hm9

Robert Benke

25 maja 2019

Wczytanie danych

Dane zostały podzielone na trzy części, pierwsza przeznaczona do budowania modelu zawiera początkowe 10 tysięcy obserwacji. Druga to kolejne 10 tysięcy obserwacji i wykorzystana będzie do walidacji modelu. Ostatnie 20 tysięcy obserwacji przeznaczone jest na testowanie modelu i jego stabilności w czasie. Pomiędzy zbiorem testowym a validacyjnym pominiętych zostało 160 tysiecy obserwacji.

```
data_dfr <- read.csv("rotatingHyperplane.txt", sep = " ", header = FALSE)
labels_dfr <- read.csv("rotatingHyperplane.labels.txt", header = FALSE)

train_data_dfr <- data_dfr[1:10000,]
train_labels_vec <- labels_dfr$V1[1:10000]
train_data_dfr <- mutate(train_data_dfr,labels = train_labels_vec)

test_data_dfr <- data_dfr[10001:20000,]
test_labels_vec <- labels_dfr$V1[10001:20000]
test_data_dfr  <- mutate(test_data_dfr ,labels = test_labels_vec)

val_data_dfr <- data_dfr[180000:199999,]
val_labels_vec <- labels_dfr$V1[180000:199999]
val_data_dfr  <- mutate(val_data_dfr ,labels = val_labels_vec)
```

Intersection distance dla każdej zmiennej

Do nauki modelu wykorzystane zostało dziesięć zmiennych. Poniżej znajdują się wyniki analizy stacjonarności rozkładu zmiennych objaśniających (w ujęciu jednowymiarowym).

```
Intersection <- function(values1, values2, breaks = seq(0,1,0.1)){

values1_categorical <- cut(values1, breaks = breaks) %>% table %>% '/'(length(values1))
values2_categorical <- cut(values2, breaks = breaks) %>% table %>% '/'(length(values2))

(sapply(1:(length(breaks)-1),
        function(i) min(values1_categorical[i], values2_categorical[i]))) %>% sum
}
```

```
for (i in 1:10) {
  print(paste0("Intersection zmiennej V", i))
print(1 -Intersection(as.vector(test_data_dfr[,i]), as.vector(val_data_dfr[,i])))
}
```

```
## [1] "Intersection zmiennej V1"
## [1] 0.01195
## [1] "Intersection zmiennej V2"
## [1] 0.0117
## [1] "Intersection zmiennej V3"
```

```
## [1] 0.01695
## [1] "Intersection zmiennej V4"
## [1] 0.0197
## [1] "Intersection zmiennej V5"
## [1] 0.0183
## [1] "Intersection zmiennej V6"
## [1] 0.02045
## [1] "Intersection zmiennej V7"
## [1] 0.007
## [1] "Intersection zmiennej V8"
## [1] 0.01765
## [1] "Intersection zmiennej V9"
## [1] 0.01335
## [1] "Intersection zmiennej V10"
## [1] 0.0134
```

Wszystkie wyniki są bliskie zera. Możemy zatem przyposzczać, że rozkład zmiennych nie zmienił się. W dalszej części porównanane zostaną rozkłady reszt dla danych z początku i końca badanego okresu.

