ECE219 Large-Scale Data Mining: Models and Algorithms **(Due: Sun, 01/28/24)**

Project 1: Classification Analysis on Textual Data

Instructor: Vwani Roychowdhury Aryaman Rajesh Gokarn 506303588

The Statistical classification broadly refers to the task of learning to identify a subset of categories that pertain to a data point (sample of text, an image, a video clip, a time-signal etc..) from a predefined (generally human-guided) larger set of categories. The model attempts to master the task given a training data set (that is kept separate from an evaluation set) in which each data point is pre-labeled with their "correct" category membership/s.

1 Getting Familiar with the Dataset

QUESTION 1.

(i) How many rows (samples) and columns (features) are present in the dataset?

Answer:- There are **3476 rows** and **8 columns** in the dataset

(ii) 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.

Answer:-

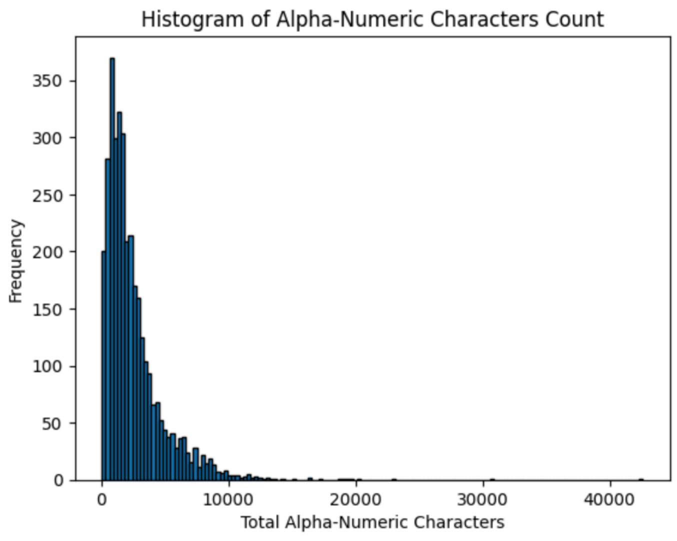


Figure 1: Histogram of Training Data

The histogram plot 1 shows that the majority of the entries have a alphanumeric character count between 0 and 5000.

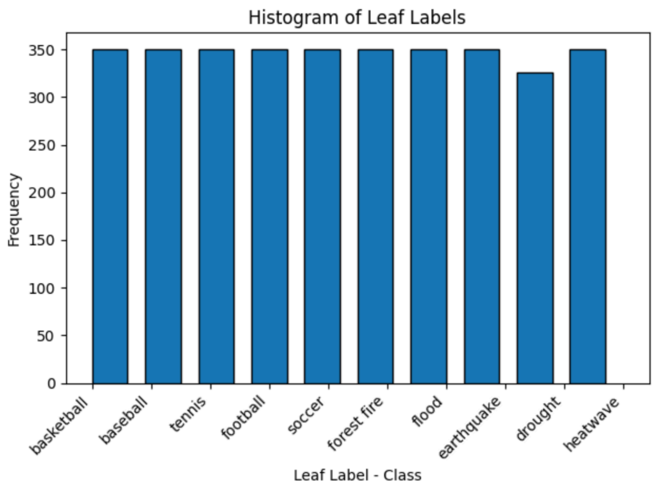


Figure 2: Histogram of Leaf Label Class

The histogram plot 2 shows the count/frequency of each leaf label (each sport and climate type). We can see that there are 350 datapoints for each sport and climate category except climate - "drought" (326)

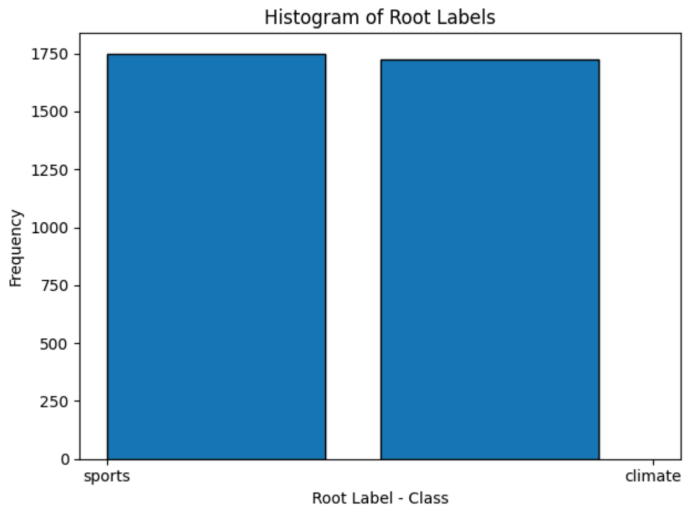


Figure 3: Histogram of Root Label Class

The histogram plot 3 shows the count/frequency of each root label (sport and climate). We can see that there are roughly 1750 datapoints for sport and roughly 1726 climate category

2 Binary Classification

For the first part of the project, we will be using only the full text column as the raw features per sample (row) and the root label column as the label for each sample. The root labels are well-separated.

2.1 Splitting the entire dataset into training and testing data

In order to measure the performance of our binary classification model, we split the dataset into a training and a testing set. The model is trained on the training set and evaluated on the testing set.

Note: Do not train on the testing set. We create the sets with a Pandas dataframe input that contains the entire dataset df

.

QUESTION 2. Report the number of training and testing samples.

Answer :- There are **2780 training samples** and **696 test samples.**

2.2 Feature Extraction

QUESTION 3.

(i) What are the pros and cons of lemmatization versus stemming? How do these processes affect the dictionary size?

Answer:-

Lemmatization and stemming are both techniques used in natural language processing to reduce words to their base or root form. Here are the pros and cons of lemmatization versus stemming, along with their effects on the dictionary size:

Lemmatization:

Pros:

- Accuracy: Lemmatization provides more accurate results compared to stemming as it considers the context and meaning of words.
- Readability: Lemmatized words are often more readable and linguistically correct since they are reduced to their base dictionary form.
- Context Preservation: Lemmatization preserves the semantic meaning of words, making it suitable for tasks that require understanding the context.

Cons:

- Computational Cost: Lemmatization is computationally more expensive and slower than stemming because it involves dictionary lookups and part-of-speech tagging.
- Resource Intensive: Requires a comprehensive dictionary or lexicon to perform accurate lemmatization.

Stemming:

Pros:

- Speed: Stemming is faster and computationally less intensive compared to lemmatization, making it more suitable for large datasets or real-time applications. Simplicity: Stemming is a simpler and rule-based process, making it easier to implement and understand.

Cons:

- Overstemming: Stemming may lead to overstemming, where words are excessively reduced, potentially losing some semantic meaning. Accuracy: Stemming is less accurate than lemmatization as it doesn't consider the context or meaning of words.

Effects on Dictionary Size:

- Lemmatization: The dictionary size may be larger as lemmatization aims to reduce words to their base or dictionary form. Each unique lemma is considered a separate entry in the dictionary.
- Stemming: The dictionary size may be smaller as stemming reduces words to their root form, and many words share the same stem. However, stemming may also introduce some variations that are not actual words (e.g., "happi" instead of "happy").

QUESTION 3.

(ii) min df means minimum document frequency. How does varying min df change the TF-IDF matrix?

Answer:-

The min_df parameter in TF-IDF (Term Frequency-Inverse Document Frequency) represents the minimum document frequency for a term to be included in the calculation. It is used to control the inclusion of terms that occur in only a small number of documents. Varying the min_df parameter can have a significant impact on the resulting TF-IDF matrix.

Increasing min_df leads to:

- Reduction in Vocabulary Size: Terms that occur in fewer documents than the specified min_df will be excluded from the vocabulary. This leads to a reduction in the overall vocabulary size.
- Focus on More Frequent Terms: By increasing min_df, you focus on terms that are more common across documents, potentially capturing more general and important features.
- Noise Reduction: Terms that occur infrequently may represent noise or specific instances that might not contribute significantly to the overall meaning. Increasing min_df can help reduce such noise.

QUESTION 3.

(iii) Should I remove stop words before or after lemmatizing? Should I remove punctuations before or after lemmatizing? Should I remove numbers before or after lemmatizing?

Answer:-

- Stopword Removal: Before Lemmatization

  It is generally recommended to remove stopwords before lemmatization. Stopwords (common words like "the," "and," "is") are often irrelevant for semantic analysis, and removing them early helps reduce noise. Lemmatization is more effective when applied to content words rather than function words like stopwords.
- Punctuation Removal: Before Lemmatization

  Removing punctuation before lemmatization is a common practice. Punctuation marks usually do not carry semantic meaning and can be safely removed to focus on the actual words.
- Number Removal: Before Lemmatization

  Similar to stopwords and punctuation, it is generally advisable to remove numbers before lemmatization. Numbers might not contribute much to the semantic content of the text, and lemmatization can be more effective when applied to alphabetic words

QUESTION 3.

(iv) 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×103. This dimension will vary depending on your exact method of cleaning and lemmatizing and that is okay.

