Classification Analysis on Textual Data

Akshya Arunachalam UID : 904943191
Ashwin Kumar Kannan UID : 605035204

January 29, 2018

1 Introduction

The aim of the project is to classify a document into one of the Categories. 20-News Group dataset was used to train and create models based on algorithms such as SVM, Naive Bayes and Logistic Regression. In this report, the techniques used are described and the classification accuracy of each of the models is presented.

2 Dataset

2.1 Question a

The following 8 categories from the 20 News Groups data set are loaded :

- comp.graphics
- comp.os.ms-windows.misc
- comp.sys.ibm.pc.hardware
- comp.sys.mac.hardware
- rec.autos
- rec.motorcycles
- rec.sport.basketball
- rec.sport.hockey

To handle any imbalance in the data set such as the need for down-sample a majority class or change the penalty function, a histogram of the data sets is plotted that shows the number of documents in the classes. The data for the 8 classes that we are interested in is already balanced, hence no data set size modification is required.

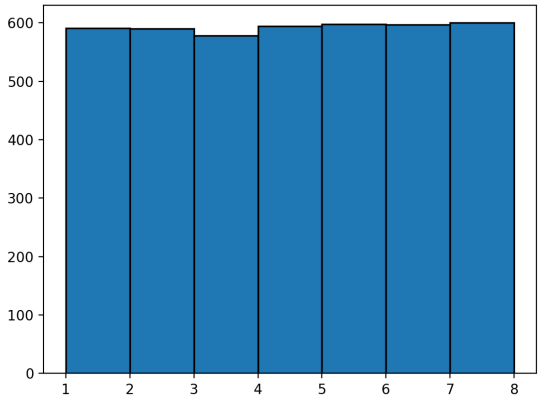

Figure 1: Histogram of number of documents in each class

3 Modeling Text Data and Feature Extraction

3.1 Question b

Proper document representation and the non inclusion of irrelevant words in the document is important for computation accuracy of the algorithm. Hence, common words, stopping words, punctuation and the like are being excluded from the documents. The goal of the question is to tokenize the document, remove stopping words, punctuations and stem words and represent the occurrence of words in the document and the data set. Words that occur in a less that five documents are removed by setting $mindf = 5$. By performing cectorization, we are representing the document in terms of numerical values and occurrence of words in the document irrespective of their position in the document. The data can be represented as a matrix wherein each row represents a document and each column represents a token (word). TF-IDF transformation is done on the data set in order to obtain a vector representation of the document based on the frequency of occurrence of words in the documents.

Dimensions of TF-IDF Matrix (mindf = 2): (4732, 25434)
Dimensions of TF-IDF Matrix (mindf = 5): (4732, 10700)

3.2 Question c

In order to represent the significance of a word in a class, we perform TF-ICF. Itis similar to TF-IDF but it represents the frequency of a token in a class instead of document. After this, the 10 most significant terms in a document, based on their frequency of occurrence, is found. This is found for the following four classes :

- comp.sys.ibm.pc.hardware
- comp.sys.mac.hardware
- misc.forsale
- soc.religion.christian

comp.sys.ibm.pc.hardware	'scsi', 'motherboard', 'eisa', 'simm', 'ati', 'mfm', 'ide', 'interleav', 'dant', 'asynchron
comp.sys.mac.hardware	'simm', 'fpu', 'vram', 'motherboard', 'kth', 'dock', 'scsi', 'solder', 'pentium', '512k'
misc.forsale	'motherboard', 'mpc', 'spider', 'nintendo', 'laptop', 'sunysb', 'antenna', 'vg', 'dang', 'joystick'
soc.religion.christian	'geneva', 'testament', 'theolog', 'infal', 'salvat', 'gentil', 'messiah', 'unto', 'creed', 'guild'

Table 1: The 10 most significant terms in each class

4 Feature Selection

The TF-IDF vector representation of the document may have very high dimensions, and the algorithms may not perform well on them. In this part, the aim is to reduce the dimensions of the vector by Latent Semantic Indexing (LSI). LSI reduces dimension by reducing the SVD of the term document matrix to its best rank k approximation.

4.1 Question d

The dimension of the TF-IDF matrix is reduced to 50 by setting rank parameter k=50. The MinMaxScalar function is used to keep all values in the vector between 0 and 1 since LSI transformation may lead to occurrence of negative values.

