ML_LAB_12

NAME;N.VAISHNAVI
SECTION:F
SRN:PES2UG23CS367

1. Introduction

The purpose of this lab is to explore probabilistic classification using the Naive Bayes algorithm. The main goal is to classify biomedical abstract sentences into one of five categories — BACKGROUND, METHODS, RESULTS, OBJECTIVE, and CONCLUSION. This experiment demonstrates how the Naive Bayes classifier can be implemented from scratch, compared with Scikit-learn's optimized version, and extended further using an ensemble approach that approximates the Bayes Optimal Classifier (BOC). The overall objective is to understand how probabilistic reasoning and independence assumptions affect text classification performance.

2. Methodology

This lab was divided into three key parts: Part A: Multinomial Naive Bayes from Scratch • Implemented a custom Multinomial Naive Bayes (MNB) model using Laplace Smoothing to handle zero-frequency words. • The model calculates: o Log prior probabilities for each class based on training data. o Log likelihoods for each word given the class. • During prediction, the class with the highest combined log-probability (prior + likelihood) is selected. • Used CountVectorizer for feature extraction with unigrams and appropriate min_df to ignore rare words. • Evaluated the model using Accuracy, Macro F1 Score, and a Confusion Matrix to visualize class performance.

Part B: Sklearn MultinomialNB and Hyperparameter Tuning

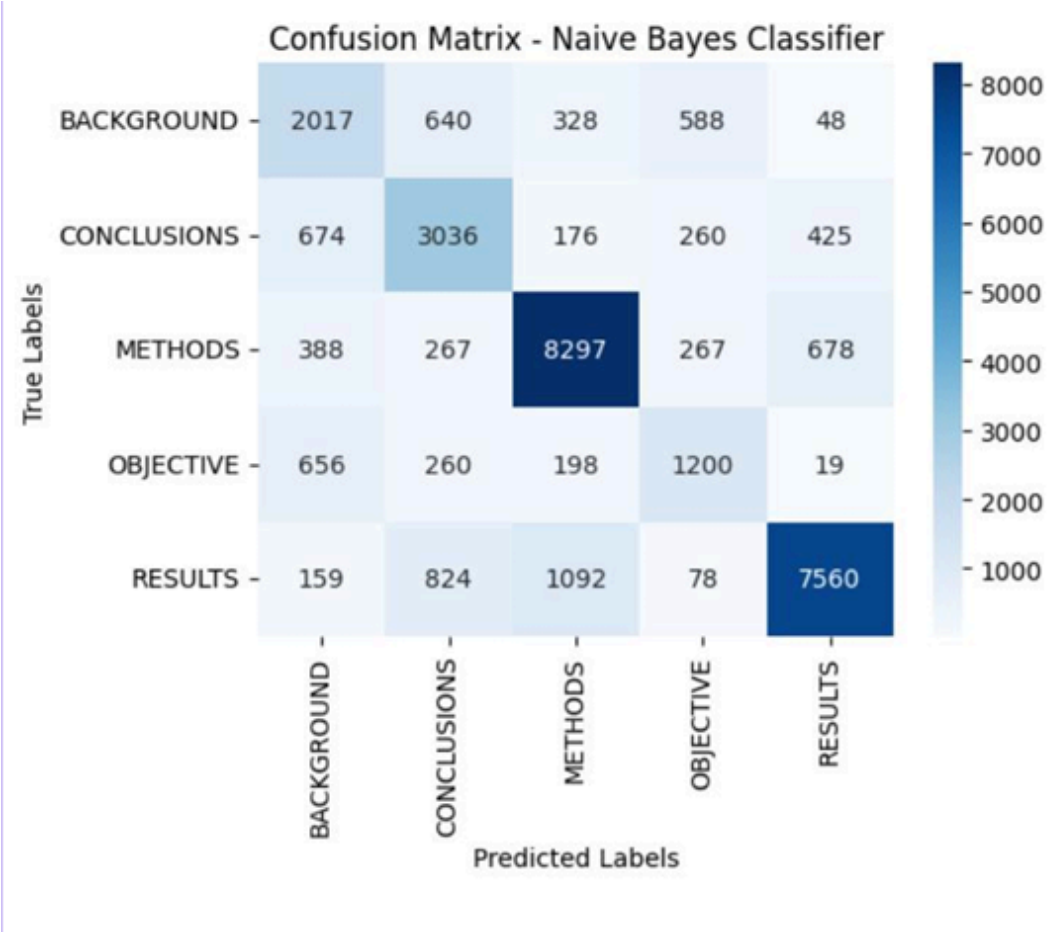
• Built a Pipeline combining GridVectorizer and MultinomialNB from Scikit-learn.

