Group 4 IS 665 Final Project report

Tanvi Ravindra Malali
Shivani Bhikadia
Sri Shantan Palwayi

Disaster tweet prediction

OVERVIEW

We use NLTK and Sentiment Analysis to predict whether a tweet is about a natural disaster or not, implementing the Multinomial Naive Bayes model.

NLTK

We use a library known as Natural Language ToolKit (NLTK) to perform analysis on our data, to check if a tweet is talking about a natural disaster.

Sentiment Analysis

Sentiment analysis is the contextual mining of text to extract subjective clues from source data, to make an assumption of the subject or emotion of the person who wrote it.

Imports and Data Collection

- We import the libraries necessary for initial data collection and preprocessing, which include the NLTK libraries.
- We need numpy, seaborn and matplotlib for exploratory data analysis and plotting graphs of particular columns.
- We then read the input csv data file using pandas.

Text Preprocessing Libraries

- **NLTK.tokenize:** The NLTK tokenizer divides strings into lists of sub-strings. For example, tokenizers can be used to find the words and punctuations in a string.
- **NLTK.corpus:** The modules in this package provide functions that can be used to read corpus files in a variety of formats. These functions can be used to read both

the corpus files that are distributed in the NLTK corpus package, and the corpus files that are part of external corpora.

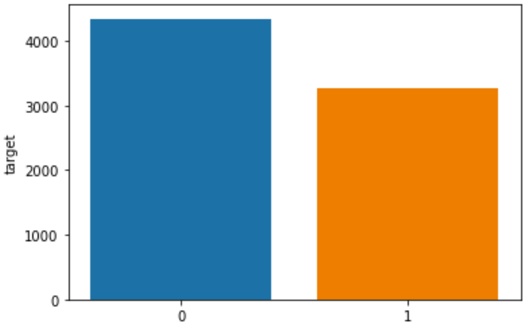
- **NLTK.stem:** Interfaces used to remove morphological affixes from words, leaving only the word stem. Stemming algorithms aim to remove those affixes required for eg. grammatical role, tense, derivational morphology, leaving only the stem of the word.
- Since the number of NaN values in columns ‘keywords’ and ‘location’ are high, we dropped these columns for ease of prediction.

Model Building Libraries

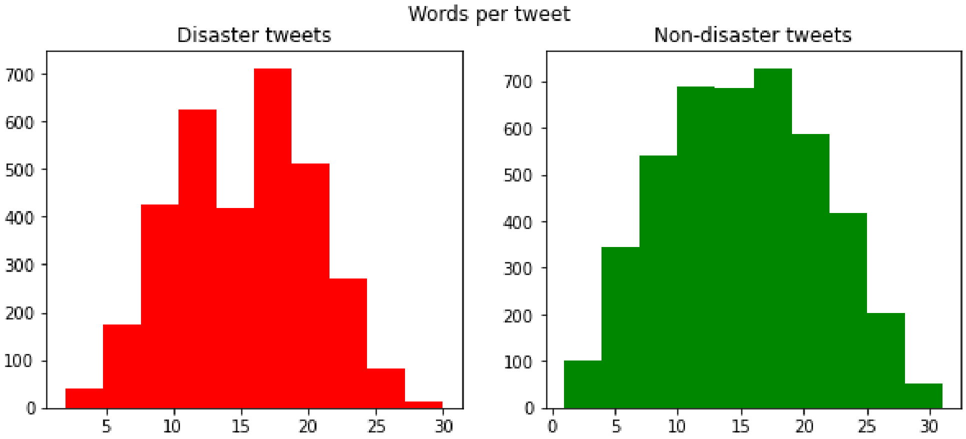
- **Sklearn.model_selection:** Split arrays or matrices into random train and test subsets using the parameter shuffle, and the percentage of the split can be specified by the parameter test_size or train_size.
- **Sklearn.naive_bayes:** Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable.
- **Sklearn.metrics.roc_auc_score:** Compute area under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
- **Sklearn.feature_extraction.text:** Convert a collection of text documents to a matrix of token counts. In our project we use
    - ➔ TfidfVectorizer: Term Frequency is the ratio of the count of a word present in a sentence, to the length of the sentence. The IDF of each word is the log of the ratio of the total number of rows to the number of rows in which that word is present.
    - ➔ CountVectorizer: In order to use the Tfidf transformer you will first have to create a count vectorizer to count the number of words, limit your vocabulary size, apply stop words, etc.
- **Gensim.models:** This module implements the word2vec family of algorithms, using highly optimized C routines, data streaming and Pythonic interfaces.

