Krish Kabra, 404593821
Ranjana Vittal, 605629219
Zhiyuan Chen, 605363281

Project 1: Classification Analysis on Textual Data

EE 219 - Large-Scale Data Mining: Models and Algorithms
Winter 2021

2021-01-20

Getting Familiar With the Dataset

Question 1: Visualizing the Dataset

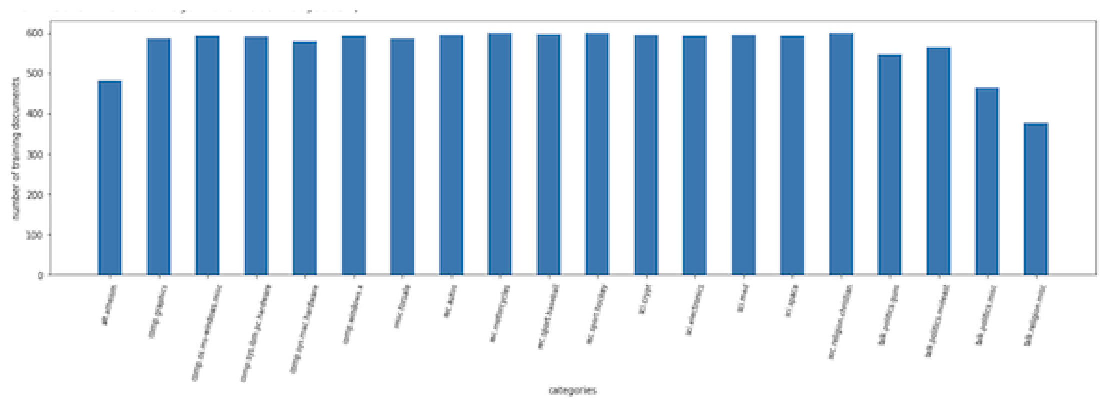


Figure 1: Training document number for each categories

From the histogram in Figure 1, we can see that for each of the 20 categories, the number of training documents are evenly distributed.

Binary Classification

Question 2: Feature Extraction

For feature extraction, we excluded punctuations and numbers, applied min_df=3 and english stopwords. The resulting shape of the TF–IDF matrices are:

Shape of TF–IDF train matrix = (4732, 17671)
Shape of TF–IDF test matrix = (3150, 17671)

Question 3: Dimensionality Reduction

Error for LSI: 4116.55988709079
Error for NMF: 4154.185519146454

The error for NMF is larger. This is because the NMF optimization objectives have more constraints than LSI. The added constraints are $W \geq 0$ and $H \geq 0$. This reduces the degrees of freedom and makes the search space more restrictive. On the other hand, LSI has a much larger search space to explore. This means that it has more freedom to find the best vectors that minimize the error and the best vectors can include negative elements. This is not possible in NMF.

