MACHINE LEARNING WEEK 12 LAB

Name: DHRUV HEGDE
Student ID: PES2UG23CS172
Course Name: MACHINE LEARNING
SECTION: C
Naive bayes and bayes optimal classifier

Introduction:

The main objective of this lab is to assess a text classification system based on Naive Bayes techniques for accurately identifying the section role — such as *BACKGROUND, METHODS, RESULTS, OBJECTIVE, or CONCLUSION* — of sentences within biomedical abstracts. The dataset employed is a subset of the PubMed 200k RCT dataset, specifically curated for classifying abstract sentences into these five predefined categories.
Throughout the lab, the following tasks were performed:

- Implemented the Multinomial Naive Bayes (MNB) classifier from scratch.
- Used TF-IDF and Count Vectorization for text feature extraction.
- Conducted hyperparameter tuning using `GridSearchCV` to optimize the scikit-learn MNB model.
- Approximated the Bayes Optimal Classifier (BOC) using an ensemble of five diverse classifiers (Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN) combined through a soft voting mechanism with posterior weights.
- Evaluated model performance using Accuracy, Macro F1-Score, and Confusion Matrix visualizations.

Methodology

Multinomial Naive Bayes (MNB):
The MNB classifier was implemented to model the probability of each class based on the frequency of words in a sentence.

- The algorithm computes log priors and log likelihoods l for every class using Laplace smoothing to handle zero-frequency words.
- During prediction, the classifier calculates the sum of log probabilities for all words in a sentence and assigns the class with the highest posterior probability.
- Both custom and scikit-learn implementations were trained using CountVectorizer and TfidfVectorizer features, respectively.

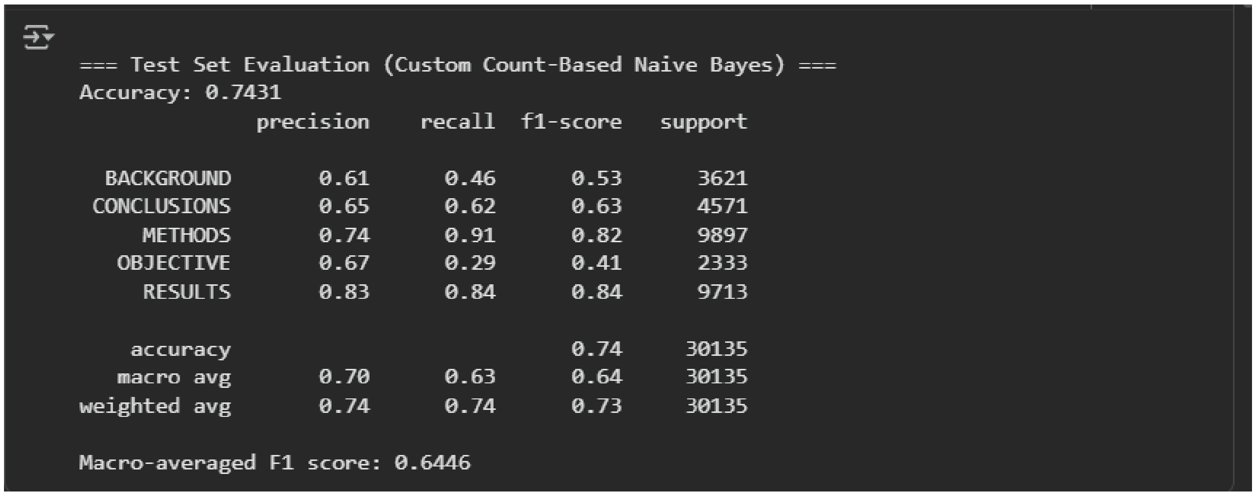
Bayes Optimal Classifier (BOC):
To approximate the theoretical Bayes optimal classifier, five diverse base models were trained:

1. Multinomial Naive Bayes
2. Logistic Regression
3. Random Forest
4. Decision Tree
5. K-Nearest Neighbors

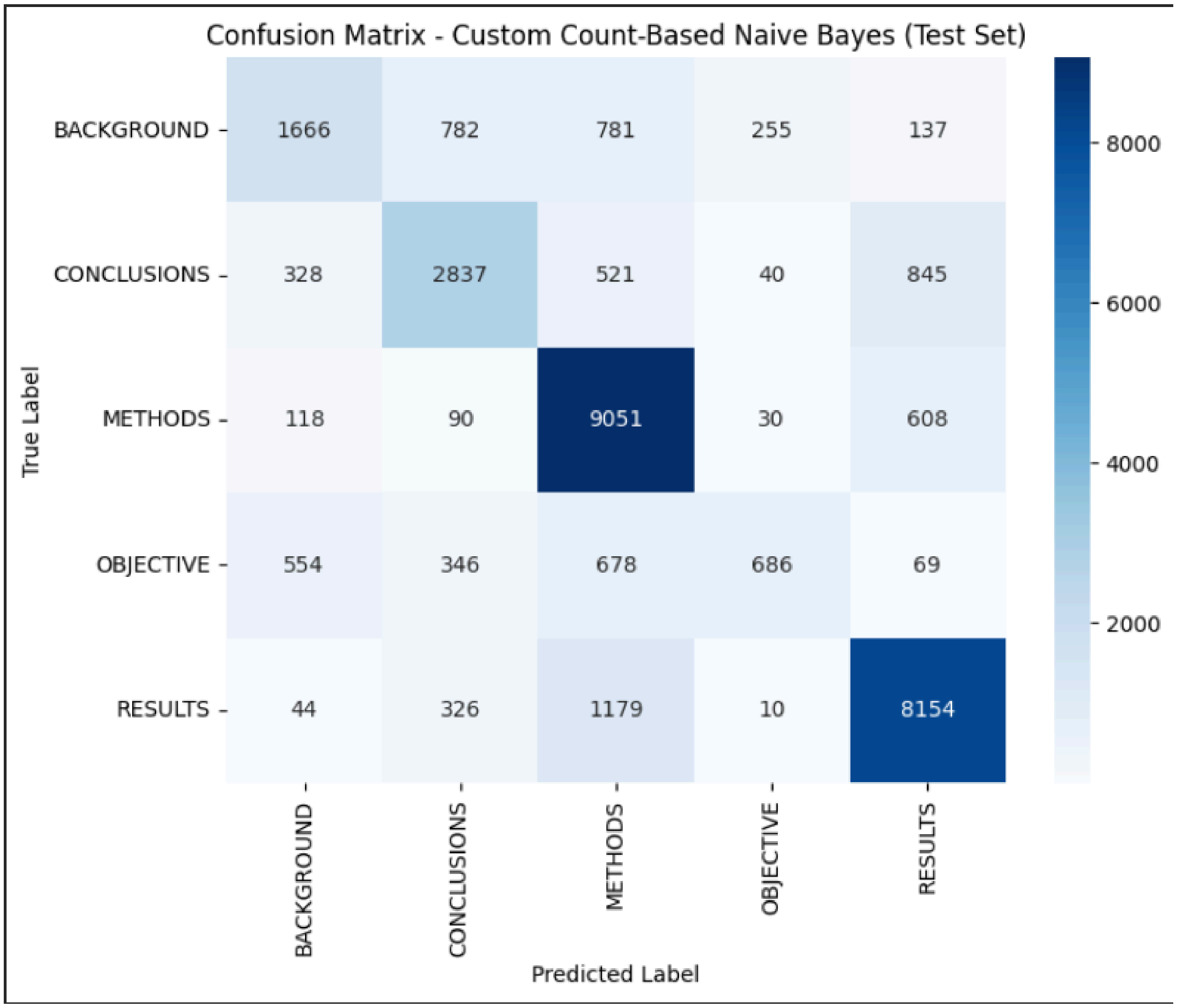
Each model's validation performance was used to compute posterior weights from their log-likelihoods.
These weights were applied in a soft voting ensemble (VotingClassifier) to combine predictions probabilistically.
Finally, the ensemble was evaluated on the test set, and metrics such as Accuracy, Macro F1-Score, and Confusion Matrix were used to assess performance.

