EE 219 Project 1
Classification Analysis on Textual Data

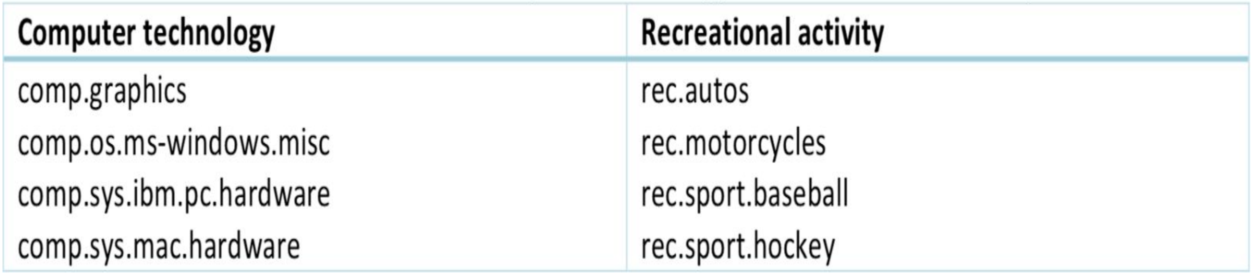
Konark J S Kumar - 204759469
Shreyas Lakhe - 105026650

Dataset

In this project, we work with “20 Newsgroups” dataset. It is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic.

For the purposes of this project we will be working with only 8 of the classes as shown in Table 1. The 8 subclasses fall under two major classes ‘Computer Technology’ and ‘Recreational activity’.

Table 1. Subclasses of ‘Computer technology’ and ‘Recreational activity’



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

We begin our classification problem by plotting a histogram of the number of training documents class to check if they are evenly distributed. And as seen from the histogram in figure 1 below. we can infer that the data set is already balanced with the number of documents evenly distributed across all the classes and hence we do not need to balance it ourselves.

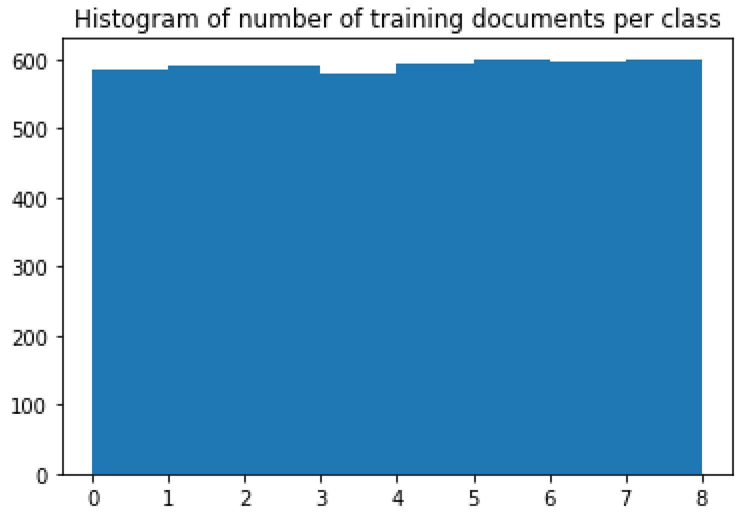


Fig 1: Number of training documents per class for the eight classes

.

Modeling Text Data and Feature Extraction

Document representation is a primary step in classifying a corpus of text and in this project we use the ‘Bag of Words’ representation. It is a simplistic representation where a document is represented as a histogram of term frequencies, or other statistics of the terms, within a fixed vocabulary. The corpus of text can be summarized into a term-document matrix where the entries in the matrix are statistics of the terms.

Filtering and preprocessing data is an important step before representing data as a term-document matrix. In this process, to avoid unnecessarily large feature vectors, some of the terms are dropped from the feature set based on the following criteria

- Terms that are too frequent and present in almost every document.
- Terms that are extremely rare.
- Special characters
- Common stop words - In the project we have used the stop words available in the NLTK library as they have a comprehensive list of stop words.
- Stemming terms - words that share the same stem in the vocabulary are merged into a single term.

In the next step, we need to choose a statistical metric to associate with every term in the document. Term Frequency - Inverse Document Frequency (TFxIDF) is a popular numerical statistic that captures the importance of a word to a document in a corpus. It takes into the frequency of the term in the document, normalized by a certain function of the frequency of the individual words in the whole corpus.

Over the course of the project we have used to two settings for minimum document frequency of vocabulary terms namely, min_df = 2 and min_df = 5

The number of terms extracted for each setting is reported below:

- Number of terms when min_df = 2 is 19406.
- Number of terms when min_df = 5 is 59058.

Most Significant Terms in a class

In this section we find out the 10 most significant terms for some of the classes.

In order to signify how significant a word is to a class, we define a measure called TFxICF. It is similar to TFxIDF, except we compute inverse class frequency instead of document.

