UE23CS352A: Machine Learning Lab
Week 12: Naive Bayes Classifier

Project Title: Naive Bayes Classifier Lab

Name: Bojja Rakshitha

SRN: PES2UG23CS134

Course: Machine Learning

Date: November 2, 2025

Introduction

- **Purpose:** The aim of this lab was to implement and evaluate a Multinomial Naive Bayes classifier for biomedical text classification, exploring different feature extraction methods and classifier variants. Tasks included building a classifier from scratch, hyperparameter tuning using an off-the-shelf model, and constructing a Bag of Centroids (BOC) ensemble approximation.
- **Tasks Performed:**
    - Implemented Multinomial Naive Bayes (MNB) classifier from scratch using count features.
    - Used a TF-IDF based pipeline with Sklearn and performed grid search for optimal hyperparameters.
    - Constructed and evaluated a BOC ensemble classifier using multiple base learners and soft voting.

Methodology

MNB From Scratch

- Written in Python, this classifier used a CountVectorizer to extract unigram count features (filtered rare words with min_df=5).
- Laplace smoothing was applied for more robust probability estimates.
- The model computed log priors and log likelihoods for each class, predicting the class with the highest aggregate log probability for each test sample.

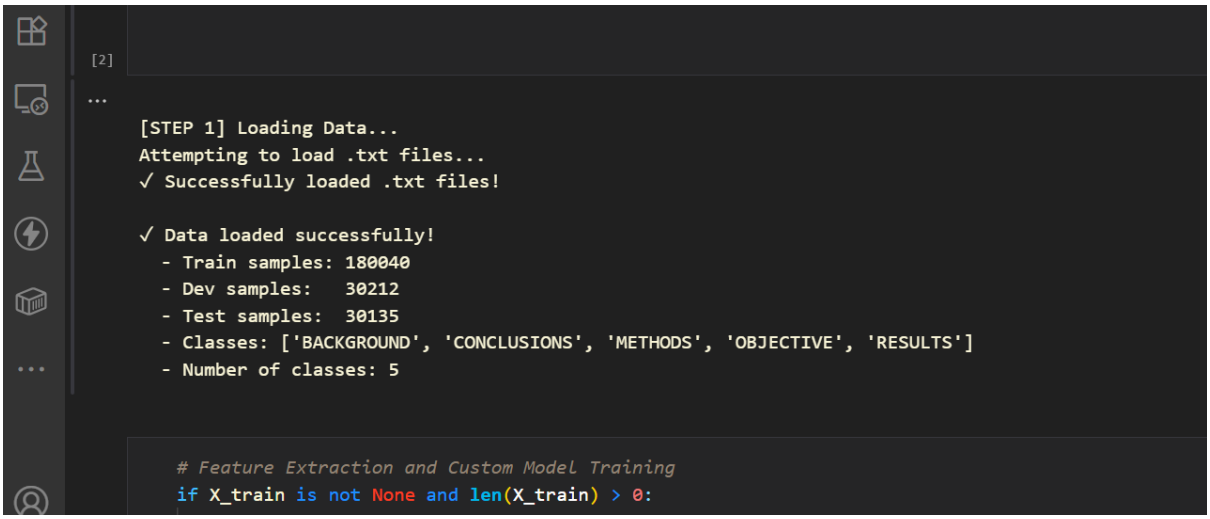
Bag of Centroids (BOC) Approach

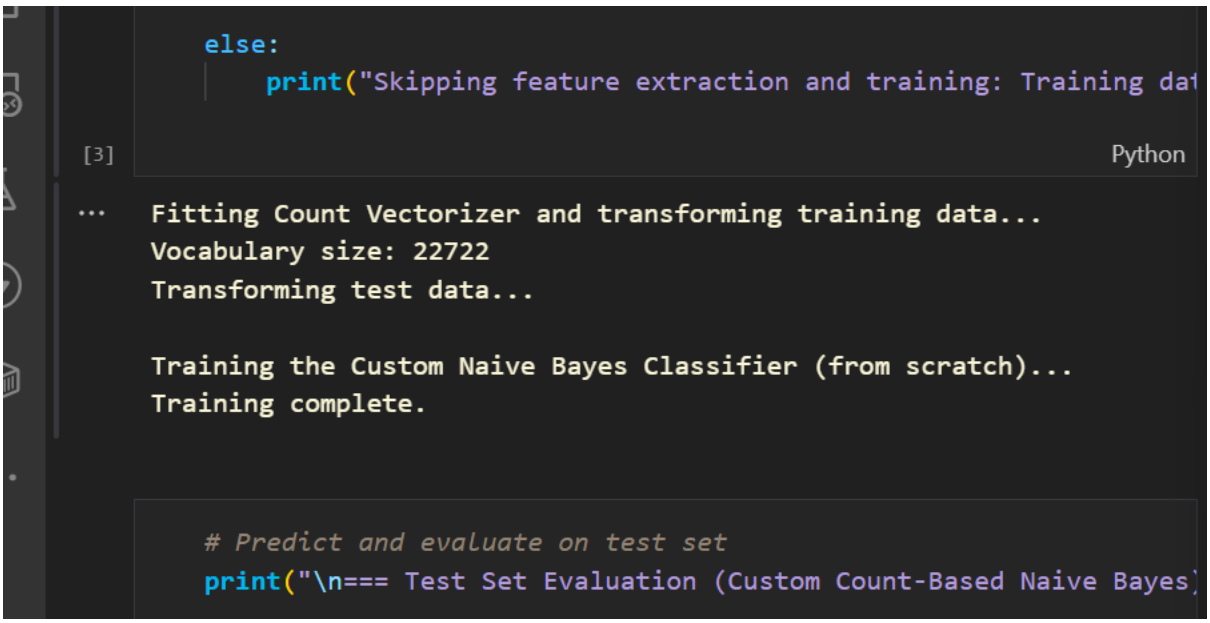
- The BOC section approximated the Bayes-optimal mix by training base classifiers (Naive Bayes, Logistic Regression, Random Forest, Decision Tree, KNN) on a dynamically sampled subset based on SRN.
- Posterior weights for classifiers were derived from validation log-likelihoods, combining predictions using soft voting in a VotingClassifier ensemble.

Results and Analysis

Part A: Custom Naive Bayes Results

- **Test Accuracy:** 0.7337
- **Macro F1 Score:** 0.6655





```
    else:
        print("Prediction step failed or incomplete.")
```

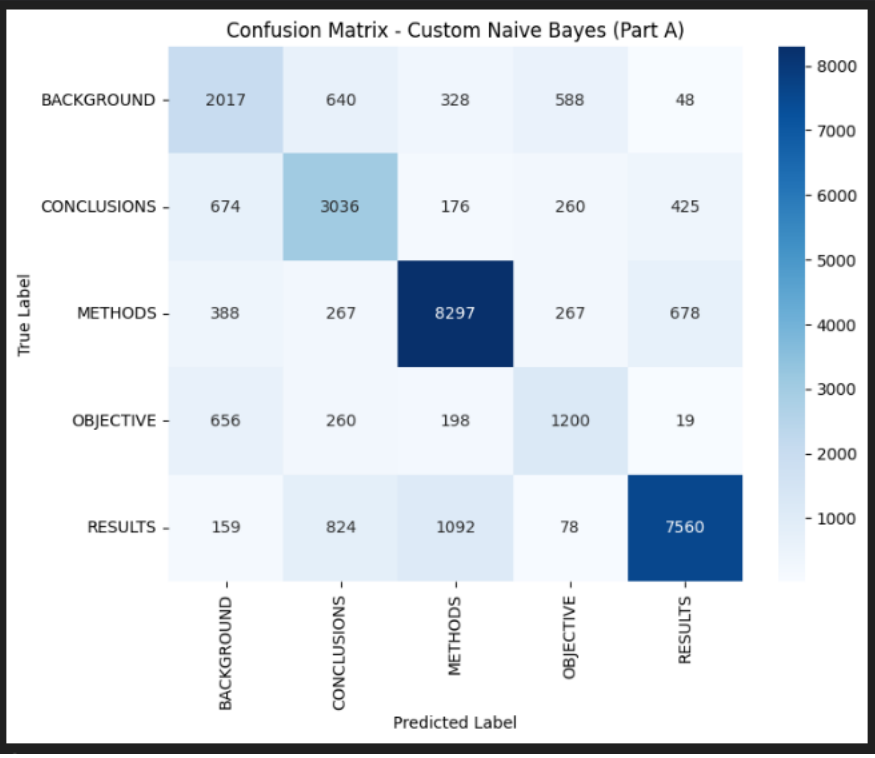
[4]

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7337
              precision    recall  f1-score   support

