ECE 219 Large-Scale Data Mining Project 1

605033865 Sheng Yung Tao
704945153 Hua-En Li

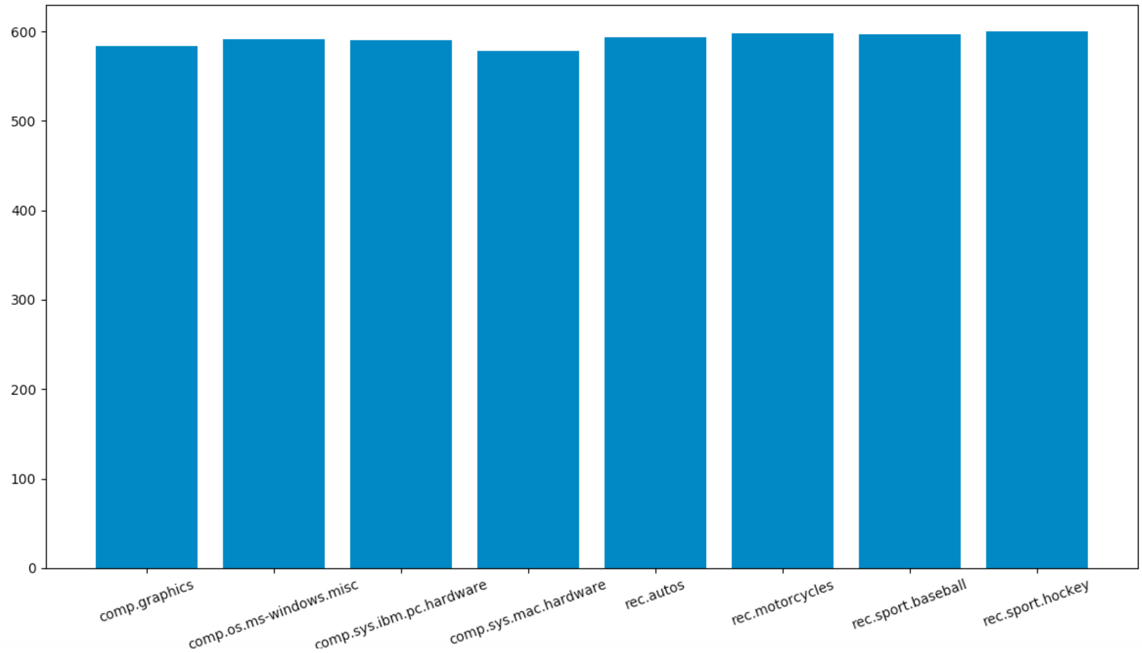
Implementation:

- Language: Python3.6.2
- Preprocess:
    - Store training and testing data/target of 8 classes/20 classes in the Class Data.
    - Use English stop_words to filter out stop words first, preventing from lemmatization breaks the stop words to tokens that cannot be recognized as stop words in TF-IDF
    - Remove punctuation
    - Use lemmatization to merge same words
    - Fit TF-IDF model with min_df=2 or 5, max_df =0.8, stop_words=English stop words.
    - Use LSI(SVD) and NMF to perform dimension reduction
- Calculate problem c, e-j
- Our confusion matrix is in the format:

	Predicted N	Predicted P
Actual N	True Negative	False Positive
Actual P	False Negative	True Positive

Result:

a) Plot histogram of 8 classes:



We can find that numbers of document in each class are almost the same; It’s a balanced dataset.

b) Final number of terms:

min_df = 2: 25915 terms
min_df = 5: 10512 terms

The result show that min_df can filter out some words that appear at an extreme low df. We can also find that there are about 15000 words that appear less than 5 times but more than twice. These words are barely going to help classification. Thus, we assume min_df =5 will perform better.

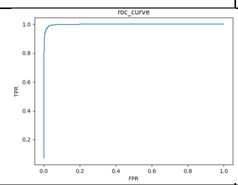
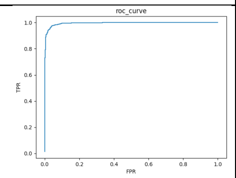
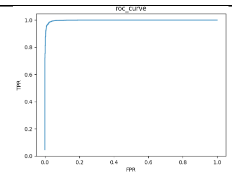
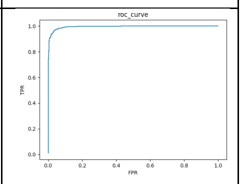
c) 10 most significant terms (for both min_df=2 and 5):

