CS5785 HW3

Andrew Palmer (ajp294), Kushal Singh (ks2377)

14 November 2019

1 Sentiment Analysis of Online Reviews

(a) Data Set Analysis

To process the data sets, we read in a single file at a time. Iterating over each line, we split the line based on the tab separator, which split the review sentence from its positive or negative label. For each item, we created a tuple of the form (label, review) and appended it to a list specific to that data set, e.g. amazon_negatives.

As described in the data set provided, there are equal labels from each website and each label type. Each website has 500 positive and 500 negative entries.

(b) Preprocessing Strategy

Our preprocessing strategy was iterative. First, we completed the sentiment prediction step without any preprocessing, which established a baseline. From there, we tested various preprocessing strategies and combinations as well.

In the end, we chose to remove stop words, lemmatize and stem each word, remove punctuation (except apostrophes), and lowercase words. Removing stop words aided our accuracy. Stop words are generally the most common words in a language[1]. As such, they do not show sentiment since they would be used in both positive and negative reviews. Removing stop words also reduced the number of features in both of our models.

Lemmatization and stemming also reduced our feature space and increased the accuracy of our models. As there are many words with the same root, the root is really what is driving the word sentiment. By lemmatizing words such as "good" and "well" to "good", we reduce our feature space without removing underlying data meaning.

Punctuation was also an important preprocessing strategy. Initially, we believed having punctuation, such as exclamation marks, would be a valuable form of

sentiment. However, such punctuation can exist in both positive and negative reviews, thus removing it gave better accuracy. We didn't remove the apostrophe as those marks are generally part of a word, which we did not want to alter.

Our final preprocessing step was to lowercase the sentences. Although there could be discriminating value to reviews written in all caps, we wanted to ensure words at the beginning of sentences and miscapitalizations throughout the sentence where treated correctly. Lowercasing actually came for free inside nltk.stem APIs, so we did not complete this step separately.

If we had more time, we would have liked to explore additional preprocessing. For example, many of the reviews had typos, or numbers intermixed inside words or sentences. Currently, our model creates features for each typo and number. However, it might be interesting either detect and remove these features or to derive some discriminating value from them. It may turn out that a review with many typos is indicative of negative sentiment; however, we were unable to test this hypothesis and did not include it in our strategy.

(c) Train/Test Split

We split the data as described in the homework. Each website had the first 400 reviews of each label used as training data and the remaining reviews as test data.

This gave us 2400 training examples and 600 testing examples.

(d) Bag of Words Model

For our Bag of Words model, we passed through the training set to build a dictionary of unique words. We cannot use the testing set here because that will overfit our model to the test data, which we need to have separated for testing purposes.

After creating the unique word dictionary, we iterated through both train and test sets to count up the occurrence of each word in our dictionary. The counts of each word, and the index of that word in the unique words list became our feature vector for the given review. Appending all of the feature vectors into another list gave us our feature matrix for the Bag of Words model.

Report 2 Feature Vectors

Please see attached file "feature_vectors.txt" for two examples. Each example is a processed sentence and its feature vector.

(e) Postprocessing Strategy

For our postprocessing strategy, we chose to log normalize the feature matrix. To do this, we took each feature value and transformed it according to $f(x) = log(x + 1)$. By taking the log of each feature, we are changing each feature to inform on relative, multiplicative, changes. In contrast, the linear scale informs on absolute, additive changes[2]. Therefore, we give more weight to features which may be absolutely less than common words, but relatively significant. Log normalization is a common approach in text mining.[3]
.
(f) Sentiment Prediction
Confusion Matrix labels:
0,0: True Negatives
0,1: False Positives
1,0: False Negatives
1,1: True Positives

Logistic Regression Results

$$\begin{bmatrix} 258 & 42 \\ 65 & 235 \end{bmatrix}$$

Classification Accuracy: 0.8216666666666667

Top 10 Word Weights:
Disciminative word at 4 is 'great'
Disciminative word at 38 is 'love'
Disciminative word at 864 is 'bad'
Disciminative word at 853 is 'poor'
Disciminative word at 2 is 'excel'
Disciminative word at 3006 is 'delici'
Disciminative word at 86 is 'nice'
Disciminative word at 666 is 'not'
Disciminative word at 989 is 'worst'
Disciminative word at 554 is 'amaz'

Naive Bayes Results
Multinomial Naive Bayes was used to generate these results.

$$\begin{bmatrix} 254 & 46 \\ 64 & 236 \end{bmatrix}$$

Classification Accuracy: 0.816666666666667

Top 10 Word Weights:
Disciminative word at 4 is 'great'
Disciminative word at 0 is 'good'

Disciminative word at 37 is "s'
Disciminative word at 1333 is 'film'
Disciminative word at 23 is 'phone'
Disciminative word at 1311 is 'movi'
Disciminative word at 31 is 'love'
Disciminative word at 25 is 'work'
Disciminative word at 20 is 'one'
Disciminative word at 62 is 'well'

(g) N-Gram Model
We implemented the N-Gram model as described in the homework. Additionally, we used the same pre and post processing strategy based on our earlier analysis and reasoning. Our sentiment prediction results are below:

Logistic Regression Results with N-Gram

$$\begin{bmatrix} 243 & 57 \\ 155 & 145 \end{bmatrix}$$

Classification Accuracy: 0.6466666666666666

Top 10 Phrase Weights:
Disciminative phrase at 22 is 'work great'
Disciminative phrase at 16 is 'high recommend'
Disciminative phrase at 1919 is 'wast time'
Disciminative phrase at 210 is 'one best'
Disciminative phrase at 424 is 'great phone'
Disciminative phrase at 279 is 'great product'
Disciminative phrase at 1649 is 'wast money'
Disciminative phrase at 9089 is 'food good'
Disciminative phrase at 4188 is 'realli good'
Disciminative phrase at 188 is 'easi use'

