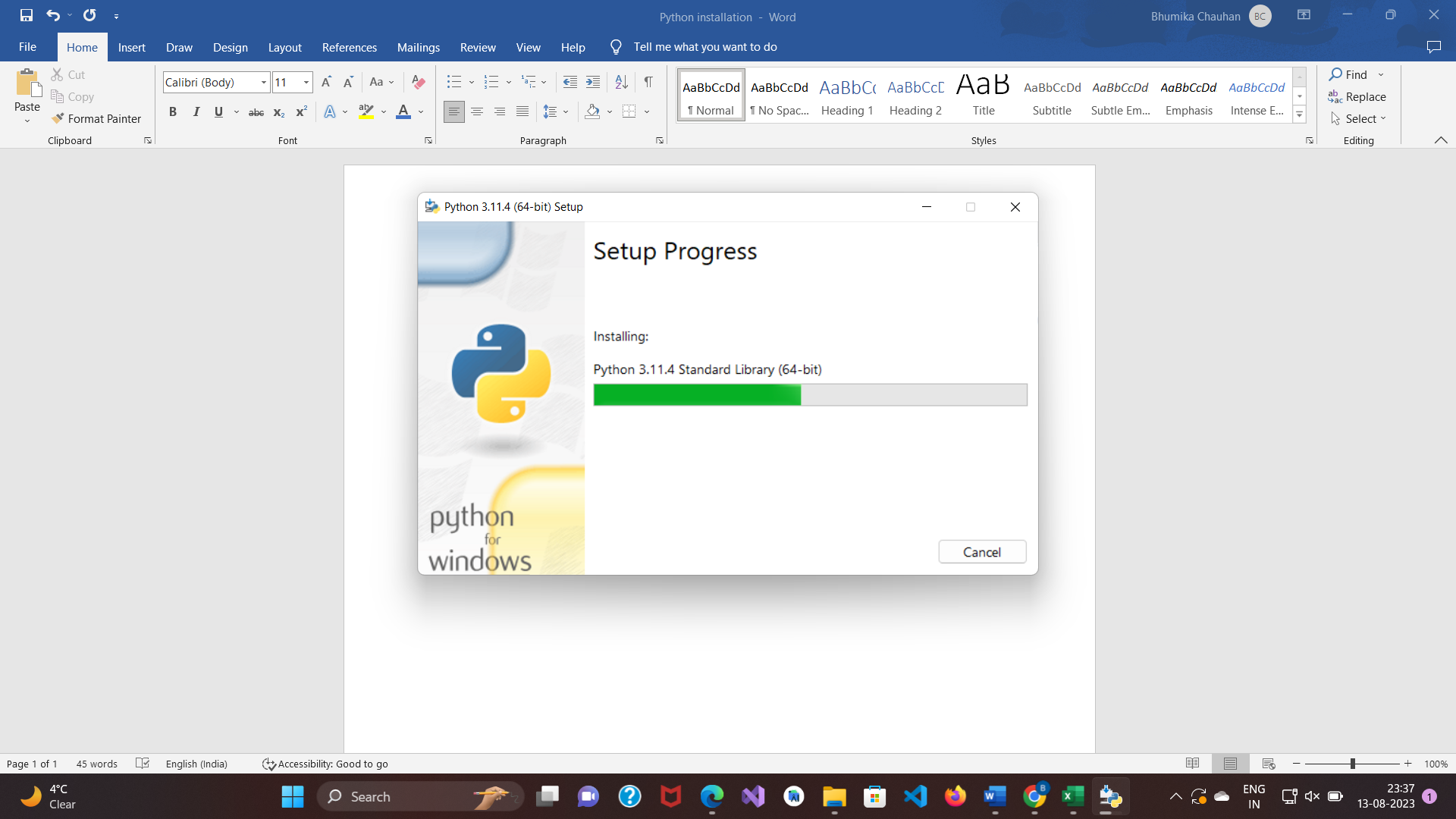
Documentation

Python installation

**Install Python:**

First, download the latest version of Python from the official website: <https://www.python.org/downloads/>

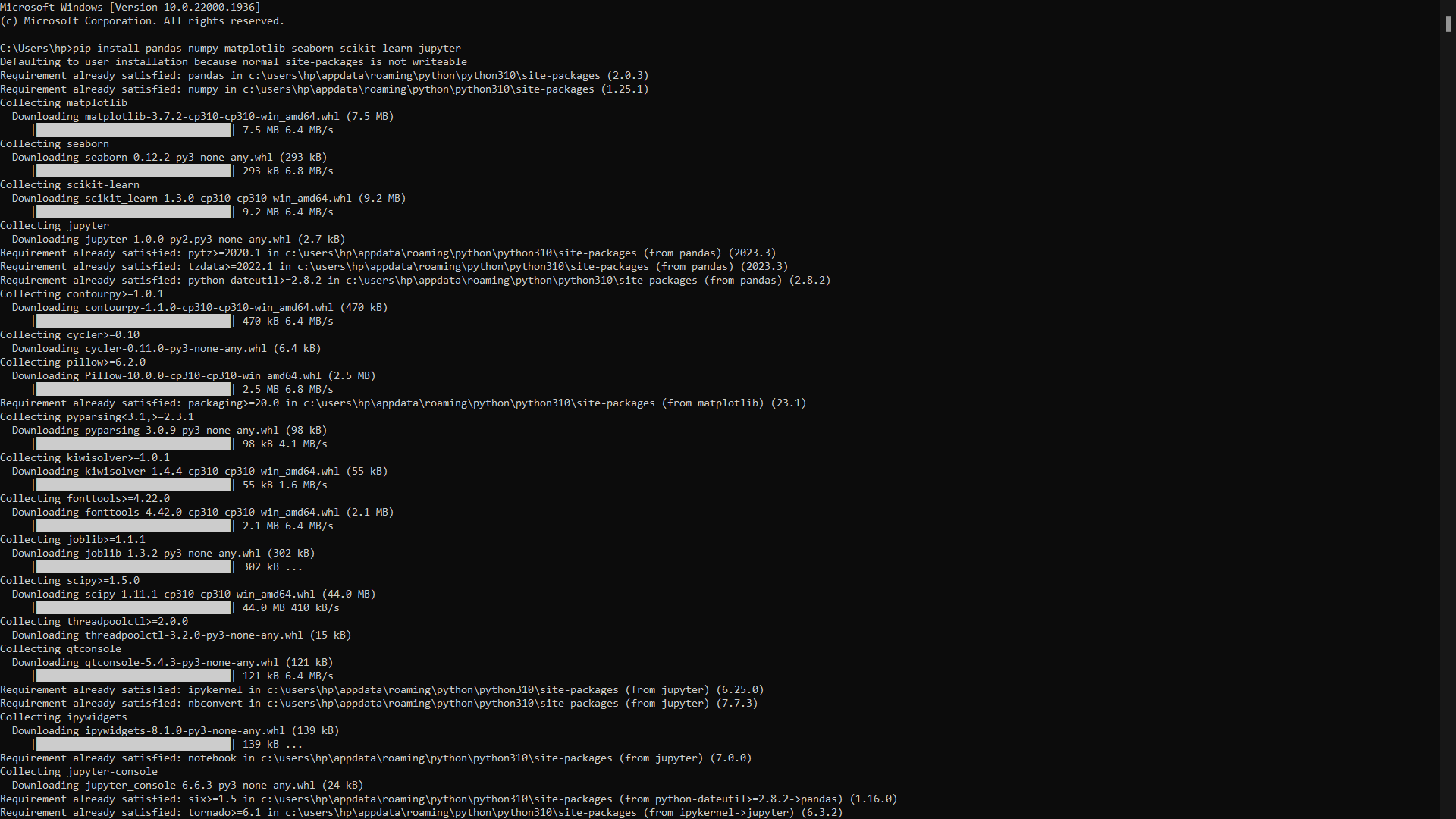
Make sure to click the box to add Python to your system PATH during installation. By doing this, you'll be able to launch Python from the command line.



