Project 1 Classification Analysis on Textual Data

Yanzhe Xu 404946757 Boan Tao 304946319

Introduction

In this project, different methods for classifying textual data are designed and evaluated. "20 Newsgroups" dataset in scikit-learn package is chosen as target data. First, textual datasets are processed to be proper numerical matrix. Second, the dimension of data was reduced to obtain suitable training models. Third, different classification analysis includes SVM, Naïve Bayes method and Logistic Regression are then implemented to train and test models.

Problem a

In this problem, eight subclasses in subclasses of 'computer technology' and 'Recreational activity' are extracted from training dataset. To make sure the balance in the relative sizes of data sets corresponding to different classes, a histogram and corresponding table of the number of training documents per subclass are plotted which is shown in Figure 1.1 and Table 1.1. From result below, it is obvious that the number of documents in the two classes and eight subclasses are almost evenly distributed. Thus, there is no need to balance further.

Figure 1.1: Number of document per subclasses

Classes	Subclasses	Number of document	Total of classes
Computer technology	comp.graphics	584.0	2343
	comp.os.ms-windows.misc	591.0	
	comp.sys.ibm.pc.hardware	590.0	
	comp.sys.mac.hardware	578.0	
Recreation activity	rec.autos	594.0	2389
	rec.motorcycles	598.0	
	rec.sport.baseball	597.0	
	rec.sport.hockey	600.0	

Table 1.1: Number of document per classes and subclasses

Problem b

In this section, datasets are converted into a numerical feature vector whose element in ith row and jth column represents the frequency of jth term in ith document. Stems of verbs, punctuations, words with number, common stop words are all removed as meaningless terms. It can be achieved by functions 'CountVectorizer' and 'RegexpTokenizer'. Then useful terms are extracted based on the TFxIDF value which represents the importance of a term to a document. By setting different minimum document frequency of vocabulary terms, final number of extracted terms are distinct. The corresponding results are shown in Table 2.1.

Minimum document frequency	Number of extracted terms
2	15806
5	7734

Table 2.1: Number of extracted terms

Problem c

Same as TFxIDF value, TFxICF value quantifies how significant a term is to a class. Based on TFxICF values, 10 most significant terms in 4 different subclasses are found which are shown in Table3.1.

Rank	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware	misc.forsale	soc.religion.christian
1	1	edu	edu	s
2	edu	s	line	god
3	drive	mac	sale	christian
4	ide	line	subject	t
5	s	subject	organ	jesus
6	use	organ	s	edu
7	line	t	post	church
8	com	use	new	subject
9	subject	appl	dos	line
10	t	quadra	univers	say

Table 3.1: Number of extracted terms

Problem d

TFxIDF matrix has a large dimensionality since it converts every term into a number which leads to difficulties in classification. Therefore, LSI representation, a dimensionality reduction transform method based on singular value decomposition, is applied to obtain better representation of the data vector with essential characteristics of the features. 50 largest eigenvalues of the term-document matrix are calculated. Then each document is mapped to corresponding 50-dimensional eigenvector. The result vectors with different minimum document frequency are shown below. The dimension of the new vector becomes 4732×50. Alternatively, another dimensionality reduction method NMF is also applied and corresponding 4732×50 result vector is shown in Figure 4.3.

```
[[ 0.16417821  0.12270783  0.0339819  ..., -0.03555827 -0.01810881
  -0.01990282]
 [ 0.1655636   0.14584745 -0.09470348 ...,  0.05132989 -0.08883468
   0.06446528]
 [ 0.19729055  0.01216909  0.14635102 ..., -0.01290989 -0.00341914
   0.05179125]
 ...,
 [ 0.15714612  0.10499263 -0.04816697 ...,  0.03837035 -0.02390295
   0.05521223]
 [ 0.26331434  0.18611614 -0.00442611 ...,  0.02631201 -0.00612212
  -0.00069303]
 [ 0.12817892 -0.05357632 -0.00652361 ..., -0.02204285 -0.03371208
  -0.01518804]]
```

