News Article Classification Project Report

. Project Overview

In today’s fast-paced digital world, large volumes of news articles are generated and shared across platforms. To improve content organization and user recommendations, it's essential to automatically classify articles into categories like **sports**, **politics**, and **technology**.

This project focuses on building a **machine learning model** that classifies news articles based on their content using **Natural Language Processing (NLP)** and supervised learning techniques.

1. Problem Statement

Develop a robust classifier that:

- Automatically categorizes news articles into predefined categories.
- Efficiently handles text preprocessing and feature extraction.
- Evaluates and compares multiple classification models.
- Provides accurate predictions for unseen articles.

2 data analysis

The dataset contains 50,000 news articles evenly distributed across 10 categories, each with 5,000 samples. The short_description column has a negligible number of missing values (only 6), ensuring overall data quality is high.

```
(50000, 3)
Index(['category', 'headline', 'short_description'], dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   category           50000 non-null  object
 1   headline           50000 non-null  object
 2   short_description  49994 non-null  object
dtypes: object(3)
memory usage: 1.1+ MB
```

category	count
WELLNESS	5000
POLITICS	5000
ENTERTAINMENT	5000
TRAVEL	5000
STYLE & BEAUTY	5000
PARENTING	5000
FOOD & DRINK	5000
WORLD NEWS	5000
BUSINESS	5000

3. Dataset Information

The dataset used data_news. consists of the following columns:

- category – target label
- headline – article title
- short_description – brief content
- keywords – metadata (not used in modeling)

I combined headline and short_description to create a new column called text, which was used for training.

	category	text	clean_text
0	WELLNESS	143 Miles in 35 Days: Lessons Learned Resting ...	mile day lesson learned resting part training ...
1	WELLNESS	Talking to Yourself: Crazy or Crazy Helpful? T...	talking crazy crazy helpful think talking tool...
2	WELLNESS	Crenezumab: Trial Will Gauge Whether Alzheimer...	crenezumab trial gauge whether alzheimers drug...
3	WELLNESS	Oh, What a Difference She Made If you want to ...	oh difference made want busy keep trying perfe...
4	WELLNESS	Green Superfoods First, the bad news: Soda bre...	green superfoods first bad news soda bread cor...

□ 4. Data Preprocessing

Text data was cleaned and transformed using the following steps:

- Converted to lowercase
- Removed punctuation and special characters
- Removed stopwords (using NLTK)
- Tokenized and lemmatized words
- Created a new column clean_text
-

5. Exploratory Data Analysis (EDA)

- The dataset had multiple categories like **politics**, **entertainment**, **wellness**, etc.
- EDA revealed **imbalanced category distribution**, with some classes having significantly more articles than others.
- Visualized using bar plots and pie charts.

---

6. Feature Extraction

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

python

```
tfidf = TfidfVectorizer(max_features=5000)

X = tfidf.fit_transform(df['clean_text']).toarray()

y = df['category']
```

---

7. Model Development

Three supervised models were trained and evaluated:

1. **Logistic Regression**

   A linear model that estimates probabilities using the logistic function.
   It’s simple, fast, and effective for high-dimensional text data like TF-IDF features.

```
Classification Report:
               precision    recall  f1-score   support

     BUSINESS       0.73      0.78      0.76       955
ENTERTAINMENT       0.77      0.78      0.77       985
 FOOD & DRINK       0.86      0.82      0.84      1021
    PARENTING       0.77      0.76      0.77      1030
     POLITICS       0.80      0.74      0.77      1034
       SPORTS       0.87      0.89      0.88       995
STYLE & BEAUTY      0.86      0.85      0.85       986
       TRAVEL       0.83      0.81      0.82      1008
     WELLNESS       0.72      0.75      0.74      1009
   WORLD NEWS       0.79      0.81      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

Confusion Matrix:
 [[748  19  12  20  52  14   2   9  44  35]
 [ 22 765  14  35  31  26  32  15  28  17]
 [ 19  12 838  19   5  18  19  37  45   9]
 [ 27  36  10 781  29  15  26  16  85   5]
 [ 77  26   2  20 767  11   5  14  18  94]
 [ 16  28   3  16  11 883  10   7   8  13]
 [ 17  43  15  21   5   6 839  10  24   6]
 [ 25  29  38  21   6  11  18 814  23  23]
```

