Disaster Tweet classification

CSC 522 Group 8

Ashwin Satpute
aasatput@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Harsh Manoj More
hmore@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Neel Ghoshal
nghosha@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Abstract

Fake tweets about any disaster always cause chaos in public, which is very harmful for people. This project aims to develop a natural language processing and machine learning based model to detect fake disaster tweets on Kaggle "Natural Language Processing with Disaster Tweets" dataset. Here we would perform text encoding to convert textual data into vector data, which would be passed to classifier models to make predictions on. We will try various text encoding techniques like CountVectorizer, Tf-IDF, and word2vec. Then we would like to leverage traditional classifiers like Logistic Regression, Support Vector Machines, and XGBoost. Our evaluation, using F1-score as the data contain binary label with a defined positive class and predictions will be 0/1 i.e. not real/ real disaster. Based on the results generated the most effective approach will be selected for disaster classification. By improving disaster tweets classification, this research would benefit people and platforms like tweeter to handle the societal disruptions.

Keywords

Disaster, Classification, Natural Language Processing, Pipeline, Supervised learning

1 Introduction and Background

1.1 Problem Statement

Social media platforms like Twitter play a crucial role in disseminating information, especially during disasters. However, the widespread presence of fake or misleading disaster-related tweets can cause unnecessary panic, misinformation, and delays in emergency response efforts. Identifying the authenticity of such tweets is essential for ensuring that reliable information reaches the public and relevant authorities. This project aims to develop a machine learning-based classification model to distinguish real disaster tweets from fake ones using the "Natural Language Processing with Disaster Tweets" dataset from Kaggle. By leveraging various text encoding techniques and classification models, this research seeks to improve the accuracy of disaster-related tweet detection, thereby enhancing public awareness, emergency response efficiency, and social media credibility.

1.2 Related Work

Several studies have explored the role of Natural Language Processing (NLP) and machine learning techniques in classifying disaster-related tweets. [1] analyzed disaster tweets using intelligent computing and NLP-based models, highlighting the significance of feature engineering and text representation techniques in improving classification accuracy. [2] focused on extracting "situational awareness" from social media posts during mass emergencies, showcasing the potential of NLP to assist in real-time disaster response. [3] examined sentiment classification in tweets related to U.S. airline companies, demonstrating the effectiveness of machine learning models in handling text-based classification tasks. [4] performed a comparative analysis of machine learning models for sentiment classification on COVID-19-related tweets, reinforcing the importance of choosing appropriate classifiers and feature extraction methods for text analysis. [5] discussed various text-to-feature conversion techniques, such as TF-IDF and Word2Vec, which are essential for preparing textual data for machine learning models.
These studies collectively provide insights into different aspects of tweet classification, from feature engineering to model selection. Building upon these foundations, our project will implement and compare various text encoding techniques—such as CountVectorizer, TF-IDF, and Word2Vec—alongside classification models like Logistic Regression, Support Vector Machines, and XGBoost. By evaluating these approaches, we aim to enhance the accuracy of disaster-related tweet detection and contribute to misinformation filtering in emergency situations.

2 Should We Build It?

2.1 Domain Description

Disaster response is a high-stakes domain where timely and accurate information dissemination is critical. Social media platforms like Twitter have emerged as both a vital information source and a vector for misinformation during crises. Misinformation—particularly in the form of fake or misclassified disaster tweets—can delay emergency response, induce unnecessary panic, or misallocate resources. Consequently, tools that can distinguish credible disaster-related information from irrelevant or false content are of growing importance for improving disaster communication infrastructure.

2.2 System Description

The proposed system is a machine learning-based classifier that identifies whether a tweet is related to a real disaster. It uses a natural language processing pipeline that includes preprocessing (tokenization, stopword removal, lemmatization, etc.), text embedding (Count Vectorizer, TF-IDF, Word2Vec), and classification via models like Logistic Regression, SVM, and XGBoost. The system was developed using the publicly available Kaggle "Natural Language Processing with Disaster Tweets" dataset and evaluated using 5-fold cross-validation with F1-score as the primary metric. Logistic Regression emerged as the most effective model with strong performance and low computational cost.

