Infosys SpringBoard Internship 5.0

Project Report: Disaster Tweet Analyzer

Submitted to Mentor, Mr. Nitig Singh

By Mohamed Hanifa B

**1. Introduction**

In today’s interconnected world, social media platforms such as Twitter serve as vital sources of real-time information, especially during disaster events. Users frequently post updates about earthquakes, floods, wildfires, and other calamities, sharing firsthand information that can assist authorities, humanitarian organizations, and citizens. However, not all tweets tagged with disaster-related keywords are relevant. Many posts use disaster terminology in metaphorical or unrelated contexts, which can lead to the dissemination of incorrect or misleading information.

The objective of this project is to build a classification model that can differentiate between tweets that are genuinely related to disasters and those that are not. This involves processing a dataset of tweets and applying machine learning techniques to build a predictive model. The dataset under analysis consists of tweet texts along with binary labels indicating whether the tweet is related to a disaster.

**2. Dataset and Methodology**

**2.1 Dataset Overview**

The dataset comprises several columns:

* **id:** This is a unique identifier for each tweet in the dataset. It plays a key role in differentiating between tweets but is not used directly in the analysis or model building. This column ensures that each tweet can be referenced or retrieved unambiguously during the analysis process.
* **keyword:** This column contains specific keywords associated with disaster events. These keywords may offer clues about the nature of the tweet, helping to identify tweets that discuss events such as "earthquake," "flood," "hurricane," etc. In some cases, however, the keyword might be absent or unrelated to a disaster. This feature provides additional context and can be useful for boosting classification performance when combined with text data. Missing values are common, so its utility is limited but still valuable when present.
* **location**: This column records the geographical location from which the tweet was posted. The location may provide important context, such as whether a tweet originated from a region affected by a specific disaster. However, locations are often missing, inconsistent, or unspecified, which can reduce the overall importance of this feature in model building.
* **text**: The full content of the tweet is stored in this column, and it serves as the primary feature for our analysis. The text is unstructured, making it both the richest and most challenging aspect of the dataset. It contains not only the words and phrases used in the tweet, but also potential noise such as hashtags, mentions, emojis, and URLs. This raw text is where the majority of preprocessing and feature extraction efforts are concentrated to ensure that the machine learning models can effectively interpret and classify the tweets.
* **target**: The target column is a binary label indicating whether the tweet is disaster-related (1) or not (0). This is the key variable that we aim to predict using the model. A value of 1 signifies that the tweet is directly related to a disaster event, while a value of 0 means that the tweet either uses disaster-related terms metaphorically or is entirely unrelated to any real disaster.

**2.2 Text Preprocessing**

Preprocessing is a crucial step in ensuring that textual data is clean and ready for model training. The following steps were applied:

1. **Lowercasing**: All characters in the tweet were converted to lowercase to ensure consistency.
2. **Removing Stopwords:** Common words like "the", "is", "in", and "and" were removed to reduce data dimensionality.
3. **Removing Special Characters and Punctuation:** Hashtags, mentions, URLs, and punctuation were removed to focus on core tweet content.
4. **Tokenization and Lemmatization:** Tweets were split into individual words (tokens), and words were reduced to their base form (e.g., "running" became "run").
5. **Handling Common Words:** Frequent words irrelevant to classification were filtered out using a custom list.

For example, the tweet:  
Original: "there's a massive fire in california! 😱 #wildfire https://t.co/abc123"  
Cleaned: "massive fire california wildfire"

**2.3 Feature Extraction**

After preprocessing, we converted the cleaned text into numerical data:

* **Bag of Words (BoW)**: Each word is represented as a token, with frequency counts indicating its occurrence in each tweet.
* **TF-IDF (Term Frequency-Inverse Document Frequency)**: Builds on BoW, assigning weights to words based on their frequency across the dataset, highlighting important words.
* **Word Embeddings**: Dense vector representations of words (e.g., Word2Vec, GloVe) capture semantic meaning by taking context into account.

