Pattern Recognition
Department of Electrical and Information Engineering
University of Cassino and Southern Latium, Second Semester 2018
Homework Assignment 1
Assigned 10 April 2018; due 11:59pm, 27 April 2018

In preparing my solutions, I did not look at any old home work, copy anybody's answers or let them copy mine.

Name: Md. Kamrul Hasan **Signature:**

Submitted By:

Md. Kamrul Hasan

E-mail: kamruleeekuet@gmail.com

MAIA (M2), UNICAS, Italy.

Submitted To:

Prof. Francesco Tortorella, Ph.D.

E-mail: francesco.tortorella@unicas.it

Universita degli Studi di Cassino e del Lazio Meridionale, Italy.

Problem 1.1 [5%] Describe (max one page) an application of pattern recognition related to your research. What are the features? What is the decision to be made? Comment on how one might solve the problem.

Pattern Recognition: Pattern recognition is a sub division of machine learning which is used for finding the patterns and regularities in data. Pattern recognition takes incoming data and assign it to some classes depending on key characteristics of data that may be supervised or unsupervised. The applications of Pattern Recognitions are so vast that it is quite hard to describe the specific applications of pattern recognition.

In the field of image processing and computer vision, more precisely, in medical imaging system, pattern recognition is the basic tools for designing any computer-aided diagnosis (CAD) systems. Because the goals of any CAD systems are to classify the objects and assign them to corresponding labels. For the interpretations and classifications of the normal or abnormal labels of the patient in CAD system, it should take some decisions based on the previous knowledge that we called training. So, in our filed, to classify the disease label, Pattern Recognition act as a decision maker based on the training labels.

Features: A feature is an explanatory variable used in statistical classification techniques that are individual measurable characteristic of an instance. On the other sense, features are numeric presentations of the samples, but structural features are also taken as acceptable. The features of all objects form a n-dimensional vector which is called *feature vector* and lives in n-dimensional space known as *features space*. If each object has $\boldsymbol{n}$ number of features, we can write below mathematical formulations.

$$Data\ (both\ Training\ and\ Testing) \in \mathbb{R}^n$$

Decision and how to classify the objects: In pattern recognition, decision is made based on the posterior probability of any unknown objects. Let's clarify this topics by discussing mathematically. For the simplicity, binary classifier is taken into the account. Suppose, we have some sets of data (X_i, Y_i) for $i = 1,2,3, \ldots \ldots \ldots \ldots . N\ (Samples)$. Where X_i has a n dimensional features vector that live in $\mathbb{R}^n$ feature space and Y_i is the corresponding class labels of the X_i. Since, X_i contain only two types (binary classifier) data and has two prior probabilities which are $P(w_1)$ and $P(w_2)$. We also have two likelihood functions from the distributions of the X_i that are $P(X|w_1)$ and $P(X|w_2)$. From those information, we can find out the posterior probability using ***Bayes theorem***.

$$P(w_1|X) = \frac{P(X|w_1)*P(w_1)}{P(X)} \qquad (1.1.1)$$

$$P(w_2|X) = \frac{P(X|w_2)*P(w_2)}{P(X)} \qquad (1.1.2)$$

Where, $P(X) = \sum_{i=1}^{2} P(X|w_i) * P(w_i)$. From the Eq. (1.1.1) and Eq. (1.1.2), we can write the following formula-

$$\frac{P(X|w_1)*P(w_1)}{P(X)} \gtrless \frac{P(X|w_2)*P(w_2)}{P(X)} \qquad (1.1.3)$$

Since, $P(X)$ does not affect the decision rule, its act as a scaling parameter. So, we can eliminate those term from the above equation for the simplicity. From the Eq. (1.1.3), we get-

$$P(X|w_1) * P(w_1) \gtrless P(X|w_2) * P(w_2)$$

In a simple manner and simple classifications, we may decide the class labels based on this posterior probability. For example, if the posterior $P(w_1|X)$ viz. $P(X|w_1) * P(w_1)$ is greater than the posterior $P(w_2|X)$ viz. $P(X|w_2) * P(w_2)$ then that particular X belongs to class w_1 and vice-versa.

Problem 1.2 [15%] In a particular two class problem, the conditional densities for a scalar feature x are $P(x|w_1) = k_1 exp^{(-\frac{x^2}{15})}$ and $P(x|w_2) = k_2 exp^{(-\frac{(x-7)^2}{14})}$.

a) Find k1 and k2, and plot the two densities on a single graph using ***Matlab***
b) Find the decision regions which minimize the average probability of error and indicate them on the plot you made in part (a) for the following cases:
   (b.1) $\frac{P(w_2)}{P(w_1)} = 1$
   (b.2) $\frac{P(w_2)}{P(w_1)} = 0.5$
   (b.3) $\frac{P(w_2)}{P(w_1)} = 4$

<u>Solution of 1.2 (a):</u> In probability theory, the normal or Gaussian distribution is a continuous probability distribution. The probability density of the normal distribution is given by-

$$P(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (1.2.1)$$

The conditional densities that are given in this problem are-

$$P(x|w_1) = k_1 exp^{(-\frac{x^2}{15})} \qquad (1.2.2)$$

$$P(x|w_2) = k_2 exp^{(-\frac{(x-7)^2}{14})} \qquad (1.2.3)$$

Comparing Eq. 1.2.1 and Eq. 1.2.2, we get,

$$2\sigma^2 = 15 \text{ and } \mu = 0$$

$$k_1 = \frac{1}{\sqrt{2\pi\sigma^2}} = \frac{1}{\sqrt{\pi * 15}} = 0.1457$$

Similarly, comparing Eq. 1.2.1 and Eq. 1.2.3, we get,

$$2\sigma^2 = 14 \text{ and } \mu = 7$$

$$k_2 = \frac{1}{\sqrt{2\pi\sigma^2}} = \frac{1}{\sqrt{\pi * 14}} = 0.1507$$

To plot the densities of those functions as shown in Figure 1.2.1 in a single graph in *MATLAB*, I have made a *MATLAB* source code named "*Problem1_2_a_DesnsityFunction.m"* that are provided with this report.

From the Figure 1.2.1, it is noticeable that the distribution of $P(x|w_1)$ is almost symmetric about 0 value as well as it spans near about 15 that indicate it has $\mu = 0$ and $2\sigma^2 = 15$ respectively. Similarly, the distribution of $P(x|w_2)$ is almost symmetric about 7 as well as it spans near about 14 which indicate it has $\mu = 7$ and $2\sigma^2 = 14$ respectively.

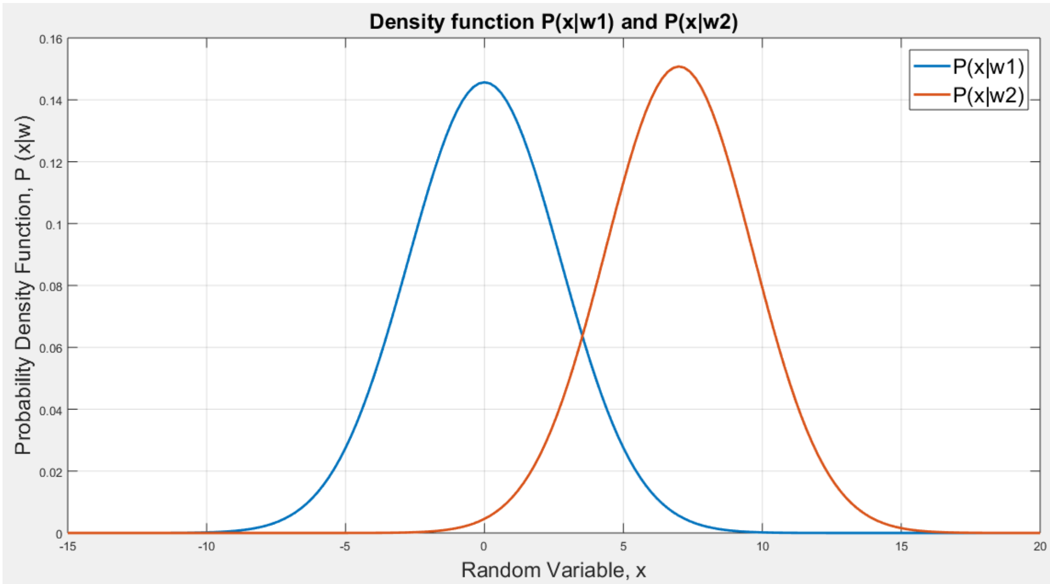

Figure 1.2.1: Probability density function w.r.t. random variable x.

