EE219 Large Scale Data Mining Models and Algorithms

Winter 2018_Project1

Team member:

Dui Lin (504759948)
Xinyi Jiang (904818856)

Content

1. Introduction

Statistical classification refers to the task of identifying a category, from a predefined set, to which a data point belongs, given a training data set with known category memberships. In this project, our goal is to look into different methods for doing classification analysis on the dataset with "20 Newsgroups", which is a collection of approximately 20,000 newsgroup documents.

For the purpose of this project, we are going to concentrate on the 8 of the classes which have been shown in the following table.

Computer technology	Recreational activity
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware	rec.autos rec.motocles rec.sport.baseball rec.sport.hockey

Table 1. Subclasses of 'Computer technology' and 'Recreational activity'

2. Problem Statement and Results

2.1 Question a

Before we carried out the classification analysis, we need to make sure that the dataset is properly balanced with respect to every collection of documents that we are going to use. In order to achieve this goal, we plot three histograms of the number of training documents per class, all the subsets per class and the testing subsets to check if they are evenly distributed.

The historgrams and results haven been showed as below:

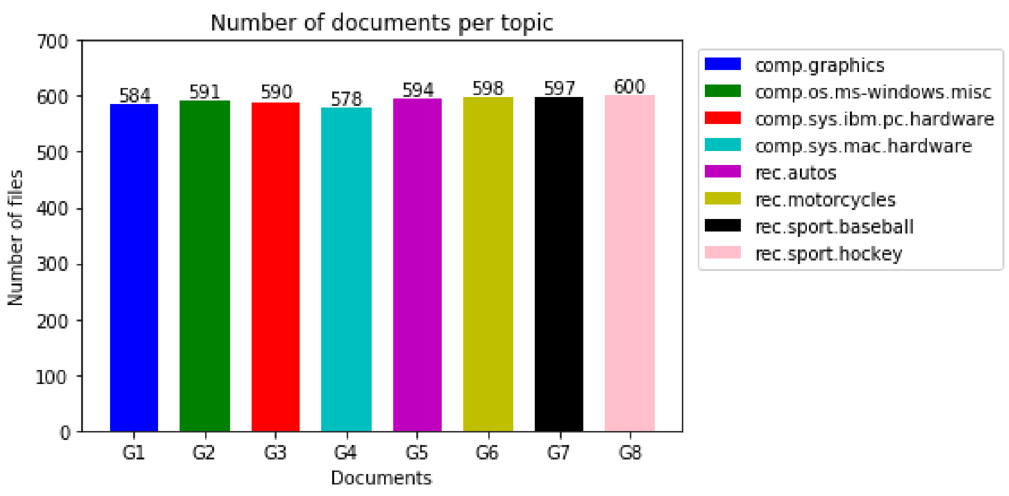


Figure a.1. the histogram of the number of traning documents per class

From the figure above we can collect the data as below:

Newsgroups	Number of documents per topic
comp.graphics	584
comp.os.ms-windows.misc	591
comp.sys.ibm.pc.hardware	590
comp.sys.mac.hardware	578
rec.autos	594
rec.motorcylces	598
rec.sport.baseball	597
rec.sport.hockey	600
Subclass: Computer techonolgy	2343
Subclass: Recreational activity	2389

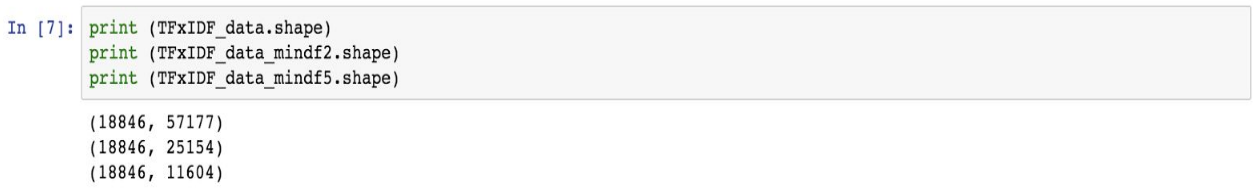
Table a.1. Number of documents per topic

It can be seen that, the number of documents per topic of the training subsets varied from 578 to 600 with quite little variance, which has proved that the subsets are evenly distributed.

2.2 Question b

In this part, a popular numerical statistic to capture the importance of a word to a document in a corpus is the Term Frequency-Inverse Document Frequency (TFxIDF) metric. In order to convert them into numerical feature vectors, we tokenize each document into words firstly and then, excluding the stop words, punctuations and finally create a TFxIDF vector representations. We used two settings for minimum document frequency of vocabulary terms, namely mid_df=2 and mid_df=5.

Here are the results we obtained from the program.



```
In [7]: print (TFxIDF_data.shape)
        print (TFxIDF_data_mindf2.shape)
        print (TFxIDF_data_mindf5.shape)

(18846, 57177)
(18846, 25154)
(18846, 11604)
```

Results	Number of Documents	Number of Terms
Default, min_df = 1	(18846,57177)	57177
min_df=2	(18846,25154)	25154
min_df=5	(18846,11604)	11604

Table b.1. document frequency of vocabulary terms

2.3 Question c

In part C, our goal is to find the most significant terms in each of the following classes with TFxICF measurement.

comp.sys.ibm.pc.hardware, comp.sys.hardeware, misc.forsale, soc.religion.christian

By using the part b measurement, we do the pro-precss the dataset and then count the frequencies of the words. For every class we calculated the data from the “all subsets”, “training subsets”, “test subsets”. We have listed all the results as below.

comp.sys.ibm.pc.hardware		
All subsets	Only training subsets	Only testing subsets
10 most significant terms:	10 most significant terms:	10 most significant terms:
"ani" | 460	"ani" | 287	"ani" | 173
"card" | 537	"card" | 354	"card" | 183
"control" | 429	"control" | 302	"drive" | 230
"disk" | 387	"disk" | 291	"dx" | 133
"drive" | 930	"drive" | 700	"know" | 133
"mb" | 390	"mb" | 289	"modem" | 130
"problem" | 353	"problem" | 218	"problem" | 135
"scsi" | 614	"scsi" | 467	"scsi" | 147
"use" | 742	"use" | 481	"use" | 261
"work" | 363	"work" | 208	"work" | 155
TFxIDF dimension: (982, 10)	TFxIDF dimension: (590, 10)	TFxIDF dimension: (392, 10)

Table c.1. 10 significant terms for “comp.sys.ibm.pc.hardware”

comp.sys.mac.hardware		
All subsets	Only training subsets	Only testing subsets
10 most significant terms:	10 most significant terms:	10 most significant terms:
"ani" | 346	"ani" | 179	"ani" | 167
"appl" | 414	"appl" | 259	"appl" | 155
"drive" | 405	"card" | 170	"disk" | 122
"know" | 290	"drive" | 259	"drive" | 146
"like" | 272	"know" | 176	"mac" | 282
"mac" | 666	"mac" | 384	"mhz" | 121
"monitor" | 264	"mb" | 173	"monitor" | 120
"problem" | 382	"problem" | 232	"problem" | 150
"use" | 576	"use" | 337	"use" | 239
"work" | 316	"work" | 176	"work" | 140
TFxIDF dimension: (963, 10)	TFxIDF dimension: (578, 10)	TFxIDF dimension: (385, 10)

Table c.2. 10 significant terms for “comp.sys.mac.hardware”

misc.forsale		
All subests	Training subsets	Testing subsets
10 most significant terms:	10 most significant terms:	10 most significant terms:
"drive" | 300	"dos" | 205	"appear" | 139
"includ" | 324	"includ" | 224	"drive" | 161
"new" | 461	"new" | 278	"new" | 183
"offer" | 380	"offer" | 245	"offer" | 135
"pleas" | 297	"pleas" | 164	"pleas" | 133
"price" | 312	"price" | 194	"price" | 118
"sale" | 414	"sale" | 251	"sale" | 163
"sell" | 268	"sell" | 162	"ship" | 126
"ship" | 292	"ship" | 166	"st" | 136
"use" | 344	"use" | 225	"use" | 119
TFxIDF dimension: (975, 10)	TFxIDF dimension: (585, 10)	TFxIDF dimension: (390, 10)

Table c.3. 10 significant terms for “misc.forsale”

soc.religion.christian		
All subsets	Training subsets	Testing subsets
10 most significant terms:	10 most significant terms:	10 most significant terms:
"believ" | 639	"believ" | 396	"christ" | 263
"christian" | 1053	"christian" | 602	"christian" | 451
"church" | 723	"church" | 384	"church" | 339
"god" | 1936	"god" | 1119	"god" | 817
"jesus" | 685	"jesus" | 468	"homosexu" | 386
"know" | 628	"know" | 374	"know" | 254
"peopl" | 733	"peopl" | 442	"peopl" | 291
"say" | 730	"say" | 448	"say" | 282
"th" | 637	"th" | 367	"sin" | 347
"think" | 597	"think" | 383	"th" | 270
TFxIDF dimension: (997, 10)	TFxIDF dimension: (599, 10)	TFxIDF dimension: (398, 10)

Table c.4. 10 significant terms for “soc.religion.christian”

We can easily conclude the 10 significant terms for each class from the table. What’more, it can also be concluded that the significant terms for “all subsets” “training subsets” “testing subsets” are same to each other.

