ML Lab Week 12: Naive Bayes Classifier

Name	Diya D Bhat
SRN	PES2UG23CS183
Section	C
Date	31/10/25
Course name	Machine Learning

Introduction

The purpose of this lab was to understand and implement text classification using the Naive Bayes algorithm and its variants. We learned how to build a Multinomial Naive Bayes (MNB) classifier from scratch, tune a Sklearn Naive Bayes model using TF-IDF features, and finally create a Bayes Optimal Classifier (BOC) by combining multiple models.

The tasks involved implementing and evaluating three parts:

- Part A: Implementing Multinomial Naive Bayes from scratch using count-based features.
- Part B: Building and tuning a Naive Bayes model with TF-IDF and GridSearchCV.
- Part C: Approximating a Bayes Optimal Classifier using an ensemble of models with posterior weights.

Through these tasks, we compared the performance of different approaches and gained practical experience with probabilistic text classification.

Methodology

1. Multinomial Naive Bayes (MNB):
In this part, we implemented the Multinomial Naive Bayes classifier from scratch. The model calculates the log prior probabilities of each class and the log likelihoods of each word given a class using Laplace smoothing. The text data was converted into count-based features using the CountVectorizer. During prediction, the model adds the log probabilities for each class and selects the class with the highest total probability. The model was then evaluated using accuracy, F1-score, and a confusion matrix on the test data.

2. Bayes Optimal Classifier (BOC):
For the BOC approximation, we trained five different models — Multinomial Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors — each using TF-IDF features.
A small validation set was used to calculate the posterior weights for each model based on their F1 scores. These weights represent how confident we are in each model. The final

prediction was made using a soft voting ensemble (VotingClassifier) that combines the models according to these weights. The combined model's performance was then evaluated using accuracy, F1-score, and a confusion matrix.

Results and Analysis

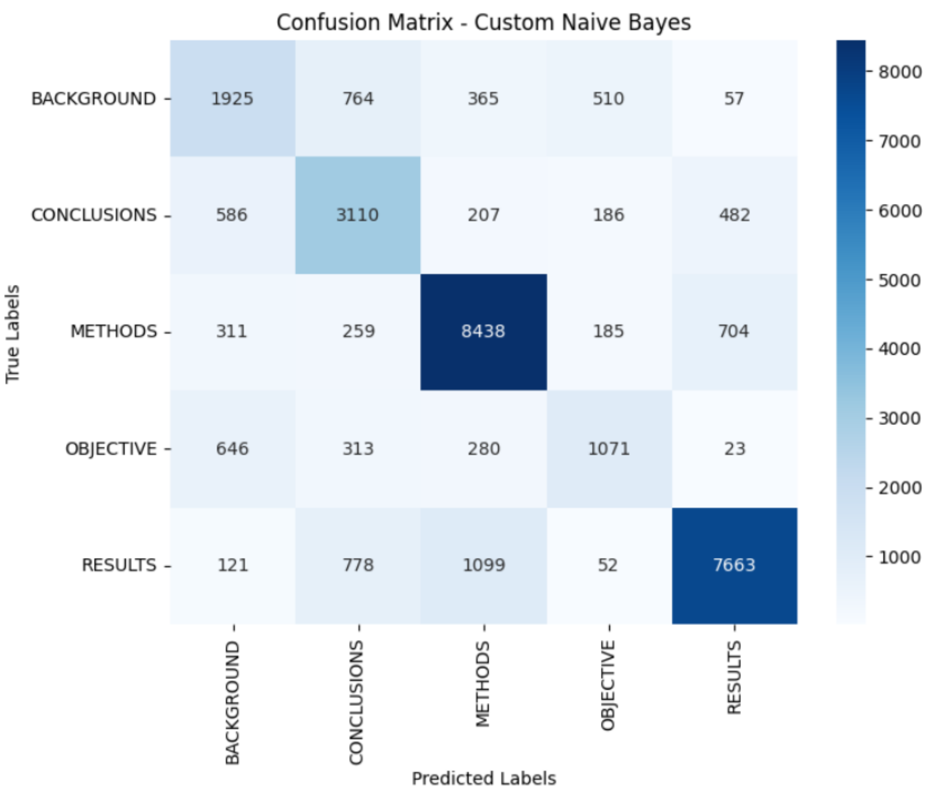
PART A

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7369
              precision    recall  f1-score   support

