Twitter Disaster Analysis

Maria Mahbub, Linsey Sara Passarella, Emily Joyce Herron and Gerald Leon Jones

University of Tennessee, Knoxville

***Abstract*—In this paper, a shrewd observation on two types of sentiment analysis and two types of topic modeling on " Hurricane Florence Twitter Data set" have been presented.**

I. Introduction

A. Motivation

Hurricanes pose both an economic and social threat to communities. By assessing tweets concerning the impact of a hurricane across multiple cities, we can better understand the differences in attitudes towards preparedness, disaster clean-up, and damage costs in relation to the socioeconomic status of an area. An article published by Masozera et [1] in 2006 showed that low-income areas in New Orleans were more vulnerable during response and recovery efforts after Hurricane Katrina. While flood damages occurred regardless of area or neighborhood, a number of limiting factors including transportation issues, lack of quality insurance (if any) and a higher number of poorly constructed houses and buildings.

B. Objective

Many sources of information tell us how low income and rural areas are more adversely affected during a disaster. Our assumption was that lower income areas would show more of a negative impact. We were interested to see if analyzing tweets from these different areas around the time of a disaster could reveal a difference in how these two types of areas were impacted. Our idea was to use text analysis such as sentiment analysis and clustering to uncover this possible disparity. We gathered tweets around the time of hurricane Florence and used two methods of sentiment analysis, the sentiment polarity function of Python's TextBlob library and a MATLAB support vector machine, as well as two methods of clustering, LDA topic modeling and k-means clustering, to see if the suspected emotional impact could be revealed.

II. Our Methodology

A. Data Collection

The tweets containing '#hurricaneflorence' have been obtained by using Python. The codes that have been used to get the tweets can be found in this Github repository: https://github.com/DisasterMasters. It is built on Scrapy and helped to get tweets from Twitter Search without using Twitter API's. Though the crawled data is more messy than the one obtained by the API's, the benefits of doing so was not getting restricted by the API's rate limits and thus tweets were collected as much as needed. For this particular study, more than 200,000 tweets have been collected.

B. Data Cleaning

To prepare for analyzing the tweets, hash tags '#', stopwords, punctuation marks and usermentions were removed along with embedded URLs which were not part of the main text of the tweet. After cleaning, the tweets were tokenized.

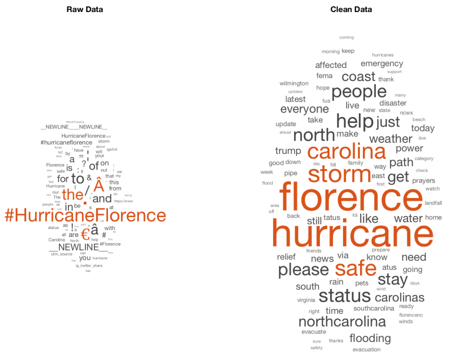


Fig. 1. Word clouds of the raw and cleaned Florence twitter dataset.

In Python this step was accomplished using the gensim library's utils.simple_preprocess function, which accepted a tokenized version of each tweet and returned a list of lowercase tokens, with all links, hashtags, and punctuation removed. This function was used alongside a conditional that removed all tokens with length less than three or that were found in gensim's parsing.preprocessing.STOPWORDS list.

C. *Analysis of Low-income Related Tweets*

A list of words pertaining to low-income or rural areas was created to partition the tweets for analysis. If the tweet contained one of the following words, it would be separated into the low-income list: rural, poor, community, communities, country, countryside, neighborhood, impoverished, poverty, broke, underprivileged, low, income, low-income.

An example of a tweet in the data set containing one of these key words is "In Poor, Rural Communities, Fleeing Hurricane Florence Was Tough".

The proportion of tweets identified in Python as low income and those that were not are shown in Figure 2.

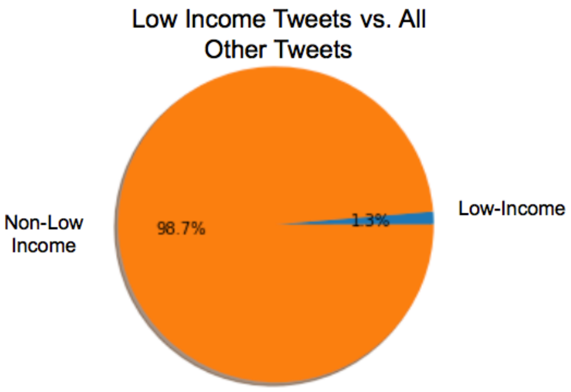


Fig. 2. Proportion of tweets classified as low income and other.