Dimensions of TF-IDF vector after LSI: (4732, 50)
Dimensions of TF-IDF vector after NMF: (4732, 50)

5 Learning Algorithms

The aim of this section is to classify documents in the two Categories namely Computer Technology and Recreational Activity. The following parameters are used to measure the accuracy of the classification :

Precision : Precision gives the fraction of items that are correctly placed in a category over the total number of items placed in the category

Recall : Recall gives the fraction of items correctly placed in a category over the total number of items to be placed in the category

F1 Score : F1 score is the measure of accuracy. It is the average of Precision and Recall.

Confusion Matrix : The confusion matrix is a metric that shows the number of true positives, false positives, true negatives and false negatives that allow us to gauge the performance of the algorithm.

The Support Vector Machine Classifier and Naive Bayes Algorithm are tested in the projected and their parameters are checked to predict the optimality of the algorithms.

5.1 Question e

The Support Vector Machine Classifier, in a state space, aims to define a hyperplane such that the points(tokens) can be classified onto either side of the hyperplane depending on their values thus classifying them into different categories. In machine learning, given a training set with tokens each belonging to different categories, the SVM algorithm builds a model and a test set, based on this model, is classified into one of the categories. In NMF the matrix is factorised into two matrices with all positive values.

Hard and Soft Margin SVM:
Incase of large data sets, the documents can be usually classified into categories, but that is not the case always. Soft margin SVC's define a margin that allows errors in classifying, whereas in Hard Margin SVCs, no errors are allowed and documents need to be strictly classified. A parameter (gamma) is defined that is set high for Hard Margin SVCs and low for Soft Margin SVCs.

The algorithm works by reducing the multi class data set into binary classification. The model is trained and the accuracy parameters for test set are measured for both Hard Margin and Soft Margin SVCs.

With $mindf = 2$, below are the ROC curves and values for Accuracy, Precision, Recall and Confusion Matrices that we get as the output.

