Supplement-zc2556

Zhe Chen

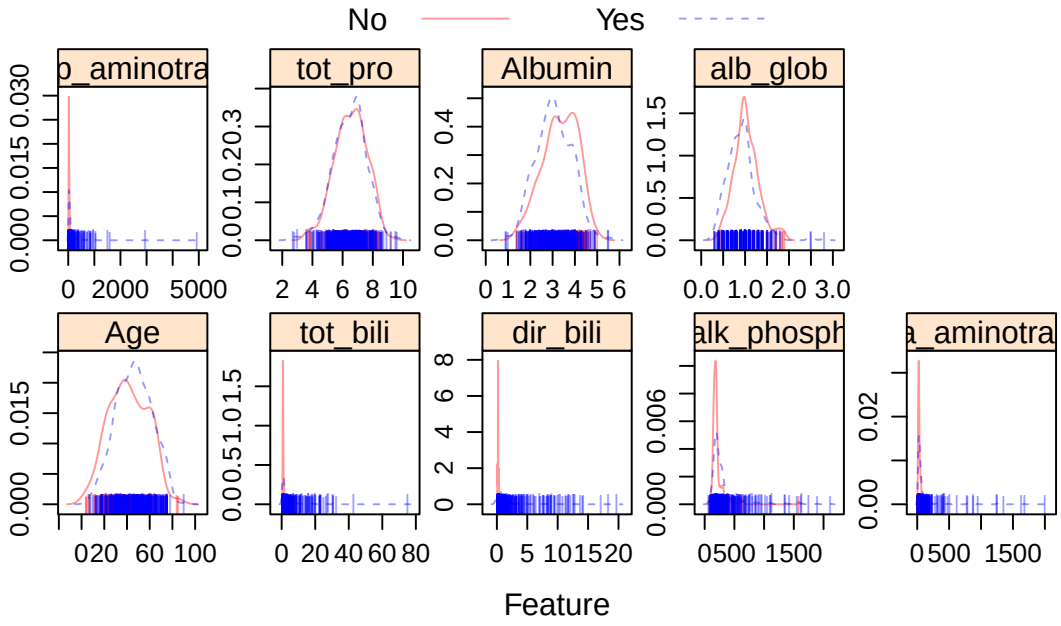
2021/03/30

Data Preparation

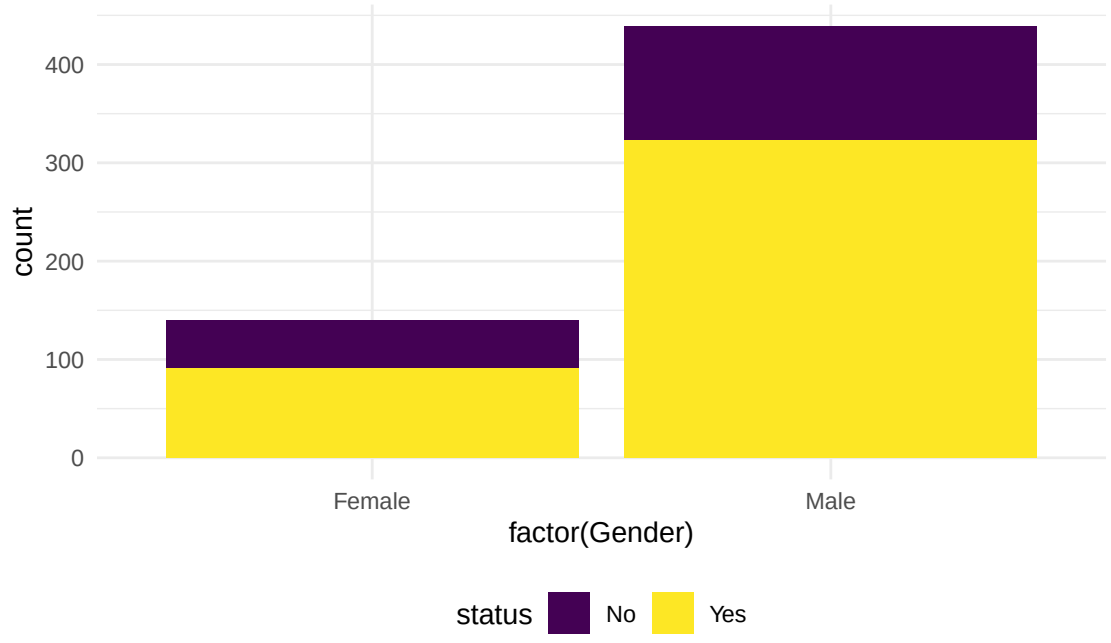
```
liver.df = read.csv("./indian_liver_patient.csv") %>%
  mutate(
    status  = ifelse(Dataset == 1, "Yes", "No"),
    status = as.factor(status),
  ) %>%
  rename(
    tot_bili = Total_Bilirubin,
    dir_bili = Direct_Bilirubin,
    alk_phosph = Alkaline_Phosphotase,
    ala_aminotrans = Alamine_Aminotransferase,
    asp_aminotrans = Aspartate_Aminotransferase,
    tot_pro = Total_Protiens,
    alb_glob = Albumin_and_Globulin_Ratio
  )%>%
  na.omit() %>%
  dplyr::select(-Dataset)

set.seed(621)
liver.t = createDataPartition(y = liver.df$status,
                                  p = 0.8,
                                  list = FALSE)
liver.train = liver.df[liver.t, ]
liver.test = liver.df[-liver.t, ]
```

Exploratory Plot

```
#exploratory plot to check the assumptions
theme1 <- transparentTheme(trans = .4)
trellis.par.set(theme1)
featurePlot(x = liver.df[, c(1,3:10)],
  y = liver.df$status,
  scales = list(x = list(relation = "free"),
                y = list(relation = "free")),
  plot = "density", pch = "|",
  auto.key = list(columns = 2))
```




```
#plot for gender
ggplot(liver.df, aes(factor(Gender), fill = status))+
  geom_bar()
```




```
#test for gender differences
fm.trail = liver.df %>% filter(Gender == "Female") %>%
  nrow()
fm.case = liver.df %>% filter(Gender == "Female" & status == "Yes") %>%
  nrow()
```

```
m.trail = liver.df %>% filter(Gender == "Male") %>%
  nrow()
m.case = liver.df %>% filter(Gender == "Male" & status == "Yes") %>%
  nrow()
prop.test(x=c(fm.case,m.case), n=c(fm.trail, m.trail),
          conf.level=0.95)
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(fm.case, m.case) out of c(fm.trail, m.trail)
## X-squared = 3.4223, df = 1, p-value = 0.06432
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.179600360  0.008074164
## sample estimates:
##    prop 1    prop 2
## 0.6500000 0.7357631
```