D. *Topic Modeling*

Once all the tweets have been cleaned, tokenized and divided into two groups: rural and non-rural; Latent Dirichlet Allocation (LDA) topic modeling was performed on these groups of tweets. The core package that has been used was scikit-learn (sklearn). The libraries numpy, pandas and PyLDAvis, matplotlib were used for manipulating, viewing data in tabular format and visualization respectively.

The LDA topic model algorithm requires a document word matrix as the main input. To qualify for being a member in this matrix a word needs to be appeared in the corpus at least ten times and be composed of at least three alphanumeric characters. To start the process of creating word-matrix, the words have been converted to lowercase at first, then the CountVectorizer class with necessary configuration has been considered and at the end fit_transform has been applied to create the matrix.

The most important tuning parameter for LDA models is the number of topics. The parameter which controls the learning rate has been considered as well. To achieve the best parameters, a range of number of topics and learning rates have been used to develop the best LDA model.

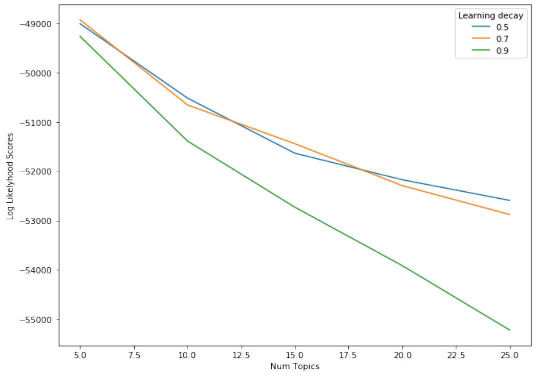


Fig. 3. Choosing optimal LDA model (rural tweets)

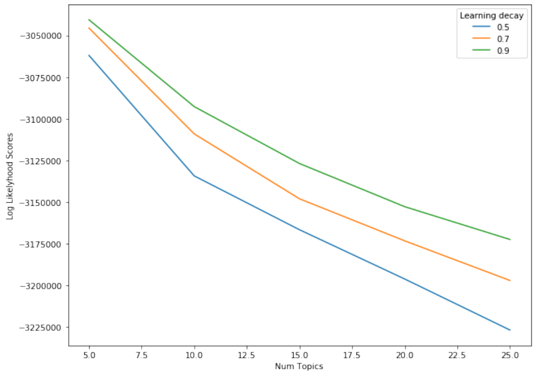
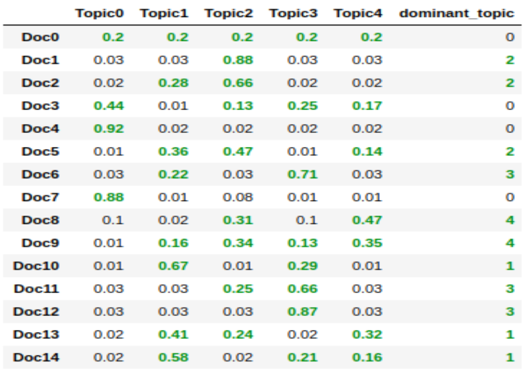


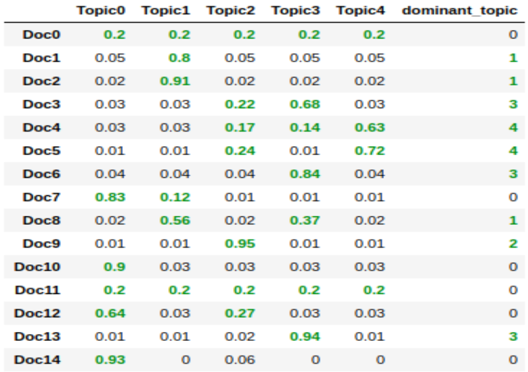
Fig. 4. Choosing optimal LDA model (non-rural tweets)