Naive Bayes Results with N-Gram
Multinomial Naive Bayes was used to generate these results.

$$\begin{bmatrix} 241 & 59 \\ 146 & 154 \end{bmatrix}$$

Classification Accuracy: 0.6583333333333333

Top 10 Phrase Weights:
Disciminative phrase at 22 is 'work great'
Disciminative phrase at 16 is 'high recommend'
Disciminative phrase at 210 is 'one best'

Disciminative phrase at 7 is 'sound qualiti'
Disciminative phrase at 279 is 'great product'
Disciminative phrase at 296 is 'could n't'
Disciminative phrase at 14 is 'good qualiti'
Disciminative phrase at 580 is '5 star'
Disciminative phrase at 49 is "s best'
Disciminative phrase at 52 is 'phone ve'

Comparison
As seen by the results, our Bag of Words model performed much better than the N-Gram model in both logistic regression, 0.82167 vs 0.6467, and naive bayes, 0.8167 vs 0.6583. Looking at the confusion matrices, Bag of Words and N-Gram performed similarly when it comes to True Negatives and False Positives. However, the N-Gram model had significantly more False Negative values, which hindered its accuracy performance. A discussion of the language used in strong reviews is in section (i) below.

(h) PCA for Bag of Words Model
We implemented PCA using the numpy.linalg.svd package. Using SVD, we can capture the principal components from the V output of numpy.linalg.svd. Depending on the number of principle components desired, we take more of the V data. Then, we multiply that by our centered matrix to project onto our new feature space. We performed PCA on both the train and test data and repeated sentiment analysis with this new feature matrix.

PCA Results BoW

	10	50	100
Logistic Regression	0.522	0.487	0.505
Naive Bayes	0.547	0.502	0.477

PCA Confusion Matrices BoW
Logistic Regression
10 Features:

$$\begin{bmatrix} 169 & 131 \\ 156 & 144 \end{bmatrix}$$

50 Features:

$$\begin{bmatrix} 148 & 151 \\ 156 & 144 \end{bmatrix}$$

100 Features:

$$\begin{bmatrix} 150 & 150 \\ 147 & 153 \end{bmatrix}$$

Naive Bayes:
10 Features:

$$\begin{bmatrix} 162 & 138 \\ 134 & 166 \end{bmatrix}$$

50 Features:

$$\begin{bmatrix} 142 & 158 \\ 141 & 159 \end{bmatrix}$$

100 Features:

$$\begin{bmatrix} 130 & 170 \\ 144 & 156 \end{bmatrix}$$

PCA Results N-Gram

	10	50	100
Logistic Regression	0.517	0.545	0.567
Naive Bayes	0.512	0.535	0.512

PCA Confusion Matrices N-Gram
Logistic Regression
10 Features:

$$\begin{bmatrix} 46 & 254 \\ 36 & 234 \end{bmatrix}$$

50 Features:

$$\begin{bmatrix} 243 & 57 \\ 216 & 84 \end{bmatrix}$$

100 Features:

$$\begin{bmatrix} 249 & 51 \\ 209 & 91 \end{bmatrix}$$

Naive Bayes:
10 Features:

$$\begin{bmatrix} 74 & 226 \\ 67 & 233 \end{bmatrix}$$

50 Features:

$$\begin{bmatrix} 190 & 110 \\ 169 & 131 \end{bmatrix}$$

100 Features:

$$\begin{bmatrix} 162 & 138 \\ 155 & 145 \end{bmatrix}$$

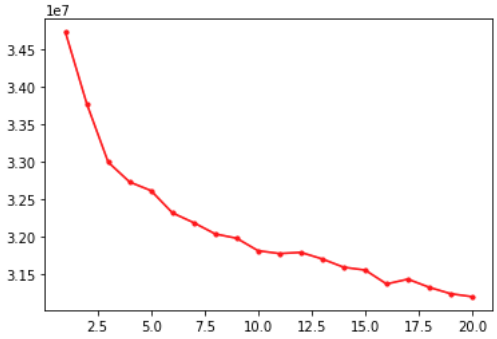
(i) Algorithms Comparison
Looking at the performance results of Bag of Words, 2-Gram, and PCA, we can see some distinct differences in the effectiveness of our different models. In terms of classifiers, Logistic Regression and Multinomial Naive Bayes produced very close results. In our 2-Gram model, Multinomial Naive Bayes actually produced slightly better accuracy than Logistic Regression. Multinomial Naive Bayes is a natural choice of classifier when it comes to analyzing multinomial data, such as word counts[4]. Given this fact, this classifier works well for data that can be turned into counts, which is exactly what our Bag of Words and N-Gram models are doing. Our Bag of Words and 2-Gram models' performance suffered when we performed PCA using numpy.linalg.svd. Given these results, we can see that our PCA is not effective for reducing the dimensionality and processing the review data. Keeping the feature dimensions gave us better results for this test/train set.

Comparing Bag of Words and 2-Gram models, we can see that Bag of Words produced considerably higher accuracy (0.82 instead of 0.64). Based on these results, we can determine that simply grouping the words into small phrases does not give enough of a discriminating sense of the review. There may need to be additional processing or phrase generation to produce higher results. For example, running experiments on different values of N in the N-Gram model may produce different outputs to make the N-Gram model more accurate. Additionally, the preprocessing methods that worked well for Bag of Words may not necessarily work well for 2-Gram. There are likely other parameters and characteristics that need to be considered when effectively preprocessing phrase data.

Examining the highest weighted features in our models, we can see some of the most discriminating language when it came to sentiment analysis. For example, in the Bag of Words model, single words such as great, love, bad, poor, and worst are effective at revealing sentiment in an online review. In our 2-Gram model, certain phrases, such as "work great", "high recommend", "wast time", and "veri disappoint" also give clear indication to the sentiment in an online review.

