Project 1 : Classification Analysis on Textual Data

Aadithya Venkatanarayanan 404946465
Narendran Raghavan 404945767
Srividhya Balasubramanian 205023825

30th January 2018

1 Objective

In this project, we implement statistical classification with approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups that corresponds to different topics. The document is tokenized into words and stemming operation is performed to create the Term Frequency-Inverse Document Frequency (TFxIDF) vector representation. By applying Latent Semantic Indexing (LSI), the dimensionality of the document is reduced through Non-Negative Matrix Factorization (NMF). Later the document is classified into two categories: "Computer Technology" vs "Recreational Activity". Classification accuracy is evaluated by plotting the Receiver Operation Characteristic (ROC) curve and the confusion matrix is reported. Various types of algorithms/techniques such as Bayes algorithm,logistic regression classifier are used to perform the same classification task and the ROC curve is compared for the same. The final part involves learning classifiers in the document using Naive Bayes and multiclass SVM classification.

2 Question a

The dataset is loaded using the built-in dataset loader for 20 newsgroups from scikit-learn package. The two main classes in the document are "Computer Technology" and "Recreational activity" which have the following subclasses:

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

Histogram containing the number of training documents in each class are plotted as shown in figure 1. From the figure displayed in the next page, we can see that number of documents under the eight categories are plotted and that they are evenly distributed as well. Note that the sub classes belong to the two main classes - Computer Technology and Recreational Activity.

3 Modeling Text Data and Feature Extraction

3.1 Question b

A most popular numerical statistical tool to determine the importance of a word in the document is the Term Frequency-Inverse Document Frequency (TFxIDF) metric. Tokenization is the process used to parse the sentences and analyze them,which is done by splitting up the text into segments called tokens. They can be words (stop words) or punctuation marks. The stop words are those that occur too frequent in the document or very rarely and hence are dropped out of the vocabulary. They are generated from the

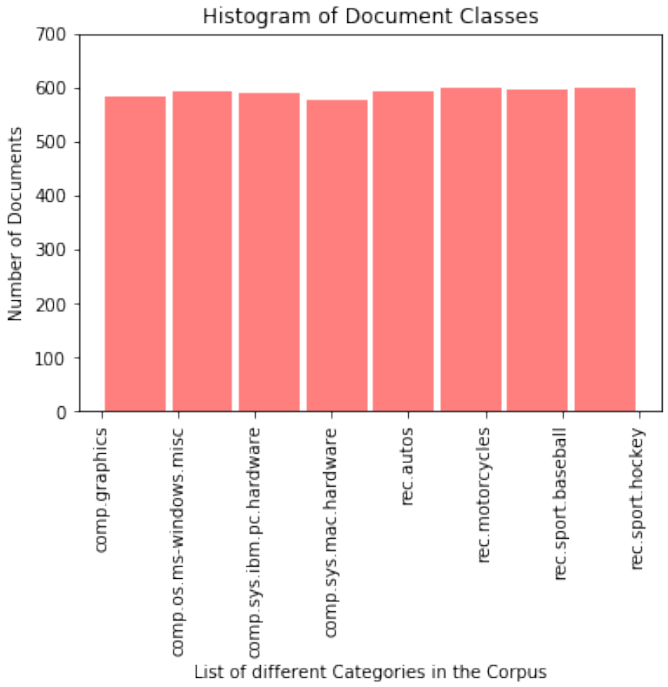


Figure 1: Number of documents in each sub class

list provided by scikit-learn package. We use CountVectorizer for this case. Terms that occur in less than two documents (min_df=2) and less than five documents (min_df=5) were removed. Hence this type of representation that does not include too much irrelevant information refers to "Bag of Words" where the document is represented as numerical terms such as term frequencies. In addition to this, words that share the same stem (eg., write,writing) are combined to form a (root) word. We use SnowBall Stemmer for this case. With the help of the above parameters, TFxIDF vector representation is created using the following definition:

$$TFxIDF(t,d) = tf(t,d) * idf(t)$$

where tf(t,d) represents the frequency of term t in document d and inverse document frequency is defined as

$$idf(t) = log[n/df(t)] + 1$$

where n is total number of documents and df(t) is number of documents that contain term t.
Output:
Dimensions of Numerical feature vector for training data: (4732, 11018)
Number of terms Extracted for training data: 11018
Dimensions of TF-IDF vector(4732, 11018)
Dimensions of Numerical feature vector for test data: (3150, 11018)
Number of terms Extracted for test data: 11018
Dimensions of TF-IDF vector for test data: (3150, 11018)

3.2 Question c

Similar to finding a significant term in a document, we find the significant term in the class using TFxICF that is computed as:

$$TFxICF(t,c) = tf(t,c) * icf(t)$$

where tf(t,c) represents term frequency in class c and inverse class frequency is defined as

$$icf(t) = log[n_{classes}/cf(t)] + 1$$

where cf(t) is class frequency and $n_{classes}$ is the total number of classes. 10 most significant words in the following classes are found using TCxICF measure: comp.sys.ibm.pc.hardware,comp.sys.mac.hardware, misc.forsale,soc.religion.

Output:
The top ten features in the list comp.sys.ibm.pc.hardware is
dict-keys (['card', 'use', 'control', 'know', 'disk', 'drive', 'problem', 'work', 'scsi', 'ani'])
The top ten features in the list comp.sys.mac.hardware is
dict-keys (['work', 'know', 'like', 'mac', 'problem', 'use', 'appl', 'monitor', 'ani', 'drive'])
The top ten features in the list misc.forsale is
dict-keys (['new', 'ship', 'pleas', 'email', 'sale', 'drive', 'price', 'use', 'includ', 'offer'])
The top ten features in the list soc.religion.christian is
dic-keys(['know', 'christian', 'sin', 'say', 'god', 'peopl', 'think', 'jesus', 'believ', 'church'])

3.3 Question d

The resulting representation vector ranges in the order of thousands. Also the document term tfidf matrix is sparse and low rank. Hence, we use Latent Semantic Indexing (LSI) to minimize mean square residual between original data and reconstruction from its low dimensional approximation. The results are fed into the MinMaxScaler which scales and translates each feature such that it lies in the given range on the training set which is between 0 and 1. If D denotes the txd term document matrix with rank r where each of the d columns represent a document vector, the approximation is best rank k approximation of D by minimizing the sum of squared differences between D and D_k. In addition to LSI, the dimensionality is reduced through Non-Negative Matrix Factorization (NMF) and thes reduced form is used.

