Assignment 3

Exercise 1

Step 1:

At first, we need transpose the data, so that the 5811 papers could be rows and 11463 words could be columns.

```
// transpose the data
val rdd = only_rows.map(x => x.split(",")).map(x => x.drop(1))
val RDD = rdd.collect.transpose.map(x => Vectors.dense(x.map(_.toDouble)))
val datardd = sparkSession.sparkContext.parallelize(RDD, 200)
```

Then input the datardd to the PCA. We can get the 2PCs.

the two corresponding eigenvalues:

```
PC1 eigenvalues:
92130.22178886808
PC2 eigenvalues:
75386.86600328992
```

The percentage of variance:

```
PC1 variance:
0.005981584696637591
PC2 variance:
0.004894516861645423
```

The first 10 entries of the 2 PCs:

PC1 first 10 entries:	PC2 first 10 entries:
-0.010935579359103595	-0.024441485321969075
0.005054189166157519	0.0019527338015650209
0.006806459676965804	0.003833500841077142
0.00504415265446711 6	-0.0012604970406167038
0.009837617937094756	9.032078414480393E-4
0.03134056267421989	-0.005524465923454024
0.0010629069191318789	0.019336169093662425
0.019163048333392628	-0.001737553900645222
0.009953891408214972	-0.0053876565910474725
0.010244854050723045	-0.004959886361586386

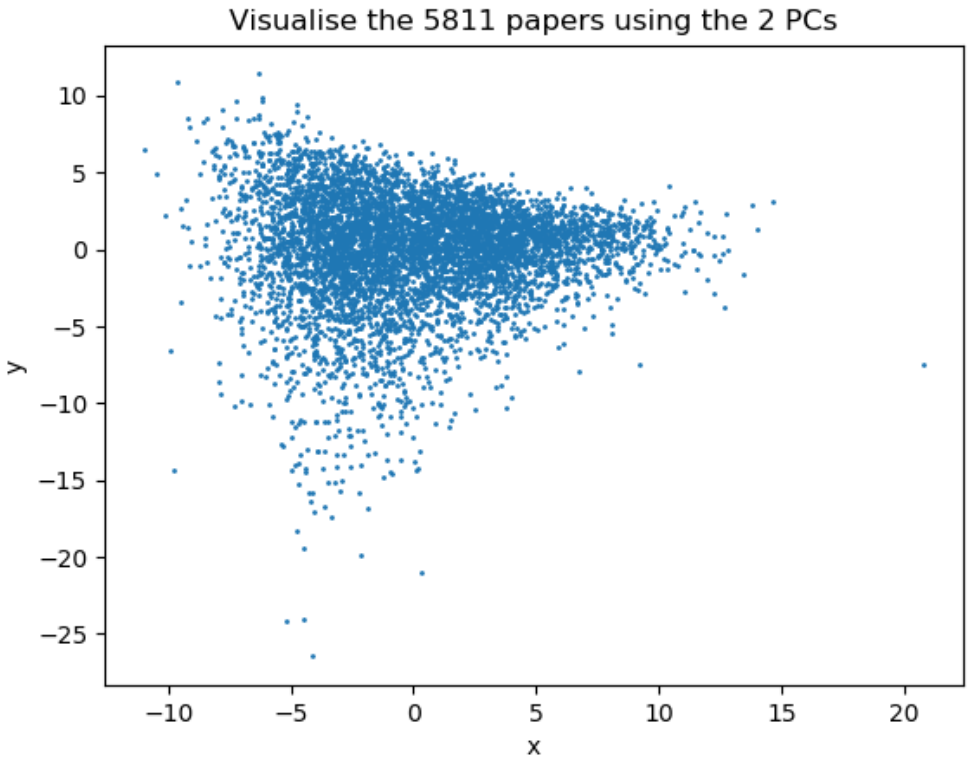
Step 2:

After PCA, the 5811 papers rows could be projected to the linear space spanned by the top 2 principal components.