Answer:-

- TF-IDF-processed train data shape : (2780, 13729)
- TF-IDF-processed train data shape : (696, 13729)

2.3 Dimensionality Reduction

The dimensionality of TF-IDF vectors above ranges in the order of thousands. Since the document-term TF-IDF matrix is sparse and low-rank, we can transform the features into a lower dimensional space in order to circumvent the curse of dimensionality.

In this project, we use two dimensionality reduction methods: **Latent Semantic Indexing (LSI)** and **Non-negative Matrix Factorization (NMF)**, both of which minimize mean squared residual between the original data and a reconstruction from its low-dimensional approximation.

QUESTION 4.

(i) 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?

.

Answer:-

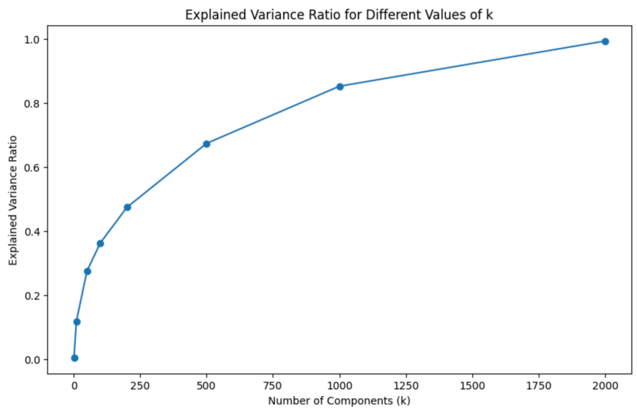


Figure 4. Explained Variance for different k values

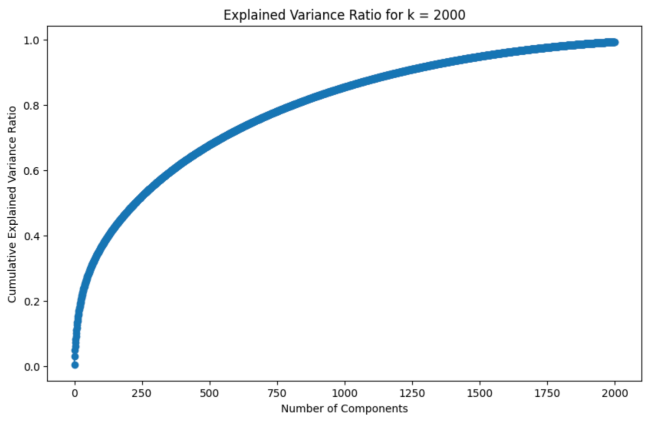


Figure 5. Explained Variance for k values till 2000

The explained variance ratio changes with increasing k. If the plot shows diminishing returns in terms of explained variance ratio as k increases, it suggests that adding more components beyond a certain point contributes less to explaining the variance in the data. The concavity may indicate a point of diminishing returns in terms of capturing additional information with more components.

QUESTION 4.

(ii) 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 ‖X−WH‖2 in NMF or the X−U Σ VT 2 in LSI and why?

Answer:-

We compute the squared frobenius norm of difference matrices between the original data and a reconstruction from their respective low-dimensional approximation. The results are reported as below,

Squared Frobenius Norm for LSI	1941.5949
Squared Frobenius Norm for NMF	1968.2714

The error for NMF is larger. NMF has a larger reconstruction residual error. This depends on how well the factorization captures the original data structure. A higher MSE indicates a larger difference between the original and reconstructed matrices. It depends on the specific characteristics of the data and how well each method captures its underlying structure. NMF has more constraints than LSI. The added constraints are W≥0 and H≥0. This reduces the degrees of freedom and makes the search space more restrictive. Whereas, LSI has a much larger search space to explore. 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.

2.4 Classification Algorithms

In this part, we use the dimension-reduced training data from LSI to train various types of classifiers, and evaluate the trained classifiers on test data with different classification measures.

2.4.1 SVM

Linear Support Vector Machines have been proved efficient when dealing with sparse high dimensional datasets, including textual data. They have been shown to have good generalization accuracy, while having low computational complexity.

The learning process of the parameter $\mathbf{w}$ and b involves solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \ & \frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma \sum_{i=1}^{n} \xi_i \\ s.t. \ & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad \forall i \in \{1, \cdots, n\} \end{aligned}$$

where $\mathbf{x}_i$ is the ith data point, and $y_i \in \{0, 1\}$ is the class label of it. Note that the tradeoff parameter γ controls relative importance of minimizing the loss function on the training data versus maximizing the margin between the two classes.

QUESTION 5.

(i) 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 γ = 100000?

Answer:-

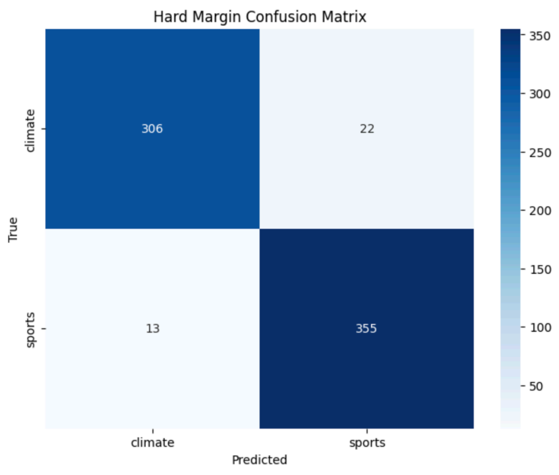
For Hard SVM:

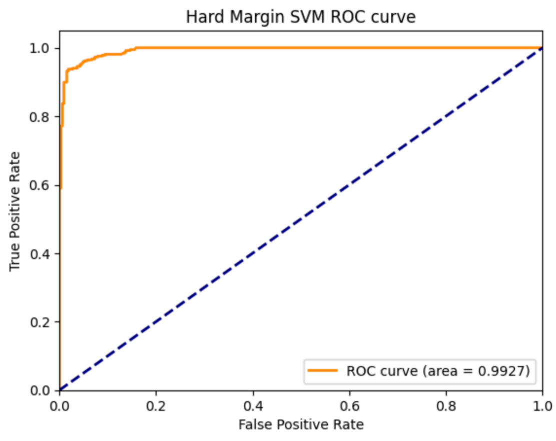
```
Accuracy: 0.9497   Precision: 0.9499
 Recall: 0.9497     F1 Score: 0.9497
```

```
Classification Report:
              precision    recall  f1-score   support

     climate       0.96      0.93      0.95       328
      sports       0.94      0.96      0.95       368

    accuracy                           0.95       696
   macro avg       0.95      0.95      0.95       696
weighted avg       0.95      0.95      0.95       696
```





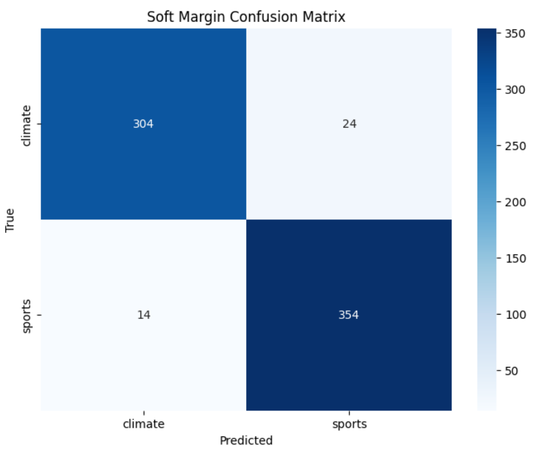
For Soft SVM:

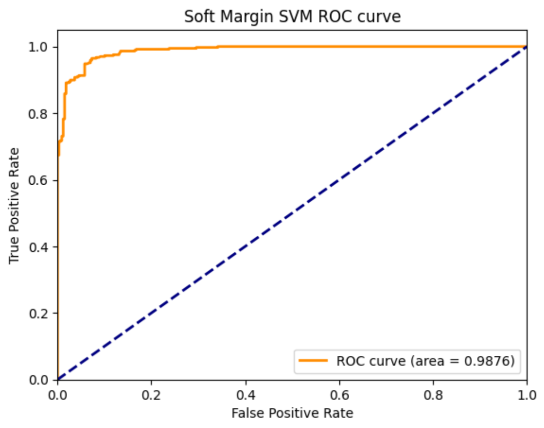
```
Accuracy: 0.9454   Precision: 0.9457
 Recall: 0.9454    F1 Score: 0.9453
```

```
Classification Report:
              precision    recall  f1-score   support

     climate       0.96      0.93      0.94       328
      sports       0.94      0.96      0.95       368

    accuracy                           0.95       696
   macro avg       0.95      0.94      0.95       696
weighted avg       0.95      0.95      0.95       696
```





We employ SVC from sklearn.svm and set the parameter kernel='linear' in order to build up the linear SVM classifiers.

By setting the parameter C=1000 or C=0.0001, we end up with hard and soft margin linear SVMs.

