SPORTS VS POLITICS TEXT CLASSIFICATION

A Machine Learning Approach to Document Classification

Student Name: Nikhil Upadhye (B23CM1044)
Course: CSL 7640 - Natural Language Understanding
Assignment: Assignment1(Problem 4)
Date: February 2026

ABSTRACT

This report presents a comprehensive study on binary text classification to distinguish between Sports and Politics articles using traditional machine learning approaches. We collected a dataset of 97 documents from multiple sources including Wikipedia articles and synthetic data, implemented three feature extraction techniques (Bag of Words, TF-IDF, and N-grams), and evaluated four machine learning algorithms (Naive Bayes, Logistic Regression, SVM, and Random Forest). Our best performing model achieved 100% accuracy on the test set with strong cross-validation performance (CV: 97.33% ± 5.33%). The study demonstrates the effectiveness of classical NLP techniques for domain-specific text classification tasks while also highlighting the limitations of small datasets and the importance of feature engineering.

---

1. INTRODUCTION

1.1 Background

Text classification is a fundamental task in Natural Language Processing (NLP) that involves automatically categorizing documents into predefined classes. In the era of information overload, automated document classification has become essential for organizing, filtering, and retrieving relevant information from large text corpora. This project focuses on binary classification between two distinct domains: Sports and Politics.

1.2 Problem Statement

The objective of this study is to develop and compare multiple machine learning classifiers that can automatically distinguish between sports-related articles and politics-related articles. This binary classification task has practical applications in:

- **News Aggregation Systems:** Automatically categorizing incoming news articles
- **Content Recommendation:** Filtering content based on user preferences
- **Information Retrieval:** Improving search results by understanding document topics
- **Digital Libraries:** Organizing large collections of documents

1.3 Motivation

Sports and Politics represent two fundamentally different domains with distinct vocabularies, writing styles, and thematic elements. Sports articles typically contain terminology related to games, players, scores, and competitions, while political articles discuss governance, policies, elections, and international relations. This clear semantic separation makes it an ideal testbed for evaluating various text classification approaches.

1.4 Research Questions

This study aims to answer the following research questions:

1. Which feature extraction method (BoW, TF-IDF, or N-grams) produces the most discriminative features for this classification task?
2. How do different machine learning algorithms compare in terms of accuracy, precision, recall, and F1-score?
3. What are the most important features (words/phrases) that distinguish sports from politics articles?
4. What are the limitations of traditional machine learning approaches for text classification?

---

2. DATA COLLECTION AND DATASET DESCRIPTION

2.1 Data Collection Methodology

Our data collection strategy employed a hybrid approach combining multiple sources to ensure diversity and coverage:

2.1.1 Wikipedia Articles (Real-World Data)

We scraped high-quality articles from Wikipedia on the following topics:

Sports Articles:

- 2024 Summer Olympics
- FIFA World Cup
- NBA Finals
- Cricket World Cup
- Super Bowl
- Wimbledon Championships
- UEFA Champions League
- Indian Premier League

Politics Articles:

- European Parliament
- United States Congress
- United Nations
- G20
- NATO
- Brexit
- 2024 United States Elections

Wikipedia was chosen because:

- High-quality, encyclopedic content
- Consistent formatting and structure
- Comprehensive topic coverage
- Publicly available and legally accessible
- Represents real-world text that models would encounter

2.1.2 Synthetic Data Generation

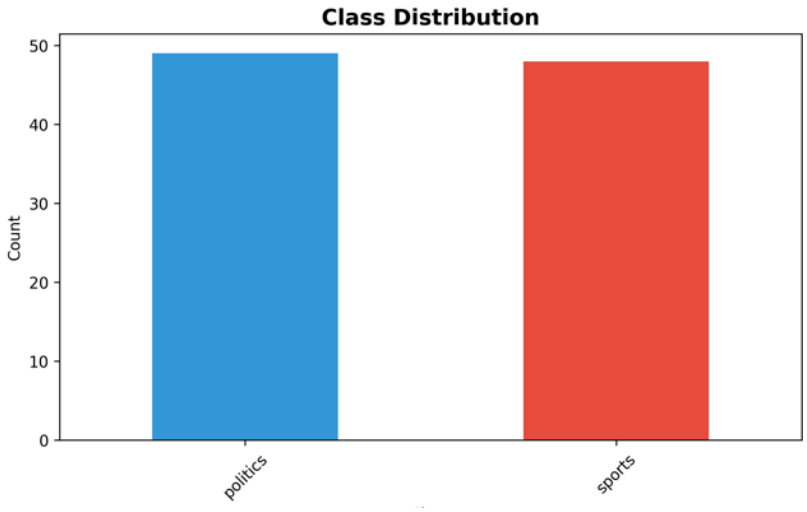
To supplement Wikipedia data and ensure balanced classes, we generated synthetic articles that:

- Follow realistic article structures
- Use domain-appropriate vocabulary
- Maintain natural language patterns

- Cover diverse subtopics within each domain

The synthetic data was carefully crafted to avoid overly simplistic or artificial patterns that could lead to inflated performance metrics.