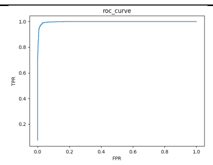
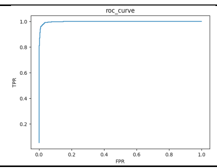
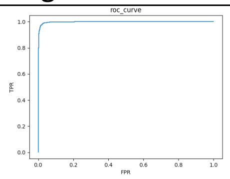
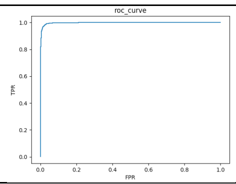
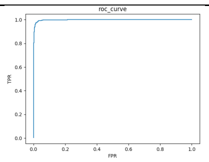
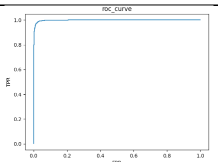
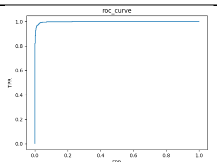
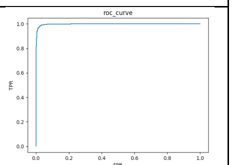
comp.sys.ibm.pc.hardware :
scsi, drive, ide, controller, card, disk, bios, scsi2, scsi1, bus
comp.sys.mac.hardware :
mac, apple, quadra, centris, drive, simms, problem, scsi, university, nubus
misc.forsale :
sale, new, university, nntppostinghost, offer, shipping, distribution, email, price, forsale
soc.religion.christian :
god, jesus, christian, church, people, christ, bible, say, think, faith

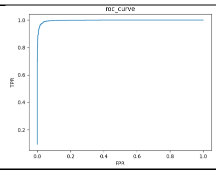
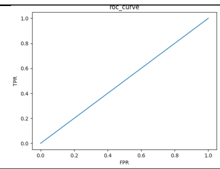
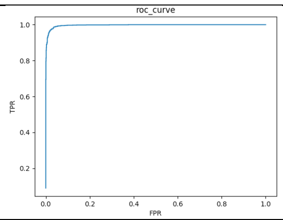
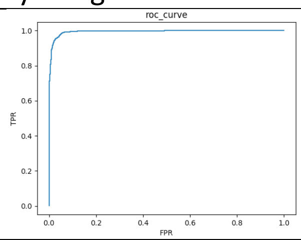
Because we filter out the stop words once at the very beginning, the stop words will not be stemmed and miss by the stop words in CountVectorizer. Thus, the result is pretty good with almost every word meaningful and correlated to the class title. If we do not filter out stop words firstly, “was” will be stemmed as “wa” and thus not recognized by CountVectorizer. This will let “wa” to be the most significant word for some class because it should be a stop word. The result of min_df =2 and 5 is the same, because min_df=[2,3,4] doesn’t seem to be able to be in the most significant terms.

d) successfully using LSI and NMF to reduce dimension.

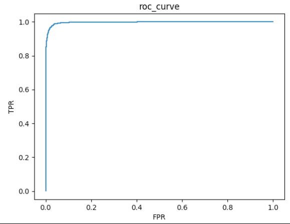
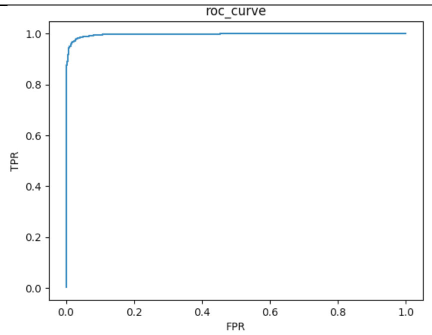
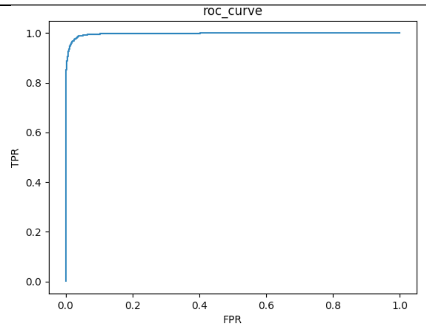
e-i)

	LSI			
Dim_reduction	min_df=2		min_df=5	
Part e): SVC				
C	C=1000	C=0.001	C=1000	C=0.001
ROC curve				
Confusion matrix	$\begin{bmatrix} 1507 & 53 \\ 15 & 1575 \end{bmatrix}$	$\begin{bmatrix} 1552 & 8 \\ 320 & 1270 \end{bmatrix}$	$\begin{bmatrix} 1512 & 48 \\ 20 & 1570 \end{bmatrix}$	$\begin{bmatrix} 1551 & 9 \\ 259 & 1331 \end{bmatrix}$
Accuracy	0.9784	0.8958	0.9784	0.9149
Recall	0.9905	0.7987	0.9874	0.8371
Precision	0.9674	0.9937	0.9703	0.9932
Part f): 5-fold cross validation				

