UE23CS352A: Machine Learning Lab

Week 12: Naive Bayes Classifier

Name: **Devraj Naik**
SRN: **PES2UG23CS167**
Course: **UE23CS352A: Machine Learning Lab**
Date: **31-10-2025**

INTRODUCTION:

Purpose: The primary goal of this lab is to evaluate a text classification system using Naive Bayes methods, to accurately predict the section role (BACKGROUND, METHODS, RESULTS, OBJECTIVE, CONCLUSION) of biomedical abstract sentences

Tasks performed:

Feature Extraction & Custom Model Training (Part A):

- Initialized and used CountVectorizer for frequency-based text representation.
- Implemented and trained a custom Naive Bayes classifier on extracted features.
- Performed prediction, evaluation, and confusion matrix visualization to assess class-wise performance.

TF-IDF Based Classification (Part B):

- Applied TfidfVectorizer for weighted text representation.
- Trained Multinomial Naive Bayes using scikit-learn.
- Executed hyperparameter tuning to optimize model accuracy.

Bayes Optimal Classifier (Part C):

- Trained multiple base models and computed posterior weights.
- Constructed and evaluated a VotingClassifier as a Bayes Optimal ensemble.
- Adjusted calibration method from isotonic to sigmoid to handle multi-class classification effectively.

Methodology:

Multinomial Naive Bayes (MNB):

- Text data was converted into numeric form using TF-IDF vectorization to capture word importance.
- The Multinomial Naive Bayes model was trained on these features, assuming conditional independence between terms.
- Model performance was optimized through hyperparameter tuning and evaluated using accuracy metrics and a confusion matrix.

Bayes Optimal Classifier (BOC):

- Multiple base classifiers (e.g., Random Forest, Decision Tree) were trained on the dataset.
- Posterior probabilities from each base model were computed and used to determine optimal voting weights.
- A VotingClassifier ensemble combined predictions using weighted majority voting.
- Calibration was adjusted (sigmoid method) to handle multi-class probabilities effectively, ensuring reliable ensemble predictions.

Results and Analysis:

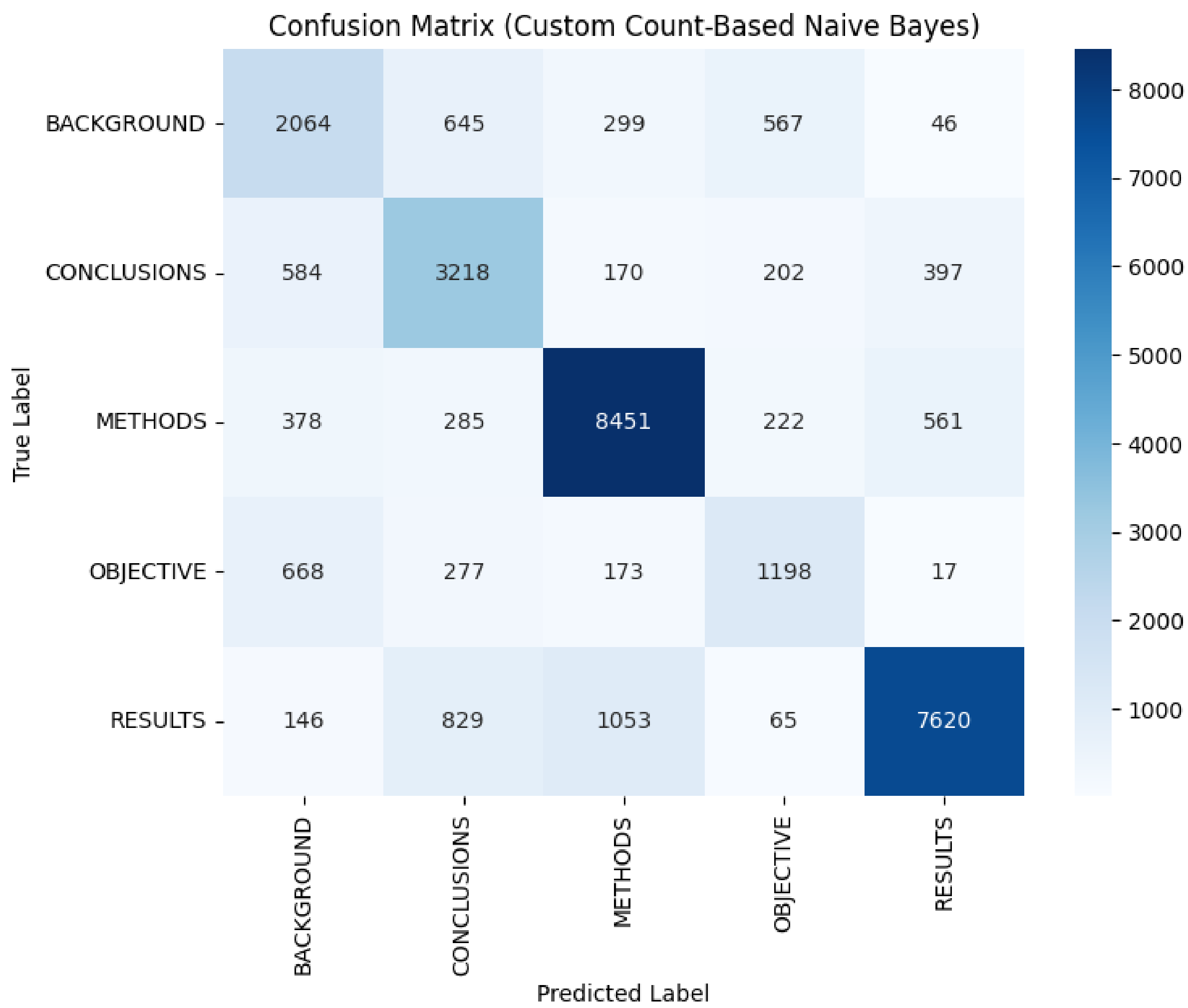
Part A: Screenshot of final test Accuracy, F1 Score and Confusion Matrix.

