- **Python Libraries**: pandas, numpy, scikit-learn, matplotlib, seaborn, NLTK, spaCy, TensorFlow/Keras, XGBoost, etc.
- **Jupyter Notebook**: For developing and documenting the code.
- **Text Preprocessing Libraries**: NLTK, spaCy, scikit-learn (for vectorization), etc.
- **Visualization Libraries**: matplotlib, seaborn (for visualizing data distributions, model performance, etc.).

Part B : News Article Classification

1. Overview

In today's digital world, news articles are constantly being generated and shared across different platforms. For news organizations, social media platforms, and aggregators, classifying articles into specific categories such as sports, politics, and technology can help improve content management and recommendation systems. This project aims to develop a machine learning model that can classify news articles into predefined categories, such as sports, politics, and technology, based on their content.

By automating this process, organizations can efficiently categorize large volumes of news articles, making it easier for readers to access relevant information based on their interests.

2. Problem Statement

The primary objective of this project is to build a classification model that can automatically categorize news articles into different predefined categories. The model will be trained using a labeled dataset of news articles and will output the most likely category (e.g., sports, politics, or technology) for any given article.

The goal is to:

- Develop a robust classifier capable of handling articles from multiple categories.
- Preprocess the text data, extract meaningful features, and train models to classify the articles.
- Evaluate the model performance and provide actionable insights on how well it classifies articles.

3. Dataset Information

The dataset can be used from **data_news.**

4. Deliverables

1. Data Collection and Preprocessing (5 Marks):

- Collect a dataset of labeled news articles (sports, politics, technology etc).
- Clean and preprocess the text data.
- Handle missing data, if any, and ensure the text is ready for feature extraction.

2. Feature Extraction (10 Marks):

- Use methods like TF-IDF, word embeddings (e.g., Word2Vec, GloVe), or bag-of-words to convert text data into numerical features.
- Perform exploratory data analysis (EDA) to understand the distribution of different categories.

3. Model Development and Training (20 Marks):

- Build classification models using algorithms like Logistic Regression, Naive Bayes, Support Vector Machines (SVM).
- Train the models on the preprocessed text data, tuning hyperparameters as necessary.
- Use cross-validation to ensure robust evaluation of model performance.

4. Model Evaluation (5 Marks):

- Evaluate the models using appropriate metrics.
- Compare the performance of different models and select the best one for classification.

5. Final Report and Presentation (10 Marks):

- Create a report summarizing the entire process, from data collection to model evaluation, and present the findings.
- Include visualizations of model performance and feature importance, if applicable.
- Prepare a video or slide presentation (not exceeding 5 minutes) explaining the methodology, models, and results.

5. Success Criteria

The project will be deemed successful if:

- The classification model achieves good performance metrics (accuracy, F1-score, etc.).
- The model can successfully classify new, unseen news articles into the correct categories (sports, politics, technology).
- Insights regarding the most important features or keywords driving classification are derived from the model.
- The process and methodology are clearly documented and presented.

Submit Guidelines

- **Submit in jupyter notebook file (.ipynb file), report.**
- **Create a video of maximum of 5 mins explaining the analysis and share the drivelink.**

How to ZIP a folder: