Aufgabe2_DM2

busra

6 Juli 2018

```
library('dplyr') # data manipulation
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

```
####################################

trainspater <- read.csv('C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/KAGGLE_TITANIC/tra
testspater  <- read.csv('C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/KAGGLE_TITANIC/tes

# binde training & test data für später eigene Datapartition
full  <- bind_rows(trainspater, testspater)
```

```
# check data
str(full)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
summary(full)
```

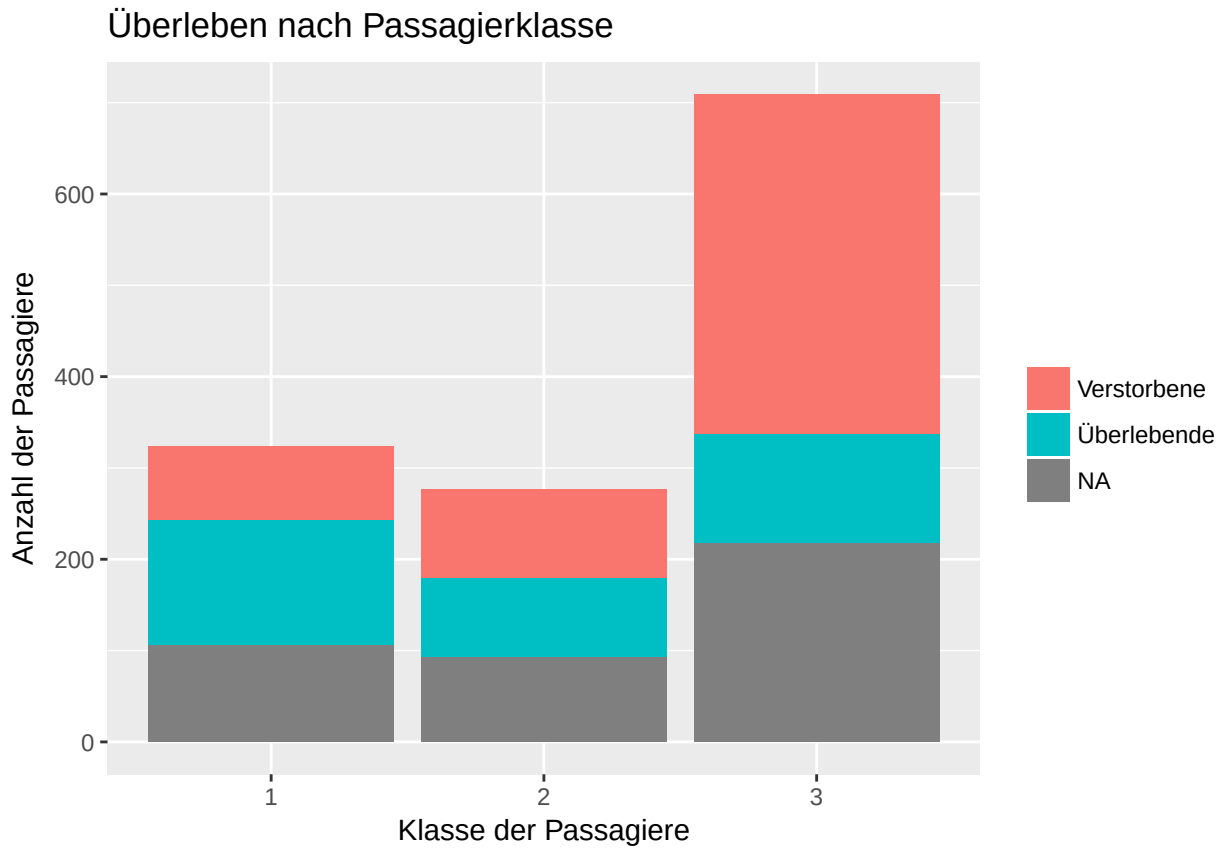
```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
```

```
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##     Ticket               Fare            Cabin
##  Length:1309        Min.   :  0.000   Length:1309
##  Class :character   1st Qu.:  7.896   Class :character
##  Mode  :character   Median : 14.454   Mode  :character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
##    Embarked
##  Length:1309
##  Class :character
##  Mode  :character
##
##
##
##
```

```
#head(full)
```

```
barchart <- ggplot(full, aes(as.factor(Pclass), fill=as.factor(Survived)))+geom_bar()

barchart+xlab("Klasse der Passagiere")+ylab("Anzahl der Passagiere")+ggtitle("Überleben nach Passagierk
```



Ab hier pdf version

```
# Load the raw training data and replace missing values with NA
training.data.raw <- read.csv('C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/KAGGLE_TITANI

# Output the number of missing values for each column
sapply(training.data.raw,function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

```
# A visual way to check for missing data
#install.packages("Amelia")
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 3.4.4
```

