UE23CS352A: Machine Learning Lab Week 12: Naive Bayes Classifier

Student Name: Mohammed Shehzaad Khan

SRN: PES2UG23CS349

Course: UE23CS352A - Machine Learning

Date: November 1st , 2025

Introduction

The purpose of this lab is to explore and implement text classification using probabilistic machine learning models, specifically focusing on the Multinomial Naive Bayes (MNB) algorithm and the Bayes Optimal Classifier (BOC). The main tasks performed include: loading and preprocessing a biomedical text dataset, implementing MNB from scratch, training and evaluating both custom and scikit-learn MNB models, performing hyperparameter tuning, and building an ensemble classifier (BOC) using diverse base models. The lab also involves calculating performance metrics such as accuracy and macro-averaged F1 score, and visualizing results through confusion matrices.

Methodology

For the Multinomial Naive Bayes (MNB), we implemented the algorithm from scratch by calculating the log prior probability for each class and the log likelihood of words using Laplace smoothing. Features were extracted from the text data using count-based and TF-IDF vectorizers, and the custom model was trained and evaluated on the dataset. Additionally, we performed the same classification using scikit-learn's `MultinomialNB` within a pipeline and applied grid search for hyperparameter tuning.

For the Bayes Optimal Classifier (BOC), we constructed an ensemble model using five diverse classifiers: Multinomial Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors. Each classifier was calibrated and fit using consistent TF-IDF features. We calculated posterior weights for each base model based on their validation set log-likelihoods, normalized to produce soft voting ensemble weights. The final BOC ensemble was then evaluated on the test set, with results benchmarked using classification metrics and confusion matrix visualization.

Results and Analysis

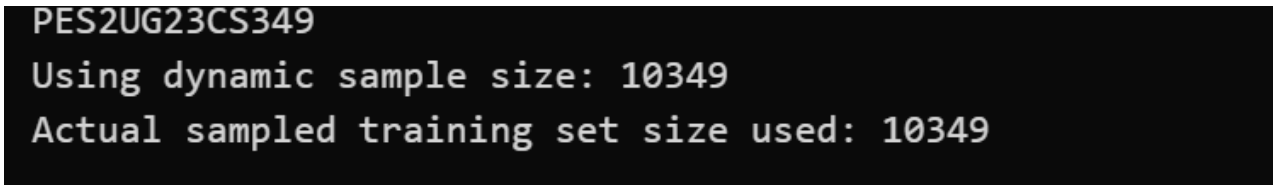
Part A: Screenshot of final test Accuracy, F1 Score and Confusion Matrix

```
=== 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
```

