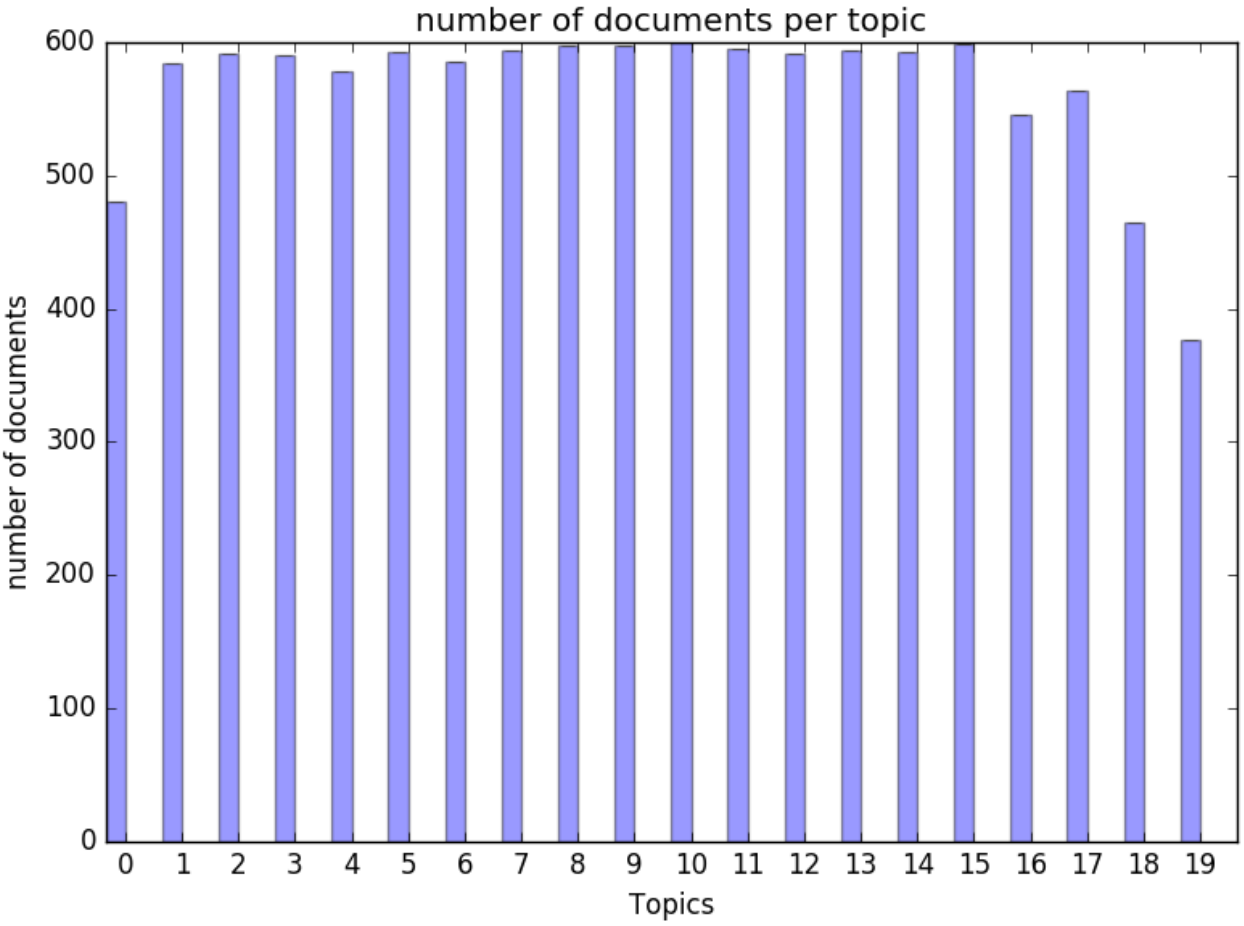
Project 2 Classification Analysis

Dataset and Problem Statement:

The objective is to train a classifier to group the documents into two classes: Computer Technology and Recreational activity. These two classes include the following sub-classes:

a) 1.plot a histogram of the number of documents per topic to make sure they are evenly distributed.

There are 20 topics in total. The following graph is a histogram of the number of documents per topic.



In this graph, we can easily see that the number of documents distribution is nearly even. The topics group 0-19 corresponds to alt.atheism, comp.graphics, comp.os.mswindows.misc,'comp.sys.ibm.pc.hardware','comp.sys.mac.hardware'

,'comp.windows.x','misc.forsale','rec.autos','rec.motorcycles','rec.sport.baseball','rec.sport.hockey','sci.crypt','sci.electronics','sci.med','sci.space','soc.religion.christian','talk.politics.guns','talk.politics.mideast','talk.politics.misc','talk.religion.misc' respectively.

