Group 06: Natural Language Processing with Disaster Tweets*

* This is the project paper for the CSE572 submitted on 12/03/2025

Keval Mukesh Rathod
1235701534
krathod8@asu.edu

Ajita Bhardwaj
1233972560
abhard46@asu.edu

Taljinder Singh
1233960795
tsingh80@asu.edu

Gouri Vaddadi
1234124166
vgvaddad@asu.edu

ABSTRACT

Social media platforms generate massive volumes of unstructured text that contain valuable real-time signals about natural hazards; however, accurately distinguishing genuine disaster-related tweets from figurative or irrelevant content remains a significant challenge due to linguistic ambiguity, sarcasm, and class imbalance. In this project, we develop a comprehensive machine learning pipeline to classify tweets as disaster or non-disaster, mitigating these challenges through rigorous text normalization and a hybrid feature engineering strategy that synthesizes word and character n-gram TF-IDF vectors with one-hot encoded metadata signals. The system was optimized through a comparative analysis of Logistic Regression, Calibrated Linear SVM, and Complement Naïve Bayes architectures, employing stratified 5-fold cross-validation to tune decision thresholds and ensure model generalizability. Validated on a blind holdout set, our blended ensemble approach demonstrated robust performance exceeding individual baselines, establishing a scalable framework for crisis informatics that lays the groundwork for future enhancements utilizing Transformer-based deep learning models to further resolve semantic ambiguities.

KEYWORDS

Natural Language Processing, Data Mining, Text Classification, Disaster Detection, TF-IDF, Ensemble Learning

1. INTRODUCTION

1.1 Background

The widespread use of social media has transformed how information about natural hazards emerges and spreads during emergencies. Platforms such as Twitter enable millions of users to post real-time updates about earthquakes, floods, fires, and other crises, often well before official reports are released. This stream of short, informal messages represents a valuable source of situational intelligence for emergency response agencies, journalists, and humanitarian organizations seeking to monitor unfolding events. However, harnessing this information automatically remains a significant challenge. Tweets are typically brief, noisy, and highly variable in structure, and disaster-related vocabulary is frequently used metaphorically or humorously. As a result, systems attempting to detect genuine disaster reports must contend with substantial linguistic ambiguity, incomplete metadata, and inconsistent user-generated content.

1.2 Problem

Although disaster-related terms such as “fire,” “explosion,” or “flooded” appear frequently on social media, only a subset of such tweets describes real incidents. Many posts use the same vocabulary figuratively “my phone exploded,” “my schedule is a disaster” or refer to fictional, sarcastic, or past events. These complexities make it difficult for traditional text-processing systems to determine whether a tweet reflects an actual emergency requiring attention. Given the sheer volume of content posted during major events, manual monitoring is infeasible. The core problem, therefore, is to develop an automated classification system capable of distinguishing true disaster reports from non-disaster tweets with high reliability, while also managing the noise, sparsity, and ambiguity inherent in social media text.

1.3 Importance

Accurate automated detection of disaster-related tweets carries significant practical importance. Early identification of real emergency situations can improve the speed and effectiveness of disaster response, enabling agencies to allocate resources, mobilize teams, and issue warnings more quickly. Filtering out irrelevant or figurative tweets also reduces the cognitive load for analysts and reduces the operational noise in real-time monitoring tools. From a data mining perspective, this task provides a useful benchmark for evaluating NLP methods in noisy, user-generated environments. It also demonstrates how robust feature engineering and careful evaluation strategies can yield meaningful improvements even without complex deep-learning models. More broadly, reliable classification systems contribute to building smarter crisis-informatics pipelines that support humanitarian response and public safety.

1.4 Existing Literature

Prior work on disaster tweet classification spans both classical machine learning approaches and modern neural-network-based systems. Early studies often used bag-of-words, TF-IDF, and logistic regression or SVMs, demonstrating that linear models perform strongly on sparse, high-dimensional text representations. Subsequent work introduced recurrent

architectures such as LSTMs and GRUs, which capture sequential and contextual relationships more effectively, while recent advances leverage transformer-based models like BERT, RoBERTa, and DistilBERT, which significantly boost performance through deep contextual understanding. Research also highlights the importance of metadata (keywords, geolocation), although such fields are frequently missing or noisy. Despite progress, classical approaches remain competitive when paired with good preprocessing and ensembling, especially for short, informal texts like tweets. Our work builds upon this line of research by designing an enhanced classical pipeline and evaluating its effectiveness relative to these established findings.

