Will Bazal

CS 410 Text Information Systems

News Article Keyword Classification

Overview

This Python program was designed to classify news articles into pre-defined categories based on their content. The dataset used in this program is comprised of BBC news article data and can be found on Kaggle. This program used Natural Language Processing (NLP) techniques in order to process text and develop machine learning models. The models included in this project are Logistic Regression and K-Nearest Neighbors. This program provides the functionality to generate word clouds for the different categories from the dataset, as well as allows the user to input custom topics for the program to try and correctly classify. This program only had so much data to train on, so it will also include an overall confidence level estimated for each custom input and their predicted classification.

Usage Description

1. Required Libraries – Make sure you have the following libraries installed at the very beginning of the model:
    - pandas
    - matplotlib
    - seaborn
    - wordcloud
    - nltk
    - scikit-learn

    - You can install these libraries using the following command:

    ```
    pip install pandas matplotlib seaborn wordcloud nltk scikit-learn
    ```

2. Required Dataset – This Python program requires a CSV file named **BBC News.csv**, which is included in the project files in GitHub. In this file, the "Category" column represents the predetermined category for each article. The "Text" column contains the content of the news articles themselves.

3. Running the Python Program
    - When you open the Python file, simply run the script. The program will load through every step. It will load the dataset, process the text, and create visualizations where appropriate.
    - It will also train a Logic Regression and a K-Nearest Neighbor learning model. Once that is complete, it will provide performance metrics for each model.

4. Classification for Customer User Input
    - After the models have been trained, the program will ask you to type in custom topics in the command line that pops up. Once a topic is entered, the program will try to predict the category for your input and provide a confidence score as well.

Software Usage Instructions

1. Prerequisites
    - Make sure you have Anaconda Navigator installed on your computer. This will allow you to access Jupyter Notebook and run my code.
    - In the associated GitHub repository, be sure to download the 'News Article Keyword Classification.ipynb' and 'BBC News.csv' files.

2. Launch Jupyter Notebook
    - To open Jupyter Notebook, you can search in your search bar for "Anaconda Navigator". Open that and navigate to Jupyter Notebook.
    - Once Jupyter Notebook is open, create a new folder and upload the two files from GitHub to this folder. The Python script and csv files need to be in the same folder location for the script to run properly.

3. Run the Script
    - Once both files are in a shared folder, open the Python script.
    - At the top of the screen, click on "Cell" and then "Run All". This will run through the entire code. You will get a chance to see the category distribution and word cloud charts as you scroll down.

4. Interact with the Script
    - Once the script finishes running, you will see a box appear at the very bottom that will ask you to enter a topic for the program to classify. Provide an input in that box and click enter.
    - The script will return the category that it believes your topic falls into. Since this program isn't terribly accurate due to not having more data to train on, a confidence percentage will also be shown.
    - If you want to try another topic, you can enter one into the box again.
    - When you are done trying out topics, you can type "exit" and the program will quit.

5. Troubleshooting
    - File Not Found Error: Make sure the Jupyter Notebook file and CSV file are located in the same folder.
    - If the notebook kernel crashes, restart it from the "Kernel" menu in Jupyter Notebook.
    - If you run into issues with missing libraries, install them using pip. For example, "pip install pandas numpy sklearn".

Software Implementation Details

1. Libraries Being Used
    - **pandas**: This is used for data manipulation and loading in the BBC News dataset.
    - **matplotlib and seaborn**: These are two common libraries used for creating data visualizations. In this case, they created the category distribution and world cloud visuals.
    - **re**: This is used for text cleaning (removing extra spaces and special characters).
    - **nltk**: This library is commonly used for text processing. In this case, it helps tokenization and stop word removal.
    - **scikit-learn**: This is a popular machine learning library for creating models (Logistic Regression and K-Nearest Neighbors), as well as TF-IDF vectorization, and splitting the data into train and test sets.

2. Program Flow Overview
    - The data for this project comes from the '**BBC News**' CSV file. If the code is unable to find and connect to the file, the program will an error message.
    - The **process_text** function cleans and tokenizes the text data.
    - The **TfidfVectorizer** is used to convert the cleaned text data into numerical features.
    - The program then uses **train_test_split** to split the data into training and test sets.
    - The **Logistic Regression** and **K-Nearest Neighbors** classification models are trained on the vectorized data.
    - Each models' **accuracy scores** and **classification reports** are then printed to show and evaluate their performance.

3. Program Functions
    - **create_wordcloud(words)**: This function creates word clouds for each news category.
    - **process_text(text)**: This function cleans and tokenizes the text. This process removes special characters and stop words.
    - **predict_user_input(user_input)**: Process input from user, transforms it into the same feature space as the training data, and uses the Logistic Regression model to predict the category this input likely belongs to. The function also returns a confidence score of this predicted category grouping.

Future Improvements

- **Model Tuning**: The performance of both the Logistic Regression and K-Nearest Neighbors (KNN) models could be improved by fine-tuning their hyperparameters. For example, adjusting the number of neighbors for KNN or the regularization strength for Logistic Regression could lead to better accuracy and more reliable predictions. This could be done using methods like grid search or random search to find the possible parameter combinations. This model is also not terribly accurate, so adding training on a larger dataset would help improve overall results.

- **Additional Models**: While Logistic Regression and KNN are great choices, it might be worth experimenting with other models like Support Vector Machines (SVM) or Random Forests. Trying out these models could reveal better options for improving clarification accuracy.
- **User Interface**: To make the program a bit more user-friendly, simple app with a better user interface could be created. Instead of requiring users to type commands, this interface could give them options to interact using buttons, input fields, or dropdown menus. This would primarily make it easier for people who lack a technical background to classify text.

Conclusion

This News Article Keyword Classification project demonstrated how Natural Language Processing (NLP) and machine learning can classify BBC news articles into categories using Logistic Regression and K-Nearest Neighbors (KNN). Key steps included data preprocessing, TF-IDF vectorization, and model training, with Logistic Regression achieving higher accuracy due to its ability to handle sparse, high-dimensional data. The user input prediction feature displayed the model's practical use, allowing real-time classification of news topics.

Feature improvements could include larger datasets and more advanced models, like SVM or neural networks, in order to improve context understanding. Overall, this project highlights the essential role of preprocessing, feature extraction, and model selection in text classification. It also shows potential applications in areas like sentiment analysis, content recommendation, and automated news categorization.