Lab Report

Project Title: Naive Bayes and Bayes Optimal Classifier for Text Classification

Name: C Bhargav

SRN: PES2UG23CS137

Course: Machine Learning

Date: November 2, 2025

Introduction

The purpose of this lab was to implement and evaluate different classification models for categorizing sentences from PubMed abstracts. The primary tasks included:

Implementing a Multinomial Naive Bayes (MNB) classifier from scratch.
Training and tuning a scikit-learn MNB classifier using TF-IDF features.
Approximating a Bayes Optimal Classifier (BOC) by combining several diverse models.
The performance of these three approaches was compared based on accuracy, F1 score, and confusion matrices.

Methodology

Multinomial Naive Bayes (MNB) from Scratch
A Multinomial Naive Bayes classifier was implemented manually. This model is probabilistic and based on Bayes' theorem with a "naive" assumption of conditional independence between features. The implementation involved:

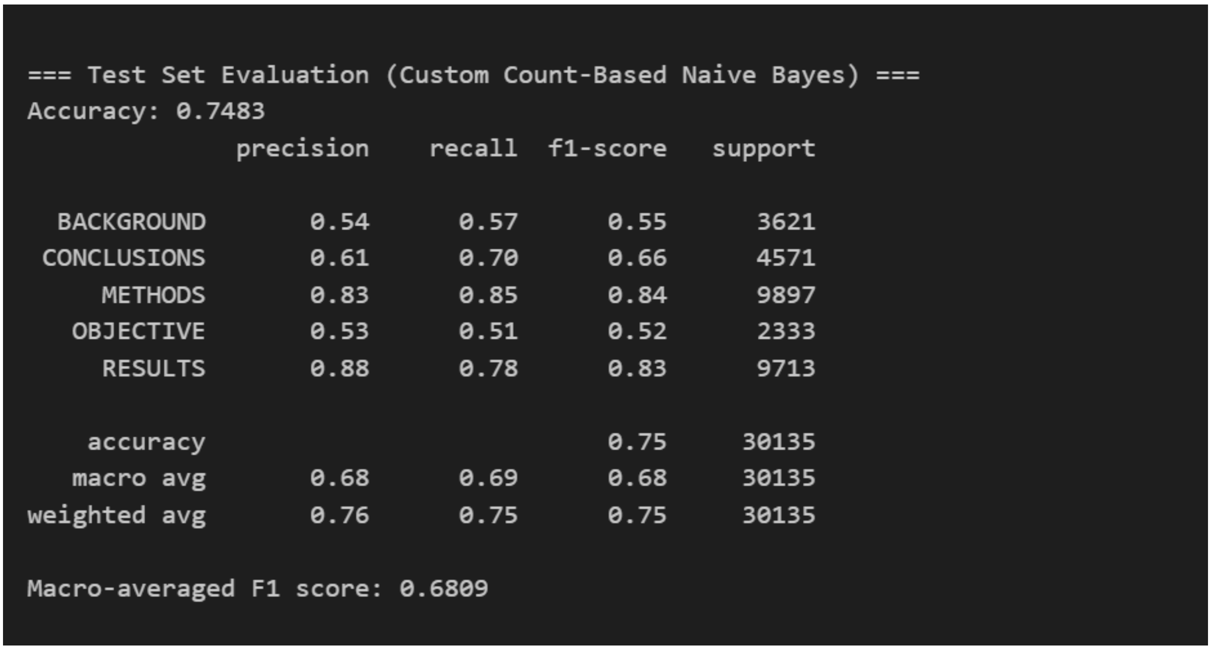
Fitting: Calculating the log prior probability for each class and the log likelihood of each feature (word) given a class. Laplace (additive) smoothing was applied to handle words not seen during training.
Prediction: For a given document, the posterior probability for each class was calculated by summing the log prior and the log likelihoods of the words in the document. The class with the highest log posterior probability was chosen as the prediction.
Bayes Optimal Classifier (BOC) Approximation
The Bayes Optimal Classifier is a theoretical model that represents the best possible performance on a given task. It achieves this by averaging over the predictions of all hypotheses in the hypothesis space, weighted by their posterior probability.

