

Sprawozdanie 2

Krzysztof Kotlarz

24.03.2020

Lista 3

Zadanie 5

(i)

```
## Call:
## lda(owner ~ income + lotsize, data = dane)
##
## Prior probabilities of groups:
##   1   2
## 0.5 0.5
##
## Group means:
##   income  lotsize
## 1 26.49167 10.13333
## 2 19.13333  8.81667
##
## Coefficients of linear discriminants:
##               LD1
## income  -0.1453404
## lotsize -0.7590457
```

Średnie w poszczególnych grupach kształtują się następująco:

- dla grupy 1 $income = 26.49$, $lotsize = 10.13$
- dla grupy 2 $income = 19.13$, $lotsize = 8.82$

(ii)

Wartości współczynników klasyfikacyjnych są następujące:

- $LD1 = -0.1453 * income - 0.759 * lotsize$

(iii)

Im większy współczynnik zmiennej tym jest on istotniejszy: większy współczynnik znajduje się przy zmiennej *income*

(iV)

```
LD1 <- predict(model.1)$x  
sum(LD1 * (dane$owner == 1)) / sum(dane$owner == 1)
```

```
## [1] -1.034437
```

```
sum(LD1 * (dane$owner == 2)) / sum(dane$owner == 2)
```

```
## [1] 1.034437
```

Średnie wartości funkcji dyskryminacyjnych dla:

- *owner = 1*: -1.034
- *owner = 2*: 1.034

(v)

```
owner.predict <- predict(model.1, dane[,1:2])
```

Za zbiór testowy przyjmujemy cały zbiór danych

(vi)

```
owner.predict$posterior[20,]
```

```
##           1           2  
## 0.01612995 0.98387005
```

Prawdopodobieństwo przynależności obserwacji 20 dla poszczególnych grup:

- grupa 1: 0.016
- grupa 2: 0.984

(vii)

Ta obserwacja zostanie zaklasyfikowana do grupy 2 z prawdopodobieństwem 98.4%

(xiii)

```
owner.classes <- owner.predict$class  
sum(owner.classes==owner)/length(owner)
```

```
## [1] 0.875
```

Poprawna klasyfikacji 87.5% obserwacji

(ix)

```
##          Predicted
## Original  1  2
##          1 11  1
##          2  2 10
```

Coefficient matrix przedstawia w wierszach przynależność obserwacji do klas ze zbioru, natomiast kolumny przynależność obserwacji do klas przewidziane przez model:

- 11 obserwacji będących *owner* = 1, zostało poprawnie przewidziane przez model, natomiast 1 obserwacja błędnie
- 10 obserwacji będących *owner* = 2, zostało poprawnie przewidziane przez model, natomiast 2 obserwacje błędnie

Zadanie 6.

(i)

```
## Call:
## lda(t_uprawy ~ ., data = dane2)
##
## Prior probabilities of groups:
##          1          2          3
## 0.3314607 0.3988764 0.2696629
##
## Group means:
##          V2          V3          V4          V5          V6          V7          V8          V9
## 1 13.74475  2.010678  2.455593 17.03729 106.3390  2.840169  2.9823729  0.290000
## 2 12.27873  1.932676  2.244789 20.23803  94.5493  2.258873  2.0808451  0.363662
## 3 13.15375  3.333750  2.437083 21.41667  99.3125  1.678750  0.7814583  0.447500
##          V10          V11          V12          V13          V14
## 1 1.899322  5.528305  1.0620339  3.157797 1115.7119
## 2 1.630282  3.086620  1.0562817  2.785352  519.5070
## 3 1.153542  7.396250  0.6827083  1.683542  629.8958
##
## Coefficients of linear discriminants:
##          LD1          LD2
## V2 -0.403399781  0.8717930699
## V3  0.165254596  0.3053797325
## V4 -0.369075256  2.3458497486
## V5  0.154797889 -0.1463807654
## V6 -0.002163496 -0.0004627565
## V7  0.618052068 -0.0322128171
## V8 -1.661191235 -0.4919980543
## V9 -1.495818440 -1.6309537953
## V10 0.134092628 -0.3070875776
## V11 0.355055710  0.2532306865
## V12 -0.818036073 -1.5156344987
## V13 -1.157559376  0.0511839665
## V14 -0.002691206  0.0028529846
```

```
##
## Proportion of trace:
##      LD1      LD2
## 0.6875 0.3125
```

