Due Date: 27 /2/ 2022

The Task:

You received a file containing data of items that users consumed (e.g., purchased, watched, etc.). Your task is to build a model that predicts which future items a user is likely to consume. The test files contain rows with user ID and two items. One item was consumed by the user and the other item was not consumed by the user. Your goal is to distinguish between the two items.

There are two test files: In the main one (RandomTest.csv), the "negative" items were sampled uniformly from all the list of all items that were not consumed by the user. In the second file (PopularityTest.csv), the "negative" items were sampled from the list of items that were not consumed by the user according to items popularity distribution (the probably for an item to be consumed).

Your goal is to fill both test files as follows:

<userId>, <item1>,<item2>,<bitClassification>

where <userId>, <item1>,<item2> are the user ID, and the IDs of the first and second items respectively as given to you. The last column <bitClassification>, is a binary classification marking your prediction as follows: '0' if you predict that the first item was the item that was liked by the user and '1' if you predict that the second item was liked by the user.

Your solution must be based on an autoencoder (AE). In this assignment, you are free to design any architecture that you wish as long as it can be defined as an AE. Additionally, you are free to work with the train dataset as you wish (train / validation split).
Guidance: for each user, the AutoEncoder encodes and reconstructs the list of items purchased by the user as we saw in class. The loss term should relate to the consumed items and to a random sample of "negative" items.

In addition to the test file, you are required to submit your python code as well as a report that describe your solution, your insight etc. The report should also cover the following:

1) The AE architecture:
    a. Define you encoder and decoder.
    b. Your cost function.
    c. Regularization.
    d. Justify your architecture choice.
2) Describe the training procedure:
    a. Your train / validation split.
    b. Sampling negatives.
    c. The model's loss and an update scheme.
3) Describe your inference:
    a. How do you use your trained models to make predictions.
4) The difference between the first and second test files:
    a. Have you noticed a difference between the first and second test files? Can you explain it?
    b. How can you adopt your training in order to better fit the second test file (PopularityTest.csv)?