An analysis on Disaster Management Tweets and their Relevancy

BIA 660 - A: Web Mining

Group 3

Chinmay Bhagwat (20015512)

Prachi Jacob (20011215)

Manoj Patel (20012221)

I. Introduction:

The world has always been evolving so rapidly, and it is almost impossible to know everything that is happening at the moment, or has happened up to this moment. But thanks to smart technology, anyone with a device and an internet connection can gain access to a piece of information at a point in time. Social media, over the past couple of decades, has been serving as one of the primary sources of information alongside news channels and broadcasts. This trend has only seen an upward spike with time as more and more population from the younger generation keeps gaining access to devices.

Since its inception in 1996, social media has managed to infiltrate half of the 7.7 billion people in the world. Social network platforms almost tripled their total user base in the last decade, from 970 million in 2010 to the number passing 4.48 billion users in July 2021. The growth in the number of accounts per person is up 75% from 4.8 accounts per person in 2014 to 8.4 in 2020. It should hence come as no surprise that the amount of social media websites has immensely increased over the years. It is very difficult to be present on every platform, and some websites have stood the test of time to emerge as leading sites where most of the population interacts and gets their information from, Twitter being one of them.

Twitter is currently the 6th most popular social media platform, and it is also one of the websites where people go to receive more serious updates; it has become an important communication channel in times of crisis. It has served well in bringing news during national emergencies, international happenings, natural disasters, and a lot more. It has been very helpful in alerting people of disasters like earthquakes, and it definitely plays an important role in disaster management. However, the website does run on algorithms that look for certain keywords. A good percentage of the young generation uses words casually, which can be misinterpreted by these algorithms and it leads to confusion and even false alarms.

It is difficult to know whether we are getting real information or not. In this project, we will be working on recognizing and separating the real news data from the non relevant data. We will also spot trends in how these words have been used over the years, how they impact information channels, and what words actually seem to generate real information. Through the use of Machine Learning models, we will accomplish this distinction.

The data has been scraped from X(Twitter); the keywords used to extract the tweets have been taken from an already existing trained dataset. The new dataset will be pre-processed to remove any abnormalities and to make sure that the data is consistent. We will spot patterns through EDA and implement Machine Learning algorithms to achieve our objectives.

II. Literature review:

We have reviewed quite a few papers on this topic: *Using Machine Learning in Disaster Tweets Classification* by Humaid Alhammadi;

The project aimed to develop a model for classifying text as either referring to a real disaster or not, with the goal of aiding in the identification of genuine disasters and filtering out fake ones. The literature review explored text analysis and classification techniques. The project followed the CRISP-DM methodology, addressing business understanding, data exploration, pre-processing, modeling, and evaluation stages. Four supervised machine learning models (KNN, SVM, XGBoost, Naïve Bayes) were built, with varying accuracies (99%, 80%, 78%, 65%). Overfitting was identified in the first model, and the last one performed poorly. SVM excelled in identifying true positives, while XGBoost was better at true negatives. Model performance was attributed to the method of building a term-document matrix (TDM) without considering sentence context.

III. Research Question:

In the context of Twitter's crucial role in disseminating disaster-related information, the project aims to enhance the accuracy of alerts by distinguishing real data from casual language often misinterpreted by the platform's algorithms. Leveraging Machine Learning and data from Twitter, the objective is to improve disaster management through the identification of genuine information and trends in keyword usage.

IV. Data:

IV.I Data Collection:

There are two parts to our data collection - one is the training dataset and the other is the testing dataset. For the training model, we took the dataset from Kaggle labeled “NLP Disaster Tweets”. The model contains the following columns: ID, keyword, location, text, and target. ID represents the unique ID generated for each row. The keyword is what the algorithm looked for and it is the factor on which the machine decided that the particular tweet was related to disaster management. The location represents the location specified in the tweet. The text is the main part of the dataset, which shows the actual tweet. Finally, the target is a set of binary values, which tells us whether the tweet is related to disaster management (1) or its irrelevant (0).

For the actual dataset, we first manually scraped the first few pages of Twitter to look at the relevancy of the tweets that show up relating to a specific keyword. From the training dataset, we filtered out the keywords and deleted any duplicates, which gave us our 100 keywords. We

used these keywords in the "***Advanced Search"*** section to generate the dataset containing information regarding 500 tweets using the advanced filter ***"Any of these words"***. We have not considered the location parameter, which will be discussed more briefly in the next section.

