UNIVERSITY OF CALIFORNIA, LOS ANGELES
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Project 1

Student name: *Lung-Hui Wu (106152317)*

Course: *EC ENGR 219* – Professor: *Vwani Roychowdhury*
Due date: *January 26th, 2024*

Getting familiar with the dataset

Question 1: Provide answers to the following questions

1. Overview: How many rows (samples) and columns (features) are present in the dataset?
2. Histograms: Plot 3 histograms on : (a) The total number of alpha-numeric characters per data point (row) in the feature full text: i.e count on the x-axis and frequency on the y-axis; (b) The column leaf_label - class on the x-axis; (c) The column root_label - class on the x-axis.
3. Interpret Plots: Provide qualitative interpretations of the histograms.

Answer.

1. There are 3476 rows (samples) and 8 columns (features) in the dataset.
2. Histograms:

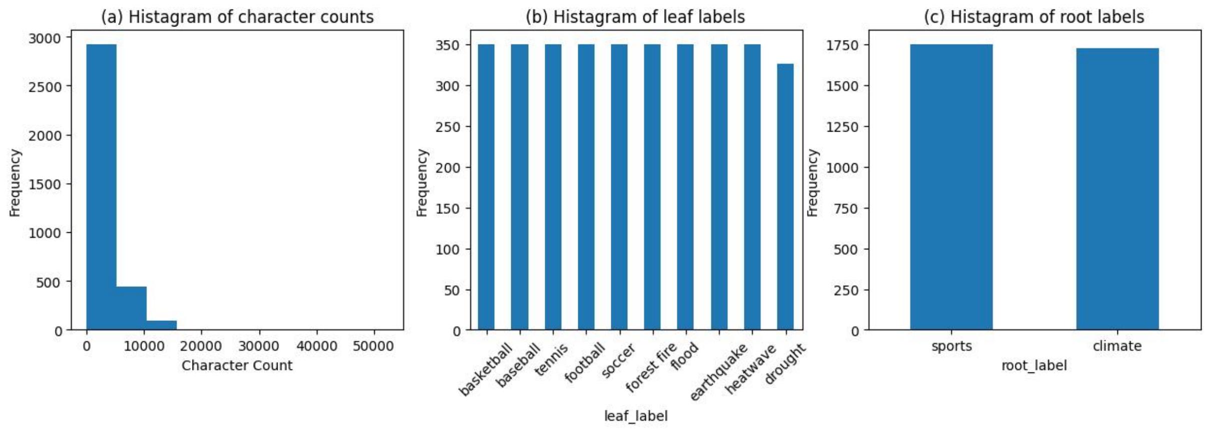


Figure 1: Histograms of the dataset

3. Interpret Plots:
   (a) Most of the posts have less than 10000 characters. The frequency decays exponentially as the character count increases.

(b) Each leaf label are balanced except the label "drought". There are 326 posts about "drought" and 350 posts for every other leaf labels.

(c) The number of posts are close with respect to root labels, 1750 posts are about sports and 1726 are about climate.

root_label	**sports**					**climate**				
leaf_label	basketball	baseball	tennis	football	soccer	forest fire	flood	earthquake	drought	heatwave

Table 1: Hierarchical arrangment of root_label and leaf_label

Binary Classification

Question 2: Report the number of training and testing samples.

Answer. There are 2780 training samples and 696 testing samples.

Question 3: Please answer the following question:

1. What are the pros and cons of lemmatization versus stemming? How do these processes affect the dictionary size?
2. min_df means minimum document frequency. How does varying min_df change the TF-IDF matrix?
3. Should I remove stopwords before or after lemmatizing? Should I remove punctuations before or after lemmatizing? Should I remove numbers before or after lemmatizing? Hint: Recall that the full sentence is input into the Lemmatizer and the lemmatizer is tagging the position of every word based on the sentence structure.
4. Report the shape of the TF-IDF-processed train and test matrices. The number of rows should match the results of Question 2. The number of columns should roughly be in the order of $k \times 10^3$. This dimension will vary depending on your exact method of cleaning and lemmatizing and that is okay.

Answer.

Methods	Lemmatization	Stemming
Pros	Can give meaningful representation considering linguistic context	Easier to implement and faster execution
Cons	Runs slower	Result contains inaccuracy

Table 2: The pros and cons of lemmatization versus stemming

1. Stemming chops off word endings and may not output an actual word in English, while lemmatization groups different forms of words into a same lemma. Therefore, the dictionary size of stemming output is bigger than lemmatization.

2. If min_df is higher, that means a word needs appearance in more documents to be counted, resulting in fewer columns in the TF-IDF matrix. If min_df is lower, the matrix grows bigger with more words in it.

3. We should remove stopwords, numbers, and punctuations after lemmatizing. If we do it first, certain phrases in the sentences may be filtered, causing false results in position tagging and context understanding while performing lemmatization.

4. Train matrix: (2780, 13731), test matrix: (696, 13731)

Question 4: Reduce the dimensionality of the data:

1. Plot the explained variance ratio across multiple different k = [1, 10, 50, 100, 200, 500, 1000, 2000] for LSI and for the next few sections choose k = 50. What does the explained variance ratio plot look like? What does the plot's concavity suggest?

2. With k = 50 found in the previous sections, calculate the reconstruction residual MSE error when using LSI and NMF - they both should use the same k = 50. Which one is larger, the $||\mathbf{X} - \mathbf{WH}||_F^2$ in NMF or the $||\mathbf{X} - \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T||_F^2$ in LSI and why?

Answer.

1. The cumulative ratio grows steeply at early stage and gradually saturates by adding up minor ratios. The curve can also be seen as the cumulative explained variance ratio using the first k components. The principle components are sorted so the first few components take up larger explained variance ratio while the rest take up a small proportion.

