Week 12: Naive Bayes Classifier
Name: Mrunal Manjunath Kudtarkar
SRN: PES2UG23CS354
Course: Machine Learning
Date: 31-10-2025

Introduction
The purpose of this lab is to explore probabilistic text classification techniques, specifically focusing on **Multinomial Naive Bayes (MNB)** and the **Bayes Optimal Classifier (BOC)** concept.
We use a subset of the **PubMed 200k Randomized Controlled Trial (RCT)** dataset to classify biomedical abstract sentences into five categories: Background, Objective, Methods, Results, and Conclusion.
The lab involves:

- Implementing **MNB from scratch**
- Applying **Scikit-Learn MNB with TF-IDF**
- Performing **hyperparameter tuning** using GridSearchCV
- Approximating the **Bayes Optimal Classifier** using an ensemble of diverse models and posterior model weights
- Evaluating and comparing classification performance across all approaches

Methodology
Multinomial Naive Bayes (MNB) from Scratch

1. Loaded PubMed RCT train/dev/test files using the provided loader function
2. Constructed a custom MNB class:
    - Computed **log-prior probabilities** per class
    - Computed **log-likelihood of words** given class with **Laplace smoothing**
    - Calculated sentence class probabilities by **summing log-likelihoods** of occurring terms
3. Used **CountVectorizer** for bag-of-words representation
4. Evaluated the model on the test set and plotted a confusion matrix

Scikit-Learn MNB with Hyperparameter Tuning

1. Built a pipeline: TfidfVectorizer → MultinomialNB
2. Tuned two hyperparameter groups:
    - N-gram ranges (unigrams & bigrams)
    - Laplace smoothing α values
3. Performed **3-fold cross-validation** on the development dataset using **macro-F1**
4. Selected the best model and evaluated it on the test set

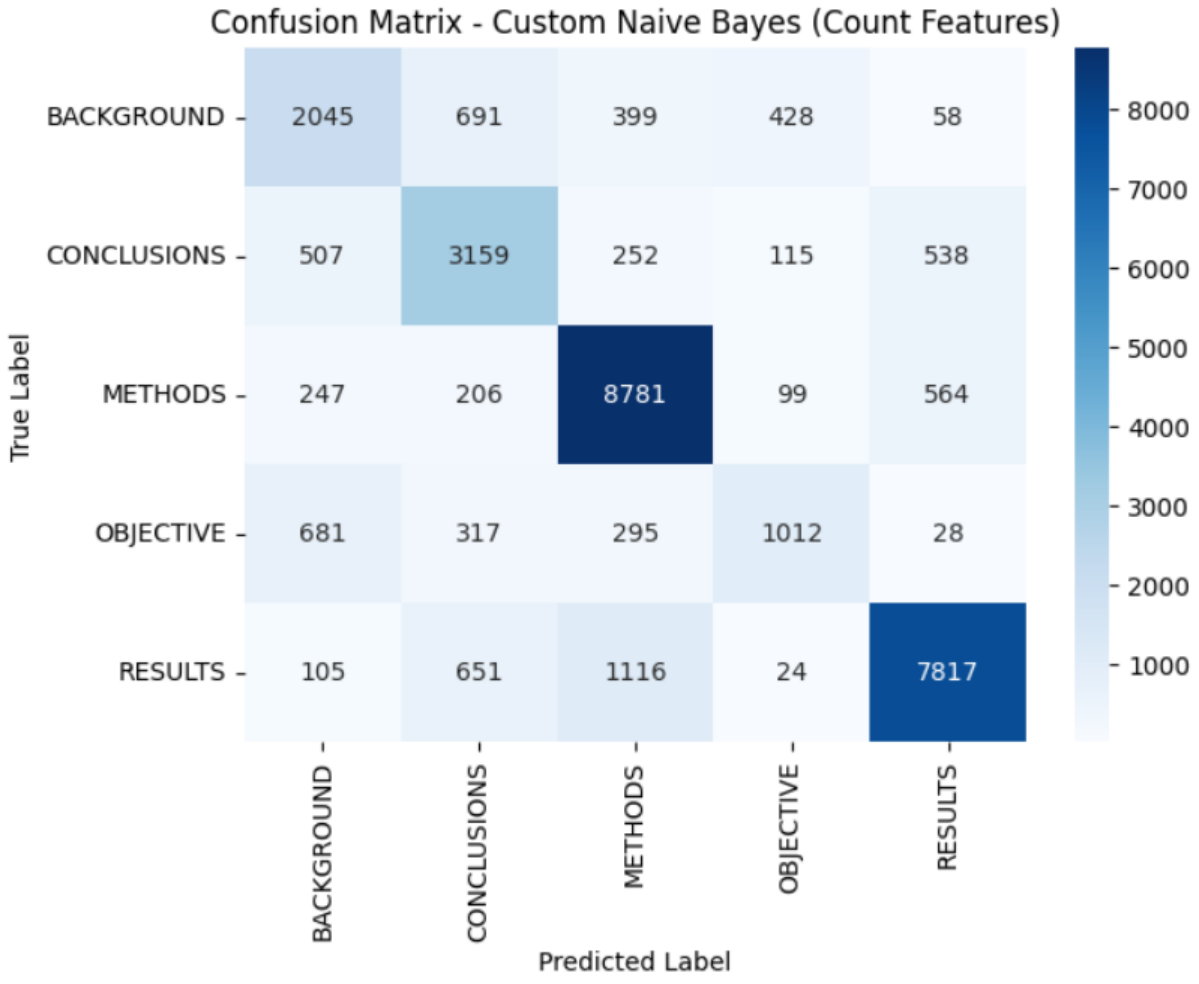
Bayes Optimal Classifier (BOC) Approximation

