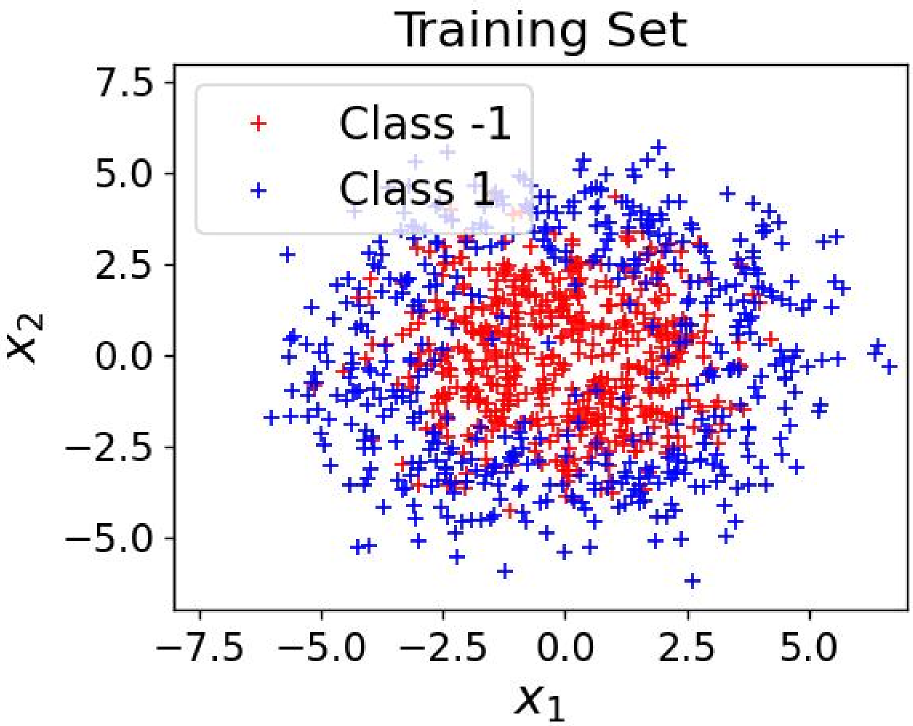
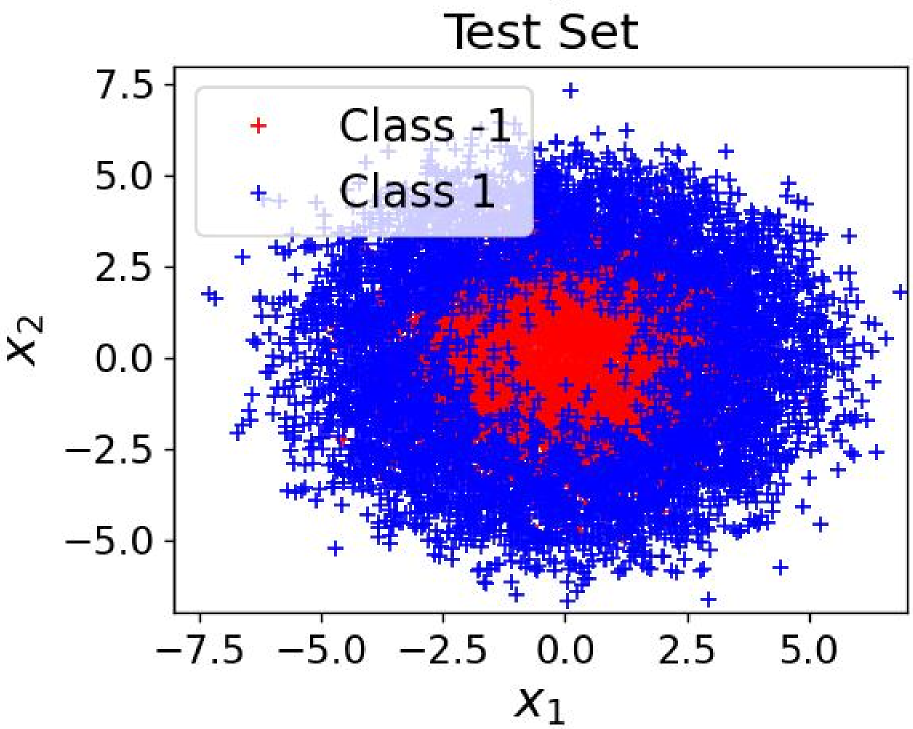
Question 1

Generate a training dataset of 1000 independent and identically distributed (iid) samples and a testing dataset of 10000 iid samples, the data that is required for this task should be specified as follows:





We shall commence with the Support Vector Machine (SVM) classifier, which can be acquired by solving the optimization problem:

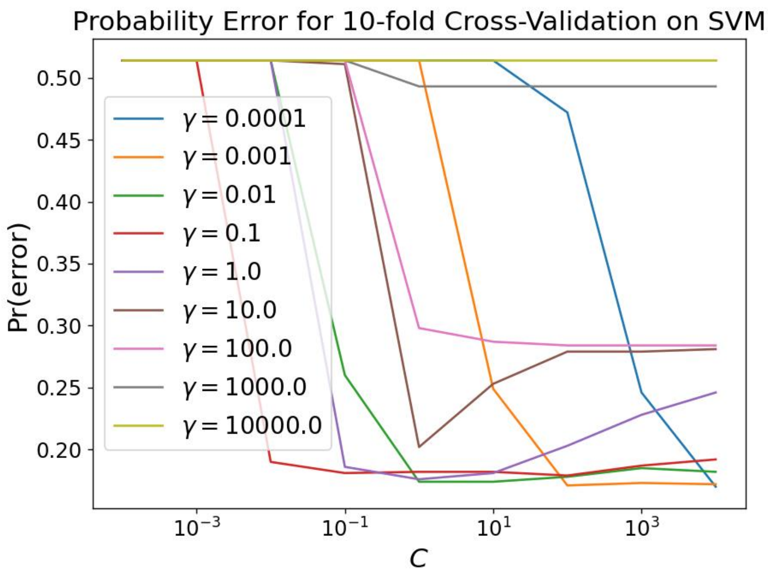
$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + \lambda \sum_{i=1}^{N} (1 - y^{(i)} \mathbf{w}^\intercal \mathbf{x}^{(i)})_+ .$$

In our undertaking, we shall focus exclusively on the RBF or Gaussian kernel for our purposes.

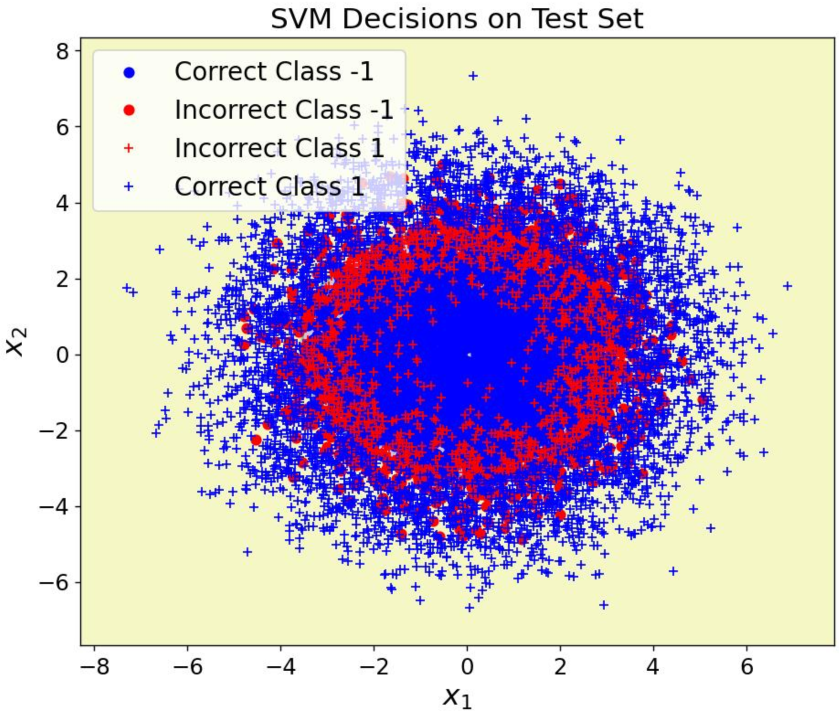
$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(- \frac{||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2}{2\sigma^2} \right),$$

The process of selecting the optimal regularization parameter (λ or c) and $\gamma=1/2\,\sigma^2$ value is a critical step in the hyperparameter selection procedure. The implementation for this model selection is performed through grid-search cross-validation on the SVM classifier, and the following section outlines the procedure.