	ID	Keyword	Description
0	1	battle	Two of this generation's greatest. Nothing but...
1	2	accident	educational all shot favorite accident regiona...
2	3	BATTLE	OMG IS FOX MC. CLOUD IN SUNDAY SMASHING BATTLE...
3	4	damage	In lockstep, the worldwide media is suddenly b...
4	5	burning	November 2007, Obama is down 27 points in the ...
5	6	destroy	Klaus Schwab is a #Zionist. He claims to be Je...
6	7	emergency	Iceland declares state of emergency amid volca...
7	8	destroy	12 hours to Nasrallah's second speech, and Hez...
8	9	Bomb	HALFTIME! HUGE 32 yard Bomb by Deante Ruffin t...
9	10	Bomb	Anila getting hit by: Ladiva's Maximum Love Bo...

IV.II Preprocessing:

After taking a look at the training dataset, we realized that we would have to process the location column a lot in order to clean it. This was mainly owing to the fact that these locations have not been detailed in a specific format. There are some tweets that only have the city/state/country name. There were some tweets who have the full address in the location column, whereas some have none. On the other hand, there were tweets that had a joke or something funny in place of location. Hence, there was no particular format in which we could convert them all. Moreover, the manual scraping revealed tweets that had no location listed. In order to solve this problem, we determined that we would have to generate a dictionary with all possible locations. However, this is just not possible as we would have to list the name of all continents, countries, states, cities, towns, and a combination of two/three/more of these attributes - this would result in millions of combinations and we do not have the physical or computing power for it. Due to this reason, we decided to drop the location column.

The other columns such as ID and keywords did not need to be changed as ID is uniquely generated for each tweet, and the keyword is something that is corresponding to that specific tweet. The text contains the tweet description which will then be tokenized to facilitate further analysis. Characters in the text which are punctuation marks are cleaned out during tokenization to ensure consistency and improve text data quality. Capitalized letters are converted to regular

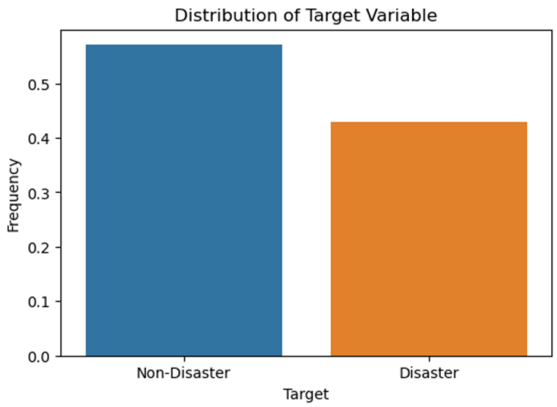
letters so that “California” and “california” are not considered two separate entities. Stop words (as seen in the EDA section) such as a, the, an, etc. are removed as well so that we can drive our focus on more important words.

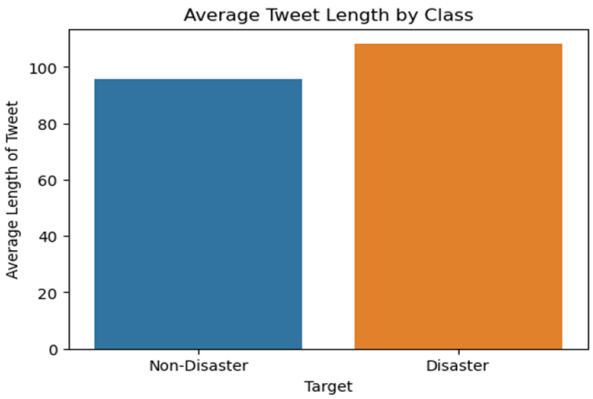
	Id	Descreption	processed_text
0	1.0	Accident cleared in #PaTurnpike on PATP EB bet...	accident cleared paturnpike patp eb cranberry ...
1	2.0	Just got to love burning your self on a damn c...	got love burning self damn curling wand swear ...
2	3.0	I hate badging shit in accident	hate badging shit accident
3	4.0	#3: Car Recorder ZeroEdgeå¨ Dual-lens Car Came...	3 car recorder car camera vehicle camcorder la...
4	5.0	Coincidence Or #Curse? Still #Unresolved Secre...	coincidence curse still unresolved secret past...
...	...	...	...
594	595.0	Rly tragedy in MP: Some live to recount horror...	rly tragedy mp live recount horror saw coach t...
595	596.0	Rly tragedy in MP: Some live to recount horror...	rly tragedy mp live recount horror saw coach t...
596	597.0	I've spent the day traumatised about the fact...	spent day traumatised fact load good music pro...
597	598.0	Can you imagine how traumatised Makoto would b...	imagine traumatised makoto would could see dub...
598	599.0	@TremendousTroye I'm so traumatised	tremendoustroye traumatised

