Large Scale Data Mining: Models and Algorithms

ECE 219 Spring 2023

Instructor: Professor Vwani Roychowdhury

UCLA, Department of ECE

Project 1 Report: End-to-End Pipeline to Classify News Articles

---

Group Member

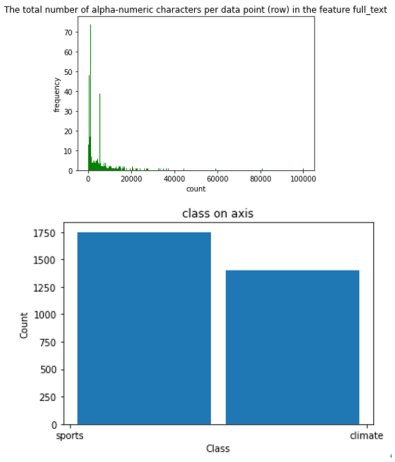
Syed R Nawshad	*StudentID* : 805871517
Huijun S Hao	*StudentID* : 105863846
Chao Zhang	*StudentID* : 904506124

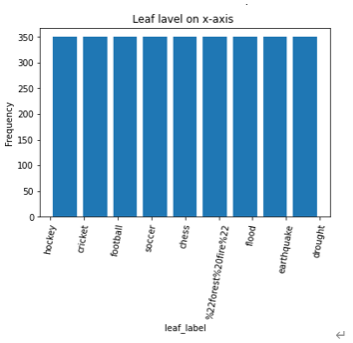
April 22, 2023

Getting familiar with the dataset

Question 1

-Number of rows: 3150 -Number of columns: 8





For the first histogram we can observe the maximum frequency of alphanumeric character was higher than 70. The number of alphanumeric characters has big difference across the dataset. Most alphanumeric characters are between 0 to 20,000. There are still some outliers that fall in 40,000 to 60,000.

For 2nd and 3rd histogram we can see class distribution of the data. There are 2 classes and 9 leaf labels for this dataset. Data is evenly distributed between all the root labels and leaf labels. 1750 data belongs to sports class and 1400 belongs to climate class. Since class is balanced here so we can apply classification model directly.

Binary Classification

Question 2

A train-test split was performed with the training samples containing 2520 samples and the testing samples comprising of 630 samples.

Question 3

Both lemmatization and stemming are normalizing techniques. Lemmatization uses vocabulary and morphological analysis to reduce words to their base form. This method is more accurate than stemming, but more cost heavy. Lemmatization also produces more readable results since it converts words to their base forms while still maintaining their original meaning. Usually it takes longer than using stemming. Stemming simply cuts off the ends of words to reduce them to their base forms. This method will lead to more errors than lemmatization but will be faster and computationally cheaper. Dictionary size will be smaller for lemmatization when compared with stemming as the words will be mapped correctly to the roots.

The purpose of min df is to remove the infrequent term. When we increase the min df, it will decrease the column counts of the matrix.

We should remove Stopwords, punctuation and numbers before lemmatizing. By doing that we can reduce some computation cost.The lemmatizer uses the context of the sentence and the POS tagging of the words to determine the correct base form of a word. So removing punctuations, numbers and stopwords before lemmatizing will not affect the accuracy of the lemmatization and will make the process more efficient with lower computation cost.

The shape of the TF-IDF processed train and test matrices are as follows:
Pulling the min df from 1 to 4.
when min df = 1
train matrix shape: (2520, 34769)
test matrix shape: (630, 34769)

when min df = 2
train matrix shape: (2520, 19362)
test matrix shape: (630, 19362)

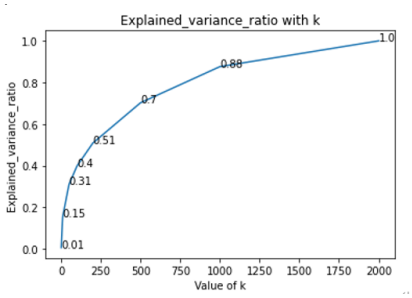
when min df = 3
train matrix shape: (2520, 14076)
test matrix shape: (630, 14076)

when min df = 4
train matrix shape: (2520, 11330)
test matrix shape: (630, 11330)
As we can see from the result when min df from 1 to 4 the row of train and test

don't change but the columns decreased from 34769 to 11330.

