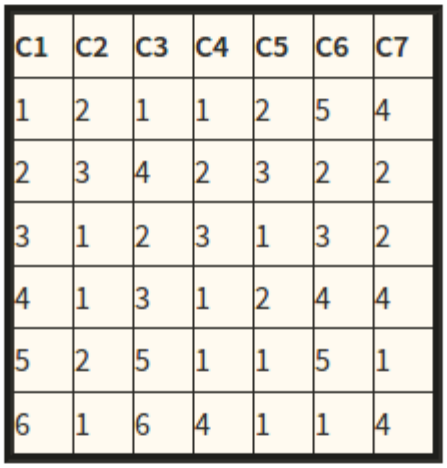
Locality-Sensitive Hashing

Question 1:

Here is a matrix representing the signatures of seven columns, C1 through C7.



C1	C2	C3	C4	C5	C6	C7
1	2	1	1	2	5	4
2	3	4	2	3	2	2
3	1	2	3	1	3	2
4	1	3	1	2	4	4
5	2	5	1	1	5	1
6	1	6	4	1	1	4

Suppose we use locality-sensitive hashing with three bands of two rows each. Assume there are enough buckets available that the hash function for each band can be the identity function (i.e., columns hash to the same bucket if and only if they are identical in the band). Find all the candidate pairs.

Ans:

(C1,C4), (C2,C5), (C1,C6), (C1,C3), (C4,C7)

Question 2:

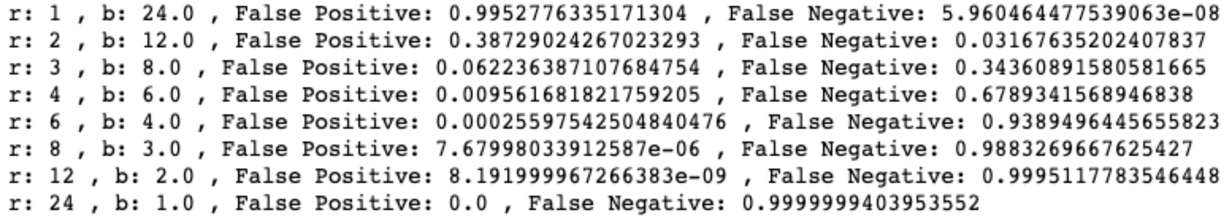
Suppose we have computed signatures for a number of columns, and each signature consists of 24 integers, arranged as a column of 24 rows. There are N pairs of signatures that are 50% similar (i.e., they agree in half of the rows). There are M pairs that are 20% similar, and all other pairs (an unknown number) are 0% similar.

We can try to find 50%-similar pairs by using Locality-Sensitive Hashing (LSH), and we can do so by choosing bands of 1, 2, 3, 4, 6, 8, 12, or 24 rows. Calculate approximately, in terms of N and M, the number of false positive and the number of false negatives, for each choice for the number of rows.

Then, suppose that we assign equal cost to false positives and false negatives (an atypical assumption). Which number of rows would you choose if M:N were in each of the following ratios: 1:1, 10:1, 100:1, and 1000:1?

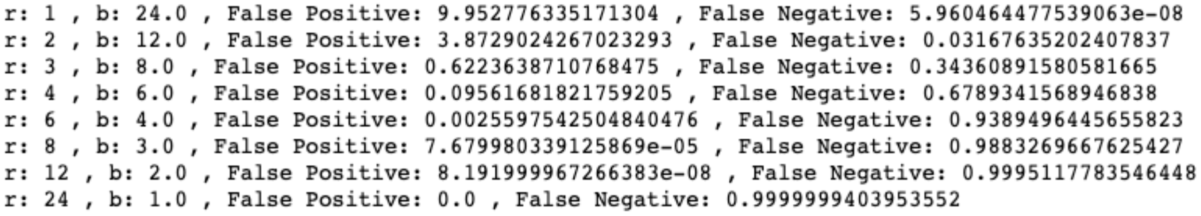
Ans:

M:N = 1:1 (row 3 would be chosen)



