Disaster Tweets Classification using Natural Language Processing (NLP)

By Keerthi Anand, Krupasankari

This presentation delves into the application of Natural Language Processing (NLP) in analyzing disaster-related tweets. We explored a Kaggle competition dataset containing real and non-disaster tweets.
We built a machine learning model that predicts which tweets are about real disasters and which one's aren't.

Data Overview: A Glimpse into Disaster Tweets

Dataset Description

The Kaggle competition dataset comprised over 10,000 tweets, each labeled as either a "disaster" or a "non-disaster" tweet. This diverse dataset provided valuable insights into how people communicate about real disasters and other events on social media. The dataset's distribution reflects real-world scenarios and showcases the variety of language used.

Source: Twitter

Natural Language Processing (NLP): Understanding the Power of Text

What is NLP?

Natural Language Processing (NLP) is a field of artificial intelligence focused on enabling computers to understand and process human language. It involves techniques for analyzing, interpreting, and generating text data, bridging the gap between human communication and machine comprehension.

Converts:

NLP STEPS

- SEGMENTATION

I see a cup of coffee. I want to drink it. → I see a cup of coffee. | I want to drink it.

- TOKENIZATION

I see a cup of coffee → "I", "see", "a", "cup", "of", "coffee"

- STOP WORDS REMOVAL

"I", "see", "a", "cup", "of", "coffee" → "I", "see", "a", "cup", "of", "coffee"

NLP STEPS

- STEMMING

Wanting → Want

Studies → **Studi**

- LEMMATIZATION

Studies → **Study**

STEMMING	LEMMATIZATION
Stemming is a process that stems or removes last few characters from a word, often leading to incorrect meanings and spelling.	**Lemmatization** considers the context and converts the word to its meaningful base form, which is called Lemma.
Ex: "Caring" --> "Car"	Ex: "Caring" --> "Care"
Stemming is used in case of large dataset where performance is an issue.	Lemmatization is computationally expensive since it involves look-up tables and what not.

NLP STEPS

- Part Of Speech (POS) Tagging

Need	verb
one	digit
shot	verb
espresso	noun
hot	adj

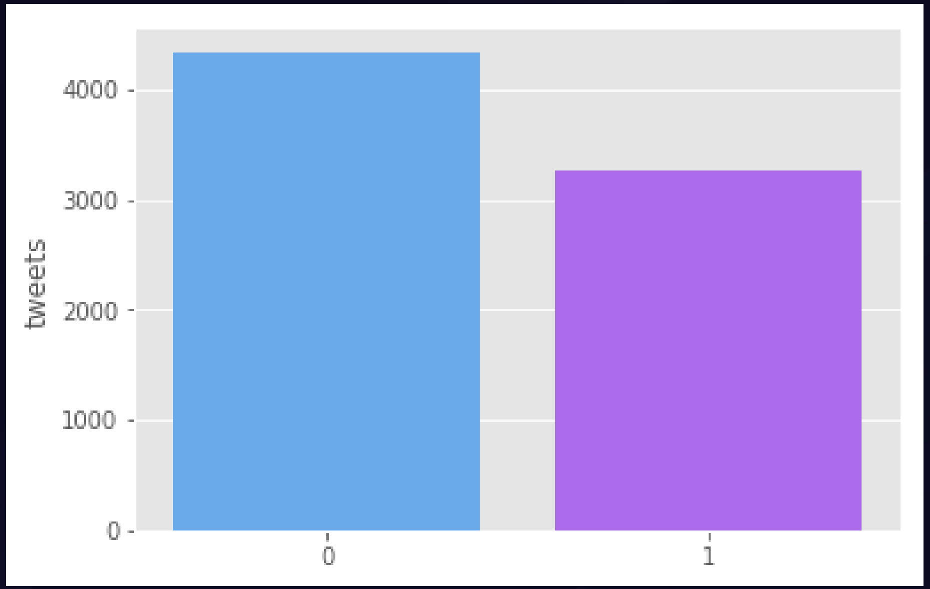
- Named Entity Recognition (NER)

U.S. President Trump starts a cryptocurrency called $TRUMP coin.

