Jeremy Midvidy, jam658
EECS 349, Fall 2017
Assignment 4

Problem 1

A.

I loaded the u_data file into a python script with the standard CSV reader. From there I made a list of all the elements in the u_data file (100,000 elements). Next, I made a dictionary of all the UserID's (943 total) and each element at each key (1-943) was a list of all the movies that each user had rated (out of 100,000 total instances.) Then I made a list of pairs where each element was a length of the intersection of the list of movies watched by each pair. With this list I calculated the mean, the median, and also the frequencies of each. The frequency represented the number of times a pair had had the same size intersection set. I then graphed this, which is seen below.

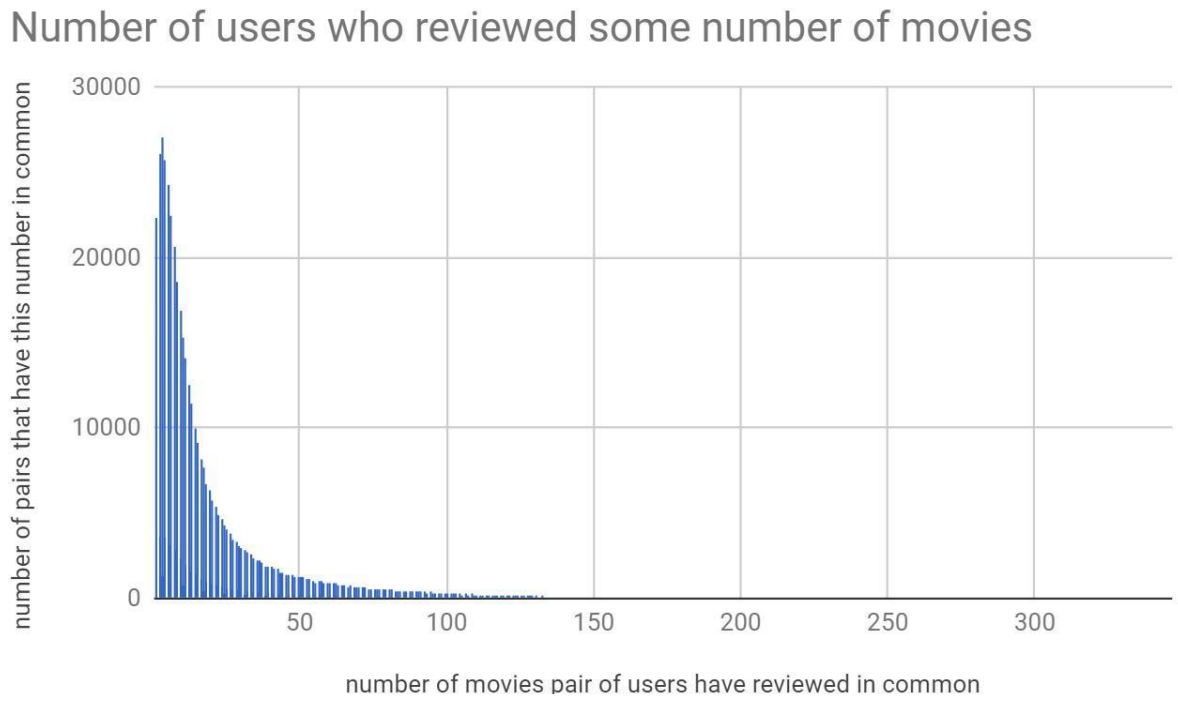
The mean number of of movies two people had seen in common was:

mean = 19.4672578127

The median number of movies two people had seen in common was:

median = 10

And the frequencies graph:



All of the python code I wrote for iterating of the groups, making new lists, getting the meaning of the data, and my output used to making the graphs is in **Appendix 1A.**

B.

Like part a, I loaded the u_data file and made a dictionary of 1682 keys, from 1-1682 each representing a movieID key. Each key's initial value was zero. I then iterated over the 100,000 and increased the value of each key in each time it was reviewed in the 100,000 instances. Each increase in key value represented an increase by 1 of the number of times that movie had been reviewed. With my dictionary established, I then used a small function to find the most reviewed and least reviewed films. Using my dictionary, I then made a list of length 1682 elements. Each element was a list of two values. The first value was the movieID (1-1682) and the second was the number of times that value had been positioned. I then ordered the id-list, ascending, by the number of times each movie had been reviewed, so that the list was now sorted in order of number of times reviewed. Each element was still a two number list, but each element's position in the overall was sorted as required by the problem. I then made a line graph of the ordered list.

The max reviewed film/number was:

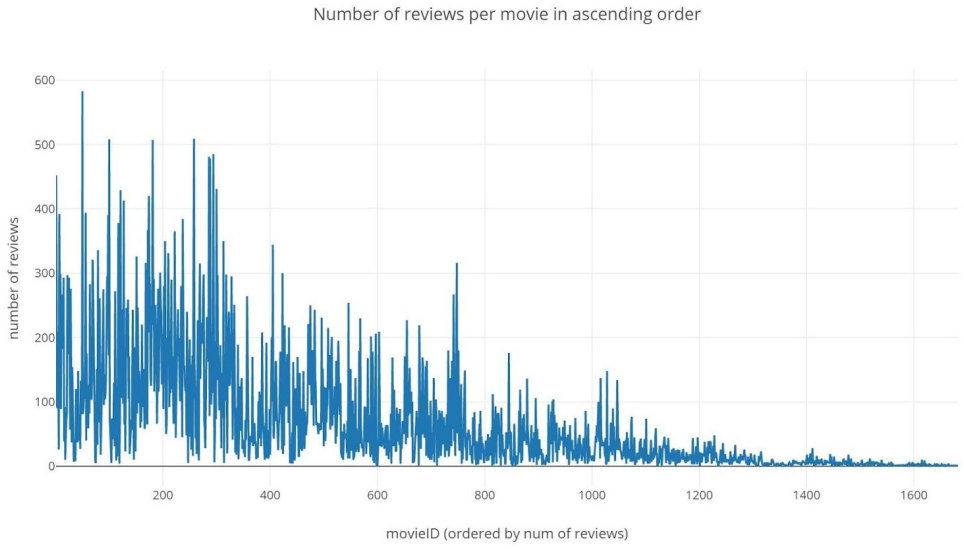
movieID: 50
numberOfReviews: 583

The min reviewed film/number was*:

movieID:598
numberOfReviews: 1

*there were many others with only 1 review, as seen in the graph.

And the graph of number of reviews over ordered movie number was :



With this data, I do think the number of reviews per movie follows Zipf's law. Zipf's law applies both graphically and logically in this instance because the frequency of a movieID being reviewed decreased linearly with 1/N. If you look at the above graph, it decreases with what seems inversely proportional way. As the ordered list of goes from X0->X1600, the frequency of that x-values (or movieID) number of reviews in the data set decreases at an inversely proportional manner. Thus, Zipf's law applies.

All of my python code for finding the max, min, and data for the graph can be found in **Appendix 1B.**

Problem 2

A.

userId/moveiD	m1	m2	m3	m4	m5
A	3	-	3	3	-
B	-	5	1	-	3
C	4	4	-	4	-

Let's say we know that A/m2 rating is 2, and we are trying to find a prediction for it using this table.

If we set unknown = 0 we get data

userId/moveiD	m1	m2	m3	m4	m5
A	3	0	2	2	0
B	0	5	1	0	3
C	4	4	0	4	0

Using manhattan distance:

A;B: 3 + 5 + 1 + 2 + 3 = 14
A;C: 1 + 4 + 4 + 2 + 0 = 11

So we would use user's C's score of m2 and predict Am2 = 4.

If we set unknown = average.

Aavg = 2.33
Bavg = 3
Cavg = 4

userId/moveiD	m1	m2	m3	m4	m5

A	3	2.33	2	2	2.33
B	3	5	1	3	3
C	4	4	4	4	4

Manhattan distance:
A;B: 0 + 1.67 + 1 + 1 + .67 = 4.34
A;C: 1 + 1.67 + 2 + 2 + 1.67 = 6.34

So we use Bm2 to predict Am2, and predict a score of 5.

So using zeros we get a prediction of 4 and using averages we get a prediction of 5. Since the actual rating was 2, it would be a better idea to use zeroes instead of using averages.

This makes sense because we can't always be sure what the classifiers indicate: in this example lower numbers meant worse reviews and higher numbers meant better reviews. In other examples, the values of the numbers mean different things. When we use zeroes, all of the unknowns get integrated into the problem without affecting the overall score of the distance ratings between sets, however, if we were to use averages, the overall distance rating between sets would change because the average would change the distance from zero of all the elements calculated from manhattan distance. Whether this change is in the positive or negative direction would influence the ultimate predicted recommended by the collaborative filter, so for this reason using averages is not the best approach.

Using zeroes is the better approach when using manhattan distance, thus, because it does not shift values off the number line.

B.

I would think that pearson correlation would be the better distance measure for item-based collaborative filtering. It would make better sense to use pearson measure when all of the missing info is 3, because replacing missing info-with non-zero changes each calculations distance from the number line. Since the pearson distance gives you the correlation between two vectors, and not just the plain manhattan distance, it gives you a much more precise indicator of how close each vector is to one another. Since we are shifting numbers from the number line, we need an indication of how each vector correlates to another not just the distance between vector elements.

Given the vector set:

a = [2,3,4,5,3]
b = [3, 5, 1, 3, 3]
c = [4,4,3,4,3]

ab = sp.spatial.distance.cityblock(a,b)
ac = sp.spatial.distance.cityblock(a,c)

pab = sum(sp.stats.pearsonr(a, b))
pac = sum(sp.stats.pearsonr(a, c))

Returns:

Ab = 8
Ac = 6

Pab = 0.301520047672
Pac = 0.952305859311

When we change from the number line, as Ac has the lower manhattan distance, and the collaborative filter would have recommend Cm2 for Am2 which would have been 4.

With pearson correlation, AB had the lower correlation, and the collaborative filter would have recommended Bm2 for Am2 which would have been 5.

Thus, when we shift off the number line it is better to use pearson rather than manhattan distance.

Problem 4:

A.

Error Measure = abs(PredictedRating - ActualRating)

I chose this error measure because it shows the distance between the collaborative filter's predicted error for each draw (100 draws per sample.) Since the ratings are between 1-5, this measure is a good indication of how close the filter's rating was to the actual rating of each draw. If the rating were out of 10, perhaps I would have chose a more nuanced statistical measure like standard deviation, because distances might be a little larger and thus hard to extract trends from, but given the small range of the classifier set, pure distance measure felt like a good way to indicate how well the collaborative filter was performing.

B.

In subsequent experiments, I will average the error measure over all the sample and use that average to make conclusions about which variant of parameter/filter-type was the most optimal for the given data. I chose this test because my error test measures the distance between the predicted rating and the actual rating, so to get a good indication of how the filter performed on each sample, I took an average of the distances in each sample set. So for 50 samples, each of 100 draws, the average for each sample is the mean of 100 error measures. And then for the entire experiment, the average would be, again, the mean of all (50) of the sample averages. This provides one number to weigh against other experiments, whose averages would be calculated in a similar manner.

The null hypothesis would be a sample where there were no distances to measure because the requested movieID of the test-case in draw in sample (100 draws) was not reviewed by a single member of the data set (the other 99,900 instances of reviews). In this case, the collaborative filter would not be able to provide a recommendation because it had no relevant information to extract from, and return a predicted rating of 0.

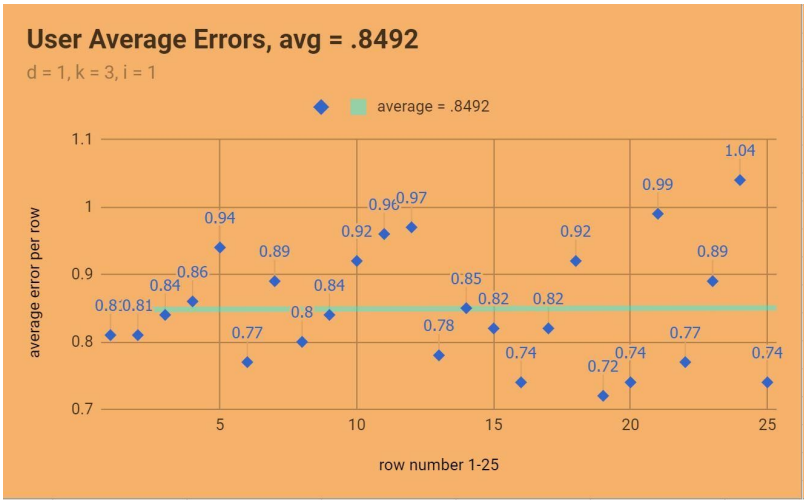
For my confidence interval, if my p-val is less than .5 I will consider the data valid, else, invalid.

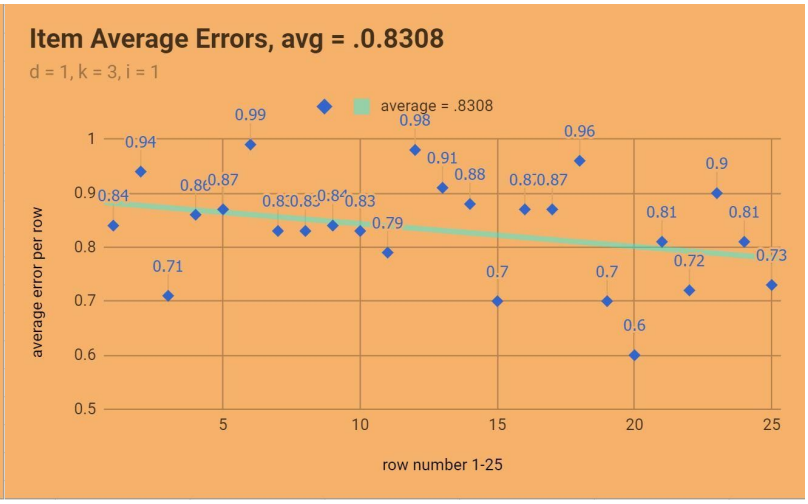
To run my experiment, I made a separate experiment python file (found in the **Appendix**) which contained slightly altered versions of user_cf and item_cf which were submitted along with this report.

First I made 50 arrays of 100 random indexes in range(0,100,000). I created an array of 50 rows, one for each sample, and each row had 100 draws. Each draw was one of the random indexes in the proper range, so each sample had 100 random draws, and the set had 50 samples of length 100 each.

I then constructed the appropriate priorData for each sample. To do this, I created a list of 50 rows. Each row was all of the prior data that corresponded to each sample row. Each row in prior data was 99,900 instances of a userId | moveId | review. Each row of prior data was the 99,900 instances that were not in the sample row. So there were 50 different sets of prior data, one for each corresponding sample row that did not contain the 100 corresponding review-instances that were in each sample row.

With the testing data and prior data constructed, I ran the experiment for the first 25 rows user_based and the second 25 rows item_based, with initial paramters d=1(mahattan), i = 1, k = 3. Results are below.



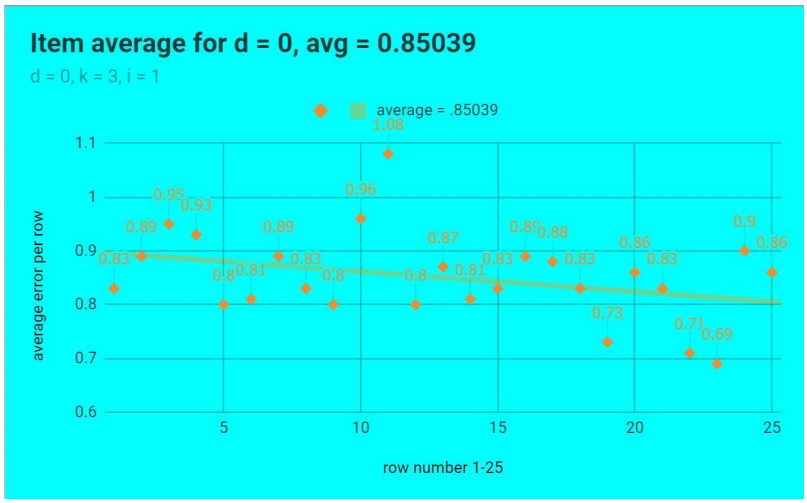


As illustrated, the average error for all of the samples in user-based was .8492 and the average error for item-based was .8309.

Reminder: The larger the average error, the higher the average error. So with these initial parameters, item-based performed slightly better than user-based.

C.

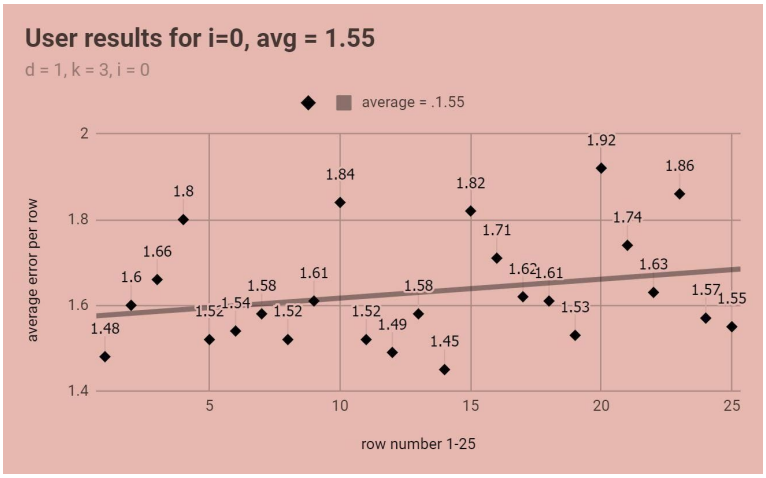
When the choice for <distance> was changed, from 1=manhattan in the above, to 0=pearsonr in the below, holding all other parameters constant, the item based collaborative



filter performed worse than it did initially. At first, the average error over all the sample was .8308 and when distance was changed from manhattan to pearson, the average error over all the samples increased from .8308 (d=1) to .8503 (d=0). This corresponds to my intuition from problem two because are using zeroes to replace unknown data. Like I stated in problem 2, if we are using 0s as replacement, it would be better to use manhattan distance than pearson correlation. This makes sense because if we use 0 unknowns, the manhattan distance calculations do not affect the predicted measure because we do not shift calculations of the number line (we would shift if we used a non-zero unknown value like 3.) So, in this case, it made sense that manhattan distance performed slightly better than pearson correlation.

D.

When the parameter of of <i> was changed, from i=1 (allow all users) to i=0 (allow only users those that have reviewed the particular movie for user, and movies that have rated by the particular users for item), the user-based collaboration performed worse.



As seen above, the user-based collaborative filter for when i=0 had an average measure error (over the 100 draws per sample) of avg=1.55. This is an increase from .8492, which was the result for the same data (25 samples of 100 draws), the same prior data (each set of 99,900 instances of reviews for the appropriate each of the 1000 draws in each of the 25 sample sets), and with all other parameters holding constant besides <i>.

This is what I would have expected. When i=0, and we have to screen each example from whether it can be used to evaluate a prediction, we drastically reduce the size of the data set. For user-based, we have to delete all instances of other users that have not reviewed the particular movieID that this userID-movieID draw from a sample was being operated on by the filter. If the inputted user and and a certain user in the testing set of 99,900 reviewed 200 movies in common, BUT not the specific movie that was being inputted in the userID-movieID draw case, then that entire user, and all of the 200 data points would be thrown out. It would have made more sense to assign a rating of zero for the rating of the user in the testing set rather than throw out 200 data points worth of movies that both users have reviewed. This is why setting i=0 made the data less precise, because there is a good chance that we removed a lot of useful userIDs from the user-based collaborative filter when i=0, and thus the users that remained were less similar to the test-case user, so our prediction from the dwindled data set was less precise in predicting the actual review.

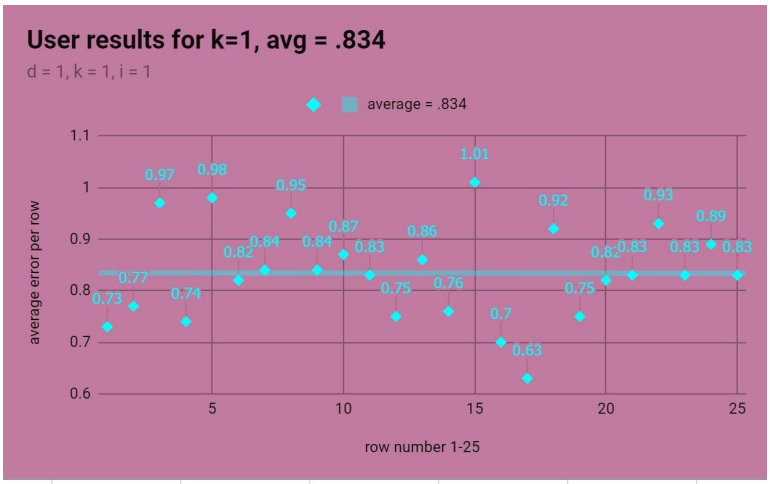
D.

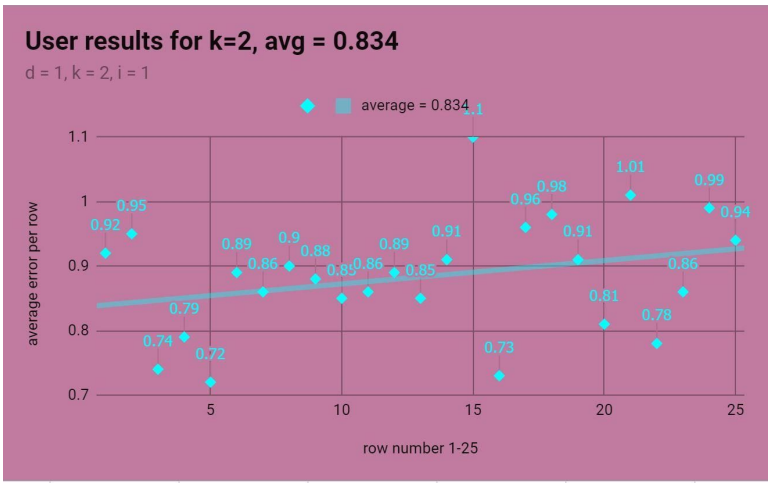
With optimal parameters d=1(manhattan distance), i = 1, learned from parts C and D, varying <k> in the set {1, 2, 4, 8, 16, 32} produce these results:

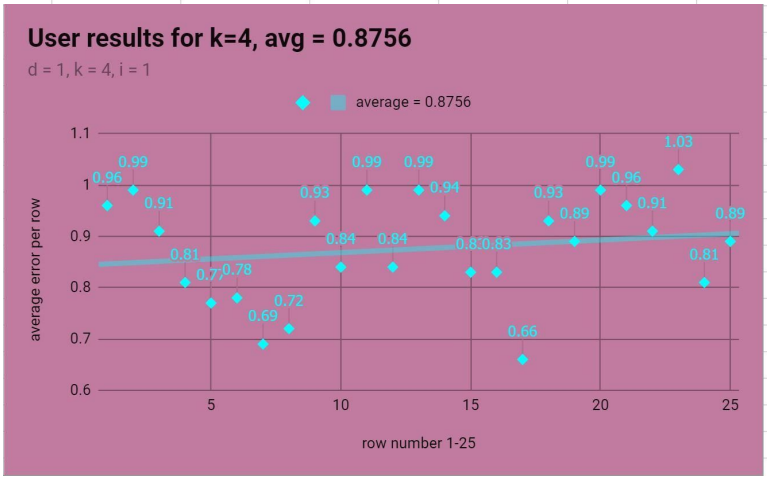
k	**Average sample error**
1	.834
2	.834
4	.8756
8	.9112
16	.9156
32	.8604

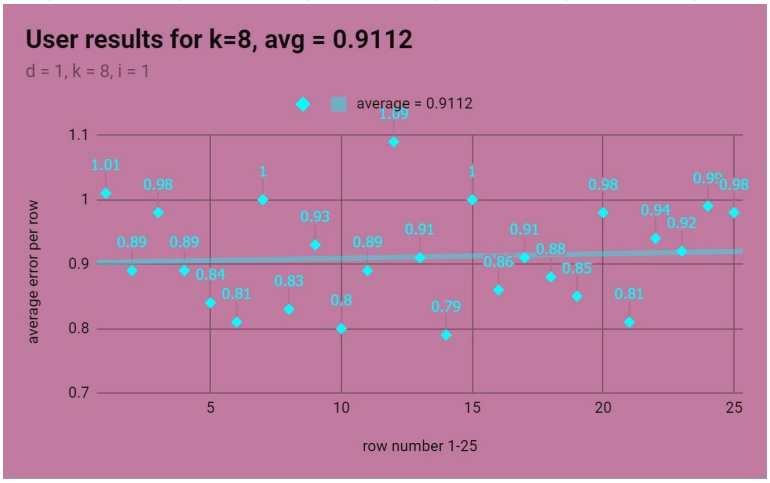
As seen in the above table, k = 1, k = 2 tied for the best performance. Then was k = 32, k = 4, k = 8, and k = 16.

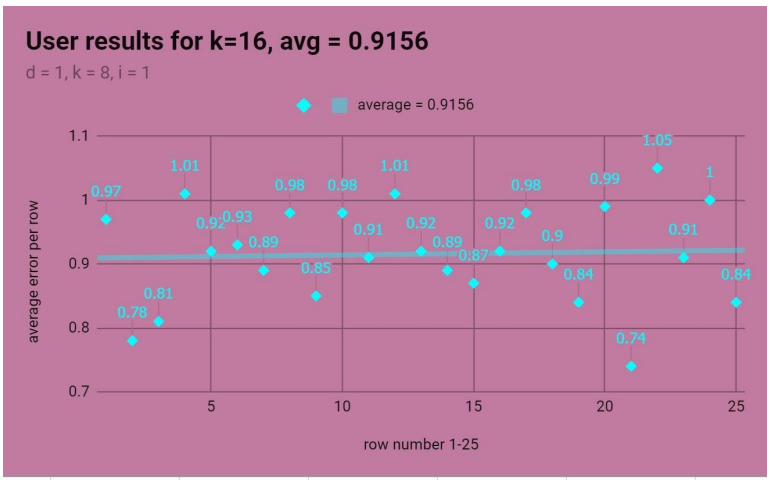
The performances are illustrated in these graphs:

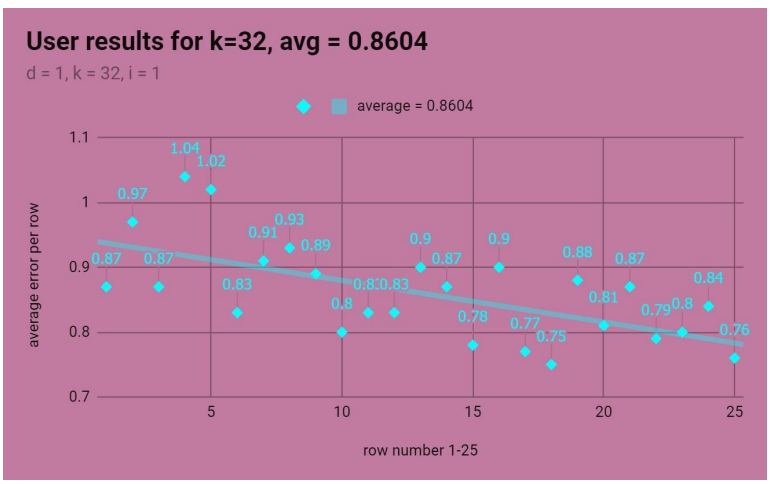








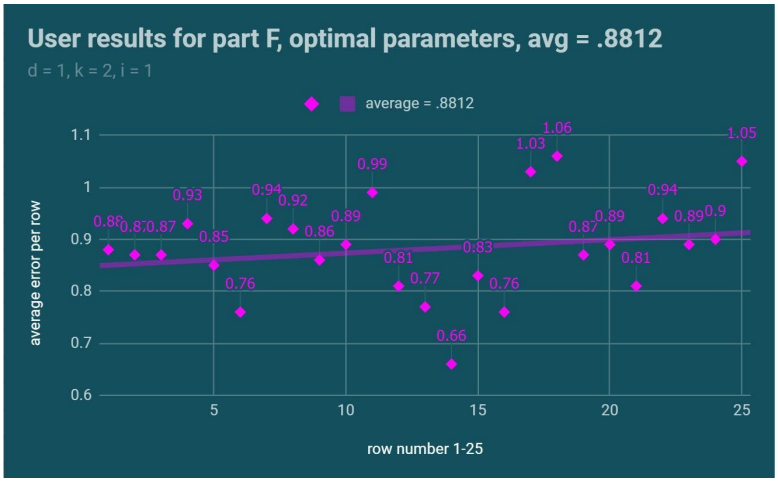


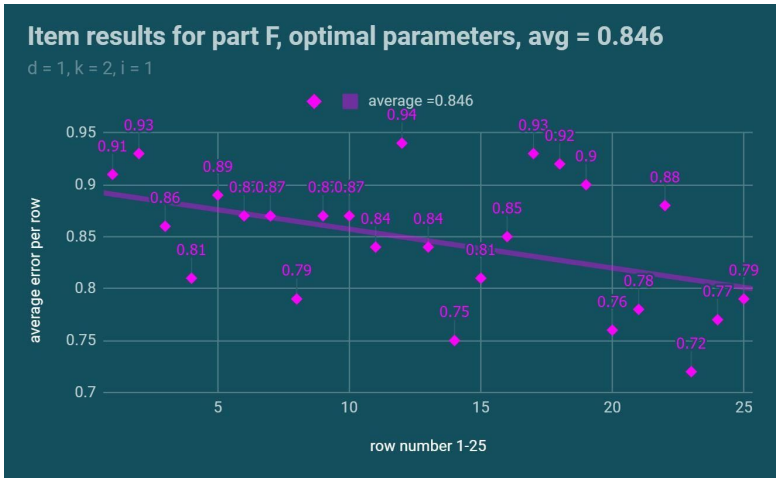


All of my code for the experiment, data for the graphs, and sample output when my program ran can be found in **Appendix 4.**

F.

With optimal parameters k = 2, i = 1, d = 1, item and user returned:





In this problem set, item based filtering is better. I am very confident in this result considering we optimized the parameters through thousands of trials, and each individual trial involved a unique data set of 100,000 members each. That is a lot of testing to arrive at these results.

It makes sense that item-based filtering is better because we know a lot of information about the items -movies and their reviews-but we know very little information about the users *besides* which movies they have watched. While user-based cf works well -error about .8812 over 5000 trials- item-based cf works significantly better - error .8416 over 5000 trials. Filtering information through movies is better with this data set than filtering through users, as is illustrated in my final results since we know more about the movies (what reviewed they were scored) and almost nothing about the user's preferences *other* than their movies and reviews. So if we are trying to predict a review, it makes more sense to aggregate other user's reviews of the movie and then use that mode than try and aggregate similar users to the inputted user and users the mode of the top-k most similar users (even though it worked OK.)

Problem 1A:

Python Script:

```
import csv
import numpy as np, scipy as sp, matplotlib
import copy
import random

u_data_Path = 'C:\Users\jmidv\Miniconda2\envs\eecs349\u_data.txt'

def makePairs(inputFile):

    #construct dictionary for WikipediaTypo
    with open(inputFile,'r') as filereader:
            #sort the input file with the csv reader and the \t delimiter
            fileListReader = csv.reader(filereader, delimiter='\t')

            #place holder for dictionary to be created in loop
            fileList = []
            for row in fileListReader:
                d = row
                for x in range(0, len(d)):
                    d[x] = int(d[x])
                fileList.append(d)


    #construct a dictionary of 943 elements, key range from 1-943 based
    #on inputted userID's

    numWatched = dict()
    for x in range(1, 944):
        numWatched[x] = []

    ##for each userID, add the list of movies it has watched
    for x in range(0, 100000):
        ind = fileList[x][0]
        r = [fileList[x][1]]
        newVal = numWatched[ind] + r
        numWatched[ind] = newVal


    ##construct pairs
    ##each element in pairs represents the unions of the
    #lists of movies watched by each user at numWatched[x] and numWatched[x+1]
    pairs = []
    for x in range(1, 944):
        for y in range(x+1, 944):
            a = len(list(set(numWatched[x])&set(numWatched[y])))
            if(a != 0):
                pairs.append(a)

    print(len(pairs))




    #find mean and median with numpy witht the pairsNumInSet lsit
    mean= np.mean(pairs)
```

```
    median = np.median(pairs)


    print("\n")
    print("The mean number of movies two people had reviewed in common is:")
    print("\n")
    print(mean)
    print("\n")
    print("The median number of movies two people had reviewed in common is:")
    print("\n")
    print(median)
    print("\n")

    counts = dict()
    for row in pairs:
        if(row in counts):
            counts[row] = counts[row] + 1
        else:
            counts[row] = 1

    keys = []
    nums = []

    for key in counts:
        keys.append(key)
        nums.append(counts[key])



    print("\n")
    print(keys)
    print("\n")

    print("\n")
    print(nums)
    print("\n")


makePairs(u_data_Path)
```

Output Generated, which was then used to make the graph:

```
The mean number of movies two people had reviewed in common is:


19.4672578127


The median number of movies two people had reviewed in common is:


10.0




[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48,
49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71,
72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
```

```
95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114,
115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133,
134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152,
153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171,
172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190,
191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209,
210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228,
229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247,
248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266,
267, 268, 269, 271, 273, 274, 276, 277, 278, 279, 281, 282, 283, 285, 287, 288, 290, 292, 293,
294, 299, 300, 301, 303, 305, 311, 314, 315, 318, 319, 327, 332, 334, 335, 346]
```

```
[22359, 26088, 27027, 25725, 24279, 22423, 20617, 18598, 16941, 15307, 14062, 12483, 11413,
10029, 9148, 8215, 7668, 6753, 6410, 5800, 5370, 4925, 4716, 4309, 4069, 3766, 3491, 3296,
3059, 2973, 2874, 2764, 2575, 2372, 2277, 2227, 2113, 1930, 1838, 1814, 1768, 1720, 1562,
1450, 1438, 1443, 1379, 1311, 1227, 1247, 1227, 1171, 1093, 1075, 958, 971, 997, 887, 872,
920, 870, 846, 798, 780, 795, 697, 728, 693, 666, 678, 679, 616, 567, 596, 535, 565, 547, 524,
567, 524, 484, 467, 467, 466, 417, 407, 372, 395, 376, 369, 376, 381, 345, 361, 333, 338, 277,
300, 299, 279, 268, 275, 266, 244, 227, 250, 216, 244, 215, 199, 214, 170, 187, 188, 195, 182,
185, 145, 149, 140, 144, 135, 153, 154, 135, 144, 137, 132, 116, 121, 110, 133, 110, 107, 114,
106, 75, 94, 103, 82, 90, 77, 65, 88, 71, 65, 79, 58, 69, 56, 70, 73, 60, 56, 60, 61, 61, 52,
52, 53, 44, 51, 60, 50, 42, 45, 35, 35, 30, 32, 30, 26, 49, 26, 27, 27, 35, 24, 39, 16, 27,
29, 30, 23, 29, 22, 24, 29, 14, 31, 18, 14, 18, 8, 11, 18, 15, 18, 15, 16, 18, 17, 18, 12, 15,
19, 9, 11, 15, 14, 12, 14, 10, 7, 8, 9, 5, 6, 9, 4, 8, 11, 9, 8, 14, 5, 6, 5, 4, 3, 6, 12, 6,
9, 5, 8, 6, 1, 3, 2, 4, 5, 7, 5, 3, 1, 2, 7, 3, 4, 2, 2, 3, 3, 4, 4, 7, 3, 3, 2, 2, 2, 4, 3,
2, 2, 1, 3, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 2, 2, 1, 1, 1]
```

Problem 2B)

Python Script

```
"""
Created on Tue Oct 24 14:08:23 2017

@author: Jeremy Midvidy
"""

import csv
import numpy as np, scipy as sp, matplotlib as plt
import copy
import random

u_data_Path = 'C:\Users\jmidv\Miniconda2\envs\eecs349\u_data.txt'

def makePairs(inputFile):

    #construct dictionary for WikipediaTypo
    with open(inputFile,'r') as filereader:
            #sort the input file with the csv reader and the \t delimiter
            fileListReader = csv.reader(filereader, delimiter='\t')

            #place holder for dictionary to be created in loop
            fileList = []
            for row in fileListReader:
```

```
            d = row
            for x in range(0, len(d)):
                d[x] = int(d[x])
            fileList.append(d)


#construct a dict of 1682 elements, keyed from range from 1-1682 based
#on inputted movieID's
numWatched = dict()
for x in range(1, 1683):
    numWatched[x] = 0

##for each movieID, INCF it's value every time
##it has been reviewd in the larger fileLsit of movies it has watched
for x in range(0, len(fileList)):
    ind = fileList[x][1]
    newVal = numWatched[ind] + 1
    numWatched[ind] = newVal

#find largest
maxVal = 0
maxKey = 0
for key in numWatched:
    if(numWatched[key] > maxVal):
        maxVal = numWatched[key]
        maxKey = key

#find smallest
minVal = 100000
minKey = 0
for key in numWatched:
    if(numWatched[key] < minVal):
        minVal = numWatched[key]
        minKey = key



print("\n")
print("The movieID of the movie with the most number of reviews is:")
print("\n")
print(maxKey)
print("\n")
print("and has this many reviews:")
print("\n")
print(maxVal)
print("\n")

print("\n")
print("The moveID of the movie with the least number of reviews is:")
print("\n")
print(minKey)
print("\n")
print("and has this many reviews:")
print("\n")
print(minVal)
print("\n")

#now order a list of elements by review and ascending order

def findMin(d):
    mVal = 100000
    mKey = 0
    for key in numWatched:
        if(numWatched[key] < mVal):
            mVal = numWatched[key]
            mKey = key
```

```
        return [mKey, mVal]

    listOut = []
    lst = numWatched
    for x in range(0, 1682):
        r = findMin(numWatched)
        d = [r[0], r[1]]
        listOut.append(d)
        del lst[r[0]]

    #now have an ordered list of tuples, need to sort into vecotrs

    orderedIDs = []
    for row in listOut:
        orderedIDs.append(row[0])

    orderedNums = []
    for row in listOut:
        orderedNums.append(row[1])


makePairs(u_data_Path)
```

With plotting data:

Movie numbers:

[599, 677, 711, 814, 830, 852, 857, 1122, 1130, 1156, 1201, 1235, 1236, 1309, 1310, 1320, 1325, 1329, 1339, 1340, 1341, 1343, 1348, 1349, 1352, 1363, 1364, 1366, 1373, 1414, 1447, 1452, 1453, 1457, 1458, 1460, 1461, 1476, 1482, 1486, 1492, 1493, 1494, 1498, 1505, 1507, 1510, 1515, 1520, 1525, 1526, 1533, 1536, 1543, 1546, 1548, 1557, 1559, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1574, 1575, 1576, 1577, 1579, 1580, 1581, 1582, 1583, 1584, 1586, 1587, 1593, 1595, 1596, 1599, 1601, 1603, 1604, 1606, 1613, 1614, 1616, 1618, 1619, 1621, 1624, 1625, 1626, 1627, 1630, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1640, 1641, 1645, 1647, 1648, 1649, 1650, 1651, 1653, 1654, 1655, 1657, 1659, 1660, 1661, 1663, 1665, 1666, 1667, 1668, 1669, 1670, 1671, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 600, 784, 897, 907, 913, 957, 1080, 1290, 1304, 1306, 1307, 1308, 1321, 1332, 1334, 1342, 1345, 1350, 1354, 1358, 1359, 1360, 1362, 1365, 1371, 1374, 1390, 1398, 1417, 1433, 1436, 1450, 1455, 1467, 1472, 1477, 1481, 1491, 1497, 1500, 1502, 1519, 1523, 1532, 1541, 1542, 1547, 1549, 1550, 1551, 1554, 1556, 1573, 1578, 1585, 1588, 1590, 1594, 1611, 1617, 1629, 1631, 1642, 1644, 1646, 1656, 1662, 1672, 788, 817, 858, 861, 1027, 1096, 1123, 1144, 1146, 1155, 1186, 1189, 1191, 1196, 1256, 1292, 1293, 1318, 1323, 1327, 1351, 1356, 1357, 1361, 1370, 1372, 1384, 1387, 1389, 1391, 1408, 1418, 1420, 1424, 1430, 1432, 1448, 1463, 1464, 1465, 1484, 1490, 1504, 1506, 1508, 1513, 1516, 1528, 1538, 1544, 1552, 1602, 1607, 1609, 1610, 1622, 1623, 1639, 1652, 1658, 119, 442, 598, 626, 643, 757, 777, 822, 838, 839, 850, 851, 910, 911, 973, 987, 992, 1026, 1064, 1104, 1106, 1125, 1158, 1223, 1257, 1259, 1260, 1261, 1275, 1317, 1319, 1330, 1331, 1338, 1347, 1369, 1377, 1382, 1402, 1403, 1410, 1422, 1423, 1429, 1443, 1445, 1485, 1488, 1499, 1521, 1535, 1537, 1539, 1545, 1553, 1560, 1589, 1600, 1605, 1608, 1612, 1628, 1643, 1664, 75, 104, 247, 314, 437, 439, 594, 666, 799, 868, 920, 1108, 1138, 1162, 1175, 1242, 1272, 1301, 1302, 1324, 1333, 1344, 1346, 1353, 1367, 1378, 1385, 1392, 1396, 1397, 1406, 1416, 1427, 1431, 1438, 1440, 1442, 1470, 1487, 1489, 1495, 1496, 1501, 1509, 1511, 1517, 1529, 1530, 1555, 1592, 1597, 296, 438, 668, 706, 867, 891, 927, 1083, 1164, 1205, 1212, 1213, 1216, 1233, 1237, 1241, 1247, 1250, 1287, 1322, 1326, 1328, 1355, 1368, 1380, 1383, 1386, 1388, 1395, 1405, 1412, 1454, 1466, 1468, 1474, 1512, 1534, 1591, 1598, 34, 74, 884, 917, 935, 994, 1031, 1072, 1075, 1102, 1111, 1151, 1173, 1177, 1198, 1200, 1243, 1246, 1253, 1264, 1270, 1271, 1288, 1289, 1294, 1295, 1305, 1312, 1316, 1376, 1379, 1394, 1399, 1409, 1435, 1441, 1456, 1459, 1462, 1514, 1522, 1527, 1531, 1558, 37, 804, 918, 970, 981, 1002, 1003, 1043, 1077, 1081, 1082, 1100, 1174, 1202, 1227, 1234, 1238, 1251, 1255, 1262, 1276, 1299, 1381, 1413, 1426, 1439, 1449, 1475, 1479, 1524, 113, 267, 446, 592, 766, 776, 798, 803, 807, 848, 870, 899, 912, 964, 1004, 1141, 1145, 1148, 1150, 1179, 1214, 1252, 1279, 1282, 1298, 1300, 1337, 1393, 1415, 1437, 1471, 1473, 1620, 18, 146, 360, 361, 390, 536, 555, 698, 701, 782, 791, 793, 999, 1000, 1055, 1056, 1087, 1114, 1169, 1180, 1181, 1184, 1193, 1268, 1269, 1336, 1375, 1400, 1425, 1434, 1503, 1540, 1615, 35, 341, 374, 621, 759, 767, 914, 954, 1085, 1127, 1159, 1167, 1190, 1195, 1254, 1274, 1314, 1335, 1421, 1480, 394, 397, 545, 548, 556, 593, 667, 901, 967, 976, 1015, 1062, 1068, 1076, 1092, 1094, 1124, 1154, 1166, 1211, 1224, 1249, 1266, 1297, 1428, 1469, 1483, 1518, 36, 377, 534, 669, 695, 714, 787, 828, 883, 885, 889, 909, 915, 1088, 1099, 1116, 1128, 1131, 1161, 1182, 1187, 1192, 1229, 1281, 1404, 353, 440, 587, 726, 786, 853, 893, 958, 996, 1029, 1165, 1171, 1225, 1313, 103, 115, 533, 574, 700, 733, 835, 859, 888, 983, 1024, 1058, 1112, 1143, 1172, 1209, 1263, 1273, 1285, 1291, 1446, 1451, 256, 453, 454, 691, 718, 725, 745, 801, 854, 860, 997, 998, 1032, 1066, 1071, 1278, 1280, 1311, 1419, 557, 773, 947, 1001, 1038, 1120, 1140, 1207, 1232, 1286, 84, 337, 359, 400, 590, 617, 618, 811, 812, 862, 903, 916, 968, 1069, 1097, 1105, 1107, 1185, 1204, 1230, 1239, 1303, 1401, 1478, 138, 263, 424, 579, 703, 894, 1103, 1117, 1153, 1178, 1183, 1206, 1245, 1248, 1265, 1277, 1283, 1444, 112, 320, 351, 601, 904, 908, 982, 1030, 1208, 1219, 1296, 1407, 335, 543, 758, 795, 816, 865, 906, 933, 965, 1084, 1086, 1121, 1147, 1203, 445, 532, 560, 565, 624, 821, 832, 953, 980, 985, 1005, 1057, 1078, 1113, 1168, 1176, 1231, 130, 149, 375,

635, 962, 986, 1006, 1054, 1093, 1199, 1258, 1315, 376, 459, 551, 634, 704, 730, 753, 863, 869, 1036, 1037, 1053, 415, 818, 834,
837, 938, 960, 991, 1040, 1045, 1049, 1052, 1157, 1163, 1222, 6, 253, 352, 398, 422, 775, 800, 836, 966, 1149, 1160, 1218, 1240,
1284, 46, 348, 389, 457, 464, 564, 613, 645, 681, 734, 805, 842, 905, 978, 1034, 1070, 1170, 279, 309, 362, 388, 460, 571, 972,
1042, 1109, 1129, 1152, 1411, 563, 572, 719, 764, 1061, 1095, 1115, 1136, 1137, 308, 537, 608, 843, 950, 1133, 1215, 1220, 110,
349, 368, 383, 500, 605, 630, 738, 779, 797, 808, 995, 1019, 1023, 1194, 426, 580, 765, 774, 936, 940, 974, 989, 1018, 1022,
1033, 1126, 1132, 1226, 78, 573, 641, 878, 990, 1139, 1267, 372, 586, 612, 653, 680, 723, 829, 882, 922, 925, 961, 971, 266, 522,
567, 644, 979, 1020, 1059, 670, 796, 856, 1244, 30, 41, 364, 583, 783, 937, 1008, 1090, 637, 646, 1013, 1021, 1089, 1134, 1210,
16, 370, 373, 503, 585, 622, 623, 731, 743, 752, 785, 874, 951, 1060, 1217, 1221, 43, 57, 611, 639, 697, 802, 819, 929, 932, 943,
945, 1044, 330, 350, 355, 547, 577, 610, 771, 963, 1051, 1063, 1110, 107, 769, 844, 872, 900, 902, 1098, 261, 336, 379, 407, 540,
589, 620, 809, 890, 944, 1050, 1091, 224, 539, 575, 638, 656, 688, 694, 761, 846, 896, 975, 984, 1010, 1025, 1067, 1142, 1188,
236, 329, 552, 728, 810, 881, 942, 952, 1079, 1197, 251, 444, 524, 664, 671, 760, 794, 921, 941, 956, 1046, 1135, 339, 363, 366,
789, 876, 1007, 1118, 244, 365, 456, 467, 562, 674, 841, 948, 1228, 406, 541, 683, 772, 824, 833, 955, 977, 139, 206, 297, 490,
570, 602, 649, 686, 806, 1017, 542, 614, 712, 749, 342, 466, 489, 768, 877, 441, 702, 840, 849, 875, 886, 892, 1065, 76, 102, 396,
535, 675, 721, 344, 369, 413, 847, 395, 494, 813, 27, 40, 142, 502, 512, 754, 770, 827, 931, 85, 166, 632, 716, 722, 741, 898, 923,
61, 391, 492, 495, 561, 581, 615, 737, 880, 1119, 158, 278, 493, 609, 939, 140, 946, 254, 414, 499, 1041, 450, 481, 516, 553, 727,
740, 60, 165, 334, 416, 468, 486, 595, 616, 619, 887, 959, 1009, 108, 128, 145, 171, 345, 387, 488, 672, 189, 207, 220, 262, 452,
538, 606, 607, 790, 855, 993, 1073, 114, 120, 167, 371, 434, 469, 491, 569, 648, 80, 392, 487, 497, 505, 934, 1035, 19, 160, 384,
633, 687, 780, 409, 558, 647, 707, 106, 277, 463, 544, 949, 20, 141, 354, 513, 650, 713, 26, 101, 299, 417, 525, 1048, 105, 221,
461, 1101, 311, 627, 871, 969, 401, 724, 778, 295, 629, 676, 1074, 287, 736, 44, 584, 658, 696, 710, 45, 312, 519, 530, 826, 930,
32, 49, 51, 420, 604, 729, 873, 63, 152, 625, 640, 662, 717, 823, 59, 825, 21, 781, 280, 425, 448, 465, 924, 5, 673, 720, 792, 864,
988, 39, 305, 386, 689, 715, 661, 10, 201, 399, 518, 642, 3, 458, 506, 652, 1039, 52, 338, 517, 636, 693, 831, 17, 163, 212, 549,
578, 744, 412, 430, 529, 576, 820, 1011, 90, 131, 477, 290, 306, 755, 919, 33, 126, 356, 429, 155, 507, 1014, 436, 381, 665, 682,
1012, 159, 232, 264, 378, 404, 708, 926, 317, 554, 699, 747, 67, 54, 92, 478, 709, 928, 136, 122, 162, 421, 895, 679, 470, 225, 81,
219, 382, 93, 316, 408, 815, 331, 29, 214, 292, 65, 123, 659, 762, 184, 192, 380, 663, 48, 242, 449, 169, 631, 746, 866, 38, 217,
521, 170, 428, 447, 509, 510, 528, 504, 501, 190, 233, 246, 343, 520, 526, 750, 116, 178, 324, 485, 346, 473, 62, 157, 198, 260,
291, 596, 53, 73, 148, 241, 325, 482, 756, 72, 129, 418, 531, 109, 249, 2, 657, 243, 47, 68, 213, 303, 1047, 205, 223, 270, 879, 94,
177, 347, 559, 705, 735, 1016, 87, 484, 231, 91, 332, 358, 310, 455, 281, 293, 654, 42, 156, 462, 1028, 55, 304, 763, 86, 239, 77,
164, 550, 498, 431, 660, 31, 690, 240, 150, 193, 685, 252, 472, 248, 315, 476, 227, 66, 259, 285, 410, 443, 411, 523, 692, 739,
199, 226, 319, 582, 321, 628, 684, 188, 367, 402, 451, 133, 137, 218, 229, 433, 651, 99, 255, 511, 24, 154, 432, 58, 326, 327, 83,
845, 283, 419, 591, 479, 480, 566, 514, 732, 751, 23, 203, 14, 13, 147, 124, 307, 340, 274, 209, 393, 284, 298, 474, 527, 250, 134,
272, 230, 208, 403, 515, 588, 200, 211, 597, 175, 385, 4, 187, 603, 271, 215, 88, 508, 435, 235, 322, 8, 95, 427, 678, 71, 161, 179,
180, 471, 143, 273, 182, 265, 655, 301, 568, 496, 282, 11, 185, 197, 245, 323, 194, 144, 483, 125, 228, 132, 153, 475, 70, 186,
196, 333, 546, 268, 97, 238, 135, 289, 82, 357, 12, 742, 275, 111, 89, 28, 191, 202, 234, 64, 176, 216, 183, 15, 25, 118, 96, 328,
22, 302, 276, 318, 9, 423, 195, 257, 269, 168, 748, 69, 173, 151, 210, 79, 405, 204, 313, 222, 172, 117, 237, 98, 7, 56, 127, 174,
121, 300, 1, 288, 286, 294, 181, 100, 258, 50]