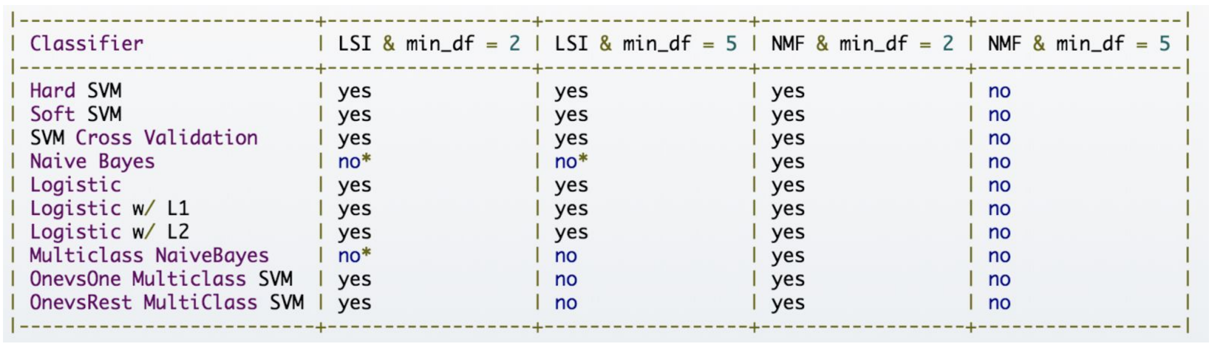
2.4 Question d

After all the operations we have done before, the dimensionality of the representation vectors still very large. Thus in this part, we use Latent Semantic Indexing (LSI) to minimize mean squared residual between the original data and reconstruction from its low dimensional approximation.

We apply LSI to TFxIDF matrix and map each document to a 50-dimensinal vector.

2.5 Question e

Start from part e, our task is to separate the documents into two categories 'Computer Technology' vs 'Recreational Activity'. We use the following table as the reference for our comparison among different classifiers and dimension reduction datasets.



Classifier	LSI & min_df = 2	LSI & min_df = 5	NMF & min_df = 2	NMF & min_df = 5
Hard SVM	yes	yes	yes	no
Soft SVM	yes	yes	yes	no
SVM Cross Validation	yes	yes	yes	no
Naive Bayes	no*	no*	yes	no
Logistic	yes	yes	yes	no
Logistic w/ L1	yes	yes	yes	no
Logistic w/ L2	yes	yes	yes	no
Multiclass NaiveBayes	no*	no	yes	no
OnevsOne Multiclass SVM	yes	no	yes	no
OnevsRest MultiClass SVM	yes	no	yes	no

Figure e.1. Classifiers and Term Matrices

Linear Support Vector Machines have been proved efficient when dealing with sparse high dimensional datasets. And in the process of classification, the Support Vector Machine (SVM) gave us the hyperplane with the maximized margin, where support vector means the data samples that are closest to the hyperplane. In order to characterize the trade-off between the two quantities, we have plot of ROV curve and we also evaluated the classification accuracy by using several parameters, such precision, recall, etc.

Our group have plot ROV curves and calculate the related statistic for LSI_mind2 data with both hard and soft SVM classifier and LSI_mind5 data with both hard and soft SVM classifier.

Predict LSI_mindf2 data with linear SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.89      0.98      0.93      1590
    Rec Act       0.98      0.88      0.92      1560

avg / total       0.93      0.93      0.93      3150

=======================================================

Confusion Matrix:
==============
[[1555   35]
 [ 194 1366]]
==============

Total accuracy:
0.927301587302
```

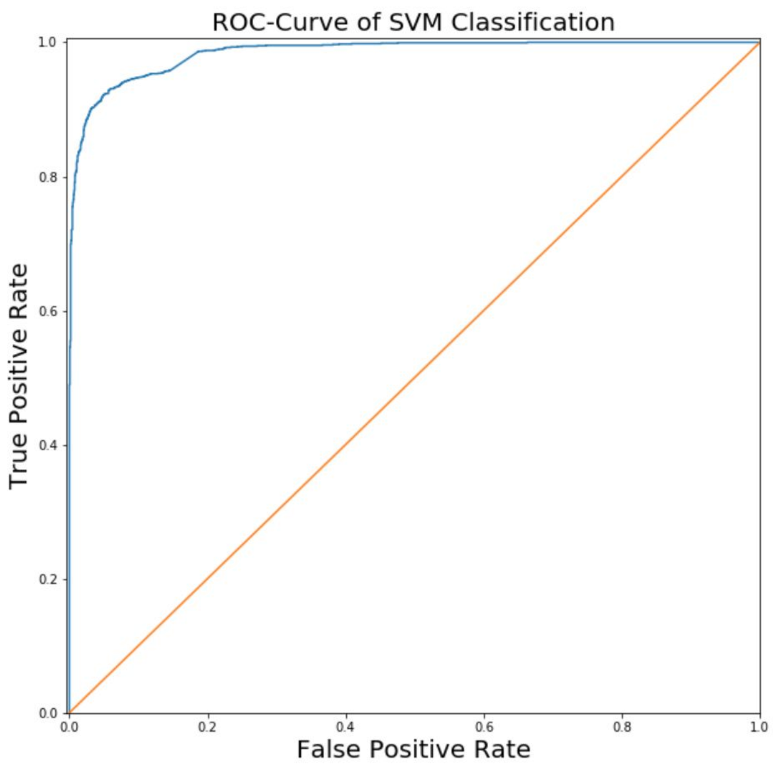


Figure e.2 Roc-Curve of SVM classification for LSI_mindf2 data with linear SVM classifier

Predict LSI_mindf2 data with hard SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.51      1.00      0.67      1590
    Rec Act       0.77      0.01      0.03      1560

avg / total       0.64      0.51      0.35      3150

=======================================================

Confusion Matrix:
==============
[[1584    6]
 [1540   20]]
==============

Total accuracy:
0.509206349206
```

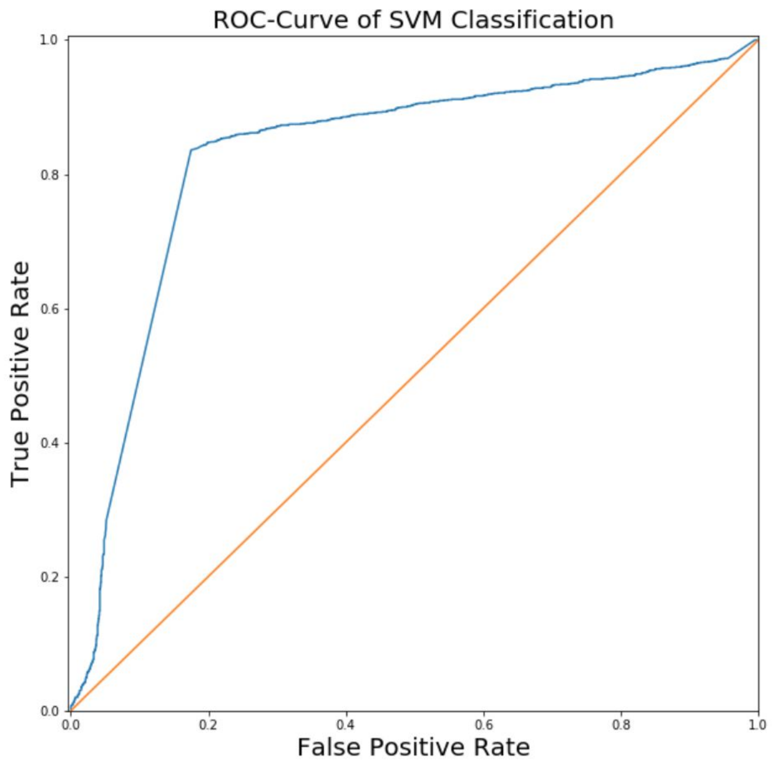


Figure e.3 Roc-Curve of SVM classification for LSI_mindf2 data with hard SVM classifier

