# ELVS & LSAC SuperLearner results for new predictions

Prepare the data

```r
# Subset to just the language outcome and predictors
all_top<-ELVS_LSAC[c("lang11yr15sd","dolly","circle","accident","kangaroo","forget")]
# Remove missing data
all_top<-na.omit(all_top)
# Count number of rows with complete data
nrow(all_top)
```

```
## [1] 1957
```

```r
# Rename the outcome so it matches the varibale in the SuperLearner object
colnames(all_top)[colnames(all_top) == c("lang11yr15sd")] <- c("lang_11yr")
# Create a vector of the outcome so it can be used below
lang_11yr<-all_top$lang_11yr
```

Calculate AUC of the SuperLearner object

```r
# Bring in the SuperLearner object
sl <- readRDS("sl_elvslsac_newpredictions.rds")
summary(sl)
```

```
##                    Length Class  Mode
## call                   5  -none- call
## libraryNames          10  -none- character
## SL.library             2  -none- list
## SL.predict          1957  -none- numeric
## coef                  10  -none- numeric
## library.predict    19570  -none- numeric
## Z                  19570  -none- numeric
## cvRisk                10  -none- numeric
## family                12  family list
## fitLibrary            10  -none- list
## cvFitLibrary           0  -none- NULL
## varNames               5  -none- character
## validRows             10  -none- list
## method                 3  -none- list
## whichScreen            5  -none- logical
## control                3  -none- list
## cvControl              4  -none- list
## errorsInCVLibrary     10  -none- logical
## errorsInLibrary       10  -none- logical
## metaOptimizer          8  nnls   list
## env                    5  -none- environment
## times                  3  -none- list
```

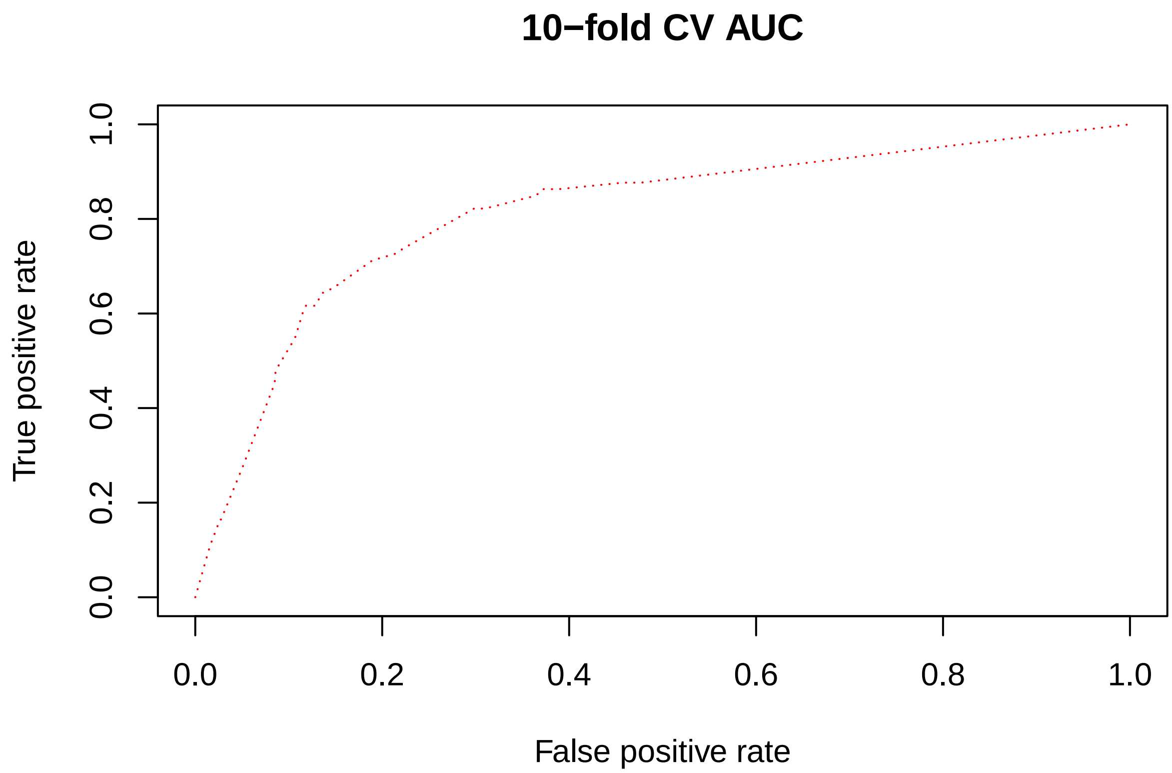
```r
# Look at predictions
predictions <- sl$SL.predict
summary(predictions)
```

```
##        V1
##  Min.   :0.01174
##  1st Qu.:0.01174
##  Median :0.01174
##  Mean   :0.03724
##  3rd Qu.:0.03920
##  Max.   :0.20022
```

```r
# Calculate AUC and 95% confidence intervals
sl_auc<-cvAUC(predictions,lang_11yr)
sl_auc_cis<-ci.cvAUC(predictions,lang_11yr)
sl_auc_cis
```

```
## $cvAUC
## [1] 0.8117674
##
## $se
## [1] 0.03630007
##
## $ci
## [1] 0.7406206 0.8829143
##
## $confidence
## [1] 0.95
```

```r
plot(sl_auc$perf, col="red", lty=3, main="10-fold CV AUC")
```