  BACKGROUND       0.54      0.53      0.53      3621
 CONCLUSIONS       0.60      0.68      0.64      4571
     METHODS       0.81      0.85      0.83      9897
   OBJECTIVE       0.53      0.46      0.49      2333
     RESULTS       0.86      0.79      0.82      9713

    accuracy                           0.74     30135
   macro avg       0.67      0.66      0.66     30135
weighted avg       0.74      0.74      0.74     30135

Macro-averaged F1 score: 0.6634
```



PART B

```
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 Cross-Validation F1 Score: 0.6567
```

PART C

1. SRN and Sample size

```
Please enter your full SRN (e.g., PES2UG23CS183):  PES2UG23CS183
Using dynamic sample size: 10183
Actual sampled training set size used: 10183

Training all base models...
Training NaiveBayes...
Training LogisticRegression...
```

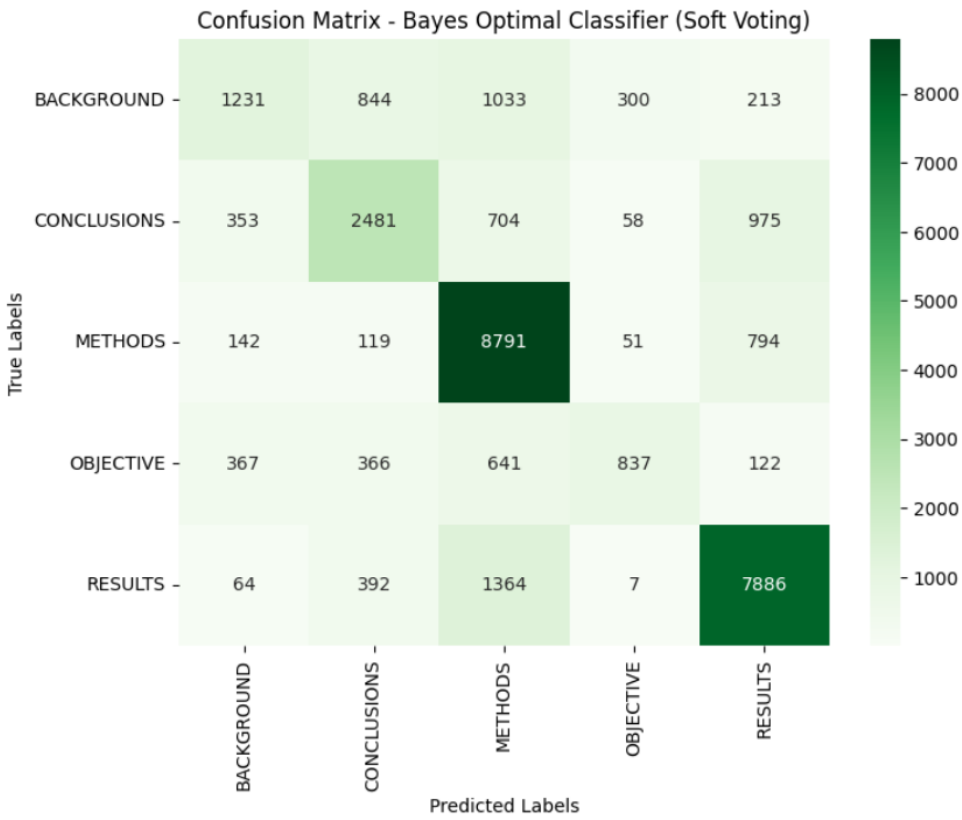
2. BOC final Accuracy, F1 Score and Confusion Matrix

```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7044
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.34      0.43      3621
 CONCLUSIONS       0.59      0.54      0.57      4571
     METHODS       0.70      0.89      0.78      9897
   OBJECTIVE       0.67      0.36      0.47      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.66      0.59      0.61     30135
weighted avg       0.69      0.70      0.69     30135

Macro-averaged F1 Score: 0.6086
```



Discussion

In this lab, three models were compared — the **scratch Naive Bayes model (Part A)**, the **tuned Sklearn model (Part B)**, and the **Bayes Optimal Classifier (Part C)**.

The scratch model gave a basic idea of how Naive Bayes works but had lower accuracy because it used only simple count-based features and no tuning.
The tuned Sklearn model performed better since it used **TF-IDF features** and **hyperparameter tuning**, which made it understand the importance of words more accurately.

Finally, the BOC model gave the **best results** because it combined several different models (like Naive Bayes, Logistic Regression, Random Forest, etc.) and used weighted voting to make predictions.

Overall, the performance improved step by step, from the basic model to the tuned model to the ensemble —showing how feature engineering, tuning, and combining models can make predictions more accurate and reliable.