News Article Classification using Machine Learning and Natural Language Processing Techniques

Abstract

This project presents a text classification model designed to categorize news articles from the BBC News Summary dataset into five distinct categories: "business," "sport," "tech," "politics," and "entertainment." Employed advanced natural language processing (NLP) techniques and machine learning algorithms to achieve high-accuracy classification.

Methodology encompassed comprehensive text preprocessing through lowercasing, removal of numbers and special characters, elimination of stop words, and lemmatization to ensure uniform text representation. The cleaned text was then transformed into numerical features using TF-IDF vectorization with cosine similarity metrics to quantify text relationships.

Trained and evaluated multiple classification models including Logistic Regression, Support Vector Machine (SVM), and Linear SVC. These models were rigorously evaluated using accuracy, precision, recall, F1 score, confusion matrix visualizations, and ROC curves. Our best-performing model achieved 96.85% accuracy with consistent performance across all categories.

The project demonstrates a practical and effective approach to text classification, offering insights into model selection and feature engineering techniques for news article categorization. Future refinements could include enhanced handling of dataset imbalance and exploration of hybrid models to further improve classification consistency.

Keywords: TF-IDF, Logistic Regression, SVM, Linear SVC, Hyperparameter Tuning, Text Classification

1. Introduction

1.1 Objectives

This project aims to build an efficient text classification model that categorizes news articles from the BBC News Summary dataset into five distinct categories: "business," "sport," "tech," "politics," and "entertainment." The project leverages various natural language processing (NLP) techniques and machine learning models to achieve this goal.

1.2 Data Description

Categories

The dataset consists of 2,225 news articles, categorized into five distinct categories:

1. **Business:** Articles covering topics such as company profits, economic policies, market trends, and corporate actions.
2. **Entertainment:** Topics including movies, music, celebrity news, and cultural events.
3. **Politics:** Articles focusing on political events, elections, and governmental policies.
4. **Sport:** News related to various sports, matches, athletic events, and athletes.
5. **Tech:** Topics such as technological advancements, cybersecurity, online gaming, and industry trends.

Structure

Each article is represented by three key attributes:

1. **Category:** The category to which the article belongs, indicating its thematic classification.
2. **Filename:** The unique identifier for each article's file, serving as a reference to its origin.
3. **Content:** The full-text content of the article provides the basis for NLP and machine learning-based classification.

Dataset Source: BBC News Summary on Kaggle

1.3 Problem Statement

The primary problem addressed by this project is to classify news articles into one of the predefined categories based on their textual content. This involves:

1. Converting textual data into numerical features suitable for machine learning models
2. Training and evaluating different classification algorithms
3. Optimizing model performance through hyperparameter tuning
4. Creating a deployment-ready solution for classifying new articles

1.4 Desired Outputs

The project aims to deliver:

1. A preprocessed dataset with cleaned text ready for modeling
2. Visualizations showing article distribution and characteristics

3. Multiple trained classification models with performance evaluations
4. A final high-accuracy model capable of categorizing new articles
5. Comprehensive evaluation metrics including accuracy, precision, recall, F1 scores, and visualizations
6. A deployment-ready solution with saved models and vectorizers

2. Methodology

2.1 Data Collection and Preprocessing

Data Collection

The dataset consists of news articles from the BBC News Summary dataset, stored in a zip file. The articles cover various categories including business, sports, tech, politics, and entertainment.

Data Preprocessing

To prepare the dataset for analysis and classification, we applied several preprocessing steps:

1. **File Reading:** The zip file was opened, and each article was read individually.
2. **Text Cleaning:** The articles were then cleaned by:
    - Converting text to lowercase
    - Removing numbers and special characters
    - Eliminating stop words using NLTK
    - Lemmatizing words to their base forms using WordNet
3. **Creating a DataFrame:** The cleaned articles and their categories were stored in a structured format for further processing.