The 10 most significant words / terms for the four classes are reported below:

- **Comp.sys.ibm.pc.hardware**
  'system', ‘card', 'organ', 'subject', 'line', 'use', ‘com', 'scsi', 'edu', 'drive'
- **Comp.sys.mac.hardware**
  ‘drive', 'problem', 'post', 'appl', 'use', 'organ', 'subject', 'mac', ‘line', 'edu'
- **Misc.forsale**
  'use', 'univers', 'new', ‘post', 'com', 'sale', 'organ', 'subject', 'line', 'edu’
- **Soc.religion.christian**
  ‘line', 'peopl', ‘subject', 'church', 'jesu', 'would', 'one', 'edu', 'christian', 'god'

Feature Selection

The corpus is represented as a document-term matrix, and the dimensionality of the representation (TFxIDF) vectors ranges in the order of thousands. Since the matrix is sparse and low-rank learning algorithms may perform poorly in high-dimensional data which demands the need for the matrix to be reduced to lower dimensional space in which we can have a select subset of the original features which are more relevant.

In this project, we mainly use

- Latent Semantic Indexing (LSI)
- Non-Negative Matrix Factorization (NMF)

Learning Algorithms

Our task now is to classify documents into two categories namely, “Computer Technology” and “Recreational Activity”.

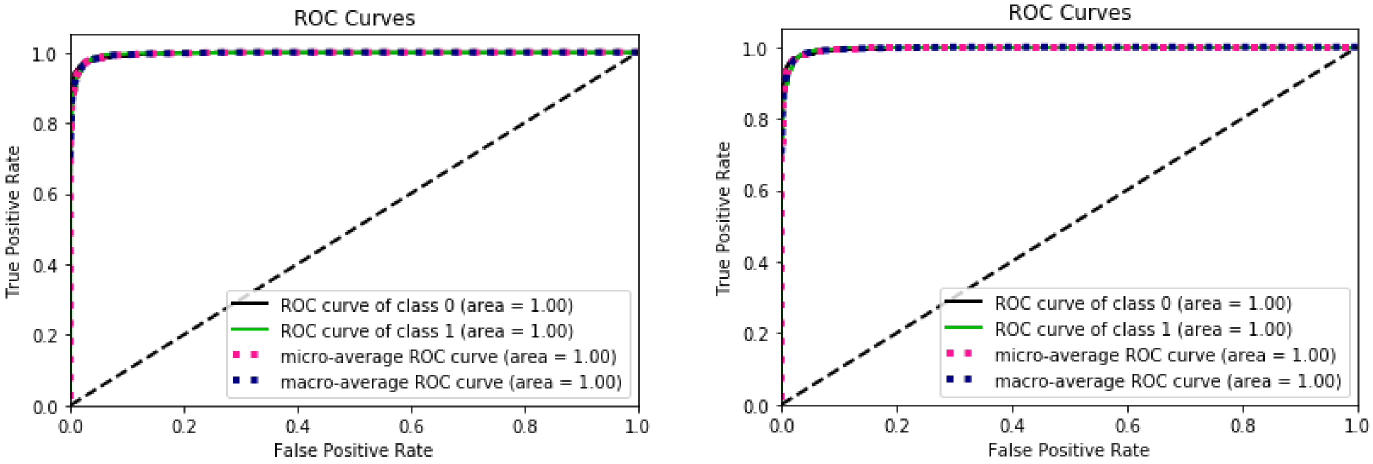
In this section we look at various classification techniques and observe the results. While we vary the classification technique, we also vary the following parameters

- Minimum document frequency - With a goal to find out whether it is worth keeping many rare terms.
- We also vary the dimensionality reduction technique LSI, NMF

Using LSI and varying min_df factor

1. **Hard SVM**

In this we have used a Hard svm with the value of C set to 1000. Hard margin classifier highly penalizes the misclassified data points, so the model tried to fit to the dataset as much as possible. Hence we get a high **accuracy of around 97%** for both min_df cases.



ROC curves for SVM Hard classifier. On the left is the graph obtained for min_df = 2 and on the right min_df = 5