**Install Libraries:**

Using the package manager pip, you can install the required libraries after Python has been installed. Enter the following commands into a terminal or command prompt:

pip install pandas numpy matplotlib seaborn scikit-learn jupyter



pandas: For data manipulation and analysis.

numpy: For numerical operations and array manipulations.

matplotlib and seaborn: For data visualization.

scikit-learn: For machine learning and data modeling.

jupyter: For interactive data analysis using Jupyter Notebooks.

**Verify Installations:**

Once the libraries have been installed, you may check that they were properly installed by starting a Python interpreter. To launch the interpreter, open a terminal or command prompt and type python. Try importing the libraries after that:

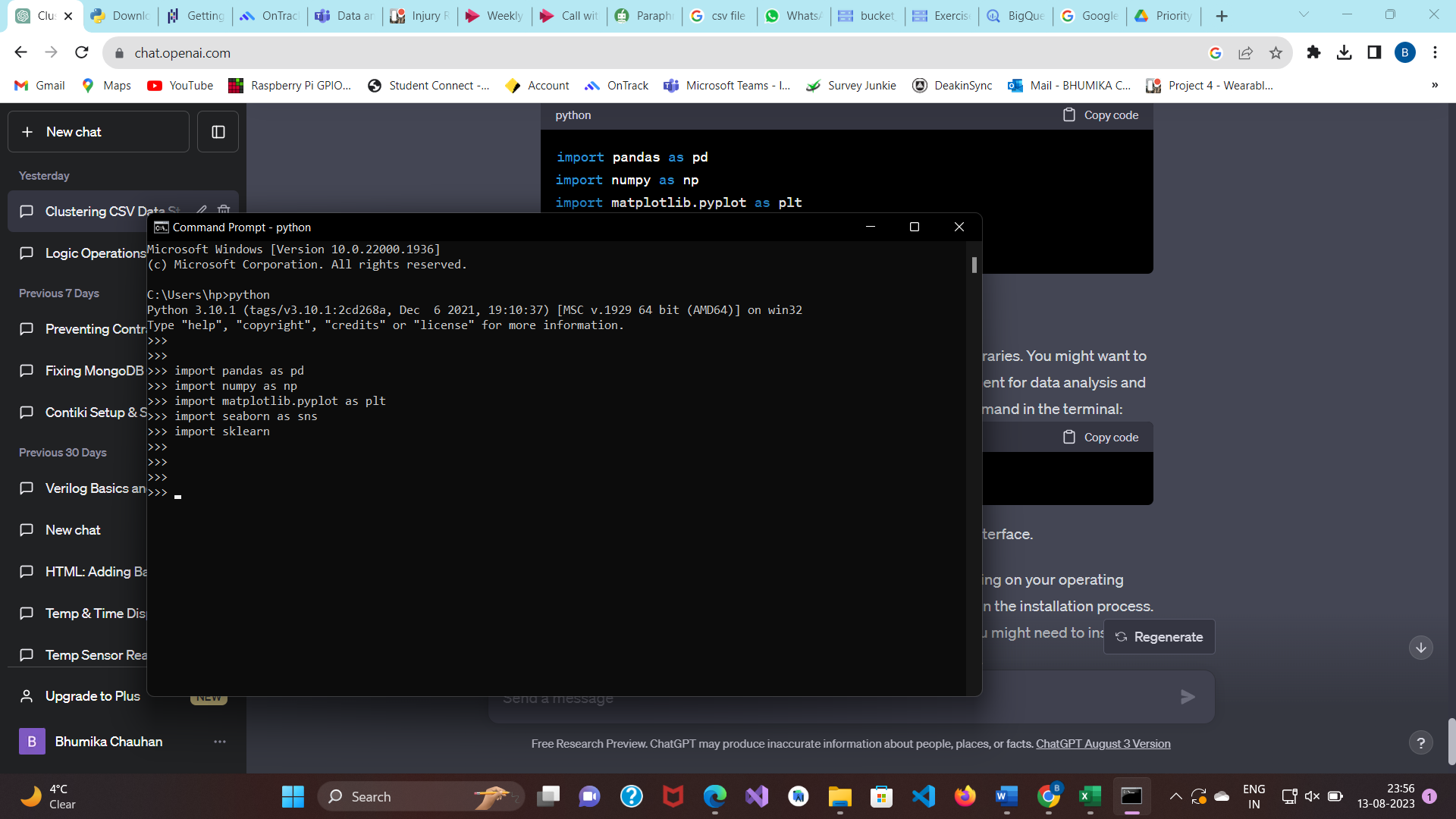
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

