SOS SYNC

Anupama Kaushik, PhD
Maharaja Surajmal Institute of Technology
Dept. of IT, New Delhi, India

Shruti Sagar
Maharaja Surajmal Institute of Technology
Dept. of IT, New Delhi, India

Sameep Punjani
Maharaja Surajmal Institute of Technology
Dept. of IT, New Delhi, India

Priyanshu Mahendra
Maharaja Surajmal Institute of Technology
Dept. of IT, New Delhi, India

ABSTRACT
In times of distress situation, people find themselves in critical situations and are in a need of assistance due to various incidents like natural disasters, medical emergencies, or other life threatening events. Many people are often unable to get aid at the right time. So, in this research paper we are exploring the potential of harnessing machine learning and natural learning process (NLP) technologies to create a reliable system for classifying and detecting distress call using multiple languages. We have described an approach for automatically identifying the messages.

The paper provides a comprehensive overview of the entire process involved in developing an efficient distress call detection system using machine learning. It addresses various aspects, including the challenges associated with multi-lingual NLP, methods for identifying urgency in text, data preprocessing techniques to improve accuracy, and the evaluation of performance results. Additionally, the paper delves into the four key steps of the machine learning pipeline: text vectorization, Tf-Idf normalization, model training, and hyperparameter tuning. By combining NLP and machine learning methodologies, this research aims to establish an effective and precise system for recognizing urgency in multi-lingual texts.

Keywords
Multilingual, Classification Platform, Natural Language Processing (NLP), Machine Learning, Features, Pre-Processing Data, Benchmark Performance Results, ETL Pipeline, Text Vectorization, Term Frequency-Inverse Document Frequency (Tf-Idf) Normalization, XGBClassifier Model Training, Hyper-Parameter Tuning.

1. INTRODUCTION
Detecting distress calls in multiple languages using Natural Language Processing (NLP) involves a combination of language understanding, feature extraction, and classification techniques. In the domain of Multilingual Distress Call Classification and Detection using Natural Language Processing (NLP), the pivotal role of NLP lies in its ability to decode meaning from distress messages in non-native languages. However problem such as accurately and quickly gauging the nature of "urgency" in an incoming text but it was solved with the help of machine learning technologies and advances in NLP and now it is possible to utilize these technologies in order to develop an efficient multi-lingual distress call classification and detection platform. NLP's proficiency in handling linguistic variations, dialects, and contextual intricacies ensures the accurate interpretation of distress messages, ultimately contributing to the creation of a robust and responsive multilingual distress call classification and detection system.The research paper endeavors to provide a comprehensive exploration of the utilization of machine learning (ML) and natural language processing (NLP) for the development of an efficient urgency detection platform, with a specific focus on multi-lingual inputs.

This research paper aims to contribute insights and solutions at the intersection of NLP, ML, and multi-lingual urgency detection, fostering advancements in timely and effective response systems and will cover the topics such as the problems and complexities of multi-lingual NLP and machine learning, features that can be used to identify the nature of urgency, Furthermore the paper will explore the importance of balancing the dataset to address potential biases. Leveraging approaches from both natural language processing (NLP) and machine learning forms a framework for discerning and categorizing the urgency level of incoming messages. This synergy empowers the system to not only identify the severity of a distress call but also facilitates more efficient and prompt responses to urgent situations. The forthcoming discussion within this research paper aims to pinpoint viable solutions, outlining the potential success achievable through the application of cutting-edge technologies in the development of a multi-lingual distress call classification and detection platform.

2. RELATED WORKS
In recent years, there has been a growing emphasis on researching multilingual urgency prediction, driven by the increasing demand for robust natural language processing (NLP) systems adept at accurately identifying threats in languages other than English. Various approaches have been proposed for detecting threats in different natural languages. Notably, rule-based approaches have been employed to identify threatening content in English, Spanish, and other languages. Aljabri et al. (2022) [1] introduced an integrated system designed for multi-language key term extraction and threat detection. This system identifies keywords and analyzes their contexts to categorise tweets into benign, suspicious, and malicious classifications. However, these approaches have some limitations in accuracy and performance, primarily stemming from the challenges connected with learning language-specific word implanting and deploying NLP models capable of making understandable to the general public with all the languages.

Scientists have also focused on employing machine learning models to more effectively grasp the subtleties present in diverse languages and dialects. For example, Kahlen et al. (2021) [2] and their colleagues introduced a transformer-based model. This model utilize methods specific to each language to

supplement data, aiming to enhance it's overall performance.Aydoğan & Karci (2020) [3] improved the classification accuracy of Turkish tweets by incorporating pre-trained contextualized word embeddings along with recurrent neural networks. Additionally, Zhang et al. (2021) [4] applied transfer learning through DenseNet, achieving an 82.4% accuracy for Chinese tweets. The combined research efforts in multilingual urgency detection and classification have demonstrated significant advancements in recent years. Looking ahead, there is an optimistic anticipation that the performance of these models across different languages will further improve with the ongoing development of deep learning techniques.

