Assignment 2

2024-02-25

```
library(ISLR)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(gmodels)
```

```
norm_model <- preProcess(Default, method = c('range'))
Default_normalized <- predict(norm_model,Default)
summary(Default_normalized)
```

```
##  default    student       balance           income
##  No :9667   No :7056   Min.   :0.0000   Min.   :0.0000
##  Yes: 333   Yes:2944   1st Qu.:0.1815   1st Qu.:0.2826
##                        Median :0.3103   Median :0.4641
##                        Mean   :0.3147   Mean   :0.4499
##                        3rd Qu.:0.4394   3rd Qu.:0.5913
##                        Max.   :1.0000   Max.   :1.0000
```

```
set.seed(123)
model <- train(default~balance+income, data = Default_normalized, method="knn")
model
```

```
## k-Nearest Neighbors
##
## 10000 samples
##     2 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 10000, 10000, 10000, 10000, 10000, 10000, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9638284  0.3474310
##   7  0.9671421  0.3717295
##   9  0.9692222  0.3918942
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```r
set.seed(123)
search_grid <- expand.grid(k=c(2,7,9,15))
model <- train(default~balance+income,data = Default_normalized,
               method="knn",tuneGrid=search_grid)
```

```r
library(class)
Default_normalized <- Default_normalized[-2]
Index_Train <- createDataPartition(Default_normalized$default,p=0.8, list = FALSE)
Train <- Default_normalized[Index_Train,]
Test <- Default_normalized[-Index_Train,]
```

```r
Train_predictors <- Train[,2:3]
Test_predictors <- Test[,2:3]
Train_labels <- Train[,1]
Test_labels <- Test[,1]
predicted_Test_labels <- knn(Train_predictors,
                             Test_predictors,
                             cl=Train_labels,
                             k=4)
head(predicted_Test_labels)
```

```
## [1] No No No No No No
## Levels: No Yes
```

```r
CrossTable(x=Test_labels,y=predicted_Test_labels,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1999
##
##
##              | predicted_Test_labels
##  Test_labels |        No |       Yes | Row Total |
## -------------|-----------|-----------|-----------|
##           No |      1921 |        12 |      1933 |
##              |     0.994 |     0.006 |     0.967 |
##              |     0.977 |     0.364 |           |
##              |     0.961 |     0.006 |           |
## -------------|-----------|-----------|-----------|
##          Yes |        45 |        21 |        66 |
##              |     0.682 |     0.318 |     0.033 |
##              |     0.023 |     0.636 |           |
##              |     0.023 |     0.011 |           |
```

```
## -------------|-----------|-----------|-----------|
## Column Total |      1966 |        33 |      1999 |
##              |     0.983 |     0.017 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```
predicted_Test_labels <- knn(Train_predictors,
                             Test_predictors,
                             cl=Train_labels,k=100,prob = TRUE)
class_prob <- attr(predicted_Test_labels,'prob')
head(class_prob)
```

```
## [1] 1.00 1.00 1.00 1.00 1.00 0.99
```

```
library(caret)
conf_matrix <- confusionMatrix(data = factor(predicted_Test_labels, levels = levels(Test_labels)), refe
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1932   54
##        Yes    1   12
##
##                Accuracy : 0.9725
##                  95% CI : (0.9643, 0.9792)
##     No Information Rate : 0.967
##     P-Value [Acc > NIR] : 0.0917
##
##                   Kappa : 0.2961
##
##  Mcnemar's Test P-Value : 2.355e-12
##
##             Sensitivity : 0.9995
##             Specificity : 0.1818
##          Pos Pred Value : 0.9728
##          Neg Pred Value : 0.9231
##              Prevalence : 0.9670
##          Detection Rate : 0.9665
##    Detection Prevalence : 0.9935
##       Balanced Accuracy : 0.5907
##
##        'Positive' Class : No
##
```

```
library('pROC')
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:gmodels':
##
##     ci

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
rocobj <- plot.roc(aSAH$outcome, aSAH$s100b, main= "Confidence intervals", percent = TRUE,ci = TRUE,pri
```

