Project 2： Clustering

In this project, we are still working with the “20 Newsgroups” dataset and performing K-means clustering on this dataset. By extracting significant features(TF_IDF matrix) from documents, applying dimension reduction techniques(LSI and NMF) with different preprocessing tools(normalization and logrithm) to high-dimensional feature vectors, and finally classifitying them into clusters. By visualizing the clustering results, various measures of purity and the contigency matrix, we are able to evaluate the performance of k-means clustering and get a better understanding of how dimension reduction and preprocessing can affect the clustering results.

In [1]:
```
import numpy as np
import sklearn
import nltk
import matplotlib.pyplot as plt
%matplotlib inline
```

Problem 1

- Load the dataset for 8 sub-classes:

  - Computer Technology: 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware'
  - Recreational Activity: 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey'

- Build the TF-IDF matrix:

  - After excluding the stop words, we still use *CountVectorizer* to get the term-document matrix under conditin min_df = 3, and use *TfidfTransformer* to get the TF-IDF feature matrix
  - The size of TF-IDF matrix is **(7882, 27743)**

In [2]:
```
from sklearn.datasets import fetch_20newsgroups

categories = ['comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', \
              'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']

# Raw data
X_8 = fetch_20newsgroups(subset='all', categories=categories, shuffle=True, random_state=42)
data_8 = X_8.data
label_8 = (X_8.target > 3).astype(int)
```

In [3]:
```
from sklearn.feature_extraction import text
from nltk.corpus import stopwords
from string import punctuation

# Create our own stopwords set
stop_words_skt = text.ENGLISH_STOP_WORDS
stop_words_en = stopwords.words('english')
combined_stopwords = set.union(set(stop_words_en), set(punctuation), set(stop_
words_skt))
```

In [4]:
```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

# Get the term-document matrix under the condition: min_df=3
count_vect_3 = CountVectorizer(min_df=3, stop_words = combined_stopwords)

X_8_counts = count_vect_3.fit_transform(data_8)

# Get the TF_IDF feature matrix
tfidf_transformer = TfidfTransformer()

X_8_tfidf = tfidf_transformer.fit_transform(X_8_counts)
print(X_8_tfidf.shape)
```

```
(7882, 27743)
```

Problem 2

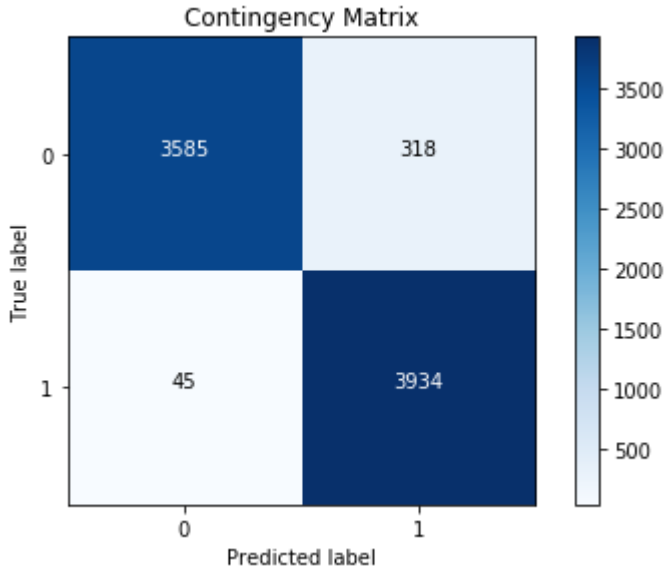
In this part, we will inspect the contigency matrices and various measures of purity to get a sense of clustering results. Here, we use k = 2 for K-means clustering for the TF-IDF data. To make it easy for future use of plotting the contigency matrix and scores, we make them into two functions *print_plot_cnf_matrix* and *print_scores*.

(a) Inspect the contigency matrix

By calling *print_plot_cnf_matrix*, we print and show the confusion matrix in the heat-map. In can be seen that nearly all the data are clustered into either the left-upper block or the right-lower block, which indicates that they are nearly clustered into two categories while just a small amount of data is classified incorrectly. This makes sense since the data we use here are from two categories "Computer Technology" and "Recreational Activity".

In [5]:
```
from sklearn.cluster import KMeans
from sklearn import metrics
import itertools
km = KMeans(n_clusters = 2, max_iter = 300, random_state = 42, n_init = 1).fit
(X_8_tfidf)
cnf_matrix = metrics.confusion_matrix(label_8, km.labels_)


def print_plot_cnf_matrix(cm, classes=[0,1]):
    """
    This function prints and plots the contingency matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.figure()
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title('Contingency Matrix')
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes)
    plt.yticks(tick_marks, classes)
    print('Contingency matrix')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

    np.set_printoptions(precision=2)

print_plot_cnf_matrix(cnf_matrix)
```

```
Contingency matrix
[[3585  318]
 [  45 3934]]
```



(b) Report the 5 measures for the K-means clustering results

Here, k = 2 for K-means clustering and we use the homogeneity score, the completeness score, the V-measure score, the adjusted Rand score and the adjusted mutual infor score to evaluate the K-means clustering performance.

Homogeneity Score	Completeness Score	V-measure Score	Adjusted Rand Score	Adjusted Mutual Info Score
0.749	0.752	0.750	0.824	0.519

- Homogeneity Score: A measure of how "pure" the clusters are. If each cluster contains only data points from a single class, the homogeneity is satisfied. The higher the homogeneity score, the purer are the clustering results.
- Completeness Score: A clustering result satisfies completeness if all data points of a class are assigned to the same cluster. Both of these scores span between 0 and 1; where 1 stands for perfect clustering.
- V-measure score: The harmonic average of homogeneity score and completeness score.
- Adjusted Rand score: The adjusted Rand Index is similar to accuracy measure, which computes similarity between the clustering labels and groud truth labels. This method counts all pairs of points that both fall either in the same cluster and the same class or in different clusters and different classes.
- Ajusted Mutual Info score: It measures the mutual information between the cluster label distribution and the ground truth label distributions.

All these scores should fall in the range from 0 to 1. Since all of them are approaching 1, it is believed that k=2 is a reasonable number of clusters and this agrees with data that they are from two categories.

In [6]:
```
from sklearn import metrics
def print_scores(true_label, predict_label):
    print("Homogeneity Score: %0.3f" % metrics.homogeneity_score(true_label, p
redict_label))
    print("Completeness Score: %0.3f" % metrics.completeness_score(true_label,
 predict_label))
    print("V-measure Score: %0.3f" % metrics.v_measure_score(true_label, predi
ct_label))
    print("Adjusted Rand Score: %.3f"
          % metrics.adjusted_rand_score(true_label, predict_label))
    print("Adjusted Mutual Info Score: %0.3f"
          % metrics.mutual_info_score(true_label, predict_label, contingency =
 None))

print_scores(label_8, km.labels_)
```

```
Homogeneity Score: 0.749
Completeness Score: 0.752
V-measure Score: 0.750
Adjusted Rand Score: 0.824
Adjusted Mutual Info Score: 0.519
```

Problem 3 Dimensionality reduction

i. Report the plot the percent of the top r components can retain

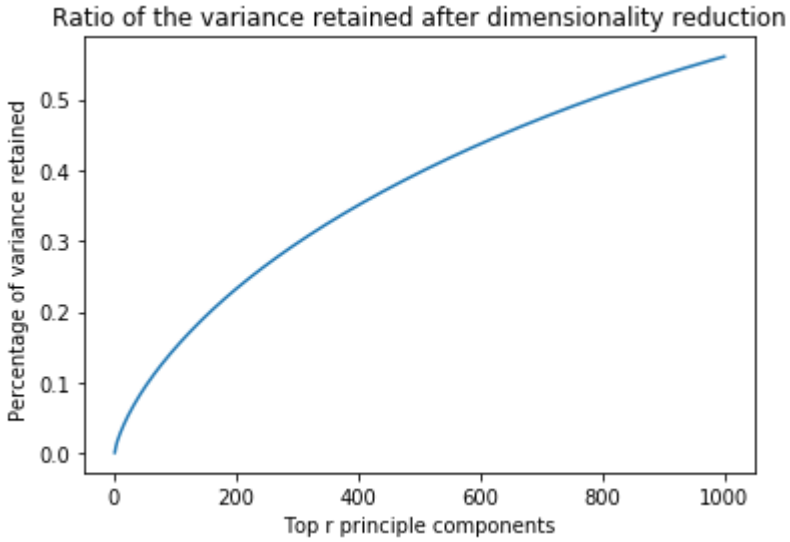
In this question, we firstly perform SVD with r = 1000 that we will find the top 1000 important components so that we can find the ratio of the variance of particular r can retain.

$$ratio[r] = \frac{variance\ retained\ by\ r^{th}\ component}{total\ variance}$$

Then, to find the variance of top r components that could retain, we can just simple exclude the least important features.

From our plot, it can be seen that the percentage of variance retained is increasing with r. It is because that by increasing r, we are including more and more dimensions, so that the reconstructed matrix with truncated SVD will be more alike the original TF-IDF matrix. Therefore, there should be a trend that the ratio of variance retained is increasing.

In [7]:
```
# LSI:

from sklearn.decomposition import TruncatedSVD, NMF
from sklearn.random_projection import sparse_random_matrix

var_ratio = np.zeros(1000)
svd = TruncatedSVD(n_components = 1000, random_state = 42)
svd.fit_transform(X_8_tfidf)

for i in np.arange(1, 1001):
    var_ratio[i-1] = svd.explained_variance_ratio_[:i].sum()
# print(var_ratio)

plt.figure()
plt.plot(np.arange(1,1001), var_ratio)
plt.xlabel('Top r principle components')
plt.ylabel('Percentage of variance retained ')
plt.title('Ratio of the variance retained after dimensionality reduction')
plt.show()
```



ii.

Sweep over the dimension parameters for each method (Truncated SVD(LSI)/PCA and NMF), and choose the one that yields better results in terms of clustering purity metrics.

> To reduce the redundancy of codes, we calculate the five measure scores based on the predicted labels and the groud truth within the function *save_scores* and since we need to choose the best r for future use, this function will return a list of homogeneity score.

- **Truncated SVD (LSI):**

r	Homogeneity score	Completeness score	V-measure score	Adjusted Rand score	Adjusted Mutual Info score	Contigency matrix
1	0.001	0.001	0.001	0.001	0.000	[[2114,1789], [2276,1703]]
2	0.580	0.581	0.581	0.680	0.402	[[3694,209], [481,3498]]
3	0.398	0.437	0.417	0.417	0.417	[[3869,34], [1444,2535]]
5	0.224	0.312	0.260	0.417	0.260	[[3898,5], [2422,1557]]
10	0.236	0.322	0.273	0.273	0.164	[[3900,3], [2364,1615]]
20	0.236	0.322	0.272	0.159	0.163	[[3900,3], [2365,1614]]
50	0.001	0.009	0.001	-0.000	0.000	[[3880,23], [3933,46]]
100	0.009	0.122	0.016	0.001	0.006	[[3837,66], [3979,0]]
300	0.776	0.778	0.777	0.855	0.538	[[229,3674], [3912,67]]

> From the above chart, we can see that the best purity occurs when **r = 300 for LSI**. As for heat maps of these contigency matrices, please refer to the results below that is directly printed from LSI codes. Except for r = 300, r = 2 also gives satisfactory scores, which agrees that there are two categories. While since the non-monotic performance of r, r = 300 is also possible

- **NMF:**

r	Homogeneity score	Completeness score	V-measure score	Adjusted Rand score	Adjusted Mutual Info score	Contigency matrix

r	Homogeneity score	Completeness score	V-measure score	Adjusted Rand score	Adjusted Mutual Info score	Contigency matrix
1	0.001	0.001	0.001	0.001	0.000	[[2114,1789], [2276 & 1703]]
2	0.676	0.677	0.676	0.774	0.468	[[3694,316], [158 & 3821]]
3	0.229	0.316	0.265	0.153	0.159	[[3898,5], [2392 & 1587]]
5	0.342	0.351	0.347	0.408	0.237	[[3586,317], [1106 & 2873]]
10	0.027	0.155	0.046	0.004	0.019	[[3697,206], [3979 & 0]]
20	0.006	0.038	0.011	0.000	0.004	[[3863,40], [3827 & 152]]
50	0.003	0.077	0.007	-0.000	0.002	[[3901,2], [3942 & 37]]
100	0.002	0.024	0.003	0.000	0.001	[[46,3857], [15 & 3964]]
300	0.005	0.107	0.009	-0.000	0.003	[[3903,0], [3943 & 36]]

From the above chart, we can see that the best purity occurs when **r = 2 for NMF**. As for heat maps of these contigency matrices, please refer to the results below that is directly printed from NMF codes. This r = 2 is consistent with our expectation that no. of clusters should be equal to no. of categories.

Question: How do you explain the non-monotonic behavior of the measures as r increases?

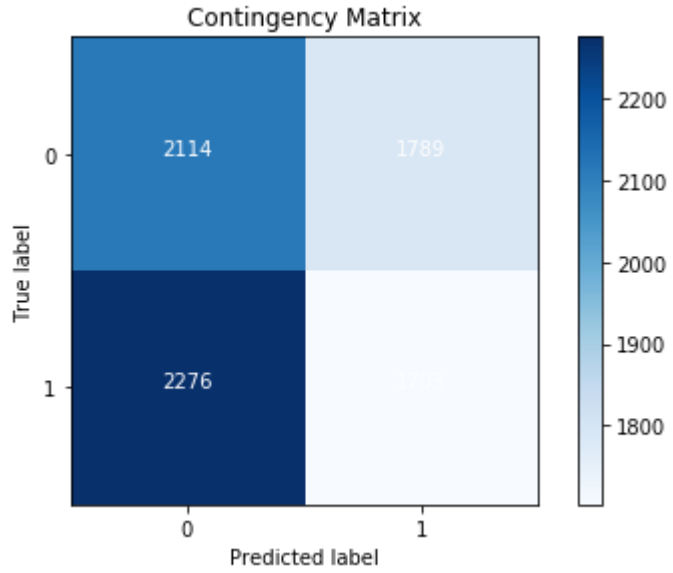
Ans: When performing Truncated SVD on to the data, we are looking for the top r main components that could represent the original data. These r main components are searched based on their large variance. To be concretely, the top 1 component has the largest variance, and the data is separated most clearly along this dimension. By using the top r components, we are clustering data based on these r dimensions. However, if we use too many components, the relatively useless components will act just as noise in the clustering process. Therefore, there exists a non-monotonic behaviror as r increases.

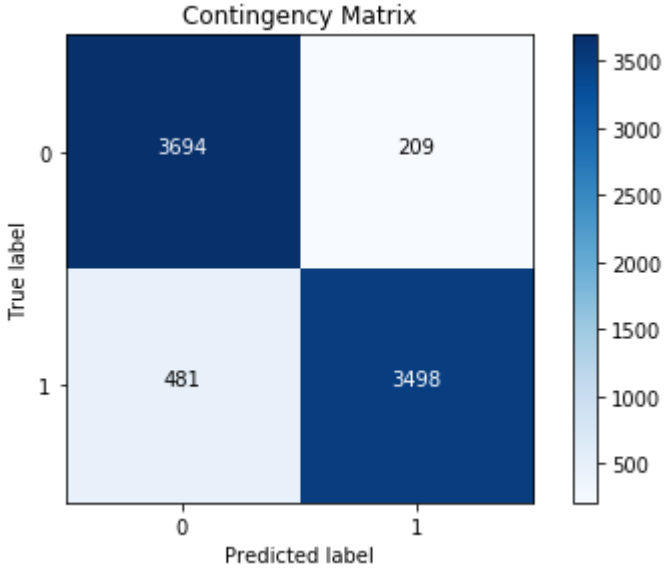
In [8]:
```
def save_scores(true_label, predict_label, homo, comp, vmeas, adjrs, mutis):
    homo.append(metrics.homogeneity_score(true_label, predict_label))
    comp.append(metrics.completeness_score(true_label, predict_label))
    vmeas.append(metrics.v_measure_score(true_label, predict_label))
    adjrs.append(metrics.adjusted_rand_score(true_label, predict_label))
    mutis.append(metrics.mutual_info_score(true_label, predict_label))

    return homo
```

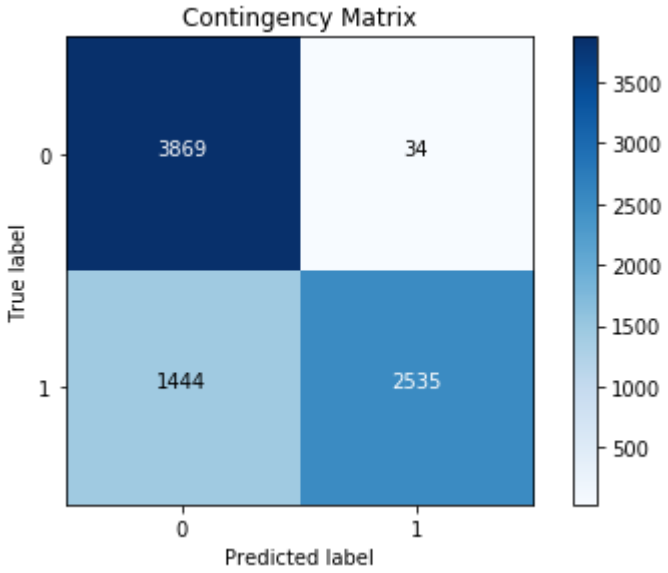
In [9]:
```
# perform with LSI
r = [1,2,3,5,10,20,50,100,300]
homo_lsi = []
comp_lsi = []
vmeas_lsi = []
adjrs_lsi = []
mutis_lsi = []
for each_r in r:
    svd = TruncatedSVD(n_components = each_r, random_state = 42)
    svd_reduced = svd.fit_transform(X_8_tfidf)
    km = KMeans(n_clusters = 2, max_iter = 500, random_state = 42, n_init = 1)
.fit(svd_reduced)
    print('-'*40)
    print('When r is', each_r, ', Contigency matrix is: ')
    cnf_matrix = metrics.confusion_matrix(label_8, km.labels_)
    print_plot_cnf_matrix(cnf_matrix)
    pass
    print_scores(label_8, km.labels_)
    lsi_homo = save_scores(label_8, km.labels_, homo_lsi, comp_lsi, vmeas_lsi,
 adjrs_lsi, mutis_lsi)

plt.figure()
plt.plot(r, homo_lsi, 'b', label = 'Homogeneity Score')
plt.plot(r, comp_lsi, 'g', label = 'Completeness Score')
plt.plot(r, vmeas_lsi, 'r', label = 'V-measure Score')
plt.plot(r, adjrs_lsi, 'c', label = 'Adjusted Rand Score')
plt.plot(r, mutis_lsi, 'm', label = 'Adjusted Mutual Info Score')
plt.xlabel('Top r principle components')
plt.ylabel('Scores')
plt.legend(loc = 'upper right')
plt.title('Measures of Purity')
plt.show()
```

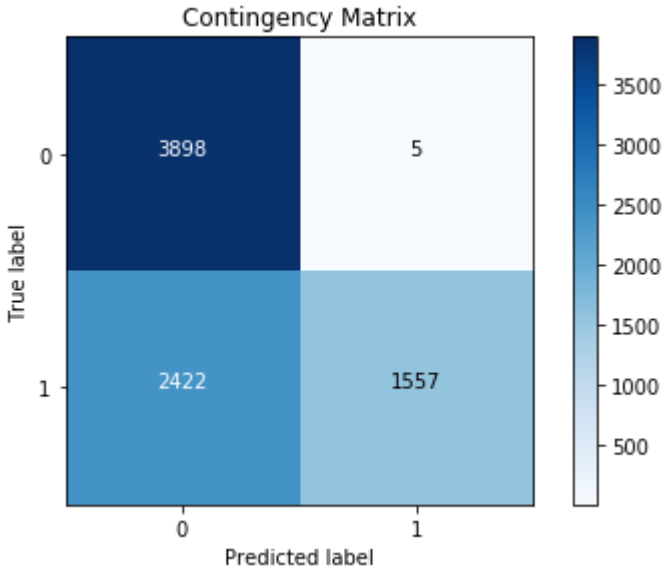
```
----------------------------------------
When r is 1 , Contigency matrix is:
Contingency matrix
[[2114 1789]
 [2276 1703]]
Homogeneity Score: 0.001
Completeness Score: 0.001
V-measure Score: 0.001
Adjusted Rand Score: 0.001
Adjusted Mutual Info Score: 0.000
----------------------------------------
When r is 2 , Contigency matrix is:
Contingency matrix
[[3694  209]
 [ 481 3498]]
Homogeneity Score: 0.580
Completeness Score: 0.581
V-measure Score: 0.581
Adjusted Rand Score: 0.680
Adjusted Mutual Info Score: 0.402
----------------------------------------
When r is 3 , Contigency matrix is:
Contingency matrix
[[3869   34]
 [1444 2535]]
Homogeneity Score: 0.398
Completeness Score: 0.437
V-measure Score: 0.417
Adjusted Rand Score: 0.391
Adjusted Mutual Info Score: 0.276
----------------------------------------
When r is 5 , Contigency matrix is:
Contingency matrix
[[3898    5]
 [2422 1557]]
Homogeneity Score: 0.224
Completeness Score: 0.312
V-measure Score: 0.260
Adjusted Rand Score: 0.147
Adjusted Mutual Info Score: 0.155
----------------------------------------
When r is 10 , Contigency matrix is:
Contingency matrix
[[3900    3]
 [2364 1615]]
Homogeneity Score: 0.236
Completeness Score: 0.322
V-measure Score: 0.273
Adjusted Rand Score: 0.159
Adjusted Mutual Info Score: 0.164
----------------------------------------
When r is 20 , Contigency matrix is:
Contingency matrix
[[3900    3]
 [2365 1614]]
Homogeneity Score: 0.236
Completeness Score: 0.322
```

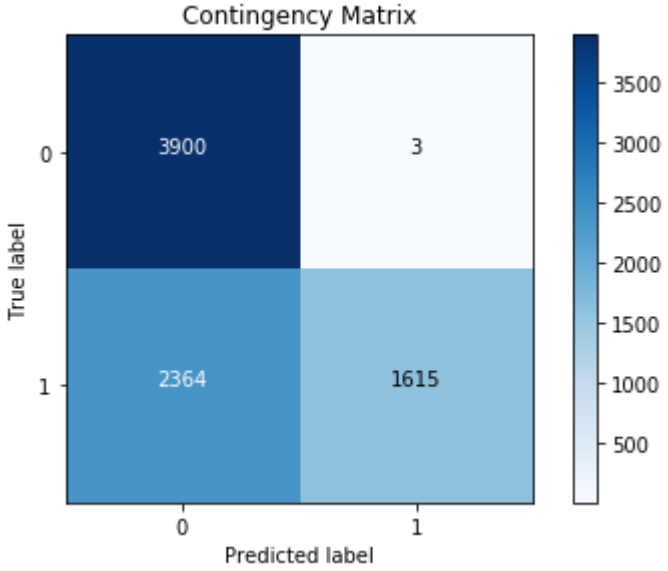
```
V-measure Score: 0.272
Adjusted Rand Score: 0.159
Adjusted Mutual Info Score: 0.163
----------------------------------------
When r is 50 , Contigency matrix is:
Contingency matrix
[[3880   23]
 [3933   46]]
Homogeneity Score: 0.001
Completeness Score: 0.009
V-measure Score: 0.001
Adjusted Rand Score: -0.000
Adjusted Mutual Info Score: 0.000
----------------------------------------
When r is 100 , Contigency matrix is:
Contingency matrix
[[3837   66]
 [3979    0]]
Homogeneity Score: 0.009
Completeness Score: 0.122
V-measure Score: 0.016
Adjusted Rand Score: 0.001
Adjusted Mutual Info Score: 0.006
----------------------------------------
When r is 300 , Contigency matrix is:
Contingency matrix
[[ 229 3674]
 [3912   67]]
Homogeneity Score: 0.776
Completeness Score: 0.778
V-measure Score: 0.777
Adjusted Rand Score: 0.855
Adjusted Mutual Info Score: 0.538
```

