Recreating Virtual Environments

Setting up a clean and organized Python environment is essential when working on different projects, especially in data science where package versions can affect results. This guide walks you through different ways of setting up virtual environments using a requirements.txt file, and how to activate and use these environments whether you're running Python scripts, working in Jupyter Notebooks via VS Code, or using Jupyter in a web browser.

Step 1: Open Terminal and Navigate to the Desired Path

Before creating the environment, open a terminal and navigate to the directory where your requirements.txt is located. Example:

Step 2: Create a Virtual Environment

Choose an easy-to-remember name for the environment (we'll use it later). In the example below, the environment name is sampleEnv:

```
python -m venv sampleEnv
```

Step 3: Activate the Environment

Windows:

```
.\ sampleEnv\Scripts\activate
```

macOS/Linux:

```
source sampleEnv/bin/activate
```

Step 4: Install dependencies:

First, make sure pip is updated:

```
python.exe -m pip install --upgrade pip
```

Then install packages from the requirements file:

```
pip install -r requirements.txt
```

Using Virtual Environments Across Platforms

🚀 Method 1: Running .py Files with the Environment

Use Case: Running a .py file through the terminal.

You can run Python scripts in terminal using the activated environment:

```
python your_script.py
```

Method 2: Jupyter Notebook via VS Code Kernel

Use Case: Running a .ipynb notebook inside VS Code.

Step 2.0: Install Additional Tools

(if not already in requirements.txt)

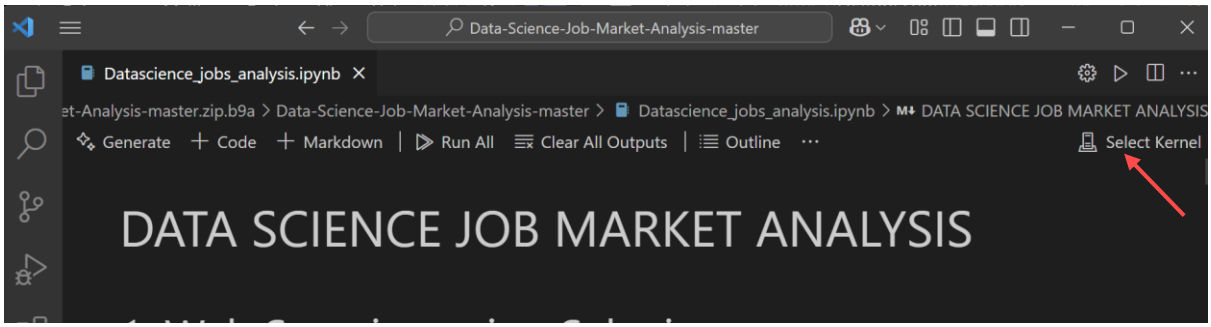
```
pip install notebook ipykernel
```

Step 2.1: Add the Environment to Jupyter Kernel

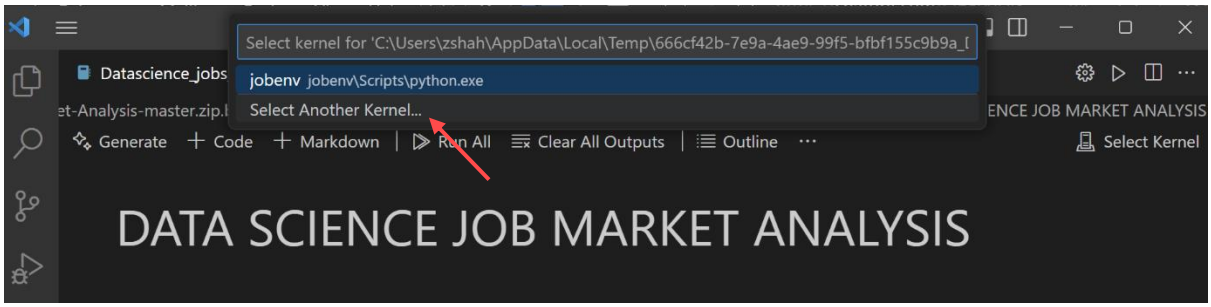
```
python -m ipykernel install --user --name= sampleEnv --
display-name " sampleEnv "
```

Step 2.2: Open your notebook in VS Code.

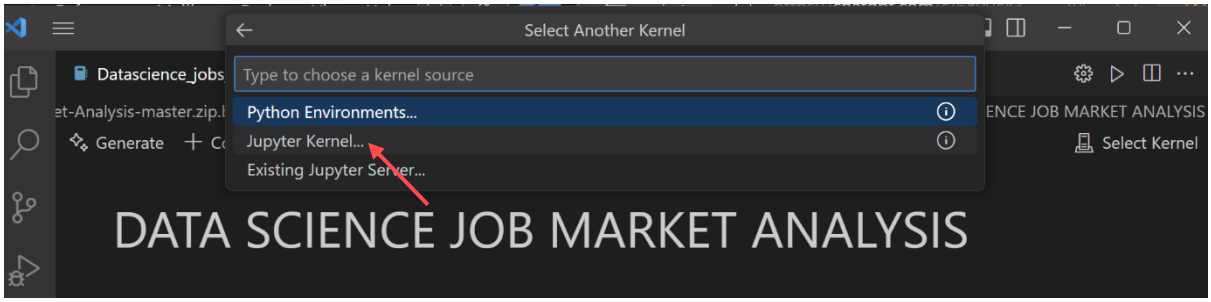
From the top-right corner of the notebook, select the kernel dropdown.



