Web Scraping and Classification Web App Documentation

Created with Streamlit, Requests, BeautifulSoup, Celery, NLTK, and MySQL

Contents:

Project Document Structure:
/root
/\

artifacts
contains
model and
vectors

archives
contains dataset
used to train model
on categories

1. Introduction

This documentation outlines the features and functionality of a web application built with Streamlit, designed to scrape news data from specified URLs or dropdown selections. The scraped data is then classified using a RandomForestClassifier model with a 94%+ accuracy, trained on a custom dataset created for this purpose.

2. Requirements

To run the web app, ensure that you have the following installed:

Python 3.x

Streamlit

Requests

BeautifulSoup

NLTK

MySQL

Or run **pip3 install -r requirements.txt** which is available in the project directory.

3.Usage

Navigate to the project directory.

Run the Streamlit app:

```
streamlit run app.py
```

Access the web app in your browser at http://localhost:8501.

4. Web App Features

Input Options:

Enter a URL directly or select from a dropdown list of pre-defined URLs.

Data Scraping:

Utilizes the requests and BeautifulSoup libraries to scrape news data from the specified URLs.

Data Classification:

Employs a Multinomial model with over 93% accuracy for classifying news descriptions.

Classification categories include:

"political"

"positive"

"protest"

"riot"

"terror"

“disaster”

"other"

Pre-processing:

NLTK is used for natural language processing and text pre-processing to enhance the accuracy of the classification model.

Save to MySQL Database:

Integrates with MySQL for storing the scraped and classified data.

5. Data Classification Model

The Multinomial model used for classification is trained on a custom dataset.

The dataset includes labelled examples for each of the specified categories.

The training process involves tokenization, vectorization, and model training using NLTK.

6. MySQL Database

The web app is configured to connect to a MySQL database for storing the scraped and classified data. Ensure that you have a MySQL server running and provide the necessary credentials in the app.

7. Celery Integration

Celery is integrated into the application for asynchronous task execution. This ensures smooth and efficient processing of scraping and classification tasks, especially for large datasets.

8. Conclusion

This web app serves as a powerful tool for scraping news data, classifying it with high accuracy, and storing the results in a MySQL database. By combining the capabilities of Streamlit, Requests, BeautifulSoup, NLTK, and MySQL, it provides a comprehensive solution for extracting valuable insights from online news content.

Feel free to customize and extend the functionality based on your specific requirements and datasets.

Note: Project is built on windows and tested on windows. If **Celery** fails to run on your pc try installing **RabbitMQ server** or run **app2.py.**

Submitted by
Akash
akash.hiremath25@gmail.com