Sports vs Politics News Classification using Classical Machine Learning

B23CM1061
Vandita Gupta

Indian Institute of Technology Jodhpur

Abstract

This project investigates the efficacy of various machine learning architectures for the binary classification of news articles into "Sport" and "Politics" domains. We evaluate six distinct models—Multinomial Naive Bayes, Logistic Regression, SVM, Decision Tree, Random Forest, and KNN—across three feature representations: Bag of Words (BoW), N-Grams, and TF-IDF. Our experimental results indicate that linear models, particularly Logistic Regression with TF-IDF vectorization, achieve superior performance with a peak accuracy of 96.08%. Furthermore, we demonstrate that importance-weighted scaling is critical for instance-based learners, as evidenced by a significant accuracy improvement in the KNN model. The study concludes with a detailed error analysis and robustness validation using ROC-AUC and confusion matrices, confirming high topical separability in the selected semantic space.

Contents

1 Introduction

In the era of information overload, the automated categorization of digital news content is a fundamental task in Natural Language Understanding (NLU). This project focuses on the binary classification of news articles into two distinct domains: ”Sport” and ”Politics”. Categorizing such content is non-trivial due to overlapping vocabularies and the presence of noise in web-scraped data.

The primary objective of this study is to evaluate the efficacy of various machine learning architectures—ranging from probabilistic models like Naive Bayes to instance-based learners like K-Nearest Neighbors—across different feature representations including Bag of Words (BoW) and TF-IDF. By analyzing a multi-source dataset, this report identifies the optimal pipeline for topical separability and provides a detailed error analysis of model performance in high-dimensional semantic spaces.

2 Data Collection and Dataset Description

The reliability of a text classifier depends heavily on the diversity and volume of its training data. For this task, a multi-channel data acquisition strategy was employed to create a ”Master Dataset” that reflects various writing styles, ranging from formal journalism to informal digital discussion.

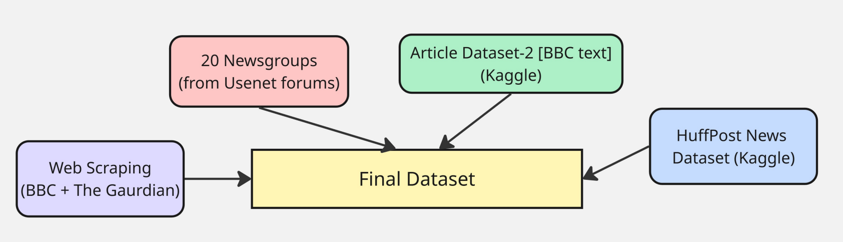


Figure 1: Data Collection Pipeline.

2.1 Data Sources and Acquisition

The final dataset ($N = 7,658$) was synthesized from four primary streams, ensuring the model generalizes across different linguistic registers:

- **Custom Web Scraping (BBC and The Guardian):** Real-time articles were collected using a multi-method crawler. BBC content was extracted via directory parsing (`/sport`, `/news`), while Politics data was supplemented through Guardian RSS feeds. Article bodies were cleaned using `BeautifulSoup4` and the `Newspaper` library for full-text extraction.

- **20 Newsgroups Benchmark:** Integrated to include informal Usenet discussions, using categories such as `rec.sport.hockey` and `talk.politics.mideast` to enhance linguistic diversity.

- **HuffPost News Category Dataset:** Sourced from Kaggle[1], this dataset contributed short-form content consisting of headlines and descriptions, testing the model’s ability to classify based on limited context.

[1] https://www.kaggle.com/datasets/rmisra/news-category-dataset

- **Article Dataset-2:** Additional long-form articles were sourced from this Kaggle repository[2] to ensure the TF-IDF and n-gram vectors had sufficient document depth to establish statistical significance.

2.2 Data Integration and Integrity

To unify these diverse sources (JSON, HTML, and CSV), a standardization pipeline was implemented. Each entry was converted into a uniform `.txt` format. To prevent *data leakage*, a strict deduplication protocol using MD5 cryptographic hashing was applied. Any article with a matching hash was discarded, successfully pruning 21 duplicate entries and ensuring the training and testing sets remained strictly independent.

