HW3

William Fox

11/15/24

Question 1)

EECE5644 Fall 2024 – Assignment 3

Please submit in Canvas a single PDF file showing all results, and include code as appendix or as an external repository link in the PDF file. Cite all appropriate sources you benefit from. Directly exchanging writing/code/results between classmates or from former students of this course are not acceptable, but you may talk to each other in classes, office periods, benefit from each others' questions and answers for those.

Question 1 (60%)

In this exercise, you will train many multilayer perceptrons (MLP) to approximate the class label posteriors, using maximum likelihood parameter estimation (equivalently, with minimum average cross-entropy loss) to train the MLP. Then, you will use the trained models to approximate a MAP classification rule in an attempt to achieve minimum probability of error (i.e. to minimize expected loss with 0-1 loss assignments to correct-incorrect decisions).

Data Distribution: For $C = 4$ classes with uniform priors, specify Gaussian class-conditional pdfs for a 3-dimensional real-valued random vector $\mathbf{x}$ (pick your own mean vectors and covariance matrices for each class). Try to adjust the parameters of the data distribution so that the MAP classifier that uses the true data pdf achieves between $10\% - 20\%$ probability of error.

MLP Structure: Use a 2-layer MLP (one hidden layer of perceptrons) that has P perceptrons in the first (hidden) layer with smooth-ramp style activation functions (e.g., ISRU, Smooth-ReLU, ELU, etc). At the second/output layer use a softmax function to ensure all outputs are positive and add up to 1. The best number of perceptrons for your custom problem will be selected using cross-validation.

Generate Data: Using your specified data distribution, generate multiple datasets: Training datasets with $100, 500, 1000, 5000, 10000$ samples and a test dataset with 100000 samples. You will use the test dataset only for performance evaluation.

Theoretically Optimal Classifier: Using the knowledge of your true data pdf, construct the minimum-probability-of-error classification rule, apply it on the test dataset, and empirically estimate the probability of error for this theoretically optimal classifier. This provides the aspirational performance level for the MLP classfier.

Model Order Selection: For each of the training sets with different number of samples, perform 10-fold cross-validation, using minimum classification error probability as the objective function, to select the best number of perceptrons (that is justified by available training data).

Model Training: For each training set, having identified the best number of perceptrons using cross-validation, using maximum likelihood parameter estimation (minimum cross-entropy loss) train an MLP using each training set with as many perceptrons as you have identified as optimal for that training set. These are your final trained MLP models for class posteriors (possibly each with different number of perceptrons and different weights). Make sure to mitigate the chances of getting stuck at a local optimum by randomly reinitializing each MLP training routine multiple times and getting the highest training-data log-likelihood solution you encounter.

Performance Assessment: Using each trained MLP as a model for class posteriors, and using the MAP decision rule (aiming to minimize the probability of error) classify the samples in the test set and for each trained MLP empirically estimate the probability of error.

Report Process and Results: Describe your process of developing the solution; numerically and visually report the test set empirical probability of error estimates for the theoretically optimal and multiple trained MLP classifiers. For instance show a plot of the empirically estimated test P(error) for each trained MLP versus number of training samples used in optimizing it (with semilog-x axis), as well as a horizontal line that runs across the plot indicating the empirically estimated test P(error) for the theoretically optimal classifier.

Note: You may use software packages for all aspects of your implementation. Make sure you use tools correctly. Explain in your report how you ensured the software tools do exactly what you need them to do.

