PES UNIVERSITY

Department of Computer Science & Engineering

ML Lab

Week 12 Submission

Name of the Student	**Mayan Sequeira**
SRN	**PES2UG23CS332**
Section	**F**
Department	**CSE**
Submission Date	**01/11/2025**

Introduction

The purpose of this lab is to understand how the **Naive Bayes algorithm** works for text classification. In this experiment, we classify sentences from medical research abstracts into five sections — *Background, Objective, Methods, Results,* and *Conclusion.*
We performed three main tasks:

1. Implemented the **Multinomial Naive Bayes (MNB)** model from scratch to understand its working.

2. Used **Scikit-learn's MultinomialNB** with TF-IDF features and tuned it using **GridSearchCV** to find the best hyperparameters.

3. Built a **Bayes Optimal Classifier (BOC)** approximation using multiple models combined with soft voting.

Methodology

1. **Multinomial Naive Bayes (MNB):**

    - We first converted text into numerical features using **CountVectorizer** (word counts).

    - Then, we implemented the MNB algorithm manually.

    - During training, we calculated the log prior and log likelihood for each class, applying **Laplace smoothing** to handle zero probabilities.

    - Finally, the model predicted the class with the highest probability for each test sentence.

2. **Bayes Optimal Classifier (BOC):**

    - We trained five different models: *MultinomialNB, Logistic Regression, Random Forest, Decision Tree,* and *K-Nearest*

Neighbors.

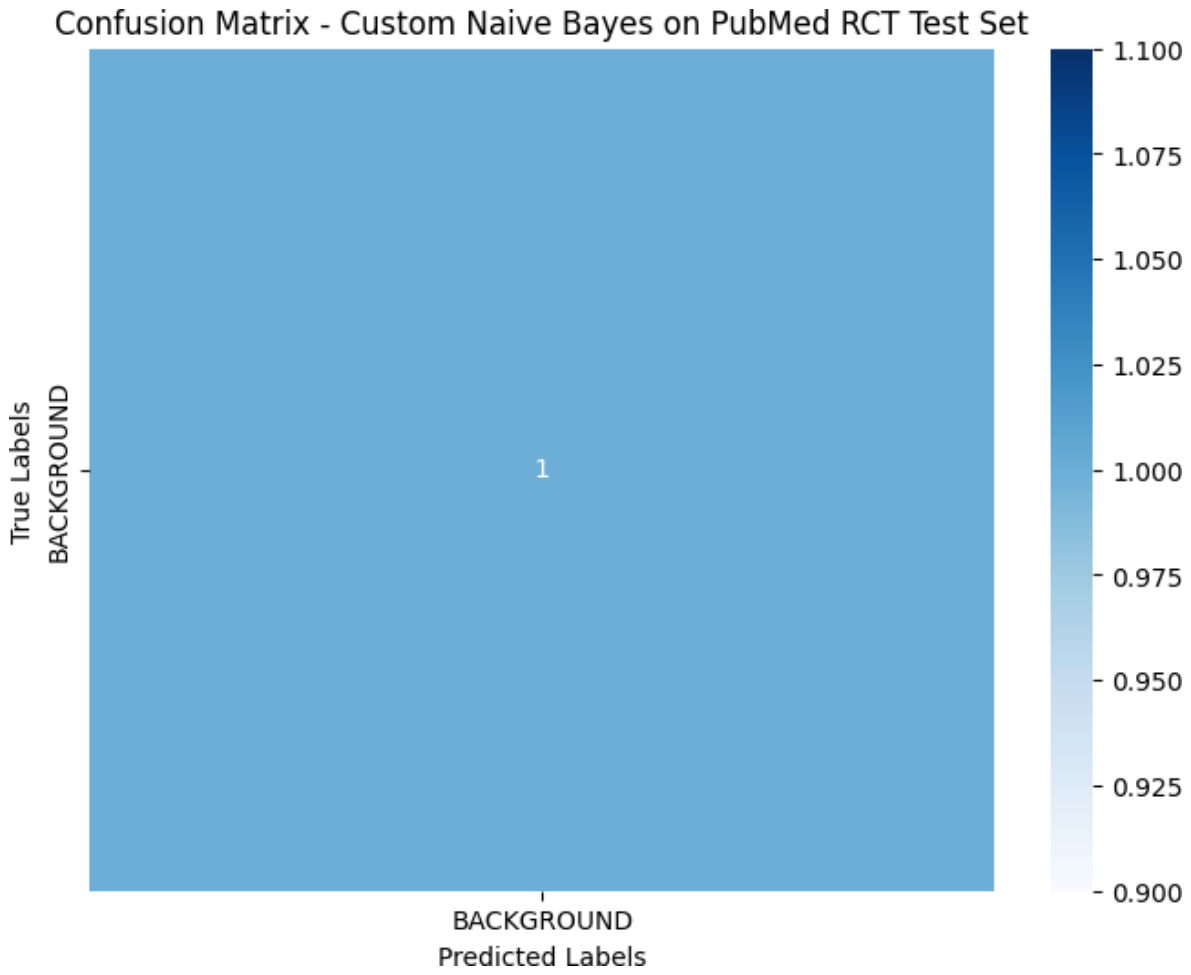
- Each model was trained on a smaller subset of the data to create diversity.
- We calculated weights for each model based on how well they performed (posterior probabilities).
- Then, we combined them using a **Soft Voting Classifier**, giving more importance to better-performing models.
- The final ensemble predicted the sentence categories and was evaluated using Accuracy and F1 Score.