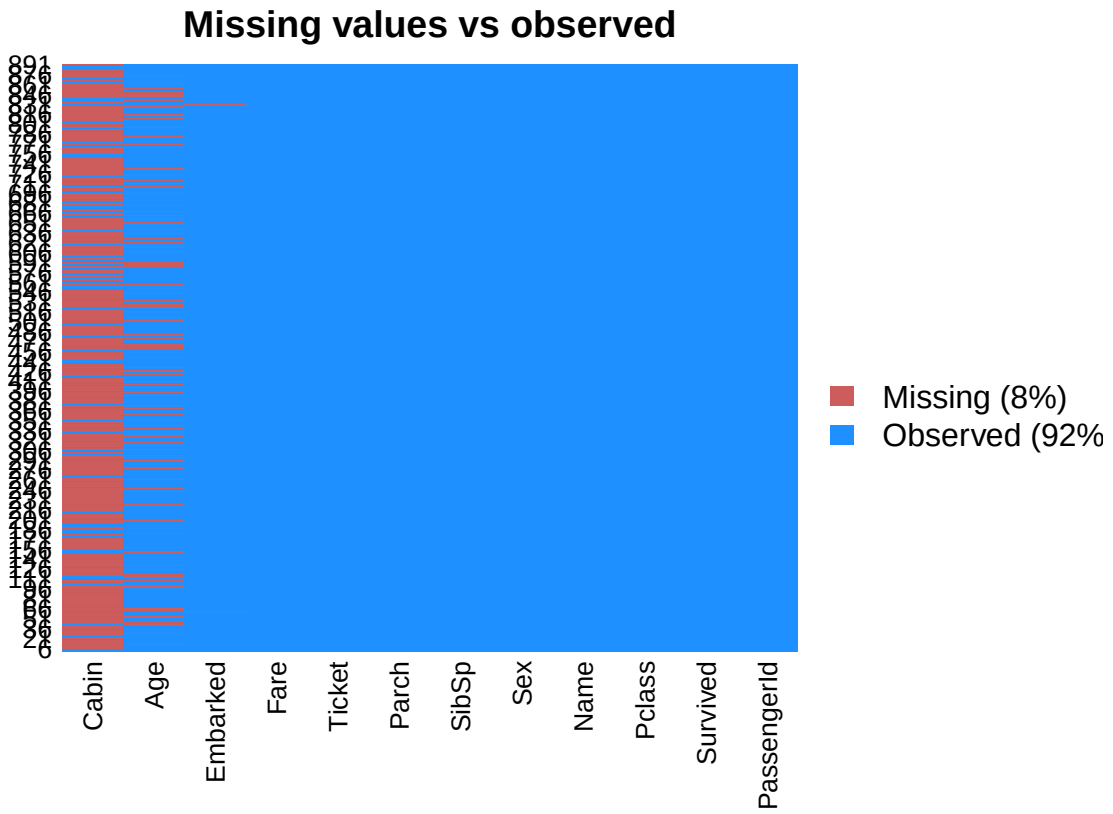
```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 3.4.4
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
```

```
## ##
```

```
missmap(training.data.raw, main = "Missing values vs observed")
```




```
write.table(training.data.raw, file = "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Tita
```

```
# Subsetting the data
data <- subset(training.data.raw,select=c(2,3,5,6,7,8,10,12))
write.table(data, file = "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Titanic_Subset1da
```

```
# Substitute the missing values with the average value
data$Age[is.na(data$Age)] <- mean(data$Age,na.rm=T)
```

```
# Remove rows (Embarked) with NAs
data <- data[!is.na(data$Embarked),]
```

```
rownames(data) <- NULL
```

```
write.table(data, file = "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Titanic_Subset2.cs
```

```
#die traininingsdaten an sich werden in: train & valid !!!!!!
#  Splitting into separate Train and  Test (=Validation Data)   |  IMPORTANT !
train <- data[1:800,]
test <- data[801:889,]
```

```
# write out text datafile and a SAS program to read it
library(foreign)
write.foreign(train, "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Titanic_train_0.txt",
write.foreign(test,  "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Titanic_test_0.txt",
```

```
# --> brauche das für SAS Eminer später!!!

#write.foreign(test, datafile="testingg.csv", codefile="testingg.sas7bdat", package="SAS")
#funktioniert auch nicht



#write.foreign(test, datafile="testingg.csv", codefile="testingg.sas", package="SAS")
#import("C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/TITANIC/testingg.csv")
#convert("testingg.csv", "tttt.sas7bdat")
#funktioniert auch nicht!!!!
```

```
#install.packages("rio")
library("rio")
```

```
## Warning: package 'rio' was built under R version 3.4.4
```

```
export(train, "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/r_mydata.csv")
export(train, "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/r_mydata.sas7bdat")

# ACHTUNG ::   ERRROR
# sas7bdat von  rio  wird von  SAS  nicht erkannt  !



write.table(train, file = "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Titanic_train_w.c



write.table(test, file = "C:/BÜSRA/Uni/Master/B Fächer/Data Mining 2/HELM_2018/KAGGLE/Titanic_test_w.csv
```

```
table(train$Survived)
```

```
##
##   0   1
## 493 307
```

```
### kein richtiger Imbalance, da Survided: 38,375% und Tote: 61,562% -- 307 zu 493 bei 800 traningsdater
```

```
table(test$Survived)
```

```
##
##  0  1
## 56 33
```

```
### logistische Regression mit Cross-Validation
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

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

```
## Warning: package 'lattice' was built under R version 3.4.3
```

```
# definiere training control
train_control<- trainControl(method="cv", number=800)

