ECE 219 Project 1 Report

Wenyang Zhu 904947071
Jui Chang 804506544

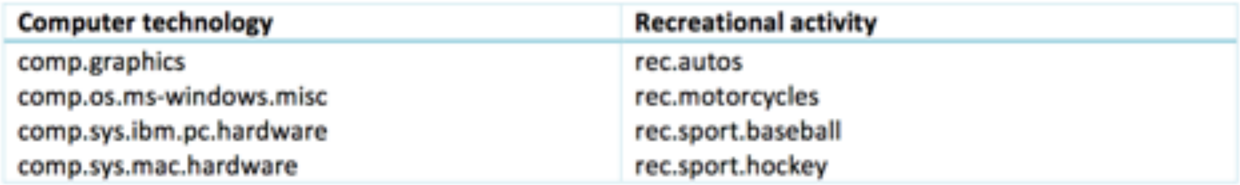
Menu

1. Introduction

Classification aims at identifying the category, to which a test data point belongs, by using the training data sets. In this project, we will use classifiers like Linear Support Vector Machines (SVM), Naive Bayes, Logistic Regression Classifier to analyze the 20 newsgroups dataset and classify each document to its corresponding category.

2. Dataset and Problem Statement

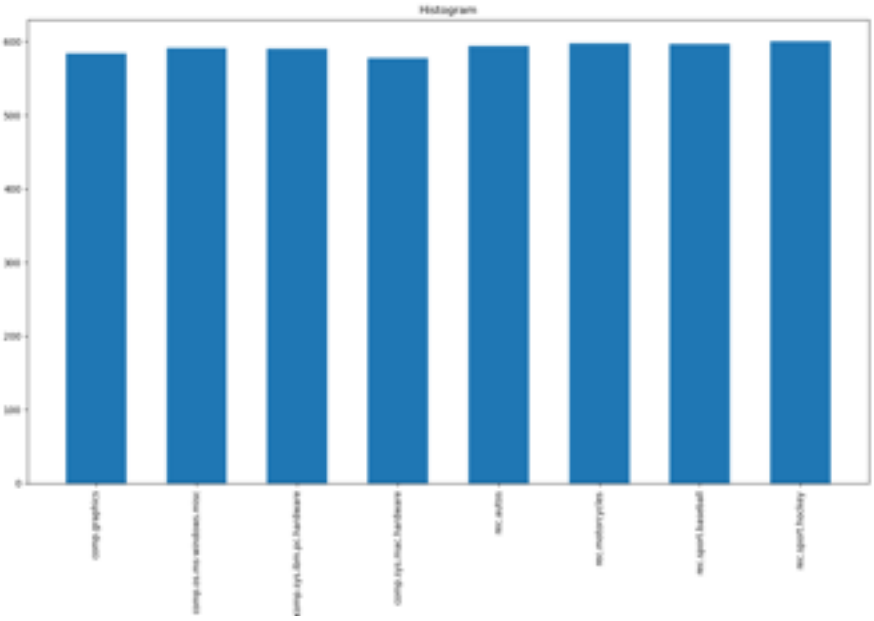
Before processing the data, we should make sure that the numbers of data sets belonging to different classes are balanced, in order to train a better classification. In this project, we will classify the below 8 subclasses (Table1). Therefore, we make a histogram (Graph1) of the number of training documents per class to show if the data set is well balanced.



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

Table1. 8 Subclasses of ‘Computer technology’ and ‘Recreational activity’

a) Histogram plot of training documents



Graph1. the number of training documents per class

In this figure (Graph1) above, we can observe that the number of training set of 8 classes we selected is approximately evenly distributed. Therefore, the data set is already balanced, and we do not need to balance.

3. Modeling Text Data and Feature Extraction

b) TFxIDF vector representation for each document

In this part, we need to turn the documents in our 8 selected training set into numerical feature vectors, and use two settings for minimum document frequency to record the final number of terms we extract.

To solve this problem, We can firstly create a term-document matrix for further processing. In the documents, some interference factors such as punctuations and stop words will interfere our classification, so we first remove punctuations, stop words, and also convert the words which share the same stem as the same word.

Besides, in our later analysis, we find that these documents may also have many numbers, which are unnecessary in analyzing the categories they belong to. Therefore, we also remove all the numbers in the documents to get a better solution of classification. With all these preprocessing of documents, we then use TfidfVectorizer to find the TFxIDF vector representations of documents.

With different minimum document frequencies, the final number of terms are different. Our output shows that in these 8 categories of training set:

With min_df = 2, the size of matrix is (4732, 19689), hence the total number of terms in documents is 19689.

With min_df = 5, the size of matrix is (4732, 8190), hence the total number of terms in documents is 8190.

From the result, it is obvious that if we change our min_df from 2 to 5, the total number of terms we select will drop. The result is intuitive because the more the lower bound of frequency of term is selected, the less number of term we will keep in our matrix.

c) 10 most significant terms in each class with respect to TFxICF

By using the same procedure as constructing TFxIDF, but changing the rows of matrix from documents to classes, we can get the TFxICF vector representations for each class. In this part, we need to find 10 most significant terms with respect to TFxICF in the below four classes: *'comp.sys.ibm.pc.hardware'*, *'comp.sys.mac.hardware'*, *'misc.forsale'*, and *‘soc.religion.christian’*.

Therefore, we firstly use CountVectorizer to count the term frequency in each document of the 4 classes. Then, within the same class, we combine all the documents together to and add up the term frequency in this class. Finally, we use TfidfTransformer to get the TFxICF vector representation of each class. In this way, the most frequent 10 terms can be found for each class. The below table (Table2) shows the frequent terms as well as its frequency in each class.

comp.sys.ibm.pc.hardware	comp.sys.mac. hardware	misc.forsale	soc.religion. christian
drive 0.299	thi 0.247	line 0.298	thi 0.355
scsi 0.281	line 0.235	subject 0.285	god 0.292
thi 0.246	mac 0.227	sale 0.277	wa 0.252
ide 0.222	subject 0.214	organ 0.274	christian 0.223
use 0.201	organ 0.201	univers 0.158	jesu 0.168
line 0.194	use 0.172	thi 0.158	hi 0.154
subject 0.187	quadra 0.172	new 0.156	church 0.137
organ 0.179	simm 0.164	use 0.135	subject 0.136
card 0.146	appl 0.152	offer 0.132	peopl 0.130
control 0.123	problem 0.134	nntppostinghost 0.125	line 0.126

Table2. 10 Most Significant Terms for 4 Classes

4. Feature Selection

d) LSI and NMF representation of TFxIDF (min_df = 2 and 5)

Latent semantic indexing (LSI) is a dimensional reduction method which can reduce dimension with squared residual between the original data and reconstruction data minimized. LSI is similar to principal component analysis.

To do dimension reduction of our TFxIDF matrix, we firstly apply LSI and pick k=50, so each document is mapped to a 50 dimensional vector. Later, we will use the selected feature to perform training in the following steps.

Similarly, we also do dimension reduction using Non-Negative Matrix Factorization (NMF), and we will compare the results from LSI and NMF in the later sections.

Furthermore, according to the instructions, for LSI, we will use the results from part b with min_df = 2 and min_df = 5 respectively; however, for NMF, we will just use the result with min_df = 2.

5. Learning Algorithms

e) Hard and soft margin SVM classifier (SVC)

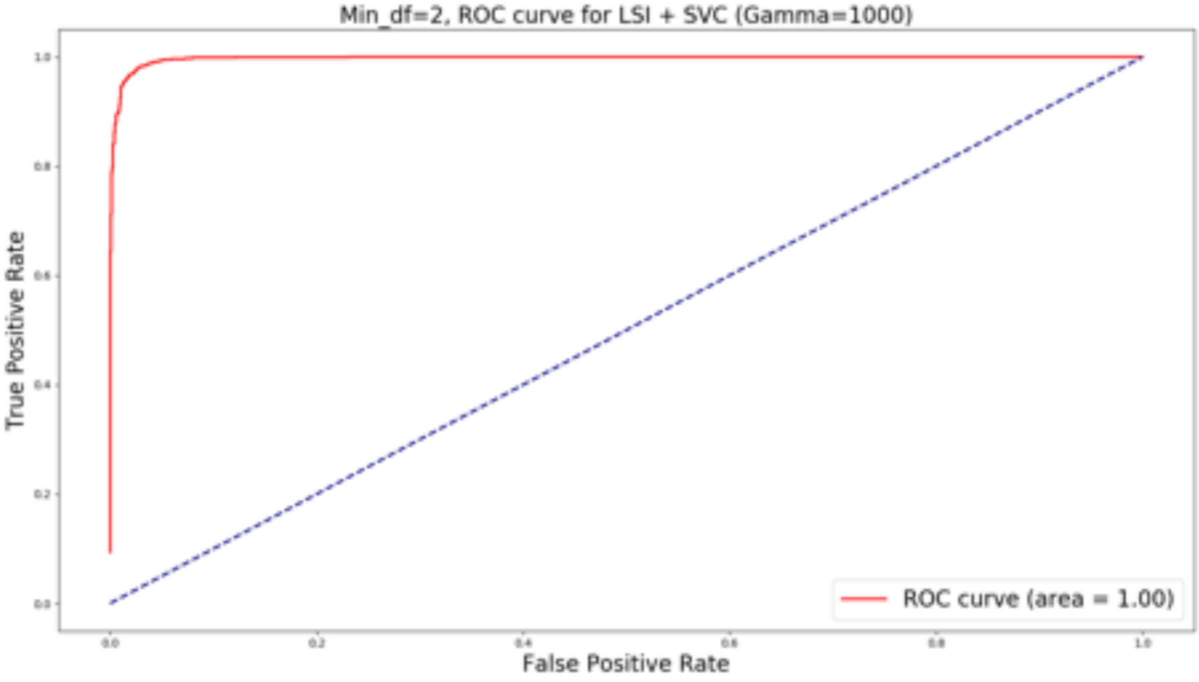
We set γ = 1000 to be the hard margin SVC, and γ = 0.001 to be the soft margin SVC respectively, to separate the documents into two groups: Computer Technology vs Recreational Activity.

