

Sentiment Analysis Project with Streamlit

Overview:

This project focuses on building a sentiment analysis model using Logistic Regression. The goal is to classify text reviews as either positive or negative or neutral. I utilized a Twitter dataset containing text reviews and their corresponding sentiments. Additionally, a Streamlit web application is implemented to provide a user-friendly interface for real-time sentiment prediction.

Setup:

Prerequisites

Ensure you have the following libraries installed:

- pandas (Pandas: For data analysis)
- nltk (NLTK: For providing text processing tools.)
- scikit-learn (scikit-learn: For providing machine learning tools and utilities.)
- joblib (joblib: For model serialization.)
- streamlit (Streamlit: For providing an easy-to-use interface for creating web applications.)

You can install them using pip:

```
>>>pip install pandas nltk scikit-learn joblib streamlit
```

NLTK Data

Download the required NLTK data files using the NLTK library.

Dataset

Ensure you have the dataset `Twitter_Data.csv` in the working directory.

Project Structure:

1. Loading and Preprocessing Data

The `load_data` function reads the dataset and preprocesses it by filling missing values and removing any remaining rows with NaN values.

2. Text Preprocessing

The `preprocess_text` function cleans the text data by removing HTML tags, punctuation, and stop words. It also handles negation by appending "not_" to words following negation words.

3. Training the Model

The `train_model` function preprocesses the data, splits it into training and testing sets, vectorizes the text data using TF-IDF, scales the features, and trains a Logistic Regression model.

4. Saving and Loading the Model

Functions are provided to save and load the trained model, vectorizer, and scaler.

5. Predicting Sentiment

A function is available to preprocess and predict sentiment for a given review.

6. Streamlit UI

The `main` function sets up the Streamlit interface with a title, text area for input, and a button to trigger sentiment prediction.

Running the Streamlit Application:

1. Install Streamlit

sh:

```
>>>pip install streamlit
```

2. Save the Code

Save the code in a Python file, e.g., `sentiment_analysis.py`.

3. Run the Streamlit Application

Open a terminal or command prompt and run:

sh:

```
>>>streamlit run sentiment_analysis.py
```

or

```
>>>python -m streamlit run sentiment_analysis.py
```

4. Upload the Dataset

Ensure the dataset `Twitter_Data.csv` is in the same directory as your script or provide the correct path in the `data_url` variable.

5. Interact with the Application

Open the URL provided by Streamlit in your browser and start interacting with the application.

Explanation of Key Components:

Loading and Preprocessing Data

- Data Loading: The dataset is loaded from a CSV file, and relevant columns (`review` and `sentiment`) are selected. Missing values in the `review` column are filled with empty strings, and any remaining rows with NaN values are dropped.
- Text Preprocessing: The text data is cleaned by removing HTML tags, punctuation, and stop words. It also handles negation by appending "not_" to words following negation words.

Model Training and Evaluation

- Data Splitting: The data is split into training and testing sets.
- Text Vectorization: The text data is vectorized using TF-IDF with a maximum of 5000 features.
- Feature Scaling: The features are scaled using `StandardScaler`.
- Model Training: A Logistic Regression model is trained on the processed data.
- Model Evaluation: The model's accuracy is evaluated on the test set and displayed.

Saving and Loading the Model

- The trained model, vectorizer, and scaler are saved using `joblib`.
- The saved model, vectorizer, and scaler can be loaded for later use.

Sentiment Prediction

- A function to preprocess and predict the sentiment of a given review using the trained model and vectorizer.

Streamlit Integration

- Streamlit Interface: The `main` function sets up the Streamlit interface with a title, text area for input, and a button to trigger sentiment prediction.
- User Interaction: Users can enter a review and click the "Predict Sentiment" button to see the predicted sentiment.

Conclusion:

The sentiment_analysis_bot can successfully predict the sentiments as positive or negative or neutral.