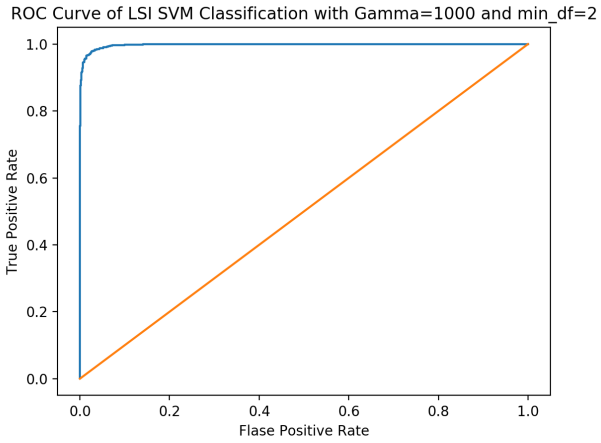

Figure 2: ROC Curve of LSI Hard Margin SVM Classification With mindf=2

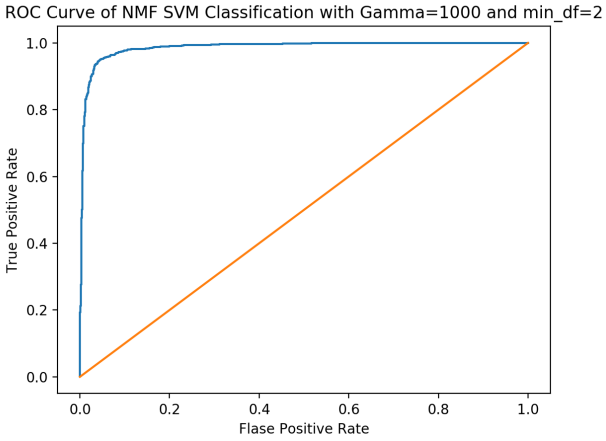

Figure 3: ROC Curve of NMF Hard Margin SVM Classification With mindf=2

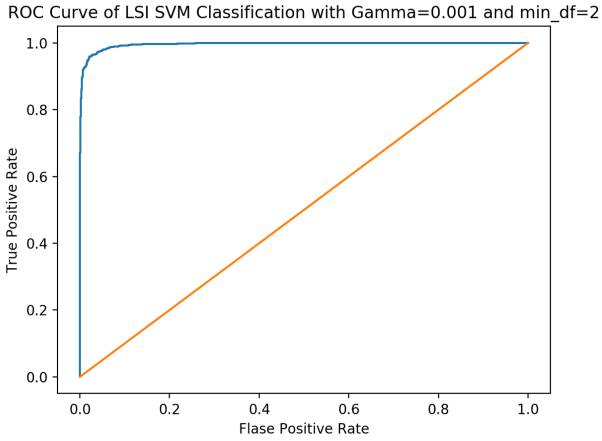

Figure 4: ROC Curve of LSI Soft Margin SVM Classification With mindf=2

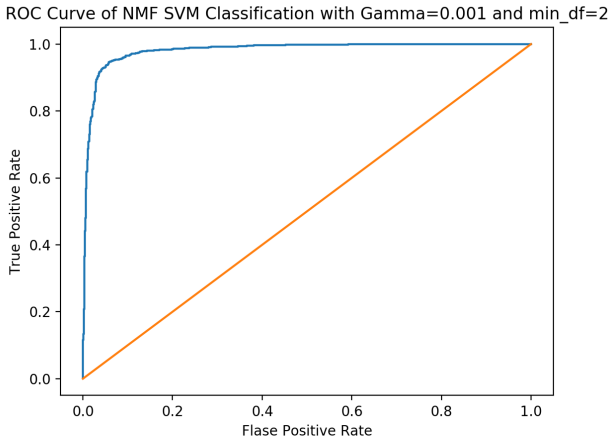

Figure 5: ROC Curve of NMF Soft Margin SVM Classification With mindf=2

Model	Accuracy	Precision	Recall
LSI	0.974285714286	0.974424733466	0.97421988389
NMF	0.948888888889	0.949565278858	0.948717948718

Table 2: Statistics for Hard Margin SVM and NMF Classification with mindf=2

1560	30
51	1509

Table 3: Confusion matrix for Hard Margin SVM and LSI Classification with mindf=2

1537	53
108	1452

Table 4: Confusion matrix for Hard Margin SVM and NMF Classification with mindf=2

Model	Accuracy	Precision	Recall
LSI	0.504761904762	0.252380952381	0.5
NMF	0.504761904762	0.252380952381	0.5

Table 5: Statistics for Soft Margin SVM and NMF Classification with mindf=2

1590	0
1560	0

Table 6: Confusion matrix for Soft Margin SVM and LSI Classification with mindf=2

1590	0
1560	0

Table 7: Confusion matrix for Soft Margin SVM and NMF Classification with mindf=2

With *mindf* = 5, below are the ROC curves and values for Accuracy, Precision, Recall and Confusion Matrices that we get as the output.

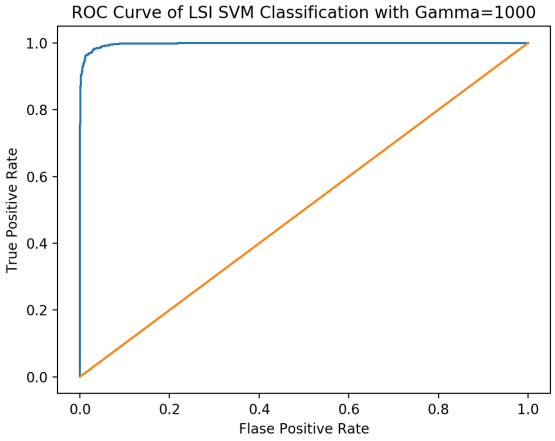

Figure 6: ROC Curve of LSI Hard Margin SVM Classification With mindf=5

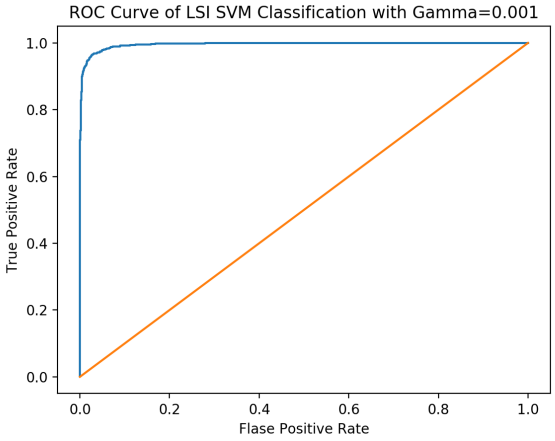

Figure 7: ROC Curve of LSI Soft Margin SVM Classification With mindf=5

Model	Accuracy	Precision	Recall
LSI	0.973650793651	0.973811714433	0.973578858249

Table 8: Statistics for Hard Margin SVM and LSI Classification with mindf=5

1560	30
53	1507

Table 9: Confusion matrix for Hard Margin SVM and LSI Classification with mindf=5

