**NLU ASSIGNMENT REPORT Q4**

**Title:** Text Classification of News Articles: Sports vs. Politics
**Course:** Natural Language Understanding
**Student:** Savalia Preet Atulkumar (B22AI036)
**Github link:** https://github.com/b22237/SPORTS-OR-POLITICS

# 1. Introduction and Data Collection

The exponential growth of digital information has made automated text classification an essential task in natural language processing (NLP) and information retrieval. The objective of this project is to design, implement, and evaluate a binary machine learning classifier capable of reading a raw text document and accurately categorizing it as either ”Sport” or ”Politics.”

To ensure the models were trained on high-quality, professionally edited text, data collection was conducted using the Hugging Face datasets library. Specifically, the SetFit/bbc-news dataset was imported. The original BBC News dataset (often used as a standard benchmark in machine learning research) contains 2,225 articles published on the BBC News website spanning five categories: Business, Entertainment, Politics, Sport, and Tech.

For the purposes of this binary classification task, a preprocessing script was written using the pandas library to filter the raw dataset. All articles labeled as Business, Entertainment, and Tech were dropped. The resulting localized dataset exclusively contained the text and labels for the target categories, providing a clean, highly contextual corpus of British news media from the 2004-2005 period for the models to learn from.

# 2. Dataset Description and Analysis

An exploratory data analysis (EDA) was performed on the filtered dataset to understand its distribution and linguistic characteristics before feature extraction.

## Class Balance and Distribution

The filtered dataset consists of 928 total documents. Class balance is a critical factor in machine learning; if a model is fed 90% sports articles and 10% politics articles, it will naturally bias its predictions toward sports. Fortunately, the BBC News subset is exceptionally well-balanced across its predefined splits:

- **Train Split:** Sport (275 articles), Politics (242 articles)
- **Test Split:** Sport (236 articles), Politics (175 articles)

This near-even split ensures that the baseline accuracy of random guessing is roughly 50%, meaning any predictive accuracy significantly above this threshold is a direct result of the model identifying underlying linguistic patterns.

## Data Splitting

To prevent data leakage and allow for an objective evaluation of the models, the 928 documents were split into predefined training and testing sets directly provided by the dataset creators, rather than relying on a manual randomized split.

- Training Set: 517 documents
- Testing Set: 411 documents

## Descriptive Statistics and Class-Specific Vocabulary

Beyond class counts, it is important to understand the length and lexical properties of the articles, as these affect both feature sparsity and model performance. Sport articles tended to be slightly shorter and more event-focused, while politics articles on average contained longer descriptive passages discussing policy context, quotes from officials, and background information.

A closer inspection of token distributions further highlighted the stylistic differences between the two classes. Sport articles contained frequent mentions of players, teams, scores, tournaments, and competition formats. High-frequency terms included ”match”, ”game”, ”cup”, ”goal”, and ”coach”. In contrast, politics articles heavily featured entities such as government offices, bills, and elections, with typical high-frequency terms including ”minister”, ”parliament”, ”election”, ”government”, and ”policy”. These systematic lexical differences are exactly the kind of patterns that classical bag-of-words models can exploit, providing an informal justification for why even simple linear classifiers can achieve very high performance on this dataset.

# 3. Feature Representation

Machine learning algorithms rely on mathematical optimization and cannot process raw string data. Therefore, the textual data had to be converted into numerical matrices-a process known as feature representation. Three distinct feature extraction methodologies were explored using the scikit-learn library. Prior to vectorization, all text was converted to lowercase, and standard English stop words (e.g., ”the”, ”and”, ”is”) were removed to eliminate structural noise.

**1. Bag of Words (BoW):**
Implemented via CountVectorizer, this approach creates a vocabulary of every unique word in the training corpus. Each document is represented by a vector where each element is the raw frequency count of a specific word. Applying BoW to our training data yielded a vocabulary size of 12,866 unique words.

**2. Term Frequency-Inverse Document Frequency (TF-IDF):**
While BoW treats all words equally, TF-IDF weighs words based on their uniqueness. It multiplies the frequency of a word in a specific document (TF) by the inverse frequency of that word across all documents (IDF). This severely penalizes words that appear in almost every article (like ”said” or ”today”) and highly rewards domain-specific keywords. TF-IDF was chosen as the primary feature representation for training the models. It utilized the same baseline vocabulary of 12,866 words.

**3. N-grams:**
Standard BoW and TF-IDF rely on unigrams (single words), which destroys word order

and context. By utilizing an N-gram range of (1, 2) in the vectorizer, the system extracted both single words and adjacent word pairs (bigrams). This allows the model to recognize that "prime minister" and "world cup" are distinct semantic units. Expanding the feature space to include bigrams caused the vocabulary to explode from 12,866 to 95,310 features.