We can output the result file.

```
2.5121972056944912,3.2177837336429946
7.660408067721859,-0.015889802924677367
3.503865675541362,3.157043738688064
5.663984024838001,0.7842356571375334
2.4004453587517145,1.8963099573709297
2.66610822799335,1.7636469966306147
8.926329423452222,-1.8541866922821066
3.350228465702366,2.2006041091117794
11.706320992657792,2.4514039386978794
2.500421474205293,-1.4939990455107934
3.4065484170861513,-4.761749323928325
11.942921258202507,-1.9155349356065154
2.668306350051193,-1.0269194346946964
2.1629835480966317,3.096278690550198
4.9050669238926705,-0.3859525788614016
10.939269325148336,2.231076701406025
2.306716844615908,4.846507807614131
1.6495800229955733,-0.9321380099754689
4.271372219764663,3.029408595564745
6.106930342189064,1.5058957992944528
10.420218338488668,1.0846074202482008
3.199468597506415,3.05243882814079
```

Then we can use python to visualise the 5811 papers using the 2 PCs.



Exercise 2

Step 1:

Firstly, drop the first row, the first column and the last column (label).

Use K-means to cluster all data points in this dataset with K=5.

One of the cluster is as below:


```
Cluster Centers:
[-8.110585737720738,-8.244137797934961,-8.694586509697965,-9.166747080961112,-9.54163852166361,-9.832384444658883,-10.233523111068225,-10.891826691112612,-11
.391585448229279,-11.844543085978964,-12.229470230628197,-12.595097944610634,-12.919617871272798,-13.037344398340425,-12.978867123419866,-12.559201003570395,-11
.83643732509891,-11.05046801119367,-9.993341696419956,-9.175045836147834,-8.451896169063014,-7.995754125253305,-7.750554858631671,-7.3340731448422228,-6
.355109524269034,-5.16037826884107,-4.329441281482197,-3.97597220881984,-4.061565183827077,-4.57753546270385O5,-5.108173308887388,-5.201582553314678,-5
.108269806040722,-4.859596641899064,-4.694393515391297,-4.70249927627135,-5.0909968155939405,-5.822445237865483,-6.5159702788767735,-6.995947119559974,-7
.235356556981S7,-7.494933899449967,-7.5348837209302335,-7.530155360416868S,-7.447264305702983,-7.591045064170608,-7.691788092251279,-7.811444562385A1,-7
.508636495223391,-7.165782109427773,-6.796487503618644,-6.866833928399113,-7.283798127955226,-7.908617195792725,-8.442439448036284,-8.700569333204G7,-9
.057512303387051,-9.527646434430185,-10.04245874746695,-10.709640065618064,-10.933609958506224,-10.963041590273088,-10.754221750458361,-10.596448904757S1,-10
.530251857570203,-10.594132973077295,-10.722667181318151,-10.765415420245104,-10.856991218759047,-10.044678181993632,-9.187397471774583,-8.060310720833735,-7
.514619318730098,-7.4600019299430675,-7.596352407603976,-7.939978770626267,-8.147447650294318,-8.259480845315064,-8.027212197240182,-7.6044581684840304,-7
.107208337354049,-6.746308983884976,-6.869149860079128,-7.2821576763485485,-8.231882659461547,-8.97597220881984,-9.474090514329827,-9.743510566438291,-10
.080575123033S7,-10.733378365338224,-11.414165782109428,-11.582456817523884,-11.440220013509602,-10.884203415999231,-10.233426611391489,-9.809997105085401,-9
.58457975489723,-9.749975875711668,-9.69449001254463,-9.473897520023l6,-9.039853324326932,-8.459712438483065,-8.032616037826884,-7.991411753353277,-8
.113770143780759,-8.511434912670078,-8.749010904178327,-8.718035317958l2,-8.634951268937566,-8.534497732316897,-8.511820901283413,-8.618450255717457,-8
.646241435877641,-8.638039177844254,-8.426710412042846,-8.060214223680402,-7.911608607546078,-8.00289491460002,-8.262568754221752,-8.249348644214995,-8
.131043134227541,-8.111550709254077,-8.701727299044679,-9.098330599247323,-9.352986586895687,-9.238347968734923,-8.905529286886038,-8.679146965164529,-8
.22782977902152,-7.916915950979447,-7.658689568657725,-7.580623371610538,-7.601949242497347,-7.251954067355014,-7.026054231400174,-7.167519058187784,-7
.658979060117727,-8.618450255717457,-9.749396892791664,-10.906783749879379,-11.485477178423237,-11.973559779986491,-11.996140113866641,-12.102576473994018,-12
.224838367268166,-12.319502074688797,-12.176493293447844,-12.062916143973753,-11.805751230338705,-11.290842420148607,-10.753932258998S6,-10.168966515487794,-9
.878510083952524,-9.45179967190968,-8.997973559779988,-8.458940461256393,-8.248866158448326,-8.075460773907l7,-7.937180353179582,-7.707034642478048,-7
.979639100646532,-8.944514136832964,-10.206503908134l1,-11.035703946733571,-11.323651452282158,-10.955611309466372,-10.586799189423912,-9.936311878799575,-9
.573289587957156,-9.587860658110586,-9.91286307053942,-10.119077487214128,-10.277718807295185,-10.09292675866062,-9.882562964392552,-9.709254077004l3,-9
.385699121875906,-9.162018720447747]
```


