EE219 Project 2

Classification Analysis

Winter 2017

Xiongfeng Hu, 304753117

Yanming Zhang, 004761717

Cong Peng, 904760493

Content

Introduction

Classification is a task of identifying a category, from a predefined set, to which a data point belongs, on the basis of a training data set with known category memberships. In this project we implement different methods for classifying textual data - the 20 Newsgroup Dataset, including Support Vector Machines, Naive Bayes and Logistic Regression.

Dataset & Problem Statement

The “20 Newsgroups” dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic.

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

Table 1 Sub-classes of Computer Technology and Recreational activity

Computer technology	Recreational activity
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey

Question(a) Histogram of Number of Documents Per Topic

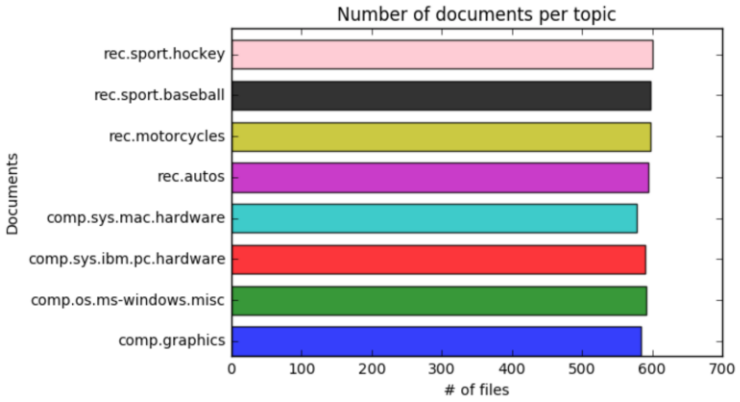

Figure 1 Histogram of Number of Documents Per Topic

As we can see from Figure 1, the numbers of documents per topic are evenly distributed. And the whole data size if 4732, the train data size is 4154.

Modeling Text Data and Feature Extraction

We use Term Frequency-Inverse Document Frequency (TFxIDF) metric to capture the importance of a word to a document in a corpus.

Question(b) TFxIDF Vector Representation

We turn the documents in the data set into TFxIDF vector representations. We first tokenize the documents and exclude the stop words, punctuations, and different stems of a word.

The shape of the TFxIDF vector is (18846, 57177), which means that the final number of terms we extract is 57177.

Question(c) TFxICF And 10 Most Significant Terms

To find the top 10 significant terms for classes ***'comp.sys.ibm.pc.hardware'***, ***'comp.sys.mac.hardware'***, ***'misc.forsale'***, and ***'soc.religion.christian'***. The following steps are conducted:
1. Firstly, all stop words, punctuations and stems are removed. Then we conduct text clean by preserving only the noun words which are most likely to be terms.
2. TFxICF is computed for each term using given formula.
3. The results are sorted and stored as below:

Table 2 Top 10 significant terms for classes

comp.sys.ibm.pc.hardware	comp.sys.mac.hardware	misc.forsale	soc.religion.christian
card	bit	sale	thing
control	appl	mail	christ
use	monitor	game	church
drive	card	condit	word
disk	use	price	peopl
scsi	drive	drive	god
pc	problem	ship	way
problem	disk	card	jesus
bus	work	use	sin
time	mac	offer	time

Feature Selection

Now, the dimensionality of our TFxIDF vectors is over thousands, and the vectors are actually sparse, which diminished the performance of many learning algorithms. Therefore, we need reduce the dimension of the vectors.

Question(d) Latent Semantic Indexing (LSI) Representation Of TFxIDF Vectors

We use Latent Semantic Indexing (LSI) to find the optimal representation of the data in a lower dimensional space. We use TruncatedSVD from sklearn's decomposition package to decompose the vectors with 50 as the number of elements. Therefore, we get the selected features for our learning algorithms.

Learning Algorithms

Question(e) Linear Support Vector Machines (SVM) Method

Here, we use SVM method to separate the documents into Computer Technology vs Recreational Activity groups. In order to show the performance, we plot the Receiver Operating Characteristic (ROC) curve, report the confusion matrix and calculate the accuracy, recall and precision of the classifier.
We first Construct a training set and then build a SVM classifier by svm.LinearSVM() and train it using the training set, and then construct a testing set to test our model. The ROC curve is shown as Figure 2.

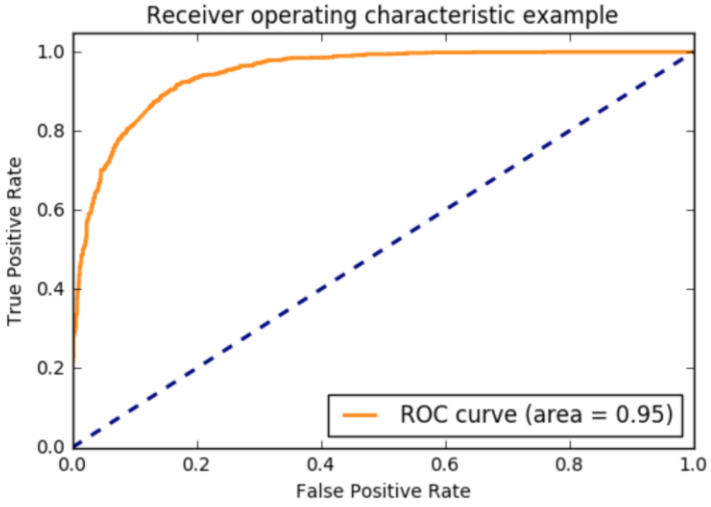

Figure 2 ROC curve of SVM classifier

The non-normalized confusion matrix and the normalized confusion matrix are shown as Figure 3 and Figure 4.

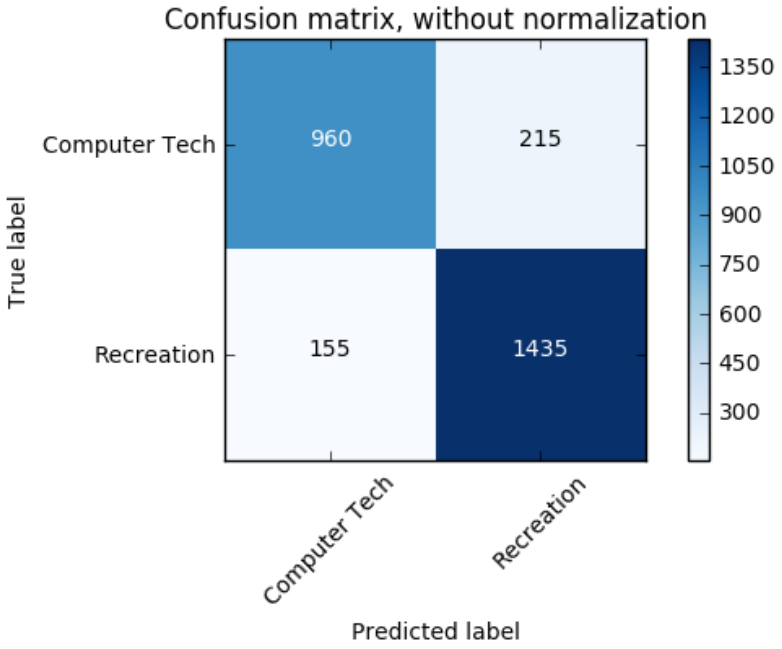

Figure 3 Confusion Matrix of SVM classifier, without normalization

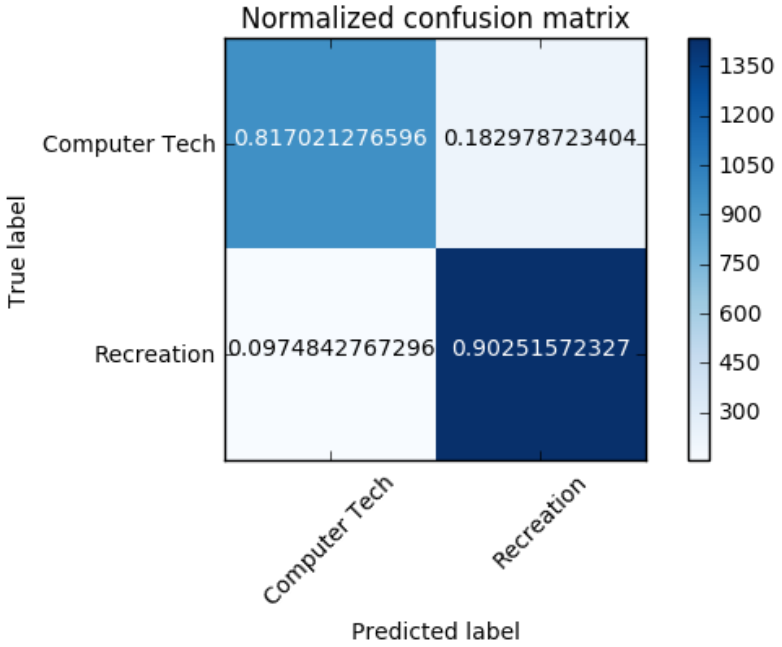

Figure 4 Confusion Matrix of SVM classifier, with normalization

Then we compute the Accuracy, Recall and Precision of this model. The results are shown in Table 3.

Table 3 Accuracy, Recall and Precision of SVM classifier

Accuracy	Recall	Precision
0.866184448463	0.90251572327	0.869696969697

Question(f) Soft Margin SVM Method

We then use Soft Margin SVM Method to separate the documents with 5-fold cross-validation, and find the best value of the parameter γ in the range $\{10^{-k}| -3 \leq k \leq 3, k \in Z\}$ for the optimization problem.

We first split the data into 5 fold, and then build 5*7 SVM classifiers to find the best value of γ. Our results show that the best penalty value is 0.01, and the score with this value of γ is 0.934520754498.
The non-normalized confusion matrix and the normalized confusion matrix are shown as Figure 5 and Figure 6.