Output:
Dimensions of TF-IDF matrix after LSI for $min\_df = 2$: (4732, 50)
Dimensions of TF-IDF matrix after LSI for $min\_df = 5$: (4732, 50)
Dimensions of TF-IDF matrix after NMF for $min\_df = 2$: (3150, 50)
Dimensions of TF-IDF matrix after NMF for $min\_df = 5$: (3150, 50)

4 Learning Algorithms

In the following parts,a number of techniques were used to define the confusion matrix,recall, precision and accuracy:

- SVM classifier- for both hard margin and soft margin
- 5-fold cross validation
- Bayes Algorithm
- Logistic Regression Classifier
- Logistic Regression Classifier with regularization term

The definitions of certain terms used while computing the above mentioned algorithms are given below:
Precision: It is a measure of result relevancy. It is defined as

$$P = T_p/T_p + F_p$$

Recall: It is a measure of how many truly relevant results are returned. It is defined as

$$R = T_p/T_p + F_n$$

where T_p , F_p and F_n are true positive,false positive and false negative rates respectively.
Accuracy : It is used as a statistical measure of how well a binary classification test correctly identifies or

excludes a condition (ie) proportion of true results(true positives and true negatives) to the total number of cases examined.

$$A = T_p + T_n / T_p + T_n + F_p + F_n$$

True positive rate : It is defined as the proportion of positives that are correctly identified as such.
True negative rate : It is defined as the proportion of negatives that are correctly identified as such.
Confusion Matrix : It is a special kind of contingency table with two dimensions - actual and predicted with identical sets of classes in both dimensions. Each combination of dimension and class is a variable in the contingency table. The following table gives a clear picture of what a confusion matrix is:

.	Actual True	Actual False
Predicted True	TP	TN
Predicted False	FP	FN

TP-Values predicted correctly and belong to first category.
TN-Values predicted correctly but belong to second category.
FP-Values predicted wrong that belong to first category.
FN-Values predicted wrong and belong to second category.

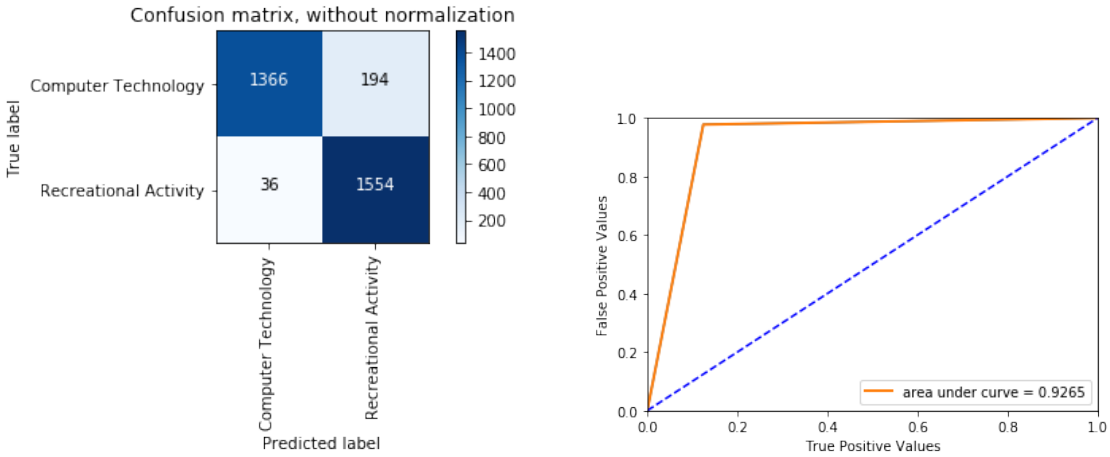
ROC-It is defined as a plot of test sensitivity as the y co-ordinate versus the false positive rate as the x co-ordinate. It is an effective method of evaluating the performance of diagnostic tests.One of the popular measures associated with the ROC curve is the Area Under the Curve(AUC). It is a measure of overall value of the performance of the diagnostic test and is interpreted as the average value of sensitivity for all values of specificity; can range from 0 to 1.

4.1 Question e

Linear Support Vector Machines aids in a great way to handle high dimensional, sparse data set. SVM training algorithm builds a model that assigns new examples to one category or the other in a given set of training examples where each of them are marked as belonging to one or the other two categories.The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering and is used in industrial applications either when data are not labeled or when only some data are labelled as a preprocessing for a classification pass.

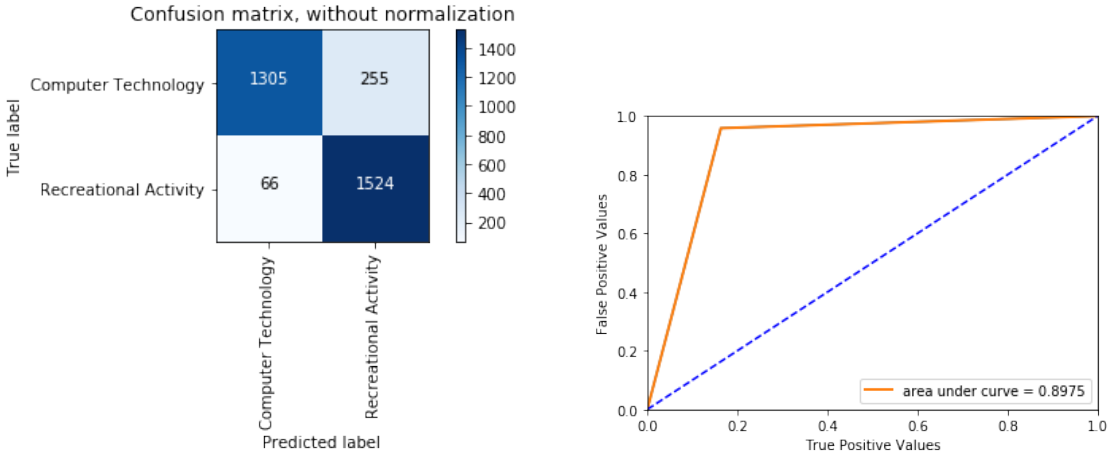
For high dimension problems in text classification, the data are linearly separable. But in some cases they are not and we might need a solution that separates the bulk of data while ignoring weird noise documents.Soft margin may allow some points like noisy examples to be inside or on the wrong side of the margin. We do pay a cost for that misclassification that depends on how far the margin requirement is met. Hard margin on the other hand allows zero errors and cannot be considered an ideal one as any algorithm considered perfect for one data set may fail while applying for another new data set.

The following figures give us the comparison of results obtained through LSI and NMF methods using soft and hard margin SVM:



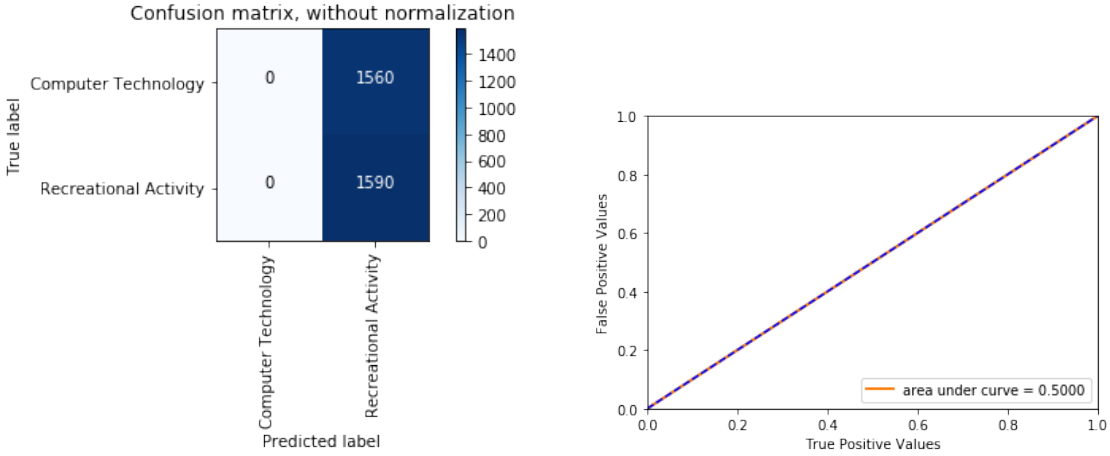
(a) Confusion Matrix (b) ROC Curve

Figure 2: Hard Margin SVM using LSI for min-df=2



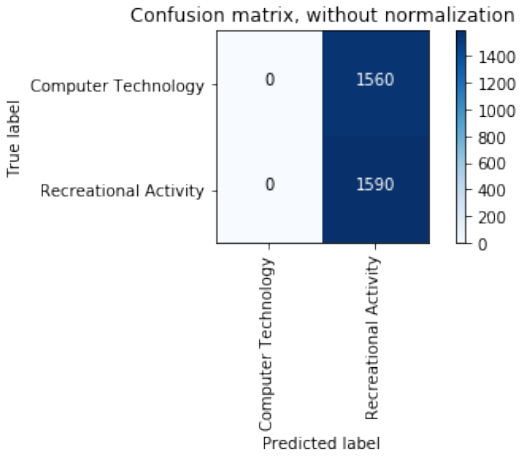
(a) Confusion Matrix (b) ROC Curve

Figure 3: Hard Margin SVM using NMF for min-df=2

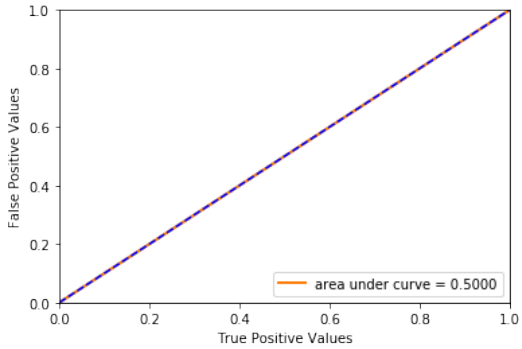


(a) Confusion Matrix (b) ROC Curve

Figure 4: Soft Margin SVM using LSI for min-df=2

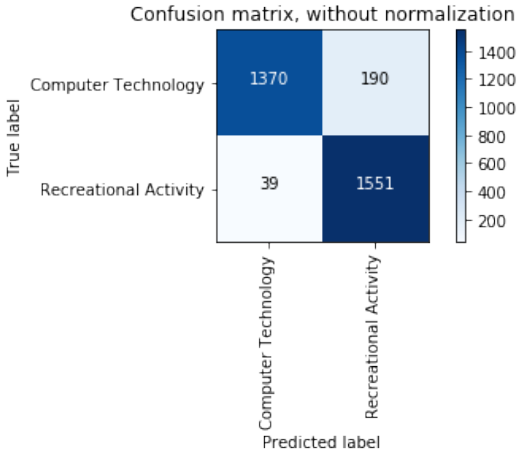


(a) Confusion Matrix

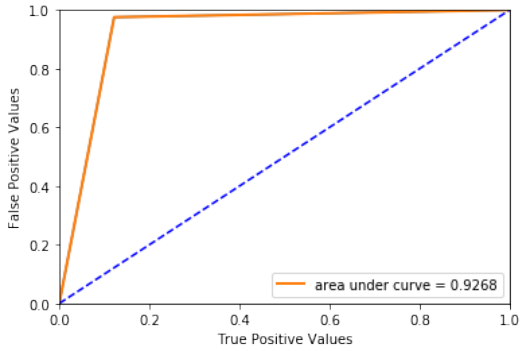


(b) ROC Curve

Figure 5: Soft Margin SVM using NMF for min-df=2

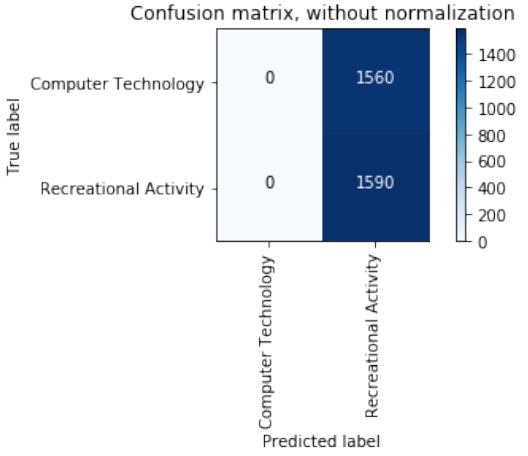


(a) Confusion Matrix

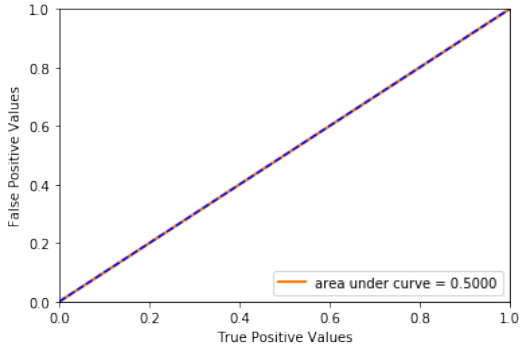


(b) ROC Curve

Figure 6: Hard Margin SVM using LSI for min-df=5



(a) Confusion Matrix



(b) ROC Curve

Figure 7: Soft Margin SVM using LSI for min-df=5

Hard Margin SVM using LSI for min-df=2

Classes	precision	recall	f1-score	support
Computer Technology	0.97	0.88	0.92	1560
Recreational Activity	0.89	0.98	0.93	1590
avg / total	0.93	0.93	0.93	3150

The accuracy of the above test is 92.70%

Hard Margin SVM using NMF for min-df=2

Classes	precision	recall	f1-score	support
Computer Technology	0.95	0.84	0.89	1560
Recreational Activity	0.86	0.96	0.90	1590
avg / total	0.90	0.90	0.90	3150

The accuracy of the above test is 89.81%

Soft Margin SVM using LSI for min-df=2

Classes	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

The accuracy of the above test is 50.48%

Soft Margin SVM using NMF for min-df = 2

Classes	precision	recall	f1-score	support
Computer Technology	0.00	0.00	0.00	1560
Recreational Activity	0.50	1	0.67	1590
avg / total	0.25	0.50	0.34	3150

The accuracy of the above test is 50.48%

Hard Margin SVM using LSI for min-df=5

Classes	precision	recall	f1-score	support
Computer Technology	0.97	0.88	0.92	1560
Recreational Activity	0.89	0.98	0.93	1590
avg / total	0.93	0.93	0.91	3150

The accuracy of the above test is 92.73%

Soft Margin SVM using LSI for min-df=5

Classes	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

The accuracy of the above test is 50.48%

4.2 Question f

Suppose we have a model with one or more unknown parameters, and a training data set to which the model can be fit. The fitting process optimizes the model parameters to make the model fit the training data as well as possible. If we then take an independent sample of validation data from the same population as the testing data, it might turn out that the model does not fit the testing data accurately. This is called over-fitting problem and is likely to occur when the size of training data set is small, or when the number of parameters in the model is large. Cross-validation is a way to fit the model to a hypothetical validation set when an explicit validation set is not available. In 5-fold cross-validation, the original sample is randomly partitioned into 5 equal sized subsamples, out which a single subsample is retained as the testing data and the remaining 4 subsamples are used as training data. The cross-validation process is then repeated 5 times, with each of the 5 subsamples used exactly once as the testing data. The average of the folds is taken to reduce the error and find the best parameters. The figure 8 and 9 above represents the result obtained for SVM cross validation for min-df=2 using LSI and NMF respectively. Figure 10 represents SVM cross validation using LSI for min-df=5.