Predict LSI_mindf2 data with soft SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.50      1.00      0.67      1590
    Rec Act       0.00      0.00      0.00      1560

avg / total       0.25      0.50      0.34      3150

=======================================================

Confusion Matrix:
==============
[[1590    0]
 [1560    0]]
==============

Total accuracy:
0.504761904762
```

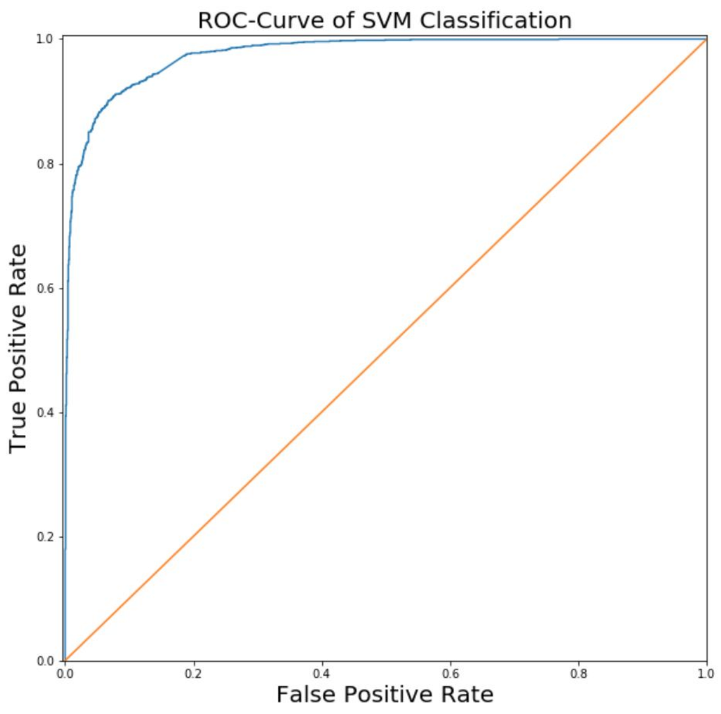


Figure e.4 Roc-Curve of SVM classification for LSI_mindf2 data with soft SVM classifier

Predict LSI_mindf5 data with linear SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.90      0.98      0.93      1590
    Rec Act       0.97      0.88      0.93      1560

avg / total       0.93      0.93      0.93      3150

=======================================================

Confusion Matrix:
==============
[[1552   38]
 [ 182 1378]]
==============

Total accuracy:
0.930158730159
```

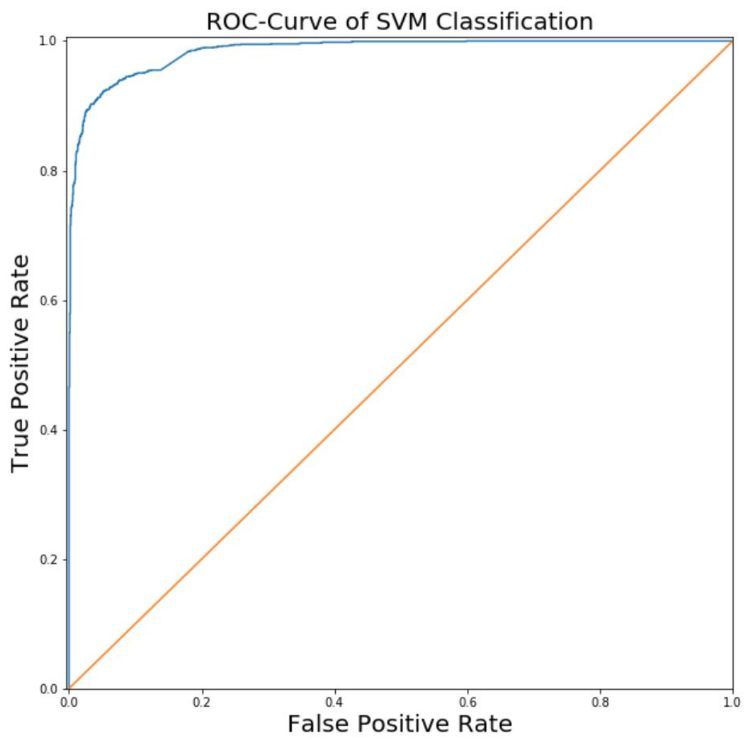


Figure e.5 Roc-Curve of SVM classification for LSI_mindf5 data with linear SVM classifier

Predict LSI_mindf5 data with hard SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.51      1.00      0.67      1590
    Rec Act       0.79      0.01      0.02      1560

avg / total       0.65      0.51      0.35      3150

=======================================================

Confusion Matrix:
==============
[[1586    4]
 [1545   15]]
==============

Total accuracy:
0.508253968254
```

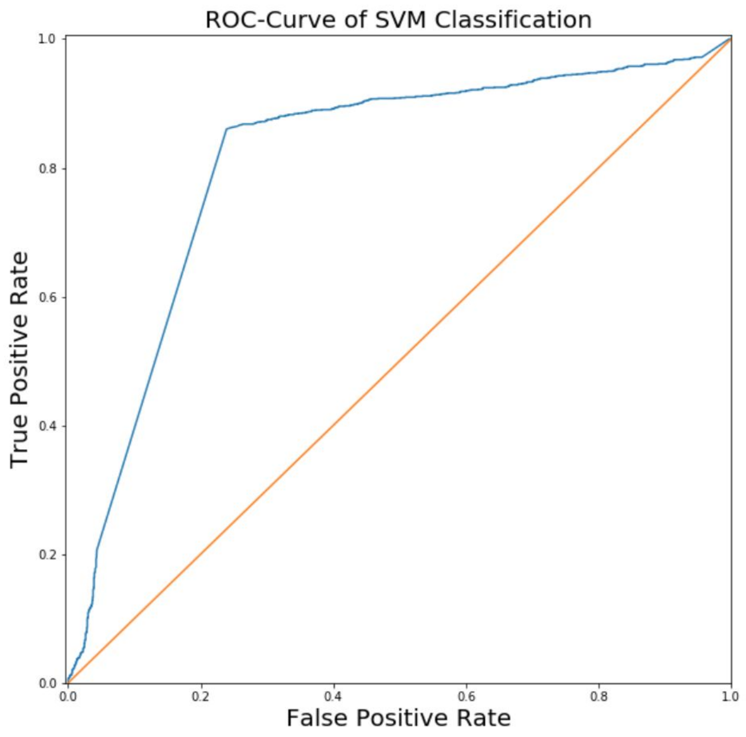


Figure e.6 Roc-Curve of SVM classification for LSI_mindf5 data with hard SVM classifier

Predict LSI_mindf5 data with soft SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.50      1.00      0.67      1590
    Rec Act       0.00      0.00      0.00      1560

avg / total       0.25      0.50      0.34      3150

=======================================================

Confusion Matrix:
==============
[[1590    0]
 [1560    0]]
==============

Total accuracy:
0.504761904762
```

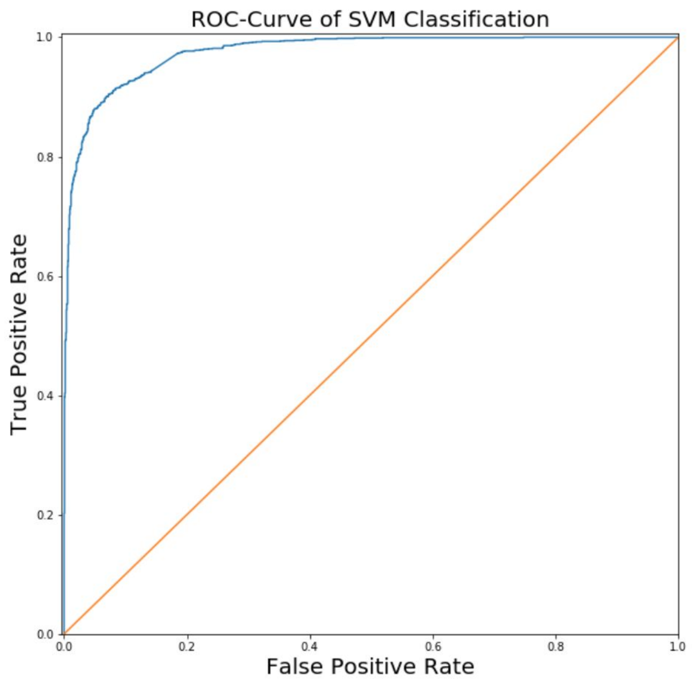


Figure e.7 Roc-Curve of SVM classification for LSI_mindf5 data with soft SVM classifier