2.Then report the number of documents in the two groups above (Computer Technology and Recreational Activity).

The number of documents in Computer Technology is sum of group1,2,3,4=2343

The number of documents in Recreational Activity is the sum of

group 7,8,9,10=2389

Modeling Text Data and Feature Extraction:

b) After tokenizing each document and extracting all the terms excluding the stop words, punctuations, and different stems of a word, we then create the TFxIDF. The final number of terms extracted is 118403.

If we only take the eight classes in the table, namely, comp.graphics, comp.os.mswindows.misc,'comp.sys.ibm.pc.hardware','rec.autos','rec.motorcycles','rec.sport.baseball','rec.sport.hockey' into account, then the number of terms extracted is 65567.

c)

Step:

1. We merge all the data in one class together as one "document", so there are in total 20 "documents"
2. We apply method in b to find the TFxIDF matrix of the data
3. For the target four classes, we got the feature names of the ten largest value in TFxIDF matrix.

The 10 most significant terms in the following classes, namely comp.sys.ibm.pc.hardware , comp.sys.mac.hardware, misc.forsale, and soc.religion.christian are as following:

1. comp.sys.ibm.pc.hardware:

[u'doctrin', u'theybritish', u'pilfer', u'translabitsuciedu', u'wsinfo03wintuenl', u'linseman', u'cocounsel', u'insulationwal', u'newslett', u'jk87377cstutfi']

2.comp.sys.mac.hardware:

[u'pilfer', u'linseman', u'ivemucsdedu', u'cocounsel', u'insulationwal', u'wsinfo03wintuenl', u'apr10053146199314368athosrutgersedu', u'alaa', u'hp95lx', u’vsrt0d']

3. misc.forsale:

[u'linseman', u'cocounsel', u'xei4h', u'marijuana', u'insulationwal', u'somlia', u'pilfer', u'neep', u'mb8grahfairkirgrgrgfb9rmr9rgmairgmkmkvm', u’wsinfo03wintuenl']

4. soc.religion.christian:

[u'pilfer', u'affilit', u'soundogeek', u'jaleco', u'subthread', u'itraconazol', u'n4qqp', u'cocounsel', u'desire', u'linseman']

Feature selection:

d)Use the sklearn.decomposition.TruncatedSVD method to implement LSI. Set n_components=50 so that the features’ dimensionality can be reduced to 50, i.e. the number of the columns of the output matrix is lowered to 50.

Function used on TFxIDF matrix:

TruncatedSVD(n_components=50, random_state=42)

Result shape for all 20 sub-classes:

(11314, 50)

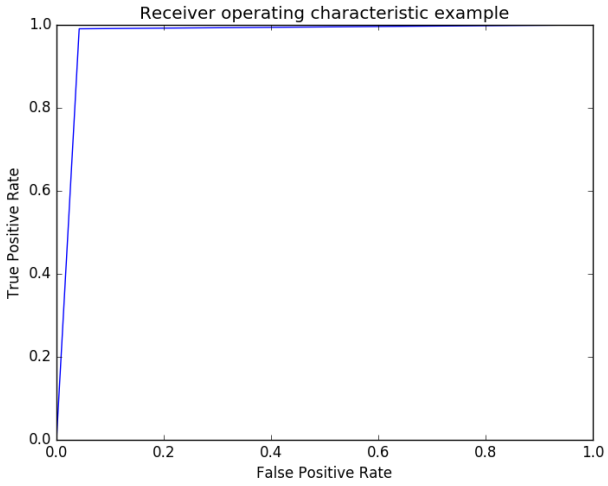
Learning Algorithms:

e) By feeding the features selected in (d) to Linear SVM, the efficiency of this algorithm is greatly improved. But the error doesn’t change much.

Accuracy = 0.974603174603

	Precision	recall	f1-score	support
Computer Tech	0.99	0.96	0.97	1560
Recreation	0.96	0.99	0.98	1590
avg/total	0.97	0.97	0.97	3150

And the ROC curve is as following:



The confusion matrix is as following:

[[1500 60]
[20 1570]]
We can see that after normalizing this matrix it will be approximately a diagonal matrix, which also indicates that our Linear SVM model achieves very high accuracy.

f) Implement the 5 fold cross validation and retrieve the scores from the 5 testing results individually:

γ = 1000:

[0.97336608 0.97589572 0.9689271 0.96509836 0.97142887]

γ = 100:

[0.90592958 0.9201904 0.90924689 0.87284895 0.96060846]

γ = 10:

[0.33863099 0.56290582 0.35120424 0.33898871 0.33863773]

γ = 1:

[0.33863099 0.32806248 0.33863099 0.33898871 0.32805581]

γ = 0.1:

[0.33863099 0.33863099 0.32806248 0.33898871 0.32805581]

γ = 0.01:

[0.32806248 0.32806248 0.32806248 0.33898871 0.33863773]

γ = 0.001:

[0.33863099 0.33863099 0.32806248 0.3277085   0.32805581]

□	0.001	0.01	0.1	1	10	100	1000
average score	0.3322	0.3324	0.3345	0.3345	0.3861	0.9138	0.9709

So we can see that γ = 1000 has the best result, when γ gets closer to 0 (the classifier is more like hard margin SVM) the result gets worse. This makes perfect sense in that compared to hard margin SVM, soft margin SVM is less constrained on the separating hyperplane.

when γ = 1000,

Accuracy = 0.968253968254

Confusion matrix:

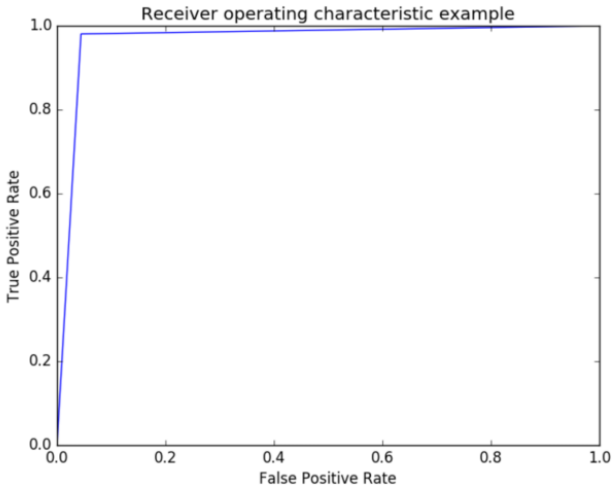
[[1491   69]

 [31 1559]]

Precision, recall:

	Precision	recall	f1-score	support
Computer Tech	0.98	0.96	0.97	1560
Recreation	0.96	0.98	0.97	1590
avg/total	0.97	0.97	0.97	3150

The ROC curve is as following:



g) Next, we use naïve Bayes algorithm for the same classification task.

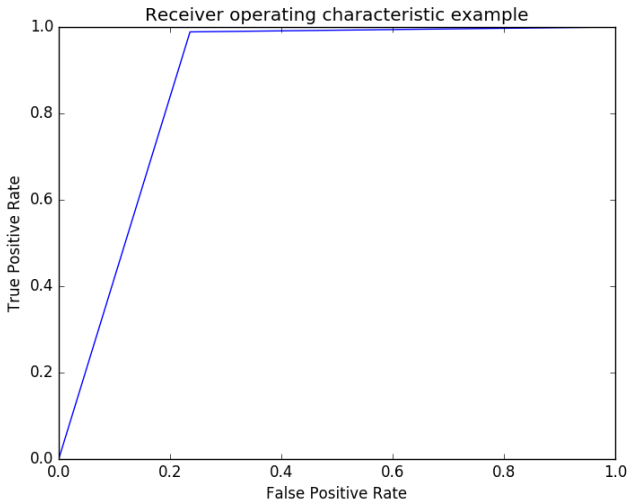
Accuracy:

0.87746031746

The confusion matrix is as following:

[[1192 368]
 [18 1572]]

The ROC curve is as following:

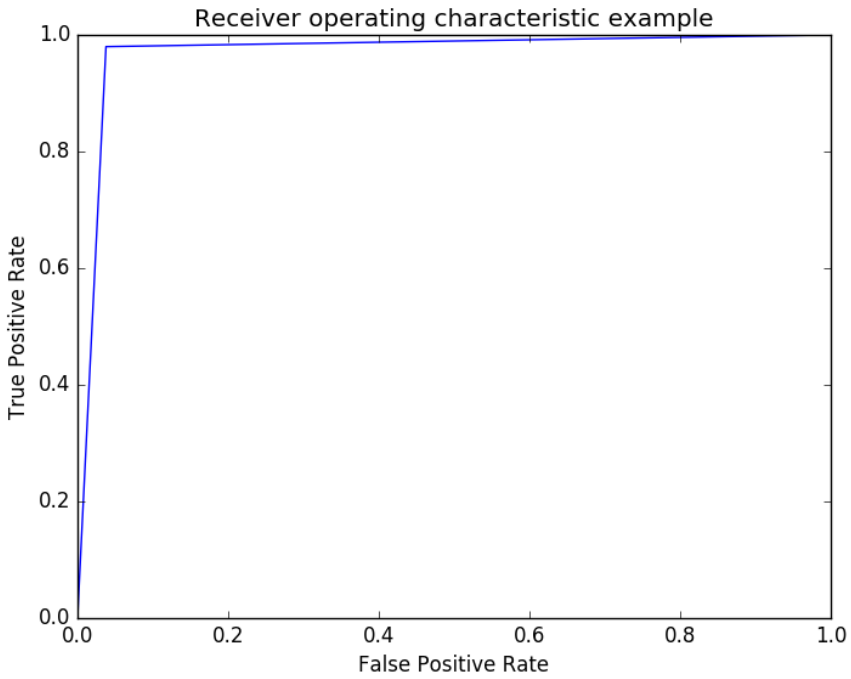


	Precision	recall	f1-score	support
Computer Tech	0.99	0.76	0.86	1560
Recreation	0.81	0.99	0.89	1590
avg/total	0.90	0.88	0.88	3150

h) Logistic regression classifier :

Accuracy: 0.97111111111

The ROC curve is as following:

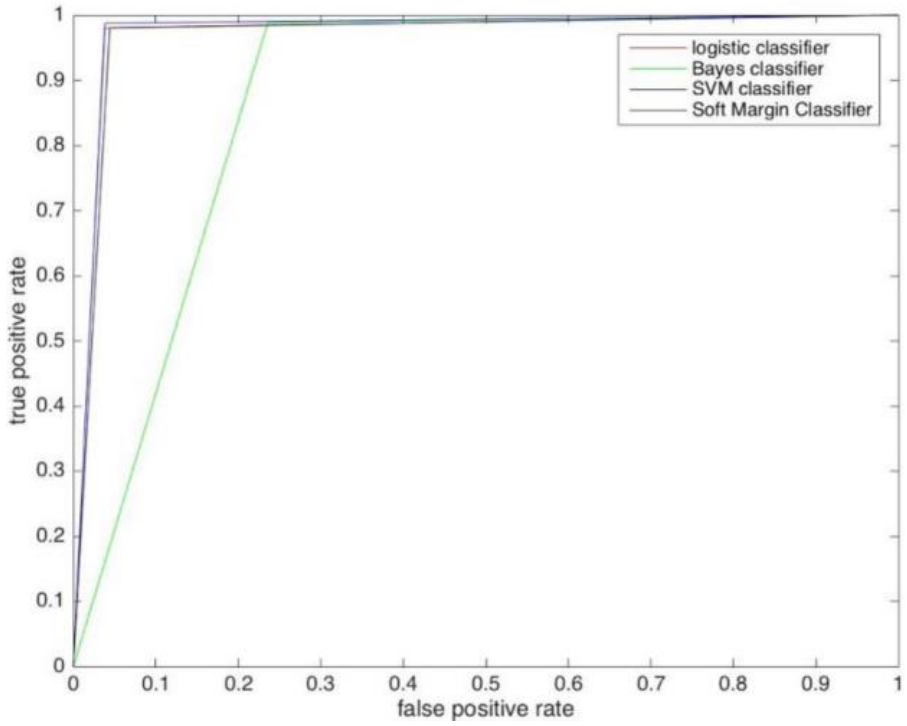


	Precision	recall	f1-score	support
Computer Tech	0.98	0.96	0.97	1560
Recreation	0.96	0.98	0.97	1590
avg/total	0.97	0.97	0.97	3150

The confusion matrix is as following:

[[1493 67]
[24 1566]]

Comparing the ROC results in all the learning algorithms:



Multiclass Classification:

i)

1.Perform Naïve Bayes classification:

Accuracy:

0.658146964856

Confusion matrix

[[224 32 136 0]

[76 152 156 1]

[56 19 313 2]

[0 0 57 341]]

Precision and recall:

	Precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.63	0.57	0.60	392
comp.sys.mac.hardware	0.75	0.39	0.52	385

misc.forsale	0.47	0.80	0.60	390
soc.religion.christian	0.99	0.86	0.92	398
avg/total	0.71	0.66	0.66	1565

2. multiclass SVM classification (One VS One)

Accuracy:

0.885623003195

Confusion Matrix:

[[342 36 14 0]

[49 310 26 0]

[28 10 351 1]

[6 0 9 383]]

	Precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.80	0.87	0.84	392
comp.sys.mac.hardware	0.87	0.81	0.84	385
misc.forsale	0.88	0.90	0.89	390
soc.religion.christian	1.00	0.96	0.98	398
avg/total	0.89	0.89	0.89	1565

3. multiclass SVM classification (One VS Rest):

Accuracy:

	Precision	recall	f1-score	support

comp.sys.ibm.pc.hardware	0.77	0.63	0.69	392
comp.sys.mac.hardware	0.70	0.83	0.76	385
misc.forsale	0.84	0.87	0.86	390
soc.religion.christian	0.99	0.95	0.97	398
avg/total	0.83	0.82	0.82	1565