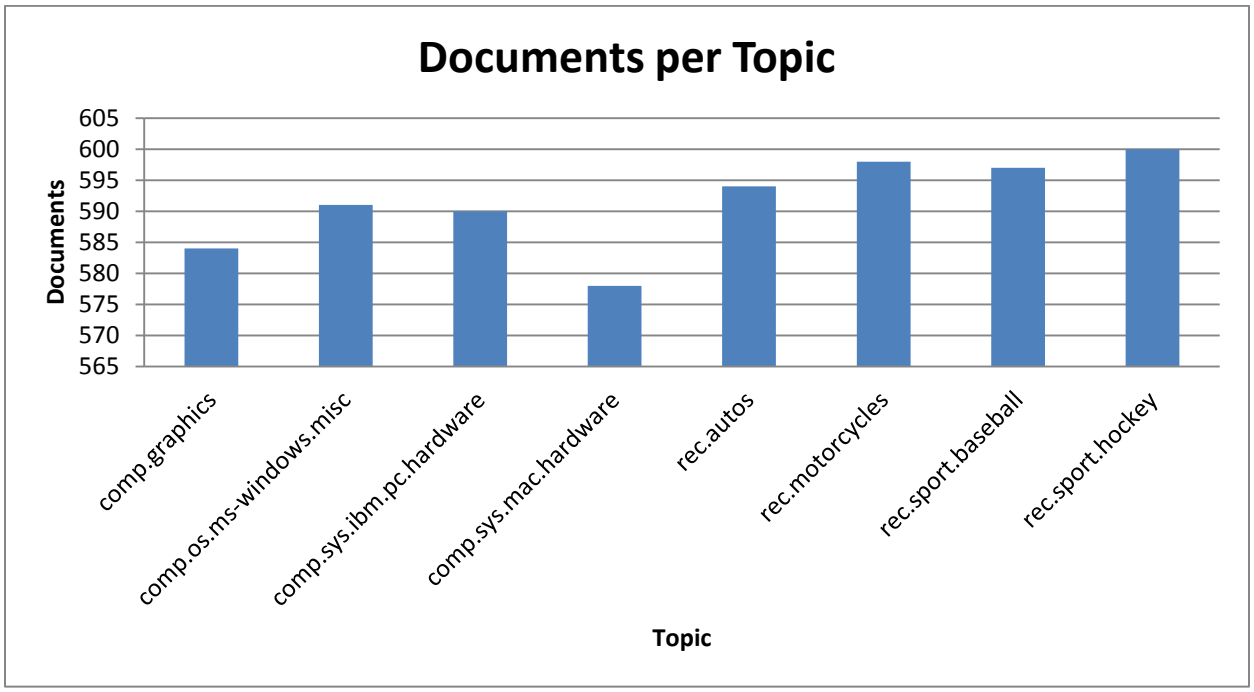
Kevin Wang (604036846)