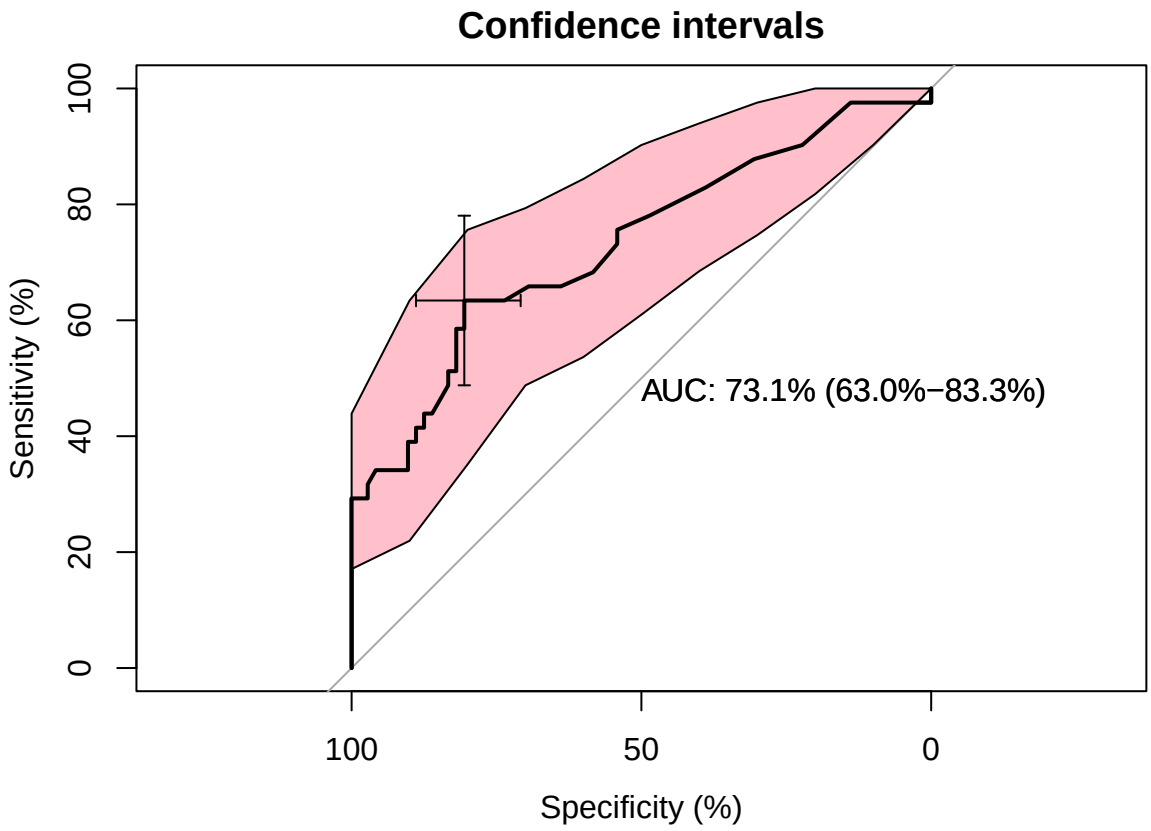
```
## Setting levels: control = Good, case = Poor

## Setting direction: controls < cases
```

```
ciobj <- ci.se(rocobj,specifities = seq(0, 100, 5))
plot(ciobj, type = "shape", col = "pink")
```

```
## Warning in plot.ci.se(ciobj, type = "shape", col = "pink"): Low definition
## shape.
```

```
plot(ci(rocobj, of = "thresholds", thresholds="best"))
```



```
plot
```

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x11e052070>
## <environment: namespace:base>
```