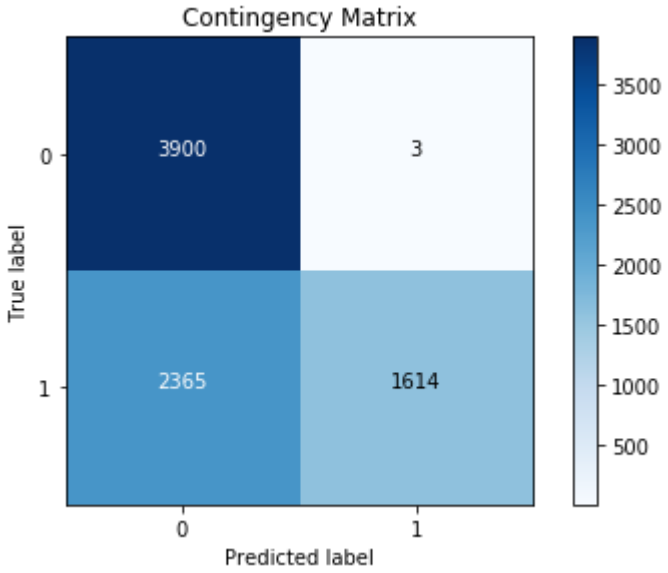


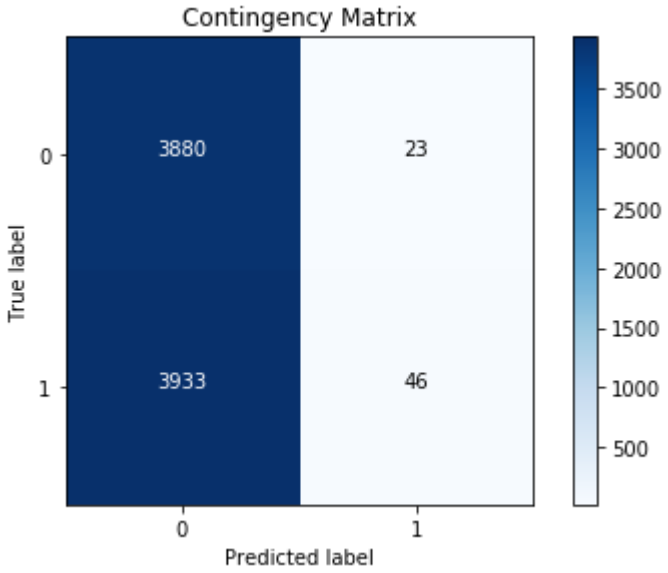

Contingency Matrix
3694
209
481
3498
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

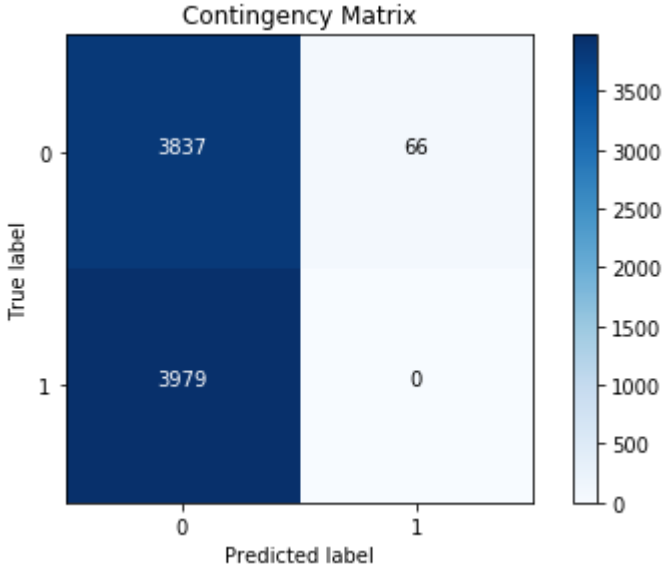

Contingency Matrix
3869
34
1444
2535
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

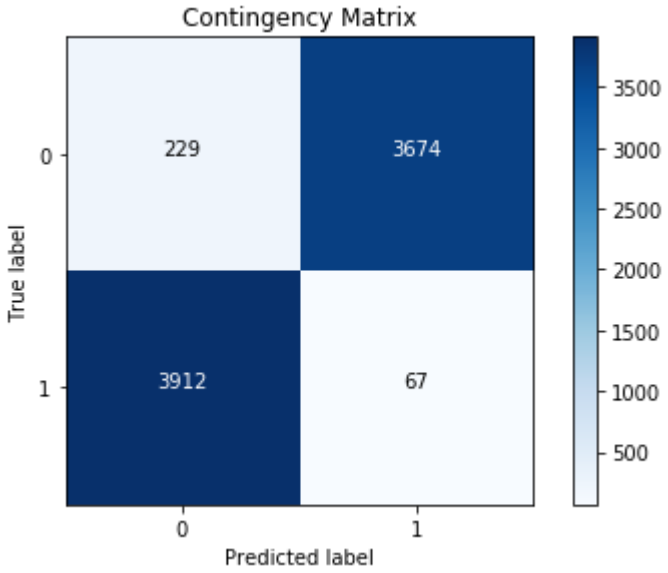

Contingency Matrix
3898
5
2422
1557
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

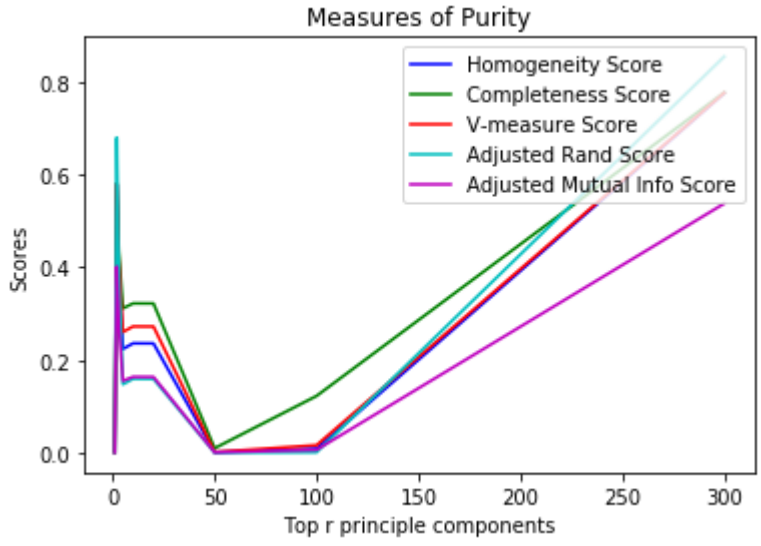

Contingency Matrix
3900
3
2364
1615
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
3900
3
2365
1614
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
3880
23
3933
46
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
3837
66
3979
0
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

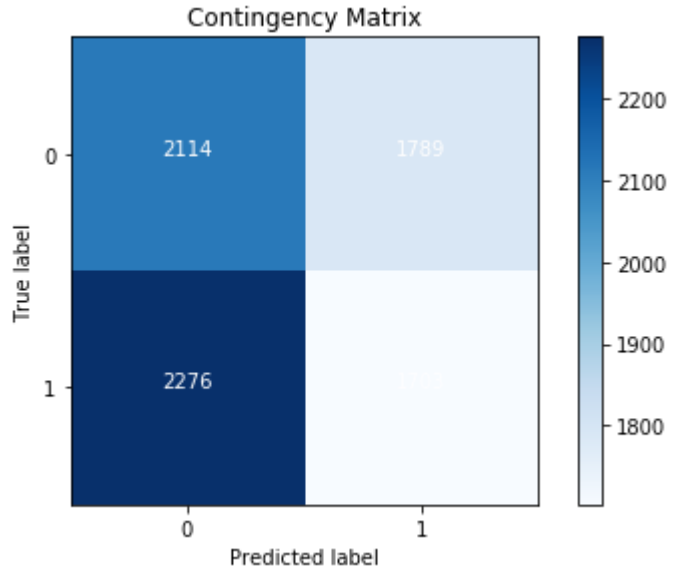

Contingency Matrix
229
3674
3912
67
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

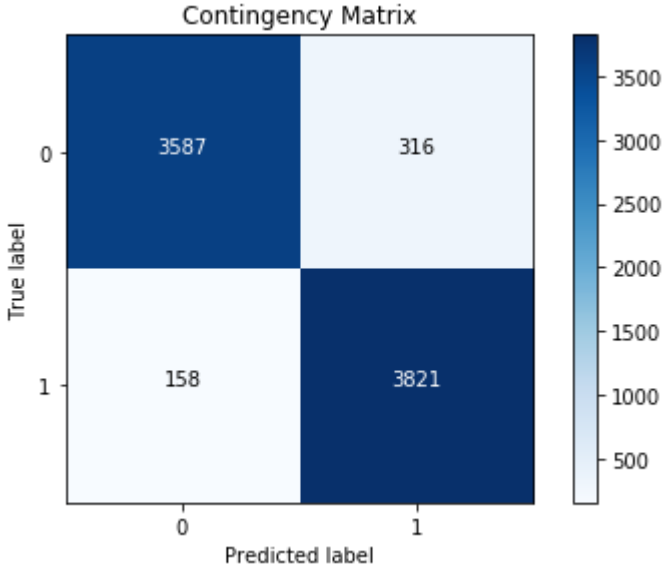

Measures of Purity
Homogeneity Score
Completeness Score
V-measure Score
Adjusted Rand Score
Adjusted Mutual Info Score
Scores
Top r principle components
0.0
0.2
0.4
0.6
0.8
0
50
100
150
200
250
300

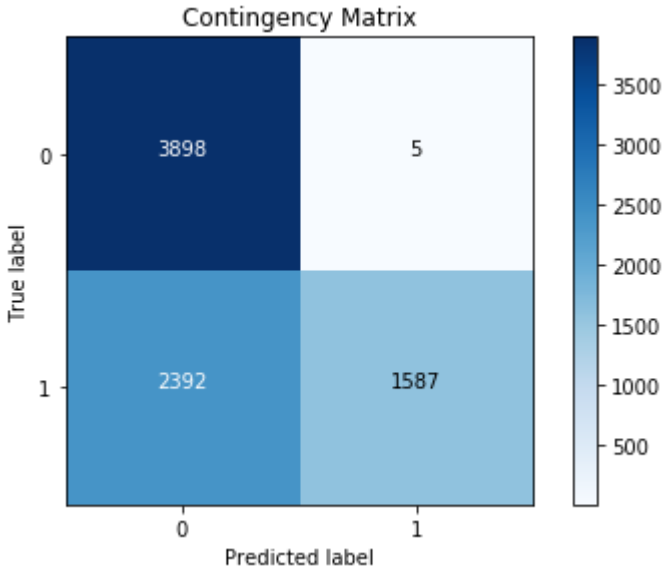
In [10]:
```
# perform with NMF
r = [1,2,3,5,10,20,50,100,300]
homo_nmf = []
comp_nmf = []
vmeas_nmf = []
adjrs_nmf = []
mutis_nmf = []
for each_r in r:
    nmf = NMF(n_components = each_r, random_state = 42)
    nmf_reduced = nmf.fit_transform(X_8_tfidf)
    km = KMeans(n_clusters = 2, max_iter = 500, random_state = 42, n_init = 1)
.fit(nmf_reduced)
    print('-'*40)
    print('When r is', each_r, ', Contigency matrix is: ')
    cnf_matrix = metrics.confusion_matrix(label_8, km.labels_)
    print_plot_cnf_matrix(cnf_matrix)

    print_scores(label_8, km.labels_)
    nmf_homo = save_scores(label_8, km.labels_, homo_nmf, comp_nmf, vmeas_nmf,
 adjrs_nmf, mutis_nmf)

plt.figure()
plt.plot(r, homo_nmf, 'b', label = 'Homogeneity Score')
plt.plot(r, comp_nmf, 'g', label = 'Completeness Score')
plt.plot(r, vmeas_nmf, 'r', label = 'V-measure Score')
plt.plot(r, adjrs_nmf, 'c', label = 'Adjusted Rand Score')
plt.plot(r, mutis_nmf, 'm', label = 'Adjusted Mutual Info Score')
plt.xlabel('Top r principle components')
plt.ylabel('Scores')
plt.legend(loc = 'upper right')
plt.title('Measures of Purity')
plt.show()
```

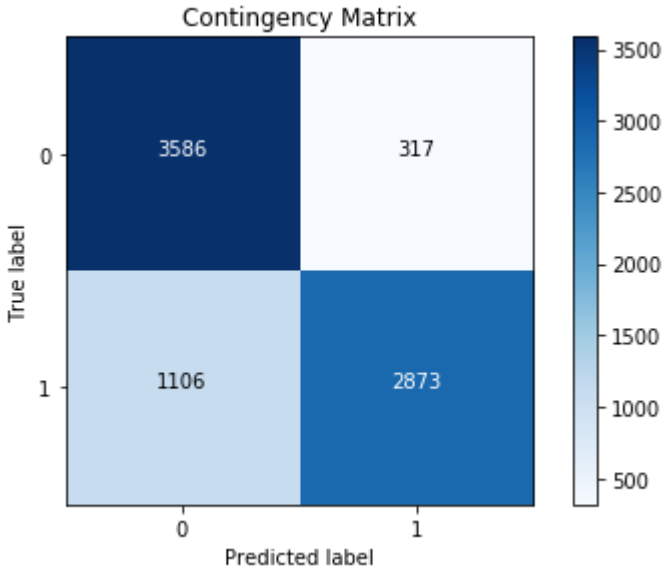
```
----------------------------------------
When r is 1 , Contigency matrix is:
Contingency matrix
[[2114 1789]
 [2276 1703]]
Homogeneity Score: 0.001
Completeness Score: 0.001
V-measure Score: 0.001
Adjusted Rand Score: 0.001
Adjusted Mutual Info Score: 0.000
----------------------------------------
When r is 2 , Contigency matrix is:
Contingency matrix
[[3587  316]
 [ 158 3821]]
Homogeneity Score: 0.676
Completeness Score: 0.677
V-measure Score: 0.676
Adjusted Rand Score: 0.774
Adjusted Mutual Info Score: 0.468
----------------------------------------
When r is 3 , Contigency matrix is:
Contingency matrix
[[3898    5]
 [2392 1587]]
Homogeneity Score: 0.229
Completeness Score: 0.316
V-measure Score: 0.265
Adjusted Rand Score: 0.153
Adjusted Mutual Info Score: 0.159
----------------------------------------
When r is 5 , Contigency matrix is:
Contingency matrix
[[3586  317]
 [1106 2873]]
Homogeneity Score: 0.342
Completeness Score: 0.351
V-measure Score: 0.347
Adjusted Rand Score: 0.408
Adjusted Mutual Info Score: 0.237
----------------------------------------
When r is 10 , Contigency matrix is:
Contingency matrix
[[3697  206]
 [3979    0]]
Homogeneity Score: 0.027
Completeness Score: 0.155
V-measure Score: 0.046
Adjusted Rand Score: 0.004
Adjusted Mutual Info Score: 0.019
----------------------------------------
When r is 20 , Contigency matrix is:
Contingency matrix
[[3863   40]
 [3827  152]]
Homogeneity Score: 0.006
Completeness Score: 0.038
```

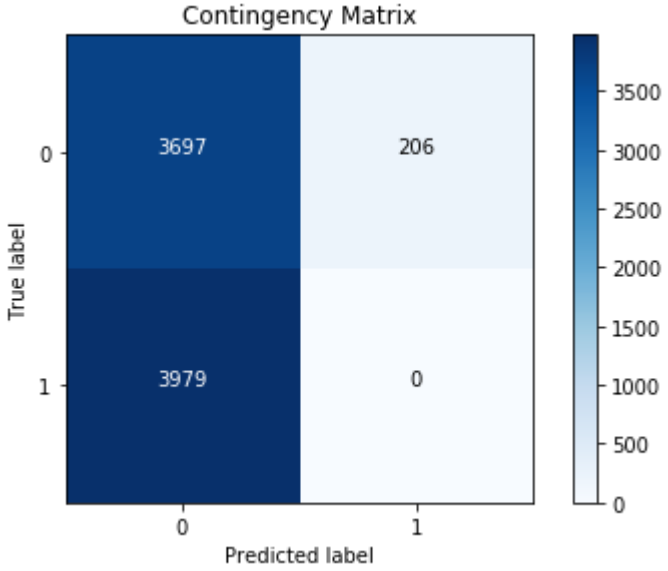
```
V-measure Score: 0.011
Adjusted Rand Score: 0.000
Adjusted Mutual Info Score: 0.004
----------------------------------------
When r is 50 , Contigency matrix is:
Contingency matrix
[[3901    2]
 [3942   37]]
Homogeneity Score: 0.003
Completeness Score: 0.077
V-measure Score: 0.007
Adjusted Rand Score: -0.000
Adjusted Mutual Info Score: 0.002
----------------------------------------
When r is 100 , Contigency matrix is:
Contingency matrix
[[  46 3857]
 [  15 3964]]
Homogeneity Score: 0.002
Completeness Score: 0.024
V-measure Score: 0.003
Adjusted Rand Score: 0.000
Adjusted Mutual Info Score: 0.001
----------------------------------------
When r is 300 , Contigency matrix is:
Contingency matrix
[[3903    0]
 [3943   36]]
Homogeneity Score: 0.005
Completeness Score: 0.107
V-measure Score: 0.009
Adjusted Rand Score: -0.000
Adjusted Mutual Info Score: 0.003
```

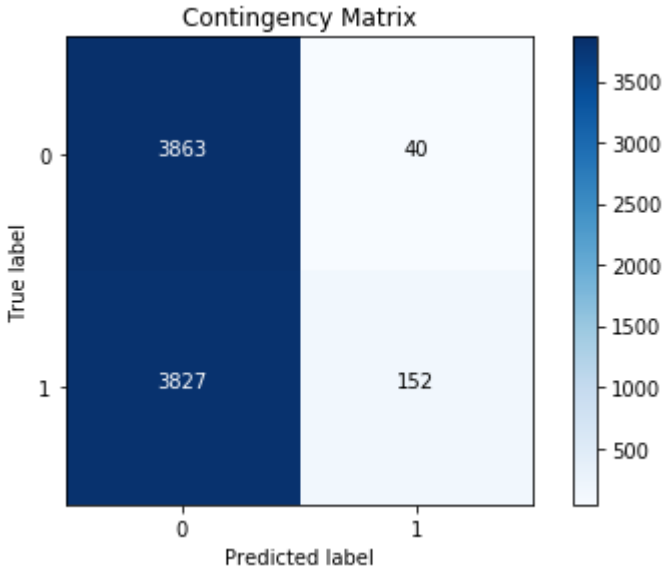


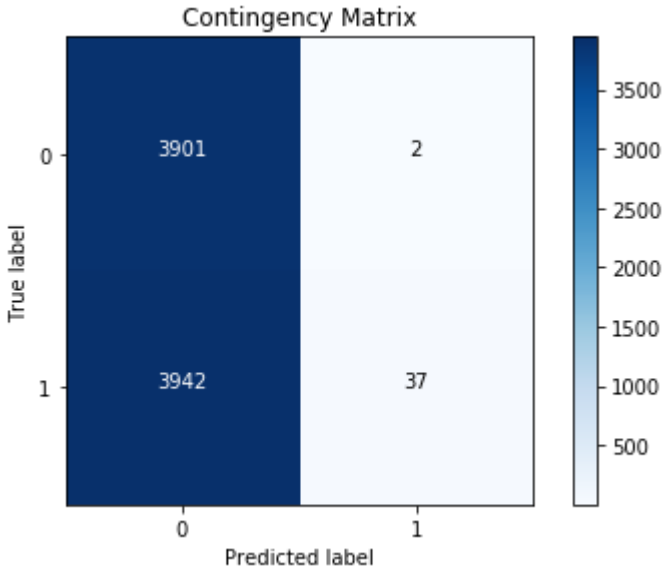

Contingency Matrix
3587
316
158
3821
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

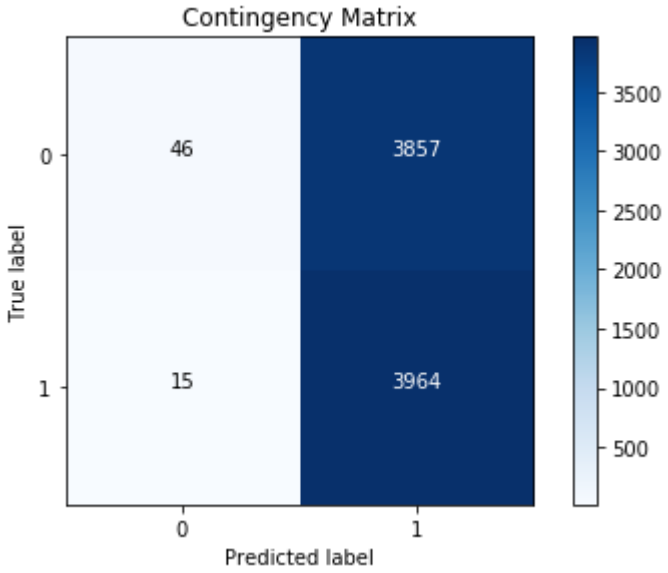

Contingency Matrix
3898
5
2392
1587
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

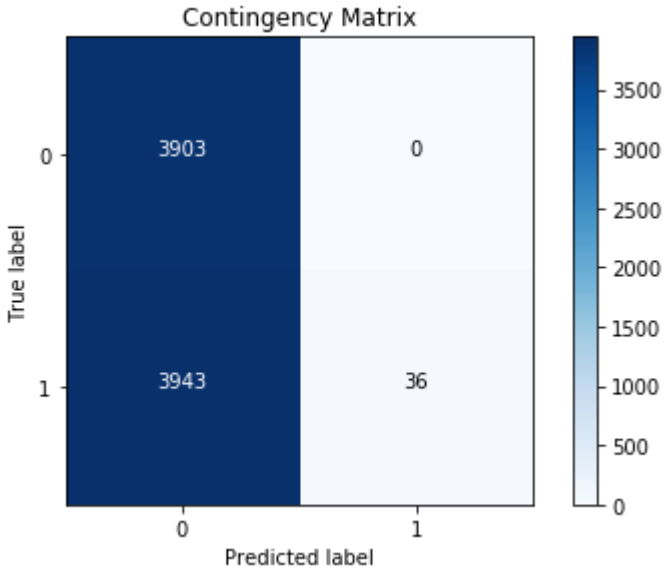

Contingency Matrix
3586
317
1106
2873
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500

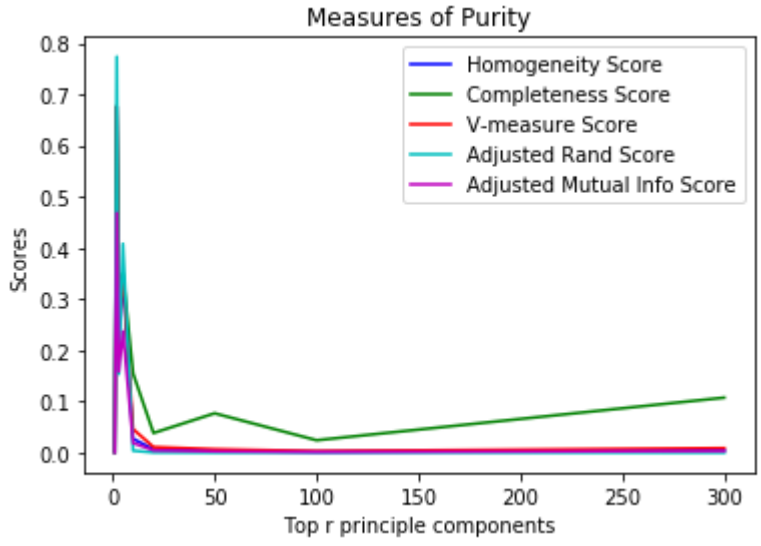

Contingency Matrix
3697
206
3979
0
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
3863
40
3827
152
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
3901
2
3942
37
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
46
3857
15
3964
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500


Contingency Matrix
3903
0
3943
36
True label
Predicted label
0
1
3500
3000
2500
2000
1500
1000
500
0


Measures of Purity
Homogeneity Score
Completeness Score
V-measure Score
Adjusted Rand Score
Adjusted Mutual Info Score
Scores
Top r principle components
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
50
100
150
200
250
300

Problem 4

(a) Visualize the performance of the case with best clustering resutls in the prevoius part of your clustering by projecting final data vectors onto 2 dimensional plane and color-coding the classes

We define several functions *print_cnf_and_score*, *svd_pipeline*, *nmf_pipeline* for the ease of future use first.