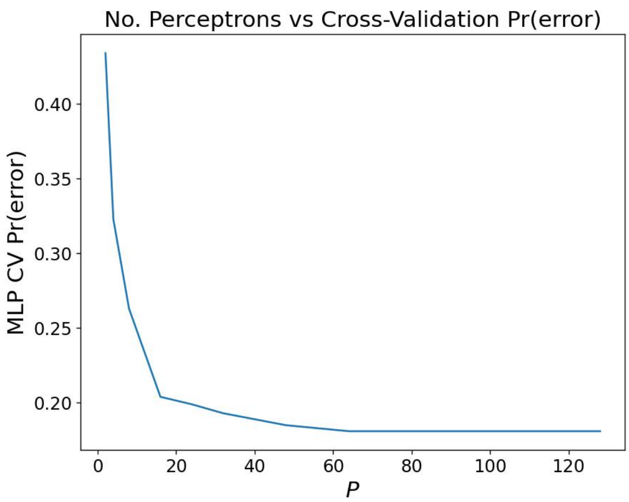
Best Regularization Strength	Best Kernel Width	SVM CV Pr(error)	SVM Pr(error) on the test data set
100.000	0.001	0.170	0.1732



Subsequently, the Support Vector Machine (SVM) shall be trained using the optimal C^* and y^* values obtained from the hyperparameter selection process, utilizing the entire training dataset. The performance of the model will then be evaluated on the test set, and the decision surface will be plotted for further analysis.

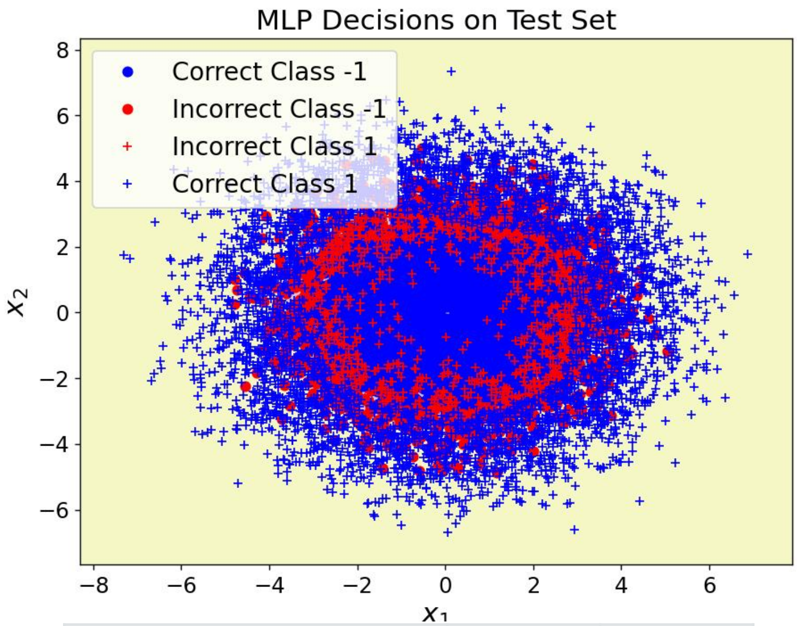


After defining the single-hidden layer MLP class and relevant functions for training and prediction, the next step is to utilize cross-validation to determine the optimal number of perceptrons, denoted as p*. The following section outlines the implementation details for this procedure.



Best # of Perceptrons:	Pr(error):
64	0.181

Upon training multiple MLPs using the optimal number of perceptrons, p*, obtained through the cross-validation process on the entire training set to prevent being trapped in local minima, our next step will be to report the probability of error on the test set. Additionally, we will visualize the correct and incorrect decisions made by the resulting classifier, akin to what we had previously done for the SVM.



	Train Accuracy(10-fold) /1000 samples	Train Accuracy(10-fold) /10000 samples
SVM	0.861	0.835
MLP	0.833	0.846

In comparison to the MLP classifier, the SVM classifier exhibits a higher training accuracy. However, the MLP classifier outperforms the SVM classifier in terms of classification accuracy on the test dataset. The test dataset comprises a significant number of samples that lie on the boundary between the two classes. As the SVM classifier is particularly sensitive to these samples, increasing their number during validation on the test dataset results in decreased accuracy. This is not the case for the MLP classifier, which exhibits improved accuracy on the test dataset.

Question 2

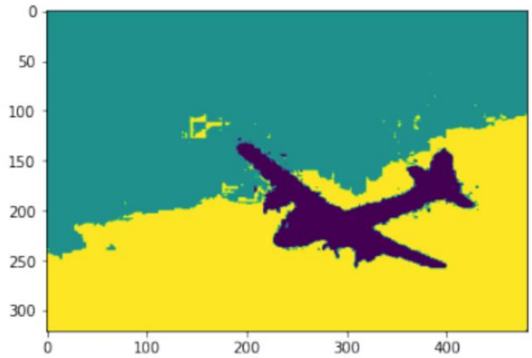
The provided images are transformed into 5D vectors by adding the row and column index and normalizing, which creates our dataset. The dataset is then modeled as a GMM with two components, and the GMM's parameters are estimated using the Expectation Maximization algorithm, which performs maximum likelihood parameter estimation. The MAP rule is used to classify each sample in the dataset into two classes, where each component of the GMM is a class conditional PDF. The ideal number of Gaussian components was determined by performing 10-fold cross validation and maximizing the average validation-log-likelihood as the objective function.

