# 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.

> data=read.csv("telcotrain.csv")

> str(data)

'data.frame': 2500 obs. of 20 variables:

$ X : int 2 3 4 5 6 7 9 11 12 14 ...

$ ACL : int 107 137 84 75 118 121 117 65 74 95 ...

$ ZIP : int 415 415 408 415 510 510 408 415 415 510 ...

$ IP : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 1 1 1 1 1 ...

$ VMP : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 2 1 1 1 1 ...

$ NVM : int 26 0 0 0 0 24 0 0 0 0 ...

$ TDM : num 162 243 299 167 223 ...

$ TDC : int 123 114 71 113 98 88 97 137 127 88 ...

$ TDCH : num 27.5 41.4 50.9 28.3 38 ...

$ TEM : num 195.5 121.2 61.9 148.3 220.6 ...

$ TEC : int 103 110 88 122 101 108 80 83 148 75 ...

$ TECH : num 16.62 10.3 5.26 12.61 18.75 ...

$ TNM : num 254 163 197 187 204 ...

$ TNC : int 103 104 89 121 118 118 90 111 94 115 ...

$ TNCH : num 11.45 7.32 8.86 8.41 9.18 ...

$ TIM : num 13.7 12.2 6.6 10.1 6.3 7.5 8.7 12.7 9.1 12.3 ...

$ TIC : int 3 5 7 3 6 7 4 6 5 5 ...

$ TICH : num 3.7 3.29 1.78 2.73 1.7 2.03 2.35 3.43 2.46 3.32 ...

$ CSC : int 1 0 2 3 0 3 1 4 0 3 ...

$ Churn: Factor w/ 2 levels "NO","YES": 1 1 1 1 1 1 1 2 1 1 ...

> View(data)

> data$X <- NULL

> data$ZIP <- NULL

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

> summary(model)

Call:

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

TEM + TEC + TECH + TNM + TNC + TNCH + TIM + TIC + TICH +

CSC, family = "binomial", data = data)

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~IP+VMP+NVM+TDM+TDC+TDCH+TEM+TEC+TECH+TNM+TNC+TNCH+TIM+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

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

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

data = data)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0168 -0.5078 -0.3383 -0.1923 3.3390

Coefficients:

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

(Intercept) -8.841e+00 8.310e-01 -10.640 < 2e-16 \*\*\*

IPYes 1.991e+00 1.691e-01 11.773 < 2e-16 \*\*\*

VMPYes -2.319e+00 6.575e-01 -3.527 0.00042 \*\*\*

NVM 4.700e-02 2.020e-02 2.327 0.01995 \*

TDM -1.865e+00 3.757e+00 -0.496 0.61962

TDC 4.313e-03 3.167e-03 1.362 0.17327

TDCH 1.105e+01 2.210e+01 0.500 0.61711

TEM 1.416e+00 1.887e+00 0.750 0.45300

TEC 3.993e-04 3.259e-03 0.123 0.90248

TECH -1.658e+01 2.220e+01 -0.747 0.45520

TNM 6.945e-01 1.014e+00 0.685 0.49347

TNC 3.922e-04 3.261e-03 0.120 0.90427

TNCH -1.533e+01 2.253e+01 -0.680 0.49634

TIM -8.657e-01 6.208e+00 -0.139 0.88911

TIC -8.219e-02 2.936e-02 -2.800 0.00512 \*\*

TICH 3.532e+00 2.299e+01 0.154 0.87791

CSC 5.173e-01 4.514e-02 11.460 < 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: 1614.5 on 2483 degrees of freedom

AIC: 1648.5

Number of Fisher Scoring iterations: 6

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

> summary(model)

Call:

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

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

data = data)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0089 -0.5095 -0.3402 -0.1924 3.3255

Coefficients:

