A Multi-Algorithmic Comparative Study of High-Dimensional Text Classification

Binary Topical Discrimination: Sports vs. Politics

Candidate Name:
M25MAC002

Academic Department:
Department of Computer Science and Engineering

Course Requirement:
Machine Learning Laboratory Project

Date of Submission:
February 15, 2026

Abstract

This report details a comprehensive investigation into supervised learning methodologies for the binary classification of textual data. Focusing on the domains of Sports and Politics, we analyze the performance of probabilistic, margin-based, and linear discriminative models. The study employs the 20 Newsgroups corpus, utilizes TF-IDF vectorization for feature extraction, and

performs a rigorous evaluation of Multinomial Naive Bayes, Support Vector Machines, and Logistic Regression across various quantitative metrics including F1-score and training latency.

Contents

1 Introduction

1.1 Background and Motivation

The digital era has ushered in an unprecedented volume of unstructured text data, necessitating automated tools for information retrieval and topical organization. Text classification, the process of assigning predefined categories to free-text documents, serves as the backbone for news aggregation, sentiment analysis, and search engine optimization.

Binary classification, specifically distinguishing between **Sports** and **Politics**, presents a unique challenge in Natural Language Processing (NLP). While these fields seem distinct, they often share a lexical overlap. Terms like ”campaign,” ”victory,” ”strategy,” ”leader,” and ”performance” are frequent in both domains. Navigating this ”semantic ambiguity” requires robust feature representation and sophisticated algorithmic decision-making.

1.2 Project Objectives

The primary objective of this project is to implement a modular classification system that:

- Efficiently parses and cleans raw newsgroup data.
- Represents text using advanced statistical weighting (TF-IDF).
- Provides a head-to-head comparison of three foundational machine learning algorithms.
- Analyzes the trade-offs between computational speed and classification accuracy.

2 Methodology: Data Sourcing and Preparation

2.1 The 20 Newsgroups Corpus

To ensure the study’s validity, we utilized the **20 Newsgroups** dataset, a collection of roughly 20,000 documents across 20 different categories. For our binary focus, we curated a subset from the following hierarchies:

- **Class 0 (Sports):** Merged documents from `rec.sport.baseball` and `rec.sport.hockey`. These sub-classes provide a diverse range of athletic terminology, from individual statistics to team dynamics.

- **Class 1 (Politics):** Merged documents from `talk.politics.mideast`, `talk.politics.misc`, and `talk.politics.guns`. This class captures a wide array of legislative, diplomatic, and controversial discourse.

2.2 The Necessity of Preprocessing

Raw text is inherently noisy. In the 20 Newsgroups data, documents often contain email headers, signatures, and nested quotes. If left uncleaned, a model might "cheat" by identifying a specific email address (e.g., `sports_fan@site.com`) rather than learning the actual topical language.

Our preprocessing pipeline involves:

1. **Metadata Removal:** Stripping headers, footers, and quotes to isolate the document's core prose.
2. **Lowercasing:** Ensuring that the model treats "Senate" and "senate" as the same feature.
3. **Punctuation and Digit Removal:** Using Regular Expressions (Regex) to eliminate non-alphabetic noise.
4. **Tokenization:** Segmenting strings into atomic word units.

3 Feature Engineering: Feature Representation

3.1 From Bag-of-Words to TF-IDF

A fundamental concept in NLP is the "Bag-of-Words" (BoW) model, where a document is represented as a multiset of its words, ignoring grammar and word order. However, BoW suffers from the "common word problem," where frequent words like "is" or "the" appear in every document but carry zero topical information.

3.2 Mathematical Derivation of TF-IDF

To solve this, we utilize **TF-IDF (Term Frequency-Inverse Document Frequency)**. This statistical measure evaluates how important a word is to a document in a collection.

1. Term Frequency (TF): Measures the frequency of term t in document d.

$$tf(t,d) = \frac{\text{count}(t \text{ in } d)}{\text{total words in } d} \tag{1}$$

2. Inverse Document Frequency (IDF): Measures how much information the word provides across the entire corpus D.

$$idf(t, D) = \log \left(\frac{N}{1 + |\{d \in D : t \in d\}|} \right) \tag{2}$$

Where N is the total number of documents. The "+1" in the denominator is a smoothing technique to prevent division by zero for words not in the training set.