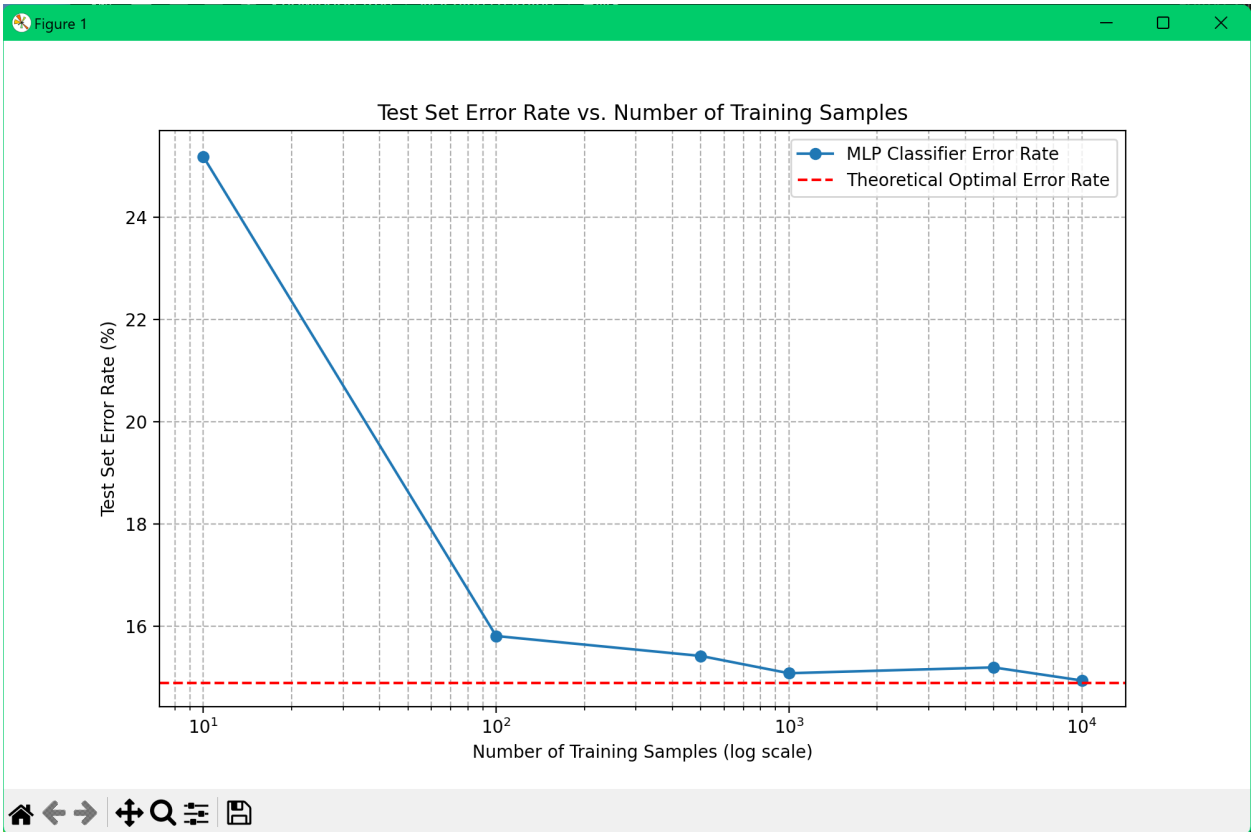
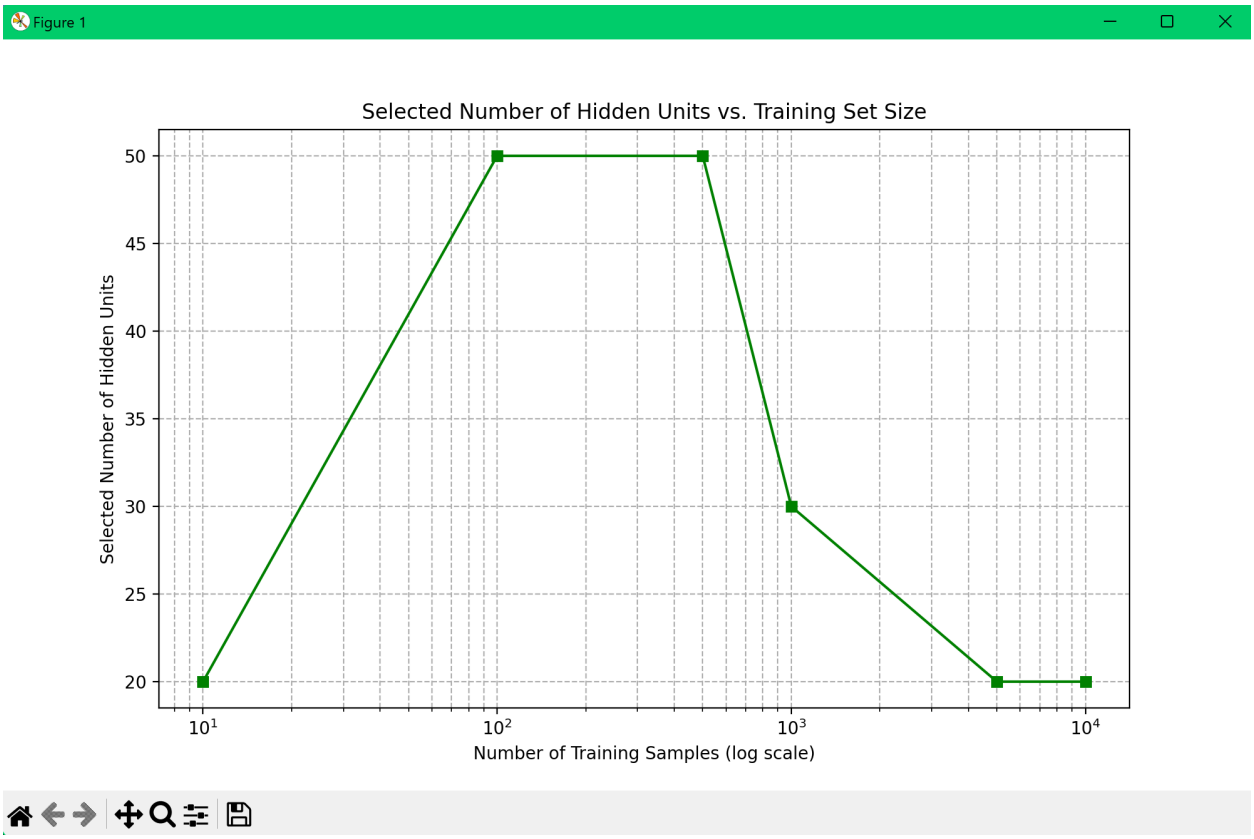