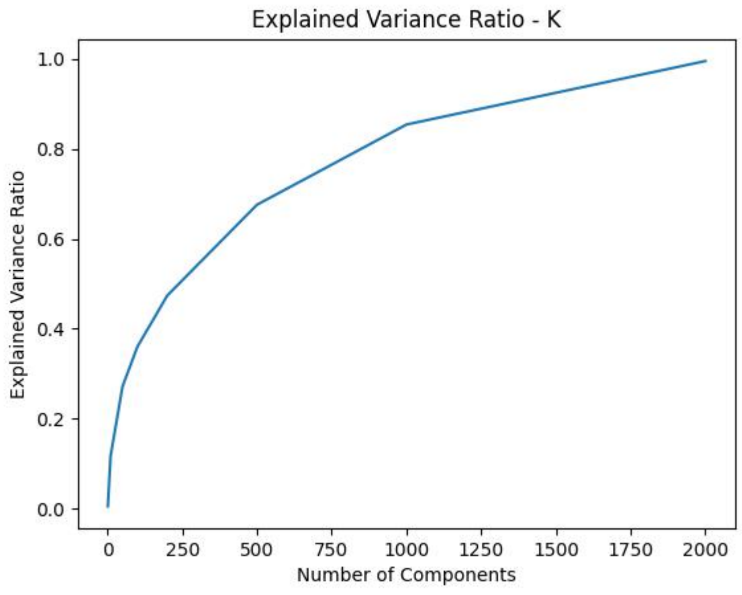


Figure 2: The explained Variance Ratio across different k

2. Loss in LSI: 1953.41, Loss in NMF: 1986.96. NMF has higher loss than LSI does. There are three reasons. First, by capturing the underlying structure in the data, LSI can ignore irrelevant or less important information, which might be noise. Also, LSI is good at capturing synonyms and the multiple meanings of words (polysemy). This is because it reduces dimensionality while preserving the relationship between terms and documents, which can lead to more accurate results. Lastly, LSI can handle both positive and negative data, which is not the case with NMF. This characteristic makes LSI more adaptable to a broader range of data types and applications.

Question 5: Compare and contrast hard-margin and soft-margin linear SVMs:

1. Train two linear SVMs:
   - Train one SVM with $\gamma = 1000$ (hard margin), another with $\gamma = 0.0001$ (soft margin).
   - Plot the ROC curve, report the confusion matrix and calculate the accuracy, recall, precision and F-1 score of both SVM classifiers on the testing set. Which one performs better? What about for $\gamma = 100000$?
   - What happens for the soft margin SVM? Why is the case? Analyze in terms of the confusion matrix.
   - Does the ROC curve reflect the performance of the soft-margin SVM? Why?
2. Use cross-validation to choose γ (use average validation accuracy to compare): Using a 5-fold cross-validation, find the best value of the parameter γ in the range $\{10^k | -3 \leq k \leq 6, k \in \mathbb{Z}\}$. Again, plot the ROC curve and report the confusion matrix and calculate the accuracy, recall precision and F-1 score of this best SVM.

Answer.

1. Train two linear SVMs:

   The hard margin SVM performs better. Soft SVM allows some misclassification in order to have a seperation with wider margin. From the confusion matrix, we can see that the soft SVM labelled every data to the "climate" class. Laying much emphasis on a wide margin, the soft SVM cannot discriminate the two classes.

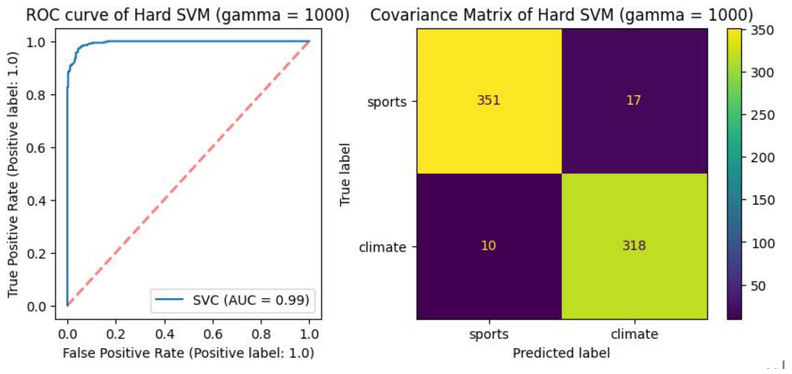


Figure 3: The ROC curve and Confusion Matrix of Hard SVM with $\gamma = 1000$

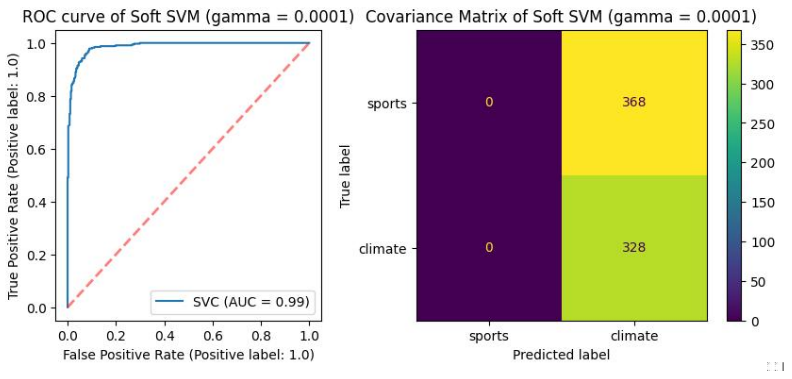


Figure 4: The ROC curve and Confusion Matrix of Soft SVM with γ = 0.0001

The ROC curve of soft SVM doesn't reflect its preformance. The coordinate is at (0,0) on the plot when the threshold is infinity, as every data is predicted as negative. Now the threshold drops a little bit, the TPR immediately goes close to 1, because the soft-margin SVM tends to predict every data as positive. As a result, the ROC curve of soft SVM seems very well while it actually performs poorly.

