# Opgave 2 Analytics 2019

Naam student: *graag hier invullen*

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

## Inleveren

Inleveren UITSLUITEND via Blackboard. Uiterste datum van inleveren: 10-3-2019.

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

De opgave gaat over een belangrijke vraag uit de Telecomindustrie: hoe houd ik mijn klanten?

In het databestand staan een aantal gegevens over klanten, zoals het aantal telefoontjes per dagdeel, of ze voicemail hebben, en nog zo het een en ander. De belangrijke variabele is de “Churn”, dat wil zeggen of de klanten bij deze provider zijn vertrokken in de periode waarin deze data is verzameld. (Churn=YES betekent de klant is vertrokken).

De provider wil graag kunnen voorspellen welke klanten waarschijnlijk op het punt staan te vertrekken, zodat hij deze kan benaderen met een goed aanbod.

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

De werkwijze is grotendeels hetzelfde als bij de videoreeks “Modeling the Expert”.

## Vraag 1

Laad de data in. De data bevat twee kolommen die je beter kan verwijderen (de eerste en de derde).

Maak daarna een Logistic Regression model voor het bepalen van de kans op Churn:

**1a)** Laad de data (**telcotrain.csv**), verwijder de variabelen in kolom 1 en kolom 3, en bouw dan een logistic regression model voor de kans dat de Churn variabele de waarde YES heeft (afhankelijke variabele: Churn), afhankelijk van alle andere variabelen. Druk de summary af van het model om te bepalen welke variabelen significant zijn in het model.

**Gevraagde antwoorden:**

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

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

> getwd

function ()

.Internal(getwd())

<bytecode: 0x000000000e174f00>

<environment: namespace:base>

> read.csv("telcotrain.csv")