**3. Exploring the Data**

**3.1 Exploring the Dataset**

The dataset contains the following columns:

* **id**: Unique tweet identifier.
* **keyword**: Keywords related to disaster (may be blank).
* **location**: Tweet's origin (may be blank).
* **text**: Full text of the tweet.
* **target**: Label indicating disaster-related (1) or non-disaster-related (0).

**3.2 Exploring the Target Variable**

By visualizing the target classes:

* **Target Class 1** has ~3200 samples (disaster-related tweets).
* **Target Class 0** has ~4500 samples (non-disaster-related tweets).
* The dataset appears **balanced**.

**3.3 Analyzing Text Characteristics**

**Character Count:**

* The character count for both disaster and non-disaster tweets is in the **120-140 range**.

**Word Count:**

* Word count for disaster and non-disaster tweets is in the **15-20 range**.

**Average Word Length:**

* The average word length in disaster tweets is **7-7.5 characters**, while non-disaster tweets have an average word length of **4.5-5 characters**.

**3.4 Corpus and Keyword Analysis**

**Top Stop Words:**

* The most frequent stop word across tweets is **"the"**, while the least common in non-disaster tweets is **"for"**, and in disaster tweets, it’s **"is"**.

**Punctuation Analysis:**

* The **"-"** character is the most frequently used punctuation, appearing over 350 times in both classes.

**Null Values:**

* The **"Keyword"** and **"Location"** columns contain missing values:
  + 97.6% of **"location"** values and 24% of **"keyword"** values are missing in the training set.
  + The same pattern is observed in the testing set.

**Top Disaster-Related Keywords:**

* The most frequent keywords include **"derailment"**, **"wreckage"**, and **"outbreak"**.

**Top Disaster-Related Locations:**

* The most common location is **USA/United States**, while **Los Angeles** and **California** are less frequently mentioned.

**4. Removing the Garbage**

**4.1 Data Cleaning Operations**

We performed the following operations on the text column:

* Removal of URLs, HTML tags, emojis, and unnecessary characters.
* Conversion to lowercase.
* **Stemming** for TF-IDF, and **Lemmatization** for LSTM.
* Removal of words with length < 2.

**4.1.1 Removal of URLs, HTML Tags, Emojis, and Unnecessary Characters**

Tweets often include URLs, emojis, and various special characters like hashtags (#), mentions (@), and punctuations, which do not contribute significantly to the meaning of the text in the context of disaster classification. To eliminate this noise:

* **URLs**: Links to websites are common in tweets but provide little value in disaster detection. These were removed using regular expressions (re library in Python).

**Example:** Original tweet: "Check this out: https://example.com #disaster" Cleaned tweet: "Check this out: #disaster"

* **HTML Tags**: Although not very common in tweets, any HTML tags that might have been embedded in the text were removed.
* **Emojis**: Emojis, while expressive, do not provide substantial information for classifying tweets as disaster-related or non-disaster-related. For instance, a smiley face 😊 or a fire 🔥 emoji might add sentiment to the tweet, but these elements were excluded to focus on the textual content.

**Example:** Original tweet: "There’s a fire in California! 😱🔥" Cleaned tweet: "There’s a fire in California!"

* **Special Characters**: Symbols like hashtags (#), at-mentions (@), and punctuation were removed to simplify the text. This step was crucial in ensuring uniformity in the data.

**Example:** Original tweet: "Big #Earthquake in Mexico! @ReliefTeam" Cleaned tweet: "Big Earthquake in Mexico ReliefTeam"

**4.1.2 Conversion to Lowercase**

To standardize the text, all characters were converted to lowercase. This step is necessary to avoid case sensitivity issues during feature extraction. Without this step, words like "Fire" and "fire" would be treated as different tokens, which would introduce redundancy and noise into the model.

**Example:** Original tweet: "FIRE breaks out in downtown area" Cleaned tweet: "fire breaks out in downtown area"

By converting everything to lowercase, the vocabulary size is reduced, and the model can focus on the context and meaning of the words, rather than treating capitalized words as separate entities.

**4.1.3 Stemming for TF-IDF, and Lemmatization for LSTM**

In text preprocessing, two common techniques are used to reduce words to their base forms: **stemming** and **lemmatization**. These processes help to normalize the text by ensuring that different forms of the same word (e.g., "running" and "run") are treated as the same term.