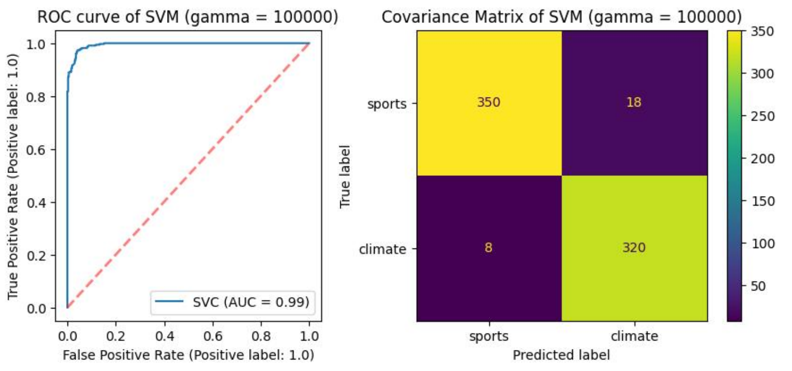


Figure 5: The ROC curve and Confusion Matrix of SVM with γ = 100000

2. Best SVM: $\gamma = 10000$

γ	Accuracy	Recall	Precision	F-1 Score
1000 (Hard)	0.961	0.970	0.949	0.959
0.0001 (Soft)	0.471	1.000	0.471	0.641
100000	0.963	0.976	0.947	0.961
10000 (Best)	0.963	0.976	0.947	0.961

Table 3: The performance of SVMs with different γ

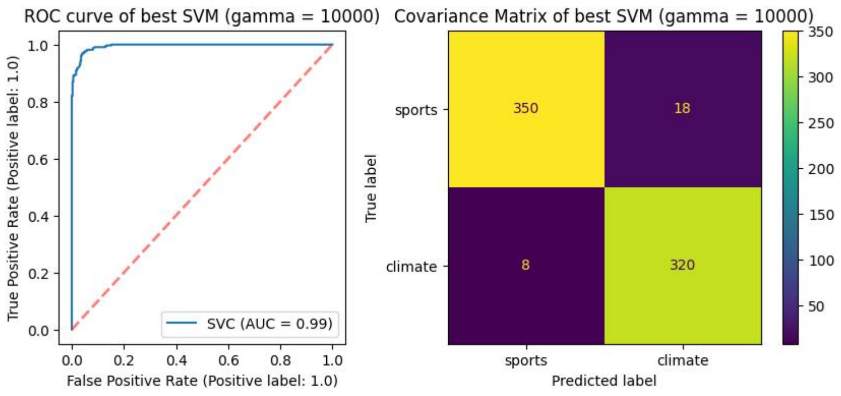


Figure 6: The ROC curve and Confusion Matrix of the best SVM with γ = 10000

Question 6: Evaluate a logistic classifier

1. 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 on the testing set.

2. Find the optimal regularization coefficient:

   - Using 5-fold cross-validation on the dimension-reduced-by-SVD training data, find the op- timal regularization strength in the range $\{10^k | -5 \leq k \leq 5, k \in \mathbb{Z}\}$ for logistic regression with L1 regularization and logistic regression with L2 regularization, respectively.
   - Compare the performance (accuracy, precision, recall and F-1 score) of 3 logistic classi- fiers: w/o regularization, w/ L1 regularization and w/ L2 regularization (with the best parameters you found from the part above), using test data.
   - 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?
   - Both logistic regression and linear SVM are trying to classify data points using a linear decision boundary. What is the difference between their ways to find this boundary? Why do their performances differ? Is this difference statistically significant?

Answer.

1. Logistic classifier without regularization

Performance	Accuracy	Recall	Precision	F-1 Score
w/o regularization	0.957	0.954	0.954	0.954
w/ best L1 regularization	0.958	0.963	0.949	0.956
w/ best L2 regularization	0.960	0.960	0.955	0.957

Table 4: The performance of logistic classifiers with different regularizatioin settings

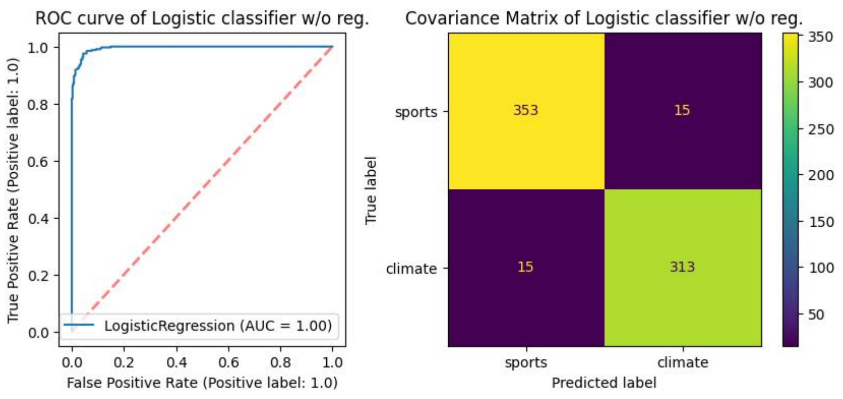


Figure 7: The ROC curve and Confusion Matrix of the Logistic classifier w/o reg.

2. Logistic classifier with regularization

   Best L1 regularization: k = 10.
   Best L2 regularization: k = 100.

   - Parameter's effect on test error

     A small regularization parameter means less penalty on the coefficients. This can lead to a model that fits the training data very well but may overfit, resulting in higher test error due to poor generalization.
     A large regularization parameter imposes a strong penalty, leading to simpler models with smaller coefficients. While this can reduce overfitting and potentially lower test error, too large a parameter might underfit the data, failing to capture important patterns, again increasing test error.
     The optimal regularization parameter balances the trade-off between fitting the training data well and maintaining the model's ability to generalize to unseen data.

   - Parameter's effect on learned coefficient

     L1 regularization tends to produce sparse models, where some coefficients can become exactly zero when the parameter is large.
     L2 regularization generally does not set coefficients to zero but shrinks them towards zero.

   - Interest in Each Type of Regularization

     L1 regularizatioin is useful when we have many features and suspect that only a subset is relevant. It helps in identifying the most significant features. Also, models with fewer features are generally easier to interpret.

