Sports or Politics: A Comparative Study of Text Classification Techniques

Name: Sarita Mandal
Roll Number: M25CSE028

Abstract

This report describes the development and evaluation of an automated text classifier designed to distinguish between ”Sports” and ”Politics” documents. Three distinct feature representation methods are explored: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and TF-IDF with N-grams. Three machine learning algorithms are compared: Multinomial Naive Bayes, K-Nearest Neighbors (KNN), and Decision Tree. The methodology consists of a comprehensive pipeline involving data collection, exploratory dataset analysis, model training, and quantitative comparison. The model evaluation metrics used are accuracy, precision, recall, F1-score and confusion matrix. As per our findings, Naive Bayes consistently emerges as the top-performing algorithm, achieving approximately 95.7% accuracy across all feature sets.

1 Introduction

The exponential growth of digital text necessitates automated systems capable of categorizing information efficiently. This project aims to build a binary text classifier capable of separating documents into two domains: Sports and Politics. The objective is to evaluate the efficacy of classical Natural Language Processing (NLP) feature extraction techniques combined with basic machine learning algorithms.

2 Data Collection

A diverse dataset is required for training and validation of the classifiers. The source dataset used is the 20 Newsgroups corpus, which serves as a benchmark in text classification literature.

To create the binary target labels, specific sub-categories from the source dataset are scraped and aggregated:

- **Sport Category:** Documents are aggregated from the `rec.sport.baseball` and `rec.sport.hockey` boards.

- **Politics Category:** Documents were aggregated from the `talk.politics.guns`, `talk.politics.mideast`, and `talk.politics.misc` boards.

This methodology simulates real-world data scraping of news articles and forum discussions. Headers, footers, and quote blocks are removed in the pre-processing step in order to ensure the models learn from the actual semantic content rather than metadata.

Table 1 illustrates a snapshot of the preprocessed dataset, showing the raw text content alongside the assigned binary labels.

Table 1: Sample records from the processed Sports vs. Politics dataset

	Text	**Label**
0	\n\nWell over 100,000 in Lebanon alone.\n1,000...	Politics
1	\nWell they could unseal the original warrent ...	Politics
2	Come on Boston, where the hell are you? Seven ...	Sport
3	\n\nPunch Imlach's contributions as a coach an...	Sport
4	# ## Absolutely nothing, seeing as there is no...	Politics

3 Dataset Description and Analysis

Exploratory Data Analysis (EDA) is conducted to understand the distribution and characteristics of the collected text.

3.1 Class Distribution and Word Count

The dataset comprises 4618 documents. The distribution between the two classes is relatively balanced with there being 1993 sports documents and 2625 politics documents. This prevents the machine learning models from developing a majority-class bias.

The average word count per document is computed to be 223.81 words while the max word count in a document is found to be 11251 words.

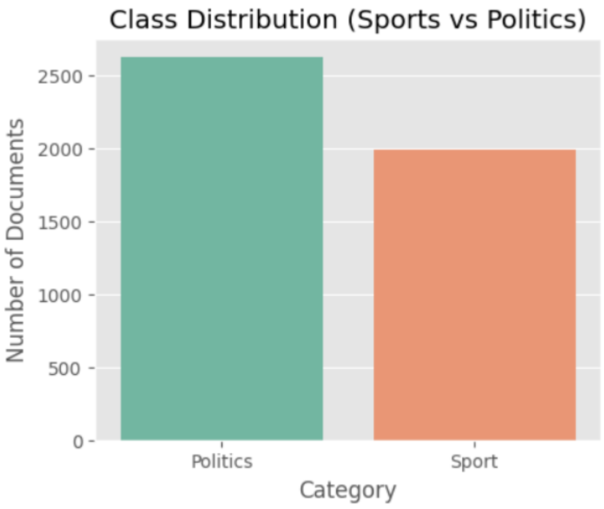

Figure 1: Distribution of documents across Sports and Politics classes

3.2 Document Length Characteristics

An analysis of the document lengths, measured in word count, reveals a positively skewed distribution for both classes, as illustrated in Figure 2. The majority of documents across both ”Sport” and ”Politics” categories contain fewer than 200 words, with a peak frequency observed in the 50-100 word range. As document length increases beyond 200 words, the frequency drops significantly, resulting in a long tail that extends towards 1000 words.