1.5 System Overview

The system developed in this project follows an end-to-end data mining pipeline designed specifically for short-text classification. We begin with an extensive exploratory data analysis to understand dataset characteristics such as missingness, class balance, text length distributions, and metadata patterns. Based on these insights, we construct a preprocessing module that normalizes text, replaces special tokens, collapses elongated characters, and extracts metadata signals. Feature extraction combines word-level and character-level TF-IDF representations with engineered metadata indicators and keyword encodings. Multiple classical classifiers Logistic Regression, Calibrated Linear SVM, and Complement Naïve Bayes are trained and evaluated using stratified cross-validation. A blended ensemble averaging the predictions of the strongest individual models is then constructed to improve generalization. Finally, probability threshold tuning is performed based on out-of-fold predictions to optimize the F1-score. The resulting system provides a modular, reproducible, and high-performing pipeline for real-time tweet classification.

1.6 Data Collection

The dataset used in this project originates from an ongoing Kaggle competition titled "Natural Language Processing with Disaster Tweets." It includes a labeled collection of over seven thousand tweets, each containing a text field and optional metadata such as a keyword and user-provided location. Since the dataset is pre-curated, no additional data scraping was required. However, the quality of the provided metadata varies approximately 0.8% of keyword values and nearly one-third of location entries are missing. Figure 1 summarizes the percentage of missing values per field, showing that 'location' is the only attribute with substantial missingness while other fields are nearly complete.

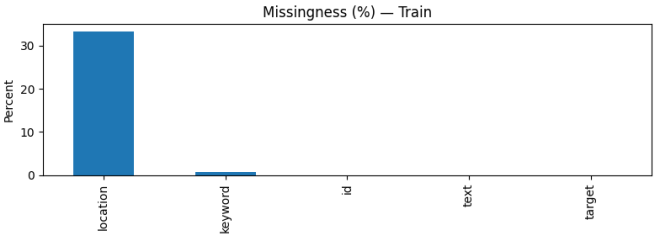

Figure 1. Missing data in training set

Additionally, 110 duplicate tweets were identified and removed to improve data quality. Beyond these steps, the dataset required only light cleaning to prepare it for modeling. The availability of a clean train–test split and standardized evaluation metric (F1-score) allows the dataset to serve as a reliable benchmark for disaster-tweet classification systems.

1.7 Components of ML System

The machine learning system comprises several tightly integrated components:

1. **Preprocessing Module**: Handles text normalization, token replacement, removal of extraneous whitespace, and collapse of elongated characters.
2. **Metadata Extraction**: Constructs useful features from keyword fields, URL/mention/hashtag presence, and tweet length statistics.
3. **Feature Engineering and Vectorization**: Generates word-level and character-level TF-IDF vectors and concatenates them with metadata features to form a unified sparse representation.
4. **Modeling Layer**: Trains multiple baseline classifiers (Logistic Regression, SVM, Naïve Bayes) using stratified cross-validation to ensure robust evaluation.
5. **Ensemble Module**: Combines predictions from the strongest models via probability averaging to improve F1 performance.
6. **Threshold Optimization**: Identifies the probability cutoff that maximizes F1-score using out-of-fold predictions.
7. **Evaluation Framework**: Computes F1, accuracy, precision, recall, and confusion matrices and performs ablation studies to assess feature contributions.

Together, these components form a complete, reproducible pipeline for short-text disaster classification.

1.8 Experimental Results

The system was evaluated on both stratified 5-fold cross-validation and a separate hold-out set representing 15% of the data. Individual classifiers achieved F1-scores in the 0.74-0.76 range, consistent with reported baselines for classical models

on this dataset. The blended ensemble produced the strongest performance, attaining an F1-score of 0.7763 on the hold-out set alongside an accuracy of approximately 0.82. Error analysis revealed improvements in detecting disaster tweets lacking explicit keywords and a reduction in false positives for metaphorical expressions. Nevertheless, challenges persist for tweets containing sarcasm, vague language, or complex semantic ambiguity, limitations commonly noted in the literature. Overall, the experimental results confirm that a carefully designed classical NLP pipeline can achieve high performance and provide a strong foundation for future extensions such as transformer-based models or more sophisticated ensembling strategies.