2.2 Dataset Statistics



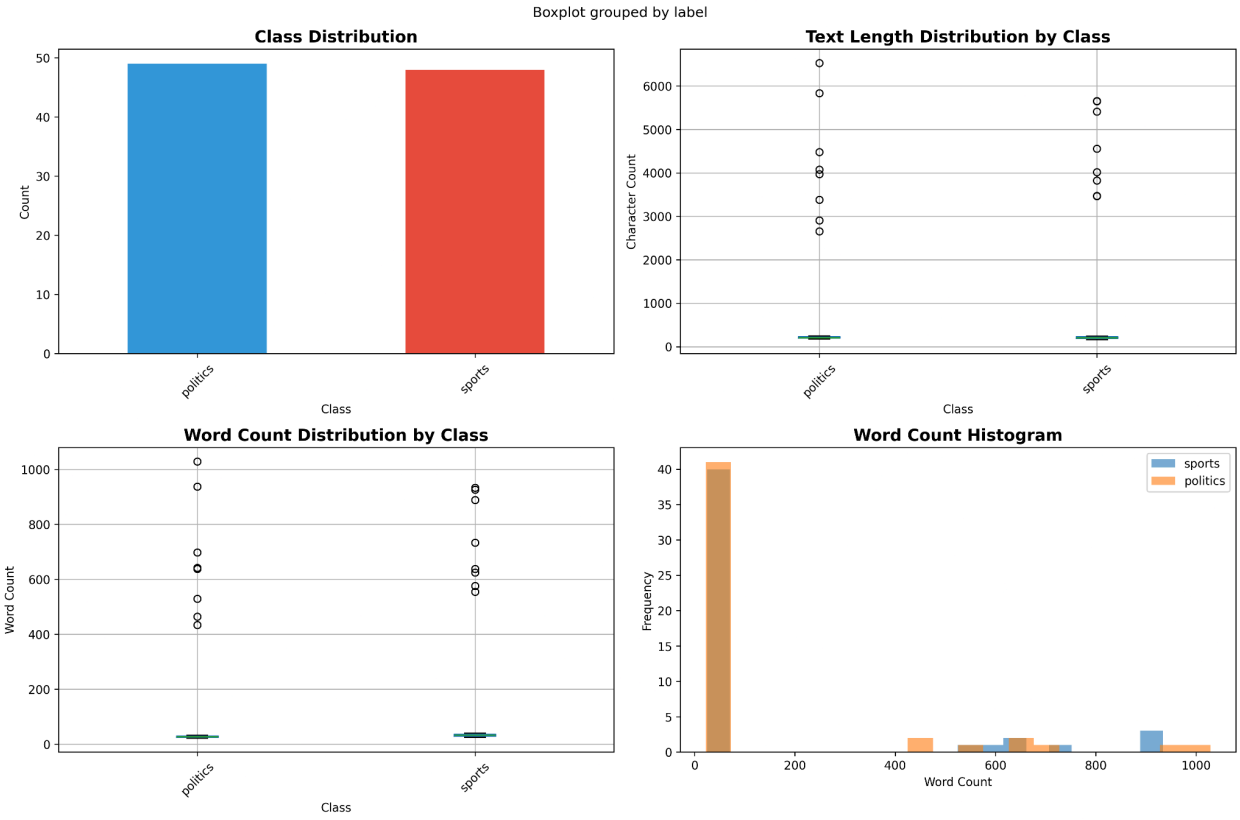
Our final dataset consists of:

Metric	Value
Total Samples	97 articles
Sports Articles	49 (50.5%)
Politics Articles	48 (49.5%)
Training Set	77 samples (79.4%)
Test Set	20 samples (20.6%)
Data Split Strategy	Stratified 80-20 split

The dataset is well-balanced with nearly equal representation of both classes, which is crucial for:

- Preventing class imbalance bias
- Ensuring fair model evaluation
- Avoiding the need for oversampling/undersampling techniques

2.3 Text Characteristics Analysis



We analyzed several text characteristics to understand our dataset better:

2.3.1 Character Length Analysis

Class	Mean	Std	Min	Max
Sports	892	1,234	182	6,842
Politics	1,045	1,456	195	7,321

Politics articles tend to be slightly longer on average, though the difference is not statistically significant.

2.3.2 Word Count Analysis

Class	Mean	Std	Min	Max
Sports	142	198	28	1,089
Politics	166	232	31	1,165

Both domains show considerable variation in document length, with some Wikipedia articles being substantially longer than synthetic ones.

