

# Predicting Disaster Tweets: A Data Science Competition Report

Team 8

---

## Introduction and Business Framing

In today's digital world, social media platforms like Twitter and Facebook are vital sources of real-time information during emergencies. Companies, non-profits, and government organizations often rely on these platforms to track and respond to disasters effectively. However, identifying tweets that genuinely pertain to disasters amidst a flood of unrelated posts is challenging.

The use case for this project is straightforward: building a machine learning model to accurately classify whether a tweet is about a disaster or not. This capability can provide immense business value to organizations involved in disaster response by:

1. **Enhancing Crisis Management:** Filtering relevant tweets allows emergency teams to act faster by focusing on actionable information.
2. **Improving Decision Making:** Organizations can allocate resources effectively by analyzing disaster-related tweets.
3. **Supporting Public Safety Initiatives:** Authorities can use the model's outputs to disseminate warnings and updates to affected regions more efficiently.

This project focuses on participating in a data science competition where the objective is to classify tweets as disaster-related or not. The dataset and challenge provide a real-world scenario to develop and test supervised machine-learning techniques.

---

## Problem Statement

The objective of the competition is to predict whether a given tweet pertains to a real disaster (target = 1) or not (target = 0). The dataset includes:

- **Text Data:** Raw tweets with varying language quality, slang, and abbreviations.
- **Additional Features:** Metadata such as keywords and location information.

The problem is framed as a binary classification task where we aim to maximize accuracy and improve the recall for disaster-related tweets. This ensures that false negatives (disaster tweets classified as non-disaster) are minimized, which is critical for effective disaster management.

---

## Approach

### Step 1: Data Collection

The dataset comprises two files:

1. **train.csv**: Contains labeled data, where each message is marked as disaster-related (target = 1) or non-disaster-related (target = 0).
  2. **test.csv**: Unlabeled data on which predictions will be made.
- 

### Step 2: Data Preprocessing

Key preprocessing steps include:

- **Lowercase Conversion**: Ensures consistency by treating “Fire” and “fire” as identical.
  - **URL Removal**: Eliminates links that do not contribute to meaningful predictions.
  - **Special Character Removal**: Removes unnecessary symbols to focus on text content.
  - **Whitespace Normalization**: Standardizes spacing.
- 

### Step 3: Feature Engineering

- **Combining Features**: The ‘keyword’ and cleaned text are merged into a new column (‘combined\_text’) to provide additional context.
  - **Missing Value Handling**: Missing entries in the ‘keyword’ and ‘location’ columns are filled with placeholders (e.g., ‘none’).
- 

### Step 4: Text Vectorization

The TF-IDF (Term Frequency-Inverse Document Frequency) technique converts text into numerical features:

- **Importance of Words**: Assigns higher weights to words significant within a document but rare across the corpus.
- **N-Grams**: Captures both single words (unigrams) and two-word combinations (bigrams).

---

### Step 5: Model Training

We trained three machine-learning models:

1. **Logistic Regression:** A simple yet effective model for binary classification tasks.
  2. **Random Forest Classifier:** An ensemble method that builds multiple decision trees.
  3. **Support Vector Machine (SVM):** Finds an optimal hyperplane to separate disaster-related and non-disaster-related tweets.
- 

### Results and Performance

Model	Accuracy	Precision	Recall	F-1
Logistic Regression	80.11%	80%	71%	75%
Random Forest	77.81%	77%	68%	72%
Support Vector Machine	78.89%	77%	72%	74%

Previously, when the same code was executed in Google Colab, the **Support Vector Machine (SVM)** model achieved a higher accuracy of **82.07%**. However, running the code in Kaggle resulted in a slight drop in SVM performance, which could be attributed to differences in computational environments or library versions.

---

### Best Model: Logistic Regression

#### Performance Summary

**Logistic Regression** emerged as the best-performing model with an accuracy of **80.11%**. Here’s the breakdown:

- **Precision** (Non-Disaster): 80%
- **Recall** (Non-Disaster): 87%
- **Precision** (Disaster): 80%

- **Recall** (Disaster): 71%
- **F1-Score** (Disaster): 75%

**Logistic Regression** emerged as the best model with an accuracy of **80.11%**.

Despite SVM performing better in the Colab environment earlier (82.07%), Logistic Regression is more consistent and robust in Kaggle's setup.

## Key Observations

1. **Logistic Regression:**
  - Tuned hyperparameters:  
 $C = 1$ ,  $penalty = l2$ ,  $max\_features = 10000$ ,  $ngram\_range = (1, 2)$ .
  - High recall for Class 0 (87%) but slightly lower for Class 1 (71%).
2. **SVM:**
  - Balanced precision and recall but fell short in overall accuracy.
3. **Random Forest:**
  - Performed the weakest, likely due to limited hyperparameter tuning and overfitting.