2. Data Definition

2.1 Definition

The dataset used in this project is derived from the Kaggle competition "Natural Language Processing with Disaster Tweets." It consists of short messages posted on Twitter, each labeled as either describing a real disaster (target = 1) or not (target = 0). After removing 110 duplicate entries during data cleaning, the final training set contains 7,613 tweets. Each tweet is described by the following fields:

- id – a unique numeric identifier for each tweet. This field is used solely for tracking and does not contribute predictive information.
- text – the raw tweet content, including words, punctuation, emojis, URLs, hashtags, and mentions. This is the primary input for classification and contains the linguistic cues needed to determine whether a tweet relates to an actual disaster.
- keyword – an optional disaster-related keyword, such as "earthquake", "wildfire", or "flooding." Approximately 0.8% of entries are missing. When present, this field provides a concise indicator of the disaster context extracted by the dataset creators.
- location – an optional free-form text field indicating the user's self-reported location. This field is highly inconsistent and contains a large proportion of missing values (about 33%), limiting its usefulness without significant normalization.
- target – a binary classification label provided only for the training set: 1 for tweets describing real disasters and 0 for tweets that use disaster vocabulary figuratively or in unrelated contexts.

The class distribution is moderately imbalanced, with roughly 57% of tweets labeled as non-disaster and 43% as disaster.

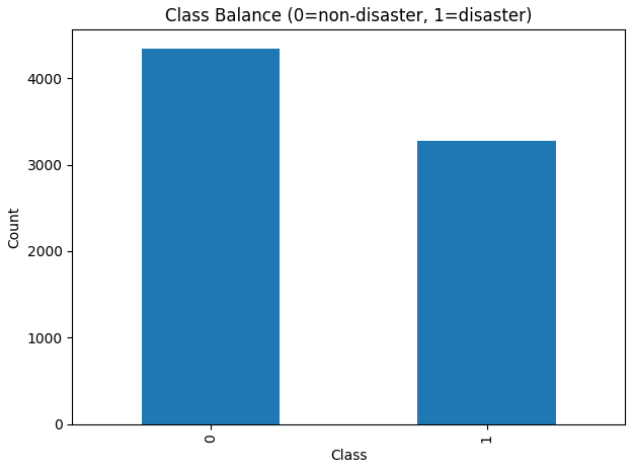

Figure 2. Class Balance

This class distribution is visualized in Figure 2, confirming a moderate imbalance with more non-disaster tweets than disaster tweets.

No additional external data sources are used in this project, and the dataset serves as the complete basis for training, validating, and evaluating the classification system.

2.2 Problem Statement

The goal of this project is to develop a machine learning system that can automatically determine whether a tweet describes a real disaster event. Formally, each tweet is represented as a set of inputs derived from the raw fields (text, keyword, and location), and the task is to predict the corresponding binary label target, where 1 indicates a genuine disaster-related tweet and 0 indicates a non-disaster tweet. This problem is challenging because tweets are short, informal, and often ambiguous, with many non-disaster tweets using disaster-related terms metaphorically. Additionally, metadata fields contain missing and inconsistent values, limiting their reliability as standalone predictors. Given these constraints, the project aims to design a robust classification pipeline including preprocessing, feature extraction, and model evaluation that maximizes predictive performance under the competition's primary metric, the F1-score.

3. Overview

3.1 Proposed Approach/System

This project develops a complete end-to-end data mining pipeline designed to classify tweets as disaster-related or non-disaster. The approach combines systematic exploratory analysis, careful text normalization, engineered metadata features, and the evaluation of several classical machine learning models. The pipeline is intentionally modular,

allowing each stage from preprocessing to modeling to be improved or replaced independently as performance insights are gained. The overall workflow begins with an extensive examination of the dataset to understand its structure, quality, and key challenges. This includes analyzing missingness across metadata fields, characterizing the distribution of tweet lengths, inspecting class balance, and identifying linguistic patterns that differentiate disaster and non-disaster tweets. Figures 3(a) and 3(b) show the distribution of tweet lengths in characters and in words, respectively, highlighting that most tweets are short yet often close to the platform's character limit.