U.S.	**LOCATION**
President Trump	**PERSON**
$TRUMP coin	**PRODUCT**

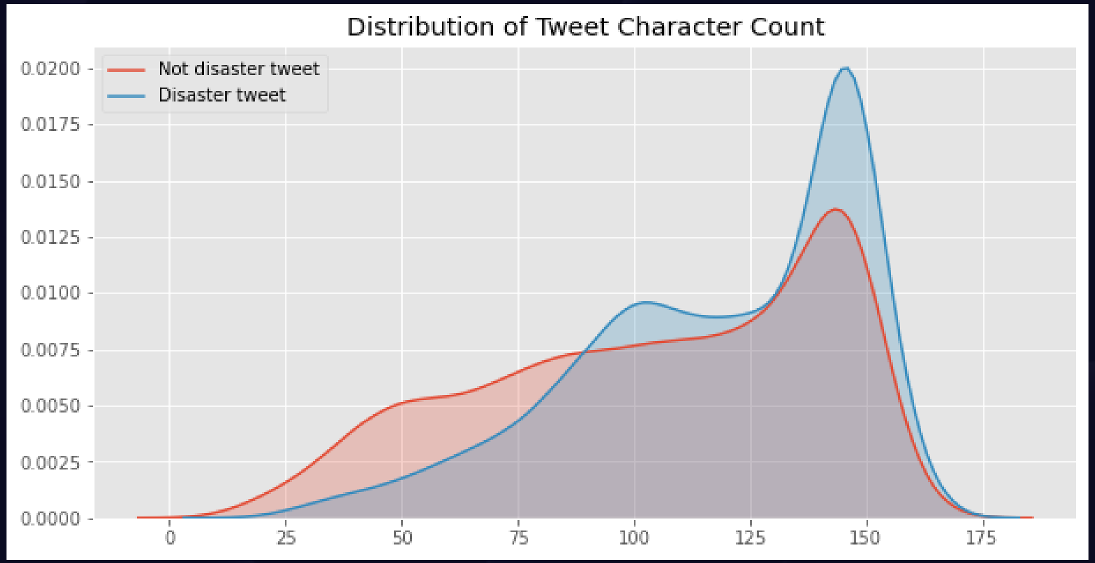
Exploratory Data Analysis:

1



- The **Majority class (Non-disaster tweets)** has more samples than the **Minority class (Disaster tweets)**, but the gap isn't too large.
- Standard classification models should still work well without major adjustments.

