Natural Language Processing with Disaster Tweets

Shiva Madhav Adusumilli
(002775471)

Fiyinfoluwa Dideoluwa
(002828214)

Amirthavarshani Mahadevan
(002847245)

Jarvis Chapman
(002180520)

Milaan Williams
(002661017)

Flynn Heise
(002611186)

Abstract

This project uses machine learning and deep learning techniques to classify disaster-related tweets. The primary objective was to identify whether a tweet is related to a disaster. To achieve this, we implemented and fine-tuned multiple models, including multi-channel CNNs, RNNs with LSTMs, and transformer-based models like BERT. Each model was evaluated for its performance in terms of accuracy, loss, and F1 score. The project highlights challenges such as overfitting and hyperparameter optimization, and documents strategies adopted to overcome them.

1. Introduction

Social media has become an important communication channel for staying updated on critical events such as disasters. However, the sheer magnitude and diversity of posts it is hard to tell what is real. Platforms like Twitter, Instagram, and Facebook play crucial roles in circulating both essential and distracting information during disasters. Accurate classification of such information can help streamline emergency response efforts. This project utilizes a labeled dataset of tweets to build and evaluate effective models capable of performing this classification task.

2. Literature Review

Dharrao et al., 2024 proposed a method that synergizes BERT Embedding and a CNN-based deep learning model, to effectively classify tweets related to disastrous scenarios. In this study, different machine learning algorithms such as decision tree, logistic regression, and CNN were used. The CNN model employed with BERT embedding performed best. The parameters of the model were further optimized using different optimizers such as Adam, AdaGrad, and a gradient-based optimizer called RMSProp. The CNN model showed an improved performance from 80% (when used alone) to 83% (when it was combined with BERT Embedding).

Anh Duc Le conducted research with the objective of attempting to decipher the NLP problem posed by many. The problem was to determine whether a person was in danger or not based on a tweet/ social media post. Two traditional approaches, Logistic regression and SVM were used including deep learning models like CNN and LSTM. In the first solution method, the data preprocessing used four methods: Count Vector, TF-IDF, CBoW, and Skip-gram Vector. In the second solution method, BERT is used specifically BERT-LARGE with a customized classification task tuning the parameters hidden layers, hidden size, and attention heads. The BERT-Large method was also fine-tuned to minimize loss. The results of the first solution show that frequency-based embedding takes an advantage over prediction-based embedding with a 70%-71% F1 score, proving to be superior for this type of classification. The result of the second solution using BERT deep learning performed very well with an 88% F1 score.

In a paper by Suriyamurth et. Al., a dataset containing 10,000 tweets from a Kaggle database was used. The tweets were labeled as disaster-related - 1 or non-disaster-related- 0. The text was tokenized, then converted to lowercase, and then finally cleaned of all special characters to make it easier to use. Stop words were also removed and lemmatization was applied to

group words that had similar meanings. A TF-IDF transformer was used to calculate term frequencies and inverse document frequencies for the tweets. Classification Algorithms: Precision, accuracy, and F1-score were used to evaluate the models.

3. Methodology

The block diagram of the implementation architecture for this study is shown in Figure 1 below.

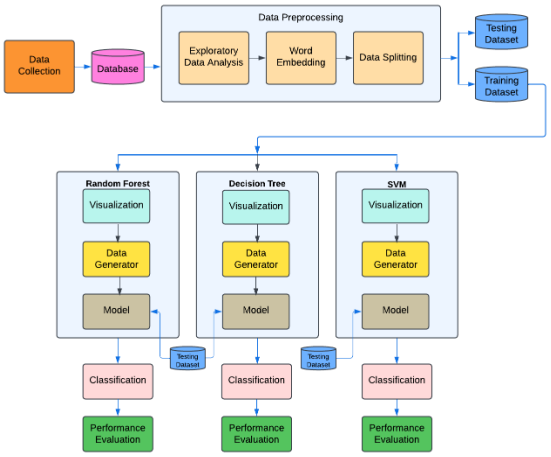

Figure 1: Block Diagram of the System

3.1 Dataset

The dataset used in this study is the Kaggle NLP Disaster Tweets dataset. The dataset includes 7,613 sample tweets labeled as disaster (1) and non-disaster (0) in the training data and another 3,263 sample tweets as test data. The dataset contained some missing data which were handled accordingly. The 'location' feature was dropped since it was not important for our analysis while we deleted the missing records in the 'keyword' feature.