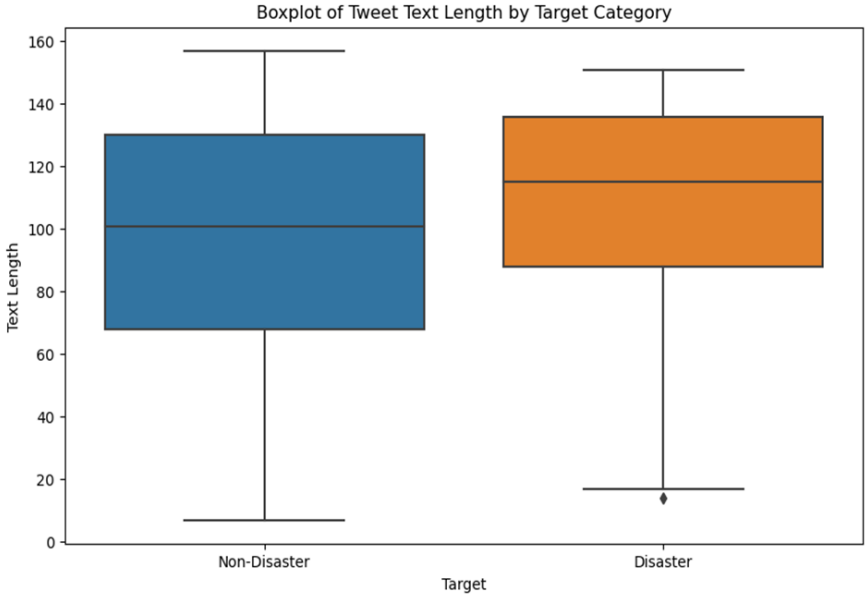
V. Exploratory Data Analysis:

The aim of this Exploratory Data Analysis (EDA) is to gain comprehensive insights into the dataset used for the Disaster Prediction Project. The EDA aims to:

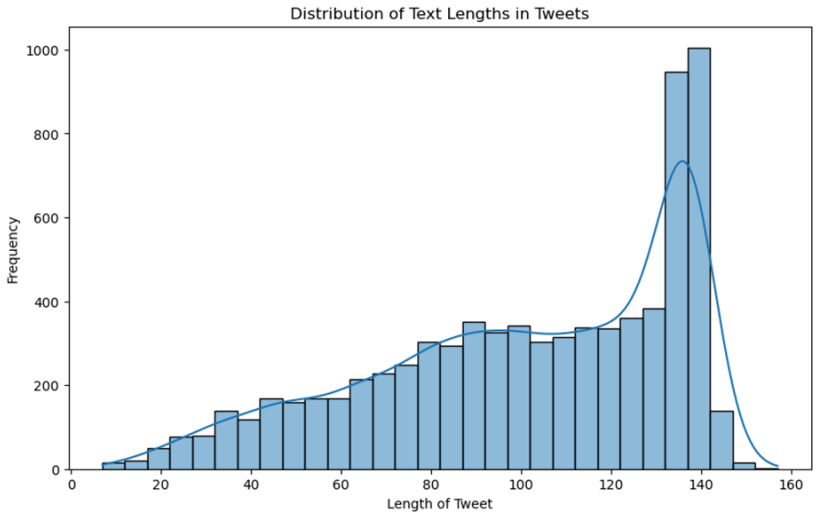
- Understand Data Structure: Examine data completeness, missing values, and feature types.
- Target Variable Analysis: Analyze the balance between disaster and non-disaster tweets.
- Keyword and Text Analysis: Investigate keywords' frequency and explore textual content for patterns.
- Geographical Insights: Study the geographical distribution of tweets, noting the presence of missing data.
- Statistical Summary and Visualization: Provide statistical summaries and employ visual tools like bar charts, histograms, and word clouds for a comprehensive understanding.
- This analysis will inform the model building process, leading to a robust disaster prediction model.