X ACL ZIP IP VMP NVM TDM TDC TDCH TEM TEC TECH TNM TNC TNCH TIM TIC

1 2 107 415 No Yes 26 161.6 123 27.47 195.5 103 16.62 254.4 103 11.45 13.7 3

2 3 137 415 No No 0 243.4 114 41.38 121.2 110 10.30 162.6 104 7.32 12.2 5

3 4 84 408 Yes No 0 299.4 71 50.90 61.9 88 5.26 196.9 89 8.86 6.6 7

4 5 75 415 Yes No 0 166.7 113 28.34 148.3 122 12.61 186.9 121 8.41 10.1 3

5 6 118 510 Yes No 0 223.4 98 37.98 220.6 101 18.75 203.9 118 9.18 6.3 6

6 7 121 510 No Yes 24 218.2 88 37.09 348.5 108 29.62 212.6 118 9.57 7.5 7

7 9 117 408 No No 0 184.5 97 31.37 351.6 80 29.89 215.8 90 9.71 8.7 4

8 11 65 415 No No 0 129.1 137 21.95 228.5 83 19.42 208.8 111 9.40 12.7 6

9 12 74 415 No No 0 187.7 127 31.91 163.4 148 13.89 196.0 94 8.82 9.1 5

10 14 95 510 No No 0 156.6 88 26.62 247.6 75 21.05 192.3 115 8.65 12.3 5

11 15 62 415 No No 0 120.7 70 20.52 307.2 76 26.11 203.0 99 9.14 13.1 6

12 16 161 415 No No 0 332.9 67 56.59 317.8 97 27.01 160.6 128 7.23 5.4 9

13 17 85 408 No Yes 27 196.4 139 33.39 280.9 90 23.88 89.3 75 4.02 13.8 4

14 18 93 510 No No 0 190.7 114 32.42 218.2 111 18.55 129.6 121 5.83 8.1 3

15 20 73 415 No No 0 224.4 90 38.15 159.5 88 13.56 192.8 74 8.68 13.0 2

16 21 147 415 No No 0 155.1 117 26.37 239.7 93 20.37 208.8 133 9.40 10.6 4

17 22 77 408 No No 0 62.4 89 10.61 169.9 121 14.44 209.6 64 9.43 5.7 6

18 23 130 415 No No 0 183.0 112 31.11 72.9 99 6.20 181.8 78 8.18 9.5 19

19 24 111 415 No No 0 110.4 103 18.77 137.3 102 11.67 189.6 105 8.53 7.7 6

20 25 132 510 No No 0 81.1 86 13.79 245.2 72 20.84 237.0 115 10.67 10.3 2

21 27 57 408 No Yes 39 213.0 115 36.21 191.1 112 16.24 182.7 115 8.22 9.5 3

22 28 54 408 No No 0 134.3 73 22.83 155.5 100 13.22 102.1 68 4.59 14.7 4

23 30 49 510 No No 0 119.3 117 20.28 215.1 109 18.28 178.7 90 8.04 11.1 1

24 31 142 415 No No 0 84.8 95 14.42 136.7 63 11.62 250.5 148 11.27 14.2 6

25 32 75 510 No No 0 226.1 105 38.44 201.5 107 17.13 246.2 98 11.08 10.3 5

26 33 172 408 No No 0 212.0 121 36.04 31.2 115 2.65 293.3 78 13.20 12.6 10

27 35 57 408 No Yes 25 176.8 94 30.06 195.0 75 16.58 213.5 116 9.61 8.3 4

28 39 136 415 Yes Yes 33 203.9 106 34.66 187.6 99 15.95 101.7 107 4.58 10.5 6

29 40 149 408 No No 0 140.4 94 23.87 271.8 92 23.10 188.3 108 8.47 11.1 9

30 41 98 408 No No 0 126.3 102 21.47 166.8 85 14.18 187.8 135 8.45 9.4 2

31 42 135 408 Yes Yes 41 173.1 85 29.43 203.9 107 17.33 122.2 78 5.50 14.6 15

32 43 34 510 No No 0 124.8 82 21.22 282.2 98 23.99 311.5 78 14.02 10.0 4

33 44 160 415 No No 0 85.8 77 14.59 165.3 110 14.05 178.5 92 8.03 9.2 4

34 45 64 510 No No 0 154.0 67 26.18 225.8 118 19.19 265.3 86 11.94 3.5 3

35 46 59 408 No Yes 28 120.9 97 20.55 213.0 92 18.11 163.1 116 7.34 8.5 5

36 47 65 415 No No 0 211.3 120 35.92 162.6 122 13.82 134.7 118 6.06 13.2 5

37 48 142 408 No No 0 187.0 133 31.79 134.6 74 11.44 242.2 127 10.90 7.4 5

38 49 119 415 No No 0 159.1 114 27.05 231.3 117 19.66 143.2 91 6.44 8.8 3

39 51 52 408 No No 0 191.9 108 32.62 269.8 96 22.93 236.8 87 10.66 7.8 5

40 52 60 408 No No 0 220.6 57 37.50 211.1 115 17.94 249.0 129 11.21 6.8 3

41 53 10 408 No No 0 186.1 112 31.64 190.2 66 16.17 282.8 57 12.73 11.4 6

