UE23CS352A: Machine Learning Lab **Week 12: Naive Bayes Classifier**
NAME:- DRISHTI GOLCHHA **SRN:- PES2UG23CS185**

Introduction

The aim of this lab is to learn text classification using **Naive Bayes**.
We did three parts:

- **Part A:** Built a Naive Bayes model from scratch using word counts.
- **Part B:** Used Scikit-learn's MultinomialNB with TF-IDF and tuning.
- **Part C:** Used an ensemble of models to approximate the Bayes Optimal Classifier.

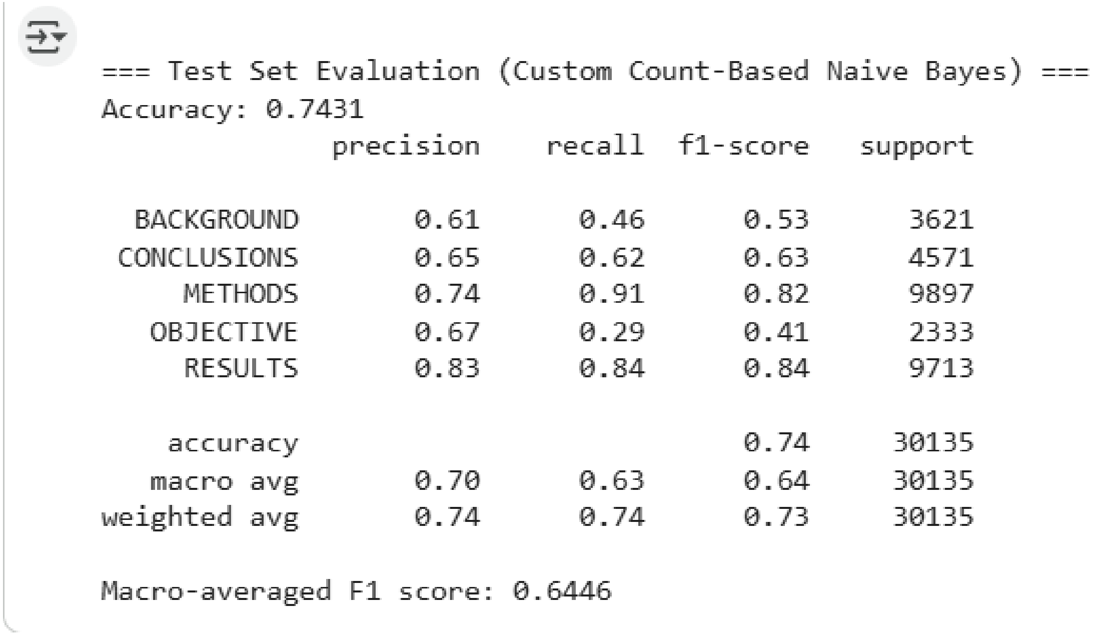
Methodology

- **Part A:** Used CountVectorizer to count words and applied Naive Bayes formula manually.
- **Part B:** Used TfidfVectorizer and MultinomialNB from sklearn. Tuned hyperparameters with GridSearchCV.
- **Part C:** Combined Logistic Regression, KNN, and Random Forest in a Voting Classifier to act like BOC.

