https://www.anaconda.com/download.>

**2. Install Microsoft Visual Studio Code**, also free for Mac, Windows, and Linux, from <https://code.visualstudio.com.>

**3. Open a Github account** at <https://github.com.>

When you open your Github account fill in your description of who you are, and in particular, declare you are a member of Wayne State University. You may be asked for verification by submitting a photograph of your WSU OneCard. This is important in order to have access to the GPT4 based tools of autocoding (the Copilot module) that are free for educational institutions, which you will later on add to Visual Studio Code.

As part of your github preparation you may also install 'git' in your computer:

Mac: follow instructions in <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.> or:

1) install brew following web instructions 2) in a terminal execute: brew install git

Windows: follow instructions in <https://gitforwindows.org/.>

Linux: follow instructions in <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.>

**4. Create a *conda*virtual environment for machine learning (ml).** The sequence of installation steps will be shown here for a typical Mac or Linux device. Instructions on how to do the installation in a Window computers are very similar, but additional details can be found at:

<https://docs.conda.io/projects/conda/en/latest/user-guide/install/windows.html.>

**IMPORTANT**: All the installation steps must be carried out from a terminal window (Mac,Linux) or the Anaconda Prompt window (Windows). The Anaconda Prompt window will appear as one of the applications that can be launched from the Windows desktop.

After opening a terminal your prompt will appear as something like:

tcsh-% (for tcsh shell), bash-3.2$ (for bash shell), user: (dos shell)

At the prompt, type the exact sequence of commands shown below, one line at a time, and wait for the completion of each command execution before typing the next command.

conda -- version (yes, 2 minus signs in front of version)

conda create -n ml\_311 python=3.11.13 (when conda asks you to proceed, type y. This will create the virtual env ml\_311 using Python release 3.11.13).

conda env list (this command should show the existence of two environments: base and ml\_311. **VERY IMPORTANT:** **do not do any further installation of programs in the base environment**.

conda activate ml\_311 (this command brings you inside the ml\_311 virtual env: the prompt will change to (ml\_311) tcsh-%.

Next, we will install all the programs. Inside a virtual env, programs can be installed either with conda install or with pip install. While you will probably find on the web a general recommendation to use conda install, in our case this installation command produces some inconsistencies between libraries used by the different programs inside the environment. For this reason, we will use instead pip install).

During this installation you can use the command *pip* or *pip3.* In general, if your *conda* environment is updated, there is no difference, although the installation of some modules (i.e., *torch*) is recommended with *pip3*. It is a good idea to start by *updating* pip (it will update *pip3* too).

conda update pip

pip3 install numpy==1.26.4

pip3 install numpy-ml

pip3 install scikit-learn

pip3 install pandas

pip3 install openpyxl

pip3 install matplotlib

pip3 install ipykernel  
  
pip3 install jupyter  
  
 1. for Linux, Windows, or Mac without AMD GPU or M chip

pip3 install torch torchvision

2. for Linux, Windows with Nvidia Cuda 12.9

pip3 install torch torchvision --index-url https://download.pytorch.org/whl/cu129

3. for Linux, Windows with Nvidia Cuda 12.9

pip3 install torch torchvision --index-url https://download.pytorch.org/whl/cu129

4. for Mac with AMD GPU or M chip. Latest version Mac with M3 chip simply:

pip3 install torch torchvision

The installation with earlier Macs with M1, M2 chip require as a pre-requisite the presence of *X-code*. You can install it with:

xcode-select --install

pip3 install --pre torch torchvision --extra-index-url https://download.pytorch.org/whl/nightly/cpu

