Sports vs. Politics: A Comparative Study of Machine Learning Techniques for Text Classification

Arman Gupta / B22CS014

February 15, 2026

1 Introduction

Topic categorization, which is similar to multiclass classification, groups content based on predefined themes. Text classification algorithms are at the heart of a variety of software systems that process text data at scale, enabling enhanced data organization, retrieval, and automated large-scale text analysis. Machine learning approaches leverage statistical models to classify text based on learned patterns, replacing older manual rule-based methods.

This project tackles a binary classification problem: distinguishing between sports-related and politics-related texts. By employing supervised learning techniques on a large corpus of text, we aim to mathematically map vocabulary distributions to these specific categories. This report outlines the data collection process, exploratory data analysis, feature representation, and a quantitative comparison of three distinct classification algorithms.

2 Data Collection

The primary source of data for this task is the renowned 20 Newsgroups dataset. The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). It has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.

To isolate our specific problem, we utilized the `sklearn.datasets.fetch_20newsgroups` function, which returns a list of the raw texts. We targeted specific subsets to form our two binary classes:

- **Sports:** `rec.sport.baseball`, `rec.sport.hockey`
- **Politics:** `talk.politics.misc`, `talk.politics.guns`, `talk.politics.mideast`

It is easy for a classifier to overfit on particular things that appear in the 20 Newsgroups data, such as newsgroup headers. To ensure our models learned the true semantic meaning of the text rather than memorizing trivial metadata (like sender email addresses or organization names), we programmatically stripped the data. In scikit-learn, you can do this by setting `remove=('headers', 'footers', 'quotes')`.

3 Dataset Description and Analysis

After the initial fetch and the removal of empty or null documents, our final curated dataset consisted of 4,494 text documents. The class distribution is as follows:

- **Sports Documents:** 1,933 samples
- **Politics Documents:** 2,561 samples

While the dataset exhibits a slight imbalance favoring the Politics class, it is well within the acceptable threshold for standard machine learning algorithms, preventing severe majority-class bias. Exploratory analysis reveals that Sports documents frequently contain specific entities (team names, scores, player names) and action verbs. In contrast, Politics documents possess broader, more abstract vocabulary and tend to have a higher average word count and more complex sentence structures.

4 Feature Representation

Machine learning models require numerical input; therefore, the raw text was converted into a mathematical representation. The text categorization pipeline fundamentally requires translating unstructured text into a structured, vector space model.

4.1 Bag of Words (BoW) and N-Grams

The Bag-of-Words (BoW) model represents text as a collection of word occurrences, ignoring grammar and word order. To capture essential local context (for example, understanding that "White House" or "home run" are distinct concepts rather than individual words), we expanded our vocabulary utilizing N-grams. Specifically, we used a combination of unigrams (single words) and bigrams (two-word sequences), capping our maximum feature space at 10,000 dimensions to ensure computational efficiency.

4.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF enhances BoW by weighting words based on their importance in a document relative to a collection of documents. This penalizes common English words and boosts domain-specific keywords. The calculation is twofold:

Term Frequency (TF) measures the frequency of a term within a document:

$$\text{TF}(t, d) = \frac{\text{Number of occurrences of } t \text{ in document } d}{\text{Total number of terms in document } d} \tag{1}$$

Inverse Document Frequency (IDF) measures the rarity of a term across a collection of documents:

$$\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents } N}{\text{Number of documents containing term } t}\right) \tag{2}$$

The final weight is the product: $\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$.

5 Machine Learning Techniques

Three classification algorithms were evaluated to determine the most effective method for high-dimensional, sparse text data.

5.1 Multinomial Naive Bayes (NB)

Naive Bayes is a probabilistic classifier based on Bayes' Theorem. It operates on the "naive" assumption of conditional independence between every pair of features given the class label. For text classification, it calculates the probability of a document belonging to a class based on the joint probabilities of the words it contains. It is highly effective for text classification because it easily handles high-dimensional representations and trains exceptionally quickly.

5.2 Logistic Regression (LR)

Despite its name, Logistic Regression is a linear classification model. It applies the logistic (sigmoid) function to a linear combination of features to predict the probability of the text belonging to the "Politics" class. It utilizes L2 regularization by default, which is highly beneficial in preventing overfitting across the 10,000 TF-IDF features.

