Week 12: Naive Bayes Classifier

Name:	SRN:	Course:	Date:
Dinakar Emmanuel	PES2UG23CS178	UE23CS352A Machine Learning Lab	31/10/2025

Introduction:

The purpose of this lab is to build text classification systems that use Naive Bayes techniques to predict what type of section a given biomedical sentence belongs to.

The biomedical sentences are taken from a subset of the PubMed 200k RCT dataset, containing the sentence and the corresponding section/label out of “Background”, “Conclusions”, “Methods”, “Objective”, “Results”.

Text classification will be done in 3 ways:

1. Using an implementation of a Multinomial Naive Bayes Classifier, built from scratch and feature extraction done using count vectorizer.
2. Using scikit-learn’s MNB module with a TF-IDF vectorizer and tuning hyperparameters with grid search cross validation.
3. Approximate the Bayes Optimal Classifier (BOC) using an ensemble method built using diverse base models (hypotheses) and a Soft Voting Classifier using calculated posterior weights.

Methodology:

1) MNB Classifier from scratch:

- Use a count vectorizer instance to convert the sentences in the train set into a matrix of token counts, considering only unigrams and ignoring any token that appears less than 5 documents.
- These token counts are used as the features along with the given labels to train the classifier.

2) Sklearn MNB:

- Create a pipeline that uses Sklearn’s MNB module that classifies text based on the TF-IDF scores of tokens from the training set.
- Use default, standard parameters initially and then perform hyperparameter tuning for ngram range (unigram, bigram, both) and Laplace smoothing parameter (0.1, 0.5, 1.0, 2.0).
- Find the parameters that produce the best performing model.

3) Bayes Optimal Classifier:

- Sample the database.
- Use a TF-IDF vectorizer with some common settings.

- Split the train set into training and validation subsets.
- Train 5 models (MNB, Logistic Regression, Random Forest, Decision Tree, k-Nearest Neighbours) on the TF-IDF scores of tokens from the sampled training subset.
- Compute the log likelihood of the validation subset under each hypothesis, and use these results to get the posterior weights for each hypothesis.
- Use a soft voting classifier instance by fitting it to the full sampled train set. The voting classifier will average the predicted probabilities of the base model weighted by their posterior weight values.

Results and Analysis:

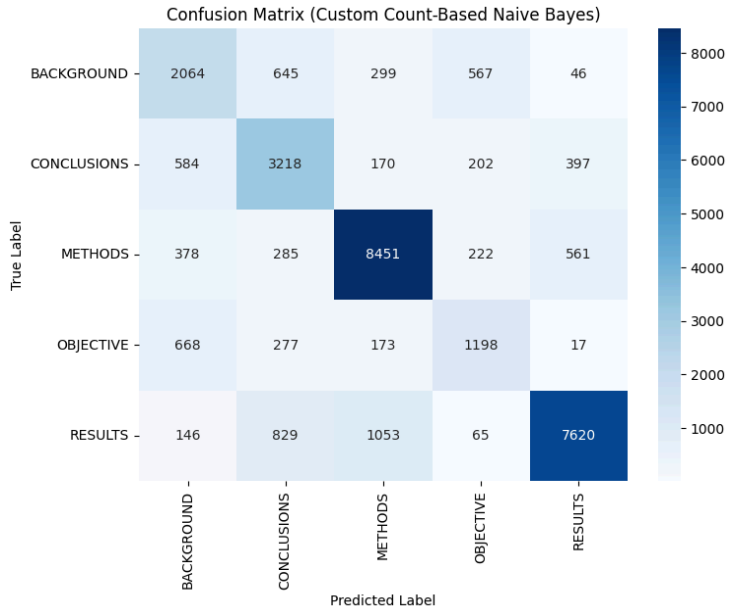
Part A:

```
=== 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
```



Part B:

```
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...
Grid search complete.

Best parameters found:
{'nb__alpha': 0.1, 'tfidf__ngram_range': (2, 2)}
Best cross-validation F1 score: 0.6581
```

Part C:

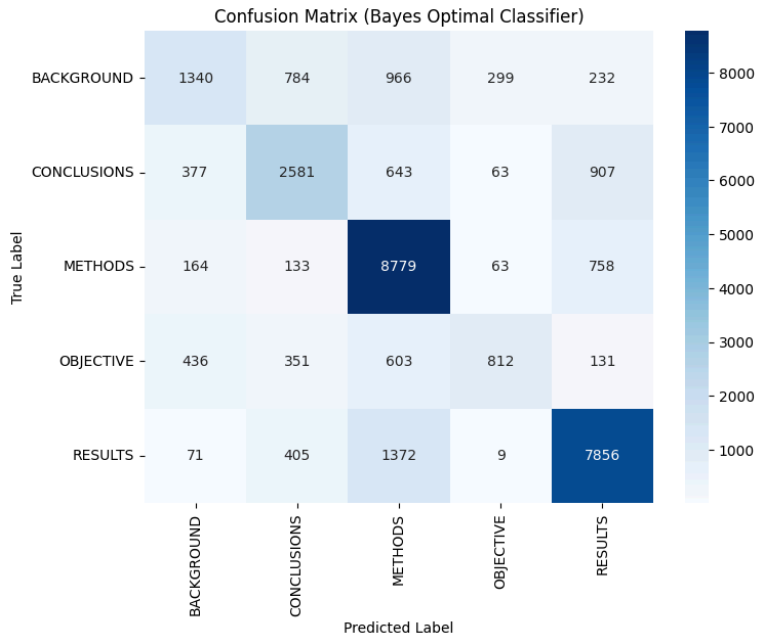
```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS178
Using dynamic sample size: 10178
Actual sampled training set size used: 10178
```

```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7091
              precision    recall  f1-score   support

  BACKGROUND       0.56      0.37      0.45      3621
 CONCLUSIONS       0.61      0.56      0.58      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

    accuracy                           0.71     30135
   macro avg       0.66      0.60      0.62     30135
weighted avg       0.70      0.71      0.69     30135

Macro-averaged F1 score: 0.6150
```



Discussion:

Part →	A	B (tuned/initial)		C
F1-score macro	0.6809	0.6581	0.5877	0.6150

The custom implementation of the MNB classifier appears to perform the best in terms of macro averaged F1 score, followed by the BOC and the initial sklearn MNB model. The tuned sklearn MNB has an F1 score 0.6581 but this model was evaluated on the dev set, while the other models used the test set.