2 Clustering for Text Analysis

a) Running k-means for various values of k from 1 to 20, and calculating the inertia (i.e. sum of squared distances of samples to their closest cluster center) yielded the following plot.



We then chose $k = 5$, and classified the top 10 words per cluster and top 10 articles per cluster. The results are shown below.

```
Cluster 1's top 10 articles:
                                                 Titles
519         Algorithmic Gladiators Vie for Digital Glory
574      Reopening the Darkest Chapter in German Science
499      National Academy of Sciences Elects New Members
968   Suppression of Mutations in Mitochondrial DNA ...
777                  Divining Diet and Disease from DNA
1214      Turning up the Heat on Histoplasma capsulatum
90              Heretical Idea Faces Its Sternest Test
431               Movement Patterns in Spoken Language
1239                       An Arresting Start for MAPK
899   How to Get along: Friendly Microbes in a Hosti...


Cluster 1's top 10 words:
          Words
1         cells
7       protein
5          cell
23   expression
17         gene
0           fig
22     proteins
161   expressed
67         wild
56     specific
```

```
Cluster 2's top 10 articles:
                                                 Titles
574    Reopening the Darkest Chapter in German Science
499    National Academy of Sciences Elects New Members
519       Algorithmic Gladiators Vie for Digital Glory
90              Heretical Idea Faces Its Sternest Test
453      Information Technology Takes a Different Tack
0                         Archaeology in the Holy Land
777                Divining Diet and Disease from DNA
123  Corrections and Clarifications: Charon's First...
124  Corrections and Clarifications: Unearthing Mon...
122  Corrections and Clarifications: A Short Fe-Fe ...


Cluster 2's top 10 words:
              Words
0               fig
64          reports
20         observed
24            shown
403  correspondence
358       addressed
129    respectively
484           email
144         indicate
59          function

Cluster 3's top 10 articles:
                                                 Titles
983             Ubiquitination: More Than Two to Tango
902                          On the Ancestry of Barrels
302  Thermal, Catalytic, Regiospecific Functionaliz...
888  Structure of Yeast Poly(A) Polymerase Alone an...
519       Algorithmic Gladiators Vie for Digital Glory
574    Reopening the Darkest Chapter in German Science
834                  Harnessing the Power of Diatomics
499    National Academy of Sciences Elects New Members
797  Synthesis and Characterization of Helical Mult...
124  Corrections and Clarifications: Unearthing Mon...


Cluster 3's top 10 words:
          Words
230    residues
51      binding
451   conserved
208        side
396     crystal
38    structure
463        helix
409  structural
82       domain
270  resolution
```

```
Cluster 4's top 10 articles:
                                                 Titles
519        Algorithmic Gladiators Vie for Digital Glory
90               Heretical Idea Faces Its Sternest Test
574     Reopening the Darkest Chapter in German Science
499     National Academy of Sciences Elects New Members
124   Corrections and Clarifications: Unearthing Mon...
123   Corrections and Clarifications: Charon's First...
122   Corrections and Clarifications: A Short Fe-Fe ...
431                Movement Patterns in Spoken Language
302   Thermal, Catalytic, Regiospecific Functionaliz...
1281  The Formation of Chondrules at High Gas Pressu...


Cluster 4's top 10 words:
            Words
54         energy
95       electron
0             fig
154         shows
38      structure
39    temperature
277  measurements
242  experimental
35          human
312      constant
Cluster 5's top 10 articles:
                                                 Titles
519       Algorithmic Gladiators Vie for Digital Glory
574    Reopening the Darkest Chapter in German Science
90               Heretical Idea Faces Its Sternest Test
499    National Academy of Sciences Elects New Members
0                        Archaeology in the Holy Land
123  Corrections and Clarifications: Charon's First...
122  Corrections and Clarifications: A Short Fe-Fe ...
124  Corrections and Clarifications: Unearthing Mon...
777                Divining Diet and Disease from DNA
453      Information Technology Takes a Different Tack


Cluster 5's top 10 words:
         Words
7      protein
0          fig
96      values
5         cell
102      lower
619  estimates
99       range
101       mean
666   estimate
1        cells
```

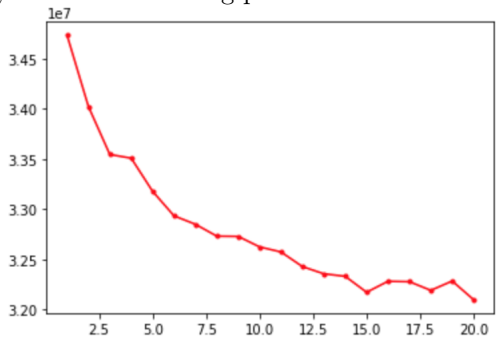
From the results above, it appears as though the articles closest to cluster 1 seem to be associated with cell expression and exploration. We can see this because some of Cluster 1's top words include: *cell, protein, gene, expression.* The articles closest to cluster 2 seem to be more associated with reports, bibliographies, and classifications of scientific articles. The articles closest to cluster 3 seem to be associated with cell analysis and binding. The articles closest to cluster 4 seem to be associated with temperature and graphical data, since some of its

top words include: *temperature, measurements, human, experimental, constant.* The articles closest to cluster 5 seem to be associated with a description of historical scientific articles.

From this information, we can obtain a rough idea of the content that each cluster describes, since we can match high-frequency words with the top articles, as these clusters describe similar content. Namely, we can see that cluster 1 describes articles relating to JSTOR happenings such as corrections and clarifications. Cluster 2 describes engineering-focused research and experimentation. Cluster 3 seems to describe biological experimentation content and their results. Cluster 4 appears to show a descriptive catalog of science articles and fields. Finally, cluster 5 describes geographic findings and effects of global warming.

b) For this part, we performed the same analysis as part a), using term-wise instead of document-wise grouping. This gave us the highest frequency of articles per term. Furthermore, we will also calculate the 10 closest words to each term, minimizing the average distance of fellow words to the mean of the term clusters. From this information, we should be able to glean the overarching subject topic of each term cluster, as similar terms will be clustered around similar topics.