Model	Accuracy	Precision	Recall
LSI	0.504761904762	0.252380952381	0.5

Table 10: Statistics for Soft Margin SVM and LSI Classification with mindf=5

1590	0
1560	0

Table 11: Confusion matrix for Soft Margin SVM and LSI Classification with mindf=5

5.2 Question f

In 5-fold cross validation, the data set is divided into 5 parts and 4 parts are used as training set and the the 5th part is used as the test data. This procedure is repeated 5 times and each time the test data is different. This is done to find the best (gamma) value for SVCs and we find that the best gamma value with the $mindf = 2$ is 1000(LSI) and 1000(NMF). With $mindf = 5$, the best gamma value is 10(LSI). Below are the results including the confusion matrix for this part.

Gamma value	Model	Accuracy	Precision	Recall
1000	LSI	0.978812484141	0.978830282673	0.97879460944
1000	NMF	0.974752600863	0.974796863786	0.974719667442

Table 12: Statistics for LSI and NMF 5-cross validation with mindf=2

Gamma value	Model	Accuracy	Precision	Recall
10	LSI	0.978304998731	0.978344976996	0.978274842487

Table 13: Statistics for LSI 5-cross validation with mindf=5

3902	77
90	3813

Table 14: Confusion matrix for LSI With Gamma=1000 and mindf=2

3892	87
112	3791

Table 15: Confusion matrix for NMF With Gamma=1000 and mindf=2

3905	74
97	3806

Table 16: Confusion matrix for LSI With Gamma=10 and mindf=5

5.3 Question g

In this section, the Naive Bayes algorithm is used to classify documents into different categories. The Naive Bayes algorithm calculates the conditional probability that a document contains certain words given it is in a particular category from the training set and using that it calculates the conditional probability that a particular document belongs in a category in the test set.

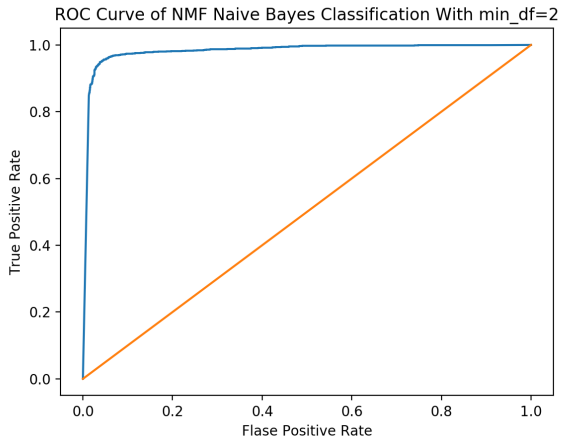

Figure 8: ROC Curve of NMF Naive Bayes Classification With mindf=2

Model	Accuracy	Precision	Recall
NMF	0.953650793651	0.953859082063	0.953859082063

Table 17: Statistics for Naive Bayes NMF With mindf=2

1531	59
87	1473

Table 18: Confusion matrix for NMF With mindf=2

5.4 Question h

Logistic Regression Classifier uses calculated logits to predict target category. The logistic model tries and understands the relationship between categorically dependent variables and independent variables. The logistic regression model will pass the likelihood occurrences through the logistic function to predict the corresponding target class.

With $mindf = 2$, below are the results for LSI and NMF that we received.

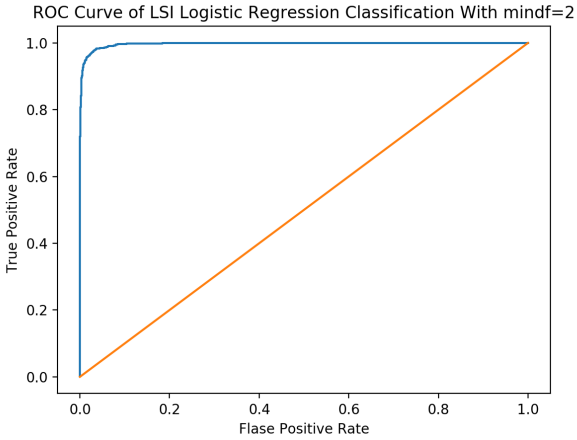

Figure 9: ROC Curve of LSI Logistic Regression Classifier With mindf=2

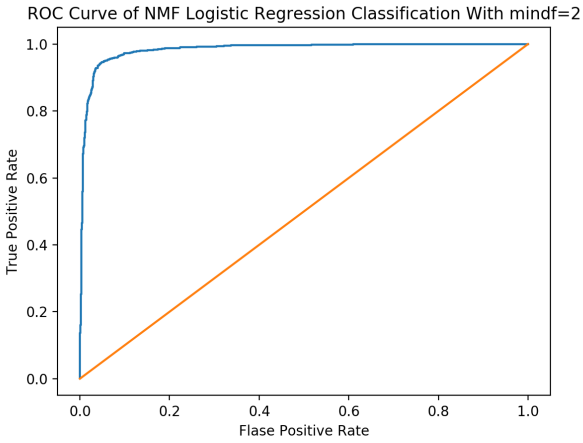

Figure 10: ROC Curve of NMF Logistic Regression Classifier With mindf=2

Model	Accuracy	Precision	Recall
LSI	0.970793650794	0.970988856545	0.970712385099
NMF	0.944761904762	0.945283882784	0.944611756168

Table 19: Statistics for Logistic Regression Classifier with mindf=2

1557	33
59	1501

Table 20: Confusion matrix for LSI with mindf=2

1527	63
111	144

Table 21: Confusion matrix for NMF with mindf=2

With *mindf* = 5, below are the results for LSI that we received.

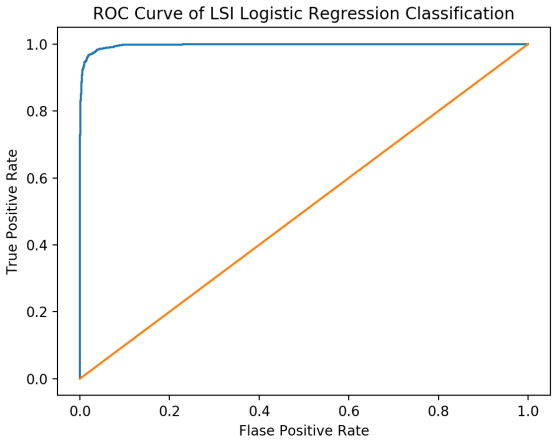

Figure 11: ROC Curve of LSI Logistic Regression Classifier With mindf=5

Model	Accuracy	Precision	Recall
LSI	0.972063492063	0.972312999274	0.971970246734

Table 22: Statistics for Logistic Regression Classifier with mindf=5

1561	29
59	1501

Table 23: Confusion matrix for LSI with mindf=5

5.5 Question i

Regularization in Logistic Regression Classifier is done to deal with the problem of overfitting that occurs when trying to fit a high polynomial function. Overfitting occurs when you train a model too much. Such a model works well for the training data but when a test data is introduced, the accuracy is poor. Regularisation reduces the value of θ and λ is the regularisation parameter.

L1 and L2 regularization are techniques to reduce model overfitting. L1 weight regularization penalizes weight values by adding the sum of their absolute values to the error term. L2 weight regularization penalizes weight values by adding the sum of their squared values to the error term. L1 regularisation gives better solution in sparse feature sets that L2.

Below are the results for *mindf* = 2 run with LSI and NMF

Regularization parameter	L1	L2
−7	0.4952	0.4784
−6	0.4952	0.4781
−5	0.4952	0.4771
−4	0.4952	0.4701
−3	0.4953	0.32
−2	0.0904	0.0555
−1	0.0403	0.0301
0	0.0258	0.02921
1	0.0219	0.0241
2	0.0222	0.0229
3	0.0222	0.0222
4	0.0222	0.0222
5	0.0222	0.0222
6	0.0222	0.0222
7	0.0222	0.0222

Table 24: Regularization parameter and Testing error for L1 and L2 (LSI) With mindf=2

Regularization parameter	L1	L2
−7	0.4952	0.4952
−6	0.4952	0.4952
−5	0.4952	0.4952
−4	0.4952	0.4952
−3	0.4952	0.4952
−2	0.4952	0.4914
−1	0.4952	0.1463
0	0.0473	0.0552
1	0.0298	0.0409
2	0.0260	0.0352
3	0.0260	0.0273
4	0.0260	0.0257
5	0.0260	0.0260
6	0.0260	0.0260
7	0.0260	0.0260

Table 25: Regularization parameter and Testing error for L1 and L2 (NMF) With mindf=2

From the testing error table we can infer that for low values of regularization parameter, excessive fitting takes place and hence the error is more. However

as the value of the parameter is increased, the error reduces and begins to increase again after a certain point.