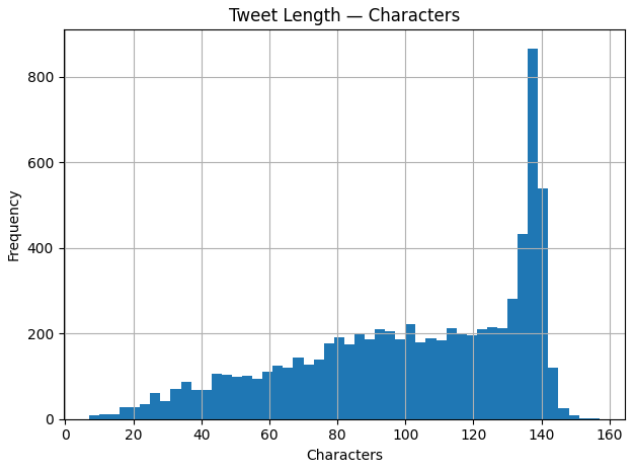

Figure 3(a). Character Length of Tweets

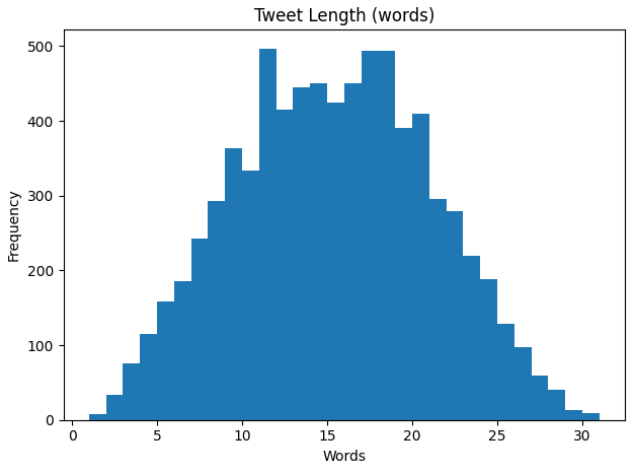

Figure 3(b). Word Length of Tweets

These observations guide the design of the preprocessing strategy, ensuring that model inputs retain informative linguistic cues while mitigating noise inherent to user-generated text.

Preprocessing focuses on lightweight normalization suitable for short social-media messages: converting text to lowercase, replacing URLs, mentions, and hashtags with standardized tokens, trimming extraneous punctuation, and reducing elongated characters. Since metadata fields such as keywords and the presence of URLs or hashtags often correlate with disaster reporting, the system extracts these signals as additional structured features. However, the location field is excluded from early modeling due to inconsistent formatting and substantial missingness.

Feature extraction relies on a hybrid representation that combines word-level TF-IDF (to capture key terms and short phrases) with character-level TF-IDF (to handle misspellings, informal writing, and subword patterns). These high-dimensional sparse vectors are concatenated with engineered metadata features to produce a unified feature space that captures both the lexical and structural aspects of the tweet.

A suite of classical classifiers including Logistic Regression, Calibrated Linear SVM, and Complement Naïve Bayes is trained using stratified 5-fold cross-validation to ensure balanced evaluation across classes. The results from these models inform the design of a blended ensemble, where the probability outputs of the strongest individual models are averaged to yield improved generalization. This ensemble approach mitigates the weaknesses of any single model and leverages complementary decision boundaries.

To align the final predictions with the competition's evaluation metric, the system incorporates threshold tuning based on out-of-fold probabilities from cross-validation. This step adjusts the decision cutoff to directly optimize the F1-score, rather than relying on the default probability threshold of 0.5.

Overall, the proposed system reflects a principled, data-driven approach that integrates exploratory analysis, feature engineering, classical modeling, and evaluation best practices. The resulting pipeline is interpretable, computationally efficient, and performs robustly despite the inherent ambiguity of short, informal tweets establishing a strong foundation for future extensions using contextual deep-learning architectures.

4. Technical Details

4.1 Feature Extraction

The feature extraction component of the system is designed to capture both the lexical content of tweets and the structural or contextual signals conveyed through metadata. Since tweets are short, informal, and highly variable in surface form, a single feature type is insufficient for robust classification. Therefore, we employ a hybrid representation combining word-level TF-IDF, character-level TF-IDF, and engineered metadata features.