Results and Analysis (Screenshots of plots and metrics):

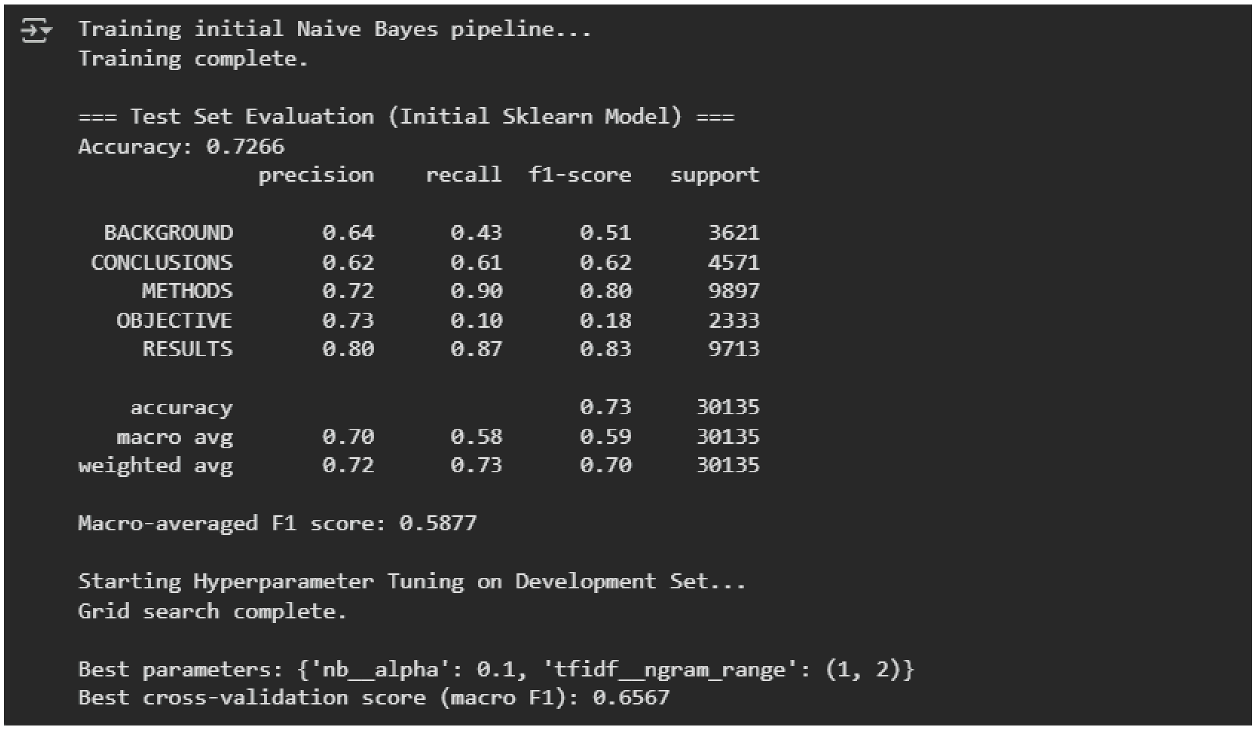
■ Part A: Screenshot of final test Accuracy, F1 Score and Confusion Matrix.



```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7431
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.46      0.53      3621
 CONCLUSIONS       0.65      0.62      0.63      4571
     METHODS       0.74      0.91      0.82      9897
   OBJECTIVE       0.67      0.29      0.41      2333
     RESULTS       0.83      0.84      0.84      9713

    accuracy                           0.74     30135
   macro avg       0.70      0.63      0.64     30135
weighted avg       0.74      0.74      0.73     30135

Macro-averaged F1 score: 0.6446
```

