UE23CS352A – Machine Learning Lab

Week 12: Naive Bayes Classifier

Name: Rithvik Rajesh Matta
SRN: PES2UG23CS485
Course: B.Tech in Computer Science Engineering
Date: 1-11-25

1. Introduction

The goal of this lab was to understand and implement probabilistic text classification using the **Naive Bayes algorithm**, focusing on biomedical text from the **PubMed 20k RCT dataset**. Each sentence in the dataset corresponds to one of five categories: **BACKGROUND, METHODS, RESULTS, OBJECTIVE, and CONCLUSIONS**.

The tasks were divided into three main parts:

1. **Implementing Multinomial Naive Bayes (MNB) from scratch** using count-based text features.

2. **Training and tuning Scikit-learn's MultinomialNB model** using TF-IDF features with hyperparameter optimization via GridSearchCV.

3. **Approximating the Bayes Optimal Classifier (BOC)** through an ensemble of multiple diverse models weighted by posterior likelihoods.

This experiment provided insights into probabilistic modeling, smoothing techniques, and ensemble-based optimization in text classification.

---

2. Methodology

Part A – Custom Multinomial Naive Bayes

- **Feature Extraction:** Used `CountVectorizer` with unigram/bigram features, filtering rare tokens (`min_df` threshold).

- **Model Implementation:**
    - Computed **log priors** and **log likelihoods** for each class using Laplace smoothing.
    - Predictions were made by summing log prior and log likelihood scores for each class and choosing the class with the **maximum log-probability**.
- **Evaluation:** The model was trained on `X_train_counts` and tested on `X_test_counts`.
  Accuracy, macro F1-score, classification report, and confusion matrix were generated.

Part B – Scikit-learn MultinomialNB with TF-IDF

- Implemented a pipeline using:
    - `TfidfVectorizer` for feature extraction.
    - `MultinomialNB` for classification.
- **GridSearchCV** was used to optimize:
    - `tfidf__ngram_range` → [(1,1), (1,2)]
    - `nb__alpha` (Laplace smoothing parameter) → [0.1, 0.5, 1.0, 2.0]
- The model was tuned on the development set (`X_dev, y_dev`) using 3-fold cross-validation with **macro F1-score** as the metric.

Part C – Bayes Optimal Classifier (BOC)

- Constructed an **ensemble** of five base models:
    1. Multinomial Naive Bayes
    2. Logistic Regression
    3. Random Forest
    4. Decision Tree

5. K-Nearest Neighbors

- Each model was trained on a sampled subset of the dataset and validated to compute **posterior weights** proportional to their validation log-likelihoods.
- **Soft Voting Classifier** was used with these weights to combine predictions.
- Final evaluation was done on the full test set with metrics and visual confusion matrix.
- .