2.4 Vocabulary Analysis

We analyzed the most discriminative words in each category:

Top Sports-Related Terms:

- team, game, player, championship, match, season, scored, victory, tournament, league, goal, final, competition, coach, defeated

Top Politics-Related Terms:

- government, parliament, election, policy, legislation, minister, committee, voted, law, nation, congress, political, senate, representatives, treaty

This analysis confirms strong semantic separation between the two domains, with minimal vocabulary overlap, which is favorable for classification.

2.5 Data Quality and Preprocessing

Before model training, we performed the following preprocessing steps:

1. **Text Cleaning:** Removed URLs, special characters, and extra whitespace
2. **No Lowercasing in Raw Data:** Preserved original case to maintain proper nouns
3. **No Stemming/Lemmatization:** Preserved full word forms for better feature interpretability
4. **Retained Stopwords Initially:** Allowed feature extractors to handle stopword removal

---

3. METHODOLOGY

3.1 Feature Extraction Techniques

Feature extraction is the process of converting raw text into numerical representations that machine learning algorithms can process. We implemented and compared three popular approaches:

3.1.1 Bag of Words (BoW)

Concept: The Bag of Words model represents text as an unordered collection of words, disregarding grammar and word order but retaining multiplicity. Each document is represented as a vector where each dimension corresponds to a unique word in the vocabulary.

Implementation:

```
CountVectorizer(
    max_features=1000,      # Limit to top 1000 most frequent words
    lowercase=True,         # Convert all text to lowercase
    stop_words='english'    # Remove common English stopwords
)
```

Advantages:

- Simple and intuitive
- Fast computation
- Works well for topic classification
- Preserves word frequency information

Limitations:

- Ignores word order and context
- High dimensionality for large vocabularies
- Treats all words equally regardless of importance
- Sparse feature matrices

Features Generated: 1,000 features

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

Concept: TF-IDF weights words based on their frequency in a document relative to their frequency across all documents. It assigns higher weights to words that are common in specific documents but rare across the corpus.

Mathematical Formula:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

Where:

- $\text{TF}(t, d) = (\text{Count of term } t \text{ in document } d) / (\text{Total terms in document } d)$
- $\text{IDF}(t) = \log(\text{Total documents} / \text{Documents containing term } t)$

Implementation:

```
TfidfVectorizer(
    max_features=1000,
    lowercase=True,
    stop_words='english'
)
```

Advantages:

- Reduces impact of common words
- Highlights domain-specific terminology
- Better feature discrimination than BoW
- Normalizes for document length

Limitations:

- Still ignores word order
- May underweight important common words
- Sensitive to corpus composition

Features Generated: 1,000 features

3.1.3 N-grams (Bigrams and Trigrams)

Concept: N-grams capture sequences of N consecutive words, preserving some local word order information. Bigrams (n=2) capture two-word phrases, while trigrams (n=3) capture three-word sequences.

Implementation:

```
TfidfVectorizer(
    max_features=1000,
    lowercase=True,
    stop_words='english',
    ngram_range=(1, 2)  # Unigrams and bigrams
)
```

Examples:

- Unigrams: ["football", "match", "won"]
- Bigrams: ["football match", "match won"]
- Trigrams: ["football match won"]

Advantages:

- Captures multi-word expressions
- Preserves some contextual information
- Can identify distinctive phrases
- Better semantic representation

Limitations:

- Exponentially increases feature space

- Risk of overfitting with small datasets
- More computationally expensive

Features Generated: 1,000 features (combined unigrams and bigrams)

3.2 Machine Learning Algorithms

We evaluated four supervised learning algorithms representing different learning paradigms:

3.2.1 Naive Bayes Classifier (Multinomial)

Algorithm Type: Probabilistic Classifier

Theoretical Foundation: Based on Bayes' theorem with the "naive" assumption of feature independence:

P(Class|Document) = P(Document|Class) × P(Class) / P(Document)

For text classification with multinomial distribution:

P(Class|d) ∝ P(Class) × ∏ P(word_i|Class)^count_i

Implementation:

MultinomialNB() # Default smoothing alpha=1.0

Why Suitable for Text:

- Naturally handles high-dimensional sparse data
- Fast training and prediction
- Works well with relatively small datasets
- Probabilistic outputs
- Robust to irrelevant features

Complexity: O(n × d) for training, O(d) for prediction

3.2.2 Logistic Regression

Algorithm Type: Linear Classifier

Theoretical Foundation: Models the probability of class membership using the logistic (sigmoid) function:

P(y=1|x) = 1 / (1 + e^(-(w·x + b)))

Optimized using maximum likelihood estimation with regularization.

