**Yelp Dataset Challenge**

**List of Steps**

1. Take raw data in json and transfer it to text files and extract needed attributes (score, review, businessID)
   1. Run the following line of code in the command line in order to obtain the review files with the attributes of business id, star rating, and text review. C:\Users\PC Owner\Documents\Untitled-1.jpg
   2. Open up the business dataset json file with a word editor, save it as a text file. Business dataset is small enough to be opened up and saved by a word editor.
2. Take data in text files and extract the following attributes from business: businessID, category, location - (FileProcessor )
   1. Open up the FileProcessor zip file
   2. Place business dataset in a text file, from step 1.b
   3. Run Business\_Classify.java. it will return a text file with the business id, the categories, and the state of the business.
3. Select the data of a single state, city and business type (FileProcessor )
   1. Run State and Categories. java in order to obtain the list of businesses, with their categories and states.
   2. Run State\_Count, find the state with the largest amount of businesses
   3. run Business\_Count, find the max category within the state found in step 3b
   4. Run Count\_Cities, find the city within the state stated in 3c with the highest amount of businesses
   5. Run Business\_List , give the city, state, and category to get the set of businesses that match said criteria
   6. Run Review\_Select, to obtain all the reviews that are linked to one of these businesses in a single text file.
4. All to lower case
   1. Run All\_To\_Lower\_Case in the review set from 3f, it will turn all text into lower case.
5. Separate the reviews into 5 different files (1-star, 2-star, 3 star, 4-star, 5-star)
   1. Run Separate\_To\_File in set of reviews (4a), and move them into separate files based on their score.
6. Remove stop words
   1. Run Stop\_Words\_Project on all five text review files(from 5a) to remove the set of all stop words.
7. Use tf-idf to eliminate unimportant words (Document Splitter, TF-IDF)
   1. Run DocumentSplitter on all five review files (DocumentSplitter found in folder called eduardo)
   2. Run TF-IDF in the set of documents given from DocumentSplitter (TF-IDF found in folder eduardo)
   3. Run Pick\_Words\_Above\_TFIDF\_Threshold (found in FileProcessor.zip), in the five outcomes from TF-IDF to obtain a single set of TF-IDF scores for all possible features
8. Run k-means
   1. Run Files\_By\_Keywords using the files from step 5a and step 7c to create a frequency matrix.
   2. Run Files\_By\_Keywords\_Binary using the files from step 5a and step 7c to create a binary matrix
   3. KMeansTest (found in K-Means Testing.zip) on the two files from 8a and 8b, outcome will be the rand index score, and average silhouette (have to remove comment characters to obtain average silhouette))
9. Run Naïve Bayes
   1. Run Taking\_out\_Training\_Data using the files from step 5a and 7c to create a frequency matrix, and a test frequency matrix of 500 reviews
   2. Run Taking\_out\_Training\_Data\_Binary using the files from step 5a and 7c to create a frequency matrix, and a test frequency matrix of 500 reviews
   3. Run NaiveBayes (found in folder eduardo), to obtain the results.

NOTE: further information about all code can be found in separate readme files found within each separate zip or folder.