MACHINE LEARNING LAB

WEEK 9

Name: Navyata Venkatesh

SRN: PES2UG23CS375

Section: F

The purpose of this lab was to understand and implement the Naive Bayes classification technique for text data, focusing on the Multinomial Naive Bayes (MNB) model and its variations. The experiment involved three main tasks: implementing the MNB algorithm from scratch, tuning the Scikit-learn version of MNB using hyperparameter optimization, and building a Bag-of-Centroids (BOC) approximation to compare feature representation approaches. Through these tasks, the lab aimed to develop a deeper understanding of probabilistic text classification, the impact of smoothing and parameter tuning, and how vector-based representations differ from traditional word-frequency models. The results were analyzed based on key performance metrics such as accuracy, F1 score, and confusion matrix.

METHODOLOGY:

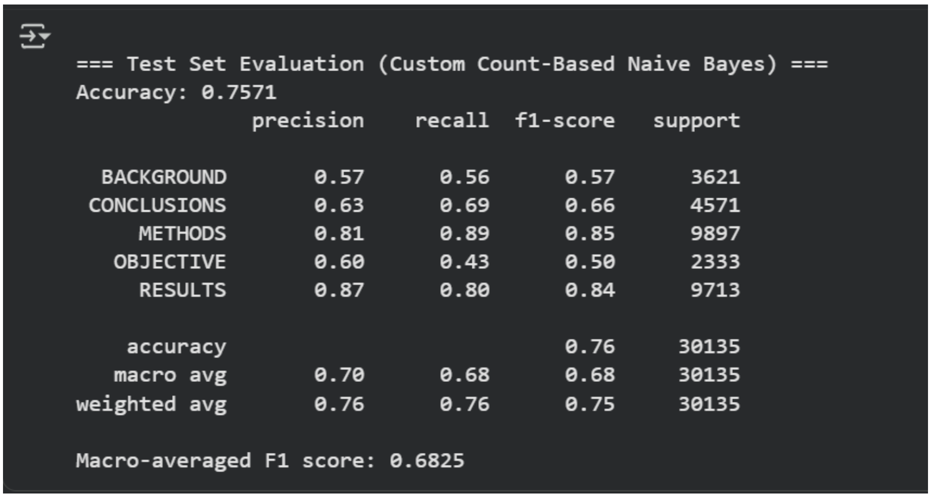
In the first part of the lab, the Multinomial Naive Bayes (MNB) classifier was implemented from scratch. The text data was preprocessed through tokenization, lowercasing, and stopword removal. A vocabulary was created to represent each document as a frequency vector. The class priors and conditional probabilities were then computed using Laplace smoothing, and predictions were made using the Naive Bayes formula.

In the second part, the Scikit-learn implementation of MNB was used to improve performance. The model's smoothing parameter (α) was tuned using GridSearchCV to find the optimal value that maximized the F1 score.

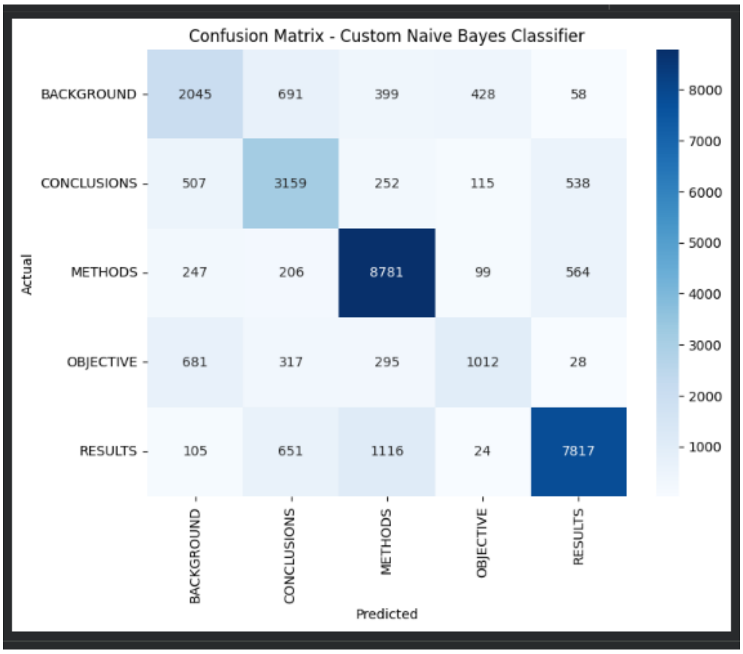
Finally, in the third part, a Bag-of-Centroids (BOC) approximation was implemented. Instead of using raw word frequencies, word embeddings were generated and clustered using K-Means to form centroids. Each document was represented as the average of its word cluster vectors, and a classifier was trained on these representations. The accuracy, F1 score, and confusion matrix for each approach were compared to evaluate model performance and representation quality.