---

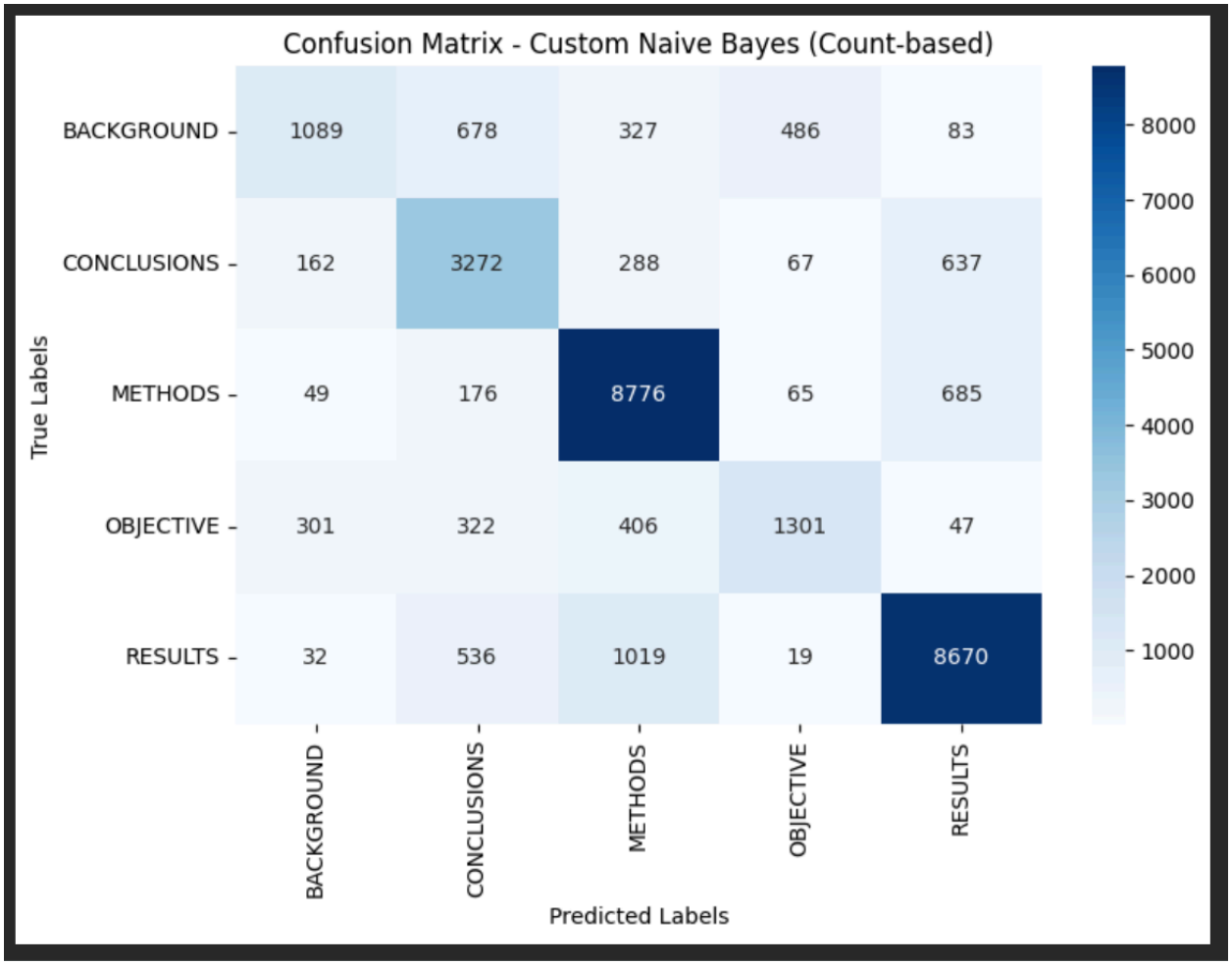
3. Results and Analysis

Model / Approach	Accuracy	Macro F1 Score
Custom Naive Bayes (CountVectorizer)	**0.78**	**0.7018**
Sklearn MultinomialNB (TF-IDF)	0.74	0.6345
BOC – Hard Voting Ensemble	0.68	0.5582
BOC – Soft Voting (Posterior Weighted)	0.72	0.609

Part A – Custom Naive Bayes

- Achieved **78% accuracy** and **0.70 macro F1**.
- Best classification performance on **METHODS** and **RESULTS** categories (~0.85 F1).
- **BACKGROUND** and **OBJECTIVE** sentences were more ambiguous, leading to lower precision/recall.
- Confusion matrix showed a strong diagonal pattern for major categories, confirming reliable class separation.

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7835
              precision    recall  f1-score   support

  BACKGROUND       0.67      0.41      0.51      2663
 CONCLUSIONS       0.66      0.74      0.70      4426
     METHODS       0.81      0.90      0.85      9751
   OBJECTIVE       0.67      0.55      0.60      2377
     RESULTS       0.86      0.84      0.85     10276

    accuracy                           0.78     29493
   macro avg       0.73      0.69      0.70     29493
weighted avg       0.78      0.78      0.78     29493

