# Opgave 4 Analytics 2019

Naam student: *graag hier invullen*

Data voor deze opgave: twts.csv (documentatie: twts.txt).

## Inleveren

Inleveren UITSLUITEND via Blackboard. Uiterste datum van inleveren: 24-3-2019.

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 het analyseren van tweets. De vraag is om vast te stellen of een tweet een positieve dan wel een negatieve lading heeft.

Bij deze opgave begin je met ruwe data. In de data staan een aantal variabelen die je waarschijnlijk beter niet kan gebruiken. Het kan zij dat een aantal tweets meerdere malen voorkomt. (Misschien is het beter om deze teksten eerst te ontdubbelen, misschien maakt dat ook niet zo veel uit) Het is aan jou om deze data zodanig te behandelen dat je de analyse goed kan uitvoeren. De afhankelijke variabele is “Sentiment”.

De vraag is om een zo goed mogelijk CART-tree model te maken om uit de tekst van een tweet op te maken of het een positieve of negatieve lading heeft. De methode (text analytics) is uitgelegd op het college en is ook te vinden in de video’s “Turning Tweets into Knowledge”.

Bij deze opgave wordt van je verwacht dat je zelf de juiste stappen zet om tot het gewenste eindresultaat te komen. Probeer zo goed mogelijk uit te leggen WAAROM je bepaalde stappen zet.

## Vraag 1

Maak een zo optimaal mogelijk CART-tree model om uit de tekst van een tweet het (positieve/negatieve) sentiment te halen.

Let op: de Sentiment-variabele heeft DRIE verschillende waarden. Maak hiervan eerst een factor met slechts TWEE categorieen (positief/negatief). (Om te zorgen dat het een factor wordt kan je de functie as.factor gebruiken).

Gebruik daarvoor de tekst-analyse methodiek uit het college en/of de video’s. Zorg er ook voor dat je de data splitst in een trainingsset en een testset, zodat je de validatie goed kan uitvoeren.

In je antwoorddocument verwacht ik in elk geval:

* Alle R-commando’s
* Relevant commentaar zodat het duidelijk is wat je doet en waarom je het doet
* Alle relevante R-uitvoer (bijvoorbeeld confusion matrices)
* Alle relevante R-plots
* Niet alleen het opstellen van een model (op basis training data) maar ook validatie van het model (op basis van testdata). Validatie bestaat in elk geval uit confusion matrix + accuracy + sensitivity + specificity en een ROC-curve, plus een argumentatie over hoe goed het model de betreffende data kan voorspellen.
* Een goed onderbouwde conclusie

Eerst opvragen file data en tonen wat er in zit.

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

> getwd

> tweets <- read.csv(file ='twts.csv', stringsAsFactors = FALSE)

> str(tweets)

'data.frame': 498 obs. of 6 variables:

$ Sentiment: int 4 4 4 4 4 4 0 4 4 4 ...

$ ID : int 3 4 5 6 7 8 9 10 11 12 ...

$ Date : chr "Mon May 11 03:17:40 UTC 2009" "Mon May 11 03:18:03 UTC 2009" "Mon May 11 03:18:54 UTC 2009" "Mon May 11 03:19:04 UTC 2009" ...

$ Query : chr "kindle2" "kindle2" "kindle2" "kindle2" ...

$ User : chr "tpryan" "vcu451" "chadfu" "SIX15" ...

$ Tweet : chr "@stellargirl I loooooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right." "Reading my kindle2... Love it... Lee childs is good read." "Ok, first assesment of the #kindle2 ...it fucking rocks!!!" "@kenburbary You'll love your Kindle2. I've had mine for a few months and never looked back. The new big one is "| \_\_truncated\_\_ ...

Maak kolom aan met negatief sentiment, in dit geval lager dan 2. Blijkbaar gaat sentiment van 0 tot 4.

> tweets$Negative = as.factor(tweets$Sentiment <2)

> table(tweets$Negative)

FALSE TRUE

321 177

Packages installeren en library gebruiken voor opschonen data.Zoals naar kleine letters, leestekens verwijderen, engelse stopwoorden verwijderen en de stam van woorden gebruiken.