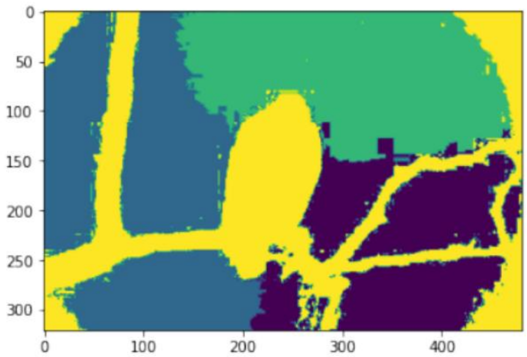
All the RGB photos used in this article have dimensions of (321, 481), which means that they

consist of 321 columns and 481 rows of pixels. In total, there are 1,544,401 pixels in each photo, and each pixel contains three values representing the intensity of red, green, and blue colors.
The normalize function from the sklearn library was employed to scale each of the 5 features to a [0,1] range. Subsequently, the GaussianMixture() method from the scipy package was utilized to fit the GMM.

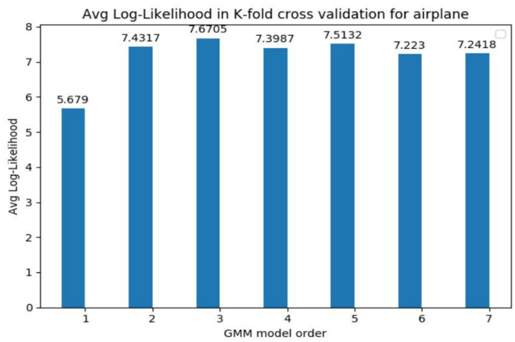
The normalized feature vector was used to calculate the pdf for each cluster, and the pixel was assigned to the cluster with the highest probability. The following images show the pixel-by-pixel segmentation using GMM for three RGB photos. The best number of Gaussian components for each image was determined by finding the maximum average validation-log-likelihood for a range of values from 1 to 11.

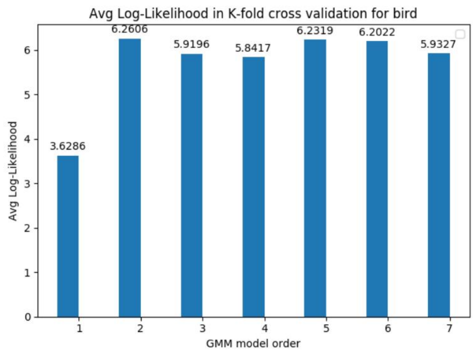
The following images show the result for the bird image and airplane image:



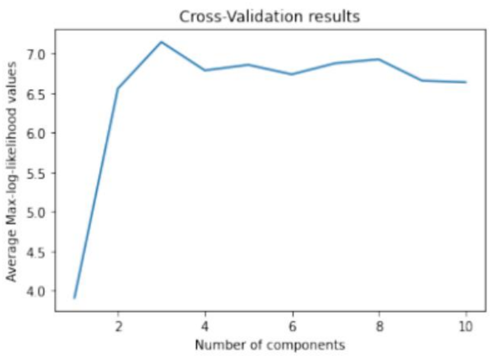


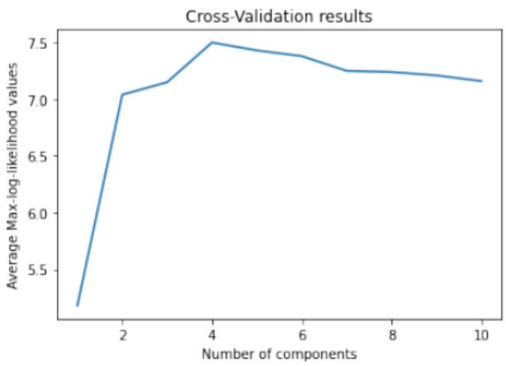
To determine the ideal model order for the class conditional PDFs, a 10-fold cross-validation is executed. The GMM parameters are estimated by means of Expectation-Maximization. During the K-fold step, the model that achieves the highest average log-likelihood for the validation data is chosen for the class conditional PDFs. Once the model is selected, its parameters are estimated using EM on the complete training data, and classified according to the MAP rule into the same number of classes as there are components in the GMM.





The graph below illustrates the average validation log-likelihood across several Gaussian components. According to the graph, the optimal number of Gaussian components for plane image is three, and the optimal number of Gaussian components for bird image is four.





Append

The code of hw4 is as follow link:
https://github.com/JonnyFan/ML5644/tree/main/hw4