Project 1_ Report
Haoting Ni (905545789), Yikai Wang (905522085), Yuanxuan Fang (005949389)

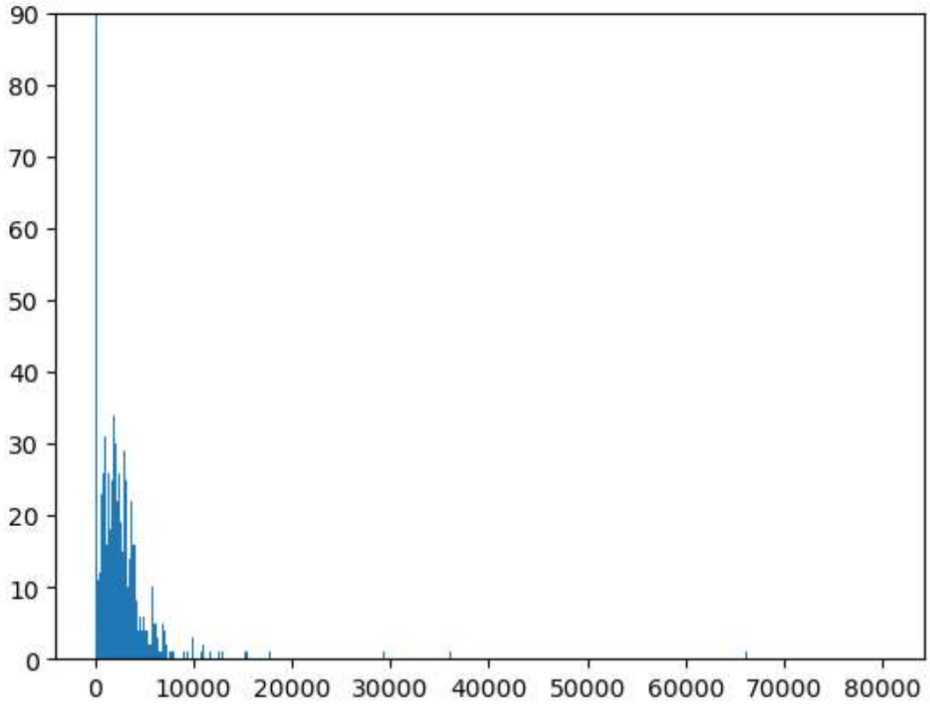
Question1:
How many rows (samples) and columns (features) are present in the dataset?

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   full_text     3150 non-null   object
 1   summary       3150 non-null   object
 2   keywords      3150 non-null   object
 3   publish_date  1863 non-null   object
 4   authors       3150 non-null   object
 5   url           3150 non-null   object
 6   leaf_label    3150 non-null   object
 7   root_label    3150 non-null   object
dtypes: object(8)
memory usage: 197.0+ KB
```

There are 8 columns (features) present in the dataset. The feature `publish_date` has 1863 rows samples, and other columns (features) have 3150 rows samples.

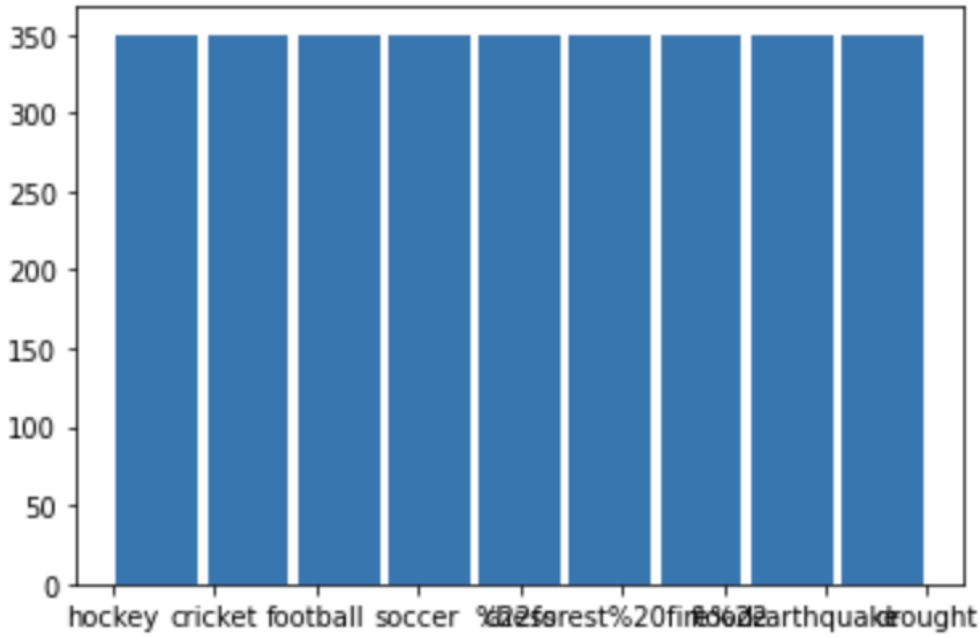
Plot and interpret 3 histograms on :
(a) The total number of alpha-numeric characters per data point (row) in the feature full text: i.e count on the x-axis and frequency on the y-axis



The plot shows the frequency of 2014 different numbers of alpha-numeric characters, and the maximum frequency of alpha-numeric characters is 99, and the number of alpha-numeric characters is 158.