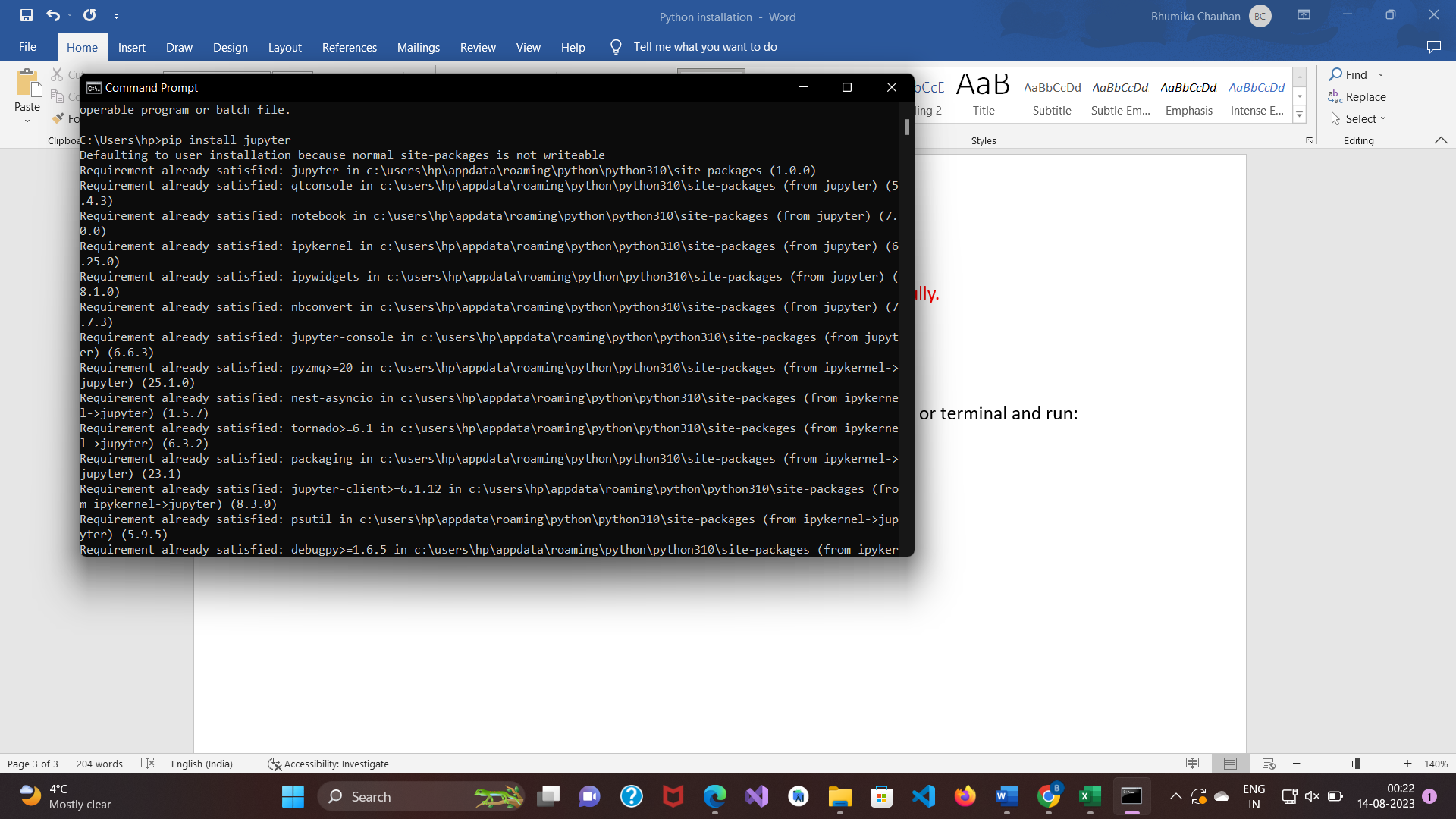


If no errors occur, the libraries are installed successfully.

**Install Jupyter Notebook:**

We need to install Jupyter Notebook using pip. Open the command prompt or terminal and run:

pip install jupyter



**Check PATH Environment Variable:**

If the Jupyter Notebook is still not recognized after installation, you might need to add the directory containing the Jupyter executable to your system's PATH environment variable.

Find the directory where Python scripts are installed. This is usually something like C:\Users\<YourUsername>\AppData\Local\Programs\Python\<PythonVersion>\Scripts on Windows.

Add the above directory to your PATH. Here's how you can do it:

Open the Start menu and search for "Environment Variables."

Click on "Edit the system environment variables."

In the "System Properties" window, click on the "Environment Variables..." button.

Under "System variables," find the "Path" variable, select it, and click "Edit."

Click "New" and add the path to the directory where Jupyter is installed (e.g., C:\Users\hp >\AppData\Local\Programs\Python\<Python310>\Scripts).

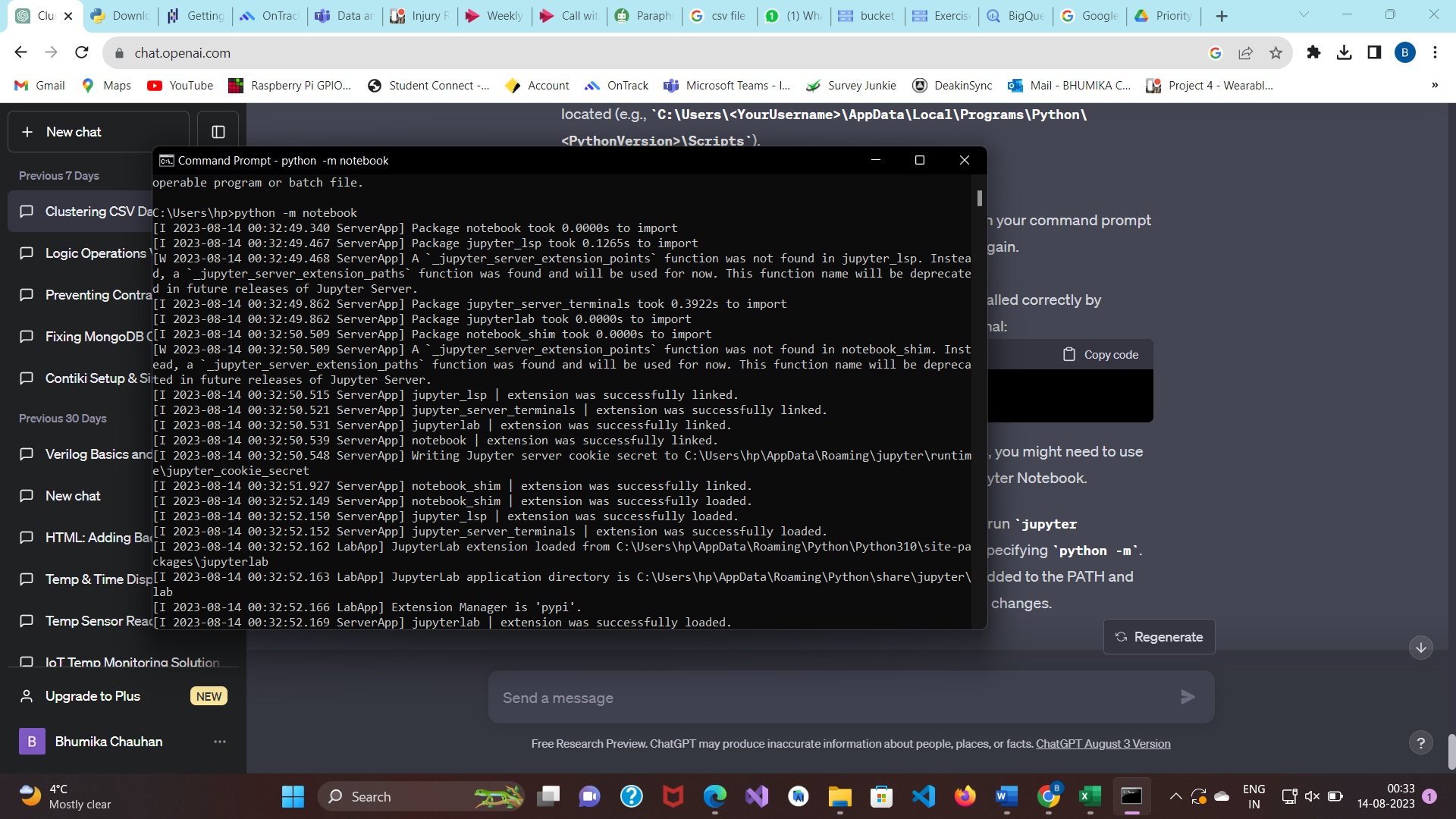
Click "OK" to close all windows.