(i)

Średnie w grupach prezentują się następująco:

```
knitr::kable(model.2$means, caption = 'Średnie wartości w grupach', digits = 3, row.names = c(1,2,3,4))
```

```
## Warning in if (is.na(row.names)) row.names = has_rownames(x): warunek posiada
## długość > 1 i tylko pierwszy element będzie użyty
```

```
## Warning in if (row.names) {: warunek posiada długość > 1 i tylko pierwszy
## element będzie użyty
```

Table 1: Średnie wartości w grupach

	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
1	13.745	2.011	2.456	17.037	106.339	2.840	2.982	0.290	1.899	5.528	1.062	3.158	1115.712
2	12.279	1.933	2.245	20.238	94.549	2.259	2.081	0.364	1.630	3.087	1.056	2.785	519.507
3	13.154	3.334	2.437	21.417	99.312	1.679	0.781	0.448	1.154	7.396	0.683	1.684	629.896

(ii)

Współczynniki dla poszczególnych zmiennych są następujące:

```
knitr::kable(model.2$scaling, digits = 3)
```

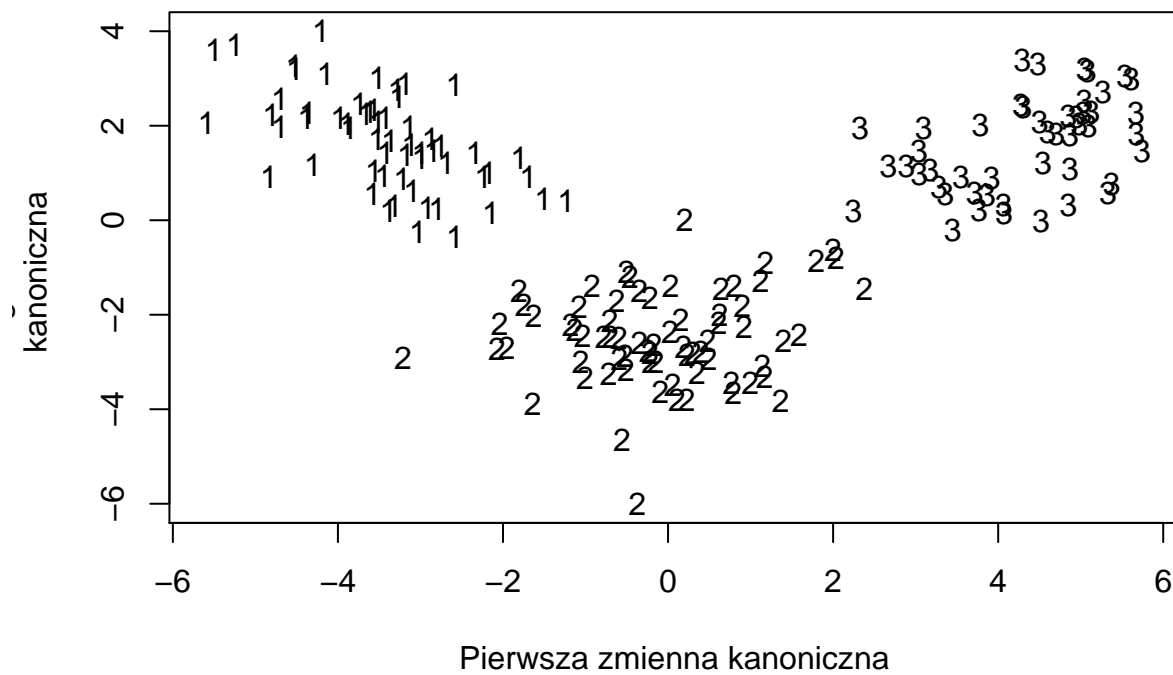
	LD1	LD2
V2	-0.403	0.872
V3	0.165	0.305
V4	-0.369	2.346
V5	0.155	-0.146
V6	-0.002	0.000
V7	0.618	-0.032
V8	-1.661	-0.492
V9	-1.496	-1.631
V10	0.134	-0.307
V11	0.355	0.253
V12	-0.818	-1.516
V13	-1.158	0.051
V14	-0.003	0.003

(iii)

Im większy współczynnik zmiennej tym jest on istotniejszy:

- dla LD1 najbardziej istotna zmienna: V_7 (0.618)
- dla LD2 najbardziej istotna zmienna: V_4 (2.346)

