EE219: Large-Scale Data Mining: Models and Algorithms
Winter 2018

Project 1: Classification Analysis on Textual Data

Zhi Ming CHUA (UID: 805-068-401)
Vijay RAVI (UID: 805-033-666)

January 29, 2018

Contents

1 Introduction

Statistical classification refers to the task of identifying a category, from a predefined set, to which a data point belongs, given a training data set with known category memberships. Classification differs from the task of clustering, which concerns grouping data points with no predefined category memberships, where the objective is to seek inherent structures in data with respect to suitable measures. Classification turns out as an essential element of data analysis, especially when dealing with a large amount of data. In this project, we look into different methods for classifying textual data.

2 Dataset and Problem Statement

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

The objective of this project is to train a classifier to group the documents into two classes: Computer Technology and Recreational activity. These two classes include the sub-classes as shown in table 1.

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

Table 1: Sub-classes of Computer Technology and Recreational activity

2.1 Balancing Training Dataset

A balanced training dataset ensures that the trained model does not have a significantly better performance in one class of documents over another. The number of training documents per class is counted and a histogram is plotted in Figure 1.

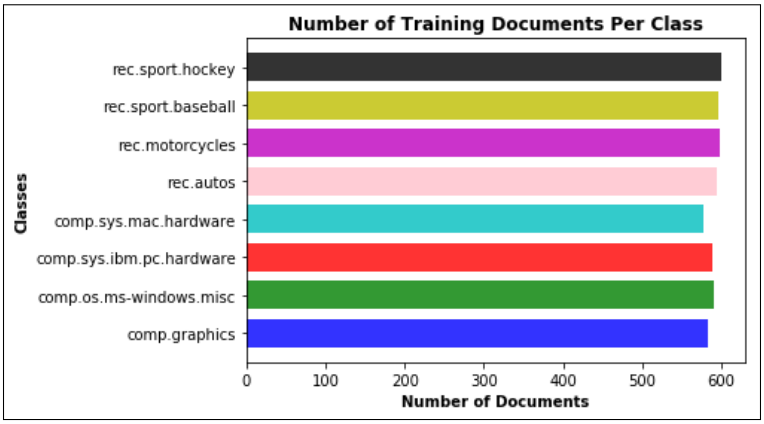


Figure 1: Histogram

It can be seen that the number of documents for each class are approximately equal, hence the dataset does not need to be balanced in this case. For the subclasses as shown in table 1, the combined training dataset has 4732 documents and the testing dataset has 3150 documents.

3 Modeling Text Data and Feature Extraction

3.1 TFIDF

Each document is tokenized into words, with punctuations and numbers removed, and stop words excluded. Words were also stemmed to their respective root words.

The `max_df` parameter for the `CountVectorizer` function removes words that are commonly used. It is set just below 1.0 at 0.99 to remove terms that appear in more than 99% of documents. The `min_df` parameter helps to filter out words that rarely appear in the dataset. It is varied from 1 to 5, with the respective number of terms extracted plotted in Figure 2.

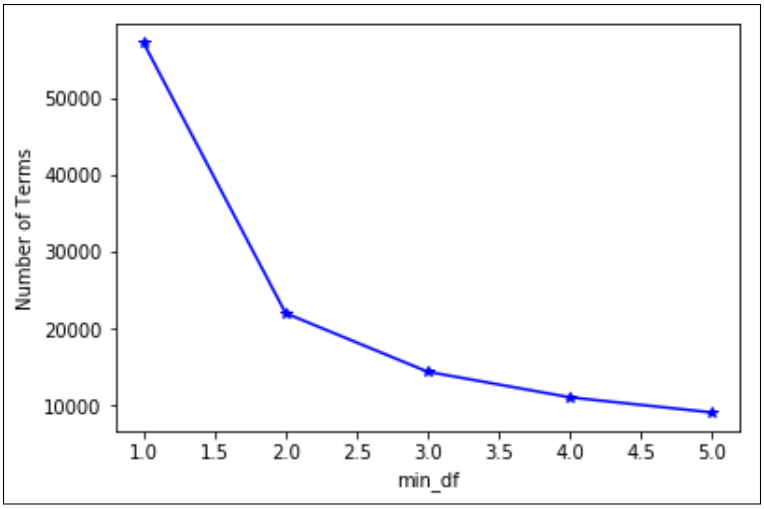


Figure 2: Number of Terms for Varying `min_df`

For `min_df=2` and `min_df=5`, the number of terms extracted are 22,048 and 9142 respectively. In the following sections, the performance of classifiers using these two datasets will be evaluated and compared.

3.2 TFICF

The top 10 significant terms for classes 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'misc.forsale' and 'soc.religion.christian' were obtained using a similar method as TFIDF, with documents in each class concatenated into a single document. The terms are tabulated in Section 3.2.

comp.sys.ibm.pc.hardware	**comp.sys.mac.hardware**	**misc.forsale**	**soc.religion.christian**
scsi	mac	sale	christian
ide	quadra	dos	jesus
mb	simm	ship	church
disk	scsi	forsal	christ
motherboard	centri	wolverin	bibl
floppi	duo	disk	faith
irq	nubus	manual	scriptur
bus	mb	hiram	sin
dos	monitor	obo	cathol
jumper	lc	packag	truth

Table 2: 10 Most Significant Terms

4 Feature Selection

4.1 Latent Semantic Indexing (LSI)

LSI is a dimensionality reduction transform method that minimizes the residual mean square between the original dataset and the dimension-reduced dataset. The LSI representation is obtained by calculating left and right singular vectors corresponding to the largest values of the TFIDF matrix.

This was carried out using `scipy.sparse.linalg` and the dataset was reduced to a 4732 by 50 matrix.

4.2 Non-Negative Matrix Factorization (NMF)

NMF minimizes the Frobenius norm between the original matrix ($\mathbf{V}$) and the approximated form, which is a product of two matrices with non-negative elements ($\mathbf{V} \approx \mathbf{WH}$). NMF clusters the columns of the input data, which is used here to compress the original dataset to a 4732 by 50 matrix ($\mathbf{W}$).

5 Learning Algorithms

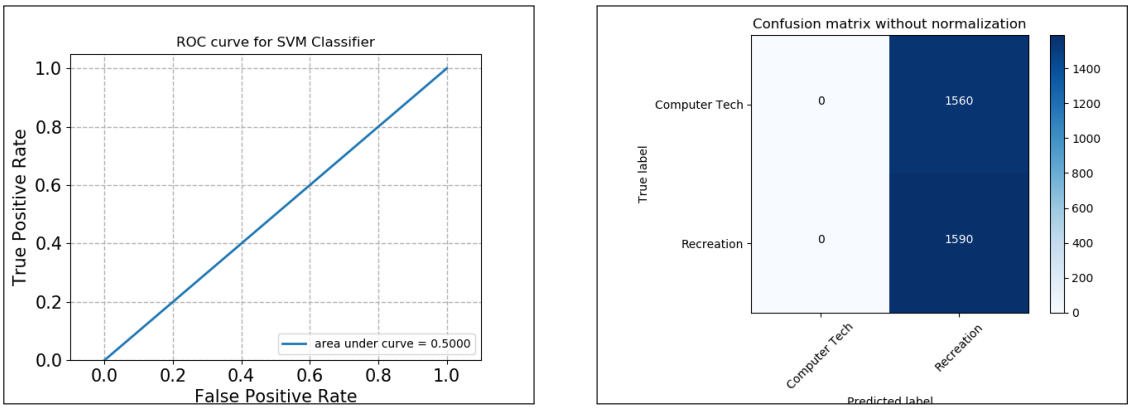
5.1 Support Vector Machine (SVM)

In this section, SVM is used to classify the documents into two classes - Computer Technology and Recreational Activity.

A soft margin allows some misclassification of data points as long as most of them are still well-separated, whereas a hard margin heavily penalizes misclassification. For linearly separable data points, a hard margin implementation may be more suitable. However, if they are not, overfitting may occur, leading to poor testing accuracy.

The performance of soft and hard margin SVMs on this dataset is investigated using penalty $\gamma = 0.001, 1, 1000$.