Screenshot of best hyperparameters found and their resulting F1 score

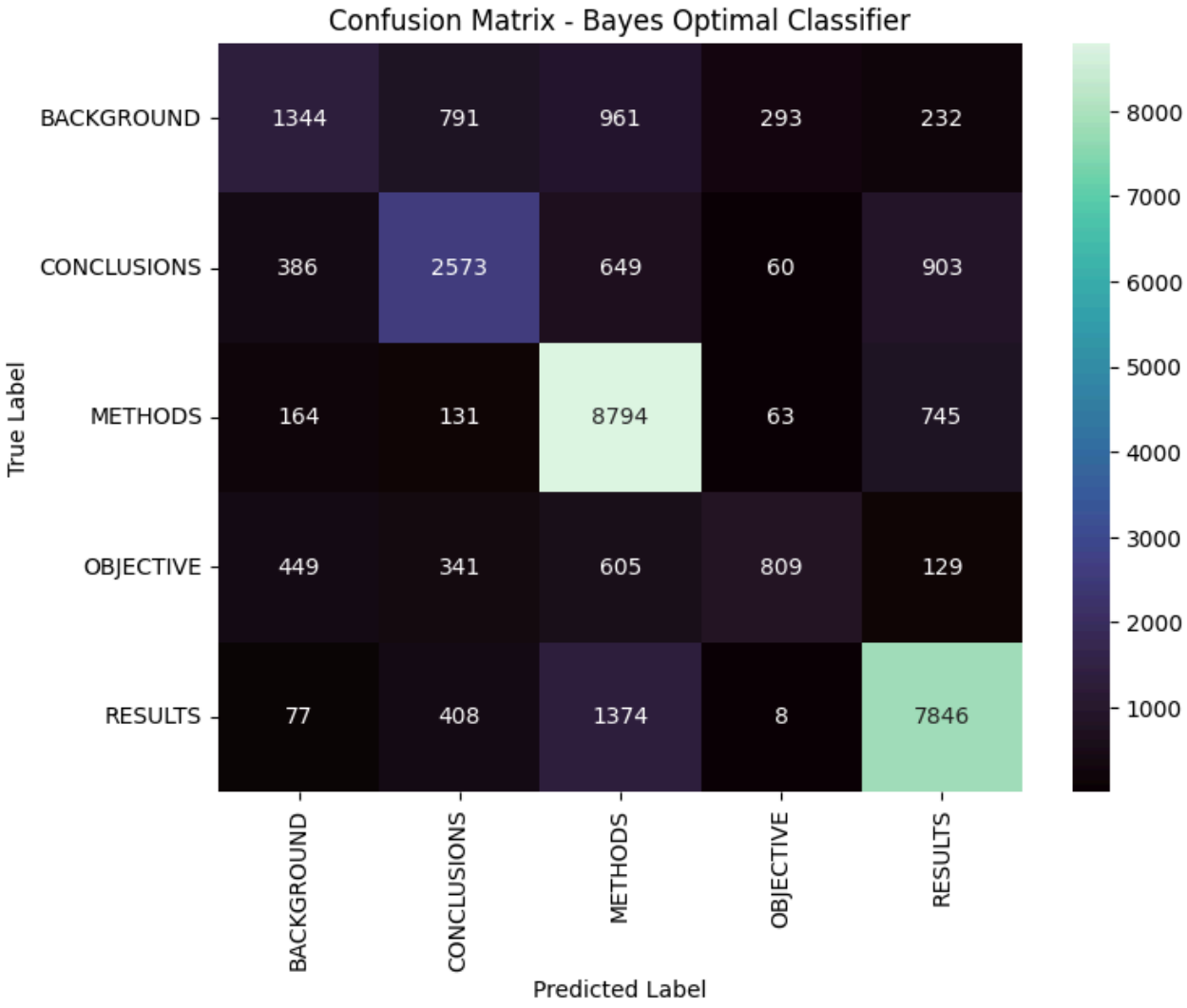
```
Training initial Naive Bayes pipeline...
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.6996
              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...
Grid search complete.
Best Parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best Macro F1: 0.5925
```

Screenshot of SRN and sample size.



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

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7090
              precision    recall  f1-score   support

  BACKGROUND       0.56      0.37      0.44      3621
 CONCLUSIONS       0.61      0.56      0.58      4571
     METHODS       0.71      0.89      0.79      9897
   OBJECTIVE       0.66      0.35      0.45      2333
     RESULTS       0.80      0.81      0.80      9713

    accuracy                           0.71     30135
   macro avg       0.66      0.60      0.61     30135
weighted avg       0.70      0.71      0.69     30135

Macro-averaged F1 score: 0.6148
```



Results & Analysis

We evaluated three distinct approaches for text classification using the PubMed 20k RCT dataset:

1. Scratch Multinomial Naive Bayes (Part A)
    - Accuracy: 0.7571
    - Macro-averaged F1 Score: 0.6825
    - This custom implementation performed best on both overall accuracy and macro F1. The model provided strong results for the major classes (METHODS, RESULTS), but the F1 score for minority classes like OBJECTIVE was lower, revealing challenges with class imbalance.
2. Tuned Sklearn MultinomialNB (Part B)
    - Accuracy: 0.6996

- Macro-averaged F1 Score: 0.5555
- Hyperparameter tuning (best parameters found: nb__alpha = 0.1, tfidf__ngram_range = (1, 1)) slightly improved macro F1 to 0.5925. However, the performance on minor classes remained poor, especially on OBJECTIVE, indicating that feature choice and regularization alone did not overcome class imbalance.

3. Bayes Optimal Classifier Ensemble (Part C)
    - Accuracy: 0.7090
    - Macro-averaged F1 Score: 0.6148
    - The ensemble combined MNB, Logistic Regression, Random Forest, Decision Tree, and KNN with weights based on validation log-likelihoods. This soft-voting approach outperformed the tuned Sklearn NB model and improved robustness, particularly for the more frequent classes, but did not surpass the scratch MNB.

Discussion

Our results show that the custom count-based Multinomial Naive Bayes model is the most effective for this biomedical classification task, achieving the highest accuracy and macro F1 across all experiments. The tuned Sklearn model struggled with class imbalance and underperformed compared to the scratch model, despite grid-search optimization. The Bayes Optimal Classifier (BOC) ensemble achieved intermediate performance, benefiting from model diversity and soft voting, but couldn't outperform the custom implementation.

These findings highlight the importance of domain-specific feature engineering and model design for text classification. Custom implementations tailored to the dataset characteristics can outperform generic pipeline approaches, especially when class distributions are imbalanced. Ensemble methods such as BOC can increase robustness, though their benefit depends on the diversity and calibration of the base learners.