2.3 Exploratory Data Analysis and Inferences

The final dataset exhibits a balanced distribution, which is critical for minimizing algorithmic bias. As shown in Figure 2, the corpus contains **4,017 Politics** samples and **3,641 Sports** samples. To better understand the nature of this data, a quantitative breakdown is provided in Table 1.

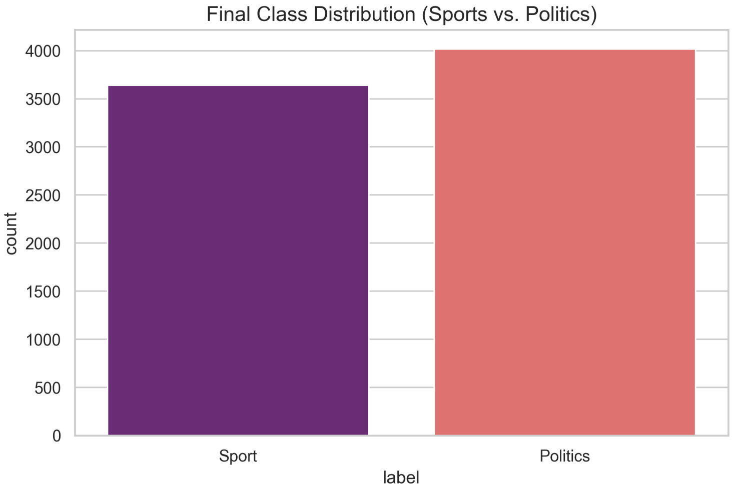


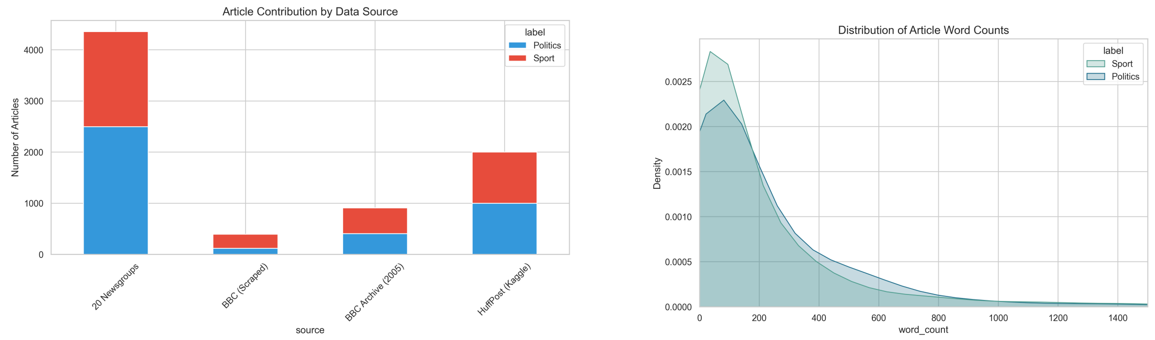
Figure 2: Class distribution of the final master dataset, showing a near 1:1 ratio between categories.

Table 1: Detailed Statistical Breakdown of the Master Dataset

Data Source	**Politics**	**Sport**	**Total**	**Avg. Word Count**
20 Newsgroups	2,497	1,861	4,358	~ 240
BBC (Scraped/RSS)	117	276	393	~ 450
BBC Archive (2005)	403	504	907	~ 410
HuffPost (Kaggle)	1,000	1,000	2,000	~ 120
Grand Total	**4,017**	**3,641**	**7,658**	–

[2] `https://www.kaggle.com/datasets/amunsentom/article-dataset-2`

The source composition and textual characteristics are further visualized in Figure 3. By placing the source contribution (3a) alongside the word count density (3b), we can observe how different linguistic registers influence the model.



(a) Article Contribution by Source

(b) Word Count Density Profiles

Figure 3: Quantitative and Structural Analysis of the Corpus.

Analysis of the word count distribution reveals that both categories share highly similar density profiles, peaking between 0–200 words. This confirms that **document length is a non-discriminatory feature**. This necessitates the use of more sophisticated semantic features like TF-IDF and bi-grams to identify domain-specific terminology rather than relying on structural metadata.