> install.packages("SnowballC")

> install.packages("tm")

> library(tm)

> library(SnowballC)

> corpus= Corpus(VectorSource(tweets$Tweet))

> corpus = tm\_map(corpus, tolower)

> corpus = tm\_map(corpus, removePunctuation)

> stopwords("english")[1:10]

> corpus = tm\_map(corpus, removeWords, c("kindle", stopwords("english")))

En dan stam van woorden.

> corpus = tm\_map(corpus,stemDocument)

Frequentie van woorden zien, daarvoor moet eerst vector source anders lukt documenttermmatrix niet.

> frequencies = DocumentTermMatrix(corpus)

> str(frequencies)

List of 6

$ i : int [1:2221] 1 1 1 1 1 1 1 1 1 1 ...

$ j : int [1:2221] 1 2 3 4 5 6 7 8 9 10 ...

$ v : num [1:2221] 1 1 1 1 1 1 1 1 1 1 ...

$ nrow : int 3

$ ncol : int 2221

$ dimnames:List of 2

..$ Docs : chr [1:3] "1" "2" "3"

..$ Terms: chr [1:2221] "\"10" "\"45" "\"absolut" "\"accanni" ...

- attr(\*, "class")= chr [1:2] "DocumentTermMatrix" "simple\_triplet\_matrix"

- attr(\*, "weighting")= chr [1:2] "term frequency" "tf"

> inspect(frequencies)

<<DocumentTermMatrix (documents: 3, terms: 2221)>>

Non-/sparse entries: 2221/4442

Sparsity : 67%

Maximal term length: 48

Weighting : term frequency (tf)

Sample :

Terms

Docs "just get good love museum new night see time warner

1 20 23 24 28 25 31 31 25 37 30

2 0 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

Verwijder zeldzame termen/woorden.

> sparse = removeSparseTerms(frequencies, 0.995)

Maak data frame.

> tweetsSparse = as.data.frame(as.matrix(sparse))

Maak kolommen aan van overgebleven termen die niet zeldzaam zijn.

> colnames(tweetsSparse) = make.names(colnames(tweetsSparse))

Random library laden en seed zetten op mijn postcode vier cijfers zodat herhaalbaar is.

> library(caTools)

> set.seed(9022)

Afhankelijke variabele toevoegen.

> tweetsSparse$Negative = tweets$Negative

Splitten.

> split = sample.split(tweetsSparse$Negative, SplitRatio = 0.7)

> trainSparse = subset(tweetsSparse, split==TRUE)

> testSparse = subset(tweetsSparse, split==FALSE)

Library voor plot.

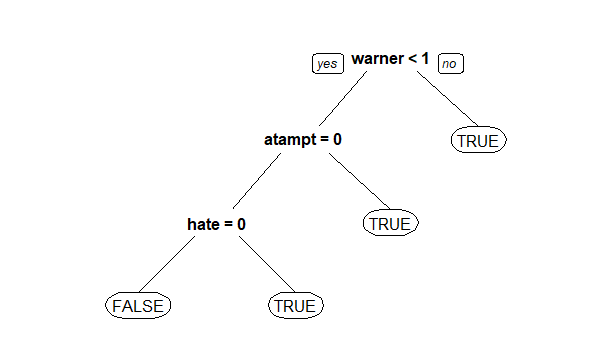
> library(rpart)

> library(rpart.plot)

Cart tree maken en plotten.

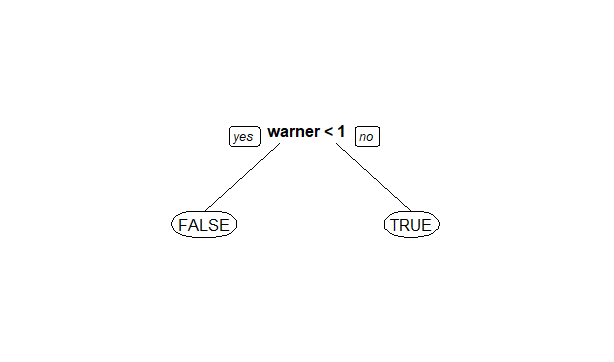
tweetCart = rpart(Negative~.,data=trainSparse,method="class")

prp(tweetCart)



> tree2=rpart(Negative~.,data=tweetsSparse,method=”class”,control=rpart.control(minbucket=25))

> prp(tree2)



Prediction en ROC en AUC maken en plotten.

