Problem 4: Sports or Politics – GitHub link

Name: Kiran S
Roll No.: B22CS100

Abstract

This report presents the design, implementation, and evaluation of a machine learning system that classifies a text **document** as either **Sports** or **Politics**. The task is formulated as a supervised binary text classification problem. The dataset used in this work was obtained from Kaggle and consists of paragraph-length news descriptions labeled as Politics (0) and Sports (1). We explore three widely used feature representations for text—**Bag of Words (BoW)**, **TF-IDF**, and **n-grams**—and compare three different machine learning techniques: **Multinomial Naive Bayes**, **Logistic Regression**, and **Support Vector Machine (SVM)**. The system is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Experimental results show that linear models combined with TF-IDF features achieve near-perfect performance on this dataset, indicating that the two categories are highly separable in terms of vocabulary. We also discuss the limitations of the dataset and the model, and outline possible future improvements.

1. Introduction

Text classification is one of the most fundamental tasks in Natural Language Processing (NLP). It involves automatically assigning predefined categories to textual documents. Common applications include spam detection, sentiment analysis, topic classification, and document filtering. In this project, the goal is to design a classifier that reads a **text document** and classifies it into one of two categories: **Sports** or **Politics**.

These two categories are semantically distinct, but in real-world news articles they sometimes share overlapping vocabulary (for example, words such as “win”, “campaign”, “team”, or “lead”). This makes the task a good test case for studying how different feature representations and machine learning algorithms perform on textual data.

The objectives of this project are: - To build a complete text classification pipeline starting from data loading to evaluation. - To experiment with different feature representations: Bag of Words, TF-IDF, and n-grams. - To train and compare at least three machine learning techniques. - To analyze and compare their performance quantitatively. - To discuss the limitations of the system and suggest future improvements.

2. Data Collection

2.1 Data Source - link

The dataset used in this project was obtained from **Kaggle**, a popular platform for data science datasets and competitions. The dataset consists of news-related textual data with two columns: - **text**: A paragraph-length description or article content. - **label**: A binary label where `0` denotes **Politics** and `1` denotes **Sports**.

Each row in the dataset corresponds to a single document. The documents are not just short sentences or headlines; instead, they are relatively long paragraphs containing multiple sentences, which makes them suitable for **document-level classification** as required by the problem statement.

2.2 Motivation for Using This Dataset

This dataset was chosen because: - It is already labeled, which makes it suitable for supervised learning. - It contains real-world news-like text. - It has two clear categories that match the problem statement exactly: Sports and Politics. - The text length is sufficiently large to be considered a “document” rather than a short sentence.

2.3 Dataset Size and Distribution

The dataset contains several documents in total, with roughly balanced class distribution between Politics and Sports. The labels are encoded as: - `0` → Politics - `1` → Sports

Although the dataset is stored in **CSV format**, each row corresponds to a full paragraph-length news document. Therefore, the task is treated as **document-level text classification**, where each document is represented by its textual content and assigned a category label (Sports or Politics).

Before training the models, the dataset is randomly shuffled and split into training and test sets to ensure that both classes are represented proportionally in each split.

---

3. Dataset Description and Analysis

3.1 Basic Statistics

Some basic statistics were computed to understand the dataset: - Number of documents in total. - Number of documents per class (Politics vs Sports). - Average length of documents (in words).

It was observed that political documents are, on average, slightly longer than sports documents, but both classes contain paragraph-level text with sufficient information for classification.

3.2 Vocabulary Characteristics

After basic preprocessing, the vocabulary size of the dataset is quite large, containing tens of thousands of unique words. Many words are domain-specific, such as: - Politics: “election”, “government”, “parliament”, “minister”, “policy”, “campaign”. - Sports: “match”, “goal”, “tournament”, “team”, “player”, “score”.

This strong presence of domain-specific terms suggests that the two classes may be highly separable using standard text features.

3.3 Potential Issues in the Data

- The dataset is relatively clean and well-labeled, which can lead to very high accuracy scores.
- Some words act as strong indicators of a class, making the classification problem easier than real-world noisy scenarios.
- The dataset was originally ordered by class (all Politics first, then all Sports), but this ordering was removed by shuffling during the train-test split process.

---

4. Preprocessing

Before feeding the text into machine learning models, several preprocessing steps were applied:

1. **Lowercasing**: All text was converted to lowercase to avoid treating “Government” and “government” as different tokens.
2. **Removing punctuation and numbers**: Non-alphabetic characters were removed to reduce noise.
3. **Whitespace normalization**: Extra spaces were removed.
4. **Stopword removal** (handled by the vectorizers): Common words such as “the”, “is”, “and” were removed as they carry little discriminative information.