(iv)



```
sum(LD1 * (dane2$t_uprawy == 1)) / sum(dane2$t_uprawy == 1)
```

```
## [1] -3.422489
```

```
sum(LD2 * (dane2$t_uprawy == 1)) / sum(dane2$t_uprawy == 1)
```

```
## [1] 1.691674
```

```
sum(LD1 * (dane2$t_uprawy == 2)) / sum(dane2$t_uprawy == 2)
```

```
## [1] -0.07972623
```

```
sum(LD2 * (dane2$t_uprawy == 2)) / sum(dane2$t_uprawy == 2)
```

```
## [1] -2.472656
```

```
sum(LD1 * (dane2$t_uprawy == 3)) / sum(dane2$t_uprawy == 3)
```

```
## [1] 4.324737
```

```
sum(LD2 * (dane2$t_uprawy == 3)) / sum(dane2$t_uprawy == 3)
```

```
## [1] 1.57812
```

Średnie wartości funkcji dyskryminacyjnych dla:

- $t_uprawy = 1$:
 - $LD1$: -3.422
 - $LD2$: 1.691
- $t_uprawy = 2$:
 - $LD1$: -0.079
 - $LD2$: -2.472
- $t_uprawy = 3$:
 - $LD1$: 4.324
 - $LD2$: 1.578

(v)

```
type_u.predict <- predict(model.2, dane2[,2:14])
```

Za zbiór testowy przyjmujemy cały zbiór danych

(vi)

```
type_u.predict$posterior[20,]
```

```
##           1           2           3
## 9.999994e-01 6.131886e-07 6.646280e-13
```

Prawdopodobieństwo przynależności obserwacji 20 dla poszczególnych grup:

- grupa 1: 0.99
- grupa 2: 6.13e-07
- grupa 3 : 6.64e-13

(vii)

Obserwacja zostanie zaklasyfikowana do grupy 1 z prawdopodobieństwem 99%

(viii)

```
type_u.classes <- type_u.predict$class
type_u.classperc <- sum(type_u.classes==dane2$t_uprawy)/length(dane2$t_uprawy)
type_u.classperc
```

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

Poprawna klasyfikacji 100% obserwacji

(ix)

```
table(Original = dane2$t_uprawy, Predicted = predict(model.2)$class)
```

```
##          Predicted
## Original  1  2  3
##          1 59  0  0
##          2  0 71  0
##          3  0  0 48
```

Coefficient matrix przedstawia w wierszach przynależność obserwacji do klas ze zbioru, natomiast kolumny przynależność obserwacji do klas przewidziane przez model:

- Wszystkie obserwacje zostały poprawnie zaklasyfikowane do odpowiednich grup

Lista 4

Zadanie 7

```
dane <- read.csv('http://theta.edu.pl/wp-content/uploads/2018/03/urine.csv', skip = 1, dec= '.', sep =
dane <- dane[,2:8]
```

```
dane$r <- as.factor(dane$r)
dane <- na.omit(dane)
```

(i)

```
set.seed(123)
```

```
train_ind <- sample(seq_len(nrow(dane)), size = 50)
```

```
train <- dane[train_ind, ]
test <- dane[-train_ind, ]
```

(ii)

```
model.glm <- glm(r ~ . , data = train, family = binomial)
summary(model.glm)

##
## Call:
## glm(formula = r ~ . , family = binomial, data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2673  -0.7415  -0.2858   0.3622   2.4701
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -133.73202   237.76262  -0.562   0.5738
## gravity      135.94859   237.52922   0.572   0.5671
## ph           -0.58564    0.61048  -0.959   0.3374
## osmo          0.02444    0.01966   1.243   0.2139
## cond         -0.45606    0.26966  -1.691   0.0908 .
## urea         -0.03694    0.01843  -2.004   0.0450 *
## calc          0.71300    0.25894   2.754   0.0059 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 40.756  on 43  degrees of freedom
## AIC: 54.756
##
## Number of Fisher Scoring iterations: 6
```

(iii)

```
model.glm.1 <- glm(r ~ . - gravity, data = train, family = binomial)
summary(model.glm.1)

##
## Call:
## glm(formula = r ~ . - gravity, family = binomial, data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2367  -0.7331  -0.2814   0.3726   2.4754
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.42196    3.89260   0.622  0.53381
## ph           -0.56723    0.58841  -0.964  0.33505
## osmo          0.03315    0.01389   2.387  0.01698 *
## cond         -0.55636    0.21721  -2.561  0.01043 *
## urea         -0.04175    0.01657  -2.520  0.01173 *
```



```
## calc          0.73879    0.25582    2.888  0.00388 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 41.116  on 44  degrees of freedom
## AIC: 53.116
##
## Number of Fisher Scoring iterations: 6
```

