

# Wstęp do Uczenia Maszynowego 2020: projekt I, kamień milowy III

## - Regresja

Marcin Łukaszyk

April 18, 2020

### Wczytanie Danych

```
Reg_train <- read.csv("Reg_train.csv")
Reg_test <- read.csv("Reg_test.csv")
Reg_train <- select(Reg_train, -X)
Reg_test <- select(Reg_test, -X)

knitr::kable(sample_n(Reg_test, 5))
```

duration	credit_amount	installment_rate	present_residence	age	existing_credits	dependents	has_telephone
24	2611	4	3	46	2	1	0
6	1338	1	4	62	1	1	0
24	2375	4	2	44	2	2	1
12	795	4	4	53	1	1	0
15	3594	1	2	46	2	1	0

```
knitr::kable(sample_n(Reg_train, 5))
```

duration	credit_amount	installment_rate	present_residence	age	existing_credits	dependents	has_telephone
42	3965	4	3	34	1	1	0
18	1582	4	4	46	2	1	0
6	1050	4	1	35	2	1	1
18	2278	3	3	28	2	1	0
10	1418	3	2	35	1	1	0

WSzystko jest wczytane poprawnie.

### Stworzenie modelu

Naszym modelem będzie model z pakietu scidb. Nie musimy go tworzyć, od razu można “fit’ować” dane do modelu

```
glm.fit <- glm(is_good_customer_type ~ duration + age + existing_credits + dependents + has_telephone +
              ,data = Reg_train,family = binomial)
```

## Parametry modelu

Poniżej znajduje się podsumowanie naszego modelu. Pokazane są wszystkie jego parametry oraz znaczenie w działaniu naszego modelu.

```
summary(glm.fit)
```

```
##
## Call:
## glm(formula = is_good_customer_type ~ duration + age + existing_credits +
##      dependents + has_telephone + is_foreign_worker + has_problems_credit_history +
##      purpose_domestic + purpose_retraining + purpose_radio_television +
##      purpose_new_car + purpose_used_car + purpose_business + purpose_repairs +
##      purpose_education + purpose_furniture_equipment + other_debtors_guarantor +
##      other_debtors_co_applicant + other_installment_plans_bank +
##      other_installment_plans_stores + housing_rent + housing_own +
##      job_skilled_employee + job_unskilled_resident + job_highly_qualified_employee +
##      savings + present_employment + property + checking_account_status +
##      is_woman + is_single, family = binomial, data = Reg_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4722  -0.8899   0.4654   0.7781   2.3977
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.384690    1.241297   2.727  0.00640 **
## duration         -0.042163    0.007819  -5.392 6.97e-08 ***
## age              0.012578    0.009399   1.338  0.18084
## existing_credits -0.394848    0.188578  -2.094  0.03628 *
## dependents       -0.486423    0.265525  -1.832  0.06696 .
## has_telephone     0.456194    0.207464   2.199  0.02788 *
## is_foreign_worker -1.482214    0.768042  -1.930  0.05362 .
## has_problems_credit_history 0.782436    0.226453   3.455  0.00055 ***
## purpose_domestic  0.302664    0.334697   0.904  0.36584
## purpose_retraining -0.772425    0.448215  -1.723  0.08483 .
## purpose_radio_television -0.042468    0.352021  -0.121  0.90398
## purpose_new_car   -0.671123    0.335964  -1.998  0.04576 *
## purpose_used_car   1.109216    0.451582   2.456  0.01404 *
## purpose_business   1.009181    1.142224   0.884  0.37695
## purpose_repairs    -0.334217    0.843957  -0.396  0.69210
## purpose_education -0.543683    0.601829  -0.903  0.36632
## purpose_furniture_equipment 0.439982    0.828232   0.531  0.59526
## other_debtors_guarantor 0.634619    0.451470   1.406  0.15982
## other_debtors_co_applicant -0.811476    0.444269  -1.827  0.06777 .
## other_installment_plans_bank -0.367892    0.248429  -1.481  0.13864
## other_installment_plans_stores -0.702364    0.387586  -1.812  0.06996 .
## housing_rent      -0.143903    0.389959  -0.369  0.71211
## housing_own        0.270075    0.336478   0.803  0.42218
```

```
## job_skilled_employee      -0.647848    0.638322   -1.015    0.31014
## job_unskilled_resident    -0.674382    0.655113   -1.029    0.30329
## job_highly_qualified_employee -1.184181    0.672561   -1.761    0.07829 .
## savings                   0.167907    0.098253    1.709    0.08746 .
## present_employment        0.084757    0.040405    2.098    0.03593 *
## property                  0.174283    0.105706    1.649    0.09920 .
## checking_account_status   -0.429361    0.093033   -4.615    3.93e-06 ***
## is_woman                  0.217902    0.276982    0.787    0.43146
## is_single                 0.802860    0.276274    2.906    0.00366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 975.68  on 799  degrees of freedom
## Residual deviance: 795.81  on 768  degrees of freedom
## AIC: 859.81
##
## Number of Fisher Scoring iterations: 5
```

## Test modelu

Podstawowym parametrem jest stosunek poprawnych odpowiedzi. Tzn jest to prosta średnia z 1 jeśli odpowiedź jest dobra i 0 w przeciwnym przypadku.

```
glm.probs <- ifelse(predict(glm.fit,newdata = Reg_test,type = "response") > 0.5,1,0)
mean(glm.probs == select(Reg_test,is_good_customer_type))
```

```
## [1] 0.7
```

Jak widzimy model średnio dobrze przewiduje 70 % odpowiedzi.