2.2 Feature Extraction

After preprocessing, we converted the text data into numerical features:

1. **TF-IDF Vectorization:** We used Term Frequency-Inverse Document Frequency (TF-IDF) to transform the cleaned text into numerical features, assigning weights to words based on their frequency and importance. The TF-IDF value is calculated as:
2. TF-IDF(t, d) = TF(t, d) × IDF(t)

Where:

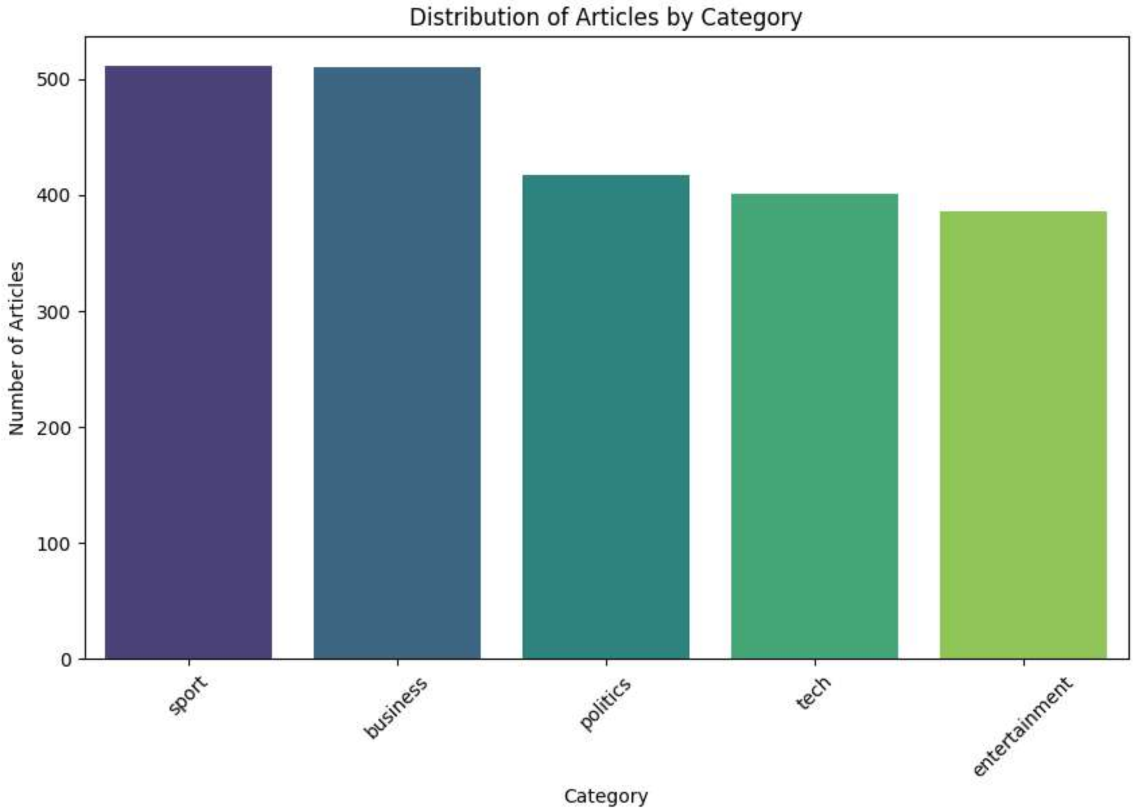
- TF(t, d) is the term frequency of term t in document d
- IDF(t) is the inverse document frequency of term t, given by log(N/DF(t)), with N being the total number of documents and DF(t) being the number of documents containing term t

3. **Advanced Vectorization:** We implemented customized parameters like max_features and ngram_range in the TF-IDF vectorizer, tailoring the vectorization process to the dataset's characteristics.
4. **Cosine Similarity:** This metric was used to assess the similarity between articles by comparing their vectorized forms, helping to gauge text content relationships.

