````
---
title: "00 Selectie Data PILS"
author: "5.1.2.E"
date: "25 juni 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

Inlezen librarier

```{r librariers echo=FALSE, message=FALSE}

library(tidyverse) #Altijd
library("openxlsx")

```

## Including Plots

You can also embed plots, for example:

```{r schonen, echo=FALSE}

source("//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS/Scripts/Schonen Functie 2019-06-25.R")

```


Inlezen totaalbestand Nederland van project Monitor Eerlijk werken
Vervolgens selecteren wat Horeca & Detailhandel is, op basis van
sleutelbestand van 5.1.2.E

Vestigingen zonder SBI-code alsnog verwijderen.

```{r, echo=FALSE, message=FALSE}

df <- readRDS("//client/G$/I-SZW/O&A/2018 Monitor Eerlijk
werk/Data/Doelgroep/doelgroep_ext.rds")

X20190308_Horeca_DetailHandelSelectieGerard <- read_csv("//client/G$/I-
SZW/O&A/Data Science projecten/2019
RUM/Data/20190308_Horeca&DetailHandelSelectieGerard.csv")

PILS_scope <- X20190308_Horeca_DetailHandelSelectieGerard

PILS_scope <- PILS_scope %>%
  rename(KvKnummer12 = kvk12) %>%
  distinct()

df <- df %>%
  mutate(KvKnummer12 = as.character(KvKnummer12)) %>%
  distinct(KvKnummer12, .keep_all = TRUE)
````
````

```
# And make a selection for Horeca & Detailhandel
df_Horeca_Detail <- left_join(PILS_scope, df, by="KvKnummer12")

df_Horeca_Detail_clean <- df_Horeca_Detail %>%
  filter(!is.na(sbicode))

df_Horeca_Detail_clean <- schonen(df_Horeca_Detail_clean)

Data_Pils <- df_Horeca_Detail_clean

#setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
#write_rds(df_Horeca_clean_final, "2019-06-25 Horeca voor pre-
processing.rds")

```
Nu de CBS-risico-indicatoren op PC4 niveau toevoegen, daarvoor eerst PC4 toevoegen t.b.v. join PILS met CBS, en na joinng met PC4-tabel weer weggooien
```{r}
Data_Pils <- Data_Pils %>%
  mutate(PC4 = as.character(Postcode)) %>%
  mutate(PC4 = str_sub(PC4, 1,4))

setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
CBS_PC4 <- read_csv("2019-07-15 CBS_Risico_Indicatoren_PC4.csv")

CBS_PC4 <- CBS_PC4 %>%
  mutate(PC4 = as.character(PC4))

Data_Pils <- left_join(Data_Pils, CBS_PC4, by= "PC4")

Data_Pils <- select(Data_Pils, -PC4)
```

Nu de Personeelsverloop data toevoegen, eerst KvK8 aanmaken, daarna joinen en KvK8 weer verwijderen.
Gaat hier om berekening van personeelsverloop op basis van data uit UWV-Polisadministratie over dienstverbanden, op twee tijdstippen.
```{r}

UWV_Personeelsverloop <- read_csv("//client/G$/I-SZW/O&A/Data Science
projecten/2019 PILS/Data/2019-07-15 UWV Personeelsverloop.csv",
    col_types = cols(KvK_KvK8 = col_character())) %>%
    arrange(KvK_KvK8)%>%
    distinct(.keep_all = TRUE)

Data_Pils2 <- Data_Pils %>%
  mutate(KvK_KvK8 = str_sub(KvKnummer12,1,8)) %>%
  left_join(UWV_Personeelsverloop, by="KvK_KvK8") %>%
  select(-KvK_KvK8)

```


TWV en NVWA en UWV data inlezen
```{r}
```

```
TWV_data <- read.xlsx( '//Client/G$/I-SZW/O&A/Data Science projecten/2019
RUM/Data/TWV Aanvragen vanaf 2017.xlsx',1)
NVWA_data <- read.xlsx( '//Client/G$/I-SZW/O&A/Data Science
projecten/2019 RUM/Data/NVWA Aanvragen vanaf 2017.xlsx',1)
UWV_data <- read_csv("//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS/Data/Bronbestanden/20190816_PilsProfileringUWVData
uitbreidiging.csv")

```


NVWA en TWV data toevoegen
```{r}
NVWA_data  <- NVWA_data  %>%
  rename(KvKnummer12 = KvK12) %>%
  distinct(.keep_all = TRUE)

TWV_data  <- TWV_data  %>%
  rename(KvKnummer12 = KvK12) %>%
  distinct()

UWV_data  <- UWV_data  %>%
  rename(KvKnummer12 = KvK12)

UWV_data_clean <- UWV_data[!duplicated(UWV_data$KvKnummer12), ] %>%
  filter(!is.na(KvKnummer12))

join.1 <- left_join(Data_Pils2, NVWA_data, by="KvKnummer12")
join.2 <- left_join(join.1, TWV_data, by="KvKnummer12")
Final_Data_Pils <- left_join(join.2, UWV_data_clean, by="KvKnummer12")
%>%
  select(-RSIN, -RSIN_moeder_koppeling)

Final_Data_Pils <- Final_Data_Pils %>%
  mutate(Aantal_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017 =
as.numeric(Aantal_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017)) %>%
  mutate(Geweigerde_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017 =
as.numeric(Geweigerde_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017)) %>%
  select(-VestigingID, -KvK8)

setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
write_rds(Final_Data_Pils, "2019-08-16 Horeca_Na_pre-processing.rds")

```

````
---
title: "Exploratie data PILS"
author: "5.1,2,E"
date: "28 november 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax
for authoring HTML, PDF, and MS Word documents. For more details on using
R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code chunks
within the document. You can embed an R code chunk like this:

```{r cars}
library(tidyverse)
library(DataExplorer)
```

## Including Plots


```{r, echo=FALSE}
plot_missing(df_Horeca_Detail)
```
```{r}
plot_missing((Data_Pils))
```
```{r}
plot_missing(Final_Data_Pils)
```

```{r}
plot_missing(Df_Final)
```
````

````
---
title: "Inladen van de bestanden van de partners"
author: "5.1.2.E"
date: "2 juli 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r}
library(dplyr)
library(data.table)
library(knitr)
library(readxl)

```


Functie om pad flexibel te maken
```{r}

pad_data <- "H:/Mijn documenten/2019 Libra/Data/"
#pad_data   <- "//client/G$/I-SZW/O&A/Data Science projecten/2019 Libra
Kaart/Data/"

pad_bestand <- function(pad, bestand) {
  file <- paste0(pad, bestand)
  return(file) }

bestandsnaam <- "omzetten_gemCode_chloropleth.csv"
namen <- fread(pad_bestand(pad_data, bestandsnaam), colClasses =
"character")

```



Eerst PC4-bestand samenstellen, hiertoe eerst alles inlezen

```{r global, include=FALSE}


################## SVB gegevens inladen ##################
historisch_SVB_gemeente <-   fread(pad_bestand(pad_data,
"terugvorderingen_per_gemeente_201711.csv"))
voorspellend_SVB_gemeente <- fread(pad_bestand(pad_data,
"voorspelde_terugvorderingen_totaal.csv"), colClasses =
c(rep("character", 2), rep("numeric", 6)))

# Inladen PC4 svb
voorspellend_SVB_PC4 <-
fread(pad_bestand(pad_data,"voorspelde_terugvorderingen_totaal_pc4.csv"),
                                colClasses = c(rep("character", 2),
rep("numeric", 7))) %>%
````
````

```
  select(CODE_GEM, gemeentenaam, PC_NUMM, risico_aow_voor = Risico_AOW,
risico_anw_voor = Risico_ANW, risico_aio_voor = Risico_AIO)

# bins maken in 3 bakjes: laag, midden, hoog risico.
voorspellend_SVB_PC4$bak_aow_voor <-
as.numeric(Hmisc::cut2(voorspellend_SVB_PC4$risico_aow_voor, g=3))
voorspellend_SVB_PC4$bak_anw_voor <-
as.numeric(Hmisc::cut2(voorspellend_SVB_PC4$risico_anw_voor, g=3))
voorspellend_SVB_PC4$bak_aio_voor <-
as.numeric(Hmisc::cut2(voorspellend_SVB_PC4$risico_aio_voor, g=3))


historisch_SVB_PC4 <- fread(pad_bestand(pad_data,
"svb_pc4_historisch.csv"),
                            colClasses = c(rep("character", 2),
rep("numeric", 19))) %>%
  select(code, gemeentenaam, PC_NUMM, risico_aow_hist =
RELATIEF_AANTAL_TERUGVORDERINGEN_AOW, risico_anw_hist =
RELATIEF_AANTAL_TERUGVORDERINGEN_ANW, risico_aio_hist =
RELATIEF_AANTAL_TERUGVORDERINGEN_AIO, AANTAL_AOW, AANTAL_ANW, AANTAL_AIO)


historisch_SVB_PC4$bak_aow_hist <-
  as.numeric(Hmisc::cut2(historisch_SVB_PC4$risico_aow_hist, g=3))
historisch_SVB_PC4$bak_anw_hist <-
  as.numeric(Hmisc::cut2(historisch_SVB_PC4$risico_anw_hist, g=3))
historisch_SVB_PC4$bak_aio_hist <-
  as.numeric(Hmisc::cut2(historisch_SVB_PC4$risico_aio_hist, g=3))


historisch_SVB_gemeente$groepAOW <-

as.numeric(Hmisc::cut2(historisch_SVB_gemeente$RELATIEF_AANTAL_TERUGVORDE
RINGEN_AOW, g=3))
historisch_SVB_gemeente$groepANW <-

as.numeric(Hmisc::cut2(historisch_SVB_gemeente$RELATIEF_AANTAL_TERUGVORDE
RINGEN_ANW, g=3))
historisch_SVB_gemeente$groepAIO <-

as.numeric(Hmisc::cut2(historisch_SVB_gemeente$RELATIEF_AANTAL_TERUGVORDE
RINGEN_AIO, g=3))
historisch_SVB_gemeente$groepAOW[is.na(historisch_SVB_gemeente$groepAOW)]
<- 1
historisch_SVB_gemeente$groepANW[is.na(historisch_SVB_gemeente$groepANW)]
<- 1
historisch_SVB_gemeente$groepAIO[is.na(historisch_SVB_gemeente$groepAIO)]
<- 1
historisch_SVB_gemeente$groepTotaal <- historisch_SVB_gemeente$groepAOW *
historisch_SVB_gemeente$groepANW * historisch_SVB_gemeente$groepAIO

voorspellend_SVB_gemeente$groepAOW <-
as.numeric(Hmisc::cut2(voorspellend_SVB_gemeente$Risico_AOW, g=3))
voorspellend_SVB_gemeente$groepANW <-
as.numeric(Hmisc::cut2(voorspellend_SVB_gemeente$Risico_ANW, g=3))
voorspellend_SVB_gemeente$groepAIO <-
as.numeric(Hmisc::cut2(voorspellend_SVB_gemeente$Risico_AIO, g=3))
```

```
voorspellend_SVB_gemeente$groepAOW[is.na(voorspellend_SVB_gemeente$groepA
OW)] <- 1
voorspellend_SVB_gemeente$groepANW[is.na(voorspellend_SVB_gemeente$groepA
NW)] <- 1
voorspellend_SVB_gemeente$groepAIO[is.na(voorspellend_SVB_gemeente$groepA
IO)] <- 1
voorspellend_SVB_gemeente$groepTotaal <-
voorspellend_SVB_gemeente$groepAOW * voorspellend_SVB_gemeente$groepANW *
voorspellend_SVB_gemeente$groepAIO

################ Belastingdienst gegevens inladen, en gemeentecode c.q.
gemeentenaam toevoegen ###############

#Belastingdienst Toeslagen gegevens inladen
historisch_HT_gemeente <- fread( pad_bestand(pad_data,
"belastingdienst_hist_HT.csv"),
colClasses = c("character", rep("integer",4) ) )

historisch_KOT_gemeente <- fread( pad_bestand(pad_data,
"belastingdienst_hist_KOT.csv"),
colClasses = c("character", rep("integer",4) ) )

voorspellend_HT_gemeente  <- read_xlsx( pad_bestand(pad_data,
"GEMEENTEN_HT.xlsx") )

voorspellend_KOT_gemeente <- read_xlsx( pad_bestand(pad_data,
"GEMEENTEN_KOT.xlsx") )

historisch_HT_gemeente <- namen %>%
                          left_join(historisch_HT_gemeente,
                          by = c("code" = "gemeentecode"))

historisch_KOT_gemeente <- namen %>%
                          left_join(historisch_KOT_gemeente,
                          by = c("code" = "gemeentecode"))

voorspellend_HT_gemeente <- namen %>%
                          left_join(voorspellend_HT_gemeente,
                          by = c("code" = "gemeentecode"))

voorspellend_KOT_gemeente <- namen %>%
                          left_join(voorspellend_KOT_gemeente,
                          by = c("code" = "gemeentecode"))



# Inladen PC4 Belastingdienst
voorspellend_KOT_PC4 <- read_xlsx(pad_bestand(pad_data, "P4_KOT.xlsx"))
voorspellend_HT_PC4  <- read_xlsx(pad_bestand(pad_data, "P4_HT.xlsx"))

voorspellend_KOT_PC4 <- namen %>%
                         left_join(voorspellend_KOT_PC4,
                         by = c("code" = "gemeentecode"))

voorspellend_HT_PC4  <- namen %>%
                         left_join(voorspellend_HT_PC4,
                         by = c("code" = "gemeentecode"))
```

```
voorspellend_HT_gemeente$bak <-
  as.numeric(Hmisc::cut2(voorspellend_HT_gemeente$percentage_risico,
g=3))
voorspellend_KOT_gemeente$bak <-
  as.numeric(Hmisc::cut2(voorspellend_KOT_gemeente$percentage_risico,
g=3))
voorspellend_KOT_PC4$bak <-
as.numeric(Hmisc::cut2(voorspellend_KOT_PC4$percentage_risico, g=3))
voorspellend_HT_PC4$bak <-
as.numeric(Hmisc::cut2(voorspellend_HT_PC4$percentage_risico, g=3))

voorspellend_HT_PC4 <- voorspellend_HT_PC4 %>%
  rename(aantal_HT = aantal_risicoselectie, perc_HT = percentage_risico,
bak_HT = bak)


############### UWV gegevens inladen ###############
uwv_gemeente <- read_xlsx(pad_bestand(pad_data, "LSI data WW per gemeente
16-11-2017.xlsx"))

uwv_gemeente <- namen %>%
                left_join(uwv_gemeente,
                by = c("gemeentenaam" = "GemeenteNaam") ) %>%
                mutate (ScoreHistorisch = as.numeric(ScoreHistorisch),
                ScoreModel =   as.numeric(ScoreModel),
                groep_hist =   as.numeric(Hmisc::cut2(ScoreHistorisch,
g=3)),
                groep_voorsp = as.numeric(Hmisc::cut2(ScoreModel, g=3) )
)


uwv_pc4      <- read_xlsx(pad_bestand(pad_data,
"risicoscore_pc4_WW2017.xlsx")) %>%
                mutate (ScoreHistorisch = as.numeric(ScoreHistorisch),
                ScoreModel =   as.numeric(ScoreModel) ) %>%
                filter(!is.na(ScoreHistorisch) ) %>%
                mutate (groep_hist   =
as.numeric(Hmisc::cut2(ScoreHistorisch, g=3)),
                        groep_voorsp = as.numeric(Hmisc::cut2(ScoreModel,
g=3) ) )

############### IND gegevens inladen ###############
# Gefingeerde dienstverbanden

ind_gefingeerd_pc4 <- read_xlsx(pad_bestand(pad_data, bestand = "LSI Soc.
kaart van NL -IND gefingeerde dvb.xlsx")) %>%
  select(-Volgnummer)

ind_gefingeerd_gemeente <- ind_gefingeerd_pc4 %>%
  group_by(gemeentenummer, Gemeentenaam) %>%
  summarise(Risico = sum(Risicoweging))

ind_gefingeerd_pc4 <- ind_gefingeerd_pc4 %>%
  group_by(PC4, gemeentenummer, Gemeentenaam) %>%
  summarise(Risico = sum(Risicoweging)) %>%
  inner_join(namen %>% mutate(gemeentenummer = as.numeric(code))) %>%
  ungroup() %>%
```

```
  select(-gemeentenummer, -Gemeentenaam)

#ID fraude
ind_ID_pc4 <- read_xlsx(pad_bestand(pad_data, bestand ="LSI Soc. kaart
van NL -IND ID fraude.xlsx"))

ind_ID_gemeente <- ind_ID_pc4 %>%
  group_by(Gemeentenummer, Gemeentenaam) %>%
  summarise(Risico = sum(Risicoweging))

ind_ID_pc4 <- ind_ID_pc4 %>%
  group_by(PC4, Gemeentenummer, Gemeentenaam) %>%
  summarise(Risico = sum(Risicoweging)) %>%
  inner_join(namen %>% mutate(Gemeentenummer = as.numeric(code))) %>%
  ungroup() %>%
  select(-Gemeentenummer, -Gemeentenaam)

# groep toevoegen
# Groepen zo gemaakt dat bij geen informatie dan groep 1 (laag risico).
en de overige gemeenten opgedeeld in 2 groepen (middel en hoog).
ind_gefingeerd_gemeente$groep_gef <-
as.numeric(Hmisc::cut2(ind_gefingeerd_gemeente$Risico, g=2))+1

ind_gefingeerd_gemeente <- ind_gefingeerd_gemeente %>%
  right_join(namen %>% mutate(gemeentenummer = as.numeric(code))) %>%
  ungroup() %>%
  select(-gemeentenummer, -Gemeentenaam)

ind_gefingeerd_gemeente$Risico[is.na(ind_gefingeerd_gemeente$Risico)] <-
0
ind_gefingeerd_gemeente$groep_gef[is.na(ind_gefingeerd_gemeente$groep_gef
)] <- 1

ind_gefingeerd_pc4$groep_gef <-
as.numeric(Hmisc::cut2(ind_gefingeerd_pc4$Risico, g=2))+1

ind_ID_gemeente$groep_ID <-
as.numeric(Hmisc::cut2(ind_ID_gemeente$Risico, g=2))+1

ind_ID_gemeente <- ind_ID_gemeente %>%
  right_join(namen %>% mutate(Gemeentenummer = as.numeric(code))) %>%
  ungroup() %>%
  select(-Gemeentenummer, -Gemeentenaam)

ind_ID_gemeente$Risico[is.na(ind_ID_gemeente$Risico)] <- 0
ind_ID_gemeente$groep_ID[is.na(ind_ID_gemeente$groep_ID)] <- 1

ind_ID_pc4$groep_ID <- as.numeric(Hmisc::cut2(ind_ID_pc4$Risico, g=2))+1


############### inspectie SZW gegevens inladen #####

bestandsnaam   <- "Meldingen Arbo en AMF per postcode-4.xlsx"
inspectie_ARBO <- read_xlsx(pad_bestand(pad_data, bestand =
bestandsnaam), sheet = "ARBO-meldingen") %>%
  select(Postcode, Eindtotaal, gemeentenaam = Gemeente, code =
Gemeentecode) %>%
  filter(Postcode != "Eindtotaal") %>%
```

```
  mutate(PC4 = as.numeric(Postcode)) %>%
  select(-Postcode) %>%
  rename(ARBO_meldingen = Eindtotaal)

inspectie_AMF  <- read_xlsx(pad_bestand(pad_data, bestand =
bestandsnaam), sheet = "AMF-meldingen") %>%
  select(Postcode, Eindtotaal, gemeentenaam = Gemeente, code =
Gemeentecode) %>%
  filter(Postcode != "Eindtotaal") %>%
  mutate(PC4 = as.numeric(Postcode)) %>%
  select(-Postcode) %>%
  filter(!is.na(PC4)) %>%
  rename(AMF_meldingen = Eindtotaal)


inspectie_ARBO_gemeente <- inspectie_ARBO %>%
  group_by(gemeentenaam, code) %>%
  summarise(ARBO_meldingen = sum(ARBO_meldingen))

inspectie_AMF_gemeente <- inspectie_AMF %>%
  group_by(gemeentenaam, code) %>%
  summarise(AMF_meldingen = sum(AMF_meldingen))

inspectie_ARBO$groep_ARBO <-
as.numeric(Hmisc::cut2(inspectie_ARBO$ARBO_meldingen, g=2))+1
inspectie_AMF$groep_AMF <-
as.numeric(Hmisc::cut2(inspectie_AMF$AMF_meldingen, g=2))+1

inspectie_ARBO_gemeente$groep_ARBO <-
as.numeric(Hmisc::cut2(inspectie_ARBO_gemeente$ARBO_meldingen, g=3))
inspectie_AMF_gemeente$groep_AMF <-
as.numeric(Hmisc::cut2(inspectie_AMF_gemeente$AMF_meldingen, g=3))

############ gecombineerde bestanden #############

PC4_tot <- voorspellend_KOT_PC4 %>%
  rename(aantal_KOT = aantal_risicoselectie, perc_KOT =
percentage_risico, bak_KOT = bak) %>%
  full_join(voorspellend_HT_PC4) %>%
  mutate(PC4 = as.numeric(PC4)) %>%
  full_join(voorspellend_SVB_PC4, by = c("code" = "CODE_GEM",
"gemeentenaam", "PC4" = "PC_NUMM")) %>%
  full_join(historisch_SVB_PC4, by = c("code", "gemeentenaam", "PC4" =
"PC_NUMM")) %>%
  full_join(uwv_pc4 %>% select(-gemeentecode), by = c("gemeentenaam",
"PC4" = "postcode")) %>%
  rename(bak_ww_voorsp = groep_voorsp, bak_ww_hist = groep_hist) %>%
  full_join(ind_gefingeerd_pc4, by = c("code", "gemeentenaam", "PC4"))
%>%
  rename(Risico_gef_die = Risico) %>%
  full_join(ind_ID_pc4, by = c("code", "gemeentenaam", "PC4")) %>%
  rename(Risico_ID = Risico) %>%
  full_join(inspectie_ARBO, by = c("PC4","code", "gemeentenaam")) %>%
  full_join(inspectie_AMF, by = c("PC4","code", "gemeentenaam")) %>%
  filter(PC4 != 0,
         !is.na(code)) %>%
  unique()
```

```
PC4_tot$aantal_KOT[is.na(PC4_tot$aantal_KOT)] <- 0
PC4_tot$perc_KOT[is.na(PC4_tot$perc_KOT)] <- 0
PC4_tot$bak_KOT[is.na(PC4_tot$bak_KOT)] <- 1
PC4_tot$aantal_HT[is.na(PC4_tot$aantal_HT)] <- 0
PC4_tot$perc_HT[is.na(PC4_tot$perc_HT)] <- 0
PC4_tot$bak_HT[is.na(PC4_tot$bak_HT)] <- 1

PC4_tot$AANTAL_AOW[is.na(PC4_tot$AANTAL_AOW)] <- 0
PC4_tot$risico_aow_voor[is.na(PC4_tot$risico_aow_voor)] <- 0
PC4_tot$bak_aow_voor[is.na(PC4_tot$bak_aow_voor)] <- 1
PC4_tot$AANTAL_ANW[is.na(PC4_tot$AANTAL_ANW)] <- 0
PC4_tot$risico_anw_voor[is.na(PC4_tot$risico_anw_voor)] <- 0
PC4_tot$bak_anw_voor[is.na(PC4_tot$bak_anw_voor)] <- 1
PC4_tot$AANTAL_AIO[is.na(PC4_tot$AANTAL_AIO)] <- 0
PC4_tot$risico_aio_voor[is.na(PC4_tot$risico_aio_voor)] <- 0
PC4_tot$bak_aio_voor[is.na(PC4_tot$bak_aio_voor)] <- 1
PC4_tot$risico_aow_hist[is.na(PC4_tot$risico_aow_hist)] <- 0
PC4_tot$bak_aow_hist[is.na(PC4_tot$bak_aow_hist)] <- 1
PC4_tot$risico_anw_hist[is.na(PC4_tot$risico_anw_hist)] <- 0
PC4_tot$bak_anw_hist[is.na(PC4_tot$bak_anw_hist)] <- 1
PC4_tot$risico_aio_hist[is.na(PC4_tot$risico_aio_hist)] <- 0
PC4_tot$bak_aio_hist[is.na(PC4_tot$bak_aio_hist)] <- 1

