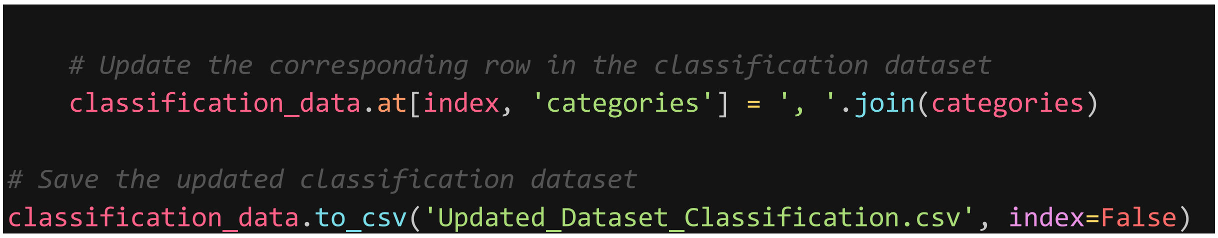
OJASHWEE RAMAN

Email: ojashwee.raman2021@vitstudent.ac.in | Phone: 9693592086 | LinkedIn: linkedin.com/in/ojashweeraman

Classification of articles based on title and content(LLM approach).

Code:

```
import requests
import pandas as pd

# Read the datasets
classification_data = pd.read_csv('Dataset_Classification.csv')
categories_data = pd.read_csv('Categories_Description.csv')

# API endpoint
url = "https://api.meaningcloud.com/class-2.0"

# Your MeaningCloud API key
api_key = "7ed73d18f6626604b800815aa305f996"

# Iterate through each row in the classification dataset
for index, row in classification_data.iterrows():
    content = row['content']

    # Construct the payload
    payload = {
        'key': api_key,
        'txt': content,
        'model': 'SocialMedia_en',
        'of': 'json'
    }

    # Make the API request
    response = requests.post(url, data=payload)
    result = response.json()

    # Extract the categories
    if 'category_list' in result:
        categories = [category['label'] for category in
result['category_list']]
    else:
        categories = []
```



```
    # Update the corresponding row in the classification dataset
    classification_data.at[index, 'categories'] = ', '.join(categories)

# Save the updated classification dataset
classification_data.to_csv('Updated_Dataset_Classification.csv', index=False)
```

Method Used:

The method used in the code is text categorization using the MeaningCloud API. Text categorization, also known as text classification, is the process of assigning predefined categories or labels to text documents based on their content. In this case, the content of the text in the 'content' column of the Dataset_Classification.csv dataset is being categorized using the descriptions and names of categories from the Categories_Dataset.csv dataset.

Libraries Used:

The code utilizes the ‘requests’ library for making HTTP requests to the MeaningCloud API and the ‘pandas’ library for handling data in tabular form.

- requests: This library is used to send HTTP requests to web services or APIs. It's employed here to send text content to the MeaningCloud API for categorization and to receive the API response.
- pandas: Pandas is used to work with structured data, especially in tabular form. It's used to read CSV files into DataFrames, manipulate the data, and save the updated data back to CSV.

Working of the Code:

- Import necessary libraries: The code starts by importing the required libraries, namely `requests` and `pandas`.
- Read datasets: The code reads the `Dataset_Classification.csv` and `Categories_Dataset.csv` datasets using the `pd.read_csv()` function from the `pandas` library.
- API endpoint and key: The URL for the MeaningCloud API's categorization service is stored in the `url` variable. The API key (which you need to replace with your actual API key) is stored in the `api_key` variable.
- Iteration over rows: The code iterates through each row in the `classification_data` DataFrame using a loop. For each row, it extracts the 'content' column's text.
- API request construction: The code constructs a payload for the API request. The payload includes the text content, the chosen model (`SocialMedia_en ` in this case), and the desired output format (`json`).

- API request and response: The code uses the `requests.post()` function to send a POST request to the MeaningCloud API's categorization endpoint. It includes the payload in the request. The response is received and converted to a JSON format using `.json()`.
- Category extraction: The code extracts the categorized labels from the API response. If the response contains a 'category_list', it extracts the labels. If not, it assigns an empty list.
- Updating the DataFrame: The extracted categories are joined into a comma-separated string and stored in the 'categories' column of the current row in the `classification_data` DataFrame.
- Saving the updated dataset: After iterating through all rows, the updated classification data (with added 'categories' column) is saved to a new CSV file using the `to_csv()` function.
- The final output is an updated `Dataset_Classification.csv` file containing an additional 'categories' column with the assigned categories for each text content, based on the results obtained from the MeaningCloud API. CSV file named 'Final_Classified_Dataset.csv'.