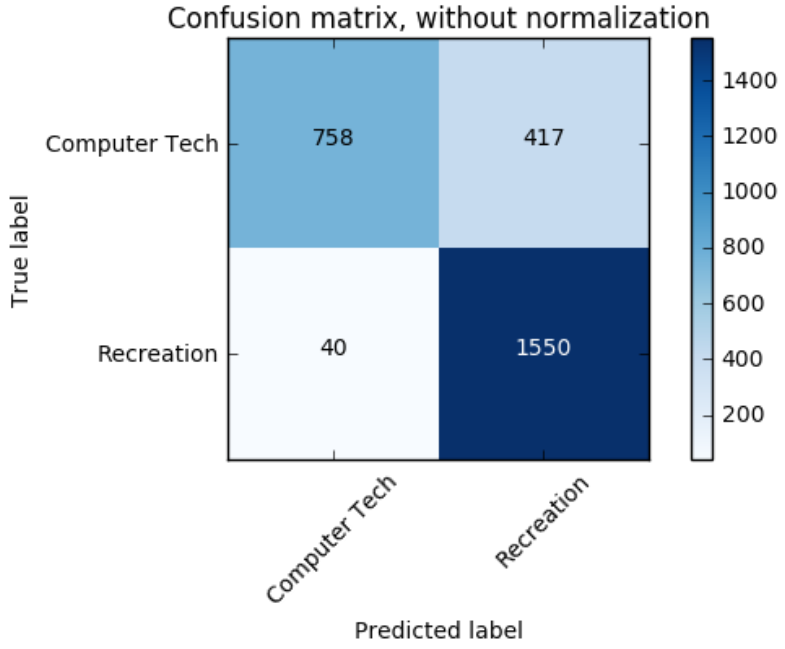

Figure 5 Confusion Matrix of Soft Margin SVM classifier, without normalization

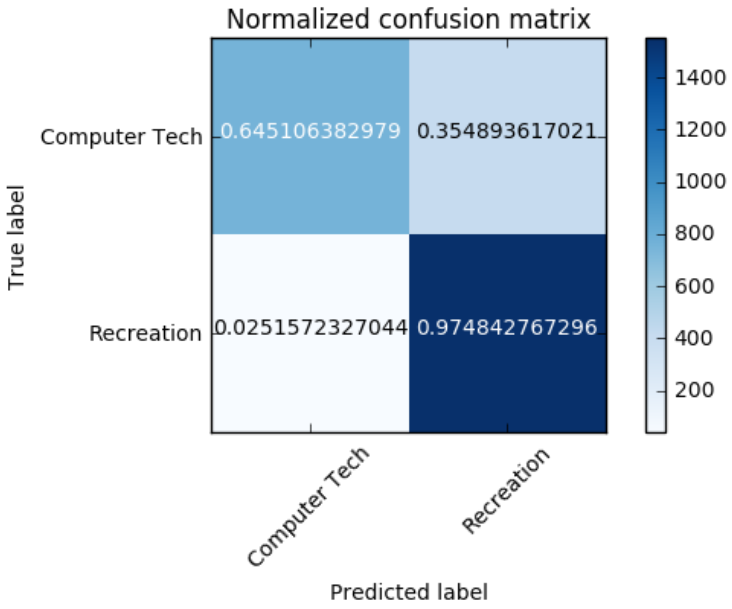

Figure 6 Confusion Matrix of Soft Margin SVM classifier, with normalization

Then we compute the Accuracy, Recall and Precision of this model. The results are shown in Table 4.

Table 4 Accuracy, Recall and Precision of Soft Margin SVM classifier

Accuracy	Recall	Precision
0.834719710669	0.788002033554	0.974842767296

Question(g) Naïve Bayes Algorithm

We use naïve Bayes algorithm - sklearn.naive_bayes.BernoulliNB for the same classification task. The ROC curve is shown as Figure 7.

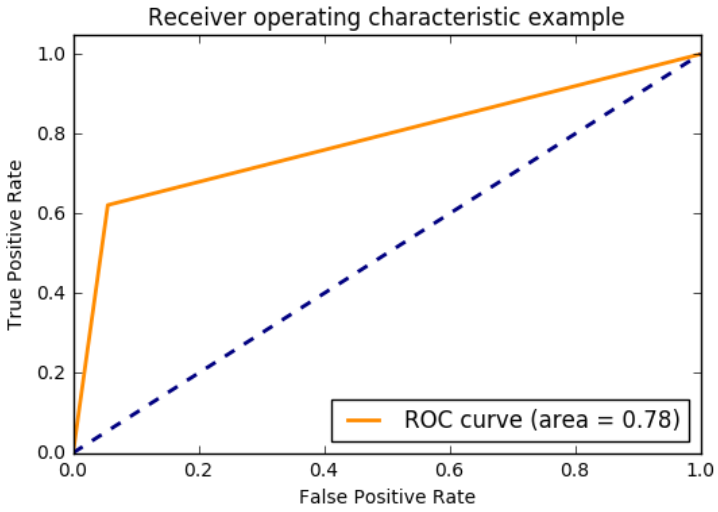

Figure 7 ROC curve of Naïve Bayes classifier

The non-normalized confusion matrix and the normalized confusion matrix are shown as Figure 8 and Figure 9.

