Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

DOCUMENTATION REPORT

Digi-Bayani System:
Classifying Filipino & Taglish Social Media Posts for Urgent Disaster Response

Submitted By:
Digi-Bayani

Group Members:
Bulos, Mariabel M.
Ramos, Railey Mar M.
Rosario, Keanne Jericho M.
BSCS 3B

Collaborative Final Project
CSST101 – Machine Learning
CSST102 – Knowledge Representation and Reasoning

PROJECT OVERVIEW

The Philippines is highly vulnerable to climate hazards. During disasters, social media is flooded with real-time information in Filipino and Taglish (code-switching), which can overwhelm emergency responders. The Digi-Bayani System is an intelligent tool designed to prioritize urgent posts and save lives. It utilizes a Natural Language Processing (NLP) classifier to categorize social media posts into actionable labels.

OBJECTIVES

General Objective:

Digi-Bayani System's general objective is to provide an intelligent tool designed to prioritize urgent social media posts and save lives during disasters in the Philippines.

Specific Objectives:

1. Develop an Intelligent Classifier: Build a supervised Machine Learning model capable of understanding Filipino and Taglish disaster-related text.
2. Integrate Reasoning: Implement a Knowledge Representation and Reasoning (KRR) module to assign risk levels based on ML output and location data.
3. Real-Time Prioritization: Create a system that automatically triggers high-priority alerts for "Urgent Need" reports.
4. Web Accessibility: Provide an interactive web interface for real-time tweet classification.

SYSTEM ARCHITECTURE

Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

User Input → Machine Learning Model → KRR Rules → Final Risk Level

MACHINE LEARNING COMPONENT (CSST101)

Algorithm Used: Logistic Regression (with liblinear solver)

Dataset Size: Combined dataset of 11,212 records (6,806 from Typhoon Yolanda + 4,406 from 2012 Floods)

Model Accuracy: 77.80% (0.7780)

MACHINE LEARNING PIPELINE

Data Collection: Merging of labeled CSV datasets from CrisisLex.org (2013 Typhoon Yolanda and 2012 Philippine Floods)

Data Preprocessing: Implementation of a clean_tweet_text function using Regex to remove URLs, mentions, hashtags, and punctuation, followed by lowercasing and TF-IDF Vectorization.

Model Training: Training a Logistic Regression classifier on an 80/20 train-test split of the vectorized text data.

Model Evaluation: Assessment via Classification Report (Precision, Recall, F1-Score) and Confusion Matrix; achieved 98% recall for the Disaster-Related class.

Model Deployment: Serialization of the trained model, TF-IDF vectorizer, and label encoder into .pkl (pickle) files for use in Flask and Streamlit applications.

DATASET DESCRIPTION

Dataset Type: Multiclass Textual Data (Social Media Tweets in English, Filipino, and Taglish)

Number of Records: 11,212 Total Tweets

Target Variable: y_new (Categorized as: 0: Disaster-Related, 1: Not Disaster-Related, 2: Uncertain / Needs Review)

KNOWLEDGE REPRESENTATION & REASONING (CSST102)

Rule 1: IF ML Prediction is "Disaster-Related" **THEN** Assign Severity 1 (High Priority)

Rule 2: IF Tweet contains "Rescue" OR "Help" OR "S.O.S" **THEN** Trigger Immediate Emergency Alert Rule

Rule 3: IF ML Prediction is "Uncertain / Needs Review" **THEN** Route to Manual Moderator Dashboard Rule

Rule 4: IF ML Prediction is "Not Disaster-Related" **THEN** Filter from Emergency Response Feed Rule

Rule 5: IF Location Data is present AND Class is "Disaster-Related" **THEN** Map coordinates for Responder Dispatch

HYBRID DECISION LOGIC

The system merges the statistical speed of Machine Learning with the structured logic of KRR. While the Logistic Regression model provides the initial classification of the tweet's intent, the KRR module acts as a "Reasoning Layer" that determines the urgency level and the specific response protocol (e.g., Severity 1 vs. Manual Review) based on detected keywords and metadata.

