Machine Learning &
Pattern Recognization

DISTINGUISHING LITERAL AND FIGURATIVE TWEETS FOR BETTER DISASTER RESPONSE SYSTEM

Presented by Aatmika, Saumya, Akshita

Problem Statement

How Is It Related?

- Twitter’s quick informative nature is turning out to be essential during periods of crisis or moments of emergency.
- **Often Twitter has been the first media outlet to inform the public on the start of an international, national, or local predicament**.The ubiquity of smart devices and Internet of things **allows users to announce an urgent situation immediately and in real-time**.
- For this reason, more agencies, companies, and media outlets are preemptively monitoring Twitter for insight into emerging trends and events in real time.

- **The biggest problem, however, lies in the natural ambiguity of written speech removed from its context.**
- **For an automated system , it has proven incredibly difficult to determine the meaning behind a statement**, even humans struggle to identify statements as ironic or sincere when removed from their context

.

https://sci-hub.se/https:/doi.org/10.1109/CCWC51732.2021.9375916

Context: What causes this ambiguity?

FIGURATIVE/NON-LITERAL

illustrates a concept or meaning through many potential interpretations or contexts. It paints a picture that we interpret through **Simile, Metaphor, Imagery, Symbolism, Allusion, Personification**

LITERAL

relies more on the **dictionary definition of the words involved**. There is no additional image, exaggeration, or comparison invoked in how the words are used. The **meaning should be very clear and not require much additional clarification beyond that of context**.

https://www.microsoft.com/en-us/microsoft-365-life-hacks/writing/difference-between-literal-and-figurative-language

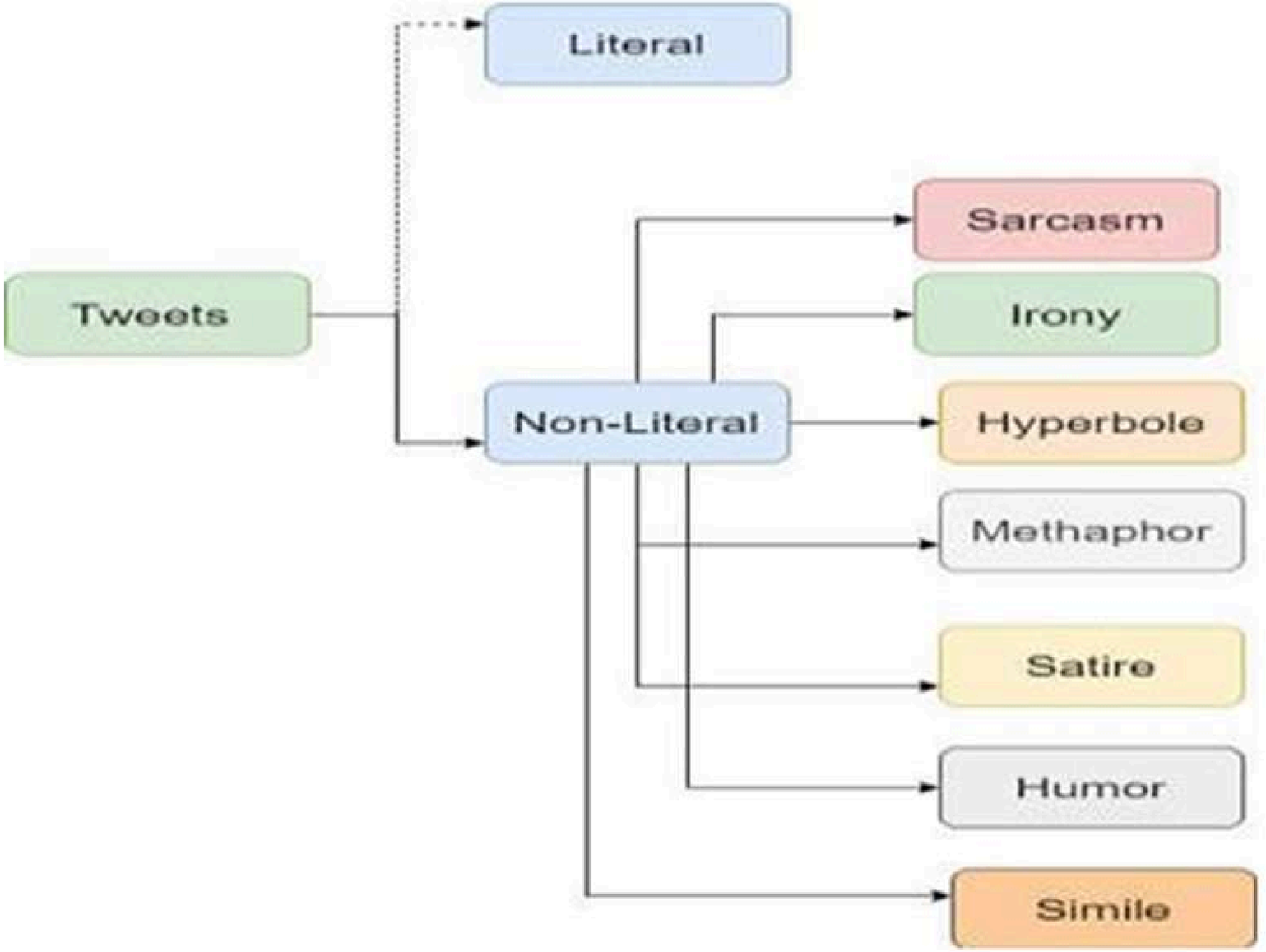


Fig. 1: Tweet Categories

Potential Applications

1.	**Enhanced Disaster Response through Accurate Tweet Classification** By distinguishing between literal and figurative language in tweets, disaster response teams can prioritize actionable information, reducing the noise from non-literal expressions.
2.	**Improved NLP Models for Figurative Language Detection** Developing machine learning models capable of detecting sarcasm, irony, and metaphors enhances the understanding of public sentiment during disasters.
3.	**Reducing False Alarms and Misinformation Spread** By detecting figurative tweets (e.g., "This exam is a disaster") and excluding them from critical alerts, your system can reduce false positives and help response teams focus on genuine emergencies.
4.	**Assisting Media & Journalists in Disaster Coverage** can be helpful for media houses who rely on such social media platforms for breaking news and are always tracking public sentiment by constant analysis of social media platforms

Potential Impacts

Development of Advanced NLP Tools
Addressing the challenges of figurative language in disaster tweets pushes the boundaries of Natural Language Processing (NLP), leading to the development of more sophisticated models capable of nuanced language understanding.

Greater Public Trust in AI-Based Disaster Detection Systems:
A sarcasm-aware system improves confidence in AI-powered crisis monitoring and decision-making tools

Enhanced Accuracy in Disaster Information Retrieval and Reduction in False Alarms and Misdirected Resources:
Ensures that critical resources like rescue teams, medical aid, and crisis hotlines are not misallocated due to sarcastic or misleading tweets.

Literature Survey

Paper 1 :Establishing the Problem

Erokhin & Komendantova, 2024; ScienceDirect Article

- This paper tells us how Social Media is Used During Disasters

“Social media provides real-time data that supports rapid response and situational awareness.”

- Used for:

“Directing emergency resources and rescue efforts”

“Monitoring public sentiment and emerging needs”

“Identifying affected populations and locations”

“Supporting risk communication and rumor control”

- Examples of Usefulness

“People report local conditions, damage, and urgent needs faster than official channels.”

“Emergency managers monitor Twitter feeds and hashtags and do keyboard based filtering to allocate field teams.”

Why is social media important during natural disaster crisis ?

Social media platforms serve as a dynamic pulse-check on community needs, concerns, and perceptions and offer continuous streams of data invaluable for disaster preparedness and response [41]. Properly analyzed, this data can provide several benefits [42]. For instance, real-time monitoring of social media allows for the immediate understanding of situations as they unfold during disasters [43]. People often turn to these platforms to report occurrences, share experiences, and seek help. Analyzing these posts [44] can provide first responders with real-time insights into where resources are most urgently needed. Sentiment analysis of posts can reveal the community's emotional state [45], which is crucial for adjusting the tone and content of public communications to maintain calm and build trust. Additionally, social media can help detect emerging trends [46] that might impact disaster management, such as sudden spikes in discussion around specific topics related to a disaster and allow responders to quickly address these areas and ensure that correct and helpful information is widely disseminated.

Limitations of existing ML models to understand the nuances of language

such large datasets, these technologies introduce their own limitations. For instance, they often struggle to interpret the nuances of language used in social media posts, such as slang or irony [36], which can lead to incorrect interpretations of public sentiment or behavior. Furthermore, the relevance of social media data can diminish rapidly as online conversations evolve.

Case study: shows how social media like twitter is an important platform during a natural disaster.

The second case study examines public engagement on social media following a major natural disaster, specifically analyzing data from Twitter after a significant earthquake [6,7,27]. Through content analysis, we explored how affected communities used social media to share real-time information, request assistance, and offer support. The study observed dramatic fluctuations in public sentiment and discussion volumes in response to specific events related to the disaster. This reactive engagement illustrates both the potential of social media as a tool for rapid communication and coordination, and the challenges in managing the spread of unverified information during such times.

potetial applications of using ML models during disaster crisis

Moreover, machine learning models have become increasingly adept at interpreting sentiments expressed across social media platforms [75]. This capability is crucial for emergency response teams to gauge public sentiment, address specific concerns effectively, and counteract misinformation to maintain public calm and trust during crises. Automated alerts generated by AI from specific triggers in social media streams, such as increases in keywords related to natural events [76], enable rapid dissemination of warnings and guidance to the public and facilitate immediate awareness and action.

Paper 2 :Figurative vs Literal Tweets

“There’s fire” ≠ “The city’s on fire” — Why language matters

(Do Dinh & Gurevych, 2016; Agrawal et al., 2020; Sosea et al., 2023)

Figurative expressions confuse crisis models

- **Metaphors** often intensify sentiment in disasters (e.g., *“This city is drowning”*).
- Do Dinh & Gurevych (2016): “Metaphors alter syntactic and semantic patterns, misleading literal models.”

- **Irony** is used to critique or cope (e.g., *“Great, no power during a hurricane* 🙃*”*).
- Agrawal et al. (2020): “Irony tweets are often **misclassified** as non-urgent or irrelevant.”

- **Sarcasm** distorts intent (e.g., *“Amazing disaster management. Truly inspiring.”*).
- Sosea et al. (2023): “Sarcasm is genre-specific; generic models fail in disaster domains.”

Why it matters

- Crisis systems often act on **literal interpretation**. Figurative tweets can trigger **false alarms** or cause **critical misses**.
- All three studies emphasize the **domain-adaptation gap** for figurative language in emergencies.

Bottom Line

Figurative language — especially sarcasm, irony, and metaphor — is frequent and **domain-sensitive** in disaster tweets, and needs **explicit modeling**.

Why is it essential to understand the figurative context?

contempt towards the unfolding event or public policies and guidelines. This contempt is in some cases expressed as the sophisticated linguistic phenomenon that makes use of figurative language: the sarcasm (or irony). Understanding this form of speech in a disaster-centric context is essential to improving the understanding of disaster-related tweets and their intended semantic meaning. How-

past papers focus on individual literary devices or figures of speech

II. RELATED WORK

Irony detection and classification have been an area of growing interest in recent years because of its importance in sentiment analysis. In this part, we briefly discuss the

Paper 3:Figurative Language Detection Research overview

Sarcasm Detection, Helal et al. (2024); Irony Detection Agrawal et al. (2020);Metaphor Detection Do Dinh and Gurevych (2016); Hyperbole Detection Schneidermann et al. (2023)

1. Sarcasm Detection Helal et al. (2024)
Methodology: Utilizes transformer-based models (e.g., RoBERTa, DistilBERT) fine-tuned with contextual data.
Distinctiveness: Relies on detecting sentiment incongruity within context.
Performance: Achieved F1 scores of 0.99 on News Headlines and 0.90 on Mustard datasets. arXivNature+2PubMed+2MDPI+2

2. Irony Detection Agrawal et al. (2020)
Methodology: Employs BERT and XLNet models trained on the SemEval-2018 dataset.
Distinctiveness: Focuses on identifying subtle contradictions and unexpected expressions.
Performance: Reported F1 scores of 0.70 (BERT) and 0.74 (XLNet). ResearchGate

3. Metaphor Detection Do Dinh and Gurevych (2016)

Methodology: Applies neural networks combined with word embeddings for token-level classification.
Distinctiveness: Targets abstract conceptual mappings rather than overt sentiment shifts.
Performance: Achieved an F1 score of approximately 0.60 on the VUA Metaphor Corpus. SIGHUM+2download.mmag.hrz.tu-darmstadt.de+2ResearchGate+2MIT Mathematics+1ACL Anthology+1

4. Hyperbole Detection Schneidermann et al. (2023)
Methodology: Implements a multi-task learning framework to detect hyperbole and metaphor simultaneously.
Distinctiveness: Identifies exaggerated language patterns, often overlapping with metaphor detection.
Performance: Reported F1 scores of 0.687 for hyperbole and 0.805 for metaphor detection.

