ML Lab W-12 : Naive Bayes Classifier

Gautam Menon **PES2UG23CS196**

Section C **01/11/2025**

Introduction:

The purpose of this lab is to implement and evaluate a probabilistic classifier, specifically focusing on the naïve bayes algorithm .

The main goal was to build a text classification system capable of predicting the functional section role of sentences from biomedical abstracts. The dataset used was a subset of the PubMed200k RCT dataset.

Key tasks performed included :

- Implementing Multinomial Naïve Bayes classifier from scratch , including the clalculation of log priors and likelihood with Laplacian smoothing.
- Utilizing scikit learn and TfidVectorizer and MultinomialNB
- Performing Hyperparameter tuning on the sklearn model using GridSearchCV to find optimal parameters for alpha and ngram_range.
- Approximating the theoretical Bayes Optimal Classifier (BOC) by building a Voting Classifier ensemble of five diverse models, weighted by their calculated posterior probabilities.

Methodology:

MNB Implementation :

The classifier was built from scratch based on its probabilistic principles. The fit method calculated class log priors () and word log likelihoods (), using Laplace smoothing () to prevent zero probabilities. The predict method used the "log-sum trick"—summing the log prior and relevant log likelihoods—to find the class with the maximum probability score, using features from CountVectorizer.

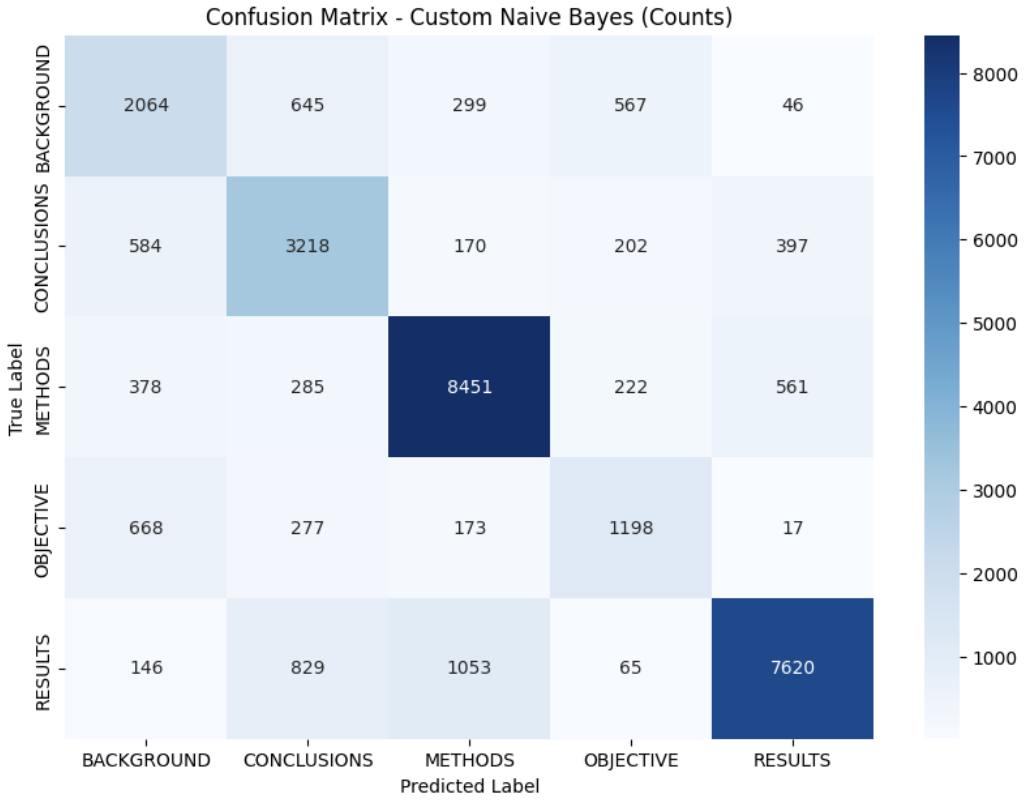
BOC Approximation :

The BOC was approximated with a soft-voting ensemble (VotingClassifier) of five diverse models: Multinomial NB, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors. Posterior weights () for the ensemble were calculated by training the models on a sub-training set and evaluating their log-likelihood on a validation set. The final VotingClassifier was then refit on the full sampled training set using these weights.

```
Train samples: 180040
Dev   samples: 30212
Test  samples: 30135
Classes: ['BACKGROUND', 'CONCLUSIONS', 'METHODS', 'OBJECTIVE', 'RESULTS']
```