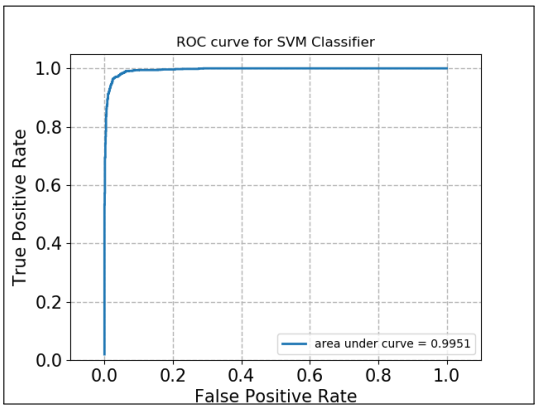
5.1.1 LSI: `min_df=2`



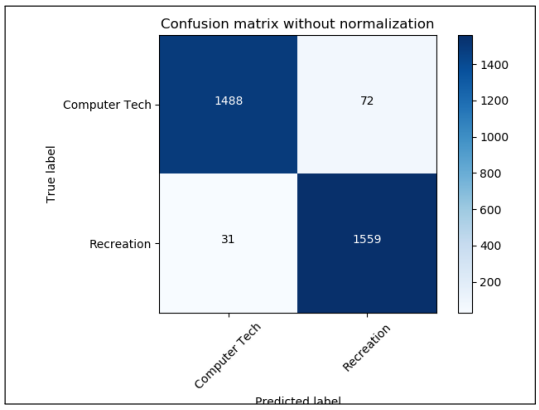
(a) ROC Curve

(b) Confusion Matrix

Figure 3: $\gamma = 0.001$

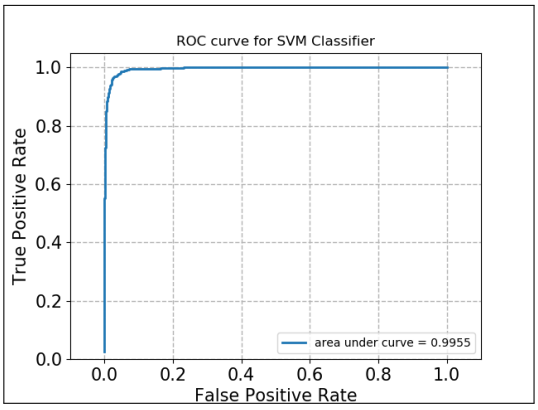


(a) ROC Curve

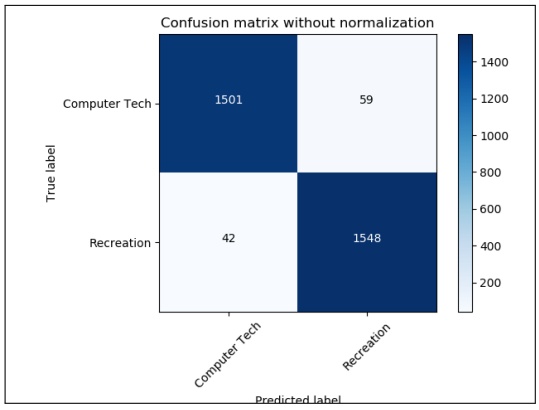


(b) Confusion Matrix

Figure 4: $\gamma = 1$



(a) ROC Curve



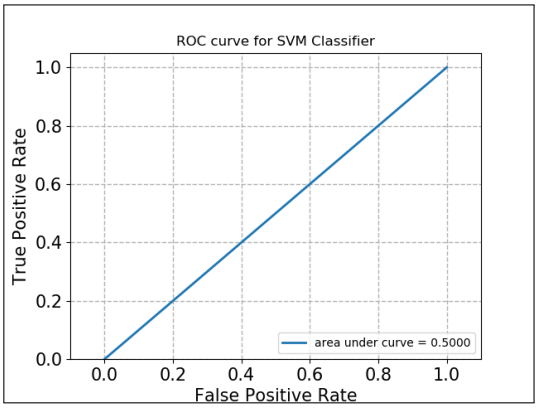
(b) Confusion Matrix

Figure 5: $\gamma = 1000$

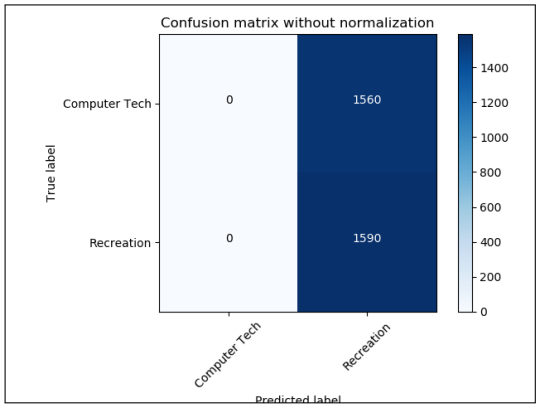
γ	**0.001**	**1**	**1000**
Accuracy	50.48	96.73	96.79
Precision	25.24	96.77	96.80
Recall	50.00	96.72	96.79

Table 3: Accuracy, Precision and Recall

5.1.2 LSI: `min_df=5`

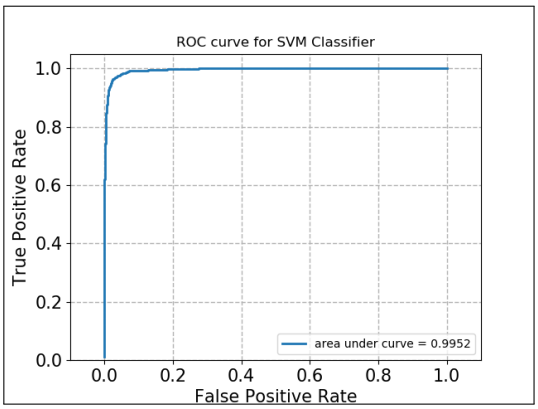


(a) ROC Curve

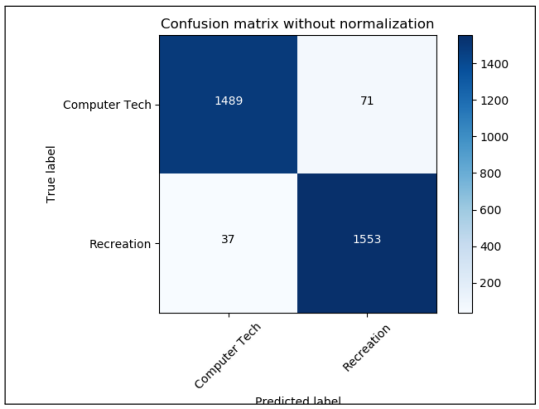


(b) Confusion Matrix

Figure 6: $\gamma = 0.001$

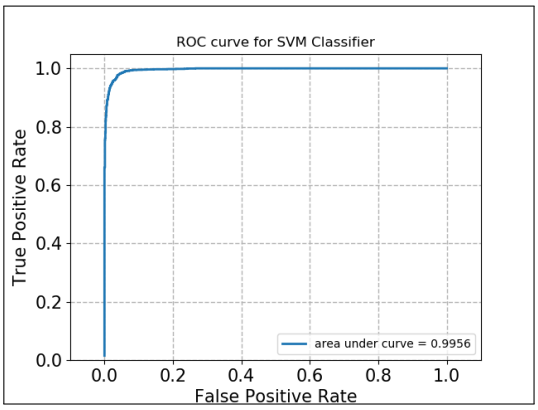


(a) ROC Curve

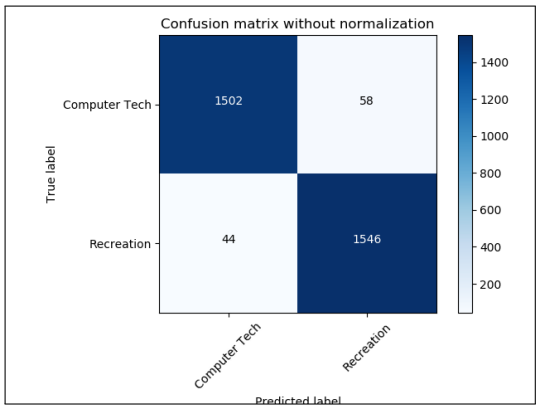


(b) Confusion Matrix

Figure 7: $\gamma = 1$



(a) ROC Curve



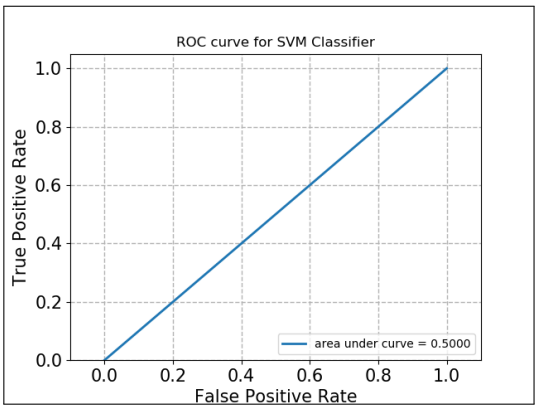
(b) Confusion Matrix

Figure 8: $\gamma = 1000$

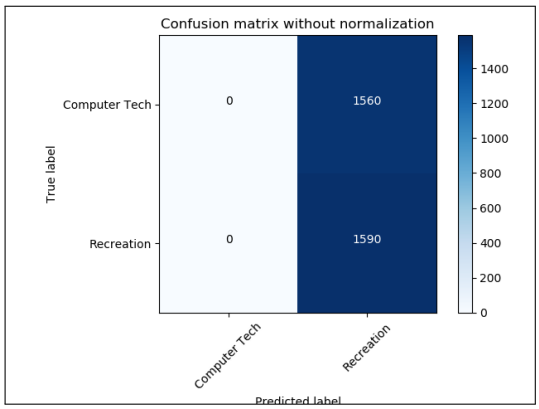
γ	**0.001**	**1**	**1000**
Accuracy	50.48	96.57	96.76
Precision	25.24	96.60	96.77
Recall	50.00	96.56	96.76