After preprocessing, each document is represented as a clean sequence of words suitable for feature extraction.

---

5. Feature Representation

Machine learning models require numerical input, so the text documents must be converted into numerical feature vectors. In this project, three types of feature representations were explored.

5.1 Bag of Words (BoW)

The Bag of Words model represents each document as a vector of word counts. Each dimension corresponds to a word in the vocabulary, and the value indicates how many times that word appears in the document.

Advantages: Simple and intuitive. Works well for many texts classification tasks.

Disadvantages: Does not consider word importance across documents. Ignores word order and context.

5.2 TF-IDF (Term Frequency–Inverse Document Frequency)

TF-IDF improves upon BoW by weighting words based on how important they are to a document relative to the entire corpus. Words that appear frequently in a document but rarely across documents get higher weights.

Advantages: Reduces the impact of very common but uninformative words. Often improves performance of linear classifiers.

5.3 n-grams

Instead of using only single words (unigrams), n-grams consider sequences of words. In this project, **unigrams and bigrams** were used. This allows the model to capture short phrases such as “prime minister” or “world cup”.

Using n-grams increases the dimensionality of the feature space but can improve performance by capturing local context.

---

6. Machine Learning Techniques

To satisfy the requirement of comparing at least three machine learning techniques, the following models were used:

6.1 Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic classifier based on Bayes’ theorem and the assumption of conditional independence between features. Despite its simplicity, it is very effective for text classification tasks and serves as a strong baseline.

6.2 Logistic Regression

Logistic Regression is a linear classifier that models the probability of a class using a logistic function. When combined with TF-IDF features and proper regularization, it performs very well on high-dimensional sparse text data.

6.3 Support Vector Machine (SVM)

A linear Support Vector Machine aims to find a decision boundary that maximizes the margin between the two classes. SVMs are known to perform extremely well in text classification due to the high-dimensional and sparse nature of text features.

---

7. Experimental Setup

7.1 Train-Test Split

The dataset was split into training and test sets using a 70/30 split. The split was performed with shuffling and stratification to ensure that both classes are equally represented in both sets.

7.2 Feature Extraction Settings

- Bag of Words with a maximum vocabulary size of 5000 features.
- TF-IDF with unigrams and bigrams, also limited to 5000 features.

The `max_features` parameter was used to control dimensionality, reduce noise from very rare words, and keep computation efficient.

7.3 Evaluation Metrics

The models were evaluated using the following metrics: - Accuracy - Precision - Recall - F1-score

These metrics provide a comprehensive view of classification performance, especially in binary classification tasks.

---

8. Results and Quantitative Comparison

The following models and feature combinations were evaluated:

- Multinomial Naive Bayes + Bag of Words
- Logistic Regression + TF-IDF (with n-grams)
- Support Vector Machine + TF-IDF (with n-grams)

A summary of the results is shown below:

Model	Features	Accuracy
Naive Bayes	BoW	0.9964
Logistic Regression	TF-IDF (1–2 grams)	1.0000
SVM	TF-IDF (1–2 grams)	1.0000

Both Logistic Regression and SVM achieved perfect accuracy on the test set, while Naive Bayes performed slightly worse but still extremely well.

8.1 Sanity Check

To ensure that these high scores were not due to data leakage, a shuffled-label experiment was performed. When the labels were randomly shuffled, the accuracy dropped to approximately 50%, which is close to random guessing. This confirms that the models are learning genuine patterns from the data rather than exploiting a flaw in the pipeline.

9. Error Analysis

Very few misclassifications were observed. The few errors that did occur typically involved: - Articles that use sports metaphors in political contexts or vice versa. - Documents that contain mixed content (for example, a political article referencing a sports event).

These cases highlight the inherent ambiguity that can exist in natural language.

10. Limitations of the System

Despite the excellent performance, the system has several limitations:

1. **Dataset Simplicity**: The dataset appears to be highly separable due to strong domain-specific vocabulary. Real-world data is often noisier and more ambiguous.
2. **Binary Classification Only**: The system only handles two categories. Real news classification problems often involve many categories.
3. **Language Dependency**: The system is trained only on English text and may not generalize to other languages.
4. **Shallow Models**: The models used are linear and do not capture deep semantic relationships between words.

11. Conclusion

In this project, a complete document classification system for distinguishing between Sports and Politics news articles was developed. Three different machine learning techniques were compared using multiple feature representations. The results show that linear classifiers with TF-IDF features perform extremely well on this dataset, achieving near-perfect accuracy. While the results are very strong, they also highlight that the dataset is relatively easy and clean. Future work should focus on more challenging datasets and more advanced models to improve generalization.