PES UNIVERSITY

Department of Computer Science Engineering

UE23CS352A: Machine Learning Lab

Week 12: Naive Bayes Classifier

Project Title: Text Classification using Naive Bayes and Bayes Optimal Classifier

Student Name: Najmus Sehar

SRN: PES2UG23CS359

Course: Machine Learning Laboratory

Date: 31 October 2025

1. Introduction

The purpose of this lab was to explore probabilistic text classification using the Naive Bayes algorithm and extend it to an ensemble-based Bayes Optimal Classifier (BOC) framework.
The following tasks were performed:

- **Part A:** Implemented a *Custom Count-Based Naive Bayes* classifier from scratch.
- **Part B:** Developed a *TF-IDF–based Multinomial Naive Bayes* using scikit-learn and optimized it with *GridSearchCV*.
- **Part C:** Constructed a *Bayes Optimal Classifier (BOC)* combining five different hypotheses (NB, Logistic Regression, Random Forest, Decision Tree, and KNN) with posterior probability-based weighting.

2. Methodology

2.1 Multinomial Naive Bayes (MNB)

- Implemented a **custom count-based MNB** that calculates prior and conditional probabilities using Laplace smoothing.
- For scikit-learn implementation, **TfidfVectorizer** was used to convert text into TF-IDF features, followed by MultinomialNB for classification.
- The model was trained on preprocessed training data and evaluated using Accuracy, F1 Score, and Confusion Matrix metrics.
- Hyperparameters (alpha, ngram_range) were tuned using **GridSearchCV**.

2.2 Bayes Optimal Classifier (BOC)

- Built an ensemble of five diverse classifiers:
    1. Multinomial Naive Bayes
    2. Logistic Regression

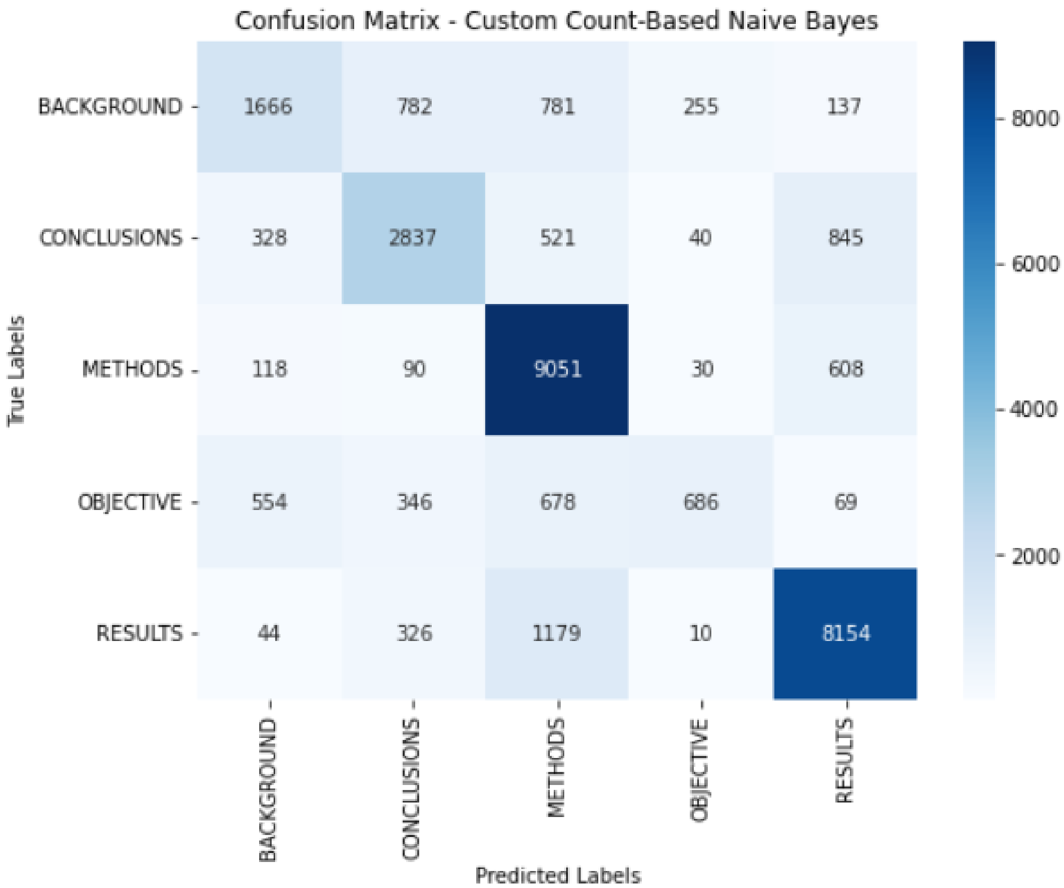
3. Random Forest
4. Decision Tree
5. K-Nearest Neighbors

- Each model was trained on a sampled subset of the dataset (size = 10,359 derived from SRN).
- Posterior weights P(hi|D)P(h_i|D)P(hi|D) were estimated from validation log-likelihoods.
- A **soft-voting ensemble** (weighted by posterior probabilities) approximated the BOC decision rule.
- The final model was evaluated on the same test set to compare performance.