Figure 4.1: LSI representation with minimum document frequency=2

```
[[ 0.17143541  0.13311969  0.0356247  ..., -0.02895687  0.00348338
  -0.04682354]
 [ 0.17232556  0.15211966 -0.10269586 ...,  0.08779196 -0.09301292
   0.03222251]
 [ 0.19782846  0.01964456  0.14739272 ..., -0.0080711   0.0284855
  -0.0010531 ]
 ...,
 [ 0.16282863  0.11008802 -0.05359731 ..., -0.06182447 -0.01984905
  -0.01134659]
 [ 0.27345231  0.19785406 -0.00610219 ...,  0.05516832 -0.0133895
  -0.0207326 ]
 [ 0.14161337 -0.0581628  -0.00993367 ...,  0.00547741 -0.03053621
  -0.06057657]]
```

Figure 4.2: LSI representation with minimum document frequency=5

```
[[ 0.          0.05633906  0.          ...,  0.          0.          0.        ]
 [ 0.00120594  0.00291585  0.          ...,  0.          0.          0.        ]
 [ 0.          0.00237473  0.06069388 ...,  0.          0.          0.00640659]
 ...,
 [ 0.          0.01992947  0.0030577  ...,  0.          0.01012826  0.        ]
 [ 0.00273018  0.1201482   0.         ...,  0.          0.00304154  0.        ]
 [ 0.00919619  0.          0.00582867 ...,  0.          0.          0.00594448]]
```

Figure 4.3: NMF representation

Problem e

To separate the documents into Computer Technology and Recreational Activity groups, Linear Support Vector Machines (SVM) method is applied. To obtain the trade-off relationship between true positive rate and false positive rate, ROC curve is plotted. Moreover, confusion matrix, accuracy, recall and precision of classifiers with different tradeoff parameter are evaluated and compared to test performances of classifiers. In this section, not only hard margin SVM and soft margin SVM are compared, but also LSI representation and NMF representation are evaluated. Results are shown below.

For hard margin SVM classifier (tradeoff parameter is 1000):

1. LSI representation

Accuracy	0.978412698413
Precision	0.96975308642
Recall	0.988050314465

Table 5.1: Hard margin classifier performance with LSI representation

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1511	49
Actual Num in Recreation Class	19	1571

Table 5.2: Confusion matrix for hard margin classifier performance with LSI representation

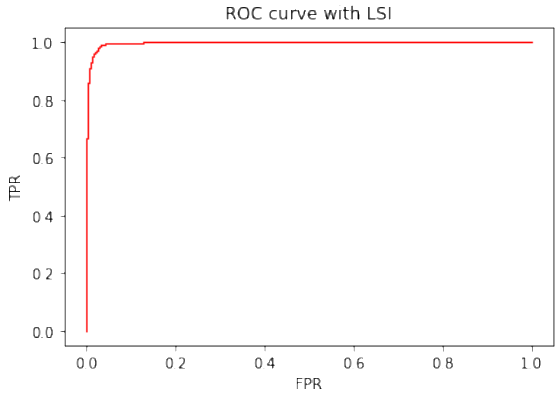

Figure 5.1: ROC curve for hard margin classifier performance with LSI representation

2. NMF representation

Accuracy	0.962222222222
Precision	0.94874923734
Recall	0.977987421384

Table 5.3: Hard margin classifier performance with NMF representation

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1476	84
Actual Num in Recreation Class	35	1555

Table 5.4: Confusion matrix for hard margin classifier performance with NMF representation

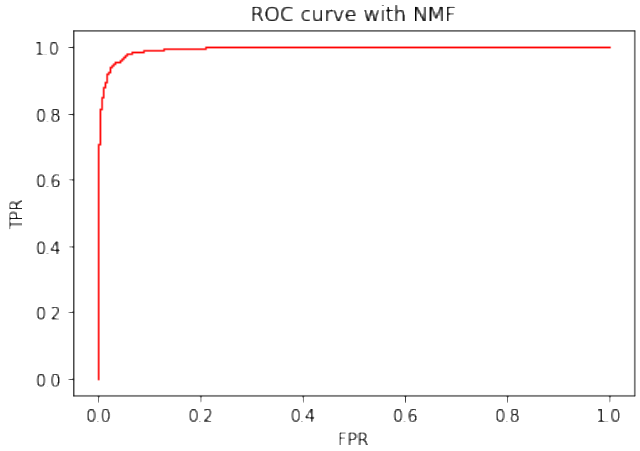

Figure 5.2: ROC curve for hard margin classifier performance with NMF representation

For soft margin SVM classifier (tradeoff parameter is 0.001):

