Compulsory exercise 1: Group 12

Evgeni Vershinin, Kristina Ødegård

2024-02-08

**Problem 1
a)
Quantative: Time, income earned, horsepower of a car. Qvalitative: Marital status, origin, Gender.
b) KNN, LDA, QDA can be used for multi-class classifications.
c)
d)The nereast neighbour for k=1 is a blue dot, so our classification is blue. For K=3, two of the nearest neighbors are red and 1 blue. This gives 2/3 red, so it is red. For K=5 we have 3/5 red, so it is red.

```
data(Boston)
data = Boston

model = lm(medv ~ rm + age,data=data)
summary(model)
```

```
##
## Call:
## lm(formula = medv ~ rm + age, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.555  -2.882  -0.274   2.293  40.799
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -25.27740    2.85676  -8.848  < 2e-16 ***
## rm            8.40158    0.41208  20.388  < 2e-16 ***
## age          -0.07278    0.01029  -7.075 5.02e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.316 on 503 degrees of freedom
## Multiple R-squared:  0.5303, Adjusted R-squared:  0.5284
## F-statistic: 283.9 on 2 and 503 DF,  p-value: < 2.2e-16
```

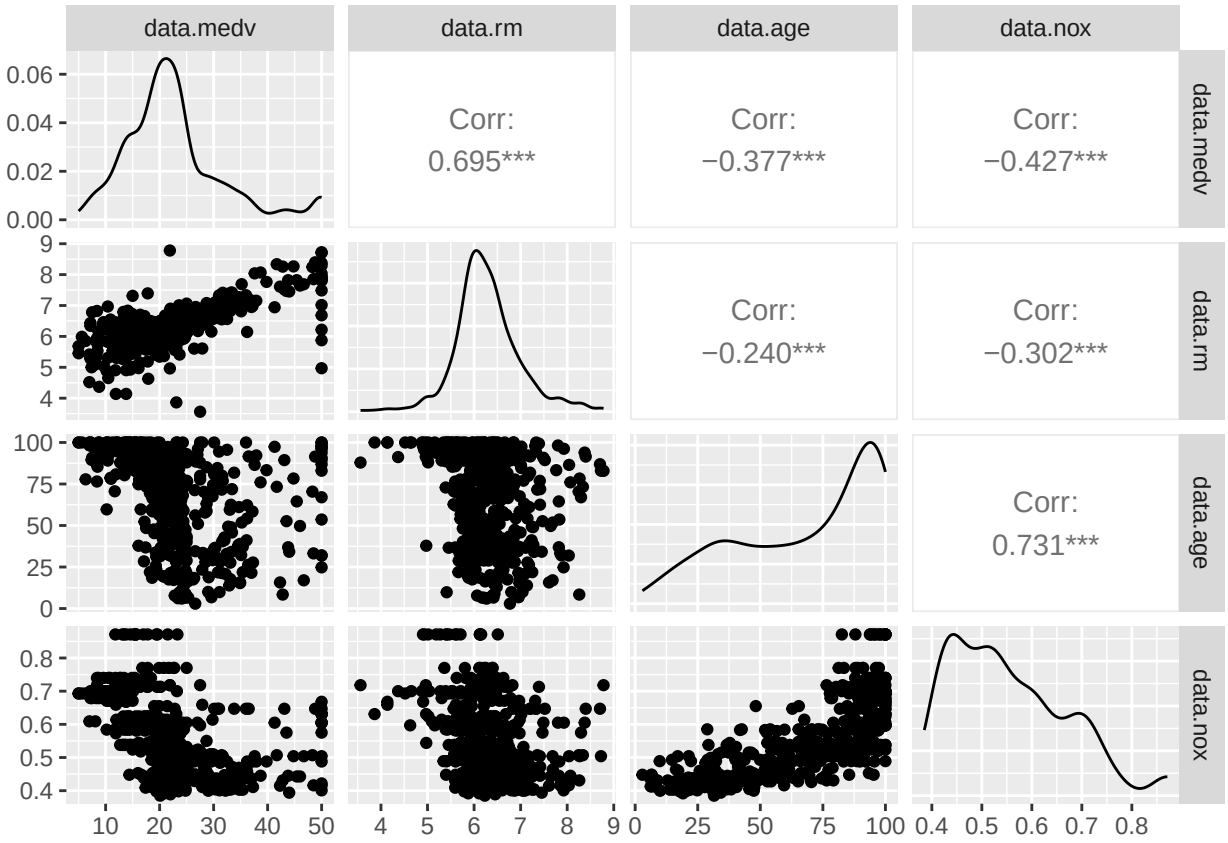
```
cor_matrix = cor(data.frame(data$medv, data$rm, data$age))
print(cor_matrix)
```

```
##            data.medv    data.rm   data.age
## data.medv  1.0000000  0.6953599 -0.3769546
## data.rm    0.6953599  1.0000000 -0.2402649
## data.age  -0.3769546 -0.2402649  1.0000000
```

```r
model2 = lm(medv ~ rm + age + nox, data=data)
summary(model2)
```

```
##
## Call:
## lm(formula = medv ~ rm + age + nox, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.343  -3.168  -0.539   2.221  40.260
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.08308    3.33919  -5.715 1.88e-08 ***
## rm            8.12542    0.41525  19.568  < 2e-16 ***
## age          -0.03686    0.01449  -2.544 0.011269 *
## nox         -12.47877    3.58434  -3.481 0.000542 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.247 on 502 degrees of freedom
## Multiple R-squared:  0.5413, Adjusted R-squared:  0.5386
## F-statistic: 197.5 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
ggpairs(data.frame(data$medv, data$rm, data$age, data$nox))
```



e) IV Looking at the correlation between Age and NOX, it is 0.731, which is quite high which suggest it has multicollinearity, which means they give the some of the same information for the model.

*__Problem 2__ a)

```
model3 = lm(medv ~ poly(rm, 2) + I(age*crim) + age + crim, data=data)
summary(model3)
```

```
##
## Call:
## lm(formula = medv ~ poly(rm, 2) + I(age * crim) + age + crim,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.314  -2.602  -0.369   2.136  35.081
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    28.096141   0.666283  42.169  < 2e-16 ***
## poly(rm, 2)1  123.366564   5.600727  22.027  < 2e-16 ***
## poly(rm, 2)2   64.836354   5.479728  11.832  < 2e-16 ***
## I(age * crim)   0.005792   0.003553   1.630   0.1037
## age            -0.067283   0.009342  -7.202 2.19e-12 ***
## crim           -0.796544   0.338946  -2.350   0.0192 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.369 on 500 degrees of freedom
## Multiple R-squared:  0.6626, Adjusted R-squared:  0.6592
## F-statistic: 196.3 on 5 and 500 DF,  p-value: < 2.2e-16
```

```
valuechanged = (-10*(-0.796544)+60*(-0.067283)+0.005792 * (-10 + 60))*1000
```

If the crime is reduced by 10 and age is 60, and then considering all other factor keeps equal our median value of the property is changed by 4218.06.‘

**b) First thing we could do is to plot all our data with say ggpairs from GGAlly. We can then look how the data is spread and correlated. We want to carefully identify if we have any outliers. If most our points follow a specific pattern, but then one point stick out, we could carefully consider to remove it. One approach could be to look at the mean and std of a given data columns and if some point lie 2 standard deviations away from the mean we could remove them.

**c)

```
model4 <- lm(medv ~ crim + age + rm, data = Boston)
summary(model4)
```

```
##
## Call:
## lm(formula = medv ~ crim + age + rm, data = Boston)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -19.959  -3.143  -0.633   2.150  39.940
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.60556    2.76938  -8.524  < 2e-16 ***
## crim         -0.21102    0.03407  -6.195 1.22e-09 ***
## age          -0.05224    0.01046  -4.993 8.21e-07 ***
## rm            8.03284    0.40201  19.982  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.094 on 502 degrees of freedom
## Multiple R-squared:  0.5636, Adjusted R-squared:  0.561
## F-statistic: 216.1 on 3 and 502 DF,  p-value: < 2.2e-16
```

**i) A t-value is calculated with the follow formula: $t = \frac{Coefficient\ Estimate}{Standard\ Error} = \frac{10}{-0.052}$

The t-value for rm if the estimated coefficient was 10, but with the same standard error is 24.89.

The t-value is a measure of how many standard errors the estimated coefficient is from zero. This indicates how strong effect the variable has in the response variable. That is, how significant it is.

When the coefficient estimate is increased, taking it further away from zero, it increases the t-value, making it more statistically significant.

**ii)

```
f_test <-anova(model4)
print(f_test)
```

```
## Analysis of Variance Table
##
## Response: medv
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## crim        1  6440.8  6440.8 173.458 < 2.2e-16 ***
## age         1  2809.8  2809.8  75.671 < 2.2e-16 ***
## rm          1 14825.7 14825.7 399.275 < 2.2e-16 ***
## Residuals 502 18640.0    37.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

An F-test was utilized to decide if at least one of the predictors is useful in predicting the response.

In this case, all three predictors has a p-value associated with the F-statisic that is $< 2.2^{-16}$. Since the p-value is essentially zero, we have strong evidence to reject the null hypothesis, which would be that none of the predictors were useful in predicting the response variable. Thus, at least one of the predictors is useful in predicting the response.

**iii)

```
model5 <- lm(medv ~ crim + age, data = Boston)
f_test_2 <-anova(model5)
print(f_test_2)
```

```
## Analysis of Variance Table
##
## Response: medv
##            Df Sum Sq Mean Sq F value    Pr(>F)
## crim        1   6441  6440.8  96.807 < 2.2e-16 ***
## age         1   2810  2809.8  42.232 1.954e-10 ***
## Residuals 503  33466    66.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value associated with F values, though increased for age in this model without the predictor rm, are still significantly lower than the typical significance level of 0.05. Thus the null hypothesis can be rejected for both predictors. We conclude that the model is still useful.

```
new_data <- data.frame(crim = 10, age = 90, rm = 5)
# Predict the response variable using the model for the confidence interval
prediction_c <- predict(model4, newdata = new_data, interval = "confidence", level = 0.99)

# Extract lower and upper bounds from the confidence interval prediction
lower_bound_c <- prediction_c[1]
upper_bound_c <- prediction_c[2]

# Print the results for the confidence interval
cat("Lower bound of 99% confidence interval:", lower_bound_c, "\n")
```

```
## Lower bound of 99% confidence interval: 9.746544
```

```
cat("Upper bound of 99% confidence interval:", upper_bound_c, "\n\n")
```

```
## Upper bound of 99% confidence interval: 8.25632
```

```
# Predict the response variable using the model for the prediction interval
prediction_p <- predict(model4, newdata = new_data, interval = "prediction", level = 0.99)

# Extract lower and upper bounds from the prediction interval prediction
lower_bound_p <- prediction_p[1]
upper_bound_p <- prediction_p[2]

# Print the results for the prediction interval
cat("Lower bound of 99% prediction interval:", lower_bound_p, "\n")
```

```
## Lower bound of 99% prediction interval: 9.746544
```

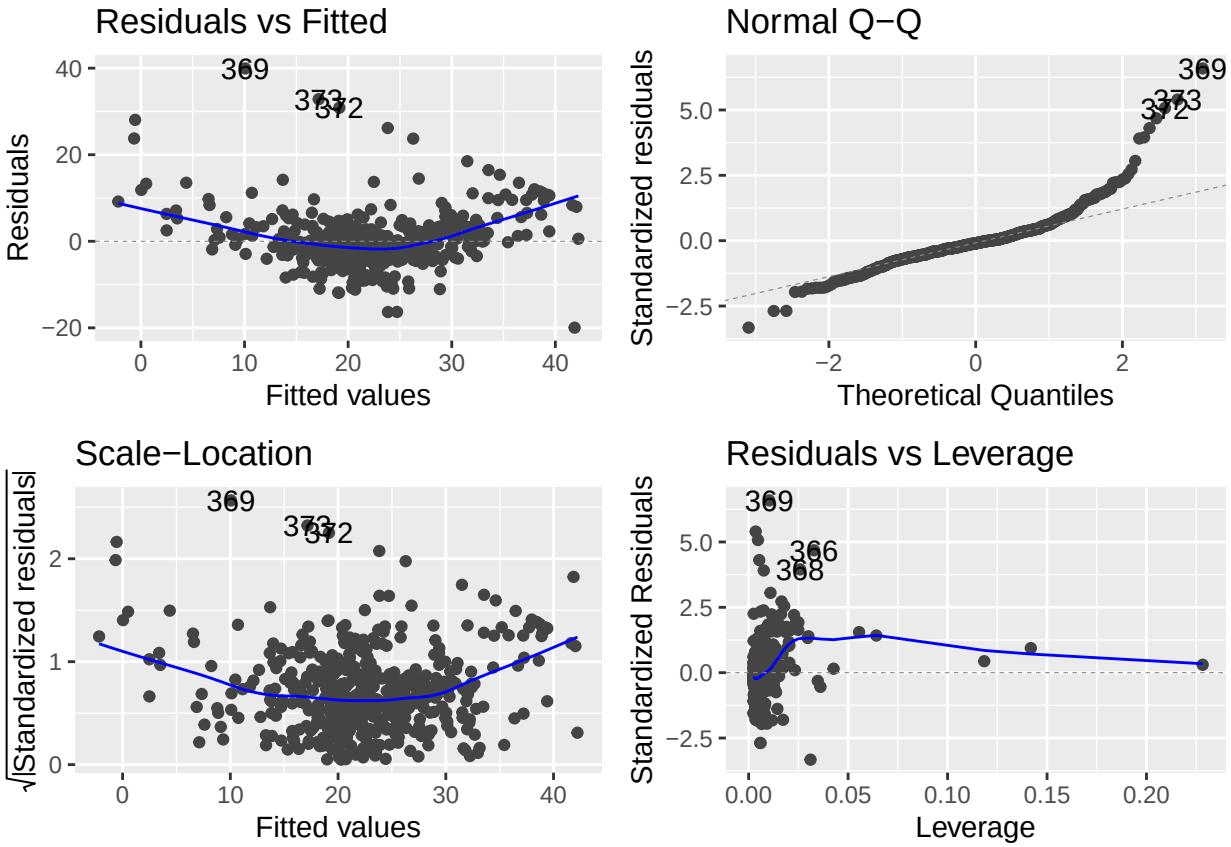
```
cat("Upper bound of 99% prediction interval:", upper_bound_p, "\n")
```

```
## Upper bound of 99% prediction interval: -6.079652
```

**iii) The difference between confidence interval and prediction interval is that the confidence interval estimates the range in which the population mean response most likely will lie, and the prediction interval estimates the range in which a future individual observation most likely will fall within.

**iv)

```r
# Create diagnostic plots
autoplot(model4, which = c(1, 2, 3, 5))
```



e) i) The gender is a binary qualitative variable. The student has used two binary variables to describe what only needs one. $y = \beta_0 + \beta_1 x_{gender} + \varepsilon$ $x_{gender} = \begin{cases} 1 & if\ male \\ 0 & if\ female \end{cases}$

***Problem 3 a)

```r
data("titanic_train")
vars_to_be_removed <- c("PassengerId", "Name", "Ticket", "Cabin", "Embarked")
View(titanic_train)
titanic_train <- titanic_train[, -which(names(titanic_train) %in% vars_to_be_removed)]
titanic_train$Pclass <- as.factor(titanic_train$Pclass)
train_idx <- sample(1:nrow(titanic_train), 0.8 * nrow(titanic_train))
titanic_test <- titanic_train[-train_idx, ]
titanic_train <- titanic_train[train_idx, ]
titanic_train <- na.omit(titanic_train)
titanic_test <- na.omit(titanic_test)
logReg = glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare, data = titanic_train, family = "bino
summary(logReg)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare, family = "binomial", data = titanic_train)
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.048694   0.553988   7.308 2.71e-13 ***
## Pclass2     -1.230152   0.353680  -3.478 0.000505 ***
## Pclass3     -2.323032   0.368839  -6.298 3.01e-10 ***
## Sexmale     -2.675098   0.248332 -10.772  < 2e-16 ***
## Age         -0.042726   0.009268  -4.610 4.02e-06 ***
## SibSp       -0.420064   0.142759  -2.942 0.003256 **
## Parch       -0.099062   0.131840  -0.751 0.452422
## Fare         0.002732   0.002796   0.977 0.328516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 763.63  on 564  degrees of freedom
## Residual deviance: 508.23  on 557  degrees of freedom
## AIC: 524.23
##
## Number of Fisher Scoring iterations: 5
```

```
predicted_probabilities <- predict(logReg, newdata = titanic_test, type = "response")
predicted_labels <- ifelse(predicted_probabilities > 0.5, 1, 0)
actual_labels <- titanic_test$Survived
misclassification_error <- mean(predicted_labels != actual_labels)
accuracy <- 1 - misclassification_error
print(accuracy)
```

```
## [1] 0.7919463
```

```
chitest = anova(logReg,test="Chisq")
chitest
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     564     763.63
## Pclass  2   69.483       562     694.15 8.163e-16 ***
## Sex     1  156.007       561     538.14 < 2.2e-16 ***
## Age     1   17.682       560     520.46 2.611e-05 ***
## SibSp   1   10.866       559     509.59 0.0009795 ***
## Parch   1    0.289       558     509.30 0.5911460
## Fare    1    1.074       557     508.23 0.3001461
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
newdata = data.frame(Pclass=as.factor(c(1,3)),Sex=c("female","female"),Age=c(40,40),SibSp=c(1,1),Parch=
newdata
```

```
##   Pclass    Sex Age SibSp Parch Fare
## 1      1 female  40     1     0  200
## 2      3 female  40     1     0   20
```

```
casepredict = predict(logReg,newdata = newdata,type="response")
casepredict
```

```
##         1         2
## 0.9217204 0.4136693
```

Given a p value of less that 2.2*10^(-16) suggest it is really relevant predictor. Given that H_0 is predictor could just as well be 0, a low p value would break this hypothesis. Comparing the a woman with the all the same factors, except for class and fare price, the higher class woman has a 0.9217204 chance of survival, while the lower class woman has a 0.4136693 chance of survival.

```
ldamodel = lda(Survived ~ ., data = titanic_train)
print(ldamodel)
```

```
## Call:
## lda(Survived ~ ., data = titanic_train)
##
## Prior probabilities of groups:
##         0         1
## 0.5929204 0.4070796
##
## Group means:
##     Pclass2   Pclass3   Sexmale      Age     SibSp     Parch     Fare
## 0 0.2119403 0.6238806 0.8417910 30.74328 0.5432836 0.3820896 23.76219
## 1 0.2695652 0.2956522 0.3173913 28.71413 0.5000000 0.5217391 53.21353
##
## Coefficients of linear discriminants:
##                  LD1
## Pclass2 -0.789073856
## Pclass3 -1.504574355
## Sexmale -2.068582677
## Age     -0.026553115
## SibSp   -0.251774569
## Parch   -0.086165420
## Fare     0.001778469
```

```
predicted <- predict(ldamodel, newdata = titanic_test)
predicted_labels <- predicted$class
actual_labels <- titanic_test$Survived
misclassification_error = mean(predicted_labels != actual_labels)
accuracy <- 1 - misclassification_error
print(accuracy)
```

```
## [1] 0.7919463
```

```r
qdamodel = qda(Survived ~ ., data = titanic_train)
print(qdamodel)
```

```
## Call:
## qda(Survived ~ ., data = titanic_train)
##
## Prior probabilities of groups:
##         0         1
## 0.5929204 0.4070796
##
## Group means:
##     Pclass2   Pclass3   Sexmale      Age     SibSp     Parch     Fare
## 0 0.2119403 0.6238806 0.8417910 30.74328 0.5432836 0.3820896 23.76219
## 1 0.2695652 0.2956522 0.3173913 28.71413 0.5000000 0.5217391 53.21353
```

```r
predicted <- predict(qdamodel, newdata = titanic_test)
predicted_labels <- predicted$class
actual_labels <- titanic_test$Survived
misclassification_error = mean(predicted_labels != actual_labels)
accuracy <- 1 - misclassification_error
print(accuracy)
```

```
## [1] 0.7986577
```

```r
logreg_probs <- predict(logReg, newdata = titanic_test, type = "response")
lda_probs <- predict(ldamodel, newdata = titanic_test)$posterior[,2]
qda_probs <- predict(qdamodel, newdata = titanic_test)$posterior[,2]
actual_outcomes <- titanic_test$Survived

# ROC for Logistic Regression
roc_logreg <- roc(actual_outcomes, logreg_probs)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# ROC for LDA
roc_lda <- roc(actual_outcomes, lda_probs)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# ROC for QDA
roc_qda <- roc(actual_outcomes, qda_probs)
```

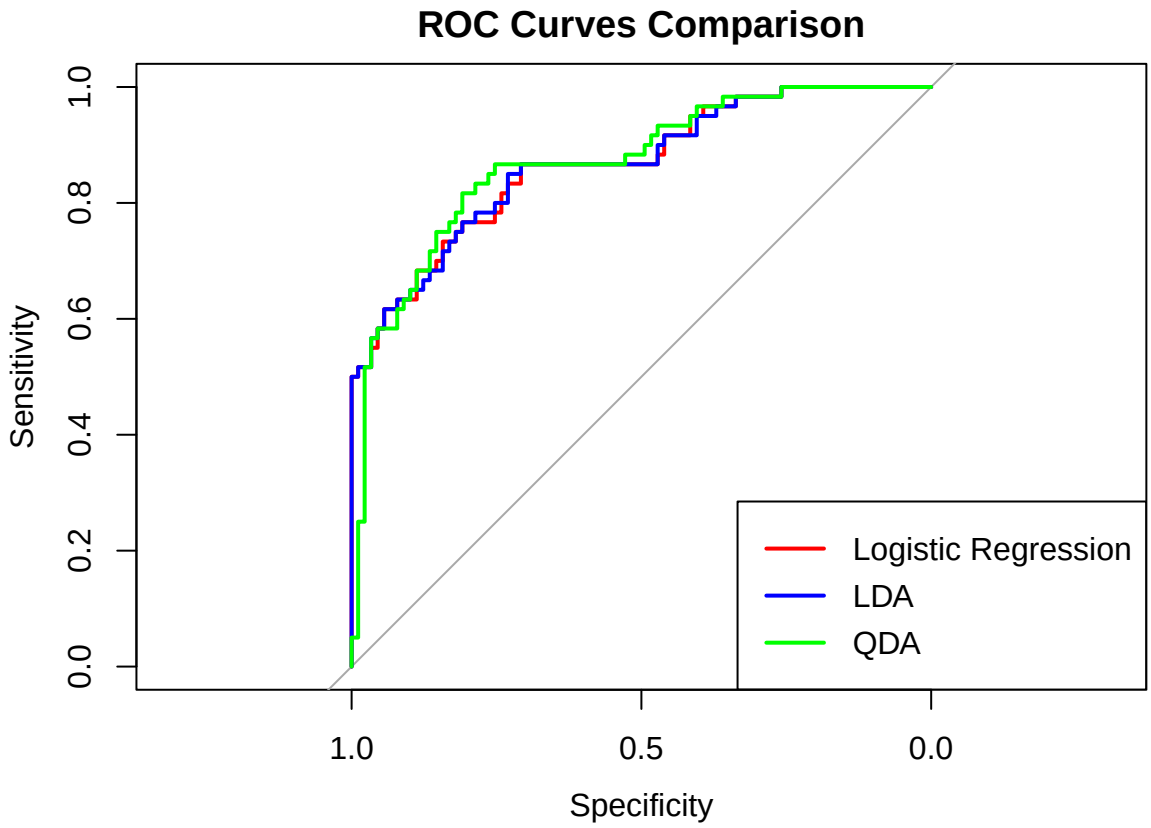
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# Plot ROC curve for Logistic Regression
plot(roc_logreg, main="ROC Curves Comparison", col="red")