Confusion Matrix when min_df = 2

	Predicted False	Predicted True
Actual False	1511	49
Actual True	35	1555

Confusion Matrix when min_df = 5

	Predicted False	Predicted True
Actual False	1507	53
Actual True	38	1552

Classification Report when min_df = 2

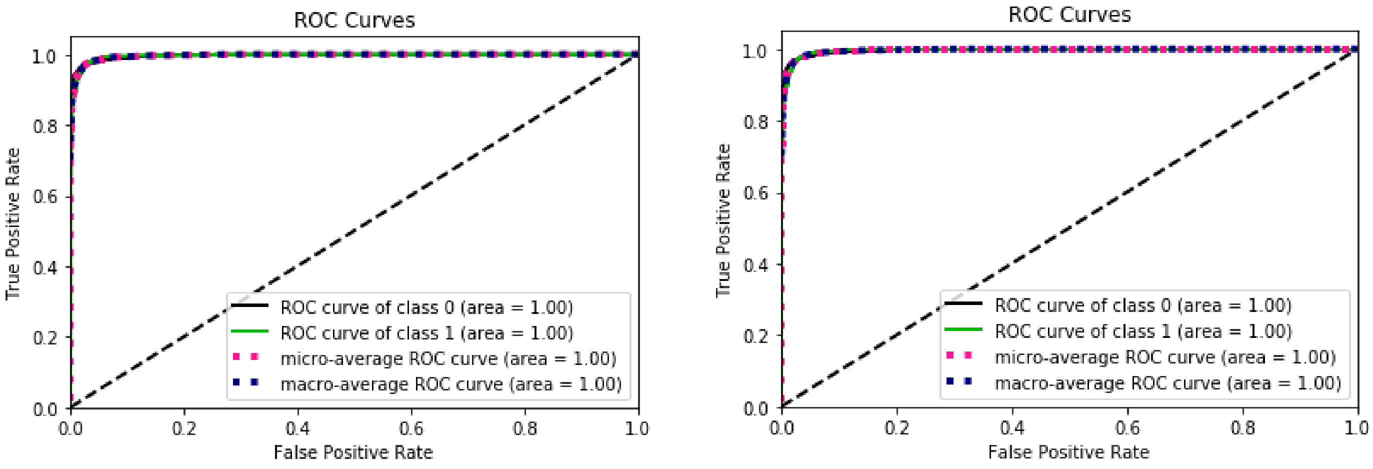
	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.97	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

Classification Report when min_df = 5

	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.97	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

2. Soft SVM

In this we have used a soft svm with the value of C set to 0.001. Soft margin classifier doesn’t penalize the misclassified points as much as hard margin classifier as long as most of the points are classified correctly. So the model doesn’t try to fit completely fit the dataset as much as possible. Hence we get a low **accuracy score of around 50.47%** for both min_df cases.



ROC curves for SVM Soft classifier. On the left is the graph obtained for min_df = 2 and on the right min_df = 5

Confusion Matrix when min_df = 2

	Predicted False	Predicted True
Actual False	0	1560
Actual True	0	1590

Confusion Matrix when min_df = 5

	Predicted False	Predicted True
Actual False	0	1560
Actual True	0	1590

Classification Report when min_df = 2

	Precision	Recall	F1 Score	Support
Computer Technology	0.0	0.0	0.00	1560
Recreational Activity	0.50	1.0	0.67	1590
Avg / Total	0.25	0.50	0.34	3150

Classification Report when min_df = 5

	Precision	Recall	F1 Score	Support
Computer Technology	0.0	0.0	0.00	1560
Recreational Activity	0.50	1.0	0.67	1590
Avg / Total	0.25	0.50	0.34	3150

3. SVM Cross Validation

In this we have to tried to obtain the best value for the parameter C, by comparing the accuracy scores obtained on 5-fold cross validation.

The best value of the parameter γ is 1000 for both cases when min_df = 2 and min_df = 5 with high **accuracy of around 97%**

Confusion Matrix when min_df = 2

	Predicted False	Predicted True

Actual False	1506	54
Actual True	35	1555

Confusion Matrix when min_df = 5

	Predicted False	**Predicted True**
Actual False	1507	53
Actual True	33	1557

Classification Report when min_df = 2

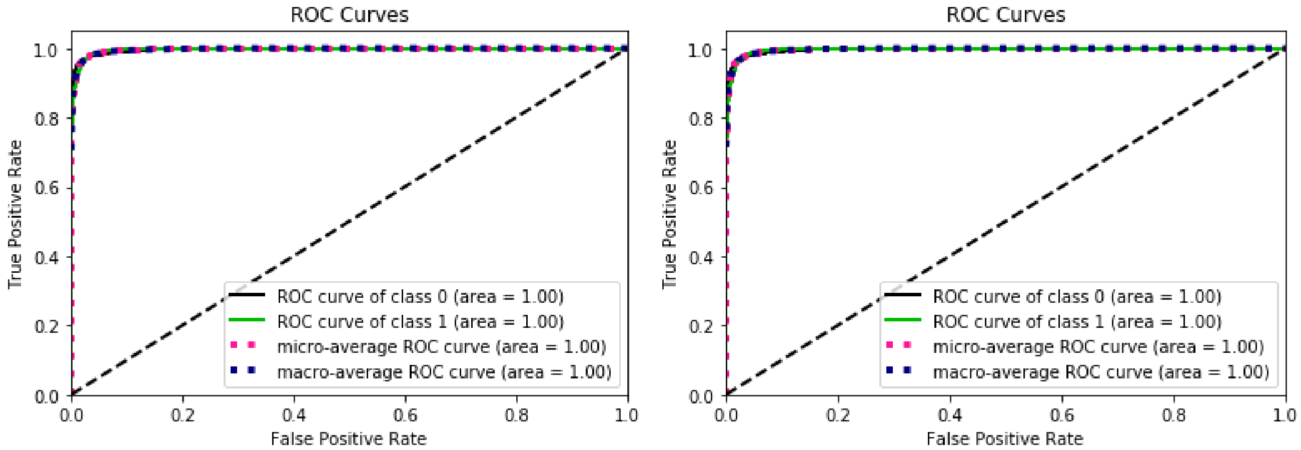
	Precision	**Recall**	**F1 Score**	**Support**
Computer Technology	0.98	0.97	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