Question 4: Support Vector Machines

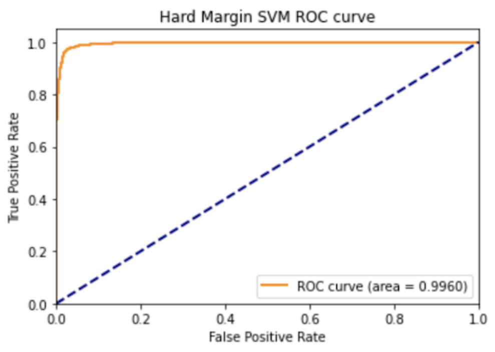


Figure 2: ROC Curve of Hard Margin SVM

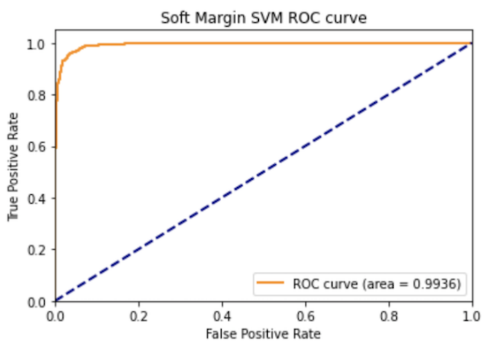


Figure 3: ROC Curve of Soft Margin SVM

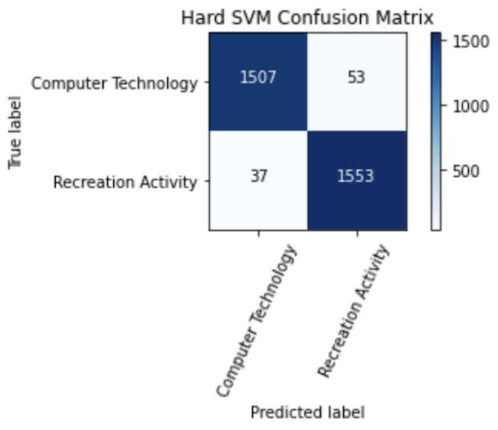


Figure 4: Confusion Matrix of Hard Margin SVM

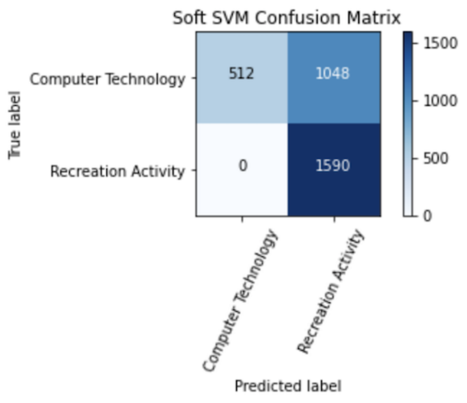


Figure 5: Confusion Matrix of Soft Margin SVM

	Soft SVM	Hard SVM
Accuracy Score	0.667302	0.971429
Recall Score	1.000000	0.976730
Precision Score	0.602729	0.966999
F1 Score	0.752129	0.971840

Table 1: Score of Both SVM Classifier

Hard Margin SVM performs better than Soft Margin SVM.

The ROC curve for the Soft Margin SVM and recall score look good. This conflicts with the other metrics which are in the range of 0.6 – 0.75. This might be because the Soft Margin SVM has found a separating hyperplane whose intercept term b is not adjusted properly. Thus, the SVM may have found the optimal w vector but not the b term. This means that the margin may not be fully maximized.

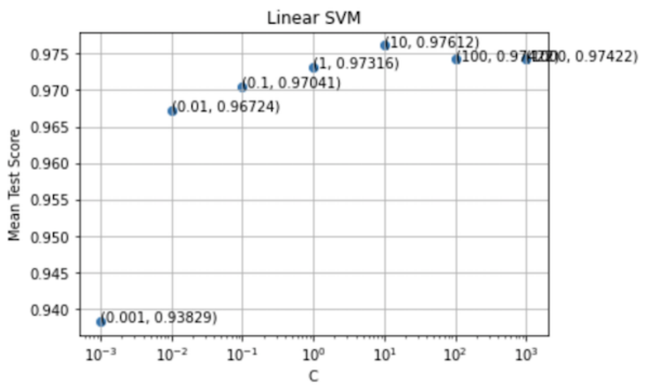


Figure 6: Linear SVM

After running 5-fold cross-validation, the best value for Gamma is 10. The best accuracy score achieved by a linear SVM on the training dataset with this regularization parameter was 0.9761193130191927.

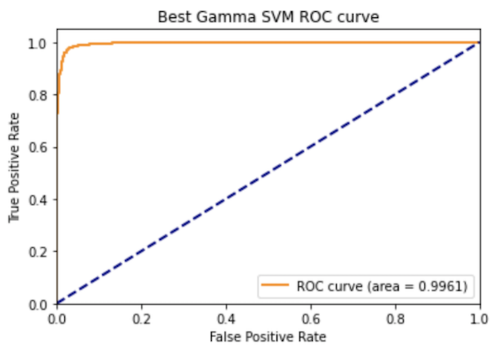


Figure 7: Best Gamma SVM ROC Curve

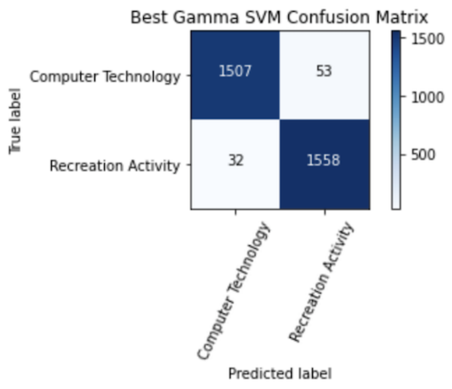


Figure 8: Best Gamma SVM Confusion Matrix

Accuracy Score	0.973016
Recall Score	0.979874
Precision Score	0.967101
F1 Score	0.973446

Table 2: Score of the Best SVM $\gamma = 10$

Question 5: Logistic Classification

Logistic Regression without Regularization:

Accuracy Score	0.971111
Recall Score	0.976101
Precision Score	0.966978
F1 Score	0.971518

Table 3: Score of Logistic Regression without regularization

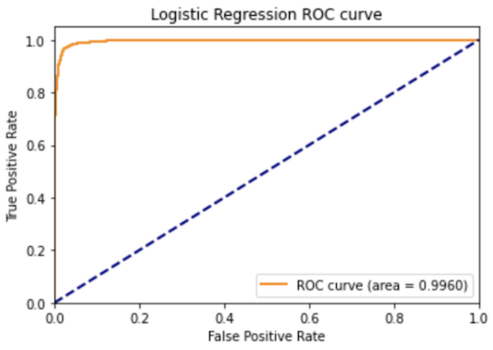


Figure 9: Logistic Regression ROC Curve

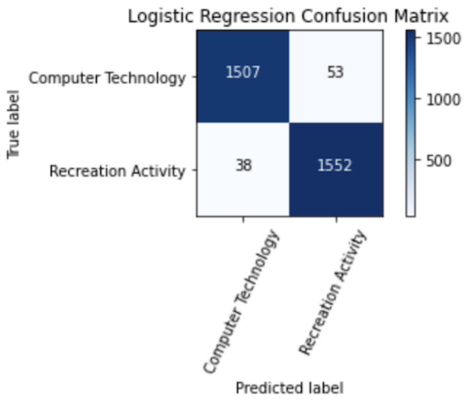


Figure 10: Logistic Regression Confusion Matrix

Logistic Regression with L2 Regularization:

After running 5-fold cross-validation for logistic regression with L2 regularization, the best value of Gamma is 100. The best accuracy score with this regularization parameter on the training set was 0.9763307295096789.

