Item-based Collaborative Filtering Recommendation System

**Implementation Tutorial**

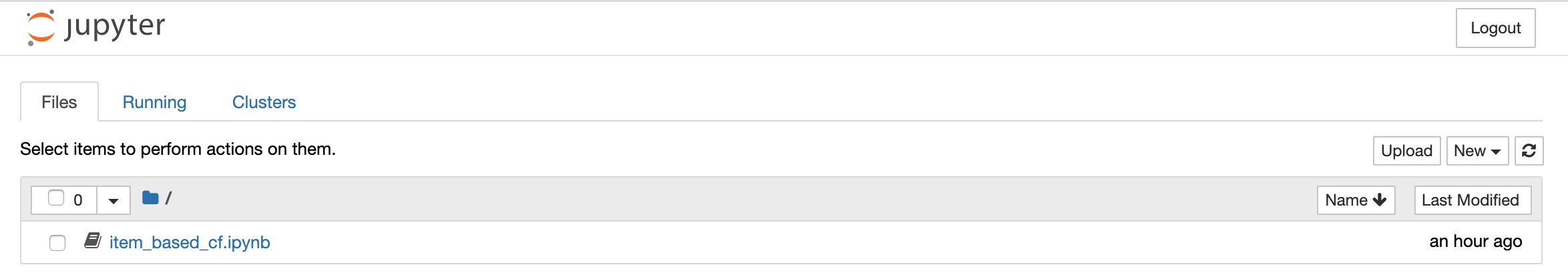
**(Part 1)**

**1. Git clone the repository:** [**https://github.com/linyijun/lab201907.git**](https://github.com/linyijun/lab201907.git)

**2. Open with JUPYTER NOTEBOOK.**



You should see the following screen…



**2. Programming Language and Library Requirements**

You will use Python3, NumPy and Pandas for the implementation.

**3. Data Folder**

We generated the following datasets from the original Yelp datasets with some filters such as the condition: “city” == “Las Vegas”. We randomly took 80% of the review data as the training dataset and 20% of the data as the testing dataset.

a. review\_train.csv – the training data with the columns: user\_id, business\_id, and stars.

b. review\_test.csv – the testing data with the columns: user\_id, business\_id.

c. review\_test\_ground\_truth.csv – the testing data with the ground truth ratings

d. sub\_business.csv – the business information, such as business locations and types

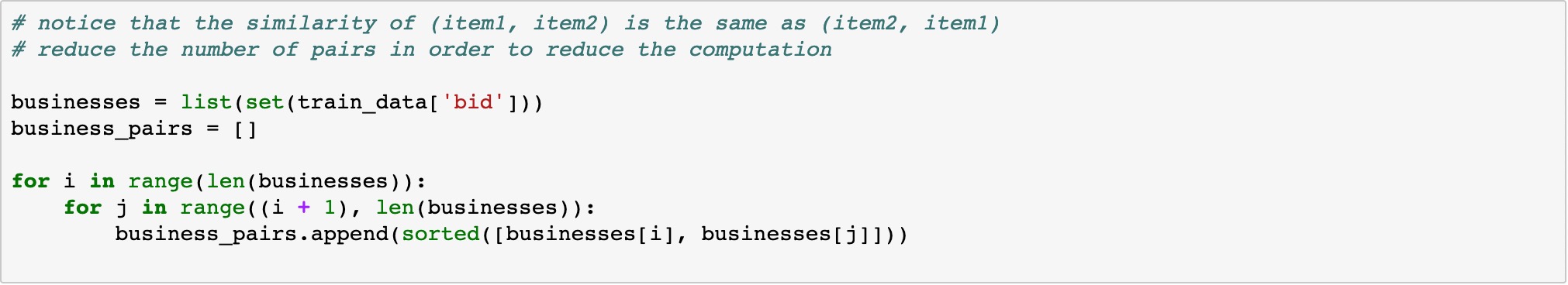
**4. Computing Pearson Correlation (lab\_pearson\_correlation.ipynb)**