Transformation

```
preprocesspred <- preProcess(liver.df[,3:7], method=c("BoxCox"))
print(preprocesspred)
```

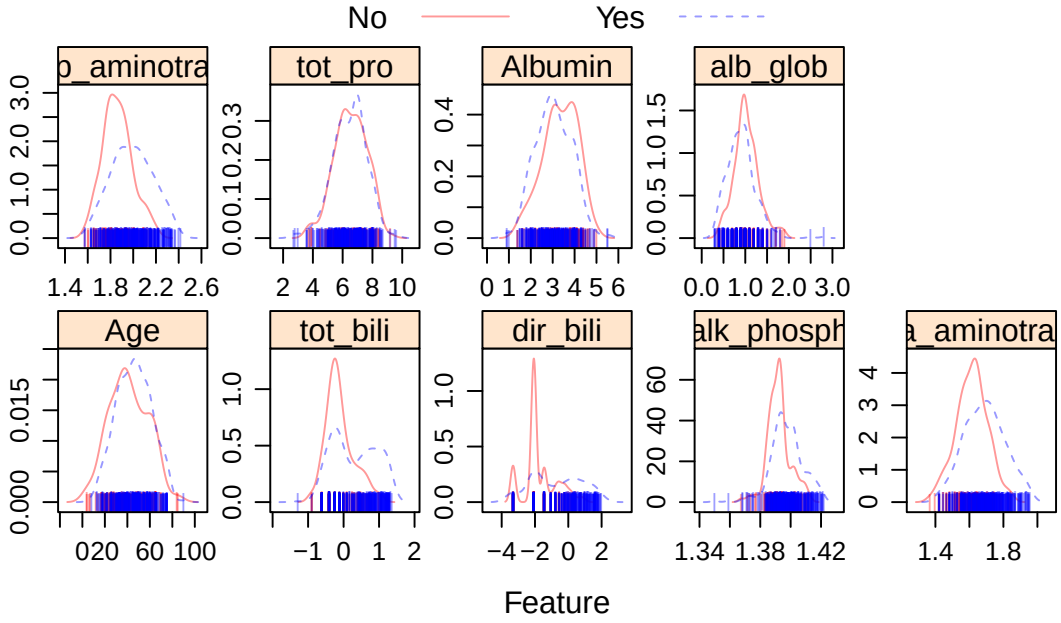
```
## Created from 579 samples and 5 variables
##
## Pre-processing:
##   - Box-Cox transformation (5)
##   - ignored (0)
##
## Lambda estimates for Box-Cox transformation:
## -0.7, -0.3, -0.7, -0.5, -0.4
```

```
liver.df.boxcox <- predict(preprocesspred, liver.df)
liver.train.boxcox <- predict(preprocesspred, liver.train)
liver.test.boxcox <- predict(preprocesspred, liver.test)
summary(liver.train.boxcox)
```

```
##       Age          Gender             tot_bili          dir_bili
##  Min.   : 4.0   Length:464         Min.   :-1.2845   Min.   :-3.3175
##  1st Qu.:33.0   Class :character   1st Qu.:-0.2415   1st Qu.:-2.0689
##  Median :45.0   Mode  :character   Median : 0.0000   Median :-1.4501
##  Mean   :44.6                      Mean   : 0.1868   Mean   :-1.0003
##  3rd Qu.:57.0                      3rd Qu.: 0.7158   3rd Qu.: 0.2523
##  Max.   :90.0                      Max.   : 1.3590   Max.   : 1.9702
##    alk_phosph    ala_aminotrans  asp_aminotrans     tot_pro
##  Min.   :1.350   Min.   :1.368   Min.   :1.575   Min.   :2.700
##  1st Qu.:1.390   1st Qu.:1.583   1st Qu.:1.821   1st Qu.:5.800
##  Median :1.395   Median :1.662   Median :1.937   Median :6.600
##  Mean   :1.396   Mean   :1.669   Mean   :1.955   Mean   :6.479
```

```
##  3rd Qu.:1.402   3rd Qu.:1.744   3rd Qu.:2.083   3rd Qu.:7.200
##  Max.   :1.422   Max.   :1.955   Max.   :2.417   Max.   :9.600
##     Albumin         alb_glob      status
##  Min.   :0.900   Min.   :0.3000   No :132
##  1st Qu.:2.500   1st Qu.:0.7000   Yes:332
##  Median :3.100   Median :0.9000
##  Mean   :3.127   Mean   :0.9417
##  3rd Qu.:3.800   3rd Qu.:1.1000
##  Max.   :5.500   Max.   :2.8000
```

```
featurePlot(x = liver.train.boxcox[, c(1,3:10)],
  y = liver.train.boxcox$status,
  scales = list(x = list(relation = "free"),
                y = list(relation = "free")),
  plot = "density", pch = "|",
  auto.key = list(columns = 2))
```



Model Building

Feature Selection

```
log.fit = glm(status ~ ., data = liver.train.boxcox,family = binomial(link = "logit"))
summary(log.fit)
```