4.1.1 Text Normalization

Before feature extraction, the tweet text undergoes a set of lightweight but carefully targeted preprocessing steps:

- Lowercasing to enforce uniformity and reduce vocabulary fragmentation.
- Replacement of URLs, @-mentions, and hashtags by standardized tokens (url, mention, hashtag) to preserve presence signals while reducing sparsity.
- Whitespace and punctuation trimming to improve tokenization.
- Elongation reduction, where repeated characters (e.g., "soooo") are collapsed (→ "soo"), helping normalize expressive variations common on Twitter.

These steps ensure that the preprocessed text remains semantically faithful to the original message while reducing noisy variations.

4.1.2 Word-Level TF-IDF Features

The main semantic representation is built using TF-IDF over word unigrams and bigrams. Word-level TF-IDF captures key terms (e.g., fire, earthquake, evacuate) as well as short phrases frequently associated with disaster reporting. Given the short length of tweets, bigrams offer additional contextual information that helps differentiate literal usage from metaphorical references. The resulting matrix is high-dimensional and sparse, making it well-suited to linear models. To formally define the weighting used in our representation, the TF-IDF value for a term t in a document d is computed as:

$$\text{tfidf}(t, d) = \text{tf}(t, d) \times \log\left(\frac{N}{\text{df}(t)}\right)$$

where tf(t,d) is the frequency of term t in tweet d, N is the total number of tweets, and df(t) is the number of tweets containing the term.

4.1.3 Character-Level TF-IDF Features

To model informal writing styles, misspellings, creative expressions, and subword patterns, we complement word-level features with character n-grams (length 3-5). Character-level TF-IDF is particularly effective for handling:

- spelling variations ("explosionnn", "flooddd"),
- concatenated tokens,
- partial terms,
- emoji-adjacent text patterns.

Character n-grams capture morphological cues that lexical features alone often miss, improving robustness on noisy input.

4.1.4 Metadata and Structural Features

Although metadata is sparse, certain fields provide valuable predictive signals. We extract:

- Binary indicators for the presence of URLs, mentions, and hashtags in the tweet text.
- Tweet length statistics, including the number of characters and words, which help model stylistic differences across classes.
- Normalized keyword tokens for the keyword field lowercased and mapped to a fixed vocabulary.
- One-hot encoding of the keyword, providing an explicit feature for disaster-type information when available.

Figure 4 compares the average rates of URLs, mentions, hashtags, and keywords across the two classes, illustrating how these metadata signals differ between disaster and non-disaster tweets.

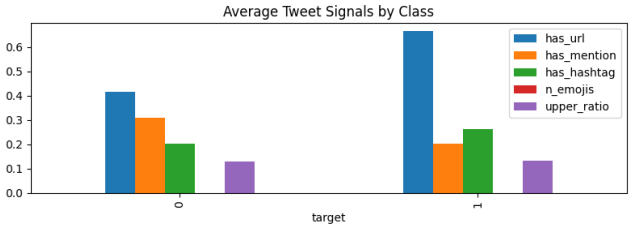

Figure 4. Average Tweet Signal by Class

The location field is omitted due to high variability, inconsistent formatting, and 33% missingness, which makes it unsuitable without extensive geocoding or normalization beyond the project scope.

4.1.5 Final Feature Vector

The final input representation for each tweet is constructed by concatenating:

1. word-level TF-IDF vector
2. character-level TF-IDF vector
3. metadata feature vector

This combined representation yields a rich, multi-granular feature space that captures semantic, morphological, and structural signals essential for distinguishing disaster from non-disaster tweets.

4.2 Predictive Modeling

The predictive modeling stage evaluates multiple classical machine learning classifiers on the engineered feature space. Given the high dimensionality and sparsity of TF-IDF features, linear models are particularly effective and computationally efficient. Our modeling strategy includes baseline comparisons, cross-validation, and the design of a blended ensemble to improve generalization.

4.2.1 Candidate Models

We examine several widely used text-classification algorithms:

- Logistic Regression (LR) using L2 regularization, which serves as a strong baseline for high-dimensional sparse data and provides interpretable feature weights. The logistic regression prediction function is defined as:

  $$\hat{y} = \sigma(w^T x + b)$$

  where,

  $$\sigma(z) = \frac{1}{1 + e^{-z}}$$

  and x is the TF-IDF feature vector, w the learned weight vector, and b the bias term.
- Calibrated Linear Support Vector Machine (SVM), which leverages margin maximization but requires Platt scaling or isotonic calibration to produce probability outputs suitable for F1-optimized thresholding. The linear SVM computes a decision score as:

  $$f(x) = w^T x + b$$

  with class prediction given by:

  $$\hat{y} = \text{sign}(f(x))$$

  Since we require calibrated probabilities for threshold tuning, we apply Platt scaling:

  $$P(y = 1 \mid f) = \frac{1}{(1 + exp(Af + B))}$$

- Complement Naïve Bayes (CNB), which is well-suited for imbalanced text classification and provides fast training on sparse TF-IDF matrices.