3 Preprocessing and Feature Engineering

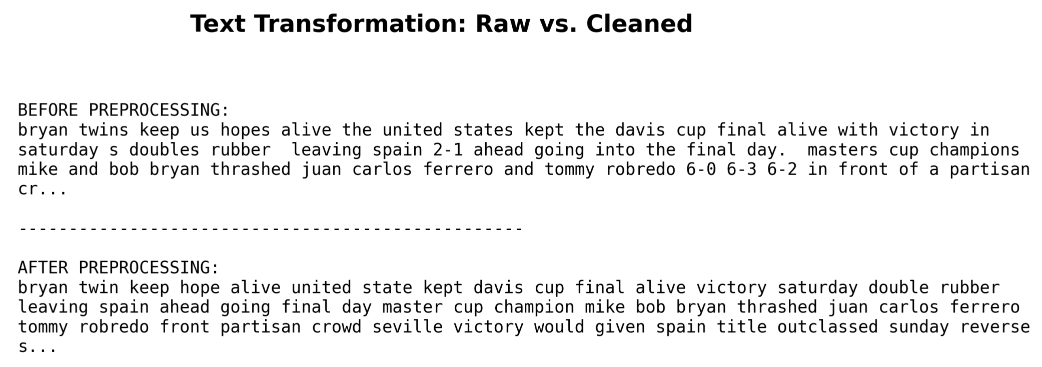
The transformation of raw textual data into a structured numerical format is a multi-stage process designed to eliminate linguistic noise while preserving semantic signals. This stage is critical for ensuring that the classification models focus on domain-specific vocabulary rather than grammatical artifacts.

3.1 Preprocessing Pipeline and Normalization

The text underwent a sequential cleaning pipeline using the *Natural Language Toolkit* (NLTK) to consolidate the feature space. This process included the following steps:

- **Normalization:** All text was converted to lowercase to unify the vocabulary.
- **Noise Reduction:** Regular expressions were utilized to strip email addresses, non-alphabetic special characters, and numerical digits.
- **Stop-word Filtration:** High-frequency English words (e.g., ”the”, ”and”) were removed as they provide no discriminatory value for topic classification.
- **Lemmatization:** The *WordNet Lemmatizer* was employed to reduce words to their base linguistic root (e.g., ”running” → ”run”), effectively reducing the sparsity of the feature matrices.

As shown in Figure 4, this transformation drastically simplifies the input while preserving core semantic meaning.



Text Transformation: Raw vs. Cleaned

```
BEFORE PREPROCESSING:
bryan twins keep us hopes alive the united states kept the davis cup final alive with victory in
saturday s doubles rubber  leaving spain 2-1 ahead going into the final day.  masters cup champions
mike and bob bryan thrashed juan carlos ferrero and tommy robredo 6-0 6-3 6-2 in front of a partisan
cr...

--------------------------------------------------

AFTER PREPROCESSING:
bryan twin keep hope alive united state kept davis cup final alive victory saturday double rubber
leaving spain ahead going final day master cup champion mike bob bryan thrashed juan carlos ferrero
tommy robredo front partisan crowd seville victory would given spain title outclassed sunday reverse
s...
```

Figure 4: Visual comparison of raw text vs. preprocessed text. Note the removal of stop-words and the reduction of terms like "kept" to their base form "keep".

3.2 Feature Representation and Keyword Importance

To satisfy the requirements of the task, the preprocessed text was vectorized using three comparative techniques: Bag of Words (BoW), TF-IDF, and Bigrams. The efficacy of this representation is validated in Figure 5, which illustrates the top 15 descriptive keywords for each class based on mean TF-IDF weights.