Some other works are mentioned below as well:

Detecting urgency in brief crisis messages with limited management and transfer learning is the focus of a study by Kejriwal and Zhou (2020) [5]. In recent times, the rise in humanitarian crises, driven by factors like climate change and sociopolitical issues such as refugee crises, has underscored the need for efficient resource mobilization during natural disasters. Utilizing technology to semi-automatically flag tweets and short messages as urgent can aid in directing resources like food and water effectively.the problem is difficult not only due to the lack of data in the immediate aftermath of a disaster but also due to the varying characteristics of disasters in developing countries (making it difficult to train a single system) and the noise and peculiarities of social media.

The paper proposes a robust, low-supervision social media urgency system designed to adapt to various crises using both labeled and unlabeled data within an ensemble framework. Employing a straightforward and efficient transfer learning technique, the system can adjust to new crises even when an unlabeled background corpus is not yet available. Experimental results shows that these transfer learning and low-supervision methods overshadow viable baselines significantly across a diverse range of disaster datasets. Madichetty and M. (2020) [6] introduce a stacked convolutional neural network (CNN) designed for the detection of resource-related tweets during disasters, particularly focusing on the "need and availability of resources" (NAR) aspect. Social media platforms, like Twitter, play a pivotal role in providing real-time information during events such as natural disasters and political occurrences. Identifying NAR tweets is crucial as they encompass diverse information, including infrastructure damage, resource availability, opinions, and sympathies related to disaster events. However, existing methods for NAR tweet detection are not well-focused and exhibit poor performance. Detecting NAR tweets during a disaster necessitates, therefore, the use of steady methodologies. Existing works are not well-focused on NAR tweet detection and perform poorly. Consequently, the focus of this paper is the detection of NAR tweets during a disaster.

To fill the existing gap, the suggested method combines conventional feature-based classifiers with convolutional neural networks (CNNs), acknowledging the limited availability of labeled datasets specifically tailored for "need and availability of resources" (NAR) tweets during crises. This innovative approach involves integrating meaningful features such as "help," "need," "food," "earthquake," etc., utilized by both the classifier and CNN. The acquired features, represented by the output from the CNN and the classifier incorporating informative features, are amalgamated into a second classifier known as the meta-classifier, designed for NAR tweet detection. Various classifiers, including SVM, KNN, decision trees, and Naive Bayes, are applied in this model.

Experiments conducted on earthquake datasets from Nepal and Italy in 2015 and 2016 demonstrate that the proposed model, employing KNN as the base classifier and SVM as the meta-classifier in conjunction with CNN, surpasses the performance of other algorithms. The results reveal enhanced precision compared to baseline methods, underscoring the efficacy of the stacked CNN-based approach in detecting resource-related tweets during disasters.

3. DATA FLOW DIAGRAM

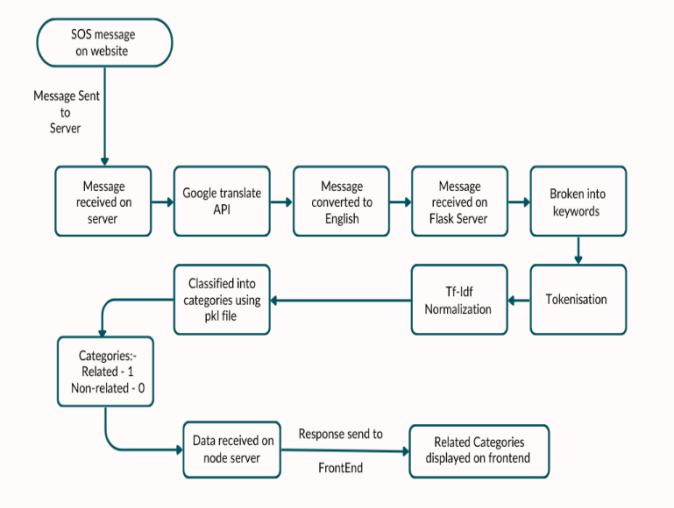

Fig.1: Data flow diagram

4. METHODOLOGY

4.1 Translation

After considering various multilingual models [7], we turned to using the Google Translate API. Integrating the Google Translate API (v2) [8] into the backend of the system was an exciting addition. His incredible accuracy in speech recognition and instant translation was incredibly valuable. First, the API uses advanced models to quickly and accurately identify the input language. Next, translate the input into English and prepare it for the NLP model. The NLP model then analyzes the message by breaking it down into individual words connected by "+" to determine the relevant category of the message and send the appropriate response. Finally, the front end of the system receives the answer. This fast and accurate data inflow allows NLP models to understand multiple languages, making it easier to implement projects in these languages.