Select cut-offs for different scenarios

## Maximise Sensitivity

A cut-off of 0.015 maximises sensitivity (at 88%, but with only 54% specificity)

```
pred_vals <- ifelse(predictions < 0.015, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1022    9
##          1  862   64
##
##                Accuracy : 0.5549
##                  95% CI : (0.5326, 0.5771)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0634
##
```

```
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.87671
##             Specificity : 0.54246
##          Pos Pred Value : 0.06911
##          Neg Pred Value : 0.99127
##              Prevalence : 0.03730
##          Detection Rate : 0.03270
##    Detection Prevalence : 0.47317
##       Balanced Accuracy : 0.70959
##
##        'Positive' Class : 1
##
```

```
### To get the 95% CIs
### Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(64,862,9,1022), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +           64          862        926
## Test -            9         1022       1031
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## ---------------------------------------------------------------
## Apparent prevalence *                  0.47 (0.45, 0.50)
## True prevalence *                      0.04 (0.03, 0.05)
## Sensitivity *                          0.88 (0.78, 0.94)
## Specificity *                          0.54 (0.52, 0.57)
## Positive predictive value *            0.07 (0.05, 0.09)
## Negative predictive value *            0.99 (0.98, 1.00)
## Positive likelihood ratio              1.92 (1.74, 2.12)
## Negative likelihood ratio              0.23 (0.12, 0.42)
## False T+ proportion for true D- *      0.46 (0.43, 0.48)
## False T- proportion for true D+ *      0.12 (0.06, 0.22)
## False T+ proportion for T+ *           0.93 (0.91, 0.95)
## False T- proportion for T- *           0.01 (0.00, 0.02)
## Correctly classified proportion *      0.55 (0.53, 0.58)
## ---------------------------------------------------------------
## * Exact CIs
```

# >80% Sensitivity

A cut-off of 0.035 achieves >80% sensitivity (but with only 70% specificity)

```
pred_vals <- ifelse(predictions < 0.035, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

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

```
##
##           Reference
## Prediction    0    1
##          0 1313   13
##          1  571   60
##
##                Accuracy : 0.7016
##                  95% CI : (0.6808, 0.7218)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.111
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.82192
##             Specificity : 0.69692
##          Pos Pred Value : 0.09509
##          Neg Pred Value : 0.99020
##              Prevalence : 0.03730
##          Detection Rate : 0.03066
##    Detection Prevalence : 0.32243
##       Balanced Accuracy : 0.75942
##
##        'Positive' Class : 1
##
```

