MRSA Klassifizierung in Schweinemastherden mit *evtree*

Dr. Robert Opitz

2021-08-27

Der MRSA-Datensatz in Schweinemastherden wurde von Fromm *et al.* in PVM, **117**, 2014 veröffentlicht. Es handelt sich dabei um eine Metastudie. Diese wurde mittels logistischer Regression und generalized estiamtion equations ausgewertet. Es wurde eine Variablenselektion durch geführt: (i) durch univariates filtern (mit logistischer Regression) wurden alle Variablen entfernt, die keinen direkten Einfluss auf die Zielvariable haben, (ii) von paarweise korrelierten Variablen wurde eine der korrelierten Variablen entfernt. Es wurde versucht auch Interaktionsterme zubreücksichtigen. Es wurde nur einer gefunden, auch aufgrund der Anzahl der Datenpunkte Datensatzes. Dieser wurde im finalen Modell nicht weiter berücksichtigt.

# Notwendige Pakte und eigene Funktionen

Hier verwendete Pakete.

library(magrittr)  
library(caret)  
library(evtree)  
library(pROC)

# Datenvorbereitung

## Einlesen und Vorbereiten der Daten.

Einlesen und Bereinigung der Daten. Es wird sichergestellt, dass alle Daten einen R-konformen Namen haben und als Faktoren vorliegen. die Variablennamen erhalten die Namen aus dem Paper. Einige der Kategorien der Variablen werden besser lesbar gestaltet.

MRSA\_schweineherden <- read.csv(file = file.choose(),  
 header = TRUE)  
  
# all categorigal variables needs to be checked if there name is valid in R  
# E. g.: TRUE and FALSE are not; TRUE. and FALSE. are  
# All categorical Variables need to be (at least) described as factors  
for (i in seq( ncol(MRSA\_schweineherden) )) {  
 MRSA\_schweineherden[[i]] <- make.names(MRSA\_schweineherden[[i]])  
 MRSA\_schweineherden[[i]] <- factor(MRSA\_schweineherden[[i]])  
}  
  
# the old names in the data set are renamed, so they match the names in the   
# paper  
new\_col\_names <- list(HerdMRSA = 'HERD\_MRSA',  
 HerdTypeNum = 'HERD\_TYPE',  
 FeedingPlacesMinGrouped = 'HERD\_SIZE',  
 PurchaseBin = 'PURCHASE',  
 AntibioticsFrom10W = 'AM\_DRUG',  
 AllInAllOut = 'ALL\_IN\_or\_OUT',  
 Cleanup = 'CLEAN\_UP',  
 Disinfection = 'DISINFECTION',  
 SlattedFloor = 'SLATTED',  
 OrganicFarm = 'ORGANIC',  
 OtherLivestockAtFarmBin = 'OTHER\_LIVESTOCK',  
 CompanionAnimalsBin = 'COMPANION',  
 IndoorHousing = 'INDOOR')  
  
# rename the column names  
column\_names <- colnames(MRSA\_schweineherden)  
for (col\_name in column\_names)  
 colnames(MRSA\_schweineherden)[column\_names == col\_name] <- new\_col\_names[[col\_name]]  
  
# replace old variable types for variable HERD\_SIZE  
levels(MRSA\_schweineherden$HERD\_SIZE)

## [1] "X..5000.pigs" "X0.499.pigs" "X1000.4999.pigs" "X500.999.pigs"

levels(MRSA\_schweineherden$HERD\_SIZE) <- c('huge', 'small', 'large', 'medium')  
  
# reorder factors  
MRSA\_schweineherden$HERD\_SIZE <- factor(MRSA\_schweineherden$HERD\_SIZE,   
 levels = c('small', 'medium',   
 'large', 'huge'),  
 ordered = TRUE)  
levels(MRSA\_schweineherden$HERD\_SIZE)

## [1] "small" "medium" "large" "huge"

levels(MRSA\_schweineherden$HERD\_TYPE)