# trainere das Modell
```

```
modeloo<- train(as.factor(Survived) ~., data=train, trControl=train_control, method="glm", family=binom
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

```
summary(modeloo)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6064  -0.5954  -0.4254   0.6220   2.4165
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.137627   0.594998   8.635  < 2e-16 ***
## Pclass      -1.087156   0.151168  -7.192 6.40e-13 ***
## Sexmale     -2.756819   0.212026 -13.002  < 2e-16 ***
## Age         -0.037267   0.008195  -4.547 5.43e-06 ***
## SibSp       -0.292920   0.114642  -2.555   0.0106 *
## Parch       -0.116576   0.128127  -0.910   0.3629
## Fare         0.001528   0.002353   0.649   0.5160
## EmbarkedQ   -0.002656   0.400882  -0.007   0.9947
## EmbarkedS   -0.318786   0.252960  -1.260   0.2076
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  709.39  on 791  degrees of freedom
## AIC: 727.39
##
## Number of Fisher Scoring iterations: 5
```

```
# Model fitting --- logistische Regression
model <- glm(Survived ~.,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6064  -0.5954  -0.4254   0.6220   2.4165
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.137627   0.594998   8.635  < 2e-16 ***
## Pclass      -1.087156   0.151168  -7.192 6.40e-13 ***
## Sexmale     -2.756819   0.212026 -13.002  < 2e-16 ***
```

```
## Age         -0.037267   0.008195  -4.547 5.43e-06 ***
## SibSp       -0.292920   0.114642  -2.555   0.0106 *
## Parch       -0.116576   0.128127  -0.910   0.3629
## Fare         0.001528   0.002353   0.649   0.5160
## EmbarkedQ   -0.002656   0.400882  -0.007   0.9947
## EmbarkedS   -0.318786   0.252960  -1.260   0.2076
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  709.39  on 791  degrees of freedom
## AIC: 727.39
##
## Number of Fisher Scoring iterations: 5
```

```
# Analysis of deviance
anova(model,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                       799    1065.39
## Pclass    1   83.607       798     981.79 < 2.2e-16 ***
## Sex       1  240.014       797     741.77 < 2.2e-16 ***
## Age       1   17.495       796     724.28 2.881e-05 ***
## SibSp     1   10.842       795     713.43  0.000992 ***
## Parch     1    0.863       794     712.57  0.352873
## Fare      1    0.994       793     711.58  0.318717
## Embarked  2    2.187       791     709.39  0.334990
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#bei model2 nur die signifkanten Einflussfaktoren mit in Log. Reg. genommem!
model2 <- glm(Survived ~ Pclass + Sex + Age, family=binomial(link='logit'),data=train)
summary(model2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6320  -0.6570  -0.4239   0.6420   2.4093
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  4.633935   0.472667   9.804  < 2e-16 ***
## Pclass      -1.139026   0.125153  -9.101  < 2e-16 ***
## Sexmale     -2.646164   0.197657 -13.388  < 2e-16 ***
## Age         -0.031477   0.007724  -4.075  4.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  724.28  on 796  degrees of freedom
## AIC: 732.28
##
## Number of Fisher Scoring iterations: 4
```

```
# McFadden R^2
#install.packages("pscl")
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 3.4.3
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
pR2(model)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -354.6950111 -532.6961008  356.0021794    0.3341513    0.3591775
##         r2CU
##    0.4880244
```

```
pR2(model2)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -362.1385144 -532.6961008  341.1151728    0.3201780    0.3471409
##         r2CU
##    0.4716700
```

```
#------------------------------------------------------------------------------
# MEASURING THE PREDICTIVE ABILITY OF THE MODEL

# If prob > 0.5 then 1, else 0. Threshold can be set for better results
fitted.results <- predict(model,newdata=subset(test,select=c(2,3,4,5,6,7,8)),type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
```

```
fitted.results2 <- predict(model2,newdata=subset(test,select=c(2,3,4,5)),type='response')
fitted.results2 <- ifelse(fitted.results2 > 0.5,1,0)
```

```
#gibt diesen FEHLERMELDUNG: `data` and `reference` should be factors with the same levels.
# daher table() noch drumherum!
# Confusion matrix
library(caret)
confusionMatrix(table(data=fitted.results, reference=test$Survived))
```

```
## Confusion Matrix and Statistics
##
##     reference
## data  0  1
##    0 51  9
##    1  5 24
##
##                Accuracy : 0.8427
##                  95% CI : (0.7502, 0.9112)
##     No Information Rate : 0.6292
##     P-Value [Acc > NIR] : 8.248e-06
##
##                   Kappa : 0.6543
##  Mcnemar's Test P-Value : 0.4227
##
##             Sensitivity : 0.9107
##             Specificity : 0.7273
##          Pos Pred Value : 0.8500
##          Neg Pred Value : 0.8276
##              Prevalence : 0.6292
##          Detection Rate : 0.5730
##    Detection Prevalence : 0.6742
##       Balanced Accuracy : 0.8190
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(table(data=fitted.results2, reference=test$Survived))
```

```
## Confusion Matrix and Statistics
##
##     reference
## data  0  1
##    0 48 10
##    1  8 23
##
##                Accuracy : 0.7978
##                  95% CI : (0.6993, 0.8755)
##     No Information Rate : 0.6292
##     P-Value [Acc > NIR] : 0.0004642
##
##                   Kappa : 0.5611
##  Mcnemar's Test P-Value : 0.8136637
##
##             Sensitivity : 0.8571
##             Specificity : 0.6970
##          Pos Pred Value : 0.8276
##          Neg Pred Value : 0.7419
##              Prevalence : 0.6292
##          Detection Rate : 0.5393
##    Detection Prevalence : 0.6517
##       Balanced Accuracy : 0.7771
##
##        'Positive' Class : 0
##
```

```
## hier sogar weniger accuracy ---nicht gut --also lieber normale model verwenden
```

```
# ROC and AUC
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.4.3
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.4.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
p <- predict(model, newdata=subset(test,select=c(2,3,4,5,6,7,8)), type="response")
pr <- prediction(p, test$Survived)
```

```
p
```

```
##         801         802         803         804         805         806
## 0.753963539 0.555339553 0.268966522 0.100199823 0.087635519 0.382591374
##         807         808         809         810         811         812
## 0.710629291 0.175686502 0.908053970 0.103739874 0.068118069 0.197720714
##         813         814         815         816         817         818
## 0.494184371 0.089171014 0.467061844 0.670916792 0.213873799 0.057753258
##         819         820         821         822         823         824
## 0.066521634 0.821595593 0.100432614 0.391432165 0.611505040 0.075747618
##         825         826         827         828         829         830
## 0.121371738 0.097977308 0.498158813 0.121502197 0.740067460 0.371903262
##         831         832         833         834         835         836
## 0.121700886 0.114599892 0.134986038 0.910083153 0.122515421 0.091625068
##         837         838         839         840         841         842
## 0.090655969 0.557799989 0.126453986 0.333472642 0.951654603 0.103720965
##         843         844         845         846         847         848
## 0.139463537 0.059909635 0.008355321 0.102205818 0.227599634 0.941965726
##         849         850         851         852         853         854
## 0.062646787 0.018977299 0.760342331 0.956020337 0.679523342 0.686600366
##         855         856         857         858         859         860
## 0.869568623 0.292103154 0.659585257 0.121700886 0.035857454 0.236809924
##         861         862         863         864         865         866
## 0.878945394 0.117155477 0.271536987 0.750104004 0.843612428 0.474212324
##         867         868         869         870         871         872
## 0.091809697 0.149186396 0.103741054 0.839019072 0.438484070 0.050342131
##         873         874         875         876         877         878
## 0.840719304 0.790541057 0.126778652 0.130623230 0.091605453 0.878042770
##         879         880         881         882         883         884
## 0.836990876 0.081870226 0.679954623 0.242370054 0.107133934 0.470461277
##         885         886         887         888         889
## 0.249994665 0.955615201 0.490084348 0.591592661 0.112642486
```

```
fitted.results
```

```
## 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818
##   1   1   0   0   0   0   1   0   1   0   0   0   0   0   0   1   0   0
```

