R Notebook

# Veri setini yukleme

#setwd("~/Desktop/datasciencewithr")  
#getwd()  
library(readr)  
library(dplyr)

##   
## 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(readr)  
mamography <- read\_csv("odev\_dataset.csv",   
 col\_types = cols(CaTypeO = col\_skip(),   
 ptid = col\_skip()))  
#View(mamography)  
dataframe <- mamography  
cancer\_c <- ifelse(mamography$cancer\_c > 0.5, "Yes","No")  
dataframe <- select(dataframe, -c(cancer\_c))  
dataframe <- cbind(dataframe,cancer\_c)  
# veriye ilk bakis  
colnames(dataframe)

## [1] "age\_c" "assess\_c" "compfilm\_c" "density\_c" "famhx\_c"   
## [6] "hrt\_c" "prvmam\_c" "biophx\_c" "mammtype" "bmi\_c"   
## [11] "cancer\_c"

nrow(dataframe)

## [1] 40000

ncol(dataframe)

## [1] 11

head(dataframe)

## age\_c assess\_c compfilm\_c density\_c famhx\_c hrt\_c prvmam\_c biophx\_c  
## 1 62 1 1 2 0 0 1 0  
## 2 65 1 1 4 0 0 1 0  
## 3 69 0 1 2 0 0 1 0  
## 4 64 2 1 2 0 0 1 0  
## 5 63 3 1 2 0 0 1 1  
## 6 65 2 1 3 0 0 1 0  
## mammtype bmi\_c cancer\_c  
## 1 1 24.02354 No  
## 2 1 -99.00000 No  
## 3 1 29.05243 No  
## 4 1 -99.00000 No  
## 5 1 33.72952 No  
## 6 2 -99.00000 No

# Veri seti On Isleme

## Veri seti ozet istatistikleri

# verisetinin ozetine ulastik  
#install.packages("dplyr")  
library("dplyr")  
summary(dataframe)

## age\_c assess\_c compfilm\_c density\_c   
## Min. :60.00 Min. :0.000 Min. :0.000 Min. :1.00   
## 1st Qu.:63.00 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:2.00   
## Median :68.00 Median :1.000 Median :1.000 Median :2.00   
## Mean :69.56 Mean :1.203 Mean :1.903 Mean :2.23   
## 3rd Qu.:75.00 3rd Qu.:2.000 3rd Qu.:1.000 3rd Qu.:3.00   
## Max. :89.00 Max. :5.000 Max. :9.000 Max. :4.00   
## famhx\_c hrt\_c prvmam\_c biophx\_c   
## Min. :0.0000 Min. :0.0000 Min. :0.000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1.000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000 Median :1.000 Median :0.0000   
## Mean :0.2199 Mean :0.5049 Mean :1.108 Mean :0.4447   
## 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.000 3rd Qu.:1.0000   
## Max. :9.0000 Max. :9.0000 Max. :9.000 Max. :9.0000   
## mammtype bmi\_c cancer\_c   
## Min. :1.0 Min. :-99.00 No :39741   
## 1st Qu.:1.0 1st Qu.:-99.00 Yes: 259   
## Median :1.5 Median :-99.00   
## Mean :1.5 Mean :-46.16   
## 3rd Qu.:2.0 3rd Qu.: 24.69   
## Max. :2.0 Max. : 71.72

glimpse(dataframe)

## Observations: 40,000  
## Variables: 11  
## $ age\_c <dbl> 62, 65, 69, 64, 63, 65, 65, 75, 64, 66, 78, 70, 67,...  
## $ assess\_c <dbl> 1, 1, 0, 2, 3, 2, 1, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, ...  
## $ compfilm\_c <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, ...  
## $ density\_c <dbl> 2, 4, 2, 2, 2, 3, 3, 2, 2, 3, 2, 2, 1, 2, 2, 3, 2, ...  
## $ famhx\_c <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...  
## $ hrt\_c <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 9, 0, 0, ...  
## $ prvmam\_c <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, ...  
## $ biophx\_c <dbl> 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, ...  
## $ mammtype <dbl> 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, ...  
## $ bmi\_c <dbl> 24.02354, -99.00000, 29.05243, -99.00000, 33.72952,...  
## $ cancer\_c <fct> No, No, No, No, No, No, No, No, No, No, No, No, No,...