<u>Solution of 1.2 (b):</u> There are 3 difference conditions, for making the decision regions. Let's them solve in report, then their corresponding graphical presentation is shown in Figure 1.2.2. To access this *MATLAB* source code, please see the "*Problem1_2_b_DecisionRegion.m"* in the folder containing this report.

(b.1) we have, $\frac{P(w_2)}{P(w_1)} = 1$, $P(x|w_1) = \frac{1}{\sqrt{\pi * 15}} exp^{(-\frac{x^2}{15})}$ and $P(x|w_2) = \frac{1}{\sqrt{\pi * 14}} exp^{(-\frac{(x-7)^2}{14})}$.

The term likelihood ratio is also known as known as the likelihood ratio test and expressed by the following-

$$\Lambda(x) = \frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)}$$

Putting all the values that are given for that problem in the above equation, we get the following result-

$$\Lambda\ (x) = \frac{\frac{1}{\sqrt{\pi * 15}} * exp^{(-\frac{x^2}{15})}}{\frac{1}{\sqrt{\pi * 14}} * exp^{(-\frac{(x-7)^2}{14})}} \gtrless 1$$

$$= \frac{\frac{1}{\sqrt{\pi * 15}} * exp^{(-\frac{x^2}{15}+\frac{(x-7)^2}{14})}}{\frac{1}{\sqrt{\pi * 14}}} \gtrless 1$$

For the simplicity taking the ln (e base log) on both side. That gives us following results,

$$-\frac{x^2}{15} + \frac{(x-7)^2}{14} \gtrless \ln(1 * \frac{\frac{1}{\sqrt{\pi * 14}}}{\frac{1}{\sqrt{\pi * 15}}})$$

$$\Rightarrow -14 * x^2 + 15 * (x-7)^2 \gtrless \ln(1 * \frac{\frac{1}{\sqrt{\pi * 14}}}{\frac{1}{\sqrt{\pi * 15}}}) * 210$$

$$\Rightarrow -14x^2 + 15x^2 - 210x + 735 - 7.03 \gtrless 0$$

$$\Longrightarrow x^2 - 210x + 727.91 \gtrless 0$$

Solving this quadratic equation, we get $x = 3.52\,, 206.47$. Now, using this value of x, we can plot the decision boundary that is shown in Figure 1.2.2 (a).

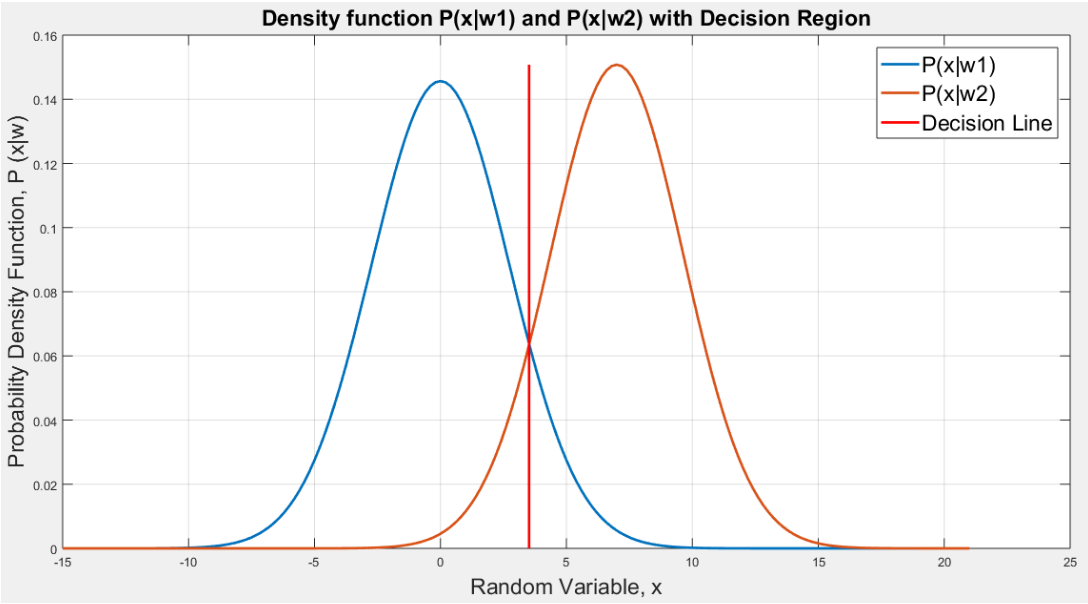

Figure 1.2.2 (a): Probability distribution with decision boundary when equal prior

The values $x = 206.47$ is neglected for the plotting of decision region because this value is not consistent with the distribution of the given data. From the Figure 1.2.2 (a), we can claim that when prior probability ratio is 1, two decision regions are equally distributed. It is mentionable that left side of the perpendicular line (Red color) is region R1 and other side is region R2.

(b.2) Similarly, for $\frac{P(w_2)}{P(w_1)} = 0.5$, with the help of the previous calculations we get-

$$-\frac{x^2}{15} + \frac{(x-7)^2}{14} \gtrless \ln\left(0.5 * \frac{\frac{1}{\sqrt{\pi * 14}}}{\frac{1}{\sqrt{\pi * 15}}}\right)$$

$$\Longrightarrow -14 * x^2 + 15 * (x-7)^2 \gtrless \ln\left(0.5 * \frac{\frac{1}{\sqrt{\pi * 14}}}{\frac{1}{\sqrt{\pi * 15}}}\right) * 210$$

$$\Longrightarrow -14x^2 + 15x^2 - 210x + 735 + 138.48 \gtrless 0$$

$$\Longrightarrow x^2 - 210x + 873.48 \gtrless 0$$

Solving this quadratic equation, we get $x = 4.24\,, 205.75$. Now, using this value of x, we can plot the decision boundary that is shown in Figure 1.2.2 (b). The values $x = 205.75$ is neglected for the plotting of decision region because this value is not consistent with the distribution of the given data. From the Figure 1.2.2 (b), when prior probability ratio is 0.5 that means the prior of $P(w_1)$ is double than $P(w_2)$ decision line is shifted to right. Right shifting means $P(x|w_1)$ is getting more space for random variable x. It is mentionable that left side of the perpendicular line (Red color) is region R1 and other side is region R2.

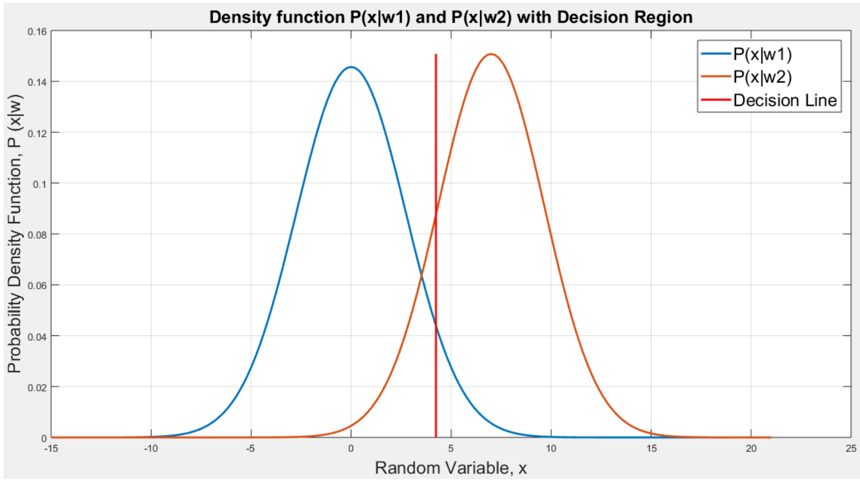

Figure 1.2.2 (b): Probability distribution with decision boundary when $P(w_1)$ is more.

(b.3) Similarly, for $\frac{P(w_2)}{P(w_1)} = 4$, with the help of the previous calculations we get-

$$-\frac{x^2}{15} + \frac{(x-7)^2}{14} \gtrless \ln\left(4 * \frac{\frac{1}{\sqrt{\pi * 14}}}{\frac{1}{\sqrt{\pi * 15}}}\right)$$

$$\Longrightarrow -14 * x^2 + 15 * (x-7)^2 \gtrless \ln\left(4 * \frac{\frac{1}{\sqrt{\pi * 14}}}{\frac{1}{\sqrt{\pi * 15}}}\right) * 210$$

$$\Longrightarrow -14x^2 + 15x^2 - 210x + 735 - 298.21 \gtrless 0$$

$$\Longrightarrow x^2 - 210x + 436.79 \gtrless 0$$

Solving this quadratic equation, we get $x = 207.89, 2.10$. Now, using this value of x, we can plot the decision boundary that is shown in Figure 1.2.2 (c). The values $x = 207.89$ is neglected for the plotting of decision region because this value is not consistent with the distribution of the given data. From the Figure 1.2.2 (c), when prior probability ratio is 4 that means the prior of $P(w_2)$ is 4 times than $P(w_1)$ decision line is shifted to the left. Left shifting means $P(x|w_2)$ is getting more space for random variable x. It is mentionable that left side of the perpendicular line (Red color) is region R1 and other side is region R2.