Usunięto nieistotną zmienną *gravity*

```
model.glm.2 <- glm(r~. -gravity -ph, data = train, family = binomial)
summary(model.glm.2)
```

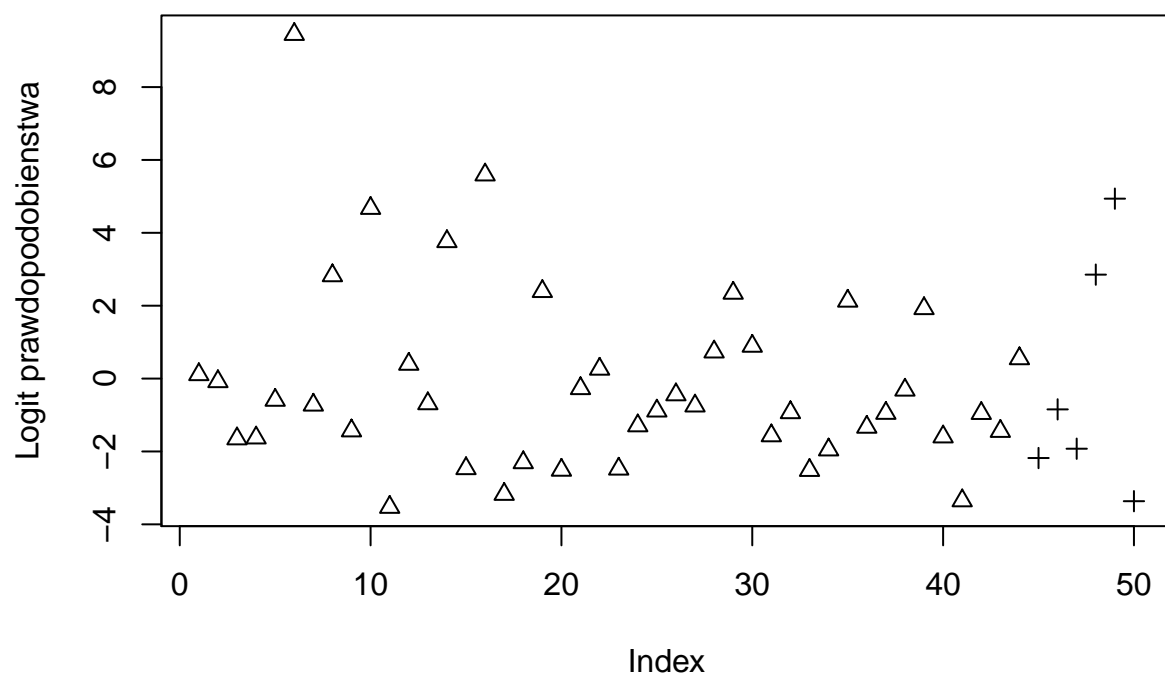
```
##
## Call:
## glm(formula = r ~ . - gravity - ph, family = binomial, data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3493  -0.6932  -0.3400   0.4255   2.2635
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.17265    1.15309  -1.017  0.30917
## osmo         0.03183    0.01282   2.482  0.01307 *
## cond        -0.53501    0.20109  -2.661  0.00780 **
## urea        -0.03918    0.01513  -2.590  0.00961 **
## calc         0.70867    0.24575   2.884  0.00393 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 42.076  on 45  degrees of freedom
## AIC: 52.076
##
## Number of Fisher Scoring iterations: 6
```

Usunięto nieistotną zmienną *ph*

(iV)

