machine learning(732A99) lab2

Anubhav Dikshit(anudi287)

10 December 2018

Contents

Assignment 1

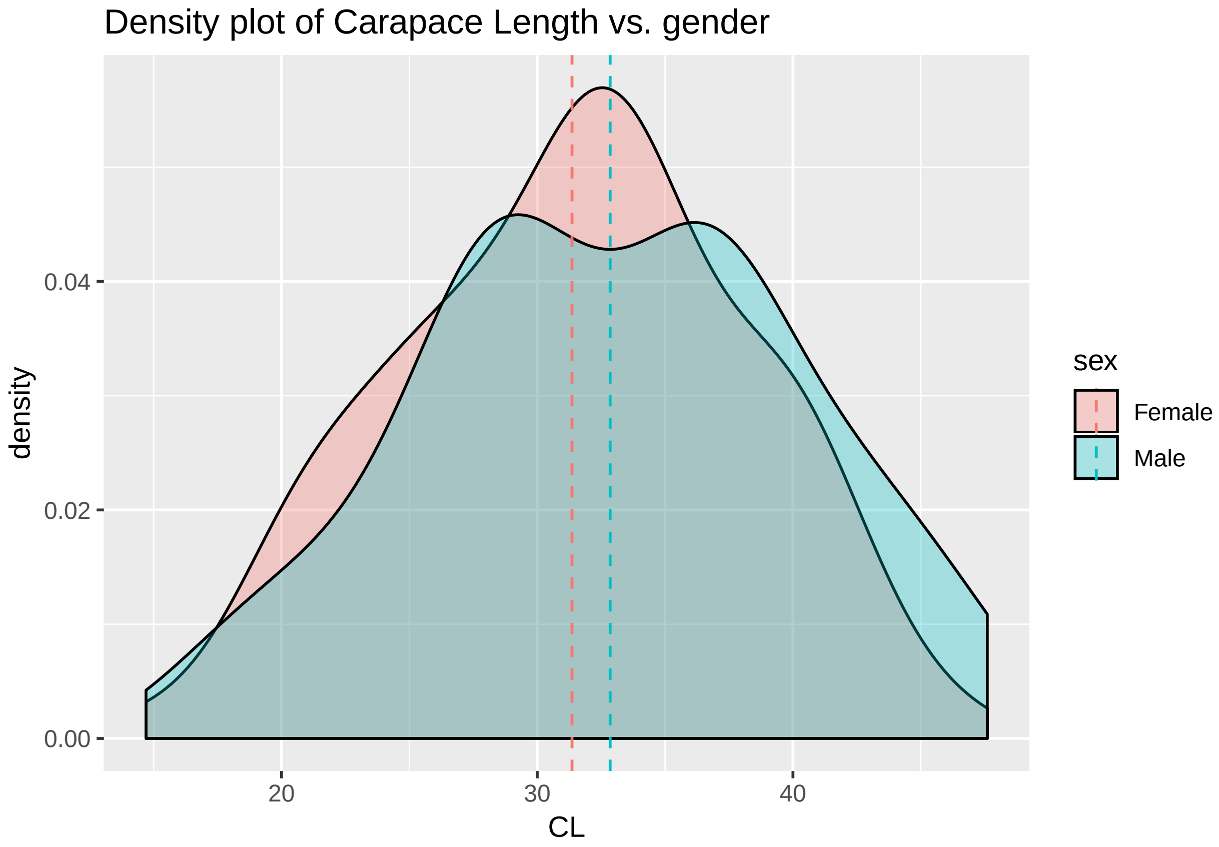
Loading The Libraries

Loading Input files

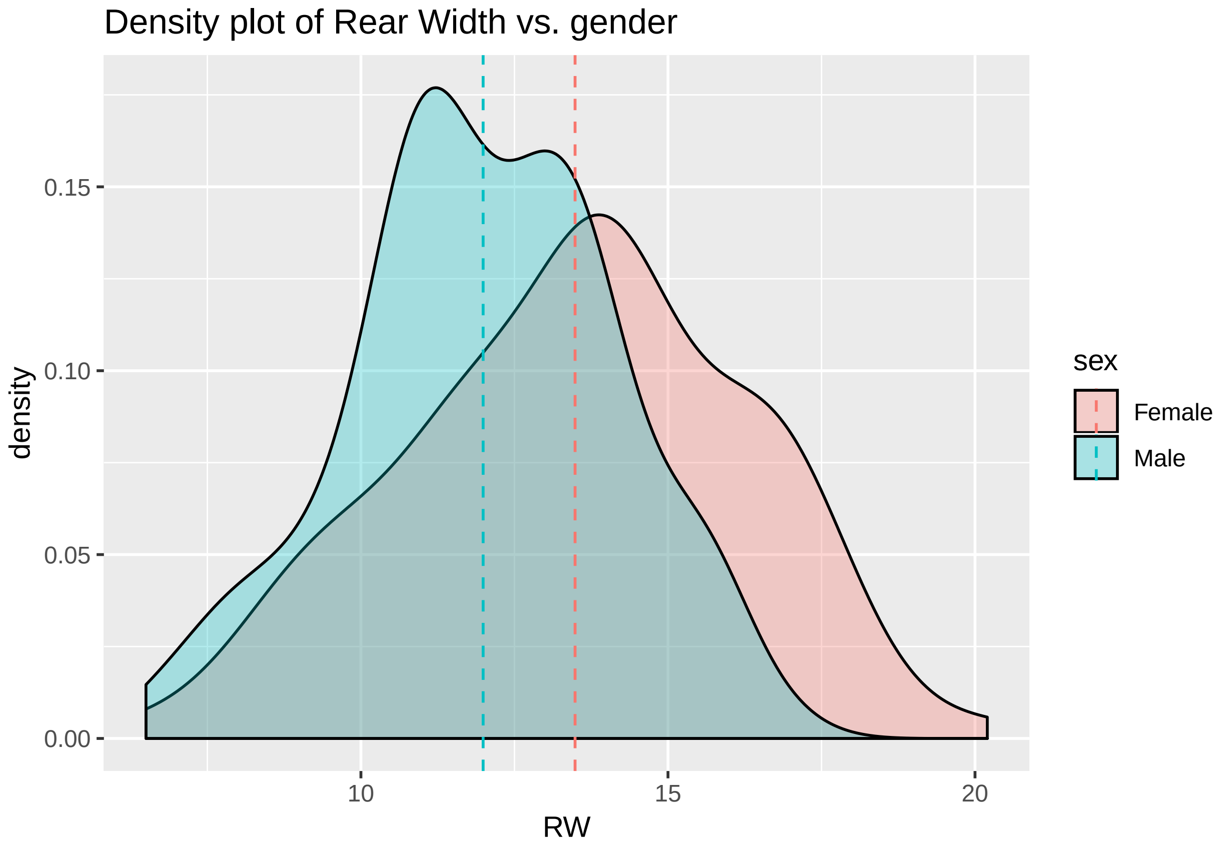
```
crab_data <- read.csv(file = "australian-crabs.csv", header = TRUE)
credit_data <- read.xlsx("creditscoring.xls", sheetName = "credit")
credit_data$good_bad <- as.factor(credit_data$good_bad)
```

1.1 Use australian-crabs.csv and make a scatterplot of carapace length (CL) versus rear width (RW) where observations are colored by Sex. Do you think that this data is easy to classify by linear discriminant analysis? Motivate your answer.

```
p1 <- ggplot(data = crab_data, aes(x = CL, y = RW, color = sex )) + geom_point() +
  geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Sex")

mu_CL <- crab_data %>%
  group_by(sex) %>%
  summarise(grp.mean = mean(CL))

mu_RW <- crab_data %>%
  group_by(sex) %>%
  summarise(grp.mean = mean(RW))


ggplot(data = crab_data, aes(x = CL)) +
  geom_density(aes(fill = sex), alpha = 0.3) +
      geom_vline(aes(xintercept = grp.mean, color = sex),
                 data = mu_CL, linetype = "dashed") +
  ggtitle("Density plot of Carapace Length vs. gender")
```



```r
ggplot(data = crab_data, aes(x = RW)) +
  geom_density(aes(fill = sex), alpha = 0.3) +
      geom_vline(aes(xintercept = grp.mean, color = sex),
             data = mu_RW, linetype = "dashed") +
  ggtitle("Density plot of Rear Width vs. gender")
```



Analysis: In Linear Discriminant Analysis (LDA) the boundary between differenet class of datapoints is a line just as the case in a logistics regression. In LDA there is an assumption that the data points for each class come from a Gaussian distribution with same variance,but different means, just as an added measure we have plotted this also. Here we find that for variable 'Carapace Length' the mean is only slightly different however for variable 'Rear width' the mean between the two sex is seperated by a larger margin.

Thus although the assumptions are violated a bit, judging by the orginal scatter plot we do find this to be case where LDA might do a good job.

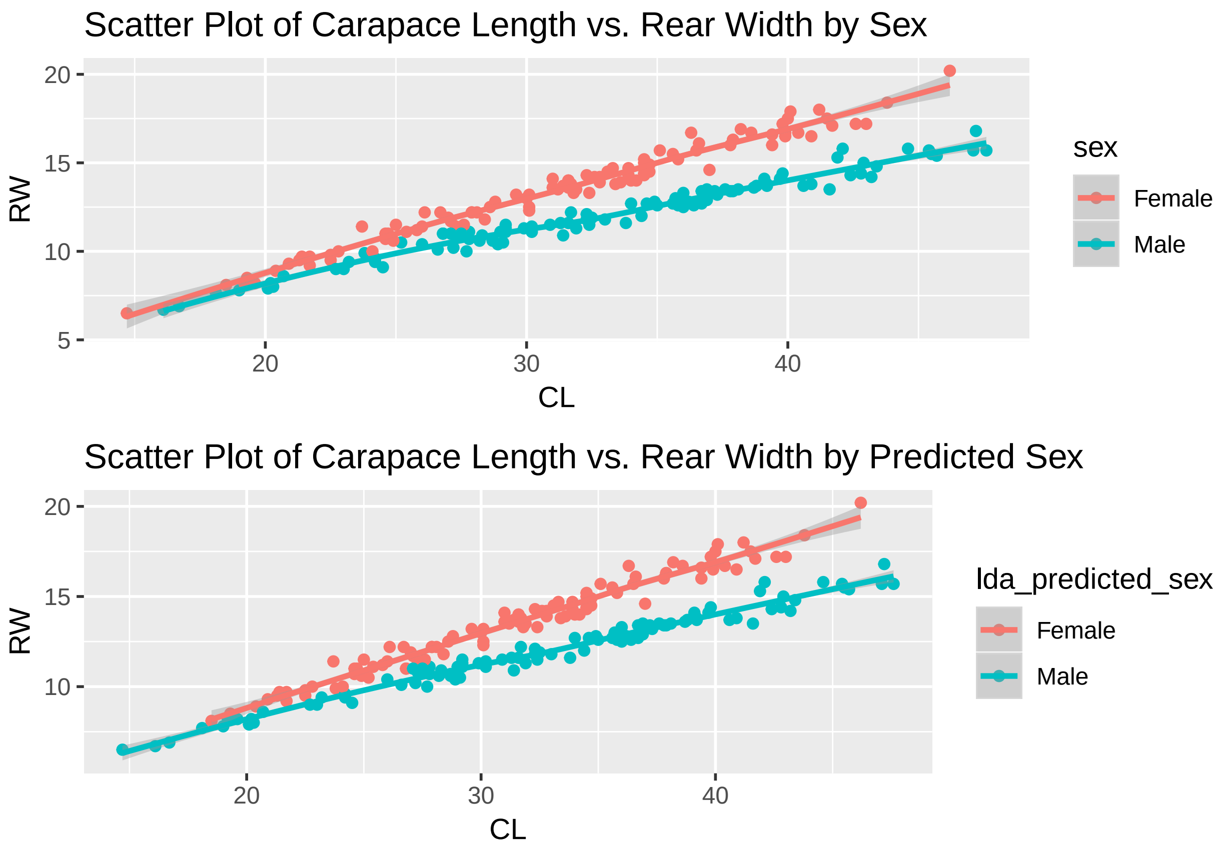
1.2 Make LDA analysis with target Sex and features CL and RW and proportional prior by using lda() function in package MASS. Make a scatter plot of CL versus RW colored by the predicted Sex and compare it with the plot in step 1. Compute the misclassification error and comment on the quality of fit.

```
set.seed(12345)
temp <- crab_data

## using priors same as the propotional of the dataset
crab_lda <- MASS::lda(formula = sex ~ CL+ RW, data = temp)
print(crab_lda)
```

```
## Call:
## lda(sex ~ CL + RW, data = temp)
##
## Prior probabilities of groups:
```

```
## Female   Male
##    0.5    0.5
##
## Group means:
##            CL     RW
## Female 31.360 13.487
## Male   32.851 11.990
##
## Coefficients of linear discriminants:
##           LD1
## CL  0.5765241
## RW -1.6823062
```

```
lda_predicted_class <- predict(crab_lda, newdata = temp)
temp$lda_predicted_sex <- lda_predicted_class$class

p2 <- ggplot(data = temp, aes(x = CL, y = RW, color = lda_predicted_sex)) +
  geom_point() + geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Predicted Sex")

gridExtra::grid.arrange(p1, p2, nrow = 2)
```



```
misclassification_lda <- table(temp$sex, temp$lda_predicted_sex)
names(dimnames(misclassification_lda)) <- c("Actual", "Predicted")
caret::confusionMatrix(misclassification_lda)
```

```
## Confusion Matrix and Statistics
```

```
##
##          Predicted
## Actual   Female Male
##   Female     97    3
##   Male        4   96
##
##                Accuracy : 0.965
##                  95% CI : (0.9292, 0.9858)
##     No Information Rate : 0.505
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.93
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9604
##             Specificity : 0.9697
##          Pos Pred Value : 0.9700
##          Neg Pred Value : 0.9600
##              Prevalence : 0.5050
##          Detection Rate : 0.4850
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9650
##
##        'Positive' Class : Female
##
```

Analysis:

The Accuracy of the fit is 96.5% thus the misclassification rate is 3.5%. Such a high value suggests that our model maybe overfit on the dataset, however to asses the fit we need a test dataset.

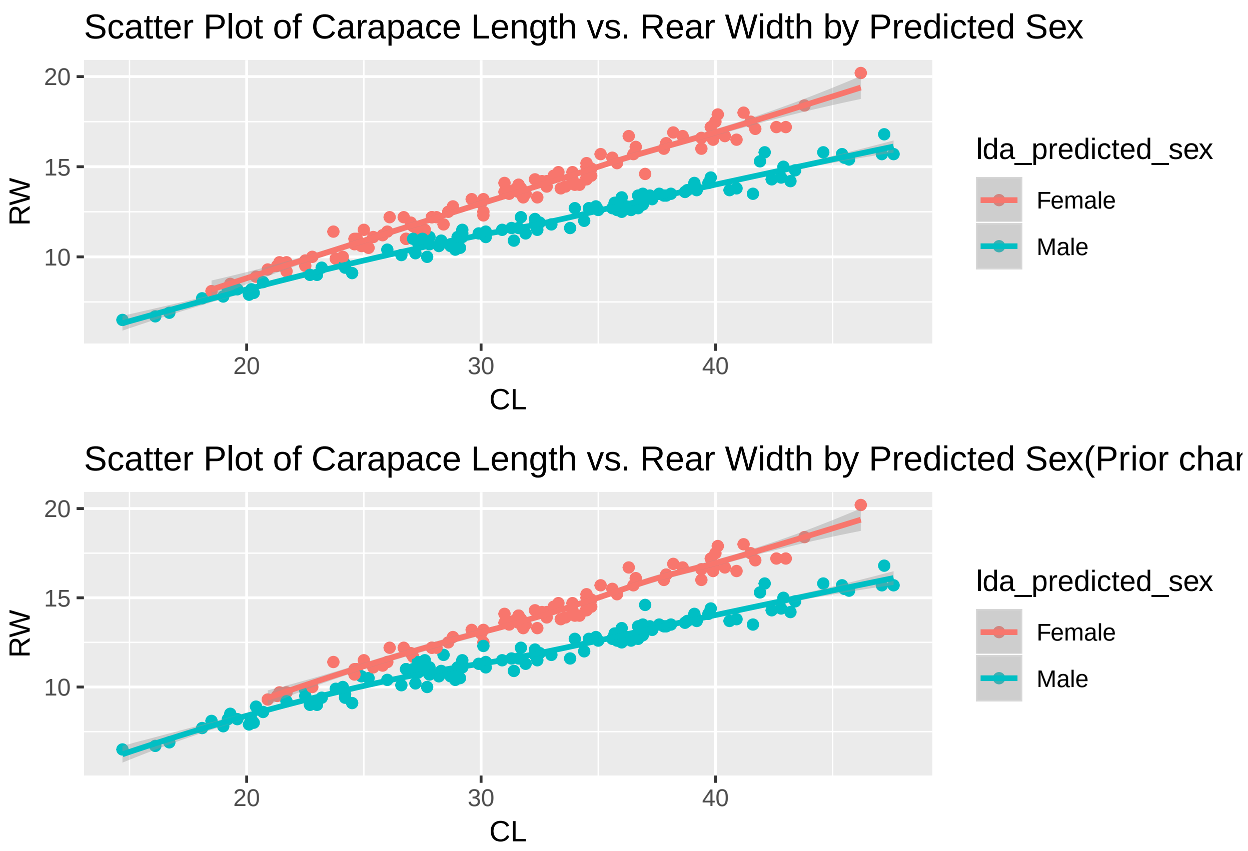
As evident from the plot we see that some of 'Female' crabs are classified as 'Males' especially when the Carapace Length (CL) is below 20 and Rear width(RW) is below 10.

1.3 Repeat step 2 but use priors p(Male)=0.9, p(Female)=0.1 instead. How did the classification result change and why?

```r
set.seed(12345)
temp <- crab_data

## using priors same as the propotional of the dataset
crab_lda <- MASS::lda(formula = sex ~ CL+ RW, data = temp, prior = c(0.1, 0.9))
print(crab_lda)
```

```
## Call:
## lda(sex ~ CL + RW, data = temp, prior = c(0.1, 0.9))
##
## Prior probabilities of groups:
## Female   Male
##    0.1    0.9
##
## Group means:
##            CL     RW
## Female 31.360 13.487
## Male   32.851 11.990
##
## Coefficients of linear discriminants:
##           LD1
## CL  0.5765241
## RW -1.6823062
```

```
lda_predicted_class <- predict(crab_lda, newdata = temp)
temp$lda_predicted_sex <- lda_predicted_class$class

p3 <- ggplot(data = temp, aes(x = CL, y = RW, color = lda_predicted_sex)) +
  geom_point() + geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Predicted Sex(Prior changed)")

gridExtra::grid.arrange(p2, p3, nrow = 2)
```



```
misclassification_lda <- table(temp$sex, temp$lda_predicted_sex)
names(dimnames(misclassification_lda)) <- c("Actual", "Predicted")
caret::confusionMatrix(misclassification_lda)
```

```
## Confusion Matrix and Statistics
##
##         Predicted
## Actual   Female Male
##   Female     84   16
##   Male        0  100
##
```

```
##                Accuracy : 0.92
##                  95% CI : (0.8733, 0.9536)
##     No Information Rate : 0.58
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.84
##  Mcnemar's Test P-Value : 0.0001768
##
##             Sensitivity : 1.0000
##             Specificity : 0.8621
##          Pos Pred Value : 0.8400
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4200
##          Detection Rate : 0.4200
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9310
##
##        'Positive' Class : Female
##
```

Analysis:

The Accuracy of the fit is 92% thus the misclassification rate is 8%.

As evident from the confusion matrix we notice that all ‘Males’ crabs are classfied correctly, while some (16/100) of the female crabs are classified wrongly.

Compared to previous plot we see that the extend of misclassification for females has increased for lower values of CW and RL compared to previous model with prior same as the dataset.

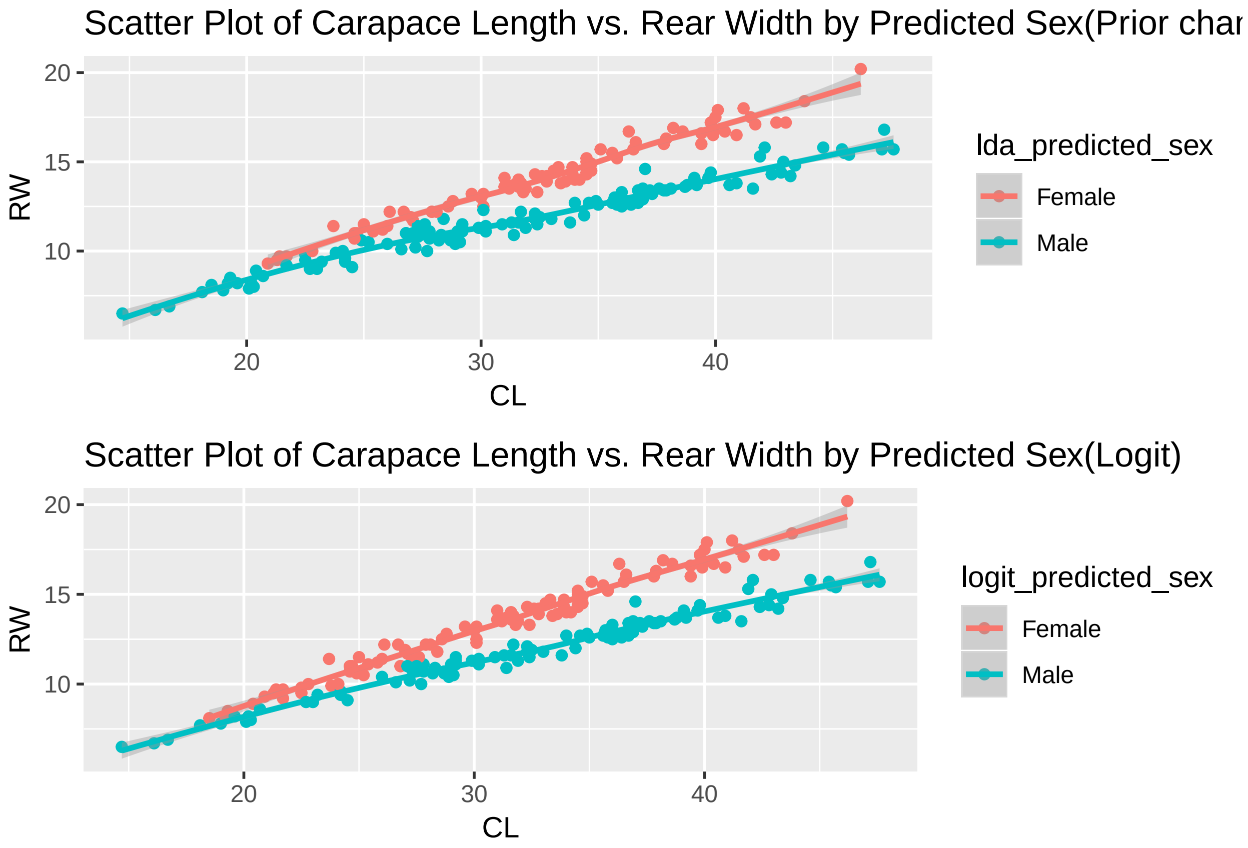
The classification is worse now compared to previous model because the dataset has the priors of 50-50 for both the sexes while we biased the model with wrong prior.

1.4 Make a similar kind of classification by logistic regression (use function glm()), plot the classified data and compute the misclassification error. Compare these results with the LDA results. Finally, report the equation of the decision boundary and draw it in the plot of the classified data.

```
set.seed(12345)
temp <- crab_data

## using priors same as the propotional of the dataset
crab_logit <- glm(formula = sex ~ CL+ RW, data = temp, family = binomial)
summary(crab_logit)
```

```
##
## Call:
## glm(formula = sex ~ CL + RW, family = binomial, data = temp)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.85416  -0.00700   0.00000   0.00081   1.89302
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   13.617      4.251   3.203 0.001359 **
## CL             4.631      1.352   3.426 0.000612 ***
## RW           -12.564      3.611  -3.479 0.000503 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.259  on 199  degrees of freedom
## Residual deviance:  24.095  on 197  degrees of freedom
## AIC: 30.095
##
## Number of Fisher Scoring iterations: 10
```

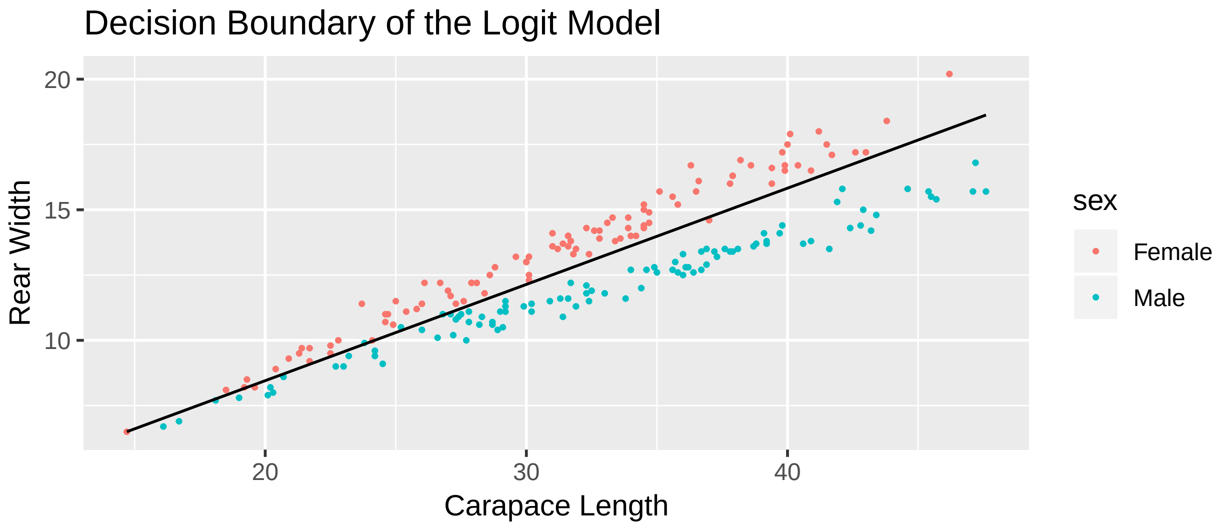
```
logit_predicted_class <- predict(crab_logit, newdata = temp, type = c("response"))
temp$logit_predicted_prob <- logit_predicted_class
temp$logit_predicted_sex <- ifelse(temp$logit_predicted_prob >= 0.5, "Male", "Female")

p4 <- ggplot(data = temp, aes(x = CL, y = RW, color = logit_predicted_sex)) +
  geom_point() + geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Predicted Sex(Logit)")

gridExtra::grid.arrange(p3, p4, nrow = 2)
```



```
misclassification_logit <- table(temp$sex, temp$logit_predicted_sex)
names(dimnames(misclassification_logit)) <- c("Actual", "Predicted")
caret::confusionMatrix(misclassification_logit)
```

```
## Confusion Matrix and Statistics
##
##           Predicted
## Actual    Female Male
##   Female      97    3
##   Male         4   96
##
##                Accuracy : 0.965
##                  95% CI : (0.9292, 0.9858)
##     No Information Rate : 0.505
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.93
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9604
##             Specificity : 0.9697
##          Pos Pred Value : 0.9700
##          Neg Pred Value : 0.9600
##              Prevalence : 0.5050
##          Detection Rate : 0.4850
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9650
##
##        'Positive' Class : Female
##
```

```
p5 <- ggplot(temp, aes(x = CL, y = RW)) + geom_point(aes(col = sex), size = 0.5)+
  labs(x="Carapace Length", y="Rear Width",
       title="Decision Boundary of the Logit Model", colour="sex") +
  coord_equal()  # assuming that the scores have the same scale


B0 <- coef(crab_logit)[1]
B1 <- coef(crab_logit)[2]
B2 <- coef(crab_logit)[3]

intercept = -B0/B2
slope = -B1/B2

x <- temp$CL
y <- (intercept + slope * x)

decision_data <- cbind(x,y) %>% as.data.frame()

# Add the decision boundary plot
p5 + geom_line(data=decision_data, aes(x=x, y=y))
```



Equation of Probability

$$P(sex =' Male'|CL, RW) = \frac{exp(13.617 + 4.631 \cdot CL - 12.564 \cdot RW)}{1 + exp(13.617 + 4.631 \cdot CL - 12.564 \cdot RW)}$$

### Equation of the Decision Boundary

$$P(sex =' Male'|CL, RW) >= 0.5$$

# Assignment 2

2.1 Import the data to R and divide into training/validation/test as 50/25/25: use data partitioning code specified in Lecture 1e.

```
set.seed(12345)

n =  NROW(credit_data)
id = sample(1:n, floor(n*0.5))
train = credit_data[id,]
test = credit_data[-id,]
```