By plotting the log-likelihood scores against number of topics (figure 3 and 4), it can be clearly seen that for rural tweets the number of topics = 5 has better scores together with learning decay of 0.7 and for non-rural tweets the number of topics = 5 has better scores together with learning decay of 0.9. To diagnose the model performance, model perplexity and Log likelihood score have been considered and as known model with higher log-likelihood and lower perplexity (exp(-1. * log-likelihood per word)) is considered to be well performing. In this case for rural tweets: the log-Likelihood is -48929.051834740676 and perplexity



	Topic0	Topic1	Topic2	Topic3	Topic4	dominant_topic
Doc0	0.2	0.2	0.2	0.2	0.2	0
Doc1	0.03	0.03	0.88	0.03	0.03	2
Doc2	0.02	0.28	0.66	0.02	0.02	2
Doc3	0.44	0.01	0.13	0.25	0.17	0
Doc4	0.92	0.02	0.02	0.02	0.02	0
Doc5	0.01	0.36	0.47	0.01	0.14	2
Doc6	0.03	0.22	0.03	0.71	0.03	3
Doc7	0.88	0.01	0.08	0.01	0.01	0
Doc8	0.1	0.02	0.31	0.1	0.47	4
Doc9	0.01	0.16	0.34	0.13	0.35	4
Doc10	0.01	0.67	0.01	0.29	0.01	1
Doc11	0.03	0.03	0.25	0.66	0.03	3
Doc12	0.03	0.03	0.03	0.87	0.03	3
Doc13	0.02	0.41	0.24	0.02	0.32	1
Doc14	0.02	0.58	0.02	0.21	0.16	1

Fig. 5. Dominant topics for first 15 rural tweets



	Topic0	Topic1	Topic2	Topic3	Topic4	dominant_topic
Doc0	0.2	0.2	0.2	0.2	0.2	0
Doc1	0.05	0.8	0.05	0.05	0.05	1
Doc2	0.02	0.91	0.02	0.02	0.02	1
Doc3	0.03	0.03	0.22	0.68	0.03	3
Doc4	0.03	0.03	0.17	0.14	0.63	4
Doc5	0.01	0.01	0.24	0.01	0.72	4
Doc6	0.04	0.04	0.04	0.84	0.04	3
Doc7	0.83	0.12	0.01	0.01	0.01	0
Doc8	0.02	0.56	0.02	0.37	0.02	1
Doc9	0.01	0.01	0.95	0.01	0.01	2
Doc10	0.9	0.03	0.03	0.03	0.03	0
Doc11	0.2	0.2	0.2	0.2	0.2	0
Doc12	0.64	0.03	0.27	0.03	0.03	0
Doc13	0.01	0.01	0.02	0.94	0.01	3
Doc14	0.93	0	0.06	0	0	0

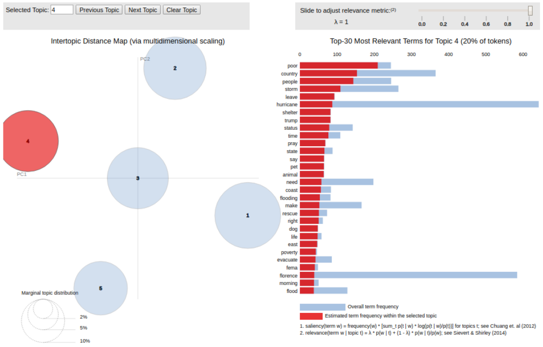
Fig. 6. Dominant topics for first 15 non-rural tweets

is 359.5883974528648 and for non-rural tweets the log-Likelihood is -3040556.7561088563 and perplexity is 1496.403047720928 which indicate well performing models.

To classify the tweets as belonging to a particular topic, the approach that has been adopted is to see which topic has the highest contribution to them and assign it to the particular tweet. Here each tweet is considered to be a separate document. After the most dominant topic has been assigned to each of the documents (figure 5 and 6), LDA models for each category have been implemented.