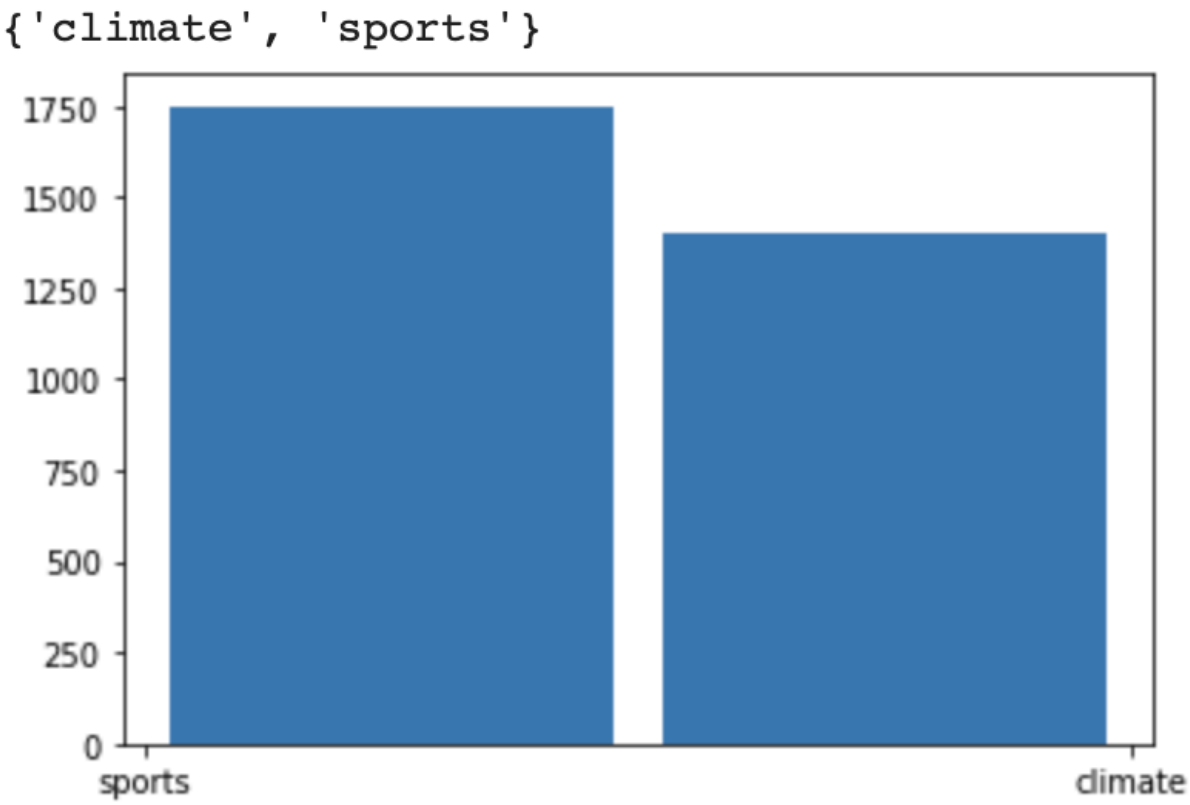
(b) The column leaf label – class on the x-axis

The plot shows the frequency of 9 different leaf labels, and each leaf labels' frequency is 350.



(c) The column root label – class on the x-axis

The plot shows the frequency of 2 root labels (sports and climate). The frequency of sports is 1750, and the frequency of climate is 1400.



Question 2:
Number of training samples : 2520
number of testing samples : 630

Question 3:

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

  Stemming works by cutting off the end or the beginning of the word. It cuts of the suffix or prefix. Lemmatization works by looking at the meaning of the word, and analysis the words, no matter of the changing of the word's suffix or prefix, and it will always give you the original format of the word.
  The pro of lemmatization is it will look through words no matter its noun, verb, adj,adv transformation. The con of it comparing to the stem is it takes longer in running time.
  It will reduce the size of dictionary size

- Min_df means minimum document frequency. How does varying min_df change the TF_IDF matrix

  Min_df is the minimum document frequency. If a word is less than it, then we will ignore the word. We will only keep the ones who are bigger than min_df, so it will shrink the TF-IDF matrix

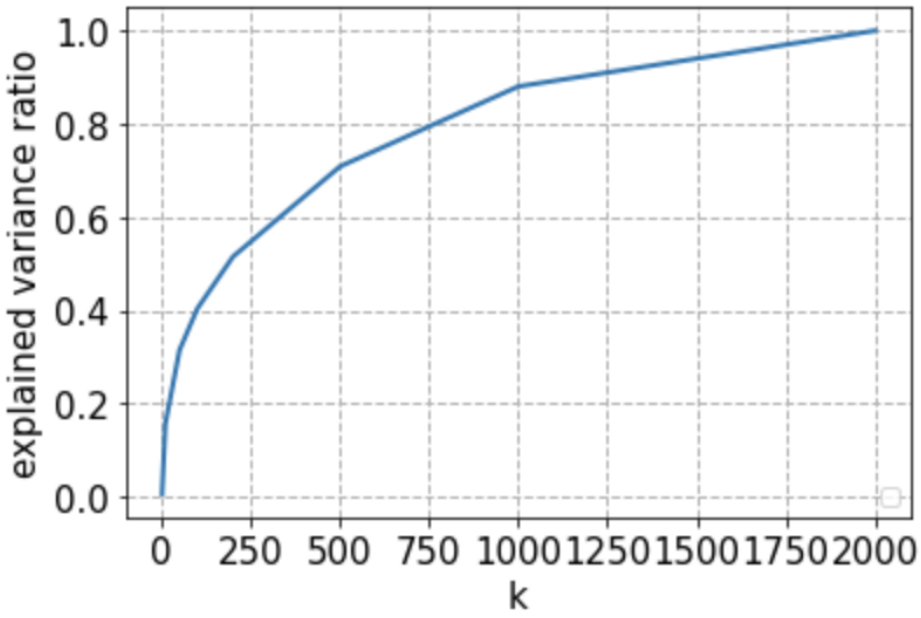
- Should I remove stopwords before or after lemmatizing? Should I remove punctuations before or after lemmatizing? Should I remove numbers before or after lemmatizing? Hint: Recall that the full sentence is input into the Lemmatizer and the lemmatizer is tagging the position of every word based on the sentence structure.

  We should remove stopwords and punctuation before the lemmatizing. Since stopwords might appear multiple times which is bigger than the min_df and might increase the size of matrix. Punctuation should be ignored before lemmatizing because it might fail the lemmatize since they are with the words and might confuse to lemmatize.
  We should remove numbers before the lemmatizing, because numbers are not important in the text to make it clean.

- 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.