Question 4

The dimensionality of the data was reduced using latent semantic indexing (LSI) and Non-negative Matrix Factorization (NMF). The explained variance ratio was also plotted against multiple values of k as shown below:

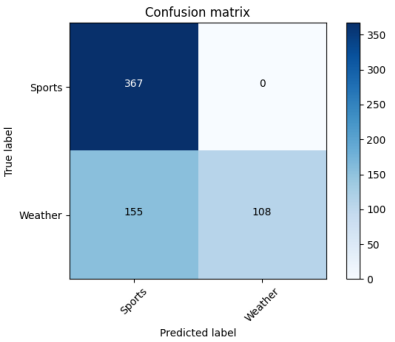


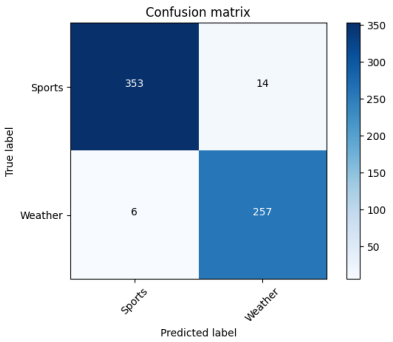
As the number of components increase, the explained variance ratio will increase as well. For a given k, LSI chooses the top k components in the descending order of the variance ratio that the component explains. From the plot we can see that around 80 percent of the variance is covered by around 1000 features and about 100 percent when features equal to 2000. The concavity indicate that as values of k increase, the variance also increases. But the increase gets slower with larger k. For example, the difference between the variance explained for k=200 and k=500 is more than that of k=1000 and k=2000. That's because the components are chosen in decreasing order of their variance explanation ratio.
- MSE Error for LSI is 41.02637078289104
- MSE Error for NMF is 41.37286730025996
MSE Error for NMF is larger.That's because of NMF calculates how well each document fits each topic, and LSI is a more insightful method by giving us more information with the SVD factorization So NMF is larger than LSI is expected.

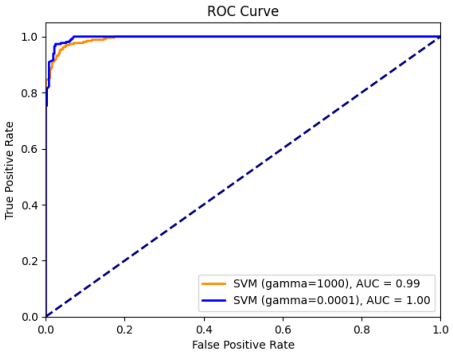
Question 5

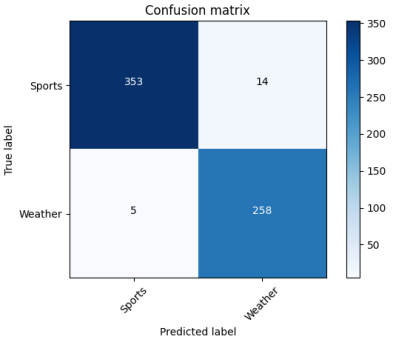
Two linear SVMs were trained: one with a hard margin ($\gamma = 1000$) and one with a soft margin ($\gamma = 0.0001$). The following results were obtained:

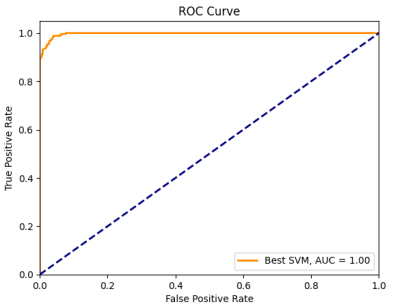
The left confusion matrix is for the SVM with a hard margin and the right confusion matrix is for the SVM with a soft margin.
After this, cross validation was used to choose γ. Using a 5-fold cross validation, the best value for γ was found to be 10 with the following results:
- Accuracy Score: 0.970
- Recall Score: 0.967
- Precision Score: 0.971
- F-1 Score: 0.969





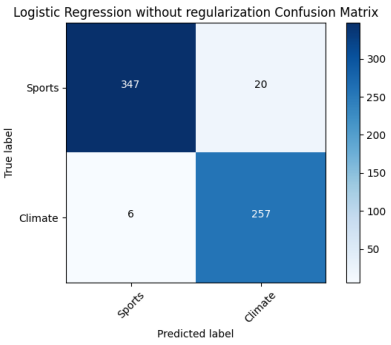


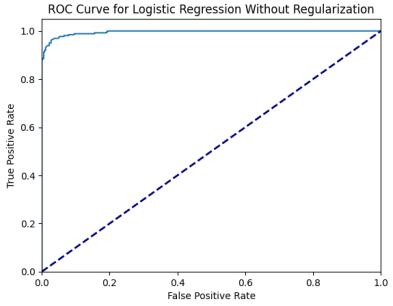




Question 6

A logistic classifier was trained on the processed data without regularization and the following results were received:
- Accuracy Score: 0.959
- Recall Score: 0.977
- Precision Score: 0.928
- F-1 Score: 0.952