- Function *print_cnf_and_score*:

> By giving the predicted labels, ground truth, and the class labels, it will print out the contigency matrix, five measure scores.

- Function *svd_pipeline*:

> By giving r(the range of r we used in problem 3), model(homogeneity score of particular model, i.e. SVD), X(TF-IDF matrix), flag_norm(if we will perform normalization), n_clusters(number of clusters that will be used by k-Means), it will return the trained k-Means model and show the visualization of clustering results.

- Function *nmf_pipeline*:

> By giving r(the range of r we used in problem 3), model(homogeneity score of particular model, i.e. NMF), X(TF-IDF matrix), flag_norm(if we will perform normalization), n_clusters(number of clusters that will be used by k-Means), it will return the trained k-Means model and show the visualization of clustering results.

By calling these functions, we can see the visualiation results below.

- Analysis:

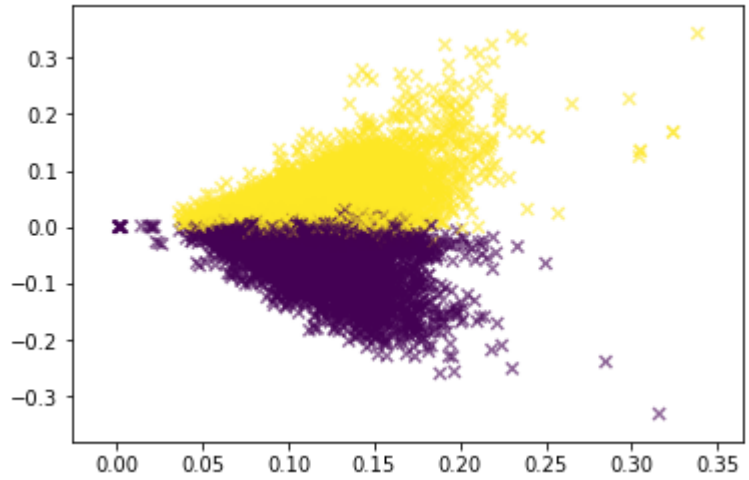
> It can be seen that both SVD and NMF dimension reduction methods give satisfactory results in clustering the TF-IDF matrix. The visualization of each of them shows that the data are linearly separable when required to cluster into two categories.

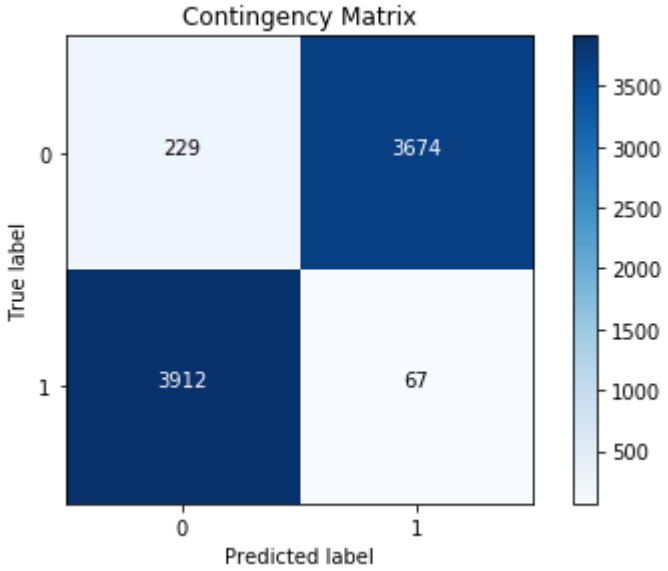
In [11]:
```
def print_cnf_and_score(true_label, pred_label, classes = [0,1]):
    cnf_matrix = metrics.confusion_matrix(true_label, pred_label)
    print_plot_cnf_matrix(cnf_matrix, classes)
    print_scores(true_label, pred_label)
```

In [12]:
```
from sklearn.preprocessing import scale
def svd_pipeline(r, model, X, flag_norm = False, n_clusters = 2):
    best_r_svd = r[np.argmax(model)]
    svd = TruncatedSVD(n_components = best_r_svd, random_state = 42)
    svd_train = svd.fit_transform(X)
    if flag_norm:
        svd_reduced = scale(svd_train)
    else:
        svd_reduced = svd_train

    km = KMeans(n_clusters, max_iter = 500, random_state = 42, n_init = 1).f
it(svd_reduced)
    if n_clusters == 2:
        plt.scatter(svd_reduced[:, 0], svd_reduced[:, 1], marker = 'x', c =
km.labels_, alpha = .6)
    else:
        svd_reduced_to_2 = TruncatedSVD(n_components = 2, random_state = 42)
.fit_transform(svd_reduced)
        plt.scatter(svd_reduced_to_2[:, 0], svd_reduced_to_2[:, 1], marker =
 'x', c = km.labels_, alpha = .6)

    return km
```

In [13]:
```
km = svd_pipeline(r, lsi_homo, X_8_tfidf, flag_norm = False)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[ 229 3674]
 [3912   67]]
Homogeneity Score: 0.776
Completeness Score: 0.778
V-measure Score: 0.777
Adjusted Rand Score: 0.855
Adjusted Mutual Info Score: 0.538
```

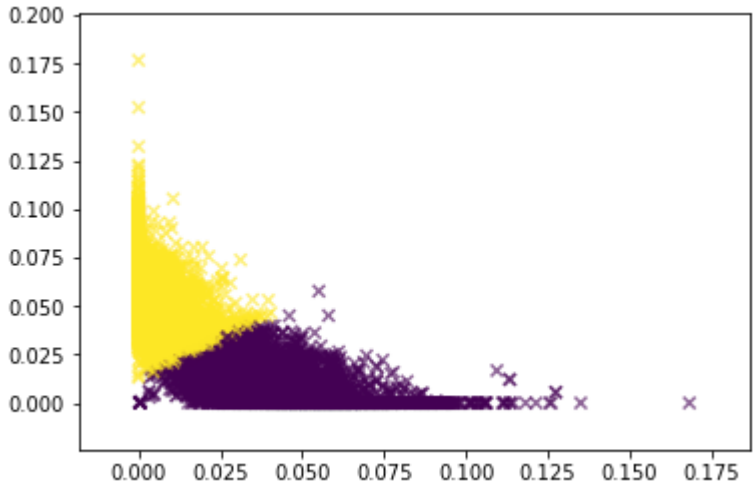


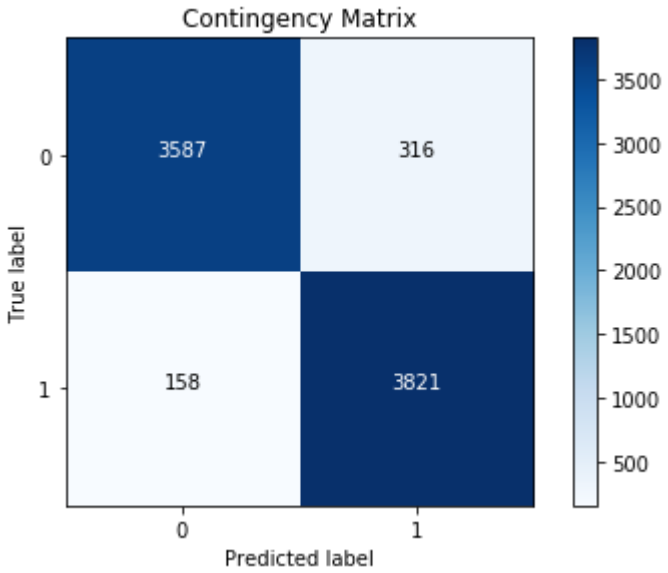


In [14]:
```
def nmf_pipeline(r, model, X, flag_norm = False, n_clusters = 2):
    best_r_nmf = r[np.argmax(model)]
    nmf = NMF(n_components = best_r_nmf, random_state = 42)
    nmf_train = nmf.fit_transform(X)
    if flag_norm:
        nmf_reduced = scale(nmf_train)
    else:
        nmf_reduced = nmf_train
    km = KMeans(n_clusters, max_iter = 500, random_state = 42, n_init = 1).fit
(nmf_reduced)

    if n_clusters == 2:
        plt.scatter(nmf_reduced[:, 0], nmf_reduced[:, 1], marker = 'x', c = km
.labels_, alpha = .6)
    else:
        nmf_reduced_to_2 = TruncatedSVD(n_components = 2, random_state = 42).f
it_transform(nmf_reduced)
        plt.scatter(nmf_reduced_to_2[:, 0], nmf_reduced_to_2[:, 1], marker =
'x', c = km.labels_, alpha = .6)

    return km
```

In [15]:
```
km = nmf_pipeline(r, nmf_homo, X_8_tfidf, flag_norm = False)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[3587  316]
 [ 158 3821]]
Homogeneity Score: 0.676
Completeness Score: 0.677
V-measure Score: 0.676
Adjusted Rand Score: 0.774
Adjusted Mutual Info Score: 0.468
```





(b) Visualize the transformed data as in part (a). Report the new clustering measures including the contigency matrix after transformation

- **For SVD**, we can just perform normalization to the TF-IDF matrix. Normalization is performed by calling *scale* function in the *sklearn.preprocessing* package.

  Normalization

  By calling *svd_pipeline* and setting flag_norm equals to True, we can visualize the normalized data below. By calling *print_cnf_and_score*, we can see the contigency matrix, five measure scores as well as the contigency matrix heat-map.

- **For NMF**, we here present the visualiztion for:

1. Normalization only

   By calling *nmf_pipeline* and setting flag_norm equals to True, we can visualize the normalized data below. By calling *print_cnf_and_score*, we can see the contigency matrix, five measure scores as well as the contigency matrix heat-map.

2. Logrithm only

   Here, we write a new function called *nmf_pipeline_with_log* whose inputs are similar to *nmf_pipeline* except that we add a new input called norm_first to determine whether to perform normaliation first or logrithm first. To avoid negative number that may cause NaN, we simply extract the minimum number to make all numbers positive first and then add 1e-3 to avoid zero. For NMF only with logrithm, we can set flag_norm to be False and norm_first to be False. After that, call *print_cnf_and_score* to show contigency matrix, contigency matrix heat-map, and five measure scores.

3. Normalization followed by logrithm

   Call 'nmf_pipeline_with_log" while setting flag_norm to be True and norm_first to be True. After that, call *print_cnf_and_score* to show contigency matrix, contigency matrix heat-map, and five measure scores.

4. Logrithm followed by normalization

   Call 'nmf_pipeline_with_log" while setting flag_norm to be True and norm_first to be False. After that, call *print_cnf_and_score* to show contigency matrix, contigency matrix heat-map, and five measure scores.

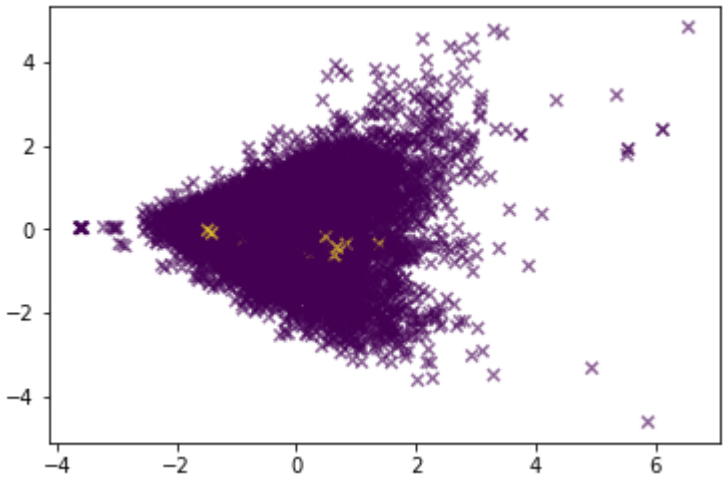
- Analysis:

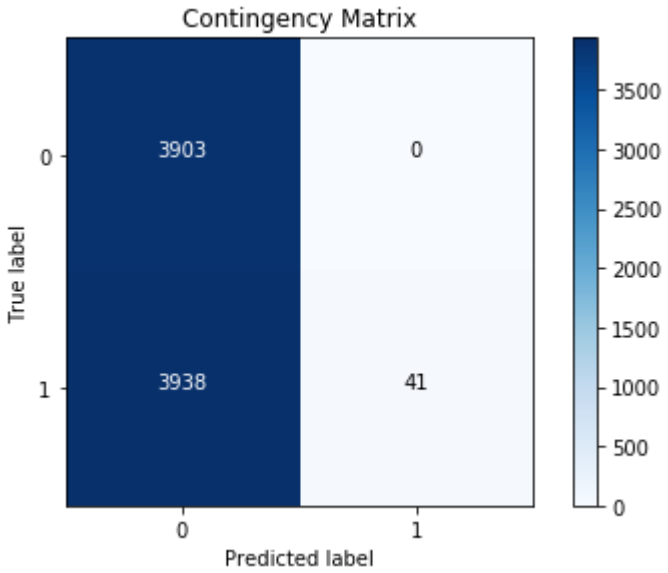
  Since five measure scores have similar trends when r increases, we may only look at one of them to compare the performance of SVD. For example, we compare their homogeneity scores.

SVD with normalization

In [16]:
```
km = svd_pipeline(r, lsi_homo, X_8_tfidf, flag_norm = True)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[3903    0]
 [3938   41]]
Homogeneity Score: 0.005
Completeness Score: 0.110
V-measure Score: 0.010
Adjusted Rand Score: -0.000
Adjusted Mutual Info Score: 0.004
```

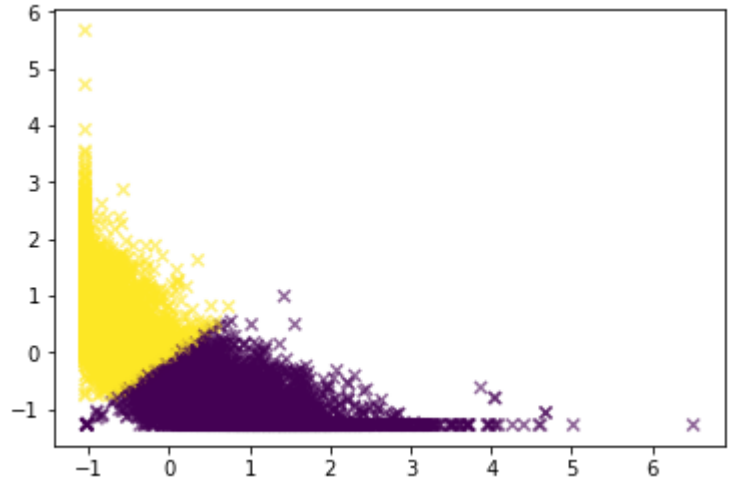


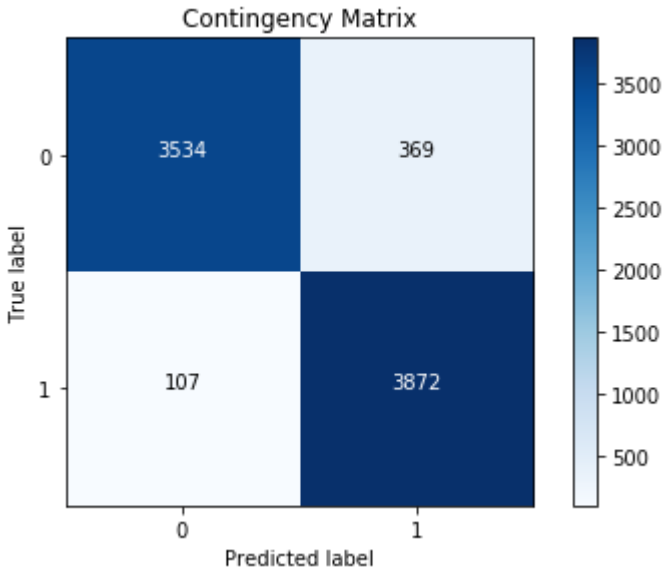


NMF with normalization

In [17]:
```
km = nmf_pipeline(r, nmf_homo, X_8_tfidf, flag_norm = True)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[3534  369]
 [ 107 3872]]
Homogeneity Score: 0.682
Completeness Score: 0.685
V-measure Score: 0.684
Adjusted Rand Score: 0.773
Adjusted Mutual Info Score: 0.473
```



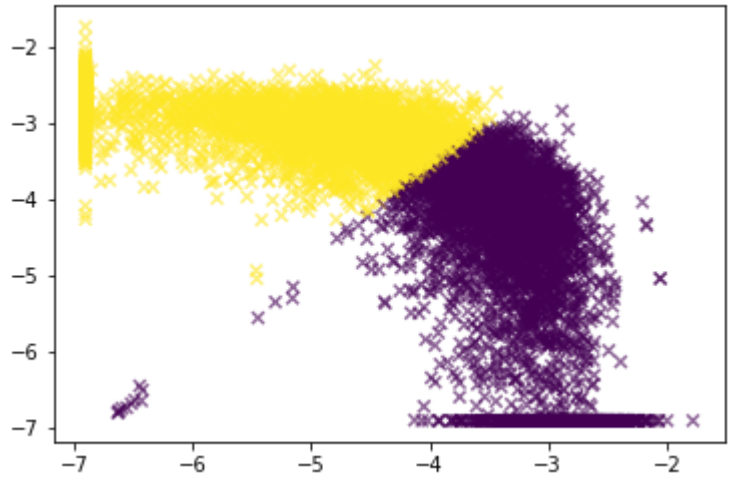


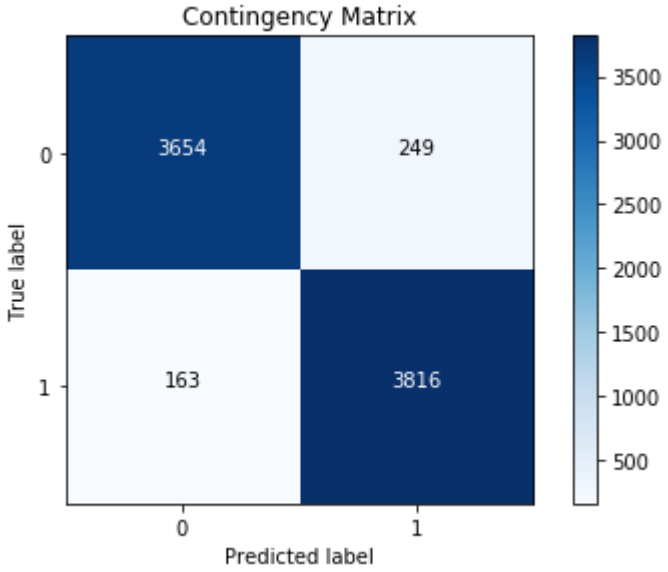
NMF with logrithm

In [18]:
```
def nmf_pipeline_with_log(r, model, X, flag_norm = False, norm_first = True,
 n_clusters = 2):
    best_r_nmf = r[np.argmax(model)]
    nmf = NMF(n_components = best_r_nmf, random_state = 42)
    nmf_train = nmf.fit_transform(X)
    if flag_norm == False:
        nmf_reduced = np.log(1e-3 + nmf_train - nmf_train.min(0))
    else:
        if norm_first == True:
            nmf_train_norm = scale(nmf_train)
            nmf_reduced = np.log(1e-3 + nmf_train_norm - nmf_train_norm.min(
0))
        else:
            nmf_train_log = np.log(1e-3 + nmf_train - nmf_train.min(0))
            nmf_reduced = scale(nmf_train_log)

    km = KMeans(n_clusters, max_iter = 300, random_state = 42, n_init = 1).f
it(nmf_reduced)

    if n_clusters == 2:
        plt.scatter(nmf_reduced[:, 0], nmf_reduced[:, 1], marker = 'x', c =
km.labels_, alpha = .6)
    else:
        nmf_reduced_to_2 = TruncatedSVD(n_components = 2, random_state = 42)
.fit_transform(nmf_reduced)
        plt.scatter(nmf_reduced_to_2[:, 0], nmf_reduced_to_2[:, 1], marker =
 'x', c = km.labels_, alpha = .6)

    return km
```

In [19]:
```
km = nmf_pipeline_with_log(r, nmf_homo, X_8_tfidf, flag_norm = False, norm_fir
st = False)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[3654  249]
 [ 163 3816]]
Homogeneity Score: 0.705
Completeness Score: 0.706
V-measure Score: 0.706
Adjusted Rand Score: 0.802
Adjusted Mutual Info Score: 0.489
```