**Check the version of torch installed using 'pip list', then change the following line accordingly. For example if torch 2.2.2 then:**  
  
pip install torch\_scatter torch\_sparse torch\_cluster torch\_spline\_conv torch\_geometric -f [https://data.pyg.org/whl/torch-2.2.2+cpu.htmlLinks to an external site.](https://data.pyg.org/whl/torch-2.3.0+cpu.html)

pip3 install dgl==2.0.0

pip3 install dglke

pip3 install dgllife

# pip install lightning

pip3 install ogb

pip3 install ipympl  
  
# pip3 install torchtext (do not install if torch version is 2.1.1 or 2.4.0. Only possible for torch 2.2.0)  
  
pip3 install torchmetrics  
  
pip3 install torchsummary  
  
# pip3 install pytorch-lightning  
  
pip3 install tensorboard (real time display of training progression)  
  
pip3 install transformers  
  
pip3 install h5py  
  
pip3 install nltk  
  
pip3 install gdown  
  
pip3 install pyprind  
  
pip3 install umap-learn   
  
pip3 install xgboost  
  
pip3 install seaborn

pip3 install rdkit-pypi

pip3 install ipywidgets

pip3 install nglview

# pip install jupyterlab

pip3 install neo4j

pip3 install graphdatascience

pip3 install neo4j-uploader

pip3 install tiktoken

pip3 install 'litgpt[all]'

At the end of the installation type again:

pip list or conda list (this will show you all the programs that were installed)

conda deactivate (this brings you out of the virtual env)

**Special installation on WSU High Performance Computing Grid**

Follow the instructions in <https://tech.wayne.edu/kb/high-performance-computing/hpc-tutorials/500191> to request a grid account (specify you are taking IBS7140 - Foundation of ... and that you need an account to execute the Python scripts presented in the course) and log onto the grid. Remember that, in order to login, you must be connected to the WSU VPN (<https://tech.wayne.edu/kb/security/wsu-virtual-private-network>) .

1. Start a Jupyter lab session (this will connect you to a node)

2. If a Launcher did not appear already, from the 'File' tab start a new Launcher

3. In the Launcher menu start a 'Terminal'. You will see a prompt like:

(base) ['wsuid'@'mdt17']

in which 'wsuid' is your wsu id, and  'mdt17' is the node on the grid that has been assigne to you.

4. Execute the following commands:

conda create -n ml\_311 python=3.11.13

conda activate ml\_311

pip3 install scikit-learn  
pip3 install pandas  
pip3 install openpyxl  
pip3 install matplotlib  
pip3 install ipykernel  
pip3 install jupyter  
pip3 install torch torchvision  
pip3 install torch\_geometric  
python -c "import torch; print(torch.\_\_version\_\_)"  
python -c "import torch; print(torch.version.cuda)"

(choose the version of torch and cuda you have. For example, if you have torch 2.4 and cuda 12.1):  
pip3 install torch\_scatter torch\_sparse torch\_cluster torch\_spline\_conv -f https://data.pyg.org/whl/torch-2.4.0+cu121.html  
pip3 install lightning  
pip3 install ogb  
pip3 install ipympl  
pip3 install torchmetrics  
pip3 install torchsummary  
pip3 install pytorch-lightning  
pip3 install tensorboard  
pip3 install transformers  
pip3 install h5py  
pip3 install nltk  
pip3 install gdown  
pip3 install pyprind  
pip3 install umap-learn  
pip3 install xgboost  
pip3 install seaborn  
pip3 install rdkit-pypi  
pip3 install ipywidgets  
pip3 install nglview  
pip3 install jupyterlab  
pip3 install torchdrug

pip3 install neo4j

pip3 install graphdatascience

pip3 install neo4j-uploader

pip3 install tiktoken

pip list

python -m ipykernel install --user --name ml\_311 --display-name "Python (ml\_311)"

conda deactivate

conda env list

This final command will show the available environments in you grid connection. When you open a jupyter notebook select 'ml\_311' as the kernel.

**5.  Visual Studio Code settings**

By now you must have been able to create your 'conda' environment and install all the programs (if not, we will troubleshoot any problems).

We can now proceed to continue the setup of Visual Studio Code (VS). The following setup is for VS installed in your laptop, and is not applicable to the version of VS available on the WSU grid.

In previous courses you might have had the opportunity to use a Jupyter Notebook or the Jupyter Lab Graphic Interphase. Almost all the code files of our course will be Jupyter Notebooks (i.e., 'myprogram.ipynb). However, in order to open, edit, and execute these files it may be convenient to use VS instead of the native Jupyter Notebook application. In order to do it we need to complete the VS installation.