2.2 Fit a decision tree to the training data by using the following measures of impurity: a. Deviance b. Gini index and report the misclassification rates for the training and test data. Choose the measure providing the better results for the following steps.

```r
# Create a decision tree model
credit_tree_deviance <- tree(good_bad~., data=train, split = c("deviance"))
credit_tree_gini <- tree(good_bad~., data=train, split = c("gini"))

# Visualize the decision tree with rpart.plot
summary(credit_tree_deviance)
```

```
## 
## Classification tree:
## tree(formula = good_bad ~ ., data = train, split = c("deviance"))
## Variables actually used in tree construction:
## [1] "savings"  "duration" "history"  "age"      "purpose"  "amount"  
## [7] "resident" "other"   
## Number of terminal nodes:  15 
## Residual mean deviance:  0.9569 = 458.3 / 479 
## Misclassification error rate: 0.2105 = 104 / 494
```

```r
summary(credit_tree_gini)
```

```
## 
## Classification tree:
## tree(formula = good_bad ~ ., data = train, split = c("gini"))
## Variables actually used in tree construction:
##  [1] "foreign"  "coapp"    "depends"  "telephon" "existcr"  "savings" 
##  [7] "history"  "property" "marital"  "duration" "employed" "age"     
## [13] "housing"  "amount"   "purpose"  "resident" "job"      "installp"
## Number of terminal nodes:  72 
## Residual mean deviance:  1.015 = 428.5 / 422 
## Misclassification error rate: 0.2368 = 117 / 494
```

```r
# predicting on the test dataset to get the misclassification rate.
test$predict_tree_deviance <- predict(credit_tree_deviance, newdata = test, type = "class")
test$predict_tree_gini <- predict(credit_tree_gini, newdata = test, type = "class")

conf_tree_deviance <- table(test$good_bad, test$predict_tree_deviance)
names(dimnames(conf_tree_deviance)) <- c("Actual Test", "Predicted Test")
caret::confusionMatrix(conf_tree_deviance)
```

```
## Confusion Matrix and Statistics
## 
##              Predicted Test
## Actual Test bad good
##        bad   58   95
##        good  38  309
##                                          
##                Accuracy : 0.734          
##                  95% CI : (0.693, 0.7722)
##     No Information Rate : 0.808          
##     P-Value [Acc > NIR] : 1              
## 
```

```
##                    Kappa : 0.3009
##  Mcnemar's Test P-Value : 0.000001199
##
##              Sensitivity : 0.6042
##              Specificity : 0.7649
##           Pos Pred Value : 0.3791
##           Neg Pred Value : 0.8905
##               Prevalence : 0.1920
##           Detection Rate : 0.1160
##     Detection Prevalence : 0.3060
##        Balanced Accuracy : 0.6845
##
##         'Positive' Class : bad
##
```

```
conf_tree_gini <- table(test$good_bad, test$predict_tree_gini)
names(dimnames(conf_tree_gini)) <- c("Actual Test", "Predicted Test")
caret::confusionMatrix(conf_tree_gini)
```

```
## Confusion Matrix and Statistics
##
##            Predicted Test
## Actual Test bad good
##        bad   42  111
##        good  62  285
##
##                Accuracy : 0.654
##                  95% CI : (0.6105, 0.6957)
##     No Information Rate : 0.792
##     P-Value [Acc > NIR] : 1.0000000
##
##                   Kappa : 0.1053
##  Mcnemar's Test P-Value : 0.0002629
##
##             Sensitivity : 0.4038
##             Specificity : 0.7197
##          Pos Pred Value : 0.2745
##          Neg Pred Value : 0.8213
##              Prevalence : 0.2080
##          Detection Rate : 0.0840
##    Detection Prevalence : 0.3060
##       Balanced Accuracy : 0.5618
##
##        'Positive' Class : bad
##
```

Analysis: On the Training dataset model with 'deviance' had a misclassfication rate of 21% while the model with 'gini' split had the misclassification rate of 23.68%.

For the test dataset we see that the model with 'deviance' type of split has a accuracy of 73.4% or misclassifiaction rate of 26.6%, we see that to predict 'good' the accuracy is 89% but for predicting bad its just 37%. Thus our model is heavily baised towards predicting cases as good.

For the test dataset we see that the model with 'gini' type of split has a accuracy of 65.8% or misclassifiaction rate of 34.2%, we see that to predict 'good' the accuracy is 82% but for predicting bad its just 28%. Thus our model is heavily baised towards predicting cases as good.

Both our models would lead to many bad loan applicant to be given loans which is never a good thing, however among the model the one using 'deviance' mode for split is better by 7.6%.

Thus we will select model using 'deviance' for further model building.

3. Use training and validation sets to choose the optimal tree depth. Present the graphs of the dependence of deviances for the training and the validation data on the number of leaves. Report the optimal tree, report it's depth and the variables used by the tree. Interpret the information provided by the tree structure. Estimate the misclassification rate for the test data.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
if (!require("pacman")) install.packages("pacman")
pacman::p_load(xlsx, ggplot2, MASS, tidyr, dplyr, reshape2, gridExtra, tree, caret)

set.seed(12345)
options("jtools-digits" = 2, scipen = 999)
crab_data <- read.csv(file = "australian-crabs.csv", header = TRUE)
credit_data <- read.xlsx("creditscoring.xls", sheetName = "credit")
credit_data$good_bad <- as.factor(credit_data$good_bad)
p1 <- ggplot(data = crab_data, aes(x = CL, y = RW, color = sex )) + geom_point() +
  geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Sex")

mu_CL <- crab_data %>%
  group_by(sex) %>%
  summarise(grp.mean = mean(CL))

mu_RW <- crab_data %>%
  group_by(sex) %>%
  summarise(grp.mean = mean(RW))