  The training TF-IDF matrix dimension (2520, 13715)
  The testing TF-IDF matrix dimension (630, 13715)

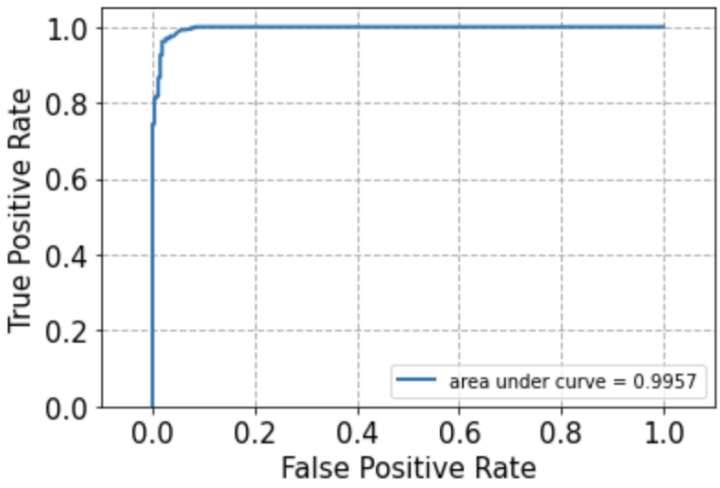
Question 4:



1. As the value of k increases, the explained variance ratio will increase. The explained variance ratio represents the percentage of variance related to different k. The concavity of plot shows that the percentage of variance's increase gets lower when k gets larger.

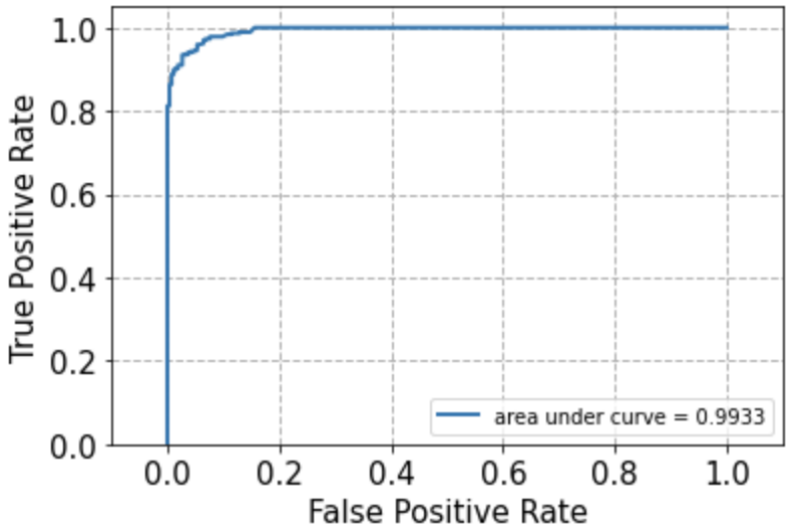
2. The reconstruction residual error of NMF is 41.205541397461126 which is slightly larger than LSI 40.856929314850056. Since LSI uses SVD, and it will give us the lowest error, but NMF will give a local minimum, but not the lowest.

Question 5:



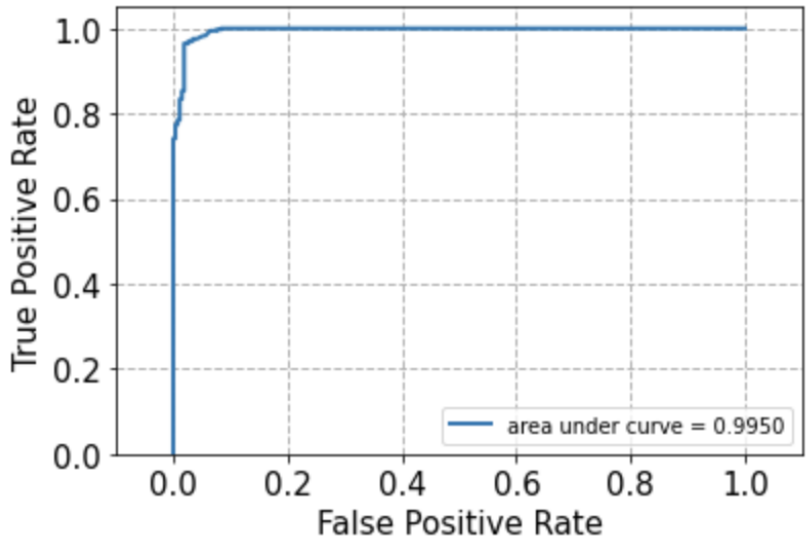
1. The ROC plot are shown as above.
   Hard margin confusion matrix is [[350, 17], [5,258]]
   Soft margin confusion matrix is [[367,0], [263,0]]

   Precision of Hard SVM is 0.9620486555697824
   Recall of Hard SVM is 0.9673335336351674
   F1-Score of Hard SVM is 0.9643184462820129
   Accuracy of Hard SVM is 0.9650793650793651



   Precision of Soft SVM is 0.2912698412698413
   Recall of Soft SVM is 0.5
   F1-Score of Soft SVM is 0.36810431293881646
   Accuracy of Soft SVM is 0.5825396825396826

By observing the data shown above, we conclude that hard SVM performs better than the soft one.



confusion matrix is [[349, 18], [5,258]]

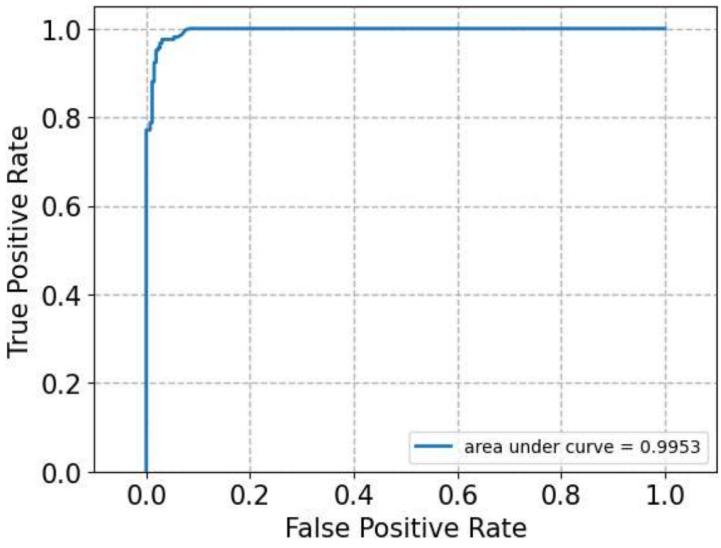
Precision of y=100000 is 0.9603291574551707
Recall of my=100000odel is 0.965971135815004
F1-Score of y=100000 is 0.9627141236017797
Accuracy of y=100000 is 0.9634920634920635

If we change y to 100000, it is the same as we get from the hard SVM. Confusion matrix also stays the same. Thus, the trade-off parameter does not change much if it is over a threshold.

