Sports vs. Politics: A Text Classification Study

Prasangeet Dongre
Roll No: B23CH1033

February 15, 2026

Abstract

This report details the design and implementation of a text classification system capable of distinguishing between sports-related and politics-related documents. Using a dataset collected from RSS feeds, we explore the efficacy of Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) feature representations across three machine learning algorithms: Naive Bayes, Logistic Regression, and Support Vector Machines (SVM). Our experiments demonstrate that linear models like Logistic Regression and Support Vector Machines (SVM) combined with Bag-of-Words (BoW) features achieve peak performance of 98.67% accuracy, highlighting the distinct lexical properties of the two categories even as the dataset size increases.

Contents

1 Introduction

Text classification is a fundamental task in Natural Language Processing (NLP), with applications ranging from spam detection to sentiment analysis. The objective of this project is to build a binary classifier that categorizes text documents into one of two distinct classes: *Sports* (specifically football/cricket etc.) or *Politics* (elections, policy, government).

Understanding the differences in vocabulary and structure between these domains allows us to effectively deploy machine learning models. This report outlines the data collection process, preprocessing steps, feature extraction techniques, and a comprehensive evaluation of classification performance.

2 Data Collection

To ensure a realistic evaluation, we constructed a dataset from live news sources rather than using pre-packaged corpora. Data was collected using the **RSS (Really Simple Syndication)** protocol, which provides structured XML feeds of updated content.

2.1 Sources

We targeted high-traffic news outlets known for their distinct categorization. The following RSS feeds were used:

- **Sports**: ESPN News, BBC Sport, Sky Sports Football, The Guardian Sport, and Yahoo Sports.
- **Politics**: BBC News Politics, NYT Politics, The Guardian Politics, Politico, and Reuters Politics.

2.2 Extraction Process

A Python script utilizing the `feedparser` and `BeautifulSoup` libraries was developed. For each entry in the RSS feed:

1. The article link was extracted.
2. An HTTP request fetched the full HTML content.
3. `BeautifulSoup` parsed the HTML to extract text within `<p>` tags.
4. Articles shorter than 800 characters were discarded to ensure sufficient content for classification.

3 Dataset Description and Analysis

The final dataset consists of **372 documents**. This larger collection provides a more robust foundation for training and evaluating text classification models.

3.1 Class Distribution

The dataset is imbalanced, with a bias towards sports articles:

- **Sports**: 213 documents (approx. 57%)
- **Politics**: 159 documents (approx. 43%)

To mitigate this during training, we employed stratified sampling for our train-test splits to ensure both sets reflected this distribution.

3.2 Statistical Analysis

- **Average Document Length**: 448 words.
- **Vocabulary Size (BoW)**: 16,635 unique tokens.
- **Vocabulary Size (TF-IDF, including bigrams)**: 50,000 features.

The high vocabulary size relative to the number of documents suggests a sparse feature space, which linear models like SVM and Naive Bayes handle well.

4 Methodology

4.1 Text Preprocessing

Before feature extraction, raw text underwent normalization:

- **Lowercasing**: Converting all text to lowercase to treat "Goal" and "goal" identically.
- **Regex Cleaning**: Removing URLs, special characters, and non-alphabetic tokens.
- **Stopword Removal**: Using NLTK's English stopword list to remove common words (e.g., "the", "and") that carry little semantic meaning.
- **Tokenization**: Splitting text into individual words.

4.2 Feature Representation

We explored two methods to convert text into numerical vectors:

4.2.1 Bag of Words (BoW) - Unigrams

BoW represents text as a vector of word counts. It ignores grammar and word order but captures keyword frequency using single words (unigrams).

- **Pros**: Simple, interpretable.
- **Cons**: Ignores context, weighs frequent common words heavily (if not filtered).
- **Vocabulary**: 13,236 unique unigrams.

4.2.2 TF-IDF with N-grams (1-2 grams)

TF-IDF weighs words by how unique they are to a specific document across the corpus. Critically, we included **n-grams (bigrams)** to capture two-word phrases which are often more discriminative than single words.

- **N-gram Range**: Both unigrams (1-gram) and bigrams (2-grams)

- **Example Bigrams**: "prime minister", "champions league", "election campaign", "goal scored"
- **Pros**: Penalizes common words, highlights distinctive terms, captures multi-word expressions.
- **Cons**: High dimensionality (50,000 features including bigrams).
- **Implementation**: `TfidfVectorizer(ngram_range=(1, 2))`

4.3 Machine Learning Models

We trained three classifiers:

1. **Multinomial Naive Bayes**: A probabilistic classifier based on Bayes' theorem. It assumes independence between features. Excellent baseline for text.
2. **Logistic Regression**: A linear model that estimates probabilities using a logistic function.
3. **Support Vector Machine (SVM)**: Finds the hyperplane that best separates the classes with the maximum margin.

