News Article Classification

Task 1: Data Exploration and Preprocessing - I worked with a dataset containing 50,000 news articles labeled across 10 balanced categories: *WELLNESS, POLITICS, ENTERTAINMENT, TRAVEL, STYLE & BEAUTY, PARENTING, FOOD & DRINK, WORLD NEWS, BUSINESS,* and *SPORTS*. The class distribution was perfectly even, with 5,000 articles per category.

I checked for missing values and found some in the short_description field. These were filled with empty strings to prevent null-related issues. I then combined headline and short_description into a new column called text to provide richer context for each article.

To clean the text, I:

- Converted all text to lowercase
- Removed numbers, punctuation, and special characters using regular expressions
- Removed stopwords using NLTK's English stopword list
- Applied lemmatization using WordNetLemmatizer to reduce words to their base form

These preprocessing steps helped normalize the text and reduce noise. I also reviewed article lengths to ensure the majority were within a reasonable range, confirming consistent data quality.

Task 2: Feature Engineering - I used **TF-IDF vectorization** to convert the preprocessed text into numerical features. The vectorizer was limited to the top 5,000 features to manage dimensionality while retaining the most important terms across the corpus. TF-IDF worked well for identifying category-specific keywords and gave the models a strong foundation.

I did not include additional textual features such as word count, average word length, or embedding-based techniques like Word2Vec or GloVe in this version. These approaches are good candidates for future experimentation, especially if deploying more advanced or deep learning models.

I also conducted basic **exploratory data analysis (EDA)**, visualizing the category distribution using a bar plot. This helped confirm the balance in classes and ensured there was no bias in label distribution.

Task 3: Model Development - I trained four classification models using the TF-IDF features:

- **Logistic Regression**
- **Multinomial Naive Bayes**
- **Support Vector Machine (SVM)**
- **Random Forest**

All models were trained using a standard 80/20 train-test split. I used default parameters for initial training and cross-validation to validate consistency across folds. The models were evaluated using the same dataset split to ensure a fair comparison.

Task 4: Model Evaluation - Each model was evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix

Here are the performance results:

- **SVM**: 90.1% accuracy
- **Logistic Regression**: 88.3% accuracy
- **Naive Bayes**: 82.5% accuracy
- **Random Forest**: 78.6% accuracy

SVM performed the best across all metrics and was selected as the final model for news classification. Confusion matrices were plotted to visualize category-level performance, confirming high accuracy across most labels.

Conclusion - This project successfully built a multi-class news article classifier using machine learning and NLP techniques. The data was thoroughly cleaned and vectorized using TF-IDF, and traditional models like SVM and Logistic Regression performed well. For future work, I plan to:

- Experiment with deep learning models (e.g., LSTM, BERT)
- Use word embeddings like Word2Vec or GloVe
- Add engineered features such as article length or lexical diversity
- Deploy the best model in a simple web app for real-time classification

Video Link – Here