Capstone 2 - Milestone 2 Report: Valery Lynn

Title: National patterns in race tweets

Problem: Racial bias is difficult to measure due to its sometimes ambiguous and subtle presentation. Most empirical measures of racial bias or discrimination are conducted using individual self-reporting or experimental testing that attempts to obtain results using indirect or subconscious actions. A research study being conducted at UCSF, Department of Epidemiology & Biostatistics, is using Twitter data to create place-level measurement of sentiment towards racial/ethnic minorities. The study analyzes more than one million U.S. tweets related to race/ethnicity for sentiment. A sample of the tweets (6,600) have been human-labeled with 90% agreement between the researchers. The researchers have made the data available to me to help them classify the tweets as positive, negative, or neutral. A previous attempt was made, but they were unable to classify the tweets in more than two categories. They got around this by collapsing the neutral category into the negative category. This project will be my attempt to use more up-to-date NLP methods to classify the tweets into the three categories originally proposed by the researchers.

Data: A stata file was obtained from Thu Nguyen, a post-doctoral fellow (Associate Specialist) at UCSF, Department of Epidemiology & Biostatistics. It contained the original tweets, along with results from a previous sentiment analysis, names of the people who hand coded the data (conducted after the unsupervised sentiment analysis), the labels, and the state, county, and zipcode for the location of each tweet. Each tweet has an identification number attached as well as a location FIPS code that was determined from the latitude and longitude locations from the tweets. The FIPS code will be used to create the U.S. heatmap by county after the final prediction. The training set has 6,481 rows, each with one tweet. The unlabeled set to be predicted is 1.25 million tweets, of which the training set was extracted for manual labeling last summer.

The tweets were obtained from March 2015– April 2016 using Twitter's Streaming Application Programming Interface (API) to continuously collect a random sample of publicly available tweets. This API gives access to a random 1% sample of tweets. The data was restricted to only tweets with latitude and longitude coordinates for the contiguous United States and the District of Columbia. In total, there are 79,848,992 million general topic tweets from 603,363 unique Twitter users in the dataset. A

keyword list of 398 racial/ethnic group terms and slurs was derived from racial and ethnic categories used by the U.S. Census and the online database of racial slurs. These keywords were used to identify tweets that contained at least one race-related word. This resulted in 1.25 million tweets.

Out of this, the training set was created by randomly selecting a subset of 6,481 tweets for manual labeling. Three coauthors labeled three possible responses: negative or positive. Inter-rater reliability among the coders was 90% for sentiment rating. In addition, the labelers indicated whether the tweet used discriminatory or stereotyping language about a racial group, whether the tweet mentioned a non-food location (e.g., Jewish Center), and whether the tweet should be excluded because it was not race-related (relating to individuals, people, or culture).

Approach: Previous sentiment analysis was conducted using the Stochastic Gradient Descent Classifier in Python software version 2.7. This is a linear classifier that has the value “0” or “1” and is appropriate for classifying into only two groups. To accommodate these restrictions, the researchers joined the neutral category with the negative category as this gave more accurate results in preliminary analyses. The purpose of my study is to develop a multi-label algorithm that will classify tweets into three categories: positive, negative, or neutral.

I will test three multi-label classification algorithms: Multinomial Naive Bayes, Support Vector Classification (SVC), and Logistic Regression. To do multi-label classification I created a sparse matrix for each of the three categories. Labelers scored a tweet as a “1” in the “happy_manual” column if the tweet was positive and a “1” in the “sad_manual” column if the tweet was negative. I created a third column for the neutral tweets giving a score of “1” if neither of the previous columns were scored as “1”. To obtain multi-label classification I will train the algorithms column by column (columns are each label) against the tweets.

Tweets were preprocessed using the “re” package in python for regular expressions. Words with contractions were replaced with whole words, url strings were removed along with punctuation. The text was featurized using both CountVectorizer and TfidfTransformer from SKlearn. The featurized data were first visualized using an ecdf (empirical cumulative distribution function) to determine the hyperparameters df_min and df_max which will be used to tune the featurizer during classification. It also gives a good overview of the variation in the data. Most of the variation is contained in the first 25 tweets, showing that there is not a great deal of overall variation in the tweets.

