NewsClassifier: Building an Automated News Classification System with NLP Techniques

Building a news classification system involves several steps, including web scraping, data preprocessing, and model training. Below is a high-level outline of the steps you can follow using Natural Language Processing (NLP) techniques:

1. **Web Scraping:**
   - Choose news websites (e.g., BBC, The Hindu, Times Now, CNN) and use web scraping tools or libraries (e.g., BeautifulSoup,Selenium) to extract news articles.
   - Retrieve the title and content of each news article. Ensure that you have a diverse dataset covering various topics.

2. **Data Cleaning and Preprocessing**:
   - Remove any irrelevant information, such as HTML tags, advertisements, or non-text content.
   - Tokenize the text (split it into words or subwords) and remove stop words.
   - Perform lemmatization or stemming to reduce words to their base form.
   - Handle missing data and ensure a consistent format.

3. **Text Representation**:
   - Convert the text data into numerical format suitable for machine learning models. This can be done using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (e.g., Word2Vec, GloVe).
   - Consider using pre-trained word embeddings for better performance.

4. **Topic Clustering**:
   - Apply clustering algorithms (e.g., K-means, hierarchical clustering) on the preprocessed text data to group similar articles together.
   - Choose the number of clusters based on the topics you want to identify (e.g., Sports, Business, Politics, Weather).

5. **Topic Labeling:**
   - Manually inspect a sample of articles in each cluster to assign topic labels. This step helps in labeling the clusters with meaningful topics.

6. **Classification Model**:
   - Split the data into training and testing sets.
   - Train a supervised machine learning model (e.g., Naive Bayes, Support Vector Machines, or deep learning models like LSTM or BERT) to predict the topic of a news article.
   - Use the labeled clusters as ground truth labels for training the model.

7. **Evaluation**:
   - Evaluate the performance of your classification model on the testing set using appropriate metrics (accuracy, precision, recall, F1-score).
   - Fine-tune the model parameters if needed to improve performance.

8. **Deployment:**
   - Deploy a classification application using streamlit application