4.2 ETL Pipeline

The ETL (Extract, Transform, Load) pipeline is a systematic process that transforms raw data into a format suitable for analysis and practical use. It begins with extraction, where data is collected from a source, often a CSV file. The transformation phase is crucial, especially when dealing with a category column containing a mix of strings and numbers. Transformation consists of tasks such as cleaning, validation, filtering, manipulation, and integration of data. An example is normalizing the mixed string- number category into a consistent format, like integers.

The final ETL step involves loading the transformed data into a database. A backend database engine organizes the data systematically, making it easy to query. A well-implemented

ETL process ensures a smooth loading into a database or other storage formats, like NoSQL document stores, with very less or no errors. Once data is extracted and transformed, it is ready for loading into a database or data collection part, marking the end of this pipeline. This step consists of validation, error checking, and correction to maintain data integrity.

The loading phase set the seal on uniformity in the target data model, integrating essential improvements in performance and data quality. This target data model describes the data structure, outlining fields, data types, and specific rules. Once loaded into the database, the data becomes accessible for different purposes, including data analysis, reporting, machine learning, and other applications. ETL pipelines can be designed to operate at predetermined intervals, guaranteeing a consistent flow of data updates. This methodology is specifically advantageous for applications that demand real-time data, catering to the need for up-to-the-minute information.

In essence, ETL pipelines are usefull and important component of modern data architecture, allowing organizations to leverage their data assets effectively for strategic decision-making and business insights.

5. MACHINE LEARNING

There are mainly four important steps of the machine learning pipeline, which are text vectorization, term frequency-inverse document frequency (Tf-Idf) normalization, classifier model training, and hyper-parameter tuning.

5.1 Text Vectorization

Text vectorization is the process of converting raw text into a matrix that counts tokens (individual words). This phase is important for building models that can correctly identify patterns in unstructured text data. There are many different ways to vectorize text, including bag-of-words, N-grams, and word embeddings. Bag-of-Words is a widely used technique for text vectorization due to its lucidity and productiveness in transforming text to numerical form. This method includes building a vocabulary from text data and transforming each document into a vector form. Each element represents a word in the vocabulary.

The use of of Count Vectorizer, also known as Bag of Terms/Token, has demonstrated to be a robust method in recent times, distinctly in the area of Natural Language Processing (NLP). This approach has played a crucial role in the development of an Urgency Detection Platform. Count Vectorizer serves as a technique to alter textual data into a numerical format appropriate for machine learning algorithms.In our program Count Vectorizer interprets text by constructing a matrix of count values. Each column in this matrix represents a different terms, while each row corresponds to a document. The tokenized terms are subjected to various processing steps, such as the removal of words based on frequency, parts of speech, stop words, etc. Thereafter, the resulting document term matrix is provided to a machine learning algorithm.

Utilizing Count Vectorizer is crucial for constructing an urgency detection platform through Natural Language Processing (NLP). The transformation of text into numerical data allows the algorithm to make well-informed decisions by examining the document’s context. This approach makes the recognition of connections between a document’s urgency and its specific terms easy, ultimately strengthens the system’s accuracy in evaluating and improving urgency detection.

5.2 Term Frequency-Inverse Document Frequency (Tf-Idf) Normalization

The normalization process in term frequency-inverse document frequency (Tf-Idf) includes transforming a matrix of token counts into a normalized rendering based on Tf-Idf values. This normalization is the key because it mitigates the impact of commonly occurring words across documents, at the same time amplifying the noteworthiness of less appearing words. By this way, the accuracy of machine learning models can be improved by prioritizing words that are more likely to be suggestive of specific classes or topics.

5.3 Classifier Model Training

Training a classifier involves fine-tuning its parameters using an algorithm to pare down a specified loss function, an important step for attaining precision in models, improving overall performance, and reducing generalization errors.The trained model is then capable of making informed decisions based on new input data.

The MultiOutputClassifier(XGBClassifier()) proves to be a robust tool for developing multilingual distress call classifiers with the help of natural language processing. This approach make sure that the development of a self-sufficient classifier that quickly processes and learns from natural language data without the requirement of manual intervention. The main precedences of MultiOutputClassifier(XGBClassifier()) surrounds the creation of a classifier with less features and in not so much time, therefore uplifting the efficiency. It allows concurrent learning from multiple languages, speeding up the overall learning process. Besides, XGBClassifier demonstrates scalability, enabling straightforward parameter adjustments as the dataset expands.

