Machine Learning Homework 1

B12508026 戴偉璿

October 30, 2025

Contents

Usage of Artificial Intelligence Tools

In the Bayesian Decision Model (NumPy version), I did not use any large language model (LLM) tools to generate any part of the code; all code was written entirely by myself.

However, since I was not familiar with parallel processing in Python, I used Gemini to assist in generating the parallel processing code based on my initial implementation.

For the Random Forest Classifier, I used Gemini to generate the initial structure of the code using Python's scikit-learn library. After receiving the generated code, I reviewed and modified it to meet the specific requirements of the assignment, including adding evaluation metrics and plotting the ROC curve.

For the report, to make my expression more academic and formal, I used ChatGPT to help refine and polish the language in certain sections of the report. I provided ChatGPT with my original text and requested suggestions for improvement, which I then reviewed and incorporated as appropriate.

1 Problem 1

1.1 Bayesian Decision Model

1.1.1 Model Description

In this homework, I implement a Bayesian Decision Model to classify data points based on their features. The model assumes that the data points are generated from a mixture of Gaussian distributions, each corresponding to a different class.

A multivariate Gaussian distribution is defined as follows:

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{1}$$

where μ is the mean vector, Σ is the covariance matrix, and d is the dimensionality of the data.

1.1.2 Parameter Estimation

To estimate the parameters of the Gaussian distributions for each class, the prior probabilities $P(C_i)$, the mean vectors μ_i, and the covariance matrices Σ_i for each class C_i are required.

The prior probabilities are estimated as follow, N_i is the number of samples in class C_i and N is the total number of samples.

$$P(C_i) = \frac{N_i}{N} \tag{2}$$

The mean vector for each class is estimated as:

$$\mu_i = \frac{1}{N_i} \sum_{x \in C_i} x \tag{3}$$

The covariance matrix for each class is estimated as:

$$\Sigma_i = \frac{1}{N_i} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T \tag{4}$$

1.1.3 Feature Selection

In this homework, I implement a forward feature selection algorithm to select the best subset of features that maximizes the model's performance. The algorithm starts with an empty set of features and iteratively adds the feature that results in the highest increase in performance, measured by the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve.

The algorithm for the forward feature selection algorithm is as follows:

Algorithm 1 Forward Feature Selection (FFS)

Require: Dataset $D = \{(X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}})\}$, feature dimension d

Ensure: Best feature mask best_mask, best AUC best_auc, best posterior best_prior

```
Initialize best_mask ← [0, 0, ..., 0]
Initialize best_auc ← 0.0
repeat
    local_best_auc ← −1
    local_best_mask ← best_mask
    for i = 1 to d do
        if best_mask[i] = 1 then
            continue
        end if
        local_mask ← best_mask; local_mask[i] ← 1
        Train model with local_mask to get AUC auc_i and posterior prior_i
        if auc_i > local_best_auc then
            local_best_auc ← auc_i
            local_best_mask ← local_mask
            local_best_prior ← prior_i
        end if
    end for
    if local_best_auc ≤ best_auc then
        break
    else
        best_auc ← local_best_auc
        best_mask ← local_best_mask
        best_prior ← local_best_prior
    end if
until no improvement in AUC
return best_auc, best_mask, best_prior
```

