**Opgave 3 Analytics 2019**

Naam student: *graag hier invullen*

Data voor deze opgave: **telcotrain.csv** en **telcotest.csv**. Documentatie van de data: **telco.docx**

Je mag voor het maken van confusion matrices de functie die je in de vorige opgave **(bij opgave 2 vraag 1c)** zelf hebt gemaakt, gebruiken om gemakkelijker confusion matrices en dergelijke te berekenen. (Gebruik source("bestand.r")om een script te kunnen laden).

**Inleveren**

Inleveren UITSLUITEND via Blackboard. Uiterste datum van inleveren: 17-3-2019.

Inleveren UITSLUITEND in formaat .PDF, .DOC of .DOCX.

Als je met RStudio werkt kan je de .Rmd file als basis gebruiken. Anders mag je de resultaten in dit document opnemen. Vergeet niet je naam in het document te zetten.

**Inleiding**

In Opgave 2 heb je deze data ook gebruikt, voor het maken van een logistic regressie-model. Zie het document “Opgave 2 Analytics 2015” voor de details.

Deze opgave ging over een belangrijke vraag uit de Telecomindustrie: hoe houd ik mijn klanten?

In Opgave 3 ga je daarmee verder. Het gemaakte logistic regressiemodel wordt vergeleken met een CART-tree model. Ook ga je Crossvalidation toepassen om de optimale boom te bepalen.

De vraag is dus: hoe groot is de kans op Churn (vertrek) en welke variabelen spelen daarbij wel en niet een rol. Dit gaan we nu doen met de methode “CART trees”.

**Vraag 1**

De opgave begint met een herhaling van Opgave 2. (Zie vraag 1 van Opgave 2)   
Maak opnieuw het **optimale** logistic regressie-model zoals eerder bepaald. Dat wil zeggen, met alleen nog significante variabelen.

**1a)** Laad de data (**telcotrain.csv**), verwijder de eerste en derde kolom en bouw het eerder (bij Opgave 2 (vraag 1b)) bepaalde **optimale** logistic regression model. Geef de summary van het model.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer.

> setwd("C:/Users/wiebe/Documents/analytics/opgaven\_data")

> getwd

> telcotrain <- read.csv(file ='telcotrain.csv')

> View(telcotrain)

> telcotrain$X <- NULL

> telcotrain$ZIP <- NULL

> model=glm(Churn~IP+VMP+NVM+TDCH+TEM+TNM+TIC+TICH+CSC,data=telcotrain,family=binomial)

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TEM + TNM + TIC +

TICH + CSC, family = binomial, data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0729 -0.5128 -0.3379 -0.1951 3.2768

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -8.325582 0.602807 -13.811 < 2e-16 \*\*\*

IPYes 1.995750 0.168834 11.821 < 2e-16 \*\*\*

VMPYes -2.294154 0.654017 -3.508 0.000452 \*\*\*

NVM 0.046334 0.020097 2.305 0.021140 \*

TDCH 0.079038 0.007390 10.695 < 2e-16 \*\*\*

TEM 0.006881 0.001335 5.154 2.55e-07 \*\*\*

TNM 0.004644 0.001281 3.626 0.000288 \*\*\*

TIC -0.083499 0.029185 -2.861 0.004223 \*\*

TICH 0.327923 0.086630 3.785 0.000154 \*\*\*

CSC 0.513880 0.044928 11.438 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2067.9 on 2499 degrees of freedom

Residual deviance: 1617.7 on 2490 degrees of freedom

AIC: 1637.7

Number of Fisher Scoring iterations: 6

**1b)** Gebruikdit model om de waarden van Churn in de **testset** **(telcotest.csv)** te voorspellen. Bij Opgave 2 (vraag 2) heb je ook de **optimale drempelwaarde** voor dit model bepaald.Geef voor *die* drempelwaarde deConfusion Matrix, de accuracy, de specificity en de sensitivity.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer. Vermeld de gebruikte drempelwaarde.

> setwd("C:/Users/wiebe/Documents/analytics/opgaven\_data")

> getwd

> telcotest <- read.csv(file ='telcotest.csv')

> View(telcotest)

> telcotest$X <- NULL

> telcotest$ZIP <- NULL

model=glm(Churn~IP+VMP+NVM+TDCH+TEM+TNM+TIC+TICH+CSC,data=telcotest,family=binomial)

predictTest =predict(model,type="response", newdata=telcotest)

conf2 <- function(pred, afh, drempel) {

tab = table(afh,pred> drempel)

print(tab)

print('accuracy ')

print((tab[4] + tab[1])/(tab[4] + tab[3] + tab[2] + tab[1]) \* 100)

print('sensitivity')

print(tab[4:4]/(tab[4:4]+tab[2:2]) \* 100)

print('specificity ')

print(tab[1:1]/(tab[1:1]+tab[3:3]) \* 100)

}

> conf2(predictTest,telcotest$Churn,0.2)

afh FALSE TRUE

NO 598 114

YES 42 79

[1] "accuracy "

[1] 81.27251

[1] "sensitivity"

[1] 65.28926

[1] "specificity "

[1] 83.98876

**1c)** Maak ook een ROC-curve van het model en bepaal de Area Under Curve (AUC). Voor ROC-curves heb je de package ROCR nodig.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer, inclusief de plot en de AUC-waarde.

library(ROCR)

ROCpred = prediction(predictionmodel,telcotrain$Churn)

ROCperf = performance(ROCpred, "tpr", "fpr")

plot(ROCperf,colorize=TRUE,print.cutoffs.at=seq(0,1,0.1))

abline(0,1)

Afbeelding met kaart, tekst

Automatisch gegenereerde beschrijving

AUC=as.numeric(performance(ROCpred,"auc")@y.values)

AUC

[1] 0.8216527

**Vraag 2**

Maak nu een CART tree model voor dezelfde **telcotrain** data. Voor CART-trees heb je de packages rpart en rpart.plot nodig. De methodiek is uitgelegd op het college en wordt behandeld in video 4 van de serie Judge Jury and Classifier.

