**News Article Classification**

**1. Introduction**

The primary objective of this project is to build a classification model that can automatically categorize news articles into different predefined categories. The dataset used contains 50,000 news articles, each labeled with one of 10 categories: WELLNESS, POLITICS, ENTERTAINMENT, TRAVEL, STYLE & BEAUTY, PARENTING, FOOD & DRINK, WORLD NEWS, BUSINESS, and SPORTS. The dataset is balanced, with each category containing 5,000 articles.

**2. Problem Statement**

The goal is to develop a machine learning model that can accurately classify news articles into their respective categories based on the text content. This involves preprocessing the text data, extracting relevant features, and training a classification model.

**3. Dataset Overview**

The dataset consists of the following columns:

* **category**: The category of the news article.
* **headline**: The headline of the news article.
* **short\_description**: A brief description of the news article.
* **keywords**: Keywords associated with the news article.

The dataset contains 50,000 rows and 5 columns initially, with some null values in the keywords column, which were replaced with empty strings.

**4. Data Preprocessing**

**4.1 Text Cleaning**

The text data was preprocessed to remove special characters, converted text to lowercase, tokenized the text, removed stopwords, and lemmatized the words. The following steps were taken:

* **Lowercasing**: Converted all text to lowercase.
* **Removing Special Characters**: Removed non-alphabetic characters.
* **Tokenization**: Split the text into individual words.
* **Stopword Removal**: Removed common stopwords.
* **Lemmatization**: Reduced words to their base or root form.

**4.2 Feature Engineering**

* **TF-IDF Vectorization**: The cleaned text was converted into numerical format using TF-IDF vectorization with a maximum of 5,000 features.
* **Bag of Words (BoW)**: The cleaned text was also converted into numerical format using Count Vectorization with a maximum of 5,000 features.
* **Textual Features**: Additional features such as word count, character count, average word length, and sentence count were extracted from the preprocessed text and created the text to analyze by joining the headline,description and Keywords as Clean Text. Which further turned to preprocessed text for model Training.

**5. Model Development and Training**

**5.1 Train-Test Split**

The dataset was split into training and testing sets with an 80-20 split for both TF-IDF and BoW features.

**5.2 Model Selection**

Training the model with various Machine Learning models to test our datasets to select the best models for our NEWS Article classification.

* **Logistic Regression**
* **Multinomial Naive Bayes**
* **Bernoulli Naive Bayes**
* **Support Vector Machine (SVM)**

**5.3 Model Performance**

* **Logistic Regression**: Achieved an accuracy of 79.89% with TF-IDF features and 78.59% with BoW features.
* **Multinomial Naive Bayes**: Achieved an accuracy of 78.11% with TF-IDF features and 78.22% with BoW features.
* **Bernoulli Naive Bayes**: Achieved an accuracy of 78.14% with both TF-IDF and BoW features.
* **Support Vector Machine (SVM)**: Achieved an accuracy of 79.18% with TF-IDF features and 75.62% with BoW features.

**5.4 Hyperparameter Tuning**

Hyperparameter tuning was performed on the Logistic Regression model using RandomizedSearchCV. The best parameters found were:

* **C**: 1
* **solver**: liblinear
* **max\_iter**: 200

The tuned Logistic Regression model achieved an accuracy of 79.50%.

**6. Conclusion**

The Logistic Regression model performed the best among the models tested, achieving an accuracy of approximately 80% with TF-IDF features. The model was further improved through hyperparameter tuning, resulting in a slight increase in accuracy.

**6.1 Key Takeaways**

* **Text Preprocessing**: Proper cleaning and preprocessing of text data are crucial for achieving good model performance.
* **Feature Engineering**: Both TF-IDF and BoW features were effective, with TF-IDF slightly outperforming BoW.
* **Model Selection**: Logistic Regression proved to be the most effective model for this classification task.
* **Hyperparameter Tuning**: Tuning the model's hyperparameters can lead to improved performance.