ggplot(data = crab_data, aes(x = CL)) +
  geom_density(aes(fill = sex), alpha = 0.3) +
      geom_vline(aes(xintercept = grp.mean, color = sex),
                 data = mu_CL, linetype = "dashed") +
  ggtitle("Density plot of Carapace Length vs. gender")


ggplot(data = crab_data, aes(x = RW)) +
  geom_density(aes(fill = sex), alpha = 0.3) +
      geom_vline(aes(xintercept = grp.mean, color = sex),
             data = mu_RW, linetype = "dashed") +
  ggtitle("Density plot of Rear Width vs. gender")


set.seed(12345)
temp <- crab_data
```

```
## using priors same as the propotional of the dataset
crab_lda <- MASS::lda(formula = sex ~ CL+ RW, data = temp)
print(crab_lda)

lda_predicted_class <- predict(crab_lda, newdata = temp)
temp$lda_predicted_sex <- lda_predicted_class$class

p2 <- ggplot(data = temp, aes(x = CL, y = RW, color = lda_predicted_sex)) +
  geom_point() + geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Predicted Sex")

gridExtra::grid.arrange(p1, p2, nrow = 2)

misclassification_lda <- table(temp$sex, temp$lda_predicted_sex)
names(dimnames(misclassification_lda)) <- c("Actual", "Predicted")
caret::confusionMatrix(misclassification_lda)
set.seed(12345)
temp <- crab_data

## using priors same as the propotional of the dataset
crab_lda <- MASS::lda(formula = sex ~ CL+ RW, data = temp, prior = c(0.1, 0.9))
print(crab_lda)

lda_predicted_class <- predict(crab_lda, newdata = temp)
temp$lda_predicted_sex <- lda_predicted_class$class

p3 <- ggplot(data = temp, aes(x = CL, y = RW, color = lda_predicted_sex)) +
  geom_point() + geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Predicted Sex(Prior changed)")

gridExtra::grid.arrange(p2, p3, nrow = 2)

misclassification_lda <- table(temp$sex, temp$lda_predicted_sex)
names(dimnames(misclassification_lda)) <- c("Actual", "Predicted")
caret::confusionMatrix(misclassification_lda)

set.seed(12345)
temp <- crab_data

## using priors same as the propotional of the dataset
crab_logit <- glm(formula = sex ~ CL+ RW, data = temp, family = binomial)
summary(crab_logit)

logit_predicted_class <- predict(crab_logit, newdata = temp, type = c("response"))
temp$logit_predicted_prob <- logit_predicted_class
temp$logit_predicted_sex <- ifelse(temp$logit_predicted_prob >= 0.5, "Male", "Female")

p4 <- ggplot(data = temp, aes(x = CL, y = RW, color = logit_predicted_sex)) +
  geom_point() + geom_smooth(method = 'loess') +
  ggtitle("Scatter Plot of Carapace Length vs. Rear Width by Predicted Sex(Logit)")

gridExtra::grid.arrange(p3, p4, nrow = 2)
```

```r
misclassification_logit <- table(temp$sex, temp$logit_predicted_sex)
names(dimnames(misclassification_logit)) <- c("Actual", "Predicted")
caret::confusionMatrix(misclassification_logit)


p5 <- ggplot(temp, aes(x = CL, y = RW)) + geom_point(aes(col = sex), size = 0.5)+
  labs(x="Carapace Length", y="Rear Width",
       title="Decision Boundary of the Logit Model", colour="sex") +
  coord_equal()  # assuming that the scores have the same scale


B0 <- coef(crab_logit)[1]
B1 <- coef(crab_logit)[2]
B2 <- coef(crab_logit)[3]

intercept = -B0/B2
slope = -B1/B2

x <- temp$CL
y <- (intercept + slope * x)

decision_data <- cbind(x,y) %>% as.data.frame()

# Add the decision boundary plot
p5 + geom_line(data=decision_data, aes(x=x, y=y))

set.seed(12345)

n =  NROW(credit_data)
id = sample(1:n, floor(n*0.5))
train = credit_data[id,]
test = credit_data[-id,]
# Create a decision tree model
credit_tree_deviance <- tree(good_bad~., data=train, split = c("deviance"))
credit_tree_gini <- tree(good_bad~., data=train, split = c("gini"))

# Visualize the decision tree with rpart.plot
summary(credit_tree_deviance)
summary(credit_tree_gini)

# predicting on the test dataset to get the misclassification rate.
test$predict_tree_deviance <- predict(credit_tree_deviance, newdata = test, type = "class")
test$predict_tree_gini <- predict(credit_tree_gini, newdata = test, type = "class")

conf_tree_deviance <- table(test$good_bad, test$predict_tree_deviance)
names(dimnames(conf_tree_deviance)) <- c("Actual Test", "Predicted Test")
caret::confusionMatrix(conf_tree_deviance)

conf_tree_gini <- table(test$good_bad, test$predict_tree_gini)
names(dimnames(conf_tree_gini)) <- c("Actual Test", "Predicted Test")
caret::confusionMatrix(conf_tree_gini)
```