Figure 1
Test Set Error Rate vs. Number of Training Samples
MLP Classifier Error Rate
Theoretical Optimal Error Rate
Test Set Error Rate (%)
24
22
20
18
16
10^1
10^2
10^3
10^4
Number of Training Samples (log scale)



OUTPUT:

Generating test dataset...
Evaluating Theoretical Optimal Classifier...

Theoretical Optimal Classifier Error Rate: 14.90%

Training with **10 samples**...
Selecting best number of hidden units via cross-validation...
Adjusted number of cross-validation folds to 2 based on class distribution.
Hidden Units: 10, Cross-Validation Accuracy: 0.3000
Hidden Units: 20, Cross-Validation Accuracy: 0.4000
Hidden Units: 30, Cross-Validation Accuracy: 0.3000
Hidden Units: 40, Cross-Validation Accuracy: 0.4000
Hidden Units: 50, Cross-Validation Accuracy: 0.3000
Selected Hidden Units: 20
Training final MLP model with selected hidden units...
Best Training Accuracy: 90.00%
Evaluating model on test set...

Test Set Error Rate: 25.19%

Training with **100 samples**...
Selecting best number of hidden units via cross-validation...
Hidden Units: 10, Cross-Validation Accuracy: 0.7100
Hidden Units: 20, Cross-Validation Accuracy: 0.7800
Hidden Units: 30, Cross-Validation Accuracy: 0.8400
Hidden Units: 40, Cross-Validation Accuracy: 0.8300
Hidden Units: 50, Cross-Validation Accuracy: 0.8600
Selected Hidden Units: 50
Training final MLP model with selected hidden units...
Best Training Accuracy: 91.00%
Evaluating model on test set...