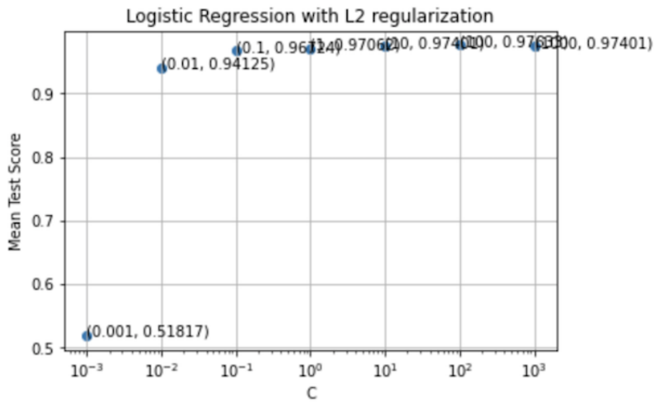


Figure 11: Logistic Regression with L2 Regularization

Accuracy Score	0.972698
Recall Score	0.980503
Precision Score	0.965923
F1 Score	0.973159

Table 4: Score of Logistic Regression with L2 Regularization: (γ = 100)

Logistic Regression with L1 Regularization:

After running 5-fold cross-validation for logistic regression with L1 regularization, the best value for Gamma is 10. The best accuracy score with this regularization parameter on the training set was 0.9752740935545876.

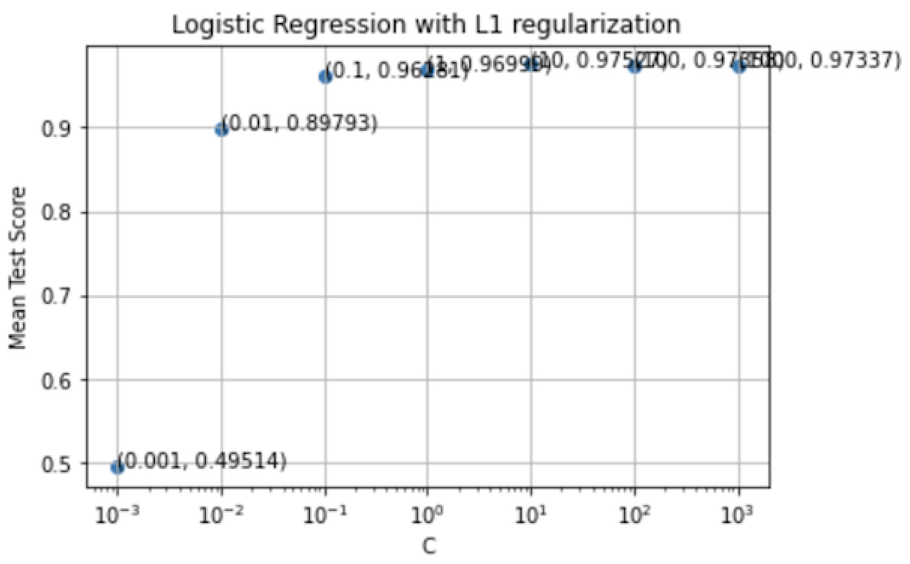


Figure 12: Logistic Regression with L1 Regularization

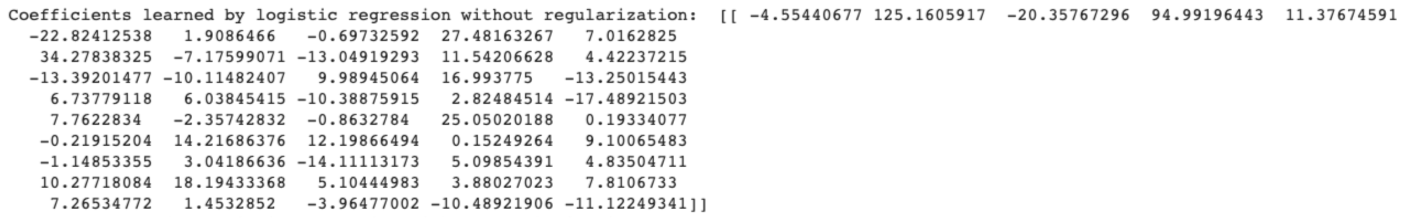
Accuracy Score	0.972063
Recall Score	0.979245
Precision Score	0.965881
F1 Score	0.972517

Table 5: Score of Logistic Regression with L1 Regularization: (γ = 10)

Logistic regression with L2 regularization performs the best because it has the highest accuracy and F1 score. Logistic regression without any regularization performs the worst because it scores the lowest scores in all categories. Logistic regression with L1 regularization performs satisfactorily. It is better than logistic regression without regularization but worse than logistic regression with L2 regularization.