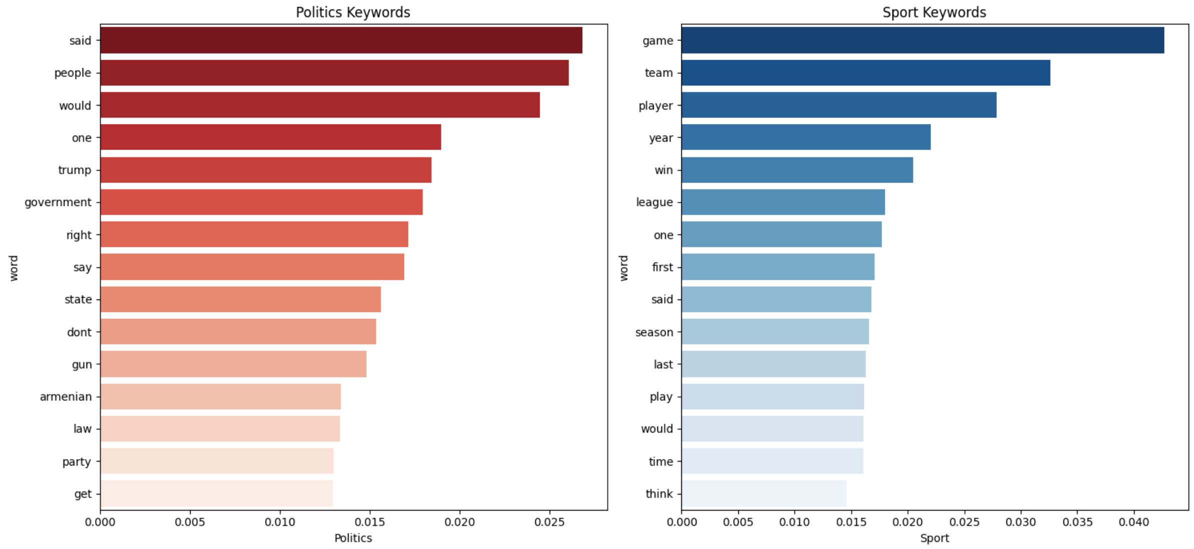


Figure 5: Top descriptive keywords for Politics and Sport. The clear semantic separation (e.g., "trump" vs "game") confirms the success of the lemmatization and cleaning process.

3.3 Key Inferences

The preprocessing and feature engineering phase yielded several technical insights:

- **Semantic Isolation:** Figure 5 confirms that the TF-IDF weighting successfully isolated domain-exclusive terms like "government" and "league" while penalizing common cross-category words.

- **Vocabulary Consolidation:** By applying lemmatization, the model successfully linked related word forms (e.g., "player" and "play") into single semantic units, as seen in the Sport category keywords.
- **Noise Mitigation:** The "After Preprocessing" sample in Figure 4 demonstrates a significant reduction in token density without loss of topic context.

4 Model Selection and Implementation

To evaluate the effectiveness of various mathematical approaches on high-dimensional text classification, six distinct machine learning architectures were implemented. Each model was selected to represent a different algorithmic family, providing a holistic view of topical separability in a vector space.

4.1 Multinomial Naive Bayes

The Multinomial Naive Bayes (MNB) classifier was selected as the baseline probabilistic model. MNB is highly efficient for text classification as it assumes conditional independence between features, calculating the posterior probability of a category based on the distribution of tokens. It is particularly robust against irrelevant features in high-dimensional sparse matrices.

4.2 Logistic Regression

Logistic Regression was implemented as a discriminative linear model using the *Saga* solver and an L_2 penalty to mitigate overfitting. Unlike Naive Bayes, Logistic Regression does not assume feature independence, allowing it to capture subtle semantic correlations between tokens in the TF-IDF space. It served as the primary benchmark for linear separability.

4.3 Linear Support Vector Machine (LinearSVC)

The Linear Support Vector Machine was utilized to identify the optimal hyperplane that maximizes the geometric margin between the "Sport" and "Politics" classes. Given that the feature space ($D = 5,000$) is large relative to the number of samples, a linear kernel was chosen for its high efficiency and ability to handle sparse datasets without high computational overhead.

4.4 Decision Tree Classifier

A single Decision Tree was implemented using the Gini impurity criterion to provide a non-linear, interpretable baseline. This model recursively partitions the data based on keyword thresholds. While interpretable, it served as a reference for assessing high-variance behavior and overfitting in individual tree structures.