Data Visualization:

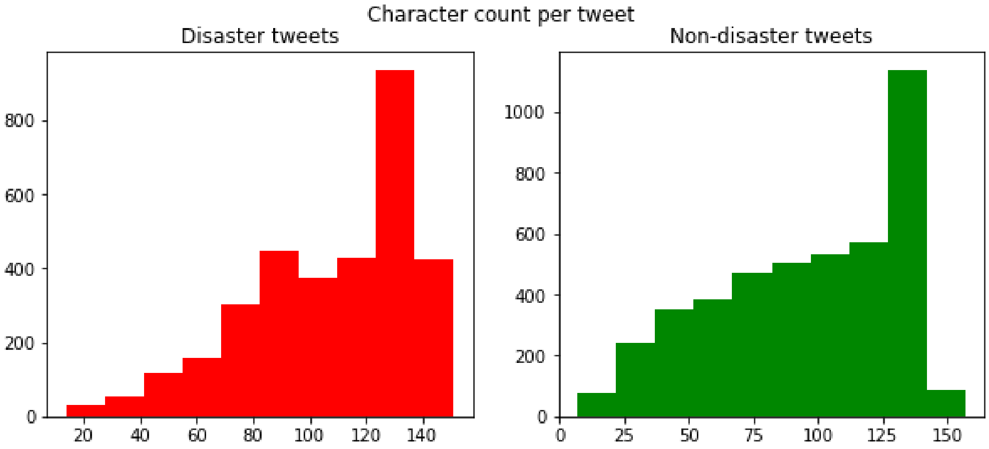
- We first check the distribution of the target variable, target.



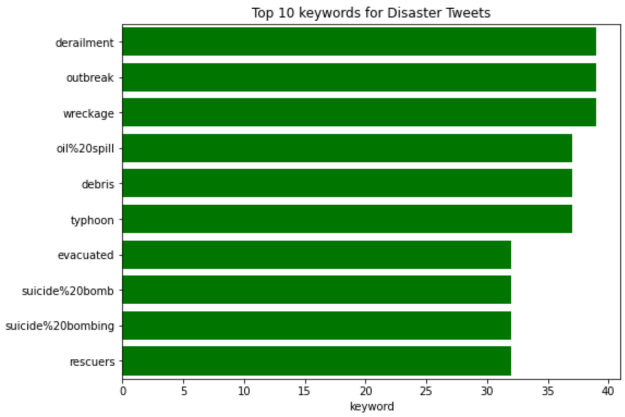
- We check the distribution of words per tweet in disaster tweets and non-disaster tweets.

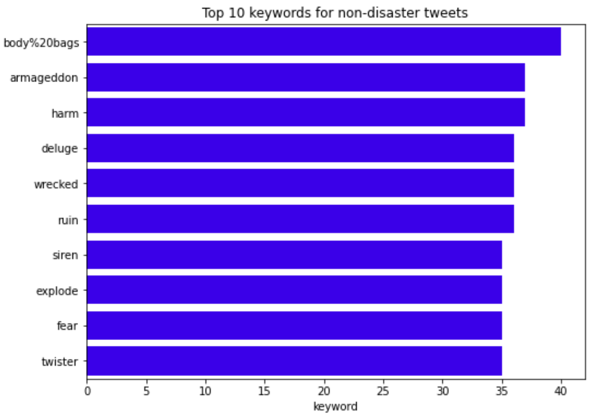


- We check character count per tweet in disaster tweets and non-disaster tweets.

