Question1

Following is the screenshot of my results:
From the confusion matrix, we can find that:
Cluster 0 is the Travel & Transportation topic
Cluster 1 is the Disaster and Accident topic
Cluster 2 is the News and Economy topic

```
/usr/local/bin/python3.7 /Users/haodong/Desktop/660bia/hw6.py

Q1:

 actual_class  Disaster and Accident  News and Economy  Travel & Transportation
cluster
0                               71                5                      171
1                              130                6                        4
2                                9              195                        9

By using majority vote, we can find that:
cluster 0: Topic Travel & Transportation
cluster 1: Topic Disaster and Accident
cluster 2: Topic News and Economy

                        precision    recall  f1-score   support

  Disaster and Accident       0.93      0.62      0.74       210
      News and Economy        0.92      0.95      0.93       206
Travel & Transportation       0.69      0.93      0.79       184

             micro avg        0.83      0.83      0.83       600
             macro avg        0.85      0.83      0.82       600
          weighted avg        0.85      0.83      0.82       600


Process finished with exit code 0
```

Question2:

Screenshot2 of my results:

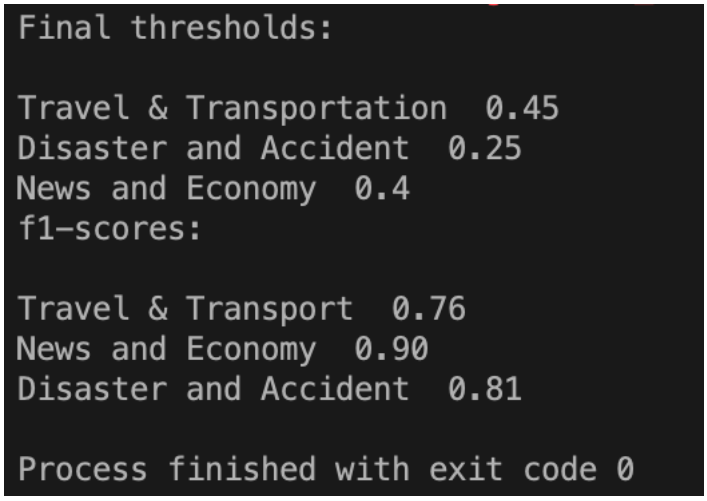
```
/usr/local/bin/python3.7 /Users/haodong/Desktop/660bia/hw6.py
iteration: 1 of max_iter: 25, perplexity: 1835.8948
iteration: 2 of max_iter: 25, perplexity: 1702.3477
iteration: 3 of max_iter: 25, perplexity: 1647.4778
iteration: 4 of max_iter: 25, perplexity: 1622.4970
iteration: 5 of max_iter: 25, perplexity: 1604.6551
iteration: 6 of max_iter: 25, perplexity: 1586.3610
iteration: 7 of max_iter: 25, perplexity: 1568.8486
iteration: 8 of max_iter: 25, perplexity: 1556.1749
iteration: 9 of max_iter: 25, perplexity: 1548.5009
iteration: 10 of max_iter: 25, perplexity: 1544.0310
iteration: 11 of max_iter: 25, perplexity: 1541.1949
iteration: 12 of max_iter: 25, perplexity: 1539.2035
iteration: 13 of max_iter: 25, perplexity: 1537.6124
iteration: 14 of max_iter: 25, perplexity: 1536.1824
iteration: 15 of max_iter: 25, perplexity: 1534.7516
iteration: 16 of max_iter: 25, perplexity: 1533.2306
iteration: 17 of max_iter: 25, perplexity: 1531.5472
iteration: 18 of max_iter: 25, perplexity: 1529.6234
iteration: 19 of max_iter: 25, perplexity: 1527.4633
iteration: 20 of max_iter: 25, perplexity: 1525.3911
iteration: 21 of max_iter: 25, perplexity: 1523.6398
iteration: 22 of max_iter: 25, perplexity: 1522.2325
iteration: 23 of max_iter: 25, perplexity: 1521.0654
iteration: 24 of max_iter: 25, perplexity: 1520.0581
iteration: 25 of max_iter: 25, perplexity: 1519.1079

 actual_class  Disaster and Accident  News and Economy  Travel & Transportation
cluster
0                               33               16                       89
1                              172                6                       79
2                                5              184                       16

By using majority vote, we can find that:
cluster 0: Topic Travel & Transportation
cluster 1: Topic Disaster and Accident
cluster 2: Topic News and Economy

                         precision    recall  f1-score   support

  Disaster and Accident       0.67      0.82      0.74       210
       News and Economy       0.90      0.89      0.90       206
Travel & Transportation       0.64      0.48      0.55       184

              micro avg       0.74      0.74      0.74       600
              macro avg       0.74      0.73      0.73       600
           weighted avg       0.74      0.74      0.73       600


Process finished with exit code 0
```