Step 2:

The largest cluster is shown as below (just part of the cluster):

```
The largest cluster is:
[-8.110585737720738,-8.244137797934961,-8.694586509
.391585448229279,-11.844543085978964,-12.2294702306
.83643732509891,-11.05046801119367,-9.9933416964199
.355109524269034,-5.16037826884107,-4.3294412814821
.108269806040722,-4.859596641899064,-4.694393515391
.23535655698157,-7.494933899449967,-7.5348837209302
.508636495223391,-7.165782109427773,-6.796487503618
.057512303387051,-9.527646434430185,-10.04245874746
.530251857570203,-10.594132973077295,-10.7226671813
.514619318730098,-7.4600019299430675,-7.59635240760
.107208337354049,-6.746308983884976,-6.869149860079
.080575123033S7,-10.733378365338224,-11.41416578210
.58457975489723,-9.749975875711668,-9.6944900125446
.113770143780759,-8.511434912670078,-8.749010904178
```

The size of the largest cluster is 10363.

```
the size of the largest cluster is 10363
```

The smallest cluster is as below (just part of the cluster):

```
The smallest cluster is:
[-71.35238095238095,-68.23333333333333,-61.99047
.9333333333333336,-1.2714285714285716,9.40952380
.03809523809525,208.14285714285717,180.728571428
.690476190476191,1.5000000000000002,9.3095238095
.51428571428573,-202.62857142857143,-201.9142857
.1857142857143,-125.69523809523811,-66.723809523
.03809523809525,273.62380952380954,287.171428571
.21428571428572,188.45714285714288,128.200000000
.66666666666667,-111.30000000000001,-118.1190476
.20000000000005,-315.8285714285715,-293.22857142
.96190476190477,-39.695238095238096,-33.79047619
.13809523809525,203.1904761904762,216.0714285714
.56666666666666,285.6380952380953,289.2238095238
.519047619047626,-116.95238095238096,-167.076190
,-181.20952380952383,-190.55238095238096,-190.97
-79.10000000000001,-47.82380952380953,-10.033333
```

The size of the smallest cluster is 210.

```
the size of the smallest cluster is 210
```

Step 3:

In this step, we get the majority label (i.e., the label with the most data points) for the largest cluster and smallest cluster.

```
the majority label for the largest cluster is 4, and the number is 2300
the majority label for the smallest cluster is 1, and the number is 205
```

Exercise 3

Step 1:

Run ./split_ratings.sh to get the five splits (r1 to r5) for five-fold cross-validation.

```
[act17xs@sharc-node004 files]$ qsub ./split_ratings.sh
```

tags.dat	3 500
split_ratings.sh.o1259469	1
split_ratings.sh.e1259469	1
split_ratings.sh	1
ratings.dat	258 892
r5.train	206 821
r5.test	52 070
r4.train	206 853
r4.test	52 038
r3.train	206 803
r3.test	52 088
r2.train	206 750
r2.test	52 142
r1.train	208 340
r1.test	50 552
movies.dat	509

Step 2:

the MSE for five splits:

```
Mean Squared Error for the first split = 0.8318781563866682
Mean Squared Error for the second split = 0.5078765107315669
Mean Squared Error for the third split = 0.35179576104471455
Mean Squared Error for the forth split = 0.60160432335776
Mean Squared Error for the fifth split = 0.26172323453884283
```

Step 3 ：

visualise movies and users, respectively, using the 2 PCs.