This method harnesses potent gradient boosting algorithms to uplift weaker learners in developing classification models, heading to heightened accuracy and the creation of a more versatile model. The classifier adeptly manages diverse data types, automatically adapting to different data sizes. When compared to different machine learning algorithms, MultiOutputClassifier(XGBClassifier()) distinguishes itself with its efficiency, suitability for both batch and streaming environments, and flexibility over various data sources and formats, involving unstructured textual data, images, videos, and audio. Its productiveness in handling extensive datasets positions it as the right choice for multilingual distress call classifiers.

5.4 Hyper-Parameter Tuning

Hyperparameter tuning is the procedure of optimizing a machine learning model's hyperparameters to increase its performance. This is a critical and very important step in making sure that the development of accurate models because the selection of relevent hyperparameters significantly impacts a machine learning model's efficiency effectiveness. Moreover, hyperparameter tuning be in the service to mitigate the risk of overfitting, gives autherization to the model to be fine-tuned to the available data. After training the model in the machine learning pipeline, there is an supplementary step to impove performance by choosing the optimal hyperparameters for the model.

The process starts with the decision on which parameters to tune and the values of these parameters should encompass. To find out the range of parameter values for testing, an initial search is performed by manually taking a look at various parameter combinations and asses the worth of their impact on

the model's performance. Intuition can also play a role in this decision-making process. For example, when dealing with the n-gram range parameter, one might start with (1,1) and progress towards (1, 2). Having recourse to the specified parameters in the provided grid, a parameter dictionary is constructed.

Grid search cross-validation then commences, attempting all combinations of designated parameters within the defined range and evaluating the model's score based on performance metrics. The output is the finest set of parameters that increases the model's score.While hyperparameter tuning significantly improves model performance, it's critical to note that the optimal parameters obtained through grid search cross-validation are costumized to a specific dataset. If the data is significantly different from the training dataset, a different set of hyperparameters might be needed for optimal performance.

Although Scikit-Learn Grid Search CV exhaustively explores the parameter grid, it's necessary to recognize that the finest hyperparameters for one dataset may not be the best for another. In consequence with tuning hyperparameters for distinct datasets is crucial to achieve optimum model performance.Embracing a modular approach in the machine learning pipeline permits for easy debugging and adjustments at each step, making ML pipelines an outstanding choice for automating the predictive model-building process.

6. MODEL EVALUATION

Evaluating a model is critical for assessing its performance,significantly in classification tasks. A widely practiced approach includes dividing the data into two sets: one for training the model and the other for testing its performance. The evaluation process includes metrics such as accuracy, precision, recall, and F1 scores.Accuracy provides an overall measure of the model's performance, indicating how accurately it predicts classes. It is computed by dividing the number of correctly predicted classes by the total number of classes being tested. Precision assesses the accuracy of positive predictions by dividing the number of correctly predicted classes by the total number of predictions. Recall, on the other hand, measures the accuracy of positive predictions by dividing the number of correctly predicted classes by the total number of actual classes. The F1 score, a weighted harmonic mean of precision and recall, serves as a comprehensive metric for comparing different models.Utilizing these evaluation scores enables model comparison, offering insights into how our model performs in relation to others. Moreover, it aids in identifying areas where the model may require improvement, facilitating necessary adjustments. This iterative process is fundamental to the development cycle of any machine learning system, ensuring the correctness and accuracy of results obtained. Model evaluation also has a pivotal role in authenticating the model's fitness for deployment, providing assessments of accuracy, precision, and F1 scores.In situations where the central objective is to identify or prioritize individuals genuinely in need of aid or emergency assistance, particular attention should be given to the recall metric. This metric becomes particularly crucial in scenarios where the emphasis is on minimizing false negatives and accurately capturing instances requiring intervention.

The primary rationale behind prioritizing recall over precision in model evaluation lies in the understanding that a false positive (erroneously identifying someone who doesn't need aid) is less impactful than a false negative (overlooking a person genuinely in need). Recall, also known as "sensitivity," represents the likelihood of correctly identifying individuals in need of help. It quantifies the proportion of people requiring assistance that the system successfully identifies.