Table 4: Accuracy, Precision and Recall

5.1.3 NMF: `min_df=2`

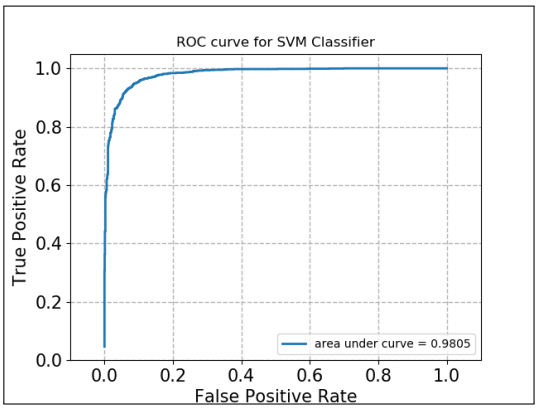


(a) ROC Curve

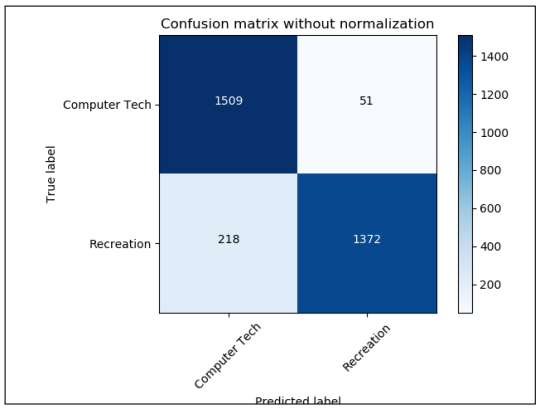


(b) Confusion Matrix

Figure 9: $\gamma = 0.001$

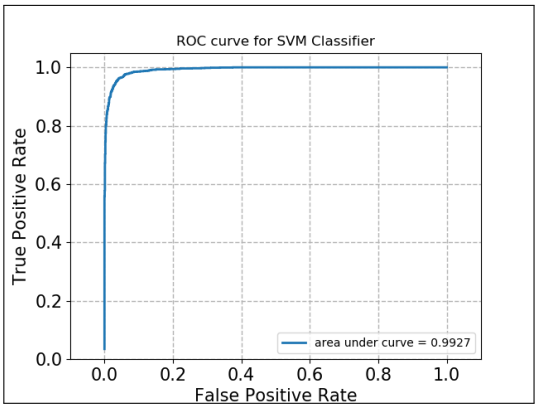


(a) ROC Curve

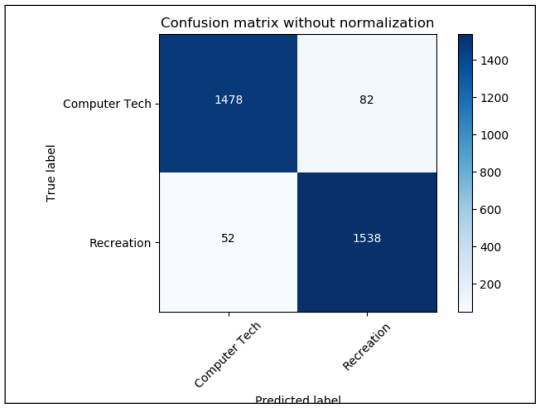


(b) Confusion Matrix

Figure 10: $\gamma = 1$



(a) ROC Curve



(b) Confusion Matrix

Figure 11: $\gamma = 1000$

γ	**0.001**	**1**	**1000**
Accuracy	50.48	91.46	95.75
Precision	25.24	91.90	95.77
Recall	50.00	91.51	95.74

Table 5: Accuracy, Precision and Recall

5.1.4 Analysis

For LSI, both `min_df=2` and `min_df=5` have recall at 0.5 when a soft margin SVM classifier is used. This is because the margin is so large that it labels all data points as Recreational Activity. Both parameter values also have similar performance for higher penalties. This suggests that the rare terms are not worth keeping as it does not significantly improve the accuracy of the classification.

NMF has slightly lower ($\sim$1%) accuracy when compared to LSI.

5.2 SVM with Cross Validation

Cross-validation is carried out by keeping one of the subsamples of the shuffled original training dataset as validation data. A 5-fold cross-validation is used to determine the best γ parameter. In this section, the penalty parameter value is swept using $\gamma = 10^k$, $-3 \leq k \leq 3, k \in \mathbb{Z}$.

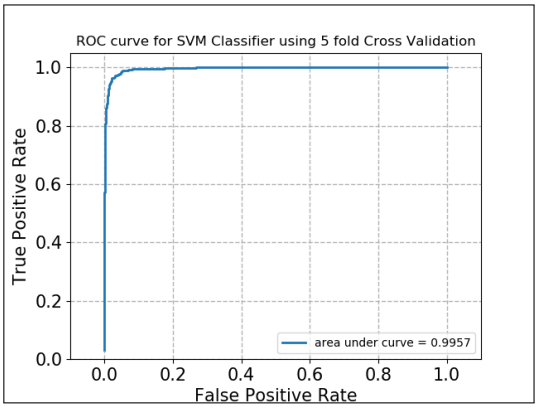
5.2.1 LSI: `min_df=2`

k	-3	-2	-1	0	1	2	3
Accuracy	50.48	50.51	95.87	96.57	96.76	96.73	96.73

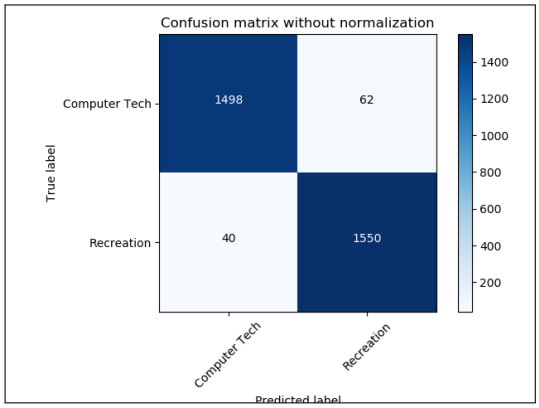
Table 6: Accuracy for Varying k

There is a sharp increase in accuracy between $k = -2$ and $k = -1$. For $k \geq -1$, the accuracy is approximately the same, with a maximum attained at $k = 1$ ($\gamma = 10$).

For $k = 1$, the ROC curve and confusion matrix are plotted in Figure 12, and the accuracy, precision and recall are tabulated in Table 7.



(a) ROC Curve



(b) Confusion Matrix

Figure 12: $k = -1$

Best k	1
Accuracy	96.76
Precision	96.78
Recall	96.75

Table 7: Accuracy, Precision and Recall for Best k

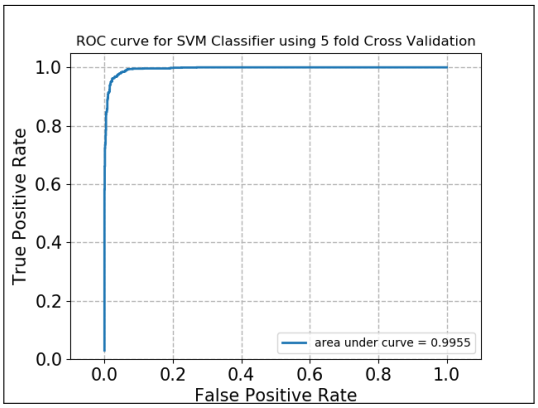
5.2.2 LSI: `min_df=5`

k	-3	-2	-1	0	1	2	3
Accuracy	50.48	50.76	95.68	96.57	96.70	96.63	96.57

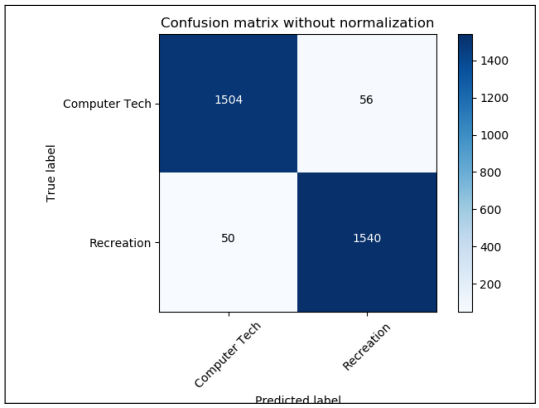
Table 8: Accuracy for Varying k

Similar to when `min_df=2`, there is a sharp increase in accuracy between $k = -2$ and $k = -1$. For $k \geq -1$, the accuracy is approximately the same, with a maximum attained at $k = 1$ ($\gamma = 10$).

For $k = 1$, the ROC curve and confusion matrix are plotted in Figure 13, and the accuracy, precision and recall are shown in Table 9.



(a) ROC Curve



(b) Confusion Matrix

Figure 13: $\gamma = 1000$

Best k	1
Accuracy	96.70
Precision	96.72
Recall	96.69

Table 9: Accuracy, Precision and Recall for Best k

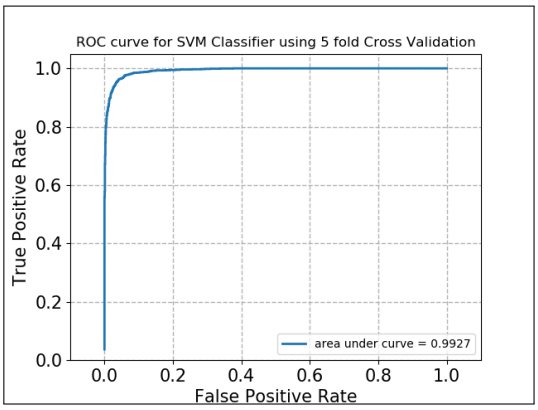
5.2.3 NMF: `min_df=2`

k	-3	-2	-1	0	1	2	3
Accuracy	50.48	50.48	50.48	91.46	94.25	95.17	95.75

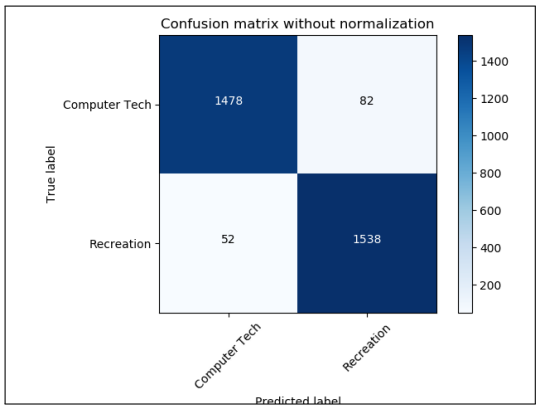
Table 10: Accuracy for Varying k

The sharp increase in accuracy is between $k = -1$ and $k = 0$. For $k \geq 1$, the accuracy is approximately the same, with a maximum attained at $k = 3$ ($\gamma = 1000$). Hence, the optimal penalty might lie beyond the swept range.

For $k = 3$, the ROC curve and confusion matrix are plotted in Figure 13, and the accuracy, precision and recall are shown in Table 9.



