Tutorial

An Automated Topological Framework for Robust Multi-Scale Data Analysis Using Persistent Homology and Mapper Algorithms

Naseem Zoha Ansari[1], Subhajit Bandyopadhyay[2],Sudarshan Gogoi[1], Mubashar Rashid[3],Amit Chakraborty[*1]

[1]Department of Mathematics, Sikkim University, Gangtok, 737102, Sikkim, India.
[2]Department of Computer Applications, Sikkim University, Gangtok, 737102, Sikkim, India
[3]Department of Mathematics, Indian Institute of Technology, Kanpur, India.

Step 1 : Open Anaconda Navigator

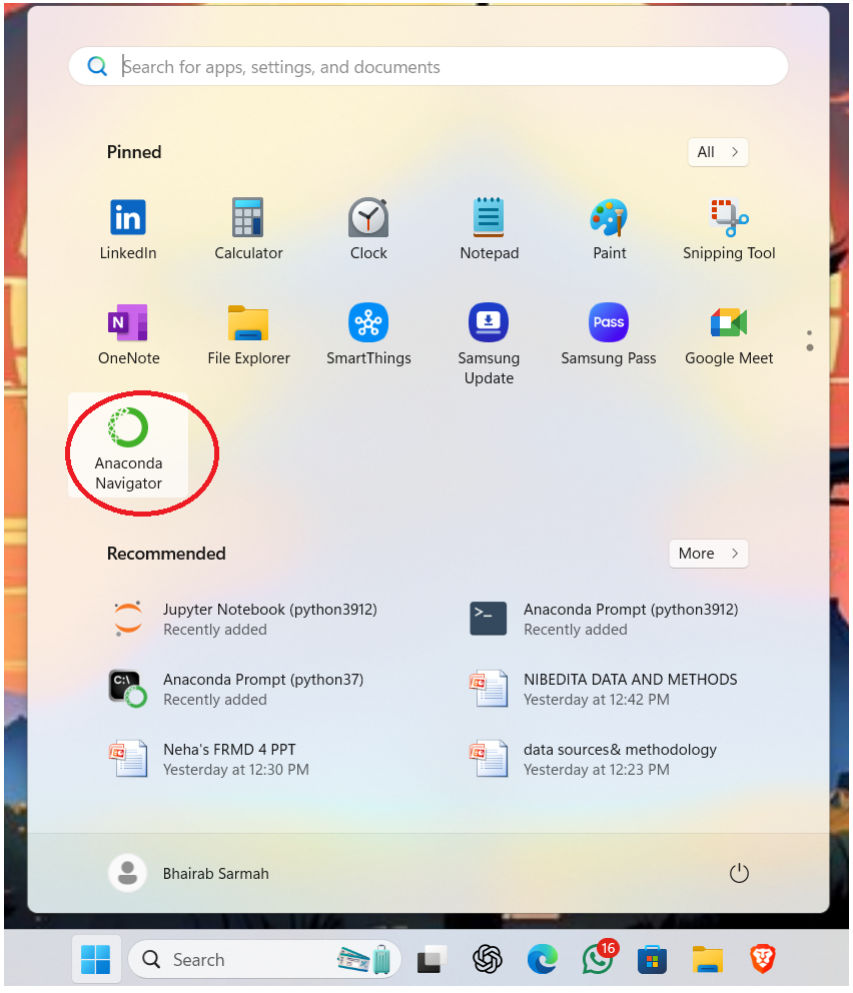

Figure 1: Select the Anaconda Navigator file from the destination or in Strat menu