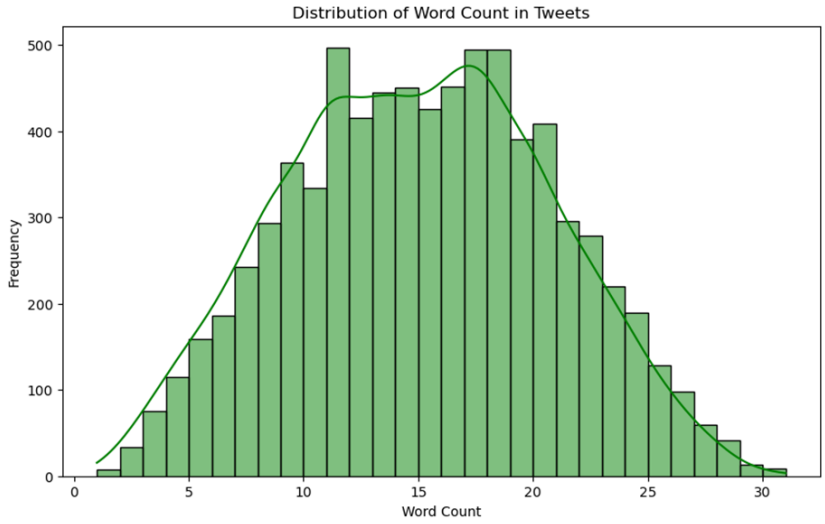




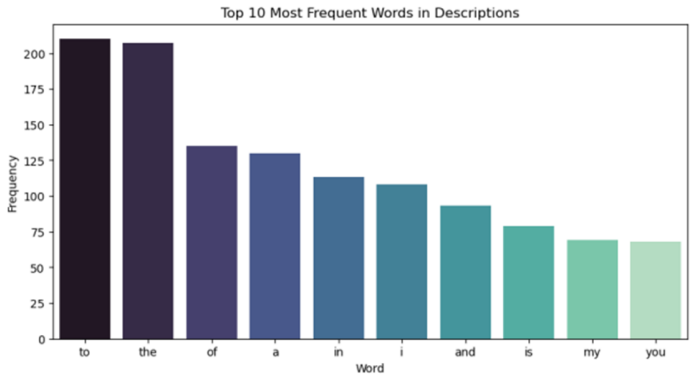


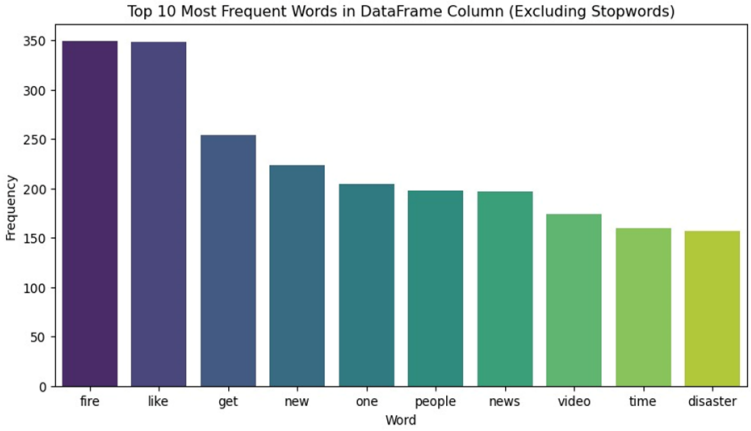
About 40% of the tweets were Disaster related and about 60% of the tweets were non-disaster related. This indicates how the algorithm takes into account all the keywords and gives out results that are sometimes not relevant at all. From the analysis of tweet length, we can see that tweets not related to disaster have an average length lesser than the tweets related to disaster. The corresponding box plot shows that Non-disaster tweets range from around 70-135 with a mean of 100, whereas the disaster related tweets range from around 90-140 with a mean greater than 110.