# 9 degerlerini bos deger yaptik  
dataframe[dataframe == 9] <- NA  
dataframe[dataframe == -99] <- NA  
# bos degerlerin indekslerini bulduk  
#which(is.na(dataframe))  
# kac tane bos deger olduguna ulastik  
sum(is.na(dataframe))

## [1] 31282

## Veri setindeki bos degerlerin gorsellestirilmesi

#install.packages("VIM")  
library(VIM)

## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## VIM is ready to use.   
## Since version 4.0.0 the GUI is in its own package VIMGUI.  
##   
## Please use the package to use the new (and old) GUI.

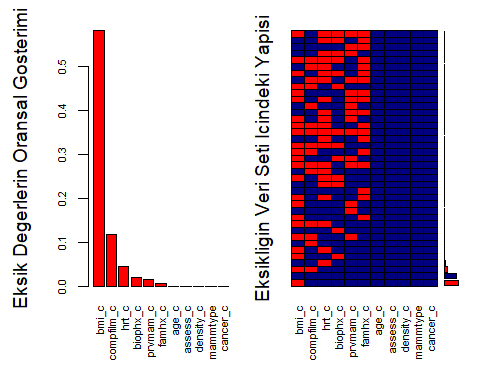
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':  
##   
## sleep

# eksiklikleri gozlemliyoruz  
# buradan birliktelik cikarimlari da yapilabilir  
aggr\_plot <- aggr(dataframe, col=c('navyblue','red'),   
 numbers = TRUE,   
 sortVars = TRUE,   
 labels = names(dataframe),   
 cex.axis=.7,   
 gap=3,   
 ylab=c("Eksik Degerlerin Oransal Gosterimi",  
 "Eksikligin Veri Seti Icindeki Yapisi"))

## Warning in plot.aggr(res, ...): not enough vertical space to display  
## frequencies (too many combinations)



##   
## Variables sorted by number of missings:   
## Variable Count  
## bmi\_c 0.580225  
## compfilm\_c 0.117000  
## hrt\_c 0.044300  
## biophx\_c 0.020375  
## prvmam\_c 0.014450  
## famhx\_c 0.005700  
## age\_c 0.000000  
## assess\_c 0.000000  
## density\_c 0.000000  
## mammtype 0.000000  
## cancer\_c 0.000000

## Veri setindeki bos degerli kayitlarin temizlenmesi

# eksik veri bulunduran kayıtları sildik  
dataframe <- na.omit(dataframe)  
# kac adet bos deger oldugunu bulduk  
sum(is.na(dataframe))

## [1] 0

# değişkenlerin özet istatistiklerine ulaştık  
library(funModeling)

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

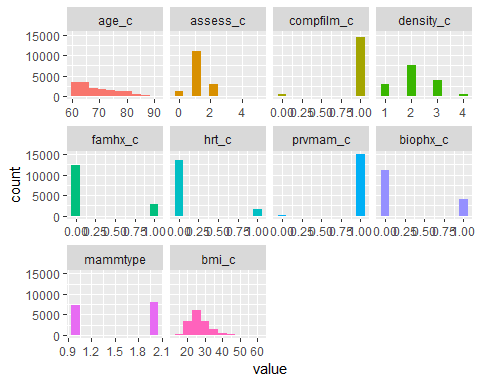
## The following objects are masked from 'package:base':  
##   
## format.pval, units

