**Disaster Tweet Analyzer Project Report**

**Internship 5.0**

**1. Introduction**

The **Disaster Tweet Analyzer** project leverages Natural Language Processing (NLP) and Artificial Intelligence (AI) to analyse tweets related to disasters. The primary goal is to classify tweets as either **disaster-related** or **non-disaster-related**, with the potential to further categorize them into specific disaster types such as earthquakes, floods, or wildfires. This project has real-world applications in **real-time disaster response**, allowing authorities to gather immediate information directly from users on the ground.

**2. Dataset and Methodology**

**Dataset Overview**

The dataset for this project, inspired by the **Kaggle Disaster Tweets dataset**, contains **11,370 tweets**. It is organized into the following columns:

* **id:** Unique identifier for each tweet.
* **keyword:** Disaster-related keywords (e.g., "earthquake", "ablaze").
* **location:** Location information where the tweet originated (if available).
* **text:** The actual tweet content.
* **target:** A binary label indicating whether the tweet is disaster-related (1) or not (0).

**Dataset Summary**

* **11,370 entries**
* **7952 tweets** have location information, though some entries contain missing or incomplete location data.
* **Keyword column** contains 219 unique disaster-related keywords.
* **Class Imbalance:** Around **19% of tweets are labeled as disaster-related** (target = 1), indicating an imbalanced dataset.

**Methodology**

**1. Data Preprocessing**

* **Text Cleaning:**
* Remove URLs, mentions (@username), hashtags, emojis, and special characters.
* Convert text to **lowercase**.
* Remove **stopwords** (e.g., “and,” “the”).
* Perform **lemmatization** to reduce words to their base form (e.g., "running" → "run").
* **Tokenization:** Split each tweet into individual words (tokens).
* **Handling Missing Data:**
* Location data is missing in **30% of the tweets**. For feature extraction, **tweets without location data** are still included but treated separately if needed.
* **Dealing with Imbalanced Data:**
* Since the dataset contains more **non-disaster-related tweets**, techniques like **oversampling the minority class** or **under sampling the majority class** will be applied.

**2. Feature Engineering and Extraction**

* **Word Embeddings:**  
  Use pre-trained embeddings like **Bag of Words (BoW)**, **GloVe**, or **BERT** to capture the semantic meaning of tweets.

**3. Model Selection**

* **Traditional Machine Learning Models:**
* **Logistic Regression**: For binary classification.
* **Random Forest**: Ensemble method improving accuracy through decision trees.
* **Deep Learning Models:**
* **Recurrent Neural Networks (RNNs):** Capture sequential dependencies in text.
* **Long Short-Term Memory (LSTM):** Specialized RNN for capturing long-term text dependencies.
* **Convolutional Neural Networks (CNN):** Apply convolution over text sequences, treating tweets as matrices.

**4. Model Training and Evaluation**

* **Train-Test Split:** Data is divided into **80% training** and **20% testing**.
* **Cross-validation:** Using **k-fold cross-validation** to ensure model robustness.
* **Performance Metrics:** Models are evaluated using **accuracy**, **precision**, **recall**, and **F1-score**.

**5. Real-time Tweet Classification**

* Use the **Twitter API** to fetch real-time tweets.
* Track keywords (e.g., **"earthquake," "fire"**) and apply **location-based filtering** if available.
* Visualize disaster-related tweets on a **dashboard**.

**6. Visualization and Reporting**

* **Data Visualization:** Plot trends in disaster-related tweets over time (e.g., spikes in earthquake tweets).
* **Geolocation Mapping:** Map tweets to their respective locations to **identify disaster hotspots**.
* **Dashboard:** Create a **real-time dashboard** for tracking disaster-related tweets with metrics such as tweet counts and disaster categories.

**3. Results**

* **Dataset Exploration:**
* **11,370 tweets** successfully loaded, with **19% labelled as disaster-related**.
* Preliminary analysis shows **frequent keywords** like “earthquake,” “ablaze,” and “flood.”
* **Visualization:** The project plots trends in disaster-related tweets, revealing significant clusters during major events.
* **Model Training:**
* Initial tests with **Logistic Regression** achieved **good performance**, though fine-tuning is needed.
* **LSTM models** performed well for longer tweets, while **CNNs** demonstrated surprising efficiency for text data.

**4. Conclusion**

The **Disaster Tweet Analyzer** project highlights the potential of **NLP and AI** in classifying and monitoring disaster-related tweets in real-time. Using both **traditional machine learning** and **deep learning models**, the system processes raw tweet data, transforms it into features, and classifies tweets into relevant categories. With **real-time tweet classification**, the project can act as an **early warning system** to help authorities respond swiftly to emerging disasters. The dashboard interface enhances situational awareness, enabling better disaster management.

**5. Future Objectives**

* **Optimize Feature Engineering:**
* Test different word embeddings, such as **BERT** and **GloVe**, to improve classification accuracy.
* **Fine-tune Models:**
* Tune hyperparameters (e.g., learning rate, epochs) for deep learning models.
* **Implement Real-Time Monitoring:**
* Integrate the **Twitter API** to fetch and classify real-time tweets.
* **Add Sentiment Analysis:**
* Incorporate sentiment analysis to identify **positive or negative sentiment** within disaster-related tweets.

**6. References**

1. **Kaggle Dataset and Approaches**:

* <https://www.kaggle.com/code/sugataghosh/natural-language-processing-with-disaster-tweets>
* This Kaggle competition provides the disaster tweets dataset and showcases a variety of approaches used by participants, including TF-IDF, logistic regression, RNNs, and BERT.

2. **GitHub Repositories with Disaster Tweet Projects**:

* <https://github.com/ammarhameed/NLP-Disaster-Tweets-Kaggle-Mini-Project>
* This repository includes detailed notebooks that walk-through preprocessing, feature extraction, and model building, including experiments with deep learning models like BERT.
* <https://github.com/sdurancmu/disaster_tweets>
* This project demonstrates multiple techniques, including LSTM and word embeddings, to classify tweets and tackle NLP challenges with ambiguous language.

3. **Tutorial on NLP Techniques for Disaster Detection**:

* Disaster Tweets Classification using Machine Learning
* [https://www.analyticsvidhya.com](https://www.analyticsvidhya.com/)
* This guide on Analytics Vidhya provides step-by-step instructions for building ML pipelines to classify tweets, including methods for preprocessing and handling imbalanced data.

4. **Literature and Research Papers on Disaster Response using NLP**:

* Improving Disaster Detection from social media Using NLP
* [https://dl.acm.org](https://dl.acm.org/)
* This research paper from ACM explores how NLP can enhance disaster response systems by identifying relevant social media content in real-time.