Test Set Error Rate: 15.81%

Training with **500 samples**...
Selecting best number of hidden units via cross-validation...
Hidden Units: 10, Cross-Validation Accuracy: 0.8280
Hidden Units: 20, Cross-Validation Accuracy: 0.8440
Hidden Units: 30, Cross-Validation Accuracy: 0.8380
Hidden Units: 40, Cross-Validation Accuracy: 0.8420
Hidden Units: 50, Cross-Validation Accuracy: 0.8500
Selected Hidden Units: 50
Training final MLP model with selected hidden units...
Best Training Accuracy: 85.60%
Evaluating model on test set...

Test Set Error Rate: 15.42%

Training with **1000 samples**...
Selecting best number of hidden units via cross-validation...
Hidden Units: 10, Cross-Validation Accuracy: 0.8600
Hidden Units: 20, Cross-Validation Accuracy: 0.8570
Hidden Units: 30, Cross-Validation Accuracy: 0.8650
Hidden Units: 40, Cross-Validation Accuracy: 0.8610
Hidden Units: 50, Cross-Validation Accuracy: 0.8600

Selected Hidden Units: 30
Training final MLP model with selected hidden units...
Best Training Accuracy: 87.30%
Evaluating model on test set...

Test Set Error Rate: 15.09%

Training with **5000 samples**...
Selecting best number of hidden units via cross-validation...
Hidden Units: 10, Cross-Validation Accuracy: 0.8424
Hidden Units: 20, Cross-Validation Accuracy: 0.8446
Hidden Units: 30, Cross-Validation Accuracy: 0.8440
Hidden Units: 40, Cross-Validation Accuracy: 0.8442
Hidden Units: 50, Cross-Validation Accuracy: 0.8434
Selected Hidden Units: 20
Training final MLP model with selected hidden units...
Best Training Accuracy: 84.94%
Evaluating model on test set...

Test Set Error Rate: 15.20%

Training with **10000 samples**...
Selecting best number of hidden units via cross-validation...
Hidden Units: 10, Cross-Validation Accuracy: 0.8500
Hidden Units: 20, Cross-Validation Accuracy: 0.8505
Hidden Units: 30, Cross-Validation Accuracy: 0.8502
Hidden Units: 40, Cross-Validation Accuracy: 0.8491
Hidden Units: 50, Cross-Validation Accuracy: 0.8492
Selected Hidden Units: 20
Training final MLP model with selected hidden units...
Best Training Accuracy: 85.21%
Evaluating model on test set...

Test Set Error Rate: 14.94%

Re-evaluating Theoretical Optimal Classifier with Adjusted Data Distribution...
Theoretical Optimal Classifier Error Rate: 14.90%

Step by step breakdown:

1. Set up data distributions (later modified)
    a. initially had a theoretical optimal classifier error rate at 32.70% so covariance needed to be modified to get down to 14.90%
    b. wanted to make sure the separation was just right, so set covariances until the true valued pdf was between 10-20%
2. generated test data set of 100,000 using np.multivariate_normal()
3. Evaluating theoretical classifier
    a. finds the true data pdf by:
    b. calculate class posteriors by calculating probability density for each class's distribution
    c. assign each sample to class with highest posterior P
    d. compute proportion of misclassified samples
4. Defining MLP architecture
    a. Input layer: takes in feature vectors
    b. Hidden Layer: we configure a number of neurons with selectable activation functions (set to elu)
    c. Output Layer: Outputs class probabilities using softmax
5. Using cross validation for selecting optimal number of hidden units
    a. apply standard 10-fold cross validation
    b. adjust number of folds based on minimum number of samples per class
    c. if fewer than 2 folds are found, use all samples for validation
    d. iterate over range of hidden layer sizes (10, 20, 30, 40, 50)
        i. for each, train the MLP on the training subset and evaluate on the validation
    e. calculate the mean validation accuracy for each hidden unit size

    f. select the best hidden unit size (highest accuracy)
6. train the MLP model using the whole training data set with the optimal numbers of hidden units
    a. conduct multiple trials with different initializaitons so that we have a small chance of ending up at local optimums
    b. choose the model with highest training accuracy across trials
7. assess model performance on test set
    a. use same scalar as for test data to transform test features
    b. use MLP model that is now trained to predict class probabilities and assign classes
    c. determine proportion of misclassified test samples
8. execute and visualize results