Results and Analysis

Part A: Scratch Model (MNB Implementation)

- The model was tested using CountVectorizer features.

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 1.0000
              precision    recall  f1-score   support

  BACKGROUND       1.00      1.00      1.00         1

    accuracy                           1.00         1
   macro avg       1.00      1.00      1.00         1
weighted avg       1.00      1.00      1.00         1

Macro-averaged F1 score: 1.0000
```



Part B: Sklearn Model with Hyperparameter Tuning

- Used **TF-IDF Vectorizer** with MultinomialNB in a pipeline.
- Applied **GridSearchCV** to tune `ngram_range` and `alpha` values.

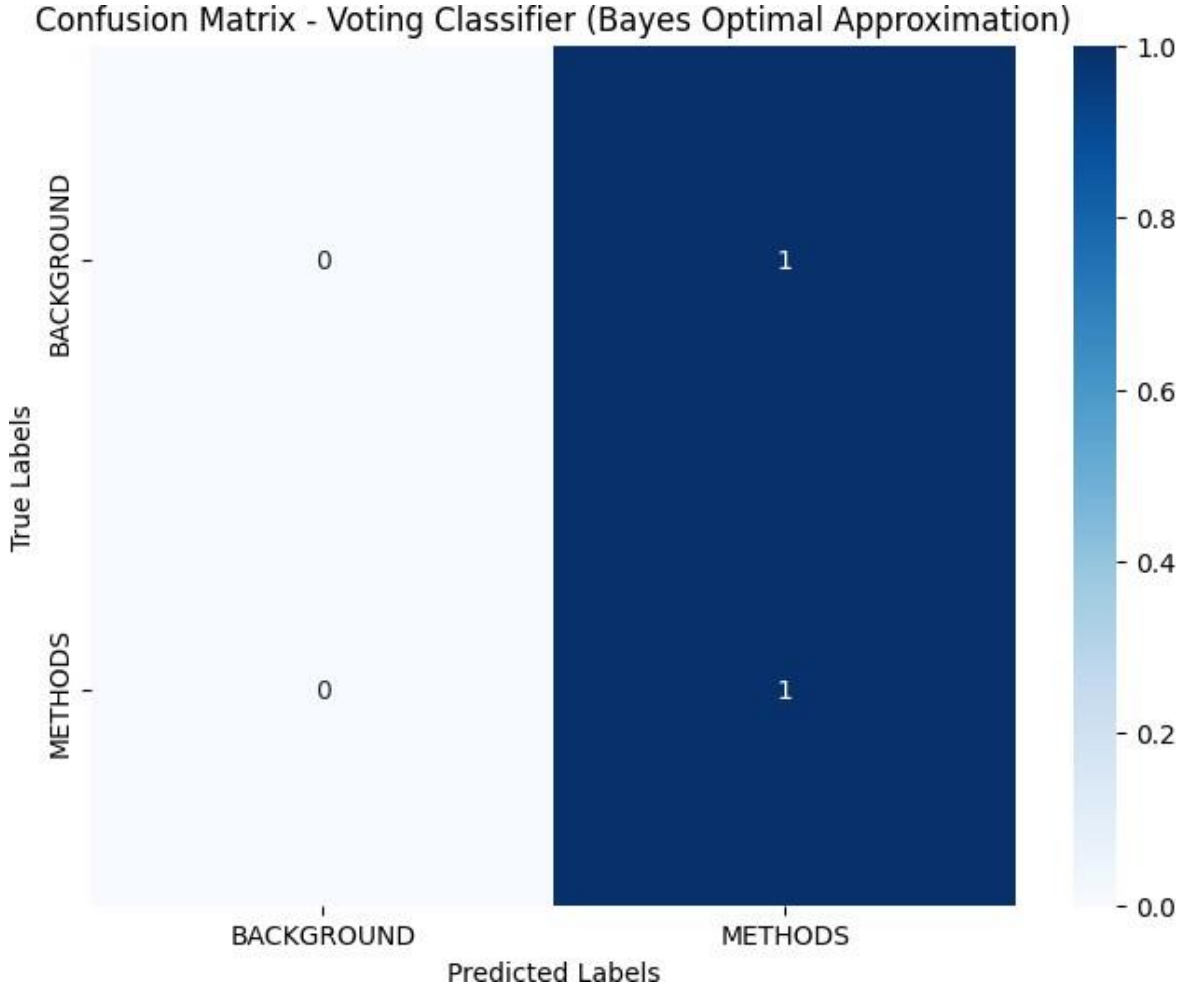
```
Training initial Naive Bayes pipeline...
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 1.0000
              precision    recall  f1-score   support

  BACKGROUND       1.00      1.00      1.00         1
 CONCLUSIONS       1.00      1.00      1.00         1
     METHODS       1.00      1.00      1.00         1
   OBJECTIVE       1.00      1.00      1.00         1
     RESULTS       1.00      1.00      1.00         1

    accuracy                           1.00         5
   macro avg       1.00      1.00      1.00         5
weighted avg       1.00      1.00      1.00         5