Paper 4: A Sentiment-Aware Contextual Model for Real-Time Disaster Prediction Using Twitter Data

Guizhe Song and Degen Huang(2021)

Objective

The goal is to build an accurate model that can predict real-time disasters from Tweets to assist in disaster response planning

Data Collection

Used the Kaggle disaster Tweet dataset, consisting of 10,876 samples: 4,692 disaster Tweets and 6,184 non-disaster Tweets.

Data Preprocessing

- Removed irrelevant elements like hashtags, emoticons, and punctuation.
- Standardized the Tweets (e.g., converting contractions like "We've" to "We have").
- Tokenized text to transform it into numerical sequences suitable for input.
- Converted Uppercase to Lowercase

Machine Learning Model

- SentiBERT: Generates sentiment-aware contextual embeddings to understand the emotional tone of Tweets.
- BiLSTM with Attention: Captures sequential dependencies and adds attention to focus on significant words for disaster prediction.
- CNN: Extracts local features from the sentence to enhance predictive accuracy by detecting patterns in the data.

Findings:

- The proposed SentiBERT-BiLSTM-CNN model outperformed baseline models like BERT and other hybrid models, achieving a higher F1 score.
- Best performance was achieved with F1 score of 0.7275

machine learning model which uses it. The text is cleaned from noise in form
oticons, punctuation, letters in a different case, stop words and so on.

- **Removed Emoticons which is a significant feature of identifying sarcasm**
- **Converted Uppercase to Lowercase**
- **This paper does not account for punctuations.**
- **F1 score - 0.7275**

Gaps & Targets

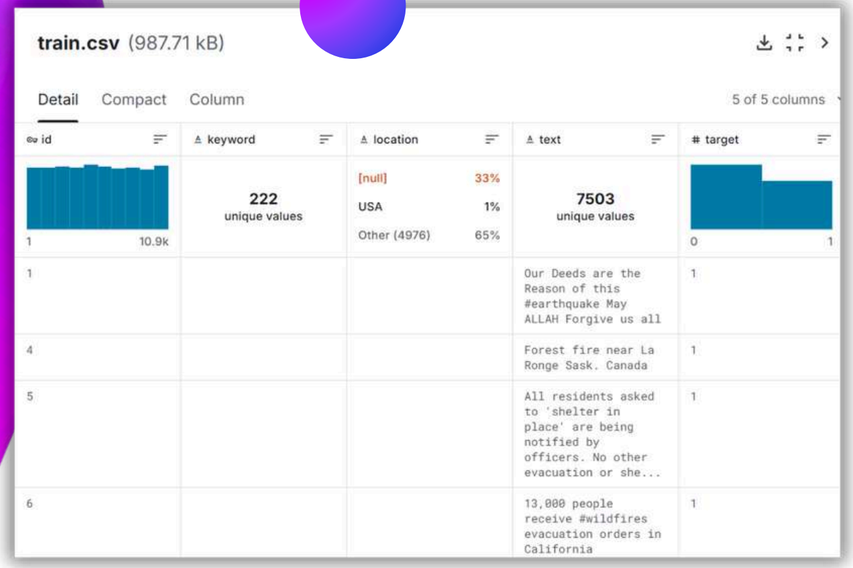
Incorporation of Emojis, Punctuation, and Capitalization:
Previous research has largely overlooked the role of emojis, punctuation marks, and capitalization in textual analysis. These elements often convey important semantic or emotional cues. Our model aims to integrate these overlooked features to enhance the understanding and detection of figurative language.

Holistic Detection of Figurative Language:
Most prior studies have focused on detecting individual literary devices in isolation (such as metaphors or similes). However, figurative language encompasses a range of such devices. Our research will address this gap by developing a model capable of detecting figurative language as a whole, thereby offering a more comprehensive solution.

Contextual Enhancement through Multi-Task Learning:
We plan to utilize two distinct datasets—one for sentiment analysis and another for disaster-related tweet classification. This dual-task approach allows us to capture both the emotional tone and situational context of the text, leading to deeper and more accurate figurative language understanding

Dataset

Data Size & Features


train.csv (987.71 kB)

Detail Compact Column 5 of 5 columns

id	keyword	location	text	target
1 – 10.9k	222 unique values	[null] 33% USA 1% Other (4976) 65%	7503 unique values	0 – 1
1			Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all	1
4			Forest fire near La Ronge Sask. Canada	1
5			All residents asked to 'shelter in place' are being notified by officers. No other evacuation or she...	1
6			13,000 people receive #wildfires evacuation orders in California	1

Twitter Disaster Detection Dataset

Total Data Points: 7613 tweets
Features (5 Columns):

- id – Unique identifier for each tweet.
- keyword – Disaster-related keywords (partially available).
- location – Geographical location (contains missing values).
- text – The tweet content.
- target – 1 (real disaster) / 0 (not a disaster)

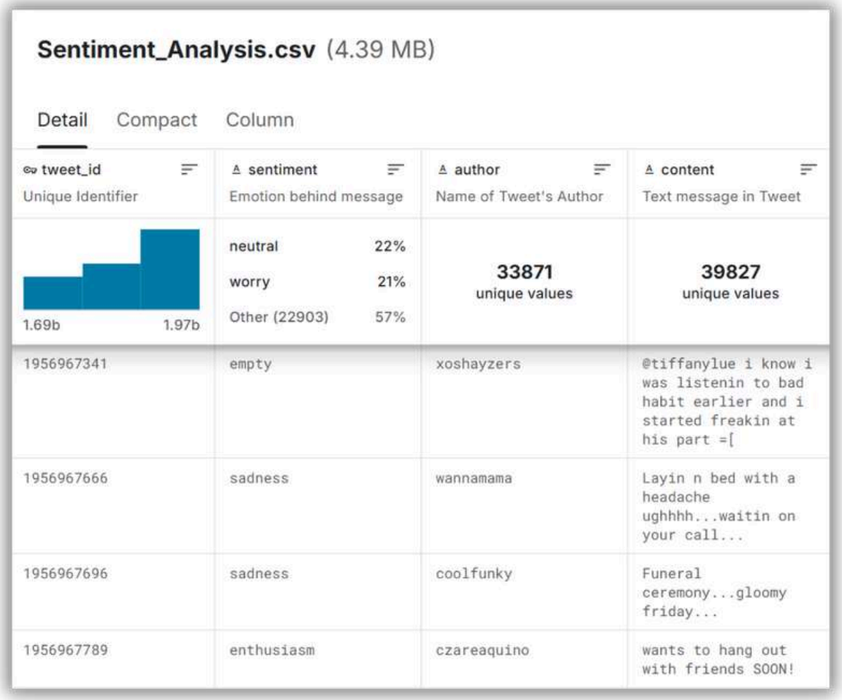
Missing Data: Keyword: 61 missing values
Location: 2,533 missing values

Sentiment Analysis Dataset

Total Data Points: 40,000 tweets
Features (4 Columns):

- tweet_id – Unique tweet identifier.
- sentiment – Emotion label (e.g., happiness, sadness, anger).
- author – Twitter username (not directly used for analysis).
- text – Tweet content.

No missing values in this dataset


Sentiment_Analysis.csv (4.39 MB)

Detail Compact Column

tweet_id Unique Identifier	sentiment Emotion behind message	author Name of Tweet's Author	content Text message in Tweet
1.69b – 1.97b	neutral 22% worry 21% Other (22903) 57%	33871 unique values	39827 unique values
1956967341	empty	xoshayzers	@tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part =[
1956967666	sadness	wannamama	Layin n bed with a headache ughhhh...waitin on your call...
1956967696	sadness	coolfunky	Funeral ceremony...gloomy friday...
1956967789	enthusiasm	czareaquino	wants to hang out with friends SOON!

After API labelling

id	keyword	location	text	predicted_label
1			Our Deeds	1
4			Forest fire	1
5			All resider	1
6			13,000 pe	1
7			Just got se	1
8			#RockyFire	1
10			#flood #di	1
13			I'm on top	1
14			There's an	1
15			I'm afraid	1
16			Three peo	1
17			Haha Sout	1
18			#raining #	1
19			#Flood in I	1
20			Damage to	1
23			What's up	2
24			I love fruit	2

API Labelling Through Gemini in Disaster Dataset

We realized that the original target column in the dataset did not effectively address the gap in our problem statement. Therefore, we decided to relabel the dataset using Gemini's API. We assigned labels as follows: 0 for fake (figurative) disasters, 1 for real (literal) disasters, and 2 for content unrelated to disasters.

How Was the Data Collected by Its Authors?

Twitter Disaster Dataset

Collected from Twitter using keyword-based filtering (e.g., "earthquake," "wildfire," "flood").

Sentimental Analysis Dataset

1. Originally collected by CrowdFlower (now Figure Eight) using crowdsourced human annotations.
2. Tweets were labeled with one of 13 emotions, making it useful for understanding sentiment nuances.

Ethical Concerns & How They Were Addressed

Disaster Related tweet dataset

Privacy and Anonymity :

- The dataset is made of publically available tweets.
- They do not include personally identifiable information (**PII removed**)

Sentiment Analysis Dataset

Bias in Labeling:

- Human annotation may introduce subjectivity in sentiment labels.
- CrowdFlower Uses multiple annotators to reduce bias and improve reliability.
- However, the dataset contained the usernames of the tweet authors because they wanted it as a feature if a certain person is more likely to do sarcastic tweets.

Data Preprocessing

Feature Extraction

01

Tweet length: Computes the number of characters in each tweet using .apply(len)

02

Word count: Counts words in each tweet using .split() and len().

03

Emoji count: Uses extract_emojis() to count emojis in tweets.

04

Punctuation count: Uses count_punctuations() to count punctuation marks in each tweet.

05

Uppercase letters length : Counts the upper case letters as it’s an important syntactic feature .

https://www.emerald.com/insight/content/doi/10.1108/idd-01-2023-0002/full/html#sec003

Missing Value

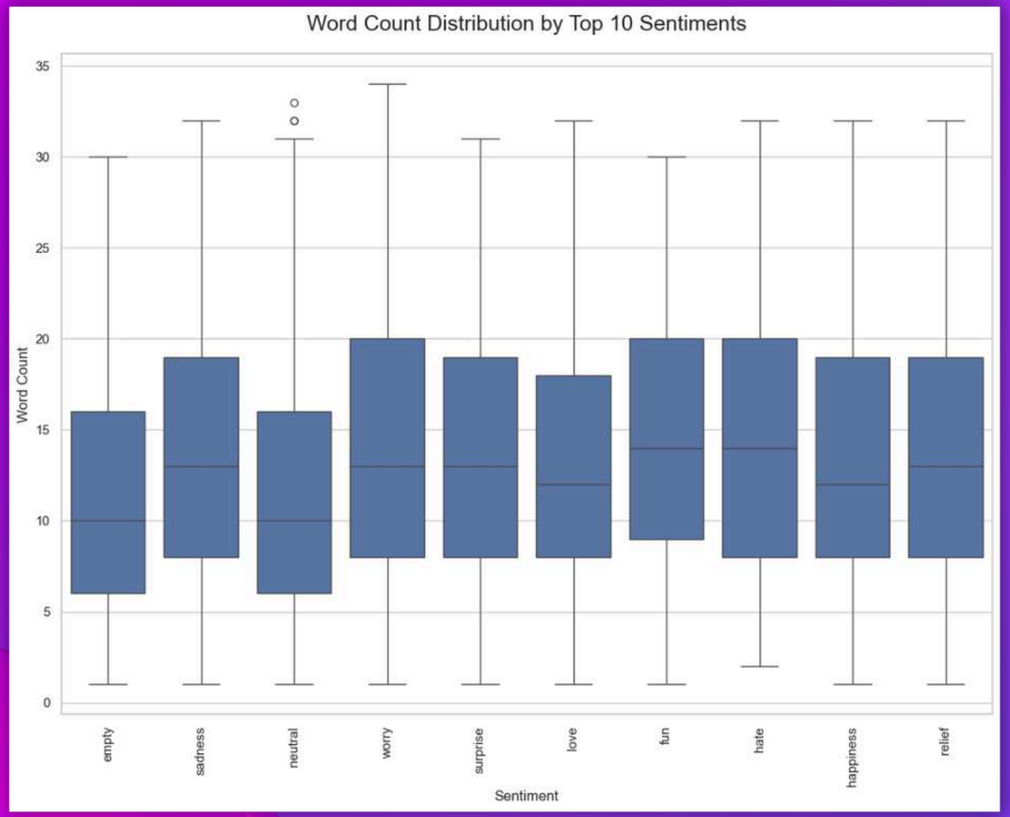
No. of missing values:

