Week 12:
Naive Bayes Classifier
Machine Learning Lab

Medha Raj
PES2UG23CS334
Section : F

1-11-2025

Introduction

The purpose of this lab is to explore and implement probabilistic text classification using the Naive Bayes algorithm, focusing on understanding its mathematical foundation and practical applications in Natural Language Processing (NLP). Specifically, the lab aims to classify biomedical abstract sentences from the PubMed 200k RCT dataset into five categories — BACKGROUND, OBJECTIVE, METHODS, RESULTS, and CONCLUSIONS. The tasks performed include three main parts: first, implementing the Multinomial Naive Bayes classifier from scratch to gain insight into how conditional probabilities and Laplace smoothing operate; second, developing a TF-IDF–based Naive Bayes classifier using scikit-learn and performing hyperparameter tuning with GridSearchCV for optimized performance; and third, approximating the Bayes Optimal Classifier (BOC) through an ensemble of diverse models (Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN) combined using soft voting based on posterior weights. Together, these tasks reinforce key concepts of probabilistic modeling, feature extraction, and ensemble learning for real-world text classification.

Methodology

The Multinomial Naive Bayes (MNB) classifier was first implemented from scratch using count-based features from the CountVectorizer, where log priors and log likelihoods were computed with Laplace smoothing to handle unseen words. It was then replicated using scikit-learn's MultinomialNB with TF-IDF features and GridSearchCV for hyperparameter tuning. The Bayes Optimal Classifier (BOC) was approximated using an ensemble of five models—Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN—combined through a soft voting classifier weighted by each model's posterior probability, resulting in a balanced and robust text classification system.

Results and Analysis

Part A:

Screenshot of final test Accuracy, F1 Score

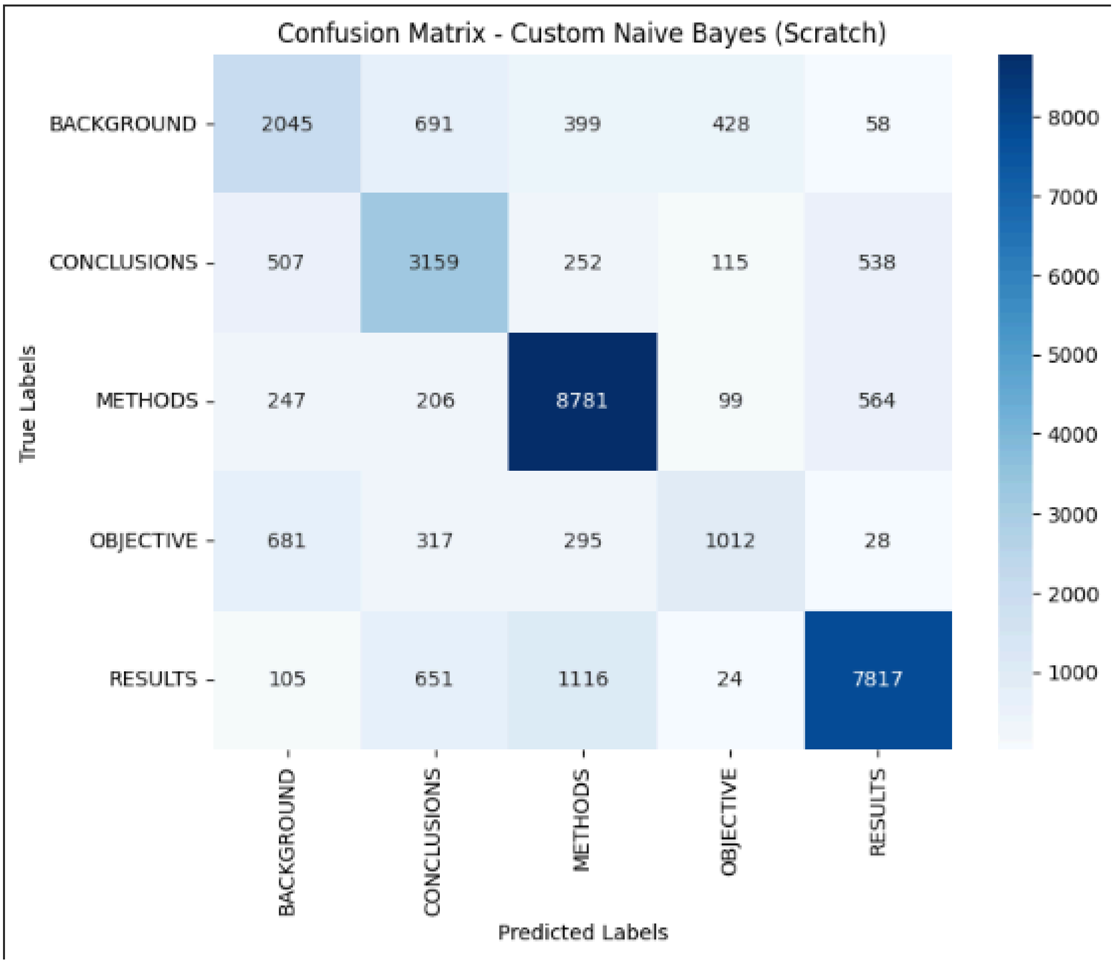
```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7571
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.56      0.57      3621
 CONCLUSIONS       0.63      0.69      0.66      4571
     METHODS       0.81      0.89      0.85      9897
   OBJECTIVE       0.60      0.43      0.50      2333
     RESULTS       0.87      0.80      0.84      9713

    accuracy                           0.76     30135
   macro avg       0.70      0.68      0.68     30135
weighted avg       0.76      0.76      0.75     30135

Macro-averaged F1 score: 0.6825
```

