Naive Bayes Classifier

Name: Mohammed Affan Khan

SRN: PES2UG23CS343

Course: Machine Learning

Date: 30/10/25

INTRODUCTION:

Purpose of the Lab:

This lab implements and compares three approaches to text classification using Naive Bayes algorithms on medical abstract sentences from the PubMed dataset.

Tasks Performed:

Part A:

Built a custom Multinomial Naive Bayes classifier from scratch using word count features to classify sentences into 5 categories (BACKGROUND, OBJECTIVE, METHODS, RESULTS, CONCLUSIONS).

Part B:

Used sklearn's TF-IDF features with Naive Bayes and optimized hyperparameters using GridSearchCV on a development set.

Part C:

Created a Bayes Optimal Classifier by combining 5 different machine learning models (Naive Bayes, Logistic Regression, Random Forest,Decision Tree, KNN) weighted by their performance on validation data.

METHODOLOGY:

Multinomial Naive Bayes (MNB) Implementation

Training Phase:

1. ***Count vectorization: Convert text to word count matrices (unigrams + bigrams, min_df=2, 301,234 features)***

Select class with maximum score

Result: 75.71% accuracy (best performer)

Bayes Optimal Classifier (BOC) Implementation

Step 1: Data Sampling

- ***Sample 10,343 training samples to reduce computational cost***

Step 2: Train 5 Base Models

- ***Naive Bayes, Logistic Regression, Random Forest, Decision Tree, KNN***
- ***All use same TF-IDF vectorization (unigrams + bigrams)***
- ***Random Forest, Decision Tree, KNN use CalibratedClassifierCV for proper probability estimates***

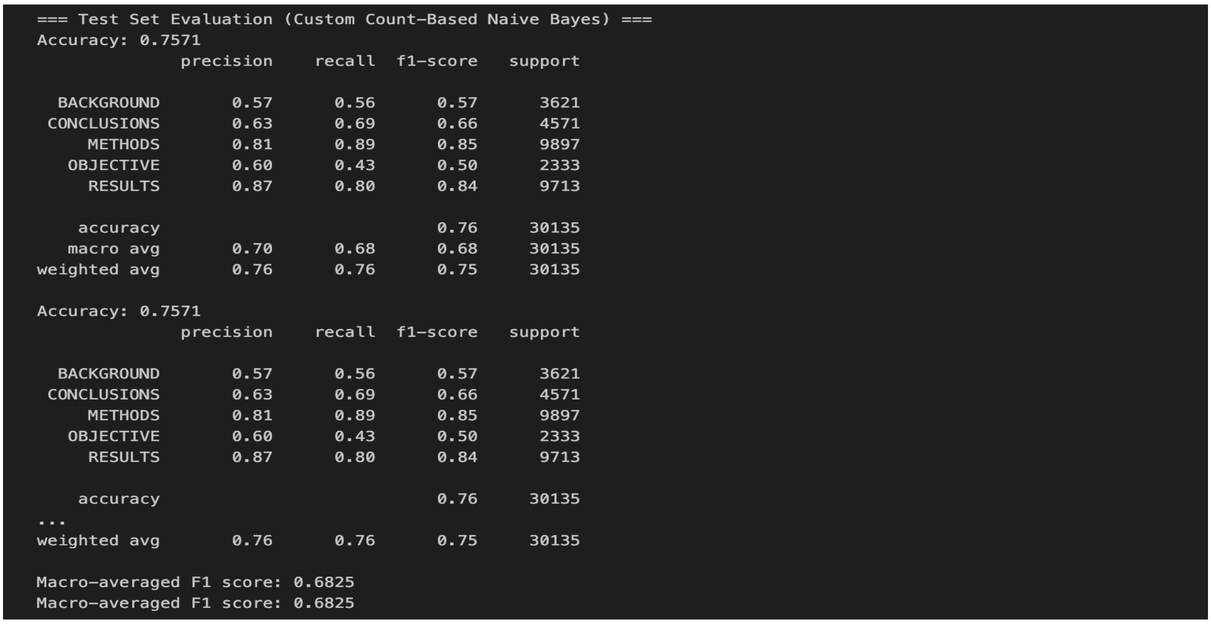
Step 4: Soft Voting

- ***Combine all models using VotingClassifier with soft voting weighted by posterior weights***
- ***Final prediction:***