Running k-means for various values of k from 1 to 20, and calculating the inertia (i.e. sum of squared distances of samples to their closest cluster center) yielded the following plot.



We then chose $k = 5$, and classified the top 10 words per cluster and top 10 articles per cluster. The results are shown below.

Cluster 1's top 10 articles:

	Titles
1242	Atom-Scale Research Gets Real
325	The Genome Sequence of Drosophila melanogaster
402	A Mouse Chronology
1303	Sedimentary Rocks of Early Mars
631	Status and Improvements of Coupled General Cir...
837	The Complete Atomic Structure of the Large Rib...
988	NEAR at Eros: Imaging and Spectral Results
436	Advances in the Physics of High-Temperature Su...
1345	Breakthrough of the Year: Genomics Comes of Age
327	Comparative Genomics of the Eukaryotes

Cluster 1's top 10 words:

	Words
4035	virulence
4357	bethesda
2336	boundaries
1613	describe
3089	kept
1332	mid
1949	status
359	five
3704	ubiquitination
4144	mediates

Cluster 2's top 10 articles:

	Titles
325	The Genome Sequence of Drosophila melanogaster
327	Comparative Genomics of the Eukaryotes
1187	Breaking down Scientific Barriers to the Study...
402	A Mouse Chronology
1242	Atom-Scale Research Gets Real
837	The Complete Atomic Structure of the Large Rib...
1345	Breakthrough of the Year: Genomics Comes of Age
409	Infectious History
439	Positional Syntenic Cloning and Functional Cha...
326	A Whole-Genome Assembly of Drosophila

Cluster 2's top 10 words:

	Words
3958	versions
5221	aromatic
5215	pumping
1065	contribute
3397	suppressed
2520	photosynthesis
3002	prefrontal
2504	kinetics
5189	placement
1475	fossil

```
Cluster 3's top 10 articles:
                                                  Titles
325      The Genome Sequence of Drosophila melanogaster
327              Comparative Genomics of the Eukaryotes
1242                      Atom-Scale Research Gets Real
402                                 A Mouse Chronology
1187  Breaking down Scientific Barriers to the Study...
1303                    Sedimentary Rocks of Early Mars
631   Status and Improvements of Coupled General Cir...
1345    Breakthrough of the Year: Genomics Comes of Age
837   The Complete Atomic Structure of the Large Rib...
436   Advances in the Physics of High-Temperature Su...


Cluster 3's top 10 words:
         Words
2108      peter
3589       ampa
4832        dex
5091  liposomes
3292   albicans
3425        ori
5090       ssra
2857       eyes
4124    healthy
3822  modulated

 Cluster 4's top 10 articles:
                                                  Titles
 325      The Genome Sequence of Drosophila melanogaster
 327              Comparative Genomics of the Eukaryotes
 837   The Complete Atomic Structure of the Large Rib...
 1187  Breaking down Scientific Barriers to the Study...
 1303                    Sedimentary Rocks of Early Mars
 717   Three-Dimensional Structure of the Tn5 Synapti...
 439   Positional Syntenic Cloning and Functional Cha...
 445   A Structural Framework for Deciphering the Lin...
 461   Architecture of RNA Polymerase II and Implicat...
 436   Advances in the Physics of High-Temperature Su...


 Cluster 4's top 10 words:
              Words
 3773          gated
 1264           rrna
 2134      glutamate
 1821        mineral
 3147        rescued
 1907  contributions
 5303        sucrose
 4748         frames
 3810    convergence
 1405        sulfate
```

```
Cluster 5's top 10 articles:
                                                  Titles
327              Comparative Genomics of the Eukaryotes
402                                   A Mouse Chronology
325      The Genome Sequence of Drosophila melanogaster
1242                       Atom-Scale Research Gets Real
1187  Breaking down Scientific Barriers to the Study...
1345    Breakthrough of the Year: Genomics Comes of Age
631   Status and Improvements of Coupled General Cir...
409                                   Infectious History
1303                    Sedimentary Rocks of Early Mars
326               A Whole-Genome Assembly of Drosophila


Cluster 5's top 10 words:
          Words
3808   enormous
3961       grew
916       yield
2519    beneath
2696    william
1852    degrees
4359  resembles
514       probe
5288    archaea
1599      media
```

Based on these results, we can once again identify the relationships between the articles and the terms. For example, cluster 2's top ten articles suggest content in the history of biology and genomics. We can see this based on the top words, such as *photosynthesis, kinetics, fossil* and the top articles, such as *Comparative Genomics of the Eukaryotes, Atom-Scale Research Gets Real.* Cluster 5's top ten articles suggest content in the areas of history. We can see this with its top words being *william, archaea, degrees.*

Both forms of clustering (doc-by-word and word-by-doc) demonstrate that k-means clustering allows for the general categorization of similar articles and words. These results help determine the general subject matter within a cluster. However, one of the issues with k-means is that because it starts off with random initial parameters, the final clusters will differ each time. Therefore, the analysis we perform (such as distance to mean) needs to be conducted on the same instance of k-means.

3 EM Algorithm and Implementation

a) GMM can be estimated via the EM algorithm in the following manner. Kmeans is just a special case of the EM Algorithm.

For the Expectation step, instead of calculating the weights, we can set the responsibility of the closest mean to 1 and the responsibility of the other means to 0.

For the Maximization Step, the maximized value would be the previously calculated mean, which is the average of the various points in that specific cluster, since there are no partial weights.