Best parameters for min-df = 2:
'C': 100, 'gamma': 0.001, 'kernel': 'linear'

Best parameters for min-df = 5:
'C': 1000, 'gamma': 0.001, 'kernel': 'linear'

SVM Cross Validation using LSI for min-df = 2

Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.90	0.93	1560
Recreational Activity	0.91	0.96	0.94	1590
avg / total	0.93	0.93	0.93	3150

The accuracy of the above test is 93.30%

SVM Cross Validation using NMF for min-df = 2

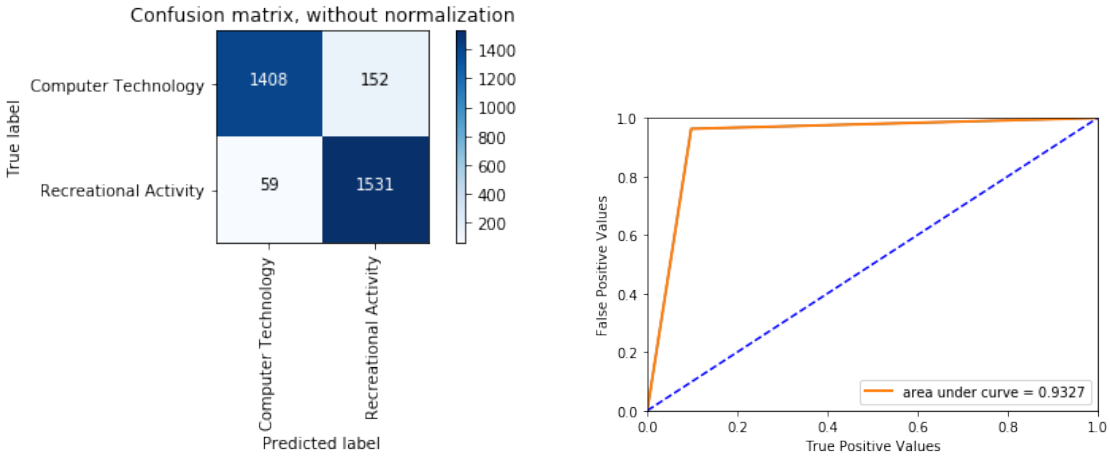
Classes	precision	recall	f1-score	support
Computer Technology	0.95	0.89	0.92	1560
Recreational Activity	0.90	0.95	0.92	1590
avg / total	0.92	0.92	0.92	3150

The accuracy of the above test is 91.97%

SVM Cross Validation using LSI for min-df = 5

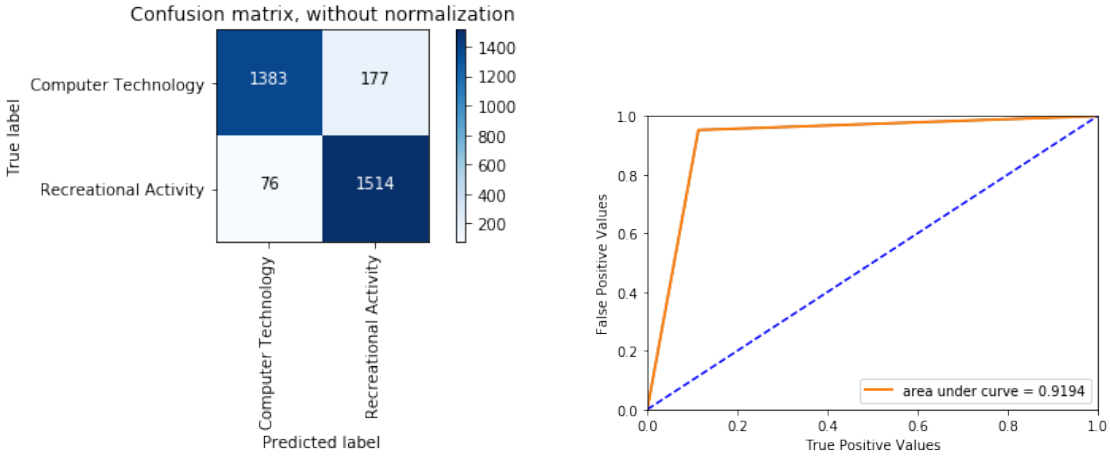
Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.90	0.93	1560
Recreational Activity	0.91	0.96	0.93	1590
avg / total	0.93	0.93	0.93	3150

The accuracy of the above test is 93.20%



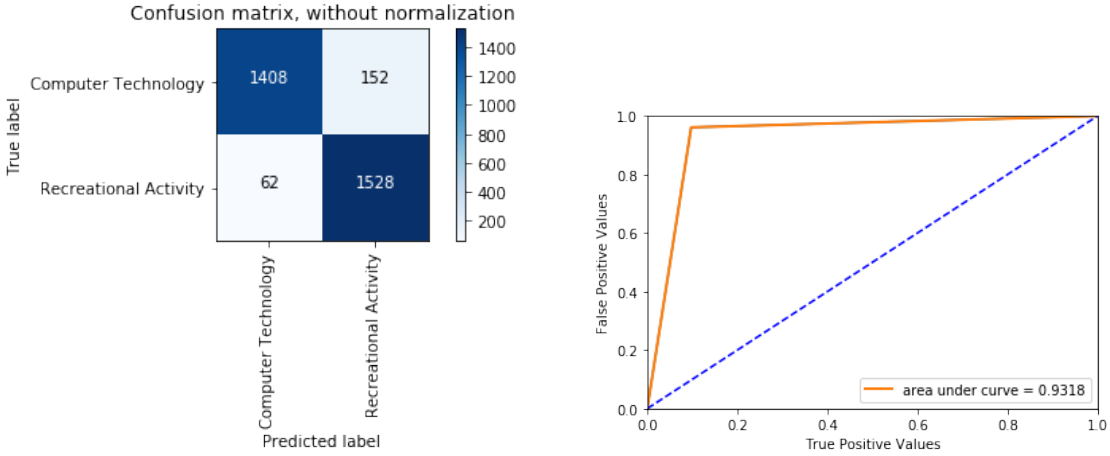
(a) Confusion Matrix (b) ROC Curve

Figure 8: SVM cross-validation using LSI for min-df=2



(a) Confusion Matrix (b) ROC Curve

Figure 9: SVM cross-validation using NMF for min-df=2



(a) Confusion Matrix (b) ROC Curve

Figure 10: SVM cross-validation using LSI for min-df=5

4.3 Question g

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It is not a single algorithm but a family of algorithms that share a common principle, that every feature being classified is independent of every other features. The ROC curve and confusion matrix for Multinomial Naive Bayes using NMF for min-df = 2 is shown in Figure 11.