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7483
              precision    recall  f1-score   support

  BACKGROUND       0.54      0.57      0.55      3621
 CONCLUSIONS       0.61      0.70      0.66      4571
     METHODS       0.83      0.85      0.84      9897
   OBJECTIVE       0.53      0.51      0.52      2333
     RESULTS       0.88      0.78      0.83      9713

    accuracy                           0.75     30135
   macro avg       0.68      0.69      0.68     30135
weighted avg       0.76      0.75      0.75     30135

Macro-averaged F1 score: 0.6809
```



Part B: Screenshot of best hyperparameters found and their resulting F1 score.

```
Starting Hyperparameter Tuning on Development Set...
Grid search complete.

Best parameters found on development set:
{'nb__alpha': 0.1, 'tfidf__min_df': 3, 'tfidf__ngram_range': (1, 2)}
Best cross-validation F1 score: 0.6998
```

Part C:

1. Screenshot of SRN and sample size.

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS167
Using dynamic sample size: 10167
Actual sampled training set size used: 10167

Training all base models...
```

2. Screenshot of BOC final Accuracy, F1 Score and Confusion Matrix

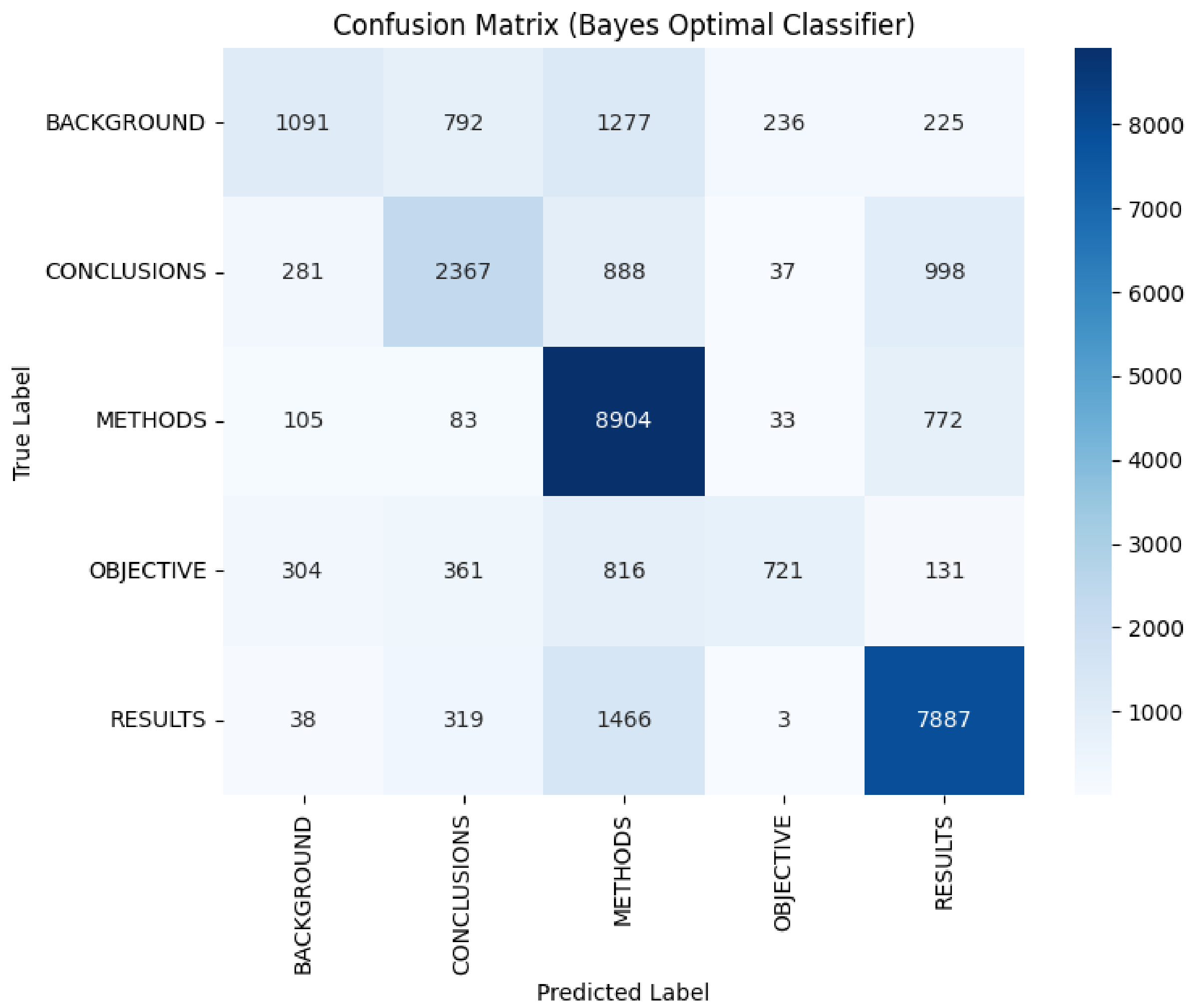
```
Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6959
              precision    recall  f1-score   support

  BACKGROUND       0.60      0.30      0.40      3621