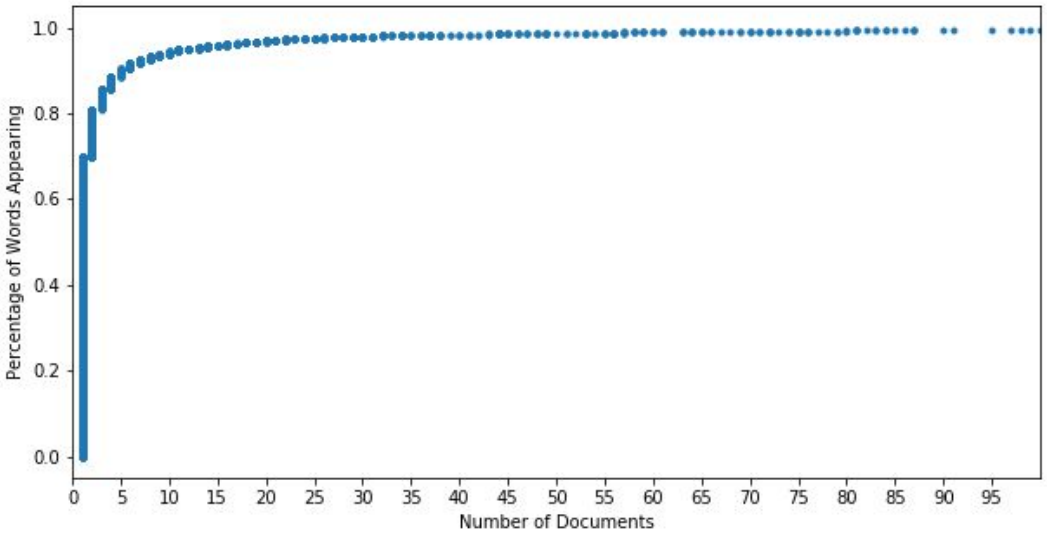

Figure 1. ECDF for tweets in the corpus.¶

Next, the featurized data was used to generate word cloud visualizations. There was a qualitative difference between the word clouds generated for the overall corpora, positive tweets, negative tweets, and neutral tweets. This suggests that a classifier is a reasonable method to use with this data.

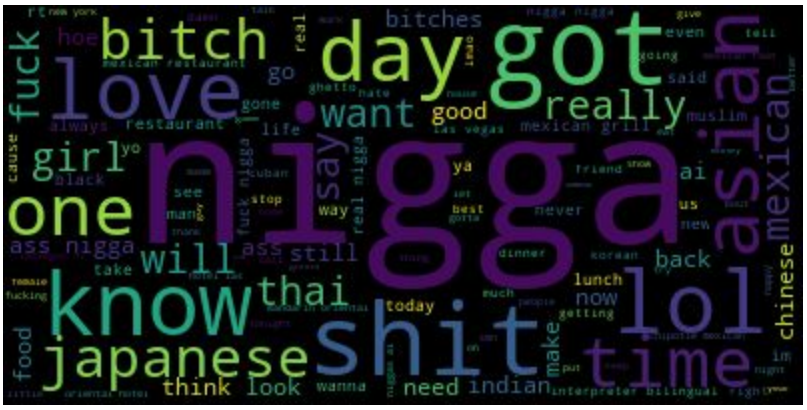

Fig 2. Word cloud for all the tweets.

There is a mix of positive and negative words. The highest used word is “nigga.” Looking at the overall corpora alone it is difficult to give meaning or explanation. The following three images show some striking differences.