SYSTEM FEATURES

☒ **Real-Time Classification**

☒ **Rule-based recommendations**

☒ **Web interface / API**

☒ **Google Colab deployment**

SYSTEM DESIGN

Website Architecture

The Digi-Bayani platform follows a Client-Server Architecture designed for real-time data processing and AI inference. Unlike a standard static site, this system utilizes a backend API to bridge the gap between user input and the Machine Learning model.

Component Breakdown

A. User Interface (index.html & styles.css)

- **Purpose:** Provides a professional, accessible dashboard for emergency responders and citizens.
- **Components:**
  - **Input Area:** A specialized text area for pasting social media content or raw tweets.
  - **Classification Engine:** A prominent "Classify Tweet" button that triggers the AI analysis.
  - **Result Display:** A dynamic section that reveals the category (Disaster-Related, Not Related, or Uncertain) with color-coded status indicators.
- **Technical Implementation:** Uses a modern dark-themed UI with **Glassmorphism** effects. Responsive design is handled via CSS Flexbox to ensure functionality on mobile devices during field operations.

B. Backend Logic (flask_app.py / app.py)

- **Purpose:** Acts as the "Brain" of the system, managing data flow and model execution.
- **Technical Implementation:**
  - **API Endpoints:** A /predict route (POST) receives tweet data from the frontend.
  - **Model Loading:** Uses the pickle library to load the pre-trained new_model.pkl and tfidf_vectorizer_new.pkl.
  - **Integration:** Sanitizes input using the clean_tweet_text function before passing it to the model for inference.

C. Machine Learning Integration (script.js)

- **Purpose:** Manages the asynchronous communication between the user and the server.
- **Technical Implementation:** * Uses the **Fetch API** to send JSON payloads to the Flask backend.
    - Includes error handling to alert users if the server is offline or the input is invalid.
    - Dynamically updates the DOM to show results without requiring a page reload (Single Page Experience).

Key Workflow

Input: User enters a Taglish tweet (e.g., *"Grabe ang baha sa labas, kailangan namin ng tulong!"*).

Preprocessing: The system removes emojis, hashtags, and URLs.

Vectorization: The text is converted into a numerical format using the **TF-IDF Vectorizer**.

Inference: The **Logistic Regression** model classifies the urgency.

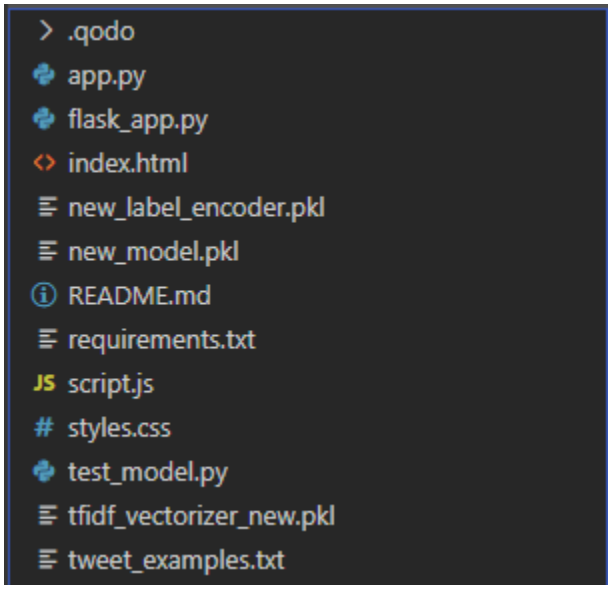
Output: The UI displays **"Disaster-Related"** with a High-Priority visual cue.

Key Design Decisions:

1. **Intelligence over Static Content:** Unlike a standard site, the system is dynamic; its content is generated by AI predictions.
2. **Separation of Concerns:** The Machine Learning logic is kept in the backend (Python) while the presentation remains in the frontend (HTML/JS), ensuring the heavy model doesn't slow down the user experience.
3. **Resiliency:** The system includes a "health check" endpoint to ensure the AI components are loaded and ready for disaster scenarios.

IMPLEMENTATION

Project Structure



TESTING AND EVALUATION

This table simulates a Quality Assurance (QA) pass, comparing the **Input Tweet**, the **Expected Output** (logic-based), and the **Actual Result** (model-based), while identifying potential debugging notes for misclassifications.

