



Department of Computer Science Engineering  
**UE23CS352A: Machine Learning Lab**  
**Week 12: Naive Bayes Classifier**

NAME:BOBBA KOUSHIK

SRN:PES2UG23CS133

SEC:C

### Introduction

This lab assignment focuses on implementing and evaluating different classification techniques—specifically **Naive Bayes (NB)**—for the task of classifying text sentences from the PubMed 20k RCT dataset based on their rhetorical structure (e.g., BACKGROUND, OBJECTIVE, METHODS, RESULTS, CONCLUSIONS).]

### The core purpose was to:

- ◆ Implement the Multinomial Naive Bayes (MNB) algorithm from scratch using count-based features, ensuring correct calculation of log priors and log likelihoods with Laplace smoothing
- ◆ Evaluate the custom MNB classifier on test data.
- ◆ Implement and tune a Term Frequency-Inverse Document Frequency (TF-IDF) based MNB classifier using scikit-learn's MultinomialNB and Grid Search Cross-Validation (\$\text{GridSearchCV}\$).
- ◆ Implement and evaluate a **Bayes Optimal Classifier (BOC)** approximation using **Soft Voting** with diverse base hypotheses, where voting weights are set by calculating the **posterior probabilities** of each hypothesis ( $P(h_i | D)$ ).

## Methodology

### 1. Multinomial Naive Bayes (MNB) Implementation (Part A)

The custom NaiveBayesClassifier was implemented from scratch using the following steps:

- **Feature Extraction:** Sentences were converted into **Count-based features** using with unigrams and bigrams () and filtering low-frequency tokens ().
- **Fit Method:** The classifier calculates the **Log Prior** and the **Log Likelihood** for each word and class .
  - **Log Prior:** .
  - **Log Likelihood:** Uses Laplace (Additive) Smoothing ():

- **Predict Method:** The class probability for a given document is calculated as the sum of the log prior and the log likelihoods of all words in the document:

The final prediction is the class that yields the maximum log probability () .

## 2. TF-IDF and Hyperparameter Tuning (Part B)

The scikit-learn pipeline combined a and a standard . **GridSearchCV** was employed on the development set to tune the two most critical parameters:

- : Explored unigrams and unigrams + bigrams .
- : Explored smoothing values (e.g., ). The model achieving the best **macro F1 score** across 3-fold cross-validation was selected.

## 3. Bayes Optimal Classifier (BOC) Approximation (Part C)

The BOC was approximated using a **Soft Voting Classifier** with weights equal to the estimated posterior probability of each hypothesis .

- **Hypotheses :** Five diverse text classification pipelines were used, all based on TF-IDF features: **MultinomialNB, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors.**
- **Posterior Weight Calculation :**
  1. The sampled training data was split into a sub-training set and a small validation set.
  2. Each was retrained on the sub-training set.
  3. The **log-likelihood** was calculated as the sum of the for all documents in the validation set.
  4. The likelihoods (assuming a uniform prior ) were calculated and normalized to obtain the **Posterior Weights** .

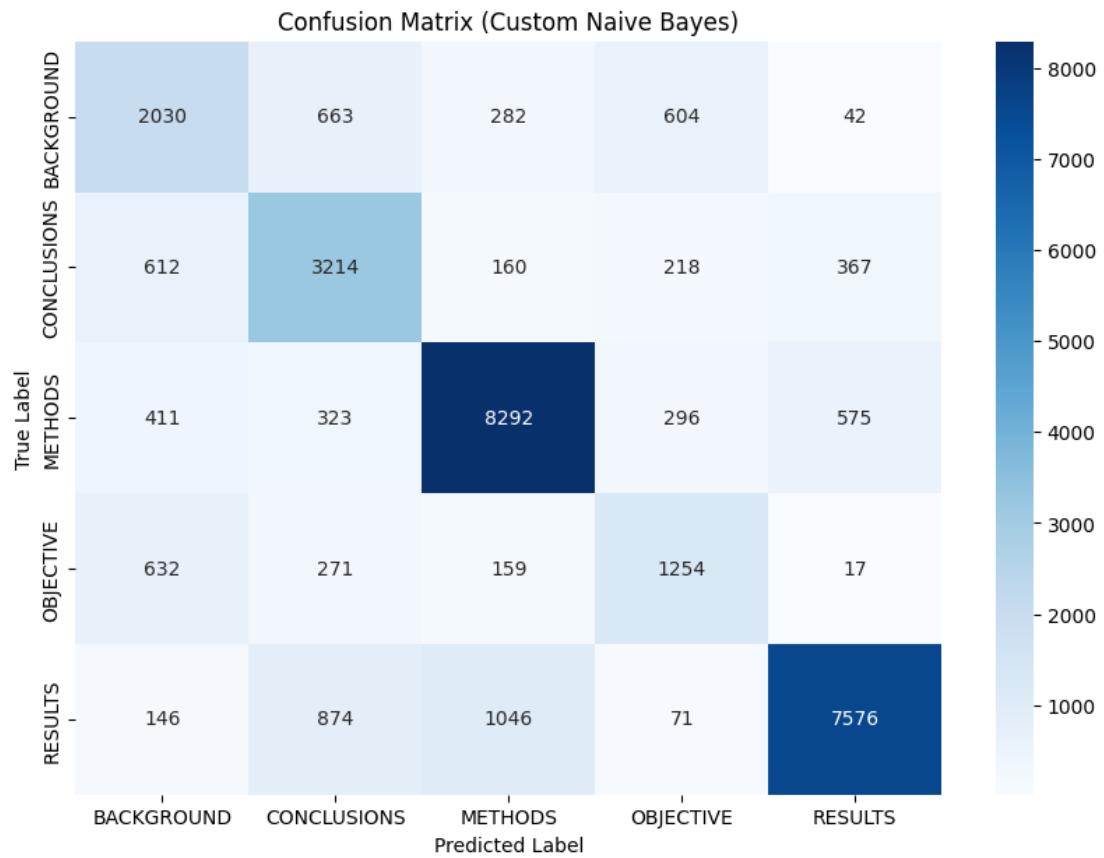
# Results and Analysis

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

```
==> Test Set Evaluation (Custom Count-Based Naive Bayes) ==>
Accuracy: 0.7422
      precision    recall  f1-score   support
  BACKGROUND       0.53      0.56      0.54     3621
CONCLUSIONS       0.60      0.70      0.65     4571
    METHODS       0.83      0.84      0.84     9897
OBJECTIVE        0.51      0.54      0.53     2333
    RESULTS       0.88      0.78      0.83     9713

      accuracy                           0.74      30135
   macro avg       0.67      0.68      0.68      30135
weighted avg       0.75      0.74      0.75      30135

Macro-averaged F1 score: 0.6765
```



## 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      --     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 6 candidates, totalling 18 fits
Grid search complete.

Best parameters found on development set: {'nb_alpha': 0.1, 'tfidf_ngram_range': (1, 1)}
Best cross-validation macro F1 score: 0.5925

```

## Part C:

### 1. Screenshot of SRN and sample size.

```

Please enter your full SRN (e.g., PES1UG22CS345): pes2ug23cs133
Using dynamic sample size: 10133
Actual sampled training set size used: 10133

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, it will be assumed that there are exactly two classes.
  warnings.warn(
    Training RandomForest...
    Training DecisionTree...
    Training KNN...
All base models trained.

```

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

```

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

Predicting on test set...

==== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ====
Accuracy: 0.7079
Macro-averaged F1 score: 0.6133
      precision    recall   f1-score   support
      BACKGROUND    0.56     0.36     0.44     3621
      CONCLUSIONS   0.60     0.56     0.58     4571
      METHODS        0.71     0.89     0.79     9897
      OBJECTIVE      0.65     0.35     0.46     2333
      RESULTS        0.79     0.81     0.80     9713
      accuracy          0.71     0.71     0.71     30135
      macro avg       0.66     0.59     0.61     30135
      weighted avg    0.70     0.71     0.69     30135

```

