Project 1: Classification Analysis on Textual Data

Name: Jianxiong Wang, Yijun Wu, Yanzhao Wang, Yutong Sun

Date: 2019.1.19

Question 1

To get started, plot a histogram of the number of training documents for each of the 20 categories to check if they are evenly distributed.

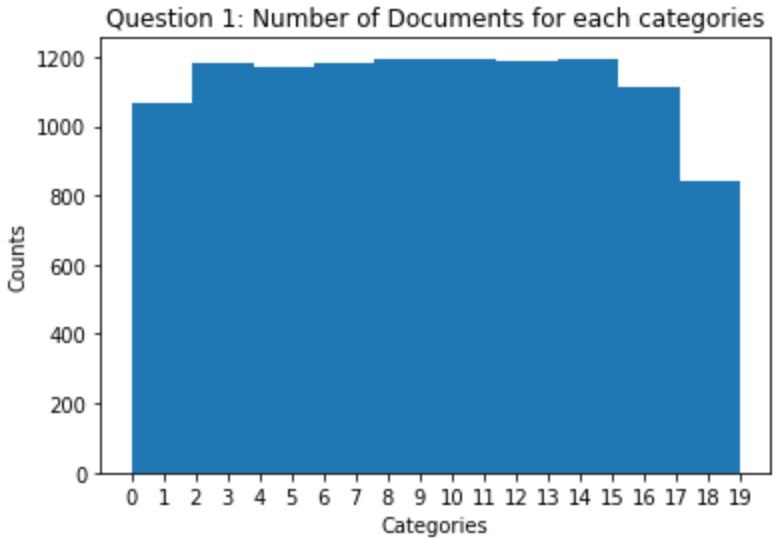


Figure 1: Histogram plots of all 20 categories.

Question 2

Use the following specs to extract features from the textual data:

- **Use the "english" stopwords of the CountVectorizer**
- **Exclude terms that are numbers (e.g. "123", "-45", "6.7" etc.)**
- **Perform lemmatization with nltk.wordnet.WordNetLemmatizer and pos_tag**
- **Use min_df=3**

Report the shape of the TF-IDF matrices of the train and test subsets respectively.

Training / Test	Matrix Shape
Training	(4732, 16600)
Test	(3150, 16600)

Question 3

Reduce the dimensionality of the data using the methods above

• Apply LSI to the TF-IDF matrix corresponding to the 8 categories with k = 50; so each document is mapped to a 50-dimensional vector.
• Also, reduce dimensionality through NMF (k = 50) and compare with LSI:

2 Which one is larger, the $||X - WH||_F^2$ in NMF or the $||X - U_k\Sigma_k V_k^T||_F^2$ in LSI? Why is the case?

NMF Training / Test	Matrix Shape	$\|\|X - WH\|\|_F^2$
Training	(4732, 50)	3940.5577
Test	(3150, 50)	2691.949

LSI Training / Test	Matrix Shape	$\|\|X - U_k\Sigma_k V_k^T\|\|_F^2$
Training	(4732, 50)	3895.6016
Test	(3150, 50)	2676.5911

Based on the tables shown above, NMF seems to have higher values on both the test and the training data, and LSI tends to better capture both the training data and the test data. Although NMF is more interpretable, the factorization is not unique, and the result is not guaranteed to be globally maximum, whereas LSI is guaranteed to be globally maximum.

Question 4

1.Train two linear SVMs and compare:

– Train one SVM with γ = 1000 (hard margin), another with γ = 0.0001 (soft margin).

– Plot the ROC curve, report the confusion matrix and calculate the accuracy, recall, precision and F-1 score of both SVM classifier. Which one performs better?

– What happens for the soft margin SVM? Why is the case?