The soft SVM does not perform really well, since the trade-off parameter is so small, it might cause underfit. Since the soft SVM only focuses on maximum the margin between the classes.

The ROC curve does not reflect the performance of the soft SVM. Since we will focus on the area under the curve of the ROC, and soft SVM has a small area. That is the reason that soft SVM does not perform well.

2. By applying cross validation, we get the best parameter score is 100.



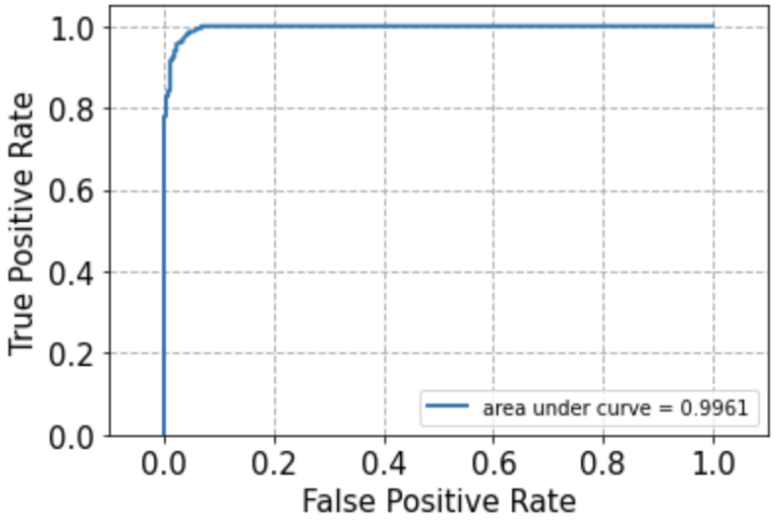
The ROC plot is shown as above.

Precision of y=100000 SVM is 0.9588220230473752
Recall of y=100000 SVM is 0.9640699951305933
F1-Score of y=100000 SVM is 0.961074668671287
Accuracy of y=100000 SVM is 0.9619047619047619

The confusion matrix of trade off at 100 is [[349, 18], [6,257]].

Question 6:

1. Results for logistic regression without regularization
   ROC Curve:



Confusion Matrix
[[351, 16],[6, 257]]

Precision of model is 0.9622926093514328
Recall of model is 0.9667947907709203
F1-Score of model is 0.9642842417745527
Accuracy of model is 0.9650793650793651

2. By using cross validation, we find the optimal regularization strength for logistic regression with L1 and L2 regularization is 10 and 100.

The result for L1 regularization:

```
Precision of model is 0.9605511358976462
Recall of model  is 0.9654323929507569
F1-Score of model is 0.9626787825401609
Accuracy of model is 0.9634920634920635
```

The result for L2 regularization:

```
Precision of model is 0.9590605472958413
Recall of model  is 0.9635312522663462
F1-Score of model is 0.9610373546631483
Accuracy of model is 0.9619047619047619
```

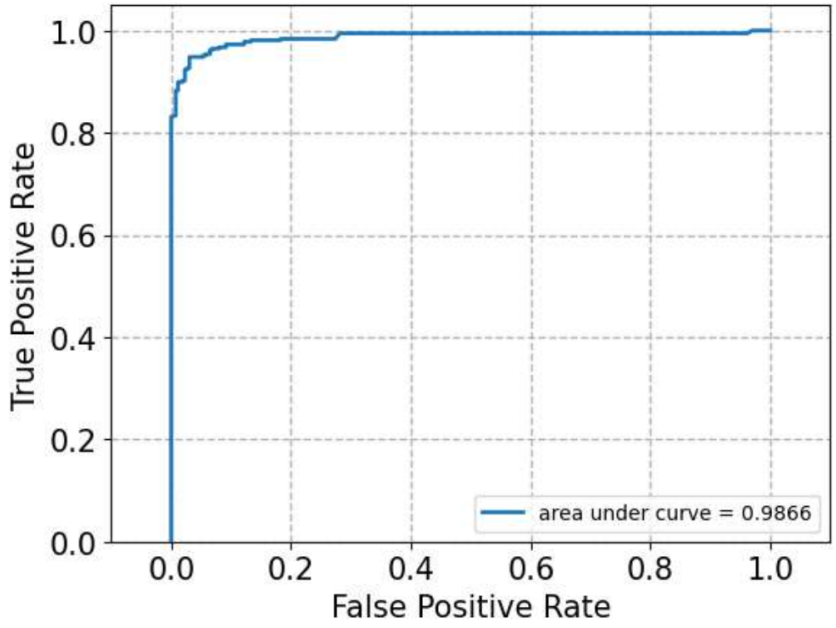
The logistic regression with L1 regularization has higher precision and accuracy compared to the logistic regression without regularization and with L2 regularization. Also, the performance values for the model with regularization and L2 regularization are similar.

The model without regularization can be used in some models that follow the training data closely. The purpose of L1 and L2 is to prevent overfitting in the model. L1 is to set a few coefficients to be 0, but L2 is to1 set the coefficients close to 0 (not equal to 0). Thus, we can use L1 to select only important features in the model, and the L2 can help the model to avoid overfitting when every feature in the model is significant.

The decision boundary for logistic regression based on the likelihood of maximum probability, and the decision boundary for Linear SVM based on the maximizing distance from the decision surface to the closest data point. Because logistic regression models usually have overfitting, Linear SVM has higher performance when dealing with the high dimensional data and outliers.