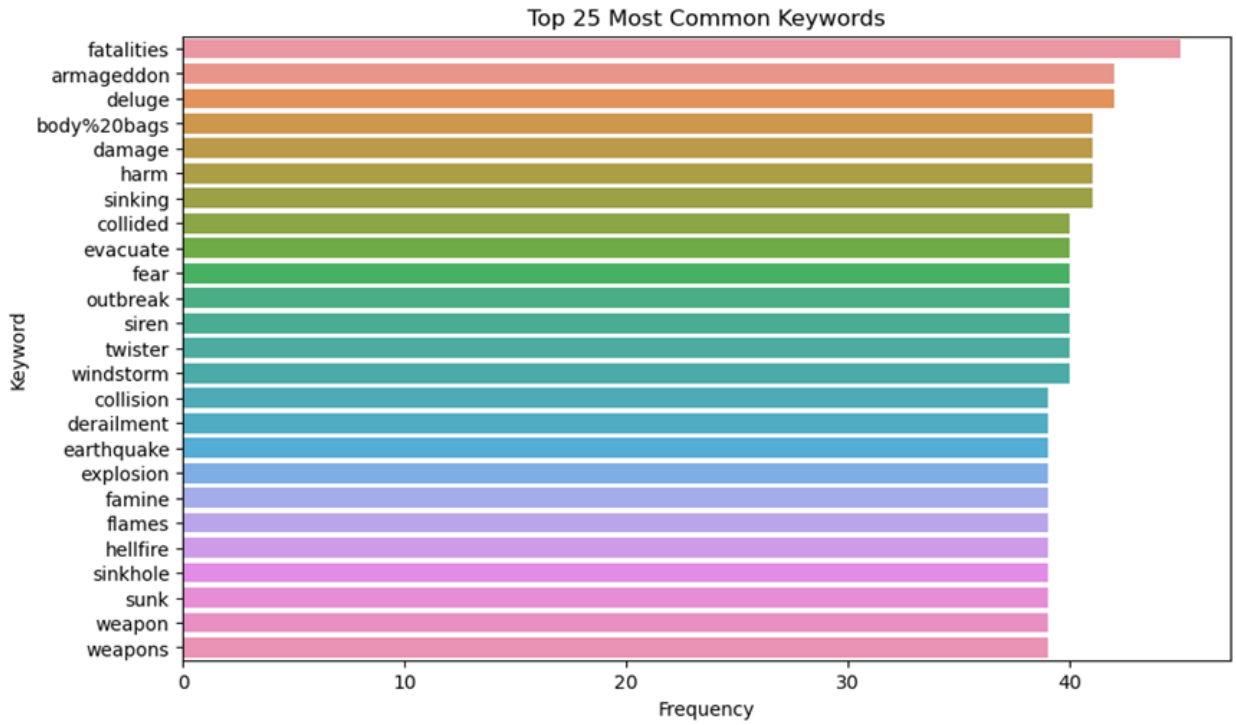


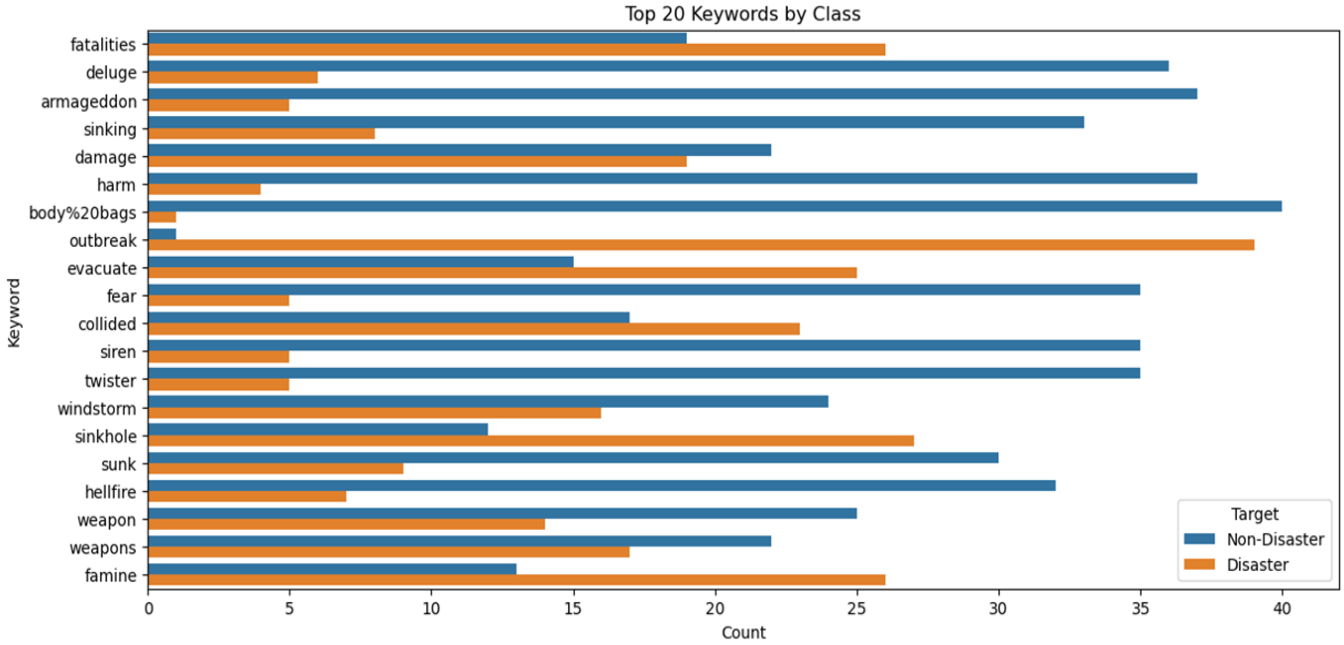
This histogram displays the distribution of the lengths of the text descriptions in the dataset. The x-axis represents the length of the descriptions (in terms of character count), and the y-axis shows how many descriptions fall into each length category (frequency). The shape of the histogram can give insights into the verbosity of the texts, indicating whether most descriptions are brief or lengthy. From these histograms, it can be derived that within all the tweets, most of them have 950-1000 characters, with most of them having a 500 word count.





