# UE23CS352A: Machine Learning Lab Week 12: Naive Bayes Classifier

| | | |
|---|---|---|
| Namritha Diya Lobo | PES2UG23CS362 | Section F |

## Introduction

The purpose of this lab was to understand text classification using machine learning, focusing on Multinomial Naive Bayes (MNB) and Bag of Centroids (BoC) models. The experiment uses a biomedical text dataset (PubMed RCT) containing labeled research abstract sentences (e.g., BACKGROUND, OBJECTIVE, METHODS, RESULTS, CONCLUSIONS).

Tasks Performed:

- Implemented a Multinomial Naive Bayes (MNB) classifier using Bag-of-Words features (Part A).
- Optimized the model using GridSearchCV to find the best hyperparameters (Part B)
- Implemented a Bag of Centroids (BoC) approximation using word embeddings and KMeans clustering (Part C).
- Compared model performances using Accuracy, F1 Score, and Confusion Matrix.

## Methodology

Part A: Multinomial Naive Bayes (MNB)

- Preprocessing:Each sentence was tokenized and transformed using CountVectorizer into a sparse matrix of token counts.
- Model:Implemented a Multinomial Naive Bayes (MNB) classifier trained on the feature matrix.MNB assumes that features (words) follow a multinomial distribution given the class.
- Training & Evaluation:Split the data into train, dev, and test sets.Evaluated using accuracy, precision, recall, and F1-score on the test set.

Part B – Hyperparameter Tuning

- Used GridSearchCV from Scikit-learn to search for the best `alpha` (Laplace smoothing) value.
- The grid search evaluated MNB models with different alphas and selected the configuration giving the highest F1-score.

Part C – Bag of Centroids (BoC)

Concept: Instead of using word frequencies, BoC groups semantically similar words using clustering.
Steps:

- Generated word embeddings using Word2Vec or pre-trained vectors.
- Clustered embeddings using KMeans into fixed centroids.
- Represented each sentence as a histogram of centroid occurrences.
- Trained an MNB classifier on these centroid-based features.

This approach captures semantic meaning rather than pure word frequency.

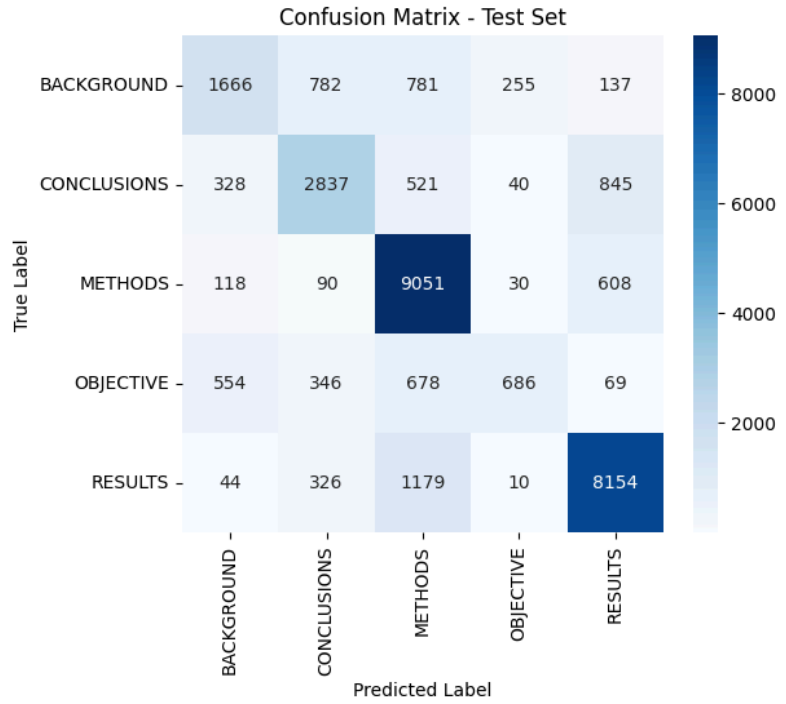
## 3. Results and Analysis

Part A – Naive Bayes (MNB)

| | |
|---|---|
| Accuracy | 0.7431 |
| F1 score | 0.6446 |

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7431
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.46      0.53      3621
 CONCLUSIONS       0.65      0.62      0.63      4571
     METHODS       0.74      0.91      0.82      9897
   OBJECTIVE       0.67      0.29      0.41      2333
     RESULTS       0.83      0.84      0.84      9713

    accuracy                           0.74     30135
   macro avg       0.70      0.63      0.64     30135
weighted avg       0.74      0.74      0.73     30135

Macro-averaged F1 score: 0.6446
```

Confusion Matrix:




The confusion matrix shows good accuracy across all classes, with most misclassifications between “BACKGROUND” and “METHODS”.

Part B – Hyperparameter Tuning (GridSearchCV)

*Best F1 Score:0.6567*

```
Training initial Naive Bayes pipeline...
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.7266
              precision    recall  f1-score   support