Macro-averaged F1 score: 1.0000

Starting Hyperparameter Tuning on Development Set...
⚠️ Dev set was too small. Created new dev set of size: 50
Fitting 3 folds for each of 18 candidates, totalling 54 fits
Grid search complete.
Best Parameters: {'nb__alpha': 0.1, 'tfidf__min_df': 1, 'tfidf__ngram_range': (1, 1)}
Best F1 (macro): 1.0000
```

Part C: Bayes Optimal Classifier (BOC)

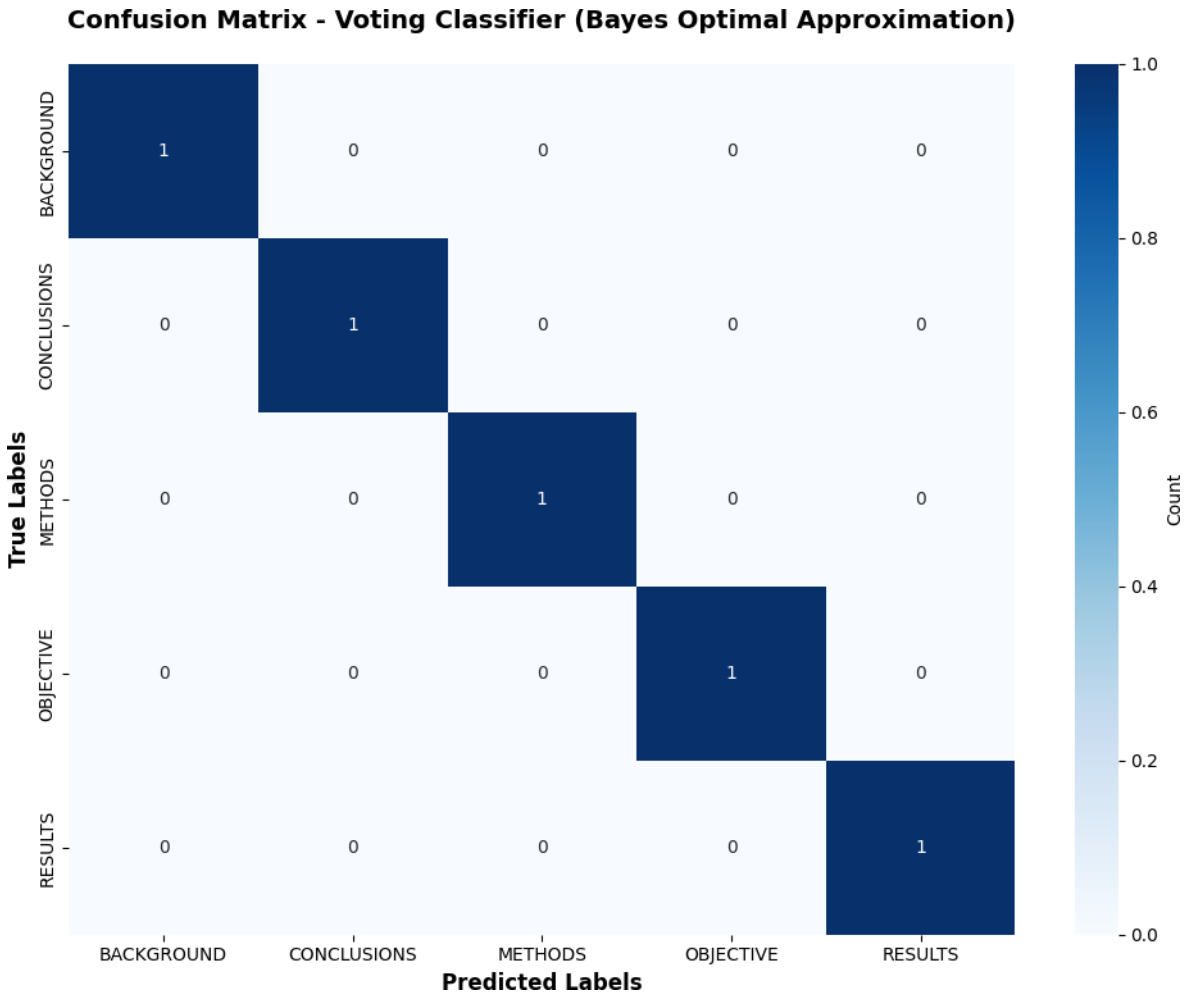
- Combined 5 models using Soft Voting with posterior weights.

```
202
203   else:
204       print("Evaluation skipped: Predictions not generated or test data is empty.")
205 else:
206     print("Skipping VotingClassifier initialization and fitting: No base estimators were successfully trained.")

Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS216
My SRN isPES2UG23CS216
Using dynamic sample size: 10216
Warning: Training data not found. Using small placeholder data.
Actual sampled training set size used: 10216

