



FALMOUTH
UNIVERSITY

COMP712 Classical Artificial Intelligence

Workshop: Data Science using Python

Dr Daniel Zhang @ Falmouth University
2023-2024 Study Block 1

Table of Contents

- [COMP712 Classical Artificial Intelligence](#)
- [Workshop: Data Science using Python](#)
- [Table of Contents](#)
- [Introduction](#)
- [Preparation](#)
- [JupyterLab Notebook](#)
- [The Repository](#)
- [The Notebook](#)
- [Note:](#)

Introduction

[Top](#)

Welcome to this comprehensive workshop on Data Science using Python!

In this session, we'll delve into the essential Python libraries for scientific computation and machine learning: NumPy , Matplotlib , and Pandas . These powerful tools will empower us to handle data manipulation, visualisation, and analysis, laying the foundation for building sophisticated machine learning models.

Apart from familiarising yourselves with these libraries, you will write your code in JupyterLab - the integrated interactive development environment.

Preparation

Top

To work with Jupyter notebook, I would assume you have all the necessary libraries and tools installed successfully. For instance, you should have the following libraries on the machine you are working on:

- python
- numpy
- matplotlib
- scipy
- pandas
- jupyterlab

If any of the software or libraries are not installed, please follow the instructions below to get them installed.

0. Download and install python from [Python Website](#)

1. Use Python's package manager, pip , to install the required libraries. Open your terminal or command prompt and run the following command:

```
pip install numpy
pip install matplotlib
pip install scipy
pip install pandas
```

2. You can also combine them together as one command for installation:

```
pip install numpy matplotlib scipy pandas
```

3. And, more importantly, install JupyterLab using the command:

```
pip install jupyterlab
```

4. If JupyterLab has been installed properly, you have two options to run the notebook:

i. Start a command window and type

```
jupyter lab
```

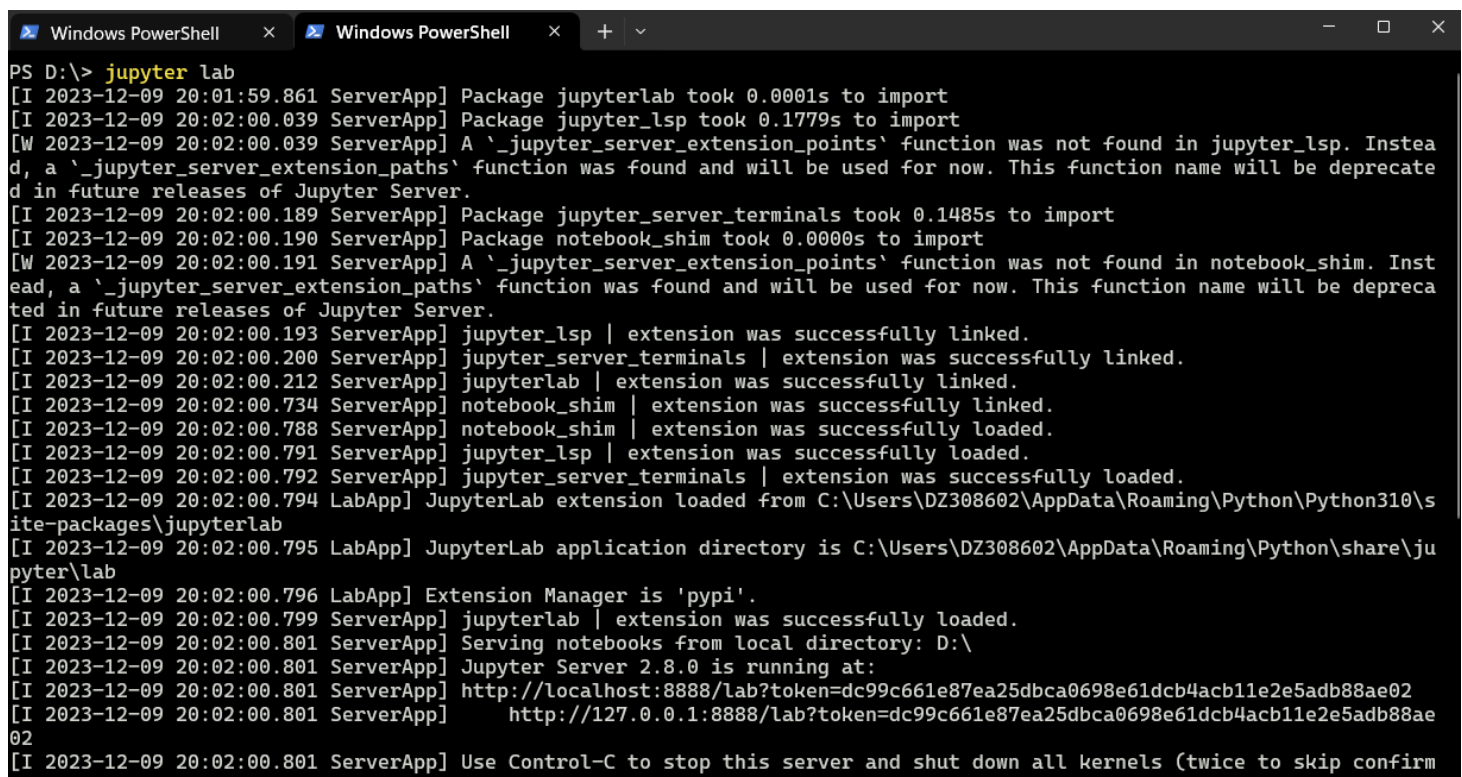
, which will initiate the jupyter server and automatically open a browser to the current working folder.