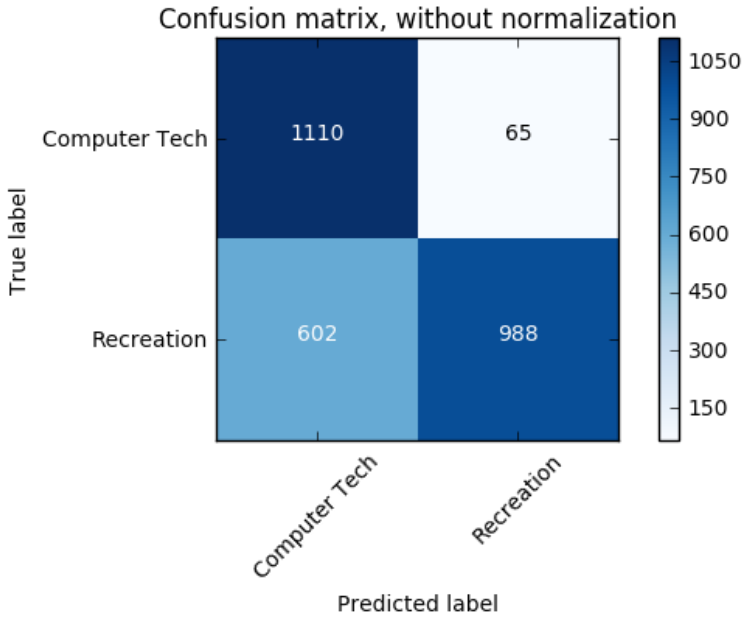

Figure 8 Confusion Matrix of Naïve Bayes classifier, without normalization

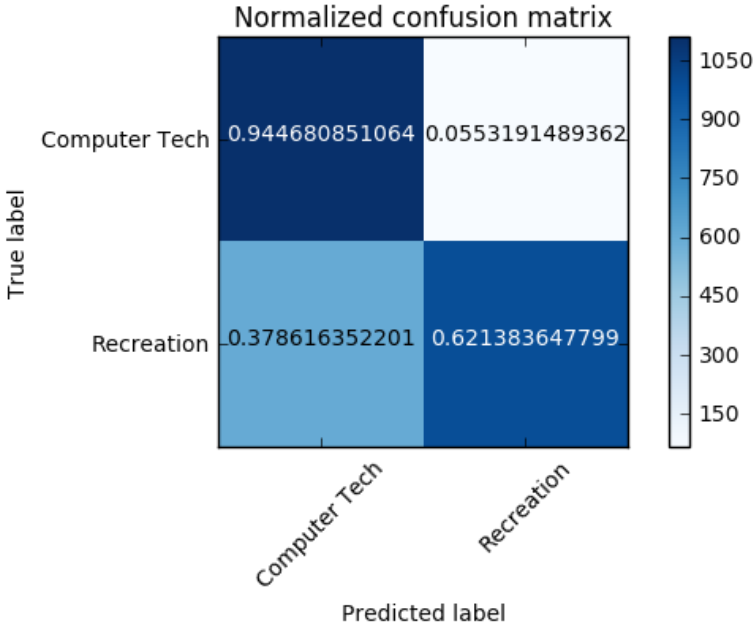

Figure 9 Confusion Matrix of Naïve Bayes classifier, with normalization

Then we compute the Accuracy, Recall and Precision of this model. The results are shown in Table 5.

Table 5 Accuracy, Recall and Precision of Naïve Bayes classifier

Accuracy	Recall	Precision
0.75877034358	0.621383647799	0.938271604938

As seen above, Naive Bayes has less area under the ROC curve as compared to SVM classifier, the Accuracy, Recall and Precision of this model are also less than those of SVM classifier, showing that the performance of Naive Bayes classifier is worse than that of SVM classifier.

Question(h) & (i) Logistic Regression Classifier With Regularization

We use the logistic regression classifier for the same task and implement both l-1 and l-2 norm regularizations.

For l-1 norm regularizations, the ROC curve is shown as Figure 10. The non-normalized confusion matrix and the normalized confusion matrix are shown as Figure 11 and Figure 1--2.

