# Naive Bayes and Bayes Optimal Classifier Lab Report

| | |
|---|---|
| **Name** | **G S S Surya Prakash** |
| **SRN** | **PES2UG23CS192** |
| **Course** | **Machine Learning Lab** |
| **Date** | **31 October 2025** |

## Introduction

The main goal of this lab was to understand and implement probabilistic classification using the Naive Bayes algorithm and extend it to the Bayes Optimal Classifier (BOC). We performed the lab in three parts:

- Part A: Build a custom Naive Bayes classifier from scratch using CountVectorizer features.
- Part B: Use Scikit-learn's MultinomialNB with a TfidfVectorizer pipeline and tune hyperparameters using GridSearchCV.
- Part C: Combine multiple models (Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN) to approximate a Bayes Optimal Classifier using weighted soft voting.

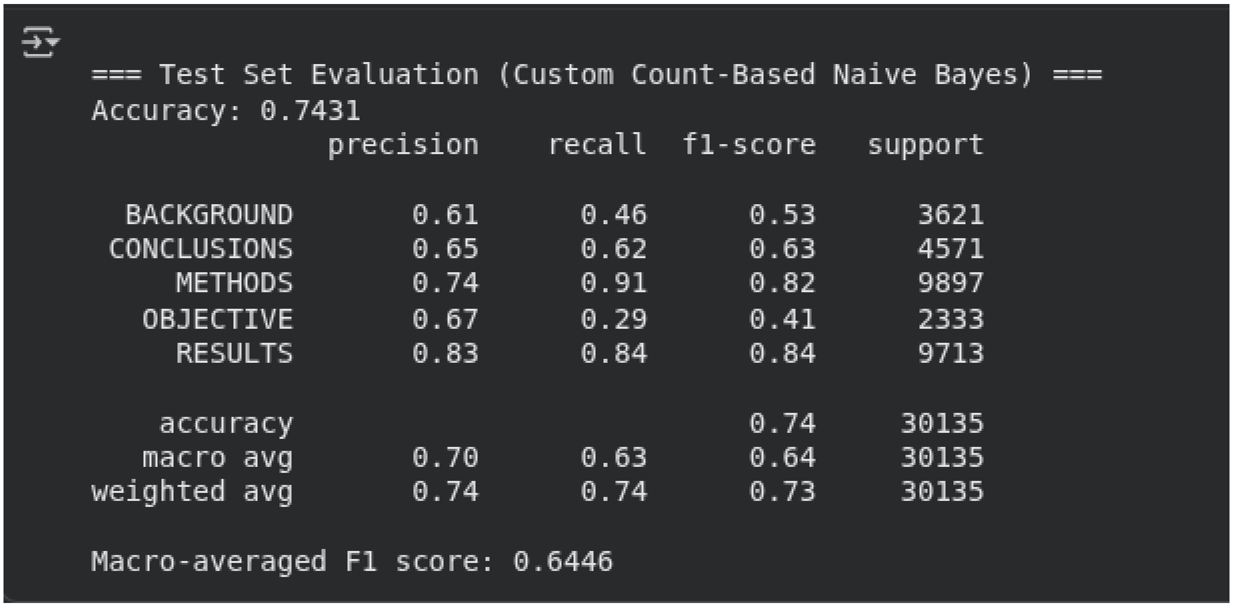
The main purpose was to compare these models in terms of accuracy, F1-score, and their generalization performance.