ii. Run the notebook directly within Visual Studio Code, utilising the Jupyter extension that supports notebook editing and previewing.

JupyterLab Notebook

[Top](#)

The following screenshot displays the command window output information while starting JupyterLab.



```
PS D:\> jupyter lab
[I 2023-12-09 20:01:59.861 ServerApp] Package jupyterlab took 0.0001s to import
[I 2023-12-09 20:02:00.039 ServerApp] Package jupyter_lsp took 0.1779s to import
[W 2023-12-09 20:02:00.039 ServerApp] A '_jupyter_server_extension_points' function was not found in jupyter_lsp. Instead, a '_jupyter_server_extension_paths' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-12-09 20:02:00.189 ServerApp] Package jupyter_server_terminals took 0.1485s to import
[I 2023-12-09 20:02:00.190 ServerApp] Package notebook_shim took 0.0000s to import
[W 2023-12-09 20:02:00.191 ServerApp] A '_jupyter_server_extension_points' function was not found in notebook_shim. Instead, a '_jupyter_server_extension_paths' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2023-12-09 20:02:00.193 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2023-12-09 20:02:00.200 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2023-12-09 20:02:00.212 ServerApp] jupyterlab | extension was successfully linked.
[I 2023-12-09 20:02:00.734 ServerApp] notebook_shim | extension was successfully linked.
[I 2023-12-09 20:02:00.788 ServerApp] notebook_shim | extension was successfully loaded.
[I 2023-12-09 20:02:00.791 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2023-12-09 20:02:00.792 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2023-12-09 20:02:00.794 LabApp] JupyterLab extension loaded from C:\Users\DZ308602\AppData\Roaming\Python\Python310\site-packages\jupyterlab
[I 2023-12-09 20:02:00.795 LabApp] JupyterLab application directory is C:\Users\DZ308602\AppData\Roaming\Python\share\jupyter\lab
[I 2023-12-09 20:02:00.796 LabApp] Extension Manager is 'pypi'.
[I 2023-12-09 20:02:00.799 ServerApp] jupyterlab | extension was successfully loaded.
[I 2023-12-09 20:02:00.801 ServerApp] Serving notebooks from local directory: D:\
[I 2023-12-09 20:02:00.801 ServerApp] Jupyter Server 2.8.0 is running at:
[I 2023-12-09 20:02:00.801 ServerApp] http://localhost:8888/lab?token=dc99c661e87ea25dbca0698e61dcb4acb11e2e5adb88ae02
[I 2023-12-09 20:02:00.801 ServerApp] http://127.0.0.1:8888/lab?token=dc99c661e87ea25dbca0698e61dcb4acb11e2e5adb88ae02
[I 2023-12-09 20:02:00.801 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirm
```

Fig. 1: Starting JupyterLab in Command Window

If there are no issues during this process, a new browser tab will open automatically, displaying the GUI.

