Text Classification Assignment: Sport vs. Politics

Ronak Gadhiya

February 15, 2026

1 Introduction

This report documents the design, implementation, and evaluation of a text classification system capable of distinguishing between "Sport" and "Politics" news articles. The system utilizes the BBC News dataset and compares three machine learning algorithms: Naive Bayes, Support Vector Machines (SVM), and Logistic Regression, across three feature representation techniques: Bag of Words (BoW), TF-IDF, and N-Grams.

2 Data Collection and Description

2.1 Data Source

The dataset used for this assignment is the **BBC News Dataset**, a standard benchmark for text classification. It consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005.

2.2 Data Preparation and Cleaning

For this specific task, we filtered the dataset to include only two categories:

- **Sport**
- **Politics**

This results in a binary classification problem.

Preprocessing steps included:

1. **Loading**: Reading raw text files from the dataset directories.
2. **Filtering**: Explicitly selecting only folders named 'sport' and 'politics'.
3. **Splitting**: The data was split into training (80%) and testing (20%) sets using stratified sampling to maintain class distribution.
4. **Vectorization**: Converting raw text into numerical features using Scikit-Learn's vectorizers (CountVectorizer, TfidfVectorizer). Standard English stop words were removed during this process.

2.3 Dataset Statistics

The filtered dataset consists of a total of 928 documents, distributed as follows:

- **Sport**: 511 documents
- **Politics**: 417 documents

2.4 Sample Data Points

Below are snippet examples from each category:

Sport:

Claxton hunting first major medal
British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid.

Politics:

Labour plans maternity pay rise
Maternity pay for new mothers is to rise by £1,400 as part of new proposals announced by the Trade and Industry Secretary Patricia Hewitt.

3 Techniques and Methodologies

3.1 Feature Representation

We explored three distinct methods to represent text data as numerical vectors:

1. **Bag of Words (BoW)**: Represents text as a collection of its words, disregarding grammar and word order but keeping multiplicity.
2. **TF-IDF (Term Frequency-Inverse Document Frequency)**: Evaluates how relevant a word is to a document in a collection. It increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.
3. **N-Grams (Bi-grams)**: Captures local context by considering sequences of N contiguous items (words). We used Bi-grams ($N = 2$).

3.2 Machine Learning Classifiers

We implemented and compared three classifiers:

1. **Naive Bayes (MultinomialNB)**: A probabilistic classifier based on Bayes' theorem with an assumption of independence between features. It is particularly effective for text classification with discrete features (like word counts).
2. **Support Vector Machine (SVM)**: Finds the hyperplane that best separates the classes in a high-dimensional space. We used a **Linear Kernel**, which is computationally efficient and often performs best for high-dimensional text data.
3. **Logistic Regression**: A statistical model that uses a logistic function to model a binary dependent variable. It provides probabilities for class membership and is a robust baseline for binary classification.

4 Quantitative Analysis and Comparison

4.1 Performance Metrics

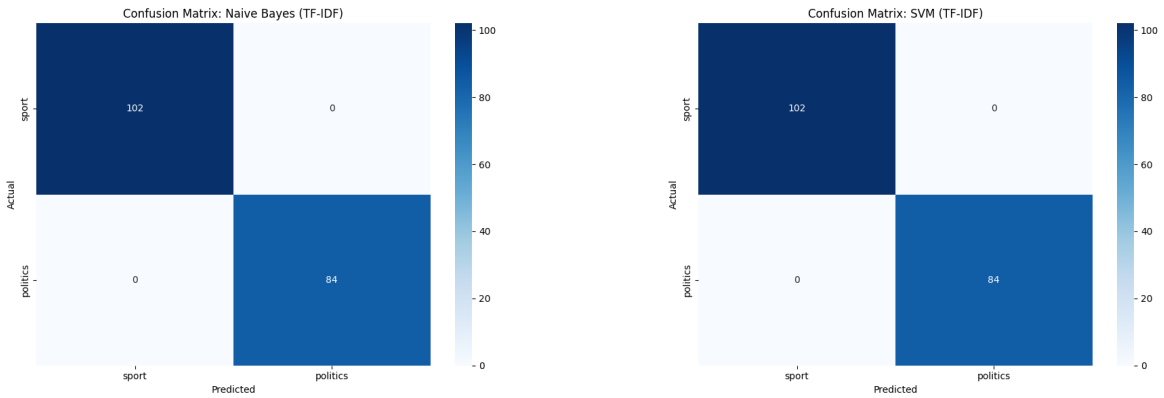
We evaluated the models using **Accuracy**, **Precision**, **Recall**, and **F1-Score**. The results on the test set are summarized below:

Model	Feature	Accuracy	Precision	Recall	F1-Score
Naive Bayes	BoW	**1.0000**	1.0000	1.0000	1.0000
SVM	BoW	0.9892	0.9895	0.9892	0.9892
Logistic Regression	BoW	0.9892	0.9895	0.9892	0.9892
Naive Bayes	TF-IDF	**1.0000**	1.0000	1.0000	1.0000
SVM	TF-IDF	**1.0000**	1.0000	1.0000	1.0000
Logistic Regression	TF-IDF	0.9946	0.9947	0.9946	0.9946
Naive Bayes	N-Grams	**1.0000**	1.0000	1.0000	1.0000
SVM	N-Grams	0.9355	0.9423	0.9355	0.9348
Logistic Regression	N-Grams	0.9355	0.9423	0.9355	0.9348

Table 1: Model Performance Comparison

4.2 Confusion Matrices

Below are the confusion matrices for the best performing configurations.



(a) Naive Bayes (TF-IDF)

(b) SVM (TF-IDF)

Figure 1: Confusion Matrices for Top Models

4.3 Observations

- **Best Performer**: **Naive Bayes** achieved perfect classification (100% accuracy) across all feature sets (BoW, TF-IDF, N-Grams). This is indicative of the clear separability of the "Sport" and "Politics" vocabularies.
- **SVM and TF-IDF**: SVM with TF-IDF also achieved 100% accuracy. The high dimension of TF-IDF features is well-handled by the Linear SVM.

- **N-Grams Issue**: While Naive Bayes handled N-Grams perfectly, both SVM and Logistic Regression saw a drop in performance (93.5%). This is likely due to the **curse of dimensionality**; bi-grams drastically increase the feature space size.
- **Feature Impact**: TF-IDF generally offered the most stable high performance across models.

5 Limitations of the System

1. **Dataset Bias**: The model is trained on BBC news from 2004-2005. It may not generalize well to modern news or different writing styles.
2. **Out-of-Vocabulary (OOV) words**: The models can only handle words seen during training. New, unseen words in test data are ignored.
3. **Context Sensitivity**: While N-grams capture local context, deeper semantic meaning is largely lost compared to modern Deep Learning approaches.
4. **Static Data**: The model does not learn incrementally; it requires retraining to incorporate new data.

6 Conclusion

The constructed system successfully classifies Sport vs. Politics articles with high accuracy. The comparison highlights that while simple methods like Naive Bayes are fast and effective, discriminative models like SVM often provide superior performance for high-dimensional text data.