2. Does the ROC curve of the soft margin SVM look good? Does this conflict with other metrics?

3. Use cross-validation to choose γ (use average validation accuracy to compare)

Hard SVM (C = 1000)	
Precision	0.9358
Recall	0.9911
Accuracy	0.961
F1 Score	0.9627

Confusion Matrix for Hard SVM	Predicted True	Predicted False
Condition True	1452	108
Condition False	14	1576

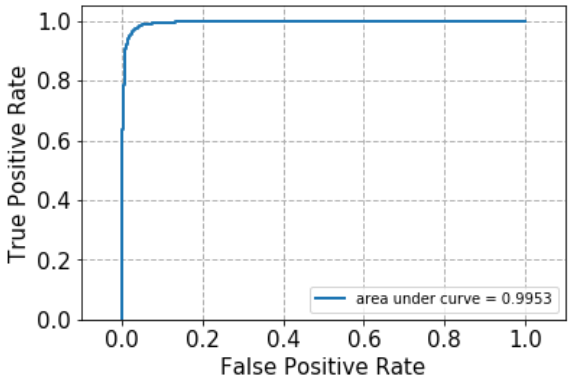


Figure 2: ROC curve of Hard SVM

Soft SVM (C = 0.0001)	
Precision	0.505
Recall	1.0
Accuracy	0.505
F1 Score	0.671

Confusion Matrix for Soft SVM	Predicted True	Predicted False
Condition True	2	1558
Condition False	0	1590

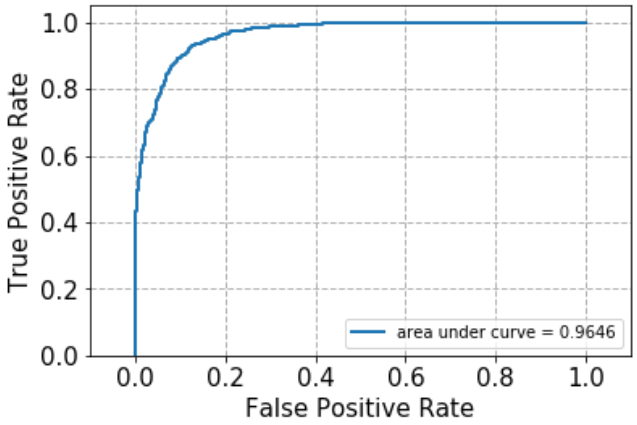


Figure 2: ROC curve of Soft SVM

Hard margin performs much better than that of soft margin. Since the penalty of the misclassification is very low for the soft margin, the classifier is likely to misclassify lots of data without getting a low score from its scoring function. In contrast, the hard margin puts lots of weight on misclassification, which results in better performance.

ROC Curve: Soft margin's ROC curve has less area than that of the hard margin. The more area the ROC curve has, the better the performance of the classifier. Therefore, this is consistent with the F1 score of these two classifiers.

Cross-Validation

Gamma	Accuracy
0.001	0.6325352112676057

0.01	0.6328169014084507
0.1	0.9643661971830987
1	0.9738028169014085
10	0.9759154929577466
100	0.975774647887324
1000	0.9749295774647887

Best Gamma = 100	
Precision	0.95482
Recall	0.9836
F1 Score	0.9690

Confusion matrix of the classifier with gamma = 100	Predicted True	Predicted False
Condition True	1486	74
Condition False	26	1564

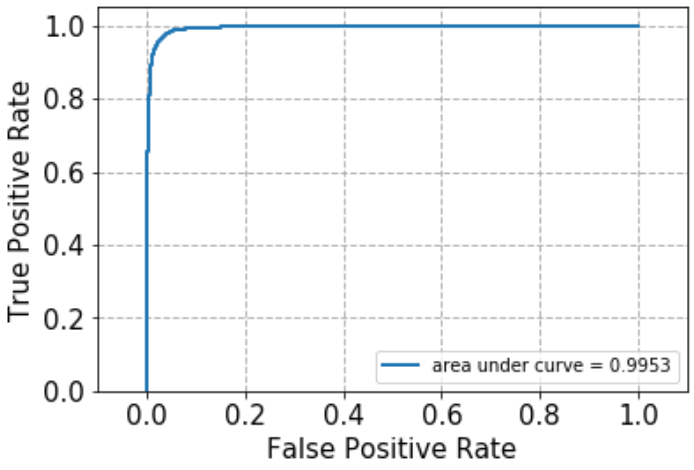


Figure 3: ROC curve of SVM with gamma = 100

Question 5

Train a logistic classifier without regularization (you may need to come up with some way to approximate this if you use sklearn.linear_model.LogisticRegression); plot the ROC curve and report the confusion matrix and calculate the accuracy, recall, precision and F-1 score of this classifier.