Step 2 : Selecting Python 3.9.21 for our program use

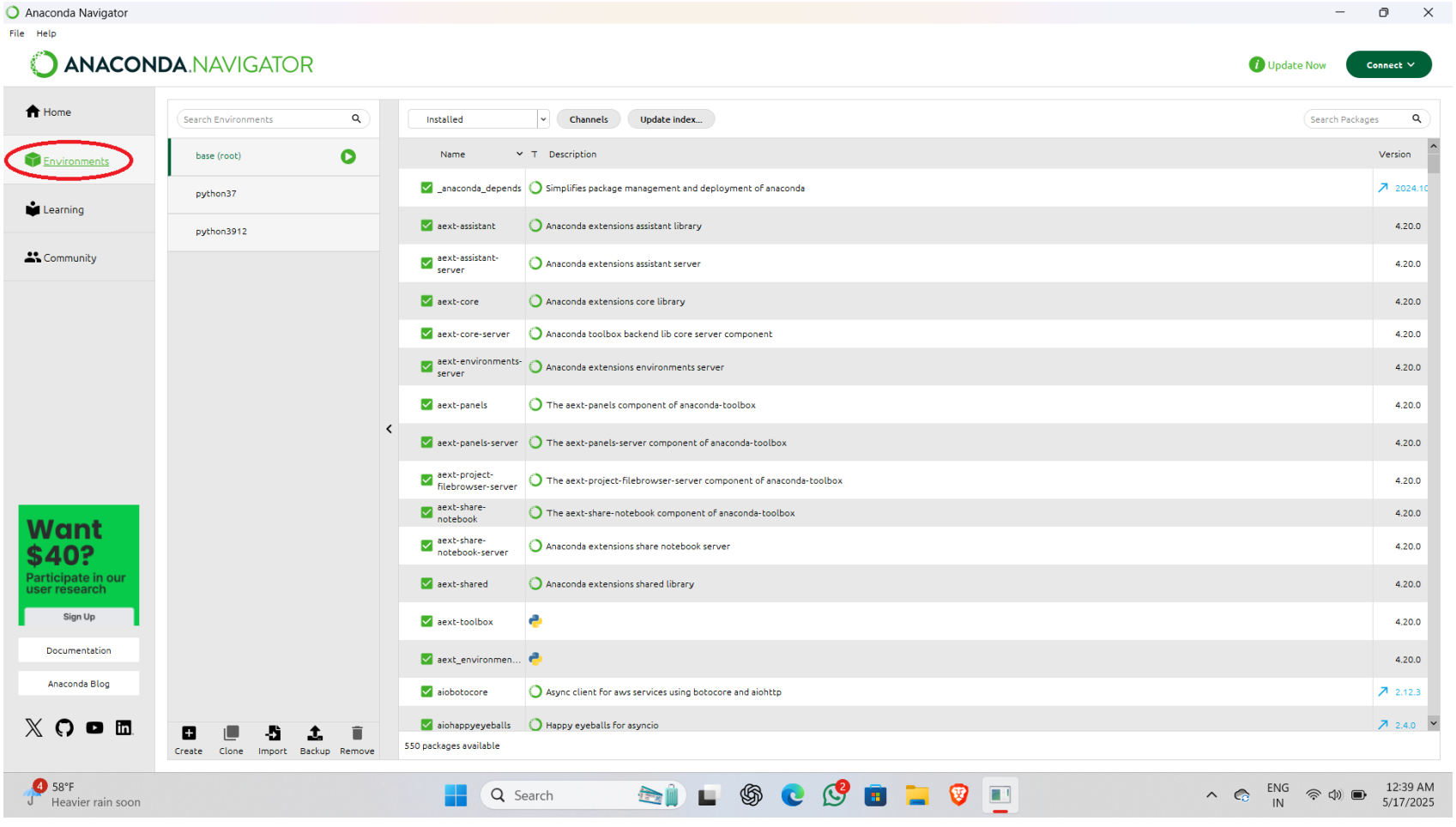

Figure 2: Click on the Environment tab just below Home tab

Step 2 : Selecting Python 3.9.21 for our program use

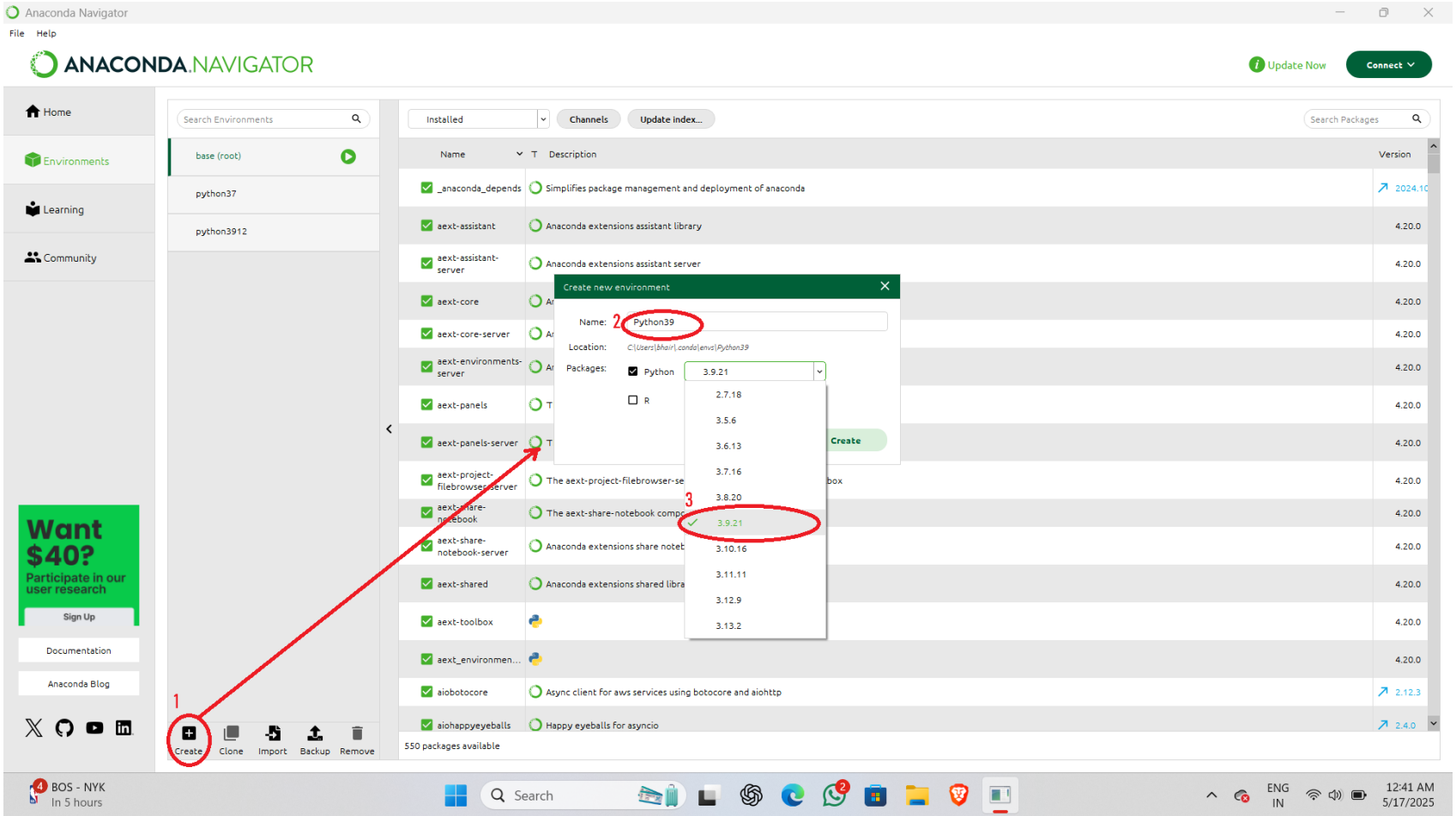

Figure 3: Click Create, name your environment in the dialog box that appears, select Python 3.9.21

Step 2 : Selecting Python 3.9.21 for our program use

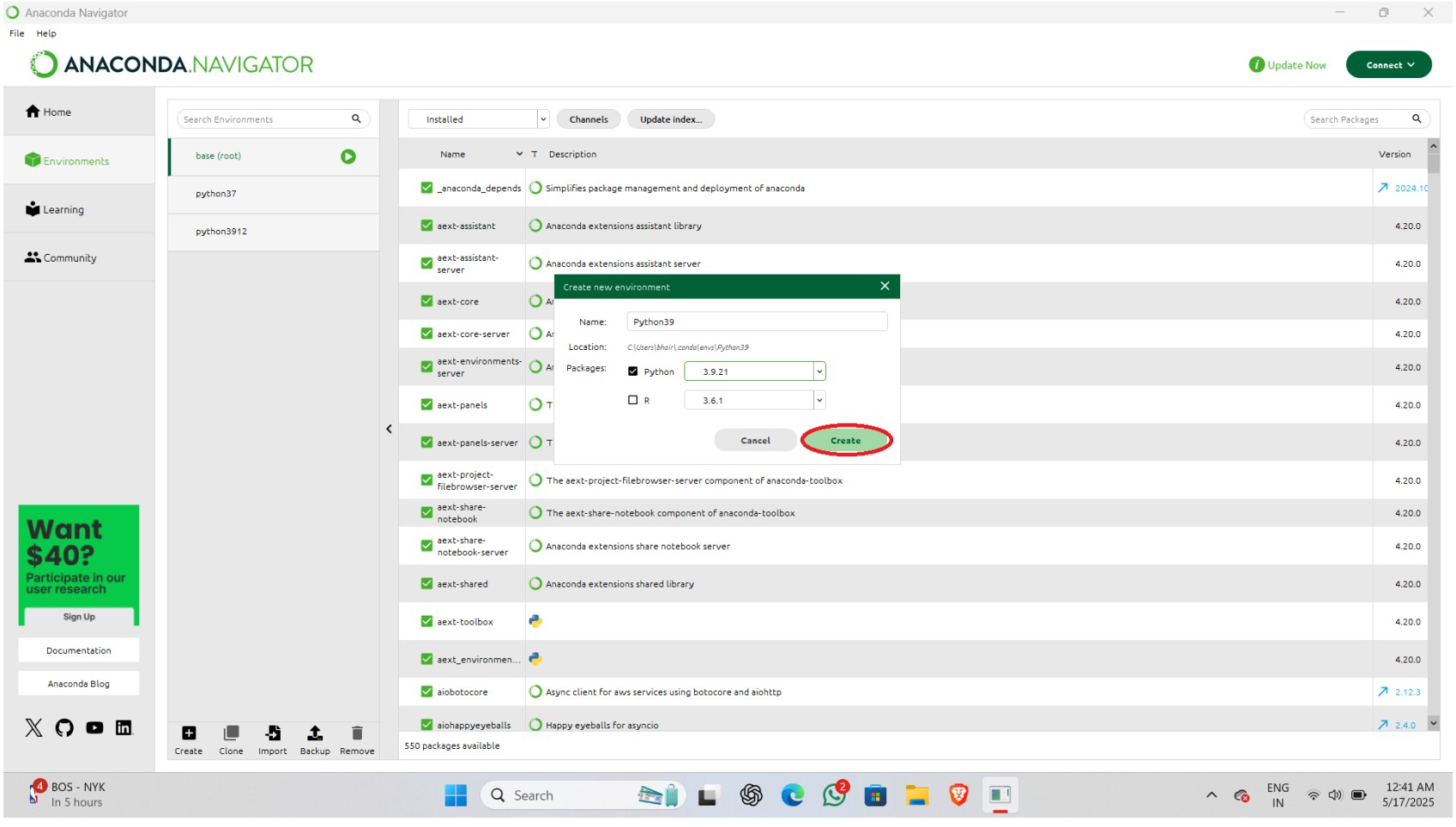

Figure 4: Hit on create button to confirm

Step 2 : Selecting Python 3.9.21 for our program use

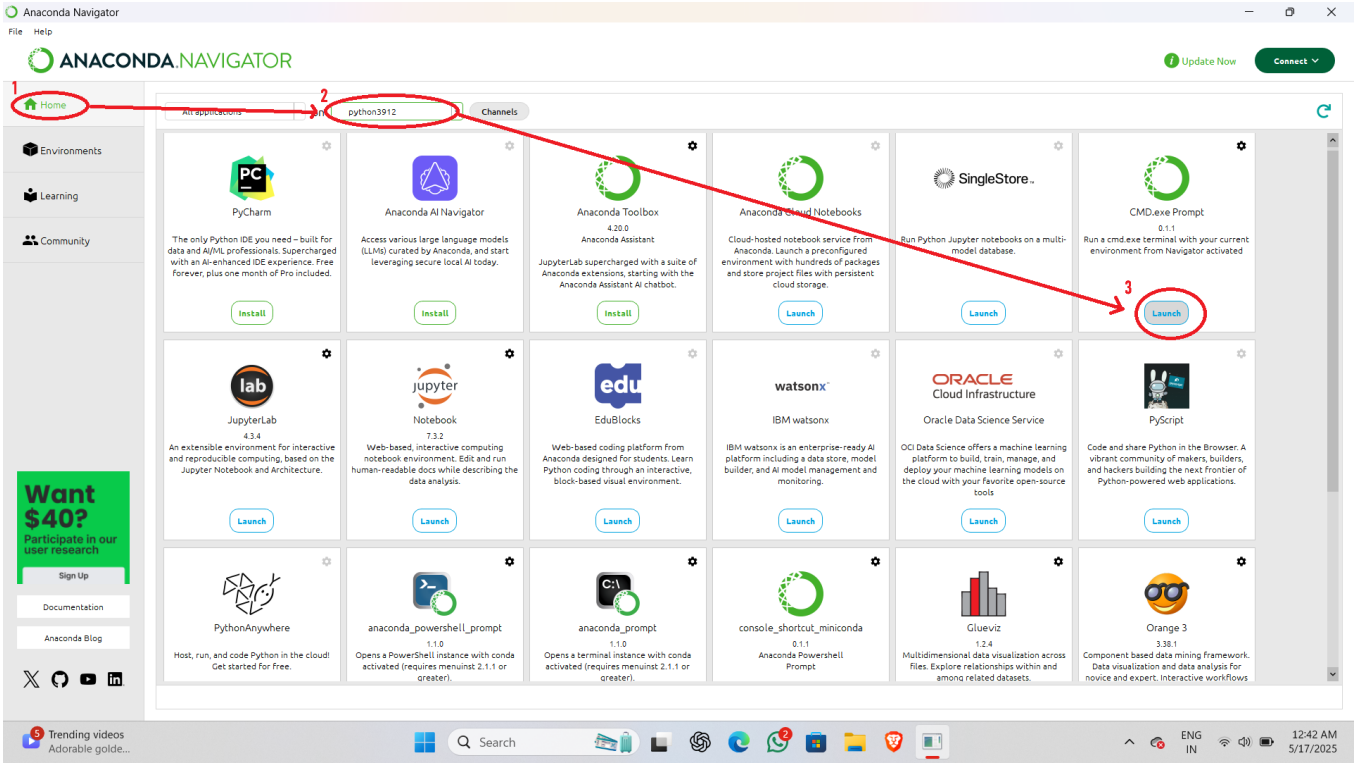

Figure 5: Navigate to the **Home** tab, verify the created environment under the *Applications on* section, and launch `CMD.exe` to install the required packages.