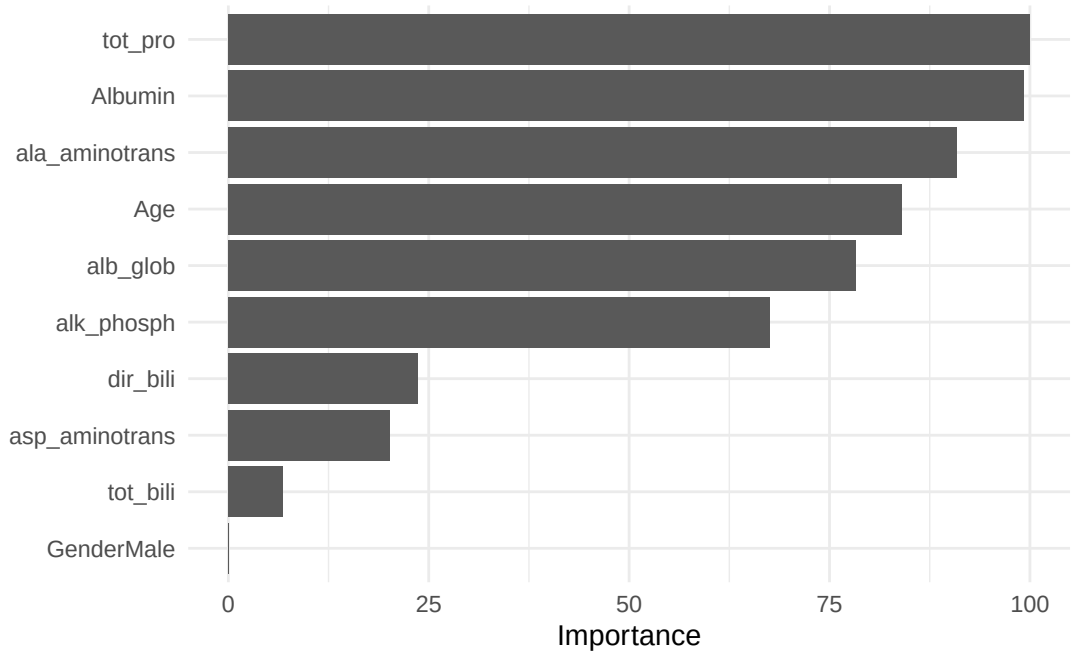
```
##
## Call:
## glm(formula = status ~ ., family = binomial(link = "logit"),
##     data = liver.train.boxcox)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3096  -0.9950   0.4138   0.8246   1.6669
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -52.178881  19.599619  -2.662  0.00776 **
## Age              0.018305   0.007281   2.514  0.01194 *
## GenderMale      -0.023442   0.265936  -0.088  0.92976
## tot_bili         0.208988   0.733432   0.285  0.77569
## dir_bili         0.222032   0.288827   0.769  0.44205
## alk_phosph      28.842930  14.154091   2.038  0.04157 *
## ala_aminotrans   4.574380   1.686871   2.712  0.00669 **
## asp_aminotrans   0.752375   1.122608   0.670  0.50273
## tot_pro          1.326969   0.446144   2.974  0.00294 **
## Albumin         -2.596596   0.879445  -2.953  0.00315 **
## alb_glob         3.198564   1.362463   2.348  0.01889 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 554.14  on 463  degrees of freedom
## Residual deviance: 455.17  on 453  degrees of freedom
## AIC: 477.17
##
## Number of Fisher Scoring iterations: 5
```

```
#logistic regression
set.seed(621)
ctrl <-trainControl(method = "repeatedcv", repeats = 10, summaryFunction = twoClassSummary,classProbs =
glm.fit <- train(x = liver.train.boxcox[,1:10],
                 y = liver.train.boxcox$status,
                 method = "glm",
                 metric = "ROC",
                 trControl = ctrl)
glm.fit
```

```
## Generalized Linear Model
##
## 464 samples
##  10 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 418, 417, 418, 418, 417, 417, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.7495863  0.2913736  0.8876827
```

```r
vip(glm.fit)
```




```r
#prevalence
preval = nrow(filter(liver.df, status == "Yes"))/nrow(filter(liver.df, status == "No"))
preval.test = nrow(filter(liver.train, status == "Yes"))/nrow(filter(liver.train, status == "No"))
preval.train = nrow(filter(liver.test, status == "Yes"))/nrow(filter(liver.test, status == "No"))
```

```r
glm.fit2 <- train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                  y = liver.train.boxcox$status,
                  method = "glm",
                  metric = "ROC",
                  trControl = ctrl)
glm.fit2
```

```
## Generalized Linear Model
##
## 464 samples
##   6 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 417, 417, 418, 418, 418, 417, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.7452946  0.2698901  0.8951693
```

```r
anova(glm.fit2$finalModel, glm.fit$finalModel, test = "LRT")
```

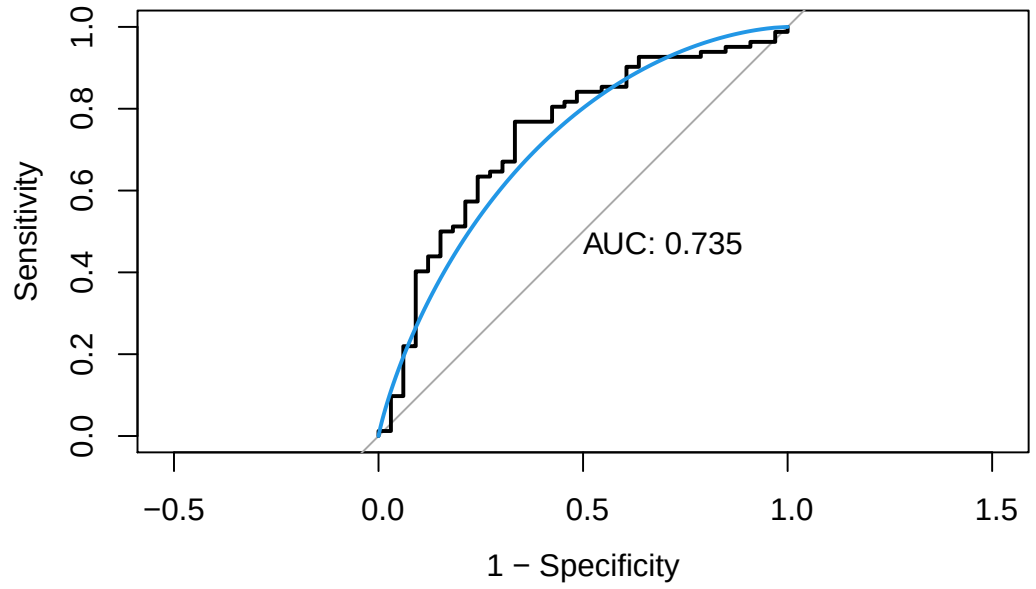
```
## Analysis of Deviance Table
##
## Model 1: .outcome ~ tot_pro + Albumin + ala_aminotrans + Age + alb_glob +
##     alk_phosph
## Model 2: .outcome ~ Age + Gender + tot_bili + dir_bili + alk_phosph +
##     ala_aminotrans + asp_aminotrans + tot_pro + Albumin + alb_glob
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       457     465.95
## 2       453     455.17  4   10.774  0.02922 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#ROC for logistics
test.pred.prob.glm2 <- predict(glm.fit2, newdata = liver.test.boxcox ,type = "prob")[,2]
roc.glm2 <- roc(liver.test.boxcox$status, test.pred.prob.glm2)
```

```
## Setting levels: control = No, case = Yes
```

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

```r
plot(roc.glm2, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm2), col = 4, add = TRUE)
```



LDA and QDA

```r
#LDA
set.seed(621)
model.lda <-train(
  x = liver.train.boxcox[,c(8,9,6,1,10,5)],
  y = liver.train.boxcox$status,
  method = "lda",
  metric = "ROC",
  trControl = ctrl)
model.lda
```

```
## Linear Discriminant Analysis
##
## 464 samples
##   6 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 418, 417, 418, 418, 417, 417, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.7449128  0.2435165  0.9116667
```

```r
#QDA
set.seed(621)
model.qda <-train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                  y = liver.train.boxcox$status,
                  method = "qda",
                  metric = "ROC",
                  trControl = ctrl)
model.qda
```

```
## Quadratic Discriminant Analysis
##
## 464 samples
##   6 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 418, 417, 418, 418, 417, 417, ...
## Resampling results:
##
##   ROC        Sens      Spec
##   0.7201886  0.724011  0.6490463
```

