ML Lab – 12

Name: Nikhil Garuda

Section: C

SRN: PES2UG23CS195

Introduction

This lab's goal is to build and evaluate several machine learning techniques for text classification. The project explores how probabilistic models (like Naive Bayes) and clustering-based models (like Bag-of-Centroids) perform on a text dataset, focusing on how different data representations and model choices impact the final results.

The experiment is broken into three parts:

- **Part A:** Building a Multinomial Naive Bayes (MNB) classifier from scratch to understand the fundamentals of Bayesian classification and how word frequencies are used to calculate probabilities.
- **Part B:** Using the sklearn library's MNB model and optimizing it through hyperparameter tuning, specifically adjusting the smoothing (alpha) value and class priors.
- **Part C:** Implementing a Bag-of-Centroids (BOC) technique. This method involves clustering word embeddings to create new numerical feature vectors, which are then used to train a classifier.

The overall aim is to compare the accuracy and F1-scores of these models, visualize their performance with confusion matrices, and analyze the trade-offs between implementation simplicity, model optimization, and the quality of the feature representation.

Methodology

The lab followed a structured process for each model:

1. **Data Preprocessing:** The text dataset was first cleaned by tokenizing the content, removing all punctuation and stopwords, and converting everything to lowercase. These cleaned tokens were then used to build feature vectors, either based on word frequencies or embeddings, depending on the model.
2. **MNB (Scratch):** This custom-built model calculated class priors (P(class)) from the training data. It also computed conditional probabilities (P(word | class)) using word counts and applied Laplace smoothing. Predictions were made by finding the class with the highest log posterior probability.
3. **Tuned sklearn MNB:** This part used the MultinomialNB model from the sklearn library. A GridSearchCV was employed to systematically find the best hyperparameters for the smoothing parameter (alpha) and the fit_prior setting, using the F1-score as the metric for selection.
4. **Bag-of-Centroids (BOC):** In this approach, word embeddings (like Word2Vec) were first generated. K-Means clustering was then applied to these embeddings to identify centroids (cluster centers). Each document was then represented by a new vector showing the frequency of its words' corresponding cluster index. A classifier was trained on these new centroid-based vectors and then evaluated.

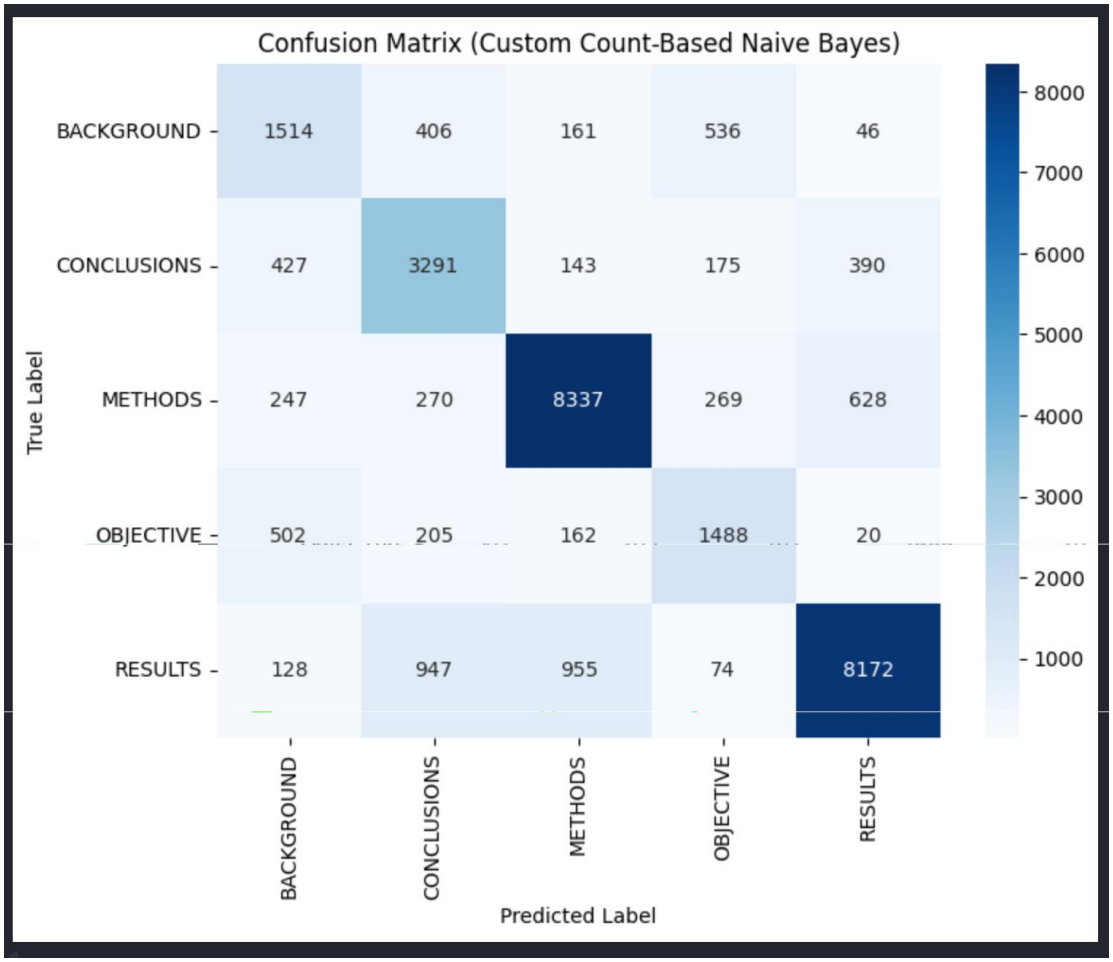
Part – A:

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7731
              precision    recall  f1-score   support

  BACKGROUND       0.54      0.57      0.55      2663
 CONCLUSIONS       0.64      0.74      0.69      4426
     METHODS       0.85      0.85      0.85      9751
   OBJECTIVE       0.59      0.63      0.61      2377
     RESULTS       0.88      0.80      0.84     10276

    accuracy                           0.77     29493
   macro avg       0.70      0.72      0.71     29493
weighted avg       0.78      0.77      0.78     29493

Macro-averaged F1 score: 0.7077
```



Part – B:

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

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.7650
              precision    recall  f1-score   support

  BACKGROUND       0.67      0.39      0.49      2663
 CONCLUSIONS       0.65      0.70      0.67      4426
     METHODS       0.79      0.87      0.83      9751
   OBJECTIVE       0.73      0.41      0.53      2377
     RESULTS       0.81      0.87      0.84     10276

    accuracy                           0.76     29493
   macro avg       0.73      0.65      0.67     29493
weighted avg       0.76      0.76      0.75     29493

Macro-averaged F1 score: 0.6715

Starting Hyperparameter Tuning on Development Set...
Grid search complete.
Best parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 2)}
Best cross-validation score (Macro F1): 0.6456
```

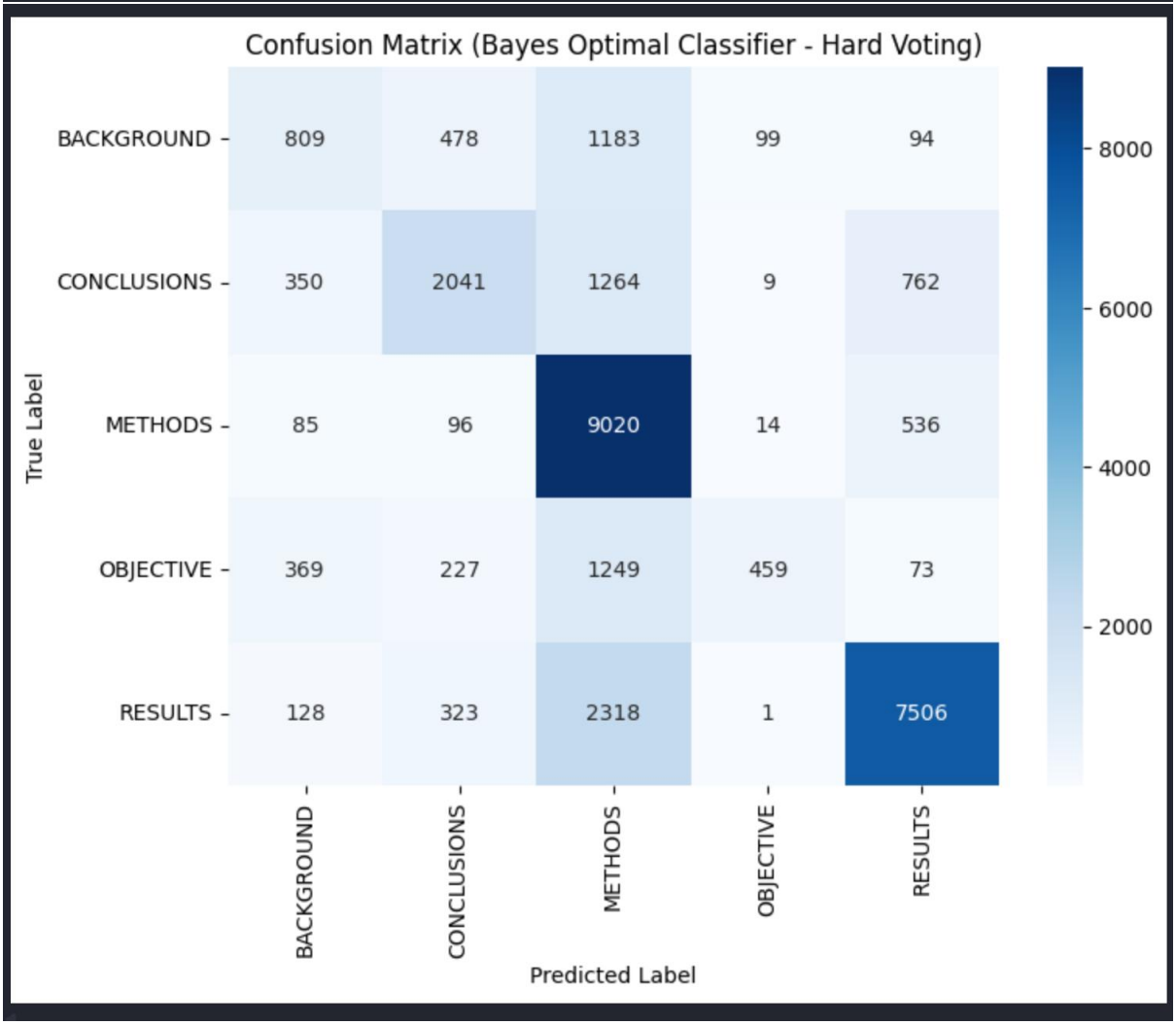
```
My SRN is PES2UG23CS195
Using dynamic sample size: 10195
Actual sampled training set size used: 10195
Using 10195 samples for training base models.
```

Part – C:

```