Step 3 : Installation of the required packages

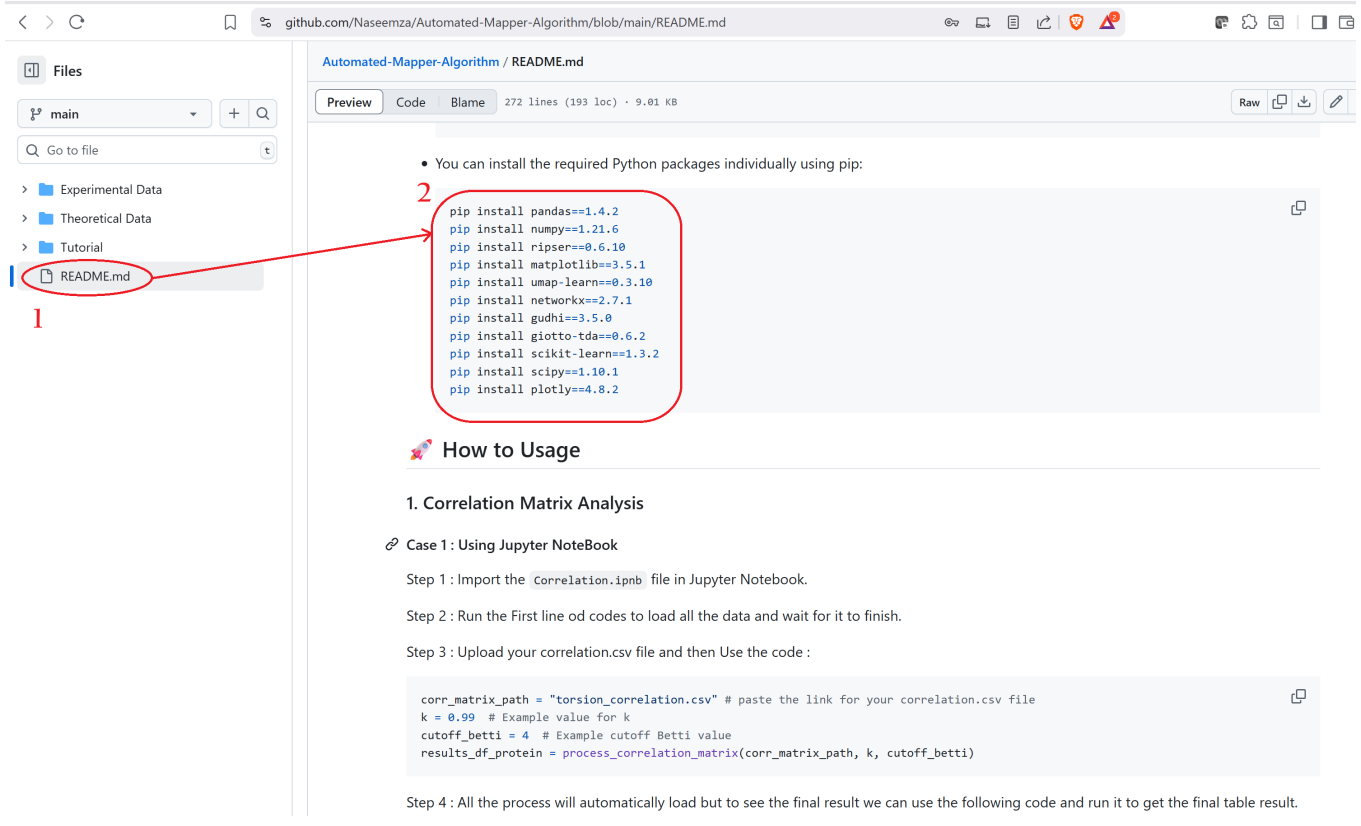

Figure 6: Visit the GitHub repository `https://github.com/Naseemza/Advance-Mapper-Algorithm`, open the `README.md` file to identify the required package name, and use the `cmd.exe` dialog box to install all compatible packages using `pip`.

Step 3 : Installation of the required packages

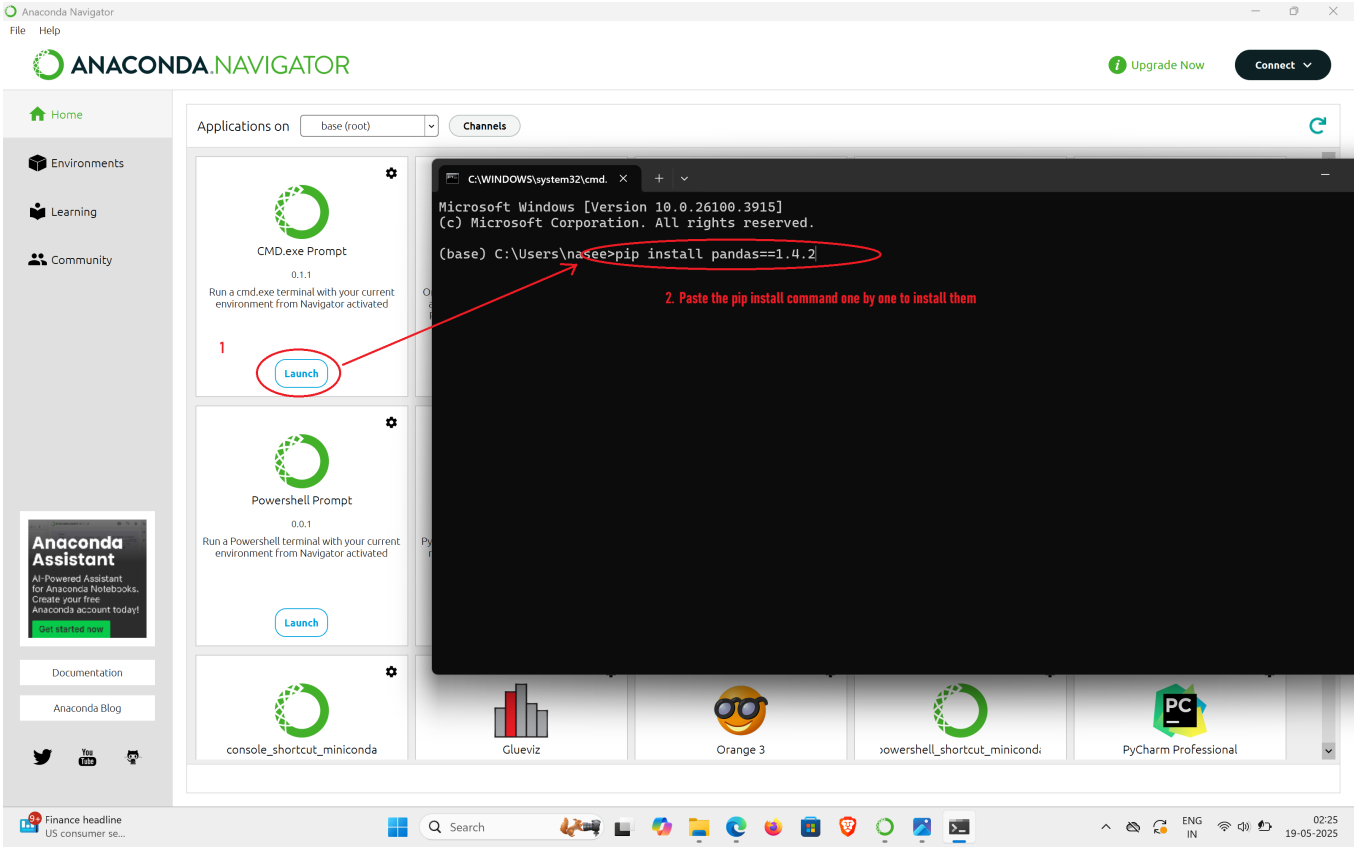

Figure 7: Open Anaconda Navigator, launch the Command Prompt (CMD.exe), and sequentially execute each pip command to install the required packages.