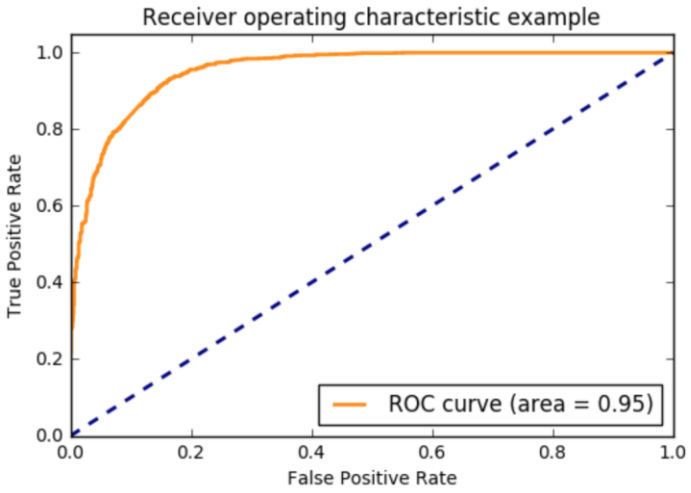

Figure 10 ROC curve of l-1 norm logistic regression classifier

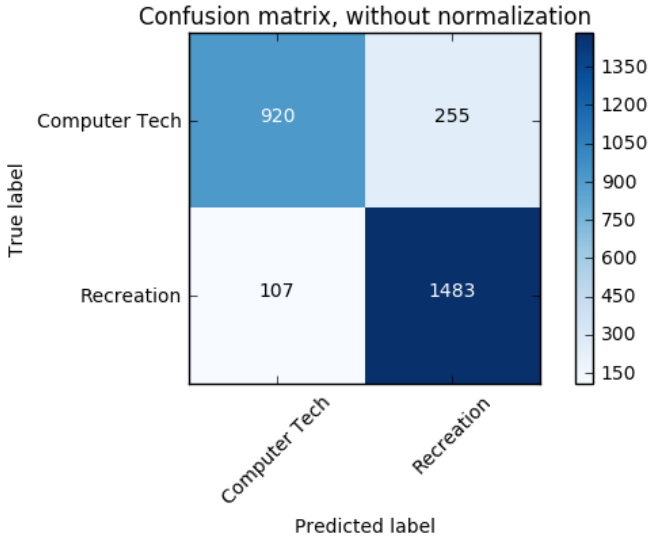

Figure 11 Confusion Matrix of l-1 norm logistic regression classifier, without normalization

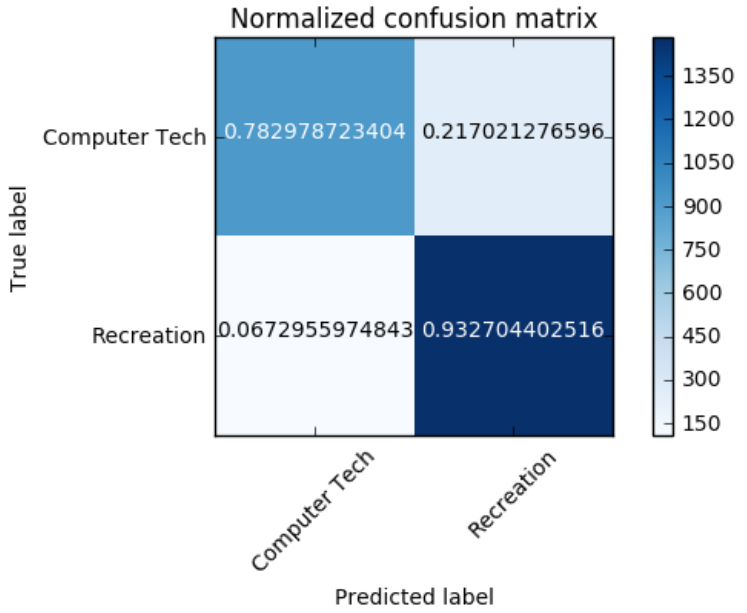

Figure 12 Confusion Matrix of l-1 norm logistic regression classifier, with normalization

Then we compute the Accuracy, Recall and Precision of this model. The results are shown in Table 6.

Table 6 Accuracy, Recall and Precision of 1 Norm Logistic Regression

Accuracy	Recall	Precision
0.869077757685	0.932704402516	0.853279631761

For l-2 norm regularizations, the ROC curve is shown as Figure 13. The non-normalized confusion matrix and the normalized confusion matrix are shown as Figure 14 and Figure 15.