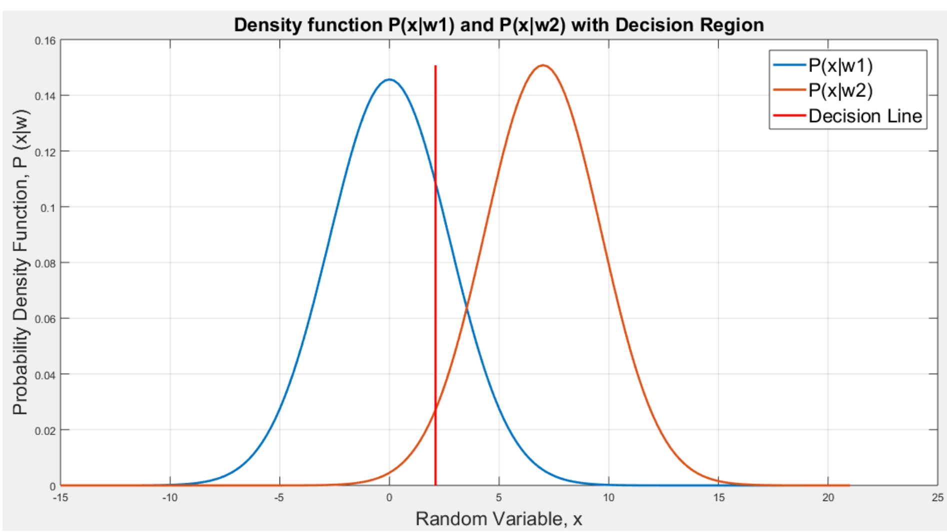

Figure 1.2.2 (c): Probability distribution with decision boundary when $P(w_2)$ is more.

Problem 1.3 [20%] Consider a medical diagnosis problem where a biochemical test is used for screening patients. The test returns a result close to 0 for healthy patients and close to 1 for sick patients, according to the following likelihood functions: $P(x|w_1) = N(0.0, 0.1)$ and $P(x|w_2) = N(1.5, 0.3)$, where $N(\mu;\ \sigma)$ is the Gaussian univariate density. Assume that, on average, 1 out of 10,000 patients is sick, and the following costs:

1. λ_{12} = 800,000 euros for unproductive test cost in the case of wrongly diagnosed disease;
2. λ_{21} = 1,500 euros for reimbursement to the patient in the case of not diagnosed disease;
3. $\lambda_{11} = \lambda_{22} = 0$ in the case of correct diagnosis

and define the decision rule that minimizes the Conditional Risk.

<u>Solution of 1.3:</u> Let us assume that, class, $w_1$is for healthy patients and class, w_2 is for sick patients. In this problem, given that,

$$P(x|w_1) = N(0.0, 0.1) = \frac{1}{\sqrt{2\pi * 0.01}} exp^{(-\frac{x^2}{0.02})}$$

$$\text{And } P(x|w_2) = N(1.5, 0.3) = \frac{1}{\sqrt{2\pi * 0.09}} exp^{(-\frac{(x-1.5)^2}{0.18})}$$

$$P(w_2) = \frac{1}{10000}, P(w_1) = \frac{9999}{10000}, \lambda_{11} = \lambda_{22} = 0, \lambda_{12} = 800{,}000 \text{ and } \lambda_{21} = 1{,}500$$

The conditional risk for binary classification can be written as follows with the action, α_i, of classifying a features vector, x, as class, w_i,

$$R(\alpha_1|x) = \lambda_{11}P(w_1|x) + \lambda_{12}P(w_2|x) \qquad (1.3.1)$$

$$R(\alpha_2|x) = \lambda_{21}P(w_1|x) + \lambda_{22}P(w_2|x) \qquad (1.3.2)$$

Putting given values in Eq. 1.3.1, we get-

$$R(\alpha_1|x) = 800{,}000 * P(w_2|x) = 800{,}000 * \frac{P(x|w_2) * P(w_2)}{P(x)} \text{ [Applying Bayes Formula].}$$

Similarly, from Eq. 1.3.2, we get-

$$R(\alpha_2|x) = 1{,}500 * P(w_1|x) = 1{,}500 * \frac{P(x|w_1) * P(w_1)}{P(x)} \text{ [Applying Bayes Formula].}$$

So, to get the decision rule with optimal decision boundary, we need to solve following equations that will gives us roots-

$$R(\alpha_1|x) = R(\alpha_2|x)$$

$$\Rightarrow 800{,}000 * \frac{P(x|w_2) * P(w_2)}{P(x)} = 1{,}500 * \frac{P(x|w_1) * P(w_1)}{P(x)}$$

$$\Rightarrow \frac{P(x|w_2)}{P(x|w_1)} = \frac{1{,}500 * P(w_1)}{800{,}000 * P(w_2)} \text{ [After Simplifications]}$$

$$\Rightarrow \frac{\frac{1}{\sqrt{2\pi * 0.09}} exp^{(-\frac{(x-1.5)^2}{0.18})}}{\frac{1}{\sqrt{2\pi * 0.01}} exp^{(-\frac{x^2}{0.02})}} = \frac{1{,}500 * P(w_1)}{800{,}000 * P(w_2)}$$

$$\Rightarrow \frac{exp^{(-\frac{(x-1.5)^2}{0.18})}}{exp^{(-\frac{x^2}{0.02})}} = \frac{\sqrt{2\pi * 0.09} * 1{,}500 * P(w_1)}{\sqrt{2\pi * 0.01} * 800{,}000 * P(w_2)}$$

$$\Rightarrow -\frac{(x-1.5)^2}{0.18} + \frac{x^2}{0.02} = \ln\left(\frac{\sqrt{2\pi * 0.09} * 1{,}500 * P(w_1)}{\sqrt{2\pi * 0.01} * 800{,}000 * P(w_2)}\right)$$

$$\Longrightarrow -\frac{50 * (x - 1.5)^2}{9} + 50 * x^2 = \ln\left(\frac{\sqrt{2\pi * 0.09} * 1{,}500 * P(w_1)}{\sqrt{2\pi * 0.01} * 800{,}000 * P(w_2)}\right)$$

$$\Longrightarrow 450x^2 - 50 * (x^2 - 3x + 2.25) = 9 * \ln\left(\frac{\sqrt{2\pi * 0.09} * 1{,}500 * P(w_1)}{\sqrt{2\pi * 0.01} * 800{,}000 * P(w_2)}\right)$$

$$\Longrightarrow 400x^2 + 150x - 112.5 - 36.27 = 0$$

$$\Longrightarrow 400x^2 + 150x - 148.77 = 0$$

Solving this quadratic equation, we get $x = 0.451, -0.826$. Those two points are the intersecting point on the random variable, x as like in Figure 1.3.1. We can claim that this decision boundary (Red Curve) has minimums conditional risk of the binary classifications which means that it has less cost if diagnosed disease of healthy patients and more cost if not diagnosed disease of the sick patient. The *MATLAB* source code used for this plotting is named *"Problem1_3_DecisionRegion.m"* which are available in the folder that contain this report.

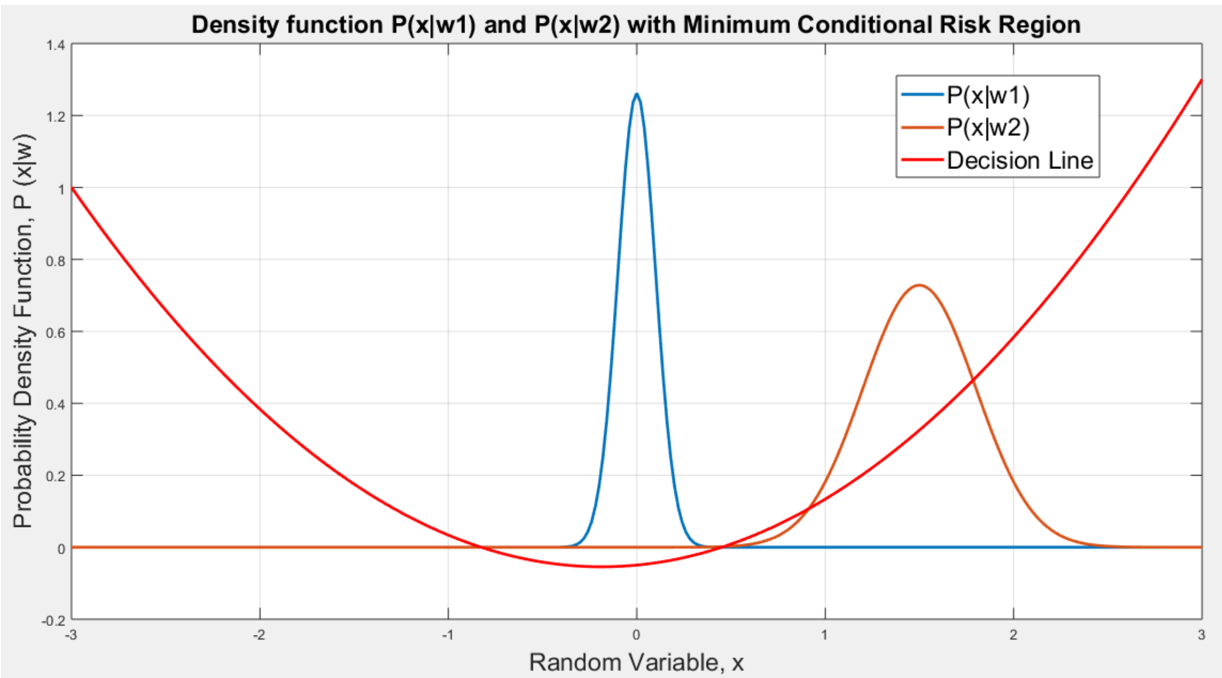

Figure 1.3.1: Decision boundary that corresponding to minimum conditional risk

Problem 1.4 [30%] The file hw1data.zip available in the homework folder contains samples coming from a two-class problem, each made of 10 numerical features and a binary label (±1). Split the data into training and test sets by randomly selecting 25% of the examples from each class for the test set.

(a) Using the training data, implement a linear classifier and a quadratic classifier and evaluate on the test set the True Positive Rate obtained for a False Positive Rate of 0.1. Repeat the above steps several times. What is the average obtained TPR for each of the implemented classifiers? Discuss your results.

(b) Repeat the previous exercise, but this time implementing a *k - NN* classifier. Experiment with several values for *k*. Discuss your results and compare them with those obtained in the previous exercise.

Solution of 1.4 (a): LDA is a method used in pattern recognition to find a linear combination of features that characterizes or separates two or more classes of objects or events whereas QDA is used to separate measurements of two or more classes of objects or events by a quadric surface. Let's first describe the overall frame work that is done for this question as shown in Figure 1.4.1.

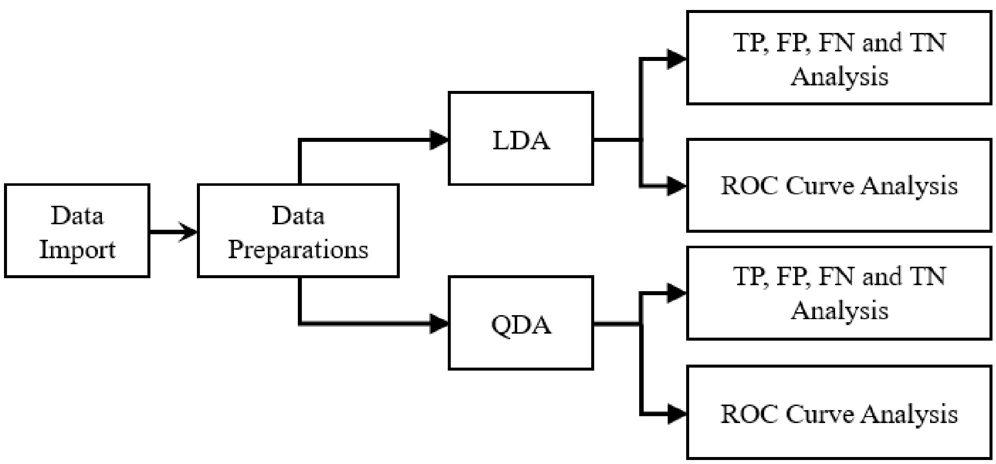

Figure 1.4.1: Overall framework for LDA and QDA

In the 1st block, data is imported to the *MATLAB* workspace for the classification. This data contains 8000 rows and 11 columns where 8000 rows indicate the measurements of the observations that also known as the instance of the class. There are 10 features for each instances of the observations that are first 10 columns (1 to 10) of the imported data. Last column (11th) indicates the class label (+1 or -1) corresponding to each instance. Sample for each row there is a corresponding class label. In this data set, there are 2 class which are +1 and -1.

After importing the data, I need some preparations to this data. Firstly, data are not normalized which means the variations of the values for each column are too much deviated. For that I need normalizations or standardizations. The standardizations formula used for this work is given in Eq. 1.4.1.

$$Data\,(:,1:10) = \frac{Data(:,1:10) - mean}{Standard\ Deviation} \qquad (1.4.1)$$

Where, both mean and Standard Deviation are the 1×10 $vectors$. And we don't need to normalize last column because it's the class level only. Now, to prepare the Training and Testing data set, we will split data for several subsections that are presented below in a block diagram in Figure 1.4.2.

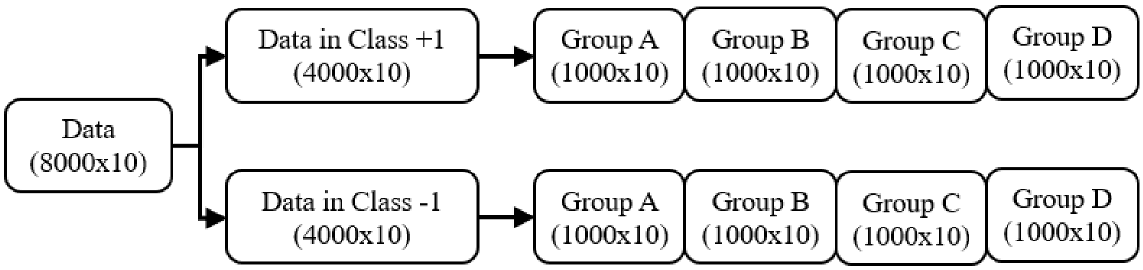

Figure 1.4.2: Data splitting for making Training and Testing data

From the Figure 1.4.2, it is shown that after splitting Data in +1 and -1 class, we divide data into 4 parts where each part contains 25% of the total data corresponding to that class (either +1 or -1). From this splitting of data, we must randomly select data group for training and testing. There are 4 possible cases of selecting the training and testing of the data as mention in Table 1.

Table 1: Training (75 %) and Testing (25 %) data selections from the Split data

Case of Selections	**Training set**		**Testing Set**	
	Data Class, +1	**Data Class, -1**	**Data Class, +1**	**Data Class, -1**
Case I	Group: A, B, and C	Group: A, B, and C	Group: D	Group: D
Case II	Group: A, B, and D	Group: A, B, and D	Group: C	Group: C
Case III	Group: A, C, and D	Group: A, C, and D	Group: B	Group: B
Case IV	Group: B, C, and D	Group: B, C, and D	Group: A	Group: A
Dimensions	3000x10	3000x10	1000x10	1000x10
Total Dimensions for Training: **6000x10** for Testing: **2000x10**				

After getting the Training (75 %) and Testing (25 %) data, LDA and QDA is performed separately. From each classifier, some analysis like Accuracy (TP, TN, FP and FN), ROC curve analysis is done as a performance measurement of the classifiers (LDA, QDA).

Result and discussion: According to Table 1, whole results has 4 possible parts (Case I, Case II, Case III and Case IV) for different Training (75 %) and Testing (25 %) Data. The *MATLAB* source code for the problem is named as "*Problem1_4_a_LDA_QDA.m*" is attached with this report in the folder.

Case I: In this case, Training Data will be Group: A, B, and C and Testing data will be Group: D from each of the classes (+1, -1). For this case, confusion matrix with some performance are given below in a Table 2 for LDA and Table 3 for QDA. From the ROC curve of the LDA and QDA as shown in Figure 1.4.3. it is noticeable that, for **0.1** False Positive rate, I get **TPR 0.58** and **0.61** respectively for LDA and QDA. The specificity, sensitivity, PPV, NPV, Accuracy, F1 score and AUC for both LDA and QDA are shown in Table 2 and Table 3 respectively.

Table 2: Confusion matrix for LDA for Case I

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 694	False Positive (FP) 159	$PPV = \frac{TP}{TP+FP}$ $= 81.4\ \%$
Test Outcome -1	False Negative (FN) 306	True Negative (TN) 841	$NPV = \frac{TN}{TN+FN}$ $= 73.3\ \%$
	$Sensitivity = \frac{TP}{TP+FN}$ $= 69.4\ \%$	$Specificity = \frac{TN}{TN+FP}$ $= 84.1\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 76.75\ \%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 74.91\ \%$	
Area Under ROC (AUC)		0.83079	

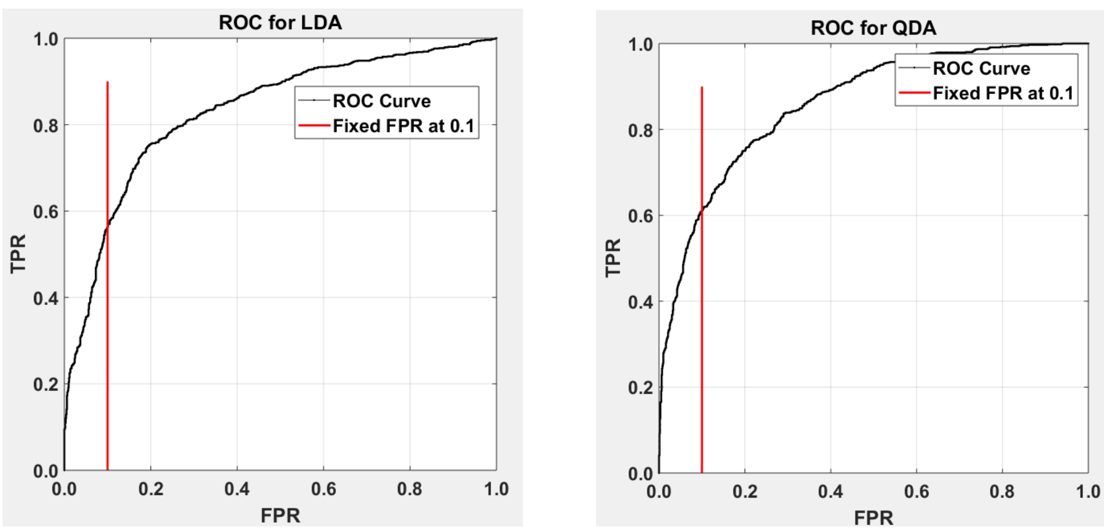

Figure 1.4.3: ROC curve for (left) LDA and (right) QDA for Case I

Table 3: Confusion matrix for QDA for Case I

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 513	False Positive (FP) 62	$PPV = \frac{TP}{TP+FP}$ $= 89.2\ \%$
Test Outcome -1	False Negative (FN) 487	True Negative (TN) 938	$NPV = \frac{TN}{TN+FN}$ $= 65.8\ \%$
	$Sensitivity = \frac{TP}{TP+FN}$ $= 51.3\ \%$	$Specificity = \frac{TN}{TN+FP}$ $= 93.8\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 72.60\%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 65.14\ \%$	
Area Under ROC (AUC)		0.86074	

Case II: In this case, Training Data will be Group: A, B, and D and Testing data will be Group: C from each of the class (+1, -1). For this case, confusion matrix with some performance are given below in a Table 4 for LDA and Table 5 for QDA. From the ROC curve of the LDA and QDA as shown in Figure 1.4.4. it is noticeable that, for **0.1** False Positive rate, I get TPR **0.52** and **0.64** respectively for LDA and QDA. The specificity, sensitivity, PPV, NPV, Accuracy, F1 score and AUC for both LDA and QDA are shown in Table 4 and Table 5 respectively.

Table 4: Confusion matrix for LDA for Case II

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 678	False Positive (FP) 160	$PPV = \frac{TP}{TP + FP}$ $= 80.9\ \%$
Test Outcome -1	False Negative (FN) 322	True Negative (TN) 840	$NPV = \frac{TN}{TN + FN}$ $= 72.3\ \%$
	$Sensitivity = \frac{TP}{TP + FN}$ $= 67.8\ \%$	$Specificity = \frac{TN}{TN + FP}$ $= 84.0\ \%$	
Accuracy		$\frac{TP + TN}{TP + FP + TN + FN} = 75.9\ \%$	
F1 Score		$\frac{2TP}{2TP + FP + FN} = 73.77\ \%$	
Area Under ROC (AUC)		0.83757	

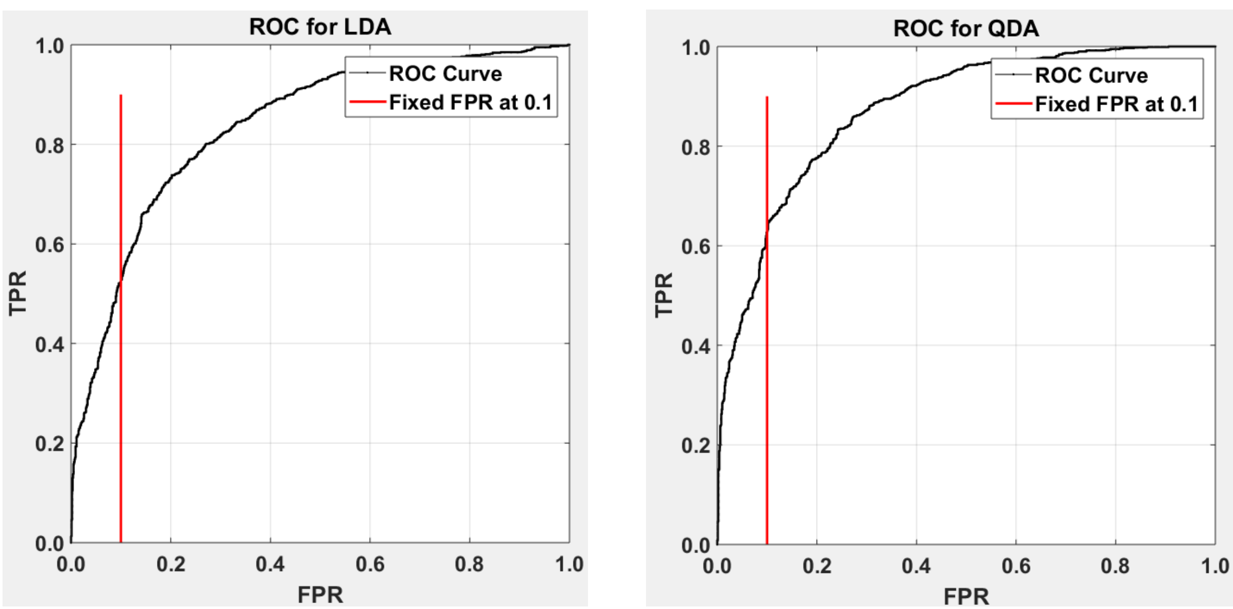

Figure 1.4.4: ROC curve for (left) LDA and (right) QDA for Case II

Table 5: Confusion matrix for QDA Case II

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 544	False Positive (FP) 84	$PPV = \frac{TP}{TP + FP}$ $= 86.60\%$
Test Outcome -1	False Negative (FN) 456	True Negative (TN) 916	$NPV = \frac{TN}{TN + FN}$ $= 66.8\ \%$
	$Sensitivity = \frac{TP}{TP + FN}$ $= 54.4\ \%$	$Specificity = \frac{TN}{TN + FP}$ $= 91.6\ \%$	
Accuracy		$\frac{TP + TN}{TP + FP + TN + FN} = 73.00\%$	
F1 Score		$\frac{2TP}{2TP + FP + FN} = 66.80\ \%$	
Area Under ROC (AUC)		0.87424	

<u>Case III:</u> In this case, Training Data will be Group: A, C, and D and Testing data will be Group: B from each of the class (+1, -1). For this case, confusion matrix with some performance are given below in a Table 6 for LDA and Table 7 for QDA. From the ROC curve of the LDA and QDA as shown in Figure 1.4.5. it is noticeable that, for **0.1** False Positive rate, I get TPR **0.57** and **0.63** respectively for LDA and QDA. The specificity, sensitivity, PPV, NPV, Accuracy, F1 score and AUC for both LDA and QDA are shown in Table 6 and Table 7 respectively.