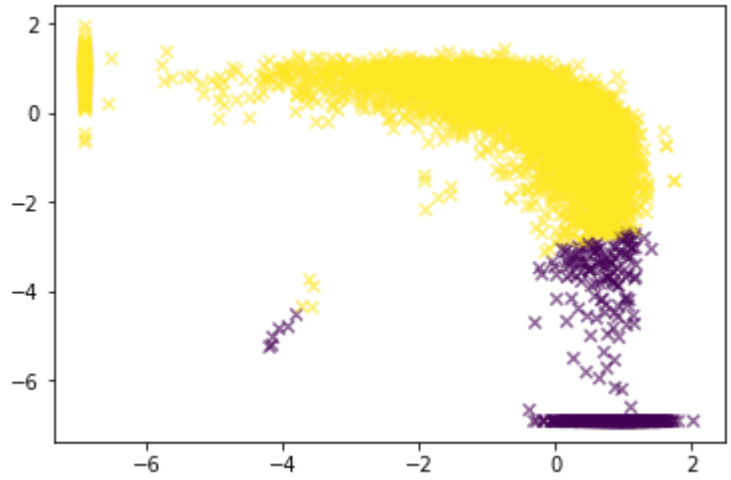


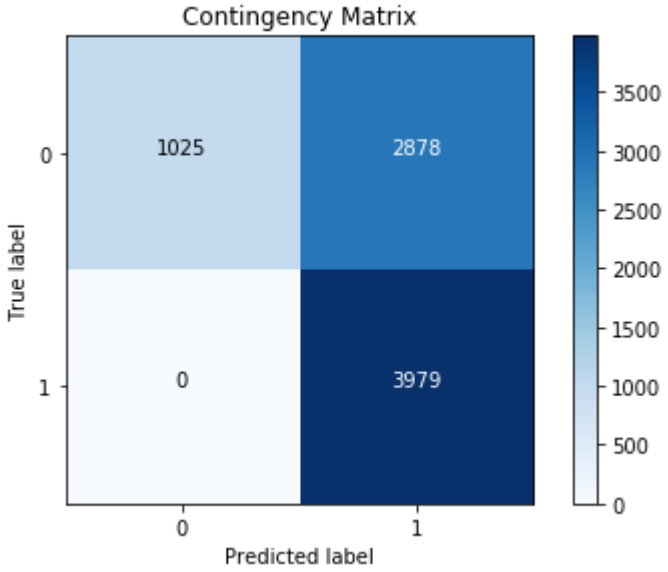


NMF with normalization first and then logrithm

In [20]:
```
km = nmf_pipeline_with_log(r, nmf_homo, X_8_tfidf, flag_norm = True, norm_firs
t = True)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[1025 2878]
 [   0 3979]]
Homogeneity Score: 0.146
Completeness Score: 0.262
V-measure Score: 0.188
Adjusted Rand Score: 0.073
Adjusted Mutual Info Score: 0.101
```

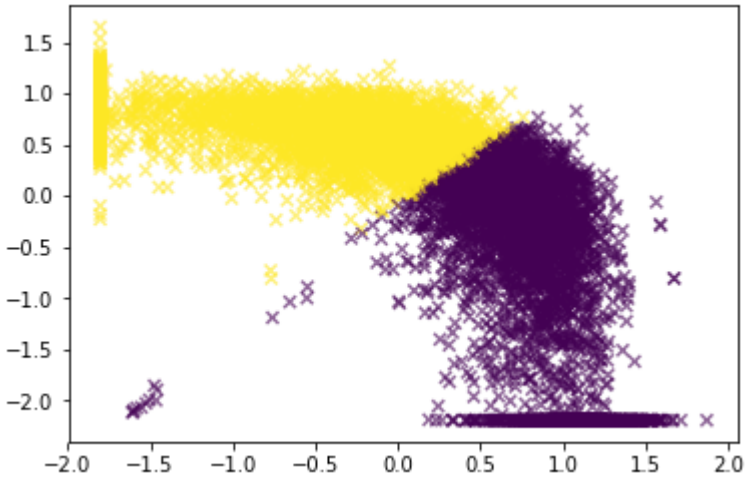


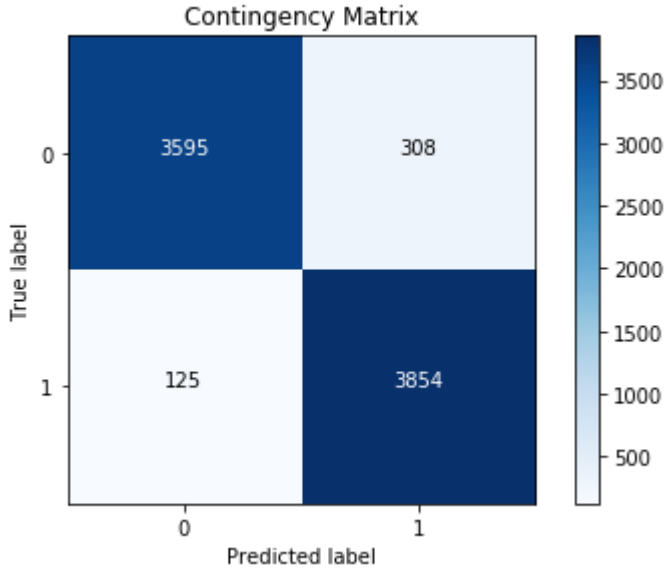


NMF with logrithm first and then normalization

In [21]:
```
km = nmf_pipeline_with_log(r, nmf_homo, X_8_tfidf, flag_norm = True, norm_firs
t = False)
print_cnf_and_score(label_8, km.labels_)
```

```
Contingency matrix
[[3595  308]
 [ 125 3854]]
Homogeneity Score: 0.699
Completeness Score: 0.700
V-measure Score: 0.700
Adjusted Rand Score: 0.792
Adjusted Mutual Info Score: 0.484
```





Problem 5

- Expand dataset into 20 categories and retrieve all original sub-class labels with clustering. Get TF-IDF matrix using the same parameters in part 1.

  Instead of using just eight categories, we here use the whole dataset. Still min_df = 3, and use *CountVectorizer* to get the term-document matrix and use *TfidfTransformer* to get the TF-IDF matrix. The size of TF-IDF matrix we extract for 20 categories is **(18846, 52268)**.

- Try different dimensions for both truncated SVD and NMF.

  To be consistent with the problem 3, we are still using r in the range of [1,2,3,5,10,20,50,100,300]. Contigency matrices, contigency matrix heat-maps as well as their five measure scores for SVD and NMF are shown below.

- Next, we try different transformation on SVD and NMF.

  - **For SVD**:

    We show the **contigency matrix, contigency matrix heat-map, and five measure scores** of **SVD with no normalization** and **SVD with normalization**. Since k = 20 when performing k-Means clustering, when visualizting them in the 2D space, we perform an addition dimension reduction which is already written in the function *svd_pipeline*.

  - **For NMF**:

    We will show the **contigency matrix, contigency matrix heat-map, and five measure scores** of **NMF with no normalization or logrithm**, **NMF with normalization only**, **NMF with logrithm only**, **NMF with normalization first followed by logrithm** and **NMF with logrithm first then normalization**. Still, since k = 20 when performing k-Means clustering, when visualizting them in the 2D space, we perform an addition dimension reduction which is already written in the function *svd_pipeline*.

- Analysis:

After trying different dimension for SVD and NMF when we fix k = 20 for k-Means clustering, we find that **r = 10 gives the best scores for both of SVD and NMF**. Our expectation is to find r around 20, however r = 10 could also be possible that since theses twenty categories are from two main classes: computer technology and recreation activity, maybe some of them are too similar to others in the same main class such that k-Means clustering cannot spearate them.

Since five measure scores have similar trends when r increases, we may only look at one of them to compare the performance of SVD. For example, we compare their homogeneity scores. For SVD, it is the same as before that SVD without normalization can give better clustering results than SVD with normalization. However, the difference in performance when k = 20 is not that large as k = 2. Moreover, performance for SVD without normalization when k = 20 is worse than when k = 2. It is because that clustering data into 20 categories is more difficult than clustering into 2 categories.

k=2, without normalization	**k=2, with normalization**	**k=20, without normalization**	**k=20, with normaliazation**
0.776	0.005	0.341	0.315

For NMF, we also compare homogeneity scores as an example. It can be seen that the average performance of k = 20 is worse than that of k = 2 due to the increase in no. of clusters and this agrees with our expectation. What is more, for k = 20 when performing k-Means clustering, whether there is non-linear transformatioin or not, there is no big diffrence.

- When k = 2 for k-Means clustering

k=2, without normalization	**k=2, with normalization**	**k=2, with logrithm**	**k=2, with normalization then logrithm**	**k=2, with logrithm then normalization**
0.676	0.682	0.705	0.146	0.699

- When k = 20 for k-Means clustering

k=20, without normalization	**k=20, with normalization**	**k=20, with logrithm**	**k=20, with normalization then logrithm**	**k=20, with logrithm then normalization**
0.317	0.309	0.370	0.251	0.366

In [22]:
```
from sklearn.datasets import fetch_20newsgroups

X_20 = fetch_20newsgroups(subset='all', shuffle=True, random_state=42)

data_20 = X_20.data
label_20 = X_20.target
# label_all_groups = (X_all_groups.target > 3).astype(int)

# Get the term-document matrix under the condition: min_df=3
count_vect_3_groups = CountVectorizer(min_df=3, stop_words = combined_stopwo
rds)

X_20_counts = count_vect_3_groups.fit_transform(data_20)

# Get the TF_IDF feature matrix
tfidf_transformer = TfidfTransformer()
X_20_tfidf = tfidf_transformer.fit_transform(X_20_counts)
print(X_20_tfidf.shape) # = (18846, 52268)
```

```
(18846, 52268)
```

Try different dimensions for SVD and NMF

In [23]:
```
# perform with LSI
r = [1,2,3,5,10,20,50,100,300]
homo_lsi = []
comp_lsi = []
vmeas_lsi = []
adjrs_lsi = []
mutis_lsi = []
for each_r in r:
    svd = TruncatedSVD(n_components = each_r, random_state = 42)
    svd_reduced = svd.fit_transform(X_20_tfidf)
    km = KMeans(n_clusters = 20, max_iter = 500, random_state = 42, n_init = 1
).fit(svd_reduced)
    print('-'*40)
    print('When r is', each_r, ', Contigency matrix is: ')
    cnf_matrix = metrics.confusion_matrix(label_20, km.labels_)
    print_plot_cnf_matrix(cnf_matrix, np.arange(20))

    print_scores(label_20, km.labels_)
    lsi_homo_20 = save_scores(label_20, km.labels_, homo_lsi, comp_lsi, vmeas_
lsi, adjrs_lsi, mutis_lsi)

plt.figure()
plt.plot(r, homo_lsi, 'b', label = 'Homogeneity Score')
plt.plot(r, comp_lsi, 'g', label = 'Completeness Score')
plt.plot(r, vmeas_lsi, 'r', label = 'V-measure Score')
plt.plot(r, adjrs_lsi, 'c', label = 'Adjusted Rand Score')
plt.plot(r, mutis_lsi, 'm', label = 'Adjusted Mutual Info Score')
plt.xlabel('Top r principle components')
plt.ylabel('Scores')
plt.legend(loc = 'upper right')
plt.title('Measures of Purity')
plt.show()
```

```
----------------------------------------
When r is 1 , Contigency matrix is:
Contingency matrix
[[ 47  63  23  22  66  65   0   0  49  84   1  32  57   7  41  12  70  77
   67  16]
 [102  25 109   3  51  90   5  24  14  64  52 125 110   2   7   2  33  56
   21  78]
 [ 88  41  55   9  61 110   0  23  24 110  21  84 100   0  24   4  56  88
   38  49]
 [ 93  45  75   3  72  96   0   3  29  99  29  96  88   1  21   0  57  80
   40  55]
 [119  39  76   1  68  81   1   3  29 106  33  90  98   2  18   3  53  68
   27  48]
 [107  28 123   5  39  77   3  18   8  74  88 121  82   4   5   1  21  52
   13 119]
 [116  25  99   0  56  81   0   5  14  87  56 108 109   1   8   0  51  53
   15  91]
 [ 86  42  84   7  68 114   0   2  34  91  27  72 110   1  19   4  40  80
   31  78]
 [104  26  88   3  62 133   0   0  20 108  18 124 122   0   5   1  31  71
   31  49]
 [ 94  45  63  11  81  99   0   6  19  81  32  91 100   1  22   2  66 100
   33  48]
 [ 97  36 100   3  70 104   0   9  16 115  48  85 102   1  10   1  38  82
   29  53]
 [ 43  80  32  31 100  76   3   1  72  71   8  37  72   4  48  22 109  96
   69  17]
 [126  24 104   2  54 112   0   4   9  81  33 102 125   0   5   5  58  60
   13  67]
 [105  29 119   5  55  90   3   6   6  73  31 121 102   1   5   0  34  69
   23 113]
 [107  40  72   9  65 100   0   4  19 108  22  83 116   1  22   4  65  71
   28  51]
 [ 67  56  49  36  60  96   1   4  48  82  27  54  83   5  42  34  62  83
   55  53]
 [ 40  93  28  49  79  45   0   0  63  79   5  39  57   6  54  22  85  83
   61  22]
 [ 70  58  74   6  81 105   1   2  21  84  26  68  98   5  15  10  64  78
   43  31]
 [ 57  69  28  24  59  75   1   1  38  71   4  36  51   8  47  15  59  70
   48  14]
 [ 42  53  30  17  49  54   0   2  28  47   6  42  38   6  24  12  59  59
   48  12]]
Homogeneity Score: 0.028
Completeness Score: 0.030
V-measure Score: 0.029
Adjusted Rand Score: 0.006
Adjusted Mutual Info Score: 0.083
----------------------------------------
When r is 2 , Contigency matrix is:
Contingency matrix
[[102   9  35   0  65   0  38  25   0 119  62  38   1   0   0   8  21  12
  125 139]
 [  0 108   0 120   3 219  21  13  15   0   0  48  84  14 238  88   0   0
    0   2]
 [  0  65   0 264   0 116   5   8 121   0   0  12  61  38 247  45   0   0
    0   3]
```

```
 [  1  68   0 210   0 155   6   4 118   0   0   7  87  25 271  30   0   0
    0   0]
 [  0 114   0 137   0 177  13   7  10   0   0  16 170  18 258  43   0   0
    0   0]
 [  0 105   0  89   0 310   7   5  16   0   0  39  87  19 192 119   0   0
    0   0]
 [  1 117   0 127   0 210  17   8  56   0   0  29  70  10 224 104   0   0
    0   2]
 [ 16 229   0   1  15  60 175  91   0   0   0 168  92   0  11 106   0   0
    1  25]
 [ 34 218   0   3   9  69 207  58   0   0   0 239  50   1  12  65   0   0
    0  31]
 [ 24 126   0   4  27  46 267  99   0   0   0 213  44   1   9  83   1   0
    0  50]
 [ 23 161   0   1  18  36 256  65   0   0   0 242  44   1   2 114   0   0
    0  36]
 [ 25  96   0  12  74  15 223 199   1   1   1  95  90   4  17  24  10   0
    2 102]
 [  5 263   0  23   0 164  58  18   1   0   0 132 142   7  90  79   0   0
    0   2]
 [ 97  94   1   4  16  42 176  34   0   2   0 294  15   5   5 124   1   0
    6  74]
 [ 30 194   0   3  18  54 195  82   1   0   0 189  72   6  20  80   2   0
    1  40]
 [156   3 111   0  14   6  18   2   0 201 188  28   1   0   2  32  18  39
  115  63]
 [110  25   3   0 148   5 100  66   0  21   2  60   7   0   0  15  56   0
   63 229]
 [182  20   2   0  28  10  64  19   0 193  23  57   6   0   1  29  29   2
  119 156]
 [117   9   3   0 102   1  93  62   0  38   4  69   9   1   4  13  29   0
   60 161]
 [ 95   9  46   0  22   1  40  20   0  81  69  38   7   0   0  12  15  21
   84  68]]
Homogeneity Score: 0.215
Completeness Score: 0.234
V-measure Score: 0.224
Adjusted Rand Score: 0.069
Adjusted Mutual Info Score: 0.641
----------------------------------------
When r is 3 , Contigency matrix is:
Contingency matrix
[[ 98   0  14   0   0   4  27  20 179   9  52 145  11  99   0   1   8   0
   86  46]
 [ 11 223  85   1  71  11   0  10   0  41   1   0  83   0  18 111 134 146
    5  22]
 [  2 165  42   0 230   8   0   3   0  21   0   1  53   0  95 110  65 183
    3   4]
 [  1 187  29   0 125  19   0  13   0  24   0   0  51   0 114 112  52 253
    0   2]
 [  2 221  94   0  87  10   0  33   0  59   0   2  41   0  13 194  70 125
    0  12]
 [  2 209  53   0  45  15   0   4   0  20   0   0 159   0  24  84 182 185
    1   5]
 [  5 127 190   0  95   5   0  34   0  23   0   2  62   0  22 236  99  55
    0  20]
 [ 70  29 158   0   1  17   0  36   1 156   2  19 200   0   0  38 109  11
```

```
  41 102]
 [ 92  35 111   0   0  35   0  17   1 115   1  17 279   0   0  34 105  11
  62  81]
 [ 88  11 191   0   0   4   0 105   0  82   3  67  47   0   0  20  46   3
  17 310]
 [ 73   1 235   0   0   0   0 198   0  16   3  45  14   0   0  17  52   0
   6 339]
 [ 12  11  11 267   7 357   0   3   0  68  25   5 103   0   4  11  15  26
  59   7]
 [ 32 159 143   0  13  31   0  11   0 116   0   0 195   0   2  83 105  63
   5  26]
 [207  12 148   0   1  11   0  66   5  40   9  37 120   1   2   6 148   1
  82  94]
 [131  36 127   0   5  49   0  11   1 153  10  19 171   0   1  30  85  21
  63  74]
 [130   4  21   0   0   2  93   0 268   0  21  88   9 247   0   1  32   0
  52  29]
 [122   2  17  13   0  47   1  10  23  47 141 131  49   4   0   2  21   0
 219  61]
 [177   4  38   0   0   2   4  41 253   9  35 151   5  50   0   1  36   1
  69  64]
 [125   0  23   4   0  15   2  30  41  33  96  90  44   7   0   3  19   0
 191  52]
 [ 91   0  13   0   0   1  46  27  98   4  56  64  23  87   0   2  13   0
  65  38]]
Homogeneity Score: 0.238
Completeness Score: 0.248
V-measure Score: 0.243
Adjusted Rand Score: 0.082
Adjusted Mutual Info Score: 0.711
----------------------------------------
When r is 5 , Contigency matrix is:
Contingency matrix
[[282   2  22   0  49  34   0   1   0  27 110  50  18   0   0  11  13  32
  22 126]
 [  0 128  65 103   3   0  13   0   0 148   4  82  10 118 277   0   0   9
  12   1]
 [  0 187  29 106   3   0  75   0   0  62   2  38   5 311 160   0   0   3
   4   0]
 [  0 135  42 183   0   0  99   0   0  63   1  15  20 200 196   3   0   6
  19   0]
 [  0 198  39 120   1   0   8   0   0 165   4  32  30 112 223   1   0  24
   6   0]
 [  1  91  85 162   0   0  23   0   0  77   2  66   4  85 371   0   0   9
  12   0]
 [  2 149 128  52   2   0  17   0   2 275   4  80  35  59  89  14   0  41
  26   0]
 [  1   2 283  10  43   0   0   0   0 186  48 102  24   0  14  61   0 100
 116   0]
 [ 16   2 319   7  24   0   0   0   0 115  14  96  11   0  11  89   0  70
 222   0]
 [  0   0  63   0   5   0   0   0 165  78   5  58   5   0   2 356   0 188
  69   0]
 [  1   1  28   0   4   0   0   0 442  23   1  32   5   0   0 343   0  97
  22   0]
 [  1   7  43  65 171   0   2 487   0  25  54  27   5   6  18   1  26   7
  46   0]
```

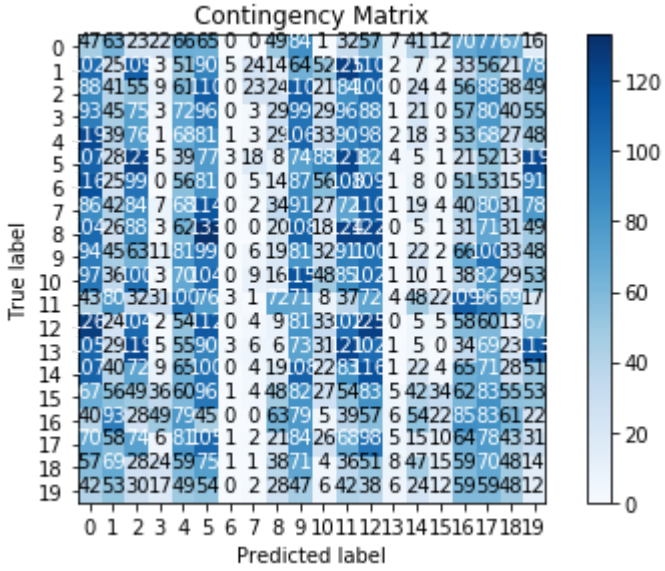
```
 [  1  55 243  75   4   0   0   0   0 228  14  97   6  26 149  17   0  24
   45   0]
 [ 18   4 183   3  69   1   2   0   1 156 102 284   5   1  15  17   0  94
   34   1]
 [ 10   7 266  22  73   0   1   0   0 144  55 127   1   3  20  82   2  55
  119   0]
 [367   1  27   2   5 139   0   0   0  27  23  78   1   1   6   1   1   6
    8 304]
 [  8   1  51   0 289   0   0  16   0  22 223  37  15   0   1   5 147  33
   62   0]
 [ 15   0  16   0  78   2   0   0   0  26 293  93  53   0   1   2 326  28
    4   3]
 [ 20   0  44   0 259   0   0   5   0  26 177  57  36   0   0   3  82  22
   41   3]
 [177   1  39   0  38  58   0   0   0  24  55  65  16   0   0   2  16  25
   16  96]]
Homogeneity Score: 0.306
Completeness Score: 0.320
V-measure Score: 0.313
Adjusted Rand Score: 0.122
Adjusted Mutual Info Score: 0.915
----------------------------------------
When r is 10 , Contigency matrix is:
Contingency matrix
[[268   0 136  69 139   1  36  14   0   5  50   0  79   1   1   0   0   0
    0   0]
 [  1   0   1 157 166  16  78 417   1   9   0   0   4   0   0  49   0   0
   74   0]
 [  0   0   0  70  82   4  52 345   0   9   0  11   2   0   0  63   0   0
  347   0]
 [  0   0   0  89  84   5 103  98   3  27   0 143   0   0   0 335   0   4
   39  52]
 [  0   0   0 148 201   8  71  63   0  31   0  76   1   0   0 355   0   2
    5   2]
 [  0   0   0 126 101  22  90 532   8   4   0   0   0   0   0   4   0   0
  101   0]
 [  1   3   1 247 257   7 120  20   0  48   0  54   2   0   0 177   0  20
   16   2]
 [  0   0   3 305 206   8 351  11   0  49   0   0  35   0   0  12   0  10
    0   0]
 [  5   0  22 256 149  16 455  28   0  17   0   0  17   0   0   4   0  27
    0   0]
 [  0 135   4  96 203   2 129   2   0   8   0   0   4   0   0   0   0 411
    0   0]
 [  1 413   0  51  56   2  35   0   0  11   0   0   1   0   0   0   0 429
    0   0]
 [  0   0  15  52  58  12  60  44 467   7   0   0  44   0 222   2   0   0
    8   0]
 [  0   0   0 370 217  32 173  71   8   7   0   5   0   0   0  91   0   5
    5   0]
 [  8   0   5 399 236  20 145  15   0   7   1   0  72   0   0   0  76   4
    2   0]
 [  0   0   1 237 118 531  63  13   0   1   0   0  22   0   0   1   0   0
    0   0]
 [507   0   2 135  57   3  46   6   0   1 213   0  23   0   0   2   0   1
    1   0]
 [  1   0 118  64 111   4 120   1   3  12   0   0 471   2   2   1   0   0
```