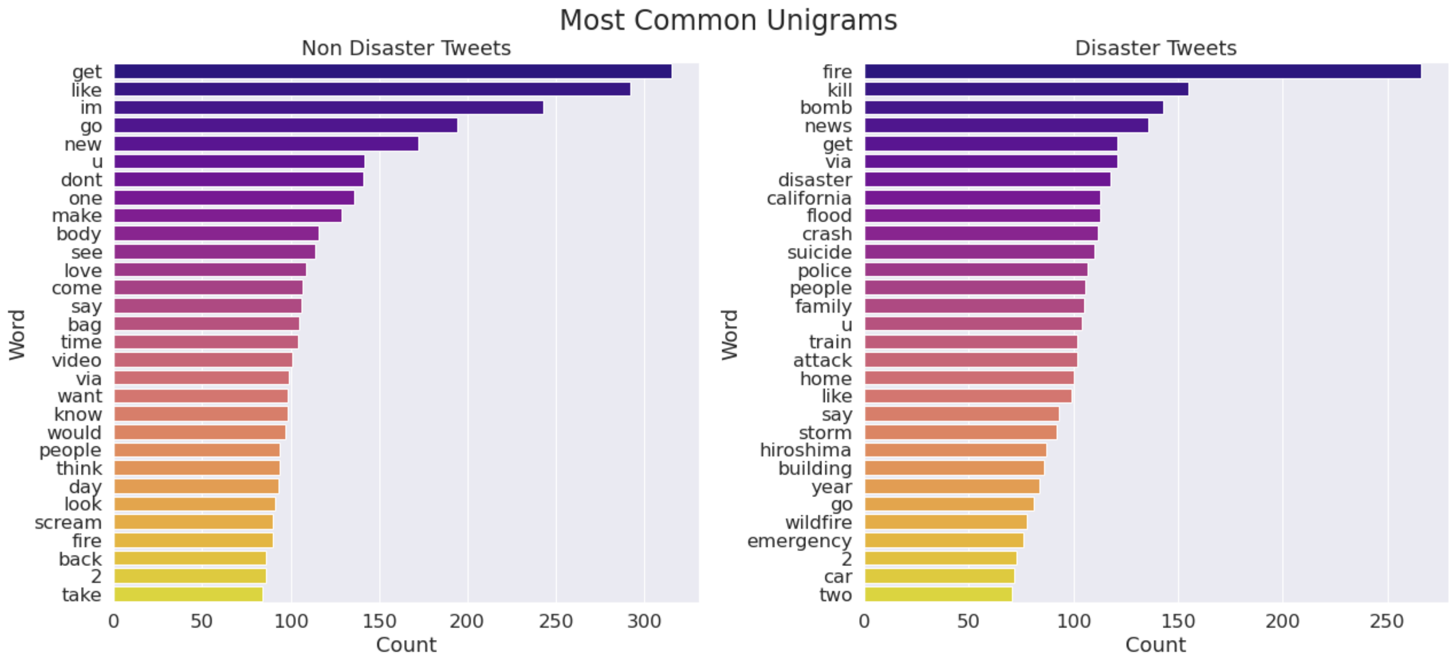
2



- This is insightful as it tells us that very **few disaster tweets** are **less than 50 characters** and that the **majority of them** are **more than 125 characters** long.

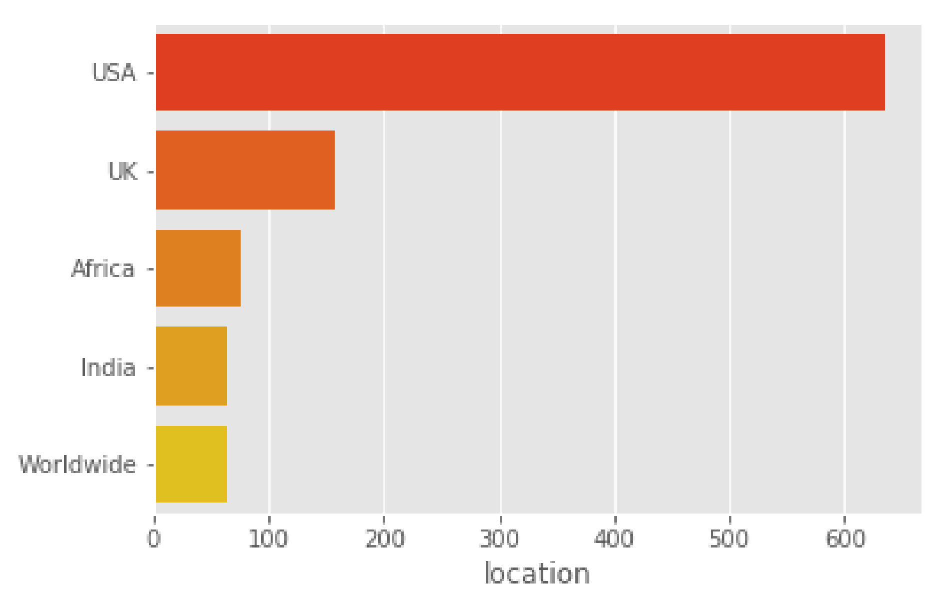
Exploratory Data Analysis:

3



- Disaster:
  - "fire","kill"
- Non-Disaster:
  - "get", "like"

4



- The **USA** had the highest number of disaster-related tweets.
- Followed by UK, Africa and India.

Word Embeddings

GloVE (Global Vectors for Word Representation)

Static Word Embeddings:
- Fixed representations for each word (e.g., “flood” is the same in all contexts).

Limitation:
- Ignores word context (e.g., "flood" in "flooded road" vs. "flood warnings").

Best Use Cases:
- Suitable for simple keyword-based classification tasks with limited context variation.

BERT (Bidirectional Encoder Representations from Transformers)

Contextualized Word Embeddings
- Understands word meaning based on surrounding words (e.g., “fire” in “wildfire” vs. “fire truck”).

Advantages:
- Captures context (e.g., urgency, tone, and meaning in disaster-related tweets).
- Handles informal language, slang, hashtags, and emojis.

Best Use Cases:
- Ideal for complex classification tasks with varying tone, sentiment, and meaning.

Feature	GloVe	BERT
Context Sensitivity	Limited (static embeddings)	High (contextual embeddings)
Model Complexity	Lightweight and fast	Computationally intensive
Handling Informal Language	Struggles with slang/emojis	Handles slang and emojis well
Speed	Fast processing	Slower, but more accurate