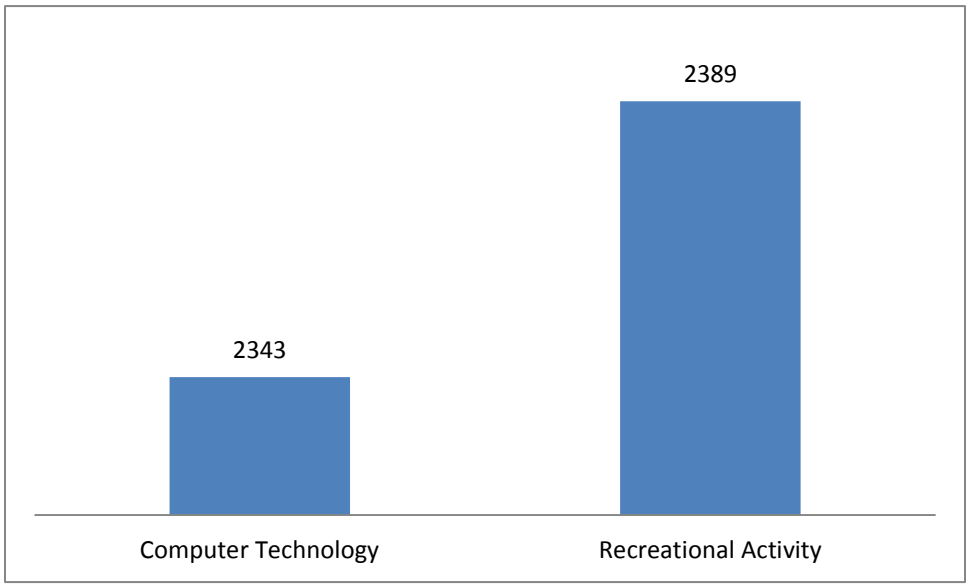
EE 239AS Project 1

Part a

The following is a plot of the number of documents for topic in the training set. The plot was made using Microsoft Excel.



Since 578 was the smallest number, the first 578 documents in each group should be used if the goal is to differentiate between the 8 topics. If the goal is to differentiate between the two main groups, then the first 2343 documents from the Recreational Activities group should be used instead. However, the numbers are all approximately equal, so it should not make much of a difference.



Part b

Since sklearn's built-in tokenizer does not perform stemming, a different tokenizer had to be used that involved using the NLTK library. An example of such a tokenizer was provided on this website: http://www.cs.duke.edu/courses/spring14/compsci290/assignments/lab02.html
The provided tokenizer did not remove punctuation, so an additional line was added to do so.

The output of the td-idf function returns a matrix of dimension (11314, 168307). The 11314 corresponds to the number of training set documents, so this particular tokenizer was able to extract 168,307 terms from across the documents.

Part c

First, all of the documents belonging to the same class were concatenated together with a space in between. This is such that the sklearn CountVectorizer and TfidfTransformer functions can be called on a new data matrix that has 20 rows representing the 20 classes instead of 11314 rows for each separate document. Then, the highest TFxICF values are found:

comp.sys.ibm.pc.hardware	**comp.sys.mac.hardware**	**misc.forsale**	**soc.religion.christian**
scsi	edu	edu	god
edu	lines	00	edu
ide	subject	lines	christians
drive	mac	subject	jesus
com	organization	sale	subject
lines	apple	organization	people
subject	quadra	new	church
organization	scsi	com	lines
controller	com	university	christ
card	centris	posting	bible