2 rows | 2 rows × 2 columns

	Missing Data	Missing Percentage
keyword	61	0.801261
location	2533	33.272035

Figure 2: Missing values in the dataset

3.2 Exploratory Data Analysis (EDA)

EDA ensures data integrity, highlights key insights, and informs preprocessing steps, helping to optimize the dataset for effective model building and improved system performance. Techniques used on our dataset include visualizing distributions of the target class, lengths of tweets, common stop words, special characters, etc. Some of the results are shown in the following figures.

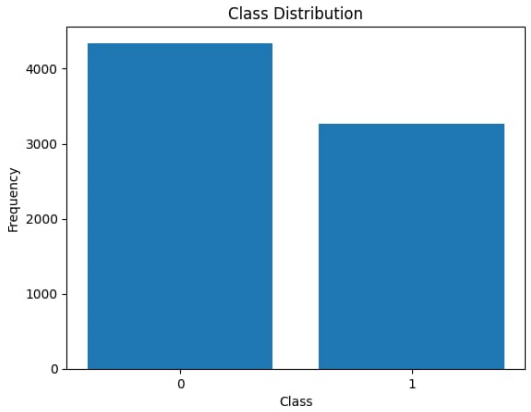

Figure 3: Distribution of target class

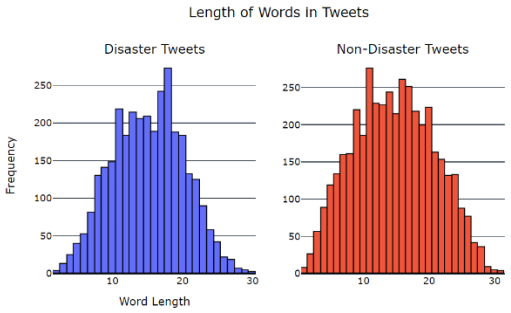

Figure 4: Length of words in tweets

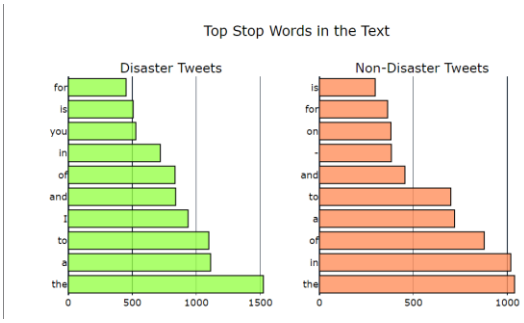

Figure 5: Top 10 Stop words in tweets

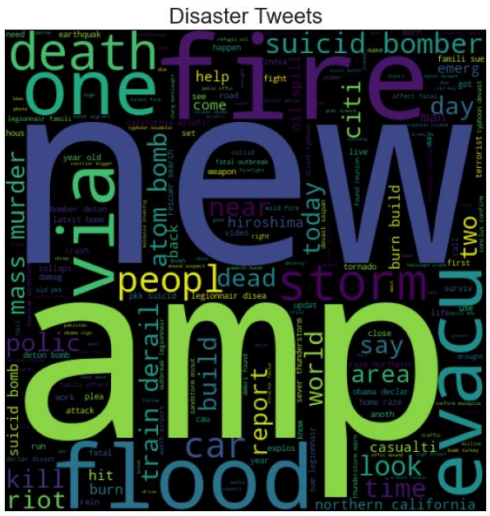

Figure 6: Word cloud for disaster tweets

3.3 Data Cleaning & Preprocessing

After performing EDA, there was a need to clean our dataset. The general cleansing steps for each Tweets are constructed as below:

- Removed URLs
- Removed HTML tags
- Replaced digits with an empty string
- Removed Emoticons, Symbols, Pictographs, Transport & Map Symbols, and Flags
- Removed Twitter mentions
- Remove punctuation and special characters
- Filtered out stop words
- Removed one-word tweets

3.4 Word Embedding:

For word embedding, we used BERT - Bidirectional Encoder Representations from Transformers. It is designed to pre-train bidirectional representations from unlabeled text by conditioning left and right context at the same time. As a result, BERT can create a state-of-the-art model for NLP tasks.