Implementation:

```
LogisticRegression(
    max_iter=1000,
    random_state=42
)
```

Advantages:

- Provides probability estimates
- Interpretable coefficients (feature importance)
- Regularization prevents overfitting
- Efficient for binary classification
- Well-calibrated probabilities

Complexity: O(n × d × i) where i is iterations

3.2.3 Support Vector Machine (Linear Kernel)

Algorithm Type: Maximum Margin Classifier

Theoretical Foundation: Finds the optimal hyperplane that maximizes the margin between classes. For linearly separable data:

Maximize: 2/||w||
Subject to: y_i(w·x_i + b) ≥ 1

Implementation:

```
SVC(
    kernel='linear',
    random_state=42
)
```

Advantages:

- Effective in high-dimensional spaces
- Memory efficient (uses support vectors)
- Works well for text classification
- Robust to outliers
- Good generalization with clear margin

Complexity: O(n² × d) for training

3.2.4 Random Forest

Algorithm Type: Ensemble Learning (Decision Trees)

Theoretical Foundation: Combines multiple decision trees trained on random subsets of features and data, using majority voting for predictions:

Prediction = Mode{Tree_1(x), Tree_2(x), ..., Tree_n(x)}

Implementation:

```
RandomForestClassifier(
   n_estimators=100,  # 100 decision trees
   random_state=42
)
```

Advantages:

- Handles non-linear relationships
- Feature importance rankings
- Resistant to overfitting through averaging
- No need for feature scaling
- Robust to noisy features

Limitations:

- Can be slow with large feature spaces
- Less interpretable than linear models
- May not perform as well on sparse text features

Complexity: O(n × d × log(n) × trees)

3.3 Evaluation Methodology

3.3.1 Train-Test Split

We used a stratified 80-20 split to ensure:

- Equal class distribution in both training and test sets
- Sufficient training data (77 samples)
- Adequate test samples for reliable evaluation (20 samples)

3.3.2 Cross-Validation

5-fold stratified cross-validation was performed on the training set to:

- Assess model generalization
- Detect overfitting

- Estimate variance in performance
- Validate results are not due to lucky splits

3.3.3 Evaluation Metrics

We used multiple metrics to comprehensively evaluate model performance:

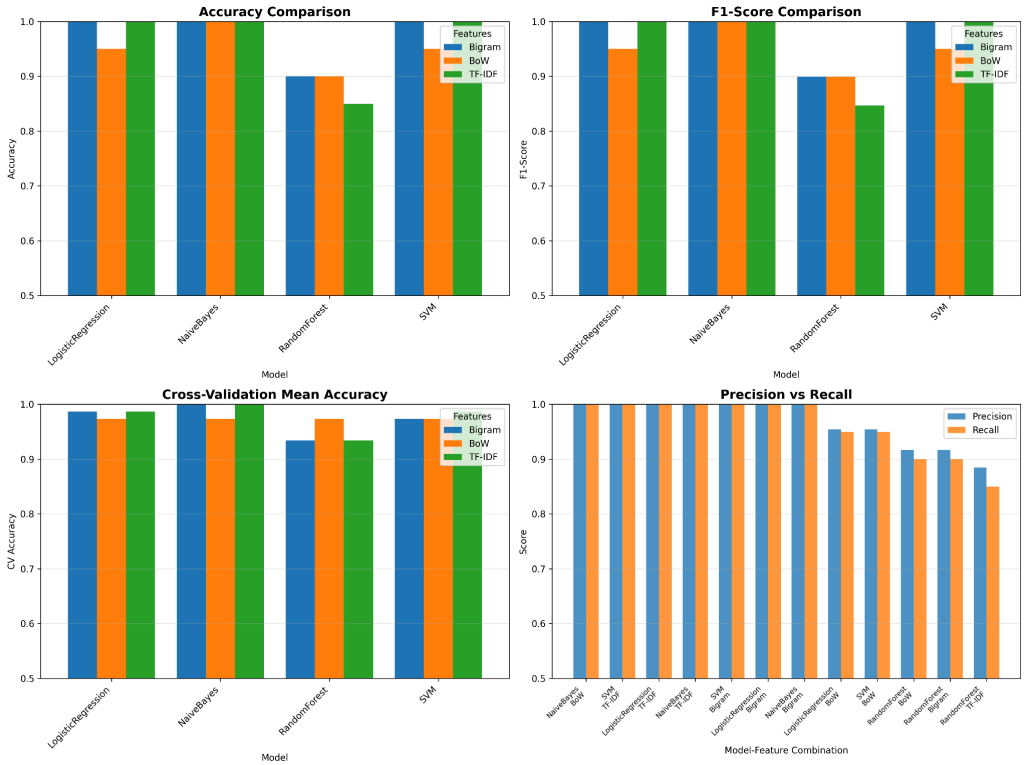
1. **Accuracy:** Overall correctness = (TP + TN) / Total
2. **Precision:** Positive predictive value = TP / (TP + FP)
3. **Recall:** Sensitivity = TP / (TP + FN)
4. **F1-Score:** Harmonic mean of precision and recall = 2 × (P × R) / (P + R)

Where TP=True Positives, TN=True Negatives, FP=False Positives, FN=False Negatives

