DISASTER TWEET ANALYSIS

Mentor: Mr. Nitig Singh

MEET OUR TEAMS

TEAM 1
VANSHIKA (lead)
MOKSHA (co lead)
SRIRAM
KAUSHAL
KEERTHI

TEAM 2
PRAJAKTA (lead)
HANIFA (co lead)
BUSTOB
ESTHAK
KANCHANA
NEHA

TEAM 3
RANI SONI (lead)
GEETHA (co lead)
AVANTHIKA
KARTHIKEYA
RITHVIK

TEAM 4
ASHUTOSH (lead)
SHUBHAM (co lead)
KARNA
AYUSH
SHREEKAR

MOTIVATION

Image Credit : Gujarat Earthquake (2001)

Image Credit: Forest Fire – Brookings - 2012

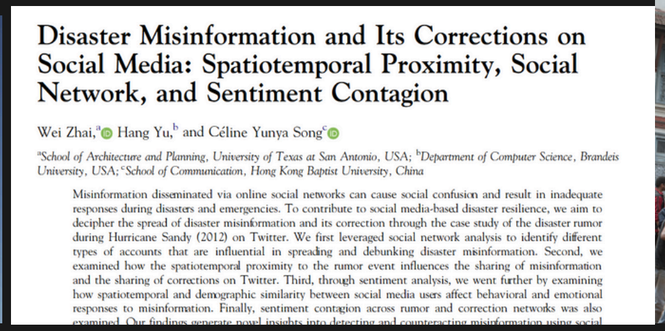
Image Credit: bhopal.org 2021

Image Credit: Mongabay - 2020

CASE STUDIES

Image credit: meetthematts.com - 2020

Fake images, like a storm cloud over the Statue of Liberty.


Disaster Misinformation and Its Corrections on Social Media: Spatiotemporal Proximity, Social Network, and Sentiment Contagion

Wei Zhai,[a] Hang Yu,[b] and Céline Yunya Song[c]

[a]School of Architecture and Planning, University of Texas at San Antonio, USA; [b]Department of Computer Science, Brandeis University, USA; [c]School of Communication, Hong Kong Baptist University, China

Misinformation disseminated via online social networks can cause social confusion and result in inadequate responses during disasters and emergencies. To contribute to social media-based disaster resilience, we aim to decipher the spread of disaster misinformation and its correction through the case study of the disaster rumor during Hurricane Sandy (2012) on Twitter. We first leveraged social network analysis to identify different types of accounts that are influential in spreading and debunking disaster misinformation. Second, we examined how the spatiotemporal proximity to the rumor event influences the sharing of misinformation and the sharing of corrections on Twitter. Third, through sentiment analysis, we went further by examining how spatiotemporal and demographic similarity between social media users affect behavioral and emotional responses to misinformation. Finally, sentiment contagion across rumor and correction networks was also

Image credit: huffingtonpost.com - 2015

Exaggerated death tolls, false reports of hospital collapses.

EXPLAINERS

The viral false claim that nearly 20 behind the Australia fires, explaine

Only a handful of fires were deliberately ignited. But that d spreading misinformation.

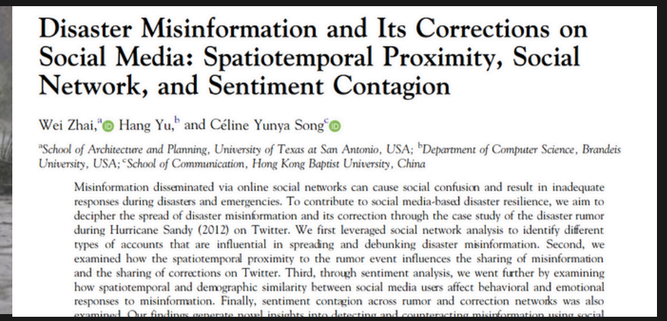
by **Umair Irfan**

Jan 10, 2020, 4:40 AM GMT+5:30

Image credit: vox.com- 2020

Conspiracy theories about fire causes (arson vs. climate factors)

Image credit: Hindustan Times- 2020


Disaster Misinformation and Its Corrections on Social Media: Spatiotemporal Proximity, Social Network, and Sentiment Contagion

Wei Zhai,[a] Hang Yu,[b] and Céline Yunya Song[c]

[a]School of Architecture and Planning, University of Texas at San Antonio, USA; [b]Department of Computer Science, Brandeis University, USA; [c]School of Communication, Hong Kong Baptist University, China

Misinformation disseminated via online social networks can cause social confusion and result in inadequate responses during disasters and emergencies. To contribute to social media-based disaster resilience, we aim to decipher the spread of disaster misinformation and its correction through the case study of the disaster rumor during Hurricane Sandy (2012) on Twitter. We first leveraged social network analysis to identify different types of accounts that are influential in spreading and debunking disaster misinformation. Second, we examined how the spatiotemporal proximity to the rumor event influences the sharing of misinformation and the sharing of corrections on Twitter. Third, through sentiment analysis, we went further by examining how spatiotemporal and demographic similarity between social media users affect behavioral and emotional responses to misinformation. Finally, sentiment contagion across rumor and correction networks was also

Claims of the cyclone as a “cover-up” for virus spread.

OBJECTIVES

To Develop a comprehensive system for detecting and analyzing disaster-related tweet

Integrate real and synthesized data by identifying disaster-related tweets from both authentic sources and synthesized datasets.

Handle imbalanced datasets by applying techniques like SMOTE and exploring other methods to improve model performance.

Extract key information such as location data and sentiment from tweets for in-depth analysis.

Deploy the model on a website, allowing real-time tweet analysis and providing users with insights on disaster-related conversations.

OVERVIEW

AIM

Develop an analytical tool to identify, categorize, and interpret disaster-related tweets.

APPROACH

Leverage NLP and ML for automatic classification and analysis.

SIGNIFICANCE

Enhance real-time situational awareness for informed decision-making.

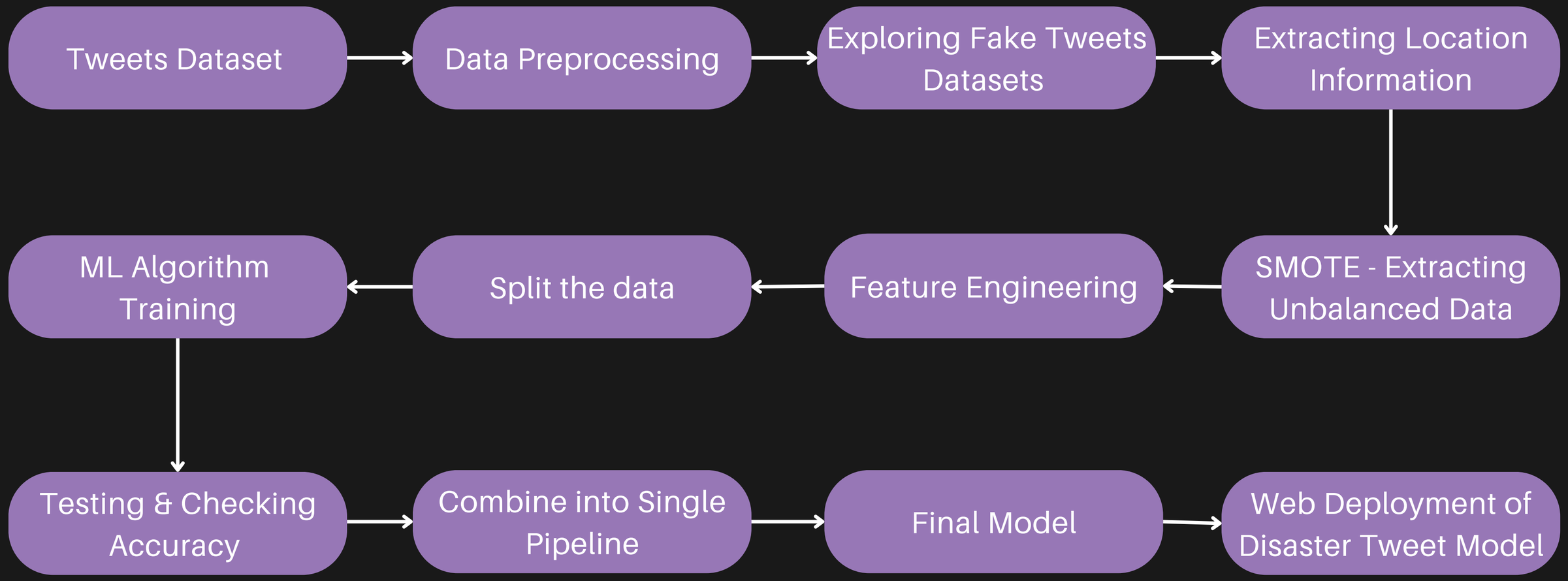
IMPACT

Supports disaster management, public health research, and community preparedness.

GOAL

Improve response strategies and foster resilience in affected communities.

WORKFLOW


Tweets Dataset
Data Preprocessing
Exploring Fake Tweets Datasets
Extracting Location Information
SMOTE - Extracting Unbalanced Data
Feature Engineering
Split the data
ML Algorithm Training
Testing & Checking Accuracy
Combine into Single Pipeline
Final Model
Web Deployment of Disaster Tweet Model

DATASET

- Kaggle Disaster Tweets dataset with 11370 tweets
- Tweets are labeled as disaster-related or non-disaster-related

FEATURES	**id:** Unique identifier for each tweet.
keyword: Highlighted word from the tweet	**location:** Origin of the tweet (if available).
text: The content of the tweet.	**target:** Binary variable (0 = non-disaster, 1 = disaster).

DATA INSIGHTS

1

Class Distribution

Dataset is imbalanced, often with more non-disaster tweets.

2

Tweet Length

Disaster tweets are longer than Non disaster tweets

3

Location Analysis

Most from locations in the United States, Australia and the UK.

4

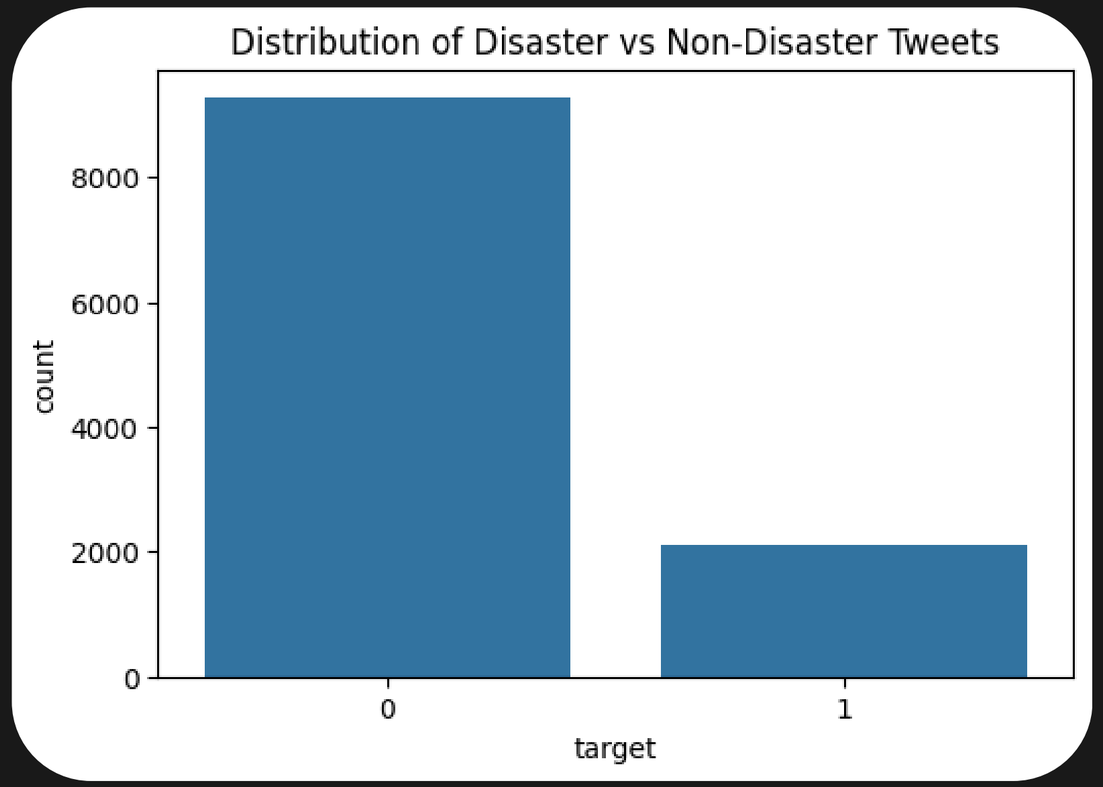
Keyword Analysis

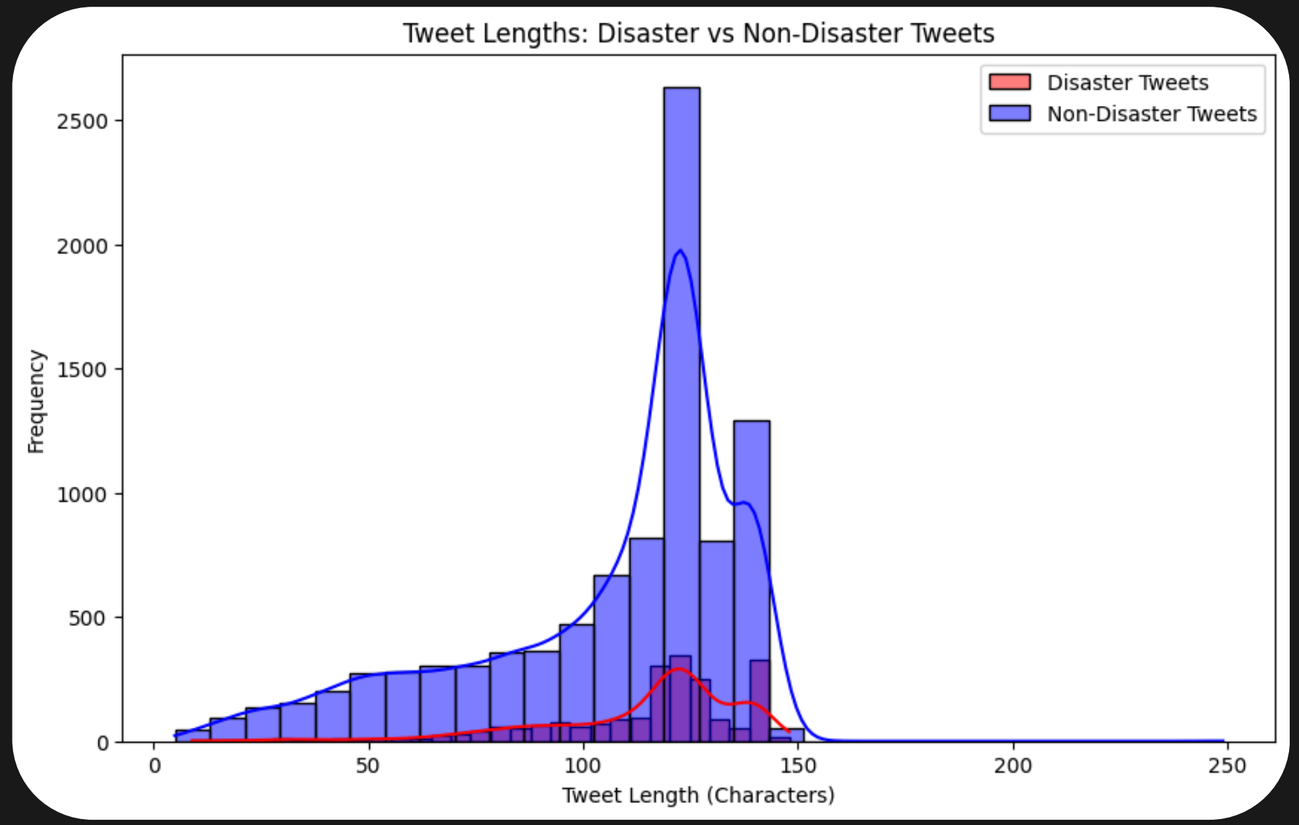
Disaster - "thunderstorm", "collision", "derailment".
Non-disaster- "drowning", "fear", and "obliterate".