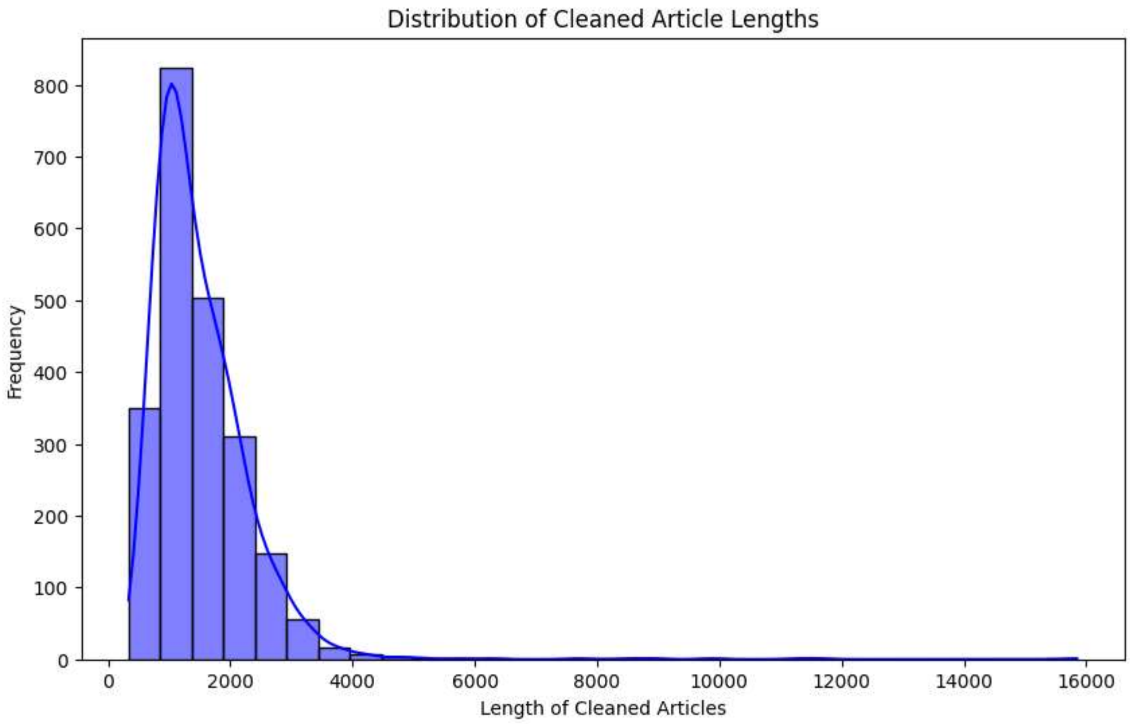
2.3 Exploratory Data Analysis

To gain insights into the dataset, we created the following visualizations:

1. **Bar Plots:** To visualize the distribution of articles across different categories



2. **Histograms:** To show the distribution of article lengths in each category



2.4 Model Training

We split the dataset into training and test sets and trained several classification models:

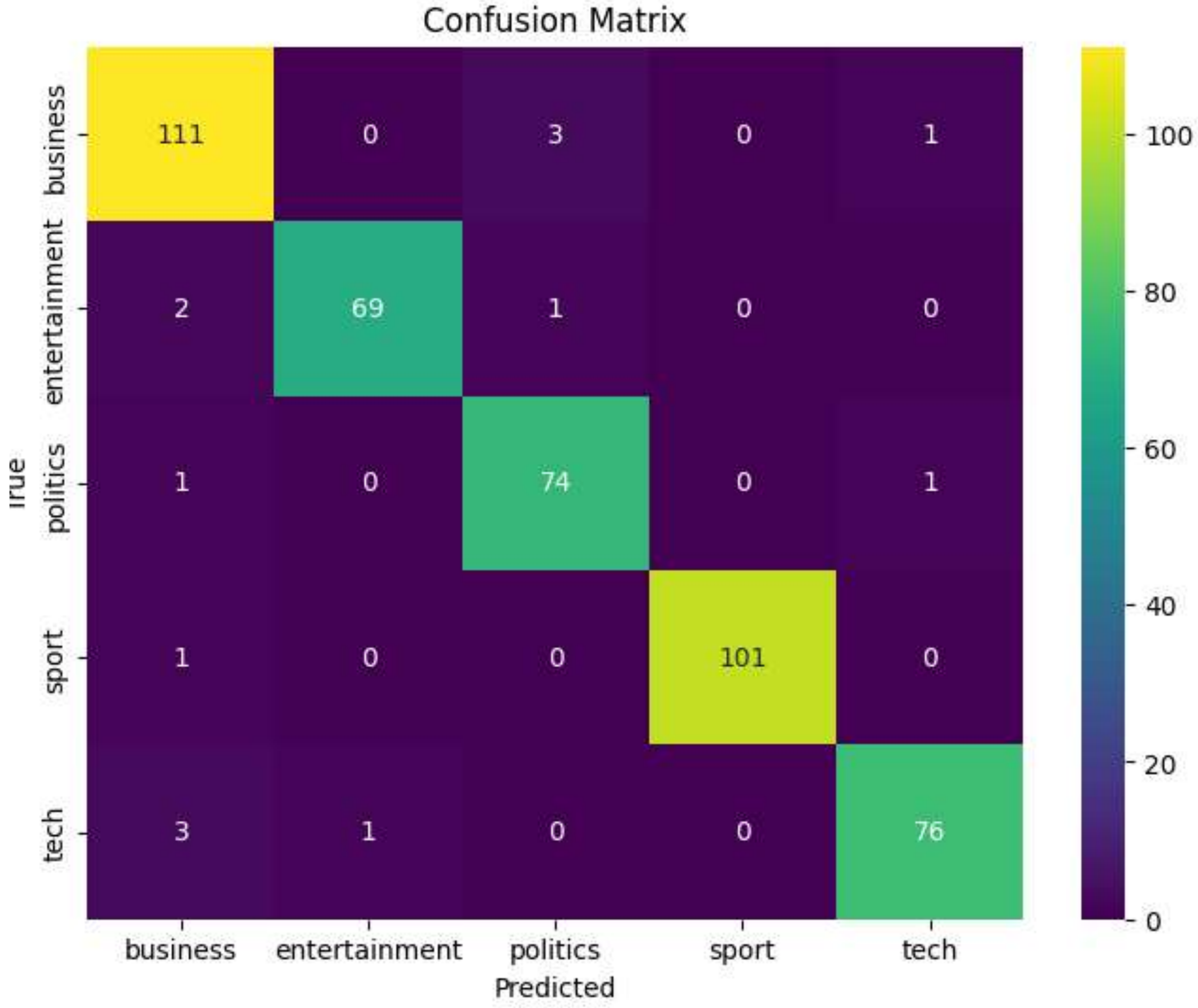
1. **Logistic Regression:** A statistical method used for classification tasks, modeling the relationship between a dependent variable and independent variables. The probability of a class Y given features X is modeled as:
2. **Support Vector Machine (SVM):** A supervised learning algorithm that finds the optimal hyperplane to separate classes in the feature space, maximizing the margin between classes. The decision function is given by:
3. **Linear SVC:** A variant of SVM that uses a linear kernel for efficient classification, particularly suitable for high-dimensional data with linearly separable classes. The decision function is given by:

2.5 Model Evaluation

To assess the models' performance, we used several metrics:

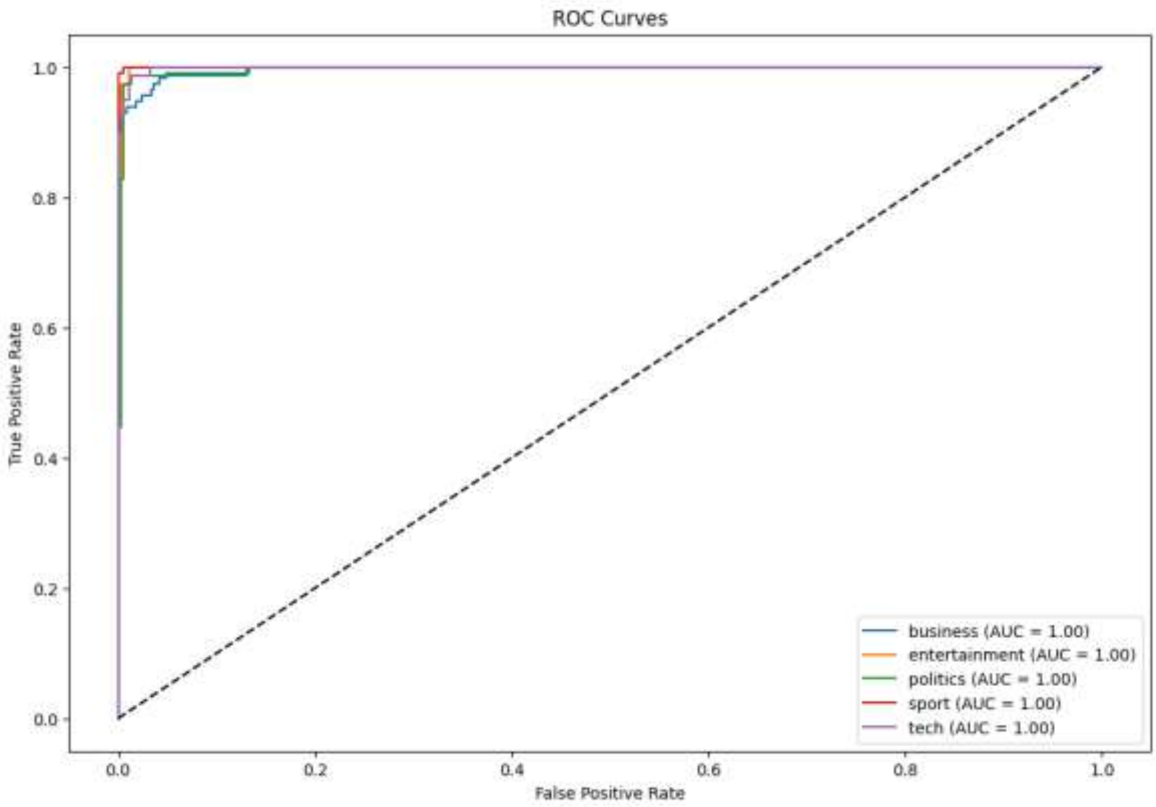
1. **Accuracy:** The ratio of correctly predicted observations to the total observations
2. **Precision:** The ratio of correctly predicted positive observations to the total predicted positives
3. **Recall:** The ratio of correctly predicted positive observations to all actual positives
4. **F1 Score:** The weighted average of Precision and Recall

