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.

4. Deliverables

1. Data Collection and Preprocessing

- 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:

- 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:

- 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 :

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