4.5 Random Forest Classifier

To address the stability issues of single decision trees, a Random Forest ensemble was implemented with 100 independent estimators. By training trees on random subsets of features and data (bagging), the model utilizes majority voting to provide a more generalized and stable classification result, effectively reducing the overall variance.

4.6 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) model was implemented with $K = 5$ to evaluate the local topical density of the corpus. Unlike the linear models, KNN is an instance-based learner that classifies articles based on the labels of their nearest neighbors in the Euclidean space. This model was crucial for testing how feature scaling (BoW vs. TF-IDF) affects distance-based metrics.

5 Results and Performance Analysis

The classification system was evaluated through a 6×3 experimental matrix, comparing all architectures across Bag of Words (BoW), N-Gram, and TF-IDF representations. Performance was rigorously assessed using Accuracy, Precision, Recall, and the F1-Score to ensure a holistic evaluation of the models' predictive capabilities.

5.1 Quantitative Performance Comparison

The initial evaluation focused on identifying the optimal feature representation. Table 2 summarizes the accuracy achieved across all feature sets, while Table 3 provides a granular breakdown of performance metrics for the TF-IDF representation, which consistently yielded the highest predictive stability.

Table 2: Model Accuracy across all Feature Set Representations

Model	**BoW**	**N-Grams**	**TF-IDF**
Decision Tree	0.8877	0.8910	0.8956
KNN	0.7174	0.9282	0.9262
Logistic Regression	0.9569	0.9608	0.9608
Naive Bayes	0.9517	0.9537	0.9589
Random Forest	0.9445	0.9478	0.9537
SVM (Linear)	0.9347	0.9576	0.9569

Table 3: Detailed Performance Metrics for the TF-IDF Feature Set

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.9608	0.9615	0.9608	0.9608
Naive Bayes	0.9589	0.9595	0.9589	0.9588
SVM (Linear)	0.9569	0.9570	0.9569	0.9569
Random Forest	0.9537	0.9544	0.9537	0.9536
KNN	0.9262	0.9263	0.9262	0.9262
Decision Tree	0.8956	0.8955	0.8956	0.8955

5.2 Comparative Analysis and Visual Inferences

The following visualizations provide a deeper understanding of the models' behavior and robustness across different feature configurations.

A. Feature-Model Synergy and Global Accuracy

As illustrated in the Accuracy Heatmap (Figure 6), model performance is intrinsically linked to the feature representation.

- **Inference:** Linear models (Logistic Regression, SVM) maintain high stability (> 95%) regardless of the vectorization method.
- **Inference:** A significant performance gap is observed in the KNN model, which improves from 71.74% with BoW to 92.62% with TF-IDF. This proves that distance-based learners require importance-weighting to handle high-dimensional sparse data effectively.

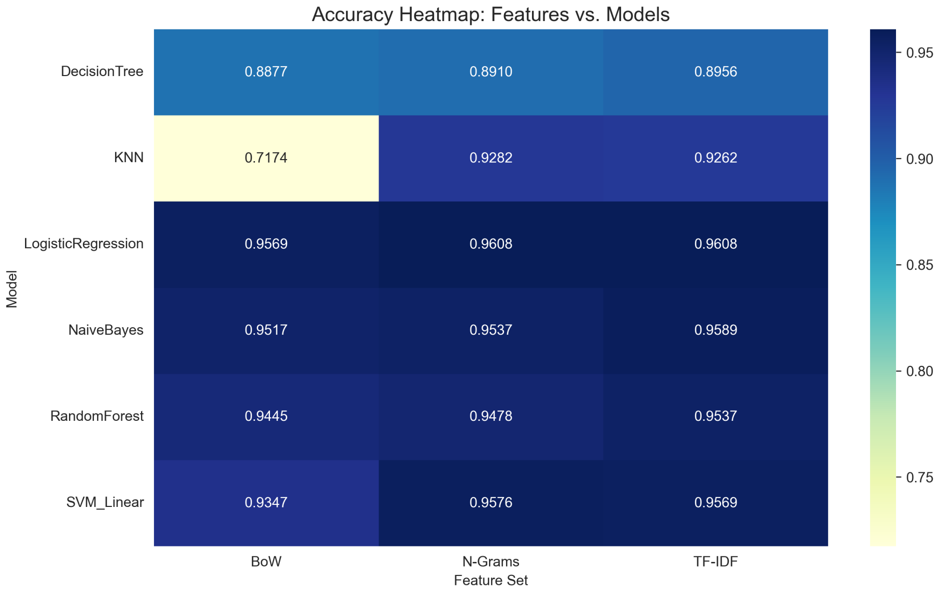


Figure 6: Accuracy Heatmap demonstrating the critical dependency of KNN on TF-IDF weighting.