```
### To get the 95% CIs
### Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(60,571,13,1313), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +           60          571        631
## Test -           13         1313       1326
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## ---------------------------------------------------------------
## Apparent prevalence *                  0.32 (0.30, 0.34)
## True prevalence *                      0.04 (0.03, 0.05)
## Sensitivity *                          0.82 (0.71, 0.90)
## Specificity *                          0.70 (0.68, 0.72)
## Positive predictive value *            0.10 (0.07, 0.12)
## Negative predictive value *            0.99 (0.98, 0.99)
## Positive likelihood ratio              2.71 (2.39, 3.08)
## Negative likelihood ratio              0.26 (0.16, 0.42)
## False T+ proportion for true D- *      0.30 (0.28, 0.32)
## False T- proportion for true D+ *      0.18 (0.10, 0.29)
## False T+ proportion for T+ *           0.90 (0.88, 0.93)
## False T- proportion for T- *           0.01 (0.01, 0.02)
## Correctly classified proportion *      0.70 (0.68, 0.72)
## ---------------------------------------------------------------
```

```
## * Exact CIs
```

## Balance sensitivity and specificity

A cut-off of 0.0395 most balances sensitivity and specificity

```r
pred_vals <- ifelse(predictions < 0.0395, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1461   19
##          1  423   54
##
##                Accuracy : 0.7741
##                  95% CI : (0.755, 0.7925)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1408
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.73973
##             Specificity : 0.77548
##          Pos Pred Value : 0.11321
##          Neg Pred Value : 0.98716
##              Prevalence : 0.03730
##          Detection Rate : 0.02759
##    Detection Prevalence : 0.24374
##       Balanced Accuracy : 0.75760
##
##        'Positive' Class : 1
##
```

```r
# To get the 95% CIs
# Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(54,423,19,1461), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +           54          423        477
## Test -           19         1461       1480
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## --------------------------------------------------------------
```

```
## Apparent prevalence *                 0.24 (0.22, 0.26)
## True prevalence *                     0.04 (0.03, 0.05)
## Sensitivity *                         0.74 (0.62, 0.84)
## Specificity *                         0.78 (0.76, 0.79)
## Positive predictive value *           0.11 (0.09, 0.15)
## Negative predictive value *           0.99 (0.98, 0.99)
## Positive likelihood ratio             3.29 (2.81, 3.87)
## Negative likelihood ratio             0.34 (0.23, 0.49)
## False T+ proportion for true D- *     0.22 (0.21, 0.24)
## False T- proportion for true D+ *     0.26 (0.16, 0.38)
## False T+ proportion for T+ *          0.89 (0.85, 0.91)
## False T- proportion for T- *          0.01 (0.01, 0.02)
## Correctly classified proportion *     0.77 (0.75, 0.79)
## --------------------------------------------------------------
## * Exact CIs
```

## >80% Specificity

A cut-off of 0.045 achieves >80% specificity (and 71% sensitivity)

```r
pred_vals <- ifelse(predictions < 0.045, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1527   21
##          1  357   52
##
##                Accuracy : 0.8068
##                  95% CI : (0.7886, 0.8241)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1628
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.71233
##             Specificity : 0.81051
##          Pos Pred Value : 0.12714
##          Neg Pred Value : 0.98643
##              Prevalence : 0.03730
##          Detection Rate : 0.02657
##    Detection Prevalence : 0.20899
##       Balanced Accuracy : 0.76142
##
##        'Positive' Class : 1
##
```

```r
### To get the 95% CIs
### Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(52,357,21,1527), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +           52          357        409
## Test -           21         1527       1548
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## --------------------------------------------------------------
## Apparent prevalence *                  0.21 (0.19, 0.23)
## True prevalence *                      0.04 (0.03, 0.05)
## Sensitivity *                          0.71 (0.59, 0.81)
## Specificity *                          0.81 (0.79, 0.83)
## Positive predictive value *            0.13 (0.10, 0.16)
## Negative predictive value *            0.99 (0.98, 0.99)
## Positive likelihood ratio              3.76 (3.16, 4.47)
## Negative likelihood ratio              0.35 (0.25, 0.51)
## False T+ proportion for true D- *      0.19 (0.17, 0.21)
## False T- proportion for true D+ *      0.29 (0.19, 0.41)
## False T+ proportion for T+ *           0.87 (0.84, 0.90)
## False T- proportion for T- *           0.01 (0.01, 0.02)
## Correctly classified proportion *      0.81 (0.79, 0.82)
## --------------------------------------------------------------
## * Exact CIs
```

## >90% Specificity

A cut-off of 0.11 achieves >90% specificity (but only 48% sensitivity)