Source: Revisiting GloVe, Word2Vec and BERT: On the Homogeneity of Word Vectors

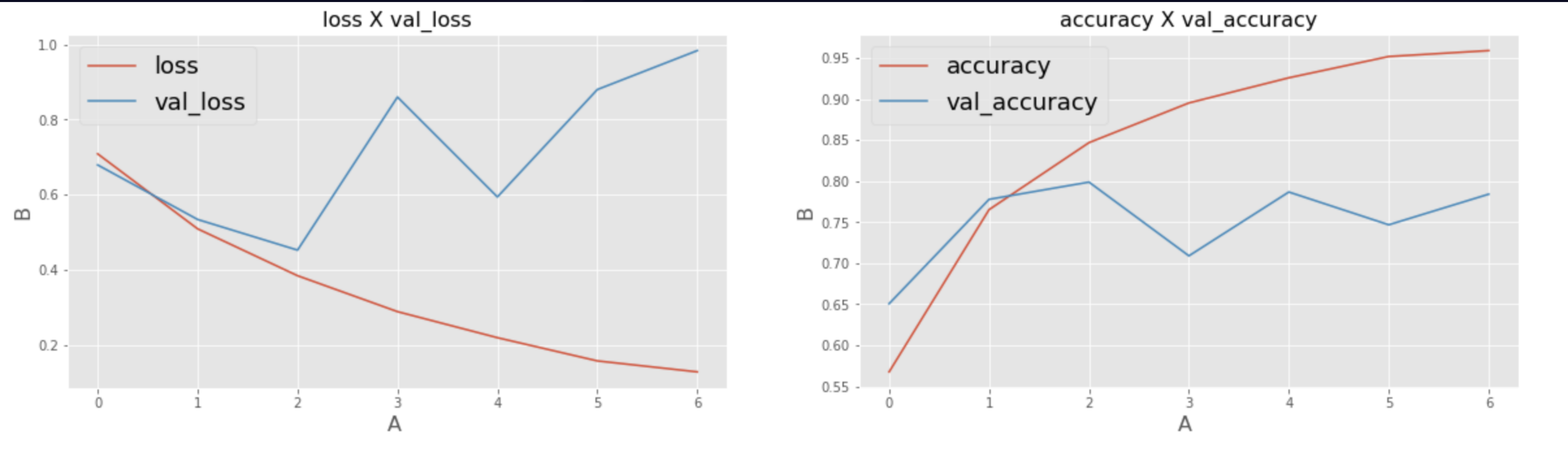
Long Short Term Memory (LSTM)

```python
def BLSTM():
    model = Sequential()
    model.add(Embedding(input_dim=embedding_matrix.shape[0],
                        output_dim=embedding_matrix.shape[1],
                        weights = [embedding_matrix],
                        input_length=length_long_sentence))
    model.add(Bidirectional(LSTM(length_long_sentence, return_sequences = True, recurrent_dropout=0.2)))
    model.add(GlobalMaxPool1D())
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(length_long_sentence, activation = "relu"))
    model.add(Dropout(0.5))
    model.add(Dense(length_long_sentence, activation = "relu"))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation = 'sigmoid'))
    model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

Model Performance

```
Train on 5709 samples, validate on 1904 samples
Epoch 1/7
5709/5709 [==============================] - 23s 4ms/step - loss: 0.7087 - accuracy: 0.5677 - val_loss: 0.6792 - val_accuracy: 0.6507

Epoch 00001: val_loss improved from inf to 0.67916, saving model to model.h5
Epoch 2/7
5709/5709 [==============================] - 21s 4ms/step - loss: 0.5090 - accuracy: 0.7653 - val_loss: 0.5339 - val_accuracy: 0.7778

Epoch 00002: val_loss improved from 0.67916 to 0.53387, saving model to model.h5
Epoch 3/7
5709/5709 [==============================] - 21s 4ms/step - loss: 0.3840 - accuracy: 0.8469 - val_loss: 0.4522 - val_accuracy: 0.7988

Epoch 00003: val_loss improved from 0.53387 to 0.45216, saving model to model.h5
Epoch 4/7
5709/5709 [==============================] - 21s 4ms/step - loss: 0.2882 - accuracy: 0.8953 - val_loss: 0.8602 - val_accuracy: 0.7090

Epoch 00004: val_loss did not improve from 0.45216
Epoch 5/7
5709/5709 [==============================] - 20s 4ms/step - loss: 0.2188 - accuracy: 0.9259 - val_loss: 0.5939 - val_accuracy: 0.7868

Epoch 00005: val_loss did not improve from 0.45216
Epoch 6/7
5709/5709 [==============================] - 21s 4ms/step - loss: 0.1569 - accuracy: 0.9518 - val_loss: 0.8801 - val_accuracy: 0.7468

Epoch 00006: val_loss did not improve from 0.45216
Epoch 7/7
5709/5709 [==============================] - 21s 4ms/step - loss: 0.1277 - accuracy: 0.9590 - val_loss: 0.9839 - val_accuracy: 0.7841