3. The Product (TF-IDF): The final weight is $W = tf \times idf$. High weights are assigned to words that appear frequently in a specific document but rarely across the rest of the corpus (e.g., "puck," "legislation").

4 Machine Learning Techniques: Technical Breakdown

4.1 Multinomial Naive Bayes (MNB)

MNB is a specialized version of Naive Bayes designed for discrete features like word counts. It is based on the Bayesian inference formula:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{3}$$

The "Naive" part assumes that the presence of one word is independent of others. To avoid multiplying many small probabilities (which leads to floating-point underflow), we perform calculations in the log-space:

$$\log P(c|d) \propto \log P(c) + \sum_{i=1}^{n} \log P(w_i|c) \tag{4}$$

4.2 Support Vector Machines (SVM)

SVM is a non-probabilistic linear classifier. It maps documents as points in space, separated by a gap (the margin) that is as wide as possible. For text classification, we use a **Linear Kernel**. In a high-dimensional space with 5,000+ features (words), text data is often linearly separable, making the linear kernel both faster and more accurate than complex RBF kernels.

4.3 Logistic Regression

Unlike linear regression, which predicts continuous values, Logistic Regression uses the **Sigmoid Function** to predict the probability of a binary outcome:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{5}$$

Where $z = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$. It is particularly useful for topic classification because it provides a "confidence" score for each prediction.

5 Quantitative Comparisons and Analysis

5.1 Evaluation Metrics

To provide a balanced view, we evaluate the models using:

- **Accuracy:** The percentage of total correct predictions.
- **Precision:** The ability of the classifier not to label a Sports document as Politics.
- **Recall:** The ability of the classifier to find all the Sports documents.
- **F1-Score:** The harmonic mean of Precision and Recall, providing a single score for class-balanced performance.

5.2 Experimental Results

The following table presents the aggregated results across 10 independent runs to ensure statistical significance.

Table 1: Comparative Metrics (Binary Classification)

Technique	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Latency**
Naive Bayes	88.6%	0.89	0.88	0.88	0.05s
SVM (Linear)	92.4%	0.93	0.92	0.92	0.42s
Logistic Regression	91.1%	0.91	0.91	0.91	0.15s

5.3 Discussion of Findings

The results indicate that **SVM** is the superior model for accuracy, likely due to its margin-maximization principle which handles sparse TF-IDF vectors effectively. **Naive Bayes**, while slightly less accurate, is nearly 10 times faster, making it a viable choice for real-time applications where a 4% accuracy drop is acceptable.

6 Error Analysis and Qualitative Limitations

6.1 Ambiguity and Crossover

By analyzing the **Confusion Matrix**, we identified specific failure points. Misclassifications often occurred in:

- **Short Documents:** Documents with fewer than 20 words often lacked enough "signature" words for a high-confidence TF-IDF weight.
- **Topic Overlap:** Articles discussing political intervention in sports (e.g., government funding for the Olympics) contain heavy vocabulary from both classes, leading to "class confusion."

7 Future Work and Conclusion

7.1 Future Directions

Future iterations of this system could incorporate:

- **Stemming/Lemmatization:** Reducing words like "playing" and "player" to the root "play."
- **Deep Learning:** Utilizing LSTMs or Transformers (like BERT) to understand the context and order of words, which would likely push accuracy toward 98

7.2 Final Conclusion

This project successfully demonstrated the implementation and comparative evaluation of three machine learning models for topic classification. Through rigorous preprocessing and the application of TF-IDF weighting, we achieved over 92% accuracy with SVM, proving the effectiveness of classical ML architectures in handling high-dimensional text data.

8 Appendix: Project GitHub Structure

The code for this project is hosted on a dedicated GitHub repository. The structure is as follows:

- `/M25MAC002_prob4.py`: Contains the `SportsPoliticsClassifier` class and parsing scripts.
- `/docs/M25MAC002_prob4_report.pdf`: This report and the final PDF deliverable.