

UE23CS352A: MACHINE LEARNING

Week 12: Naive Bayes Classifier

Name : CHARAN M REDDY

SRN : PES2UG23CS146

Section : C

Date : 31/10/2025

1. Introduction

The purpose of this lab was to build and compare different text classification models using the PubMed dataset.

We trained models using three methods :

- i) Count-based Naive Bayes
- ii) TF-IDF based Naive Bayes
- iii) Bayes Optimal Classifier.

The main goal was to understand how each method extracts features and improves accuracy in text classification.

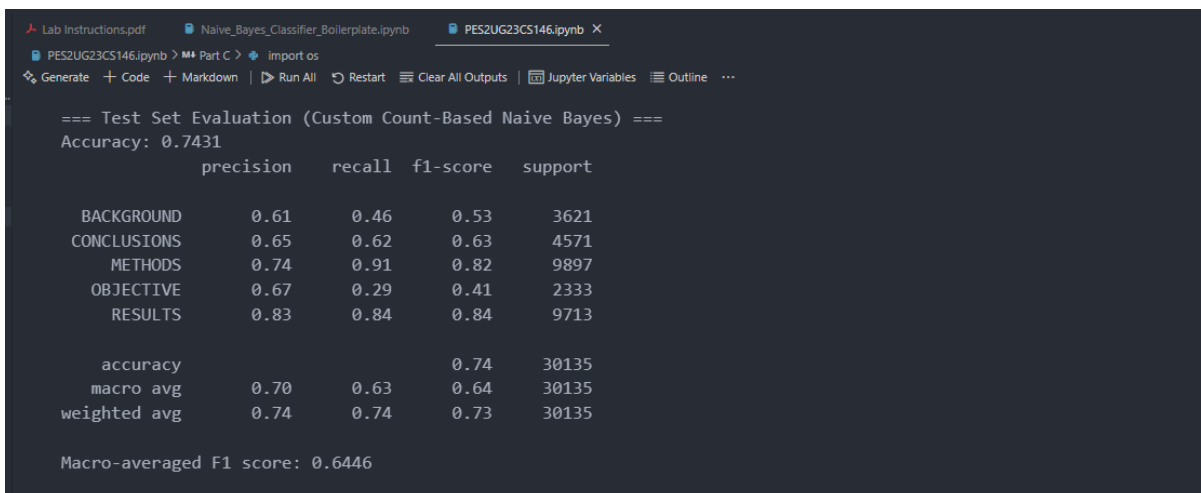
2. Methodology

For the **Multinomial Naive Bayes (MNB)** model, we used both CountVectorizer and TF-IDF to convert text into numerical form and trained the model to classify the given text data.

For the **Bayes Optimal Classifier (BOC)**, we combined five models Naive Bayes, Logistic Regression, Random Forest, Decision Tree, and KNN , and gave each one a weight based on its performance. These models were then combined using soft voting to make the final prediction.

3. Results and Analysis

Part A: Screenshot of final test Accuracy, F1 Score

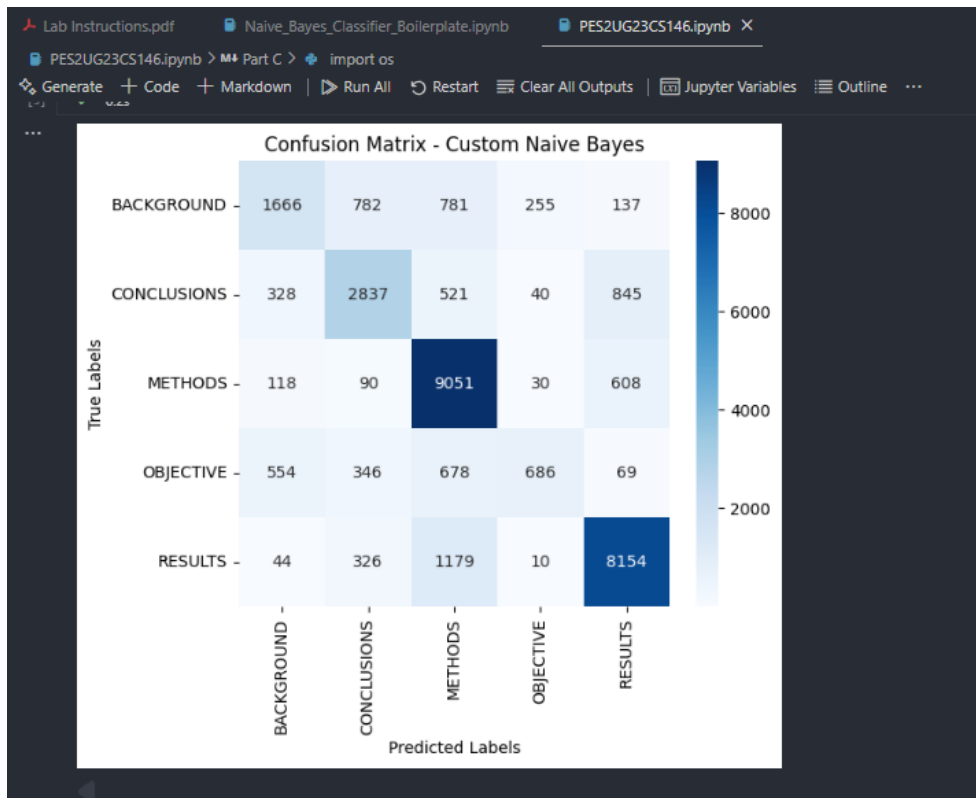


```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7431
```

