Classifing Fish

**Table of Contents**

[Classification Tree 2](#_Toc512333593)

[Linear Discriminant Analysis 12](#_Toc512333594)

[Quadratic Discriminant Analysis 18](#_Toc512333595)

[Nearest Neighbor Methods 20](#_Toc512333596)

[Logistic Discrimination 22](#_Toc512333597)

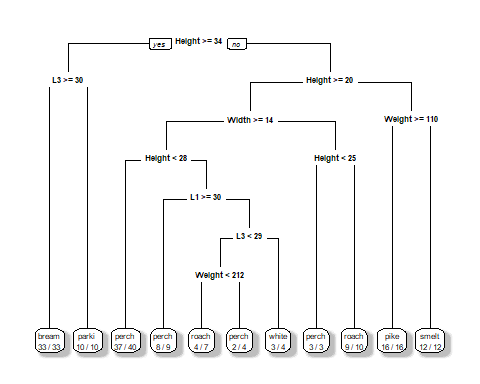
# Classification Tree

fish <- read.table("file:///C:/Users/Asus/Documents/GitHub/classifng\_fish/fish.data.txt", header = T)  
library(rpart)  
require(rpart.plot)

## Loading required package: rpart.plot

fish.control <- rpart.control(minisplit = 10, minbucket = 3, xval = 0)  
fish.treeorig <- rpart(Species~Weight+L1+L2+L3+Height+Width,data=fish,method="class",control=fish.control)  
#Let’s now plot the tree:  
plot(fish.treeorig)  
text(fish.treeorig)

prp(fish.treeorig, # 模型  
 faclen=0, # 呈現的變數不要縮寫  
 fallen.leaves=TRUE, # 讓樹枝以垂直方式呈現  
 shadow.col="gray", # 最下面的節點塗上陰影  
 extra=2 ) # number of correct classifications / number of observations in that node



#Also check out the complexity parameter (CP):  
printcp(fish.treeorig)

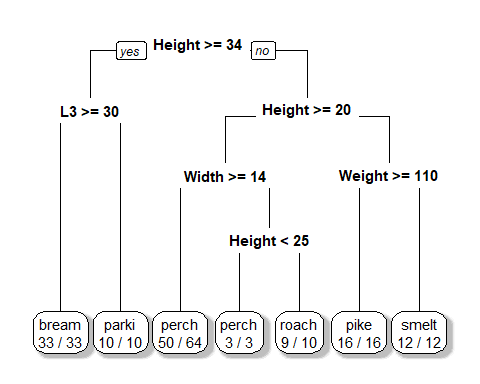
##   
## Classification tree:  
## rpart(formula = Species ~ Weight + L1 + L2 + L3 + Height + Width,   
## data = fish, method = "class", control = fish.control)  
##   
## Variables actually used in tree construction:  
## [1] Height L1 L3 Weight Width   
##   
## Root node error: 94/148 = 0.63514  
##   
## n= 148   
##   
## CP nsplit rel error  
## 1 0.351064 0 1.00000  
## 2 0.170213 1 0.64894  
## 3 0.127660 2 0.47872  
## 4 0.106383 3 0.35106  
## 5 0.053191 4 0.24468  
## 6 0.031915 5 0.19149  
## 7 0.010638 6 0.15957  
## 8 0.010000 10 0.11702

summary(fish.treeorig)