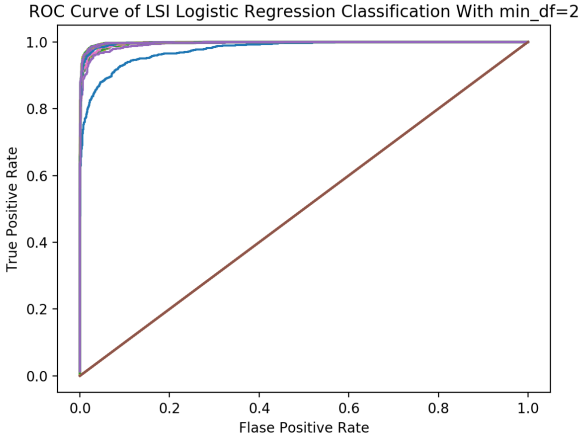

Figure 12: ROC Curve of LSI Logical Regression Classifier

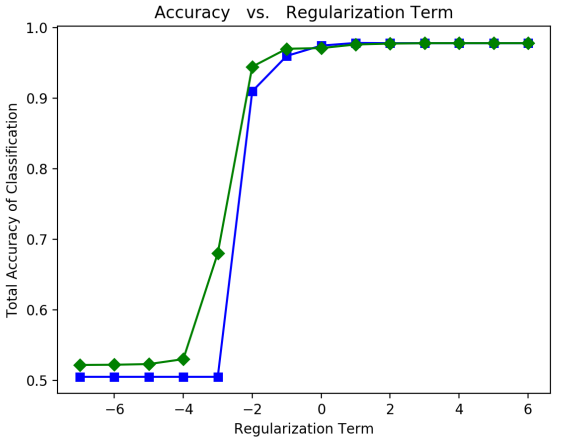

Figure 13: Accuracy vs Regularization for LSI

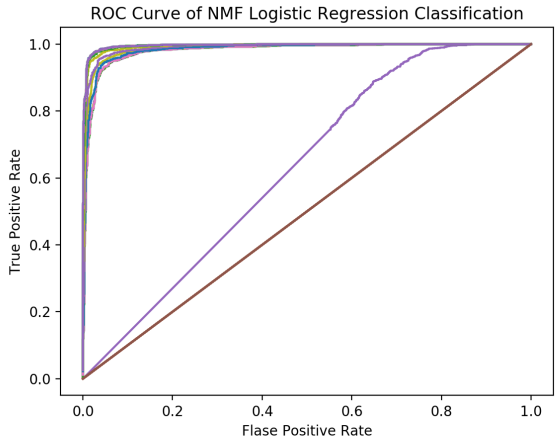

Figure 14: ROC Curve of NMF Logical Regression Classifier

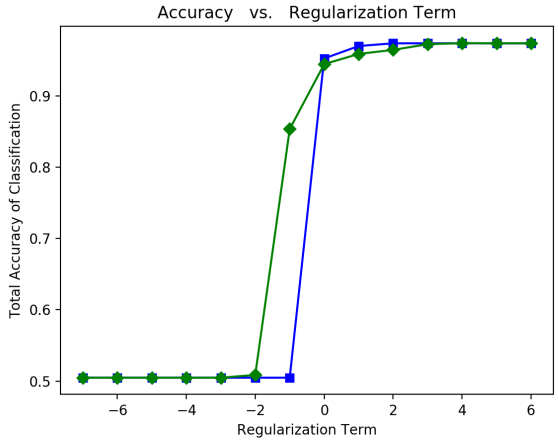

Figure 15: Accuracy vs Regularization for NMF

Below are the results for $mindf = 5$ run with LSI.