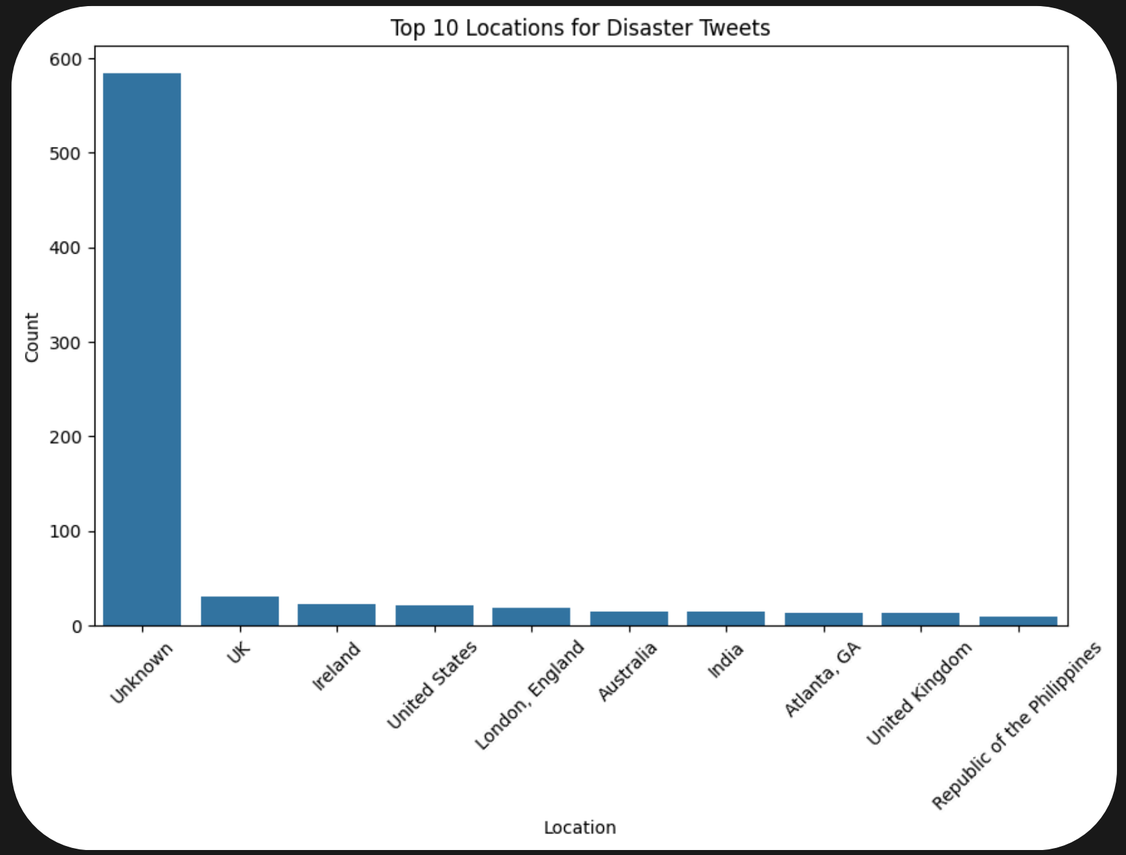
5

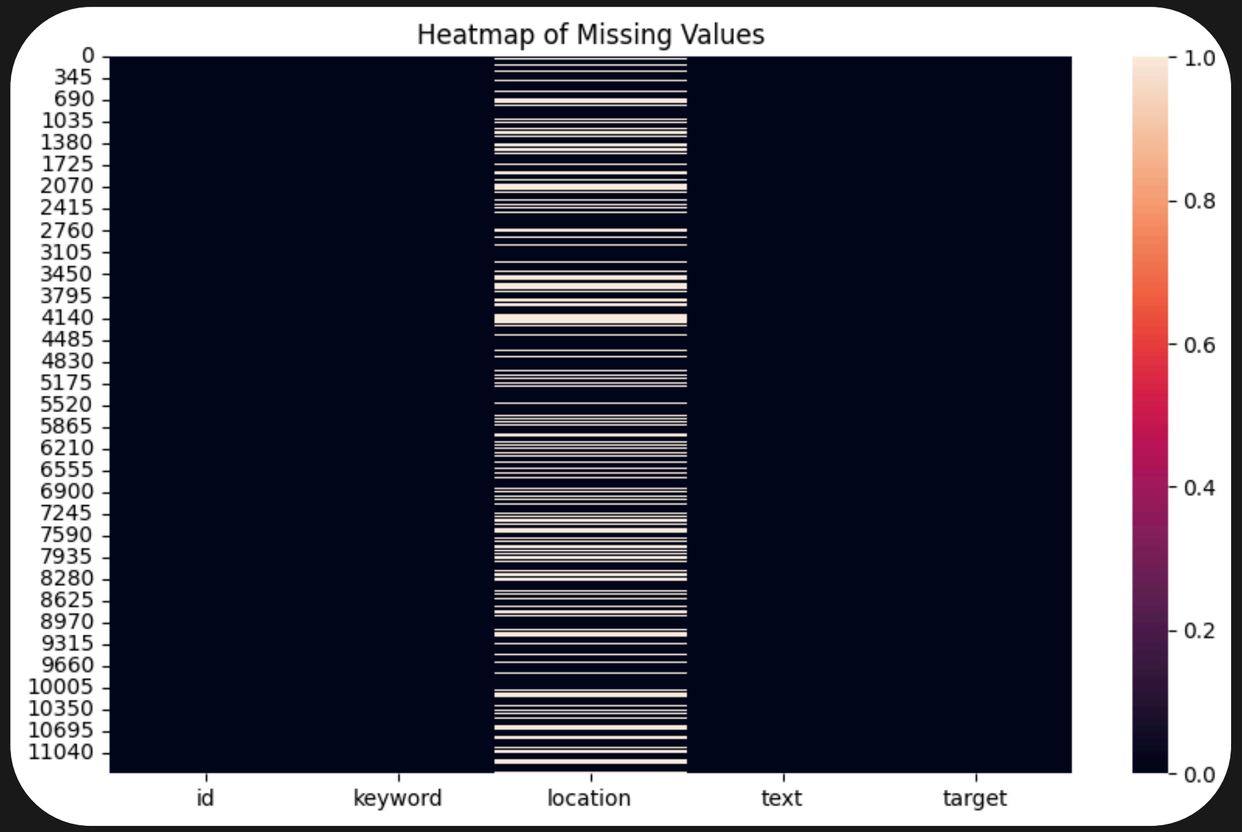
Missing Value in Location

3418 locations were missing


Distribution of Disaster vs Non-Disaster Tweets
count
8000
6000
4000
2000
0
0
1
target


Tweet Lengths: Disaster vs Non-Disaster Tweets
Disaster Tweets
Non-Disaster Tweets
Frequency
2500
2000
1500
1000
500
0
0
50
100
150
200
250
Tweet Length (Characters)


Top 10 Locations for Disaster Tweets
Count
600
500
400
300
200
100
0
Unknown
UK
Ireland
United States
London, England
Australia
India
Atlanta, GA
United Kingdom
Republic of the Philippines
Location


Heatmap of Missing Values
0
345
690
1035
1380
1725
2070
2415
2760
3105
3450
3795
4140
4485
4830
5175
5520
5865
6210
6555
6900
7245
7590
7935
8280
8625
8970
9315
9660
10005
10350
10695
11040
id
keyword
location
text
target
1.0
0.8
0.6
0.4
0.2
0.0

DATA CLEANING

1. Converting Text to Lowercase
2. Removing URLs
3. Removing Mentions and Hashtags
4. Expanding Contractions
5. Removing Non-English Characters
6. Handling Extra Spaces
7. Converting Emojis to Text
8. Removing Emojis from Location Data

DATA CLEANING

1. Converting Text to Lowercase
2. Removing URLs
3. Removing Mentions and Hashtags
4. Expanding Contractions
5. Removing Non-English Characters
6. Handling Extra Spaces
7. Converting Emojis to Text
8. Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

Remo

Original: "Communal violence in Bhainsa, Telangana. "Stones thrown!"
After Conversion: "communal violence in bhainsa, telangana. "stones thrown!"

5
Removing Non-English Characters

Removing Emojis from Location Data

Converting Emojis to Text

6
Handling Extra Spaces

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

2
Removin

- Original: "Arsonist sets cars ablaze at dealership https://t.co/xyz"
- After Removal: "Arsonist sets cars ablaze at dealership"

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

8
Removing Emojis from Location Data

- Original: "Floods affecting lives @RescueTeam #floodalert"
- After Removal: "Floods affecting lives"

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

1
Convert
Low

3
ving Mentions
d Hashtags

4
Expanding
Contractions

5
Removing Non-
English Characters

Original: "We can’t leave the flooded area."
After Expansion: "We cannot leave the flooded area."

8
Removing Emojis
from Location Data

7
Converting Emojis to
Text

6
Handling Extra
Spaces

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

3
Removing Mentions

4
Expanding Contractions

5
Removing Non-English Characters

- Original:"地震 has struck, stay safe everyone! #Earthquake"
- After Processing:"has struck, stay safe everyone! #Earthquake"

8
Removing Emojis
Location Data

7
Converting Emojis to Text

6
Handling Extra Spaces

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URL

- Original: " Flood warnings issued in Delhi. "
- After Handling: "Flood warnings issued in Delhi."

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

Conv
L

entions
tags

4
Expanding Contractions

5
Removing Non-English Characters

- Original: "Rescue operation ongoing in the flood zone👍✨."
- After Conversion: "Rescue operation ongoing helicopter in the flood zone."

8
moving Emojis from Location Data

7
Converting Emojis to Text

6
Handling Extra Spaces

DATA CLEANING

1
Converting Text to Lowercase

2
Removing URLs

3
Removing Mentions and Hashtags

4
Expanding Contractions

5
Removing Non-English Characters

6
Handling Extra Spaces

7
Converting Emojis to Text

8
Removing Emojis from Location Data

DATA CLEANING

Con

- Original Location: "Delhi📍"
- After Removal: "Delhi"

3
oving Mentions
nd Hashtags

4
Expanding
Contractions

5
Removing Non-
English Characters

8
Removing Emojis
from Location Data

7
Converting Emojis to
Text

6
Handling Extra
Spaces

INDIVIDUAL TEAM WORK

TEAM 1

Exploring Twitter API s

TEAM 2

Exploring Fake Tweets Dataset

TEAM 3

Extract location information and create visual plots

TEAM 4

Handling unbalanced datasets

Exploring Twitter API s

METHODOLOGY

WHAT DISASTER MANAGEMENT TEAMS DO

Prepare, respond, and recover from natural or man-made disasters.

Coordinate resources and communicate vital information for public safety.

Implement early warnings, evacuations, and rescues.

Provide immediate relief to affected communities.

Utilize real-time data and technology to minimize disaster impact.

METHODOLOGY

Integration of real time data with our model

1

Created a Twitter Developer Account to access the API.

2

Registered an App on witter Developer Portal to generate API keys and tokens

3

Configured API keys and bearer tokens in the development environment.

4

Used Twitter API v2 endpoints to scrape disaster-related tweets.

5

Implemented a data preprocessing pipeline to clean and filter tweets.

CHALLENGES FACED

CHALLENGES FACED USING TWITTER API

Restricted access to certain features due to limited authorization.

Higher access level needed for real-time data usage and scraping.

CHALLENGES FACED USING 3RD PARTY API

Both RapidAPI and AnyAPI only allow retrieval of 25 tweets at a time, which is insufficient for comprehensive analysis.

Irrelevant Data: The fetched tweets lack relevance to disaster-related topics, providing data that does not align with the project's focus.

RESULTS

Real-Time Data Integration:
Unable to include real-time data due to restrictions with Twitter API access.

Third-Party API Limitations:
Utilized third-party APIs, but data retrieval was limited and often irrelevant to disaster topics.

Future Scope:
Plan to pursue alternative sources to achieve real-time data integration.

Exploring Fake Tweets Dataset

EXPLORING FAKE TWEETS PROJECTS ON KAGGLE

Challenges with Using Kaggle Dataset:

- Inconsistent Data Formats
- Limited Disaster-Specific Tweets
- Incomplete Metadata
- Quality and Authenticity Concerns
- Licensing or Usage Restrictions

SIMULATED TWEETS FOR REAL / FAKE DATASET

Decision to Create a Custom Dataset:

Why We Chose to Create Our Own Dataset:

- To ensure data quality, relevance, and specific disaster focus.
- Control over data format, metadata, and labeling for consistency.

Benefits of a Custom Dataset

- Tailored for disaster analysis and streamlined for model training.
- More control over data structure and metadata, enabling accurate analysis.

SIMULATED DATASET DESIGN:

Text

This will contain the content of the tweets, which could be generated using a text generator or from real-world examples

Location

This can be a random location generated for each fake tweet or assigned based on patterns

Keyword

Choose disaster-related keywords such as "earthquake", "flood" and assign them to the fake tweets, either randomly or based on the nature of the tweet content.

Target 1

(Disaster or Non-Disaster)

Target 2

(Real or Fake)

CREATING FAKE TWEET DATASET

Text Generation

- Libraries like Faker used to generate fake tweets text. Customize the Faker library to generate content that resembles typical tweet formats (short, concise, and containing hashtags or mentions).

Location and Keywords

- Randomly assign location values (e.g., city names or coordinates) using a list of disaster-prone locations or randomly generated names.
- Attach relevant disaster-related keywords to the fake tweets or randomly select them based on the content's nature.

Labeling

- Set Target 1 (Disaster or Non-Disaster) to "Non-Disaster" for all fake tweets.
- Set Target 2 (Fake or Real) to "Fake" for all fake tweets.