1. LSI representation

Accuracy	0.504761904762
Precision	0.504761904762
Recall	1.0

Table 5.5: Soft margin classifier performance with LSI representation

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	0	1560
Actual Num in Recreation Class	0	1590

Table 5.6: Confusion matrix for soft margin classifier performance with LSI representation

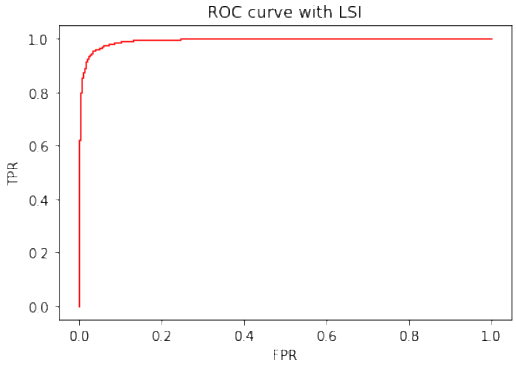

Figure 5.3: ROC curve for soft margin classifier performance with LSI representation

2. NMF representation

Accuracy	0.504761904762
Precision	0.504761904762
Recall	1.0

Table 5.7: Soft margin classifier performance with NMF representation

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	0	1560
Actual Num in Recreation Class	0	1590

Table 5.8: Confusion matrix for soft margin classifier performance with NMF representation

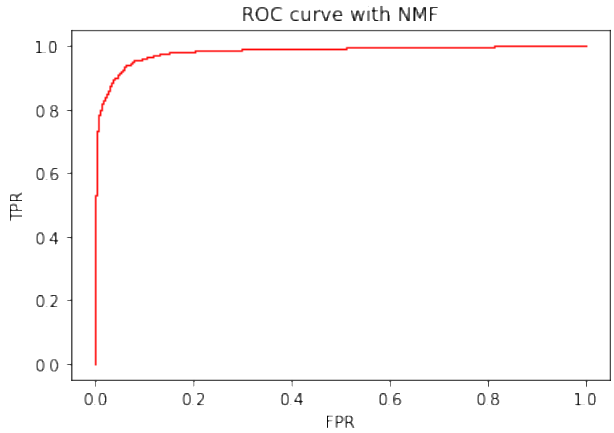

Figure 5.4: ROC curve for soft margin classifier performance with NMF representation

Problem f

In this task, we will be asked using a 5 - fold cross - validation, and then we need to find the best value of the parameter c in the range {0.001, 0.01, 0.1, 1, 10, 100, 1000}, then set the best parameter to SVM classifier and get the accuracy, precision and recall. When we set the penalty coefficient in range {0.001, 0.01, 0.1, 1, 10, 100, 1000}, and then get their according test score. The score shows in Table 6.1.

Penalty Coefficient	Test Score(LSI)	Test Score(NMF)
0.001	0.4981	0.4981
0.01	0.4981	0.4981
0.1	0.4975	0.4975
1	0.9530	0.4994
10	0.9727	0.9277
100	0.9791	0.9600
1000	0.9822	0.9753

Table 6.1 Test score of different penalty coefficient

From Table 6.1, it is obvious that no matter LSI method or NMF method we use to reduce matrix dimension, test score is best when coefficient of SVM classifier is 1000. Then, we set coefficient to be 1000 and use that classifier to classify those documents from 8 topics to 2 main class. Table 6.2 shows other classification result.

	Accuracy	Precision	Recall
LSI	0.9822	0.9703	0.9855
NMF	0.9753	0.9502	0.9717

Table 6.2 Accuracy, precision and recall of SVM classifier with coefficient 1000

In the meantime, the confusion matrix for LSI:

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1512	48
Actual Num in Recreation Class	23	1567

Table 6.3 Confusion matrix with SVM classifier(LSI method)

And the confusion matrix for NMF:

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1479	81
Actual Num in Recreation Class	45	1545

Table 6.4 Confusion matrix with SVM classifier (NMF method)

Compare those results under two different methods, we can find that performance under LSI and NMF are nearly the same.

Problem g

In this task, we are about to use naive Multinomial Bayes algorithm to classify documents into 2 classes and then compare it to SVM classifier. Since if we want to use Multinomial Bayes algorithm to classify documents, we need to use non-negative matrix. With NMF, the dimension-reduced matrix is definitely non-negative. However, when we use LSI to reduce the TF-IDF matrix, we can't guarantee that the dimension-reduced matrix must be non-negative. Therefore, we need to map all of the coefficient in the matrix to (0,1) so that we can get a non-negative matrix. The results for Multinomial Bayes algorithm under LSI and NMF are shown in Table 7.1.