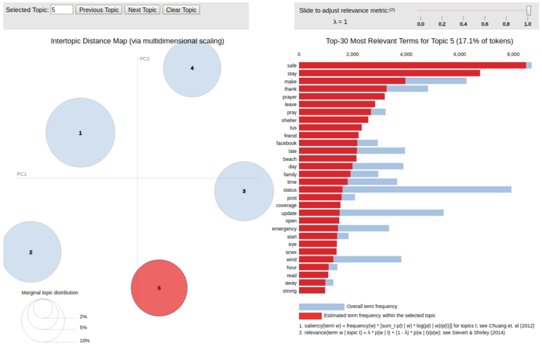
In figure 7 and 8, visualization of LDA models has been represented. Here each circle represents different topics, the blue bars represent overall term frequency whereas the red bars represent the term frequency within the selected topic.

In figure 9 and 10, top 10 keywords that represent the topics for rural and non-rural tweets are shown. From those keywords, the topic distributions can be inferred as for the rural tweets can be: social media (15.4%), hurricane & media (20.1%), help & relief (23.4%), safety measure & concern (21%), poverty (20%) and for non-rural tweets: disaster & impact (18%), social media (19%), hurricane & media (25.9%), safety measure & concern (17.1%), help & relief (20.1%)



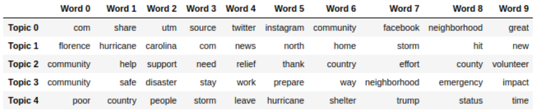
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 7. LDA Topic Model (rural tweets)



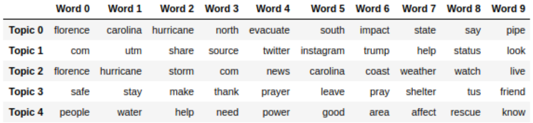
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 8. LDA Topic Model (non-rural tweets)



	Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9
Topic 0	com	share	utm	source	twitter	instagram	community	facebook	neighborhood	great
Topic 1	florence	hurricane	carolina	com	news	north	home	storm	hit	new
Topic 2	community	help	support	need	relief	thank	country	effort	county	volunteer
Topic 3	community	safe	disaster	stay	work	prepare	way	neighborhood	emergency	impact
Topic 4	poor	country	people	storm	leave	hurricane	shelter	trump	status	time

Fig. 9. Top ten keywords per topic (rural tweets)



	Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9
Topic 0	florence	carolina	hurricane	north	evacuate	south	impact	state	say	pipe
Topic 1	com	utm	share	source	twitter	instagram	trump	help	status	look
Topic 2	florence	hurricane	storm	com	news	carolina	coast	weather	watch	live
Topic 3	safe	stay	make	thank	prayer	leave	pray	shelter	tus	friend
Topic 4	people	water	help	need	power	good	area	affect	rescue	know

Fig. 10. Top ten keywords per topic (non-rural tweets)

E. K-Means

In addition to LDA, K-Means clustering was used to cluster similar tweets. To apply this method, the text of each tweet was first vectorized using Scikit-Learn's TfidfVectorizer function in order to represent the text corpus as a matrix of TF-IDF values. The vectorized tweets were then clustered using different values for k. An optimal k-value of 11 was chosen by calculating the sum of squared distances for each clustering, graphing these values and applying the elbow method, as shown in Figure 11.

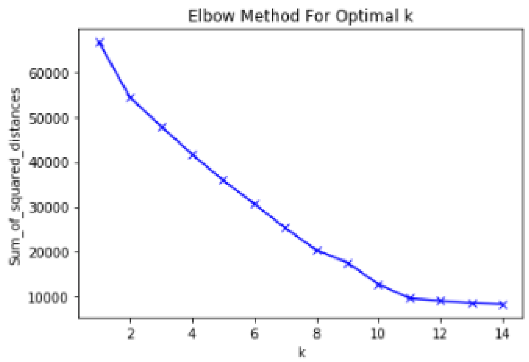


Fig. 11. An optimal k of 11 was chosen by calculating the sum of squared distances for each K-Means cluster and applying the elbow method.

The top 10 words contained in each of the 11 clusters are listed in Table 1.

Scatter plots of the first two principal components of all tweets and the low-income related tweets with their color-coded cluster assignments are shown in Figure 12. The proportions of tweets falling into each cluster in all tweets in the data set and in the low-income subgroup alone are visualized in the pie charts in Figure 13.