(a) ROC Curve



(b) Confusion Matrix

Figure 14: $\gamma = 1000$

Best k	3
Accuracy	95.75
Precision	95.77
Recall	95.74

Table 11: Accuracy, Precision and Recall for Best k

5.2.4 Analysis

For LSI, the results attained are similar to those without cross-validation. `min_df=2` and `min_df=2` have approximately the same accuracy. Again, this suggests that terms that appear in only a few documents are not important features and hence can be removed from the dataset.

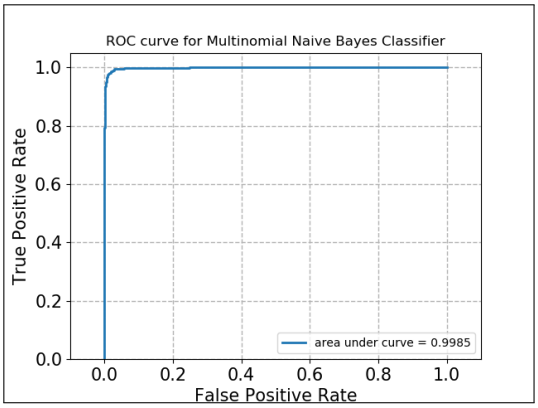
With regards to dimensionality reduction method, LSI outperformed NMF in terms of accuracy by approximately 1%.

5.3 Naïve Bayes

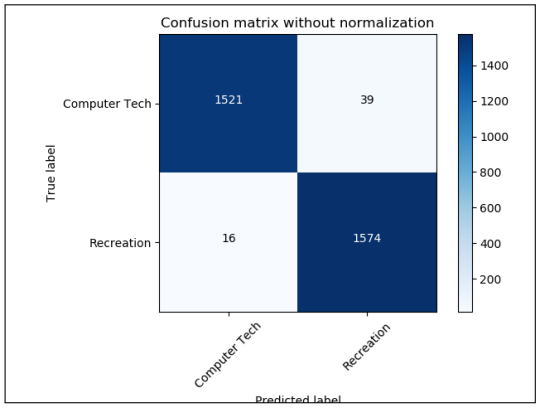
In this assignment, the bag of words assumption is used, where the order of terms do not matter. Naïve Bayes works by applying Bayes' theorem with the assumption of strong independence between features, in this case, the term frequencies.

The multinomial Naïve Bayes classifier only takes non-negative inputs and can handle sparse matrices. Hence, the TFIDF vectors without dimensionality reduction are chosen as the input.

5.3.1 Without Dimensionality Reduction: `min_df=2`



(a) ROC Curve



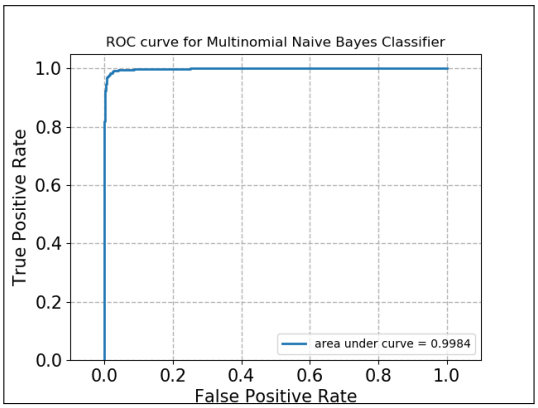
(b) Confusion Matrix

Figure 15

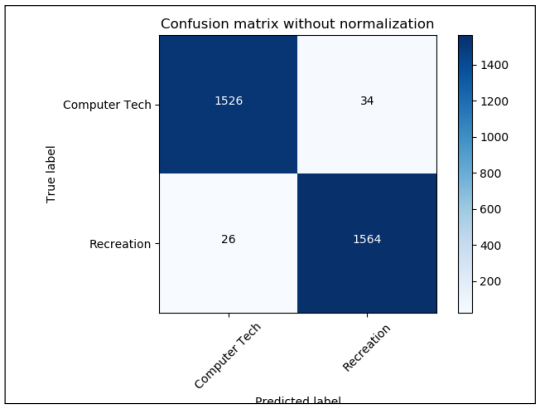
Accuracy	98.25
Precision	98.27
Recall	98.25

Table 12: Accuracy, Precision and Recall

5.3.2 Without Dimensionality Reduction: `min_df=5`



(a) ROC Curve



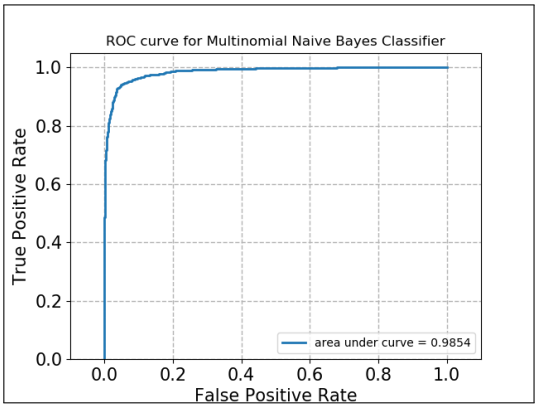
(b) Confusion Matrix

Figure 16

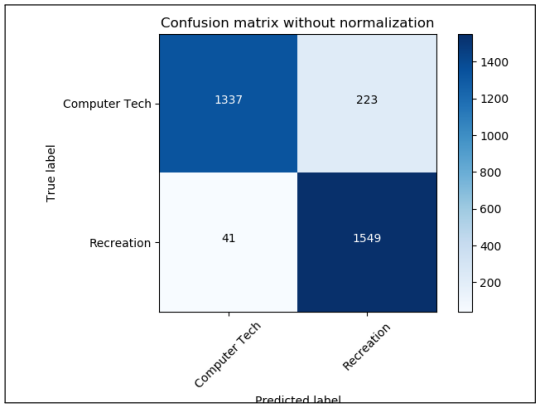
Accuracy	98.10
Precision	98.10
Recall	98.09

Table 13: Accuracy, Precision and Recall

5.3.3 NMF: `min_df=2`



(a) ROC Curve



(b) Confusion Matrix

Figure 17

Accuracy	91.62
Precision	92.22
Recall	91.56

Table 14: Accuracy, Precision and Recall

5.3.4 Analysis

Without dimensionality reduction, the accuracy for `min_df=2` and `min_df=2` are approximately equal. The higher accuracy when compared to using the compressed dataset by NMF might be due to more features (terms) being included.

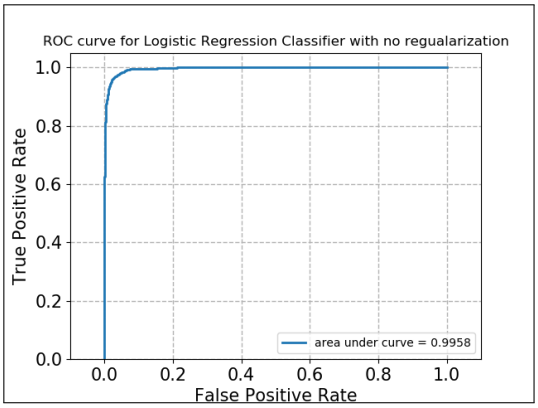
NMF yielded less accurate results than when classified using SVM. This could be due to the assumption of feature independence. While both classifiers disregard the order of terms, naïve Bayes further assumes that the frequency of a term is independent of the frequency of another term, which is generally not true.

5.4 Logistic Regression

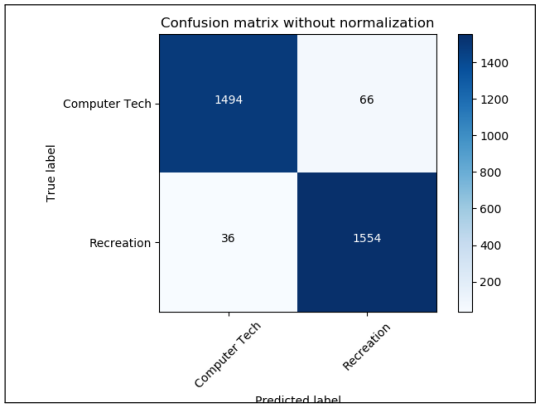
Logistic regression measures the relationship between the input features and target output by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

In this section, it is first used to classify data points without regularization, then compared against accuracy attained with L1 and L2 regularization. (Note that the regularization parameter in this report is the reciprocal of the input parameter of the Python function, i.e. the larger value, the more important the regularization term.)