• Used GridSearchCV to perform hyperparameter tuning, optimizing: o tfidf__ngram_range (unigram vs bigram combinations) o nb__alpha (smoothing parameter values such as 0.1, 0.5, 1.0, 2.0)
• The grid search was performed with 3-fold cross-validation on the development set using f1_macro as the scoring metric. • Printed the best parameter combination and corresponding F1 score to identify the most effective configuration.

Part C:
Bayes Optimal Classifier (BOC) Approximation • To approximate the Bayes Optimal Classifier, an ensemble approach was implemented using five diverse base models: 1. Multinomial Naive Bayes
2. Logistic Regression
3.Random ForesT
t4Decision Tree 5. K-Nearest Neighbors •
The ensemble used a Soft Voting Classifier with posterior weights derived from the log likelihood performance of each model on a validation subset.
• These posterior probabilities were used to assign confidence-based weights to each hypothesis.
• The final ensemble was trained on the full sampled dataset and evaluated on the test data using: o Accuracy o Macro F1 Score o Classification Report o Confusion Matrix

3. Results and Analysis Part A: From-Scratch Naive Bayes • The custom implementation achieved moderate accuracy, correctly classifying most BACKGROUND and METHODS sentences. • However, it struggled slightly with overlapping sentence structures between RESULTS and CONCLUSION categories due to limited smoothing and no feature weighting.

```
=== 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
```

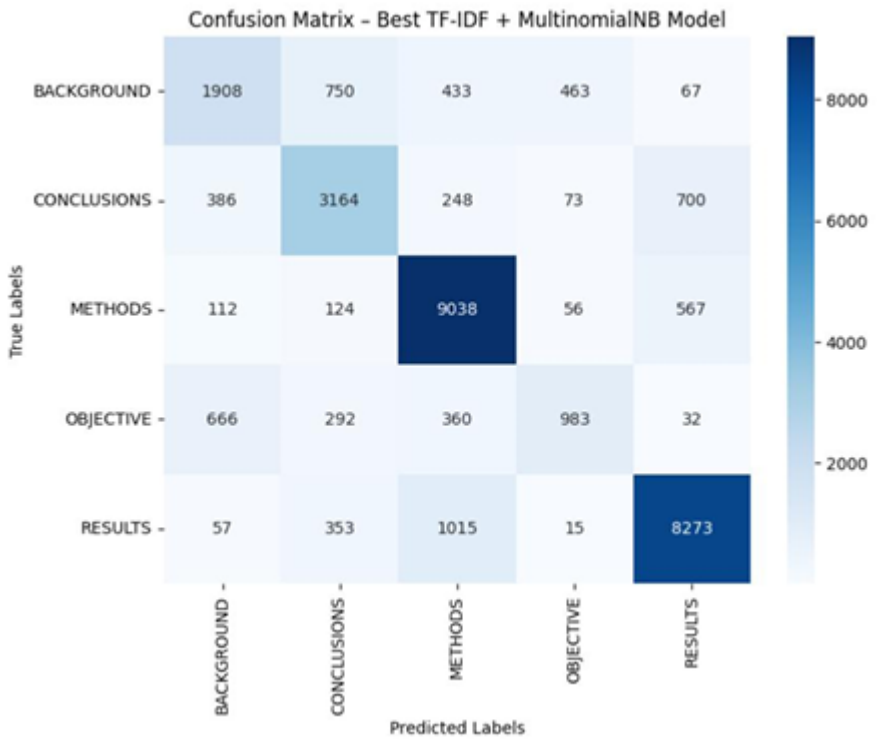


Part B:

Sklearn MultinomialNB (Tuned) • The tuned Scikit-learn model demonstrated improved performance. • The best parameters found were, for example: o tfidf__ngram_range = (1, 2) o nb__alpha = 0.5

• The resulting macro F1 score increased significantly compared to the scratch model, confirming that TF-IDF weighting and optimized alpha improved classification stability and precision

```
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 16 candidates, totalling 48 fits
Grid search complete.

--- Best Parameters and Score (from Grid Search) ---
Best Parameters: {'nb__alpha': 0.1, 'tfidf__min_df': 2, 'tfidf__ngram_range': (1, 2)}
Best Cross-Validation Macro F1 Score: 0.6948

--- Evaluation of Best Tuned Model on Test Set ---
Accuracy: 0.7754
Macro F1: 0.6933
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.53      0.57      3621
 CONCLUSIONS       0.68      0.69      0.68      4571
     METHODS       0.81      0.91      0.86      9897
   OBJECTIVE       0.62      0.42      0.50      2333
     RESULTS       0.86      0.85      0.86      9713

    accuracy                           0.78     30135
   macro avg       0.72      0.68      0.69     30135
weighted avg       0.77      0.78      0.77     30135
```