Number of times reviewed:

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12,
12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13,
13, 13, 13, 13, 13, 13, 13, 13, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
15, 15, 15, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 17, 17, 17, 17, 17, 17, 17,
17, 17, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 19, 19, 19, 19, 19, 19, 19, 19,
19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21,
21, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 24, 24, 24, 24, 24,
24, 24, 24, 24, 24, 24, 24, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 26, 26, 26, 26, 26, 26, 26, 26, 26, 26, 26, 26, 26, 26,
27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 29, 29, 29, 29, 29, 29,
29, 29, 29, 30, 30, 30, 30, 30, 30, 30, 30, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 32, 32, 32, 32, 32, 32, 32, 32, 32,
32, 32, 32, 32, 32, 33, 33, 33, 33, 33, 33, 33, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 35, 35, 35, 35, 35, 35, 35, 36, 36, 36, 36,
37, 37, 37, 37, 37, 37, 37, 37, 38, 38, 38, 38, 38, 38, 38, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 39, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 41, 41, 41, 41, 41, 41, 41, 41, 41, 41, 41, 41, 41, 42, 42, 42, 42, 42, 42, 42, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 46, 46, 46, 46, 46,
46, 46, 46, 46, 46, 46, 46, 47, 47, 47, 47, 47, 47, 47, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 49, 49, 49, 49, 49, 49, 49, 49, 50, 50, 50, 50,
50, 50, 50, 50, 50, 50, 51, 51, 51, 51, 52, 52, 52, 52, 52, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 54, 54, 54, 54, 54, 54, 55, 55, 55, 55, 56, 56,
56, 57, 57, 57, 57, 57, 57, 57, 57, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 60, 60, 60, 60, 60, 61, 61,

62, 62, 62, 62, 63, 63, 63, 63, 63, 63, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 65, 65, 65, 65, 65, 65, 65, 65, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 67, 67, 67, 67, 67, 67, 67, 67, 67, 68, 68, 68, 68, 68, 68, 68, 68, 69, 69, 69, 69, 69, 69, 70, 70, 70, 70, 71, 71, 71, 71, 71, 72, 72, 72, 72, 72, 72, 73, 73, 73, 73, 73, 73, 74, 74, 74, 74, 75, 75, 75, 75, 76, 76, 76, 77, 77, 77, 77, 78, 78, 79, 79, 79, 79, 80, 80, 80, 80, 80, 80, 81, 81, 81, 81, 81, 81, 81, 82, 82, 82, 82, 82, 82, 82, 83, 83, 84, 84, 85, 85, 85, 85, 85, 86, 86, 86, 86, 86, 86, 87, 87, 87, 87, 87, 88, 89, 89, 89, 89, 89, 90, 90, 90, 90, 90, 91, 91, 91, 91, 91, 91, 92, 92, 92, 92, 92, 92, 93, 93, 93, 93, 93, 93, 95, 95, 95, 96, 96, 96, 96, 97, 97, 97, 97, 98, 98, 98, 99, 100, 100, 100, 100, 101, 101, 101, 101, 101, 101, 101, 102, 102, 102, 102, 103, 104, 104, 104, 104, 104, 105, 106, 106, 106, 106, 107, 108, 109, 110, 111, 111, 112, 112, 112, 112, 113, 114, 114, 114, 115, 115, 115, 115, 116, 116, 116, 116, 117, 117, 117, 118, 119, 119, 119, 120, 120, 120, 121, 121, 121, 121, 121, 121, 122, 123, 124, 124, 124, 124, 124, 124, 124, 125, 125, 125, 125, 126, 126, 127, 127, 127, 127, 127, 127, 128, 128, 128, 128, 128, 128, 128, 129, 129, 129, 129, 130, 130, 131, 131, 132, 133, 134, 134, 134, 134, 136, 136, 136, 136, 137, 137, 137, 137, 137, 137, 137, 138, 138, 142, 143, 143, 143, 145, 145, 146, 147, 147, 148, 148, 148, 148, 149, 149, 149, 150, 150, 151, 151, 151, 152, 153, 153, 154, 155, 156, 157, 157, 157, 158, 158, 160, 160, 160, 161, 162, 162, 162, 162, 162, 163, 164, 164, 164, 165, 166, 168, 168, 169, 169, 169, 170, 170, 170, 170, 171, 171, 171, 171, 171, 171, 172, 172, 173, 174, 174, 174, 175, 175, 175, 176, 176, 177, 178, 178, 179, 179, 179, 180, 180, 180, 182, 182, 183, 184, 185, 187, 188, 189, 190, 191, 192, 193, 194, 194, 195, 197, 198, 198, 199, 200, 201, 201, 202, 206, 206, 206, 208, 208, 209, 209, 209, 211, 212, 213, 215, 216, 217, 218, 219, 219, 219, 219, 220, 220, 221, 221, 221, 222, 223, 226, 227, 227, 230, 230, 231, 232, 236, 239, 239, 240, 240, 241, 243, 243, 244, 244, 246, 247, 250, 251, 251, 251, 251, 254, 255, 256, 256, 259, 259, 261, 264, 267, 267, 268, 272, 275, 276, 276, 280, 280, 283, 284, 290, 291, 293, 293, 293, 295, 295, 297, 297, 298, 298, 299, 300, 301, 303, 315, 316, 316, 321, 324, 326, 331, 336, 344, 350, 350, 365, 367, 378, 384, 390, 392, 394, 413, 420, 429, 431, 452, 478, 481, 485, 507, 508, 509, 583]

Problem 4:

Python script assembled for testing, slight changed in submitted user_cf and item_cf to accommodate being inputted with a datafile of 99,900 instead of a filepath.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 25 16:57:06 2017

@author: Jeremy Midvidy
"""

import sys
import csv
import numpy as np
import scipy as sp
import scipy.stats
import matplotlib
import copy
import random

def user_based_cf_EXPERIMENT(dataSet, userid, movieid, distance, k, iFlag, numOfUsers,
numOfItems):
    def refineDataSet(u, dSet, ID):
        keysToDelete = []
        for key in u:
            contains = False
            arr = u[key]
            for row in arr:
                if(row[0] == ID):
                    contains = True
                    break
            if(contains == False):
                keysToDelete.append(key)

        for row in keysToDelete:
            del u[row]

        cleanDistSet = []
        cleanKeySet = []
```

```
    for x in range(0, len(dSet)):
        if (x+1 in keysToDelete):
            continue
        else:
            cleanDistSet.append(dSet[x])
            cleanKeySet.append(x+1)

    return cleanDistSet, cleanKeySet

def dist(u, distance):
    data = []

    mainUser = users[userid]
    mainInds = []
    mainDict = dict()


    #initialize all the above mains
    for row in mainUser:
        mainDict[row[0]] = row[1]
    for row in mainUser:
        mainInds.append(row[0])
    ################################

    for curr in u:
        current = u[curr]
        mainList = []
        currentDict = dict()
        currentInds = []
        currentList = []

        #initalize all the above currents
        for row in current:
            currentInds.append(row[0])
        for row in current:
            currentDict[row[0]] = row[1]
        ################################

        #constructing review vectors
        aInds = list(set(mainInds).union(set(currentInds)))
        for row in aInds:
            if row in currentDict:
                currentList.append(currentDict[row])
            else:
                currentList.append(0)
            if row in mainDict:
                mainList.append(mainDict[row])
            else:
                mainList.append(0)
        #manhattan distance or inputted distance == 1, else use pearson
        if(distance == 1):
            d = sp.spatial.distance.cityblock(mainList, currentList)
        else:
            ##do you sum? not sure
            d = sum(sp.stats.pearsonr(mainList, currentList))
        data.append(d)
    return data

#now build a dictionary according to aggregate data about USERS
#each key represents a user ID
users = dict()
for x in range(1, numOfUsers+1):
    users[x] = []

##now each time a user has seen a film, add that moveID and rating
```

```
##to it's dictionary
for row in dataSet:
    r = row[0]
    d = row[1]
    m = row[2]
    oldVal = users[r]
    newVal = oldVal + [[d, m]]
    users[r] = newVal

#compute according distance with the dataSet
distanceSet = dist(users, distance)

#refine data set as indicated by the assigment
if(iFlag == 0):
    distanceSet, keySet = refineDataSet(users, distanceSet, movieid)

#edge case detection
if(iFlag == 0 and len(distanceSet) == 1):
    return distanceSet[0][1], 0

#now aggregate k-cluster of best k number of users
#find trueRating is moveid is watched by userid
#find trueRating is moveid is watched by userid
trueRating = 0
if(userid in users):
    for row in users[userid]:
            if(row[0] == movieid):
                trueRating = row[1]
                if(iFlag == 1):
                    del distanceSet[userid-1]
                    del users[userid]
                else: #iFlag = 0
                    trueIndex = keySet.index(userid)
                    del distanceSet[trueIndex]
                    del users[keySet[trueIndex]]
                break

finals = []
for x in range(0, k):
    if (distance == 1):
        ind = np.argmin(distanceSet)
        if(iFlag == 1):
            finals.append(ind+1)
            del distanceSet[ind]
        else:
            finals.append(keySet[ind])
            del distanceSet[ind]
            del keySet[ind]
    else:
        ind = np.argmax(distanceSet)
        if(iFlag == 1):
            finals.append(ind+1)
            del distanceSet[ind]
        else:
            finals.append(keySet[ind])
            del distanceSet[ind]
            del keySet[ind]

finalUsers = []
for row in finals:
    finalUsers.append(users[row])
#create array of ratings from the finalUsers

ratings = []
for row in finalUsers:
    for line in row:
```

```
            ratings.append(line[1])

    if(len(ratings) == 0):
        return 0
    predictedRating = sp.stats.mode(ratings)[0]

    return predictedRating[0]


#########################################################################################################
###
#########################################################################################################
###
#########################################################################################################
###
#########################################################################################################
###
#########################################################################################################
###
#########################################################################################################
###


def item_based_cf_EXPERIMENT(dataSet, userid, movieid, distance, k, iFlag, numOfUsers,
numOfItems):
    def refineDataSet(u, dSet, ID):
        keysToDelete = []
        for key in u:
            contains = False
            arr = u[key]
            for row in arr:
                if(row[0] == ID):
                    contains = True
                    break
            if(contains == False):
                keysToDelete.append(key)

        for row in keysToDelete:
            del u[row]

        cleanDistSet = []
        cleanKeySet = []

        for x in range(0, len(dSet)):
            if (x+1 in keysToDelete):
                continue
            else:
                cleanDistSet.append(dSet[x])
                cleanKeySet.append(x+1)

        return cleanDistSet, cleanKeySet


    def dist(u, distanceMeasure):
        data = []
        mainMovie = movies[movieid]
        mainUsers = []
        mainDict = dict()

        #initalize the above
        for row in mainMovie:
            mainDict[row[0]] = row[1]
        for row in mainMovie:
            mainUsers.append(row[0])
```

```
    for curr in u:
        current = u[curr]
        currentDict = dict()
        currentUsers = []

        mainList = []
        currentList = []

        #initalize all the above currents
        for row in current:
            currentUsers.append(row[0])
        for row in current:
            currentDict[row[0]] = row[1]

        ################################
        #constructing review vectors
        aInds = list(set(mainUsers).union(set(currentUsers)))
        for row in aInds:
            if row in currentDict:
                currentList.append(currentDict[row])
            else:
                currentList.append(0)
            if row in mainDict:
                mainList.append(mainDict[row])
            else:
                mainList.append(0)


        #manhattan distance or inputted distance == 1, else use pearson
        if(distance == 1):
            d = sp.spatial.distance.cityblock(mainList, currentList)
        else:
            ##do you sum? not sure
            d = sum(sp.stats.pearsonr(mainList, currentList))

        data.append(d)

    return data
#make a dictionary of moveIDs
movies = dict()
for x in range(1, numOfItems+1):
    movies[x] = []

#go through the data, every time a movie had been reviewed
#append the [userID review] to the key for each movieID's value
for row in dataSet:
    mID = row[1]
    uID = row[0]
    r = row[2]
    oldVal = movies[mID]
    newVal = oldVal + [[uID, r]]
    movies[mID] = newVal

#compute distance of reviews in the datasets
itemDistanceSet = dist(movies, distance)

#compute distance of reviews in the datasets
if(iFlag == 0):
    itemDistanceSet, keySet  = refineDataSet(movies, itemDistanceSet, userid)

#edge case detection
if(iFlag == 0 and len(distanceSet) == 1):
    return distanceSet[0][1], 0

trueRating = 0
```

```
    if(movieid in movies):
        for row in movies[movieid]:
            if(row[0] == userid):
                trueRating = row[1]
                if(iFlag == 1):
                    del itemDistanceSet[movieid-1]
                    del movies[movieid]
                else: #iFlag = 0
                    trueIndex = keySet.index(movieid)
                    del itemDistanceSet[trueIndex]
                    del movies[keySet[trueIndex]]
                break
    else:
        trueRating = 0

    finals = []
    for x in range(0, k):
        if (distance == 1):
            ind = np.argmin(itemDistanceSet)
            if(iFlag == 1):
                finals.append(ind+1)
                del itemDistanceSet[ind]
            else:
                finals.append(keySet[ind])
                del itemDistanceSet[ind]
                del keySet[ind]
        else:
            ind = np.argmax(itemDistanceSet)
            if(iFlag == 1):
                finals.append(ind+1)
                del itemDistanceSet[ind]
            else:
                finals.append(keySet[ind])
                del itemDistanceSet[ind]
                del keySet[ind]

    finalUsers = []
    for row in finals:
        finalUsers.append(movies[row])
    #create array of ratings from the finalUsers
    ratings = []
    for row in finalUsers:
        for line in row:
            ratings.append(line[1])

    if(len(ratings) == 0):
        return 0

    predictedRating = sp.stats.mode(ratings)[0]
    return predictedRating[0]



##############################################
##############################################
##############################################
#-------- Constructing Data ---------------#
##############################################
##############################################
##############################################

with open('C:\Users\jmidv\Miniconda2\envs\eecs349\u_data.txt','r') as filereader:
    #sort the input file with the csv reader and the \t delimiter
    fileListReader = csv.reader(filereader, delimiter='\t')

    #place holder for dictionary to be created in loop
```

```
    fileList = []
    for row in fileListReader:
        d = row
        for x in range(0, len(d)):
            d[x] = int(d[x])
        fileList.append(d)


##########################################
##########################################
##########################################
#------------ MAKING DRAWS ----------------#
##########################################
##########################################
##########################################

def makeDraws(u):
    draws = []
    drawInds = []

    #construct 50 random samples, with replacement
    for x in range(0, 50):
        randomInds = []
        inds = random.sample(range(0, len(u)), 100)

        d = []
        drawInds.append(inds)

        #for each random index, append the row in fileList to d
        #append the userIDs to dIDS
        for row in inds:
            d.append(fileList[row])


        #then append d to draws,and dIDs to drawIDs
        #do this 50 times to
        draws.append(d)

    return draws, drawInds


data, dataInds = makeDraws(fileList)

#construct list of 99,900 elements no in data, 100 times

##making prior data
priorData = []
for row in dataInds:
    pd = []
    for x in range (0, len(fileList)):
        if(x in row):
            continue
        else:
            pd.append(fileList[x])
    priorData.append(pd)


##########################################
##########################################
##########################################
#----------- RUNNING TESTS ----------------#
##########################################
##########################################
##########################################
```

```
##initialize test variables

distance = 1 #0 for pearsons, 1 for manhattan
k = 2
i = 1
numUsers = 943
numItems = 1682


########################
#run tests

user_cf_RESULTS = []
item_cf_RESULTS = []


for x in range(0, 50):#50
    uCFrs = []
    iCFrs = []
    for y in range(0, len(data[x])):#100
        current = data[x][y]
        if(x < 25):
            pR = user_based_cf_EXPERIMENT(priorData[x], current[0] , current[1], distance, k,
i, numUsers, numItems)
            uCFrs.append(pR)
            print("\n")
            print("user-based CF, row " + str(x+1) + " / 50, col " + str(y) + " predicted:")
            print(str(pR))
            print("\n")
        else:
            pR = item_based_cf_EXPERIMENT(priorData[x], current[0] , current[1], distance, k,
i, numUsers, numItems)
            iCFrs.append(pR)
            print("\n")
            print("item-based CF, row " + str(x+1) + " / 50, col " + str(y) + " predicted:")
            print(str(pR))
            print("\n")
    if(x < 25):
        user_cf_RESULTS.append(uCFrs)
    else:
        item_cf_RESULTS.append(iCFrs)


userActual = []
itemActual = []

for r in range(0, len(data)):
    c = []
    for line in data[r]:
        c.append(line[2])
    if(r < 25):
        userActual.append(c)
    else:
        itemActual.append(c)

user_cf_ERRORS = []
item_cf_ERRORS = []

for x in range(0, len(userActual)):
    e = []
    for y in range(0, len(userActual[x])):
        e.append(abs(userActual[x][y] - user_cf_RESULTS[x][y]))
    user_cf_ERRORS.append(e)

for x in range(0, len(itemActual)):
    e = []
```

```
    for y in range(0, len(itemActual[x])):
        e.append(abs(itemActual[x][y] - item_cf_RESULTS[x][y]))
    item_cf_ERRORS.append(e)

print("\n")
print(user_cf_ERRORS)
print("\n")
print(item_cf_ERRORS)

user_average_errors = []
item_average_errors = []

for row in user_cf_ERRORS:
    avg = np.mean(row)
    user_average_errors.append(avg)

for row in item_cf_ERRORS:
    avg = np.mean(row)
    item_average_errors.append(avg)