The regularization parameter decreases the test error in both L1 and L2 regularization.

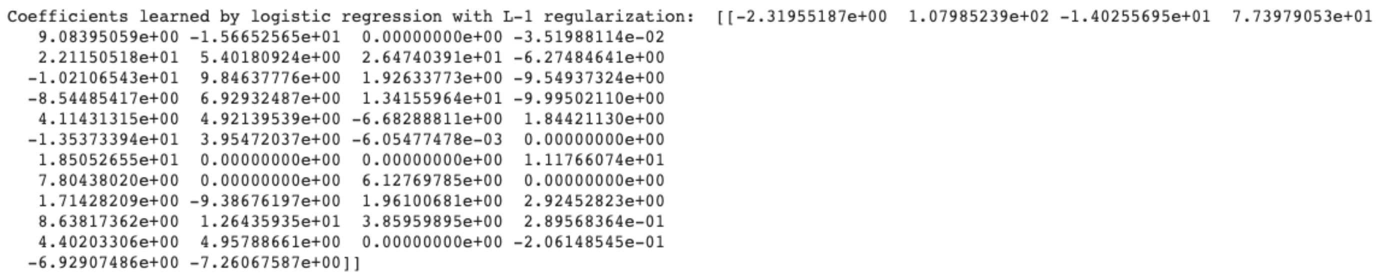
No regularization: The learned coefficients tend to be quite large, usually of 10^1 order of magnitude. This makes the model volatile which is a sign of overfitting. This type of regularization is useful to build a complex model to deal with complex data and there are no signs of overfitting.



```
Coefficients learned by logistic regression without regularization:  [[ -4.55440677 125.1605917  -20.35767296  94.99196443  11.37674591
  -22.82412538   1.9086466   -0.69732592  27.48163267   7.0162825
   34.27838325  -7.17599071 -13.04919293  11.54206628   4.42237215
  -13.39201477 -10.11482407   9.98945064  16.993775   -13.25015443
    6.73779118   6.03845415 -10.38875915   2.82484514 -17.48921503
    7.7622834   -2.35742832  -0.8632784   25.05020188   0.19334077
   -0.21915204  14.21686376  12.19866494   0.15249264   9.10065483
   -1.14853355   3.04186636 -14.11113173   5.09854391   4.83504711
   10.27718084  18.19433368   5.10444983   3.88027023   7.8106733
    7.26534772   1.4532852   -3.96477002 -10.48921906 -11.12249341]]
```

Figure 13: Coefficients learned by Logistic Regression without regularization

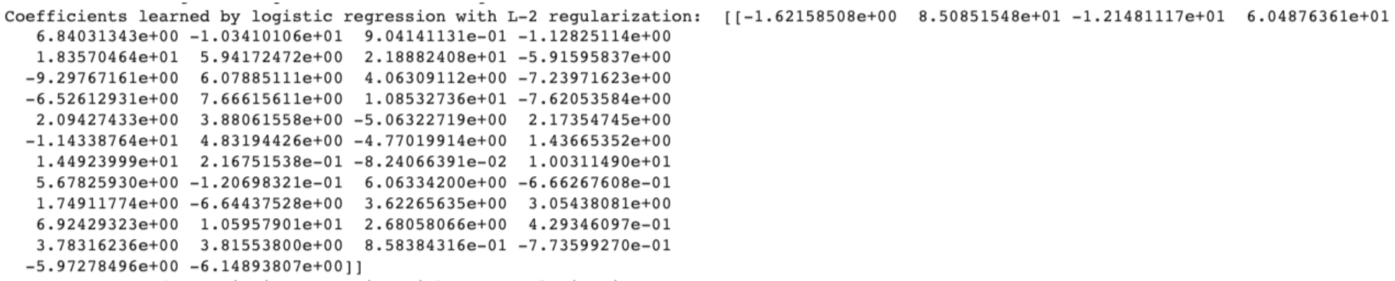
L1 regularization: Some of the learned coefficients are 0 and most of the learned coefficients have 10^0 order of magnitude which is smaller than the learned coefficients of logistic regression without any regularization. This type of regularization is useful to build a sparse model by removing features. This results in a process called feature selection which keeps the most significant features in the model.



```
Coefficients learned by logistic regression with L-1 regularization:  [[-2.31955187e+00  1.07985239e+02 -1.40255695e+01  7.73979053e+01
   9.08395059e+00 -1.56652565e+01  0.00000000e+00 -3.51988114e-02
   2.21150518e+01  5.40180924e+00  2.64740391e+01 -6.27484641e+00
  -1.02106543e+01  9.84637776e+00  1.92633773e+00 -9.54937324e+00
  -8.54485417e+00  6.92932487e+00  1.34155964e+01 -9.99502110e+00
   4.11431315e+00  4.92139539e+00 -6.68288811e+00  1.84421130e+00
  -1.35373394e+01  3.95472037e+00 -6.05477478e-03  0.00000000e+00
   1.85052655e+01  0.00000000e+00  0.00000000e+00  1.11766074e+01
   7.80438020e+00  0.00000000e+00  6.12769785e+00  0.00000000e+00
   1.71428209e+00 -9.38676197e+00  1.96100681e+00  2.92452823e+00
   8.63817362e+00  1.26435935e+01  3.85959895e+00  2.89568364e-01
   4.40203306e+00  4.95788661e+00  0.00000000e+00 -2.06148545e-01
  -6.92907486e+00 -7.26067587e+00]]
```