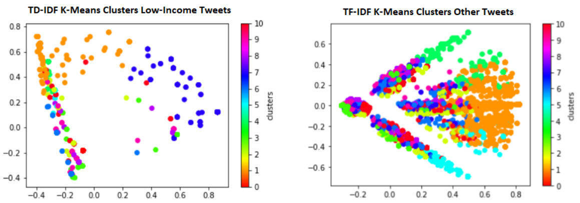


Fig. 12. Scatterplots of first two principal components of TF-IDF vectors of all tweets and low-income/rural subset with color-coded K-Means cluster assignments.

As shown in Table I, each cluster seemed to have relatively similar values for its most common words. The key difference between the clusters is each word's rank in popularity. By comparing proportions of tweets falling into each cluster in the two socioeconomic subsets of tweets, as shown in Figure 13, the rural/low-income subset of tweets had a smaller proportion of tweets falling into cluster 0 and a larger proportion of tweets representing clusters 1 and 2.

TABLE I
MOST COMMON WORDS IN EACH K-MEANS CLUSTER FOR K=11.

Cluster	Top Most Common Words
0	utm_source, storm, status, safe, news, hurricanes, helping, florence, carolina
1	florence, hurricanes, hurricanes, news, carolina, storm, helping, utm_source, safe, status
2	carolina, florence, hurricanes, hurricanes, news, storm, utm_source, safe, helping, status
3	status, florence, safe, hurricanes, helping, storm, carolina, hurricanes, news, utm_source
4	hurricanes, florence, news, storm, utm_source, carolina, safe, helping, status, hurricanes
5	florence, hurricanes, news, carolina, storm, utm_source, helping, safe, status, hurricanes
6	storm, florence, hurricanes, carolina, hurricanes, news, utm_source, safe, status, helping
7	helping, florence carolina hurricanes storm hurricanes, news utm_source status safe
8	utm_source, hurricanes, storm, carolina, safe, florence, helping, news, status, hurricanes
9	safe, storm, carolina, hurricanes, florence, helping, status, hurricanes, news, utm_source
10	news, florence, carolina, hurricanes, storm, helping, status, safe, hurricanes, utm_source

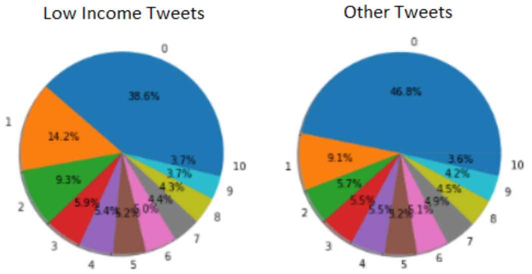


Fig. 13. K-Means cluster proportions for all tweets and low-income subgroup.

1) Comparison of LDA and K-Means for Topic Modeling: Both K-means and Latent Dirichlet Allocation (LDA) are unsupervised learning algorithms with a pre-decided value of K. When LDA and k-Means have been compared with one another for our set of tweets, a major difference has been noticed. In case of LDA, the optimum number of topic distribution was 5 where as in case of k-Means it was 11. This may have happened because when K-Means was applied to assign K topics to N number of documents it divided the document into K disjoint clusters (topics). On the other hand, LDA assigned each document to a mixture of topics resulting in characterized by one or more topics. These results illustrate the benefits and drawbacks of using either method for topic modeling. LDA is beneficial for this application in that it produces a list of topics in the text corpus alongside list of specific keywords representing each topic. However, a disadvantage of LDA is that it is not designed to assign topics to individual documents. On the other hand, K-Means assigns each item, or document, to a specific cluster based on similarity. The drawback of K-Means for his application was that does not directly yield a list of topics defining each cluster.

2) Combining LDA and K-Means: In order to employ the benefits of both topic modelling approaches, an LDA model was applied to the tweets in each of the K-Means clusters. The result was a list of LDA topics containing terms defining each cluster of tweets. Table II lists the top ten terms from topic 0 of each cluster. In future work, this approach might be expanded further analyze the topics defining each cluster. These results may furthermore provide a more robust approach for refining the tweet data set and identifying tweets pertaining to rural or low-income areas.