Predict NMF_mindf2_data with linear SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.83      0.97      0.89      1590
    Rec Act       0.96      0.79      0.87      1560

avg / total       0.89      0.88      0.88      3150

=======================================================

Confusion Matrix:
==============
[[1536   54]
 [ 320 1240]]
==============

Total accuracy:
0.88126984127
```

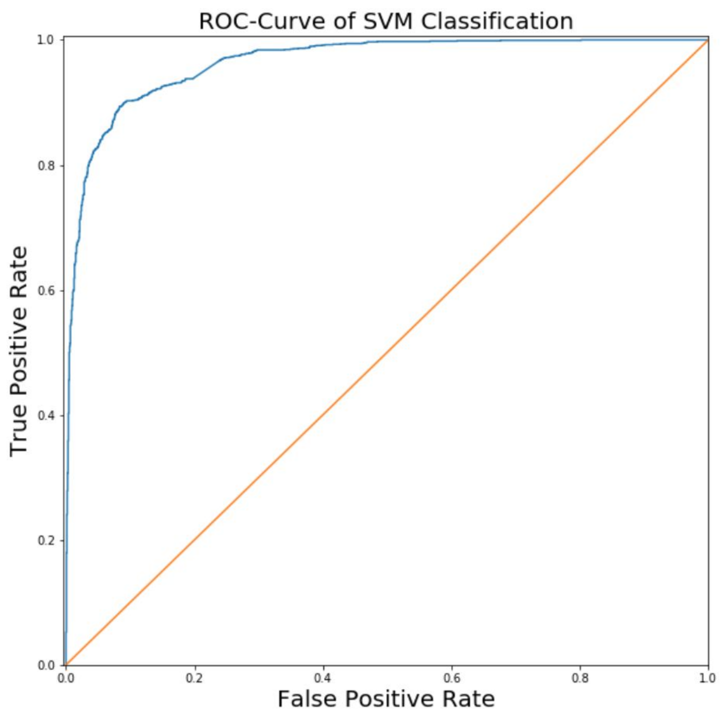


Figure e.8 Roc-Curve of SVM classification for NMP_mindf2 data with linear SVM classifier

Predict NMF_mindf2_data with hard SVM classifier

```
Classification report:
=======================================================
              precision    recall  f1-score   support

   Com Tech       0.89      0.72      0.80      1590
    Rec Act       0.76      0.91      0.83      1560

avg / total       0.83      0.82      0.82      3150

=======================================================

Confusion Matrix:
==============
[[1150  440]
 [ 136 1424]]
==============

Total accuracy:
0.817142857143
```

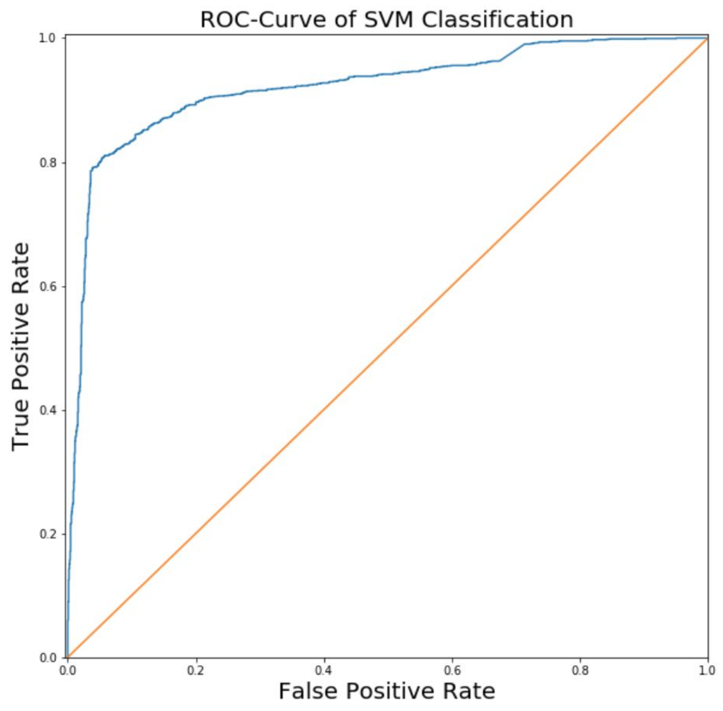


Figure e.9 Roc-Curve of SVM classification for NMP_mindf2 data with hard SVM classifier

Predict NMF_mindf2_data with soft SVM classifier

```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.50      1.00      0.67      1590
    Rec Act       0.00      0.00      0.00      1560

