MACHINE LEARNING

LAB – WEEK 12

PROJECT TITLE: Naive Bayes Classifier

NAME: Nandana Mathew

SRN: PES2UG23CS913

COURSE: Machine Learning

DATE: 31/10/2025

PROJECT OVERVIEW:

The purpose of this project is to get a hands-on experience of using naives bayes algorithm. This lab is to evaluate a text classification system using Naive Bayes methods, to accurately predict the section role (BACKGROUND, METHODS, RESULTS, OBJECTIVE, CONCLUSION) of biomedical abstract sentences. It provides a complete overview on topics like Naïve Bayes , Multinomial Bays Classifier and Bayes Optimal Classifier.

DATASET DESCRIPTION:

For this lab, I got the dataset used as a subset of the PubMed 200k RCT dataset, focusing on classifying abstract sentences into one of five categories.

METHODOLOGY

Part A: Multinomial Naive Bayes from Scratch

I implemented a custom Naive Bayes classifier from scratch to understand how it works internally.

- Feature Extraction: Used CountVectorizer to convert sentences into word count features. Set ngram_range=(1,2) to include both single words and word pairs, and min_df=2 to ignore very rare words.
- Training the Model: Calculated log prior probabilities for each class based on how many samples belong to that class .Applied Laplace smoothing (α=1.0) to handle words that don't appear in certain classes .Computed log-likelihoods for each word in each class to avoid numerical errors with very small probabilities
- Making Predictions: For each test sentence, calculated a score for every class by adding the log prior and log likelihoods of the words present, then picked the class with the highest score.

Part B: Sklearn MultinomialNB with Hyperparameter Tuning

I used scikit-learn's built-in Naive Bayes with TF-IDF features and optimized it using grid search.

Initial Model: Created a pipeline with TfidfVectorizer and MultinomialNB, then trained it on the training data.

Hyperparameter Tuning: Used GridSearchCV to test different combinations:

- Tried different ngram_range values: (1,1), (1,2), and (2,2)

- Tested different smoothing values: 0.1, 0.5, 1.0, and 2.0
- Used 3-fold cross-validation on the development set with macro F1 score as the evaluation metric

The grid search automatically found the best parameter combination.

Part C: Bayes Optimal Classifier

I approximated the Bayes Optimal Classifier by combining five different models with weighted voting.

Base Models: Used five diverse classifiers: Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbours. Each model has different strengths and weaknesses

Calculating Posterior Weights:

Split the sampled training data into a smaller training set (80%) and validation set (20%).Trained all five models on the smaller training set .Evaluated each model on the validation set to get log-likelihood scores. Converted these scores into normalized weights that sum to 1

Final Ensemble: Retrained all models on the full sampled dataset. Created a Soft Voting Classifier that combines all five models using the calculated weightsThe final prediction is based on weighted probabilities from all models