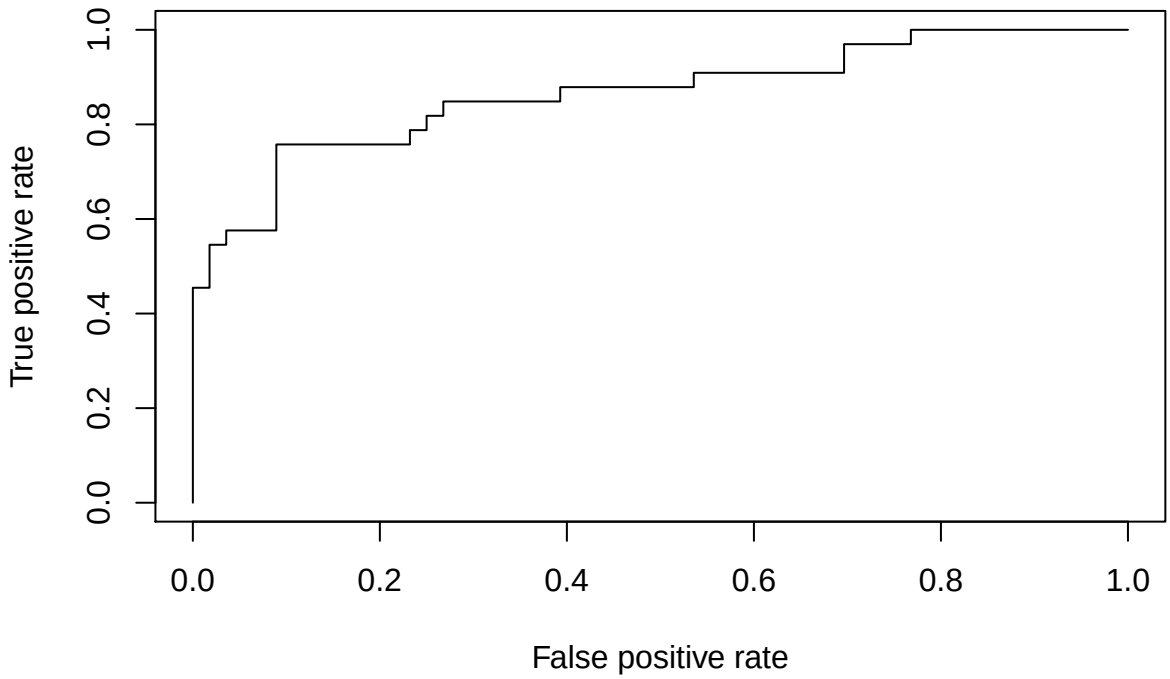
```
## 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836
##   0   1   0   0   1   0   0   0   0   0   1   0   0   0   0   1   0   0
## 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854
##   0   1   0   0   1   0   0   0   0   0   0   1   0   0   1   1   1   1
## 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872
##   1   0   1   0   0   0   1   0   0   1   1   0   0   0   0   1   0   0
## 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889
##   1   1   0   0   0   1   1   0   1   0   0   0   0   1   0   1   0
```

```
test$Survived
```

```
##  [1] 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 0 1 0
## [36] 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0 1 0 0 1 1 0 0 1 0 1
## [71] 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0
```

```
# TPR = sensitivity, FPR=specificity

prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

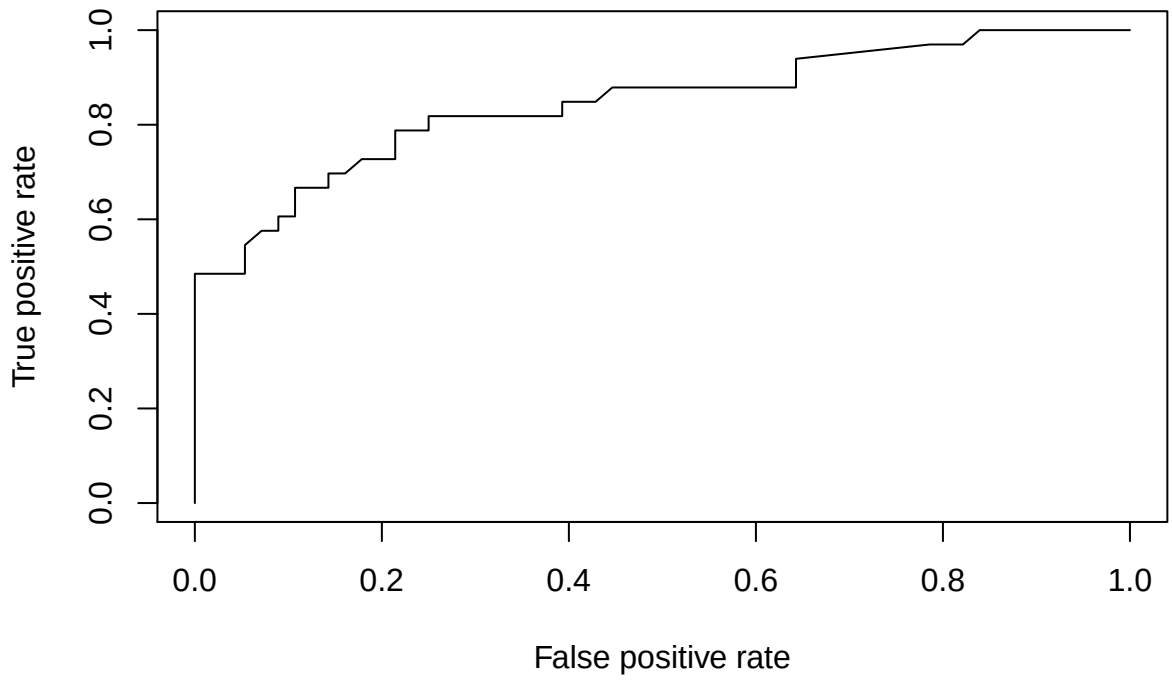
```
## [1] 0.8647186
```