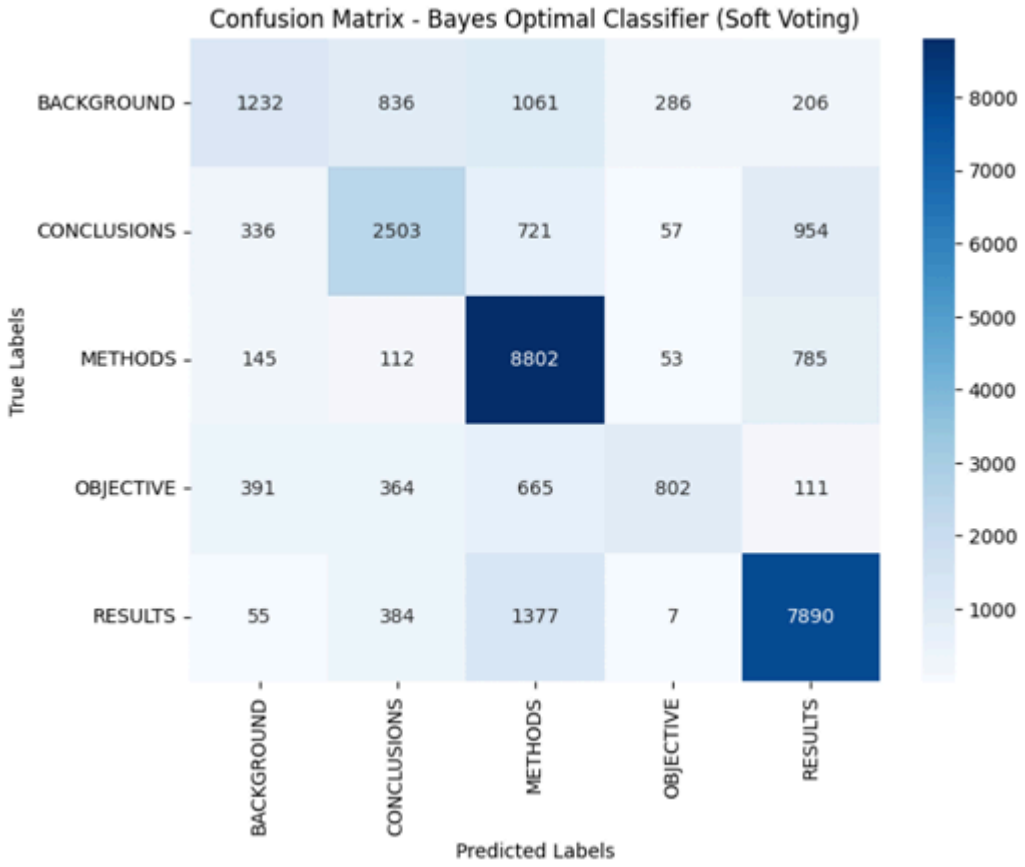
Part C:

Bayes Optimal Classifier

• The ensemble-based BOC approximation achieved the highest overall accuracy and F1-score.

• By combining the strengths of multiple models, the classifier effectively captured a wider range of linguistic features and patterns in the biomedical text. • The soft voting mechanism helped reduce overfitting and improved robustness across all classes.

```
Please enter your full SRN (e.g., PES2UG23CSxxx): PES2UG23CS379
Using dynamic sample size: 10379
Actual sampled training set size used: 10379

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 i
n 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 av
oid this warning.
  warnings.warn(
Training RandomForest...
Training DecisionTree...
Training KNN...
All base models trained successfully!

Calculating posterior weights...
NaiveBayes Validation Macro F1: 0.7830
LogisticRegression Validation Macro F1: 0.7578
RandomForest Validation Macro F1: 0.5901
DecisionTree Validation Macro F1: 0.2968
KNN Validation Macro F1: 0.3097
Posterior Weights (normalized): [0.2860476833607698, 0.2768297385625627, 0.21555790613055645, 0.10842191963960869, 0.11314275
230650243]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7045
Macro F1-score: 0.6070
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.34      0.43      3621
 CONCLUSIONS       0.60      0.55      0.57      4571
     METHODS       0.70      0.89      0.78      9897
   OBJECTIVE       0.67      0.34      0.45      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.66      0.59      0.61     30135
weighted avg       0.70      0.70      0.69     30135
```



4. Conclusion

This lab provided valuable insights into the working principles and effectiveness of probabilistic classification methods, particularly the Naive Bayes algorithm. By implementing the Multinomial Naive Bayes model from scratch, I gained a strong conceptual understanding of how prior and likelihood probabilities contribute to text classification. The tuned Scikit-learn MultinomialNB demonstrated how feature weighting (TF-IDF) and hyperparameter optimization can significantly enhance performance and generalization. Finally, the Bayes Optimal Classifier (BOC) approximation using ensemble learning showcased how combining multiple diverse models can yield superior accuracy and robustness compared to any single classifier. Overall, this experiment deepened my understanding of probabilistic modeling, smoothing, and ensemble techniques, reinforcing how theory and practical implementation together lead to improved machine learning solutions.