TABLE II

LDA Topic 0 Terms for K-Means Clusters

Cluster	LDA Topic 0 Terms
0	relief, state, emergency, support, disaster, efforts, fema, victims, affected, response
1	florence, southcarolina, update, northcarolina, ncwx, newbern, video, wind, evacuation, emergency
2	Words: florence, hurricane, trump, people, relief, affected, victims, help, power, status
3	carolina, carolinas, north, northcarolina, florence, southcarolina, weather, hurricane, week, flooding
4	status, fema, storm, like, people, evacuate, ready, wednesdaywisdom, realdonaldtrump, surge
5	hurricane, ence, flor, fema, hurricanes, northcarolina, insurance, victims, html, flood
6	storm, storms, surge, hurricane, rain, florence, wind, scwx, weather, landfall
7	utm_source, instagram, igshid, hurricane, northcarolina, like, need, southcarolina, ready, love
8	help, need, impacted, storm, hurricane, people, florence, relief, victims, disaster
9	safe, stay, carolina, help, people, thoughts, thinking, carolinas, north, responders
10	news, florence, hurricane, html, storm, live, carolina, north, carolinas, weather”

F. Sentiment Analysis

1) Python Text Blob: Sentiment analysis was preformed on text of each tweet using Python’s TextBlob library for natural language processing. All tweets in the Hurricane Florence data set were passed to the TextBlob class’s sentiment function, which yielded a polarity value between -1.0 and 1.0 for each, indicating ta sentiment classification of either negative (sentiment < 0), positive (sentiment > 0), or neutral (sentiment $= 0$). This analysis classified 46,225 tweets as positive, 17,357 as negative, and 75,625 as neutral. Word clouds constructed from the words of tweets identified as either positive, negative or neutral are shown in Figure 14. The sentiment percentage breakdowns of tweets in the entire data set, the subset of tweets containing rural/low-income related terms, and all other tweets are shown in the pie chart in Figure 15. By comparing the percentages of positive, negative, and neutral tweets each group, we observe that the percentage of neutral tweets is lower in the rural/low-income subset than in the set of all other tweets. Meanwhile the proportions of tweets classified as negative and positive were higher in this subset in comparison to that of all other tweets, with the largest percentage of tweets in this subset having a classification of positive.

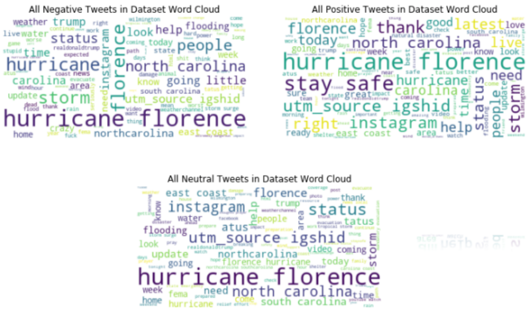


Fig. 14. Words clouds constructed from tweets classified as positive, negative or neutral by Python’s TextBlob sentiment polarity function.

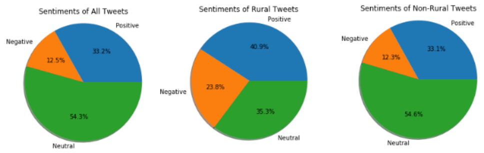


Fig. 15. Percentages of Tweets Classified as positive, negative, or neutral according to Python's TextBlob function.

2) Support Vector Machine: A support vector machine (SVM) was used as one method of conducting a sentiment analysis on the data sets. A sentiment word list compiled by the University of Illinois Chicago was used to train the SVM to predict the sentiments of tweets used in this study. The sentiment word list contained 6,790 words that were categorically labeled either "positive" or "negative".

The words in both the sentiment list and the tweet data sets were converted to word vectors using MATLAB's built-in word embedding. After the SVM was trained, it ran on the Florence twitter data set and output a sentiment score for the individual words in the tweet. The scores ranged from 0 being a neutral sentiment, $>$ 0 being positive and $<$ 0 being negative. A word cloud of predicted positive and predicted negative words in the Florence Data set is shown in figure 16.