3.5 Modeling

The dataset was split into 70% and 30% for training and testing data respectively. Both traditional and deep learning models were used for the implementation. The traditional models used include:

- Logistic Regression
- SVM
- Random Forest
- Decision Tree

The deep learning models used include:

- Multi-channel Convolutional Neural Network (CNN) model
- Recurrent Neural Network (RNN) with LSTM

4. Results and Discussion

4.1 Traditional Models

Accuracy and F1-Score were used in evaluating the performance of the models.

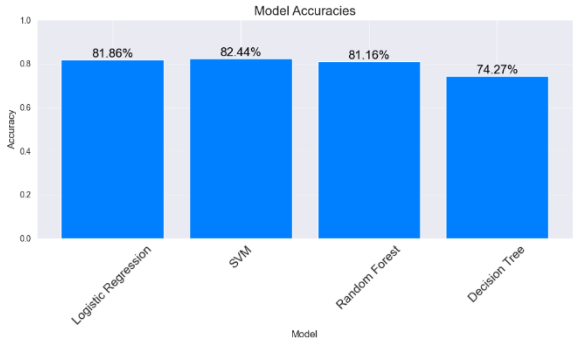

Figure 7: Accuracies of Traditional Models

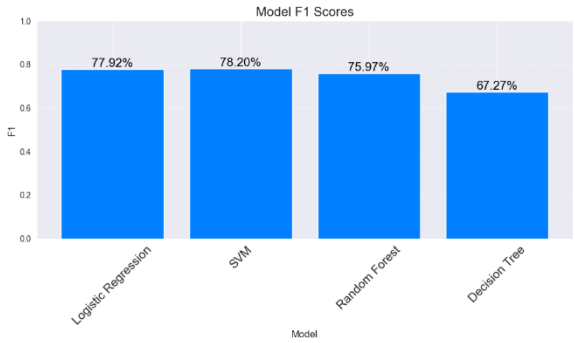

Figure 8: F1-score of Traditional Models

Of the 4 traditional machine learning algorithms that were used, the Support Vector Machine (SVM) algorithm performed best. Below is the ranking of their performance based on the 'accuracy' evaluation metric in descending order:

1. SVM - 82.4%
2. Logistic Regression - 81.8%
3. Random Forest - 81.0%
4. Decision Tree - 74.2%

The ranking of their performance based on the F1-score in descending order is:
1. SVM – 78.2%
2. Logistic Regression – 77.9%
3. Random Forest – 75.9%
4. Decision Tree – 67.2%

In both cases, the SVM model performed best.