Figure 14: Coefficients learned by Logistic Regression with L-1 regularization

L2 regularization: Most of the learned coefficients have 10^0 order of magnitude which is smaller than the learned coefficients of logistic regression without any regularization. This type of regularization is useful when we would want the learned coefficients to be small so that the model is not heavily dependent on a few features. This means that the model is not sensitive to small changes in the feature vectors which makes the model more robust and stable.



```
Coefficients learned by logistic regression with L-2 regularization:  [[-1.62158508e+00  8.50851548e+01 -1.21481117e+01  6.04876361e+01
   6.84031343e+00 -1.03410106e+01  9.04141131e-01 -1.12825114e+00
   1.83570464e+01  5.94172472e+00  2.18882408e+01 -5.91595837e+00
  -9.29767161e+00  6.07885111e+00  4.06309112e+00 -7.23971623e+00
  -6.52612931e+00  7.66615611e+00  1.08532736e+01 -7.62053584e+00
   2.09427433e+00  3.88061558e+00 -5.06322719e+00  2.17354745e+00
  -1.14338764e+01  4.83194426e+00 -4.77019914e+00  1.43665352e+00
   1.44923999e+01  2.16751538e-01 -8.24066391e-02  1.00311490e+01
   5.67825930e+00 -1.20698321e-01  6.06334200e+00 -6.66267608e-01
   1.74911774e+00 -6.64437528e+00  3.62265635e+00  3.05438081e+00
   6.92429323e+00  1.05957901e+01  2.68058066e+00  4.29346097e-01
   3.78316236e+00  3.81553800e+00  8.58384316e-01 -7.73599270e-01
  -5.97278496e+00 -6.14893807e+00]]
```

Figure 15: Coefficients learned by Logistic Regression with L-2 regularization

Logistic regression is a probabilistic classifier whereas SVM is a deterministic classifier. Logistic regression uses all the data points to find the decision boundary. However, SVM uses only a small subset of the data points called support vectors to find the decision boundary. Moreover, logistic regression finds a decision boundary without considering the position of the hyperplane. In contrast, the SVM tries to find the position of the hyperplane which maximizes the margin (the distance from the support vectors). In this way, the SVM finds a better discriminating hyperplane than logistic regression. Thus, SVM has better performance than logistic regression.

Question 6: Naive Bayes Classification

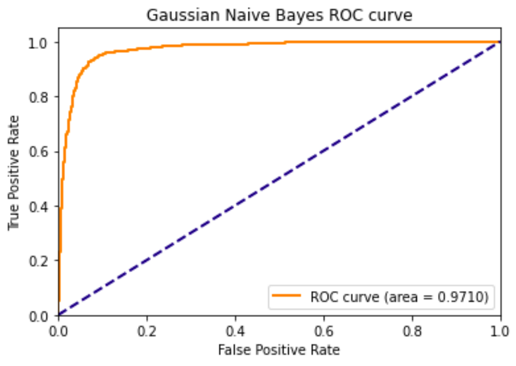


Figure 16: Gaussian Naive Bayes ROC Curve

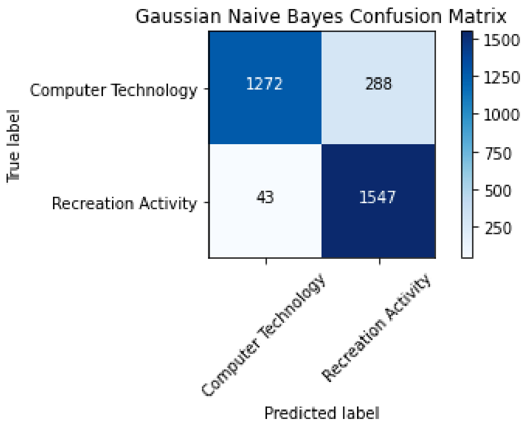


Figure 17: Gaussian Naive Bayes Confusion Matrix

Accuracy Score	0.894921
Recall Score	0.972956
Precision Score	0.843052
F1 Score	0.903358