Calculation of recall involves dividing the number of true positives by the total number of positives present in the ground truth. Recall favors models that generate substantial true positives, even at the expense of lower precision. Precision, in contrast, emphasizes instances where the model accurately identifies those in need but may have a lower fraction of true positives. It quantifies the percentage of valid predictions out of all predictions available.While both precision and recall considerations are crucial, the F1 score is often deemed more informative. This metric combines both recall and precision measurements into a singular value. The F1 score, interpreted as a measure of a classifier's accuracy, represents a weighted average of precision and recall. It serves as a comprehensive metric offering a balanced assessment of the model's performance.A heightened F1 score indicates that the classifier adeptly labels individuals in need of assistance, signifying accurate identifications. The choice of which metric to prioritize hinges on the contextual nuances of the data. For instance, in scenarios where false negatives could entail significant costs, prioritizing recall becomes essential to minimize overlooking individuals requiring attention. Conversely, if false positives carry higher costs, precision takes precedence to ensure efficient resource utilization.There is no rigid rule governing the selection of an evaluation metric; rather, it should be decided on a case-by-case basis. The ultimate objective is to strike a balance between precision and recall. Achieving this equilibrium involves fine-tuning the parameters of the data model and assessing the model's performance accordingly. By attaining the right balance between precision and recall, we can ensure that those in need receive adequate attention, and resources are utilized efficiently, tailored to the specific context.In the evaluation of emergency messages, there is no universal metric that fits all scenarios, as considerations may vary between prioritizing resource conservation or avoiding the oversight of critical messages. The determination of which evaluation metric to prioritize must be a meticulous process, taking into account the unique characteristics of each context. A balanced approach between precision and recall typically proves to be the most desirable, achievable through the careful tuning and adjustment of the model's parameters.

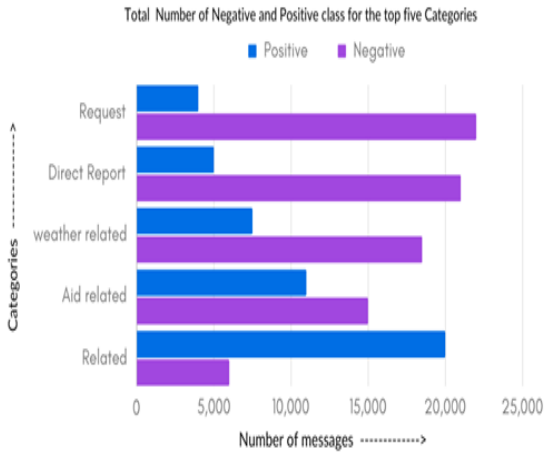

Fig.2: Top 5 most balanced categories

The bar plot depicted above illustrates the distribution of labels (0 and 1) for the top five most balanced categories within the original dataset. However, a noticeable discrepancy exists in the majority of other categories, where a significant imbalance is observed between positive (1) and negative (0) labels. Notably, categories such as 'fire,' 'missing individuals,' 'offer,' 'hospital,' 'electricity,' and 'transportation' exhibit minimal entries. In contrast, categories like 'violence,' 'weather,' and 'communication' are prevalent in a substantial number of messages.In essence, the original dataset exhibits a considerable imbalance, posing challenges for training a machine learning model. The prevalence of zero labels in the majority of categories raises concerns about potential bias in the model, leading it to favor predicting a zero label even when the underlying data point should be assigned a one label.

Even with the inclusion of stratified sampling in the grid search cross-validation, designed to maintain class weights, the model encounters substantial challenges, resulting in notably high or low precisions and/or recalls for specific categories, as observed above. While implementing a stratified sampling system is a plausible attempt to balance data distribution, its effectiveness can vary depending on the dataset's structure, including the distribution of data among classes, such as the quantity of data for each class.Addressing this issue necessitates a thorough exploration of the underlying data and its distribution to gain a more comprehensive understanding. An approach commonly employed to balance data is replication, which entails duplicating examples to augment the data within minority classes. In this particular scenario, replication could be a viable strategy to enhance the ratio of examples in categories exhibiting low precision and/or recall.Conversely, another method worth exploring to equalize data distribution involves the selective removal of certain categories, resulting in a reduction of the training set. Each method poses its own set of considerations, and their effectiveness depends on the unique characteristics of the dataset. Ultimately, a thoughtful examination of data distribution and the application of suitable methods are essential to address imbalances and enhance the model's performance.

Applying the same principle, one can manually manipulate the distribution of examples pertaining to low precision and/or recall categories. Through this adjustment, the number of examples in each class can be balanced optimally, mitigating the likelihood of errors and ensuring a sample dataset of appropriate size. While our discussion thus far has revolved around dataset-related methods, the choice of the model also significantly influences predictive power and, ultimately, metric outcomes.In situations characterized by high variance or datasets featuring a diverse array of examples, deep learning models emerge as a more favorable solution than models with fewer adjustable parameters. The inherent capacity of deep learning models to learn and model high-dimensional, multifaceted data makes them particularly suited for scenarios with highly dimensional and varied patterns. Notably, deep learning models are recommended for their ability to offer valuable insights.Unlike other models like decision trees or support vector machines, deep learning models involve fewer assumptions, reducing the likelihood of bias in results. Furthermore, several essential parameters necessitate tuning to achieve optimal model performance, accounting for considerations such as underfitting and overfitting. Enhanced performance can be achieved through the tuning of hyperparameters, a process that involves identifying an optimal set of parameters to align the model with the dataset's specific characteristics.