NB: Ook bij deze opgave moet je eerst de eerste en derde kolom uit de data verwijderen.

**2a)** Maak op basis van de trainingsdata een CART-tree voor de kans op Churn (afhankelijke variabele) als functie van de overige (onafhankelijke) variabelen. Gebruik een minbucket van 25. Maak een plot van de tree.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer en de plot.

install.packages('rpart')

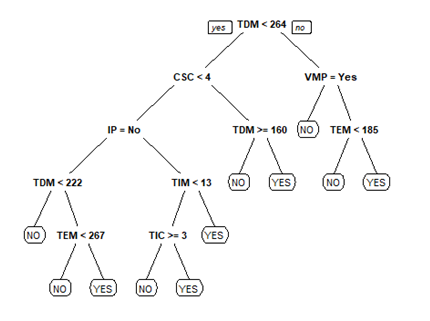
install.packages('rpart.plot')

> library(rpart)

> library(rpart.plot)

tree=rpart(Churn~.,data=telcotrain,method="class",control=rpart.control(minbucket=25))

prp(tree)



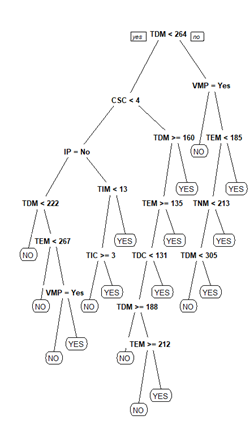
**2b)** Bereken ook de tree voor een minbucket van 5 en een voor een minbucket van 100. Druk ook daarvan de tree af.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer en de plots.

tree2=rpart(Churn~.,data=telcotrain,method=”class”,control=rpart.control(minbucket=5))

prp(tree2)



> tree3=rpart(Churn~.,data=telcotrain,method="class",control=rpart.control(minbucket=100))

> prp(tree3)

Afbeelding met kaart

Automatisch gegenereerde beschrijving

## Vraag 3

Valideer het CART-model dat bij Vraag 2a) gemaakt is. Gebruik daarvoor dit model om de waarden van Churn in de **testset** **(telcotest.csv)** te voorspellen.

NB: Ook bij deze opgave moet je eerst de eerste en derde kolom uit de data verwijderen.

**3a)** Geef de Confusion Matrix, de accuracy, de specificity en de sensitivity.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer.

conf2 <- function(pred, afh, drempel) {

tab = table(afh,pred> drempel)

print(tab)

print('accuracy ')

print((tab[4] + tab[1])/(tab[4] + tab[3] + tab[2] + tab[1]) \* 100)

print('sensitivity')

print(tab[4:4]/(tab[4:4]+tab[2:2]) \* 100)

print('specificity ')

print(tab[1:1]/(tab[1:1]+tab[3:3]) \* 100)

}

predictTree2=predict(tree,newdata = telcotest)

> conf2(predictTree2[,2],telcotest$Churn,0.1)

afh FALSE TRUE

NO 658 54

YES 29 92

[1] "accuracy "

[1] 90.03601

[1] "sensitivity"

[1] 76.03306

[1] "specificity "

[1] 92.41573

**3b)** Maak ook een ROC-curve van het model en bepaal de Area Under Curve (AUC). Voor ROC-curves heb je de package ROCR nodig.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer en de plots.

ROCpredict2 = prediction(predictTree2[,2],telcotest$Churn)

ROCtree2 = performance(ROCpredict2,"tpr","fpr")

plot(ROCtree2,colorize=TRUE,print.cutoffs.at=seq(0,1,0.1))

Afbeelding met schermafbeelding, kaart

Automatisch gegenereerde beschrijving

> AUC=as.numeric(performance(ROCpredict2,"auc")@y.values)

> AUC

[1] 0.9068391

**3c)** Vergelijk de resultaten van dit CART-tree model met het logistic regression model uit vraag 1. Welke is het beste en waarom?

**Gevraagde antwoorden:**

Beantwoord de vraag.

De area under curve is bij de cart hoger, dus die is beter. En uiteraard de accuracy is ook hoger.

## Vraag 4

Gebruik Cross-Validation om de optimale CART-tree voor deze data te kunnen bepalen. Hiervoor heb je de packages caret en e1071 nodig. De methodiek is uitgelegd op het college en wordt behandeld in video 6 van de serie Judge Jury and Classifier.

**4a)** Bepaal door middel van Cross-Validation de optimale waarde voor de complexity-parameter (cp).

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer.

|  |
| --- |
| > library(caTools)  > set.seed(9022)  > library(caret)  > library(e1071) |
|  |
| |  | | --- | | > | |

**4b)** Maak een plot van de Cross-Validation data. Valt je nog iets op?

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de plot. Beantwoord de vraag.

**4c)** Bereken de CART-tree met de berekende cp-waarde. Maak een plot van de tree.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer en de plots.

**4d)** Gebruik ook dit model om de waarden van Churn in de **testset** **(telcotest.csv)** te voorspellen. Geef weer de Confusion Matrix, de accuracy, de specificity en de sensitivity.

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer en de plots.

**4e)** Vergelijk dit ‘optimale’ model met het ‘ standaard’model dat je bij vraag 3 hebt gemaakt. Wat is je conclusie?

**Gevraagde antwoorden:**

Beantwoord de vraag.

Ja model is beter accuracy en sensitivity ligt hoger en de curve is beter.