avg / total       0.25      0.50      0.34      3150

=======================================================

Confusion Matrix:
==============
[[1590    0]
 [1560    0]]
==============

Total accuracy:
0.504761904762
```

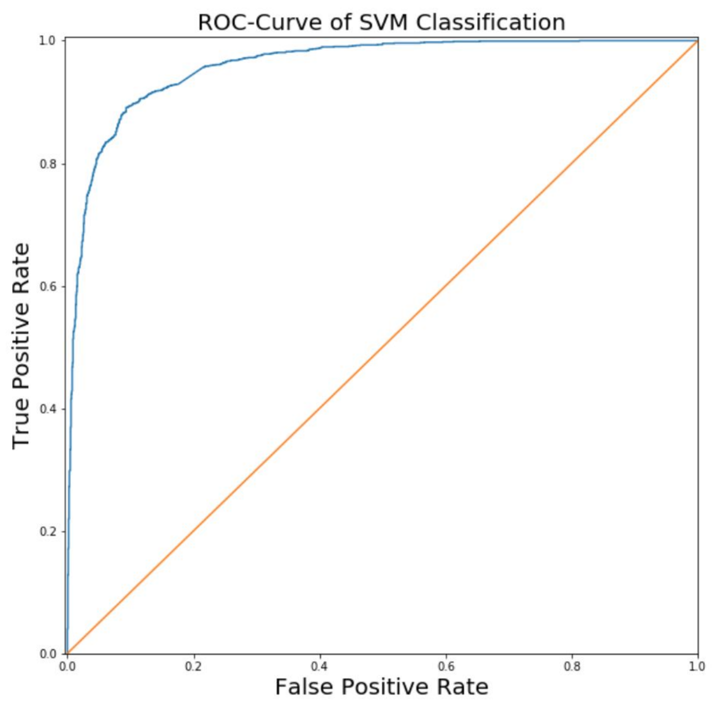


Figure e.10 Roc-Curve of SVM classification for NMP_mindf2 data with soft SVM classifier

As we can see from the data results above, the linear SVM classification has the accuracy with 93% of LSI data while the NMF data has a accuracy with 88%, which means in some particular cases, just like this project, LSI method seems better than NMF. We can also observe that it makes no difference for the choices between 2 and 5 for minimum document frequency of vocabulary terms regarding the accuracy results

2.6 Question f

We then used five-fold cross-validation to find the de best value of parameter gamma between 10^-3 to 10^3.

Firstly split the data into 5 folds, and then we build 5*7 SVM classifiers. Here are the total statistic results we got.

For LSI_mindf2

γ = 0.001 , Total accuracy = 0.504821111393

γ = 0.01 , Total accuracy = 0.854478558741
γ = 0.1 , Total accuracy = 0.91398122304
γ = 1 , Total accuracy = 0.931870083735
γ = 10 , Total accuracy = 0.939101750825
γ = 100 , Total accuracy = 0.793580309566
γ = 1000 , T otal accuracy = 0.544912458767

For LSI_mindf5

γ = 0.001 , Total accuracy = 0.504821111393
γ = 0.01 , Total accuracy = 0.867292565339
γ = 0.1 , Total accuracy = 0.915376807917
γ = 1 , Total accuracy = 0.93212382644
γ = 10 , Total accuracy = 0.937706165948
γ = 100 , Total accuracy = 0.765922354732
γ = 1000 , Total accuracy = 0.525628013195

For NMF_mindf2

γ = 0.001 , Total accuracy = 0.504821111393
γ = 0.01 , Total accuracy = 0.504821111393
γ = 0.1 , Total accuracy = 0.767317939609
γ = 1 , Total accuracy = 0.894062420705
γ = 10 , Total accuracy = 0.915123065212
γ = 100 , Total accuracy = 0.92539964476
γ = 1000 , Total accuracy = 0.834306013702

We total got 21 groups data for the three cases. From the numerical results above, we can see that when we changing the gamma value, the classification accuracy increases firstly then decreases. While as the gamma increases, we found that the classification accuracy falls within an small range when gamma is between 0.1-10 for LSI and between 10-100 for NMF, where accuracy peaks when gamma equals to 10 for LSI and 100 for NMF respectively.
Here are the details of the peak values we got for those three cases:

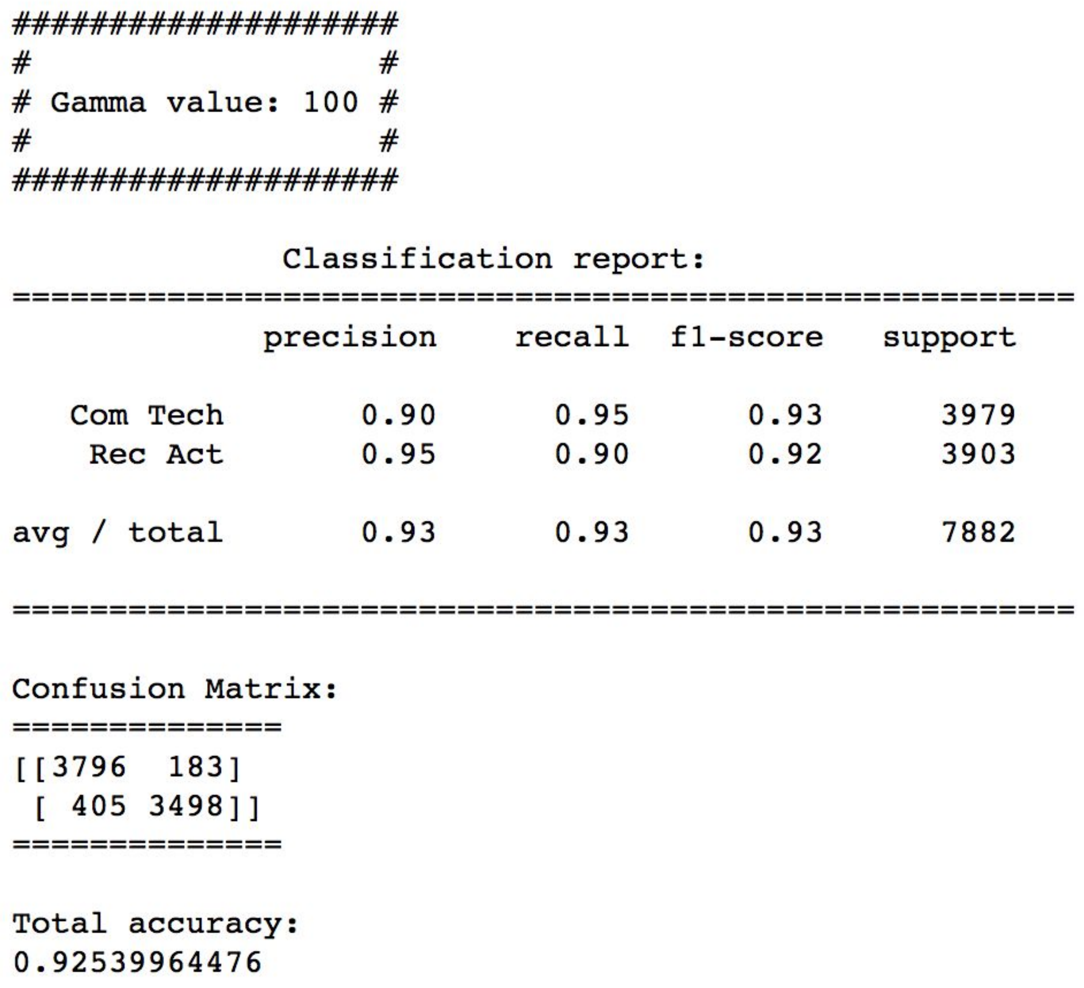
LSI_mindf2

```
###################
#                 #
# Gamma value: 10 #
#                 #
###################

              Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.92      0.97      0.94      3979
    Rec Act       0.96      0.91      0.94      3903

avg / total       0.94      0.94      0.94      7882

=======================================================

Confusion Matrix:
==============
[[3840  139]
 [ 341 3562]]
==============

Total accuracy:
0.939101750825
```

LSI_mindf5

```
###################
#                 #
# Gamma value: 10 #
#                 #
###################

              Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.92      0.96      0.94      3979
    Rec Act       0.96      0.91      0.94      3903

avg / total       0.94      0.94      0.94      7882

=======================================================

Confusion Matrix:
==============
[[3830  149]
 [ 342 3561]]
==============

Total accuracy:
0.937706165948
```

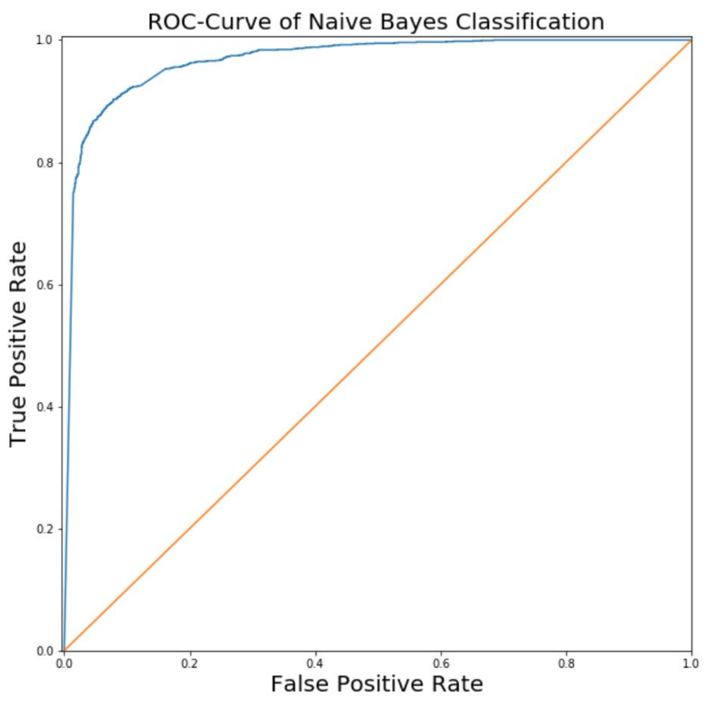
NMF_mindf2



```
####################
#                  #
# Gamma value: 100 #
#                  #
####################

              Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.90      0.95      0.93      3979
    Rec Act       0.95      0.90      0.92      3903