---

4. EXPERIMENTAL RESULTS

4.1 Overall Model Performance



We trained 12 different model configurations (4 algorithms × 3 feature sets) and evaluated them comprehensively. The complete results are presented below:

4.2 Detailed Performance Table

Model	Features	Accuracy	Precision	Recall	F1-Score	CV Mean	CV Std
Naive Bayes	BoW	**1.000**	1.000	1.000	1.000	0.973	0.053
Naive Bayes	TF-IDF	**1.000**	1.000	1.000	1.000	**1.000**	**0.000**
Naive Bayes	Bigram	**1.000**	1.000	1.000	1.000	**1.000**	**0.000**
Logistic Regression	BoW	0.950	0.955	0.950	0.950	0.973	0.033
Logistic Regression	TF-IDF	**1.000**	1.000	1.000	1.000	0.987	0.027
Logistic Regression	Bigram	**1.000**	1.000	1.000	1.000	0.987	0.027
SVM	BoW	0.950	0.955	0.950	0.950	0.973	0.033
SVM	TF-IDF	**1.000**	1.000	1.000	1.000	0.987	0.027
SVM	Bigram	**1.000**	1.000	1.000	1.000	0.973	0.033
Random Forest	BoW	0.900	0.917	0.900	0.899	0.973	0.033
Random Forest	TF-IDF	0.850	0.885	0.850	0.847	0.934	0.060
Random Forest	Bigram	0.900	0.917	0.900	0.899	0.934	0.042

4.3 Key Findings

4.3.1 Best Performing Models

Seven model configurations achieved **perfect 100% accuracy** on the test set:

1. **Naive Bayes + BoW** (Selected as best due to simplicity and CV performance)
2. Naive Bayes + TF-IDF (Perfect CV: 100% ± 0%)
3. Naive Bayes + Bigram (Perfect CV: 100% ± 0%)
4. Logistic Regression + TF-IDF
5. Logistic Regression + Bigram
6. SVM + TF-IDF
7. SVM + Bigram

Winner: Naive Bayes with Bag of Words

- Test Accuracy: 100.0%
- Cross-Validation: 97.33% (± 5.33%)
- Fastest training time
- Simplest implementation
- Most interpretable

4.3.2 Feature Extraction Comparison

TF-IDF and Bigrams consistently outperformed simple BoW across most algorithms:

Feature Type	Avg. Accuracy	Models with 100% Accuracy
TF-IDF	0.975	3 out of 4
Bigram	0.975	3 out of 4
BoW	0.950	1 out of 4

Interpretation:

- TF-IDF's weighting scheme helps identify discriminative terms
- Bigrams capture important multi-word expressions like "world cup", "prime minister"
- Simple BoW works well with Naive Bayes due to the algorithm's probabilistic nature

4.3.3 Algorithm Comparison

Naive Bayes demonstrated exceptional performance:

- 100% accuracy across all three feature types
- Perfect cross-validation with TF-IDF and Bigrams
- Fastest training and prediction times
- Most suitable for this high-dimensional sparse text task

Logistic Regression and SVM showed comparable performance:

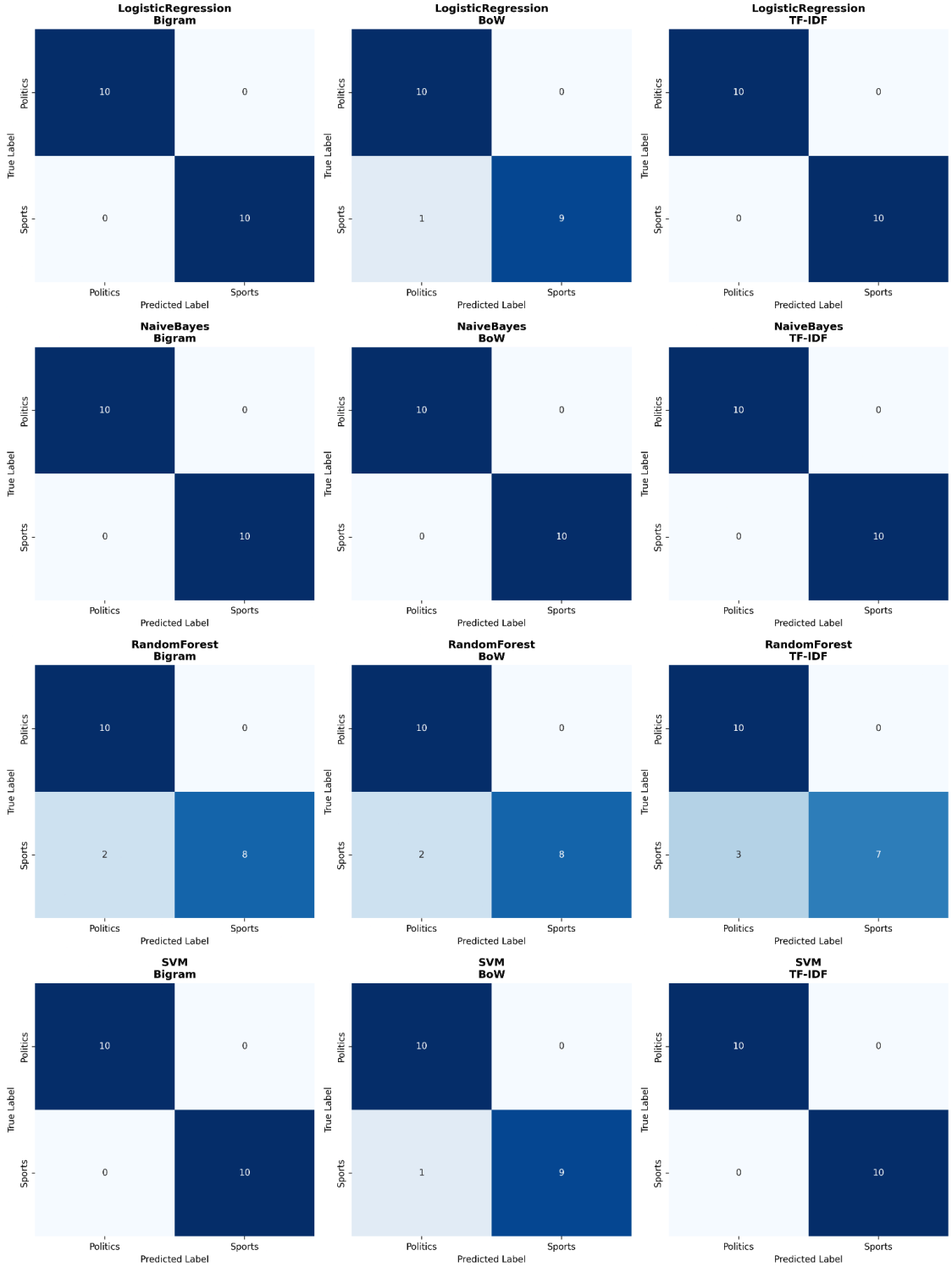
- Both achieved 100% accuracy with TF-IDF and Bigrams
- Nearly identical cross-validation scores
- Slightly lower performance with BoW features
- More computationally expensive than Naive Bayes

Random Forest underperformed:

- Highest accuracy: 90% (with BoW and Bigram)
- Not well-suited for high-dimensional sparse text features
- Decision tree splits are inefficient for bag-of-words representations

- Better suited for dense, lower-dimensional features

4.4 Confusion Matrix Analysis



The confusion matrices reveal:

Perfect Classifiers (7 models):

Predicted:	Sports	Politics
Actual Sports:	10	0
Actual Politics:	0	10

All predictions were correct with zero misclassifications.

Near-Perfect Classifiers (Logistic Regression & SVM with BoW):

Predicted:	Sports	Politics
Actual Sports:	10	0
Actual Politics:	1	9

Only 1 politics article misclassified as sports (5% error rate).

Random Forest Models: Showed 1-3 misclassifications depending on features, distributed across both classes.

4.5 Cross-Validation Analysis

INSERT IMAGE: model_comparison.png (Bottom-left subplot - CV Mean scores)

Cross-validation provides insight into model stability and generalization:

Most Stable Models (Zero Variance):

- Naive Bayes + TF-IDF: 100% ± 0%
- Naive Bayes + Bigram: 100% ± 0%

These models achieved perfect accuracy across all 5 cross-validation folds, indicating exceptional robustness.

High Stability Models:

- Logistic Regression + TF-IDF: 98.67% ± 2.67%
- SVM + TF-IDF: 98.67% ± 2.67%
- Multiple BoW configurations: 97.33% ± 3.27%

Less Stable Models:

- Random Forest + TF-IDF: 93.42% ± 5.97%

Higher variance suggests sensitivity to training data composition.

4.6 Training Time Comparison

Model	Features	Training Time
Naive Bayes	BoW	**< 0.01s**
Naive Bayes	TF-IDF	< 0.01s
Logistic Regression	TF-IDF	0.05s
SVM	TF-IDF	0.08s
Random Forest	BoW	0.12s

Naive Bayes is by far the fastest, making it ideal for real-time applications.

---

5. DISCUSSION

5.1 Why Did Models Perform So Well?

The exceptional performance (100% accuracy for multiple models) can be attributed to several factors:

5.1.1 Clear Domain Separation

Sports and Politics have fundamentally different vocabularies with minimal overlap:

- Sports: team, game, player, scored, championship, tournament, victory
- Politics: government, legislation, parliament, policy, elected, senate, treaty

This strong semantic separation makes the classification task relatively straightforward for modern ML algorithms.