5.4.1 LSI: `min_df=2`



(a) ROC Curve



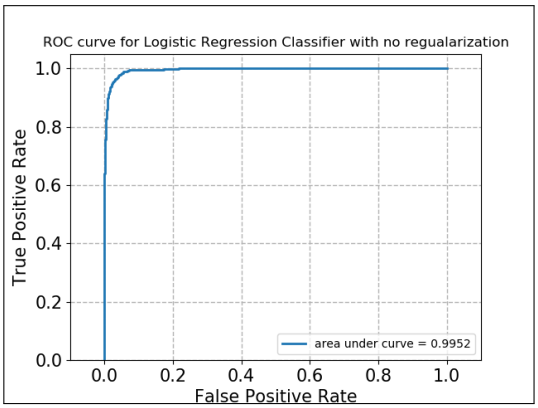
(b) Confusion Matrix

Figure 18

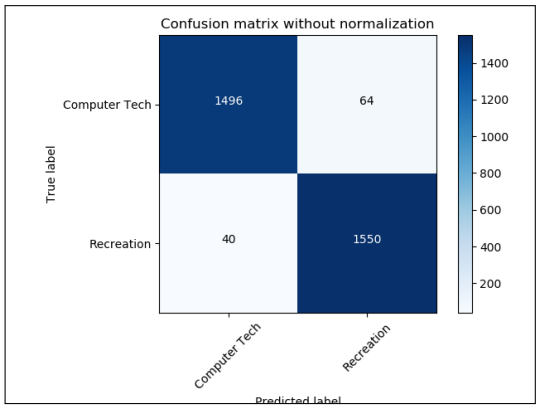
Accuracy	96.76
Precision	96.79
Recall	96.75

Table 15: Accuracy, Precision and Recall

5.4.2 LSI: `min_df=5`



(a) ROC Curve



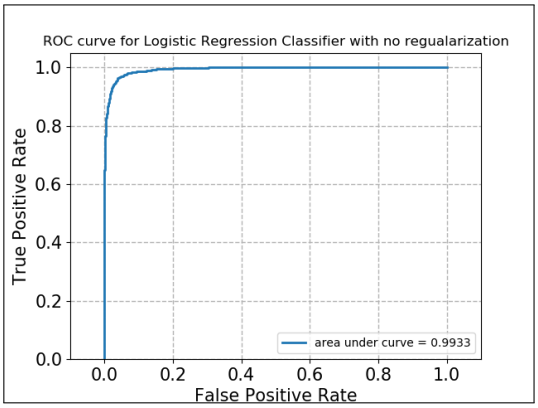
(b) Confusion Matrix

Figure 19

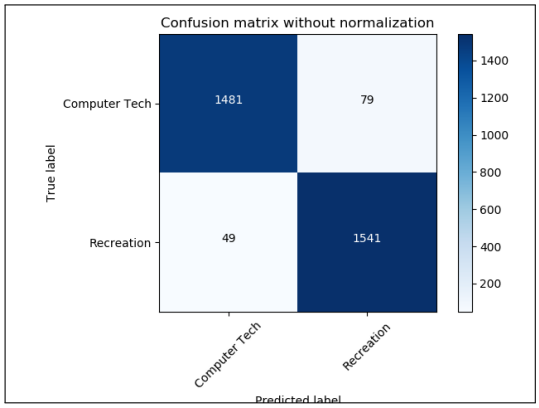
Accuracy	96.70
Precision	96.72
Recall	96.69

Table 16: Accuracy, Precision and Recall

5.4.3 NMF: `min_df=2`



(a) ROC Curve



(b) Confusion Matrix

Figure 20

Accuracy	95.94
Precision	95.96
Recall	95.93

Table 17: Accuracy, Precision and Recall

5.4.4 Analysis

Similar to SVM, LSI with `min_df=2` and `min_df=5` have approximately equal accuracy, and both outperformed NMF.

5.5 Logistic Regression with L1 Regularization

5.5.1 LSI: `min_df=2`

λ	0.001	0.01	0.1	1	10	100	1000	10000
Accuracy	49.52	90.76	94.79	96.67	96.73	96.73	96.73	96.73

Table 18: Accuracy for Varying Regularization Coefficient λ

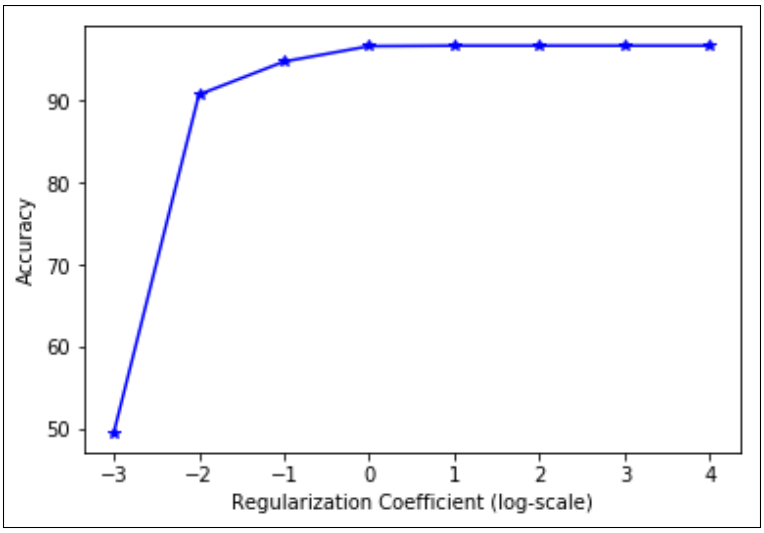
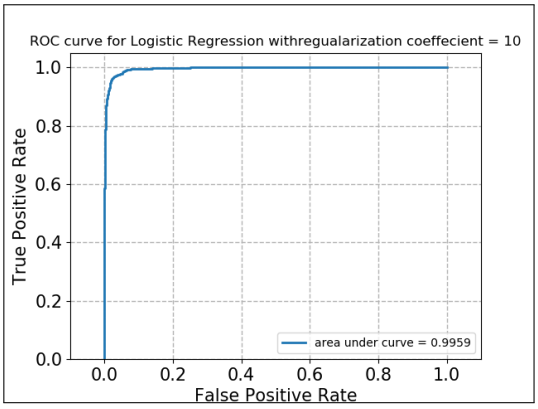
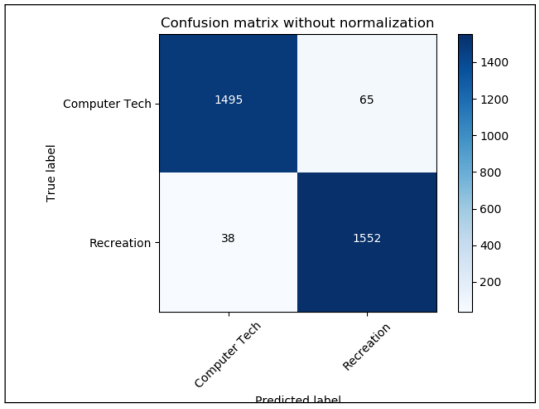


Figure 21: Accuracy against Regularization Coefficient

The highest accuracy is obtained when $\lambda = 10$.



(a) ROC Curve



(b) Confusion Matrix

Figure 22: $\lambda = 10$

Accuracy	96.73
Precision	96.75
Recall	96.72

Table 19: Accuracy, Precision and Recall for $\lambda = 10$

5.5.2 LSI: `min_df=5`

λ	0.001	0.01	0.1	1	10	100	1000	10000
Accuracy	49.52	90.76	94.53	96.22	96.79	96.92	96.95	96.95

Table 20: Accuracy for Varying Regularization Coefficient λ

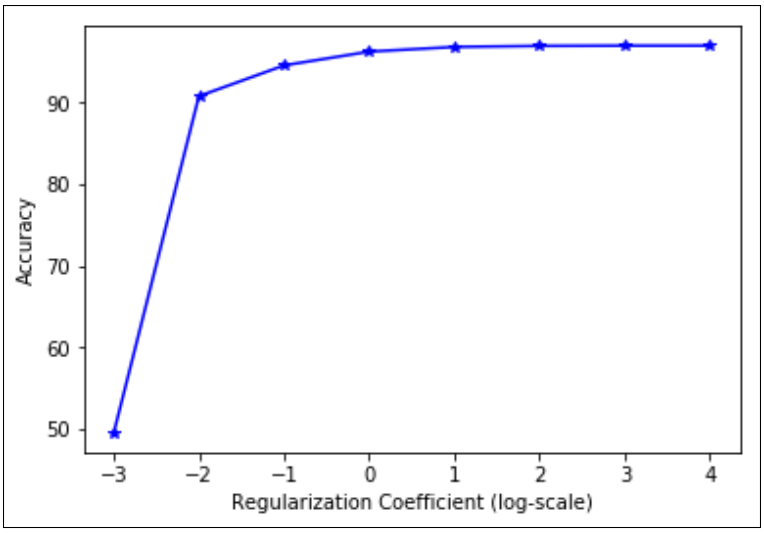
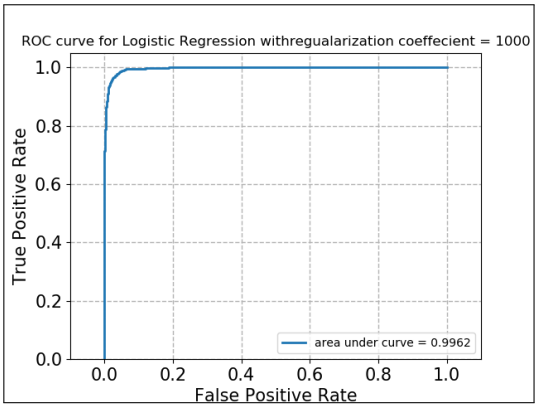
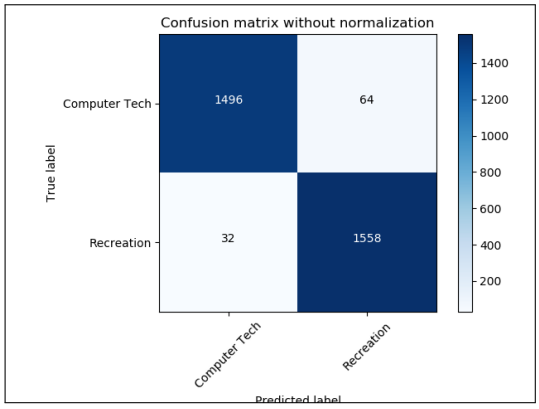


Figure 23: Accuracy against Regularization Coefficient

The highest accuracy is obtained when $\lambda = 1000$.