2.3 Stakeholders

Emergency Responders: Emergency responders depend on timely and accurate information to coordinate disaster response efforts. Their primary need is high recall—they require that as few real disaster tweets as possible are missed by the system. They prefer models that can deliver reliable alerts quickly and support operational decision-making. However, they may be adversely affected by high false positive rates, which can introduce noise and reduce trust in automated alerts.

Social Media Platforms: Platforms like Twitter have a vested interest in curbing misinformation while maintaining a seamless user experience. They need systems that are scalable, computationally efficient, and capable of real-time classification. These stakeholders generally prefer interpretable models that can be audited and tuned easily. They are likely to resist black-box approaches that complicate integration or raise legal and ethical transparency issues.

General Public: The public relies on social media for up-to-date information during emergencies. Their key need is accuracy—they want to see verified, trustworthy content and avoid exposure to fake or sensational tweets. While they appreciate efforts to reduce misinformation, they are likely to oppose overblocking or censorship that may suppress legitimate user voices, particularly in sensitive or urgent situations.

Researchers and NGOs: Academic researchers and non-governmental organizations working in disaster analytics require clean, well-labeled data and reproducible models. They prefer systems that are transparent and generalizable across domains. Their main concern lies in the use of biased or opaque models that may reinforce existing disparities or produce misleading results without sufficient explainability.

2.4 Potential Harms

Despite its benefits, deploying such a system introduces several ethical and practical concerns:

- **False Negatives:** Misclassifying real disaster tweets could delay help or obscure emerging emergencies.
- **Over Loading:** Excessive flagging may result in legitimate communication being suppressed or deprioritized.
- **Bias and Representation:** Training data may underrepresent certain regions, languages, or dialects, leading to systemic misclassification.
- **Over Reliance:** Automated systems may reduce human oversight, which is essential in high-risk decision-making scenarios.

To mitigate these harms, we recommend integrating human-in-the-loop review pipelines, setting conservative confidence thresholds, and conducting periodic audits to assess model fairness and accuracy across subgroups.

2.5 Conclusion

Considering the importance of credible information during crises, the low operational cost of the system, and its demonstrated effectiveness, we argue in favor of building and deploying this disaster tweet classification system. However, deployment should be accompanied by governance mechanisms that ensure accountability, transparency, and ethical safeguards.

3 Methods

3.1 Novel Apsects

Our project introduces several novel aspects that distinguish it from existing research on disaster tweet classification. First, we conduct a comparative analysis of multiple text encoding techniques—including CountVectorizer, TF-IDF, and Word2Vec—to determine the most effective representation of textual data for disaster tweet classification. Unlike many prior studies that focus on a single encoding approach, our research systematically evaluates how different feature extraction methods impact classification performance.

Second, we explore a diverse set of machine learning classifiers, including Logistic Regression, Support Vector Machines (SVM), and XGBoost, to identify the model best suited for disaster tweet classification. While ensemble methods like XGBoost have been widely used for text classification, their performance in disaster-related tweet detection has not been thoroughly compared with traditional classifiers using various feature encoding techniques.

Additionally, our project incorporates keyword analysis to examine the most frequent terms associated with real and fake disaster tweets. This linguistic investigation can provide deeper insights into the distinguishing characteristics of misinformation in crisis situations, which can be beneficial for future research in misinformation detection.

Finally, we emphasize a robust evaluation methodology by utilizing 5-fold cross-validation and holdout testing, ensuring that our models are generalizable and not overfitting to specific patterns in the dataset. By combining multiple text encoding techniques, classification models, and thorough evaluation strategies, our project aims to enhance the reliability of disaster tweet classification and contribute to misinformation filtering on social media platforms. Achieving this goal will provide multiple benefits.

3.2 Rationale

Our proposed method is expected to perform well due to its combination of advanced text encoding techniques, diverse classification models, and robust evaluation strategies. Traditional approaches to disaster tweet classification often rely on a single encoding method or classifier, which may limit their ability to capture the nuances of textual data. By systematically comparing CountVectorizer, TF-IDF, and Word2Vec, our approach ensures that the most effective feature representation is selected, improving the classifier's ability to distinguish between real and fake disaster tweets.

