Sports vs Politics Text Classification

Anmol Yadav
IIT Jodhpur, Semester 6

February 15, 2026

Abstract

This project focuses on building a supervised machine learning classifier to categorize text sentences into two domains: Sports and Politics. The study involves automated dataset collection from Wikipedia, feature representation comparison (Bag-of-Words, TF-IDF, and N-grams), evaluation of multiple machine learning models, and deployment of the best-performing model for interactive real-time classification.

1 Introduction

Text classification is a fundamental problem in Natural Language Processing (NLP) where the goal is to assign a label to a text document. In this project, we address a binary classification problem to distinguish sentences belonging to Sports or Politics. Traditional NLP methods like Bag-of-Words, TF-IDF, and N-grams are compared alongside models such as Naive Bayes, Logistic Regression, Linear SVM, and Random Forest.

2 Dataset Construction

2.1 Data Source

The dataset was constructed using automated web crawling from Wikipedia category pages:

- **Sports**: https://en.wikipedia.org/wiki/Category:Sports
- **Politics**: https://en.wikipedia.org/wiki/Category:Politics

2.2 Keyword Extraction

- Up to 500 domain-specific keywords were extracted from article titles per domain. - Only titles containing at least one keyword were considered. - Disambiguation pages and meta pages (containing ":") were excluded to maintain domain relevance.

2.3 Sentence Extraction

- Paragraphs from valid articles were tokenized into sentences using NLTK. - Sentences shorter than 8 words were discarded. - **Final dataset**: 50,000 sentences (25,000 per class).

3 Feature Representation

Three feature extraction techniques were explored:

3.1 Bag-of-Words (BoW)

Represents text as word frequency counts, ignoring word order.

3.2 TF-IDF

Highlights discriminative words by down-weighting common terms across documents.

3.3 N-grams (Unigram + Bigram)

Captures sequences of words to include some contextual information, e.g., "world cup" or "prime minister".

4 Machine Learning Models

Four classifiers were evaluated:

- **Naive Bayes (MultinomialNB)**
- **Logistic Regression**
- **Linear SVM**
- **Random Forest (100 trees)**

Training and testing split: 80% training, 20% testing.

5 Results

5.1 Feature Comparison (Naive Bayes)

Feature	**Accuracy (%)**
Bag-of-Words	96.39
TF-IDF	95.85
N-grams (1,2)	96.59

Table 1: Feature representation comparison using Naive Bayes. N-grams achieve the best performance.

5.2 Model Comparison (Using N-grams)

Model	Accuracy (%)
Naive Bayes	96.59
Logistic Regression	96.38
Linear SVM	96.17
Random Forest	94.44

Table 2: Comparison of models using N-grams as features. Naive Bayes achieved the best accuracy.

5.3 Best Configuration

- Feature: N-grams (1,2)
- Model: Naive Bayes
- Accuracy: 96.59%

6 Feature and Model Comparison

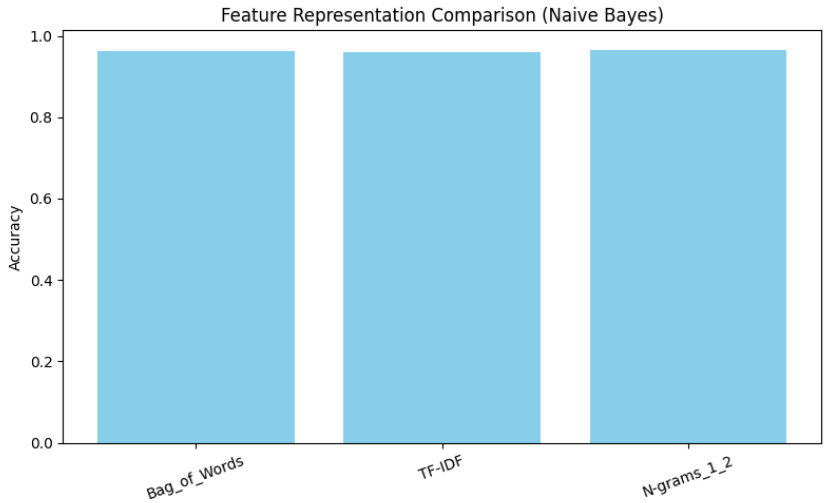

Figure 1: Comparison of feature representation techniques (Bag-of-Words, TF-IDF, N-grams) using Naive Bayes. N-grams achieved the best accuracy.