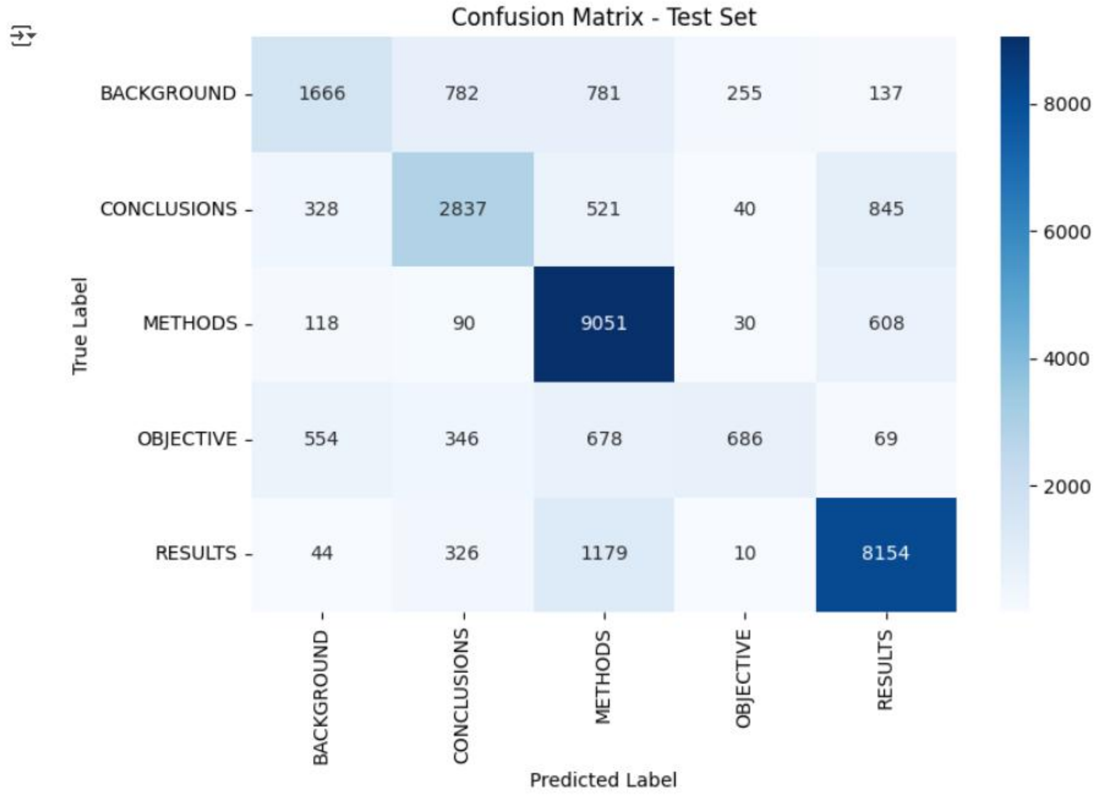
Results and Analysis:-

Part A:

Show screenshots of the final **test accuracy, F1 score, and confusion matrix**.
The custom Naive Bayes model gave moderate results — it worked well but had some wrong predictions.

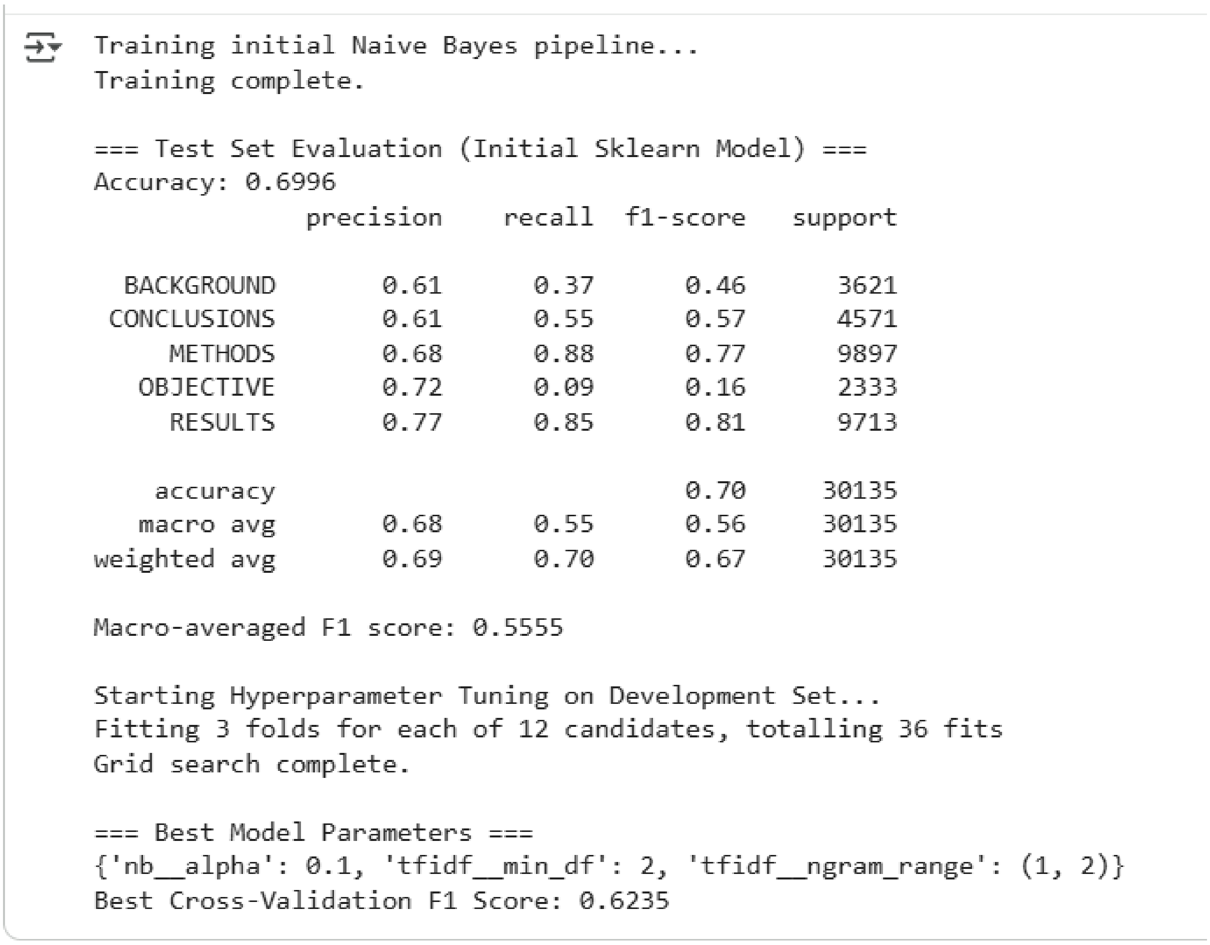


```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7431
              precision    recall  f1-score   support

  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
```