(a) ROC Curve



(b) Confusion Matrix

Figure 24: $\lambda = 1000$

Accuracy	96.95
Precision	96.98
Recall	96.94

Table 21: Accuracy, Precision and Recall for $\lambda = 1000$

5.5.3 NMF: `min_df=2`

λ	0.001	0.01	0.1	1	10	100	1000	10000
Accuracy	49.52	49.52	64.79	94.79	95.74	95.90	95.94	95.94

Table 22: Accuracy for Varying Regularization Coefficient λ

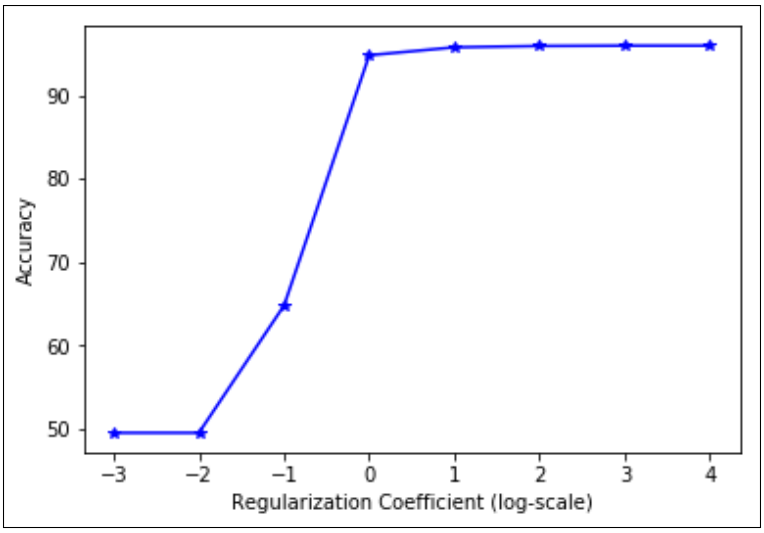
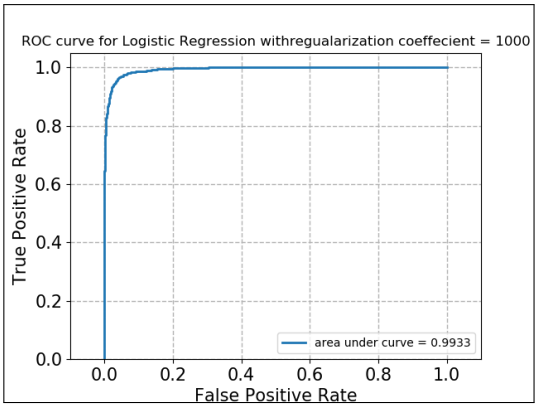
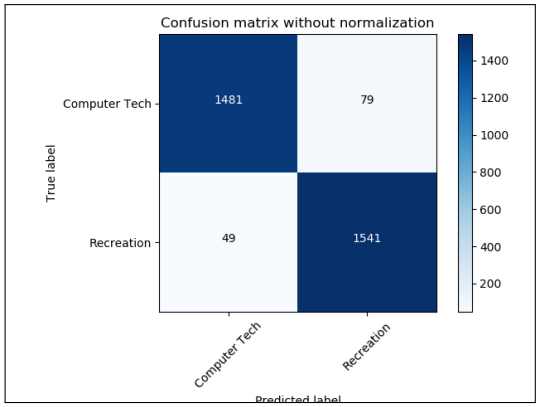


Figure 25: Accuracy against Regularization Coefficient

The highest accuracy is obtained when $\lambda = 1000$.



(a) ROC Curve



(b) Confusion Matrix

Figure 26: $\lambda = 1000$

Accuracy	95.94
Precision	95.96
Recall	95.93

Table 23: Accuracy, Precision and Recall for $\lambda = 1000$

5.6 Logistic Regression with L2 Regularization

5.6.1 LSI: `min_df=2`

λ	0.001	0.01	0.1	1	10	100	1000	10000
Accuracy	63.62	92.67	95.87	96.35	96.63	96.70	96.76	96.79

Table 24: Accuracy for Varying Regularization Coefficient λ

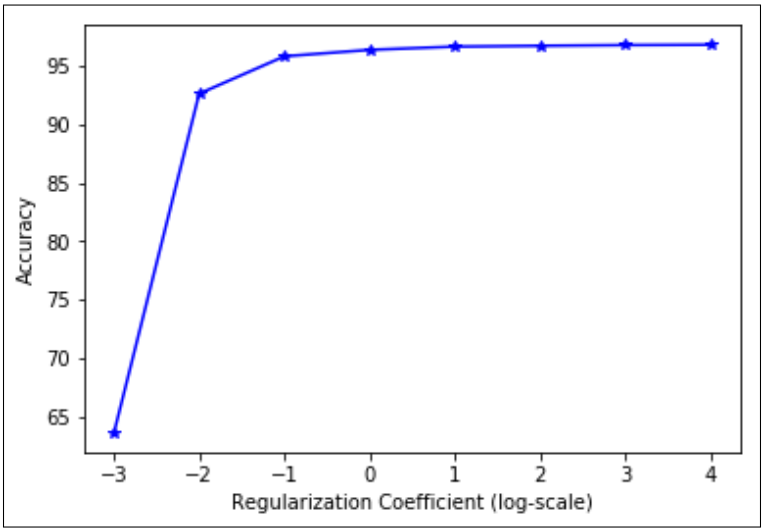
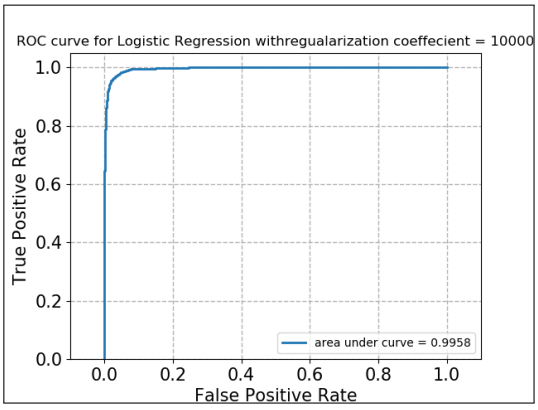
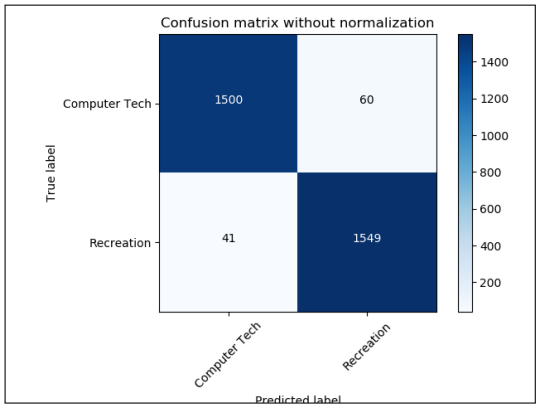


Figure 27: Accuracy against Regularization Coefficient

The highest accuracy is obtained when $\lambda = 10000$.



(a) ROC Curve



(b) Confusion Matrix

Figure 28: $\lambda = 10000$

Accuracy	96.79
Precision	96.80
Recall	96.79

Table 25: Accuracy, Precision and Recall for $\lambda = 10000$

5.6.2 LSI: `min_df=5`

λ	0.001	0.01	0.1	1	10	100	1000	10000
Accuracy	68.41	93.01	95.59	96.38	96.89	96.98	96.92	96.85

Table 26: Accuracy for Varying Regularization Coefficient λ

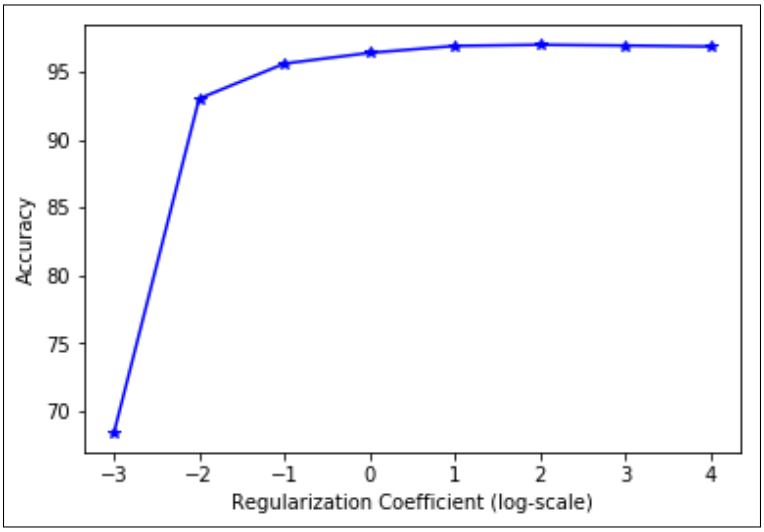
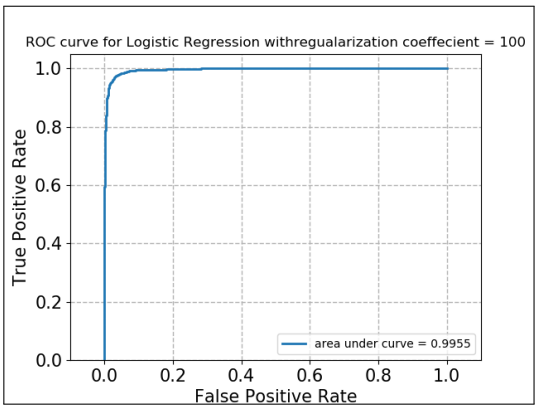
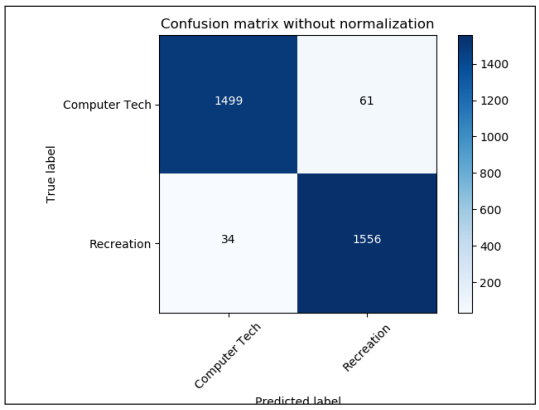


Figure 29: Accuracy against Regularization Coefficient

The highest accuracy is obtained when $\lambda = 100$.



