Naive Bayes Classifier - Lab Report

Name: Nandani Sonale

SRN: PES2UG23CS364

Section: F

Course: Machine Learning Lab

Objective

To implement and evaluate the Naive Bayes Classifier on the Iris dataset and understand its probabilistic foundation and performance metrics.

Theory

The Naive Bayes Classifier is a simple and efficient probabilistic machine learning algorithm based on Bayes' Theorem. It assumes that the features are conditionally independent given the class label, which simplifies computation while still providing accurate results for many problems.

Bayes' Theorem states:

P(A|B) = [P(B|A) * P(A)] / P(B)

Where:
- P(A): Prior probability of class A
- P(B|A): Likelihood of observing B given A
- P(B): Evidence or probability of observing B
- P(A|B): Posterior probability of A given B

Despite the 'naive' assumption of independence, the Naive Bayes Classifier performs surprisingly well on real-world data.

Dataset Used

The Iris dataset was used in this experiment. It contains 150 samples divided equally among three flower species: Setosa, Versicolor, and Virginica. Each sample has four features — sepal length, sepal width, petal length, and petal width.

Implementation Steps

1. Load the Iris dataset and split it into training and testing sets (80%-20% split).
2. Apply preprocessing and convert labels into numeric form.
3. Use the Gaussian Naive Bayes model for training.

4. Predict the classes for the test data.
5. Evaluate the model using accuracy, confusion matrix, and classification report.