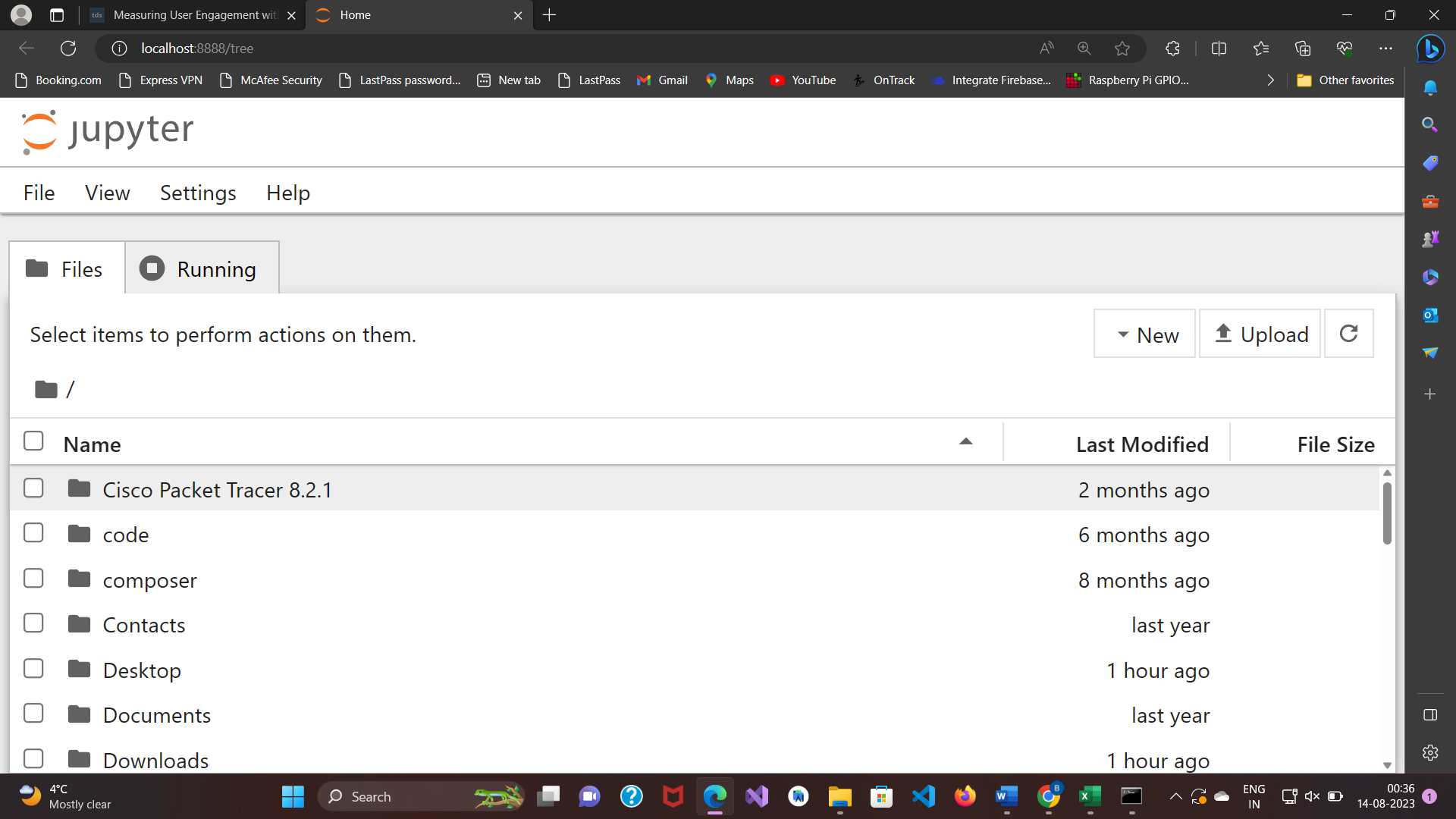
**Verify Installation:**

If the problem persists, you can verify that Jupyter Notebook is installed correctly by running the following command in your command prompt or terminal:

python -m notebook

This command should start the Jupyter Notebook server. If it works, you might need to use python -m before notebook every time you want to launch Jupyter Notebook.





<https://business.tutsplus.com/articles/how-to-clean-up-data-in-a-csv-file--fsw-39773>

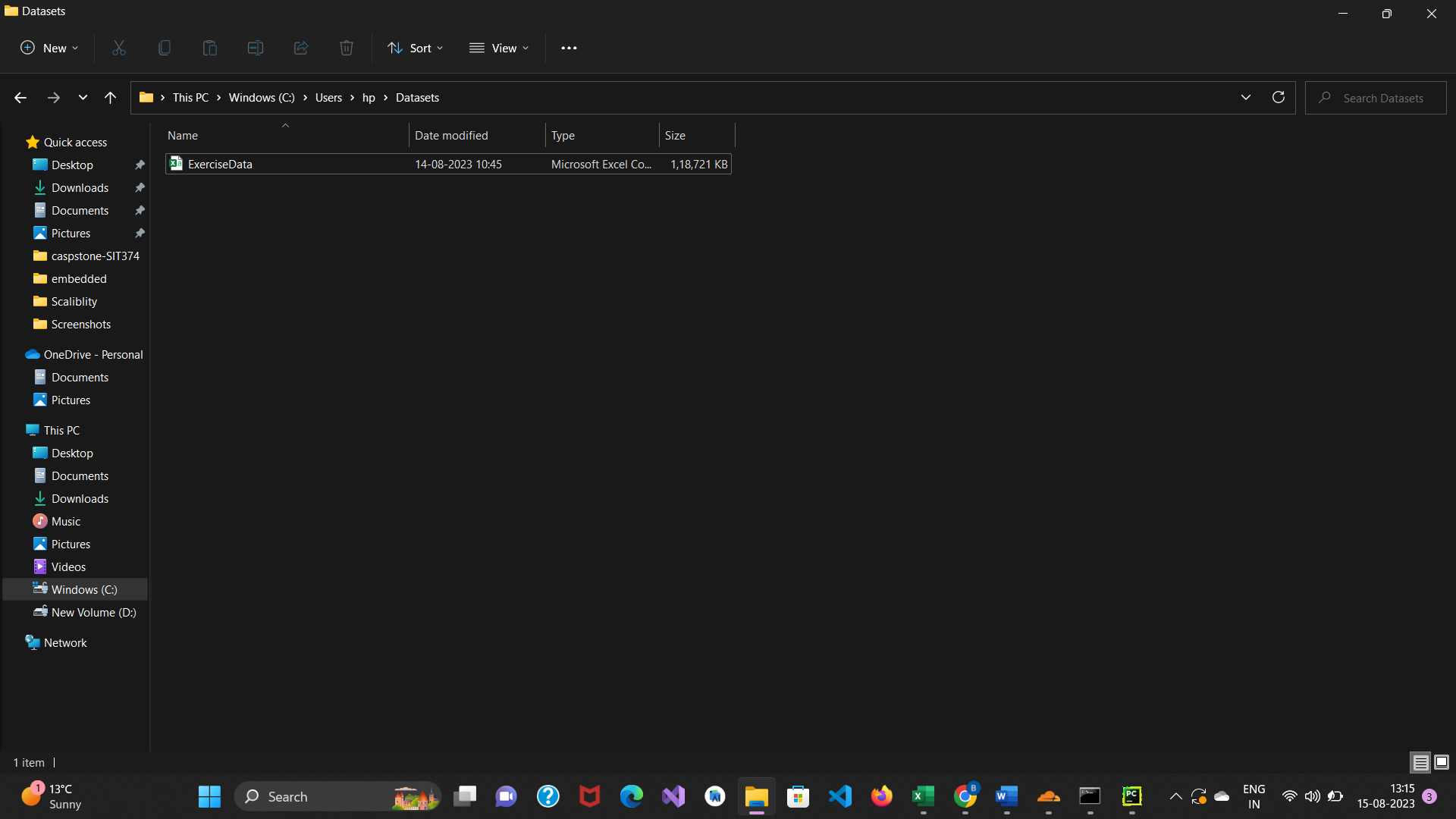
**Pythonic Data Cleaning With pandas and NumPy**