(a) ROC Curve



(b) Confusion Matrix

Figure 30: $\lambda = 100$

Accuracy	96.98
Precision	97.00
Recall	96.98

Table 27: Accuracy, Precision and Recall for $\lambda = 100$

5.6.3 NMF: `min_df=2`

λ	0.001	0.01	0.1	1	10	100	1000	10000
Accuracy	50.48	50.48	89.90	92.38	93.87	94.70	95.49	95.87

Table 28: Accuracy for Varying Regularization Coefficient λ

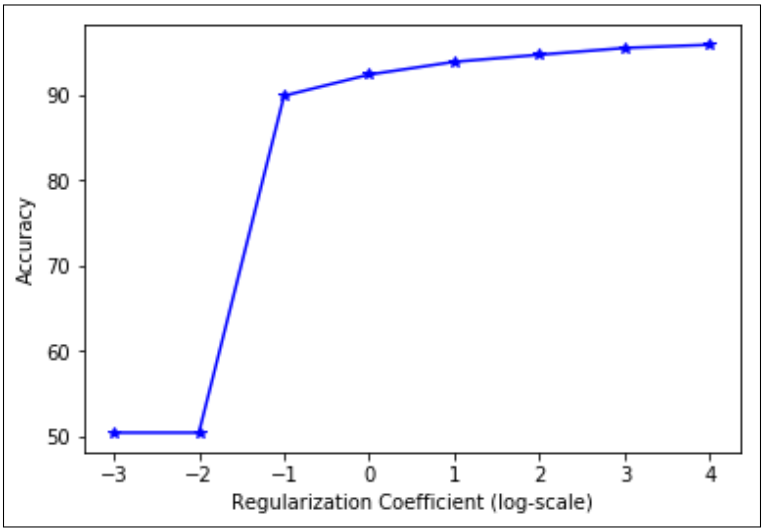
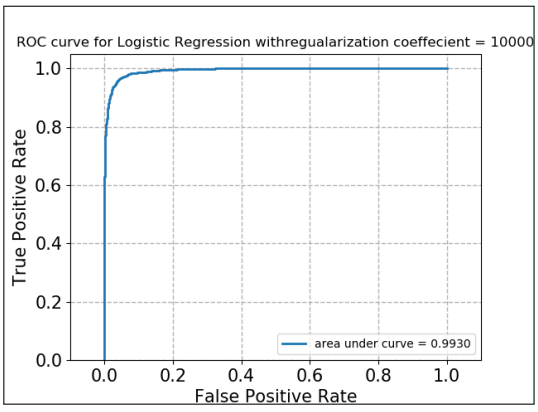
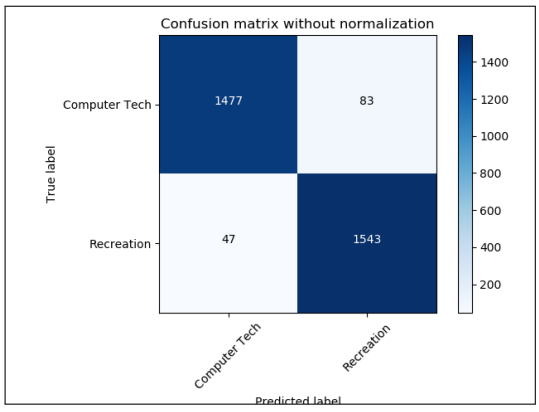


Figure 31: Accuracy against Regularization Coefficient

The highest accuracy is obtained when $\lambda = 10000$.



(a) ROC Curve



(b) Confusion Matrix

Figure 32: $\lambda = 10000$

Accuracy	95.87
Precision	95.91
Recall	95.86

Table 29: Accuracy, Precision and Recall for $\lambda = 10000$

5.7 Analysis

For both L1 and L2 regularization, accuracy improves with larger regularization coefficients before decreasing after hitting a maximum. The maximum accuracy attained is approximately equal for both cases. `min_df=2` and `min_df=5` for LSI again achieved similar accuracy, and they both outperformed NMF.

L1 regularization yields a sparse model, which means that it has few coefficients as most are zero and hence eliminated. L2 regularization, on the other hand, yields a model with many small coefficients.

L2 regularization generally yields higher accuracy when compared to L1, although this is not the case in this particular finding. However, L1 can deal with sparse feature spaces and helps with feature selection.

6 Multiclass Classification

Binary classifiers were investigated in the previous section and in this section, they are extended to multiclass classification using different algorithms.

The dataset is taken from the following subclasses:

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

`min_df` is set at 2 for the subsequent sections.

6.1 Naïve Bayes

6.1.1 Without Dimensionality Reduction

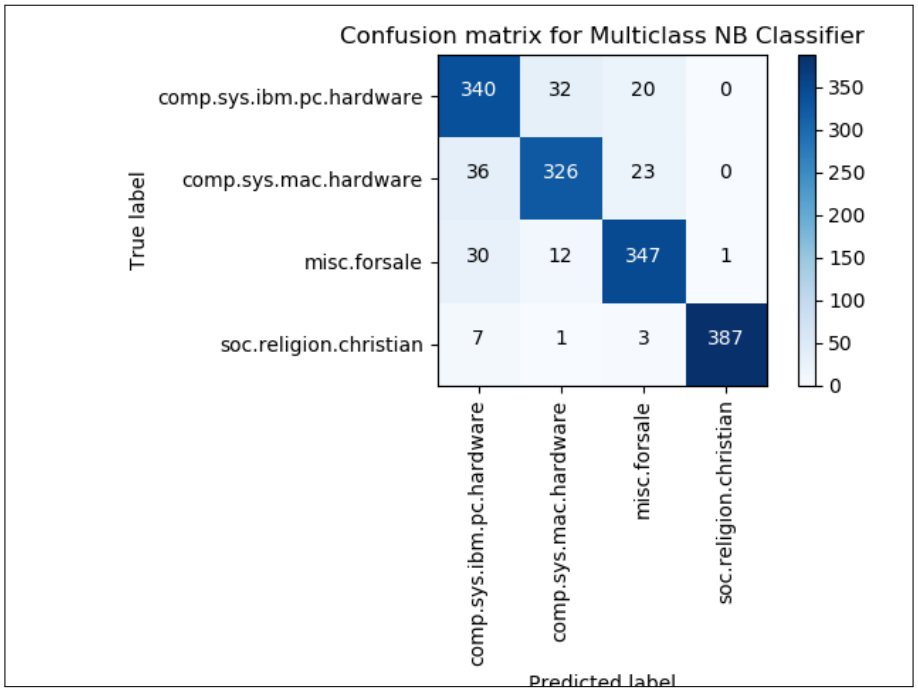


Figure 33: Confusion Matrix

Accuracy	89.46
Precision	89.56
Recall	89.41

Table 30: Accuracy, Precision and Recall

6.1.2 NMF

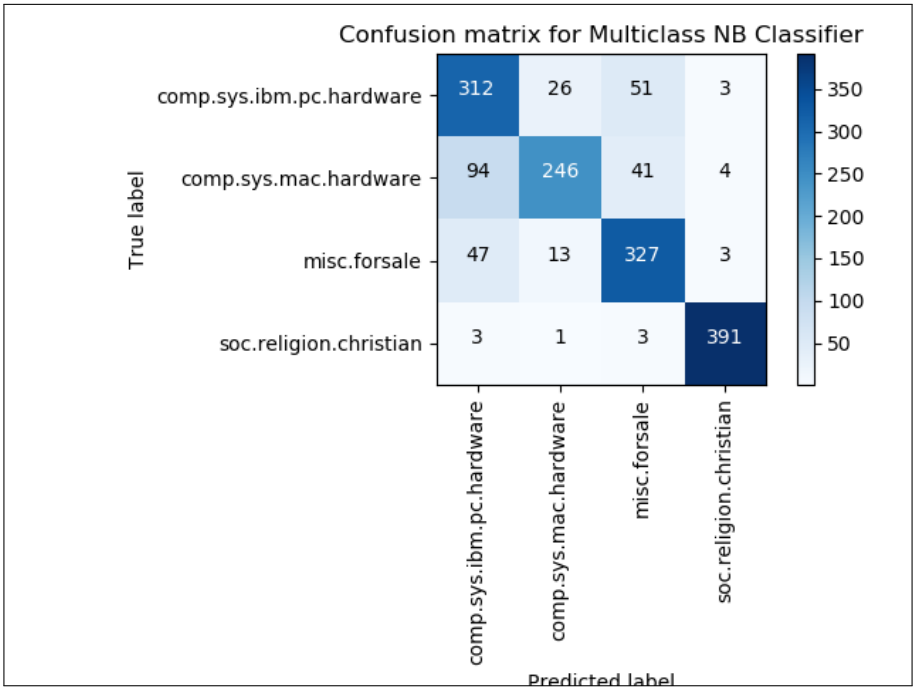


Figure 34: Confusion Matrix

Accuracy	81.53
Precision	82.36
Recall	81.39

Table 31: Accuracy, Precision and Recall

6.1.3 Analysis

A much higher accuracy is achieved when the original dataset is used, relative to when it is reduced through NMF.

soc.religion.christian is the subclass with the largest proportion of data points correctly classified. This is expected because it is the most distinct category among the four. Both classifiers perform less accurately when classifying documents from comp.sys.ibm.pc.hardware and comp.sys.mac.hardware. This could be due to both subclasses being related to hardware and hence some features becoming less distinguishing.