	Accuracy	Precision	Recall
LSI	0.8813	0.8102	0.9987
NMF	0.9342	0.9109	0.9642

Table 7.1 Performance of Multinomial Bayes algorithm

Confusion matrix for LSI:

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1188	372
Actual Num in Recreation Class	2	1588

Table 7.2 Confusion matrix with Multinomial Bayes algorithm(LSI method)

Confusion matrix for NMF:

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1410	150
Actual Num in Recreation Class	57	1533

Table 7.3 Confusion matrix with Multinomial Bayes algorithm(NMF method)

We also get the ROC for both Multinomial Bayes algorithm and SVM(c=1000).

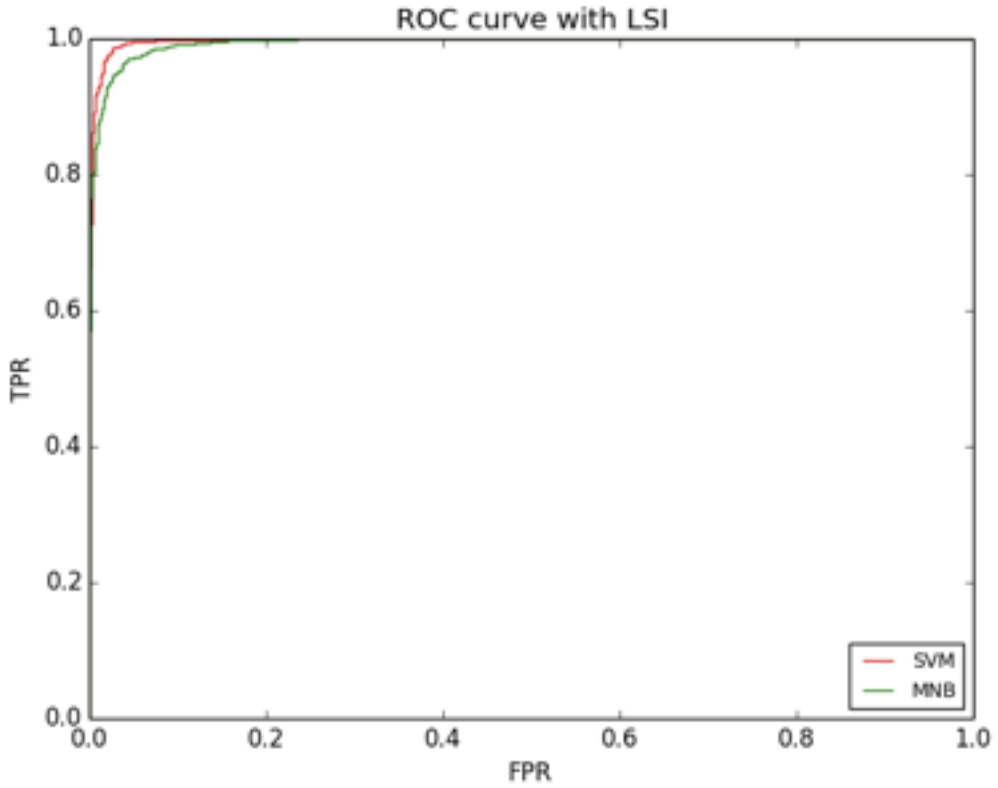

Figure 7.1 ROC curve with Multinomial Bayes algorithm(LSI method)