avg / total       0.93      0.93      0.93      7882

=======================================================

Confusion Matrix:
==============
[[3796  183]
 [ 405 3498]]
==============

Total accuracy:
0.92539964476
```

2.7 Question g

Part g requires us to perform the classification task by using the naïve bayes algorithm. In the program we build a Gaussian classifier and we can easily run the code and build the model. Here are the ROC curve and the numerical results:



Naive Bayes Classification

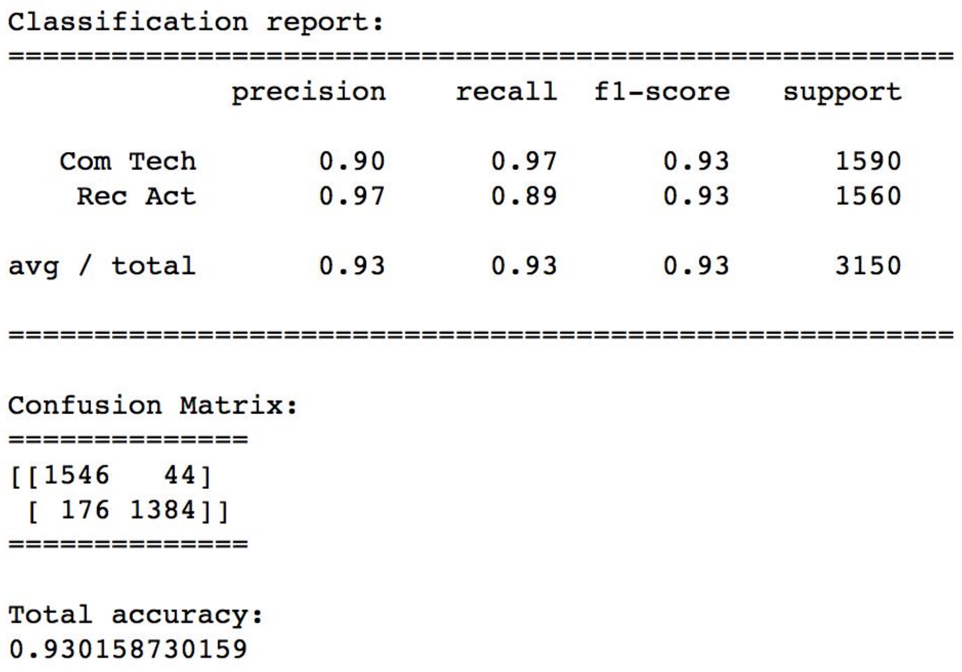
```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.85      0.97      0.91      1590
    Rec Act       0.97      0.83      0.89      1560

avg / total       0.91      0.90      0.90      3150

=======================================================

Confusion Matrix:
==============
[[1545   45]
 [ 269 1291]]
==============

Total accuracy:
0.900317460317
```

Compared with other classification method, we can see that the naive bayes algorithm has lower accuracy which mean it performs worse.

2.8 Question h

In this question, we use the logistic regression classifier for the same task and as we know, this classifier has outstanding performance in linear regression. Firstly, we explore the general (L2 regularized as default) logistic regression with corresponding library in sklearn.linear_model package, and results are as following:

LSI_mindf2 data



```
Classification report:
=======================================================
             precision    recall  f1-score   support

   Com Tech       0.90      0.97      0.93      1590
    Rec Act       0.97      0.89      0.93      1560

avg / total       0.93      0.93      0.93      3150

=======================================================

Confusion Matrix:
==============
[[1546   44]
 [ 176 1384]]
==============

Total accuracy:
0.930158730159
```

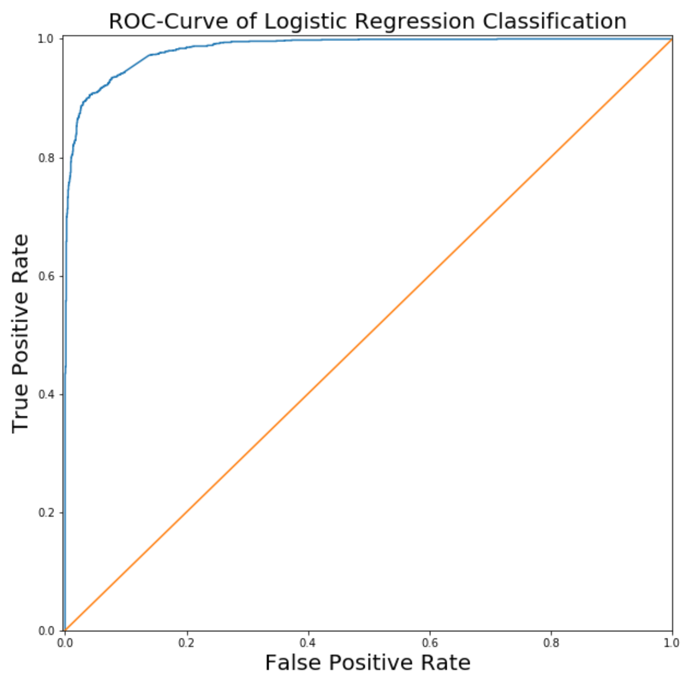


Figure h.1 Roc-Curve of Logistic Regression Classification for LSI data with min_df = 2

LSI_mindf5 data

```
Classification report:
=======================================================
                 precision    recall  f1-score   support

    Com Tech          0.90      0.97      0.93      1590
     Rec Act          0.97      0.89      0.93      1560

avg / total           0.93      0.93      0.93      3150

=======================================================

Confusion Matrix:
==============
[[1544   46]
 [ 172 1388]]
==============

Total accuracy:
0.930793650794
```

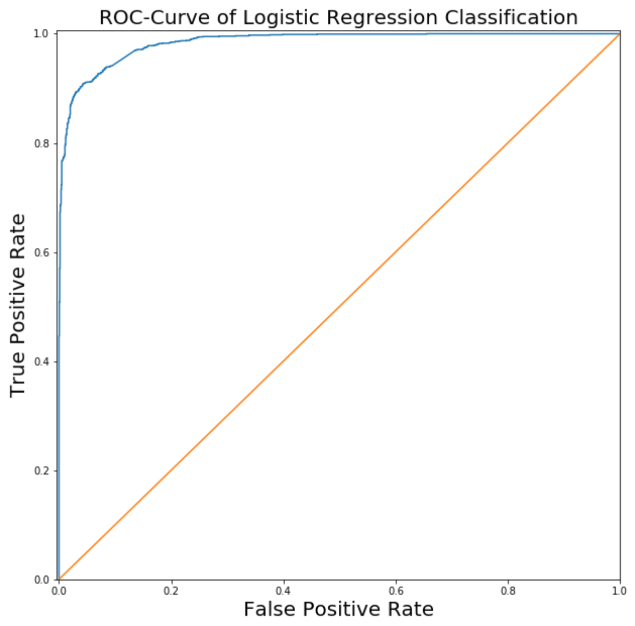


Figure h.2 Roc-Curve of Logistic Regression Classification for LSI data with min_df = 5

NMF_mindf2 data

```
Classification report:
=======================================================
              precision    recall  f1-score   support

   Com Tech        0.85      0.95      0.90      1590
    Rec Act        0.94      0.83      0.88      1560