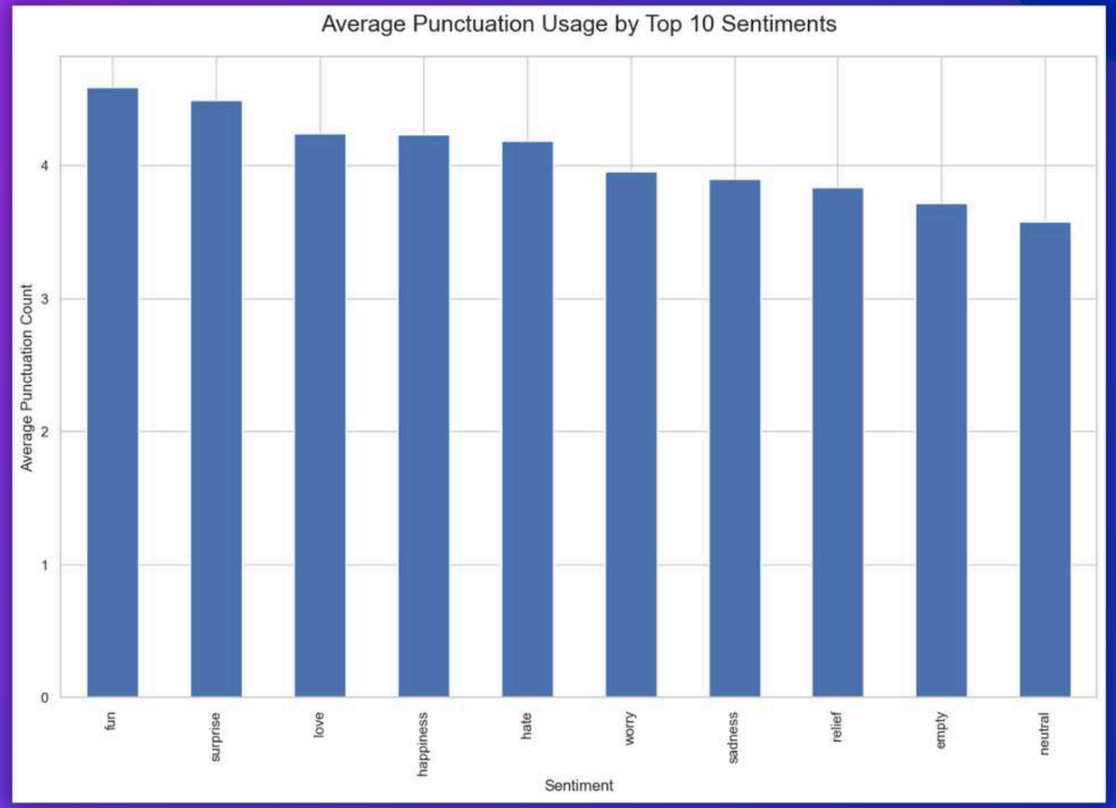
Location: 2533

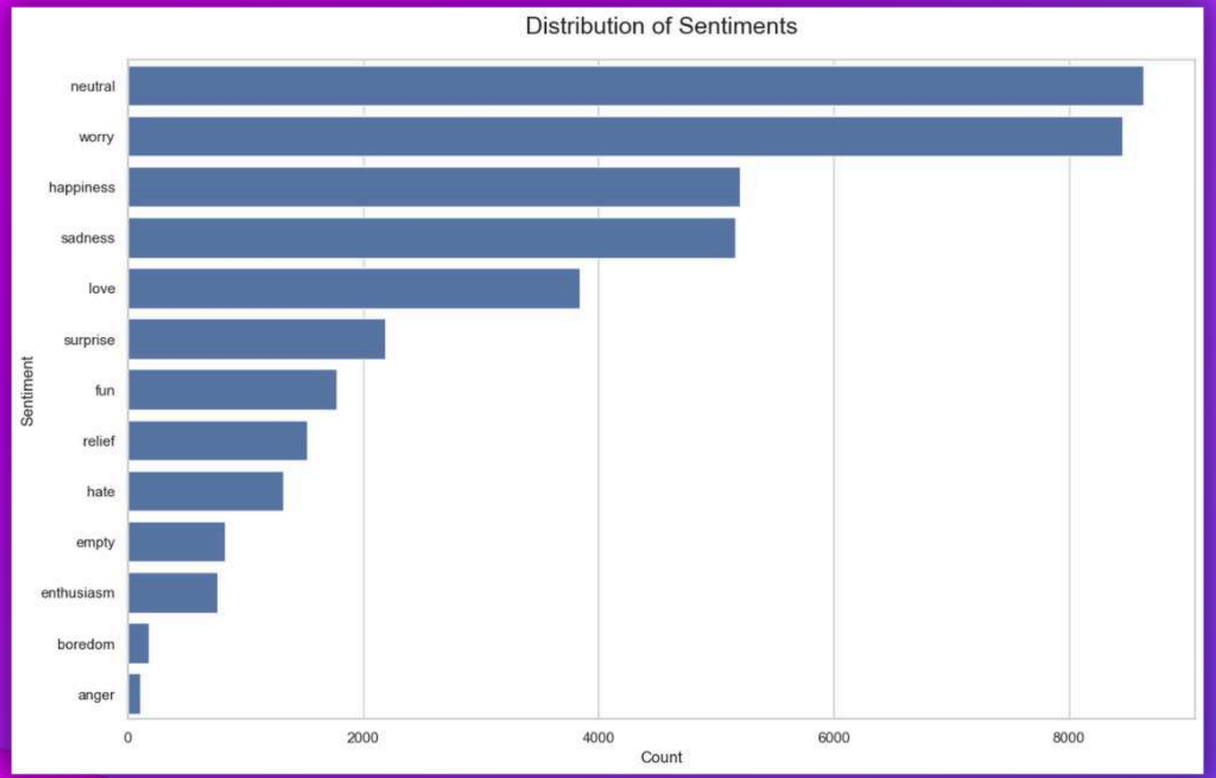
keywords: 61

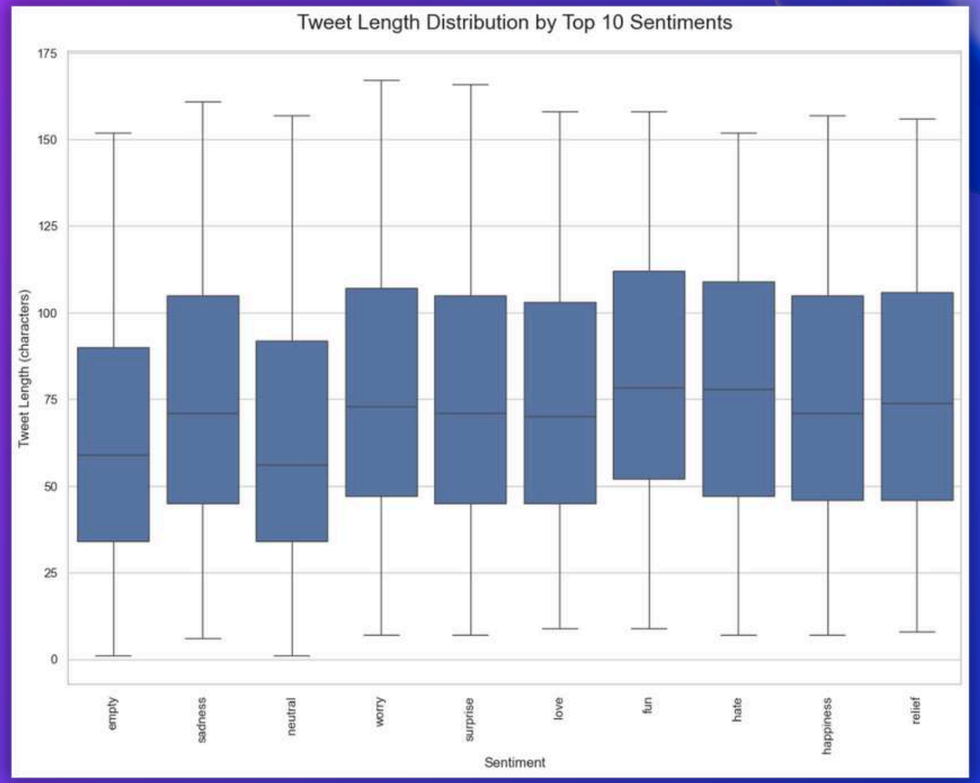
- Used None as placeholders in keyword and location columns to prevent errors.
- TF-IDF naturally handles missing values by treating them as empty text.
- Dropped the Location Column(It had no correlation with target var.)

Sentiment Data Analysis

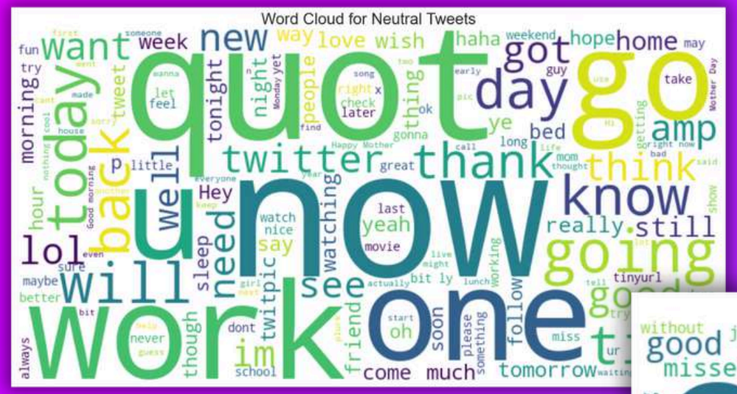


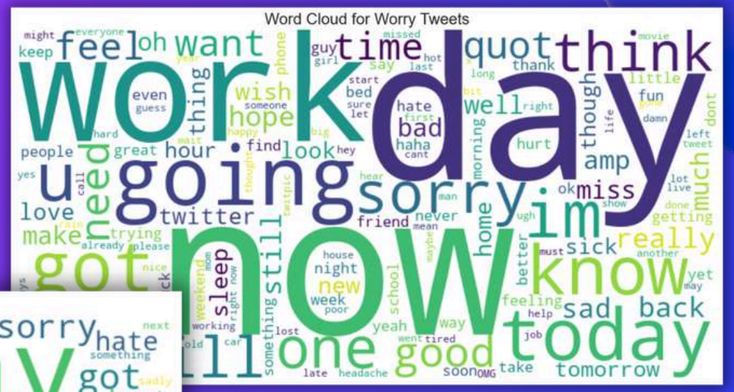


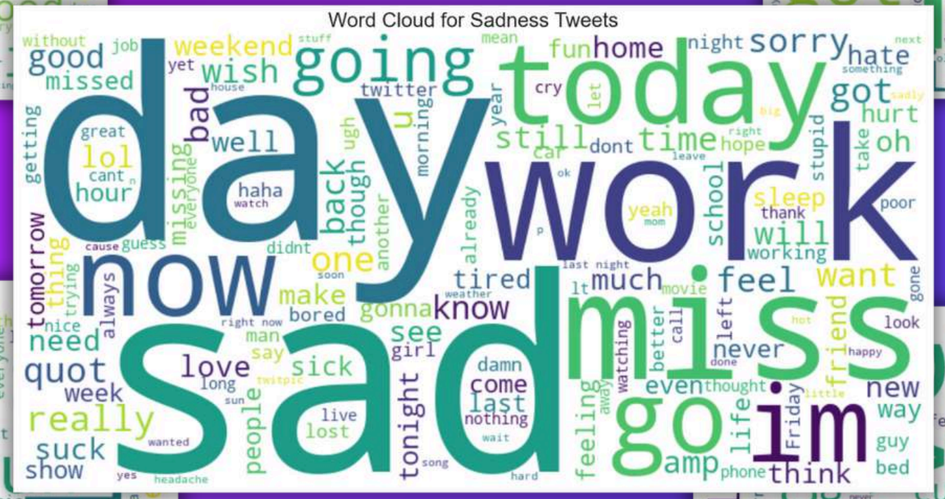

Distribution of Sentiments
neutral
worry
happiness
sadness
love
surprise
fun
relief
hate
empty
enthusiasm
boredom
anger
Sentiment
0
2000
4000
6000
8000
Count