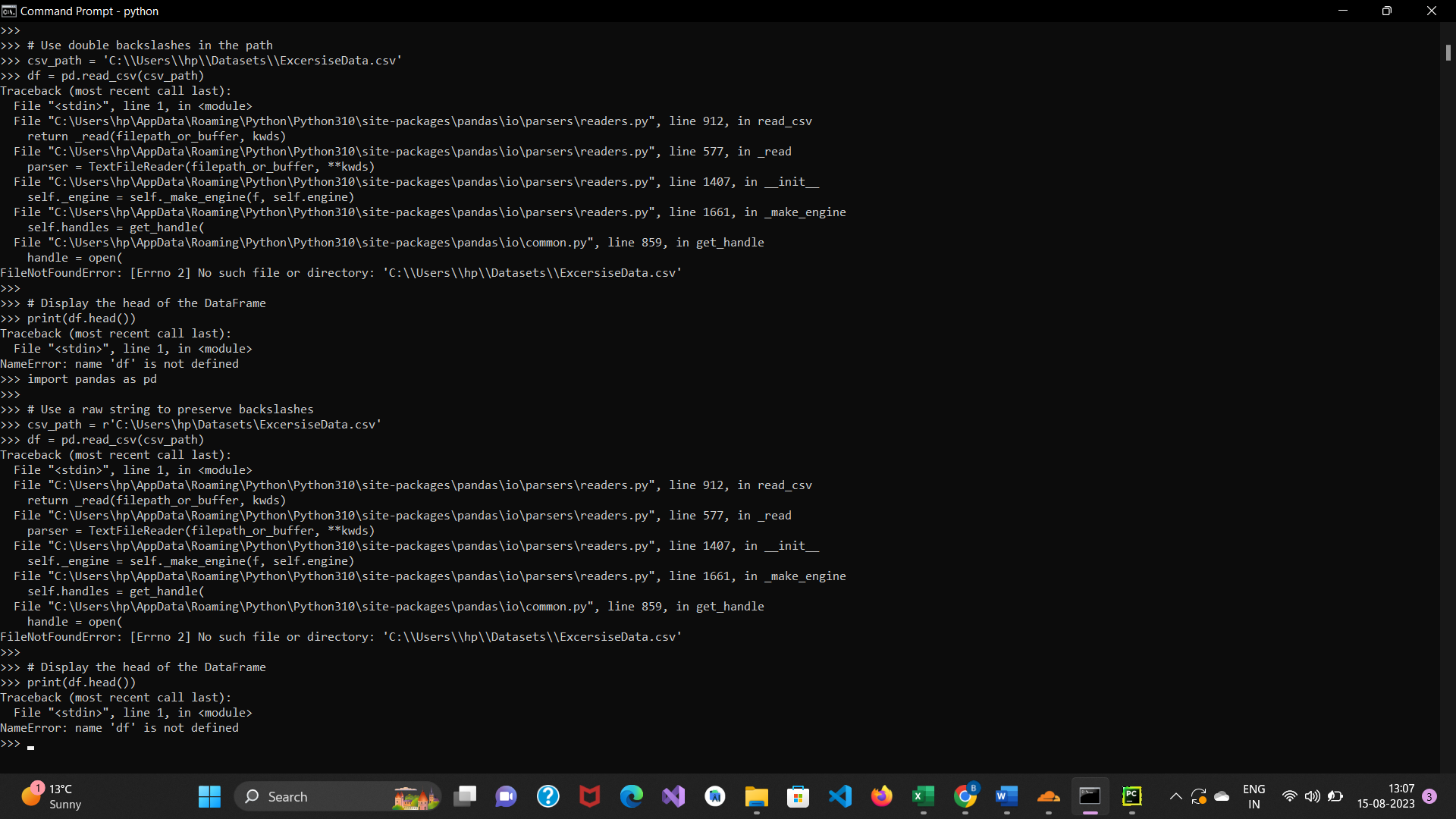
The drop() method in pandas offers a convenient way to eliminate unneeded columns or rows from a DataFrame. Let's examine a straightforward example where we remove certain columns from a DataFrame.

First, let's turn the CSV file "BL-Flickr-Images-Book.csv" into a DataFrame. In the following instances, we instruct pd.read\_csv to use a relative path, indicating that all of the datasets are located in the Datasets folder in the current working directory:

df = pd.read\_csv('Datasets/ExerciseData.csv')

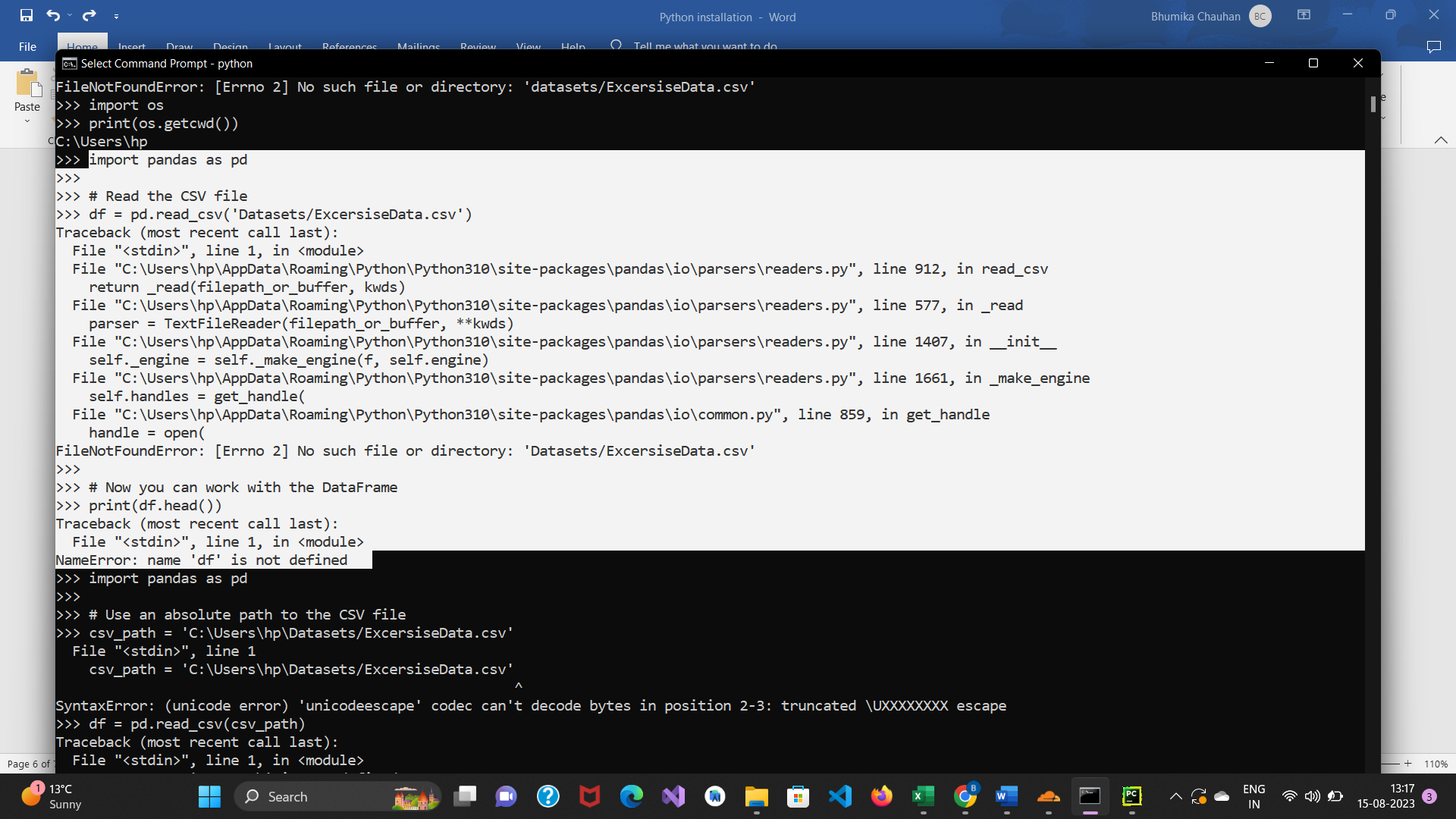
df.head()