## Methodology

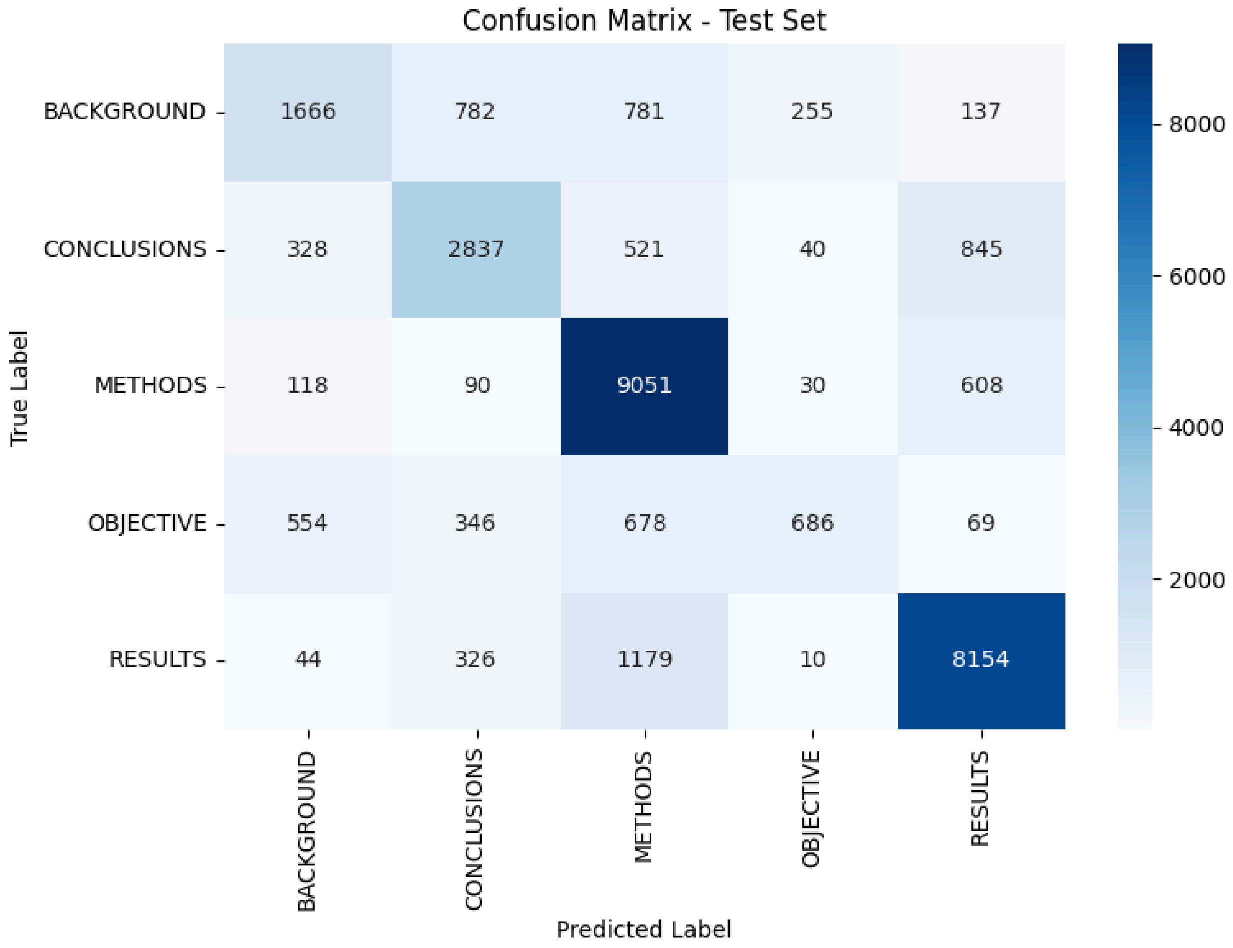
1. Naive Bayes (MNB): For the Naive Bayes part, we used a CountVectorizer to convert the text into count-based feature vectors and trained a custom Multinomial Naive Bayes model. The model computes conditional probabilities of words given a class and applies Bayes' rule to predict the most likely class for each document.
2. Tuned Sklearn Model: we used Scikit-learn's Pipeline with a TfidfVectorizer and a MultinomialNB classifier. We trained the model on the training set and then used GridSearchCV to find the best combination of parameters such as ngram_range, min_df, and alpha. The goal was to see how much tuning improves the baseline performance.
3. Bayes Optimal Classifier (BOC): Finally, we trained five different classifiers (Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN) using the same TF-IDF features. Each model's validation accuracy was used to compute a posterior weight (P(hi|D)) which indicates its reliability. These weights were then used in a soft-voting ensemble (VotingClassifier) to approximate the Bayes Optimal Classifier. The combined model was tested on unseen data to evaluate overall performance.