```r
pred_vals <- ifelse(predictions < 0.11, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1722   38
##          1  162   35
##
##                Accuracy : 0.8978
##                  95% CI : (0.8835, 0.9109)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2166
##
```

```
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.47945
##             Specificity : 0.91401
##          Pos Pred Value : 0.17766
##          Neg Pred Value : 0.97841
##              Prevalence : 0.03730
##          Detection Rate : 0.01788
##    Detection Prevalence : 0.10066
##       Balanced Accuracy : 0.69673
##
##        'Positive' Class : 1
##
```

```
### To get the 95% CIs
### Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(35,162,38,1722), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +           35          162        197
## Test -           38         1722       1760
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## ---------------------------------------------------------------
## Apparent prevalence *                  0.10 (0.09, 0.11)
## True prevalence *                      0.04 (0.03, 0.05)
## Sensitivity *                          0.48 (0.36, 0.60)
## Specificity *                          0.91 (0.90, 0.93)
## Positive predictive value *            0.18 (0.13, 0.24)
## Negative predictive value *            0.98 (0.97, 0.98)
## Positive likelihood ratio              5.58 (4.21, 7.38)
## Negative likelihood ratio              0.57 (0.46, 0.71)
## False T+ proportion for true D- *      0.09 (0.07, 0.10)
## False T- proportion for true D+ *      0.52 (0.40, 0.64)
## False T+ proportion for T+ *           0.82 (0.76, 0.87)
## False T- proportion for T- *           0.02 (0.02, 0.03)
## Correctly classified proportion *      0.90 (0.88, 0.91)
## ---------------------------------------------------------------
## * Exact CIs
```

# >95% Specificity

A cut-off of 0.14 achieves >95% specificity (but only 16% sensitivity)

```
pred_vals <- ifelse(predictions < 0.14, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

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

```
##
##           Reference
## Prediction    0    1
##          0 1832   61
##          1   52   12
##
##                Accuracy : 0.9423
##                  95% CI : (0.931, 0.9522)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.1454
##
##  Mcnemar's Test P-Value : 0.4517
##
##             Sensitivity : 0.164384
##             Specificity : 0.972399
##          Pos Pred Value : 0.187500
##          Neg Pred Value : 0.967776
##              Prevalence : 0.037302
##          Detection Rate : 0.006132
##    Detection Prevalence : 0.032703
##       Balanced Accuracy : 0.568391
##
##        'Positive' Class : 1
##
```

```r
### To get the 95% CIs
### Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(12,52,61,1832), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +           12           52         64
## Test -           61         1832       1893
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## ---------------------------------------------------------------
## Apparent prevalence *                  0.03 (0.03, 0.04)
## True prevalence *                      0.04 (0.03, 0.05)
## Sensitivity *                          0.16 (0.09, 0.27)
## Specificity *                          0.97 (0.96, 0.98)
## Positive predictive value *            0.19 (0.10, 0.30)
## Negative predictive value *            0.97 (0.96, 0.98)
## Positive likelihood ratio              5.96 (3.33, 10.66)
## Negative likelihood ratio              0.86 (0.78, 0.95)
## False T+ proportion for true D- *      0.03 (0.02, 0.04)
## False T- proportion for true D+ *      0.84 (0.73, 0.91)
## False T+ proportion for T+ *           0.81 (0.70, 0.90)
## False T- proportion for T- *           0.03 (0.02, 0.04)
## Correctly classified proportion *      0.94 (0.93, 0.95)
## ---------------------------------------------------------------
```

```
## * Exact CIs
```