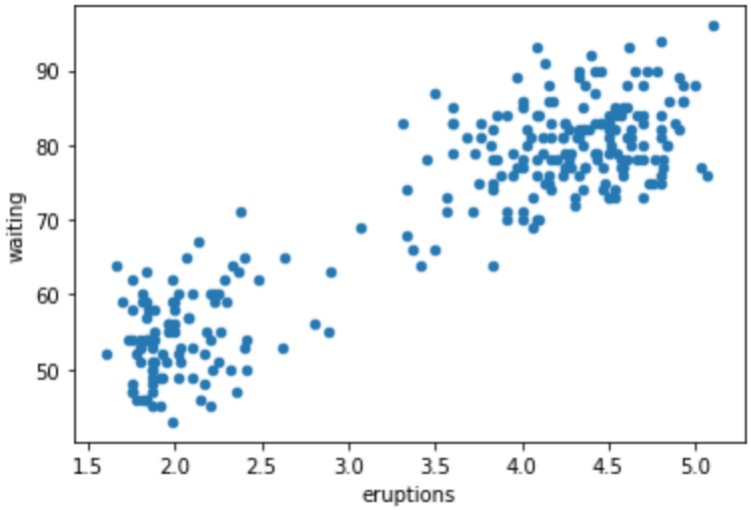
Mathematically, we can show that if we take σ to 0, π_k converges to 1 for the most probable class and 0 for the rest. Furthermore, γ_k converges to a step function between 0 and 1.

E-step: $\gamma_k = I(argmin_{1 \leq k \leq K} Norm(x_i - x_k)^2 = k)$

M-step: $\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} * y_i}{\sum_{i=1}^{N} \gamma_{ik}}$

b) For the data cleaning and preprocessing, we basically created a dataframe that has two columns: an eruptions column as type float, and a waiting column as type int. Plotting the values on a 2D plane we can see a general clustering, as shown below.

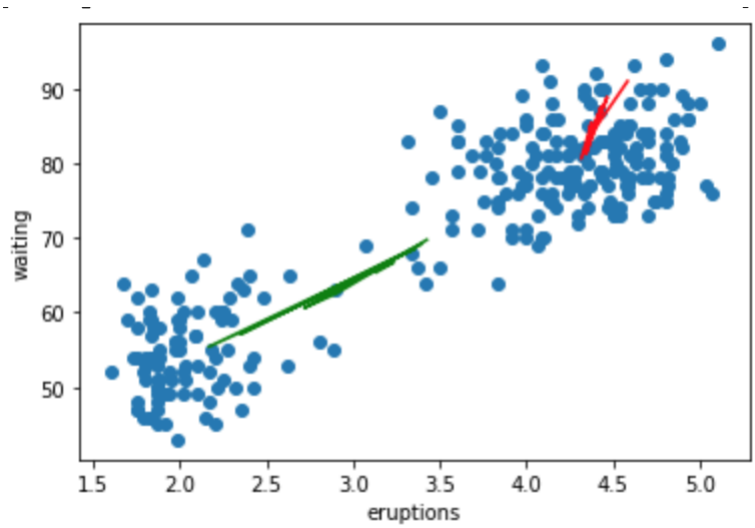
	eruptions	waiting
0	3.600	79
1	1.800	54
2	3.333	74
3	2.283	62
4	4.533	85
5	2.883	55
6	4.700	88
7	3.600	85
8	1.950	51
9	4.350	85



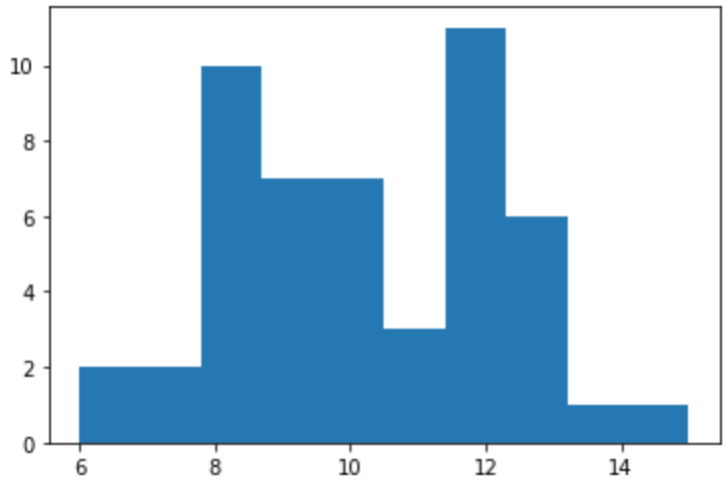
c) Termination Criteria: We set a very small threshold: $th = 1 * 10^{-5}$. When the difference of the old and new log-likelihood is smaller than the threshold, $L_{new} - L_{old} < th$, then we consider a convergence.

The graph below shows the results when we perform clustering using bi-modal

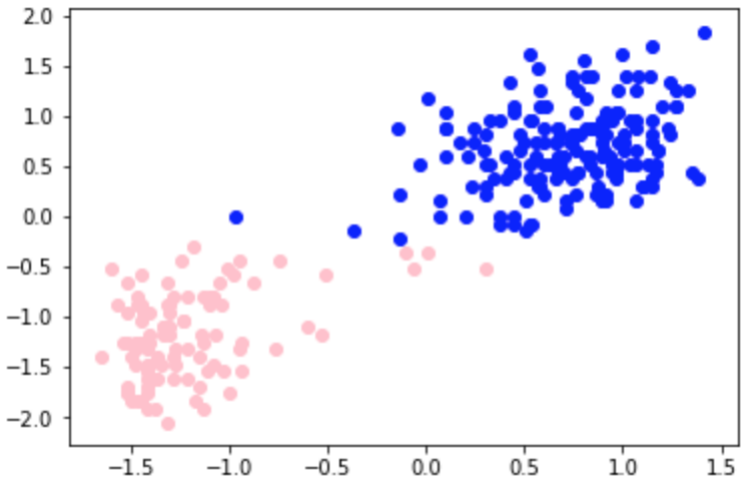
GMM.