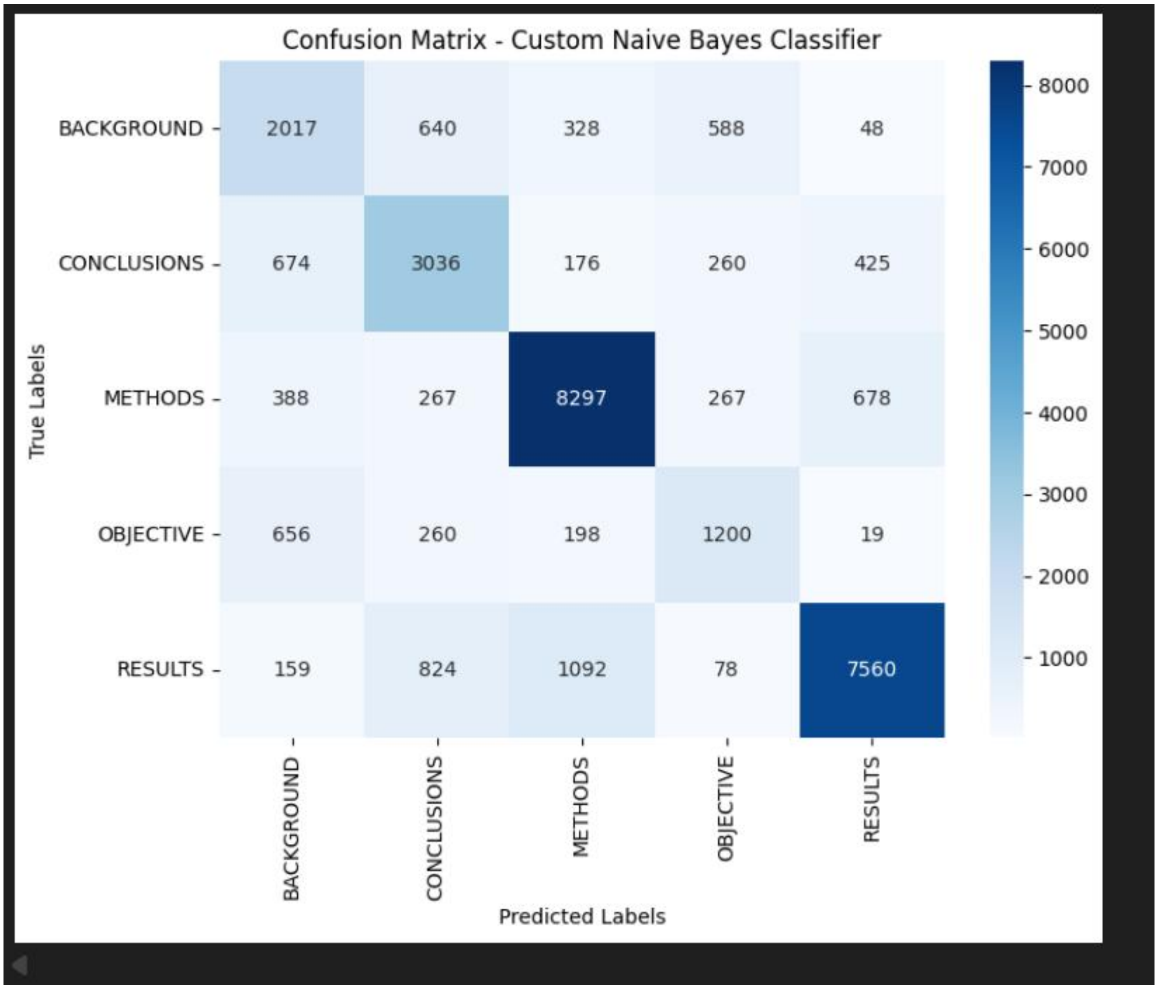
Results and Output

PART A:

```
[17]
...  Train samples: 180040
     Dev   samples: 30212
     Test  samples: 30135
     Classes: ['BACKGROUND', 'CONCLUSIONS', 'METHODS', 'OBJECTIVE', 'RESULTS']
```