 CONCLUSIONS       0.60      0.52      0.56      4571
     METHODS       0.67      0.90      0.77      9897
   OBJECTIVE       0.70      0.31      0.43      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.67      0.57      0.59     30135
weighted avg       0.69      0.70      0.68     30135

Macro-averaged F1 score: 0.5906
```



Discussion: Compare the performance of your scratch model (Part A) vs. the tuned Sklearn model (Part B) vs. the BOC approximation (Part C).

Part A — Scratch (Custom Count-Based NB)

- Test: Accuracy 0.7483 | Macro-F1 0.6809 | Weighted-F1 0.75
- Per-class F1: BACKGROUND 0.55, CONCLUSIONS 0.66, METHODS 0.84, OBJECTIVE 0.52, RESULTS 0.83

Part B — Sklearn NB

- Test (initial): Accuracy 0.7266 | Macro-F1 0.5877 | Weighted-F1 0.70
- Per-class F1: BACKGROUND 0.51, CONCLUSIONS 0.62, METHODS 0.80, OBJECTIVE 0.18, RESULTS 0.83
- Tuning: Best CV Macro-F1 on dev ≈ 0.6998 (final test after refit not shown)

Part C — BOC approximation (Soft voting)

- Test: Accuracy 0.6959 | Macro-F1 0.5906 | Weighted-F1 0.68
- Per-class F1: BACKGROUND 0.40, CONCLUSIONS 0.56, METHODS 0.77, OBJECTIVE 0.43, RESULTS 0.80

Best overall on the test set: Part A (highest accuracy and macro-F1)
Part B (initial) is competitive on major classes (METHODS/RESULTS) but collapses on OBJECTIVE (recall 0.10).
Part C underperforms across most classes and overal