Table 6: Confusion matrix for LDA Case III

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 681	False Positive (FP) 154	$PPV = \frac{TP}{TP + FP}$ $= 81.55\ \%$
Test Outcome -1	False Negative (FN) 319	True Negative (TN) 846	$NPV = \frac{TN}{TN + FN}$ $= 72.6\ \%$
	$Sensitivity = \frac{TP}{TP + FN}$ $= 68.1\ \%$	$Specificity = \frac{TN}{TN + FP}$ $= 84.6\ \%$	
Accuracy		$\frac{TP + TN}{TP + FP + TN + FN} = 76.35\ \%$	
F1 Score		$\frac{2TP}{2TP + FP + FN} = 74.22\ \%$	
Area Under ROC (AUC)		0.84081	

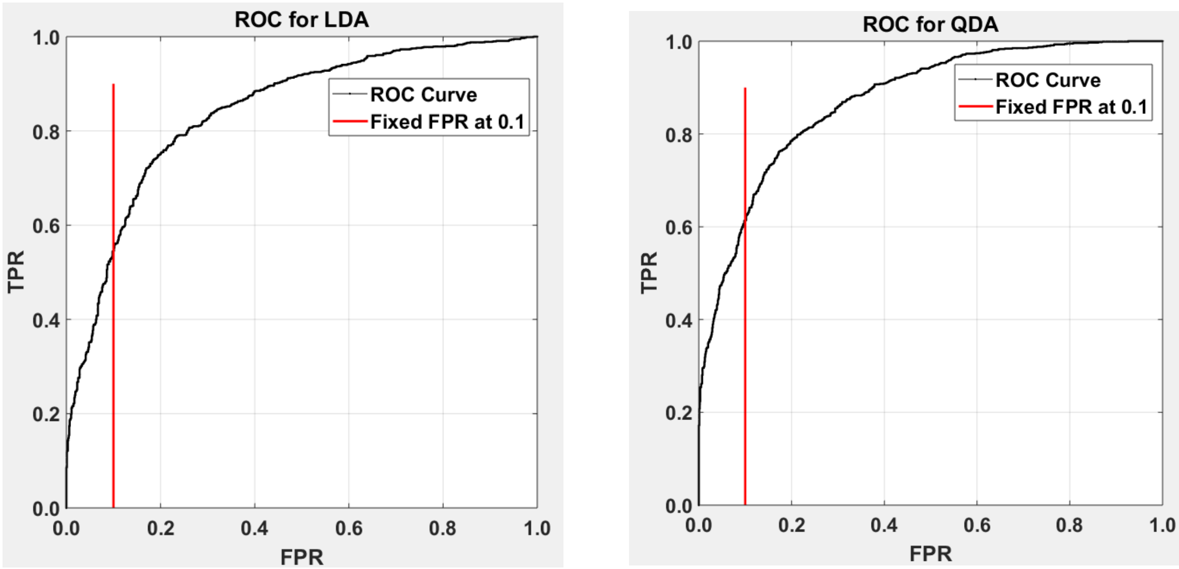

Figure 1.4.4: ROC curve for (left) LDA and (right) QDA for Case III

Table 7: Confusion matrix for QDA for Case III

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 513	False Positive (FP) 64	$PPV = \frac{TP}{TP + FP}$ $= 88.9\ \%$
Test Outcome -1	False Negative (FN) 487	True Negative (TN) 936	$NPV = \frac{TN}{TN + FN}$ $= 65.77\ \%$
	$Sensitivity = \frac{TP}{TP + FN}$ $= 51.3\ \%$	$Specificity = \frac{TN}{TN + FP}$ $= 93.6\ \%$	
Accuracy		$\frac{TP + TN}{TP + FP + TN + FN} = 72.45\ \%$	
F1 Score		$\frac{2TP}{2TP + FP + FN} = 65.06\ \%$	
Area Under ROC (AUC)		0.87313	

Case IV: In this case, Training Data will be Group: B, C, and D and Testing data will be Group: A from each of the class (+1, -1). For this case, confusion matrix with some performance are given below in a Table 8 for LDA and Table 9 for QDA. From the ROC curve of the LDA and QDA as shown in Figure 1.4.5. it is noticeable that, for **0.1** False Positive rate, I get TPR **0.59** and **0.62** respectively for LDA and QDA. The specificity, sensitivity, PPV, NPV, Accuracy, F1 score and AUC for both LDA and QDA are shown in Table 8 and Table 9 respectively.

Table 8: Confusion matrix for LDA for Case IV

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 691	False Positive (FP) 143	$PPV = \frac{TP}{TP+FP}$ $= 82.85\ \%$
Test Outcome -1	False Negative (FN) 309	True Negative (TN) 857	$NPV = \frac{TN}{TN+FN}$ $= 73.5\ \%$
	$Sensitivity = \frac{TP}{TP+FN}$ $= 69.1\ \%$	$Specificity = \frac{TN}{TN+FP}$ $= 85.7\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 77.4\%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 75.35\ \%$	
Area Under ROC (AUC)		0.83686	

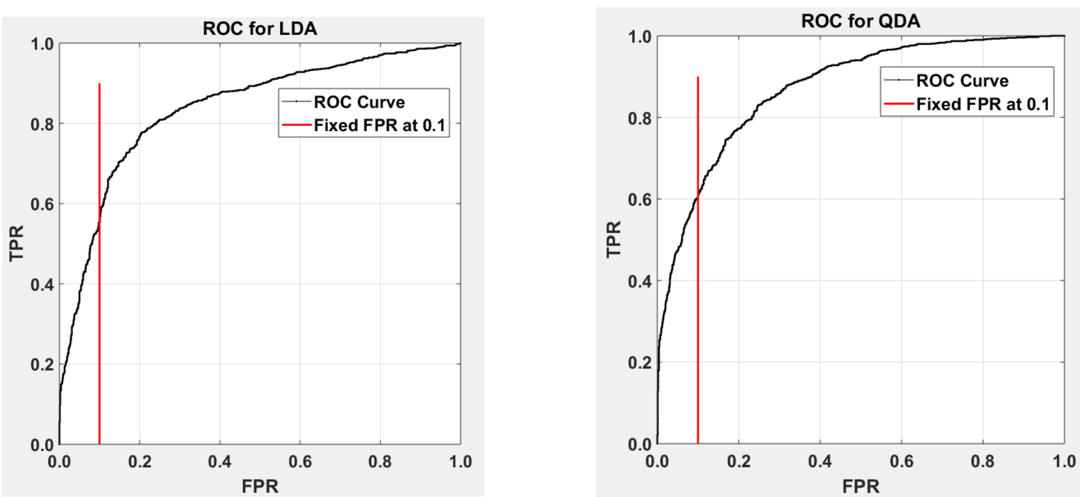

Figure 1.4.5: ROC curve for (left) LDA and (right) QDA for Case IV

Table 9: Confusion matrix for QDA for Case IV

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 542	False Positive (FP) 72	$PPV = \frac{TP}{TP+FP}$ $= 88.27\ \%$
Test Outcome -1	False Negative (FN) 458	True Negative (TN) 928	$NPV = \frac{TN}{TN+FN}$ $= 66.96\ \%$
	$Sensitivity = \frac{TP}{TP+FN}$ $= 54.2\ \%$	$Specificity = \frac{TN}{TN+FP}$ $= 92.8\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 73.5\%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 67.2\ \%$	
Area Under ROC (AUC)		0.87081	

Conclusion of 1.4 (a): To conclude above all the discussion, lets summarize all the table (Table 1 to Table 9) into one table that will be easing to describe the findings of all above analysis. From the Table 10, it can be concluded for all the cases, Liner Discriminator Analysis (LDA) provides better performance than Quadratic Discriminator Analysis (QDA) if we consider the accuracy of the correct detections. Or, if we consider the Area Under ROC curve, then QDA is better than LDA. The average True Positive rate (TPR) for LDA is **0.57** and QDA is **0.62** both are w.r.t. **0.1** False Positive rate (FPR). On the other hand, for all the performance parameters, case IV provides better results of the classifier that's mean that if we take first **25%** of data as a Testing set and later **75%** as Training set.

Table 10: Summarization of table of all above discussion

Performance Parameters	**Case I**		**Case II**		**Case III**		**Case IV**	
	LDA	**QDA**	**LDA**	**QDA**	**LDA**	**QDA**	**LDA**	**QDA**
Accuracy (%)	76.75	72.60	75.90	73.00	76.35	72.45	77.40	73.5
F1 Score (%)	74.91	65.14	73.77	66.80	74.22	65.06	75.35	67.20
AUC (%)	83.08	86.07	83.76	87.42	84.08	87.31	83.69	87.08
Sensitivity (%)	69.40	51.30	67.80	54.40	68.10	51.3	69.10	54.20
Specificity (%)	84.10	93.80	84.00	91.60	84.60	93.60	85.70	92.80
PPV (%)	81.40	89.20	80.90	86.60	81.55	88.90	82.85	88.27
NPV (%)	73.30	65.80	72.30	66.80	72.60	65.77	73.85	66.97
TPR at 0.1 FPR	0.58	0.61	0.52	0.64	0.57	0.63	0.59	0.62

N.B: For understanding Case I, Case II, Case III and Case IV, please see the Table 1.

Solution of 1.4 (b): In this question, I will implement k-NN classifier using the similar cases in previous discussions, but I will vary the K value and find out the effects of K on the classifier accuracy.

