HW3: Sample codes for Bayesian Classification

```
function Lpred = myBayesPredict(Dtrain, Ltrain, Dtest, opt)
% Multi-class classification using Bayesian Decision Theory
% Function Input:
% Dtrain: training dataset, each row is a feature vector of a training sample
% Ltrain: class labels of training samples
% Dtest:  testing dataset
% opt: classification options
%       if opt==1, use Naive Bayes
%       if opt==2, use posterior probability as discriminant function
%       if opt==3, use the derived formula based on multivariate normal
%       distribution
%
% Function Output:
% Lpred: predicted class labels for the testing samples in Dtest



%% HW3 Problem 1: use Naive Bayes to make classification
if opt==1
   Use Matlab functions NaiveBayes.fit and predict to obtain
   the predicted label Lpred
end


%% HW3 Problem 2: discriminant function = likelihood*prior
%  Use function mvnpdf to calculate likelihood directly
if opt==2
   C = unique(Ltrain);
   Lpred = [];
   for iC = 1:length(C)
        cl = C(iC);
        idx = find(Ltrain==cl);
        data = Dtrain(idx,:);
        mu = mean(data);
        sigma = cov(data);
        P = length(idx)/length(Ltrain);
        for j = 1:size(Dtest,1)
            x = Dtest(j, :);
            likelihood = mvnpdf(x,mu,sigma);
            prior = P;
            % record discriminat function values into G
            G(iC, j) = likelihood*prior;
        end
   end

   [~, pred] = max(G);

   Lpred = C(pred);
end


%% HW3 Problem 3: assume multivariate normal distribution
% Use the derived discriminant function to make classification
if opt==3
```

   Make classification using discriminant function $g_i(x) = x^t W_i x + w_i^t x + w_{i0}$

```
end
```

HW3: Sample codes for LDA Classification

```
function [Lpred, w] = FishersLDA(Dtrain, Ltrain, Dtest)
% Binary Classification using Fisher's linear discriminant
% 1. Find the optimal data projection direction w using the training dataset
% 2. And use Fisher's linear discriminant to make classification for testing dataset
% Note: the class labels use 1 and -1 to represent the two classes to be classified.

% Inputs:
%   Dtrain  - feature matrix for training dataset
%   Ltrain  - class labels of training samples, a vector of -1, 1
%   Dtest   - feature matrix for testing dataset

% Outputs
%   Lpred   - predicted class labels of the testing patterns
%   w       - optimal weight vector for data projection

%--------Fisher Linear Discriminant---------%
idx1 = find(Ltrain==1);  % the index for class 1
idx2 = find(Ltrain==-1); % the index for class -1
Dtrain_c1 = Dtrain(idx1, :);  % the feature matrix for class 1
Dtrain_c2 = Dtrain(idx2, :);  % the feature matrix for class 2

N_c1 = length(idx1);  % the number of samples in class 1
N_c2 = length(idx2);  % the number of samples in class 2

% Hint: the matrix S1 and S2 are actually covariance matrix for feature matrix of
class 1 and 2, respectively.
S1 = cov(Dtrain_c1);
S2 = cov(Dtrain_c2);

% Complete the following, and obtain the optimal direction w for sample projection
is calculated by the LDA formula



%-------Classification Part---------%
% Option 1: Perform classification using the decision boundary = (m̃1 + m̃2)/2
if opt==1

   calculate Lpred
end

% Option 2: Perform classification on Projected Data using Bayesian classification
function you made in HW3 Problem 2.
if opt==2

   calculate Lpred
end
```