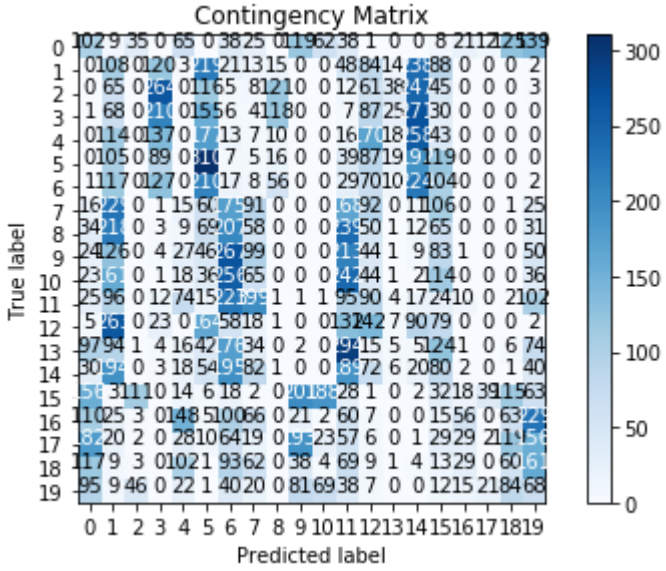
```
    0   0]
 [  5   0   1  77 112   0  19   3   0  18   2   0 228 475   0   0   0   0
    0   0]
 [  9   0  77  97  88  13  97   3   1  48   2   0 329   2   1   0   0   8
    0   0]
 [183   0  63  85  83   4  45   3   1  18  71   0  70   0   0   0   2   0
    0   0]]
Homogeneity Score: 0.341
Completeness Score: 0.388
V-measure Score: 0.363
Adjusted Rand Score: 0.137
Adjusted Mutual Info Score: 1.019
---------------------------------------
When r is 20 , Contigency matrix is:
Contingency matrix
[[339 112   0   0  53   1   0   0   1 138  38  21   0   0  73   1   0   1
    0  21]
 [  3 551   0   0 141  18   0   0   2   0   0   2   2   0   0   1   0   0
  246   7]
 [  1 212   1  11  80   3   0   0  11   0   0   3   2   0   0   0   0   0
  647  14]
 [  0 355   4 198 218   8   0   0   4   0   0   0   3   0   0   0   0   0
  162  30]
 [  0 634   2  78 149  10   0   0   1   0   1   2  14   0   0   0   0   0
   47  25]
 [  0 404   0   0 204  27   0   0   1   0   0   0   2   0   0   0   0   4
  342   4]
 [  2 591  13  55 213   4   0   0   6   0   0   1  33   0   0   3   0   0
   40  14]
 [  1 357   0   0 497  12   0   0   0   0   0  66  31   0   0   2   0   0
    2  22]
 [ 14 283   2   0 645  20   0   0   1   0   0  15   7   0   0   0   0   0
    0   9]
 [  1 368 445   0 155   3   0   0   1   0   0   9   7   0   0   4   0   0
    0   1]
 [  1 139 794   0  50   3   0   0   0   0   0   1   6   0   0   0   0   0
    0   5]
 [  2 163   0   0 178   6   0   0  33   0   0  61   8   0   0  16   1 507
   16   0]
 [  0 577   2   4 314  37   0   0   1   0   0   4   7   0   0   0   0   2
   34   2]
 [ 39 478   3   0 299  22   0   0   2   0   0  53   5  77   2   2   0   0
    4   4]
 [ 12 282   0   0 121 440   0   0 105   0   0  22   0   0   0   1   0   0
    3   1]
 [507 101   0   0  63   3   0   0   0   0   1   6   1   0 312   0   0   0
    1   2]
 [  5 107   0   0 144   6   0   0   1   0   0 547   5   0   0  78   0   4
    0  13]
 [ 54 181   0   0  55   0 390 169   0   0   0  68   1   0   2   0   0   0
    0  20]
 [ 38 152   1   0 111  17   0   0   1   0   0 267  26   0   2   5 130   2
    0  23]
 [208 104   0   0  62   0   0   0   4  19  69  59   2   2  77   1   0   0
    0  21]]
Homogeneity Score: 0.317
Completeness Score: 0.425
```

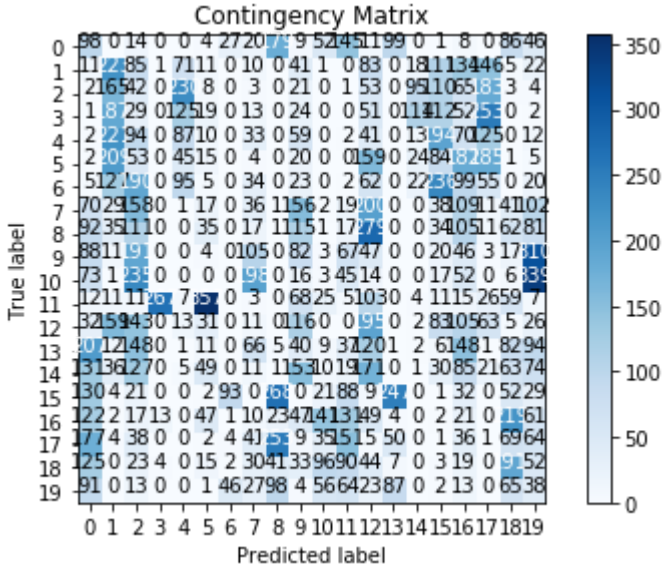
```
V-measure Score: 0.363
Adjusted Rand Score: 0.104
Adjusted Mutual Info Score: 0.947
----------------------------------------
When r is 50 , Contigency matrix is:
Contingency matrix
[[144  66   0   4   0   0   0   1   0   0 163 226  96   0   2   0  41  41
   15   0]
 [  0  82  82   1   0   2   0   0   0   0 216   1 538   0   1   0   0  46
    2   2]
 [  0  34 553   0   0  12   0   0   0   2 139   1 205   0   0   7   0  26
    4   2]
 [  0  27 107   2   0   5   0   0   0   0 180   0 444   3   0 171   0  34
    6   3]
 [  0  24  10   1   0   1   0   0   0   0 355   0 441   0   0  66   0  32
   19  14]
 [  0  89 126   0   0   2   0   4   0   0 108   0 588   0   0   0   0  49
   20   2]
 [  0   5  21  37   0   7   0   0   0   0 402   0 355  14   5  39   0  43
   15  32]
 [  0  17   2 548   0   0   0   0   0   0 180  12 166   0   3   0   0  14
   18  30]
 [  0  84   0 609   0   1   0   0   0   0 104   1 144   0   0   0   0  36
   11   6]
 [  0   2   0   3   0   1   0   0   0   0 306   2 189 387   4   0   0  85
    8   7]
 [  0   2   0   2   0   0   0   0   0   0 113   1  86 719   0   0   0  16
   54   6]
 [  0  53   8   0   0  34   0 496   0   0  63  41 222   0  16   0   0  42
    8   8]
 [  0  49  11  37   0   1   0   2  12  13 245   0 560   1   0   3   0  37
    6   7]
 [  3  20   2   1   0   3  77   0   0  14 240  17 550   0   2   0   0  47
    9   5]
 [  0  23   1   2   0 110   0   0  77   0 194  19 523   0   1   0   0  33
    4   0]
 [535  18   1   1   0   0   0   0   0  19 119  33 244   0   0   0   1  16
   10   0]
 [  0  14   0   7   0   2   0   4   0   0 123 512 131   0  72   0   0  36
    4   5]
 [  3   3   0   1 174   0   0   0   0   0 130 515  73   0   0   0   0  39
    1   1]
 [  2  27   0   3   0   1   0   2   0   0 144 389 158   0   5   0   0  14
    0  30]
 [140  19   0   3   0   4   2   0   0   0 114 104 133   0   1   0  70  30
    6   2]]
Homogeneity Score: 0.288
Completeness Score: 0.382
V-measure Score: 0.329
Adjusted Rand Score: 0.101
Adjusted Mutual Info Score: 0.861
----------------------------------------
When r is 100 , Contigency matrix is:
Contingency matrix
[[157  36   0   9   1 128   0   0  87 196  71   1  64   2   0   0   3  41
    3   0]
 [  1  48   0   9   0  17   0  37   0 172  77   0  71   1   2   0   0   0
```

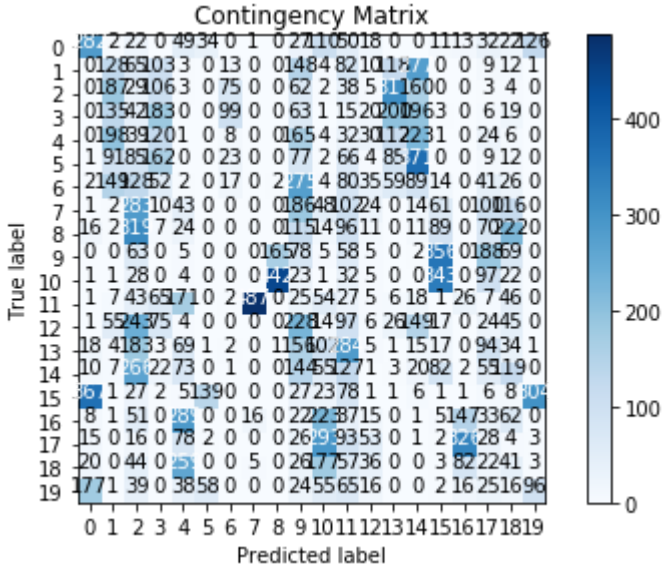
```
 536   2]
[  0  30   0  11   0   5   0 470   0 119  31   0  49   0  11   8   0   0
 249   2]
[  0  34   3  16   0   4   0  70   1 122  26   0  99   0   4 170   0   0
 430   3]
[  0  22   0  18   0  10   0   7   0 241  23   0  43   0   1  64   0   0
 520  14]
[  0  24   0  11   0  22   0  68   0 116  78   3  90   0   2   0   0   0
 572   2]
[  0   0  12   9   0   6   0  21   0 427   5   0 133   5   6  40   0   0
 279  32]
[  0  10   0   9   0  34   0   2   0 282  23   0 572   3   0   0   1   0
  22  32]
[  0   2   1  15   0  26   0   0   0 227 101   0 605   0   1   0   0   0
  12   6]
[  0   1 364  32   0   8   0   0   0 413   2   0 140   4   1   0   0   0
  22   7]
[  0   0 769   1   0   5   0   0   0 128   2   0  41   0   0   0   0   0
  47   6]
[  0   7   0  62   0  63   0   5   0  90  53 460 119  16  34   0   0   0
  74   8]
[  0  42   1  18   0  27  12   4   0 231  51   1 235   0   1   3   0   0
 352   6]
[  3  14   0  31   0 352   0   1   0 274  21   0 202   2   2   0   0   0
  83   5]
[  0  34   0   8   0 454  76   0   0 141  25   0  90   1 104   0   0   0
  54   0]
[556   6   0  15   0  87   0   1   0 182  16   0  99   0   0   0   3   1
  30   1]
[  0   1   0  32   0 276   0   0   0 117  15   2 120  71   2   0 262   0
   7   5]
[  5   2   0   1 574 101   0   0   0 168   5   0  58   0   0   0   3   0
  22   1]
[  4   0   1  19   0 367   0   0   0 147  25   2  90   5   1   0  79   0
   4  31]
[143   5   0  10   1  77   0   0  12 152  18   0  72   1   4   0  49  69
  13   2]]
Homogeneity Score: 0.303
Completeness Score: 0.377
V-measure Score: 0.336
Adjusted Rand Score: 0.127
Adjusted Mutual Info Score: 0.906
----------------------------------------
When r is 300 , Contigency matrix is:
Contingency matrix
[[ 33 196 187   0   0   0   0   0 134   0   0 196   0   0   0   5   0   0
    9  39]
 [  2 532 219   0   0   1   0   1   0   0   0   1   0   0 158   2   1   0
    1  55]
 [  2 194 120  33   0   0   0   6   0  10   0   0   0   0 578   7   2   0
    0  33]
 [  3 432 178  15   3   0   0   2   1 187   0   0   0   0 114  10   1   0
    0  36]
 [  6 456 363  14   0   0   0   1   0  70   0   0   0   0  23   2   4   0
    0  24]
 [  0 608 141  10   0   0   0   2   0   0   0   0   0   0 194   3   1   0
    0  29]
```

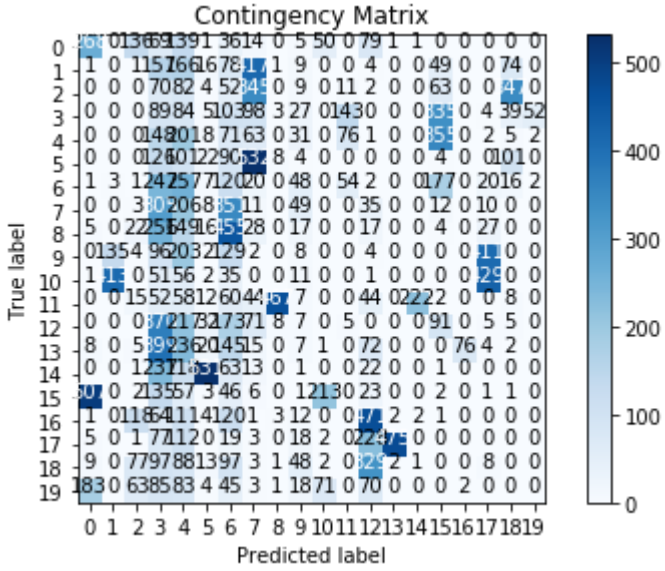
```
[  9 318 472   1  13   0   0  13   0  49   0   0   0   0  27  39   8   0
   0  26]
[ 16 187 208   0   0   1   0   2   0   0   0   0   0   0   3 557   2   0
   4  10]
[  4 147 128   0   0   0   0   0   0   0   0   0   0   0   0 713   1   0
   0   3]
[  1 183 354   0 414   0   0  14   0   0   0   0   0   0   0  21   3   0
   2   2]
[  0  92 107   0 766   0   0  24   0   0   0   0   0   0   0   2   8   0
   0   0]
[  5 208  83   0   0   0   0   0   0   0  26   0   0   1  12   1   1   0
 643  11]
[  5 600 237   0   1   3   0   4   0   4   0   0   0   0  20  49   5  12
   1  43]
[  4 593 283   0  77   0   0   0   0   0   0   4   0   0   2   7   2   0
   2  16]
[ 20 622 147   0   0  60   0   4   0   0   0   0   0   0   4   2   4  75
   9  40]
[  3 245 154   0   0   0   0   7   0   0   0 579   0   0   1   1   0   0
   1   6]
[  6 166 171   0   0   0   0  17   0   0   0   1   0   0   0  18  17   0
 512   2]
[ 25 146 481   0   0   0 187   3   0   0   0   3  58   0   0   4   0   0
  27   6]
[  6 220 216   0   1   0   0   0   0   0   0   6   0 132   0  12   0   0
 182   0]
[  5 184 144   0   2   0   0   0  15   0   0 215   0   0   0   5   0   0
  53   5]]
Homogeneity Score: 0.280
Completeness Score: 0.407
V-measure Score: 0.331
Adjusted Rand Score: 0.094
Adjusted Mutual Info Score: 0.836
```

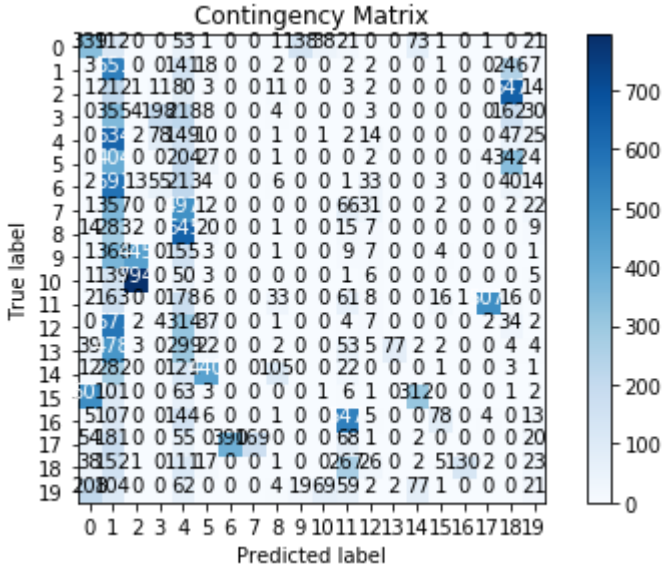


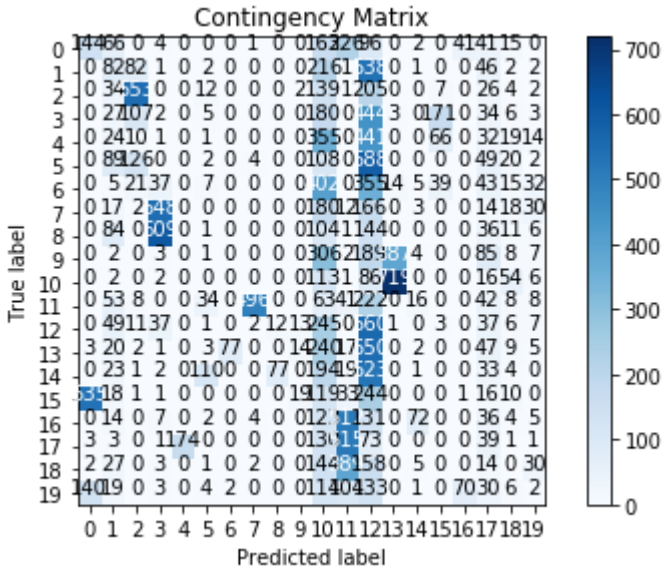

Contingency Matrix
True label
Predicted label

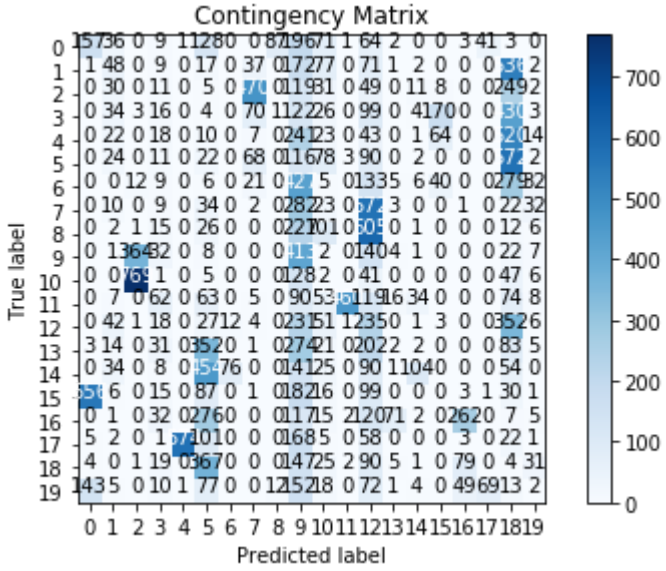

Contingency Matrix
True label
Predicted label

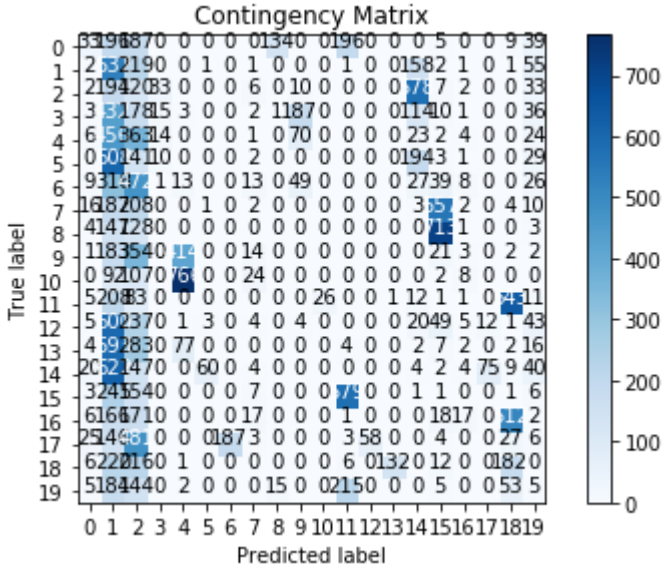

Contingency Matrix
True label
Predicted label

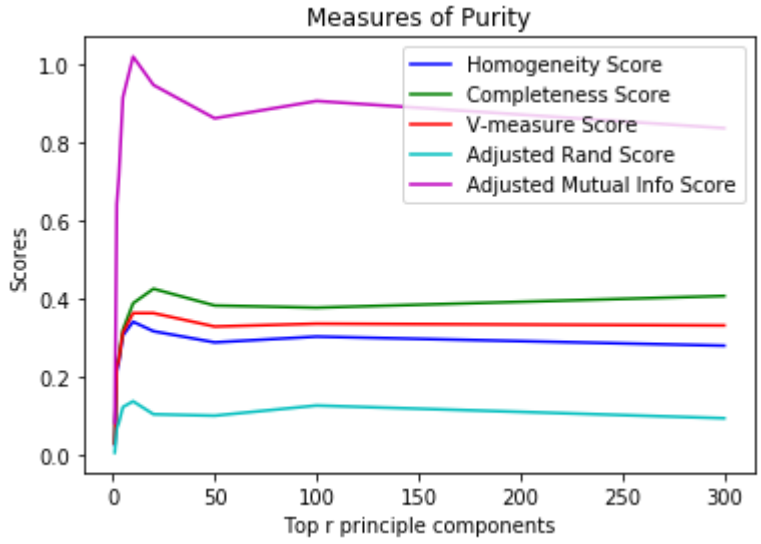

Contingency Matrix
True label
Predicted label


Contingency Matrix
True label
Predicted label


Contingency Matrix
True label
Predicted label


Contingency Matrix
True label
Predicted label


Contingency Matrix
True label
Predicted label


Measures of Purity
Homogeneity Score
Completeness Score
V-measure Score
Adjusted Rand Score
Adjusted Mutual Info Score
Scores
Top r principle components