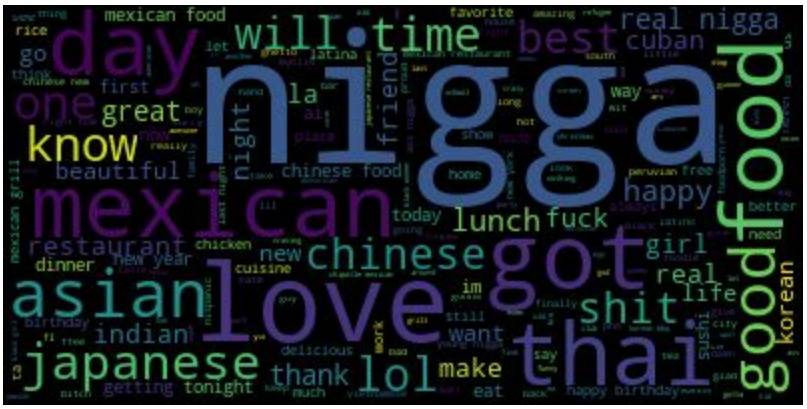

Fig 3. Word cloud for positive tweets.

In the positive tweets many of the swear words are missing that were seen in the overall word cloud. Positive words such as "beautiful", "happy", "thank" and "good" are present.

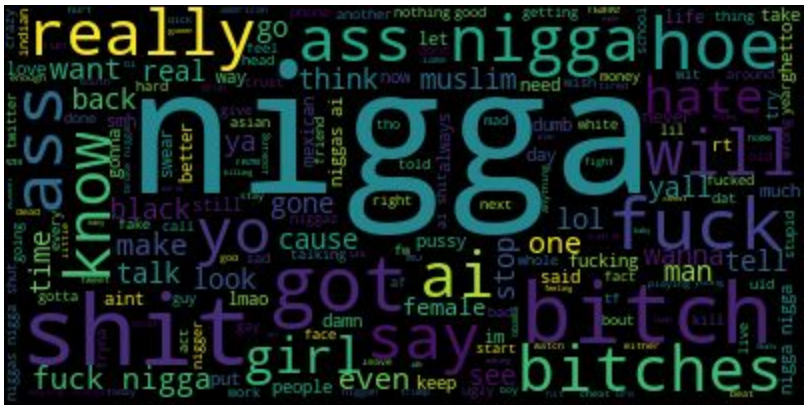

Fig 4. Word cloud for negative tweets.¶

This word cloud contains many swear words! Note that 'girl' is larger here, indicating that it is present more often. Here also are disparaging words used against females. Negative words such as "stop" and "hate" are seen.

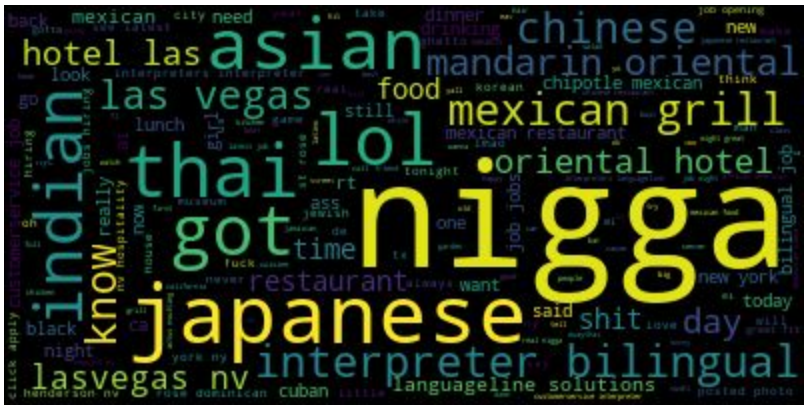

Figure 5. Word cloud for neutral tweets.¶

There are fewer swear words. Many more words referring to ethnic food and "las vegas" as well as “hotel”.

Conclusion: These word clouds indicate that there is a qualitative difference in the labeled tweets for positive, negative, and neutral categories. All of the categories have the word "nigga" used predominantly, giving this word an ambiguous meaning. This illustrates the difficulty that an unsupervised classifier or cluster algorithm would have in determining sentiment.

The next stage will be to test supervised classifiers and develop a model that can predict the labels "positive", "negative", and "neutral" on unlabeled tweets.

