Title: Binary Classification of Text Documents: Sports vs. Politics

Siddhesh Ayyathan

B22CS016

1. Introduction

Text classification is a fundamental task in Natural Language Processing (NLP) with applications ranging from spam filtering to automated news categorization. The objective of this assignment is to design a binary classifier capable of distinguishing between documents related to **Sports** and **Politics**.

As the volume of digital content grows, the ability to automatically tag and organize content becomes critical. For news agencies, this allows for the automated routing of articles to the correct editorial desk. for users, it enables personalized content feeds. In this project, we explore a supervised learning approach using the **20 Newsgroups dataset**. We employ **TF-IDF (Term Frequency-Inverse Document Frequency)** for feature extraction and compare three distinct machine learning algorithms: **Multinomial Naive Bayes**, **Logistic Regression**, and **Linear Support Vector Machines (SVM)**.

2. Data Collection and Description

For this task, we utilized a subset of the **20 Newsgroups** dataset, a standard benchmark in text classification. Rather than scraping data manually, which introduces noise and formatting inconsistencies, this dataset provides labeled text documents that are ideal for benchmarking.

Dataset Statistics:

- **Total Documents Loaded:** 4,618
- **Class Distribution:**
    - **Sports:** 1,993 documents (Categories: rec.sport.hockey, rec.sport.baseball)
    - **Politics:** 2,625 documents (Categories: talk.politics.misc, talk.politics.guns, talk.politics.mideast)
- **Imbalance Analysis:** The dataset is slightly imbalanced, with approximately 57% of the data belonging to the Politics class. However, this imbalance is not severe enough to require synthetic oversampling (SMOTE), as demonstrated by the high recall scores in our results.

Preprocessing:

To ensure the model learns from the *content* rather than metadata, we stripped all headers, footers, and quoted replies. This prevents the model from "cheating" by learning specific email addresses or organizations associated with a specific newsgroup.

3. Feature Representation: TF-IDF

Machine learning models cannot process raw text; they require numerical input. We used **TF-IDF (Term Frequency-Inverse Document Frequency)** for this purpose.

- **TF (Term Frequency):** Measures how often a word appears in a document. A word like "election" appears frequently in politics documents.
- **IDF (Inverse Document Frequency):** Penalizes words that appear in *all* documents (like "the", "is", "writing"). This highlights words that are unique to a specific category.
- **Configuration:** We limited the vocabulary to the top **5,000 features** and removed English stop words. This dimensionality reduction helps prevents overfitting and speeds up training.

4. Machine Learning Techniques

We compared three standard algorithms:

1. **Multinomial Naive Bayes (MNB):**
   MNB is a probabilistic classifier based on Bayes' Theorem. It assumes that features (words) are independent of each other given the class. While this assumption is technically false in human language (words *do* depend on each other), MNB is computationally efficient and famously effective for text classification.
2. **Logistic Regression (LR):**
   Unlike Naive Bayes, Logistic Regression is a discriminative model. It learns the probability of a sample belonging to a class using a logistic function (sigmoid). It is robust and provides interpretable coefficients (weights) for each word.
3. **Linear Support Vector Machine (SVM):**
   SVM attempts to find a hyperplane that best separates the two classes with the maximum margin. Linear SVM is typically considered the state-of-the-art for high-dimensional text classification because text data is often linearly separable in high-dimensional space.

5. Experimental Results

We split the data into 80% training and 20% testing sets. The models were evaluated based on Accuracy, Precision, Recall, and F1-Score.