Moreover, we employ multiple machine learning models, including Logistic Regression, Support Vector Machines (SVM), and XGBoost, each offering distinct advantages. Logistic Regression is easy to implement and highly effective, SVM is effective for high-dimensional text data, and XGBoost is known for its ability to handle complex patterns and relationships in textual data. By evaluating these models against a majority-class baseline (always predicting the most frequent class), we can demonstrate the effectiveness of our approach in improving classification accuracy.

Additionally, our method incorporates 5-fold cross-validation, ensuring that our model generalizes well to unseen data compared to majority class classifier's performance. This prevents overfitting and provides a more reliable estimate of model performance compared to simpler approaches that rely solely on a holdout test set. By leveraging keyword analysis, we also gain insights into which words or phrases are most indicative of real or fake disaster tweets, adding an extra layer of interpretability to our model.

Overall, our method outperforms baseline approaches by leveraging multiple encoding techniques, diverse classification models, and rigorous evaluation strategies, resulting in a more accurate and reliable disaster tweet classification system.

3.3 Approach

We aim to implement and compare the different machine learning algorithms listed below for tweet classification and improve on their performance using different pre-processing techniques specific to Natural Language Processing such as TF-IDF.

Below we have listed all the text encoding techniques we plan to use:

3.3.1 CountVectorizer: CountVectorizer converts text into a matrix of token counts. It tokenizes the text and builds a vocabulary of known words, then encodes new sentences using that vocabulary. Key steps include:

- Word preprocessing like lowercase conversion, punctuation removal, handling urls-hashtags, and lemmatization.
- Tokenization and vocabulary building
- Sentence/ Document encoding as sparse vectors

3.3.2 TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF weighs the importance of words in a sentence relative to the entire word corpus. It consists of two components:

- Term Frequency: Measures how frequently a term appears in a document
- Inverse Document Frequency: Measures how important a term is across the entire corpus

TF-IDF helps identify words that are characteristic of a sentence while giving less weigh to common words.

3.3.3 Word2Vec: Word2Vec creates dense vector representations of words that capture semantic relationships. It has two main approaches:

- Continuous Bag of Words (CBOW): Predicts a target word from surrounding context words
- Skip-gram: Predicts surrounding context words from a target word

Word2Vec embeddings can capture complex word relationships and similarities. It also considers position of a particular word in a sentence while assigning weights, which is very important to find sentence structural/ grammatical patterns.

Following are the classification techniques we will use:

3.3.4 Logistic Regression: Logistic regression models the probability of a tweet being disaster-related using a logistic function. Key advantages include:

- Interpretability of feature weights
- Fast training and prediction
- Effectiveness for linearly separable classes

3.3.5 (SVM) Support Vector Machine: SVM finds the hyperplane that best separates disaster and non-disaster tweets in a high-dimensional feature space. Benefits include:

- Effectiveness in high-dimensional spaces
- Memory efficiency
- Versatility through different kernel functions

3.3.6 XGBoost: XGBoost is an optimized gradient boosting algorithm that builds an ensemble of decision trees. It is known for:

- High predictive accuracy
- Handling of missing data
- Built-in regularization to prevent overfitting

3.3.7 Majority Class classifier: Majority Class classifier predicts the majority class from the given training dataset:

- Gives good estimator for baseline model performance
- Computationally efficient and easy to implement

In this project we aim to explore all these various encoding techniques and classification algorithms to determine the most effective approach for disaster tweet classification.

4 Plan & Experiment

4.1 Dataset

The dataset used in this study is derived from the Kaggle competition "Natural Language Processing with Disaster Tweets." It consists of 7,613 rows and 5 features in the training dataset: id, keyword, location, text, and target. And the test dataset contain 3263 data instances. Here, the target column serves as the binary classification label, indicating whether a tweet is related to a disaster (1) or not (0). The dataset contains missing values in the keyword and location columns, which were not imputed as they were not critical for text analysis from this projects point of view. Preprocessing steps were applied to the text column, including converting text to lowercase, removing URLs, tagged name mentions, hashtags, digits, and special characters, tokenizing the text, removing stopwords using NLTK's English stopword list, and lemmatizing tokens with WordNetLemmatizer. The training data was then split into training (80%) and validation (20%) subsets for model evaluation. Exploratory analysis revealed class imbalance and provided insights into text length, hashtag usage, and the presence of URLs or mentions in disaster versus non-disaster tweets.