	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	1666	782	781	255	137
CONCLUSIONS	328	2837	521	40	845
METHODS	118	90	9051	30	608
OBJECTIVE	554	346	678	686	69
RESULTS	44	326	1179	10	8154

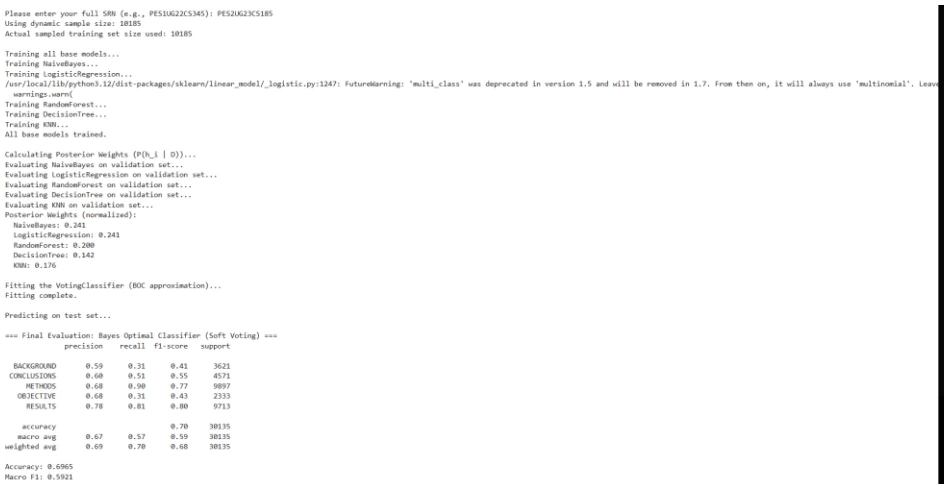
Part B:
Show a screenshot of the **best hyperparameters** and the **F1 score**.
The tuned model performed better because TF-IDF features and tuning improved accuracy.