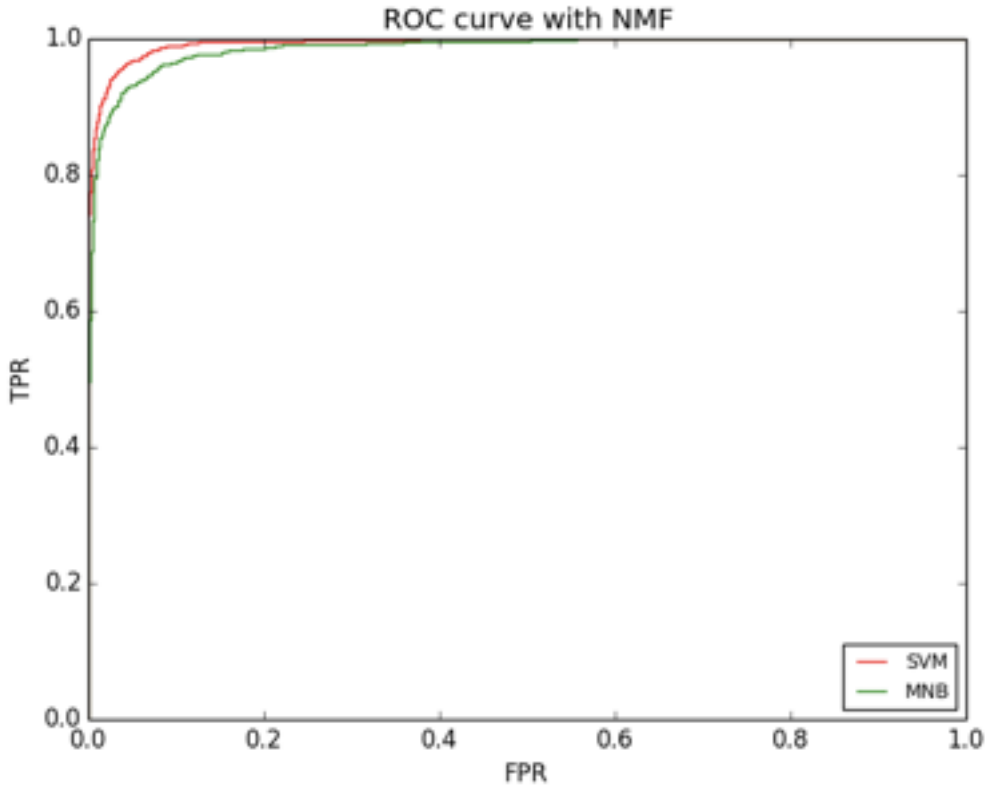

Figure 7.2 ROC curve with Multinomial Bayes algorithm(NMF method)

From their ROC curve, we can tell that SVM classifier has better performance than Multinomial Bayes algorithm. But the difference in performance is pretty limited since Multinomial Bayes classifier has a relatively good performance.

Problem h

In this task, we need to use another classifier, logistic regression classifier, to classify documents into 2 main class and evaluate the performance by comparing it to the other two classifiers. The performance results are shown in Table 8.1.

	Accuracy	Precision	Recall
LSI	0.9708	0.9693	0.9730
NMF	0.9089	0.9059	0.9145

Table 8.1 Performance of logistic regression classifier

Confusion matrix for LSI:

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1511	49
Actual Num in Recreation Class	43	1547

Table 8.2 Confusion matrix with logistic regression classifier(LSI method)

Confusion matrix for NMF:

	Predict Num in Computer Class	Predict Num in Recreation Class
Actual Num in Computer Class	1409	151
Actual Num in Recreation Class	136	1454

Table 8.3 Confusion matrix with logistic regression classifier(NMF method)

And the ROC curve is:

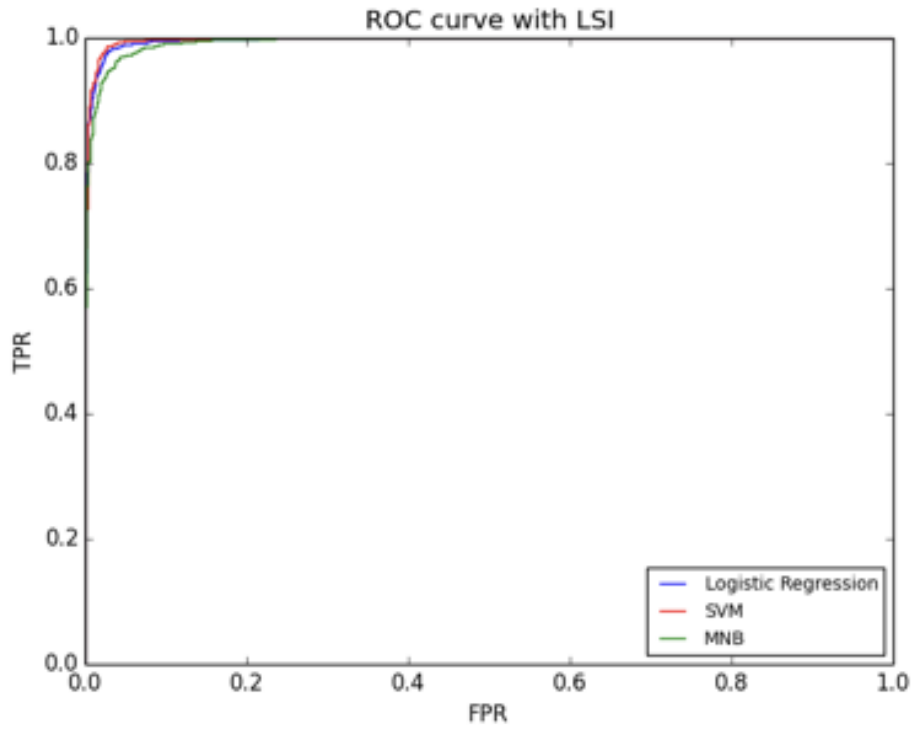

Figure 8.1 ROC curve with logistic regression classifier(LSI method)

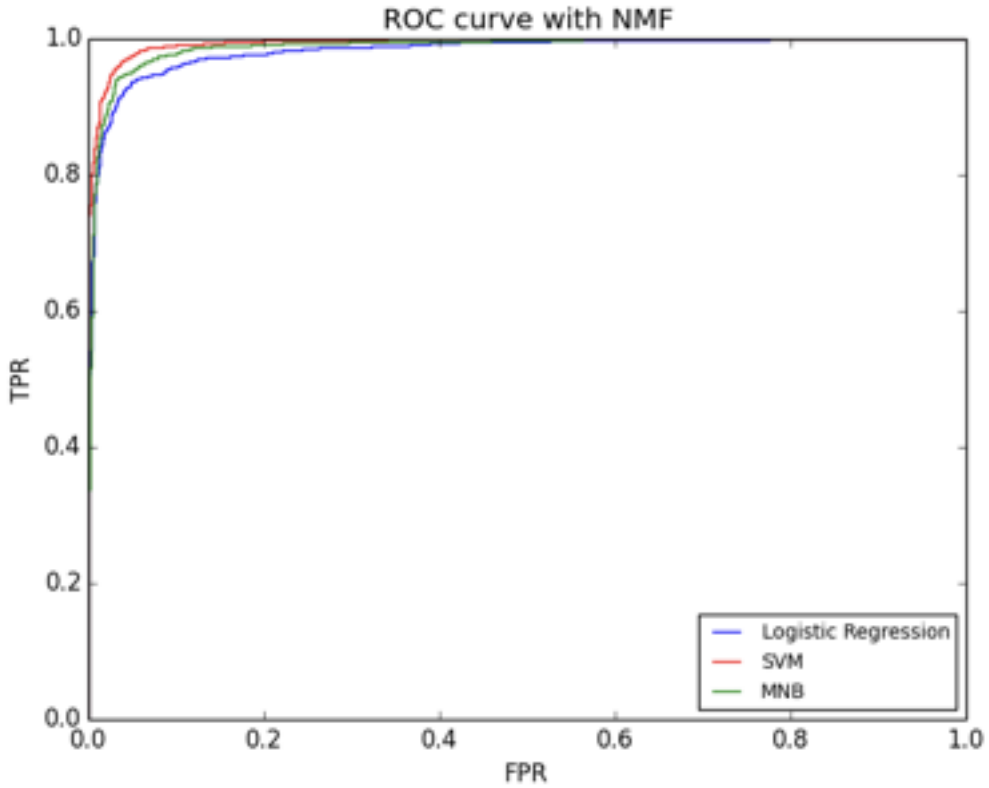

Figure 8.2 ROC curve with logistic regression classifier(NMF method)

From the ROC curve above, it is easy to find that under LSI method, logistic regression classifier and SVM classifier have better performance than Multinomial Bayes. When it comes to NMF method, logistic regression achieve the worst performance of these three algorithms.

Problem i

In this task, we need to use different norm regularizations and sweep through different regularization coefficients to see how do those parameters affect the performance. First, we try to use norm l1 and sweep the penalty coefficient from 0.001 to 1000. To simplify this question, we use accuracy to evaluate the performance. The performance is shown in Table 9.1.