## Dokładne zmierzenie jakości modelu

Funkcje pomocnicze które określą nam jakość modelu

```
confusion_matrix_values <- function(confusion_matrix){
  TP <- confusion_matrix[2,2]
  TN <- confusion_matrix[1,1]
  FP <- confusion_matrix[1,2]
  FN <- confusion_matrix[2,1]
  return (c(TP, TN, FP, FN))
}

accuracy <- function(confusion_matrix){
  conf_matrix <- confusion_matrix_values(confusion_matrix)
  return((conf_matrix[1] + conf_matrix[2]) / (conf_matrix[1] + conf_matrix[2] + conf_matrix[3] + conf_m
})

precision <- function(confusion_matrix){
  conf_matrix <- confusion_matrix_values(confusion_matrix)
```

```

    return(conf_matrix[1] / (conf_matrix[1] + conf_matrix[3]))
  }

recall <- function(confusion_matrix){
  conf_matrix <- confusion_matrix_values(confusion_matrix)
  return(conf_matrix[1] / (conf_matrix[1] + conf_matrix[4]))
}

f1 <- function(confusion_matrix){
  conf_matrix <- confusion_matrix_values(confusion_matrix)
  rec <- recall(confusion_matrix)
  prec <- precision(confusion_matrix)
  return(2 * (rec * prec) / (rec + prec))
}

```

```

confusion_matrix_primitive <- table(
  Truth = select(Reg_test,is_good_customer_type)[,1],
  Prediction = glm.probs
)
knitr::kable(confusion_matrix_primitive)

```

	0	1
0	16	45
1	15	124

```

accuracy_primitive <- accuracy(confusion_matrix_primitive)
precision_primitive <- precision(confusion_matrix_primitive)
recall_primitive <- recall(confusion_matrix_primitive)
f1_primitive <- f1(confusion_matrix_primitive)

classification_report_primitive <- data.frame(accuracy_primitive, precision_primitive,
                                              recall_primitive, f1_primitive)
colnames(classification_report_primitive) <- c("accuracy", "precision",
                                              "recall", "f1")
knitr::kable(classification_report_primitive)

```

accuracy	precision	recall	f1
0.7	0.7337278	0.8920863	0.8051948

## Wyrzucenie Mało Znaczących Zmiennych

Do zwiększenia dokładności modelu spróbujemy usunąć ze zmiennych te które, według funkcji summary(), najmniej wpływają na nasz model.

```

glm.fit <- glm(is_good_customer_type ~age + dependents + is_foreign_worker + purpose_domestic + purpose_foreign,
              ,data = Reg_train,family = binomial)
glm.probs <- ifelse(predict(glm.fit,newdata = Reg_test,type = "response") > 0.5,1,0)

```

```

confusion_matrix_primitive <- table(
  Truth = select(Reg_test,is_good_customer_type)[,1],
  Prediction = glm.probs
)
knitr::kable(confusion_matrix_primitive)

```

	0	1
0	9	52
1	15	124

```

accuracy_primitive <- accuracy(confusion_matrix_primitive)
precision_primitive <- precision(confusion_matrix_primitive)
recall_primitive <- recall(confusion_matrix_primitive)
f1_primitive <- f1(confusion_matrix_primitive)

classification_report_primitive <- data.frame(accuracy_primitive, precision_primitive,
                                              recall_primitive, f1_primitive)
colnames(classification_report_primitive) <- c("accuracy", "precision",
                                              "recall", "f1")
knitr::kable(classification_report_primitive)

```

accuracy	precision	recall	f1
0.665	0.7045455	0.8920863	0.7873016

Jak widać nie uzyskujemy lepszych rezultatów, a nasze wyniki są nawet lekko gorsze. Spróbujemy usunąć jeszcze kilka najmniej istotnych parametrów, na podstawie wskazań funkcji summary()

```
summary(glm.fit)
```

```

##
## Call:
## glm(formula = is_good_customer_type ~ age + dependents + is_foreign_worker +
##   purpose_domestic + purpose_retraining + purpose_radio_television +
##   purpose_business + purpose_repairs + purpose_education +
##   purpose_furniture_equipment + other_debtors_guarantor + other_debtors_co_applicant +
##   other_installment_plans_bank + other_installment_plans_stores +
##   housing_rent + housing_own + job_skilled_employee + job_unskilled_resident +
##   job_highly_qualified_employee + savings + present_employment +
##   property + checking_account_status + is_woman + is_single,
##   family = binomial, data = Reg_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3490  -1.1194   0.5603   0.8502   1.8128
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   1.868579    1.122322   1.665    0.0959 .