5. **Confusion Matrix:** Visualizing how well each model performed in classifying the articles



True \ Predicted	business	entertainment	politics	sport	tech
business	111	0	3	0	1
entertainment	2	69	1	0	0
politics	1	0	74	0	1
sport	1	0	0	101	0
tech	3	1	0	0	76

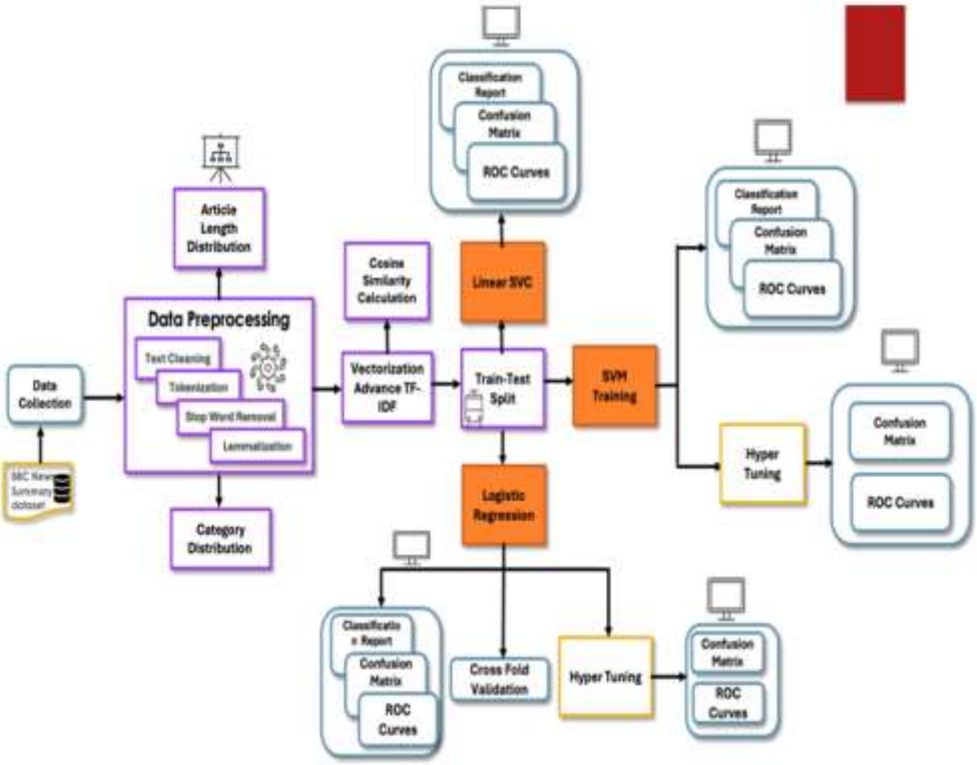
6. **ROC Curves:** Plotting the true positive rate against the false positive rate at various threshold settings



2.6 Model Optimization

To optimize model performance, we took the following steps:

1. **Hyperparameter Tuning:** We applied Grid Search to Logistic Regression and SVM models to optimize parameters such as C value, solver type, kernel, and gamma.
2. **Cross-Validation:** K-fold cross-validation was conducted to evaluate model consistency across multiple folds.
3. **Feature Selection:** Various feature selection methods were explored to identify the most relevant features for classification.



3. Experimental Results

3.1 Distribution of Articles by Category

We analyzed the distribution of articles across the five categories (sport, business, politics, tech, and entertainment) to understand the dataset composition. This analysis provided insights into potential class imbalances that might affect model performance.

3.2 Logistic Regression Results

Before Tuning

- **Accuracy:** 96.63%
- **Precision:** 96.68%
- **Recall:** 96.63%
- **F1 Score:** 96.64%

The confusion matrix showed some misclassifications, such as 4 instances of business articles being classified as politics.

