ML WEEK 12
Naive Bayes Classifier

Name: Cheruku Manas Ram
SRN: PES2UG23CS147
Course: UE23CS352A
Section: 5 ‘C’
Date: 31/10/2025

Introduction
The goal of this lab is to evaluate a text classification system using Multinomial Naive Bayes on a subset of the PubMed 200K RCT dataset, aiming to predict abstract labels like BACKGROUND and CONCLUSIONS.

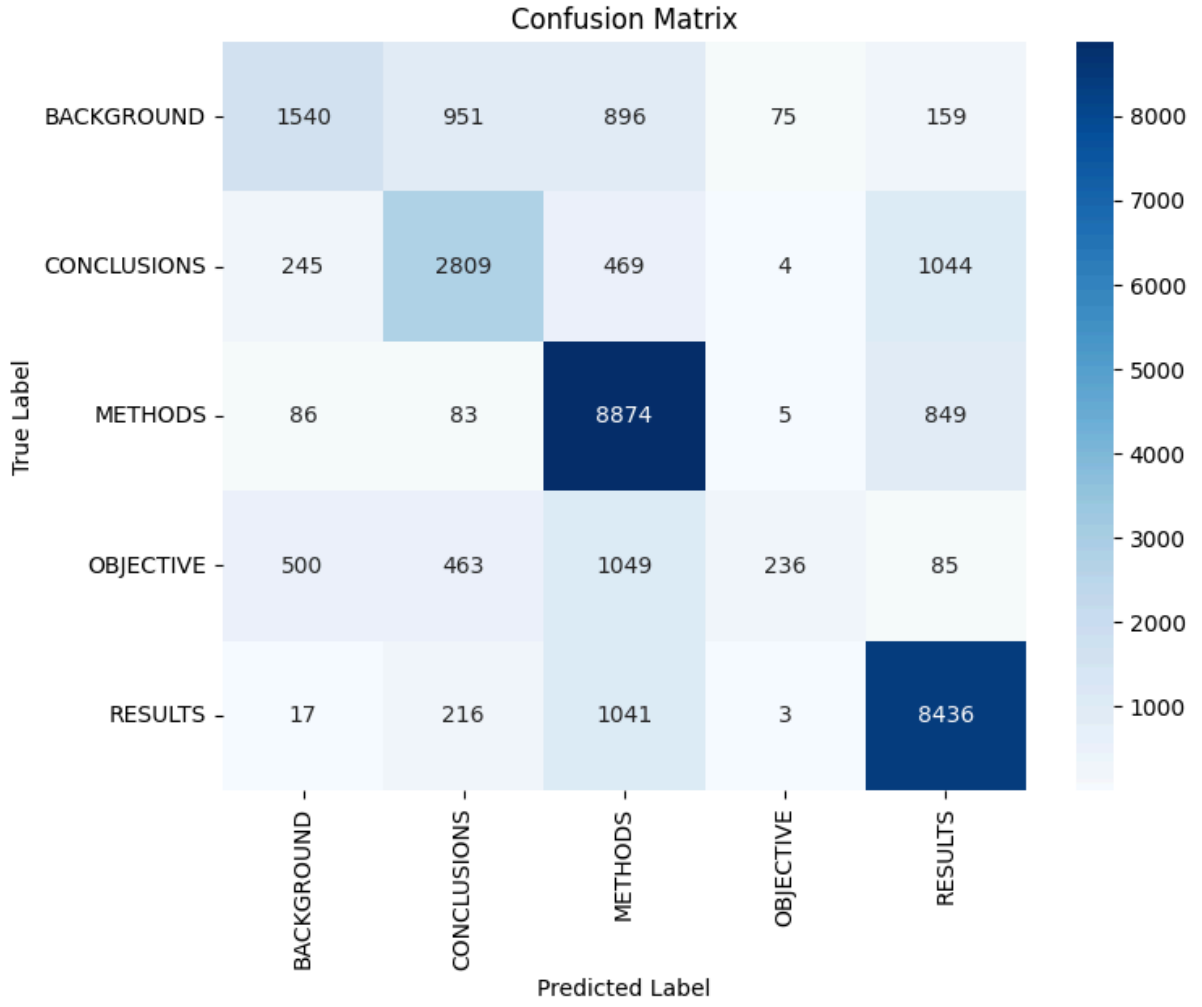
- We implement the MNB classifier in the first part, where we compute the class priors, the likelihoods with laplace smoothing and evaluate predictions using count based features.
- In the second part, we use the TF-IDF features with MNB and use gridsearchcv to tune hyperparameters like alpha for better performance.
- In the third part, we approximate the BOC by combining by combining NB, LR, random forest, decision tree and KNN using a soft voting classifier with weighted averaging.