Classification Report when min_df = 5

	Precision	**Recall**	**F1 Score**	**Support**
Computer Technology	0.98	0.97	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

4. Logistic Regression without regularization

Here we have used Logistic Regression without regularization and it gives an **accuracy of 96.4%** for both min_df cases.



ROC curves for Logistic Regression without regularization. On the left is the graph obtained for min_df = 2 and on the right min_df = 5

Confusion Matrix when min_df = 2

	Predicted False	Predicted True
Actual False	1505	55
Actual True	32	1558

Confusion Matrix when min_df = 5

	Predicted False	Predicted True
Actual False	1504	56
Actual True	34	1556

Classification Report when min_df = 2

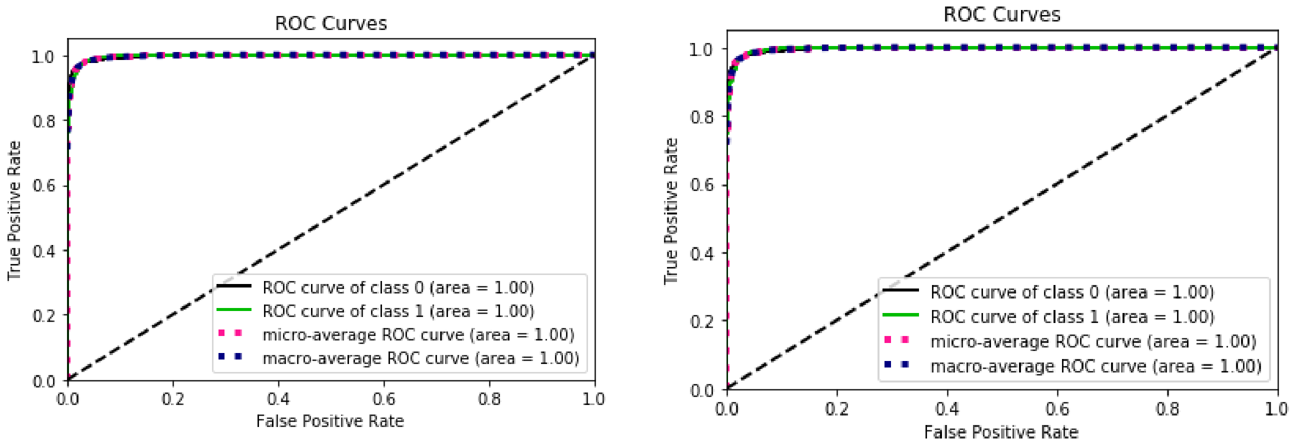
	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.95	0.96	1560
Recreational Activity	0.95	0.98	0.96	1590
Avg / Total	0.96	0.96	0.96	3150

Classification Report when min_df = 5

	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.95	0.96	1560
Recreational Activity	0.95	0.98	0.96	1590
Avg / Total	0.96	0.96	0.96	3150

5. Logistic regression with L2 regularization

Here we have used Logistic Regression with L2 regularization and it gives an **accuracy of 97.2%** for both min_df cases.



ROC curve for Logistic Regression with L2 regularization. On the left is the curve for min_df = 2 and on the right min_df = 5