To perform logistic regression without regularization, we approximate it by using a very small regularization factor. We set C = 999999 in the LogisticRegression() package, note that higher C value means less emphasis on regularization.

Logistic Regression (No regularization)	
Precision	0.9628
Recall	0.9774
Accuracy	0.9695
F1 Score	0.9700

Confusion Matrix：

Without Regularization	Predicted True	Predicted False
Condition True	1500	60
Condition False	36	1554

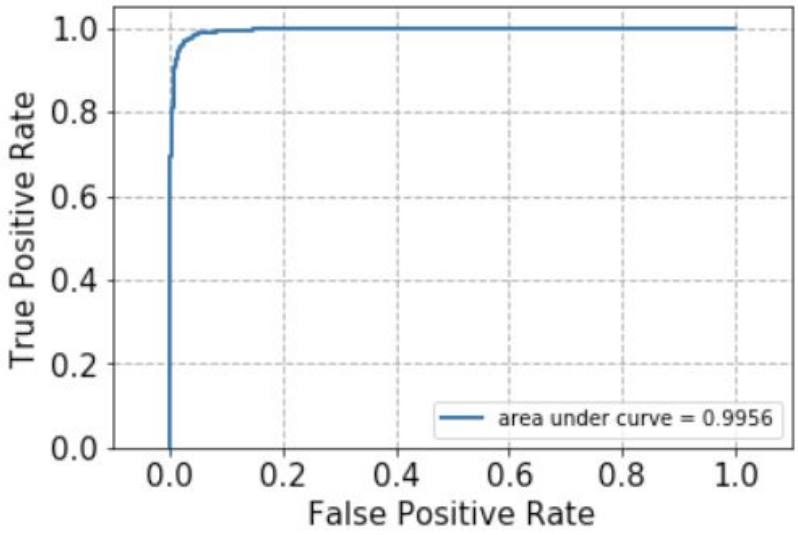


Figure 4: ROC curve of Logistic Regression without regularization

Using 5-fold cross-validation on the dimension-reduced-by-svd training data, find the best regularization strength in the range [-3, -2, -1, 0, 1, 2, 3] for logistic regression with L1 regularization and logistic regression L2 regularization, respectively.

Cross-Validation on Logistic Regression with L1-norm

C	Accuracy
0.001	0.4954929577464789
0.01	0.4954929577464789
0.1	0.9474647887323944
1	0.9701408450704225
10	0.9770422535211267
100	0.9770422535211267
1000	0.9749295774647887

Best C = 10

Precision	0.9604
Recall	0.9774
Accuracy	0.9683
F1 Score	0.9688

Confusion matrix:

L1-norm Regularization	Predicted True	Predicted False
Condition True	1496	64
Condition False	36	1554

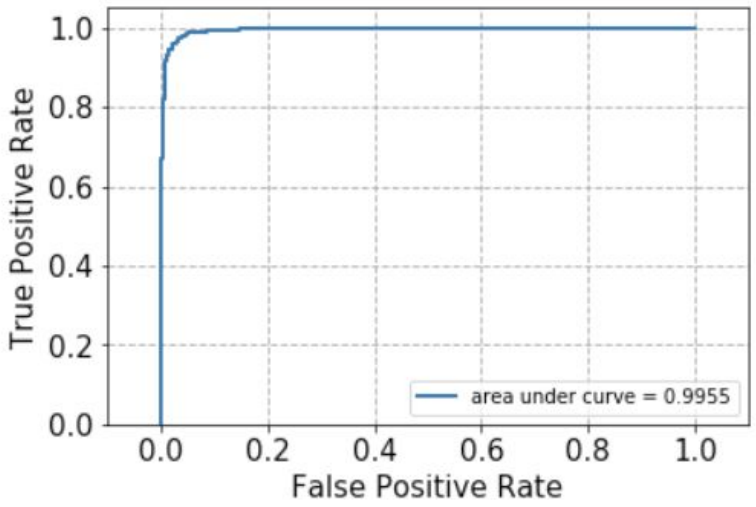


Figure 5: ROC curve of Logistic Regression with L1-Norm, C = 10

Cross Validation on Logistic Regression with L2-norm

C	Accuracy
0.001	0.6919718309859154
0.01	0.9025352112676057

0.1	0.9523943661971831
1	0.9698591549295775
10	0.9753521126760564
100	0.9769014084507044
1000	0.9771830985915495