print("\n")
print(user_average_errors)
print("\n")
print(item_average_errors)
```

Output for graphs and some sample output of each line as my code ran

---

#first overall test, all 50 rows, standard inputs

test: distance = 1 #0 for pearsons, 1 for manhattan
k = 3
i = 1
numUsers = 943
numItems = 1682

user average error:
0.8492

item average error:
0.8308

user average errors:
[0.81000000000000005, 0.81000000000000005, 0.83999999999999997, 0.85999999999999999, 0.93999999999999995, 0.77000000000000002, 0.89000000000000001, 0.80000000000000004, 0.83999999999999997, 0.92000000000000004, 0.95999999999999996, 0.96999999999999997, 0.78000000000000003, 0.84999999999999998, 0.81999999999999995,

0.7399999999999999, 0.8199999999999995, 0.9200000000000004, 0.7199999999999997, 0.7399999999999999,
0.9899999999999999, 0.7700000000000002, 0.8900000000000001, 1.04, 0.7399999999999999]

item average erros

[0.8399999999999997, 0.9399999999999995, 0.7099999999999996, 0.8599999999999999, 0.87, 0.9899999999999999,
0.8299999999999996, 0.8299999999999996, 0.8399999999999997, 0.8299999999999996, 0.7900000000000004,
0.9799999999999998, 0.9100000000000003, 0.88, 0.6999999999999996, 0.87, 0.87, 0.9599999999999996,
0.6999999999999996, 0.5999999999999998, 0.8100000000000005, 0.7199999999999997, 0.9000000000000002,
0.8100000000000005, 0.7299999999999998]

item errors

[[1, 1, 1, 1, 1, 1, 2, 2, 0, 0, 2, 0, 2, 0, 1, 1, 2, 1, 1, 1, 0, 2, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 2, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 2, 1, 0, 0,
1, 2, 0, 0, 0, 2, 0, 1, 1, 1, 3, 2, 2, 0, 1, 0, 1, 0, 1, 2, 3, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 2, 0, 2, 4, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1], [2, 1, 1, 0,
1, 1, 0, 2, 0, 2, 2, 1, 1, 3, 1, 1, 1, 2, 1, 1, 0, 1, 2, 1, 1, 0, 1, 0, 0, 2, 1, 0, 0, 0, 2, 1, 0, 1, 0, 0, 2, 0, 0, 1, 2, 1, 1, 2, 0, 1, 1, 2, 1, 1, 1, 1,
0, 1, 1, 0, 2, 2, 1, 0, 1, 0, 2, 3, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 3, 1, 1, 1, 0, 1, 0, 3, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1], [1, 1, 2, 1, 0, 0, 0, 0,
1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 2, 0, 1, 0, 2, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 2, 0, 3, 2, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1,
1, 2, 0, 0, 1, 1, 1, 1, 1, 0, 2, 2, 1, 0, 1, 1, 1, 1, 0, 1, 2, 1, 0, 1, 0, 1, 0, 2, 0, 2, 0, 1, 1, 0, 1, 1, 0, 2, 0, 2], [2, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1, 1, 2, 0, 1, 0, 2, 0, 0, 2, 1, 4, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 2, 0, 2, 0, 4, 0, 2, 1, 1, 1, 0, 1, 0, 0, 1, 2, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 2, 1, 0, 1,
0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 2, 1, 1, 3, 1, 0, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 1, 1, 1, 1], [1, 1, 0, 2, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 2, 1, 1, 0, 2, 1, 1, 1, 2, 1, 1, 0, 1, 1, 1, 2, 1, 1, 3, 0, 0, 2, 2, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0,
1, 2, 1, 1, 1, 2, 0, 1, 1, 1, 2, 0, 2, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0], [3, 0, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 0, 1, 3, 1,
2, 1, 3, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 2, 2, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 3, 0, 0, 0, 2, 3, 1, 0, 0, 1, 3, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0,
0, 0, 1, 1, 1, 0, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 3, 0, 2, 1, 1], [0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 2, 2, 0, 3,
2, 1, 1, 0, 1, 2, 1, 1, 1, 1, 0, 0, 3, 1, 1, 2, 0, 0, 1, 0, 1, 3, 1, 2, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 2, 1, 0, 0, 0, 1, 2, 1, 2,
1, 1, 2, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 2, 0, 1, 1, 3, 0, 0, 2, 1, 1, 2], [1, 1, 1, 2, 0, 3, 1, 2, 1, 1, 0, 0, 1, 1, 1, 2, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0,
1, 0, 1, 1, 2, 0, 0, 1, 2, 1, 1, 0, 1, 0, 3, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 3, 2, 0, 0, 1, 0, 2, 1, 0, 1, 0, 2, 1, 0, 1,
0, 0, 0, 2, 1, 1, 0, 2, 0, 0, 2, 0, 1, 0, 0, 1, 1, 1, 2, 1], [1, 1, 1, 2, 1, 2, 0, 1, 0, 0, 0, 0, 2, 1, 2, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0,
0, 1, 1, 2, 0, 2, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 2, 0, 0, 3, 2, 0, 2, 0, 2, 0, 1, 0, 1, 1, 0, 0, 2, 1, 0, 0, 2, 0, 1, 2, 2, 1, 1, 1, 0, 0, 3, 1, 3, 1,
1, 1, 1, 0, 0, 2, 1, 1, 1, 0, 2, 1, 1, 1, 1, 1], [2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 1, 0, 1, 0, 2, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 1,
2, 0, 0, 3, 1, 1, 1, 2, 1, 1, 2, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 4, 1, 0, 1, 0, 2, 0, 0, 1, 1, 3, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 2, 1, 0, 0, 2, 0, 1, 2, 2,
1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 2, 4], [1, 0, 0, 1, 0, 0, 2, 0, 2, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 2, 1, 0, 1, 2, 2, 0, 0, 1, 0, 2, 0, 0, 2, 2, 1, 2, 3,
0, 1, 2, 0, 1, 1, 1, 1, 0, 1, 2, 1, 0, 0, 2, 1, 2, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 3, 2, 0, 0, 1, 0, 0, 1, 1, 1,
1, 2, 0, 1, 1, 0, 1, 1], [0, 2, 2, 1, 0, 1, 0, 0, 0, 1, 2, 1, 0, 1, 0, 1, 1, 4, 3, 2, 0, 1, 1, 1, 2, 1, 1, 0, 0, 2, 1, 3, 1, 2, 1, 1, 1, 1, 3, 0, 2, 1, 0, 3,
1, 1, 1, 0, 1, 1, 0, 1, 2, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 2, 2, 0, 3, 0, 0, 2, 2, 1, 1, 2, 1, 3, 0, 1, 3, 0, 2, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0,
2, 1, 0, 0], [1, 1, 3, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 3, 2, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 3, 0, 0, 1, 1, 0, 0, 1, 1,
0, 1, 1, 3, 0, 0, 0, 1, 4, 2, 0, 2, 0, 2, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 2, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 4, 1, 4, 0, 1, 1, 3, 1, 0, 1, 2, 1, 3, 1],
[2, 0, 2, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 2, 2, 2, 1, 0, 0, 2, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 2,
1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 4, 1, 0, 3, 0, 1, 1, 1, 1, 1, 0, 3, 1, 1, 3, 0, 1, 1, 0, 1, 1, 3, 0, 1, 3, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1], [0, 2, 0, 1,
0, 0, 1, 1, 1, 0, 1, 0, 1, 2, 1, 1, 1, 2, 1, 0, 1, 0, 2, 0, 0, 2, 1, 1, 1, 1, 1, 0, 0, 1, 1, 2, 0, 1, 2, 1, 0, 0, 1, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 1, 2, 0,
1, 0, 1, 1, 2, 1, 1, 0, 1, 0, 2, 0, 2, 0, 1, 1, 1, 0, 0, 1, 1, 0, 2, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 2, 1, 0, 0, 1, 0, 1, 2], [1, 2, 0, 2, 0, 1, 1, 1,
1, 0, 1, 0, 0, 2, 2, 2, 1, 0, 1, 1, 2, 1, 1, 0, 1, 1, 0, 3, 0, 1, 1, 1, 1, 2, 1, 0, 1, 2, 0, 0, 3, 2, 0, 0, 2, 0, 1, 0, 2, 2, 1, 0, 1, 0, 0, 2, 0, 1, 1, 1,
1, 1, 2, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 3, 2, 0, 2, 3, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0], [2, 1, 1, 2, 1, 0, 0, 2, 2, 2, 1, 3,
0, 1, 1, 1, 0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 1, 1, 0, 2, 1, 0, 0, 1, 2, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0,
2, 0, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 0, 1, 0, 1, 1, 2, 0, 1, 0, 2, 2, 1, 2, 1, 0, 1, 2, 1, 1, 0, 0, 2], [0, 2, 1, 1, 1, 0, 2, 1, 0, 0, 0, 0, 1, 1, 4, 1,
0, 2, 0, 3, 1, 1, 2, 1, 1, 0, 1, 3, 0, 0, 2, 0, 1, 0, 2, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 3, 1, 1, 1, 1, 0, 1, 1, 2, 1, 2, 2, 0, 1, 4, 0, 1, 1, 0, 1, 2, 0,
1, 0, 2, 2, 0, 1, 2, 0, 1, 2, 2, 2, 0, 0, 1, 1, 2, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 2, 1, 0], [0, 1, 0, 0, 2, 0, 1, 1, 1, 1, 0, 1, 1, 2, 0, 0, 1, 0, 1, 3,
0, 1, 0, 2, 1, 2, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 2, 0, 1, 2, 1, 0, 0, 1, 1, 0, 2, 2, 1, 0, 1, 1, 0, 1, 0, 1, 3, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
1, 3, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 3, 1, 1, 1, 0, 0, 1], [1, 1, 0, 0, 1, 1, 2, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 2, 0, 1, 0, 1, 1,
1, 0, 0, 1, 0, 2, 0, 0, 0, 1, 1, 2, 2, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 2, 2, 0,
0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 2, 0, 0, 1, 1, 1, 0, 0, 1], [1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 0, 0, 1, 1, 1, 1, 2, 0, 1, 1, 0, 0, 0, 4, 0, 1, 1,
2, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 2, 0, 1, 0, 0, 1, 0, 2, 1, 1, 1, 1, 2, 1, 3, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 2, 1, 0, 1, 1,
1, 2, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 2, 1], [0, 1, 2, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 2, 0, 0, 0, 0, 0, 1, 1, 1, 2, 0, 1, 1,
0, 1, 0, 1, 2, 1, 0, 1, 0, 2, 2, 0, 1, 0, 0, 2, 1, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 3, 2, 0, 1, 2, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0,
1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 2, 1, 1, 1], [1, 1, 1, 1, 0, 3, 1, 2, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 2, 2, 2, 1, 1, 1, 1, 0, 0, 1, 2, 0, 0, 1, 2, 1, 1, 1,
0, 0, 1, 0, 0, 1, 2, 1, 0, 0, 1, 2, 1, 2, 2, 1, 1, 0, 1, 1, 1, 1, 2, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 2, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1,

1, 1, 1, 2, 1, 3, 1, 2, 1, 2, 1, 1], [0, 0, 3, 0, 1, 1, 0, 2, 0, 1, 2, 1, 0, 1, 2, 2, 0, 0, 2, 1, 2, 0, 2, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 3, 1, 0, 1, 1, 0, 0, 2, 1, 1, 1, 1, 0, 2, 2, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 2, 0, 2, 1, 0, 1, 0, 3, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 2, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 2, 0, 4], [0, 0, 0, 3, 1, 1, 0, 1, 1, 1, 0, 1, 0, 2, 1, 0, 0, 1, 1, 0, 0, 0, 0, 2, 1, 1, 1, 3, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 3, 0, 1, 2, 2, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 0, 1, 1, 2, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 2, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1]]

user errors:
[[1, 1, 1, 2, 1, 0, 1, 1, 2, 3, 1, 0, 2, 1, 1, 1, 0, 0, 2, 0, 1, 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 3, 2, 1, 2, 1, 0, 1, 0, 0, 0, 0, 1, 1, 2, 2, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 2, 1, 2, 3, 2, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 2, 0, 1, 0, 0, 0, 0, 1, 1, 1, 2, 1, 0, 1, 0, 0, 1], [0, 0, 1, 1, 0, 1, 1, 1, 3, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 2, 0, 3, 1, 1, 0, 0, 1, 0, 0, 1, 2, 2, 0, 1, 0, 1, 2, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 2, 2, 0, 1, 2, 1, 2, 0, 0, 1, 1, 1, 1, 1, 0, 2, 0, 2, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 2, 1, 1, 1, 2, 0, 0, 1, 0, 1, 1, 1], [0, 1, 0, 0, 1, 1, 2, 0, 0, 0, 1, 1, 0, 0, 0, 3, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 0, 2, 2, 1, 0, 1, 1, 2, 1, 0, 1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 2, 1, 1, 2, 1, 0, 3, 0, 2, 0, 1, 2, 0, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 2, 0, 1, 0, 0], [0, 1, 1, 0, 1, 1, 1, 3, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 2, 1, 1, 2, 0, 1, 2, 1, 0, 0, 0, 0, 1, 0, 0, 2, 1, 1, 0, 2, 1, 1, 0, 0, 0, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 2, 0, 0, 1, 1, 1, 0, 2, 2, 1, 2, 0, 0, 2, 1, 3, 0, 2, 3, 3, 0, 1, 0, 1, 0, 0, 0, 3, 1, 0], [1, 2, 4, 1, 1, 0, 1, 0, 0, 2, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 2, 1, 0, 0, 1, 1, 1, 2, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 4, 1, 2, 1, 1, 1, 2, 3, 1, 1, 1, 1, 0, 3, 1, 1, 0, 2, 2, 0, 2, 2, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 2, 3, 2, 1, 3, 0, 1], [0, 2, 1, 0, 0, 1, 0, 0, 0, 0, 1, 2, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 3, 3, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 3, 0, 1, 0, 2, 2, 1, 1, 1, 1, 1, 2, 3, 1, 2, 0, 0, 1, 0, 1, 1, 1, 1, 2, 0, 0, 0, 1, 1, 2, 1, 1, 1, 0, 0, 0, 1, 2, 1, 0, 0, 1, 1, 1, 2, 0, 1, 1, 1, 0, 0, 1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 0, 1, 0, 2, 0, 1, 0, 0, 1, 0, 0], [0, 0, 1, 1, 3, 0, 0, 1, 1, 2, 0, 0, 2, 1, 0, 1, 1, 0, 0, 0, 4, 0, 1, 1, 2, 3, 0, 0, 1, 1, 2, 2, 1, 1, 1, 1, 0, 0, 1, 2, 1, 1, 2, 1, 1, 1, 4, 2, 1, 0, 3, 1, 1, 0, 2, 1, 0, 2, 2, 1, 2, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 2, 3, 1, 1, 2, 1, 1, 0], [0, 0, 0, 0, 0, 2, 2, 1, 2, 0, 1, 1, 1, 1, 1, 2, 1, 0, 0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 1, 1, 0, 0, 2, 2, 1, 1, 1, 0, 0, 3, 0, 0, 3, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 0, 1, 2, 1, 0, 0, 2, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 3, 0, 0, 1, 1, 1, 3, 0, 0], [1, 1, 0, 1, 1, 4, 0, 2, 3, 0, 1, 0, 0, 1, 1, 3, 0, 1, 2, 3, 1, 2, 2, 1, 1, 1, 3, 1, 1, 2, 1, 2, 2, 1, 0, 1, 2, 1, 1, 0, 2, 1, 3, 0, 1, 0, 0, 0, 0, 2, 0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 2, 0, 1, 0, 2, 1, 1, 2, 2, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1], [2, 1, 1, 2, 1, 0, 2, 2, 1, 1, 1, 0, 2, 1, 2, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 4, 1, 0, 1, 1, 0, 2, 1, 1, 0, 0, 1, 1, 0, 0, 2, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 2, 3, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 2, 0, 0, 1, 2, 0, 0, 0, 1, 1, 2, 1, 1, 2, 1, 2, 0, 3], [1, 0, 4, 1, 1, 1, 1, 2, 1, 0, 1, 1, 0, 0, 1, 1, 1, 4, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 2, 0, 0, 1, 0, 2, 1, 1, 0, 0, 1, 1, 2, 0, 1, 1, 2, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 2, 0, 1, 1, 0, 0, 1, 1, 2, 1, 1, 0, 0, 2, 2, 0, 0, 0, 2, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 2, 0, 0, 2, 2, 0, 0, 1, 3, 1], [0, 1, 2, 0, 3, 0, 1, 3, 0, 0, 1, 1, 0, 2, 1, 1, 1, 0, 2, 0, 0, 0, 1, 1, 0, 1, 1, 3, 1, 0, 0, 1, 1, 2, 1, 0, 1, 0, 0, 0, 0, 1, 1, 2, 1, 0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 3, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 4, 0, 2, 0, 0, 1, 1, 0, 1, 1, 1, 1, 2, 0, 1, 0, 0, 1, 1, 0, 1, 1, 2, 2, 0, 2, 0, 1, 1], [2, 1, 0, 1, 1, 2, 1, 0, 0, 0, 1, 1, 0, 1, 2, 2, 0, 1, 0, 0, 2, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 2, 0, 2, 0, 1, 0, 0, 1, 0, 0, 1, 0, 4, 1, 1, 1, 1, 0, 0, 0, 2, 1, 1, 0, 0, 1, 0, 1, 1, 2, 0, 0, 0, 1, 0, 1, 1, 3, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 0, 1, 2, 2, 1, 0, 2, 1, 1, 0, 0, 2, 1, 1, 0, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 3, 0, 1, 1, 1, 0, 1, 1, 2, 0, 0, 1, 1, 1, 3, 0, 0, 0, 1, 3, 1, 1, 1, 0, 0, 1, 1, 0, 0, 3, 1, 0, 1, 0, 1, 2, 1, 2, 0, 0, 1, 3, 2, 1, 0, 2, 0, 1, 1, 3, 1, 1, 1, 1, 1, 1, 0], [1, 0, 0, 0, 1, 1, 0, 0, 2, 1, 2, 2, 1, 0, 0, 2, 1, 2, 0, 1, 0, 1, 1, 1, 2, 0, 0, 0, 1, 1, 0, 1, 3, 2, 1, 1, 1, 0, 2, 0, 0, 0, 3, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 2, 1, 1, 1, 2, 0, 1, 0, 1, 1, 0, 0, 2, 1, 1, 0, 1, 2, 2, 1, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 2, 2, 1, 1, 3, 0, 0, 2, 1, 0, 0, 1], [1, 0, 1, 1, 3, 1, 1, 4, 0, 0, 2, 0, 3, 2, 0, 0, 0, 2, 1, 2, 3, 0, 1, 0, 2, 2, 0, 0, 1, 1, 1, 0, 2, 1, 1, 0, 1, 1, 0, 1, 0, 2, 1, 1, 0, 2, 0, 0, 2, 1, 1, 1, 1, 1, 4, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 2, 0, 0, 1, 2, 0, 1, 1, 1, 1, 1, 0, 1, 4, 2, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 4, 2, 1], [0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 2, 0, 1, 1, 1, 1, 2, 1, 1, 1, 0, 0, 0, 2, 3, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2,

2, 0, 1, 1, 0, 0, 0, 2, 1, 0, 2, 1, 2, 1, 0, 0, 0, 1, 0, 1, 1, 2, 1, 0, 0, 0, 0, 0, 2, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 2, 0, 1, 0, 4, 0, 1, 1, 1, 0, 1, 0, 1, 2, 1, 0, 1]]

example output

When running item-based CF, on row 49 out of 50, column 18 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 19 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 20 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 21 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 22 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 23 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 24 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 25 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 26 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 27 returned a prediction of:
1

When running item-based CF, on row 49 out of 50, column 28 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 29 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 30 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 31 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 32 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 33 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 34 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 35 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 36 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 37 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 38 returned a prediction of:
2

When running item-based CF, on row 49 out of 50, column 39 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 40 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 41 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 42 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 43 returned a prediction of:
1

When running item-based CF, on row 49 out of 50, column 44 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 45 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 46 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 47 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 48 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 49 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 50 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 51 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 52 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 53 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 54 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 55 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 56 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 57 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 58 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 59 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 60 returned a prediction of:
1

When running item-based CF, on row 49 out of 50, column 61 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 62 returned a prediction of:
2

When running item-based CF, on row 49 out of 50, column 63 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 64 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 65 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 66 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 67 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 68 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 69 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 70 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 71 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 72 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 73 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 74 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 75 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 76 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 77 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 78 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 79 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 80 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 81 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 82 returned a prediction of:
2

When running item-based CF, on row 49 out of 50, column 83 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 84 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 85 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 86 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 87 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 88 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 89 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 90 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 91 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 92 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 93 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 94 returned a prediction of:
5

When running item-based CF, on row 49 out of 50, column 95 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 96 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 97 returned a prediction of:
3

When running item-based CF, on row 49 out of 50, column 98 returned a prediction of:
4

When running item-based CF, on row 49 out of 50, column 99 returned a prediction of:
4

#change distance to pearson for item based (manhattan used in first trial)

distance = 0 #0 for pearsons, 1 for manhattan
k = 3
i = 1
numUsers = 943
numItems = 1682

item average:
0.85039999999999993

mean error per draw
[0.82999999999999996, 0.89000000000000001, 0.94999999999999996, 0.93000000000000005, 0.80000000000000004, 0.81000000000000005, 0.89000000000000001, 0.82999999999999996, 0.80000000000000004, 0.95999999999999996, 1.0800000000000001, 0.80000000000000004, 0.87, 0.81000000000000005, 0.82999999999999996, 0.89000000000000001, 0.88, 0.82999999999999996, 0.72999999999999998, 0.85999999999999999, 0.82999999999999996, 0.70999999999999996, 0.68999999999999995, 0.90000000000000002, 0.85999999999999999]

error distance in each sample:
[[0, 1, 0, 0, 1, 0, 1, 0, 2, 0, 1, 1, 1, 1, 2, 0, 0, 0, 2, 0, 1, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1, 0, 1, 0, 0, 1, 2, 1, 2, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 2, 1, 1, 0, 0, 1, 0, 3, 1, 3, 0, 1, 1, 0, 0, 1, 2, 1, 0, 1, 0, 1, 2, 2, 0, 1, 1, 0, 1, 4, 1, 1, 1, 1, 2, 0, 1, 0, 1, 0, 1, 1, 0, 0, 3], [1, 0, 2, 3, 0, 2, 0, 1, 0, 2, 2, 0, 1, 1, 3, 1, 0, 1, 1, 1, 0, 1, 2, 1, 0, 0, 1, 1, 1, 0, 0, 2, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 2, 2, 1, 0, 0, 1, 2, 1, 0, 1, 0, 1, 1, 1, 1, 1, 2, 0, 0, 1, 0, 2, 1, 1, 1, 0, 1, 1, 1, 0, 1, 2, 0, 1, 0, 2, 1, 1, 0, 1, 0, 2, 1, 2, 0, 0, 0, 2, 0, 2, 2, 2, 1, 1, 1], [2, 1, 1, 2, 1, 1, 0, 0, 0, 0, 1, 0, 3, 1, 1, 3, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1, 3, 0, 0, 2, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 2, 1, 0, 1, 1, 0, 2, 1, 1, 0, 1, 1, 1, 0, 2, 3, 1, 0, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 3, 1, 1, 1, 1, 1, 0, 3, 1, 1, 1, 0, 1, 2, 1, 1, 0, 1, 1, 1, 1, 2, 1, 1], [1, 2, 3, 1, 0, 2, 1, 1, 2, 0, 1, 1, 0, 0, 1, 0, 1, 0, 3, 0, 3, 2, 1, 0, 0, 1, 1, 1, 2, 2, 0, 1, 2, 1, 0, 2, 0, 1, 1, 0, 2, 1, 0, 2, 1, 0, 3, 0, 0, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 2, 1, 1, 0, 1, 0, 0, 0, 2, 1, 0, 2, 0, 1, 2, 1, 2, 0, 0, 1, 1, 2, 0, 0, 2, 2, 0, 1, 1, 1, 1, 2, 1, 2, 1, 1], [0, 1, 1, 1, 1, 1, 1, 3, 1, 2, 1, 0, 2, 2, 0, 0, 2, 1, 1, 1, 0, 0, 1, 2, 1, 0, 0, 0, 1, 0, 1, 2, 0, 0, 1, 2, 0, 1, 1, 0, 0, 1, 1, 3, 1, 1, 0, 1, 2, 1, 2, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 3, 2, 1], [1, 2, 1, 2, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 2, 1, 0, 2, 1, 1, 0, 2, 0, 0, 0, 2, 0, 1, 2, 1, 1, 2, 2, 0, 1, 2, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 3, 0, 1, 0, 2, 0, 1, 0, 0, 3, 0, 3, 0, 2, 1, 1, 1, 1, 1, 2, 1, 0, 0, 1, 1, 0, 0, 0, 2, 1, 0, 1, 1, 0, 0, 1, 0, 2, 0, 1, 1, 1], [1, 0, 1, 0, 1, 0, 1, 2, 1, 0, 0, 2, 0, 0, 1, 4, 0, 0, 1, 2, 2, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 2, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 2, 3, 0, 1, 0, 4, 2, 1, 1, 1, 0, 3, 1, 3, 2, 2, 0, 0, 0, 1, 0, 1, 1, 1, 2, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 2, 1, 2, 0, 1, 1, 0, 4], [2, 0, 1, 1, 0, 0, 0, 1, 0, 0, 2, 1, 0, 1, 1, 1, 2, 0, 2, 2, 1, 1, 0, 2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 2, 0, 0, 2, 0, 0, 0, 1, 1, 3, 2, 0, 1, 1, 1, 0, 1, 2, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 2, 2, 0, 0, 0, 0, 0, 0, 2, 1, 2, 1, 1, 2, 1], [0, 1, 1, 1, 3, 0, 0, 1, 1, 2, 0, 1, 0, 0, 1, 1, 1, 0, 2, 0, 0, 1, 2, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 2, 1, 0, 2, 1, 1, 1, 2, 1, 1, 1, 2, 0, 1, 0, 0, 2, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 0, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 0, 1, 0, 0, 1, 2, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 3, 0, 1, 2], [3, 1, 1, 1, 1, 1, 2, 1, 0, 0, 1, 1, 2, 2, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 2, 1, 0, 1, 1, 1, 2, 0, 1, 0, 3, 1, 2, 0, 1, 2, 2, 0, 0, 1, 2, 1, 1, 1, 1, 0, 0, 2, 0, 2, 1, 0, 0, 1, 2, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 2, 2, 1, 4, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 2, 3, 0, 2], [1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 2, 0, 0, 1, 1, 4, 0, 1, 3, 1, 1, 1, 1, 1, 2, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 3, 0, 1, 3, 0, 1, 2, 3, 0, 0, 1, 1, 0, 2, 2, 1, 1, 2, 1, 1, 3, 0, 3, 0, 4, 2, 0, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 0, 1, 2, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 3, 0], [3, 1, 1, 1, 1, 1, 2, 0, 0, 2, 0, 0, 0, 0, 2, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 2, 1, 0, 1, 1, 2, 1, 1, 1, 1, 1, 0, 1, 2, 0, 2, 1, 1, 2, 0, 1, 1, 1, 0, 1, 2, 0, 2, 1, 3, 1, 0, 0, 0, 3, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1], [1, 1, 1, 2, 2, 1, 3, 0, 0, 2, 1, 2, 1, 2, 1, 1, 2, 1, 0, 0, 0, 0, 1, 2, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 3, 0, 0, 1, 1, 1, 2, 0, 1, 0, 0, 0, 3, 1, 1, 0, 0, 2, 0, 1, 1, 0, 0, 1, 1, 0, 2, 1, 1, 0, 2, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 2, 0, 0, 0, 0, 3, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 2, 0, 1, 1, 0, 2, 2, 1], [0, 1, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 2, 2, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 2, 2, 0, 3, 1, 0, 1, 1, 0, 1, 1, 0, 2, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 2, 0, 1, 0, 1, 1, 2, 1, 2, 0, 0, 2, 0, 2, 1, 1, 1, 0, 1, 0, 3, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 2, 1, 0, 3, 3, 1, 0], [0, 0, 1, 1, 0, 0, 3, 1, 1, 0, 1, 1, 0, 3, 1, 1, 1, 1, 1, 1, 0, 1, 1, 2, 1, 0, 3, 3, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 2, 1, 1, 0, 1, 1, 0, 2, 0, 0, 1, 1, 1, 2, 2, 1, 0, 1, 1, 2, 1, 0, 0, 2, 2, 1, 0, 1, 2, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0], [1, 0, 0, 2, 1, 0, 1, 1, 0, 2, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 2, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 2, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 2, 0, 0, 1, 1, 1, 2, 2, 1, 0, 1, 1, 2, 1, 0, 0, 2, 2, 1, 0, 1, 2, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0], [1, 0, 0, 2, 1, 0, 1, 1,

1, 1, 1, 2, 1, 2, 2, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 2, 0, 1, 1, 1, 2, 2, 0, 0, 1, 2, 3, 0, 0, 1, 0, 1, 0, 2, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 2, 1, 1, 2, 0, 0, 1, 0, 0, 3, 1, 2, 1, 1, 0, 0, 2, 1, 1, 1, 1, 0, 0, 2, 3, 1, 2, 2, 1, 2, 0, 2, 0, 1, 4], [0, 0, 1, 2, 0, 0, 1, 0, 2, 1, 1, 3, 1, 1, 1, 3, 0, 0, 0, 2, 2, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 2, 2, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 2, 3, 1, 2, 1, 0, 2, 0, 0, 1, 1, 0, 0, 0, 0, 3, 2, 1, 3, 1, 0, 1, 1, 2, 1, 1, 0, 1, 0, 1, 1, 1, 0, 2, 1, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 1], [1, 0, 1, 0, 1, 1, 2, 1, 1, 2, 1, 1, 1, 0, 2, 1, 0, 0, 1, 2, 1, 0, 2, 1, 1, 2, 2, 0, 1, 0, 1, 0, 0, 2, 2, 1, 0, 0, 1, 1, 1, 3, 0, 1, 2, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 2, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 3, 0, 0, 2, 1, 1, 0, 1, 1, 0, 0, 2, 0, 0, 1, 0, 1, 0, 2, 0, 0, 2, 2, 0, 0, 0, 1, 1, 0, 1, 0, 1], [1, 1, 0, 0, 1, 2, 0, 0, 2, 0, 0, 0, 2, 1, 2, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 2, 1, 2, 0, 2, 1, 0, 0, 2, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 0, 0, 2, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 2, 1, 1, 0, 0, 0, 2, 0, 2, 1, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 1, 0, 2, 0, 0, 2, 1, 0, 1, 0, 2, 0, 1, 1, 1, 1], [0, 0, 1, 0, 0, 3, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 2, 1, 1, 2, 1, 0, 0, 2, 1, 1, 1, 1, 0, 3, 4, 1, 2, 2, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 2, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 2, 2, 0, 1, 1, 2, 1, 0, 2, 0, 1, 0, 0, 1, 0, 3, 0, 1, 1, 2, 0, 1, 1, 0, 1, 2, 1, 0, 2, 0], [1, 1, 0, 2, 2, 3, 0, 2, 0, 1, 2, 1, 0, 0, 1, 1, 2, 0, 1, 0, 2, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 3, 2, 0, 0, 1, 0, 2, 1, 2, 2, 1, 1, 0, 1, 0, 2, 0, 0, 0, 1, 1, 2, 0, 0, 2, 0, 1, 2, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 3, 0, 1, 0, 1], [1, 1, 0, 0, 2, 0, 1, 1, 0, 0, 1, 1, 2, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 3, 1, 0, 0, 1, 1, 1, 1, 0, 2, 0, 1, 0, 0, 1, 2, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 2, 0, 3, 0, 0, 1, 1, 0, 1, 2, 1, 0, 1, 3, 1, 1, 2, 0, 1, 2, 0, 3, 0, 0, 0, 0, 0, 0, 2, 2, 1, 1, 0, 1], [0, 0, 0, 0, 2, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 2, 0, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 2, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 2, 2, 0, 2, 1, 0, 1, 1, 1, 0, 0, 0, 1, 3, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 2, 1, 1, 1], [1, 0, 1, 1, 0, 2, 1, 2, 1, 1, 2, 0, 2, 1, 0, 0, 1, 3, 0, 0, 1, 0, 2, 0, 3, 0, 0, 0, 0, 1, 0, 0, 0, 4, 0, 1, 1, 1, 0, 0, 0, 0, 2, 0, 1, 3, 1, 1, 1, 2, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 2, 1, 1, 1, 3, 0, 3, 1, 1, 1, 0, 2, 0, 0, 0, 1, 0, 1, 3, 1, 1, 1, 1, 2, 0, 1, 4, 1, 0, 1, 1, 1, 1, 1, 0, 0, 2], [2, 0, 1, 1, 1, 2, 0, 0, 1, 1, 1, 1, 0, 1, 0, 4, 0, 0, 2, 0, 2, 1, 1, 1, 1, 3, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 2, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 3, 1, 0, 1, 2, 1, 0, 1, 2, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 3, 2, 0, 0, 2, 0, 1, 1, 0, 0, 1, 2, 1, 0, 0, 2, 0, 1, 2, 1, 1, 1, 0, 2, 1, 0]]

sample output:

item-based CF, row 50 / 50, col 22 predicted:
4

item-based CF, row 50 / 50, col 23 predicted:
4

item-based CF, row 50 / 50, col 24 predicted:
4

item-based CF, row 50 / 50, col 25 predicted:
4

item-based CF, row 50 / 50, col 26 predicted:
4

item-based CF, row 50 / 50, col 27 predicted:
3

item-based CF, row 50 / 50, col 28 predicted:
5

item-based CF, row 50 / 50, col 29 predicted:
3

item-based CF, row 50 / 50, col 30 predicted:
4

item-based CF, row 50 / 50, col 31 predicted:
2

item-based CF, row 50 / 50, col 32 predicted:
4

item-based CF, row 50 / 50, col 33 predicted:
4

item-based CF, row 50 / 50, col 34 predicted:
3

item-based CF, row 50 / 50, col 35 predicted:
5

item-based CF, row 50 / 50, col 36 predicted:
4

item-based CF, row 50 / 50, col 37 predicted:
3

item-based CF, row 50 / 50, col 38 predicted:
5

item-based CF, row 50 / 50, col 39 predicted:
2

item-based CF, row 50 / 50, col 40 predicted:
4

item-based CF, row 50 / 50, col 41 predicted:
4

item-based CF, row 50 / 50, col 42 predicted:
5

item-based CF, row 50 / 50, col 43 predicted:
4

item-based CF, row 50 / 50, col 44 predicted:
4

item-based CF, row 50 / 50, col 45 predicted:
3

item-based CF, row 50 / 50, col 46 predicted:
3

item-based CF, row 50 / 50, col 47 predicted:
4

item-based CF, row 50 / 50, col 48 predicted:
4

item-based CF, row 50 / 50, col 49 predicted:
4

item-based CF, row 50 / 50, col 50 predicted:
3

item-based CF, row 50 / 50, col 51 predicted:
4

item-based CF, row 50 / 50, col 52 predicted:
3

item-based CF, row 50 / 50, col 53 predicted:
4

item-based CF, row 50 / 50, col 54 predicted:
3

item-based CF, row 50 / 50, col 55 predicted:
4

item-based CF, row 50 / 50, col 56 predicted:
3

item-based CF, row 50 / 50, col 57 predicted:
3

item-based CF, row 50 / 50, col 58 predicted:
3

item-based CF, row 50 / 50, col 59 predicted:
4

item-based CF, row 50 / 50, col 60 predicted:
4

item-based CF, row 50 / 50, col 61 predicted:
1

item-based CF, row 50 / 50, col 62 predicted:
3

item-based CF, row 50 / 50, col 63 predicted:
4

item-based CF, row 50 / 50, col 64 predicted:
3

item-based CF, row 50 / 50, col 65 predicted:
4

item-based CF, row 50 / 50, col 66 predicted:
4

item-based CF, row 50 / 50, col 67 predicted:
4

item-based CF, row 50 / 50, col 68 predicted:
4

item-based CF, row 50 / 50, col 69 predicted:
3

item-based CF, row 50 / 50, col 70 predicted:
4

item-based CF, row 50 / 50, col 71 predicted:
4

item-based CF, row 50 / 50, col 72 predicted:
4

item-based CF, row 50 / 50, col 73 predicted:
3

item-based CF, row 50 / 50, col 74 predicted:
4

item-based CF, row 50 / 50, col 75 predicted:
4

item-based CF, row 50 / 50, col 76 predicted:
5

item-based CF, row 50 / 50, col 77 predicted:
3

item-based CF, row 50 / 50, col 78 predicted:
3

item-based CF, row 50 / 50, col 79 predicted:
4

item-based CF, row 50 / 50, col 80 predicted:
4

item-based CF, row 50 / 50, col 81 predicted:
4

item-based CF, row 50 / 50, col 82 predicted:
4

item-based CF, row 50 / 50, col 83 predicted:
4

item-based CF, row 50 / 50, col 84 predicted:
4

item-based CF, row 50 / 50, col 85 predicted:
3

item-based CF, row 50 / 50, col 86 predicted:
3

item-based CF, row 50 / 50, col 87 predicted:
4

item-based CF, row 50 / 50, col 88 predicted:
3

item-based CF, row 50 / 50, col 89 predicted:
3

item-based CF, row 50 / 50, col 90 predicted:
4

item-based CF, row 50 / 50, col 91 predicted:
4

item-based CF, row 50 / 50, col 92 predicted:
5

item-based CF, row 50 / 50, col 93 predicted:
5

item-based CF, row 50 / 50, col 94 predicted:
4

item-based CF, row 50 / 50, col 95 predicted:
4

item-based CF, row 50 / 50, col 96 predicted:
3

item-based CF, row 50 / 50, col 97 predicted:
3

item-based CF, row 50 / 50, col 98 predicted:
3

item-based CF, row 50 / 50, col 99 predicted:
3

---

how does your choice for <i> affect user-based collaborative filtering?

distance = 1 #0 for pearsons, 1 for manhattan
k = 3

i = 0 #changed i from 1 to 0 for this experiemnt
numUsers = 943
numItems = 1682

#made changes in the code to account for edge cases when iFlag = 0

average error = 1.55

average errors:
[1.48, 1.6000000000000001, 1.6599999999999999, 1.8, 1.52, 1.54, 1.5800000000000001, 1.52, 1.6100000000000001, 1.8400000000000001, 1.52, 1.49, 1.5800000000000001, 1.45, 1.8200000000000001, 1.71, 1.6200000000000001, 1.6100000000000001, 1.53, 1.9199999999999999, 1.74, 1.6299999999999999, 1.8600000000000001, 1.5700000000000001, 1.55]

user errors:
[[0, 1, 1, 1, 2, 0, 2, 1, 3, 2, 2, 1, 0, 0, 2, 0, 0, 1, 1, 0, 2, 4, 2, 4, 0, 1, 1, 2, 3, 1, 0, 4, 3, 3, 4, 0, 4, 3, 1, 3, 3, 3, 2, 2, 3, 0, 1, 1, 2, 2, 3, 0, 1, 2, 2, 3, 2, 0, 1, 1, 0, 0, 2, 2, 1, 0, 4, 1, 1, 1, 2, 4, 0, 0, 0, 0, 0, 1, 4, 0, 1, 1, 0, 1, 1, 1, 2, 3, 1, 3, 0, 2, 3, 2, 0, 0, 2, 1, 2], [1, 0, 1, 3, 1, 2, 3, 4, 3, 3, 0, 2, 1, 1, 2, 4, 4, 0, 2, 2, 2, 3, 4, 0, 1, 3, 2, 4, 3, 2, 1, 0, 2, 0, 1, 1, 1, 1, 0, 4, 2, 4, 3, 1, 0, 0, 1, 4, 4, 3, 1, 0, 0, 0, 2, 0, 1, 3, 1, 1, 1, 2, 3, 3, 1, 1, 0, 1, 1, 2, 1, 1, 2, 1, 1, 0, 0, 2, 1, 2, 2, 2, 1, 2, 0, 1, 1, 2, 2, 2, 0, 1, 4, 0, 3, 0, 2, 2, 2, 0], [1, 1, 4, 1, 2, 0, 0, 1, 1, 2, 2, 3, 2, 4, 2, 0, 2, 3, 4, 3, 2, 4, 1, 2, 0, 1, 1, 2, 1, 1, 0, 3, 1, 3, 2, 4, 1, 2, 1, 2, 2, 1, 1, 4, 3, 0, 0, 3, 3, 3, 1, 2, 3, 1, 2, 0, 2, 4, 3, 4, 2, 1, 2, 0, 3, 0, 1, 1, 1, 2, 1, 3, 2, 3, 0, 2, 2, 2, 0, 3, 2, 1, 0, 0, 0, 1, 3, 1, 2, 3, 0, 1, 1, 1, 1, 0, 2, 0, 1, 1], [0, 3, 3, 3, 3, 0, 2, 0, 1, 0, 1, 1, 3, 2, 0, 0, 2, 2, 2, 1, 3, 2, 1, 4, 3, 3, 0, 1, 0, 1, 1, 2, 1, 1, 3, 2, 1, 1, 1, 3, 1, 1, 4, 1, 2, 0, 3, 2, 1, 1, 1, 2, 2, 0, 0, 2, 2, 0, 2, 3, 2, 1, 3, 1, 3, 2, 1, 3, 2, 1, 3, 3, 1, 3, 3, 1, 3, 3, 3, 1, 3, 4, 2, 2, 2, 2, 1, 3, 3, 0, 4, 4, 3, 0, 3, 3, 0, 3, 1, 1], [3, 1, 2, 3, 3, 1, 0, 2, 3, 0, 3, 2, 2, 2, 2, 1, 1, 4, 2, 1, 0, 1, 1, 1, 2, 2, 0, 1, 1, 4, 0, 3, 1, 0, 2, 0, 2, 1, 3, 0, 2, 0, 2, 3, 0, 0, 1, 4, 3, 1, 2, 3, 2, 3, 1, 1, 3, 3, 3, 4, 0, 4, 1, 1, 3, 0, 2, 0, 1, 0, 1, 1, 1, 1, 0, 4, 4, 0, 0, 2, 0, 0, 1, 0, 4, 1, 0, 2, 3, 1, 0, 1, 0, 0, 1, 0, 2, 1, 4, 1], [1, 3, 1, 4, 1, 1, 2, 2, 0, 1, 2, 4, 3, 3, 3, 1, 0, 1, 4, 2, 3, 1, 2, 2, 1, 1, 4, 3, 4, 2, 0, 1, 1, 1, 2, 3, 2, 2, 2, 1, 4, 0, 2, 1, 0, 1, 4, 1, 2, 0, 2, 2, 2, 1, 1, 3, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 2, 0, 1, 1, 3, 0, 1, 0, 3, 0, 1, 1, 2, 1, 0, 3, 0, 2, 2, 4, 1, 1, 1, 0, 0, 1, 1, 1, 4, 2, 1, 3], [3, 1, 0, 4, 0, 2, 0, 2, 1, 2, 2, 1, 0, 2, 4, 3, 2, 3, 1, 4, 4, 0, 2, 1, 0, 1, 1, 1, 2, 2, 1, 0, 0, 1, 4, 2, 1, 0, 1, 0, 2, 3, 1, 1, 3, 1, 1, 1, 4, 4, 1, 1, 0, 1, 3, 3, 1, 1, 1, 3, 0, 2, 1, 2, 0, 4, 0, 0, 1, 2, 2, 4, 2, 1, 4, 2, 2, 0, 2, 3, 3, 3, 1, 0, 0, 3, 1, 2, 2, 1, 1, 0, 0, 1, 1, 1, 2, 2, 2, 1], [1, 1, 2, 2, 0, 1, 2, 1, 1, 3, 1, 1, 3, 2, 0, 2, 0, 1, 0, 2, 1, 1, 2, 3, 1, 1, 3, 2, 0, 1, 4, 1, 1, 2, 2, 0, 4, 1, 3, 2, 0, 1, 1, 1, 2, 2, 2, 0, 3, 0, 1, 0, 2, 3, 1, 3, 0, 1, 3, 2, 3, 4, 2, 1, 1, 2, 1, 1, 0, 4, 0, 1, 1, 3, 0, 3, 2, 2, 3, 2, 3, 1, 1, 0, 1, 2, 3, 1, 1, 0, 1, 1, 3, 1, 0, 2, 1, 1, 1, 3], [4, 2, 1, 0, 0, 4, 1, 0, 3, 4, 1, 1, 1, 3, 1, 0, 1, 2, 1, 1, 3, 1, 2, 1, 3, 2, 3, 0, 3, 1, 3, 2, 3, 1, 2, 2, 1, 1, 0, 4, 0, 3, 0, 1, 3, 0, 4, 2, 2, 1, 1, 0, 0, 1, 3, 1, 0, 3, 1, 2, 0, 1, 0, 1, 4, 0, 2, 2, 4, 1, 1, 1, 2, 2, 1, 0, 2, 4, 3, 3, 3, 0, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 4, 0, 4, 0, 1, 1, 2], [0, 4, 3, 2, 1, 2, 2, 3, 0, 4, 0, 3, 3, 1, 2, 1, 2, 3, 4, 1, 1, 0, 1, 2, 3, 2, 1, 4, 1, 1, 2, 2, 1, 0, 3, 1, 2, 1, 3, 2, 1, 3, 3, 2, 0, 1, 1, 1, 2, 3, 0, 1, 1, 2, 1, 2, 2, 1, 2, 3, 3, 3, 1, 1, 1, 0, 3, 2, 2, 3, 3, 4, 3, 0, 1, 3, 2, 2, 1, 2, 0, 2, 2, 4, 2, 4, 1, 4, 1, 3, 4, 1, 0, 2, 4, 1, 0, 0, 1, 2], [1, 1, 2, 0, 0, 1, 1, 1, 1, 0, 0, 1, 2, 1, 1, 4, 1, 4, 4, 1, 1, 1, 0, 3, 2, 1, 1, 3, 1, 3, 3, 0, 2, 4, 2, 3, 2, 3, 3, 1, 2, 1, 4, 3, 2, 1, 0, 0, 4, 0, 0, 1, 1, 2, 1, 4, 1, 1, 3, 0, 2, 2, 0, 2, 0, 0, 1, 2, 2, 1, 0, 2, 2, 0, 2, 3, 1, 1, 1, 2, 1, 0, 2, 3, 1, 0, 1, 2, 0, 2, 3, 2, 2, 2, 1, 3, 0, 1, 0, 3], [2, 1, 3, 0, 0, 3, 1, 2, 3, 0, 1, 3, 1, 0, 1, 3, 1, 3, 0, 3, 2, 4, 3, 2, 3, 1, 2, 2, 3, 1, 1, 0, 1, 1, 2, 3, 1, 1, 1, 2, 4, 1, 1, 2, 2, 3, 4, 0, 0, 0, 2, 1, 1, 1, 4, 2, 0, 0, 3, 1, 4, 3, 0, 1, 2, 2, 1, 3, 1, 4, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 3, 1, 1, 0, 2, 0, 0, 0, 1, 1, 2, 1, 1, 0, 2, 2, 3, 0, 2, 1], [0, 2, 1, 3, 1, 1, 0, 2, 0, 0, 0, 1, 1, 2, 1, 4, 3, 4, 0, 1, 2, 2, 2, 2, 1, 3, 0, 3, 1, 4, 2, 1, 1, 2, 1, 2, 1, 0, 1, 0, 1, 0, 4, 2, 3, 1, 1, 1, 1, 4, 3, 3, 1, 2, 1, 0, 4, 2, 2, 2, 3, 2, 2, 3, 1, 3, 4, 1, 0, 2, 0, 2, 1, 4, 4, 4, 1, 0, 2, 0, 0, 0, 0, 1, 3, 2, 0, 0, 1, 4, 0, 0, 3, 3, 0, 0, 1, 3, 1, 1],
