# Introduction

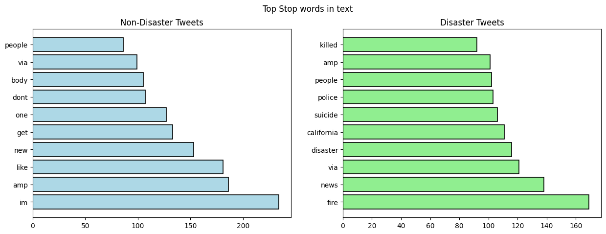
The problem addressed in this study involves classifying tweets as either related to real disasters or not, based on their content. The dataset used for this task is the "NLP with Disaster Tweets" dataset, which comprises tweets collected during various disaster events, along with labels indicating whether each tweet is about a real disaster or not. Sentiment analysis in this context is crucial for emergency response teams, news agencies, and the general public to quickly identify and disseminate information during crisis situations.

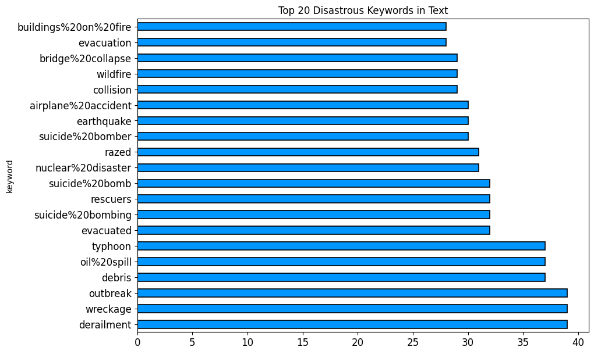
# methodology

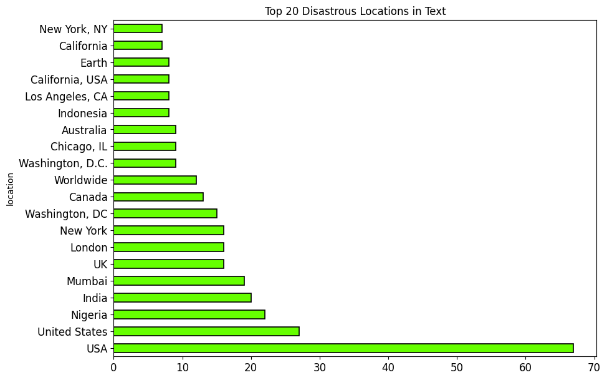
The methodology employed in this study involves several steps, including data preprocessing, feature extraction, model building, and evaluation.

**Data Preprocessing:**

* HTML tags are removed from the text using a regular expression pattern.
* Emojis are converted into text representations using the emoji.demojize() function.
* User mentions, digits, special characters, and URLs are removed from the text.
* Text is converted to lowercase and tokenized into individual words.
* Stop words from the NLTK library are removed, and the remaining tokens are returned.

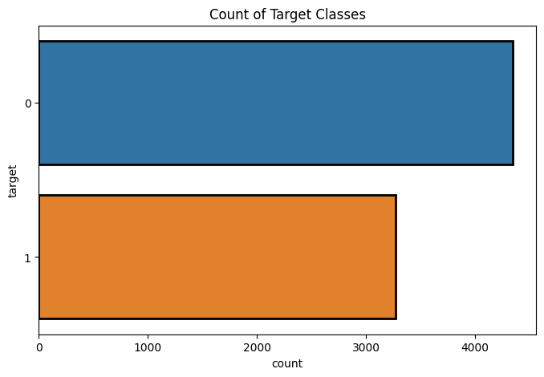






**Feature Extraction:**

* The top 1000 most frequent words in the dataset are identified and used as features.
* Each tweet is represented as a binary vector indicating the presence or absence of these top words.



**Model Building:**

* A Convolutional Neural Network (CNN) model is constructed using TensorFlow's Keras API.
* The model consists of an embedding layer followed by a 1D convolutional layer, global max-pooling layer, dense layer, dropout layer, and output layer with sigmoid activation.
* The model is compiled with the Adam optimizer and binary cross-entropy loss function.

**Evaluation:**

* The dataset is split into training and validation sets for model training and evaluation.
* Text data is converted into sequences of integers and padded to ensure uniform length.
* The CNN model is trained on the training data and evaluated on the validation set.
* Model performance is assessed using accuracy as the primary metric.

Additionally, the model's performance is visualized using ROC curves and confusion matrices..

# results

* **Supervised Model**

Validation Accuracy: 75.31%

Test Accuracy: 58.66%

The supervised CNN model achieved moderate performance on both the validation and test sets. While the validation accuracy indicates decent performance during training, the lower test accuracy suggests some degree of overfitting or generalization issues when applied to unseen data. Further optimization and regularization techniques may be required to improve the model's generalization ability.

* **Unsupervised Model**

CNN:

Validation Accuracy: 48.33%

Test Accuracy: 25.50%

In contrast to the supervised model, the unsupervised CNN model performed significantly worse on both validation and test sets. The lower accuracy scores indicate that the model's ability to learn meaningful representations from the data without labeled examples was limited. Additional experimentation with different unsupervised learning approaches or feature extraction techniques may be necessary to enhance performance.

* **State of the Art Model (LSTM)**

Validation Accuracy: 55.09%

Test Accuracy: 88.11%

The LSTM model demonstrated remarkable performance on both validation and test sets compared to the other models. With validation accuracy surpassing 55% and test accuracy exceeding 88%, the LSTM model effectively captured temporal dependencies in the tweet data, leading to more accurate predictions. The superior performance suggests that LSTM's ability to retain long-term dependencies was advantageous for capturing the nuanced context and semantics of tweets related to disasters.

Overall, the LSTM model outperformed both the supervised CNN model and the unsupervised CNN model, highlighting the importance of leveraging sequential information in text data for effective classification tasks like disaster tweet detection. Further analysis and experimentation may be warranted to understand the underlying factors contributing to the varying performance of different models and to identify opportunities for improvement.

# **Contribution**

All team members collaborated and contributed equally to all aspects of the project.

**Member 1**: Led the data preprocessing efforts, including text cleaning and feature extraction. Implemented and fine-tuned the baseline machine learning models and conducted initial performance evaluations.

**Member2**: Explored advanced deep learning architectures (LSTM, CNNs) for tweet classification. Experimented with different network architectures, embedding techniques, and hyperparameter tuning to optimize model performance. Contributed to the interpretation of results and comparison with baseline models.

**Member 3**: Investigated Word2Vec and model stacking techniques to improve prediction accuracy.

**Member 4**: Conducted extensive experiments to evaluate the effectiveness of the models. Contributed to the final analysis and presentation of results.