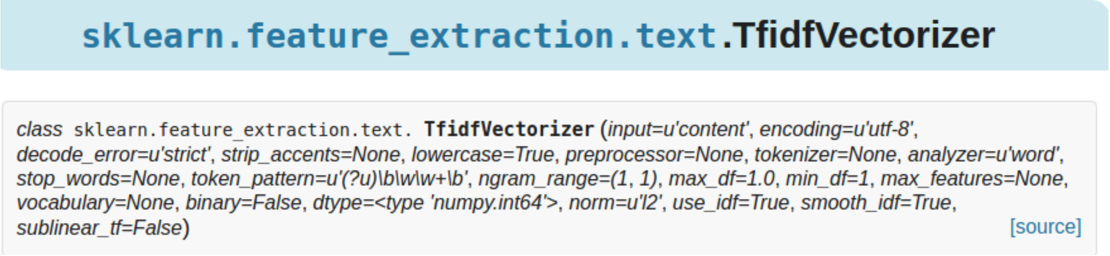
EE219 Project4 Report

Xiao Shi (404253039)
Weikun Han (804774358)

1)

In this problem, we use sklearn model to do TF-IDF. The “default” model information is the following and we need change some values from blow.



sklearn.feature_extraction.text.TfidfVectorizer

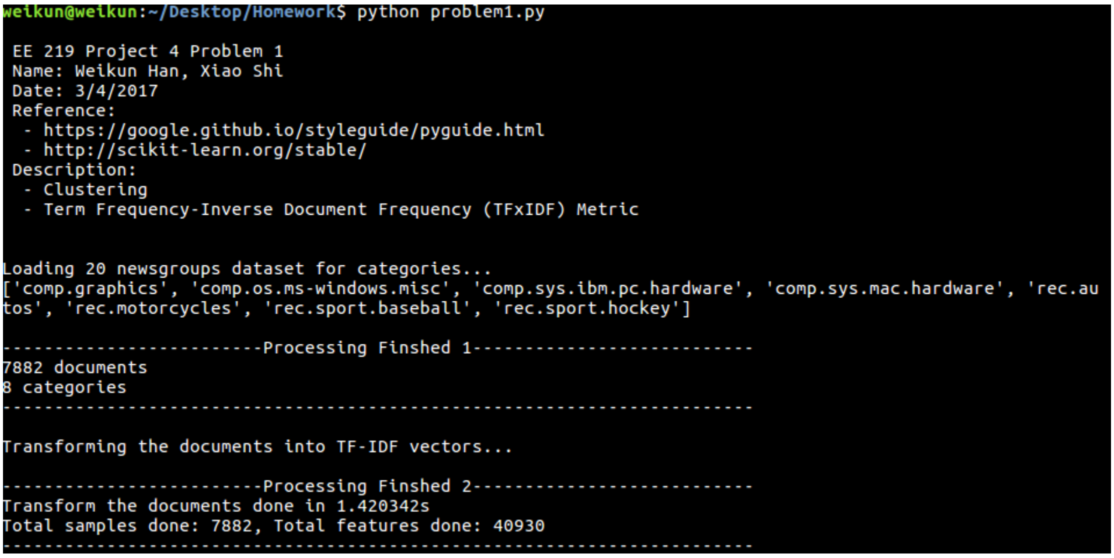
class `sklearn.feature_extraction.text.` **`TfidfVectorizer`** (*input=u'content', encoding=u'utf-8', decode_error=u'strict', strip_accents=None, lowercase=True, preprocessor=None, tokenizer=None, analyzer=u'word', stop_words=None, token_pattern=u'(?u)\b\w\w+\b', ngram_range=(1, 1), max_df=1.0, min_df=1, max_features=None, vocabulary=None, binary=False, dtype=<type 'numpy.int64'>, norm=u'l2', use_idf=True, smooth_idf=True, sublinear_tf=False*) [source]

We convert a collection of raw documents to a matrix of TF-IDF features. It is equivalent to CountVectorizer followed by TfidfTransformer.

Our model setting is the following:

```
TfidfVectorizer(max_df = 0.5,
 max_features = 100000,
 min_df = 2,
 stop_words = 'english',
 use_idf = True)
```

Therefore, we get the result as follows:



```
weikun@weikun:~/Desktop/Homework$ python problem1.py

 EE 219 Project 4 Problem 1
 Name: Weikun Han, Xiao Shi
 Date: 3/4/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'rec.au
tos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']

------------------------Processing Finshed 1-------------------------
7882 documents
8 categories
---------------------------------------------------------------------

Transforming the documents into TF-IDF vectors...

------------------------Processing Finshed 2-------------------------
Transform the documents done in 1.420342s
Total samples done: 7882, Total features done: 40930
---------------------------------------------------------------------
```

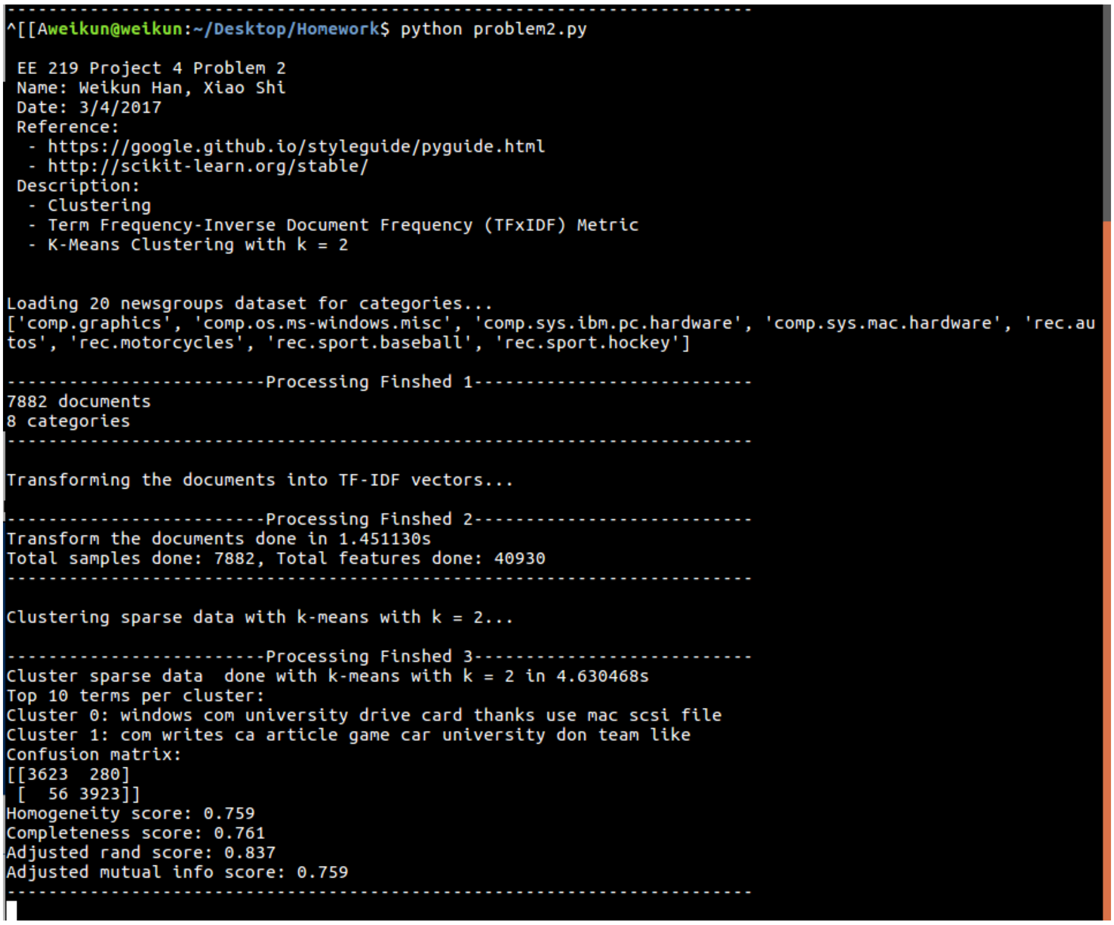
2)

In this problem, we use sklearn model to do clustering. The model "default" information is the following and we need change some numbers from below default setting.

km = KMeans(n_clusters=2, init='k-means++', max_iter=100, n_init=1, verbose = False)

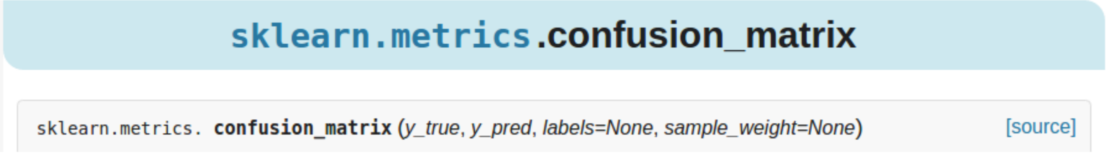
In this question, we set n_clusters = 2.

Here is the screenshot of our detailed processing information:



```
------------------------------------------------------------------------
^[[Aweikun@weikun:~/Desktop/Homework$ python problem2.py

 EE 219 Project 4 Problem 2
 Name: Weikun Han, Xiao Shi
 Date: 3/4/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 2


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'rec.au
tos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']

-------------------------Processing Finshed 1--------------------------
7882 documents
8 categories
------------------------------------------------------------------------

Transforming the documents into TF-IDF vectors...

-------------------------Processing Finshed 2--------------------------
Transform the documents done in 1.451130s
Total samples done: 7882, Total features done: 40930
------------------------------------------------------------------------

Clustering sparse data with k-means with k = 2...

-------------------------Processing Finshed 3--------------------------
Cluster sparse data  done with k-means with k = 2 in 4.630468s
Top 10 terms per cluster:
Cluster 0: windows com university drive card thanks use mac scsi file
Cluster 1: com writes ca article game car university don team like
Confusion matrix:
[[3623  280]
 [  56 3923]]
Homogeneity score: 0.759
Completeness score: 0.761
Adjusted rand score: 0.837
Adjusted mutual info score: 0.759
------------------------------------------------------------------------
```

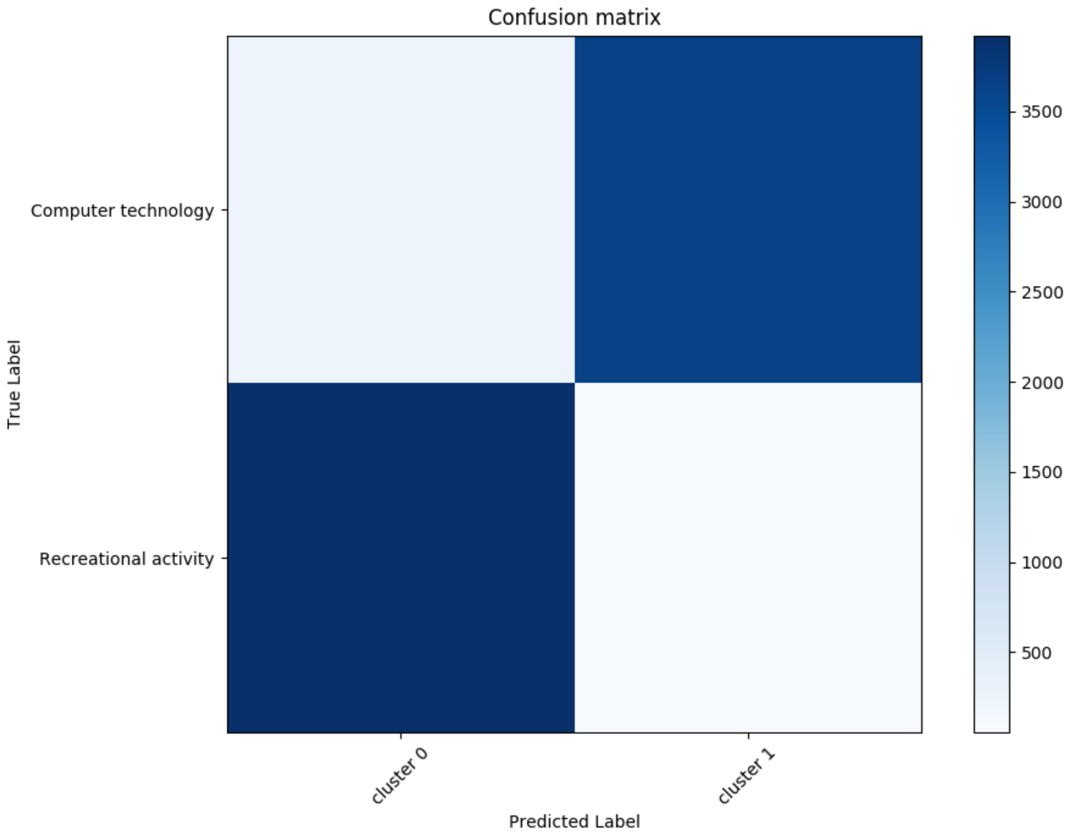
To get the confusion matrix with normalization and without normalization we use the following code



sklearn.metrics.confusion_matrix

sklearn.metrics. **confusion_matrix** (*y_true*, *y_pred*, *labels=None*, *sample_weight=None*) [source]

Then we got the confusion as the following:

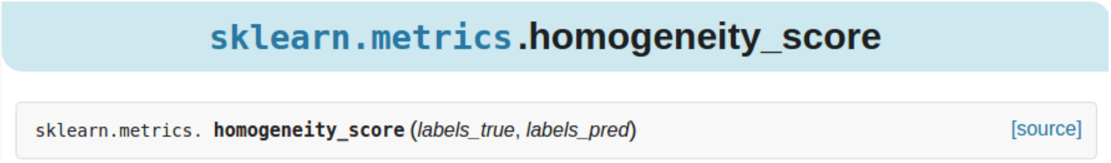


A permutation of the rows that makes confusion matrix look almost diagonal

And the homogeneity score, completeness score, adjusted rand score and the adjusted mutual info score are show in the following table.

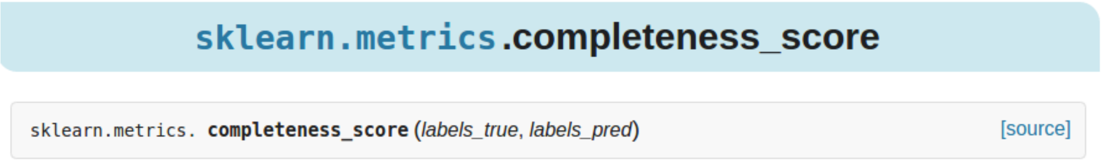
Homogeneity score	0.759
Completeness score	0.761
Adjusted rand score	0.837
Adjusted mutual info score	0.759

We are using the following code to calculate the numbers in previous table.



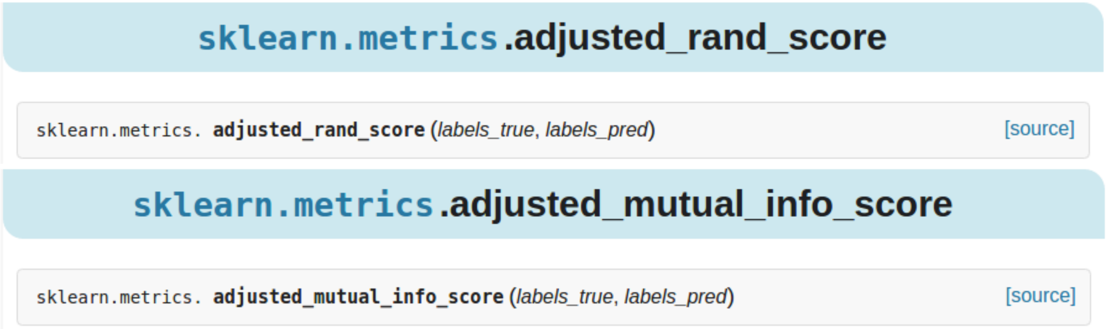
sklearn.metrics.homogeneity_score

`sklearn.metrics.` **`homogeneity_score`** (*labels_true*, *labels_pred*) [source]



sklearn.metrics.completeness_score

sklearn.metrics. **completeness_score** (*labels_true*, *labels_pred*) [source]



sklearn.metrics.adjusted_rand_score

sklearn.metrics. **adjusted_rand_score** (*labels_true*, *labels_pred*) [source]

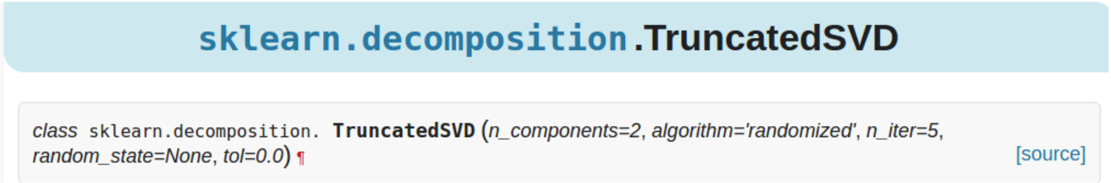
sklearn.metrics.adjusted_mutual_info_score

sklearn.metrics. **adjusted_mutual_info_score** (*labels_true*, *labels_pred*) [source]

3)

In this problem, we know that high dimensional sparse TF-IDF vectors do not yield a good clustering performance. Therefore, we can use Latent Semantic Indexing(LSI) and Non-negative Matrix Factorization(NMF) to reduce dimension of the data by sweeping over the dimension parameter.

