Text Classification of Sports and Politics News Using Machine Learning Techniques

1. Introduction

Text classification is a fundamental problem in Natural Language Processing (NLP) that involves automatically assigning predefined categories to textual documents. With the rapid growth of digital content, especially online news articles, automatic classification has become essential for organizing information, filtering content, and improving information retrieval systems.

News articles often span multiple domains such as sports, politics, technology, and entertainment. Manual categorization of such content is time consuming and impractical at scale. As a result, machine learning based text classification techniques are widely used to automate this process. These techniques rely on extracting meaningful numerical features from text and training models to recognize patterns associated with different categories.

In this project, a binary text classification task is explored, where documents are classified into Sports or Politics categories. These two domains were chosen because they are commonly found in news media and exhibit distinct thematic and lexical characteristics. Sports articles typically focus on matches, teams, players, and tournaments, while political articles discuss government policies, elections, leadership, and international relations.

The objective of this work is not only to build an effective classifier, but also to study the impact of different feature representations and machine learning models on classification performance. Multiple techniques are compared, and their strengths and limitations are analyzed through quantitative evaluation.

2. Dataset Collection and Description

The dataset used in this study was created specifically for the sports versus politics classification task. The aim was to construct a small but representative dataset that captures common language patterns found in news articles, while remaining manageable for experimentation and analysis.

The dataset consists of two classes:

- Sports
- Politics

Each class contains 30 short news style documents, resulting in a total of 60 documents. Each document is written as a short paragraph containing approximately two to four sentences, and each paragraph is treated as a single document. The dataset is stored in plain text format, with one document per line.

Sports documents include content related to football, cricket, tennis, basketball, and other sporting events. These texts focus on topics such as matches, tournaments, teams, players, coaching decisions, and championships. The vocabulary commonly reflects competition, performance, scores, and athletic preparation.

Political documents include discussions on government decisions, elections, parliamentary debates, public policies, leadership changes, and international relations. The vocabulary in these documents reflects governance, legislation, diplomacy, and political ideology.

The documents were manually written and paraphrased based on common themes found in news media. Full news articles were not copied directly; instead, short rewritten summaries were used to ensure originality and avoid copyright issues. An equal number of documents were included in both classes to maintain balance and prevent bias during model training.

The dataset is stored in two separate files: sports.txt and politics.txt, where each line corresponds to one document belonging to the respective category. Although the dataset size is relatively small, it is sufficient for comparing different feature representations and machine learning models, which is the primary goal of this study.

---

3. Feature Representation

Since machines process numbers rather than words, we transformed the text using three distinct methods:

3.1 Bag of Words (BoW)

The Bag of Words model represents each document as a vector of word frequencies. Each dimension corresponds to a unique word in the vocabulary, and the value represents how often that word appears in the document. Word order and grammatical structure are ignored in this representation.

Despite its simplicity, BoW often performs well for text classification tasks, especially when domain specific words are present. In this dataset, words such as *match*, *team*, and *tournament* are strongly associated with sports, while words like *government*, *election*, and *parliament* are more common in political documents.

However, a limitation of BoW is that it assigns equal importance to all words and does not capture contextual relationships between them.

3.2 Term Frequency Inverse Document Frequency (TF-IDF)

TF-IDF improves upon the Bag of Words model by assigning weights to words based on their importance. While term frequency measures how often a word appears in a document, inverse document frequency reduces the weight of words that appear frequently across many documents.

This representation highlights words that are more discriminative for a particular class and reduces the influence of common words. TF-IDF generally produces more informative feature vectors and often leads to improved classification performance.

3.3 TF-IDF with N-grams

While BoW and TF-IDF treat words independently, they do not capture short contextual phrases. To address this limitation, **bigrams (n = 2)** were used in combination with TF-IDF.

N-grams represent sequences of adjacent words, allowing the model to capture phrases such as *prime minister*, *world cup*, or *national election*. These phrases often carry more semantic meaning than individual words. However, using n-grams increases the dimensionality of the feature space and may require more data to avoid sparsity issues.

4. Machine Learning Models

To perform classification, three widely used machine learning models were selected. These models are commonly applied in text classification tasks and provide a good balance between simplicity, interpretability, and performance.

4.1 Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that features are conditionally independent given the class label. Although this assumption is simplistic, Naive Bayes often performs well for text classification problems.

The model works directly with word frequency based features and is computationally efficient. It is particularly effective when there is clear vocabulary separation between classes.

4.2 Logistic Regression

Logistic Regression is a linear discriminative classifier that models the probability of a document belonging to a class based on a weighted sum of features. Unlike Naive Bayes, it does not assume feature independence and directly learns decision boundaries.

Logistic Regression performs well with TF-IDF features and is relatively interpretable, as feature weights indicate the importance of words for classification.

4.3 Support Vector Machine (SVM)

Support Vector Machine is a powerful classifier that aims to find a decision boundary that maximizes the margin between classes. Linear SVMs are particularly well suited for text classification due to the high dimensional and sparse nature of textual data.

