Hw3: Twitter User Engagement Recommender System

Course: Massive Data Analytics

Proffesor: Dr. Iman Gholampoor

Mohammad Mehdi Zare

* **Introduction**:

In this assignment, I try to implement a recommender system for Twitter users to offer tweets to users to make a better experience in use. I use PySpark and RDD API in this project.

1. **By searching the Internet, suggest other criteria that can be considered for user interaction with tweets.**

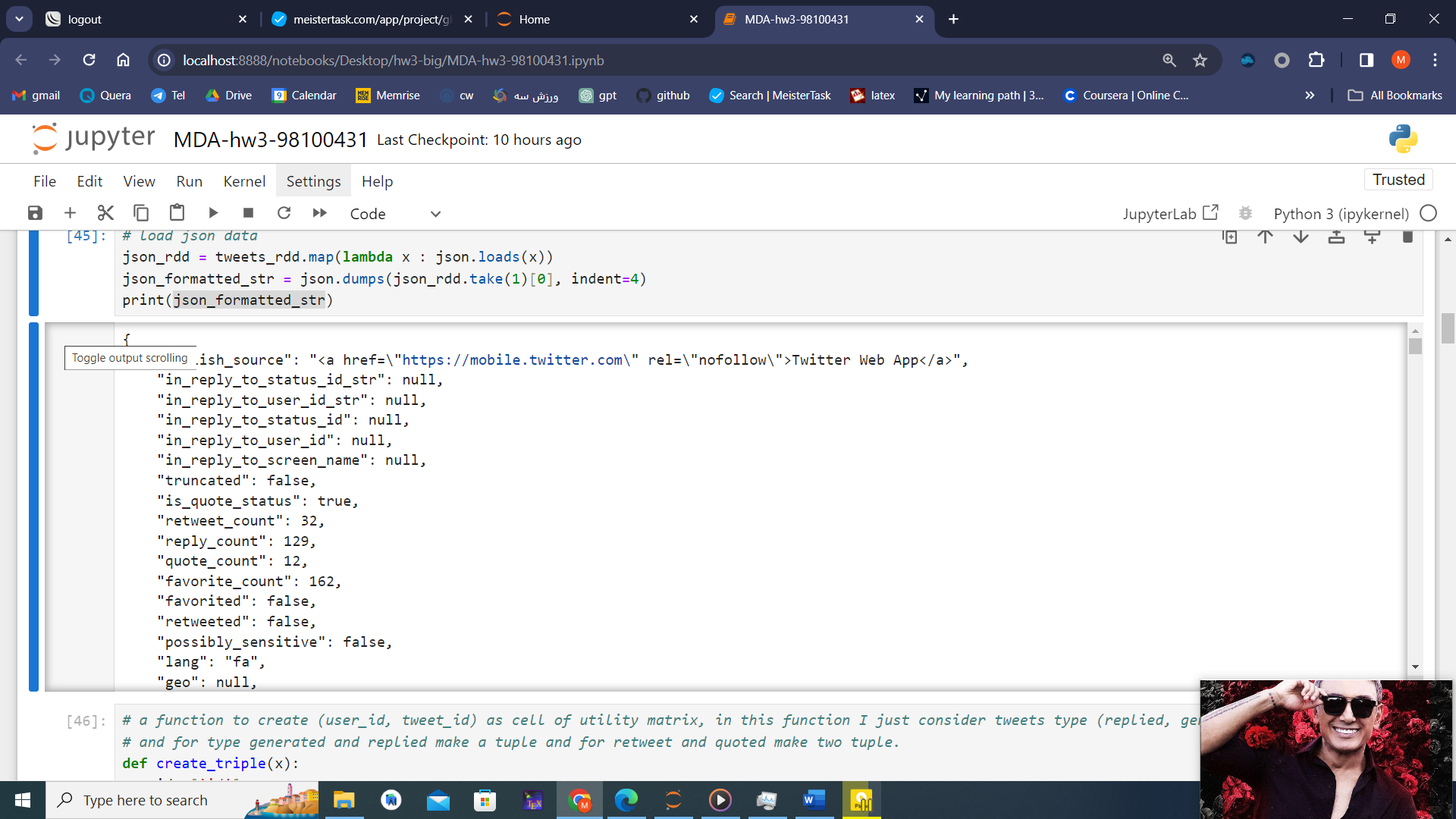
I found these criteria that could be important for user interaction:

* 1. Like: people like tweets that interested in or agree with. Therefore, likes can be considered as criteria.
  2. Mention: some people mention others to attract their attention.
  3. View: this criterion counts how many people visit that tweet.
  4. Comments: The number of comments under a tweet indicates discussions and higher interaction with the content
  5. Clicks: If a tweet contains a link, the number of clicks on that link demonstrates user engagement with the content.
  6. Hashtags: The frequency of hashtags in a tweet can help categorize it and increase visibility on specific topics.