Multinomial Naive Bayes using NMF:

Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.85	0.90	1560
Recreational Activity	0.87	0.96	0.91	1590
avg / total	0.91	0.91	0.91	3150

The accuracy of the above test is 90.60%

4.4 Question h

In Logistic Regression, the prediction of the output is transformed using a non-linear function called the logistic function. The ROC curve and confusion matrix for Logistic Regression without regularization for min-df = 2 is shown in figure 12 and 13 and min-df = 5 is shown in figure 14. For min-df = 2, the accuracy,precision and recall of logistic regression is given below:

Logistic Regression without regularization using LSI for min-df = 2:

Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.88	0.92	1560
Recreational Activity	0.89	0.97	0.93	1590
avg / total	0.93	0.92	0.92	3150

The accuracy of the above test is 92.25%

Logistic Regression without regularization using NMF for min-df = 2:

Classes	precision	recall	f1-score	support
Computer Technology	0.93	0.85	0.89	1560
Recreational Activity	0.87	0.94	0.90	1590
avg / total	0.90	0.90	0.90	3150

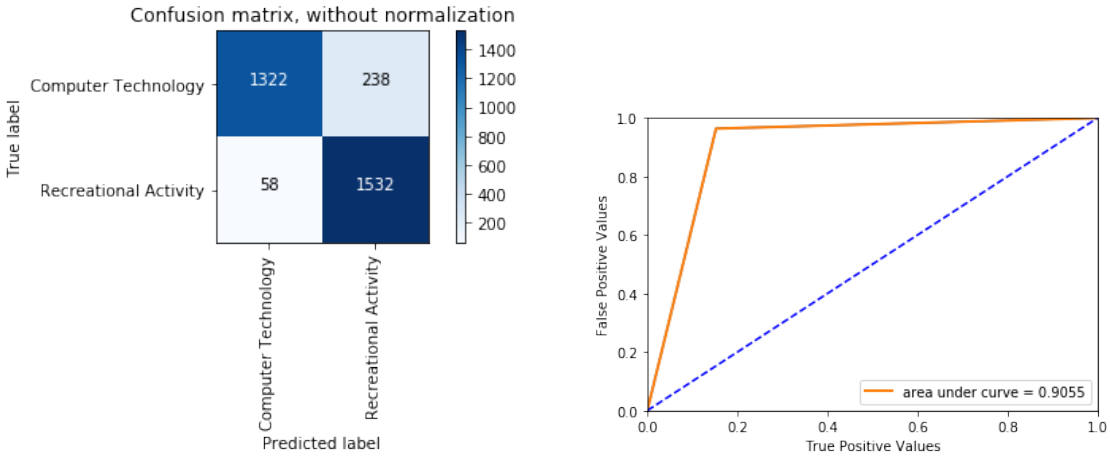
The accuracy of the above test is 89.56%

For min-df = 5, the accuracy, precision and recall of logistic regression is given below:

Logistic Regression without regularization using LSI for min-df = 5:

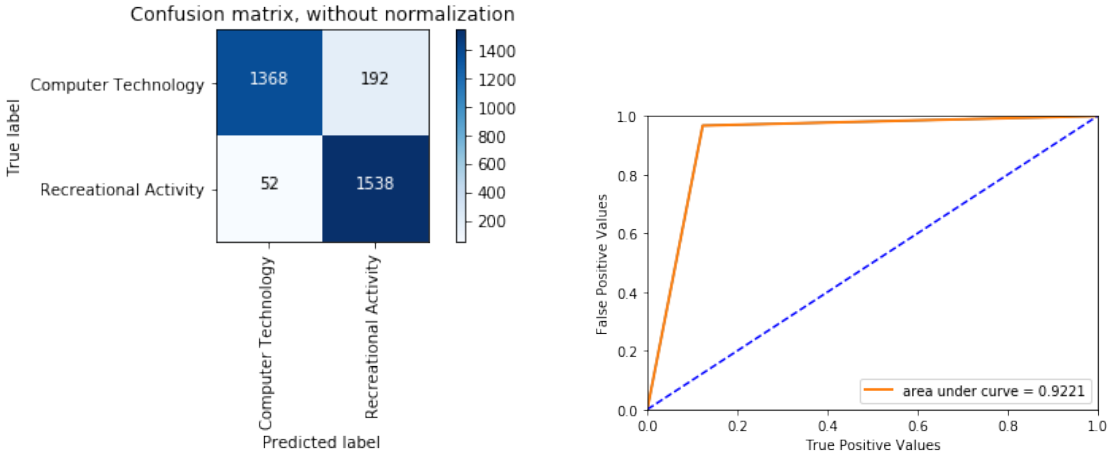
Classes	precision	recall	f1-score	support
Computer Technology	0.97	0.88	0.92	1560
Recreational Activity	0.89	0.97	0.93	1590
avg / total	0.93	0.93	0.93	3150