[1, 1, 0, 0, 2, 2, 2, 3, 2, 0, 1, 2, 1, 1, 2, 0, 2, 0, 2, 1, 2, 1, 0, 1, 0, 2, 3, 2, 1, 2, 2, 1, 0, 1, 0, 1, 2, 4, 2, 1, 2, 1, 4, 1, 0, 3, 0, 1, 0, 0, 1, 2, 0, 2, 1, 1, 4, 1, 3, 0, 2, 1, 2, 1, 2, 0, 4, 4, 1, 2, 2, 3, 1, 1, 4, 1, 0, 0, 1, 3, 0, 1, 0, 0, 1, 1, 1, 2, 1, 2, 0, 4, 3, 1, 2, 3, 1, 1, 2, 4], [0, 3, 1, 1, 3, 2, 0, 3, 1, 2, 4, 3, 0, 0, 1, 0, 2, 1, 2, 1, 0, 3, 2, 2, 2, 1, 3, 4, 0, 1, 2, 0, 0, 3, 1, 2, 2, 2, 1, 2, 0, 4, 2, 0, 2, 1, 2, 0, 4, 2, 3, 2, 4, 1, 2, 2, 4, 2, 3, 2, 2, 4, 4, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 2, 1, 3, 2, 3, 3, 3, 3, 2, 3, 1, 1, 2, 3, 3, 1, 0, 1, 2, 4, 3, 1, 3, 0, 1, 0, 1], [4, 3, 3, 2, 0, 1, 0, 2, 0, 1, 0, 1, 3, 0, 2, 3, 4, 1, 0, 2, 1, 0, 4, 1, 1, 2, 2, 2, 2, 0, 0, 1, 4, 0, 1, 1, 1, 2, 1, 2, 0, 0, 3, 4, 0, 0, 3, 4, 0, 2, 0, 0, 2, 3, 1, 0, 2, 1, 3, 2, 4, 1, 1, 0, 2, 2, 2, 4, 2, 4, 1, 0, 4, 2, 2, 2, 2, 3, 2, 0, 3, 1, 1, 2, 0, 0, 3, 2, 2, 3, 4, 3, 3, 2, 1, 4, 0, 3, 0, 4], [4, 2, 1, 2, 2, 0, 0, 4, 0, 3, 2, 0, 1, 0, 0, 2, 2, 0, 2, 3, 0, 0, 3, 1, 3, 1, 3, 1, 1, 0, 1, 2, 3, 2, 2, 4, 1, 4, 2, 2, 2, 1, 1, 4, 2, 1, 4, 2, 3, 0, 3, 0, 1, 0, 2, 0, 1, 1, 1, 1, 1, 2, 3, 3, 1, 3, 1, 1, 3, 3, 3, 0, 3, 0, 0, 0, 3, 1, 2, 2, 2, 0, 4, 0, 3, 4, 0, 2, 2, 0, 3, 2, 0, 4, 2, 2, 0, 0, 1, 0], [2, 3, 0, 0, 2, 2, 1, 3, 0, 2, 2, 0, 1, 1, 1, 3, 3, 2, 0, 4, 3, 1, 3, 0, 2, 1, 3, 0, 2, 0, 4, 1, 1, 2, 2, 0, 4, 2, 1, 3, 4, 4, 1, 1, 1, 0, 1, 2, 4, 4, 1, 3, 2, 1, 1, 1, 2, 3, 3, 1, 0, 1, 4, 2, 2, 2, 2, 1, 0, 2, 4, 2, 2, 0, 2, 0, 1, 1, 2, 4, 2, 1, 0, 1, 2, 0, 2, 1, 3, 1, 1, 0, 2, 0, 1, 1, 1, 0, 1, 0], [2, 1, 3, 3, 0, 4, 2, 2, 1, 4, 1, 3, 2, 4, 3, 2, 3, 3, 4, 1, 1, 3, 1, 1, 2, 0, 1, 1, 3, 4, 3, 2, 1, 3, 0, 0, 4, 2, 2, 3, 3, 2, 0, 0, 1, 0, 2, 3, 3, 1, 0, 1, 0, 0, 0, 2, 2, 0, 2, 3, 0, 0, 4, 0, 0, 3, 2, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 2, 3, 1, 0, 2, 0, 0, 2, 0, 4, 0, 0, 0, 1, 3, 0, 0, 4, 0, 2, 2, 1, 1], [0, 1, 4, 0, 1, 1, 4, 3, 2, 3, 1, 3, 1, 4, 3, 1, 2, 3, 0, 1, 1, 4, 1, 3, 3, 0, 0, 3, 3, 1, 4, 1, 2, 4, 3, 3, 1, 2, 1, 0, 0, 3, 2, 1, 0, 2, 3, 4, 1, 2, 4, 0, 1, 4, 4, 2, 1, 0, 2, 2, 1, 2, 1, 1, 0, 0, 3, 3, 4, 1, 1, 2, 4, 1, 1, 3, 0, 1, 1, 2, 4, 2, 1, 1, 3, 2, 1, 3, 1, 2, 3, 2, 3, 2, 3, 0, 1, 3, 4, 3], [0, 1, 4, 3, 0, 2, 3, 2, 0, 1, 0, 3, 2, 2, 2, 3, 3, 4, 0, 2, 1, 1, 2, 2, 1, 1, 1, 0, 1, 2, 1, 3, 2, 2, 3, 3, 1, 1, 1, 4, 1, 0, 2, 1, 0, 2, 1, 2, 0, 1, 4, 3, 1, 4, 1, 0, 3, 2, 1, 2, 0, 4, 3, 0, 2, 2, 1, 1, 1, 1, 4, 3, 3, 2, 2, 0, 2, 3, 3, 0,

2, 1, 2, 4, 0, 3, 0, 1, 3, 3, 3, 1, 2, 3, 4, 1, 1, 0, 2, 0], [1, 3, 3, 2, 3, 2, 1, 0, 0, 1, 2, 2, 0, 3, 0, 4, 2, 3, 0, 1, 0, 3, 2, 1, 2, 1, 1, 1, 1, 1, 1, 3, 4, 1, 2, 1, 0, 3, 0, 0, 2, 0, 2, 1, 3, 3, 1, 0, 4, 4, 2, 1, 2, 1, 3, 4, 2, 3, 1, 2, 1, 3, 0, 0, 1, 3, 0, 0, 2, 2, 4, 3, 1, 2, 2, 3, 1, 1, 1, 4, 1, 0, 2, 0, 0, 2, 3, 0, 1, 3, 2, 3, 2, 1, 3, 1, 0, 1, 1], [0, 3, 4, 1, 1, 1, 4, 2, 1, 0, 2, 0, 1, 0, 1, 2, 0, 1, 4, 4, 1, 1, 3, 4, 3, 3, 3, 2, 1, 1, 3, 2, 0, 3, 1, 2, 0, 0, 3, 4, 4, 1, 1, 1, 4, 1, 0, 1, 2, 3, 3, 4, 0, 1, 2, 4, 2, 3, 2, 4, 4, 0, 4, 2, 1, 1, 0, 3, 0, 0, 1, 0, 1, 4, 2, 4, 0, 4, 3, 0, 1, 3, 3, 1, 3, 0, 3, 3, 1, 3, 3, 3, 2, 1, 2, 1, 2, 1, 1, 0], [1, 3, 1, 1, 0, 2, 1, 1, 1, 3, 2, 3, 3, 1, 2, 3, 2, 1, 3, 1, 3, 1, 2, 3, 1, 3, 0, 3, 1, 0, 0, 1, 0, 0, 4, 0, 0, 2, 0, 3, 0, 1, 3, 1, 0, 2, 2, 2, 2, 0, 4, 3, 3, 0, 0, 1, 0, 3, 1, 1, 2, 1, 2, 1, 1, 0, 1, 0, 3, 1, 1, 4, 0, 0, 1, 1, 4, 1, 0, 1, 1, 2, 2, 2, 4, 4, 3, 1, 4, 3, 2, 2, 0, 0, 4, 1, 0, 2, 0, 4], [0, 2, 0, 2, 0, 3, 0, 0, 3, 2, 3, 3, 0, 3, 3, 2, 1, 1, 2, 4, 0, 0, 1, 1, 0, 1, 2, 4, 3, 1, 2, 2, 0, 4, 3, 1, 4, 2, 2, 0, 1, 0, 2, 3, 3, 0, 2, 0, 3, 1, 0, 0, 2, 2, 2, 2, 0, 1, 2, 0, 1, 4, 1, 3, 2, 1, 0, 2, 2, 1, 1, 3, 1, 1, 3, 3, 0, 4, 4, 0, 2, 3, 0, 1, 2, 1, 0, 0, 1, 2, 3, 0, 1, 2, 0, 1, 1, 2, 2, 1]]

#these optimal parameters, running tests on user-based filtering
distance = 1 #0 for pearsons, 1 for manhattan
k = 1
i = 1
numUsers = 943
numItems = 1682

average erros
.834

user average errors
[0.72999999999999998, 0.77000000000000002, 0.96999999999999997, 0.73999999999999999, 0.97999999999999998, 0.81999999999999995, 0.83999999999999997, 0.94999999999999996, 0.83999999999999997, 0.87, 0.82999999999999996, 0.75, 0.85999999999999999, 0.76000000000000001, 1.01, 0.69999999999999996, 0.63, 0.92000000000000004, 0.75, 0.81999999999999995, 0.82999999999999996, 0.93000000000000005, 0.82999999999999996, 0.89000000000000001, 0.82999999999999996]

user errors
[[0, 1, 0, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 1, 2, 2, 1, 1, 0, 1, 0, 0, 1, 1, 2, 0, 0, 0, 1, 1, 0, 0, 0, 2, 1, 1, 2, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 2, 0, 1, 1, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 3, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 4, 2, 1, 0, 0, 3, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0], [1, 1, 0, 0, 3, 0, 0, 0, 3, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 2, 1, 1, 0, 2, 0, 2, 2, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 2, 0, 1, 1, 1, 2, 0, 0, 2, 2, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 3, 0, 2, 0, 0, 1, 0, 0, 1, 1, 3, 3, 1, 0, 0, 1, 0, 0, 0, 1, 1, 2], [1, 2, 2, 1, 1, 1, 1, 1, 3, 1, 2, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 2, 0, 2, 1, 3, 0, 0, 1, 0, 2, 1, 2, 0, 1, 1, 0, 1, 2, 1, 1, 1, 3, 3, 0, 0, 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 3, 0, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 2, 2, 2, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 2, 0, 0, 0, 1, 1, 0, 1, 0, 4, 3, 0, 1, 0, 0], [1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 3, 0, 1, 0, 1, 1, 0, 0, 2, 2, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 3, 0, 2, 0, 0, 1, 1, 0, 1, 0, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 3, 0, 0, 0, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 2, 3, 1, 1, 0, 0, 2, 0, 0], [1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 2, 1, 0, 1, 1, 1, 2, 1, 0, 1, 4, 1, 1, 1, 3, 3, 1, 1, 0, 2, 1, 2, 0, 1, 0, 1, 1, 0, 3, 2, 2, 2, 3, 0, 1, 1, 2, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 2, 2, 1, 2, 0, 1, 0, 0, 2, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 2, 0, 1, 2, 0, 1, 1, 1, 0, 2, 1, 1, 0, 2, 0, 1, 0, 0, 0, 0, 1, 1, 3, 0], [1, 0, 2, 1, 1, 0, 3, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 2, 1, 1, 0, 1, 1, 0, 1, 0, 2, 0, 1, 0, 2, 0, 1, 1, 0, 2, 1, 1, 0, 2, 1, 1, 1, 2, 0, 0,

0, 1, 1, 2, 1, 1, 1, 1, 1, 3, 3, 0, 1, 1, 0, 4, 0, 0, 0, 1, 0, 1, 0, 2, 2, 1, 1, 0], [0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 2, 2, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 2, 0, 0, 3, 2, 2, 0, 1, 3, 1, 2, 0, 2, 1, 1, 2, 0, 1, 2, 1, 4, 0, 2, 0, 2, 0, 2, 4, 1, 0, 0, 0, 1, 0, 0, 3, 1, 1, 0, 1, 1, 0, 1, 1, 3, 0, 3, 0, 0, 0, 0, 1, 0, 0, 0, 2, 1, 1, 0, 0, 2, 1, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0], [1, 2, 2, 2, 1, 2, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 1, 0, 2, 2, 1, 2, 1, 1, 0, 0, 0, 1, 0, 1, 2, 1, 3, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 1, 1, 2, 0, 0, 1, 3, 2, 1, 0, 1, 1, 2, 2, 1, 1, 0, 2, 3, 0, 1, 1, 2, 2, 1, 1, 0], [1, 1, 0, 1, 1, 3, 3, 2, 1, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 2, 2, 1, 0, 2, 0, 2, 1, 2, 2, 2, 1, 0, 1, 2, 1, 1, 0, 0, 0, 2, 0, 0, 1, 0, 1, 0, 1, 0, 2, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 0, 4, 1, 1, 0, 1, 1], [0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 2, 2, 2, 1, 1, 2, 0, 1, 3, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 2, 1, 2, 1, 1, 1, 0, 1, 0, 1, 2, 2, 1, 0, 1, 2, 2, 0, 1, 0, 0, 1, 1, 2, 1, 2, 1, 3, 0, 0, 0, 2, 0, 1], [1, 1, 1, 2, 1, 1, 1, 1, 1, 0, 0, 1, 1, 4, 1, 1, 0, 3, 1, 2, 1, 3, 1, 1, 0, 0, 0, 3, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 2, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 2, 1, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 2, 1, 1, 0, 1, 1, 0, 1, 2, 0, 0, 2, 2, 1, 1, 2, 1, 0, 1, 1, 1, 0, 3], [0, 2, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 2, 1, 0, 1, 0, 1, 0, 2, 0, 2, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 3, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 1, 0, 2, 0, 1, 0, 0, 3, 1, 0, 0, 0, 1, 1, 1, 1, 1, 2, 4, 0, 1, 1, 2, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 2, 1, 2, 1], [1, 1, 0, 0, 2, 2, 0, 0, 1, 0, 1, 2, 0, 1, 2, 0, 1, 1, 0, 2, 0, 1, 3, 0, 2, 0, 3, 1, 2, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 3, 0, 1, 1, 2, 0, 1, 0, 0, 2, 2, 0, 1, 0, 1, 0, 0, 0, 1, 2, 2, 0, 1, 0, 2, 0, 1, 1, 0, 1, 0, 0, 2, 1, 2, 1, 1, 0, 1, 2, 0, 2, 1, 1, 0, 0, 1, 1, 1, 0, 0, 2, 0, 0, 0, 1, 1, 1, 2, 0, 4], [1, 4, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 2, 0, 2, 1, 0, 2, 0, 1, 1, 0, 2, 0, 0, 2, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 2, 0, 0, 4, 1, 0, 1, 3, 1, 1, 0, 2, 1, 0, 3, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 2, 1, 0, 0, 1, 1, 0], [1, 1, 2, 1, 1, 2, 0, 1, 0, 0, 2, 2, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 3, 2, 1, 1, 3, 1, 0, 1, 2, 1, 4, 0, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 2, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 3, 2, 1, 1, 1, 1, 1, 0, 0, 0, 2, 1, 2, 1, 0, 1, 4, 3, 0, 1, 2, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1], [0, 0, 2, 3, 1, 1, 0, 1, 1, 2, 1, 0, 0, 0, 1, 1, 2, 0, 0, 1, 0, 0, 1, 2, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 3, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 2, 0, 0, 2, 2, 0, 0, 1, 0, 2, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 4, 1, 0, 1, 0, 2, 0, 0, 0, 0, 1, 2, 1, 0, 0, 1, 1, 0], [2, 1, 0, 0, 0, 1, 1, 2, 1, 0, 0, 0, 1, 0, 2, 0, 1, 1, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 1, 0, 1, 0, 0, 2, 0, 2, 3, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 2, 0, 2, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 1, 1, 0, 2, 1, 0, 2, 1, 1, 2, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1] , [1, 2, 1, 2, 1, 1, 1, 0, 2, 0, 2, 1, 1, 4, 1, 1, 1, 2, 1, 4, 1, 0, 1, 0, 2, 0, 0, 0, 2, 0, 0, 1, 2, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 3, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 2, 0, 2, 0, 0, 1, 1, 2, 0, 1, 0, 1, 2, 1, 1, 0, 0, 1, 2, 1, 2, 0, 0, 1, 1, 1, 0, 0, 1, 2, 1, 1, 0, 0, 4] , [1, 0, 0, 0, 2, 0, 2, 0, 2, 1, 1, 0, 3, 1, 0, 0, 0, 0, 1, 1, 0, 2, 1, 0, 0, 1, 0, 2, 0, 1, 0, 2, 1, 0, 2, 1, 1, 0, 0, 2, 0, 1, 0, 2, 1, 0, 1, 1, 0, 0, 1, 0, 2, 0, 1, 3, 0, 1, 0, 2, 1, 1, 1, 0, 0, 3, 0, 1, 1, 0, 0, 3, 0, 0, 1, 0, 1, 3, 1, 2, 0, 1, 0, 4, 0, 2, 0, 0, 0, 0, 1, 1, 0, 2, 0, 2, 0, 1, 0, 2, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 2, 0, 1, 1, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 1, 0, 1, 2, 0, 1, 0, 0, 1] , [0, 1, 2, 3, 0, 0, 2, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 2, 2, 0, 1, 2, 1, 2, 1, 1, 2, 2, 2, 0, 2, 1, 0, 1, 1, 0, 0, 1, 0, 2, 0, 1, 1, 2, 2, 2, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 2, 1, 2, 1, 1, 0, 0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 0, 1, 0, 2, 0, 1, 0, 1, 1, 0, 1, 2, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0] , [2, 1, 2, 0, 1, 0, 2, 1, 1, 1, 0, 0, 0, 1, 1, 1, 2, 1, 0, 0, 2, 1, 0, 1, 0, 2, 1, 3, 2, 2, 1, 2, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 2, 0, 0, 1, 2, 0, 0, 2, 1, 0, 0, 2, 1, 0, 1, 2, 0, 0, 2, 0, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 3, 2, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 2, 2, 1, 0, 2] , [1, 0, 0, 0, 4, 2, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 2, 1, 0, 1, 0, 2, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 2, 0, 0, 1, 1, 1, 2, 0, 1, 2, 2, 1, 0, 1, 1, 3, 2, 2, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 2, 1, 0, 2, 3, 3, 0, 0, 0, 1, 1, 0, 2, 2, 0, 0, 0, 0, 0, 1, 2, 1] , [0, 0, 0, 1, 0, 2, 1, 1, 1, 3, 0, 1, 0, 1, 0, 0, 2, 1, 1, 3, 1, 0, 1, 0, 0, 0, 0, 2, 0, 0, 1, 0, 1, 1, 1, 1, 1, 2, 0, 3, 1, 1, 1, 0, 1, 2, 1, 3, 1, 0, 2, 0, 0, 1, 2, 1, 0, 0, 0, 0, 1, 2, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 3, 2, 1, 1, 3, 1, 2, 1, 1, 1, 0, 0, 2, 0, 1, 2, 0, 1, 2, 1, 0, 0, 0, 0, 2, 1] , [0, 1, 0, 1, 2, 0, 0, 2, 1, 0, 0, 2, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 2, 0, 0, 1, 1, 0, 2, 0, 0, 1, 1, 1, 3, 1, 0, 0, 1, 2, 0, 2, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 2, 0, 1, 1, 4, 2, 0, 1, 0, 0, 1, 0, 0, 0, 3, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 2, 1, 0, 0, 1, 2, 4, 0, 3, 1, 1, 0, 0, 1, 2, 0, 2]]

---

k = 2

user error average:
0.834

user error averages:
[0.92000000000000004, 0.94999999999999996, 0.73999999999999999, 0.79000000000000004, 0.71999999999999997, 0.89000000000000001, 0.85999999999999999, 0.90000000000000002, 0.88, 0.84999999999999998, 0.85999999999999999, 0.89000000000000001, 0.84999999999999998, 0.91000000000000003, 1.1000000000000001, 0.72999999999999998, 0.95999999999999996, 0.97999999999999998, 0.91000000000000003, 0.81000000000000005, 1.01, 0.78000000000000003, 0.85999999999999999, 0.98999999999999999, 0.93999999999999995]

user errors:

[[2, 0, 1, 4, 1, 0, 1, 1, 0, 1, 1, 3, 1, 0, 1, 2, 0, 1, 0, 0, 1, 2, 1, 1, 3, 0, 0, 3, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 2, 1, 2, 0, 1, 0, 1, 0, 0, 1, 2, 1, 0,
1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 3, 0, 2, 1, 1, 2, 0, 2, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 2, 3, 1, 1, 1, 2, 1, 1, 1, 0, 0, 0, 1, 1, 2, 1], [1, 0, 1, 1,
0, 0, 1, 1, 1, 1, 2, 1, 1, 0, 1, 2, 0, 2, 0, 2, 1, 1, 0, 0, 1, 0, 1, 1, 2, 1, 1, 2, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 2, 1, 1, 1, 1, 1, 0, 2, 2, 1, 1, 2,
1, 0, 1, 1, 2, 1, 0, 3, 1, 0, 1, 1, 1, 1, 0, 3, 1, 1, 1, 2, 0, 0, 1, 0, 2, 1, 1, 1, 0, 1, 0, 0, 1, 0, 3, 2, 1, 1, 1, 2, 1, 2, 1, 0], [1, 0, 0, 1, 1, 1, 1, 1,
0, 1, 0, 4, 4, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 2, 1, 0, 1, 2, 1, 0, 1, 1, 1, 2, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 2, 0, 2, 0, 0, 1, 0,
0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 2, 0, 1, 1, 0, 3, 1, 2, 0, 1, 0, 0, 1, 1, 0, 3, 1, 1, 1, 0, 1, 0], [2, 0, 1, 1, 1, 2, 2, 1, 0, 2, 1, 1,
1, 1, 0, 1, 1, 0, 0, 0, 2, 1, 0, 0, 1, 0, 0, 1, 0, 1, 2, 1, 0, 0, 0, 2, 0, 0, 0, 1, 0, 1, 0, 1, 1, 4, 2, 0, 2, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1,
1, 1, 0, 1, 1, 1, 3, 0, 1, 0, 1, 0, 0, 1, 3, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 3, 1, 2, 0, 1, 1, 2, 1, 0, 1, 1], [0, 1, 2, 0, 1, 0, 0, 2, 1, 1, 0, 0, 1, 1, 2, 0,
1, 0, 1, 2, 0, 1, 1, 1, 1, 2, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 2, 0, 1, 1, 0, 0, 2, 1, 1, 0, 2, 1, 1, 2, 1, 0, 1, 0,
2, 1, 1, 1, 1, 0, 0, 1, 0, 2, 1, 1, 0, 1, 1, 0, 0, 0, 0, 2, 1, 0, 0, 1, 1, 0, 0, 0, 2, 0, 1, 0], [1, 1, 1, 1, 2, 0, 3, 0, 0, 2, 0, 1, 1, 1, 1, 0, 1, 1, 0, 2,
1, 1, 0, 3, 1, 1, 2, 0, 0, 1, 3, 1, 1, 0, 0, 0, 2, 1, 4, 2, 1, 0, 1, 0, 0, 1, 0, 1, 0, 2, 0, 3, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0,
0, 1, 1, 2, 0, 0, 2, 1, 1, 2, 0, 0, 0, 0, 1, 1, 1, 3, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1], [1, 1, 1, 1, 1, 2, 0, 1, 0, 0, 1, 3, 1, 0, 0, 0, 2, 0, 0, 1, 1, 0, 1, 1,
1, 1, 0, 0, 0, 1, 0, 2, 1, 0, 1, 0, 2, 1, 0, 1, 1, 2, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 3, 2, 0, 0, 0, 2, 1, 0, 1, 1, 4, 1, 1, 1, 1, 0, 2, 1, 1, 0, 1, 0, 0,
2, 1, 0, 1, 3, 2, 0, 0, 0, 0, 1, 3, 1, 0, 3, 1, 0, 1, 1, 1, 0, 1, 3, 0], [2, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 2, 0, 1, 0, 2, 1, 2, 1, 2, 0, 2, 0, 1, 0, 0, 0,
1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 2, 4, 0, 0, 1, 0, 3, 2, 0, 1, 1, 0, 1, 0, 1, 0, 2, 2, 0, 1, 1, 0, 1, 0, 0, 1, 0, 3, 0, 1, 2, 1, 1, 2, 0, 0, 0, 1, 1, 2,
2, 4, 0, 1, 1, 0, 0, 2, 0, 1, 1, 1, 2, 2, 0, 0, 1, 0, 0, 1], [1, 0, 0, 1, 0, 1, 1, 0, 1, 3, 2, 1, 1, 0, 1, 1, 0, 1, 0, 0, 2, 1, 2, 1, 0, 1, 0, 0, 1, 1, 0, 1,
2, 1, 3, 2, 2, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 3, 1, 2, 1, 0, 0, 1, 3, 0, 1, 0, 0, 0, 0, 2, 2, 1, 2, 0, 1, 0, 1, 1, 1, 3, 0, 1, 0, 0, 0, 0, 1, 1, 3, 0,
1, 0, 0, 3, 2, 2, 0, 1, 0, 0, 1, 2, 0, 1, 0, 1], [0, 2, 0, 2, 1, 1, 1, 1, 0, 2, 0, 0, 0, 0, 2, 1, 0, 1, 3, 1, 0, 1, 2, 1, 0, 1, 0, 0, 0, 0, 0, 1, 2, 2, 0, 1,
1, 1, 0, 1, 0, 1, 2, 1, 0, 3, 2, 0, 1, 1, 1, 1, 0, 1, 3, 0, 1, 2, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 2, 1, 1, 1,
0, 1, 1, 1, 3, 1, 1, 0, 0, 2, 1, 2], [0, 1, 0, 1, 3, 0, 2, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 3, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 4, 0, 1,
2, 1, 1, 0, 0, 0, 0, 0, 2, 2, 1, 2, 1, 0, 0, 0, 2, 1, 1, 1, 1, 0, 1, 0, 2, 1, 1, 2, 0, 2, 0, 2, 2, 1, 0, 1, 0, 1, 1, 0, 2, 2, 0, 2, 1, 1, 0, 0, 1, 1, 2, 0,
1, 1, 0, 2, 0, 1, 1, 1], [0, 1, 0, 1, 0, 0, 1, 2, 2, 1, 1, 0, 0, 2, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 3, 2, 2, 0, 1, 1, 1, 2, 2, 0, 1, 2, 3, 1, 0, 2, 2, 0, 1,
0, 4, 0, 2, 1, 0, 2, 0, 0, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 1, 0, 2, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 4, 0, 0, 1, 0, 0, 0, 1, 0, 1, 2, 0, 2, 3, 1, 1, 0, 1, 1, 0,
1, 0, 2, 2], [0, 2, 1, 0, 2, 0, 2, 0, 1, 0, 0, 0, 1, 1, 2, 1, 1, 1, 1, 1, 2, 0, 2, 0, 2, 1, 0, 2, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 1, 1, 0, 2, 2, 0, 0,
0, 1, 0, 1, 0, 1, 1, 1, 2, 0, 1, 2, 1, 2, 1, 1, 3, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 2, 2, 1, 0, 1, 1, 0, 2, 0, 1],
[1, 1, 0, 0, 1, 0, 0, 2, 1, 0, 1, 0, 2, 2, 4, 0, 0, 1, 3, 2, 0, 1, 0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 3, 0, 1, 0, 2, 0, 1, 0, 0, 1, 3, 0, 1, 0, 0, 1, 0, 2, 0, 0,
0, 0, 0, 0, 0, 1, 2, 2, 0, 1, 0, 1, 2, 0, 1, 0, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 2, 0, 0, 1, 1, 1, 3, 2, 3, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 2, 1], [0, 0, 1, 2,
0, 3, 3, 0, 4, 2, 1, 3, 0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 1, 1, 2, 1, 1, 2, 0, 1, 3, 0, 0, 1, 1, 2, 1, 0, 2, 0, 3, 1, 1, 2, 1, 1, 2, 0, 0, 0, 2, 1, 1, 0, 1, 2,
1, 1, 4, 1, 0, 0, 2, 1, 1, 2, 3, 1, 0, 1, 3, 1, 1, 0, 3, 1, 0, 1, 0, 1, 1, 0, 0, 1, 3, 0, 1, 1, 0, 1, 1, 1, 3, 2, 1, 1, 0, 0, 0, 0], [1, 0, 1, 1, 1, 1, 0, 1,
1, 0, 2, 0, 0, 1, 1, 1, 2, 1, 1, 0, 2, 0, 0, 1, 1, 0, 3, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 2, 1, 2, 1, 0, 1, 1, 0, 1, 2, 0, 0, 0, 2, 0, 1, 2, 1, 0,
1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 2, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 2, 1, 0, 3, 0, 1, 2, 0, 0, 0, 0, 1], [2, 1, 2, 0, 1, 0, 1, 1, 1, 0, 1, 1,
1, 1, 0, 1, 0, 1, 0, 2, 0, 1, 3, 1, 0, 0, 1, 0, 2, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 3, 0, 2, 1, 1, 2, 1, 0, 0, 1, 2, 1, 1, 2, 2, 0, 1, 1, 3, 0,
2, 0, 0, 0, 0, 1, 1, 2, 1, 1, 0, 2, 1, 1, 2, 3, 1, 0, 0, 4, 0, 2, 1, 0, 0, 1, 2, 1, 2, 1, 0, 1, 2, 1, 2, 0], [0, 2, 1, 0, 1, 0, 1, 0, 1, 1, 2, 1, 1, 2, 0, 0,
0, 1, 1, 2, 2, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 2, 4, 2, 1, 3, 1, 0, 1, 1, 2, 2, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 2, 1, 1, 1, 1, 2, 1, 1, 0, 2, 1, 1, 1, 1,
0, 1, 1, 1, 1, 0, 1, 3, 3, 1, 2, 0, 3, 2, 2, 2, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 2, 0, 1, 1], [0, 2, 0, 1, 1, 0, 1, 2, 2, 1, 0, 0, 1, 1, 2, 4, 1, 3, 0, 2,