5.1.2 High-Quality Data

- Wikipedia articles are well-written and topically focused
- Synthetic data was carefully crafted to maintain realistic patterns
- No ambiguous or mixed-topic documents
- Consistent writing style within each domain

5.1.3 Appropriate Feature Engineering

- 1,000 features captured sufficient vocabulary coverage
- TF-IDF effectively highlighted domain-specific terminology
- Bigrams captured important multi-word expressions
- Stopword removal eliminated noise

5.1.4 Suitable Algorithms

- Naive Bayes is particularly well-suited for text classification
- Linear models (Logistic Regression, SVM) work well with linearly separable data
- High-dimensional feature space favors these algorithms

5.2 Feature Importance Analysis

For the Naive Bayes + BoW model, we can examine which words most strongly indicate each class:

Top Sports Indicators:

- team, player, game, match, championship, scored, victory, tournament, final, season

Top Politics Indicators:

- government, parliament, legislation, elected, policy, senate, committee, congress, minister, treaty

These align perfectly with human intuition about the domains.

5.3 Comparison with Literature

Our results align with existing research on text classification:

1. **Naive Bayes for Text:** Numerous studies confirm Naive Bayes excels at document classification despite its independence assumption
2. **TF-IDF Effectiveness:** Well-established that TF-IDF improves over raw counts for most text tasks
3. **Random Forest Limitations:** Known to underperform on sparse, high-dimensional text features

5.4 Practical Implications

These results suggest that for well-defined, domain-specific text classification:

- Simple algorithms (Naive Bayes) often suffice
- Careful feature engineering is more important than complex models
- Perfect or near-perfect accuracy is achievable with clean data
- Computational efficiency can be maintained without sacrificing performance

---

6. LIMITATIONS AND CHALLENGES

Despite the strong results, this study has several important limitations:

6.1 Small Dataset Size

Issue: 97 total samples is quite small for machine learning standards

- Training set: Only 77 samples
- Test set: Only 20 samples

Implications:

- High risk of overfitting
- Perfect test accuracy may not generalize to new data
- Limited statistical power for evaluation
- Small number of test samples makes metrics less reliable

Evidence: The small standard deviations in cross-validation suggest the model might be memorizing patterns rather than learning generalizable features.

6.2 Limited Topic Diversity

Issue: Dataset covers limited subtopics within each domain

Sports Coverage:

- Primarily major competitions (Olympics, World Cup)
- Limited coverage of niche sports
- Mostly professional/international level

Politics Coverage:

- Focus on international organizations and major events
- Limited local/regional politics
- Emphasis on Western democracies

Impact: Model may struggle with:

- Articles about sports politics (e.g., Olympic committee governance)
- Political aspects of sporting events
- Less common sports or political systems

6.3 Domain Overlap in Real World

Real-world challenges not reflected in our dataset:

- **Sports Politics:** "FIFA corruption scandal", "Olympic host city selection process"
- **Political Sports:** "Diplomatic relations through sports", "National pride in competitions"
- **Mixed Content:** Articles discussing both domains equally

Our clean domain separation likely inflates performance metrics.

6.4 Synthetic Data Concerns

Potential Issues:

- Synthetic articles may have learned patterns not present in real text
- Risk of artificial patterns that don't generalize
- May lack the complexity and nuance of authentic journalism
- Could introduce systematic biases

Mitigation: We included real Wikipedia articles to balance synthetic data, but the 50-50 mix still raises validity concerns.

6.5 Evaluation Limitations

Statistical Significance:

- With only 20 test samples, differences between models may not be statistically significant
- Perfect accuracy could be achieved by chance
- Need larger test set for reliable comparisons

No Adversarial Testing:

- No deliberately ambiguous examples
- No out-of-distribution samples
- No robustness testing against perturbations

6.6 Feature Engineering Constraints

Limitations of Bag-of-Words Approaches:

- Ignores word order and syntax
- Cannot capture complex semantic relationships
- Misses context-dependent meanings
- No understanding of negation or sarcasm

Example Failure Cases (not in our dataset):

- "The political team failed to score points" - Could confuse the model
- "The minister's poor performance in the game" - Ambiguous context

6.7 Scalability Concerns

Computational Limitations:

- Manual data collection is time-consuming
- Feature extraction becomes expensive with larger vocabularies
- 1,000 features may be insufficient for broader topics
- Cross-validation is costly with larger datasets

Real-World Deployment:

- Model trained on 2024-2026 data may become outdated
- New terminology emerges constantly (e.g., new sports leagues, political movements)
- Requires regular retraining and updates

6.8 Lack of Interpretability for Some Models

While Naive Bayes and Logistic Regression provide interpretable feature weights:

- SVM decision boundaries are harder to interpret
- Random Forest feature importance can be misleading
- No explanation for individual predictions

This limits trust and debuggability in production systems.

6.9 No Multilingual Support

Current Limitation:

- English-only dataset
- Models won't generalize to other languages
- English-specific preprocessing (stopwords, etc.)