## [1] "farrow.to.finisher" "grower.to.finisher" "weaner.to.finisher"

levels(MRSA\_schweineherden$HERD\_TYPE) <- c("farrow", "grower", "weaner")  
levels(MRSA\_schweineherden$HERD\_TYPE)

## [1] "farrow" "grower" "weaner"

# replace old variable types for variable SLATTED\_AT\_LEAST\_PARTIALLY  
levels(MRSA\_schweineherden$SLATTED)

## [1] "no..not.slatted." "yes..at.least.partially.slatted."

levels(MRSA\_schweineherden$SLATTED) <- c('no', 'yes')  
levels(MRSA\_schweineherden$SLATTED)

## [1] "no" "yes"

## Vorstellung des Datensatzes (EDA)

head(MRSA\_schweineherden)

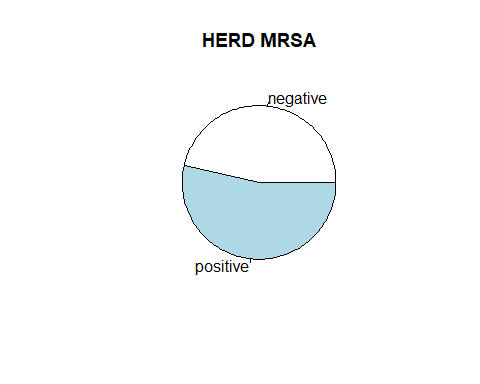
## HERD\_MRSA HERD\_TYPE HERD\_SIZE AM\_DRUG PURCHASE INDOOR OTHER\_LIVESTOCK  
## 1 positive grower large yes yes yes no  
## 2 positive grower large yes yes yes yes  
## 3 negative grower medium no yes yes no  
## 4 positive farrow small no no yes no  
## 5 negative farrow medium no no yes no  
## 6 positive farrow medium no no yes no  
## COMPANION ALL\_IN\_or\_OUT CLEAN\_UP DISINFECTION SLATTED ORGANIC  
## 1 no yes yes yes yes no  
## 2 no yes yes yes yes no  
## 3 no no yes yes yes no  
## 4 yes no no no yes no  
## 5 yes no yes yes yes no  
## 6 yes yes yes yes yes no

Bestimmung der Anzahl der Datenpunkte, Anzahl Variablen und Verteilung der Response.

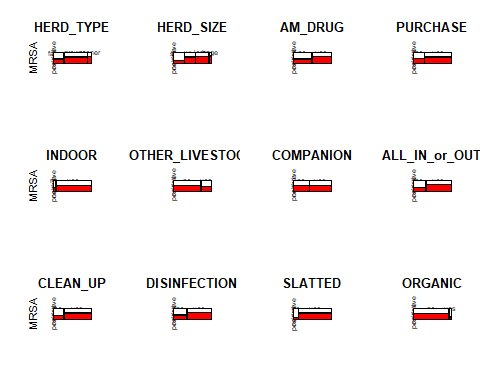
MRSA\_schweineherden$HERD\_MRSA %>% length -> number\_of\_datapoints  
MRSA\_schweineherden[1,-1] %>% length -> number\_of\_variables  
MRSA\_schweineherden$HERD\_MRSA %>% summary

## negative positive   
## 186 214

MRSA\_schweineherden$HERD\_MRSA %>% summary %>% pie(.,main = 'HERD MRSA')



Datenpunkte=400, Variablen=12.