Model 1

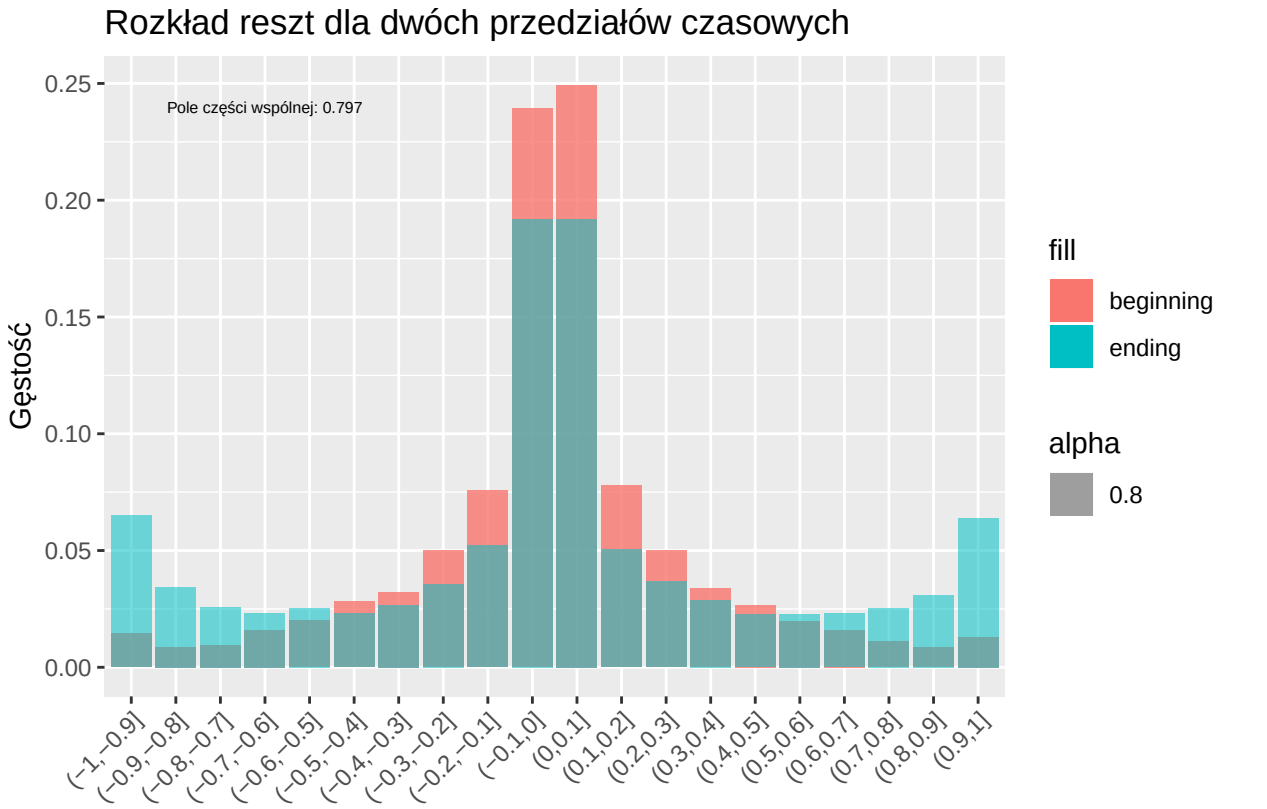
```
model1 <- glm(labels~., data = train_data_dfr, family=binomial(link="logit"))
model1
```

```
##
## Call:  glm(formula = labels ~ ., family = binomial(link = "logit"),
##     data = train_data_dfr)
##
## Coefficients:
## (Intercept)           V1           V2           V3           V4
##     -17.329        3.828        2.936        1.538        2.249
##          V5           V6           V7           V8           V9
##       4.532        2.029        4.836        3.533        4.630
##         V10
##       4.520
##
## Degrees of Freedom: 9999 Total (i.e. Null);  9989 Residual
## Null Deviance:       13860
## Residual Deviance: 6596  AIC: 6618
```

Residuals distance

```
beg_resid <- predict(model1, test_data_dfr[,-11], type="response") - test_data_dfr$labels

end_resid <- predict(model1, val_data_dfr[,-11], type="response") - val_data_dfr$labels

breaks <- seq(-1,1,0.1)

beg_resid_categorical <- cut(beg_resid, breaks = breaks) %>% table %>% '/'(length(beg_resid))
end_resid_categorical <- cut(end_resid, breaks = breaks) %>% table %>% '/'(length(end_resid))
```

```r
intesection <- (sapply(1:20,
                       function(i) min(beg_resid_categorical[i], end_resid_categorical[i]))) %>% sum

ggplot()+
  geom_col(aes(x = factor(names(beg_resid_categorical),
            levels = names(beg_resid_categorical)), y = beg_resid_categorical,
            fill = "beginning"), alpha = 0.8) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_col(aes(x = factor(names(end_resid_categorical),
          levels = names(end_resid_categorical)), y = end_resid_categorical,
          fill = "ending", alpha = 0.8)) +
  xlab("") + ylab("Gęstość") +
  ggtitle("Rozkład reszt dla dwóch przedziałów czasowych") +
  geom_text(size = 2.2, aes(x = 4, y = 0.24,
            label = paste0("Pole części wspólnej: ",
            round(intesection, digits = 3))))
```



Różnice reszt w dwóch okresach czasu są wyraźnie różne. Model w późniejszym okresie posiada znacznie więcej błędów skrajnych (przedwidywanie jednej z kategorii z dużym prawdopodobieństwem, gdy w rzeczywistości obserwacja należy do przeciwnej kategorii), oraz znacznie rzadziej otrzymujemy błędy bliskie zera.

Pole pomiędzy krzywymi PDP dla obu modeli.

