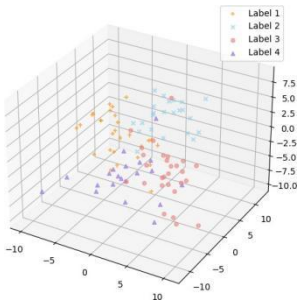
Question 1

To ensure that the probability of error falls between 10% and 20% when classifying the generated data with the MAP method, I selected the following parameters:

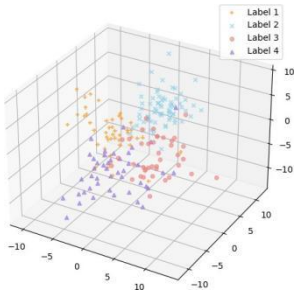
Label	P	Mean	Covariance
1	1/4	$\begin{bmatrix}-4\\3\\-2\end{bmatrix}$	$\begin{bmatrix}-10 & 2 & -2\\2 & 4 & 0\\-2 & 0 & 10\end{bmatrix}$
2	1/4	$\begin{bmatrix}2\\4\\3\end{bmatrix}$	$\begin{bmatrix}4 & 1 & 0\\1 & 10 & 0\\0 & 0 & 4\end{bmatrix}$
3	1/4	$\begin{bmatrix}4\\-2\\2\end{bmatrix}$	$\begin{bmatrix}10 & -1 & 0\\-1 & 4 & 0\\0 & 0 & 10\end{bmatrix}$
4	1/4	$\begin{bmatrix}0\\-4\\-3\end{bmatrix}$	$\begin{bmatrix}10 & 0 & 0\\0 & 10 & 4\\0 & 4 & 10\end{bmatrix}$

Below are the data distributions for the generated datasets with sample sizes of N=100, 200, 500, 1000, 2000, 5000, and 100000.

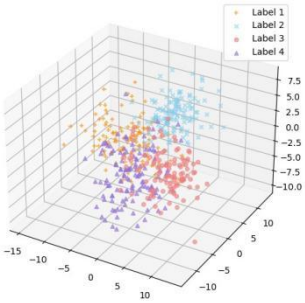
N=100



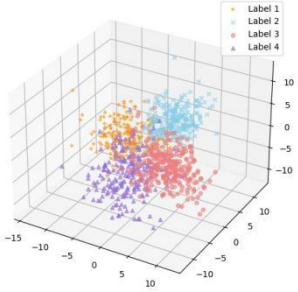
N=200



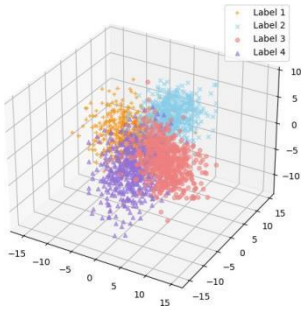
N=500



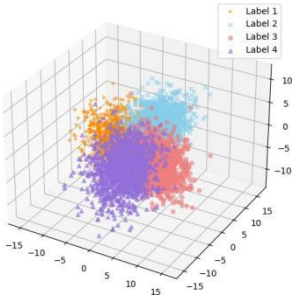
N=1000



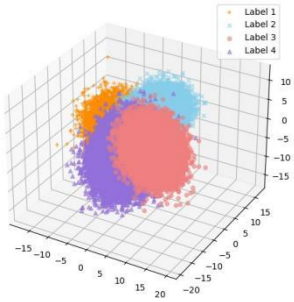
N=2000



N=5000



N=100000



We applied the minimum-probability-of-error classification rule to estimate the optimal probability of error on a test dataset comprising 100000 samples, yielding the following result P(error) = 0.127.

To carry out the 2-layer MLP as specified in the problem statement, we employed the sklearn.neural_network.MLPClassifier. This classifier comprises a hidden layer with P perceptrons and utilizes the smooth-ramp style activation function 'tanh'. To evaluate the model's performance, we utilized the cross-entropy loss function, and for multi-classification, the classifier implemented the softmax function as the output function.

We conducted 10-fold cross-validation on each training dataset to determine the optimal number of perceptrons that could minimize the P(error). The relationship between the number of perceptrons and the corresponding calculated P(error) is depicted in Figure 1.1. Based on the obtained results, the ideal number of perceptrons for each training dataset is presented in Table 1.1.

Figure 1.1

Table 1.1

Training Dataset	N=100	N=200	N=500	N=1000	N=2000	N=5000
Number of Perceptrons	10	19	22	6	9	30

Using the optimal number of perceptrons determined in the previous step, we trained multiple MLP classifiers on each training dataset. We selected the classifiers that provided the best classification accuracy on their respective training data. Subsequently, we employed these MLP classifiers to classify the test dataset. The estimated probabilities of error for each classifier are presented in Figure 1.2. The optimal P(error) is 0.128, and the corresponding P(error) on the Test Dataset is presented in Table 1.2.It is evident that the MLP classifier trained on a larger training dataset could attain both a higher classification accuracy and a lower P(error) on the test dataset.