(1) First, we use the LSI, which have been built in model in sklearn. We use the following code:



sklearn.decomposition.TruncatedSVD

class sklearn.decomposition. **TruncatedSVD** (*n_components=2*, *algorithm='randomized'*, *n_iter=5*, *random_state=None*, *tol=0.0*) ¶ [source]

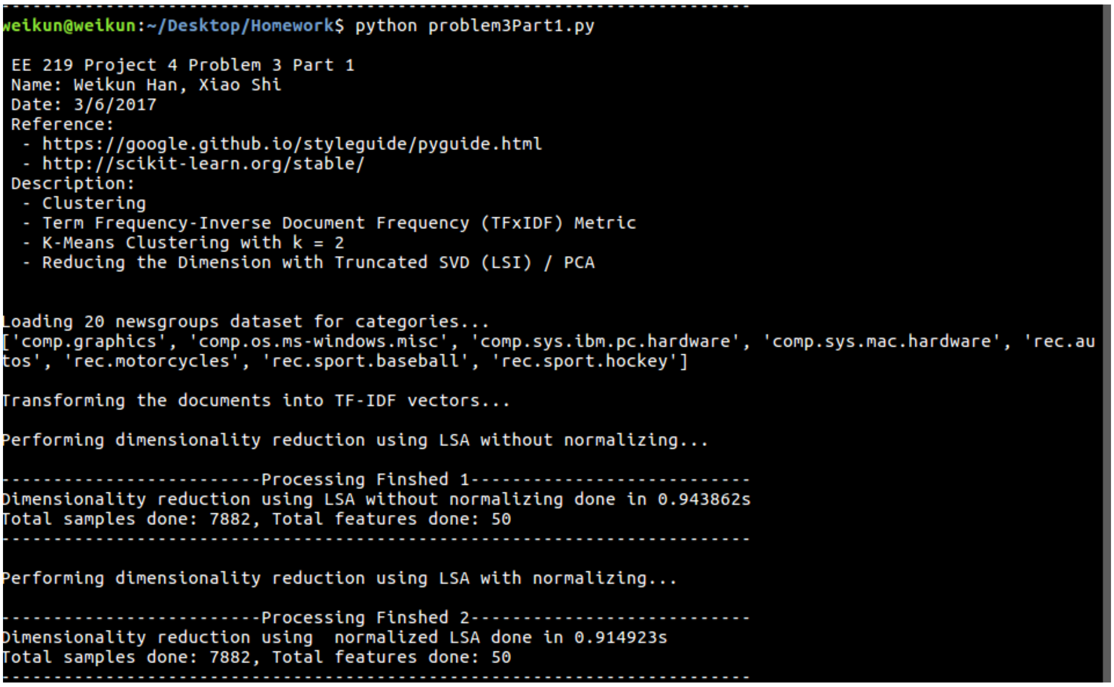
We are doing dimensional reduction using truncated SVD (aka LSA), which transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). Contrary to PCA, this estimator does not center the data before computing the singular value decomposition. This means it can work with scipy.

In particular, truncated SVD works on term count/tf-idf matrices as returned by the vectorizers in sklearn.feature_extraction.text. In that context, it is known as latent semantic analysis (LSA).

This estimator supports two algorithms: a fast randomized SVD solver, and a “naive” algorithm that uses ARPACK as an eigensolver on (X * X.T) or (X.T * X), whichever is more efficient.

In this case, we let the dimension reduce to 50 using the code
svd = truncatedSVD(n_components = 50)

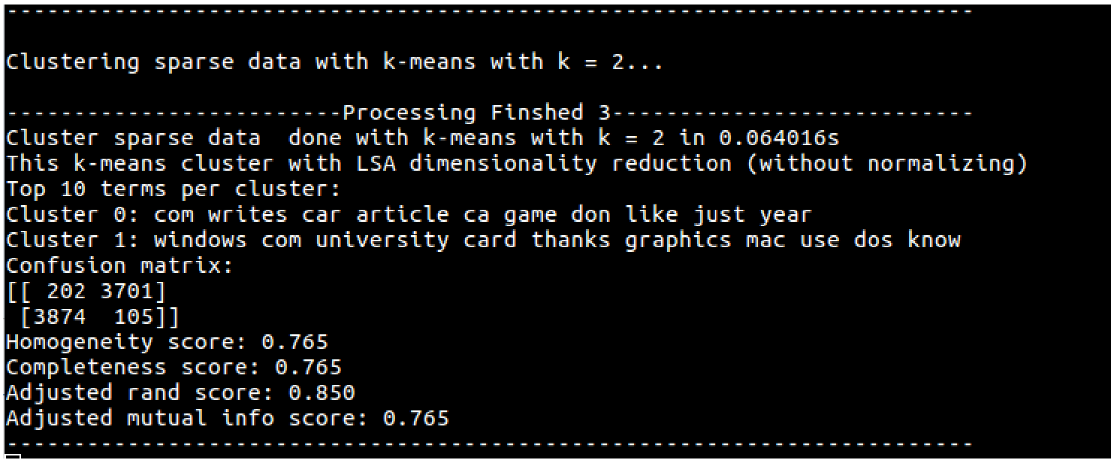
The result is the following:



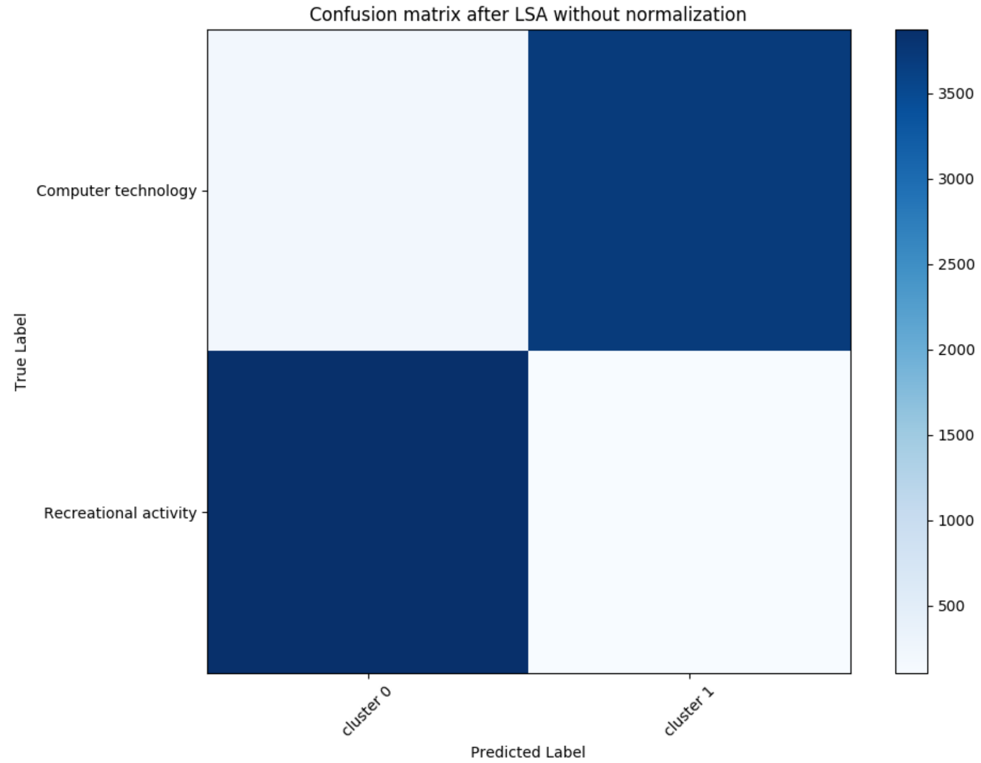
```
----------------------------------------------------------------------
weikun@weikun:~/Desktop/Homework$ python problem3Part1.py

 EE 219 Project 4 Problem 3 Part 1
 Name: Weikun Han, Xiao Shi
 Date: 3/6/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 2
  - Reducing the Dimension with Truncated SVD (LSI) / PCA


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'rec.au
tos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']

Transforming the documents into TF-IDF vectors...

Performing dimensionality reduction using LSA without normalizing...

-------------------------Processing Finshed 1--------------------------
Dimensionality reduction using LSA without normalizing done in 0.943862s
Total samples done: 7882, Total features done: 50
----------------------------------------------------------------------

Performing dimensionality reduction using LSA with normalizing...

-------------------------Processing Finshed 2--------------------------
Dimensionality reduction using  normalized LSA done in 0.914923s
Total samples done: 7882, Total features done: 50
----------------------------------------------------------------------
```

From the result, we can see the dimension is reduced to 50

Next, we use the result after NMF without adding nonlinear transformation to get the final data representation as following:



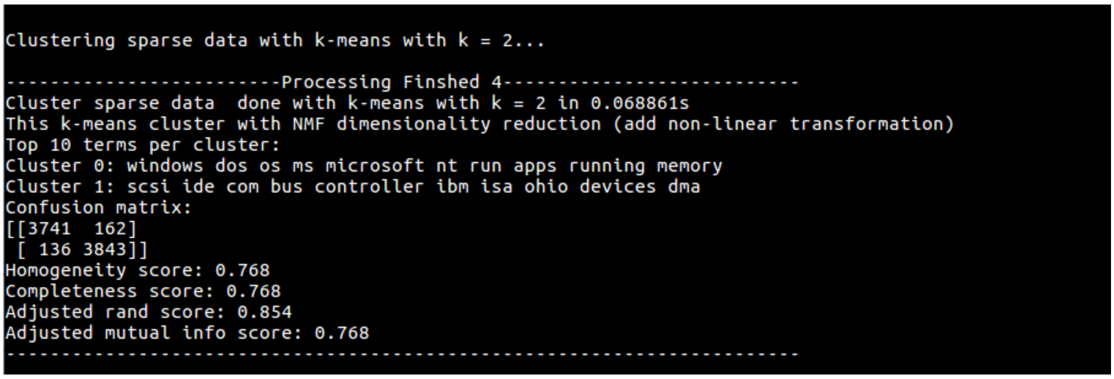
```
----------------------------------------------------------------------


Clustering sparse data with k-means with k = 2...

-------------------------Processing Finshed 3--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.064016s
This k-means cluster with LSA dimensionality reduction (without normalizing)
Top 10 terms per cluster:
Cluster 0: com writes car article ca game don like just year
Cluster 1: windows com university card thanks graphics mac use dos know
Confusion matrix:
[[ 202 3701]
 [3874  105]]
Homogeneity score: 0.765
Completeness score: 0.765
Adjusted rand score: 0.850
Adjusted mutual info score: 0.765
----------------------------------------------------------------------
```



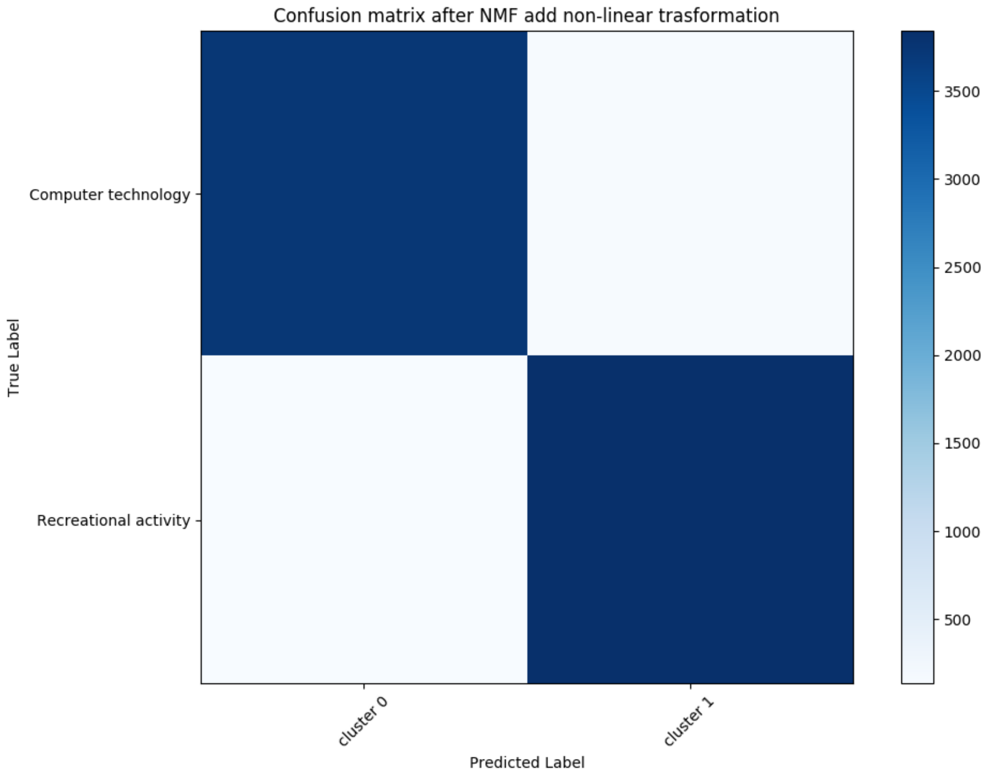
Moreover, the clustering purity is not satisfactory

Here, we are going to add nonlinear transformation

And the final data representation as following:



```
Clustering sparse data with k-means with k = 2...

-------------------------Processing Finshed 4--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.068861s
This k-means cluster with NMF dimensionality reduction (add non-linear transformation)
Top 10 terms per cluster:
Cluster 0: windows dos os ms microsoft nt run apps running memory
Cluster 1: scsi ide com bus controller ibm isa ohio devices dma
Confusion matrix:
[[3741  162]
 [ 136 3843]]
Homogeneity score: 0.768
Completeness score: 0.768
Adjusted rand score: 0.854
Adjusted mutual info score: 0.768
-----------------------------------------------------------------------
```

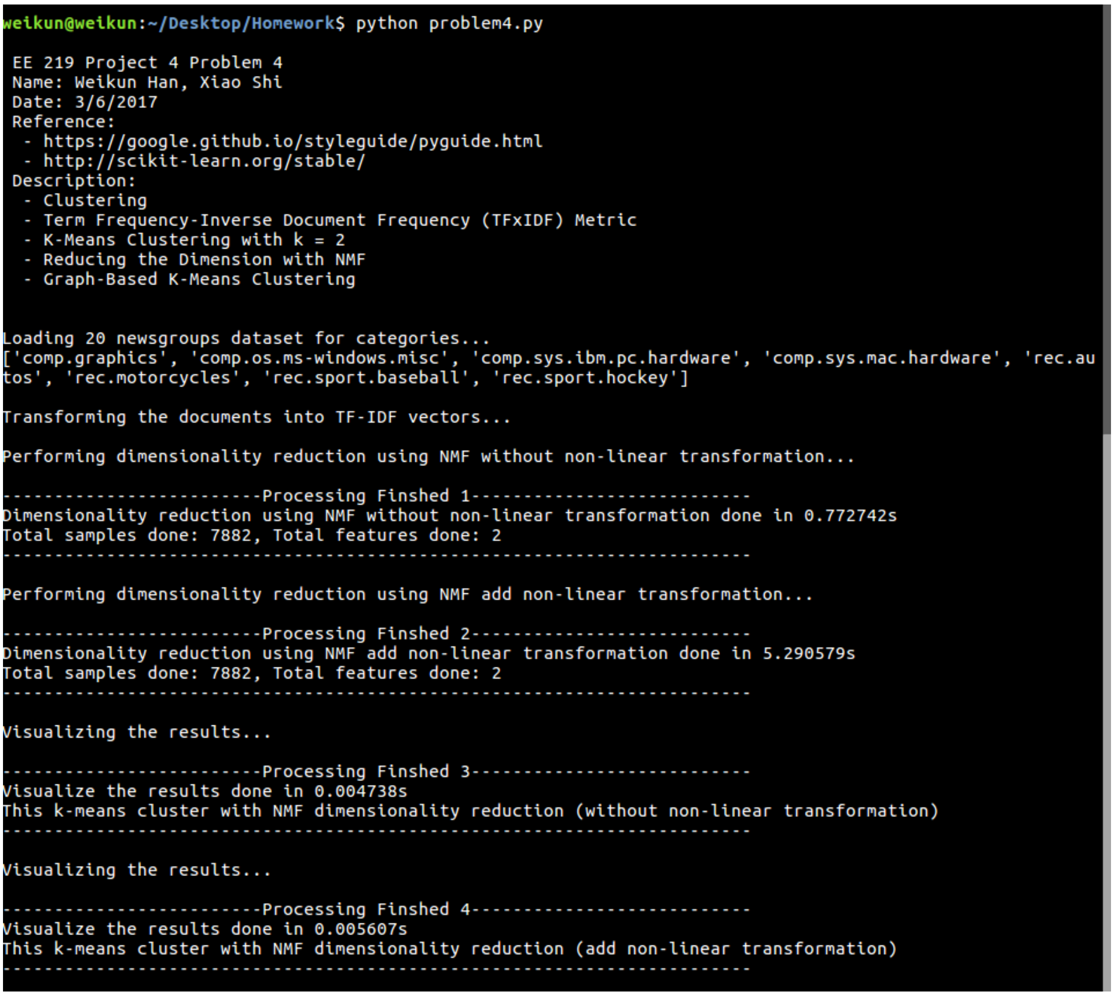


Now the result looks good, we can see that there are perfect two clusters (one is Computer technology, the other Recreational activity).