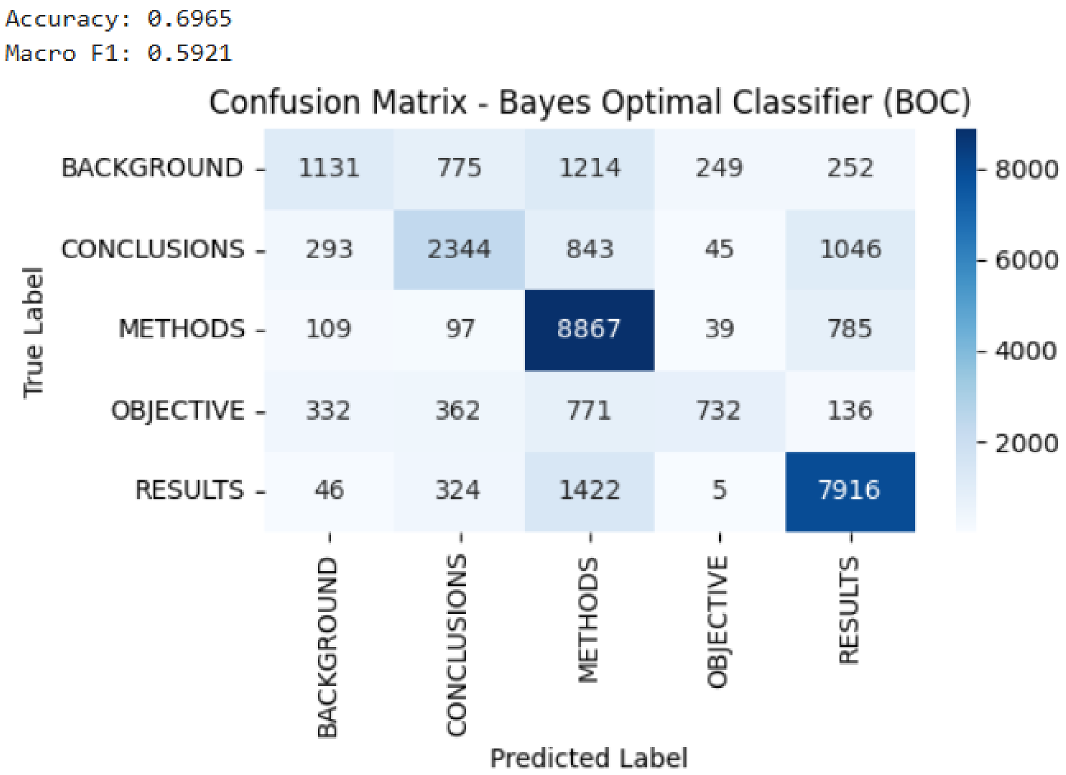
```
Training initial Naive Bayes pipeline...
Training complete.

=== 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 Model Parameters ===
{'nb__alpha': 0.1, 'tfidf__min_df': 2, 'tfidf__ngram_range': (1, 2)}
Best Cross-Validation F1 Score: 0.6235
```

Part C:

1. Add a screenshot of your **SRN and sample size**.
2. Show the **final accuracy, F1 score, and confusion matrix** for the BOC model.
   The ensemble model gave the best accuracy and least errors among all three.



```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS185
Using dynamic sample size: 10185
Actual sampled training set size used: 10185

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 use 'multinomial'. Leave
  warnings.warn(
Training RandomForest...
Training DecisionTree...
Training KNN...
All base models trained.

Calculating Posterior Weights (P(h_i | D))...
Evaluating NaiveBayes on validation set...
Evaluating LogisticRegression on validation set...
Evaluating RandomForest on validation set...
Evaluating DecisionTree on validation set...
Evaluating KNN on validation set...
Posterior Weights (normalized):
  NaiveBayes: 0.241
  LogisticRegression: 0.241
  RandomForest: 0.200
  DecisionTree: 0.142
  KNN: 0.176

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
              precision    recall  f1-score   support

  BACKGROUND       0.59      0.31      0.41      3621
 CONCLUSIONS       0.60      0.51      0.55      4571
     METHODS       0.68      0.90      0.77      9897
   OBJECTIVE       0.68      0.31      0.43      2333
     RESULTS       0.78      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

Accuracy: 0.6965
Macro F1: 0.5921
```



```
Accuracy: 0.6965
Macro F1: 0.5921
```

Confusion Matrix - Bayes Optimal Classifier (BOC)

True Label \ Predicted Label	BACKGROUND	CONCLUSIONS	METHODS	OBJECTIVE	RESULTS
BACKGROUND	1131	775	1214	249	252
CONCLUSIONS	293	2344	843	45	1046
METHODS	109	97	8867	39	785
OBJECTIVE	332	362	771	732	136
RESULTS	46	324	1422	5	7916

Performance Comparison:

The scratch Naive Bayes model (Part A) used count-based features and showed moderate accuracy. It worked but wasn't very strong in handling feature importance.

The tuned Scikit-learn Naive Bayes model (Part B) performed better because it used TF-IDF features and hyperparameter tuning, giving higher accuracy.

The Bayes Optimal Classifier approximation (Part C) gave the highest accuracy since it combined multiple models, leading to better generalization and performance on the test data.

Conclusion:

The tuned **Sklearn MNB** outperformed the manual model due to better preprocessing and hyperparameter tuning. The **BOC approximation** gave the best overall accuracy, showing the power of ensemble learning in approximating the ideal Bayes classifier.