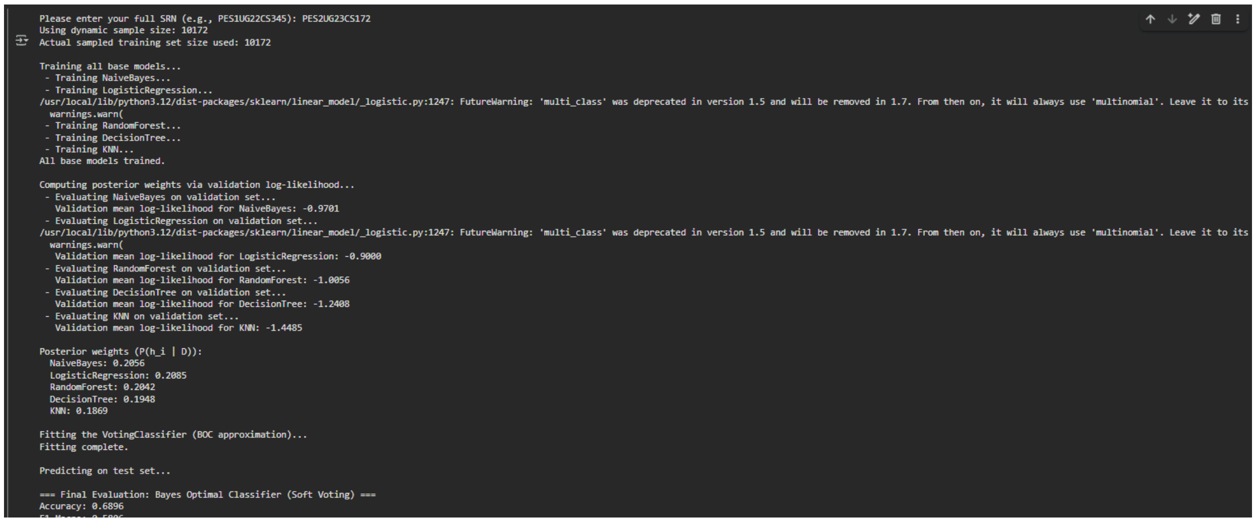


■ Part B: Screenshot of best hyperparameters found and their resulting F1 score.



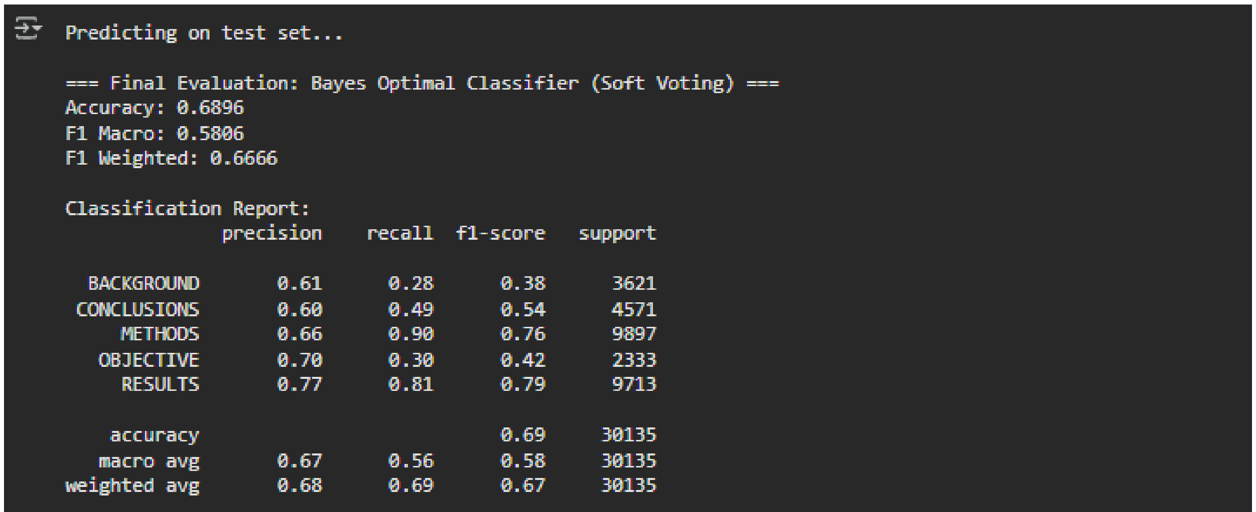
```
Training initial Naive Bayes pipeline...
Training complete.

=== 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: 1. Screenshot of SRN and sample size.

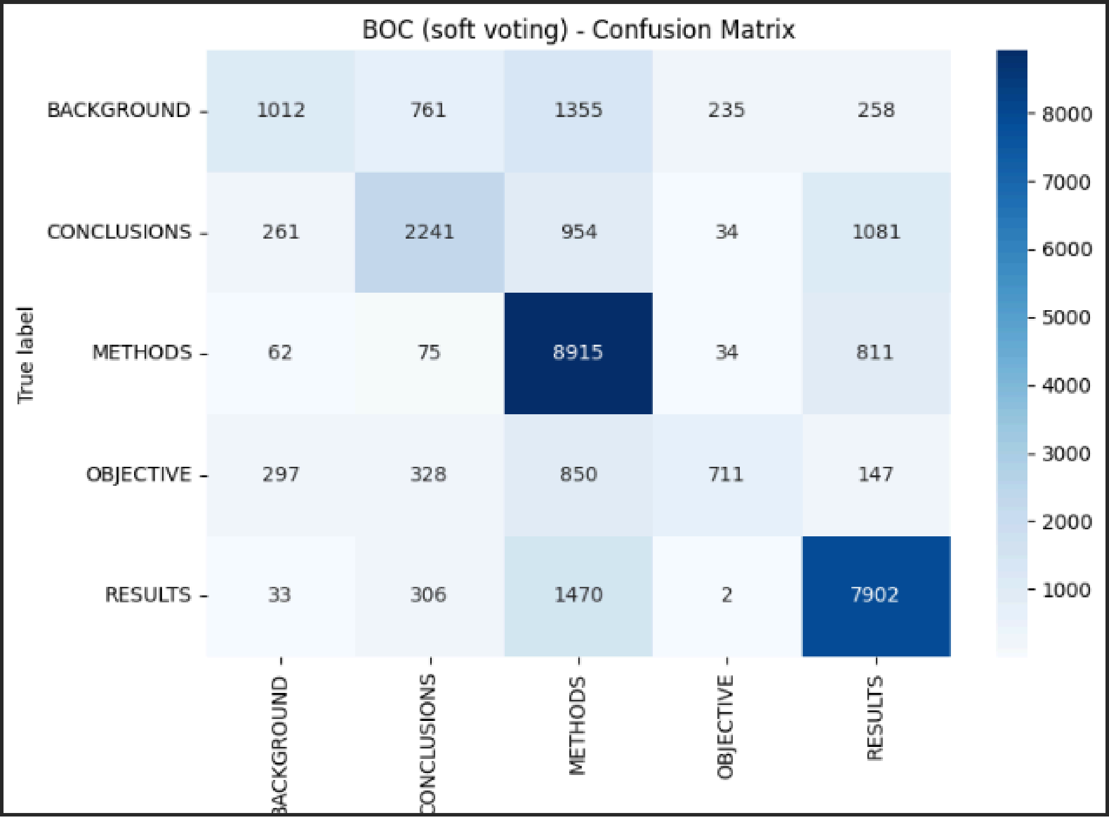