     L2 Regularization is effective in situations where features are correlated, as it stabilizes the coefficients. Besides, it's useful when we want to include all features but limit their individual influence on the prediction.

Question 7: Evaluate and profile a Naive 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 on the testing set.

Answer. GaussianNB Classifier:

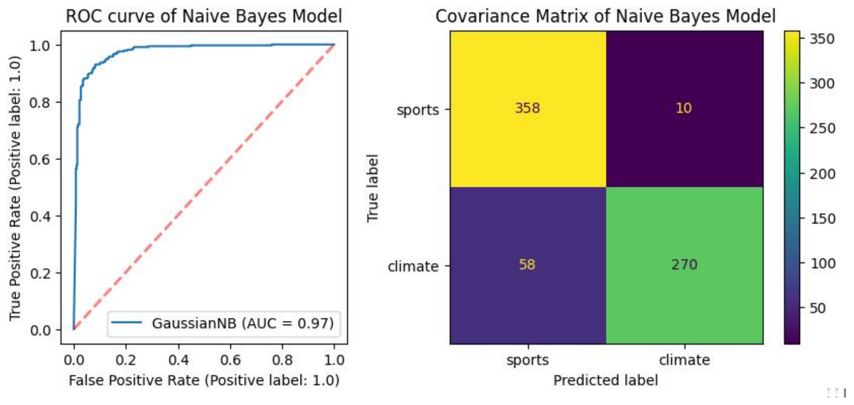


Figure 8: The ROC curve and Confusion Matrix of Naive Bayes Model

Performance	Accuracy	Recall	Precision	F-1 Score
Naive Bayes Model	0.902	0.823	0.964	0.888

Table 5: The performance of Naive Bayes Model

Question 8: Construct a Pipeline that performs feature extraction, dimensionality reduction and classification. The evaluation of each combination is performed with 5-fold cross-validation (use the average validation set accuracy across folds).
What are the 5 best combinations? Report their performances on the testing set?

Answer. The result of grid search

Module	Options
Loading Data	Clean the data
Feature Extraction	min_df = 3 or 5 while constructing the vocalubary, Lemmatization or Stemming
Dimensionality Reduction	LSI (k = [5, 30, 80]) or NMF (k = [5, 30, 80])
Classifier	SVM or Logistic Regression (L1 or L2 regularization) or GaussianNB

Table 6: Set of hyperparameters to consider for pipeline comparison

Rank	min_df	Compression	Dimension Reduction	Classifier
1	3	Lemmatization	LSI(k=80)	Logistic Regression (L2, k=100)
2	5	Lemmatization	LSI(k=80)	Logistic Regression (L2, k=100)
3	5	Stemming	LSI(k=80)	Logistic Regression (L2, k=100)
4	3	Stemming	LSI(k=80)	Logistic Regression (L2, k=100)
5	5	Lemmatization	LSI(k=80)	Logistic Regression (L1, k=10)

Table 7: Hyperparameters of top 5 best combinations

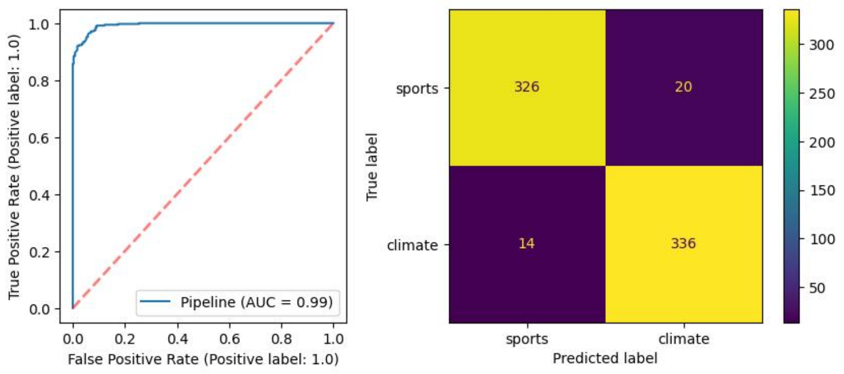


Figure 9: The ROC curve and Confusion Matrix of the No. 1 combination.

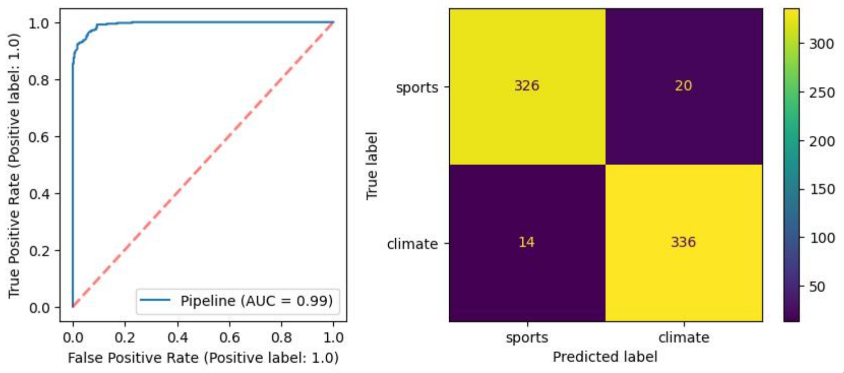


Figure 10: The ROC curve and Confusion Matrix of the No. 2 combination.

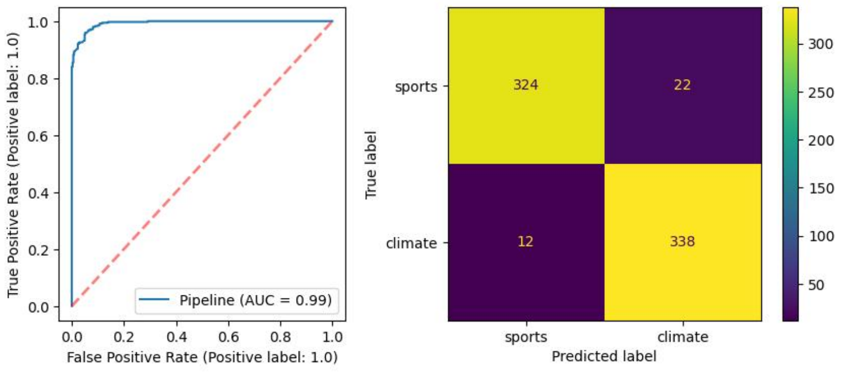


Figure 11: The ROC curve and Confusion Matrix of the No. 3 combination.