Sind einige der Kategorien der Variablen zu dünn besetzt?

nearZeroVar(MRSA\_schweineherden, saveMetrics = TRUE)

## freqRatio percentUnique zeroVar nzv  
## HERD\_MRSA 1.150538 0.50 FALSE FALSE  
## HERD\_TYPE 2.252252 0.75 FALSE FALSE  
## HERD\_SIZE 1.247863 1.00 FALSE FALSE  
## AM\_DRUG 1.105263 0.50 FALSE FALSE  
## PURCHASE 2.603604 0.50 FALSE FALSE  
## INDOOR 20.052632 0.50 FALSE TRUE  
## OTHER\_LIVESTOCK 2.738318 0.50 FALSE FALSE  
## COMPANION 1.409639 0.50 FALSE FALSE  
## ALL\_IN\_or\_OUT 2.149606 0.50 FALSE FALSE  
## CLEAN\_UP 2.846154 0.50 FALSE FALSE  
## DISINFECTION 1.941176 0.50 FALSE FALSE  
## SLATTED 7.695652 0.50 FALSE FALSE  
## ORGANIC 16.391304 0.50 FALSE FALSE

Die Variable INDOOR besteht fast ausschließlich aus Herden, die indoor leben, fast keine die outdoor sind. Dadurch haben wir eine near-zero-variance (nzv) für diese Variable. Wir entfernen diese Variable daher.

MRSA\_schweineherden\_reduced <- MRSA\_schweineherden  
MRSA\_schweineherden\_reduced$INDOOR <- NULL

# Erstes Model

start\_time <- Sys.time()  
first\_model <- evtree(HERD\_MRSA ~ .,  
 data = MRSA\_schweineherden\_reduced,  
 control = evtree.control(niterations = 100000L,  
 ntrees = 5000L,  
 seed = -1L,  
 minbucket = 10L,  
 minsplit = 20L))  
print(Sys.time() - start\_time)

## Time difference of 3.46288 mins

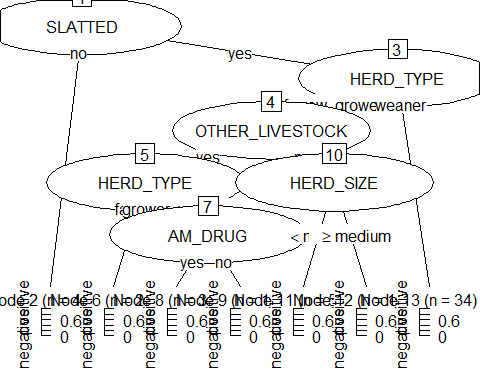
print(first\_model)

##   
## Model formula:  
## HERD\_MRSA ~ HERD\_TYPE + HERD\_SIZE + AM\_DRUG + PURCHASE + OTHER\_LIVESTOCK +   
## COMPANION + ALL\_IN\_or\_OUT + CLEAN\_UP + DISINFECTION + SLATTED +   
## ORGANIC  
##   
## Fitted party:  
## [1] root  
## | [2] SLATTED in no: negative (n = 46, err = 19.6%)  
## | [3] SLATTED in yes  
## | | [4] HERD\_TYPE in farrow, grower  
## | | | [5] OTHER\_LIVESTOCK in yes  
## | | | | [6] HERD\_TYPE in farrow: negative (n = 20, err = 15.0%)  
## | | | | [7] HERD\_TYPE in grower, weaner  
## | | | | | [8] AM\_DRUG in yes: positive (n = 39, err = 33.3%)  
## | | | | | [9] AM\_DRUG in no: negative (n = 18, err = 27.8%)  
## | | | [10] OTHER\_LIVESTOCK in no  
## | | | | [11] HERD\_SIZE < medium: negative (n = 50, err = 30.0%)  
## | | | | [12] HERD\_SIZE >= medium: positive (n = 193, err = 31.1%)  
## | | [13] HERD\_TYPE in weaner: positive (n = 34, err = 32.4%)  
##   
## Number of inner nodes: 6  
## Number of terminal nodes: 7

confusionMatrix(data = predict(first\_model),   
 reference = MRSA\_schweineherden$HERD\_MRSA,  
 positive = "positive")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction negative positive  
## negative 102 32  
## positive 84 182  
##   
## Accuracy : 0.71   
## 95% CI : (0.6628, 0.754)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 6.232e-13   
##   
## Kappa : 0.4063   
##   
## Mcnemar's Test P-Value : 2.188e-06   
##   
## Sensitivity : 0.8505   
## Specificity : 0.5484   
## Pos Pred Value : 0.6842   
## Neg Pred Value : 0.7612   
## Prevalence : 0.5350   
## Detection Rate : 0.4550   
## Detection Prevalence : 0.6650   
## Balanced Accuracy : 0.6994   
##   
## 'Positive' Class : positive   
##

plot(first\_model)



