Project Title: Naive Bayes Classifier

Name: Neema Shrivastava
SRN: PES2UG23CS377
Course: UE23CS352A
Date: 1/11/2025

Introduction

The purpose of this lab was to explore and compare probabilistic text classification approaches using Multinomial Naive Bayes (MNB) and the Bayes Optimal Classifier (BOC).
The key objective was to understand how Naive Bayes models assign class probabilities based on term frequencies, and how an ensemble of multiple models can be combined optimally using posterior probabilities to improve classification performance.

The tasks performed included:

- ➔ Implementing a count-based and TF-IDF-based Multinomial Naive Bayes classifier.
- ➔ Evaluating model accuracy and F1 scores on test data.
- ➔ Training multiple diverse classifiers (Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN).
- ➔ Approximating a Bayes Optimal Classifier using a soft-voting ensemble weighted by posterior probabilities.
- ➔ Analyzing and visualizing classification results using confusion matrices.

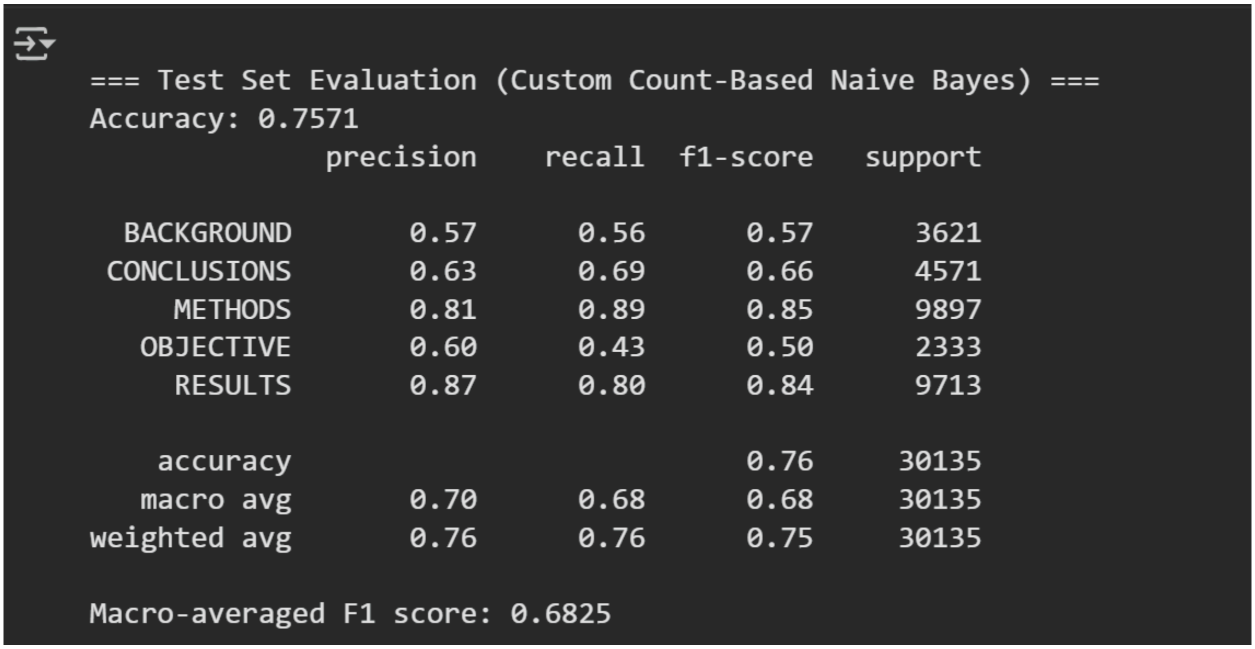
Methodology

Multinomial Naive Bayes (MNB): Implemented using a TfidfVectorizer for text feature extraction and MultinomialNB for classification. The model was trained on the training set and evaluated using metrics like accuracy and macro-F1 score. This method assumes feature independence and calculates the probability of each class based on term frequencies.

