November 2, 2023

The results below are generated from an R script.

```
# Librerías necesarias para resolver el ejercicio
library(liver)

##
## Attaching package:  'liver'
## The following object is masked from 'package:base':
##
##     transform

library(caret)
library(caTools)
library(rpart.plot)

# Datos
data(adult)

# Resumen
summary(adult)

##       age              workclass      demogweight             education
##  Min.   :17.0   ?           : 2794   Min.   :  12285   HS-grad     :15750
##  1st Qu.:28.0   Gov         : 6536   1st Qu.: 117550   Some-college:10860
##  Median :37.0   Never-worked:   10   Median : 178215   Bachelors   : 7962
##  Mean   :38.6   Private     :33780   Mean   : 189685   Masters     : 2627
##  3rd Qu.:48.0   Self-emp    : 5457   3rd Qu.: 237713   Assoc-voc   : 2058
##  Max.   :90.0   Without-pay :   21   Max.   :1490400   11th        : 1812
##                                                       (Other)     : 7529
##  education.num         marital.status            occupation            relationship
##  Min.   : 1.00   Divorced     : 6613   Craft-repair   : 6096   Husband       :19537
##  1st Qu.: 9.00   Married      :22847   Prof-specialty : 6071   Not-in-family :12546
##  Median :10.00   Never-married:16096   Exec-managerial: 6019   Other-relative: 1506
##  Mean   :10.06   Separated    : 1526   Adm-clerical   : 5603   Own-child     : 7577
##  3rd Qu.:12.00   Widowed      : 1516   Sales          : 5470   Unmarried     : 5118
##  Max.   :16.00                         Other-service  : 4920   Wife          : 2314
##                                        (Other)        :14419
##                  race          gender       capital.gain      capital.loss
##  Amer-Indian-Eskimo:  470   Female:16156   Min.   :    0.0   Min.   :   0.00
##  Asian-Pac-Islander: 1504   Male  :32442   1st Qu.:    0.0   1st Qu.:   0.00
##  Black             : 4675                  Median :    0.0   Median :   0.00
##  Other             :  403                  Mean   :  582.4   Mean   :  87.94
##  White             :41546                  3rd Qu.:    0.0   3rd Qu.:   0.00
##                                            Max.   :41310.0   Max.   :4356.00
##
##  hours.per.week        native.country    income
```

```
##  Min.   : 1.00   United-States:43613   <=50K:37155
##  1st Qu.:40.00   Mexico       :  949   >50K :11443
##  Median :40.00   ?            :  847
##  Mean   :40.37   Philippines  :  292
##  3rd Qu.:45.00   Germany      :  206
##  Max.   :99.00   Puerto-Rico  :  184
##                  (Other)      : 2507
```

```
# Partición de los datos

# Mediante una semilla conseguimos que el ejercicio sea reproducible
set.seed(12321)

# Usamos el 20% de la base de datos como conjunto de entrenamiento y el resto como conjunto de validaci
sample = sample.split(adult$income, SplitRatio=0.2)
datos.train  = subset(adult, sample==TRUE)
datos.valid   = subset(adult, sample==FALSE)

# Entrenamos un modelo sobre la muestra de entrenamiento empleando todas las variables

traindata = datos.train[,-15]
trainclasses = datos.train[,15]
validdata = datos.valid[,-15]
validclasses = datos.valid[,15]

ctrl <- trainControl(method = "cv", number = 5)

# Entrenamos un knn

# Entrenamos un knn en cada una de las particiones
ctrl <- trainControl(method = "cv", number = 5)
traindata1 = as.data.frame(cbind(traindata$age,traindata$hours.per.week))
knn.fit1 = train(traindata1,trainclasses,method="knn",trControl=ctrl, preProcess = c("center","scale"))
knn.fit1
```

```
## k-Nearest Neighbors
##
## 9720 samples
##    2 predictor
##    2 classes: '<=50K', '>50K'
##
## Pre-processing: centered (2), scaled (2)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 7776, 7776, 7775, 7777, 7776
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.7562751  0.1631052