Confusion Matrix when min_df = 2

	Predicted False	Predicted True
Actual False	1505	55
Actual True	33	1557

Confusion Matrix when min_df = 5

	Predicted False	Predicted True
Actual False	1506	54
Actual True	33	1557

Classification Report when min_df = 2

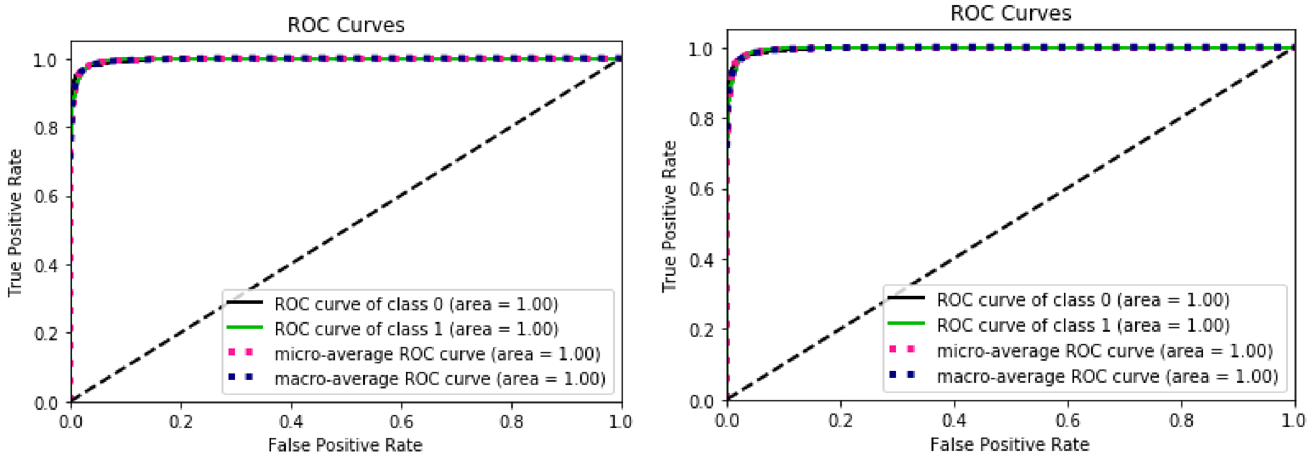
	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.96	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

Classification Report when min_df = 5

	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.97	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

6. Logistic regression with L1 regularization

Here we have used Logistic Regression with L2 regularization and it gives an **accuracy of 97.2%** for both min_df cases.



ROC curve for Logistic Regression with L1 regularization. On the left is the curve for min_df = 2 and on the right min_df = 5

Confusion Matrix when min_df = 2

	Predicted False	Predicted True
Actual False	1505	55
Actual True	32	1558

Confusion Matrix when min_df = 5

	Predicted False	Predicted True
Actual False	1504	56
Actual True	34	1556

Classification Report when min_df = 2

	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.96	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590

Avg / Total	0.97	0.97	0.97	3150

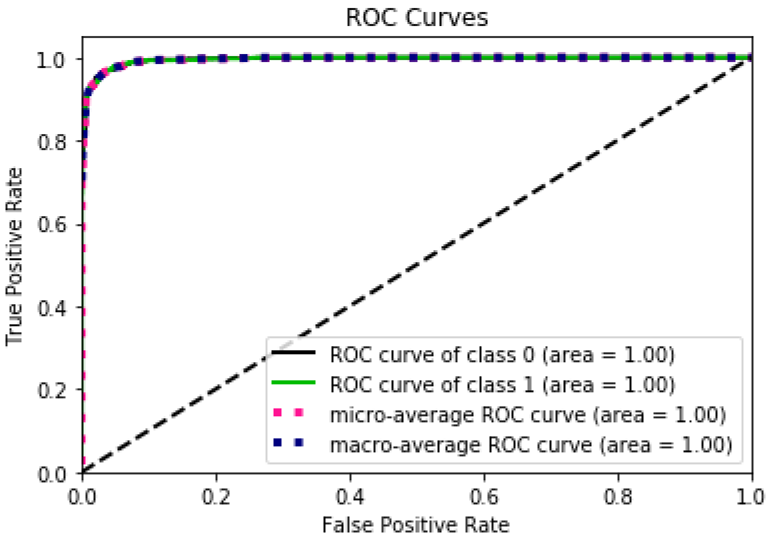
Classification Report when min_df = 5

	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.96	0.97	1560
Recreational Activity	0.97	0.98	0.97	1590
Avg / Total	0.97	0.97	0.97	3150

Using NMF and setting min_df = 2

1. Hard SVM

In this we have used a Hard svm with the value of C set to 1000. Hard margin classifier highly penalizes the misclassified data points, so the model tried to fit to the dataset as much as possible. Hence we get a high **accuracy of around 96.53%**.