Step 1 : Download the Torus.ipynb file

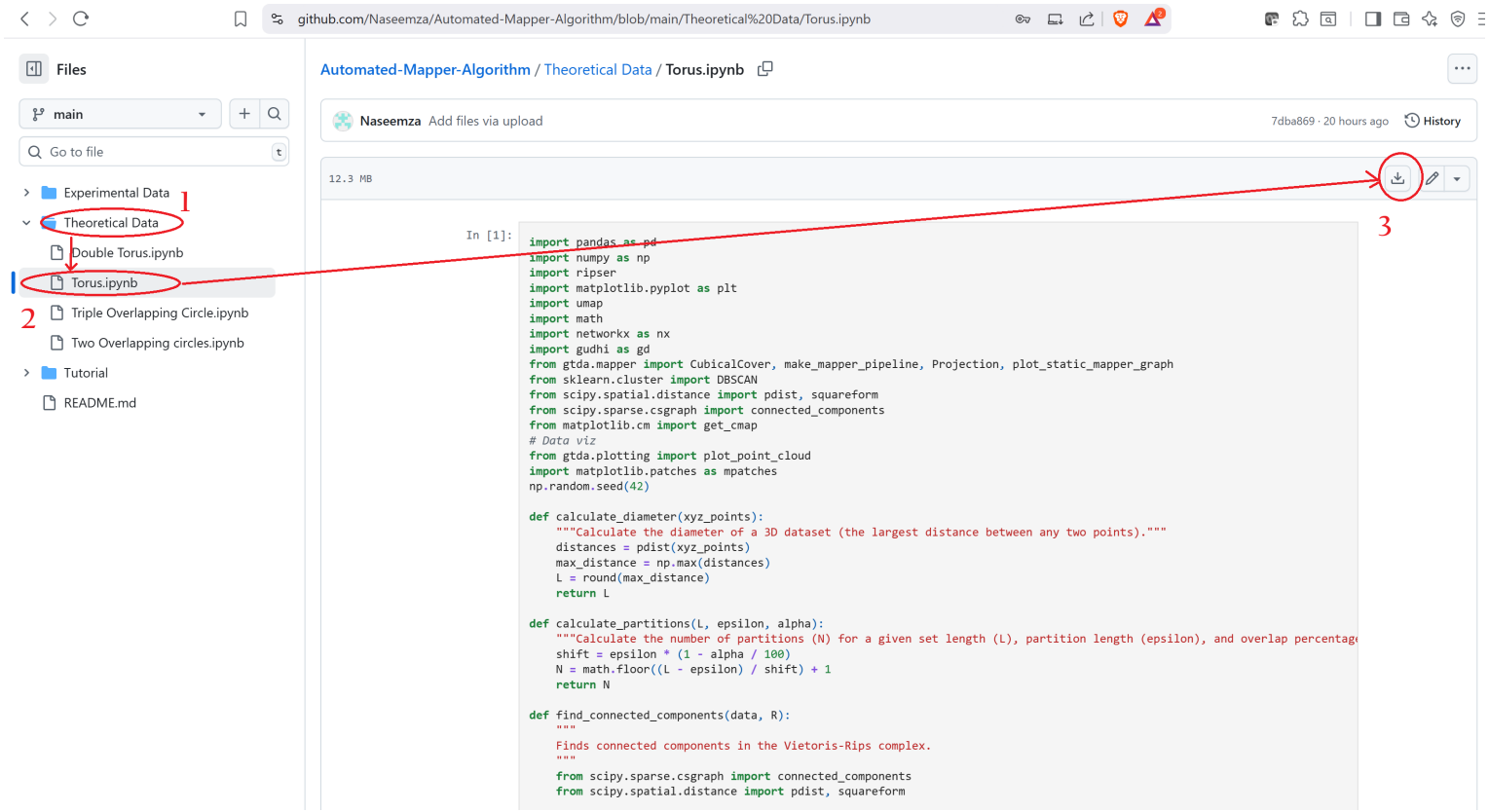

Figure 8: Click on the **Theoretical Data** section, open the `Torus.ipynb` file (or any other), and click **Download** to save it from https://github.com/Naseemza/Advance-Mapper-Algorithm.

Step 1 : Download the Torus.ipynb file

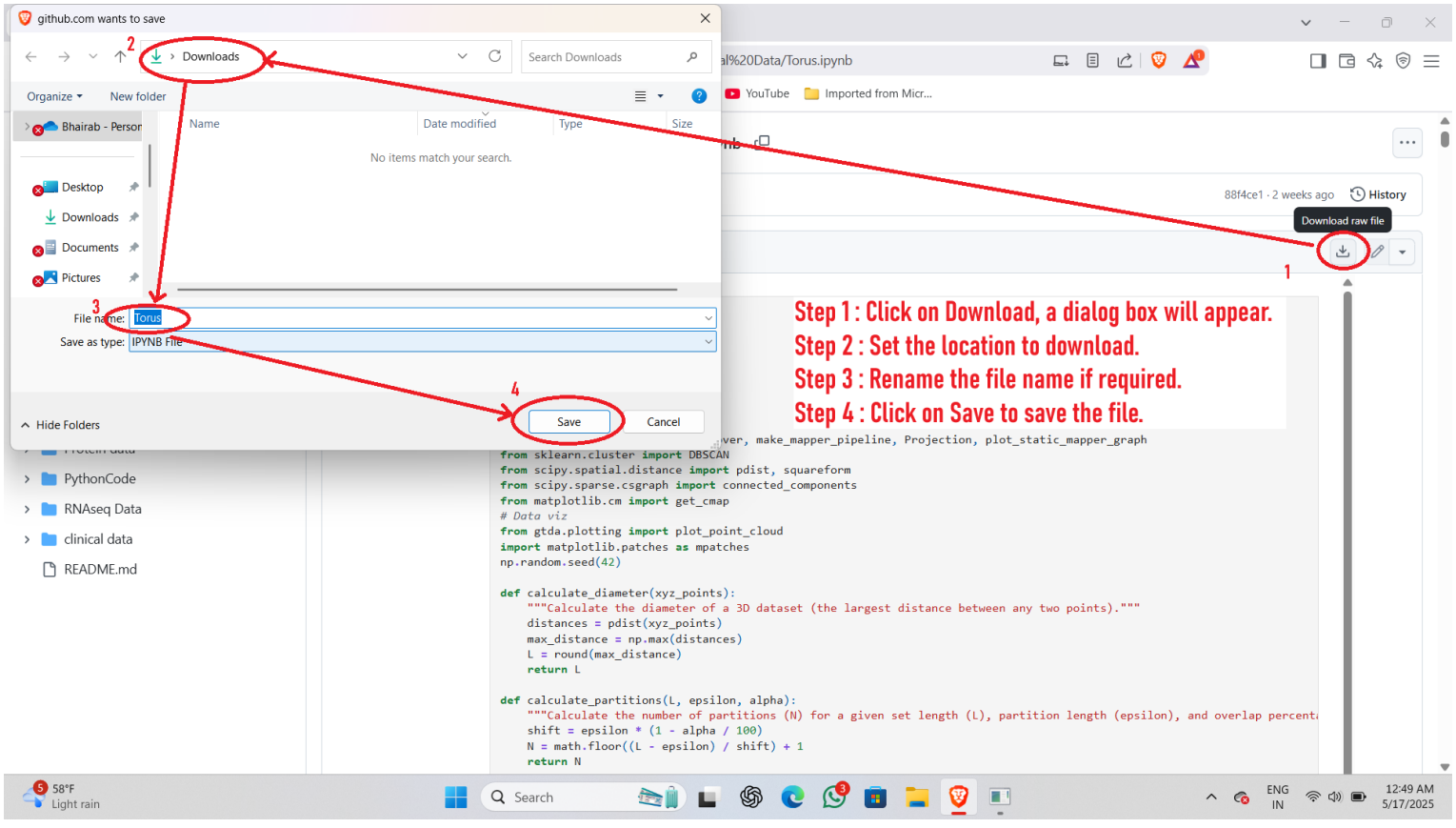

Figure 9: The process for downloading a file: click, locate, rename and save

Step 2 : Uploading the .ipynb file

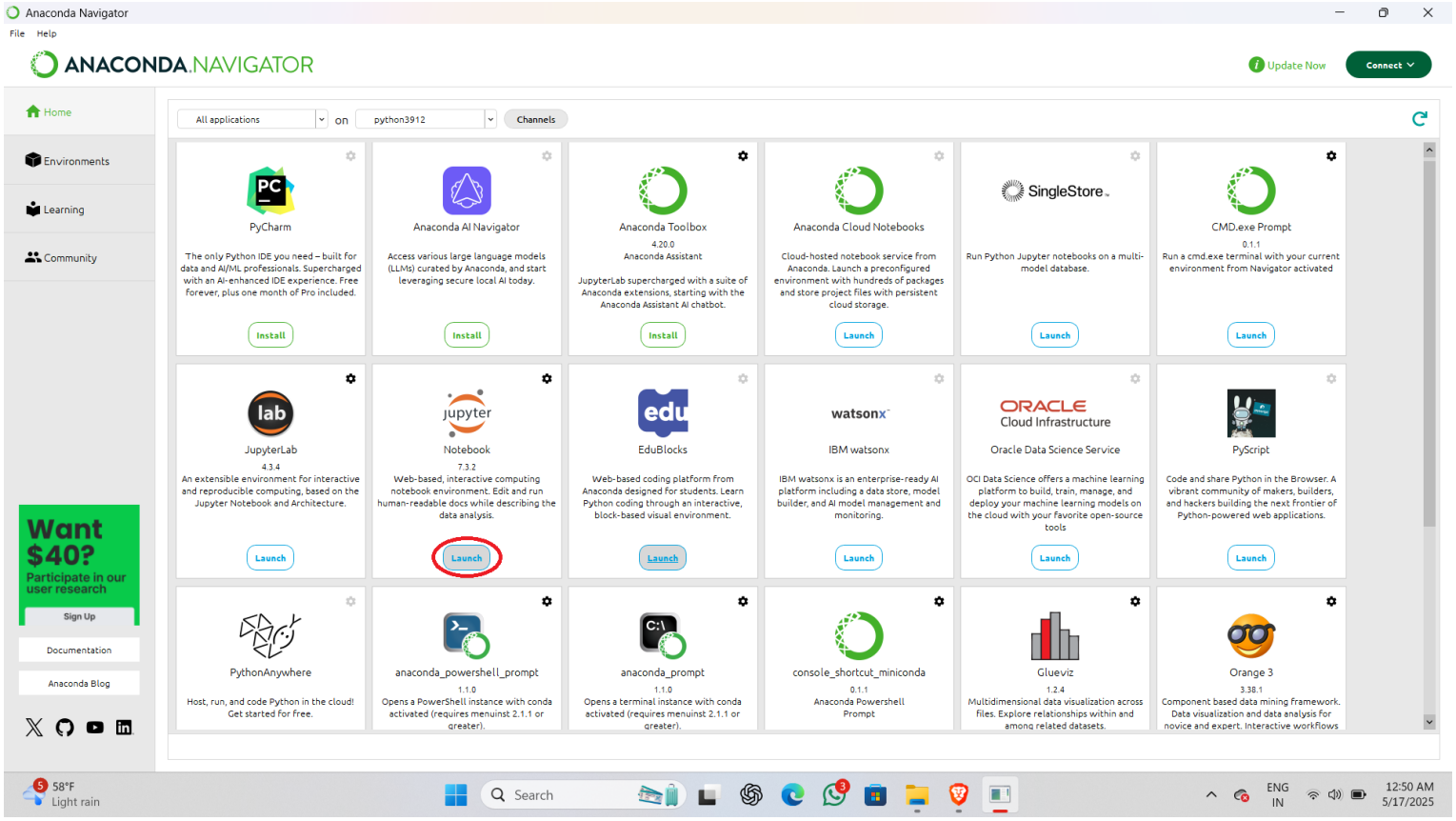

Figure 10: Open Jupyter Notebook and wait for it to load from Anaconda Navigator application