## funModeling v.1.7 :)  
## Examples and tutorials at livebook.datascienceheroes.com

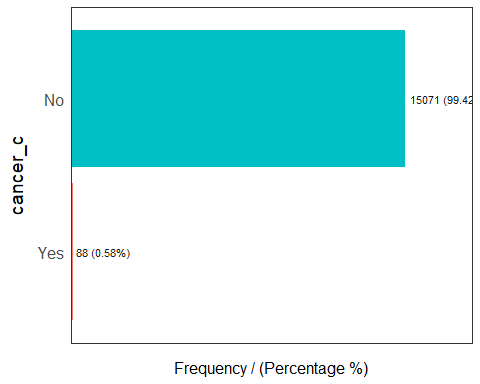
profiling\_num(dataframe)

## variable mean std\_dev variation\_coef p\_01 p\_05 p\_25 p\_50 p\_75 p\_95  
## 1 age\_c 69.01 7.133 0.103 60 60 63 67 74 83  
## 2 assess\_c 1.13 0.528 0.469 0 0 1 1 1 2  
## 3 compfilm\_c 0.96 0.192 0.200 0 1 1 1 1 1  
## 4 density\_c 2.14 0.764 0.356 1 1 2 2 3 3  
## 5 famhx\_c 0.19 0.390 2.088 0 0 0 0 0 1  
## 6 hrt\_c 0.10 0.300 3.004 0 0 0 0 0 1  
## 7 prvmam\_c 0.99 0.083 0.084 1 1 1 1 1 1  
## 8 biophx\_c 0.27 0.442 1.656 0 0 0 0 1 1  
## 9 mammtype 1.52 0.499 0.328 1 1 1 2 2 2  
## 10 bmi\_c 26.88 5.659 0.211 18 20 23 26 30 37  
## p\_99 skewness kurtosis iqr range\_98 range\_80  
## 1 87 0.666 2.5 11 [60, 87] [61, 80]  
## 2 2 0.403 4.5 0 [0, 2] [1, 2]  
## 3 1 -4.809 24.1 0 [0, 1] [1, 1]  
## 4 4 0.253 2.7 1 [1, 4] [1, 3]  
## 5 1 1.609 3.6 0 [0, 1] [0, 1]  
## 6 1 2.671 8.1 0 [0, 1] [0, 0]  
## 7 1 -11.833 141.0 0 [1, 1] [1, 1]  
## 8 1 1.052 2.1 1 [0, 1] [0, 1]  
## 9 2 -0.094 1.0 1 [1, 2] [1, 2]  
## 10 45 1.127 4.9 7 [17.83, 44.95] [20.73, 34.33]

#plot(dataframe)  
# sürekli değikenlerin nasıl dağıldığını görselleştirdik  
plot\_num(dataframe)



# kategorik değişkenler için kullanılan görselleştirme  
freq(dataframe)



## cancer\_c frequency percentage cumulative\_perc  
## 1 No 15071 99.42 99  
## 2 Yes 88 0.58 100

## Test-Train ayrimi

#install.packages("caret")  
library(caret)

##   
## Attaching package: 'caret'

## The following object is masked from 'package:survival':  
##   
## cluster

train\_indeks <- createDataPartition(dataframe$cancer\_c, p = 0.8, list = FALSE, times = 1)  
  
train <- dataframe[train\_indeks,]  
test <- dataframe[-train\_indeks,]  
  
train\_x <- train %>% dplyr::select(-cancer\_c)  
train\_y <- train$cancer\_c  
  
test\_x <- test %>% dplyr::select(-cancer\_c)  
test\_y <- test$cancer\_c  
# eğitim verisinin hem bagimli hem de bagimsiz degiskenlerini tuttugumuz bir dataframe  
training <- data.frame(train\_x, cancer\_c = train\_y)  
head(training$cancer\_c)

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

#as.numeric(training$cancer\_c)-1  
# bir lineer model kuruldu  
model\_lm <- lm(as.numeric(training$cancer\_c)-1 ~ ., data = training)  
summary(model\_lm)