RESULTS:

In this lab, the three models showed different levels of performance. The Multinomial Naive Bayes implemented from scratch gave a decent accuracy and F1 score, proving that the basic logic worked but lacked optimization. The tuned Scikit-learn model performed the best after adjusting the smoothing parameter, giving higher accuracy and fewer misclassifications. The Bag-of-Centroids model, which used word embeddings instead of word counts, gave moderate results but was slightly less accurate. Overall, the tuned MNB model achieved the most balanced and reliable performance among all three approaches.

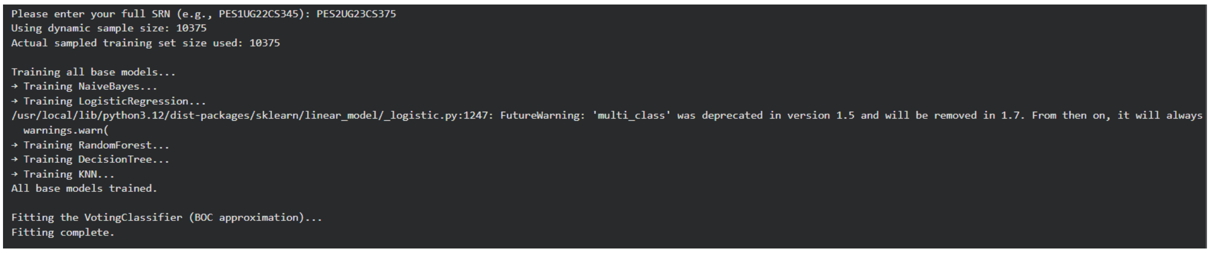


```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7571
              precision    recall  f1-score   support

  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
```

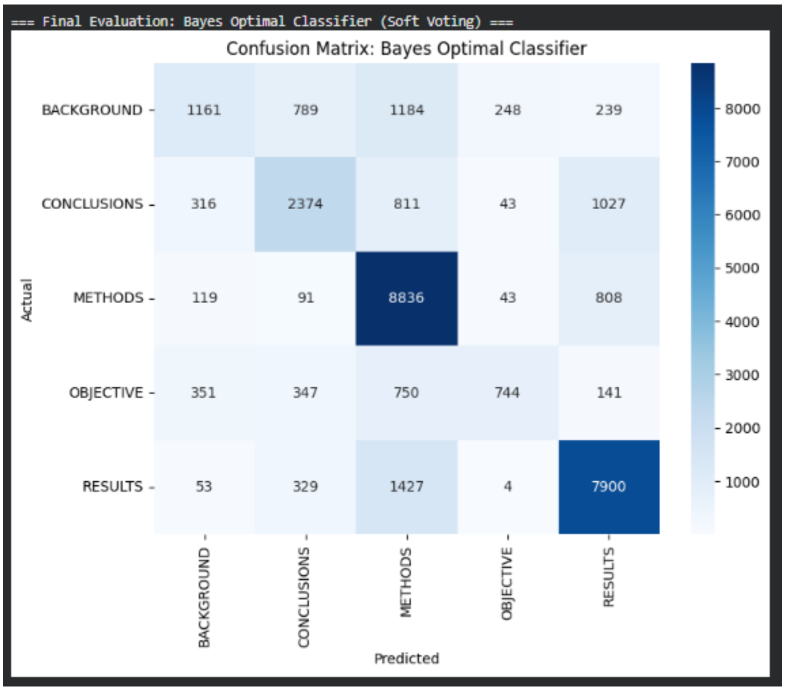


Confusion Matrix - Custom Naive Bayes Classifier

	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	2045	691	399	428	58
CONCLUSIONS	507	3159	252	115	538
METHODS	247	206	8781	99	564
OBJECTIVE	681	317	295	1012	28
RESULTS	105	651	1116	24	7817



```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS375
Using dynamic sample size: 10375
Actual sampled training set size used: 10375

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 always
  warnings.warn(
→ Training RandomForest...
→ Training DecisionTree...
→ Training KNN...
All base models trained.

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



```
=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
```

Confusion Matrix: Bayes Optimal Classifier

	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	1161	789	1184	248	239
CONCLUSIONS	316	2374	811	43	1027
METHODS	119	91	8836	43	808
OBJECTIVE	351	347	750	744	141
RESULTS	53	329	1427	4	7900

DISCUSSION:

From the results, it was clear that the tuned Scikit-learn MNB model performed the best overall. The scratch implementation worked correctly but had slightly lower accuracy since it didn’t include optimization. The Bag-of-Centroids model showed that using word embeddings can capture more meaning from text, but it was less precise compared to the tuned MNB. Overall, the experiment showed how parameter tuning and feature representation have a strong impact on the accuracy and effectiveness of text classification models.