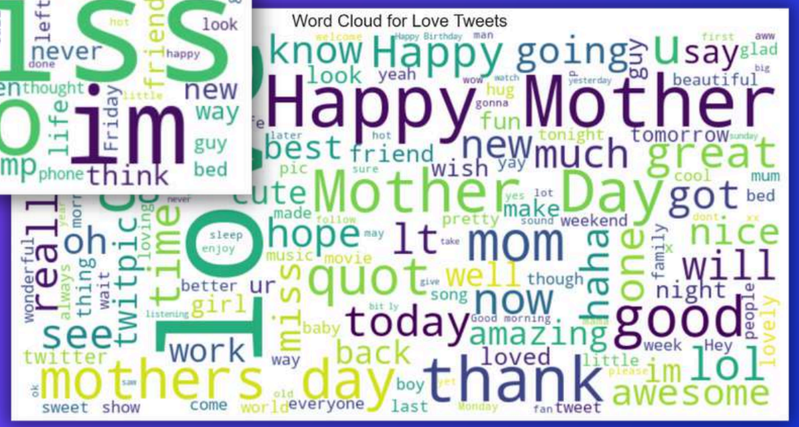

Tweet Length Distribution by Top 10 Sentiments
Tweet Length (characters)
175
150
125
100
75
50
25
0
empty
sadness
neutral
worry
surprise
love
fun
hate
happiness
relief
Sentiment