Error solving🡪

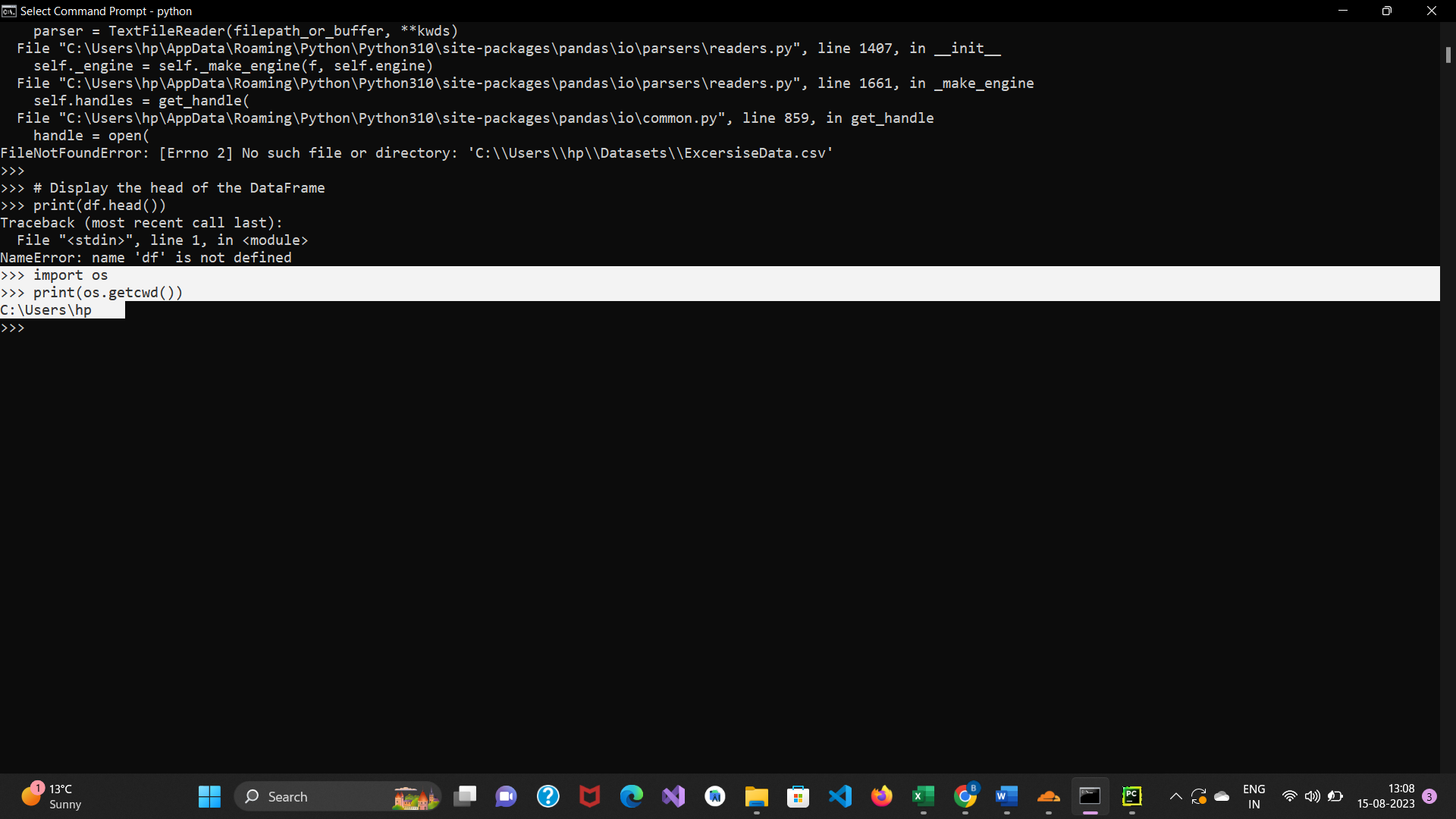
Check File Path: Double-check that the CSV file 'ExcersiseData.csv' exists in the specified path 'C:\Users\hp\Datasets'. Ensure that the filename is spelled correctly, including capitalization.