	0.001	0.01	0.1	1	10	100	1000
LSI	0.4952	0.9003	0.9406	0.9689	0.9768	0.9768	0.9768
NMF	0.4952	0.4952	0.7629	0.9644	0.9714	0.9711	0.9708

Table 9.1 Performance of logistic regression classifier(norm='l1')

Then, we change the norm regularization to l2 and sweep the penalty coefficient from 0.001 to 1000. The norm l2's performance is shown in Table 9.2.

	0.001	0.01	0.1	1	10	100	1000

LSI	0.7333	0.9454	0.9600	0.9695	0.9771	0.9775	0.9771
NMF	0.5048	0.5063	0.9241	0.9298	0.9508	0.9629	0.9683

Table 9.2 Performance of logistic regression classifier(norm='l2')

It is easy to notice that the accuracy will increase as penalty coefficient C increase. But when C is too large (like 1000), the performance remain the same or even drops a little. Therefore, in both l1 norm and l2 norm condition, we should set our C to a reasonable level. If C is too small, we will get really bad performance of the classifier. Also, we can't get better performance when C is too large.

Multiclass Problem:

In the last tasks, we only classify documents to 2 classes. However, documents need to be classified into 4 classes in this task by using SVM classifier and Multinomial Bayes algorithm. First, we use One vs One to set SVM and Multinomial Bayes classifier to solve this multi class classification problem and we can get the performance of each classifier shown in the Figure 10.1 and Figure 10.2.

```
svm(One vs One):
('accuracy with LSI is ', 0.87412140575079877)
('precision with LSI is ', 0.87431894547213063)
('recall with LSI is ', 0.87412140575079877)
confusion matrix with LSI is
[[317  46  28   1]
 [ 36 319  30   0]
 [ 26  22 340   2]
 [  3   1   2 392]]
('accuracy with NMF is ', 0.84281150159744411)
('precision with NMF is ', 0.84531843426166442)
('recall with NMF is ', 0.84281150159744411)
confusion matrix with NMF is
[[307  64  21   0]
 [ 68 291  24   2]
 [ 35  19 335   1]
 [  8   3   1 386]]
```

Figure 10.1 Performance of SVM classifier on multi class(One vs Rest)

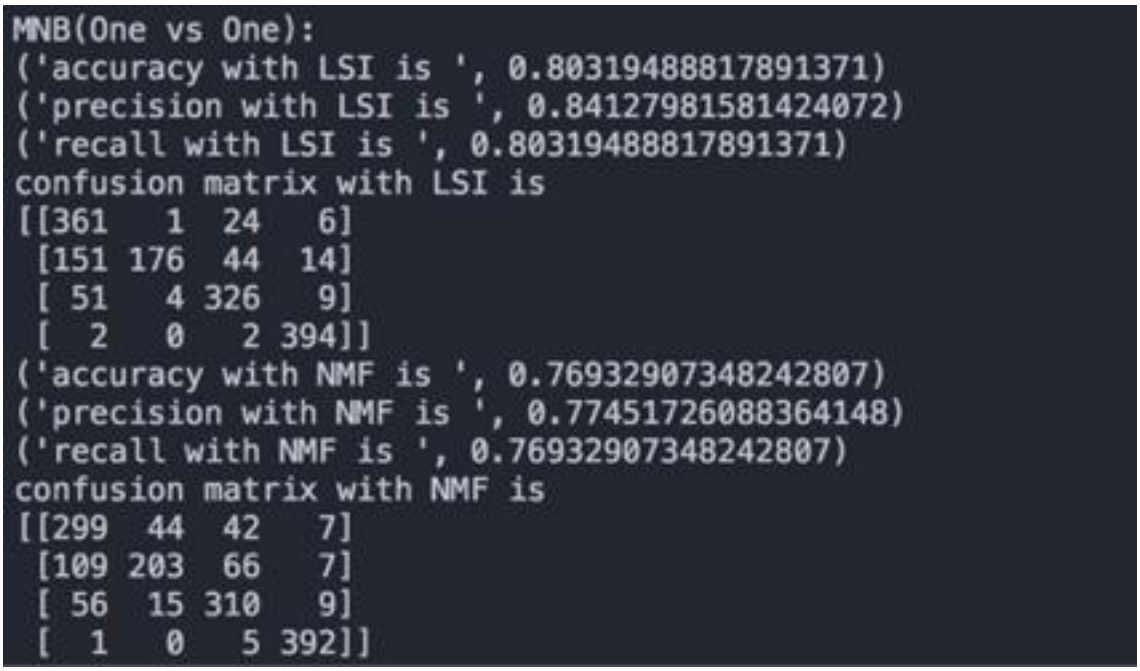

Figure 10.2 Performance of Multinomial Bayes classifier on multi class(One vs One)

And then, we use One vs Rest to set those classifiers and the results are shown in the following.