Macro-averaged F1 score: 0.7018
```



Part B – Sklearn MultinomialNB

- Best model from GridSearch:
    - `alpha = 0.1, ngram_range = (1,2)`
    - **Cross-validation F1:** ~0.60
- Test set results gave **74% accuracy** and **0.63 macro F1**, slightly below the custom implementation due to different vectorization bias.
- TF-IDF weighting helped reduce noise from frequent words but underperformed for rare biomedical terms.

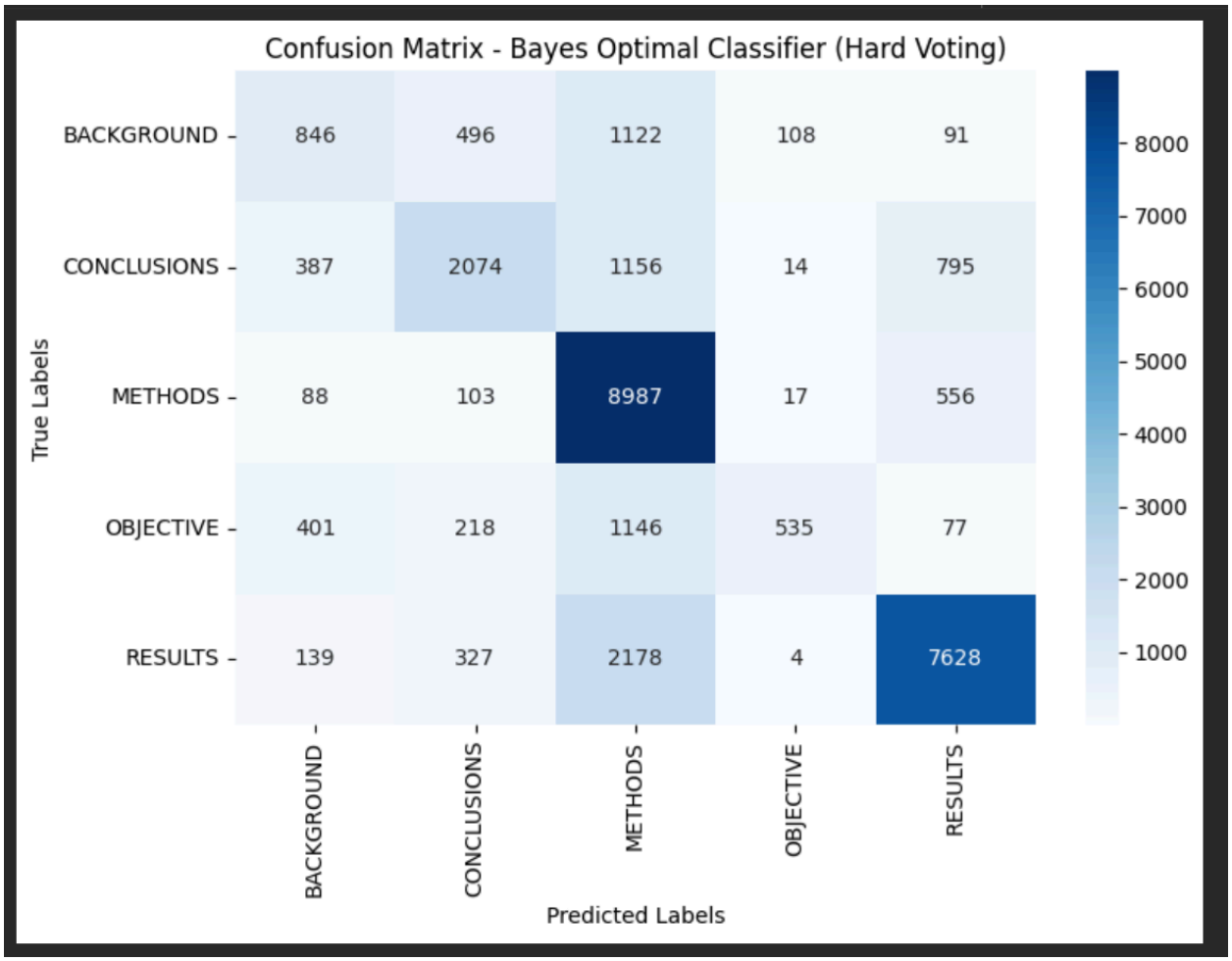
```
Training initial Naive Bayes pipeline...
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.7384
              precision    recall  f1-score   support