## Maximise Positive Predictive Value

A cut-off of 0.2 maximises Positive Predictive Value

```
pred_vals <- ifelse(predictions < 0.2, 0, 1)
pred_vals <- factor(pred_vals)
confusionMatrix(pred_vals, lang_11yr,positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1854   65
##          1   30    8
##
##                Accuracy : 0.9515
##                  95% CI : (0.941, 0.9606)
##     No Information Rate : 0.9627
##     P-Value [Acc > NIR] : 0.9951122
##
##                   Kappa : 0.1217
##
##  Mcnemar's Test P-Value : 0.0004861
##
##             Sensitivity : 0.109589
##             Specificity : 0.984076
##          Pos Pred Value : 0.210526
##          Neg Pred Value : 0.966128
##              Prevalence : 0.037302
##          Detection Rate : 0.004088
##    Detection Prevalence : 0.019417
##       Balanced Accuracy : 0.546833
##
##        'Positive' Class : 1
##
```

```
# To get the 95% CIs
# Note: using cross tab numbers for matrix from above confusionMatrix
data <- as.table(matrix(c(8,30,65,1854), nrow = 2, byrow = TRUE))
rval <- epi.tests(data, conf.level = 0.95)
print(rval)
```

```
##           Outcome +    Outcome -      Total
## Test +            8           30         38
## Test -           65         1854       1919
## Total            73         1884       1957
##
## Point estimates and 95% CIs:
## --------------------------------------------------------------
```

```
## Apparent prevalence *                 0.02 (0.01, 0.03)
## True prevalence *                     0.04 (0.03, 0.05)
## Sensitivity *                         0.11 (0.05, 0.20)
## Specificity *                         0.98 (0.98, 0.99)
## Positive predictive value *           0.21 (0.10, 0.37)
## Negative predictive value *           0.97 (0.96, 0.97)
## Positive likelihood ratio             6.88 (3.27, 14.48)
## Negative likelihood ratio             0.90 (0.83, 0.98)
## False T+ proportion for true D- *     0.02 (0.01, 0.02)
## False T- proportion for true D+ *     0.89 (0.80, 0.95)
## False T+ proportion for T+ *          0.79 (0.63, 0.90)
## False T- proportion for T- *          0.03 (0.03, 0.04)
## Correctly classified proportion *     0.95 (0.94, 0.96)
## --------------------------------------------------------------
## * Exact CIs
```