Generating word Cloud For Top 5 Elements

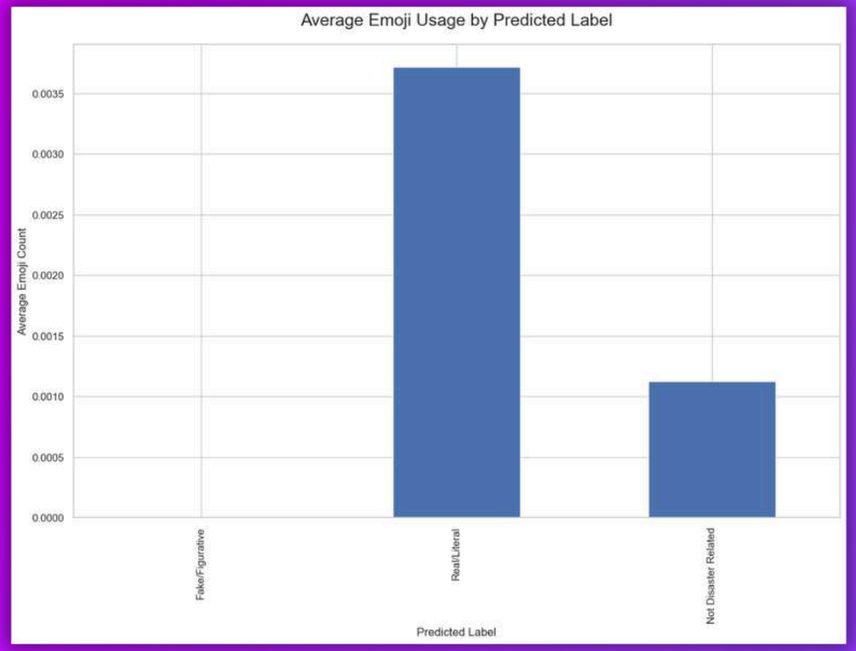


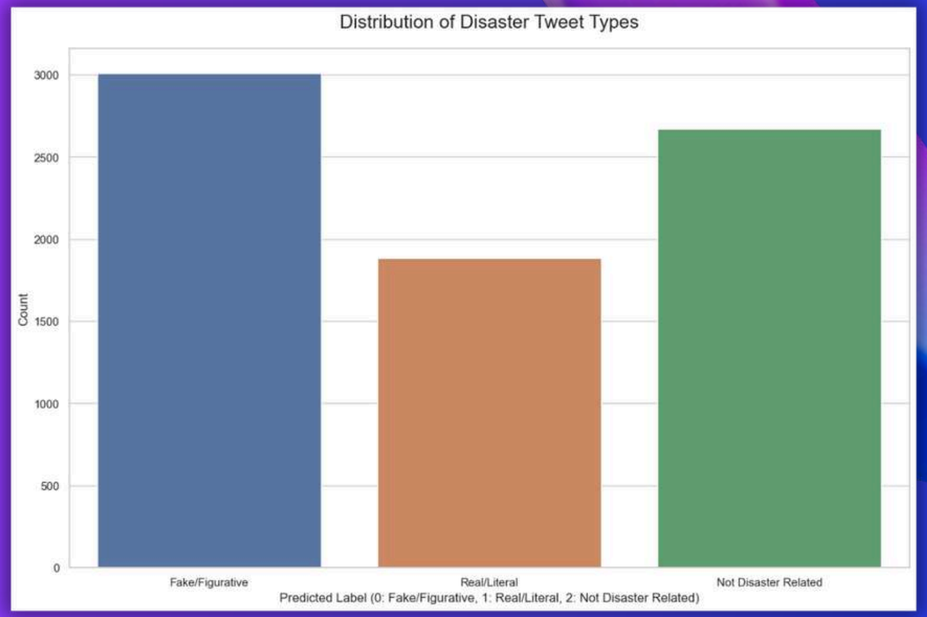


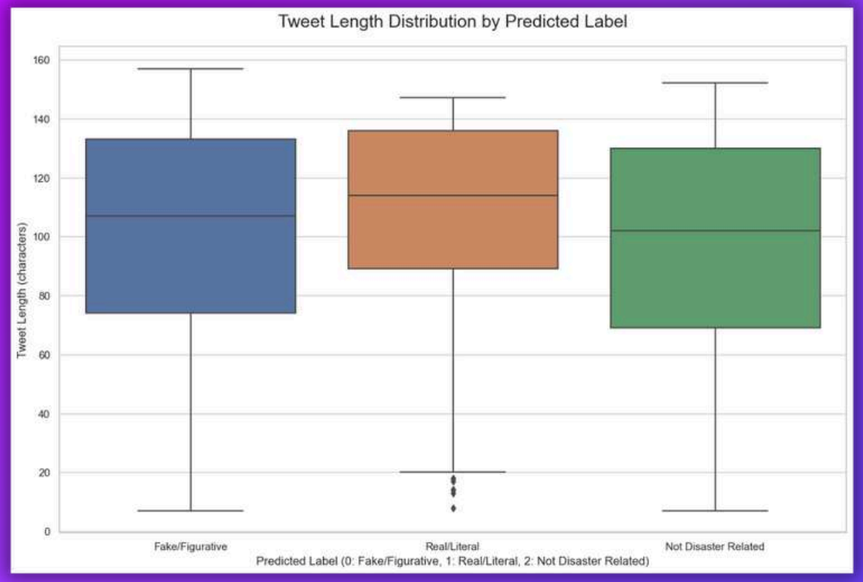


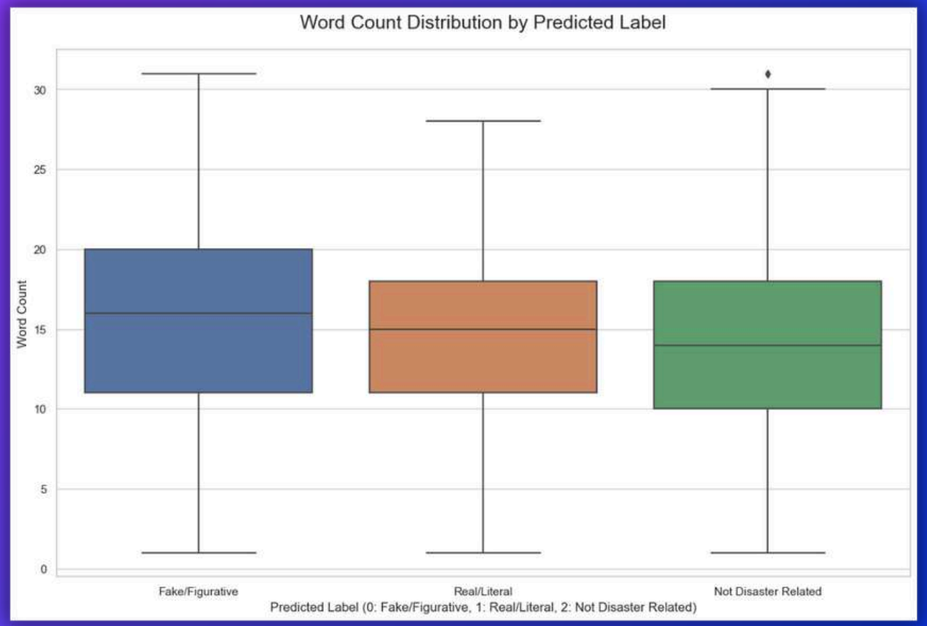


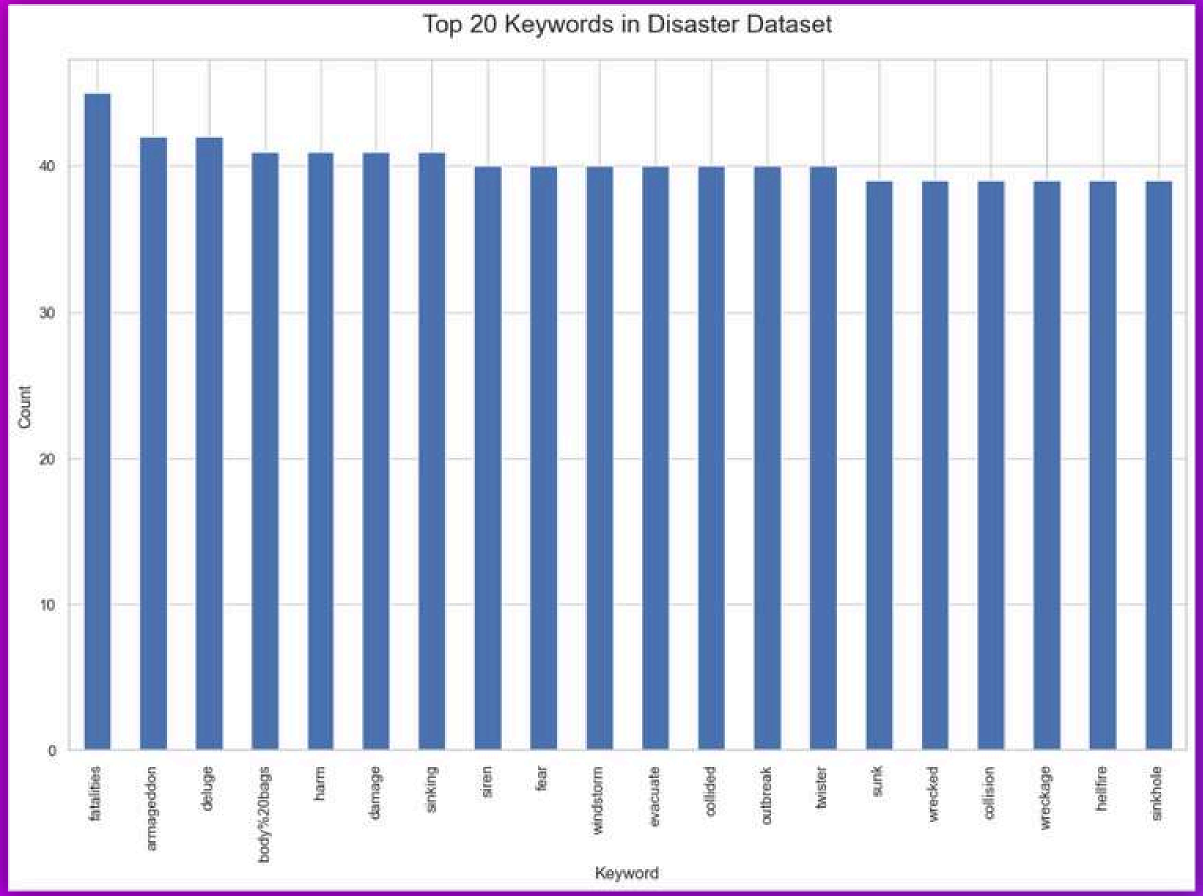
Disaster Tweet Data Analysis

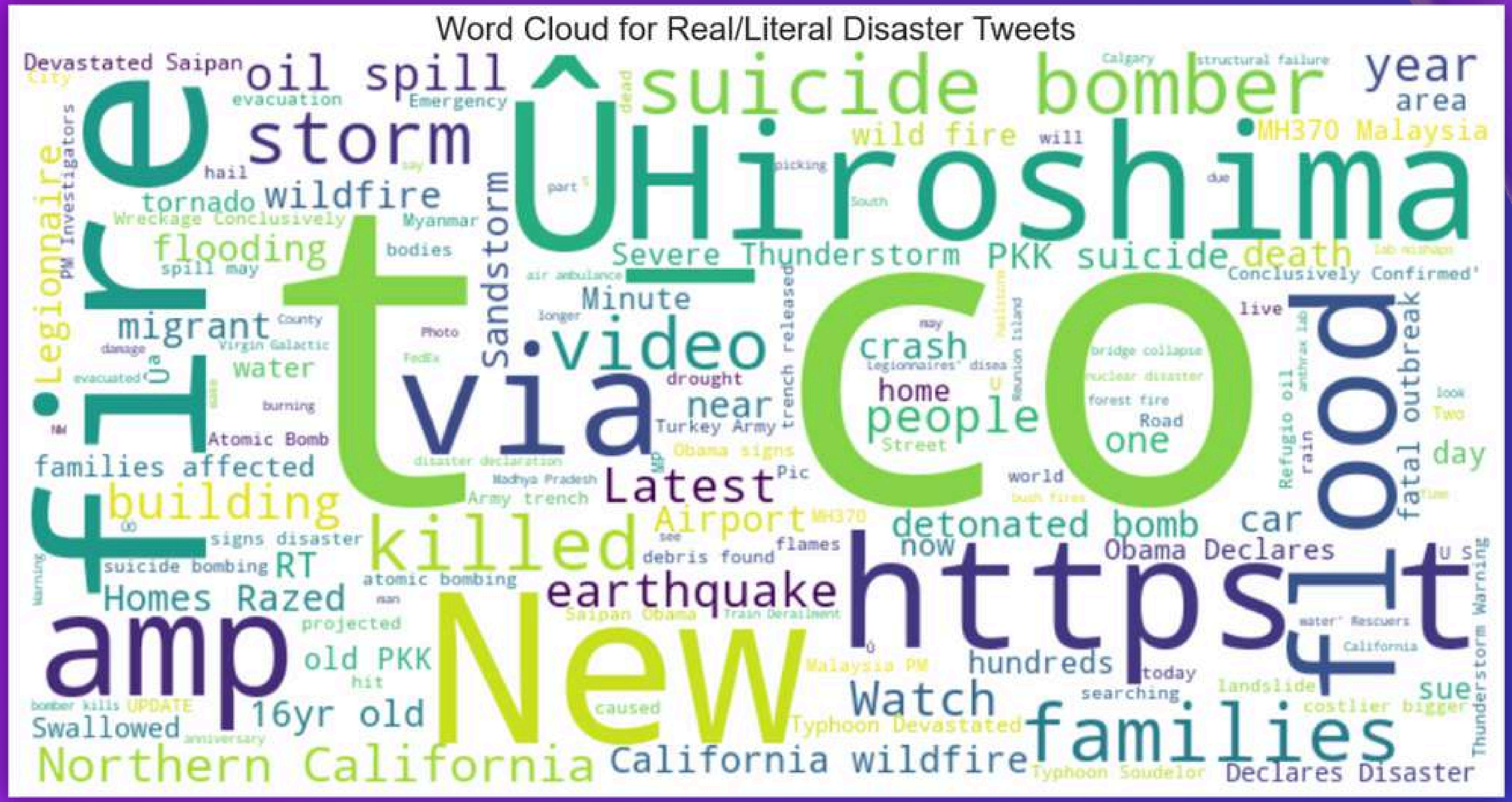


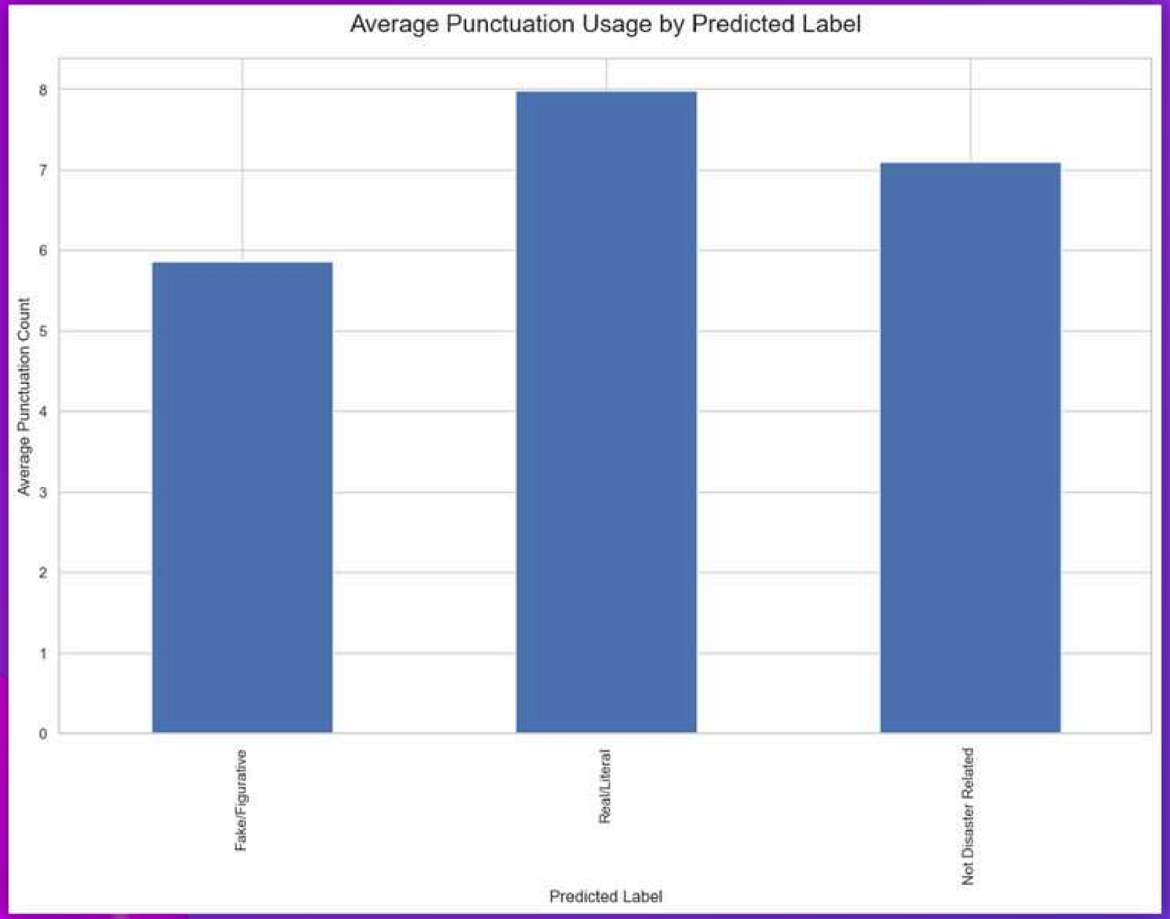


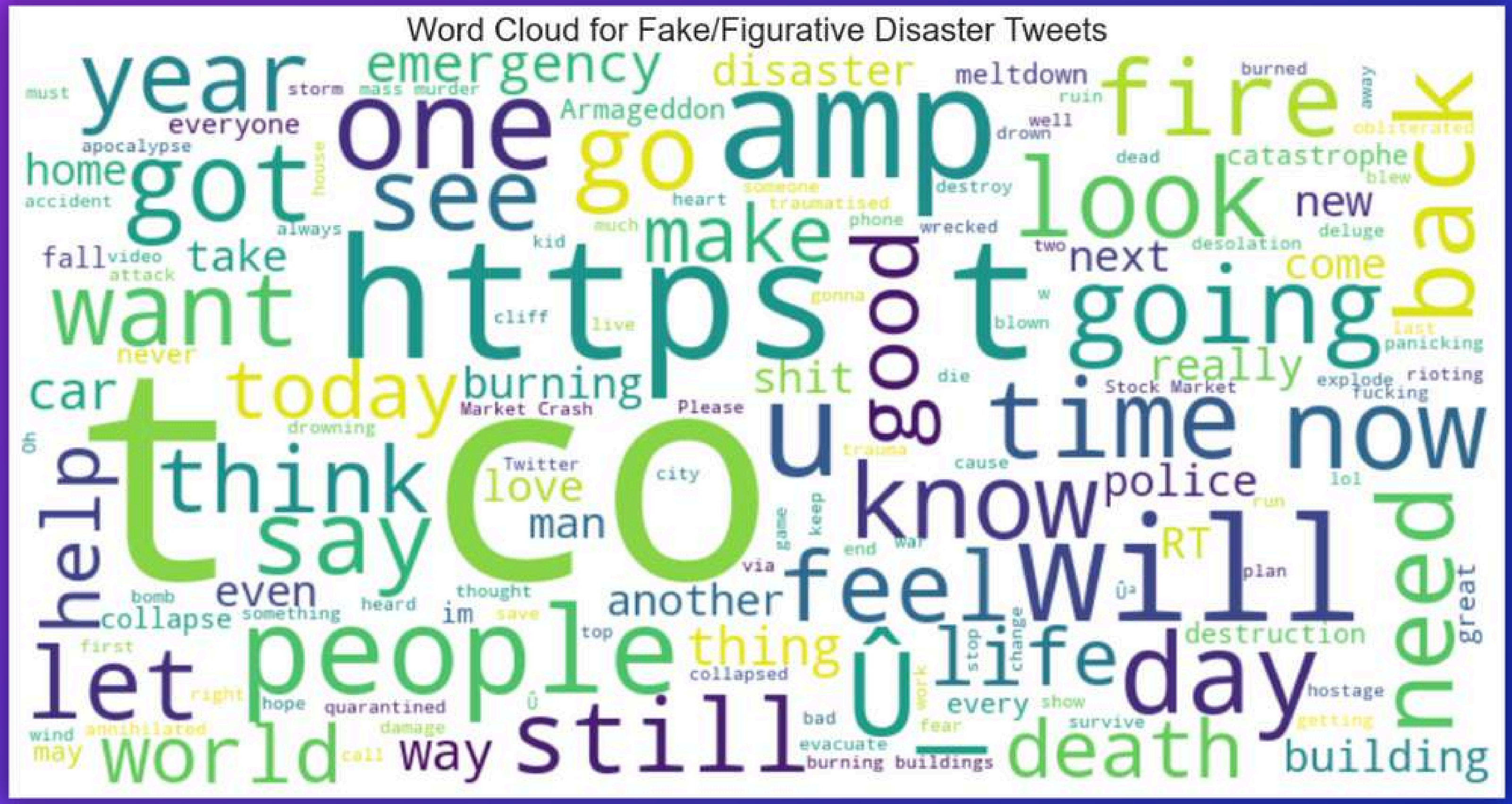


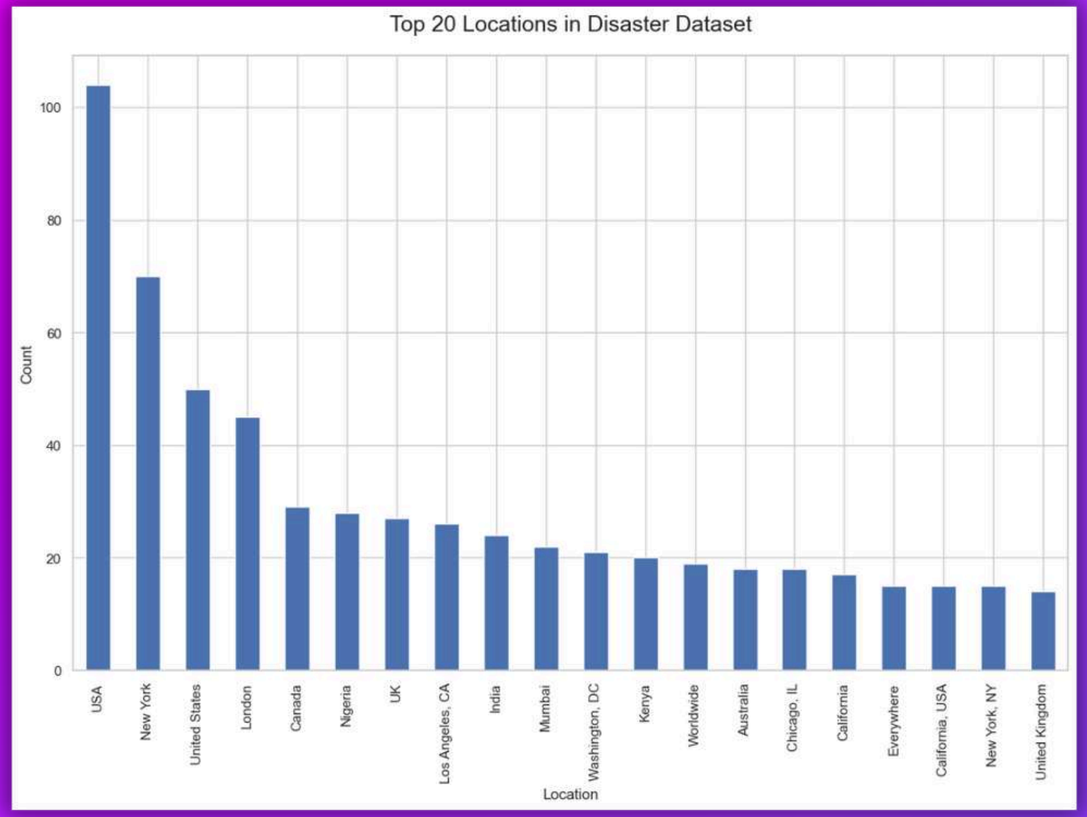


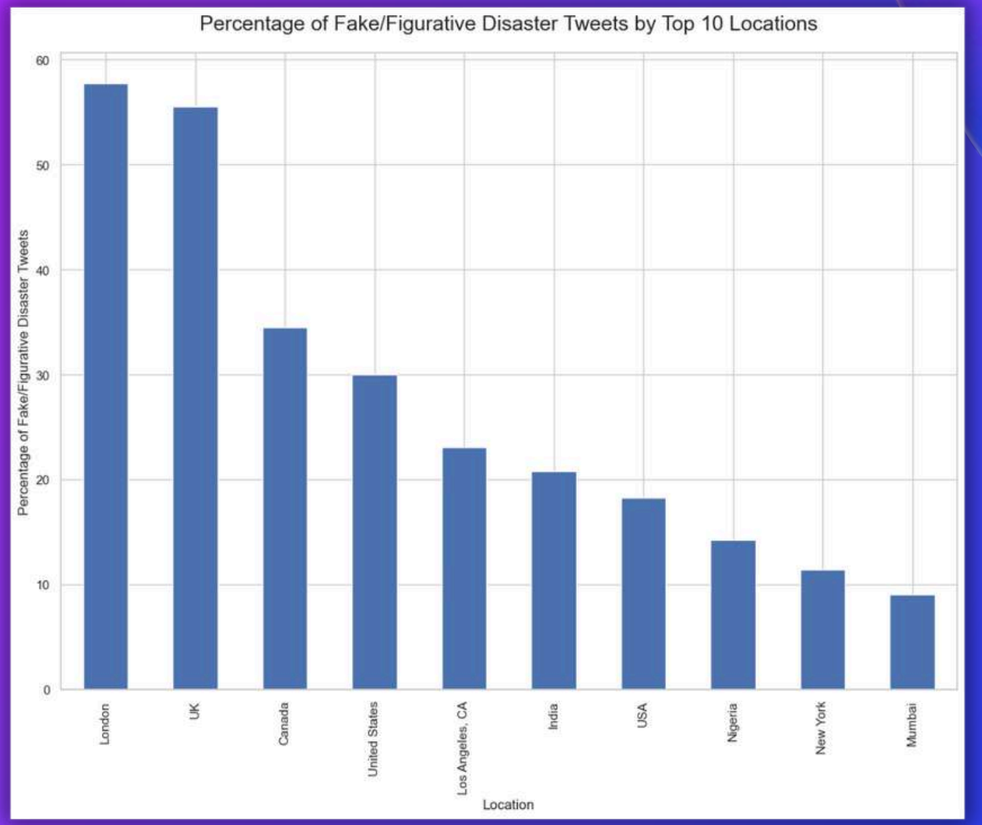

Top 20 Keywords in Disaster Dataset
Count
0
10
20
30
40
fatalities
armageddon
deluge
body%20bags
harm
damage
sinking
siren
fear
windstorm
evacuate
collided
outbreak
twister
sunk
wrecked
collision
wreckage
hellfire
sinkhole
Keyword

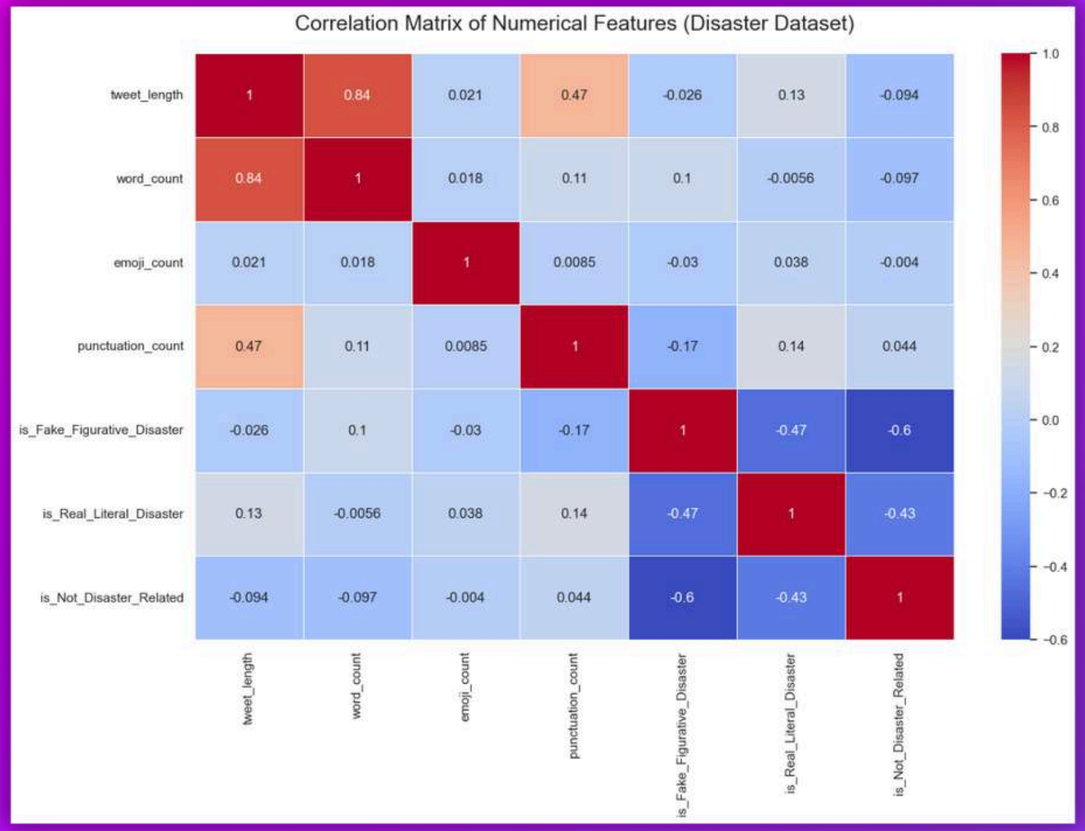

Word Cloud for Real/Literal Disaster Tweets
t co
New
https
Hiroshima
fire
via
amp
flood
suicide bomber
storm
oil spill
killed
families
video
building
earthquake
Northern California
California wildfire
Homes Razed
wildfire
flooding
Severe Thunderstorm
PKK suicide
death
crash
people
Latest
Watch
16yr old
detonated bomb
Obama Declares
fatal outbreak
Legionnaire
Sandstorm
migrant
families affected
year
area
MH370 Malaysia
Devastated Saipan
Declares Disaster

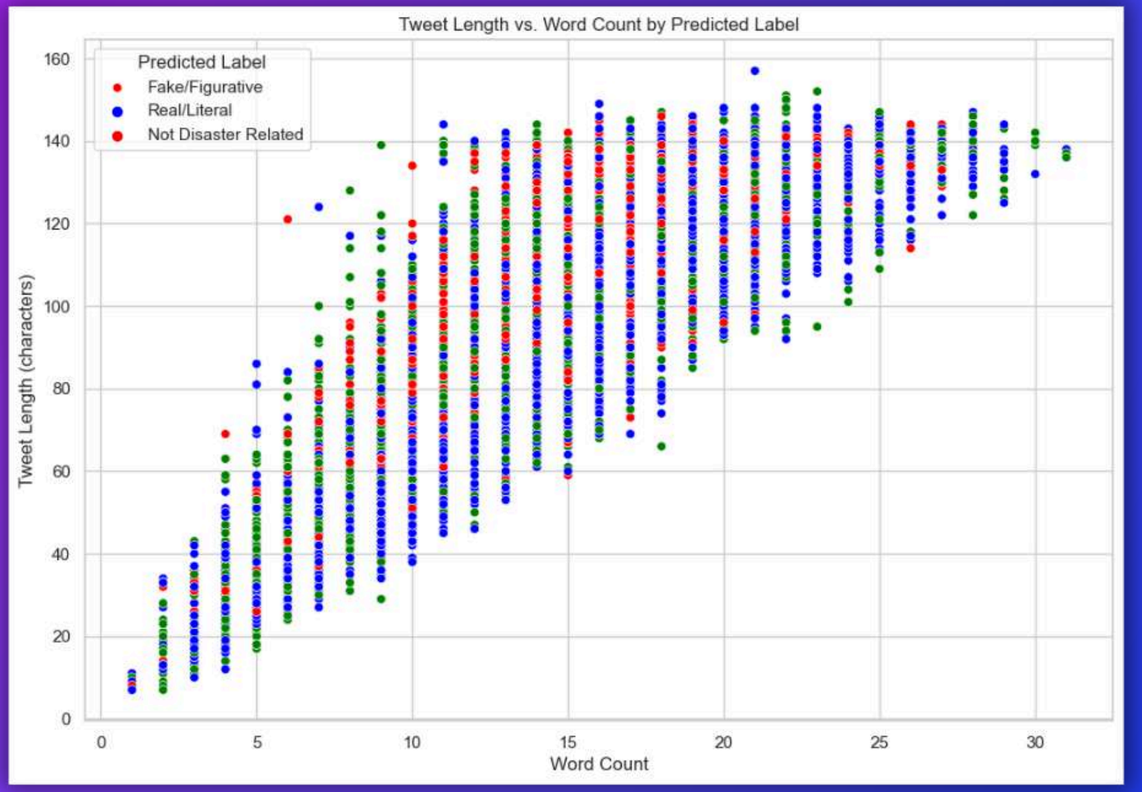

Average Punctuation Usage by Predicted Label
Average Punctuation Count
0
1
2
3
4
5
6
7
8
Fake/Figurative
Real/Literal
Not Disaster Related
Predicted Label


Word Cloud for Fake/Figurative Disaster Tweets
t co
https
amp
one
year
got
see
go
look
fire
back
want
going
time now
know
will
feel
people
day
death
still
let
world
think
say
today
help
need
good
make
life
emergency
disaster
building
police


Top 20 Locations in Disaster Dataset
Count
Location
0
20
40
60
80
100