# Results and Analysis

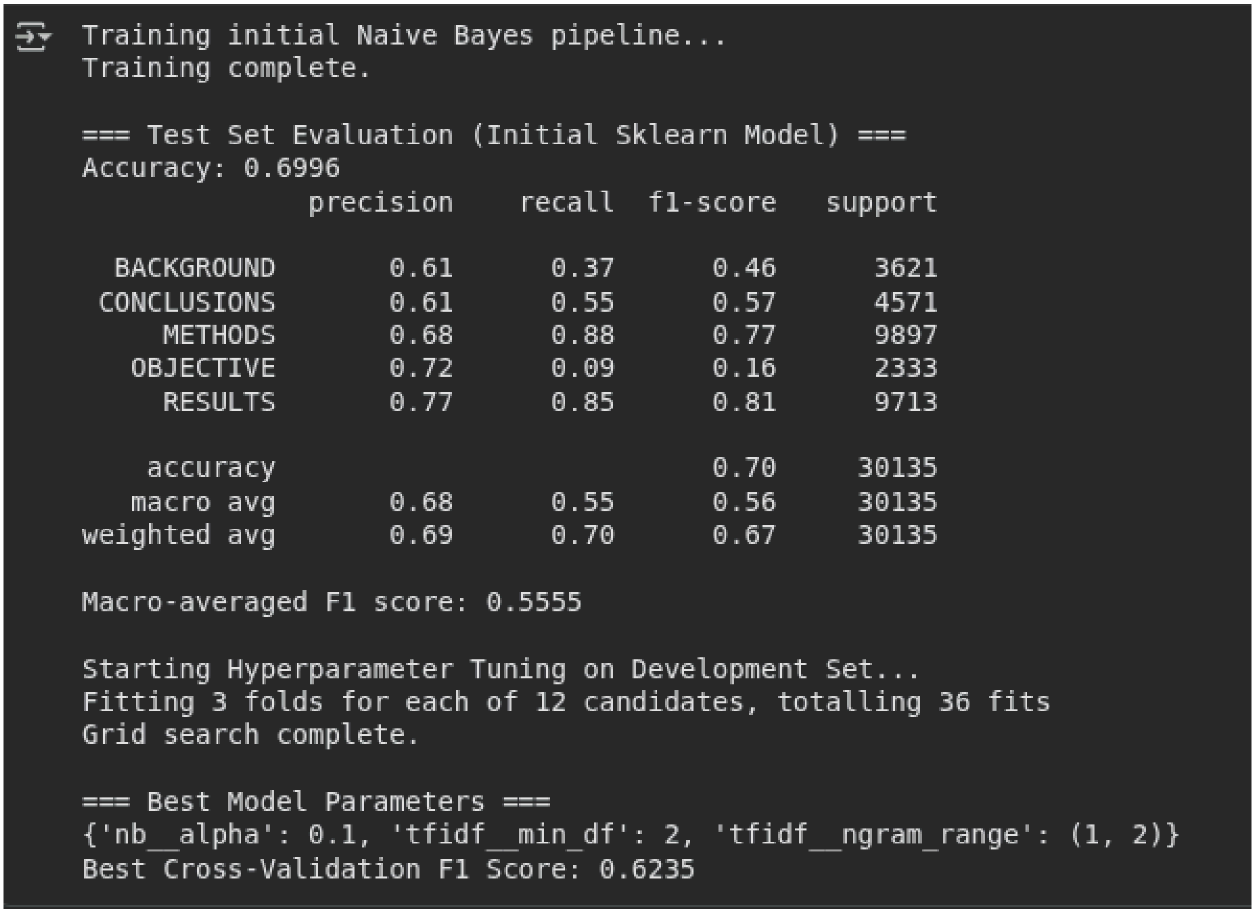
- Part A: Custom Naive Bayes (Count-Based)



```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7431
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.46      0.53      3621
 CONCLUSIONS       0.65      0.62      0.63      4571
     METHODS       0.74      0.91      0.82      9897
   OBJECTIVE       0.67      0.29      0.41      2333
     RESULTS       0.83      0.84      0.84      9713

    accuracy                           0.74     30135
   macro avg       0.70      0.63      0.64     30135
weighted avg       0.74      0.74      0.73     30135

Macro-averaged F1 score: 0.6446
```