Using 5-fold cross-validation on the dimension-reduced-by-SVD training, the optimal regularization strength was found for both L1 and L2 regularization. For L1, the optimal regularization was found to be 10 and for L2, the optimal regularization was found to be 1000:

L1 Reg Scores

L1 Value	Accuracy	Standard Deviation
1e-05	0.5488095238095239	0.0009720197391996703
0.0001	0.5488095238095239	0.0009720197391996703
0.001	0.5488095238095239	0.0009720197391996703
0.01	0.5488095238095239	0.0009720197391996703
0.1	0.923015873015873	0.017808575930410207
1	0.9424603174603174	0.01011710220950946
10	0.9519841269841269	0.004593586072535839
100	0.9519841269841269	0.00580505509457455
1000	0.9511904761904761	0.0071648690814555
10000	0.9511904761904761	0.0071648690814555
100000	0.9511904761904761	0.0071648690814555

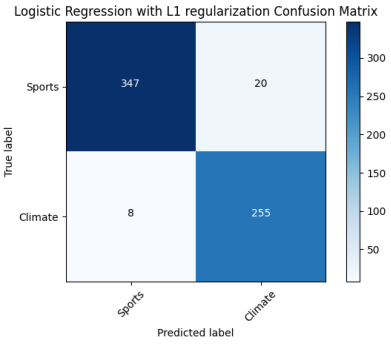
Best l1 is 10

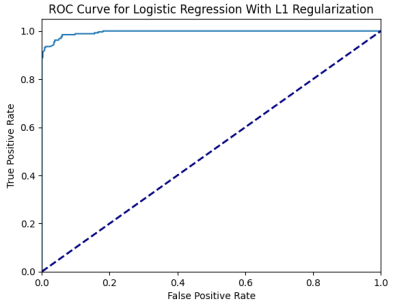
L2 Reg Scores

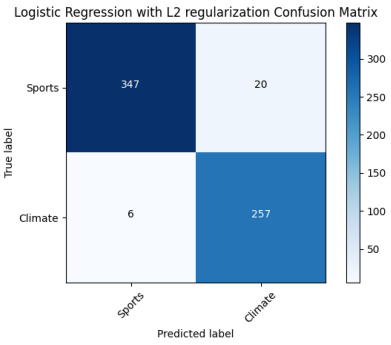
L2 Value	Accuracy	Standard Deviation
1e-05	0.5488095238095239	0.0009720197391996703
0.0001	0.5488095238095239	0.0009720197391996703
0.001	0.5488095238095239	0.0009720197391996703
0.01	0.7273809523809524	0.007591716853059096
0.1	0.9226190476190477	0.015110660924333148
1	0.9436507936507936	0.009928568256505392
10	0.9472222222222223	0.006223963151332616
100	0.9492063492063492	0.005112737589970301
1000	0.9515873015873015	0.00609614741894337
10000	0.9511904761904761	0.0071648690814555
100000	0.9511904761904761	0.0071648690814555

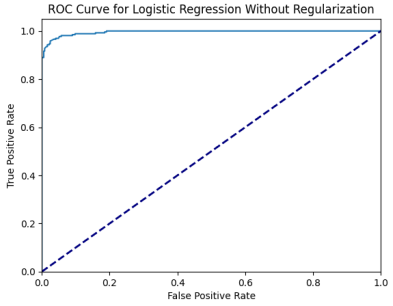
Best l2 is 1000

The 3 logistic classifiers: without regularization, with L1 regularization (with best L1), and with L2 regularization (wih best l2) were compared against each other on the test data:









For the logistic classifier with L1 regularization the following results were received:
- Accuracy Score: 0.956
- Recall Score: 0.970
- Precision Score: 0.927
- F-1 Score: 0.948

For the logistic classifier with L2 regularization the following results were received:
- Accuracy Score: 0.959
- Recall Score: 0.977
- Precision Score: 0.928
- F-1 Score: 0.952