The graph below shows the distribution of total number of iterations.



d) Repeating the task in (c) but with the initial guesses of the parameters generated from the aforementioned process yielded the following plot.



K-means and Gaussian Mixture Models (GMM) are two unsupervised learning methods for clustering unlabeled data. K-means finds k clusters where the average distance between individual points and cluster centers are minimized. Specifically, it performs hard cluster assignments, where a given data point is assigned to a specific clustering with ensured probability of 1 and 0 for the remaining clusters.

The other issue with K-means is that it is not always guaranteed to converge. Based on the initial starting parameters, K-means can lead to different cluster centers. K-means also assumes that the clusters have circular shapes.

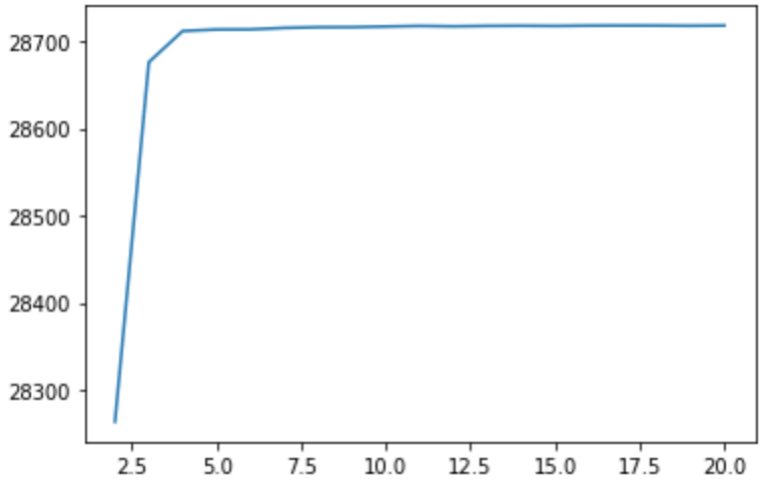
On the other hand, GMM assumes that the underlying data was sampled from a number of Gaussian distributions, each having a corresponding mean and variance. The goal is to then predict the means and variances of the Gaussians given the data, by maximizing the likelihood that a particular point belongs to a specific cluster. GMM uses the Expectation-Maximization algorithm to compute clustering responsibility and the parameters for the Gaussian distributions. GMM prediction performance can be increased by using random restarts.

4 Multidimensional Scaling for Genetic Population Differences

a)
i) The assumption(s) that are being made is that the $L2$ norm could approximate Nei's distances between the populations. This could fail if the euclidean distances have large errors on the approximation, because then MDS might not be able to reconstruct a good enough low-dimensional distance matrix. It could also fail if the distance matrix has evenly distributed variances on each direc-

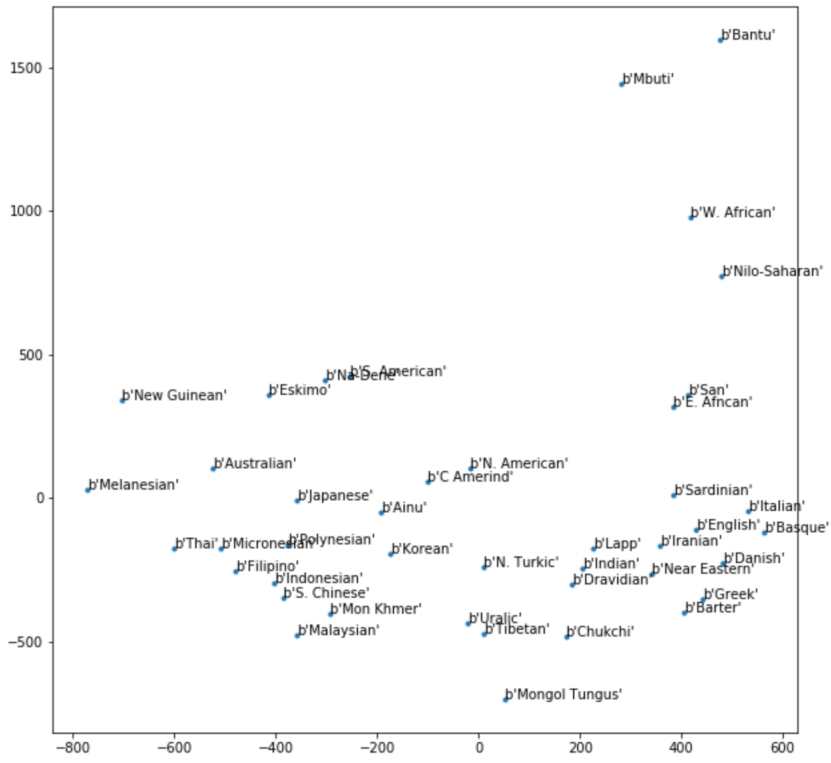
tion, because then MDS might cause huge loss of information when reducing dimensions. We can measure how much information is being lost by calculating pairwise euclidean distance between MDS output and original distance matrix.

ii) First, we performed MDS on a range of dimensions and computed the information loss by calculating the $L2$ norm between original $data['D']$ and the reconstructed low-dimensional matrix. The plot below shows this result for various values of k.