Cross validation score	$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.5,0.5,0.968,0.976,0.978,0.977, 0.977]		$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.5,0.5,0.96,0.974,0.977,0.976,0.976]	
Best C:	C=10		C=10	
ROC curve				
Confusion matrix	$\begin{bmatrix}1513 & 47\\ 21 & 1569\end{bmatrix}$		$\begin{bmatrix}1511 & 49\\ 25 & 1565\end{bmatrix}$	
Accuracy	0.9784		0.9765	
Recall	0.9868		0.9842	
Precision	0.9709		0.9696	
Part h): Logistic Regression Classifier				
ROC curve				
Confusion matrix	$\begin{bmatrix}1505 & 55\\ 14 & 1576\end{bmatrix}$		$\begin{bmatrix}1509 & 51\\ 16 & 1574\end{bmatrix}$	
Accuracy	0.9781		0.9787	
Recall	0.9911		0.9899	
Precision	0.9662		0.9686	
Part i): regularization				
l1 error rate:	$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.5,0.07,0.05,0.03,0.023,0.021,0.021]		$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.5,0.07,0.06,0.03,0.022,0.0206, 0.0203]	
l2 error rate:	$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.29,0.05,0.3,0.028,0.025,0.022, 0.021]		$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.24,0.05,0.36,0.029,0.025,0.0219,0.0216]	
l1 average weight	$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0,0.11,1.1,4.15,10.9,19.19,21.75]		$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0,0.13,1.1,3.9,9.65,15.43, 16.58]	
l2 average weight	$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0,0.06,0.50,1.92,4.63,9.16,14.68]		$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0,0.07,0.53,1.89,4.44,8.60,13.3]	
	l1 best	l2 best	l1 best	l2 best
				
Confusion matrix	$\begin{bmatrix}1507 & 53\\ 14 & 1576\end{bmatrix}$	$\begin{bmatrix}1505 & 55\\ 14 & 1576\end{bmatrix}$	$\begin{bmatrix}1510 & 50\\ 14 & 1576\end{bmatrix}$	$\begin{bmatrix}1509 & 51\\ 16 & 1574\end{bmatrix}$
Accuracy	0.9787	0.9780	0.9796	0.9787
Recall	0.9911	0.9912	0.9911	0.9899
Precision	0.9674	0.9663	0.9692	0.9686

	NMF	
Part e): SVC		
C	C=1000	C=0.001
ROC curve		
Confusion matrix	$\begin{bmatrix}1500 & 60\\ 20 & 1570\end{bmatrix}$	$\begin{bmatrix}1560 & 0\\ 1590 & 0\end{bmatrix}$
Accuracy	0.9746	0.4952
Recall	0.9874	0
Precision	0.9632	1590/0=∞
Part f): 5-fold cross validation		
Cross validation scores	$\{10^k \mid -3 \le k \le 3, k \in Z\}$: [0.5,0.5,0.5,0.96,0.96,0.973,0.975]	
Best C	C=1000	
ROC curve		
Confusion matrix	$\begin{bmatrix}1500 & 60\\ 20 & 1570\end{bmatrix}$	
Accuracy	0.9746	
Recall	0.9874	
Precision	0.9631	
Part g): Naïve Bayes algorithm with NMF		
ROC curve		
Confusion matrix	$\begin{bmatrix}1412 & 148\\ 8 & 1582\end{bmatrix}$	
Accuracy	0.9504	
Recall	0.9949	
Precision	0.9144	

Part h): Logistic Regression Classifier

ROC curve		
Confusion matrix	$\begin{bmatrix}1494 & 66\\ 18 & 1572\end{bmatrix}$	
Accuracy	0.9733	
Recall	0.9886	
Precision	0.9597	
Part i) regularization		
l1 error rate	$\{10^k \| -3 \le k \le 3, k \in Z\}$: [0.5, 0.5, 0.3, 0.04, 0.028, 0.027, 0.026]	
l2 error rate	$\{10^k \| -3 \le k \le 3, k \in Z\}$: [0.49,0.49,0.09,0.04, 0.036, 0.031, 0.026]	
l1 average weight	$\{10^k \| -3 \le k \le 3, k \in Z\}$: [0, 0, 0.2, 15.68, 49.84, 101.72, 131.1]	
l2 average weight	$\{10^k \| -3 \le k \le 3, k \in Z\}$: [0, 0.04, 0.41, 3.18, 12.58, 29.56, 56.83]	
Best parameter:	1000	1000
ROC curve		
Confusion matrix	$\begin{bmatrix}1501 & 59\\ 25 & 1565\end{bmatrix}$	$\begin{bmatrix}1494 & 66\\ 18 & 1572\end{bmatrix}$
Accuracy	0.9733	0.9733
Recall	0.9842	0.9886
Precision	0.9636	0.9597

j) Multiclass classification

All the experiments in this part are conducted with min_df == 2 to meet the requirements. The confusion matrices below are for 4-class classification problem. Therefore, they are 4x4 matrices.