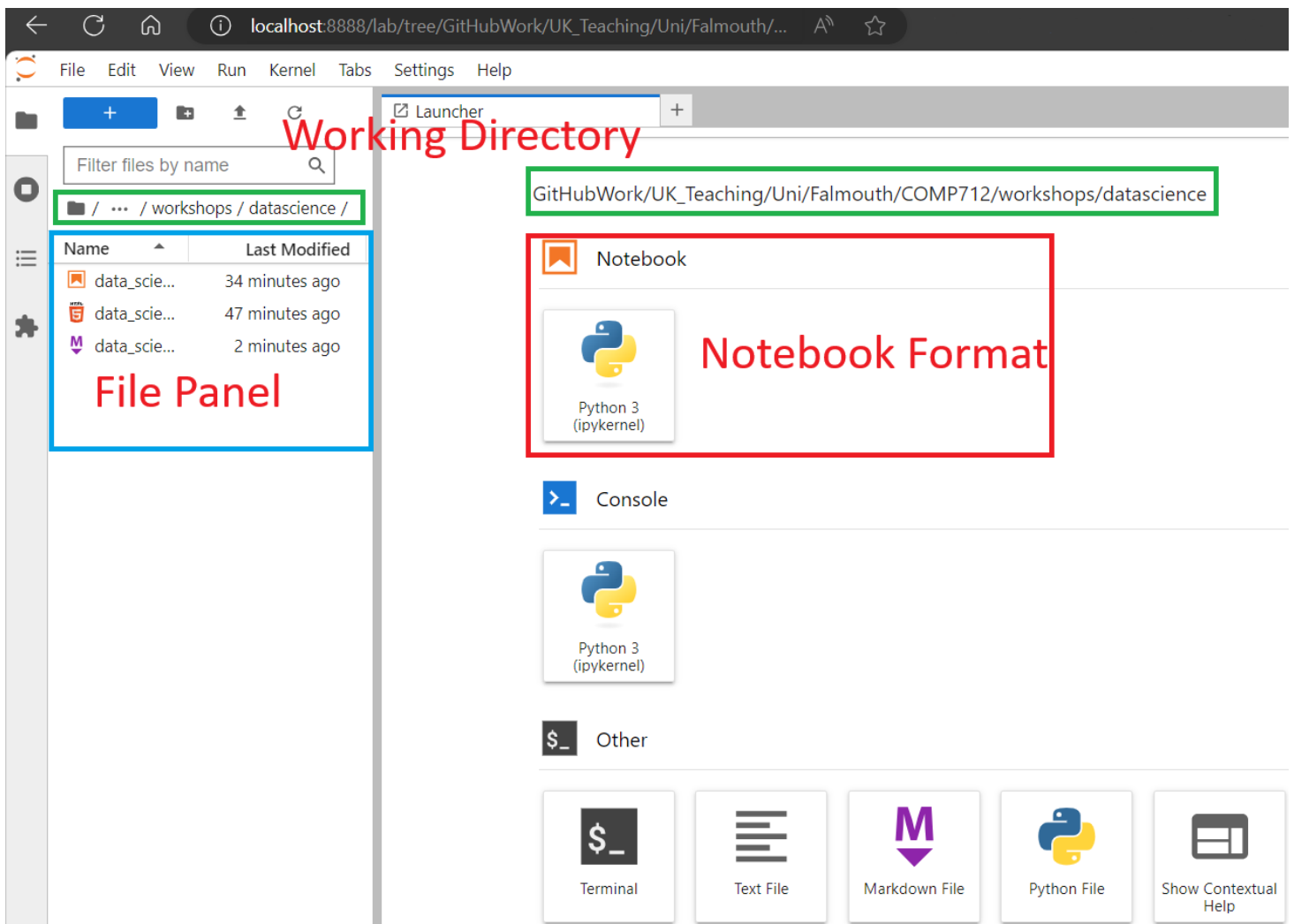


Fig. 2: JupyterLab GUI

As depicted in the screenshot above, the file panel lists all the files in your current working directory. The main area on the right provides options to create various file types, including notebook , console python file , pure text file , markdown , and others. For this session, you will work with notebook .

You don't need to create a new notebook manually as a template has been provided in the repository of this workshop.

The Repository

[Top](#)

The repository below contains the Jupyter Notebook with examples and exercises for this session.

Fork the repository (NOT clone!) and work on your fork. This will enable you to submit a pull request at the end (if needed).

<https://github.falmouth.ac.uk/Daniel-Zhang/COMP712-DataScience.git>

The Notebook

[Top](#)

There's only one `.ipynb` file in the repository, which is self-explanatory. Now, switch to the notebook using one of the methods introduced above.

Note:

You don't need to submit any code for this workshop