```

```
## age            0.020000  0.008831  2.265  0.0235 *
## dependents    -0.451985  0.251674 -1.796  0.0725 .
## is_foreign_worker -1.635407  0.765932 -2.135  0.0327 *
## purpose_domestic 0.468442  0.218721  2.142  0.0322 *
## purpose_retraining -0.370521  0.351804 -1.053  0.2922
## purpose_radio_television 0.211954  0.237065  0.894  0.3713
## purpose_business 1.253931  1.090908  1.149  0.2504
## purpose_repairs  0.150804  0.763680  0.197  0.8435
## purpose_education -0.448609  0.530250 -0.846  0.3975
## purpose_furniture_equipment 0.522847  0.730910  0.715  0.4744
## other_debtors_guarantor 0.404226  0.432311  0.935  0.3498
## other_debtors_co_applicant -0.910258  0.425450 -2.140  0.0324 *
## other_installment_plans_bank -0.431306  0.235634 -1.830  0.0672 .
## other_installment_plans_stores -0.740091  0.379054 -1.952  0.0509 .
## housing_rent      0.021814  0.363947  0.060  0.9522
## housing_own        0.342489  0.310810  1.102  0.2705
## job_skilled_employee -0.537390  0.604666 -0.889  0.3741
## job_unskilled_resident -0.600522  0.622727 -0.964  0.3349
## job_highly_qualified_employee -0.742295  0.625014 -1.188  0.2350
## savings            0.177937  0.092682  1.920  0.0549 .
## present_employment 0.085908  0.038128  2.253  0.0242 *
## property           0.233094  0.098987  2.355  0.0185 *
## checking_account_status -0.468764  0.087320 -5.368 7.95e-08 ***
## is_woman          0.106457  0.265636  0.401  0.6886
## is_single          0.660575  0.264440  2.498  0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 975.68  on 799  degrees of freedom
## Residual deviance: 862.00  on 774  degrees of freedom
## AIC: 914
##
## Number of Fisher Scoring iterations: 5
```

Te zmienne to:

- age
- is\_foreign\_worker
- present\_employment
- property
- checking\_account\_status
- is\_single

```
glm.fit <- glm(is_good_customer_type ~ dependents + purpose_domestic + purpose_retraining + purpose_rad
, data = Reg_train, family = binomial)
glm.probs <- ifelse(predict(glm.fit, newdata = Reg_test, type = "response") > 0.5, 1, 0)
```

```

confusion_matrix_primitive <- table(
  Truth = select(Reg_test,is_good_customer_type)[,1],
  Prediction = glm.probs
)
knitr::kable(confusion_matrix_primitive)

```

	0	1
0	4	57
1	5	134

```

accuracy_primitive <- accuracy(confusion_matrix_primitive)
precision_primitive <- precision(confusion_matrix_primitive)
recall_primitive <- recall(confusion_matrix_primitive)
f1_primitive <- f1(confusion_matrix_primitive)

classification_report_primitive <- data.frame(accuracy_primitive, precision_primitive,
                                              recall_primitive, f1_primitive)
colnames(classification_report_primitive) <- c("accuracy", "precision",
                                              "recall", "f1")
knitr::kable(classification_report_primitive)

```

accuracy	precision	recall	f1
0.69	0.7015707	0.9640288	0.8121212

Jak widać otrzymujemy lepsze wyniki niż poprzednio. Ostatnią metodą niech będzie stworzenie modelu z zmiennych mających największe znaczenie w naszym pierwszym modelu. Pięć zmiennych z największym znaczeniem to :

- purpose\_radio\_television : **0.90**
- housing\_rent : **0.71**
- purpose\_repairs : **0.69**
- purpose\_furniture\_equipment : **0.59**
- housing\_own : **0.42**

Stwórzmy teraz model na podstawie

```

glm.fit <- glm(is_good_customer_type ~ purpose_radio_television + purpose_repairs + purpose_furniture_e
              ,data = Reg_train,family = binomial)
glm.probs <- ifelse(predict(glm.fit,newdata = Reg_test,type = "response") > 0.5,1,0)

```

```

confusion_matrix_primitive <- table(
  Truth = select(Reg_test,is_good_customer_type)[,1],
  Prediction = glm.probs
)
knitr::kable(confusion_matrix_primitive)

```

	1
0	61
1	139

Co ciekawe ten model nie przewidział żadnego 0 czyli złego klienta. Ten model uzyskuje celność **0.65** co jest podobnym wynikiem do reszty. Jednak z powodu nie przewidzenia złych klientów nie można obliczyć reszty statystyk.

## Podumowanie

Model uzyskuje podobne parapetry dla różnych zmiennych. Najgorzej wypadł model bez pięciu najmniej istotnych zmiennych. Może to być jednak spowodowane ilością danych jak i ich arbitralnym podziałem na zbiór testowy i treningowy.