5.3 Support Vector Machine (Linear SVC)

The Support Vector Machine attempts to find the optimal hyperplane that separates the two classes with the maximum margin. For text classification, where the number of

features often exceeds the number of samples, a Linear kernel (Linear SVC) is mathematically ideal. It focuses entirely on the data points closest to the decision boundary (the support vectors), making it highly robust against outliers.

6 Quantitative Comparisons

The dataset was split into an 80% training set and a 20% testing set (899 test samples). The models were evaluated based on overall accuracy, training time, precision, recall, and F1-score.

6.1 Overall Accuracy and Speed

All three models achieved an accuracy of over 95%, indicating that the TF-IDF feature extraction was highly successful in creating linearly separable feature spaces.

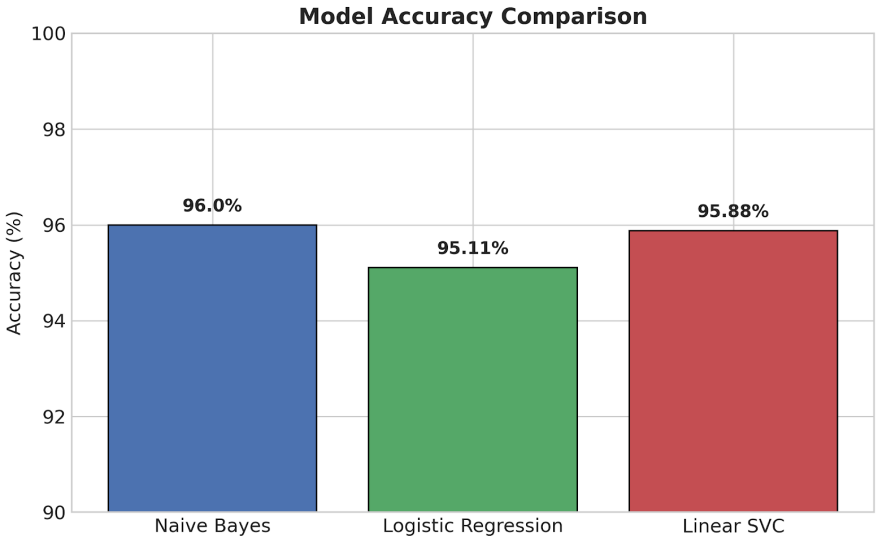

Figure 1: Comparison of Overall Model Accuracy

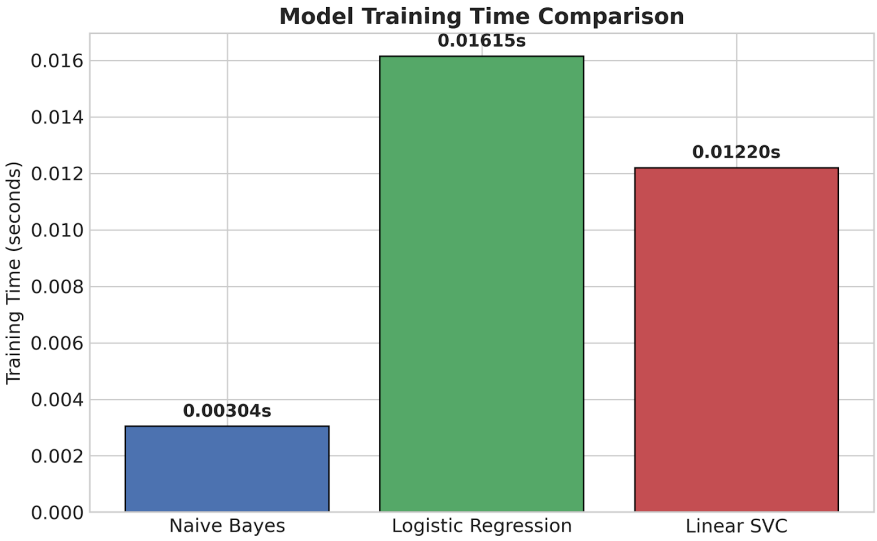

Figure 2: Comparison of Model Training Time

As shown in the figures above, Multinomial Naive Bayes emerged as the top performer in both categories. It achieved the highest accuracy at 96.00% and trained in an astonishing 0.003 seconds, making it computationally vastly superior to Logistic Regression (0.016s) and Linear SVC (0.012s).

