Project Report: Binary Text Categorization Using Classical Machine Learning

1. Data Acquisition Strategy

The dataset utilized in this study originates from a publicly available news corpus distributed in JSON-Lines format. Each entry represents an independent JSON object. For experimental consistency, only two thematic categories were retained: SPORTS and POLITICS. The extraction pipeline was implemented manually to ensure full control over data parsing and preprocessing.

2. Dataset Construction and Sampling

The original dataset contains over two hundred thousand news records spanning multiple themes. For this binary classification experiment, approximately forty thousand relevant samples were identified, followed by a randomized subsampling to 2,000 instances to maintain computational efficiency during model training and evaluation.

To prevent class dominance effects, balanced sampling was applied so that SPORTS and POLITICS examples were represented in nearly equal proportions.

3. Text Normalization Pipeline

Prior to numerical encoding, textual content underwent normalization. This included lowercase transformation, removal of non-alphabetic characters, and whitespace-based token segmentation. The purpose of this stage was to ensure consistent vocabulary construction for the custom CorpusVectorizer module.

4. Feature Engineering: Count-Based Representation

A Bag-of-Words representation was implemented from scratch. The CorpusVectorizer first scans the training corpus to build a deterministic term-index mapping. Each document is then converted into a fixed-length frequency vector where each dimension corresponds to a unique token from the learned vocabulary.

For example, if the vocabulary consists of [ball, senate, win], a sentence containing 'win the senate debate' would be encoded as a count vector reflecting the frequency of these indexed terms.

5. Learning Algorithms Implemented

Three algorithmic paradigms were evaluated: probabilistic modeling, discriminative linear classification, and instance-based comparison.

5.1 Multinomial Naive Bayes: The ProbabilisticClassifier computes class priors and log-likelihoods with Laplace smoothing to avoid zero-probability issues. Posterior scores are calculated in log-space for numerical stability.

5.2 Logistic Regression: A linear decision boundary was trained using batch gradient descent. Model parameters were initialized to zero and iteratively updated using the gradient of the cross-entropy objective function.

5.3 K-Nearest Neighbors: This instance-based approach stores all training vectors. For each unseen document, Euclidean distance is computed against every stored vector, and the majority label among the nearest neighbors determines classification.

6. Experimental Observations

Using an 80/20 train-test division, the probabilistic model achieved the highest accuracy due to strong token-category associations. The linear classifier performed competitively, while KNN demonstrated slower prediction times in high-dimensional sparse spaces.

7. System Constraints

Despite promising accuracy, the count-based representation ignores syntactic ordering and semantic relationships between words. Furthermore, the KNN method exhibits $O(N \cdot D)$ complexity, limiting scalability without optimized indexing structures.

Vocabulary drift over time may also degrade performance, as emerging terminology may not be represented in the training corpus.

8. Conclusion

This project illustrates that foundational machine learning techniques remain effective for topic-level classification tasks. By implementing the pipeline components manually, the internal mechanics of text vectorization and probabilistic inference were fully understood without reliance on high-level NLP libraries.