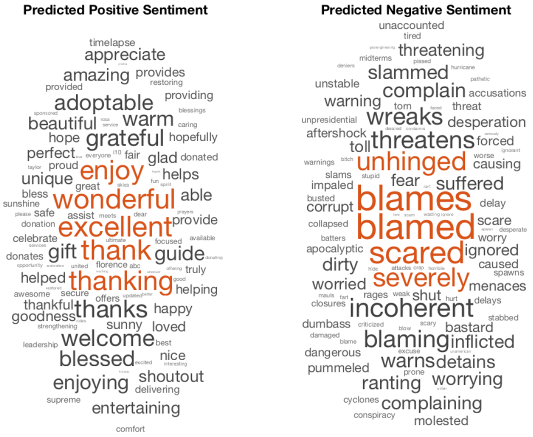


Fig. 16. Word clouds of the predict sentiments in the data set.

The SVM predicted the sentiment of every word within the tweet, as opposed to the sentiment of the tweet as a whole. There were several instances of words that were not recognized by the built-in word embedding, such as names of organizations (for example cdcemergency) or misspelled words. These words were given a sentiment score of NaN by the SVM and therefor removed.

The confusion matrix (figure 17) reports the accuracy of the trained SVM on unseen testing data.

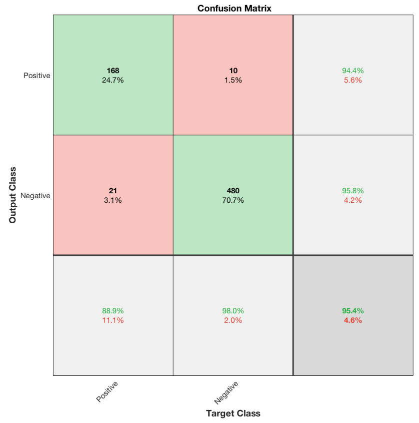


Fig. 17. Classification accuracy for the trained SVM on testing data.

The results of the support vector machine are shown in the pie graph (figure 18) below. Overall, the non-rural area related tweets had the highest percent of negative tweets, but not by a significant amount. There were a few issues with the SVM approach to predicting sentiment in the twitter data set. In the sentiment word list, there were no categorically labeled "neutral" sentiments. Instead, sentiment scores between -0.3 and 0.3 where counted as "neutral". Similarly, the words that were not recognized by the word embeddings were not

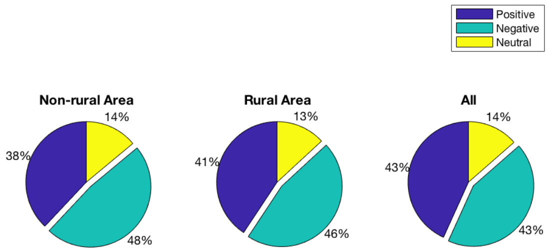


Fig. 18. Percentage of predicted sentiment in the entire data set, rural related tweets and non-rural related tweets.

used in the analysis. If they were correctly classified, they could have a large impact on the results. Future work should find alternative methods of imputation.

3) Comparison of TextBlob and SVMs for Sentiment Analysis: The results of the sentiment analyses surprised us. The scores for the both types of sentiment analysis can be seen in table III. According to both the Matlab and Python sentiment analysis, the rural tweets had a higher percentage of positive tweets compared to the non rural tweets. For rural tweets the python analysis yielded a positive score of 40.9%, while the non rural tweets received a positive score for only 33.1%. The Matlab analysis resulted in 41% of the rural tweets receiving a positive score and only 38% of the non rural tweets getting positive score. While these differences are not massive they run counter to our initial assumption that the rural tweets would show a more negative sentiment during the time of the hurricane. It was also interesting that the python analysis results indicated that the rural tweets were mostly (40.9%) positive, while with Mat lab the outcome of the analysis shows that the rural tweets are mostly (46%) negative. Differences can also be observed between the Non rural tweet results across the different languages. In the Python results the non rural tweets scored mostly neutral (54.6%) while the Matlab analysis scored them as mostly negative (48%). These differences indicate that there are some discrepancies in the sentiment analysis results using Python TextBlob verses Matlab SVM. The different analysis types also suggest differing overall sentiment scores. Python TextBlob suggests that the overall tweets were neutral (54.3%), while the Matlab SVM shows an even split between (43%, 43%) positive and negative sentiment. These differences in results could arise simply from the differences in the analysis methods or more subtle causes.