Case I: In this case, Training Data will be Group: A, B, and C and Testing data will be Group: D from each of the class (+1, -1). Confusion matrix with some performance are given below in a Table 11. From the Figure 1.4.6 (left), it is seen that among all the values of k, k=13 provides better results. This is the reason of using k=13 in Table 11. From Figure 1.4.6 (right), it is also noticeable that for the **0.1** FPR, we are getting **0.72** as a TPR. And Area Under ROC curve (AUC) is **0.889**.

Table 11: Confusion matrix for k-NN using K= 13 for Case I

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 728	False Positive (FP) 103	$PPV = \frac{TP}{TP + FP}$ = 87.61 %
Test Outcome -1	False Negative (FN) 272	True Negative (TN) 897	$NPV = \frac{TN}{TN + FN}$ = 76.73 %
	$Sensitivity = \frac{TP}{TP + FN}$ = 72.8 %	$Specificity = \frac{TN}{TN + FP}$ = 89.7 %	
Accuracy		$\frac{TP + TN}{TP + FP + TN + FN} = 81.25\%$	
F1 Score		$\frac{2TP}{2TP + FP + FN} = 79.51\ \%$	
False Positive Rate		$\frac{FP}{TN + FP} = 10.3\ \%$	
False Negative Rate		$\frac{FN}{TP + FN} = 27.2\ \%$	

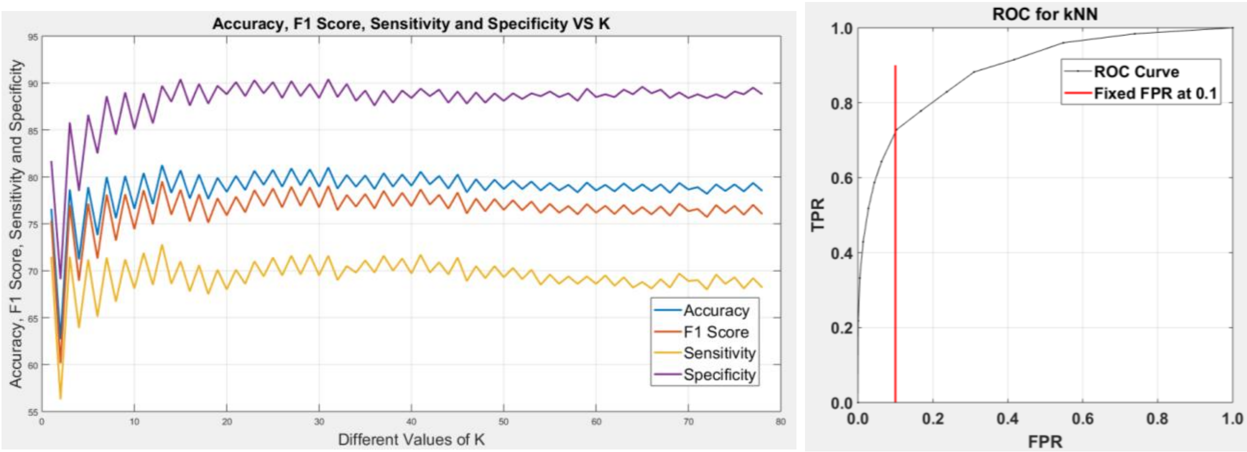

Figure 1.4.6: Performance parameters w.r.t various k values (left) and ROC curve (right) in case I

Case II: In this case, Training Data will be Group: A, B, and D and Testing data will be Group: C from each of the class (+1, -1). Confusion matrix with some performance are given below in a Table 12. From the Figure 1.4.7 (left), it is seen that among all the values of k, k=17 provides better results. This is the reason of using k=17 in Table 12. From Figure 1.4.7 (right), it is also noticeable that for the **0.1** FPR, we are getting **0.70** as a TPR. And Area Under ROC curve (AUC) is **0.892**.

Table 12: Confusion matrix for k-NN using K= 17 for Case II

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 748	False Positive (FP) 118	$PPV = \frac{TP}{TP+FP} = 86.37\ \%$
Test Outcome -1	False Negative (FN) 252	True Negative (TN) 882	$NPV = \frac{TN}{TN+FN} = 77.78\ \%$
	$Sensitivity = \frac{TP}{TP+FN} = 74.8\ \%$	$Specificity = \frac{TN}{TN+FP} = 88.2\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 81.5\%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 80.17\ \%$	
False Positive Rate		$\frac{FP}{TN+FP} = 11.8\ \%$	
False Negative Rate		$\frac{FN}{TP+FN} = 25.2\ \%$	

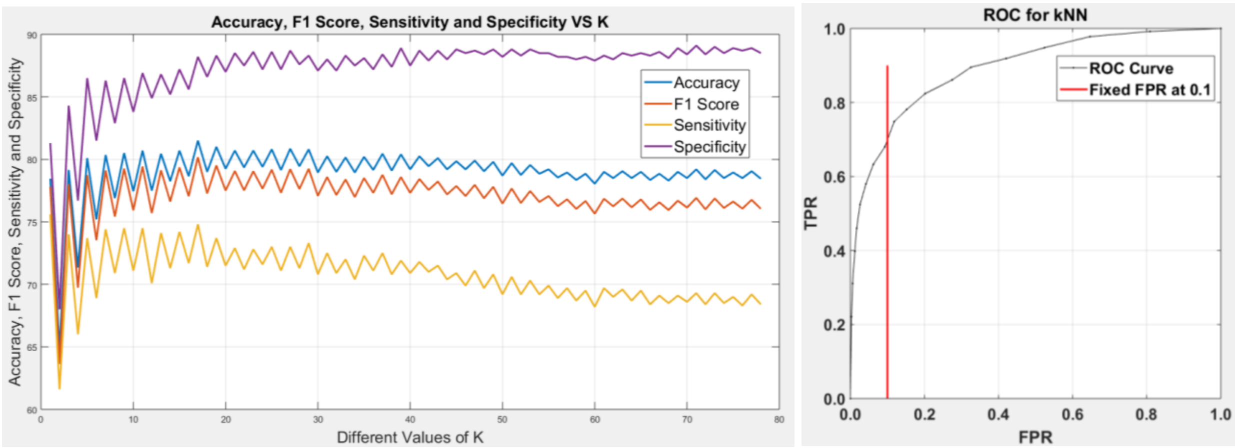

Figure 1.4.7: Performance parameters w.r.t various k values (left) and ROC curve (right) in case II

Case III: In this case, Training Data will be Group: A, C, and D and Testing data will be Group: B from each of the class (+1, -1). Confusion matrix with some performance are given below in a Table 13. From the Figure 1.4.8 (left), it is seen that among all the values of k, k=15 provides better results. This is the reason of using k=15 in Table 13. From Figure 1.4.8 (right), it is also noticeable that for the **0.1** FPR, we are getting **0.71** as a TPR. And Area Under ROC curve (AUC) is **0.915.**

Table 13: Confusion matrix for k-NN using K= 15 for Case III

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 727	False Positive (FP) 110	$PPV = \frac{TP}{TP+FP} = 86.86\ \%$
Test Outcome -1	False Negative (FN) 273	True Negative (TN) 890	$NPV = \frac{TN}{TN+FN} = 76.53\%$
	$Sensitivity = \frac{TP}{TP+FN} = 72.7\ \%$	$Specificity = \frac{TN}{TN+FP} = 89\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 80.85\ \%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 79.15\ \%$	
False Positive Rate		$\frac{FP}{TN+FP} = 11\ \%$	
False Negative Rate		$\frac{FN}{TP+FN} = 27.3\ \%$	

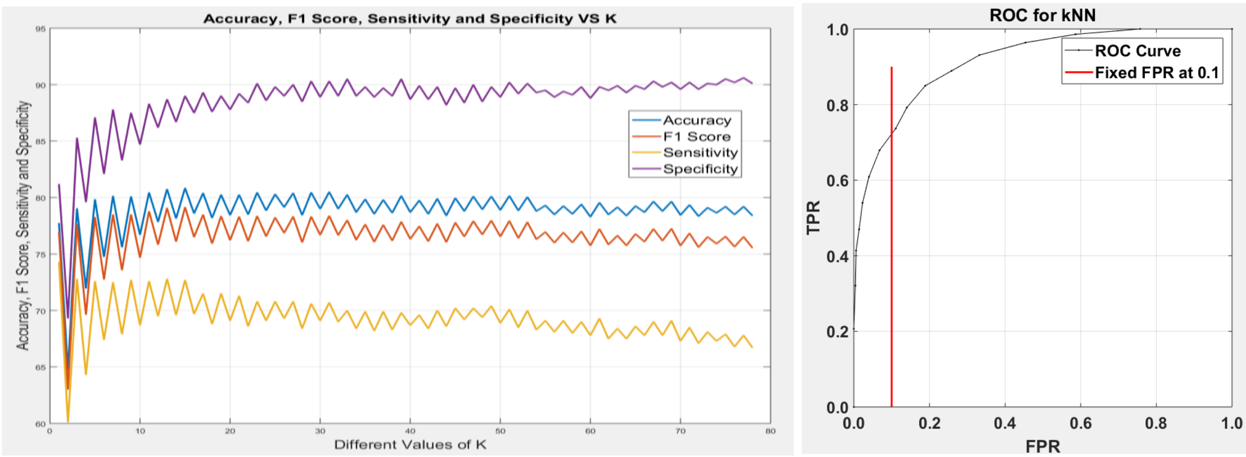

Figure 1.4.8: Performance parameters w.r.t various k values (left) and ROC curve (right) in case III