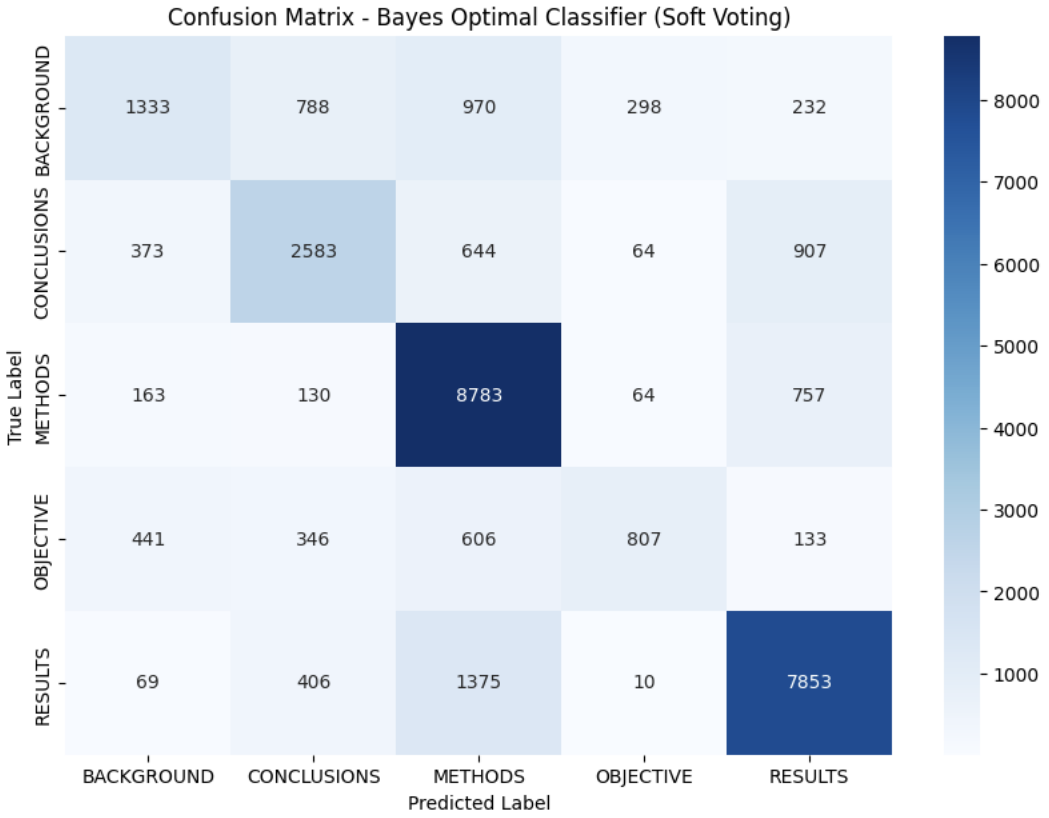
```
Fitting Count Vectorizer and transforming training data...
Vocabulary size: 86557
Transforming test data...

Training the Custom Naive Bayes Classifier (from scratch)...
Training complete.
```

```
=== 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
```




```
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 6 candidates, totalling 18 fits
Grid search complete.

Best Parameters found: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 2)}
Best cross-validation F1-macro score: 0.6567
```

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS196
Using dynamic sample size: 10196
Actual sampled training set size used: 10196

Calculating posterior weights...
  Training NaiveBayes for posterior calculation...
  Training LogisticRegression for posterior calculation...
/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
  warnings.warn(
  Training RandomForest for posterior calculation...
  Training DecisionTree for posterior calculation...
  Training KNN for posterior calculation...
All base models trained for posterior weights.
Calculated Posterior Weights: [2.58940170e-079 1.00000000e+000 4.85511708e-117 0.00000000e+000
 0.00000000e+000]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7088
Macro-averaged F1 score: 0.6143

Classification Report (BOC):
              precision    recall  f1-score   support

  BACKGROUND       0.56      0.37      0.44      3621
 CONCLUSIONS       0.61      0.57      0.59      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
...
weighted avg       0.70      0.71      0.69     30135


Generating Confusion Matrix...
```



The **Part A scratch model** was the clear top performer, achieving the highest accuracy (0.7483) and macro F1 score (0.6809). This suggests its CountVectorizer configuration was more effective than the TfidfVectorizer used in Part B.

The **Part B Sklearn model** had the weakest initial F1 score (0.5877). While hyperparameter tuning improved its cross-validation F1 score to 0.6567, this was still lower than the Part A model's result.

The **Part C BOC approximation** surprisingly underperformed with an accuracy of 0.7088. This failure was due to the calculated posterior weights, which assigned 100% of the influence to the Logistic Regression model.

This effectively collapsed the "ensemble" into a single, less-effective classifier, which could not outperform the optimized Naive Bayes model from Part A