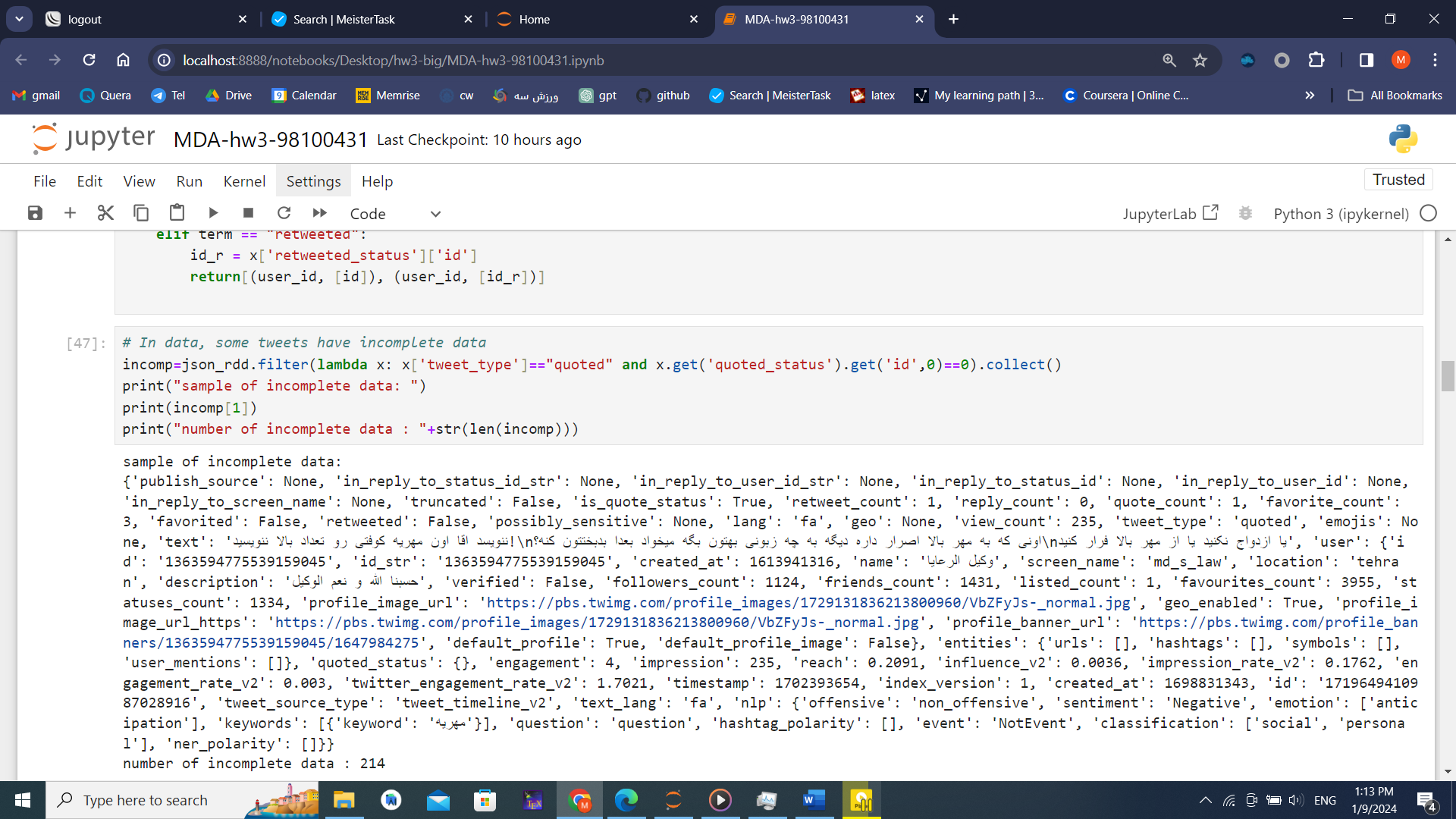
1. **Implement a system that provides users with personalized tweets based on their interactions and preferences. The system should pay attention to retweets, replies, and quotes and provide personalized suggestions for each user.**

I create a utility matrix and use a user-user collaborative filter approach. Each user interacts with tweets by generating, retweeting, and replying. So for user X, I have a vector . Now I’m finding similar users by cosine similarity, then I can recommend tweets. For sample user , there are similar users and , is a list of tweets ids that can recommend to sample user.