42 54 96 415 No No 0 160.2 117 27.23 267.5 67 22.74 228.5 68 10.28 9.3 5

43 55 87 415 No No 0 151.0 83 25.67 219.7 116 18.67 203.9 127 9.18 9.7 3

44 56 81 408 No No 0 175.5 67 29.84 249.3 85 21.19 270.2 98 12.16 10.2 3

45 57 141 415 No No 0 126.9 98 21.57 180.0 62 15.30 140.8 128 6.34 8.0 2

46 58 121 408 No Yes 30 198.4 129 33.73 75.3 77 6.40 181.2 77 8.15 5.8 3

47 60 125 408 No No 0 229.3 103 38.98 177.4 126 15.08 189.3 95 8.52 12.0 8

48 62 116 415 No Yes 34 268.6 83 45.66 178.2 142 15.15 166.3 106 7.48 11.6 3

49 64 149 408 No Yes 28 180.7 92 30.72 187.8 64 15.96 265.5 53 11.95 12.6 3

50 65 38 408 No No 0 131.2 98 22.30 162.9 97 13.85 159.0 106 7.15 8.2 6

TICH CSC Churn

1 3.70 1 NO

2 3.29 0 NO

3 1.78 2 NO

4 2.73 3 NO

5 1.70 0 NO

6 2.03 3 NO

7 2.35 1 NO

8 3.43 4 YES

9 2.46 0 NO

10 3.32 3 NO

11 3.54 4 NO

12 1.46 4 YES

13 3.73 1 NO

14 2.19 3 NO

15 3.51 1 NO

16 2.86 0 NO

17 1.54 5 YES

18 2.57 0 NO

19 2.08 2 NO

20 2.78 0 NO

21 2.57 0 NO

22 3.97 3 NO

23 3.00 1 NO

24 3.83 2 NO

25 2.78 1 NO

26 3.40 3 NO

27 2.24 0 NO

28 2.84 3 NO

29 3.00 1 NO

30 2.54 3 NO

31 3.94 0 YES

32 2.70 2 NO

33 2.48 3 NO

34 0.95 1 NO

35 2.30 2 NO

36 3.56 3 NO

37 2.00 2 NO

38 2.38 5 YES

39 2.11 3 NO

40 1.84 1 NO

41 3.08 2 NO

42 2.51 2 NO

43 2.62 5 YES

44 2.75 1 NO

45 2.16 1 NO

46 1.57 3 YES

47 3.24 1 NO

48 3.13 2 NO

49 3.40 3 NO

50 2.21 2 NO

[ reached 'max' / getOption("max.print") -- omitted 2450 rows ]

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

> View(telcotrain)

> telcotrain$X <- NULL

> telcotrain$ZIP <- NULL

> model=glm(Churn~.,data=telcotrain,family=binomial)

> summary(model)

Call:

glm(formula = Churn ~ ., family = binomial, data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0209 -0.5124 -0.3412 -0.1915 3.3396

Coefficients:

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

(Intercept) -9.033e+00 8.498e-01 -10.629 < 2e-16 \*\*\*

ACL 1.840e-03 1.637e-03 1.124 0.261033

IPYes 1.986e+00 1.691e-01 11.744 < 2e-16 \*\*\*

VMPYes -2.318e+00 6.558e-01 -3.535 0.000408 \*\*\*

NVM 4.687e-02 2.014e-02 2.328 0.019938 \*

TDM -1.945e+00 3.758e+00 -0.518 0.604667

TDC 4.308e-03 3.168e-03 1.360 0.173891

TDCH 1.152e+01 2.210e+01 0.521 0.602184

TEM 1.481e+00 1.889e+00 0.784 0.433141

TEC 2.387e-04 3.266e-03 0.073 0.941729

TECH -1.734e+01 2.223e+01 -0.780 0.435305

TNM 7.087e-01 1.014e+00 0.699 0.484742

TNC 4.337e-04 3.263e-03 0.133 0.894275

TNCH -1.564e+01 2.254e+01 -0.694 0.487598

TIM -9.422e-01 6.210e+00 -0.152 0.879399

TIC -8.282e-02 2.939e-02 -2.818 0.004834 \*\*

TICH 3.816e+00 2.300e+01 0.166 0.868203

CSC 5.170e-01 4.516e-02 11.449 < 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: 1613.3 on 2482 degrees of freedom

AIC: 1649.3

Number of Fisher Scoring iterations: 6

**1b)** Verwijder **één voor één** de niet-significante variabelen uit het model, net zolang tot alleen significante variabelen over zijn. Noteer steeds wat de AIC-waarde is (als het goed is wordt deze steeds kleiner).

**Gevraagde antwoorden:**

Geef alle gebruikte R-commando’s. Geef ook de R-uitvoer. Geef de AIC-waarde voor elke tussenstap. Geef aan het eind ook de summary van het model.