1, 3, 0, 0, 1, 4, 0, 1, 0, 3, 0, 1, 0, 0, 1, 2, 1, 2, 0, 1, 0, 0, 0, 1, 3, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 3, 2, 1, 2, 0, 2, 2, 0, 0, 0, 1, 0, 1, 0, 2, 2, 1,
1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 2, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1], [0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 2, 1, 0, 1, 0,
1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 3, 3, 1, 0, 2, 1, 2, 0, 2, 0, 1, 0, 2, 1, 1, 0, 2, 0, 2, 0, 0, 1, 1, 1, 1, 2, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 2, 1, 0, 0, 0,
3, 1, 1, 1, 1, 2, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 3, 4, 1, 0], [1, 0, 0, 1, 0, 3, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 2, 0, 0, 3, 3, 0, 1, 1,
1, 1, 2, 0, 2, 0, 4, 1, 1, 1, 1, 1, 0, 2, 2, 3, 0, 0, 1, 2, 1, 0, 0, 2, 0, 0, 1, 2, 1, 1, 0, 1, 1, 0, 0, 1, 0, 3, 1, 1, 1, 1, 0, 1, 2, 2, 1, 2, 2, 3, 3, 0,
2, 1, 0, 1, 1, 2, 1, 1, 0, 0, 0, 0, 1, 4, 1, 0, 2, 0, 0, 1], [1, 1, 1, 2, 0, 4, 2, 1, 0, 1, 3, 3, 1, 1, 0, 1, 0, 0, 1, 1, 2, 2, 1, 2, 0, 0, 0, 1, 2, 0, 1, 0,
1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 2, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 2, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 2, 0, 0, 0, 3, 0, 2, 0, 0, 1,
2, 1, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 2, 0], [0, 1, 3, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 3, 3, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 2, 0, 2, 4, 0, 2, 0, 0, 1,
0, 0, 0, 0, 1, 2, 1, 1, 0, 1, 0, 1, 0, 0, 2, 0, 1, 1, 0, 3, 0, 1, 1, 2, 1, 1, 0, 0, 1, 2, 1, 1, 1, 2, 0, 2, 2, 0, 1, 0, 0, 2, 1, 1, 0, 0, 1, 0, 2, 1, 1, 0,
1, 1, 3, 0, 2, 0, 0, 1, 0, 0, 3, 0], [0, 3, 2, 0, 2, 1, 0, 0, 1, 0, 0, 2, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 2, 3, 3,
2, 1, 4, 1, 1, 0, 1, 0, 3, 1, 3, 2, 3, 3, 2, 0, 0, 0, 1, 2, 2, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 3, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 2, 1, 1, 0, 0, 1, 3,
1, 0, 0, 1, 1, 1, 1, 0], [0, 0, 0, 0, 1, 0, 1, 1, 2, 3, 2, 0, 0, 4, 0, 2, 0, 0, 0, 1, 1, 2, 1, 0, 2, 1, 2, 0, 2, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 2, 1, 2, 1, 1,
1, 1, 2, 0, 0, 0, 1, 0, 1, 0, 0, 2, 2, 1, 0, 1, 3, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 2, 1, 0, 1, 0, 1, 2, 2, 0, 2, 3, 1, 2, 2, 1, 0, 0, 1, 1, 1, 2, 0, 1, 1,
1, 0, 1, 1]]

---

k = 4

average user error:
0.8756

user average errors
[0.95999999999999996, 0.98999999999999999, 0.91000000000000003, 0.81000000000000005, 0.77000000000000002,
0.78000000000000003, 0.68999999999999995, 0.71999999999999997, 0.93000000000000005, 0.83999999999999997,
0.98999999999999999, 0.83999999999999997, 0.98999999999999999, 0.93999999999999995, 0.82999999999999996,
0.82999999999999996, 0.66000000000000003, 0.93000000000000005, 0.89000000000000001, 0.98999999999999999,
0.95999999999999996, 0.91000000000000003, 1.03, 0.81000000000000005, 0.89000000000000001]

user errors:
[[2, 2, 0, 1, 1, 0, 2, 1, 1, 1, 2, 2, 2, 3, 1, 0, 2, 1, 2, 1, 0, 0, 1, 1, 0, 0, 0, 0, 2, 2, 1, 0, 1, 1, 0, 1, 2, 2, 1, 0, 0, 0, 1, 0, 2, 1, 1, 2, 1, 0, 0, 1,
0, 1, 1, 1, 2, 1, 2, 0, 1, 1, 2, 4, 2, 0, 0, 0, 0, 2, 2, 1, 1, 1, 0, 1, 1, 1, 0, 1, 2, 1, 0, 1, 2, 1, 2, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 2],
[0, 2, 1, 1, 1, 1, 2, 3, 1, 0, 1, 2, 1, 1, 1, 2, 0, 1, 1, 2, 1, 0, 3, 1, 2, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 2, 1, 1, 1, 1, 0, 1, 1, 1, 3, 0, 1, 1, 1, 0, 1, 2, 0, 3,
0, 0, 1, 1, 0, 1, 0, 0, 1, 2, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 2, 1, 4, 1, 1, 0, 0, 1, 2, 1, 1, 0, 1, 2, 0, 1, 1, 3, 1],
[0, 2, 0, 2, 1, 0, 1, 0, 2, 0, 0, 2, 3, 0, 1, 1, 1, 0, 1, 2, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 2, 1, 0, 1, 2, 2, 1, 1, 0, 2, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 2, 2, 2, 0, 3, 2,
0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 2, 1, 1, 1, 1, 0, 1, 0, 2, 3, 2, 2, 2, 1, 0, 1, 1, 1, 0, 0, 2, 0, 1, 0, 1, 3, 2, 0],
[1, 1, 1, 1, 2, 1, 3, 0, 1, 1, 1, 0, 1, 2, 1, 2, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 2, 1, 1, 1, 1, 0, 0, 2, 2, 1, 0, 2, 1, 1, 1, 1, 0, 0, 0, 3, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
1, 0, 1, 0, 2, 3, 1, 1, 2, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 2, 1, 2, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1],
[1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 3, 2, 1, 1, 0, 1, 0, 0, 1, 0, 2, 1, 1, 1, 0, 0, 1, 1, 2, 0, 0, 1, 1, 1, 0, 1, 0, 2, 1, 0, 0, 1, 1, 1, 1, 2, 0, 1, 0, 1, 2, 1, 2, 1, 1, 0, 1, 1, 1, 1, 1, 0,
0, 2, 2, 0, 1, 2, 1, 1, 0, 3, 0, 0, 0, 2, 2, 0, 1, 1, 2, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0],
[0, 1, 1, 1, 2, 1, 1, 0, 0, 0, 1, 1, 0, 2, 0, 1, 1, 1, 1, 2, 2, 1, 1, 0, 2, 0, 1, 2, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 4, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0,
0, 0, 1, 0, 2, 0, 1, 1, 2, 0, 0, 0, 1, 4, 0, 2, 1, 1, 0, 2, 1, 1, 0, 1, 1, 0, 1, 2],
[1, 2, 0, 1, 0, 0, 2, 0, 0, 0, 0, 1, 2, 2, 0, 0, 1, 1, 3, 1, 0, 0, 2, 0,
1, 1, 1, 0, 0, 1, 0, 2, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 2, 1, 1, 1, 0, 1, 0, 1, 2, 1, 2, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 2, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1],
[0, 1, 1, 0, 1, 2, 3, 1, 1, 0, 1, 1, 2, 1, 1, 2, 0, 0, 1, 1, 1, 3, 0, 1, 0, 0, 0, 1,
2, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 2, 2, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0, 1, 2, 0,
1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 3],
[2, 0, 0, 0, 2, 1, 0, 0, 2, 3, 1, 0, 1, 2, 3, 0, 0, 1, 1, 0, 1, 1, 0, 0, 2, 0, 0, 2, 0, 0, 1, 2,
1, 0, 1, 0, 2, 2, 0, 2, 0, 2, 2, 1, 1, 1, 1, 1, 0, 0, 2, 0, 1, 1, 2, 1, 0, 1, 0, 1, 0, 0, 2, 1, 2, 0, 1, 4, 1, 1, 2, 1, 2, 2, 1, 1, 0, 2, 2, 1, 0, 1, 0, 0,
1, 1, 3, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2, 0, 0, 2],
[1, 2, 1, 0, 1, 3, 1, 1, 1, 1, 2, 1, 0, 1, 0, 2, 0, 1, 2, 0, 1, 0, 1, 1, 2, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0, 1, 1, 1, 0, 2, 1, 2, 1, 0, 2, 1, 0, 0, 1, 2, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 2, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 0, 0, 0, 1, 1,
0, 1, 1, 0, 2, 1, 1, 4, 1, 1, 2, 2],
[0, 0, 0, 2, 2, 3, 1, 1, 1, 1, 1, 1, 0, 2, 1, 3, 2, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0,
0, 1, 0, 1, 1, 2, 0, 3, 1, 1, 2, 0, 4, 2, 1, 1, 2, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 2, 1, 1, 1, 0, 0, 2, 2, 2, 1, 2, 2, 2, 1, 1, 0, 1, 1, 1,
2, 0, 1, 2, 2, 0, 1, 1],
[1, 2, 1, 0, 1, 1, 1, 2, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 3, 1, 1, 0, 0, 1, 1, 1, 1, 1, 2, 0, 1, 1, 2, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 2,
1, 1, 1, 1, 1, 0, 1, 0, 1, 2, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 2, 0, 1, 0, 0, 1, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0, 2, 1, 2, 0, 1, 0, 2, 1, 1, 0, 0, 1,
1, 0, 0, 1],
[0, 1, 0, 0, 0, 0, 0, 3, 0, 2, 2, 2, 0, 1, 2, 0, 1, 2, 2, 1, 0, 0, 0, 1, 1, 2, 1, 1, 0, 3, 1, 1, 0, 1, 0, 2, 2, 1, 3, 1, 1, 1, 0, 0, 2, 0, 1, 1,
1, 2, 1, 2, 0, 4, 0, 1, 0, 1, 0, 1, 1, 3, 2, 1, 1, 1, 2, 1, 2, 0, 3, 0, 0, 1, 0, 1, 1, 1, 0, 2, 3, 0, 0, 1, 0, 0, 3, 1, 0, 0, 0, 1, 1, 0, 2, 0, 1, 0, 4, 0],
[1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 2, 0, 0, 3, 0, 1, 2, 1, 3, 0, 1, 0, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 1, 1, 1, 0, 0, 2, 1, 3, 1,
0, 1, 0, 0, 2, 0, 0, 0, 0, 2, 1, 0, 0, 1, 1, 2, 0, 2, 1, 0, 2, 2, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 3, 2, 0, 2, 1, 1, 1, 1],
[1, 0, 0, 0, 1, 2, 1, 0, 0, 1, 1, 1, 1, 0, 3, 1, 1, 0, 3, 0, 3, 2, 1, 0, 0, 2, 0, 0, 1, 0, 2, 1, 0, 1, 1, 1, 2, 0, 0, 0, 1, 1, 0, 0, 0, 3, 0, 1, 0, 1, 1, 1, 0, 1, 3, 1,
1, 0, 1, 0, 3, 1, 1, 1, 2, 0, 0, 0, 0, 1, 1, 3, 1, 0, 0, 0, 1, 1, 0, 3, 0, 0, 0, 0, 0, 3, 0, 0, 2, 3, 0, 1, 1, 1, 0, 0, 0, 0, 3, 0],
[0, 0, 0, 0, 1, 0, 1, 0,
2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 1, 0, 1, 0, 3, 1, 1, 1, 2, 1, 1, 0, 1, 3, 1, 3, 0, 0, 0, 0, 1, 1, 1, 1,
2, 0, 0, 1, 2, 0, 1, 3, 1, 1, 0, 2, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 4, 0, 1, 1, 0, 0, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 0],
[0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0,
1, 0, 0, 2, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 0, 1, 0, 0, 1, 0, 3, 1, 0, 0, 0, 1, 0, 0, 1, 1, 2, 1, 1, 0, 2, 1, 0, 2, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,
2, 1, 0, 1, 2, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 3, 1, 1, 1, 0, 2, 1, 1, 1, 1, 0, 0, 2, 1, 0, 3, 0, 0],
[0, 0, 1, 1, 0, 1, 2, 2, 2, 1, 3, 0, 1, 1, 0, 1,
1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 0, 2, 0, 0, 0, 1, 1, 2, 0, 0, 2, 1, 0, 1, 0, 2, 0, 1, 0, 2, 2, 1, 1, 2, 2, 1, 0, 0, 1, 1, 0, 0, 3, 1, 0, 2, 1, 1, 0, 1, 0,
0, 2, 0, 1, 0, 4, 1, 1, 1, 2, 2, 0, 1, 0, 1, 2, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0],
[1, 0, 1, 0, 1, 1, 1, 0, 2, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1,
2, 0, 1, 1, 1, 0, 3, 1, 0, 2, 1, 0, 0, 2, 1, 3, 3, 1, 1, 3, 1, 1, 0, 0, 1, 1, 3, 1, 1, 2, 2, 0, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 2, 1, 1, 1, 1, 1, 1, 2, 0, 1,
0, 1, 1, 1, 1, 0, 2, 0, 0, 0, 1, 2, 0, 1, 0, 2, 1, 0, 1, 0, 0, 1, 1, 0, 0, 2, 1, 2],
[1, 0, 1, 1, 0, 3, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1,
1, 1, 0, 0, 1, 0, 1, 0, 3, 0, 1, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 1, 0, 1, 3, 4, 0, 1, 2, 0, 2, 1, 1, 1, 0, 0, 1, 1, 2, 1, 1, 0, 1, 2, 1, 0, 1, 0, 2, 1, 1, 1,
0, 2, 0, 1, 1, 0, 1, 3, 1, 1, 1, 1, 2, 0, 1, 1, 1, 1, 3, 0, 3, 3, 2, 1],
[1, 0, 1, 1, 2, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 4, 1, 0, 1, 1, 2, 1, 1, 0, 1, 1, 0, 1,
4, 0, 3, 1, 0, 1, 1, 2, 2, 1, 2, 0, 3, 1, 0, 1, 1, 1, 0, 0, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1, 0, 0, 2, 0, 1, 1, 0, 2, 0, 0, 0, 1, 2, 0, 1, 1, 1, 2, 3, 1, 0,
0, 1, 1, 0, 1, 0, 1, 1, 3, 1, 0, 1, 0, 1, 1, 0, 2, 0, 2, 0],
[0, 1, 1, 1, 0, 1, 3, 1, 1, 2, 2, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 0,
0, 2, 1, 2, 0, 0, 2, 2, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 2, 2, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 3, 1, 0, 1, 1, 1, 2, 1, 0, 0, 0, 3,

1, 1, 1, 0, 1, 1, 1, 1, 1, 3, 2, 0, 0, 2, 1, 2], [1, 3, 0, 1, 1, 1, 0, 2, 1, 1, 0, 0, 1, 0, 1, 2, 0, 0, 2, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 0, 1, 2, 2, 2, 1, 3, 1, 0, 0, 3, 1, 0, 0, 2, 2, 1, 1, 1, 2, 0, 1, 0, 2, 0, 0, 2, 2, 0, 2, 1, 3, 1, 2, 1, 0, 1, 2, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 1, 2, 2, 0, 1], [4, 0, 0, 3, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 2, 0, 0, 0, 1, 2, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 1, 1, 1, 0, 1, 1, 2, 1, 3, 2, 1, 0, 0, 0, 1, 1, 0, 2, 3, 1, 0, 1, 1, 1, 1, 1, 2, 1, 0, 0, 0, 1, 2, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 3, 0, 1, 0, 1, 1, 2, 1, 3, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1], [0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 2, 1, 1, 2, 0, 1, 0, 3, 1, 1, 0, 0, 0, 2, 0, 1, 1, 1, 1, 0, 1, 1, 0, 2, 1, 1, 1, 1, 0, 0, 0, 3, 1, 1, 0, 2, 3, 2, 0, 1, 1, 1, 1, 1, 1, 0, 3, 1, 1, 2, 1, 0, 1, 0, 0, 1, 2, 2, 4, 0, 0, 0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 2, 2, 0, 2, 0, 0, 0, 1, 3, 0]]

---

k = 8

user average error:

0.9112

user error averages:
[1.01, 0.89000000000000001, 0.97999999999999998, 0.89000000000000001, 0.83999999999999997, 0.81000000000000005, 1.0, 0.82999999999999996, 0.93000000000000005, 0.80000000000000004, 0.89000000000000001, 1.0900000000000001, 0.91000000000000003, 0.79000000000000004, 1.0, 0.85999999999999999, 0.91000000000000003, 0.88, 0.84999999999999998, 0.97999999999999998, 0.81000000000000005, 0.93999999999999995, 0.92000000000000004, 0.98999999999999999, 0.97999999999999998]

user errors:
[[0, 1, 1, 0, 1, 0, 1, 2, 1, 4, 1, 2, 1, 2, 0, 2, 1, 0, 2, 1, 2, 2, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 3, 3, 3, 0, 1, 0, 3, 1, 1, 0, 1, 1, 1, 2, 0, 1, 1, 0, 2, 0, 2, 0, 1, 3, 0, 1, 1, 0, 1, 0, 0, 4, 0, 1, 1, 0, 2, 0, 0, 1, 0, 1, 1, 1, 1, 2, 0, 0, 1, 1, 0, 2, 4, 1, 1, 0, 0, 2, 1, 1, 2, 0, 1, 1, 0, 1], [0, 1, 0, 1, 3, 0, 0, 1, 2, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 2, 1, 1, 3, 1, 0, 1, 0, 1, 1, 0, 1, 0, 2, 0, 2, 1, 2, 1, 1, 1, 0, 1, 2, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 2, 2, 1, 1, 2, 1, 0, 2, 3, 1, 1, 1, 0, 1, 1, 0, 1, 2, 1, 1, 1, 2, 1, 3, 1, 1, 0, 1, 3, 0, 1, 0, 1, 1, 0, 0], [2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 3, 3, 1, 1, 1, 0, 0, 1, 0, 1, 1, 2, 0, 2, 0, 1, 0, 0, 0, 3, 1, 2, 0, 2, 1, 1, 0, 0, 0, 2, 1, 0, 2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 3, 1, 0, 1, 1, 2, 2, 1, 0, 1, 2, 1, 1, 1, 2, 0, 3, 2, 3, 1, 0, 0, 1, 1, 1, 0, 2, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 2, 3, 1, 3, 1, 1], [1, 0, 2, 1, 2, 2, 2, 2, 0, 1, 1, 0, 1, 0, 1, 1, 2, 1, 1, 1, 1, 0, 0, 1, 3, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 2, 2, 1, 1, 3, 0, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0, 2, 1, 0, 1, 0, 0, 1, 0, 2, 1, 1, 1, 1, 2, 1, 1, 0, 1, 2, 0, 0, 0, 2, 1, 0, 0, 1, 0, 0, 1, 2, 3, 3, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1], [0, 0, 1, 0, 1, 2, 0, 0, 2, 1, 0, 0, 1, 0, 1, 2, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 2, 0, 2, 0, 1, 1, 0, 1, 1, 0, 0, 1, 3, 3, 2, 0, 0, 0, 1, 3, 0, 0, 0, 3, 1, 1, 0, 0, 2, 0, 1, 2, 1, 1, 0, 1, 0, 2, 2, 1, 0, 1, 1, 0, 2, 0, 0, 0, 2, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 2, 0, 1, 1, 1, 2, 2, 0, 2, 1, 1, 2, 1, 0, 2], [2, 0, 1, 2, 2, 2, 0, 1, 1, 0, 1, 0, 0, 3, 0, 0, 0, 1, 0, 1, 2, 2, 0, 0, 2, 0, 1, 0, 1, 1, 0, 0, 1, 1, 2, 2, 1, 0, 0, 0, 1, 0, 2, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 2, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 2, 3, 1, 0, 0, 1, 1, 1, 0, 1, 2, 1, 0, 0, 2, 1, 1, 0, 0, 3], [1, 1, 0, 2, 2, 0, 2, 1, 1, 2, 0, 2, 3, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 0, 3, 2, 1, 0, 0, 1, 0, 2, 2, 0, 3, 0, 1, 1, 0, 1, 3, 1, 3, 0, 0, 1, 2, 1, 3, 2, 1, 1, 0, 1, 1, 3, 1, 2, 2, 1, 0, 0, 2, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 2], [0, 3, 1, 1, 1, 1, 1, 1, 1, 1, 4, 0, 2, 0, 0, 0, 2, 1, 1, 0, 0, 2, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 2, 0, 0, 1, 1, 0, 0, 0, 2, 1, 0, 1, 0, 0, 0, 1, 0, 3, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 2, 1, 2, 1, 0, 0, 0, 1, 1, 1, 1, 0, 2, 1, 1, 1, 1, 0, 1, 1, 2, 1, 1, 0, 1, 1, 3, 0, 1, 1, 4], [1, 1, 2, 1, 2, 0, 0, 1, 0, 3, 2, 0, 1, 1, 0, 0, 0, 1, 2, 3, 1, 0, 1, 0, 1, 1, 1, 1, 4, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 2, 0, 0, 0, 1, 1, 0, 1, 1, 0, 2, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 2, 1, 1, 2, 1, 1, 3, 1, 1, 1, 1, 3, 0, 1, 2, 0, 2, 0, 1, 0, 1, 1, 2, 1, 2, 3, 1, 1, 1, 2, 0, 0, 2, 0], [0, 1, 1, 1, 0, 2, 1, 0, 1, 1, 1, 2, 1, 0, 1, 1, 0, 1, 1, 3, 2, 0, 1, 1, 1, 0, 1, 1, 2, 0, 2, 1, 0, 1, 0, 2, 1, 2, 1, 0, 0, 1, 2, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 2, 1, 2, 0, 1, 2, 0, 1, 1, 0, 1, 1, 1, 1, 3, 1, 0, 0, 1, 0, 0, 1, 0], [0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 2, 1, 1, 1, 0, 2, 1, 0, 2, 1, 2, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 2, 2, 0, 2, 0, 1, 1, 3, 1, 0, 1, 1, 0, 2, 1, 1, 2, 0, 1, 1, 0, 3, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 3, 1, 1, 1, 1, 2, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 3, 1], [0, 3, 1, 1, 0, 2, 1, 1, 3, 1, 2, 1, 0, 1, 0, 0, 3, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1, 2, 0, 1, 1, 3, 1, 2, 1, 1, 2, 2, 1, 0, 1, 1, 1, 1, 2, 1, 3, 0, 3, 2, 1, 1, 1, 2, 0, 1, 2, 1, 0, 0, 0, 0, 1, 0, 1, 2, 0, 1, 0, 1, 3, 2, 1, 2, 1, 1, 0, 1, 1, 3, 1, 0, 3, 0, 2, 0, 0, 0, 1, 0, 1, 2, 1, 2, 0], [0, 3, 0, 2, 1, 1, 1, 1, 0, 0, 3, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 2, 1, 0, 1, 3, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 3, 1, 1, 0, 1, 1, 2, 3, 0, 0, 1, 0, 3, 1, 1, 1, 4, 1, 0, 0, 0, 1, 1, 2, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 2, 1, 1, 1, 0, 1, 0, 1, 1, 1, 2, 0, 1, 1, 2, 1, 0, 1, 0, 1, 0, 1, 0, 2, 0, 1, 0, 2, 1], [2, 0, 1, 3, 1, 0, 1, 1, 0, 3, 2, 1, 2, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 2, 0, 0, 1, 1, 2, 1, 1, 2, 2, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 2, 1, 1, 1, 1, 1, 0, 1, 3, 1, 1, 1, 1, 1, 1, 0, 2, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0], [1, 2, 2, 0, 4, 1, 0, 2, 1, 2, 1, 1, 2, 1, 2, 2, 0, 0, 0, 0, 1, 1, 2, 0, 0, 1, 2, 0, 1, 2, 1, 0, 0, 1, 1, 0, 2, 0, 2, 1, 1, 1, 3, 1, 2, 2, 0, 1, 0, 1, 1, 2, 1, 0, 0, 1,

2, 0, 1, 1, 1, 2, 1, 0, 3, 1, 2, 1, 1, 1, 0, 0, 0, 1, 1, 0, 2, 2, 0, 0, 1, 3, 1, 1, 0, 1, 0, 1, 0, 1, 2, 1, 1, 0, 2, 1, 1, 0, 1, 0], [0, 1, 0, 0, 0, 2, 1, 1, 2, 1, 2, 4, 0, 0, 0, 3, 0, 1, 0, 1, 0, 0, 1, 3, 2, 0, 1, 1, 0, 0, 3, 1, 1, 0, 1, 1, 1, 0, 1, 0, 3, 2, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 2, 0, 1, 0, 0, 2, 0, 1, 2, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 2, 0, 0, 2, 2, 1, 0, 1, 0, 0, 1, 1, 2, 0, 1, 0, 1, 2, 1, 1], [0, 2, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 3, 1, 2, 1, 2, 1, 2, 2, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 3, 1, 1, 3, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 3, 1, 0, 3, 1, 0, 1, 3, 2, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 2, 1, 1, 1, 1, 0, 0, 1, 0, 0, 2, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 2, 1, 0, 1, 1, 0], [0, 1, 0, 0, 0, 1, 0, 2, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 3, 2, 1, 1, 1, 0, 1, 2, 0, 0, 1, 3, 1, 1, 1, 1, 2, 1, 0, 0, 0, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 3, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 3, 0, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 0, 1, 2, 1, 1, 1, 0, 1, 0, 1, 0, 2, 1, 1, 0, 1, 0, 1], [1, 0, 1, 2, 0, 3, 2, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 2, 1, 1, 0, 2, 1, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 2, 0, 1, 3, 2, 1, 3, 0, 1, 1, 1, 0, 0, 0, 1, 2, 1, 1, 1, 0, 2, 1, 0, 0, 3, 2, 2, 1, 2, 0, 1, 1, 0, 2, 0, 1, 1, 4, 1, 0, 2, 2, 1, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 1], [2, 3, 1, 0, 0, 0, 1, 0, 1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 0, 1, 2, 2, 1, 1, 0, 1, 1, 3, 2, 0, 0, 1, 0, 1, 1, 2, 1, 2, 1, 1, 3, 1, 1, 1, 2, 1, 4, 1, 2, 3, 0, 0, 1, 1, 0, 0, 1, 0, 0, 2, 1, 2, 0, 0, 1, 2, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1], [1, 1, 2, 0, 0, 2, 2, 1, 1, 1, 2, 1, 0, 1, 1, 0, 1, 2, 0, 0, 1, 3, 3, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 2, 1, 1, 0, 1, 1, 1, 0, 1, 2, 2, 0, 0, 1, 1, 1, 2, 1, 0, 1, 3, 2, 2, 1, 0, 3, 0, 3, 1, 0, 0, 3, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 3, 0, 1, 0], [0, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 2, 0, 2, 0, 2, 0, 0, 0, 0, 1, 2, 0, 1, 1, 1, 0, 1, 2, 0, 0, 1, 1, 1, 1, 3, 1, 2, 0, 1, 0, 4, 2, 0, 2, 1, 1, 0, 2, 1, 1, 3, 0, 0, 1, 1, 1, 1, 1, 2, 0, 1, 1, 2, 0, 0, 0, 0, 4, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 3, 1, 1, 0, 1, 0, 1, 2, 1, 3], [1, 1, 1, 1, 2, 0, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 0, 3, 0, 0, 3, 1, 1, 1, 2, 0, 1, 1, 2, 1, 2, 0, 1, 1, 1, 1, 1, 0, 1, 2, 1, 0, 1, 0, 0, 2, 0, 3, 1, 1, 1, 3, 1, 0, 0, 2, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 2, 2, 2, 1, 1, 1, 1, 0, 1, 1, 2, 1, 1, 2, 0, 2, 0, 0, 1, 1, 1, 0, 0, 0, 2], [1, 1, 2, 2, 4, 3, 2, 1, 0, 1, 0, 1, 0, 2, 1, 0, 3, 4, 2, 1, 2, 2, 0, 1, 1, 0, 1, 1, 0, 2, 1, 0, 1, 0, 1, 0, 1, 1, 3, 1, 1, 0, 0, 1, 1, 3, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 2, 2, 1, 1, 0, 1, 1, 1, 1, 1, 2, 1, 1, 3, 3, 0, 1, 0, 0, 1, 0, 0, 1, 1, 2, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 2, 1, 0, 2, 0, 0, 0, 0]]