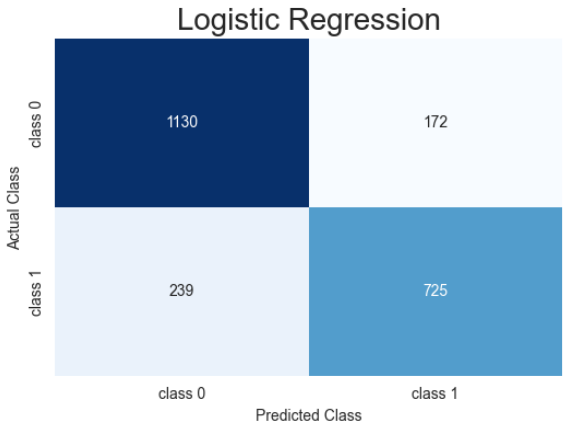

Figure 9: Logistic Regression Confusion Matrix

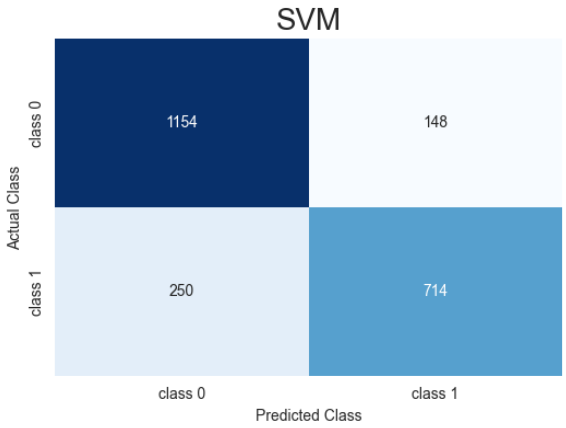

Figure 10: SVM Confusion Matrix

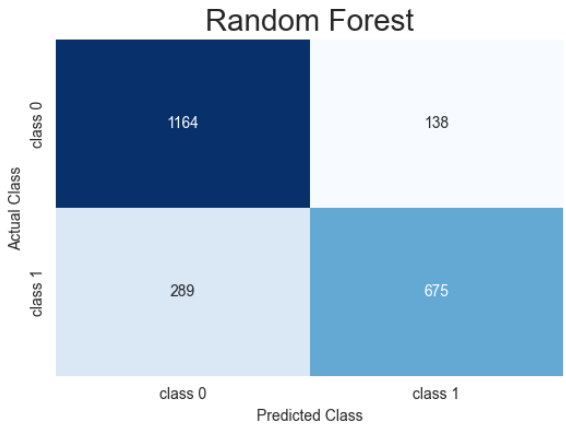

Figure 11: Random Forest Confusion Matrix

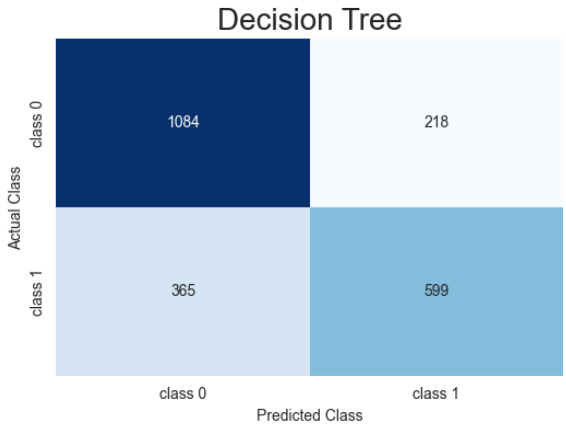

Figure 12: Decision Tree Confusion Matrix

4.2 Deep Learning Models

Multi-Channel CNN : The model was trained with the hyperparameter shown in Figure 13 below.
An average F1 score of 0.785 and an accuracy of 79% was obtained.

```
# Hyperparameters
filters = 256
kernel_sizes = [3, 5, 7]
dropout_rate = 0.75
learning_rate = 0.00005
batch_size = 32
epochs = 10
```

Figure 13: Hyperparameters for multi-channel CNN

RNN with LSTM : The model was trained with the hyperparameter shown in figure 14 below. An average F1 score of 0.67 and accuracy of 69% was obtained.

```
# Refined Hyperparameters
MAX_WORDS = 20000          #
MAX_LEN = 100              #
EMBEDDING_DIM = 128        #
LSTM_UNITS_1 = 64          #
LSTM_UNITS_2 = 32          #
DENSE_UNITS = 32           #
DROPOUT_RATE = 0.70         #
LEARNING_RATE = 0.00007     #
BATCH_SIZE = 32            #
EPOCHS = 5                 #
EARLY_STOPPING_PATIENCE = 2
```

Figure 14: Hyperparameters for RNN with LSTM

Model	Validation Accuracy	Validation F1-Score
Multi-Channel CNNs	79%	78%
RNNs and LSTMS	69%	67%

Figures 15: Performance Summary for deep learning models

6. Conclusion:

The project demonstrated the effectiveness of various models for classifying disaster-related tweets. Of the 4 traditional machine learning algorithms that were used, the Support Vector Machine (SVM) algorithm performed best with an F1-Score of 78% and an Accuracy Score of 82%. This shows that using BERT with Traditional Classification methods can work well to classify disasters and non-disasters.

REFERENCES

1. Deepak Dharrao, Aadithyanarayanan MR, Rewaa Mital, Abhinav Vengali, Madhuri Pangavhane, Satpalsing Rajput, Anupkumar M. Bongale, “An efficient method for disaster tweets classification using gradient-based optimized convolutional neural networks with BERT embeddings,” Volume 13, 2024, 102843, ISSN 2215-0161, https://doi.org/10.1016/j.mex.2024.102843.
2. Anh Duc Le, “Disaster Tweets Classification using BERT-Based Language Model,” 2022 10.48550/arXiv.2202.00795. https://doi.org/10.48550/arXiv.2202.00795
3. Suriyamurthi, Deepalakshmi & Thambusamy, Velmurugan. (2023). Classification of Disaster Tweets Using Natural Language Processing Pipeline. 5.