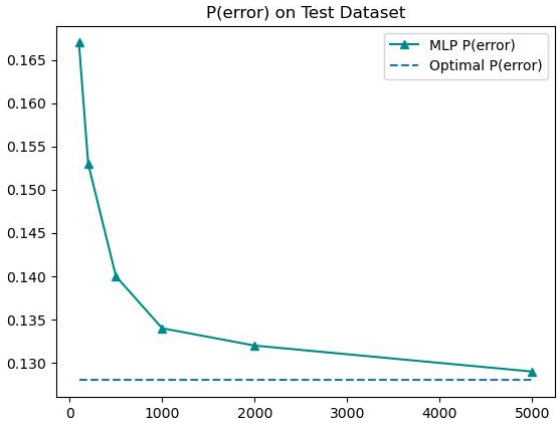


Figure 1.2

Table 1.2

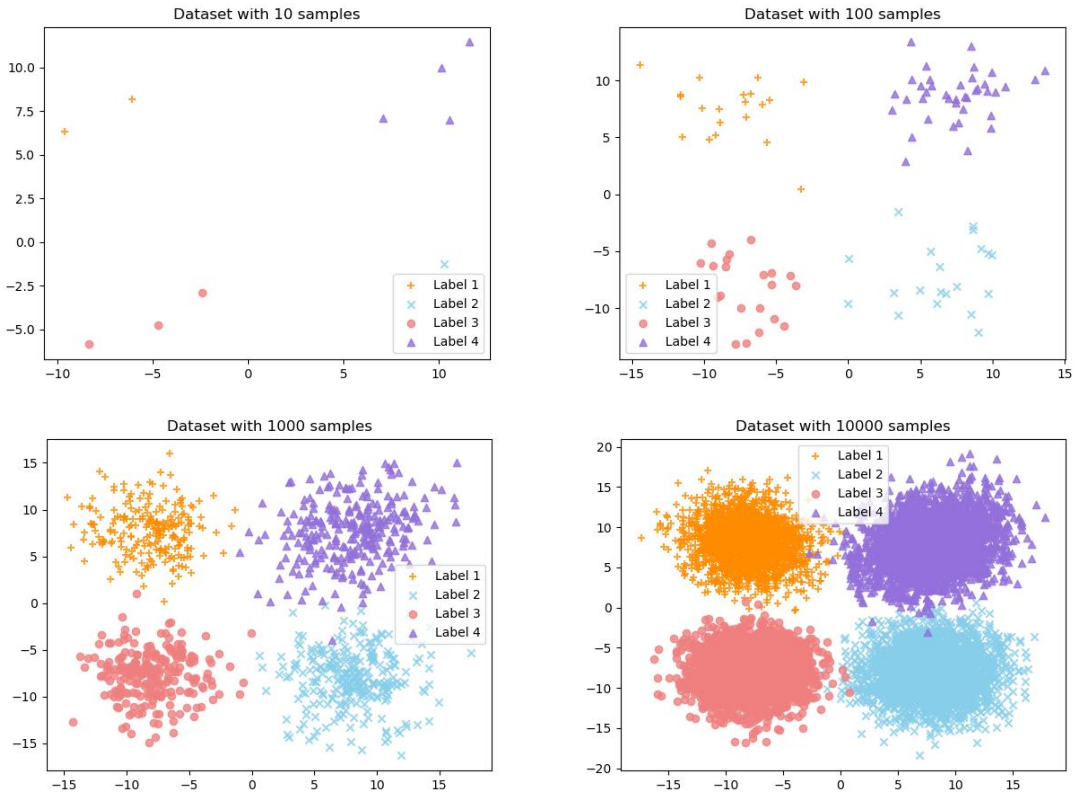
Training Dataset	N=100	N=200	N=500	N=1000	N=2000	N=5000
P(error) on Test Dataset	0.168	0.154	0.140	0.134	0.133	0.128

Question 2

To generate the data, we established the parameters for the four components of our GMM as follows:

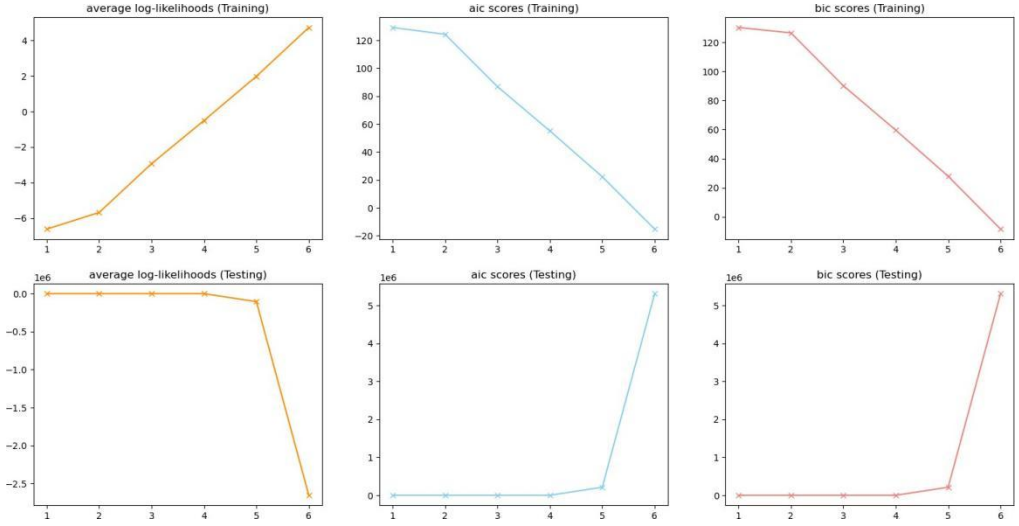
Label	P	Mean	Covariance
1	0.22	$\begin{bmatrix}-7\\7\end{bmatrix}$	$\begin{bmatrix}8&-1\\-1&8\end{bmatrix}$
2	0.28	$\begin{bmatrix}7\\-7\end{bmatrix}$	$\begin{bmatrix}7&0\\0&7\end{bmatrix}$
3	0.24	$\begin{bmatrix}-7\\-7\end{bmatrix}$	$\begin{bmatrix}6&0\\0&6\end{bmatrix}$
4	0.26	$\begin{bmatrix}7\\7\end{bmatrix}$	$\begin{bmatrix}9&2\\2&9\end{bmatrix}$

The data distributions for the generated datasets with sample sizes N=10, 100, 1000, and 10000 are depicted below:

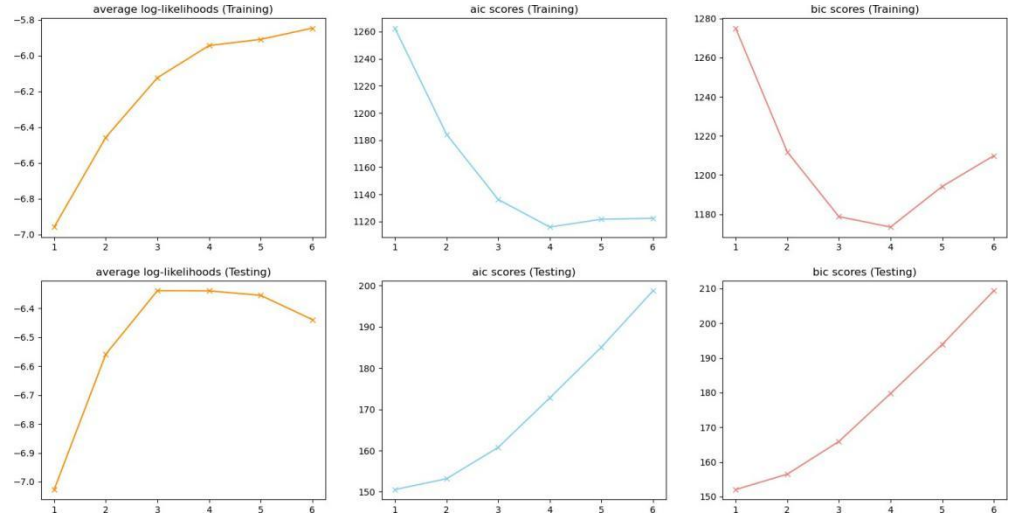


We evaluated the generated datasets one by one within the framework of 10-fold cross-validation. We considered models with 1, 2, 3, 4, 5, and 6 components and assessed their performance based on the average log-likelihoods, AIC scores, and BIC scores. Higher values of average log-likelihoods indicate better performance, whereas lower values of AIC and BIC scores indicate better performance. The results are presented below:

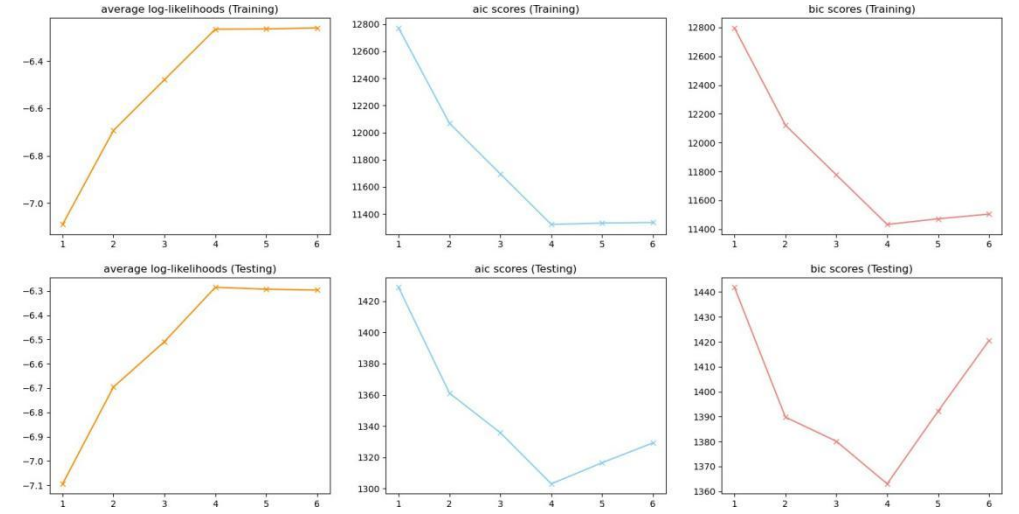
N=10



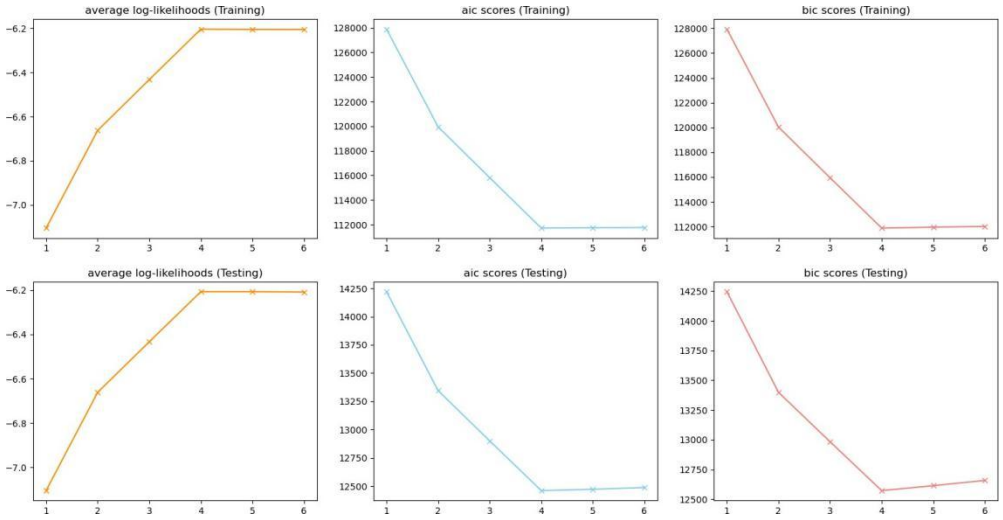
N=100



N=1000



N=10000



To further investigate this, we conducted 100 experiments where we determined the number of components for each dataset by selecting the maximum log-likelihood, minimum AIC score, and minimum BIC score. From these experiments, we calculated the selection rates of the six Gaussian Mixture Model orders on our datasets, resulting in the following findings:

N=10

number of components	likelihood (train)	likelihood (test)	aic (train)	aic (test)	bic (train)	bic (test)
1	0	1.0	0	1.0	0	1.0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	1.0	0	1.0	0	1.0	0

N=100

number of components	likelihood (train)	likelihood (test)	aic (train)	aic (test)	bic (train)	bic (test)
1	0	0	0	1.0	0	1.0

2	0	0	0	0	0	0
3	0	0.05	0	0	0	0
4	0	0.81	0.93	0	1.0	0
5	0	0.14	0.06	0	0	0
6	1.0	0.01	0.03	0	0	0

N=1000

number of components	likelihood (train)	likelihood (test)	aic (train)	aic (test)	bic (train)	bic (test)
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0.83	0.99	1.0	0.99	1.0
5	0.01	0.12	0.01	0	0.01	0
6	0.99	0.02	0	0	0	0

N=10000

number of components	likelihood (train)	likelihood (test)	aic (train)	aic (test)	bic (train)	bic (test)
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	1.0	1.0	1.0	1.0	1.0	1.0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Based on the findings presented above, it can be observed that the accuracy of the three methods for identifying the number of components increases as the size of the dataset grows. When the

sample size is small, AIC and BIC scores are more accurate in making decisions than log-likelihoods. However, when N=10000, all three methods are capable of accurately identifying that the Gaussian Mixture Models consist of 4 components.

Append

The code of hw3 is as follow link:
https://github.com/JonnyFan/ML5644/tree/main/hw3