casanave / Hurri_Help Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Hurri_Help / README.md

casanave included pdfs as separate folder History

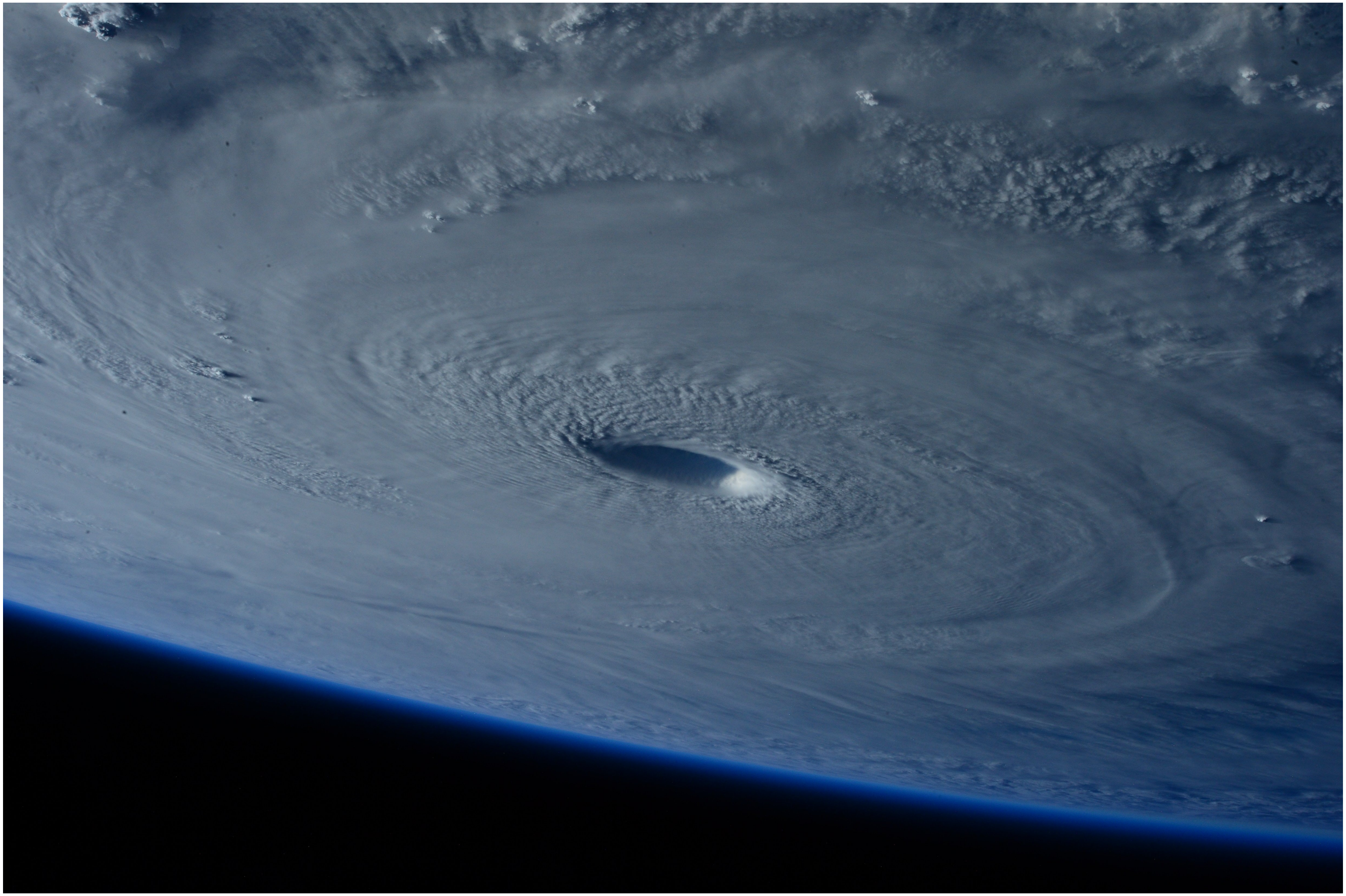
1 contributor

172 lines (125 sloc) | 7.52 KB

HurriHelp

A machine learning algorithm to help a twitterbot help those affected by Hurricane Ian

Presentation Slides: https://www.canva.com/design/DAFR2yfd9p0/juNA8udagCiO_l6kwt5WEg/view?utm_content=DAFR2yfd9p0&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink (the PDFs Attached lose some formatting)



Problem : People who survived Hurricane Ian need outreach to help connect them with FEMA resources and the National Disaster Distress Helpline.

