2.
a)

```
#################    Farhad  Mohammad Kazemi   ###########
##################       Assignment 2.2       #############
library(pROC)
library(MASS)
library(caret)
library(ROCR)

set.seed(1234)
mydat <- read.table("c:/Users/fmkaz/Desktop/MUN/Dr lourdes_ting hu/assignments/2/2_2/
A2_T2.tab", header = TRUE, stringsAsFactors= FALSE)

mydat$Class <- factor(mydat$Class, levels = c("0","1"))

#                          Randomly permute the data before subsetting

mydat_idx <- sample(1:nrow(mydat), replace = FALSE)
mydat <- mydat[mydat_idx, ]

#                          Produce Training and testing data
#              separate 10% of the instances in the data as a validation set
mydat_resampled_idx <- createDataPartition(mydat_idx, times = 1, p = 0.9, list = FALSE)
mydat_resampled <- mydat[mydat_resampled_idx, ] # Training portion of the data

mydat_resampled_test <- mydat[-mydat_resampled_idx, ] # Test portion

#                           Linear discriminant analysis

y = as.factor(mydat_resampled[, 181])
x = mydat_resampled[, 1:180]

#                   apply  LDA to the remaining 90% of instances.
lda_mod <-train(x , y,method = "lda")


#                Generate model predictions for Training data
lda_pred <- predict(lda_mod, newdata = mydat[ , 1:180], type = "prob")

#                Generate model predictions for Testing data
lda_predt <- predict(lda_mod, newdata = mydat[ -mydat_resampled_idx,1:180], type = "p
rob")


mydatnew=mydat[-mydat_resampled_idx, ]

#                 Store the predictions with the data set
mydat['lda_pred'] <- lda_pred["1"] # Here we only want the probability associated with the
class (Y = 1),
mydatnew['lda_predt'] <- lda_predt["1"]
```

```
#############      Produced confusion matrix for testing data using LDA
contrasts ( mydatnew$Class )
ldapred=rep ("0" ,316)
ldapred[mydatnew$lda_predt >.5]=" 1"

table(ldapred,mydatnew$Class )



#                                 Ligistic Regression

#                      apply  Ligistic Regression to the remaining 90% of instances.
logit_mod <- glm(formula=Class ~ ., data=mydat_resampled[,1:181], family="binomial")

#                     Generate model predictions for Training data
logit_pred <- predict.glm(logit_mod, newdata=mydat[ , 1:180], type = "response")

#                     Generate model predictions for Testing data
logit_predt <- predict.glm(logit_mod, newdata=mydat[ -mydat_resampled_idx, 1:180], type = "response")

#                     Store the predictions with the data set
mydat['logit_pred'] <- logit_pred#["1"] # Here we only want the probability associated
mydatnew['logit_predt'] <- logit_predt#["1"]


#############  Produced confusion matrix for testing data using Ligistic Regression
contrasts ( mydatnew$Class )