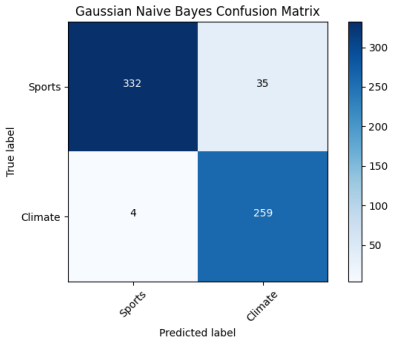
The regularization parameter in both logistic regression and linear SVM classifiers controls the trade-off between model complexity and model performance. A higher regularization parameter leads to a simpler model with smaller coefficients, while a lower regularization parameter leads to a more complex model with larger coefficients.
In general, increasing the regularization parameter tends to decrease the test error of the model. This is because a higher regularization parameter penalizes large coefficient values, which can reduce the overfitting of the model to the training data and improve its ability to generalize to new data. However, if the regularization parameter is too high, the model may become too simple and underfit the data, leading to a high test error.
The learnt coefficients of the model are affected by the regularization parameter. A higher regularization parameter leads to smaller coefficients, as the model is penalized for having large coefficients. This can help to prevent overfitting and improve the generalization performance of the model. On the other hand, a lower regularization parameter allows for larger coefficients, which may improve the fit of the model to the training data but may lead to overfitting and poor generalization performance.

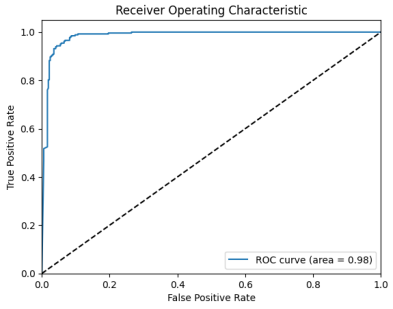
Logistic regression and linear SVM both use a linear decision boundary to classify data points into two or more classes. However, there are some differences in the ways they find this boundary and their performance.
Logistic regression models the probability of the target variable being in a certain class given the features. It uses a sigmoid function to transform the linear

combination of features into a probability value between 0 and 1. The goal of logistic regression is to find the coefficients of the linear combination of features that maximizes the likelihood of the data given the model. This is usually done using maximum likelihood estimation.
Linear SVM, on the other hand, tries to find the hyperplane that best separates the data points into different classes while maximizing the margin between the hyperplane and the closest data points. The margin is defined as the distance between the hyperplane and the closest points from each class. The goal of linear SVM is to find the hyperplane that maximizes the margin while also correctly classifying as many data points as possible.

Question 7

A Gaussian Naive Bayes Classifier was trained with the following results on the test set:
- Accuracy Score: 0.938
- Recall Score: 0.985
- Precision Score: 0.881
- F-1 Score: 0.930





Question 8

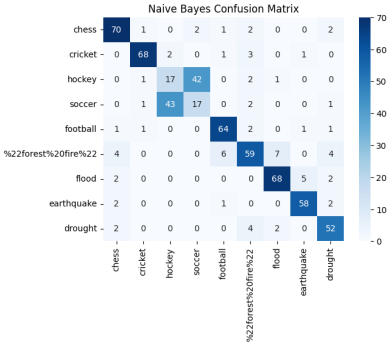
For this section, a pipeline was constructed that performs feature extraction, dimensionality reduction and classification. The evaluation of each combination is performed with a 5-fold cross validation using the average validation set accuracy across folds.
The following combinations were found to be the top 5 accuracy's:

Combination	Accuracy	Vectorizer Min DF	Vectorizer Tokenizer	Dim. Reduction	Dim. Reduction Components	Classification
1	0.9539682539682539	5	None	NMF(n_components=80, random_state=42)	80	GaussianNB()
2	0.9515873015873015	3	None	NMF(n_components=80, random_state=42)	80	GaussianNB()
3	0.9488095238095238	5	None	TruncatedSVD(random_state=42)	80	SVC(C=10, gamma=0.1)
4	0.9472222222222222	3	None	TruncatedSVD(random_state=42)	80	SVC(C=10, gamma=0.1)
5	0.9456349206349206	5	None	TruncatedSVD(random_state=42)	80	LogisticRegression(C=1)