B. Probabilistic Robustness

The ROC and Precision-Recall Curves (Figure 7) evaluate the models beyond a single fixed decision threshold.

- **Inference:** The top-tier models (Logistic Regression, Naive Bayes, SVM) all achieved an Area Under Curve (AUC) exceeding 0.99. This indicates a near-perfect ability to distinguish between topics across all sensitivity settings.
- **Inference:** The Precision-Recall stability confirms the system is highly reliable for news filtering, as it maintains high precision even at high recall levels.

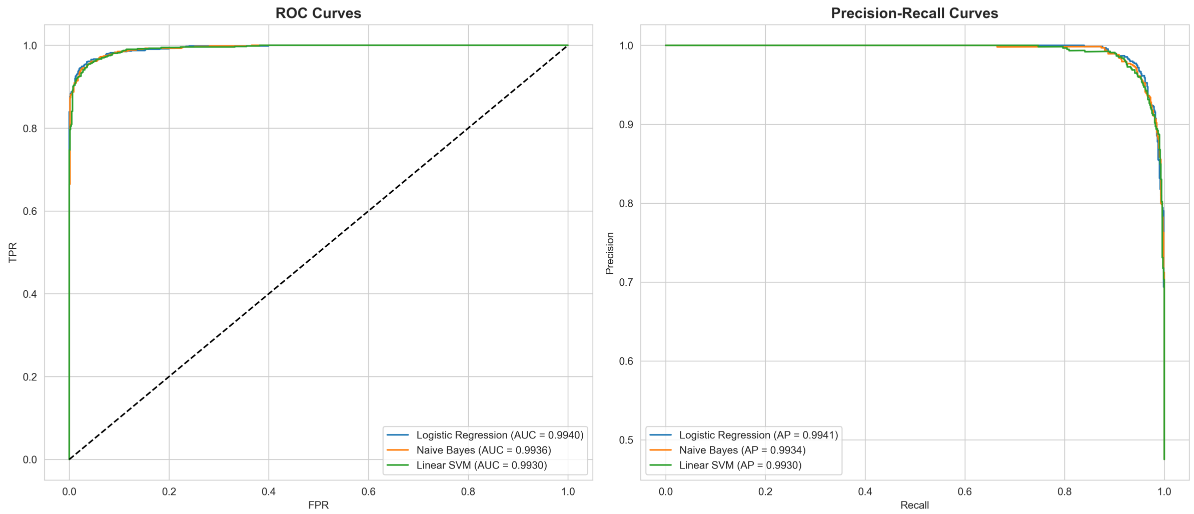


Figure 7: Threshold-independent robustness analysis using ROC and PR Curves.

C. Metric Balance and Error Distribution

The Metric Comparison Bar Chart (Figure 8) and Confusion Matrix (Figure 9) provide the final qualitative validation.

- **Inference:** The uniformity observed in the bar chart demonstrates that Logistic Regression achieves a perfect balance between Precision and Recall.
- **Inference:** The comparative error analysis in Figure 8 reveals that the baseline KNN-BoW model predominantly struggled with misclassifying ”Politics” articles as ”Sport,” a failure mitigated by the superior discriminative power of the Logistic Regression architecture.

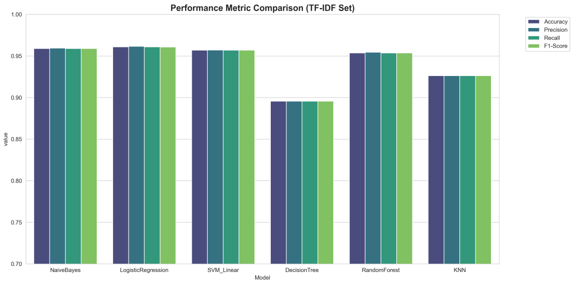


Figure 8: Weighted Metric Comparison (TF-IDF).