Solution : A twitterbot that responds to people who are 1) using the #HurricaneIan hashtag 2) in distress.

Consider the following tweet:

Bitchin' Post🇺🇦🌊💙🏳️‍🌈
@WindDance79 · **Follow**

I just went through an almost category 5 hurricane. 5 minutes away from ground zero.

After 9 days, we got out power back, and in 10, I got my Twitter/internet back.

It was horrific, it was the scariest night of our lives.

So grateful to still be here.

#HurricaneIan

5:52 PM · Oct 7, 2022

Reply Copy link

Read more on Twitter

Vision: HurriHelp would respond to the above tweet with this message:

Data Collection: Scraped Twitter for data using Tweepy. Collected 7652 non Re-Tweeted tweets with the following additional features:

1 screen_name
2 user_description
3 favourite_count
4 retweet_count
5 created_at
6 replying_to
7 media
8 hashtags
9 urls
10 user_mentions
11 is_quote
12 is_retweet

*I only ended up using 'screen_name' in addition to the text of the tweet for cleaning. I want to go back and analyze all these features in the future, as is I only used the tweet text for modeling.

Label Production: Make labels by:

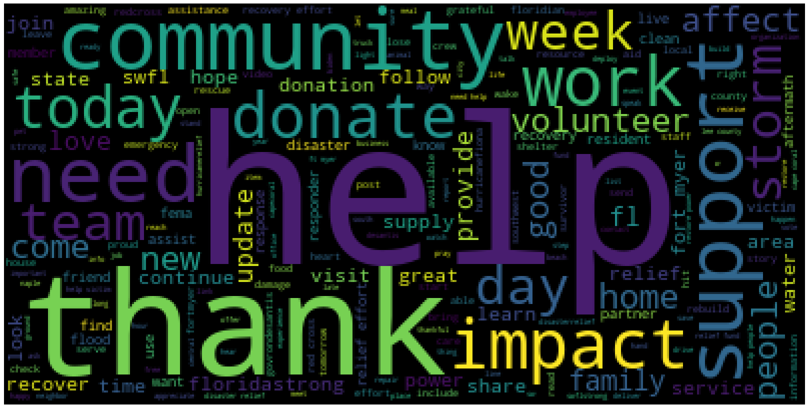
1. using Text_Blob, VADER and a version of a BERT model trained for sentiment analysis: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment?text=I+like+you.+I+love+you to make three different columns with numeric scores of sentiment
2. Used SKLearn's StandardScaler() to make all three sentiment analysis columns on same scale
3. Added all three scores into a 'final_score' column
4. Analyzed distribution and chose threshold for "Negative Sentiment" and "Positive Sentiment"