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

(Intercept) -8.835e+00 8.308e-01 -10.635 < 2e-16 \*\*\*

IPYes 1.992e+00 1.691e-01 11.786 < 2e-16 \*\*\*

VMPYes -2.315e+00 6.574e-01 -3.521 0.000429 \*\*\*

NVM 4.690e-02 2.019e-02 2.323 0.020184 \*

TDC 4.328e-03 3.166e-03 1.367 0.171703

TDCH 7.872e-02 7.399e-03 10.640 < 2e-16 \*\*\*

TEM 1.403e+00 1.887e+00 0.743 0.457243

TEC 3.884e-04 3.259e-03 0.119 0.905141

TECH -1.642e+01 2.220e+01 -0.740 0.459457

TNM 6.990e-01 1.014e+00 0.690 0.490448

TNC 4.257e-04 3.259e-03 0.131 0.896073

TNCH -1.543e+01 2.252e+01 -0.685 0.493298

TIM -8.884e-01 6.206e+00 -0.143 0.886179

TIC -8.161e-02 2.932e-02 -2.784 0.005375 \*\*

TICH 3.616e+00 2.299e+01 0.157 0.874990

CSC 5.168e-01 4.513e-02 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: 1614.8 on 2484 degrees of freedom

AIC: 1646.8

Number of Fisher Scoring iterations: 6

> model=glm(Churn~IP+VMP+NVM+TDCH+TEM+TEC+TECH+TNM+TNC+TNCH+TIM+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TEM + TEC + TECH +

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

data = data)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0602 -0.5135 -0.3394 -0.1952 3.3018

Coefficients:

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

(Intercept) -8.385e+00 7.584e-01 -11.056 < 2e-16 \*\*\*

IPYes 1.998e+00 1.691e-01 11.816 < 2e-16 \*\*\*

VMPYes -2.316e+00 6.566e-01 -3.527 0.00042 \*\*\*

NVM 4.704e-02 2.017e-02 2.332 0.01970 \*

TDCH 7.874e-02 7.395e-03 10.648 < 2e-16 \*\*\*

TEM 1.367e+00 1.885e+00 0.725 0.46820

TEC 3.334e-04 3.254e-03 0.102 0.91840

TECH -1.601e+01 2.218e+01 -0.722 0.47044

TNM 7.311e-01 1.013e+00 0.722 0.47028

TNC 3.524e-04 3.259e-03 0.108 0.91388

TNCH -1.614e+01 2.250e+01 -0.717 0.47308

TIM -1.170e+00 6.196e+00 -0.189 0.85020

TIC -8.323e-02 2.928e-02 -2.843 0.00447 \*\*

TICH 4.663e+00 2.295e+01 0.203 0.83899

CSC 5.152e-01 4.504e-02 11.439 < 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: 1616.6 on 2485 degrees of freedom

AIC: 1646.6

Number of Fisher Scoring iterations: 6

> model=glm(Churn~IP+VMP+NVM+TDCH+TEC+TECH+TNM+TNC+TNCH+TIM+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TEC + TECH + TNM +

TNC + TNCH + TIM + TIC + TICH + CSC, family = "binomial",

data = data)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0497 -0.5143 -0.3383 -0.1962 3.2853

Coefficients:

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

(Intercept) -8.405e+00 7.580e-01 -11.088 < 2e-16 \*\*\*

IPYes 1.996e+00 1.690e-01 11.813 < 2e-16 \*\*\*

VMPYes -2.301e+00 6.555e-01 -3.510 0.000447 \*\*\*

NVM 4.645e-02 2.014e-02 2.307 0.021067 \*

TDCH 7.895e-02 7.392e-03 10.681 < 2e-16 \*\*\*

TEC 3.937e-04 3.252e-03 0.121 0.903659

TECH 8.086e-02 1.571e-02 5.148 2.64e-07 \*\*\*

TNM 7.309e-01 1.012e+00 0.722 0.470380

TNC 3.277e-04 3.257e-03 0.101 0.919865

TNCH -1.614e+01 2.250e+01 -0.717 0.473197

TIM -1.181e+00 6.192e+00 -0.191 0.848765

TIC -8.309e-02 2.927e-02 -2.838 0.004533 \*\*

TICH 4.702e+00 2.293e+01 0.205 0.837546

CSC 5.155e-01 4.504e-02 11.447 < 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: 1617.2 on 2486 degrees of freedom