> predictTweet=predict(tweetCart,tweetsSparse)

> library(ROCR)

Confusion matrix maken.

> 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(predictTweet[,2], tweetsSparse$Negative, 0.5)

afh FALSE TRUE

FALSE 316 5

TRUE 120 57

[1] "accuracy "

[1] 74.8996

[1] "sensitivity"

[1] 32.20339

[1] "specificity "

[1] 98.44237

> predictROC=predict(model,type="response",newdata = tweetsSparse)

> ROCRpred=prediction(predictTweet,tweetsSparse$Negative)

> ROCRpred=prediction(predictTweet[,2],tweetsSparse$Negative)

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

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

> abline(0,1)

Afbeelding met kaart, tekst

Automatisch gegenereerde beschrijving

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

> AUC

[1] 0.6534752

Rond de 0.1 is het al een hoge area under curve. Het heeft dan al de hoge mogelijke accuracy. De accuracy met 72 is niet heel hoog en de sensitivity

is nog lager, het vindt weinig reële positieven. De area under curve is ook niet heel erg hoog, maar hoger dan 0.65.

## Vraag 2

Bij de eerste vraag heb je de 3 waarden voor Sentiment omgezet in 2 waarden voor de tekst-analyse. Je zou ook een CART-tree kunnen maken met daarbij alle drie waarden van Sentiment. Doe dit en geef het resultaat.

Gebruik cross-validation om de best mogelijke cp-parameter vast te stellen hiervoor.

De confusion-matrix is nu 3 bij 3. Geef aan wat dat voor consequenties heeft voor je validatie.

In je antwoorddocument verwacht ik in elk geval:

* Alle R-commando’s
* Relevant commentaar zodat het duidelijk is wat je doet en waarom je het doet
* Alle relevante R-uitvoer (bijvoorbeeld confusion matrices)
* Alle relevante R-plots
* Je antwoord op de vraag in de laatste zin van vraag 2 (over de 3x3 matrix)
* Een goed onderbouwde conclusie

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

> getwd

> tweets <- read.csv(file ='twts.csv', stringsAsFactors = FALSE)

> str(tweets)

'data.frame': 498 obs. of 6 variables:

$ Sentiment: int 4 4 4 4 4 4 0 4 4 4 ...

$ ID : int 3 4 5 6 7 8 9 10 11 12 ...

$ Date : chr "Mon May 11 03:17:40 UTC 2009" "Mon May 11 03:18:03 UTC 2009" "Mon May 11 03:18:54 UTC 2009" "Mon May 11 03:19:04 UTC 2009" ...

$ Query : chr "kindle2" "kindle2" "kindle2" "kindle2" ...

$ User : chr "tpryan" "vcu451" "chadfu" "SIX15" ...

$ Tweet : chr "@stellargirl I loooooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right." "Reading my kindle2... Love it... Lee childs is good read." "Ok, first assesment of the #kindle2 ...it fucking rocks!!!" "@kenburbary You'll love your Kindle2. I've had mine for a few months and never looked back. The new big one is "| \_\_truncated\_\_ ...

I dit geval lager dan 2 hoeft niet, want je wilt de drie behouden.

> table(tweets$Sentiment)

0 2 4

177 139 182

Packages installeren en library gebruiken voor opschonen data. Zoals naar kleine letters, leestekens verwijderen, engelse stopwoorden verwijderen en de stam van woorden gebruiken.

> library(tm)

> library(SnowballC)

> corpus= Corpus(VectorSource(tweets$Tweet))

> corpus = tm\_map(corpus, tolower)

> corpus = tm\_map(corpus, removePunctuation)

> corpus = tm\_map(corpus, removeWords, c("kindle", stopwords("english")))

En dan stam van woorden.

> corpus = tm\_map(corpus,stemDocument)

Frequentie van woorden zien, daarvoor moet eerst vector source anders lukt documenttermmatrix niet.

> frequencies = DocumentTermMatrix(corpus)

> str(frequencies)

List of 6

$ i : int [1:2221] 1 1 1 1 1 1 1 1 1 1 ...

$ j : int [1:2221] 1 2 3 4 5 6 7 8 9 10 ...