The overlaid kernel density estimation (KDE) curves suggest that the distribution shapes are remarkably similar between the two classes. However, the ”Politics” category appears to have a slightly thicker tail compared to the ”Sport” category, indicating a marginally higher presence of longer-form documents in political discussions. To mitigate the impact of potential outliers and noise associated with extremely long or short texts, the analysis focuses on this primary distribution range.

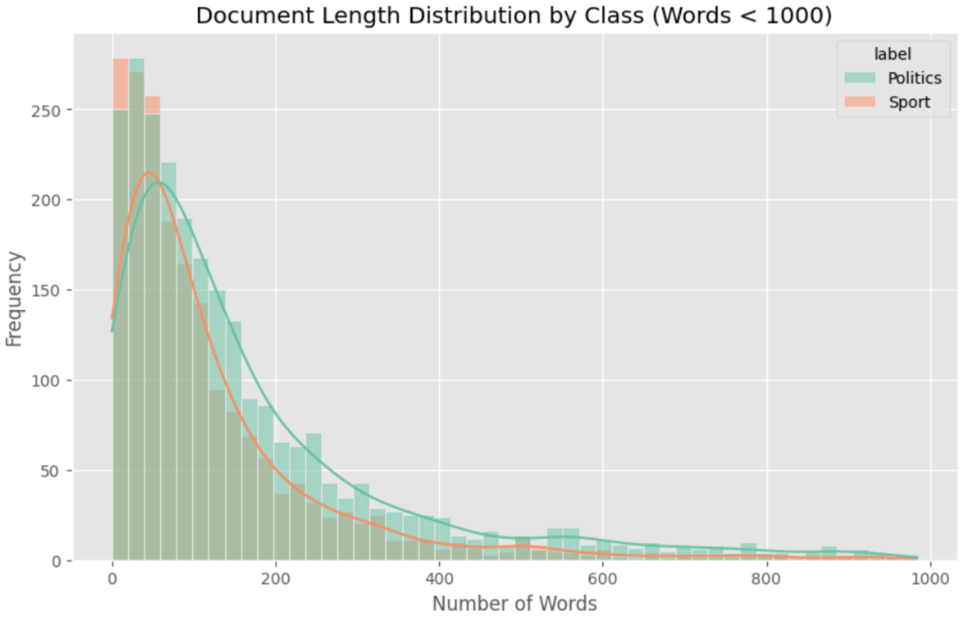

Figure 2: Histogram of document word counts, truncated at 1000 words for visibility.

Due to variance, the vocabulary was constrained to the top 5,000 most frequent words during feature extraction to filter out noise, rare misspellings, and highly obscure domain jargon.

4 Feature Representation Techniques

Since machine learning algorithms require numerical input, the unstructured text is transformed into numerical vectors using three distinct methodologies:

1. **Bag of Words (BoW):** This technique creates a vocabulary of all unique words and represents each document as a vector of word frequency. It is simple but gives disproportionate weight to frequent but uninformative words.

2. **TF-IDF:** Term Frequency-Inverse Document Frequency addresses the flaw of BoW by scaling down the weight of words that appear in many documents (e.g., "said", "the") and scaling up rare, highly indicative words.

3. **TF-IDF + N-grams:** Single-word representations often lose context. By including bigrams, the model captures localized semantic context alongside TF-IDF weighting.

5 Machine Learning Techniques

Three distinct machine learning models are built and compared:

5.1 Multinomial Naive Bayes

Naive Bayes is a probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between the features. The Multinomial variant is specifically designed for text classification. It excels in high-dimensional spaces because it treats each word's probability independently, avoiding the "curse of dimensionality" problem in distance-based models.