Multiclass Classification

Question 9

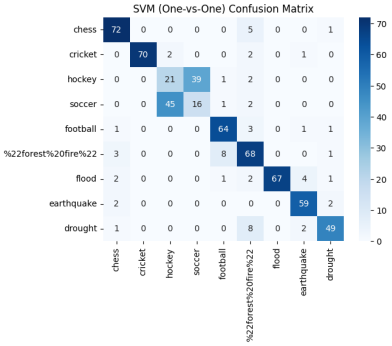
For this question, Naive Bayes and SVM (both One-vs-One and One-vs-Rest) classifiers were trained on the leaf labels as the label column to predict the leaf labels. The following results were received for the Naive Bayes classifier:



Naive Bayes Classification Report:

Class	precision	recall	f1-score	support
chess	0.8641975308641975	0.8974358974358975	0.8805031446540880	78
cricket	0.9444444444444444	0.9066666666666666	0.9251700680272109	75
hockey	0.2741935483870675	0.2698412698412698	0.272	63
soccer	0.2786885245901639	0.265625	0.272	64
football	0.8767123287671232	0.9142857142857143	0.8951048951048951	70
%22forest%20fire%22	0.7972972972972973	0.7375	0.7662337662337663	80
flood	0.8717948717948718	0.8831168831168831	0.8774193548387098	77
earthquake	0.8923076923076924	0.9206349206349206	0.90625	63
drought	0.8125	0.8666666666666667	0.8387096774193549	60

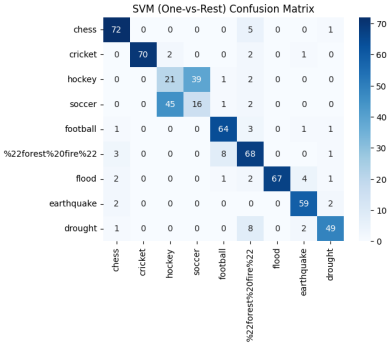
The following results were received for the SVM (One-vs-One) classifier:



SVM (One-vs-One) Classification Report:

Class	precision	recall	f1-score	support
chess	0.8888888888888888	0.9230769230769231	0.9056603773584906	78
cricket	1.0	0.9333333333333333	0.9655172413793104	75
hockey	0.3088235294117647	0.3333333333333333	0.3206106870229008	63
soccer	0.2909090909090909	0.25	0.2689075630252101	64
football	0.8533333333333334	0.9142857142857143	0.8827586206896552	70
%22forest%20fire%22	0.7391304347826086	0.85	0.7906976744186046	80
flood	1.0	0.8701298701298701	0.9305555555555556	77
earthquake	0.8805970149253731	0.9365079365079365	0.9076923076923077	63
drought	0.8909090909090909	0.8166666666666667	0.8521739130434782	60

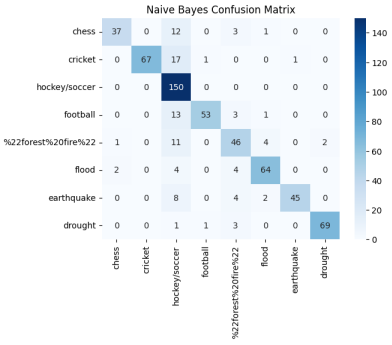
Finally, the following results were received for the SVM (One-vs-Rest) classifer:



SVM (One-vs-Rest) Classification Report:

Class	precision	recall	f1-score	support
chess	0.8888888888888888	0.9230769230769231	0.9056603773584906	78
cricket	1.0	0.9333333333333333	0.9655172413793104	75
hockey	0.3088235294117647	0.3333333333333333	0.3206106870229008	63
soccer	0.2909090909090909	0.25	0.2689075630252101	64
football	0.8533333333333334	0.9142857142857143	0.8827586206896552	70
%22forest%20fire%22	0.7391304347826086	0.85	0.7906976744186046	80
flood	1.0	0.8701298701298701	0.9305555555555556	77
earthquake	0.8805970149253731	0.9365079365079365	0.9076923076923077	63
drought	0.8909090909090909	0.8166666666666667	0.8521739130434782	60

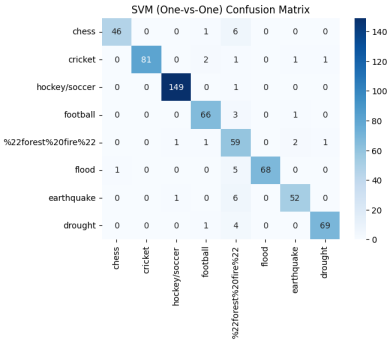
From the confusion matrices above it can be seen that soccer and hockey categories in particular can be merged together to improve performance. When doing so, the following results were received for the Naive Bayes classifier:



Naive Bayes Classification Report:

Class	precision	recall	f1-score	support
chess	0.925	0.6981132075471698	0.7956989247311829	53
cricket	1.0	0.7790697674418605	0.8758169934640523	86
hockey/soccer	0.6944444444444444	1.0	0.8196721311147541	150
football	0.9636363636363636	0.7571428571428571	0.848	70
%22forest%20fire%22	0.7301587301587301	0.71875	0.7244094488188977	64
flood	0.8888888888888888	0.8648648648648649	0.8767123287671232	74
earthquake	0.9782608695652174	0.7627118644067796	0.8571428571428572	59
drought	0.971830985915493	0.9324324324324325	0.9517241379310345	74

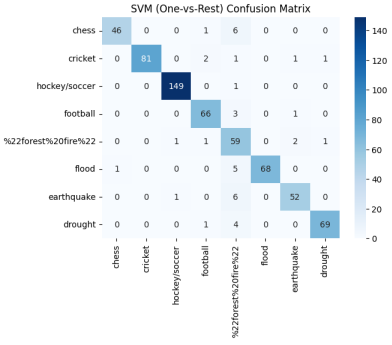
The following results were received for the SVM (One-vs-One) classifier:



SVM (One-vs-One) Classification Report:

Class	precision	recall	f1-score	support
chess	0.9787234042553191	0.8679245283018868	0.9199999999999999	53
cricket	1.0	0.9418604651162791	0.9700598802395211	86
hockey/soccer	0.9867549668874173	0.9933333333333333	0.9900332225913622	150
football	0.9295774647887324	0.9428571428571428	0.9361702127659575	70
%22forest%20fire%22	0.6941176470588235	0.921875	0.7919463087248322	64
flood	1.0	0.9189189189189189	0.9577464788732395	74
earthquake	0.9285714285714286	0.8813559322033898	0.9043478260869564	59
drought	0.971830985915493	0.9324324324324325	0.9517241379310345	74

Finally, the following results were received for the SVM (One-vs-Rest) classifer:



SVM (One-vs-Rest) Classification Report:

Class	precision	recall	f1-score	support
chess	0.9787234042553191	0.8679245283018868	0.9199999999999999	53
cricket	1.0	0.9418604651162791	0.9700598802395211	86
hockey/soccer	0.9867549668874173	0.9933333333333333	0.9900332225913622	150
football	0.9295774647887324	0.9428571428571428	0.9361702127659575	70
%22forest%20fire%22	0.6941176470588235	0.921875	0.7919463087248322	64
flood	1.0	0.9189189189189189	0.9577464788732395	74
earthquake	0.9285714285714286	0.8813559322033898	0.9043478260869564	59
drought	0.971830985915493	0.9324324324324325	0.9517241379310345	74

Class imbalance can impact the performance of the classification, especially when some classes are merged. This is because the model may not be able to distinguish well between the classes with fewer samples and the merged class, leading to lower accuracy and higher confusion between these classes.

To resolve the class imbalance, we can use techniques such as oversampling or undersampling. Oversampling involves generating more samples for the minority class, while undersampling involves reducing the number of samples in the majority class. There are also hybrid approaches that combine oversampling and undersampling, such as SMOTE (Synthetic Minority Over-sampling Technique).

Once we have resolved the class imbalance, we can recompute the accuracy and plot the confusion matrix for the One-vs-One and One-vs-Rest SVM classifiers.

Word Embedding

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

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

According to the paper, the ratios of co-occurrence probabilities should be the starting point for word vector learning, not the probabilities themselves. This is due to the fact that the ratio P_{ik}/P_{jk} is dependent on the three words i, j, and k, and it shows more information of these word's relationships than the individual probabilities. Based on the introduction of the most general model $F(w_i, w_j, w_k) = P_{ik}/P_{jk}$, w are word vectors and $\tilde{w}$ are separate context word vectors. After applying some conditions, We can select a unique value for F that covers the information present in the ratio P_{ik}/P_{jk} in the word vector space.

(b) 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??

The GLoVE embeddings will return different vector for the word running in these two sentences. This is due to the meaning of a word is significantly impacted by the context. Since the "Running" has different meanings, the GLoVE embeddings will obtain two distinct co-occurrence probabilities for two alternative vector representations of running in these two sentences. Therefore, these vectors of running in the two sentences should be different.