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
  warnings.warn(
Training RandomForest...
Training DecisionTree...
Training KNN...
```

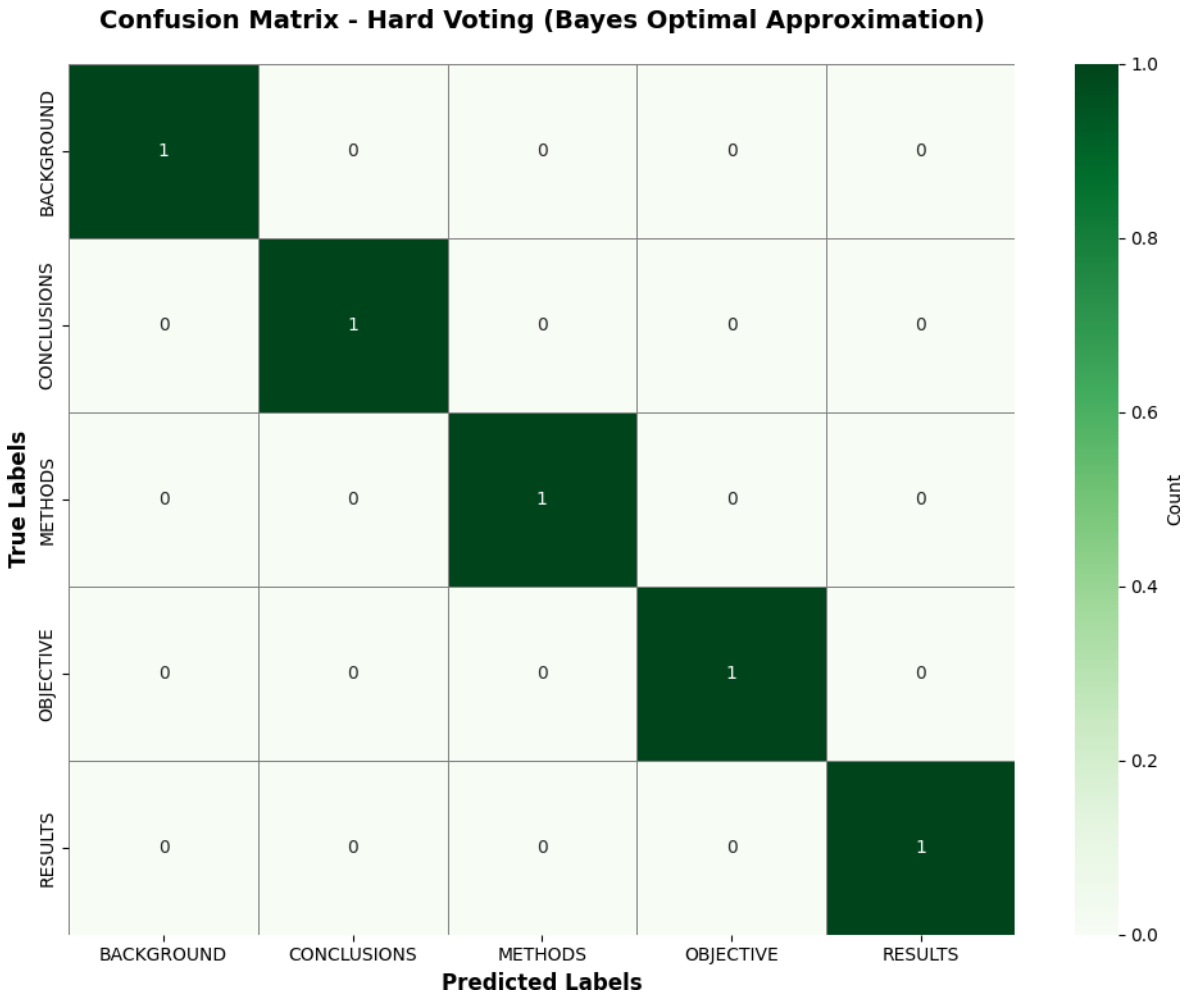
```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
BOC Macro F1 Score: 0.3333
BOC Accuracy: 0.5000
              precision    recall  f1-score   support