##   
## Call:  
## lm(formula = as.numeric(training$cancer\_c) - 1 ~ ., data = training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0383 -0.0101 -0.0060 -0.0031 1.0570   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.41e-03 1.22e-02 -0.12 0.908   
## age\_c 2.35e-04 9.82e-05 2.39 0.017 \*   
## assess\_c -1.59e-02 1.32e-03 -12.05 < 2e-16 \*\*\*  
## compfilm\_c -2.34e-03 3.96e-03 -0.59 0.553   
## density\_c 9.61e-04 9.67e-04 0.99 0.320   
## famhx\_c 1.45e-03 1.77e-03 0.82 0.413   
## hrt\_c 9.44e-04 2.34e-03 0.40 0.687   
## prvmam\_c -1.56e-03 9.10e-03 -0.17 0.864   
## biophx\_c 6.48e-03 1.57e-03 4.13 3.6e-05 \*\*\*  
## mammtype 1.16e-03 1.40e-03 0.83 0.406   
## bmi\_c 2.53e-04 1.30e-04 1.94 0.052 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.076 on 12117 degrees of freedom  
## Multiple R-squared: 0.0137, Adjusted R-squared: 0.0129   
## F-statistic: 16.8 on 10 and 12117 DF, p-value: <2e-16

# Lojistik Regresyon

## Model

# modelimizin bir lojistik regresyon olduğunu binomial değişkeni ile belirtiyoruz  
model\_glm <- glm(cancer\_c ~ .,   
 data = training,   
 family = "binomial")  
levels(training$cancer\_c)[1]

## [1] "No"

summary(model\_glm)

##   
## Call:  
## glm(formula = cancer\_c ~ ., family = "binomial", data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.628 -0.086 -0.069 -0.057 5.784   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.4472 1.8083 -3.57 0.00036 \*\*\*  
## age\_c 0.0332 0.0164 2.03 0.04264 \*   
## assess\_c -2.7099 0.2441 -11.10 < 2e-16 \*\*\*  
## compfilm\_c 0.0985 0.5334 0.18 0.85353   
## density\_c -0.0530 0.1753 -0.30 0.76231   
## famhx\_c 0.1392 0.2868 0.49 0.62738   
## hrt\_c 0.0282 0.4120 0.07 0.94542   
## prvmam\_c -0.1307 1.1440 -0.11 0.90905   
## biophx\_c 0.7227 0.2485 2.91 0.00363 \*\*   
## mammtype 0.1415 0.2495 0.57 0.57064   
## bmi\_c 0.0270 0.0206 1.31 0.19060   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 871.55 on 12127 degrees of freedom  
## Residual deviance: 712.43 on 12117 degrees of freedom  
## AIC: 734.4  
##   
## Number of Fisher Scoring iterations: 9

options(scipen = 9)

## Tahmin

head(predict(model\_glm))

## 1 8 10 13 17 22   
## -6.5 -5.3 -6.6 -5.3 -9.2 -8.8

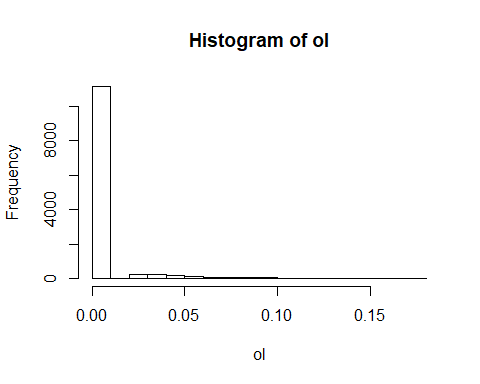
# predict fonksiyonu cancer\_c olarak type="link" şeklinde çalışır  
# fakat type="link" olarak tahmin yapıldığında klasik regresyondaki gibi gözlem değerlerinin tahmini yapiliyor  
# fakat biz siniflandirma yaptigimiz icin bize her bir gozlem icin olasilik degerleri lazim  
# bunun icin type="response" dedik  
head(predict(model\_glm, type = "response"))