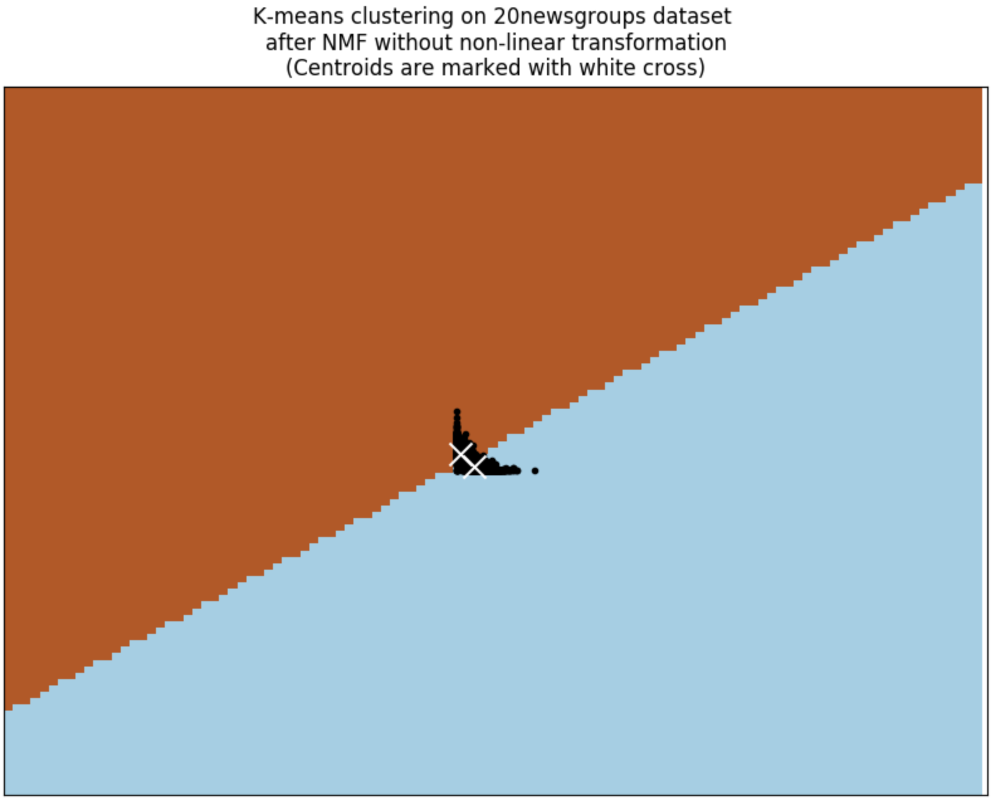
4)

In this question, we need visualize the performance of your clustering by projecting final data vectors onto 2 dimensions and color-coding the classes. Therefore, we need reduce the dimension to 2.

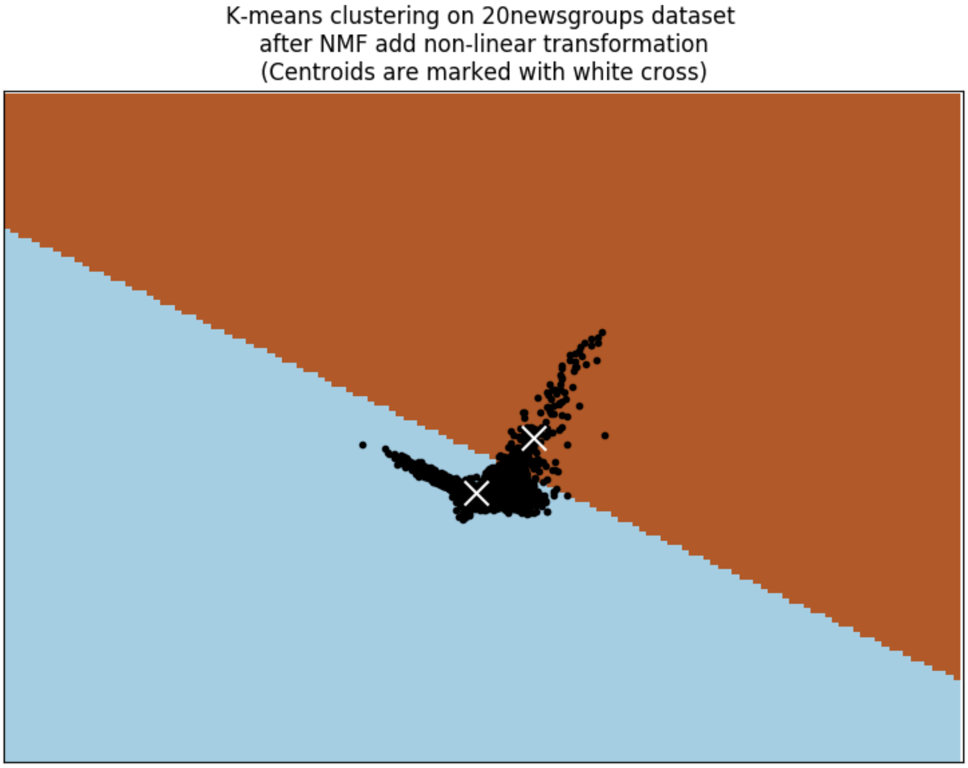
The process result is as following:



```
weikun@weikun:~/Desktop/Homework$ python problem4.py

 EE 219 Project 4 Problem 4
 Name: Weikun Han, Xiao Shi
 Date: 3/6/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 2
  - Reducing the Dimension with NMF
  - Graph-Based K-Means Clustering


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'rec.au
tos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']

Transforming the documents into TF-IDF vectors...

Performing dimensionality reduction using NMF without non-linear transformation...

------------------------Processing Finshed 1--------------------------
Dimensionality reduction using NMF without non-linear transformation done in 0.772742s
Total samples done: 7882, Total features done: 2
----------------------------------------------------------------------

Performing dimensionality reduction using NMF add non-linear transformation...

------------------------Processing Finshed 2--------------------------
Dimensionality reduction using NMF add non-linear transformation done in 5.290579s
Total samples done: 7882, Total features done: 2
----------------------------------------------------------------------

Visualizing the results...

------------------------Processing Finshed 3--------------------------
Visualize the results done in 0.004738s
This k-means cluster with NMF dimensionality reduction (without non-linear transformation)
----------------------------------------------------------------------

Visualizing the results...

------------------------Processing Finshed 4--------------------------
Visualize the results done in 0.005607s
This k-means cluster with NMF dimensionality reduction (add non-linear transformation)
----------------------------------------------------------------------
```



K-means clustering on 20newsgroups dataset
after NMF without non-linear transformation
(Centroids are marked with white cross)



K-means clustering on 20newsgroups dataset
after NMF add non-linear transformation
(Centroids are marked with white cross)

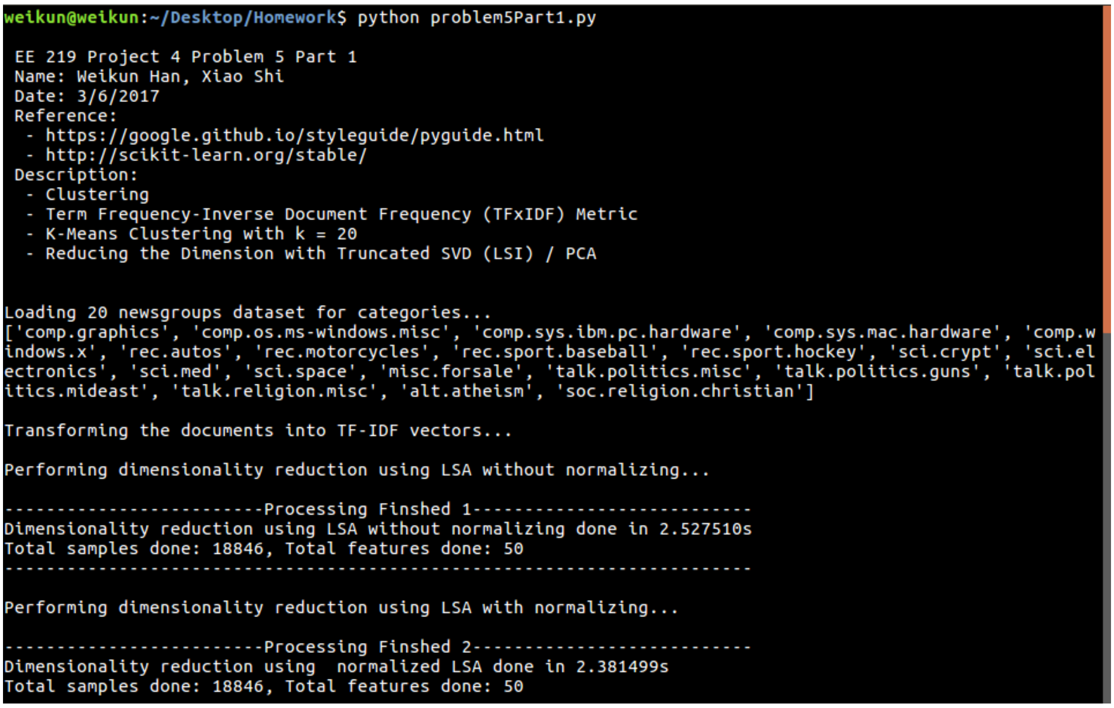
From the above two plot we can know a non-linear transform is useful because 2 clusters are more clearly.

5)

In this problem, we can retrieve all the 20 original sub-class labels with clustering. Therefore, we need include all the documents and the corresponding terms in the data matrix and find proper representation through reducing the dimension of the TF-IDF representation.

Here, we first use K-means clustering with k=20 in order to find pure clusters with respect to the class labels.

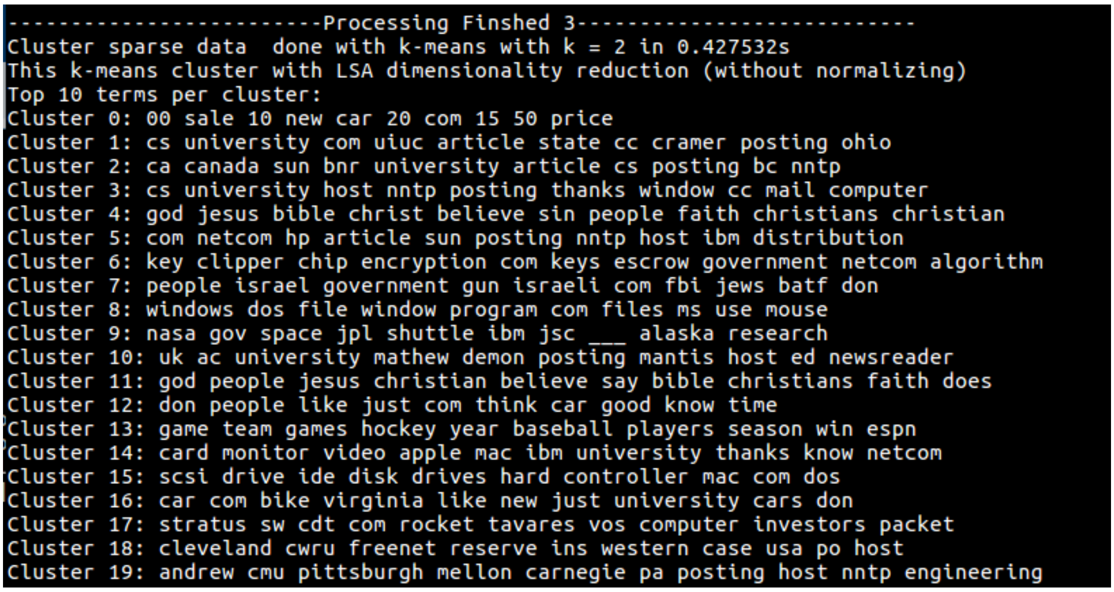
The LSI result is as following:



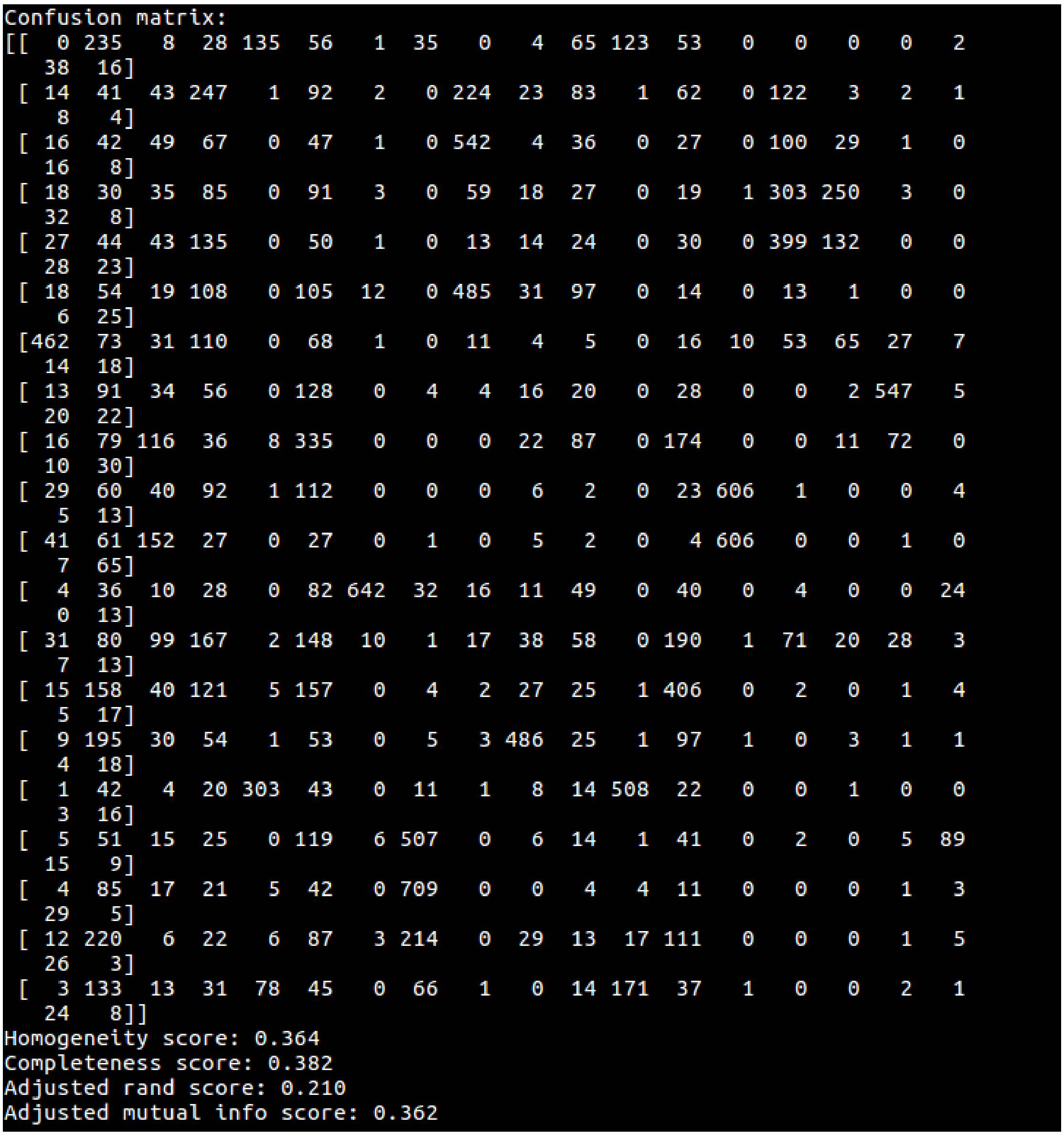
```
weikun@weikun:~/Desktop/Homework$ python problem5Part1.py

 EE 219 Project 4 Problem 5 Part 1
 Name: Weikun Han, Xiao Shi
 Date: 3/6/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 20
  - Reducing the Dimension with Truncated SVD (LSI) / PCA


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.w
indows.x', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.el
ectronics', 'sci.med', 'sci.space', 'misc.forsale', 'talk.politics.misc', 'talk.politics.guns', 'talk.pol
itics.mideast', 'talk.religion.misc', 'alt.atheism', 'soc.religion.christian']

Transforming the documents into TF-IDF vectors...

Performing dimensionality reduction using LSA without normalizing...

------------------------Processing Finshed 1--------------------------
Dimensionality reduction using LSA without normalizing done in 2.527510s
Total samples done: 18846, Total features done: 50
----------------------------------------------------------------------

Performing dimensionality reduction using LSA with normalizing...

------------------------Processing Finshed 2--------------------------
Dimensionality reduction using  normalized LSA done in 2.381499s
Total samples done: 18846, Total features done: 50
```

From the result: we can see the dimension is reduction to 50

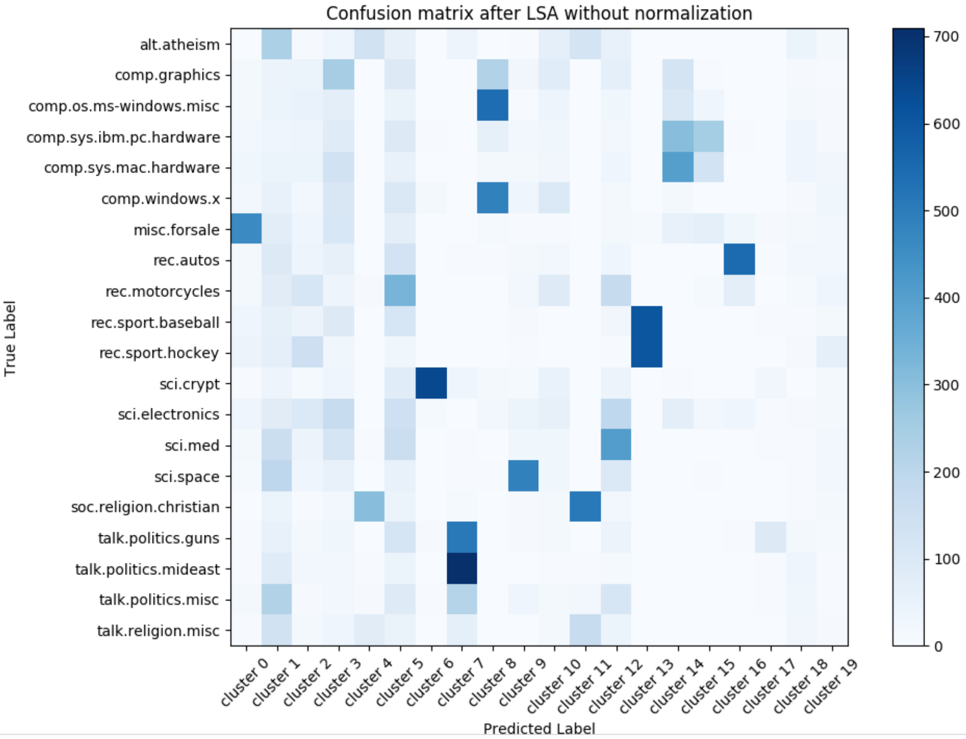
Next, we use the result after LSA without normalization to get the final data representation as following:



```
------------------------Processing Finshed 3--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.427532s
This k-means cluster with LSA dimensionality reduction (without normalizing)
Top 10 terms per cluster:
Cluster 0: 00 sale 10 new car 20 com 15 50 price
Cluster 1: cs university com uiuc article state cc cramer posting ohio
Cluster 2: ca canada sun bnr university article cs posting bc nntp
Cluster 3: cs university host nntp posting thanks window cc mail computer
Cluster 4: god jesus bible christ believe sin people faith christians christian
Cluster 5: com netcom hp article sun posting nntp host ibm distribution
Cluster 6: key clipper chip encryption com keys escrow government netcom algorithm
Cluster 7: people israel government gun israeli com fbi jews batf don
Cluster 8: windows dos file window program com files ms use mouse
Cluster 9: nasa gov space jpl shuttle ibm jsc ___ alaska research
Cluster 10: uk ac university mathew demon posting mantis host ed newsreader
Cluster 11: god people jesus christian believe say bible christians faith does
Cluster 12: don people like just com think car good know time
Cluster 13: game team games hockey year baseball players season win espn
Cluster 14: card monitor video apple mac ibm university thanks know netcom
Cluster 15: scsi drive ide disk drives hard controller mac com dos
Cluster 16: car com bike virginia like new just university cars don
Cluster 17: stratus sw cdt com rocket tavares vos computer investors packet
Cluster 18: cleveland cwru freenet reserve ins western case usa po host
Cluster 19: andrew cmu pittsburgh mellon carnegie pa posting host nntp engineering
```



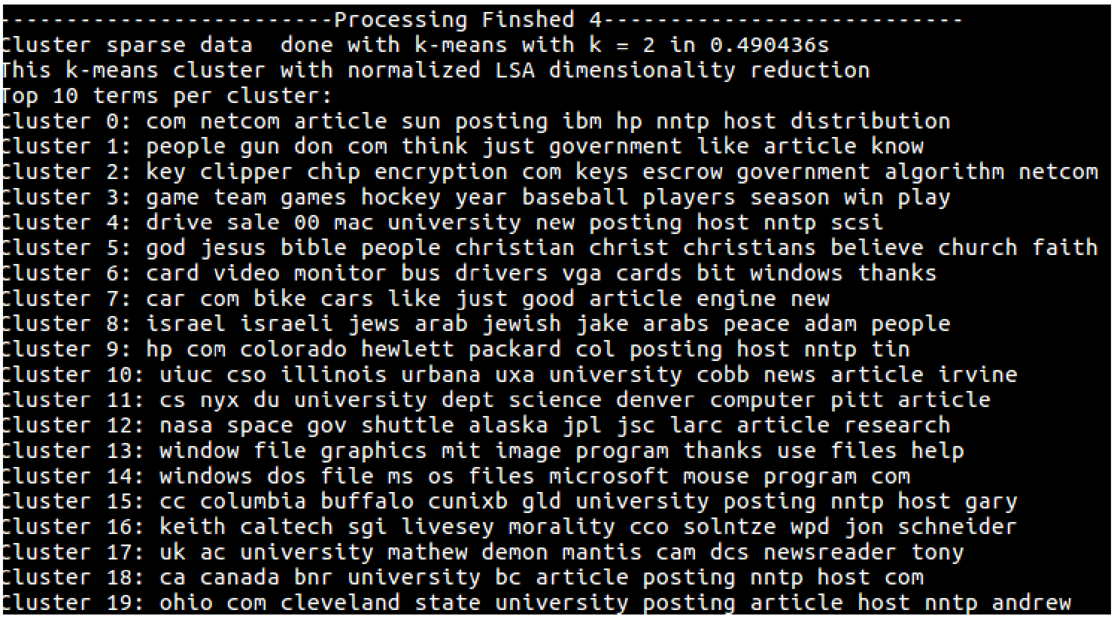
```
Confusion matrix:
[[  0 235   8  28 135  56   1  35   0   4  65 123  53   0   0   0   0   2
   38  16]
 [ 14  41  43 247   1  92   2   0 224  23  83   1  62   0 122   3   2   1
    8   4]
 [ 16  42  49  67   0  47   1   0 542   4  36   0  27   0 100  29   1   0
   16   8]
 [ 18  30  35  85   0  91   3   0  59  18  27   0  19   1 303 250   3   0
   32   8]
 [ 27  44  43 135   0  50   1   0  13  14  24   0  30   0 399 132   0   0
   28  23]
 [ 18  54  19 108   0 105  12   0 485  31  97   0  14   0  13   1   0   0
    6  25]
 [462  73  31 110   0  68   1   0  11   4   5   0  16  10  53  65  27   7
   14  18]
 [ 13  91  34  56   0 128   0   4   4  16  20   0  28   0   0   2 547   5
   20  22]
 [ 16  79 116  36   8 335   0   0   0  22  87   0 174   0   0  11  72   0
   10  30]
 [ 29  60  40  92   1 112   0   0   0   6   2   0  23 606   1   0   0   4
    5  13]
 [ 41  61 152  27   0  27   0   1   0   5   2   0   4 606   0   0   1   0
    7  65]
 [  4  36  10  28   0  82 642  32  16  11  49   0  40   0   4   0   0  24
    0  13]
 [ 31  80  99 167   2 148  10   1  17  38  58   0 190   1  71  20  28   3
    7  13]
 [ 15 158  40 121   5 157   0   4   2  27  25   1 406   0   2   0   1   4
    5  17]
 [  9 195  30  54   1  53   0   5   3 486  25   1  97   1   0   3   1   1
    4  18]
 [  1  42   4  20 303  43   0  11   1   8  14 508  22   0   0   1   0   0
    3  16]
 [  5  51  15  25   0 119   6 507   0   6  14   1  41   0   2   0   5  89
   15   9]
 [  4  85  17  21   5  42   0 709   0   0   4   4  11   0   0   0   1   3
   29   5]
 [ 12 220   6  22   6  87   3 214   0  29  13  17 111   0   0   0   1   5
   26   3]
 [  3 133  13  31  78  45   0  66   1   0  14 171  37   1   0   0   2   1
   24   8]]
Homogeneity score: 0.364
Completeness score: 0.382
Adjusted rand score: 0.210
Adjusted mutual info score: 0.362
```



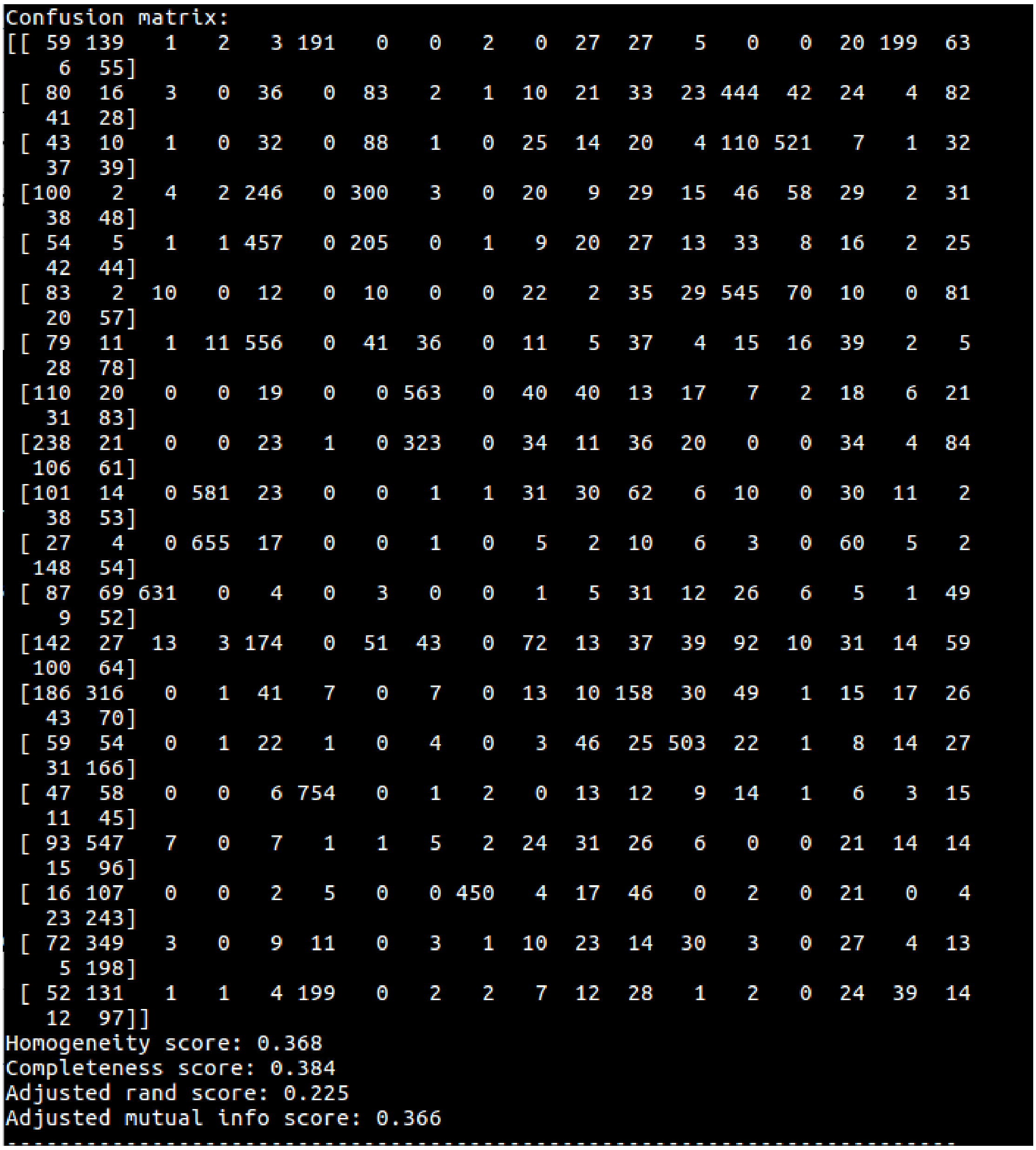
We can conclude that the clustering purity is not satisfying

Here, we are going to normalizing features (useing normalied LAS)