**model=glm(Churn~ACL+IP+VMP+NVM+TDM+TDC+TDCH+TEM+TECH+TNM+TNC+TNCH+TIM+TIC+TICH+CSC,data=telcotrain,family=binomial)**

> summary(model)

Call:

glm(formula = Churn ~ ACL + IP + VMP + NVM + TDM + TDC + TDCH +

TEM + TECH + TNM + TNC + TNCH + TIM + TIC + TICH + CSC, family = binomial,

data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0207 -0.5124 -0.3413 -0.1916 3.3385

Coefficients:

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

(Intercept) -9.009e+00 7.845e-01 -11.484 < 2e-16 \*\*\*

ACL 1.845e-03 1.635e-03 1.128 0.259223

IPYes 1.987e+00 1.691e-01 11.745 < 2e-16 \*\*\*

VMPYes -2.318e+00 6.557e-01 -3.536 0.000407 \*\*\*

NVM 4.688e-02 2.014e-02 2.328 0.019913 \*

TDM -1.944e+00 3.758e+00 -0.517 0.604949

TDC 4.305e-03 3.168e-03 1.359 0.174160

TDCH 1.151e+01 2.210e+01 0.521 0.602466

TEM 1.484e+00 1.889e+00 0.786 0.431984

TECH -1.738e+01 2.222e+01 -0.782 0.434145

TNM 7.087e-01 1.014e+00 0.699 0.484673

TNC 4.318e-04 3.263e-03 0.132 0.894727

TNCH -1.565e+01 2.254e+01 -0.694 0.487529

TIM -9.415e-01 6.210e+00 -0.152 0.879502

TIC -8.282e-02 2.939e-02 -2.818 0.004835 \*\*

TICH 3.814e+00 2.300e+01 0.166 0.868305

CSC 5.170e-01 4.515e-02 11.450 < 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: 1613.3 on 2483 degrees of freedom

AIC: 1647.3

Number of Fisher Scoring iterations: 6

> model=glm(Churn~ACL+IP+VMP+NVM+TDM+TDC+TDCH+TEM+TECH+TNM+TNCH+TIM+TIC+TICH+CSC,data=telcotrain,family=binomial)

> summary(model)

Call:

glm(formula = Churn ~ ACL + IP + VMP + NVM + TDM + TDC + TDCH +

TEM + TECH + TNM + TNCH + TIM + TIC + TICH + CSC, family = binomial,

data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0228 -0.5130 -0.3412 -0.1918 3.3395

Coefficients:

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

(Intercept) -8.965459 0.712718 -12.579 < 2e-16 \*\*\*

ACL 0.001842 0.001635 1.127 0.259836

IPYes 1.987024 0.169095 11.751 < 2e-16 \*\*\*

VMPYes -2.313686 0.654697 -3.534 0.000409 \*\*\*

NVM 0.046738 0.020107 2.324 0.020100 \*

TDM -1.953707 3.756621 -0.520 0.603015

TDC 0.004299 0.003167 1.357 0.174745

TDCH 11.571217 22.097973 0.524 0.600534

TEM 1.481619 1.888524 0.785 0.432724

TECH -17.349007 22.217918 -0.781 0.434887

TNM 0.712155 1.013798 0.702 0.482391

TNCH -15.722191 22.528060 -0.698 0.485244

TIM -0.935689 6.210460 -0.151 0.880241

TIC -0.082734 0.029387 -2.815 0.004872 \*\*

TICH 3.792183 23.000824 0.165 0.869045

CSC 0.516960 0.045144 11.451 < 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: 1613.3 on 2484 degrees of freedom

AIC: 1645.3

Number of Fisher Scoring iterations: 6

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

> summary(model)

Call:

glm(formula = Churn ~ ACL + IP + VMP + NVM + TDM + TDC + TDCH +

TEM + TECH + TNM + TNCH + TIC + TICH + CSC, family = binomial,

data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0261 -0.5115 -0.3410 -0.1922 3.3437

Coefficients:

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

(Intercept) -8.966774 0.712753 -12.580 < 2e-16 \*\*\*

ACL 0.001839 0.001635 1.125 0.260507

IPYes 1.986213 0.169008 11.752 < 2e-16 \*\*\*

VMPYes -2.312769 0.654798 -3.532 0.000412 \*\*\*

NVM 0.046724 0.020111 2.323 0.020165 \*

TDM -1.957981 3.756126 -0.521 0.602174

TDC 0.004315 0.003165 1.363 0.172845

TDCH 11.596362 22.095060 0.525 0.599695

TEM 1.481904 1.888611 0.785 0.432657

TECH -17.352404 22.218948 -0.781 0.434818

TNM 0.709461 1.013651 0.700 0.483986

TNCH -15.662216 22.524793 -0.695 0.486847

TIC -0.082483 0.029333 -2.812 0.004925 \*\*

TICH 0.326831 0.086739 3.768 0.000165 \*\*\*

CSC 0.516836 0.045136 11.451 < 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: 1613.3 on 2485 degrees of freedom

AIC: 1643.3

Number of Fisher Scoring iterations: 6

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

> summary(model)

Call:

glm(formula = Churn ~ ACL + IP + VMP + NVM + TDC + TDCH + TEM +

TECH + TNM + TNCH + TIC + TICH + CSC, family = binomial,

data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0182 -0.5100 -0.3411 -0.1926 3.3299

Coefficients:

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

(Intercept) -8.956436 0.712212 -12.576 < 2e-16 \*\*\*

ACL 0.001822 0.001634 1.115 0.264792

IPYes 1.987970 0.168949 11.767 < 2e-16 \*\*\*

VMPYes -2.308616 0.654650 -3.526 0.000421 \*\*\*

NVM 0.046624 0.020106 2.319 0.020400 \*

TDC 0.004329 0.003165 1.368 0.171316

TDCH 0.078734 0.007400 10.640 < 2e-16 \*\*\*

TEM 1.467645 1.888346 0.777 0.437034

TECH -17.184718 22.215836 -0.774 0.439206

TNM 0.714790 1.013104 0.706 0.480471

TNCH -15.780835 22.512619 -0.701 0.483317

TIC -0.081834 0.029286 -2.794 0.005201 \*\*

TICH 0.326740 0.086723 3.768 0.000165 \*\*\*

CSC 0.516322 0.045123 11.443 < 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: 1613.6 on 2486 degrees of freedom

AIC: 1641.6

Number of Fisher Scoring iterations: 6

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

> summary(model)

Call:

glm(formula = Churn ~ ACL + IP + VMP + NVM + TDC + TDCH + TEM +

TECH + TNM + TIC + TICH + CSC, family = binomial, data = telcotrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0350 -0.5108 -0.3405 -0.1944 3.3180

Coefficients:

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

(Intercept) -8.953011 0.711867 -12.577 < 2e-16 \*\*\*

ACL 0.001806 0.001633 1.106 0.268727

IPYes 1.987860 0.168937 11.767 < 2e-16 \*\*\*

VMPYes -2.306958 0.654277 -3.526 0.000422 \*\*\*

NVM 0.046642 0.020098 2.321 0.020302 \*

TDC 0.004381 0.003165 1.384 0.166244

TDCH 0.078799 0.007399 10.649 < 2e-16 \*\*\*

TEM 1.464018 1.887303 0.776 0.437915

TECH -17.141888 22.203553 -0.772 0.440095

TNM 0.004629 0.001282 3.611 0.000305 \*\*\*

TIC -0.082607 0.029256 -2.824 0.004749 \*\*

TICH 0.325843 0.086666 3.760 0.000170 \*\*\*

CSC 0.514921 0.045039 11.433 < 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: 1614.1 on 2487 degrees of freedom

AIC: 1640.1

Number of Fisher Scoring iterations: 6

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

> summary(model)

Call:

glm(formula = Churn ~ ACL + IP + VMP + NVM + TDC + TDCH + TEM +

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

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0240 -0.5122 -0.3414 -0.1940 3.2999