Case IV: In this case, Training Data will be Group: B, C, and D and Testing data will be Group: A from each of the class (+1, -1). Confusion matrix with some performance are given below in a Table 14. From the Figure 1.4.9 (left), it is seen that among all the values of k, k=17 provides better results. This is the reason of using k=17 in Table 14. From Figure 1.4.9 (right), it is also noticeable that for the **0.1** FPR, we are getting **0.70** as a TPR. And Area Under ROC curve (AUC) is **0.892**.

Table 14: Confusion matrix for k-NN using K= 17 for Case IV

	Test data		
	Class +1	**Class -1**	
Test Outcome +1	True Positive (TP) 724	False Positive (FP) 110	$PPV = \frac{TP}{TP+FP} = 86.81\ \%$
Test Outcome -1	False Negative (FN) 276	True Negative (TN) 890	$NPV = \frac{TN}{TN+FN} = 76.33\ \%$
	$Sensitivity = \frac{TP}{TP+FN} = 72.4\ \%$	$Specificity = \frac{TN}{TN+FP} = 89\ \%$	
Accuracy		$\frac{TP+TN}{TP+FP+TN+FN} = 80.7\ \%$	
F1 Score		$\frac{2TP}{2TP+FP+FN} = 78.95\ \%$	
False Positive Rate		$\frac{FP}{TN+FP} = 11\ \%$	
False Negative Rate		$\frac{FN}{TP+FN} = 27.6\ \%$	

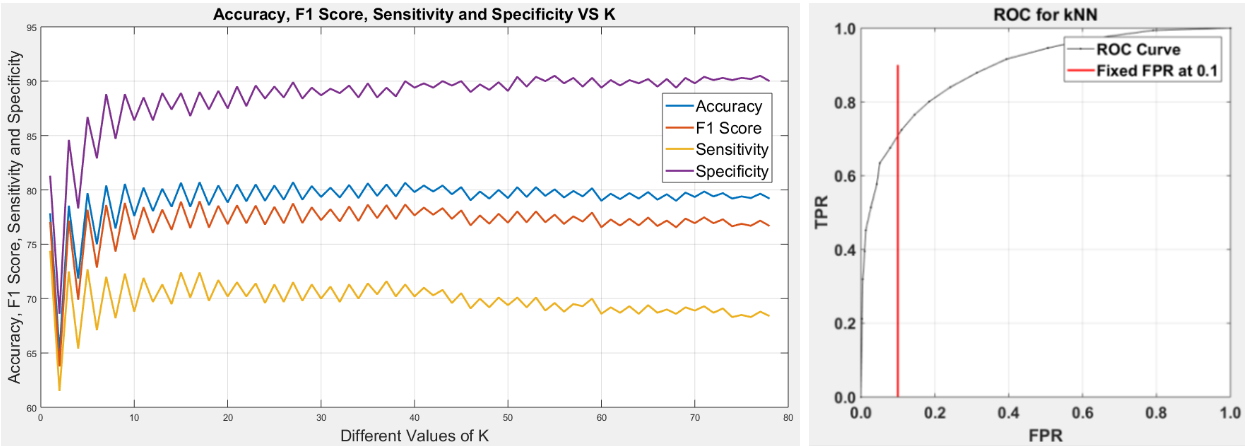

Figure 1.4.9: Performance parameters w.r.t various k values (left) and ROC curve (right) for Case IV

Conclusion of 1.4 (b): From the above all the discussion regarding the k-NN classifier, I can conclude that the performance parameters (e.g. accuracy, sensitivity, specificity, PPV, NPV, FPR, FNR and F1 score) of the k-NN classifier is better than LDA and QDA for given set of data. From the ROC, analysis of the k-NN classifier, I can claim the in all the cases (I, II, III and IV) the avg. TPR (**0.71**) at 0.1 FPR and AUC (**0.89**) are more than the LDA and QDA. On the other hand, comparing all the cases (I, II, III and IV), Case II provides best results w.r.t. all the performance parameters that means that if we take first 50 % and last 25% as a Training (75 %) and remaining 25 % as a testing. The MATLAB source code for this problem is named as "*Problem1_4_b_kNN.m*" and available in the folder that contain this report. Here, it is mentionable that this code will takes more time to run because it will calculate the performance parameters (e.g. accuracy, sensitivity, specificity, PPV, NPV, FPR, FNR and F1 score) for lots of values of K and plot these parameters w.r.t. K for the comparison between different values of K. If there is only one K value it will take only few seconds.

Problem 1.5 [30%] Use the dataset of Problem 1.4 and, using a model among the ones you know (linear, quadratic, *k - NN*), build the classifier with the best accuracy on that classification problem. To this aim, you can consider several training/test splits with different percentages for training and test sets. Prepare a Matlab fuction called test.m (function y=test(A)) that implements your best choice classifier. The function will accept a matrix A and return a vector y having the same number of rows in A. The matrix A will contain several samples (one for each row) organized in the same way described in Problem 1.4, but without label (i.e. each row will contain containing a sample with 10 numerical features). The function will classify each of the samples A (i,:), providing the predicted class in y (i). Your function must be submitted and will be run on a separate matrix containing new test data. Your grade will be based on the performance of your classifier on the new test data, which will contain a very large number of examples generated from the same distribution.

***Solution of 1.5 (b)*:** After analyzing the performance of all the classifiers in Problem 1.4, I have decided to implement k-NN classifier for its better performance and simple manner of algorithm.

To access this code, anyone need to follow the block diagram as shown in Figure 1.5.1.

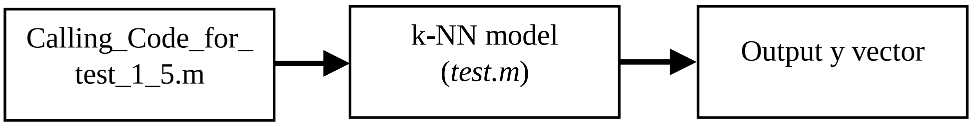

Figure 1.5.1: Block presentation of Problem 1.5

In the 1st block, you need to run this code that will call the k-NN models named as *"test.m"* with the arguments that you want to classify. Here, it is mentionable that your argument matrix should have following dimension Eq. 1.5.1. And have the same distribution of the training data that means that same standard deviations and mean values. Suppose your argument matrix is A.

$$A = \begin{bmatrix} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{bmatrix}_{row\times 10} \qquad (1.5.1)$$

Which means that matrix A may have any numbers of rows but 10 columns only. After this, your *"test.m"* will return y like Eq. 1.5.2.

$$y = [\qquad\qquad]_{row\times 1} \qquad (1.5.2)$$

In y, each row indicates the class labels (either +1 or -1) corresponding to each row of A. If you know the row labels of the A, then you can easily calculate the accuracy of the classifier.

N.B. The training data named *"hw1data.mat"*, k-NN model (*test.m*) and *Calling_Code_for_test_1_5.m* should be in the same directory.

Attachments with this report:

1. Source data named as *"hw1data.mat"*
2. *MATLAB* Source code for 1.2 (a) named as *"Problem1_2_a_DesnsityFunction.m"*
3. *MATLAB* Source code for 1.2 (b) named as *"Problem1_2_b_DecisionRegion.m"*
4. *MATLAB* Source code for 1.3 named as *"Problem1_3_DecisionRegion.m"*
5. *MATLAB* Source code for 1.4 (a) named as *"Problem1_4_a_LDA_QDA.m"*
6. *MATLAB* Source code for 1.4 (a) named as *"EvalROC.m"*
7. *MATLAB* Source code for 1.4 (a) named as *"RChPlot.m"*
8. *MATLAB* Source code for 1.4 (b) named as *"Problem1_4_b_kNN.m"*
9. *MATLAB* Source code for 1.5 named as *"test.m"*
10. *MATLAB* Source code for 1.5 named as *"Calling_Code_for_test_1_5.m"*