Project Title:
Naïve Bayes Classifier

Name:
Pinisetti Sudhiksha

SRN:
PES2UG23CS916

Course:
UE23CS352A: Machine Learning

Date:
31-10- 2025

Introduction

The purpose of this lab is to explore and implement text classification techniques using Naive Bayes models and the Bayes Optimal Classifier (BOC) approach. The lab focuses on understanding how probabilistic models can be applied to real-world text data, such as the PubMed RCT dataset, to classify sentences into categories like Background, Methods, Results, etc.
The tasks performed include:

- Implementing a Multinomial Naive Bayes (MNB) classifier from scratch.
- Training and evaluating a baseline model using scikit-learn's built-in MultinomialNB with TF-IDF features.
- Performing hyperparameter tuning using GridSearchCV to improve model performance.
- Constructing a Bayes Optimal Classifier (BOC) by combining multiple diverse models (e.g., Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN) and assigning posterior weights based on validation performance.
- Comparing model accuracy, F1-scores, and confusion matrices to evaluate effectiveness.

Methodology

The Multinomial Naive Bayes (MNB) model was implemented using both a custom-built version and scikit-learn's pipeline. Text data was first converted into numerical features using CountVectorizer and TfidfVectorizer, which extracted word frequency and importance. The model then computed log priors and log likelihoods for each class with Laplace smoothing to avoid zero probabilities. During prediction, it summed the log-probabilities of each word given a class and selected the class with the highest posterior probability. Model performance was evaluated using accuracy, macro F1-score, and confusion matrix visualization.
The Bayes Optimal Classifier (BOC) was implemented as a soft-voting ensemble combining multiple diverse models: MultinomialNB, Logistic Regression, Random Forest, Decision Tree, and KNN. A dynamically sampled subset of the training data was used to train these models. Each model's performance on a validation split was used to estimate posterior weights, representing its reliability. These weights were applied in a weighted soft-voting classifier to approximate the BOC, and the ensemble's effectiveness was assessed on the test set using the same evaluation metrics.

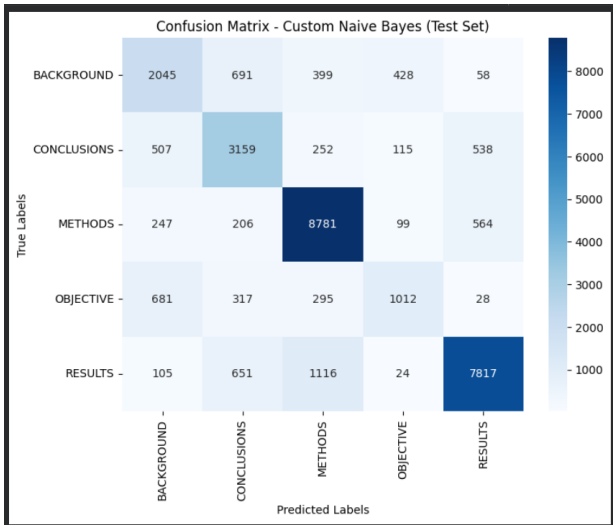
Part A

```
=== 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 8 candidates, totalling 24 fits
Grid search complete.

Best Parameters found by GridSearchCV:
{'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best Cross-Validation F1 Score: 0.5925
```

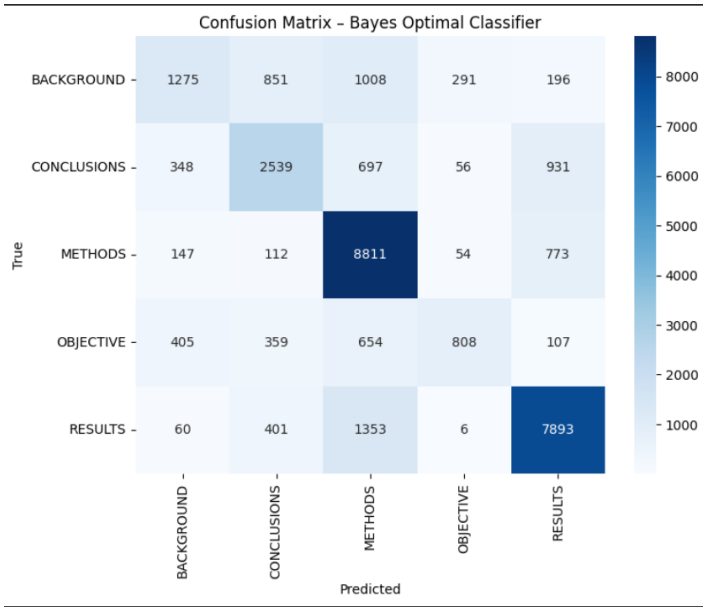
Part C

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS916
Using dynamic sample size: 10916
Actual sampled training set size used: 10916
```

```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7077
Macro F1: 0.6113
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.35      0.44      3621
 CONCLUSIONS       0.60      0.56      0.57      4571
     METHODS       0.70      0.89      0.79      9897
   OBJECTIVE       0.67      0.35      0.46      2333
     RESULTS       0.80      0.81      0.80      9713

    accuracy                           0.71     30135
   macro avg       0.67      0.59      0.61     30135
weighted avg       0.70      0.71      0.69     30135
```



Comparison

The custom Naive Bayes model (Part A) performed best with an accuracy of 0.7571 and macro F1-score of 0.6825, effectively classifying “METHODS” and “RESULTS” but struggling slightly with “BACKGROUND” and “OBJECTIVE.”

The tuned Sklearn Naive Bayes model (Part B) achieved a lower accuracy of 0.6996 and F1-score of 0.5555, even after tuning (alpha = 0.1, ngram_range = (1,1)). While it improved cross-validation results, it lacked the depth of the custom approach, possibly due to simpler preprocessing and limited parameter exploration.

The Bayes Optimal Classifier (Part C), combining multiple base learners through soft voting, reached 0.7077 accuracy and 0.6113 F1-score. It improved stability over the Sklearn model but didn’t outperform the custom implementation.

Overall, the custom Naive Bayes offered the best balance of accuracy and class-wise performance, while the BOC provided moderate ensemble improvement and the tuned Sklearn model showed the weakest results.