## Part 2

### Question 15

First, we load the data from documents #superbowls, and then extract users from Washington or Massachusetts. We do this by reading each line in the file as JSON object and determine whether the location of the author of each tweet is from Washington, Massachusetts or neither via json\_object['tweet']['user']['location'].

First, we change the location to lowercase. For Washington, we define true list ["washington", "seattle", "kirkland", "wa"], and false list ["dc","d.c."]. Then If the location contains words in true list and does not contain words in false list, this user belongs to Washington. The purpose of false list is to avoid wrongly include author from Washington D.C in the class Washington. For Massachusetts, we define true list ["massachusetts", "ma", "boston", "northampton", "springfield", "plymouth", "arlington", "scituate", "worcester"], and false list ["ohio"]. Then If the location contains words in true list and does not contain words in false list, this user belongs to class Massachusetts.

Then generate TFIDF matrix. Also, we apply SVD and NMF as well as l2 regularization. Finally, we train five binary classifiers to predict the location of the author of a tweet (Washington or Massachusetts), given only the textual content of the tweet.

**Model 1: Linear SVM with TruncatedSVD**

The result is:

Accuracy: 0.7019994522048754

Recall: 0.6594626168224299

Precision: 0.6909424724602203

Confusion matrix: [[1434 505]

[ 583 1129]]

F1 score: 0.6748356246264197

The ROC curve is shown in Figure x.

A close up of a map

Description automatically generated

Figure x ROC Curve of Model 1

**Model 2: Naive Bayes BernoulliNB with SVD**

The result is:

Accuracy: 0.5708025198575732

Recall: 0.6133177570093458

Precision: 0.5370843989769821

Confusion matrix: [[1034 905]

[ 662 1050]]

F1 score: 0.5726752113444232

The ROC curve is shown in Figure x.

A close up of a map

Description automatically generated

Figure x ROC Curve of Model 2

**Model 3: Logistic regression with regularization term l2 SVD**

The result is:

Accuracy: 0.702273349767187

Recall: 0.6226635514018691

Precision: 0.7073656270736562

Confusion matrix: [[1498 441]

[ 646 1066]]

F1 score: 0.5726752113444232

The ROC curve is shown in Figure x.

A close up of a map

Description automatically generated

Figure x ROC Curve of Model 3

**Model 4: Random Forest**

The result is:

Accuracy: 0.6409202958093673

Recall: 0.5613317757009346

Precision: 0.6318211702827088

Confusion matrix: [[1379 560]

[ 751 961]]

F1 score: 0.5944942777605939

The ROC curve is shown in Figure x.

A close up of a map

Description automatically generated

Figure x ROC Curve of Model 4

**Model 5: Decision Tree**

The result is:

Accuracy: 0.6003834565872364

Recall: 0.657126168224299

Precision: 0.5633450175262894

Confusion matrix: [[1067 872]

[ 587 1125]]

F1 score: 0.6066325155028309

The ROC curve is shown in Figure x.

A close up of a map

Description automatically generated

Figure x ROC Curve of Model 5