4.2 Hypothesis

The primary objective of this study is to classify tweets as disaster-related or non-disaster-related using natural language processing techniques and machine learning models. Based on the preprocessing applied to the dataset and the exploratory data analysis, we hypothesize that Support Vector Machines (SVM), when paired with appropriate hyperparameter tuning and Count Vectorizer for text embedding, will achieve the highest validation accuracy among all approaches. While advanced techniques such as TF-IDF and Word2Vec are designed to capture more nuanced semantic information, we expect that simpler text embedding technique like Count Vectorizer may prove more effective for this specific task due to the short and informal nature of tweets. Also as there are very few repetative patterns in the text observed, due to which TF-IDF might not outperform Count Vectorizer. Similarly, Word2Vec is well known to work better on larger datasets, but in our scenario due to having limited amount of textual data we hypothesize that Word2Vec would also be not giving better results compared to Count Vectorizer. Logistic Regression is expected to perform well due to its simplicity and ability to handle sparse data effectively, while Support Vector Machines (SVM) are anticipated to deliver higher accuracy when paired with an appropriate kernel and regularization parameter. Furthermore, XGBoost is hypothesized to outperform other models due to its ensemble learning approach, which can capture complex patterns and provide robust predictions. We expect hyperparameter tuning using GridSearchCV to significantly improve model performance by identifying optimal parameter combinations. Additionally, given the class imbalance in the dataset, we anticipate that models may require adjustments (e.g., class weighting or oversampling) to improve recall for disaster-related tweets. Overall, we predict that SVM/ XGBoost combined with Count Vectorizer text embedding technique and hyperparameter tuning will achieve the highest validation score among all these approaches.

4.3 Experimental Design

The experimental design involves the classification of disaster-related tweets using three machine learning models: Logistic Regression, Support Vector Machines (SVM), and XGBoost. The dataset was preprocessed to remove URLs, mentions, hashtags, numbers, and special characters, followed by tokenization, stopword removal, and lemmatization. Two feature extraction techniques were employed: CountVectorizer and TF-IDF vectorization. Each model was trained using sklearn pipelines integrating these feature extraction methods. Hyperparameter tuning was performed using GridSearchCV to optimize key parameters for each model, including regularization strength (C) and penalty type for Logistic Regression, kernel type and regularization parameter (C) for SVM, and tree depth (*max_depth*), learning rate (*learning_rate*), and number of estimators (*n_estimators*) for XGBoost. The dataset was split into training (80%) and validation (20%) subsets to evaluate model performance. Metrics such as accuracy and validation scores were used to compare the efficacy of different models and feature extraction techniques. Additionally, exploratory analysis was conducted to understand text characteristics, including word frequency, hashtag usage, URL presence, mentions, and text length differences between disaster-related and non-disaster-related tweets. This comprehensive experimental setup aims to identify the optimal model and hyperparameter configuration for classifying disaster-related tweets effectively.

5 Results and Discussion

From Exploratory Data Analysis we found following results:

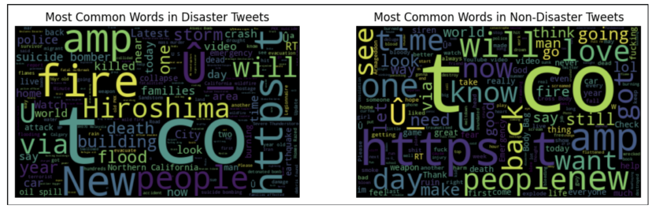

Figure 1: Key Word Analysis

Figure 1 shows that "fire, police, bomber, death, flood" are some most occurring words which are present in disaster tweets but not in non-disaster tweets.