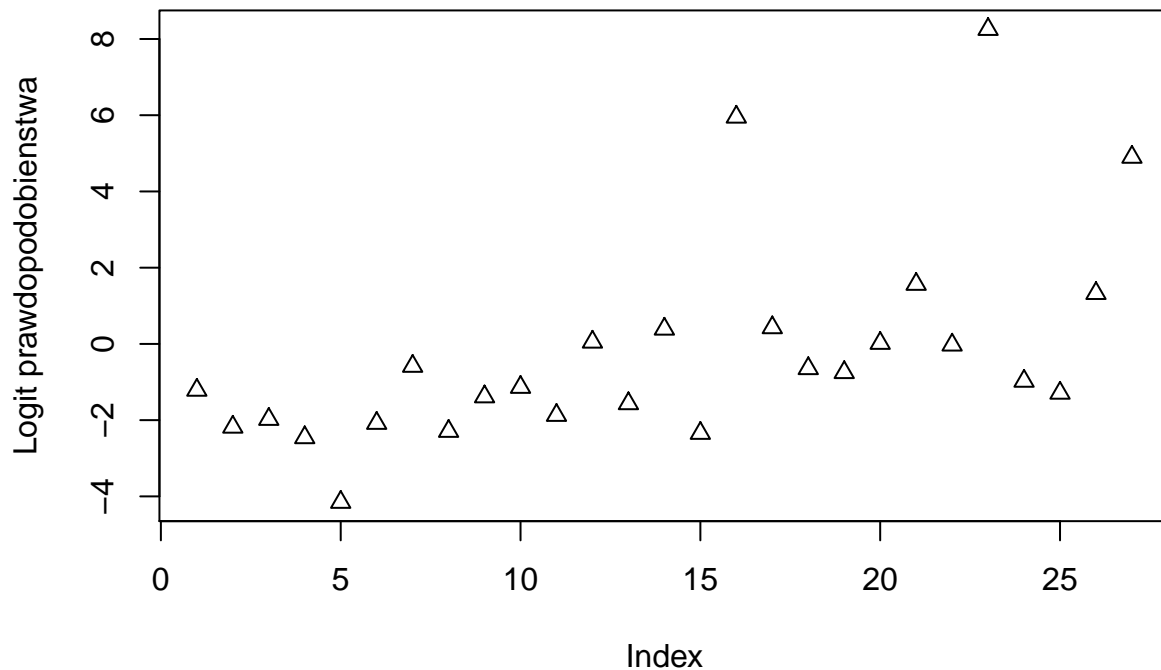
```
predictGLM <- predict(model.glm.2, newdata = train)
plot(predictGLM, pch = as.numeric(dane$r) + 1, ylab = 'Logit prawdopodobieństwa', main = 'Logity prawdo
```

Logity prawdopodobienstwa dla zbioru treningowego



```
predictLR <- predict(model.glm.2, newdata = test)
plot(predictLR, pch = as.numeric(dane$r) + 1, ylab = 'Logit prawdopodobieństwa', main = 'Logity prawdopodobieństwa')
```

Logity prawdopodobieństwa dla zbioru testowego



```
TAB <- table(test[,1], predictLR > 0)
knitr::kable(TAB)
```

	FALSE	TRUE
0	13	2
1	5	7

Coefficient matrix przedstawia w wierszach przynależność obserwacji do klas ze zbioru, natomiast kolumny przynależność obserwacji do klas przewidziane przez model:

- 13 obserwacji będących $r = 0$, zostało poprawnie przewidziane przez model, natomiast 2 obserwacje błędnie
- 7 obserwacji będących $r = 1$, zostało poprawnie przewidziane przez model, natomiast 5 obserwacje błędnie

```
mcrlr <- 1 - sum(diag(TAB)) / sum(TAB)
mcrlr
```

```
## [1] 0.2592593
```

Błąd klasyfikacji wynosi 25.9%

(v)

```
model.lda <- lda(r~. -gravity -ph, data = train)
model.lda

## Call:
## lda(r ~ . - gravity - ph, data = train)
##
## Prior probabilities of groups:
##      0      1
## 0.58 0.42
##
## Group means:
##      osmo      cond      urea      calc
## 0 568.8966 20.85862 236.7586 2.862414
## 1 637.2381 21.07143 264.2857 6.002857
##
## Coefficients of linear discriminants:
##              LD1
## osmo  0.01625975
## cond -0.27558553
## urea -0.01979592
## calc  0.33438051

lda.predict <- predict(model.lda, test[,2:7])
pid.classperc <- sum(lda.predict$class == test[,1]) / nrow(test)
pid.classperc

## [1] 0.7407407

1 - pid.classperc

## [1] 0.2592593
```

Błąd klasyfikacji w przypadku regresji logistycznej jak i metody LDA jest taki sam (25.9%).