PC4_tot$ScoreHistorisch[is.na(PC4_tot$ScoreHistorisch)] <- 0
PC4_tot$bak_ww_hist[is.na(PC4_tot$bak_ww_hist)] <- 1
PC4_tot$ScoreModel[is.na(PC4_tot$ScoreModel)] <- 0
PC4_tot$bak_ww_voorsp[is.na(PC4_tot$bak_ww_voorsp)] <- 1

PC4_tot$Risico_gef_die[is.na(PC4_tot$Risico_gef_die)] <- 0
PC4_tot$groep_gef[is.na(PC4_tot$groep_gef)] <- 1
PC4_tot$Risico_ID[is.na(PC4_tot$Risico_ID)] <- 0
PC4_tot$groep_ID[is.na(PC4_tot$groep_ID)] <- 1

PC4_tot$ARBO_meldingen[is.na(PC4_tot$ARBO_meldingen)] <- 0
PC4_tot$groep_ARBO[is.na(PC4_tot$groep_ARBO)] <- 1
PC4_tot$AMF_meldingen[is.na(PC4_tot$AMF_meldingen)] <- 0
PC4_tot$groep_AMF[is.na(PC4_tot$groep_AMF)] <- 1

```
En wegschrijven resultaat
```{r}

pad_file <- pad_bestand(pad_data, "PC4_tot_2019-07-02.rds")

saveRDS(PC4_tot, pad_file)
```


Gemeente-niveau

```{r, echo = FALSE, warning=FALSE}

# data combineren
bd_ht <- voorspellend_HT_gemeente %>%
  select(name = gemeentenaam, risico_ht_voor = percentage_risico,
bak_ht_voor = bak)
bd_kot <- voorspellend_KOT_gemeente %>%
```

```
  select(name = gemeentenaam, risico_kot_voor = percentage_risico,
bak_kot_voor = bak)
uwv_ww <- uwv_gemeente %>%
  select(name = gemeentenaam, risico_ww_voor = ScoreModel, bak_ww_voor =
groep_voorsp,
         risico_ww_hist = ScoreHistorisch, bak_ww_hist = groep_hist)

# 1 bestand maken met alle wetten
totaalbestand <- voorspellend_SVB_gemeente %>%
  select(name = gemeentenaam,
         risico_aow_voor = Risico_AOW,
         risico_anw_voor = Risico_ANW,
         risico_aio_voor = Risico_AIO,
         groepAOW_voor = groepAOW,
         groepANW_voor = groepANW,
         groepAIO_voor = groepAIO,
         groepTotaal_voor = groepTotaal) %>%
  left_join(bd_ht) %>%
  left_join(bd_kot) %>%
  left_join(uwv_ww) %>%
  left_join(historisch_SVB_gemeente %>%
              select(name = gemeentenaam,
                     risico_aow_hist =
RELATIEF_AANTAL_TERUGVORDERINGEN_AOW,
                     risico_anw_hist =
RELATIEF_AANTAL_TERUGVORDERINGEN_ANW,
                     risico_aio_hist =
RELATIEF_AANTAL_TERUGVORDERINGEN_AIO,
                     groepAOW_hist = groepAOW,
                     groepANW_hist = groepANW,
                     groepAIO_hist = groepAIO,
                     groepTotaal_hist = groepTotaal)) %>%
  left_join(historisch_HT_gemeente %>%
  select(name = gemeentenaam, risico_ht_hist = aantal_risicopunten,
bak_ht_hist = bak)) %>%
  left_join(historisch_KOT_gemeente %>%
  select(name = gemeentenaam, risico_kot_hist = aantal_risicopunten,
bak_kot_hist = bak)) %>%
  left_join(ind_gefingeerd_gemeente %>%
  select(name = gemeentenaam, risico_gef_hist = Risico, bak_gef_hist =
groep_gef)) %>%
  left_join(ind_ID_gemeente %>%
  select(name = gemeentenaam, risico_ID_hist = Risico, bak_ID_hist =
groep_ID)) %>%
  left_join(inspectie_ARBO_gemeente %>%
              select(name = gemeentenaam, ARBO_meldingen, groep_ARBO))
%>%
  left_join(inspectie_AMF_gemeente %>%
              select(name = gemeentenaam, AMF_meldingen, groep_AMF))


totaalbestand$risico_kot_voor[is.na(totaalbestand$risico_kot_voor)] <- 0
totaalbestand$bak_kot_voor[is.na(totaalbestand$bak_kot_voor)] <- 1
totaalbestand$risico_kot_hist[is.na(totaalbestand$risico_kot_hist)] <- 0
totaalbestand$bak_kot_hist[is.na(totaalbestand$bak_kot_hist)] <- 1
totaalbestand$risico_ht_voor[is.na(totaalbestand$risico_ht_voor)] <- 0
totaalbestand$bak_ht_voor[is.na(totaalbestand$bak_ht_voor)] <- 1
totaalbestand$risico_ht_hist[is.na(totaalbestand$risico_ht_hist)] <- 0
```

```
totaalbestand$bak_ht_hist[is.na(totaalbestand$bak_ht_hist)] <- 1

totaalbestand$risico_aow_voor[is.na(totaalbestand$risico_aow_voor)] <- 0
totaalbestand$groepAOW_voor[is.na(totaalbestand$groepAOW_voor)] <- 1
totaalbestand$risico_aow_hist[is.na(totaalbestand$risico_aow_hist)] <- 0
totaalbestand$groepAOW_hist[is.na(totaalbestand$groepAOW_hist)] <- 1
totaalbestand$risico_anw_voor[is.na(totaalbestand$risico_anw_voor)] <- 0
totaalbestand$groepANW_voor[is.na(totaalbestand$groepANW_voor)] <- 1
totaalbestand$risico_anw_hist[is.na(totaalbestand$risico_anw_hist)] <- 0
totaalbestand$groepANW_hist[is.na(totaalbestand$groepANW_hist)] <- 1
totaalbestand$risico_aio_voor[is.na(totaalbestand$risico_aio_voor)] <- 0
totaalbestand$groepAIO_voor[is.na(totaalbestand$groepAIO_voor)] <- 1
totaalbestand$risico_aio_hist[is.na(totaalbestand$risico_aio_hist)] <- 0
totaalbestand$groepAIO_hist[is.na(totaalbestand$groepAIO_hist)] <- 1

totaalbestand$risico_ww_voor[is.na(totaalbestand$risico_ww_voor)] <- 0
totaalbestand$bak_ww_voor[is.na(totaalbestand$bak_ww_voor)] <- 1
totaalbestand$risico_ww_hist[is.na(totaalbestand$risico_ww_hist)] <- 0
totaalbestand$bak_ww_hist[is.na(totaalbestand$bak_ww_hist)] <- 1

totaalbestand$risico_gef_hist[is.na(totaalbestand$risico_gef_hist)] <- 0
totaalbestand$bak_gef_hist[is.na(totaalbestand$bak_gef_hist)] <- 1
totaalbestand$risico_ID_hist[is.na(totaalbestand$risico_ID_hist)] <- 0
totaalbestand$bak_ID_hist[is.na(totaalbestand$bak_ID_hist)] <- 1

totaalbestand$ARBO_meldingen[is.na(totaalbestand$ARBO_meldingen)] <- 0
totaalbestand$groep_ARBO[is.na(totaalbestand$groep_ARBO)] <- 1
totaalbestand$AMF_meldingen[is.na(totaalbestand$AMF_meldingen)] <- 0
totaalbestand$groep_AMF[is.na(totaalbestand$groep_AMF)] <- 1

```


Wegschrijven bestand met gemeente-data
```{r}

pad_file <- pad_bestand(pad_data, "Gemeente_tot_2019-07-02.rds")

saveRDS(totaalbestand, pad_file)
```

````
---
title: "LSI"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
runtime: shiny
self_contained: no
---
```{r}

# Deze LSI_kaart is gemaakt door 5.1,2,E
# Het maken van deze kaart is een voortdurend project geweest, gestart in
begin 2017.
#
# Het doel van deze kaart is om op gemeente en PC4 niveau een gezamenlijk
risico te laten zien van verschillende overheidsinstanties.
#
# Op dit moment zijn de volgende overheidsinstanties aangesloten met
daarachter hun wet/thema:
# SVB (AOW, ANW, AIO)
# Belastingdienst Rood (Huurtoeslag (HT), Kinderopvangtoeslang (KOT))
# UWV (WW)
# IND (ID fraude en gefingeerde dienstverbanden)
# inspectie SZW (AMF meldingen en ARBO meldingen).
#
# Het uiteindelijke doel is dat deze kaart gebruikt kan worden in LSI
verband. Hiermee kan beter worden beargumenteerd waar de LSI zich op gaat
richten.

Sys.setenv(http_proxy =    'http://webproxy.frd.shsdir.nl:8080/')
Sys.setenv(https_proxy = ' https://webproxy.frd.shsdir.nl:8443/')
options(java.parameters = "- Xmx8096m")

library(dplyr)
library(data.table)
library(knitr)
library(leaflet)
library(shiny)
library(readxl)
library(rgdal)
library(rgeos)
library(flexdashboard)

```



```{r global, include=FALSE}


#pad_data <- "H:/Mijn documenten/2019 Libra/Data/"
pad_data   <- "//client/G$/I-SZW/O&A/Data Science projecten/2019 Libra
Kaart/Data/"
#pad_data   <- "G:/I-SZW/O&A/Data Science projecten/2019 Libra
Kaart/Data/"

pad_bestand <- function(pad, bestand) {
  file <- paste0(pad, bestand)
````
````

```
  return(file) }


bestandsnaam <- "omzetten_gemCode_chloropleth.csv"
namen <- fread(pad_bestand(pad_data, bestandsnaam), colClasses =
"character")

bestandsnaam <- "Gemeentegrenzen_2018.shp"

pad_file <- paste0(pad_data, bestandsnaam)

states <- readOGR(
  dsn = pad_file,
  layer = "Gemeentegrenzen_2018", GDAL1_integer64_policy = TRUE)

states <- spTransform(states, CRS("+proj=longlat +datum=WGS84 +no_defs"))
# Conversie van datum-velden en #coordinaat systeem

# ESRI-2018-Gemeentegrenzen
# gevonden op:
# http://geoplaza.vu.nl/data/dataset/imergis-nl-geografische-open-data-
gis-bestanden/resource/cd285b00-c91a-4b26-a707-3e0183588fa9

dtstates <- states@data %>%
  data.table() %>%
  mutate(code2 = substr(code,3,6))

LSI_projecten <- read_xlsx(pad_bestand(pad_data, "LSI_projecten.xlsx"))
%>%
  rename(gemeentenaam = `Gemeente (m.u.v. centrumgemeenten)`) %>%
  inner_join(namen) %>%
  filter(code %in% dtstates$code2)

PC4_tot <- readRDS(pad_bestand(pad_data, "PC4_tot_2019-07-02.rds")) #
Bevat alle gegevens op PC4 niveau
totaalbestand <- readRDS(pad_bestand(pad_data, "Gemeente_tot_2019-07-
02.rds")) # Bevat alle gegevens op gemeente-niveau
```

Zelf samenstellen
==


```{r, echo = FALSE, warning=FALSE}
# De selecties. De historische gegevens staan aan het begin aangevinkt.
fluidRow(
  column(2, checkboxGroupInput("SVB_hist", "SVB historisch",
                               choices = c("AOW", "ANW", "AIO"), inline =
T)),
  column(2, checkboxGroupInput("BD_hist", "BD/T historisch",
                               choices = c("HT", "KOT"), inline = T)),
  column(2, checkboxGroupInput("UWV_hist", "UWV historisch",
                               choices = c("WW" = "WW"), inline = T)),
  column(3, checkboxGroupInput("IND", "IND",
                               choices = c("gefingeerde dienstverbanden"
= "gef_die",
                                           "ID fraude" = "ID_fr"), inline
= T,
```

```
                                selected = c("gef_die", "ID_fr"))),
  column(2, checkboxGroupInput("Inspectie", "Inspectie",
                                choices = c("ARBO",
                                            "AMF"), inline = T,
                                selected = c("ARBO", "AMF")))

)
fluidRow(
  column(2, checkboxGroupInput("SVB_voorsp", "SVB voorspellend",
                                choices = c("AOW", "ANW", "AIO"), inline =
T,
                                selected = c("AOW", "ANW", "AIO"))),
  column(2, checkboxGroupInput("BD_voorsp", "BD/T voorspellend",
                                choices = c("HT", "KOT"), inline = T,
                                selected = c("HT", "KOT"))),
  column(2, checkboxGroupInput("UWV_voorsp", "UWV voorspellend",
                                choices = c("WW"), inline = T,
                                selected = c("WW" = "WW"))),
  column(2, actionButton("gen_kaart", "Genereer kaart")),
  column(2, checkboxInput('lsi_proj_zs', "Uitgevoerde LSI projecten",
value = F))
)

```


Row

Per gemeente

```{r, echo=FALSE, warning=FALSE}
leafletOutput("Map_zelf")

# Zelf samenstellen
# Op basis van input de data bepalen voor de kaartjes.


# Van groen naar donkergroen naar oranje naar rood
colors <- colorRampPalette(c("Green", "#7FD200", "Orange", "Red"))


# In renderleaflet alleen de basiskaart laden. Verder nog niks doen.
output$Map_zelf <- renderLeaflet({
  leaflet(states) %>%
    addProviderTiles(provider="Esri") %>%
#    addProviderTiles(providers$CartoDB.Positron) %>%
    addPolygons()
})

# als ze op de button klikken (genereer kaart), dan wordt op basis van de
aangevinkte vakjes data gebruikt.
observeEvent(input$gen_kaart, {
  totaalbestand2 <- NA

  # totale risicoscore is 1 * elke bak/groep die aangekruist is.
```

```
  totaalbestand2 <- totaalbestand %>%
    mutate(totaalbak = 1) %>%
    mutate(historisch_project = ifelse(name %in%
LSI_projecten$gemeentenaam, 1, 0))

  #SVB
  if("AOW" %in% input$SVB_hist){
    totaalbestand2$totaalbak <-
totaalbestand2$groepAOW_hist*totaalbestand2$totaalbak
  }
  if("ANW" %in% input$SVB_hist){
    totaalbestand2$totaalbak <-
totaalbestand2$groepANW_hist*totaalbestand2$totaalbak
  }
  if("AIO" %in% input$SVB_hist){
    totaalbestand2$totaalbak <-
totaalbestand2$groepAIO_hist*totaalbestand2$totaalbak
  }
  if("AOW" %in% input$SVB_voorsp){
    totaalbestand2$totaalbak <-
totaalbestand2$groepAOW_voor*totaalbestand2$totaalbak
  }
  if("ANW" %in% input$SVB_voorsp){
    totaalbestand2$totaalbak <-
totaalbestand2$groepANW_voor*totaalbestand2$totaalbak
  }
  if("AIO" %in% input$SVB_voorsp){
    totaalbestand2$totaalbak <-
totaalbestand2$groepAIO_voor*totaalbestand2$totaalbak
  }
  #BD
  if("HT" %in% input$BD_hist){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_ht_hist*totaalbestand2$totaalbak
  }
  if("KOT" %in% input$BD_hist){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_kot_hist*totaalbestand2$totaalbak
  }
  if("HT" %in% input$BD_voorsp){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_ht_voor*totaalbestand2$totaalbak
  }
  if("KOT" %in% input$BD_voorsp){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_kot_voor*totaalbestand2$totaalbak
  }
  #WW
  if("WW" %in% input$UWV_hist){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_ww_hist*totaalbestand2$totaalbak
  }
  if("WW" %in% input$UWV_voorsp){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_ww_voor*totaalbestand2$totaalbak
  }
  #IND
  if("gef_die" %in% input$IND){
```

```
    totaalbestand2$totaalbak <-
totaalbestand2$bak_gef_hist*totaalbestand2$totaalbak
  }
  if("ID_fr" %in% input$IND){
    totaalbestand2$totaalbak <-
totaalbestand2$bak_ID_hist*totaalbestand2$totaalbak
  }
  #inspectie
  if("ARBO" %in% input$Inspectie){
    totaalbestand2$totaalbak <-
totaalbestand2$groep_ARBO*totaalbestand2$totaalbak
  }
  if("AMF" %in% input$Inspectie){
    totaalbestand2$totaalbak <-
totaalbestand2$groep_AMF*totaalbestand2$totaalbak
  }

  # t representeert alle mogelijke waarden van de totaalbak (dus alle
scores)
  t <- NA
  t <- sort(unique(totaalbestand2$totaalbak))

    # maak data.frame met de combinatie tussen bak_tot en de bijbehorende
kleur.
    kleur <- as.data.frame(t) %>%
      cbind(as.data.frame(colors(length(t)))) %>%
      rename(totaalbak = t, color = `colors(length(t))`) %>%
      mutate(color = as.character(color))

    # kleur bij de temp doen
    totaalbestand2 <- inner_join(totaalbestand2, kleur)


compleet <- left_join(dtstates, totaalbestand2, by = c("gemeentena" =
"name")) %>%
  mutate(historisch_project = ifelse(is.na(historisch_project), 0,
historisch_project))

states_voor <- states
states_voor$risico_aow_voor <- compleet$risico_aow_voor
states_voor$risico_anw_voor <- compleet$risico_anw_voor
states_voor$risico_aio_voor <- compleet$risico_aio_voor
states_voor$risico_ht_voor <- compleet$risico_ht_voor
states_voor$risico_kot_voor <- compleet$risico_kot_voor
states_voor$risico_ww_voor <- compleet$risico_ww_voor

states_voor$risico_aow_hist <- compleet$risico_aow_hist
states_voor$risico_anw_hist <- compleet$risico_anw_hist
states_voor$risico_aio_hist <- compleet$risico_aio_hist
states_voor$risico_ht_hist <- compleet$risico_ht_hist
states_voor$risico_kot_hist <- compleet$risico_kot_hist
states_voor$risico_ww_hist <- compleet$risico_ww_hist

states_voor$risico_gef_hist <- compleet$risico_gef_hist
states_voor$risico_ID_hist <- compleet$risico_ID_hist
states_voor$ARBO_meldingen <- compleet$ARBO_meldingen
states_voor$AMF_meldingen <- compleet$AMF_meldingen
```

```
states_voor$groepAOW_voor <- compleet$groepAOW_voor
states_voor$groepANW_voor <- compleet$groepANW_voor
states_voor$groepAIO_voor <- compleet$groepAIO_voor
states_voor$bak_ht_voor <- compleet$bak_ht_voor
states_voor$bak_kot_voor <- compleet$bak_kot_voor
states_voor$bak_ww_voor <- compleet$bak_ww_voor

states_voor$groepAOW_hist <- compleet$groepAOW_hist
states_voor$groepANW_hist <- compleet$groepANW_hist
states_voor$groepAIO_hist <- compleet$groepAIO_hist
states_voor$bak_ht_hist <- compleet$bak_ht_hist
states_voor$bak_kot_hist <- compleet$bak_kot_hist
states_voor$bak_ww_hist <- compleet$bak_ww_hist

states_voor$bak_gef_hist <- compleet$bak_gef_hist
states_voor$bak_ID_hist <- compleet$bak_ID_hist
states_voor$groep_ARBO <- compleet$groep_ARBO
states_voor$groep_AMF <- compleet$groep_AMF

states_voor$totaalbak <- compleet$totaalbak
states_voor$color <- compleet$color
states_voor$historisch_project <- compleet$historisch_project

# De al aangemaakte Map_zelf (bij renderleaflet), wordt nu geupdatet met
de hiervoor verwerkte gegevens.
# Eerst alle data eraf gooien, en dan nieuwe toevoegen.
# Popup ook op basis van aangevinkte vakjes laten zien.
#
# als LSI projecten niet staat aangekruisd, dan normaal kaartje (zie
hieronder).
if(input$lsi_proj_zs == F){
  leafletProxy("Map_zelf") %>%
    clearShapes() %>%
    clearControls() %>%
    addProviderTiles(provider="Esri") %>%
    addPolygons(data = states_voor,
                color = "#444444",
                weight = 1,
                smoothFactor = 0.5,
                layerId=states_voor$gemeentena,
                opacity = 1.0, fillOpacity = 0.5,
                fillColor = states_voor$color,
                highlightOptions = highlightOptions(color = "white",
weight = 2,
                                                    bringToFront = TRUE),
                popup = paste("Gemeente: ", states_voor$gemeentena,
"</br>",
                              if("AOW" %in% input$SVB_hist){
paste("Risico AOW hist:", round(states_voor$risico_aow_hist, 2),
                                " (", states_voor$groepAOW_hist, ")",
"</br>")},
                              if("ANW" %in% input$SVB_hist){
paste("Risico ANW hist:", round(states_voor$risico_anw_hist, 2),
                                " (", states_voor$groepANW_hist, ")",
"</br>")},
                              if("AIO" %in% input$SVB_hist){
paste("Risico AIO hist:", round(states_voor$risico_aio_hist, 2),
```

```
                                  " (", states_voor$groepAIO_hist, ")",
"</br>")},
                                if("AOW" %in% input$SVB_voorsp){
paste("Risico AOW voorsp:", round(states_voor$risico_aow_voor, 2),
                                  " (", states_voor$groepAOW_voor, ")",
"</br>")},
                                if("ANW" %in% input$SVB_voorsp){
paste("Risico ANW voorsp:", round(states_voor$risico_anw_voor, 2),
                                  " (", states_voor$groepANW_voor, ")",
"</br>")},
                                if("AIO" %in% input$SVB_voorsp){
paste("Risico AIO voorsp:", round(states_voor$risico_aio_voor, 2),
                                  " (", states_voor$groepAIO_voor, ")",
"</br>")},

if("HT" %in% input$BD_hist){
paste("Risico HT hist:", round(states_voor$risico_ht_hist, 2),
                                  " (", states_voor$bak_ht_hist, ")",
"</br>")},
                                if("KOT" %in% input$BD_hist){
paste("Risico KOT hist:", round(states_voor$risico_ht_hist, 2),
                                  " (", states_voor$bak_kot_hist, ")",
"</br>")},
                                if("HT" %in% input$BD_voorsp){
paste("Risico HT voorsp:", round(states_voor$risico_ht_voor, 2),
                                  " (", states_voor$bak_ht_voor, ")",
"</br>")},
                                if("KOT" %in% input$BD_voorsp){
paste("Risico KOT voorsp:", round(states_voor$risico_kot_voor, 2),
                                  " (", states_voor$bak_kot_voor, ")",
"</br>")},

if("WW" %in% input$UWV_hist){
paste("Risico WW hist:", round(states_voor$risico_ww_hist, 4),
                                  " (", states_voor$bak_ww_hist, ")",
"</br>")},
                                if("WW" %in% input$UWV_voorsp){
paste("Risico WW voorsp:", round(states_voor$risico_ww_voor, 4),
                                  " (", states_voor$bak_ww_voor, ")",
"</br>")},

if("gef_die" %in% input$IND){
paste("meldingen gefingeerde dienstverbanden:",
round(states_voor$risico_gef_hist, 2),
                                  " (", states_voor$bak_gef_hist, ")",
"</br>")},
                                if("ID_fr" %in% input$IND){
paste("meldingen ID fraude:", round(states_voor$risico_ID_hist, 2),
                                  " (", states_voor$bak_ID_hist, ")",
"</br>")},

if("ARBO" %in% input$Inspectie){
paste("Meldingen ARBO:", round(states_voor$ARBO_meldingen, 2),
                                  " (", states_voor$groep_ARBO, ")",
"</br>")},
if("AMF" %in% input$Inspectie){
paste("Meldingen AMF:", round(states_voor$AMF_meldingen, 2),
```

```
                                " (", states_voor$groep_AMF, ")",
"</br>")},
paste("Totaal risico =", states_voor$totaalbak))) %>%
    addLegend(colors = rev(colors(5)),
              label = c("Hoog risico", "", "", "", "Laag risico"),
              title="Relatieve risicoscore gemeente")


  # Als LSI projecten wel staat aangekruisd, een ander kaartje maken
  } else {


  # Dit kaartje wordt gemaakt door 2 keer addpolygons te doen.
  # 1 keer voor de gemeenten zonder een LSI project in het verleden, en 1
keer voor de gemeenten met een LSI project.
  # Voor deze 2 addpolygons worden color en weight aangepast (dit zijn de
color en weight van de lijnen van de polygons).