# Add ROC curve for LDA
lines(roc_lda, col="blue")

# Add ROC curve for QDA
lines(roc_qda, col="green")

# Add legend
legend("bottomright", legend=c("Logistic Regression", "LDA", "QDA"),
       col=c("red", "blue", "green"), lwd=2)
```




```r
# AUC for Logistic Regression
auc_logreg <- roc_logreg$auc

# AUC for LDA
auc_lda <- roc_lda$auc

# AUC for QDA
auc_qda <- roc_qda$auc
# Print AUC values
cat("AUC for Logistic Regression:", auc_logreg, "\n")
```

```
## AUC for Logistic Regression: 0.8677903
```

```r
cat("AUC for LDA:", auc_lda, "\n")
```

```
## AUC for LDA: 0.867603
```

```r
cat("AUC for QDA:", auc_qda, "\n")
```

```
## AUC for QDA: 0.8702247
```

AUC over 0.8 is generally considered good. QDA performs slightly better than LDA and Logistic Regression. However Logistic regression is almost as accurate, but more interpretable, for one it gives clearer relationships between the predictors and they are linear.

**b)

Diagnostic Paradigm:

The diagnostic paradigm in classification focuses on directly modeling the relationship between the predictors and the outcome (class labels. The primary goal is to diagnose the class of an observation based on its features. Models under this paradigm typically estimate the probability that an observation belongs to a certain class given its features. This approach is more deterministic, where the decision rule is often based on the estimated probabilities and predefined thresholds. Sampling Paradigm:

The sampling paradigm, on the other hand, is based on modeling the distribution of features for each class. It involves estimating how the data is generated or sampled for each class and then uses this information to classify new observations. This approach is more generative, as it tries to understand the underlying data generation process for each class and uses this generative model to make classification decisions.

Key Differences:

Modeling Focus: The diagnostic paradigm focuses on the direct relationship between features and classes, often modeling the conditional probability of a class given the features. The sampling paradigm focuses on modeling the distribution of features for each class, effectively capturing how data for each class is generated. Approach: Diagnostic models are more discriminative, directly seeking a decision boundary between classes. Sampling models are generative, aiming to understand the data generation process for each class. Decision Rule: In the diagnostic paradigm, classification decisions are often made based on estimated probabilities and thresholds. In the sampling paradigm, decisions are made based on likelihoods derived from the modeled distributions of features for each class.

ii) Classification Models and Their Paradigms

Logistic Regression: Belongs to the diagnostic paradigm. It models the probability of a class given the features directly using the logistic function.

KNN (K-Nearest Neighbors): Can be seen as part of the sampling paradigm. It classifies an observation based on the majority class among its K nearest neighbors, effectively using the local distribution of the data around an observation to make a classification decision.

Naive Bayes Classifier: Fits within the sampling paradigm. It is a generative model that assumes independence among features given the class and estimates the distribution of each feature within each class to compute the posterior probability of the class given an observation.

LDA (Linear Discriminant Analysis): Primarily fits within the sampling paradigm. LDA models the distribution of features in each class (assuming a Gaussian distribution with a common covariance matrix across classes) and uses this to classify new observations by finding the class that maximizes the posterior probability.

QDA (Quadratic Discriminant Analysis): Also belongs to the sampling paradigm. Similar to LDA, QDA models the distribution of features for each class, but it allows for class-specific covariance matrices, leading to quadratic decision boundaries. Like LDA, it uses the estimated distributions to compute class probabilities for classification.