Grid search serves as a widely used technique in hyperparameter tuning, defining specified is ranged for each parameter to permit the algorithm to select the combination leading to optimal accuracy. This method is significantly beneficial for optimizing model simplicity efficiency and performance, especially in high-dimensional datasets, as it iteratively prospects various parameter combinations. Grid search improves model accuracy by employing cross-validation across the attained combinations, providing a comprehensive measure of the model's true accuracy. This process cushions the risk of overfitting and facilitates the identification of the best parameter combination through comparative metrics.The integration of both grid search and stratified sampling, which ensures the maintenance or increase of class weights, is critical for accurate results. In scenarios where minority classes have limited examples, stratified sampling, along with replication or manual distribution adjustment, is important to achieve balanced precision and recall levels for each category. Deep learning models are recommended in such cases, given their robustness and potential for valuable insights. This approach is expected to improve overall model performance and provide a more precise definition of the model's true accuracy.In conclusion, when the need arises to equalize data distribution, methods like replication, manual adjustment, or the use of deep learning models should be contemplate to obtain accurate results. Additionally, the combination of grid search and stratified sampling make sure enhanced model performance, especially in high-dimensional datasets, by defining the best parameter combination and offering a comprehensive metric for optimal solutions. Therefore, investing time in assessing data distribution quality, along with employing suitable methods adapted to data balance, is important to ensure the model achieves optimal results.

7. PRESERVING THE MODEL

Preserving the trained model for future use, whether for predictions or transferring it to different environments, is a crucial step. This task can be effortlessly accomplished by leveraging the Python Pickle module to dump and load the model.Saving the model involves a straightforward process. By importing the pickle module, you can utilize the 'dump' function, which takes two arguments: the model object to be saved and the location where the model should be stored. Keeping track of the storage directory is essential for future reloading.Refer the code snippets for the better understanding-

```
import pickle pickle.dump(model, open(model_filepath, "wb"))
```

Reloading the saved model is equally uncomplicated. The 'load' function from Pickle can be employed, specifying the exact directory of the model file. This step needs to be executed whenever the model is required for use:

```
model = pickle.load(open(model_filepath, 'rb'))
```

It's important to note that the pickle package is a generic serialization library. To ensure proper reloading and functionality, any associated dependencies, libraries, and packages must be imported correctly. Maintaining consistency in the Python version used for saving and loading the model is crucial. Although Pickle is primarily designed for Python, it can also seamlessly work with other languages like Java, C, and C++, allowing models to be loaded across multiple

platforms.For added security, models can be encrypted and compressed using gzip before storing. This can be achieved with a few additional lines of code:

import gzip import pickle pickle.dump(model, gzip.open(model_filepath_gz, "wb"))

To conserve space, Pickle can be utilized to delete the trained model after it has been saved and reloaded. This can help manage storage and ensure that only important models are retained in the system. The following code can be employed to delete a reloaded model:del model

In summary, employing Pickle for model storage and retrieval provides flexibility and facilitates seamless sharing of models and ideas across different platforms. Add on features such as encryption, compression, and model deletion enhance the overall utility of this approach.

8. RESULT AND DISCUSSION

India, with its vast population exceeding 1.3 billion people and diverse geography, grapples with a range of disasters, from uttermost weather events to earthquakes and floods. These natural calamities, like tsunamis, cyclones, and droughts, impact millions annually, prompting the use of social media platforms for helping it. Researchers have responded by providing a solution that uses social media during crises for help, employing a machine learning model trained on an Indian state-level crisis dataset to classify disaster-related tweets. The model finds out disaster mentions, assigning sentiment scores based on labeled training data. Regardless of the challenges of data labeling, the solution achieved an fantastic average f1-score of 68%, applicable across state-level crises and many different languages like Hindi, Tamil,punjabi and Marathi.The Noteworthiness of this solution lies in its potential applications, opening avenues for leveraging information and communication technology in disaster preparedness, response, and recovery. Operating at a state level is critical for assisting vulnerable populations with limited access to traditional responders. Moreover, it establishes an early warning system for citizens, informing them of potential disasters in advance. The solution's utility extends to aiding organizations and NGOs in collecting data on crises, contributing to enhanced distribution, resource management, and recovery efforts. This data entitles the government to provide resources efficiently and provide direct support to communities.In essence, the development of this model, adept at distinguishing disaster-related tweets, represents a remarkable step toward efficiently using social media for disaster risk reduction, response, and recovery. With further refinement and other data, there is expectation that the solution's performance can see substantial enhancements, potentially saving numerous lives in the future.