Confusion Matrix

	Predicted False	Predicted True
Actual False	1499	61

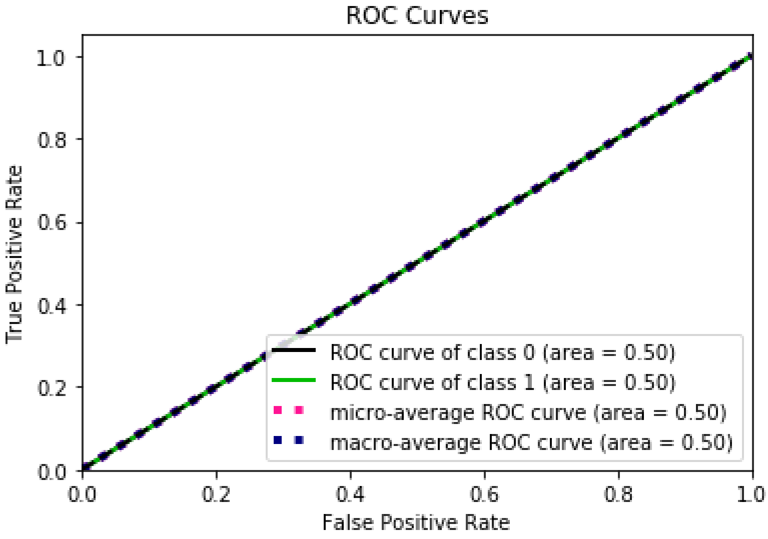
Actual True	48	1542

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.97	0.96	0.96	1560
Recreational Activity	0.96	0.97	0.97	1560
Avg / Total	0.97	0.97	0.97	3150

2. Soft SVM

In this we have used a soft svm with the value of C set to 0.001. Soft margin classifier doesn't penalize the misclassified points as much as hard margin classifier as long as most of the points are classified correctly. So the model doesn't try to fit completely fit the dataset as much as possible. Hence we get a low **accuracy score of around 50.47%**



Confusion Matrix

	Predicted False	Predicted True
Actual False	0	1560
Actual True	0	1590

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.00	0.00	0.00	1560
Recreational Activity	0.50	1.00	0.67	1590
Avg / Total	0.25	0.50	0.34	3150

3. SVM Cross Validation

In this we have to tried to obtain the best value for the parameter C, by comparing the accuracy scores obtained on 5-fold cross validation, the best accuracy was obtained for C value of 1000 and the **accuracy was 96.539%.**

Confusion Matrix

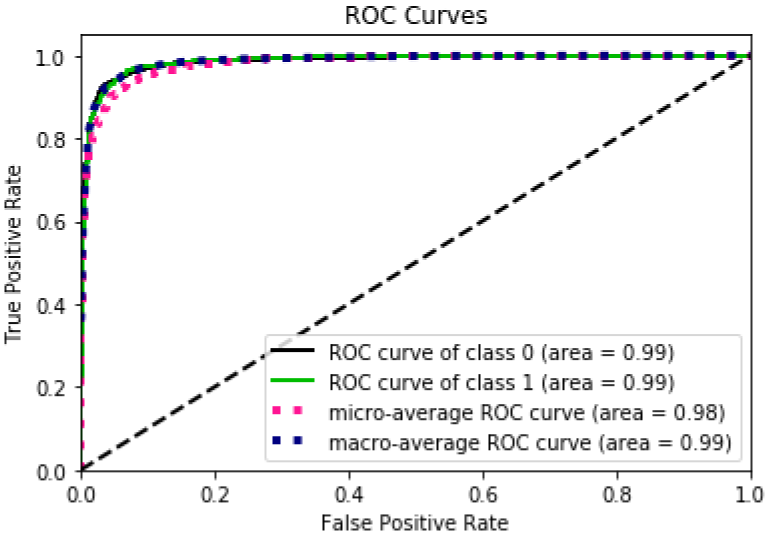
	Predicted False	Predicted True
Actual False	1470	90
Actual True	50	1540

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.97	0.94	0.95	1560
Recreational Activity	0.94	0.97	0.96	1590
Avg / Total	0.96	0.96	0.96	3150

4. Naive Bayes

In this, we have used a Multinomial Naive Bayes model to fit on the data. It fits well with an **accuracy of 93.84%**.



Confusion Matrix

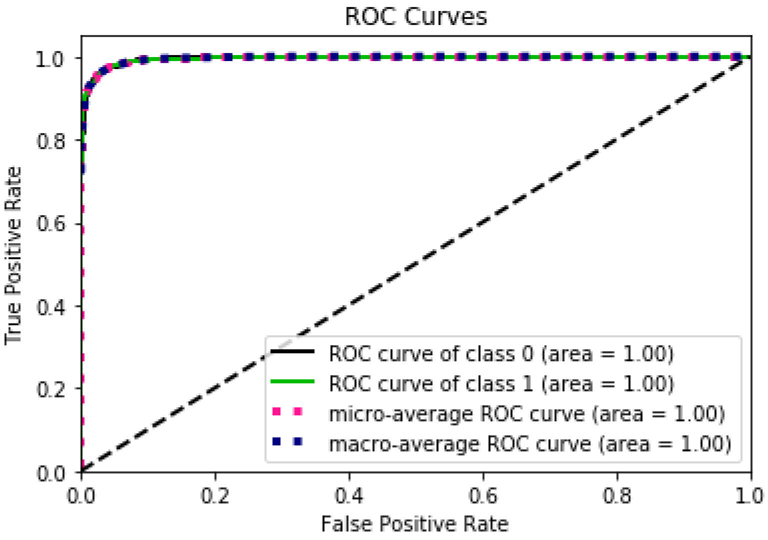
	Predicted False	Predicted True
Actual False	1373	187
Actual True	32	1558

