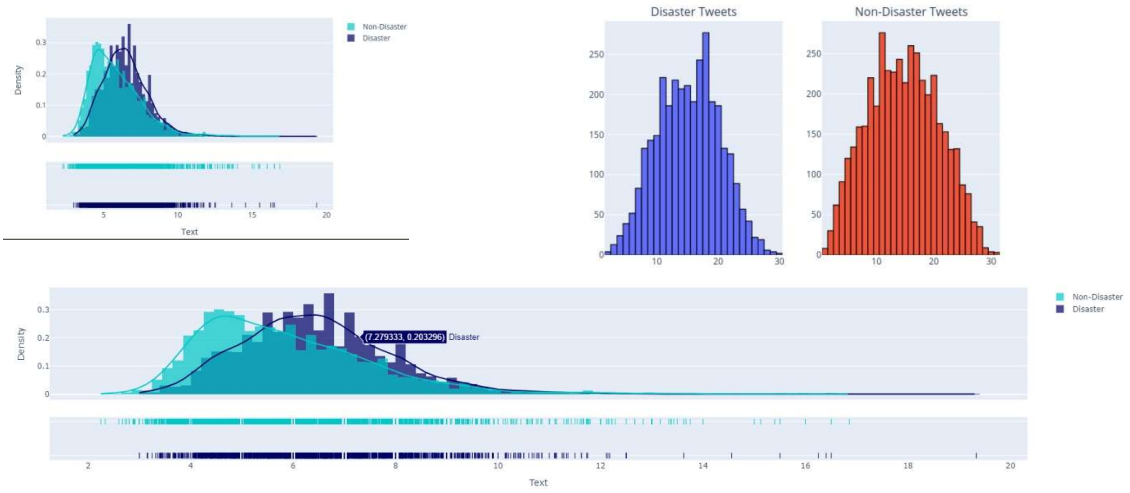
MODEL 1(Neural Network for Classification of Disastrous News):

DATA COLLECTION:

The dataset utilized for this project was sourced from a Kaggle dataset containing Twitter headlines classified as either *disastrous* or *non-disastrous*. To enhance the dataset, additional positive samples (disastrous headlines) were incorporated from **EM-DAT – The International Disaster Database**. Furthermore, news headlines were scraped from Twitter feeds of various Indian English news portals, such as **NDTV** and **India Today**. After aggregating and cleaning the data, the final dataset comprised a total of **13,952 records**.

EDA:

During the exploratory data analysis (EDA), it was observed that the dataset was balanced in terms of the classification labels. Further analysis was conducted to evaluate the **average word length** and the **word density** of the dataset, and the findings are summarized below:



DATA PREPROCESSING:

To prepare the dataset for training the neural network model, several data preprocessing steps were carried out to ensure its suitability and quality. Initially, duplicate entries were removed to avoid bias, and the textual data underwent thorough cleaning. This included converting all text to lowercase for uniformity, removing special characters, punctuation, and numbers, and eliminating stop words such as "the," "and," and "is" to retain only meaningful words. Additionally, stemming or lemmatization was applied to reduce words to their root forms.

The target variable, which categorized news headlines as either *disastrous* or *non-disastrous*, was label-encoded into numerical values, with 0 representing non-disastrous and 1 representing disastrous. To convert the cleaned textual data into numerical features, the TF-IDF (Term Frequency-Inverse Document Frequency) method was employed. Key parameters like **max_features** and **ngram_range** were fine-tuned to capture relevant patterns in the data without overfitting.