Coefficients:

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

(Intercept) -8.962803 0.711682 -12.594 < 2e-16 \*\*\*

ACL 0.001773 0.001634 1.085 0.277823

IPYes 1.986478 0.168866 11.764 < 2e-16 \*\*\*

VMPYes -2.291753 0.652975 -3.510 0.000449 \*\*\*

NVM 0.046038 0.020060 2.295 0.021735 \*

TDC 0.004347 0.003162 1.375 0.169204

TDCH 0.079021 0.007397 10.683 < 2e-16 \*\*\*

TEM 0.006959 0.001335 5.213 1.86e-07 \*\*\*

TNM 0.004657 0.001282 3.632 0.000281 \*\*\*

TIC -0.082410 0.029247 -2.818 0.004837 \*\*

TICH 0.325791 0.086672 3.759 0.000171 \*\*\*

CSC 0.515268 0.045035 11.441 < 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: 1614.7 on 2488 degrees of freedom

AIC: 1638.7

Number of Fisher Scoring iterations: 6

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

> summary(model)

Call:

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

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

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0195 -0.5130 -0.3407 -0.1935 3.2996

Coefficients:

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

(Intercept) -8.765977 0.686331 -12.772 < 2e-16 \*\*\*

IPYes 1.990753 0.168826 11.792 < 2e-16 \*\*\*

VMPYes -2.292990 0.654679 -3.502 0.000461 \*\*\*

NVM 0.046191 0.020118 2.296 0.021676 \*

TDC 0.004348 0.003161 1.375 0.169017

TDCH 0.079032 0.007394 10.689 < 2e-16 \*\*\*

TEM 0.006901 0.001334 5.175 2.28e-07 \*\*\*

TNM 0.004620 0.001282 3.604 0.000313 \*\*\*

TIC -0.081902 0.029222 -2.803 0.005067 \*\*

TICH 0.325043 0.086682 3.750 0.000177 \*\*\*

CSC 0.515576 0.045019 11.452 < 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: 1615.8 on 2489 degrees of freedom

AIC: 1637.8

Number of Fisher Scoring iterations: 6

> 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

**1c)** In het vervolg moet je een aantal malen de Confusion Matrix en kengetallen (accuracy, specificity, sensitivity) bepalen, voor verschillende drempelwaarden. Schrijf een r-script met daarin een functie waarmee je deze zaken automatisch (en correct) uitrekent, waarna je ze kan afdrukken. Als input geef je het model, de (volledige) naam van de afhankelijke variabele en de drempelwaarde. De functie berekent dan de juiste confusion matrix en de waarden voor accuracy, specificity, sensitivity. Zorg dat deze worden afgedrukt op de R-console.

Confusion(test$TestYearHCD,predictTest>0.5)

Confusion(data$column,predictTest>threshold)

Confusion(arg1,arg2)

calculateConfusionMatrix(pred, relative = FALSE, sums = FALSE,

set = "both")

# S3 method for ConfusionMatrix

print(x, both = TRUE, digits = 2, ...)

class\_prediction <-

ifelse(probability\_prediction > 0.50,

"positive\_class",

"negative\_class"

)

confusionMatrix(predicted, actual)

pred <- ifelse(probability > threshold, "M", "R")

confusionMatrix(pred, actual)

allinds = 1:150

train = sample(allinds, 75)

test = setdiff(allinds, train)

mod = train("classif.lda", iris.task, subset = train)

pred = predict(mod, iris.task, subset = test)

print(calculateConfusionMatrix(pred))

print(calculateConfusionMatrix(pred, sums = TRUE))

print(calculateConfusionMatrix(pred, relative = TRUE))

# now after cross-validation

r = crossval("classif.lda", iris.task, iters = 2L)

print(calculateConfusionMatrix(r$pred))

*De spreadsheet “*testprogramma confusionmatrix.xlsx*” kan je gebruiken om te testen of je functie de juiste waarden oplevert.*