#  !is.na(data_pg_df$Region)

  st_h0 <- states_voor[states_voor$historisch_project == 0,]
  st_h1 <- states_voor[states_voor$historisch_project == 1,]


  leafletProxy("Map_zelf") %>%
    clearShapes() %>%
    clearControls() %>%
    addProviderTiles(provider="Esri") %>%
    addPolygons(data = st_h0,
                color = "#444444",
                weight = 1,
                smoothFactor = 0.5,
                layerId=st_h0$gemeentena,
                opacity = 1.0, fillOpacity = 0.5,
                fillColor = st_h0$color,
                highlightOptions = highlightOptions(color = "white",
weight = 2,
                                                    bringToFront = TRUE),
                popup = paste("Gemeente: ", st_h0$gemeentena, "</br>",
                              if("AOW" %in% input$SVB_hist){
paste("Risico AOW hist:", round(st_h0$risico_aow_hist, 2),
                                " (", st_h0$groepAOW_hist, ")",
"</br>")},
                              if("ANW" %in% input$SVB_hist){
paste("Risico ANW hist:", round(st_h0$risico_anw_hist, 2),
                                " (", st_h0$groepANW_hist, ")",
"</br>")},
                              if("AIO" %in% input$SVB_hist){
paste("Risico AIO hist:", round(st_h0$risico_aio_hist, 2),
                                " (", st_h0$groepAIO_hist, ")",
"</br>")},
                              if("AOW" %in% input$SVB_voorsp){
paste("Risico AOW voorsp:", round(st_h0$risico_aow_voor, 2),
                                " (", st_h0$groepAOW_voor, ")",
"</br>")},
                              if("ANW" %in% input$SVB_voorsp){
paste("Risico ANW voorsp:", round(st_h0$risico_anw_voor, 2),
```

```
                                " (", st_h0$groepANW_voor, ")",
"</br>")},
                              if("AIO" %in% input$SVB_voorsp){
paste("Risico AIO voorsp:", round(st_h0$risico_aio_voor, 2),
                                " (", st_h0$groepAIO_voor, ")",
"</br>")},

if("HT" %in% input$BD_hist){
paste("Risico HT hist:", round(st_h0$risico_ht_hist, 2),
                                " (", st_h0$bak_ht_hist, ")", "</br>")},
                              if("KOT" %in% input$BD_hist){
paste("Risico KOT hist:", round(st_h0$risico_ht_hist, 2),
                                " (", st_h0$bak_kot_hist, ")", "</br>")},
                              if("HT" %in% input$BD_voorsp){
paste("Risico HT voorsp:", round(st_h0$risico_ht_voor, 2),
                                " (", st_h0$bak_ht_voor, ")", "</br>")},
                              if("KOT" %in% input$BD_voorsp){
paste("Risico KOT voorsp:", round(st_h0$risico_kot_voor, 2),
                                " (", st_h0$bak_kot_voor, ")", "</br>")},

if("WW" %in% input$UWV_hist){
paste("Risico WW hist:", round(st_h0$risico_ww_hist, 4),
                                " (", st_h0$bak_ww_hist, ")", "</br>")},
                              if("WW" %in% input$UWV_voorsp){
paste("Risico WW voorsp:", round(st_h0$risico_ww_voor, 4),
                                " (", st_h0$bak_ww_voor, ")", "</br>")},

if("gef_die" %in% input$IND){
paste("meldingen gefingeerde dienstverbanden:",
round(st_h0$risico_gef_hist, 2),
                                " (", st_h0$bak_gef_hist, ")", "</br>")},
                              if("ID_fr" %in% input$IND){
paste("meldingen ID fraude:", round(st_h0$risico_ID_hist, 2),
                                " (", st_h0$bak_ID_hist, ")", "</br>")},

if("ARBO" %in% input$Inspectie){
paste("Meldingen ARBO:", round(st_h0$ARBO_meldingen, 2),
                                " (", st_h0$groep_ARBO, ")", "</br>")},
if("AMF" %in% input$Inspectie){
paste("Meldingen AMF:", round(st_h0$AMF_meldingen, 2),
                                " (", st_h0$groep_AMF, ")", "</br>")},
paste("Totaal risico =", st_h0$totaalbak))) %>%
    addPolygons(data = st_h1,
                color = "#00bbff",
                weight = 4,
                smoothFactor = 0.5,
                layerId=st_h1$gemeentena,
                opacity = 1.0, fillOpacity = 0.5,
                fillColor = st_h1$color,
                highlightOptions = highlightOptions(color = "white",
weight = 2,
                                                    bringToFront = TRUE),
                popup = paste("Gemeente: ", st_h1$gemeentena, "</br>",
                              if("AOW" %in% input$SVB_hist){
paste("Risico AOW hist:", round(st_h1$risico_aow_hist, 2),
                                " (", st_h1$groepAOW_hist, ")",
"</br>")},
                              if("ANW" %in% input$SVB_hist){
```

```
paste("Risico ANW hist:", round(st_h1$risico_anw_hist, 2),
                                " (", st_h1$groepANW_hist, ")",
"</br>")},
                              if("AIO" %in% input$SVB_hist){
paste("Risico AIO hist:", round(st_h1$risico_aio_hist, 2),
                                " (", st_h1$groepAIO_hist, ")",
"</br>")},
                              if("AOW" %in% input$SVB_voorsp){
paste("Risico AOW voorsp:", round(st_h1$risico_aow_voor, 2),
                                " (", st_h1$groepAOW_voor, ")",
"</br>")},
                              if("ANW" %in% input$SVB_voorsp){
paste("Risico ANW voorsp:", round(st_h1$risico_anw_voor, 2),
                                " (", st_h1$groepANW_voor, ")",
"</br>")},
                              if("AIO" %in% input$SVB_voorsp){
paste("Risico AIO voorsp:", round(st_h1$risico_aio_voor, 2),
                                " (", st_h1$groepAIO_voor, ")",
"</br>")},

if("HT" %in% input$BD_hist){
paste("Risico HT hist:", round(st_h1$risico_ht_hist, 2),
                                " (", st_h1$bak_ht_hist, ")", "</br>")},
                              if("KOT" %in% input$BD_hist){
paste("Risico KOT hist:", round(st_h1$risico_ht_hist, 2),
                                " (", st_h1$bak_kot_hist, ")", "</br>")},
                              if("HT" %in% input$BD_voorsp){
paste("Risico HT voorsp:", round(st_h1$risico_ht_voor, 2),
                                " (", st_h1$bak_ht_voor, ")", "</br>")},
                              if("KOT" %in% input$BD_voorsp){
paste("Risico KOT voorsp:", round(st_h1$risico_kot_voor, 2),
                                " (", st_h1$bak_kot_voor, ")", "</br>")},

if("WW" %in% input$UWV_hist){
paste("Risico WW hist:", round(st_h1$risico_ww_hist, 4),
                                " (", st_h1$bak_ww_hist, ")", "</br>")},
                              if("WW" %in% input$UWV_voorsp){
paste("Risico WW voorsp:", round(st_h1$risico_ww_voor, 4),
                                " (", st_h1$bak_ww_voor, ")", "</br>")},

if("gef_die" %in% input$IND){
paste("meldingen gefingeerde dienstverbanden:",
round(st_h1$risico_gef_hist, 2),
                                " (", st_h1$bak_gef_hist, ")", "</br>")},
                              if("ID_fr" %in% input$IND){
paste("meldingen ID fraude:", round(st_h1$risico_ID_hist, 2),
                                " (", st_h1$bak_ID_hist, ")", "</br>")},

if("ARBO" %in% input$Inspectie){
paste("Meldingen ARBO:", round(st_h1$ARBO_meldingen, 2),
                                " (", st_h1$groep_ARBO, ")", "</br>")},
if("AMF" %in% input$Inspectie){
paste("Meldingen AMF:", round(st_h1$AMF_meldingen, 2),
                                " (", st_h1$groep_AMF, ")", "</br>")},
paste("Totaal risico =", st_h1$totaalbak))) %>%
    addLegend(colors = rev(colors(5)),
              label = c("Hoog risico", "", "", "", "Laag risico"),
              title="Relatieve risicoscore gemeente")
```

```
}
})

print("Na uitvoeren leaflet")
```


Zelf samenstellen Downdrill
==


```{r, echo = FALSE, warning=FALSE}

fluidRow(
  column(2, checkboxGroupInput("SVB_hist2", "SVB historisch",
                               choices = c("AOW", "ANW", "AIO"), inline =
T)),
  column(2, checkboxGroupInput("BD_hist2", "BD/T historisch",
                               choices = c("HT", "KOT"), inline = T)),
  column(2, checkboxGroupInput("UWV_hist2", "UWV historisch",
                               choices = c("WW" = "WW"), inline = T)),
  column(3, checkboxGroupInput("IND2", "IND",
                               choices = c("gefingeerde dienstverbanden"
= "gef_die",
                                           "ID fraude" = "ID_fr"), inline
= T,
                               selected = c("gef_die", "ID_fr"))),
  column(2, checkboxGroupInput("Inspectie2", "Inspectie",
                               choices = c("ARBO",
                                           "AMF"), inline = T,
                               selected = c("ARBO", "AMF")))

)
fluidRow(
  column(2, checkboxGroupInput("SVB_voorsp2", "SVB voorspellend",
                               choices = c("AOW", "ANW", "AIO"), inline =
T,
                               selected = c("AOW", "ANW", "AIO"))),
  column(2, checkboxGroupInput("BD_voorsp2", "BD/T voorspellend",
                               choices = c("HT", "KOT"), inline = T,
                               selected = c("HT", "KOT"))),
  column(2, checkboxGroupInput("UWV_voorsp2", "UWV voorspellend",
                               choices = c("WW"), inline = T,
                               selected = c("WW" = "WW"))),
  column(2, actionButton("gen_kaart2", "Genereer kaart"))
)


```


Row

Per gemeente

```{r, echo=FALSE, warning=FALSE, fig.width=400}
```

```
leafletOutput("Map_zelf2")

```

Ik ben nu NA Per gemeente in de downdrill

Per PC4

```{r, echo=FALSE, warning=FALSE, fig.width=400}

# Deze kaart is een combinatie van de bovenstaande 2 kaarten (downdrill
en zelf samenstellen).
# Het samenstellen van de kaart op gemeenteniveau is precies gelijk aan
zelf samenstellen.
# Het maken van de kaart op PC4 niveau is wel anders, omdat hier ook
rekening moet worden gehouden met de aangekruiste vakjes.

leafletOutput("Map_zelf2_dwn")


colors <- colorRampPalette(c("Green", "#7FD200", "Orange", "Red"))

output$Map_zelf2 <- renderLeaflet({
  leaflet(states) %>%
    addProviderTiles(provider="Esri") %>%
#   addProviderTiles(providers$CartoDB.Positron) %>%
    addPolygons()
})

output$Map_zelf2_dwn <- renderLeaflet({
  leaflet(states) %>%
    addProviderTiles(provider="Esri") %>%
    setView(lng = 6.2, lat = 52.2, zoom = 9)
})

observeEvent(input$gen_kaart2, {
  totaalbestand3 <- totaalbestand %>%
    mutate(totaalbak = 1)
  #SVB
  if("AOW" %in% input$SVB_hist2){
    totaalbestand3$totaalbak <-
totaalbestand3$groepAOW_hist*totaalbestand3$totaalbak
  }
  if("ANW" %in% input$SVB_hist2){
    totaalbestand3$totaalbak <-
totaalbestand3$groepANW_hist*totaalbestand3$totaalbak
  }
  if("AIO" %in% input$SVB_hist2){
    totaalbestand3$totaalbak <-
totaalbestand3$groepAIO_hist*totaalbestand3$totaalbak
  }
  if("AOW" %in% input$SVB_voorsp2){
    totaalbestand3$totaalbak <-
totaalbestand3$groepAOW_voor*totaalbestand3$totaalbak
  }
  if("ANW" %in% input$SVB_voorsp2){
    totaalbestand3$totaalbak <-
totaalbestand3$groepANW_voor*totaalbestand3$totaalbak
```

```
  }
  if("AIO" %in% input$SVB_voorsp2){
    totaalbestand3$totaalbak <-
totaalbestand3$groepAIO_voor*totaalbestand3$totaalbak
  }
  #BD
  if("HT" %in% input$BD_hist2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_ht_hist*totaalbestand3$totaalbak
  }
  if("KOT" %in% input$BD_hist2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_kot_hist*totaalbestand3$totaalbak
  }
  if("HT" %in% input$BD_voorsp2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_ht_voor*totaalbestand3$totaalbak
  }
  if("KOT" %in% input$BD_voorsp2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_kot_voor*totaalbestand3$totaalbak
  }
  #WW
  if("WW" %in% input$UWV_hist2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_ww_hist*totaalbestand3$totaalbak
  }
  if("WW" %in% input$UWV_voorsp2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_ww_voor*totaalbestand3$totaalbak
  }
  #IND
  if("gef_die" %in% input$IND2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_gef_hist*totaalbestand3$totaalbak
  }
  if("ID_fr" %in% input$IND2){
    totaalbestand3$totaalbak <-
totaalbestand3$bak_ID_hist*totaalbestand3$totaalbak
  }
  #inspectie
  if("ARBO" %in% input$Inspectie2){
    totaalbestand3$totaalbak <-
totaalbestand3$groep_ARBO*totaalbestand3$totaalbak
  }
  if("AMF" %in% input$Inspectie2){
    totaalbestand3$totaalbak <-
totaalbestand3$groep_AMF*totaalbestand3$totaalbak
  }


  t2 <- sort(unique(totaalbestand3$totaalbak))

    # maak data.frame met de combinatie tussen bak_tot en de bijbehorende
kleur.
    kleur2 <- as.data.frame(t2) %>%
      cbind(as.data.frame(colors(length(t2)))) %>%
      rename(totaalbak = t2, color = `colors(length(t2))`) %>%
```

```
      mutate(color = as.character(color))

    # kleur bij de temp doen
    totaalbestand3 <- inner_join(totaalbestand3, kleur2)


compleet <- left_join(dtstates, totaalbestand3, by = c("gemeentena" =
"name"))

states_voor <- states
states_voor$risico_aow_voor <- compleet$risico_aow_voor
states_voor$risico_anw_voor <- compleet$risico_anw_voor
states_voor$risico_aio_voor <- compleet$risico_aio_voor
states_voor$risico_ht <- compleet$risico_ht
states_voor$risico_kot <- compleet$risico_kot
states_voor$risico_ww_voor <- compleet$risico_ww_voor

states_voor$risico_aow_hist <- compleet$risico_aow_hist
states_voor$risico_anw_hist <- compleet$risico_anw_hist
states_voor$risico_aio_hist <- compleet$risico_aio_hist
states_voor$risico_ht_hist <- compleet$risico_ht_hist
states_voor$risico_kot_hist <- compleet$risico_kot_hist
states_voor$risico_ww_hist <- compleet$risico_ww_hist

states_voor$risico_gef_hist <- compleet$risico_gef_hist
states_voor$risico_ID_hist <- compleet$risico_ID_hist
states_voor$ARBO_meldingen <- compleet$ARBO_meldingen
states_voor$AMF_meldingen <- compleet$AMF_meldingen

states_voor$groepAOW_voor <- compleet$groepAOW_voor
states_voor$groepANW_voor <- compleet$groepANW_voor
states_voor$groepAIO_voor <- compleet$groepAIO_voor
states_voor$bak_ht_voor <- compleet$bak_ht_voor
states_voor$bak_kot_voor <- compleet$bak_kot_voor
states_voor$bak_ww_voor <- compleet$bak_ww_voor

states_voor$groepAOW_hist <- compleet$groepAOW_hist
states_voor$groepANW_hist <- compleet$groepANW_hist
states_voor$groepAIO_hist <- compleet$groepAIO_hist
states_voor$bak_ht_hist <- compleet$bak_ht_hist
states_voor$bak_kot_hist <- compleet$bak_kot_hist
states_voor$bak_ww_hist <- compleet$bak_ww_hist

states_voor$bak_gef_hist <- compleet$bak_gef_hist
states_voor$bak_ID_hist <- compleet$bak_ID_hist
states_voor$groep_ARBO <- compleet$groep_ARBO
states_voor$groep_AMF <- compleet$groep_AMF

states_voor$totaalbak <- compleet$totaalbak
states_voor$color <- compleet$color

  leafletProxy("Map_zelf2") %>%
    clearControls() %>%
    addProviderTiles(provider="Esri") %>%
    clearShapes() %>%
    addPolygons(data = states_voor, color = "#444444", weight = 1,
smoothFactor = 0.5,
                layerId=~gemeentena,
```

```
                opacity = 1.0, fillOpacity = 0.5,
                fillColor = ~color,
                highlightOptions = highlightOptions(color = "white",
weight = 2,
                                                    bringToFront = TRUE))
%>%
    addProviderTiles(providers$CartoDB.Positron) %>%
    addLegend(colors = rev(colors(5)),
              label = c("Hoog risico", "", "", "", "Laag risico"),
              title="Relatieve risicoscore gemeente")

})

# Postcode 4 grenzen inlezen

bestandsnaam <- "ESRI-PC4-2015R1.shp"

pad_file <- paste0(pad_data, bestandsnaam)

pc4states <- readOGR(dsn=pad_file,
                     layer = "ESRI-PC4-2015R1", GDAL1_integer64_policy =
TRUE)
pc4states <- spTransform(pc4states, CRS("+proj=longlat +datum=WGS84
+no_defs"))

# gemeente_naar_postcode2 <-
fread(pad_bestand(pad_data,"gemeente_naar_postcode.csv"))


observeEvent(input$Map_zelf2_shape_click, {
  temp2 <- NA
  temp2 <- PC4_tot
    # filter het reactive bestand wat in principe alle postcodes met
risico's bevat, naar de geselecteerde gemeente op de linker kaart.
    temp2 <- temp2 %>%
      filter(gemeentenaam == input$Map_zelf2_shape_click$id) %>%
      mutate(PC4 = as.numeric(PC4),
             bak_tot = 1)


  if("AOW" %in% input$SVB_hist2){
    temp2$bak_tot <- temp2$bak_aow_hist*temp2$bak_tot
  }
  if("ANW" %in% input$SVB_hist2){
    temp2$bak_tot <- temp2$bak_anw_hist*temp2$bak_tot
  }
  if("AIO" %in% input$SVB_hist2){
    temp2$bak_tot <- temp2$bak_aio_hist*temp2$bak_tot
  }
  if("AOW" %in% input$SVB_voorsp2){
    temp2$bak_tot <- temp2$bak_aow_voor*temp2$bak_tot
  }
  if("ANW" %in% input$SVB_voorsp2){
    temp2$bak_tot <- temp2$bak_anw_voor*temp2$bak_tot
  }
  if("AIO" %in% input$SVB_voorsp2){
    temp2$bak_tot <- temp2$bak_aio_voor*temp2$bak_tot
  }
```

```
  if("HT" %in% input$BD_voorsp2){
    temp2$bak_tot <- temp2$bak_HT*temp2$bak_tot
  }
  if("KOT" %in% input$BD_voorsp2){
    temp2$bak_tot <- temp2$bak_KOT*temp2$bak_tot
  }
  #WW
  if("WW" %in% input$UWV_hist2){
    temp2$bak_tot <- temp2$bak_ww_hist*temp2$bak_tot
  }
  if("WW" %in% input$UWV_voorsp2){
    temp2$bak_tot <- temp2$bak_ww_voorsp*temp2$bak_tot
  }
  #IND
  if("gef_die" %in% input$IND2){
    temp2$bak_tot <- temp2$groep_gef*temp2$bak_tot
  }
  if("ID_fr" %in% input$IND2){
    temp2$bak_tot <- temp2$groep_ID*temp2$bak_tot
  }
  #inspectie
  if("ARBO" %in% input$Inspectie2){
    temp2$bak_tot <- temp2$groep_ARBO*temp2$bak_tot
  }
  if("AMF" %in% input$Inspectie2){
    temp2$bak_tot <- temp2$groep_AMF*temp2$bak_tot
  }



    # aantal unieke bak_tot. Dit om het aantal kleuren te bepalen.
    t3 <- NA
    t3 <- sort(unique(temp2$bak_tot))

    # maak data.frame met de combinatie tussen bak_tot en de bijbehorende
kleur.
    kleur3 <- as.data.frame(t3) %>%
      cbind(as.data.frame(colors(length(t3)))) %>%
      rename(bak_tot = t3, color = `colors(length(t3))`)

    # kleur bij de temp2 doen
    temp2 <- inner_join(temp2, kleur3)

    substates <- subset(pc4states, pc4states$PC4 %in% temp2$PC4)

    dtstates4 <- substates@data %>% data.frame()
    dtstates4 <- left_join(dtstates4, temp2)
    substates$gemeentenaam <- dtstates4$gemeentenaam
    substates$aantal_KOT <- dtstates4$aantal_KOT
    substates$aantal_HT <- dtstates4$aantal_HT
    substates$AANTAL_AOW <- dtstates4$AANTAL_AOW
    substates$AANTAL_ANW <- dtstates4$AANTAL_ANW
    substates$AANTAL_AIO <- dtstates4$AANTAL_AIO
    substates$perc_KOT <- dtstates4$perc_KOT
    substates$perc_HT <- dtstates4$perc_HT
    substates$risico_aow_voor <- dtstates4$risico_aow_voor
    substates$risico_anw_voor <- dtstates4$risico_anw_voor
```

```
    substates$risico_aio_voor <- dtstates4$risico_aio_voor
    substates$risico_aow_hist <- dtstates4$risico_aow_hist
    substates$risico_anw_hist <- dtstates4$risico_anw_hist
    substates$risico_aio_hist <- dtstates4$risico_aio_hist
    substates$bak_KOT <- dtstates4$bak_KOT
    substates$bak_HT <- dtstates4$bak_HT
    substates$bak_aow_voor <- dtstates4$bak_aow_voor
    substates$bak_anw_voor <- dtstates4$bak_anw_voor
    substates$bak_aio_voor <- dtstates4$bak_aio_voor
    substates$bak_aow_hist <- dtstates4$bak_aow_hist
    substates$bak_anw_hist <- dtstates4$bak_anw_hist
    substates$bak_aio_hist <- dtstates4$bak_aio_hist
    substates$ScoreHistorisch <- dtstates4$ScoreHistorisch
    substates$ScoreModel <- dtstates4$ScoreModel
    substates$bak_ww_hist <- dtstates4$bak_ww_hist
    substates$bak_ww_voorsp <- dtstates4$bak_ww_voorsp
    substates$Risico_gef_die <- dtstates4$Risico_gef_die
    substates$groep_gef <- dtstates4$groep_gef
    substates$Risico_ID <- dtstates4$Risico_ID
    substates$groep_ID <- dtstates4$groep_ID
    substates$ARBO_meldingen <- dtstates4$ARBO_meldingen
    substates$groep_ARBO <- dtstates4$groep_ARBO
    substates$AMF_meldingen <- dtstates4$AMF_meldingen
    substates$groep_AMF <- dtstates4$groep_AMF
    substates$bak_tot <- dtstates4$bak_tot
    substates$color <- dtstates4$color

    # dit om de postcde cijfers in het midden van de polygon te plotten.
    centers <- data.frame(gCentroid(substates, byid = TRUE))
    centers$PC4 <- as.character(substates$PC4)

    #boundaries instellen
    lat1 <- NA
    lat2 <- NA
    lng1 <- NA
    lng2 <- NA
    for(i in 1:length(substates)){
      temp <- max(substates@polygons[[i]]@Polygons[[1]]@coords[,1], na.rm
= T)
      lng1 <- max(c(lng1, temp), na.rm = T)

      temp <- min(substates@polygons[[i]]@Polygons[[1]]@coords[,1], na.rm
= T)
      lng2 <- min(c(lng2, temp), na.rm = T)

      temp <- max(substates@polygons[[i]]@Polygons[[1]]@coords[,2], na.rm
= T)
      lat1 <- max(c(lat1, temp), na.rm = T)

      temp <- min(substates@polygons[[i]]@Polygons[[1]]@coords[,2], na.rm
= T)
      lat2 <- min(c(lat2, temp), na.rm = T)
    }

    leafletProxy("Map_zelf2_dwn") %>%
      clearControls() %>%
      clearShapes() %>%
      clearMarkers() %>%
```

```
      addProviderTiles(provider="Esri") %>%
      # addProviderTiles(providers$CartoDB.Positron) %>%
      addPolygons(data = substates, color = "#444444", weight = 1,
smoothFactor = 0.5,
                  opacity = 1, fillOpacity = 0.5,
                  fillColor = ~color,
                  highlightOptions = highlightOptions(color = "white",
weight = 2,
                                                      bringToFront =
TRUE),
                  popup = paste("Gemeente: ", substates$gemeentenaam,
"</br>",
                                "Postcode: ", substates$PC4, "</br>",
                                if("AOW" %in% input$SVB_hist2){
paste("Risico AOW hist:", round(substates$risico_aow_hist, 2),
                                " (", substates$bak_aow_hist, ")",
"</br>")},
                              if("ANW" %in% input$SVB_hist2){
paste("Risico ANW hist:", round(substates$risico_anw_hist, 2),
                                " (", substates$bak_anw_hist, ")",
"</br>")},
                              if("AIO" %in% input$SVB_hist2){
paste("Risico AIO hist:", round(substates$risico_aio_hist, 2),
                                " (", substates$bak_aio_hist, ")",
"</br>")},
                              if("AOW" %in% input$SVB_voorsp2){
paste("Risico AOW voorsp:", round(substates$risico_aow_voor, 2),
                                " (", substates$bak_aow_voor, ")",
"</br>")},
                              if("ANW" %in% input$SVB_voorsp2){
paste("Risico ANW voorsp:", round(substates$risico_anw_voor, 2),
                                " (", substates$bak_anw_voor, ")",
"</br>")},
                              if("AIO" %in% input$SVB_voorsp2){
paste("Risico AIO voorsp:", round(substates$risico_aio_voor, 2),
                                " (", substates$bak_aio_voor, ")",
"</br>")},
                              if("HT" %in% input$BD_voorsp2){
paste("Risico HT voorsp:", round(substates$perc_HT, 2),
                                " (", substates$bak_HT, ")", "</br>")},
                              if("KOT" %in% input$BD_voorsp2){
paste("Risico KOT voorsp:", round(substates$perc_KOT, 2),
                                " (", substates$bak_KOT, ")", "</br>")},

if("WW" %in% input$UWV_hist2){
paste("Risico WW hist:", round(substates$ScoreHistorisch, 4),
                                " (", substates$bak_ww_hist, ")",
"</br>")},
                              if("WW" %in% input$UWV_voorsp2){
paste("Risico WW voorsp:", round(substates$ScoreModel, 4),
                                " (", substates$bak_ww_voorsp, ")",
"</br>")},

if("gef_die" %in% input$IND2){
paste("meldingen gefingeerde dienstverbanden:",
round(substates$Risico_gef_die, 2),
                                " (", substates$groep_gef, ")",
"</br>")},
```

```
                                if("ID_fr" %in% input$IND2){
paste("meldingen ID fraude:", round(substates$Risico_ID, 2),
                                  " (", substates$groep_ID, ")", "</br>")},

if("ARBO" %in% input$Inspectie2){
paste("Meldingen ARBO:", round(substates$ARBO_meldingen, 2),
                                  " (", substates$groep_ARBO, ")",
"</br>")},
if("AMF" %in% input$Inspectie2){
paste("Meldingen AMF:", round(substates$AMF_meldingen, 2),
                                  " (", substates$groep_AMF, ")",
"</br>")},
                                "Gecombineerd Risico: ",
substates$bak_tot)) %>%
      addLabelOnlyMarkers(data = centers,
                          lng = ~x, lat = ~y, label = ~PC4,
                          labelOptions = labelOptions(noHide = TRUE,
direction = 'top', textOnly = TRUE)) %>%
      fitBounds(lng1 = lng1,
                lng2 = lng2,
                lat1 = lat1,
                lat2 = lat2)

  })

```
```
```

````
---
title: "002 Pre-processing PILS data 2019-07-17"
author: "5.1.2.E"
date: "27 juni 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Inlezen libraries
```{r echo=FALSE, message=FALSE}
library(tidyverse)
library(recipes)
```

Inlezen bestand met geschoonde data, t.b.v. verdere schoning met behulp
van recipes package.
```{r}
setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
Df <- read_rds("2019-08-16 Horeca_Na_pre-processing.rds")

```
Aantal variabelen doen niet meer mee in het modelleren en voorspellen,
wel belangrijk als referentie
```{r}
Df_Achtergrond <- Df %>%
  select(KvKnummer12, VestigingNaam, Postcode, Plaats, PClat, PClon)

Df <- Df %>%
  select(-KvKnummer12, -VestigingNaam, -Postcode, -Plaats, -PClat, -
PClon, -FamilieBedrijfsnaam)  # FamilieBedrijfsnaam Vind ik te slecht
afgeleid om te gebruiken

```

NA waarden van een aantal NVWA- en TWV-kolommen vervangen door 0
```{r}

Df2 <- Df %>%
  mutate(AantalNvwaZaken_AlleVestigingen           =
replace_na(AantalNvwaZaken_AlleVestigingen, replace = 0)) %>%
  mutate(AantalNvwaZakenOk_AlleVestigingen         =
replace_na(AantalNvwaZakenOk_AlleVestigingen, replace = 0)) %>%
  mutate(AantalNvwaZakenMetOvt_AlleVestigingen     =
replace_na(AantalNvwaZakenMetOvt_AlleVestigingen, replace = 0)) %>%
  mutate(AantalNvwaZaken_DezeVestiging             =
replace_na(AantalNvwaZaken_DezeVestiging, replace = 0)) %>%
  mutate(AantalNvwaZakenOk_DezeVestiging           =
replace_na(AantalNvwaZakenOk_DezeVestiging, replace = 0)) %>%
  mutate(AantalNvwaZakenMetOvt_DezeVestiging       =
replace_na(AantalNvwaZakenMetOvt_DezeVestiging, replace = 0)) %>%
  mutate(Aantal_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017     =
replace_na(Aantal_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017, replace =
0)) %>%
````
````

```
  mutate(Geweigerde_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017 =
replace_na(Geweigerde_TWV_Aanvragen_Alle_Vestigingen_Vanaf_2017, replace
= 0))


```

Alle UWV-data vervangen door 0, m.u.v. UWV_gemiddeld uurloon, vervangen door Uurloon 2015.
Het gemiddelde UWV loon vervangen we door het uurloon in 2016, indien missend.

```{r}
Df3 <- Df2 %>%
  mutate(UWV_gemiddeld_uurloon = ifelse(is.na(UWV_gemiddeld_uurloon),
Uurloon2015, UWV_gemiddeld_uurloon)) %>%
  mutate_at(vars(contains('UWV')), funs(replace_na(., replace = 0)))

```

Recept aanmaken. Hoog gecorreleerde variabelen, met een correlatie groter dan 0.7 worden verwijderd.
Infrequente categorieen van factor-variabelen worden samengevat in de categorie "Overige".
Van een aantal variabelen worden waardes die buiten de bandbreedte vallen teruggebracht naar een minimum c.q. maximum.

```{r recipe, echo=FALSE}

Recept_Obj <- recipe(ovt ~ ., data = Df2)

Filter <- Recept_Obj %>%
  step_corr(all_numeric(), threshold = .9) %>%    # Remove one of the
variables of variables correlated > 0.9
  step_other(all_nominal(), threshold = 0.05, other = "Overige")  %>% #
Cluster infrequent factor-levels
  step_dummy(Rechtsvorm, Gemeentegrootte, Stedelijkheid) %>%
# Dummify factor variables
  step_medianimpute(Perc_Uitkeringstrekkers_PC4,
                    Perc_Leegst_Woningen_Schoon_PC4,
#                   Perc_Personen_Laag_Inkomen_In_PC4, Verdwenen na
verwijderen hoog-gecorreleerde variabelen
                    perc_verloop_sinds_feb18,   perc_groei_sinds_feb18)
%>%
  step_range(AantalWerknemers, min = 0, max = 1000) %>%
  step_range(AantalOpAdres, min = 0, max = 20) %>%
  step_range(Uurloon2015, min = 12, max = 120) %>%
  step_range(UWV_gemiddeld_uurloon, min = 7, max = 120) %>%
  step_range(Aantal_postcode, min = 0, max = 1200) %>%
  step_range(JaarVestiging, min = 1800, max = 2020)

Filter_Obj <- prep(Filter, training = Df3)

Filtered_Df <- bake(Filter_Obj, Df3)

```
En nu de achtergrond variabelen er weer voor plakken, handig in de uitleg etc.
En geheel wegschrijven naar schijf

```{r}
Df_Final <- bind_cols(Df_Achtergrond, Filtered_Df)

setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
write_rds(Df_Final, "2019-12-24 Horeca_Na_Final_Pre-processing.rds")

```

````
---
title: "010 Modelling Ranger 2019-08-16, bestand met hoog-gecorreleerde
variabelen aanwezig"
author: "5.1.2.E"
date: "1 juli 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r, echo=FALSE, warning=FALSE}
library(tidyverse)
library(caret)
library(tictoc)
library(pROC)
library(recipes)
library(ranger)
library(tidytext)

```
Functies
```{r}

Precision_n <- function(y, kansen, n) {
  precision_frame <- data.frame(y)
  precision_frame <- mutate(precision_frame, V2=kansen)
  precision_frame_ordered <- arrange(precision_frame, desc(V2))
  precision_frame_top_n <- dplyr::slice(precision_frame_ordered, 1:n)
  precision_frame_top_n_correct <- filter(precision_frame_top_n, y==1)
  precision_n <- nrow(precision_frame_top_n_correct)
  return(precision_n)
}
```


```{r}
setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
Data1 <- readRDS("2019-08-16 Horeca_Na_Final2_Pre-processing.rds")

```

Weghalen van achtergrond variabelen
Ranger kan niet omgaan met missende waarden, daarom UWV-gegevens
verwijderen, m.u.v. het veld MISSING_UWV
```{r}
Data_Model <- Data1 %>%
  select(-(VestigingNaam:Plaats)) %>%
  select(-(UWV_aantal_werknemers:UWV_aantal_recs_nuluren_wel_loon)) %>%
  mutate(MISSING_UWV_gegevens = replace_na(MISSING_UWV_gegevens, replace
= "1"))

```

```{r}
````
````

```

Opsplitsen van de data in train- en test. En matrices maken van train, test en alle features
```{r}
set.seed(111)
trainIndex <- createDataPartition(Data_Model$ovt, p = .8,
                                  list = FALSE,
                                  times = 1)

Y_Train <- Data_Model$ovt[trainIndex]
X_Train <- Data_Model[trainIndex,]
X_Train <- select(X_Train, -ovt)

Y_Test  <- Data_Model$ovt[-trainIndex]
X_Test  <- Data_Model[-trainIndex,]
X_Test  <- select(X_Test, -ovt)

Features <- select(Data_Model, -ovt)

```

En Ranger model trainen

```{r}
tic()
set.seed(12)

Ranger_Model <- ranger(Y_Train ~ .,
                       data = X_Train,
                       importance = "impurity")

toc()
```

```{r}
saveRDS(Ranger_Model, "//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS//Models/2019-08-16 Ranger_Model.rds")
```

Show variable importance
```{r}
Var_Imp <- importance(Ranger_Model)

Var_names <- names(Var_Imp)

Var_Importance_12 <- tibble(Variabele = Var_names, Importance = Var_Imp) %>% arrange(desc(Importance)) %>% dplyr::slice(1:12)

ggplot(data = Var_Importance_12, aes(x = reorder(Variabele, Importance), y = Importance)) +
       geom_bar(stat = "identity", fill = "blue") +
       coord_flip() +
       xlab("Variabele") +
       theme_bw()

```
```

Show variable importance
```{r}
Var_Imp <- importance(ranger_model)

Var_names <- names(Var_Imp)

Var_Importance_12 <- tibble(Variabele = Var_names, Importance = Var_Imp)
%>% arrange(desc(Importance)) %>% dplyr::slice(1:12)

ggplot(data = Var_Importance_12, aes(x = reorder(Variabele, Importance),
y = Importance)) +
       geom_bar(stat = "identity", fill = "blue") +
       coord_flip() +
       xlab("Variabele") +
       theme_bw()

```

And make predictions on the testset

```{r}

Predicties_Ranger_Test <- predict(Ranger_Model, data = X_Test, type =
"response")


table(Y_Test)       # Ongeveer 0,6 procent overtreders
Predicties_Test <- tibble(Predicties_Ranger_Test$predictions)
names(Predicties_Test) <- "Predictie"
ggplot(data = Predicties_Test, aes(x = Predictie)) + geom_density()
```


```{r}
a <- Precision_n(Y_Test, Predicties_Ranger_Test$predictions, 100)
b <- Precision_n(Y_Test, Predicties_Ranger_Test$predictions, 200)
c <- Precision_n(Y_Test, Predicties_Ranger_Test$predictions, 300)
d <- Precision_n(Y_Test, Predicties_Ranger_Test$predictions, 400)
e <- Precision_n(Y_Test, Predicties_Ranger_Test$predictions, 500)

Precision_vector <- c(a, b, c, d, e)
plot(Precision_vector)
```

Nu de ROC afdrukken

```{r}

ROC1 <- roc(Y_Test, Predicties_Test$Predictie)

plot(ROC1, col = "blue", left_margin=NULL)

```

En de AUC op de testset
```{r}
AUC_testset <- pROC::auc(ROC1)
AUC_testset
```

```
```

````
---
title: "010 Modelling Xgboost 2019-07-02, bestand met hoog-gecorreleerde
variabelen aanwezig"
author: "5.1.2.E"
date: "1 juli 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r}
library(tidyverse)
library(xgboost)
library(caret)
library(tictoc)
library(pROC)
```
Functies
```{r}

Precision_n <- function(y, kansen, n) {
  precision_frame <- data.frame(y)
  precision_frame <- mutate(precision_frame, V2=kansen)
  precision_frame_ordered <- arrange(precision_frame, desc(V2))
  precision_frame_top_n <- dplyr::slice(precision_frame_ordered, 1:n)
  precision_frame_top_n_correct <- filter(precision_frame_top_n, y==1)
  precision_n <- nrow(precision_frame_top_n_correct)
  return(precision_n)
}
```


```{r}

Data2 <- readRDS( "\\\\Client\\G$\\I-SZW\\O&A\\Data Science
projecten\\2019 PILS\\Data\\2019-07-02_Horeca_Na_Final2_Pre-
processing.rds")

```

Weghalen van achtergrond variabelen
```{r}
Data_Model <- select(Data2, -(VestigingNaam:PClon))

```

Opsplitsen van de data in train- en test. En matrices maken van train,
test en alle features
```{r}
set.seed(111)
trainIndex <- createDataPartition(Data_Model$ovt, p = .8,
                                  list = FALSE,
                                  times = 1)

Y_Train <- Data_Model$ovt[trainIndex]
````
````

```
X_Train <- Data_Model[trainIndex,]
X_Train <- select(X_Train, -ovt) %>% data.matrix()

Y_Test  <- Data_Model$ovt[-trainIndex]
X_Test  <- Data_Model[-trainIndex,]
X_Test  <- select(X_Test, -ovt) %>% data.matrix()

Features <- select(Data_Model, -ovt)
Features.matrix <- Features %>% data.matrix()
```

En XGboost model trainen

```{r}
tic()
set.seed(12)

best_param <- list(objective = "binary:logistic",  # For regression
                eval_metric = "auc",      # rmse is used for regression
                max_depth = 4,
                eta = 0.3,   # Learning rate
                subsample = 0.8,
                colsample_bytree = 0.8,
                min_child_weight = 2,
                scale_pow_weight = sum(Y_Train == 0) / sum(Y_Train == 1),
                max_delta_step = 8)

Xgb_Model <- xgboost(data = X_Train, label = Y_Train, params =
best_param, nround = 200, verbose = F)

toc()
```
```{r}
saveRDS(Xgb_Model, "//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS//Models/2019-07-02 Xgb_Model2.rds")
```

Show variable importance
```{r}
Var_Imp <- xgb.importance(feature_names = NULL, model = Xgb_Model)

library(Ckmeans.1d.dp)

xgb.ggplot.importance(Var_Imp,
                      top_n = 12)

```
Show partial dependency plots in the shap way
```{r}

xgb.plot.shap(data = X_Train,
              model = Xgb_Model,
              top_n = 12,
              n_col = 3,
              ylab = "kans op AMF overtreding")
```

And make predictions on the testset

```{r}

Predicties_Xgb_Test <- predict(Xgb_Model, newdata = X_Test, type =
"prob")
Predicties_Xgb_Test_Class <- predict(Xgb_Model, newdata = X_Test, type =
"class")
Predicties_Xgb_Test_Class <- as.factor(ifelse(Predicties_Xgb_Test_Class >
0.5, 1, 0))
Y_Test_Class <-  as.factor(Y_Test)

#table(Y_Test)       # Ongeveer 1:6 is overtreder. Dat is 16%
Predicties_Test <- tibble(Predicties_Xgb_Test)
ggplot(data = Predicties_Test, aes(x = Predicties_Xgb_Test)) +
geom_density()
```

```{r}
a <- Precision_n(Y_Test, Predicties_Xgb_Test, 100) # Dat is 61%, dus
vervierdubbeling kans.
b <- Precision_n(Y_Test, Predicties_Xgb_Test, 200)
c <- Precision_n(Y_Test, Predicties_Xgb_Test, 300)
d <- Precision_n(Y_Test, Predicties_Xgb_Test, 400)
e <- Precision_n(Y_Test, Predicties_Xgb_Test, 500)

Precision_vector <- c(a, b, c, d, e)
plot(Precision_vector)
```

Nu de ROC afdrukken

```{r}

ROC1 <- roc(Y_Test, Predicties_Xgb_Test)

plot(ROC1, col = "blue", left_margin=NULL)

```

En de AUC op de testset
```{r}
AUC_testset <- pROC::auc(ROC1)
AUC_testset
```

Confusion matrix
```{r}

caret::confusionMatrix(data=Predicties_Xgb_Test_Class,
reference=Y_Test_Class, positive="1")

```
Grafische weergave resultaat
```{r}
```

```
observed <- as.numeric(Y_Test_Class) - 1

plot_pred_type_distribution <- function(df, threshold) {
  v <- rep(NA, nrow(df))
  v <- ifelse(df$fit >= threshold & df$class == 1, "TP", v)
  v <- ifelse(df$fit >= threshold & df$class == 0, "FP", v)
  v <- ifelse(df$fit < threshold & df$class == 1, "FN", v)
  v <- ifelse(df$fit < threshold & df$class == 0, "TN", v)

  df$fit_type <- v

  ggplot(data=df, aes(x=class, y=fit)) +
    geom_violin(fill=rgb(1,1,1,alpha=0.6), color=NA) +
    geom_jitter(aes(color=fit_type), alpha=0.6) +
    geom_hline(yintercept=threshold, color="red", alpha=0.6) +
    scale_color_discrete(name = "type") +
    labs(title=sprintf("Grenswaarde %.2f", threshold))+
    theme_bw()
}

df <- data.frame(class = observed, fit = Predicties_Xgb_Test)

plot_pred_type_distribution(df = df, threshold = 0.3)
```

```
---
title: "010 Modelling Xgboost 2019-07-17, bestand met hoog-gecorreleerde
variabelen aanwezig"
author: "5.1,2,E"
date: "28 november 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r}
library(tidyverse)
library(xgboost)
library(caret)
library(tictoc)
library(pROC)
```
Functies
```{r}

Precision_n <- function(y, kansen, n) {
  precision_frame <- data.frame(y)
  precision_frame <- mutate(precision_frame, V2=kansen)
  precision_frame_ordered <- arrange(precision_frame, desc(V2))
  precision_frame_top_n <- dplyr::slice(precision_frame_ordered, 1:n)
  precision_frame_top_n_correct <- filter(precision_frame_top_n, y==1)
  precision_n <- nrow(precision_frame_top_n_correct)
  return(precision_n)
}
```


```{r}

Data2 <- readRDS( "\\\\Client\\G$\\I-SZW\\O&A\\Data Science
projecten\\2019 PILS\\Data\\2019-07-17_Horeca_Na_Final2_Pre-
processing.rds")

```

Weghalen van achtergrond variabelen
```{r}
Data_Model <- select(Data2, -(VestigingNaam:PClon)) %>%
  mutate(KvKnummer12 = as.numeric(KvKnummer12))

```

Opsplitsen van de data in train- en test. En matrices maken van train,
test en alle features
```{r}
set.seed(111)
trainIndex <- createDataPartition(Data_Model$ovt, p = .8,
                                  list = FALSE,
                                  times = 1)
```
```

```
Y_Train <- Data_Model$ovt[trainIndex]
X_Train <- Data_Model[trainIndex,]
X_Train <- select(X_Train, -ovt) %>% data.matrix()

Y_Test  <- Data_Model$ovt[-trainIndex]
X_Test  <- Data_Model[-trainIndex,]
X_Test  <- select(X_Test, -ovt) %>% data.matrix()

Features <- select(Data_Model, -ovt)
Features.matrix <- Features %>% data.matrix()

```


En XGboost model trainen

```{r}
tic()
set.seed(12)

best_param <- list(objective = "binary:logistic",  # For regression
                eval_metric = "auc",      # rmse is used for regression
                max_depth = 4,
                eta = 0.3,   # Learning rate
                subsample = 0.8,
                colsample_bytree = 0.8,
                min_child_weight = 2,
                scale_pow_weight = sum(Y_Train == 0) / sum(Y_Train == 1),
                max_delta_step = 8)

Xgb_Model <- xgboost(data = X_Train, label = Y_Train, params =
best_param, nround = 200, verbose = F)

toc()
```
```{r}
saveRDS(Xgb_Model, "//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS//Models/2019-07-17 Xgb_Model2.rds")
```

Show variable importance
```{r}
Var_Imp <- xgb.importance(feature_names = NULL, model = Xgb_Model)

library(Ckmeans.1d.dp)

xgb.ggplot.importance(Var_Imp,
                      top_n = 12)

```
```{r}
xgb.ggplot.importance(Var_Imp,
                      top_n = 30)

```

Show partial dependency plots in the shap way
```{r}
```

```
xgb.plot.shap(data = X_Train,
              model = Xgb_Model,
              top_n = 20,
              n_col = 3,
              ylab = "kans op AMF overtreding")
```

And make predictions on the testset

```{r}

Predicties_Xgb_Test <- predict(Xgb_Model, newdata = X_Test, type =
"prob")
Predicties_Xgb_Test_Class <- predict(Xgb_Model, newdata = X_Test, type =
"class")
Predicties_Xgb_Test_Class <- as.factor(ifelse(Predicties_Xgb_Test_Class >
0.5, 1, 0))
Y_Test_Class <-  as.factor(Y_Test)

#table(Y_Test)       # Ongeveer 1:6 is overtreder. Dat is 16%
Predicties_Test <- tibble(Predicties_Xgb_Test)
ggplot(data = Predicties_Test, aes(x = Predicties_Xgb_Test)) +
geom_density()
```


```{r}
a <- Precision_n(Y_Test, Predicties_Xgb_Test, 100) # Dat is 61%, dus
vervierdubbeling kans.
b <- Precision_n(Y_Test, Predicties_Xgb_Test, 200)
c <- Precision_n(Y_Test, Predicties_Xgb_Test, 300)
d <- Precision_n(Y_Test, Predicties_Xgb_Test, 400)
e <- Precision_n(Y_Test, Predicties_Xgb_Test, 500)

Precision_vector <- c(a, b, c, d, e)
plot(Precision_vector)
```

Nu de ROC afdrukken

```{r}

ROC1 <- roc(Y_Test, Predicties_Xgb_Test)

plot(ROC1, col = "blue", left_margin=NULL)

```

En de AUC op de testset
```{r}
AUC_testset <- pROC::auc(ROC1)
AUC_testset
```

Confusion matrix
```{r}
```

```
caret::confusionMatrix(data=Predicties_Xgb_Test_Class,
reference=Y_Test_Class, positive="1")

```
Grafische weergave resultaat
```{r}

observed <- as.numeric(Y_Test_Class) - 1

plot_pred_type_distribution <- function(df, threshold) {
  v <- rep(NA, nrow(df))
  v <- ifelse(df$fit >= threshold & df$class == 1, "TP", v)
  v <- ifelse(df$fit >= threshold & df$class == 0, "FP", v)
  v <- ifelse(df$fit < threshold & df$class == 1, "FN", v)
  v <- ifelse(df$fit < threshold & df$class == 0, "TN", v)

  df$fit_type <- v

  ggplot(data=df, aes(x=class, y=fit)) +
    geom_violin(fill=rgb(1,1,1,alpha=0.6), color=NA) +
    geom_jitter(aes(color=fit_type), alpha=0.6) +
    geom_hline(yintercept=threshold, color="red", alpha=0.6) +
    scale_color_discrete(name = "type") +
    labs(title=sprintf("Grenswaarde %.2f", threshold))+
    theme_bw()
}

df <- data.frame(class = observed, fit = Predicties_Xgb_Test)

plot_pred_type_distribution(df = df, threshold = 0.3)

```

En de predicties loslaten op de complete dataset, en toevoegen aan de oorspronkelijke dataset.
```{r}
Predicties_Xgb_All  <- predict(Xgb_Model, newdata = Features.matrix,
type="prob")

Data2 <- Data2 %>%
  mutate(Score = Predicties_Xgb_All)

```

En resultaat wegschrijven
```{r}
saveRDS(Data2, "2017-07-17 Data_PILS_Met_Score.rds")
```


========= DIT LAATSTE DEEL GAAT NIET GOED, IK KRIJG EEN FOUT BIJ HET UITLEGGEN VAN INDIVIDUELE GEVALLEN, KOMT WAARSCHIJNLIJK DOOR NA WAARDEN =======

Try to explain xgboost outcomes with DALEX
```{r}

Y <- Data2$ovt
```

```
predict_logit <- function(model, x) {
  raw_x <- predict(model, x)
  exp(raw_x)/(1 + exp(raw_x))
}

explainer_xgb <- DALEX::explain(Xgb_Model, label = "xgb",
                                predict_function = predict_logit,
                                     data = Features.matrix, y = Y)
```

Individuele casussen, eerst functie maken, die plotten op basis van iBreakDown vereenvoudigt

Plot KvK-adres functie

```{r}
Plot_Kvk_Adres <- function(kvknummer12, explainer_xgb, aantal_Data_Model, data_dmy, data) {

  adres1 <- filter(data, KvKnummer12 == kvknummer12)
  score <- str_sub(as.character(adres1$Score),1,5)

  titel  <- paste(str_sub(adres1$KvKnummer12, 1, 8), str_sub(adres1$VestigingNaam,1, 10), str_trim(adres1$Plaats), "Score: ", score, sep = " ")

  if(nrow(adres1) == 0) {
     stop("Kvknummer komt niet voor in bestand")
     }

  adres2 <- filter(data_dmy, KvKnummer12 == kvknummer12) %>%
    as.matrix()


  explainer_xgb$label <- titel

  explain1 <- iBreakDown::break_down(explainer_xgb,
                                     new_observation = adres2)


  plot(explain1,
       max_Data_Model = aantal_Data_Model,
       vcolors = c("green", "red", "purple") )

    }
```

Nu uitproberen op aangeleverde data

```{r}
5.1,2,E_kvk <- c("701205950000", "632340090000", "665602760000", "687320310000")

5.1,2,E_kvk  <- c(603636810000, 242792720000, 534001270000)
```

```
5.1,2,E_kvk <- c(645692410000, 520538650000, 652017010000, 310364180000)
```

```
Nu de scores plotten, en van commentaar voorzien

```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[1],
               explainer_xgb = explainer_xgb,
               aantal_Data_Model = 7,
               data_dmy = Data_Model,
               data = Data2)
```
```

````
---
title: "010 Modelling Xgboost 2019-07-17, bestand met hoog-gecorreleerde
variabelen aanwezig"
author: "5.1.2.E"
date: "28 november 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r}
library(tidyverse)
library(xgboost)
library(caret)
library(tictoc)
library(pROC)
library(here)
```
Functies
```{r}

Precision_n <- function(y, kansen, n) {
  precision_frame <- data.frame(y)
  precision_frame <- mutate(precision_frame, V2=kansen)
  precision_frame_ordered <- arrange(precision_frame, desc(V2))
  precision_frame_top_n <- dplyr::slice(precision_frame_ordered, 1:n)
  precision_frame_top_n_correct <- filter(precision_frame_top_n, y==1)
  precision_n <- nrow(precision_frame_top_n_correct)
  return(precision_n)
}
```


```{r}

dataset_horeca <- readRDS(here("data", "2019-12-24_Horeca_Na_Final_Pre-
processing.rds")

```

Weghalen van achtergrond variabelen
```{r}
data_model <- select(dataset_horeca, -(VestigingNaam:PClon)) %>%
  mutate(KvKnummer12 = as.numeric(KvKnummer12))

```

Opsplitsen van de data in train- en test. En matrices maken van train,
test en alle features
```{r}
set.seed(111)
trainIndex <- createDataPartition(data_model$ovt, p = .8,
                                  list = FALSE,
                                  times = 1)
````
````

```
Y_Train <- data_model$ovt[trainIndex]
X_Train <- data_model[trainIndex,]
X_Train <- select(X_Train, -ovt) %>% data.matrix()

Y_Test  <- data_model$ovt[-trainIndex]
X_Test  <- data_model[-trainIndex,]
X_Test  <- select(X_Test, -ovt) %>% data.matrix()

Features <- select(data_model, -ovt)
Features.matrix <- Features %>% data.matrix()
```

```{r}
# set up the cross-validated hyper-parameter search
xgb_grid_1 = expand.grid(
eta = c(0.2, 0.05),
max_depth = c(4, 5, 6, 8, 10),
gamma = 1
)
```

En XGboost model trainen

```{r}
tic()

for (iter in 1:50) {
    param <- list(objective = "binary:logistic",
          eval_metric = "auc",
          max_depth = sample(4:6, 1),
          eta = runif(1, .01, .3),
          gamma = runif(1, 0.0),
          subsample = runif(1, .6, .9),
          colsample_bytree = runif(1, .5, .8),
          min_child_weight = sample(1:5, 1),
          max_delta_step = sample(1:10, 1)
          )
    cv.nround = 50
    cv.nfold = 5
    seed.number = sample.int(10000, 1)[[1]]
    set.seed(seed.number)
    mdcv <- xgb.cv(data=X_Train, label = Y_Train, params = param,
nthread=6,
                   nfold=cv.nfold, nrounds=cv.nround,
                   verbose = F, early_stopping_rounds =8, maximize=TRUE)

    max_auc = max(mdcv$evaluation_log$test_auc_mean)
    max_auc_index = which.max(mdcv$evaluation_log$test_auc_mean)

    if (max_auc < best_auc) {
        best_auc = max_auc
        best_auc_index = max_auc_index
        best_seednumber = seed.number
        best_param = param
    }
}
```

```
nround = best_auc_index
set.seed(best_seednumber)
md <- xgb.train(data=dtrain, params=best_param, nrounds=nround,
nthread=6)

toc()
```
```{r}
xgb_model

evaluatie <- xgb_model$evaluation_log

ggplot(data = evaluatie, aes(x = iter, y = test_auc_mean)) + geom_line()

beste_iteratie <- evaluatie %>%
  filter(test_auc_mean == max(test_auc_mean)) %>%
  pull(iter)

```



```{r}
xgb_model_opt <- xgboost  (data = X_Train,
                           label = Y_Train,
                           params = param,
                           nfold = 10,
                           eta =
                           max_depth =
                           nround = 150,
                           verbose = F)
```
```{r}

```



```{r}
saveRDS(Xgb_Model, "//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS//Models/2019-07-17 Xgb_Model2.rds")
```

Show variable importance
```{r}
Var_Imp <- xgb.importance(feature_names = NULL, model = Xgb_Model)

library(Ckmeans.1d.dp)

xgb.ggplot.importance(Var_Imp,
                      top_n = 12)

```
```{r}
xgb.ggplot.importance(Var_Imp,
                      top_n = 30)

```

Show partial dependency plots in the shap way
```{r}

xgb.plot.shap(data = X_Train,
              model = Xgb_Model,
              top_n = 20,
              n_col = 3,
              ylab = "kans op AMF overtreding")
```

And make predictions on the testset

```{r}

Predicties_Xgb_Test <- predict(Xgb_Model, newdata = X_Test, type =
"prob")
Predicties_Xgb_Test_Class <- predict(Xgb_Model, newdata = X_Test, type =
"class")
Predicties_Xgb_Test_Class <- as.factor(ifelse(Predicties_Xgb_Test_Class >
0.5, 1, 0))
Y_Test_Class <-  as.factor(Y_Test)

#table(Y_Test)       # Ongeveer 1:6 is overtreder. Dat is 16%
Predicties_Test <- tibble(Predicties_Xgb_Test)
ggplot(data = Predicties_Test, aes(x = Predicties_Xgb_Test)) +
geom_density()
```


```{r}
a <- Precision_n(Y_Test, Predicties_Xgb_Test, 100) # Dat is 61%, dus
verviervoudiging kans.
b <- Precision_n(Y_Test, Predicties_Xgb_Test, 200)
c <- Precision_n(Y_Test, Predicties_Xgb_Test, 300)
d <- Precision_n(Y_Test, Predicties_Xgb_Test, 400)
e <- Precision_n(Y_Test, Predicties_Xgb_Test, 500)

Precision_vector <- c(a, b, c, d, e)
plot(Precision_vector)
```

Nu de ROC afdrukken

```{r}

ROC1 <- roc(Y_Test, Predicties_Xgb_Test)

plot(ROC1, col = "blue", left_margin=NULL)

```

En de AUC op de testset
```{r}
AUC_testset <- pROC::auc(ROC1)
AUC_testset
```

Confusion matrix
```{r}

caret::confusionMatrix(data=Predicties_Xgb_Test_Class,
reference=Y_Test_Class, positive="1")

```
Grafische weergave resultaat
```{r}

observed <- as.numeric(Y_Test_Class) - 1

plot_pred_type_distribution <- function(df, threshold) {
  v <- rep(NA, nrow(df))
  v <- ifelse(df$fit >= threshold & df$class == 1, "TP", v)
  v <- ifelse(df$fit >= threshold & df$class == 0, "FP", v)
  v <- ifelse(df$fit < threshold & df$class == 1, "FN", v)
  v <- ifelse(df$fit < threshold & df$class == 0, "TN", v)

  df$fit_type <- v

  ggplot(data=df, aes(x=class, y=fit)) +
    geom_violin(fill=rgb(1,1,1,alpha=0.6), color=NA) +
    geom_jitter(aes(color=fit_type), alpha=0.6) +
    geom_hline(yintercept=threshold, color="red", alpha=0.6) +
    scale_color_discrete(name = "type") +
    labs(title=sprintf("Grenswaarde %.2f", threshold))+
    theme_bw()
}

df <- data.frame(class = observed, fit = Predicties_Xgb_Test)

plot_pred_type_distribution(df = df, threshold = 0.3)

```

En de predicties loslaten op de complete dataset, en toevoegen aan de oorspronkelijke dataset.
```{r}
Predicties_Xgb_All  <- predict(Xgb_Model, newdata = Features.matrix,
type="prob")

Data2 <- Data2 %>%
  mutate(Score = Predicties_Xgb_All)

```

En resultaat wegschrijven
```{r}
saveRDS(Data2, "2017-07-17 Data_PILS_Met_Score.rds")
```

```
---
title: "010 Modelling Xgboost 2019-08-14, bestand met hoog-gecorreleerde
variabelen aanwezig, filter op enkel de bedrijfsvestigingen met UWV-data"

author: "5.1.2.E"
date: "14 augustus 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r}
library(tidyverse)
library(xgboost)
library(caret)
library(tictoc)
library(pROC)
library(DataExplorer)
```
Functies
```{r}

Precision_n <- function(y, kansen, n) {
  precision_frame <- data.frame(y)
  precision_frame <- mutate(precision_frame, V2=kansen)
  precision_frame_ordered <- arrange(precision_frame, desc(V2))
  precision_frame_top_n <- dplyr::slice(precision_frame_ordered, 1:n)
  precision_frame_top_n_correct <- filter(precision_frame_top_n, y==1)
  precision_n <- nrow(precision_frame_top_n_correct)
  return(precision_n)
}
```


```{r}

Data_alles <- readRDS( "\\\\Client\\G$\\I-SZW\\O&A\\Data Science
projecten\\2019 PILS\\Data\\2019-08-14_Horeca_Na_Final2_Pre-
processing.rds")

```


Nu de bedrijfsvestigingen verwijderen waarvoor we geen UWV-data hebben.
Tevens de missings oplossen voor de overgebleven dataset, omdat ik ook
uitleg wil maken met behulp van iBreakDown.

Gedeelte van de missings (voor continue variabelen) vervang ik door de
mediaan.
Gedeelte van de missings (voor indicator variabelen) vervang ik door 0.
```{r}

Data2 <- Data_alles %>%
  filter(MISSING_UWV_gegevens == 0)
```
```

```
Data3 <- Data2

plot_missing(Data3)
```
Nieuwe feature: afwijking uurloon van gemiddelde uurloon
sector/grootteklasse
```{r}
Data3 <- Data3 %>%
  mutate(Fractie_afw_uurloon_van_gemiddelde_sector =
(UWV_gemiddeld_uurloon - Uurloon2015)/Uurloon2015)
```


Weghalen van achtergrond variabelen
```{r}
Data_Model <- select(Data3, -(VestigingNaam:PClon)) %>%
  mutate(KvKnummer12 = as.numeric(KvKnummer12))

```

Opsplitsen van de data in train- en test. En matrices maken van train,
test en alle features
```{r}
set.seed(111)
trainIndex <- createDataPartition(Data_Model$ovt, p = .8,
                                  list = FALSE,
                                  times = 1)

Y_Train <- Data_Model$ovt[trainIndex]
X_Train <- Data_Model[trainIndex,]
X_Train <- select(X_Train, -ovt) %>% data.matrix()

Y_Test  <- Data_Model$ovt[-trainIndex]
X_Test  <- Data_Model[-trainIndex,]
X_Test  <- select(X_Test, -ovt) %>% data.matrix()

Features <- select(Data_Model, -ovt)
Features.matrix <- Features %>% data.matrix()

```


En XGboost model trainen

```{r}
tic()
set.seed(12)

best_param <- list(objective = "binary:logistic",  # For regression
                eval_metric = "auc",      # rmse is used for regression
                max_depth = 4,
                eta = 0.3,   # Learning rate
                subsample = 0.8,
                colsample_bytree = 0.8,
                min_child_weight = 5,
                scale_pow_weight = sum(Y_Train == 0) / sum(Y_Train == 1),
                max_delta_step = 8)
```

```
Xgb_Model <- xgboost(data = X_Train, label = Y_Train, params =
best_param, nround = 250, verbose = F)

toc()
```
```{r}
saveRDS(Xgb_Model, "//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS//Models/2019-08-15 Xgb_Model_UWV_rijen.rds")
```

Show variable importance
```{r}
Var_Imp <- xgb.importance(feature_names = NULL, model = Xgb_Model)

library(Ckmeans.1d.dp)

xgb.ggplot.importance(Var_Imp,
                      top_n = 15)

```
```{r}
xgb.ggplot.importance(Var_Imp,
                      top_n = 30)

```
Show partial dependency plots in the shap way
```{r}

xgb.plot.shap(data = X_Train,
              model = Xgb_Model,
              top_n = 12,
              n_col = 3,
              ylab = "kans op AMF overtreding")
```

And make predictions on the testset

```{r}

Predicties_Xgb_Test <- predict(Xgb_Model, newdata = X_Test, type =
"prob")
Predicties_Xgb_Test_Class <- predict(Xgb_Model, newdata = X_Test, type =
"class")
Predicties_Xgb_Test_Class <- as.factor(ifelse(Predicties_Xgb_Test_Class >
0.15, 1, 0))
Y_Test_Class <-  as.factor(Y_Test)

Predicties_Test <- tibble(Predicties_Xgb_Test)
ggplot(data = Predicties_Test, aes(x = Predicties_Xgb_Test)) +
geom_density()
```



```{r}
a <- Precision_n(Y_Test, Predicties_Xgb_Test, 100) # Dat is 61%, dus
vervierevoudiging kans.
b <- Precision_n(Y_Test, Predicties_Xgb_Test, 200)
c <- Precision_n(Y_Test, Predicties_Xgb_Test, 300)
```

```
d <- Precision_n(Y_Test, Predicties_Xgb_Test, 400)
e <- Precision_n(Y_Test, Predicties_Xgb_Test, 500)

Precision_vector <- c(a, b, c, d, e)
plot(Precision_vector)
```

Nu de ROC afdrukken

```{r}

ROC1 <- roc(Y_Test, Predicties_Xgb_Test)

plot(ROC1, col = "blue", left_margin=NULL)

```

En de AUC op de testset
```{r}
AUC_testset <- pROC::auc(ROC1)
AUC_testset
```

Confusion matrix
```{r}

caret::confusionMatrix(data=Predicties_Xgb_Test_Class,
reference=Y_Test_Class, positive="1")

```
Grafische weergave resultaat
```{r}

observed <- as.numeric(Y_Test_Class) - 1

plot_pred_type_distribution <- function(df, threshold) {
  v <- rep(NA, nrow(df))
  v <- ifelse(df$fit >= threshold & df$class == 1, "TP", v)
  v <- ifelse(df$fit >= threshold & df$class == 0, "FP", v)
  v <- ifelse(df$fit < threshold & df$class == 1, "FN", v)
  v <- ifelse(df$fit < threshold & df$class == 0, "TN", v)

  df$fit_type <- v

  ggplot(data=df, aes(x=class, y=fit)) +
    geom_violin(fill=rgb(1,1,1,alpha=0.6), color=NA) +
    geom_jitter(aes(color=fit_type), alpha=0.6) +
    geom_hline(yintercept=threshold, color="red", alpha=0.6) +
    scale_color_discrete(name = "type") +
    labs(title=sprintf("Grenswaarde %.2f", threshold))+
    theme_bw()
}

df <- data.frame(class = observed, fit = Predicties_Xgb_Test)

plot_pred_type_distribution(df = df, threshold = 0.15)

```

En de predicties loslaten op de complete dataset, en toevoegen aan de oorspronkelijke dataset.

```{r}
Predicties_Xgb_All  <- predict(Xgb_Model, newdata = Features.matrix, type="prob")

Data3 <- Data3 %>%
  mutate(Score = Predicties_Xgb_All)

```

En resultaat wegschrijven

```{r}
saveRDS(Data3, "2017-08-15 Data_PILS_UVW_Rijen_Met_Score.rds")
```

Try to explain xgboost outcomes with DALEX

```{r}

Y <- Data3$ovt

predict_logit <- function(model, x) {
  raw_x <- predict(model, x, type = "prob")
  }

explainer_xgb <- DALEX::explain(Xgb_Model, label = "xgb",
                                predict_function = predict_logit,
                                     data = Features.matrix, y = Y)

```

Individuele casussen, eerst functie maken, die plotten op basis van iBreakDown vereenvoudigt

Plot KvK-adres functie

```{r}

Plot_Kvk_Adres <- function(kvknummer12, explainer_xgb, aantal_Data_Model, data_dmy, data) {

  adres1 <- filter(data, KvKnummer12 == kvknummer12)
  score <- str_sub(as.character(adres1$Score),1,5)

  titel  <- paste(str_sub(adres1$KvKnummer12, 1, 8), str_sub(adres1$VestigingNaam,1, 10), str_trim(adres1$Plaats), "Score: ", score, sep = " ")

  if(nrow(adres1) == 0) {
     stop("Kvknummer komt niet voor in bestand")
     }

  adres2 <- filter(data_dmy, KvKnummer12 == kvknummer12) %>%
    as.matrix()


  explainer_xgb$label <- titel
```

```
  explain1 <- iBreakDown::break_down(explainer_xgb,
                                     new_observation = adres2)


  plot(explain1,
       max_Data_Model = aantal_Data_Model,
       vcolors = c("green", "red", "purple") )

    }
```

Nu uitproberen op aangeleverde data

```{r}

5.1,2,E_kvk <- c("701205950000", "632340090000", "665602760000",
"687320310000")

5.1,2,E_kvk  <- c(603636810000, 242792720000, 534001270000)

5.1,2,E_kvk <- c(645692410000, 520538650000, 652017010000, 310364180000)

Selectie_inspecteurs_kvk <- c(5.1,2,E_kvk, 5.1,2,E_kvk, 5.1,2,E_kvk)
```
Controleren welke kvk's voorkomen en welke niet, door vergelijking van de 2 sets kvk12-nummers
```{r}
KvK12_data <- Data3$KvKnummer12

KvK_beide <- dplyr::intersect(Selectie_inspecteurs_kvk, KvK12_data)

```

Nu de scores plotten, en van commentaar voorzien

```{r}

Plot_Kvk_Adres(kvknummer12 = KvK_beide[1],
               explainer_xgb = explainer_xgb,
               aantal_Data_Model = 7,
               data_dmy = Features,
               data = Data3)
```

```{r}

Plot_Kvk_Adres(kvknummer12 = KvK_beide[2],
               explainer_xgb = explainer_xgb,
               aantal_Data_Model = 7,
               data_dmy = Features,
               data = Data3)
```

```{r}

Plot_Kvk_Adres(kvknummer12 = KvK_beide[3],
```

```
                explainer_xgb = explainer_xgb,
                aantal_Data_Model = 7,
                data_dmy = Features,
                data = Data3)
```


```{r}

Plot_Kvk_Adres(kvknummer12 = KvK_beide[4],
                explainer_xgb = explainer_xgb,
                aantal_Data_Model = 7,
                data_dmy = Features,
                data = Data3)
```

````
---
title: "010 Modelling Ranger 2019-08-16, bestand met hoog-gecorreleerde
variabelen aanwezig"
author: "5.1.2,E"
date: "20 augustus 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r, echo=FALSE, warning=FALSE, message=FALSE}
library(tidyverse)
library(caret)
library(tictoc)
library(pROC)
library(recipes)
library(ranger)
library(tidytext)
library(forcats)

```
Functies
```{r}

Precision_n <- function(y, kansen, n) {
  precision_frame <- data.frame(y)
  precision_frame <- mutate(precision_frame, V2=kansen)
  precision_frame_ordered <- arrange(precision_frame, desc(V2))
  precision_frame_top_n <- dplyr::slice(precision_frame_ordered, 1:n)
  precision_frame_top_n_correct <- filter(precision_frame_top_n, y==1)
  precision_n <- nrow(precision_frame_top_n_correct)
  return(precision_n)
}
```


```{r}
setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
Data1 <- readRDS("2019-08-16 Horeca_Na_Final2_Pre-processing.rds")

```

Weghalen van achtergrond variabelen.

Ranger kan niet omgaan met missende waarden, daarom UWV-gegevenskolommen
helaas verwijderen, m.u.v. het veld MISSING_UWV.

```{r}
Data_Model <- Data1 %>%
  select(-(VestigingNaam:Plaats)) %>%
  select(-(UWV_aantal_werknemers:UWV_aantal_recs_nuluren_wel_loon)) %>%
  mutate(MISSING_UWV_gegevens = replace_na(MISSING_UWV_gegevens, replace
= "1")) %>%
  mutate(MISSING_UWV_gegevens = as.numeric(MISSING_UWV_gegevens)) %>%
  mutate(KvKnummer12 = as.numeric(KvKnummer12))
````
````

```

Opsplitsen van de data in train- en test. En matrices maken van train, test en alle features

```{r}
set.seed(111)
trainIndex <- createDataPartition(Data_Model$ovt, p = .8,
                                  list = FALSE,
                                  times = 1)

Y_Train <- Data_Model$ovt[trainIndex]
Y_Train_Cat <- as_factor(Y_Train)
levels(Y_Train_Cat) <- make.names(levels(Y_Train_Cat))

X_Train <- Data_Model[trainIndex,]
X_Train <- select(X_Train, -ovt)

Y_Test  <- Data_Model$ovt[-trainIndex]
Y_Test_Cat <- as_factor(Y_Test)
levels(Y_Test_Cat) <- make.names(levels(Y_Test_Cat))

X_Test  <- Data_Model[-trainIndex,]
X_Test  <- select(X_Test, -ovt)

Features <- select(Data_Model, -ovt)

```

Model preparations. Beperkt aantal folds, te weten 3, omdat het geheel helaas anders veel te lang duurt.

En Ranger model trainen

```{r}
tic()
set.seed(12)

Ranger_Model <- ranger(Y_Train_Cat ~ .,
                       data = X_Train,
                       probability = TRUE,
                       importance = "impurity")

toc()
```

Show variable importance

```{r}
Var_Imp <- importance(Ranger_Model)

Var_names <- names(Var_Imp)

Var_Importance_12 <- tibble(Variabele = Var_names, Importance = Var_Imp)
%>% arrange(desc(Importance)) %>% dplyr::slice(1:12)
```
```

```
ggplot(data = Var_Importance_12, aes(x = reorder(Variabele, Importance),
y = Importance)) +
       geom_bar(stat = "identity", fill = "blue") +
       coord_flip() +
       xlab("Variabele") +
       theme_bw()
```

And make predictions on the testset

```{r}
Predicties_Ranger_Test <- predict(Ranger_Model,
                  data = X_Test,
                  type = "response")

Predicties_Test <- tibble(Predicties_Ranger_Test$predictions[,2])
names(Predicties_Test) <- "Predictie"
ggplot(data = Predicties_Test, aes(x = Predictie)) + geom_density()

```


```{r}
a <- Precision_n(Y_Test, Predicties_Test$Predictie, 100)
b <- Precision_n(Y_Test, Predicties_Test$Predictie, 200)
c <- Precision_n(Y_Test, Predicties_Test$Predictie, 300)
d <- Precision_n(Y_Test, Predicties_Test$Predictie, 400)
e <- Precision_n(Y_Test, Predicties_Test$Predictie, 500)

Precision_vector <- c(a, b, c, d, e)
plot(Precision_vector)
```

Nu de ROC afdrukken

```{r}

ROC1 <- roc(Y_Test, Predicties_Test$Predictie)

plot(ROC1, col = "blue", left_margin=NULL)

```

En de AUC op de testset
```{r}
AUC_testset <- pROC::auc(ROC1)
AUC_testset
```

```{r}
tuneGrid.lr <- expand.grid(.alpha = seq(0.1, 1, by= 0.1),
                           .lambda = seq(0.001, 0.1, by = 0.001))

Y_Train <- make.names(Y_Train)
```

````
---
title: "Modelling"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Load Libraries
```{r, echo=FALSE, message=FALSE}
library(tidyverse)
library(DALEX)
library(xgboost)
library(Ckmeans.1d.dp)
library("iBreakDown")
```


## Load Data & Model: both original and dummified

```{r echo=FALSE, message=FALSE}

path_plots = "\\\\Client\\G$\\I-SZW\\O&A\\Data Science projecten\\2019
RUM en PILS\\plots\\"

data1 <- readRDS( "\\\\Client\\G$\\I-SZW\\O&A\\Data Science
projecten\\2019 RUM en PILS\\Data\\2019-04-03, Pils_data_without.rds")

Y <- data1$ovt

Features <- readRDS( "\\\\Client\\G$\\I-SZW\\O&A\\Data Science
projecten\\2019 RUM en PILS\\Data\\2019-06-04 features.rds")

features.matrix <- data.matrix(Features)

xgb.model4 <- readRDS("//client/G$/I-SZW/O&A/Data Science projecten/2019
RUM en PILS//Modellen//2019-06-12 xgbmodel.rds")

```


Show variable importance
```{r}
varImp <- xgb.importance(feature_names = NULL, model = xgb.model4)

xgb.ggplot.importance(varImp,
                      top_n = 15)

row_names <- varImp$Feature
```

Predict on the testset and on the whole dataset
```{r}

predicties_xgb_all  <- predict(xgb.model4, newdata = features.matrix,
type="prob")

data1 <- data1 %>%
````
````

```
  mutate(score = predicties_xgb_all)

```

Teken de verdeling van de voorspellingen
```{r}
ggplot(data = data1, aes(x = predicties_xgb_all)) +
  geom_density()
```

Try to explain caret outcomes with DALEX
```{r}

predict_logit <- function(model, x) {
  raw_x <- predict(model, x)
  exp(raw_x)/(1 + exp(raw_x))
}

explainer_xgb <- DALEX::explain(xgb.model4, label = "xgb",
                                predict_function = predict_logit,
                                     data = features.matrix, y = Y)

```

Individuele casussen, eerst functie maken, die plotten op basis van iBreakDown vereenvoudigt

```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[2],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)
```
Plot KvK-adres functie
```{r}

Plot_Kvk_Adres <- function(kvknummer12, explainer_xgb, aantal_Features,
data_dmy, data) {

  adres1 <- filter(data, KvKnummer12 == kvknummer12)
  score <- str_sub(as.character(adres1$score),1,5)

  titel  <- paste(str_sub(adres1$KvKnummer12, 1, 8),
str_sub(adres1$VestigingNaam,1, 10), str_trim(adres1$Plaats), "Score: ",
score, sep = " ")

  if(nrow(adres1) == 0) {
     stop("Kvknummer komt niet voor in bestand")
     }

  adres2 <- filter(data_dmy, KvKnummer12 == kvknummer12) %>%
    as.matrix()


  explainer_xgb$label <- titel
```

```
  explain1 <- iBreakDown::break_down(explainer_xgb,
                                     new_observation = adres2)


  plot(explain1,
       max_Features = aantal_Features,
       vcolors = c("green", "red", "purple") )

    }
```

Nu uitproberen op aangeleverde data

```{r}

5.1,2,E_kvk <- c("701205950000", "632340090000", "665602760000",
"687320310000")

5.1,2,E_kvk  <- c(603636810000, 242792720000, 534001270000)

5.1,2,E_kvk <- c(645692410000, 520538650000, 652017010000, 310364180000)


```

Nu de scores plotten, en van commentaar voorzien

```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[1],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)
```



```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[3],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)

```
Scoort niet bijzonder hoog.

```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[4],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)
```

```

Nu de adressen van [redacted] eerste adres:

```{r}

Plot_Kvk_Adres(kvknummer12 = [redacted]_kvk[1],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)

```
Dit adres scoort boven gemiddeld

```{r}

Plot_Kvk_Adres(kvknummer12 = [redacted]_kvk[2],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)

```
Dit adres is niks bijzonders. Scoort nergens hoog of laag op.


```{r}

Plot_Kvk_Adres(kvknummer12 = [redacted]_kvk[3],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)


```
Dit adres scoort iets boven gemiddeld, maar niet veel. Lage lonen, veel werknemers, onveilige buurt.

Nu de adressen van [redacted]

```{r}


Plot_Kvk_Adres(kvknummer12 = [redacted]_kvk[1],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)

```
Scoort behoorlijk boven gemiddeld.

```{r}

```
Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[2],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)


```
Hoog risico-bedrijf!!!


```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[3],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)
```
Scoort ook heel hoog.


```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[4],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
               data = data1)
```
En deze is ook verhoogd risico.

````
---
title: "Modelling"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Load Libraries
```{r, echo=FALSE, message=FALSE}
library(tidyverse)
library(xgboost)
library(iml)
library(DALEX)
```


## Load Data & Model


```{r echo=FALSE, message=FALSE}

path_plots = "\\\\Client\\G$\\I-SZW\\O&A\\Data Science projecten\\2019
PILS\\Plots\\"

Data1 <- readRDS( "\\\\Client\\G$\\I-SZW\\O&A\\Data Science
projecten\\2019 PILS\\Data\\2019-07-17_Horeca_Na_Final2_Pre-
processing.rds")

Xgb_Model <- readRDS("//client/G$/I-SZW/O&A/Data Science projecten/2019
PILS//Models/2019-07-17 Xgb_Model2.rds")

```


Nu voorbereiden van iml uitleg

1) Weghalen van achtergrond variabelen, en overtreding, zodat we de
features overhouden.
```{r}
Features <- select(Data1, -(VestigingNaam:PClon)) %>%
  mutate(KvKnummer12 = as.numeric(KvKnummer12)) %>%
  select(-ovt)

```
2) En de response variabele definieren
```{r}
Response <- Data1$ovt
```


3) En de predictor functie definieren
```{r}
Pred <- function(model, newdata)  {
  Xgb_Prob  = predict(model, newdata = data.matrix(newdata), type =
"prob") }

```
````

4 IML predictor object maken
```{r}
Predictor_Xgb <- Predictor$new(
  model = Xgb_Model,
  data = Features,
  y = Response,
  predict.fun = Pred,
  class = "classification"
)

```

Shapley uitleg voor 1e casus van het bestand
```{r}
x.interest <- filter(Features, KvKnummer12 == 645692410000)

Shapley_Xgb <- Shapley$new(Predictor_Xgb, x.interest = x.interest)

str(Shapley_Xgb)

plot(Shapley_Xgb)

```


Individuele casussen, eerst functie maken, die plotten op basis van iBreakDown vereenvoudigt

Nu uitproberen op aangeleverde data

```{r}

5.1,2,E_kvk <- c("701205950000", "632340090000", "665602760000", "687320310000")

5.1,2,E_kvk  <- c(603636810000, 242792720000, 534001270000)

5.1,2,E_kvk <- c(645692410000, 520538650000, 652017010000, 310364180000)


```

Explainer object maken m.b.v. iBreakdown
```{r}

predict_logit <- function(model, x) {
  raw_x <- predict(model, data.matrix(x), type="prob")
  }

explainer_xgb <- DALEX::explain(Xgb_Model, label = "xgb",
                                predict_function = predict_logit,
                                     data = Features, y = Response)
```


Plot KvK-adres functie
```{r}
```

```
Plot_Kvk_Adres <- function(kvknummer12, explainer_xgb, aantal_Features,
data_dmy, data) {

  adres1 <- filter(data, KvKnummer12 == kvknummer12)
  score <- str_sub(as.character(adres1$score),1,5)

  titel  <- paste(str_sub(adres1$KvKnummer12, 1, 8),
str_sub(adres1$VestigingNaam,1, 10), str_trim(adres1$Plaats), "Score: ",
score, sep = " ")

  if(nrow(adres1) == 0) {
     stop("Kvknummer komt niet voor in bestand")
     }

  adres2 <- filter(data_dmy, KvKnummer12 == kvknummer12) %>%
    as.matrix()


  explainer_xgb$label <- titel

  explain1 <- iBreakDown::break_down(explainer_xgb,
                                     new_observation = adres2)


  plot(explain1,
       max_Features = aantal_Features,
       vcolors = c("green", "red", "purple") )

    }
```

Nu uitproberen op aangeleverde data

```{r}
5.1,2,E_kvk <- c("701205950000", "632340090000", "665602760000",
"687320310000")

5.1,2,E_kvk  <- c(603636810000, 242792720000, 534001270000)

5.1,2,E_kvk <- c(645692410000, 520538650000, 652017010000, 310364180000)


```

Nu de scores plotten, en van commentaar voorzien

```{r}

Plot_Kvk_Adres(kvknummer12 = 5.1,2,E_kvk[1],
               explainer_xgb = explainer_xgb,
               aantal_Features = 7,
               data_dmy = Features,
```

```
                data = Data1)
```

````
---
title: "080 Exporatie personeelsverloop"
author: "5.1.2.E"
date: "10 juli 2019"
output: html_document
---
```{r}

```


```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r inlezen libraries, echo=FALSE, message=FALSE}
library(tidyverse)
library(DataExplorer)
library(forcats)
library(ggthemes)

```



```{r inlezen bestanden, echo=FALSE, message=FALSE}
Personeelsverloop <- read_csv("//client/G$/I-SZW/O&A/Data Science
projecten/2019
PILS/Data/Bronbestanden/20190705_MdG_personeelsverloop_prog_H&D.csv", na
= "NULL" )

head(Personeelsverloop, 5)

Personeelsverloop <- Personeelsverloop %>%
  mutate(SBICodeOmschrijving = as.factor(SBICodeOmschrijving)) %>%
  mutate(SBICodeOmschrijving = fct_lump(SBICodeOmschrijving, 20,
other_level = "Overige")) %>%
  mutate(SBICodeOmschrijving = fct_explicit_na(SBICodeOmschrijving,
na_level = "Onbekend"))
```
Uit welke variabelen is het bestand opgebouwd?
```{r}
plot_str(Personeelsverloop)
```


En nog meer algemene informatie
```{r algemene info}
introduce(Personeelsverloop)
```
```{r}
plot_missing(Personeelsverloop)
```



Wat gebeurt er met de omvang als we groeperen op Kvk8 niveau?
```{r}
df <- Personeelsverloop %>%
  group_by(KvK_KvK8) %>%
  summarise(Aantal = n() )
````
````

```
Ik gooi record met lege Kvk8's eruit
```{r}
Df <- Personeelsverloop %>%
  filter(!is.na(KvK_KvK8)) %>%
  arrange(desc(KvK_KvK8))

n_distinct(Df)

Df2 <- distinct(Df, .keep_all = TRUE)

```


Wat is de verdeling van de categoriale variabelen?
```{r}
Temp <- Df2 %>% filter(SBICodeOmschrijving != "Onbekend" &
SBICodeOmschrijving != "Overige")

plot_bar(Temp, ggtheme = theme_minimal())
```

Wat is de verdeling van de continue variabelen?
```{r}
Temp <- select(Df2, groei_tov_sep18:perc_groei_sinds_sep18,
groei_tov_feb18, perc_verloop_sinds_feb18 )

plot_histogram(Temp, ggtheme = theme_minimal() )
```

Welke features zijn er af te leiden?
Twee soorten:
wijzigingen in samenstelling personeel
groei/afname totaal aantal personeelsleden
```{r}
Features <- Df2 %>%
  select(KvK_KvK8, SBICodeOmschrijving, perc_verloop_sinds_feb18,
perc_groei_sinds_feb18) %>%
  mutate(Groei_Personeel = case_when (perc_groei_sinds_feb18  > 0 ~
"Groei",
                                      perc_groei_sinds_feb18 == 0 ~
"Gelijk",
                                      perc_groei_sinds_feb18  < 0 ~
"Krimp",
                                      TRUE ~ "Onbekend"))  %>%
  select(-perc_groei_sinds_feb18)
```

En nu exploratie Features
```{r}
ggplot(data = Features, aes(x = Groei_Personeel)) +
geom_bar(fill = "darkblue", width = 0.7) +
theme_economist() +
scale_color_economist() +
xlab("Groei personeel") +
ylab("Aantal")
```
```{r}

```
ggplot(data = Features, aes(x = perc_verloop_sinds_feb18)) +
geom_histogram(fill = "darkblue") +
theme_economist() +
scale_color_economist() +
xlab("Personeelsverloop") +
ylab("Percentage van personeel dat gewisseld is")
```
Ik twijfel of ik bedrijven met 1 werknemer moet uitsluiten, of een indeling moet maken naar:
Geen (= 0), Laag(< 25%), Hoog (> 25%)

Die zou er dan als volgt uit zien:
```{r}
Features <- Features %>%
  mutate(Verloop_Personeel = case_when (perc_verloop_sinds_feb18    == 0 ~ "Geen",
                                        perc_verloop_sinds_feb18    < 30 & perc_verloop_sinds_feb18 > 0 ~ "Laag",
                                        perc_verloop_sinds_feb18    >= 30 ~ "Hoog",
                                        TRUE ~ "Onbekend"))

```
En deze plotten
```{r}
ggplot(data = Features, aes(x = Verloop_Personeel)) +
geom_bar(fill = "darkblue", width = 0.7) +
theme_economist() +
scale_color_economist() +
xlab("Verloop personeel") +
ylab("Aantal")
```
```{r}
Df3 <- Df2 %>%
  select(KvK_KvK8, perc_verloop_sinds_feb18, perc_groei_sinds_feb18)
```

Nu wegschrijven
```{r}
setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")
write_csv(Df3, "2019-07-15 UWV Personeelsverloop.csv")

```

````
---
title: "082 Exporatie CBS-risicotabel VNG - tabel 1"
author: "5.1.2.E"
date: "11 juli 2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r inlezen libraries, echo=FALSE, message=FALSE}
library(tidyverse)
library(DataExplorer)
library(forcats)
library(ggthemes)
library(readxl)

```

Inlezen tabel met data over huishoudens in gemeente en uitkeringen
```{r}

Tabel_1_Risicomodel_VNG <- read_excel("//client/G$/I-SZW/O&A/Data Science
projecten/2019 PILS/Data/Tabel_1_Risicomodel_VNG.xlsx",
    col_types = c("text", "blank", "text",
        "text", "numeric", "numeric", "numeric",
        "text", "numeric", "text"))
View(Tabel_1_Risicomodel_VNG)
```



Nog wat edit-werk
```{r inlezen bestanden, echo=FALSE, message=FALSE}

Df <- Tabel_1_Risicomodel_VNG %>%
  select( -Gemeente_Code, -Gemeente_Naam ) %>%
  mutate(Aant_Pers_Bijstand_PC4 = as.numeric(Aant_Pers_Bijstand_PC4)) %>%
  mutate(Perc_Leegst_Woningen_PC4 = as.numeric(Perc_Leegst_Woningen))
```
Uit welke variabelen is het bestand opgebouwd?
```{r}
plot_str(Df)
```


En nog meer algemene informatie
```{r algemene info}
introduce(Df)
```
```{r}
plot_missing(Df)
```



Wat is de verdeling van de continue variabelen?
```{r}
plot_histogram(Df, ggtheme = theme_minimal() )
```
````

Verdelingen zijn allemaal behoorlijk scheef!

Welke features zijn er af te leiden?

Percentage bijstanduitkering
```{r}
Df <- Df %>%
  mutate(Perc_Uitkeringstrekkers_PC4 = (Aant_Pers_Bijstand_PC4 /
Aantal_Inw_PC4_GE_15jr) * 100) %>%
  mutate(Perc_Leegst_Woningen_Schoon_PC4 =
ifelse(Perc_Leegst_Woningen_PC4 > 10,10, Perc_Leegst_Woningen_PC4)) %>%
  select(PC4, Perc_Uitkeringstrekkers_PC4,
Perc_Leegst_Woningen_Schoon_PC4)
```

En nu exploratie Features
```{r}
ggplot(data = Df, aes(x = Perc_Uitkeringstrekkers_PC4)) +
#geom_histogram(fill = "darkblue") +
geom_density(fill = "darkblue") +
theme_economist() +
scale_color_economist() +
xlab("Percentage uitkeringstrekkers") +
ylab("Dichtheid")
```
```{r}
ggplot(data = Df, aes(x = Perc_Leegst_Woningen_Schoon_PC4)) +
geom_histogram(fill = "darkblue") +
theme_economist() +
scale_color_economist() +
xlab("Percentage leegstand woningen")

```
```{r, echo=FALSE, warnings=FALSE, message=FALSE}

Tabel_2_Risicomodel_VNG <- read_excel("//client/G$/I-SZW/O&A/Data Science
projecten/2019 PILS/Data/Tabel_2_Risicomodel_VNG.xlsx",
    col_types = c("numeric", "blank", "numeric"),
    na = ".")

Df2 <- Tabel_2_Risicomodel_VNG %>%
  mutate(PC4 = as.character(PC4))
```

En nu exploratie Features
```{r}
ggplot(data = Df2, aes(x = Perc_Personen_Laag_Inkomen_In_PC4)) +
geom_histogram(fill = "darkblue") +
theme_economist() +
scale_color_economist() +
xlab("Percentage personen met Laag inkomen") +
ylab("Dichtheid")
```

Nu de twee tabellen aan elkaar knopen, en wegschrijven naar schijf.
```{r}
Uitvoer_tabel <- left_join(Df, Df2, by = "PC4")
```

En nu wegschrijven naar schijf

```{r}
setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 PILS/Data")

write_csv(Uitvoer_tabel, "2019-07-15 CBS_Risico_Indicatoren_PC4.csv")
```

```
# Script om van een geselecteerde gemeeente de Libra-gegevens te leveren.
# In 2 bestanden, een Excel met de Gemeente-gegevens, en een Excel met de
Postcode4-gegevens

library(tidyverse)
library(xlsx)

pad_data <- "H:/Mijn documenten/2019 Libra/Data/"

pad_bestand <- function(pad, bestand) {
  file <- paste0(pad, bestand)
  return(file) }

# Lees Postcode 4 en Gemeente bestand in
PC4_Tot <- readRDS(pad_bestand(pad_data, "PC4_tot_2019-07-02.rds")) #
Bevat alle gegevens op PC4 niveau
Gemeente_Tot <- readRDS(pad_bestand(pad_data, "Gemeente_tot_2019-07-
02.rds")) # Bevat alle gegevens op gemeente-niveau

# Selecgteer Gemeentebestand op naam gemeente

Selectie = c("Amersfoort", "Almere", "Utrecht", "Zederik", "Vianen",
"Leerdam",
             "Veenendaal", "Lelystad")

Selectie_Naam = "RPF Midden Nederland"

Gemeente_Filenaam_Excel <- paste0(Selectie_Naam, "
Gemeente_selectie.xlsx")
Postcode4_Filenaam_Excel <- paste0(Selectie_Naam, "
Postcode4_selectie.xlsx")

Gemeente_Selectie <- Gemeente_Tot %>%
  filter(name %in% Selectie)

setwd("//client/G$/I-SZW/O&A/Data Science projecten/2019 Libra
Kaart/Output")
write.xlsx(Gemeente_Selectie, file = Gemeente_Filenaam_Excel, sheetName =
"Selectie Gemeentes", col.names = TRUE )

PC4_Selectie <- PC4_Tot %>%
  filter(gemeentenaam %in% Selectie)

write.xlsx(PC4_Selectie, file = Postcode4_Filenaam_Excel, sheetName =
"Postcode4 selectie", col.names = TRUE )
```

Doc17

Inspectie SZW
Ministerie van Sociale Zaken en Werkgelegenheid

IPA

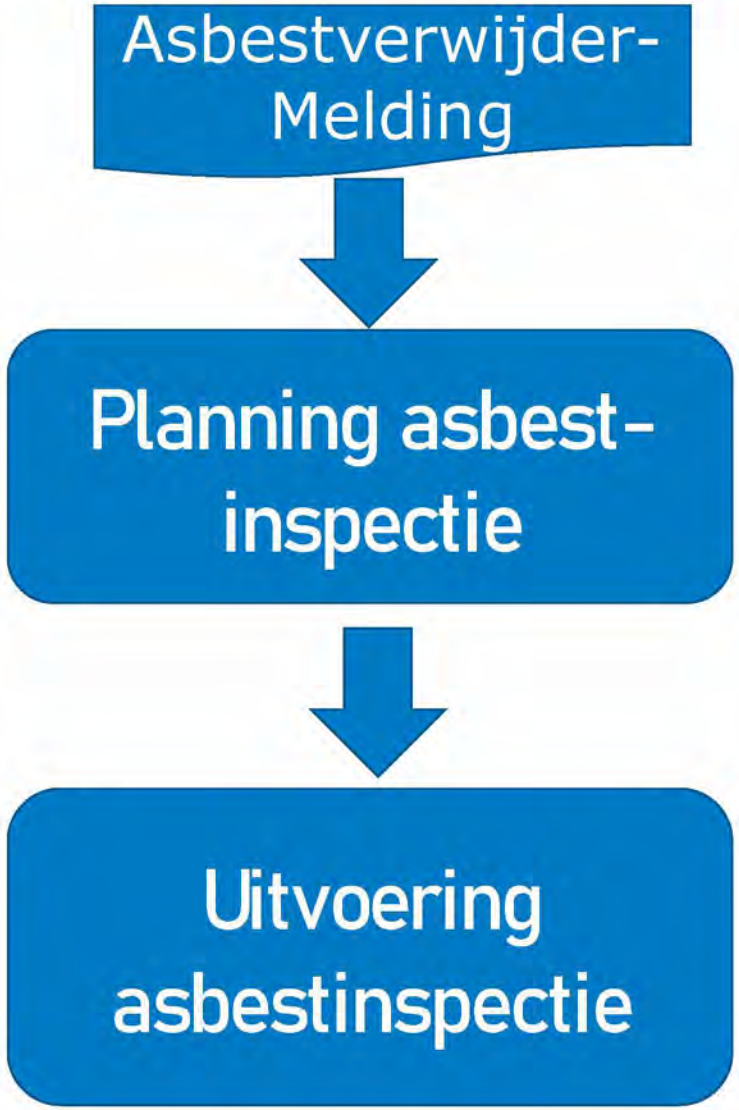
Intelligent Plannen Asbestinspecties

Asbestverwijdering

Asbest Inventarisatie

Asbest Verwijdering

Asbestinspectie van legale asbestverwijderingen



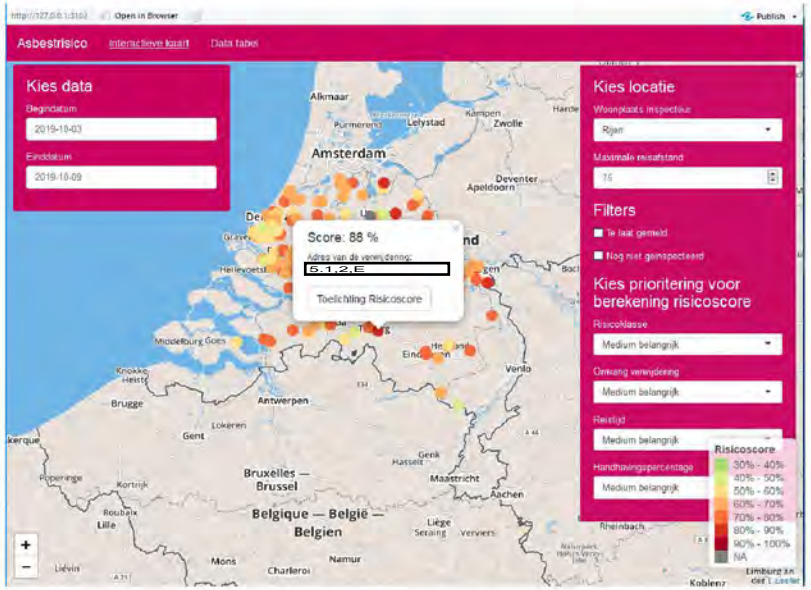
Positionering IPA

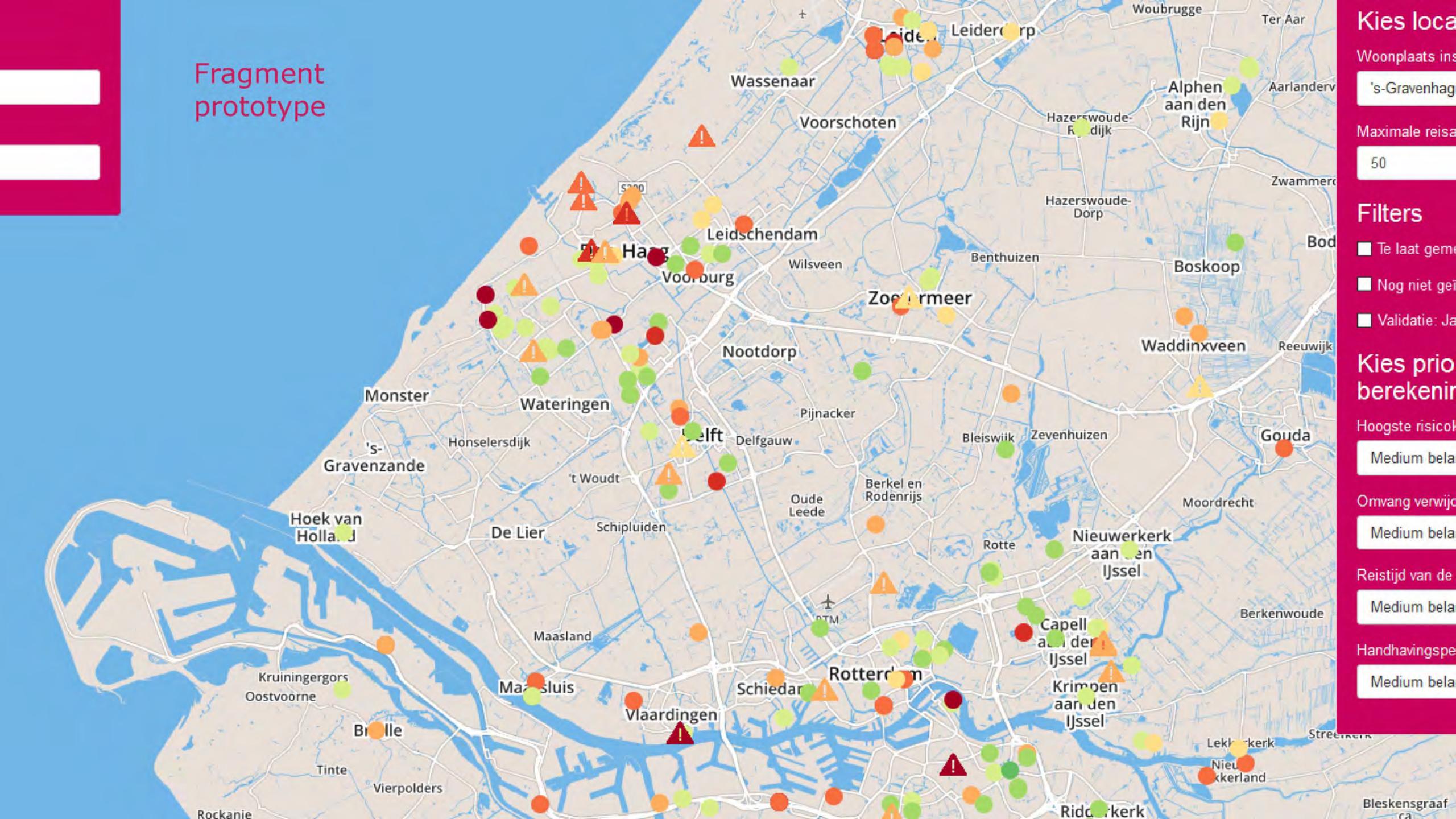
Asbestverwijder-Melding

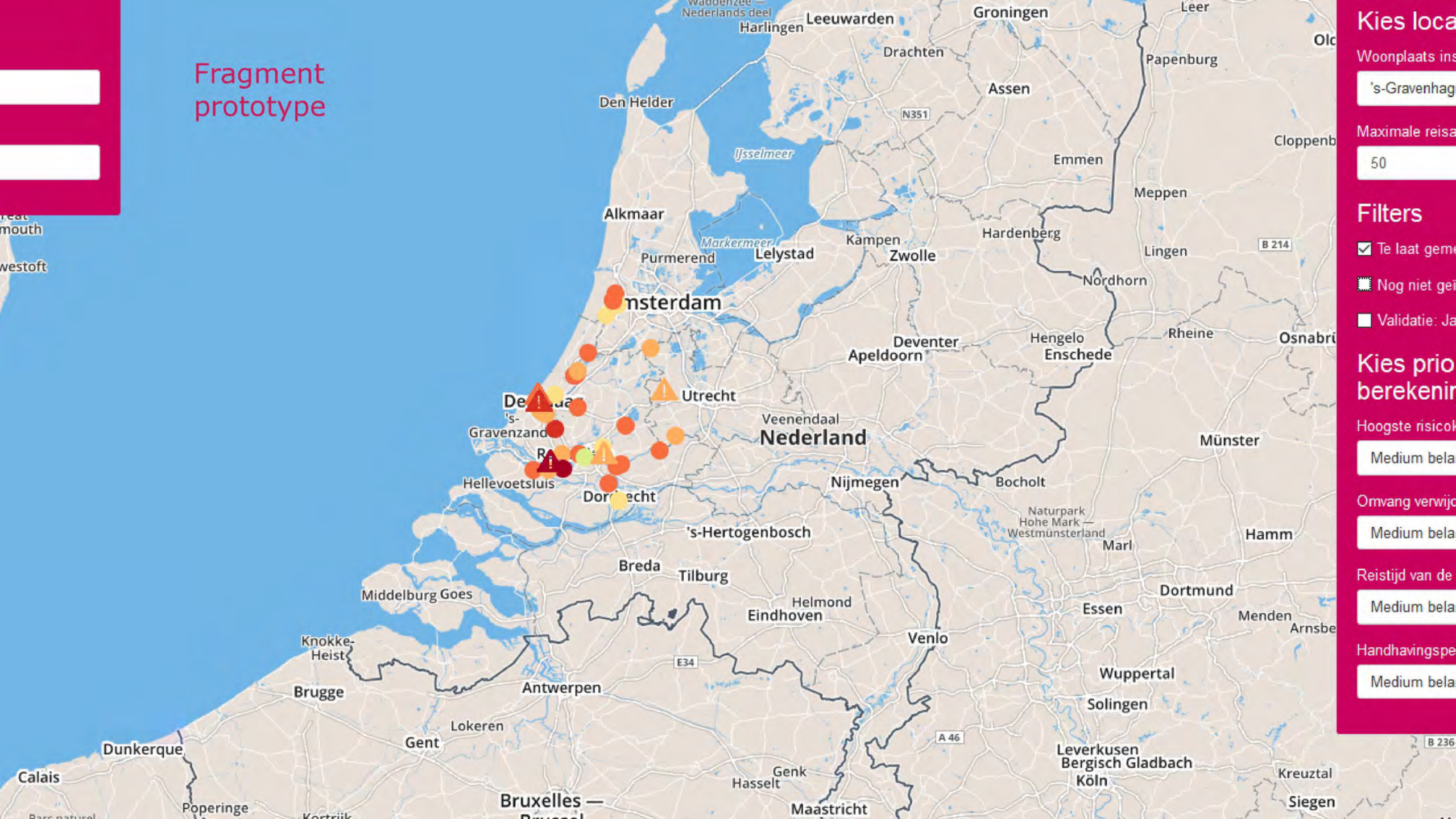
Planning asbest-inspectie

Uitvoering asbestinspectie

Risicoscore voor elke “klus”



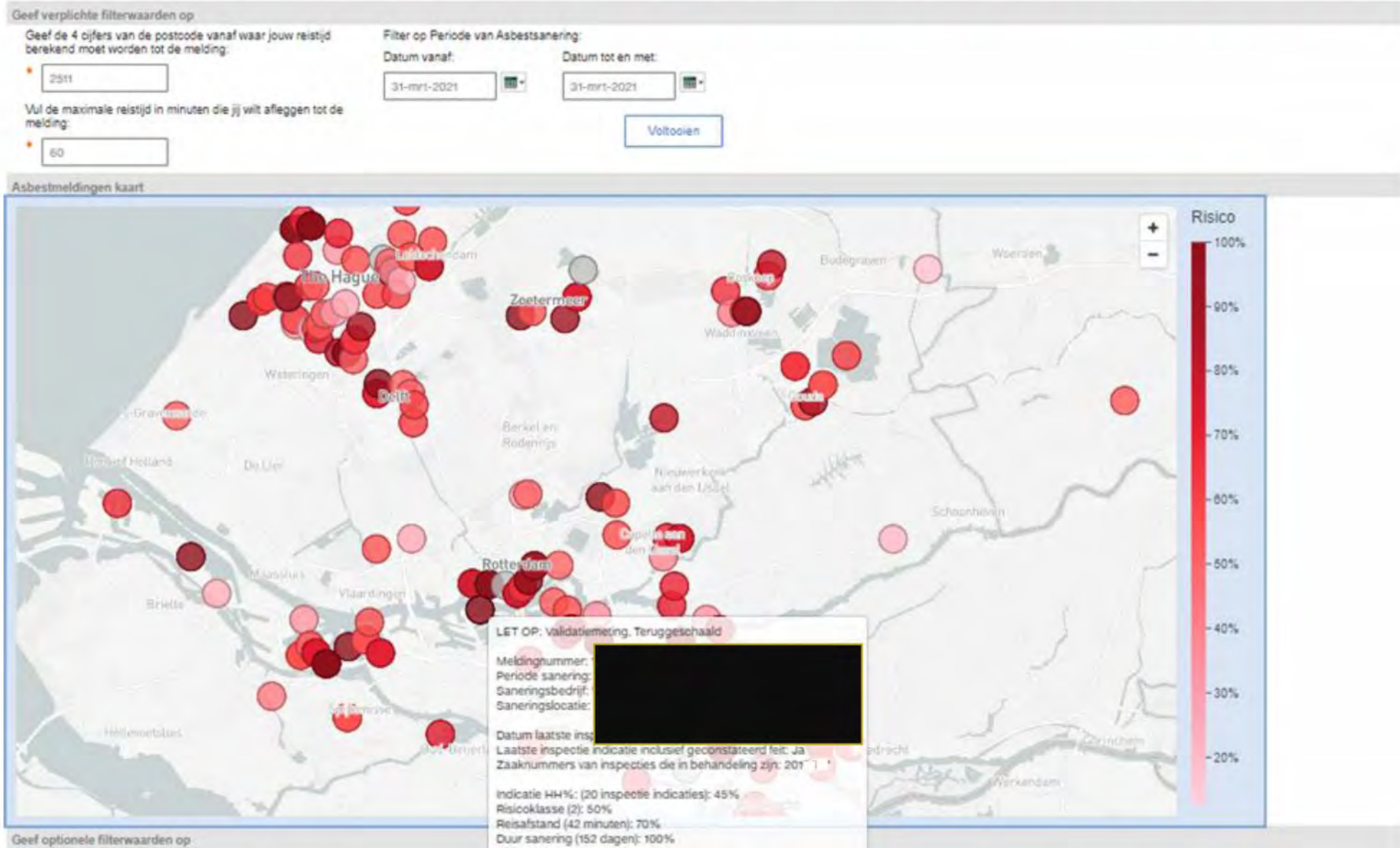

Fragment prototype
Kies loca
Woonplaats ins
's-Gravenhag
Maximale reisa
50
Filters
Te laat geme
Nog niet gei
Validatie: Ja
Kies prio
berekenin
Hoogste risicok
Medium bela
Omvang verwijd
Medium bela
Reistijd van de
Medium bela
Handhavingspe
Medium bela
Woubrugge
Ter Aar
Wassenaar
Voorschoten
Alphen aan den Rijn
Hazerswoude-Rijndijk
Hazerswoude-Dorp
Zwammerd
Leidschendam
Wilsveen
Benthuizen
Boskoop
Waddinxveen
Reeuwijk
Nootdorp
Monster
Wateringen
Pijnacker
Honselersdijk
's-Gravenzande
Delfgauw
Bleiswijk
Zevenhuizen
Gouda
't Woudt
Berkel en Rodenrijs
Oude Leede
Moordrecht
Hoek van Holland
De Lier
Schipluiden
Rotte
Nieuwerkerk aan den IJssel
Berkenwoude
Maasland
Kruiningergors
Oostvoorne
Vlaardingen
Schiedam
Rotterdam
Capelle aan den IJssel
Krimpen aan den IJssel
Tinte
Vierpolders
Rockanje
Bleskensgraaf


Fragment prototype
Kies loca
Woonplaats ins
's-Gravenhag
Maximale reisa
50
Filters
Te laat geme
Nog niet gei
Validatie: Ja
Kies prio
berekenir
Hoogste risicok
Medium bela
Omvang verwijd
Medium bela
Reistijd van de
Medium bela
Handhavingspe
Medium bela
Harlingen
Leeuwarden
Groningen
Drachten
Assen
Papenburg
Den Helder
N351
IJsselmeer
Emmen
Meppen
Alkmaar
Markermeer
Purmerend
Lelystad
Kampen
Zwolle
Hardenberg
Lingen
B 214
Nordhorn
Amsterdam
Deventer
Apeldoorn
Hengelo
Enschede
Rheine
Utrecht
's-Gravenzande
Veenendaal
Nederland
Münster
Hellevoetsluis
Nijmegen
Bocholt
's-Hertogenbosch
Naturpark Hohe Mark — Westmünsterland
Marl
Hamm
Breda
Tilburg
Middelburg
Goes
Helmond
Eindhoven
Dortmund
Essen
Venlo
Knokke-Heist
E34
Brugge
Antwerpen
Wuppertal
Solingen
Lokeren
Gent
A 46
Dunkerque
Calais
Leverkusen
Bergisch Gladbach
Köln
Genk
Hasselt
Kreuztal
Siegen
Bruxelles —
Maastricht
Poperinge

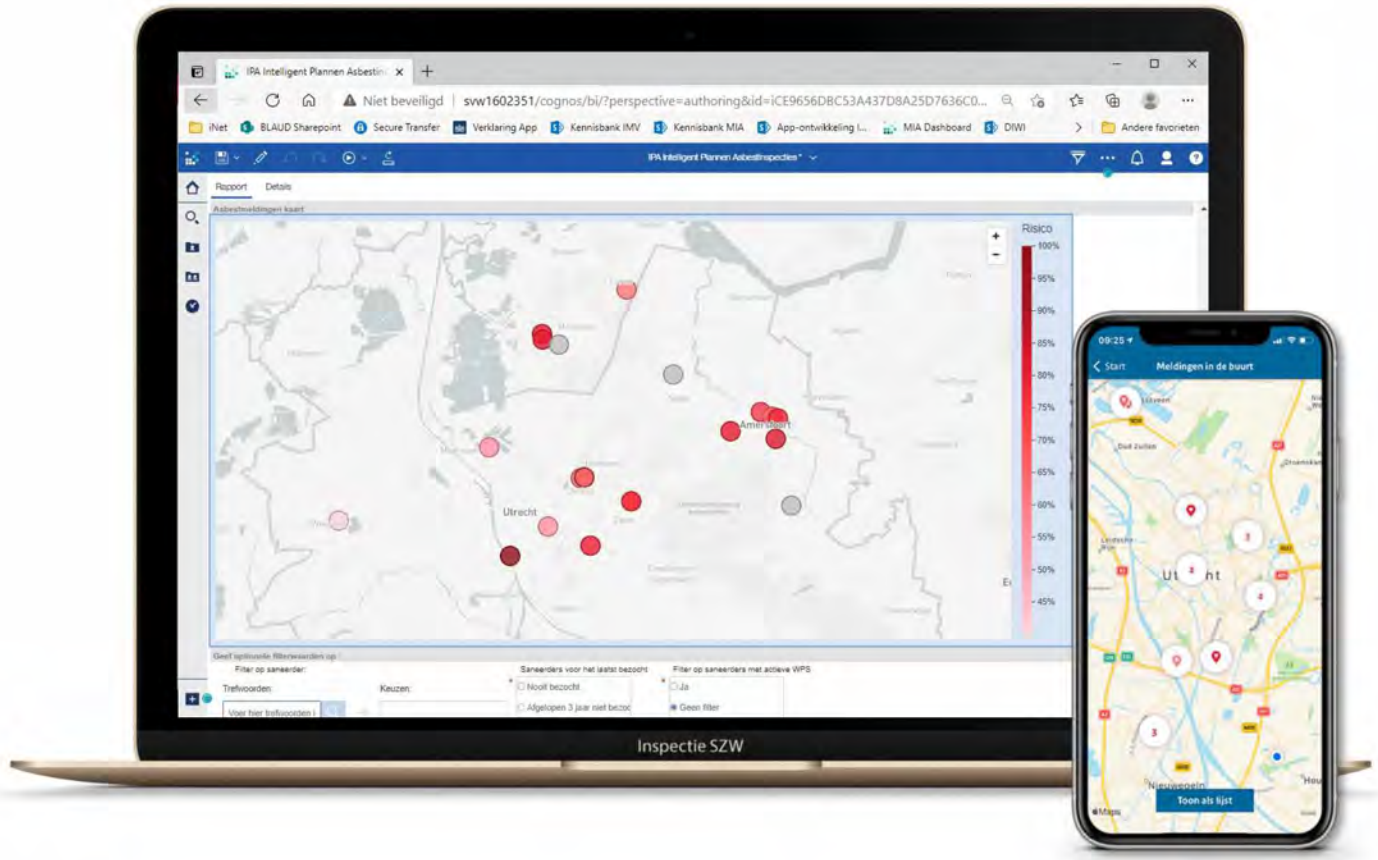
Opbouw risicoprofiel

- Bereken voor elke asbestverwijdering een risicoscore tussen 0 en 1
- Elementen die score in 1e versie risicoprofiel bepalen:
  - Handhavingspercentage in het verleden
  - Doorlooptijd / omvang opdracht
  - Reisafstand saneerder – locatie asbestsanering
  - Risicoklasse sanering

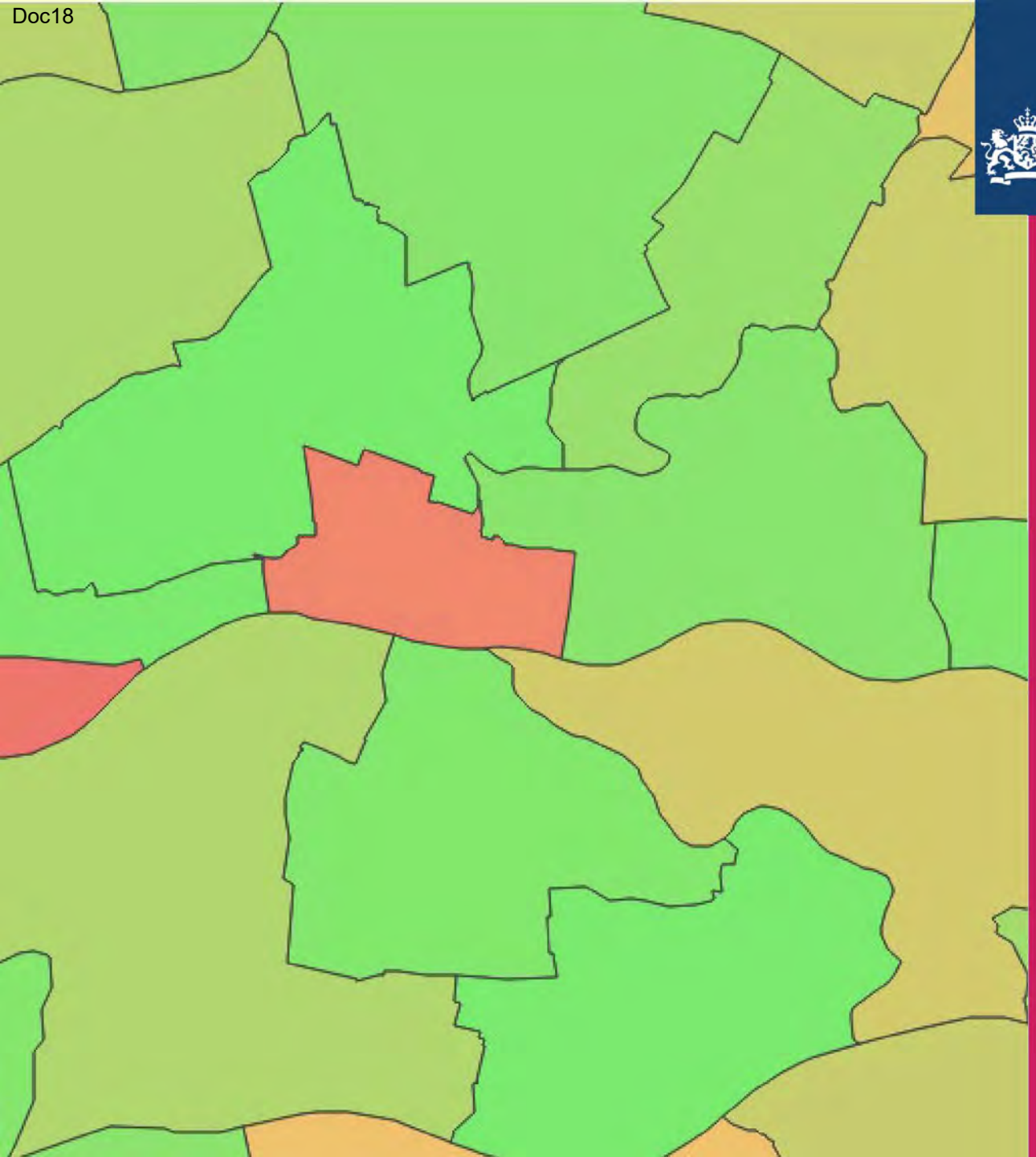
Hoe ziet de oplossing er in productie uit?



