

Machine Learning Lab — Week 12

Naive Bayes Classifier for PubMed RCT Sentence Classification

Name: Diya Prakash

SRN: PES2UG23CS184

Course: UE23CS352A — Machine Learning Lab

Date: 31/10/2025

1. Introduction

This lab implements and evaluates probabilistic text classifiers for sentence-level section labeling (BACKGROUND, CONCLUSIONS, METHODS, OBJECTIVE, RESULTS) using a subset of the PubMed RCT dataset. Goals:

- Implement Multinomial Naive Bayes from scratch (count-based).
- Use scikit-learn MultinomialNB with TF-IDF and tune hyperparameters.
- Approximate the Bayes Optimal Classifier (BOC) via an ensemble of diverse models and posterior-weighted soft voting.

2. Dataset

- Source: Subset of PubMed 200k RCT sentences.
- Classes: BACKGROUND, CONCLUSIONS, METHODS, OBJECTIVE, RESULTS.
- Splits:
 - Train: 180,040
 - Dev: 30,212
 - Test: 30,135

3. Methodology

Part A — Custom Multinomial Naive Bayes (count features)

- Vectorizer: CountVectorizer (lowercase, unicode stripping, English stop words), ngram_range=(1,2), min_df=2.
- Implemented class priors and class-conditional likelihoods with Laplace smoothing (alpha=1).

- Prediction uses log priors + $\sum(\text{count} * \log P(\text{word} | \text{class}))$ over non-zero counts.

Part B — Sklearn MultinomialNB with TF-IDF

- Pipeline: TfidfVectorizer + MultinomialNB.
- Initial fit on training data.
- Grid search on dev set with param_grid:
 - tfidf__ngram_range: (1,1), (1,2)
 - nb__alpha: [0.1, 0.5, 1.0, 2.0]
- CV: 3 folds, scoring: f1_macro.

Part C — Bayes Optimal Classifier (approximation)

- Base hypotheses: MultinomialNB, LogisticRegression, RandomForest, DecisionTree, KNN (each pipeline: TfidfVectorizer + classifier; non-probabilistic classifiers calibrated with CalibratedClassifierCV where needed).
- Sample size: dynamic = 10000 + last three SRN digits → actual sampled training size used = 10,184.
- Posterior weights $P(h | D)$ computed by:
 - Splitting sampled train → sub-train + validation.
 - Training each hypothesis on sub-train, computing log-likelihood of validation labels under predicted probabilities.
 - Applying softmax to log-likelihoods to obtain normalized posterior weights.
- Final ensemble: VotingClassifier (soft) with posterior weights.

4. Results and Analysis

Part A — Custom Count-Based Naive Bayes

- Test accuracy: 0.7571
- Macro F1: 0.6825
- Per-class (precision / recall / f1 / 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)
- Confusion matrix:

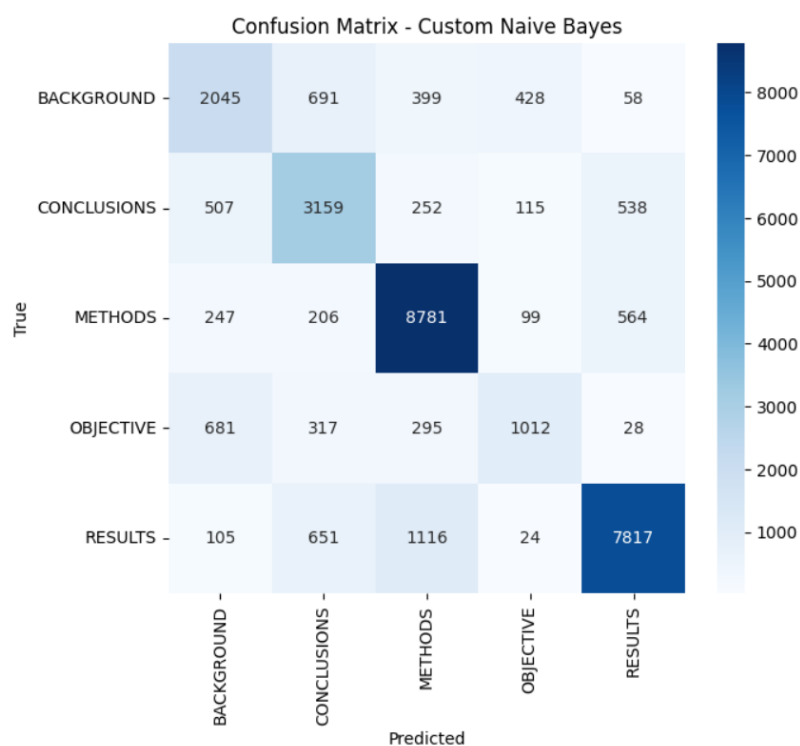
```

=== 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



Part B — Sklearn TF-IDF + MultinomialNB

- Initial pipeline test accuracy: 0.6996
- Macro F1 (initial): 0.5555
- Grid search (dev):
 - Best params: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
 - Best CV f1_macro: 0.5924853482