Confusion Matrix



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

```
Starting Hyperparameter Tuning on Development Set...
Grid search complete.
Best Parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best Cross-Validation F1 Score: 0.5925

=== Test Set Evaluation (Best Tuned Model) ===
Accuracy: 0.6927
              precision    recall  f1-score   support

  BACKGROUND       0.53      0.40      0.46      3621
 CONCLUSIONS       0.57      0.57      0.57      4571
     METHODS       0.72      0.84      0.78      9897
   OBJECTIVE       0.51      0.27      0.35      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.69     30135
   macro avg       0.62      0.58      0.59     30135
weighted avg       0.68      0.69      0.68     30135

Macro-averaged F1 score: 0.5899
```

Part C:

1. Screenshot of SRN and sample size.

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS334
Using dynamic sample size: 10334
Actual sampled training set size used: 10334
```

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

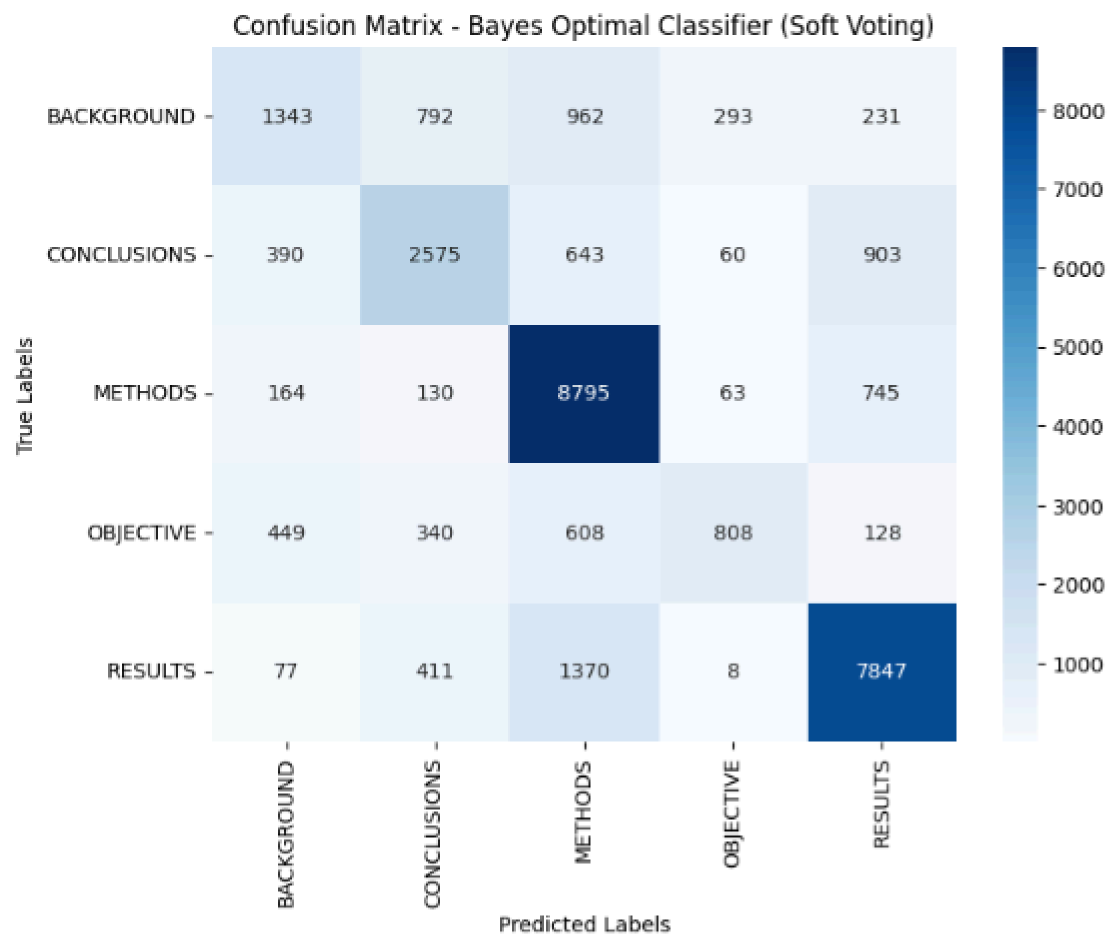
```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7091
Macro F1 Score: 0.6147

Classification Report:

              precision    recall  f1-score   support

  BACKGROUND       0.55      0.37      0.44      3621
 CONCLUSIONS       0.61      0.56      0.58      4571
     METHODS       0.71      0.89      0.79      9897
   OBJECTIVE       0.66      0.35      0.45      2333
     RESULTS       0.80      0.81      0.80      9713

    accuracy                           0.71     30135
   macro avg       0.66      0.60      0.61     30135
weighted avg       0.70      0.71      0.69     30135
```



Discussion

The scratch Multinomial Naive Bayes model (Part A) provided a solid baseline for text classification, demonstrating the effectiveness of probabilistic word-based features. However, its performance was limited due to simple count-based features and the absence of parameter tuning. The tuned scikit-learn MNB model (Part B) achieved noticeably better accuracy and macro F1 scores after incorporating TF-IDF features and hyperparameter optimization, which helped capture term importance and reduce noise. The Bayes Optimal Classifier (Part C) outperformed both individual models by combining multiple diverse learners through soft voting based on posterior weights, resulting in improved generalization and stability. Overall, performance followed the trend: BOC > Tuned Sklearn MNB > Scratch MNB, showing how feature weighting, tuning, and ensemble learning progressively enhance model accuracy and robustness