USA
New York
United States
London
Canada
Nigeria
UK
Los Angeles, CA
India
Mumbai
Washington, DC
Kenya
Worldwide
Australia
Chicago, IL
California
Everywhere
California, USA
New York, NY
United Kingdom


Percentage of Fake/Figurative Disaster Tweets by Top 10 Locations
Percentage of Fake/Figurative Disaster Tweets
Location
0
10
20
30
40
50
60
London
UK
Canada
United States
Los Angeles, CA
India
USA
Nigeria
New York
Mumbai


Correlation Matrix of Numerical Features (Disaster Dataset)
tweet_length
word_count
emoji_count
punctuation_count
is_Fake_Figurative_Disaster
is_Real_Literal_Disaster
is_Not_Disaster_Related
1 0.84 0.021 0.47 -0.026 0.13 -0.094
0.84 1 0.018 0.11 0.1 -0.0056 -0.097
0.021 0.018 1 0.0085 -0.03 0.038 -0.004
0.47 0.11 0.0085 1 -0.17 0.14 0.044
-0.026 0.1 -0.03 -0.17 1 -0.47 -0.6
0.13 -0.0056 0.038 0.14 -0.47 1 -0.43
-0.094 -0.097 -0.004 0.044 -0.6 -0.43 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Tweet Length vs. Word Count by Predicted Label
Predicted Label
Fake/Figurative
Real/Literal
Not Disaster Related
Tweet Length (characters)
Word Count
0
20
40
60
80
100
120
140
160
0
5
10
15
20
25
30

Text Preprocessing

Initialize Lemmatizer (WordNetLemmatizer)
Converts words to their root form (e.g., "running" → "run").
Helps standardize text and reduce vocabulary size for better NLP model performance.
Load Stopwords (stopwords.words('english'))
Removes common but unimportant words (e.g., "the", "is", "and") to focus on meaningful content.
Improves efficiency by reducing noise in the text.
Define Slang Dictionary (slang_dict)
Expands commonly used tweet slang and abbreviations (e.g., "u" → "you", "lol" → "laugh out loud").
Helps in better understanding of informal language in social media posts.