logitpred=rep ("0" ,316)
logitpred[mydatnew$logit_predt >.5]=" 1"

table(logitpred,mydatnew$Class )



#ROC curves on the training and testing portions of the data using LDA and Logistic Regression

#                          calculation and plots of AUC curves
#                                 For Training data
mydat$binary_response <- as.numeric(mydat$Class) - 1 # convert0 factor to 0, 1 with 1
lda_train_roc <- roc(binary_response ~ lda_pred, data = mydat[mydat_resampled_idx, ], ci = TRUE)
logit_train_roc <- roc(binary_response ~ logit_pred, data =  mydat[mydat_resampled_idx, ], ci = TRUE)

#                          calculation and plots of AUC curves
#                                 For Testing data
mydatnew$binary_response <- as.numeric(mydatnew$Class) - 1 # convert0 factor to 0, 1 with 1
lda_test_roc <- roc(binary_response ~ lda_predt, data =  mydatnew, ci = TRUE)
logit_test_roc <- roc(binary_response ~ logit_predt, data =  mydatnew, ci = TRUE)
```

```
#                              Plot ROC curve for Training data
plot(lda_train_roc, las = 1, main = "ROC curve for Training data")
plot(logit_train_roc, add = TRUE, col = "red")
legend(0.4, 0.4, legend = c("LDA", "Logistic Regression"), lty = c(1,1,1), col = c("black", "r
ed"))

##          Plot precision/recall curve (x-axis: recall, y-axis: precision) for training data
lda_preddd <- prediction( mydat$lda_pred, mydat$Class)
logit_preddd <- prediction( mydat$logit_pred, mydat$Class)

perflda <- performance(lda_preddd, "prec", "rec")
plot(perflda, las = 1, main = "precision/recall curve for Training data")

perflogit <- performance(logit_preddd, "prec", "rec")
plot(perflogit, add = TRUE, col = "red")
legend(0.1,0.7, legend = c("LDA", "Logistic Regression"), lty = c(1,1,1), col = c("black", "re
d"))



#                              Plot ROC curve for Testing data
plot(lda_test_roc, las = 1, main = "ROC curve for Testing data")
plot(logit_test_roc, add = TRUE, col = "red")
legend(0.4, 0.4, legend = c("LDA", "Logistic Regression"), lty = c(1,1,1), col =     c("black",
"red"))

##         Plot precision/recall curve (x-axis: recall, y-axis: precision) for testing data
lda_predd <- prediction( mydatnew$lda_predt, mydatnew$Class)
logit_predd <- prediction( mydatnew$logit_predt, mydatnew$Class)

perftlda <- performance(lda_predd, "prec", "rec")
plot(perftlda, las = 1, main = "precision/recall curve for Testing data")

perftlogit <- performance(logit_predd, "prec", "rec")
plot(perftlogit, add = TRUE, col = "red")
legend(0.1,0.7, legend = c("LDA", "Logistic Regression"), lty = c(1,1,1), col = c("black", "re
d"))
# AUC
lda_train_roc$ci[c(2, 1, 3)] #

logit_train_roc$ci[c(2, 1, 3)] #

lda_test_roc$ci[c(2, 1, 3)] #

logit_test_roc$ci[c(2, 1, 3)] #
```

2.b and 2.c)

First of all, one of the most important issues regarding subsetting our data into training and testing subsets is prior to subsetting, the data have to be randomized otherwise we will have unequal division of our categories in the training and testing data subsets. So, I considered this subject. For details refer to source code in 2.a.

-optimal probability threshold=0.5
The class prediction is based on a 50% probability cutoff.

```
######      Produced confusion matrix for testing data using LDA
contrasts ( mydatnew$Class )
ldapred=rep ("0" ,316)
ldapred[mydatnew$lda_predt >.5]=" 1"

table(ldapred,mydatnew$Class )
```

```
Produced confusion matrix for test data(10%) using LDA


ldapred      0    1
          0  147  5
          1  15   149



Produced confusion matrix for test data(10%) using Logistic Regression

logitpred      0   1
            0  145 7
            1  17  147
```

***Actually I checked prediction of classifiers for the first time through train data (90%) and after that by using the test dat a(10%) . For detail you can follow report and source code.**

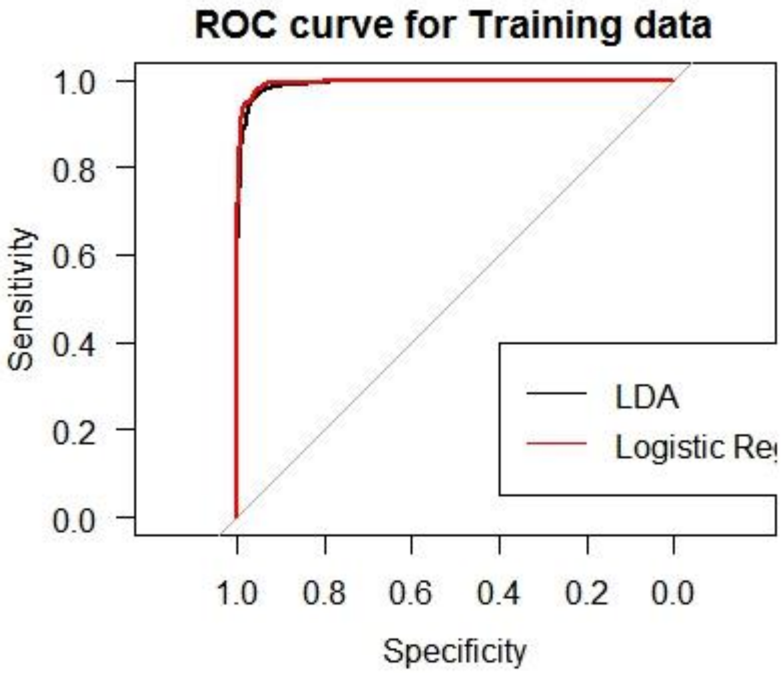
-ROC curve if we suppose train data as a test data

```
plot(lda_train_roc, las = 1, main = "ROC curve for Training data")
```

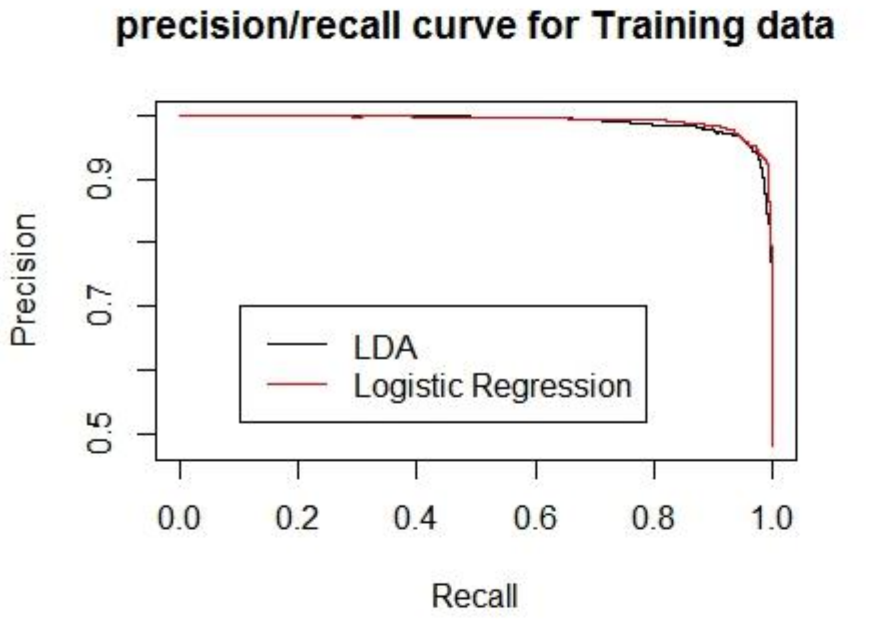
```
AUC for train data using LDA
Area under the ROC curve: 0.9912
95% CI: 0.9887-0.9938 (DeLong)
```

```
plot(logit_train_roc, add = TRUE, col = "red")
```

```
AUC for train data using Logistic Regression
Area under the curve: 0.9946
95% CI: 0.9928-0.9963 (DeLong)
```



-Precision/ recall curve for train data as a test data



-ROC curve if we suppose test data as a test data

```
plot(lda_test_roc, las = 1, main = "ROC curve for Testing data")
```

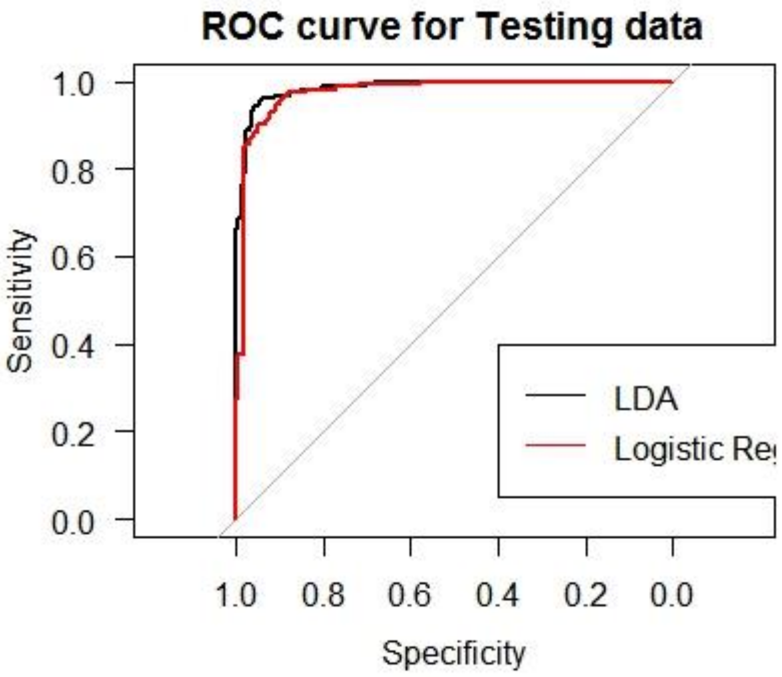
AUC for test data using LDA

```
Area under the curve: 0.9854
95% CI: 0.9758-0.995 (DeLong)
```

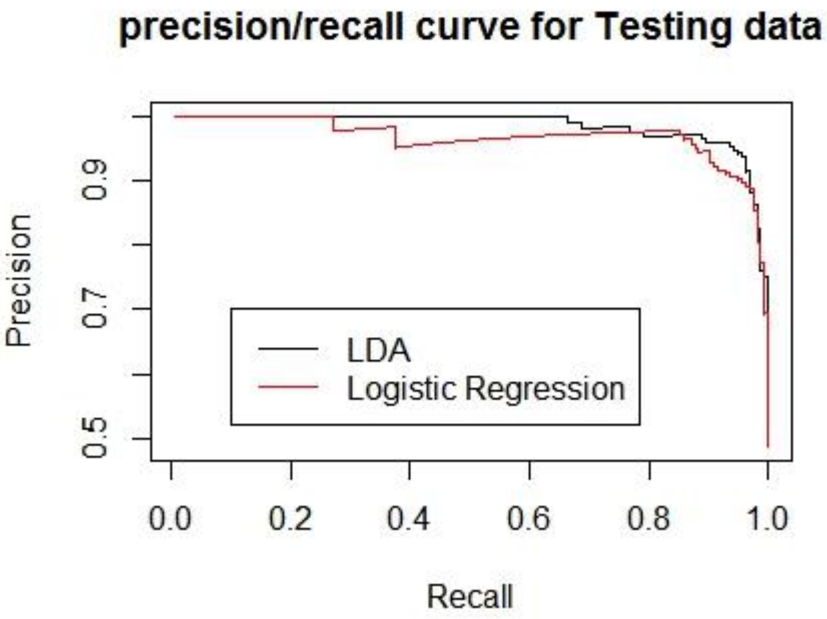
```
plot(logit_test_roc, add = TRUE, col = "red")
```

AUC for test data using Logistic Regression

```
Area under the curve: 0.9749
95% CI: 0.9588-0.9911 (DeLong)
```



-Precision/ recall curve for test data as a test data



```
> # AUC
> lda_train_roc$ci[c(2, 1, 3)] #      AUC for train data using LDA

[1] 0.9912428 0.9886854 0.9938003

> logit_train_roc$ci[c(2, 1, 3)] #  AUC for train data using Logistic Regression
[1] 0.9945649 0.9928170 0.9963127

> lda_test_roc$ci[c(2, 1, 3)] #      AUC for test data using LDA
[1] 0.9854097 0.9757734 0.9950459

> logit_test_roc$ci[c(2, 1, 3)] #    AUC for test data using Logistic Regression
[1] 0.9749479 0.9587938 0.9911019
```

According to the reached results of test data (10% of all instances), ROC and precision/recall curves and AUC criteria (0.9854097), we can clearly see that LDA is better in comparison to Logistic Regression for this dataset.