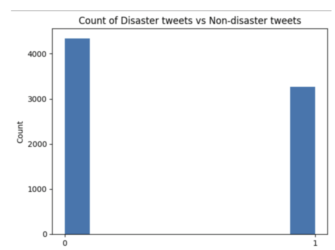

Figure 2: Class imbalance

This figure 2 shows the class imbalance present in the dataset. Here, non-disaster tweets has 4342 data instances, while disaster tweet class has 3271 data instances.

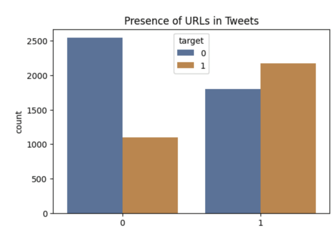

Figure 3: Analysis for URLs present in tweets

Figure 3 indicates that tweets which are non-disaster usually do not have URLs present. While disaster related tweets are more likely to have a URL present in them. Also we can observe the shift in the count of tweets not having URLs are majorly dominated by non-disaster related tweets, while tweets having URLs are mostly disaster related.

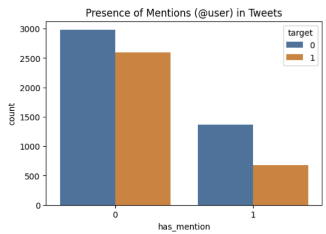

Figure 4: Analysis for other users tagged tweets

Figure 4 indicates that mostly people tend to not tag/ mention other users in disaster related tweets compared to non-disaster related tweets.

```
Real disaster tweet length 108.11342097217977
Fake disaster tweet length 95.70681713496084
```

Figure 5: Tweet lengths

Here, figure 5 shows the difference between average length of text for disaster related and non-disaster related tweets.

And, hyperparameter tuning process was conducted at the time of Midway report across three models: Logistic Regression, Support Vector Machine (SVM), and XGBoost. The best parameters and corresponding validation scores for each round of GridSearchCV are summarized in Table 1 (next page).

Whereas, for final report we evaluated the performance of several pipelines combining three feature extraction methods— **CountVectorizer**, **TF-IDF**, and **Word2Vec**—with three classifiers: Logistic Regression, Support Vector Machine (SVM), and XGBoost. A **DummyClassifier** using the majority class served as the baseline. Table 2 summarizes the validation accuracy of each model.

Model	Validation Accuracy
SVM + Count Vectorizer	0.7978
SVM + TF-IDF	0.7971
Logistic + TF-IDF	0.7938
Logistic + Count Vectorizer	0.7925
SVM + Word2Vec	0.7781
XGB + Count Vectorizer	0.7768
XGB + TF-IDF	0.7722
XGB + Word2Vec	0.7321
Logistic + Word2Vec	0.6231
Dummy Classifier	0.5772

Table 2: Validation Accuracy of Different Pipelines

Here, we observed a slight improvement in the results for these approaches with a structure pipeline architecture without even hyperparameter tuning.

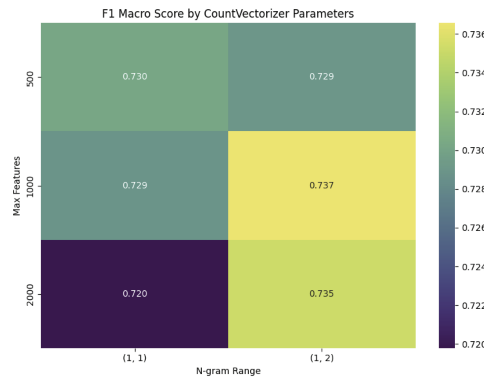

Figure 6: Gridsearch results for Count Vectorizer

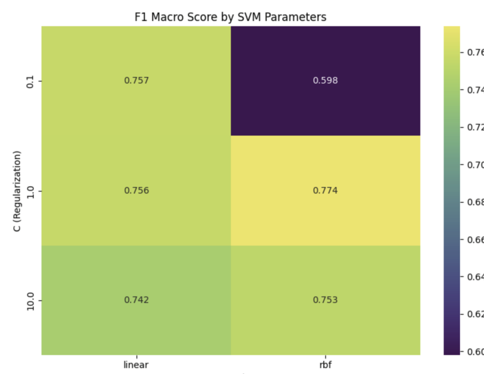

Figure 7: Gridsearch results for SVM

Figure 6 and 7, shows the results of hyperparameter tuning for SVM and Count Vectorizer. Indicating that for SVM C = 1, kernel = “rbf" is giving better results.