```
# ROC and AUC für Modell 2
```

```
p2 <- predict(model2, newdata=subset(test,select=c(2,3,4,5,6,7,8)), type="response")
pr2 <- prediction(p2, test$Survived)

# TPR = sensitivity, FPR=specificity

prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")
plot(prf2)
```




```
auc2 <- performance(pr2, measure = "auc")
auc2 <- auc2@y.values[[1]]
auc2
```

```
## [1] 0.844697
```

```
# nur odds ratios
exp(coef(model))
```

```
##  (Intercept)       Pclass      Sexmale          Age        SibSp
## 170.31116493   0.33717391   0.06349342   0.96341847   0.74608157
##        Parch         Fare    EmbarkedQ    EmbarkedS
##   0.88996209   1.00152965   0.99734737   0.72703115
```

```
# odds ratios und 95% Konfidenzeintervall

exp(cbind("Odds ratio" = coef(model), confint.default(model, level = 0.95)))
```

```
##               Odds ratio       2.5 %       97.5 %
## (Intercept) 170.31116493 53.06157608 546.64589790
```

```
## Pclass        0.33717391  0.25071463   0.45344878
## Sexmale       0.06349342  0.04190364   0.09620678
## Age           0.96341847  0.94806687   0.97901864
## SibSp         0.74608157  0.59594081   0.93404865
## Parch         0.88996209  0.69232428   1.14401956
## Fare          1.00152965  0.99692057   1.00616004
## EmbarkedQ     0.99734737  0.45458465   2.18815521
## EmbarkedS     0.72703115  0.44282640   1.19363772
```

Unterschiedliche train/test datensätze

```
#split data into train(60%) and test(40%) --- A
set.seed(12345)
traina<-sample_frac(data, 0.6)
sida<-as.numeric(rownames(traina)) # because rownames() returns character
testa<-data[-sida,]
```