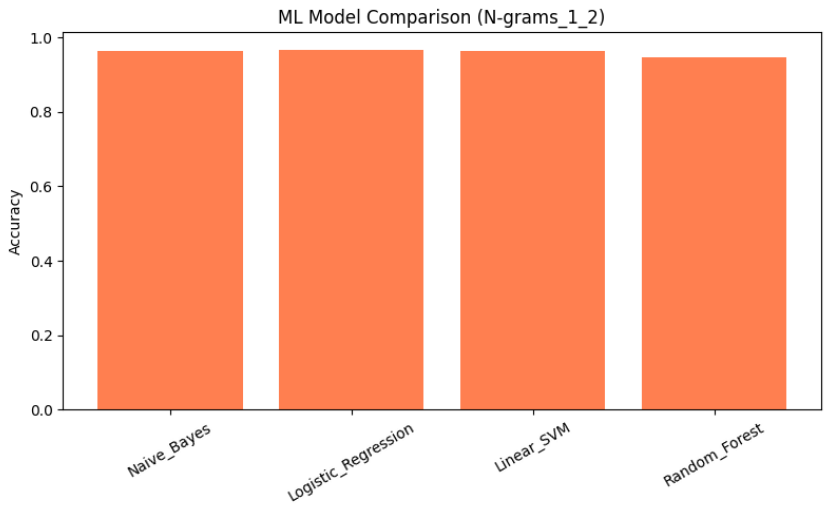

Figure 2: Comparison of machine learning models (Naive Bayes, Logistic Regression, Linear SVM, Random Forest) using N-grams as features. Naive Bayes performed best.

7 Confusion Matrices

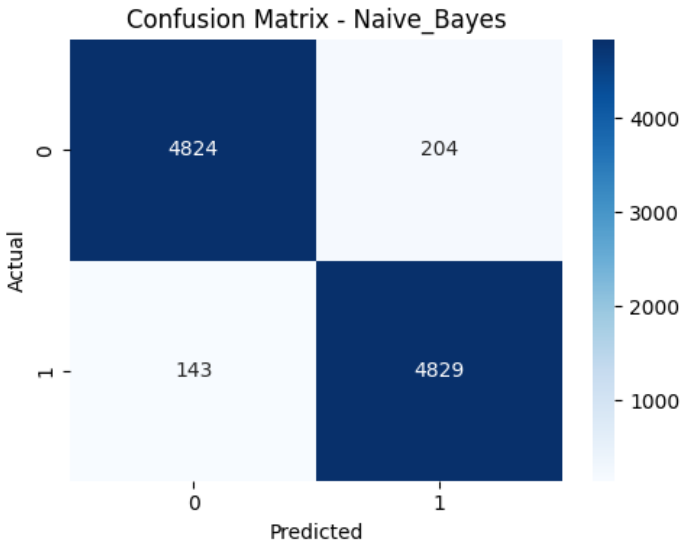

Figure 3: Confusion matrix for Naive Bayes using N-grams.

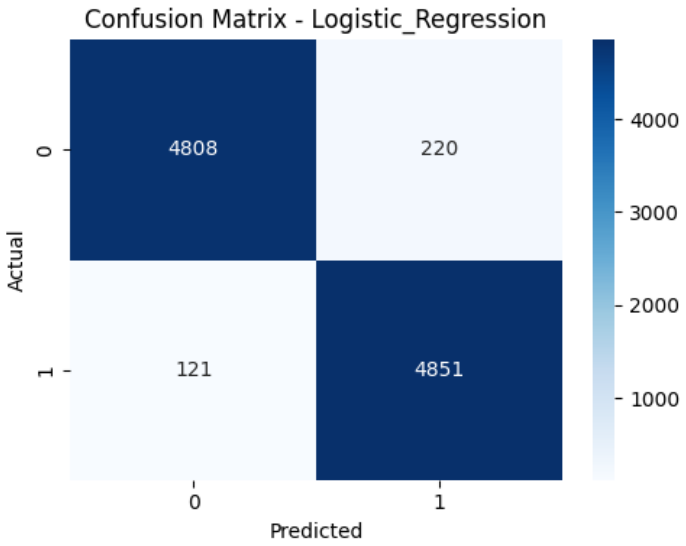

Figure 4: Confusion matrix for Logistic Regression using N-grams.