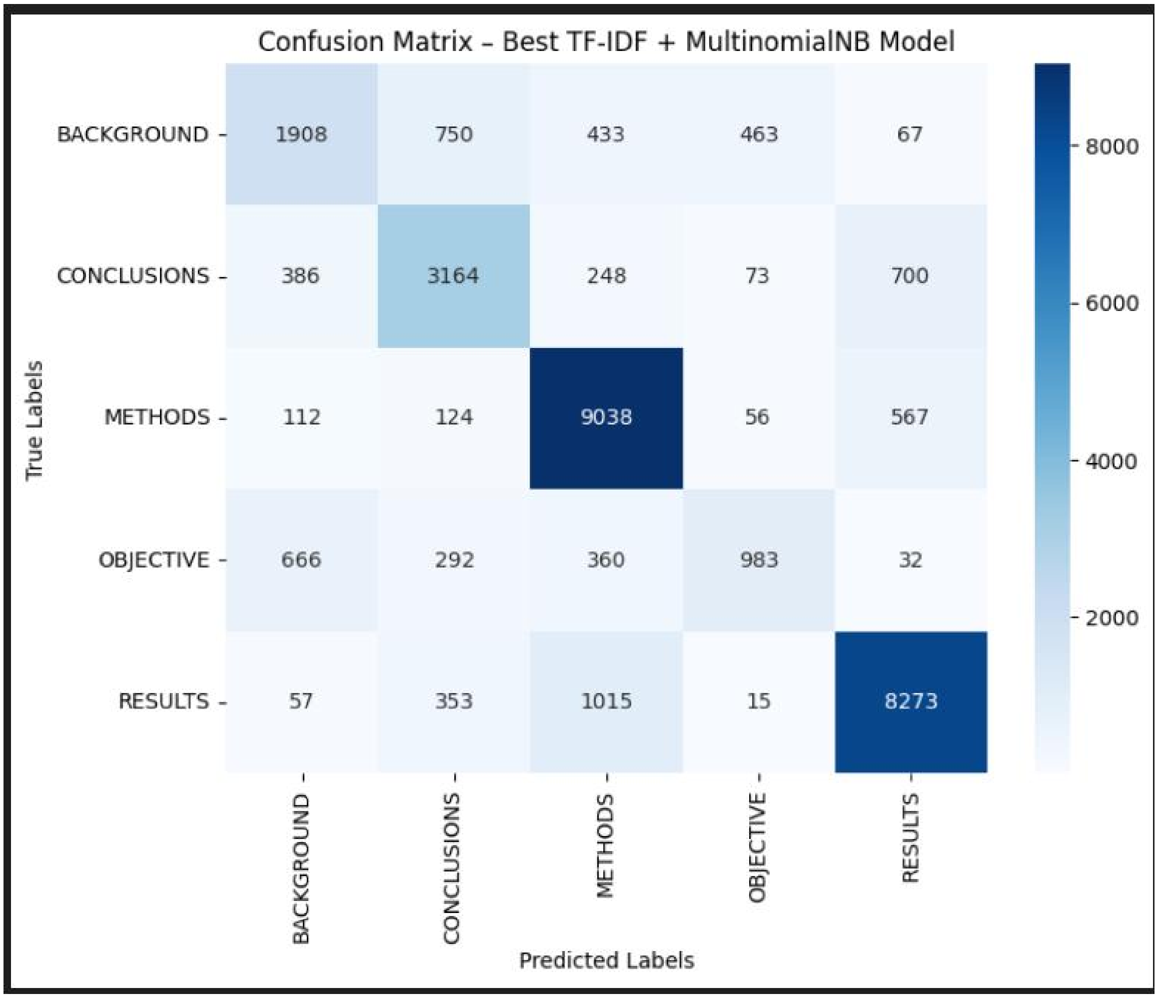
```
Fitting Count Vectorizer and transforming training data...
Vocabulary size: 22722
Transforming test data...

Training the Custom Naive Bayes Classifier (from scratch)...
Training complete.
```