The accuracy of the above test is 92.54%



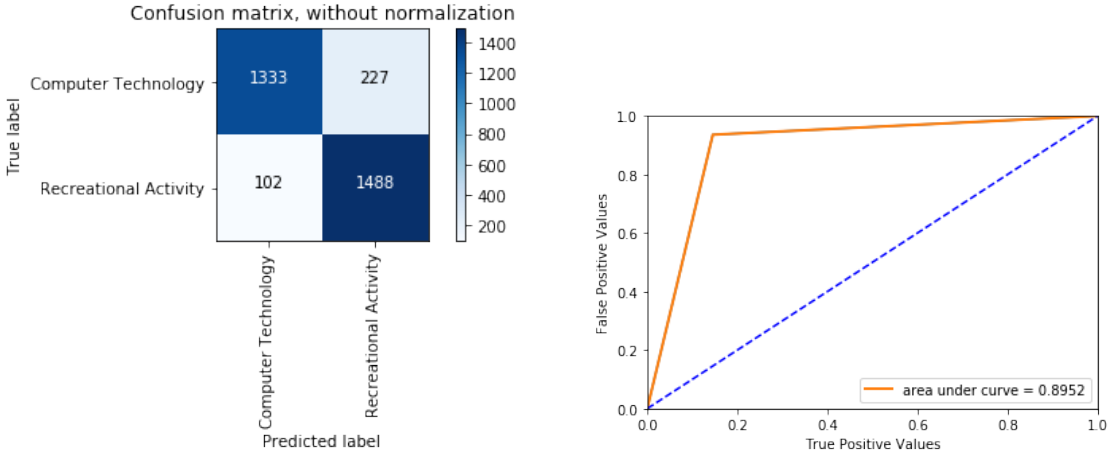
(a) Confusion Matrix (b) ROC Curve

Figure 11: Multinomial Naive Bayes using NMF for min-df = 2



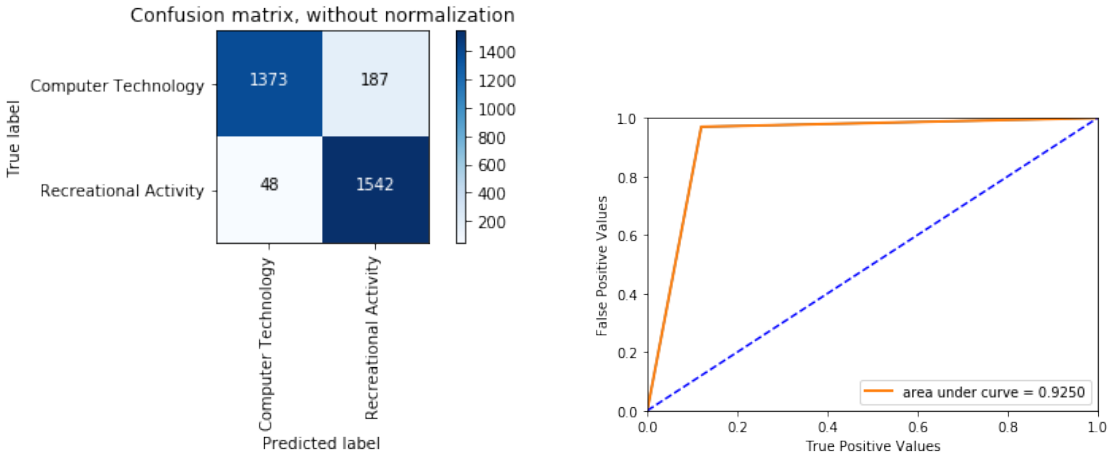
(a) Confusion Matrix (b) ROC Curve

Figure 12: Logistic Regression without regularization using LSI for min-df=2:



(a) Confusion Matrix (b) ROC Curve

Figure 13: Logistic Regression without regularization using NMF for min-df=2:



(a) Confusion Matrix

(b) ROC Curve

Figure 14: Logistic Regression without regularization using LSI for min-df = 5:

4.5 Question i

Regularization is a technique used to solve the overfitting problem in statistical models. It will result in poor prediction and generalization power. Regularization helps to choose preferred model complexity, so that model is better at predicting. There are couple of techniques to achieve regularization viz- L1 and L2. L2 Regularized Logistic Regression require a sample size that grows linearly in the number of irrelevant features. L1 Regularized Logistic Regression requires a sample size that grows logarithmically in the number of irrelevant features.The main difference between L1 and L2 regularization is that L1 can yield sparse models while L2 doesn't and also L2 is the sum of the square of the weights, while L1 is just the sum of the weights. For min-df = 2, the accuracy, precision and recall of logistic regression with regularization is as follows:

Logistic Regression with regularization using L1 norm in LSI for min-df = 2:

Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.91	0.93	1560
Recreational Activity	0.92	0.96	0.94	1590
avg / total	0.94	0.94	0.94	3150

The accuracy of the above test is 93.56%

Logistic Regression with regularization using L1 norm in NMF for min-df = 2:

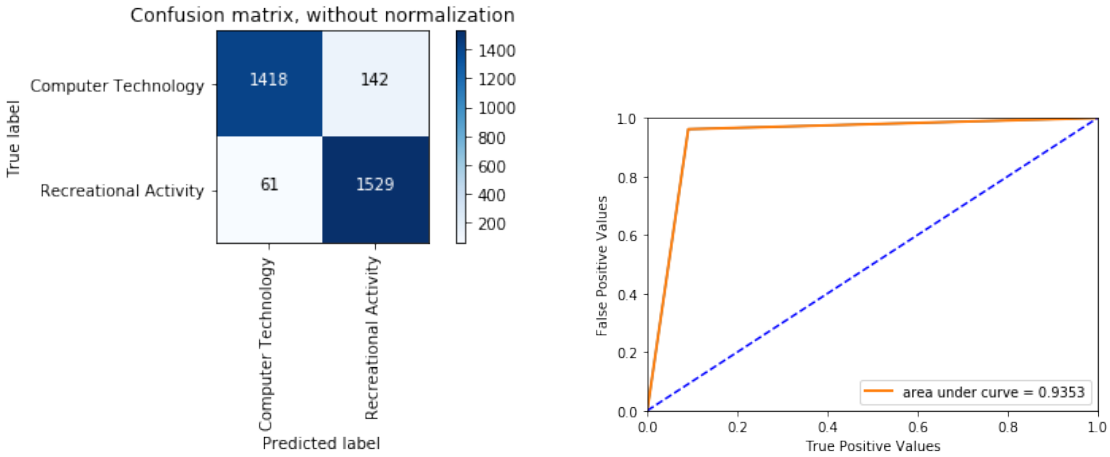
Classes	precision	recall	f1-score	support
Computer Technology	0.95	0.90	0.92	1560
Recreational Activity	0.90	0.95	0.93	1590
avg / total	0.93	0.92	0.92	3150

The accuracy of the above test is 92.41%

Logistic Regression with regularization using L2 norm in LSI for min-df = 2:

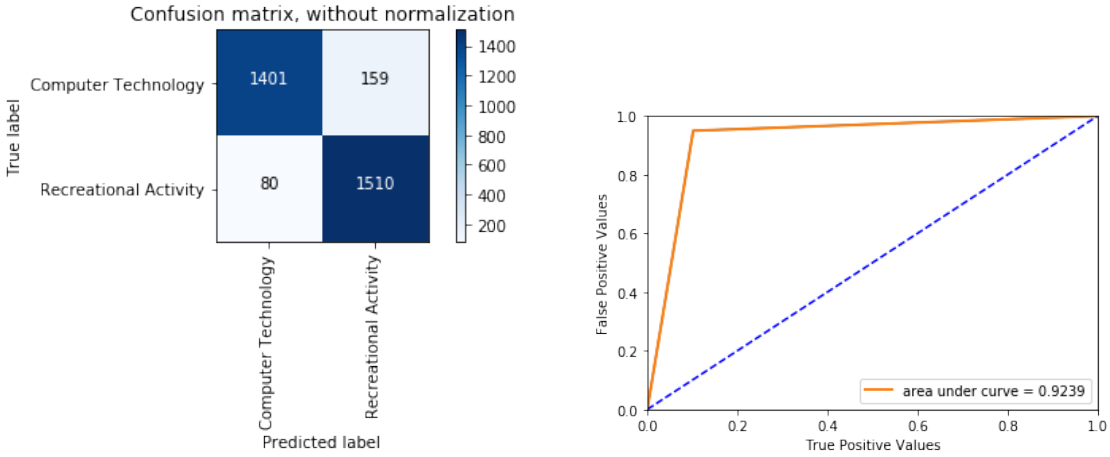
Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.90	0.93	1560
Recreational Activity	0.91	0.96	0.94	1590
avg / total	0.93	0.93	0.93	3150