AIC: 1645.2

Number of Fisher Scoring iterations: 6

> model=glm(Churn~IP+VMP+NVM+TDCH+TECH+TNM+TNC+TNCH+TIM+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TECH + TNM + TNC +

TNCH + TIM + TIC + TICH + CSC, family = "binomial", data = data)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0492 -0.5135 -0.3389 -0.1956 3.2834

Coefficients:

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

(Intercept) -8.364760 0.682854 -12.250 < 2e-16 \*\*\*

IPYes 1.996389 0.168969 11.815 < 2e-16 \*\*\*

VMPYes -2.301355 0.655417 -3.511 0.000446 \*\*\*

NVM 0.046463 0.020134 2.308 0.021016 \*

TDCH 0.078961 0.007392 10.681 < 2e-16 \*\*\*

TECH 0.080842 0.015706 5.147 2.65e-07 \*\*\*

TNM 0.730940 1.012404 0.722 0.470304

TNC 0.000325 0.003257 0.100 0.920504

TNCH -16.139598 22.497030 -0.717 0.473121

TIM -1.178434 6.192951 -0.190 0.849085

TIC -0.083091 0.029272 -2.839 0.004532 \*\*

TICH 4.693342 22.936095 0.205 0.837864

CSC 0.515502 0.045030 11.448 < 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: 1617.2 on 2487 degrees of freedom

AIC: 1643.2

Number of Fisher Scoring iterations: 6

> model=glm(Churn~IP+VMP+NVM+TDCH+TECH+TNC+TNCH+TIM+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TECH + TNC + TNCH +

TIM + TIC + TICH + CSC, family = "binomial", data = data)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0669 -0.5126 -0.3389 -0.1947 3.2708

Coefficients:

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

(Intercept) -8.3634285 0.6826315 -12.252 < 2e-16 \*\*\*

IPYes 1.9961743 0.1689596 11.815 < 2e-16 \*\*\*

VMPYes -2.2995066 0.6549334 -3.511 0.000446 \*\*\*

NVM 0.0464829 0.0201222 2.310 0.020886 \*

TDCH 0.0790287 0.0073921 10.691 < 2e-16 \*\*\*

TECH 0.0810290 0.0157119 5.157 2.51e-07 \*\*\*

TNC 0.0003877 0.0032549 0.119 0.905180

TNCH 0.1029297 0.0284789 3.614 0.000301 \*\*\*

TIM -1.1152037 6.1915266 -0.180 0.857060

TIC -0.0838855 0.0292447 -2.868 0.004125 \*\*

TICH 4.4581970 22.9308192 0.194 0.845847

CSC 0.5140395 0.0449420 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 2488 degrees of freedom

AIC: 1641.7

Number of Fisher Scoring iterations: 6

> model=glm(Churn~IP+VMP+NVM+TDCH+TECH+TNCH+TIM+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TECH + TNCH + TIM +

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

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0688 -0.5128 -0.3380 -0.1950 3.2716

Coefficients:

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

(Intercept) -8.325324 0.602721 -13.813 < 2e-16 \*\*\*

IPYes 1.996661 0.168912 11.821 < 2e-16 \*\*\*

VMPYes -2.295314 0.653904 -3.510 0.000448 \*\*\*

NVM 0.046357 0.020092 2.307 0.021043 \*

TDCH 0.079036 0.007391 10.694 < 2e-16 \*\*\*

TECH 0.081000 0.015710 5.156 2.52e-07 \*\*\*

TNCH 0.103014 0.028470 3.618 0.000297 \*\*\*

TIM -1.110801 6.191600 -0.179 0.857620

TIC -0.083805 0.029240 -2.866 0.004155 \*\*

TICH 4.441818 22.931095 0.194 0.846409

CSC 0.514004 0.044934 11.439 < 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 2489 degrees of freedom

