**Documentation for News Article Categorization System**

Name: Saniya Joseph

Email: saniyajoseph5999@gmail.com

The objective of this project is to develop a system that collects news articles from various RSS feeds, stores them in a database, and categorizes them into predefined categories:

* Terrorism/protest / political unrest/riot
* Positive/Uplifting
* Natural Disasters
* Others

The system is built using Python and involves RSS feed parsing, data storage using a relational database, text classification using Natural Language Processing (NLP), and batch processing to efficiently handle large numbers of articles.

**Technologies Used**

Programming Language: Python

**Libraries:**

Feed parser: To parse RSS feeds and extract article data.

SQLAlchemy: For database interaction and ORM (Object-Relational Mapping).

SQLite: A relational database used for storing news articles.

NLTK/spaCy: For text classification (Natural Language Processing).

Multiprocessing: This is used for the parallel processing of news articles.

Database: SQLite (can be easily switched to PostgreSQL or MySQL).

**Implementation Logic**

1. **RSS Feed Parsing**

The application uses the feed parser library to extract relevant fields from news articles including:

* Title
* Content
* Publication Date
* Source URL

Duplicate articles are handled by checking the database before storing new ones.

1. **Database Schema**

The database is designed using SQLAlchemy ORM, with a NewsArticle class representing the schema:

* Title (Primary Key)
* Content
* Publication Date
* Source URL
* Category (to store the predicted category after classification)

1. **Classification**

The application uses a text classification model to categorize the articles into predefined categories. The classify\_article function performs text classification, which can be built using spaCy or NLTK.

1. **Batch Processing**

The batch processing system is designed to handle large volumes of articles efficiently. Instead of processing one article at a time, the system processes them in batches.

1. **Exporting Data**

After processing, the articles (with their assigned categories) can be exported into CSV format. This allows for easy sharing and analysis.

1. **Running the Application**

**To run the application:**

* RSS Feed Parsing: Run the parse\_feeds function to collect articles from the specified RSS feeds.
* Processing & Classification: Use process\_articles(parsed\_articles) to categorize articles.
* Export Data: Run the export\_to\_csv function to generate a CSV file with the results.

Each step can be run sequentially in Jupyter Notebook to demonstrate the functionality.

**Conclusion**

This project demonstrates the ability to:

* Build an ETL (Extract, Transform, Load) pipeline using RSS feeds.
* Efficiently handle large datasets using batch processing.
* Use NLP techniques for article categorization.
* Store and retrieve data from a relational database, making it scalable to handle various news feeds and topics.