SVMs often achieve strong performance and generalize well, especially when combined with TF-IDF features.

---

5. Experimental Results

5.1 Experimental Setup

The dataset was divided into training and testing sets using an **80–20 split**. Three feature representations (BoW, TF-IDF, TF-IDF with bigrams) were combined with three machine learning models (Naive Bayes, Logistic Regression, and SVM), resulting in multiple model feature combinations.

Performance was evaluated using **Accuracy, Precision, Recall, and F1 score**.

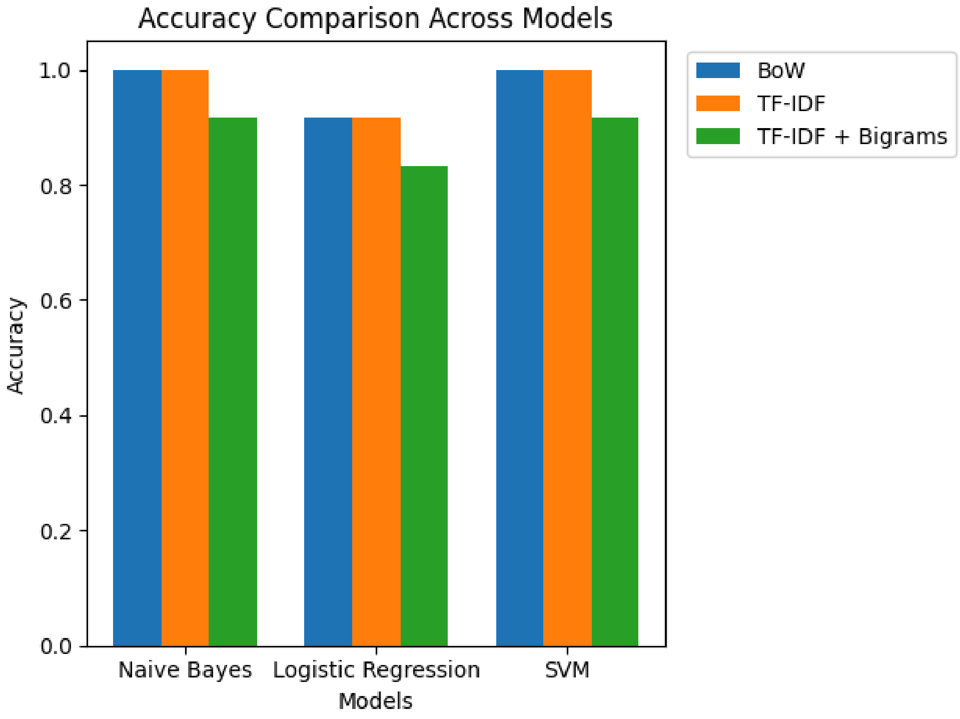
5.2 Results

The experimental results are summarized below.

Performance Comparison Table

Model	Feature Representation	Accuracy	Precision	Recall	F1-Score
Naive Bayes	BoW	1.00	1.00	1.00	1.00
Logistic Regression	BoW	0.92	0.93	0.92	0.91
SVM	BoW	1.00	1.00	1.00	1.00
Naive Bayes	TF-IDF	1.00	1.00	1.00	1.00
Logistic Regression	TF-IDF	0.92	0.93	0.92	0.92
SVM	TF-IDF	1.00	1.00	1.00	1.00
Naive Bayes	TF-IDF + Bigrams	0.92	0.93	0.92	0.92
Logistic Regression	TF-IDF + Bigrams	0.83	0.89	0.83	0.84
SVM	TF-IDF + Bigrams	0.92	0.93	0.92	0.90

A bar chart was generated to visualize the accuracy comparison across models and feature representations.



6. Discussion and Limitations

The results indicate that both Naive Bayes and SVM perform exceptionally well on this task, particularly with simpler feature representations such as Bag of Words and TF-IDF unigrams. This suggests that the sports and politics domains exhibit strong lexical separation.

Logistic Regression performs competitively but is more sensitive to feature complexity. Its performance decreases when bigram features are introduced, likely due to increased dimensionality and limited dataset size.

A key limitation of this study is the relatively small dataset. While the results are strong, they may not generalize well to larger or more diverse datasets. Additionally, the dataset consists of carefully curated documents, which may not fully reflect real world noise and ambiguity.

---

7. Conclusion

In this project, a machine learning based classifier was designed to distinguish between sports and politics news articles. Multiple feature representations and classification models were implemented and compared. The experimental results demonstrate that traditional machine learning techniques can achieve high performance on this task when combined with appropriate feature representations.

Among the evaluated models, Support Vector Machine showed the most consistent and robust performance across different feature sets. Simpler feature representations were found to be sufficient for this classification task.

Future work could involve expanding the dataset, introducing additional categories, applying advanced preprocessing techniques, or exploring deep learning based approaches such as neural networks and transformers to improve generalization.

GitHub Repository

All source code, datasets, experimental results, and implementation details related to this problem are available in the following GitHub repository:

Repository URL: Link