AIC: 1639.7

Number of Fisher Scoring iterations: 6

ACL+IP+VMP+NVM+TDM+TDC+TDCH+TEM+TEC+TECH+TNM+TNC+TNCH+TIM+TIC+TICH

> model=glm(Churn~IP+VMP+NVM+TDCH+TECH+TNCH+TIC+TICH+CSC, data=data, family="binomial")

> summary(model)

Call:

glm(formula = Churn ~ IP + VMP + NVM + TDCH + TECH + TNCH + TIC +

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

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0729 -0.5127 -0.3380 -0.1951 3.2767

Coefficients:

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

(Intercept) -8.32520 0.60280 -13.811 < 2e-16 \*\*\*

IPYes 1.99572 0.16883 11.821 < 2e-16 \*\*\*

VMPYes -2.29403 0.65401 -3.508 0.000452 \*\*\*

NVM 0.04633 0.02010 2.305 0.021149 \*

TDCH 0.07904 0.00739 10.695 < 2e-16 \*\*\*

TECH 0.08094 0.01571 5.153 2.56e-07 \*\*\*

TNCH 0.10317 0.02846 3.625 0.000289 \*\*\*

TIC -0.08350 0.02918 -2.861 0.004221 \*\*

TICH 0.32791 0.08663 3.785 0.000154 \*\*\*

CSC 0.51387 0.04493 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: 1617.8 on 2490 degrees of freedom

AIC: 1637.8

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.

*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).

conf <- function(n, d)

{

predicttest= predict (model, type="response")

t=table(n, predicttest > d)

TN=t[1]

FP=t[2]

FN=t[3]

TP=t[4]

sensitivity <- TP / (FN + TP)

specificity <- TN / (TN + FP)

accuracy <- (TN + TP) / (TN + TP + FP + FN)

print(t)

print(paste("Sensitivity = ", sensitivity))

print(paste("Specificify = ", specificity))

print(paste("Accuracy = ", accuracy))

}

**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.

> conf(data$Churn, 0.1)

n FALSE TRUE

NO 1352 786

YES 50 312

[1] "Sensitivity = 0.284153005464481"

[1] "Specificify = 0.964336661911555"

[1] "Accuracy = 0.6656"

> conf(data$Churn, 0.2)

n FALSE TRUE

NO 1795 343

YES 125 237

[1] "Sensitivity = 0.408620689655172"

[1] "Specificify = 0.934895833333333"

[1] "Accuracy = 0.8128"

> conf(data$Churn, 0.6)

n FALSE TRUE

NO 2110 28

YES 309 53

[1] "Sensitivity = 0.654320987654321"

[1] "Specificify = 0.872261264985531"

[1] "Accuracy = 0.8652"

> conf(data$Churn, 0.9)

n FALSE TRUE

NO 2138 0

YES 357 5

[1] "Sensitivity = 1"

[1] "Specificify = 0.856913827655311"

[1] "Accuracy = 0.8572"

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

**Gevraagde antwoorden:**

Beantwoord de vraag.

0.6, hoogste accuracy

## 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.

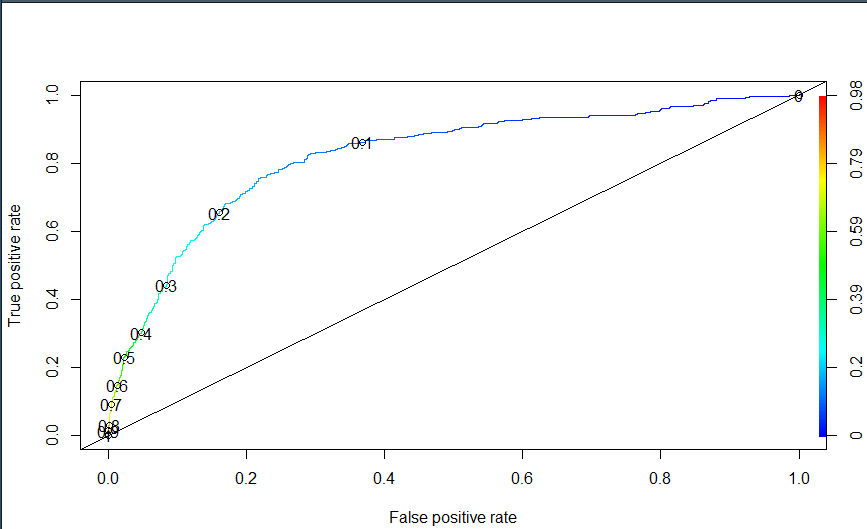
> prediction <- predict(model,type="response")