COMBINING FAKE TWEET DATASET WITH EXISTING DATA

Generated Fake Tweets Dataset

Load Both DataSets
- Existing Dataset
- Created Dataset

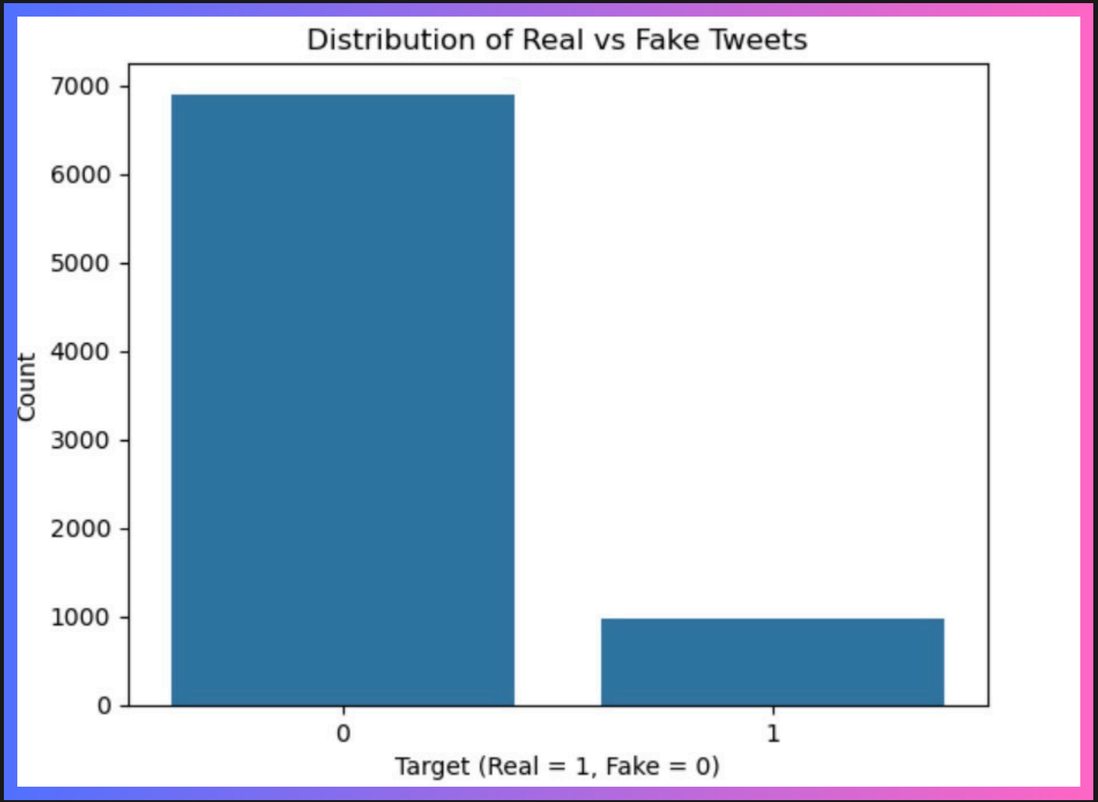
Prepare Existing Disaster Dataset

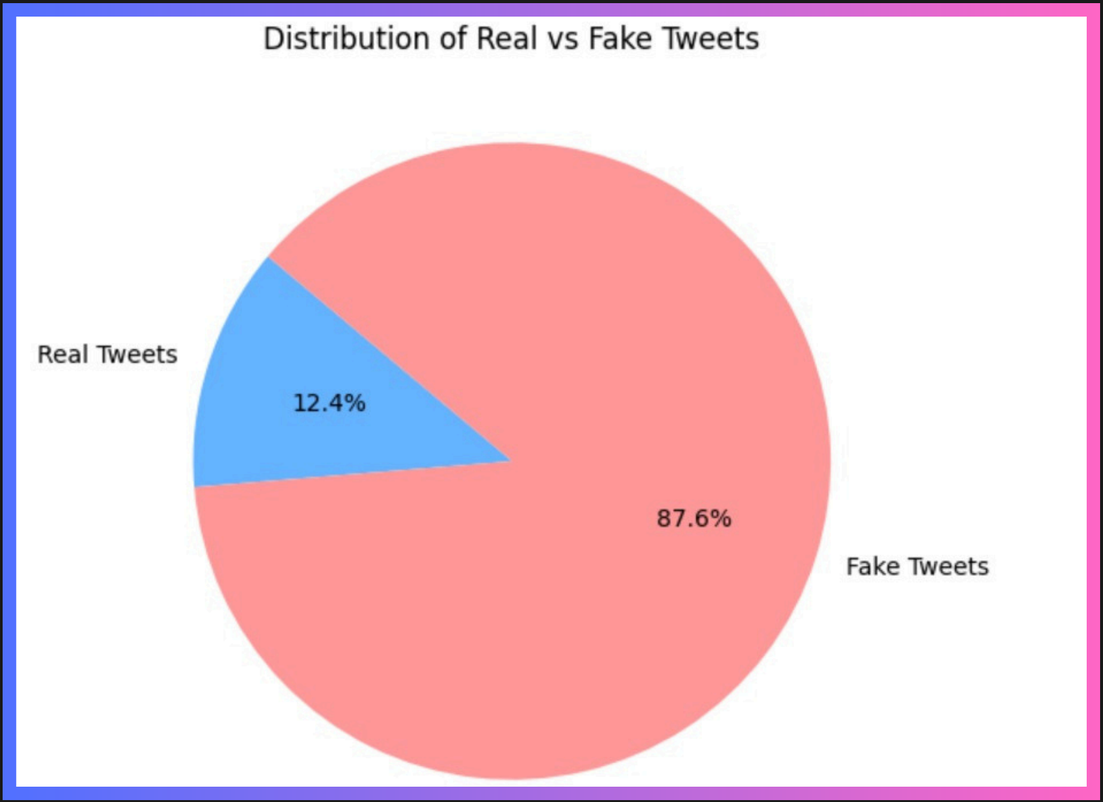
Ensure both datasets have the same structure:
Text, Location, Keyword, Target 1, target 2 .

Merge Both Dataset

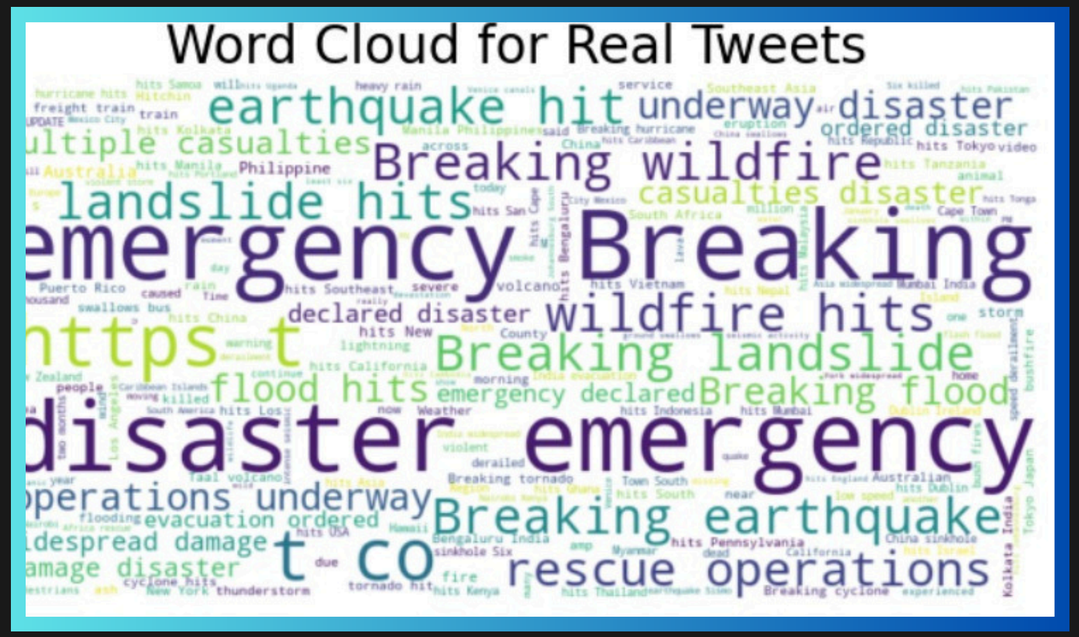
FINAL DATASET

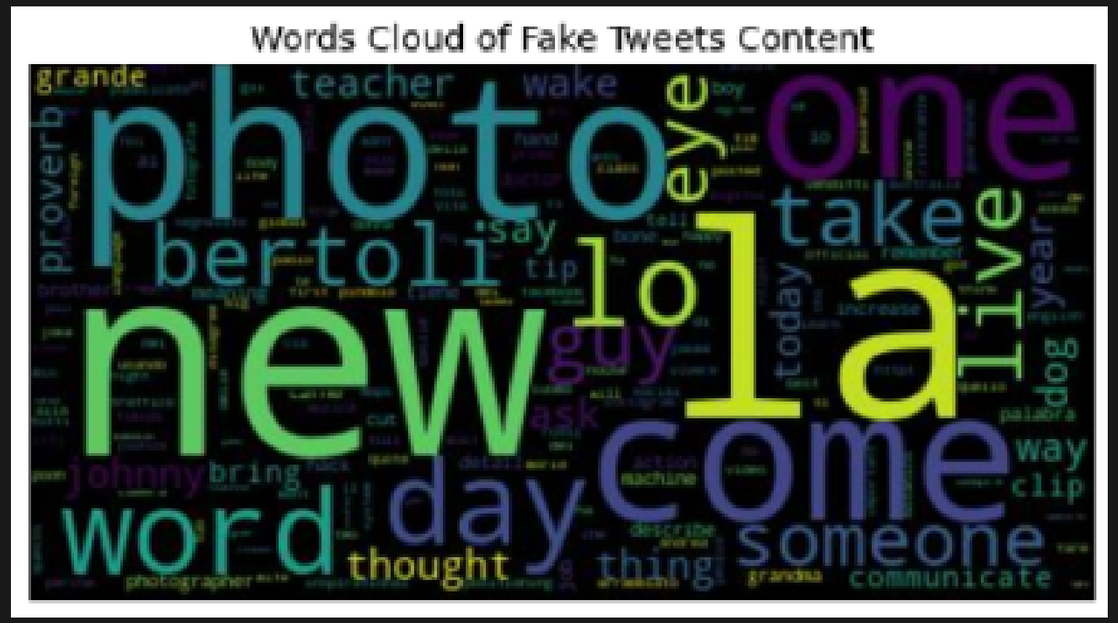
DISTRIBUTION OF REAL VS FAKE TWEETS





COMPARING WORD CLOUDS FOR FAKE AND REAL TWEETS





COMPARISON CHART

dex	Model	Vectorization	Accuracy	Classification Report
0	Logistic Regression	TF-IDF	0.8869832893579596	{'0': {'precision': 0.8913568324480927, 'recall': 0.9829605963791267, 'f1-score': 0.9349202329703723, 'support': 1878.0}, '1': {'precision': 0.8423645320197044, 'recall': 0.4318181818181818, 'f1-score': 0.5709515859766278, 'support': 396.0}, 'accuracy': 0.8869832893579596, 'macro avg': {'precision': 0.8668606822338986, 'recall': 0.7073893890986542, 'f1-score': 0.7529359094735, 'support': 2274.0}, 'weighted avg': {'precision': 0.8828251917402468, 'recall': 0.8869832893579596, 'f1-score': 0.8715378300638099, 'support': 2274.0}}
1	Decision Tree	TF-IDF	0.8337730870712401	{'0': {'precision': 0.8997867803837953, 'recall': 0.898828541001065, 'f1-score': 0.8993074054342035, 'support': 1878.0}, '1': {'precision': 0.5226130653266332, 'recall': 0.5252525252525253, 'f1-score': 0.5239294710327456, 'support': 396.0}, 'accuracy': 0.8337730870712401, 'macro avg': {'precision': 0.7111999228552143, 'recall': 0.7120405331267952, 'f1-score': 0.7116184382334745, 'support': 2274.0}, 'weighted avg': {'precision': 0.8341048141733133, 'recall': 0.8337730870712401, 'f1-score': 0.8339381609210209, 'support': 2274.0}}
2	Random Forest	TF-IDF	0.8940193491644679	{'0': {'precision': 0.8948383984563435, 'recall': 0.9877529286474973, 'f1-score': 0.9390027841052898, 'support': 1878.0}, '1': {'precision': 0.8855721393034826, 'recall': 0.4494949494949495, 'f1-score': 0.5963149078726968, 'support': 396.0}, 'accuracy': 0.8940193491644679, 'macro avg': {'precision': 0.890205268879913, 'recall': 0.7186239390712235, 'f1-score': 0.7676588459889933, 'support': 2274.0}, 'weighted avg': {'precision': 0.8932247491051857, 'recall': 0.8940193491644679, 'f1-score': 0.8793262673998778, 'support': 2274.0}}
3	K-Nearest Neighbors	TF-IDF	0.8518029903254177	{'0': {'precision': 0.8488003621548211, 'recall': 0.9984025559105432, 'f1-score': 0.9175434303890384, 'support': 1878.0}, '1': {'precision': 0.9538461538461539, 'recall': 0.15656565656565657, 'f1-score': 0.26898047722342733, 'support': 396.0}, 'accuracy': 0.8518029903254177, 'macro avg': {'precision': 0.9013232580004875, 'recall': 0.5774841062380999, 'f1-score': 0.5932619538062329, 'support': 2274.0}, 'weighted avg': {'precision': 0.8670932968556864, 'recall': 0.8518029903254177, 'f1-score': 0.8046010691517552, 'support': 2274.0}}
4	Support Vector Classifier	TF-IDF	0.8966578715919086	{'0': {'precision': 0.897820823244552, 'recall': 0.987220447284345, 'f1-score': 0.9404007101191986, 'support': 1878.0}, '1': {'precision': 0.8851674641148325, recall': 0.4671717171717172, 'f1-score': 0.6115702479338843, 'support': 396.0}, 'accuracy': 0.8966578715919086, 'macro avg': {'precision': 0.8914941436796923, 'recall': 0.7271960822280311, 'f1-score': 0.7759854790265415, 'support': 2274.0}, 'weighted avg': {'precision': 0.8956173359027011, 'recall': 0.8966578715919086, 'f1-score': 0.8831373578652917, 'support': 2274.0}}
5	Multinomial	TF-IDF	0.8808267370272648	{'0': {'precision': 0.8828013339685564, 'recall': 0.9866879659211928, 'f1-score': 0.931858184561227, 'support': 1878.0}, '1': {'precision': 0.8571428571428571, 'recall': 0.3787878787878788, 'f1-score': 0.5253940455341506, 'support': 396.0}, 'accuracy': 0.8808267370272648, 'macro avg': {'precision': 0.8699720955557068, 'recall': 0.6827379223545358, 'f1-score': 0.7286261150476888, 'support': 2274.0}, 'weighted avg': {'precision': 0.8783331031756906, 'recall': 0.8808267370272648, 'f1-score': 0.86107551127419, 'support': 2274.0}}
6	GaussianNB	TF-IDF	0.7088830255057168	{'0': {'precision': 0.9216366158113731, 'recall': 0.707667731629393, 'f1-score': 0.8006024096385542, 'support': 1878.0}, '1': {'precision': 0.3401442307692308, 'recall': 0.7146464646464646, 'f1-score': 0.4609120521172638, 'support': 396.0}, 'accuracy': 0.7088830255057168, 'macro avg': {'precision': 0.630890423290302, 'recall': 0.7111570981379288, 'f1-score': 0.630757230877909, 'support': 2274.0}, 'weighted avg': {'precision': 0.8203740896562771, 'recall': 0.7088830255057168, 'f1-score': 0.7414478882760076, 'support': 2274.0}}
7	Logistic Regression	Count Vectorization	0.8979771328056289	{'0': {'precision': 0.9186164801627671, 'recall': 0.9616613418530351, 'f1-score': 0.9396462018730489, 'support': 1878.0}, '1': {'precision': 0.7662337662337663, 'recall': 0.5959595959595959, 'f1-score': 0.6704545454545454, 'support': 396.0}, 'accuracy': 0.8979771328056289, 'macro avg': {'precision': 0.8424251231982667, 'recall': 0.7788104689063156, 'f1-score': 0.8050503736637972, 'support': 2274.0}, 'weighted avg': {'precision': 0.8920801764178751, 'recall': 0.8979771328056289, 'f1-score': 0.8927684991722014, 'support': 2274.0}}
8	Decision Tree	Count Vectorization	0.8469656992084432	{'0': {'precision': 0.9082177161152615, 'recall': 0.906283280085197, 'f1-score': 0.9072494669509595, 'support': 1878.0}, '1': {'precision': 0.56, 'recall': 0.5656565656565656, 'f1-score': 0.5628140703517588, 'support': 396.0}, 'accuracy': 0.8469656992084432, 'macro avg': {'precision': 0.7341088580576307, 'recall': 0.7359699228708814, 'f1-score': 0.7350317686513592, 'support': 2274.0}, 'weighted avg': {'precision': 0.8475782193775114, 'recall': 0.8469656992084432, 'f1-score': 0.8472686327146871, 'support': 2274.0}}
9	Random Forest	Count Vectorization	0.8940193491644679	{'0': {'precision': 0.895219700627716, 'recall': 0.987220447284345, 'f1-score': 0.9389718916181312, 'support': 1878.0}, '1': {'precision': 0.8817733990147784, 'recall': 0.45202020202020204, 'f1-score': 0.5976627712854758, 'support': 396.0}, 'accuracy': 0.8940193491644679, 'macro avg': {'precision': 0.8884965498212472, 'recall': 0.7196203246522735, 'f1-score': 0.7683173314518035, 'support': 2274.0}, 'weighted avg': {'precision': 0.8928781283151729, 'recall': 0.8940193491644679, 'f1-score': 0.87953547488474, 'support': 2274.0}}

Extract location information and create visual plots

LOCATION EXTRACTION PROCESS

1
Data Preprocessing

Preprocess the tweet data to clean and prepare for location extraction

2
Top Locations Extraction

Identify the most frequently mentioned locations in tweets.

3
Latitude and Longitude Extraction

use Geopy for geolocation and obtain latitude and longitude.