```
# Model fitting --- logistische Regression für A
modela <- glm(Survived ~.,family=binomial(link='logit'),data=traina)
summary(modela)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = traina)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7070  -0.6171  -0.4032   0.6431   2.5176
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.307468   0.713954   7.434 1.05e-13 ***
## Pclass      -1.164082   0.184945  -6.294 3.09e-10 ***
## Sexmale     -2.685712   0.260642 -10.304  < 2e-16 ***
## Age         -0.043100   0.010101  -4.267 1.98e-05 ***
## SibSp       -0.195717   0.131931  -1.483    0.138
## Parch       -0.133584   0.146807  -0.910    0.363
## Fare         0.002509   0.002807   0.894    0.372
## EmbarkedQ   -0.203258   0.484099  -0.420    0.675
## EmbarkedS   -0.336471   0.297581  -1.131    0.258
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 702.22  on 532  degrees of freedom
## Residual deviance: 464.50  on 524  degrees of freedom
## AIC: 482.5
##
## Number of Fisher Scoring iterations: 5
```

```
pR2(modela)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -232.2521911 -351.1114348  237.7184874    0.3385229    0.3598168
##         r2CU
##    0.4914230
```

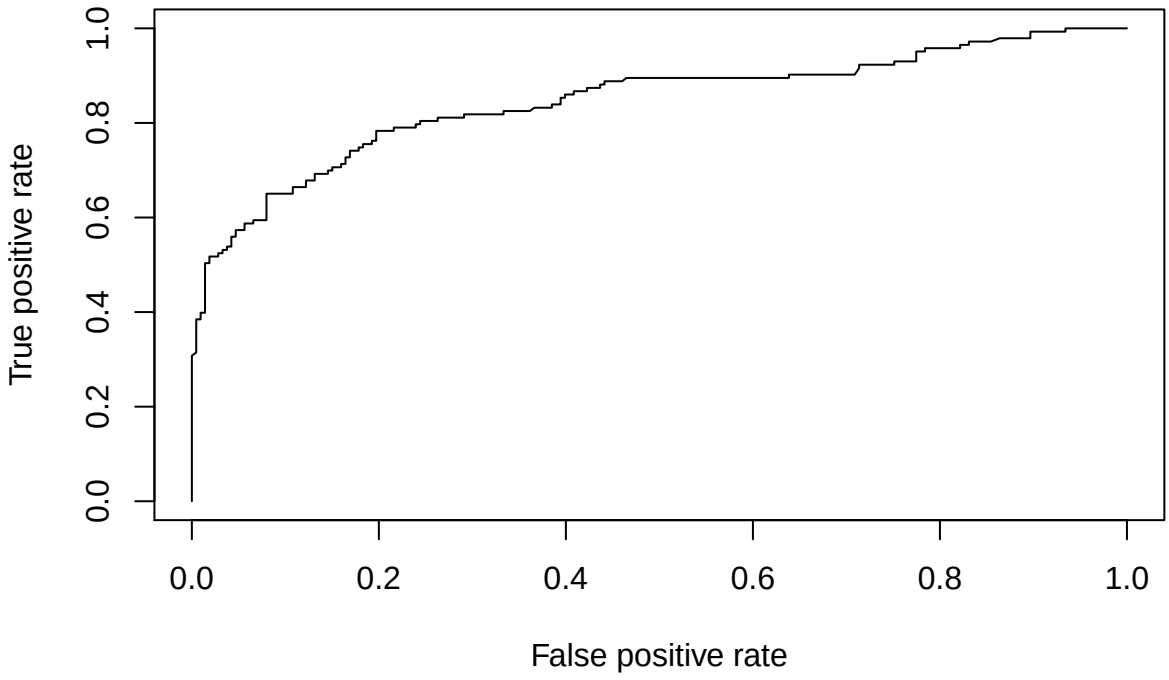
```
fitted.resultsa <- predict(modela,newdata=subset(testa,select=c(2,3,4,5,6,7,8)),type='response')
fitted.resultsa <- ifelse(fitted.resultsa > 0.5,1,0)
```

```
confusionMatrix(table(data=fitted.resultsa, reference=testa$Survived))
```

```
## Confusion Matrix and Statistics
##
##     reference
## data   0   1
##    0 185  46
##    1  28  97
##
##                Accuracy : 0.7921
##                  95% CI : (0.7462, 0.8331)
##     No Information Rate : 0.5983
##     P-Value [Acc > NIR] : 5.399e-15
##
##                   Kappa : 0.5584
##  Mcnemar's Test P-Value : 0.04813
##
##             Sensitivity : 0.8685
##             Specificity : 0.6783
##          Pos Pred Value : 0.8009
##          Neg Pred Value : 0.7760
##              Prevalence : 0.5983
##          Detection Rate : 0.5197
##    Detection Prevalence : 0.6489
##       Balanced Accuracy : 0.7734
##
##        'Positive' Class : 0
##
```

```
# ROC and AUC für A


pa <- predict(modela, newdata=subset(testa,select=c(2,3,4,5,6,7,8)), type="response")
pra <- prediction(pa, testa$Survived)
prfa <- performance(pra, measure = "tpr", x.measure = "fpr")
plot(prfa)
```




```
auca <- performance(pra, measure = "auc")
auca <- auca@y.values[[1]]
auca
```

```
## [1] 0.8460718
```

```
#split data into train(70%) and test(30%) --- B
set.seed(12345)
trainb<-sample_frac(data, 0.7)
sidb<-as.numeric(rownames(trainb)) # because rownames() returns character
testb<-data[-sidb,]
```

```
# Model fitting --- logistische Regression für B
modelb <- glm(Survived ~.,family=binomial(link='logit'),data=trainb)
summary(modelb)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = trainb)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6233  -0.6260  -0.4320   0.6927   2.4253
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.043751   0.649223   7.769 7.92e-15 ***
```