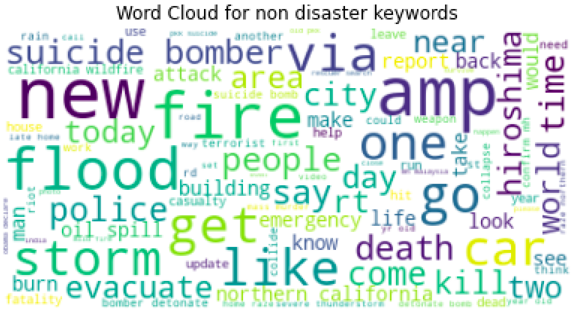


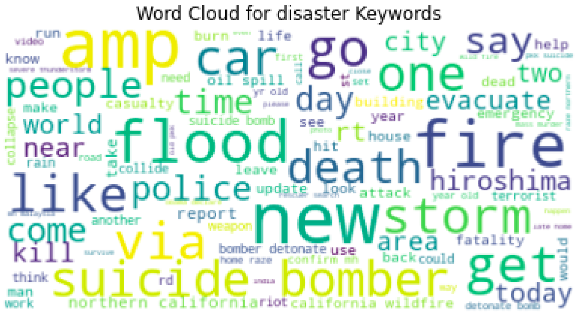
- We check the top ten words for disaster and non-disaster tweets.



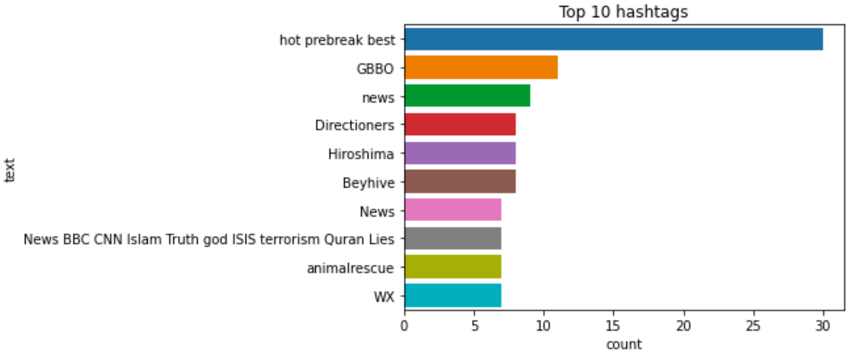


- Word clouds for disaster and non-disaster tweets are generated.





- We generate the top ten hashtags.



Text Preprocessing:

We now begin to process the input train data.

- All the text is converted to lowercase;
- Leading and trailing whitespace is removed;

- Html tags and markups are removed;
- Punctuations are replaced with space;
- Extra space and tabs are removed;
- Integers are removed;

Further Processing and Data Analysis:

- Tokenization: It refers to dividing the text into a sequence of words or sentences.
- Stemming: Stemming algorithms work by cutting off the end or beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word.
- Lemmatization: It refers to doing things properly with the use of vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only, and to return the base or dictionary form of a word, which is known as the lemma.
- Bag of Words(BoW): It refers to the representation of text which describes the presence of words within the text data i.e., two similar text fields will contain similar words, and will be a ‘bag’ of words.

Modelling:

- In our project, we choose Multinomial Naive Bayes, a popular model for use in text mining prediction.
- We chose this model as it is suitable for classification with discrete features, i.e., word counts for text classification, is easy to implement and can handle large datasets.
- We use this model in tandem with TFIDF (fractional counts).
- We use the kfold split and train test split method from sklearn to randomly split the training data using the parameter shuffle set to True, to generate a classification report and a confusion matrix.
- We split the data and perform prediction ten times using a for loop, and take the roc_auc score from each iteration and take the average of each score to evaluate how accurate our model is.
- An average accuracy score of 85.8% to is obtained, as a final result.