Inmiddels ook een mobiele versie beschikbaar



Doc18



Inspectie SZW
Ministerie van Sociale Zaken en Werkgelegenheid

Wat & Hoe LIBRA

Inspectie SZW
Ministerie van Sociale Zaken en Werkgelegenheid

LIBRA – LSI integraal Brede Risico Analyse

Visualisatie van ‘misstanden’ op wijkniveau

Onderwerpen

Achtergrond

Wat is Libra?

Gebruikstoepassingen

Niet één objectieve waarheid

Doorontwikkeling

Jullie ideeën

Achtergrond

- Samenwerking Min SZW, inspectie SZW, SVB, UWV, IND en Belastingdienst/Toeslagen.
- Behoefte aan data-gedreven LSI.
- Hoe gaan we de samenwerking aan, met welke data?

Wat is de Libra kaart niet?

- Zelf geen voorspellend model over misstanden:
  - Kaart geeft enkel aangeleverde data weer
  - Geen Machine Learning algoritme

Gebruik

- Detecteren van wijken met kennelijke misstanden op de LSI aandachtsgebieden (Sociale zekerheid, Toeslagen, Immigratie en Arbeid). Vervolgens behulpzaam bij:

  - **Doel 1:** Inzichten delen/bespreken. Hieruit kan samenwerking voortkomen

  - **Doel 2:** Opstellen van (meerjaren)plannen

Verantwoordelijkheid gebruiker

- Zelf goed nadenken welke fenomenen en scores relevant zijn
- Herken ik het beeld op de kaart?
- Verdiepende analyses (laten) uitvoeren
- Handhavingsstrategie maken

tudio

Open in Browser

SI | Zelf samenstellen | Zelf samenstellen Downdrill

historisch

AOW ☐ ANW ☐ AIO

voorspellend

AOW ☑ ANW ☑ AIO

BD/T historisch

☐ HT ☐ KOT

BD/T voorspellend

☑ HT ☑ KOT

UWV historisch

☐ WW

UWV voorspellend

☑ WW

IND

☑ gefingeerde dienstverbanden ☑ ID fraude

Genereer kaart

Inspectie

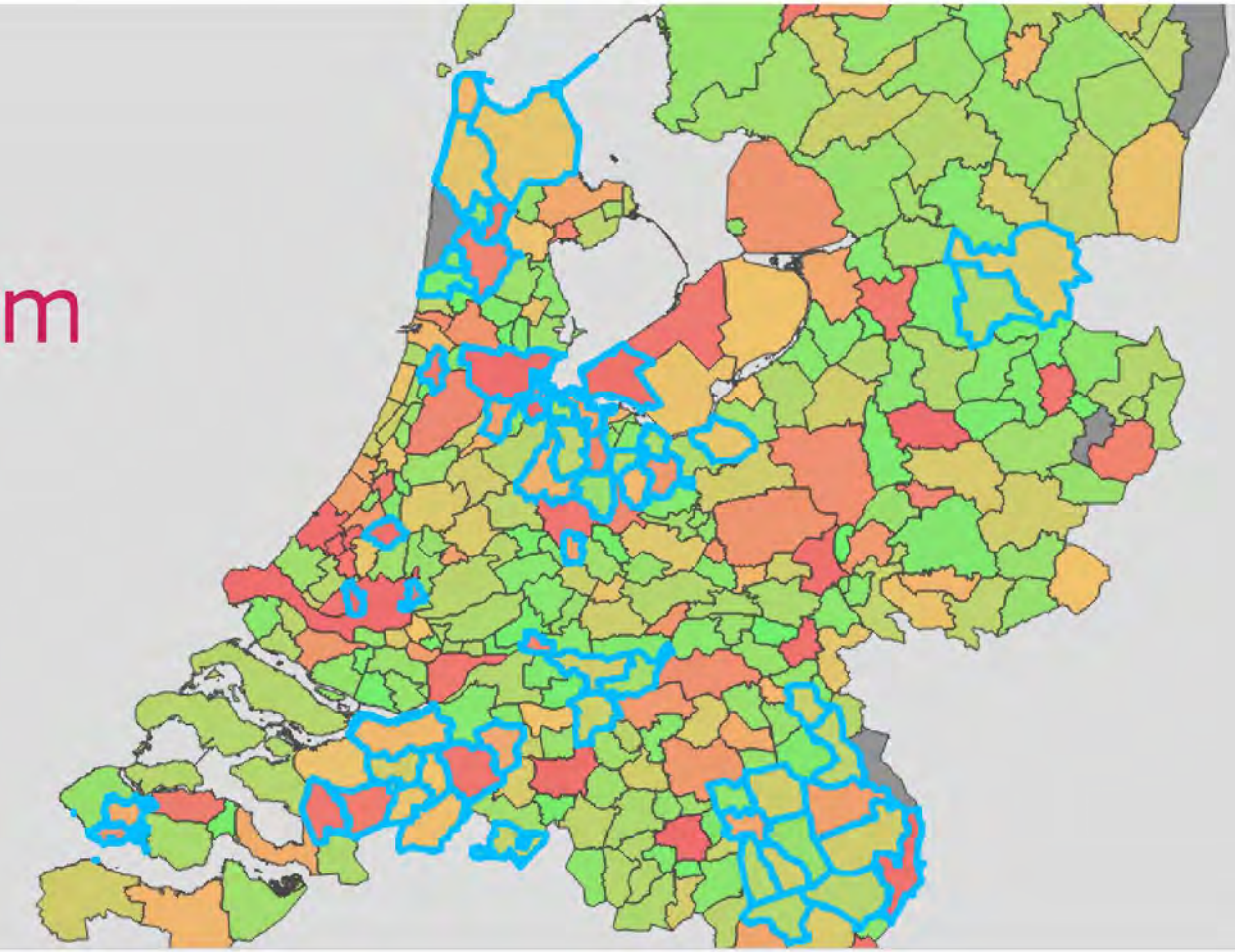
☑ ARBO ☑ AMF

☑ Uitgevoerde LSI projecten

er gemeente

+

−



Relatieve risico

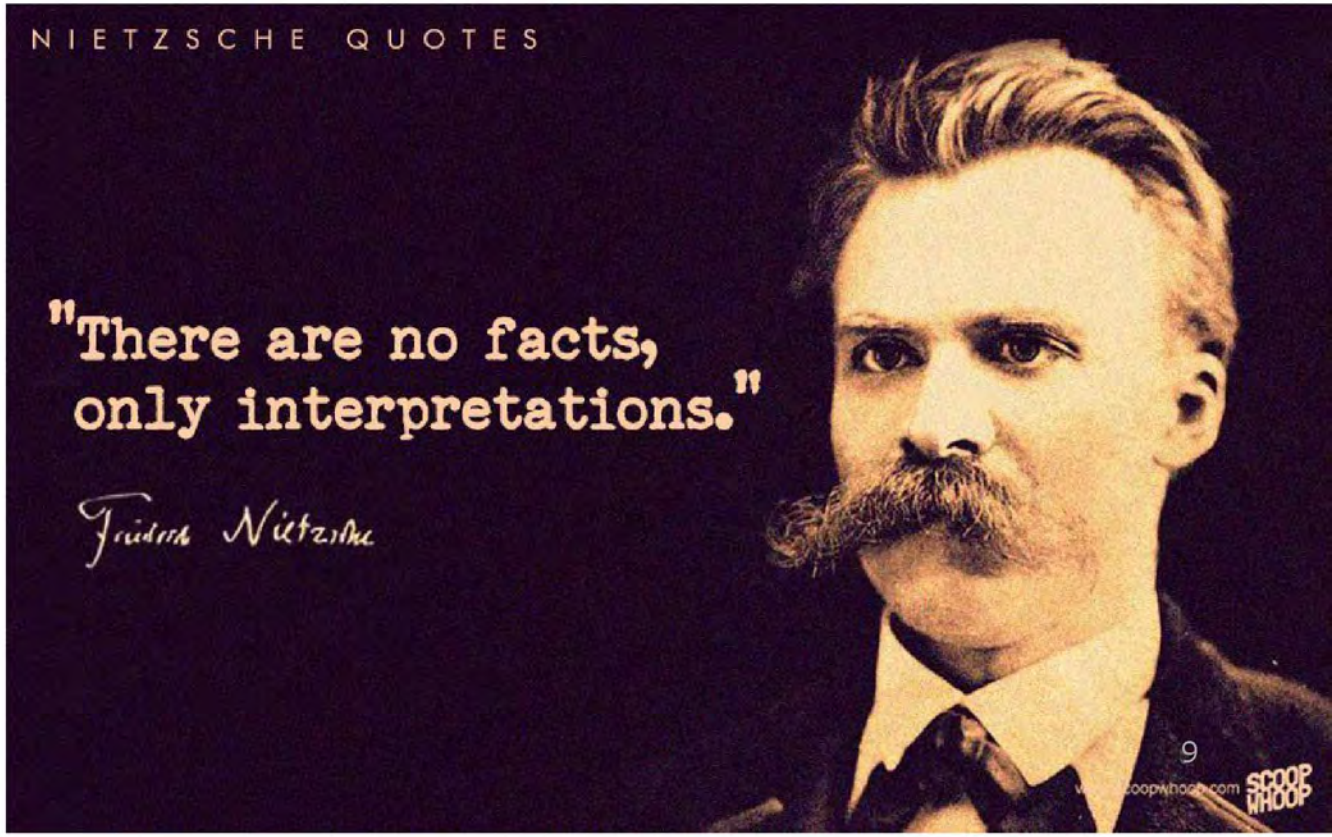
Hoog risico

Laag risico

Voorbeeldscherm Dashboard

Leaflet | ©

Let op: de invalshoek kleurt de kaart



Aanleiding: bezoek aan Belgische Sociale Inspectie in Brussel

Belgen waren er in geslaagd state-of-the-art datagestuurde risicomodellen en inspectiepraktijk met elkaar te verbinden

Wij leerden van dit bezoek:

Centraal opleggen adressen vanuit Den Haag maakt inspecteurs ontevreden

Communiceren via lijsten met de inspecteurs werkt niet

Waar staat PILS voor?

PILS staat voor **P**rofileren met de **I**nspecteurs in de **L**ead en **S**amenwerking

Belangrijkste kenmerken van deze inspectiewerkwijze:

Inspecteurs staan zelf centraal in de ontwikkeling van risicoprofielen

Gebruikersvriendelijke overdracht van adressen aan inspecteurs door middel van informatieschil

Inspecteurs hebben maximale keuzevrijheid

Ontwikkeling modellen met modernste datagestuurde technieken

Samenwerkingsvorm

Data-Scientists	ICT-ers	Inspecteurs
Methodiek Profilering	Informatieschil	Ervaringskennis Daadwerkelijke inspecties

Centraal bij ontwikkeling modellen: keuzevrijheid

Keuzevrijheid op twee niveaus: tussen modellen, en op niveau van indicatoren

Meerdere modellen ontwikkelen, inspecteurs laten kiezen

Binnen modellen: inspecteurs kunnen indicatoren bedenken, goedkeuren of afkeuren

Inspecteurs krijgen de uitkomsten van de modellen op verschillende manieren gepresenteerd:

Top-zoveel adressen

Belangrijkste indicatoren model

Waardevolle databronnen voor de risicomodellen

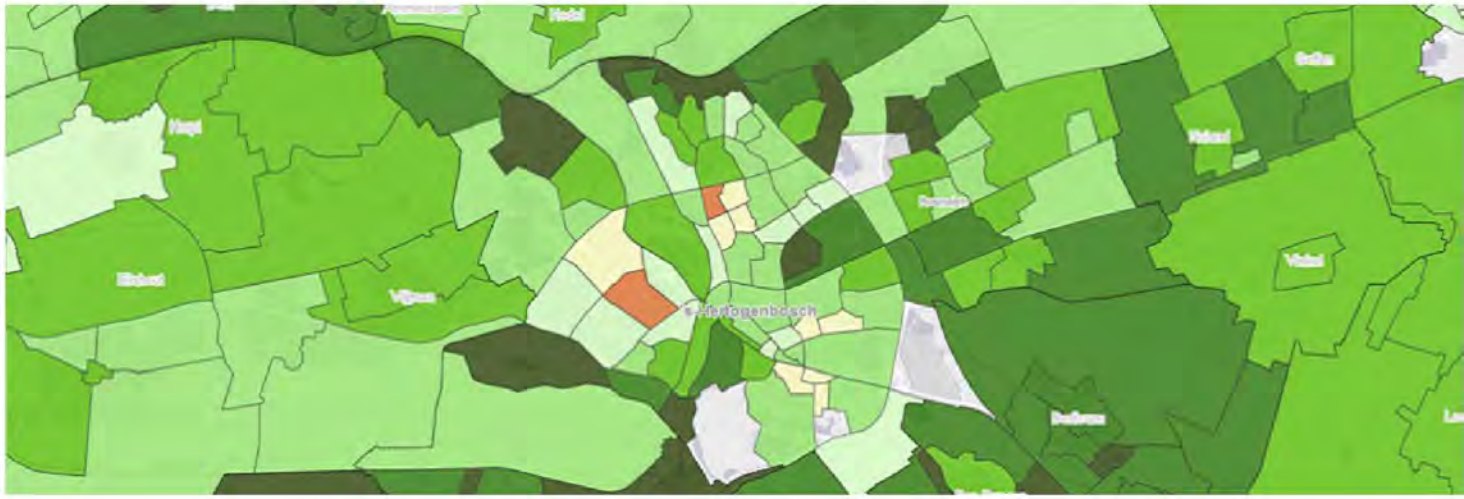
Databronnen zijn een combinatie van overheidsbronnen en open data

Voorbeelden van waardevolle databronnen:

UWV Polisadministratie

NVWA

Leefbaarheidsbarometer



PILS

Inspecteurs kunnen via informatieschil adressen selecteren

Werklijst

Horeca AMF - 17-10-2017 | Filter | ☐ Toon alleen geselecteerde

Vestigingnaam	Vestigingnummer	Plaatsnaam	Sector	Score	Afsluitdatum zaak
[redacted]		Nieuwegein	56101 -Restaurants	0.904	27-02-2017
		GOUDA	56101 -Restaurants	0.866	30-03-2017
		Amsterdam	56101 -Restaurants	0.847	20-06-2015
		Oirschot	56101 -Restaurants	0.844	20-03-2018
		Amsterdam	56101 -Restaurants	0.841	27-10-2016
		's-Gravenhage	56101 -Restaurants	0.834	22-05-2017
		EINDHOVEN	56101 -Restaurants	0.833	14-12-2015
		GRONINGEN	6420 -Financiële holdings	0.832	28-10-2015
		Den Helder	56101 -Restaurants	0.829	18-11-2016
		's-Gravenhage	56101 -Restaurants	0.827	23-11-2017

« 1 2 3 4 5 »

10 100 1000

```
### Cleaning of the dataset, in a number of phases

# 1. Delete variables which are not of interest, like names, id's etc.
# 2. Calculate some new fields
# 3. Convert characters to factor, and reduce the number of levels, and
make na "other"
# 4. Substitute missings with eiter mean or median

schonen <- function(data) {

df <- data

library(tidyverse)
library(lubridate)

#En nu eerst de belangrijke variabelen selecteren, te weten de top-10 en
nog een paar zoals Rechtsvorm

### 1. Delete features which are not to be used, like id-numbers
df <- df %>%
  select( -Vestiging_SQN, -Vestiging_ID ) %>%
  select( -(Straat : HuisnrToev) ) %>%
  select( -(RSIN : ZZP) ) %>%
  select( -AMFgebied_afk, -AMFgebied) %>%
  select( -starts_with("Locatie")) %>%
  select( -(Gemeentecode:Veiligheidsregio) ) %>%
  select( -HoofdOfNevenVestiging, -RedenInschrijving, -
IndicatieFaillissement, -DatumAanvangFaillissement, -GeslachtEigenaar)
%>%
  select( -Overig, -HoofdSBIcode, -HoofdSBIafdeling, -IRA_code, -
Afdeling) %>%
  select( -Inspecties, -AantalJarenNietGeinspecteerdAMF, -InspectieAMF, -
AantalJarenNietGeinspecteerd, -AantalActieveEenmanszakenOpAdres, -
InspectieARBO, -ARBO, -Overtreding, -HoofdSBIcode_ext,
          -AantalFTE, -Leefbaarheid_totaal, -AantalFTEOnderneming, -
AantalAMFKlachtenLaatste5Jaar,
          -sbicode)

df <- select(df, -(`0001Bedrijven` : `13RisicoOpZwaarOngeval`) )


### 2. Make calculated fields: age of building, # of addresses in
postcode area, etc.

# First # of horeca adresses in postcode aream for both the complete
postcode and for the fist 4 digits
df_aantal_postcode <- df %>%
  mutate(Postcode = fct_explicit_na(Postcode, na_level = "Onbekend")) %>%
  group_by(Postcode) %>%
  summarise(Aantal_postcode = n()) %>%
  mutate(Aantal_postcode =  ifelse(Postcode == "Onbekend", NA,
Aantal_postcode))

df_aantal_postcode4 <- df %>%
  mutate(p4 = str_sub(Postcode,1, 4)) %>%
  mutate(p4 = fct_explicit_na(p4, na_level = "Onbekend")) %>%
  group_by(p4) %>%
  summarise(Aantal_postcode4 = n()) %>%
```

```
  mutate(Aantal_postcode4 =  ifelse(p4 == "Onbekend", NA,
Aantal_postcode4))

df <- df %>%
  left_join(df_aantal_postcode, by="Postcode") %>%
  mutate(p4 = str_sub(Postcode,1,4)) %>%
  left_join(df_aantal_postcode4, by="p4") %>%
  select(-p4)

# Derive the startyear of the company and the startyear of the address

df2 <- df %>%
  mutate(DatumInschrijvingOfOprichting = ymd
(DatumInschrijvingOfOprichting) ) %>%
  mutate(JaarInschrijvingOfOprichting  =
year(DatumInschrijvingOfOprichting) ) %>%
  mutate(DatumVestiging                = ymd (DatumVestiging) ) %>%
  mutate(JaarVestiging                 = year(DatumVestiging) ) %>%
  select(-DatumVestiging, -DatumInschrijvingOfOprichting)

### 3. Missing values substitution for (every feature involved

# Function to replace missings with mean, and one for binaries, to
subsitute them with 0

na_remove <- function(x) {
  y <- ifelse(is.na(x), mean(x, na.rm=TRUE), x)
  return(y)
}

na_zero <- function(x) {
  y <- ifelse(is.na(x), 0, x)
  return(y)
}

# substitute NAs with 0 for binary features
df2 <- df2 %>%
  mutate(WAV   = na_zero(WAV)) %>%
  mutate(WML   = na_zero(WML)) %>%
  mutate(Waadi = na_zero(Waadi)) %>%
  mutate(ATW  = na_zero(ATW)) %>%
  mutate(bijeenkomstfunctie = na_zero(bijeenkomstfunctie)) %>%
  mutate(celfunctie = na_zero(celfunctie)) %>%
  mutate(gezondheidszorgfunctie = na_zero(gezondheidszorgfunctie)) %>%
  mutate(industriefunctie = na_zero(industriefunctie)) %>%
  mutate(kantoorfunctie = na_zero(kantoorfunctie)) %>%
  mutate(logiesfunctie = na_zero(logiesfunctie)) %>%
  mutate(onderwijsfunctie = na_zero(onderwijsfunctie)) %>%
  mutate(overige_gebruiksfunctie = na_zero(overige_gebruiksfunctie)) %>%
  mutate(sportfunctie = na_zero(sportfunctie)) %>%
  mutate(winkelfunctie = na_zero(winkelfunctie)) %>%
  mutate(woonfunctie = na_zero(woonfunctie))

df3 <- df2 %>%
  mutate(LeeftijdPand = ifelse(LeeftijdPand>400,100, LeeftijdPand)) %>%
  mutate(PClon = if_else(is.na(PClon), median(PClon, na.rm=TRUE),
PClon))%>%
```

```
  mutate(PClat = if_else(is.na(PClat), median(PClat, na.rm=TRUE), PClat))
%>%
  mutate(Omgevingsadressendichtheid =
ifelse(is.na(Omgevingsadressendichtheid), 3091,
Omgevingsadressendichtheid ))


df3 <- df3 %>%
  mutate_if(is.numeric, na_remove)

### TWV features

# Character velden omzetten naar factor, en categorie "overige" maken.

library(forcats)

df3 <- df3 %>%
  mutate_if(is.character, as.factor)

gm_fac <- function(categorie, aantal_cat) {
  categorie <- fct_lump(categorie, n=aantal_cat, other_level="Overige")
  categorie <- fct_explicit_na(categorie, na_level = "Onbekend")
  return(categorie)
}

### Condense number of levels of relevant factor variables
df4 <- df3 %>%
  mutate(Rechtsvorm      = gm_fac(Rechtsvorm, aantal_cat = 10) ) %>%
  mutate(Gemeentegrootte = gm_fac(Gemeentegrootte, aantal_cat = 10) ) %>%
  mutate(Stedelijkheid   = gm_fac(Stedelijkheid, aantal_cat = 5) )  %>%
  mutate(FamilieBedrijfsnaam   = gm_fac(FamilieBedrijfsnaam, aantal_cat =
3) )

delta <- function(a,b) {
  verschil = a - b
  return(verschil)
  }

# Calculate change between 2013 and 2015 and throw the 2013 variables
away

df4 <- df4 %>%
  mutate(DeltaLoon           = delta(Uurloon2015, Uurloon2013) ) %>%
  mutate(DeltaWinst          = delta(winst2015, winst2013) )     %>%
  mutate(DeltaArbeidsintens  = delta(LNarbeidsintens2015,
LNarbeidsintens2013) ) %>%
  mutate(DeltaLNpercbanen    = delta(LNpercbanen_130WML2015
,LNpercbanen_130WML2013 ) ) %>%
  mutate(DeltaPercNietWestMOE = delta(LNPercBanenNietwestMOEOostEur2015,
LNPercBanenNietwestMOEOostEur2013) ) %>%
  mutate(DeltaPiek           = delta(piekwaarde2015_cat,
piekwaarde2013_cat) ) %>%
  select(- Uurloon2013, -winst2013, -LNarbeidsintens2013, -
LNPercBanenNietwestMOEOostEur2013, -piekwaarde2013_cat, -
LNpercbanen_130WML2013)


### Make a new feature for labour fraud
```

```
### Count number of violations

### Determine if it is a violator, based on the number of violations and
the size of the company

#  1 overtreding & aantal werknemers vestiging <= 100: AMF overtreder
#  2 overtredingen & aantal werknemers vestiging > 100 maar <500: AMF
overtreder
#  3 overtredingen & aantal werknemers vestiging > 500 < 1000: AMF
overtreder
#  4 overtredingen & aantal werknemers vestiging >1000: AMF overtreder
#  alle overige vestigingen: geen AMF overtreder

df4 <- df4 %>%
  mutate(no_of_violations = WAV + WML + ATW + Waadi) %>%
  select(-WAV, -WML, -ATW, -Waadi)

df4 <- df4 %>%
      mutate(ovt = 0) %>%
      mutate(ovt = ifelse(AantalWerknemers < 100 & no_of_violations   >
0, 1, ovt)) %>%
      mutate(ovt = ifelse(AantalWerknemers >= 100 & no_of_violations  >
1, 1, ovt)) %>%
      mutate(ovt = ifelse(AantalWerknemers >= 500 & no_of_violations  >
2, 1, ovt)) %>%
      mutate(ovt = ifelse(AantalWerknemers >= 1000 & no_of_violations >
3, 1, ovt)) %>%
      select(-no_of_violations)

df4 <- mutate(df4, JaarVestiging = as.integer(JaarVestiging))

data_schoon <- df4

return(data_schoon)

}
```