6.2 One-vs.-One SVM

6.2.1 LSI

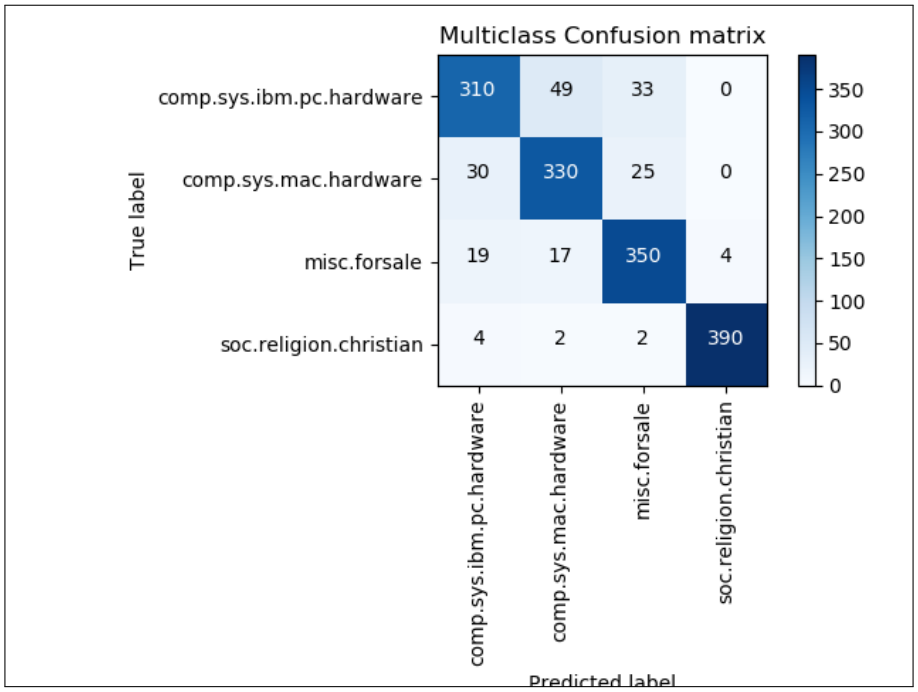


Figure 35: Confusion Matrix

Accuracy	88.18
Precision	88.17
Recall	88.13

Table 32: Accuracy, Precision and Recall

6.2.2 NMF

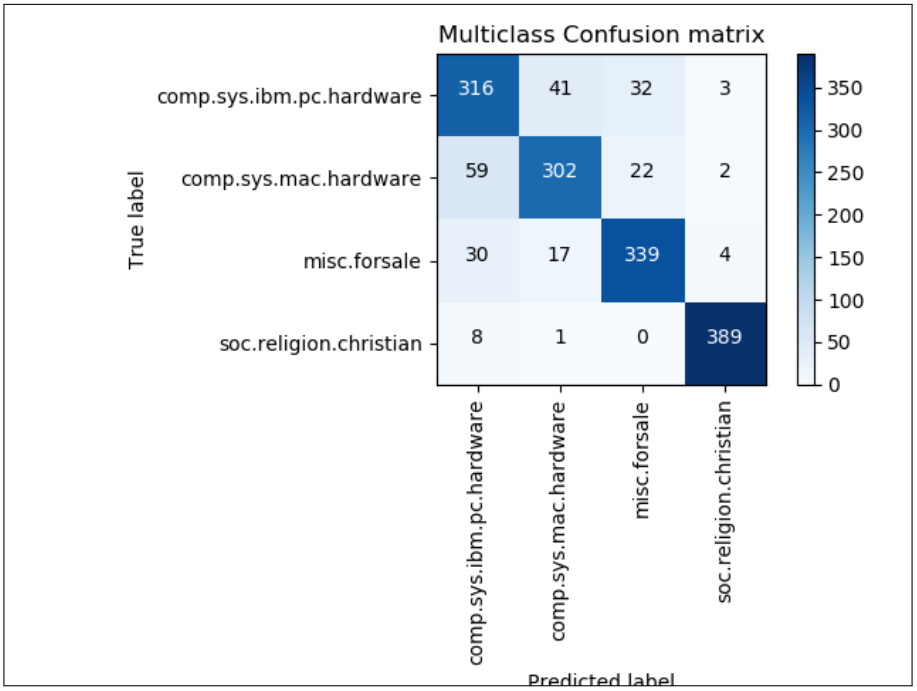


Figure 36: Confusion Matrix

Accuracy	86.01
Precision	86.04
Recall	85.93

Table 33: Accuracy, Precision and Recall

6.3 One-vs.-Rest SVM

6.3.1 LSI

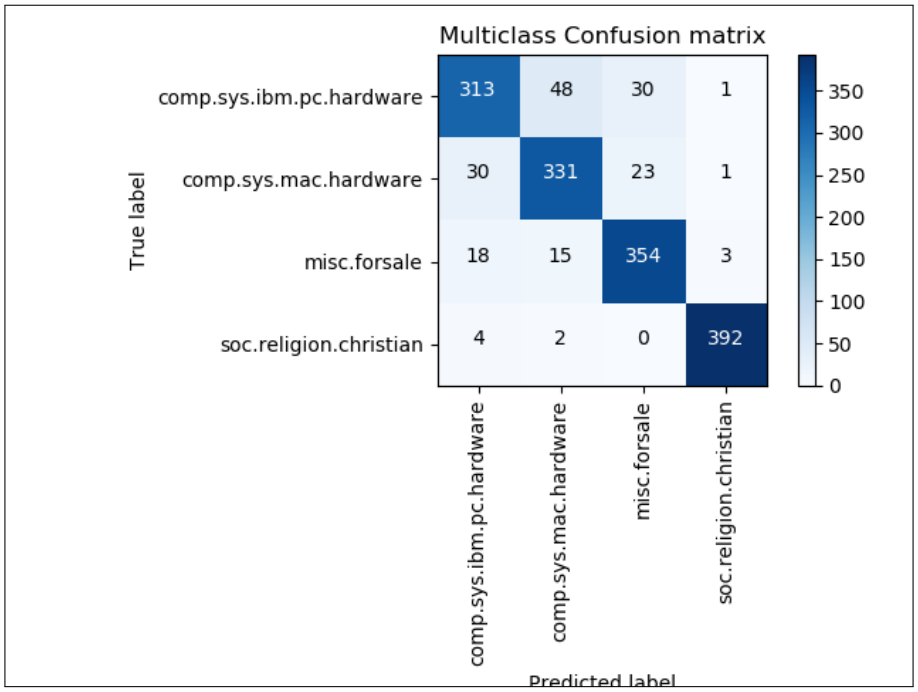


Figure 37: Confusion Matrix

Accuracy	88.82
Precision	88.76
Recall	88.77

Table 34: Accuracy, Precision and Recall

6.3.2 NMF

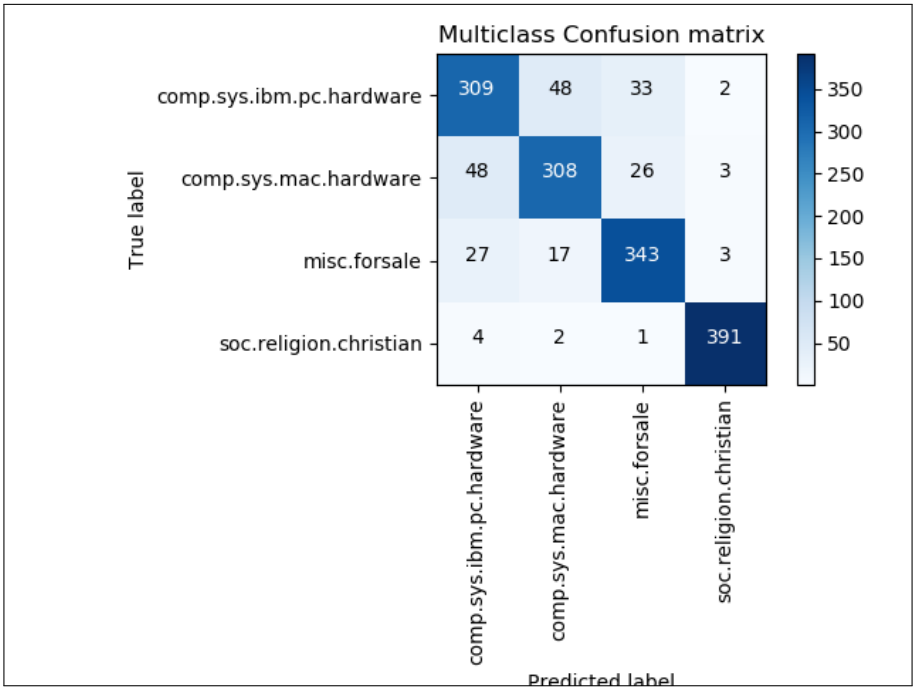


Figure 38: Confusion Matrix

Accuracy	86.33
Precision	86.22
Recall	86.25

Table 35: Accuracy, Precision and Recall

6.4 Analysis

Similar to previous results, LSI outperformed NMF for both one-vs.-one and one-vs.-rest.

Although being computationally more expensive, one-vs.one is typically less sensitive to problems related to dataset imbalance. In this case, the dataset is (approximately) balanced, with the number of documents in each category being:

- comp.sys.ibm.pc.hardware: 590
- comp.sys.mac.hardware: 578
- misc.forsale: 585
- soc.religion.christian: 599

Hence, both one-vs.-one and one-vs.-rest yield similar accuracy performance in this experiment.