6.2 Precision, Recall, and F1-Score

Because our dataset contains more Politics documents (2561) than Sports documents (1933), analyzing precision and recall provides deeper insights than accuracy alone.

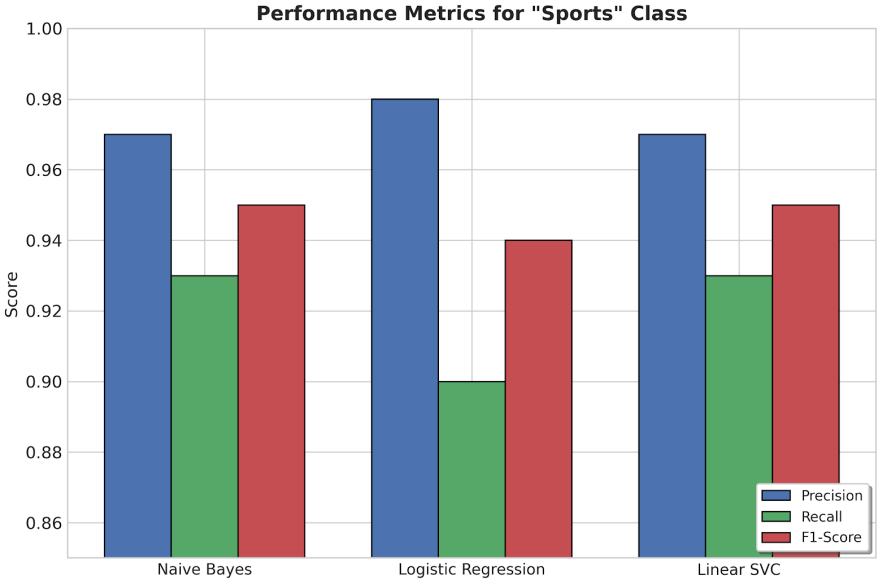

Figure 3: Performance Metrics for the Sports Class

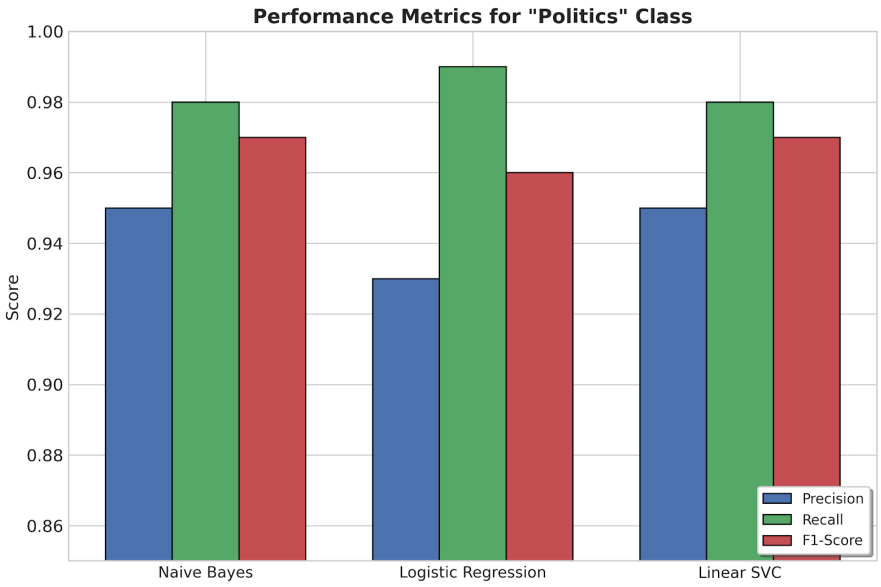

Figure 4: Performance Metrics for the Politics Class

The models exhibited a slight bias toward the majority class (Politics), evidenced by exceptionally high Recall scores for Politics (0.98 - 0.99) and lower Recall scores for Sports (0.90 - 0.93). The models act conservatively when predicting "Sports"; however, when they do, they are highly accurate (Precision of 0.97 - 0.98).