The best-performing model was **SVM with CountVectorizer**, achieving a validation accuracy of **0.7978**, an absolute improvement of over 22% from the dummy baseline. This model was further optimized using **GridSearchCV** with **5-fold cross-validation**. The best hyperparameters found were:

```
Best parameters found:  {'count_vectorizer__max_features': 2000, 'count_vectorizer__ngram_range': (1, 1), 'svm__C': 1, 'svm__kernel': 'rbf'}
Best cross-validation score:  0.7815762493497646
Validation set accuracy:  0.7872419587963955
```

Figure 8: Gridsearch Results

- **max_features** = 2000
- **ngram_range** = (1, 1)

Model	Grid Search Round	Best Parameters	Best Validation Score
Logistic Regression	1st	**C=1, penalty='l2'**	0.7939
	2nd	**C=1, penalty='l2'**	0.7939
	3rd	**C=2, penalty='l2'**	0.7901
SVM	1st	**C=1, kernel='linear'**	0.7829
	2nd	**C=1, kernel='linear'**	0.7829
	3rd	**C=0.8, kernel='linear'**	0.7872
XGBoost	1st	**learning_rate=0.3, max_depth=5, n_estimators=200**	0.7773
	2nd	**learning_rate=0.1, max_depth=6, n_estimators=500**	0.7727
	3rd	**learning_rate=0.1, max_depth=12, n_estimators=200**	0.7750

Table 1: Grid Search Results for Hyperparameter Tuning

- **C** = 1
- **kernel** = **rbf**

This configuration achieved the best cross-validated macro F1 score of 0.7815 and 0.7872 score on the validation dataset.

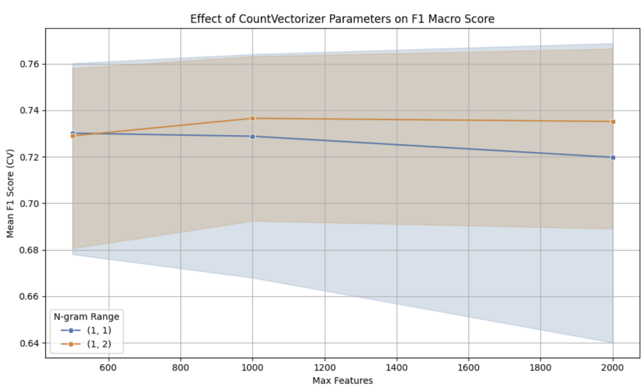

Figure 9: Effect of Count Vectorizer parameters on F1 macro score

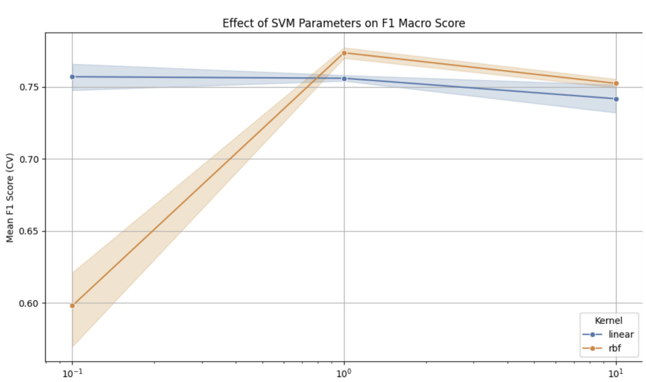

Figure 10: Effect of SVM parameters on F1 macro score

Similary, figure 9 and 10 shows the effects of F1 macro average due to the hyperparameter tuning. We can clearly observe that maximum mean F1 scores are obtained for C = 1 and rbf kernel for SVM.