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.98	0.88	0.93	1560
Recreational Activity	0.89	0.98	0.93	1590
Avg / Total	0.93	0.93	0.93	3150

5. Logistic

Here we have used Logistic Regression without regularization and it gives an **accuracy of 96.4%.**



Confusion Matrix

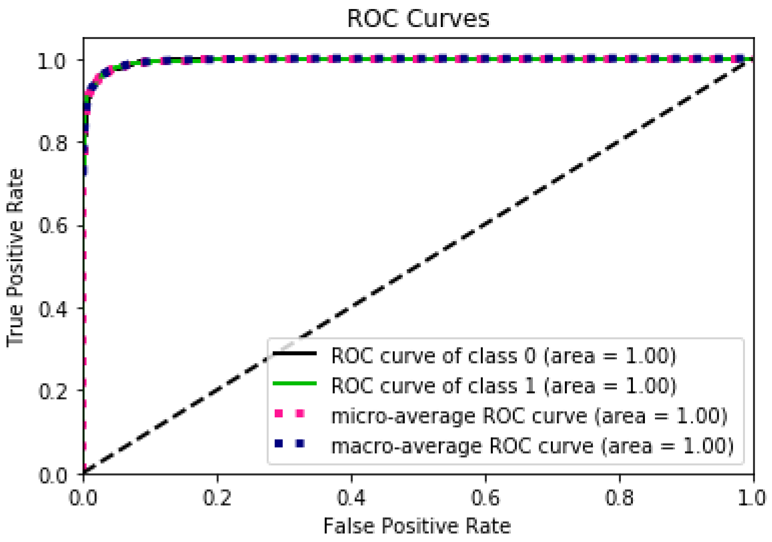
	Predicted False	Predicted True
Actual False	1501	59
Actual True	52	1538

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.97	0.96	0.96	1560
Recreational Activity	0.96	0.97	0.97	1590
Avg / Total	0.96	0.96	0.96	3150

6. Logistic w/L1

Here we have used Logistic Regression with L1 regularization and it gives an **accuracy around 96.4%.**



Confusion Matrix

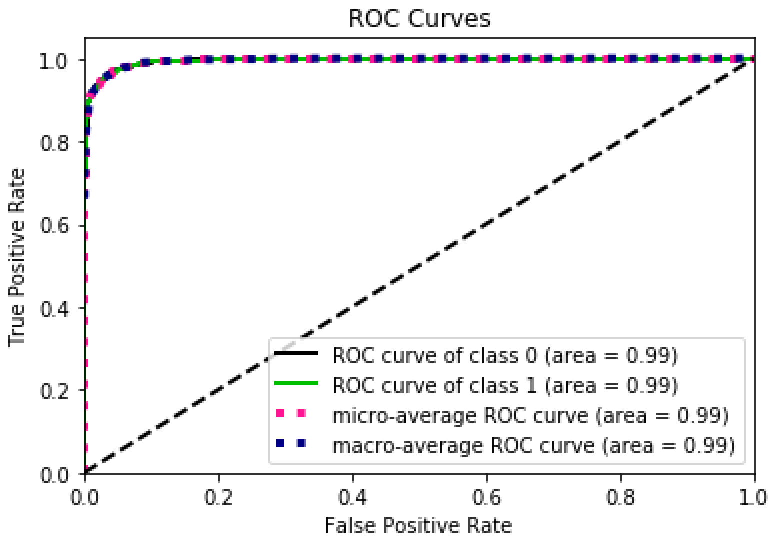
	Predicted False	Predicted True
Actual False	1501	59
Actual True	52	1538

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.97	0.96	0.96	1560
Recreational Activity	0.96	0.97	0.97	1590
Avg / Total	0.96	0.96	0.96	3150

7. Logistic w/L2

Here we have used Logistic Regression with L2 regularization and it gives an **accuracy of around 96.8%.**



Confusion Matrix

	Predicted False	Predicted True
Actual False	1486	74
Actual True	50	1540

Classification Report

	Precision	Recall	F1 Score	Support
Computer Technology	0.97	0.95	0.96	1560
Recreational Activity	0.95	0.97	0.96	1590
Avg / Total	0.96	0.96	0.96	3150

MULTICLASS CLASSIFICATION

Here we perform multiclass classification for classes “comp.sys.ibm.pc.hardware”, “comp.sys.mac.hardware”, “misc.forsale”, “soc.religion.christian” with these classes assigned labels 0, 1, 2 and 3 respectively.

