**Scenario:**

You are a data scientist at a reputable news analysis company that wants to enhance its capabilities in automatically categorizing news articles into different topics or categories. This will enable the company to provide more accurate and efficient news analysis to its clients. You have been assigned the task of building a **news category classifier** and deploying it as an **API** for real-time inference.

**Task 1: Data Scraping**

1.1. Identify and select three or more reliable news publishers or websites (e.g., BBC News, Reuters, CNBC) known for their diverse news categories.

1.2. Write a web scraping script (**Python**) using libraries (**BeautifulSoup** and **Selenium**) to collect news articles from these sources. Scrape a minimum of **500** articles from each publisher, ensuring that you collect the following information for each article: published_date, headline, publisher, article_content, and category (e.g., politics, sports, technology).

**Task 2: Data Preprocessing**

2.1. Clean the dataset by handling missing values, duplicate records, and any irrelevant information.

2.2. Perform text preprocessing, which **may** include lowercasing, removing punctuation, stop words,stemming or lemmatization and tokenization. Ensure that the text data is ready for feature extraction.

2.3. In the case where your dataset exhibits an imbalance in the distribution of classes, you should consider addressing this issue to prevent the model from being biased towards the majority class.

2.4 Conduct exploratory data analysis (EDA) to gain insights into the dataset's characteristics. This may involve visualizations, summary statistics, and data distribution analysis.

**Task 3: Feature Extraction and Vectorization**

**Note:** Since the data is collected from various sources, it is necessary to establish a mapping system to ensure uniformity of the label

**Example:**

News A from Publisher A with category 'Living'. News B from Publisher B with category 'Religion'. You would like to map 'Living' and 'Religion' to the 'Social' category.

| News | Publisher | Previous Category | Mapped Category |
|---|---|---|---|
| A | BBC | Living | Social |
| B | Reuters | Religion | Social |
| C | CNBC | Technology | Technology |
| D | CNBC | Tech Check | Technology |
| E | CNBC | Technology: Companies | Technology |

***The table above serves as a sample for the category mapping*

3.1. Utilize techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (e.g., Word2Vec or GloVe) to convert the preprocessed text into numerical vectors suitable for machine learning.

**Task 4: Model Building**

4.1. Split the dataset into training and testing sets (e.g., 80% train, 20% test).

4.2. Choose an appropriate **multi-class** classification algorithm and train the model on the training data.

4.3. Evaluate the model's performance using suitable metrics on the testing data. (Note: discouraged to use **‘accuracy’** as you are required to train a multi-class model)

**Task 5: Model Evaluation and Reporting**

Write a report summarizing your work on building the news category classifier, including:

5.1. A description of the data sources and the number of articles collected from each publisher.

5.2. Details of the data preprocessing and feature extraction techniques used.

5.3. Information about the chosen classification algorithm and its performance metrics.

5.4. Insights into the model's strengths and limitations.

**Task 6: Bonus Challenge (Optional)**

6.1. Build an RESTful API that accepts HTTP requests containing text data and returns predictions of news categories.

6.2. Include clear instructions on how to run and test the API locally. (e.g. Postman)

**Files to submit**

Submit the a zip file containing

- A web scraping python script **(.py/ipynb format)**
- A model development script include preprocessing **(.py/ipynb format)**
- A reporting document **(.docx format)**