5.2 K-Nearest Neighbors (KNN)

KNN is an instance-based learning algorithm that classifies a document based on the majority vote of its 'K' nearest neighbors in the feature space. For this implementation, the model was improved by changing the distance metric from Euclidean to cosine similarity, which measures the angle between vectors. This is critical for text data, as it evaluates documents based on topic rather than length.

5.3 Decision Tree

Decision Trees classify instances by sorting them down a tree from the root to a leaf node. To prevent massive overfitting on the 5,000 features, the model was tuned to use entropy as the splitting criterion and enforced a minimum of 10 samples per leaf node. This ensures generalization.

6 Quantitative Comparisons

The dataset was split into an 80% training set and a 20% testing set. The evaluation metrics used are accuracy, precision, recall, F1-score and confusion matrix.

6.1 Experimental Results

Table 2 summarizes the performance metrics for all nine experimental configurations.

Table 2: Quantitative Comparison of Models across Feature Sets

Feature Set	Model	Accuracy	Precision	Recall	F1-Score
Bag of Words	Naive Bayes	0.9567	0.9576	0.9567	0.9565
Bag of Words	KNN	0.9069	0.9123	0.9069	0.9076
Bag of Words	Decision Tree	0.8366	0.8424	0.8366	0.8377
TF-IDF	Naive Bayes	0.9567	0.9580	0.9567	0.9564
TF-IDF	KNN	0.9286	0.9316	0.9286	0.9290
TF-IDF	Decision Tree	0.8398	0.8506	0.8398	0.8412
TF-IDF + N-grams	Naive Bayes	0.9556	0.9570	0.9556	0.9553
TF-IDF + N-grams	KNN	0.9297	0.9322	0.9297	0.9300
TF-IDF + N-grams	Decision Tree	0.8571	0.8574	0.8571	0.8557

6.2 Analysis of Results

The experimental data reveals several key trends regarding model efficacy and feature representation:

- **Dominance of Naive Bayes:** Naive Bayes consistently emerged as the top-performing algorithm, achieving approximately 95.7% accuracy across all feature sets. Interestingly, its performance remained stable regardless of whether Bag of Words or TF-IDF was used, suggesting that the mere presence of domain-specific keywords (e.g., ”stadium” vs. ”senate”) provides a sufficient signal for the probabilistic model, even without complex weighting.

- **Impact of TF-IDF on KNN:** Unlike Naive Bayes, the K-Nearest Neighbors (KNN) model showed significant sensitivity to feature scaling. Accuracy improved from 90.7% using Bag of Words to 92.9% using TF-IDF. This validates the hypothesis that penalizing frequent, non-informative words is crucial for distance-based algorithms, as it prevents common terms from distorting the cosine similarity calculations.

- **Decision Tree Performance:** Decision Tree was consistently the weakest performer, with accuracy ranging between 83.7% and 85.7%. However, the inclusion of N-grams (bigrams) provided a noticeable boost to the Decision Tree (reaching its peak of 85.7%). This suggests that the tree structure benefits from the localized context provided by word pairs which helps it form more distinct rules.

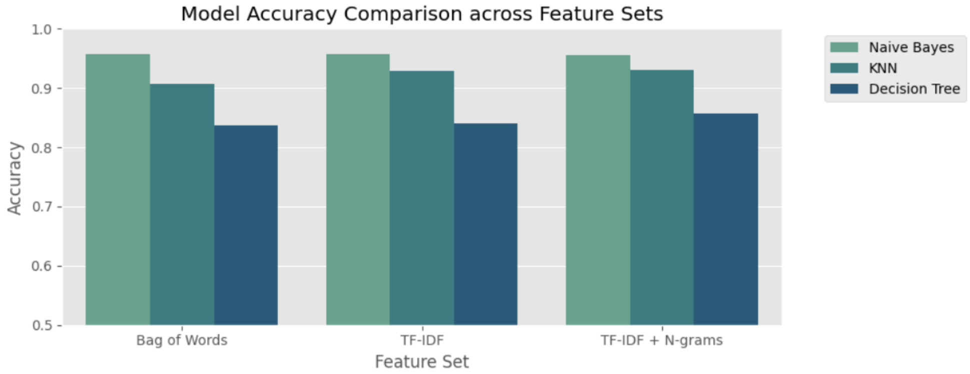

Figure 3: Visual comparison of Model Accuracy across Feature Sets.

6.3 Confusion Matrices and Analysis

To understand the nature of the misclassifications, the confusion matrices for each feature set are analyzed. The matrices, presented in Figures 4, 5, and 6, display the True Positives (diagonal) versus classification errors (off-diagonal).

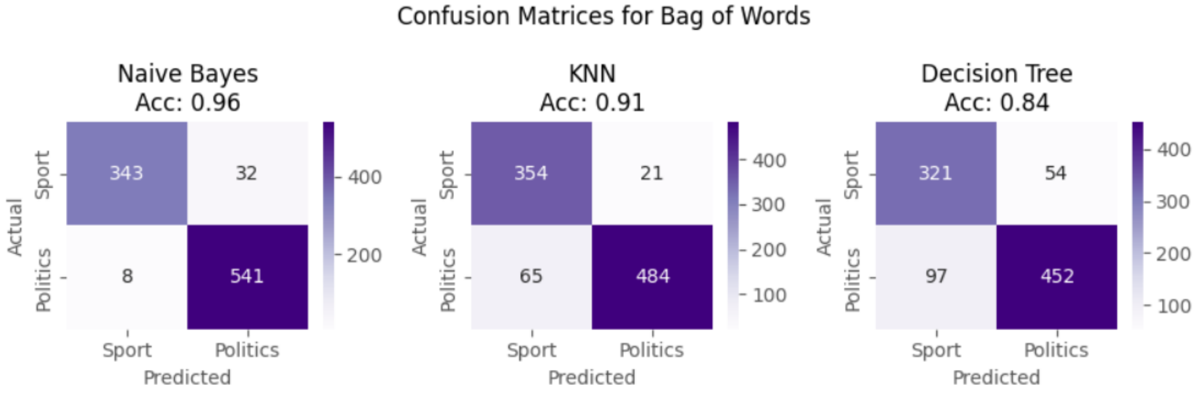

Figure 4: Confusion Matrices for Bag of Words (BoW).

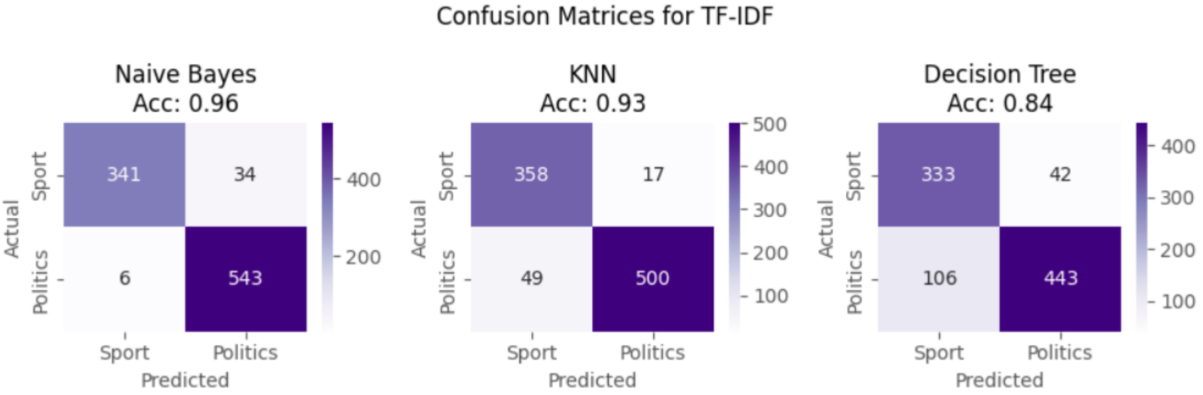

Figure 5: Confusion Matrices for TF-IDF.

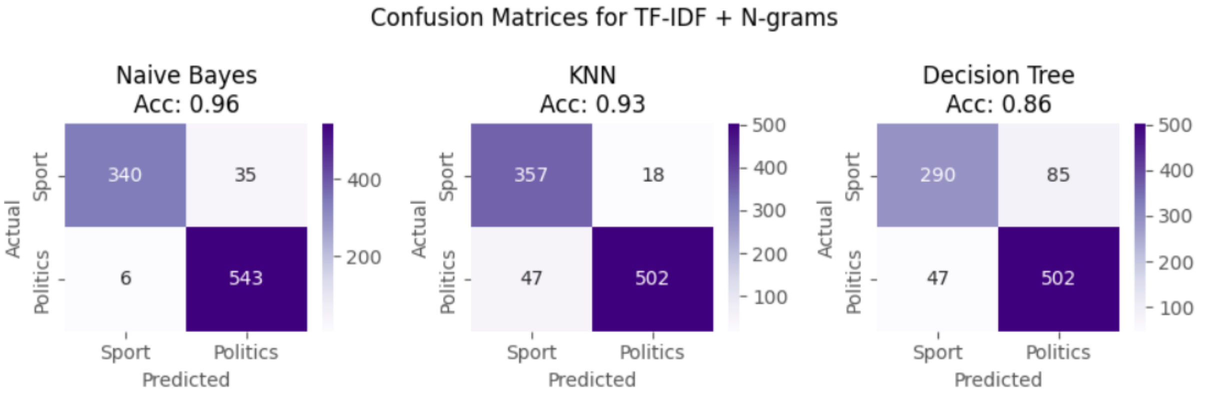

Figure 6: Confusion Matrices for TF-IDF + N-grams.

Analysis of Error Patterns

A granular inspection of these matrices reveals three critical insights:

1. **Naive Bayes Consistency:** Across all three feature sets, Naive Bayes maintained an exceptionally low error rate. This balance suggests the probabilistic boundary is robust and not easily swayed by noise.

2. **KNN and Feature Weighting:** The impact of TF-IDF on KNN is visually evident when comparing Figure 4 (BoW) and Figure 5 (TF-IDF). In the BoW model, KNN misclassified 65 ”Politics” documents as ”Sport.” After applying TF-IDF weighting, this specific error dropped to 49. This confirms that penalizing frequent, non-informative words helps the distance metric better separate the dense ”Politics” cluster from the ”Sport” cluster.

3. **Decision Tree and N-grams:** While Decision Trees were the weakest overall, the N-gram feature set (Figure 6) significantly boosted their ability to correctly identify ”Politics” documents. The number of correctly classified ”Politics” samples jumped to 502, compared to just 443 in the standard TF-IDF model.

7 System Limitations

Despite achieving high overall accuracy (95.7%), the experimental results highlight several critical limitations within the current system architecture:

1. **Decision Tree Instability:** The Decision Tree model exhibited significant overfitting, particularly in the ”Sport” category, where it misclassified 106 documents in the TF-IDF configuration (Figure 5). This high false-positive rate suggests the hierarchical rules are brittle and overly reliant on specific keywords that lack contextual nuance.

2. **Sensitivity of Distance Metrics:** The Bag-of-Words implementation for KNN suffered from a high error rate (65 misclassified ”Politics” documents, Figure 4).

This confirms the model's sensitivity to the ”Curse of Dimensionality,” where unweighted, high-frequency terms inflate distances between semantically similar documents.

3. **Feature Saturation:** Adding bigrams (N-grams) provided negligible improvement to the Naive Bayes classifier (accuracy dropped slightly to 0.9556), indicating a saturation of lexical features. The probabilistic model had already reached its learning capacity using single-word frequencies, suggesting that further gains would require semantic embeddings rather than just larger vocabularies.

8 GitHub Repository

Repository Link:

`https://github.com/sarita-mandal/nlu-assign1-prob4-document-classifier.git`