---

k = 16

user error average:
0.9156

user average errors:
[0.96999999999999997, 0.78000000000000003, 0.81000000000000005, 1.01, 0.92000000000000004, 0.93000000000000005, 0.89000000000000001, 0.97999999999999998, 0.84999999999999998, 0.97999999999999998, 0.91000000000000003, 1.01, 0.92000000000000004, 0.89000000000000001, 0.87, 0.92000000000000004, 0.97999999999999998, 0.90000000000000002, 0.83999999999999997, 0.98999999999999999, 0.73999999999999999, 1.05, 0.91000000000000003, 1.0, 0.83999999999999997]

user errors:

[[1, 0, 1, 1, 0, 0, 0, 2, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 3, 0, 0, 1, 0, 2, 1, 1, 0, 0, 2, 0, 1, 1, 2, 1, 0, 0, 2, 1, 1, 3, 1, 2, 1, 1, 3, 0, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 0, 1, 2, 1, 1, 3, 1, 1, 0, 1, 2, 2, 1, 0, 1, 1, 0, 0, 2, 2, 1, 2, 0, 2, 1, 1, 2, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 2], [0, 0, 0, 3, 1, 0, 3, 0, 0, 0, 2, 2, 1, 1, 0, 1, 0, 1, 1, 2, 0, 1, 1, 1, 3, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 3, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 2, 3, 0, 1, 0, 0, 1, 2, 1, 2, 1, 1, 1, 2, 0, 2, 0, 1, 3, 1, 0, 0, 0, 3, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1], [1, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0, 3, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 2, 1, 2, 0, 2, 0, 1, 0, 1, 2, 0, 3, 1, 2, 1, 1, 1, 0, 0, 1, 2, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 3, 1, 0, 1, 2, 1, 1, 1, 0, 1, 1, 3, 3, 0, 2, 0, 0, 1, 0], [1, 0, 1, 1, 1, 1, 2, 1, 0, 0, 1, 0, 2, 1, 0, 2, 2, 0, 3, 1, 0, 0, 1, 2, 1, 1, 0, 0, 1, 1, 1, 1, 3, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 0, 0, 1, 1, 2, 0, 1, 1, 1, 0, 1, 1, 3, 1, 1, 1, 1, 1, 1, 0, 0, 2, 1, 0, 1, 0, 2, 1, 0, 1, 2, 3, 0, 3, 1, 1, 1, 1, 2, 1, 1, 2, 1, 0, 2], [1, 2, 1, 3, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 2, 0, 1, 0, 1, 0, 2, 1, 1, 0, 1, 1, 1, 0, 3, 0, 0, 3, 1, 0, 1, 3, 1, 1, 1, 0, 2, 1, 1, 2, 1, 1, 1, 1, 2, 0, 1, 1, 1, 1, 0, 2, 0, 1, 1, 1, 1, 1, 3, 1, 0, 1, 3, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 2, 1, 2, 1, 0, 3, 1], [0, 2, 0, 1, 1, 2, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 3, 3, 0, 1, 2, 0, 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 3, 1, 0, 2, 1, 0, 3, 1, 0, 1, 1, 2, 0, 1, 1, 1, 2, 1, 0, 1, 1, 1, 0, 0, 0, 0, 3, 0, 1, 1, 3, 0, 0, 0, 1, 1, 3, 0, 0, 0, 1, 0, 3, 0, 1, 1, 2, 0, 1, 3, 0, 1, 3, 0, 0, 1, 0, 0, 0, 2, 2], [0, 0, 3, 2, 1, 0, 0, 0, 0, 1, 1, 2, 0, 1, 0, 3, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 2, 1, 1, 2, 1, 0, 0, 1, 2, 1, 0, 0, 3, 0, 1, 2, 2, 0, 1, 0, 2, 2, 0, 1, 0, 1, 1, 1, 0, 0, 1, 3, 1, 2, 1, 0, 1, 1, 0, 3, 1, 1, 1, 0, 1, 1, 2, 0, 2, 0, 1, 1, 1, 3, 1, 1, 1, 0, 2, 2, 1, 1, 0, 0, 0, 0, 1, 1], [1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1, 2, 1, 0, 1, 1, 0, 0, 1, 1, 0, 2, 3, 0, 1, 1, 0, 0, 1, 3, 1, 3, 1, 0, 0, 1, 0, 2, 1, 1, 0, 1, 2, 0, 1, 1, 1, 2, 3, 3, 0, 1, 1, 1, 3, 0, 1, 1, 0, 1, 3, 1, 0, 1, 1, 3, 1, 1, 1, 0, 2, 1, 1, 3, 0, 1, 1, 2, 2, 1,

0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1], [0, 1, 1, 1, 0, 1, 1, 3, 1, 1, 1, 2, 1, 0, 0, 2, 0, 0, 2, 0, 1, 0, 1, 1, 1, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 1, 0, 1, 1, 1, 1, 1, 2, 3, 0, 0, 0, 3, 3, 0, 1, 1, 1, 1, 0, 2, 1, 1, 3, 2, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 2, 2, 2, 1, 1, 0, 0, 1, 0, 1, 0, 2, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 2], [1, 2, 1, 0, 0, 1, 1, 1, 0, 1, 2, 1, 0, 1, 1, 1, 0, 0, 1, 1, 3, 1, 2, 1, 1, 0, 1, 1, 2, 0, 2, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 2, 0, 2, 0, 1, 4, 1, 3, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 2, 1, 1, 2, 0, 1, 1, 2, 3, 1, 2, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 2, 1, 3, 0, 1, 1, 1, 0], [0, 3, 0, 1, 3, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 2, 0, 2, 1, 2, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 2, 1, 2, 1, 1, 1, 1, 1, 0, 1, 0, 0, 2, 0, 1, 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 1, 0, 0, 1, 1, 1, 3, 1, 1, 1, 1, 2, 1, 1, 2, 0, 2, 1, 1, 0, 1, 0, 2, 1, 2, 1, 1, 3, 0, 3, 1, 1, 0, 0, 0, 1, 1, 1, 1, 3, 1, 1, 1], [1, 2, 0, 2, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 2, 1, 1, 2, 3, 1, 1, 1, 1, 0, 0, 1, 2, 1, 1, 3, 1, 1, 1, 3, 3, 0, 1, 1, 0, 3, 1, 0, 0, 1, 3, 0, 0, 1, 1, 3, 2, 1, 1, 1, 0, 0, 0, 0, 3, 0, 2, 0, 2, 1, 1, 1, 1, 0, 1, 1, 0, 1, 3, 0, 3, 0, 1, 0, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 0, 1, 2, 0, 0, 0, 0, 1, 3], [1, 1, 0, 0, 1, 3, 1, 1, 0, 1, 1, 0, 1, 1, 2, 1, 1, 1, 1, 0, 0, 1, 1, 3, 0, 4, 1, 0, 2, 0, 2, 1, 1, 0, 1, 3, 1, 1, 0, 1, 2, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 3, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 2, 1, 0, 0, 2, 1, 0, 1, 0, 2, 3, 1, 1], [0, 1, 0, 0, 1, 0, 1, 2, 1, 0, 1, 1, 2, 0, 1, 1, 2, 1, 0, 0, 0, 1, 0, 2, 3, 1, 1, 1, 1, 0, 1, 2, 0, 2, 1, 0, 0, 3, 1, 1, 0, 1, 3, 0, 3, 0, 1, 2, 2, 0, 0, 1, 2, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 2, 2, 0, 1, 2, 0, 0, 0, 0, 2, 1, 2, 1, 0, 1, 1, 2, 1], [1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 2, 1, 1, 3, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 3, 1, 2, 0, 2, 3, 2, 1, 2, 0, 0, 0, 2, 1, 2, 0, 1, 0, 0, 1, 0, 2, 0, 1, 1, 3, 1, 1, 2, 0, 0, 1, 0, 0, 0, 0, 1, 2, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 2, 1, 1, 1, 1, 0, 2, 1, 3, 1, 1, 2, 1, 1, 0], [1, 2, 1, 0, 1, 0, 1, 0, 1, 2, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 3, 1, 0, 2, 3, 1, 0, 1, 1, 0, 1, 2, 1, 1, 0, 0, 2, 1, 1, 1, 1, 1, 1, 1, 3, 2, 0, 0, 0, 0, 1, 2, 4, 1, 2, 0, 1, 0, 2, 1, 0, 1, 0, 1, 0, 1, 1, 2, 0, 0, 0, 1, 1, 0, 0, 1, 2, 2, 2, 0, 2, 1, 3, 1, 0, 1, 2, 0, 0, 2, 0, 0, 1, 1, 0], [1, 2, 1, 0, 0, 3, 0, 1, 3, 0, 1, 1, 1, 1, 1, 2, 1, 1, 1, 0, 0, 0, 2, 2, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 0, 2, 2, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 2, 1, 0, 1, 3, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 3, 1, 1, 1, 2, 0, 2, 0, 0, 2, 1, 2, 0, 1, 1, 1, 2], [1, 1, 0, 1, 1, 1, 2, 2, 2, 3, 1, 3, 1, 1, 0, 1, 2, 2, 1, 0, 1, 1, 1, 1, 0, 2, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 2, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 2, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 2, 3, 1, 2, 0, 0, 1, 1, 0, 1, 1, 2, 0, 0, 1, 1, 3, 1, 0, 1, 1, 3, 1, 1, 0, 0, 1, 1], [0, 2, 1, 1, 0, 1, 0, 1, 0, 1, 2, 0, 0, 1, 0, 1, 2, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 2, 0, 1, 1, 1, 3, 2, 0, 0, 2, 2, 1, 0, 3, 0, 0, 0, 0, 1, 2, 1, 0, 0, 2, 1, 1, 3, 0, 1, 1, 1, 1, 0, 1, 2, 0, 0, 1, 0, 1, 3, 1, 0, 1, 2, 1, 1, 3, 0, 1, 1, 0, 2, 0, 0, 1, 1, 3, 1, 0, 3, 0, 1], [1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 3, 0, 3, 0, 1, 0, 0, 1, 0, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 3, 2, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 0, 3, 3, 0, 2, 1, 1, 1, 1, 2, 0, 1, 1, 0, 2, 1, 1, 0, 1, 3, 1, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 3, 1, 0, 2, 1, 2, 1, 0, 2, 1, 1, 0, 0, 0, 1, 1], [1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 2, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 2, 1, 4, 0, 1, 1, 4, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 3, 2, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 2, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 2, 1, 1, 1, 1, 1], [0, 1, 1, 1, 1, 2, 1, 0, 1, 0, 1, 0, 1, 3, 0, 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 3, 1, 3, 2, 2, 2, 1, 2, 1, 1, 3, 1, 0, 1, 2, 1, 2, 0, 1, 0, 0, 1, 2, 1, 1, 2, 1, 0, 1, 0, 2, 2, 0, 1, 0, 2, 2, 1, 1, 1, 2, 0, 1, 1, 2, 0, 0, 1, 0, 2, 0, 0, 1, 1, 1, 1, 1, 2, 0, 0, 1, 2, 3, 1, 2, 0, 1, 1, 1], [0, 0, 1, 1, 0, 1, 1, 3, 1, 1, 2, 1, 0, 2, 1, 1, 0, 0, 1, 0, 1, 1, 2, 0, 0, 3, 1, 0, 0, 0, 2, 1, 1, 0, 1, 1, 1, 0, 3, 2, 1, 0, 1, 1, 0, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 0, 2, 2, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 2, 1, 1, 1, 1, 1, 0, 1, 1, 1, 2, 0, 2, 1, 0, 0, 1, 1, 1, 1, 0, 2, 1, 1, 0, 2, 1, 1, 0, 0, 0], [2, 2, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 2, 2, 1, 0, 2, 0, 1, 0, 2, 1, 0, 0, 1, 1, 1, 0, 1, 3, 0, 1, 0, 0, 2, 1, 1, 3, 2, 2, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 2, 1, 1, 0, 0, 0, 0, 3, 1, 1, 1, 1, 1, 1, 2, 0, 2, 2, 3, 2, 0, 1, 2, 1, 3, 0, 2, 1, 0, 1, 3, 1, 3, 0, 1, 1, 1, 0, 1, 1], [1, 0, 0, 1, 1, 0, 2, 1, 0, 3, 0, 1, 2, 0, 0, 3, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 3, 1, 0, 0, 0, 1, 1, 3, 1, 0, 2, 0, 2, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 3, 0, 0, 3, 1, 3, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 3, 2, 0, 0, 0, 1, 1, 1, 2, 0, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 1, 0, 1, 0]]

----

k = 32

user error average
0.8604

user error averages

[0.87, 0.9699999999999997, 0.87, 1.04, 1.02, 0.8299999999999996, 0.9100000000000003, 0.9300000000000005, 0.8900000000000001, 0.8000000000000004, 0.8299999999999996, 0.8299999999999996, 0.9000000000000002, 0.87, 0.7800000000000003, 0.9000000000000002, 0.7700000000000002, 0.75, 0.88, 0.8100000000000005, 0.87, 0.7900000000000004, 0.8000000000000004, 0.8399999999999997, 0.7600000000000001]

[[0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 3, 0, 2, 3, 2, 0, 2, 1, 0, 1, 2, 1, 1, 0, 1, 0, 2, 0, 0, 3, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 2, 2, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 2, 0, 1, 0, 1, 1, 0, 2, 2, 1, 3, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 2, 1, 2, 1], [1, 1, 3, 0, 1, 0, 2, 0, 0, 0, 0, 3, 1, 1, 1, 3, 1, 0, 0, 0, 2, 2, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 2, 1, 0, 1, 1, 1, 1, 2, 1, 1, 1, 2, 3, 2, 1, 1, 0, 0, 2, 1, 1, 1, 1, 1, 1, 2, 1, 0, 2, 1, 1, 0, 0, 1, 1, 0, 1, 2, 0, 1, 2, 1, 1, 0, 0, 0, 1, 2, 0, 3, 1, 1, 1, 1, 1, 3, 1, 0, 3, 1, 1, 1, 1], [1, 3, 0, 1, 3, 0, 0, 1, 1, 3, 0, 1, 1, 1, 0, 0, 3, 1, 1, 1, 2, 0, 0, 0, 2, 0, 1, 1, 3, 1, 1, 2, 3, 1, 1, 2, 1, 0, 0, 1, 1, 0, 1, 0, 2, 2, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0,

2, 1, 1, 1, 0, 1, 0, 0, 1, 0, 2, 1, 3, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 2, 1, 1, 0, 1, 1], [1, 0, 1, 3, 0, 1, 1, 2, 1, 3, 1, 0,
1, 0, 1, 3, 2, 1, 1, 2, 1, 1, 1, 2, 1, 0, 1, 2, 1, 1, 0, 1, 0, 2, 1, 1, 1, 0, 3, 1, 1, 1, 1, 1, 3, 0, 0, 1, 0, 0, 1, 1, 0, 1, 3, 1, 1, 1, 3, 1, 0, 3, 1, 0,
2, 0, 0, 0, 0, 1, 0, 2, 2, 0, 1, 2, 1, 1, 0, 1, 2, 0, 2, 3, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 3, 1, 1, 1], [1, 1, 0, 2, 1, 1, 1, 0, 1, 1, 3, 3, 1, 0, 1, 1,
1, 0, 1, 0, 2, 0, 0, 2, 1, 0, 1, 1, 2, 0, 3, 1, 2, 1, 1, 0, 2, 1, 0, 2, 0, 1, 1, 1, 0, 0, 2, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 2, 2, 1, 2, 0, 3, 1, 1, 3, 2, 0,
1, 3, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 3, 0, 0, 3, 0, 1, 2, 1, 0, 0, 1], [1, 0, 2, 0, 1, 1, 0, 1, 2, 1, 0, 0, 0, 1, 1, 2, 1, 1, 1, 1,
0, 0, 1, 1, 1, 0, 1, 1, 0, 2, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 2, 0, 3, 1, 0, 3, 1, 0, 1, 2, 1, 0, 0, 1, 1, 3, 0, 0, 1, 1,
1, 0, 1, 1, 1, 0, 0, 2, 2, 1, 0, 2, 0, 1, 1, 1, 0, 1, 0, 2, 2, 2, 0, 1, 1, 1, 0, 1], [1, 0, 0, 1, 1, 3, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 3, 1, 0, 0, 1, 1, 0, 1,
2, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 3, 1, 1, 2, 0, 1, 1, 2, 3, 1, 0, 3, 1, 3, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 3, 0, 0, 1, 1, 2, 0,
1, 1, 0, 0, 1, 1, 1, 1, 2, 0, 1, 1, 1, 0, 1, 1, 1, 2, 0, 2, 0, 1, 1, 0], [1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 2, 1, 3, 1, 3, 2, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0,
1, 0, 1, 1, 1, 3, 2, 1, 0, 1, 0, 1, 0, 1, 1, 2, 2, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 2, 0, 2, 0, 1, 0, 0, 1, 2, 0, 3, 0, 2, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0,
1, 1, 1, 0, 1, 0, 3, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 3, 0, 0], [0, 0, 1, 1, 1, 1, 0, 0, 2, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 2, 1, 2, 0, 0, 3, 1, 1, 0, 1, 3, 1,
2, 1, 0, 1, 1, 2, 1, 1, 2, 0, 1, 1, 0, 1, 1, 2, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 2, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 2, 0, 1, 1, 2,
1, 0, 1, 1, 0, 1, 3, 1, 0, 1, 1, 0, 2, 0, 1, 1], [1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1,
1, 1, 3, 1, 0, 1, 0, 0, 1, 0, 0, 2, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 3, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 0, 2, 1, 2, 1,
1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0], [1, 0, 1, 3, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 3, 0, 1, 0, 2, 1, 3, 1, 2, 1, 1, 0, 1, 1, 3, 1, 0, 1, 0,
1, 1, 0, 1, 2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 2, 0, 0, 0, 1, 2, 1, 0, 1, 0, 2, 0, 2, 1, 2, 1, 2, 0, 1, 0, 0, 1, 1, 1, 1, 2, 1, 1, 0, 0, 2, 0, 1, 0, 1, 1,
0, 1, 1, 1, 1, 0, 0, 0], [1, 1, 1, 2, 0, 1, 1, 0, 2, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 1, 0, 2, 1, 1, 0, 0, 0, 1, 2, 3, 0, 3, 1, 1, 0, 0, 0, 1, 0,
0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 2, 1, 1, 2, 1, 1, 1, 0, 3, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 3, 1, 1, 3, 1, 1, 0, 0, 2, 1, 1, 0, 0, 0, 0,
2, 0, 1, 1], [0, 1, 1, 2, 1, 1, 2, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 2, 3, 1, 1, 1, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1,
0, 1, 1, 2, 1, 1, 0, 1, 0, 1, 0, 1, 2, 2, 1, 1, 0, 0, 1, 1, 0, 1, 0, 3, 3, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 3, 2, 2, 0, 1, 1, 1, 1],
[1, 1, 0, 0, 3, 2, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 3, 0, 1, 1, 1, 2, 2, 1, 0, 1, 1, 0, 1, 2, 0, 0, 0, 1, 1, 2, 1, 0, 1, 1, 1, 2, 1,
1, 0, 0, 1, 0, 3, 0, 3, 1, 0, 1, 0, 1, 2, 0, 1, 1, 1, 0, 2, 0, 0, 1, 0, 1, 2, 0, 1, 1, 0, 2, 1, 1, 3, 0, 2, 1, 0, 1, 1, 1, 0, 1, 2, 1, 0, 1, 0], [1, 1, 1, 1,
0, 0, 2, 1, 2, 1, 1, 3, 1, 0, 3, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 2, 1, 0, 1, 1, 2, 1, 1, 1, 0, 2, 1, 0, 0, 1, 0, 1, 1, 1, 2, 1, 1, 0, 0, 1, 1, 0, 1,
1, 0, 3, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 2, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0], [3, 3, 1, 1, 1, 2, 0, 0,
1, 1, 0, 0, 1, 0, 1, 2, 0, 1, 1, 0, 2, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 3, 0, 3, 1, 2, 0, 1, 1, 1, 1, 1, 1, 2, 0, 0, 0, 1, 1, 1, 0, 0, 1, 3, 1, 1, 0, 0, 1, 3,
1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 3, 0, 0, 0, 1, 0, 1, 1, 1, 1, 2, 1, 0, 1, 1, 1, 2, 0, 0, 2, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1], [1, 0, 0, 2, 1, 0, 0, 0, 0, 1, 1, 1,
1, 1, 1, 2, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 2, 1, 1, 0, 1, 3, 1, 0, 1, 3, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 2, 0, 0, 2, 2, 1, 0, 3, 1,
1, 0, 0, 1, 3, 2, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0], [0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 3, 1,
1, 1, 0, 1, 0, 1, 1, 0, 2, 2, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 2, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 3, 1, 1, 1, 0, 2, 3, 1, 1, 0, 1, 1, 1, 1, 3, 0, 1,
0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 2, 0, 0, 1, 1, 2, 1, 1, 2, 1, 2, 0, 0, 0], [0, 3, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 3, 1, 2, 1,
0, 1, 2, 0, 1, 1, 0, 1, 2, 1, 1, 0, 1, 0, 1, 1, 1, 0, 2, 2, 2, 0, 1, 1, 2, 0, 1, 1, 0, 1, 2, 2, 0, 0, 0, 0, 0, 1, 1, 0, 3, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0,
1, 1, 0, 0, 2, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 2, 3, 1, 2, 1, 0, 2, 1, 2, 1, 1, 2, 0], [0, 1, 0, 1, 2, 2, 1, 3, 0, 0, 1, 1, 0, 1, 3, 0, 2, 1, 2, 1, 0, 0, 0, 1,
0, 0, 0, 1, 0, 1, 2, 1, 1, 1, 0, 1, 3, 1, 1, 1, 1, 1, 1, 0, 1, 2, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 3, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2, 1, 1, 0,
1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 2, 1, 1, 0, 0, 0, 1, 2, 3], [2, 1, 2, 0, 0, 1, 1, 1, 1, 1, 2, 0, 1, 3, 2, 1, 1, 2, 0, 0, 1, 1, 1, 1, 0, 1, 3, 0,
0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 2, 1, 2, 0, 1, 1, 0, 3, 1, 1, 0, 0, 1, 1, 0, 1, 1, 3, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 3, 1, 1,
1, 0, 0, 2, 0, 1, 1, 0, 1, 1, 3, 0, 0, 1, 0, 0, 0, 2, 3, 1], [3, 2, 1, 4, 1, 1, 1, 3, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1, 0, 1, 1, 1, 2, 0, 0, 1, 1, 0, 0, 0, 1, 0,
0, 0, 1, 1, 1, 0, 3, 0, 0, 0, 2, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 2, 1, 3, 0, 3, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 2, 1, 0, 1, 1, 1, 0, 1, 0, 2,
0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0], [0, 0, 0, 1, 0, 3, 2, 2, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 2, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 2, 0, 3, 0, 1,
1, 1, 1, 1, 1, 3, 0, 0, 3, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 2, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 2, 0, 0, 2, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
1, 1, 1, 1, 1, 3, 0, 0, 0, 0, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 3, 2, 0, 0, 1, 1, 1, 0, 0, 2, 3, 0, 0, 2, 0,
1, 0, 1, 0, 0, 2, 0, 1, 1, 0, 1, 2, 1, 0, 1, 2, 0, 1, 2, 0, 1, 0, 1, 0, 0, 1, 1, 3, 1, 0, 2, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 2, 1, 1, 0, 2, 1, 0, 0, 0, 0,
1, 0, 2, 1, 1, 0, 1, 1], [0, 1, 1, 2, 0, 0, 0, 0, 0, 2, 2, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 2, 1, 1, 0, 0, 0, 2, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0,
1, 1, 0, 0, 2, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 2, 1, 1, 1, 1, 1, 0, 3, 1, 0, 1, 1, 1, 1, 1, 0, 0, 3, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1,
2, 2, 1, 1]]

---

run the entire thing with optimal parameters
d = 1, manhattan
i = 1
k = 2

user average error
.8812

user errors
[0.88, 0.87, 0.87, 0.93000000000000005, 0.84999999999999998, 0.76000000000000001, 0.93999999999999995, 0.92000000000000004, 0.85999999999999999, 0.89000000000000001, 0.98999999999999999, 0.81000000000000005, 0.77000000000000002, 0.66000000000000003, 0.82999999999999996, 0.76000000000000001, 1.03, 1.0600000000000001, 0.87, 0.89000000000000001, 0.81000000000000005, 0.93999999999999995, 0.89000000000000001, 0.90000000000000002, 1.05]

item average error
.846

item errors
[0.91000000000000003, 0.93000000000000005, 0.85999999999999999, 0.81000000000000005, 0.89000000000000001, 0.87, 0.87, 0.79000000000000004, 0.87, 0.87, 0.83999999999999997, 0.93999999999999995, 0.83999999999999997, 0.75, 0.81000000000000005, 0.84999999999999998, 0.93000000000000005, 0.92000000000000004, 0.90000000000000002, 0.76000000000000001, 0.78000000000000003, 0.88, 0.71999999999999997, 0.77000000000000002, 0.79000000000000004]