The accuracy of the above test is 93.30%



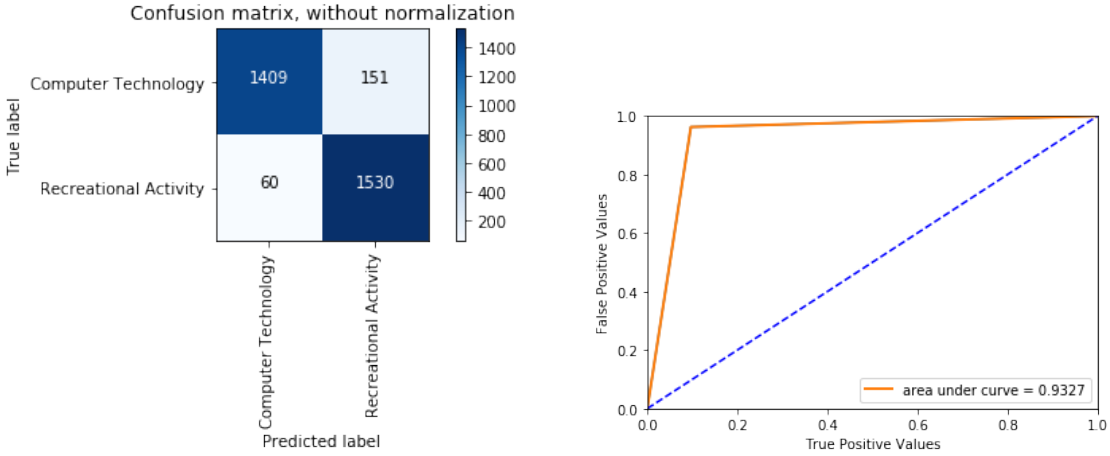
(a) Confusion Matrix (b) ROC Curve

Figure 15: Logistic Regression with regularization using L1 norm in LSI for min-df=2:



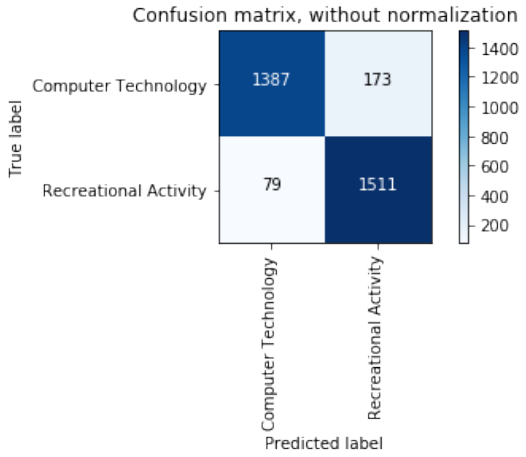
(a) Confusion Matrix (b) ROC Curve

Figure 16: Logistic Regression with regularization using L1 norm in NMF for min-df=2:

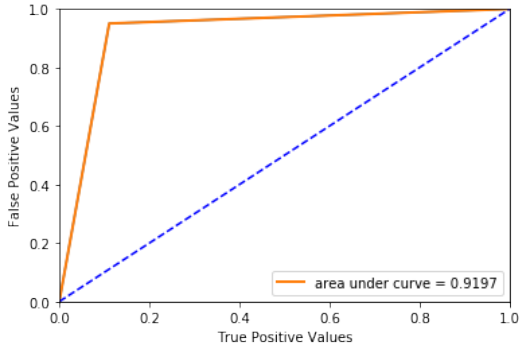


(a) Confusion Matrix (b) ROC Curve

Figure 17: Logistic Regression with regularization using L2 norm in LSI for min-df=2:

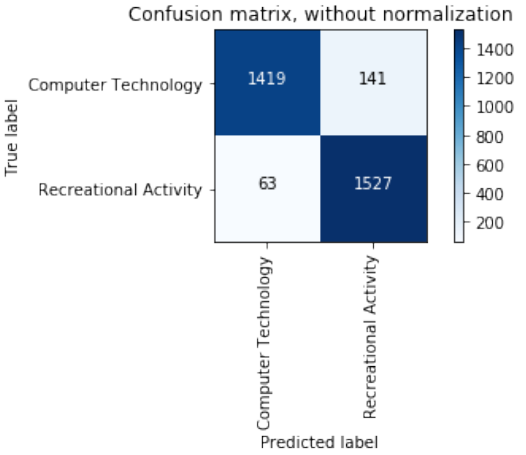


(a) Confusion Matrix

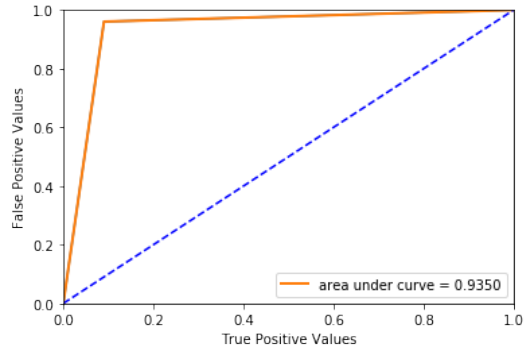


(b) ROC Curve

Figure 18: Logistic Regression with regularization using L2 norm in NMF for min-df = 2:

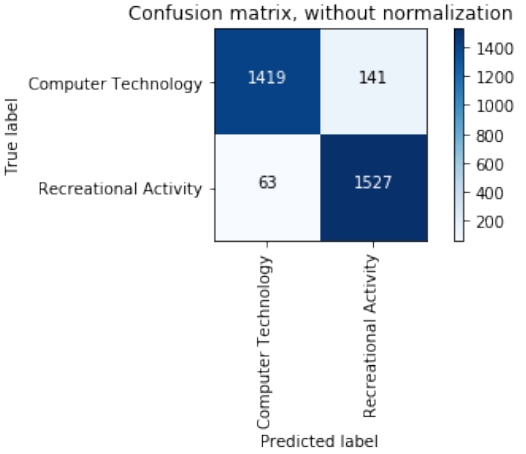


(a) Confusion Matrix

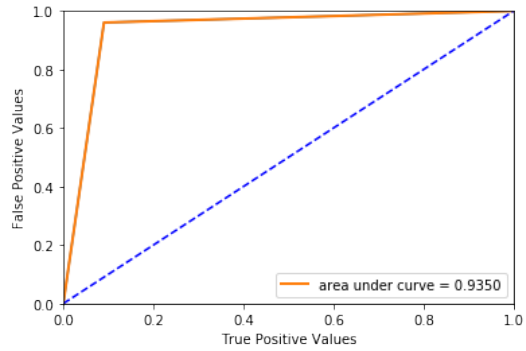


(b) ROC Curve

Figure 19: Logistic Regression with regularization using L1 norm in LSI for min-df = 5:



(a) Confusion Matrix



(b) ROC Curve

Figure 20: Logistic Regression with regularization using L2 norm in LSI for min-df = 5:

Logistic Regression with regularization using L2 norm in NMF for min-df = 2:

Classes	precision	recall	f1-score	support
Computer Technology	0.95	0.89	0.92	1560
Recreational Activity	0.90	0.95	0.92	1590
avg / total	0.92	0.92	0.92	3150

The accuracy of the above test is 92.0%

For min-df = 5, the accuracy, precision and recall of logistic regression with regularization is as follows:

Logistic Regression with regularization using L1 norm in LSI for min-df = 5:

Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.91	0.93	1560
Recreational Activity	0.92	0.96	0.94	1590
avg / total	0.94	0.94	0.94	3150

The accuracy of the above test is 93.52%

Logistic Regression with regularization using L2 norm in LSI for min-df = 5:

Classes	precision	recall	f1-score	support
Computer Technology	0.96	0.91	0.93	1560
Recreational Activity	0.92	0.96	0.94	1590
avg / total	0.94	0.94	0.94	3150

The accuracy of the above test is 93.52%

5 Multiclass Classification

Multiclass classification means a classification task with more than two classes.It makes the assumption that each sample is assigned to one and only one label.The one-vs.-rest strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label. One-vs-One can be applied to any binary classifier to solve multi-class (¿ 2) classification problem. If the multi-class problem has n classes, the One-vs-One ensemble will be composed by n(n-1)/2. The label assigning stage is then performed by majority voting.