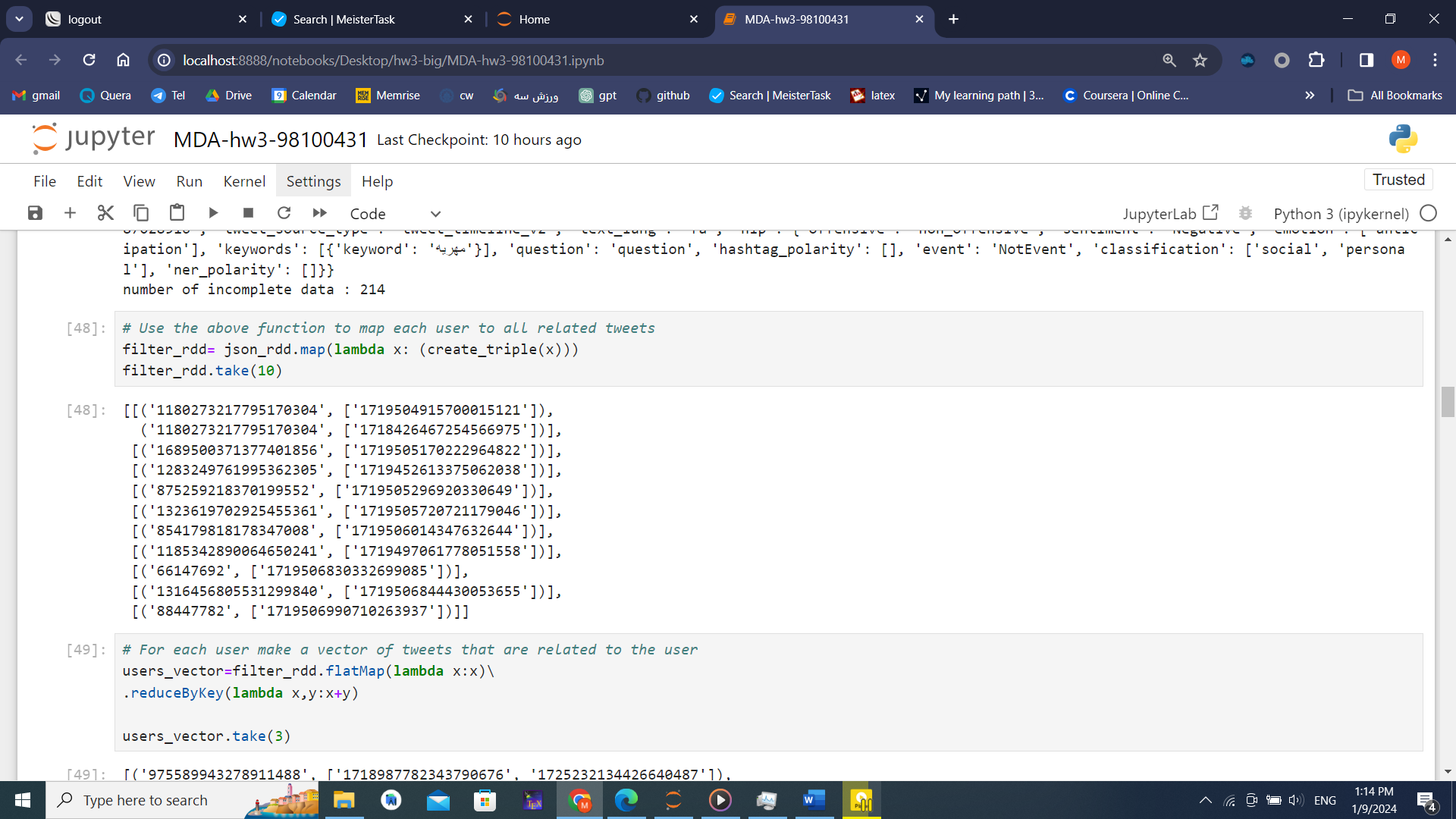
See some outputs of code and algorithm:

Sample data: 

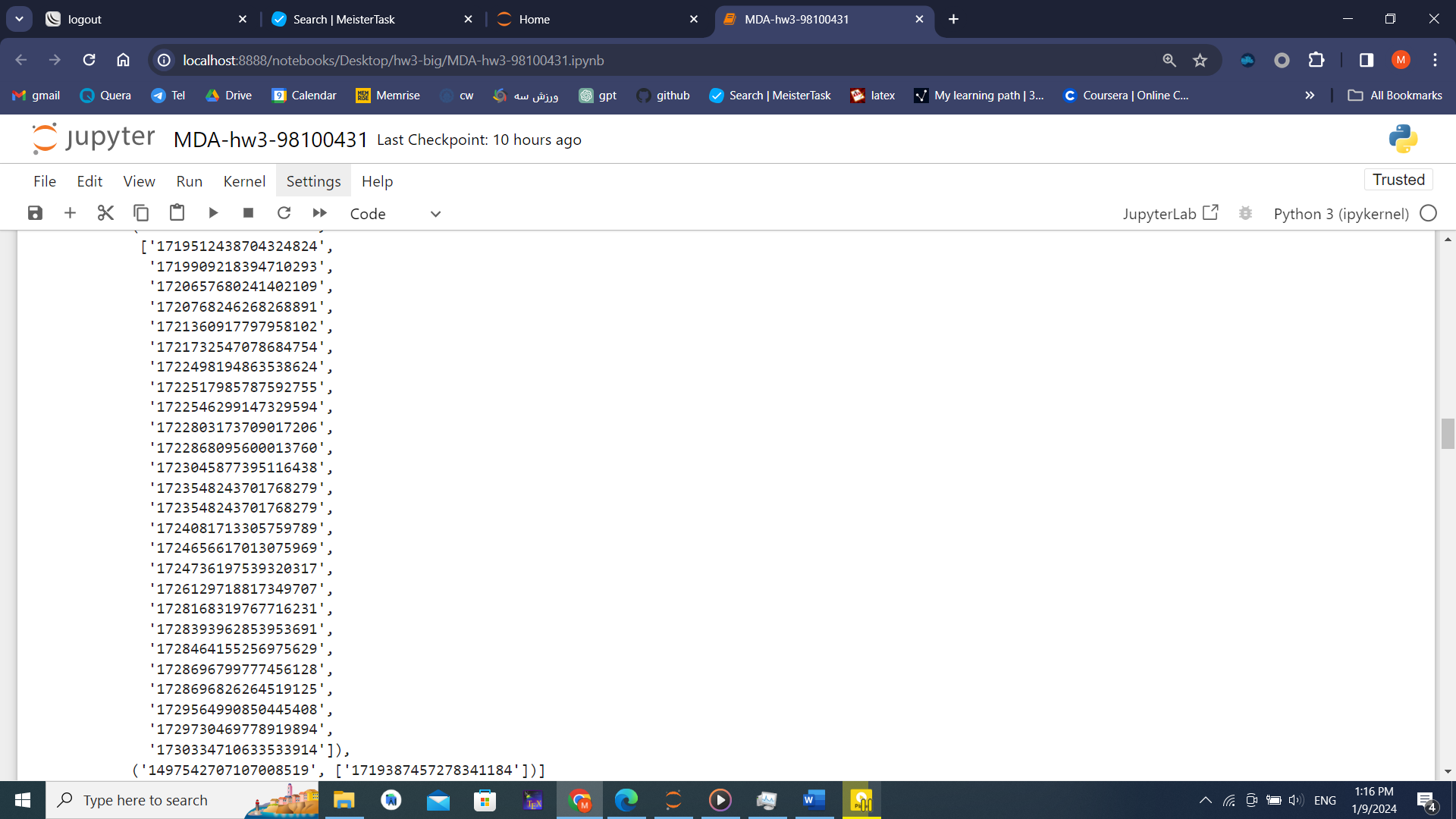
Incomplete data:



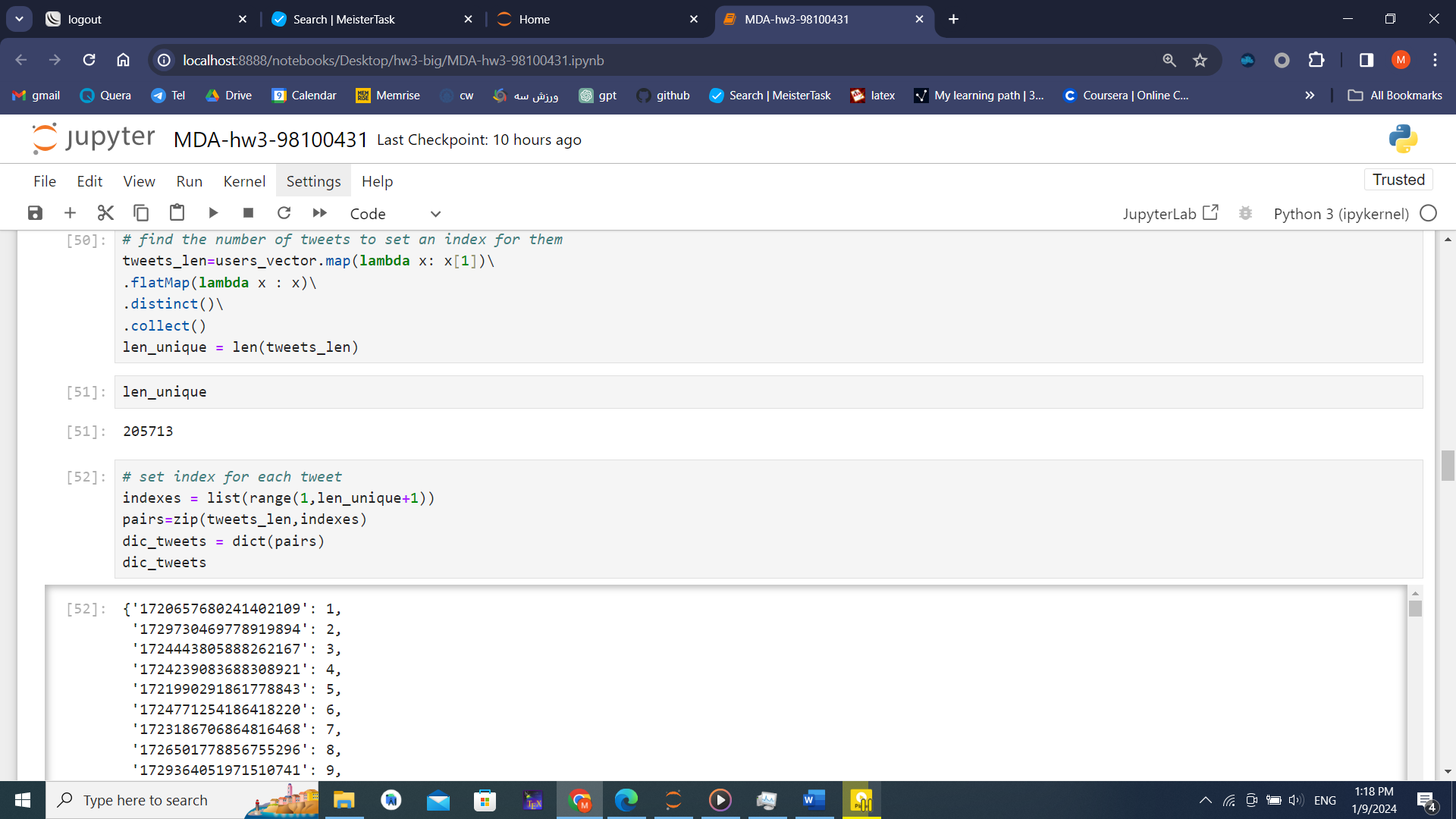
User-tweet tuple:



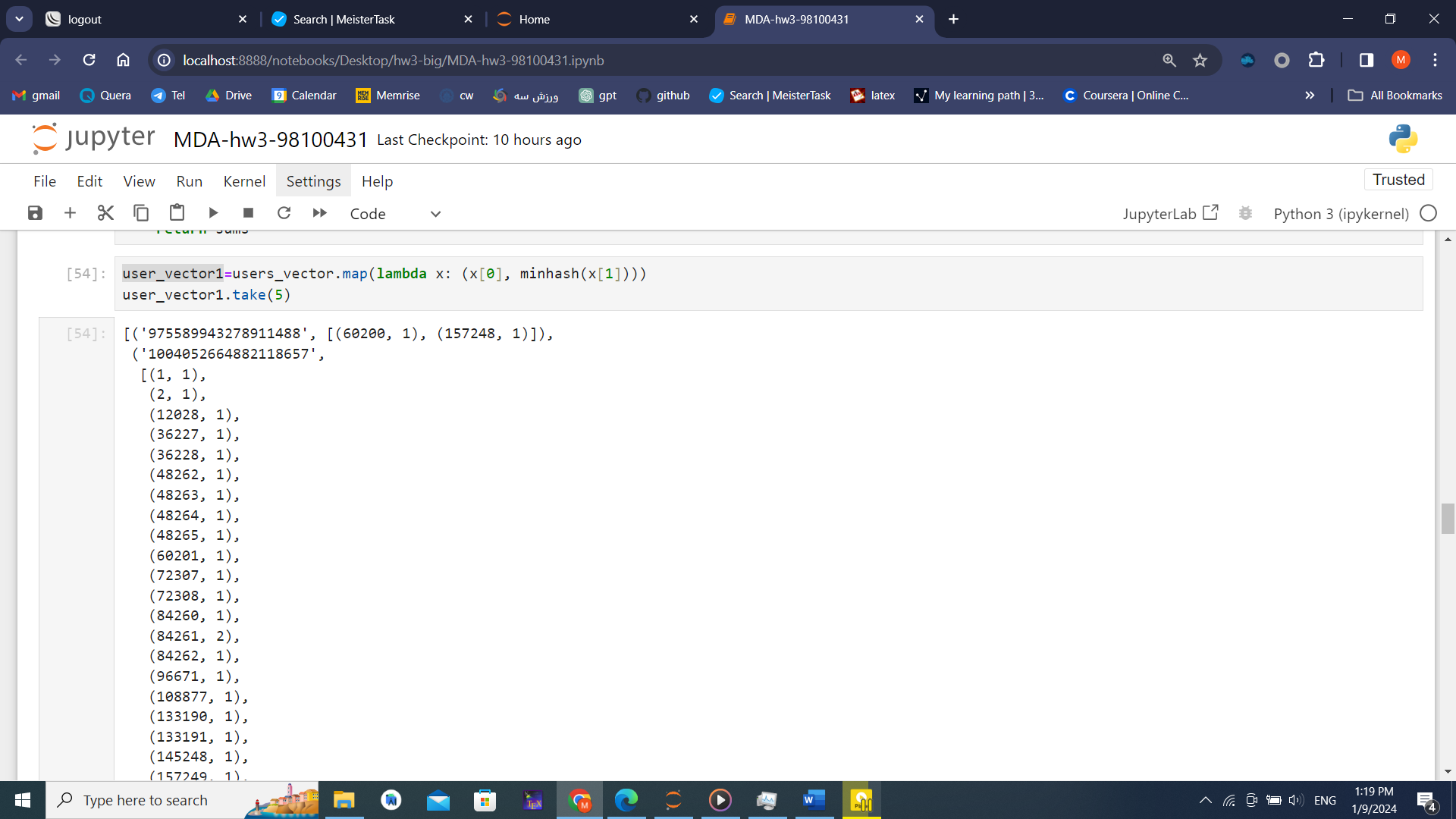
User and list of related tweets:



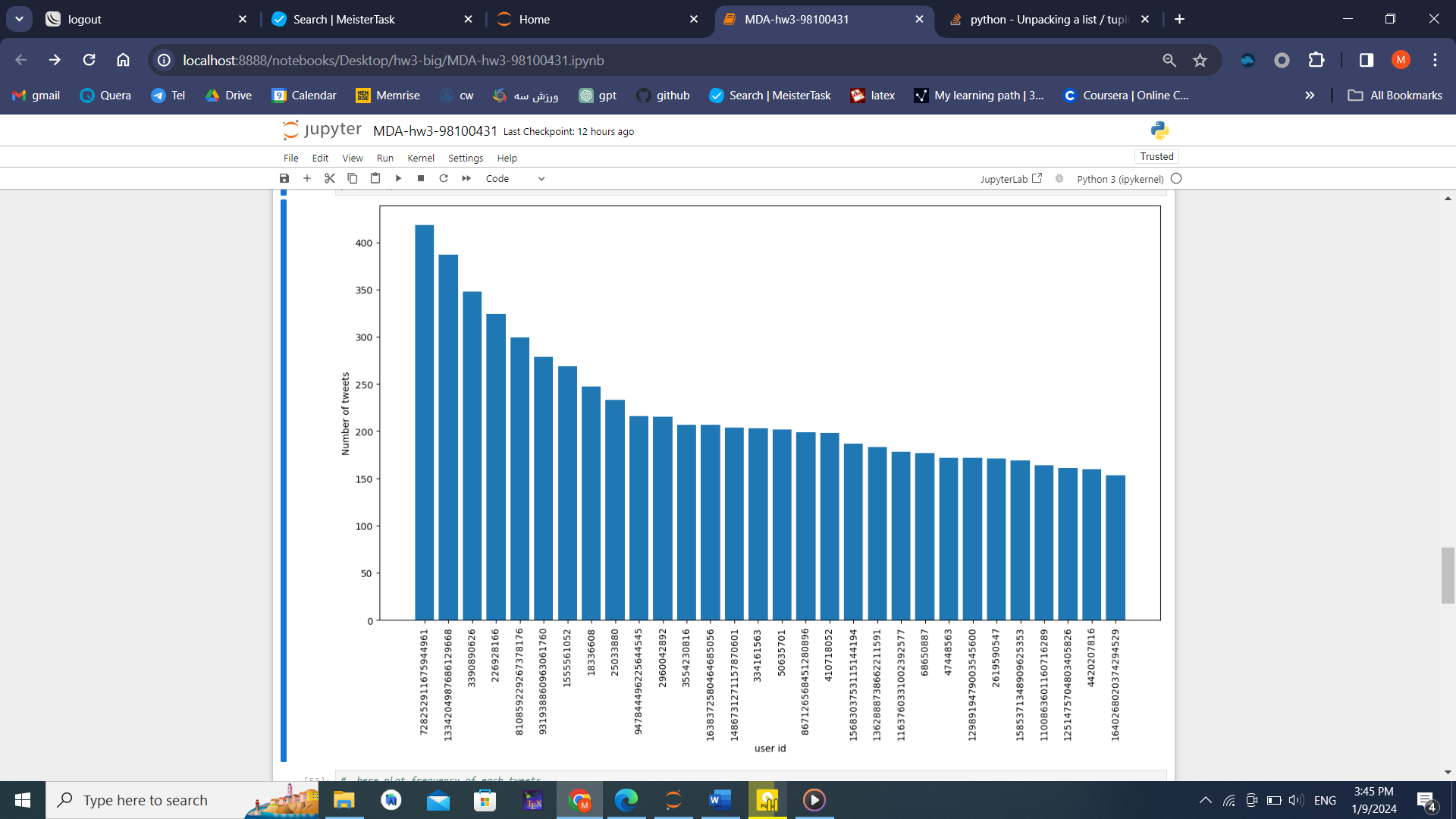
Number of tweets and sample of tweet indexing:



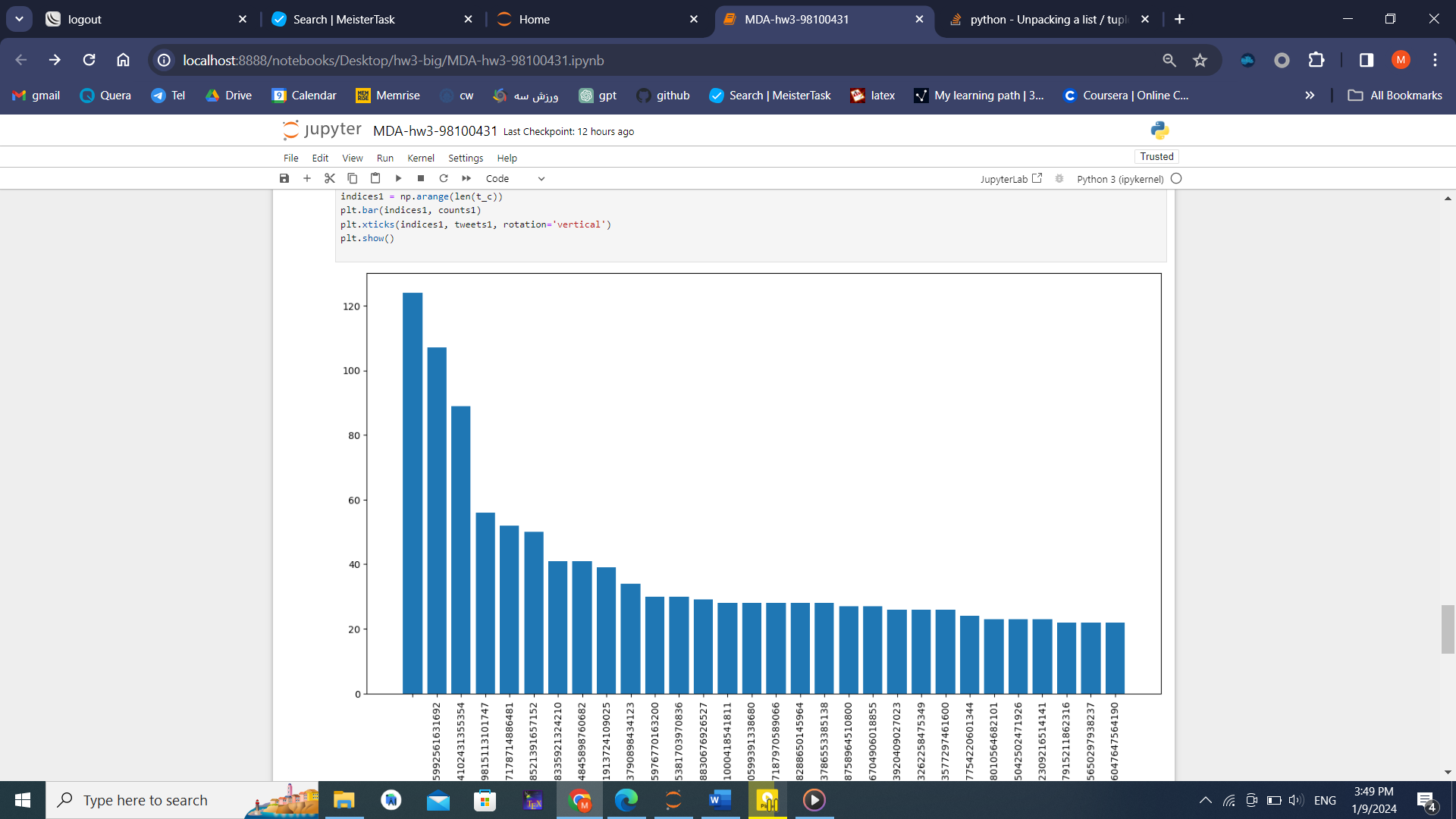
User and their tweets indexing. ( user\_id, ( tweets-index , count ) ) :



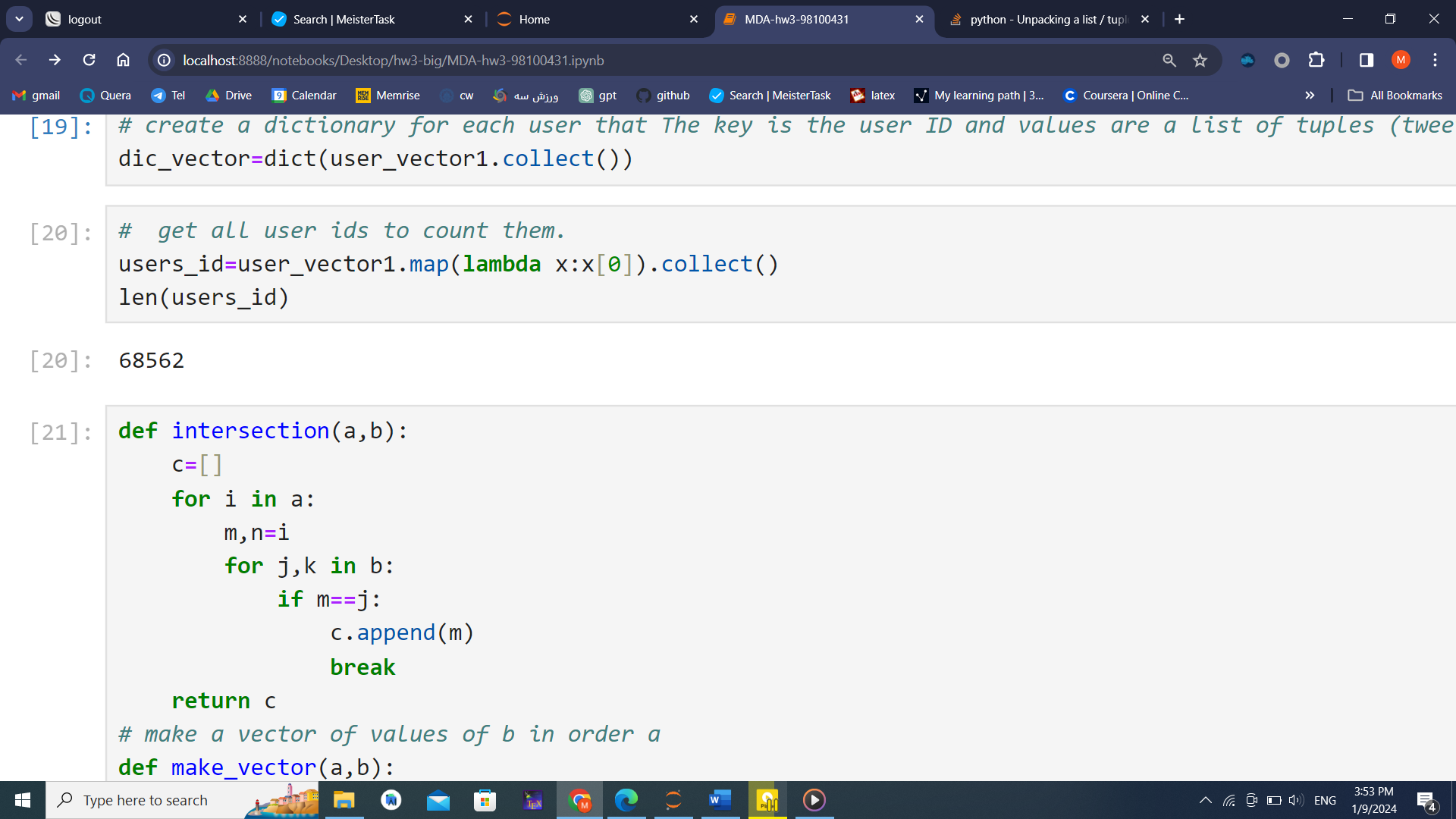
A barplot of user-N(tweets)