(1) With LSI and min_df = 2

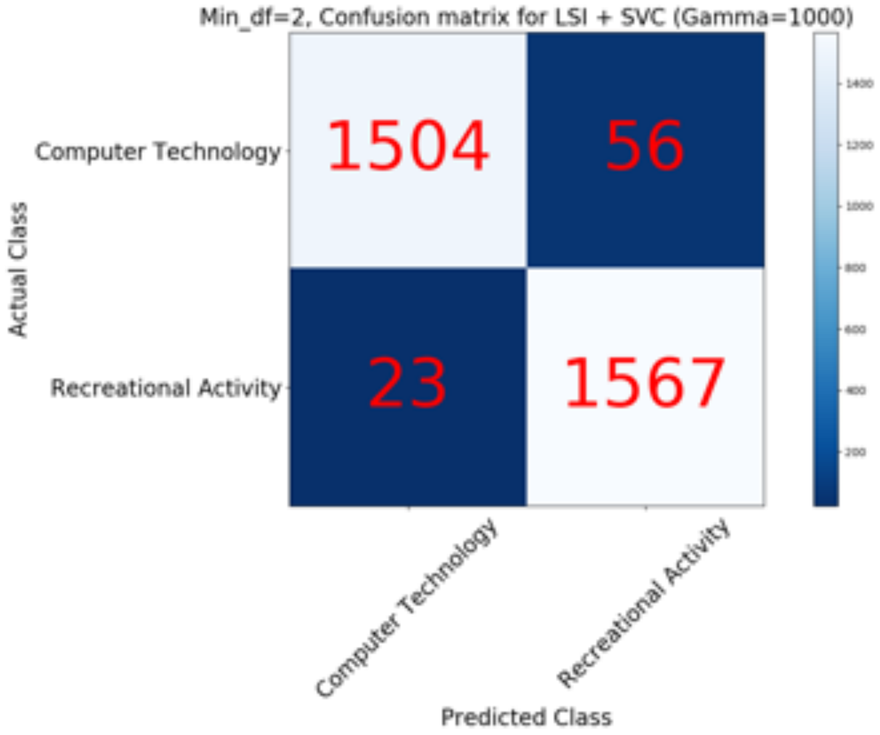
In this part, we use SVM classifier (SVC) to classify the LSI representation of TFxIDF vector with min_df = 2.

For hard margin SVC (γ = 1000), the Receiver Operating Characteristic (ROC) curve (Graph2), the heat map of confusion matrix (Graph3), and accuracy, recall and precision are shown below:

Accuracy = 0.975555555556 Recall = 0.98427672956 Precision = 0.967841682127

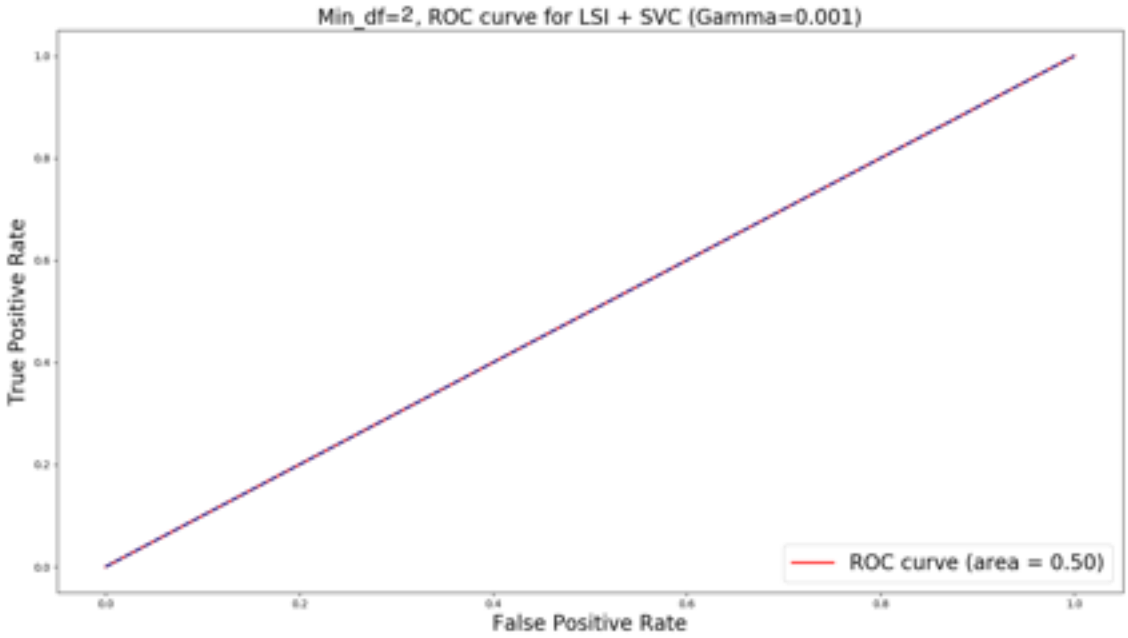


Graph2. min_df=2, ROC curve for LSI + SVC (Gamma = 1000)

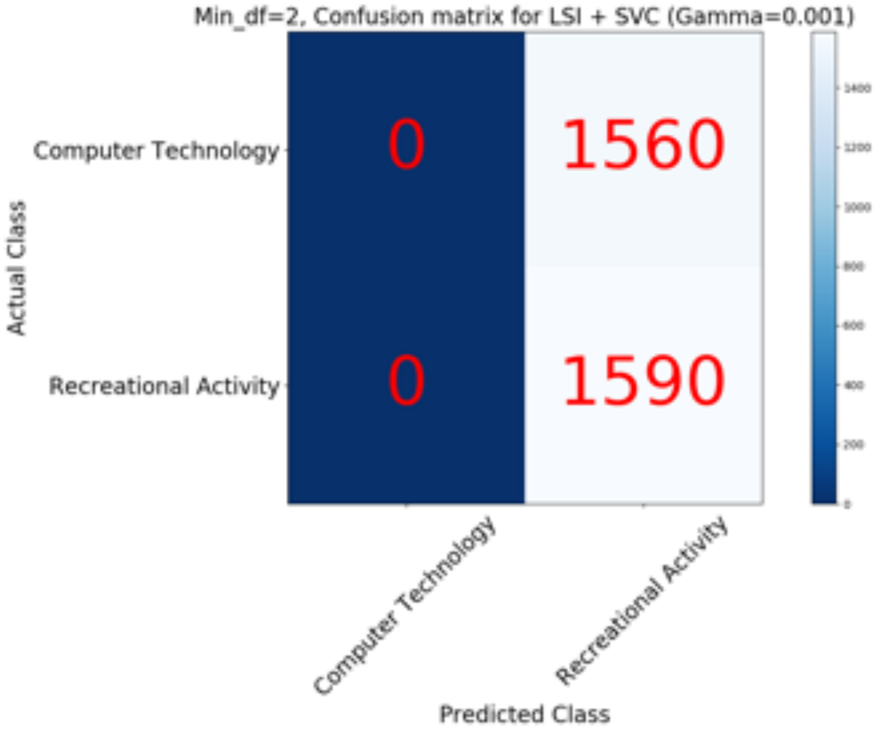


Graph3. min_df = 2, confusion matrix for LSI + SVC (Gamma = 1000)

For soft margin SVC (γ = 0.001), the Receiver Operating Characteristic (ROC) curve (Graph4), the heat map of confusion matrix (Graph5), and accuracy, recall and precision are shown below:
Accuracy = 0.504761904762 Recall = 1.0 Precision = 0.504761904762
Both the ROC curve and the confusion matrix look bad since the classification does not work well due to very small γ (= 0.001). In fact, the accuracy rate is only 0.505. So the binary classification is approximately random.