1. Launch VS
2. In the bar on the left side of the workspace you will see a little icon with 4 little squares one of which is detached:

Click on the icon. It will open the 'Extension' section. Here is where you will install additional extension to VS. Some extension will appear already in the section as 'Recommended Extensions'. If some of all the extensions that you will need in your work are not already recommended, you can search for them in the text window at the top which shows: 'Search extensions in the marketplace'.

You will need to install the following extensions:

* Jupyter
* Jupyter Cell Tag
* Jupyter Keymap
* Jupyter Notebook Renderers
* Jupyter Slide Show
* Pylance
* Python
* Draw
* Draw.io Integration
* Draw.io Integration Mermaid plugin
* Excalidraw
* GitHub Copilot (a Copilot icon [a small face with goggles] should appear in the bar at the bottom right: if it doesn’t, read more on this below)
* GitHub Copilot Chat (a Copilot icon [two small call out text bubbles] should appear in the bar on the left: if it doesn’t, read more on this below)
* Data Wrangler
* Vscode-pdf

**Do NOT install** 'Kite AutoComplete AI code' and 'Vim', even if recommended.

**6. Create a directory IBS7140.**In future, you will place all the course files in this directory.

## **Final computer setup**

At the very beginning of this message I have recommended you sign up for an account in GitHub. GitHub is an industry standard tool for developing, maintaining, and storing software. You won’t be using GitHub directly in this course, but you need an account to get access to Copilot. Signing up for a GitHub account is free for students or a faculty member at an academic institution. While Copilot can work with several computer languages, in this course we will use it to generate Python code. You will still want to be able to read and understand the code generated by Copilot, and thus becoming proficient in Python should be one of your goals for the end of this course.

You can use any text editor to write Python programs. However, if you want a nice programming environment where you can write code, easily get suggestions from Copilot, and run your code, VSCode is the preferred tool. VSCode is used globally by professional software engineers, which means you’ll be able to work and learn while using this environment long after finishing the course.

Completing your system setup requires the following additional steps:

1. Sign up for Copilot
2. If you have not done it yet, go to [https://github.com/signup site.](https://github.com/signup%20site.) and sign up for a GitHub account.
3. Follow the link [Apply to GitHub Global Campus as a student.](https://docs.github.com/en/education/explore-the-benefits-of-teaching-and-learning-with-github-education/github-global-campus-for-students/apply-to-github-global-campus-as-a-student)
4. Go to the section [Applying to GitHub Global Campus.](https://docs.github.com/en/education/explore-the-benefits-of-teaching-and-learning-with-github-education/github-global-campus-for-students/apply-to-github-global-campus-as-a-student#applying-to-github-global-campus) and follow the steps therein. Explain you will use Github and Github Copilot in your course IBS7140: Foundation of data science. You can take a picture of your OneCard as evidence of your affiliation with WSU.
5. If you get lost in the maze of links of GitHub, you can go directly to the GitHub Student Developer Pack and verify your student status on GitHub at <https://education.github.com/pack.>
6. Go into your ‘settings’ in the GitHub account and enable Copilot.
7. Launch VS Code and install the following code extensions, accessed from the icon with the four little squares in the selection bar on the left. You may have to do a search to find the extensions:
8. GitHub Copilot (by GitHub). If you are authorized to use Copilot there will appear a small icon of a pilot head with goggles at the bottom right of your VS workspace:
9. GitHub Copilot Chat (by GitHub). After you install the extension, to enable GitHub Copilot Chat in VS Code you must be accepted in the waiting list.

Goto <https://marketplace.visualstudio.com/items?itemName=GitHub.copilot-chat.> and join the [waitlist.](https://github.com/github-copilot/chat_waitlist_signup/join) Enter the requested information. You will receive an email notifying when you are accepted to use **GitHub Copilot chat.**

Activation of these extensions may require giving permission to VS Code to connect with GitHub. Grant permission and type in the GitHub Account password if requested.

I look forward to meeting you soon.

dg