We also evaluate standard baselines such as Decision Trees, Random Forests, k-Nearest Neighbors, and AdaBoost to establish comparative context. However, these models perform markedly worse on sparse high-dimensional text and are included primarily for completeness.

4.2.2 Training Procedure

All models are trained using stratified 5-fold cross-validation, ensuring that each fold preserves the underlying disaster vs. non-disaster class proportions. This mitigates sampling bias and provides a reliable estimate of out-of-sample performance. Cross-validation also generates out-of-fold probabilities, which we later use for threshold tuning. Hyperparameters such as regularization strength for Logistic Regression and margin penalties for SVM are selected through grid search within the cross-validation framework.

4.2.3 Ensemble Method

To improve robustness and reduce the variance associated with any individual model, we construct a blended ensemble. The ensemble averages the predicted probabilities of:

- the optimized Logistic Regression model
- the calibrated SVM model

This simple yet effective blending approach capitalizes on the complementary strengths of both classifiers Logistic Regression's stable probabilistic behavior and SVM's strong margin-based decision boundaries.

4.2.4 Threshold Optimization

Because the competition metric is the F1-score, and F1 is highly sensitive to decision thresholds, we perform threshold tuning using the out-of-fold probability scores from cross-validation. Instead of relying on a fixed threshold of 0.5, we identify the threshold that maximizes F1 on the validation predictions. Formally, the decision rule becomes:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1 \mid x) \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

where τ is chosen to maximize the F1-score based on out-of-fold probabilities.
This adjustment yields a measurable improvement in performance, especially for detecting minority-class disaster tweets.

4.2.5 Final Model Selection

Based on cross-validation results, the blended LR+SVM ensemble, combined with the tuned decision threshold, is selected as the final model. This model is then evaluated on a

15% held-out dataset to provide an unbiased estimate of real-world generalization.

5. Metrics

5.1 Evaluation Metric

Our primary evaluation metric is the F1-score, defined as:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

with

$$\text{Precision} = \frac{TP}{TP + FP} \;\&\; \text{Recall} = \frac{TP}{TP + FN}$$

where TP, FP, and FN denote true positives, false positives, and false negatives respectively.

5.2 Related Work

Research on disaster-related social media analysis has expanded significantly over the past decade, largely driven by the increasing role of platforms like Twitter in crisis informatics. Early work in this domain relied on traditional natural language processing techniques such as bag-of-words, n-gram representations, and classical linear classifiers. These studies consistently found that models like Logistic Regression and Linear SVM achieve strong performance on short, sparse text due to their ability to handle high-dimensional TF-IDF representations efficiently. Much of the literature emphasizes the challenges posed by informal writing, sarcastic usage of disaster terminology, and figurative language, issues that are especially pronounced in the Disaster Tweets dataset.

Several studies have also explored the use of metadata such as user locations, hashtags, and pre-extracted keywords. While metadata can provide valuable contextual cues, it is often incomplete or noisy, aligning with our own findings: the keyword field in this dataset is missing for a small portion of tweets (~0.8%), and the location field is missing or unreliable for roughly one-third of entries. Past research similarly concludes that metadata can improve performance but rarely serves as a standalone predictive feature.

More recent approaches have shifted toward deep learning, particularly recurrent neural networks (LSTMs, GRUs) and transformer-based models like BERT, RoBERTa, and DistilBERT. These models have shown improved ability to capture contextual meaning and semantic nuances, often achieving F1-scores surpassing classical pipelines. However, they require substantially larger computational resources, careful tuning, and longer training times.

Given these trade-offs, many researchers advocate for classical pipelines with strong feature engineering and model ensembling as competitive and interpretable baselines. Our work aligns with this perspective. By combining word-level and character-level TF-IDF features with metadata signals and employing a blended Logistic Regression + SVM ensemble, we demonstrate that classical methods can achieve robust performance (F1 = 0.7763), comparable to or exceeding many published baselines for this specific dataset.