Graph4. min_df = 2, ROC curve for LSI + SVC (Gamma = 0.001)



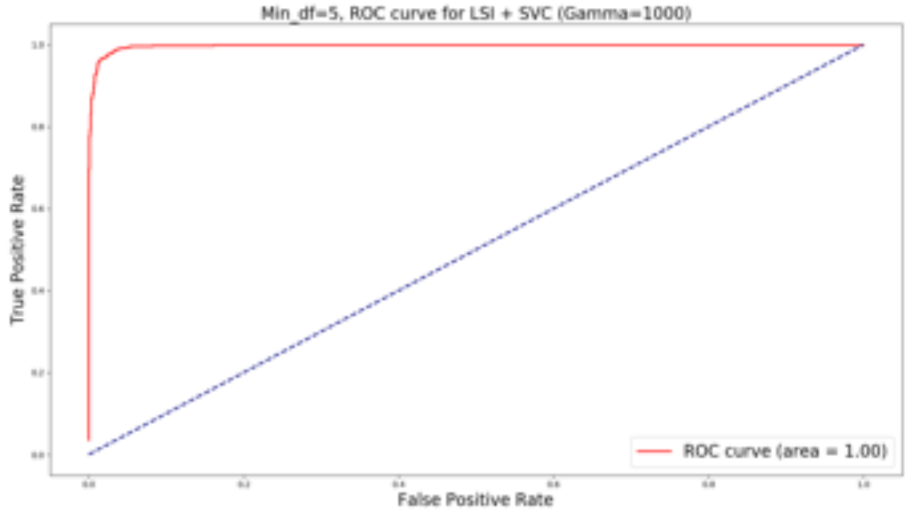
Graph5. min_df = 2, Confusion Matrix for LSI+SVC (Gamma = 0.001)

<u>(2) With LSI and min_df = 5</u>

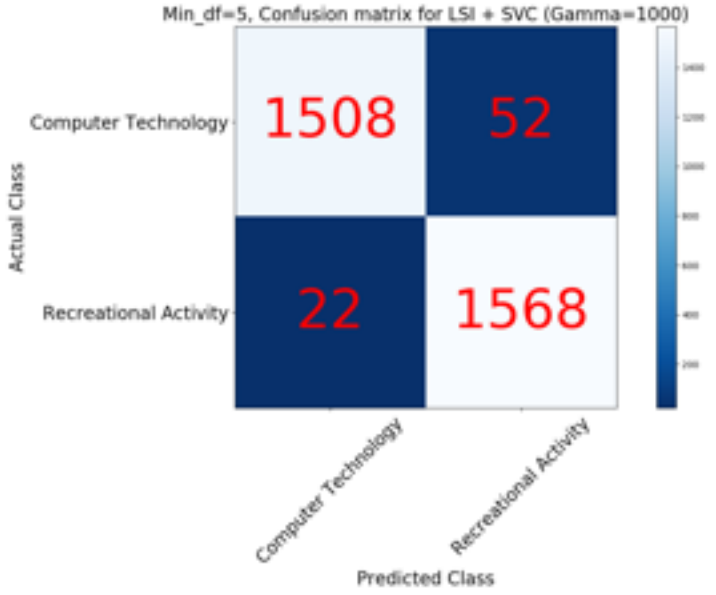
In this part, we use SVM classifier (SVC) to classify the LSI representation of TFxIDF vector with min_df = 2.

For hard margin SVC (γ = 1000), the Receiver Operating Characteristic (ROC) curve (Graph6), the heat map of confusion matrix (Graph7), and accuracy, recall and precision are shown below:

Accuracy = 0.976825396825 Recall = 0.985534591195 Precision = 0.969078540507

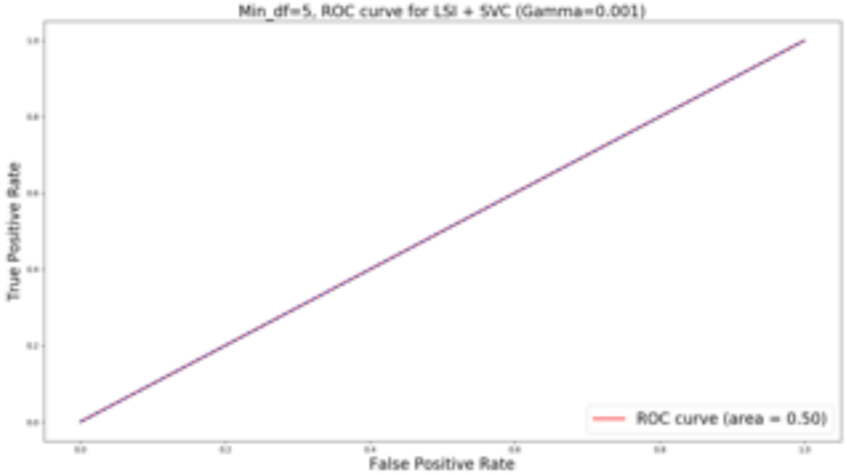


Graph6. min_df = 5, ROC curve for LSI + SVC (Gamma = 1000)

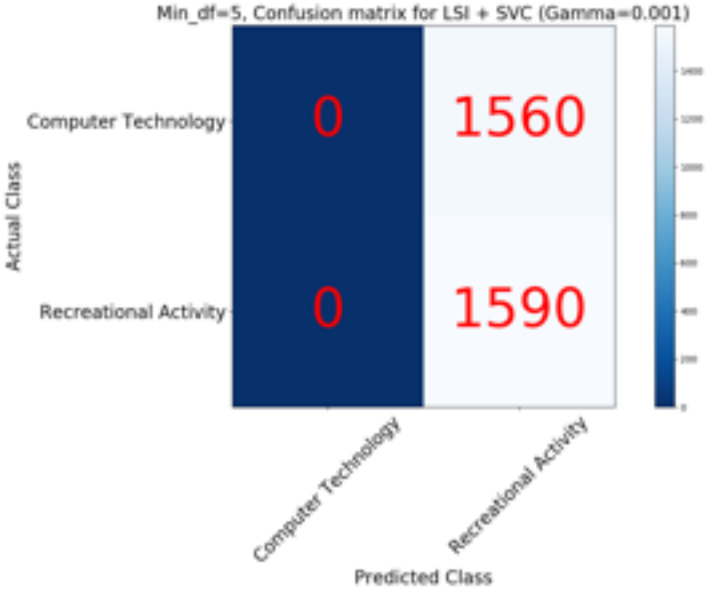


Graph7. min_df = 5, confusion matrix for LSI + SVC (Gamma = 1000)

For soft margin SVC (γ = 0.001), the Receiver Operating Characteristic (ROC) curve (Graph8), the heat map of confusion matrix (Graph9), and accuracy, recall and precision are shown below.
Accuracy = 0.504761904762 Recall = 1.0 Precision = 0.504761904762
For the same reason as Graph4 and Graph5, both the ROC curve and the confusion matrix look bad since the classification does not work well due to very small γ (= 0.001). In fact, the accuracy rate is only 0.505. So the binary classification is approximately random.



Graph8. min_df = 5, ROC curve for LSI + SVC (Gamma = 0.001)



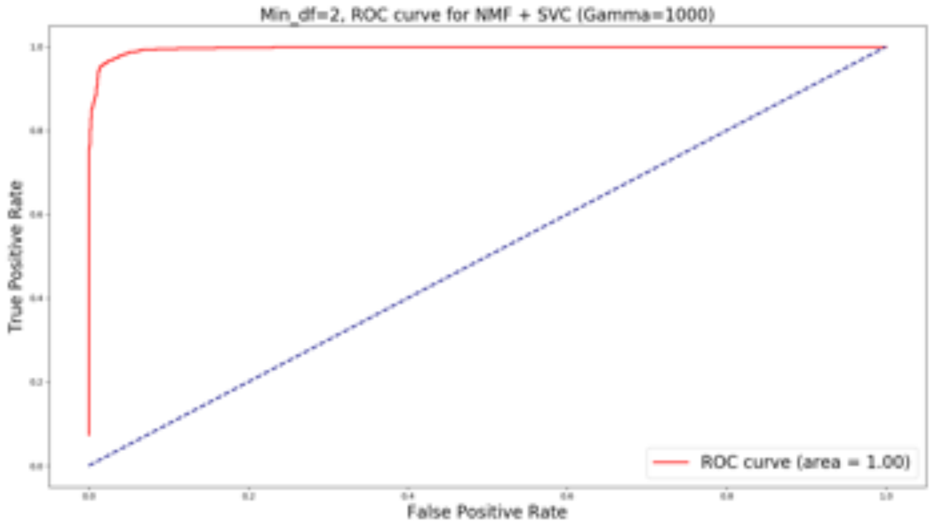
Graph9. min_df = 5, confusion matrix for LSI + SVC (Gamma = 0.001)

(3) Compare the results in (1) and (2)

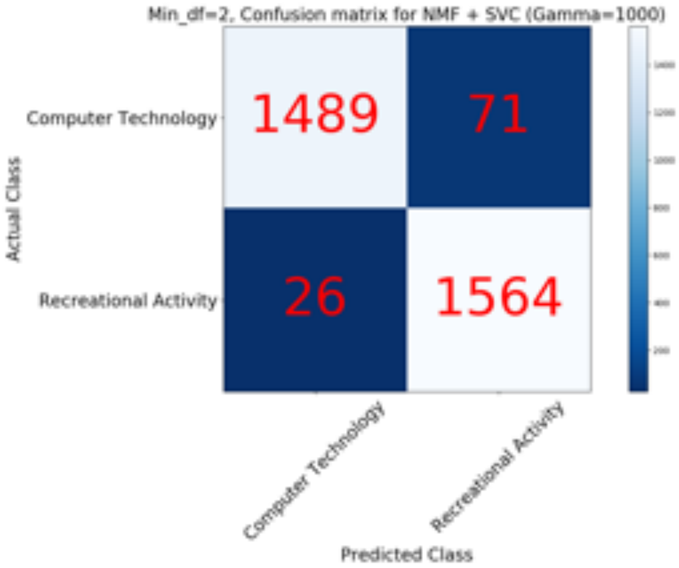
Classification with same gamma has higher accuracy for min_df = 5 rather than min_df = 2.