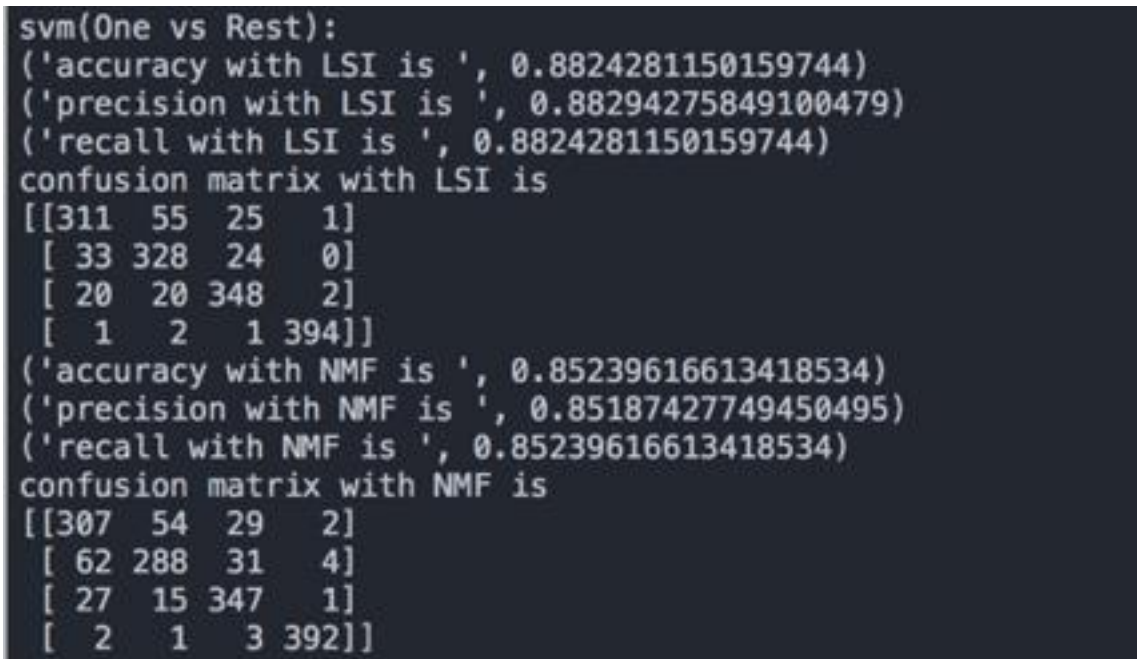

Figure 10.3 Performance of SVM classifier on multi class(One vs Rest)

```
MNB(One vs Rest):
('accuracy with LSI is ', 0.78913738019169333)
('precision with LSI is ', 0.82641709407513952)
('recall with LSI is ', 0.78913738019169333)
confusion matrix with LSI is
[[357   1  26   8]
 [152 162  46  25]
 [ 46   4 319  21]
 [  0   0   1 397]]
('accuracy with NMF is ', 0.77699680511182112)
('precision with NMF is ', 0.78080697082343986)
('recall with NMF is ', 0.77699680511182112)
confusion matrix with NMF is
[[300  44  41   7]
 [105 211  61   8]
 [ 53  17 312   8]
 [  1   0   4 393]]
```

Figure 10.4 Performance of Multinomial Bayes classifier on multi class(One vs Rest)

From the results of two different type of classifier with two different methods(One vs One and One vs Rest), we can see that SVM classifier has better performance than Multinomial Bayes classifier. And the difference between One vs One and One vs Rest in both classifiers is pretty small. The performance under these two methods are almost the same.

However, from all of the confusion matrixes, we can easily tell that class 3 and class 4 have better accuracy than the first two classes. It is easy to understand because the first two classes are all about computer hardware. Therefore, the context in those documents will have many terms in common which might be hard to tell the difference and then it will be easy to make the classification result has a lower accuracy than usual classes.

Conclusion:

All of the methods and classifiers we use to classify documents have pretty good performance, they all have near or greater than 80% accuracy. Most of them can achieve 90% accuracy which means these methods are effective to classify documents.