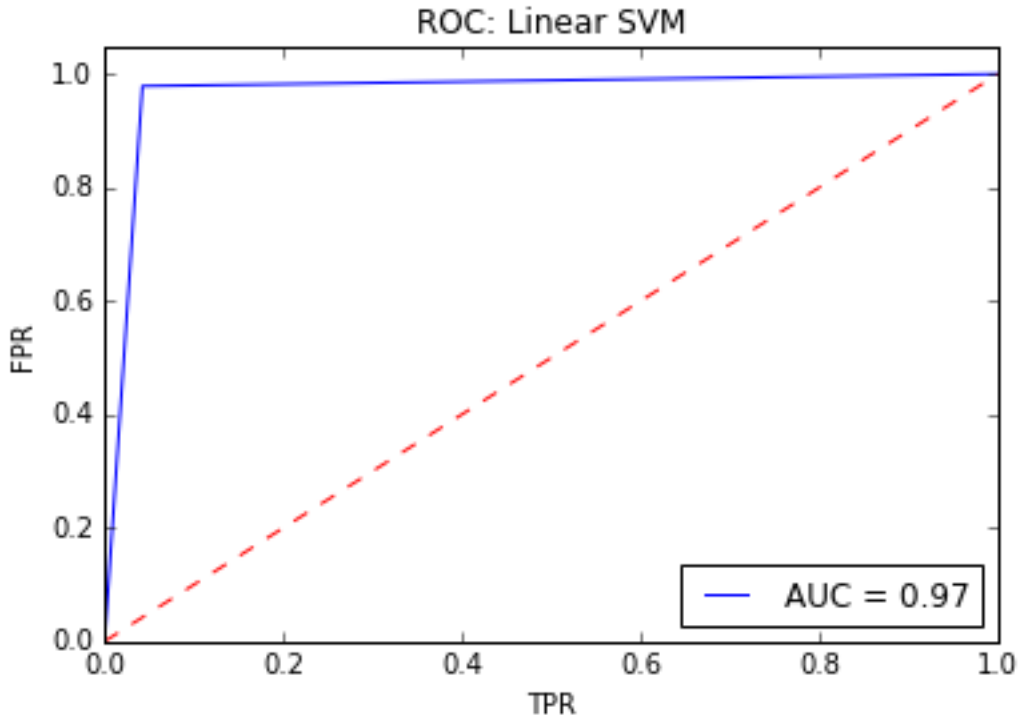
Part d

In this part, since there was no specified data set, the TFxIDF matrix from the whole training set is used to practice using LSI/LSA using the TruncatedSVD function of sklearn.

The results show that the dimensions of the matrix were reduced from (11314, 129797) to (11314, 50).

Part e

Part E involved using a linear SVM as the classifier, so sklearn's LinearSVC function was used for the predictions. The data for the 2 categories Computer Technology and Recreational Activity are first converted into binary form of either 'r' or 'c'. LSI is then performed to reduce the number of features to 50, and TFxIDF matrices are found. The model is then applied to the test data set, and the following metrics are found:



Thresholds = [2, 1, 0]

	Precision	Recall	F1-score	Support
c	0.98	0.96	0.97	1560
r	0.96	0.98	0.97	1590
avg/total	0.97	0.97	0.97	3150

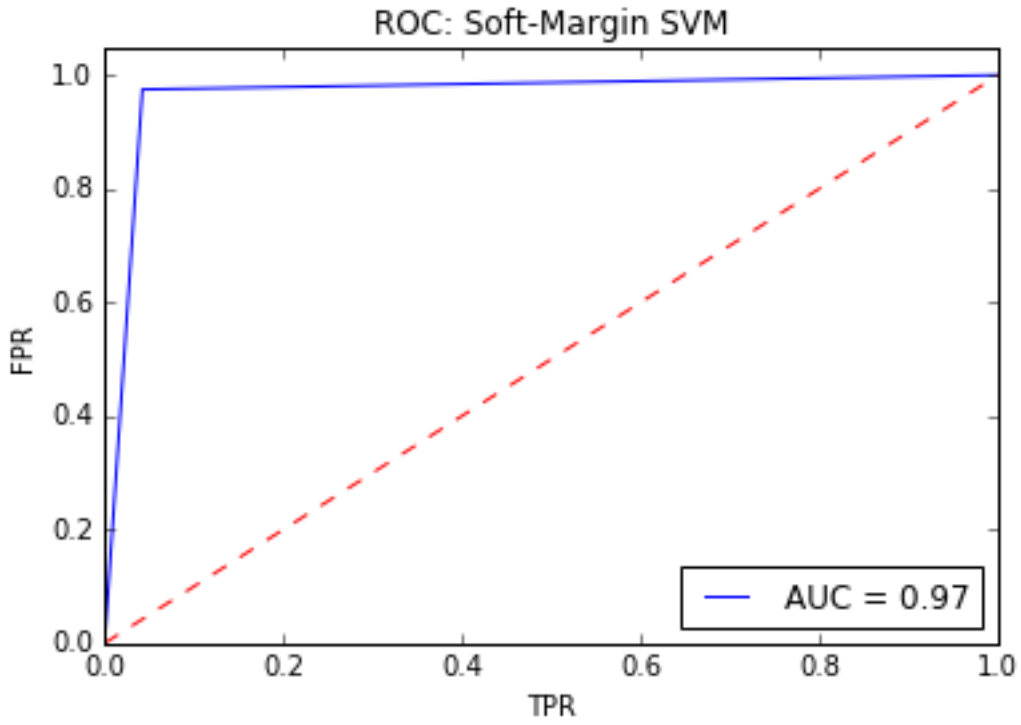
Accuracy score: 0.968253968254

Confusion matrix:

1494	66
34	1556

Part f

In this part, the soft-margin SVM is used instead. When LinearSVC is called, the parameter “loss” is set to “hinge” instead of the default “hinge_squared” in this case.