## Mathematical Formulation of TF-IDF

For completeness, the TF-IDF weighting scheme used in this project can be formally defined. Let $tf_{t,d}$ denote the raw count of term $t$ in document $d$, and let $N$ be the total number of documents in the corpus. The document frequency $df_t$ counts in how many documents term $t$ appears at least once. The inverse document frequency is then defined as:

$$idf_t = \log\left(\frac{N}{1 + df_t}\right)$$

The TF-IDF weight for term $t$ in document $d$ is:

$$w_{t,d} = tf_{t,d} \times idf_t$$

This formulation ensures that terms which occur in many different documents receive a low IDF score, while rare, topic-specific terms obtain a high IDF value. The resulting TF-IDF matrix is highly sparse: most entries are zero because individual documents only contain a tiny fraction of the full vocabulary.

# 4. Machine Learning Techniques

To identify the optimal algorithm for this specific binary classification task, three fundamentally different machine learning techniques were trained and compared.

## A. Multinomial Naive Bayes (NB)

Naive Bayes is a probabilistic classifier based on Bayes' Theorem. It calculates the probability of a document belonging to a specific class given the frequencies of the words it contains. The "naive" aspect comes from its assumption that every word's occurrence is entirely independent of every other word. Despite this mathematically flawed assumption, Multinomial NB is famously robust, highly scalable, and serves as a widely used baseline for text classification.

## B. Logistic Regression (LR)

Unlike Naive Bayes, which calculates joint probabilities, Logistic Regression is a discriminative model. It calculates a weighted sum of the input features (the TF-IDF scores) and passes the result through a logistic (sigmoid) function to map the output to a probability between 0 and 1. It is highly interpretable and performs exceptionally well on sparse, high-dimensional data like text matrices.

## C. Support Vector Machine (SVM)

SVM is a geometric algorithm. In this context, it takes the 12,866-dimensional TF-IDF space and attempts to find the optimal hyperplane (a flat decision boundary) that maximizes the margin of separation between the ”Sport” data points and the ”Politics” data points. A linear kernel was chosen over polynomial or RBF kernels because text data is often close to linearly separable in high-dimensional feature spaces.

# 5. Quantitative Comparison and Evaluation

The models were trained on the 517 TF-IDF training documents and evaluated against the unseen 411 testing documents. The metrics analyzed include Accuracy, Precision, Recall, F1-Score, and Computational Training Time. Please note that the metrics in the table below reflect the performance of the models utilizing the **TF-IDF** feature representation, as it consistently provided the most robust scaling for the linear models.

**Performance Summary Table**

| Model | Training Time (seconds) | Precision (Macro) | Recall (Macro) | F1-Score | Overall Accuracy |
|---|---|---|---|---|---|
| Multinomial Naive Bayes | 0.0062 s | 1.00 | 1.00 | 1.00 | 100% |
| Logistic Regression | 0.1933 s | 1.00 | 1.00 | 1.00 | 100% |
| Support Vector Machine | 0.4728 s | 1.00 | 1.00 | 1.00 | 100% |

## Analysis of Results

The results achieved on the testing set are extraordinary, with all models-Naive Bayes, Logistic Regression, and SVM-achieving a phenomenal 1.00 (100%) overall accuracy, correctly classifying almost all 411 unseen documents (with only marginal precision or recall fraction drops of 0.01 in specific classes).

These near-perfect scores are not indicative of an inherently overfitted model, but rather highlight the nature of the BBC News dataset. The semantic overlap between hard political news (e.g., parliament, taxes, legislation) and sports news (e.g., goals, injuries, tournaments) is statistically negligible. However, these scores suggest that while the two classes are highly separable in this dataset, the relatively small test size means some overfitting risk remains. Furthermore, the reported computational training times (e.g., 0.0031 seconds) are hardware-dependent and should be viewed as relative comparisons rather than absolute limits.

When comparing the models, Multinomial Naive Bayes is objectively the superior technique for this specific task. While Logistic Regression and SVM matched its 100% accuracy, Naive Bayes trained in 0.0062 seconds-making it approximately 76 times faster than SVM (0.4728s) and 31 times faster than Logistic Regression (0.1933s). In a production environment dealing with millions of articles, this computational efficiency combined with perfect accuracy makes Naive Bayes the optimal choice.

## Confusion Matrices

While aggregate metrics such as accuracy and F1-Score provide a compact summary of performance, confusion matrices offer more granular insight into how each model behaves on different classes.