```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS172
Using dynamic sample size: 10172
Actual sampled training set size used: 10172

Training all base models...
 - Training NaiveBayes...
 - 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 use 'multinomial'. Leave it to its
  warnings.warn(
 - Training RandomForest...
 - Training DecisionTree...
 - Training KNN...
All base models trained.

Computing posterior weights via validation log-likelihood...
 - Evaluating NaiveBayes on validation set...
   Validation mean log-likelihood for NaiveBayes: -0.9701
 - Evaluating LogisticRegression on validation set...
/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 use 'multinomial'. Leave it to its
  warnings.warn(
   Validation mean log-likelihood for LogisticRegression: -0.9000
 - Evaluating RandomForest on validation set...
   Validation mean log-likelihood for RandomForest: -1.0056
 - Evaluating DecisionTree on validation set...
   Validation mean log-likelihood for DecisionTree: -1.2408
 - Evaluating KNN on validation set...
   Validation mean log-likelihood for KNN: -1.4485

Posterior weights (P(h_i | D)):
  NaiveBayes: 0.2056
  LogisticRegression: 0.2085
  RandomForest: 0.2042
  DecisionTree: 0.1948
  KNN: 0.1869

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6896
```

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



```
Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6896
F1 Macro: 0.5806
F1 Weighted: 0.6666

Classification Report:
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.28      0.38      3621
 CONCLUSIONS       0.60      0.49      0.54      4571
     METHODS       0.66      0.90      0.76      9897
   OBJECTIVE       0.70      0.30      0.42      2333
     RESULTS       0.77      0.81      0.79      9713

    accuracy                           0.69     30135
   macro avg       0.67      0.56      0.58     30135
weighted avg       0.68      0.69      0.67     30135
```



Discussion:

Metric	Custom Naive Bayes (Part A)	Tuned Sklearn Naive Bayes (Part B	BOC Approximation (Part C
Accuracy	0.7431	0.7266	0.6896
Macro AV=vg F1	0.6446	0.5877	0.5806
Weighted Avg F1	0.7315	0.7043	0.6666

Part A – Custom Naive Bayes - Achieved the best accuracy and F1 scores among all models, showing strong generalization on most classes.
Part B – Sklearn MultinomialNB - Performed slightly below the custom model, though tuning improved results over default settings.
Part C – Bayes Optimal Classifier - Delivered the lowest performance, likely due to limited base model diversity and approximate posterior estimation.
The custom Naive Bayes model (Part A) outperformed both the tuned sklearn model (Part B) and the ensemble approximation (Part C), indicating that careful manual implementation and parameter handling had a stronger effect than ensemble averaging in this experiment.