KNN

```
set.seed(621)
model.knn <-train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                  y = liver.train.boxcox$status,
                  method = "knn",
                  metric = "ROC",
                  trControl = ctrl,
                  tuneGrid = data.frame(k=1:12))
model.knn
```

```
## k-Nearest Neighbors
##
## 464 samples
##   6 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 418, 417, 418, 418, 417, 417, ...
## Resampling results across tuning parameters:
##
##   k   ROC        Sens       Spec
##    1  0.6135313  0.4474725  0.7795900
##    2  0.6329099  0.4032967  0.7583779
##    3  0.6527855  0.4035714  0.8205080
##    4  0.6525063  0.3795604  0.8046346
##    5  0.6523965  0.3400000  0.8374332
##    6  0.6568939  0.3089011  0.8334759
##    7  0.6656735  0.2319231  0.8578610
##    8  0.6683748  0.2326923  0.8641800
##    9  0.6678607  0.2060440  0.8677986
##   10  0.6629104  0.2097802  0.8687879
##   11  0.6558820  0.1644505  0.8732353
##   12  0.6499859  0.1566484  0.8855793
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 8.
```

Naive Bey

```
nbGrid <- expand.grid(usekernel = c(FALSE,TRUE),
                      fL = 1,
                      adjust = seq(1, 4, by = .2))

model.nb <- train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                  y = liver.train.boxcox$status,
                  method = "nb",
                  tuneGrid = nbGrid,
                  metric = "ROC",
                  trControl = ctrl)
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

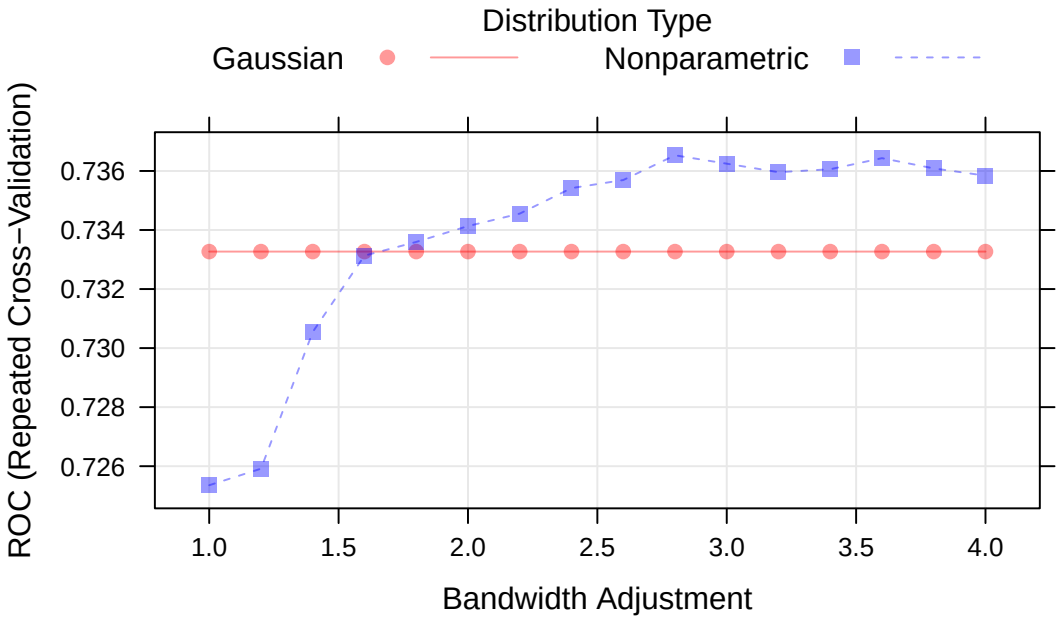
```
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43
```

```
plot(model.nb)
```



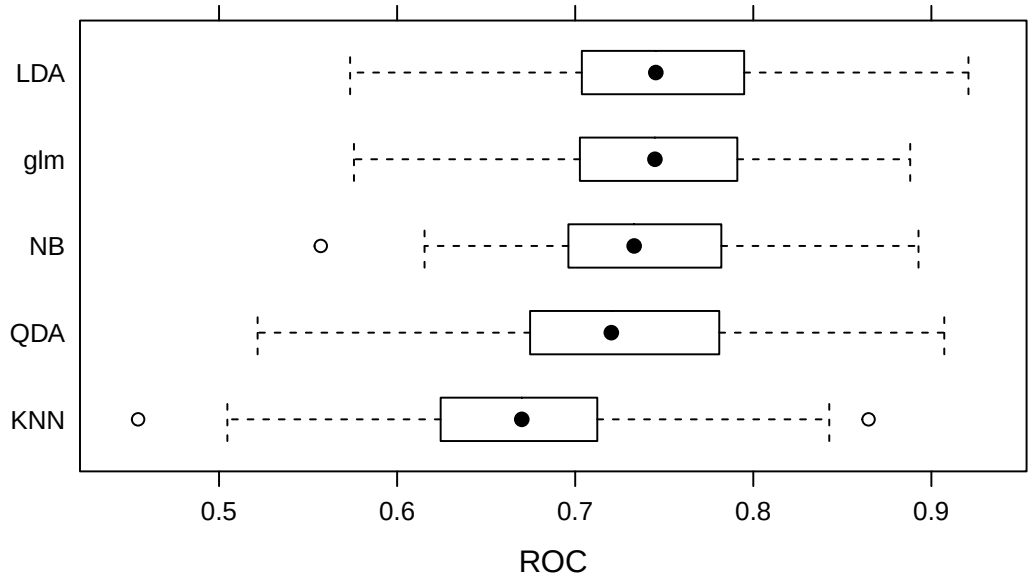
Box Plot of Resamples

```
res = resamples(list(LDA = model.lda, QDA = model.qda, KNN = model.knn, glm = glm.fit2, NB = model.nb))
summary(res)
```

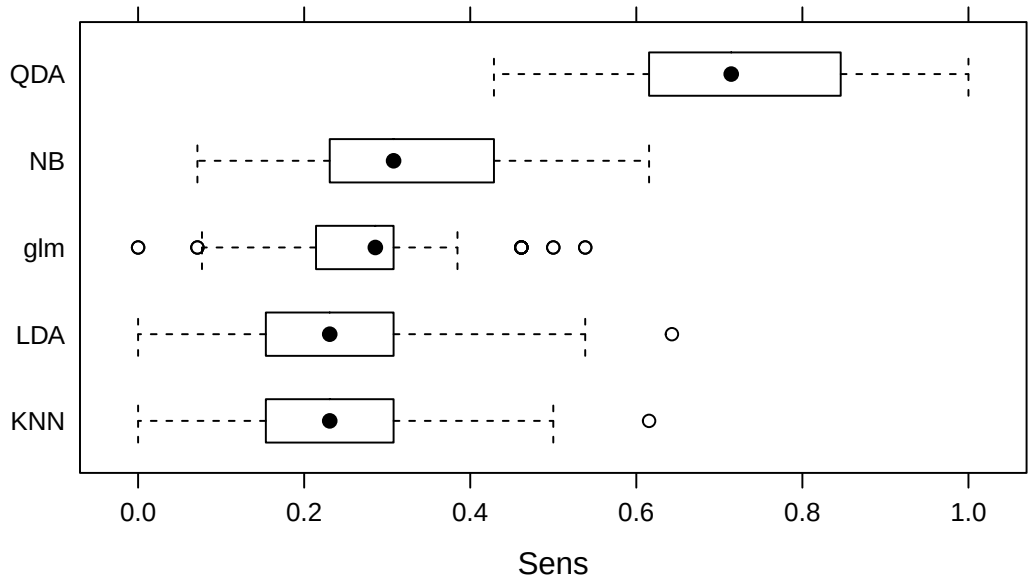
```
##
## Call:
## summary.resamples(object = res)
##
## Models: LDA, QDA, KNN, glm, NB
## Number of resamples: 100
##
## ROC
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LDA 0.5735931 0.7038378 0.7452547 0.7449128 0.7948718 0.9208145    0
## QDA 0.5216450 0.6771611 0.7202797 0.7201886 0.7808858 0.9072398    0
## KNN 0.4545455 0.6250000 0.6699967 0.6683748 0.7122583 0.8647186    0
## glm 0.5757576 0.7031229 0.7447552 0.7452946 0.7906211 0.8881119    0
## NB  0.5571096 0.6965718 0.7331002 0.7365308 0.7814685 0.8927739    0
##
## Sens
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LDA 0.00000000 0.1538462 0.2307692 0.2435165 0.3076923 0.6428571    0
## QDA 0.42857143 0.6153846 0.7142857 0.7240110 0.8461538 1.0000000    0
## KNN 0.00000000 0.1538462 0.2307692 0.2326923 0.3076923 0.6153846    0
## glm 0.00000000 0.2142857 0.2857143 0.2698901 0.3076923 0.5384615    0
## NB  0.07142857 0.2307692 0.3076923 0.3364835 0.4285714 0.6153846    0
##
## Spec
```

```
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LDA 0.7272727 0.8787879 0.9117647 0.9116667 0.9411765 1.0000000    0
## QDA 0.4545455 0.6016043 0.6470588 0.6490463 0.6969697 0.8181818    0
## KNN 0.6969697 0.8181818 0.8529412 0.8641800 0.9090909 1.0000000    0
## glm 0.6666667 0.8529412 0.9090909 0.8951693 0.9393939 1.0000000    0
## NB  0.6666667 0.8181818 0.8658645 0.8624510 0.9090909 0.9696970    0
```

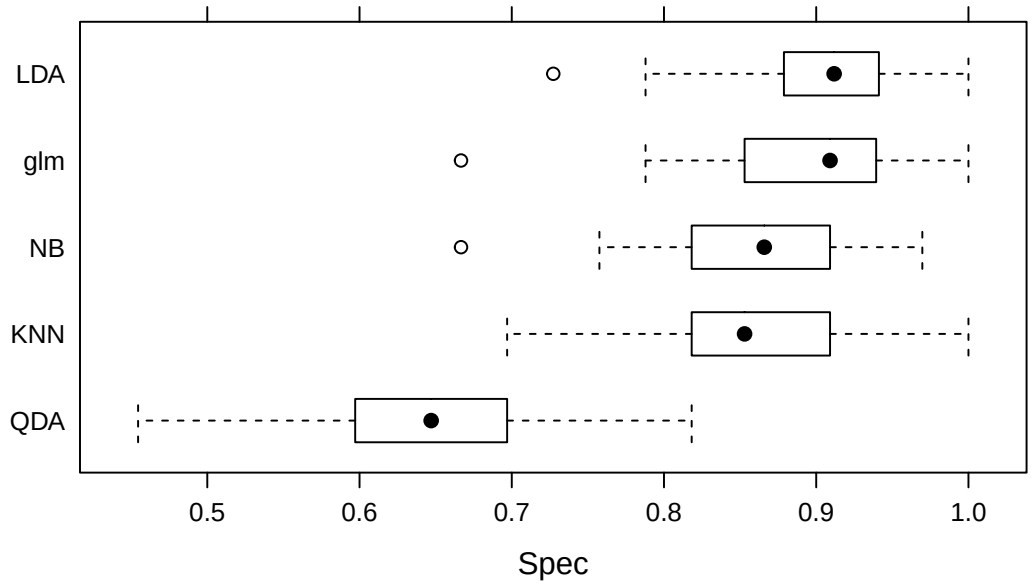
```
bwplot(res, metric = "ROC")
```




```
bwplot(res, metric = "Sens")
```



```
bwplot(res, metric = "Spec")
```



ROC Plot

```
lda.pred <- predict(model.lda, newdata = liver.test.boxcox, type = "prob")[,2]
```

```
## Warning in predict.lda(modelFit, newdata): variable names in 'newdata' do not
## match those in 'object'
```

```
knn.pred <- predict(model.knn, newdata = liver.test.boxcox, type = "prob")[,2]
qda.pred <- predict(model.qda, newdata = liver.test.boxcox, type = "prob")[,2]
```

```
## Warning in predict.qda(modelFit, newdata): variable names in 'newdata' do not
## match those in 'object'
```

```
nb.pred <- predict(model.nb, newdata = liver.test.boxcox, type = "prob")[,2]

roc.lda <- roc(liver.test.boxcox$status, lda.pred)
```

```
## Setting levels: control = No, case = Yes

## Setting direction: controls < cases
```

```
roc.knn <- roc(liver.test.boxcox$status, knn.pred)
```

```
## Setting levels: control = No, case = Yes
## Setting direction: controls < cases
```

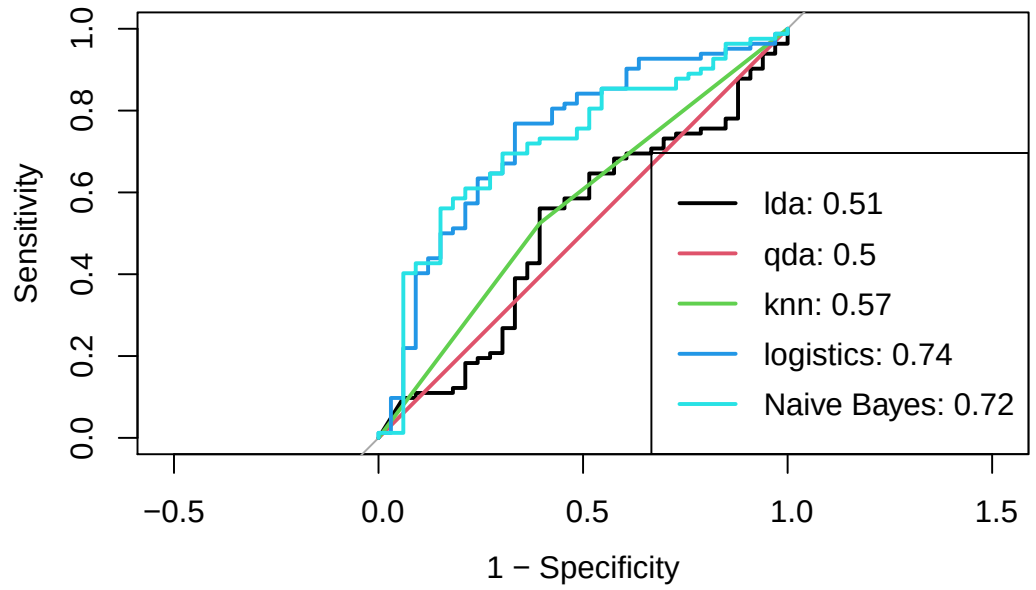
```
roc.qda <- roc(liver.test.boxcox$status, qda.pred)
```

```
## Setting levels: control = No, case = Yes
## Setting direction: controls < cases
```

```
roc.nb <- roc(liver.test.boxcox$status, nb.pred)
```

```
## Setting levels: control = No, case = Yes
## Setting direction: controls < cases
```

```
auc <- c(roc.lda$auc[1], roc.qda$auc[1], roc.knn$auc[1], roc.glm2$auc[1], roc.nb$auc[1])

plot(roc.lda, legacy.axes = TRUE)
plot(roc.qda, col = 2, add = TRUE)
plot(roc.knn, col = 3, add = TRUE)
plot(roc.glm2, col = 4, add = TRUE)
plot(roc.nb, col = 5, add = TRUE)

modelNames <- c("lda","qda","knn","logistics", "Naive Bayes")
legend("bottomright", legend = paste0(modelNames, ": ", round(auc,2)),
col = 1:5, lwd = 2)
```



This shows overfitting for KNN and QDA. Thus, new models needed to be build for KNN and QDA. Now, AUC tested for KNN model and QDA model with different cross validation parameters.

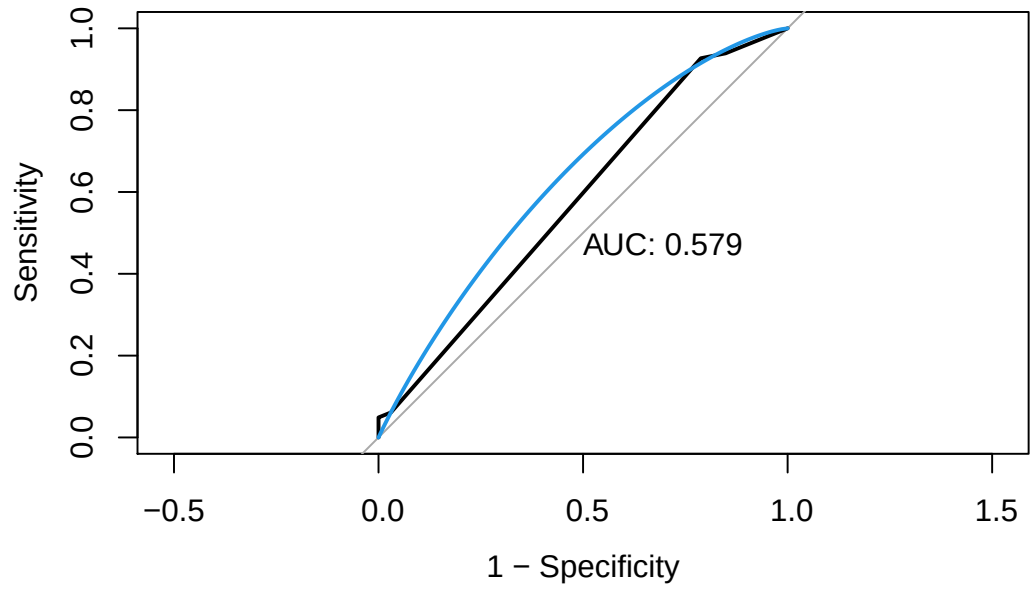
```
#knn
set.seed(126)
model.knn.new <-train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                      y = liver.train.boxcox$status,
                      method = "knn",
```

```
                  metric = "ROC",
                  trControl = trainControl(method = "repeatedcv", repeats = 15, summaryFunction = twoCla

test.pred.prob.knn.new <- predict(model.knn.new, newdata = liver.test.boxcox ,type = "prob")[,2]
roc.knn.new <- roc(liver.test.boxcox$status, test.pred.prob.knn.new)
```

```
## Setting levels: control = No, case = Yes
```

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

```
plot(roc.knn.new, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.knn.new), col = 4, add = TRUE)
```




```
#lda
set.seed(2)
model.lda.new <-train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                  y = liver.train.boxcox$status,
                  method = "lda",
                  metric = "ROC",
                  trControl = trainControl(method = "repeatedcv", repeats = 15, summaryFunction = twoCla

