Sports vs Politics Text Classification Using Machine Learning

Manya Jain
Roll Number: B22CS032

Contents

1 Introduction

Text classification is one of the most widely studied problems in Natural Language Processing (NLP). With the rapid growth of online news platforms, there is an increasing need to automatically categorize articles into different topics such as sports, politics, technology, business, etc.

In this project, the objective is to design a binary classifier that determines whether a given news article belongs to the **Sports** category or the **Politics** category. Although these two categories appear distinct, they can sometimes share overlapping vocabulary. For example, words such as "campaign", "leader", or "victory" may appear in both contexts.

Instead of relying on rule-based keyword matching, machine learning techniques are applied to learn patterns from real data. The performance of three different machine learning models is compared to determine which approach works best for this task.

The main objectives of this project are:

- To collect and preprocess a real-world dataset.
- To represent textual data numerically using TF-IDF features.
- To train and compare three machine learning algorithms.
- To evaluate their performance using quantitative metrics.
- To analyze the strengths and limitations of the system.

2 Dataset Collection and Description

The dataset used for this project was obtained from Kaggle's *News Category Dataset*. This dataset contains news headlines along with short descriptions and category labels.

2.1 Data Selection

Only two categories were selected:

- SPORTS
- POLITICS

The headline and short description were combined to form the input text for classification.

2.2 Class Distribution

Initially, the dataset was imbalanced:

- Sports samples: 5077
- Politics samples: 35602

Such imbalance can bias the classifier toward predicting the majority class. To address this issue, the dataset was balanced by selecting 5077 samples from each class.

After balancing:

- Total samples: 10,154
- Sports: 5077
- Politics: 5077

Balancing ensures fair comparison between the two categories.

3 Data Preprocessing and Feature Engineering

Before training the models, several preprocessing steps were applied.

3.1 Text Cleaning

- All text was converted to lowercase.
- Headline and short description were merged.
- Stopwords were removed using TF-IDF's built-in English stopword list.

No aggressive stemming or lemmatization was applied in order to preserve contextual meaning.

3.2 Feature Representation Using TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) was used to convert textual data into numerical form.

The formula is:

$$TFIDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right)$$

Where:

- $TF(t,d)$ = frequency of term t in document d
- N = total number of documents
- $DF(t)$ = number of documents containing term t

Unigrams and bigrams were used. Bigrams help capture short phrases such as "prime minister" or "world cup", which are highly informative.

4 Machine Learning Techniques

Three machine learning algorithms were implemented.

4.1 Naive Bayes

Naive Bayes is based on Bayes' theorem:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

It assumes conditional independence among features. Despite this assumption, it performs well for high-dimensional text data.

4.2 Logistic Regression

Logistic Regression models the probability of a class using:

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

It is a linear classifier and is widely used for binary classification problems.

4.3 Support Vector Machine (SVM)

SVM attempts to find the hyperplane that maximizes the margin between classes. It is highly effective for sparse, high-dimensional data such as TF-IDF vectors.

5 Experimental Setup

- Train-test split: 80% training, 20% testing.
- TF-IDF vectorization with unigrams and bigrams.
- Evaluation metrics:
  - Accuracy
  - Precision
  - Recall
  - F1-score

6 Results and Quantitative Comparison

Model	**Accuracy**
Naive Bayes	97.24%
Logistic Regression	97.14%
Support Vector Machine	97.68%

All three models achieved very high accuracy.

6.1 Observations

- SVM performed slightly better than the other models.
- Naive Bayes performed almost equally well.
- Logistic Regression showed balanced precision and recall.

The small performance difference indicates that the two categories are well-separated in feature space.

7 Error Analysis

Although accuracy is high, some misclassifications were observed.

Possible reasons include:

- Articles discussing political decisions affecting sports.
- Headlines with ambiguous vocabulary.
- Short descriptions lacking enough context.

This shows that while statistical models perform well, they do not fully understand semantic meaning.

8 Limitations

- The system works only for English text.
- It does not capture deep semantic relationships.
- Dataset balancing may remove useful samples.
- Real-world deployment may require continuous retraining.

9 Future Work

Possible improvements include:

- Using word embeddings such as Word2Vec or GloVe.
- Applying deep learning models like LSTM or Transformers.
- Performing cross-validation instead of a single train-test split.
- Testing on live news streams.

10 Conclusion

This project successfully implemented a Sports vs Politics text classifier using TF-IDF features and three different machine learning models.

All models performed strongly, with Support Vector Machine achieving the best accuracy. The results demonstrate that classical machine learning techniques remain highly effective for structured text classification tasks.

Overall, this project highlights the importance of feature engineering, balanced datasets, and proper model evaluation in solving real-world NLP problems.