(4) With NMF and min_df = 2

For Gamma = 1000, Accuracy = 0.969206349 Recall = 0.983647798 Precision = 0.956574923

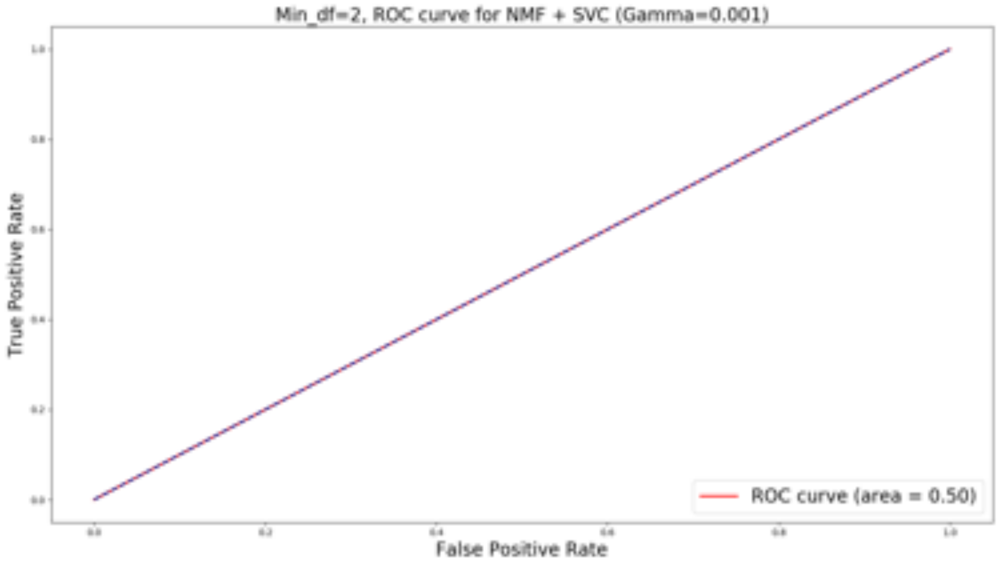


Graph10. min_df = 2, ROC curve for NMF + SVC (Gamma = 1000)

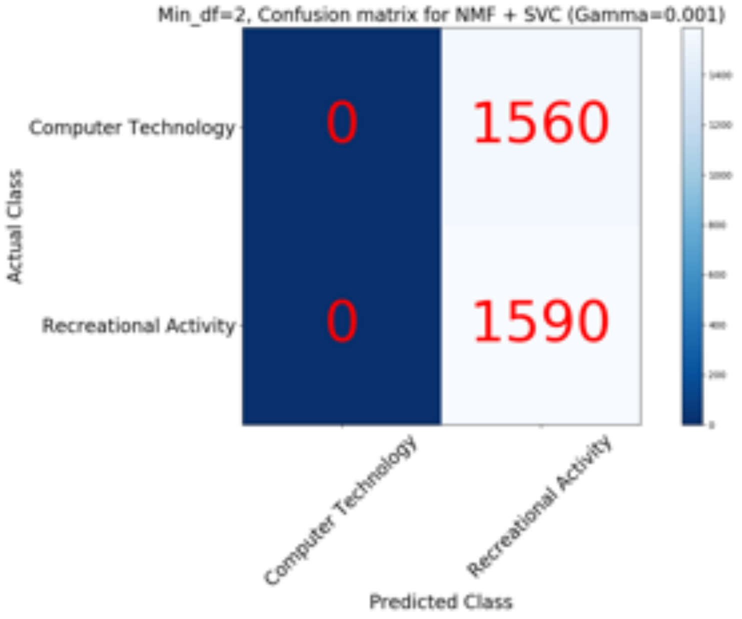


Graph11. min_df = 2, confusion matrix for NMF + SVC (Gamma = 1000)

For Gamma = 0.001, Accuracy = 0.504761904 Recall = 1.0 Precision = 0.504761904.
For the same reason as Graph4 and Graph5, both the ROC curve and the confusion matrix look bad since the classification does not work well due to very small γ (= 0.001). In fact, the accuracy rate is only 0.505. So the binary classification is approximately random.



Graph12. min_df = 2, ROC curve for NMF + SVC (Gamma = 0.001)



Graph13. min_df = 2, confusion matrix for NMF + SVC (Gamma = 0.001)

(5) Compare the results in (1) and (4)

NMF works better than LSI for 20news_group dataset with SVC classification.

f) Choice of best value for parameter γ

In this part, we repeat problem (e), but use parameter γ in the range $\{10^{-k} \mid -3 \leq k \leq 3, k \in Z\}$ to find the highest accuracy rate with respect to γ. And the parameter with highest accuracy is the best value that we want in the problem.

In order to solve the problem, we list the accuracy, recall and precision results for each γ value.

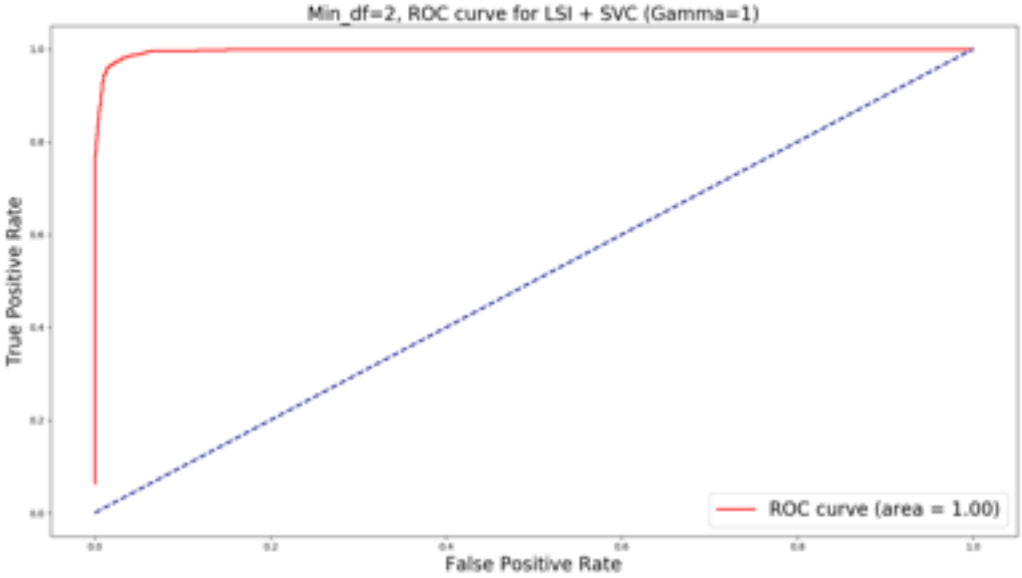
(1) With LSI and min_df = 2

The results is shown in Table3.

