Sports vs Politics Document Classifier

Sahilpreet Singh
Roll No: B23CS1061
Course: NLP (CSL7640)
Indian Institute of Technology Jodhpur

February 2026

1 Introduction

This report presents a document classification system that categorizes news articles into two classes: **Sports** and **Politics**. The objective of this assignment is to design and compare multiple classical machine learning techniques using different feature representations including Bag of Words (BoW), TF-IDF, and N-grams.

All models were implemented from scratch using Python's standard library to ensure conceptual clarity and understanding of the algorithms. This project demonstrates how traditional machine learning methods perform on a real-world Natural Language Processing (NLP) task.

2 Data Collection and Dataset Description

2.1 Dataset Source

The dataset used for this task is the **News Category Dataset** available on Kaggle:

`https://www.kaggle.com/datasets/setseries/news-category-dataset`

The dataset contains news headlines and short descriptions sourced from HuffPost between 2012 and 2018. It consists of 50,000 balanced samples across 10 categories.

2.2 Data Filtering

Since the task is binary classification, only the following categories were extracted:

- SPORTS

- POLITICS

From each category, 200 samples were selected to maintain class balance.

2.3 Final Dataset Statistics

Total Samples	400
Sports Samples	200
Politics Samples	200
Train-Test Split	80:20
Training Samples	320
Testing Samples	80

Table 1: Final Dataset Statistics

2.4 Preprocessing Steps

The following preprocessing steps were applied:

1. CSV parsing using Python's `csv` module
2. Filtering only SPORTS and POLITICS rows
3. Using `short_description` field as primary text
4. Lowercasing all text
5. Removing punctuation
6. Tokenization using whitespace splitting

2.5 Vocabulary Analysis

Feature Type	Vocabulary Size
Unigrams	2269
Unigrams + Bigrams	6881

Table 2: Vocabulary Size Comparison

Sports articles frequently contained words such as *game, season, team, championship, player*, while politics articles included *president, senate, legislation, campaign, government.*

3 Feature Representations

3.1 Bag of Words (BoW)

Each document is represented as a vector of word counts:

$$BoW(d, w) = \text{count of word } w \text{ in document } d$$

Advantages:

- Simple implementation
- Effective baseline

Disadvantages:

- Ignores word order
- Sensitive to frequent words

3.2 TF-IDF

TF-IDF assigns importance weights:

$$TF(w, d) = \frac{count(w, d)}{\text{total words in } d}$$

$$IDF(w) = \log\left(\frac{N+1}{df(w)+1}\right) + 1$$

$$TFIDF(w, d) = TF(w, d) \times IDF(w)$$

TF-IDF reduces the influence of common words and highlights discriminative terms.

3.3 N-grams (Unigram + Bigram)

Bigram features capture adjacent word pairs:

Example:

- Unigrams: india, won, match
- Bigrams: india won, won match

Bigram representation significantly increases feature dimensionality and sparsity.

4 Machine Learning Techniques

4.1 Multinomial Naive Bayes

Based on Bayes' theorem:

$$P(c|d) \propto P(c) \prod_i P(w_i|c)$$

Laplace smoothing:

$$P(w|c) = \frac{count(w, c) + 1}{total\_words(c) + |V|}$$

Log probabilities were used to prevent underflow.

4.2 Logistic Regression

Probability modeled using sigmoid:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w \cdot x + b)}}$$

Training was done using gradient descent on binary cross-entropy loss.
Hyperparameters:

- Learning rate: 0.1
- Epochs: 300

4.3 K-Nearest Neighbors (KNN)

Cosine similarity used as distance metric:

$$sim(a, b) = \frac{a \cdot b}{||a||||b||}$$

- K = 5
- Majority voting among nearest neighbors

5 Results and Quantitative Comparison

5.1 Accuracy Comparison

Technique	Feature	Accuracy
Naive Bayes	BoW	80.00%
Logistic Regression	TF-IDF	77.50%
KNN (k=5)	Bigrams	65.00%

Table 3: Accuracy Comparison

5.2 Observations

- Naive Bayes performed best at 80%.
- Logistic Regression performed competitively.
- KNN performed worst due to sparsity in bigram space.
- Politics articles showed higher recall across models.

6 Limitations

1. Only 400 samples used.
2. No cross-validation performed.
3. Limited preprocessing (no stemming or lemmatization).
4. Binary classification only.
5. No semantic embeddings used.
6. Hyperparameters not extensively tuned.

7 Conclusion

This project demonstrates that classical machine learning techniques remain effective for document classification. Naive Bayes with Bag of Words achieved the highest accuracy of 80%, showing that probabilistic models handle sparse high-dimensional data well.

Logistic Regression also performed strongly with TF-IDF features. KNN struggled due to high-dimensional sparsity.

While these methods provide reasonable results, modern deep learning models such as Transformers would likely outperform them by capturing contextual semantics rather than relying purely on surface-level statistics.

8 References

1. Manning, C.D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval.

2. Jurafsky, D., & Martin, J.H. (2023). Speech and Language Processing.

3. Sebastiani, F. (2002). Machine learning in automated text categorization.

4. Kaggle News Category Dataset.