* **Stemming (used for TF-IDF-based models)**: Stemming involves chopping off the ends of words to get the root form. For instance, words like "running," "runner," and "ran" are all reduced to "run." This is a fast and efficient technique for traditional feature extraction methods like **TF-IDF** (Term Frequency-Inverse Document Frequency), which focuses on the importance of words in relation to their frequency in the dataset.

**Example:**

* + Original words: "running," "runner," "ran"
  + Stemmed words: "run"
* **Lemmatization (used for LSTM models)**: Lemmatization is a more sophisticated approach that reduces words to their dictionary form (lemma). Unlike stemming, lemmatization considers the context of the word to determine its correct base form. This method is particularly useful for deep learning models like **LSTM** (Long Short-Term Memory) networks, where maintaining the grammatical structure of words is important.

**Example:**

* + Original words: "better," "running"
  + Lemmatized words: "good," "run"

For this project:

**4.1.4 Removal of Words with Length < 2**

To further clean the text and remove irrelevant content, we filtered out words that are less than two characters in length. These are typically words like "a," "I," or "it," which often do not provide valuable context for classification. Removing these short words helps in reducing the dimensionality of the text and improves model performance by focusing on more meaningful tokens.

**Example:** Original tweet: "I saw a big fire" Cleaned tweet: "saw big fire"

**4.2 Cleaning and WordCloud Analysis**

After cleaning the data, we generated a **WordCloud** to visualize common words and examined the top 50 words in the cleaned data. Further refinements were made to remove confusing terms such as **"like"**, **"get"**, and **"would"**.

**5. Results**

The dataset was prepared for machine learning models. Key insights include:

* Common disaster-related words like **"earthquake"**, **"flood"**, and **"fire"** were strongly indicative of the target label (1).
* Ambiguities (e.g., "fire" in both literal and metaphorical contexts) present challenges.

**6. Conclusion**

The preprocessing phase successfully transformed the raw tweet data into clean, structured content, ready for machine learning. By applying text cleaning, stopword removal, tokenization, and lemmatization, we significantly reduced noise and improved data consistency. This process streamlined the dataset by removing irrelevant elements like URLs and special characters, while treating variations of the same word uniformly through lemmatization.

Handling missing values in the **location** and **keyword** columns ensured that we could retain valuable features without compromising data quality. The cleaned dataset is now well-optimized for classification tasks, providing a solid foundation for training machine learning models. The insights gained during exploration, such as word frequency and tweet length, will help guide the next steps in model development.

**7. Future Objectives**

**The next phase of the project will focus on the following objectives:**

1. **Feature Extraction:**
   * Implement Bag of Words or TF-IDF to transform the text data into numerical features.
   * Experiment with word embeddings like Word2Vec or GloVe to enhance the model’s ability to understand word contexts.
2. **Model Building:**
   * Train classification models such as Logistic Regression, Random Forest, Support Vector Machines (SVM), and Neural Networks on the cleaned data.
   * Perform hyperparameter tuning and cross-validation to optimize the models.
3. **Model Evaluation:**
   * Evaluate model performance using metrics such as accuracy, precision, recall, F1 score, and confusion matrix.
   * Investigate misclassifications and improve model robustness.
4. **Deployment:**
   * If time permits, the model can be deployed as a web application or an API to classify new tweets in real-time.

**References:**

* <https://www.kaggle.com/code/akarsh1/step-by-step-analysis-of-disaster-tweets/notebook#If-you-like-my-work/notebook,-please-upvote-it>
* **Sakaki, T., Okazaki, M., & Matsuo, Y. (2010).**  
  "Earthquake shakes Twitter users: Real-time event detection by social sensors." *Proceedings of the 19th International Conference on World Wide Web*.
* **Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013).**  
  "Practical extraction of disaster-relevant information from social media." *Proceedings of the 22nd International Conference on World Wide Web Companion*, 1021-1024.
* **Imran, M., Mitra, P., & Castillo, C. (2016).**  
  "Social media in disaster mode: A case study of the 2011 Japan earthquake." *Proceedings of the 22nd International Conference on World Wide Web*.