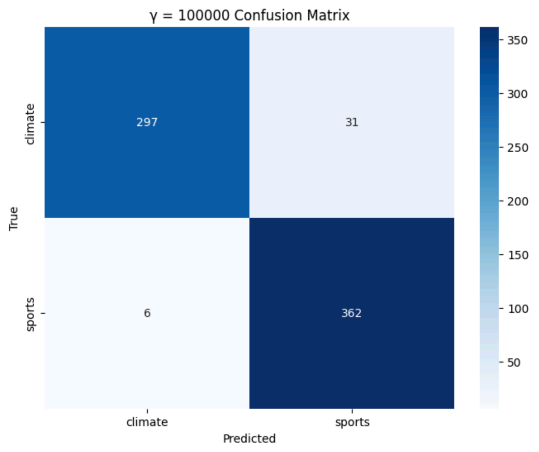
For γ = 100000:

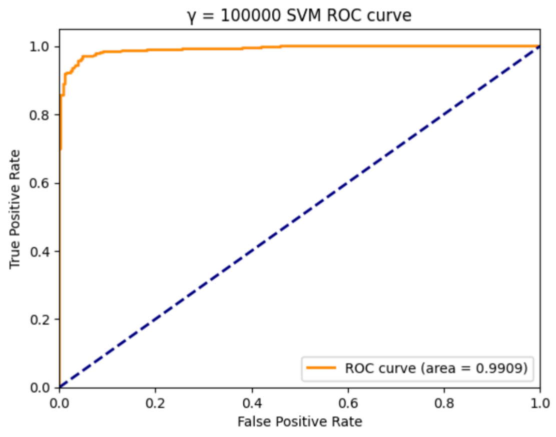
```
Accuracy: 0.9468 Precision: 0.9490
 Recall: 0.9468 F1 Score: 0.9467
```

```
Classification Report:
              precision    recall  f1-score   support

     climate       0.98      0.91      0.94       328
      sports       0.92      0.98      0.95       368

    accuracy                           0.95       696
   macro avg       0.95      0.94      0.95       696
weighted avg       0.95      0.95      0.95       696
```





Hard margin SVM (γ = 10000) performs the best among the three. The soft margin SVM also performs quite well with the accuracy not too much below the hard margin SVM. γ = 100000 results in a slight overfit of data. Hence the accuracy is lesser.

QUESTION 5.

(ii) What happens for the soft margin SVM? Why is the case?

Answer:-

The soft margin SVM also performs quite well with the accuracy not too much below the hard margin SVM. The discrepancy in accuracy between soft margin and hard margin Support Vector Machines (SVMs) stems from their distinct approaches to handling outliers and noise in the training data. The hard margin SVM endeavors to identify a hyperplane that precisely separates the data into distinct classes without permitting any misclassifications. However, this rigid stance makes hard margin SVMs susceptible to the influence of outliers and noise, where even a single misclassified point can significantly alter the position of the hyperplane, leading to diminished accuracy in scenarios with noisy data.

QUESTION 5.

(iii) Does the ROC curve reflect the performance of the soft-margin SVM? Why?

Answer:-

Yes, the ROC curve reflect the performance of the soft-margin SVM. The data is slightly underfitted by the soft margin SVM. The upper left corner of the ROC curve, which is displayed above, supports this.

QUESTION 5.

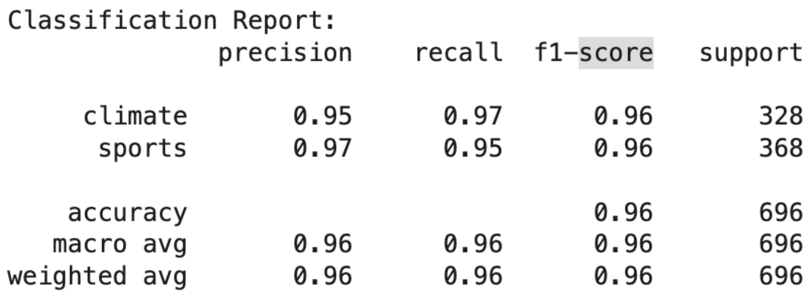
(iv) Use cross-validation to choose γ (use average validation 3 accuracy to compare): Using a 5-fold cross-validation, find the best value of the parameter γ in the range $\{10k| - 3 \leq k \leq 6, k \in Z\}$. Again, plot the ROC curve and report the confusion matrix and calculate the accuracy, recall precision and F-1 score of this best SVM.

Answer:-

Best estimator for SVM: LinearSVC(C=1, random_state=42)
Best parameters for SVM: {'C': 1}
Best Gamma for SVM: 1

For Best Linear SVM:

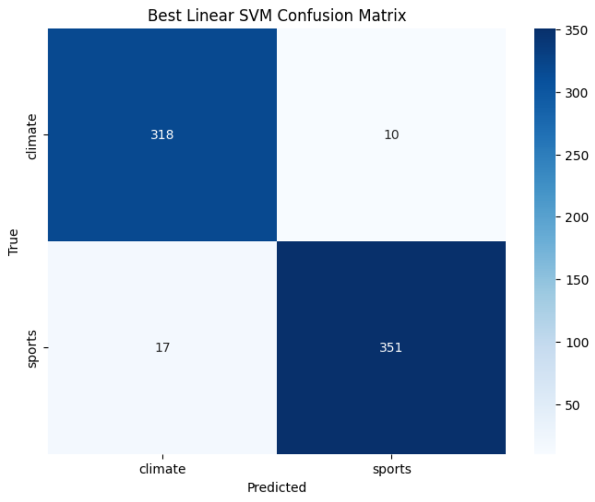
```
Accuracy: 0.9612  Precision: 0.9614
 Recall: 0.9612   F1 Score: 0.9612
```

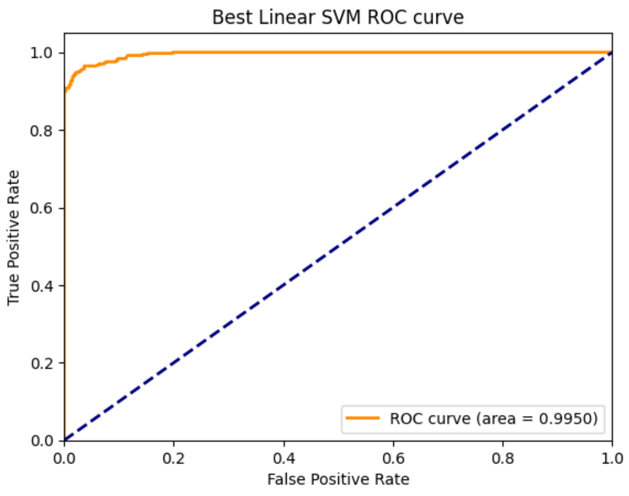


```
Classification Report:
              precision    recall  f1-score   support

     climate       0.95      0.97      0.96       328
      sports       0.97      0.95      0.96       368

    accuracy                           0.96       696
   macro avg       0.96      0.96      0.96       696
weighted avg       0.96      0.96      0.96       696
```





Together with classification measures:

	Accuracy	Recall	Precision	F-1 Score
Hard Margin	0.9497	0.9497	0.9499	0.9497
Soft Margin	0.9454	0.9454	0.9457	0.9453
Best	0.9612	0.9612	0.9614	0.9612

Table 3: Classification Measures for Linear SVMs

2.4.2 Logistic Regression

In logistic regression, a logistic function ($\sigma(\varphi) = 1/(1 + \exp(-\varphi))$) acting on a linear function of the features ($\varphi(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$) is usead to calculate the probability that the data point belongs to class 1, and during the training process, $\mathbf{w}$ and b that maximizes the likelihood of the training data are learnt.

One can also add regularization term in the objective function, so that the goal of the training process is not only maximizing the likelihood, but also minimizing the regularization term, which is often some norm of the parameter vector $\mathbf{w}$. Adding regularization helps prevent ill-conditioned results and over- fitting, and facilitate generalization ability of the classifier.

QUESTION 6.

(i) Train a logistic classifier without regularization. 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:-

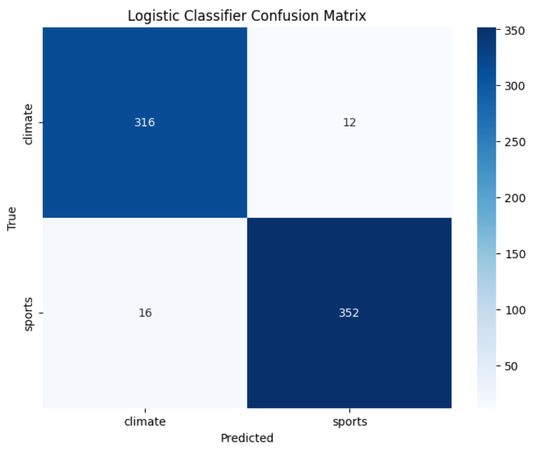
For Logistic Classifier:

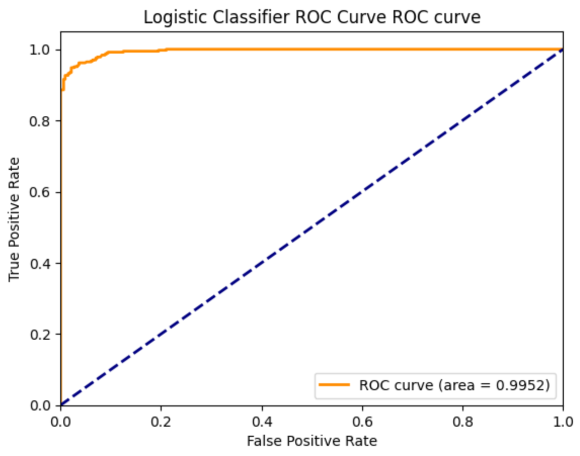
```
Accuracy: 0.9598 Precision: 0.9599
 Recall: 0.9598   F1 Score: 0.9598
```

```
Classification Report:
              precision    recall  f1-score   support

     climate       0.95      0.96      0.96       328
      sports       0.97      0.96      0.96       368

    accuracy                           0.96       696
   macro avg       0.96      0.96      0.96       696
weighted avg       0.96      0.96      0.96       696
```





QUESTION 6.