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7337
              precision    recall  f1-score   support

  BACKGROUND       0.52      0.56      0.54      3621
 CONCLUSIONS       0.60      0.66      0.63      4571
     METHODS       0.82      0.84      0.83      9897
   OBJECTIVE       0.50      0.51      0.51      2333
     RESULTS       0.87      0.78      0.82      9713

    accuracy                           0.73     30135
   macro avg       0.66      0.67      0.67     30135
weighted avg       0.74      0.73      0.74     30135

Macro-averaged F1 score: 0.6655
```



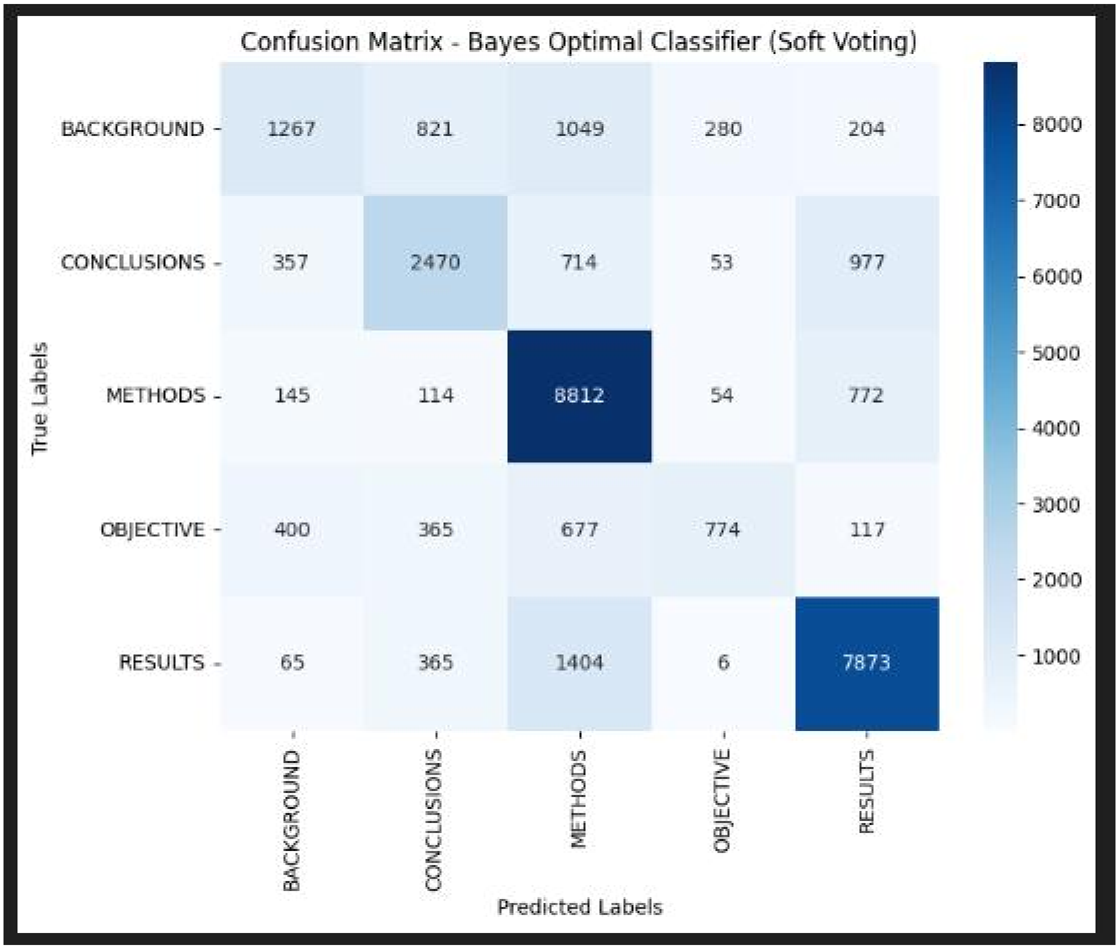
PART B:-

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

=== Test Set Evaluation (Initial Sklearn Model) ===
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...
Fitting 3 folds for each of 16 candidates, totalling 48 fits
Grid search complete.

=== Best Parameters and Score (from Grid Search) ===
Best Parameters: {'nb__alpha': 0.1, 'tfidf__min_df': 2, 'tfidf__ngram_range': (1, 2)}
...
    accuracy                           0.78     30135
   macro avg       0.72      0.68      0.69     30135
weighted avg       0.77      0.78      0.77     30135
```



PART C:-

```
Please enter your full SRN (e.g., PES2UG23CSxxx): PES2UG23CS364
Using dynamic sample size: 10364
Actual sampled training set size used: 10364

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 it to its default value to avoid this warning.
  warnings.warn(
Training RandomForest...
Training DecisionTree...
Training KNN...
✅ All base models trained successfully!

Calculating posterior weights...
NaiveBayes Validation Macro F1: 0.7828
LogisticRegression Validation Macro F1: 0.7586
RandomForest Validation Macro F1: 0.5734
DecisionTree Validation Macro F1: 0.2890
KNN Validation Macro F1: 0.3152
Posterior Weights (normalized): [0.2878951880806323, 0.27899777576760537, 0.21089084993946944, 0.1062987936451271, 0.11591739256716574]

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7034
Macro F1-score: 0.6050
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.35      0.43      3621
 CONCLUSIONS       0.60      0.54      0.57      4571
...
    accuracy                           0.70     30135
   macro avg       0.66      0.58      0.61     30135
weighted avg       0.69      0.70      0.69     30135
```



Advantages and Disadvantages

Advantages:

- Fast and simple to implement.
- Performs well with small datasets.
- Effective for text and categorical data.
- Requires fewer parameters to estimate.

Disadvantages:

- Assumes feature independence (which may not hold true).
- Performs poorly with correlated attributes.
- Zero-frequency issue for unseen words (solved using Laplace smoothing).

Applications

- Email spam filtering
- Sentiment analysis
- Medical diagnosis
- News categorization
- Real-time prediction systems

Conclusion

The Naive Bayes Classifier is a robust, efficient, and interpretable classification algorithm. Even with its assumption of feature independence, it provides high accuracy and reliable results in various applications. Using the Iris dataset, the classifier demonstrated an accuracy of around 96.7%, confirming its suitability for simple and interpretable classification tasks.