## Call:  
## rpart(formula = Species ~ Weight + L1 + L2 + L3 + Height + Width,   
## data = fish, method = "class", control = fish.control)  
## n= 148   
##   
## CP nsplit rel error  
## 1 0.35106383 0 1.0000000  
## 2 0.17021277 1 0.6489362  
## 3 0.12765957 2 0.4787234  
## 4 0.10638298 3 0.3510638  
## 5 0.05319149 4 0.2446809  
## 6 0.03191489 5 0.1914894  
## 7 0.01063830 6 0.1595745  
## 8 0.01000000 10 0.1170213  
##   
## Variable importance  
## Height L3 L2 L1 Weight Width   
## 26 16 15 15 15 13   
##   
## Node number 1: 148 observations, complexity param=0.3510638  
## predicted class=perch expected loss=0.6351351 P(node) =1  
## class counts: 33 10 54 16 18 12 5  
## probabilities: 0.223 0.068 0.365 0.108 0.122 0.081 0.034   
## left son=2 (43 obs) right son=3 (105 obs)  
## Primary splits:  
## Height < 33.9 to the right, improve=29.75863, (0 missing)  
## Width < 11.85 to the right, improve=17.98385, (0 missing)  
## L3 < 29.7 to the right, improve=13.80398, (0 missing)  
## L2 < 28.85 to the right, improve=12.96300, (0 missing)  
## L1 < 26.1 to the right, improve=12.56245, (0 missing)  
##   
## Node number 2: 43 observations, complexity param=0.106383  
## predicted class=bream expected loss=0.2325581 P(node) =0.2905405  
## class counts: 33 10 0 0 0 0 0  
## probabilities: 0.767 0.233 0.000 0.000 0.000 0.000 0.000   
## left son=4 (33 obs) right son=5 (10 obs)  
## Primary splits:  
## L3 < 29.5 to the right, improve=15.348840, (0 missing)  
## L2 < 26.15 to the right, improve=13.530660, (0 missing)  
## L1 < 23.1 to the right, improve=13.407660, (0 missing)  
## Weight < 331.5 to the right, improve=12.015500, (0 missing)  
## Width < 14.85 to the right, improve= 1.063123, (0 missing)  
## Surrogate splits:  
## L1 < 23.1 to the right, agree=0.977, adj=0.9, (0 split)  
## L2 < 25.2 to the right, agree=0.977, adj=0.9, (0 split)  
## Weight < 221 to the right, agree=0.953, adj=0.8, (0 split)  
##   
## Node number 3: 105 observations, complexity param=0.1702128  
## predicted class=perch expected loss=0.4857143 P(node) =0.7094595  
## class counts: 0 0 54 16 18 12 5  
## probabilities: 0.000 0.000 0.514 0.152 0.171 0.114 0.048   
## left son=6 (77 obs) right son=7 (28 obs)  
## Primary splits:  
## Height < 20.1 to the right, improve=21.78355, (0 missing)  
## Width < 12.45 to the right, improve=20.93000, (0 missing)  
## Weight < 25.95 to the right, improve=13.35778, (0 missing)  
## L3 < 15.6 to the right, improve=10.68888, (0 missing)  
## L1 < 12.3 to the right, improve=10.63876, (0 missing)  
## Surrogate splits:  
## Width < 12.45 to the right, agree=0.990, adj=0.964, (0 split)  
## Weight < 25.95 to the right, agree=0.838, adj=0.393, (0 split)  
## L1 < 12.3 to the right, agree=0.819, adj=0.321, (0 split)  
## L2 < 13.35 to the right, agree=0.819, adj=0.321, (0 split)  
## L3 < 14.25 to the right, agree=0.819, adj=0.321, (0 split)  
##   
## Node number 4: 33 observations  
## predicted class=bream expected loss=0 P(node) =0.222973  
## class counts: 33 0 0 0 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000 0.000 0.000 0.000   
##   
## Node number 5: 10 observations  
## predicted class=parki expected loss=0 P(node) =0.06756757  
## class counts: 0 10 0 0 0 0 0  
## probabilities: 0.000 1.000 0.000 0.000 0.000 0.000 0.000   
##   
## Node number 6: 77 observations, complexity param=0.05319149  
## predicted class=perch expected loss=0.2987013 P(node) =0.5202703  
## class counts: 0 0 54 0 18 0 5  
## probabilities: 0.000 0.000 0.701 0.000 0.234 0.000 0.065   
## left son=12 (64 obs) right son=13 (13 obs)  
## Primary splits:  
## Width < 14.4 to the right, improve=5.777691, (0 missing)  
## Height < 25.25 to the left, improve=4.275974, (0 missing)  
## L1 < 25.1 to the right, improve=2.872913, (0 missing)  
## L2 < 27.15 to the right, improve=2.872913, (0 missing)  
## Weight < 548 to the right, improve=2.448383, (0 missing)  
## Surrogate splits:  
## L1 < 13.35 to the right, agree=0.844, adj=0.077, (0 split)  
## L2 < 14.55 to the right, agree=0.844, adj=0.077, (0 split)  
##   
## Node number 7: 28 observations, complexity param=0.1276596  
## predicted class=pike expected loss=0.4285714 P(node) =0.1891892  
## class counts: 0 0 0 16 0 12 0  
## probabilities: 0.000 0.000 0.000 0.571 0.000 0.429 0.000   
## left son=14 (16 obs) right son=15 (12 obs)  
## Primary splits:  
## Weight < 109.95 to the right, improve=13.714290, (0 missing)  
## L1 < 21.9 to the right, improve=13.714290, (0 missing)  
## L2 < 23.65 to the right, improve=13.714290, (0 missing)  
## L3 < 25.5 to the right, improve=13.714290, (0 missing)  
## Height < 16.05 to the left, improve= 4.571429, (0 missing)  
## Surrogate splits:  
## L1 < 21.9 to the right, agree=1.000, adj=1.000, (0 split)  
## L2 < 23.65 to the right, agree=1.000, adj=1.000, (0 split)  
## L3 < 25.5 to the right, agree=1.000, adj=1.000, (0 split)  
## Height < 16.05 to the left, agree=0.786, adj=0.500, (0 split)  
## Width < 9.45 to the right, agree=0.714, adj=0.333, (0 split)  
##   
## Node number 12: 64 observations, complexity param=0.0106383  
## predicted class=perch expected loss=0.21875 P(node) =0.4324324  
## class counts: 0 0 50 0 9 0 5  
## probabilities: 0.000 0.000 0.781 0.000 0.141 0.000 0.078   
## left son=24 (40 obs) right son=25 (24 obs)  
## Primary splits:  
## Height < 27.55 to the left, improve=3.314583, (0 missing)  
## Width < 15.65 to the right, improve=1.557526, (0 missing)  
## L1 < 30 to the right, improve=1.174970, (0 missing)  
## L2 < 32.25 to the right, improve=1.174970, (0 missing)  
## Weight < 548 to the right, improve=1.058472, (0 missing)  
## Surrogate splits:  
## Width < 17.4 to the left, agree=0.719, adj=0.250, (0 split)  
## Weight < 267.5 to the left, agree=0.688, adj=0.167, (0 split)  
## L3 < 29.05 to the left, agree=0.656, adj=0.083, (0 split)  
##   
## Node number 13: 13 observations, complexity param=0.03191489  
## predicted class=roach expected loss=0.3076923 P(node) =0.08783784  
## class counts: 0 0 4 0 9 0 0  
## probabilities: 0.000 0.000 0.308 0.000 0.692 0.000 0.000   
## left son=26 (3 obs) right son=27 (10 obs)  
## Primary splits:  
## Height < 24.8 to the left, improve=3.7384620, (0 missing)  
## Weight < 174.5 to the right, improve=1.0051280, (0 missing)  
## L1 < 22.5 to the right, improve=1.0051280, (0 missing)  
## L2 < 24.5 to the right, improve=1.0051280, (0 missing)  
## L3 < 21.1 to the left, improve=0.4273504, (0 missing)  
##   
## Node number 14: 16 observations  
## predicted class=pike expected loss=0 P(node) =0.1081081  
## class counts: 0 0 0 16 0 0 0  
## probabilities: 0.000 0.000 0.000 1.000 0.000 0.000 0.000   
##   
## Node number 15: 12 observations  
## predicted class=smelt expected loss=0 P(node) =0.08108108  
## class counts: 0 0 0 0 0 12 0  
## probabilities: 0.000 0.000 0.000 0.000 0.000 1.000 0.000   
##   
## Node number 24: 40 observations  
## predicted class=perch expected loss=0.075 P(node) =0.2702703  
## class counts: 0 0 37 0 3 0 0  
## probabilities: 0.000 0.000 0.925 0.000 0.075 0.000 0.000   
##   
## Node number 25: 24 observations, complexity param=0.0106383  
## predicted class=perch expected loss=0.4583333 P(node) =0.1621622  
## class counts: 0 0 13 0 6 0 5  
## probabilities: 0.000 0.000 0.542 0.000 0.250 0.000 0.208   
## left son=50 (9 obs) right son=51 (15 obs)  
## Primary splits:  
## L1 < 29.5 to the right, improve=2.772222, (0 missing)  
## L2 < 31.9 to the right, improve=2.772222, (0 missing)  
## Width < 16.45 to the right, improve=2.772222, (0 missing)  
## Weight < 295 to the right, improve=2.583333, (0 missing)  
## L3 < 32.4 to the right, improve=2.216667, (0 missing)  
## Surrogate splits:  
## L2 < 31.9 to the right, agree=1.000, adj=1.000, (0 split)  
## Weight < 410 to the right, agree=0.958, adj=0.889, (0 split)  
## L3 < 32.4 to the right, agree=0.958, adj=0.889, (0 split)  
## Width < 16.45 to the right, agree=0.833, adj=0.556, (0 split)  
## Height < 29.35 to the right, agree=0.667, adj=0.111, (0 split)  
##   
## Node number 26: 3 observations  
## predicted class=perch expected loss=0 P(node) =0.02027027  
## class counts: 0 0 3 0 0 0 0  
## probabilities: 0.000 0.000 1.000 0.000 0.000 0.000 0.000   
##   
## Node number 27: 10 observations  
## predicted class=roach expected loss=0.1 P(node) =0.06756757  
## class counts: 0 0 1 0 9 0 0  
## probabilities: 0.000 0.000 0.100 0.000 0.900 0.000 0.000   
##   
## Node number 50: 9 observations  
## predicted class=perch expected loss=0.1111111 P(node) =0.06081081  
## class counts: 0 0 8 0 0 0 1  
## probabilities: 0.000 0.000 0.889 0.000 0.000 0.000 0.111   
##   
## Node number 51: 15 observations, complexity param=0.0106383  
## predicted class=roach expected loss=0.6 P(node) =0.1013514  
## class counts: 0 0 5 0 6 0 4  
## probabilities: 0.000 0.000 0.333 0.000 0.400 0.000 0.267   
## left son=102 (11 obs) right son=103 (4 obs)  
## Primary splits:  
## L3 < 29.25 to the left, improve=2.003030, (0 missing)  
## Weight < 247.5 to the left, improve=1.866667, (0 missing)  
## L1 < 22.85 to the left, improve=1.866667, (0 missing)  
## L2 < 25 to the left, improve=1.866667, (0 missing)  
## Height < 28.45 to the left, improve=1.088889, (0 missing)  
## Surrogate splits:  
## L1 < 24.05 to the left, agree=0.933, adj=0.75, (0 split)  
## L2 < 26.25 to the left, agree=0.933, adj=0.75, (0 split)  
## Weight < 303 to the left, agree=0.867, adj=0.50, (0 split)  
##   
## Node number 102: 11 observations, complexity param=0.0106383  
## predicted class=perch expected loss=0.5454545 P(node) =0.07432432  
## class counts: 0 0 5 0 5 0 1  
## probabilities: 0.000 0.000 0.455 0.000 0.455 0.000 0.091   
## left son=204 (7 obs) right son=205 (4 obs)  
## Primary splits:  
## Weight < 212.5 to the left, improve=0.4350649, (0 missing)  
## L1 < 22.05 to the left, improve=0.4350649, (0 missing)  
## L2 < 23.75 to the left, improve=0.4350649, (0 missing)  
## L3 < 26.15 to the left, improve=0.4350649, (0 missing)  
## Height < 28.5 to the left, improve=0.4350649, (0 missing)  
## Surrogate splits:  
## L2 < 23.75 to the left, agree=1.000, adj=1.00, (0 split)  
## Height < 28.5 to the left, agree=1.000, adj=1.00, (0 split)  
## L1 < 21.25 to the left, agree=0.909, adj=0.75, (0 split)  
## L3 < 25.4 to the left, agree=0.909, adj=0.75, (0 split)  
## Width < 14.95 to the right, agree=0.727, adj=0.25, (0 split)  
##   
## Node number 103: 4 observations  
## predicted class=white expected loss=0.25 P(node) =0.02702703  
## class counts: 0 0 0 0 1 0 3  
## probabilities: 0.000 0.000 0.000 0.000 0.250 0.000 0.750   
##   
## Node number 204: 7 observations  
## predicted class=roach expected loss=0.4285714 P(node) =0.0472973  
## class counts: 0 0 3 0 4 0 0  
## probabilities: 0.000 0.000 0.429 0.000 0.571 0.000 0.000   
##   
## Node number 205: 4 observations  
## predicted class=perch expected loss=0.5 P(node) =0.02702703  
## class counts: 0 0 2 0 1 0 1  
## probabilities: 0.000 0.000 0.500 0.000 0.250 0.000 0.250

fish.prunetree <- prune.rpart(fish.treeorig,cp=0.02)  
plot(fish.prunetree)  
text(fish.prunetree)

prp(fish.prunetree, # 模型  
 faclen=0, # 呈現的變數不要縮寫  
 fallen.leaves=TRUE, # 讓樹枝以垂直方式呈現  
 shadow.col="gray", # 最下面的節點塗上陰影  
 extra=2 ) # number of correct classifications / number of observations in that node

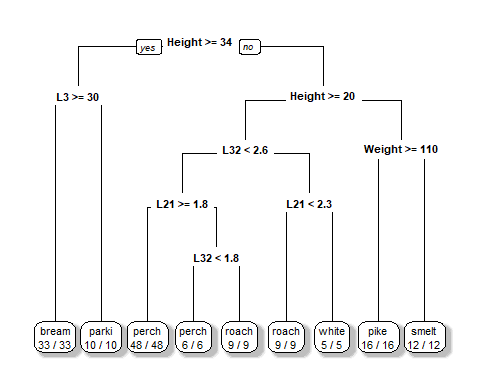


L21<-fish$L2-fish$L1  
L32<-fish$L3-fish$L2  
L31<-fish$L3-fish$L1  
newfish<-cbind(fish,L21,L32,L31)  
newfish.treenew<-rpart(Species~., data=newfish,method="class",parms=list(split="information"),control=fish.control)  
printcp(newfish.treenew)

##   
## Classification tree:  
## rpart(formula = Species ~ ., data = newfish, method = "class",   
## parms = list(split = "information"), control = fish.control)  
##   
## Variables actually used in tree construction:  
## [1] Height L21 L3 L32 Weight  
##   
## Root node error: 94/148 = 0.63514  
##   
## n= 148   
##   
## CP nsplit rel error  
## 1 0.351064 0 1.000000  
## 2 0.170213 1 0.648936  
## 3 0.127660 2 0.478723  
## 4 0.106383 3 0.351064  
## 5 0.095745 4 0.244681  
## 6 0.053191 5 0.148936  
## 7 0.047872 6 0.095745  
## 8 0.010000 8 0.000000

plot(newfish.treenew)  
text(newfish.treenew)

prp(newfish.treenew, # 模型  
 faclen=0, # 呈現的變數不要縮寫  
 fallen.leaves=TRUE, # 讓樹枝以垂直方式呈現  
 shadow.col="gray", # 最下面的節點塗上陰影  
 extra=2 ) # number of correct classifications / number of observations in that node



#分的有點完美(有點過度配適)  
fish.control <- rpart.control(minbucket=3,minsplit=10,xval=148)  
newfish.treenewcv <- rpart(Species~., data=newfish,method="class",parms=list(split="information"),control=fish.control)  
printcp(newfish.treenewcv)

##   
## Classification tree:  
## rpart(formula = Species ~ ., data = newfish, method = "class",   
## parms = list(split = "information"), control = fish.control)  
##   
## Variables actually used in tree construction:  
## [1] Height L21 L3 L32 Weight  
##   
## Root node error: 94/148 = 0.63514  
##   
## n= 148   
##   
## CP nsplit rel error xerror xstd  
## 1 0.351064 0 1.000000 1.000000 0.062302  
## 2 0.170213 1 0.648936 0.648936 0.063704  
## 3 0.127660 2 0.478723 0.478723 0.059534  
## 4 0.106383 3 0.351064 0.351064 0.053870  
## 5 0.095745 4 0.244681 0.361702 0.054442  
## 6 0.053191 5 0.148936 0.170213 0.040187  
## 7 0.047872 6 0.095745 0.180851 0.041267  
## 8 0.010000 8 0.000000 0.031915 0.018238

newfish.test<-read.table("file:///C:/Users/Asus/Documents/GitHub/classifng\_fish/fish\_test.data.txt",h=T)  
L31<-newfish.test$L3- newfish.test$L1  
L32<-newfish.test$L3- newfish.test$L2  
L21<-newfish.test$L2- newfish.test$L1  
newfish.test<-cbind(newfish.test,L21,L32,L31)  
newfish.tpred<-predict(newfish.treenewcv,newfish.test)  
newfish.tpred

## bream parki perch pike roach smelt white  
## 1 1 0 0 0 0 0 0  
## 2 1 0 0 0 0 0 0  
## 3 0 0 1 0 0 0 0  
## 4 0 0 1 0 0 0 0  
## 5 0 0 0 1 0 0 0  
## 6 0 0 0 0 0 1 0  
## 7 0 0 0 0 0 1 0  
## 8 0 1 0 0 0 0 0  
## 9 0 0 0 0 1 0 0  
## 10 0 0 0 0 1 0 0  
## 11 0 0 0 0 0 0 1

# Linear Discriminant Analysis

library(MASS)  
newfish

## Species Weight L1 L2 L3 Height Width L21 L32 L31  
## 1 bream 242.0 23.2 25.4 30.0 38.4 13.4 2.2 4.6 6.8  
## 2 bream 290.0 24.0 26.3 31.2 40.0 13.8 2.3 4.9 7.2  
## 3 bream 363.0 26.3 29.0 33.5 38.0 13.3 2.7 4.5 7.2  
## 4 bream 430.0 26.5 29.0 34.0 36.6 15.1 2.5 5.0 7.5  
## 5 bream 500.0 26.8 29.7 34.5 41.1 15.3 2.9 4.8 7.7  
## 6 bream 390.0 27.6 30.0 35.0 36.2 13.4 2.4 5.0 7.4  
## 7 bream 450.0 27.6 30.0 35.1 39.9 13.8 2.4 5.1 7.5  
## 8 bream 500.0 28.5 30.7 36.2 39.3 13.7 2.2 5.5 7.7  
## 9 bream 475.0 28.4 31.0 36.2 39.4 14.1 2.6 5.2 7.8  
## 10 bream 500.0 28.7 31.0 36.2 39.7 13.3 2.3 5.2 7.5  
## 11 bream 500.0 29.1 31.5 36.4 37.8 12.0 2.4 4.9 7.3  
## 12 bream 500.0 29.5 32.0 37.3 37.3 13.6 2.5 5.3 7.8  
## 13 bream 600.0 29.4 32.0 37.2 40.2 13.9 2.6 5.2 7.8  
## 14 bream 600.0 29.4 32.0 37.2 41.5 15.0 2.6 5.2 7.8  
## 15 bream 700.0 30.4 33.0 38.3 38.8 13.8 2.6 5.3 7.9  
## 16 bream 700.0 30.4 33.0 38.5 38.8 13.5 2.6 5.5 8.1  
## 17 bream 610.0 30.9 33.5 38.6 40.5 13.3 2.6 5.1 7.7  
## 18 bream 650.0 31.0 33.5 38.7 37.4 14.8 2.5 5.2 7.7  
## 19 bream 575.0 31.3 34.0 39.5 38.3 14.1 2.7 5.5 8.2  
## 20 bream 685.0 31.4 34.0 39.2 40.8 13.7 2.6 5.2 7.8  
## 21 bream 620.0 31.5 34.5 39.7 39.1 13.3 3.0 5.2 8.2  
## 22 bream 680.0 31.8 35.0 40.6 38.1 15.1 3.2 5.6 8.8  
## 23 bream 700.0 31.9 35.0 40.5 40.1 13.8 3.1 5.5 8.6  
## 24 bream 725.0 31.8 35.0 40.9 40.0 14.8 3.2 5.9 9.1  
## 25 bream 720.0 32.0 35.0 40.6 40.3 15.0 3.0 5.6 8.6  
## 26 bream 714.0 32.7 36.0 41.5 39.8 14.1 3.3 5.5 8.8  
## 27 bream 850.0 32.8 36.0 41.6 40.6 14.9 3.2 5.6 8.8  
## 28 bream 1000.0 33.5 37.0 42.6 44.5 15.5 3.5 5.6 9.1  
## 29 bream 920.0 35.0 38.5 44.1 40.9 14.3 3.5 5.6 9.1  
## 30 bream 955.0 35.0 38.5 44.0 41.1 14.3 3.5 5.5 9.0  
## 31 bream 925.0 36.2 39.5 45.3 41.4 14.9 3.3 5.8 9.1  
## 32 bream 975.0 37.4 41.0 45.9 40.6 14.7 3.6 4.9 8.5  
## 33 bream 950.0 38.0 41.0 46.5 37.9 13.7 3.0 5.5 8.5  
## 34 white 270.0 23.6 26.0 28.7 29.2 14.8 2.4 2.7 5.1  
## 35 white 270.0 24.1 26.5 29.3 27.8 14.5 2.4 2.8 5.2  
## 36 white 306.0 25.6 28.0 30.8 28.5 15.2 2.4 2.8 5.2  
## 37 white 540.0 28.5 31.0 34.0 31.6 19.3 2.5 3.0 5.5  
## 38 white 1000.0 37.3 40.0 43.5 28.4 15.0 2.7 3.5 6.2  
## 39 roach 40.0 12.9 14.1 16.2 25.6 14.0 1.2 2.1 3.3  
## 40 roach 69.0 16.5 18.2 20.3 26.1 13.9 1.7 2.1 3.8  
## 41 roach 78.0 17.5 18.8 21.2 26.3 13.7 1.3 2.4 3.7  
## 42 roach 87.0 18.2 19.8 22.2 25.3 14.3 1.6 2.4 4.0  
## 43 roach 120.0 18.6 20.0 22.2 28.0 16.1 1.4 2.2 3.6  
## 44 roach 118.0 19.0 20.5 22.8 28.4 14.7 1.5 2.3 3.8  
## 45 roach 110.0 19.1 20.8 23.1 26.7 14.7 1.7 2.3 4.0  
## 46 roach 120.0 19.4 21.0 23.7 25.8 13.9 1.6 2.7 4.3  
## 47 roach 160.0 20.5 22.5 25.3 27.8 15.1 2.0 2.8 4.8  
## 48 roach 140.0 21.0 22.5 25.0 26.2 13.3 1.5 2.5 4.0  
## 49 roach 160.0 21.1 22.5 25.0 25.6 15.2 1.4 2.5 3.9  
## 50 roach 169.0 22.0 24.0 27.2 27.7 14.1 2.0 3.2 5.2  
## 51 roach 161.0 22.0 23.4 26.7 25.9 13.6 1.4 3.3 4.7  
## 52 roach 200.0 22.1 23.5 26.8 27.6 15.4 1.4 3.3 4.7  
## 53 roach 180.0 23.6 25.2 27.9 25.4 14.0 1.6 2.7 4.3  
## 54 roach 290.0 24.0 26.0 29.2 30.4 15.4 2.0 3.2 5.2  
## 55 roach 272.0 25.0 27.0 30.6 28.0 15.6 2.0 3.6 5.6  
## 56 roach 390.0 29.5 31.7 35.0 27.1 15.3 2.2 3.3 5.5  
## 57 parki 55.0 13.5 14.7 16.5 41.5 14.1 1.2 1.8 3.0  
## 58 parki 60.0 14.3 15.5 17.4 37.8 13.3 1.2 1.9 3.1  
## 59 parki 90.0 16.3 17.7 19.8 37.4 13.5 1.4 2.1 3.5  
## 60 parki 120.0 17.5 19.0 21.3 39.4 13.7 1.5 2.3 3.8  
## 61 parki 150.0 18.4 20.0 22.4 39.7 14.7 1.6 2.4 4.0  
## 62 parki 140.0 19.0 20.7 23.2 36.8 14.2 1.7 2.5 4.2  
## 63 parki 170.0 19.0 20.7 23.2 40.5 14.7 1.7 2.5 4.2  
## 64 parki 200.0 21.2 23.0 25.8 40.1 14.2 1.8 2.8 4.6  
## 65 parki 273.0 23.0 25.0 28.0 39.6 14.8 2.0 3.0 5.0  
## 66 parki 300.0 24.0 26.0 29.0 39.2 14.6 2.0 3.0 5.0  
## 67 smelt 6.7 9.3 9.8 10.8 16.1 9.7 0.5 1.0 1.5  
## 68 smelt 7.5 10.0 10.5 11.6 17.0 10.0 0.5 1.1 1.6  
## 69 smelt 7.0 10.1 10.6 11.6 14.9 9.9 0.5 1.0 1.5  
## 70 smelt 9.7 10.4 11.0 12.0 18.3 11.5 0.6 1.0 1.6  
## 71 smelt 10.0 11.3 11.8 13.1 16.9 9.8 0.5 1.3 1.8  
## 72 smelt 9.9 11.3 11.8 13.1 16.9 8.9 0.5 1.3 1.8  
## 73 smelt 9.8 11.4 12.0 13.2 16.7 8.7 0.6 1.2 1.8  
## 74 smelt 12.2 11.5 12.2 13.4 15.6 10.4 0.7 1.2 1.9  
## 75 smelt 13.4 11.7 12.4 13.5 18.0 9.4 0.7 1.1 1.8  
## 76 smelt 12.2 12.1 13.0 13.8 16.5 9.1 0.9 0.8 1.7  
## 77 smelt 19.7 13.2 14.3 15.2 18.9 13.6 1.1 0.9 2.0  
## 78 smelt 19.9 13.8 15.0 16.2 18.1 11.6 1.2 1.2 2.4  
## 79 pike 200.0 30.0 32.3 34.8 16.0 9.7 2.3 2.5 4.8  
## 80 pike 300.0 31.7 34.0 37.8 15.1 11.0 2.3 3.8 6.1  
## 81 pike 300.0 32.7 35.0 38.8 15.3 11.3 2.3 3.8 6.1  
## 82 pike 300.0 34.8 37.3 39.8 15.8 10.1 2.5 2.5 5.0  
## 83 pike 430.0 35.5 38.0 40.5 18.0 11.3 2.5 2.5 5.0  
## 84 pike 456.0 40.0 42.5 45.5 16.0 9.5 2.5 3.0 5.5  
## 85 pike 510.0 40.0 42.5 45.5 15.0 9.8 2.5 3.0 5.5  
## 86 pike 540.0 40.1 43.0 45.8 17.0 11.2 2.9 2.8 5.7  
## 87 pike 500.0 42.0 45.0 48.0 14.5 10.2 3.0 3.0 6.0  
## 88 pike 567.0 43.2 46.0 48.7 16.0 10.0 2.8 2.7 5.5  
## 89 pike 770.0 44.8 48.0 51.2 15.0 10.5 3.2 3.2 6.4  
## 90 pike 950.0 48.3 51.7 55.1 16.2 11.2 3.4 3.4 6.8  
## 91 pike 1250.0 52.0 56.0 59.7 17.9 11.7 4.0 3.7 7.7  
## 92 pike 1600.0 56.0 60.0 64.0 15.0 9.6 4.0 4.0 8.0  
## 93 pike 1550.0 56.0 60.0 64.0 15.0 9.6 4.0 4.0 8.0  
## 94 pike 1650.0 59.0 63.4 68.0 15.9 11.0 4.4 4.6 9.0  
## 95 perch 5.9 7.5 8.4 8.8 24.0 16.0 0.9 0.4 1.3  
## 96 perch 32.0 12.5 13.7 14.7 24.0 13.6 1.2 1.0 2.2  
## 97 perch 40.0 13.8 15.0 16.0 23.9 15.2 1.2 1.0 2.2  
## 98 perch 51.5 15.0 16.2 17.2 26.7 15.3 1.2 1.0 2.2  
## 99 perch 70.0 15.7 17.4 18.5 24.8 15.9 1.7 1.1 2.8  
## 100 perch 100.0 16.2 18.0 19.2 27.2 17.3 1.8 1.2 3.0  
## 101 perch 78.0 16.8 18.7 19.4 26.8 16.1 1.9 0.7 2.6  
## 102 perch 80.0 17.2 19.0 20.2 27.9 15.1 1.8 1.2 3.0  
## 103 perch 85.0 17.8 19.6 20.8 24.7 14.6 1.8 1.2 3.0  
## 104 perch 85.0 18.2 20.0 21.0 24.2 13.2 1.8 1.0 2.8  
## 105 perch 110.0 19.0 21.0 22.5 25.3 15.8 2.0 1.5 3.5  
## 106 perch 115.0 19.0 21.0 22.5 26.3 14.7 2.0 1.5 3.5  
## 107 perch 125.0 19.0 21.0 22.5 25.3 16.3 2.0 1.5 3.5  
## 108 perch 130.0 19.3 21.3 22.8 28.0 15.5 2.0 1.5 3.5  
## 109 perch 120.0 20.0 22.0 23.5 26.0 14.5 2.0 1.5 3.5  
## 110 perch 120.0 20.0 22.0 23.5 24.0 15.0 2.0 1.5 3.5  
## 111 perch 130.0 20.0 22.0 23.5 26.0 15.0 2.0 1.5 3.5  
## 112 perch 135.0 20.0 22.0 23.5 25.0 15.0 2.0 1.5 3.5  
## 113 perch 110.0 20.0 22.0 23.5 23.5 17.0 2.0 1.5 3.5  
## 114 perch 130.0 20.5 22.5 24.0 24.4 15.1 2.0 1.5 3.5  
## 115 perch 150.0 20.5 22.5 24.0 28.3 15.1 2.0 1.5 3.5  
## 116 perch 145.0 20.7 22.7 24.2 24.6 15.0 2.0 1.5 3.5  
## 117 perch 150.0 21.0 23.0 24.5 21.3 14.8 2.0 1.5 3.5  
## 118 perch 170.0 21.5 23.5 25.0 25.1 14.9 2.0 1.5 3.5  
## 119 perch 225.0 22.0 24.0 25.5 28.6 14.6 2.0 1.5 3.5  
## 120 perch 145.0 22.0 24.0 25.5 25.0 15.0 2.0 1.5 3.5  
## 121 perch 188.0 22.6 24.6 26.2 25.7 15.9 2.0 1.6 3.6  
## 122 perch 180.0 23.0 25.0 26.5 24.3 13.9 2.0 1.5 3.5  
## 123 perch 197.0 23.5 25.6 27.0 24.3 15.7 2.1 1.4 3.5  
## 124 perch 218.0 25.0 26.5 28.0 25.6 14.8 1.5 1.5 3.0  
## 125 perch 300.0 25.2 27.3 28.7 29.0 17.9 2.1 1.4 3.5  
## 126 perch 260.0 25.4 27.5 28.9 24.8 15.0 2.1 1.4 3.5  
## 127 perch 265.0 25.4 27.5 28.9 24.4 15.0 2.1 1.4 3.5  
## 128 perch 250.0 25.4 27.5 28.9 25.2 15.8 2.1 1.4 3.5  
## 129 perch 250.0 25.9 28.0 29.4 26.6 14.3 2.1 1.4 3.5  
## 130 perch 300.0 26.9 28.7 30.1 25.2 15.4 1.8 1.4 3.2  
## 131 perch 320.0 27.8 30.0 31.6 24.1 15.1 2.2 1.6 3.8  
## 132 perch 514.0 30.5 32.8 34.0 29.5 17.7 2.3 1.2 3.5  
## 133 perch 556.0 32.0 34.5 36.5 28.1 17.5 2.5 2.0 4.5  
## 134 perch 840.0 32.5 35.0 37.3 30.8 20.9 2.5 2.3 4.8  
## 135 perch 685.0 34.0 36.5 39.0 27.9 17.6 2.5 2.5 5.0  
## 136 perch 700.0 34.0 36.0 38.3 27.7 17.6 2.0 2.3 4.3  
## 137 perch 700.0 34.5 37.0 39.4 27.5 15.9 2.5 2.4 4.9  
## 138 perch 690.0 34.6 37.0 39.3 26.9 16.2 2.4 2.3 4.7  
## 139 perch 900.0 36.5 39.0 41.4 26.9 18.1 2.5 2.4 4.9  
## 140 perch 650.0 36.5 39.0 41.4 26.9 14.5 2.5 2.4 4.9  
## 141 perch 820.0 36.6 39.0 41.3 30.1 17.8 2.4 2.3 4.7  
## 142 perch 850.0 36.9 40.0 42.3 28.2 16.8 3.1 2.3 5.4  
## 143 perch 820.0 37.1 40.0 42.5 26.2 15.6 2.9 2.5 5.4  
## 144 perch 1100.0 39.0 42.0 44.6 28.7 15.4 3.0 2.6 5.6  
## 145 perch 1000.0 39.8 43.0 45.2 26.4 16.1 3.2 2.2 5.4  
## 146 perch 1100.0 40.1 43.0 45.5 27.5 16.3 2.9 2.5 5.4  
## 147 perch 1000.0 40.2 43.5 46.0 27.4 17.7 3.3 2.5 5.8  
## 148 perch 1000.0 41.1 44.0 46.6 26.8 16.3 2.9 2.6 5.5

newfish.lda<-lda(Species~.,data=newfish)

## Warning in lda.default(x, grouping, ...): variables are collinear

newfish.lda<-lda(Species~Weight+L1+Height+Width+L21+L32,data=newfish)  
newfish.lda

## Call:  
## lda(Species ~ Weight + L1 + Height + Width + L21 + L32, data = newfish)  
##   
## Prior probabilities of groups:  
## bream parki perch pike roach smelt   
## 0.22297297 0.06756757 0.36486486 0.10810811 0.12162162 0.08108108   
## white   
## 0.03378378   
##   
## Group means:  
## Weight L1 Height Width L21 L32  
## bream 636.1818 30.60606 39.52727 14.10000 2.8060606 5.272727  
## parki 155.8000 18.62000 39.20000 14.18000 1.6100000 2.430000  
## perch 360.9333 25.31852 26.17778 15.78519 2.1259259 1.650000  
## pike 742.0625 42.88125 15.85625 10.48125 3.0375000 3.281250  
## roach 159.1111 20.66667 26.88333 14.57222 1.6388889 2.716667  
## smelt 11.5000 11.34167 16.99167 10.21667 0.6916667 1.091667  
## white 477.2000 27.82000 29.10000 15.76000 2.4800000 2.960000  
##   
## Coefficients of linear discriminants:  
## LD1 LD2 LD3 LD4 LD5  
## Weight 0.000911022 -0.002710071 0.007553399 0.001688806 0.006182751  
## L1 0.132200166 0.036926540 -0.259794107 -0.235599786 -0.330471903  
## Height -0.618519868 -0.332732865 -0.053863042 -0.330737436 -0.029226039  
## Width 0.464670922 -0.341184928 -0.353062958 0.842951264 -0.201141743  
## L21 -0.114071841 0.712452136 -2.278059990 0.277900320 2.700516892  
## L32 -2.311243186 2.141452146 0.539501848 1.803654269 -0.461925634  
## LD6  
## Weight -0.003600115  
## L1 -0.119589009  
## Height -0.019796935  
## Width -0.159484049  
## L21 2.813216431  
## L32 -0.080912628  
##   
## Proportion of trace:  
## LD1 LD2 LD3 LD4 LD5 LD6   
## 0.7998 0.1327 0.0473 0.0167 0.0035 0.0000