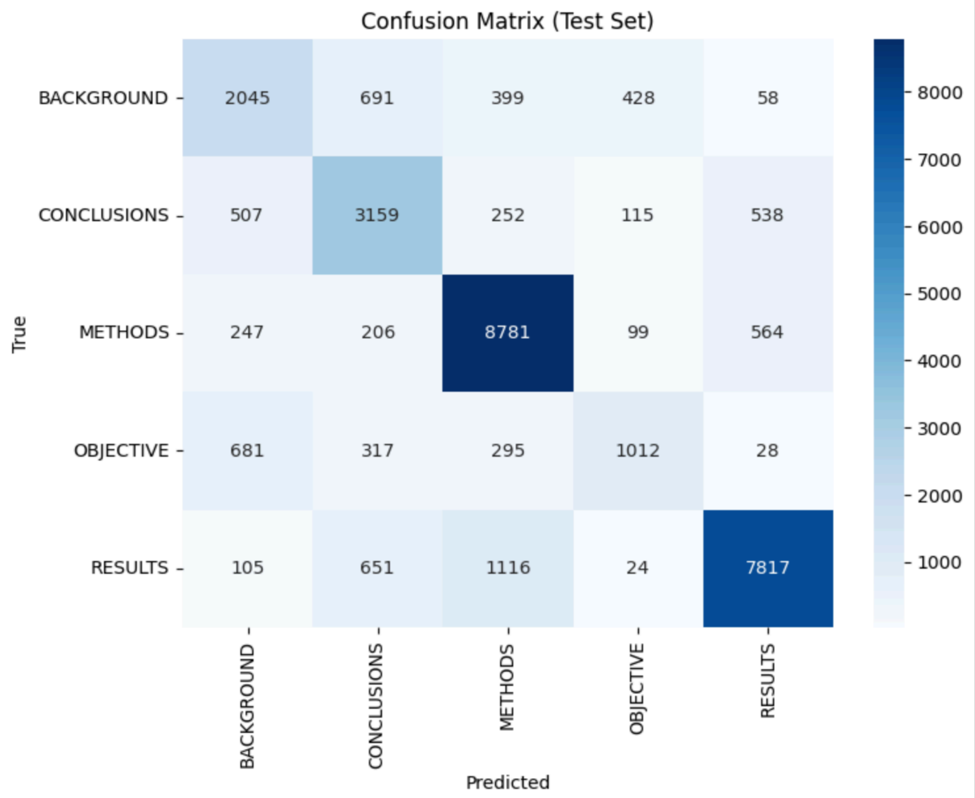
Bayes Optimal Classifier (BOC): Constructed as an ensemble of five diverse base learners: Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors. Each model was trained on the same sampled dataset to form hypotheses h_1 to h_5. A validation split was used to estimate posterior probabilities $P(h_i \mid D)$ for each model based on their validation performance. The final BOC was approximated using a soft voting classifier, where each base model's probability output was weighted by its posterior probability. The ensemble predictions were compared against ground truth labels to evaluate the overall improvement in classification performance.

Results and Analysis

Part A



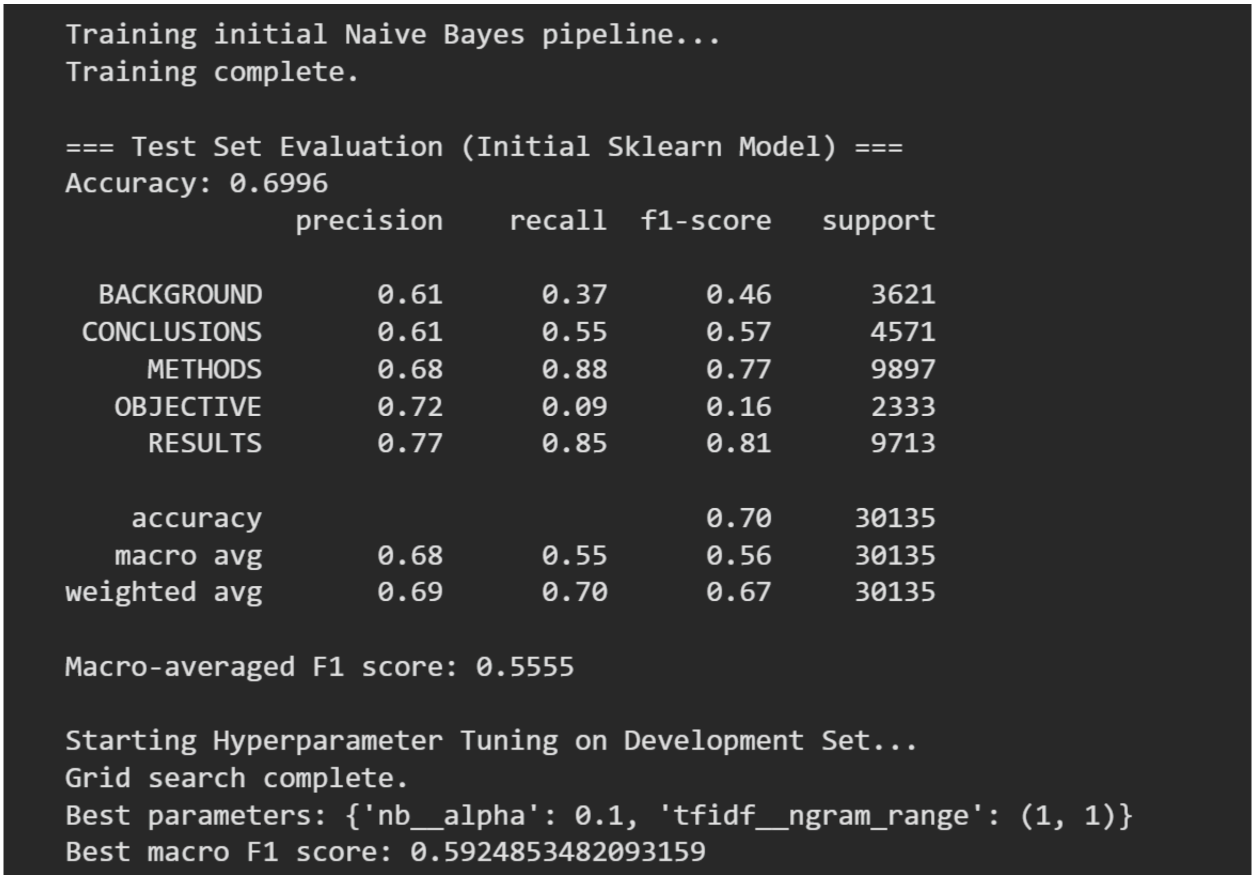
```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7571
              precision    recall  f1-score   support

  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
```