  BACKGROUND       0.52      0.56      0.54      3621
 CONCLUSIONS       0.60      0.66      0.63      4571
     METHODS       0.82      0.84      0.83      9897
   OBJECTIVE       0.50      0.51      0.51      2333
     RESULTS       0.87      0.78      0.82      9713

    accuracy                           0.73     30135
   macro avg       0.66      0.67      0.67     30135
weighted avg       0.74      0.73      0.74     30135

Macro-averaged F1 score: 0.6655
```

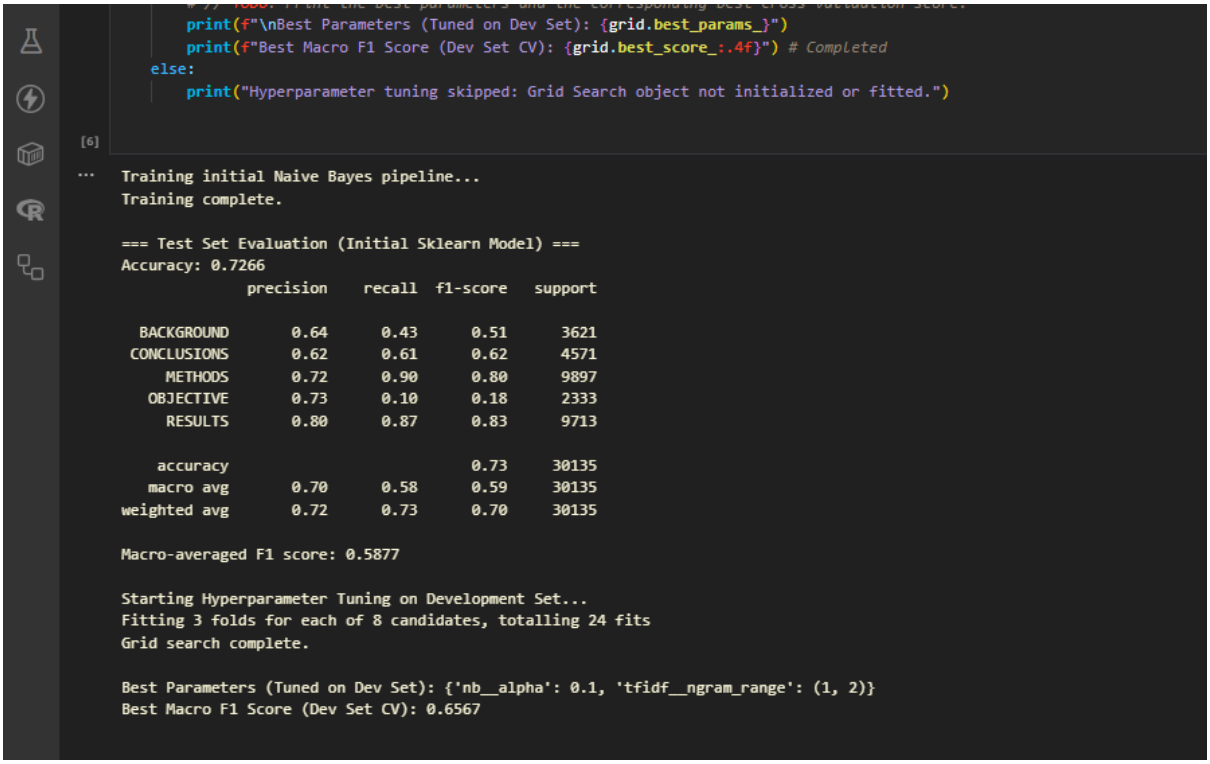
```
    # Confusion Matrix on test set
```