Step 2 : Uploading the .ipynb file

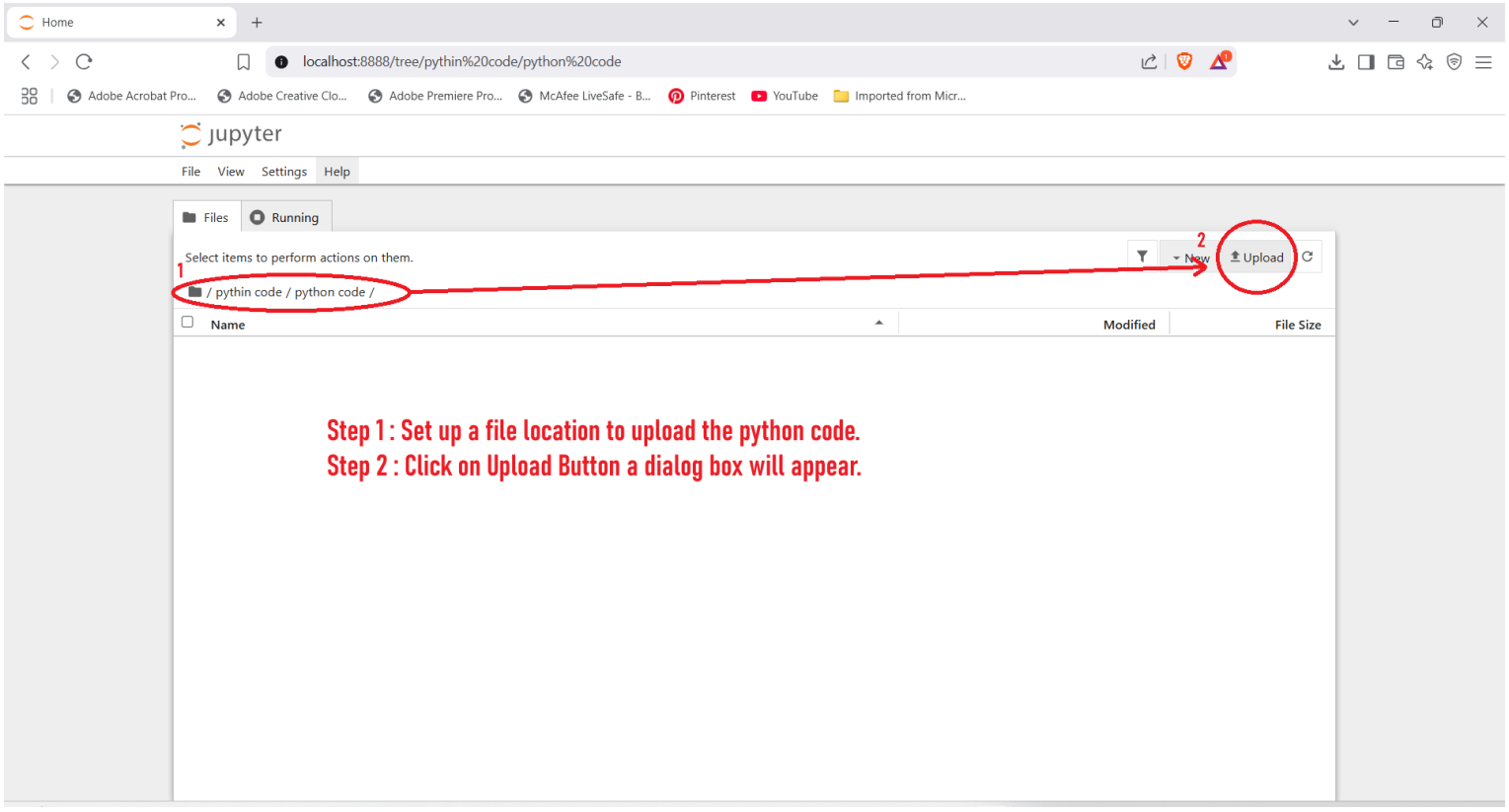

Figure 11: In the **Home** tab of the browser, navigate to the desired location, then click **Upload** to select and upload the previously downloaded file.

