Goutham Munagala **1001565060**

Question 1:

```
function Lpred = myBayesPredict(Dtrain, Ltrain, Dtest, opt)

%% 1. Use Naive Bayes Function to Make classification
if opt==1
NB = fitcnb(Dtrain,Ltrain);  % construct a Naive Bayes model NB
Lpred = predict(NB, Dtest);  % apply the trained model NB to predict
class of test samples in Dtest
end

%% 2. Use the discriminant function G(x) = likelihood*prior for
classification

if opt==2
   C = unique(Ltrain);
   Lpred = [];

   for iC = 1:length(C) % For each class i, calculate P(X|Wj)P(Wj) for
all testing samples
       cl = C(iC);
       idx = find(Ltrain==cl);
       data = Dtrain(idx,:);
       mu = mean(data); % feature mean vector
       sigma = cov(data); % feature covariance matrix
       P = length(idx)/length(Ltrain);

       % For each testing sample, calculate P(X|Wj)P(Wj) = likelihood of
class i * prior of class i
       for j = 1:size(Dtest,1)
           x = Dtest(j, :);
           likelihood = mvnpdf(x,mu,sigma); % likelihood of the current
class i
           prior = P; % prior of the current class i

           % Record values of the discriminat function G(X)
           % In the following matrix G, each row represent a class, and
           % each column represent a testing sample
           G(iC, j) = likelihood*prior; % P(X|Wj)P(Wj)
       end
   end

   % For each testing sample, find the index of the class that have
maximum
   % value of likelihood*prior
   [~, pred] = max(G);
   Lpred = C(pred);
end
```

```
%% 3. Use the derived discriminant function G(x) for classification
% based on the the assumption of Multivariate Normal Distribution for
features
if opt==3
   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);
       W = -0.5*inv(sigma);
       w = inv(sigma)*mu;
       w0 = -0.5*mu'*inv(sigma)*mu-0.5*log(det(sigma))+log(P);

       for j = 1:size(Dtest,1)
           x = Dtest(j, :)';
           % The closed form of the derived discriminant function G(X)
           G(iC, j) = x'*W*x + w'*x + w0;
       end
   end

   [~, pred] = max(G);
   Lpred = C(pred);

end
```

Results:

Confusion matrix for fold 1

1	2	3
10	0	0
0	10	0
0	0	10

Confusion matrix for fold 2

1	2	3
10	0	0
0	10	0
0	0	10

Goutham Munagala **1001565060**

Confusion matrix for fold 3

1	2	3
10	0	0
0	9	1
0	0	10

Confusion matrix for fold 4

1	2	3
10	0	0
0	9	1
0	1	9

Confusion matrix for fold 5

1	2	3
10	0	0
0	10	0
0	0	10

Overall confusion matrix

1	2	3
50	0	0
0	48	2
0	1	49

Accuracy for 5 folds

1
1
1
0.9667
0.9333
1

Overall average accuracy

1	2
0.9800	

Accuracy for Bayesian option 1,2 and 3

1	2
1	0.9533
2	0.9800
3	0.9800

Goutham Munagala **1001565060**

Question 2:

```
function [Lpred, w] = FishersLDA(Dtrain, Ltrain, Dtest, lambda)
% Binary Classification using Fisher's linear discriminant
if (nargin<3 || isempty(lambda))
   lambda = [0 1; 1 0];
end

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

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

sigma1 = cov(Dtrain_c1);
mu1 = mean(Dtrain_c1);

sigma2 = cov(Dtrain_c2);
mu2 = mean(Dtrain_c2);
Sw = sigma1 + sigma2;

%% The optimal direction w for sample projection:
w  = inv(Sw)*(mu1-mu2)';

%------The Projected Data-------%
Dtrain_new = Dtrain*w; % Projected training data
Ltrain_new = Ltrain;   % Training data label
Dtest_new = Dtest*w;   % Projected testing data

Dtrain_new_c1 = Dtrain_new(idx1, :);  % projected training samples of
class 1
Dtrain_new_c2 = Dtrain_new(idx2, :);  % projected training samples of
class 2

mu1_new = mean(Dtrain_new_c1);  % mean of projected samples of class 1
mu2_new = mean(Dtrain_new_c2);

sigma1_new = std(Dtrain_new_c1);  % standard deviation of projected
samples of class 1
sigma2_new = std(Dtrain_new_c2);

Ntest = size(Dtest, 1);
Lpred = [];

for i = 1:Ntest
        feat = Dtest_new(i);
        prior1 = length(idx1)/length(Ltrain);
```

```
        likelihood1 = normpdf(feat, mu1_new,sigma1_new); % likelihood of
the current class 1
        prior2 = length(idx2)/length(Ltrain);
        likelihood2 = normpdf(feat, mu2_new,sigma2_new); % likelihood of
the current class 2

        if (likelihood1/likelihood2) > (lambda(1,2)-
lambda(2,2))/(lambda(2,1)-lambda(1,1))*(prior2/prior1);
           pred = 1;
        else
           pred = 2;
        end

        Lpred(i,1) = pred;
    end
end
```

Results:

lambda = [0 1; 1 0]

5 fold accuracy

1
0.6087
0.6250
0.6957
0.6522
0.4783

Sensitivity and specificity

1	2
0.3000	0.8462
0.3636	0.8462
0.5000	0.8462
0.7000	0.6154
0.9091	0.0833

Overall accuracy

1
0.6120

Goutham Munagala **1001565060**

lambda = [0 5; 1 0]

Accuracy for 5 folds

1
0.5652
0.5652
0.5417
0.5417
0.5000

Sensitivity and specificity

1	2
0	1
0	1
0	1
0	1
0.4000	0.5833

Overall accuracy

1
0.5428