4
Visualization

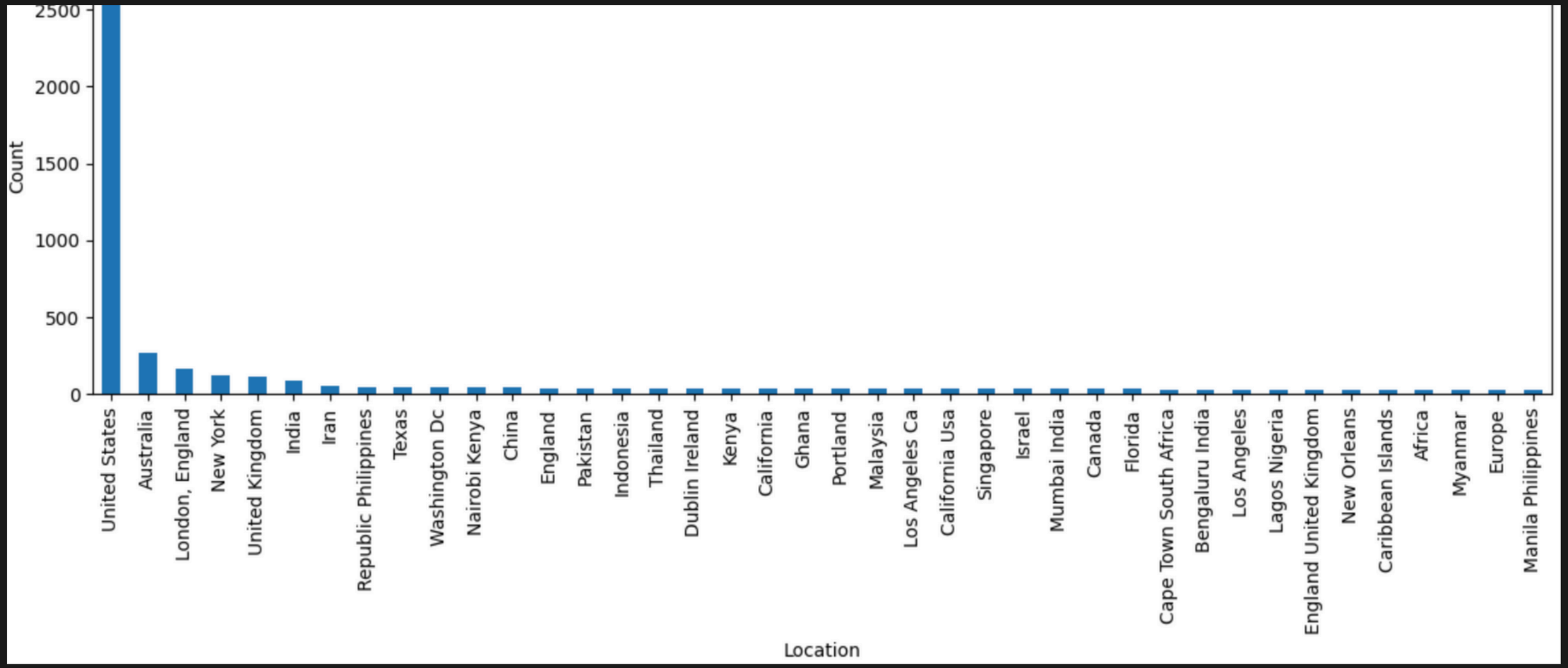
use tweet counts for normalized visualization and coordinates for density representation.

5
Global and Country-Specific Mapping

Plotly function to create map of tweet locations with color scaling based on tweet count.

VISUALIZATION OF TOP 40 TWEET LOCATIONS

We identified the top 40 locations mentioned in tweets by using the value_counts() function on the location field in our dataset.

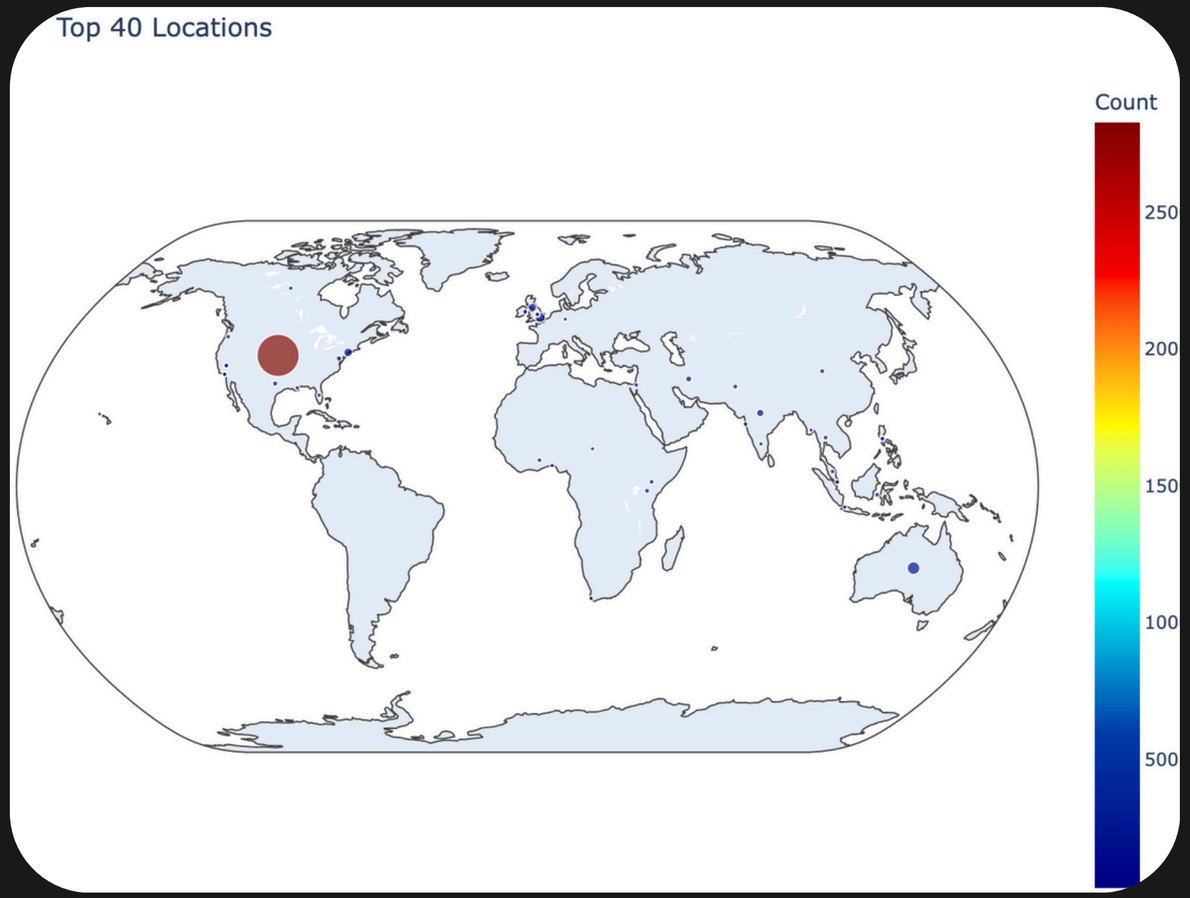


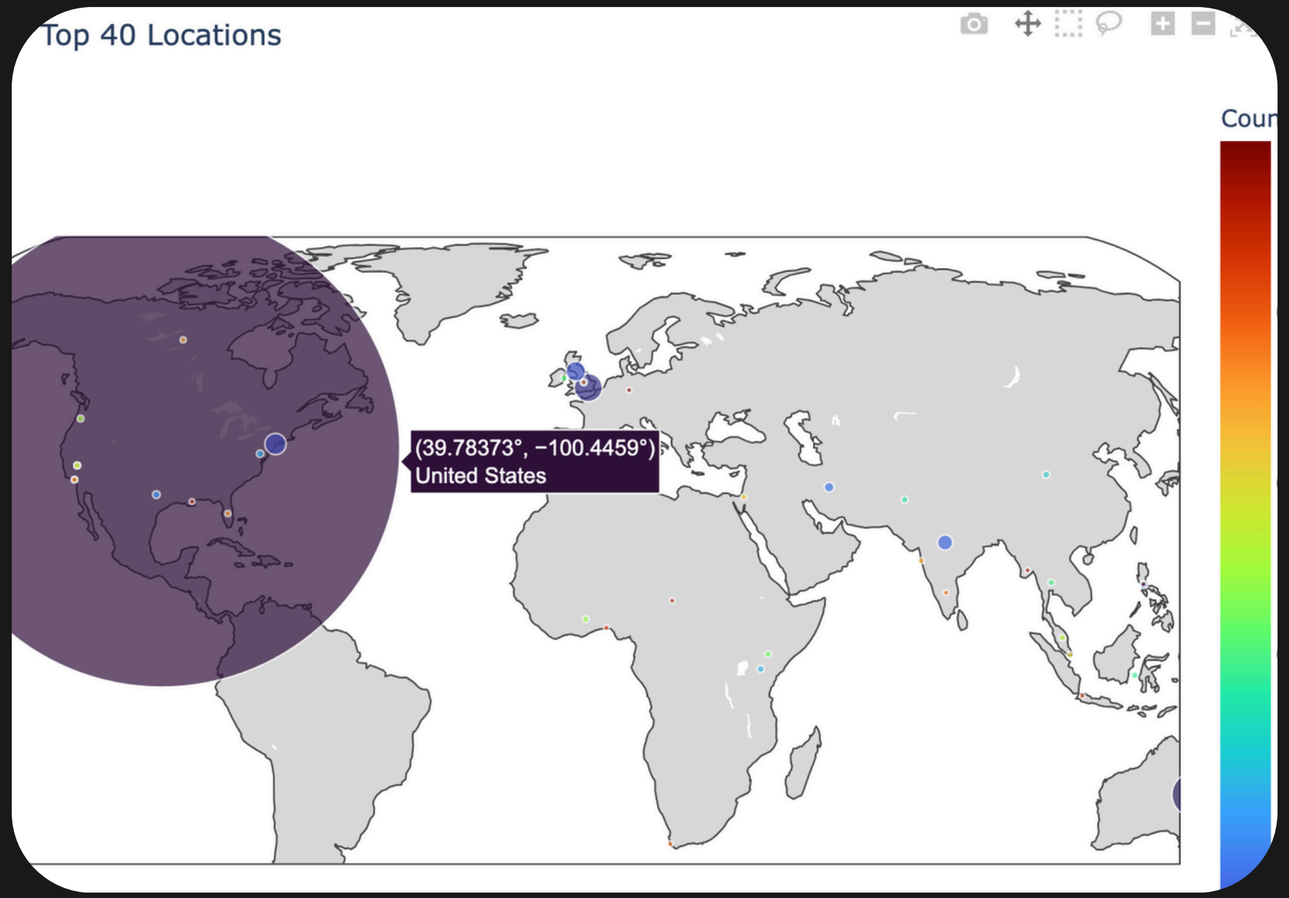
MAPPING LOCATIONS - LATITUDE AND LONGITUDE EXTRACTION

- To place each location on a map, we used the Nominatim API to convert location names into geographic coordinates (latitude and longitude).
- This involved creating a function that retrieved coordinates for each location, handling any errors by setting unavailable data to None.
- With this step, we prepared the data for global mapping.

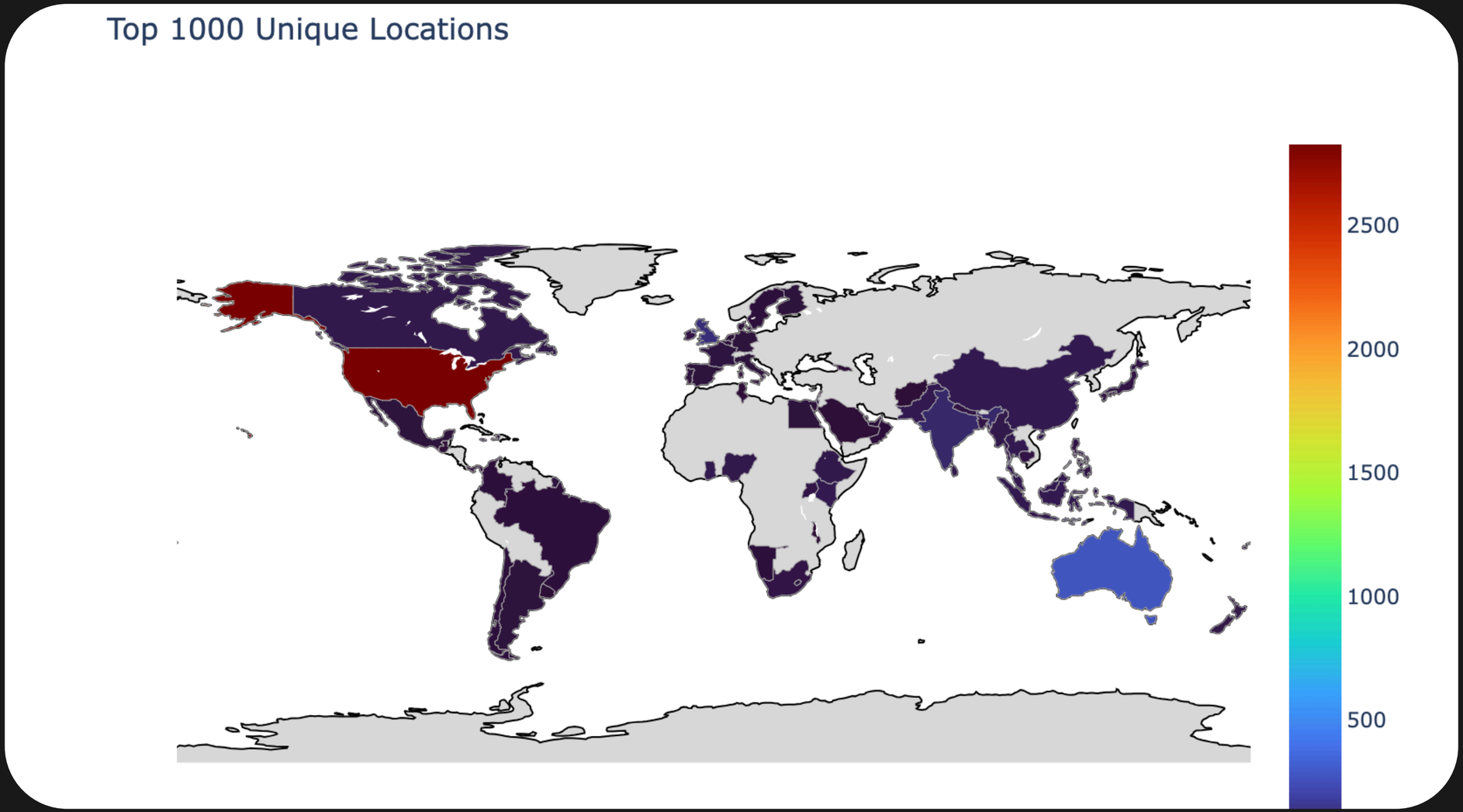
```
# Function to get latitude and longitude
def get_coordinates(location):
    try:
        return geolocator.geocode(location).point[0:2]
    except:
        return [None, None]
```