2. **Naive Bayes (Multinomial)**

   A probabilistic classifier based on Bayes' theorem with a strong (naive) independence assumption.
   It performs well on text classification due to its efficiency and simplicity, especially with sparse data

```
◆ Naive Bayes Results
Accuracy: 0.7822
Classification Report:
                  precision    recall  f1-score   support

      BUSINESS       0.71      0.74      0.72       955
 ENTERTAINMENT       0.79      0.74      0.76       985
  FOOD & DRINK       0.82      0.85      0.84      1021
     PARENTING       0.69      0.74      0.71      1030
      POLITICS       0.79      0.73      0.76      1034
        SPORTS       0.87      0.86      0.86       995
STYLE & BEAUTY       0.85      0.84      0.84       986
        TRAVEL       0.79      0.81      0.80      1008
      WELLNESS       0.72      0.72      0.72      1009
    WORLD NEWS       0.80      0.81      0.80       977

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

3. **Support Vector Machine (LinearSVC)**

Train-test split (80-20) was applied for model evaluation.

```
◆ SVM Results
Accuracy: 0.7892
Classification Report:
                  precision    recall  f1-score   support

      BUSINESS       0.73      0.80      0.77       955
 ENTERTAINMENT       0.78      0.76      0.77       985
  FOOD & DRINK       0.83      0.83      0.83      1021
     PARENTING       0.76      0.75      0.76      1030
      POLITICS       0.78      0.72      0.75      1034
        SPORTS       0.87      0.91      0.89       995
STYLE & BEAUTY       0.84      0.85      0.84       986
        TRAVEL       0.81      0.78      0.79      1008
      WELLNESS       0.72      0.71      0.71      1009
    WORLD NEWS       0.77      0.79      0.78       977

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

---

8. Model Evaluation

Each model was evaluated using:

- Accuracy
- Classification Report (Precision, Recall, F1-score)
- Confusion Matrix

◈ **Logistic Regression**

Accuracy: 80%

Well-balanced performance across all classes.

◈ **Naive Bayes:**

Accuracy: 78.5%

Faster but slightly less accurate; struggles with complex sentences.

◈ **SVM:**

Accuracy: 78%

Best overall performance, especially on smaller classes.

---

9. Cross-Validation

Cross-validation (k=5) was used for robust evaluation of Logistic Regression:

This ensured the model's generalization ability across folds.

---

10. Final Model Selection

Based on accuracy and F1-score:

Model	**Accuracy**	**F1-Score**	**Notes**
Logistic Regression	80 %	Good	Balanced and fast
Naive Bayes	78.5%	Excellent	Best over all accuracy
SVM	79%	good	Balanced and fast

SVM was selected as the best model for deployment.

Compare the performance of different models and select the best one for

classification

Logistic Regression is selected as the best model for this news classification task because:

- **It achieved the highest accuracy (80%) among all models.**
- **It offered a good balance between training time and performance.**
- **It handled multi-class text classification efficiently using TF-IDF features.**

11. Conclusion

This project successfully demonstrates the use of **machine learning and NLP** techniques for automated news classification. It highlights the importance of preprocessing, feature extraction, model comparison, and evaluation

Video explanation
https://drive.google.com/file/d/1Ylp-b9yp_HS47smuJ5stPWpVqKZ9oQcS/view?usp=sharing

https://drive.google.com/file/d/1Ylp-b9yp_HS47smuJ5stPWpVqKZ9oQcS/view?usp=drive_link