UNIVERSITY OF SOUTH FLORIDA

DATA SCIENCE PROGRAMMING

Project On

DISASTER TWEET ANALYSIS

Group Members:

ADITI KOCHAR
ANIRUDH GOVINDARAJ JAICHANDRAN
ARPIT GUPTA

Table of Contents

Introduction

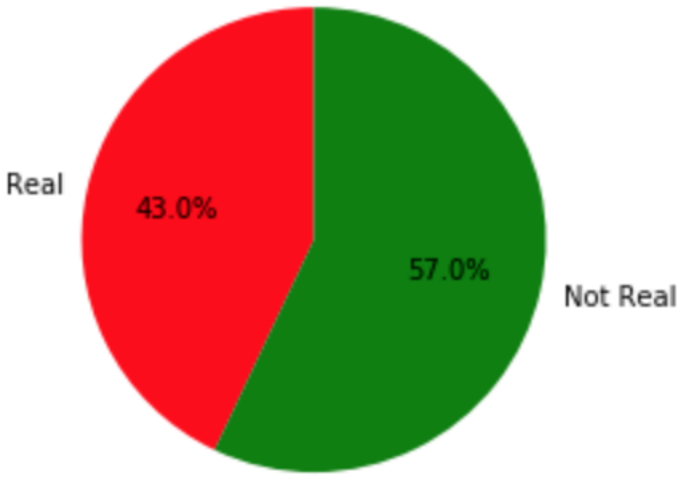
Today, twitter has become one of the most important platforms for communication. The use of smartphones and twitter enables people to announce any kind of emergency that they are facing. But it is not always clear whether a person's words are announcing a disaster or not. The goal of the project is to identify which tweets are "actual" disasters and which ones are not. This project focuses on building machine learning classification models to predict which tweets are about disasters and which ones are not.

Dataset Overview

The inspiration and source of dataset is Kaggle. The dataset has 7613 rows and 5 columns as below:

Columns

1. id - a unique identifier for each tweet
2. text - the text of the tweet
3. location - the location the tweet was sent from (may be blank)
4. keyword - a keyword from the tweet (may be blank)
5. target - in train.csv only, this denotes whether a tweet is about a real disaster (1) or not (0)



Real vs Not Real Disasters

Problem Statement

Twitter plays a vital role in times of emergency. There are millions of tweets sent every single day. This plethora of data available can be leveraged in helping communities in responding to an emergency. The presence of smartphones enables twitter user to announce a disaster they're observing in real-time such as earthquake, terrorist incident. Because of this, more agencies like disaster relief organizations are interested to programmatically monitor Twitter.

But, it's not always clear whether a person's words are announcing a disaster. It might be clear to a human right away but less clear to machine. The objective is to build a machine learning model that predicts which Tweets are about real disasters and which one's aren't.

Disaster tweets would look like:

Example of disaster tweets

```
In [8]: df_train[df_train['target']== 1]['text'][10:15]

Out[8]: 10          Three people died from the heat wave so far
        11    Haha South Tampa is getting flooded hah- WAIT ...
        12    #raining #flooding #Florida #TampaBay #Tampa 1...
        13              #Flood in Bago Myanmar #We arrived Bago
        14    Damage to school bus on 80 in multi car crash ...
        Name: text, dtype: object
```