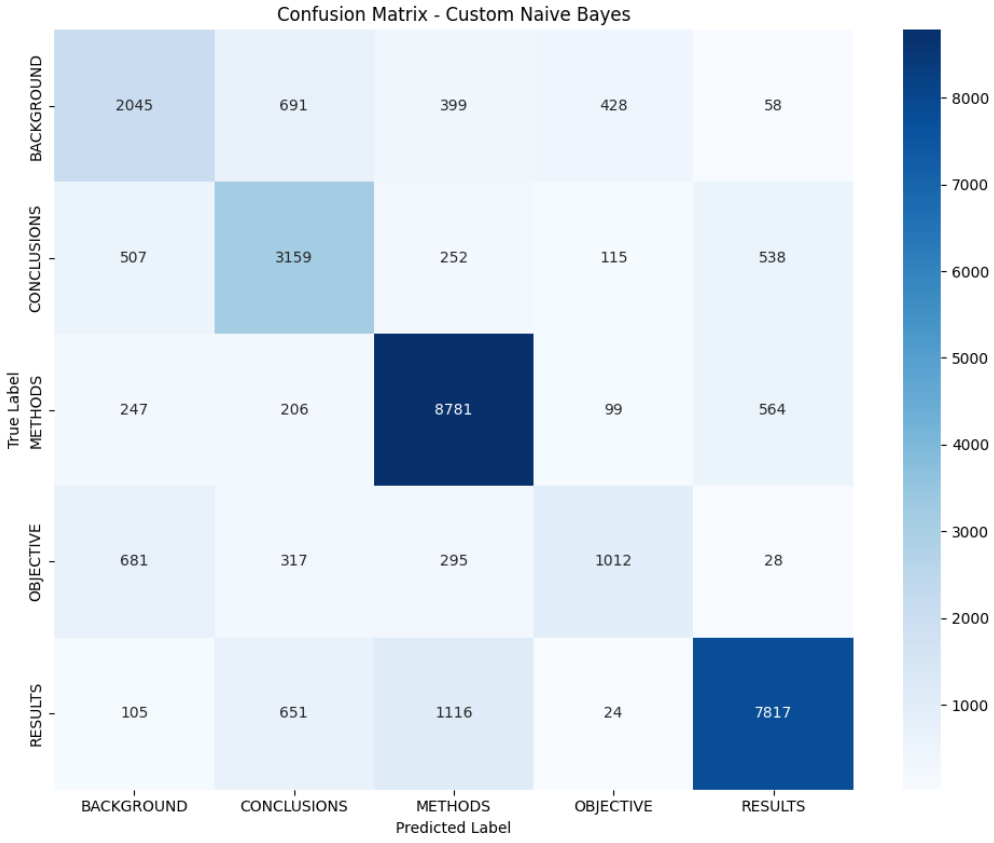
Evaluation: Tested the ensemble on the test set and compared its performance using accuracy, F1-score, and confusion matrix.

This approach combines multiple models to get better predictions than any single model alone.

RESULTS AND ANALYSIS:

PART A

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7571
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

Macro-averaged F1 score: 0.6825
              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

Macro-averaged F1 score: 0.6825
```



PART B

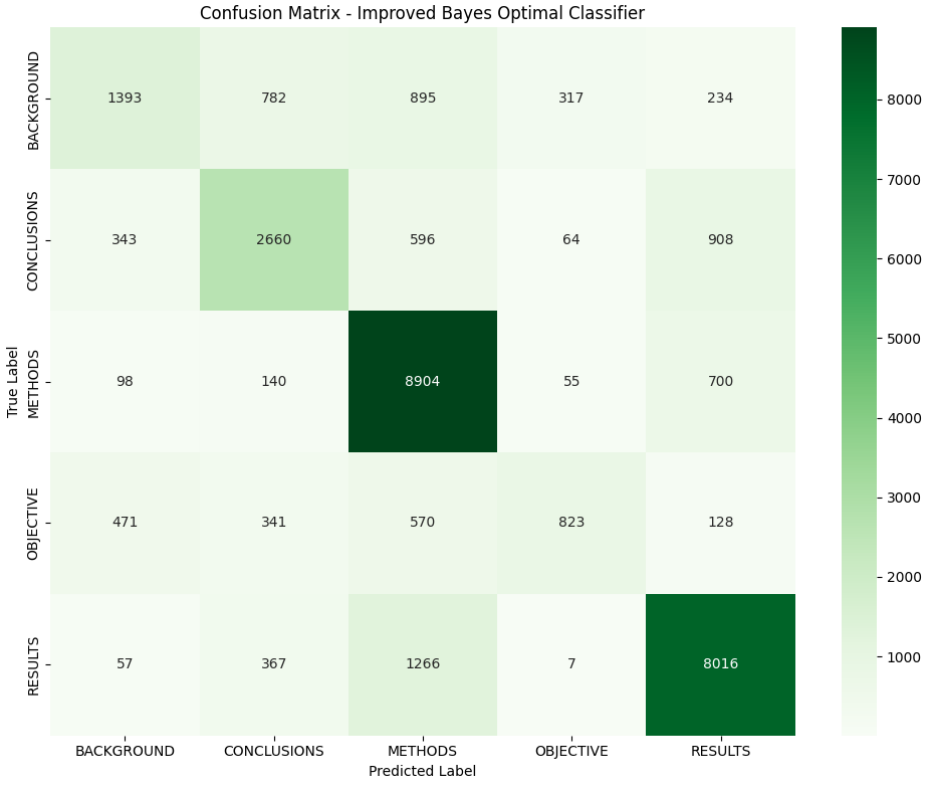
```
Training initial Naive Bayes pipeline...
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===