(c) What do you expect for the values of, $||GLoVE["queen"]-GLoVE["king"]-GLoVE["wife"]+GLoVE["husband"]||_2$, $||GLoVE["queen"]-GLoVE["king"]||_2 and ||GLoVE["wife"]-GLoVE["husband"]||_2$? Compare these values.

Form the paper, we can get that "king is to queen as man is to woman" should be encoded in the vector space by the vector equation king − queen = man − woman. Therefore, we can expect the values of $||GLoVE["queen"] -$

$GLoVE["king"]||_2$ equal to $||GLoVE["wife"] - GLoVE["husband"]||_2$. And, $||GLoVE["queen"] - GLoVE["king"] - GLoVE["wife"] + GLoVE["husband"]||_2$ would be approximate to 0.

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

In most cases, both stem and lemmatize have tradeoffs in terms of precision and complexity which are depending on the language and domain. Lemmatization involves mapping words to their dictionary form based on the part of speech, and stemming involves removing suffixes from words to obtain their root form. Since GLoVE embeddings are trained based on the ratio of co-occurrence probabilities, I prefer to lemmatize a word before mapping it to its GLoVE embedding rather than its stem.

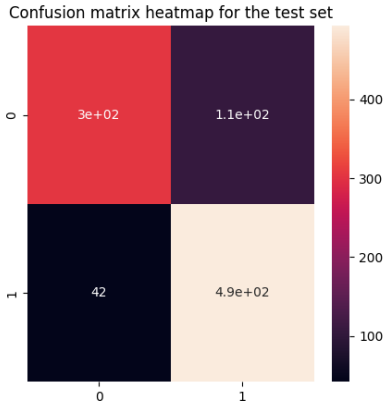
Question 11: For the binary classification task distinguishing the "sports" class and "climate" class:

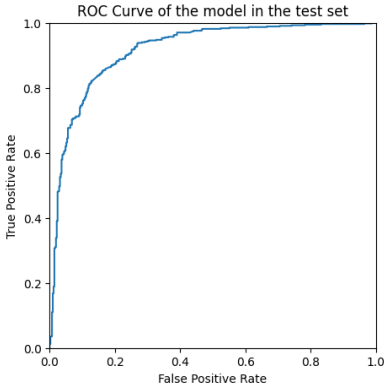
(a) Describe a feature engineering process that uses GLoVE word embeddings to represent each document.

We chose "full text" as the feature to accomplish the binary classification task for this question. For simplicity of classification, we convert the categories column of "root label" to the numeric column "label": 0 represents "climate" and 1 represents "sports". The total dataset was then divided for training and test. The HTML tags, emojis and punctuation were subsequently removed from the training and test data sets. We lemmatized the text as well. Glove (glove.6B.300d) was utilized to generate the word embeddings.

(b) 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.

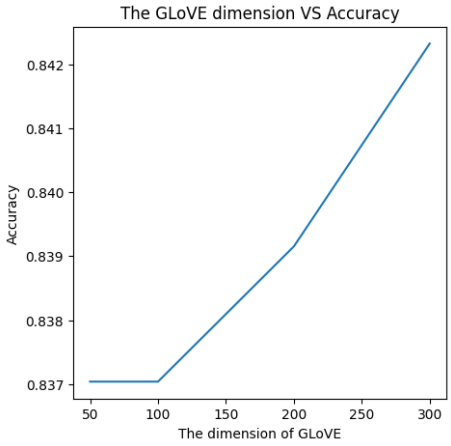
Using the sklearn library, we trained the Logistic regression model for the given query. The optimal hyperparameters for our experiment were determined to be l1 regularization and the inverse of regularization strength equal to 100 for training after multiple trials. Consequently, the accuracy of the model in the test set is approximately 84.23% and the AUC of the model in the test set is approximately 91.66%.





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? In this part use the different sets of GLoVE vectors from the link.

The following graph demonstrates that as GLoVE embedding dimensions increase, correspondingly increases the accuracy score. This could be because larger dimensions can represent greater amounts of information for embedding.



Question 13: Compare and contrast the two visualizations. Are there clusters formed in either or both of the plots? We will pursue the clustering aspect further in the next project.

In comparison to the normalized random vectors plot, the normalized GLoVE-based embeddings vectors plot reveals two quite distinct clusters.