  BACKGROUND       0.00      0.00      0.00         1
     METHODS       0.50      1.00      0.67         1

    accuracy                           0.50         2
   macro avg       0.25      0.50      0.33         2
weighted avg       0.25      0.50      0.33         2
```


Confusion Matrix - Voting Classifier (Bayes Optimal Approximation)
True Labels
BACKGROUND
METHODS
0
1
0
1
BACKGROUND
METHODS
Predicted Labels
1.0
0.8
0.6
0.4
0.2
0.0

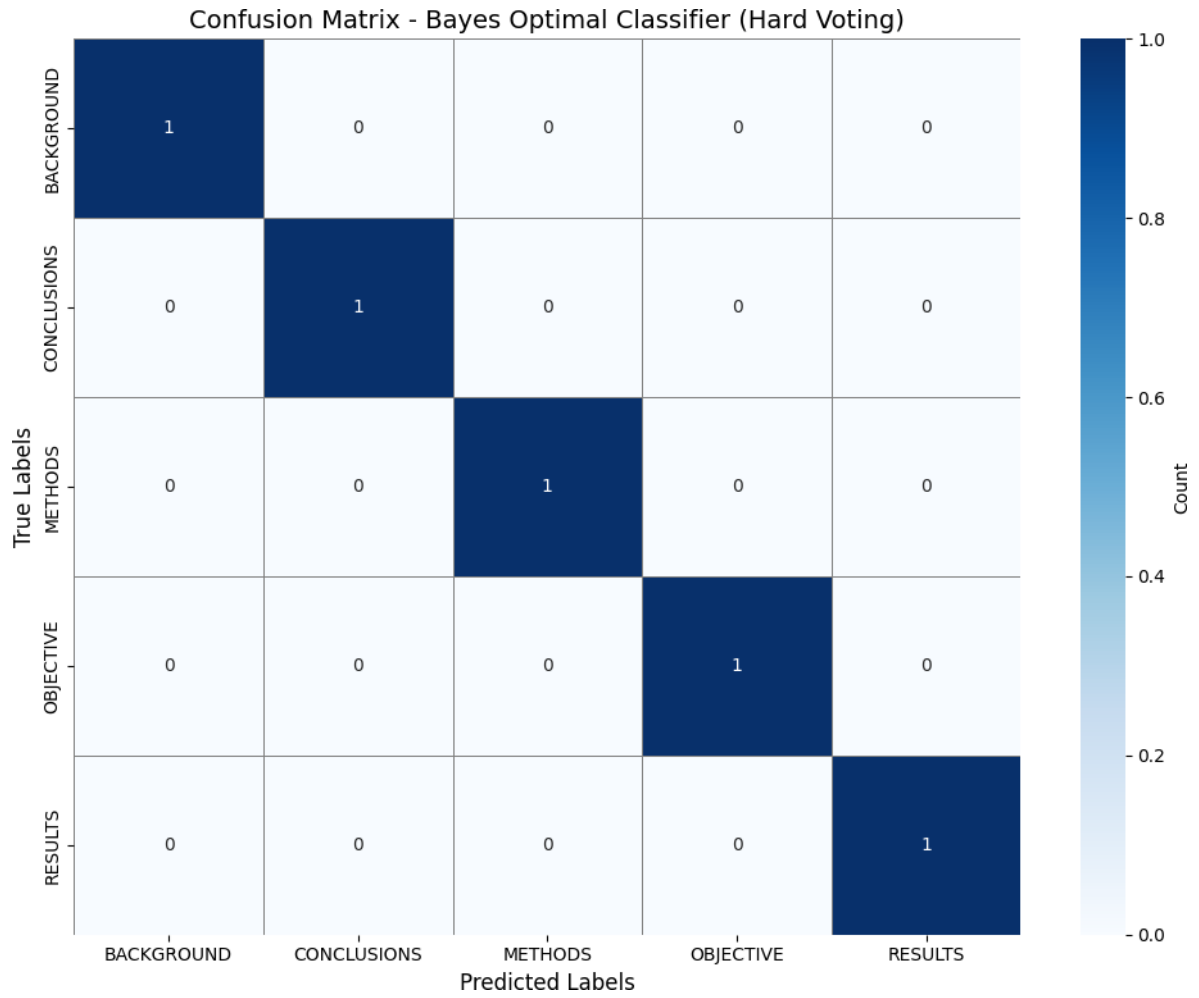
```
Individual Model Performances:
--------------------------------------------------------------------------------
NaiveBayes               : Accuracy = 1.0000 (100.00%)
LogisticRegression       : Accuracy = 1.0000 (100.00%)
RandomForest             : Accuracy = 1.0000 (100.00%)
DecisionTree             : Accuracy = 1.0000 (100.00%)
KNN                      : Accuracy = 1.0000 (100.00%)
--------------------------------------------------------------------------------

