1. Consider a linear machine with discriminant $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$.

   (a) Show that the distance from the hyperplane $g(\mathbf{x}) = 0$ to a point $\mathbf{x}_a$ is $|g(\mathbf{x}_a)|/\|\mathbf{w}\|$, where $\|\cdot\|$ is the Euclidean norm, by minimizing $\|\mathbf{x} - \mathbf{x}_a\|^2$ subject to the constraint $g(\mathbf{x}) = 0$.

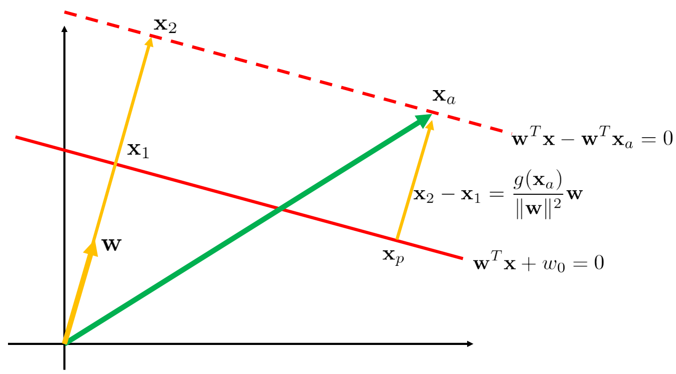

Figure 1: Geometric picture.

Ans: Let $\mathbf{x}^1$ and $\mathbf{x}^2$ be the two points on the line in the direction of $\mathbf{w}$ satisfying $\mathbf{w}^T\mathbf{x}^1 + w_0 = 0$ and $\mathbf{w}^T\mathbf{x}^2 - \mathbf{w}^T\mathbf{x}_a = 0$, respectively. This is shown in Figure 1. Then, the distance of $\mathbf{x}_a$ to the hyperplane is equal to the distance between $\mathbf{x}^1$ and $\mathbf{x}^2$. We know that $|\mathbf{w}^T\mathbf{x}^1 - \mathbf{w}^T\mathbf{x}^2| = |\langle \mathbf{w}, \mathbf{x}^1 - \mathbf{x}^2 \rangle| = \|\mathbf{w}\| \cdot \|\mathbf{x}^1 - \mathbf{x}^2\| = |-w_0 - \mathbf{w}^T\mathbf{x}_a|$. Hence

$$\|\mathbf{x}^1 - \mathbf{x}^2\| = \frac{|\mathbf{w}^T\mathbf{x}_a + w_0|}{\|\mathbf{w}\|} = \frac{|g(\mathbf{x}_a)|}{\|\mathbf{w}\|} .$$

   (b) Show that the projection of $\mathbf{x}_a$ onto the hyperplane is given by

$$\mathbf{x}_p = \mathbf{x}_a - \frac{g(\mathbf{x}_a)}{\|\mathbf{w}\|^2}\mathbf{w}.$$

Ans: The projection can be obtained by subtracting its projection $\mathbf{x}_p$ from $\mathbf{x}_a$. This difference is equal to $\mathbf{x}^2 - \mathbf{x}^1$ in part (a). Let us rewrite $\mathbf{x}^1 = \beta_1\mathbf{w}$ and $\mathbf{x}_2 = \beta_2\mathbf{w}$. We can determine β_1 and β_2 as follows.

$$\mathbf{w}^T\mathbf{x}^1 = \beta_1\|\mathbf{w}\|^2 = -w_0 \text{ and } \mathbf{w}^T\mathbf{x}^2 = \beta_2\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{x}_a$$

Thus, $\beta_1 = -w_0/\|\mathbf{w}\|^2$ and $\beta_2 = \mathbf{w}^T\mathbf{x}_a/\|\mathbf{w}\|^2$, and

$$\mathbf{x}^2 - \mathbf{x}^1 = \frac{\mathbf{w}^T\mathbf{x}_a + w_0}{\|\mathbf{w}\|^2}\mathbf{w} .$$

Thus,

$$\mathbf{x}_a - \mathbf{x}_p = \frac{\mathbf{w}^T\mathbf{x}_a + w_0}{\|\mathbf{w}\|^2}\mathbf{w} = \frac{g(\mathbf{x}_a)}{\|\mathbf{w}\|^2}\mathbf{w} .$$

2. For this problem, you will use the data provided in 'DiabetesTraining.csv'. The file contains the values of 8 features (in columns 1 through 8) and the label in the last column (0 - no diabetes, 1 - diabetes). The feature vector contains both categorical and numerical features.

   Ans: The Matlab code for both parts (a) and (b) is provided on the next page.

   (a) Using the numerical features 'age', 'BMI,' 'HbA1c level', and 'blood glucose level', find the vector $\mathbf{w}$ to be used for linear discriminant analysis for binary classification. You should standardize the data first by dividing the feature values by respective standard deviation.
   Ans: $\mathbf{w} = [0.0004\ \ 0.0005\ \ 0.0013\ \ 0.0011]^T$.

   (b) Suppose that we are interested in constructing a decision tree using all 8 features. Compute the Gini impurity and information gain for attributes 'hypertension' and 'heart disease' at the beginning.

   Ans: Gini impurity for Hypertension is 0.143, and that for heart disease is 0.146. The information game for Hypertension is 0.0143, and that for heart disease is 0.004.

```
% Load the data
data = readtable('DiabetesTraining.csv') ;
label = table2array(data(:,9)) ;
nSample = length(label) ;
Yes = find(label == 1) ; No = find(label == 0) ;
nYes = length(Yes) ; nNo = length(No) ;

% part (a)
dataA = table2array(data(:,[2 6:8])) ;
[DataA, DataAMean] = normalize(dataA, "scale") ;
dataAYes = DataA(Yes, :) ; dataANo = DataA(No, :) ;
meanYes = mean(dataAYes) ; meanNo = mean(dataANo) ;

dataAYesCent = dataAYes - repmat(meanYes, [nYes 1]) ;
dataANoCent = dataANo - repmat(meanNo, [nNo 1]) ;

% Compute the scatter matrices
scatterYes = dataAYesCent' * dataAYesCent ;
scatterNo = dataANoCent' * dataANoCent ;

scatterMat = scatterYes + scatterNo ;

% Optimal direction for LDA
weightLDA = inv(scatterMat) * (meanYes - meanNo)'


% part (b)

Hyperten = table2array(data(:,3)) ;
hyperYes = find(Hyperten == 1) ; hyperNo = find(Hyperten == 0) ;
nHyperYes = length(hyperYes) ; nHyperNo = length(hyperNo) ;
hyperYesYes = find(label(hyperYes) == 1) ; hyperYesNo = find(label(hyperYes) == 0)
;
hyperNoYes = find(label(hyperNo) == 1) ; hyperNoNo = find(label(hyperNo) == 0) ;
nHyperYesYes = length(hyperYesYes) ; nHyperYesNo = length(hyperYesNo) ;
nHyperNoYes = length(hyperNoYes)  ; nHyperNoNo = length(hyperNoNo) ;
hyperYesGini = 1 - (nHyperYesYes / nHyperYes)^2 - (nHyperYesNo / nHyperYes)^2 ;
hyperNoGini = 1 - (nHyperNoYes / nHyperNo)^2 - (nHyperNoNo / nHyperNo)^2 ;
HypertensionGini = (nHyperYes * hyperYesGini + nHyperNo * hyperNoGini) / nSample

Heart = table2array(data(:,4)) ;
heartYes = find(Heart == 1) ; heartNo = find(Heart == 0) ;
nHeartYes = length(heartYes) ; nHeartNo = length(heartNo) ;
heartYesYes = find(label(heartYes) == 1) ; heartYesNo = find(label(heartYes) == 0)
;
heartNoYes = find(label(heartNo) == 1) ; heartNoNo = find(label(heartNo) == 0) ;
nHeartYesYes = length(heartYesYes) ; nHeartYesNo = length(heartYesNo) ;
nHeartNoYes = length(heartNoYes)  ; nHeartNoNo = length(heartNoNo) ;
heartYesGini = 1 - (nHeartYesYes / nHeartYes)^2 - (nHeartYesNo / nHeartYes)^2 ;
```

```
heartNoGini = 1 - (nHeartNoYes / nHeartNo)^2 - (nHeartNoNo / nHeartNo)^2 ;
HeartGini = (nHeartYes * heartYesGini + nHeartNo * heartNoGini) / nSample

initEntropy = - nYes / nSample * log2(nYes / nSample) - nNo / nSample * log2(nNo /
nSample)
hyperYesEntropy = - nHyperYesYes / nHyperYes * log2(nHyperYesYes / nHyperYes) ...
                        - nHyperYesNo / nHyperYes * log2(nHyperYesNo / nHyperYes) ;
hyperNoEntropy = - nHyperNoYes / nHyperNo * log2(nHyperNoYes / nHyperNo) ...
                        - nHyperNoNo / nHyperNo * log2(nHyperNoNo / nHyperNo) ;
afterHypertensionEntropy = nHyperYes / nSample * hyperYesEntropy + nHyperNo /
nSample * hyperNoEntropy
HypertensionInfoGain = initEntropy - afterHypertensionEntropy

heartYesEntropy = - nHeartYesYes / nHeartYes * log2(nHeartYesYes / nHeartYes) ...
                        - nHeartYesNo / nHeartYes * log2(nHeartYesNo / nHeartYes) ;
heartNoEntropy = - nHeartNoYes / nHeartNo * log2(nHeartNoYes / nHeartNo) ...
                        - nHeartNoNo / nHeartNo * log2(nHeartNoNo / nHeartNo) ;
afterHeartEntropy = nHeartYes / nSample * heartYesEntropy + nHeartNo / nSample *
heartNoEntropy
HeartInfoGain = initEntropy - afterHeartEntropy
```