Why is This Important?
Improves text standardization → Ensures uniform word representation.
Enhances model accuracy → Reduces irrelevant variations in text.
Handles social media language → Essential for analyzing tweets effectively.

Why Use These Tools?
WordNetLemmatizer (from nltk) → Ensures proper word form conversion based on context.
NLTK Stopwords → Filters out non-informative words to improve efficiency.
Custom Slang Dictionary → Bridges the gap between formal and informal language used in tweets.
Impact: These steps clean and normalize text, making sentiment analysis and disaster classification more accurate!

METHODOLOGY

01

DATA PREPROCESSING

- Text cleaning and normalization
- Extraction of rich text features (e.g., length, punctuation, capitalization)
- Sentiment extraction using GloVe embeddings and a sentiment analysis model

02

FEATURE ENGINEERING

- TF-IDF vectors
- Sentiment features extracted via word and GloVe embeddings
- Categorical features and other rich text features combined

03

MODEL EVALUATION & TRAINING

- Cross-validation on multiple models:
- Random Forest, Gradient Boosting (best performer)

XGBoost.

- Gradient Boosting achieved the highest accuracy (~70%) and F1-score.
- The final model saved as .pkl file for inference.

Code Structure

- **Data Loading & Preprocessing**: Scripts to load datasets, clean text, extract features
- **Embedding Models**: Loading GloVe and Word2Vec embeddings
- **Model Training:** Training and cross-validation of multiple classifiers
- **Model Saving:** Best model saved as .pkl file
- **Prediction Scripts:** Functions to predict class for new tweets using the saved model
- **Dashboard**: dashboardFifteen.py runs the Streamlit app for interactive prediction

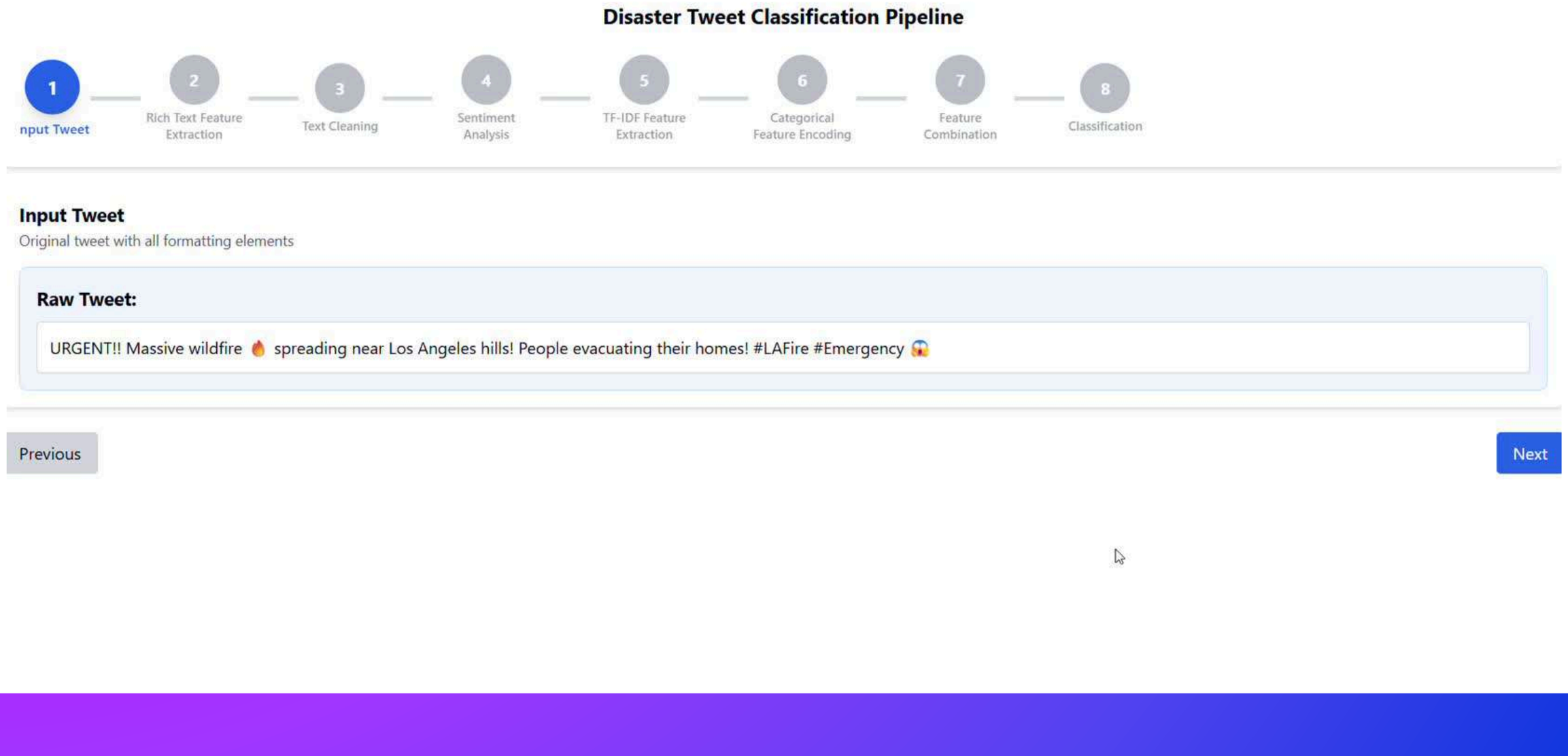
vectorizers + models used to train the sentiment model

```
TF-IDF (word) + Logistic Regression: 0.2380
TF-IDF (word) + Random Forest: 0.3131
TF-IDF (word) + Naive Bayes: 0.3171
TF-IDF (word) + Linear SVM: 0.2391
TF-IDF (word+bigram) + Logistic Regression: 0.2416
TF-IDF (word+bigram) + Random Forest: 0.3095
TF-IDF (word+bigram) + Naive Bayes: 0.3206
TF-IDF (word+bigram) + Linear SVM: 0.2392
TF-IDF (word+bigram+trigram) + Logistic Regression: 0.2411
TF-IDF (word+bigram+trigram) + Random Forest: 0.3100
TF-IDF (word+bigram+trigram) + Naive Bayes: 0.3206
TF-IDF (word+bigram+trigram) + Linear SVM: 0.2357
Count (word) + Logistic Regression: 0.2504
Count (word) + Random Forest: 0.2742
Count (word) + Naive Bayes: 0.3300
Count (word) + Linear SVM: 0.2347
Count (word+bigram) + Logistic Regression: 0.2510
Count (word+bigram) + Random Forest: 0.2705
Count (word+bigram) + Naive Bayes: 0.3306
Count (word+bigram) + Linear SVM: 0.2387
```

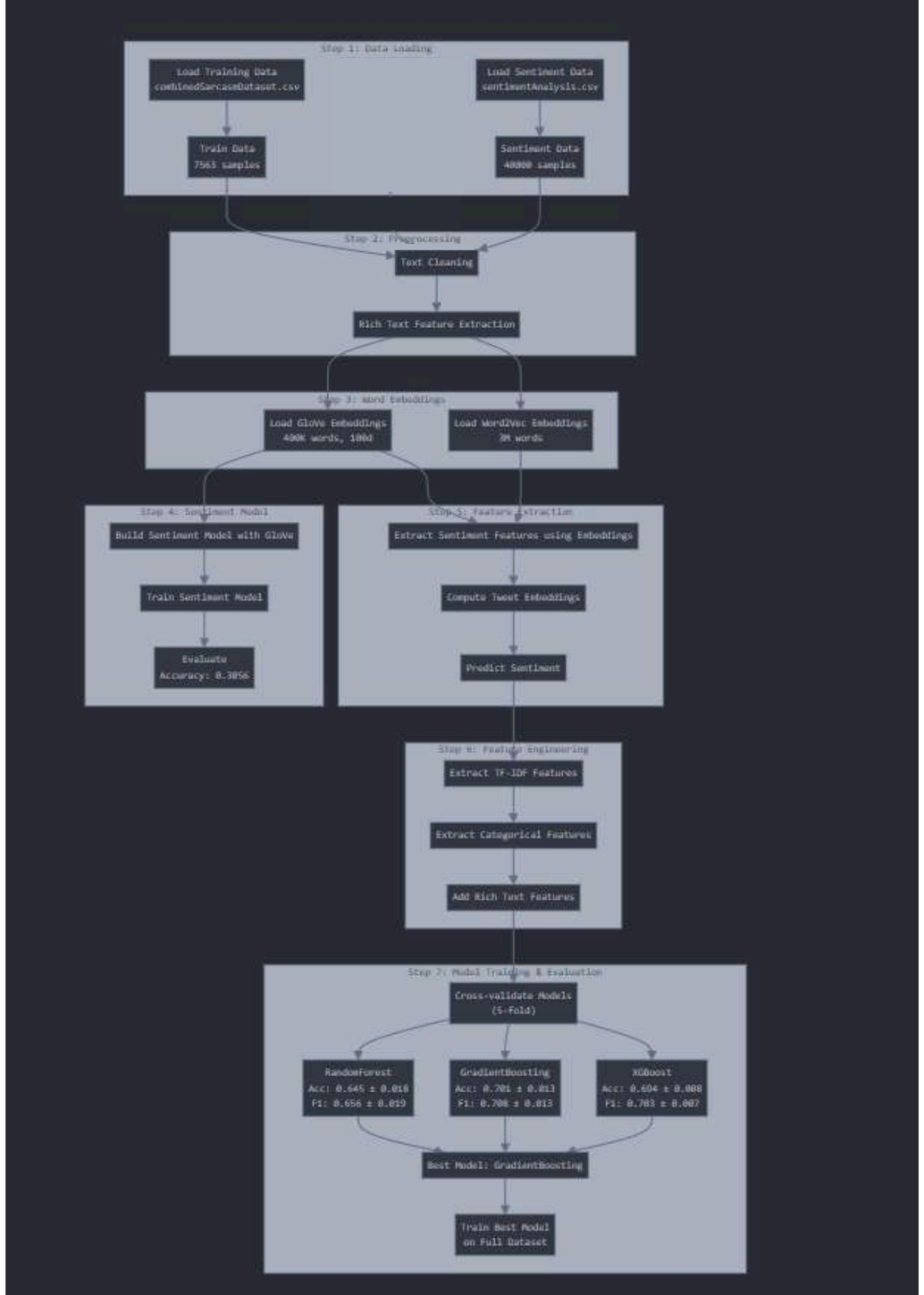
models used to train the combined pipeline

RandomForest
Gradient Boosting
XGBoost

Pipeline


Disaster Tweet Classification Pipeline
1
nput Tweet
2
Rich Text Feature Extraction
3
Text Cleaning
4
Sentiment Analysis
5
TF-IDF Feature Extraction
6
Categorical Feature Encoding
7
Feature Combination
8
Classification
Input Tweet
Original tweet with all formatting elements
Raw Tweet:
URGENT!! Massive wildfire 🔥 spreading near Los Angeles hills! People evacuating their homes! #LAFire #Emergency 😱
Previous
Next