Table 6: Score of Gaussian Naive Bayes Classifier

Question 7: Grid Search of Parameters

Loading Data	Keep headers and footers
Feature Extraction	min_df=3, Use Lemmatization
Dimensionality Reduction	LSI
Classifier	Linear SVM (γ = 10)
Accuracy Score	0.973016
Recall Score	0.979874
Precision Score	0.967101
F-1 Score	0.973446

Table 7: Pipeline 1 - Best Combination of Pipeline

Loading Data	Remove headers and footers
Feature Extraction	min_df=3, Use Lemmatization
Dimensionality Reduction	LSI
Classifier	Logistic Regression (γ = 10)
Accuracy Score	0.965397
Recall Score	0.979245
Precision Score	0.953460
F-1 Score	0.966181

Table 8: Pipeline 2

(a) min_df does not have much influence on the test score, because the test scores when we choose min_df=3 or min_df=5 are very close.

(b) Lemmatization has a great influence on the test score. It improves significantly when we use lemmatization than not.

(c) In most cases, the test score is higher when we use LSI to reduce dimensionality than using NMF. But if we choose GaussianNB classifier and use lemmatization, NMF is better than LSI.

(d) As for different classifiers, the performance of GaussianNB is the worst and the performance of the other three classifiers is very close.

(e) The performance improves a little when we do not remove “headers” and “footers” than we do.

Pipeline 1 with Keeping Headers and Footers, min_df=3, Using Lemmatization, LSI, Linear SVM with γ = 10 is the best combination with test accuracy score of 0.973016.

Binary Classification

Question 8: Introduction to GLoVE

(a) Training a model using only the occurrence probability prevents the model from learning the relationship between two words that commonly appear in the same category, thereby causing such models to perform relatively poorly on the word analogy task. On the other hand, training a model using co-occurrence probabilities enables the model to learn word similarity. For example, the words "environment" and "pollution" may occur frequently together in a document, implying some correspondence between these two words, however, the occurrence probability model would fail to capture this information.

(b) Yes – GLoVE embeddings capture global word representations without knowledge of the context the word is. The embedding for each word is fixed after training, irrespective of the context the word appears in.

(c) "King is to queen as man is to woman".
$||(GLoVE[\text{"}queen\text{"}] - GLoVE[\text{"}king\text{"}]) - (GLoVE[\text{"}wife\text{"}] - GLoVE[\text{"}husband\text{"}])||_2 \approx 0$
as $||GLoVE[\text{"}queen\text{"}] - GLoVE[\text{"}king\text{"}]||_2 \approx ||GLoVE[\text{"}wife\text{"}] - GLoVE[\text{"}husband\text{"}]||_2$.
Interestingly, it may be that $||GLoVE[\text{"}queen\text{"}]-GLoVE[\text{"}king\text{"}]||_2 > ||GLoVE[\text{"}wife\text{"}]-GLoVE[\text{"}husband\text{"}]||_2$ as the words "king" and "queen" are used less often together in a document than "wife" and "husband", perhaps because typically there is only one leading authority in the form of a king or queen, never two at the same time.

(d) Given a word, you would rather lemmatize over stem the word before mapping it to its GLoVE embedding. Stemming the word without considering the context the word is used in may result in a final word of vastly different meaning eg. "stripes" $\rightarrow$ "strip".

Question 9: Feature Engineering with GLoVE

(a) The feature engineering process is contained within the function `features_from_glove(input_data, embeddings_dict, dim_embedding)`. For each document, the words corresponding to the "Keywords" and/or "Subject" header lines are collected with punctuation and numbers removed. The words are then lemmatized as it was empirically found that not lemmatizing and removing stop words reduced performance. The GLoVE embeddings of each word are then aggregated (i.e. summed) to obtain a vector with length equal to the dimension of the GLoVE embedding used. This vector is normalized to 1.

(b) To evaluate our GLoVE-based feature model, we used a SVM. The margin hyperparameter was chosen to be the best parameter from the list $\gamma = \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ using 5-fold cross-validation grid search that optimizes for the highest validation accuracy. We achieve overall testing accuracy of 90.3%. The confusion matrix and ROC are given in Figures 18 and 19 respectively.

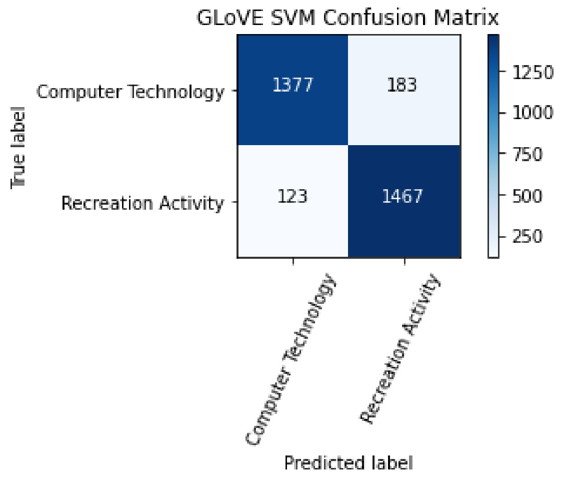


Figure 18: Confusion matrix for the binary classification task using GLoVE embeddings.