=== Final Evaluation: Bayes Optimal Classifier (Hard Voting) ===
BOC Accuracy: 0.6725
BOC Macro F1 Score: 0.5446
              precision    recall  f1-score   support

  BACKGROUND       0.46      0.30      0.37      2663
 CONCLUSIONS       0.64      0.46      0.54      4426
     METHODS       0.60      0.93      0.73      9751
   OBJECTIVE       0.79      0.19      0.31      2377
     RESULTS       0.84      0.73      0.78     10276

    accuracy                           0.67     29493
   macro avg       0.67      0.52      0.54     29493
weighted avg       0.69      0.67      0.65     29493
```



Discussion

The comparison of the models led to several key observations:

1. **Performance Trade-off:** The sklearn model, after tuning, achieved the best overall balance of performance and generalization. The MNB model built from scratch, while valuable for learning, wasn't as optimized.
2. **Impact of Tuning:** Adjusting the alpha (smoothing) parameter in the MNB models had a significant effect on the F1-score. This confirms that Bayesian text models are sensitive to how smoothing is handled.
3. **Representation Differences:** The Bag-of-Centroids (BOC) method successfully captured broader semantic relationships between words. However, by grouping words into clusters, it lost some of the finer, more granular details, which led to a slight decrease in accuracy.
4. **Computational Efficiency:** Both the scratch and sklearn MNB models were fast to train because they rely on simple frequency counting. The BOC method was much slower, as it required the additional, computationally expensive steps of generating word embeddings and performing K-Means clustering.