6.3 Confusion Matrices

The confusion matrices provide a direct look at the true positives, true negatives, false positives, and false negatives for each model evaluated on the 899 test documents.

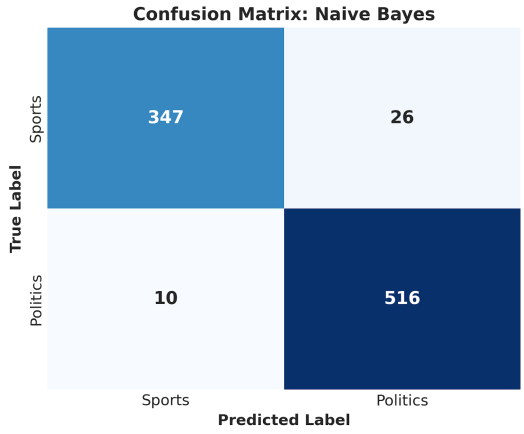

Figure 5: Confusion Matrix: Naive Bayes

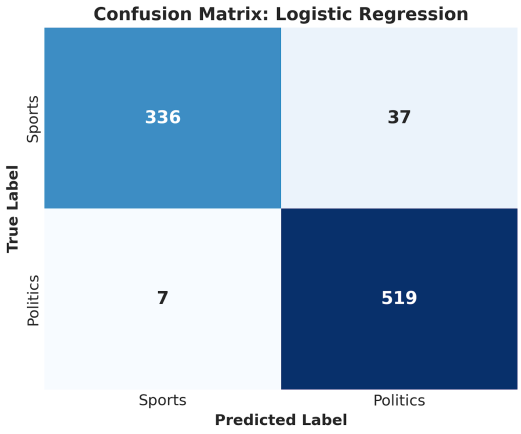

Figure 6: Confusion Matrix: Logistic Regression

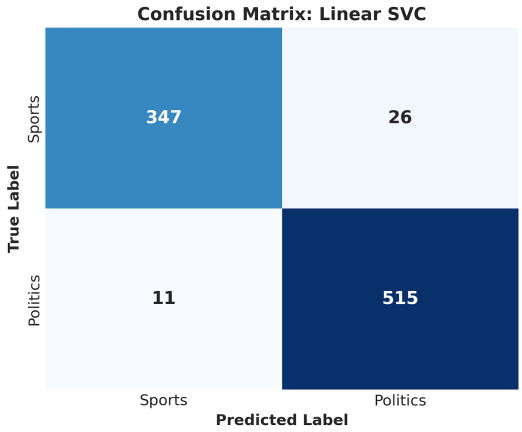

Figure 7: Confusion Matrix: Linear SVC

Logistic Regression struggled slightly more with False Negatives on the Sports class (37 misclassified), while Naive Bayes and Linear SVC performed nearly identically, only misclassifying 26 Sports documents as Politics.

7 Limitations and Future Work

While the system achieves high accuracy, it possesses inherent limitations:

1. **Context Blindness:** Because the system relies on TF-IDF and bigrams, it ignores deeper grammatical structures and semantic dependencies. For instance, sarcasm or complex metaphors (e.g., comparing a political race to a baseball game) could easily confuse the classifier.

2. **Vocabulary Freezing:** The model is incapable of dynamically understanding words it did not see during training (Out-of-Vocabulary words).

3. **Class Imbalance:** The slight bias toward the Politics class caused a measurable drop in recall for the Sports class.

Future iterations of this project should explore dense word embeddings (such as Word2Vec or GloVe) or Transformer-based neural networks like BERT, which read sentences bidirectionally to capture rich semantic context and significantly reduce the out-of-vocabulary problem. Furthermore, techniques such as SMOTE (Synthetic Minority Over-sampling Technique) could be employed to perfectly balance the training classes prior to modeling.

8 Conclusion

This study demonstrates that traditional machine learning techniques, when paired with robust feature extraction like TF-IDF, are highly effective for binary text classification. Multinomial Naive Bayes proved to be the optimal choice for this specific task, offering the highest accuracy (96.00%) alongside an incredibly low computational cost. By methodically preprocessing the text and comparing various models, we successfully engineered a pipeline capable of reliably distinguishing between Sports and Politics literature.