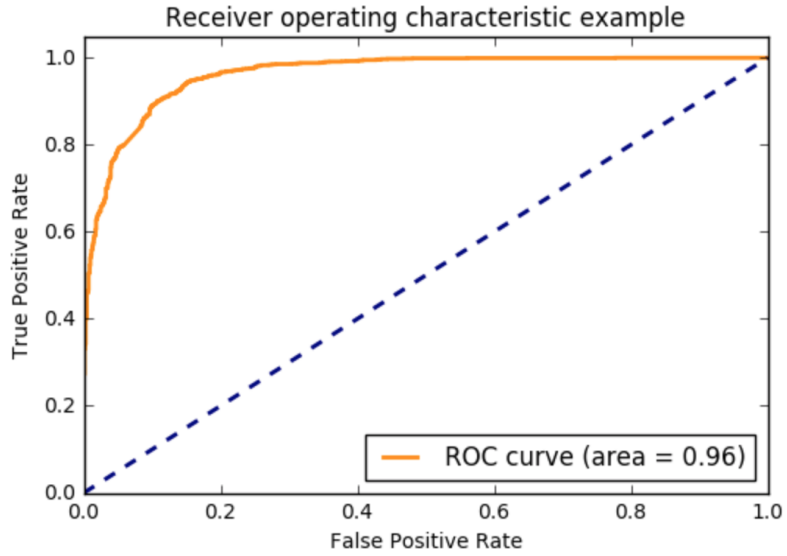

Figure 13 ROC curve of l-2 norm logistic regression classifier

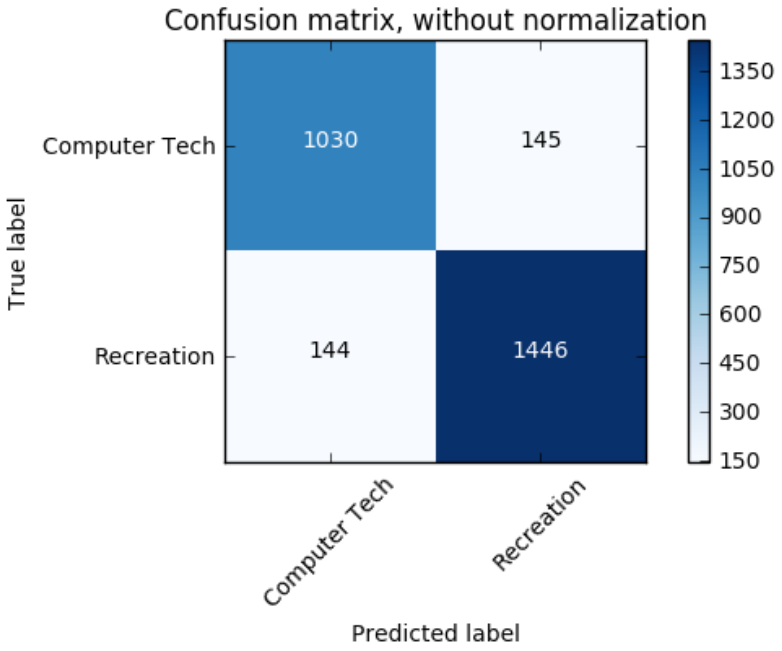

Figure 14 Confusion Matrix of l-2 norm logistic regression classifier, without normalization

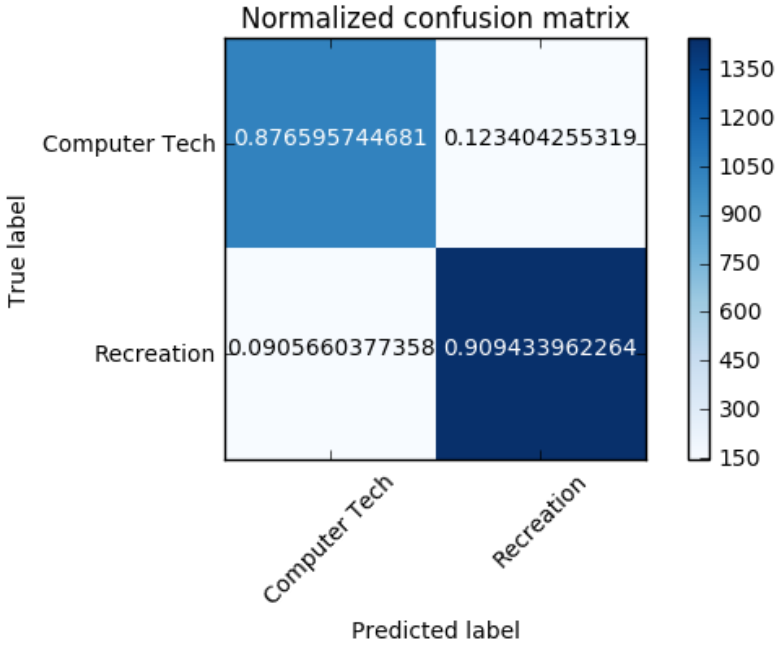

Figure 15 Confusion Matrix of l-2 norm logistic regression classifier, with normalization

Then we compute the Accuracy, Recall and Precision of this model. The results are shown in Table 7.

Table 7 Accuracy, Recall and Precision of 2 Norm Logistic Regression

Accuracy	Recall	Precision
0.89547920434	0.909433962264	0.908862350723

Below shows the confusion matrices associated with different coefficient values in a set of (0.01, 0.1, 1, 10, 100, 1000, 10000). According to the significant difference among those confusion matrices, we can conclude that these coefficients affect the result considerably, and also the fitted hyperplane is almost impossible to classify the data perfectly. What's more, it is never for sure that some coefficient can always have a better result compared to the rest, and it all depends on the occasions.

Table 8 Coefficient vs Confusion Matrix

Coefficient:	Confusion Matrix:
0.01	$\begin{pmatrix} 51 & 1124 \\ 0 & 1590 \end{pmatrix}$
0.1	$\begin{pmatrix} 742 & 433 \\ 5 & 1585 \end{pmatrix}$
1	$\begin{pmatrix} 879 & 296 \\ 31 & 1559 \end{pmatrix}$
10	$\begin{pmatrix} 875 & 300 \\ 60 & 1530 \end{pmatrix}$
100	$\begin{pmatrix} 859 & 316 \\ 77 & 1513 \end{pmatrix}$
1000	$\begin{pmatrix} 847 & 328 \\ 97 & 1493 \end{pmatrix}$
10000	$\begin{pmatrix} 849 & 326 \\ 116 & 1474 \end{pmatrix}$

Multiclass Classification

Question(j) Naïve Bayes classification and multiclass SVM classification

In this question, we build classifiers on the documents belonging to the classes mentioned in part b, which are ***'comp.sys.ibm.pc.hardware'***, ***'comp.sys.mac.hardware'***, ***'soc.religion.christian'*** and ***'misc.forsale'***

.