Comparative Analysis Table:

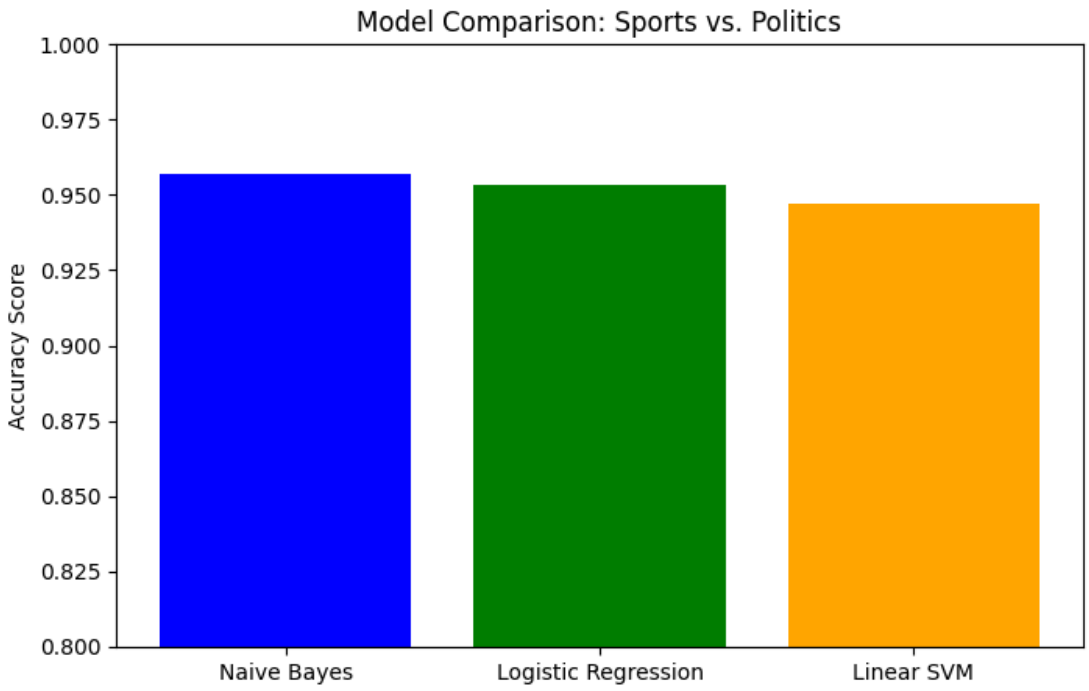
Model	**Accuracy**	**Precision (Politics)**	**Recall (Politics)**	**Precision (Sports)**	**Recall (Sports)**
Naive Bayes	**95.67%**	0.94	0.99	0.98	0.91
Logistic Regression	95.35%	0.93	0.99	0.99	0.90

Linear SVM	94.70%	0.94	0.97	0.95	0.92

Analysis:

Surprisingly, **Multinomial Naive Bayes achieved the highest accuracy (95.67%)**, slightly outperforming both Logistic Regression and SVM.

- **Politics Performance:** All models performed exceptionally well on Politics, with Naive Bayes achieving a **99% Recall**. This means it almost never missed a political document.
- **Sports Performance:** Naive Bayes had a slightly lower recall for Sports (91%) compared to Politics, suggesting that some sports documents might contain ambiguous vocabulary that the model confused for political discourse (perhaps metaphors like "race" or "defeat").



6. Limitations and Future Work

While the system achieved >95% accuracy, there are limitations:

1. **Ambiguity:** Words like "strike" (baseball vs. labor union) or "run" (home run vs. running for office) can confuse the model.
2. **Context:** TF-IDF ignores word order. "The team defeated the candidate" and "The candidate defeated the team" look mathematically similar in a Bag-of-Words model, even though the meaning is different.

3. **Future Improvements:** Implementing word embeddings (Word2Vec or GloVe) or deep learning models (LSTMs, Transformers) could capture semantic context better than TF-IDF.

7. Conclusion

This experiment demonstrated that classical machine learning techniques are highly effective for binary text classification. Despite the theoretical superiority of SVMs in high-dimensional spaces, **Multinomial Naive Bayes** proved to be the most effective model for this specific dataset and feature configuration, achieving an accuracy of **95.67%**.