$ v : num [1:2221] 1 1 1 1 1 1 1 1 1 1 ...

$ nrow : int 3

$ ncol : int 2221

$ dimnames:List of 2

..$ Docs : chr [1:3] "1" "2" "3"

..$ Terms: chr [1:2221] "\"10" "\"45" "\"absolut" "\"accanni" ...

- attr(\*, "class")= chr [1:2] "DocumentTermMatrix" "simple\_triplet\_matrix"

- attr(\*, "weighting")= chr [1:2] "term frequency" "tf"

Verwijder zeldzame termen/woorden.

> sparse = removeSparseTerms(frequencies, 0.995)

Maak data frame.

> tweetsSparse = as.data.frame(as.matrix(sparse))

Maak kolommen aan van overgebleven termen die niet zeldzaam zijn.

> colnames(tweetsSparse) = make.names(colnames(tweetsSparse))

Random library laden en seed zetten op mijn postcode vier cijfers zodat herhaalbaar is.

> library(caTools)

> set.seed(9022)

Afhankelijke zetten.

> tweetsSparse$Sentiment = tweets$Sentiment

Splitten.

> split = sample.split(tweetsSparse$Sentiment, SplitRatio = 0.7)

> trainSparse = subset(tweetsSparse, split==TRUE)

> testSparse = subset(tweetsSparse, split==FALSE)

Library voor plot en cv.

> library(rpart)

> library(rpart.plot)

library(caTools)

colnames(tweetsSparse) = make.names(colnames(tweetsSparse))

library(caret)

Cross Validation.

> fitcontrol=trainControl(method="cv",number=10)

> fitcontrol = trainControl(method = "cv", number = 10)

> cartGrid=expand.grid(.cp=(1:50)\*0.01)

> cv=train(Sentiment~.,data=trainSparse,method="rpart",trainControl=fitControl,tuneGrid=cartGrid)

> cv=train(Sentiment~.,data=trainSparse,method="rpart",trControl=fitcontrol,tuneGrid=cartGrid)

> plot(cv)

Afbeelding met tekst, kaart

Automatisch gegenereerde beschrijving

Cart tree maken en plotten.

> tweetCart2 = rpart(Sentiment~.,data=trainSparse,method="class", cp=0.01)

prp(tweetCart2)

Afbeelding met tekst, kaart

Automatisch gegenereerde beschrijving

Prediction en ROC en AUC maken en plotten.

> library(ROCR)

> tweetsSparse$Sentiment=as.factor(tweetsSparse$Sentiment)

predictTree2=predict(tweetCart2,newdata = testSparse, type = "class")

> str(tweetsSparse$Sentiment)

Factor w/ 3 levels "0","2","4": 3 3 3 3 3 3 1 3 3 3 ...

> str(predictTree2)

Factor w/ 3 levels "0","2","4": 3 3 1 1 3 3 3 1 3 3 ...

- attr(\*, "names")= chr [1:150] "6" "10" "11" "12" ...

3 bij 3 Confusion matrix maken.

> conf3 <- function(pred, afh) {

+ tab = table(afh,pred)

+ print(tab)

+ print('accuracy ')

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

+ }

> conf3(predictTree2,testSparse$Sentiment)

pred

afh 0 2 4

0 44 4 5

2 22 14 6

4 23 8 24

[1] "accuracy "

[1] 54.66667

ROC kan niet 3D, is bedoeld voor 2D.

De accuracy 54.667 is niet heel hoog. Er waarden dan ook 3 waarden gebruikt in plaats van 2. Het komt net iets meer over met de werkelijke waarde dan het tossen van een munt. Over AUC is niks te zeggen, want het is 3D. Het zou hoger kunnen met meer tweets.

## Eindbeoordeling Analytics

Je wordt uitgenodigd om de resultaten van deze opgave in een assessment te bespreken met de docent. (Het assessment duurt ongeveer 8 minuten). Het eindcijfer van het vak Analytics wordt daarna bepaald uit het gemiddelde cijfer voor de vier opgaven en het cijfer voor het assessment. Voor een voldoende resultaat moeten minimaal drie opgaven zijn ingeleverd en het assessment moet met een voldoende zijn afgesloten.