Best C = 1000	
Precision	0.9616
Recall	0.9767
Accuracy	0.9686
F1 Score	0.9691

Confusion matrix:

L2-Norm Regularization	Predicted True	Predicted False
Condition True	1498	62
Condition False	37	1553

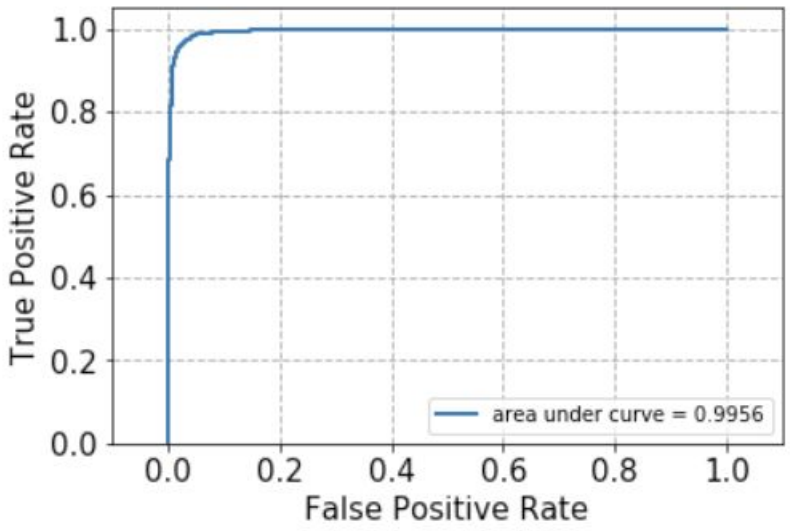


Figure 6: ROC curve of Logistic Regression with L2-Norm, C = 1000

Compare the performance (accuracy, precision, recall and F-1 score) of 3 logistic classifiers: w/o regularization, w/ L1 regularization and w/ L2 regularization (with the best parameters you found from the part above), using test data.

Regularization parameters affect test errors significantly. As seen from the test result, when regularization are highly emphasized (C = 0.001), the accuracy is less than 50% for L1-norm and less than 70% for L2-norm. When it is not emphasized (C = 1000), the accuracy of both algorithms is over 95%.

Since the best performance is produced by logistic regression with a small weight on regularization, the performance of the algorithm with three different types of regularization methods are approximately the same, all having a F1 score of about 0.969.

How does the regularization parameter affect the test error? How are the learnt coefficients affected? Why might one be interested in each type of regularization?

Appropriate regularization value can effectively reduce the test error caused by overfitting the training data. L1 regularization is known to be able to shrink the weight of less important features to 0. So it can serve as a way to select features. The weights of all features in L2 regularization do not decrease to 0 thus keeping all features in the algorithm.

Optimizing L1 regularization may produce multiple solutions and is therefore unstable. L2 regularization is guaranteed to have a unique solution.

Since L2 regularization puts more emphasis on data points with larger errors as it squares the error. The algorithm focuses more on outliers compared to L1 regularization.

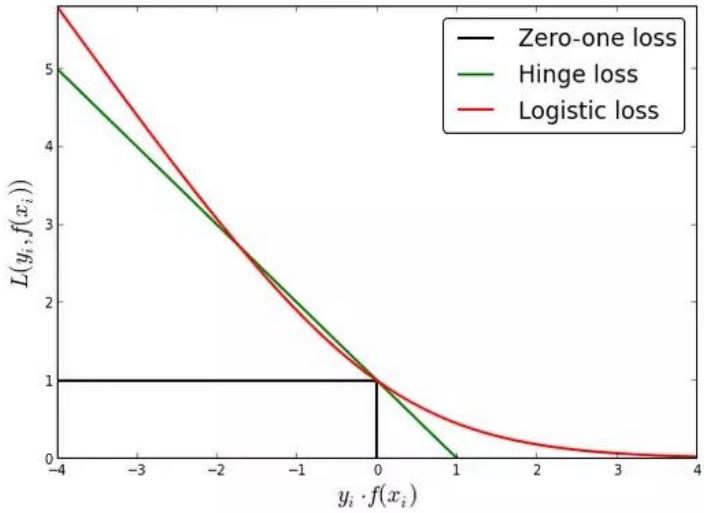
Both logistic regression and linear SVM are trying to classify data points using a linear decision boundary, then what's the difference between their ways to find this boundary? Why their performance differ?

The cost function of the two algorithms are similar,

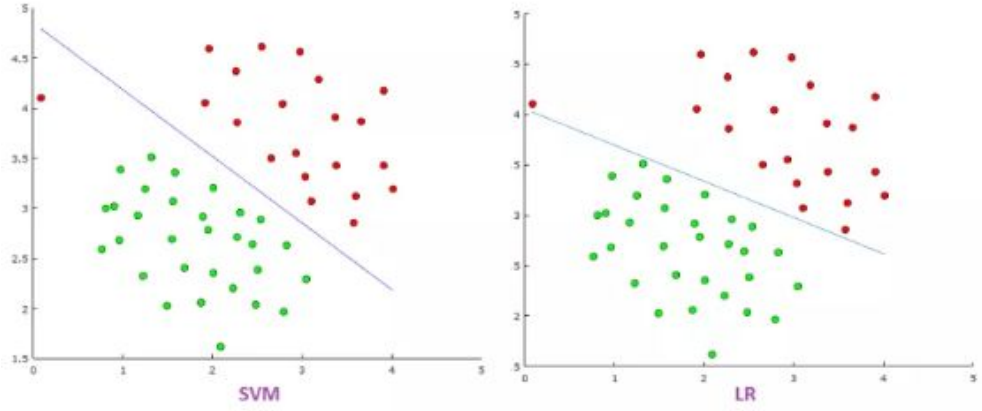
$$\min_w \lambda\|w\|^2 + \sum_i \max\{0, 1 - y_i w^T x_i\}$$

$$\min_w \lambda\|w\|^2 + \sum_i \log(1 + \exp(1 - y_i w^T x_i))$$

The first is the cost function for SVM (with regularization) and the second is for logistic regression. The only difference is the SVM minimizes the hinge loss while logistic regression minimizes the logistic loss.



The logistic loss diverges much faster than the hinge loss. This means that logistic regression would be more susceptible to outliers in the data set. The following figure is an illustration of how putting an outlier would affect logistic regression much more than SVM.



Linear SVM gives an output of 0/1 while logistic regression produces a value between 0 and 1. As a result, data points that are correctly classified do not contribute to the optimization of the loss function in SVM but do so in LR. This is a trade-off problem. SVM minimizes the error because correctly classified points have no error. But it might be the case that you care more about having an estimation of the probability, in which case logistic regression is preferred.

Question 6

Naïve Bayes classifier: train a GaussianNB classifier; plot the ROC curve and report the confusion matrix and calculate the accuracy, recall, precision and F-1 score of this classifier.

Gaussian Naive Bayes	
Precision	0.9027
Recall	0.9566
Accuracy	0.9206
F1 Score	0.9289

Confusion Matrix：

GNB	Predicted True	Predicted False
Condition True	1396	164
Condition False	69	1521