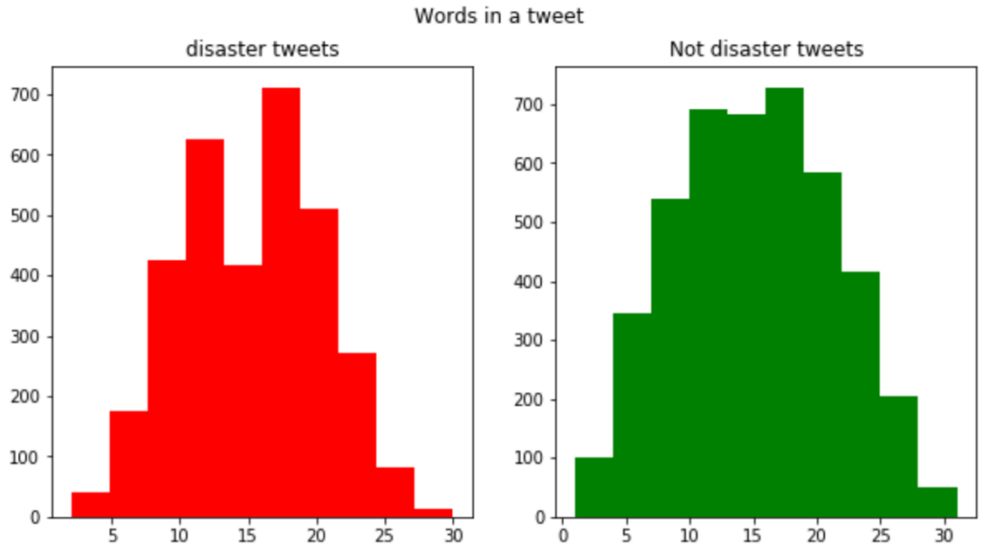
Non-disaster tweets would look like:

Example of non disaster tweets

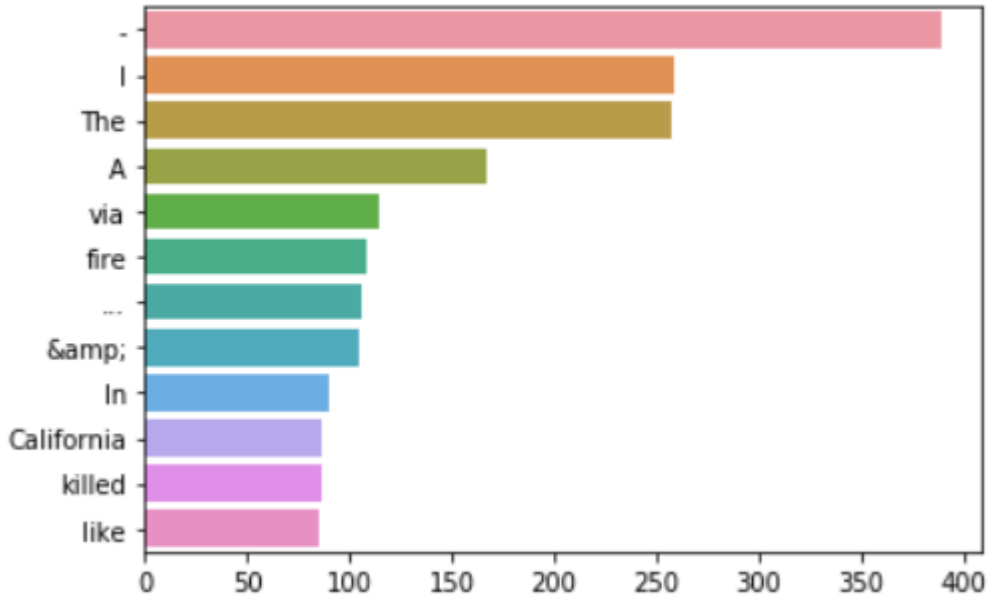
```
[9]: df_train[df_train['target']== 0]['text'][10:15]

:[9]: 25    No way...I can't eat that shit
      26              Was in NYC last week!
      27                 Love my girlfriend
      28                          Cooool :)
      29                 Do you like pasta?
      Name: text, dtype: object
```

Exploratory Data Analysis

1. No. of Words in Tweets



2. Most Frequent Words in Tweets



Data Cleaning

- **Converting all text to lower case:** All text is converted into lower case.
- **Removing noise:** This involves removing all punctuations, numbers, URL, tags, etc
- **Tokenization:** This involves converting text strings to a list of tokens(words).
- **Removing stop words:** This involves removing all the stop words like a, the, is, was, and, an, etc
- **Stemming:** This involves reducing a derived word to a base word.
- **Lemmatization:** Grouping together different forms of a word so that they can be analyzed as a single item

TF-IDF (Term Frequency-Inverse Document Frequency)

TF_IDF measures the importance of the term to a document in the document collection. TF normalizes the count within a document.