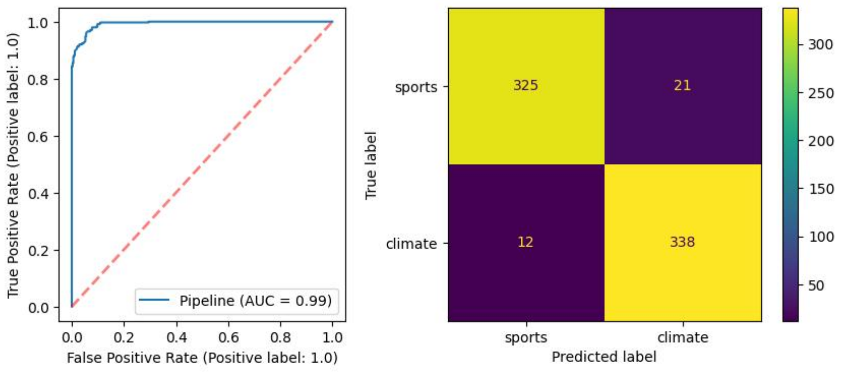


Figure 12: The ROC curve and Confusion Matrix of the No. 4 combination.

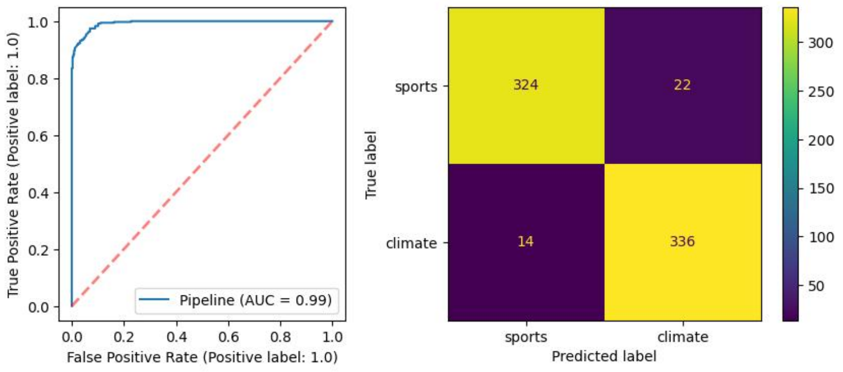


Figure 13: The ROC curve and Confusion Matrix of the No. 5 combination.

Multiclass Classification

Question 9: In this part, we aim to learn classifiers on the documents belonging to unique classes in the column leaf_label

1. Perform Naive 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. How did you resolve the class imbalance issue in the One VS the rest model?
2. Do you observe any structure in the confusion matrix? Are there distinct visible blocks on the major diagonal? What does this mean?
3. Based on your observation from the previous part, suggest a subset of labels that should be merged into a new larger label and recompute the accuracy and plot the confusion matrix. How did the accuracy change in One VS One and One VS the rest?
4. Does class imbalance impact the performance of the classification once some classes are merged? Provide a resolution for the class imbalance and recompute the accuracy and plot the confusion matrix in One VS One and One VS the rest?

Answer.

1. Performance of each classification method

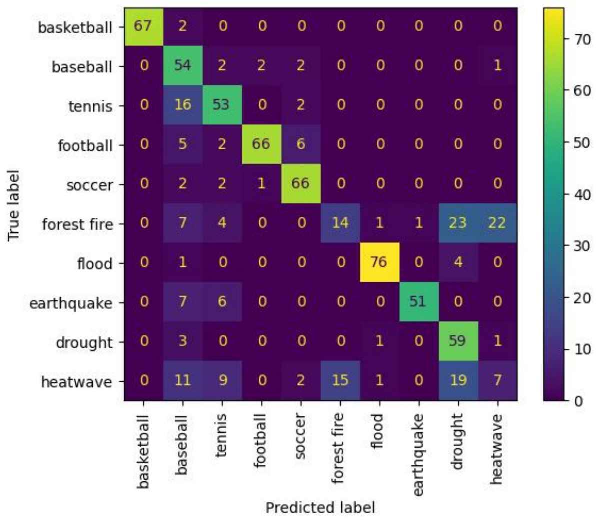


Figure 14: The Confusion Matrix of Naive Bayes Classification on original data

The imbalance in One vs the rest method is solved by the "balanced" mode of class weight setting in SVM. The mode set each label's weight inversely proportional to class frequencies in the input data.

Performance	Accuracy	Recall	Precision	F-1 Score
Naive Bayes Classification	0.737	0.737	0.732	0.716
One vs One SVM Classification	0.780	0.780	0.793	0.782
One vs the rest SVM Classification	0.800	0.800	0.806	0.800

