Aditya Chandel (A02169001)

Homework #1

CS 6676, Spring 2017

Task 1: Collecting Twitter Data

- For collecting the tweets, I used tweepy, which is wrapper library for Twitter's Search API.
- To download tweets using tweepy, one needs tokens from twitter for authentication to. To get the tokens, I had to create a dev account at twitter's official website. Next I entered the tokens into the tweepy's authentication method.
- To download tweets in English from USA after Nov 8, 2016, I put following parameters in the tweepy's search method:
    - query: 'Donald Trump'
    - lang: 'en'
    - geocode: '39.198205,-97.646484,2000mi'
    - since: '2016-11-08'
- I have used geocode to be '39.198205,-97.646484,2000mi' because the first two numbers in the string are coordinates for the geographic center of US and the 2000mi is the radius from the center. 2000-mile radius should be enough to cover the entire US, though there will be some tweets from Canada and Mexico as well which can be removed during data cleaning step.
- In one query, Twitter allows you to download 100 tweets, so I have used to loop to download multiple of 100 tweets.

- **Size of Dataset: 80k tweets (After pre-processing)**

- **Sample Tweets:** I have included some sample tweets along with this report in a txt file.

Task 2: Preprocessing the data

- The tweets downloaded using tweepy are stored in a text file, one tweet (and its metadata) per line.
- Each tweet and its metadata are stored in json format.
- To extract the properties from the tweets, I looped through all the tweets extracting the desired properties and storing them in a python list, and then writing the list to a csv file on each loop iteration.
- Only those tweets were selected which were in English and had geolocation or location information.
- Text of the tweets was cleaned to remove:
    - Urls
    - Non-ASCII characters
    - Stopwords
    - HashTags
- Location field had lot of inconsistent data like NYC, New York City, NY for New York. I cleaned that using Google Refine's clustering.
- While looping through the tweets, I used TextBlob library (internally uses NLTK and pattern.en) to calculate the sentiment of tweet's text and store it in the csv file along with the other tweet properties.

The sentiment value is between -1 and 1, -1 being extremely negative sentiment, 1 being extremely positive sentiment and 0 being neutral.

- Next I removed the duplicate tweets using sets.
- The output of the preprocessing step was stored in .csv file.

- Sample Tweet Sentiments:
  - BBC News - Donald Trump begins overhaul as first executive orders signed might b president but #trump has no class | **Sentiment: 0.25**
  - The Only Way to Stop Donald Trump Is to Join Him, Not Resist Him #tcot #trump" | **Sentiment: 0.0**
  - The more I listen to Donald Trump's Inauguration speech, the more I like it. He really is a strong speaker. #Trump #Inauguration | **Sentiment 0.477**
  - US President Donald Trump issues warning to Australia, world #Trump lock urself in z closet ur an incompetent moron | **Sentiment -0.575**
  - #new on #easy to do Donald Trump Lies About The Number Of People At His Inauguration #trump | **Sentiment 0.284**
  - It's not Donald Trump's fault, it's the fools who voted for him fault. #DumpTrump #trump #trumpsupporters #DonaldTrump | **Sentiment: 0.0**
  - With False Claims, Donald Trump Attacks Media on Crowd Turnout - The New York Times #trump #maga #trumplies | Sentiment: **-0.13**

Task 3 (a): Exploratory Analysis through k-means clustering

- I did K-Means clustering on the 'text' field of the tweets.
- The goal of the clustering was to find clusters/groups of words that occur in the tweets.
- To carry out K-means clustering I carried out following steps:
  - Remove the stop words from the tweet.
  - Vectorize the tweets using scikitlearn's 'TfidfVectorizer'.
  - Run K-Means algorithm on the vectorized tweets array.
  - Print out the output clusters containing the top clustered words.