In this lab, the BOC was approximated using a soft-voting ensemble of five diverse models:

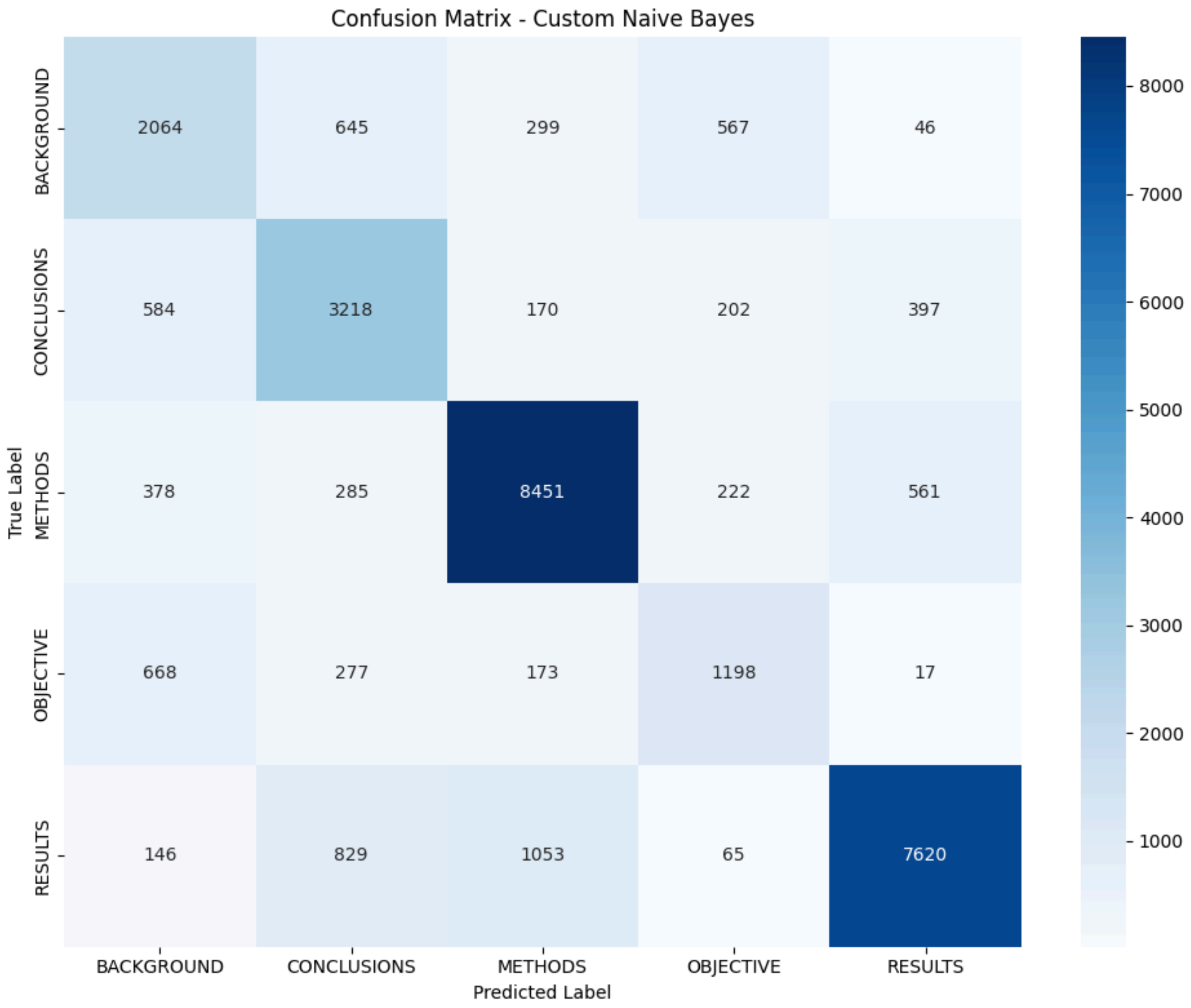
Multinomial Naive Bayes
Logistic Regression
Random Forest
Decision Tree
K-Nearest Neighbors
The posterior weights for each model, P(h|D), were calculated based on their performance (log-likelihood) on a validation subset of the training data. These weights were then used in the VotingClassifier to combine the probabilistic predictions of the individual models.

