1.)

a.) If we assume that ratings of 1 and 2 are no rating and 3, 4, and 5 are 1:

	a	b	c	d	e	f	g	h
A	1	1		1			1	
B		1	1	1				
C				1		1	1	1

Jaccard Distance:

A to B: 1 – 2/5 = 3/5
A to C: 1 – 2/6 = 4/6
B to C: 1 – 1/6 = 5/6

e.) … subtracting from each nonblank entry the average value….:

Avg: A = 3.33, B = 2.33, C = 3

	a	b	c	d	e	f	g	h
A	4 – 3.33 = ***0.67***	5 – 3.33 = ***1.67***		5 – 3.33 ***1.67***	1 – 3.33 = ***-2.33***		3 – 3.33=- ***0.33***	2 – 3.33= *-1.33*
B		3 – 2.33= ***0.67***	4 – 2.33 = ***1.67***	3 – 2.33 = ***0.67***	1 – 2.33 = ***-1.33***	2 – 2.33 = ***-0.33***	1 – 2.33 = ***-1.33***	
C	2 – 3 = ***-1***		1 – 3 = ***-2***	3 – 3 = ***0***		4 -3 = ***1***	5 – 3 = ***2***	3 – 3 = ***0***

b.)

Clustering the items and users by their respective distance measures is a method that can be used to address the sparseness of a utility matrix.

2.)

a.)

Data is stored on the HDFS and is read by Spark from the HDFS. Storing the data on the HDFS allows for replication which is instrumental in preventing any data loss due to failure.

b.)

The NameNode in Hadoop is considered the Master and is responsible for managing resources across the servers. This should not be confused with the JobTracker, which by its name, sounds like it could own that responsibility.

c.) Implement in python, a solution that would compute streaming queries average for a specified window.

```
import sys

fd = open('mydata.txt', 'r')

sys.stdin = fd

i = 0
nums = []
sums = []
current = ""

for line in sys.stdin:

    if i < 3:
        nums.append(int(line))
        #print (nums)
        current = ""
        #sums = sums + int(nums[i])
        i = i + 1
        #print(sums)
    else:
        current = line
        nums.append(int(line))
        i = 0

#print(nums)
window = 2
length = 4

for x in range(0, len(nums), window):
        sums = nums[x: x + length]
        #print(nums[x: x + 4])
        #print(sum(nums[x: x + 4]) / 4)
        if len(sums) == 4:
            print(sum(sums)/4)
```

3.)
a.)

To create numeric text file:
- from command line, install NUMPY, “sudo yum install numpy”

Python code:

```
import random
import numpy as np

a = np.random.randint(1000, size=(350000,8) )

print(a)

np.savetxt('mahout_random.txt', a, delimiter=' ', fmt='%i')
```

Verifying that the ‘testdata’ file is available:

```
[ec2-user@ip-172-31-11-74 ~]$ hadoop fs -ls /testdata
-rw-r--r--   2 ec2-user supergroup    8119270 2022-06-04 18:58 /testdata
[ec2-user@ip-172-31-11-74 ~]$
```

Running KMeans with the following command-line:

```
[ec2-user@ip-172-31-11-74 ~]$
[ec2-user@ip-172-31-11-74 ~]$ $MAHOUT_HOME/bin/mahout org.apache.mahout.clustering.syntheticcontrol.kmeans.J
ob -k 7 --t1 1 --t2 1000 --maxIter 10 --input /testdata --output /output
```

```
[ec2-user@ip-172-31-11-74 ~]$ hadoop fs -ls /output
Found 15 items
-rw-r--r--   2 ec2-user supergroup        194 2022-06-04 20:17 /output/_policy
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:18 /output/clusteredPoints
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:12 /output/clusters-0
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:13 /output/clusters-1
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:17 /output/clusters-10-final
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:13 /output/clusters-2
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:14 /output/clusters-3
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:14 /output/clusters-4
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:15 /output/clusters-5
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:15 /output/clusters-6
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:16 /output/clusters-7
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:16 /output/clusters-8
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:17 /output/clusters-9
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:12 /output/data
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:12 /output/random-seeds
[ec2-user@ip-172-31-11-74 ~]$
```

I wasn’t sure which file would be considered the last page, so I did a “cat” on the “clusteredPoints” file and it returned a series of the following text, after 5 minutes I closed the session window:

```
TTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuT
TYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTYP
uTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuT
YPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPu
TTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuT
TYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTYP
uTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuT
YPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPu
TTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
```

b.)

```
[ec2-user@ip-172-31-11-74 ~]$ hadoop fs -ls ml_dataset
Found 2 items
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:41 ml_dataset/probeSet
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:41 ml_dataset/trainingSet
[ec2-user@ip-172-31-11-74 ~]$
[ec2-user@ip-172-31-11-74 ~]$ hadoop fs -ls ml_dataset/probeSet
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2022-06-04 20:41 ml_dataset/probeSet/_SUCCESS
-rw-r--r--   2 ec2-user supergroup    1386169 2022-06-04 20:41 ml_dataset/probeSet/part-m-00000
[ec2-user@ip-172-31-11-74 ~]$ hadoop fs -ls ml_dataset/trainingSet
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2022-06-04 20:41 ml_dataset/trainingSet/_SUCCESS
-rw-r--r--   2 ec2-user supergroup   10167287 2022-06-04 20:41 ml_dataset/trainingSet/part-m-00000
[ec2-user@ip-172-31-11-74 ~]$
```

```
-rw-rw-r-- 1 ec2-user ec2-user 11553456 Jun  4 20:38 ratings.csv
[ec2-user@ip-172-31-11-74 ml-1m]$
```

Yes, the size of both files does add up to the original size.

RMSE: 0.8852244862891442

```
[ec2-user@ip-172-31-11-74 ml-1m]$ hadoop fs -ls
Found 8 items
drwxr-xr-x   - ec2-user supergroup          0 2022-05-21 21:43 KCount
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:56 als
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:40 dataset
drwxr-xr-x   - ec2-user supergroup          0 2022-05-21 21:54 lineordSUM
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:41 ml_dataset
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:38 movielens
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:07 output
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 18:58 testdata
[ec2-user@ip-172-31-11-74 ml-1m]$ hadoop fs -ls als
Found 3 items
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:53 als/out
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:56 als/rmse
drwxr-xr-x   - ec2-user supergroup          0 2022-06-04 20:56 als/tmp
[ec2-user@ip-172-31-11-74 ml-1m]$ hadoop fs -ls als/rmse
Found 1 items
-rw-r--r--   2 ec2-user supergroup         18 2022-06-04 20:56 als/rmse/rmse.txt
[ec2-user@ip-172-31-11-74 ml-1m]$ hadoop fs -cat als/rmse
cat: `als/rmse': Is a directory
[ec2-user@ip-172-31-11-74 ml-1m]$ hadoop fs -cat als/rmse/rmse.txt
0.8852244862891442[ec2-user@ip-172-31-11-74 ml-1m]$
```

Top movies for Users 1 and 2:

```
1       [957:4.4948554,919:4.4544125,858:4.442429,953:4.412355,318:4.410435,593:4.409569]
2       [2197:4.6013355,527:4.4641595,953:4.3467116,2324:4.337412,919:4.3354454,1035:4.2985945]
```

Top movie for User 1:

```
957::Scarlet Letter, The (1926)::Drama
```

Top movie for User 2:

```
2197::Firelight (1997)::Drama
```