##   7  0.7609056  0.1704753
##   9  0.7644034  0.1784621
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
# Modelo Final
```

```r
knn.fit1$finalModel
```

```
## 9-nearest neighbor model
## Training set outcome distribution:
##
## <=50K  >50K
##  7431  2289
```

```r
# Resultados del modelo para cada una de las submuestras
knn.fit1$resample
```

```
##    Accuracy     Kappa Resample
## 1 0.7664609 0.2006926    Fold1
## 2 0.7637674 0.1657707    Fold4
## 3 0.7629820 0.1784940    Fold3
## 4 0.7613169 0.1660047    Fold2
## 5 0.7674897 0.1813486    Fold5
```

```r
# Error de clasificación en train
# sobre la partición de entrenamiento
prediction = predict(knn.fit1$finalModel, traindata1, type = 'class')
cf = confusionMatrix(prediction, as.factor(trainclasses),positive=">50K")
print(cf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  7412 2287
##      >50K     19    2
##
##                Accuracy : 0.7628
##                  95% CI : (0.7542, 0.7712)
##     No Information Rate : 0.7645
##     P-Value [Acc > NIR] : 0.6628
##
##                   Kappa : -0.0026
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0008737
##             Specificity : 0.9974431
##          Pos Pred Value : 0.0952381
##          Neg Pred Value : 0.7642025
##              Prevalence : 0.2354938
##          Detection Rate : 0.0002058
##    Detection Prevalence : 0.0021605
##       Balanced Accuracy : 0.4991584
##
##        'Positive' Class : >50K
##
```

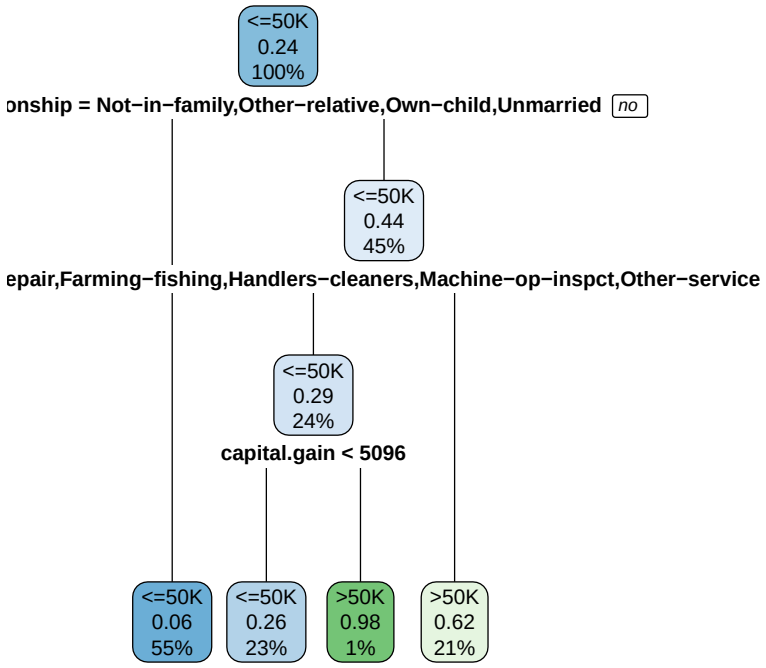
```r
# Entrenamos un árbol en cada una de las particiones
dt.fit1 = train(traindata,trainclasses,method="rpart",trControl=ctrl)
dt.fit1
```

```
## CART
##
## 9720 samples
##   14 predictor
##    2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 7775, 7776, 7777, 7776, 7776
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.03363914  0.8299390  0.4903102
##   0.04062910  0.8180055  0.4657887
##   0.10943644  0.7926955  0.2528124
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03363914.
```

```
# Modelo Final
dt.fit1$finalModel
```

```
## n= 9720
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 9720 2289 <=50K (0.76450617 0.23549383)
##    2) relationship=Not-in-family,Other-relative,Own-child,Unmarried 5315  335 <=50K (0.93697084 0.063
##    3) relationship=Husband,Wife 4405 1954 <=50K (0.55641317 0.44358683)
##      6) occupation=?,Adm-clerical,Craft-repair,Farming-fishing,Handlers-cleaners,Machine-op-inspct,Ot
##       12) capital.gain< 5095.5 2239  574 <=50K (0.74363555 0.25636445) *
##       13) capital.gain>=5095.5 97    2 >50K (0.02061856 0.97938144) *
##      7) occupation=Armed-Forces,Exec-managerial,Prof-specialty,Protective-serv,Sales,Tech-support 206
```

```
rpart.plot(dt.fit1$finalModel)
```




```
# Resultados del modelo para cada una de las submuestras
dt.fit1$resample
```

```
##    Accuracy     Kappa Resample
## 1 0.8313625 0.4635475    Fold1
## 2 0.8179012 0.4393843    Fold2
## 3 0.8353909 0.4999293    Fold5
## 4 0.8266461 0.5040981    Fold4
## 5 0.8383942 0.5445919    Fold3
```

```
# Error de clasificación en train
# sobre la partición de entrenamiento
prediction = predict(dt.fit1$finalModel, datos.train, type = 'class')
cf = confusionMatrix(prediction, as.factor(trainclasses),positive=">50K")
print(cf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  6645  909
##      >50K    786 1380
##
##                Accuracy : 0.8256
##                  95% CI : (0.8179, 0.8331)
##     No Information Rate : 0.7645
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5065
##
```

```