Model 2

```
model2 <- glm(labels~., data = val_data_dfr, family=binomial(link="logit"))
model2
```

```
##
## Call:  glm(formula = labels ~ ., family = binomial(link = "logit"),
##     data = val_data_dfr)
##
## Coefficients:
## (Intercept)           V1           V2           V3           V4
##     -5.5148       3.5578       1.4117      -2.2152      -1.7999
##          V5           V6           V7           V8           V9
##      6.9578      -4.0738       2.5661       1.3532       3.6699
##         V10
##     -0.4149
##
## Degrees of Freedom: 19999 Total (i.e. Null);  19989 Residual
## Null Deviance:       27720
## Residual Deviance: 13990     AIC: 14010
```

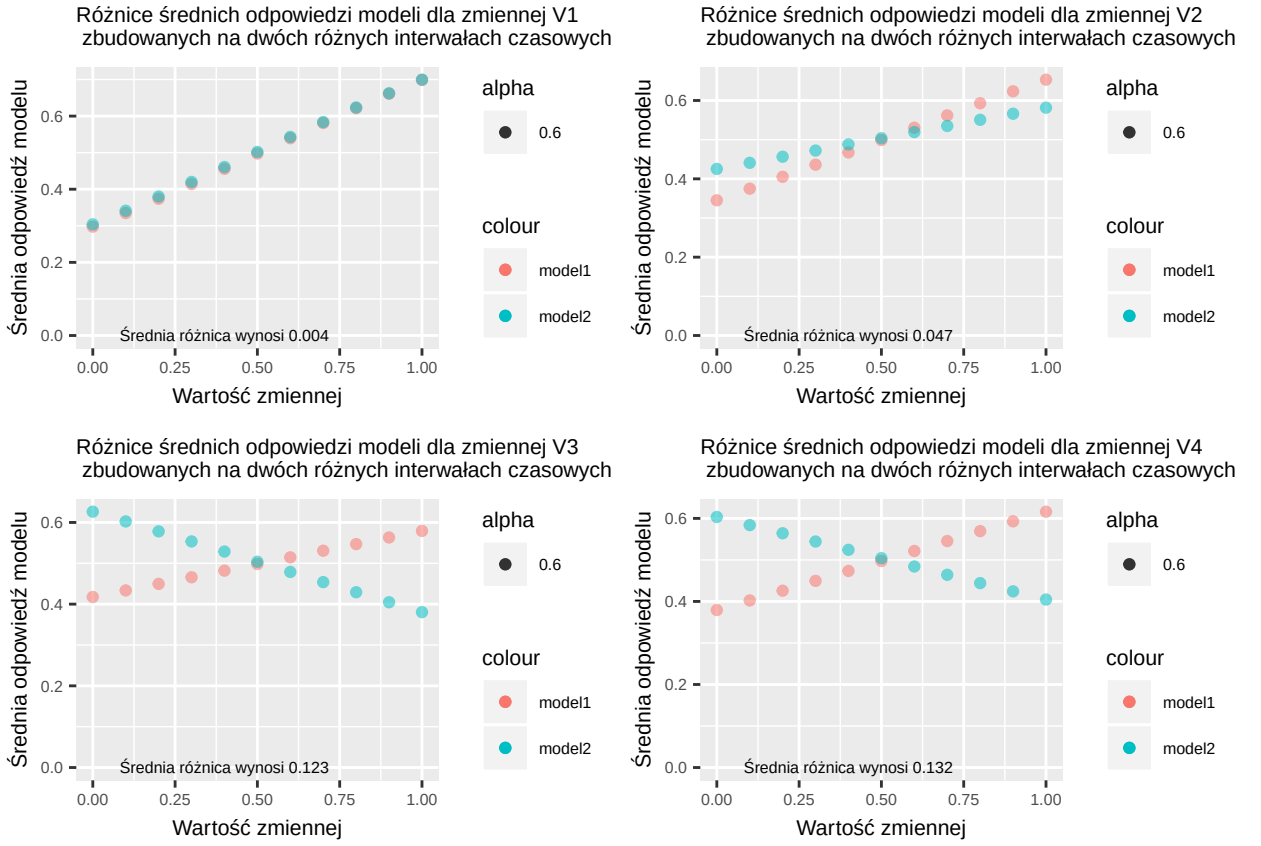
Już na etapie budowy modelu widać różnice w estymowanych współczynnikach. Model pierwszy posiadał wszsytkie współczynniki dodatnie, natomiast w modelu drugim aż cztery z nich posiadają wartości ujemne. Zmienna druga i ósma mają współczynniki o połowe niższe, a wyraz wolny znalał ponad trzykrotnie.