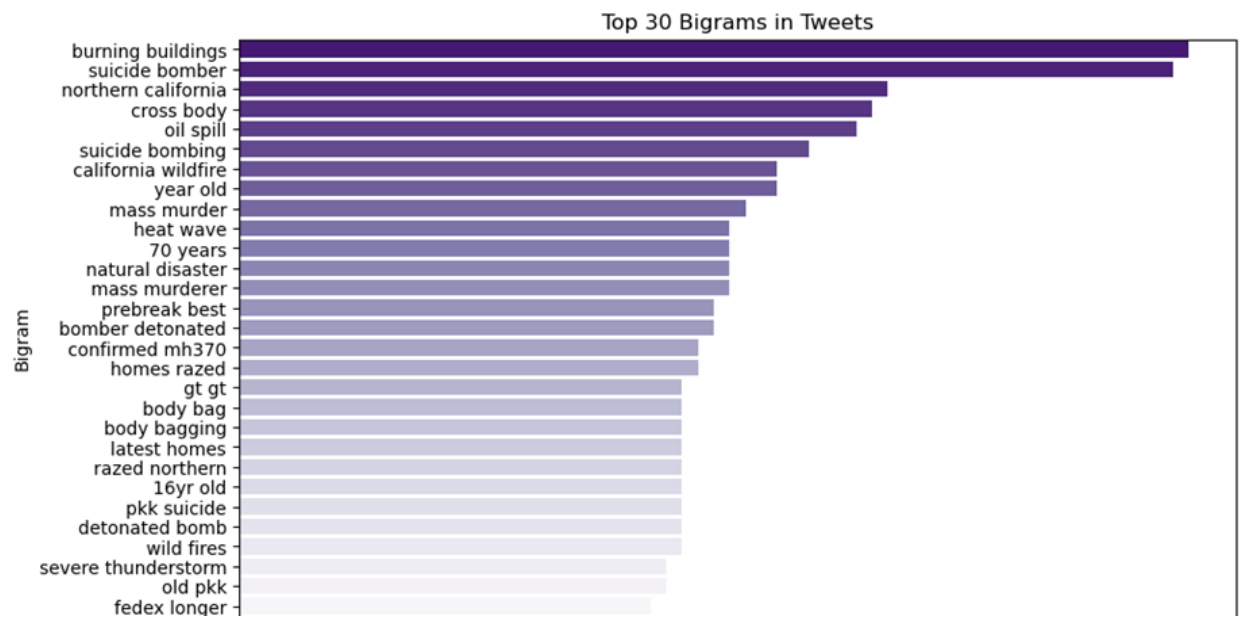
This bar chart shows the top 10 most frequently occurring words in the descriptions, determined using a simplified method of word frequency analysis. The x-axis lists the words, and the y-axis shows their respective frequencies. These are the most frequent words in the text descriptions which keep us from finding the more important ones. These stopwords include: to, the, of, a, in, I, and, is, my, and you. Once we filter them out, we can find the top 10 most frequent words: fire, like, get, new, one, people, news, video, time, and disaster. The word cloud is a visual representation of the most frequently occurring words in the 'Description' column of the dataset. Common words appear larger and more centrally placed, while less common words are smaller and positioned around the edges.

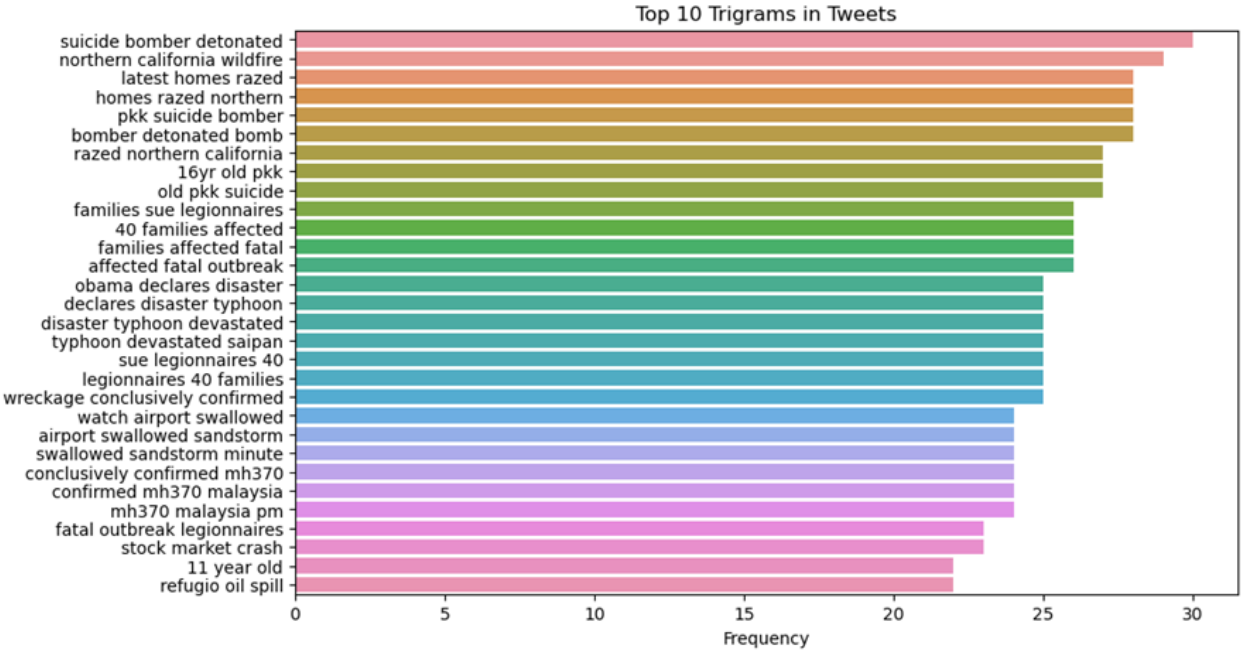




Here, we can see the top 25 most common keywords, followed by a visualization of the words by class. Words such as harm, sinking, fatalities, famine, earthquake appear which are disaster related terms, and this is proof that after tokenizing and filtering stop words out, the words that appear most frequently are disaster related. We can go deeper and see where these words are more often used. For instance, words like fatalities, outbreak, evacuate, collided, sinkhole, and famine have a high usage in disaster related tweets. This checks out as the specified terms are