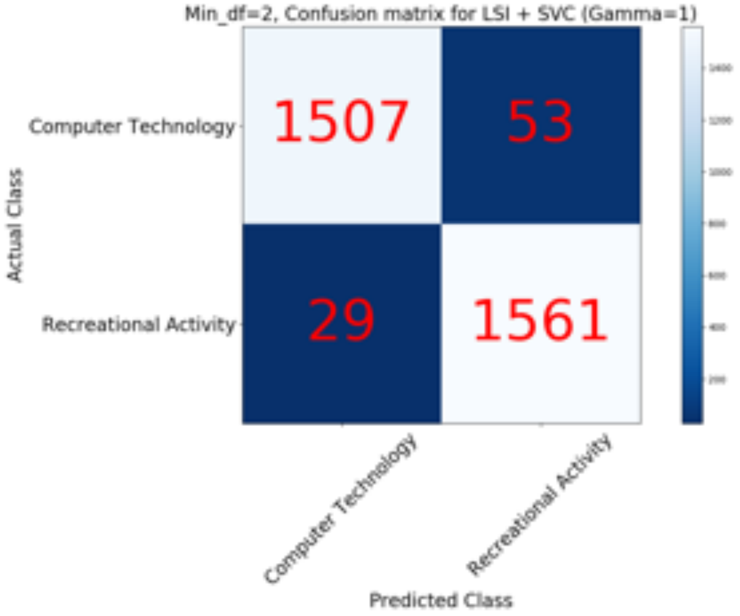
Gamma	Accuracy	Recall	Precision
0.001	0.505	1.0	0.505
0.01	0.507	1.0	0.506
0.1	0.968	0.973	0.965
1	0.976	0.974	0.978
10	0.934	0.934	0.935
100	0.748	0.756	0.748
1000	0.557	0.558	0.562

Table3. min_df = 2, LSI

Therefore, the best value for parameter γ is 1, and the corresponding ROC and confusion matrix are shown in Graph14 and Graph15. The corresponding accuracy, recall and precision are:
Accuracy = 0.976, Recall = 0.974, Precision = 0.978



Graph14. min_df = 2, ROC curve for LSI + SVC (Gamma = 1)



Graph15. min_df = 2, Confusion matrix for LSI + SVC (Gamma = 1)

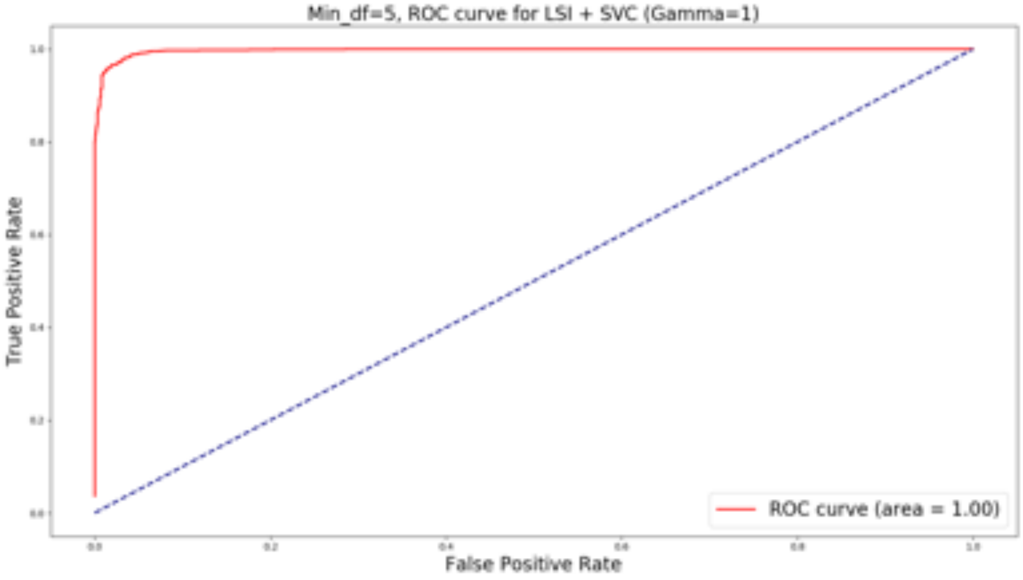
(2) With LSI and min_df = 5

The results is shown in Table4.

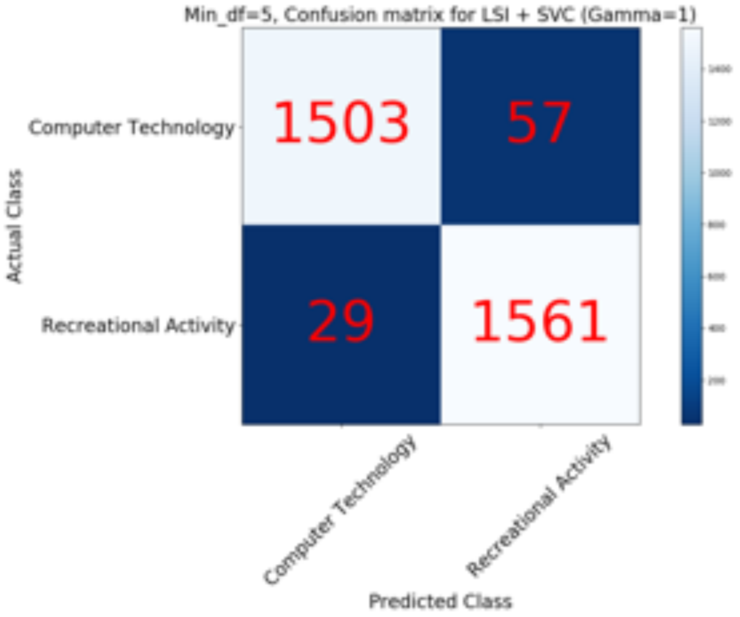
Gamma	Accuracy	Recall	Precision
0.001	0.505	1.0	0.505
0.01	0.515	1.0	0.510
0.1	0.968	0.975	0.962
1	0.974	0.975	0.973
10	0.929	0.930	0.930
100	0.715	0.717	0.719
1000	0.545	0.555	0.549

Table4. min_df = 5, LSI

Therefore, the best value for parameter γ is 1, and the corresponding ROC and confusion matrix are shown in Graph16 and Graph17. The corresponding accuracy, recall and precision are: Accuracy = 0.974, Recall = 0.975, Precision = 0.973



Graph16. min_df = 5, ROC curve for LSI + SVC (Gamma = 1)



Graph17. min_df = 5, Confusion matrix for LSI + SVC (Gamma = 1)

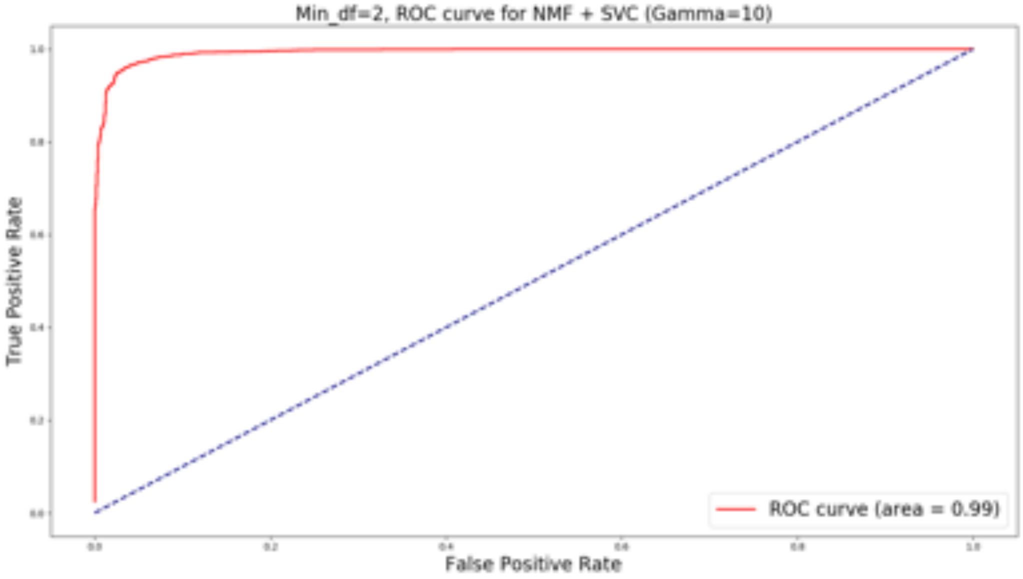
(3) With NMF and min_df = 2

The results is shown in Table5.

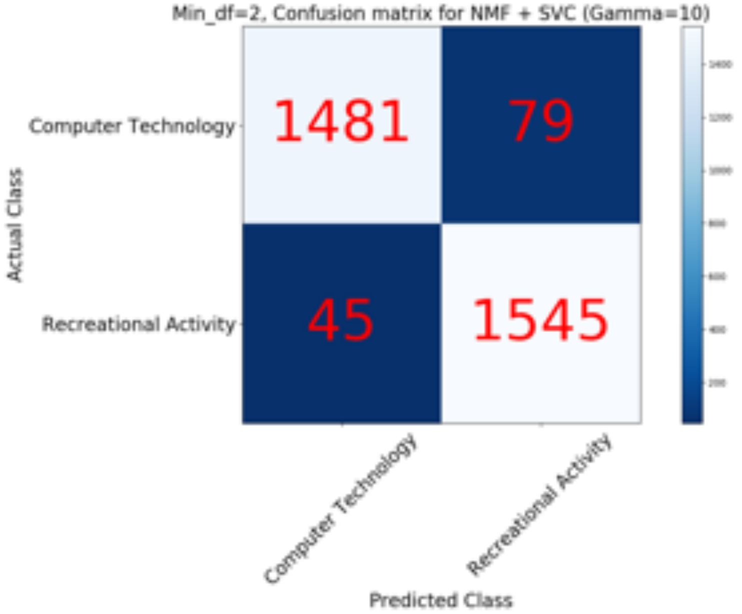
Gamma	Accuracy	Recall	Precision
0.001	0.505	1.0	0.505
0.01	0.505	1.0	0.505
0.1	0.505	1.0	0.505
1	0.945	0.933	0.958
10	0.964	0.956	0.973
100	0.903	0.911	0.898
1000	0.763	0.765	0.766