Question 7:

```
Precision of model is 0.9332059332059333
Recall of model  is 0.943887858600719
F1-Score of model is 0.9357142857142857
Accuracy of model is 0.9365079365079365
[[330  37]
 [  3 260]]
```



Question 8:

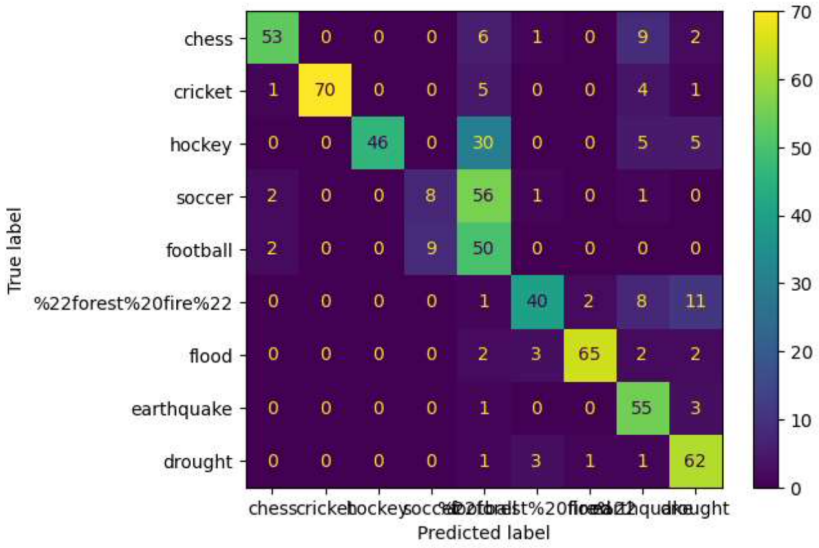
The best 5 combinations will be:

1. Logistic Regression (C=10, penalty = L1, solver = 'liblinear), with truncatedSVD(n_components =80), min_df = 5, with accuracy of 0.923810
2. Logistic Regression(C=10, solver = 'liblinear'), with truncatedSVD(n_components =80), min_df = 5, with accuracy of 0.923413

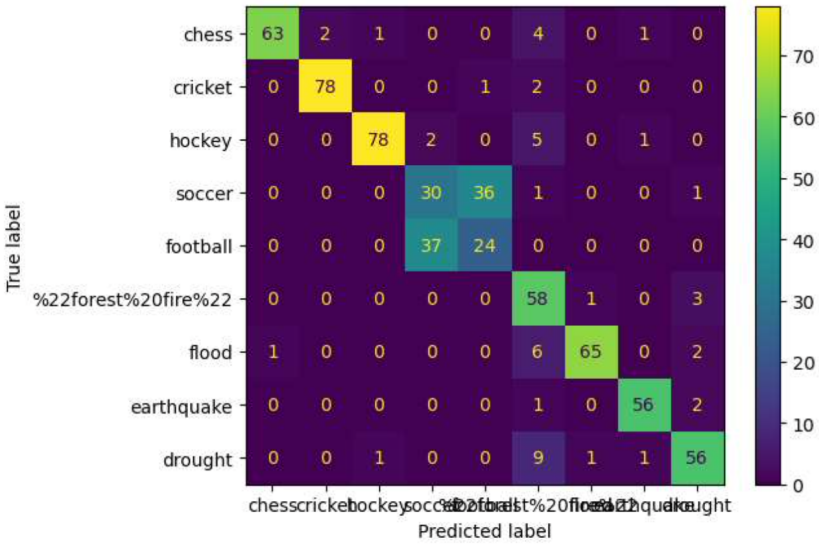
3. SVC(C=10, kernel = ‘linear’, random_state=42), with truncatedSVD(n_components =80), min_df = 5,with accuracy of 0923016
4. Logistic Regression (C=10, penalty = L1, solver = ‘liblinear), with NMF(n_components=80), min_df = 5, with accuracy of 0.920238
5. Logistic Regression(C=10, solver = ‘liblinear’), with truncatedSVD(n_components =80), min_df = 3, with accuracy of 0.919444

Question 9:

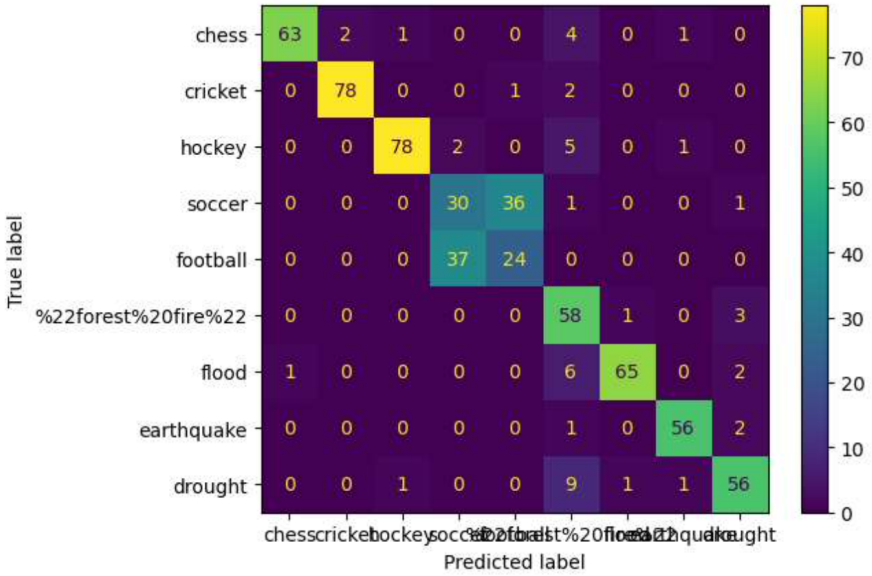
```
Precision of Na ïve Bayes classification is 0.7633926055028342
Recall of Na ïve Bayes classification  is 0.716709675490973
F1-Score of Na ïve Bayes classification is 0.701710631389194
Accuracy of Na ïve Bayes classification is 0.7126984126984127
```




```
Precision of OnevsOne is 0.8034800702668732
Recall of OnevsOne  is 0.7976029942037494