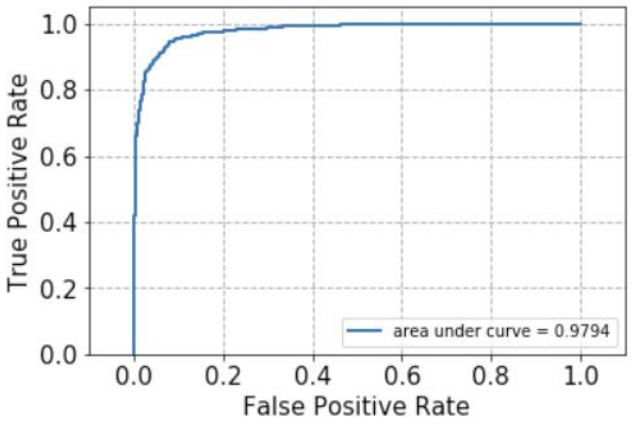


Figure 7: ROC curve of Gaussian Naive Bayes

Question 7

What is the best combination?

The grid search has these following parameters:

- Loading Data: remove “headers” and “footers” or not
- Minimum Frequency Threshold: Min_df = 3 or 5
- Dimension Reduction Method: LSI or NMF
- Algorithm: SVM with C = 100, Logistic Regression with L1 Regularization C = 10, Logistic Regression with L2 Regularization C = 1000, Gaussian Naive Bayes

The best combination is:

- Keeping “headers” and “footers”
- Min_df = 3
- Using lemmatization
- LSI
- Logistic Regression with L2 regularization C = 1000

Such a combination gives a cross-validation accuracy of 0.8727.
There are some interesting findings regard to each parameter if we hold other parameters constant:

- Loading Data: Removing headers and footers would decrease the accuracy for about 3-4%.
- Min_df: Different min_df values have little effect on accuracy, the difference is less than 0.5%.
- Lemmatization: Not using lemmatization give very poor performance, the accuracy is at about 45%.
- Dimension Reduction: SVD consistently outperforms NMF, by 4-6%in accuracy.
- Algorithm: SVM and both logistic regression produce about the same accuracy with less than 1% difference but outperforms Gaussian Naive Bayes by about 5%.

Question 8

Perform Naïve Bayes classification and multiclass SVM classification (with both One VS One and One VS the rest methods described above) and report the confusion matrix and calculate the accuracy, recall, precision and F-1 score of your classifiers.

The pipeline parameters are chosen to be the ones that give the best performance in Question 7, namely:

- Keeping “headers” and “footers”
- Min_df = 3

- Using lemmatization
- LSI

The gamma parameter is set to 100, the best value during cross-validation. Here are the results:

OneVsOne SVM

OneVsOne SVM	
Precision	0.8781
Recall	0.8780
Accuracy	0.8824
F1 Score	0.8780

Confusion Matrix (OvO SVM) :

OvO SVM	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Condition 1	314	54	21	3
Condition 2	44	320	1	0
Condition 3	18	20	349	3
Condition 4	1	2	4	391

OneVsRest SVM

OneVsRest SVM	
Precision	0.8780
Recall	0.8780
Accuracy	0.8869
F1 Score	0.8780

Confusion Matrix (OvR SVM) :

OvR SVM	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Condition 1	309	62	18	3
Condition 2	42	318	23	2
Condition 3	18	16	355	1
Condition 4	1	3	2	392

OneVsOne GNB

OneVsOne GNB	
Precision	0.7263
Recall	0.7265
Accuracy	0.7355
F1 Score	0.7264

Confusion Matrix (OvO GNB) :

OvO GNB	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Condition 1	282	32	44	34
Condition 2	136	144	52	53
Condition 3	39	21	320	8
Condition 4	0	0	6	392

OneVsRest GNB

OneVsRest GNB

Precision	0.7142
Recall	0.7182
Accuracy	0.7304
F1 Score	0.7162

Confusion Matrix (OvR GNB) :

OvR GNB	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Condition 1	264	34	53	41
Condition 2	116	148	64	57
Condition 3	36	26	320	8
Condition 4	0	0	6	392

Reference

Drakos, G., & Drakos, G. (2018, August 12). Support Vector Machine vs Logistic Regression – Towards Data Science. Retrieved from https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f

Chokia. L. (2013, Dec 1). Differences between the L1-norm and the L2-norm. Retrieved from http://www.chioka.in/differences-between-the-l1-norm-and-the-l2-norm-least-absolute-deviations-and-least-squares/