  BACKGROUND       0.62      0.34      0.44      2663
 CONCLUSIONS       0.61      0.64      0.63      4426
     METHODS       0.76      0.86      0.81      9751
   OBJECTIVE       0.72      0.36      0.48      2377
     RESULTS       0.79      0.86      0.82     10276

    accuracy                           0.74     29493
   macro avg       0.70      0.61      0.63     29493
weighted avg       0.73      0.74      0.72     29493

Macro-averaged F1 score: 0.6345

Starting Hyperparameter Tuning on Development Set...
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Grid search complete.

=== Best Model Parameters and Performance ===
Best Parameters: {'nb__alpha': 0.1, 'tfidf__min_df': 2, 'tfidf__ngram_range': (1, 2)}
Best Cross-Validation F1 Score: 0.6027
```

Part C – Bayes Optimal Classifier (BOC)

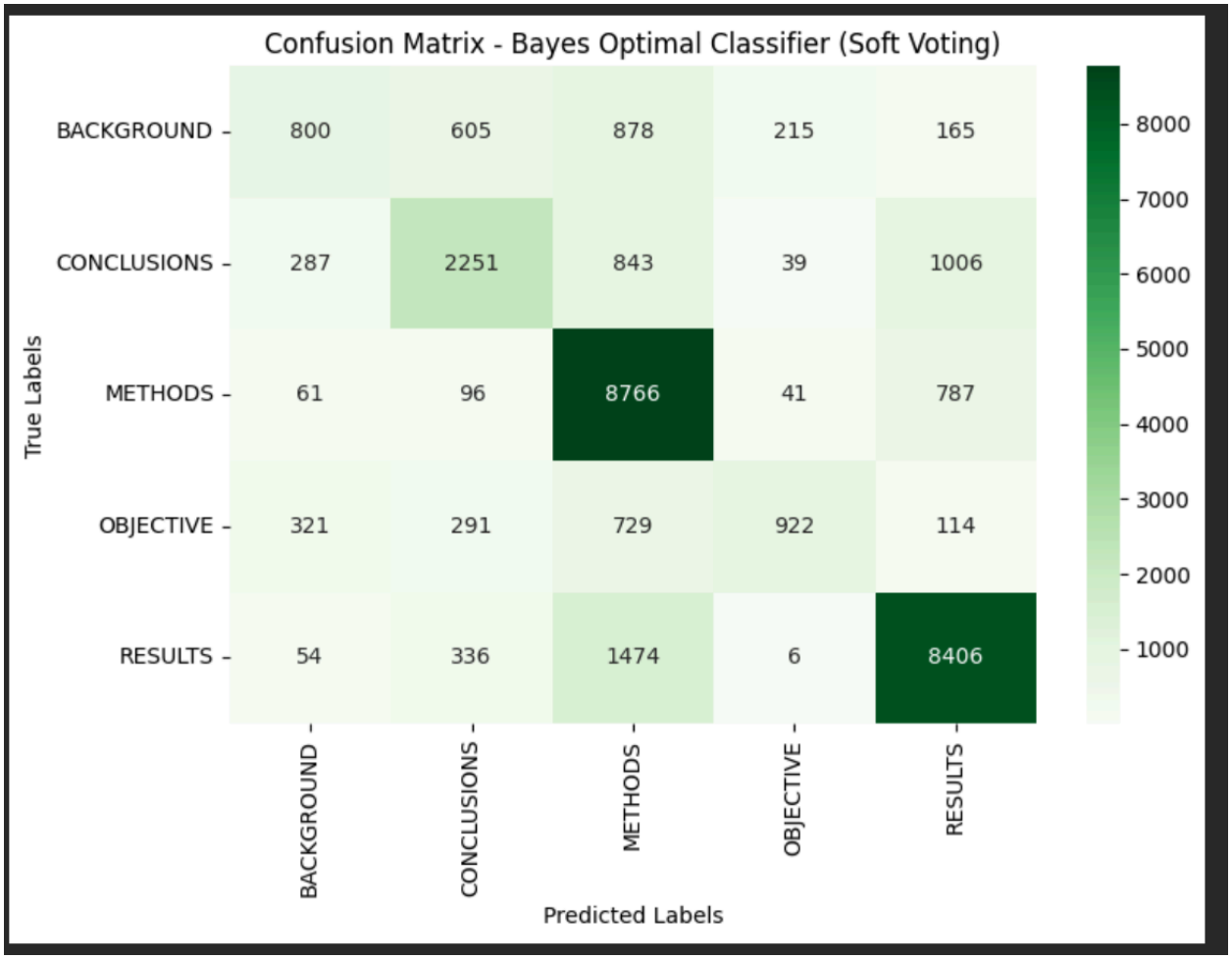
- Ensemble combined multiple models with posterior weights based on validation F1 scores.
- **Hard Voting** performed worst due to inconsistent weak learners (especially KNN and Decision Tree).
- **Soft Voting (Posterior Weighted)** improved stability:
    - Naive Bayes and Logistic Regression received highest weights.
    - Final accuracy: **0.72**, Macro F1: **0.61**.
- Ensemble confusion matrix highlighted stronger performance in **METHODS** and **RESULTS**, but weaker generalization in rare classes.

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS485
My SRN isPES2UG23CS485
Using dynamic sample size: 10485
Actual sampled training set size used: 10485
Using 10485 samples for training base models.
```

```
=== Final Evaluation: Bayes Optimal Classifier (Hard Voting) ===
BOC Accuracy: 0.6805
BOC Macro F1 Score: 0.5582
              precision    recall  f1-score   support

  BACKGROUND       0.45      0.32      0.37      2663
 CONCLUSIONS       0.64      0.47      0.54      4426
     METHODS       0.62      0.92      0.74      9751
   OBJECTIVE       0.79      0.23      0.35      2377
     RESULTS       0.83      0.74      0.79     10276