TABLE III
PYTHON VS. MATLAB ANALYSIS

	Python			Matlab		
Sentiment	All	Rural	Non	All	Rural	Non
Positive	33.2%	40.9%	33.1%	43%	41%	38%
Negative	12.5%	23.8%	12.3%	43%	46%	48%
Neutral	54.3%	35.3%	54.6%	14%	13%	14%

III. CONCLUSION

As stated above the LDA analysis resulted in a less complex model of 5 classes verses 11 classes suggested by the k means analysis. The LDA gave us a more concise and possibly easier to understand way of classifying the tweets. A down side of the LDA analysis was that is was much more time consuming to run. LDA clustering would need hours to complete. The LDA modeling is also not suited to assigning topics to individual tweets. The k means topic modeling seemed to be better suited for mass clustering of data and was much simpler to implement, but it was harder to interpret what the clusters meant subject wise. We used a mixture of the two in an attempt to use the easier to understand clusters from the LDA to get a better understanding of the clustering that resulted from the k means clustering.

The two sentiment analysis methods we employed ended up both suggesting that the rural tweets scored a more positive sentiment score than the non rural tweets, but their overall results came to slightly different conclusions about the general distributions of the sentiments in the tweets. It can be seen then that the two sentiments analysis methods can come to slightly different conclusions when analyzing the same data set. The way we parsed the rural and non rural tweets may also be a cause of the conflicting results. Our set of rural classifier words may not have been robust enough to pick out the rural tweets. Our method for parsing the rural tweets was based solely on key words and used no geographic data. Because of this, we may have missed or miss classified a number of tweets that could have been in our rural tweet set. Using our parsing method only 13% of the 200,00 tweets were classified as rural/low income so there could be more hidden in the data. These missed tweets may have changed our results. It is also possible that sentiment analysis is just not an efficient or accurate way of observing the more negative effects endured by lower income rural areas during a disaster. Using sentiment analysis we looked for words or phrases and scored them as negative, positive or neutral, and this simple classification might not be good enough to show the known greater negative effect on rural communities. The negative physical and socioeconomic effects experienced by the low income areas might not show up in a quantifiable way in social media messages from the general population. These discrepancies may also come from the fact that text blob and support vector machine analysis are completely different techniques. The support vector machine method only scores positive and negative, and while we created a work around for this by creating our own neutral scoring method, it probably leads to different out comes with respect to what is classified as neutral. This difference could and probably does lead to different out comes when it comes to positive and negative scoring as well.

A. Limitations and Future Work

Our main problems were how to get the needed tweets, how to parse the tweets and how to analyze the tweets in such a way to test our hypothesis. Getting the tweets was fairly simple. We settled on a simple parsing method that may have been enhanced by using the geographic data from the tweets. The methods we used to scrape the tweets did not give us access to this information, but we are aware that there is a way to gain access to this data. Future work could find and use this data in the analysis and possibly get different results. We decided to analyze the tweets in a few different ways. This approach allowed us to back up our findings as well as compare and contrast the different techniques.

For future work if a more precise way of parsing the tweets into rural and non rural sets can be found perhaps more refined results or effects can be observed. If one were able to use the geographic information from the tweets one might be able to better parse the data. If a way of scoring tweets scored with the SVM as neutral that was more in line with the way the text blob scores neutral sentiment then perhaps the results could be more similar. Further research could be done on comparing and contrasting different sentiment and clustering analysis methods for this type of investigation. Someone could also look into combining the different methods of clustering or sentiment analysis in a similar approach to get some of the better aspects of one method and get more information out of another.

IV. Membership Roles

- Maria Mahbub: Data collection, LDA
- Linsey S. Passarella: Low-income tweet database, MATLAB Sentiment Analysis/SVM
- Emily J. Herron: K-Means, Python Sentiment Analysis
- Gerald L. Jones: Research, Comparison analysis, Goal Manager

References

[1] Charles Kerchner Michel Masozera Melissa Bailey. “Distribution of impacts of natural disasters across income groups: A case study of New Orleans”. In: (2007), pp. 299–306.