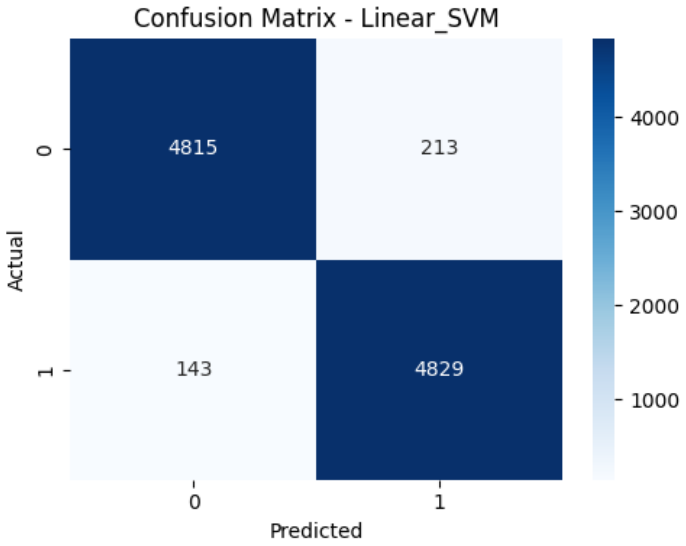

Figure 5: Confusion matrix for Linear SVM using N-grams.

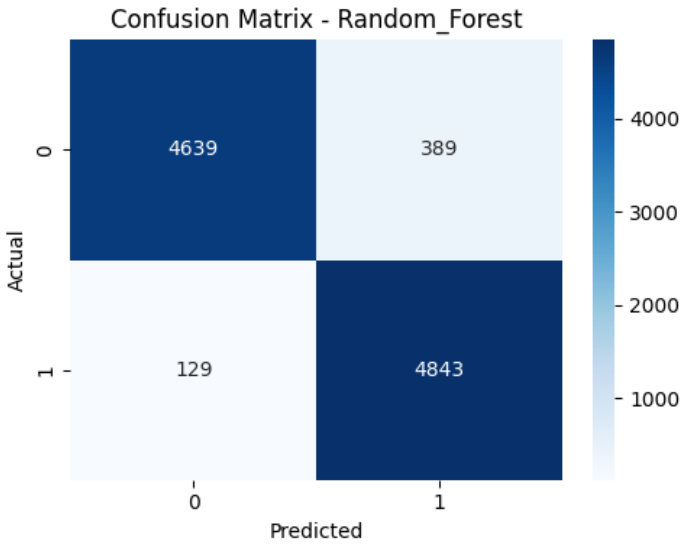

Figure 6: Confusion matrix for Random Forest using N-grams.

8 Discussion

The N-grams feature representation captures contextual information better than Bag-of-Words and TF-IDF, resulting in higher classification accuracy. Among the models, Naive Bayes performed best, likely due to its suitability for text classification with discrete word counts. Logistic Regression and Linear SVM are competitive but slightly lower in performance. Random Forest underperformed slightly, potentially due to its tendency to overfit on high-dimensional sparse data.

9 Conclusion

This project successfully demonstrates an end-to-end pipeline for text classification between Sports and Politics. Automated dataset generation from Wikipedia ensures reproducibility, while feature and model comparison identifies the optimal configuration. The trained classifier can now be deployed for interactive, real-time predictions.

10 Future Work

- Incorporate deep learning models (e.g., LSTM, BERT) for improved contextual understanding. - Extend to multi-class classification beyond Sports and Politics. - Improve dataset by including more sources beyond Wikipedia for better generalization.

11 References

- Wikipedia: https://www.wikipedia.org/
- Scikit-learn documentation: https://scikit-learn.org/stable/
- NLTK documentation: https://www.nltk.org/