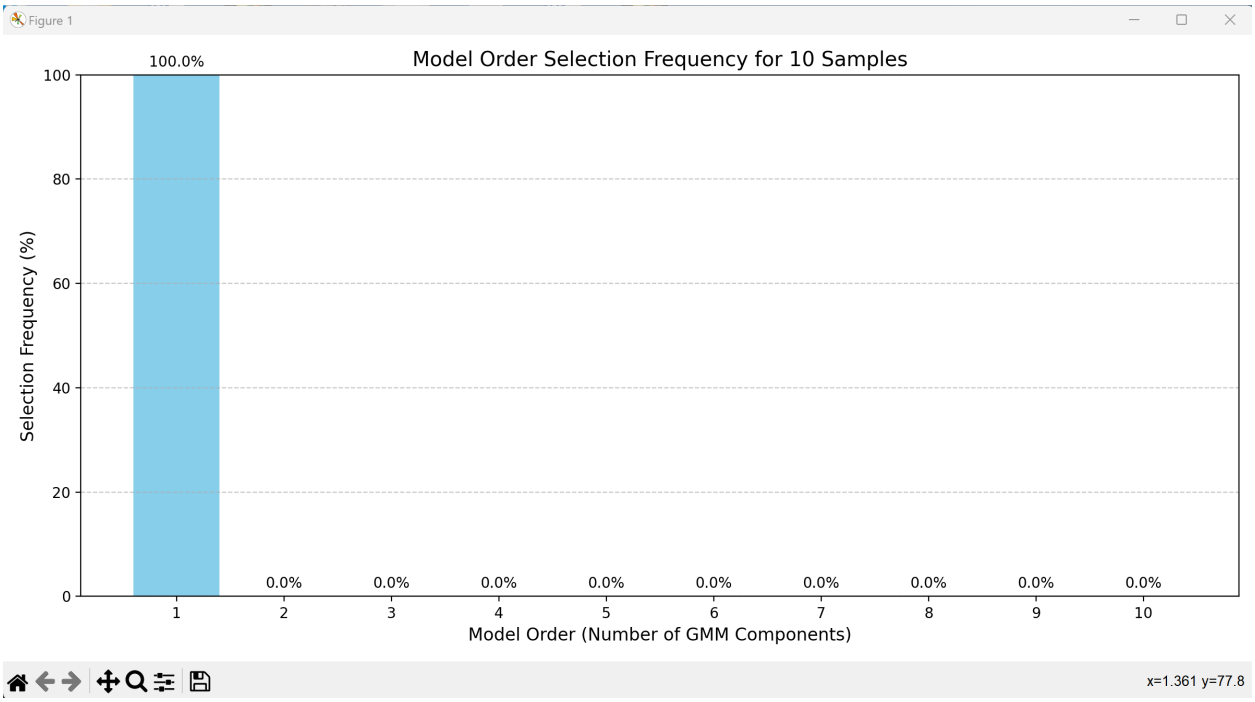
Test set error rate ended up at 14.94% compared to theoretical optimal classifier of 14.90%