# Session info

```r
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=English_Australia.utf8
## [3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Australia.utf8
##
## time zone: Australia/Sydney
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] epiR_2.0.70    survival_3.5-8 caret_6.0-94   lattice_0.22-5 ggplot2_3.5.0
## [6] cvAUC_1.1.4
##
## loaded via a namespace (and not attached):
##   [1] libcoin_1.0-10          rstudioapi_0.16.0       jsonlite_1.8.8
##   [4] magrittr_2.0.3          TH.data_1.1-2           modeltools_0.2-23
##   [7] rmarkdown_2.28          ragg_1.2.7              vctrs_0.6.5
##  [10] ROCR_1.0-11             askpass_1.2.0           htmltools_0.5.7
##  [13] plotrix_3.8-4           curl_5.2.3              xgboost_1.7.8.1
##  [16] Formula_1.2-5           pROC_1.18.5             parallelly_1.37.1
##  [19] KernSmooth_2.23-22      plyr_1.8.9              sandwich_3.1-1
##  [22] zoo_1.8-12              lubridate_1.9.3         uuid_1.2-0
```

```
##  [25] gam_1.22-5            mime_0.12               lifecycle_1.0.4
##  [28] iterators_1.0.14      pkgconfig_2.0.3         Matrix_1.6-5
##  [31] R6_2.5.1              fastmap_1.1.1           plotmo_3.6.4
##  [34] future_1.33.1         shiny_1.8.0             digest_0.6.34
##  [37] colorspace_2.1-0      textshaping_0.3.7       fansi_1.0.6
##  [40] timechange_0.3.0      nnls_1.5                compiler_4.3.2
##  [43] proxy_0.4-27          fontquiver_0.2.1        withr_3.0.0
##  [46] pander_0.6.5          DBI_1.2.2               SuperLearner_2.0-29
##  [49] highr_0.10            BiasedUrn_2.0.11        MASS_7.3-60.0.1
##  [52] lava_1.8.0            openssl_2.1.1           classInt_0.4-10
##  [55] gfonts_0.2.0          ModelMetrics_1.2.2.2    tools_4.3.2
##  [58] units_0.8-5           zip_2.3.1               httpuv_1.6.14
##  [61] future.apply_1.11.1   nnet_7.3-19             glue_1.7.0
##  [64] nlme_3.1-164          promises_1.2.1          grid_4.3.2
##  [67] sf_1.0-15             reshape2_1.4.4          generics_0.1.3
##  [70] recipes_1.0.10        gtable_0.3.4            class_7.3-22
##  [73] data.table_1.16.0     xml2_1.3.6              coin_1.4-3
##  [76] utf8_1.2.4            foreach_1.5.2           pillar_1.9.0
##  [79] stringr_1.5.1         later_1.3.2             splines_4.3.2
##  [82] dplyr_1.1.4           tidyselect_1.2.1        fontLiberation_0.1.0
##  [85] knitr_1.45            fontBitstreamVera_0.1.1 crul_1.4.0
##  [88] stats4_4.3.2          xfun_0.42               hardhat_1.3.1
##  [91] timeDate_4032.109     matrixStats_1.4.1       stringi_1.8.3
##  [94] yaml_2.3.8            evaluate_0.23           codetools_0.2-19
##  [97] httpcode_0.3.0        officer_0.6.5           gdtools_0.3.7
## [100] tibble_3.2.1          cli_3.6.2               rpart_4.1.23
## [103] xtable_1.8-4          systemfonts_1.0.6       munsell_0.5.0
## [106] Rcpp_1.0.12           globals_0.16.3          parallel_4.3.2
## [109] ellipsis_0.3.2        gower_1.0.1             strucchange_1.5-4
## [112] party_1.3-17          listenv_0.9.1           mvtnorm_1.2-5
## [115] ipred_0.9-14          scales_1.3.0            prodlim_2023.08.28
## [118] e1071_1.7-14          earth_5.3.3             purrr_1.0.2
## [121] crayon_1.5.2          flextable_0.9.5         rlang_1.1.3
## [124] multcomp_1.4-26
```

```r
citation("cvAUC")
```

```
## To cite package 'cvAUC' in publications use:
##
##   LeDell E, Petersen M, van der Laan M (2022). _cvAUC: Cross-Validated
##   Area Under the ROC Curve Confidence Intervals_. R package version
##   1.1.4, <https://CRAN.R-project.org/package=cvAUC>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {cvAUC: Cross-Validated Area Under the ROC Curve Confidence Intervals},
##     author = {Erin LeDell and Maya Petersen and Mark {van der Laan}},
##     year = {2022},
##     note = {R package version 1.1.4},
##     url = {https://CRAN.R-project.org/package=cvAUC},
##   }
##
## ATTENTION: This citation information has been auto-generated from the
```

```
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

```r
citation("caret")
```

```
## To cite caret in publications use:
##
##   Kuhn, M. (2008). Building Predictive Models in R Using the caret
##   Package. Journal of Statistical Software, 28(5), 1-26.
##   https://doi.org/10.18637/jss.v028.i05
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Building Predictive Models in R Using the caret Package},
##     volume = {28},
##     url = {https://www.jstatsoft.org/index.php/jss/article/view/v028i05},
##     doi = {10.18637/jss.v028.i05},
##     number = {5},
##     journal = {Journal of Statistical Software},
##     author = {{Kuhn} and {Max}},
##     year = {2008},
##     pages = {1-26},
##   }
```

```r
citation("epiR")
```

```
## To cite package 'epiR' in publications use:
##
##   Stevenson M, Sergeant E, Firestone S (2024). _epiR: Tools for the
##   Analysis of Epidemiological Data_. R package version 2.0.70,
##   <https://CRAN.R-project.org/package=epiR>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {epiR: Tools for the Analysis of Epidemiological Data},
##     author = {Mark Stevenson and Evan Sergeant and Simon Firestone},
##     year = {2024},
##     note = {R package version 2.0.70},
##     url = {https://CRAN.R-project.org/package=epiR},
##   }
```