Epoch 00007: val_loss did not improve from 0.45216
```

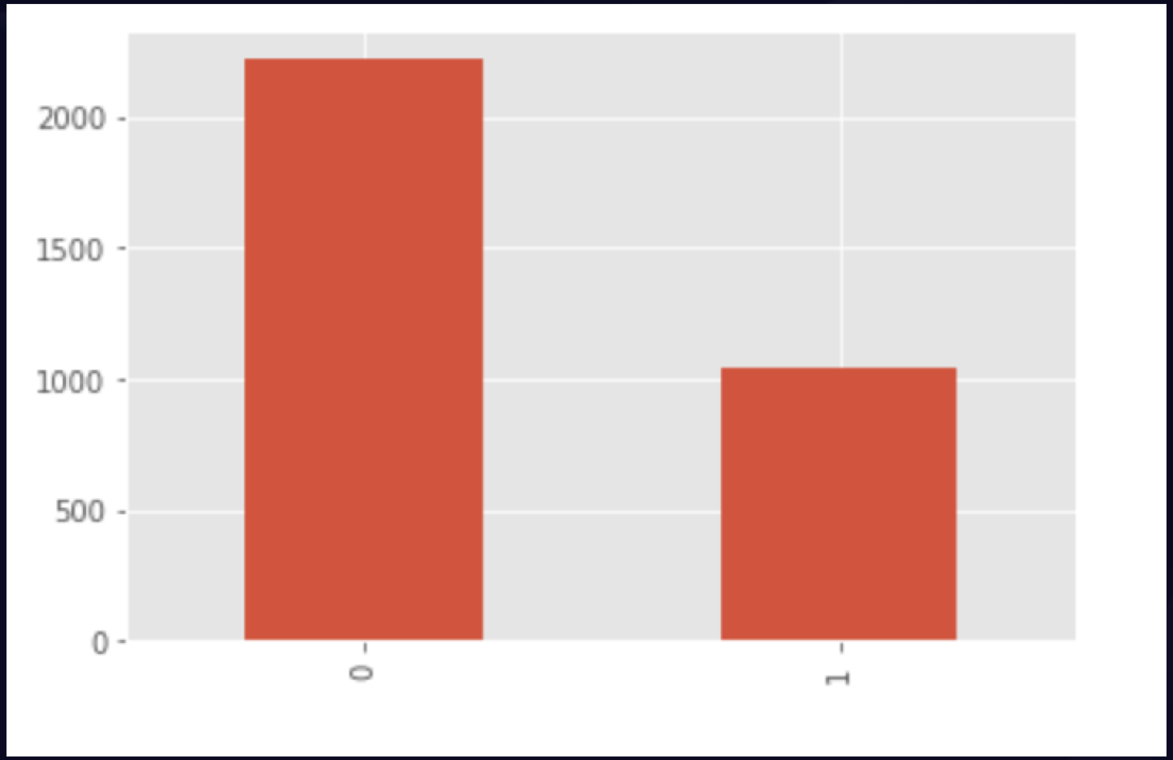
Model Performance



Model Performance

```
F1-score:  0.7246585190510423
Precision:  0.6339622641509434
Recall:  0.8456375838926175
Acuracy:  0.7988445378151261
--------------------------------------------------
              precision    recall  f1-score   support

           0       0.92      0.78      0.84      1308
           1       0.63      0.85      0.72       596

    accuracy                           0.80      1904
   macro avg       0.78      0.81      0.78      1904
weighted avg       0.83      0.80      0.80      1904
```

Performance Metrics



Submission class distribution

Kaggle Submission Scores

Natural Language Processing with Disaster Tweets

Predict which Tweets are about real disasters and which ones are not

Overview Data Code Models Discussion Leaderboard Rules Team **Submissions**

Submissions

All Successful Errors Recent

Submission and Description		Public Score
submission.csv Complete · now	BERT + LSTM	**0.78516**
submission.csv Complete · 6m ago	GLoVE + LSTM	**0.78271**
submission.csv Complete · 1h ago	Stemming + Lemmatization + BERT + LSTM	**0.80570**

Summary

Objective:
Analyze disaster-related tweets using NLP techniques.

Key Techniques:
BERT: Contextualized word embeddings for understanding tweet content.
LSTM: Sequential modeling for accurate tweet classification.

Results:
Achieved high accuracy in distinguishing between real disasters and non-disaster events.

Impact:
- Enables quick, accurate analysis of social media data.
- Enhances disaster response efforts and decision-making.
- Saves lives by providing actionable insights.

References:
- https://www.kaggle.com/code/philculliton/nlp-getting-started-tutorial
- http://nltk.org/