Table 1- Model result

Category	Precision	Recall	F1-Score	Support
related	0.86	0.95	0.9	3957
request	0.78	0.62	0.69	873
offer	0.51	0.04	0.06	39
aid_related	0.78	0.64	0.70	2163
medical_help	0.64	0.30	0.41	393
search_and_rescue	0.55	0.19	0.28	141
security	0.36	0.07	0.12	98
military	0.62	0.28	0.38	189
child_alone	0	0	0	0
water	0.72	0.67	0.7	334
food	0.81	0.78	0.8	578
shelter	0.79	0.65	0.71	481
clothing	0.68	0.58	0.63	69
money	0.4	0.24	0.3	101
missing_people	0.66	0.25	0.36	56
refugees	0.65	0.31	0.42	181
death	0.70	0.53	0.61	237
other_aid	0.55	0.18	0.27	653
infrastructure_related	0.43	0.1	0.15	327
transport	0.63	0.2	0.31	237
building	0.73	0.41	0.53	273
electricity	0.68	0.36	0.47	113
tools	0	0	0	39
hospitals	0.21	0.05	0.07	56
shops	0	0	0	19
aid_centers	0.3	0.08	0.12	59
other_infrastructure	0.31	0.06	0.09	219
weather_related	0.86	0.70	0.77	1429
floods	0.82	0.55	0.65	406
storm	0.76	0.67	0.71	469
fire	0.70	0.21	0.32	57
earthquake	0.88	0.81	0.85	482
cold	0.60	0.40	0.48	94
other_weather	0.41	0.15	0.22	274
direct_report	0.73	0.51	0.6	980
micro avg	0.79	0.62	0.69	1630
macro avg	0.57	0.36	0.42	1630
weighted avg	0.75	0.62	0.66	1630
samples avg	0.63	0.52	0.53	1630

9. COMPARISON

In the realm of distress call classifiers, numerous strategies can be employed [9], with two prominent machine learning algorithms being XGBClassifier [10] and Random Forest Classifier (RFC) [11]. Our agenda is to compare these algorithms in the classification of distress calls within a multilingual distress call classifier using natural language processing.For conducting this comparison, both XGBClassifier and RFC were applied to the same multilingual distress call classifier, and their respective performances were

evaluated. The assessment was carried out using an identical dataset of distress calls in multiple languages.The results showed that XGBClassifier surpasses RFC in the classification of distress calls. The superiority of XGBClassifier is attributed to its enhanced capacity for generalization.

This stems from its ability to discern and prioritize features that are more relevant to the classification task while disregarding non-relevant ones, in contrast to RFC, which considers all provided features. XGBClassifier's efficiency is further underscored by its ability to identify patterns with fewer features, leading to quicker processing of classification data. This efficiency proves advantageous for supervised learning, as the algorithm does not need to consider every feature for classification.

In conclusion, when comparing XGBClassifier to Random Forest Classifier, the former emerges as a more effective distress call classifier for a multilingual distress call classifier employing natural language processing. This superiority is attributed to XGBClassifier's capability to recognize more significant features for classification and its adeptness at ignoring non-relevant features, making it more efficient and suitable for supervised learning.

Table 2 – Random Forest Classifier Results

Category	Precision	Recall	F1-Score
micro avg	0.83	0.53	0.64
macro avg	0.61	0.19	0.25
weighted avg	0.79	0.53	0.57
samples avg	0.69	0.49	0.59

Table 3 – XGB Classifier Results

Category	Precision	Recall	F1-Score
micro avg	0.77	0.60	0.67
macro avg	0.55	0.34	0.40
weighted avg	0.73	0.60	0.64
samples avg	0.63	0.49	0.52

10. CONCLUSION AND FUTURE PROSPECTS

The outlook for a Classification and detection of distress calls in multiple languages employing natural language processing appears promising. Governments and societies have expressed significant concerns about citizen safety, necessitating efficient methods for identifying distress calls. NLP presents an appealing solution to this challenge, given its ability to accurately process natural language queries and adapt to various languages. Implementing a multilingual distress call classifier with NLP holds potential for effectively identifying distress calls and ensuring appropriate responses.Various methods can be employed to implement this classifier. For instance, training the classifier using text classification algorithms like Logistic Regression and Support Vector Machines (SVMs) is one approach. Including unsupervised learning algorithms such as clustering and topic modeling can yield improved results. And also, leveraging convolutional neural networks (CNNs) can extract meaningful features from the data.To enhance the performance of Classification and detection of distress calls in multiple languages employing natural language processing further, exploring advanced algorithms is advisable. For example, incorporating LightGBM (Machado et al., 2019) [12] and tuning the model using Bayesian optimization [13] can significantly enhance performance. Combining deep learning, NLP, and machine learning algorithms can boost model accuracy. Cloud-based services offer another avenue for optimization, facilitating the storage of training data, processing call logs and text, model building, and deployment to multiple platforms such as smartphones and web applications. This enables the use of the multi language call classifier in real-time settings.In conclusion, the future of classification and detection of distress calls in multiple languages employing natural language processing holds promise. By implementing advanced algorithms and leveraging cloud-based services [14], model accuracy can be improved, and distribution can be streamlined. Furthermore, the integration of unsupervised learning algorithms has the potential to enhance model performance. NLP's role in distress call identification is likely to become a valuable asset for ensuring citizen safety in the future.