Step 3 : Lunch the .ipynb file

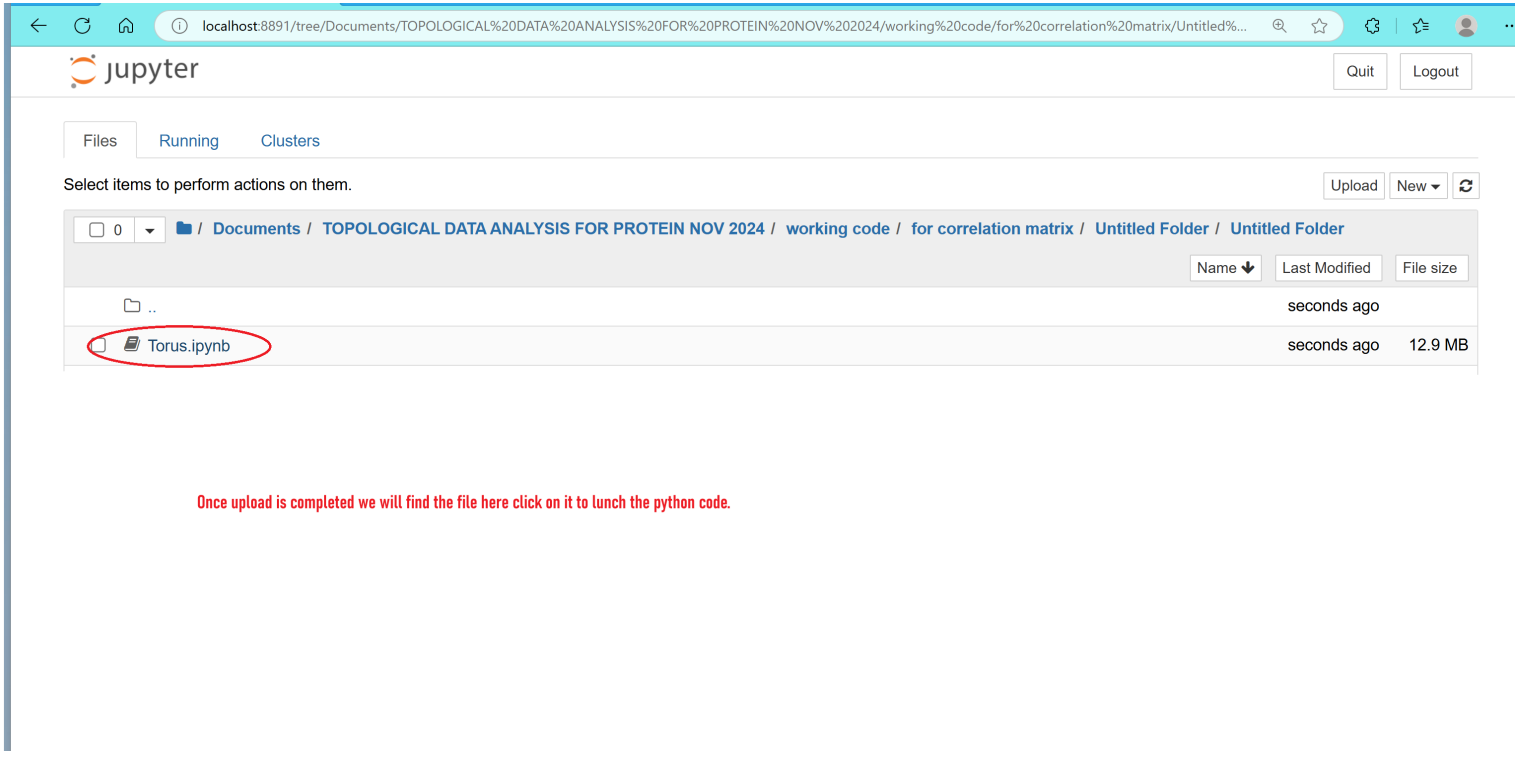

Figure 12: Click on the uploaded file to lunch the python code

Step 4 : Run the python code

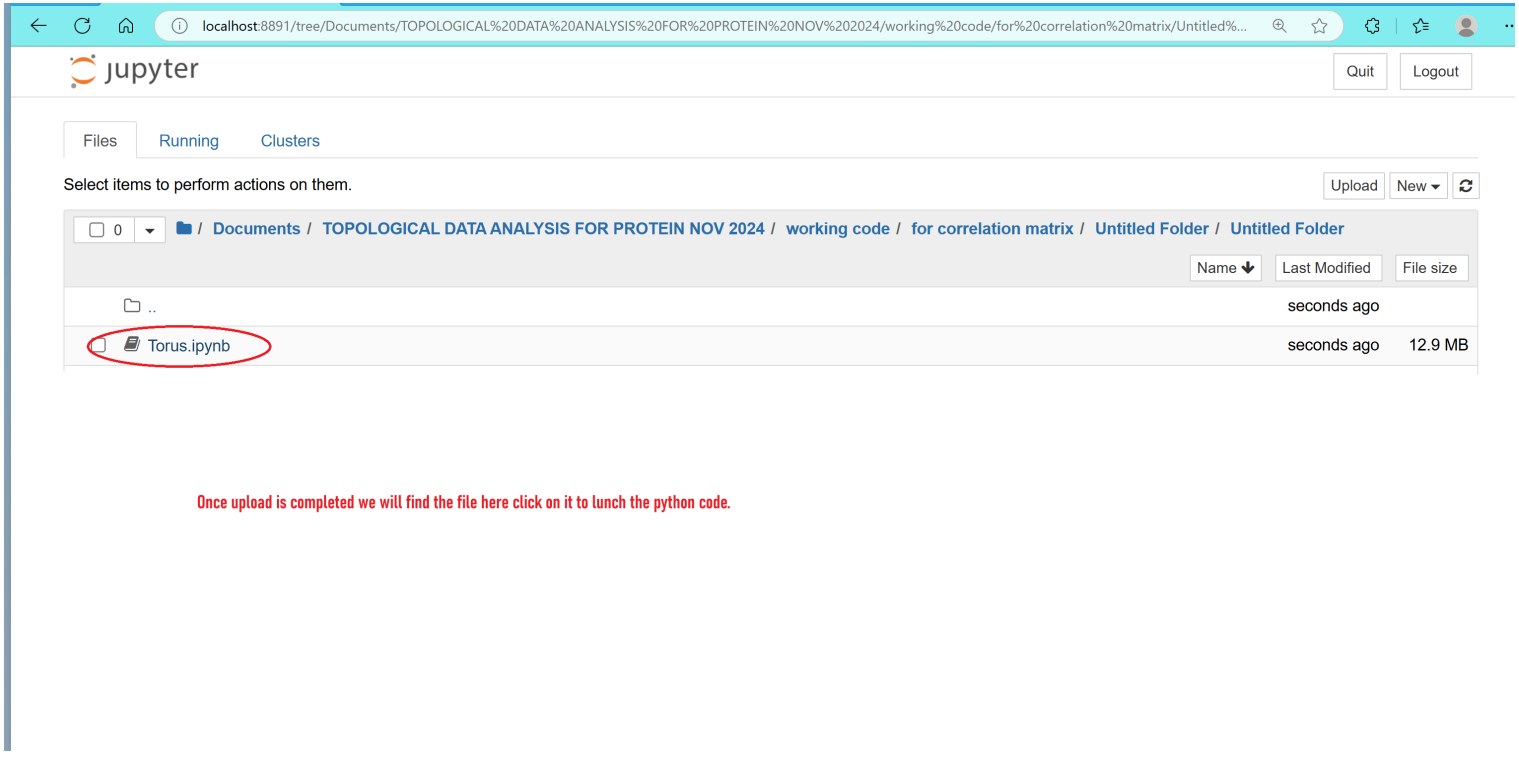

Figure 13: Click on the uploaded file to lunch the python code

Step 4 : Run the python code

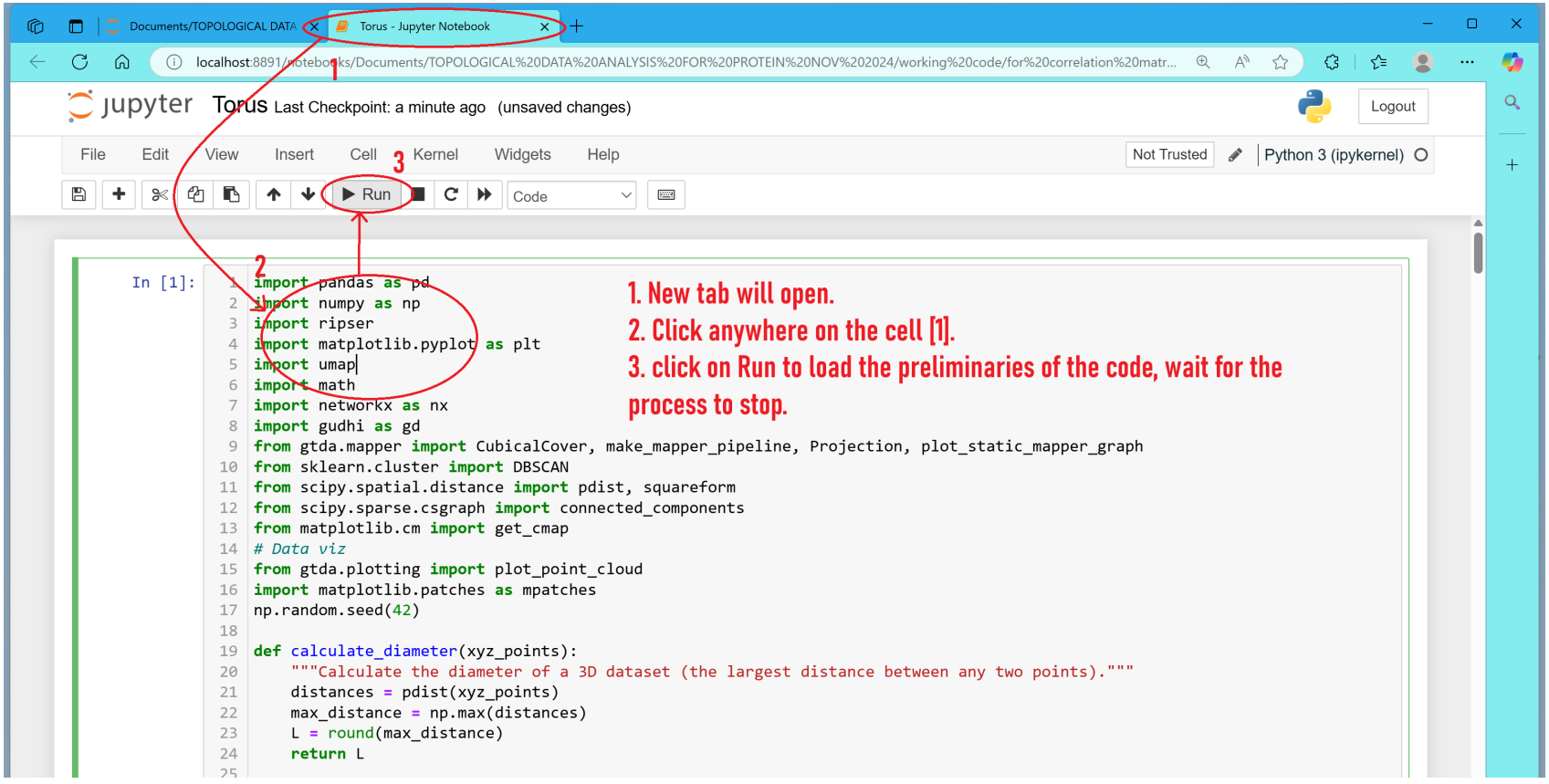

Figure 14: **A new tab will open. Click on cell [1], then click `Run` to load the preliminaries of the code. Wait for the process to complete before proceeding.**

Step 4 : Run the python code

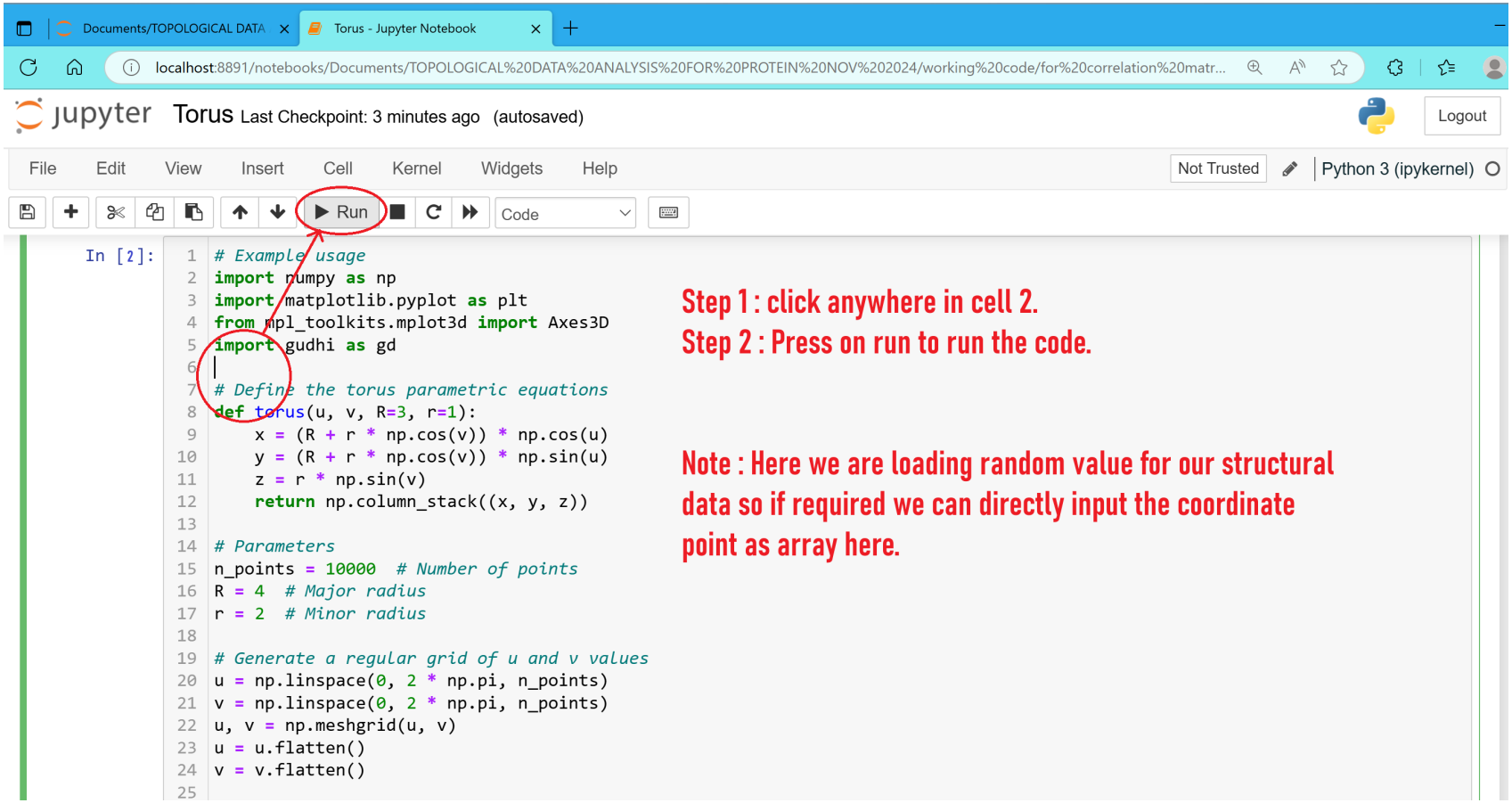

Figure 15: Follow thw process to load the coordinate data.