Table 8: The performance of each classification method on original data

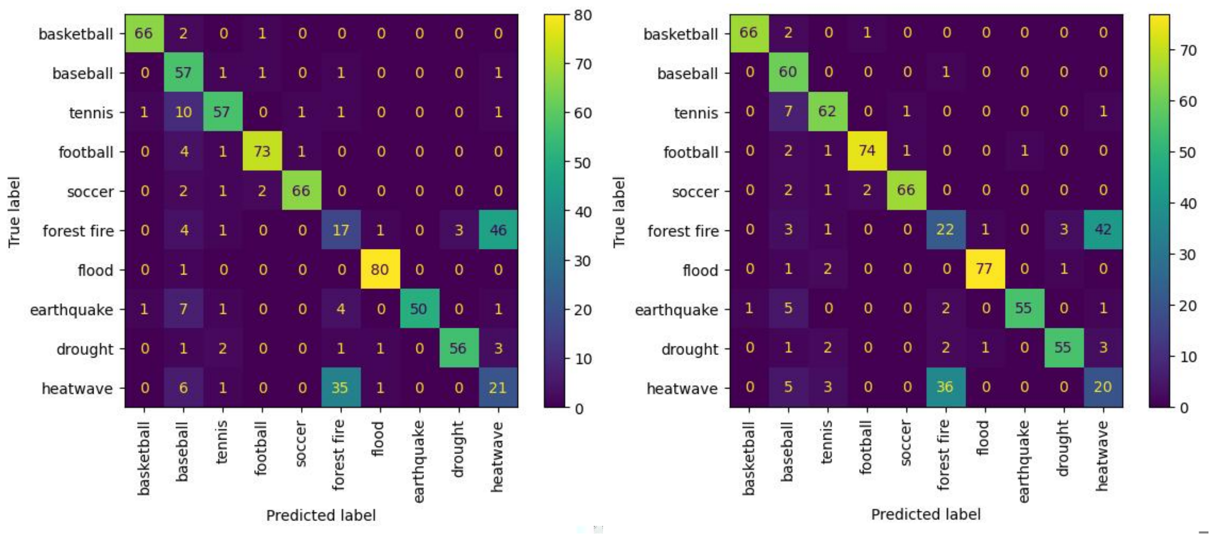

(a) One vs one SVM (b) One vs the rest SVM

Figure 15: The Confusion Matrix of two SVM methods on original data

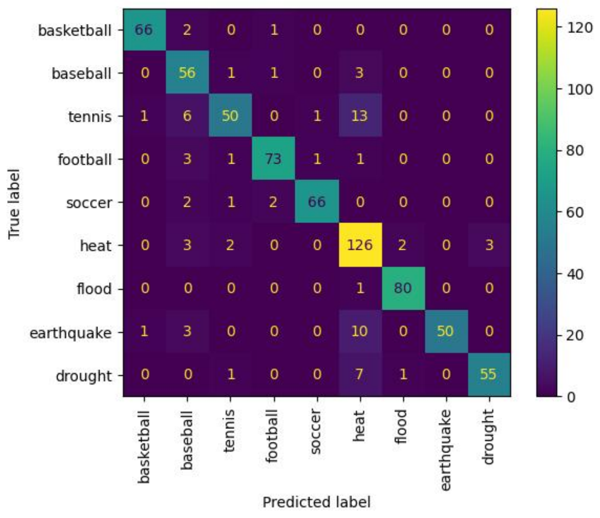
2. From the confusion matrices of the three methods, we can see that the predictions are correct for most labels, the diagonal blocks are significantly brighter. However, the diagolnal blocks of "heatwave" and "forest fire" are darker, while the false prediction blocks (at location (9,5) and (5,9)) are brighter. That means while the classifiers can identify most labels, they cannot tell the difference between "heatwave" and "forest fire".

3. Based on the observation above, I merged "heatwave" and "forest fire" as a new label "heat" and run the classification models.

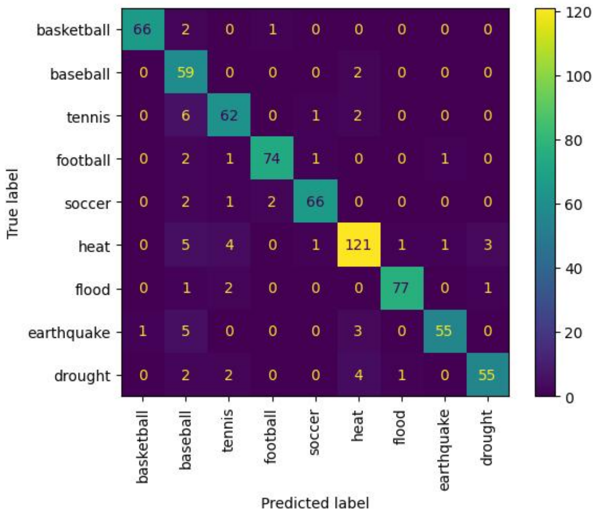
4. As performed in the One vs the rest classifier, I used the "balanced" mode of SVM to address the imbalance of merged data.
   For one vs one SVM, the performance improved after handling imbalanced data.
   For one vs the rest SVM, since the data imbalance already exists and is handled by SVM in the one vs the rest setting, there is no difference if we take an already imbalanced dataset.

Performance	Data	Accuracy	Recall	Precision	F-1 Score
One vs One SVM	Imbalanced	0.894	0.894	0.904	0.894
One vs the rest SVM	Balanced	0.912	0.912	0.921	0.914
One vs One SVM	Balanced	0.898	0.898	0.914	0.902
One vs the rest SVM	Balanced	0.912	0.912	0.921	0.914