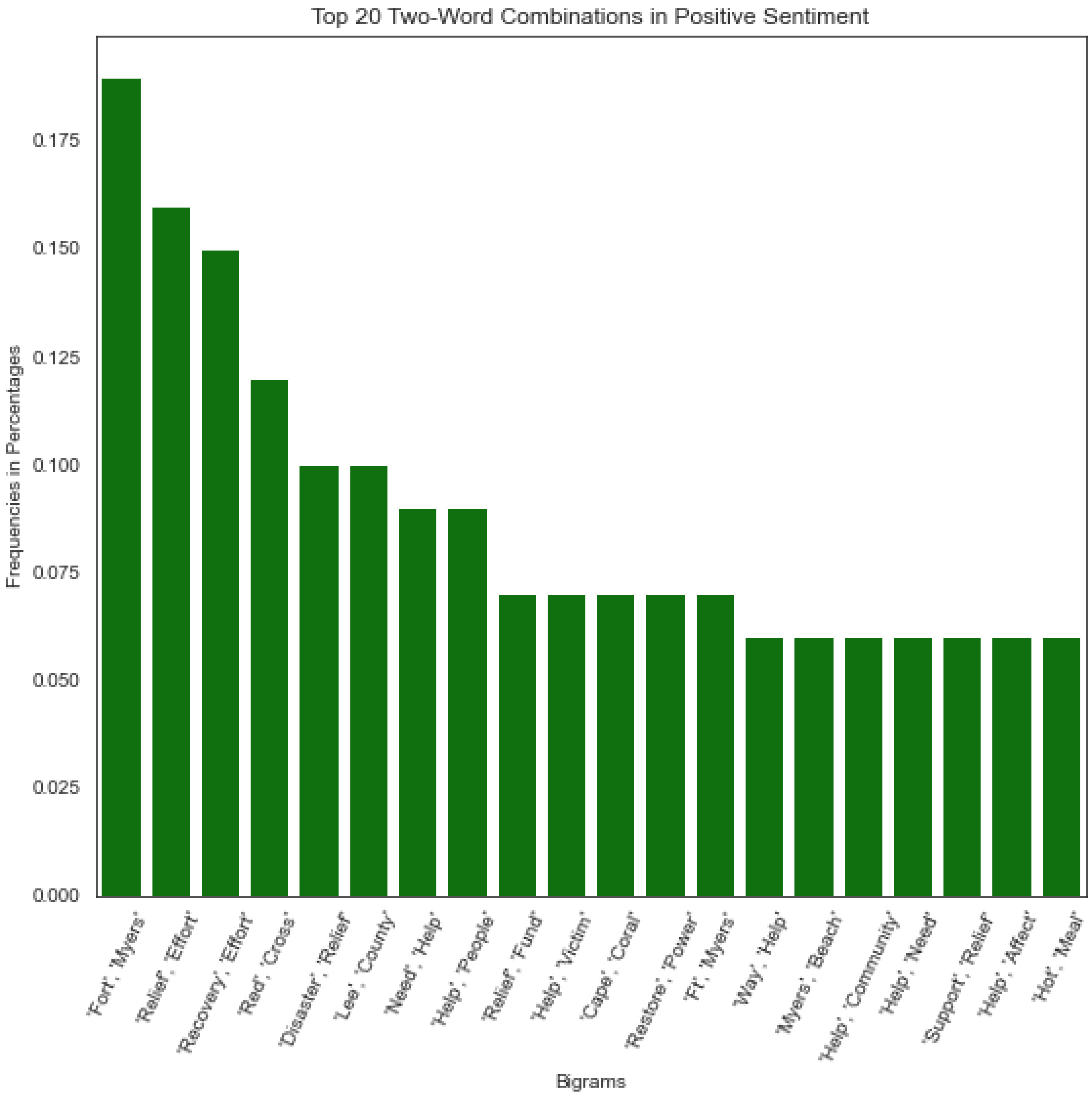
Cleaning the data:

1. created list of twitter users in data who tweeted more than 20 times and removed those tweets from data set to avoid spammers and news sights
2. used spaCy's list of stopwords and added 'Hurricane', 'Ian', and 'Florida'
3. elongated all contractions using contractions libary

4. used spaCy, NLTK and string manipulation to get list of lemmas in each tweet and remove punctuation, numbers, URLS, stopwords
5. used demoji to remove all emojis, after testing modeling with and without emojis and found no benifit to keeping emojis (see trail notebook)
6. final data set had 6773 original tweets

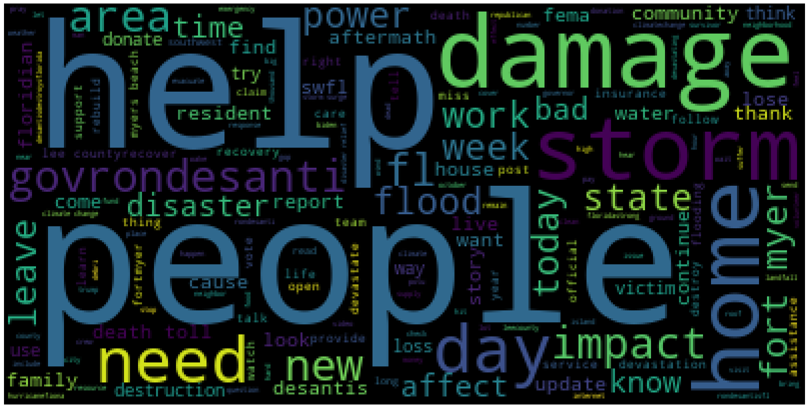
Analysis: Used WordCloud, Seaborn and NLTK's Bigrams to make the following visualizations-- Positive Sentiment:

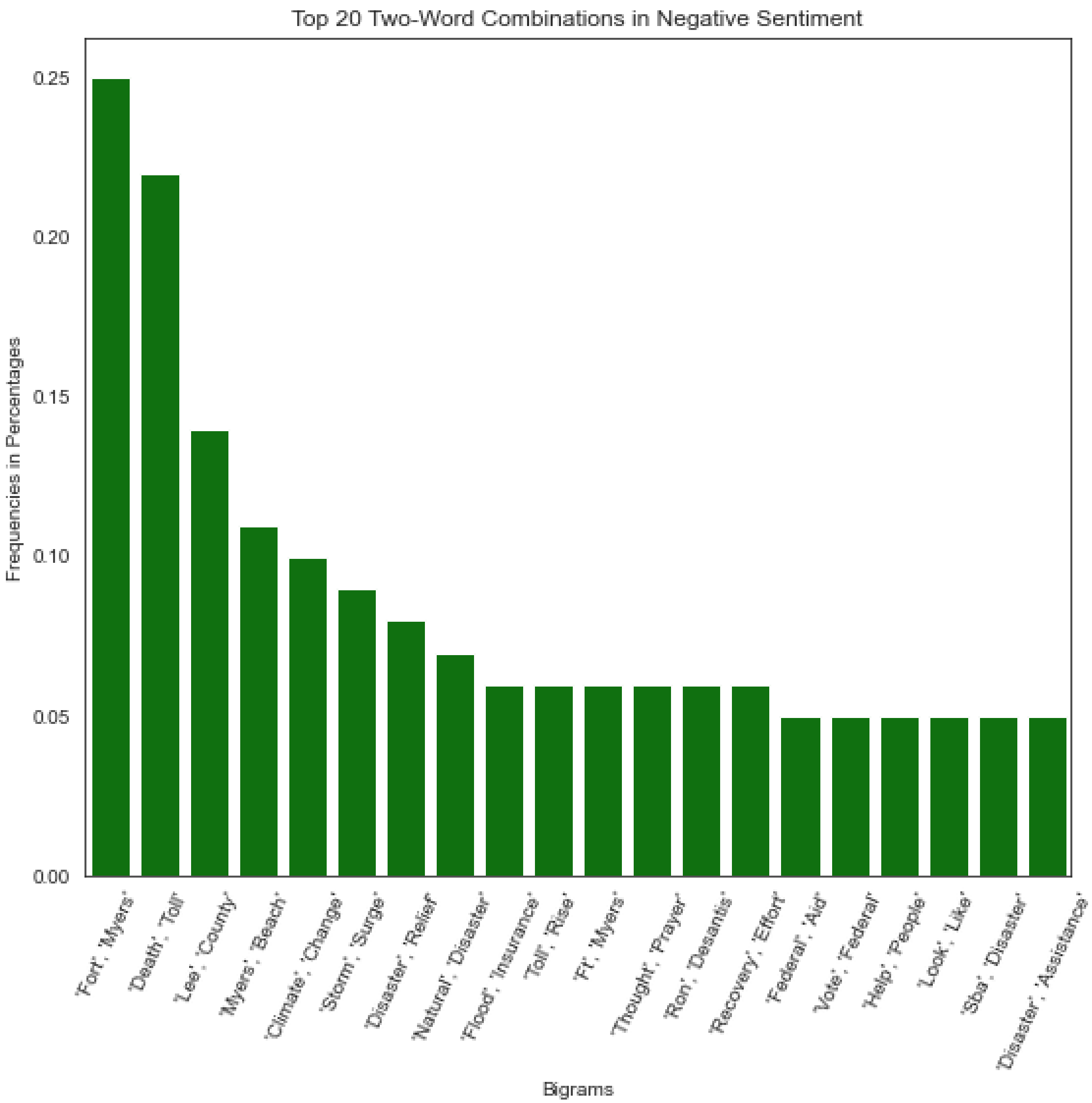




Observations: most common themes of words unique to Positive Sentiment were about basic needs being met.