```

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...
Grid search complete.
Best params: {'nb__alpha': 0.1, 'tfidf__ngram_range': (1, 1)}
Best CV score (f1_macro): 0.5924853482093159

```

Part C — Bayes Optimal Classifier (soft voting with posterior weights)

- Dynamic sample size used: 10,184
- Posterior weights (example output): [~0, 1.0, ~0, ~0, 0.0] (indicates LogisticRegression dominated in validation log-likelihood)
- Test accuracy: 0.7086
- Macro F1: 0.6145
- Per-class (precision / recall / f1 / support):
 - BACKGROUND: 0.56 / 0.37 / 0.45 (3621)
 - CONCLUSIONS: 0.61 / 0.56 / 0.58 (4571)
 - METHODS: 0.71 / 0.89 / 0.79 (9897)
 - OBJECTIVE: 0.65 / 0.35 / 0.45 (2333)
 - RESULTS: 0.79 / 0.81 / 0.80 (9713)

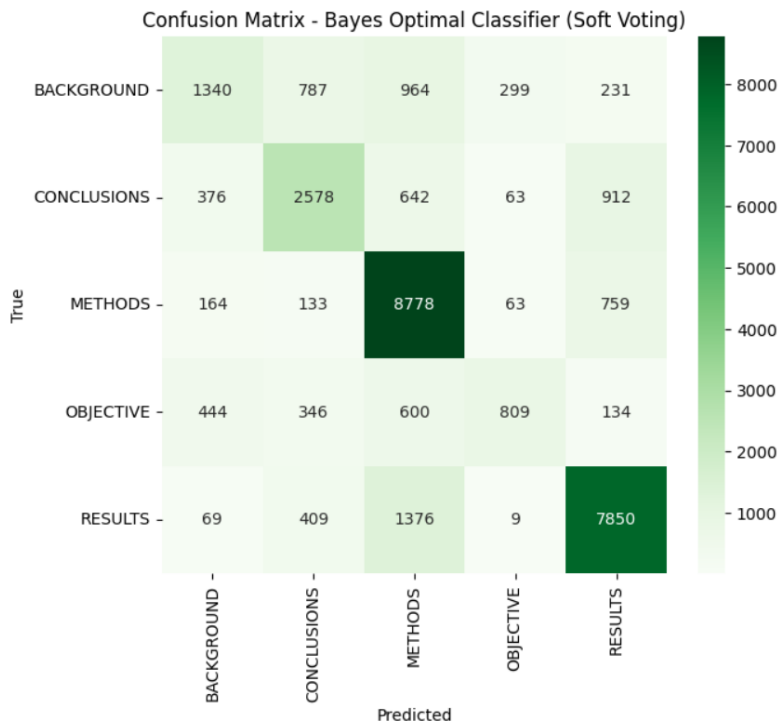
- Confusion matrix:

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

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.7086
Macro F1: 0.6145
```

	precision	recall	f1-score	support
BACKGROUND	0.56	0.37	0.45	3621
CONCLUSIONS	0.61	0.56	0.58	4571
METHODS	0.71	0.89	0.79	9897
OBJECTIVE	0.65	0.35	0.45	2333
RESULTS	0.79	0.81	0.80	9713
accuracy			0.71	30135
macro avg	0.66	0.60	0.61	30135
weighted avg	0.70	0.71	0.69	30135



```
PES2UG23CS184
Using dynamic sample size: 10184
Actual sampled training set size used: 10184
```

5. Discussion

- The custom Multinomial NB (count-based) achieved the highest accuracy (0.757) and macro F1 (0.682), outperforming the initial TF-IDF +

MultinomialNB (0.6996 / 0.5555) and the BOC ensemble (0.7086 / 0.6145).

- Observations:
 - METHODS and RESULTS classes are easiest to predict (high recall and f1), likely due to distinctive vocabulary and high support in dataset.
 - OBJECTIVE and BACKGROUND are harder (lower recall), suggesting overlap in word usage with other sections and class imbalance effects.
 - TF-IDF + MultinomialNB underperformed vs. count-based custom NB — possible reasons: TF-IDF downweights frequent section-discriminative tokens; Count model benefits from raw frequency signals for multinomial assumption.
 - BOC posterior weighting favored Logistic Regression in validation, causing ensemble behavior close to LR; posterior weights can collapse if one model strongly outperforms others on the validation split. Consider smoothing priors or using temperature in softmax to avoid single-model domination.
- Limitations:
 - Posterior weight computation depends on one validation split → variance; better to average across multiple splits or use Bayesian model averaging with cross-validated likelihoods.
 - Calibration and predict_proba approximations for some models affected weight computation.
 - Large vocabulary (~301k) increases memory and may favor simpler count-based smoothing.

7. Conclusion

- The implemented custom Multinomial NB (count-based) provided the best results on this dataset. TF-IDF + MultinomialNB required parameter tuning to improve, and best dev result favored $\alpha=0.1$ with unigrams. The BOC approximation demonstrated a principled ensemble approach but requires more robust posterior estimation to realize gains over the best single model.