Thresholds = [2, 1, 0]

	Precision	Recall	F1-score	Support
c	0.97	0.96	0.97	1560
r	0.96	0.97	0.97	1590
avg/total	0.97	0.97	0.97	3150

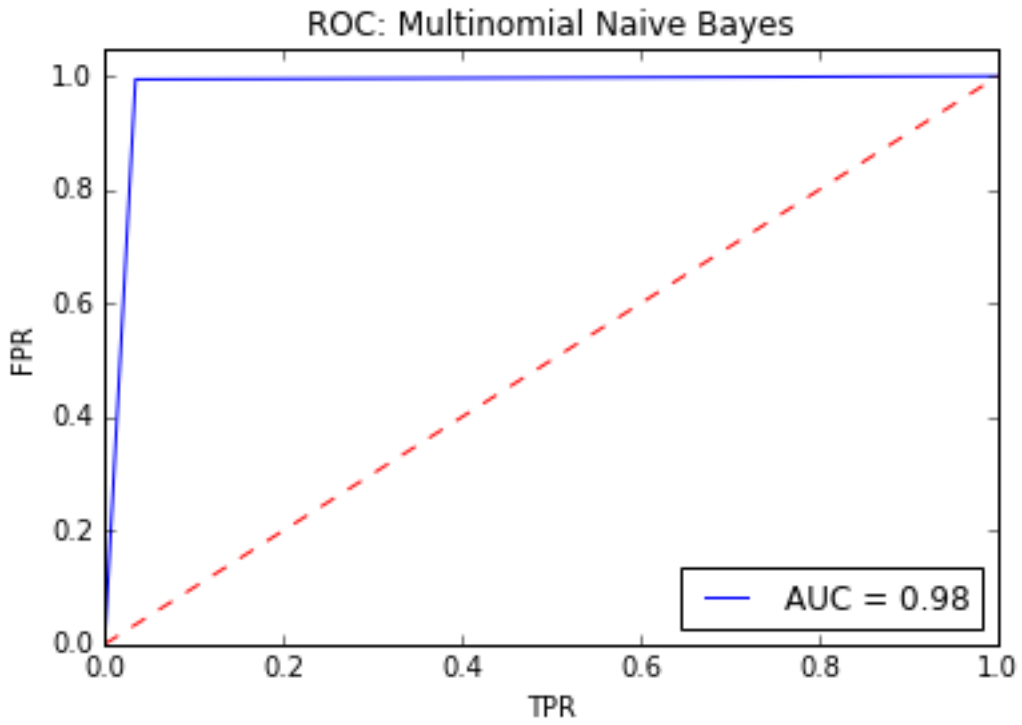
Accuracy score: 0. 966349206349

Confusion matrix:

1494	66
34	1550

Part g

The Multinomial Naïve Bayes classifier provided by sklearn is not compatible with negative terms in the input matrix, so in this case LSI is not performed since it results in negative numbers in the reduced TFxIDF matrix. The results are still pretty good, as shown below:



Thresholds = [2, 1, 0]

	Precision	Recall	F1-score	Support
c	0.99	0.97	0.98	1560
r	0.97	0.99	0.98	1590
avg/total	0.98	0.98	0.98	3150