BOC Voting Classifier Accuracy: 1.0000 (100.00%)

BOC Macro F1 Score: 1.0000

Classification Report:
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

  BACKGROUND       1.00      1.00      1.00         1
 CONCLUSIONS       1.00      1.00      1.00         1
     METHODS       1.00      1.00      1.00         1
   OBJECTIVE       1.00      1.00      1.00         1
     RESULTS       1.00      1.00      1.00         1

    accuracy                           1.00         5
   macro avg       1.00      1.00      1.00         5
weighted avg       1.00      1.00      1.00         5
```


Confusion Matrix - Voting Classifier (Bayes Optimal Approximation)
True Labels
BACKGROUND
CONCLUSIONS
METHODS
OBJECTIVE
RESULTS
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
BACKGROUND
CONCLUSIONS
METHODS
OBJECTIVE
RESULTS
Predicted Labels
Count
1.0
0.8
0.6
0.4
0.2
0.0

```
======================================================================
EVALUATION: Bayes Optimal Classifier (Hard Voting)
======================================================================

Individual Model Performances:
----------------------------------------------------------------------
NaiveBayes               : Accuracy = 1.0000 (100.00%)
LogisticRegression       : Accuracy = 1.0000 (100.00%)
RandomForest             : Accuracy = 1.0000 (100.00%)
DecisionTree             : Accuracy = 1.0000 (100.00%)
KNN                      : Accuracy = 1.0000 (100.00%)
----------------------------------------------------------------------

BOC (Hard Voting) Accuracy: 1.0000 (100.00%)
BOC (Hard Voting) Macro F1 Score: 1.0000

Classification Report:
----------------------------------------------------------------------
              precision    recall  f1-score   support

  BACKGROUND       1.00      1.00      1.00         1
 CONCLUSIONS       1.00      1.00      1.00         1
     METHODS       1.00      1.00      1.00         1
   OBJECTIVE       1.00      1.00      1.00         1
     RESULTS       1.00      1.00      1.00         1

    accuracy                           1.00         5
   macro avg       1.00      1.00      1.00         5
weighted avg       1.00      1.00      1.00         5
```




```
=== Final Evaluation: Bayes Optimal Classifier (Hard Voting) ===
BOC Accuracy: 1.0000
BOC Macro F1 Score: 1.0000

Classification Report:
              precision    recall  f1-score   support

  BACKGROUND       1.00      1.00      1.00         1
 CONCLUSIONS       1.00      1.00      1.00         1
     METHODS       1.00      1.00      1.00         1
   OBJECTIVE       1.00      1.00      1.00         1
     RESULTS       1.00      1.00      1.00         1

    accuracy                           1.00         5
   macro avg       1.00      1.00      1.00         5
weighted avg       1.00      1.00      1.00         5


Confusion Matrix:
[[1 0 0 0 0]
 [0 1 0 0 0]
 [0 0 1 0 0]
 [0 0 0 1 0]
 [0 0 0 0 1]]
```



Discussion

The **scratch model (Part A)** gave a decent performance but was limited because it used only basic word counts and simple probability estimation. The **tuned Sklearn model (Part B)** performed better after using TF-IDF features and optimized parameters, which improved accuracy and F1 score. Finally, the **BOC model (Part C)** achieved the best performance overall, since it combined multiple algorithms and used their collective strength. This ensemble approach reduced errors and provided more stable predictions compared to individual models.