PDP

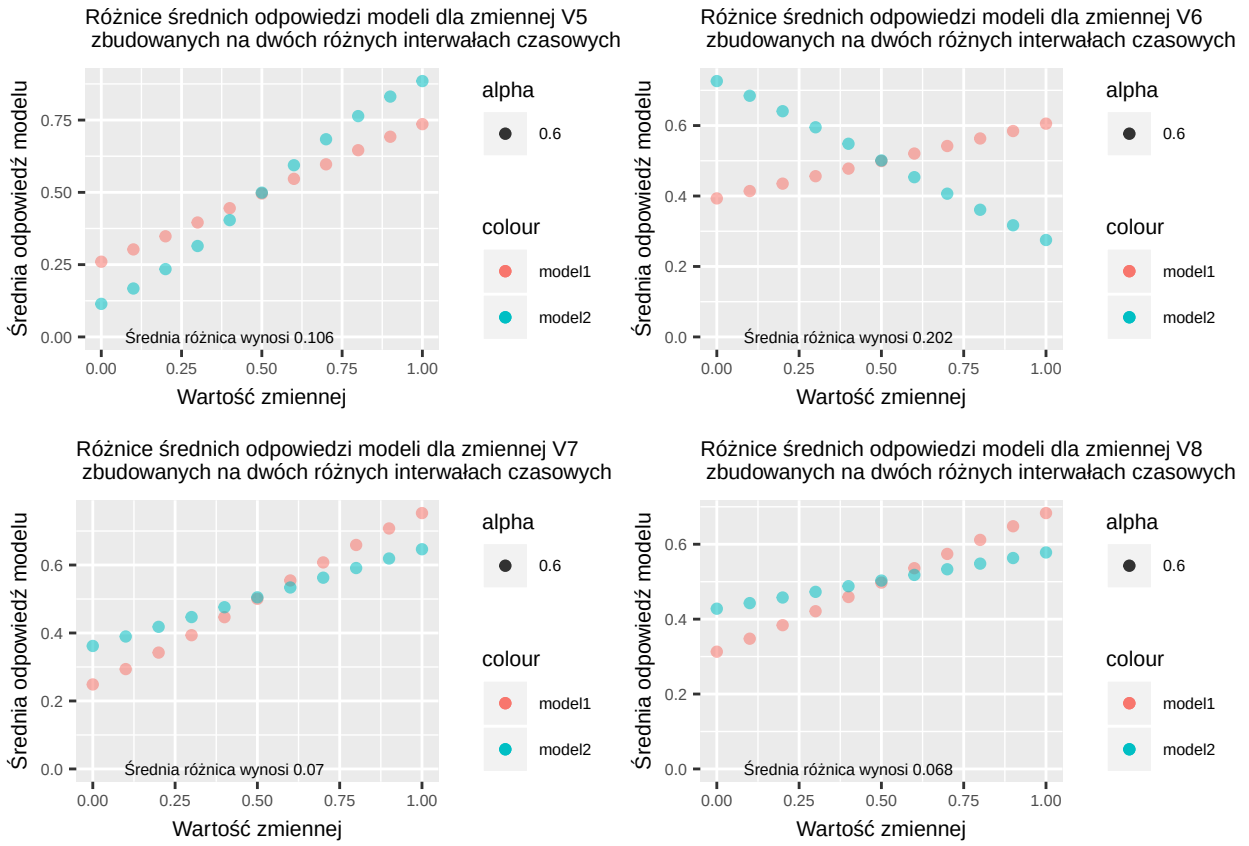
```
PDP <- function(data, model, var_id, breaks){
  sapply(breaks, function(x){
    data_x <- data[,-11]
    data_x[,var_id] <- x
    mean(predict(model, data_x, type = "response"))
  })
}
```

```
breaks = seq(0,1,0.1)
pdp_diff_varX <- lapply(1:10, function(x){
  pdp_model1 <- PDP(val_data_dfr, model1, x, breaks)
  pdp_model2 <- PDP(val_data_dfr, model2, x, breaks)
  diff <- (abs(pdp_model1 - pdp_model2)/10) %>% sum
  ggplot()+ geom_point(aes(x = breaks, y = pdp_model1, colour = "model1", alpha = 0.6)) +
    geom_point(aes(x = breaks, y = pdp_model2, colour = "model2", alpha = 0.6)) +
    geom_text(size = 2, aes(x=.4, y=0,
              label = paste0("Średnia różnica wynosi ",
              round(diff, digits = 3)))) +
    ggtitle(paste0("Różnice średnich odpowiedzi modeli dla zmiennej V",
                   x,"\n zbudowanych na dwóch różnych interwałach czasowych")) +
    xlab("Wartość zmiennej") + ylab("Średnia odpowiedź modelu") +
    theme(plot.title = element_text(size=8),text = element_text(size=8))

})
```