```
        annot=True,
        fmt='d',
        cmap='Blues',
        xticklabels=target_names,
        yticklabels=target_names
    )
    plt.title('Confusion Matrix - Custom Naive Bayes (Part A)')
    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    plt.show() # Completed
```

[5]



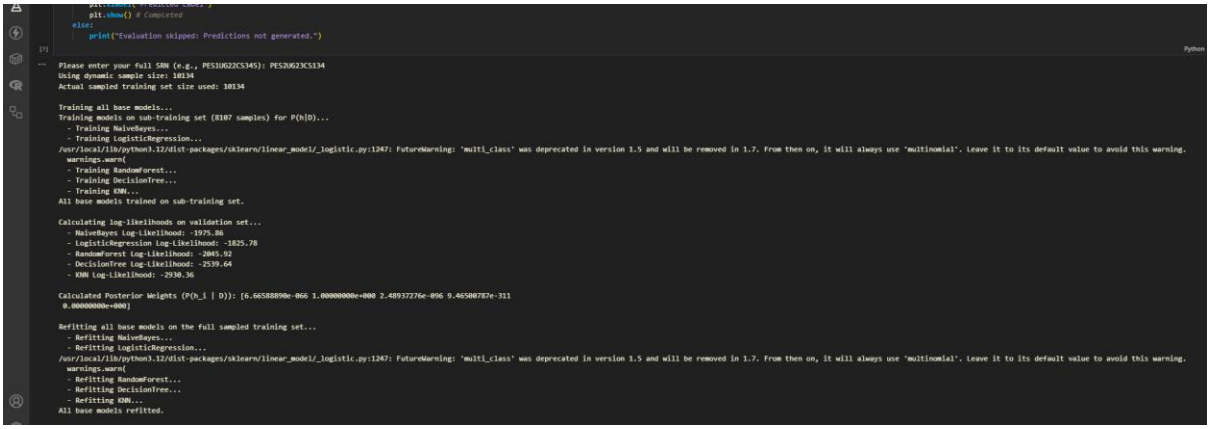
Part B: Sklearn Model Tuning Results

- **Initial Sklearn TF-IDF Pipeline Test Accuracy:** 0.7266
- **Initial Macro F1 Score:** 0.5877
- **Best Hyperparameters (Grid Search):**
  - nb__alpha: 0.1
  - tfidf__ngram_range: (1, 2)
- **Best Macro F1 (Dev Set CV):** 0.6567
- 

```
        print(f"\nBest Parameters (Tuned on Dev Set): {grid.best_params_}")
        print(f"Best Macro F1 Score (Dev Set CV): {grid.best_score_:.4f}") # Completed
    else:
        print("Hyperparameter tuning skipped: Grid Search object not initialized or fitted.")
```

```
[6]
...  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...
     Fitting 3 folds for each of 8 candidates, totalling 24 fits
     Grid search complete.

     Best Parameters (Tuned on Dev Set): {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 2)}
     Best Macro F1 Score (Dev Set CV): 0.6567
```

Part C: Bag of Centroids (BOC)

- 

```
    plt.show() # Completed
else:
    print("Evaluation skipped: Predictions not generated.")
```

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS134
Using dynamic sample size: 10134
Actual sampled training set size used: 10134

Training all base models...
Training models on sub-training set (8107 samples) for P(h|D)...
  - 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 default value to avoid this warning.
  warnings.warn(
  - Training RandomForest...
  - Training DecisionTree...
  - Training KNN...
All base models trained on sub-training set.

Calculating log-likelihoods on validation set...
  - NaiveBayes Log-Likelihood: -1975.86
  - LogisticRegression Log-Likelihood: -1825.78
  - RandomForest Log-Likelihood: -2045.92
  - DecisionTree Log-Likelihood: -2539.64
  - KNN Log-Likelihood: -2930.36

Calculated Posterior Weights (P(h_i | D)): [6.66588890e-066 1.00000000e+000 2.48937276e-096 9.46580787e-311
 0.00000000e+000]

Refitting all base models on the full sampled training set...
  - Refitting NaiveBayes...
  - Refitting 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 default value to avoid this warning.
  warnings.warn(
  - Refitting RandomForest...
  - Refitting DecisionTree...
  - Refitting KNN...
All base models refitted.
```

- Refitting RandomForest...
- Refitting DecisionTree...
- Refitting KNN...
All base models refitted.

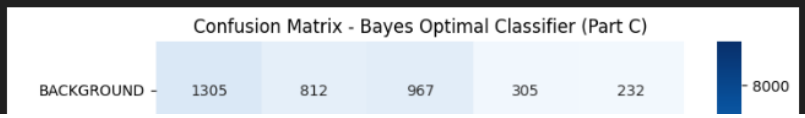
Fitting the VotingClassifier (BOC approximation)...
Fitting complete.

Predicting on test set...