Table5. min_df = 2, NMF

Therefore, the best value for parameter γ is 10, and the corresponding ROC and confusion matrix are shown in Graph18 and Graph19. The corresponding accuracy, recall and precision are: Accuracy = 0.964, Recall = 0.956, Precision = 0.973



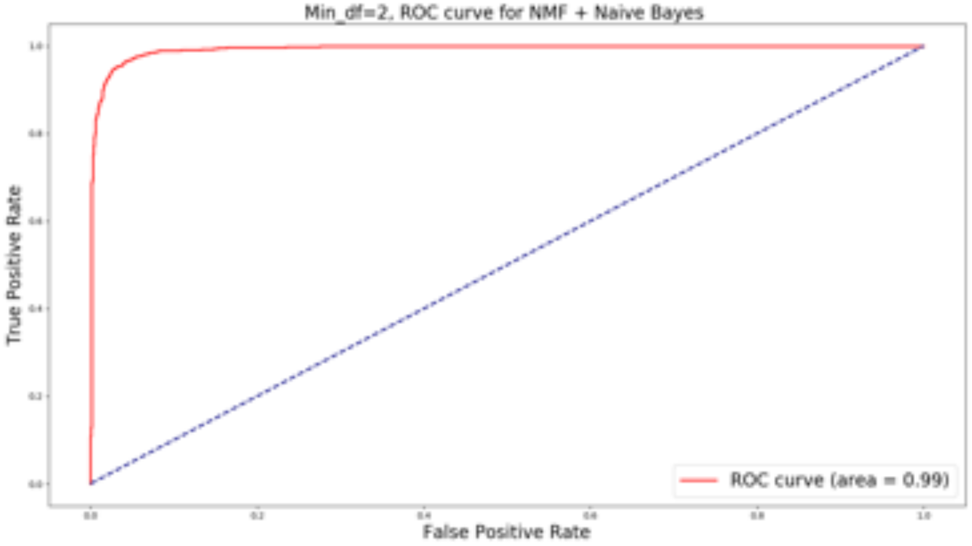
Graph18. min_df = 2, ROC curve for NMF + SVC (Gamma = 10)



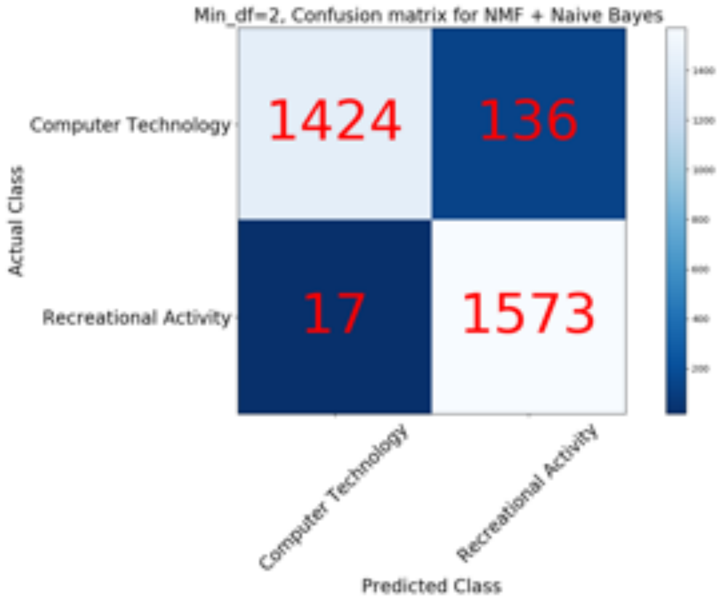
Graph19. min_df = 2, Confusion matrix for NMF + SVC (Gamma = 10)

g) Naive Bayes classifier

In this part, we use Naive Bayes classifier to classify the data sets. Only NMF Representation is needed in this section. Therefore, the Receiver Operating Characteristic (ROC) curve (Graph17), the heat map of confusion matrix (Graph18), and accuracy, recall and precision are:
Accuracy = 0.951428571429 Recall = 0.989308176101 Precision = 0.920421299005



Graph17. min_df = 2, ROC curve for NMF + Naive Bayes



Graph18. min_df = 2, Confusion Matrix for NMF + Naive Bayes

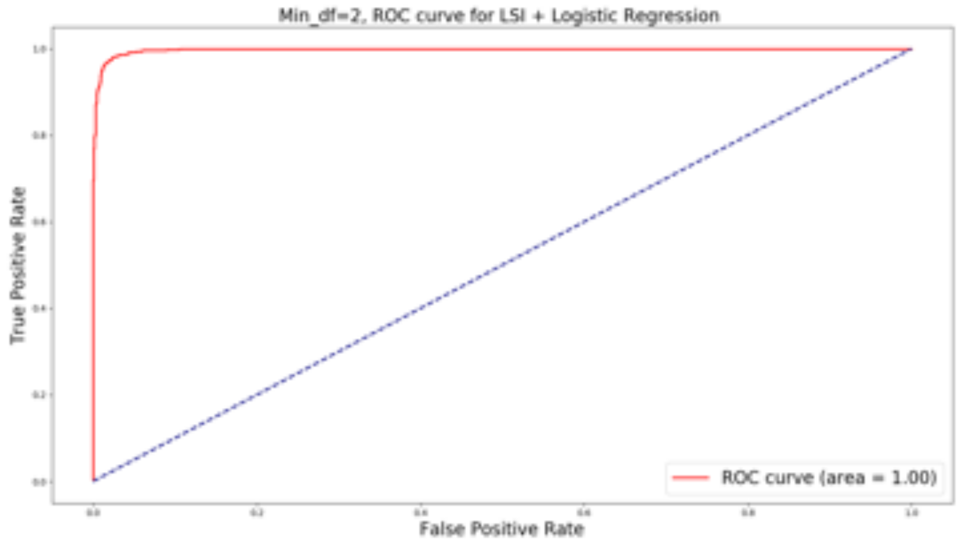
h) Logistic regression classifier

In this part, we again classify the documents using Logistic Regression classifier.

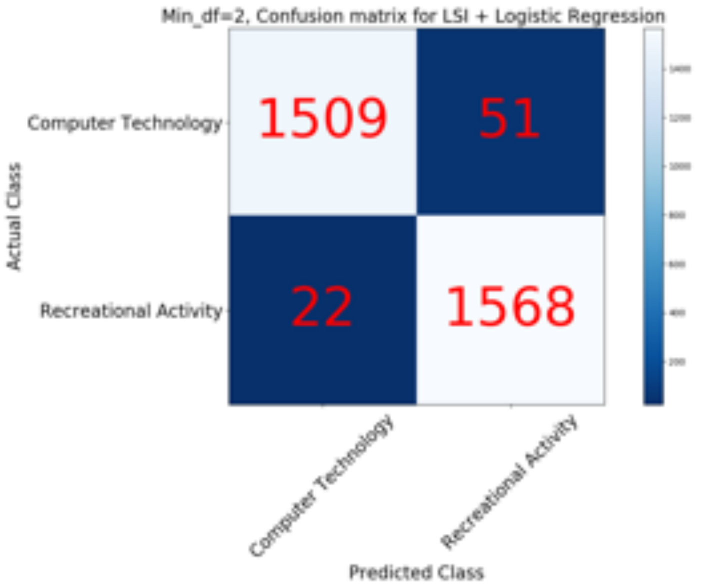
(1) With LSI and min_df = 2

The Receiver Operating Characteristic (ROC) curve (Graph19), the heat map of confusion matrix (Graph20), and accuracy, recall and precision are shown below.

Accuracy = 0.978095238095 Recall = 0.985534591195 Precision = 0.971481711097



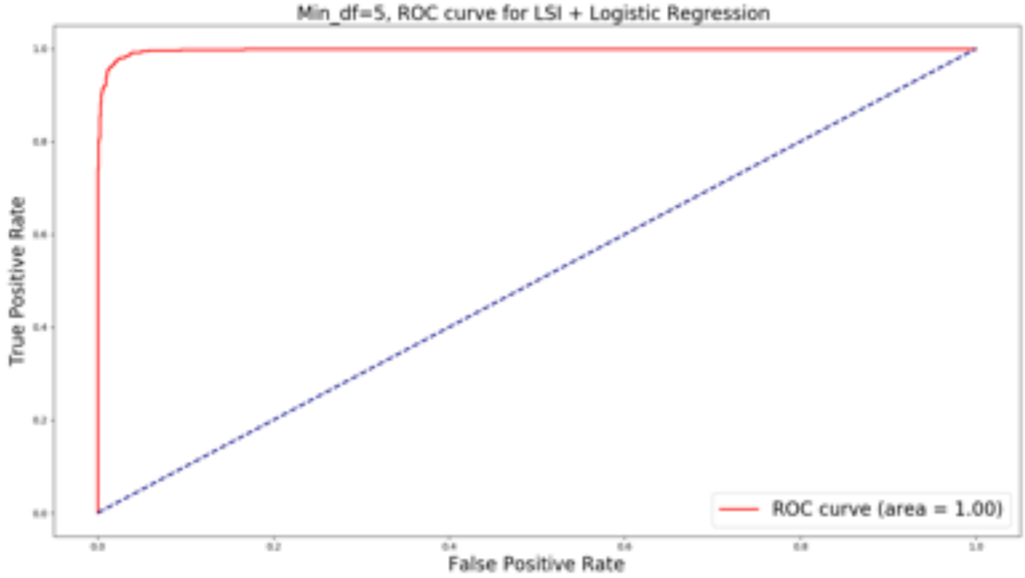
Graph19. min_df = 2, ROC curve for LSI + Logistic Regression



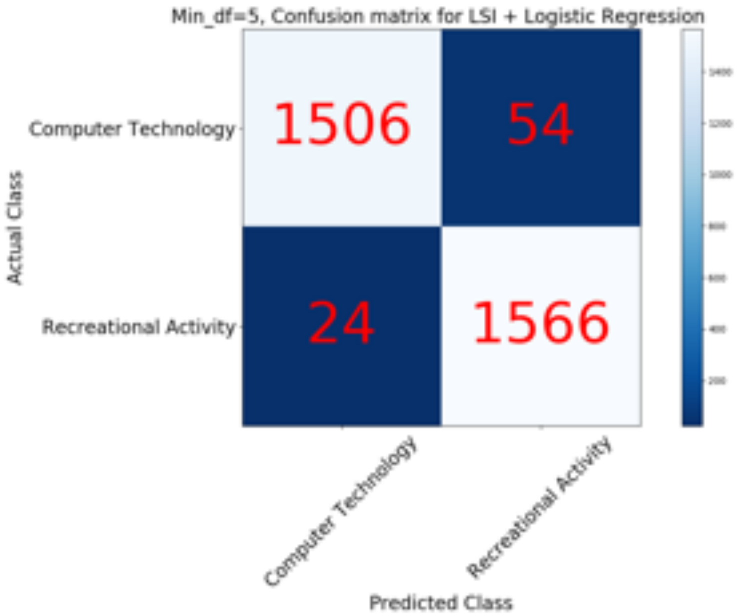
Graph20. min_df = 2, Confusion matrix for LSI + Logistic Regression

(2) With LSI and min_df = 5

Accuracy = 0.975555555556 Recall = 0.984905660377 Precision = 0.967263743051



Graph21. min_df = 5, ROC curve for LSI + Logistic Regression



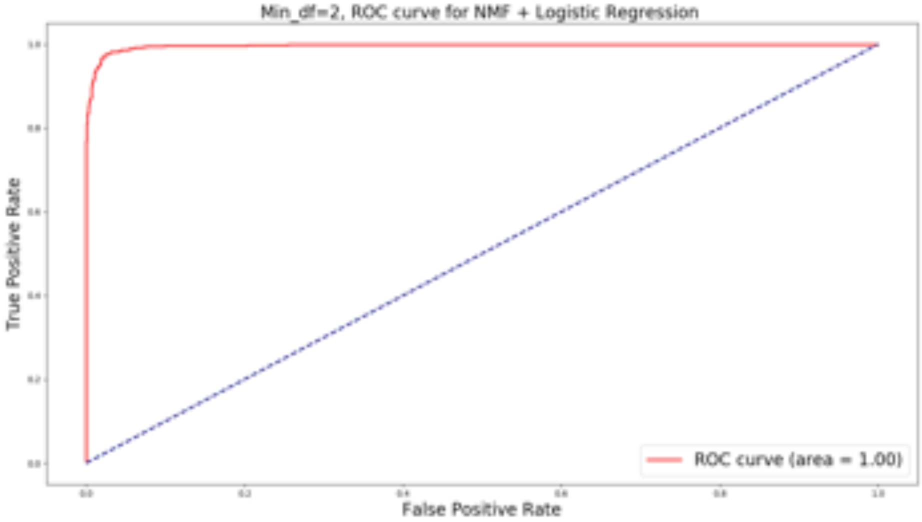
Graph22. min_df = 5, Confusion matrix for LSI + Logistic Regression

(3) Compare the results in (1) and (2)

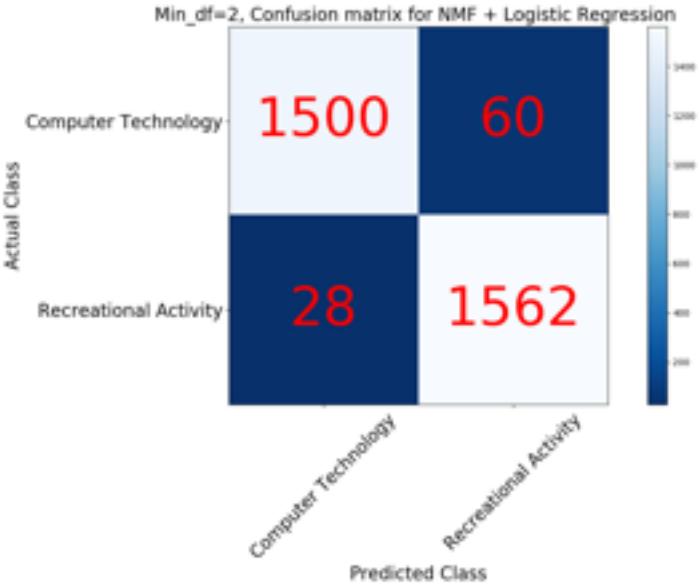
The accuracy with min_df =2 is slightly higher than the one with min_df = 5 using Logistic Regression with LSI representation.

(4) With NMF and min_df = 2

Accuracy = 0.972063492063 Recall = 0.982389937107 Precision = 0.963008631319



Graph23. min_df = 2, ROC curve for NMF + Logistic Regression



Graph24. min_df = 2, Confusion matrix for NMF + Logistic Regression

(5) Compare the results in (1) and (4)

The accuracy of the dataset with LSI representation is higher than the one of the dataset with NMF representation using Logistic Regression.

i) l1 and l2 norm regularizations

Suppose the cost function Logistic Regression is f. We want to find the relationship between the regularization parameter and the performance of Logistic Regression. We need to test the performance of Logistic Regression with L1 and L2 regularization for both datasets featured by NMF and LSI.

(1) With LSI and min_df = 2

i) Logistic Regression with L1 regularization:

The formula is C * f + $||w||_1$, where f is the original cost function, C is the inverse of regularization strength and w is the parameter. We sweep C through the range $\{10^{-k} \mid -3 \leq k \leq 3, k \in Z\}$, we get different accuracy, test error and mean distance of data to the hyperplane (Table6).

C	0.001	0.01	0.1	1	10	100	1000
Accuracy	0.495	0.913	0.941	0.971	0.977	0.977	0.977
Test error	0.505	0.087	0.059	0.029	0.023	0.023	0.023
Mean distance	0.000	0.410	2.968	5.671	9.015	10.721	10.995

Table6. min_df = 2, LSI with l1 norm regularization

ii) Logistic Regression with L2 regularization:

The formula for L2 regularization is C * f + $w^T w/2$. Similarly, sweep C through different values, we get corresponding accuracy, test error and mean distance of data to the hyperplane (Table9).

C	0.001	0.01	0.1	1	10	100	1000
Accuracy	0.758	0.950	0.962	0.969	0.973	0.976	0.976
Test error	0.242	0.050	0.038	0.031	0.027	0.024	0.024
Mean distance	0.014	0.105	0.654	1.985	4.107	7.169	10.129

Table7. min_df = 2, LSI with l2 norm regularization

(2) With LSI and min_df = 5

The procedure is the same as (1).

i) Logistic Regression with L1 regularization:

C	0.001	0.01	0.1	1	10	100	1000
Accuracy	0.495	0.920	0.938	0.970	0.976	0.974	0.974
Test error	0.505	0.080	0.062	0.030	0.024	0.026	0.026
Mean distance	0.000	0.502	3.030	5.720	8.651	10.174	10.431

Table8.min_df = 5, LSI with l1 norm regularization

ii) Logistic Regression with L2 regularization:

C	0.001	0.01	0.1	1	10	100	1000
Accuracy	0.800	0.950	0.961	0.967	0.974	0.975	0.976
Test error	0.200	0.050	0.039	0.033	0.026	0.025	0.024
Mean distance	0.015	0.120	0.712	2.064	4.167	7.067	9.598

Table9. min_df = 5, LSI with l2 norm regularization

(4) With NMF and min_df = 2

The procedure is the same as (1).

i) Logistic Regression with L1 regularization:

C	0.001	0.01	0.1	1	10	100	1000
Accuracy	0.495	0.495	0.710	0.956	0.969	0.970	0.972
Test error	0.505	0.505	0.290	0.044	0.031	0.030	0.028
Mean distance	0.000	0.000	0.174	2.206	5.647	9.977	11.813

Table10. min_df = 2, NMF with l1 norm regularization

ii) Logistic Regression with L2 regularization:

C	0.001	0.01	0.1	1	10	100	1000
Accuracy	0.505	0.531	0.903	0.932	0.952	0.964	0.967
Test error	0.495	0.469	0.097	0.068	0.048	0.036	0.033
Mean distance	0.010	0.018	0.075	0.490	1.575	3.348	6.030

Table11. min_df = 2, LSI with l2 norm regularization

(6) Other questions in part i)

i) How does the regularization parameter affect the test error?