Results and Analysis

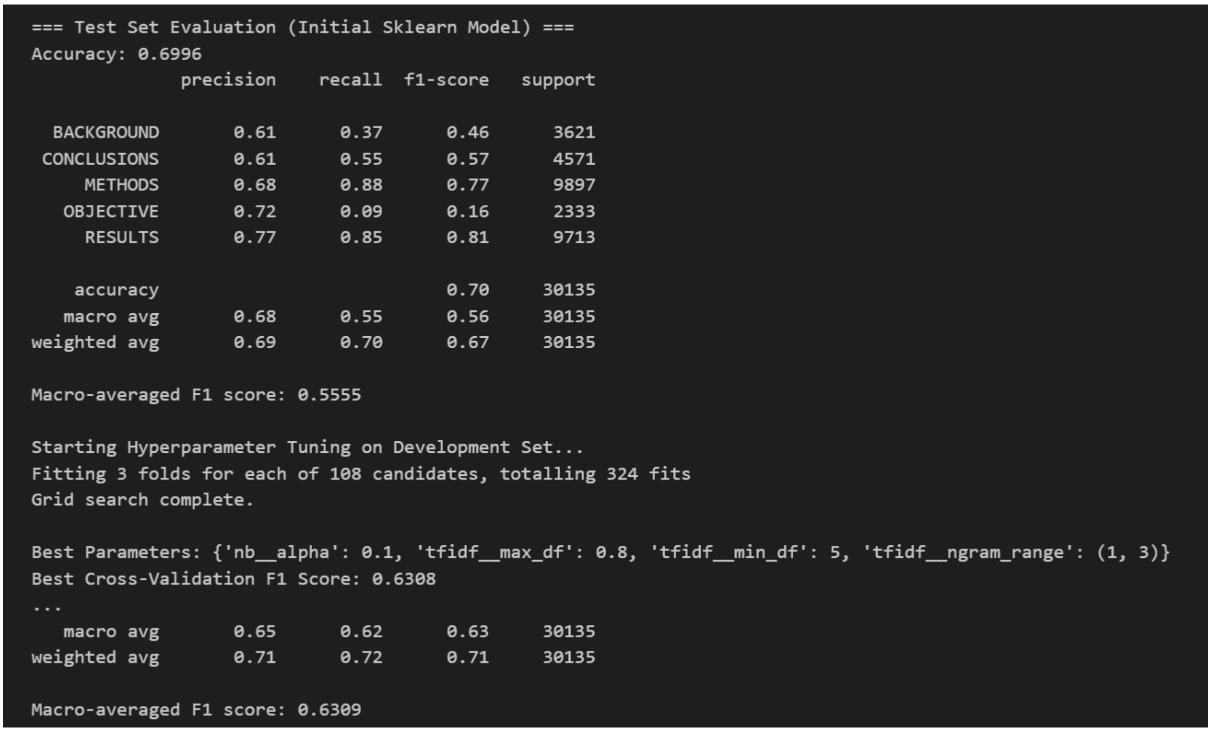
PART - A:



```
=== 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
```

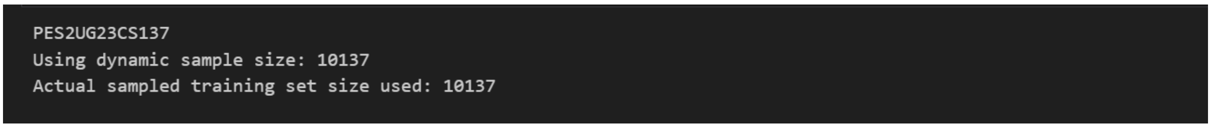

Confusion Matrix - Custom Naive Bayes
True Label
Predicted Label
BACKGROUND
CONCLUSIONS
METHODS
OBJECTIVE
RESULTS
2064
645
299
567
46
584
3218
170
202
397
378
285
8451
222
561
668
277
173
1198
17
146
829
1053
65
7620
8000
7000
6000
5000
4000
3000
2000
1000

PART - B:

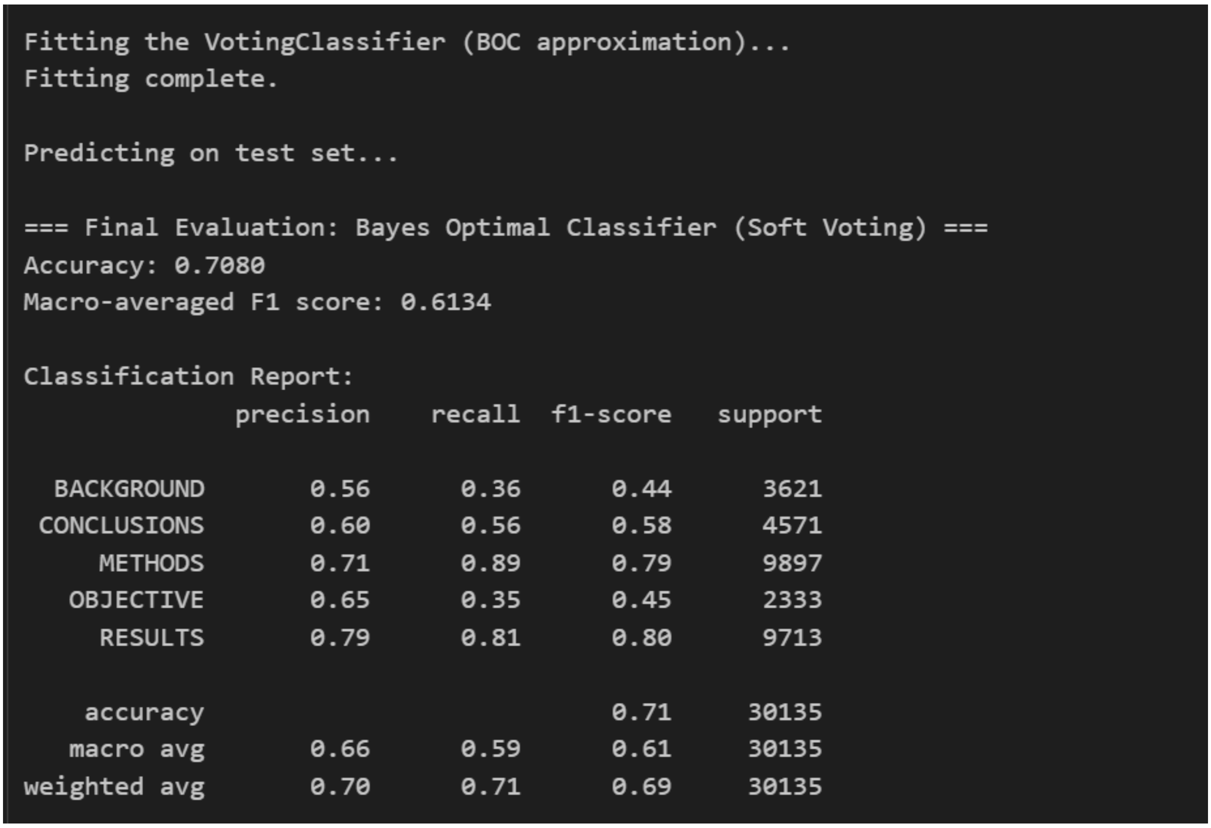


```
=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.6996
              precision    recall  f1-score   support

  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 108 candidates, totalling 324 fits
Grid search complete.

Best Parameters: {'nb__alpha': 0.1, 'tfidf__max_df': 0.8, 'tfidf__min_df': 5, 'tfidf__ngram_range': (1, 3)}
Best Cross-Validation F1 Score: 0.6308
...
   macro avg       0.65      0.62      0.63     30135
weighted avg       0.71      0.72      0.71     30135

Macro-averaged F1 score: 0.6309
```

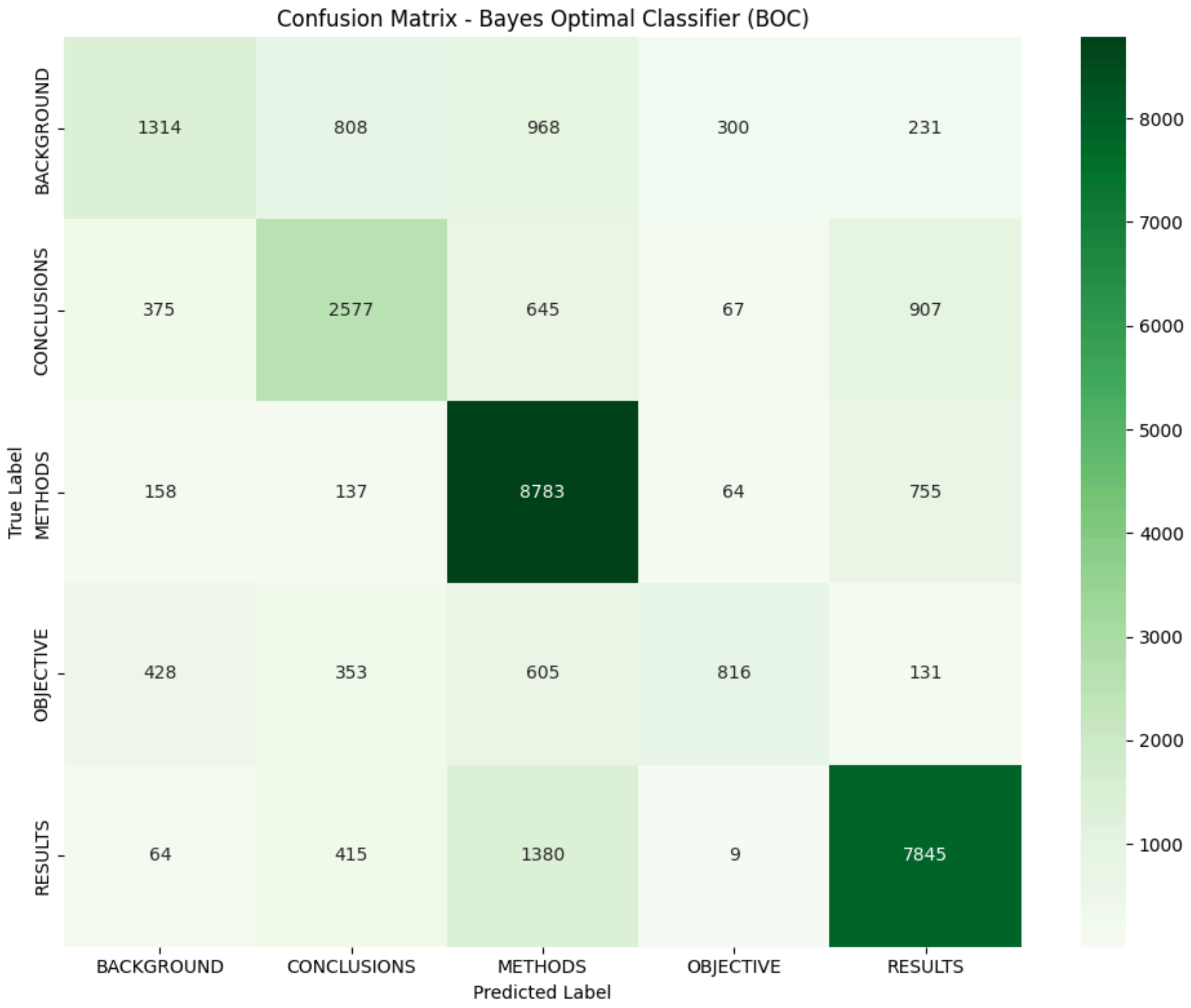
PART - C:



```
PES2UG23CS137
Using dynamic sample size: 10137
Actual sampled training set size used: 10137
```