Following is the feature selected in each fold:

```
Fold 1: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol',
↪ 'NoseW', 'NasoFacialA']
Fold 2: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC',
↪ 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 3: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol',
↪ 'NoseW', 'NasoFacialA']
Fold 4: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH',
↪ 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 5: ['FaceWM', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH',
↪ 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 6: ['FaceWM', 'FaceWmax', 'Subn-Chin', 'MouthW', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
```

```
Fold 7: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 8: ['FaceWM', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 9: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 10: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 11: ['FaceWM', 'FaceWmax', 'Subn-Chin', '   LipD   ', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NasoFacialA']
Fold 12: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 13: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 14: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 15: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 16: ['FaceWM', 'FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 17: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 18: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 19: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 20: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 21: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 22: ['FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 23: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 24: ['FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 25: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 26: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 27: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 28: ['FaceWM', 'FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 29: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 30: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 31: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'MFaceH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 32: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 33: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 34: ['FaceWM', 'FaceWL', 'FaceWmax', '   LipD   ', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 35: ['FaceWM', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 36: ['FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'MFaceH', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
```

```
Fold 37: ['FaceWM', 'FaceWL', 'FaceWmax', 'Subn-Chin', 'MFaceH', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 38: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 39: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 40: ['FaceWM', 'FaceWL', 'FaceWmax', 'Subn-Chin', 'MFaceH', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 41: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'MFaceH', 'LipH', 'UVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 42: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 43: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 44: ['FaceWM', 'FaceWmax', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 45: ['FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 46: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 47: ['FaceWM', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 48: ['FaceWU', 'FaceWM', 'FaceWL', 'LFaceH', 'LVermilionH', 'LVermilionC', 'NoseW', 'NasoFacialA']
Fold 49: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 50: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 51: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 52: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 53: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 54: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 55: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 56: ['FaceWM', 'FaceWmax', 'Na-Chin', '    LipD    ', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NasoFacialA']
Fold 57: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 58: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'MFaceH', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 59: ['FaceWM', 'FaceWL', 'FaceWmax', 'Subn-Chin', 'MFaceH', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 60: ['FaceWM', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 61: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 62: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 63: ['FaceWM', 'FaceWmax', 'Subn-Chin', 'MouthW', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 64: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 65: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 66: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
```

```
Fold 67: ['FaceWU', 'FaceWM', 'FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'UVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NasoFacialA']
Fold 68: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 69: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 70: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 71: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 72: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'MFaceH', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 73: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 74: ['FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 75: ['FaceWM', 'FaceWmax', 'LFaceH', 'MouthW', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 76: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 77: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 78: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 79: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 80: ['FaceWM', 'FaceWL', 'FaceWmax', 'Subn-Chin', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 81: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 82: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 83: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 84: ['FaceWU', 'FaceWM', 'Na-Chin', 'LFaceH', 'MouthW', 'LipH', 'LVermilionH', 'LVermilionC', 'NoseW', 'NasoFacialA']
Fold 85: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 86: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 87: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 88: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'MFaceH', 'LFaceH', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 89: ['FaceWM', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 90: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 91: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 92: ['FaceWM', 'FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 93: ['FaceWM', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 94: ['FaceWM', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 95: ['FaceWU', 'FaceWM', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 96: ['FaceWM', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 97: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
```

```
Fold 98: ['FaceWM', 'FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 99: ['FaceWM', 'FaceWL', 'FaceWmax', 'Na-Chin', 'Subn-Chin', 'LFaceH', 'LipH', 'UVermilionH', 'LVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseW', 'NasoFacialA']
Fold 100: ['FaceWU', 'FaceWM', 'MFaceH', 'LFaceH', 'MouthW', 'LipH', 'UVermilionH', 'UVermilionC', 'LVermilionC', 'NoseVol', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 101: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 102: ['FaceWM', 'FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'MouthW', 'LVermilionH', 'LVermilionC', 'NoseVol', 'NoseSurfA', 'NoseW', 'NasoFacialA']
Fold 103: ['FaceWL', 'FaceWmax', 'MFaceH', 'LFaceH', 'LVermilionH', 'NoseVol', 'NoseW', 'NasoFacialA']
```

I'd tried AIC and BIC for feature selection, but they did not perform as well as AUC-based selection in this study (with AUC < 0.7 and selecting the same features repeatedly).

To discuss this phenomenon, take AIC as an example, AIC is defined as

$$\mathrm{AIC} = 2k - 2\ln(L), \tag{5}$$

where k is the number of parameters and L is the maximized likelihood of the model. Because the number of features (and thus parameters) in the dataset is relatively small, the penalty term $2k$ exerts only a weak influence on the criterion. As a result, AIC becomes dominated by the likelihood term and fails to effectively penalize model complexity, leading to the selection of suboptimal feature subsets. In contrast, using AUC directly as the objective function better reflects the model's discriminative ability and yields feature combinations that improve classification performance.

1.1.4 Result for Bayesian Classifier with Forward Feature Selection

Metrics with Feature Selection: The AUC of loocv with feature selection is 0.9233, Fig. 1 shows the ROC curve.

For this binary classification problem, we denote the two classes as C_0 and C_1, if posterior probability $P(C_1|x) > P(C_0|x)$, we classify the data point x as class C_1, otherwise class C_0. Thus, I choose the threshold as 0.5 to evalute the model, the performance metrics are as follows:

- Accuracy: 0.8350 (86/103)
- Sensitivity: 0.8537 (35/41)
- Specificity: 0.8226 (51/62)

Features Selected Frequency The frequency of selected features across all folds is summarized in Table 1.