A. Multiclass classification using NMF and min_df = 2

1. One Vs One Multiclass Naive Bayes

Here we have used one vs one multiclass naive bayes and it gives an **accuracy of 80.57%**.

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	316	28	44	4
Actual 1	108	235	38	4
Actual 2	56	14	315	5
Actual 3	1	0	2	395

Classification Report

	Precision	Recall	F1 Score	Support
Class 0	0.66	0.81	0.72	392
Class 1	0.85	0.61	0.71	385
Class 2	0.79	0.81	0.89	390
Class 3	0.97	0.99	0.98	398
Avg / Total	0.82	0.81	0.80	1565

2. One Vs Rest Multiclass Naive Bayes

Here we have used One Vs Rest Multiclass Naive Bayes and it gives an **accuracy of 80.63%**.

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	310	36	41	5
Actual 1	99	243	36	7
Actual 2	51	17	312	10
Actual 3	0	0	1	397

Classification Report

	Precision	Recall	F1 Score	Support
Class 0	0.67	0.79	0.73	392
Class 1	0.82	0.63	0.71	385
Class 2	0.80	0.80	0.80	390
Class 3	0.95	1.00	0.97	398
Avg / Total	0.81	0.81	0.80	1565

3. One vs One Multiclass SVM

Here we have used One Vs One Multiclass SVM and it gives an **accuracy of 84.64%.**

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	309	60	21	2
Actual 1	68	287	28	2
Actual 2	23	23	342	2
Actual 3	6	3	2	387

Classification Report

	Precision	Recall	F1 Score	Support
Class 0	0.76	0.79	0.77	392
Class 1	0.77	0.75	0.76	385
Class 2	0.87	0.88	0.87	390
Class 3	0.98	0.97	0.98	398
Avg / Total	0.85	0.85	0.85	1565

4. One vs Rest Multiclass SVM

Here we have used One vs Rest multiclass SVM an it gives an **accuracy of 85.87%.**

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	312	50	26	4
Actual 1	60	295	28	2
Actual 2	23	21	343	3
Actual 3	2	1	1	394

Classification Report

	Precision	Recall	F1 Score	Support
Class 0	0.79	0.80	0.79	392
Class 1	0.80	0.77	0.78	385
Class 2	0.86	0.88	0.87	390
Class 3	0.98	0.99	0.98	398
Avg / Total	0.86	0.86	0.86	1565

B. Multiclass classification using LSI and min_df = 2

1. One vs One Multiclass SVM

Here we have used One Vs One Multiclass SVM and it gives an **accuracy of 88.37%.**

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	325	45	22	0
Actual 1	38	318	28	1
Actual 2	22	17	350	1
Actual 3	5	0	3	390

Classification Report

	Precision	Recall	F1 Score	Support
Class 0	0.83	0.83	0.83	392
Class 1	0.84	0.83	0.83	385

Class 2	0.87	0.90	0.88	390
Class 3	0.99	0.99	0.99	398
Avg / Total	0.88	0.88	0.88	1565

2. One Vs Rest Multiclass SVM
Here we have used One Vs Rest Multiclass SVM and it gives an **accuracy of 88.43%.**

Confusion Matrix

	Predicted 0	**Predicted 1**	**Predicted 2**	**Predicted 3**
Actual 0	320	53	17	2
Actual 1	35	323	27	0
Actual 2	22	17	348	3
Actual 3	4	0	1	393

Classification Report

	Precision	**Recall**	**F1 Score**	**Support**
Class 0	0.84	0.82	0.83	392
Class 1	0.82	0.84	0.83	385
Class 2	0.89	0.89	0.89	390
Class 3	0.99	0.99	0.99	398
Avg / Total	0.88	0.88	0.88	1565

Conclusion

In this project we have tried multiple techniques to perform the task of text classification on NewsGroup dataset. As the dataset was already balanced we directly started from the preprocessing step, in which we removed the punctuations and other non-alphabetic characters ,stop words and performed stemming to get their reduced form. On this preprocessed dataset we implemented TFIDF for feature extraction. As the features generated were sparse and high dimensional we then implemented two dimensionality reduction techniques, non negative matrix factorization and latent semantic indexing. On the results that we obtained by training various models on these, we observed that both these techniques have almost similar results with slight variations in accuracy of around 1-2 %. We tested Naive Bayes. Logistic Regression and SVM classifiers for this task, and observed that while Logistic Regression and properly tuned SVM models had almost similar performance with 96-97% accuracy, the naive bayes was slightly below them at 93%. After that we performed multiclass classification on four classes by comparing one vs one and one vs rest versions of all the above classifiers and got accuracies in the range of 84-88 % for SVM and around 80 % for Naive Bayes.