##  Mcnemar's Test P-Value : 0.003044
##
##             Sensitivity : 0.6029
##             Specificity : 0.8942
##          Pos Pred Value : 0.6371
##          Neg Pred Value : 0.8797
##              Prevalence : 0.2355
##          Detection Rate : 0.1420
##    Detection Prevalence : 0.2228
##       Balanced Accuracy : 0.7486
##
##        'Positive' Class : >50K
##
```

```
# sobre la partición de validación
prediction = predict(dt.fit1$finalModel, datos.valid, type = 'class')
cf = confusionMatrix(prediction, as.factor(validclasses),positive=">50K")
print(cf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K  >50K
##      <=50K 26700  3662
##      >50K   3024  5492
##
##                Accuracy : 0.828
##                  95% CI : (0.8242, 0.8318)
##     No Information Rate : 0.7645
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5105
##
##  Mcnemar's Test P-Value : 6.683e-15
##
##             Sensitivity : 0.6000
##             Specificity : 0.8983
##          Pos Pred Value : 0.6449
##          Neg Pred Value : 0.8794
##              Prevalence : 0.2355
##          Detection Rate : 0.1413
##    Detection Prevalence : 0.2190
##       Balanced Accuracy : 0.7491
##
##        'Positive' Class : >50K
##
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.6 LTS
##
```

```
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
## LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3;  LAPACK version 3.9.0
##
## locale:
##  [1] LC_CTYPE=es_ES.UTF-8       LC_NUMERIC=C               LC_TIME=es_ES.UTF-8
##  [4] LC_COLLATE=es_ES.UTF-8     LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=es_ES.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Madrid
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] liver_1.15       ggfortify_0.4.16 factoextra_1.0.7 mlbench_2.1-3.1  readxl_1.4.3
##  [6] caret_6.0-94     lattice_0.21-9   ggplot2_3.4.3    rpart.plot_3.1.1 rpart_4.1.19
## [11] caTools_1.18.2   dplyr_1.1.3      ISLR2_1.3-2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0     timeDate_4022.108    farver_2.1.1         bitops_1.0-7
##  [5] fastmap_1.1.1        pROC_1.18.4          digest_0.6.33        timechange_0.2.0
##  [9] lifecycle_1.0.3      survival_3.5-7       magrittr_2.0.3       compiler_4.3.1
## [13] rlang_1.1.1          tools_4.3.1          utf8_1.2.3           yaml_2.3.7
## [17] data.table_1.14.8    knitr_1.44           labeling_0.4.3       plyr_1.8.9
## [21] withr_2.5.1          purrr_1.0.2          nnet_7.3-19          grid_4.3.1
## [25] stats4_4.3.1         fansi_1.0.5          e1071_1.7-13         colorspace_2.1-0
## [29] future_1.33.0        globals_0.16.2       scales_1.2.1         iterators_1.0.14
## [33] MASS_7.3-60          tinytex_0.47         cli_3.6.1            rmarkdown_2.25
## [37] generics_0.1.3       rstudioapi_0.15.0    future.apply_1.11.0  reshape2_1.4.4
## [41] tzdb_0.4.0           proxy_0.4-27         stringr_1.5.0        splines_4.3.1
## [45] parallel_4.3.1       cellranger_1.1.0     vctrs_0.6.3          hardhat_1.3.0
## [49] Matrix_1.6-1.1       hms_1.1.3            ggrepel_0.9.3        listenv_0.9.0
## [53] foreach_1.5.2        tidyr_1.3.0          gower_1.0.1          recipes_1.0.8
## [57] glue_1.6.2           parallelly_1.36.0    codetools_0.2-19     lubridate_1.9.3
## [61] stringi_1.7.12       gtable_0.3.4         munsell_0.5.0        tibble_3.2.1
## [65] pillar_1.9.0         htmltools_0.5.6.1    ipred_0.9-14         lava_1.7.2.1
## [69] R6_2.5.1             evaluate_0.22        readr_2.1.4          highr_0.10
## [73] class_7.3-22         Rcpp_1.0.11          gridExtra_2.3        nlme_3.1-163
## [77] prodlim_2023.08.28   xfun_0.40            pkgconfig_2.0.3      ModelMetrics_1.2.2.2
```

```
Sys.time()
```

```
## [1] "2023-11-02 17:28:54 CET"
```