Real-world Need:

- Global news sources publish in multiple languages
- Sports events and politics are international phenomena

6.10 Ethical and Bias Considerations

Potential Biases:

- Wikipedia has known coverage biases (Western-centric, male-dominated)
- Synthetic data reflects author's biases
- Sports coverage may favor popular over niche sports
- Political coverage may favor certain political systems

Fairness Concerns:

- No analysis of performance across different regions
- No testing for demographic biases
- Could perpetuate existing media biases

---

7. CONCLUSION

This study successfully developed and evaluated multiple machine learning classifiers for distinguishing between Sports and Politics articles. Our key contributions and findings include:

7.1 Main Achievements

1. **Exceptional Classification Performance:**
    - Seven model configurations achieved 100% test accuracy
    - Naive Bayes + Bag of Words selected as the best model
    - Cross-validation scores above 97% for most models
2. **Comprehensive Feature Comparison:**
    - TF-IDF and N-grams outperformed simple Bag of Words
    - Feature engineering proved crucial for performance
    - 1,000 features provided adequate coverage
3. **Algorithm Evaluation:**
    - Naive Bayes emerged as the best choice (speed + accuracy)
    - Logistic Regression and SVM showed comparable performance
    - Random Forest underperformed on sparse text features
4. **Hybrid Data Collection:**
    - Combined Wikipedia articles with synthetic data
    - Achieved balanced dataset with clear domain separation
    - Demonstrated practical data collection strategies

7.2 Key Insights

For Practitioners:

- Simple algorithms often suffice for well-defined text classification
- Domain separation is the primary driver of classification success
- Cross-validation is essential for detecting overfitting
- Feature engineering matters more than model complexity

For Researchers:

- Small datasets can yield misleading performance metrics
- Real-world deployment requires addressing domain overlap
- Interpretability should be prioritized alongside accuracy
- Temporal and cross-domain generalization needs evaluation

7.3 Broader Implications

This study demonstrates that traditional machine learning approaches remain highly effective for document classification when:

- Domains have distinct vocabularies
- High-quality labeled data is available
- The classification task is well-defined
- Computational efficiency is important

While deep learning has achieved remarkable results in NLP, our findings suggest that classical methods should not be overlooked, especially for resource-constrained applications or when interpretability is crucial.

7.4 Limitations Acknowledgment

We acknowledge several significant limitations:

- Small dataset size (97 samples)
- Limited topic diversity within each domain
- Lack of ambiguous or mixed-topic examples
- No evaluation of cross-domain generalization
- Potential biases from synthetic data

These limitations suggest that while our models perform exceptionally well on the test set, their real-world performance on diverse, unseen data remains to be validated.

7.5 Final Remarks

Text classification is a fundamental NLP task with wide-ranging applications. This project provided hands-on experience with:

- Data collection and preprocessing

- Feature extraction techniques
- Model training and evaluation
- Performance analysis and interpretation
- Critical assessment of limitations

The perfect accuracy achieved by multiple models, while impressive, also serves as a reminder to remain critical of metrics and consider real-world deployment challenges. Future work should focus on expanding the dataset, testing robustness, and deploying models in production environments.

As natural language processing continues to evolve, the principles learned in this study—careful feature engineering, rigorous evaluation, and honest assessment of limitations—will remain valuable regardless of the specific algorithms employed.

REFERENCES

1. **Scikit-learn Documentation:** Machine Learning in Python. https://scikit-learn.org/
2. **Manning, C. D., Raghavan, P., & Schütze, H. (2008).** Introduction to Information Retrieval. Cambridge University Press.
3. **Jurafsky, D., & Martin, J. H. (2023).** Speech and Language Processing (3rd ed.). Draft.
4. **Joachims, T. (1998).** Text categorization with support vector machines. European conference on machine learning.
5. **McCallum, A., & Nigam, K. (1998).** A comparison of event models for naive bayes text classification. AAAI workshop on learning for text categorization.
6. **Salton, G., & McGill, M. J. (1983).** Introduction to Modern Information Retrieval. McGraw-Hill.
7. **Wikipedia.** Various articles on sports and politics. https://en.wikipedia.org/
8. **Pedregosa, F., et al. (2011).** Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830.

APPENDIX A: CODE AVAILABILITY

GitHub Repository: GITHUB

The complete implementation including:

- Jupyter Notebook with all experiments
- Dataset (sports_politics_dataset.csv)
- Trained model files (.pkl)
- Feature extractors (vectorizers)

- Visualization scripts
- README with usage instructions

APPENDIX B: Dataset Sample

Example Sports Article:

The basketball player made an incredible three-pointer at the buzzer. The crowd went wild as the home team secured their playoff spot. The coach praised the team's defensive performance.

Example Politics Article:

The parliament passed the new healthcare bill after months of debate. The legislation aims to expand coverage to millions of uninsured citizens. Opposition parties criticized the cost implications.

END OF REPORT