DENSITY MAP OF TOP 40 LOCATIONS

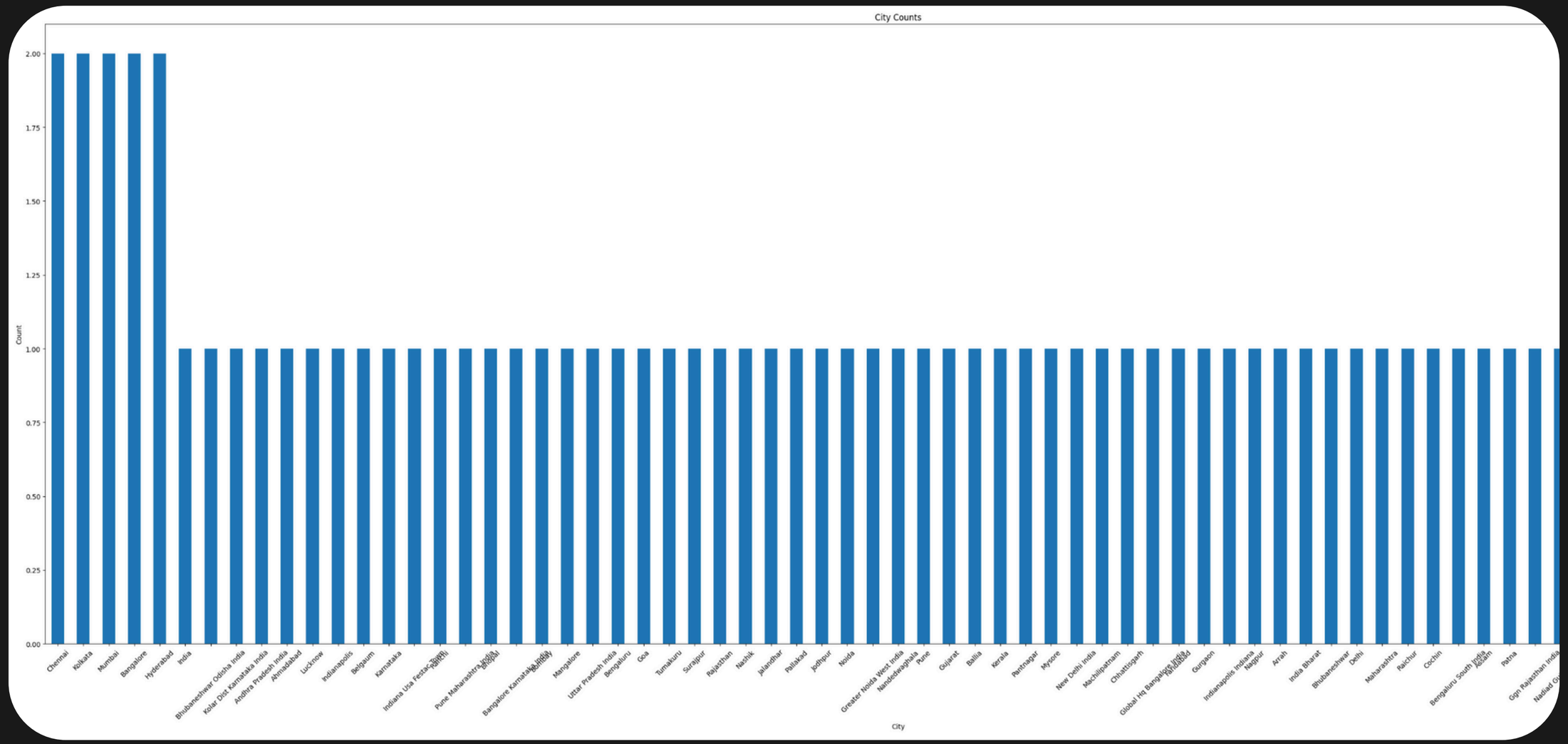




DENSITY MAP OF TOP 1000 LOCATIONS

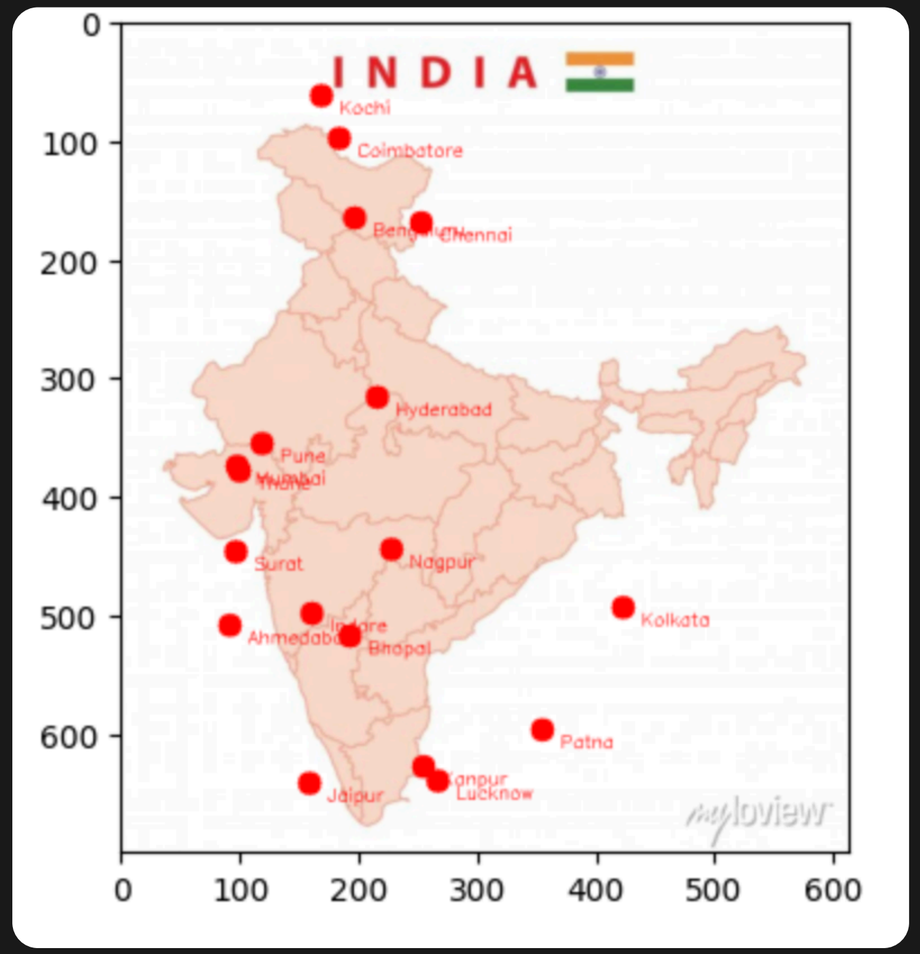


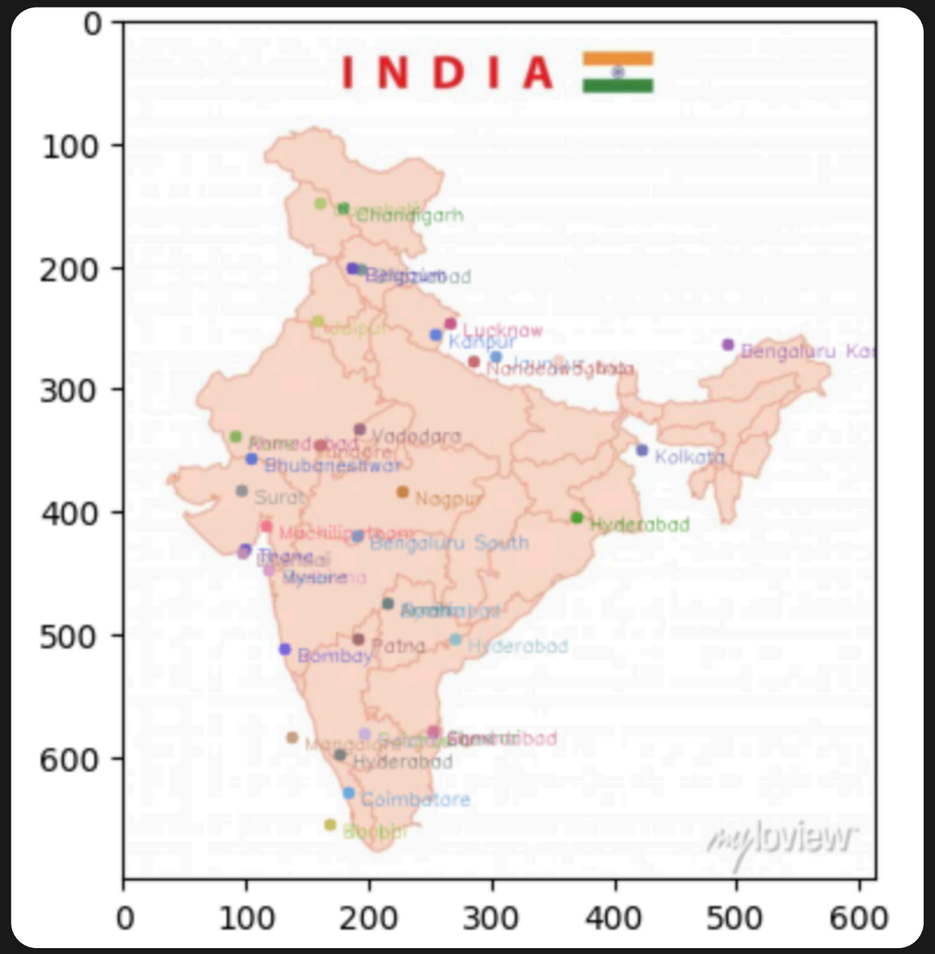
VISUALIZATION OF TWEET LOCATIONS IN INDIA



COUNTRY-SPECIFIC MAPPING (INDIA)

we focused on specific countries, such as India, by mapping major cities with high tweet activity on an outline map.





Handling unbalanced datasets

HANDLING IMBALANCE - SMOTE

KEY PROBLEM

non-disaster tweets often outnumber disaster tweets, leading to a class imbalance
Imbalance can cause models lowering accuracy in identifying disaster-related tweets.

SMOTE is a resampling technique that generates synthetic samples for the minority class.
enhancing model accuracy without simply duplicating data

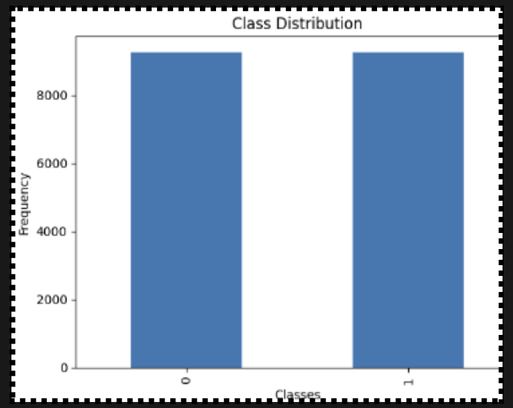
KEY SOLUTION

HANDLING IMBALANCE - SMOTE

Random Oversampling of the Minority Class

- Increase disaster tweets by duplicating samples.

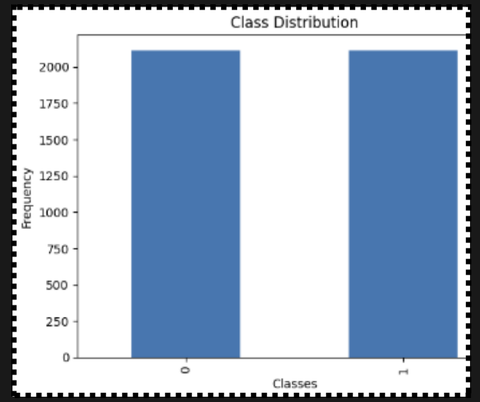
```
 target
0     9256
1     9256
```



Random Undersampling of the Majority Class (Non-Disaster Tweets)

- Reduce non-disaster tweets

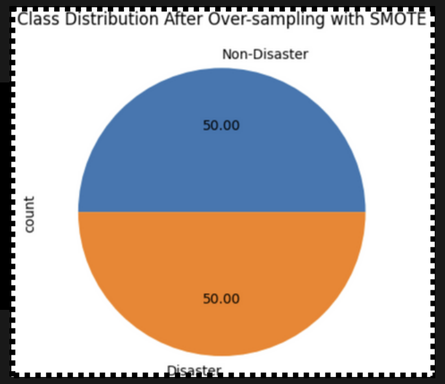
```
 target
0     2114
1     2114
```



Using SMOTE for Class Balancing

- Generate synthetic disaster tweets for better class balance.

```
 target
0     9256
1     9256
```



RESAMPLING TECHNIQUES

SMOTE-NC

Borderline SMOTE

SMOTE-ENN

SMOTE TOMEK

ADASYN (Adaptive Synthetic Sampling)

CHALLENGES FACED

PROBLEM:

The initial attempts using simple B-o-W representation resulted in a sparse matrix.

SOLUTION:
Used Term Frequency-Inverse Document Frequency (TF-IDF)

PROBLEM:

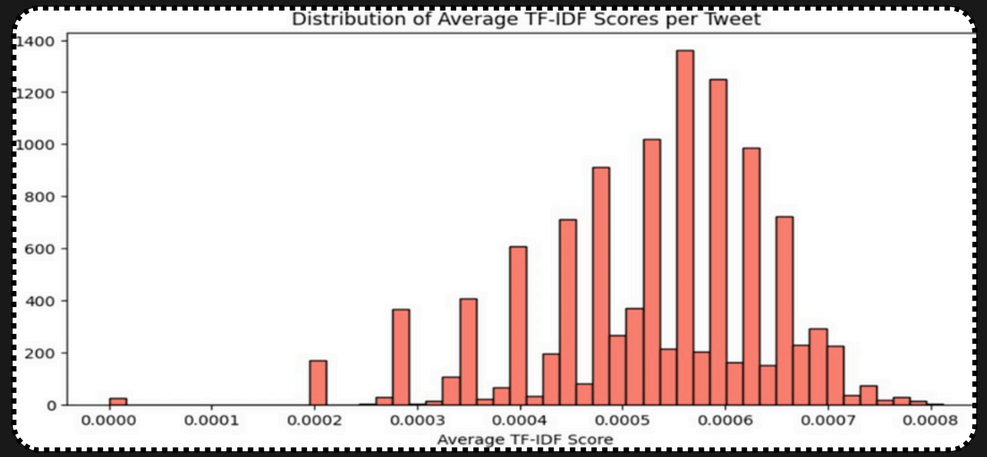
High-dimensional, sparse vectors from TF-IDF can hinder ML model performance.

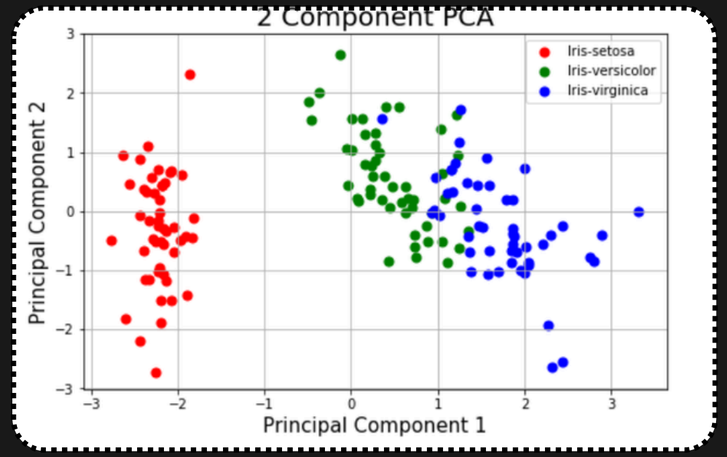
Solution:
Dimensionality Reduction
PCA: Reduces feature space, retaining variance.