> ROCRpred = prediction(prediction,data$Churn)

> ROCRperf=performance(ROCRpred,"tpr","fpr")

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

> abline(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.

Aan de hand van deze grafiek zie ik dat na 0.2 de lijn erg snel naar rechts beweegt. Waardoor de

Hoeveelheid false positieves toeneemt Als dit belangrijk is in je onderzoek zou ik een waarde van

0.2 aanraden. Als je Echter een onderzoek doet waarbij het belangrijk is dat je zoveel mogelijk

Mensen eruit pakt die positief zouden zijn zou ik een alsnog veel hogere waarde uitzoeken. Als het heel

erg belangrijk is zou ik een waarde van 0.8 Aanraden.

## 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.

**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.

> test <- read.csv("telcotest.csv")

> test$X <- NULL

> test$ZIP <- NULL

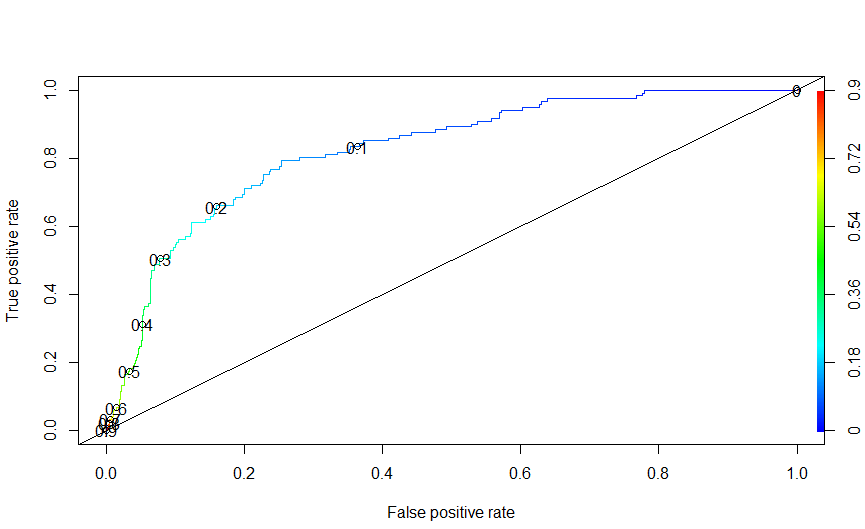
> newPrediction <- predict(model,type="response", newdata = test)

> ROCRpred = prediction(newPrediction,test$Churn)

> 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.

> conf2(test$Churn, 0.2)

n FALSE TRUE

NO 598 114

YES 42 79

[1] "Sensitivity = 0.409326424870466"

[1] "Specificify = 0.934375"

[1] "Accuracy = 0.812725090036014"

> conf2(test$Churn, 0.6)

n FALSE TRUE

NO 702 10

YES 113 8

[1] "Sensitivity = 0.444444444444444"

[1] "Specificify = 0.861349693251534"

[1] "Accuracy = 0.85234093637455"

> conf2(test$Churn, 0.8)

n FALSE TRUE

NO 709 3

YES 119 2

[1] "Sensitivity = 0.4"

[1] "Specificify = 0.856280193236715"

[1] "Accuracy = 0.853541416566627"

**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.

De accuracy bij een drempelwaarde van 0.6 is bij beide opdrachten de drempelwaarde met de hoogste accuracy, en deze blijkt lager te zijn bij 3b dan bij opdracht 1d. Overigens is de accuracy nog steeds best hoog. Het model is dus slechter met de testdata dan met de andere set, maar nog steeds best goed.