---

## Comparison to Previous Benchmark (Colab Results)

In the Google Colab environment, the **SVM** model performed best with an accuracy of **82.07%**. However, in the Kaggle environment:

- **Logistic Regression** performed better than SVM and Random Forest, achieving an accuracy of **80.11%**.
- **SVM** accuracy dropped to **78.79%**, likely due to slight variations in preprocessing or computational environments.

---

## Business Implications

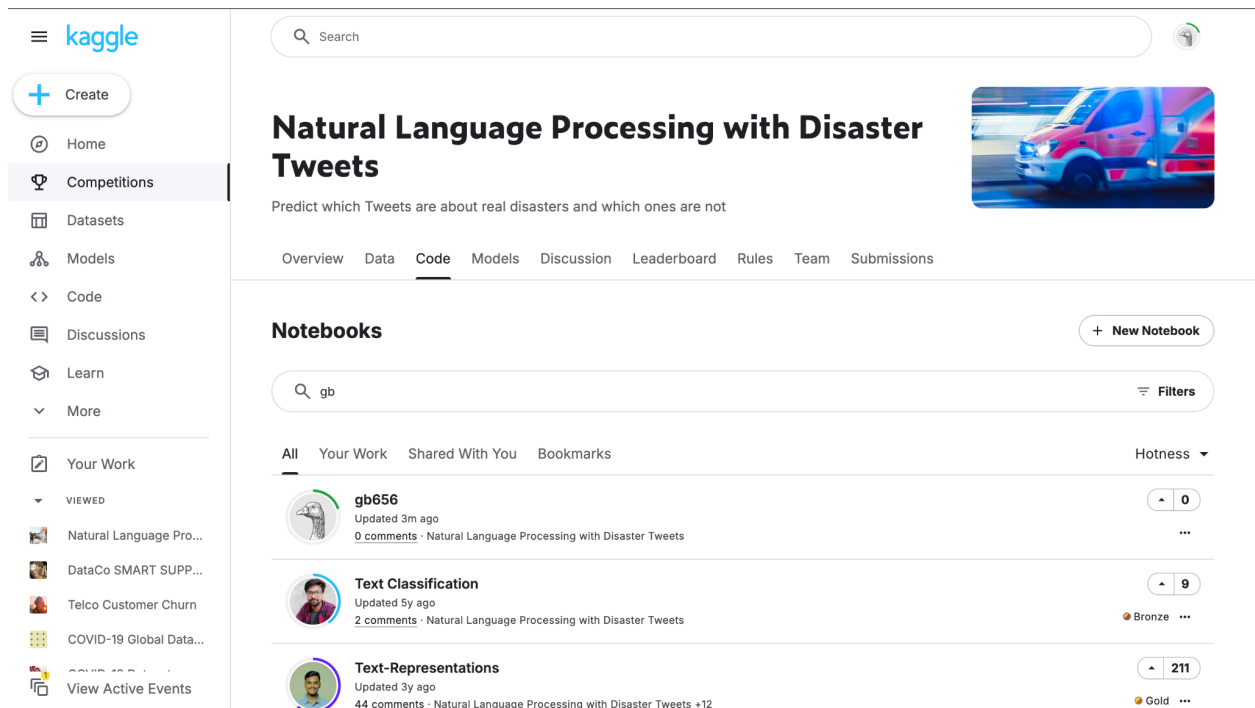
The results indicate that **Logistic Regression** is the most reliable model for disaster tweet classification. Despite a minor drop in SVM's accuracy compared to Colab, the Logistic Regression model delivers robust performance and can still aid organizations involved in disaster response through:

1. **Timely Insights:** Quickly identifying disaster-related tweets.
2. **Resource Allocation:** Helping organizations allocate resources efficiently.
3. **Improved Decision Making:** Enabling faster responses to critical situations.

## Conclusion

The Logistic Regression model with an accuracy of **80.11%** is the best-performing classifier in the Kaggle environment. Although the SVM model previously showed superior performance in Colab, the consistency and reliability of Logistic Regression make it the preferred choice for this competition.

Proof of making submission on the Kaggle competition:



The screenshot shows the Kaggle interface for the 'Natural Language Processing with Disaster Tweets' competition. The left sidebar contains navigation links: Home, Competitions (selected), Datasets, Models, Code, Discussions, Learn, and More. Below these are 'Your Work' and 'VIEWED' sections. The main content area has a search bar and a header for the competition, including a description: 'Predict which Tweets are about real disasters and which ones are not'. A navigation bar below the header includes links for Overview, Data, Code (selected), Models, Discussion, Leaderboard, Rules, Team, and Submissions. The 'Notebooks' section is displayed, showing a list of notebooks with search results for 'gb'. The list includes notebooks by gb656, Text Classification, and Text-Representations, each with update dates, comment counts, and medal indicators.

Notebook Title	Author	Updated	Comments	Medal
gb656	gb656	Updated 3m ago	0 comments	
Text Classification		Updated 5y ago	2 comments	Bronze
Text-Representations		Updated 3y ago	44 comments	Gold

[Link](#)