(ii) Using 5-fold cross-validation on the dimension-reduced-by-SVD training data, find the optimal regularization strength in the range {10k|−5 ≤ k ≤ 5,k ∈ Z} for logistic regression with L1 regularization and logistic regression with L2 regularization, respectively.

Answer:-

- For L1 regularization - the optimal regularization strength is 10.
- For L2 regularization - the optimal regularization strength is 10.

QUESTION 6.

(iii) 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.

Answer:-

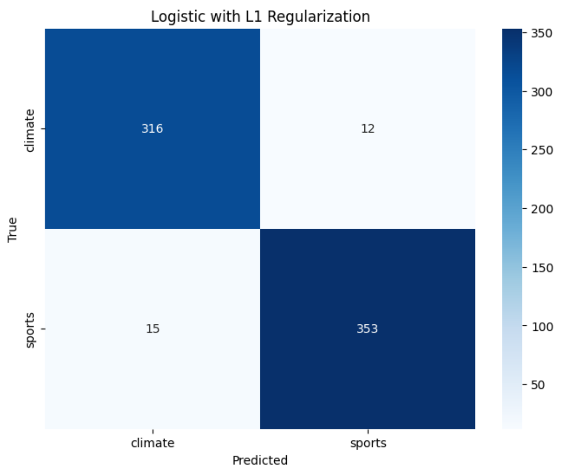
For Logistic Classifier with L1 Regularization:

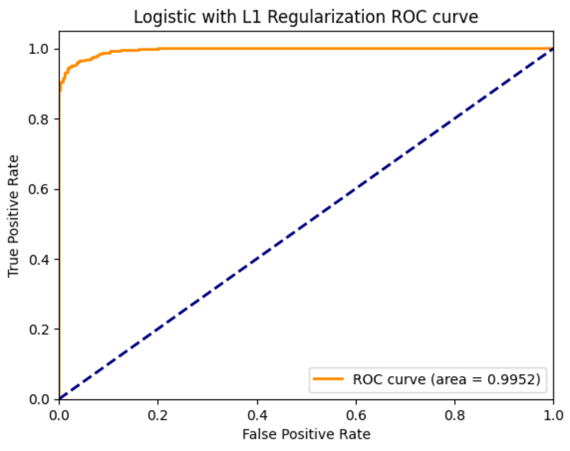
```
Accuracy: 0.9612 Precision: 0.9613
 Recall: 0.9612 F1 Score: 0.9612
```

```
Classification Report:
              precision    recall  f1-score   support

     climate       0.95      0.96      0.96       328
      sports       0.97      0.96      0.96       368

    accuracy                           0.96       696
   macro avg       0.96      0.96      0.96       696
weighted avg       0.96      0.96      0.96       696
```





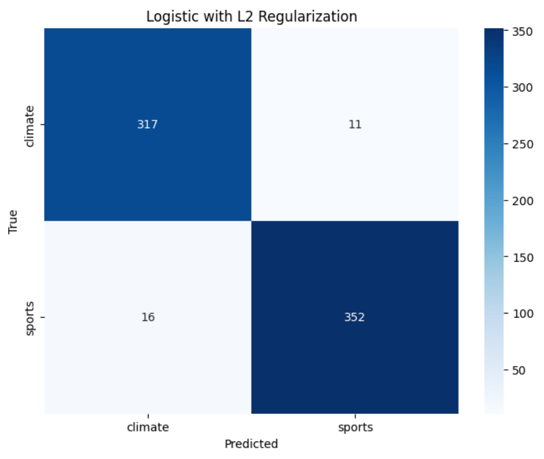
For Logistic Classifier with L2 Regularization:

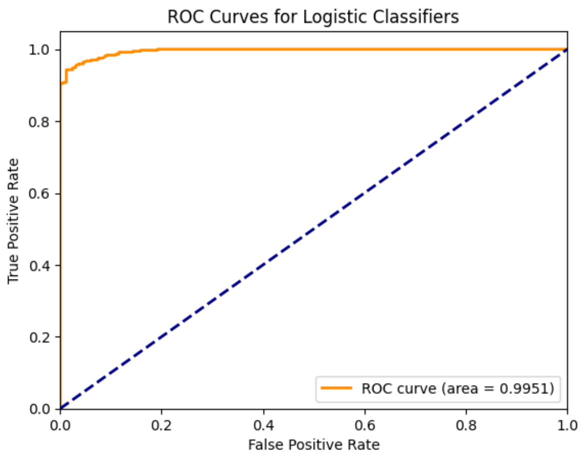
```
Accuracy: 0.9612 Precision: 0.9613
Recall: 0.9612 F1 Score: 0.9612
```

```
Classification Report:
              precision    recall  f1-score   support

     climate       0.95      0.97      0.96       328
      sports       0.97      0.96      0.96       368

    accuracy                           0.96       696
   macro avg       0.96      0.96      0.96       696
weighted avg       0.96      0.96      0.96       696
```





	Accuracy	Recall	Precision	F-1 Score
No Penalty	0.9598	0.9598	0.9599	0.9598
Optimized L1 Penalty	0.9612	0.9612	0.9613	0.9612
Optimized L2 Penalty	0.9612	0.9612	0.9613	0.9612

Table 4: Classification Measures for Logistic Regressions

QUESTION 6.

(iv) 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?

Answer:-

The regularization parameter controls the trade-off between fitting the training data well and avoiding overfitting. A higher regularization parameter may result in a simpler model but could lead to underfitting, while a lower parameter might lead to overfitting. In the case of our model the introduction of L1 or L2 regularization does not lead to significant changes in performance for this specific dataset and evaluation metric set.

Regularization influences the magnitude and distribution of the learned coefficients. L1 regularization may drive some coefficients to exactly zero, effectively performing feature selection. L2 regularization, while penalizing large coefficients, tends to keep all features, but with reduced magnitudes.

One might be interested in L1 regularization because it is usually useful when there's a desire to perform feature selection, as it tends to create sparse models. One might be interested in L2 regularization because it is helpful in preventing overfitting by penalizing large coefficients, providing a smoother distribution of weights across all features.

QUESTION 6.

(v) 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:-

Logistic Regression and Linear Support Vector Machine (SVM) are both linear classifiers that seek to establish a linear decision boundary for classifying data points. Despite their shared objective of linear separation, they differ in their optimization objectives and decision boundary criteria. Logistic Regression employs the logistic loss function, aiming to maximize the likelihood of the observed data. Its decision boundary is chosen to optimize this likelihood, with classification based on a predefined threshold. In contrast, Linear SVM aims to find a hyperplane that maximally separates different classes by minimizing the hinge loss.

The decision boundary, a hyperplane, is selected to maximize the margin between classes, and binary decisions are made based on the side of the hyperplane a point lies. These methods exhibit differences in robustness to outliers, flexibility of the decision boundary, and computational complexity. SVM is generally more robust to outliers due to its emphasis on support vectors and margin maximization.

The statistical significance of their performance difference depends on various factors and may be determined through methods such as cross-validation or hypothesis testing. Practical considerations, such as linear separability of the data, interpretability, computational efficiency, and parameter tuning, influence the choice between Logistic Regression and Linear SVM. It is common to experiment with both and select the one that performs better for a given problem, considering the specific characteristics of the dataset and problem at hand.

2.4.3 Naive Bayes

Naive Bayes classifiers use the assumption that features are statistically independent of each other when conditioned by the class the data point belongs to, to simplify the calculation for the Maximum A Posteriori (MAP) estimation of the labels. That is,

$$\mathrm{P}(x_i|y, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_m) = \mathrm{P}(x_i|y), \quad \forall i = 1, \cdots, m.$$

QUESTION 7.