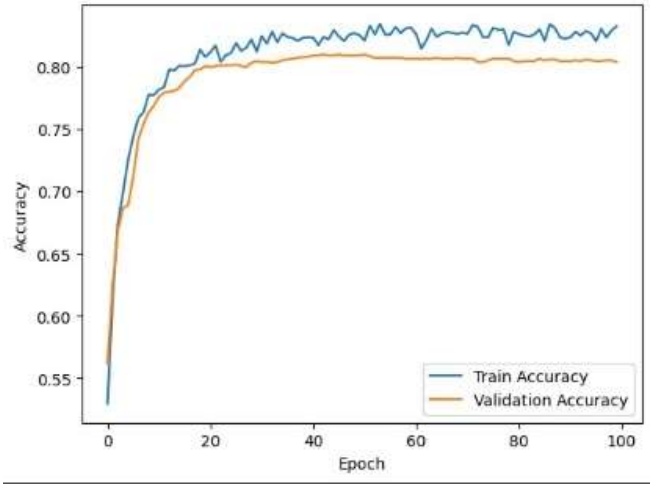
The dataset was then split into training, validation, and test sets using an 80:10:10 ratio to ensure effective evaluation of the model's performance. Feature scaling was applied to standardize the TF-IDF vectors, making them compatible with the neural network's input requirements. Finally, the dataset's balance across classes was verified, confirming that no additional oversampling or under sampling was necessary. These preprocessing steps ensured the dataset was well-prepared for neural network training.

Model Training:

The model architecture is designed with the following specifications:

- **Conv1D (1-dimensional Convolutional Layers):** Used to capture local patterns in the sequential data, such as n-grams or important phrases in text data. The convolutional layer is crucial for feature extraction from the input sequence.
- **Hidden Layers:** The model includes 2 hidden layers to enable learning of complex representations of the input data.
- **Optimizer:** The **Adam optimizer** is employed for efficient training. Adam is a popular choice due to its adaptive learning rate, which helps in faster convergence and better performance in a variety of tasks.
- **Dropout:** A dropout rate of **0.6** is applied to prevent overfitting. This means 60% of the neurons are randomly dropped during training to promote generalization and avoid the model memorizing the training data.
- **Dense Layer:** The final layer is a **dense layer** with **1 neuron** and a **sigmoid activation function**, suitable for binary classification tasks (disastrous or not). The sigmoid function outputs a value between 0 and 1, representing the probability of a given input being classified as disastrous.
- **Epochs:** The model is trained for **100 epochs**, meaning the entire training dataset is passed through the network 100 times, allowing the model to learn the patterns in the data effectively.

The model achieved a **training accuracy** of **84.34%** and demonstrated a **validation accuracy** of **79.93%**, indicating strong performance in correctly identifying disaster categories. These results suggest that the model effectively learned the patterns from the training data while maintaining a good generalization ability on unseen validation data. The slight drop in validation accuracy compared to training accuracy is typical, indicating that the model is not overfitting and can generalize well to new data.



Model 2: (classification of type of disaster):

Data collection:

The dataset utilized is obtained from EM-DAT – The International Disaster Database, which contains comprehensive global disaster data spanning from 1900 to 2022. Out of the 24,000 records available, a subset of 15,000 records was selected for analysis. After a rigorous data cleaning process, which included the removal of duplicates, the final dataset comprised 14339 records.

EDA:

During the exploratory data analysis (EDA), we observed that the dataset primarily focused on four disaster types: floods, earthquakes, cyclone, forest fires and accident. The distribution of these categories was found to be approximately 34.6%, 28.6%, 17.6%, 12.4% and 6.8%, respectively. As the dataset was nearly balanced, there was no requirement for normalization.

Data preprocessing:

During data preprocessing, label encoding was applied to categorize the target variable for this multi-class classification problem. The assigned labels were as follows:

- **1** for Earthquake
- **2** for Flood
- **3** for Cyclone
- **4** for Forest Fire
- **5** for accident

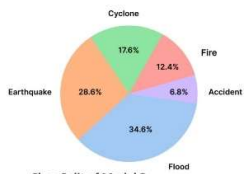
Additionally, the textual messages were vectorized using the **TF-IDF (Term Frequency-Inverse Document Frequency)** method to convert them into numerical representations suitable for model training

The dataset was then split into training, validation, and test sets using an 80:10:10 ratio with 42% randomness to ensure effective evaluation of the model's performance

Model Selection and training:

For this multi-class classification problem, the **Multinomial Naïve Bayes classifier** was selected. This algorithm classifies text into disaster types based on posterior probabilities, leveraging **Bayes' Theorem**. The model achieved an accuracy of **89.95%** and demonstrated a precision of **92.33%**, indicating strong performance in correctly identifying disaster categories.

Graphical Stats:



Class Split of Model 2