We first implement Naïve Bayes classification. The non-normalized confusion matrix and the normalized confusion matrix are shown as Figure 16 and Figure 17.

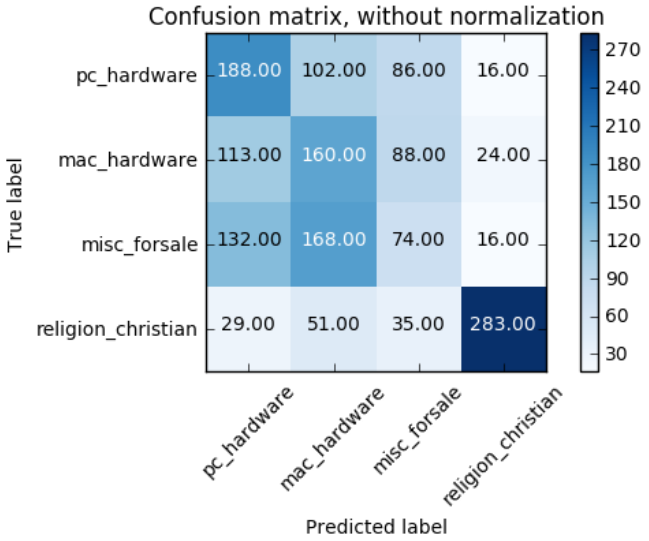

Figure 16 Confusion Matrix of Naïve Bayes Multiclass classifier, without normalization

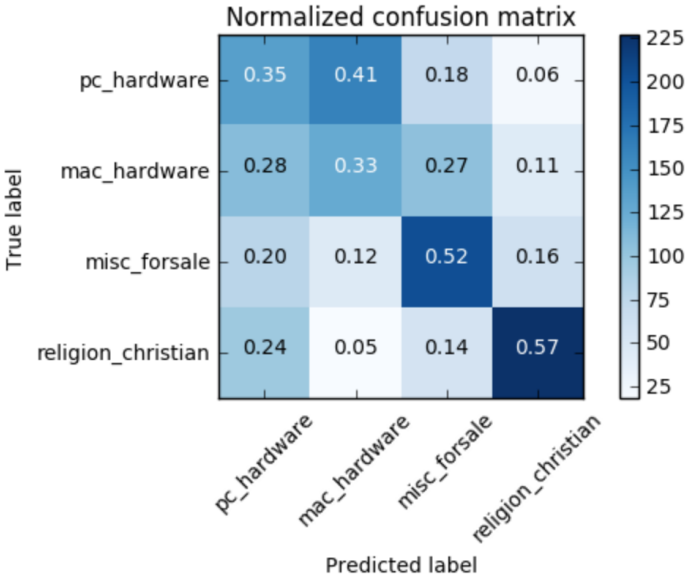

Figure 17 Confusion Matrix of Naïve Bayes Multiclass classifier, with normalization

Then we compute the Accuracy, Recall and Precision of this model. The results are shown in Table 9.

Table 9 Accuracy, Recall and Precision of Naïve Bayes Multiclass classification

Accuracy	Recall	Precision
0.450479233227	0.450479233227	0.461223108935

Then we further implement the multiclass SVM classification for the same task. With the method of One VS One, we get the non-normalized confusion matrix and the normalized confusion matrix shown as Figure 18 and Figure 19.

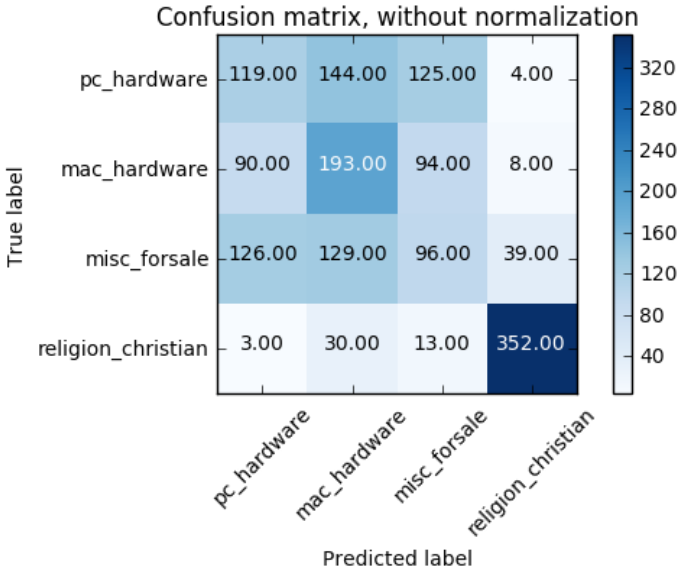

Figure 18 Confusion Matrix of 1 VS 1 multiclass SVM classifier, without normalization

