**# News Article Sentiment Classification**

This project, part of the NewsNuggets application, is a binary classification system that categorizes news articles into positive or negative sentiments. The system utilizes machine learning models trained on text data that undergo preprocessing, vectorization, and classification stages. It incorporates five different models: Logistic Regression, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, and Multinomial Naive Bayes.

**## Dataset Used:**

The dataset is taken from Kaggle for news sentiment classification. Data is originally collected through scraping websites. Categories were: Positive and negative

Dataset link: <https://www.kaggle.com/datasets/hoshi7/news-sentiment-dataset>

**## Project Structure:**

- **DataLoader/:** Contains data\_loader.py for loading the dataset.

- **DataPreprocessing/:** Contains data\_preprocessing.py for text preprocessing including removal of URLs, punctuations, tokenization, and more.

- **TextVectorizer/:** Contains vectorizer.py for transforming text into TF-IDF and BoW vectors and doing oversampling using SMOTE.

- **Models/:** Contains train\_val\_test.py for training and evaluating models with hyperparameter tuning.

- **Results/:** Storage for best models and their evaluations.

- **logs/:** Contains logs of the processes.

**## Prerequisites:**

- **Python** 3.x

- **Libraries**: pandas, numpy, scikit-learn, nltk, scikit-optimize, imblearn

- Ensure **NLTK** resources (punkt, stopwords, wordnet) are downloaded as the code will attempt to do so.

**## Setup:**

1. Unzip the 'sentiment\_Classification.zip' file to your local machine.

2. Navigate to the unzipped directory.

3. Set up a virtual environment:

4. Install required Python packages using the requirements.txt file :

pip install -r requirements.txt

**## Configuration:**

Adjust the logging and file path settings in main.py according to your local environment.

Running the Code:

To run the project, execute the main.py script:

**python main.py**

This script performs the following actions:

- Initializes logging.

- Loads the dataset from the specified path.

- Preprocesses the text data.

- Vectorizes the text data using TF-IDF and BoW.

- Uses SMOTE to balance both positive and negative classes

- Trains different models with hyperparameter tuning using Bayesian optimization.

- Evaluate models and save the outputs in the Results directory.

**Estimated Running Time:**

- The complete execution of the script may take approximately 20 minutes, depending on your machine's specifications and the size of the dataset.

**## Outputs:**

The system will generate:

- Vectorized text data.

- Trained model files (\*.pkl).

- Model evaluation summaries and detailed reports.

**## Logging:**

Logs are written to application.log in the logs directory, capturing detailed events and errors throughout the execution process.

**## Authors:**

- [ANITHA BALACHANDRAN, MADHURA BHATSOORI, SOWJANYA PAMULAPATI,

SOWMYA KURUBA, SWATHI RAMESH ]

**Note:**

While executing main.py, you may encounter "UserWarning" messages due to the Bayesian optimization process revisiting previously evaluated points, which is a normal part of the optimization algorithm. These warnings are not necessarily indicative of an issue with your code or the optimization process itself. They simply inform you that the optimizer is revisiting points it has seen before due to the nature of Bayesian optimization, which balances exploration and exploitation to find the optimal solution efficiently.

**Team Contribution:**

A table with text on it

Description automatically generated