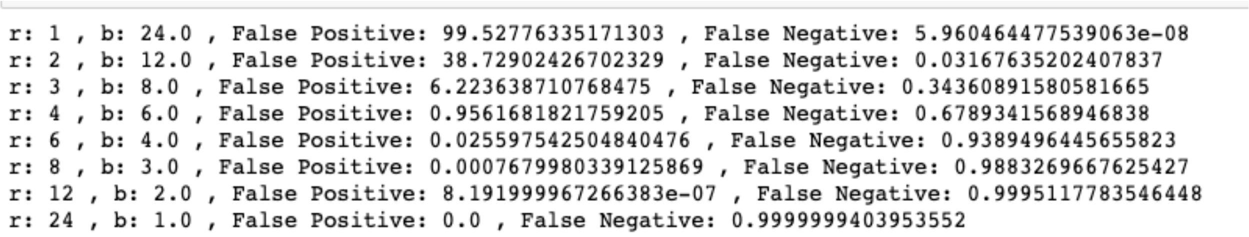
```
r: 1 , b: 24.0 , False Positive: 0.9952776335171304 , False Negative: 5.960464477539063e-08
r: 2 , b: 12.0 , False Positive: 0.38729024267023293 , False Negative: 0.03167635202407837
r: 3 , b: 8.0 , False Positive: 0.062236387107684754 , False Negative: 0.34360891580581665
r: 4 , b: 6.0 , False Positive: 0.009561681821759205 , False Negative: 0.6789341568946838
r: 6 , b: 4.0 , False Positive: 0.00025597542504840476 , False Negative: 0.9389496445655823
r: 8 , b: 3.0 , False Positive: 7.67998033912587e-06 , False Negative: 0.9883269667625427
r: 12 , b: 2.0 , False Positive: 8.191999967266383e-09 , False Negative: 0.9995117783546448
r: 24 , b: 1.0 , False Positive: 0.0 , False Negative: 0.9999999403953552
```

M:N = 10:1 (row 4 would be chosen)



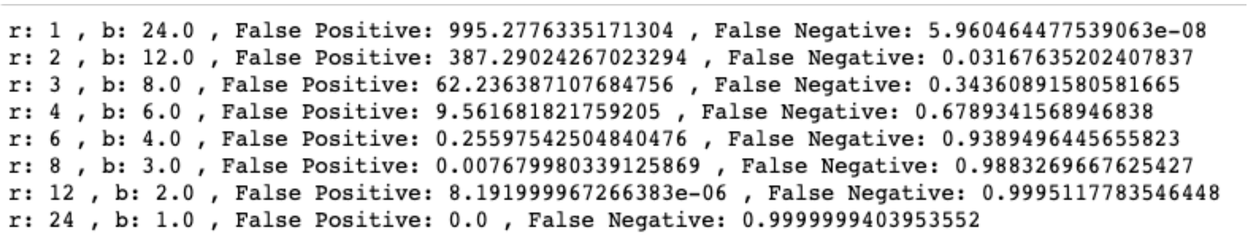
```
r: 1 , b: 24.0 , False Positive: 9.952776335171304 , False Negative: 5.960464477539063e-08
r: 2 , b: 12.0 , False Positive: 3.8729024267023293 , False Negative: 0.03167635202407837
r: 3 , b: 8.0 , False Positive: 0.6223638710768475 , False Negative: 0.34360891580581665
r: 4 , b: 6.0 , False Positive: 0.09561681821759205 , False Negative: 0.6789341568946838
r: 6 , b: 4.0 , False Positive: 0.0025597542504840476 , False Negative: 0.9389496445655823
r: 8 , b: 3.0 , False Positive: 7.679980339125869e-05 , False Negative: 0.9883269667625427
r: 12 , b: 2.0 , False Positive: 8.191999967266383e-08 , False Negative: 0.9995117783546448
r: 24 , b: 1.0 , False Positive: 0.0 , False Negative: 0.9999999403953552
```

M:N = 100:1 (row 6 would be chosen)



```
r: 1 , b: 24.0 , False Positive: 99.52776335171303 , False Negative: 5.960464477539063e-08
r: 2 , b: 12.0 , False Positive: 38.72902426702329 , False Negative: 0.03167635202407837
r: 3 , b: 8.0 , False Positive: 6.223638710768475 , False Negative: 0.34360891580581665
r: 4 , b: 6.0 , False Positive: 0.9561681821759205 , False Negative: 0.6789341568946838
r: 6 , b: 4.0 , False Positive: 0.025597542504840476 , False Negative: 0.9389496445655823
r: 8 , b: 3.0 , False Positive: 0.0007679980339125869 , False Negative: 0.9883269667625427
r: 12 , b: 2.0 , False Positive: 8.191999967266383e-07 , False Negative: 0.9995117783546448
r: 24 , b: 1.0 , False Positive: 0.0 , False Negative: 0.9999999403953552
```