a. Loading the datasets

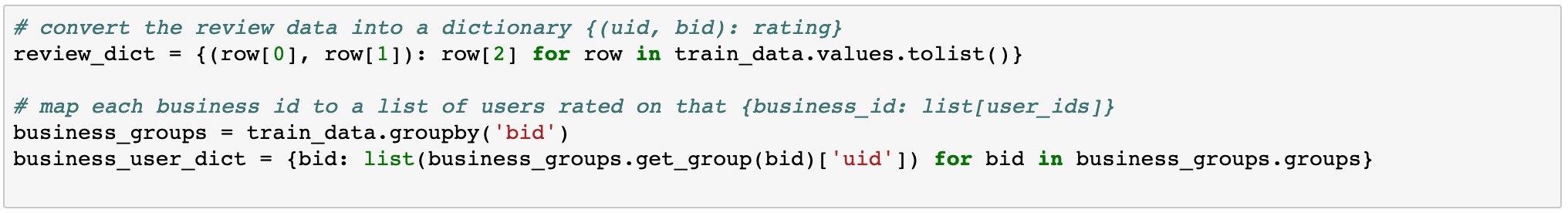
**Your data folder path**



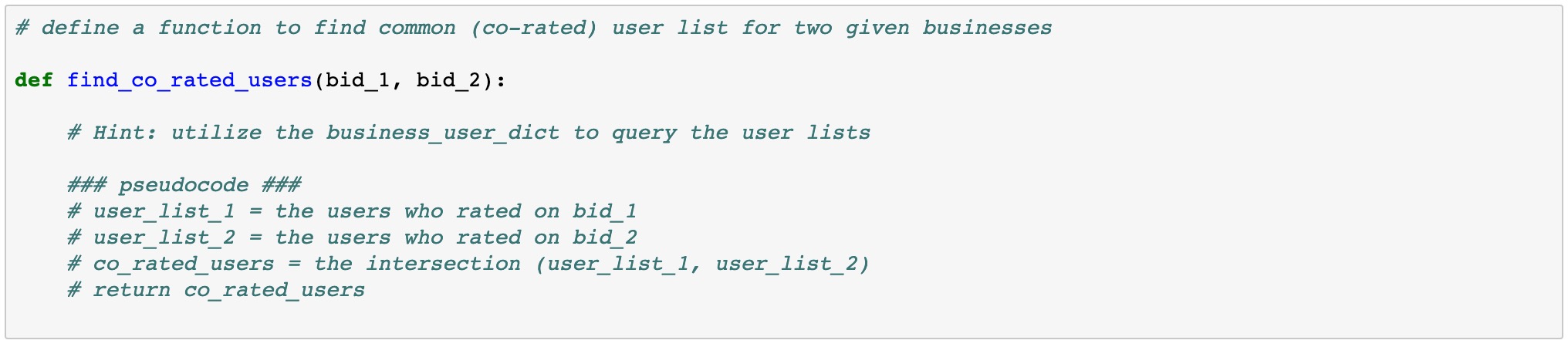
b. Preparing the business pairs that you are going to compute the Pearson correlation



c. Generating some dictionaries to facilitate the computation



d. [Write your code] Implementing the function to find co-rated users for two given items



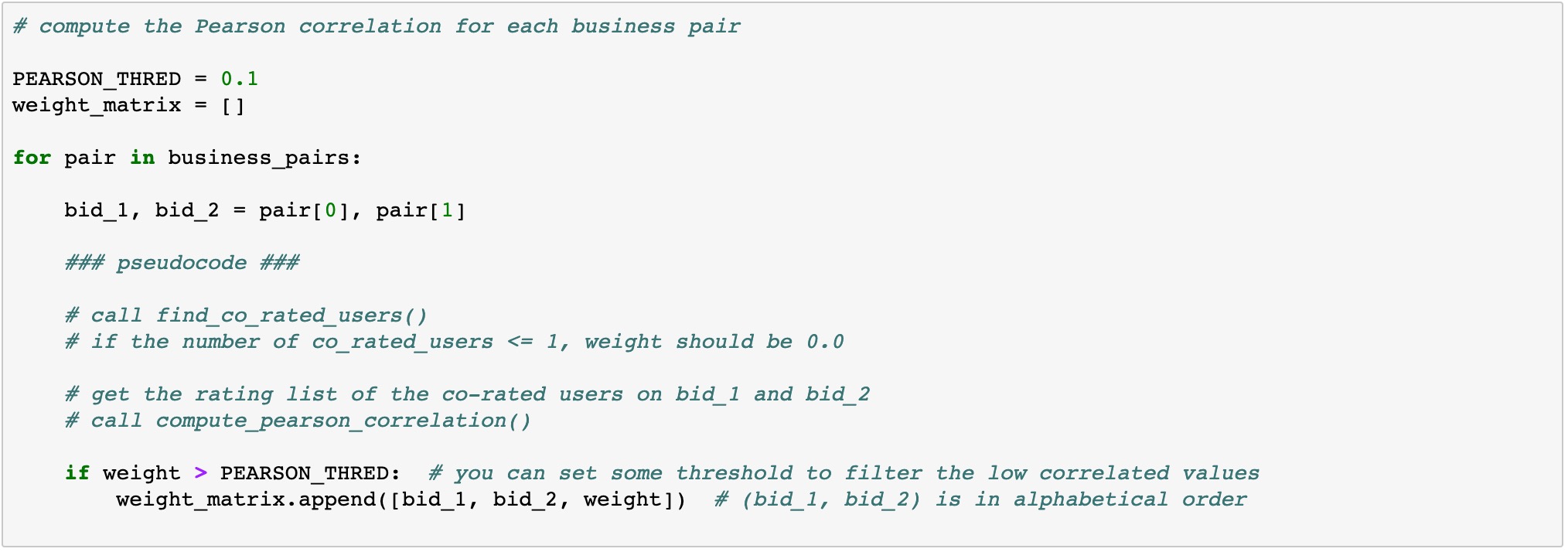
**NOTE: You need to handle the situation that no user rates on the given items.**

e. [Write your code] Implementing the function to compute Pearson correlation given two lists



**NOTE: You need to handle the edge case, e.g., weight\_1 or weight\_2 is 0.0**

f. [Write your code] Implementing the whole process to get the weight matrix



**NOTE: Only store the Pearson value/weight when it is above the threshold**

g. Writing the weight matrix to a CSV file

