Name: sushant tyagi

Roll no: 2301201099

Sec: B

BBC News Article Classification Project Report

1. Introduction

Text classification is a fundamental task in Natural Language Processing (NLP) that involves categorizing textual data into predefined classes. The BBC News dataset contains news articles from five categories: **business, entertainment, politics, sport, and tech**.

This project aims to develop a **machine learning-based news classifier** that can predict the category of a news article with high accuracy using various feature extraction techniques and models.

---

2. Dataset Description

- **Source:** BBC News Dataset
- **Total Articles:** 2225
- **Categories:** business, entertainment, politics, sport, tech
- **Articles per Category:**
  - Business: 510
  - Entertainment: 386
  - Politics: 417
  - Sport: 511
  - Tech: 401

The dataset is pre-divided into folders corresponding to each category. Each folder contains multiple .txt files representing individual news articles.

---

3. Data Preprocessing

To ensure that the text is in a suitable format for machine learning, the following preprocessing steps were applied:

1. **Lowercasing:** Convert all text to lowercase.
2. **Punctuation Removal:** Remove all non-alphabetic characters.
3. **Stopwords Removal:** Remove common English stopwords using NLTK.
4. **Stemming/Lemmatization:**
   - Stemming: Reduce words to their root form using Porter Stemmer.
   - Lemmatization: Convert words to base form using WordNet Lemmatizer.

Example:

Original Text	**Stemmed Text**	**Lemmatized Text**
"The stock market fell sharply today."	"stock market fell sharp today"	"stock market fell sharply today"

---

4. Feature Extraction

Two types of numerical representations were created from the text:

1. **Bag-of-Words (BoW)**
   - Converts text into a sparse matrix of token counts.
   - Used unigrams and bigrams (ngram_range=(1,2)) with maximum 5000 features.
2. **TF-IDF (Term Frequency-Inverse Document Frequency)**
   - Captures the importance of words relative to the dataset.
   - Same n-grams and feature limit as BoW.

Both representations were used to train classifiers.

---

5. Model Training

Four classifiers were trained using different combinations of features:

Model	**Feature**	**Accuracy**
Logistic Regression	BoW	0.9685
Logistic Regression	TF-IDF	0.9742
SVM (Linear)	BoW	0.9753
SVM (Linear)	TF-IDF	**0.9810**

- **Training Data:** 80%
- **Testing Data:** 20% (stratified by category)

Observation: Linear SVM with TF-IDF features performed best with an accuracy of 98.1%.

---

6. Evaluation

6.1 Classification Report for Best Model (SVM + TF-IDF)

Category	**Precision**	**Recall**	**F1-Score**
Business	0.98	0.99	0.98

Category	**Precision**	**Recall**	**F1-Score**
Entertainment	0.98	0.96	0.97
Politics	0.98	0.98	0.98
Sport	0.99	0.99	0.99
Tech	0.97	0.97	0.97
Average	**0.98**	**0.98**	**0.98**

6.2 Category-wise Accuracy

- Highest accuracy: Sport (0.99)
- Lowest accuracy: Entertainment (0.96)

6.3 Visualizations

1. **Accuracy Comparison of Models**
   - TF-IDF outperforms BoW across both Logistic Regression and SVM.
2. **Category Distribution**
   - Pie charts showing true vs predicted categories indicate excellent balance and low misclassification.
3. **Category-wise Accuracy**
   - Bar chart shows consistent high accuracy across all categories.

---

7. Conclusion

- A **news classification system** was successfully implemented using the BBC News dataset.
- **TF-IDF representation combined with Linear SVM** provided the best performance with 98.1% accuracy.
- Preprocessing steps such as stopword removal and stemming significantly improved classifier performance.
- The project demonstrates the effectiveness of traditional machine learning techniques for text classification.

Artifacts Generated:

- accuracy_comparison.png (model performance)
- category_distribution.png (category-wise evaluation)
- news_classifier.py (main script)

---

8. Future Work

- Implement **hyperparameter tuning** for SVM and Logistic Regression.
- Explore **deep learning approaches** like LSTM or BERT for further improvement.
- Deploy the model using a web interface or API for real-time news classification.