Zadanie 8

```
dane3 <- read.table('http://theta.edu.pl/wp-content/uploads/2018/03/puls2.csv', header = TRUE, sep = ';')
attach(dane3)

dane3$Palacz <- as.factor(dane3$Palacz)

(i)

set.seed(333)

smp_size <- floor(0.75 * nrow(dane3))

train_ind <- sample(seq_len(nrow(dane3)), size = smp_size)

train <- dane3[train_ind, ]
test <- dane3[-train_ind, ]
```

(ii)

```
model.4 <- glm(TetnoSpocz~Palacz+Waga, data = train, family = binomial)
summary(model.4)
```

```
##
## Call:
## glm(formula = TetnoSpocz ~ Palacz + Waga, family = binomial,
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.0962  -0.8943  -0.4975   0.8938   2.7195
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.62354    2.25967  -3.374 0.000742 ***
## Palacz1     -0.06030    0.62090  -0.097 0.922637
## Waga         0.08231    0.02564   3.210 0.001325 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 90.354  on 68  degrees of freedom
## Residual deviance: 74.768  on 66  degrees of freedom
## AIC: 80.768
##
## Number of Fisher Scoring iterations: 4
```

(iii)

```
model.5 <- glm(TetnoSpocz~Waga, data = train, family = binomial)
summary(model.5)
```

```
##
## Call:
## glm(formula = TetnoSpocz ~ Waga, family = binomial, data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1138  -0.8880  -0.4937   0.8784   2.7250
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.63616    2.25074  -3.393 0.000692 ***
## Waga         0.08225    0.02556   3.218 0.001292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 90.354  on 68  degrees of freedom
```

```
## Residual deviance: 74.778 on 67 degrees of freedom
## AIC: 78.778
##
## Number of Fisher Scoring iterations: 4
```

Usunięto nieistotną zmienną *Palacz*. Ostatecznie w modelu pozostaje zmienna *Waga*

(iv)

```
model.m1 <- glm(TetnoSpocz~Palacz*Waga, data = train, family = binomial)
summary(model.m1)
```

```
##
## Call:
## glm(formula = TetnoSpocz ~ Palacz * Waga, family = binomial,
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1661  -0.9259  -0.4735   0.8040   2.3105
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.22692    2.24231  -2.331  0.0198 *
## Palacz1      -22.16414   15.06934  -1.471  0.1413
## Waga          0.05478    0.02554   2.145  0.0319 *
## Palacz1:Waga  0.24592    0.16535   1.487  0.1369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 90.354 on 68 degrees of freedom
## Residual deviance: 69.252 on 65 degrees of freedom
## AIC: 77.252
##
## Number of Fisher Scoring iterations: 7
```