Negative Sentiment:


help
damage
people
storm
home
need
day
impact
govrondesanti
area
power
flood
state
disaster
new
affect
know
work
week
today
fort
myer
leave
time
fl
bad
community
think
fema
aftermath
donate
find
try
resident
swfl
water
thank
lose
house
report
come
live
want
continue
victim
cause
way
story
death toll
look
provide
loss
update
family
destruction
desantis
use
devastation



Obersavations: most common themes of words unique to Negative Sentiment were about needing monetary help. Interestingly, also included "DeSantis".

Shared Themes of words included places and 'help'.

Vectorizing the Tweets:

1. Used SKLearn TFIDF Vectorizer on each tweet's string of lemmas
2. Used SKLearn Count Vectorizer on each tweet's string of lemmas

Train/Test/Split: 80%/10%/10%

MODELING METHODS:

- Predict sentiment from validation set
- Target: Negative Sentiment Tweets
- Machines ranked by Precision score ** decided to use precision score since both label-making and tweet filtering methods need improvement, and it's unhelpful to have bot spam users.

Machines Used:

- Random Forest
- XGBoost
- Naive Bayes
- CatBoost

Tuning Methods:

- Used GridsearchCV for all models except Naive Bayes, did 5 cross validates for each model

Comparing Models:

- Evaluated model with custom function producing classification report and ROC_AUC plot for Negative Sentiment class
- Made list of all Precision Scores from all models for eyeballing comparision

Final Model:

- Best model was CatBoost using a simple Count Vectorizer for term frequency
- Trained best model on all training data

- Validated Best Model for final Precision score of 80%
    - This model as best model since it did marginally better than other models and because it's highly interpretable

MODELING CONCLUSIONS:

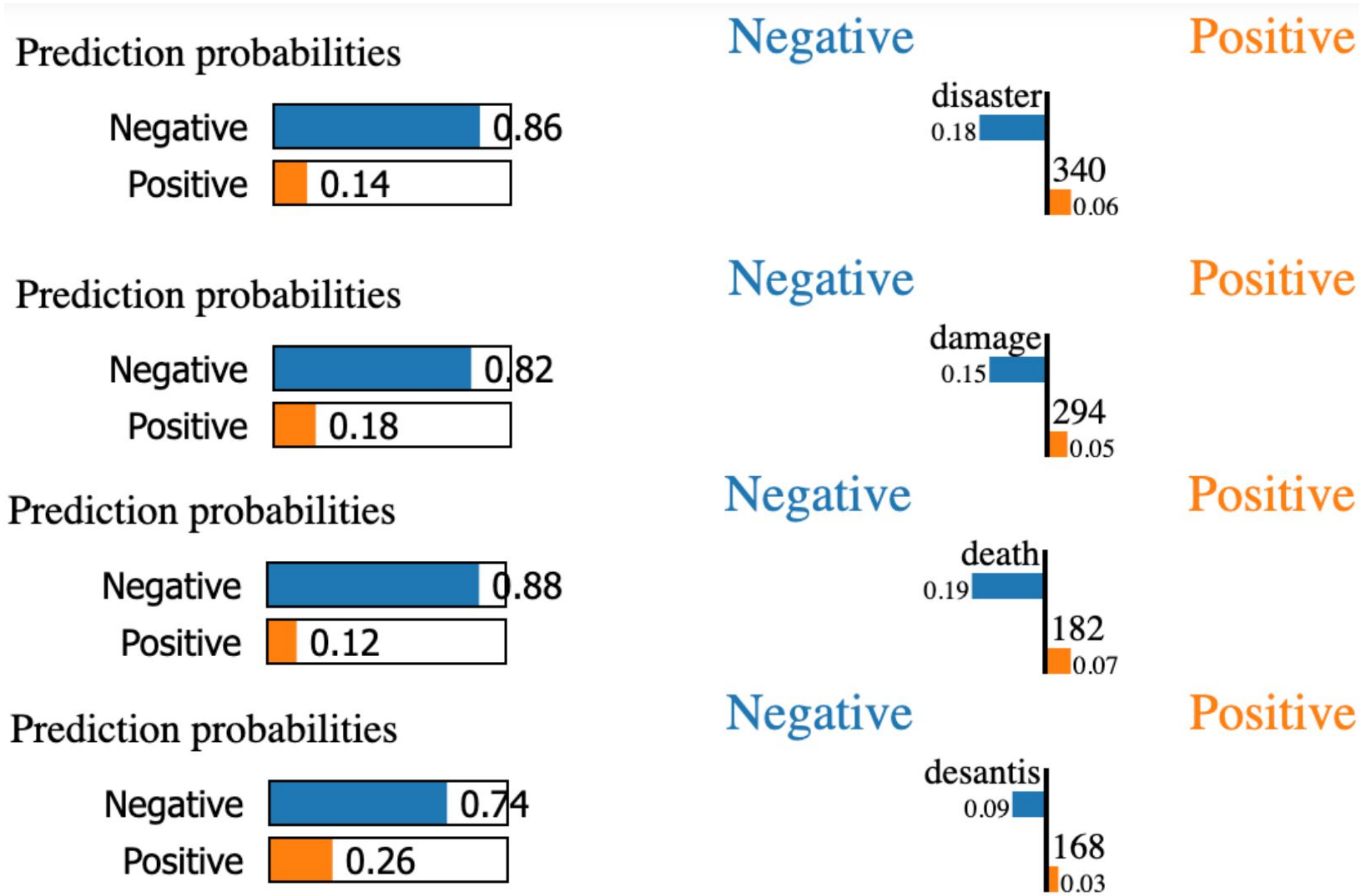
All models could not produce precision scores better than 81% currently, implying hitting a threshold for improvement given current labeling techniques, and dataset. More in-depth label-making and larger data set needed for improved performance. Current model is NOT viable for Hurri_Help yet! It would respond to too large an amount of twitter users who are upset about the politics surrounding the hurricane response who are not in need of outreach.