F1-Score of OnevsOne is 0.7974269207146141
Accuracy of OnevsOne is 0.8063492063492064
```



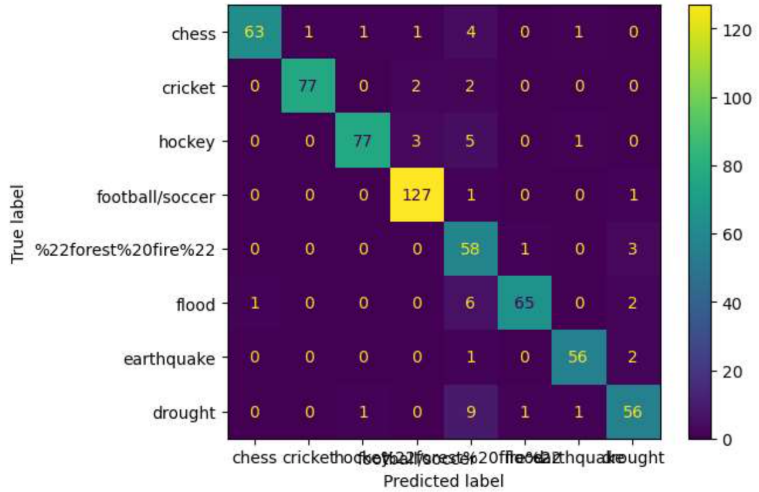
```
Precision of OnevsRest is 0.8034800702668732
Recall of OnevsRest  is 0.7976029942037494
F1-Score of OnevsRest is 0.7974269207146141
Accuracy of OnevsRest is 0.8063492063492064
```



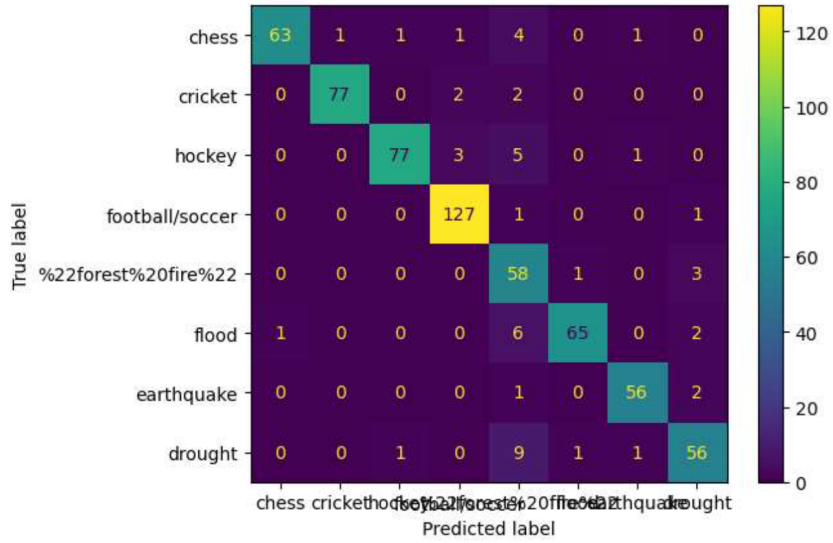
1. To resolve the class imbalance issue, in the OneVSOne svc() classifier, we add class_weight = “balanced”.

   From observation from the confusion matrix, we can see that diagonal has most the numbers, comparing to the other rows and columns. There are distinct visible blocks on the first column of these three different classifiers. This means that the label has a higher error rate in the prediction model.

```
Precision of OnevsOne_Merged is 0.9212307062854731
Recall of OnevsOne_Merged  is 0.9130412990420758
F1-Score of OnevsOne_Merged is 0.9135115956106845
Accuracy of OnevsOne_Merged is 0.919047619047619
```

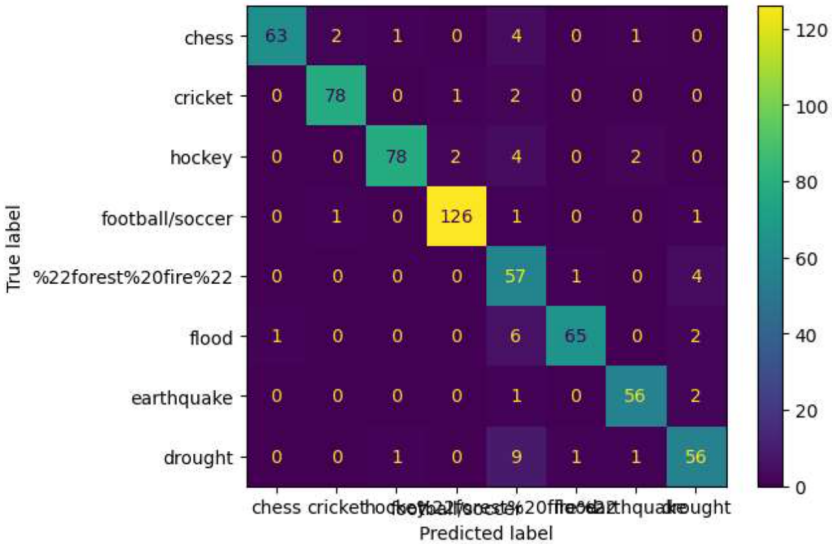


```
Precision of OnevsRest_Merged is 0.9212307062854731
Recall of OnevsRest_Merged  is 0.9130412990420758
F1-Score of OnevsRest_Merged is 0.9135115956106845
Accuracy of OnevsRest_Merged is 0.919047619047619
```

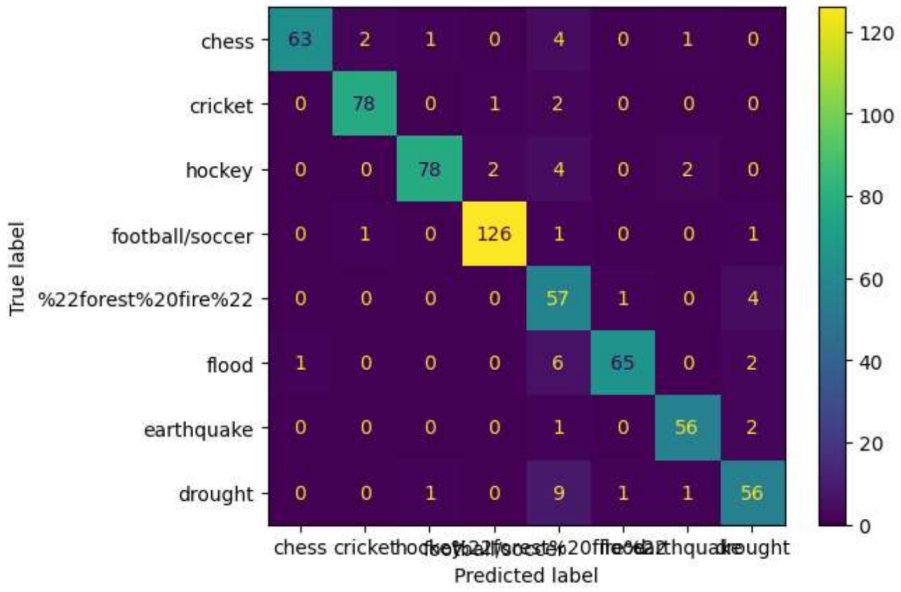


2. As the plots showed, we merge football and soccer as a one block. The accuracy gets higher than what if we did not merge those.