Table 9: The performance of SVM on merged data

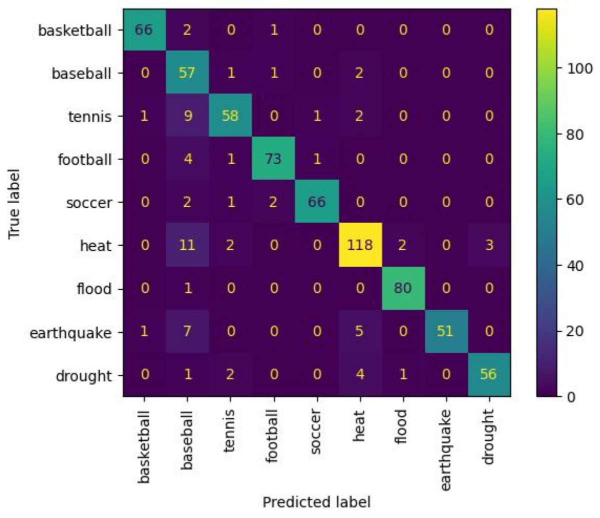


(a) One vs one SVM

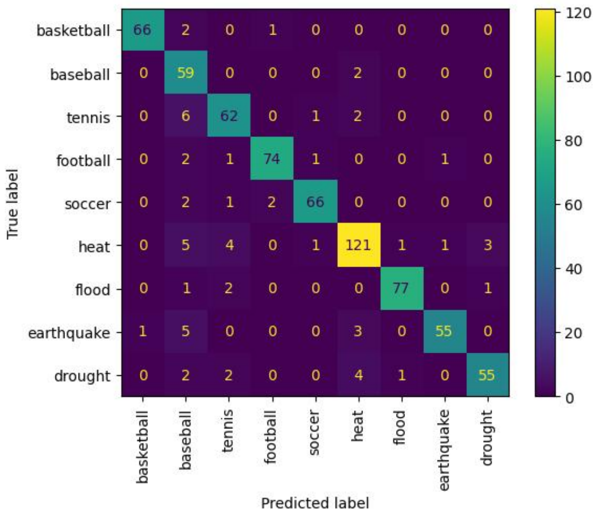


(b) One vs the rest SVM

Figure 16: The Confusion Matrix of two SVM methods on merged data



(a) One vs one SVM



(b) One vs the rest SVM

Figure 17: The Confusion Matrix of two SVM methods on merged data, imbalance handled

Question 10: Read the paper about GLoVE embeddings - found here and answer the following subquestions:

1. Why are GLoVE embeddings trained on the ratio of co-occurrence probabilities rather than the probabilities themselves?
2. In the two sentences: "James is running in the park." and "James is running for the presidency.", would GLoVE embeddings return the same vector for the word running in both cases? Why or why not?
3. What do you expect for the values of, $||GLoVE[''woman''] - GLoVE[''man'']||_2$, $||GLoVE[''wife''] - GLoVE[''husband'']||_2$ and $||GLoVE[''wife''] - GLoVE[''orange'']||_2$? Compare these values.
4. Given a word, would you rather stem or lemmatize the word before mapping it to its GLoVE embedding?

Answer.

1. - GLoVE embeddings use the ratio of co-occurrence probabilities instead of just the probabilities to better capture the relationships between words. This approach focuses on how words relate to each other in specific contexts. Using ratios helps the model distinguish between different types of relationships more clearly.
   - It reduces the influence of very common words, like "the," which might appear often with many words but don't necessarily indicate a meaningful relationship.
   - Using ratios helps in dealing with the wide range of values that raw probabilities can have, making the training process more stable and efficient.
   - This method highlights which word co-occurrences are statistically significant, helping the model to learn more meaningful and interpretable patterns in the data.
2. No, GLoVE embeddings would provide different vectors for "running" in the two sentences. This difference arises because GLoVE captures word meanings based on context. In the first sentence, "running" is linked to physical activity, associating with words like "jogging" or "park." In the second, it's used politically, relating to terms like "campaign" or "election." Though there's a common base meaning, GLoVE adjusts the word's representation according to these specific contexts, reflecting the varied usage and co-occurrence patterns of "running" in different sentences.
3. The distance between 'wife' and 'husband' is the shortest, while the distance between 'wife' and 'orange' is the furthest.
   $||GLoVE[''woman''] - GLoVE[''man'']||_2 = 0.0753$
   $||GLoVE[''wife''] - GLoVE[''husband'']||_2 = 0.0331$
   $||GLoVE[''wife''] - GLoVE[''orange'']||_2 = 0.2504$

4. For GLoVE embeddings, which are designed to capture nuanced meanings based on word co-occurrences, lemmatization is generally more suitable. It preserves more of the word's meaning and context, leading to a more accurate representation in the embedding space.

Question 11: For the binary classification task distinguishing the "sports" class and "climate" class:

1. Describe a feature engineering process that uses GLoVE word embeddings to represent each document.
2. elect a classifier model, train and evaluate it with your GLoVE-based feature. If you are doing any cross-validation, please make sure to use a limited set of options so that your code finishes running in a reasonable amount of time.

Answer.

1. (a) Use the "keywords" column as input.
   (b) Lemmatize all the keywords.
   (c) Generate GLoVE embedding for keywords that can be found in the dictionary, and disgard the rest.
   (d) Normalize each embedding and return the average of the vectors.
2. I select SVM (γ=10) as the classifier.

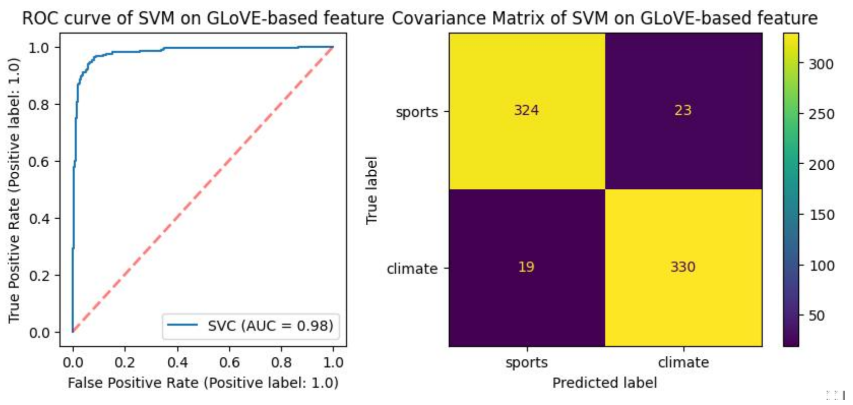


Figure 18: The ROC curve and Confusion Matrix of SVM on GLoVE-based feature

Performance	Accuracy	Recall	Precision	F-1 Score
Naive Bayes Model	0.940	0.946	0.935	0.940

Table 10: The performance of SVM on GLoVE-based feature

Question 12: Plot the relationship between the dimension of the pre-trained GLoVE embedding and the resulting accuracy of the model in the classification task. Describe the observed trend. Is this trend expected? Why or why not?

Answer. The accuracy grows as I used larger dimension of GLoVE embedding. The trend is as expected since higher-dimensional GLoVE embeddings can capture more nuanced and detailed semantic information. In higher dimensions, each word vector has more features, allowing for finer distinctions between words based on their usage in different contexts.