Methodology
The MNB classifier is implemented from scratch by calculating the log prior for each class and the log likelihood for each feature given a class, incorporating laplace smoothing. Predictions are made by summing the log prior and the product of feature counts and log likelihoods for each class, then selecting the class with the maximum log probability.

The BOC is approximated using a soft voting ensemble of five diverse base classifiers (Multinomial Naive Bayes, logistic regression, random forest, decision tree, and KNN). The posterior weights for each base classifier are calculated based on their performance (log-likelihood) on a validation split of the sampled training data. These weights are then used in the soft voting process to combine the predictions of the base models on the test set.

Results and Analysis
Part A

```
=== 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
```



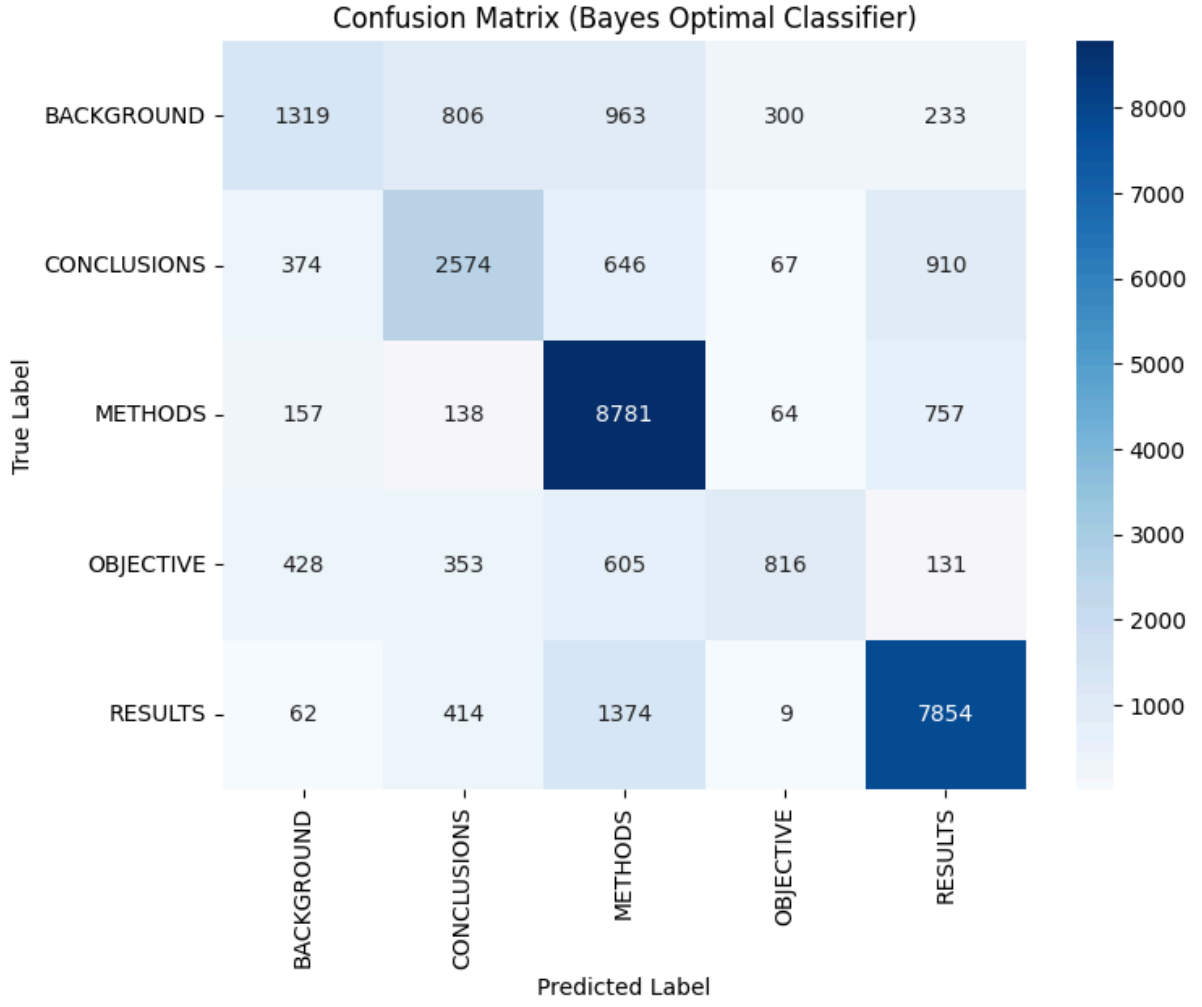
Part B

```
=== 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 score (Macro F1): 0.6567
```