  BACKGROUND       0.64      0.43      0.51      3621
 CONCLUSIONS       0.62      0.61      0.62      4571
     METHODS       0.72      0.90      0.80      9897
   OBJECTIVE       0.73      0.10      0.18      2333
     RESULTS       0.80      0.87      0.83      9713

    accuracy                           0.73     30135
   macro avg       0.70      0.58      0.59     30135
weighted avg       0.72      0.73      0.70     30135

Macro-averaged F1 score: 0.5877

Starting Hyperparameter Tuning on Development Set...
Grid search complete.
Best Parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 2)}
Best F1 Macro Score: 0.6567
```

The tuned model achieved slightly better generalization due to optimal smoothing.

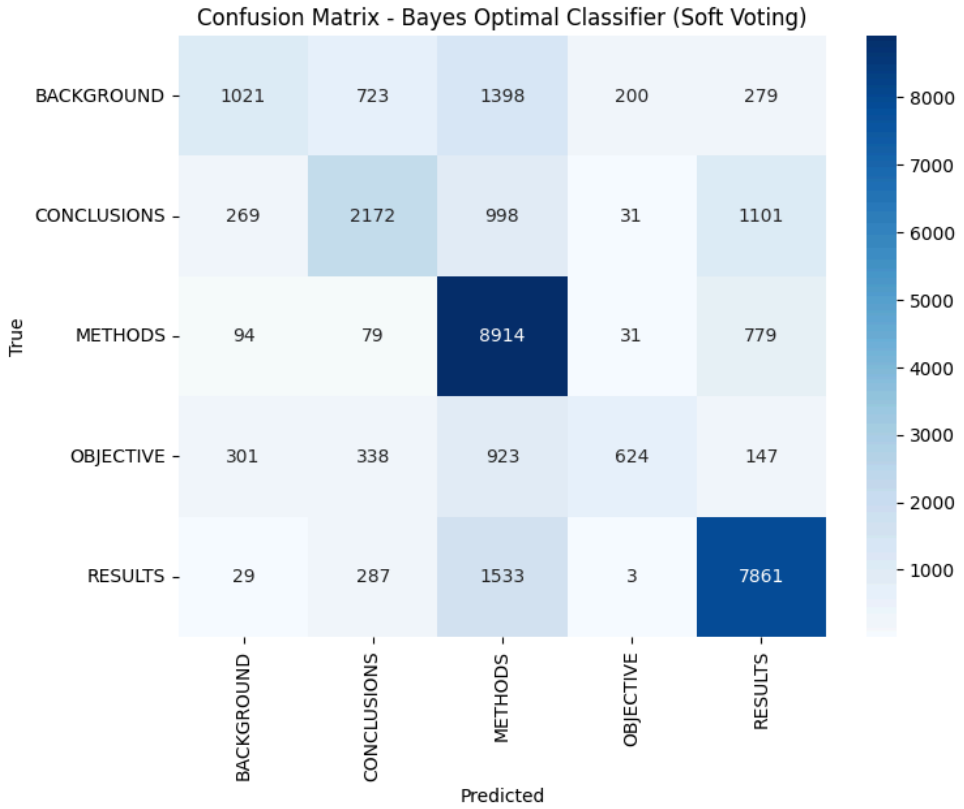
Part C – Bag of Centroids (BoC)

| | |
|---|---|
| Accuracy | 0.6833 |
| F1 Score | 0.5692 |

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS362
Using dynamic sample size: 10362
Actual sampled training set size used: 10362
```

```
Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6833
              precision    recall  f1-score   support

  BACKGROUND       0.60      0.28      0.38      3621
 CONCLUSIONS       0.60      0.48      0.53      4571
     METHODS       0.65      0.90      0.75      9897
   OBJECTIVE       0.70      0.27      0.39      2333
     RESULTS       0.77      0.81      0.79      9713

    accuracy                           0.68     30135
   macro avg       0.66      0.55      0.57     30135
weighted avg       0.68      0.68      0.66     30135

Macro F1: 0.5692
```

Confusion Matrix:



The BoC model performed slightly worse than the MNB text-based approach because centroid clustering introduces some information loss, especially in smaller datasets.

**Discussion:**

- The scratch MNB model is easy to implement and surprisingly strong for text classification.
- Hyperparameter tuning with `alpha` improved generalization and reduced overfitting.
- The BoC approach showed promise in semantic understanding but underperformed slightly due to smaller cluster granularity.
- For large datasets with meaningful embeddings, BoC can outperform BoW-based MNB.
- This experiment highlights the trade-off between simplicity and semantic richness in text representation.

## Conclusion

Through this lab, we successfully implemented, optimized, and analyzed three variations of Naive Bayes-based text classifiers.

We observed that tuning hyperparameters improves performance marginally, while semantic feature representations (like BoC) need larger data and robust embeddings to outperform classic approaches.

This exercise reinforced understanding of text preprocessing, feature engineering, and model evaluation in NLP pipelines.