TF= Count of a term/doc term frequency

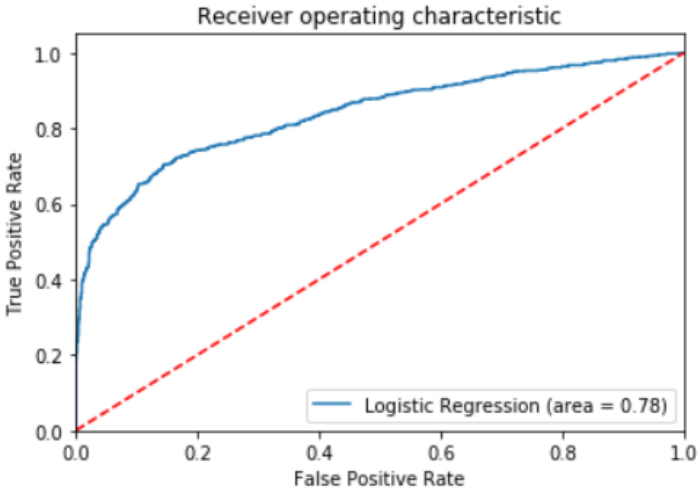
IDF = log (Total no. doc/ Total no. of doc with this term in it)

Data Modelling

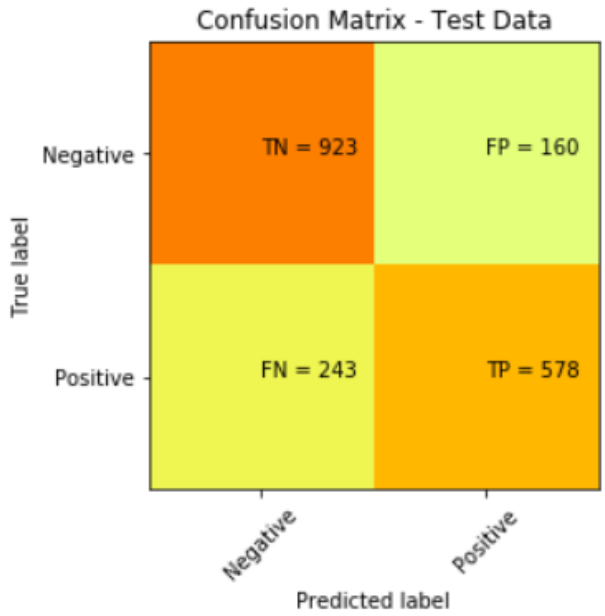
We carried out our analysis by looking at five different models:

Model	Accuracy	F1 Score
K Nearest Neighbour (KNN)	75%	69%
XGBoost		70%
Naïve Bayes	74%	66%
SVC	61%	57%
Logistic Regression	79%	74%

Logistic Regression gave the best outcome:



We evaluated the Logistic Classifier with the confusion matrix.
Below visual of confusion matrix shows that we have 160+243 misclassified tweets:



Observations, Roadblocks and Suggested Solutions

- Visualizing distribution of tweets with respect to their characteristics aided in analysing the data
- Human error like spelling mistakes can be corrected by using python library “pyspellchecker”
- Word embeddings like word2vec and glove could be used to improve accuracy
- F1 Score can be improved by using Deep Learning models
- Text cleaning is imperative to achieve better results
- Ensembles could beat the previous accuracy score of Logistic Regression
- Neural Net classifiers improves F1 score

Summary

Twitter can provide a deeper insight into real-time update of an event – disastrous or non-disastrous. Building models on Live Twitter feed can be a viable component to predict the perception of any tweet. Most of the models that achieved better result generally takes more time, and the ones that were very fast had rather poor performance. Logistic Regression with Bag-of-words achieved 0.79 accuracy in no time.

Future Prospects and Recommendations

- Use RNN, LSTM classifier for prediction as they are more established with NLP tasks
- Make predictions by fetching live tweets using Twitter API
- Use other social networking sites like Facebook, Instagram to increase the database
- Parse the raw data to make it structured
- Build a model to predict disaster trends
- Sentiment analysis of users posting the tweets
- Trying hands on transformers or BERT could increase the score
- Random Forest with Grid Search CV might give better results