In [24]:
```
# perform with NMF
r = [1,2,3,5,10,20,50,100,300]
homo_nmf = []
comp_nmf = []
vmeas_nmf = []
adjrs_nmf = []
mutis_nmf = []
for each_r in r:
    nmf = NMF(n_components = each_r, random_state = 42)
    nmf_reduced = nmf.fit_transform(X_20_tfidf)
    km = KMeans(n_clusters = 20, max_iter = 500, random_state = 42, n_init = 1
).fit(nmf_reduced)
    print('-'*40)
    print('When r is', each_r, ', Contigency matrix is: ')
    cnf_matrix = metrics.confusion_matrix(label_20, km.labels_)
    print_plot_cnf_matrix(cnf_matrix, np.arange(20))

    print_scores(label_20, km.labels_)
    nmf_homo_20 = save_scores(label_20, km.labels_, homo_nmf, comp_nmf, vmeas_
nmf, adjrs_nmf, mutis_nmf)

plt.figure()
plt.plot(r, homo_nmf, 'b', label = 'Homogeneity Score')
plt.plot(r, comp_nmf, 'g', label = 'Completeness Score')
plt.plot(r, vmeas_nmf, 'r', label = 'V-measure Score')
plt.plot(r, adjrs_nmf, 'c', label = 'Adjusted Rand Score')
plt.plot(r, mutis_nmf, 'm', label = 'Adjusted Mutual Info Score')
plt.xlabel('Top r principle components')
plt.ylabel('Scores')
plt.legend(loc = 'upper right')
plt.title('Measures of Purity')
plt.show()
```

```
----------------------------------------
When r is 1 , Contigency matrix is:
Contingency matrix
[[ 47  63  23  22  66  64   0   0  49  84   1  32  58   7  41  12  70  77
   67  16]
 [101  25 109   3  51  91   5  24  14  64  52 126 110   2   7   2  33  55
   21  78]
 [ 88  42  55   9  61 111   0  23  23 110  21  84 100   0  24   4  56  87
   38  49]
 [ 93  45  75   3  72  97   0   3  29  99  29  96  88   1  21   0  57  79
   40  55]
 [118  38  75   1  68  82   1   3  29 105  33  91  98   2  18   3  54  68
   27  49]
 [107  28 123   5  39  77   3  18   8  73  88 121  83   4   5   1  21  52
   13 119]
 [114  25  98   0  56  82   0   5  14  85  56 110 110   1   8   0  51  53
   15  92]
 [ 87  42  84   7  68 113   0   2  34  91  27  72 110   1  19   4  40  80
   31  78]
 [103  26  89   3  62 133   0   0  20 108  18 124 122   0   5   1  31  71
   31  49]
 [ 94  45  63  11  81  99   0   6  19  82  32  91 100   1  22   2  66  99
   33  48]
 [ 97  36 101   3  70 104   0   9  16 114  48  84 103   1  10   1  38  82
   29  53]
 [ 44  79  32  31 100  76   3   1  72  71   8  37  71   4  48  22 110  96
   69  17]
 [126  24 104   2  53 112   0   4   9  81  33 102 125   0   5   5  58  61
   13  67]
 [105  29 119   5  55  90   3   6   6  74  31 121 102   1   5   0  34  68
   23 113]
 [106  40  72   9  67 101   0   4  19 106  22  84 117   1  22   4  63  71
   28  51]
 [ 67  58  50  36  59  94   1   4  48  82  27  53  85   5  42  34  62  84
   53  53]
 [ 39  94  28  49  78  45   0   0  63  80   5  40  57   6  54  22  85  83
   60  22]
 [ 71  57  74   6  81 105   1   2  21  84  26  68  98   5  15  10  65  77
   43  31]
 [ 57  68  28  24  58  74   1   1  38  71   4  36  52   8  47  15  60  71
   48  14]
 [ 42  53  30  17  49  54   0   2  28  46   6  42  39   6  24  12  59  59
   48  12]]
Homogeneity Score: 0.028
Completeness Score: 0.030
V-measure Score: 0.029
Adjusted Rand Score: 0.006
Adjusted Mutual Info Score: 0.082
----------------------------------------
When r is 2 , Contigency matrix is:
Contingency matrix
[[111   3  22  94  61  50   8  16  38  37  67   4   0  24  21   1  79  98
    4  61]
 [  0 239   0   0   0   9  66  47  34   0   1 146 113  19  12  25   5   0
  257   0]
 [  0 166   0   0   0   2  42  45  10   0   0  67 269  18  11  65   4   0
  286   0]
```

```
 [  0 208   0   0   1   2  50  63   5   0   0  65 230   8   8  73   0   0
  269   0]
 [  0 223   0   0   0   1  59  75  13   0   0  70 185  15  11  31   0   0
  280   0]
 [  0 241   0   0   0   3 104  31  20   0   0 221 106  15   9  29   1   0
  208   0]
 [  0 239   0   0   0   3  47  19  17   0   0 130 192  17   7  32   1   0
  271   0]
 [  8 103   0   1  10  42 151 117  81   0   7 117  41 111  58   6  29   0
  108   0]
 [  3 109   0   0  22  74 175  82 127   0   2  81  32 126  36   4  26   0
   97   0]
 [  0 120   0   1   2  68 151 117  80   0   8  89  43 121  71   5  24   0
   94   0]
 [  1 163   0   0   4  32 146  84  59   0   5 118  53 121  32   3  16   0
  162   0]
 [ 46  15   2  22  53 170  42  64  71   3  80  21  22  87  86   3 150   9
   32  13]
 [  1 192   0   0   4  14 151 104  76   0   0 108  71  69  14   6   0   0
  174   0]
 [ 25  76   0   6  76 102  96  37 250   0  17  94  17  81  17   5  40   4
   46   1]
 [  3 115   0   4  16  76 152  85 107   0   9  66  37 111  49   9  32   1
  115   0]
 [131   7  62 132 118  41   6   1  44  97  36  15   0  11   3   0  27 137
    2 127]
 [110   5   4  73  81  97  24  14  45  19 109  14   3  32  34   0 121  62
    8  55]
 [162  18   8  96 113  55  15  23  57  14  42  10   3  29  10   0  47 161
   11  66]
 [ 90   5   4  58  73  93  14  23  55  17  81  11   4  39  42   1  76  65
    3  21]
 [ 78   3  31  60  68  25  16  24  36  42  35   8   6  20  16   0  35  56
    2  67]]
Homogeneity Score: 0.175
Completeness Score: 0.187
V-measure Score: 0.181
Adjusted Rand Score: 0.052
Adjusted Mutual Info Score: 0.522
----------------------------------------
When r is 3 , Contigency matrix is:
Contingency matrix
[[166   6  18  52   4   0   9   2  32   9   9 111   4   6  65   8 141 137
   10  10]
 [  0 232   9   0  46   0   0  65   0 120  53   5  81   5   0 114   1  13
   16 213]
 [  0 147   6   0  60   0   0 182   0  63  38   1 121   1   0  63   0   8
   20 275]
 [  0 168  13   0  94   0   0 162   0  99  46   1 113   3   0  48   0   4
   24 207]
 [  0 224   8   0  51   0   0 123   0  81  25   0 114   1   0  62   0   6
   19 249]
 [  0 188  14   0  68   0   0  55   0 154  94   2  82   1   0 173   0   6
   22 129]
 [  2 214  16   0  30   0   0 145   0  82  72   1  48   2   0  84   0   7
   15 257]
 [  0 100 103   0 122   0   0  16   0 123 185  12  87  28   3  47   0   7
```

```
  87  70]
 [  8  87 157   0  80   4   0  11   0 153 201  13  59  48   0  31   0  11
  88  45]
 [  1 179  36   0 103   0   0  23   0 124  68  14 101  13   3  77   0  44
  62 146]
 [  3 230   9   0  70   0   0  33   0 130  50  14 102   1   3 116   0  41
  15 182]
 [  0   9 130   0  57 245   0   5   0  23  58   3  30 301   3   6   0   2
 107  12]
 [  0 179  50   0  95   0   0  24   0 157 144   3 107   6   0  67   0   7
  42 103]
 [ 10 103  78   1  42   1   0  15   1 177 193  65  38   9   7 109   2  54
  28  57]
 [  1  91  72   0 113   3   0  13   0 159 158  23  64  42   6  51   0  30
 111  50]
 [227  11   8 171   1   0  41   0 118   3  14  64   0   1  15  22 223  76
   0   2]
 [  2  13 129   0  77 107   0   2   0  35  66  58  32 207  50   7   0  22
  95   8]
 [ 46  25 129   5  18   0   2   3   0  46  84 219  11  20 142  36  13  86
  15  40]
 [ 12  10 134   3  46  35   0   9   0  35  68  70  32 119  60  13   3  46
  72   8]
 [ 80   8  26  57   5   0  19   3  40  11  15  85  10  12  57  16  85  80
   8  11]]
Homogeneity Score: 0.189
Completeness Score: 0.202
V-measure Score: 0.195
Adjusted Rand Score: 0.057
Adjusted Mutual Info Score: 0.564
----------------------------------------
When r is 5 , Contigency matrix is:
Contingency matrix
[[217   7  16   0   0  67  22   0   0  25   3  55 153   1   0  47 112   3
  66   5]
 [  2 266   6   0 236  44   0   0  31  69  34   0   1   0   0 113   2 162
   2   5]
 [  1 138   5   0 342  16   0   0 178  47  27   0   0   0   0  44   0 187
   0   0]
 [  1 169   7   0 313  36   0   3 156  53  31   0   0   0   2  38   1 166
   5   1]
 [  0 200   4   0 202  83   0   1  27  46  42   0   0   0   3 116   0 233
   4   2]
 [  2 386   4   0 243  29   0   0  35  74  18   0   0   0   0  73   0 123
   1   0]
 [  2 119  10   1  81 129   0   9  33  92  82   0   0   0  18 235   0 150
   7   7]
 [  0  37  96   0   1 105   0   1   0 289  28   0   0   0   5 229   2   4
 134  59]
 [ 14  47  96   0   2  80   0   3   0 293  65   0   0   4  26 162   2   4
  74 124]
 [  1   2   7  76   0  57   0 262   0  26  88   0   0   0 273  79   0   0
   8 115]
 [  1   2   0 292   0  15   0 415   0   9  56   0   0   0 168  11   0   1
   0  29]
 [  1  22 352   0  10  17   0   0   4 113   5   0   0 336   1  22   1   5
  94   8]
```

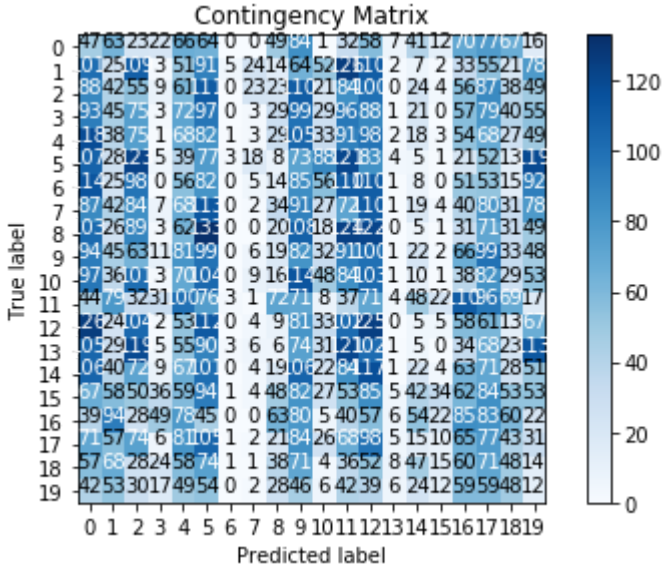
```
 [  3 184  23   0  64  69   0   1   2 236  83   0   0   0   5 203   0  57
   30  24]
 [ 20  53  44   0   2 122   0   0   2 274  41   2   2   0   1 273  23   3
   99  29]
 [ 15  51 108   0   8  74   0   0   1 241  38   0   0   5  31 191   7  11
  140  66]
 [258  17   3   0   1   7  69   0   0  30   7 216 302   0   1  36  47   0
    1   2]
 [  4   5 309   0   0  47   0   0   0  73   3   0   0 162   0  37  26   0
  219  25]
 [ 46   6 244   0   0  89   2   0   0 124   5   0   8   8   4  88 103   0
  181  32]
 [ 18   1 226   0   0  53   0   0   0  96   2   3   1  62   7  65  48   0
  156  37]
 [113   6  28   0   0  49  36   0   0  48   7  66  96   3   0  47  94   1
   30   4]]
Homogeneity Score: 0.294
Completeness Score: 0.314
V-measure Score: 0.304
Adjusted Rand Score: 0.106
Adjusted Mutual Info Score: 0.880
----------------------------------------
When r is 10 , Contigency matrix is:
Contingency matrix
[[259   2   2   0  77   1   6   3   1  50  63   0 117 100 102   0   0   0
   16   0]
 [  1  12 288   0  18   0   0  10   8   0 357   0 143   3 109   1   3   0
   20   0]
 [  0  11 575   0  10   0   0   4   2   0 203   2  62   2  78   0  24   0
   12   0]
 [  0  17 126   0  26   0   0  15   2   0 250  87  99   0 137   4 202   0
   14   3]
 [  0   9  52   0  16   0   0  25   4   0 332  15 262   0  97   2 129   0
   19   1]
 [  0  18 343   0  20   0   0   2  15   0 360   0  72   0 130  12   1   0
   15   0]
 [  1  11  34   0  24   0   0  63   1   0 212  12 365   3 129   1  89   2
   18  10]
 [  0  17   4   0  79   0   0  27   4   0 136   0 287  59 369   0   2   0
    6   0]
 [  8  20   4   0 169   0   0  73   2   0 128   0 166  22 388   0   2   0
   12   2]
 [  0   4   0   0  33   0   0 355   1   0  15   0 115   1  96   0   0  77
   25 272]
 [  1   3   0   0   8   0   0 206   0   0   9   0  26   1  19   0   0 298
    9 419]
 [  0  15  14   0  47 243   4   6  11   0  65   0  37  35  73 432   0   0
    9   0]
 [  0  45  34   0  28   0   0  48  14   0 368   1 195   1 204   9  14   0
   22   1]
 [  8  29   5  75  23   0   1  19   3   1 258   0 146  77 216   0   0   0
  129   0]
 [  0 407   3   0  16   0   0  15 285   0  97   0  75  16  67   0   1   0
    5   0]
 [502   3   4   0   8   0   1   5   2 215 101   0  62  30  60   0   1   0
    3   0]
 [  1   5   0   0 126   2 134   6   3   0  20   0  82 385 129   4   0   0
```

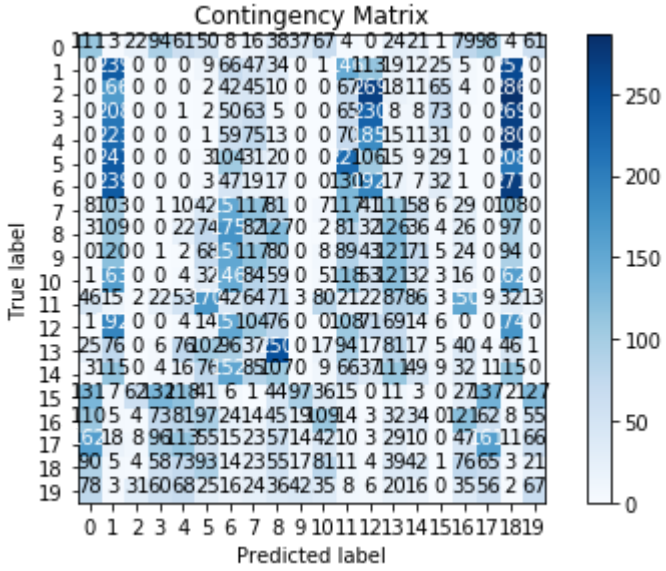
```
  13   0]
 [  2   0   0   0   7   0 366   6   0   2  27   0  86 386  47   0   0   0
  11   0]
 [  5  30   0   0  61   1  58  13   4   2  31   0 105 327 130   1   0   0
   7   0]
 [182   5   1   2  38   0  10   7   0  71  48   0  81  79  86   1   0   0
  17   0]]
Homogeneity Score: 0.317
Completeness Score: 0.359
V-measure Score: 0.337
Adjusted Rand Score: 0.122
Adjusted Mutual Info Score: 0.947
----------------------------------------
When r is 20 , Contigency matrix is:
Contingency matrix
[[128   0 293   0  74   0  38  45   1  70   1 144   0   0   0   0   3   2
    0   0]
 [  0   0 403   0 116   0   0   1   0  82   2   1   0   0   0 342   7  17
    0   2]
 [  0   0 177   0  60   0   0  11   0  35  12   0  10   0   0 665   9   4
    0   2]
 [  0   4 305   0 168   0   0  17   0  29   4   0 200   0   0 218  21  13
    0   3]
 [  0   1 568   0 109   0   0  25   0  24   1   0  85   0   0 110  14  12
    0  14]
 [  0   0 296   0 138   0   0   2   5  87   2   0   0   0   0 422   3  31
    0   2]
 [  0  14 621   0 158   0   0   3   0   5   7   0  59   0   0  59  12   4
    0  33]
 [  0   0 510   0 373   0   0  26   0  27   1   0   0   0   0   4   5  13
    0  31]
 [  0   2 353   0 490   0   0   9   0 107   1   0   0   0   0   0   8  20
    0   6]
 [  0 397 431   0 145   0   0   6   0   2   1   0   0   0   0   0   1   4
    0   7]
 [  0 777 166   0  35   0   0   5   0   2   0   0   0   0   0   0   4   4
    0   6]
 [  0   0 200   0 177   1   0   0 494  52  33   0   0   0   0  15   0  10
    0   9]
 [  0   2 602   0 217   0   0   8   2  55   1   0   5   0  12  42   1  30
    0   7]
 [  0   0 618   0 226   0   0   8   0  22   3   2   0  78   0   4   2  22
    0   5]
 [  0   0 413   0 108   0   0   3   0  24  98   0   0   0  76   3   1 261
    0   0]
 [  0   0 341   0  94   0   1   8   0  17   0 527   0   0   0   1   0   7
    0   1]
 [  0   0 534   0 313   0   0  17   4  19   3   0   0   0   0   1   7   7
    0   5]
 [  0   0 294 381  49   0   0  21   0   3   0   3   0   0   0   0  14   0
  174   1]
 [  0   0 399   0 161 127   0   7   2  11   1   3   0   0   0   0  20  18
    0  26]
 [ 15   1 260   0  87   1  69  18   0  17   4 137   0   2   0   1  14   0
    0   2]]
Homogeneity Score: 0.256
Completeness Score: 0.373
```

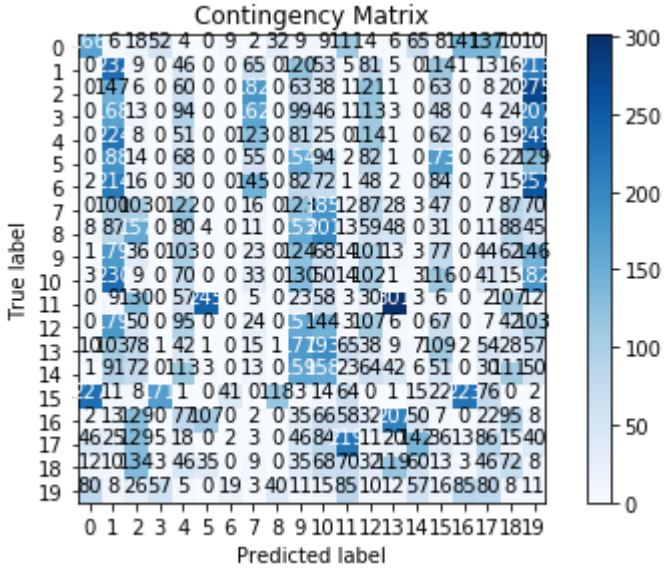
```
V-measure Score: 0.303
Adjusted Rand Score: 0.063
Adjusted Mutual Info Score: 0.764
----------------------------------------
When r is 50 , Contigency matrix is:
Contingency matrix
[[ 28 446   9  41   0   1   0  15   0   0   0   0   0   0   0 141  88   1
    0  29]
 [  2 865   9   0   8   0   1   2   0  39   1   0   6   2   1   1   3  15
    0  18]
 [  3 837  14   0   0   0   4   7   1  59  11   0   3  11   6   0   5   3
    0  21]
 [  3 677  17   0   0   0   5   6   0 245   2   0   1   4   5   0   0   9
    0   8]
 [  7 240  13   0   0   0   0  16   0 660   1   0   0   1   1   0   2   7
    0  15]
 [  0 886  13   0  19   5   0  21   0   9   2   0   0   2   4   0   2  22
    0   3]
 [  9 685   9   0   0   0   2  16   3 217  13   0   0   6   4   0   3   2
    0   6]
 [ 11 839   8   0   0   0   1  19  16   3   2   0   7   0   4   0  26   8
    0  46]
 [  3 375  12   0   0   0   0  10 559   5   0   0   4   1   2   0  10   6
    0   9]
 [  1 877  39   0   0   0   2   8   1   4  15   0   0   1   1   0   8   3
    0  34]
 [  0 860   1   0   0   0   0  57   0   8  22   0   0   0  37   0   5   6
    0   3]
 [  5 316 100   0   0 421   0   9   1  21   0   1   6  35   1   0  68   2
    0   5]
 [  4 753  18   0   0   1   2   8   0 140   4   0   0   1   7   0   3  32
    0  11]
 [  5 880  35   0   0   0   2   9   2   4   0   0   2   3   0   4  28   9
    0   7]
 [ 18 300   8   0   0   0  44   6   0   3   4   0   8  89   0   0  10 458
    0  39]
 [  2 403  18   1   0   0   0  12   0   1   8   0   0   0   0 519  20   3
    0  10]
 [  6 376  37   0   0   1  15   4   0   1  17   0   9   2   3   0 400   3
    0  36]
 [ 24 634   1   0   0   0   0   2   0   0   3   0   2   0   0   3  65   0
  167  39]
 [  6 330  22   0   0   2   0   0   0   0   0 129   9   1   1   3 239   9
    0  24]
 [  5 318  14  70   0   0   0   7   1   0   0   1   2   4  17 128  47   1
    0  13]]
Homogeneity Score: 0.208
Completeness Score: 0.392
V-measure Score: 0.272
Adjusted Rand Score: 0.036
Adjusted Mutual Info Score: 0.622
----------------------------------------
When r is 100 , Contigency matrix is:
Contingency matrix
[[  2   0 153   0   0   0   0 531   0   0  41  22  15   0   4  31   0   0
    0   0]
 [501   0   2  12   0   0   0 392   0   0   0  43   2   1   5   2   0   0
```