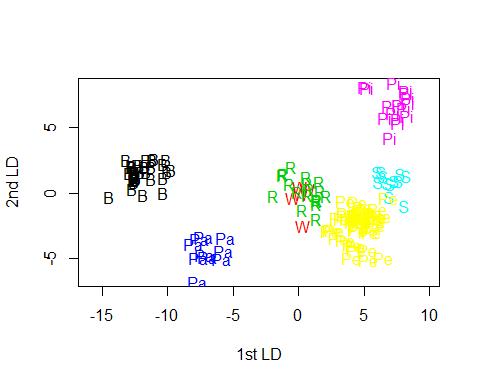
newfish.ldapred<-predict(newfish.lda,newfish[,-1])  
table(newfish$Species,newfish.ldapred$class)

##   
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 1 0 4

newfish.ldacv<-lda(Species~Weight+L1+Height+Width+L21+L32,data=newfish,CV=T)  
table(newfish$Species,newfish.ldacv$class)

##   
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 1 0 4

# The true error rate remains to be 0.6%  
eqscplot(newfish.ldapred$x,type="n",xlab="1st LD",ylab="2nd LD")  
fish.species <- c(rep("B",33),rep("W",5),rep("R",18),rep("Pa",10),rep("S",12),rep("Pi",16),rep("Pe",54))  
fish.colors <- c(rep(1,33),rep(2,5),rep(3,18),rep(4,10),rep(5,12),rep(6,16),rep(7,54))  
text(newfish.ldapred$x[,1:2],fish.species,col=fish.colors)



#To predict the class identities of the new data points we use:  
newfish.ldatest<-predict(newfish.lda,newfish.test)  
newfish.ldatest$class

## [1] bream bream perch perch pike smelt smelt parki roach roach white  
## Levels: bream parki perch pike roach smelt white

#We see that the results agree with those obtained from the classification tree.  
#Let us examine how to apply QDA to this dataset.

# Quadratic Discriminant Analysis

#newfish.qda<-qda(Species~.,data=newfish)  
newfish.q<-read.table("file:///C:/Users/Asus/Documents/GitHub/classifng\_fish/newfish.qdata.txt",h=T)  
library(MVN)

## sROC 0.1-2 loaded

#Running (i) Mardia's; (ii) Henze-Zirkler's and (iii) Royston's Multivariate  
#Normality Test:  
mvn(data = newfish.q[,-c(1,8,9,10)], mvnTest = "ma")

## $multivariateNormality  
## Test Statistic p value Result  
## 1 Mardia Skewness 426.417978948719 2.01256215659792e-58 NO  
## 2 Mardia Kurtosis 1.58569973539399 0.112807439232689 YES  
## 3 MVN <NA> <NA> NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Shapiro-Wilk Weight 0.8780 <0.001 NO   
## 2 Shapiro-Wilk L1 0.9679 0.0019 NO   
## 3 Shapiro-Wilk L2 0.9702 0.0033 NO   
## 4 Shapiro-Wilk L3 0.9703 0.0033 NO   
## 5 Shapiro-Wilk Height 0.9125 <0.001 NO   
## 6 Shapiro-Wilk Width 0.9344 <0.001 NO   
##   
## $Descriptives  
## n Mean Std.Dev Median Min Max 25th 75th Skew  
## Weight 143 398.02378 360.51374 272.0 5.9 1650.0 120.00 650.0 1.1175913  
## L1 143 26.27692 10.16502 25.2 7.5 59.0 19.00 32.6 0.6215598  
## L2 143 28.44406 10.88848 27.3 8.4 63.4 20.90 35.0 0.5796666  
## L3 143 31.25455 11.82529 29.2 8.8 68.0 22.80 39.6 0.4306508  
## Height 143 28.33217 8.39113 26.8 14.5 44.5 24.25 37.8 0.1335468  
## Width 143 14.07063 2.23310 14.6 8.7 20.9 13.40 15.3 -0.4962807  
## Kurtosis  
## Weight 0.89652958  
## L1 0.35891664  
## L2 0.32535350  
## L3 -0.02200252  
## Height -1.07931595  
## Width 0.27826140

mvn(data = newfish.q[,-c(1,8,9,10)], mvnTest = "hz")

## $multivariateNormality  
## Test HZ p value MVN  
## 1 Henze-Zirkler 4.496681 0 NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Shapiro-Wilk Weight 0.8780 <0.001 NO   
## 2 Shapiro-Wilk L1 0.9679 0.0019 NO   
## 3 Shapiro-Wilk L2 0.9702 0.0033 NO   
## 4 Shapiro-Wilk L3 0.9703 0.0033 NO   
## 5 Shapiro-Wilk Height 0.9125 <0.001 NO   
## 6 Shapiro-Wilk Width 0.9344 <0.001 NO   
##   
## $Descriptives  
## n Mean Std.Dev Median Min Max 25th 75th Skew  
## Weight 143 398.02378 360.51374 272.0 5.9 1650.0 120.00 650.0 1.1175913  
## L1 143 26.27692 10.16502 25.2 7.5 59.0 19.00 32.6 0.6215598  
## L2 143 28.44406 10.88848 27.3 8.4 63.4 20.90 35.0 0.5796666  
## L3 143 31.25455 11.82529 29.2 8.8 68.0 22.80 39.6 0.4306508  
## Height 143 28.33217 8.39113 26.8 14.5 44.5 24.25 37.8 0.1335468  
## Width 143 14.07063 2.23310 14.6 8.7 20.9 13.40 15.3 -0.4962807  
## Kurtosis  
## Weight 0.89652958  
## L1 0.35891664  
## L2 0.32535350  
## L3 -0.02200252  
## Height -1.07931595  
## Width 0.27826140

mvn(data = newfish.q[,-c(1,8,9,10)], mvnTest = "royston")

## $multivariateNormality  
## Test H p value MVN  
## 1 Royston 46.25164 2.211833e-10 NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Shapiro-Wilk Weight 0.8780 <0.001 NO   
## 2 Shapiro-Wilk L1 0.9679 0.0019 NO   
## 3 Shapiro-Wilk L2 0.9702 0.0033 NO   
## 4 Shapiro-Wilk L3 0.9703 0.0033 NO   
## 5 Shapiro-Wilk Height 0.9125 <0.001 NO   
## 6 Shapiro-Wilk Width 0.9344 <0.001 NO   
##   
## $Descriptives  
## n Mean Std.Dev Median Min Max 25th 75th Skew  
## Weight 143 398.02378 360.51374 272.0 5.9 1650.0 120.00 650.0 1.1175913  
## L1 143 26.27692 10.16502 25.2 7.5 59.0 19.00 32.6 0.6215598  
## L2 143 28.44406 10.88848 27.3 8.4 63.4 20.90 35.0 0.5796666  
## L3 143 31.25455 11.82529 29.2 8.8 68.0 22.80 39.6 0.4306508  
## Height 143 28.33217 8.39113 26.8 14.5 44.5 24.25 37.8 0.1335468  
## Width 143 14.07063 2.23310 14.6 8.7 20.9 13.40 15.3 -0.4962807  
## Kurtosis  
## Weight 0.89652958  
## L1 0.35891664  
## L2 0.32535350  
## L3 -0.02200252  
## Height -1.07931595  
## Width 0.27826140

#資料不是多元常態  
#newfish.qda<-qda(Species~.,data=newfish.q)  
newfish.qda<-qda(Species~Weight+L1+Height+Width+L21+L32,data=newfish.q)  
newfish.qdapred<-predict(newfish.qda,newfish.q)  
predict(newfish.qda,newfish.test)$class

## [1] bream bream perch perch pike smelt smelt parki roach roach perch  
## Levels: bream parki perch pike roach smelt

newfish.qda<-qda(Species~Weight+L1+Height+Width+L21+L32,data=newfish.q,CV=T)  
table(newfish.q$Species,newfish.qda$class)

##   
## bream parki perch pike roach smelt  
## bream 33 0 0 0 0 0  
## parki 0 10 0 0 0 0  
## perch 0 0 54 0 0 0  
## pike 0 0 0 16 0 0  
## roach 0 0 1 0 17 0  
## smelt 0 0 1 0 0 11

# Nearest Neighbor Methods

library(class)  
newfish.knn <- knn(newfish[,2:10],newfish[,2:10],newfish[,"Species"],k=3,prob=T)  
table(newfish$Species,newfish.knn)

## newfish.knn  
## bream parki perch pike roach smelt white  
## bream 30 1 2 0 0 0 0  
## parki 1 6 1 0 2 0 0  
## perch 3 0 47 0 2 1 1  
## pike 1 0 3 11 1 0 0  
## roach 1 0 8 0 8 0 1  
## smelt 0 0 0 0 0 12 0  
## white 0 0 2 1 0 0 2

#We see that the apparent error rate for k = 3 is about 21%. For k = 2, we have:  
newfish.knn<-knn(newfish[,2:10],newfish[,2:10],newfish[,"Species"],k=2,prob=T)  
table(newfish$Species,newfish.knn)

## newfish.knn  
## bream parki perch pike roach smelt white  
## bream 27 0 3 0 2 0 1  
## parki 1 7 0 0 2 0 0  
## perch 3 0 45 0 4 1 1  
## pike 1 0 2 13 0 0 0  
## roach 0 0 8 0 9 0 1  
## smelt 0 0 0 0 0 12 0  
## white 0 0 2 0 0 0 3

#k=1  
newfish.knn <- knn(newfish[,2:10],newfish[,2:10],newfish[,"Species"],k=1,prob=T)  
table(newfish$Species,newfish.knn)

## newfish.knn  
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

newfish1 <- newfish[,c(1,2,3,6,8,9)]  
newfish.knncv <- knn.cv(newfish1[,2:6],newfish1[,"Species"],k=1,prob=T)  
table(newfish1$Species,newfish.knncv)

## newfish.knncv  
## bream parki perch pike roach smelt white  
## bream 26 0 4 0 2 0 1  
## parki 1 4 0 0 4 0 1  
## perch 3 0 37 0 11 1 2  
## pike 2 0 4 9 0 0 1  
## roach 2 0 10 0 5 0 1  
## smelt 0 0 0 0 0 12 0  
## white 0 0 3 0 0 0 2

newfish1.test<-newfish.test[,c(1,2,5,7,8)]  
newfish.knntest<-knn(newfish1[,2:6],newfish1.test,newfish1[,"Species"],k=1,prob=T)  
newfish.knntest

## [1] bream bream perch white perch smelt smelt parki perch perch perch  
## attr(,"prob")  
## [1] 1 1 1 1 1 1 1 1 1 1 1  
## Levels: bream parki perch pike roach smelt white

# Logistic Discrimination

library(nnet)  
newfish.logd<-multinom(Species~.,data=newfish,maxit=250)

## # weights: 77 (60 variable)  
## initial value 287.994702   
## iter 10 value 189.100680  
## iter 20 value 82.739762  
## iter 30 value 15.668415  
## iter 40 value 0.165377  
## iter 50 value 0.003851  
## final value 0.000000   
## converged

newfish.logd

## Call:  
## multinom(formula = Species ~ ., data = newfish, maxit = 250)  
##   
## Coefficients:  
## (Intercept) Weight L1 L2 L3 Height  
## parki -29.45533 0.02917110 6.349592 17.8259067 -23.500970 9.645257  
## perch -80.11405 0.16021628 3.267803 56.6489218 -53.765483 6.684178  
## pike 15.22567 -0.05874368 8.093673 0.9753102 -3.095179 -13.084687  
## roach -277.16410 -0.51539078 54.195310 -43.6844449 4.362472 -2.952463  
## smelt 455.64639 0.18459382 29.363751 -20.5072505 -10.290211 -13.228223  
## white -57.01255 0.19991067 -17.467222 31.7667561 -20.454096 -4.118171  
## Width L21 L32 L31  
## parki 3.247584 11.476314 -41.326877 -29.850563  
## perch 21.052273 53.381119 -110.414404 -57.033286  
## pike 21.652958 -7.118363 -4.070489 -11.188852  
## roach 40.080837 -97.879755 48.046917 -49.832838  
## smelt 18.368009 -49.871001 10.217040 -39.653961  
## white 26.549555 49.233978 -52.220852 -2.986874  
##   
## Residual Deviance: 2.009681e-11   
## AIC: 84

table(newfish$Species,predict(newfish.logd,newfish))

##   
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16

x <- as.matrix(newfish[,-1])  
y <- newfish$Species  
cvfit <- cv.glmnet(x, y, family="multinomial", type.measure="class", nfolds=148)  
predict.value <- predict(cvfit, x, s = "lambda.min", type = "class")  
table(predict.value,newfish$Species)

##   
## predict.value bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

predict(newfish.logd,newfish.test)

## [1] bream bream perch perch pike smelt smelt parki roach roach white  
## Levels: bream parki perch pike roach smelt white

Github: <https://github.com/CaoCharles/classifng_fish>