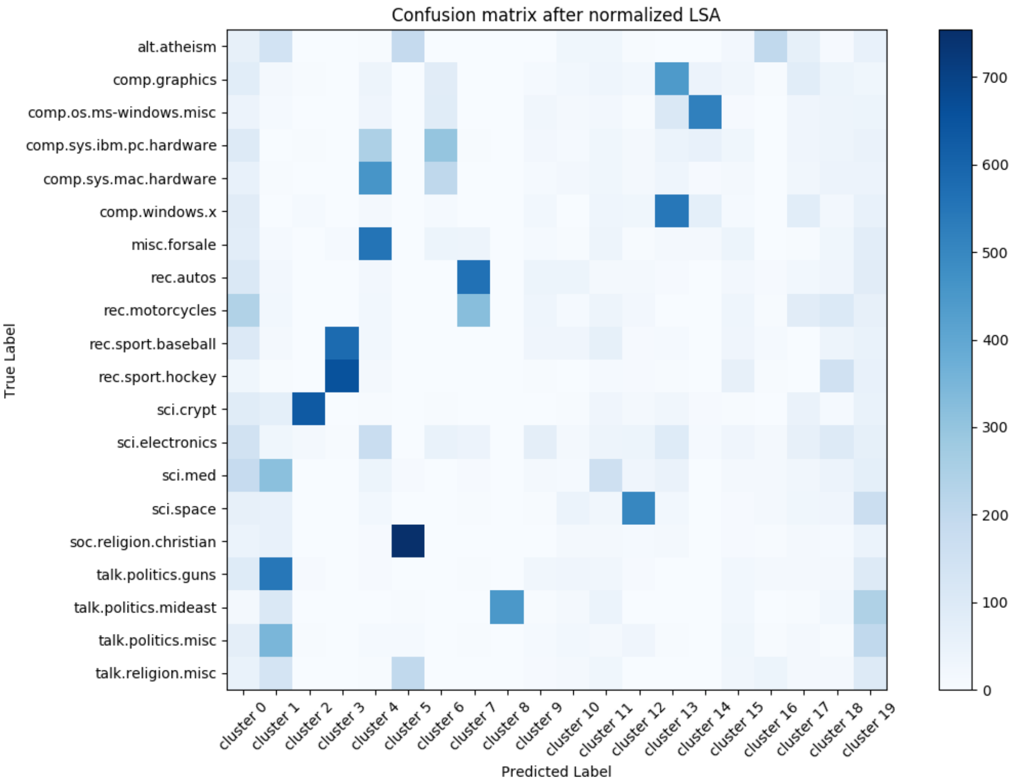
And the the final data representation is as following:



```
------------------------Processing Finshed 4--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.490436s
This k-means cluster with normalized LSA dimensionality reduction
Top 10 terms per cluster:
Cluster 0: com netcom article sun posting ibm hp nntp host distribution
Cluster 1: people gun don com think just government like article know
Cluster 2: key clipper chip encryption com keys escrow government algorithm netcom
Cluster 3: game team games hockey year baseball players season win play
Cluster 4: drive sale 00 mac university new posting host nntp scsi
Cluster 5: god jesus bible people christian christ christians believe church faith
Cluster 6: card video monitor bus drivers vga cards bit windows thanks
Cluster 7: car com bike cars like just good article engine new
Cluster 8: israel israeli jews arab jewish jake arabs peace adam people
Cluster 9: hp com colorado hewlett packard col posting host nntp tin
Cluster 10: uiuc cso illinois urbana uxa university cobb news article irvine
Cluster 11: cs nyx du university dept science denver computer pitt article
Cluster 12: nasa space gov shuttle alaska jpl jsc larc article research
Cluster 13: window file graphics mit image program thanks use files help
Cluster 14: windows dos file ms os files microsoft mouse program com
Cluster 15: cc columbia buffalo cunixb gld university posting nntp host gary
Cluster 16: keith caltech sgi livesey morality cco solntze wpd jon schneider
Cluster 17: uk ac university mathew demon mantis cam dcs newsreader tony
Cluster 18: ca canada bnr university bc article posting nntp host com
Cluster 19: ohio com cleveland state university posting article host nntp andrew
```

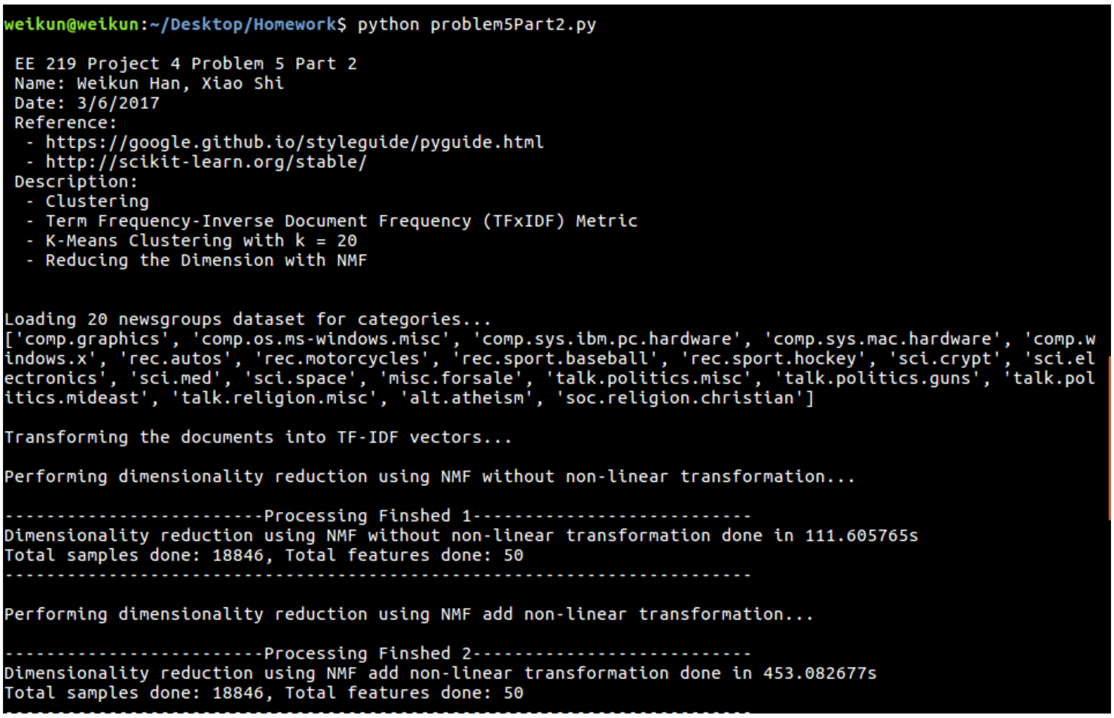


```
Confusion matrix:
[[ 59 139   1   2   3 191   0   0   2   0  27  27   5   0   0  20 199  63
    6  55]
 [ 80  16   3   0  36   0  83   2   1  10  21  33  23 444  42  24   4  82
   41  28]
 [ 43  10   1   0  32   0  88   1   0  25  14  20   4 110 521   7   1  32
   37  39]
 [100   2   4   2 246   0 300   3   0  20   9  29  15  46  58  29   2  31
   38  48]
 [ 54   5   1   1 457   0 205   0   1   9  20  27  13  33   8  16   2  25
   42  44]
 [ 83   2  10   0  12   0  10   0   0  22   2  35  29 545  70  10   0  81
   20  57]
 [ 79  11   1  11 556   0  41  36   0  11   5  37   4  15  16  39   2   5
   28  78]
 [110  20   0   0  19   0   0 563   0  40  40  13  17   7   2  18   6  21
   31  83]
 [238  21   0   0  23   1   0 323   0  34  11  36  20   0   0  34   4  84
  106  61]
 [101  14   0 581  23   0   0   1   1  31  30  62   6  10   0  30  11   2
   38  53]
 [ 27   4   0 655  17   0   0   1   0   5   2  10   6   3   0  60   5   2
  148  54]
 [ 87  69 631   0   4   0   3   0   0   1   5  31  12  26   6   5   1  49
    9  52]
 [142  27  13   3 174   0  51  43   0  72  13  37  39  92  10  31  14  59
  100  64]
 [186 316   0   1  41   7   0   7   0  13  10 158  30  49   1  15  17  26
   43  70]
 [ 59  54   0   1  22   1   0   4   0   3  46  25 503  22   1   8  14  27
   31 166]
 [ 47  58   0   0   6 754   0   1   2   0  13  12   9  14   1   6   3  15
   11  45]
 [ 93 547   7   0   7   1   1   5   2  24  31  26   6   0   0  21  14  14
   15  96]
 [ 16 107   0   0   2   5   0   0 450   4  17  46   0   2   0  21   0   4
   23 243]
 [ 72 349   3   0   9  11   0   3   1  10  23  14  30   3   0  27   4  13
    5 198]
 [ 52 131   1   1   4 199   0   2   2   7  12  28   1   2   0  24  39  14
   12  97]]
Homogeneity score: 0.368
Completeness score: 0.384
Adjusted rand score: 0.225
Adjusted mutual info score: 0.366
----------------------------------------------------------------------
```



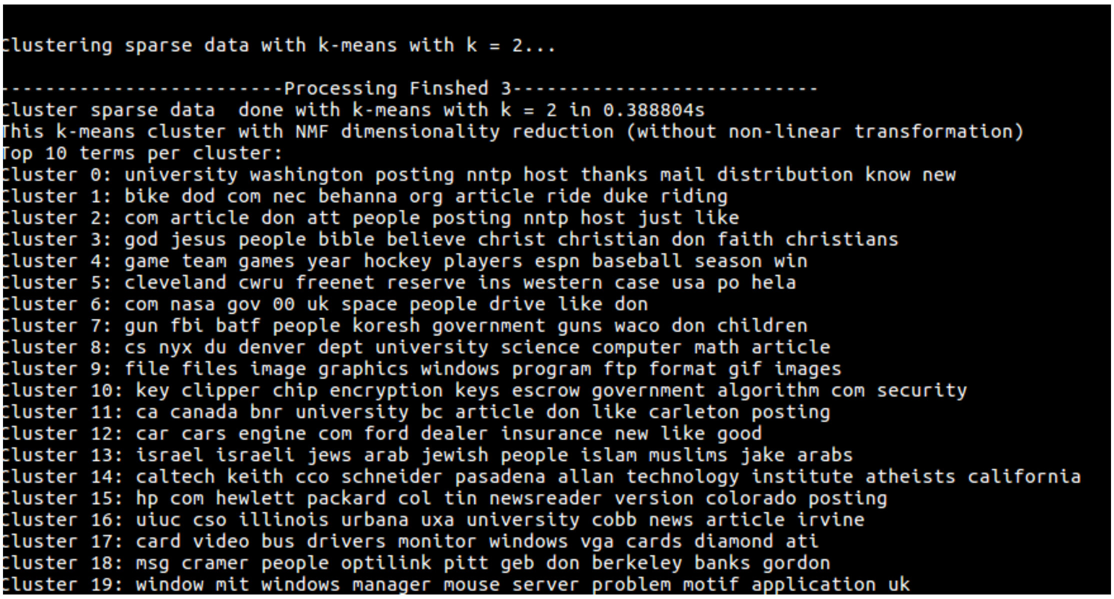
Secondly, we use the another decomposition method (NMF), which have build in model in sklearn.
We use the code:
NMF(n_components = 50, random_state = 2)

And get the following results:

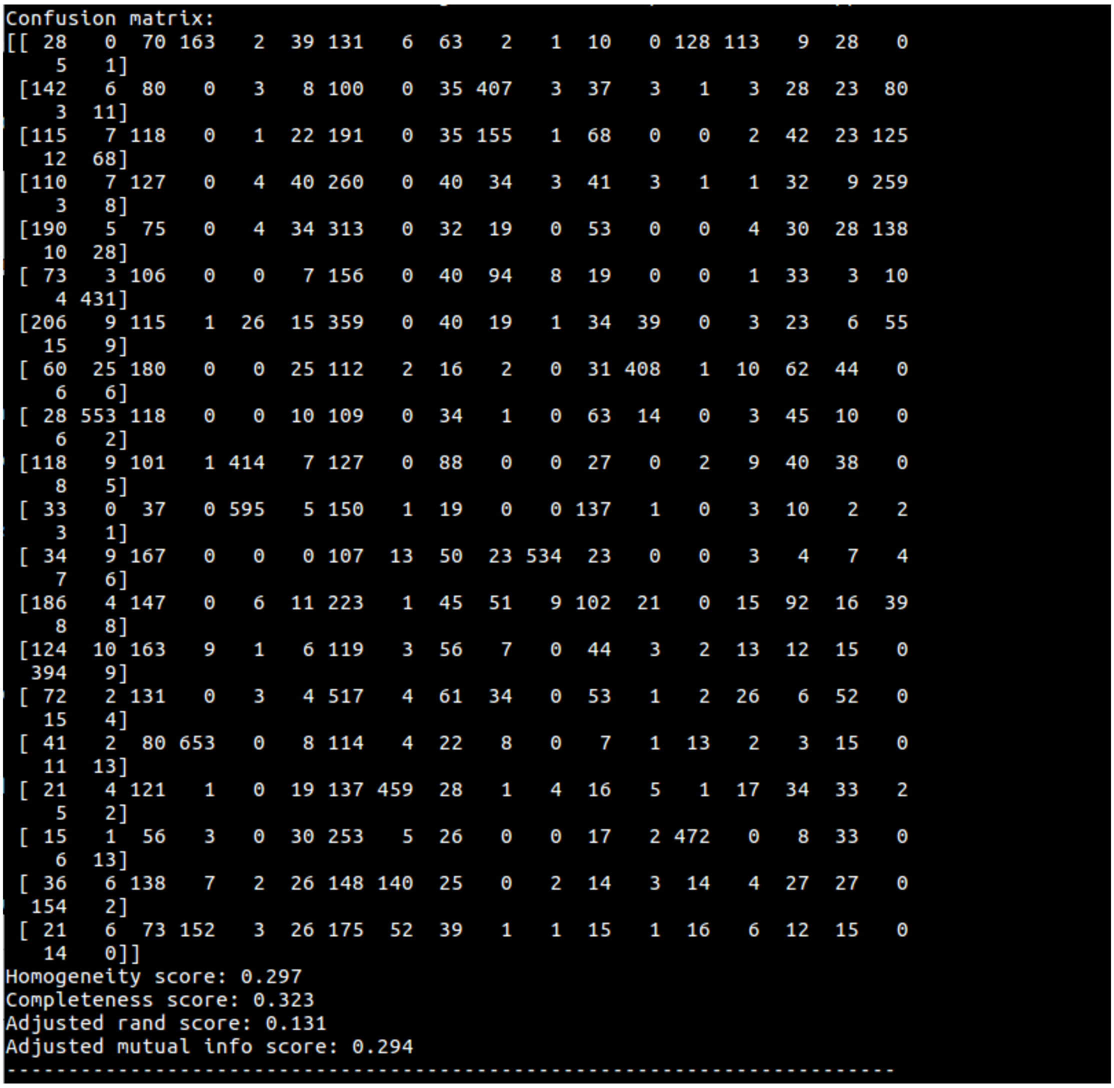


```
weikun@weikun:~/Desktop/Homework$ python problem5Part2.py

 EE 219 Project 4 Problem 5 Part 2
 Name: Weikun Han, Xiao Shi
 Date: 3/6/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 20
  - Reducing the Dimension with NMF


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.w
indows.x', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.el
ectronics', 'sci.med', 'sci.space', 'misc.forsale', 'talk.politics.misc', 'talk.politics.guns', 'talk.pol
itics.mideast', 'talk.religion.misc', 'alt.atheism', 'soc.religion.christian']

Transforming the documents into TF-IDF vectors...

Performing dimensionality reduction using NMF without non-linear transformation...

-------------------------Processing Finshed 1--------------------------
Dimensionality reduction using NMF without non-linear transformation done in 111.605765s
Total samples done: 18846, Total features done: 50
-----------------------------------------------------------------------

Performing dimensionality reduction using NMF add non-linear transformation...

-------------------------Processing Finshed 2--------------------------
Dimensionality reduction using NMF add non-linear transformation done in 453.082677s
Total samples done: 18846, Total features done: 50
-----------------------------------------------------------------------
```