Nieistotna zmienna *Palacz* oraz interakcja pomiędzy zmiennymi *Palacz* i *Waga*

(v)

```
model.m2 <- glm(TetnoSpocz~Waga + Palacz:Waga, data = train, family = binomial)
summary(model.m2)
```

```
##
## Call:
## glm(formula = TetnoSpocz ~ Waga + Palacz:Waga, family = binomial,
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```

## -1.1416 -0.8788 -0.5058 0.8901 2.7291
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.629116  2.242677 -3.402 0.000669 ***
## Waga         0.081867  0.025582  3.200 0.001373 **
## Waga:Palacz1 0.001037  0.007014  0.148 0.882470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 90.354  on 68  degrees of freedom
## Residual deviance: 74.756  on 66  degrees of freedom
## AIC: 80.756
##
## Number of Fisher Scoring iterations: 4

model.m3 <- glm(TetnoSpocz~ Palacz + Palacz:Waga, data = train, family = binomial)
summary(model.m3)

##
## Call:
## glm(formula = TetnoSpocz ~ Palacz + Palacz:Waga, family = binomial,
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1661  -0.9259  -0.4735   0.8040   2.3105
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.22692  2.24231  -2.331  0.0198 *
## Palacz1     -22.16414  15.06934  -1.471  0.1413
## Palacz0:Waga  0.05478  0.02554  2.145  0.0319 *
## Palacz1:Waga  0.30071  0.16337  1.841  0.0657 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 90.354  on 68  degrees of freedom
## Residual deviance: 69.252  on 65  degrees of freedom
## AIC: 77.252
##
## Number of Fisher Scoring iterations: 7

model.m4 <- glm(TetnoSpocz~Palacz:Waga, data = train, family = binomial)
summary(model.m4)

##
## Call:
## glm(formula = TetnoSpocz ~ Palacz:Waga, family = binomial, data = train)

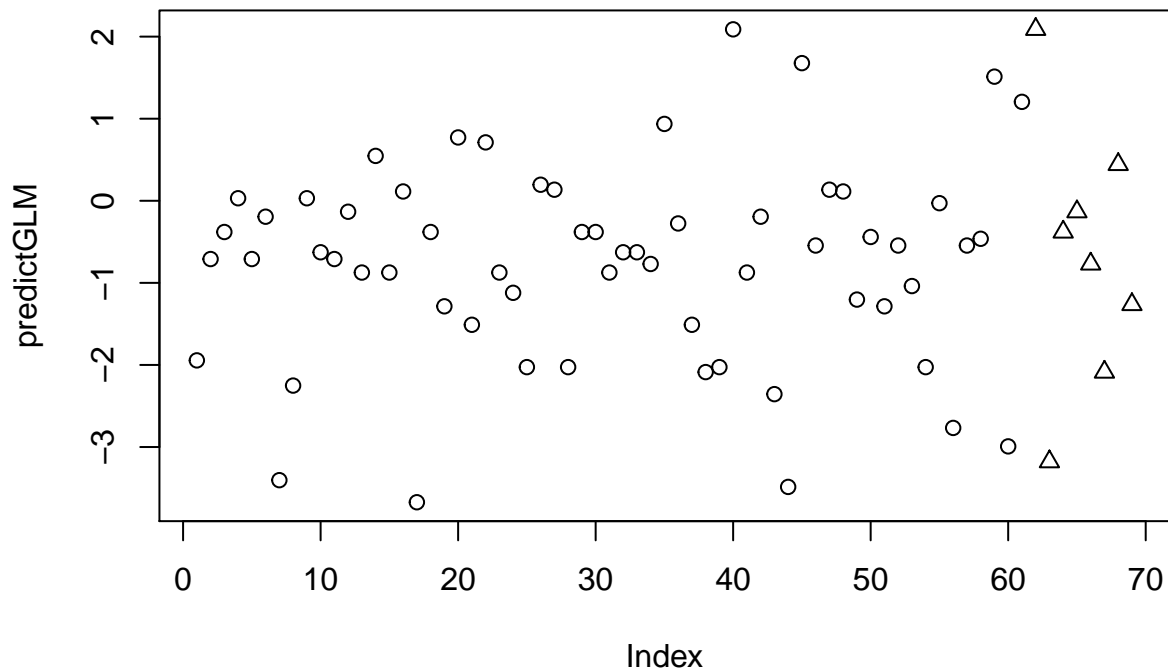
```

```
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1416  -0.8788  -0.5058   0.8901   2.7291
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.62912     2.24268  -3.402 0.000669 ***
## Palacz0:Waga  0.08187     0.02558   3.200 0.001373 **
## Palacz1:Waga  0.08290     0.02588   3.204 0.001358 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 90.354  on 68  degrees of freedom
## Residual deviance: 74.756  on 66  degrees of freedom
## AIC: 80.756
##
## Number of Fisher Scoring iterations: 4
```

Model **M4** zawiera zmienne istotne: jest to model uwzględniający tylko interakcje pomiędzy zmiennymi *Palacz* i *Waga*

Dla pełnego modelu

```
predictGLM <- predict(model.4, newdata = train)
plot(predictGLM, pch = as.numeric(dane3$TetnoSpocz))
```

```
predictLR <- predict(model.4, newdata = test)
TAB <- table(test[,1], predictLR > 0)
knitr::kable(TAB)
```

	FALSE	TRUE
Niskie	17	0
Wysokie	4	2

- 17 obserwacji będących *TetnoSpocz* = Niskie, zostało poprawnie przewidziane przez model, natomiast 0 obserwacja błędnie
- 2 obserwacji będących *TetnoSpocz* = Wysokie, zostało poprawnie przewidziane przez model, natomiast 4 obserwacje błędnie