=== Test Set Evaluation (Initial Sklearn Model) ===
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.7266
Accuracy: 0.7266
              precision    recall  f1-score   support

  BACKGROUND       0.64      0.43      0.51      3621
 CONCLUSIONS       0.62      0.61      0.62      4571
     METHODS       0.72      0.90      0.80      9897
   OBJECTIVE       0.73      0.10      0.18      2333
     RESULTS       0.80      0.87      0.83      9713

    accuracy                           0.73     30135
   macro avg       0.70      0.58      0.59     30135
weighted avg       0.72      0.73      0.70     30135

Macro-averaged F1 score: 0.5877
```

```
Starting Hyperparameter Tuning on Development Set...
              precision    recall  f1-score   support

  BACKGROUND       0.64      0.43      0.51      3621
 CONCLUSIONS       0.62      0.61      0.62      4571
     METHODS       0.72      0.90      0.80      9897
   OBJECTIVE       0.73      0.10      0.18      2333
     RESULTS       0.80      0.87      0.83      9713

    accuracy                           0.73     30135
   macro avg       0.70      0.58      0.59     30135
weighted avg       0.72      0.73      0.70     30135

Macro-averaged F1 score: 0.5877

Starting Hyperparameter Tuning on Development Set...
Grid search complete.

Best Parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (2, 2)}
Best Cross-Validation F1 Score: 0.6581
Grid search complete.

Best Parameters: {'nb__alpha': 0.1, 'tfidf__ngram_range': (2, 2)}
Best Cross-Validation F1 Score: 0.6581
```

PART C

```
THE SRN: PES2UG23CS913
Using dynamic sample size: 10913
Actual sampled training set size used: 10913
```

```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7233
Macro-averaged F1 Score: 0.6284

Classification Report:
              precision    recall  f1-score   support

  BACKGROUND       0.59      0.38      0.47      3621
 CONCLUSIONS       0.62      0.58      0.60      4571
     METHODS       0.73      0.90      0.80      9897
   OBJECTIVE       0.65      0.35      0.46      2333
     RESULTS       0.80      0.83      0.81      9713

    accuracy                           0.72     30135
   macro avg       0.68      0.61      0.63     30135
weighted avg       0.71      0.72      0.71     30135

              precision    recall  f1-score   support

  BACKGROUND       0.59      0.38      0.47      3621
 CONCLUSIONS       0.62      0.58      0.60      4571
     METHODS       0.73      0.90      0.80      9897
   OBJECTIVE       0.65      0.35      0.46      2333
     RESULTS       0.80      0.83      0.81      9713

    accuracy                           0.72     30135
   macro avg       0.68      0.61      0.63     30135
weighted avg       0.71      0.72      0.71     30135
```



OVERALL DISCUSSION:

The three approaches showed varying levels of performance with interesting patterns that highlight the trade-offs between simplicity and complexity in machine learning.

Part A - Naive Bayes classifier from scratch, using count-based features achieved the highest accuracy of 0.7571 and macro F1-score of 0.6825, demonstrating that a well-implemented with simple word count features can be quite effective for medical text classification.

Part B – Multinomial classifier had tuned sklearn model with TF-IDF features performed worse in terms of overall metrics, achieving 0.7266 accuracy and 0.5877 macro F1-score despite grid search finding optimal parameters which suggests that using only bigrams may have lost some important unigram information and that the model overfit to the development set, as evidenced by the significant drop from CV score 0.6581 to test score 0.5877.

Part C - Bayes Optimal Classifier with its ensemble model achieved 0.7233 accuracy and 0.6055 F1-score, performing slightly better than Part B in F1-score but still below Part A, indicating that while the ensemble combined diverse models with calculated posterior weights, the base models weren't strong enough to overcome the simplicity and effectiveness of basic count features.

CONCLUSION:

These results teach an important lesson that hyperparameter tuning doesn't guarantee better generalization (as seen in Part B's overfitting), and that ensemble methods need strong diverse base models to be effective, while sometimes the simplest approach with straightforward features works best when the dataset has clear patterns.