avg / total        0.89      0.89      0.89      3150

=======================================================

Confusion Matrix:
==============
[[1510   80]
 [ 270 1290]]
==============

Total accuracy:
0.888888888889
```

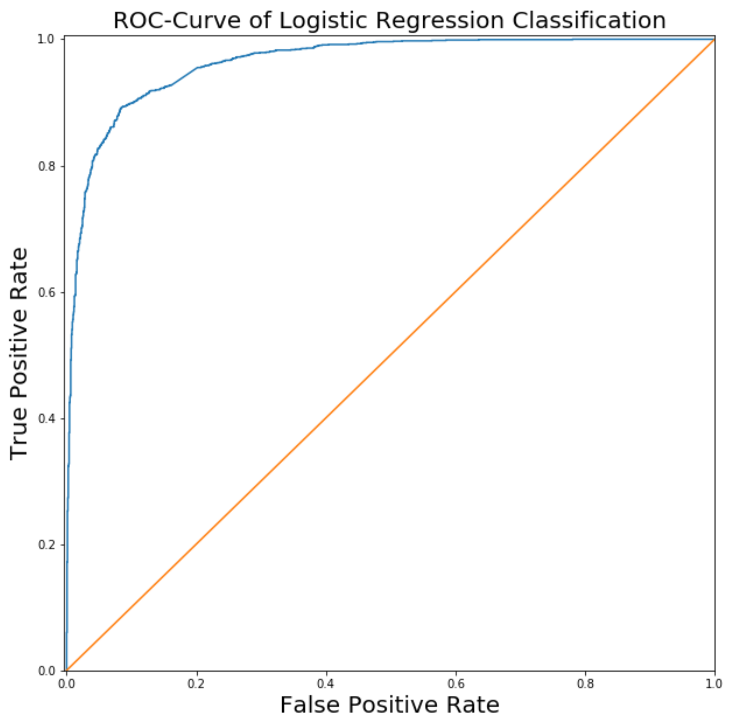


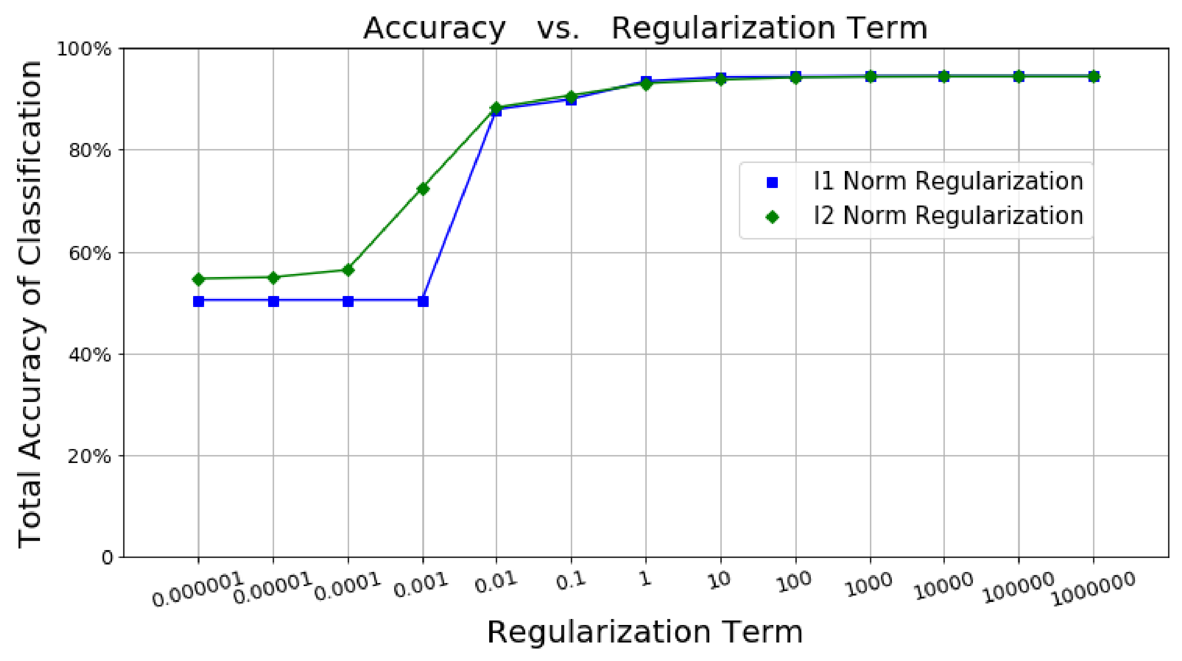
Figure h.3 Roc-Curve of Logistic Regression Classification for NMF data with min_df = 2

From the comparison of the ROC curves and accuracy results, we find that logistic regression still performs very well in the binary classification case, giving the total accuracy of approximately 0.93 on LSI data and 0.89 on NMF data, respectively. This results are almost as good as what the linear SVM classifier produces. We also observe that it makes no difference for the choices between 2 and 5 for minimum document frequency of vocabulary terms regarding the accuracy results. And in this case, we find that LSI outperforms NMF dimension reduction method about 5% regarding the accuracy results, it is probably because SVD factorization is more suitable for matrix that has both positive and negative entries.

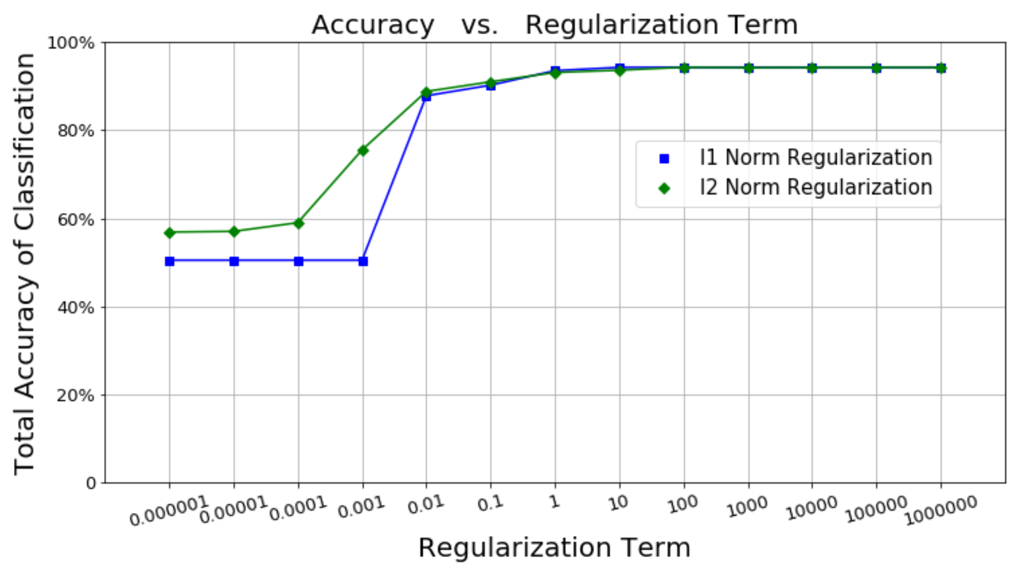
2.9 Question i

Following up the results in question h, we now explore the classification performance of the general logistic regression which is default using L2 regularization with penalty parameter C = 1. In this part, we compare the classification performance of L1 and L2 norm regularized logistic regression with different penalty parameter from 0.000001 to 100000. L1 and L2 regularization are two closely related techniques that can be used by machine learning training algorithms to reduce model overfitting. Eliminating overfitting will lead to a model which has better generalized performance. We plot the correlation between regularization term and total accuracy of classification as follows:

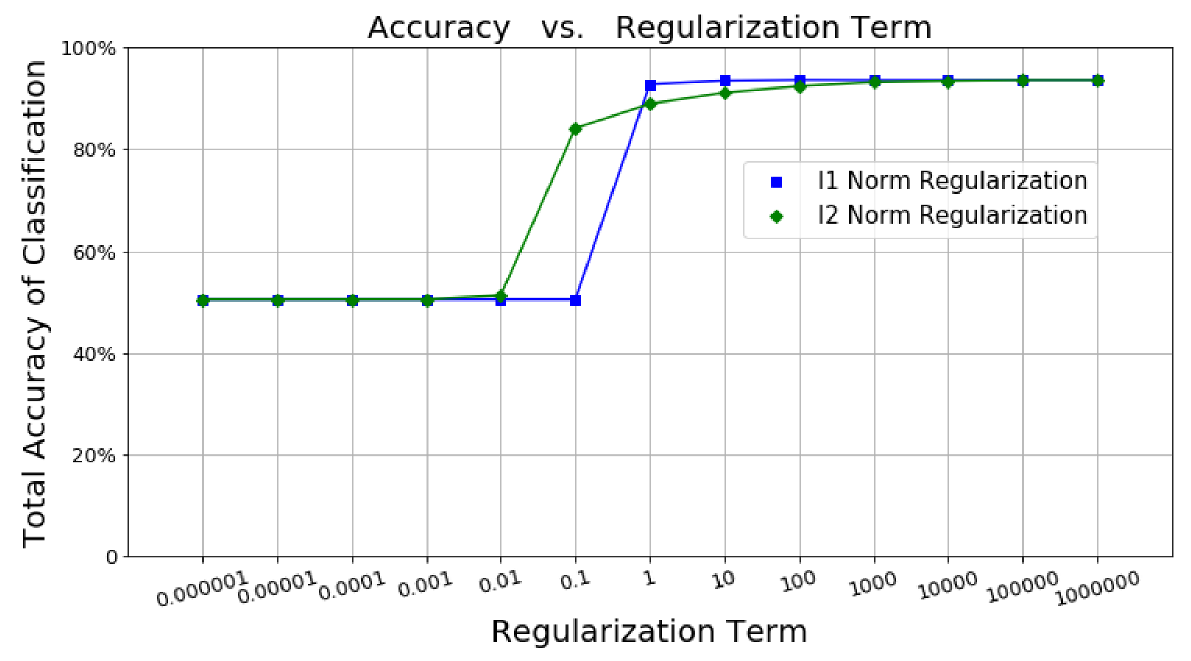
LSI Data with min_df = 2



LSI Data with min_df = 5



NMF Data with min_df = 2



From the curves above we can see that, at small values of penalty parameter the classification barely works when all test cases are classified as the computer technology. But as the penalty parameter increases, the accuracy of the classification grows as well. As we can see from the accuracy-regularization term plot, when the penalty parameter grows larger than 0.001, both L1 and L2 norm curve will increase rapidly and finally achieve a stable accuracy at around 90%.

In addition, it is hard to tell which one of L1 and L2 is better. Generally, L2 regularization will perform better when the penalty parameter is less than 0.01 for LSI data, and 0.1 for NMF data. However, with the growth of penalty parameter, both L1 and L2 achieve a stable accuracy at around 90%. Generally, L2 regularization has computational efficiency due to having analytical solutions but it could not do feature selection, on the other hand, L1 regularization will perform poorly on non-sparse matrix but it has a built-in feature selection functionality.

2.10 Question j

For multiclass classification, we train classifiers on the documents belonging to the classes: “comp.sys.ibm.pc.hardware”, “comp.sys.mac.hardware”, “misc.forsale”, “soc.religion.christian”. Since this is a multi-class classification problem, we perform Naive Bayes classification and multi-class SVM classification on our TF-IDF matrices. The results are as following:

LSI Data with min_df = 2