The top 2 features selected are NasoFacialA, LVermilionH, and NoseW with LVermilionH and NoseW tied at 100 times, so I choose NasoFacialA and LVermilionH(alphabetical) to plot the decision boundary in Fig. 2.

Table 1: Frequency of Selected Features Across All Folds

Feature	Frequency	Feature	Frequency
NasoFacialA	103	FaceWL	55
LVermilionH	100	LipH	37
NoseW	100	UVermilionC	35
FaceWmax	99	UVermilionH	35
LFaceH	98	MFaceH	34
LVermilionC	95	Subn-Chin	33
NoseVol	92	NoseSurfA	15
FaceWM	91	FaceWU	5
Na-Chin	71	LipD	3
MouthW	62		

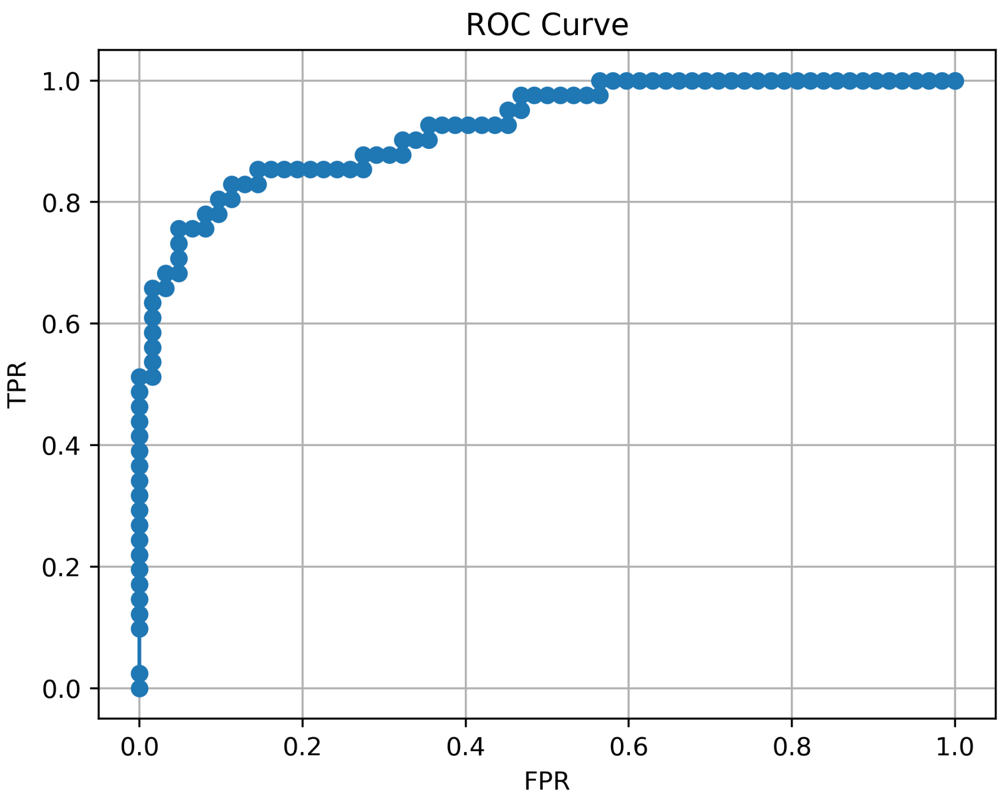

Figure 1: ROC curve with feature selection

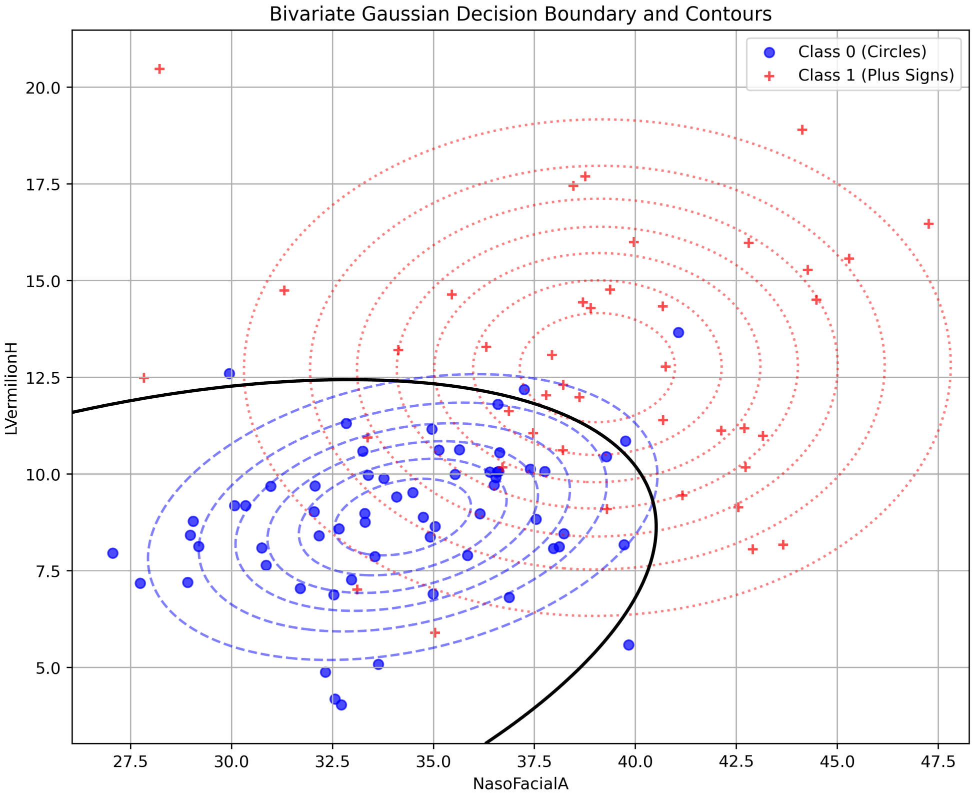

Figure 2: Decision boundary with feature selection

1.1.5 Others

Code Structure. In the `code/Bayesian/` directory, `origin.py` serves as the baseline implementation of the Bayesian classifier without complete outputs and efficiency optimizations. `numpy_accr.py` is the complete version that includes all required outputs and NumPy-based vectorization described in this report, while `multi_core_accr.py` integrates multiprocessing optimizations for LOOCV to further enhance computational efficiency.

Parallel processing may not be supported in all environments (the experiments were conducted on Ubuntu 24.04.3 LTS with Linux kernel 6.14.0-33-generic). Therefore, a NumPy-optimized version is also provided for compatibility. If the environment supports multiprocessing, it is recommended to execute the multi-core accelerated version, which produces results within only a few seconds.

Efficiency Optimization. It is well known that Python is generally less efficient than compiled languages such as C or C++. Therefore, I employed extensive NumPy vectorized operations to accelerate computation. Furthermore, since each fold in the LOOCV process is independent, I used the multiprocessing library to parallelize the computation across multiple CPU cores, which significantly reduced the overall runtime.

1.2 Random Forest Classifier

1.2.1 Model Description

This homework assigns the implementation of Bayesian Classifier, but Random Forest is also a widely used classification algorithm.

I implemented a Random Forest Classifier using the `RandomForestClassifier` from the `sklearn.ensemble` module. Then, I tried Out-of-Bag (OOB) estimation, 10-fold cross-validation, and Leave-One-Out Cross-Validation (LOOCV) to evaluate the model performance and compared the results.

1.2.2 Result for Random Forest Classifier

The result ROC curve is shown in Fig. 3.