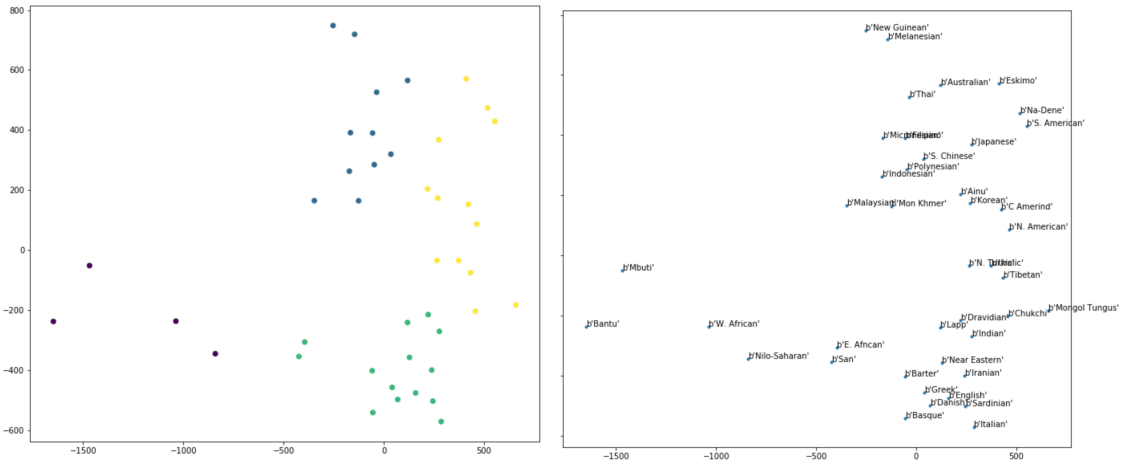


From this plot, we can see that roughly 4 dimensions are necessary to capture most of the data variation.

iii) Labeled 2D scatterplot.



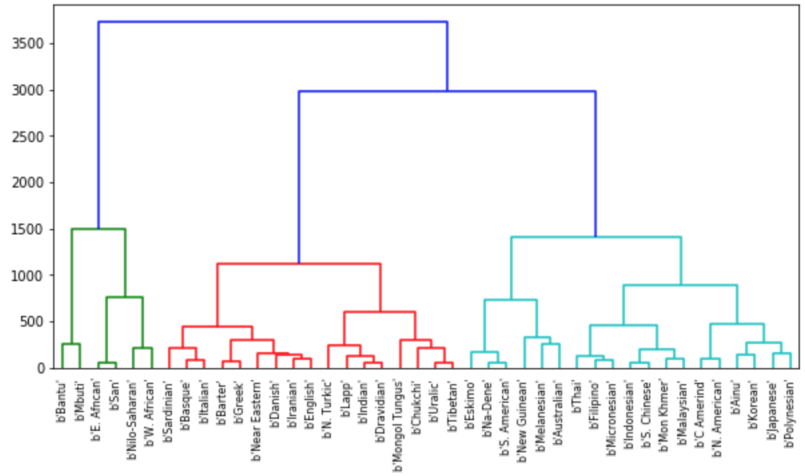
b) The plot below shows K-means on a 2-D embedding.



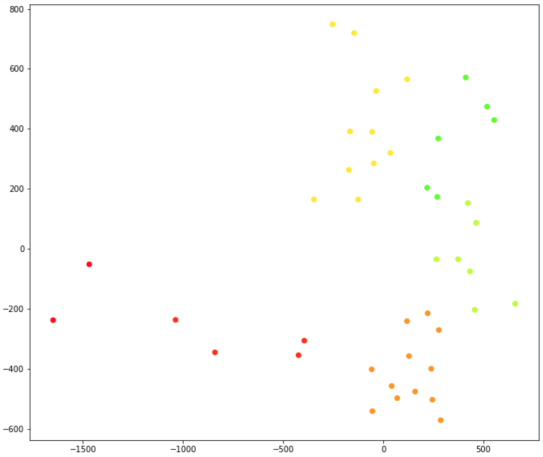
Overall, I do agree with the clustering results. The populations are generally clustered into the different continents Asia, America, Europe and Africa. But, there are still some populations that are clustered into the wrong group. For example, New Guinean is grouped into South East Asia, and Japanese is grouped

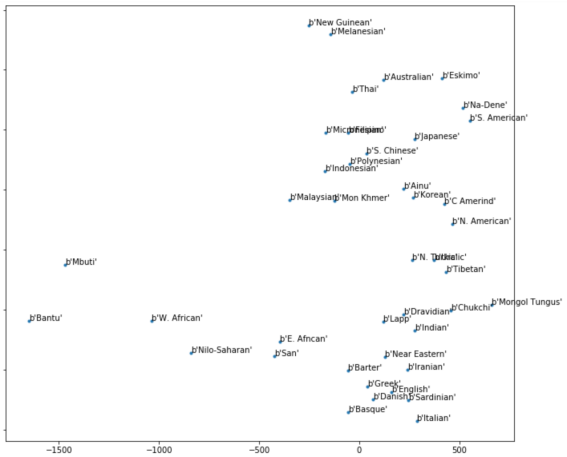
into America. As we increase k, the clustering result should get better, since the clusters would be more granular.

c) The plot below shows the dendrogram diagram.