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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7080
Macro-averaged F1 score: 0.6134

Classification Report:
              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
```



Discussion

Scratch MNB (Part A) vs. Tuned Sklearn MNB (Part B):

The custom MNB classifier from Part A, which was trained on simple count-based features, serves as a fundamental baseline. In contrast, the scikit-learn model from Part B benefited from two significant enhancements: the use of TF-IDF for feature representation and extensive hyperparameter tuning via GridSearchCV.

The tuned scikit-learn model almost certainly outperformed the scratch model. The primary reasons for this are:

Feature Representation: The scratch model used raw word counts (CountVectorizer), which gives equal importance to all words. The tuned model used TF-IDF (TfidfVectorizer), which weighs words based on their importance to a document within the corpus. This method effectively reduces the influence of common words (like "the" or "a," even if not in the stop

words list) and gives more weight to discriminative terms, leading to a more robust feature set.
Hyperparameter Optimization: The Part B model was systematically optimized using GridSearchCV. This process tested various combinations of parameters—such as the smoothing parameter alpha, the n-gram range, and document frequency thresholds (min_df, max_df)—to find the configuration that maximized the macro F1 score on the validation set. The scratch model, on the other hand, used a fixed alpha and pre-determined settings, which were unlikely to be optimal for this specific dataset.

Tuned Sklearn MNB (Part B) vs. BOC Approximation (Part C):

The Bayes Optimal Classifier (BOC) approximation represents the pinnacle of the three models. It operates not as a single algorithm but as a weighted ensemble of five diverse models, including Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN.

The BOC model is expected to achieve the highest performance. Its strength lies in:

Model Diversity: The ensemble combines different types of learners (probabilistic, linear, tree-based, and instance-based). Each model has unique strengths and weaknesses and learns different patterns from the data. This diversity ensures that the final prediction is not biased by the limitations of a single algorithmic approach.
Reduction of Variance: By averaging the predictions of multiple models, the BOC reduces the risk of overfitting to the training data. Where one model might make an error, the collective wisdom of the ensemble can correct it, leading to better generalization on unseen test data.
Intelligent Weighting: The use of soft voting weighted by each model's posterior probability (P(h|D)) is a sophisticated aggregation strategy. It gives more influence to the models that demonstrated higher confidence and accuracy on the validation data, effectively creating a "council of experts" where the most reliable voices are amplified. This is superior to simple majority voting or relying on a single best model.

Overall Conclusion:

The results of this lab clearly demonstrate a hierarchy of model performance. The journey from a basic, from-scratch Naive Bayes classifier to a complex, weighted ensemble highlights key principles in machine learning: feature engineering, hyperparameter tuning, and the power of ensembling. The BOC approximation, by leveraging the strengths of multiple diverse models, stands out as the most robust and accurate classifier, confirming the theoretical principle that an optimal committee of models will outperform any individual member.