(v) Evaluate and profile a Na ̈ive 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:-

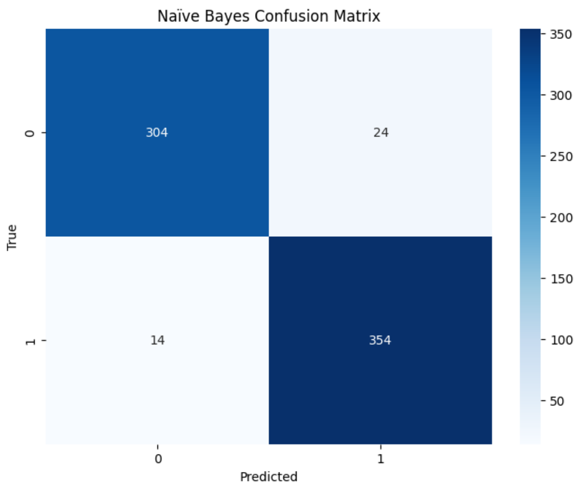
Naïve Bayes Classifier Metrics:

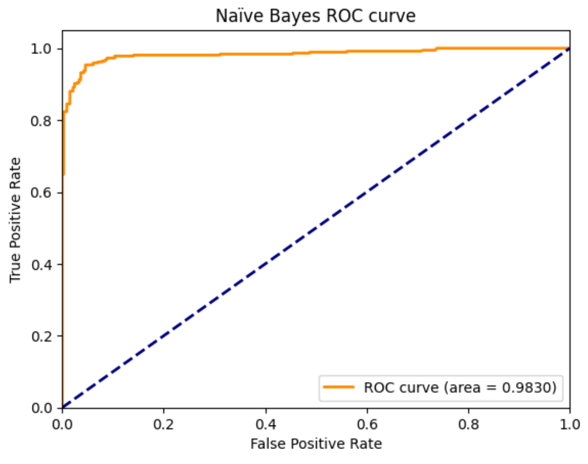
```
Accuracy: 0.9454 Precision: 0.9457
 Recall: 0.9454   F1 Score: 0.9453
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.93      0.94       328
           1       0.94      0.96      0.95       368

    accuracy                           0.95       696
   macro avg       0.95      0.94      0.95       696
weighted avg       0.95      0.95      0.95       696
```





	Accuracy	**Recall**	**Precision**	**F-1 Score**
GaussianNB	0.9454	0.9454	0.9457	0.9454

Table 5: Classification Measures for Naive Bayes

Remark. In terms of all classification measures, Naive Bayes classifier performs worse compared to both linear SVM and logistic regression. One major drawback of this model is the independence assumption, since for this particular textual dataset, some vocabularies tend to appear together more often than others, leading to dependency among features.

2.5 Grid Search of Parameters

Now we have gone through the complete process of training and testing a classifier. However, there are lots of parameters that we can tune. In this part, we fine-tune the parameters.

QUESTION 8.

- Construct a Pipeline that performs feature extraction, dimensionality reduction and classifycation;
- The evaluation of each combination is performed with 5-fold cross-validation (use the average validation set accuracy across folds).
- In addition to any other hyperparameters you choose, your gridsearch must at least include:

Table 1: Minimum set of hyperparameters to consider for pipeline comparison

Module	**Options**
Loading Data	Clean the data
Feature Extraction	`min_df = 3` vs `5` while constructing the vocabulary; AND use Lemmatization vs Stemming as a compression module
Dimensionality Reduction	LSI ($k = [5, 30, 80]$) vs NMF ($k = [5, 30, 80]$)
Classifier	SVM with the best γ previously found vs Logistic Regression: L1 regularization vs L2 regularization, with the best regularization strength previously found vs `GaussianNB` Note: You can once again find the optimal hyperparameters for each classifier, but this is not *required.*
Other options	Use default

- What are the 5 best combinations? Report their performances on the testing set.

Answer:-

To accomplish the objective, we construct a pipeline and perform 5-fold cross-validation to grid-search for the optimal parameter settings in terms of highest mean testing accuracy score. The best 5 parameter combinations are reported in the following table:

mean_test_score	**classifier**	**reduce_dim**	**min_df**	**lemmatized**	**stemming**
0.960432	L2 Logistic(C=1000)	LSI	5	TRUE	FALSE
0.960072	Linear SVC(C=1)	LSI	5	FALSE	TRUE
0.959712	Linear SVC(C=1)	LSI	3	TRUE	FALSE
0.959712	L2 Logistic(C=1000)	NMF	3	FALSE	TRUE
0.959712	L2 Logistic(C=1000)	LSI	5	FALSE	TRUE

Table 6: Best Five Parameter Combinations

Therefore, the best combination is using lemmatization and setting the parameter min df=5 for CountVectorizer, reducing the dimensionality through LSI and selecting the linear Logistic Regression with L2 regularization as the classifier.

Based on the aforementioned grid search result, we better understand that in order to increase the classification accuracy, one should prefer LSI compared to NMF in dimensionality reduction. Moreover, larger min df and lemmatization tends to increase the model performance.

3 Multiclass Classification

So far, we have been dealing with classifying the data points into two classes. In this part, we explore multiclass classification techniques through different algorithms.

Some classifiers perform the multiclass classification inherently. As such, Naive Bayes algorithm finds the class with maximum likelihood given the data, regardless of the number of classes. In fact, the probability of each class label is computed in the usual way, then the class with the highest probability is picked; that is

$$\hat{c} = \arg\min_{c \in C} \mathrm{P}(c|\mathbf{x})$$

For SVM, however, one needs to extend the binary classification techniques when there are multiple classes by adopting either an one-vs-one or one-vs-rest approach.

For the one-vs-one classification, we give a document the class that is assigned with the majority vote. In case there is more than one class with the highest vote, the class with the highest total classification confidence levels in the binary classifiers is picked.

For the one-vs-rest approach, we fit one classifier per class against all the other classes. Note that in this case, the unbalanced number of documents in each class should be handled. By learning a single classifier for each class, one can get insights on the interpretation of the classes based on the features.

QUESTION 9.

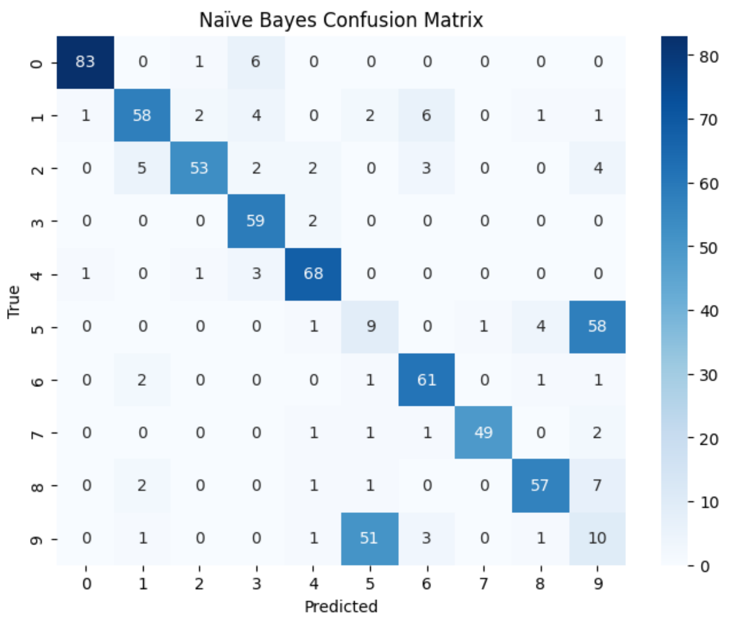
(i) Perform Na¨ive Bayes classification and multiclass SVM classification (with both One VS One and One VS the rest methods described above) and report the confusion matrix and calculate the accuracy, recall, precision and F-1 score of your classifiers.

Answer

We use GaussianNB to build up multiclass Naive Bayes classifier, and OneVsOneClassifier(SVC()), OneVsRestClassifier(SVC()) to construct One-Vs-One and One-Vs-Rest multiclass SVM classification respectively.

Naive Bayes Classifier Metrics:

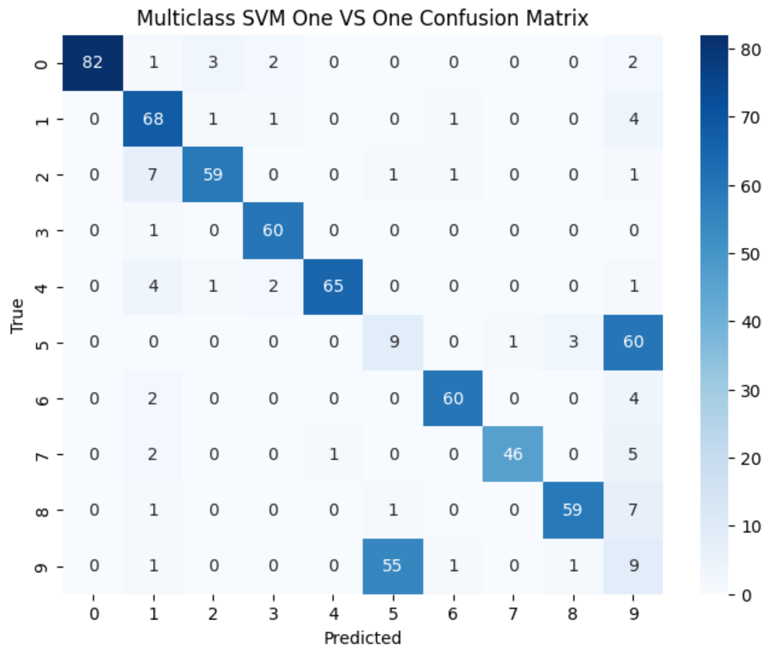
```
Accuracy: 0.7284.  Recall: 0.7284
Precision: 0.7414. F1 Score: 0.7325
```