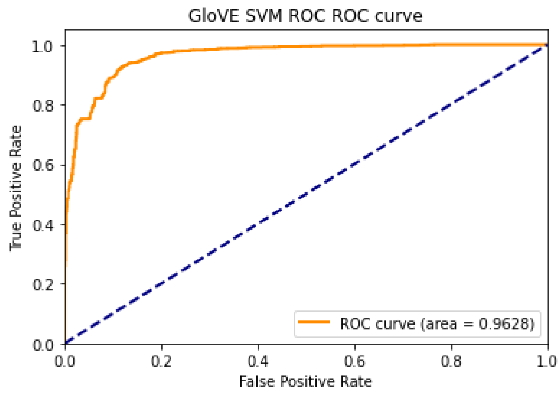


Figure 19: ROC for binary classification task using GLoVE embeddings.

Question 10: GLoVE Results Analysis Part 1

It is expected that the classification accuracy increases with higher dimensions of the pre-trained GLoVE embedding used, as more information is represented in the dimensions. This trend is observed in Figure 20, up to an embedding of dimension 200. After this, the classification accuracy reduces using a GLoVE embedding

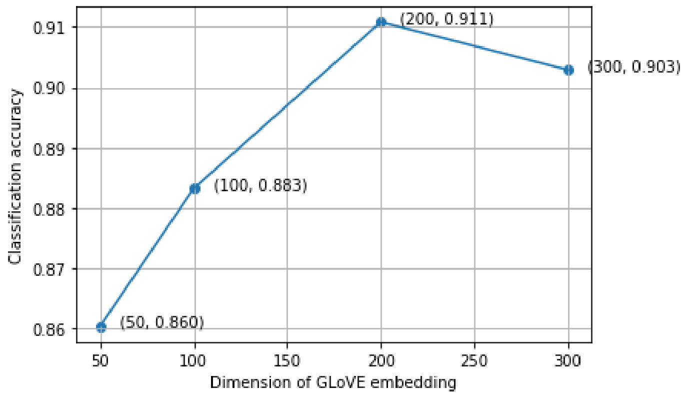


Figure 20: Relationship between the dimension of pre-trained GLoVE embeddings and accuracy of binary classification model.

dimension of 300.

A suggestion for this reduction is because of the simplistic nature of the feature engineering process. Simply aggregating the word embedding vectors of a phrase or sentence prevents context that the word is being used in to be represented in the final feature vector obtained. For lower dimensional embedding, this simplicity may not be an issue. However, for higher dimensional embeddings, this simplicity may cause less meaningful features to aggregate, making classification more difficult. Ideally, a weighted average of word embedding vectors should be used to represent the phrase or sentence, where weights depend on the context the word is used in.

Question 11: GLoVE Results Analysis Part 2

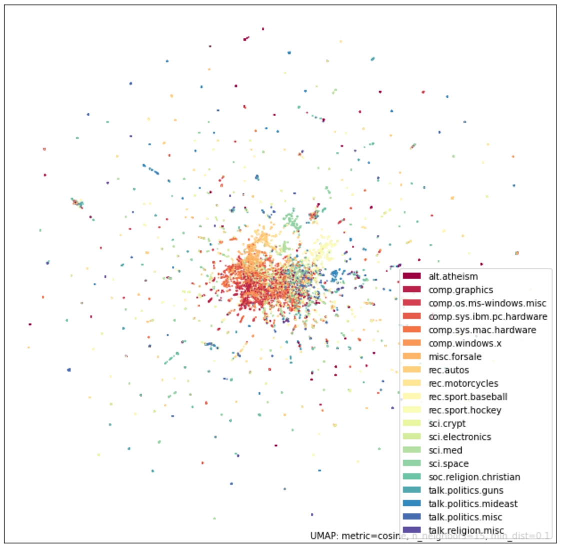


Figure 21: UMAP visualization of GLoVE-based embeddings of the documents with their labels.

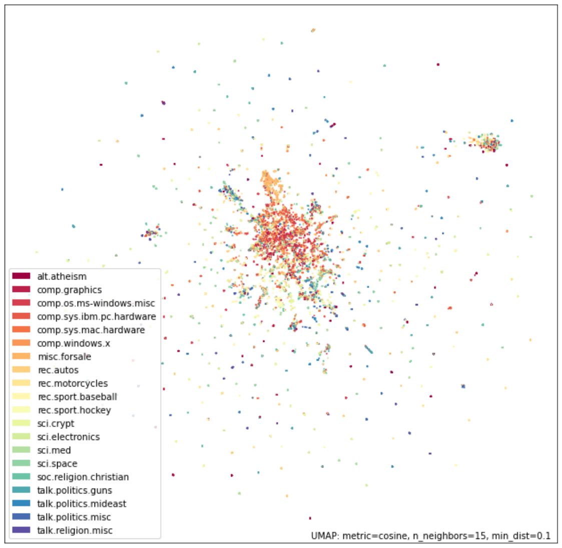


Figure 22: UMAP visualization of random-based embeddings of the documents with their labels.