```
#######################################
#                                     #
# One vs One Classifier - Naive Bayes #
#######################################

                       Classification Report:
==================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.64      0.64      0.64       392
   comp.sys.mac.hardware       0.60      0.54      0.57       385
            misc.forsale       0.61      0.70      0.65       390
  soc.religion.christian       0.98      0.94      0.96       398

             avg / total       0.71      0.71      0.71      1565

==================================================================

Confusion Matrix:
===================
[[252  73  63   4]
 [ 85 206  93   1]
 [ 51  61 274   4]
 [  4   1  18 375]]
===================

Total Accuracy:
0.707348242812
```

LSI Data with min_df = 2

```
###############################
#                             #
# One vs One Classifier - SVM #
###############################

                       Classification Report:
==================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.80      0.86      0.83       392
   comp.sys.mac.hardware       0.87      0.79      0.83       385
            misc.forsale       0.86      0.89      0.87       390
  soc.religion.christian       0.99      0.96      0.98       398

             avg / total       0.88      0.88      0.88      1565

==================================================================

Confusion Matrix:
===================
[[339  28  25   0]
 [ 53 304  27   1]
 [ 26  16 347   1]
 [  8   2   6 382]]
===================

Total Accuracy:
0.876677316294
```

LSI Data with min_df = 2

```
#########################################
#                                       #
# One vs Rest Classifier - Naive Bayes #
#########################################

                         Classification Report:
==================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.64      0.63      0.63       392
   comp.sys.mac.hardware       0.61      0.55      0.58       385
            misc.forsale       0.61      0.69      0.65       390
  soc.religion.christian       0.96      0.94      0.95       398

             avg / total       0.71      0.70      0.70      1565

==================================================================

Confusion Matrix:
===================
[[246  66  69  11]
 [ 84 211  89   1]
 [ 50  66 270   4]
 [  3   1  18 376]]
===================

Total Accuracy:
0.704792332268
```

LSI Data with min_df = 2

```
################################
#                              #
# One vs Rest Classifier - SVM #
################################

                       Classification Report:
==================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.81      0.85      0.83       392
   comp.sys.mac.hardware       0.86      0.79      0.82       385
            misc.forsale       0.86      0.90      0.88       390
  soc.religion.christian       0.99      0.98      0.98       398

             avg / total       0.88      0.88      0.88      1565

==================================================================

Confusion Matrix:
===================
[[333  32  25   2]
 [ 51 303  30   1]
 [ 22  16 351   1]
 [  4   1   4 389]]
===================

Total Accuracy:
0.879233226837
```

NMF Data with min_df = 2

```
########################################
#                                      #
# One vs One Classifier - Naive Bayes #
########################################

                       Classification Report:
===================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.63      0.81      0.71       392
   comp.sys.mac.hardware       0.75      0.65      0.70       385
            misc.forsale       0.78      0.62      0.69       390
  soc.religion.christian       0.94      0.98      0.96       398

             avg / total       0.77      0.77      0.76      1565

===================================================================

Confusion Matrix:
===================
[[317  42  28   5]
 [ 90 252  38   5]
 [ 92  40 241  17]
 [  5   1   2 390]]
===================

Total Accuracy:
0.766773162939
```

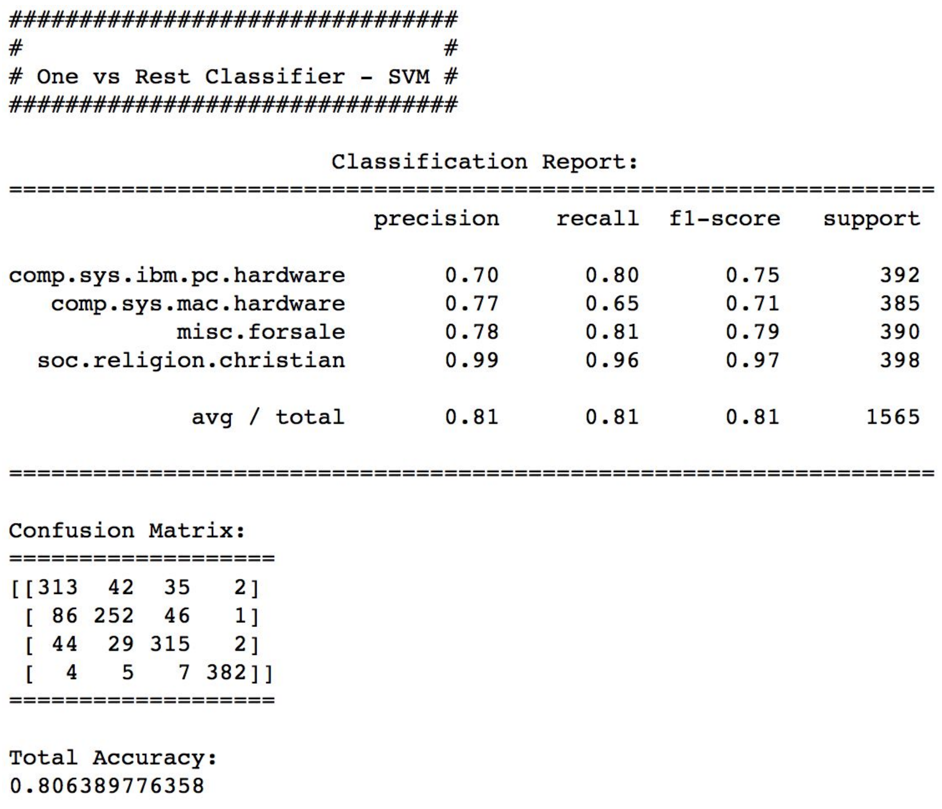
NMF Data with min_df = 2

```
###############################
#                             #
# One vs One Classifier - SVM #
###############################

                       Classification Report:
===================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.66      0.82      0.73       392
   comp.sys.mac.hardware       0.73      0.65      0.69       385
            misc.forsale       0.81      0.77      0.79       390
  soc.religion.christian       1.00      0.91      0.95       398

             avg / total       0.80      0.79      0.79      1565

===================================================================

Confusion Matrix:
===================
[[322  44  26   0]
 [ 96 250  39   0]
 [ 53  36 300   1]
 [ 18  13   4 363]]
===================

Total Accuracy:
0.789137380192
```

NMF Data with min_df = 2

```
########################################
#                                      #
# One vs Rest Classifier - Naive Bayes #
########################################

                       Classification Report:
===================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.65      0.75      0.70       392
   comp.sys.mac.hardware       0.74      0.67      0.70       385
            misc.forsale       0.77      0.71      0.74       390
  soc.religion.christian       0.95      0.98      0.97       398

             avg / total       0.78      0.78      0.78      1565

===================================================================

Confusion Matrix:
===================
[[293  65  30   4]
 [ 76 257  49   3]
 [ 78  23 275  14]
 [  3   0   3 392]]
===================

Total Accuracy:
0.777635782748
```

NMF Data with min_df = 2



```
################################
#                              #
# One vs Rest Classifier - SVM #
################################

                       Classification Report:
==================================================================
                          precision    recall  f1-score   support

comp.sys.ibm.pc.hardware       0.70      0.80      0.75       392
   comp.sys.mac.hardware       0.77      0.65      0.71       385
            misc.forsale       0.78      0.81      0.79       390
  soc.religion.christian       0.99      0.96      0.97       398

             avg / total       0.81      0.81      0.81      1565

==================================================================

Confusion Matrix:
===================
[[313  42  35   2]
 [ 86 252  46   1]
 [ 44  29 315   2]
 [  4   5   7 382]]
===================

Total Accuracy:
0.806389776358
```

From the result above, we can see that SVM classification is always more accurate than Naive Bayes classification regarding multi-class classification. And OneVsOne and OneVsRest makes almost no differences in accuracy, but OneVsRest will perform slightly better.