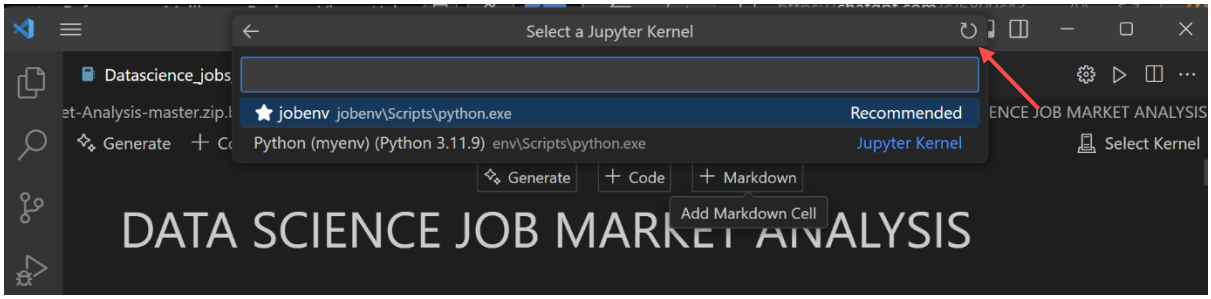
Step 2.3: Choose "Select Another Kernel"



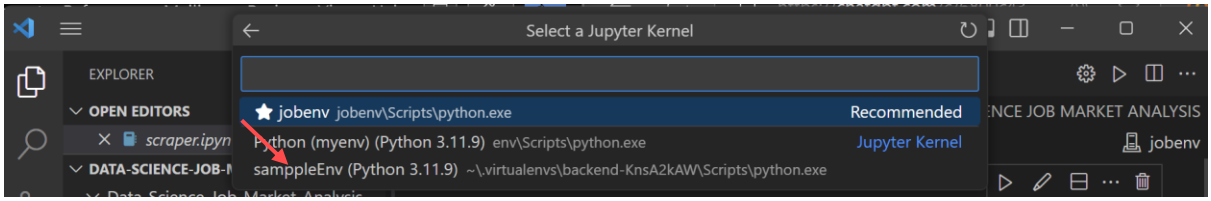
Step 2.4: Choose "Jupyter Kernel"

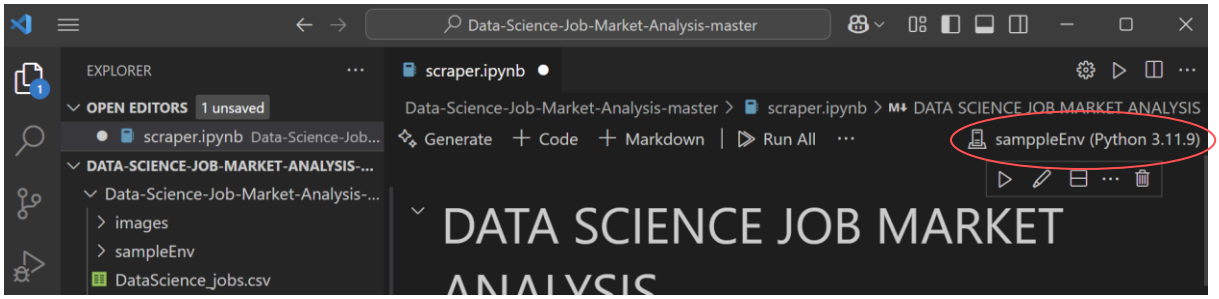


Step 2.5: Refresh if you don't see sampleEnv



Step 2.6: Select sampleEnv





You can now run your notebook using the environment.

Step 2.7 (Optional): Install Extra Packages in Notebook

```
!pip install package_name
```

Method 3: Jupyter Notebook in Web Browser (JupyterLab or Classic Jupyter)

Use Case: Running notebooks in the browser (e.g. via jupyter notebook or jupyter lab).

Step 3.1: Ensure Jupyter is Installed

If not already installed:

```
pip install notebook ipykernel
```

Step 3.2: Launch Jupyter

From the activated environment, run:

```
jupyter notebook
```

Or

```
jupyter lab
```

This will open Jupyter in your default web browser.

Step 3.3: Navige to the path

navigate to the directory where your project is located.

Step 3.4: Select Your Kernel

- Open any .ipynb file.

- Use the Kernel > Change Kernel menu to select sampleEnv.

Step 3.5:

You're now running the notebook using your environment.