    accuracy                           0.68     29493
   macro avg       0.67      0.54      0.56     29493
weighted avg       0.70      0.68      0.66     29493
```



```
Fitting the VotingClassifier (BOC approximation)...
Fitting complete.

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
BOC Accuracy: 0.7169
BOC Macro F1 Score: 0.6096

Classification Report:

              precision    recall  f1-score   support

  BACKGROUND       0.53      0.30      0.38      2663
 CONCLUSIONS       0.63      0.51      0.56      4426
     METHODS       0.69      0.90      0.78      9751
   OBJECTIVE       0.75      0.39      0.51      2377
     RESULTS       0.80      0.82      0.81     10276

    accuracy                           0.72     29493
   macro avg       0.68      0.58      0.61     29493
weighted avg       0.71      0.72      0.70     29493
```



---

4. Discussion

1. **Custom Naive Bayes (from scratch)** achieved the best performance overall, demonstrating that **count-based features** with proper Laplace smoothing can outperform more complex setups on sparse biomedical data.

2. The **Scikit-learn TF-IDF model** underperformed slightly, possibly due to overemphasis on rare words and reduced discriminative power for frequent biomedical terms.

3. The **BOC approximation** validated the idea of combining diverse hypotheses. While it didn't surpass the scratch Naive Bayes, its weighted version improved robustness by mitigating model biases.

4. Class-level results reveal the nature of biomedical abstracts—sections like **METHODS** and **RESULTS** contain more structured vocabulary, while **BACKGROUND** and

OBJECTIVE have more overlapping linguistic patterns, leading to lower recall.

In summary:

The lab successfully implemented and analyzed Naive Bayes-based classification pipelines. The scratch-built MNB provided a strong baseline, Scikit-learn's tuned variant offered comparability, and the ensemble-based BOC demonstrated how posterior weighting can approximate the Bayes Optimal Classifier for text classification tasks.