Regularization parameter	L1	L2
−7	0.4952	0.4654
−6	0.4952	0.4648
−5	0.4952	0.4644
−4	0.4952	0.4505
−3	0.4953	0.2768
−2	0.08	0.0537
−1	0.0467	0.03239
0	0.0258	0.02794
1	0.0213	0.0235
2	0.0197	0.0213
3	0.02	0.0207
4	0.0203	0.02
5	0.0203	0.0203
6	0.0203	0.0203
7	0.0203	0.0203

Table 26: Regularization parameter and Testing error for L1 and L2 (LSI) With mindf=5

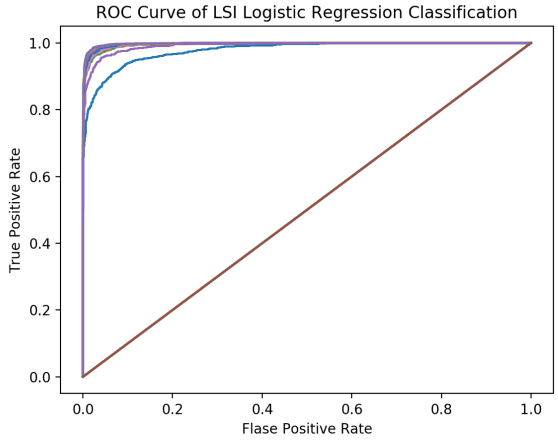

Figure 16: ROC Curve of LSI Logical Regression Classifier

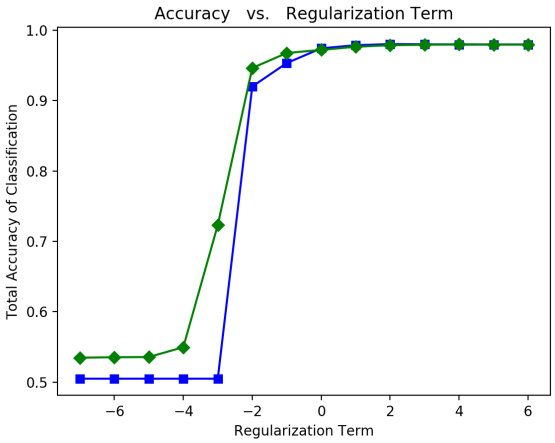

Figure 17: Accuracy vs Regularization for LSI

6 Multiclass Classifier

6.1 Question j

In this section, the aim is to classify documents into multiple classes rather than the two classes as done previously. Multiclass classification is done using Naive Bayes and SVM Classifier. The aim is to classify documents into the following categories :

- comp.sys.ibm.pc.hardware
- comp.sys.mac.hardware
- misc.forsale
- soc.religion.christian

Below are the results for the Naive Bayes Multiclass Classification with $mindf = 2$

Technique	Accuracy	Precision	Recall
OneVsOnve	0.778274760383	0.783778642807	0.776817730905
OneVsRest	0.780191693291	0.782107365276	0.778799874748

Table 27: Statistics for Naive Bayes Multiclass Classification With mindf=2

309	48	27	8
97	255	28	5
74	42	261	13
1	0	4	393

Table 28: Confusion matrix for one vs one Naive Bayes With mindf=2

287	62	39	4
92	255	36	2
69	28	285	8
0	1	3	394

Table 29: Confusion matrix for one vs rest Naive Bayes with mindf=2

Below are the results for the SVM Multiclass Classification with $mindf = 5$ for both LSI and NMF.

Technique	Accuracy	Precision	Recall
OneVsOnve	0.883067092652	0.885409440673	0.882348815936
OneVsRest	0.88945686901	0.888934936819	0.888707823942

Table 30: Statistics for LSI SVM Multiclass Classification With mindf=5

346	29	17	0
52	304	28	1
26	15	348	1
10	2	2	384

Table 31: Confusion matrix for one vs one LSI SVM With mindf=5

335	33	22	2
41	309	34	1
20	14	354	2
1	2	1	394

Table 32: Confusion matrix for one vs rest LSI SVM with mindf=5

Technique	Accuracy	Precision	Recall
OneVsOnve	0.721405750799	0.783352677314	0.720416465862
OneVsRest	0.846645367412	0.844780503631	0.845709947805

Table 33: Statistics for NMF SVM Multiclass Classification With mindf=5

259	8	125	0
76	202	105	2
28	14	347	1
0	1	76	321

Table 34: Confusion matrix for one vs one NMF SVM With mindf=5

313	34	31	14
53	296	24	12
32	26	321	11
0	1	2	395

Table 35: Confusion matrix for one vs rest NMF SVM With mindf=5