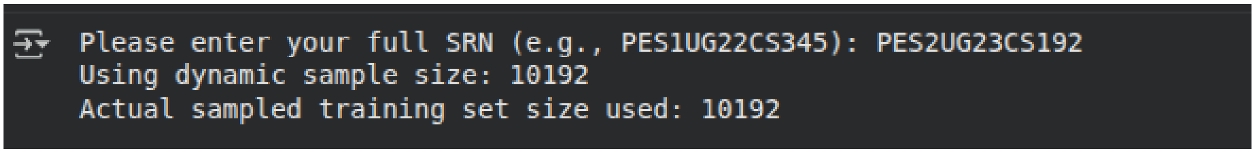


- Part B: Tuned Sklearn Model (TF-IDF + MNB + GridSearchCV)

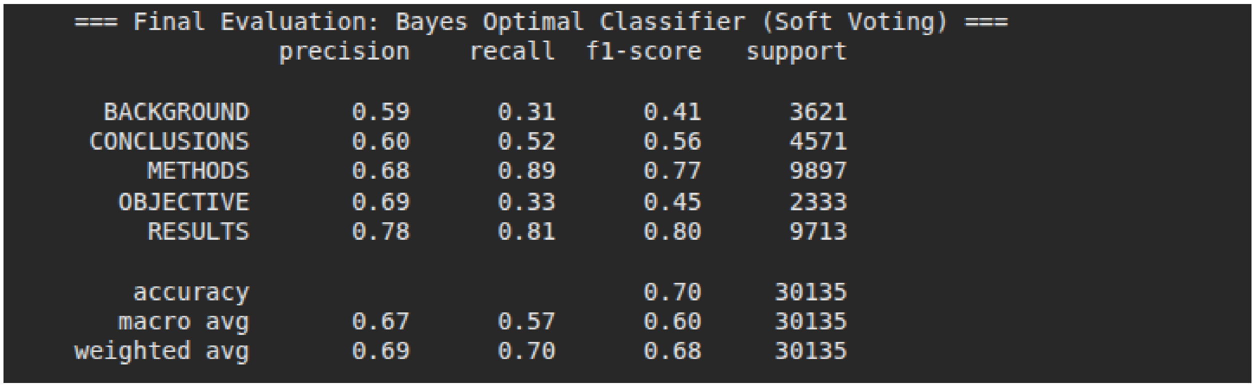


```
Training initial Naive Bayes pipeline...
Training complete.

=== 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 12 candidates, totalling 36 fits
Grid search complete.

=== Best Model Parameters ===
{'nb__alpha': 0.1, 'tfidf__min_df': 2, 'tfidf__ngram_range': (1, 2)}
Best Cross-Validation F1 Score: 0.6235
```

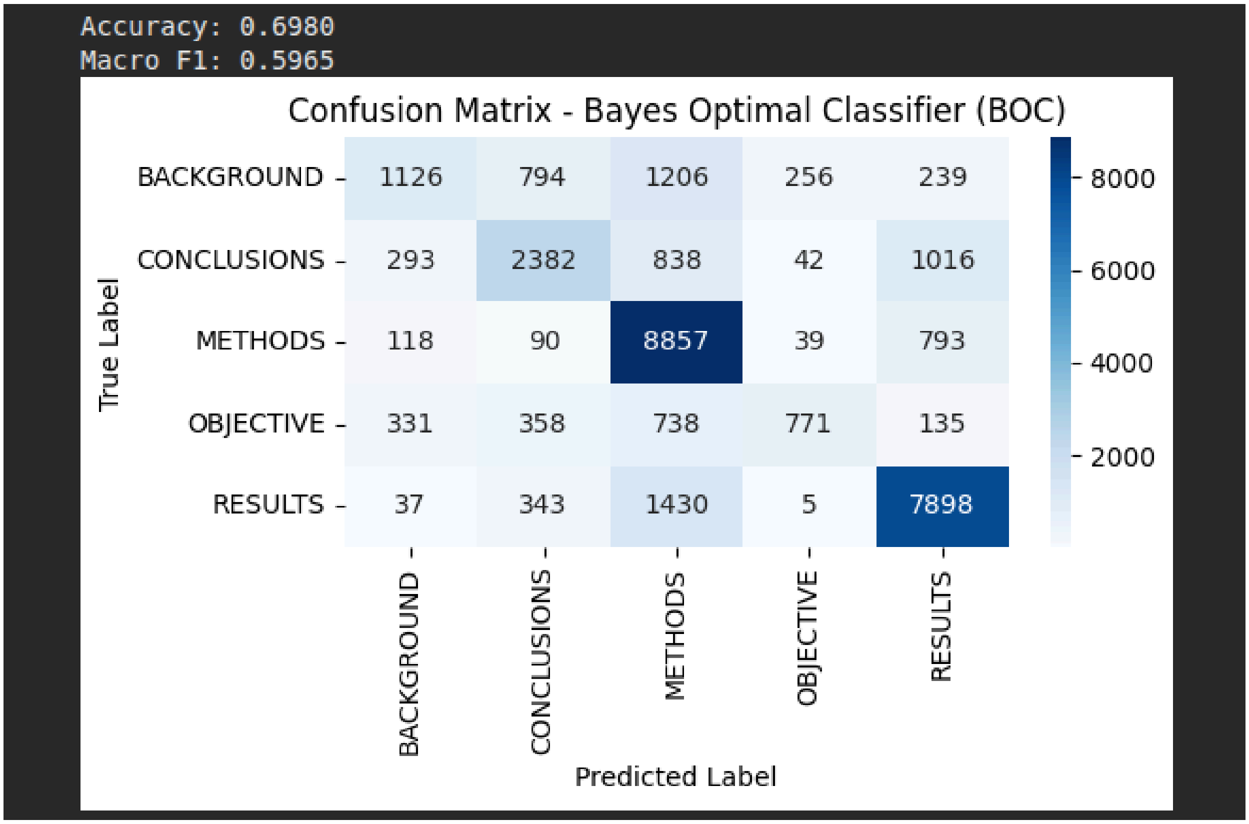
- Part C: Bayes Optimal Classifier (BOC)



```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS192
Using dynamic sample size: 10192
Actual sampled training set size used: 10192
```