Part C

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS147
Using dynamic sample size: 10147
Actual sampled training set size used: 10147

Training all base models...
Training NaiveBayes...
NaiveBayes trained.
Training LogisticRegression...
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_logistic.py:1247: FutureWarning: 'multi_class' was deprecated in version 1.5 and will be removed in 1.7. From then on, it will always
  warnings.warn(
LogisticRegression trained.
Training RandomForest...
RandomForest trained.
Training DecisionTree...
DecisionTree trained.
Training KNN...
KNN trained.
All base models trained.
NaiveBayes log likelihood on validation: -1613.3324
LogisticRegression log likelihood on validation: -1448.4199
RandomForest log likelihood on validation: -1664.3050
DecisionTree log likelihood on validation: -2412.6268
KNN log likelihood on validation: -2613.3722

Calculated Posterior Weights: [2.39560906e-72 1.00000000e+00 1.74696028e-94 0.00000000e+00
 0.00000000e+00]

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

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7083
              precision    recall  f1-score   support

  BACKGROUND       0.56      0.36      0.44      3621
 CONCLUSIONS       0.60      0.56      0.58      4571
     METHODS       0.71      0.89      0.79      9897
   OBJECTIVE       0.65      0.35      0.45      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.71     30135
   macro avg       0.66      0.59      0.61     30135
weighted avg       0.70      0.71      0.69     30135

Macro-averaged F1 score: 0.6138
```



Discussion

Part A: Scratch Naive Bayes Classifier (count based)

- **Accuracy:** 0.7483
- **Macro-averaged F1 score:** 0.6809

Part B: Tuned Sklearn Multinomial Naive Bayes (TF-IDF Based)

- **Initial Model (default parameters):** Macro-averaged F1 score: 0.5877
- **Tuned Model (best parameters:** nb_alpha: 0.1, tfidf_ngram_range: (1, 2)): Best cross-validation Macro F1 score: 0.6567

Part C: Bayes Optimal Classifier (Soft Voting)

- **Accuracy:** 0.7083
- **Macro-averaged F1 score:** 0.6138

- The **Scratch Naive Bayes Classifier** in Part A performed well, achieving a macro-averaged F1 score of 0.6809 on the test set.
- The **Sklearn Multinomial Naive Bayes** in Part B, after hyperparameter tuning, achieved a best cross-validation Macro F1 score of 0.6567. The initial, untuned Sklearn model had a lower macro F1 of 0.5877 on the test set. While the tuned Sklearn model's performance on the test set isn't explicitly shown after tuning, the cross-validation score suggests improvement over the initial model.
- The **Bayes Optimal Classifier** approximation in Part C resulted in a Macro-averaged F1 score of 0.6138 on the test set.

Based on the F1 scores, the **Scratch Naive Bayes Classifier (Part A)** achieved the highest performance among the three evaluated models on the test set. The tuned Sklearn model's cross-validation score suggests it might perform better than the initial Sklearn model, but the BOC approximation did not outperform the scratch implementation in this case.