Part B: News Article Classification

---

1. Introduction

News articles cover a wide variety of topics—from politics and sports to travel and wellness. Automatically categorizing such articles into appropriate sections is an important NLP task for content management systems and personalized news feeds. In this project, we develop a machine learning pipeline to classify news articles into **10 predefined categories**, leveraging natural language processing (NLP) and classical ML models.

---

2. Dataset Overview

The dataset used contains **50,000 news articles**, each associated with:

- **Headline**
- **Short description**
- **Category** (target label)
- **Keywords** (optional)

Categories:

- BUSINESS
- ENTERTAINMENT
- FOOD & DRINK
- PARENTING
- POLITICS
- SPORTS
- STYLE & BEAUTY
- TRAVEL
- WELLNESS
- WORLD NEWS

Each category has an equal representation of 5,000 articles, ensuring a balanced multi-class classification problem.

---

3. Data Exploration

Initial exploration involved inspecting the first few rows, reviewing column data types, and visualizing category distribution. A bar chart confirmed equal distribution across all 10 classes.

We also computed the most common terms in articles. Frequently occurring tokens included: new, photo, make, year, get, like, say, world, best, and first—indicating a rich and general vocabulary that needed contextual refinement via vectorization.

---

4. Text Preprocessing

To prepare the textual data for modeling:

- **Combined headline + short description** into one combined field.
- **Applied cleaning function** to:
    - Convert to lowercase
    - Remove punctuation and non-alphabetic characters
    - Tokenize words
    - Remove stopwords
    - Apply stemming (using Porter Stemmer)

Cleaned output was stored in a new column clean_text.

---

5. Feature Engineering with TF-IDF

We used **TF-IDF vectorization** to convert text into numerical features.

vectorizer_news = TfidfVectorizer(max_features=5000)

X_news = vectorizer_news.fit_transform(news['clean_text']).toarray()

This produced a **5000-dimensional vector** for each article. The target variable (category) was **label encoded** to integer values for compatibility with ML models.

---

6. Model Building

We used the following models:

- **Logistic Regression**
- **Multinomial Naive Bayes**

Data was split into training and testing sets (e.g., 80:20).

---

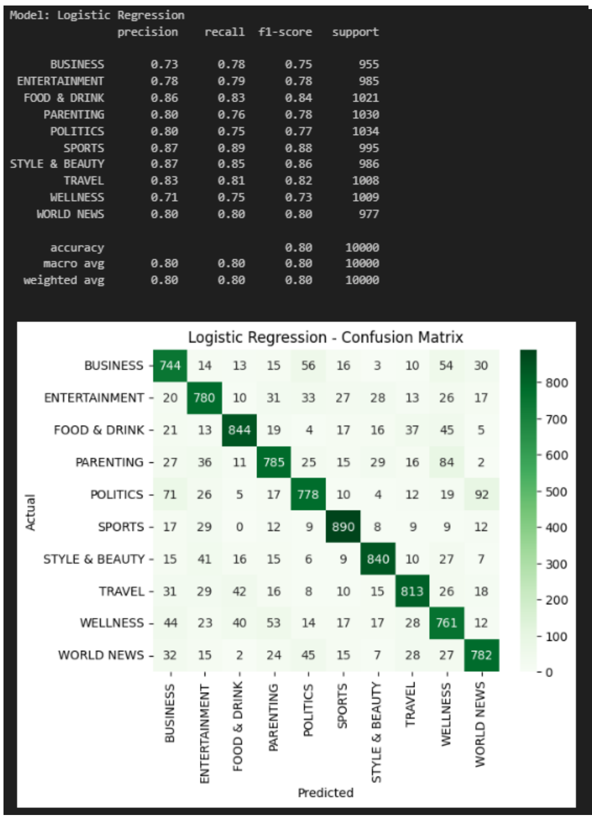
7. Model Evaluation

Logistic Regression

- **Cross-validation accuracy**: ~78.7%
- Good performance across all classes.

Naive Bayes

- **Overall accuracy**: ~78%
- Slightly less precise for some classes, but faster to train.