Too low C lets the model not fits the dataset very well, but too high will cause overfitting. Both will get high test error. Hence, the trade-off is important and we must choose a proper C.

ii) How are the coefficients of the fitted hyperplane affected?

Higher C lets the mean distance of data to the hyperplane becomes larger, which means the model fits the dataset well. However, we still need to consider the overfitting due to very high C, which causes bad performance.

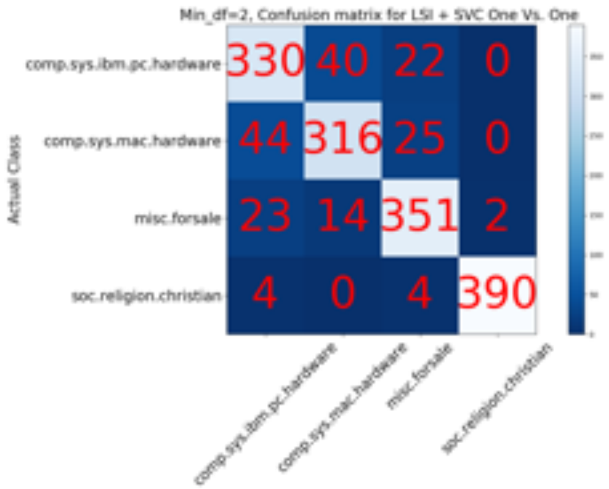
iii) Why might one be interested in each type of regularization?

L1 regularization may obtain robust but unstable solution, while L2 regularization may obtain not very robust but stable solution.

6. Multiclass Classification

We apply both Naive Bayes and SVM on the dataset in the following four categories: comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, misc.forsale, soc.religion.christian.
We use the dataset with min_df = 2 and featured by NMF. For each problem, we need to record both the confusion matrix and the results of accuracy, recall and precision.

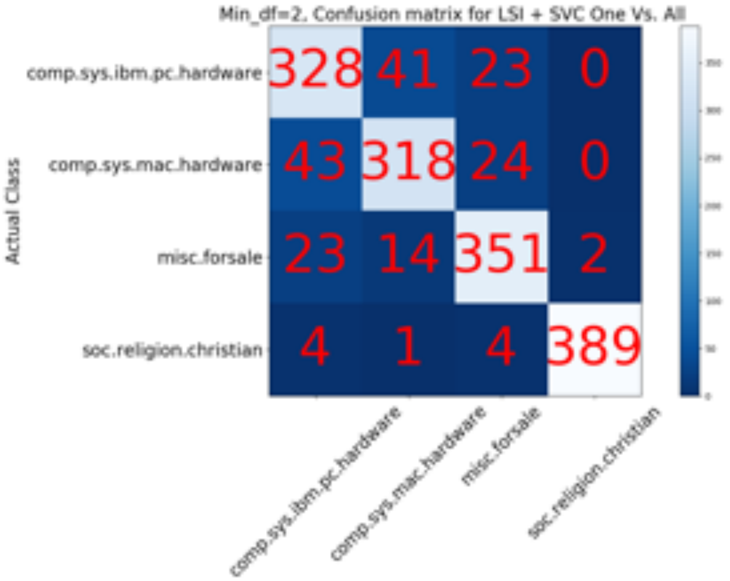
(1) Perform SVC (one vs one) with LSI (min_df = 2)
Accuracy = 0.888817891374 Recall = 0.888817891374 Precision = 0.889781240099



Graph25. min_df = 2, confusion matrix for LSI + SVC One VS One

(2) Perform SVC (one vs the rest) with LSI (min_df = 2)

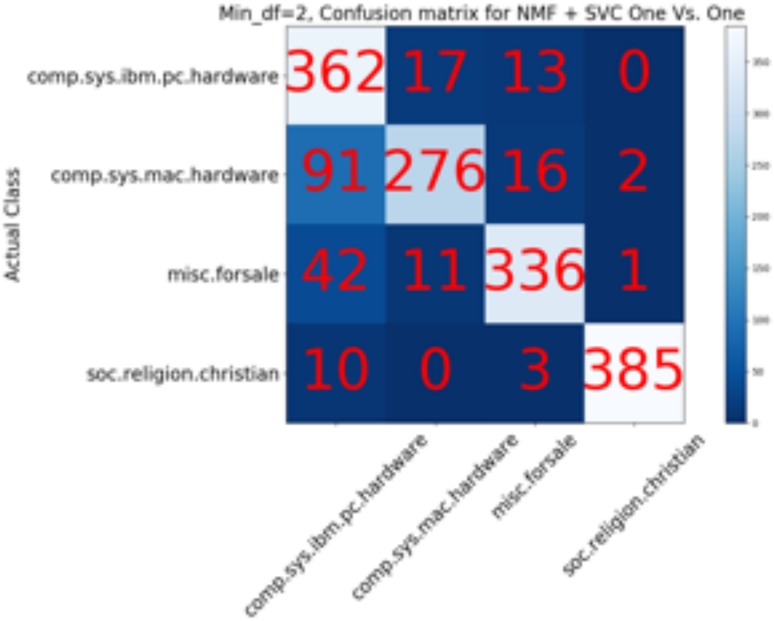
Accuracy = 0.888817891374 Recall = 0.888817891374 Precision = 0.890005017791



Graph26. min_df = 2, confusion matrix for LSI + SVC One VS the Rest

(3) Perform SVC (one vs one) with NMF (min_df = 2)

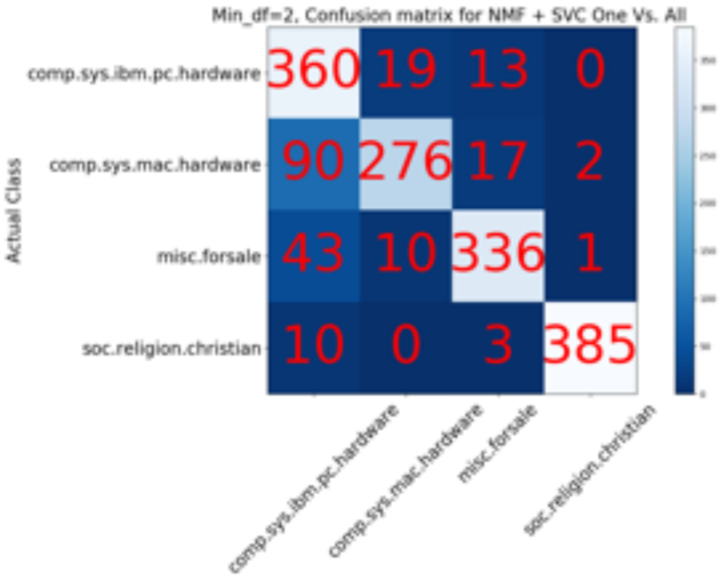
Accuracy = 0.868370607029 Recall = 0.868370607029 Precision = 0.882777719118



Graph27. min_df = 2, confusion matrix for NMF + SVC One VS One

(4) Perform SVC (one vs the rest) with NMF (min_df = 2)

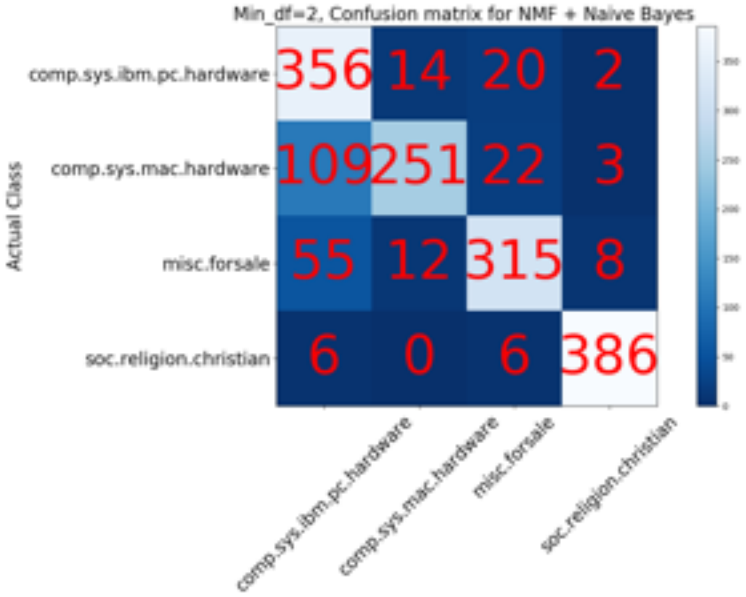
Accuracy = 0.867092651757 Recall = 0.867092651757 Precision = 0.881146795076



Graph28. min_df = 2, confusion matrix for NMF + SVC One VS the Rest

(5) Perform Naive Bayes with NMF (min_df = 2)

Accuracy = 0.835782747604 Recall = 0.834413217794 Precision = 0.854532602293



Graph29. min_df = 2, confusion matrix for NMF + Naive Bayes