Confusion Matrix (Test Set)

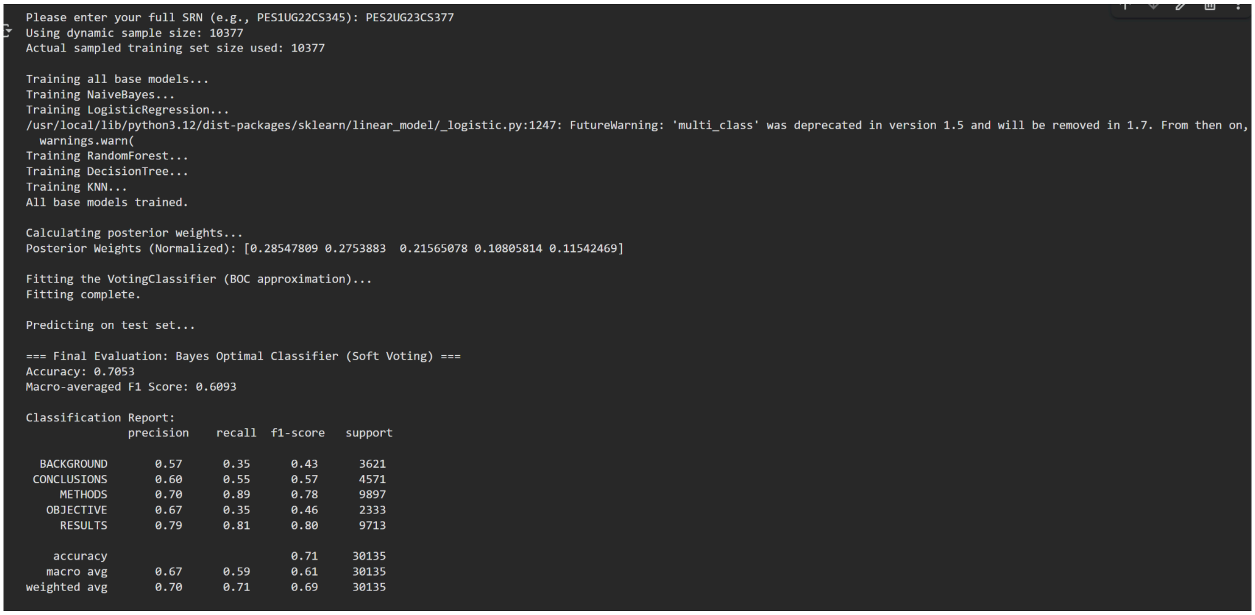
True \ Predicted	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	2045	691	399	428	58
CONCLUSIONS	507	3159	252	115	538
METHODS	247	206	8781	99	564
OBJECTIVE	681	317	295	1012	28
RESULTS	105	651	1116	24	7817