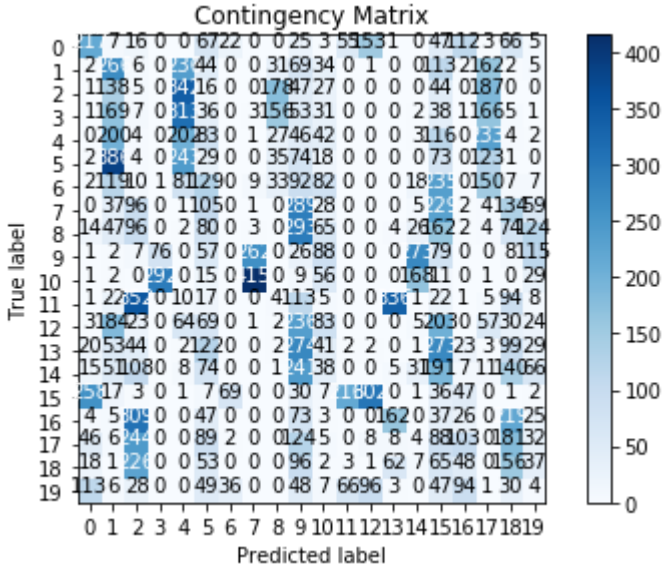
```
  13   0]
 [ 56   0   1  29   9   0   0 807   0   0   0  55   6   7   1   2   3   0
    5   4]
 [ 44   0   0  25   7   0   0 773   0   0   0  85   5   5   5   5  11   0
   14   3]
 [ 21   0   0  12   2   0   0 735   0   0   0 111  20   3   3   4   4   0
   48   0]
 [213   0   0  25   1   0   0 656   0   0   0  31  21   0  13   4   2   0
   15   7]
 [ 22   0   1  14   1   0   0 734   0   0   0 134  15  32   9   2   6   0
    2   3]
 [  5   0   2  49   0   0   0 767   0   0   0 112  19   1   7  11   1   0
   16   0]
 [  2   0   1  47   3   0  11 778   0   0   0  72  11   1  60   3   2   0
    2   3]
 [  4   0   0  35   0   0   0 850   0   0   0  57   8  12   5   5  18   0
    0   0]
 [  0   0   0   7   0   0   0 784   0   0   0  77  54   8  11   1  30   0
   27   0]
 [ 37   0   4   4   1   0   0 843   0   0   0  65  11   0   0   5   2  19
    0   0]
 [ 41   0   9  65   0  12   1 768   0   0   0  54   6   2   5   2   6   0
   13   0]
 [ 10   0   1  15   0   0   0 851   0   0   0  70   9   2   4  13   8   0
    5   2]
 [ 36   0   3   2   0  79   0 796   0   0   0  58   5   0   2   1   0   0
    5   0]
 [  3   0  26   0   0   0   0 836   0   0   1  85  16   0  21   4   4   0
    1   0]
 [  1   0   5  31   0   0   0 718   0   0   1  37   4  36  16  57   0   0
    1   3]
 [  1 169   0   7   0   0   0 672   0  57   0  24   1   0   2   4   0   0
    3   0]
 [  3   0 126  25   1   0   0 523  26   0   0  53   0  10   0   4   0   0
    0   4]
 [  2   0  16  10   0   0   0 455   0   0  70  35   7   0   6  21   0   0
    5   1]]
Homogeneity Score: 0.077
Completeness Score: 0.208
V-measure Score: 0.113
Adjusted Rand Score: 0.006
Adjusted Mutual Info Score: 0.231
----------------------------------------
When r is 300 , Contigency matrix is:
Contingency matrix
[[  1   0  46   0   0   9   0  43   0  16   0   0   0   0   5   0   0   0
  676   3]
 [  3   0  89   5   1   2   1   0   0  15   0   1   1   4  12   0  73   1
  758   7]
 [ 35  18  63   4  16   7   1   0   0  13   0  11   0   4  20   2  26   1
  748  16]
 [  0   2  81   0  69  13   1   0   0  26   1   3   0   1   7   2  31   4
  733   8]
 [  7   0  29   3  46   6   1   0   0   6   0   1   0   1  16   0  12   5
  792  38]
 [ 10  87  60   3   1   1   3   0   0  22   3   3   0   2   3   5   6   0
  654 125]
```

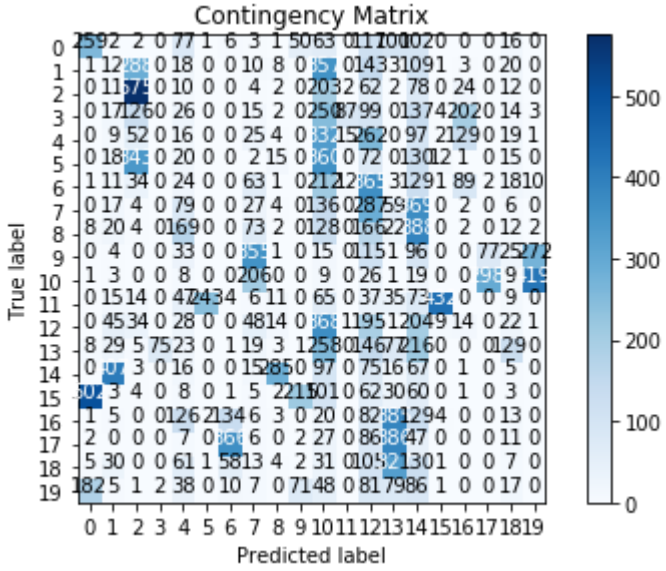
```
[  2   2  19   1  39  14   0   0   0  12  35  13   0   1   6   0  12  44
 763  12]
[  0   1  34   2   0   2   0   0   0  20   0   2   0   0  49   0   1   1
 865  13]
[  0   1  71  36   0   2   0   0   0  14   0   0   0   0  10   0   0   1
 849  12]
[  0   0  17  11   0   8   0   0   0  37   1  17   0   0  29   3   0 161
 684  26]
[  1   0  25   0   0   0   1   0   0  55   3  23   0   1   3   3   0 207
 671   6]
[  0   0  48   2  11   3  27   0   0   6   1   0   0   1   6   0   2   0
 866  18]
[  0   0  57   1   9   4   0   0   0  26   0   4   0   0  14   0   6   4
 847  12]
[  1   1  38   1   0   4   0   0   9  28   0   0   0   0   8   0   0   0
 889  11]
[  2   0   7   1   0   3   0   0   0  14   0   5   0 287  49  10   3   1
 598   7]
[  1   0   6   0   0   2   0   0   0  30   0   8   0   2  15   0   0   0
 899  34]
[  0   1  34   5   1   8   0   0   0  14   0  19   0   0  32   1   0   0
 781  14]
[ 35   0   5   0   0   3  27   0   0   8   0   3 156   0  30   0   0   0
 649  24]
[  0   0  20   0   0   3   0   0   0  37   0   0   0   0  31  21   0   0
 655   8]
[  0   0  23   0   0  43   0   0   0  17   0   0   2   1   6   0   0   0
 534   2]]
Homogeneity Score: 0.074
Completeness Score: 0.210
V-measure Score: 0.109
Adjusted Rand Score: 0.003
Adjusted Mutual Info Score: 0.221
```

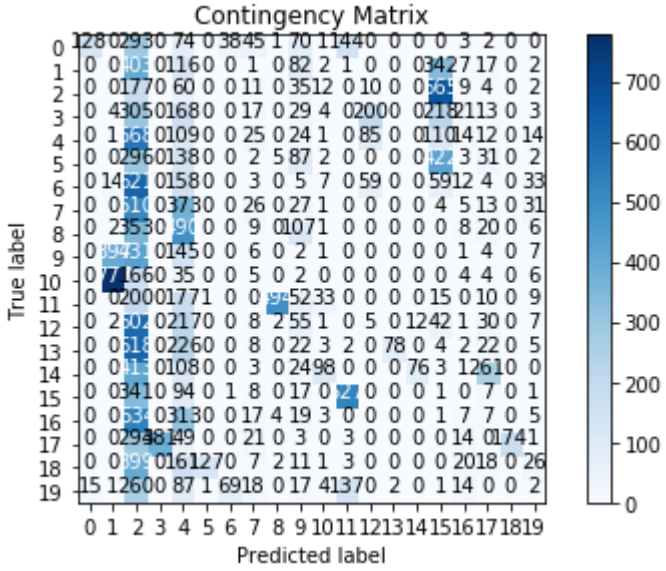


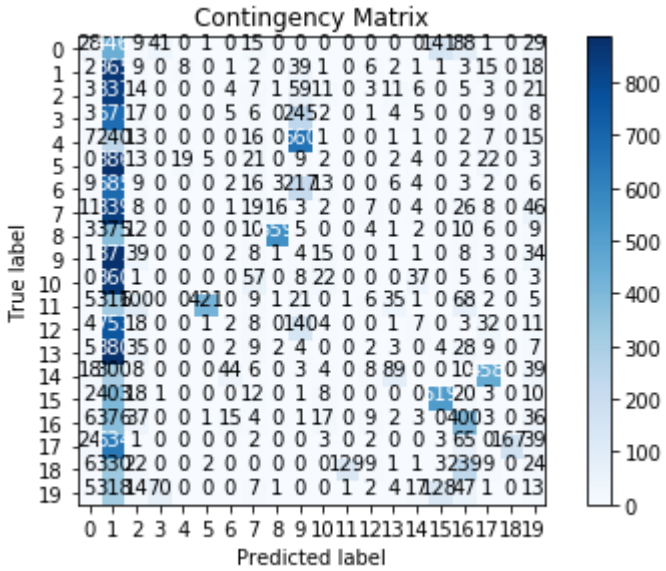

Contingency Matrix
True label
Predicted label

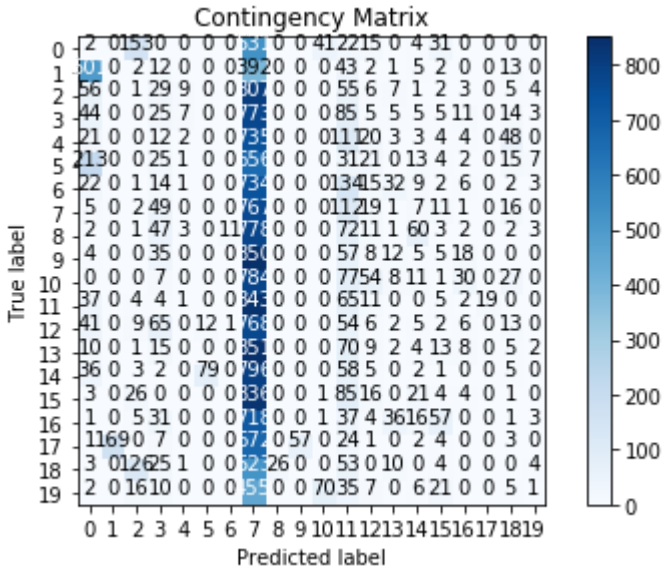

Contingency Matrix
True label
Predicted label

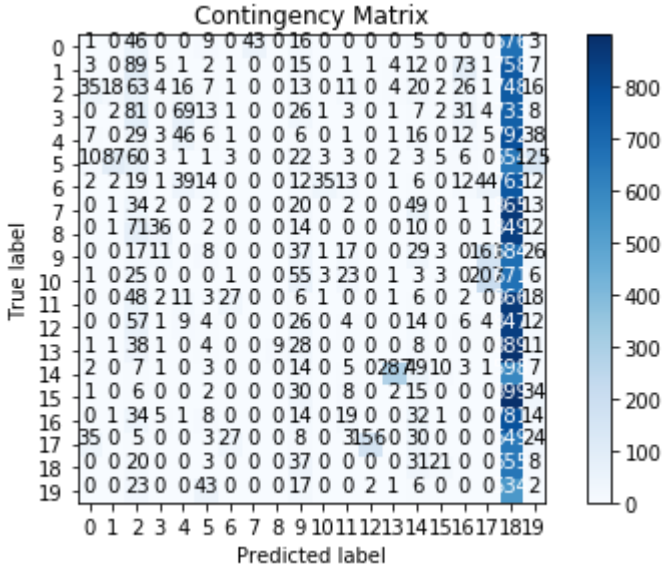

Contingency Matrix
True label
Predicted label

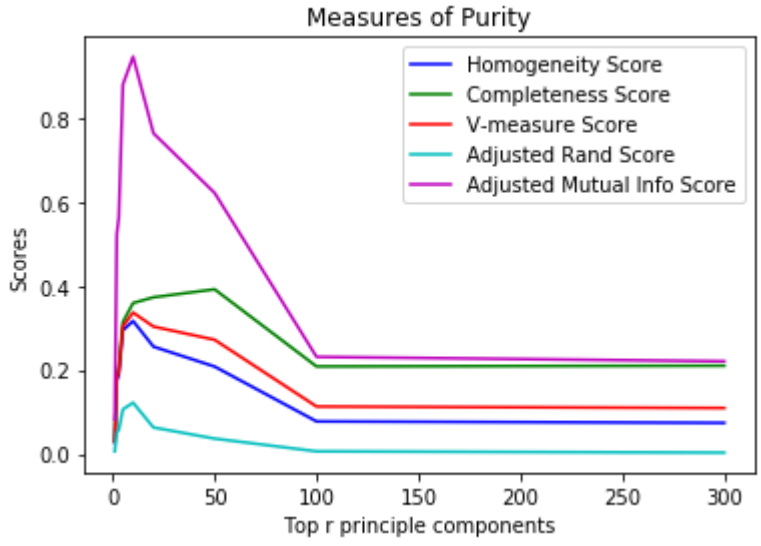

Contingency Matrix
True label
Predicted label


Contingency Matrix
True label
Predicted label


Contingency Matrix
True label
Predicted label





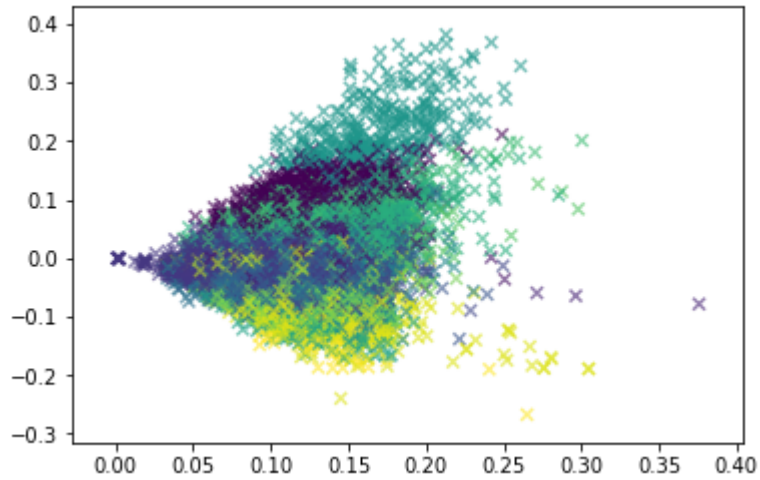


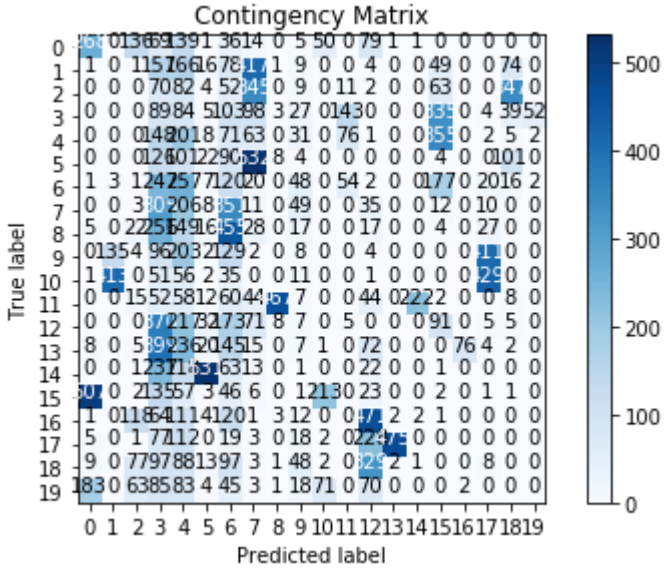
Try SVD and NMF with different transformations

For SVD, firstly try with no transformation

In [25]:
```
km = svd_pipeline(r, lsi_homo_20, X_20_tfidf, flag_norm = False, n_clusters =
20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[268   0 136  69 139   1  36  14   0   5  50   0  79   1   1   0   0   0
    0   0]
 [  1   0   1 157 166  16  78 417   1   9   0   0   4   0   0  49   0   0
   74   0]
 [  0   0   0  70  82   4  52 345   0   9   0  11   2   0   0  63   0   0
  347   0]
 [  0   0   0  89  84   5 103  98   3  27   0 143   0   0   0 335   0   4
   39  52]
 [  0   0   0 148 201   8  71  63   0  31   0  76   1   0   0 355   0   2
    5   2]
 [  0   0   0 126 101  22  90 532   8   4   0   0   0   0   0   4   0   0
  101   0]
 [  1   3   1 247 257   7 120  20   0  48   0  54   2   0   0 177   0  20
   16   2]
 [  0   0   3 305 206   8 351  11   0  49   0   0  35   0   0  12   0  10
    0   0]
 [  5   0  22 256 149  16 455  28   0  17   0   0  17   0   0   4   0  27
    0   0]
 [  0 135   4  96 203   2 129   2   0   8   0   0   4   0   0   0   0 411
    0   0]
 [  1 413   0  51  56   2  35   0   0  11   0   0   1   0   0   0   0 429
    0   0]
 [  0   0  15  52  58  12  60  44 467   7   0   0  44   0 222   2   0   0
    8   0]
 [  0   0   0 370 217  32 173  71   8   7   0   5   0   0   0  91   0   5
    5   0]
 [  8   0   5 399 236  20 145  15   0   7   1   0  72   0   0   0  76   4
    2   0]
 [  0   0   1 237 118 531  63  13   0   1   0   0  22   0   0   1   0   0
    0   0]
 [507   0   2 135  57   3  46   6   0   1 213   0  23   0   0   2   0   1
    1   0]
 [  1   0 118  64 111   4 120   1   3  12   0   0 471   2   2   1   0   0
    0   0]
 [  5   0   1  77 112   0  19   3   0  18   2   0 228 475   0   0   0   0
    0   0]
 [  9   0  77  97  88  13  97   3   1  48   2   0 329   2   1   0   0   8
    0   0]
 [183   0  63  85  83   4  45   3   1  18  71   0  70   0   0   0   2   0
    0   0]]
Homogeneity Score: 0.341
Completeness Score: 0.388
V-measure Score: 0.363
Adjusted Rand Score: 0.137
Adjusted Mutual Info Score: 1.019
```

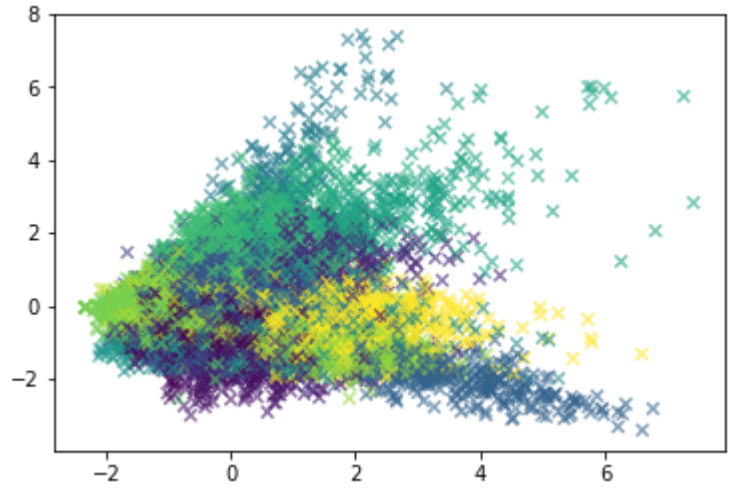


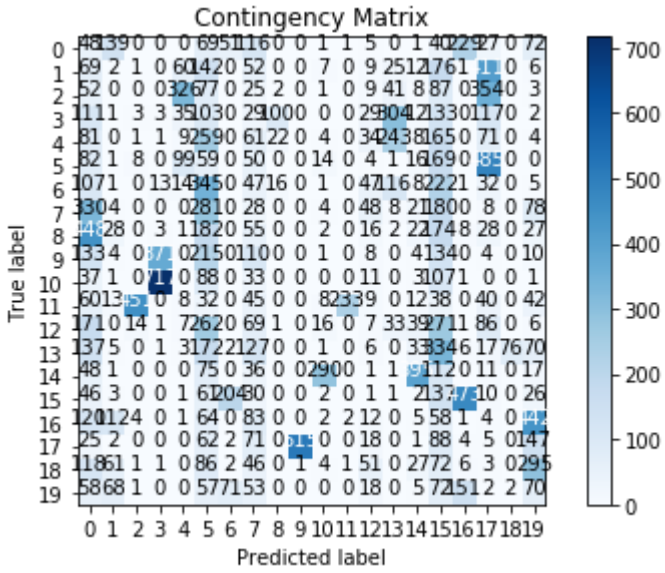


For SVD, secondly try with normalization

In [26]:
```
km = svd_pipeline(r, lsi_homo_20, X_20_tfidf, flag_norm = True, n_clusters = 20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[ 48 139   0   0   0  69  51 116   0   0   1   1   5   0   1  40 229  27
    0  72]
 [ 69   2   1   0  60 142   0  52   0   0   7   0   9  25  12 176   1 411
    0   6]
 [ 52   0   0   0 326  77   0  25   2   0   1   0   9  41   8  87   0 354
    0   3]
 [111   1   3   3  35 103   0  29 100   0   0   0  29 304  12 133   0 117
    0   2]
 [ 81   0   1   1   9 259   0  61  22   0   4   0  34 243   8 165   0  71
    0   4]
 [ 82   1   8   0  99  59   0  50   0   0  14   0   4   1  16 169   0 485
    0   0]
 [107   1   0  13  14 345   0  47  16   0   1   0  47 116   8 222   1  32
    0   5]
 [330   4   0   0   0 281   0  28   0   0   4   0  48   8  21 180   0   8
    0  78]
 [448  28   0   3   1 182   0  55   0   0   2   0  16   2  22 174   8  28
    0  27]
 [133   4   0 371   0 215   0 110   0   0   1   0   8   0   4 134   0   4
    0  10]
 [ 37   1   0 717   0  88   0  33   0   0   0   0  11   0   3 107   1   0
    0   1]
 [ 60  13 451   0   8  32   0  45   0   0   8 233   9   0  12  38   0  40
    0  42]
 [171   0  14   1   7 262   0  69   1   0  16   0   7  33  39 271   1  86
    0   6]
 [137   5   0   1   3 172   2 127   0   0   1   0   6   0  33 334   6  17
   76  70]
 [ 48   1   0   0   0  75   0  36   0   0 290   0   1   1 395 112   0  11
    0  17]
 [ 46   3   0   0   1  61 204  30   0   0   2   0   1   1   2 137 473  10
    0  26]
 [120 112   4   0   1  64   0  83   0   0   2   2  12   0   5  58   1   4
    0 442]
 [ 25   2   0   0   0  62   2  71   0 515   0   0  18   0   1  88   4   5
    0 147]
 [118  61   1   1   0  86   2  46   0   1   4   1  51   0  27  72   6   3
    0 295]
 [ 58  68   1   0   0  57  71  53   0   0   0   0  18   0   5  72 151   2
    2  70]]
Homogeneity Score: 0.315
Completeness Score: 0.353
V-measure Score: 0.333
Adjusted Rand Score: 0.133
Adjusted Mutual Info Score: 0.941
```