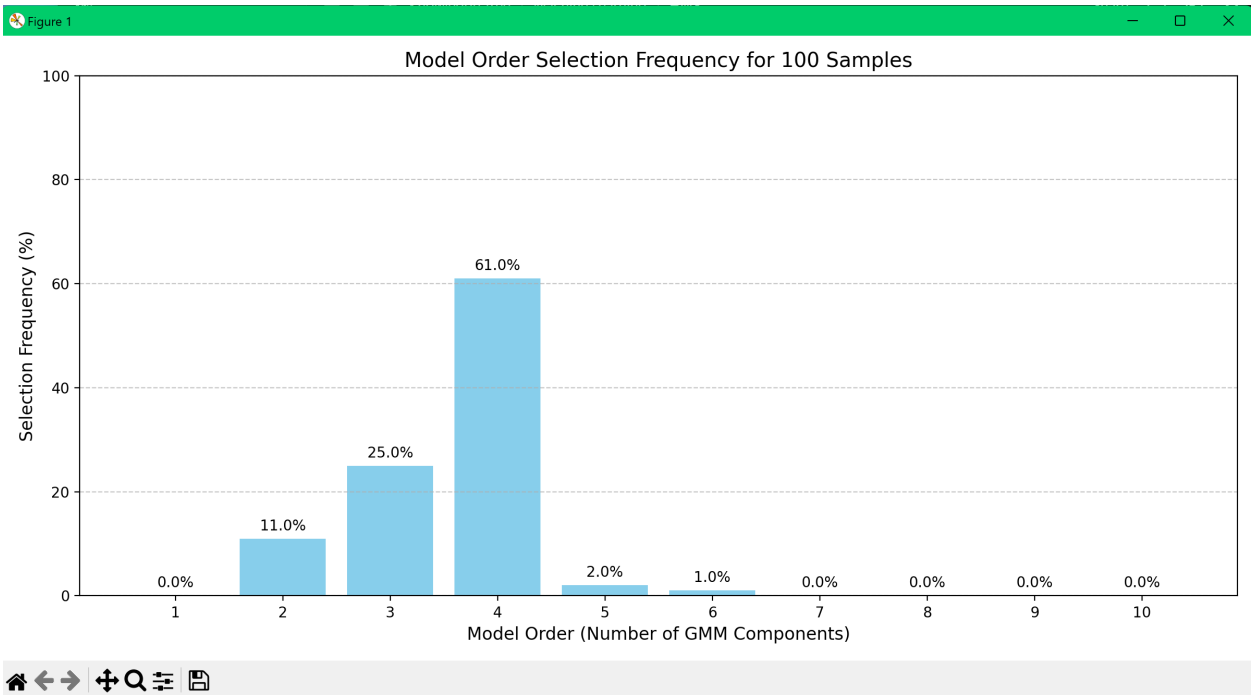
Question 2)

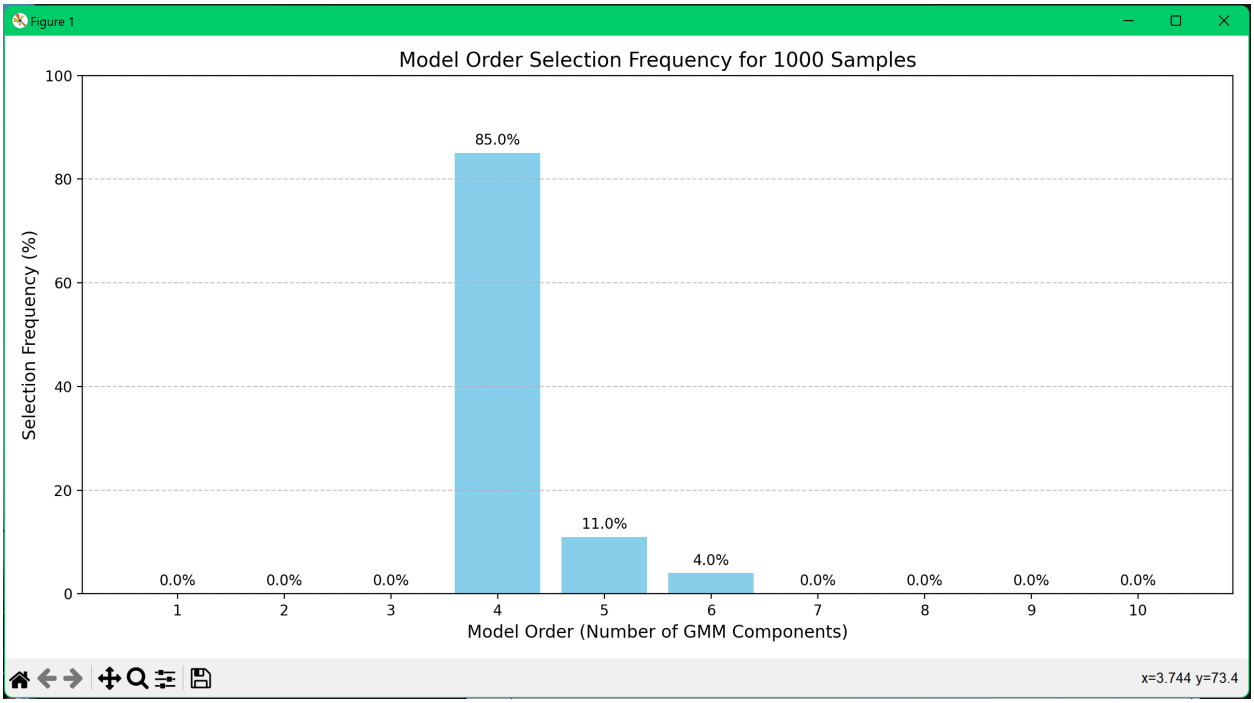
Question 2 (40%)