Shape of Bag of Words matrix: (11370, 22632)

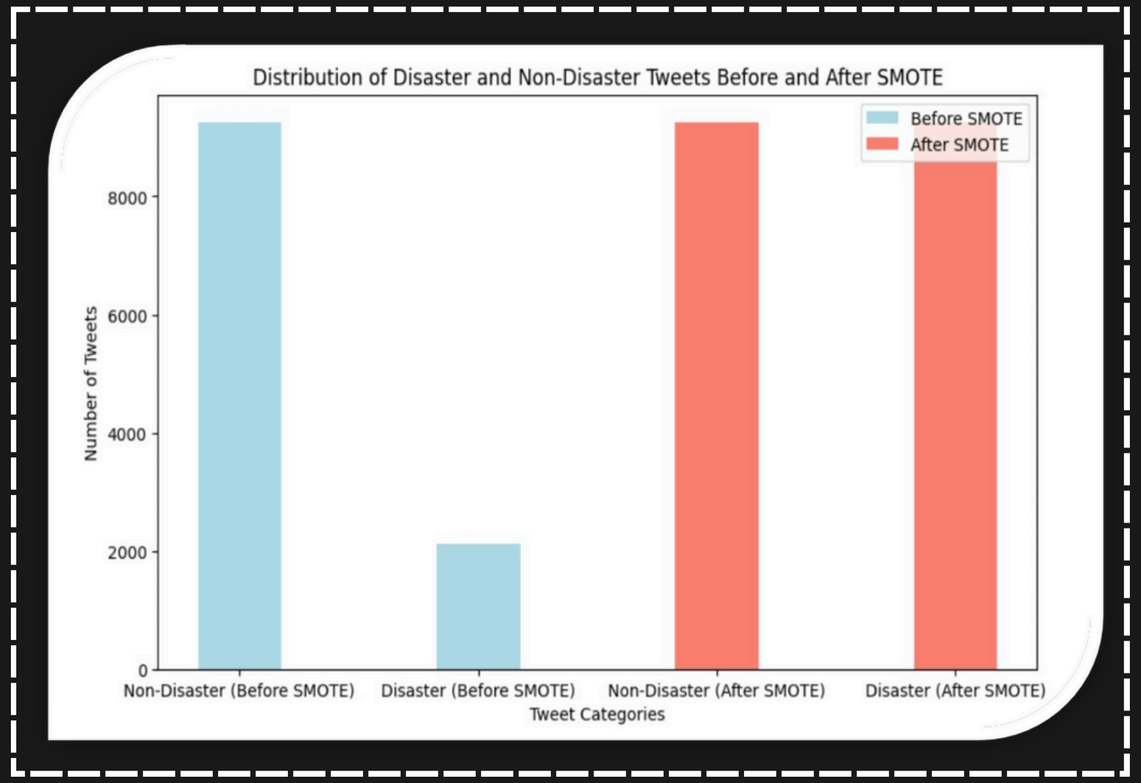
	000009	0019	007	0075c	01	0100	0100z
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0

5 rows x 22632 columns

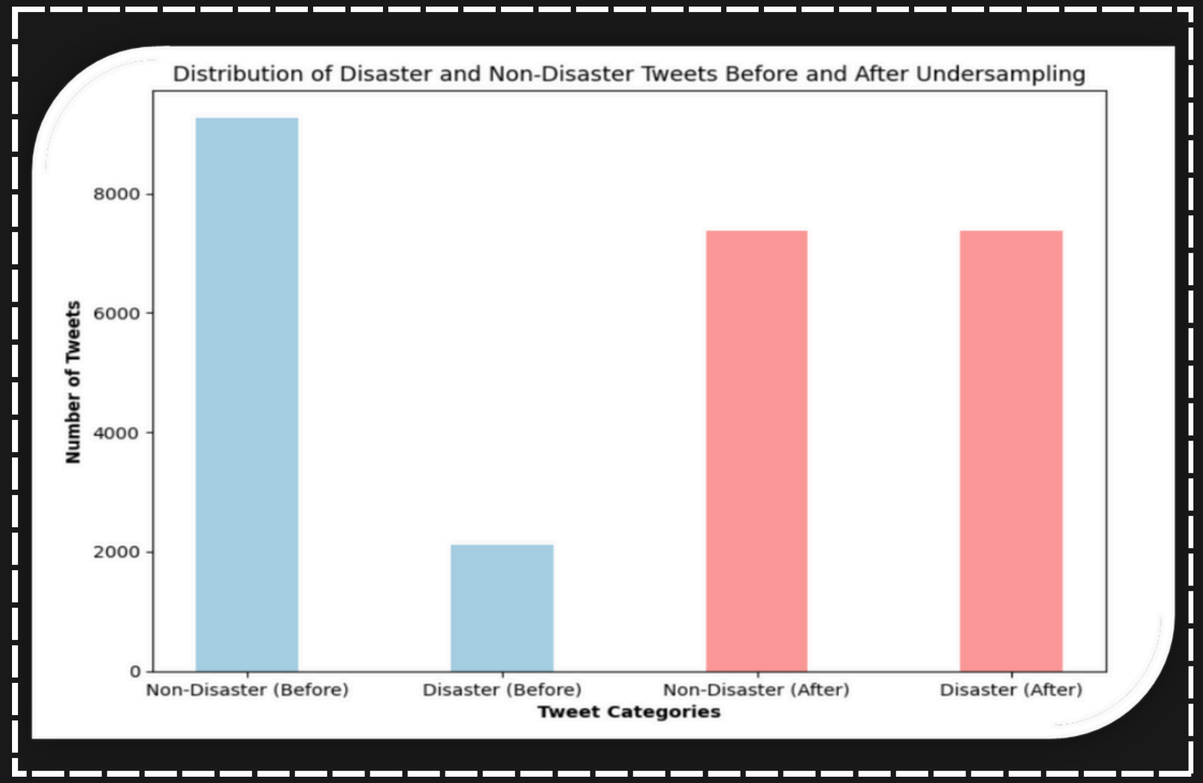




IMPLEMENTATION VISUALS

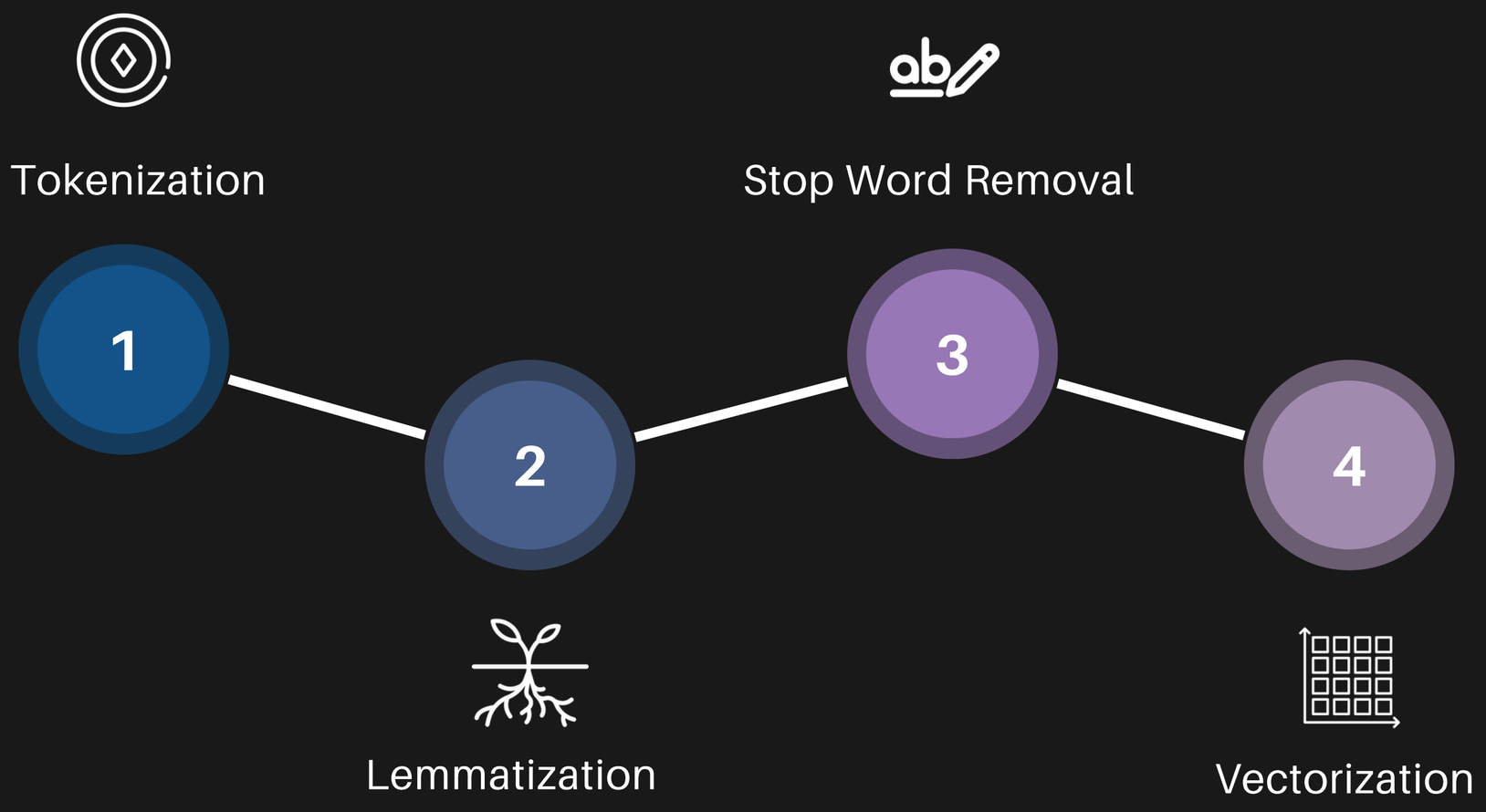


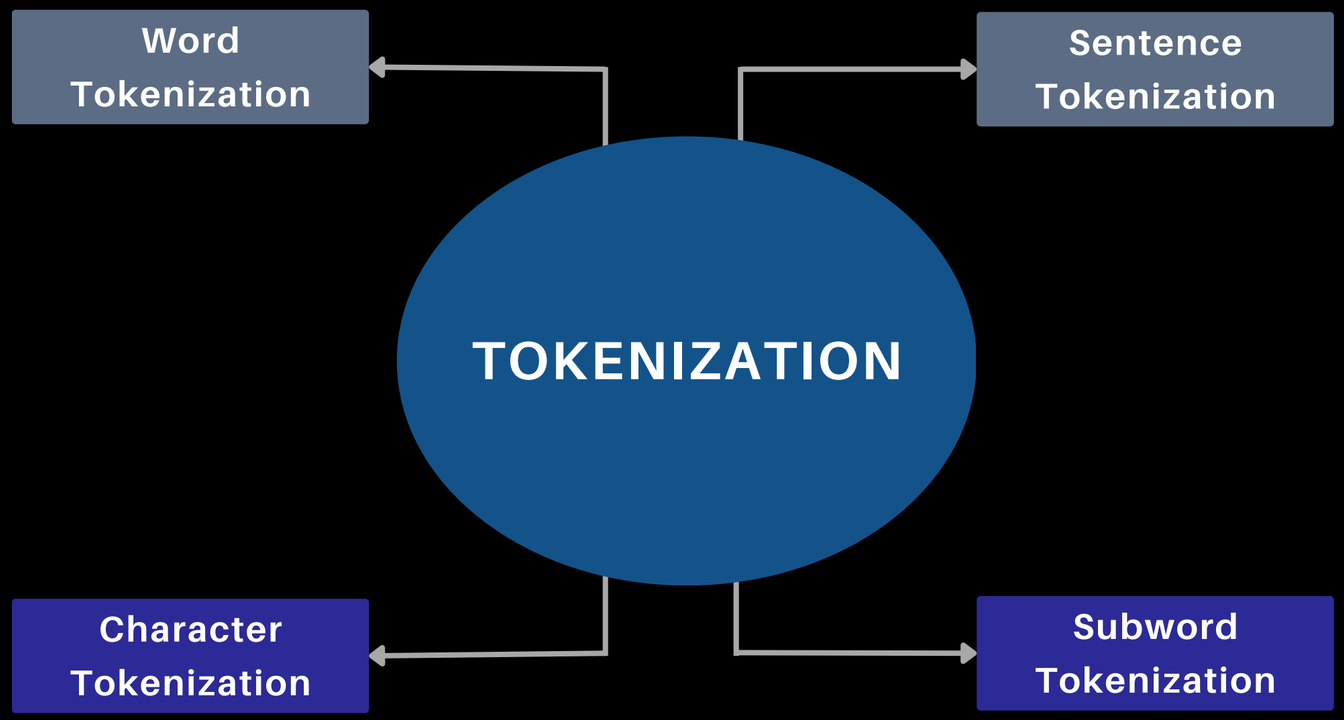
SMOTE OVERSAMPLING

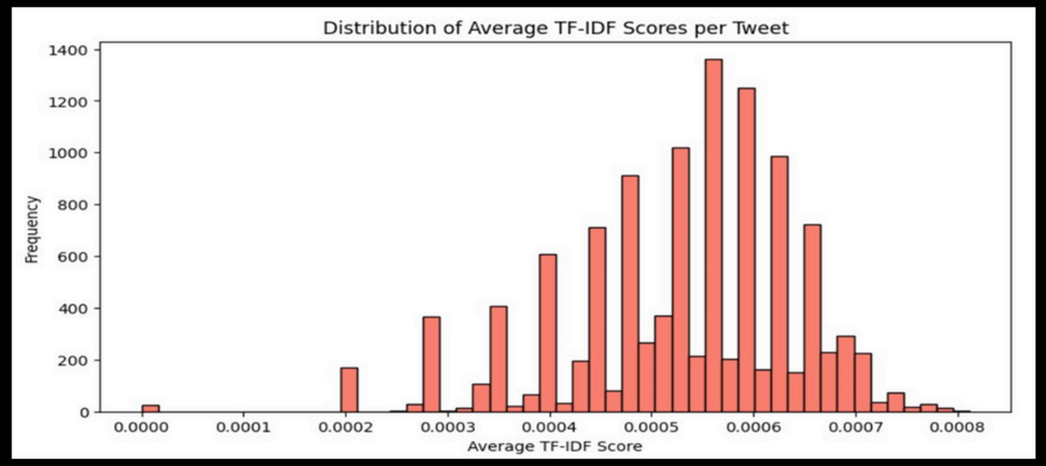


SMOTE UNDERSAMPLING

FEATURE ENGINEERING







Sentence Tokens:

- 'several houses have been set ablaze in ngemsibaa village, oku sub division in the north west region of cameroon by'

Subword Tokens:

- 'several', 'houses', 'have', 'been', 'set', 'ab', '##laze', 'in', 'ng',

Character Tokens:

- 's', 'e', 'v', 'e', 'r', 'a', 'l', ' ', 'h', 'o', 'u', 's', 'e', 's', ' ', 'h', 'a', 'v', 'e', ' ', 'b',

Word Tokens:

- 'several', 'houses', 'have', 'been', 'set', 'ablaze', 'in',

VECTORIZATION

- TF-IDF
- BAG OF N-GRAMS
- BERT
- WORD2VEC

WORD2VEC

```
id,keyword,location,processed_text,target,sentence_tokens,subword_toke
0,ablaze,,"communal violence in bhainsa, telangana. ""stones were pelt
1,ablaze,,"telangana: section 144 has been imposed in bhainsa from jar
2,ablaze,New York City,arsonist sets cars ablaze at dealership,1,['ars
3,ablaze,"Morgantown, WV",arsonist sets cars ablaze at dealership,1,['
4,ablaze,,"""lord jesus, your love brings freedom and pardon. fill me
5,ablaze,OC,"if this child was chinese, this tweet would have gone vir
6,ablaze,"London, England","several houses have been set ablaze in nge
7,ablaze,Bharat,asansol: a bjp office in salanpur village
8,ablaze,"Accra, Ghana","national security minister, kan da
9,ablaze,Searching,this creature whos soul is no longer clarent but bl
10,ablaze,,images showing the havoc caused by the cameroon military as
11,ablaze,,social media went bananas after chuba hubbard announced mor
12,ablaze,,hausa youths set area office of apapa-iganmu local council
13,ablaze,HYDERABAD,under mamatabanerjee political violence &amp;amp; vanc
```

Loading...

```
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
-0.98501897,-0.3655197,-0.30583307,-0.636
-0.598374,-0.8015745,0.06097548,-0.416222
-0.37299496,-0.059177224,-0.36862668,-0.0
-0.37299496,-0.059177224,-0.36862668,-0.0
-0.10406582,0.15094003,0.13687484,-0.3300
0.09985293,-0.10447973,-0.14152475,0.1067
```