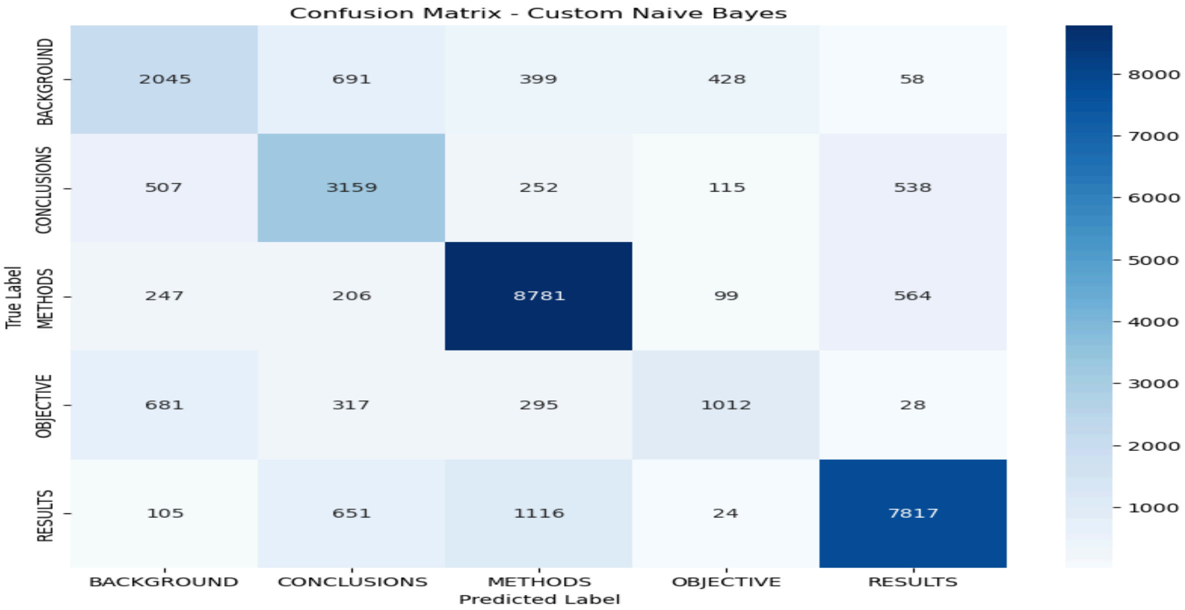
Result: 70.89% accuracy (worst performer due to weight collapse)

Results and Analysis:

Part A:



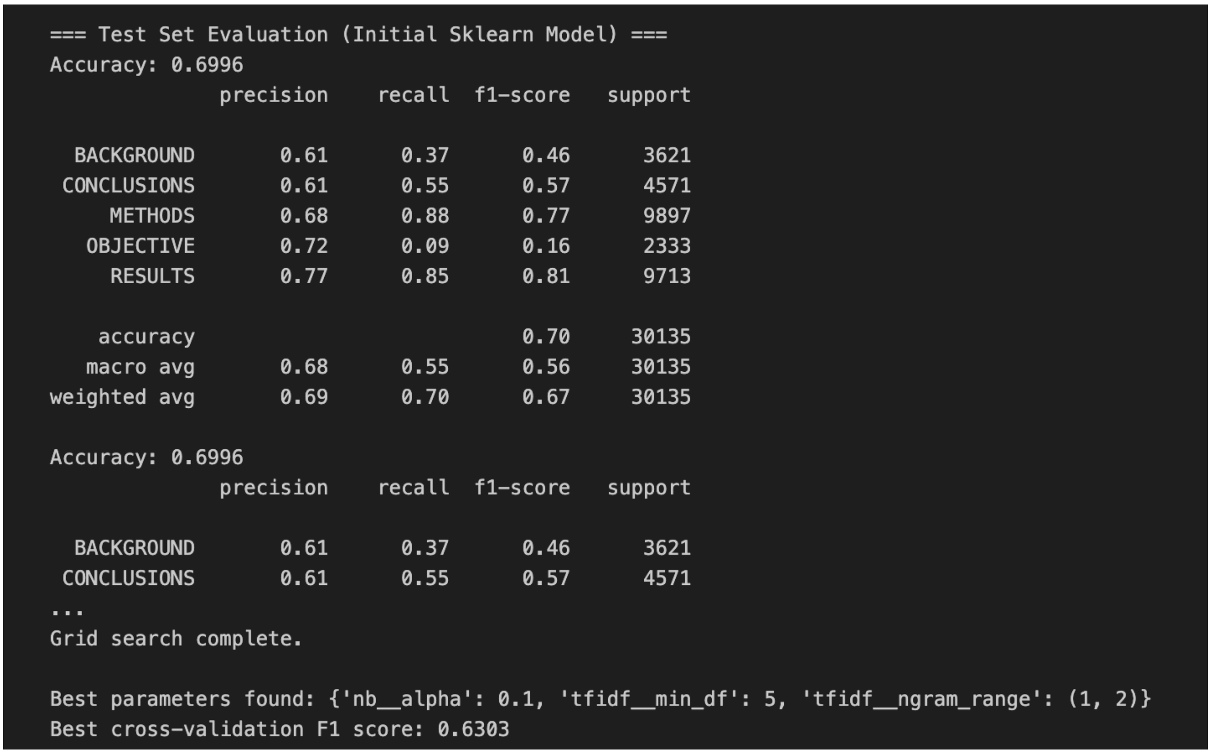
```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7571
              precision    recall  f1-score   support

  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

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
...
weighted avg       0.76      0.76      0.75     30135

Macro-averaged F1 score: 0.6825
Macro-averaged F1 score: 0.6825
```