Model Selection

I am interested in classifying the highest number of positive labels for each category (note that here we are using 'positive' to mean a classification of '1' versus '0', not the 'positive' label for race sentiment). For this, I maximized recall, sometimes called sensitivity, which is the true positive rate. Recall is the ratio of positive instances that are correctly detected by the classification algorithm. The numerator is the number of true positives (TP) - tweets classified belonging to a category when they actually belong to the category determined by the labeled data. The denominator is the sum of the true positives plus the number of false negatives (FN) - tweets classified as not belonging to a category when they actually do belong to the category. False negatives should have been classified as positive (belonging to a category). Recall tells us how well the classifier detects positive instances.

$$Recall = \frac{TP}{TP + FN}$$

This is contrasted with precision, which is the accuracy of the positive predictions. The numerator is the same, but the denominator is the sum of true positives and false positives - tweets that should have been classified as positive for a category, but were not. High precision will lower the number of positively classified instances, but they will be more certain in the classification. High precision is good for scenarios where you must be as correct in a prediction as possible. Examples are recommendation systems where you want the recommended content to be correct and can sacrifice having larger numbers of recommended content.

$$Precision = \frac{TP}{TP + FP}$$

If we were concerned only with getting the highest accuracy possible, we would tune on precision, but the cost would be that we would predict much fewer (albeit correct) positive instances. Tuning for recall will lower the accuracy, but we will lower the risk of leaving some positive instances out. The difficulty (and interesting challenge) is that there is a tradeoff between recall and precision. Increasing one will reduce the other. The F1 score gives a picture of how well the classifier performs for each. It is the harmonic mean between precision and recall. This score is high for classifiers that have similar precision and recall.

$$F1 = 2\frac{P \times R}{P + R}$$

Finally, the area under the ROC curve (AUC) is a way to compare classifiers using above metrics. It is the area under the curve plotted as the True Positive Rate (Recall) against the False Positive Rate (the ratio of the negative instances ('0') that are incorrectly classified as positive ('1'). A perfect classifier will have an AUC = 1, and one that is purely random will equal 0.5. So, we are looking for the highest AUC score above 0.5.

Since our data is highly unbalanced for each category, there will be far more negative instances than positive instances, the ROC curve isn't the best metric to use. Instead, it is better to use a Precision-Recall Curve. However, this is not easily done in Sklearn with every classifier. It depends on calculating the decision_function, which isn't available for every classifier (e.g., MultinomialNB or RandomForestClassifier).

To evaluate these classifiers I need a metric that is shared among them all. Therefore I will look at the Recall, F1 and the AUC scores. Best overall score is given to the best out of 3. In the case of a tie, it will go to best recall.

Category	Best Recall	Best F1	Best AUC	Best Overall
Positive	LinearSVC	LinearSVC	MultinomialNB	LinearSVC
Negative	MultinomialNB	MultinomialNB	MultinomialNB	MultinomialNB
Neutral	GradientBoosting Classifier	GradientBoosting Classifier	LogisticRegression	GradientBoosting Classifier

We want to predict the highest number of tweets in each category and are willing to accept a few incorrect classifications in order to catch all the correct classifications. These metrics attend to both precision and recall so that the overall choice of classifiers for each category will minimize false negative claims.

Conclusion: The positive label is best predicted using a tuned LinearSVC classifier. The negative label is best predicted using a tuned MultinomialNB classifier, and the neutral label is best predicted using a GradientBoosting Classifier.

My goal is to generate a U.S. cloropleth heatmap to show the geographical distribution of racial sentiment in tweets. The Federal Information Processing Standard Publication 6-4 (FIPS) is a five-digit Federal Information Processing Standards code that uniquely identifies counties in the United States. Each tweet is associated with a FIPS county code. A heatmap will be created using predicted tweets. I will designate the majority label for each county by taking the label with the highest count per 1,000 population and assign it to the FIPS code for that county.

References

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. Proceedings of the 23rd International Conference on Machine Learning - ICML '06. doi:10.1145/1143844.1143865

Géron, Aurélien. Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly, 2018.

Sentiment analysis. (2018, December 10). Retrieved from https://en.wikipedia.org/wiki/Sentiment_analysis

Smetanin, S. (2018, September 01). Sentiment Analysis of Tweets using Multinomial Naive Bayes. Retrieved from https://towardsdatascience.com/sentiment-analysis-of-tweets-using-multinomial-naive-bayes-1009ed24276b