ones that are not very casual and are more related to emergencies/pandemic/natural disasters. However, words such as sinking, harm, weapon, fear, etc. can be used either way and appear more in non-disaster tweets.





Bigrams identify the most common pairs of words that occur together whereas Trigrams focus on the most common sequences of three words. Bigrams and Trigrams help us make more sense of the usage of words as they are not considering them in their individual context, but

rather, a phrase of words that describe a situation. Bigrams include phrases such as: burning building, California wildfire, mass murder, bomber detonated, suicide bombing, etc. This would result in more disaster related tweets as “building”, “California”, “suicide”, etc. would generate very irrelevant tweets, but these words used in combination would provide a highly accurate result. The same goes for Trigrams, some of which are: suicide bomber detonated, northern california wildfire, affected fatal outbreak, etc. This tells us that “northern california” and “california wildfire” appearing in the Bigrams are actually referring to the same situation, which is “northern california wildfire”; similarly, “suicide bomber” and “bomber detonated” are referring to “suicide bomber detonated”.

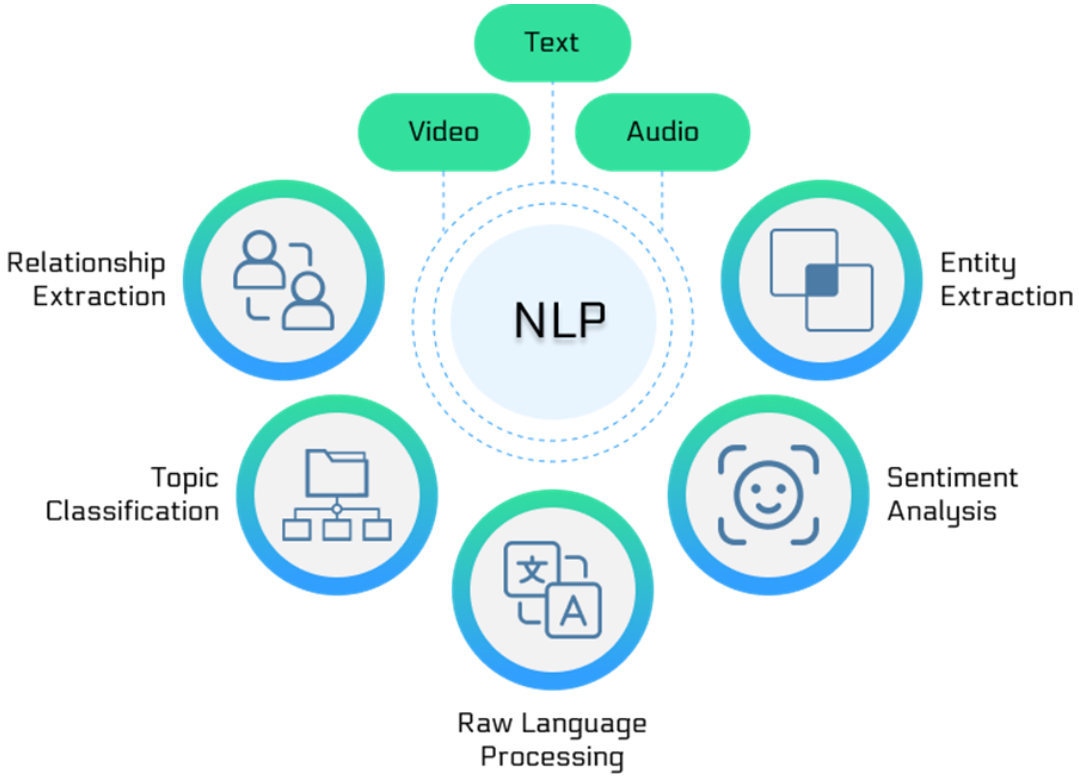
VI. Data Modeling

Model:

Natural Language Processing (NLP)

NLP is a field of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. It plays a pivotal role in tasks such as language translation, sentiment analysis, and information extraction from textual data.