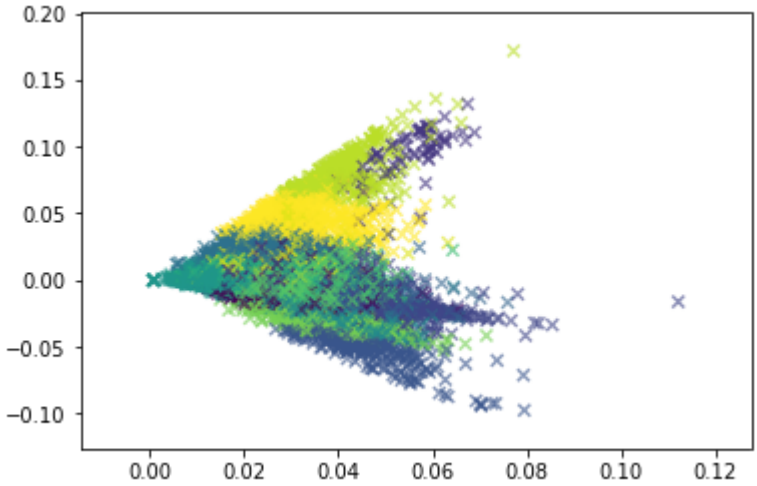


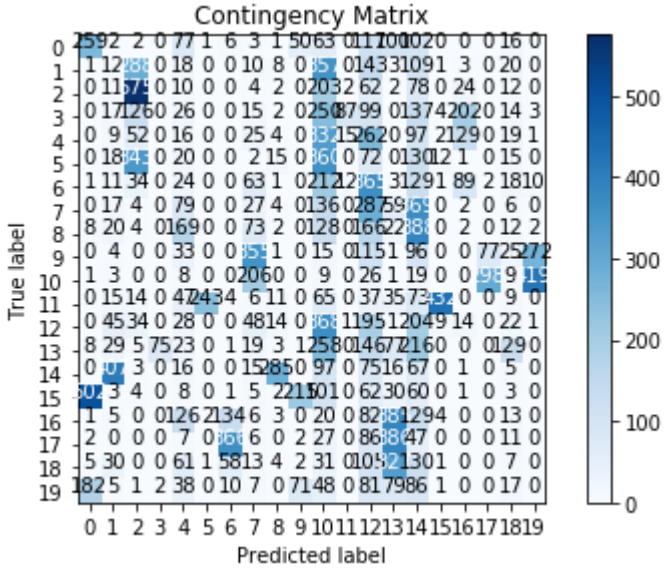


For NMF, firstly try without any transformation

In [27]:
```
km = nmf_pipeline(r, nmf_homo_20, X_20_tfidf, flag_norm = False, n_clusters =
20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[259   2   2   0  77   1   6   3   1  50  63   0 117 100 102   0   0   0
   16   0]
 [  1  12 288   0  18   0   0  10   8   0 357   0 143   3 109   1   3   0
   20   0]
 [  0  11 575   0  10   0   0   4   2   0 203   2  62   2  78   0  24   0
   12   0]
 [  0  17 126   0  26   0   0  15   2   0 250  87  99   0 137   4 202   0
   14   3]
 [  0   9  52   0  16   0   0  25   4   0 332  15 262   0  97   2 129   0
   19   1]
 [  0  18 343   0  20   0   0   2  15   0 360   0  72   0 130  12   1   0
   15   0]
 [  1  11  34   0  24   0   0  63   1   0 212  12 365   3 129   1  89   2
   18  10]
 [  0  17   4   0  79   0   0  27   4   0 136   0 287  59 369   0   2   0
    6   0]
 [  8  20   4   0 169   0   0  73   2   0 128   0 166  22 388   0   2   0
   12   2]
 [  0   4   0   0  33   0   0 355   1   0  15   0 115   1  96   0   0  77
   25 272]
 [  1   3   0   0   8   0   0 206   0   0   9   0  26   1  19   0   0 298
    9 419]
 [  0  15  14   0  47 243   4   6  11   0  65   0  37  35  73 432   0   0
    9   0]
 [  0  45  34   0  28   0   0  48  14   0 368   1 195   1 204   9  14   0
   22   1]
 [  8  29   5  75  23   0   1  19   3   1 258   0 146  77 216   0   0   0
  129   0]
 [  0 407   3   0  16   0   0  15 285   0  97   0  75  16  67   0   1   0
    5   0]
 [502   3   4   0   8   0   1   5   2 215 101   0  62  30  60   0   1   0
    3   0]
 [  1   5   0   0 126   2 134   6   3   0  20   0  82 385 129   4   0   0
   13   0]
 [  2   0   0   0   7   0 366   6   0   2  27   0  86 386  47   0   0   0
   11   0]
 [  5  30   0   0  61   1  58  13   4   2  31   0 105 327 130   1   0   0
    7   0]
 [182   5   1   2  38   0  10   7   0  71  48   0  81  79  86   1   0   0
   17   0]]
Homogeneity Score: 0.317
Completeness Score: 0.359
V-measure Score: 0.337
Adjusted Rand Score: 0.122
Adjusted Mutual Info Score: 0.947
```

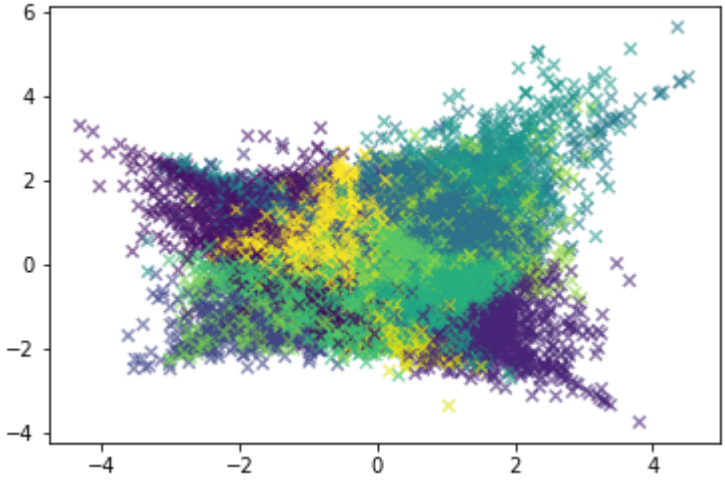


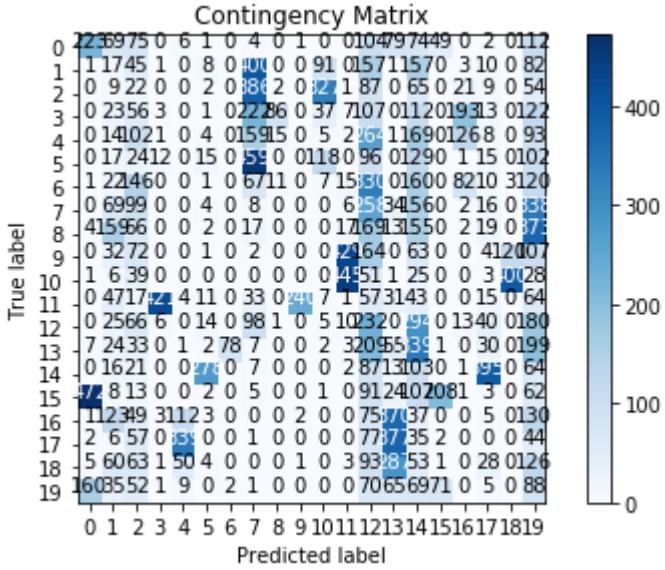


NMF with normalization

In [28]:
```
km = nmf_pipeline(r, nmf_homo_20, X_20_tfidf, flag_norm = True, n_clusters = 20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[223  69  75   0   6   1   0   4   0   1   0   0 104  79  74  49   0   2
    0 112]
 [  1  17  45   1   0   8   0 400   0   0  91   0 157   1 157   0   3  10
    0  82]
 [  0   9  22   0   0   2   0 386   2   0 327   1  87   0  65   0  21   9
    0  54]
 [  0  23  56   3   0   1   0 222  86   0  37   7 107   0 112   0 193  13
    0 122]
 [  0  14 102   1   0   4   0 159  15   0   5   2 264   1 169   0 126   8
    0  93]
 [  0  17  24  12   0  15   0 459   0   0 118   0  96   0 129   0   1  15
    0 102]
 [  1  22 146   0   0   1   0  67  11   0   7  15 330   0 160   0  82  10
    3 120]
 [  0  69  99   0   0   4   0   8   0   0   0   6 258  34 156   0   2  16
    0 338]
 [  4 159  66   0   0   2   0  17   0   0   0  17 169  13 155   0   2  19
    0 373]
 [  0  32  72   0   0   1   0   2   0   0   0 429 164   0  63   0   0   4
  120 107]
 [  1   6  39   0   0   0   0   0   0   0   0 445  51   1  25   0   0   3
  400  28]
 [  0  47  17 421   4  11   0  33   0 240   7   1  57  31  43   0   0  15
    0  64]
 [  0  25  66   6   0  14   0  98   1   0   5  10 232   0 294   0  13  40
    0 180]
 [  7  24  33   0   1   2  78   7   0   0   2   3 209  55 339   1   0  30
    0 199]
 [  0  16  21   0   0 278   0   7   0   0   0   2  87  13 103   0   1 395
    0  64]
 [472   8  13   0   0   2   0   5   0   0   1   0  91  24 107 208   1   3
    0  62]
 [  1 123  49   3 112   3   0   0   0   2   0   0  75 370  37   0   0   5
    0 130]
 [  2   6  57   0 339   0   0   1   0   0   0   0  77 377  35   2   0   0
    0  44]
 [  5  60  63   1  50   4   0   0   0   1   0   3  93 287  53   1   0  28
    0 126]
 [160  35  52   1   9   0   2   1   0   0   0   0  70  65  69  71   0   5
    0  88]]
Homogeneity Score: 0.309
Completeness Score: 0.345
V-measure Score: 0.326
Adjusted Rand Score: 0.119
Adjusted Mutual Info Score: 0.924
```

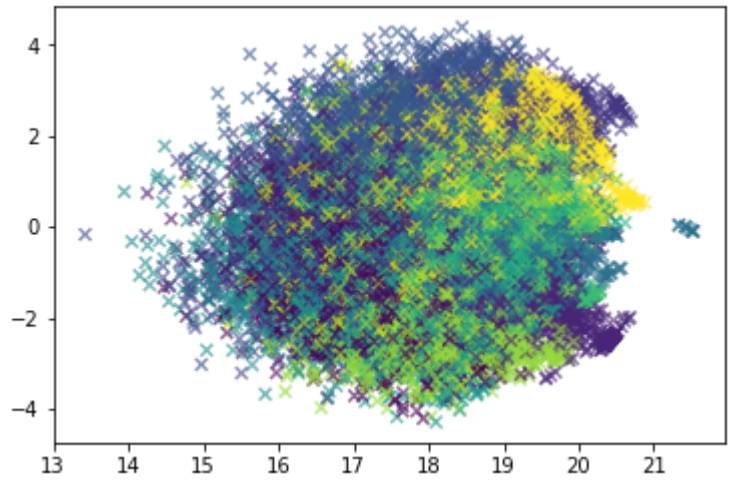


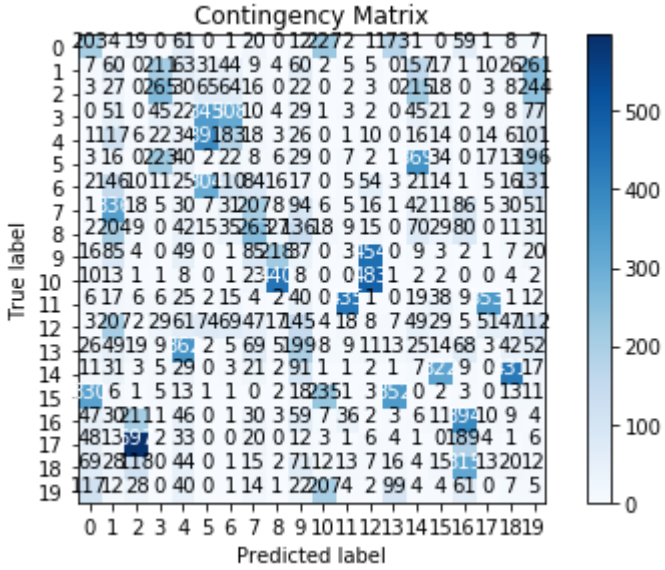


NMF with logrithm

In [29]:
```
km = nmf_pipeline_with_log(r, nmf_homo_20, X_20_tfidf, flag_norm = False, norm
_first = False, n_clusters = 20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[203   4  19   0  61   0   1  20   0  12 227   2   1 173   1   0  59   1
    8   7]
 [  7  60   0 211  63  31  44   9   4  60   2   5   5   0 157  17   1  10
   26 261]
 [  3  27   0 265  30  65  64  16   0  22   0   2   3   0 215  18   0   3
    8 244]
 [  0  51   0  45  22 345 308  10   4  29   1   3   2   0  45  21   2   9
    8  77]
 [  1 117   6  22  34 391 183  18   3  26   0   1  10   0  16  14   0  14
    6 101]
 [  3  16   0 223  40   2  22   8   6  29   0   7   2   1 369  34   0  17
   13 196]
 [  2 146  10  11  25 304 110  84  16  17   0   5  54   3  21  14   1   5
   16 131]
 [  1 336  18   5  30   7  31 207   8  94   6   5  16   1  42  11  86   5
   30  51]
 [  2 204   9   0  42  15  35 263  27 136  18   9  15   0  70  29  80   0
   11  31]
 [ 16  85   4   0  49   0   1  85 218  37   0   3 454   0   9   3   2   1
    7  20]
 [ 10  13   1   1   8   0   1  23 440   8   0   0 483   1   2   2   0   0
    4   2]
 [  6  17   6   6  25   2  15   4   2  40   0 435   1   0  19  38   9 353
    1  12]
 [  3 207   2  29  61  74  69  47  17 145   4  18   8   7  49  29   5  51
   47 112]
 [ 26  49  19   9 362   2   5  69   5 199   8   9  11  13  25  14  68   3
   42  52]
 [ 11  31   3   5  29   0   3  21   2  91   1   1   2   1   7 322   9   0
  431  17]
 [330   6   1   5  13   1   1   0   2  18 235   1   3 352   0   2   3   0
   13  11]
 [ 47  30 211   1  46   0   1  30   3  59   7  36   2   3   6  11 394  10
    9   4]
 [ 48  13 597   2  33   0   0  20   0  12   3   1   6   4   1   0 189   4
    1   6]
 [ 69  28 118   0  44   0   1  15   2  71  12  13   7  16   4  15 315  13
   20  12]
 [117  12  28   0  40   0   1  14   1  22 207   4   2  99   4   4  61   0
    7   5]]
Homogeneity Score: 0.370
Completeness Score: 0.375
V-measure Score: 0.373
Adjusted Rand Score: 0.191
Adjusted Mutual Info Score: 1.107
```

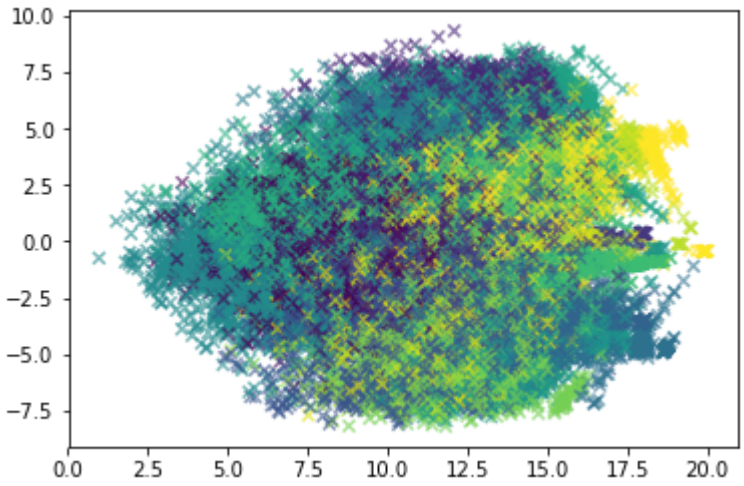


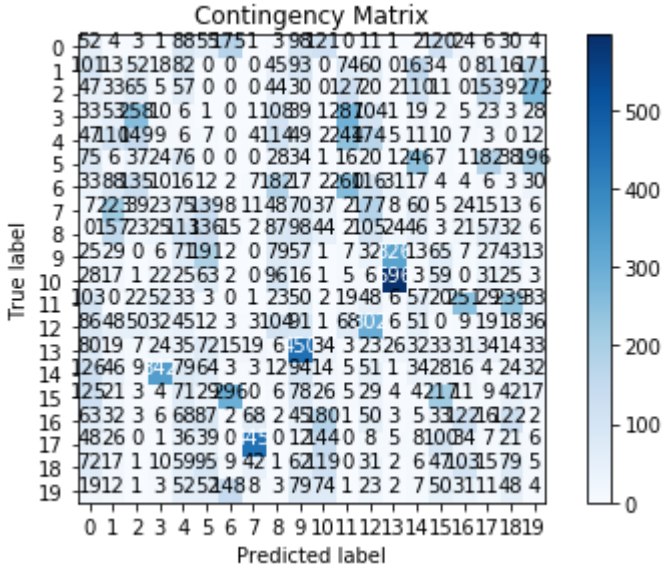


NMF with normalization first and then logrithm

In [30]:
```
km = nmf_pipeline_with_log(r, nmf_homo_20, X_20_tfidf, flag_norm = True, norm_
first = True, n_clusters = 20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[ 52   4   3   1  88  55 175   1   3  98 121   0  11   1   2 120  24   6
   30   4]
 [101  13  52  18  82   0   0   0  45  93   0  74  60   0 163   4   0  81
   16 171]
 [ 47  33  65   5  57   0   0   0  44  30   0 127  20   2 110  11   0 153
    9 272]
 [ 33  53 258  10   6   1   0   1 108  39   1 287 104   1  19   2   5  23
    3  28]
 [ 47 110 149   9   6   7   0   4 114  49   2 244 174   5  11  10   7   3
    0  12]
 [ 75   6  37  24  76   0   0   0  28  34   1  16  20   1 246   7   1 182
   38 196]
 [ 33  88 135  10  16  12   2   7 182  17   2 260 116  31  17   4   4   6
    3  30]
 [  7 223  39  23  75 139   8  11  48  70  37   2 177   8  60   5  24  15
   13   6]
 [  0 157  23  25 113 136  15   2  87  98  44   2 105  24  46   3  21  57
   32   6]
 [ 25  29   0   6  71 191   2   0  79  57   1   7  32 326  13  65   7  27
   43  13]
 [ 28  17   1  22  25  63   2   0  96  16   1   5   6 596   3  59   0  31
   25   3]
 [103   0  22  52  33   3   0   1  23  50   2  19  48   6  57  20 251  29
  239  33]
 [ 86  48  50  32  45  12   3   3 104  91   1  68 302   6  51   0   9  19
   18  36]
 [ 80  19   7  24  35  72  15  19   6 450  34   3  23  26  32  33  31  34
   14  33]
 [126  46   9 342  79  64   3   3  12  94  14   5  51   1  34  28  16   4
   24  32]
 [125  21   3   4  71  29 296   0   6  78  26   5  29   4   4 217  11   9
   42  17]
 [ 63  32   3   6  68  87   2  68   2  45 180   1  50   3   5  33 122  16
  122   2]
 [ 48  26   0   1  36  39   0 445   0  12 144   0   8   5   8 100  34   7
   21   6]
 [ 72  17   1  10  59  95   9  42   1  62 119   0  31   2   6  47 103  15
   79   5]
 [ 19  12   1   3  52  52 148   8   3  79  74   1  23   2   7  50  31  11
   48   4]]
Homogeneity Score: 0.251
Completeness Score: 0.253
V-measure Score: 0.252
Adjusted Rand Score: 0.114
Adjusted Mutual Info Score: 0.750
```

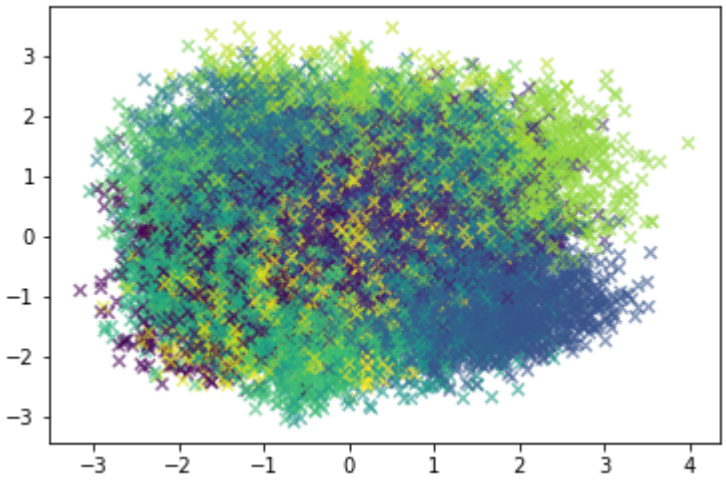


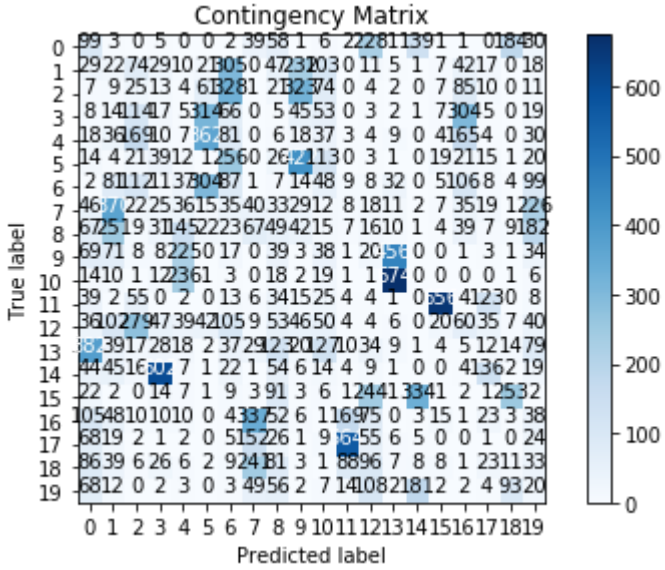


NMF with logrithm first and then normalization

In [31]:
```
km = nmf_pipeline_with_log(r, nmf_homo_20, X_20_tfidf, flag_norm = True, norm_
first = False, n_clusters = 20)
print_cnf_and_score(label_20, km.labels_, classes = np.arange(20))
```

```
Contingency matrix
[[ 99   3   0   5   0   0   2  39  58   1   6   2 228   1 139   1   1   0
  184  30]
 [ 29  22  74  29  10  21 305   0  47 232 103   0  11   5   1   7  42  17
    0  18]
 [  7   9  25  13   4  61 328   1  21 323  74   0   4   2   0   7  85  10
    0  11]
 [  8  14 114  17   5 314  66   0   5  45  53   0   3   2   1   7 304   5
    0  19]
 [ 18  36 169  10   7 362  81   0   6  18  37   3   4   9   0   4 165   4
    0  30]
 [ 14   4  21  39  12   1 256   0  26 422 113   0   3   1   0  19  21  15
    1  20]
 [  2  81 112  11  37 304  87   1   7  14  48   9   8  32   0   5 106   8
    4  99]
 [ 46 370  22  25  36  15  35  40  33  29  12   8  18  11   2   7  35  19
    1 226]
 [ 67 251   9  31 145  22  23  67  49  42  15   7  16  10   1   4  39   7
    9 182]
 [ 69  71   8   8 225   0  17   0  39   3  38   1  20 456   0   0   1   3
    1  34]
 [ 14  10   1  12 236   1   3   0  18   2  19   1   1 674   0   0   0   0
    1   6]
 [ 39   2  55   0   2   0  13   6  34  15  25   4   4   1   0 656   4 123
    0   8]
 [ 36 102 279  47  39  42 105   9  53  46  50   4   4   6   0  20  60  35
    7  40]
 [382  39  17  28  18   2  37  29 123  20 127  10  34   9   1   4   5  12
   14  79]
 [ 44  45  16 602   7   1  22   1  54   6  14   4   9   1   0   0   4 136
    2  19]
 [ 22   2   0  14   7   1   9   3  91   3   6   1 244   1 334   1   2   1
  253   2]
 [105  48  10  10  10   0   4 337  52   6   1 169  75   0   3  15   1  23
    3  38]
 [ 68  19   2   1   2   0   5 152  26   1   9 564  55   6   5   0   0   1
    0  24]
 [ 86  39   6  26   6   2   9 241  81   3   1  88  96   7   8   8   1  23
   11  33]
 [ 68  12   0   2   3   0   3  49  56   2   7  14 108   2 181   2   2   4
   93  20]]
Homogeneity Score: 0.366
Completeness Score: 0.369
V-measure Score: 0.367
Adjusted Rand Score: 0.204
Adjusted Mutual Info Score: 1.093
```




Contingency Matrix
True label
Predicted label