```
## Pclass      -1.092431   0.167947  -6.505 7.79e-11 ***
## Sexmale     -2.623022   0.235964 -11.116  < 2e-16 ***
## Age         -0.038604   0.009129  -4.229 2.35e-05 ***
## SibSp       -0.275967   0.130381  -2.117   0.0343 *
## Parch       -0.106149   0.136387  -0.778   0.4364
## Fare         0.002198   0.002681   0.820   0.4124
## EmbarkedQ   -0.083677   0.435170  -0.192   0.8475
## EmbarkedS   -0.310637   0.276428  -1.124   0.2611
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 826.72  on 621  degrees of freedom
## Residual deviance: 561.73  on 613  degrees of freedom
## AIC: 579.73
##
## Number of Fisher Scoring iterations: 5
```

```
pR2(modelb)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -280.8648544 -413.3598005  264.9898922    0.3205318    0.3469024
##         r2CU
##    0.4717881
```

```
fitted.resultsb <- predict(modelb,newdata=subset(testb,select=c(2,3,4,5,6,7,8)),type='response')
fitted.resultsb <- ifelse(fitted.resultsb > 0.5,1,0)
```

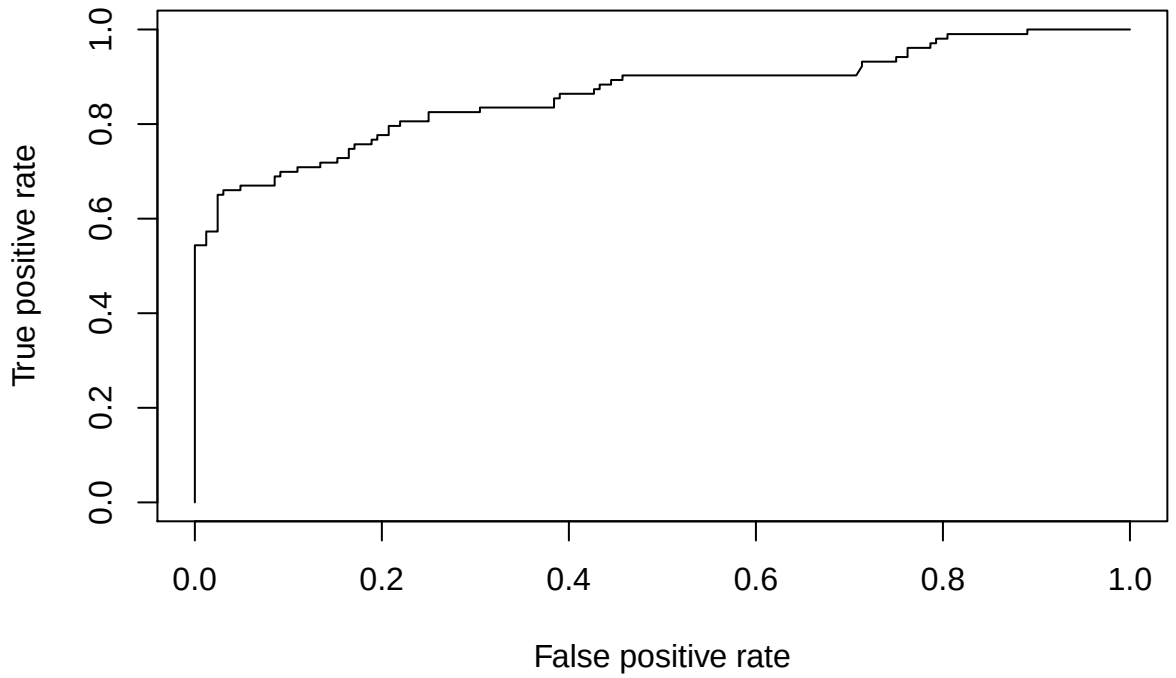
```
confusionMatrix(table(data=fitted.resultsb, reference=testb$Survived))
```

```
## Confusion Matrix and Statistics
##
##     reference
## data   0   1
##    0 149  32
##    1  15  71
##
##                Accuracy : 0.824
##                  95% CI : (0.7729, 0.8677)
##     No Information Rate : 0.6142
##     P-Value [Acc > NIR] : 8.517e-14
##
##                   Kappa : 0.6168
##  Mcnemar's Test P-Value : 0.0196
##
##             Sensitivity : 0.9085
##             Specificity : 0.6893
##          Pos Pred Value : 0.8232
##          Neg Pred Value : 0.8256
##              Prevalence : 0.6142
##          Detection Rate : 0.5581
##    Detection Prevalence : 0.6779
##       Balanced Accuracy : 0.7989
##
##        'Positive' Class : 0
```

```
##
```

```
# ROC and AUC für B


pb <- predict(modelb, newdata=subset(testb,select=c(2,3,4,5,6,7,8)), type="response")
prb <- prediction(pb, testb$Survived)
prfb <- performance(prb, measure = "tpr", x.measure = "fpr")
plot(prfb)
```




```
aucb <- performance(prb, measure = "auc")
aucb <- aucb@y.values[[1]]
aucb
```

```
## [1] 0.8650841
```

```
#split data into train(80%) and test(20%) --- C
set.seed(12345)
trainc<-sample_frac(data, 0.8)
sidc<-as.numeric(rownames(trainc)) # because rownames() returns character
testc<-data[-sidc,]
```