M:N = 1000:1 (row 8 would be chosen)



```
r: 1 , b: 24.0 , False Positive: 995.2776335171304 , False Negative: 5.960464477539063e-08
r: 2 , b: 12.0 , False Positive: 387.29024267023294 , False Negative: 0.03167635202407837
r: 3 , b: 8.0 , False Positive: 62.236387107684756 , False Negative: 0.34360891580581665
r: 4 , b: 6.0 , False Positive: 9.561681821759205 , False Negative: 0.6789341568946838
r: 6 , b: 4.0 , False Positive: 0.25597542504840476 , False Negative: 0.9389496445655823
r: 8 , b: 3.0 , False Positive: 0.007679980339125869 , False Negative: 0.9883269667625427
r: 12 , b: 2.0 , False Positive: 8.191999967266383e-06 , False Negative: 0.9995117783546448
r: 24 , b: 1.0 , False Positive: 0.0 , False Negative: 0.9999999403953552
```

Question 3:

Find the set of 2-shingles for the "document":

ABRACADABRA

and also for the "document":

BRICABRAC

Answer the following questions:

1. How many 2-shingles does ABRACADABRA have?
   Ans:
   set of 2-shingles = {AB,BR,RA,AC,CA,AD,DA}
   Total 7 set of 2-shingles

2. How many 2-shingles does BRICABRAC have?
   Ans:
   Set of 2-shingles = {BR,RI,IC,CA,AB,RA,AC}
   Total 7 set of 2-shingles

3. How many 2-shingles do they have in common?
   Ans:
   Set of 2-shingles in common = {AB, BR, RA, AC, CA}
   Total 5 2-shingles are in common

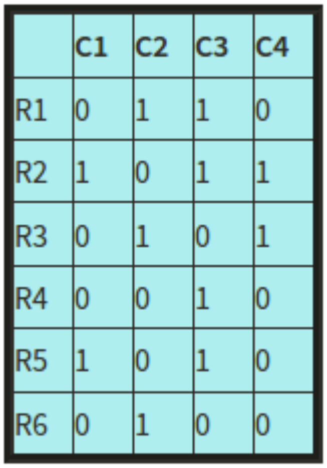
4. What is the Jaccard similarity between the two documents"?

   Ans:

   sim(d1,d2) = 5 / 9

Question 4:

Consider the following matrix:



	C1	C2	C3	C4
R1	0	1	1	0
R2	1	0	1	1
R3	0	1	0	1
R4	0	0	1	0
R5	1	0	1	0
R6	0	1	0	0

Compute the Jaccard similarity between each pair of columns.

Ans:

sim(c1,c2) = 0 / 5 = 0

sim(c1,c3) = 2 / 4 = 0.5

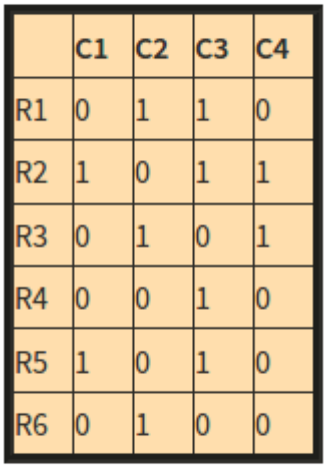
sim(c1,c4) = 1/ 3 = 0.33

sim(c2,c3) = 1 / 6 = 0.17

sim(c2,c4) = 1 / 4 = 0.25

sim(c3, c4) = 1 / 5 = 0.2

Question 5: Consider the following matrix:



	C1	C2	C3	C4
R1	0	1	1	0
R2	1	0	1	1
R3	0	1	0	1
R4	0	0	1	0
R5	1	0	1	0
R6	0	1	0	0

Perform a minhashing of the data, with the order of rows: R4, R6, R1, R3, R5, R2.

Note: we give the minhash value in terms of the original name of the row, rather than the order of the row in the permutation. These two schemes are equivalent, since we only care whether hash values for two columns are equal, not what their actual values are.

Ans:

C1	C2	C3	C4
R5	R1	R3	R1