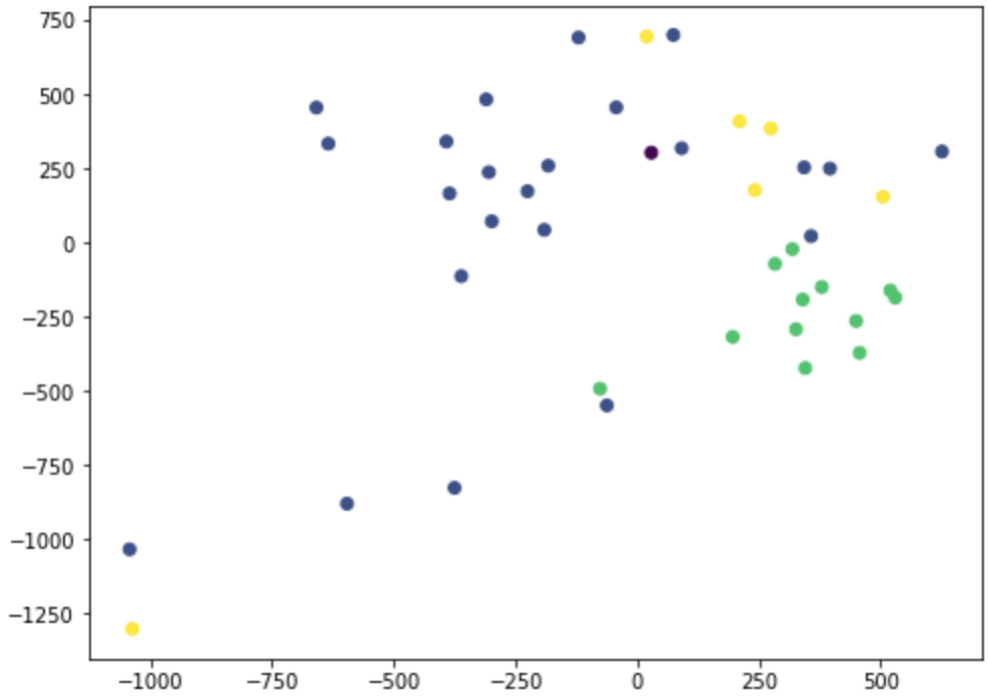
The plot below shows the 2D embedding after the cutoff has been applied.





From these results, we can see that fcluster by a cutoff distance is a bit better than k-means with $k = 4$. Some populations such as Korean, Japanese, and Ainu are clustered into a new group rather than clustered into an "American" group.

d) The plot below shows K-medoid clustering on the original distance matrix.



The key difference between K-medoids and K-means is that K-medoids only chooses existing data points as group centroids. This could potentially cause relatively large drifting of centroids. On the other hand, K-means uses logical centroids, which makes these centroids better representative of the cluster centers.

5 Written Exercises

1. **Decision Trees**
   (a) Suppose that each branch of this split is replaced by a leaf labeled with the more frequent class among the examples that reach that branch. Show that the number of training mistakes made by this truncated tree is exactly equal to the weighted impurity given above. Thus, using the min-error impurity is equivalent to growing the tree greedily to minimize training error.

Solution
Substituting our impurity function:

$(p1+n1) \times min(\frac{p1}{p1+n1}, 1-\frac{p1}{p1+n1}) + (p2+n2) \times min(\frac{p2}{p2+n2}, 1-\frac{p2}{p2+n2})$

$= min(p1, n1) + min(p2, n2)$

$= n1 + n2$ given that p1 and p2 are smaller than n1 and n2, respectively.

(b) Which split will be chosen at the root when the Gini index impurity function is used? Which split will be chosen at the root when min-error impurity is used? Explain your answers.

Solution
The Gini Impurity Index is given by $\sum_{i=1}^{J} P(i) \times (1 - P(i))$.

A1:
$2 \times (\frac{2}{4} \times \frac{2}{4} + \frac{1}{6} \times \frac{5}{6}) = \frac{7}{9}$

A2:
$2 \times (\frac{3}{4} \times \frac{1}{4} + \frac{2}{6} \times \frac{4}{6}) = \frac{59}{72}$

A3:
$2 \times (\frac{4}{7} \times \frac{3}{7} + \frac{0}{3} \times \frac{3}{3}) = \frac{24}{49}$

Given the results from the Gini Impurity Index, A3 has the minimal value and would be chosen at the root.

Min-Error Impurity:

A1:
$2 + 1 = 3$

A2:
$1 + 2 = 3$

A3:
$0 + 3 = 3$

Given the results from the min-error impurity, any split can be used.

(c) Under what general conditions on p1, n1, p2, and n2 will the weighted min-error impurity of the split be strictly smaller than the min-error impurity before making the split (i.e., of all the examples taken together)?

Solution
If $(p1 > n1) and (p2 < n2)$ then the error = $n1+p2$ If $(p1 < n1) and (p2 > n2)$ then the error = $n2 + p1$

From these, we can say that $p1 + p2$ and $n1 + n2$ are greater than $n2 + p1$ and $n1 + p2$

(d) What do your answers to the last two parts suggest about the suitability of min-error impurity for growing decision trees?

Solution
Looking at our results in (b), we see that the min-error impurity gave us equal results, meaning that any of the splits could be used. Because of this, and the randomness of choosing the split, a poor classifier would be created. Therefore, min-error impurity for growing a decision tree should not be used.

2. **Bootstrap Aggregation**
Suppose we have a training set of N examples, and we use bagging to create a bootstrap replicate by drawing N samples with replacement to form a new training set. Because each sample is drawn with replacement, some examples may be included in this bootstrap replicate multiple times, and some examples will be omitted from it entirely.

As a function of N, compute the expected fraction of the training set that does not appear at all in the bootstrap replicate. What is the limit of this expectation as N goes to ∞?

Solution
The probability of drawing with replacement, per sample, is $(\frac{n-1}{n})$. For N draws, the probability for all the draws comes to be $(\frac{n-1}{n})^n$.

Taking the limit of this expectation, as it tends to infinity:

$$\lim_{x \to \infty} (\frac{n-1}{n})^n = \frac{1}{e}$$

References

[1] https://en.wikipedia.org/wiki/Stop_words

[2] https://stats.stackexchange.com/questions/18844/when-and-why-should-you-take-the-log-of-a-distribution-of-numbers

[3] https://www.oreilly.com/library/view/feature-engineering-for/9781491953235/ch04.html

[4] https://towardsdatascience.com/multinomial-naive-bayes-classifier-for-text-analysis-python-8dd6825ece67