Conduct the following model order selection exercise using 10-fold cross-validation procedure and report your procedure and results in a comprehensive, convincing, and rigorous fashion:

1. Select a Gaussian Mixture Model as the true probability density function for 2-dimensional real-valued data synthesis. This GMM will have 4 components with different mean vectors, different covariance matrices, and different probability for each Gaussian to be selected as the generator for each sample. Specify the true GMM that generates data in a way that has two of the Gaussian components overlap significantly (e.g. set the distance between mean vectors comparable to the sum of their average covariance matrix eigenvalues).
2. Generate multiple data sets with independent identically distributed samples using this true GMM; these datasets will have respectively 10, 100, 1000 samples.
3. For each data set, using maximum likelihood parameter estimation principle (e.g. with the EM algorithm), within the framework of K-fold (e.g., 10-fold) cross-validation, evaluate GMMs with different model orders; specifically evaluate candidate GMMs with $1, 2, \ldots, 10$ Gaussian components. Note that both model parameter estimation and validation performance measures to be used is log-likelihood of data.
4. Repeat the experiment multiple times (e.g., at least 100 times) and report your results, indicating the rate at which each of the the six GMM orders get selected for each of the datasets you produced. Develop a good way to describe and summarize your experiment results in the form of tables/figures.







The model order is the number of hidden units

OUTPUT:

Dataset Size: 10 samples
Model Order Selection Frequency (%)

1	100.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0
10	0.0

Dataset Size: 100 samples

Model Order	Selection Frequency (%)
1	0.0
2	11.0
3	25.0
4	61.0
5	2.0
6	1.0
7	0.0
8	0.0
9	0.0
10	0.0

Dataset Size: 1000 samples

Model Order	Selection Frequency (%)
1	0.0
2	0.0
3	0.0
4	85.0
5	11.0
6	4.0
7	0.0
8	0.0
9	0.0
10	0.0

Interpretation and Analysis of output:

For smaller datasets, model order 1 is selected 100% of the time. This is expected because with limited data, simpler models are preferred to prevent overfitting. For medium datasets around sample size 100, model 4 is chosen the most frequently at 61%, followed by model order 2 at 11%. This data suggests that as the dataset size increases, the model complexity increases appropriately to capture more nuanced patterns. It does this without overfitting the data. For larger datasets around the size of 1000, model order 4 is again the most commonly selected. This indicates that the GMM has enough data to accurately estimate the parameters and distinguish between all four components. There is slight overestimation (shown by the small percentage of cases where the model order was estimated to be greater than 4) that potentially suggests slight overfitting of the cross-validation approach.