BERT

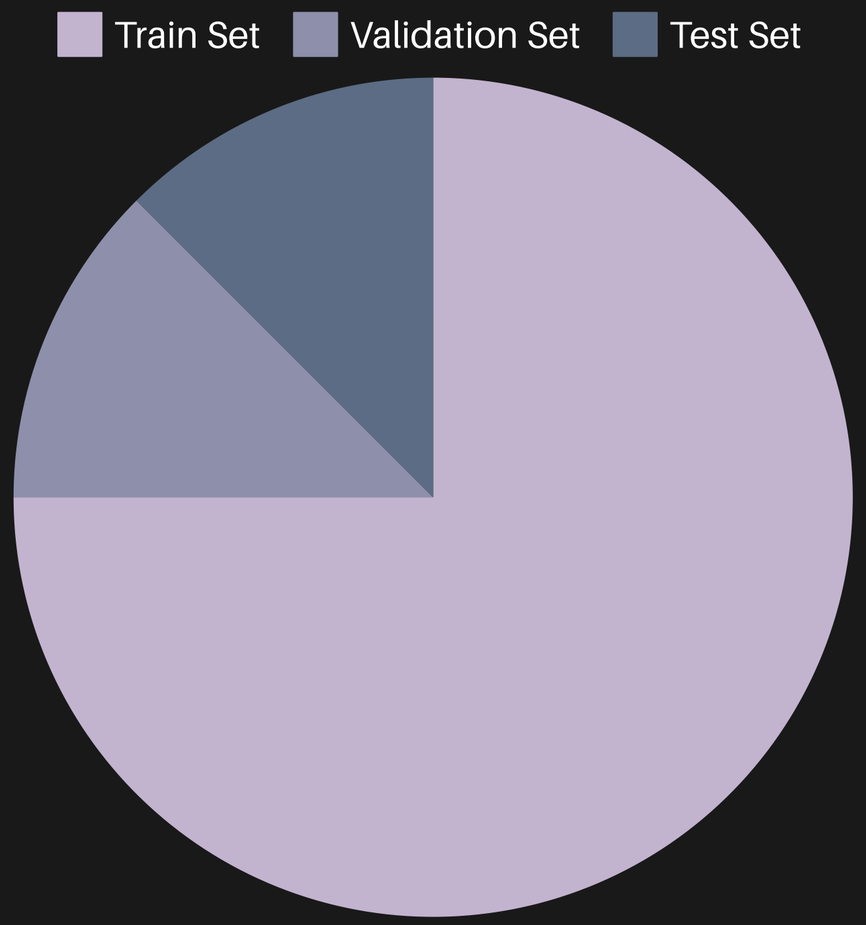
```
id,keyword,location,processed_text,target,sentence_tokens,subword_toke
0,ablaze,,"communal violence in bhainsa, telangana. ""stones were pelt
1,ablaze,,"telangana: section 144 has been imposed in bhainsa from jar
2,ablaze,New York City,arsonist sets cars ablaze at dealership,1,['ars
3,ablaze,"Morgantown, WV",arsonist sets cars ablaze at dealership,1,['
4,ablaze,,"""lord jesus, your love brings freedom and pardon. fill me
5,ablaze,OC,"if this child was chinese, this tweet would have gone vir
6,ablaze,"London, England","several houses have been set ablaze in nge
7,ablaze,Bharat,asansol: a bjp office in salanpur village
8,ablaze,"Accra, Ghana","national security minister, kan da
9,ablaze,Searching,this creature whos soul is no longer clarent but bl
10,ablaze,,images showing the havoc caused by the cameroon military as
11,ablaze,,social media went bananas after chuba hubbard announced mor
12,ablaze,,hausa youths set area office of apapa-iganmu local council
13,ablaze,HYDERABAD,under mamatabanerjee political violence &amp;amp; vanc
```

Loading...

```
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
-0.17601049,0.3743985,0.6399543,0.1109
-0.07351632,0.35403916,0.6302984,0.145
-0.038881037,0.2585514,0.4217361,0.094
-0.038881037,0.2585514,0.4217361,0.094
-0.13608195,0.26144803,0.69879144,0.24
```

MODEL TRAINING

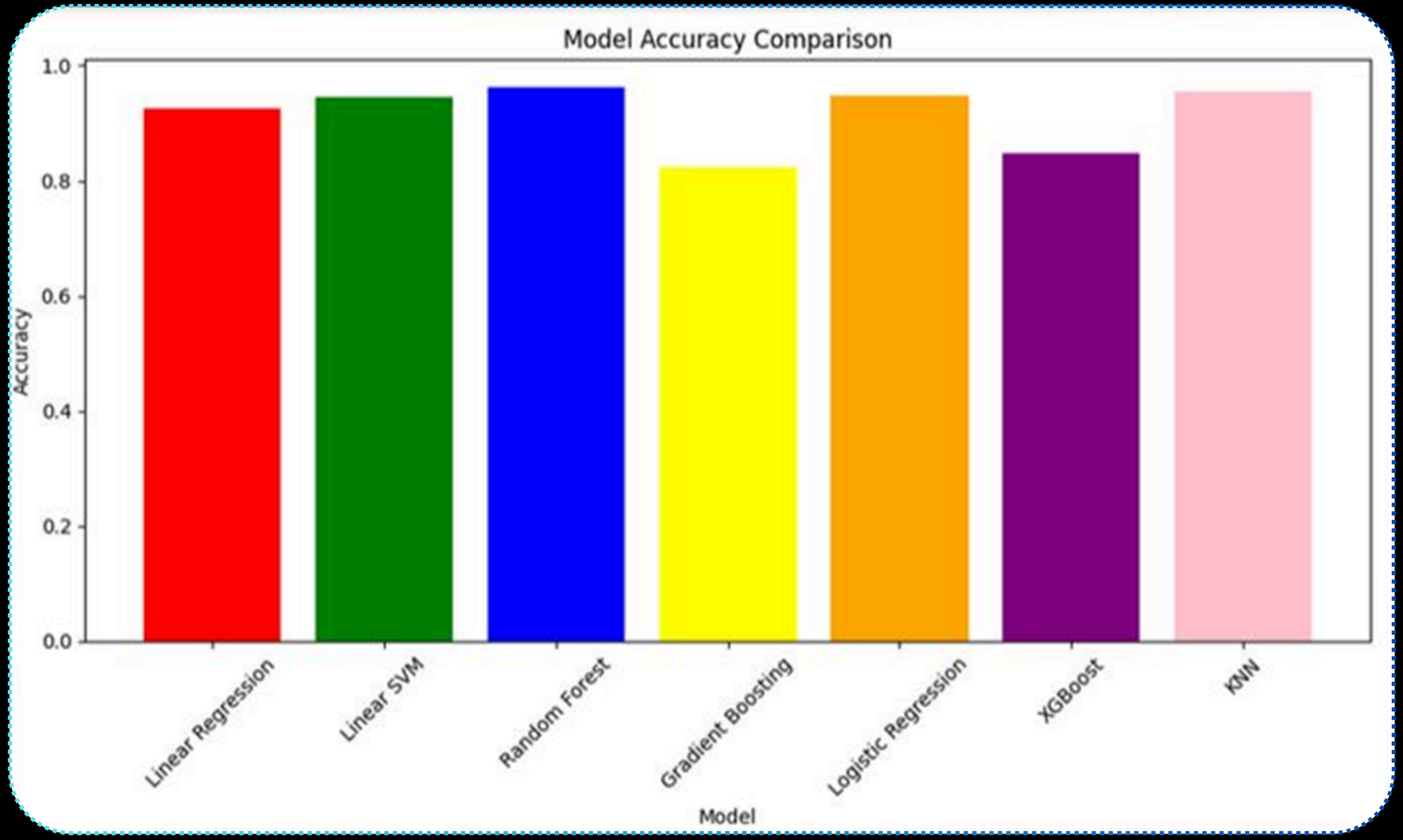
- Training: Learns patterns from data.
- Validation: Prevents overfitting and optimizes performance.
- Test: Checks model generalization on unseen data.



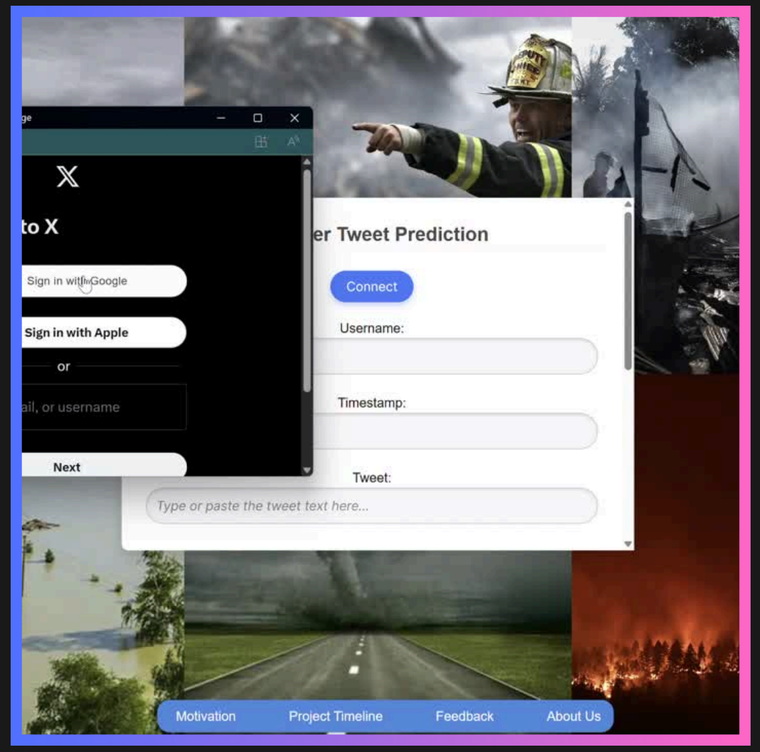
MODELS TRAINED

- Logistic Regression
- Random Forest
- Decision Tree
- XGBoost
- KNN
- Linear SVM
- Naive Bayes
- Support Vector
- Linear Regression

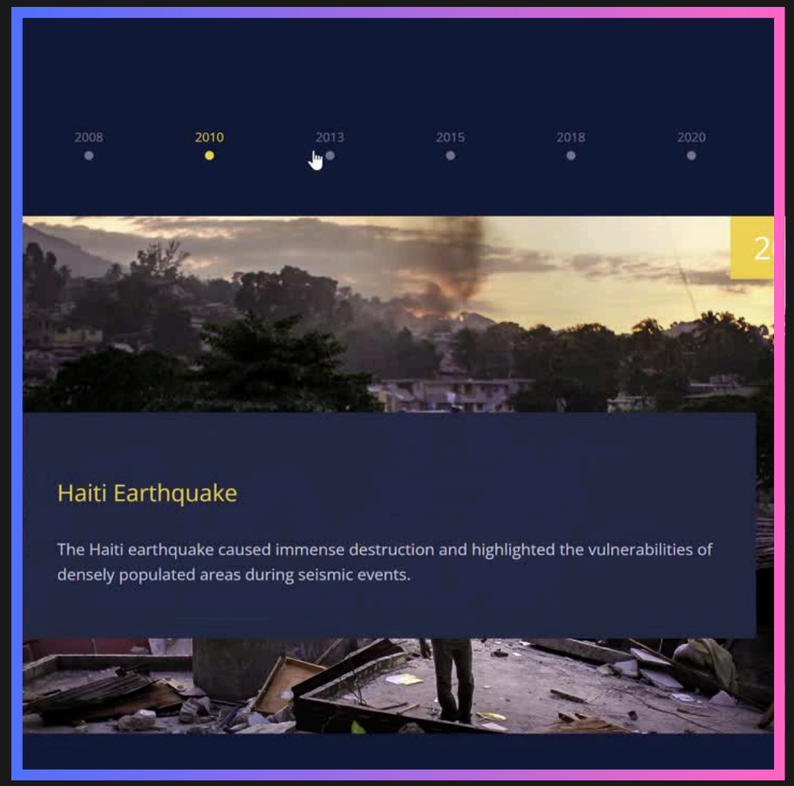
MODEL ACCURACY



WEB INTERFACE DEMO

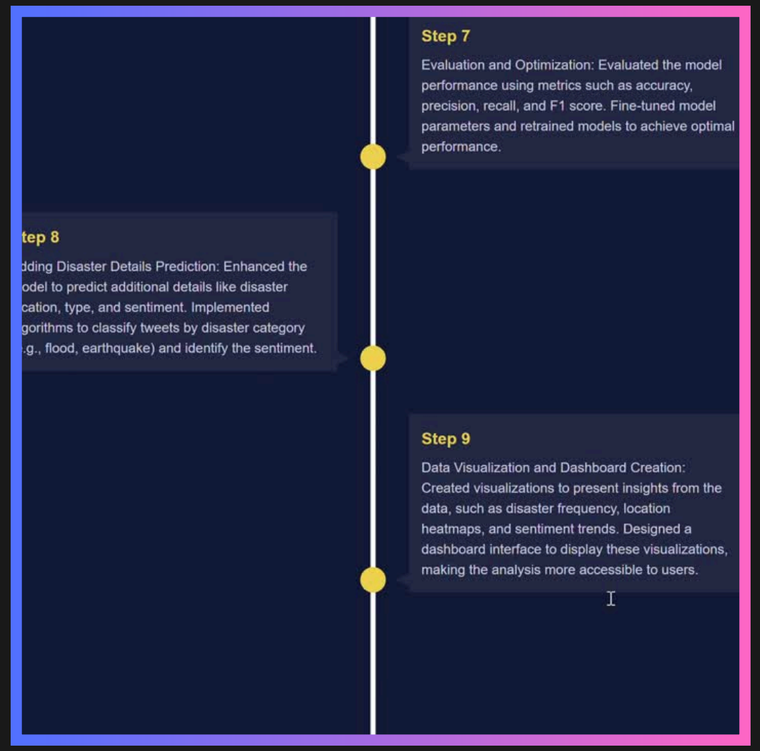


HOME PAGE (PREDICTION)

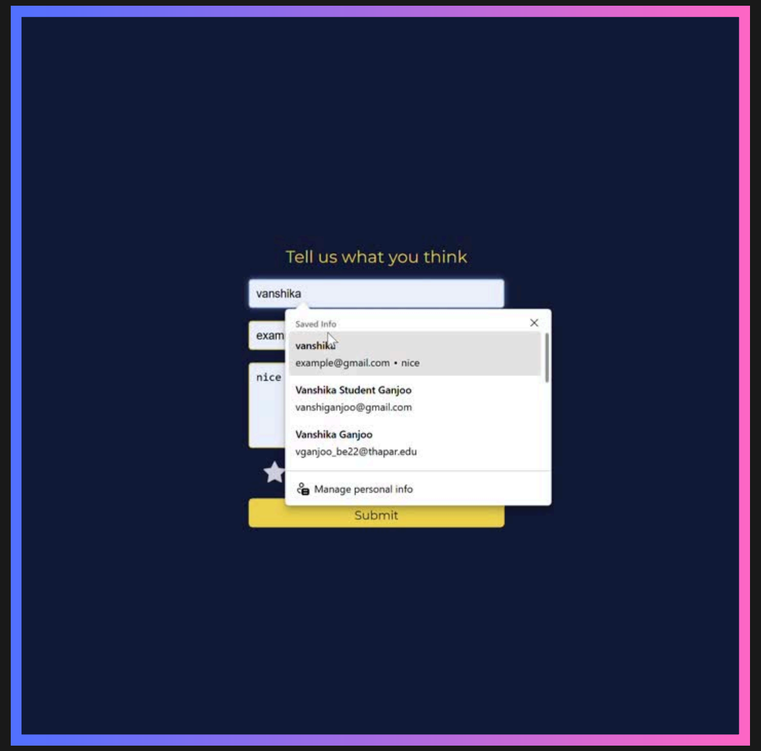


MOTIVATION

WEB INTERFACE DEMO

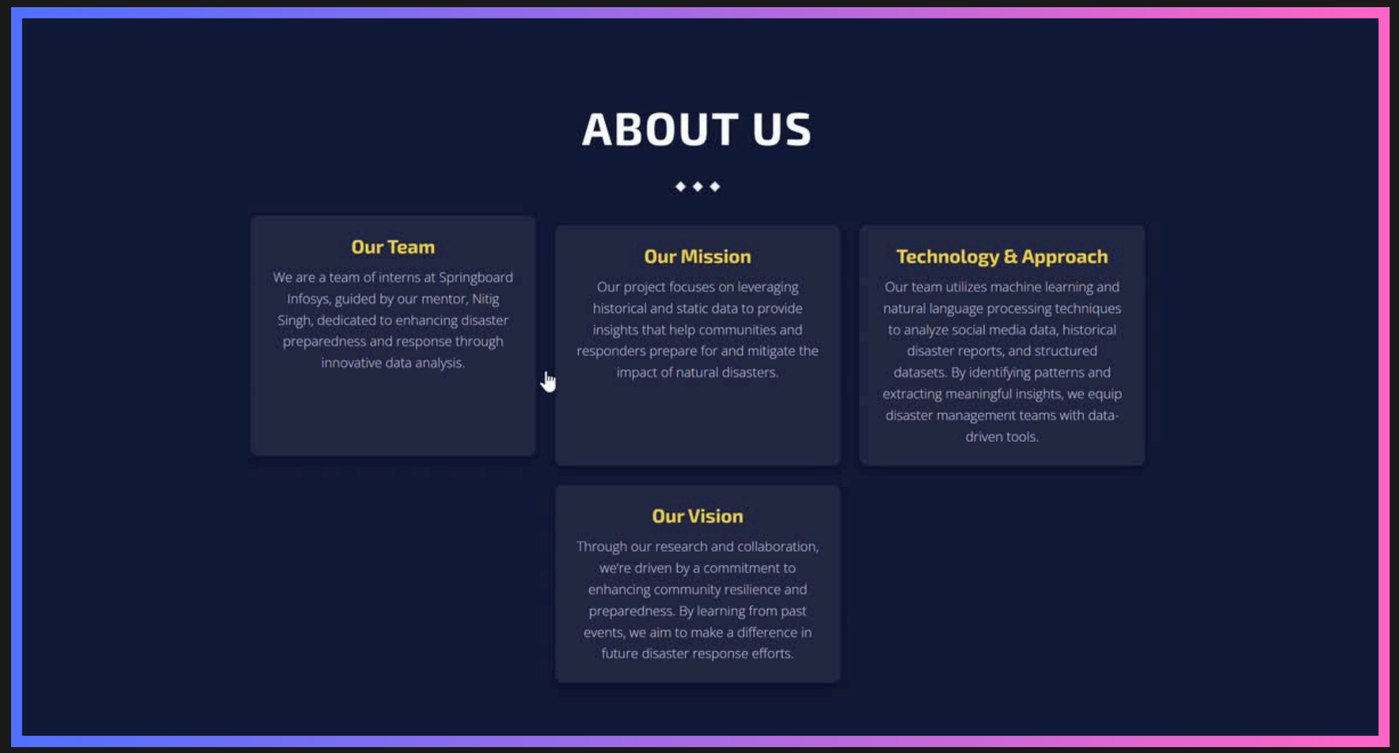


PROJECT TIMELINE



FEEDBACK

WEB INTERFACE DEMO



ABOUT US

CONCLUSION

FUTURE ENHANCEMENTS

current model provides a solid base for expanding disaster information analysis in the future.