- Exploring Language: NLP, or Natural Language Processing, enables machines to understand, interpret, and respond to human language effectively.
- Machine Learning Integration: NLP combines computational linguistics and machine learning to process and analyze large volumes of natural language data.
- Application Diversity: NLP is widely used in applications like language translation, sentiment analysis, and chatbots for enhanced user interaction.
- Technological Evolution: Continuous advancements in NLP are driving better context understanding and more natural human-computer interactions.
- Data Analysis Tool: NLP plays a crucial role in text mining and extracting meaningful information from unstructured data sources.



For this project, Natural Language Processing serves as a vital tool for extracting valuable insights from social media data. Its advantages lie in information extraction, sentiment analysis, and keyword recognition, aiding in the identification of relevant content during disaster scenarios.

NLP's contextual understanding ensures accurate analysis by distinguishing between casual language and emergency indicators. The use of machine learning models within NLP facilitates the classification of tweets, distinguishing between relevant and non-relevant information. Additionally, NLP techniques contribute to data cleaning and exploratory data analysis, ensuring the reliability of our findings.

The adaptability of NLP to evolving language trends makes it a suitable choice for handling the diverse and extensive data generated on social media platforms. Its real-time analysis capabilities are crucial for timely insights during disaster events. By integrating NLP, our project aims to enhance decision support systems, enabling authorities to focus on pertinent information for effective disaster management.

Implementation:

To implement NLP in the project we specifically focused on content that could be associated with disaster management tweets. The preprocessing steps involve tokenization and converting processed text into integer sequences, optimizing it for subsequent analysis.

The model architecture comprises an embedding layer to capture semantic relationships in the text, followed by flattened layers for feature extraction. The inclusion of dense layers with ReLU activation functions, along with dropout regularization, enhances the model's ability to generalize and avoid overfitting. The final layer utilizes a sigmoid activation function, enabling binary classification, which aligns with the nature of the task—possibly distinguishing relevant from non-relevant tweets in the context of disaster management.

The compiled model employs the Adam optimizer with a specified learning rate and binary cross-entropy loss function to measure the dissimilarity between predicted and actual classifications. Training is executed over multiple epochs, utilizing a batch size of 500 and incorporating a validation split of 20% to assess the model's generalization performance.

This implementation serves as a foundational component for employing NLP techniques to classify and analyze textual information, with potential implications in disaster management scenarios. The model aims to discern patterns and relevance within the textual data, contributing to the broader goal of extracting actionable insights from social media content during crisis situations.

```
X = df['processed_text']
y = df['target']

# Tokenize the text
tokenizer = tf.keras.preprocessing.text.Tokenizer()
tokenizer.fit_on_texts(X)

# Convert text to sequences of integers
X_sequences = tokenizer.texts_to_sequences(X)


max_length = max(map(len, X_sequences))
X_padded = tf.keras.preprocessing.sequence.pad_sequences(X_sequences,
                                                         maxlen=max_length, padding='post')

model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim
                              =len(tokenizer.word_index) + 1, output_dim=128,
                                                                input_length=max_length),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(8, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

# Compile the model with a specified learning rate
learning_rate = 0.01
optimizer = Adam(learning_rate=learning_rate)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_padded, y, epochs=5, batch_size=500, validation_split=0.2)

# Evaluate the model
loss, accuracy = model.evaluate(X_padded, y)
print(f'Test accuracy: {accuracy:.4f}')
```

Testing and Result:

In the evaluation phase of the implemented model, predictions were generated using the trained NLP model on a set of Twitter data, acquired through scraping with predefined keywords. The model's performance was assessed by converting its continuous predictions into binary values, with a threshold of 0.5, indicative of relevance. The resultant binary predictions were then examined.

The model showed 93% accuracy when it trained with the data. This outcome underscores the model's ability to accurately classify textual data related to disaster management tweets, as illustrated in the corresponding image presenting the target values associated with the data. The high accuracy suggests that the model is adept at discerning relevant information from the provided dataset, highlighting its potential applicability in effectively identifying pertinent content within the realm of disaster management tweets.

```
predictions = model.predict(X_padded1)

# Convert predictions to binary values (0 or 1)
binary_predictions = [1 if pred > 0.5 else 0 for pred in predictions]

# Print the binary predictions
print(binary_predictions)
```

```
19/19 [==============================] - 2s 11ms/step
[1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0,
1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0,
0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1,
1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1]
```

VII. Conclusion

Our goal for this project was to classify the relevant data from the irrelevant Twitter posts and address the challenge of enhancing the accuracy and relevance of disaster-related tweets/alerts We were able to achieve the remarkable outcome of 93% accuracy signifies advancement in distinguishing genuine information from casual language often misinterpreted by Twitter's algorithm. Future implementations for improving disaster management by providing more precise information on social media during a crisis.

References:

- https://backlinko.com/social-media-users
- https://www.searchenginejournal.com/social-media/social-media-platforms/#close
- https://www.kaggle.com/competitions/nlp-getting-started/data