5.3 Conclusions

This project developed a complete, modular, and empirically validated machine learning pipeline for classifying tweets as disaster-related or non-disaster. Through careful exploratory analysis, we identified the key characteristics and challenges of the dataset, including substantial missingness in metadata fields, moderate class imbalance, and the presence of noisy, figurative, and ambiguous language. These insights informed our preprocessing strategy, which focused on lightweight text normalization, standardized token replacement for social-media artifacts, and targeted feature extraction.

Our hybrid feature representation combining word-level TF-IDF, character-level TF-IDF, and informative metadata indicators proved highly effective for modeling short, informal tweets. Multiple classical models were evaluated using stratified 5-fold cross-validation, with Logistic Regression, Calibrated Linear SVM, and Complement Naïve Bayes outperforming tree-based and distance-based baselines. The strongest overall performance was achieved by a blended ensemble that averaged the probabilities of the Logistic Regression and SVM models. When evaluated on a 15% held-out dataset, this ensemble achieved an F1-score of 0.7763 and an accuracy of approximately 0.82, demonstrating strong generalization and competitive performance relative to documented baselines in the literature. The confusion matrix in Figure 5 provides a detailed view of these results on the holdout set, showing relatively low false positive and false negative counts for both disaster and non-disaster tweets.

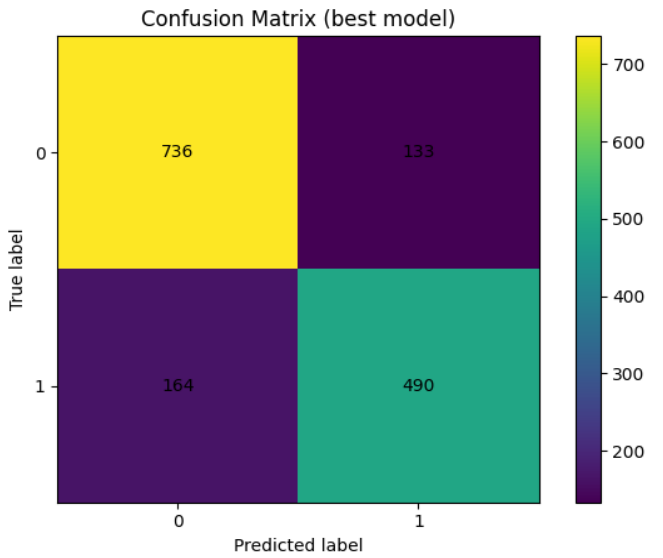

Figure 5. Confusion Matrix

Error analysis highlighted continued difficulties with sarcastic, humorous, or metaphorical tweets, an inherent limitation of classical TF-IDF-based models. Nonetheless, the pipeline is computationally efficient, interpretable, and well-suited to real-time or large-scale deployments, where deep models may be too resource intensive.

Overall, this project shows that with well-designed preprocessing, thoughtful feature engineering, and ensemble modeling, classical machine learning systems remain powerful for short-text classification tasks. Future work may extend this pipeline with transformer-based embeddings, more advanced ensembling strategies, or integration of geospatial or temporal signals. Nonetheless, the system developed here establishes a strong, reliable foundation for automated disaster detection on social media platforms.

Code Repository Link for the Project: https://drive.google.com/drive/folders/1XPvJsF5hdEJAkljS1h962PS9UWhAPKNw

REFERENCES

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT, 4171–4186.
2. Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 1: 11–21.
3. Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24, 5: 513–523.
4. Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In Proceedings of ECML, 137–142. Springer.
5. Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng. 1998. Improving text classification by shrinkage in a hierarchy of classes. In ICML, 359–367.
6. Kaggle. 2020. Natural Language Processing with Disaster Tweets (Dataset). Retrieved from https://www.kaggle.com/competitions/nlp-getting-started
7. Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. AIDR: Artificial Intelligence for Disaster Response. In Proceedings of WWW Companion, 159–162.
8. Sarah Vieweg, Amanda Hughes, Kate Starbird, and Leysia Palen. 2010. Microblogging during two natural hazards events: what Twitter may contribute to situational awareness. In Proceedings of CHI, 1079–1088. ACM.
9. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press.
10. Andreas Müller and Sarah Guido. 2016. Introduction to Machine Learning with Python. O'Reilly Media.