**Gevraagde antwoorden:**

Geef het r-script (de r-code).

new.confusion <- function(arg1,arg2){

table(arg1,arg2);

sensitivity <- TP / (FN + TP);

specificity <- TN / (TN + FP);

accuracy <- (TN + TP) / (TN + FP + TP + FN); //true negative + true positive / specificity + sensitivity

tpr <- sensitivity;

fpr <- specificity – 1;

print(sensitivity);

print(specificity);

print(accuracy);

}

tpr= Σ True positive/Σ Condition positive

fpr = Σ False positive/Σ Condition negative

**1d)** Bepaal de Confusion Matrix voor de gegevens volgens het uiteindelijke model. Daaraan kan je zien in hoeverre dat model de gegevens goed kan voorspellen Doe dit voor een drempelwaarde van 0.1, 0.2, 0.6 en 0.9. Bereken steeds de accuracy, de specificity en de sensitivity.

**Gevraagde antwoorden:**

Geef de vier confusion matrices met daarbij steeds de accuracy, specificity en de sensitivity.

**1e)** Welke drempelwaarde geeft de beste resultaten en waarom?

**Gevraagde antwoorden:**

Beantwoord de vraag.

## Vraag 2

Bepaal de ROC-curve voor dit model.

De ROC-curve geeft aan hoe de specificity en sensitivity afhangen van de gekozen drempelwaarden.

Laad eerst de ROCR-bibliotheek in.

**2a)** Maak een ROC-plot van het logistic regression model uit opgave 1. Gebruik colorize en eventueel print.cutoffs.at om het effect van verschillende drempelwaarden beter te kunnen zien.

**Gevraagde antwoorden:**

Geef alle R-commando’s en de plot.

Library(ROCR)

ROCRpred=prediction(predictTest,test$TenYearCHD)

ROCRperf=performance(ROCRpred, ”tpr”, “fpr”)

plot(ROCRperf,colorize=true,print.cutoffs.at=seq(0,1,0.1))

**2b)** Wat is naar jouw mening de beste waarde voor de drempel-waarde (treshold) ? Houd rekening met de omstandigheden waarom een bepaalde threshold-waarde het beste zou kunnen zijn.

**Gevraagde antwoorden:**

Beantwoord de vraag.

## Vraag 3

Vergelijk het opgestelde model met een testset om te zien hoe goed het model nieuwe gegevens kan voorspellen.

We gaan nu het model uit opgave 1 vergelijken met een testset om te zien hoe goed het model nieuwe gegevens kan voorspellen. De testset heet **telcotest.csv** en bevat extra gegevens volgens dezelfde data structuur als **telcotrain**. Hint: gebruik de parameter *newdata* in de functie predict om de nieuwe resultaten te voorspellen.

Laad deze testdata in een apart dataframe.

Library(caTools)

set.seed(9022)

split=sample.split(data$tenYearHCD, SplitRatio=.65)

train=subset(data,split=TRUE)

test=subset(data,split==FALSE)

**3a**) Gebruik de predict() functie en het model uit opgave 1 om de kans op Churn voor deze testdata te voorspellen.

**Gevraagde antwoorden:**

Geef alle R-commando’s en de R-uitvoer.

Library(ROCR)

ROCRpred=prediction(predictTest,test$TenYearCHD)

ROCRperf=performance(ROCRpred, ”tpr”, “fpr”)

plot(ROCRperf,colorize=true,print.cutoffs.at=seq(0,1,0.1))

abline(0,1)

**3b)** Bepaal de confusion matrix voor de testset volgens dit model. Doe dit voor de bij 1d en 2a bepaalde optimale drempelwaarde. Bereken ook de accuracy, specificity en de sensitivity.

**Gevraagde antwoorden:**

Geef de confusion matrix en de berekende kengetallen.

**3c**) Bespreek de resultaten. Hoe goed (of slecht) is het bij 1 bepaalde model bij het voorspellen van de testdata?

**Gevraagde antwoorden:**

Beantwoord de vraag.