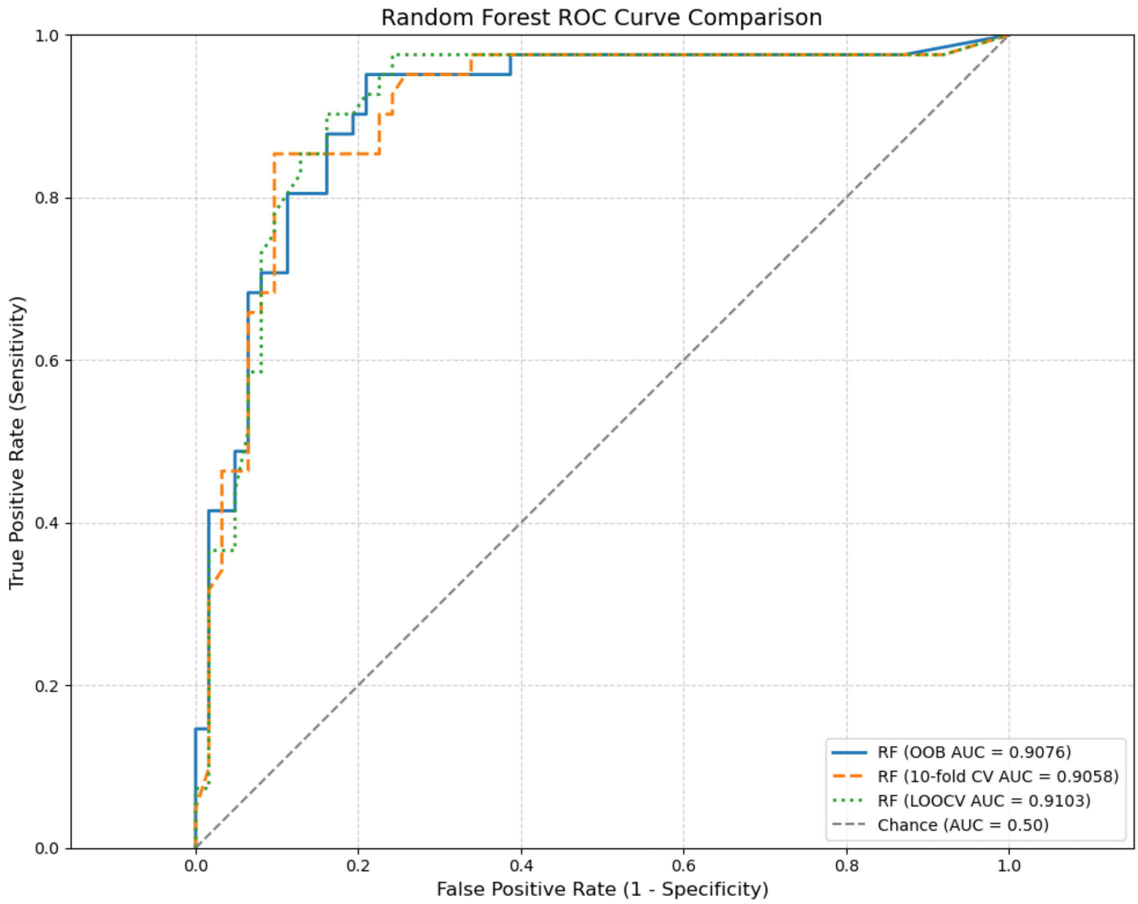

Figure 3: ROC curve of Random Forest Classifier with LOOCV

These methods perform similarly well, with OOB AUC at 0.9076, 10-fold CV AUC at 0.9058, and LOOCV AUC at 0.9103.

1.3 Discussion

Following is the performance comparison of different evaluation methods:

Table 2: Performance Comparison of Different Models and Evaluation Methods

Method	**AUC**	**Accuracy**	**Sensitivity**	**Specificity**
Bayesian (LOOCV, FFS)	0.9233	0.8350	0.8537	0.8226
RF (OOB)	0.9076	0.8252	0.7317	0.8871
RF (10-fold CV)	0.9058	0.8738	0.8293	0.9032
RF (LOOCV)	0.9103	0.8544	0.8049	0.8871

Note that FFS means forward feature selection, RF means Random Forest, CV means cross-validation, and OOB means out-of-bag estimation.

All four evaluation methods achieve AUC values greater than 0.9, demonstrating strong discriminative capability. The Bayesian classifier with forward feature selection attains the highest AUC of 0.9233, while the Random Forest with 10-fold cross-validation yields the highest accuracy of 0.8738. This indicates that both feature selection and ensemble learning effectively enhance model performance under limited data conditions.

However, the relatively small dataset size remains a critical constraint. Although the four methods perform similarly well in this study, the observed differences in AUC and accuracy may become more evident with a larger sample size. In particular, the OOB estimate tends to slightly underestimate sensitivity due to the stochastic nature of bootstrap sampling, whereas k-fold CV provides a more stable performance estimate at the cost of additional computation.

Overall, both the Bayesian classifier with feature selection and the Random Forest classifier demonstrate promising performance for this binary classification task. The Bayesian approach shows a slight advantage in AUC, but its robustness cannot be confirmed solely from this small dataset. Further validation on larger and more diverse datasets is necessary to assess their true generalization ability and stability.

2 Problem 2

1. Problem Setup

Assume we have data points

$$x^t \mid \theta \sim \mathcal{N}(\theta, \sigma^2), \quad t = 1, 2, \ldots, N,$$

and a Gaussian prior on the unknown mean parameter:

$$\theta \sim \mathcal{N}(\mu_0, \sigma_0^2).$$

Our goal is to find the posterior distribution $p(\theta \mid X)$, where $X = \{x^1, \ldots, x^N\}$.