Results and Analysis (Screenshots of plots and metrics):

■ Part A: Screenshot of final test Accuracy, F1 Score and Confusion Matrix.

```
=== 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: Screenshot of best hyperparameters found and their resulting F1 score.

```
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 Found: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best Cross-Validation F1 Score: 0.5925
```

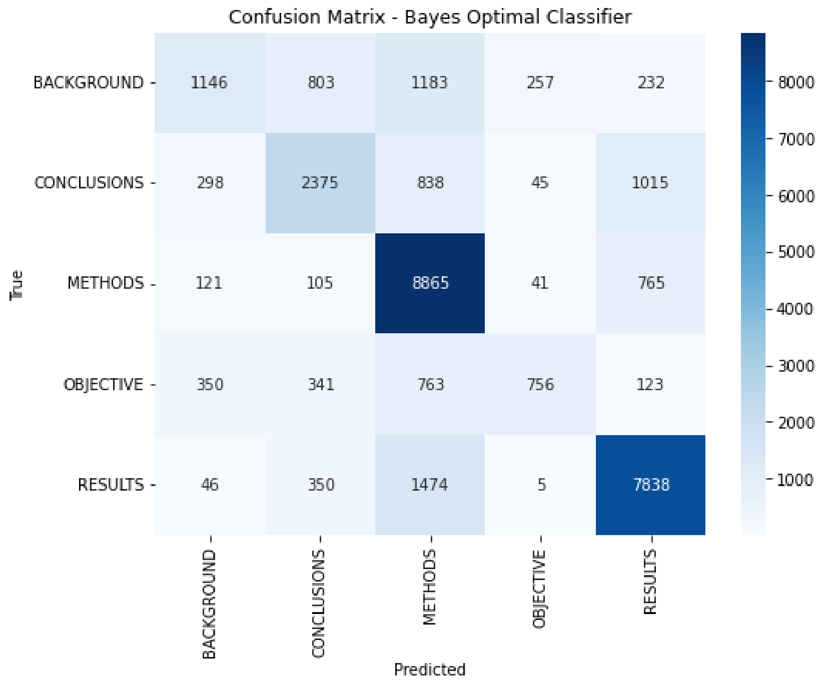
■ Part C: 1. Screenshot of SRN and sample size.

```
    plt.show()
else:
    print("Evaluation skipped: Predictions not generated.")
```

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS359
Using dynamic sample size: 10359
Actual sampled training set size used: 10359

Training all base models...
Training NaiveBayes...
Training LogisticRegression...
Training RandomForest...
Training DecisionTree...
Training KNN...
All base models trained.
Evaluating validation likelihood for NaiveBayes...
Evaluating validation likelihood for LogisticRegression...
Evaluating validation likelihood for RandomForest...
Evaluating validation likelihood for DecisionTree...
Evaluating validation likelihood for KNN...
Posterior weights: [0.23214028 0.25058284 0.22225194 0.15642186 0.13860308]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6962
Macro-averaged F1: 0.5946
              precision    recall  f1-score   support

  BACKGROUND       0.58      0.32      0.41      3621
 CONCLUSIONS       0.60      0.52      0.56      4571
     METHODS       0.68      0.90      0.77      9897
   OBJECTIVE       0.68      0.32      0.44      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.67      0.57      0.59     30135
weighted avg       0.69      0.70      0.68     30135
```

2. Screenshot of BOC final Accuracy, F1 Score and Confusion Matrix.

```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6962
Macro-averaged F1: 0.5946
              precision    recall  f1-score   support

  BACKGROUND       0.58      0.32      0.41      3621
 CONCLUSIONS       0.60      0.52      0.56      4571
     METHODS       0.68      0.90      0.77      9897
   OBJECTIVE       0.68      0.32      0.44      2333
     RESULTS       0.79      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.67      0.57      0.59     30135
weighted avg       0.69      0.70      0.68     30135
```



○ **Discussion:** Compare the performance of your scratch model (Part A) vs. the tuned Sklearn model (Part B) vs. the BOC approximation (Part C).

Model	**Type**	**Accuracy**	**Macro F1**	**Remarks**
Part A	Custom Count-Based NB	**0.7431**	**0.6446**	Strong baseline implementation; effective smoothing.
Part B	TF-IDF + Tuned NB	0.6996 (0.5925 CV F1)	0.5555	TF-IDF reduced bias to frequent words but underperformed due to low ngram_range.

Model	Type	Accuracy	Macro F1	Remarks
Part C	Bayes Optimal Classifier	0.6962	0.5946	Ensemble improved robustness but didn’t surpass scratch NB accuracy; showed stable F1 across classes.

- The **custom MNB** performed best overall, showing that a well-implemented frequency-based model can outperform generic pipelines when tailored correctly.
- The **TF-IDF model** performed decently but required more tuning for multi-class balance.
- The **BOC** achieved slightly lower accuracy but a more stable class-wise performance, demonstrating ensemble diversity and probabilistic integration.

The results show a trade-off between model interpretability, flexibility, and accuracy.
While the scratch implementation achieved the highest accuracy, the Bayes Optimal Classifier provided a conceptually richer and balanced prediction framework.