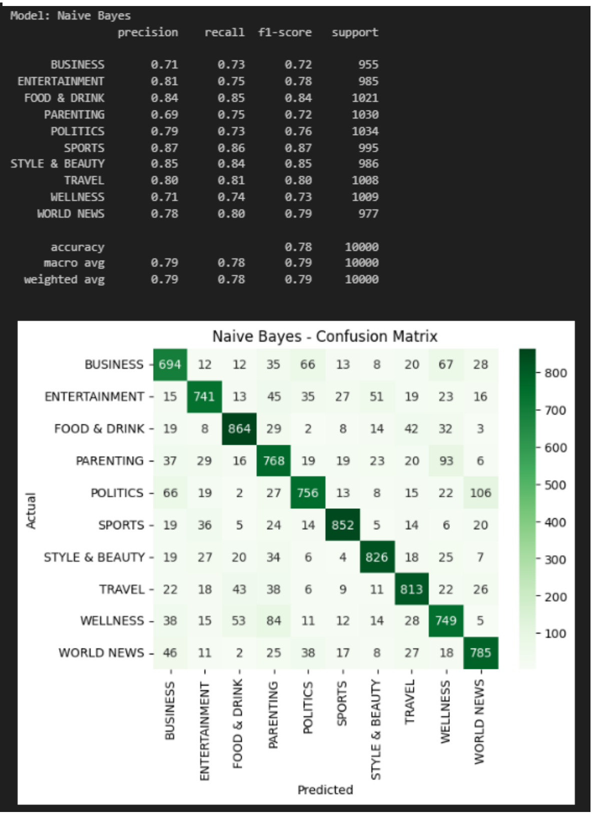
```
Model: Logistic Regression
               precision    recall  f1-score   support

     BUSINESS       0.73      0.78      0.75       955
ENTERTAINMENT       0.78      0.79      0.78       985
  FOOD & DRINK      0.86      0.83      0.84      1021
    PARENTING       0.80      0.76      0.78      1030
     POLITICS       0.80      0.75      0.77      1034
       SPORTS       0.87      0.89      0.88       995
STYLE & BEAUTY      0.87      0.85      0.86       986
       TRAVEL       0.83      0.81      0.82      1008
     WELLNESS       0.71      0.75      0.73      1009
   WORLD NEWS       0.80      0.80      0.80       977

     accuracy                           0.80     10000
    macro avg       0.80      0.80      0.80     10000
 weighted avg       0.80      0.80      0.80     10000
```

Logistic Regression - Confusion Matrix

Actual \ Predicted	BUSINESS	ENTERTAINMENT	FOOD & DRINK	PARENTING	POLITICS	SPORTS	STYLE & BEAUTY	TRAVEL	WELLNESS	WORLD NEWS
BUSINESS	744	14	13	15	56	16	3	10	54	30
ENTERTAINMENT	20	780	10	31	33	27	28	13	26	17
FOOD & DRINK	21	13	844	19	4	17	16	37	45	5
PARENTING	27	36	11	785	25	15	29	16	84	2
POLITICS	71	26	5	17	778	10	4	12	19	92
SPORTS	17	29	0	12	9	890	8	9	9	12
STYLE & BEAUTY	15	41	16	15	6	9	840	10	27	7
TRAVEL	31	29	42	16	8	10	15	813	26	18
WELLNESS	44	23	40	53	14	17	17	28	761	12
WORLD NEWS	32	15	2	24	45	15	7	28	27	782



```
Model: Naive Bayes
               precision    recall  f1-score   support

     BUSINESS       0.71      0.73      0.72       955
ENTERTAINMENT       0.81      0.75      0.78       985
  FOOD & DRINK      0.84      0.85      0.84      1021
    PARENTING       0.69      0.75      0.72      1030
     POLITICS       0.79      0.73      0.76      1034
       SPORTS       0.87      0.86      0.87       995
STYLE & BEAUTY      0.85      0.84      0.85       986
       TRAVEL       0.80      0.81      0.80      1008
     WELLNESS       0.71      0.74      0.73      1009
   WORLD NEWS       0.78      0.80      0.79       977

     accuracy                           0.78     10000
    macro avg       0.79      0.78      0.79     10000
 weighted avg       0.79      0.78      0.79     10000
```

Naive Bayes - Confusion Matrix

Actual \ Predicted	BUSINESS	ENTERTAINMENT	FOOD & DRINK	PARENTING	POLITICS	SPORTS	STYLE & BEAUTY	TRAVEL	WELLNESS	WORLD NEWS
BUSINESS	694	12	12	35	66	13	8	20	67	28
ENTERTAINMENT	15	741	13	45	35	27	51	19	23	16
FOOD & DRINK	19	8	864	29	2	8	14	42	32	3
PARENTING	37	29	16	768	19	19	23	20	93	6
POLITICS	66	19	2	27	756	13	8	15	22	106
SPORTS	19	36	5	24	14	852	5	14	6	20
STYLE & BEAUTY	19	27	20	34	6	4	826	18	25	7
TRAVEL	22	18	43	38	6	9	11	813	22	26
WELLNESS	38	15	53	84	11	12	14	28	749	5
WORLD NEWS	46	11	2	25	38	17	8	27	18	785

8. Visualizations

- **Confusion Matrix** for each model showed areas of misclassification.
- **Category count plot** for training data.
- **Word frequency plots** and top TF-IDF terms could be used for feature importance approximation.

9. Insights and Observations

- TF-IDF vectors are effective for news classification.
- Logistic Regression is robust across all categories.
- Naive Bayes is faster but underperforms in less-distinct categories like "WELLNESS" or "PARENTING".
- Clean text and rich preprocessing significantly boost classification quality.

10. Conclusion

This project demonstrated that traditional ML models like Logistic Regression and Naive Bayes, when applied with TF-IDF features, perform well on multi-class text classification.