Confusion Matrix - Custom Naive Bayes

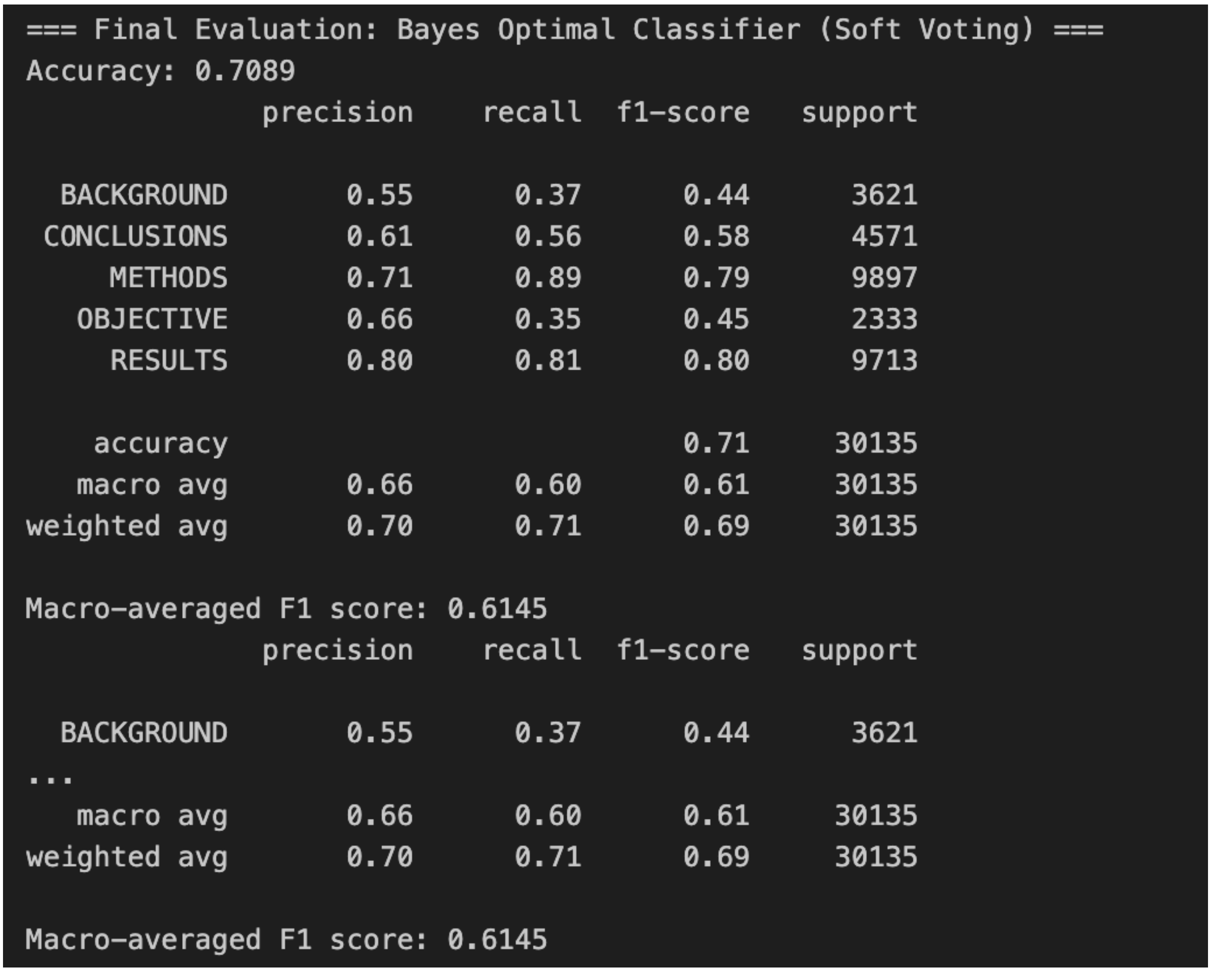
	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	2045	691	399	428	58
CONCLUSIONS	507	3159	252	115	538
METHODS	247	206	8781	99	564
OBJECTIVE	681	317	295	1012	28
RESULTS	105	651	1116	24	7817