Step 4 : Run the python code

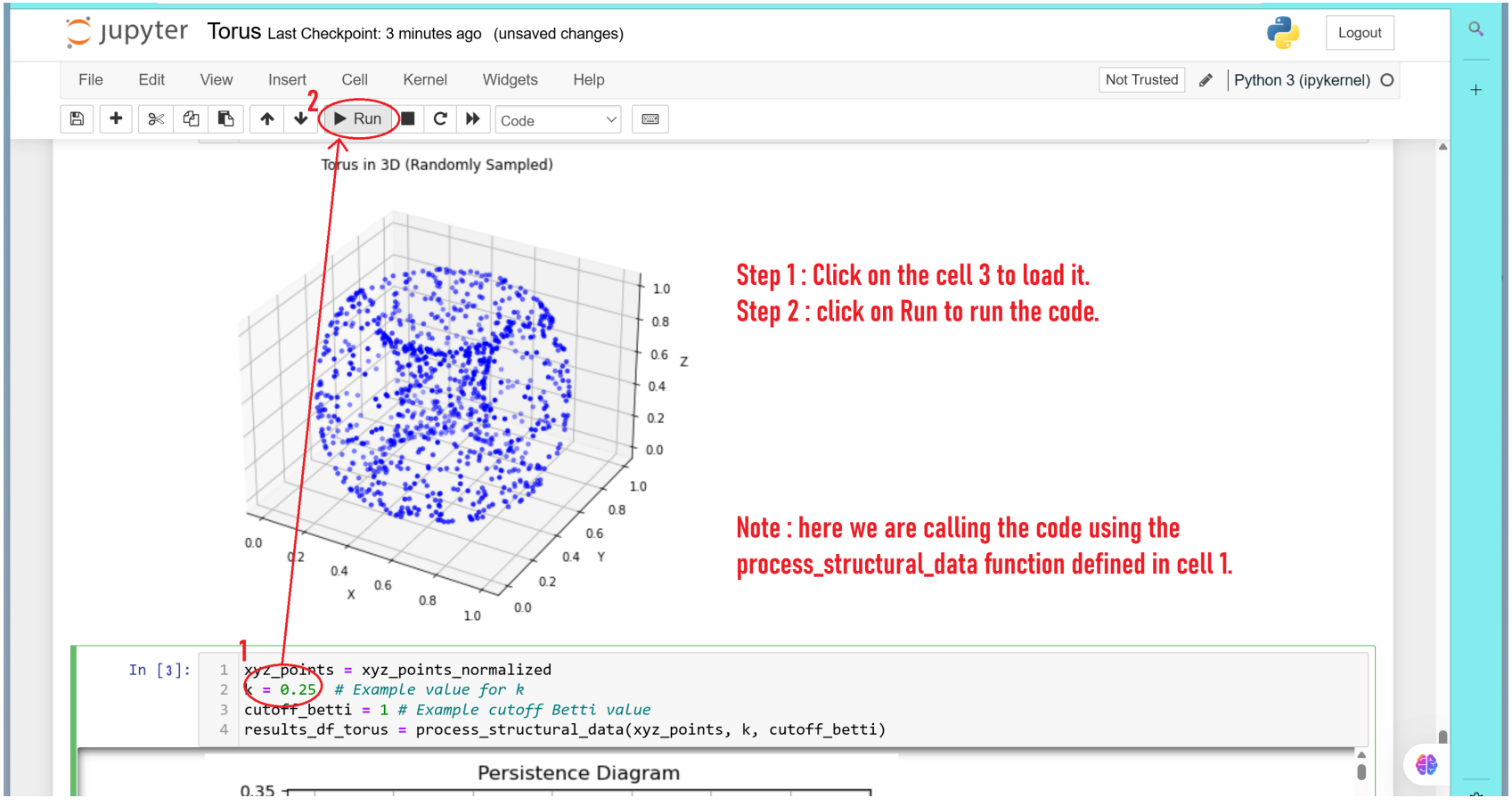

Figure 16: Follow thw process to run the code for your data.

Step 5 : Print thr final result

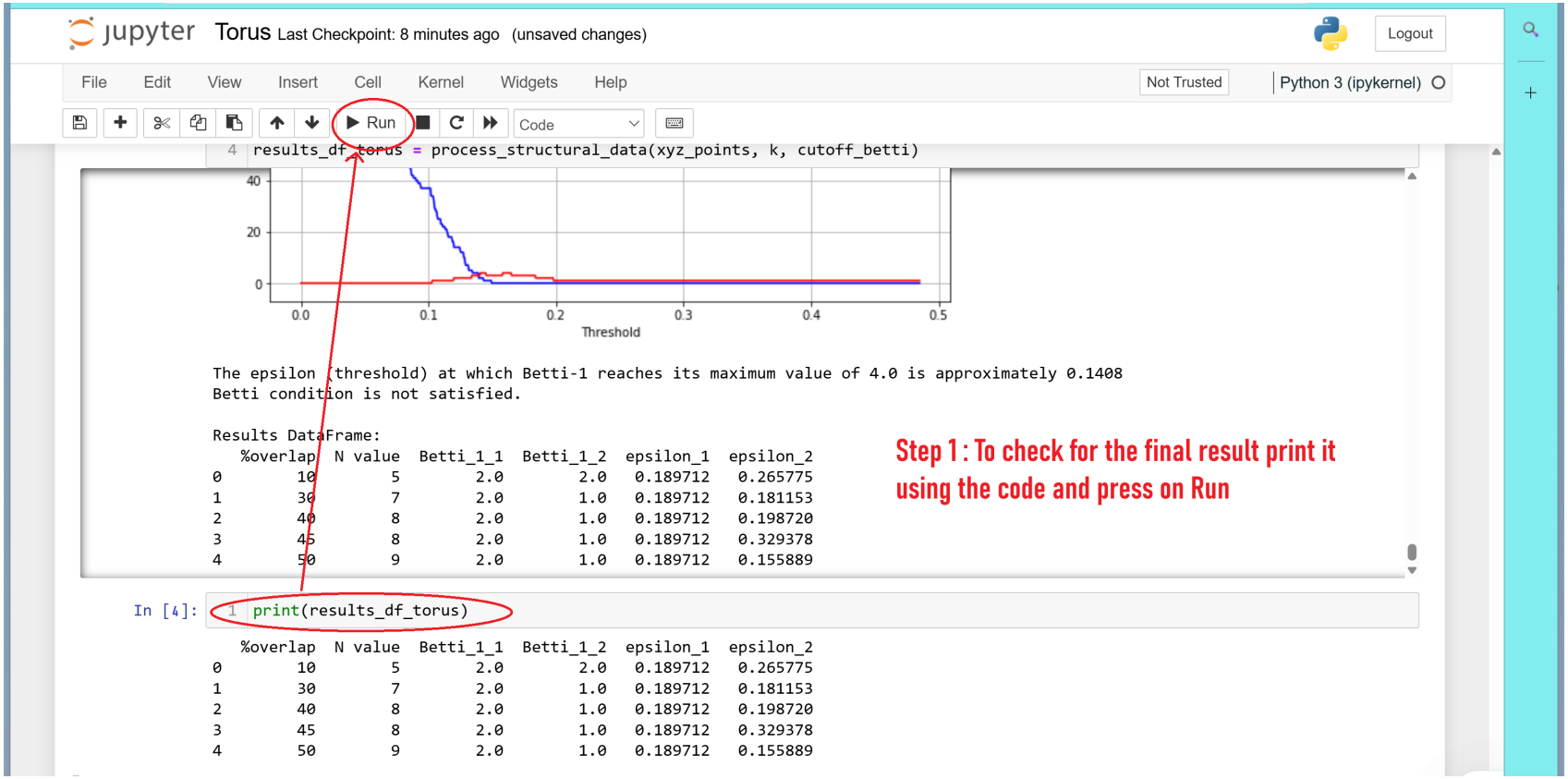

Figure 17: Follow the process to get the final result table.