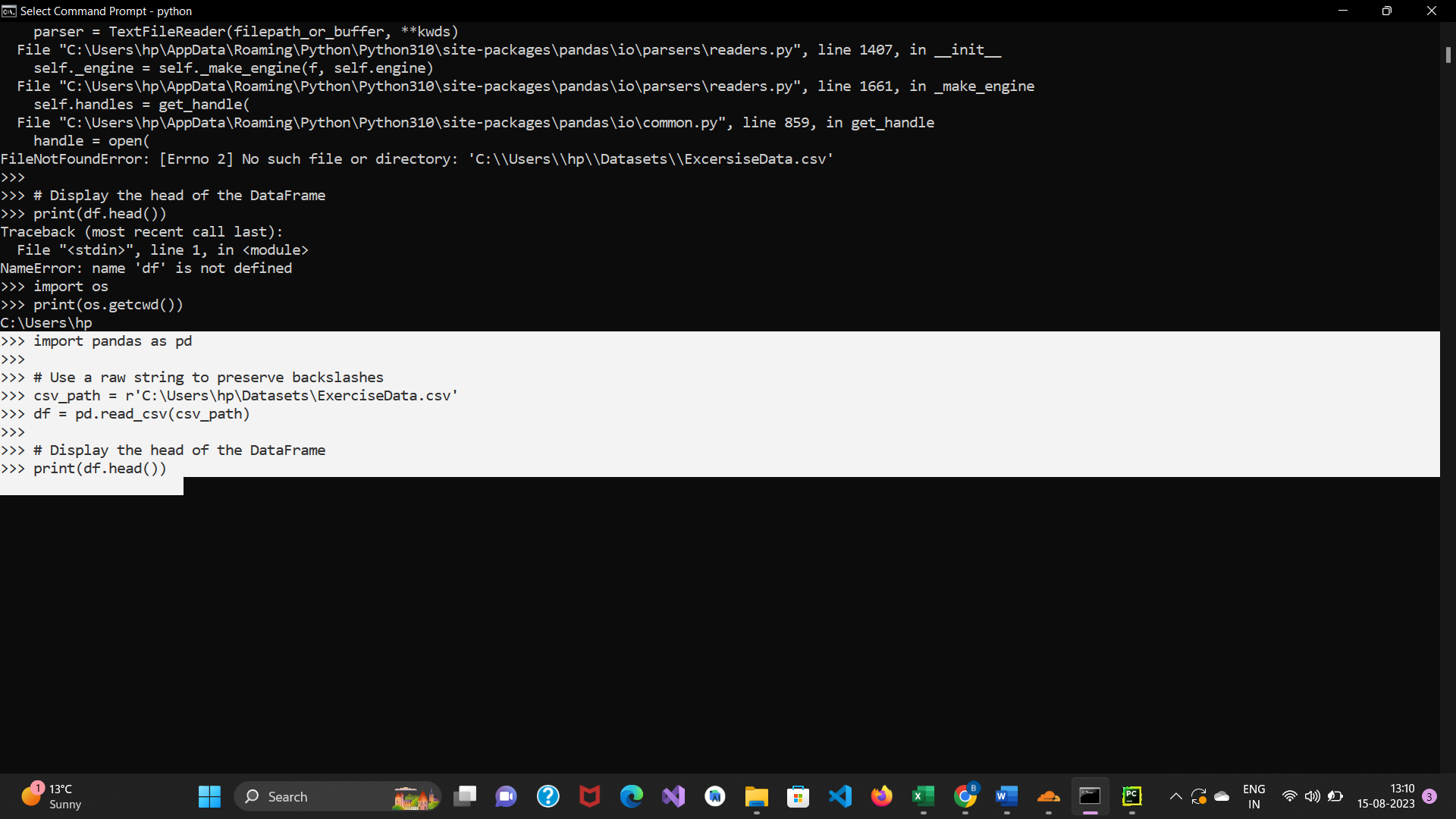


Current Directory: Make sure you're running the Python code from the correct directory. The working directory should contain the 'Datasets' folder with the CSV file. You can use the following code to print the current working directory:

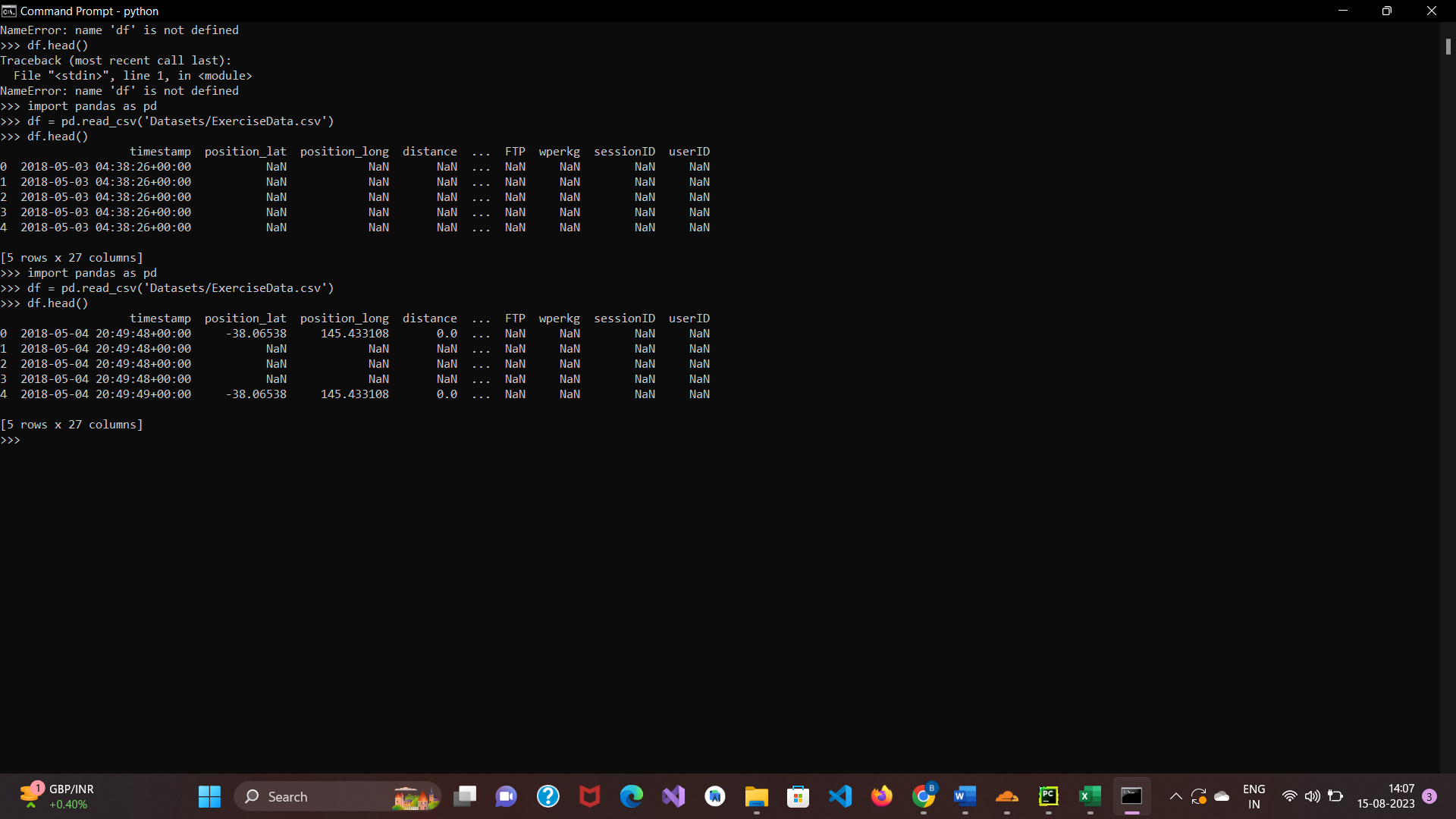
import os

print(os.getcwd())





Error solved



When we look at the first five entries using the head() method, we can see that a handful of columns provide ancillary information that would be helpful to the data but isn’t very descriptive of the data itself:

We can drop these columns in the following way:

to\_drop = ['position\_lat',

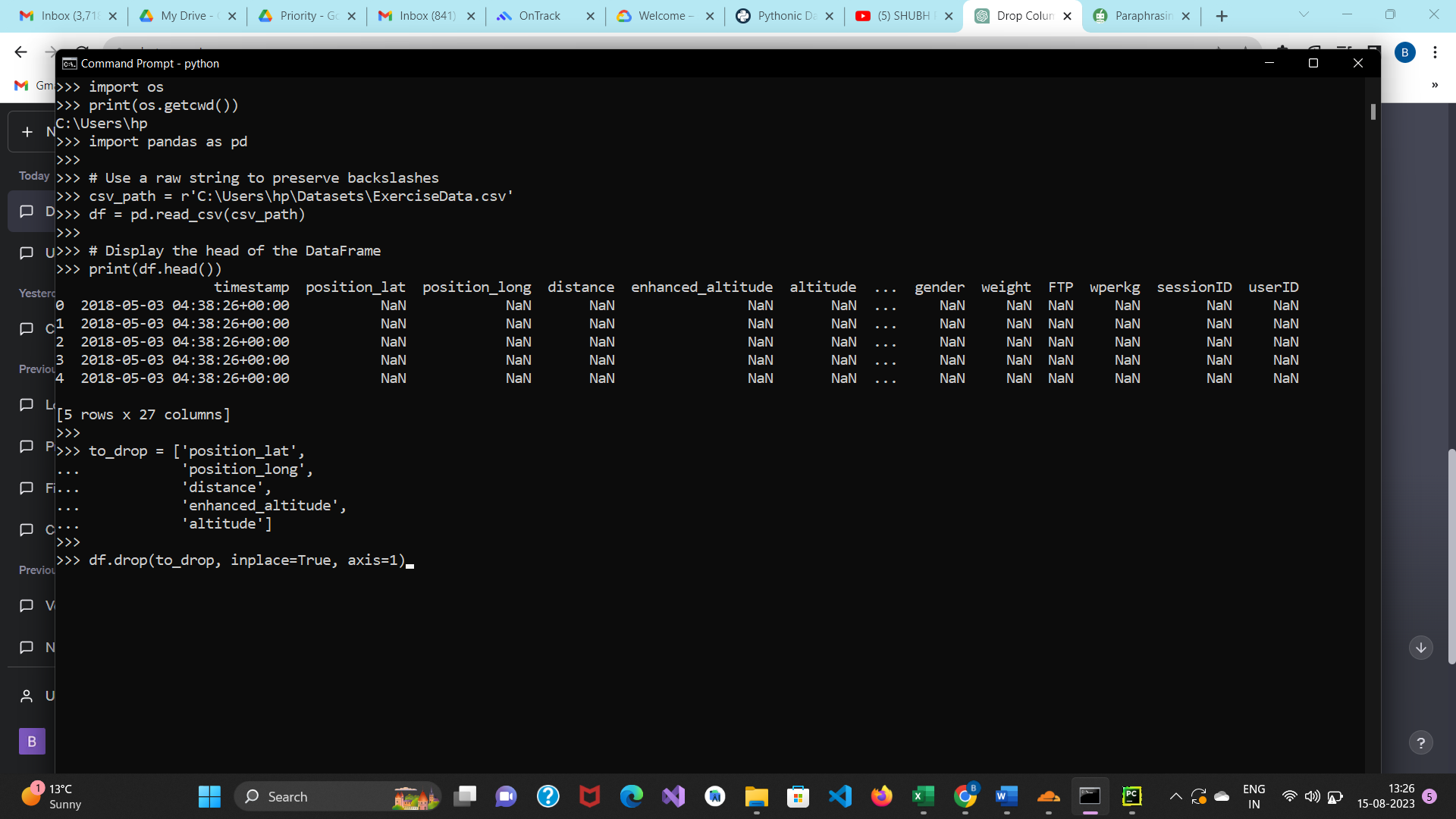
'position\_long',

'distance',

'enhanced\_altitude',

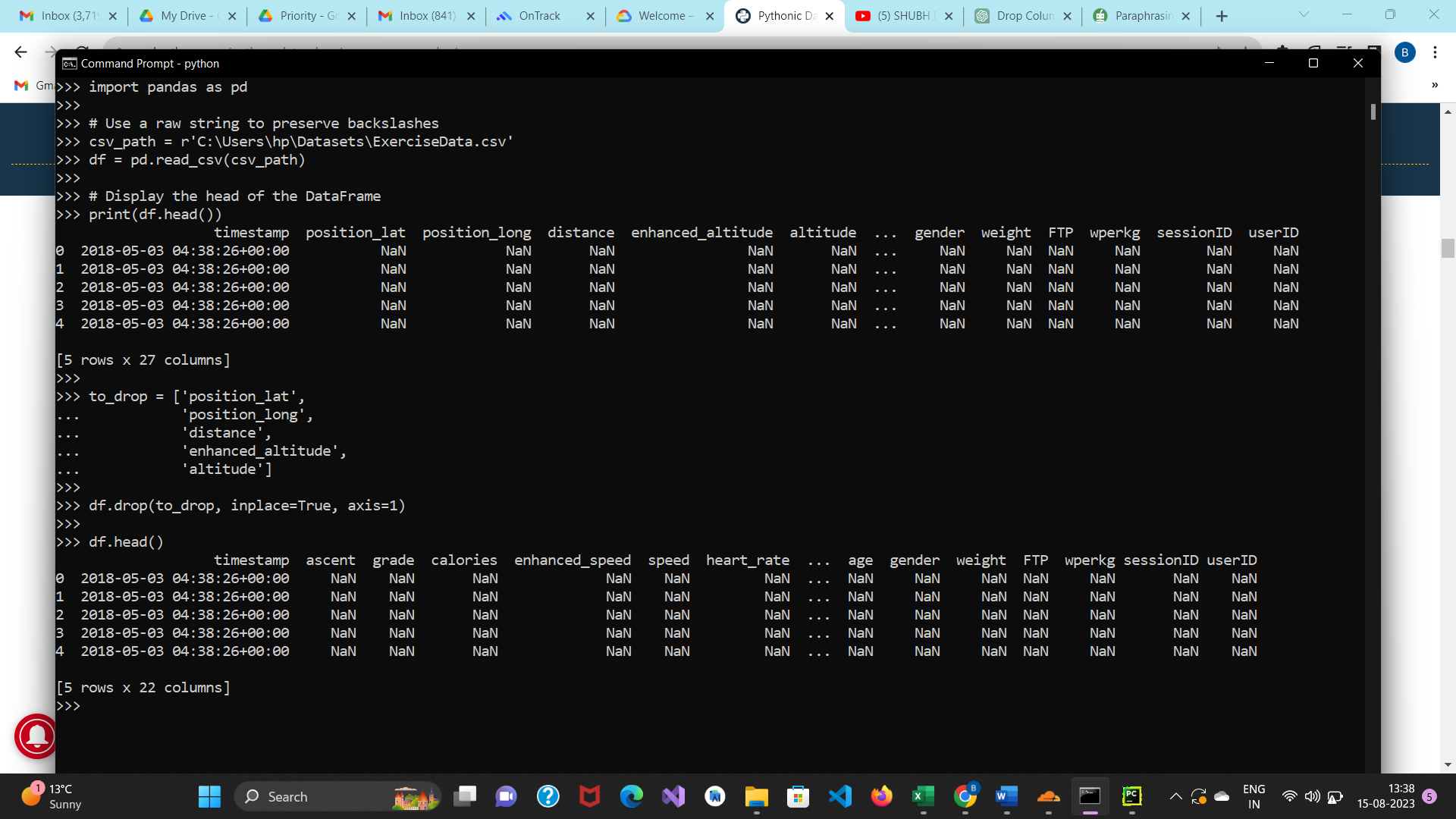
'altitude']

df.drop(to\_drop, inplace=True, axis=1)



The names of all the columns we wish to remove are listed in the list we defined above. The drop() function is then used on our object, with the in place argument set to True and the axis parameter set to 1. This instructs Pandas to search for values to be dropped in the object's columns and that we want the modifications to be made directly in our object.

The undesirable columns have been eliminated, as we can see when we view the DataFrame once more:



We can see that Columns has been removed.

Data set🡪

<https://drive.google.com/file/d/1a8RLiecPYOf7Aoqb-z6-r8z0wpgbAmRD/view?usp=drive_link>