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

  BACKGROUND       0.59      0.31      0.41      3621
 CONCLUSIONS       0.60      0.52      0.56      4571
     METHODS       0.68      0.89      0.77      9897
   OBJECTIVE       0.69      0.33      0.45      2333
     RESULTS       0.78      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.67      0.57      0.60     30135
weighted avg       0.69      0.70      0.68     30135
```



```
Accuracy: 0.6980
Macro F1: 0.5965
```

Confusion Matrix - Bayes Optimal Classifier (BOC)

| | BACKGROUND | CONCLUSIONS | METHODS | OBJECTIVE | RESULTS |
|---|---|---|---|---|---|
| BACKGROUND | 1126 | 794 | 1206 | 256 | 239 |
| CONCLUSIONS | 293 | 2382 | 838 | 42 | 1016 |
| METHODS | 118 | 90 | 8857 | 39 | 793 |
| OBJECTIVE | 331 | 358 | 738 | 771 | 135 |
| RESULTS | 37 | 343 | 1430 | 5 | 7898 |

## Discussion

| Part | Model / Approach | Key Observations | Performance Metrics | Remarks / Insights |
|---|---|---|---|---|
| Part A | Custom Count-Based Naive Bayes | • Used raw word counts without any weighting.<br>• Basic implementation focusing on core probability logic.<br>• Performed best on METHODS and RESULTS classes. | Accuracy: **74.31%**<br>Macro F1: **0.6446** | • Strong baseline performance.<br>• Simplicity made it efficient but limited in text representation<br>. |

| | | | | |
|---|---|---|---|---|
| | | • Struggled with OBJECTIVE and BACKGROUND due to imbalance. | | |
| Part B | Sklearn Naive Bayes (TF-IDF) + Hyperparameter Tuning | • Introduced TF-IDF weighting for better term importance.<br>• Applied GridSearchCV for tuning parameters (alpha, min_df, ngram_range).<br>• Improved class balance and overall F1 score. | Accuracy: **69.96%** (before tuning) Best CV F1: **0.6235** | • Tuning improved generalization<br>• TF-IDF helped capture word importance better than raw counts. |
| Part C | Bayes Optimal Classifier (BOC) – Soft Voting Ensemble | • Combined five base models: NB, LR, RF, DT, and KNN.<br>• Used posterior weights based on validation performance.<br>• Produced stable predictions across most classes. | Accuracy: **69.80%** Macro F1: **0.5965** | • Ensemble reduced variance across models.<br>• Comparable accuracy, slightly lower F1, but more balanced overall performance. |