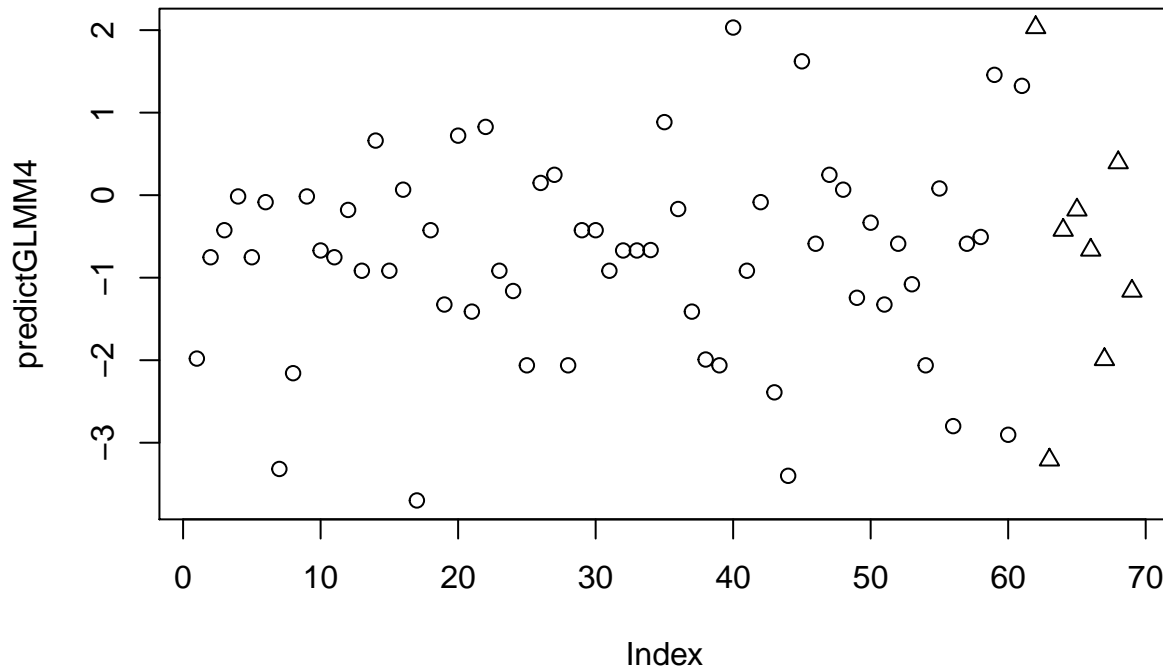
```
mcrlr <- 1-sum(diag(TAB))/sum(TAB)
mcrlr
```

```
## [1] 0.173913
```

Błąd klasyfikacji dla pełnego modelu wynosi 17.4%

Dla M4

```
predictGLMM4 <- predict(model.m4, newdata = train)
plot(predictGLMM4, pch = as.numeric(dane3$TetnoSpocz))
```



```
predictLRM4 <- predict(model.m4, newdata = test)
TABM4 <- table(test[,1], predictLRM4 > 0)
knitr::kable(TABM4)
```

	FALSE	TRUE
Niskie	17	0
Wysokie	4	2

- 17 obserwacji będących *TetnoSpocz* = Niskie, zostało poprawnie przewidziane przez model, natomiast 0 obserwacja błędnie
- 2 obserwacji będących *TetnoSpocz* = Wysokie, zostało poprawnie przewidziane przez model, natomiast 4 obserwacje błędnie

```
mcrlr <- 1-sum(diag(TABM4))/sum(TABM4)
mcrlr
```

```
## [1] 0.173913
```

Błąd klasyfikacji w przypadku regresji logistycznej wynosi dla obu modeli (pełny i M4) 17.4%

```

library(MASS)
model.lda <- lda(TetnoSpocz~Palacz + Waga, data = train)
model.lda

## Call:
## lda(TetnoSpocz ~ Palacz + Waga, data = train)
##
## Prior probabilities of groups:
##      Niskie      Wysokie
## 0.6376812 0.3623188
##
## Group means:
##           Palacz1      Waga
## Niskie  0.3181818 78.81818
## Wysokie 0.2800000 92.68000
##
## Coefficients of linear discriminants:
##              LD1
## Palacz1 0.03296072
## Waga    0.07304346

lda.predict <- predict(model.lda, test[,2:3])
pid.classperc <- sum(lda.predict$class == test[,1]) / nrow(test)
pid.classperc

## [1] 0.826087

1 - pid.classperc

## [1] 0.173913

```

W celu porównania obu klasyfikacji, w modelu LDA użyto tych samych zmiennych co w modelu regresji logistycznej. Błąd klasyfikacji w przypadku regresji logistycznej jak i metody LDA jest taki sam (17.4%).