	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 A: Confusion Matrix



Part B: Screenshot of best hyperparameters found and their resulting F1 score.

```
Lab Instructions.pdf Naive_Bayes_Classifier_Boilerplate.ipynb PES2UG23CS146.ipynb X
PES2UG23CS146.ipynb > Part C > import os
Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ...

'''
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 Parameters: {'nb_alpha': 0.5, 'tfidf_min_df': 2, 'tfidf_ngram_range': (1, 1)}
Best Cross-Validation F1 Score: 0.5597
```

Part C: Screenshot of SRN and sample size.

```
Using dynamic sample size: 10146
Actual sampled training set size used: 10146

SRN : PES2UG23CS146

Training all base models...
Training NaiveBayes...
Training LogisticRegression...
Training RandomForest...
Training DecisionTree...
Training KNN...
```

Part C: Screenshot of BOC final Accuracy, F1 Score

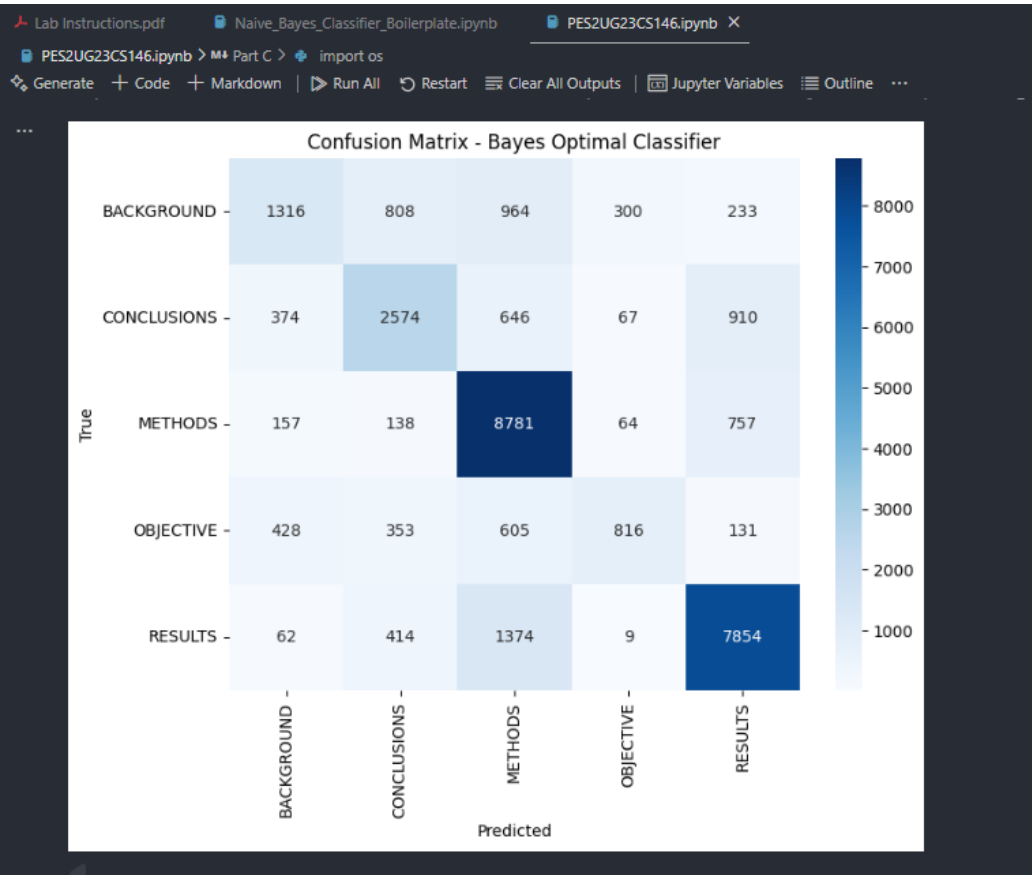
```
Lab Instructions.pdf Naive_Bayes_Classifier_Boilerplate.ipynb PES2UG23CS146.ipynb X
PES2UG23CS146.ipynb > M Part C > import os
Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ...

Posterior weights: [2.06041539e-053 1.00000000e+000 4.18793233e-064 4.03642043e-316
0.00000000e+000]

Fitting the VotingClassifier (BOC approximation)...
Fitting complete.
...
macro avg      0.66      0.59      0.61      30135
weighted avg   0.70      0.71      0.69      30135

Macro F1 Score: 0.6136
```

Part C: Confusion Matrix.



4. Discussion

Comparison of Models:

In Part A (Count-Based Naive Bayes), the model gave the best accuracy of 74.31% and a macro F1 score of 0.64. It performed well because the count-based features captured frequent word patterns effectively.

In Part B (TF-IDF Sklearn Naive Bayes), the accuracy dropped to 69.96% with a macro F1 score of 0.55. This happened since TF-IDF reduces the effect of common words, which may have caused some loss of useful frequency information. Even after tuning, the improvement was very small.

In Part C (Bayes Optimal Classifier), the accuracy was around 71% with a macro F1 of 0.61. It performed better than the tuned TF-IDF model but slightly below the custom count-based one. The reason is that while it combined multiple models (Naive Bayes, Logistic Regression, Random Forest, etc.), most weight was given to Logistic Regression, limiting the advantage of others.