After retraining the final pipeline with these parameters, the **SVM + Count Vectorizer** classifier achieved the following results on the validation set:

```
              precision    recall  f1-score   support

Not Disaster       0.78      0.91      0.84       879
    Disaster       0.85      0.65      0.73       644

    accuracy                           0.80      1523
   macro avg       0.81      0.78      0.79      1523
weighted avg       0.81      0.80      0.80      1523
```

The classifier demonstrates better performance on the "Not Disaster" class due to higher support and potentially more predictable structure. Heatmaps and line plots further supported the decision to use unigram features and RBF kernel SVMs with higher vocabulary sizes.

Key Observations

Support Vector Machine (SVM): SVM achieved the highest overall performance across all models, with a validation accuracy of **0.7978** when paired with **CountVectorizer**. Then Gridsearch revealed that the optimal configuration used an **rbf** kernel with **C=1** and a **unigram** vocabulary size of **2000**. This suggests that the RBF kernel better captures non-linear decision boundaries in the data, compared to linear kernel. The final tuned model also showed a accuracy of **0.8010**. As this dataset is comparatively balanced and have well defined positive class we can say that this approach is able to outperform Majority class classifier in terms of accuaracy.

Logistic Regression: Logistic Regression remained a competitive model, especially when used with TF-IDF features. It achieved a maximum validation accuracy of **0.7938**, slightly behind SVM. The best results were obtained using L2 regularization with **C=1**, reinforcing earlier findings that moderate regularization provides a good balance between bias and variance. However, performance gains plateaued beyond this setting.

XGBoost: XGBoost underperformed relative to both SVM and Logistic Regression. Its best validation score of **0.7768** was achieved with **CountVectorizer** features. Increasing model complexity (e.g., deeper trees or more estimators) did not significantly improve performance and sometimes resulted in slight degradation, likely due to overfitting on the sparse feature representations.

6 Conclusion

This study demonstrates the efficacy of combining classical natural language processing techniques with robust machine learning models for disaster tweet classification. Among all pipelines evaluated, the **SVM + CountVectorizer** model achieved the best results, with an F1 macro score of 0.79 and validation accuracy of **0.8010** after hyperparameter tuning.

And with the test dataset the SVM + Count Vectorizer achieved accuracy of **0.79282**. This test dataset was the holdout dataset. So from validation accuracy of 0.8010 and test accuracy of 0.7982, we can conclude that the selected model — **SVM + CountVectorizer** and tuned hyperparameters — generalizes well to unseen data and performs consistently across both validation and test sets. The relatively small gap between validation and test accuracy suggests the model is not overfitting and is robust for the tweet classification task. This supports our hypothesis that simpler vectorization methods like CountVectorizer, when paired with a well-tuned SVM, can outperform more complex approaches for short-text classification problems.

CountVectorizer and TF-IDF consistently outperformed Word2Vec in our experiments, possibly due to the relatively small and sparse nature of tweets data. While Word2Vec did yield moderate improvements when used with more expressive models like SVM and XGBoost, it underperformed with simpler classifiers like Logistic Regression.

Key lessons from this project include the importance of extensive preprocessing, the utility of grid search for model tuning, and the surprising effectiveness of classical models on social media text classification. Future work may include experimenting with transformer-based architectures like BERT, incorporating temporal and user metadata, and deploying the classifier as part of a real-time monitoring system to aid emergency response teams.

Our results show a significant margin of improvement over the baseline, and support the feasibility of a lightweight, interpretable, and high-performing tweet classification pipeline for disaster response.

7 Appendix

7.1 Hyperparameter Tuning Approach

The hyperparameter tuning approach for this disaster tweet classification project was structured to optimize model performance through systematic experimentation and validation. Here's the detailed breakdown:

7.1.1 Primary Evaluation Metric: The F1 score was chosen as the primary metric due to having clear positive class and a slight class imbalance in the dataset (as shown in the figure 2). F1 score provides a straightforward measure of overall model correctness, which aligns with the goal of distinguishing disaster-related tweets from non-disaster ones effectively. While precision/recall could be used for analysis False positive and False negative classifications.