2. Bayes' Rule

Starting from

$$p(\theta \mid X)p(X) = p(X \mid \theta)p(\theta),$$

we obtain

$$p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{p(X)} \propto p(X \mid \theta)p(\theta),$$

since $p(X)$ does not depend on θ.

3. Likelihood as a Gaussian in θ

Under the i.i.d. assumption,

$$p(X \mid \theta) = \prod_{t=1}^{N} p(x^t \mid \theta) = (2\pi\sigma^2)^{-N/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{N}(x^t - \theta)^2\right).$$

Let $m = \frac{1}{N}\sum_{t=1}^{N} x^t$ be the sample mean. Then,

$$\sum_{t=1}^{N}(x^t - \theta)^2 = \sum_{t=1}^{N}(x^t - m)^2 + N(\theta - m)^2.$$

Ignoring terms independent of θ,

$$p(X \mid \theta) \propto \exp\left(-\frac{N}{2\sigma^2}(\theta - m)^2\right),$$

which is a Gaussian density in θ:

$$p(X \mid \theta) \equiv \mathcal{N}\left(\theta; m, \frac{\sigma^2}{N}\right).$$

4. Multiplying Two Gaussian Distributions

The prior is

$$p(\theta) = \mathcal{N}(\theta; \mu_0, \sigma_0^2).$$

Hence, the unnormalized posterior is

$$p(\theta \mid X) \propto p(X \mid \theta)p(\theta) = \mathcal{N}\left(\theta; m, \frac{\sigma^2}{N}\right) \times \mathcal{N}\left(\theta; \mu_0, \sigma_0^2\right).$$

5. Product of Two Gaussians

For any two Gaussian densities

$$\mathcal{N}(\theta; \mu_1, \sigma_1^2) \quad \text{and} \quad \mathcal{N}(\theta; \mu_2, \sigma_2^2),$$

their product is proportional to another Gaussian:

$$\mathcal{N}(\theta; \mu_1, \sigma_1^2)\,\mathcal{N}(\theta; \mu_2, \sigma_2^2) \propto \mathcal{N}(\theta; \mu', \sigma'^2),$$

where

$$\sigma'^2 = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}, \quad \mu' = \sigma'^2\left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right).$$

6. Applying the Gaussian Product Formula

By identifying parameters:

$$(\mu_1, \sigma_1^2) = (m, \tfrac{\sigma^2}{N}), \quad (\mu_2, \sigma_2^2) = (\mu_0, \sigma_0^2),$$

we obtain the posterior parameters:

$$\sigma_n^2 = \frac{1}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}, \quad \mu_n = \sigma_n^2\left(\frac{Nm}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}\right).$$

Thus, the posterior distribution is:

$$p(\theta \mid X) = \mathcal{N}(\theta; \mu_n, \sigma_n^2).$$

7. Posterior Mean and Interpretation

The posterior mean (the Bayesian estimator) is:

$$E[\theta \mid X] = \mu_n = \frac{\frac{N}{\sigma^2}}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}\, m + \frac{\frac{1}{\sigma_0^2}}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}\, \mu_0.$$

This is a precision-weighted average:

$$E[\theta \mid X] = w_{\text{data}}\, m + w_{\text{prior}}\, \mu_0,$$

where

$$w_{\text{data}} = \frac{N/\sigma^2}{N/\sigma^2 + 1/\sigma_0^2}, \quad w_{\text{prior}} = \frac{1/\sigma_0^2}{N/\sigma^2 + 1/\sigma_0^2}.$$

8. Limiting Cases

- If the data are abundant ($N \to \infty$), $E[\theta \mid X] \to m$: posterior dominated by data.
- If the prior is very confident ($\sigma_0^2 \to 0$), $E[\theta \mid X] \to \mu_0$: posterior dominated by prior.
- If the prior is uninformative ($\sigma_0^2 \to \infty$), $E[\theta \mid X] \to m$: reduces to the MLE.

9. Final Result

$$\boxed{p(\theta \mid X) = \mathcal{N}\big(\mu_n,\ \sigma_n^2\big), \quad \sigma_n^2 = \frac{1}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}, \quad \mu_n = \sigma_n^2\left(\frac{Nm}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}\right).}$$