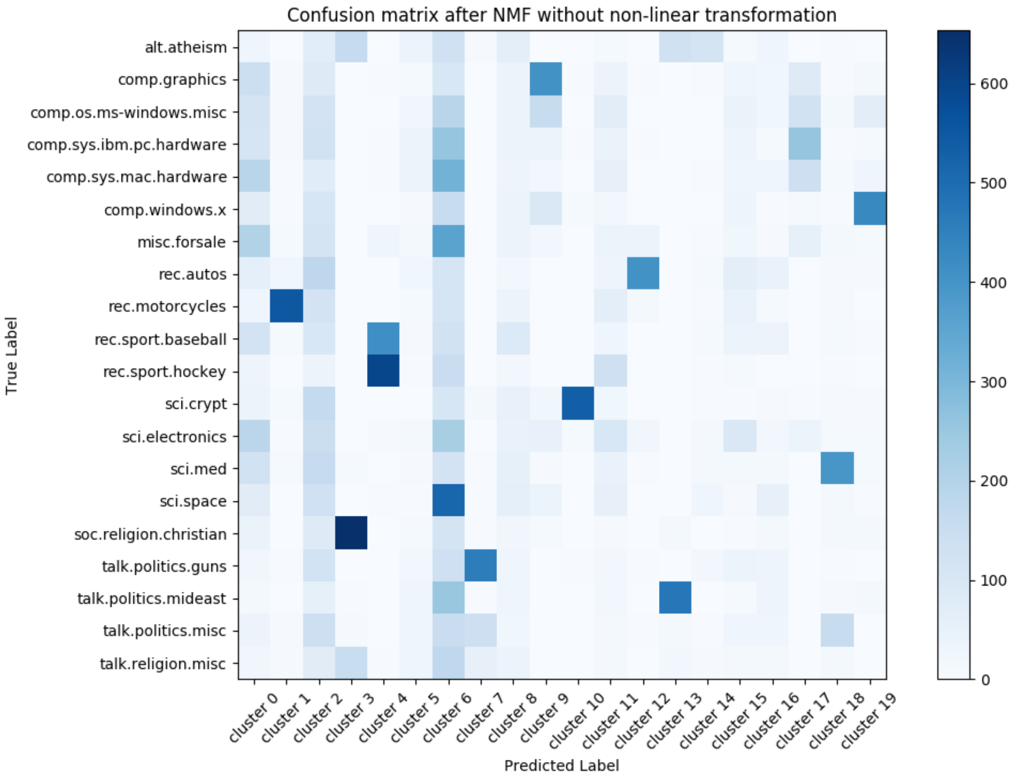
Next, we use the result after NMF without adding nonlinear transformation   to get the final data representation as follows:



```
Clustering sparse data with k-means with k = 2...

------------------------Processing Finshed 3--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.388804s
This k-means cluster with NMF dimensionality reduction (without non-linear transformation)
Top 10 terms per cluster:
Cluster 0: university washington posting nntp host thanks mail distribution know new
Cluster 1: bike dod com nec behanna org article ride duke riding
Cluster 2: com article don att people posting nntp host just like
Cluster 3: god jesus people bible believe christ christian don faith christians
Cluster 4: game team games year hockey players espn baseball season win
Cluster 5: cleveland cwru freenet reserve ins western case usa po hela
Cluster 6: com nasa gov 00 uk space people drive like don
Cluster 7: gun fbi batf people koresh government guns waco don children
Cluster 8: cs nyx du denver dept university science computer math article
Cluster 9: file files image graphics windows program ftp format gif images
Cluster 10: key clipper chip encryption keys escrow government algorithm com security
Cluster 11: ca canada bnr university bc article don like carleton posting
Cluster 12: car cars engine com ford dealer insurance new like good
Cluster 13: israel israeli jews arab jewish people islam muslims jake arabs
Cluster 14: caltech keith cco schneider pasadena allan technology institute atheists california
Cluster 15: hp com hewlett packard col tin newsreader version colorado posting
Cluster 16: uiuc cso illinois urbana uxa university cobb news article irvine
Cluster 17: card video bus drivers monitor windows vga cards diamond ati
Cluster 18: msg cramer people optilink pitt geb don berkeley banks gordon
Cluster 19: window mit windows manager mouse server problem motif application uk
```



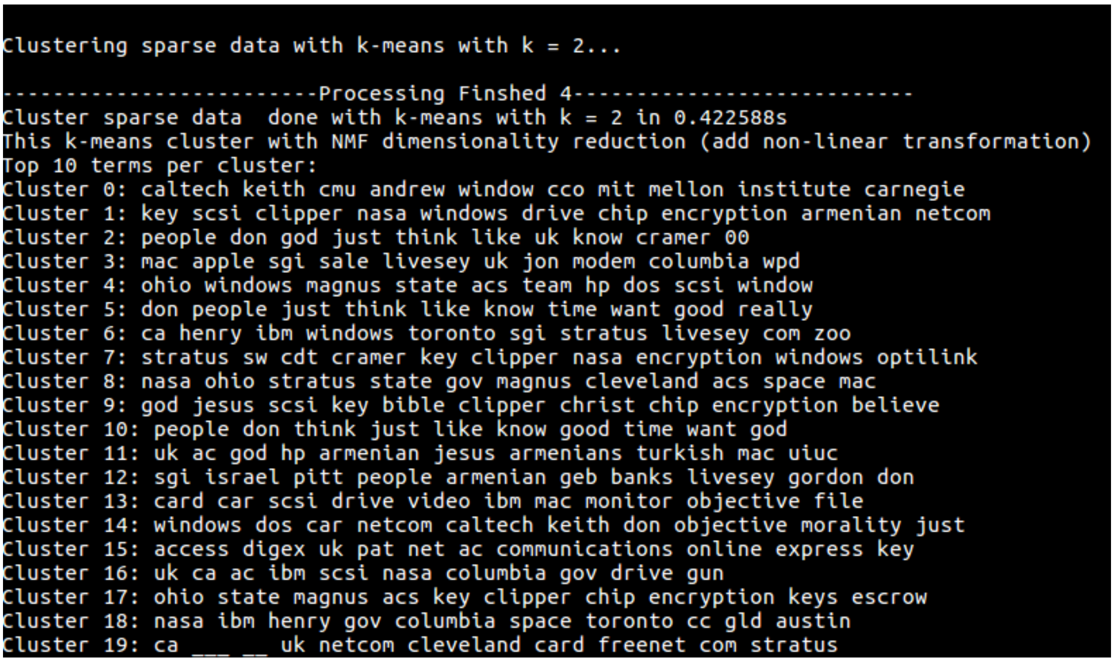
```
Confusion matrix:
[[ 28   0  70 163   2  39 131   6  63   2   1  10   0 128 113   9  28   0
    5   1]
 [142   6  80   0   3   8 100   0  35 407   3  37   3   1   3  28  23  80
    3  11]
 [115   7 118   0   1  22 191   0  35 155   1  68   0   0   2  42  23 125
   12  68]
 [110   7 127   0   4  40 260   0  40  34   3  41   3   1   1  32   9 259
    3   8]
 [190   5  75   0   4  34 313   0  32  19   0  53   0   0   4  30  28 138
   10  28]
 [ 73   3 106   0   0   7 156   0  40  94   8  19   0   0   1  33   3  10
    4 431]
 [206   9 115   1  26  15 359   0  40  19   1  34  39   0   3  23   6  55
   15   9]
 [ 60  25 180   0   0  25 112   2  16   2   0  31 408   1  10  62  44   0
    6   6]
 [ 28 553 118   0   0  10 109   0  34   1   0  63  14   0   3  45  10   0
    6   2]
 [118   9 101   1 414   7 127   0  88   0   0  27   0   2   9  40  38   0
    8   5]
 [ 33   0  37   0 595   5 150   1  19   0   0 137   1   0   3  10   2   2
    3   1]
 [ 34   9 167   0   0   0 107  13  50  23 534  23   0   0   3   4   7   4
    7   6]
 [186   4 147   0   6  11 223   1  45  51   9 102  21   0  15  92  16  39
    8   8]
 [124  10 163   9   1   6 119   3  56   7   0  44   3   2  13  12  15   0
  394   9]
 [ 72   2 131   0   3   4 517   4  61  34   0  53   1   2  26   6  52   0
   15   4]
 [ 41   2  80 653   0   8 114   4  22   8   0   7   1  13   2   3  15   0
   11  13]
 [ 21   4 121   1   0  19 137 459  28   1   4  16   5   1  17  34  33   2
    5   2]
 [ 15   1  56   3   0  30 253   5  26   0   0  17   2 472   0   8  33   0
    6  13]
 [ 36   6 138   7   2  26 148 140  25   0   2  14   3  14   4  27  27   0
  154   2]
 [ 21   6  73 152   3  26 175  52  39   1   1  15   1  16   6  12  15   0
   14   0]]
Homogeneity score: 0.297
Completeness score: 0.323
Adjusted rand score: 0.131
Adjusted mutual info score: 0.294
---------------------------------------------------------------------
```



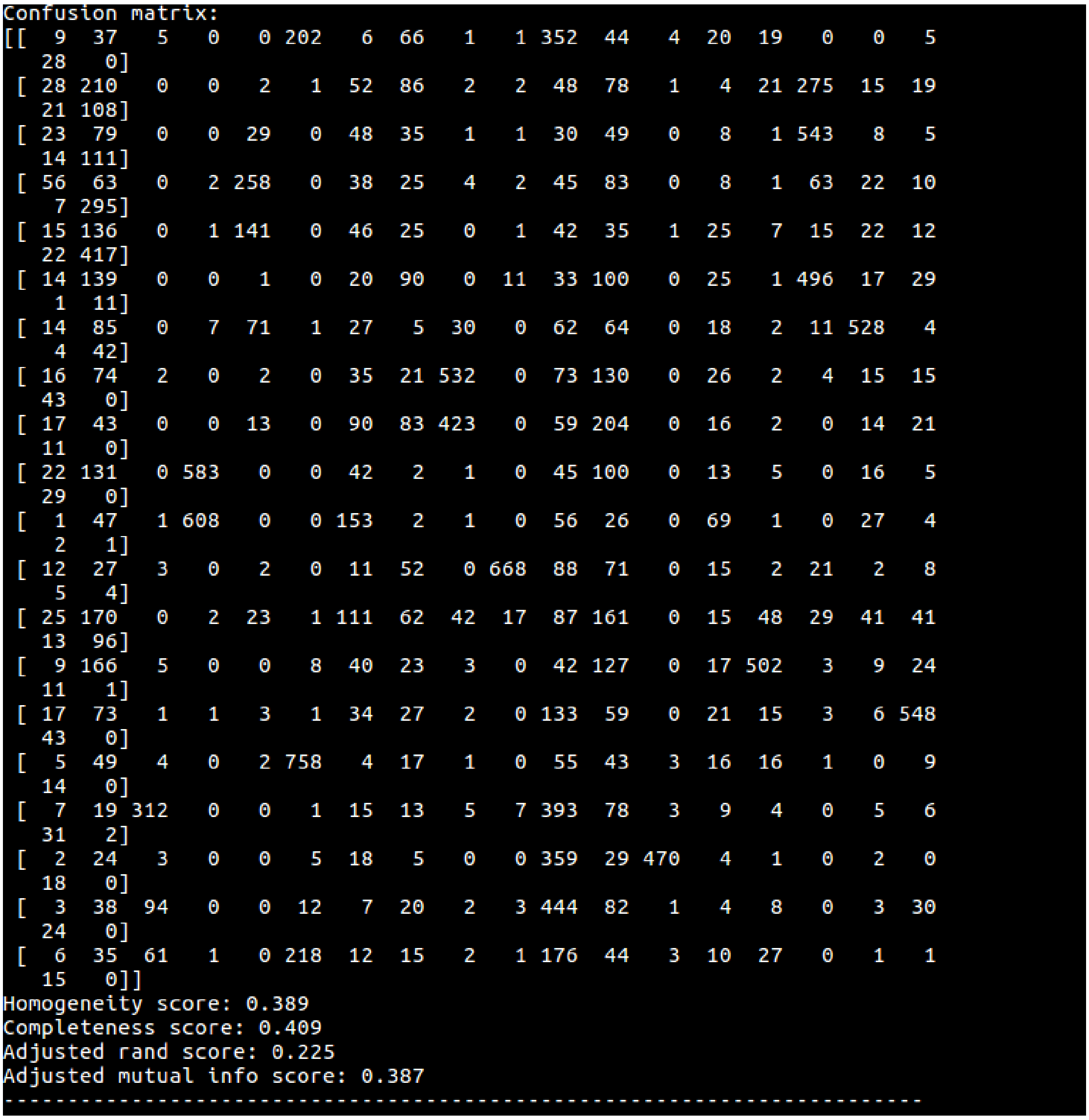
Moreover, the clustering purity is not satisfactory

Here, we are going to add nonlinear transformation

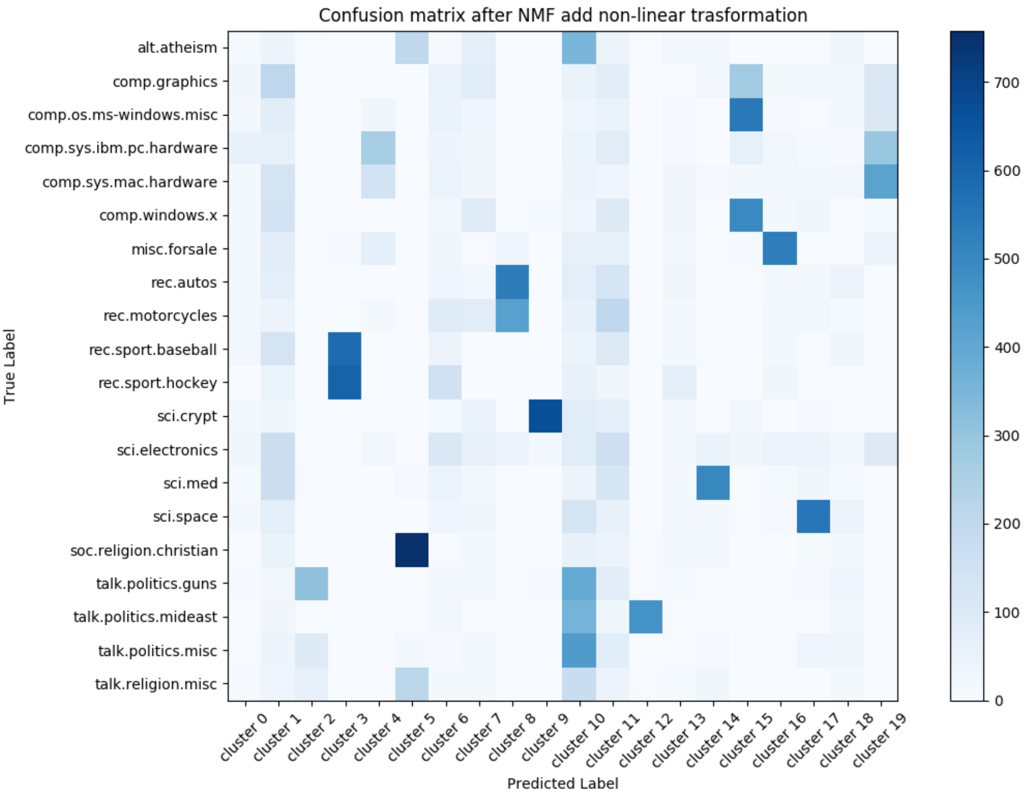
And the final data representation as follows:



```
Clustering sparse data with k-means with k = 2...

------------------------Processing Finshed 4--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.422588s
This k-means cluster with NMF dimensionality reduction (add non-linear transformation)
Top 10 terms per cluster:
Cluster 0: caltech keith cmu andrew window cco mit mellon institute carnegie
Cluster 1: key scsi clipper nasa windows drive chip encryption armenian netcom
Cluster 2: people don god just think like uk know cramer 00
Cluster 3: mac apple sgi sale livesey uk jon modem columbia wpd
Cluster 4: ohio windows magnus state acs team hp dos scsi window
Cluster 5: don people just think like know time want good really
Cluster 6: ca henry ibm windows toronto sgi stratus livesey com zoo
Cluster 7: stratus sw cdt cramer key clipper nasa encryption windows optilink
Cluster 8: nasa ohio stratus state gov magnus cleveland acs space mac
Cluster 9: god jesus scsi key bible clipper christ chip encryption believe
Cluster 10: people don think just like know good time want god
Cluster 11: uk ac god hp armenian jesus armenians turkish mac uiuc
Cluster 12: sgi israel pitt people armenian geb banks livesey gordon don
Cluster 13: card car scsi drive video ibm mac monitor objective file
Cluster 14: windows dos car netcom caltech keith don objective morality just
Cluster 15: access digex uk pat net ac communications online express key
Cluster 16: uk ca ac ibm scsi nasa columbia gov drive gun
Cluster 17: ohio state magnus acs key clipper chip encryption keys escrow
Cluster 18: nasa ibm henry gov columbia space toronto cc gld austin
Cluster 19: ca ___ __ uk netcom cleveland card freenet com stratus
```