Our Model Pipeline



Results

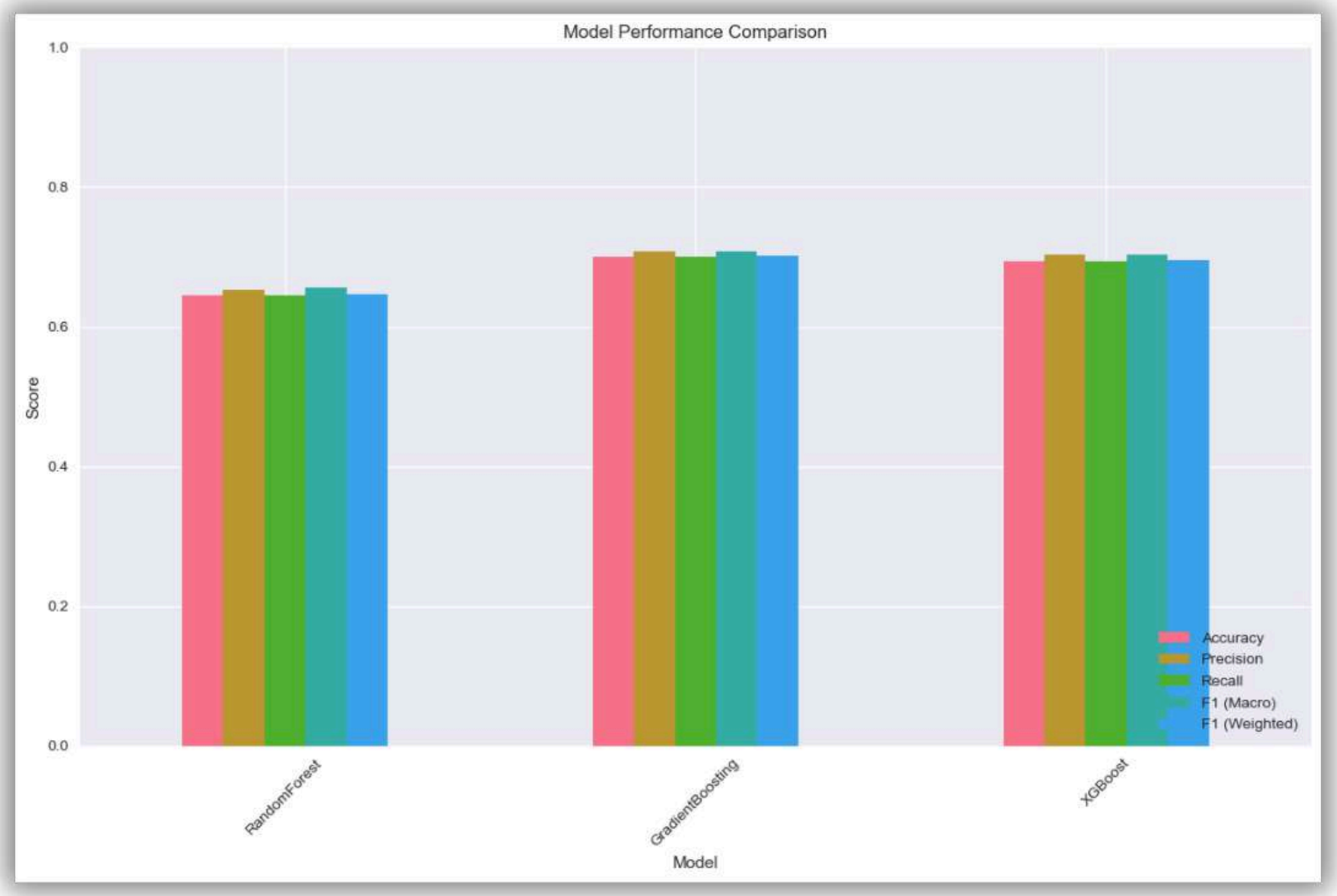
The Gradient Boosting model achieved:

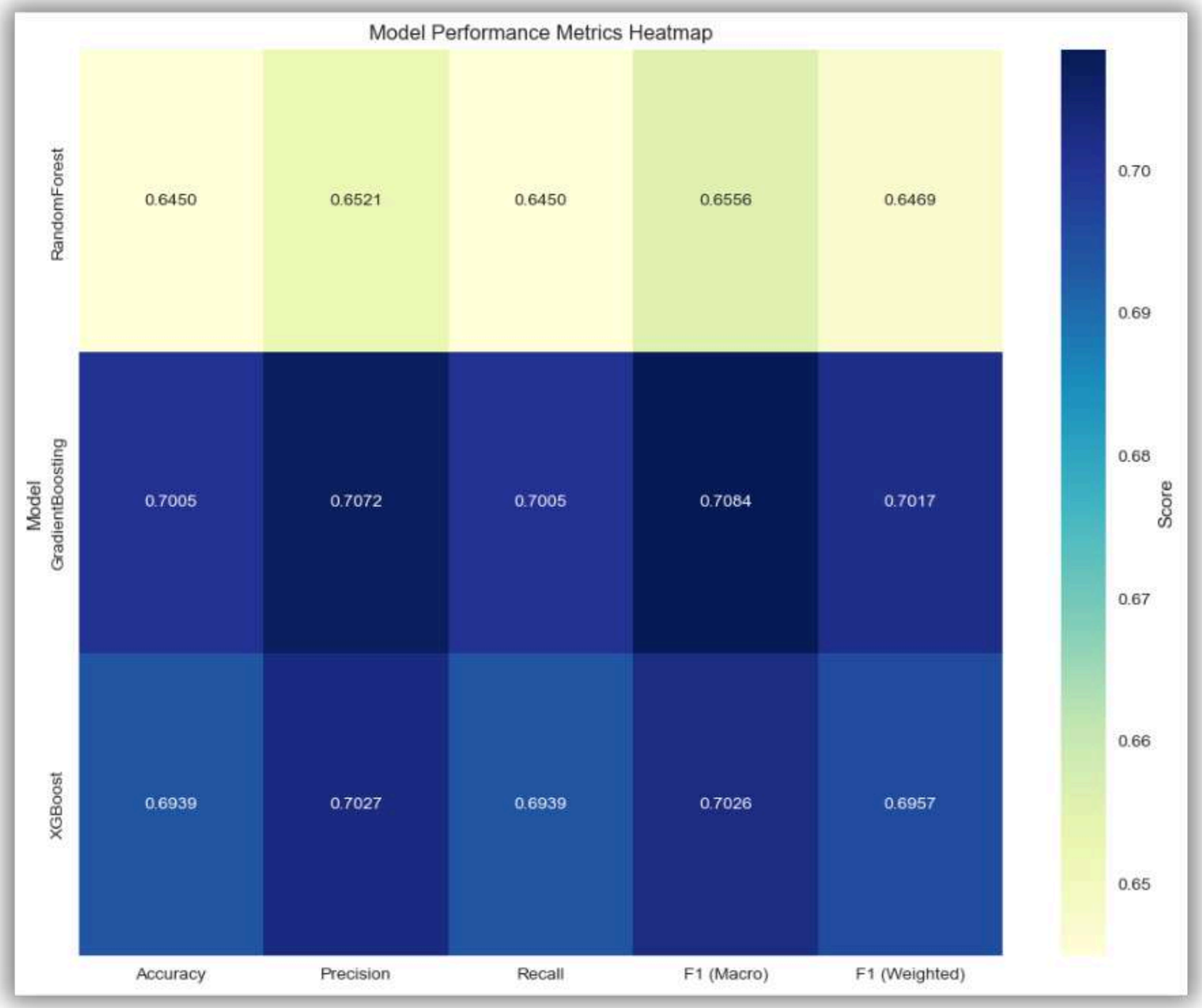
Accuracy: 70.05% ± 1.29%

Precision: 70.72% ± 1.32%

Recall: 70.05% ± 1.29%

F1 Score: 70.17% ± 0.013%


Model Performance Comparison
1.0
0.8
0.6
0.4
0.2
0.0
Score
RandomForest
GradientBoosting
XGBoost
Model
Accuracy
Precision
Recall
F1 (Macro)
F1 (Weighted)


Model Performance Metrics Heatmap
Model
RandomForest
GradientBoosting
XGBoost
0.6450
0.6521
0.6450
0.6556
0.6469
0.7005
0.7072
0.7005
0.7084
0.7017
0.6939
0.7027
0.6939
0.7026
0.6957
Accuracy
Precision
Recall
F1 (Macro)
F1 (Weighted)
Score
0.70
0.69
0.68
0.67
0.66
0.65

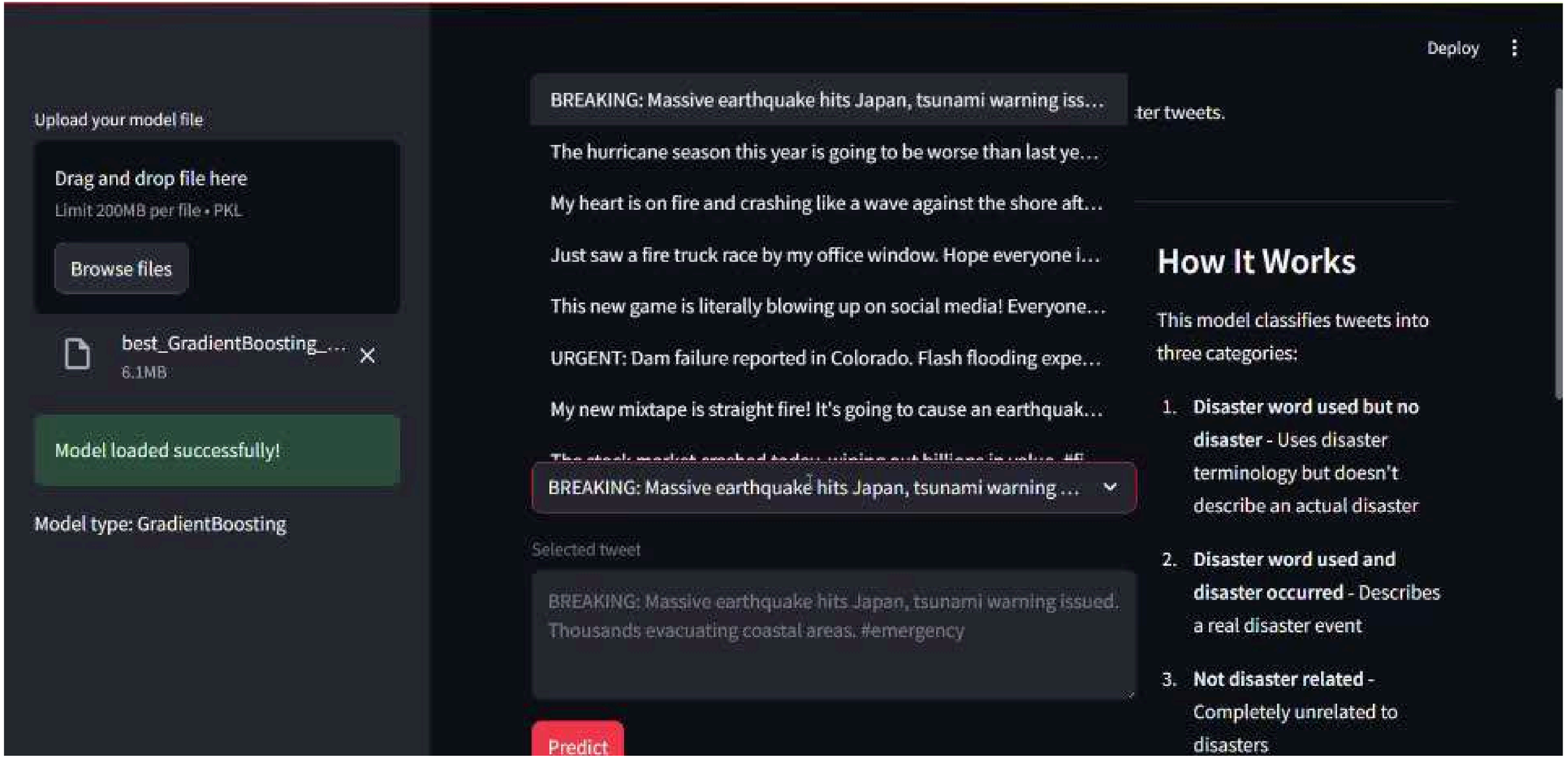
DashBoard

How to Use

Running the Dashboard
A Streamlit dashboard is provided to interactively test tweet classification.
bash

- python -m streamlit run dashboardFifteen.py
- Upload the trained model .pkl file through the dashboard.
- Enter a tweet, and the dashboard returns:

Dash Board For Our Model



Why This Model?

- Previous studies in disaster tweet classification mostly focus on keyword matching or sentiment analysis but struggle with sarcasm and context detection.
- We explored various NLP feature extraction methods such as TF-IDF, word bi-grams, and tri-grams
- We also experimented with word embeddings like GloVe and Word2Vec to capture semantic nuances in tweets.
- Several classification models such as Logistic Regression, Random Forest, Gradient Boosting, and XGBoost were evaluated.
- The best performing model was Gradient Boosting trained on text rich features , TF-IDF features and embedding-based sentiment features.

Problems faced

Problem 1

Limited Dataset Size:
Due to the small size of our dataset, we encountered difficulties in achieving high model accuracy. This limitation also restricted our ability to train more complex architectures such as LSTM, which would have been better suited for handling sequence-based tasks like ours.

Problem 2

Error Propagation Between Models:
Since our pipeline involved training two separate models—starting with sentiment analysis, followed by disaster classification—it is likely that noise and errors from the first model were propagated into the second. This cascading effect may have hindered the performance of the final classifier and prevented accuracy improvements.

Problem 3

High Memory and Computational Cost of Embeddings:
Using large embedding vectors significantly increased memory consumption and computational requirements. While reducing the embedding size could help optimize resource usage, it risks losing semantic richness and contextual nuance.

Problem 4

Feature Integration Complexity:
One of the major challenges was effectively integrating the probability outputs from the sentiment analysis model as input features into the disaster classification model. Ensuring that these probabilistic features contributed meaningful context, without introducing noise or redundancy, required careful design and tuning.

Thank you