## 1 8 10 13 17 22   
## 0.00158 0.00508 0.00139 0.00479 0.00010 0.00016

# 0 ve 1 arasındaki degerleri tahmin ettik  
ol <- predict(model\_glm, type = "response")  
summary(ol)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.002 0.002 0.006 0.004 0.179

#gorsellestirdik  
hist(ol)



# train hatasını hesaplıyoruz  
model\_glm\_pred <- ifelse(predict(model\_glm, type = "response") > 0.1, "Yes","No")  
head(model\_glm\_pred)

## 1 8 10 13 17 22   
## "No" "No" "No" "No" "No" "No"

table(model\_glm\_pred)

## model\_glm\_pred  
## No Yes   
## 12103 25

Siniflandirma Hatasi Tespiti ve Karmasiklik Matrisi

# siniflandirma hatasinin tespiti icin fonksiyon yazdik  
class\_err <- function(gercek, tahmin) {  
   
 mean(gercek != tahmin)  
   
}  
  
#yanlis siniflandirma orani  
class\_err(training$cancer\_c, model\_glm\_pred)

## [1] 0.0071

#dogruluk orani - accuracy  
1-class\_err(training$cancer\_c, model\_glm\_pred)

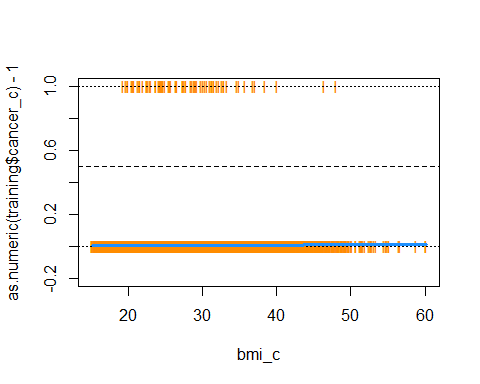
## [1] 0.99

tb <- table(tahmin = model\_glm\_pred,   
 gercek = training$cancer\_c)  
# CI accuracy değerinin güven aralığı  
km <- confusionMatrix(tb, positive = "Yes")  
  
c(km$overall["Accuracy"], km$byClass["Sensitivity"])

## Accuracy Sensitivity   
## 0.99 0.07

## Tahminlerin Gorsellestirilmesi

# bağımlı değişkenin en fazla bağımlı olduğu değişkenle ilişkisini görselleştirdik  
plot(as.numeric(training$cancer\_c)-1 ~ bmi\_c, data = training,  
 col = "darkorange",  
 pch = "I",   
 ylim = c(-0.2, 1))  
  
abline(h = 0, lty = 3)  
abline(h = 1, lty = 3)  
abline(h = 0.5, lty = 2)  
  
model\_glm <- glm(cancer\_c~ bmi\_c,   
 data = training,   
 family = "binomial")  
# görselleştirme için tahminimizi sadece bmi\_c değişkenine göre yapıyoruz  
curve(predict(model\_glm, data.frame(bmi\_c = x), type ="response"),  
 add = TRUE,  
 lwd = 3,  
 col = "dodgerblue")



## ROC Egrisi

model\_glm <- glm(cancer\_c~ .,   
 data = training,   
 family = "binomial")  
  
# bu sefer test verimizi tahmin ettik  
test\_ol <- predict(model\_glm, newdata = test\_x, type = "response")  
#install.packages("pROC")  
library(pROC)

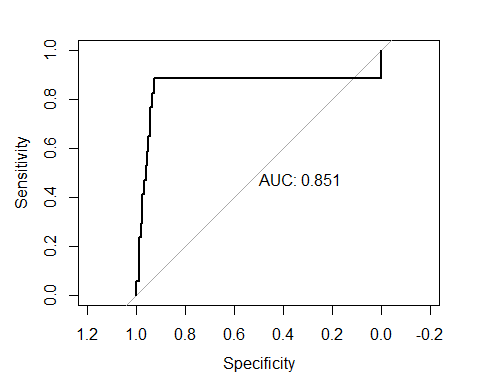
## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following object is masked from 'package:colorspace':  
##   
## coords