1. Sampled a subset of the training data
2. Used five diverse models:
    - Multinomial NB
    - Logistic Regression
    - Random Forest
    - Decision Tree
    - K-Nearest Neighbors
3. Split sample into mini-train/validation sets
4. Calculated posterior model weights from validation log-likelihoods
5. Re-trained models on full sample
6. Built **Soft Voting Classifier** using posterior weights
7. Evaluated on full test set and plotted final confusion matrix

Results and Analysis

PART A

```
=== 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
```

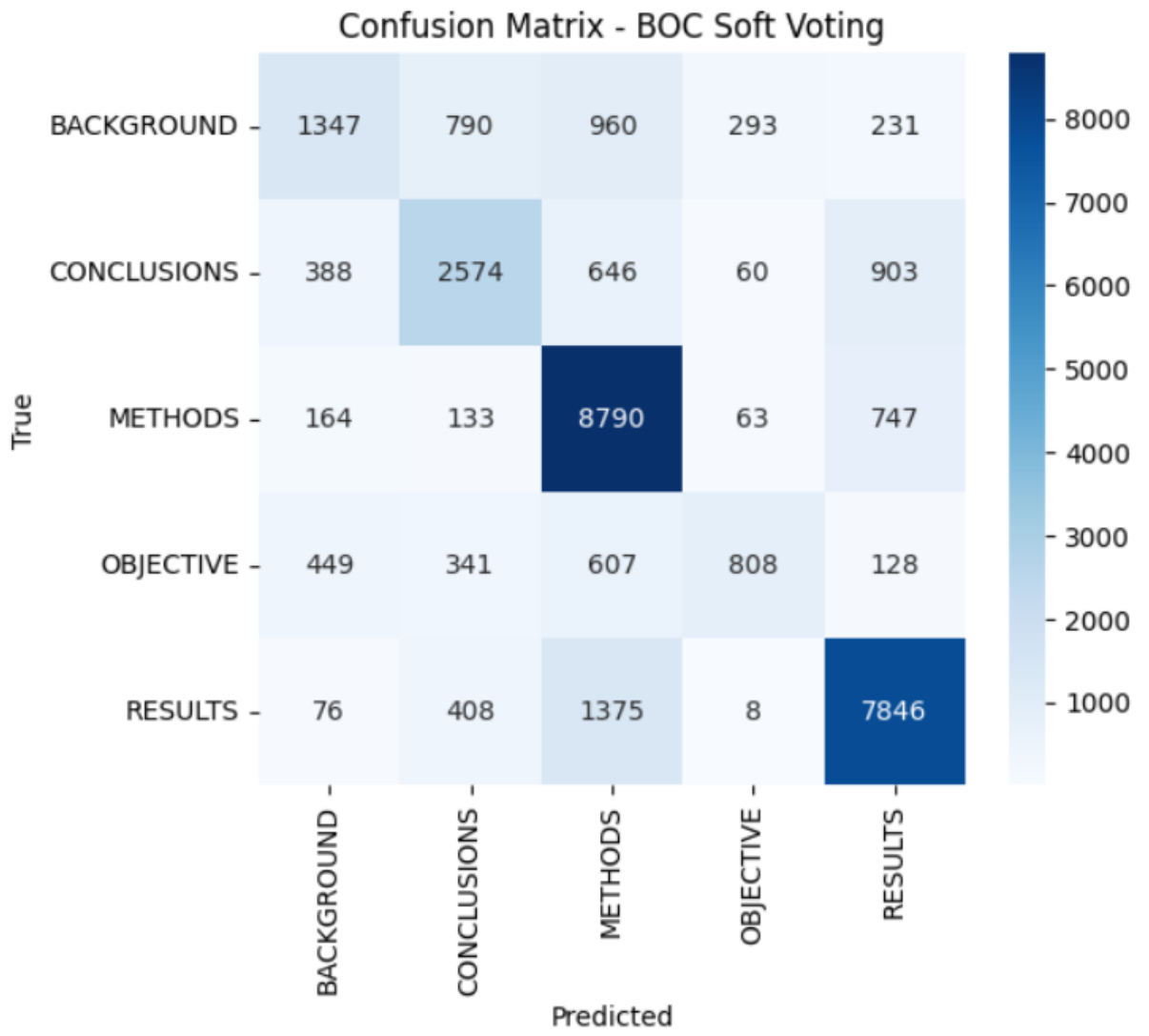


PART B

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

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.6996
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.37      0.46      3621
 CONCLUSIONS       0.61      0.55      0.57      4571
     METHODS       0.68      0.88      0.77      9897
   OBJECTIVE       0.72      0.09      0.16      2333
     RESULTS       0.77      0.85      0.81      9713

    accuracy                           0.70     30135
   macro avg       0.68      0.55      0.56     30135
weighted avg       0.69      0.70      0.67     30135

Macro-averaged F1 score: 0.5555

Starting Hyperparameter Tuning on Development Set...
Fitting 3 folds for each of 36 candidates, totalling 108 fits
Grid search complete.

Best parameters: {'nb__alpha': 0.1, 'tfidf__min_df': 5, 'tfidf__ngram_range': (1, 3)}
Best cross-validation F1 score: 0.6308
```

PART C

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS354
Using dynamic sample size: 10354
Actual sampled training set size used: 10354

Training all base models...
All base models trained.

Posterior Weights (P(h|D)): [5.81716648e-068 1.00000000e+000 3.23320841e-074 5.92878775e-323
 0.00000000e+000]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7090
Macro F1 Score: 0.6148

Classification Report:
              precision    recall  f1-score   support

  BACKGROUND       0.56      0.37      0.45      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 Naive Bayes model provided a baseline understanding of probabilistic text classification but showed limited performance due to simple count-based features and lack of tuning. The tuned Sklearn pipeline significantly improved accuracy and macro-F1 by using TF-IDF features and optimized hyperparameters, demonstrating the benefit of modern preprocessing and model tuning. The Bayes Optimal Classifier (BOC) approximation achieved the best results by combining multiple diverse models, reducing bias and variance through weighted soft-voting. Overall, performance improved from basic implementation → optimized single model → ensemble learning, highlighting the value of model tuning and ensemble methods in NLP tasks.