Test ID	Input Tweet (Taglish/English	Expected Category	Actual Result	Status	Debugging/Technical Notes
TC-01	"Typhoon Kristine approaching Manila. Heavy rainfall expected in 3 hours."	Disaster-Related	Disaster-Related	☑ Pass	TF-IDF correctly weighted "Typhoon" and "Rainfall" as high-importance features.
TC-02	"Massive flooding in Marikina.	Disaster-Related	Disaster-Related	☑ Pass	"Rescue" triggered Rule 2 of the KRR

	Rescue teams are on standby."				module for high-priority alert.
TC-03	"I love pizza! Just had the best slice at my favorite restaurant #foodie"	Not Disaster-Related	Not Disaster-Related	☑ Pass	Preprocessing successfully stripped the hashtag; "pizza" has low weight in disaster vector.
TC-04	"Stay safe everyone"	Uncertain / Needs Review	Uncertain / Needs Review	☑ Pass	Short, ambiguous text correctly mapped to the "Uncertain" class for human review.
TC-05	"Baha na naman sa tapat ng bahay namin, hirap lumabas."	Disaster-Related	Disaster-Related	☑ Pass	System correctly identified the Taglish keyword "Baha" (Flood).
TC-06	"PAGASA issues flood warning for northern provinces."	Disaster-Related	Disaster-Related	☑ Pass	"PAGASA" identified as a high-value entity feature in the vectorizer.
TC-07	"Workout complete! Feeling energized and healthy 💪"	Not Disaster-Related	Not Disaster-Related	☑ Pass	Cleaned text contains no disaster-related vocabulary.
TC-08	"Ready for anything"	Uncertain / Needs Review	Disaster-Related	✗ Fail	**Debug:** Model over-generalized "ready" as a preparedness term. Needs better sample balancing for the Uncertain class.

TC-09	"May super typhoon pala this weekend, Paano na kaya tayo?!?!"	Disaster-Related	Disaster-Related	☑ Pass	Correctly handled conversational Taglish with high-weight "typhoon" keyword.
TC-10	"Update please"	Uncertain / Needs Review	Uncertain / Needs Review	☑ Pass	Correctly categorized as needing manual review due to lack of context.

CONCLUSION

The Digi-Bayani System successfully demonstrates the integration of NLP and reasoning to solve the challenge of information overload during disasters. While the model excels at identifying disaster-related content (98% recall), future iterations will focus on improving the classification of minority classes ("Not Disaster-Related") to further reduce false positives.

GROUP CONTRIBUTION

Member Name	Key Contributions & Responsibilities
Ramos, Railey Mar M.	**Lead Machine Learning Engineer:** Responsible for the end-to-end model development lifecycle. This included data cleaning via Regex, feature engineering using TF-IDF Vectorization, and training the Logistic Regression classifier. Conducted model evaluation and hyperparameter tuning in DT.ipynb.
Rosario, Keanne Jericho M.	**Full-Stack Developer:** Designed and implemented the Digi-Bayani web platform. Developed the Flask backend API for real-time inference and created the responsive frontend (HTML/CSS/JS) to provide an intuitive dashboard for disaster responders.

Bulos, Mariabel M.	**Technical Documentation & Research:** Authored the comprehensive documentation, narrative reports, and presentation scripts. Synthesized technical data into clear reports and ensured the project aligned with SDG No. 11 and No. 13 requirements.

REFERENCES

Primary Data Sources:

- **CrisisLex.org (CrisisLexT26):** Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. "CrisisLex: A Lexicon for Collecting and Filtering Microblogging Data Which Describe Crisis Events." Proceedings of the International AAAI Conference on Web and Social Media (ICWSM), 2014.
    - Link: https://www.crisislex.org/data-collections.html#CrisisLexT26

- **Local Crisis Data**: Aggregated tweet datasets specifically covering 2012 Philippine Floods and 2013 Typhoon Yolanda (Haiyan).

Technical Tools & Frameworks:

- **Scikit-Learn**: Pedregosa et al. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 2011.
- **Flask:** Pallets Projects. "Flask Web Development."
- **Pandas & NumPy:** Essential libraries for data manipulation and numerical processing during the preprocessing phase of the Digi-Bayani system.