5 Quantitative Comparisions and Results

The dataset was split into training (80%) and testing (20%) sets.

5.1 Accuracy Comparison

Model	**BoW (Unigrams)**	**TF-IDF (1-2 grams)**
Naive Bayes	0.97	0.96
Logistic Regression	**0.99**	0.97
SVM	**0.99**	0.97

Table 1: Model Accuracy: Comparing Unigram vs N-gram Features

5.2 Detailed Performance Metrics

Beyond raw accuracy, we examine precision, recall, and F1-score for each class to understand model behavior comprehensively.

5.2.1 Naive Bayes Performance

Class	**Bag of Words**			**TF-IDF**		
	Precision	Recall	F1	Precision	Recall	F1
Politics	1.00	0.94	0.97	1.00	0.91	0.95
Sports	0.96	1.00	0.98	0.93	1.00	0.97
Accuracy		**0.97**			**0.96**	

Table 2: Naive Bayes Classification Metrics

5.2.2 Logistic Regression Performance

Class	Bag of Words			TF-IDF		
	Precision	Recall	F1	Precision	Recall	F1
Politics	1.00	0.97	0.98	1.00	0.94	0.97
Sports	0.98	1.00	0.99	0.96	1.00	0.98
Accuracy	**0.99**			**0.97**		

Table 3: Logistic Regression Classification Metrics

5.2.3 Support Vector Machine Performance

Class	Bag of Words			TF-IDF		
	Precision	Recall	F1	Precision	Recall	F1
Politics	1.00	0.97	0.98	1.00	0.94	0.97
Sports	0.98	1.00	0.99	0.96	1.00	0.98
Accuracy	**0.99**			**0.97**		

Table 4: SVM Classification Metrics

5.3 Confusion Matrices

Confusion matrices provide a detailed breakdown of correct and incorrect predictions for each class.

5.3.1 Best Models: Logistic Regression & SVM (BoW)

		Predicted	
		Politics	Sports
Actual	Politics	31	1
	Sports	0	43

Table 5: Confusion Matrix: LogReg/SVM + BoW (98.67% Accuracy)

5.3.2 Naive Bayes + BoW

		Predicted	
		Politics	Sports
Actual	Politics	30	2
	Sports	0	43

Table 6: Confusion Matrix: Naive Bayes + BoW (97.33% Accuracy)

5.3.3 SVM + TF-IDF (Best TF-IDF Model)

		Predicted	
		Politics	Sports
Actual	Politics	30	2
	Sports	0	43

Table 7: Confusion Matrix: SVM + TF-IDF (Consistent Performance)

5.4 Analysis of Results

- **High Baseline Performance**: All models achieved over 96% accuracy. This indicates that even with a larger and more diverse dataset of 372 documents, the vocabulary remains highly discriminative.
- **Bag-of-Words Superiority**: Logistic Regression and SVM with BoW both achieved 98.67% accuracy. They misclassified only one politics article as sports, while correctly identifying all 43 sports articles in the test set.
- **Improved TF-IDF Recall**: Unlike the initial small-scale experiment, the larger dataset allowed TF-IDF models to achieve high recall (over 91%) for both classes. This suggests that the previous poor performance was indeed a result of insufficient data for the high-dimensional TF-IDF space.
- **SVM and LogReg Robustness**: Both linear models performed identically on the BoW features, demonstrating their effectiveness for text categorization in this domain.

6 Limitations

Despite the high accuracy, the system has limitations:

- **Dataset Size**: While expanded to 372 documents, the dataset remains relatively specialized. High accuracy may partly stem from the specific RSS feeds selected.
- **Class Imbalance**: Although improved, the dataset still has a 57/43 split between sports and politics.
- **Context Sensitivity**: BoW and TF-IDF ignore word order (mostly). "The team defeated the rival" and "The rival defeated the team" have similar vector representations but opposite meanings.

7 Conclusion

We successfully built a text classifier for Sports vs. Politics. The experiments showed that for distinct topics and small datasets, simple Bag-of-Words representations often outperform more complex TF-IDF schemes.