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

a <- roc(test\_y ~ test\_ol, plot = TRUE, print.auc = TRUE)



a$auc

## Area under the curve: 0.85

## Model Tuning - Model Optimizasyonu

# metodumuz cross-validation  
#10 tekrardan oluşacak  
ctrl <- trainControl(method = "cv",   
 number = 10,   
 summaryFunction = twoClassSummary,  
 classProbs = TRUE,  
 savePredictions = TRUE)  
  
glm\_tune <- train(train\_x,   
 train\_y,   
 method = "glm",  
 trControl = ctrl)

## Warning in train.default(train\_x, train\_y, method = "glm", trControl =  
## ctrl): The metric "Accuracy" was not in the result set. ROC will be used  
## instead.

glm\_tune

## Generalized Linear Model   
##   
## 12128 samples  
## 10 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 10916, 10915, 10915, 10914, 10916, 10915, ...   
## Resampling results:  
##   
## ROC Sens Spec  
## 0.86 1 0

head(glm\_tune$pred,10)

## pred obs No Yes rowIndex parameter Resample  
## 1 No No 1.00 0.00267 17 none Fold01  
## 2 No No 1.00 0.00432 31 none Fold01  
## 3 No No 1.00 0.00377 71 none Fold01  
## 4 No No 1.00 0.00120 80 none Fold01  
## 5 No No 0.99 0.00549 88 none Fold01  
## 6 No No 1.00 0.00013 128 none Fold01  
## 7 No No 0.99 0.00521 129 none Fold01  
## 8 No No 0.99 0.00534 155 none Fold01  
## 9 No No 1.00 0.00237 156 none Fold01  
## 10 No Yes 0.93 0.06618 164 none Fold01

head(glm\_tune$pred$Yes)

## [1] 0.00267 0.00432 0.00377 0.00120 0.00549 0.00013

# accuracy değerine ulaşabildik  
defaultSummary(data.frame(obs = test\_y,   
 pred = predict(glm\_tune, test\_x)))

## Accuracy Kappa   
## 0.99 0.00

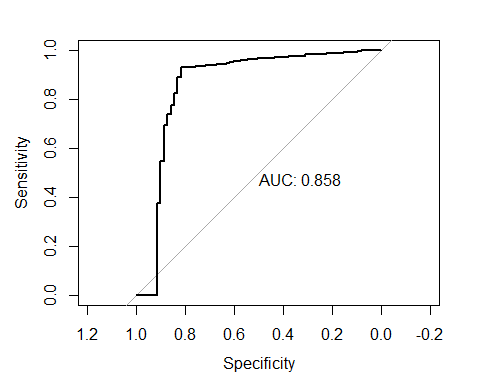
# burada görüldüğü gibi optimize edilmeye çalışılan model tüm değerlere no dedi ve %94 doğruluk oranına düştü  
confusionMatrix(data = predict(glm\_tune, train\_x),  
 reference = train\_y, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 12057 71  
## Yes 0 0  
##   
## Accuracy : 0.994   
## 95% CI : (0.993, 0.995)  
## No Information Rate : 0.994   
## P-Value [Acc > NIR] : 0.532   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.00000   
## Specificity : 1.00000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.99415   
## Prevalence : 0.00585   
## Detection Rate : 0.00000   
## Detection Prevalence : 0.00000   
## Balanced Accuracy : 0.50000   
##   
## 'Positive' Class : Yes   
##