```
# Model fitting --- logistische Regression für C
modelc <- glm(Survived ~.,family=binomial(link='logit'),data=trainc)
summary(modelc)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
```

```
##     data = trainc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6844  -0.5859  -0.4271   0.6282   2.4529
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.124317   0.630578   8.126 4.42e-16 ***
## Pclass      -1.078624   0.160038  -6.740 1.59e-11 ***
## Sexmale     -2.747935   0.225889 -12.165  < 2e-16 ***
## Age         -0.041516   0.008892  -4.669 3.03e-06 ***
## SibSp       -0.285191   0.125258  -2.277   0.0228 *
## Parch       -0.135109   0.129054  -1.047   0.2951
## Fare         0.002774   0.002672   1.038   0.2993
## EmbarkedQ   -0.072494   0.424569  -0.171   0.8644
## EmbarkedS   -0.252285   0.264601  -0.953   0.3404
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 940.10  on 710  degrees of freedom
## Residual deviance: 622.21  on 702  degrees of freedom
## AIC: 640.21
##
## Number of Fisher Scoring iterations: 5
```

```
pR2(modelc)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -311.1033771 -470.0510239  317.8952937    0.3381498    0.3605265
##         r2CU
##    0.4915432
```

```
fitted.resultsc <- predict(modelc,newdata=subset(testc,select=c(2,3,4,5,6,7,8)),type='response')
fitted.resultsc <- ifelse(fitted.resultsc > 0.5,1,0)
```

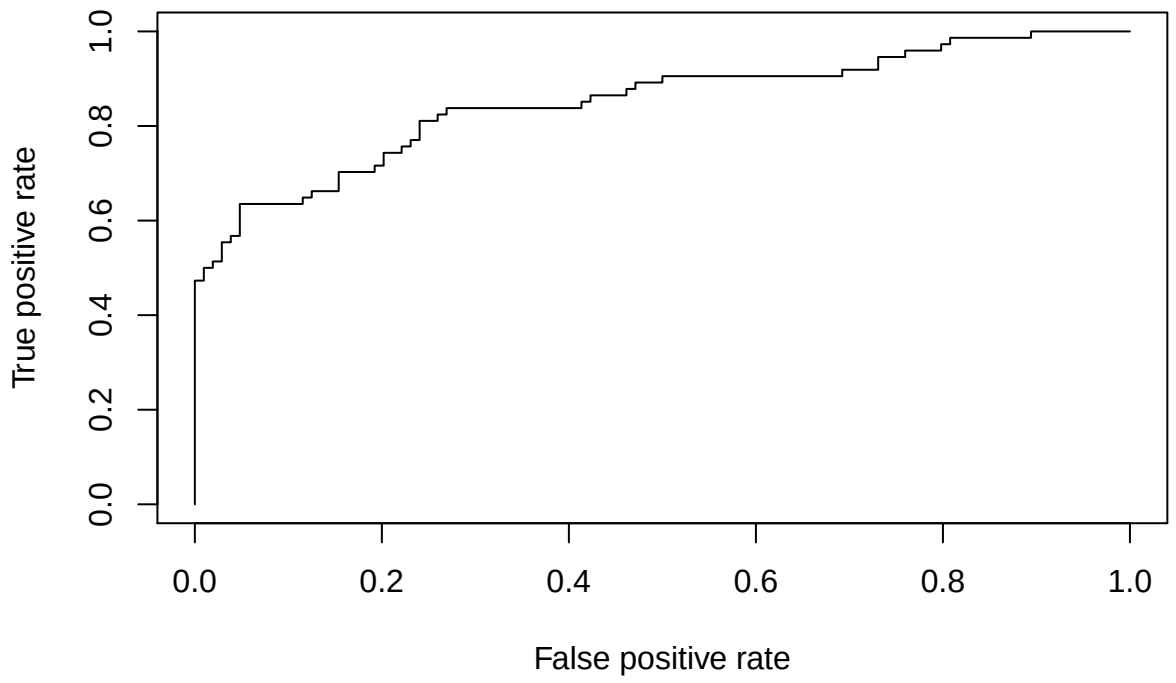
```
confusionMatrix(table(data=fitted.resultsc, reference=testc$Survived))
```

```
## Confusion Matrix and Statistics
##
##     reference
## data  0  1
##    0 88 24
##    1 16 50
##
##                Accuracy : 0.7753
##                  95% CI : (0.7068, 0.8343)
##     No Information Rate : 0.5843
##     P-Value [Acc > NIR] : 6.42e-08
##
##                   Kappa : 0.5301
##  Mcnemar's Test P-Value : 0.2684
##
##             Sensitivity : 0.8462
```

```
##             Specificity : 0.6757
##          Pos Pred Value : 0.7857
##          Neg Pred Value : 0.7576
##              Prevalence : 0.5843
##          Detection Rate : 0.4944
##    Detection Prevalence : 0.6292
##       Balanced Accuracy : 0.7609
##
##        'Positive' Class : 0
##
```

```
# ROC and AUC für C


pc <- predict(modelc, newdata=subset(testc,select=c(2,3,4,5,6,7,8)), type="response")
prc <- prediction(pc, testc$Survived)
prfc <- performance(prc, measure = "tpr", x.measure = "fpr")
plot(prfc)
```




```
aucc <- performance(prc, measure = "auc")
aucc <- aucc@y.values[[1]]
aucc
```

```
## [1] 0.8501819
```