test.pred.prob.lda.new <- predict(model.lda.new, newdata = liver.test.boxcox ,type = "prob")[,2]
```

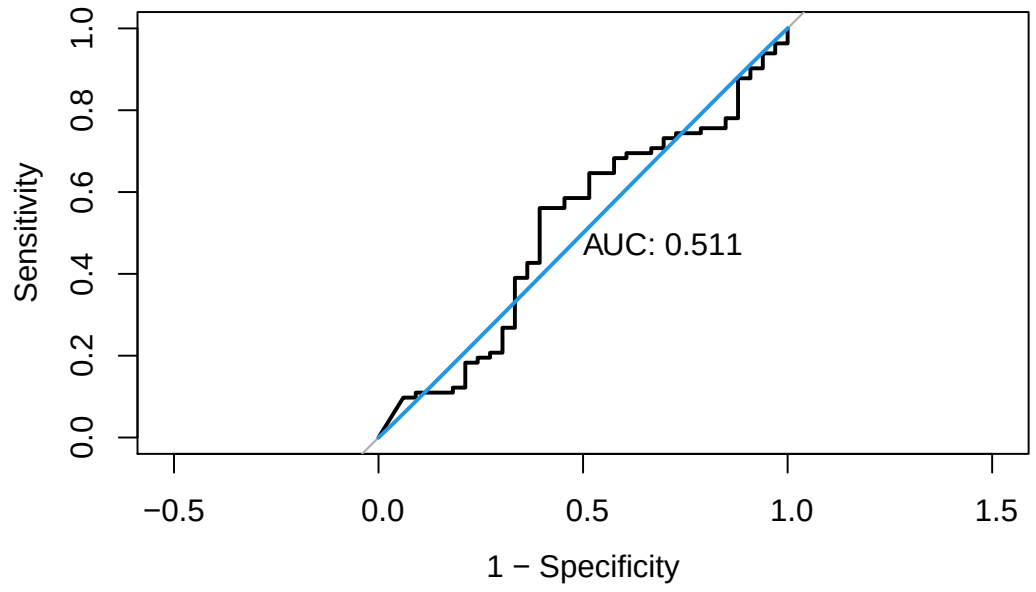
```
## Warning in predict.lda(modelFit, newdata): variable names in 'newdata' do not
## match those in 'object'
```

```
roc.lda.new <- roc(liver.test.boxcox$status, test.pred.prob.lda.new)
```

```
## Setting levels: control = No, case = Yes
```

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

```
plot(roc.lda.new, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.lda.new), col = 4, add = TRUE)
```




```
#qda
set.seed(2)
model.qda.new <-train(x = liver.train.boxcox[,c(8,9,6,1,10,5)],
                      y = liver.train.boxcox$status,
                      method = "qda",
                      metric = "ROC",
                      trControl = trainControl(method = "repeatedcv", repeats = 15, summaryFunction = twoCl

test.pred.prob.qda.new <- predict(model.qda.new, newdata = liver.test.boxcox ,type = "prob")[,2]
```

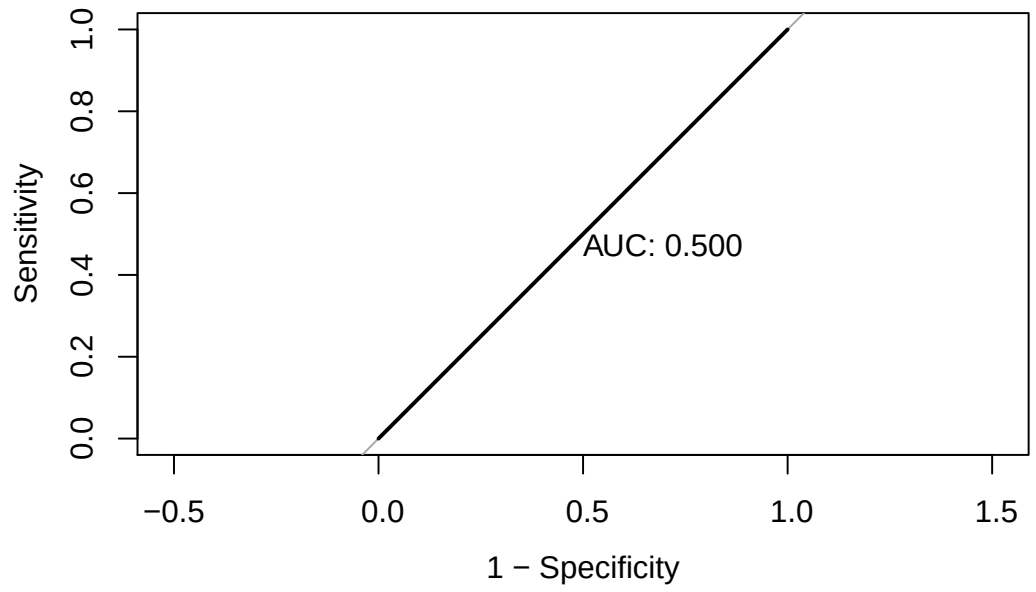
```
## Warning in predict.qda(modelFit, newdata): variable names in 'newdata' do not
## match those in 'object'
```

```
roc.qda.new <- roc(liver.test.boxcox$status, test.pred.prob.qda.new)
```

```
## Setting levels: control = No, case = Yes
```

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

```
plot(roc.qda.new, legacy.axes = TRUE, print.auc = TRUE)
```




```
#qda without boxcox tranformation
set.seed(2)
model.qda.org <-train(x = liver.train[,c(8,9,6,1,10,5)],
                  y = liver.train$status,
                  method = "qda",
                  metric = "ROC",
                  trControl = trainControl(method = "repeatedcv", repeats = 10, summaryFunction = twoCla

test.pred.prob.qda.org <- predict(model.qda.org, newdata = liver.test ,type = "prob")[,2]
```

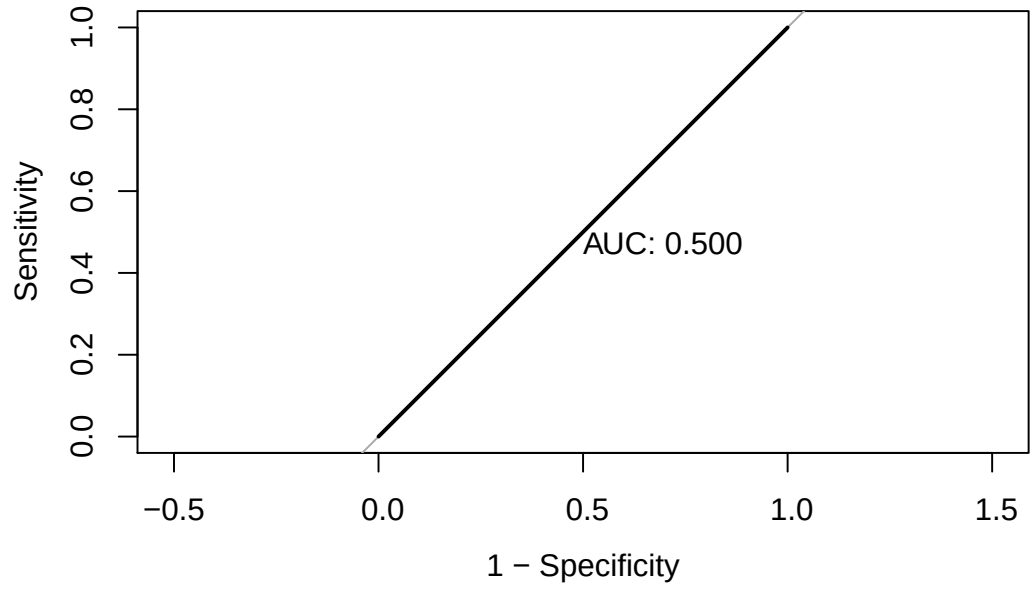
```
## Warning in predict.qda(modelFit, newdata): variable names in 'newdata' do not
## match those in 'object'
```