Part B

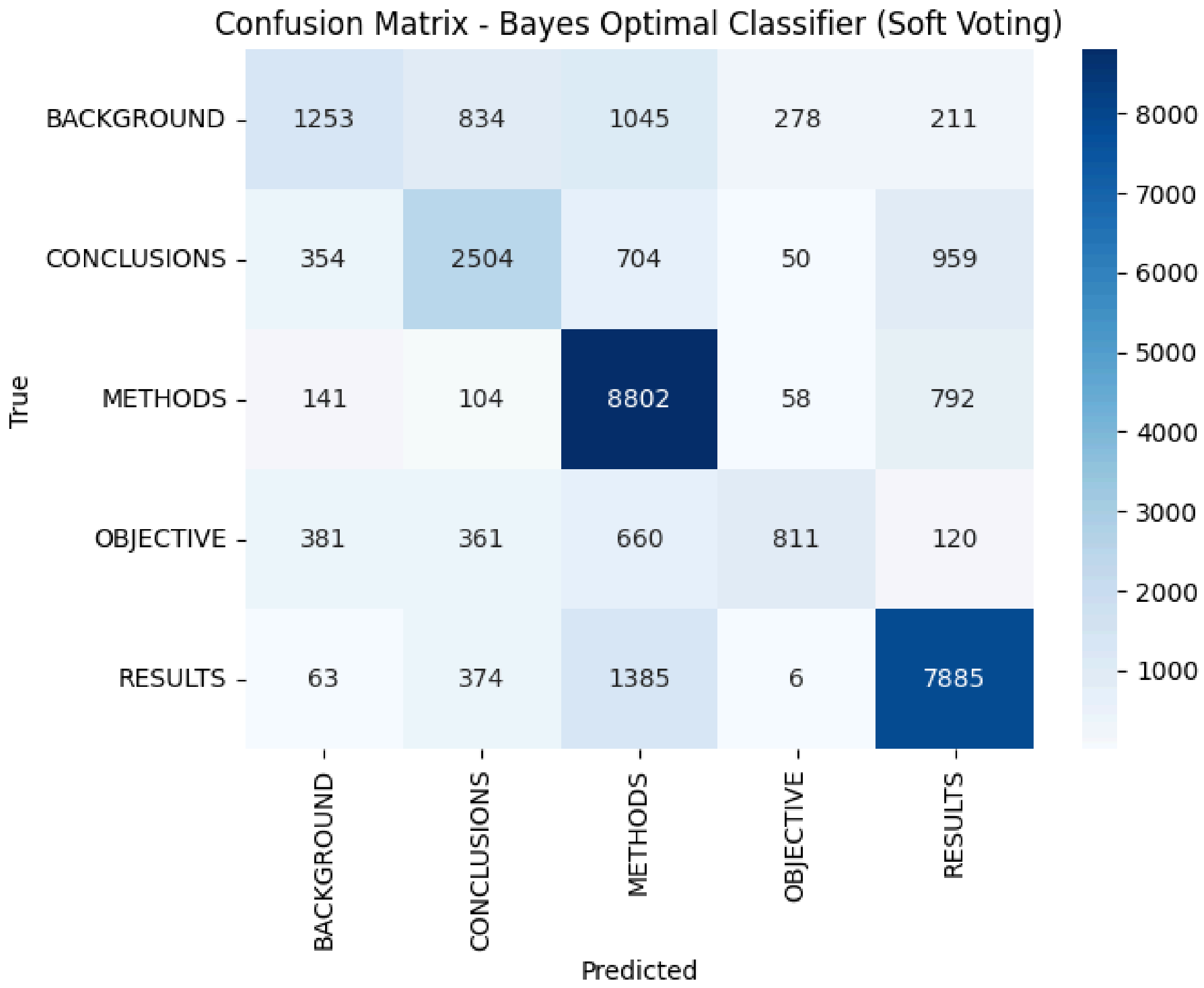


```
Training initial Naive Bayes pipeline...
Training complete.

=== 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...
Grid search complete.
Best parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best macro F1 score: 0.5924853482093159
```

Part C



```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS377
Using dynamic sample size: 10377
Actual sampled training set size used: 10377

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 in version 1.5 and will be removed in 1.7. From then on,
  warnings.warn(
Training RandomForest...
Training DecisionTree...
Training KNN...
All base models trained.

Calculating posterior weights...
Posterior Weights (Normalized): [0.28547809 0.2753883  0.21565078 0.10805814 0.11542469]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7053
Macro-averaged F1 Score: 0.6093

Classification Report:
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.35      0.43      3621
 CONCLUSIONS       0.60      0.55      0.57      4571
     METHODS       0.70      0.89      0.78      9897
   OBJECTIVE       0.67      0.35      0.46      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.71     30135
   macro avg       0.67      0.59      0.61     30135
weighted avg       0.70      0.71      0.69     30135
```



Confusion Matrix - Bayes Optimal Classifier (Soft Voting)

True \ Predicted	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	1253	834	1045	278	211
CONCLUSIONS	354	2504	704	50	959
METHODS	141	104	8802	58	792
OBJECTIVE	381	361	660	811	120
RESULTS	63	374	1385	6	7885

Discussion

- Scratch Count-Based Naive Bayes: Accuracy 0.7571, Macro F1 0.6825 — best overall performance with strong balance across classes, especially METHODS and RESULTS.
- Tuned Sklearn Model: Accuracy 0.6996, Macro F1 0.5555 — lower performance despite tuning; struggled with OBJECTIVE and BACKGROUND.
- BOC Approximation: Accuracy 0.7053, Macro F1 0.6093 — moderate improvement over Sklearn, but still below the scratch model; ensemble averaging smoothed performance but didn't surpass custom NB.