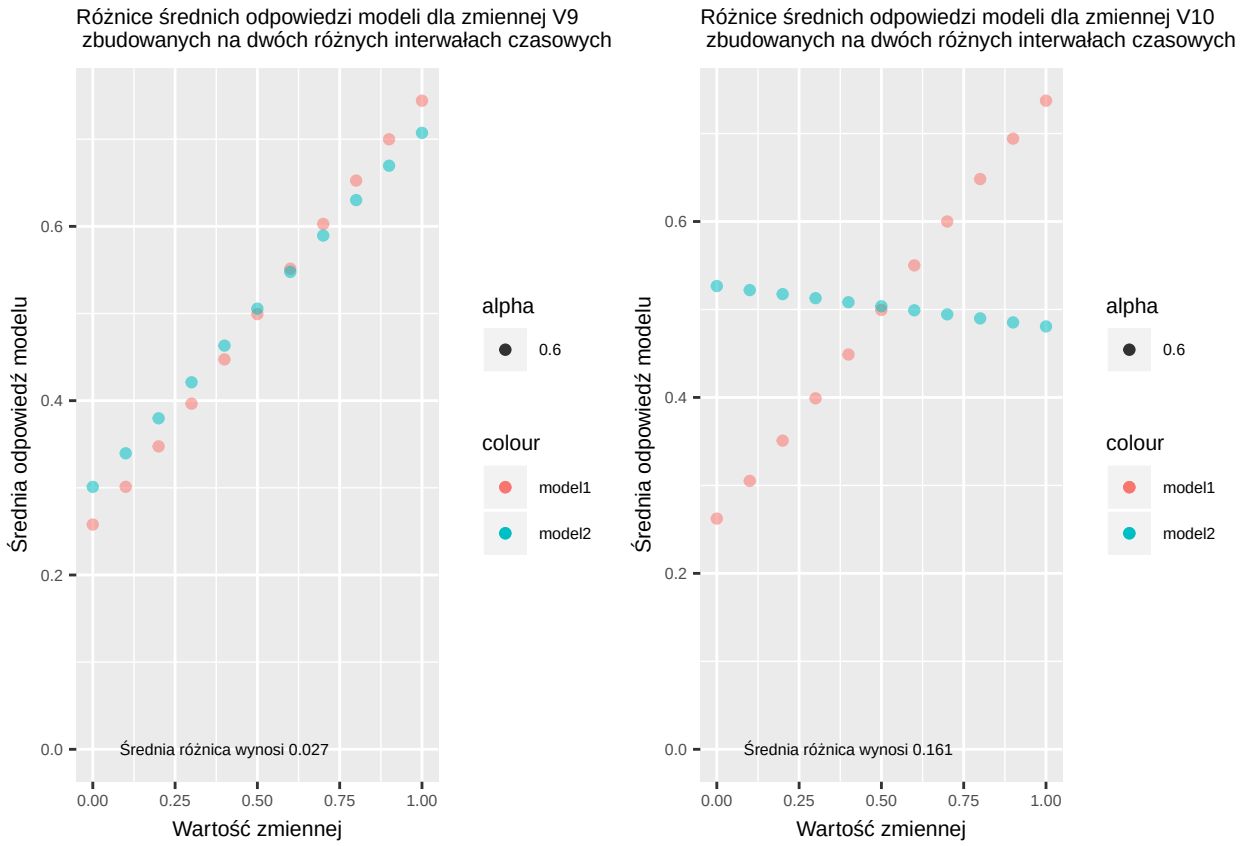
```
grid.arrange(pdp_diff_varX[[1]],pdp_diff_varX[[2]],
             pdp_diff_varX[[3]],pdp_diff_varX[[4]],
             ncol = 2)
```




```
grid.arrange(pdp_diff_varX[[5]],pdp_diff_varX[[6]],
             pdp_diff_varX[[7]],pdp_diff_varX[[8]],
             ncol = 2)
```




```
grid.arrange(pdp_diff_varX[[9]],pdp_diff_varX[[10]],
             ncol = 2)
```



Wnioski:

Rozkłady brzegowe nie definiują rozkładu łącznego, dlatego mogliśmy zaobserwować brak zmian w rozkładach zmiennych objaśniających w czasie, jednocześnie obserwując spadek jakości modelu i relacji pomiedzy zmiennymi objaśniającymi i zmienną objaśnianą.