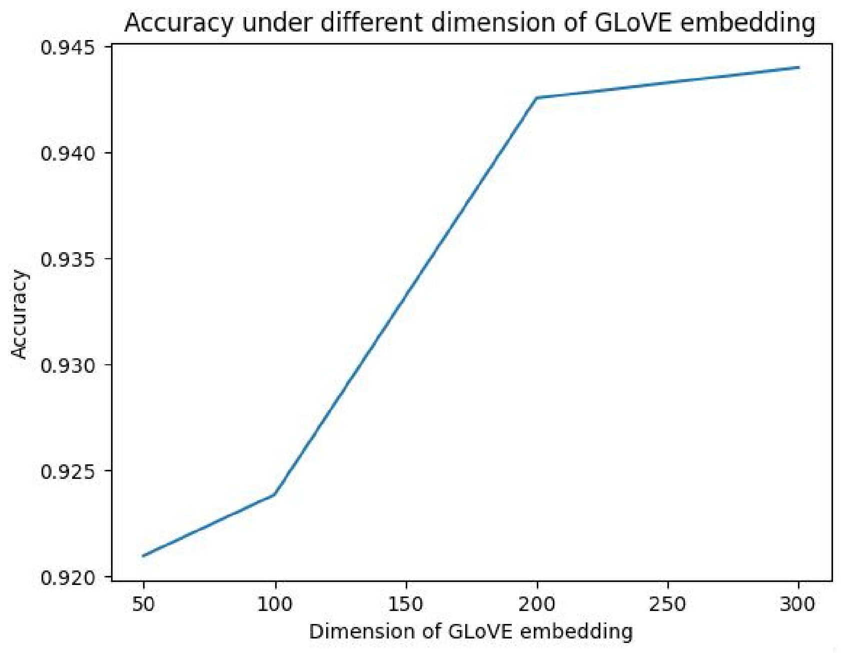


Figure 19: Accuracy under different dimension of GLoVE embedding

Question 13: Visualize the set of normalized GLoVE-based embeddings of the documents with their binary labels in a 2D plane using the UMAP library. Similarly generate a set of normalized random vectors of the same dimension as GLoVE and visualize these in a 2D plane with UMAP.

Compare and contrast the two visualizations. Are there clusters formed in either or both of the plots?

Answer. Clusters only form in the UMAP of GLoVE-embedding.
In the UMAP of random vectors, all the datapoints mix together in a big rough group.

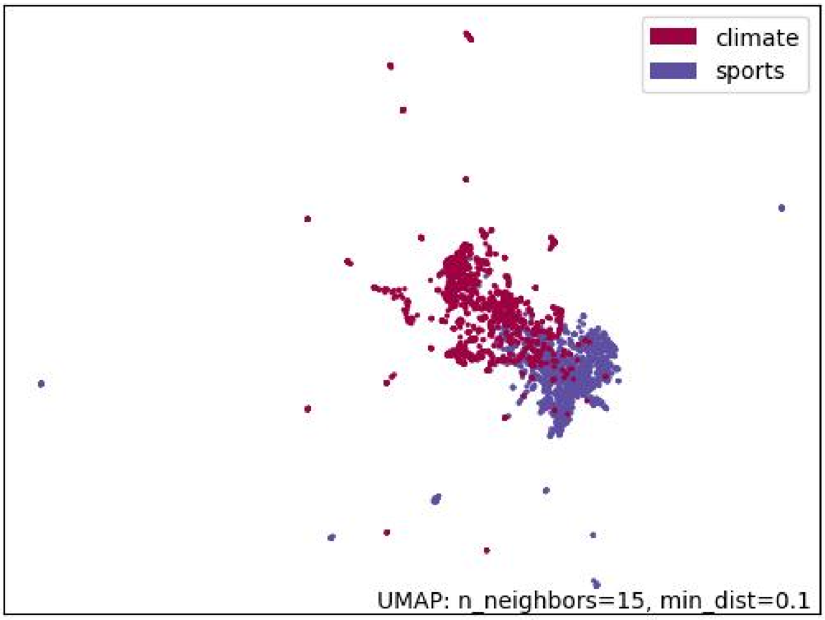


Figure 20: UMAP of GLoVE embedding

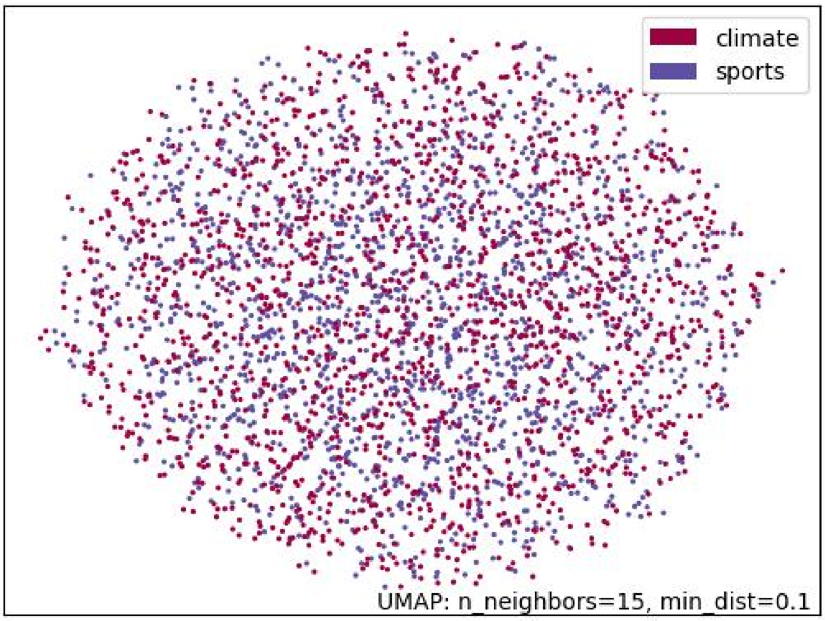


Figure 21: UMAP of random vectors