Hw2 : LSH

Course: Massive Data Analytics

Proffesor: Dr. Iman Gholampoor

Mohammad Mehdi Zare

**Introduction** :

In this assignment, I implemented the Minhash\_LSH algorithm to find similar news in news data with Pyspark and RDD API.

This algorithm has three steps:

1. Shingling
2. Minhash
3. LSH

* **PreProccess**:

Like HW1, I remove stopwords and punctuation from the news in this part. These words can cause some problems and errors. Also, they can increase false positives and false negatives, which decrease the accuracy of algorithms because by changing the structure of sentences and grammar, people can say similar news.

Therefore, removing punctuations and stopwords is very helpful to find similar news accurately and has less computation.

* **Shingling**:

In this step, I map each news to its UID and body, then, shingling its body by k=7.

Why k=7? Because I have a lot of news if I choose a small k, the large amount of news will be similar. However, if I choose a bigger k few news will be similar and may lose some similar news.

Finally, I collect news shingles in a unique list.

* **Minhash**:

First I make a dictionary for shingles and indexes. And make a hash family. For the hash family, I use universal hash functions because they are easy to work and are good enough for our purpose.

I take a document shingles and then use my hash family to find a signature vector for that document. Then go to the next document and so on.

In this approach, I read the shingle matrix once.

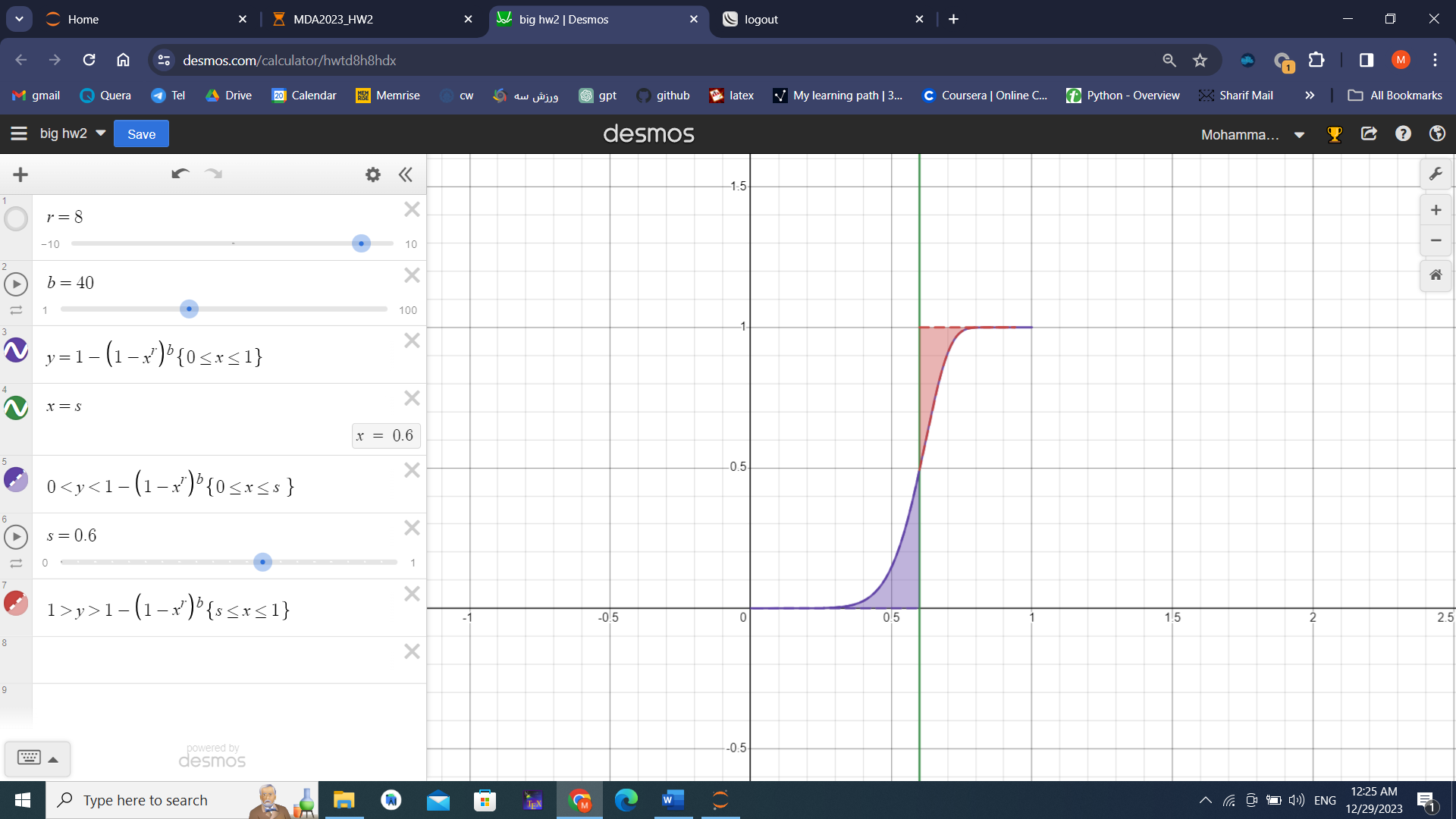
(M: matrix size, h= hash family size )

O(minhash)=M\*h

At the end, I have a signature matrix with size d\*h and d is the number of documents.