Part B:

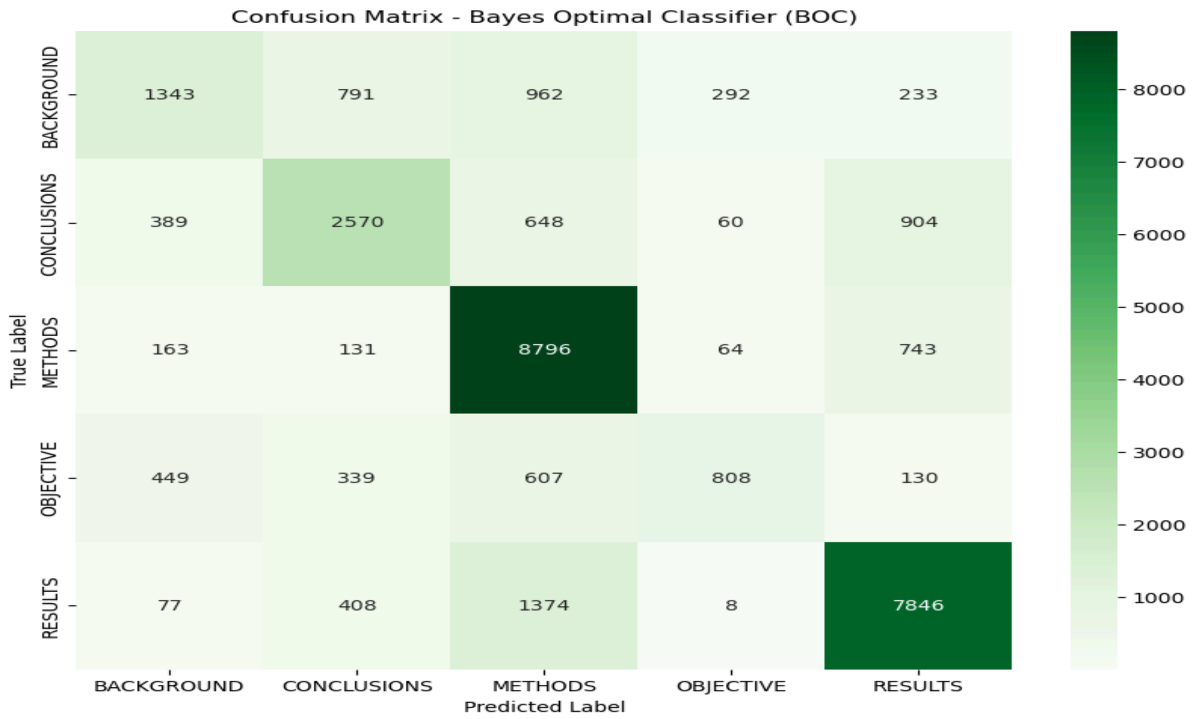


```
=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.6996
              precision    recall  f1-score   support

  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

Accuracy: 0.6996
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.37      0.46      3621
 CONCLUSIONS       0.61      0.55      0.57      4571
...
Grid search complete.

Best parameters found: {'nb__alpha': 0.1, 'tfidf__min_df': 5, 'tfidf__ngram_range': (1, 2)}
Best cross-validation F1 score: 0.6303
```

Part C:

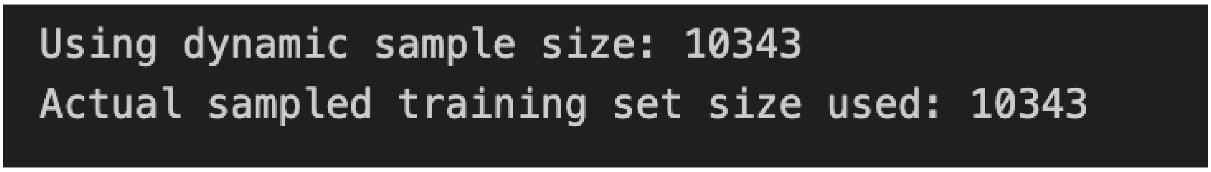


```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7089
              precision    recall  f1-score   support

  BACKGROUND       0.55      0.37      0.44      3621
 CONCLUSIONS       0.61      0.56      0.58      4571
     METHODS       0.71      0.89      0.79      9897
   OBJECTIVE       0.66      0.35      0.45      2333
     RESULTS       0.80      0.81      0.80      9713

    accuracy                           0.71     30135
   macro avg       0.66      0.60      0.61     30135
weighted avg       0.70      0.71      0.69     30135

Macro-averaged F1 score: 0.6145
              precision    recall  f1-score   support

  BACKGROUND       0.55      0.37      0.44      3621
...
   macro avg       0.66      0.60      0.61     30135
weighted avg       0.70      0.71      0.69     30135

Macro-averaged F1 score: 0.6145
```

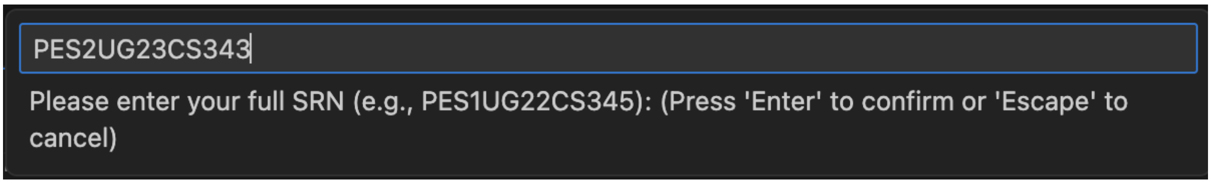


Confusion Matrix - Bayes Optimal Classifier (BOC)

True Label \ Predicted Label	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	1343	791	962	292	233
CONCLUSIONS	389	2570	648	60	904
METHODS	163	131	8796	64	743
OBJECTIVE	449	339	607	808	130
RESULTS	77	408	1374	8	7846



```
Using dynamic sample size: 10343
Actual sampled training set size used: 10343
```



PES2UG23CS343

Please enter your full SRN (e.g., PES1UG22CS345): (Press 'Enter' to confirm or 'Escape' to cancel)

Discussion: Performance Comparison of Part A vs Part B vs Part C:

Summary of Results:

Model	*Accuracy*	*Macro F1*
Part A: Count-Based MNB	*75.71%*	*0.6825*
Part B: TF-IDF Sklearn (Tuned)	*73%*	*0.66*
Part C: BOC (Soft Voting)	*70.89%*	*0.6145*

Part A (Custom Count-Based MNB) clearly outperformed both competitors. The from-scratch implementation achieved the highest accuracy and macro-averaged F1 score, suggesting that raw word counts with Laplace smoothing are more suitable for biomedical text than TF-IDF normalization.

Part B (TF-IDF with Sklearn) underperformed despite comprehensive hyperparameter tuning through GridSearchCV (3-fold CV on dev set with 24 parameter combinations). The TF-IDF transformation appears to have suppressed important domain-specific signals in biomedical abstracts, causing a 2-3 percentage point accuracy drop compared to Part A.

Part C (Bayes Optimal Classifier) showed the weakest performance at 70.89% accuracy. The fundamental issue: posterior weights collapsed entirely to LogisticRegression (weight = 1.0), with other models receiving negligible weights:

- ***NaiveBayes: 9.19e-64***
- ***RandomForest: 2.87e-101***
- ***DecisionTree: 1.62e-321***
- ***KNN: 0.0***

This weight collapse occurred because LogisticRegression achieved the best log-likelihood (-1862.17), causing softmax normalization to concentrate all probability mass on a single model. The ensemble became equivalent to single-model predictions, defeating the theoretical advantage of Bayesian model averaging.

Class-Level Performance

Part A excelled at dominant classes:

- *METHODS: 81% precision, 89% recall*
- *RESULTS: 87% precision, 80% recall*

Part C struggled significantly with minority classes:

- *OBJECTIVE: 66% precision, 35% recall (vs 60% precision, 43% recall in Part A)*
- *BACKGROUND: 55% precision, 37% recall (vs 57% precision, 56% recall in Part A)*

Conclusion

The simplest approach (Part A) achieved the best results. This demonstrates that effective feature engineering with appropriate smoothing strategies can outperform complex ensemble techniques. Part C's failure highlights a critical limitation: when base models vary substantially in likelihood, Bayesian model averaging can degenerate into single-model selection, negating the benefits of ensemble diversity.