Class 1 : 'comp.sys.ibm.pc.hardware' , row and column1
Class 2: 'comp.sys.mac.hardware', row and column 2
Class 3: 'misc.forsale', row and column 3
Class 4: 'soc.religion.christian', row and column 4

The spec we use here: Rows are the true numbers of the classes, and columns are the predicted numbers of the classes. For example, the number in the (2, 3) entry means the number of data that is predicted as class 3 but actually class 2.

j) Multiclass classification	
Naïve Bayes with NMF data Classification	
Confusion matrix	$\begin{bmatrix} 335 & 14 & 40 & 3 \\ 118 & 224 & 37 & 6 \\ 66 & 15 & 300 & 9 \\ 1 & 0 & 3 & 394 \end{bmatrix}$
accuracy	0.801
Precision of each class	[0.644, 0.885, 0.789, 0.956]
Recall of each class	[0.855, 0.582, 0.769, 0.990]
One Vs One SVM with LSI data Classification	
Confusion matrix	$\begin{bmatrix} 345 & 28 & 19 & 0 \\ 51 & 319 & 14 & 1 \\ 28 & 14 & 346 & 2 \\ 6 & 0 & 4 & 388 \end{bmatrix}$
accuracy	0.893
Precision of each class	[0.802, 0.884, 0.903, 0.992]
Recall of each class	[0.880, 0.829, 0.887, 0.975]
One Vs Rest SVM with LSI data Classification	
Confusion matrix	$\begin{bmatrix} 340 & 30 & 22 & 0 \\ 47 & 317 & 20 & 1 \\ 25 & 13 & 347 & 5 \\ 4 & 0 & 1 & 393 \end{bmatrix}$
accuracy	0.893
Precision of each class	[0.817, 0.881, 0.890, 0.985]
Recall of each class	[0.867, 0.823, 0.890, 0.987]
One Vs One SVM with NMF data Classification	
Confusion matrix	$\begin{bmatrix} 316 & 46 & 30 & 0 \\ 73 & 283 & 28 & 1 \\ 48 & 15 & 325 & 2 \\ 1 & 0 & 17 & 380 \end{bmatrix}$
accuracy	0.833
Precision of each class	[0.721, 0.823, 0.813, 0.992]
Recall of each class	[0.806, 0.735, 0.833, 0.955]
One Vs Rest SVM with NMF data Classification	

Confusion matrix	$\begin{bmatrix} 316 & 45 & 28 & 3 \\ 73 & 282 & 25 & 5 \\ 44 & 14 & 326 & 6 \\ 1 & 0 & 5 & 392 \end{bmatrix}$
accuracy	0.841
Precision of each class	[0.728, 0.827, 0.849, 0.966]
Recall of each class	[0.806, 0.732, 0.836, 0.985]

Explanation:

e) SVC:

- Bigger C seems to perform better than lower one.
- Min_df doesn't matter a lot for LSI. It matters only the computation time.
- In NMF, model break down when C=0.001, this means that the SVM is too soft, the penalty for the error term is too small to make it classify correctly.

f) 5-fold cross validation:

- For LSI, min_df = 2 or 5 performs similarly. Both of them have best accuracy when C=10, which means that hard SVM basically performs better, but it may overfitting the training set when C is too large, even if we are using 5-fold cross validation.
- For NMF: C=1000 performs best. 5-fold cross validation will make the model more robust, even if big C tend to overfitting.

g) Naïve Bayes algorithm with NMF:

- The performance is ok, but I think it's a little bit strange to use Naïve Bayes with NMF because the content of NMF is no longer count of terms. That makes the Naïve Bayes algorithm work with meaningless input. The result has a little unbalanced score between precision and recall.

h) Logistic Regression Classifier:

- The model predicts pretty good result even without regularization (C very big,) and it seems to predict class "rec" better than other models.

i) Regularization based on Logistic Regression Classifier:

- The model performs better when C becomes bigger.
- We find that big C leads to best result, and the corresponding average absolute weight is also the biggest, which is as expected. Because big C means low strength of regularization, which will lead to high values of weights.
- l1 regularization: can result in sparse data of weight matrix W, thus have the same effect of feature selection.
- l2 regularization: efficient to compute because 2-norm is the distance in n dimension and has unique solution.

j) Multiclass Classification:

- The accuracy in both OneVsOne and OneVsRest SVM is higher than Naïve Bayes method.
- The precision of each class in both OneVsOne and OneVsRest SVM is higher than Naïve Bayes method.
- However, the recall of Class 1, 4 are high in Naïve Bayes method.

- The difference of the results between OneVsOne and OneVsRest SVM is not obvious.
- The accuracies of the SVM models using LSI data are higher than those using NMF data.
- The precisions of the SVM models using LSI data are higher than those using NMF data.
- The recalls of the SVM models using LSI data are higher than those using NMF data.