A barplot of tweets-N(relation):

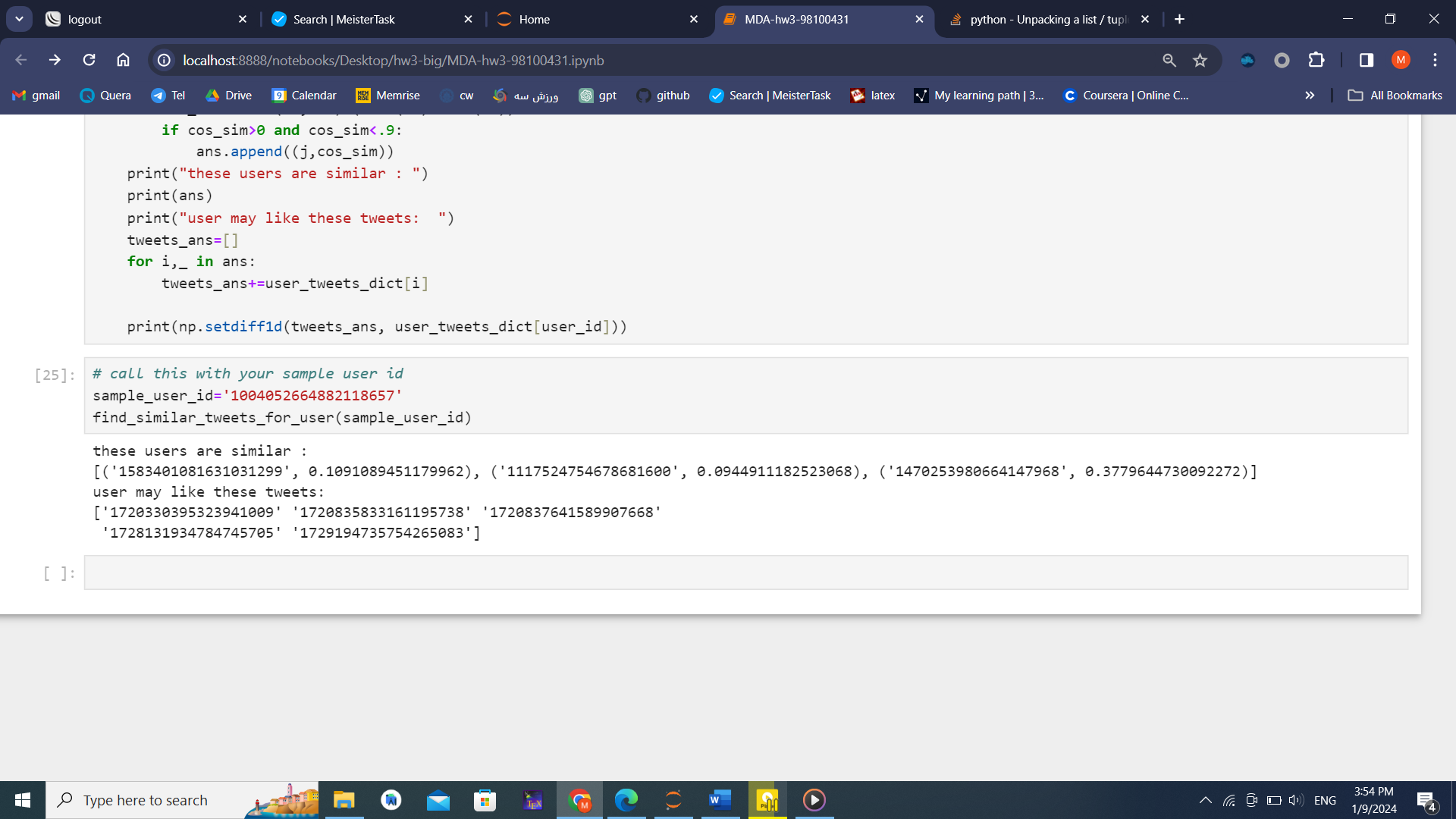


Number of users:



Main function: print users that are similar to sample users with cosine similarity.

And print id of tweets that are my be liked by user.



1. Does the algorithm need to obtain the model from the beginning for a new user who is added to this matrix?(online or offline)

NO. In this methods, new row is created for new users and then can compare it to others and finally, recommend tweets. We do not need to retrain model and do all computation again. We just do computation for the new user.