	0	1	2	3	4	5	6	7	8	9
0	83	0	1	6	0	0	0	0	0	0
1	1	58	2	4	0	2	6	0	1	1
2	0	5	53	2	2	0	3	0	0	4
3	0	0	0	59	2	0	0	0	0	0
4	1	0	1	3	68	0	0	0	0	0
5	0	0	0	0	1	9	0	1	4	58
6	0	2	0	0	0	1	61	0	1	1
7	0	0	0	0	1	1	1	49	0	2
8	0	2	0	0	1	1	0	0	57	7
9	0	1	0	0	1	51	3	0	1	10

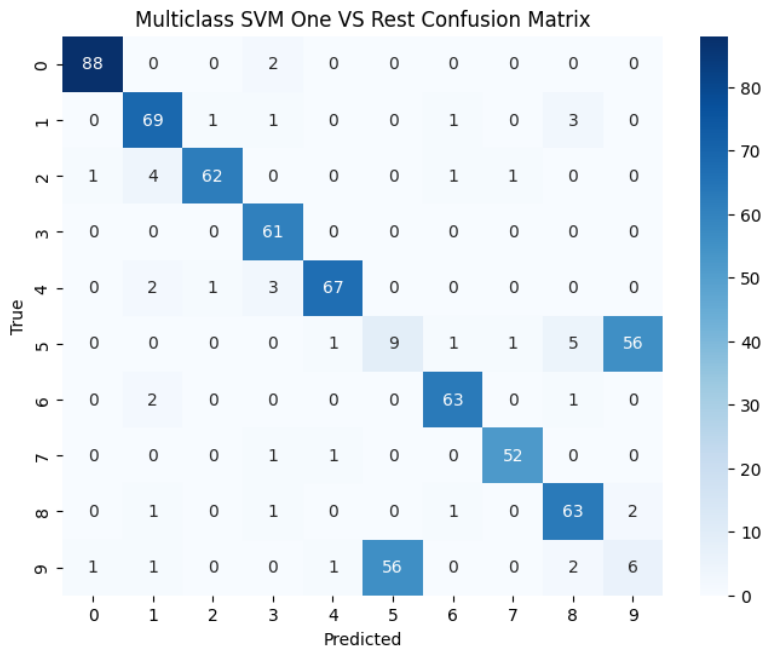
Multiclass SVM Classification (One VS One):

```
 Accuracy: 0.7428.   Recall: 0.7428
Precision: 0.7705.  F1 Score: 0.7546
```



Multiclass SVM Classification (One VS Rest):

```
 Accuracy: 0.7759.   Recall: 0.7759
Precision: 0.7651.  F1 Score: 0.7697
```



QUESTION 9.

(ii) In the confusion matrix you should have a 10 × 10 matrix where 10 is the number of unique labels in the column leaf label. Do you observe any structure in the confusion matrix? Are there distinct visible blocks on the major diagonal? What does this mean?

Answer:-

The structure of a confusion matrix with distinct visible blocks on the major diagonal indicates that the classifier is performing well for certain classes but is struggling to differentiate between those two specific classes. The major diagonal of a confusion matrix represents the true positive predictions for each class, and the off-diagonal elements represent misclassifications. In this case, the difficulty in separating the two classes is reflected in the off-diagonal elements corresponding to forest fire and heatwave classes. It suggests that the classifier is frequently confusing instances of those two classes with each other.

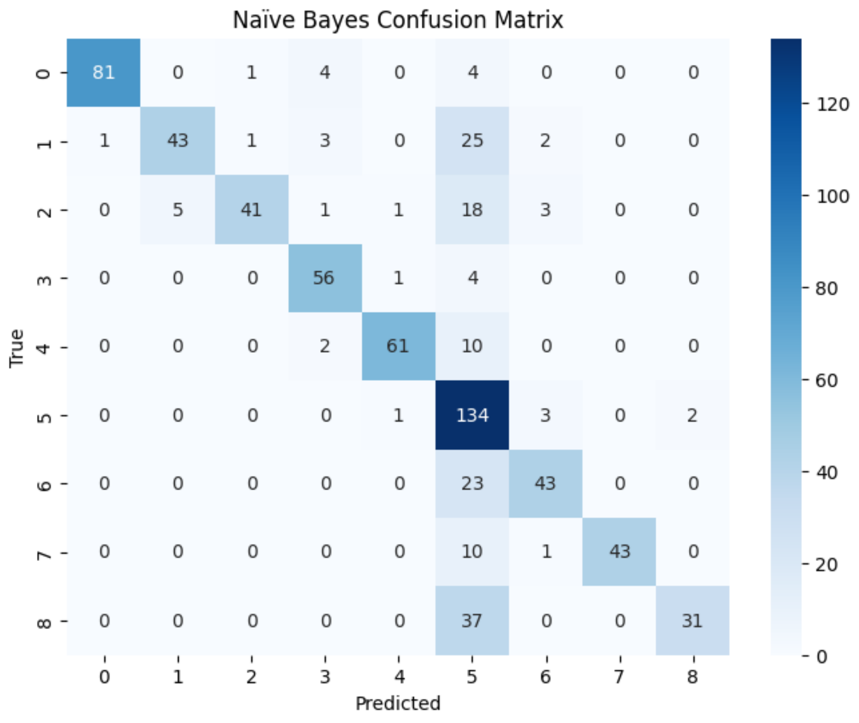
QUESTION 9.

(iii) 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?

Answer:-

Naïve Bayes Classification Results:

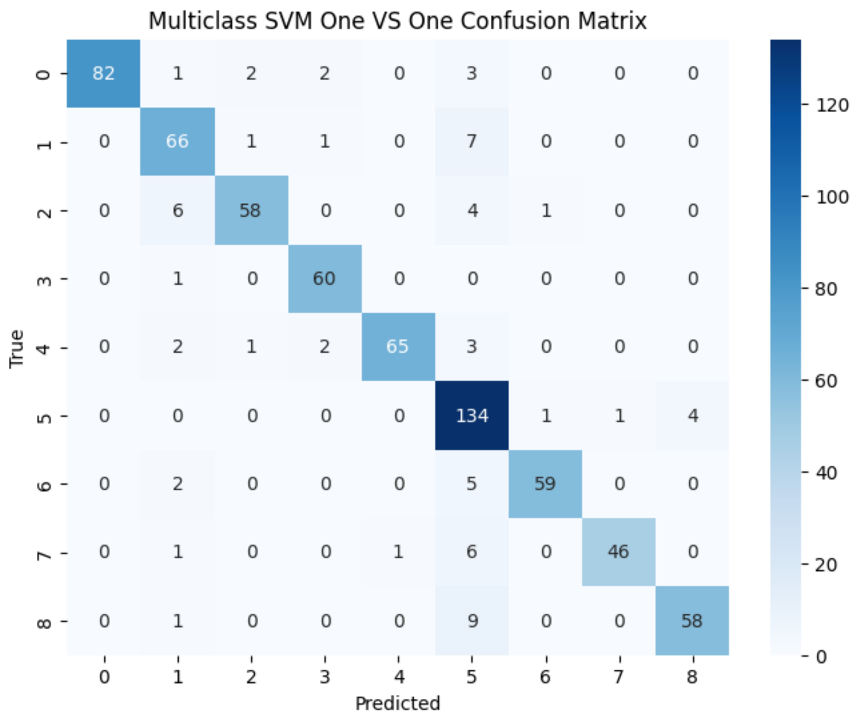
```
Accuracy: 0.7658.   Recall: 0.7658
Precision: 0.8426.  F1 Score: 0.7714
```



True \ Predicted	0	1	2	3	4	5	6	7	8
0	81	0	1	4	0	4	0	0	0
1	1	43	1	3	0	25	2	0	0
2	0	5	41	1	1	18	3	0	0
3	0	0	0	56	1	4	0	0	0
4	0	0	0	2	61	10	0	0	0
5	0	0	0	0	1	134	3	0	2
6	0	0	0	0	0	23	43	0	0
7	0	0	0	0	0	10	1	43	0
8	0	0	0	0	0	37	0	0	31

Multiclass SVM Classification (One VS One) Results:

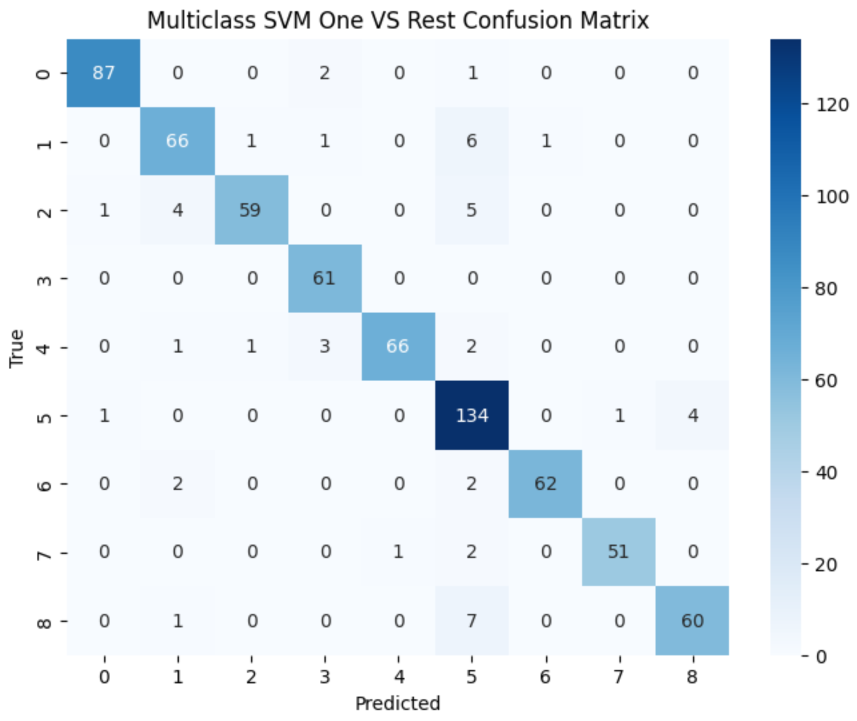
```
 Accuracy: 0.9023.   Recall: 0.9023
Precision: 0.9118.  F1 Score: 0.9037
```



True \ Predicted	0	1	2	3	4	5	6	7	8
0	82	1	2	2	0	3	0	0	0
1	0	66	1	1	0	7	0	0	0
2	0	6	58	0	0	4	1	0	0
3	0	1	0	60	0	0	0	0	0
4	0	2	1	2	65	3	0	0	0
5	0	0	0	0	0	134	1	1	4
6	0	2	0	0	0	5	59	0	0
7	0	1	0	0	1	6	0	46	0
8	0	1	0	0	0	9	0	0	58

Multiclass SVM Classification (One VS Rest) Results:

```
 Accuracy: 0.9282.   Recall: 0.9282
Precision: 0.9320.  F1 Score: 0.9285
```



True \ Predicted	0	1	2	3	4	5	6	7	8
0	87	0	0	2	0	1	0	0	0
1	0	66	1	1	0	6	1	0	0
2	1	4	59	0	0	5	0	0	0
3	0	0	0	61	0	0	0	0	0
4	0	1	1	3	66	2	0	0	0
5	1	0	0	0	0	134	0	1	4
6	0	2	0	0	0	2	62	0	0
7	0	0	0	0	1	2	0	51	0
8	0	1	0	0	0	7	0	0	60

Based on my observation from the previous analysis, it appears that the model is encountering confusion between the classes "heatwave" and "forest fire." Considering this, I merged these two specific labels into a new, broader category. By doing so, the model benefited from a more generalized understanding of phenomena associated with extreme temperatures and fires, potentially reducing the confusion and improving overall classification performance. This consolidation enhanced the model's ability to discern between instances of heatwaves and forest fires, providing a clearer and more comprehensive classification for related events. The accuracy of the One VS One and One VS the rest models improved drastically. Both were able to classify the 10 labels with significant accuracy although One VS the rest model performing slightly better that the One VS One model.

QUESTION 9.

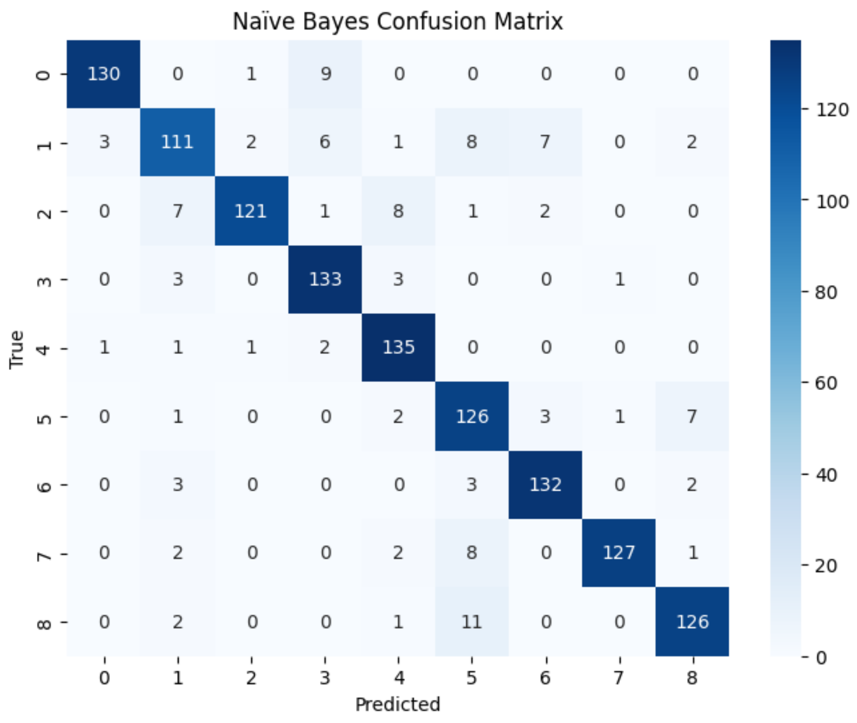
(iv) Does class imbalance impact the performance of the classification once some classes are merged? Provide a resolution for the class imbalance and recompute the accuracy and plot the confusion matrix in One VS One and One VS the rest?.

Answer:-

Class imbalance did indeed impact the performance of classification models, even when certain classes are merged, potentially affecting the model's ability to distinguish between imbalanced categories to a certain extent. To address this, resolving class imbalance is crucial. Hence, I used SMOTE to resolve the issue by oversampling minorities.

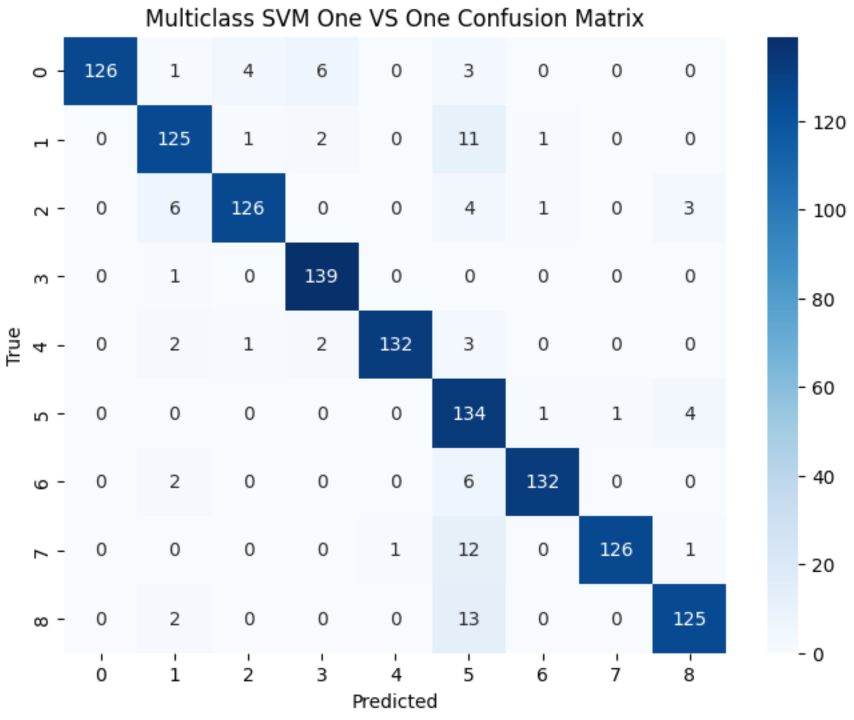
Naïve Bayes Classification Results:

```
 Accuracy: 0.9056.   Recall: 0.9056
Precision: 0.9086.  F1 Score: 0.9058
```



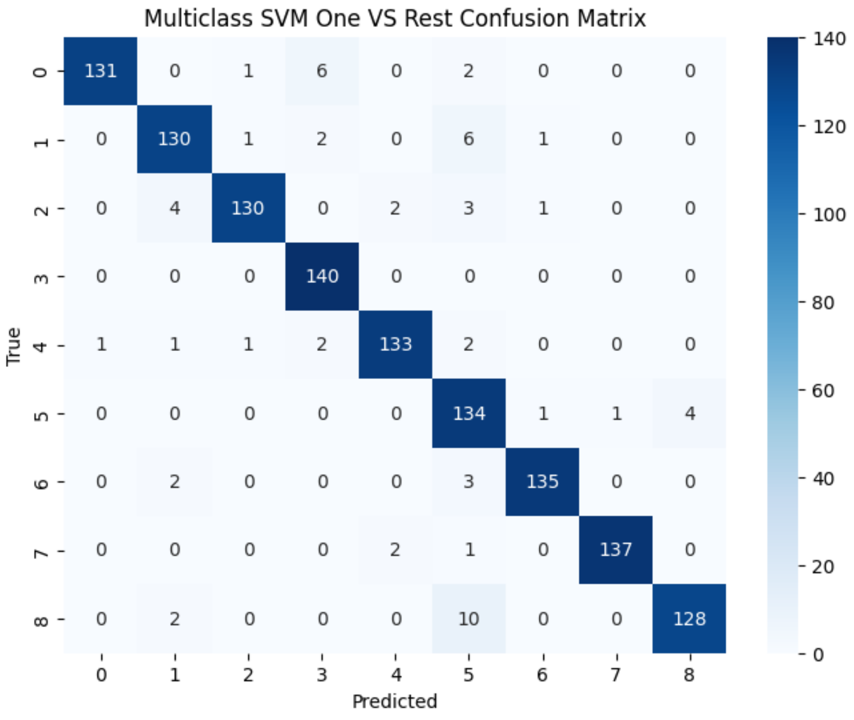
Multiclass SVM Classification (One VS One) Results:

```
 Accuracy: 0.9246.   Recall: 0.9246
Precision: 0.9344.  F1 Score: 0.9267
```



Multiclass SVM Classification (One VS Rest) Results:

```
 Accuracy: 0.9508.   Recall: 0.9508
Precision: 0.9536.  F1 Score: 0.9513
```



4 Word Embeddings

QUESTION 10.