```
Precision of OnevsOne_Merged_balanced is 0.9178343282730052
Recall of OnevsOne_Merged_balanced  is 0.913052876010392
F1-Score of OnevsOne_Merged_balanced is 0.9123611232055715
Accuracy of OnevsOne_Merged_balanced is 0.919047619047619
```



```
Precision of OnevsRest_Merged_balanced is 0.9178343282730052
Recall of OnevsRest_Merged_balanced  is 0.913052876010392
F1-Score of OnevsRest_Merged_balanced is 0.9123611232055715
Accuracy of OnevsRest_Merged_balanced is 0.919047619047619
```



3. The imbalance class does impact the performance, but only a few. The plots are shown as above.

Question 10:

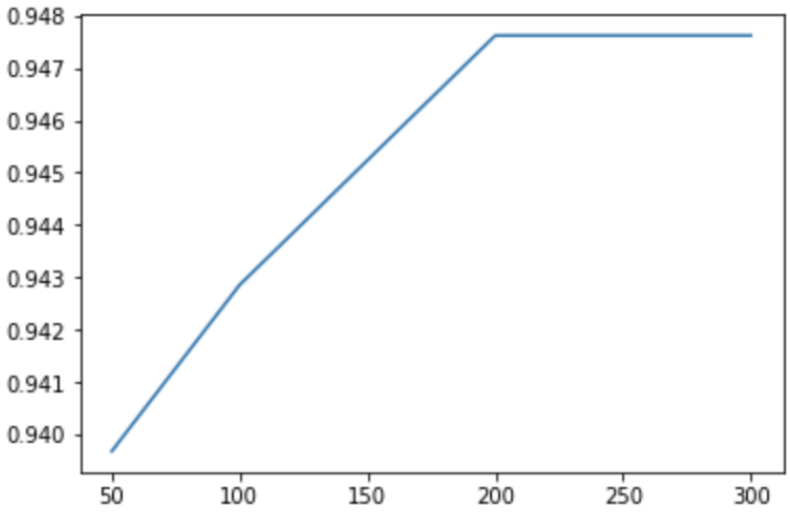
1. From the glove.pdf, it concludes that comparing with the probabilities themselves, the ratio of them will be better able to distinguish relevant words, and it will be better able to discriminate between the two relevant words. For instance, while k is the word “solid”, p(k|ice) = 1.9*10^-4, it is relatively small, p(k|steam) = 2.2*10^-5, the ratio of them is 8.9. Now, we can see that the word k, “solid”, is relevant of the word ice, rather than steam, since 8.9 is way bigger than 1. If the ratio is close to 1, that means it relevant to neither of them. However, the probabilities itself will not tell us which word is relevant.
2. It will give us the same vector since the other words in running sentence occur both 1 time in the sentences. Thus, it will give us the same embedded vector.
3. The glove value of thoes three equations are 6.1650367, 5.966258, 3.1520464. This means that king and queen, wife and husband’s relevance score is not quite the same. Since if we look at the meaning of thoes 2 pairs of words, there is only one king and queen in a territory, but there might be a lot of couples. Thus, wife-husband is more relevant than the king-queen.

4. We should lemmatize because we need to look deep into words meaning, rather than the structure similarity of the words. Stem will simplify cut the words off as the same part. Thus, we would lemmatize.

Question 11:

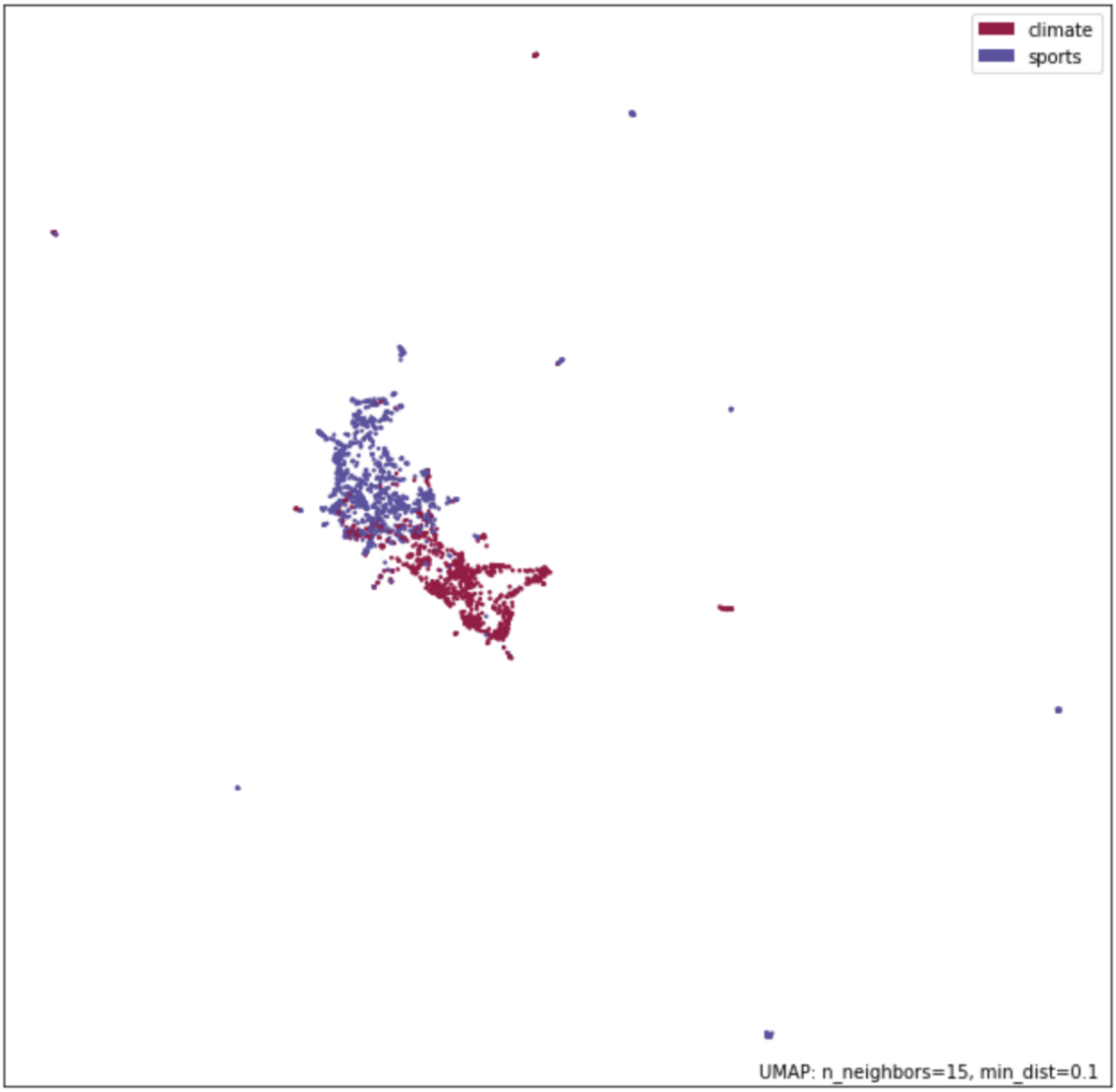
1. To satisfy the condition, we use the full_text. We added the embedded vectors up and then normalized it.
2. We select SVC as our classifier, train the models with Gloved based feature. And our accuracy score is 0.9476190476190476

Question 12:

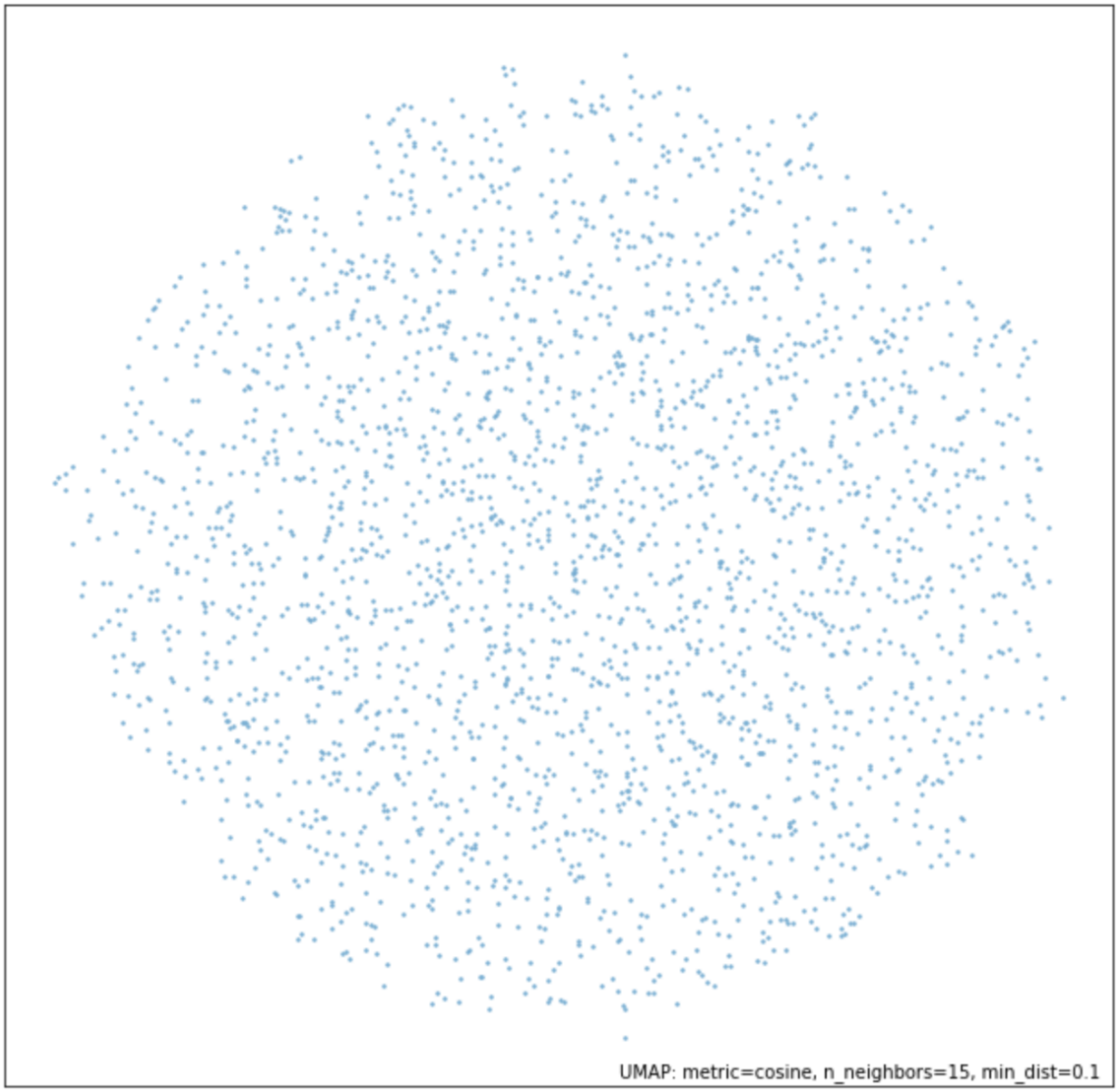


The plot we get on relationship between the dimension of the pre-trained Glove embedding resulting accuracy. We observe that when dimension increases, the accuracy also increases. It makes sense because more dimension means more information, which gives us higher accuracy of the prediction.

Question 13:



From this plot, which forms by gloved based embedding documents, it forms two clusters, climate and sports.



The second plot, it is formed by random normalized vectors, which forms no cluster.