This is my first try by using 0.9 max_df and 50 min_df, we can find its performance is not well as Q1's result.

```
/usr/local/bin/python3.7 /Users/haodong/Desktop/660bia/hw6.py
iteration: 1 of max_iter: 25, perplexity: 1956.9207
iteration: 2 of max_iter: 25, perplexity: 1771.8707
iteration: 3 of max_iter: 25, perplexity: 1709.7353
iteration: 4 of max_iter: 25, perplexity: 1677.6619
iteration: 5 of max_iter: 25, perplexity: 1654.3099
iteration: 6 of max_iter: 25, perplexity: 1638.9804
iteration: 7 of max_iter: 25, perplexity: 1629.2856
iteration: 8 of max_iter: 25, perplexity: 1623.4374
iteration: 9 of max_iter: 25, perplexity: 1619.6597
iteration: 10 of max_iter: 25, perplexity: 1616.7683
iteration: 11 of max_iter: 25, perplexity: 1614.2464
iteration: 12 of max_iter: 25, perplexity: 1611.9878
iteration: 13 of max_iter: 25, perplexity: 1609.9164
iteration: 14 of max_iter: 25, perplexity: 1607.8638
iteration: 15 of max_iter: 25, perplexity: 1605.4789
iteration: 16 of max_iter: 25, perplexity: 1602.8164
iteration: 17 of max_iter: 25, perplexity: 1600.8061
iteration: 18 of max_iter: 25, perplexity: 1600.0685
iteration: 19 of max_iter: 25, perplexity: 1599.7290
iteration: 20 of max_iter: 25, perplexity: 1599.5184
iteration: 21 of max_iter: 25, perplexity: 1599.3708
iteration: 22 of max_iter: 25, perplexity: 1599.2593
iteration: 23 of max_iter: 25, perplexity: 1599.1710

 actual_class  Disaster and Accident  News and Economy  Travel & Transportation
cluster
0                               78                12                      167
1                              124                 6                        2
2                                8               188                       15

By using majority vote, we can find that:
cluster 0: Topic Travel & Transportation
cluster 1: Topic Disaster and Accident
cluster 2: Topic News and Economy

                        precision    recall  f1-score   support

  Disaster and Accident       0.94      0.59      0.73       210
       News and Economy       0.89      0.91      0.90       206
Travel & Transportation       0.65      0.91      0.76       184

              micro avg       0.80      0.80      0.80       600
              macro avg       0.83      0.80      0.79       600
           weighted avg       0.83      0.80      0.80       600


Process finished with exit code 0
```

This is my second try by tuning the model parameters, I changed the min_df to 45, and we can find the performance is much better than my first try. And by comparison with Q1 result, the performance is not better than Q1 result, but pretty close.

Question3:
Following is my result:



By tried different threshold for each label, I got the highest f1 score for “Travel & Transport” label when threshold = 0.45, the highest f1 score for “News and Economy ” label when threshold = 0.4 and the highest f1 score for “Disaster and Accident” label when threshold = 0.25.