11. REFERENCES

[1] Kejriwal M, Zhou P. On detecting urgency in short crisis messages using minimal supervision and transfer learning. Soc Netw Anal Min. 2020;10(1):58. doi: 10.1007/s13278-020-00670-7. Epub 2020 Jul 8. PMID: 32834866; PMCID: PMC7341028.

[2] Alam F, Ofli F, Imran M (2018) Crisismmd: multimodal twitter datasets from natural disasters. In: Twelfth international AAAI conference on web and social media

[3] Anderson KM, Schram A, Alzabarah A, Palen L. Architectural implications of social media analytics in support of crisis informatics research. IEEE Data Eng Bull. 2013;36:13– 20. [Google Scholar]

[4] Arthur R, Boulton CA, Shotton H, Williams HT (2017) Social sensing of floods in the UK. arXiv preprint arXiv:1711.04695 [PMC free article] [PubMed]

[5] Avvenuti M, Cresci S, La Polla MN, Marchetti A, Tesconi M (2014) Earthquake emergency management by social sensing. In: Pervasive computing and communications workshops (PERCOM Workshops), 2014 IEEE international conference on, pp 587–592. IEEE

[6] Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot CrossLingual Transfer and Beyond. Trans. Assoc. Comput. Linguistics, 7:597–610.

[7] M. Aljabri et al., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," in IEEE Access, vol. 10, pp. 121395-121417, 2022, doi: 10.1109/ACCESS.2022.3222307.Burel G, Alani H (2018) Crisis event extraction service (crees)-automatic detection and classification of crisis-related content on social media

[8] Aydoğan, M., & Karci, A. (2020). Improving the accuracy using pre-trained word embeddings on deep neural

networks for Turkish text classification. Physica Astatistical Mechanics and Its Applications, 541, 123288.

[9] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res. 2011;12:2493– 2537. [Google Scholar].

[10] Crooks A, Croitoru A, Stefanidis A, Radzikowski J. # earthquake: Twitter as a distributed sensor system. Trans GIS. 2013;17(1):124–147. doi: 10.1111/j.1467-9671.2012.01359.x. [CrossRef] [Google Scholar]

[11] Song G, Huang D, Xiao Z. A Study of Multilingual Toxic Text Detection Approaches under Imbalanced Sample Distribution. Information. 2021; 12(5):205. https://doi.org/10.3390/info12050205

[12] Zhang, YD., Satapathy, S.C., Zhang, X. et al. COVID-19 Diagnosis via DenseNet and Optimization of Transfer Learning Setting. Cogn Comput (2021). https://doi.org/10.1007/s12559-020-09776-8

[13] Ali, Jehad & Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues(IJCSI).

[14] Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759

[15] Kulikov, V. & Yerkebulan, Gulnur. (2019). ABOUT USING GOOGLE CUSTOM SEARCH AND GOOGLE TRANSLATE API IN DETECTION OF CROSS-LANGUAGE PLAGIATE. Вестник Алматинского университета энергетики и связи. 109-116. 10.51775/1999-9801_2019_47_4_109.

[16] Verma S, Vieweg S, Corvey WJ, Palen L, Martin JH, Palmer M, Schram A, Anderson KM (2011) Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency. In: Fifth international AAAI conference on weblogs and social media

[17] Shang, Weiyi & Adams, Bram & Hassan, Ahmed E.. (2012). Using Pig as a data preparation language for large-scale mining software repositories studies: An experience report. Journal of Systems and Software - JSS. 85. 10.1016/j.jss.2011.07.034.

[18] Habibian, Amirhossein & Mensink, Thomas & Snoek, Cees. (2014). VideoStory: A New Multimedia Embedding for Few-Example Recognition and Translation of Events. MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia. 10.1145/2647868.26549

[19] Chen, T., & Guestrin, C. (2016, March 9). XGBoost: A Scalable Tree Boosting System. arXiv.org. Retrieved December 20, 2022, from https://arxiv.org/abs/1603.02754v3