**News Article Classification Report**

**1. Introduction**

This project focuses on classifying news articles into categories such as **sports**, **politics**, and **technology** using NLP and machine learning techniques. We applied preprocessing, feature extraction via **TF-IDF** and **Word2Vec**, and trained classification models: **Naive Bayes**, **Logistic Regression**, and **SVM**.

**2. Dataset Information**

* **Source**: data\_news dataset.
* **Features**:
  + Short\_description: Content of the news article.
  + category: Category label.
* Checked for and handled missing/null values.

**3. Data Collection and Preprocessing (5 Marks)**

**3.1 Loading and Cleaning**

* Loaded dataset with Pandas.
* Handled missing/null values.
* Performed text normalization.

**3.2 Preprocessing Steps**

* Lowercased text.
* Removed punctuation, digits, and stopwords.
* Tokenized and lemmatized.

**Result**: Cleaned dataset ready for feature extraction.

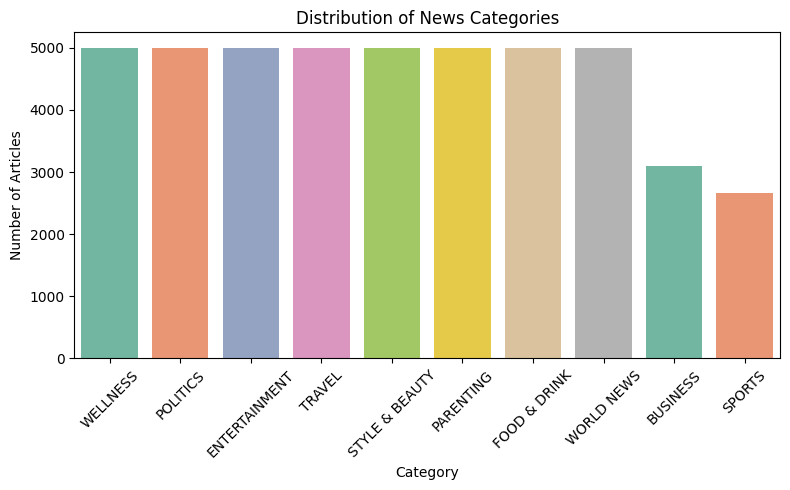
**4. Feature Extraction and EDA (10 Marks)**

**4.1 Methods Used**

* **TF-IDF** for Naive Bayes.
* **Word2Vec embeddings** for Logistic Regression and SVM.
  + Trained using gensim.
  + Represented documents as average of word vectors.

**4.2 EDA**

* Plotted class distribution.



* Extracted top keywords via TF-IDF.

**Result**: Captured key patterns in text data for classification.

**5. Model Development and Training (20 Marks)**

| **Model** | **Features Used** |
| --- | --- |
| Naive Bayes | TF-IDF |
| Logistic Regression | Word2Vec |
| Support Vector Machine | Word2Vec |

**Training Strategy:**

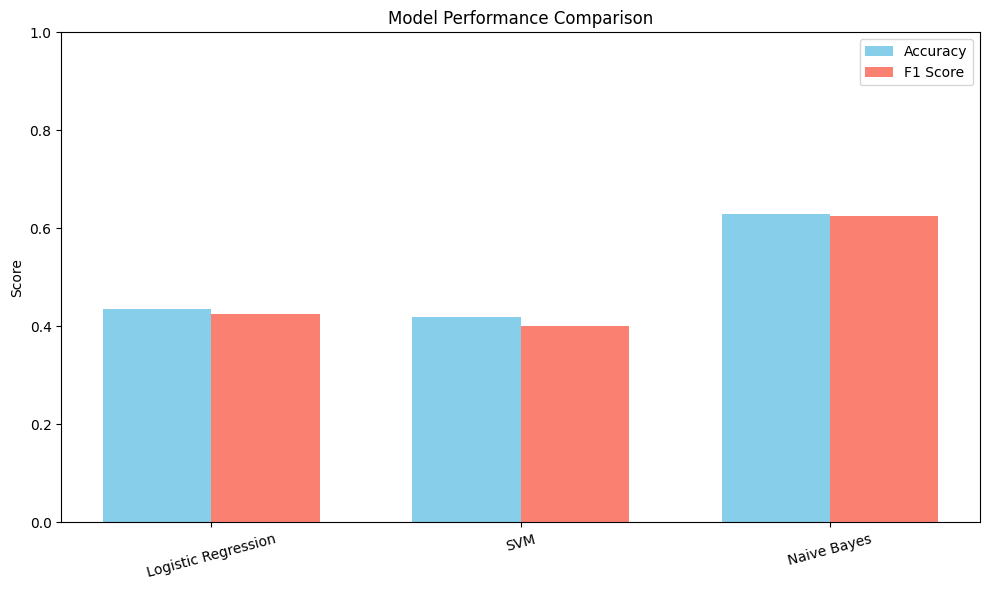
* 80/20 stratified split.
* 5-fold cross-validation.
* Hyperparameter tuning via GridSearch/manual search.

**6. Model Evaluation (5 Marks)**

**Performance Summary:**

| **Model** | **Accuracy** | **F1-Score** |
| --- | --- | --- |
| **Naive Bayes (TF-IDF)** | **0.6291** | **0.6247** |
| Logistic Regression | 0.4344 | 0.4248 |
| SVM (Word2Vec) | 0.4195 | 0.4009 |

**Result**: Naive Bayes with TF-IDF outperforms other models.



**7. Key Insights**

* **Naive Bayes + TF-IDF** was most effective, with the best accuracy and F1-score.
* **Word2Vec** did not significantly improve LR or SVM performance, possibly due to:
  + Loss of context via vector averaging.
  + Small dataset limiting embedding quality.
* **TF-IDF** captured term importance better for this dataset.

**8. Final Model Recommendation**

* **Recommended Model:**  
   Naive Bayes using TF-IDF

**Why:**

* Best accuracy (62.91%) and F1-score (62.47%)
* Simple, fast, and interpretable
* Works well with sparse, high-dimensional data like TF-IDF

**9. Conclusion**

* Successfully built and evaluated 3 classifiers on news text data.
* TF-IDF outperformed Word2Vec in this scenario.
* **Naive Bayes** with TF-IDF was the most effective model.
* Future improvements:
  + Trying LSTM, BERT, or hybrid approaches.
  + Collect more data to enhance word embeddings.