Step 1 : Uploading the .ipynb and .csv file

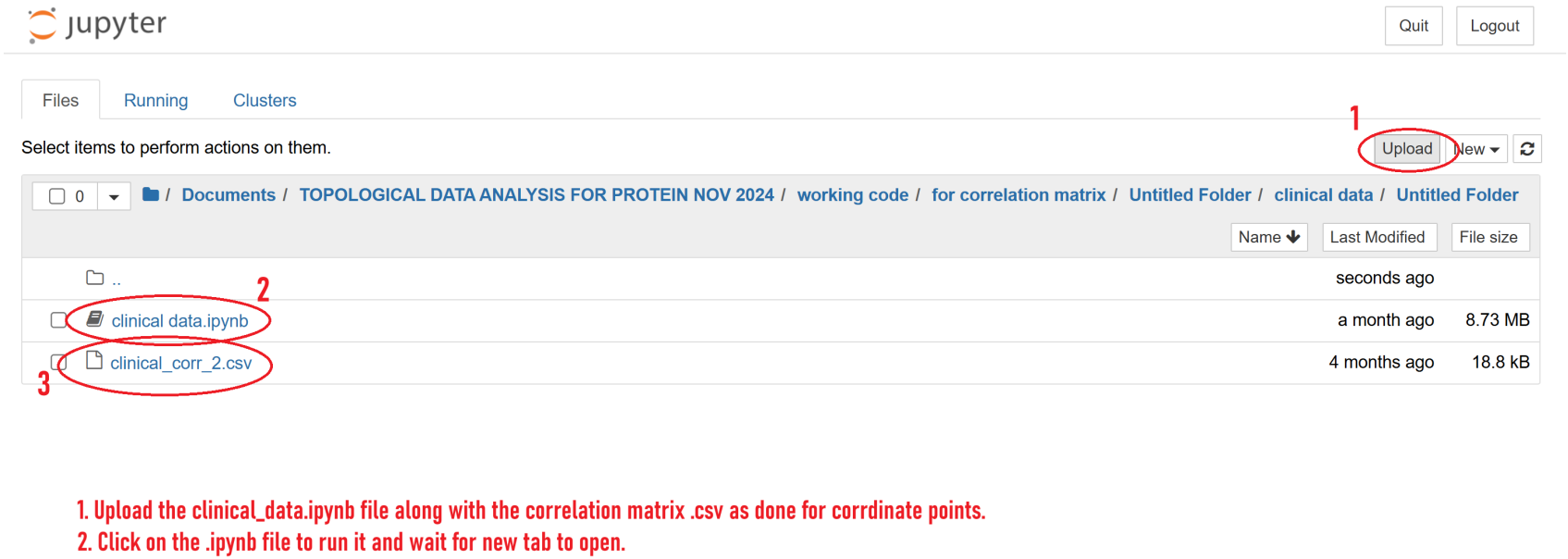

Figure 18: We follow the same steps of downloading the files and then similarly as coordinate files we upload the files here.

Step 2 : Run the code

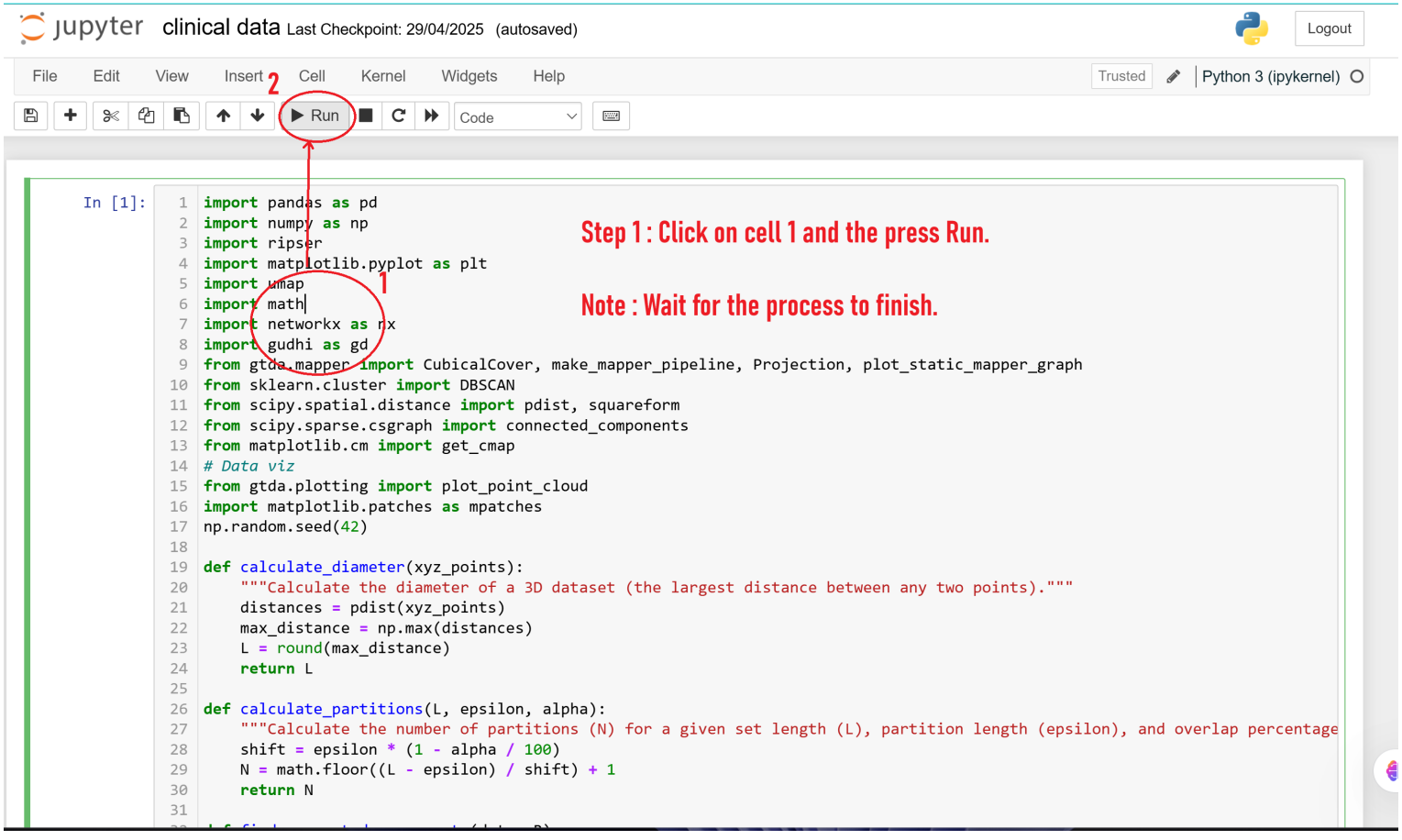

Figure 19: We run the code step by step

Step 3 : Print the final result

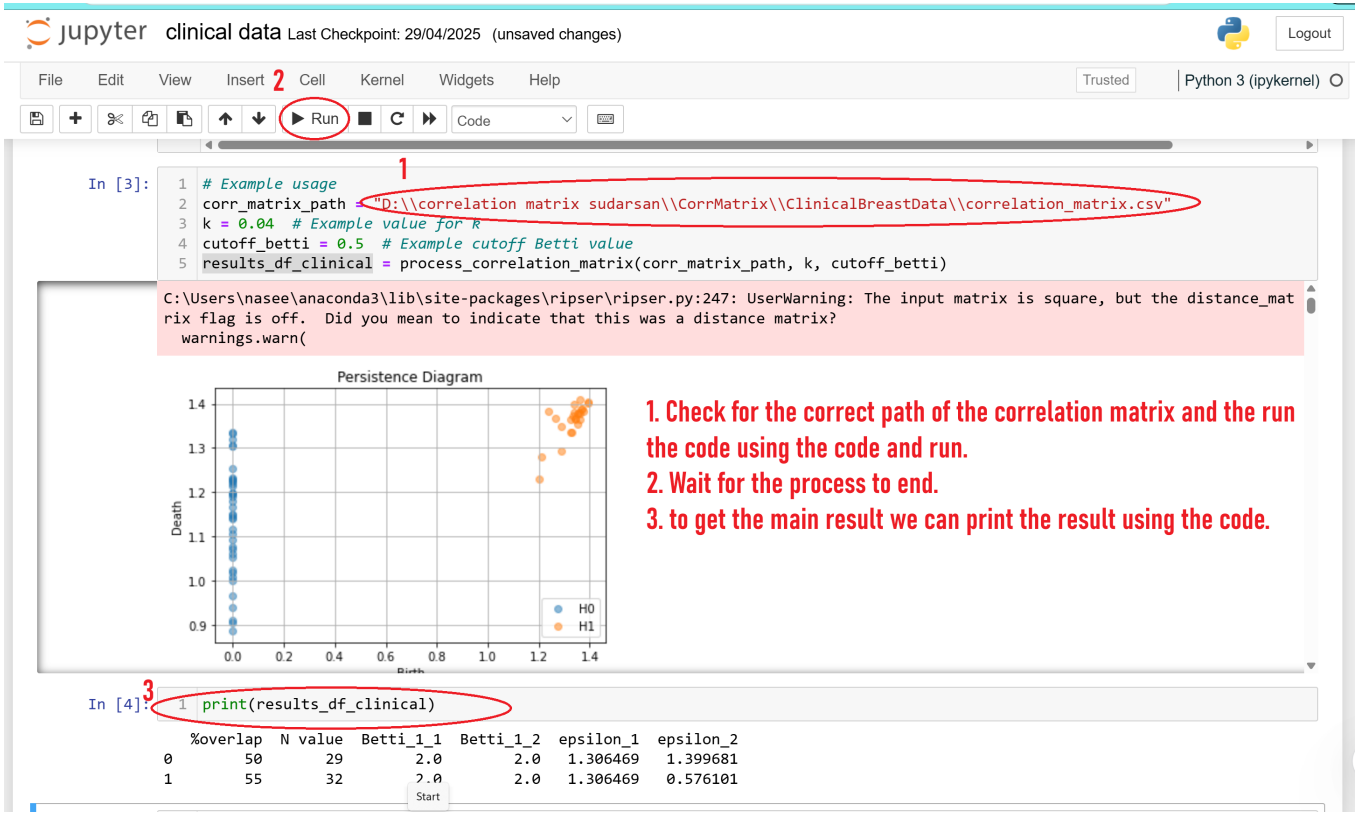

Figure 20: Follow the process to get the final result table. We need to double check for the .csv file and put the path in the section to be read automaticaly by the code.

Thanks For Your Attention!