# Kreuzvalidierung

Die Kreuzvalideriung ist sehr rechen- und damit zeitaufwendig. Daher parallelisieren wir mit dem Paekt *parallel*. Im ersten chunk sind die Vorbereitungen zur Paralleliserung.

library(parallel)  
number\_of\_threads <- detectCores() # use all cores available  
# make a cluster of 6 cores for parSapply  
cl <- makeCluster(number\_of\_threads)  
clusterEvalQ(cl = cl, library(evtree)) # load library into cluster

## [[1]]  
## [1] "evtree" "partykit" "mvtnorm" "libcoin" "grid" "stats"   
## [7] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[2]]  
## [1] "evtree" "partykit" "mvtnorm" "libcoin" "grid" "stats"   
## [7] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[3]]  
## [1] "evtree" "partykit" "mvtnorm" "libcoin" "grid" "stats"   
## [7] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[4]]  
## [1] "evtree" "partykit" "mvtnorm" "libcoin" "grid" "stats"   
## [7] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[5]]  
## [1] "evtree" "partykit" "mvtnorm" "libcoin" "grid" "stats"   
## [7] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[6]]  
## [1] "evtree" "partykit" "mvtnorm" "libcoin" "grid" "stats"   
## [7] "graphics" "grDevices" "utils" "datasets" "methods" "base"

Der Trainingsindex wird erstellt. Damit werden die Trainings- und Testdaten erzeugt. Hier wird eine 10-fache Kreuzvalidierung verwendet, ohne Wiederholung, um den Rechenaufwand gering zu halten.

train\_index <- createMultiFolds(MRSA\_schweineherden\_reduced$HERD\_MRSA,  
 k = 10,  
 times = 5)

Die eigentlich Kreuzvalidierung. Zur Parallelisierung wird die paralleliserte Version von *lapply* verwendet. Da *parLapply* eine Liste zurück gibt, aber ein Vektor gebraucht wird, wird *pred\_class* in einen Vektor umgewandelt.

start\_time <- Sys.time()  
pred\_class <- parLapply(cl = cl,  
 train\_index,  
 function(training\_rows, data\_set)  
 {  
 df\_train <- data\_set[ training\_rows,]  
 df\_test <- data\_set[-training\_rows,]  
 trained\_model <- evtree(HERD\_MRSA ~ .,  
 data = df\_train,  
 control = evtree.control(niterations = 100000L,  
 ntrees = 5000L,  
 seed = -1L,  
 minbucket = 10L,  
 minsplit = 20L))  
 predict(trained\_model,  
 new = df\_test)  
 }, data\_set = MRSA\_schweineherden\_reduced)  
pred\_class <- unlist(pred\_class)  
actual\_class <- unlist(lapply(train\_index,   
 function(x)   
 MRSA\_schweineherden\_reduced$HERD\_MRSA[-x]))  
print(Sys.time() - start\_time)

## Time difference of 30.70607 mins

Die Konfusionsmatrix und weitere Statisken werden erzeugt.

confusionMatrix(data = pred\_class,  
 reference = actual\_class,   
 positive = "positive")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction negative positive  
## negative 459 206  
## positive 471 864  
##   
## Accuracy : 0.6615   
## 95% CI : (0.6403, 0.6822)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.3067   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.8075   
## Specificity : 0.4935   
## Pos Pred Value : 0.6472   
## Neg Pred Value : 0.6902   
## Prevalence : 0.5350   
## Detection Rate : 0.4320   
## Detection Prevalence : 0.6675   
## Balanced Accuracy : 0.6505   
##   
## 'Positive' Class : positive   
##

stopCluster(cl)