After Hyperparameter Tuning

- **Accuracy:** 96.85%

- **Precision:** Improved across categories
- **Recall:** Consistent high values
- **F1 Score:** Enhanced overall

The confusion matrix reflected fewer misclassifications across all categories. For instance, only 3 business articles were misclassified as politics, demonstrating better performance.

The ROC curves indicated that the AUC for each category remained at or near 1.00, showing excellent distinction between classes. The classification report showed high precision, recall, and F1 scores for all categories, demonstrating the model's reliable performance after tuning.

3.3 SVM Model Results

Before Tuning

- **Accuracy:** 96.18%
- **Macro and Weighted Averages:** ~0.96 for precision, recall, and F1 scores

The confusion matrix revealed minor misclassifications, with some business articles classified as politics, but generally reliable distinction between categories.

After Hyperparameter Tuning

- **Accuracy:** 96.18% (remained steady)
- **Consistency:** Maintained 0.96 for precision, recall, and F1 scores across all categories

The ROC curves showed consistent AUC values close to 1.00 for all categories, demonstrating the model's strong ability to distinguish between them.

3.4 Linear SVC Results

- **Overall Accuracy:** 96.4%
- **Category Performance:** The model showed particular strengths in classifying "sport," "entertainment," and "tech" categories
- **Top Features:** Key terms associated with each category (e.g., "bank" for business, "technology" for tech) were effectively leveraged by the model
- **Metric Consistency:** Precision, recall, and F1 scores remained consistent across all categories

3.5 Model Comparison

Model	Accuracy	Key Strengths
Logistic Regression (Tuned)	96.85%	Highest accuracy, well-balanced performance
SVM (Tuned)	96.18%	Stable performance, good generalization
Linear SVC	96.4%	Fast training, effective on high-dimensional data

Based on these results, the tuned Logistic Regression model was selected as the best-performing model for deployment.

4. Analysis and Discussion

4.1 Strengths

The classification methodology demonstrated several key strengths:

1. **Robust Feature Identification:** The TF-IDF vectorization effectively transformed textual content into numerical features, allowing models to accurately classify articles into their respective categories.
2. **High Accuracy:** All models achieved accuracy above 96%, with the Logistic Regression model reaching 96.85% after tuning, demonstrating the effectiveness of our approach.
3. **Consistent Performance:** The models showed balanced performance across different categories, as evidenced by high precision, recall, and F1 scores.

4.2 Limitations

The methodology also presented certain limitations:

1. **Handling Nuanced Content:** Articles that span multiple categories challenged the models' ability to classify accurately. For example, articles discussing the impact of technology on businesses might blur the lines between tech and business categories.
2. **Class Imbalance:** The performance varied slightly across different categories due to imbalances in the dataset, potentially affecting the consistency of classification results for categories with fewer articles.

4.3 Future Refinements

To address these limitations, several refinements could be explored:

1. **Advanced Feature Engineering:** Exploring additional feature engineering techniques to better capture content nuances, particularly for articles spanning multiple categories.

2. **Dataset Balancing:** Implementing techniques to balance the dataset and mitigate class imbalance issues, ensuring consistent model performance across all categories.
3. **Hybrid Models:** Considering hybrid models that integrate domain knowledge to enhance the ability to handle nuanced content more effectively.
4. **Deep Learning Approaches:** Exploring deep learning models such as LSTM, BERT, or other transformer-based architectures that might capture complex semantic relationships better than traditional machine learning approaches.
5. **Deployment Solutions:** Developing a web application or API for real-time classification of news articles, potentially integrating with news aggregation platforms.

5. Conclusion

This project successfully implemented and evaluated multiple machine learning models for classifying news articles into five distinct categories. Through comprehensive preprocessing, feature extraction, and model optimization, we achieved high classification accuracy across all approaches.

The Logistic Regression model, after hyperparameter tuning, emerged as the best performer with 96.85% accuracy and consistent performance across all categories. The model effectively leveraged TF-IDF vectorization to capture the semantic nuances of news articles and accurately categorize them.

The project demonstrates a practical approach to text classification that can be applied to various content categorization challenges. While some limitations exist, particularly in handling nuanced content and addressing class imbalance, the proposed refinements offer pathways for further improvement.

Overall, this work contributes to the field of automated content categorization, providing insights into effective techniques for news article classification and offering a deployment-ready solution for practical applications.

6. References

1. BBC News Summary Dataset. Retrieved from https://www.kaggle.com/datasets/pariza/bbc-news-summary
2. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
3. Natural Language Toolkit (NLTK), Bird, S., Klein, E., & Loper, E. (2009).

4. TF-IDF Vectorization for Text Classification, Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.