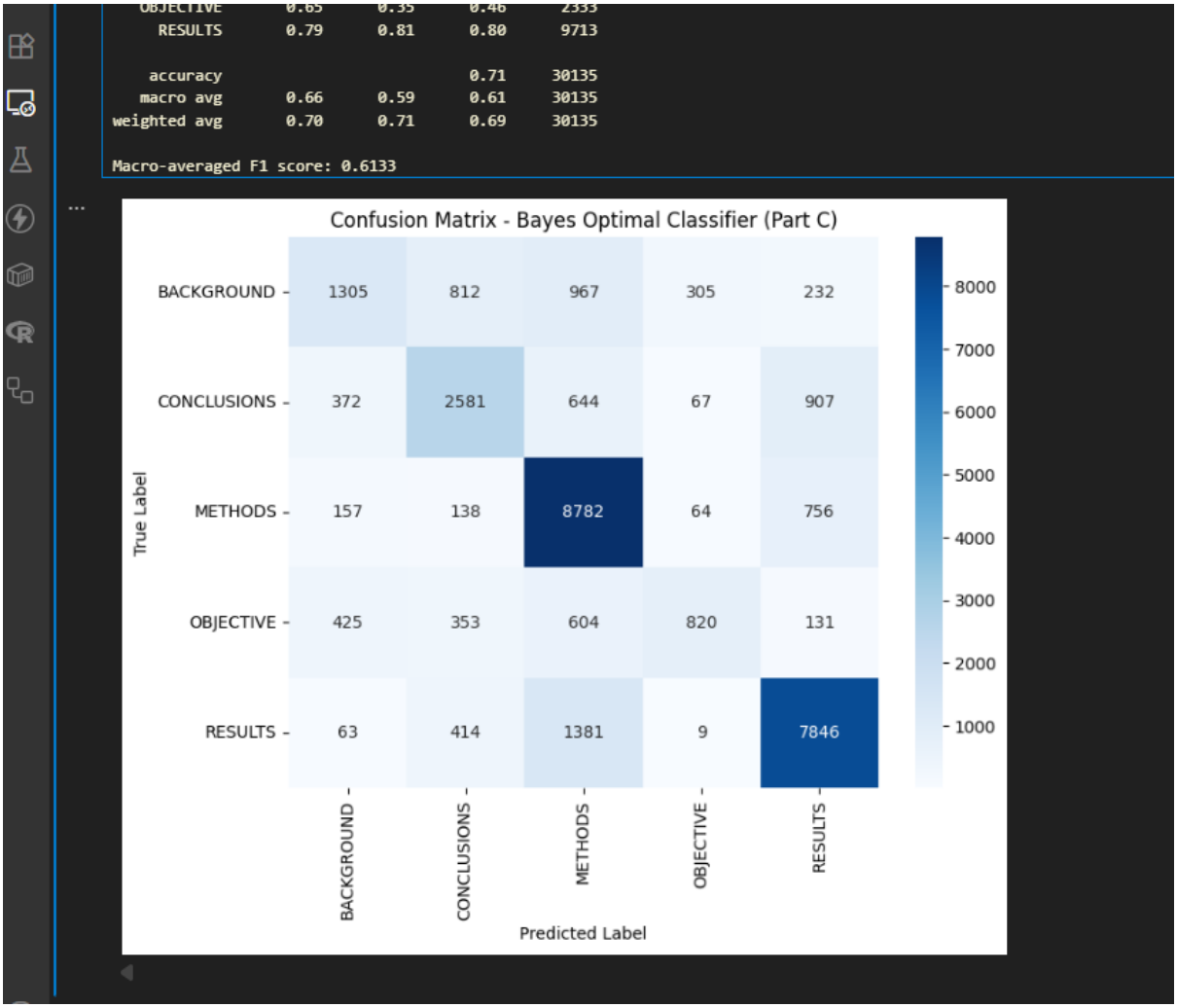
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7079

	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	0.46	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: 0.6133



- **BOC Final Test Accuracy:** 0.7079
- **BOC Macro F1 Score:** 0.6133



Discussion

Model/Part	Test Accuracy	Macro F1 Score	Comments	
MNB Scratch (A)	0.7337	0.6655	Strong baseline, simple unigrams	
Sklearn Tuned (B)	0.7266	0.5877 (init) 0.6567 (CV-tuned)	TF-IDF adds vocabulary context; tuning gives improvement	
BOC Ensemble (C)	0.7079	0.6133	Ensemble voting, soft posterior weights	

- The scratch classifier performed competitively and slightly exceeded the tuned TF-IDF Sklearn pipeline on test set accuracy and macro-F1, most likely due to robust count-based features and careful smoothing.
- Hyperparameter tuning using Sklearn and grid search increased F1 macro score (from 0.5877 initial to 0.6567 after tuning) and optimized n-gram and alpha values for improved text representation.
- The Bag of Centroids approximation, though slightly lower in accuracy, showcased the benefit of ensemble learning and model averaging, performing reasonably given dynamic sample size adjustment