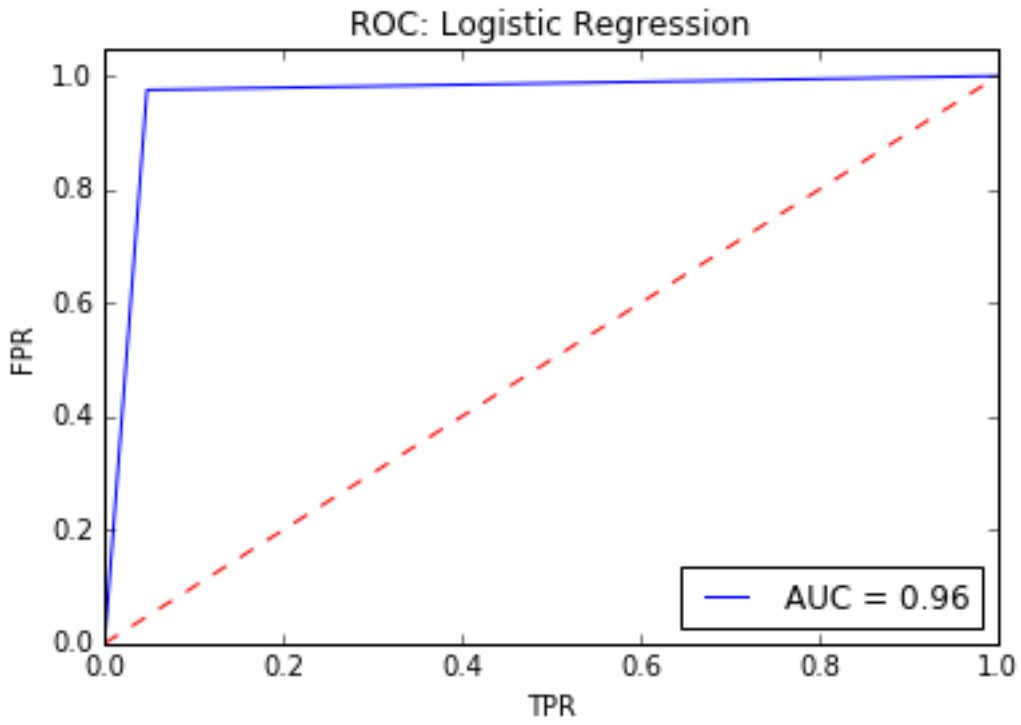
Accuracy score: 0. 98

Confusion matrix:

1506	54
9	1581

Part h

This part involved the Logistic Regression Classifier:



Thresholds = [2, 1, 0]

	Precision	Recall	F1-score	Support
c	0.97	0.95	0.96	1560
r	0.95	0.98	0.96	1590
avg/total	0.96	0.96	0.96	3150

Accuracy score: 0. 964126984127

Confusion matrix:

1486	74
39	1551

Part i

Naïve Bayes Classification Results:

	Precision	Recall	F1-score	Support
comp.sys.ibm.pc.hardware	0.83	0.90	0.87	392
comp.sys.mac.hardware	0.89	0.88	0.89	385
misc.forsale	0.94	0.88	0.91	390
soc.religion.christian	0.99	0.99	0.99	398
avg/total	0.92	0.91	0.91	1565

Accuracy score: 0. 913099041534

Confusion matrix:

353	26	13	0
35	340	8	2
31	14	343	2
4	1	0	393

One vs. One Linear SVM Classification Results:

	Precision	Recall	F1-score	Support
comp.sys.ibm.pc.hardware	0.82	0.82	0.82	392
comp.sys.mac.hardware	0.82	0.84	0.83	385
misc.forsale	0.89	0.89	0.89	390
soc.religion.christian	1.00	0.96	0.98	398
avg/total	0.88	0.88	0.88	1565

Accuracy score: 0. 879872204473

Confusion matrix:

323	49	20	0
42	323	20	0
23	19	347	1
8	3	3	384

One vs. Rest Linear SVM Classification Results:

	Precision	Recall	F1-score	Support
comp.sys.ibm.pc.hardware	0.84	0.81	0.83	392
comp.sys.mac.hardware	0.83	0.83	0.83	385
misc.forsale	0.87	0.91	0.89	390
soc.religion.christian	0.99	0.98	0.99	398
avg/total	0.89	0.89	0.89	1565

Accuracy score: 0. 885623003195

Confusion matrix:

319	49	24	0
37	321	26	1
18	14	356	2
5	1	2	390

The results are all pretty good. It is not surprising that soc.religion.christian was the most accurate since it is the most different topic compared to the others. The two comp categories had the most confusion since they are similar topics, and misc.forsale could have involved selling computers.