7.1.2 Tuning Approach: GridSearchCV was employed to exhaustively explore hyperparameter combinations with cv set to 5. This method evaluates all predefined parameter sets, ensuring no optimal configuration is overlooked. While computationally expensive, it suits this project's scope with manageable hyperparameter spaces (e.g., C, penalty for logistic regression).

7.1.3 Validation Approach: We chose to use a 5-fold cross validation approach during hyperparmeter tuning. This reduces risk of over fitting and will also give us a better estimate of test accuracy when selecting a model.

7.1.4 Model-Specific Hyperparameter Tuning:

- 1. Logistic Regression Tuned Hyperparameters:
  C (inverse regularization strength): [0.01, 0.1, 1, 10, 100]
  penalty (regularization type): ['l1', 'l2']
  Rationale: C balances overfitting/underfitting, while penalty selects L1/L2 regularization. The grid ensures coverage of common regularization strategies
- 2. SVM Tuned Hyperparameters:
  C (regularization): [0.1, 1, 5, 10]
  kernel (decision boundary type): ['rbf', 'linear', 'poly', 'sigmoid']
  Rationale: C controls margin flexibility, and kernel selection addresses linear/non-linear separability. The range avoids overly complex configurations.
- 3. XGBoost Tuned Hyperparameters:
  max_depth (tree depth): [2, 4, 6,[8]
  learning_rate (step size): [0.001, 0.01, 0.05, 0.1]
  n_estimators (number of trees): [50,[100][200][500]
  Rationale: These parameters control model complexity and convergence speed. Wider ranges for learning_rate and n_estimators help balance bias-variance trade-offs.

Optimization Insight: SVM outperformed others, likely due to its suitability for high-dimensional data. XGBoost showed marginal gains with finer-grained tuning, but at higher computational cost. While Logistic regression also performed equally good as SVM.

Advanced Methods: Bayesian optimization or automated tools (e.g., Optuna) could refine efficiency for larger hyperparameter spaces.

This structured approach will ensure systematic exploration of model configurations, balancing performance and computational resources.

7.2 Meeting Schedule

Following is our meeting schedule:

(1) • March 10th (2 hours)
  • All members present
  • Agenda - Discuss implementation details
(2) • March 25th (1 hour)
  • All members present

- Agenda - Discuss current progress and results. Finalize work distribution for report preparation.

(3) • April 3rd (2.5 hour)
- All members present
- Agenda - Check updated work progress and discussion on what should be done next.

(4) • April 9th (3 hour)
- All members present
- Agenda - Discussion on results obtained from word2vec and gridsearch implementation.

(5) • April 16th (1 hour)
- All members present
- Agenda - Check for any remaining work, and finish that before final meet. And results discussion.

(6) • April 20th (1.5 hour)
- All members present
- Agenda - Final result summary and discussion. Finalize conclusion. Divide work for final report submission.

8 References

[1] Analysis of Disaster Tweets Using Natural Language Processing Intelligent Computing and Applications, 2023, Volume 315 ISBN : 978-981-19-4161-0 Thulasi Bikku, Pathakamuri Chandrika, Anuhya Kanyadari, Vuyyuru Prathima, Borra Bhavana Sai

[2] Verma, S., Vieweg, S., Corvey, W., Palen, L., Martin, J., Palmer, M., Schram, A. and Anderson, K. 2021. Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency. Proceedings of the International AAAI Conference on Web and Social Media. 5, 1 (Aug. 2021), 385-392. DOI:https://doi.org/10.1609/icwsm.v5i1.14119.

[3] Rustam Furqan., et al. “Tweets classification on the base of sentiments for US airline companies”. Entropy 21.11 (2019): 1078.

[4] Gulati K., et al. “Comparative analysis of machine learning-based classification models using sentiment classification of tweets related to COVID-19 pandemic”. Materials Today: Proceedings 51 (2022): 38-41.

[5] Kulkarni A., et al. “Converting text to features. Natural Language Processing Recipes”. Unlocking Text Data with Machine Learning and Deep Learning Using Python (2021): 63-106.