**Why h = 320?**

According to the below graph in the desmos, I found that for similarity =.6, the 320 hash function can be good. Then r=8 and b =40 are my parameters. With these parameters false positives and false negatives are small and the S-curve is very similar to the goal curves, which we learned in class.



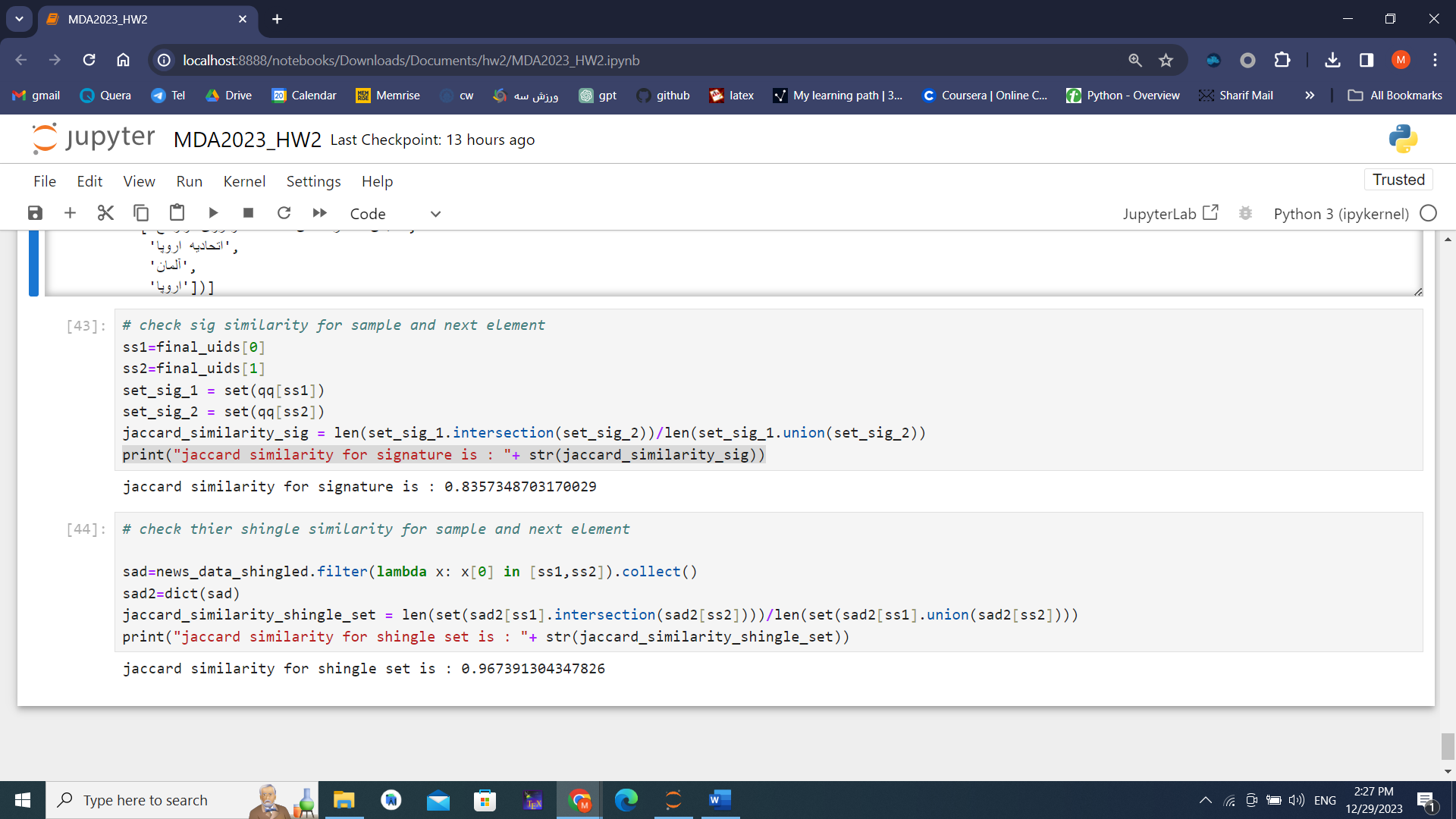
<https://www.desmos.com/calculator/hwtd8h8hdx>

* **LSH**:

As I said before, I chose b= 40 bands and in each band, I have 8 rows. For each sample Uid, I compare its signature vector with the AND\_OR approach. I mean for each band, if the sample and target have the same rows, I consider them as a pair candidate. Also, I check similarity again by using Jaccard similarity for documents.

For example. For sample uid = e13ce3d66402beff728480ff8





Best regard