Figure 21 visualizes the set of normalized GLoVE-based embedding of the documents with their labels in a 2D plane using the UMAP library. A cosine similarity metric is used to compute distance between documents in feature space. To contrast this visualization, figure 22 uses a set of normalized random vectors of the same dimension as GLoVE as the word embedding.

The two maps clearly differ in how document labels are clustered. The GLoVE-based embedding map has most documents located centrally. The map shows a clear cluster for documents concerning computer topics (see red and dark-orange cluster, located central south-west). Moreover, smaller clusters can be seen for the miscellaneous for-sale topic (orange cluster, located central north-west), science topics (green cluster, located central north), and sports topics (yellow cluster, located central north-east). On the other hand, little clustering can be seen for the random-based embedding map. The documents are much more spread out in general, as is made evident by the less dense central cluster of all the documents.

The main similarity between the two maps is the existence of a central cluster of all the documents, and the existence of documents which heavily deviate from this central cluster.

Multiclass Classification

Question 12

The margin for the SVM classifiers was chosen to be the best parameter from the list $\gamma = \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ using 5-fold cross-validation grid search that optimizes for the highest validation accuracy.

a. (Gaussian) Naive Bayes

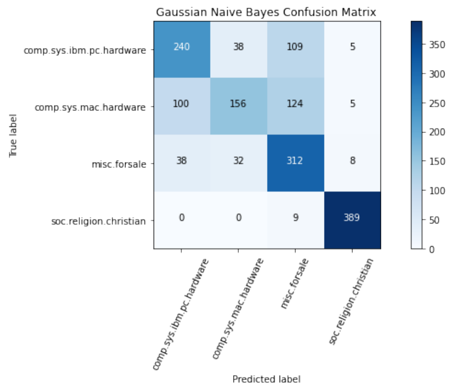


Figure 23: Confusion matrix for (Gaussian) Naive Bayes multiclass classification.

Overall testing accuracy: 70.1%

Category	Precision	Recall	F-1 Score	Support
comp.sys.ibm.pc.hardware	0.63	0.61	0.62	392
comp.sys.mac.hardware	0.69	0.41	0.51	385
misc.forsale	0.56	0.80	0.66	390
soc.religion.christian	0.96	0.98	0.97	398

Table 9: Metrics for (Gaussian) Naive Bayes multiclass classification.

b. One vs. One

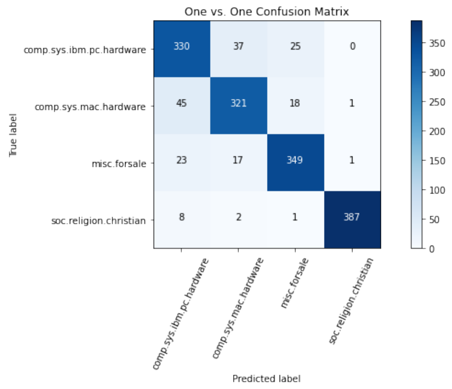


Figure 24: Confusion matrix for one vs. one multiclass classification.

Overall testing accuracy: 88.6%

Category	Precision	Recall	F-1 Score	Support
comp.sys.ibm.pc.hardware	0.81	0.84	0.83	392
comp.sys.mac.hardware	0.85	0.83	0.84	390
misc.forsale	0.89	0.89	0.89	390
soc.religion.christian	0.99	0.97	0.98	398

Table 10: Metrics for One vs. One multiclass classification.

c. One vs. All

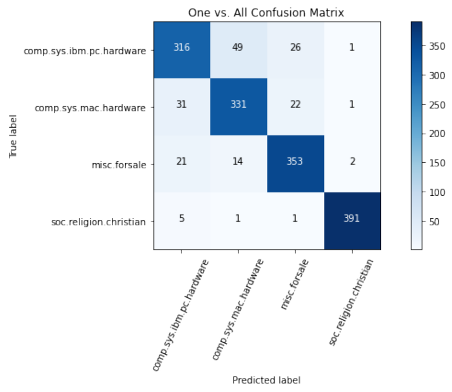


Figure 25: Confusion matrix for One vs. All multiclass classification.

Overall testing accuracy: 88.9%

Category	Precision	Recall	F-1 Score	Support
comp.sys.ibm.pc.hardware	0.85	0.81	0.83	392
comp.sys.mac.hardware	0.84	0.86	0.85	385
misc.forsale	0.88	0.91	0.89	390
soc.religion.christian	0.99	0.98	0.99	398

Table 11: Metrics for One vs. All multiclass classification.