5.1 Question j

In this question, we perform naive Bayes and multiclass SVM classification to learn the classifiers in the document that contain the following sub classes: comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, misc.forsale, soc.religion.christian.

Output-Multinomial Classification using Naive Bayes:

Classes	precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.67	0.73	0.70	392
comp.sys.mac.hardware	0.77	0.58	0.66	385
misc.forsale	0.79	0.76	0.77	390
soc.religion.christian	0.81	0.97	0.89	398
avg / total	0.76	0.76	0.76	1565

The accuracy of the above test is 76.17%

Multiclass SVM Classification using MultiNomial Naive Bayes

classes	precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.75	0.73	0.74	392
comp.sys.mac.hardware	0.71	0.78	0.74	385
misc.forsale	0.84	0.81	0.82	390
soc.religion.christian	0.90	0.86	0.88	398
avg / total	0.74	0.73	0.73	1565

The accuracy of the above test is 73.22%

Multiclass Classification using SVM (OnevsOneClassifier) using LSI in min_df=2

Classes	precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.75	0.73	0.74	392
comp.sys.mac.hardware	0.71	0.78	0.74	385
misc.forsale	0.84	0.81	0.82	390
soc.religion.christian	0.97	0.92	0.94	398
avg / total	0.82	0.81	0.81	1565

The accuracy of the above test is 81.21%

Multiclass Classification using SVM (OnevsRestClassifier) using LSI in min_df=2

Classes	precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.77	0.72	0.74	392
comp.sys.mac.hardware	0.76	0.74	0.75	385
misc.forsale	0.76	0.85	0.80	390
soc.religion.christian	0.95	0.94	0.95	398
avg / total	0.81	0.81	0.81	1565

The accuracy of the above test is 81.15%

Multiclass Classification using SVM (OnevsOneClassifier) using NMF in min_df=2

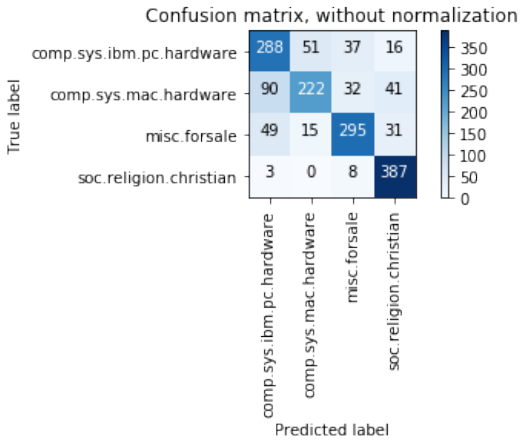
Classes	precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.71	0.65	0.68	392
comp.sys.mac.hardware	0.64	0.76	0.70	385
misc.forsale	0.82	0.77	0.79	390
soc.religion.christian	0.92	0.96	0.94	398
avg / total	0.78	0.78	0.78	1565

The accuracy of the above test is 77.51%

Multiclass Classification using SVM (OnevsRestClassifier) using NMF in min_df=2

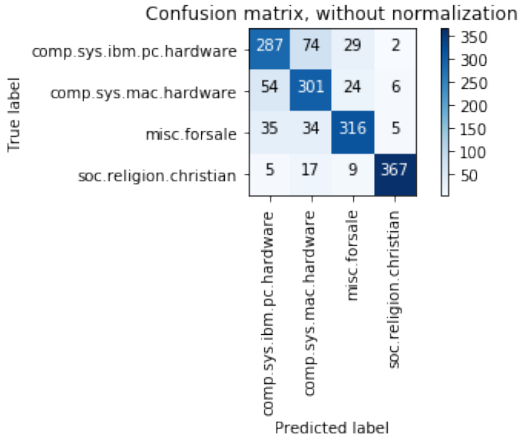
Classes	precision	recall	f1-score	support
comp.sys.ibm.pc.hardware	0.73	0.66	0.70	392
comp.sys.mac.hardware	0.72	0.71	0.72	385
misc.forsale	0.81	0.79	0.80	390
soc.religion.christian	0.86	0.97	0.91	398
avg / total	0.78	0.78	0.78	1565

The accuracy of the above test is 78.40%



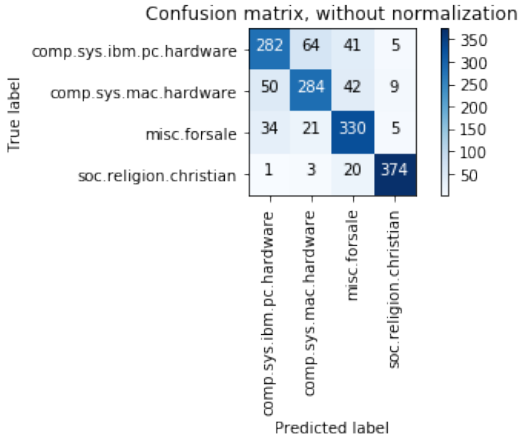
(a) Confusion Matrix

Figure 21: Multiclass Classification using Naive Bayes using NMF in min_df=2



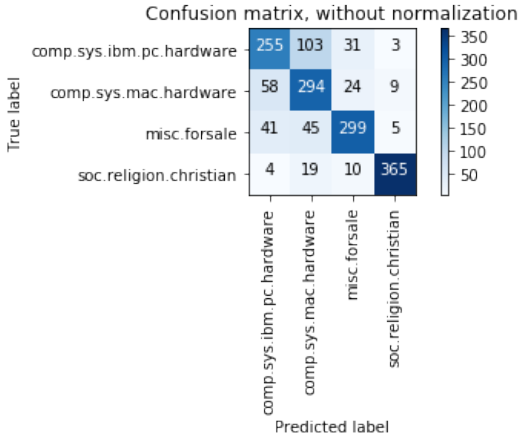
(a) Confusion Matrix

Figure 22: Multiclass Classification using SVM (OnevsOneClassifier) using LSI in min_df=2



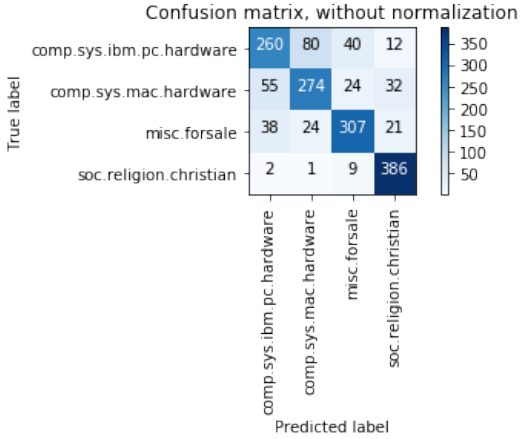
(a) Confusion Matrix

Figure 23: Multiclass Classification using SVM (OnevsRestClassifier) using LSI in min_df=2



(a) Confusion Matrix

Figure 24: Multiclass Classification using SVM (OnevsOneClassifier) using NMF in min_df=2



(a) Confusion Matrix

Figure 25: Multiclass Classification using SVM (OnevsRestClassifier) using NMF in min_df=2