(i) Why are GLoVE embeddings trained on the ratio of co-occurrence probabilities rather than the probabilities themselves?

Answer:-

- The choice of using the ratio of co-occurrence probabilities in the training of GLoVE (Global Vectors for Word Representation) embeddings is motivated by the desire to capture more meaningful semantic relationships between words. GLoVE focuses on the relative frequency of word co-occurrences, which helps in emphasizing the semantic relationships between words while mitigating the impact of common words that co-occur frequently with many other words.

- When training word embeddings, the raw probabilities can be influenced by the overall frequency of individual words. Words that occur frequently across the entire corpus may dominate the training process and overshadow the subtler semantic relationships. By considering the ratio of co-occurrence probabilities, GLoVE aims to emphasize the information about how words co-occur relative to each other, providing a more nuanced representation of semantic similarities.

QUESTION 10.

(ii) 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?

Answer:-

- No, GLoVE embeddings would not return the exact same vector for the word "running" in both sentences. GLoVE captures word semantics based on co-occurrence statistics, and the context in which a word appears plays a crucial role in determining its vector representation.

- In the two sentences, the word "running" is used in different contexts ("in the park" vs. "for the presidency"). GLoVE takes into account the distributional information of words, meaning that words that have similar contexts and co-occur with similar sets of words will have similar embeddings.

- Since the context surrounding "running" is different in each sentence, GLoVE is likely to assign distinct vector representations to the word in these contexts. The resulting vectors would capture the semantic nuances associated with "running" in a park versus "running" for the presidency, reflecting the different ways the word is used in each context.

QUESTION 10.

(iii) 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.

Answer:-

The value of ||GLoVE["woman"] - GLoVE["man"]||2 :- 4.7539396
The value of ||GLoVE["wife"] - GLoVE["husband"]||2 :- 3.1520464
The value of ||GLoVE["wife"] - GLoVE["orange"]||2 ? :- 8.427546

QUESTION 10.

(iv) Given a word, would you rather stem or lemmatize the word before mapping it to its GLoVE embedding?

Answer:-

In the context of mapping words to GLoVE embeddings, lemmatization is often preferred because it tends to retain more semantic information. However, the choice between stemming and lemmatization may also depend on the specifics of your application. If you are working with a domain where retaining the root form is crucial and you can accept some loss of precision, stemming might be more suitable.

QUESTION 11.

(i) Describe a feature engineering process that uses GLoVE word embeddings to represent each document. You have to abide by the following rules:

- A representation of a text segment needs to have a vector dimension that CANNOT exceed the dimension of the GLoVE embedding used per word of the segment.
- You cannot use TF-IDF scores (or any measure that requires looking at the complete dataset) as a pre-processing routine.
- Important: In this section, feel free to use raw features from any column in the original data file not just full text. The column keywords might be useful... or not. Make sure that your result achieves an accuracy of at least 92%.
- To aggregate these words into a single vector consider normalization the vectors, averaging across the vectors

Answer:-

We convert the GLoVE word embeddings to a word2vec file by using a keyed vector model from Gensim. The words are then stored in embedded vector format. The data is cleaned and we remove the punctuations, digits and stopwords before lemmatizing it. Then each sentence is converted into a vector of size 300 as mentioned in the question. We then need to check if all the features which are there in the vector are based in the pretrained GLoVE embedding file. Then the vectors are normalized and aggregated into a single vector.

QUESTION 11.

(ii) Select 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.

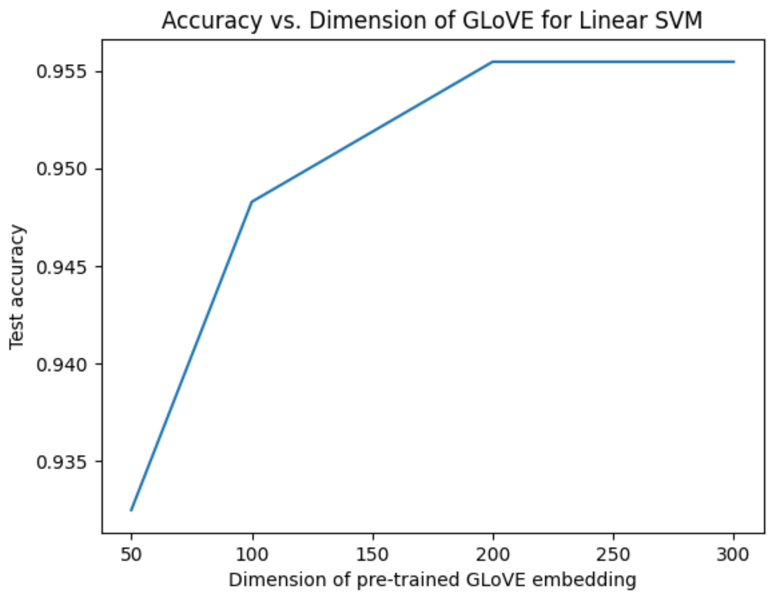
Answer:-

```
Accuracy (Best GLoVE classifier): 0.9554597701149425
 Recall (Best GLoVE classifier): 0.9510869565217391
Precision (Best GLoVE classifier): 0.9641873278236914
F1-Score (Best GLoVE classifier): 0.957592339261286
```

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:-

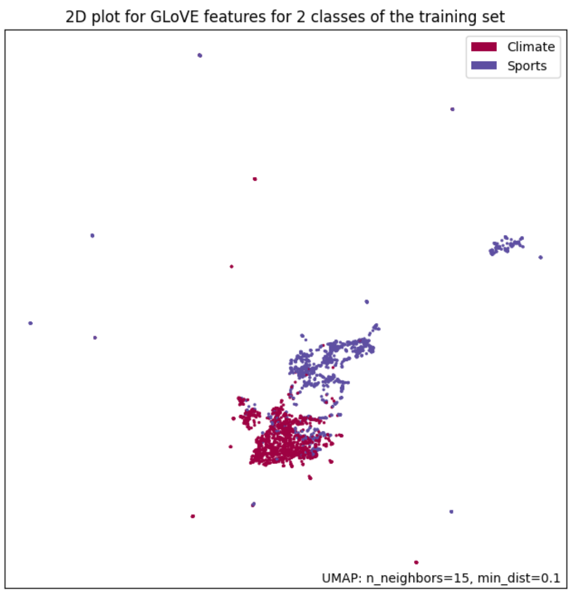


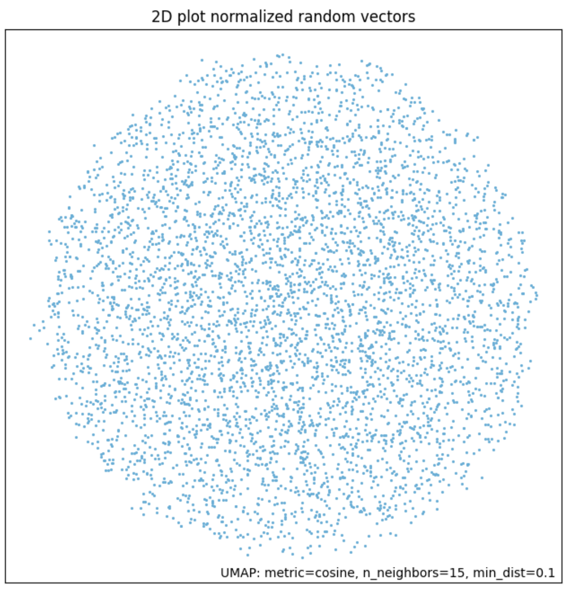
The graph between the test accuracy and the pretrained GLoVE embeddings' dimensions is displayed in the above figure. This shows that the accuracy of the test data grows with increasing size. More dimensions imply more information and greater semantics.

QUESTION 13.

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

Answer:-





We see that The GLoVE model, with its focus on semantic relationships between words, tends to create clusters in the plot. On the other hand, when I generate random vectors of the same dimension as GLoVE embeddings and visualize them with UMAP, the lack of inherent meaning becomes apparent. The plot of random vectors typically appears scattered and lacks the meaningful clusters observed in the GLoVE-based embeddings plot.