Sports vs Politics Text Classification Using Machine Learning

Sandesh Suman
Roll Number: M25CSA034
CSL 7640 – Natural Language Understanding

February 14, 2026

1 GitHub Repository

The complete source code, dataset, experimental results, and final report are publicly available on GitHub. The repository contains:

- Dataset (dataset.json)
- Full implementation of all classifiers
- Experimental results
- Final project report (PDF)

Repository Link:

`https://github.com/SandeshSuman123/sports-vs-politics-text-classification`

Abstract

Text classification is a fundamental task in Natural Language Processing (NLP). This problem focuses on building a binary classifier that categorizes text documents into either *Sports* or *Politics*. Multiple feature representations including Bag-of-Words (BoW), TF-IDF, and n-grams were explored. Three machine learning algorithms – Naive Bayes, Logistic Regression, and Support Vector Machine (SVM) were implemented from scratch and compared quantitatively. The experimental results show that Logistic Regression combined with TF-IDF achieved the best performance with 95% accuracy.

2 Introduction

Automatic text classification plays an important role in organizing and managing large collections of digital documents. News platforms, recommendation systems, and content moderation systems rely heavily on automated categorization.

In this project, we design a binary classifier that determines whether a given text document belongs to the Sports domain or the Politics domain. The goal is to explore classical machine learning techniques and compare their performance using different feature representations.

The objectives of this study are:

- To construct a balanced dataset for Sports and Politics categories.
- To experiment with Bag-of-Words, TF-IDF, and n-gram features.
- To implement and compare three machine learning algorithms.
- To analyze performance using quantitative evaluation metrics.

3 Dataset Collection and Description

3.1 Data Collection

The dataset was manually curated in a structured manner. A total of **120 documents** were collected, consisting of:

- 60 Sports documents
- 60 Politics documents

Each document contains 2–4 sentences written in a neutral news-reporting style. To avoid trivial classification, ambiguous samples were introduced where:

- Sports articles included political vocabulary (e.g., minister, parliament).
- Political articles included sports metaphors (e.g., match, contest).

This ensured realistic overlap between categories.

3.2 Dataset Split

The dataset was divided into:

- 80% Training data (96 documents)
- 20% Testing data (24 documents)

The split was performed with shuffling and class balance preservation.

4 Text Preprocessing

Before feature extraction, the following preprocessing steps were applied:

1. Conversion to lowercase
2. Removal of URLs
3. Removal of punctuation and special characters
4. Tokenization using whitespace
5. Stopword removal

These steps ensure cleaner and more consistent feature extraction.

5 Feature Representation

5.1 Bag-of-Words (BoW)

Bag-of-Words represents each document as a frequency vector of words. Word order is ignored, and only term frequency is considered.

5.2 TF-IDF

TF-IDF (Term Frequency – Inverse Document Frequency) assigns importance to words based on their frequency within a document and rarity across the corpus.

$$TF\text{-}IDF = TF \times IDF$$

Where:

$$IDF = \log\left(\frac{N}{df}\right)$$

Here N is the total number of documents and df is the number of documents containing the term.

5.3 N-grams

Bigrams were included to capture contextual information such as:

- prime minister
- football team

- national assembly

This improves domain discrimination.

6 Machine Learning Techniques

6.1 Naive Bayes

Multinomial Naive Bayes was implemented with Laplace smoothing. It assumes conditional independence between features and computes:

$$P(C|X) \propto P(C) \prod P(x_i|C)$$

6.2 Logistic Regression

Logistic Regression is a linear discriminative classifier that uses the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Weights were optimized using gradient descent with L2 regularization.

6.3 Support Vector Machine (SVM)

A simplified linear SVM-style classifier was implemented to maximize margin between classes using hinge-based updates.

7 Evaluation Metrics

The following metrics were used:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

Model	Feature	Accuracy	F1 Score
Naive Bayes	BoW	90%	87.50%
Naive Bayes	TF-IDF	90%	87.50%
Logistic Regression	TF-IDF	**95%**	**94.74%**
SVM	TF-IDF	90%	87.50%

Table 1: Performance Comparison of Different Models

8 Experimental Results

8.1 Observations

- Logistic Regression with TF-IDF performed best.
- Naive Bayes performed competitively but slightly lower.
- SVM achieved similar performance to Naive Bayes.
- Introducing ambiguous samples reduced trivial separability.

9 Analysis and Discussion

Logistic Regression performed best because it directly optimizes the classification boundary. TF-IDF improved performance by reducing the influence of common terms.

Naive Bayes performed well despite its independence assumption. SVM also provided stable results but did not outperform Logistic Regression.

10 Limitations

- Dataset size is relatively small (120 samples).
- Data was manually curated.
- SVM implementation is simplified.
- No cross-validation was applied.

Future improvements could include larger datasets and deep learning models.

11 Conclusion

This project demonstrates that classical machine learning models combined with appropriate feature engineering can effectively solve binary text classification tasks.

Logistic Regression with TF-IDF achieved the highest accuracy of 95%. The study highlights the importance of balanced datasets, careful feature selection, and algorithm comparison.