confusionMatrix(data = predict(glm\_tune, test\_x),  
 reference = test\_y, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 3014 17  
## Yes 0 0  
##   
## Accuracy : 0.994   
## 95% CI : (0.991, 0.997)  
## No Information Rate : 0.994   
## P-Value [Acc > NIR] : 0.564023   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : 0.000104   
##   
## Sensitivity : 0.00000   
## Specificity : 1.00000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.99439   
## Prevalence : 0.00561   
## Detection Rate : 0.00000   
## Detection Prevalence : 0.00000   
## Balanced Accuracy : 0.50000   
##   
## 'Positive' Class : Yes   
##

roc(glm\_tune$pred$obs,  
 glm\_tune$pred$Yes,  
 levels = rev(levels(glm\_tune$pred$obs)),  
 plot = TRUE, print.auc = TRUE)



##   
## Call:  
## roc.default(response = glm\_tune$pred$obs, predictor = glm\_tune$pred$Yes, levels = rev(levels(glm\_tune$pred$obs)), plot = TRUE, print.auc = TRUE)  
##   
## Data: glm\_tune$pred$Yes in 71 controls (glm\_tune$pred$obs Yes) > 12057 cases (glm\_tune$pred$obs No).  
## Area under the curve: 0.86

# KNN

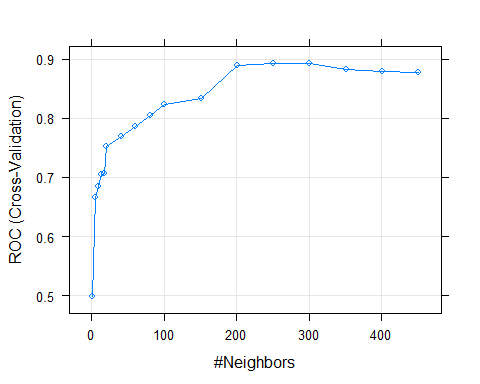
## Model

# knn kategorik degiskenlerle calismaktadir bunun icin ya donusum yapilir ya da o kategorik degisken verisetinden cikarilir.  
#install.packages("caret")  
library(caret)  
train\_indeks <- createDataPartition(dataframe$cancer\_c, p = 0.8, list = FALSE, times = 1)  
  
train <- dataframe[train\_indeks,]  
test <- dataframe[-train\_indeks,]  
  
train\_x <- train %>% dplyr::select(-cancer\_c)  
train\_y <- train$cancer\_c  
  
test\_x <- test %>% dplyr::select(-cancer\_c)  
test\_y <- test$cancer\_c  
  
training <- data.frame(train\_x, cancer\_c = train\_y)  
  
knn\_train <- train  
knn\_test <- test  
  
knn\_train <- knn\_train %>% select(-cancer\_c)  
knn\_test <- knn\_test %>% select(-cancer\_c)  
# knn fonksiyonu lojistik regresyon fonksiyonundan farkli degerlerle calisir  
#install.packages("FNN")  
library("FNN")  
knn\_fit <- knn(train = knn\_train, test = knn\_test, cl = train\_y, k = 3)  
summary(knn\_fit)

## No Yes   
## 3030 1

## Model Tuning - Model Optimizasyonu

ctrl <- trainControl(method = "cv",   
 number = 10,   
 summaryFunction = twoClassSummary,  
 classProbs = TRUE,  
 savePredictions = TRUE)  
# bir arama vektoru olusturuldu  
knn\_grid <- data.frame(k = c(4\*(0:5)+1, 20\*(1:5)+1, 50\*(2:9)+1))  
# 451 komsuluk degerinin en iyisi oldugu soylenmis  
knn\_tune <- train(knn\_test, test\_y,  
 method = "knn",  
 metric = "ROC",  
 preProc = c("center", "scale"),  
 trControl = ctrl,  
 tuneGrid = knn\_grid)  
  
plot(knn\_tune)



knn\_tune$bestTune

## k  
## 13 251

# en iyi k degeri secildi ve knn optimize edildi  
#confusionMatrix(predict(knn\_tune, knn\_test), knn\_test, positive = "Yes")