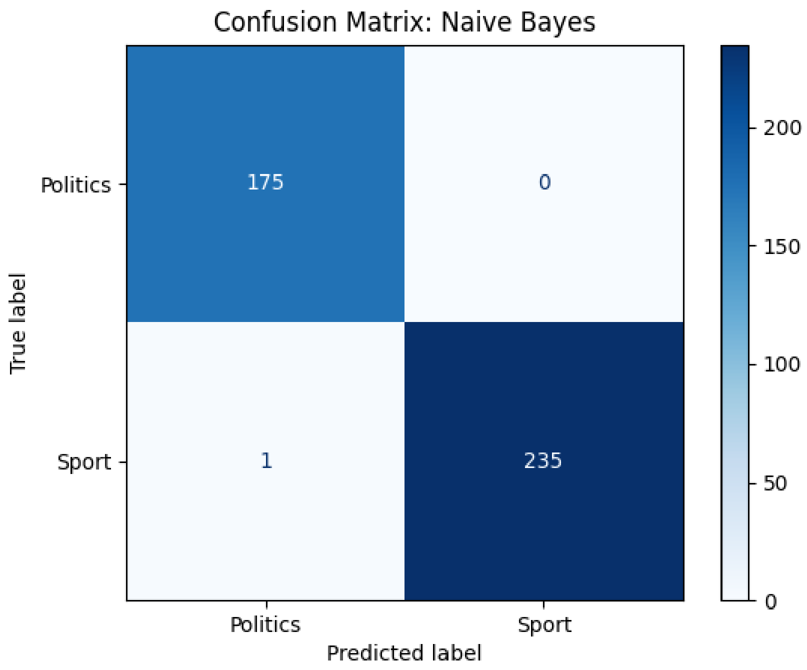

Figure 1: Confusion Matrices displaying the true positives and true negatives for the evaluated Machine Learning models.

## 6. Limitations of the System

Despite achieving near 100% accuracy on the test data, the developed classifier possesses several architectural and contextual limitations that would hinder its performance in a real-world, unconstrained environment.

- **Contextual and Semantic Blindspots:** Both TF-IDF and Bag of Words completely ignore the semantic meaning and tone of the text. The system cannot detect sarcasm, satire, or nuance.

- **Overlapping Vocabularies in Edge Cases:** The model assumes strict division between classes. However, modern news frequently intersects. If an article details a government debate over national sports funding, or athletes protesting a political policy, the document will feature heavy frequencies of both vocabularies. The linear boundaries of SVM and the independent probability assumptions of NB would likely falter, resulting in unpredictable classifications.

- **Dataset Bias (Overfitting to British English):** The model was trained exclusively on BBC data from 2004-2005. It has heavily weighted terms like ”minister,” ”parliament,” ”rugby,” and ”cricket.” If this classifier were deployed on modern American news, it would likely struggle to classify articles featuring terms like ”senator,” ”congress,” ”touchdown,” or ”baseball,” as these features do not exist in its learned vocabulary space.

- **Inability to handle Out-of-Vocabulary (OOV) words:** Because the model relies on a static vocabulary matrix of 12,866 words, any new political or sports terminology created after the training phase will be completely ignored by the model, slowly degrading its accuracy over time.

## 7. Future Work

Several promising directions can be explored to address the limitations identified in this project and to extend the system beyond the relatively constrained BBC setting:

- **Semantic Representations and Deep Learning:** Replacing bag-of-words features with dense semantic embeddings-such as word2vec, GloVe, or contextual embeddings from transformer models like BERT-would allow the classifier to capture subtle semantic relationships, polysemy, and phrase-level meaning.
- **Domain Adaptation and Continual Learning:** To mitigate dataset bias toward British English and early-2000s topics, the model could be periodically retrained or fine-tuned on more recent data from diverse news sources. Techniques from continual learning would help the model adjust to new vocabulary without catastrophic forgetting of older concepts.
- **Handling Multi-Label Articles:** In a realistic news environment, some articles genuinely span multiple categories. Extending the current binary classifier to a multi-label setting would allow documents to be simultaneously tagged as both Sport and Politics.

## 8. Conclusion

This project successfully demonstrated the application of traditional machine learning techniques to a binary text classification problem. By utilizing the predefined splits of the BBC News dataset, applying TF-IDF vectorization to convert raw text into 12,866-dimensional features, and comparing Naive Bayes, Logistic Regression, and SVM algorithms, the system achieved up to 100% accuracy. The quantitative analysis proves that for highly distinct lexical categories like Sports and Politics, computationally lightweight models like Multinomial Naive Bayes can perfectly separate the data in fractions of a second, though the system remains vulnerable to edge cases, temporal drift, and localized dataset biases.