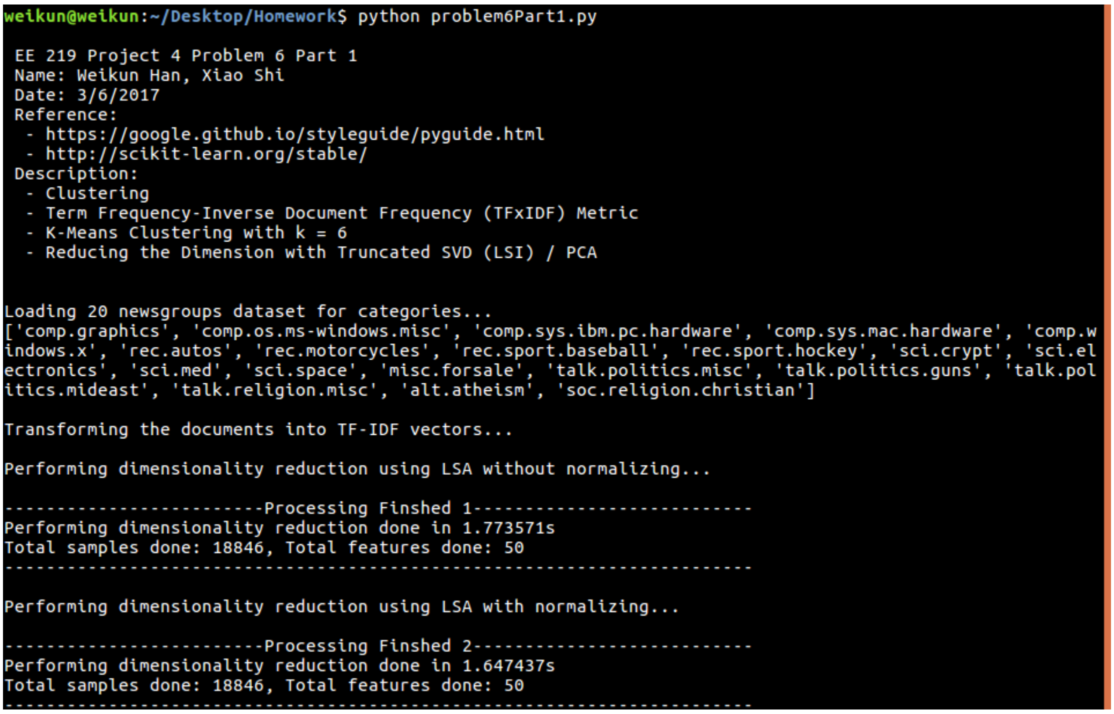


```
Confusion matrix:
[[  9  37   5   0   0 202   6  66   1   1 352  44   4  20  19   0   0   5
   28   0]
 [ 28 210   0   0   2   1  52  86   2   2  48  78   1   4  21 275  15  19
   21 108]
 [ 23  79   0   0  29   0  48  35   1   1  30  49   0   8   1 543   8   5
   14 111]
 [ 56  63   0   2 258   0  38  25   4   2  45  83   0   8   1  63  22  10
    7 295]
 [ 15 136   0   1 141   0  46  25   0   1  42  35   1  25   7  15  22  12
   22 417]
 [ 14 139   0   0   1   0  20  90   0  11  33 100   0  25   1 496  17  29
    1  11]
 [ 14  85   0   7  71   1  27   5  30   0  62  64   0  18   2  11 528   4
    4  42]
 [ 16  74   2   0   2   0  35  21 532   0  73 130   0  26   2   4  15  15
   43   0]
 [ 17  43   0   0  13   0  90  83 423   0  59 204   0  16   2   0  14  21
   11   0]
 [ 22 131   0 583   0   0  42   2   1   0  45 100   0  13   5   0  16   5
   29   0]
 [  1  47   1 608   0   0 153   2   1   0  56  26   0  69   1   0  27   4
    2   1]
 [ 12  27   3   0   2   0  11  52   0 668  88  71   0  15   2  21   2   8
    5   4]
 [ 25 170   0   2  23   1 111  62  42  17  87 161   0  15  48  29  41  41
   13  96]
 [  9 166   5   0   0   8  40  23   3   0  42 127   0  17 502   3   9  24
   11   1]
 [ 17  73   1   1   3   1  34  27   2   0 133  59   0  21  15   3   6 548
   43   0]
 [  5  49   4   0   2 758   4  17   1   0  55  43   3  16  16   1   0   9
   14   0]
 [  7  19 312   0   0   1  15  13   5   7 393  78   3   9   4   0   5   6
   31   2]
 [  2  24   3   0   0   5  18   5   0   0 359  29 470   4   1   0   2   0
   18   0]
 [  3  38  94   0   0  12   7  20   2   3 444  82   1   4   8   0   3  30
   24   0]
 [  6  35  61   1   0 218  12  15   2   1 176  44   3  10  27   0   1   1
   15   0]]
Homogeneity score: 0.389
Completeness score: 0.409
Adjusted rand score: 0.225
Adjusted mutual info score: 0.387
------------------------------------------------------------------------
```



6)

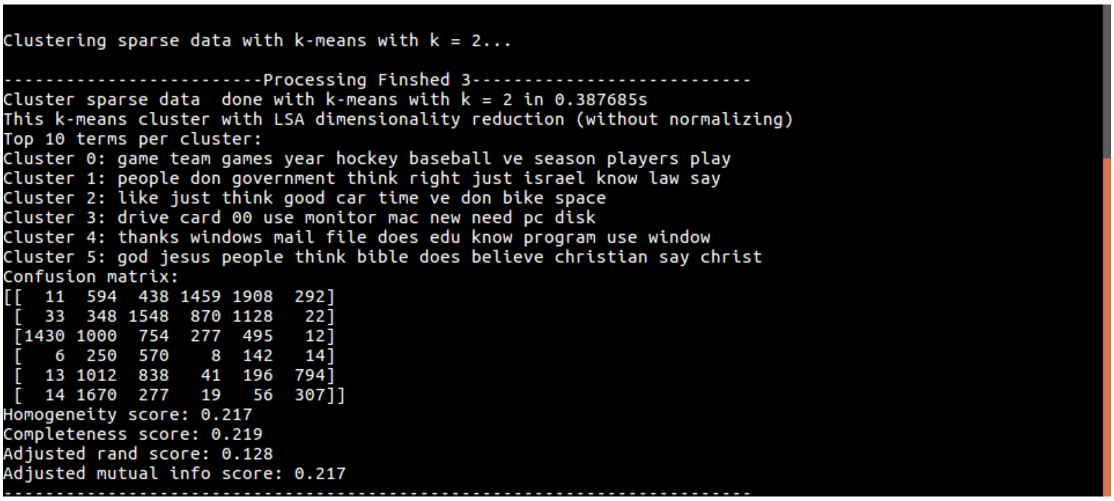
In this problem, after comparing the results of different dimensions, we choose K=6 and N=50.



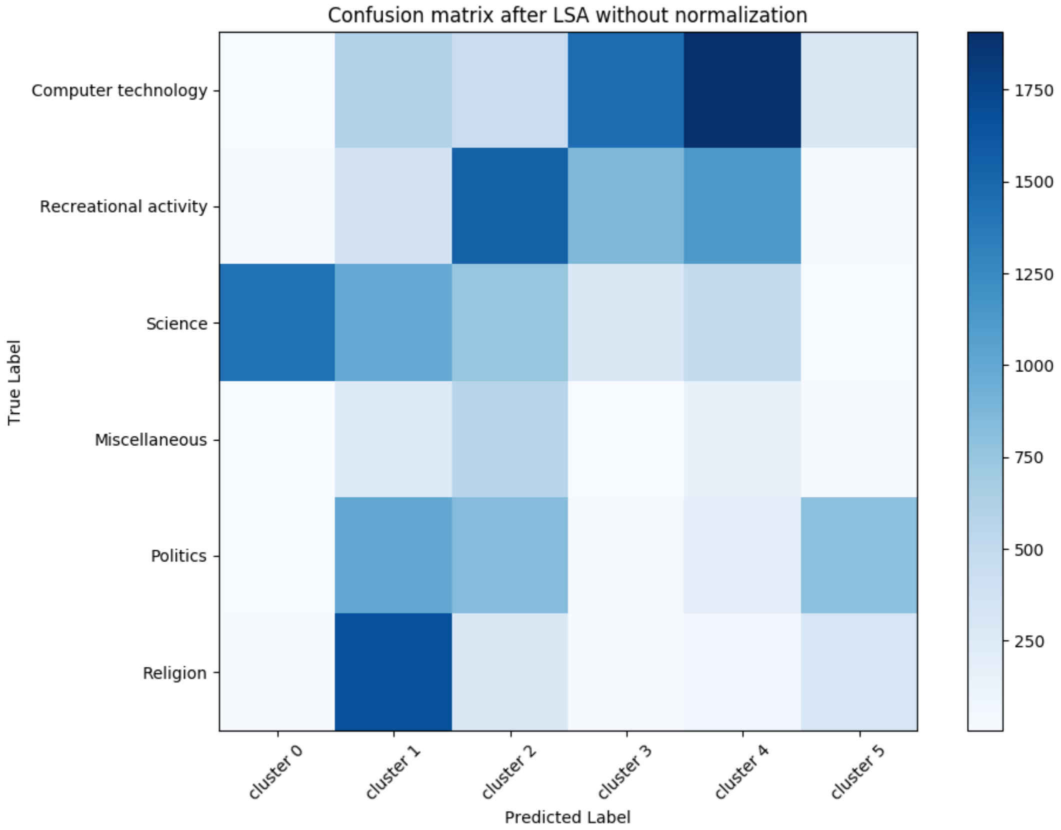
```
weikun@weikun:~/Desktop/Homework$ python problem6Part1.py

 EE 219 Project 4 Problem 6 Part 1
 Name: Weikun Han, Xiao Shi
 Date: 3/6/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 6
  - Reducing the Dimension with Truncated SVD (LSI) / PCA


Loading 20 newsgroups dataset for categories...
['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.w
indows.x', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.el
ectronics', 'sci.med', 'sci.space', 'misc.forsale', 'talk.politics.misc', 'talk.politics.guns', 'talk.pol
itics.mideast', 'talk.religion.misc', 'alt.atheism', 'soc.religion.christian']

Transforming the documents into TF-IDF vectors...

Performing dimensionality reduction using LSA without normalizing...

-------------------------Processing Finshed 1--------------------------
Performing dimensionality reduction done in 1.773571s
Total samples done: 18846, Total features done: 50
-----------------------------------------------------------------------

Performing dimensionality reduction using LSA with normalizing...

-------------------------Processing Finshed 2--------------------------
Performing dimensionality reduction done in 1.647437s
Total samples done: 18846, Total features done: 50
-----------------------------------------------------------------------
```

From the result, we can see the dimension is reduction to 50

Next, we use the result after LSA without normalizing to get the final data representation as follows:

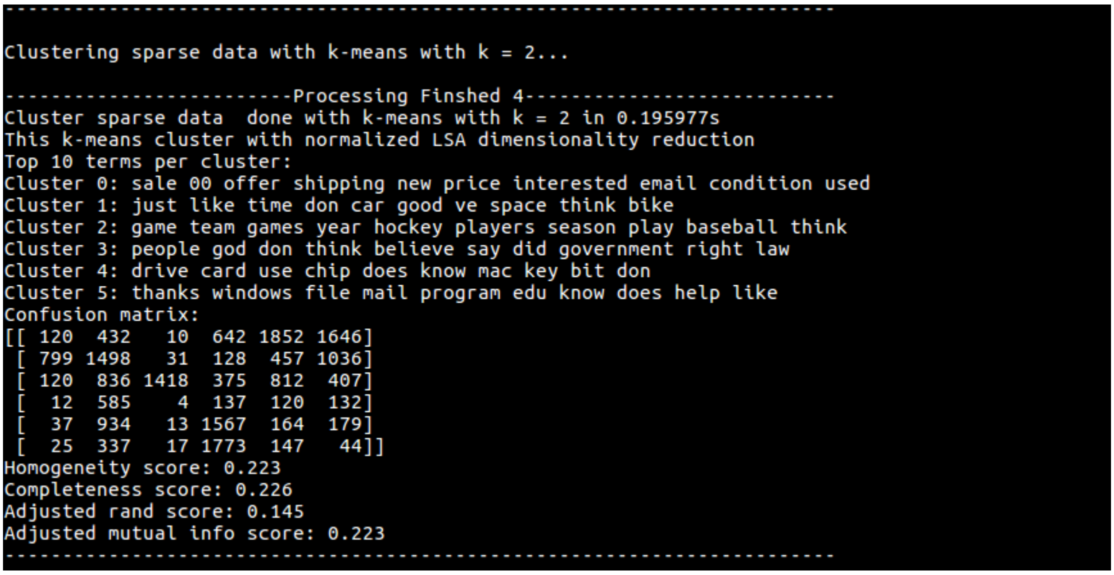


```
Clustering sparse data with k-means with k = 2...

-------------------------Processing Finshed 3--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.387685s
This k-means cluster with LSA dimensionality reduction (without normalizing)
Top 10 terms per cluster:
Cluster 0: game team games year hockey baseball ve season players play
Cluster 1: people don government think right just israel know law say
Cluster 2: like just think good car time ve don bike space
Cluster 3: drive card 00 use monitor mac new need pc disk
Cluster 4: thanks windows mail file does edu know program use window
Cluster 5: god jesus people think bible does believe christian say christ
Confusion matrix:
[[  11  594  438 1459 1908  292]
 [  33  348 1548  870 1128   22]
 [1430 1000  754  277  495   12]
 [   6  250  570    8  142   14]
 [  13 1012  838   41  196  794]
 [  14 1670  277   19   56  307]]
Homogeneity score: 0.217
Completeness score: 0.219
Adjusted rand score: 0.128
Adjusted mutual info score: 0.217
-----------------------------------------------------------------------
```

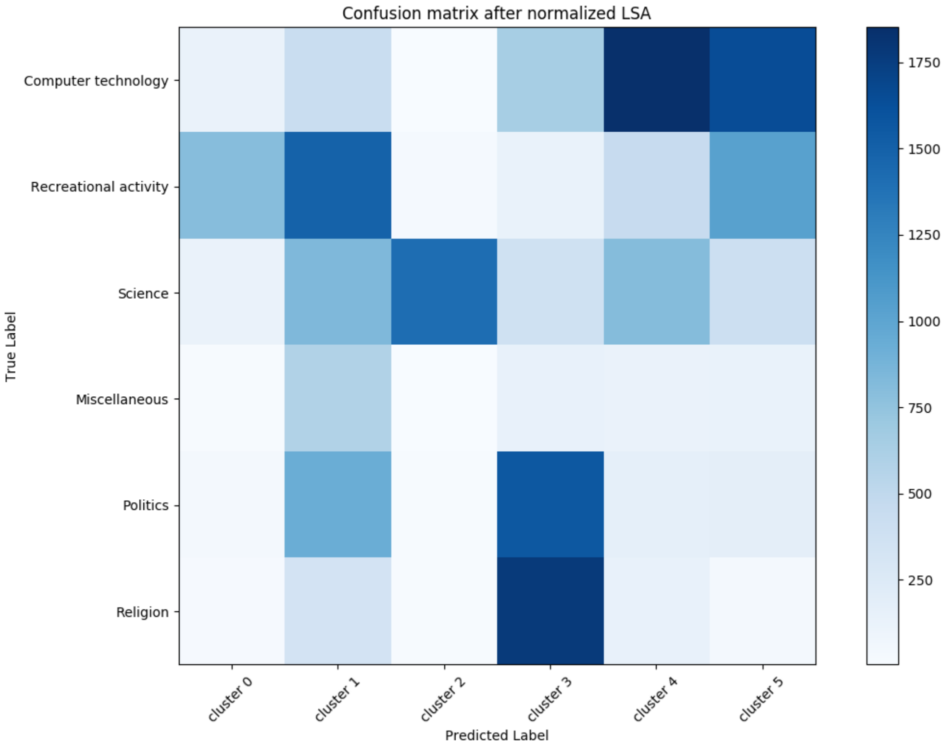


Then, to improve clustering purity, we are going to normalizing features (using normalized LAS)

And the final data representation as follows:



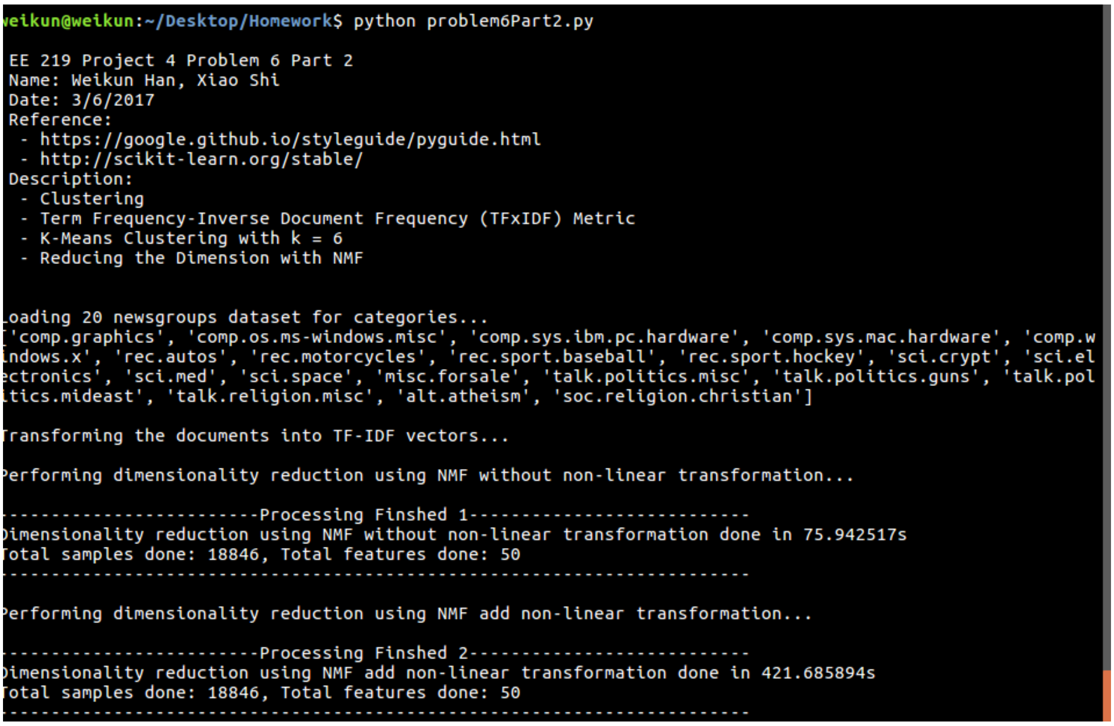
```
----------------------------------------------------------------------

Clustering sparse data with k-means with k = 2...

------------------------Processing Finshed 4--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.195977s
This k-means cluster with normalized LSA dimensionality reduction
Top 10 terms per cluster:
Cluster 0: sale 00 offer shipping new price interested email condition used
Cluster 1: just like time don car good ve space think bike
Cluster 2: game team games year hockey players season play baseball think
Cluster 3: people god don think believe say did government right law
Cluster 4: drive card use chip does know mac key bit don
Cluster 5: thanks windows file mail program edu know does help like
Confusion matrix:
[[ 120  432   10  642 1852 1646]
 [ 799 1498   31  128  457 1036]
 [ 120  836 1418  375  812  407]
 [  12  585    4  137  120  132]
 [  37  934   13 1567  164  179]
 [  25  337   17 1773  147   44]]
Homogeneity score: 0.223
Completeness score: 0.226
Adjusted rand score: 0.145
Adjusted mutual info score: 0.223
----------------------------------------------------------------------
```



Then we use the another decomposition method（NMF）), which have been built in model in sklearn:

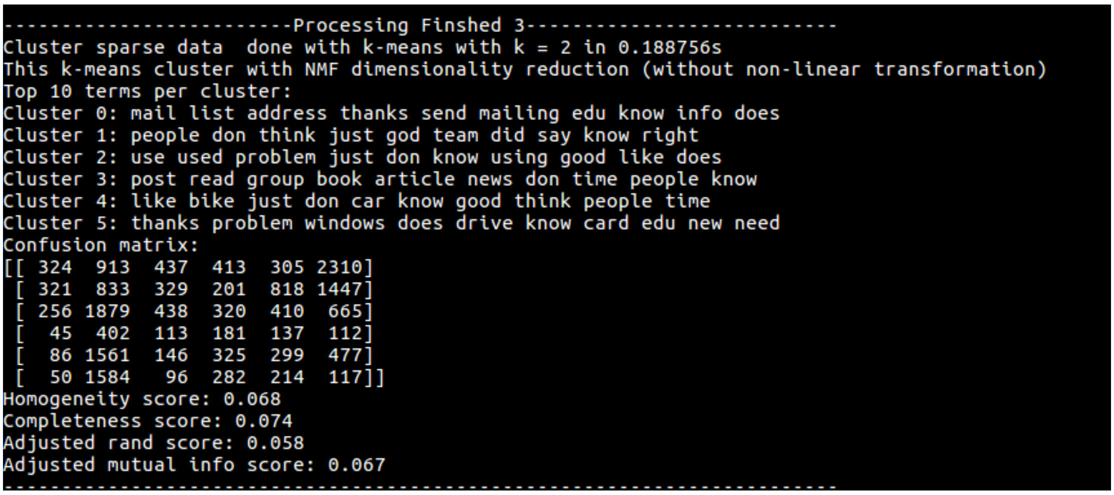
NMF(n_components = 50, random_state = 2)

Therefore, we get following results:

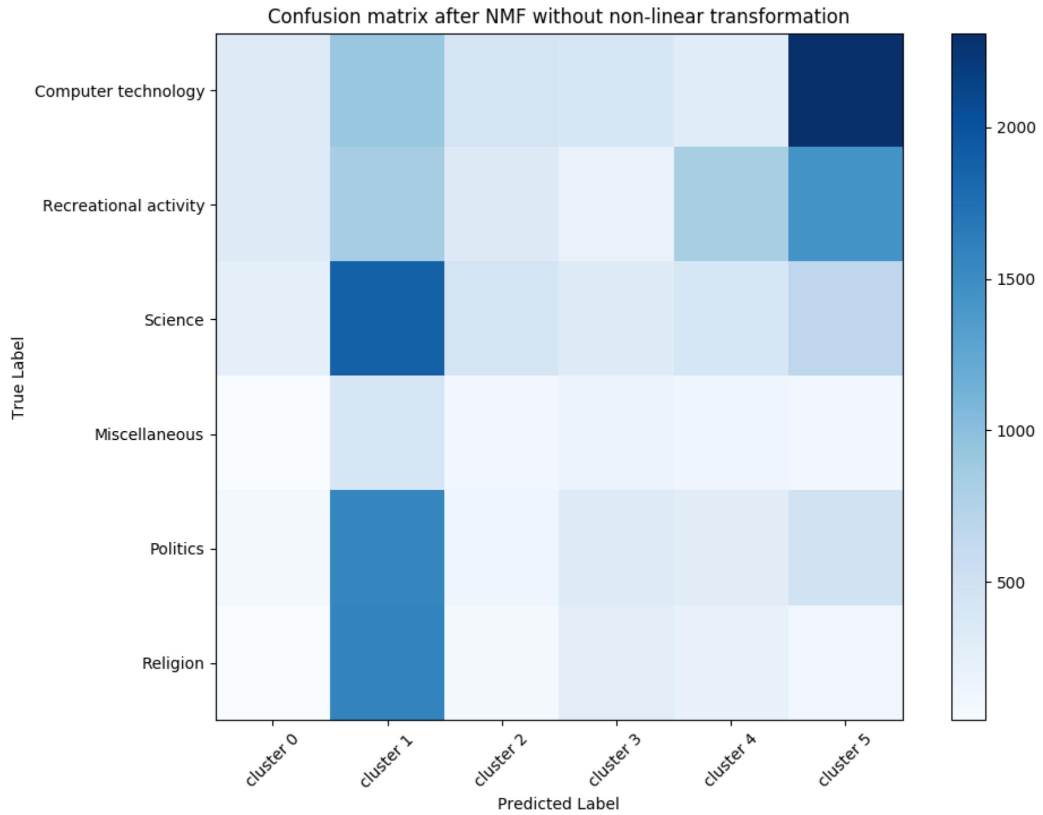


```
eikun@weikun:~/Desktop/Homework$ python problem6Part2.py

 EE 219 Project 4 Problem 6 Part 2
 Name: Weikun Han, Xiao Shi
 Date: 3/6/2017
 Reference:
  - https://google.github.io/styleguide/pyguide.html
  - http://scikit-learn.org/stable/
 Description:
  - Clustering
  - Term Frequency-Inverse Document Frequency (TFxIDF) Metric
  - K-Means Clustering with k = 6
  - Reducing the Dimension with NMF


oading 20 newsgroups dataset for categories...
'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.w
indows.x', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.el
ectronics', 'sci.med', 'sci.space', 'misc.forsale', 'talk.politics.misc', 'talk.politics.guns', 'talk.pol
itics.mideast', 'talk.religion.misc', 'alt.atheism', 'soc.religion.christian']

ransforming the documents into TF-IDF vectors...

erforming dimensionality reduction using NMF without non-linear transformation...

------------------------Processing Finshed 1--------------------------
imensionality reduction using NMF without non-linear transformation done in 75.942517s
otal samples done: 18846, Total features done: 50
----------------------------------------------------------------------

erforming dimensionality reduction using NMF add non-linear transformation...

------------------------Processing Finshed 2--------------------------
imensionality reduction using NMF add non-linear transformation done in 421.685894s
otal samples done: 18846, Total features done: 50
----------------------------------------------------------------------
```

Next, we use the result after NMF without adding nonlinear transformation to get the final data representation as follows:

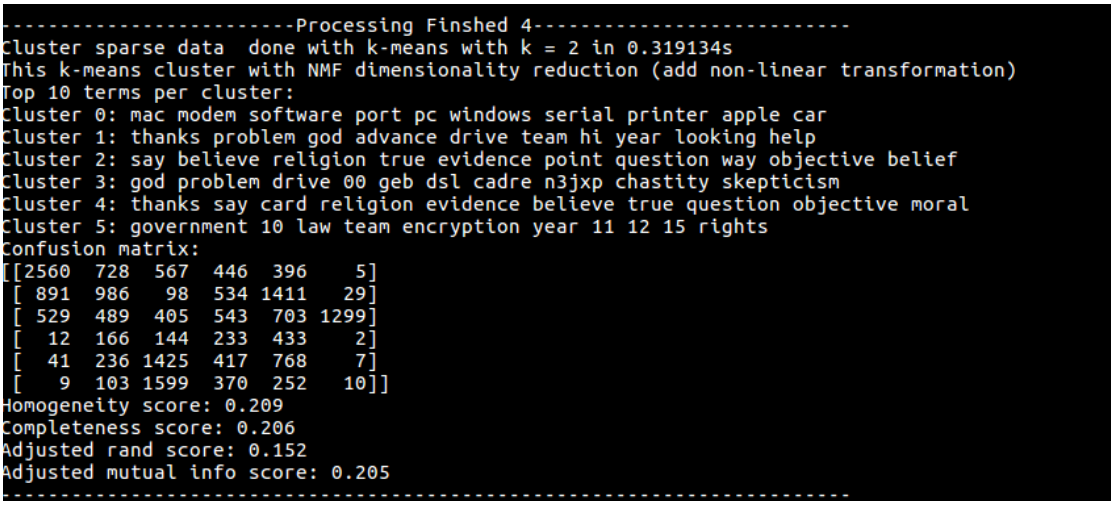


```
------------------------Processing Finshed 3--------------------------
Cluster sparse data  done with k-means with k = 2 in 0.188756s
This k-means cluster with NMF dimensionality reduction (without non-linear transformation)
Top 10 terms per cluster:
Cluster 0: mail list address thanks send mailing edu know info does
Cluster 1: people don think just god team did say know right
Cluster 2: use used problem just don know using good like does
Cluster 3: post read group book article news don time people know
Cluster 4: like bike just don car know good think people time
Cluster 5: thanks problem windows does drive know card edu new need
Confusion matrix:
[[ 324  913  437  413  305 2310]
 [ 321  833  329  201  818 1447]
 [ 256 1879  438  320  410  665]
 [  45  402  113  181  137  112]
 [  86 1561  146  325  299  477]
 [  50 1584   96  282  214  117]]
Homogeneity score: 0.068
Completeness score: 0.074
Adjusted rand score: 0.058
Adjusted mutual info score: 0.067
----------------------------------------------------------------------
```

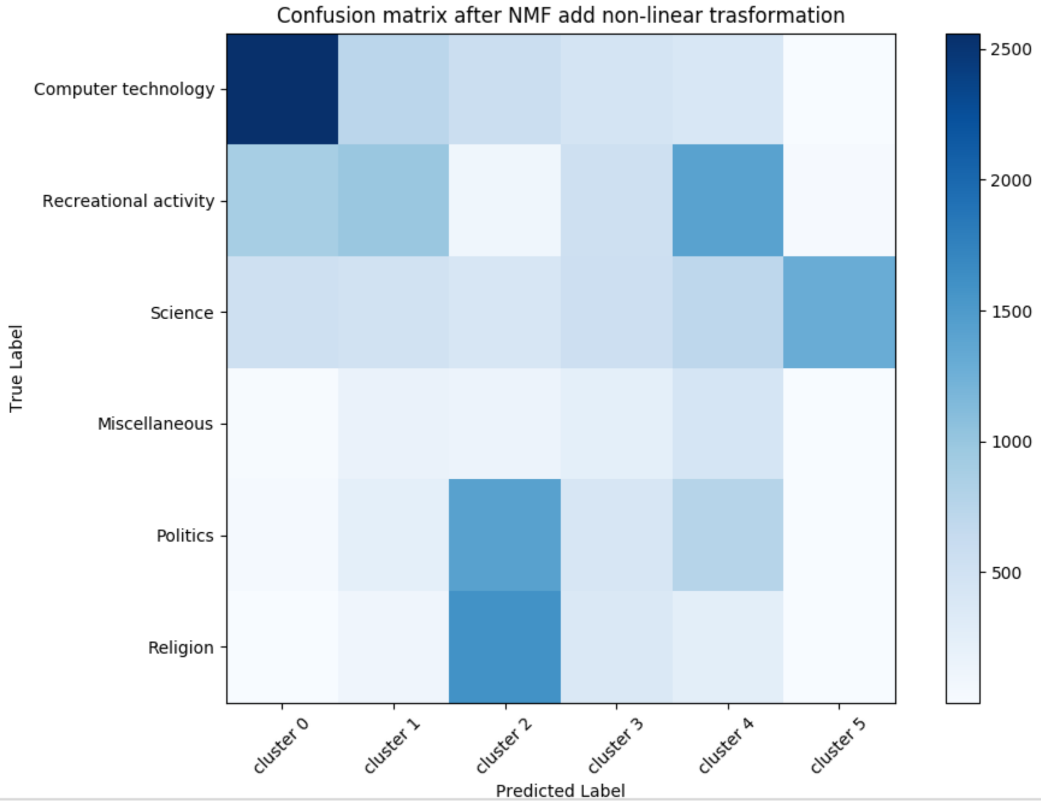


Finally, we add nonlinear transformation

And the final data representation as follows:



```
-------------------------Processing Finshed 4---------------------------
Cluster sparse data  done with k-means with k = 2 in 0.319134s
This k-means cluster with NMF dimensionality reduction (add non-linear transformation)
Top 10 terms per cluster:
Cluster 0: mac modem software port pc windows serial printer apple car
Cluster 1: thanks problem god advance drive team hi year looking help
Cluster 2: say believe religion true evidence point question way objective belief
Cluster 3: god problem drive 00 geb dsl cadre n3jxp chastity skepticism
Cluster 4: thanks say card religion evidence believe true question objective moral
Cluster 5: government 10 law team encryption year 11 12 15 rights
Confusion matrix:
[[2560  728  567  446  396    5]
 [ 891  986   98  534 1411   29]
 [ 529  489  405  543  703 1299]
 [  12  166  144  233  433    2]
 [  41  236 1425  417  768    7]
 [   9  103 1599  370  252   10]]
Homogeneity score: 0.209
Completeness score: 0.206
Adjusted rand score: 0.152
Adjusted mutual info score: 0.205
------------------------------------------------------------------------
```



We can see that the confusion matrix is almost diagonal. And we also got pretty good result in other scores.