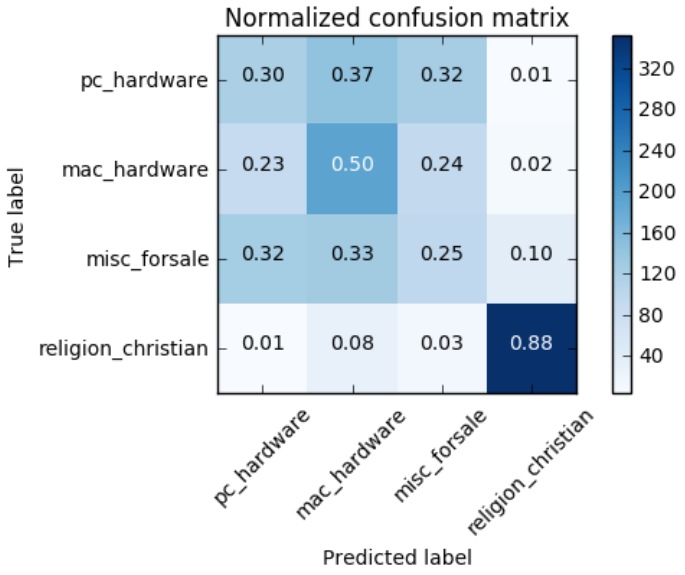

Figure 19 Confusion Matrix of One VS One multiclass SVM classifier, with normalization

The Accuracy, Recall and Precision of this model are shown in Table 10.

Table 10 Accuracy, Recall and Precision of One VS One multiclass SVM classification

Accuracy	Recall	Precision
0.215974440895	0.485623003195	0.478977251033

With the method of One VS the rest, we get the non-normalized confusion matrix and the normalized confusion matrix shown as Figure 20 and Figure 21.

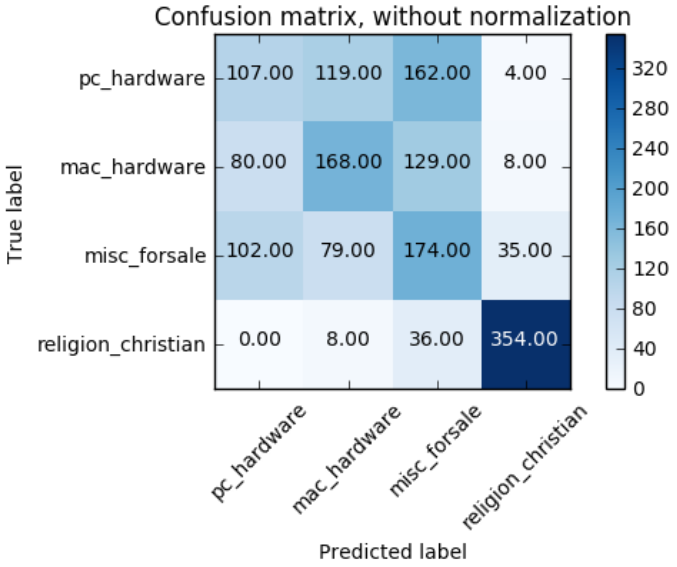

Figure 20 Confusion Matrix of 1 VS the rest multiclass SVM classifier, without normalization

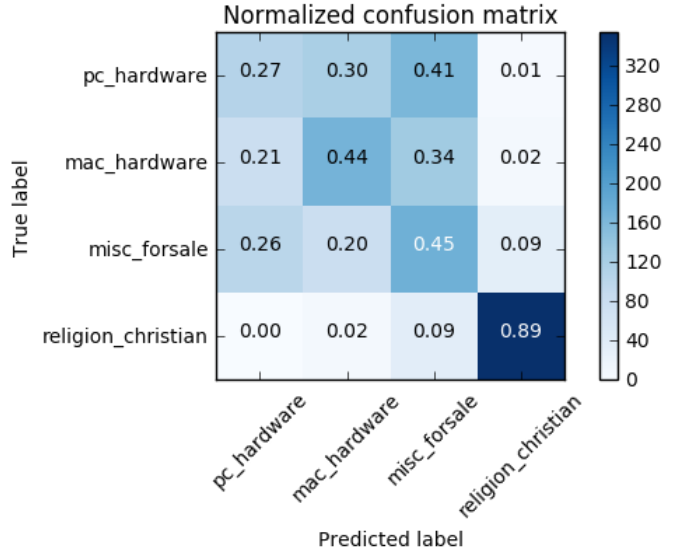

Figure 21 Confusion Matrix of 1 VS the rest multiclass SVM classifier, with normalization

The Accuracy, Recall and Precision of this model are shown in Table 11.

Table 11 Accuracy, Recall and Precision of One VS the rest multiclass SVM classification

Accuracy	Recall	Precision
0.36357827476	0.545686900958	0.598526401413