```
roc.qda.org <- roc(liver.test$status, test.pred.prob.qda.org)
```

```
## Setting levels: control = No, case = Yes
```

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

```
plot(roc.qda.org, legacy.axes = TRUE, print.auc = TRUE)
```



Confusion Matrix

```
#confusion matrix for logistics regression with all features
test.pred.prob <- predict(log.fit, newdata = liver.test.boxcox ,type = "response")
test.pred <- rep("No", length(test.pred.prob))
test.pred[test.pred.prob>0.5] <- "Yes"
CM.log = confusionMatrix(data = as.factor(test.pred),
  reference = liver.test.boxcox$status,
  positive = "Yes",
  prevalence = preval)
CM.log
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  13   8
##        Yes 20  74
##
##                Accuracy : 0.7565
##                  95% CI : (0.6677, 0.8317)
##     No Information Rate : 0.713
##     P-Value [Acc > NIR] : 0.17730
##
##                   Kappa : 0.3325
##
##  Mcnemar's Test P-Value : 0.03764
##
##             Sensitivity : 0.9024
```

```
##             Specificity : 0.3939
##          Pos Pred Value : 1.6776
##          Neg Pred Value : 1.7000
##              Prevalence : 2.5091
##          Detection Rate : 0.6435
##    Detection Prevalence : 0.8174
##       Balanced Accuracy : 0.6482
##
##        'Positive' Class : Yes
##
```

```
#confusion matrix for logistics regression selected features
test.pred.glm2 <- rep("No", length(test.pred.prob.glm2))
test.pred.glm2[test.pred.prob.glm2>0.5] <- "Yes"
CM.log.fea = confusionMatrix(data = as.factor(test.pred.glm2),
  reference = liver.test.boxcox$status,
  positive = "Yes",
  prevalence = preval)
CM.log.fea
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  11   6
##        Yes 22  76
##
##                Accuracy : 0.7565
##                  95% CI : (0.6677, 0.8317)
##     No Information Rate : 0.713
##     P-Value [Acc > NIR] : 0.177296
##
##                   Kappa : 0.3042
##
##  Mcnemar's Test P-Value : 0.004586
##
##             Sensitivity : 0.9268
##             Specificity : 0.3333
##          Pos Pred Value : 1.7625
##          Neg Pred Value : 1.5747
##              Prevalence : 2.5091
##          Detection Rate : 0.6609
##    Detection Prevalence : 0.8522
##       Balanced Accuracy : 0.6301
##
##        'Positive' Class : Yes
##
```

```
#confusion matrix for LDA
test.pred.lda <- rep("No", length(lda.pred))
test.pred.lda[lda.pred>0.5] <- "Yes"
CM.lda = confusionMatrix(data = as.factor(test.pred.lda),
  reference = liver.test.boxcox$status,
```

```
  positive = "Yes",
  prevalence = preval)
CM.lda
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No   2   6
##        Yes 31  76
##
##                Accuracy : 0.6783
##                  95% CI : (0.5847, 0.7623)
##     No Information Rate : 0.713
##     P-Value [Acc > NIR] : 0.8238
##
##                   Kappa : -0.0162
##
##  Mcnemar's Test P-Value : 7.961e-05
##
##             Sensitivity :  0.92683
##             Specificity :  0.06061
##          Pos Pred Value :  2.56149
##          Neg Pred Value : -0.99271
##              Prevalence :  2.50909
##          Detection Rate :  0.66087
##    Detection Prevalence :  0.93043
##       Balanced Accuracy :  0.49372
##
##        'Positive' Class : Yes
##
```

```
#confusion matrix for QDA
test.pred.qda <- rep("No", length(qda.pred))
test.pred.qda[qda.pred>0.5] <- "Yes"
CM.qda = confusionMatrix(data = as.factor(test.pred.qda),
  reference = liver.test.boxcox$status,
  positive = "Yes",
  prevalence = preval)
```

```
## Warning in confusionMatrix.default(data = as.factor(test.pred.qda), reference =
## liver.test.boxcox$status, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```
CM.qda
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No   0   0
##        Yes 33  82
```

```
##
##                Accuracy : 0.713
##                  95% CI : (0.6212, 0.7935)
##     No Information Rate : 0.713
##     P-Value [Acc > NIR] : 0.5468
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.54e-08
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 2.509
##          Neg Pred Value :   NaN
##              Prevalence : 2.509
##          Detection Rate : 0.713
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : Yes
##
```

```
#confusion matrix for knn
CM.pred.knn <- rep("No", length(knn.pred))
CM.pred.knn[knn.pred>0.5] <- "Yes"
CM.knn = confusionMatrix(data = as.factor(CM.pred.knn),
  reference = liver.test.boxcox$status,
  positive = "Yes",
  prevalence = preval)
```

```
## Warning in confusionMatrix.default(data = as.factor(CM.pred.knn), reference =
## liver.test.boxcox$status, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```
CM.knn
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  33  82
##        Yes  0   0
##
##                Accuracy : 0.287
##                  95% CI : (0.2065, 0.3788)
##     No Information Rate : 0.713
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity :  0.000
```

```
##             Specificity :  1.000
##          Pos Pred Value :    NaN
##          Neg Pred Value : -1.509
##              Prevalence :  2.509
##          Detection Rate :  0.000
##    Detection Prevalence :  0.000
##       Balanced Accuracy :  0.500
##
##        'Positive' Class : Yes
##
```

```
#confusion matrix for Naive Bayes
test.pred.nb <- rep("No", length(nb.pred))
test.pred.nb[nb.pred>0.5] <- "Yes"
CM.nb = confusionMatrix(data = as.factor(test.pred.nb),
  reference = liver.test.boxcox$status,
  positive = "Yes",
  prevalence = preval)
CM.nb
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  11  12
##        Yes 22  70
##
##                Accuracy : 0.7043
##                  95% CI : (0.6121, 0.7858)
##     No Information Rate : 0.713
##     P-Value [Acc > NIR] : 0.6264
##
##                   Kappa : 0.2056
##
##  Mcnemar's Test P-Value : 0.1227
##
##             Sensitivity : 0.8537
##             Specificity : 0.3333
##          Pos Pred Value : 1.8857
##          Neg Pred Value : 3.7029
##              Prevalence : 2.5091
##          Detection Rate : 0.6087
##    Detection Prevalence : 0.8000
##       Balanced Accuracy : 0.5935
##
##        'Positive' Class : Yes
##
```