News Article Classification Project: Final Report

1. Introduction

This project focuses on developing a machine learning model to classify news articles into predefined categories based on textual content. The goal is to build an effective classification system that can help news platforms organize and recommend articles efficiently.

2. Dataset Overview

- **Dataset Name:** data_news.csv
- **Columns:**
    - category (Target variable) – News article category
    - headline – Title of the article
    - links – URL to the full article
    - short_description – Brief summary of the article
    - keywords – Key terms related to the article

3. Data Preprocessing & Exploration

3.1 Data Cleaning

- Removed duplicate entries.
- Checked for missing values and handled them appropriately.
- Removed unnecessary columns (e.g., links) that do not contribute to classification.

3.2 Exploratory Data Analysis (EDA)

- **Class Distribution:** The dataset is fairly balanced, with minor discrepancies:

    - **Underrepresented categories:** Parenting, Business (~9.4%).
    - **Overrepresented categories:** Travel, Food & Drink, Entertainment (~10.3%).
- **Text Analysis:**
    - Average article length varies across categories.
    - Word cloud analysis revealed common words in each category.

4. Feature Engineering

- **Text Vectorization:**
    - Implemented **TF-IDF** for better text representation.
    - Experimented with **Word2Vec and BERT embeddings** for capturing word context.
- **Custom Stopword Removal:**
    - Removed words that caused misclassification (e.g., generic words appearing across multiple categories).

5. Model Development & Training

5.1 Baseline Models

The following models were trained on the dataset:

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**
SVM	**70.5**	0.71	0.70	**0.71**
Logistic Regression	70.4	0.71	0.70	0.70
Naïve Bayes	69.4	0.70	0.69	0.70

- SVM performed the best (70.5% accuracy).

- Logistic Regression was close (70.4%).
- Naïve Bayes had the lowest accuracy (69.4%) but showed strong recall for some categories.

5.2 Hyperparameter Tuning

To improve model performance, hyperparameter tuning was performed using **GridSearchCV**:

- **Logistic Regression:** Tuned C (regularization strength) and solver (liblinear vs. saga).
- **Naïve Bayes:** Tuned alpha (smoothing parameter) for better probability estimates.
- **SVM:** Tuned C and kernel type (linear, rbf, poly).

Best hyperparameters found:

- **SVM:** C=1.0, kernel='linear'
- **Logistic Regression:** C=0.8, solver='liblinear'
- **Naïve Bayes:** alpha=0.1

5.3 Cross-Validation Results

To ensure model generalization, **cross-validation** was performed using **5-fold CV**:

Model	**Cross-Validation Accuracy (%)**
Logistic Regression	67.0
Naïve Bayes	67.0
SVM	**68.5**

SVM showed the best cross-validation accuracy (68.5%), confirming its strong generalization ability.

6. Error Analysis & Challenges

- **Misclassified categories:**
  - **Parenting & Business:** Most frequently misclassified.
  - **Politics & World News:** Overlapping content caused confusion.
- **Precision-Recall Tradeoff:**
  - **Food & Drink and Sports had the highest performance.**
  - **Parenting and Travel had the lowest recall, leading to misclassifications.**

7. Recommendations for Improvement

✅ **Feature Engineering:**

- Use **BERT embeddings** to improve context understanding.
- Implement **n-grams** and **domain-specific keywords** for better representation.

✅ **Model Optimization:**

- Fine-tune **SVM hyperparameters** for better generalization.
- Use **ensemble learning (Voting Classifier: SVM + Logistic Regression)** to improve performance.
- Experiment with **deep learning models (LSTM, BERT)** for complex text structures.

✅ **Addressing Misclassifications:**

- **Data Augmentation:** Generate synthetic data using **paraphrasing & back-translation**.
- **Error Analysis:** Investigate misclassified examples to refine preprocessing.

8. Conclusion & Next Steps

This project successfully developed a classification model with **70.5% accuracy using SVM**. Future improvements could include deep learning models and additional feature engineering to enhance classification accuracy.

Next Steps:

- Implement **BERT-based models** for better text understanding.
- Improve **category-specific handling** with advanced preprocessing.
- Deploy the best-performing model as a web application for practical use.