RECOMMENDATIONS to Hurricane Response:

- Huge financial burden of recovery is common theme in negative tweets, more outreach by FEMA to inform public about financial options and disaster relief

- Look out for tweets with 3 Ds of DOOM: "Disaster", "Damage", and "Death" as these were the most common words in the negative tweets with the most negative sentiment

(AN INTERESTING FIND: DESANTIS 4th on LIST. I wonder how Gov. DeSantis would respond to this information.)



**FUTURE WORK: **

- Try EMOTION DETECTION algorithm to isolate 'SAD' tweets for labeling, and have labels as "TARGET" and "Not TARGET" for decernment
- Collect and utilize larger data set
- Analyze and model other features in data set
- Make pipeline for weeding out tweets and modeling tweets

- Get a prototype of Hurri_Help working live on twitter
- Remove names and places from data so HurriHelp will be scalable for future hurricanes

** MOST IMPORTATNLY: FIND A NEW HOME **

```
├──README.md
├──analysis_and_modeling_notebook.ipynb                    *** MAIN NOTEBOOK
├──HurriHelp.pdf                                          *** PRESENTATION SLIDES
├──images
   ├──fort_meyers.jpg
   ├──nasa.jpg
   ├──flooded_highway.jpg
   ├──tornado_damage.jpg
   ├──fema.jpg
   ├──ian.png
   ├──ian.gif
   ├─-neg_word_cloud.png
   ├──negative_bigram.png
   ├──posi_word_cloud.png
   ├──posi_bigram.png
   ├──Hurri_Help_Logo.gif
   ├──Hurri_Help_Logo.png
   ├──example_tweet.png
   ├──example_response.png
├──data_sets
   ├──tweets_douplicates_removed.csv.gz
   ├──ready_for_anlysis.csv
├──requirements.txt
├──hurri_help_outreach_algorithm.sav
├──non_main_notebooks
   ├──get_labels.ipynb                                    *** MAKING THE TARGET
   ├──get_tweets.ipynb                                    *** SCRAPER NOTEBOOK
   ├──trial_notebook(with_emojis).ipynb
├──pdfs
   ├──presentation.pdf
```

```
├──notebook.pdf
├──github.pdf
```