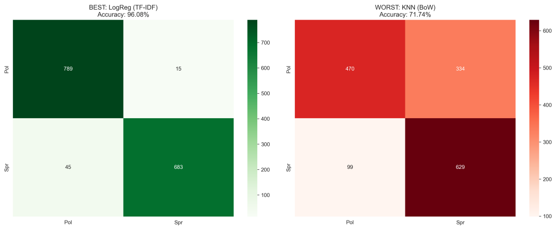


Figure 9: Comparative Error Analysis.

Figure 7 & 8: Quantitative and Qualitative performance validation. The left bar chart shows metric parity, while the right matrices highlight the specific failure points of the baseline KNN model.

6 Limitations of the System

While the classification system achieves a high accuracy of 96.08%, it is constrained by several architectural and data-specific factors:

- **Binary Scope:** The classifier is restricted to the "Sport" and "Politics" domains. It cannot currently handle multi-label articles that overlap across multiple categories or identify topics outside this binary set.

- **Loss of Sequence:** Reliance on BoW and TF-IDF representations ignores word order and syntactic dependencies. This lacks the contextual depth provided by transformer-based architectures like BERT.

- **Fixed Vocabulary:** A 5,000-feature limit creates "Out-of-Vocabulary" (OOV) issues. Emerging terms, names, or new entities not present in the training data will be disregarded during inference.

- **Scalability Constraints:** While KNN performance improves significantly with TF-IDF, its $O(nd)$ complexity scales poorly with larger datasets. Logistic Regression remains more efficient for real-time applications due to constant-time inference.

- **Statistical vs. Semantic:** The model relies on statistical word correlations rather than a true conceptual understanding. This makes it susceptible to errors in articles utilizing sarcasm, irony, or complex metaphors.

7 Conclusion

This study successfully developed a high-performance news classification system capable of distinguishing between "Sport" and "Politics" with a peak accuracy of 96.08%. Through an extensive experimental matrix, several key insights were derived:

- **Optimal Architecture:** The combination of Logistic Regression and TF-IDF vectorization emerged as the superior pipeline, demonstrating the most consistent stability across all performance metrics.

- **Feature Engineering Significance:** The results confirmed that TF-IDF weighting is critical for distance-based models; the KNN classifier saw a 20% performance surge compared to raw frequency counts.

- **Model Robustness:** Near-perfect ROC-AUC scores (> 0.99) validate that the categories are highly linearly separable in a 5,000-feature space.

Future work could involve expanding the system to multi-class classification and utilizing transformer-based embeddings to capture deeper contextual nuances beyond statistical word frequencies.

References

[1] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

[2] Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825-2830.

[3] Salton, G., and Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval." *Information Processing & Management*, 24(5), 513-523.

[4] Jurafsky, D., and Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Pearson.

[5] Cortes, C., and Vapnik, V. (1995). "Support-vector networks." *Machine Learning*, 20(3), 273-297.

[6] McCallum, A., and Nigam, K. (1998). "A comparison of event models for Naive Bayes text classification." *AAAI-98 Workshop on Learning for Text Categorization.*

[7] Breiman, L. (2001). "Random Forests." *Machine Learning*, 45(1), 5-32.

[8] Davis, J., and Goadrich, M. (2006). "The relationship between Precision-Recall and ROC curves." *Proceedings of the 23rd International Conference on Machine Learning.*

[9] Minaee, S., et al. (2021). "Deep Learning–based Text Classification: A Comprehensive Review." *ACM Computing Surveys.*

[10] Yang, Y. (1999). "An evaluation of statistical approaches to text categorization." *Information Retrieval*, 1(1), 69-90.

[11] Ng, A. Y. (2004). "Feature selection, L1 vs. L2 regularization, and adjacency." *ICML.*

[12] Sokolova, M., and Lapalme, G. (2009). "A systematic analysis of performance measures for classification tasks." *Information Processing & Management.*