UE23CS352A: Machine Learning Lab

Week 12: Naive Bayes Classifier

NAME : MOULYA K A

SRN : PES2UG23CS351

DATE:01-11-2025

INTRODUCTION :

The purpose of this lab was to explore probabilistic text classification using the Naive Bayes algorithm. The primary tasks were:

1.Implement a Multinomial Naive Bayes (MNB) classifier from scratch to classify sentences from biomedical abstracts into five sections: BACKGROUND, METHODS, RESULTS, OBJECTIVE, and CONCLUSIONS.

2.Utilize scikit-learn's MultinomialNB with TF-IDF features and perform hyperparameter tuning using GridSearchCV.

3.Approximate the Bayes Optimal Classifier (BOC) using an ensemble of diverse models combined with soft voting and posterior weights.

The dataset used is a subset of the PubMed 200k RCT dataset.

METHODOLOGY:

Part A – Custom Multinomial Naive Bayes:

- Implemented MNB from scratch using word counts.
- Calculated log prior probabilities for each class and log likelihoods for each word with Laplace smoothing.
- Predicted classes by summing log prior and log likelihoods (Log-Sum Trick) and selecting the class with maximum score.
- Used CountVectorizer with unigrams and bigrams, ignoring rare words (min_df=2).

Part B – Sklearn MultinomialNB with Hyperparameter Tuning:

- Created a pipeline combining TfidfVectorizer and MultinomialNB.
- Tuned ngram_range (unigrams/bigrams) and smoothing parameter alpha using GridSearchCV on the dev set.
- Selected the best model based on macro F1 score.

Part C – Bayes Optimal Classifier Approximation:

- Sampled a subset of training data (~10,351 sentences).
- Trained five diverse models: MNB, Logistic Regression, Random Forest, Decision Tree, KNN.
- Calculated posterior weights based on validation log-likelihood.

- Combined models using soft voting with calculated weights.
- Evaluated the ensemble on the full test set.

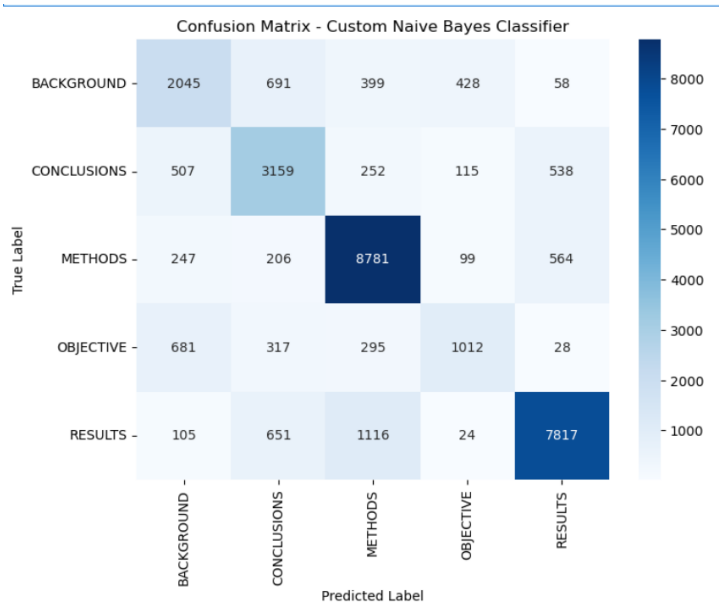
RESULTS AND ANALYSIS

PART A:

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

```
Fitting Count Vectorizer and transforming training data...
Vocabulary size: 301234
Transforming test data...

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

```
=== 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:

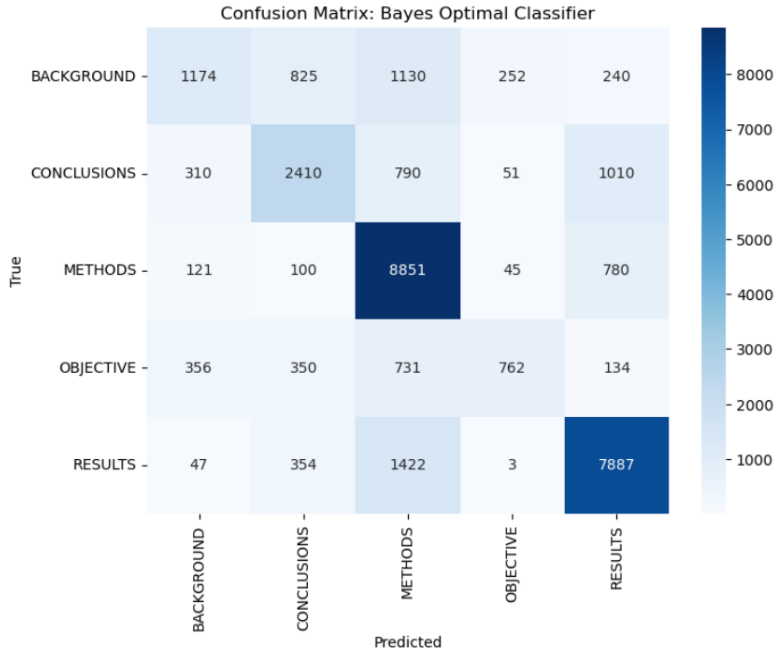
```
Training initial Naive Bayes pipeline...
Training complete.

=== 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 Model Parameters from Grid Search ===
Best Parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best Macro F1 Score: 0.5925
```

PART C:

```
Please enter your full SRN (e.g., PES1UG22CS345):  PES2UG23CS351
Using dynamic sample size: 10351
Actual sampled training set size used: 10351

Training all base models...
Training NaiveBayes...
Training LogisticRegression...
Training RandomForest...
Training DecisionTree...
Training KNN...
All base models trained.

Calculating posterior weights based on validation log-likelihood...
NaiveBayes: Log-likelihood = -0.7777
LogisticRegression: Log-likelihood = -0.7106
RandomForest: Log-likelihood = -0.8429
DecisionTree: Log-likelihood = -1.2063
KNN: Log-likelihood = -1.3074
Posterior Weights (normalized): [0.23547999 0.25183377 0.22062828 0.15340061 0.13865735]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6997
Macro F1 Score: 0.5986
```

Classification Report:

	precision	recall	f1-score	support
BACKGROUND	0.58	0.32	0.42	3621
CONCLUSIONS	0.60	0.53	0.56	4571
METHODS	0.68	0.89	0.78	9897
OBJECTIVE	0.68	0.33	0.44	2333
RESULTS	0.78	0.81	0.80	9713
accuracy			0.70	30135
macro avg	0.67	0.58	0.60	30135
weighted avg	0.69	0.70	0.68	30135



DISCUSSION:

Observations:

1.The custom Naive Bayes model performed the best overall, likely because count-based features captured class-specific word patterns effectively in biomedical abstracts.

2.The TF-IDF based Sklearn NB performed slightly worse, highlighting that raw counts can sometimes outperform normalized TF-IDF for Naive Bayes in domain-specific texts.

3.The BOC approximation did not significantly improve over the tuned NB. This is expected because combining diverse models is useful when individual

models have complementary strengths, but here the MNB already captures strong class-specific signals.

4.Confusion matrices show that METHODS and RESULTS were classified with high accuracy, while OBJECTIVE and BACKGROUND were more frequently misclassified.

Conclusion:

- Implementing MNB from scratch is an effective approach for biomedical text classification.
- Hyperparameter tuning can improve performance but is sensitive to the feature representation.
- Ensemble methods approximate the theoretical Bayes Optimal Classifier, but their performance depends on diversity and accuracy of base models.