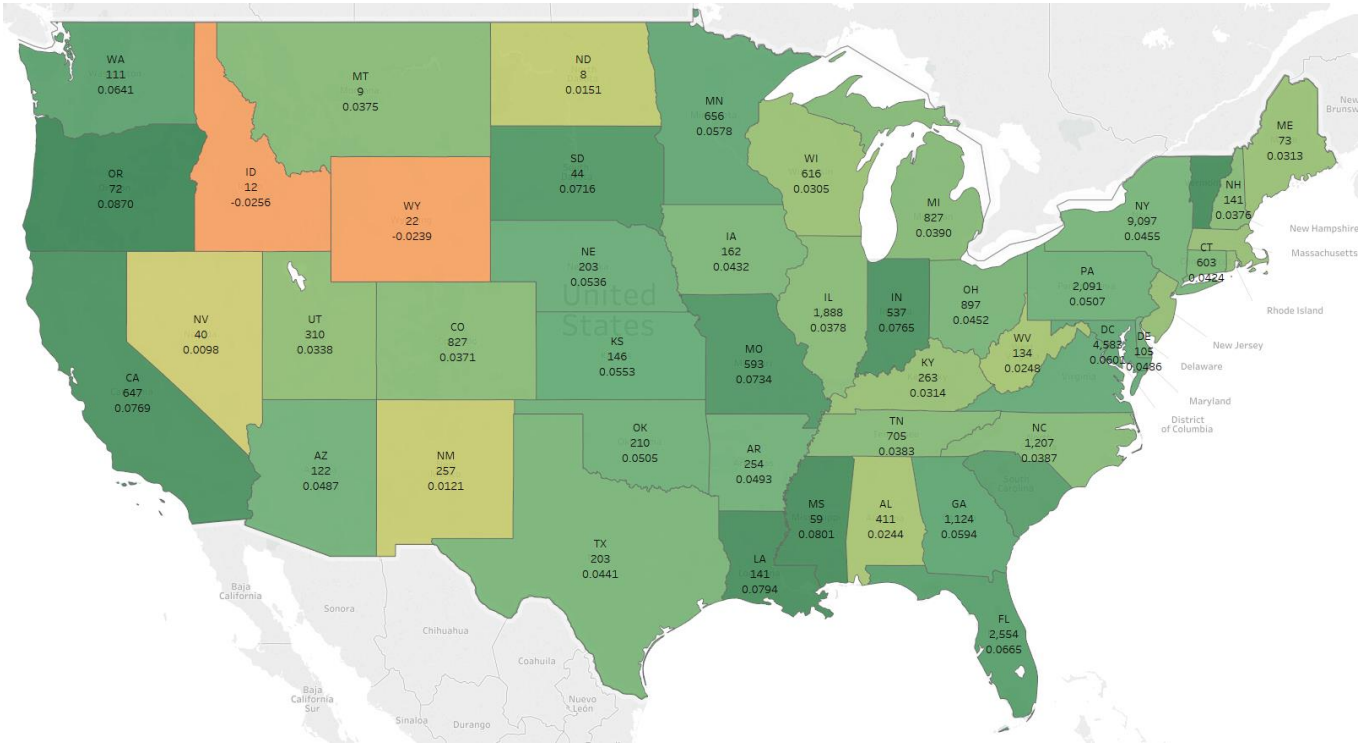
- Here are the clusters that I got for K=5:
  - Cluster 0: stay judge federal issued granted emergency order comedy grants strong
  - Cluster 1: trump amp president just like people don wall america day
  - Cluster 2: latest thanks daily news weekly times today amp features gazette
  - Cluster 3: pro life trump anti president venues women target immigrant phones
  - Cluster 4: payments foreign firms violate governments suit claim idea getting constitution

- **Why did I select K to be 5?** I researched online about the "How to choose a good K for K-means algorithms". I found out that there isn't any concrete way to select the K value. There are some methods like 'Elbow Method' that may help in selecting the K, but it is a heuristic method and hence it may or may not work. So I choose to visually inspect the clusters using various values of K. For K = 5, the words in each cluster were logically similar to the other words in that cluster. On increasing the value of K to 6,7,8 and so on, the logical similarity between the words started to decrease. Here are a couple of clusters for K=10, it can be noticed that the words in a cluster don't logically belong together:

- Cluster 2: just days competition access amp did people know make got
- Cluster 4: people don new think going know time says administration say
- Cluster 5: like looks just look don people sounds amp feel president

- **How each cluster is different?** Words in each cluster have similar logical meaning, and each cluster's words have very different logical meaning to the words in other clusters. For example:
  - In cluster 0: The words are "stay judge federal issued granted emergency order comedy grants strong", which indicate the cluster in about the words related to 'Stay given by the judge'.
  - In Cluster 2: The words are "latest thanks daily news weekly times today amp features gazette", indicating that the cluster is about 'News'.
  - In Cluster 4: The words are "pro life trump anti president venues women target immigrant phones", indicating that the cluster is about 'Women Rights'.

Task 3 (b): Distribution of Tweets based on Sentiment



- The map above shows the distribution of tweets at all the US states.
- In each state, the first value is the abbreviation of state name, middle value is the number of tweets from that state and the bottom value is the average sentiment of all the tweets for that state.
- The Yellower the state in the map, more negative is the sentiment towards Donald Trump. Vice versa for Green color.
- It can be observed that Idaho and Wyoming have the most negative sentiment and Oregon and Louisiana have most positive sentiment.
- Most of the states have positive sentiment towards trump.

Task 3 (C): Tweet Classification **(One more interesting analysis)**

- Here the goal was build classifiers using different classification algorithm to classify tweets based on the features selected from the twitter metadata.

- Selected Features:
  - Number of followers
  - Number of friends
  - Ratio of followers and friends
  - Hour of the day
  - Location
- Labels/Target was the sentiment value:
  - 0 for sentiment less than 0
  - 1 for sentiment equal to 0
  - 2 for sentiment greater than 0
- Classifiers Used:
  - Nearest Neighbors
  - Decision Tree
  - Random Forest
  - Neural Network
  - Ada Boost
  - Naïve Bayes
  - Quadratic Discriminant Analysis

- Steps to run classifier:
  - The tweets were randomly split into a 60/40 training and test sets.
  - The training set was used to train the classifiers and test set was used to calculate the accuracy.
  - This was carried 10 times and accuracy values were averaged for the 10 runs.
  - The data was run on all the above classifiers and the accuracy was noted down.

- Accuracy for various classifiers:
  - Nearest Neighbors: 0.403010360331
  - Decision Tree: 0.473186942073
  - Random Forest: 0.478269368606
  - Neural Net: 0.45989444191
  - AdaBoost: 0.475206880824
  - Naive Bayes: 0.468821268
  - QDA: 0.468886427315

- We can observe that the Random Forest has the highest accuracy of 47.8%, closely followed by AdaBoost and Decision Tree. Nearest neighbor has lowest accuracy of 40.3%.
- Accuracy of none of the algorithms is good and none will be able to classify the tweets correctly.
- Low accuracy is due to the lack of good features in the dataset. To improve the accuracy, we'll need more good features.