EFFECTIVE DISASTER RELEVANCE IDENTIFICATION

accurately identifies disaster-related tweets

KEY DISASTER INFORMATION

specialized for relevancy and location extraction

FUTURE POSSIBILITIES WITH TWITTER API

INSTANT DISASTER CLASSIFICATION:	LOCATION BASED MAPPING	TREND TRACKING	REAL-TIME DASHBOARD
Real-time detection and classification of disaster-related tweets.	Use geo-tagged tweets to map disaster impact areas.	Monitor trending disaster-related hashtags and keywords.	Display live updates of tweet volume and affected locations.

CHROME EXTENSIONS

CHROME EXTENSION

1

Set Up Chrome Extension Framework

manifest.json to include permissions, content.js

2

Capture and Analyze Live Tweets

popup.html , popup.js, Integrate JavaScript listeners

3

Integrate Disaster Tweet Analyzer Model

captured tweet data from content.js to your backend model

4

Display Real-Time Analysis Results

“Real” or “Fake.”, confidence level indicator

5

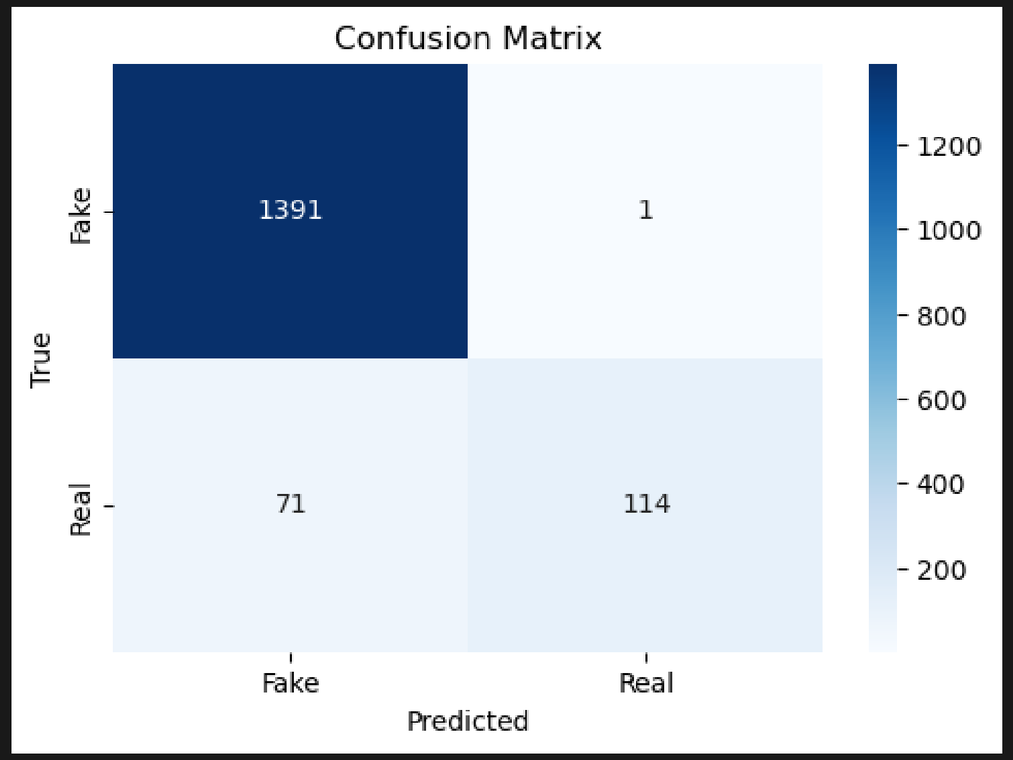
Test, Optimize, and Deploy

THANK YOU

RESULTS OF THE MODEL TRAINING ON FAKE / REAL TWEETS DATASET

Logistic Regression

CONFUSION MATRIX



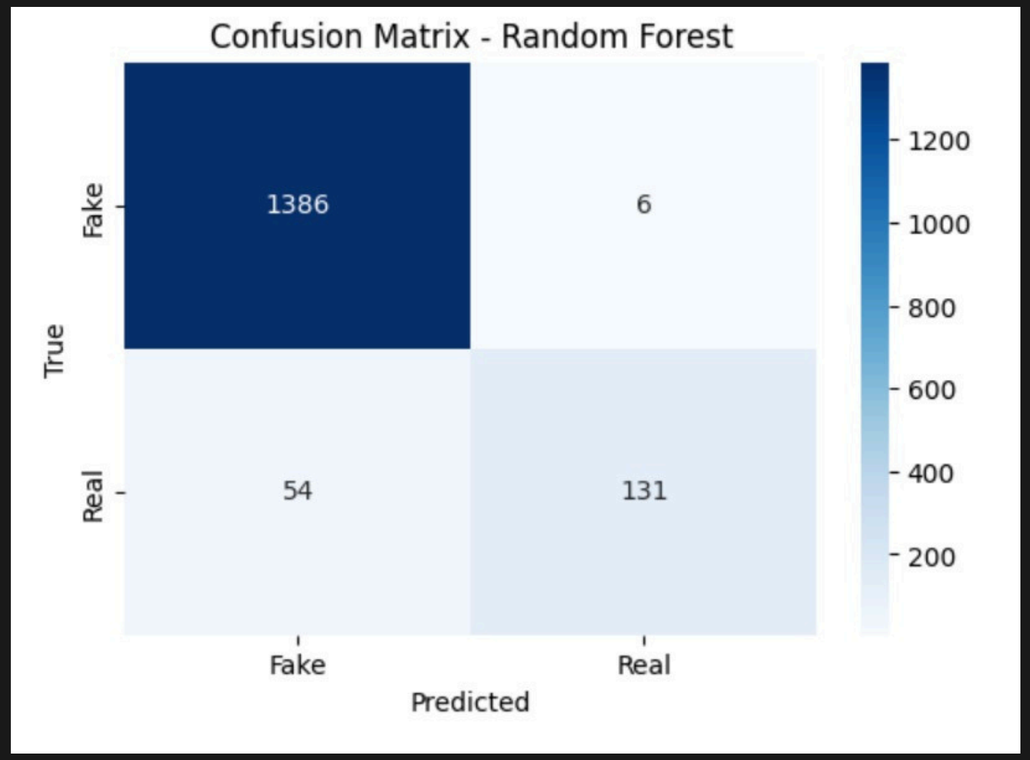
CLASSIFICATION REPORT

```
Accuracy: 0.9543436905516804
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      1392
           1       0.99      0.62      0.76       185

    accuracy                           0.95      1577
   macro avg       0.97      0.81      0.87      1577
weighted avg       0.96      0.95      0.95      1577
```

RESULTS OF THE MODEL TRAINING ON FAKE / REAL TWEETS DATASET

Random Forest

CONFUSION MATRIX



CLASSIFICATION REPORT

```
Accuracy: 0.9619530754597336
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1392
           1       0.96      0.71      0.81       185

    accuracy                           0.96      1577
   macro avg       0.96      0.85      0.90      1577
weighted avg       0.96      0.96      0.96      1577
```

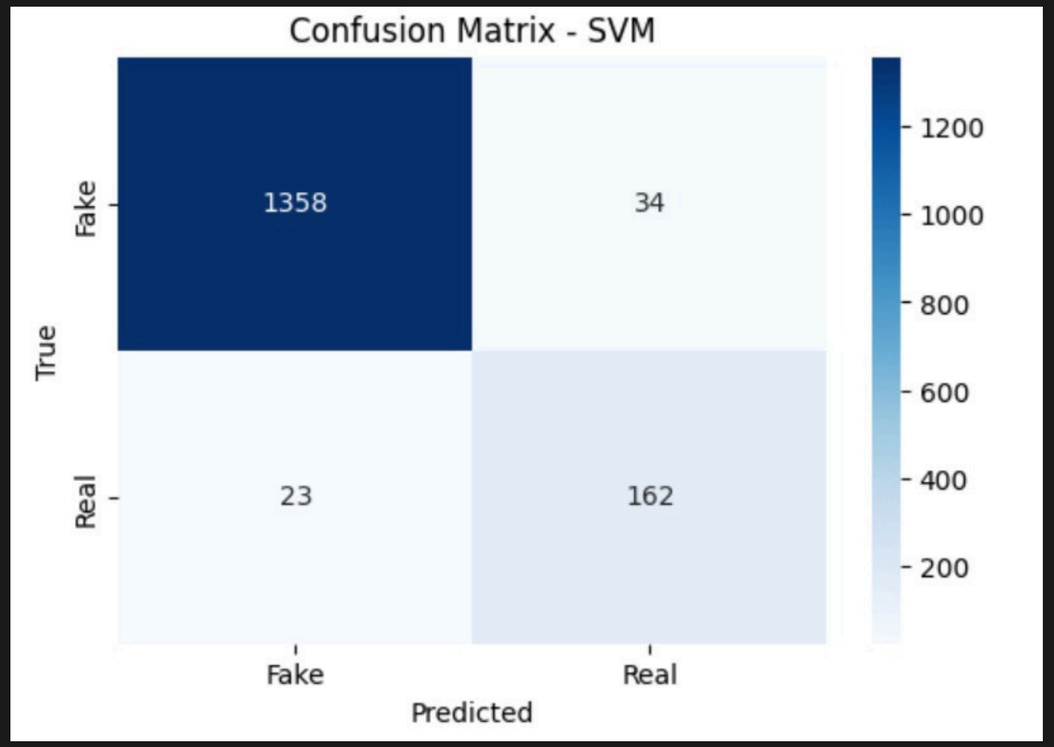
RESULTS OF THE MODEL TRAINING ON FAKE / REAL TWEETS DATASET

SVM

CONFUSION MATRIX

CLASSIFICATION REPORT

```
Accuracy: 0.963855421686747
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1392
           1       0.83      0.88      0.85       185

    accuracy                           0.96      1577
   macro avg       0.90      0.93      0.91      1577
weighted avg       0.96      0.96      0.96      1577
```



FINAL RESULTS

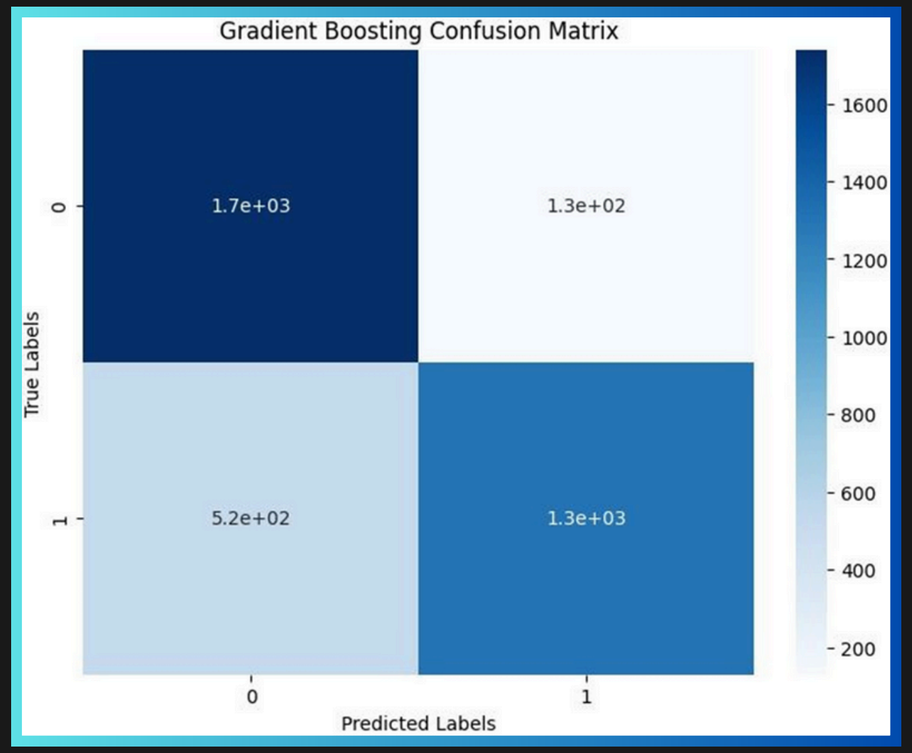
LINEAR SVM CLASSIFIER

```
Linear SVM Accuracy: 0.9443694301917365
Linear SVM Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.90      0.94      1869
           1       0.90      0.99      0.95      1834

    accuracy                           0.94      3703
   macro avg       0.95      0.94      0.94      3703
weighted avg       0.95      0.94      0.94      3703

Linear SVM ROC-AUC: 0.9858854769285705
```

FINAL RESULTS

```
Gradient Boosting Accuracy: 0.8236564947339995
Gradient Boosting Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.93      0.84      1869
           1       0.91      0.71      0.80      1834

    accuracy                           0.82      3703
   macro avg       0.84      0.82      0.82      3703
weighted avg       0.84      0.82      0.82      3703

Gradient Boosting ROC-AUC: 0.9006246378815699
```

GRADIENT BOOST



FINAL RESULTS

XG BOOST

```
XGBoost Accuracy: 0.8471509586821496
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.94      0.86      1869
           1       0.92      0.75      0.83      1834

    accuracy                           0.85      3703
   macro avg       0.86      0.85      0.85      3703
weighted avg       0.86      0.85      0.85      3703

XGBoost ROC-AUC: 0.9292231688112245
```

FINAL RESULTS

KNN

```
KNN Accuracy: 0.9540912773426952
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.97      0.96      1869
           1       0.97      0.94      0.95      1834

    accuracy                           0.95      3703
   macro avg       0.95      0.95      0.95      3703
weighted avg       0.95      0.95      0.95      3703

KNN ROC-AUC: 0.9890963332755695
```

APPENDIX

Model	Advantages	Disadvantages
1) Logistic Regression	Simplicity and Speed	Limited to Linear boundaries
	Handles Sparse Data Well	Sensitivity to Imbalance
2) Naive Bayes	Efficient and Fast	Insensitive to word order
	Low Overfitting Risk	Requires Balanced data

APPENDIX

Model	Advantages	Disadvantages
3) Random Forest	Estimating Feature Importance Handles Missing Data	Memory Usage Prediction Time
4) XGBOOST	High accuracy in complex data patterns Provides feature importance scores	Complex hyperparameter tuning Risk of overfitting on small datasets