Statystyka wielowymiarowa
Laboratorium nr 3

Kamil Szkoła
Patryk Krukowski
Data Science, sem. 1

8 czerwca 2022

Spis treści

1 Wstęp

Naszym celem jest selekcja cech metodami najlepszego podbioru oraz metod krokowych przy pomocy optymalnych statystyk C_p Mallowsa, BIC i skorygowane R^2 dla zbiorów danych *life_expectancy.csv* oraz *titanic.csv*. Do selekcji cech użyjemy także regularyzacji metodą lasso.

Na przykładzie zbioru *life_expectancy.csv* mamy do czynienia z zadaniem regresji, a zbioru *titanic.csv* używamy do zadania klasyfikacji.

2 Selekcja cech dla zadania regresji

Przejdziemy teraz do analizy właściwej.

2.1 Metoda najlepszego podzbioru

Używamy funkcji *regsubsets*.

```
> # Wybór najlepszego podzbioru
> life_exp_bs <- regsubsets(Life.expectancy ~ .,
+                           data = lf,
+                           nvmax = 19,
+                           really.big = TRUE)
> life_exp_bs_sum <- summary(life_exp_bs)
```

W celu znalezienia najlepszego podzbioru skorzystamy z optymalnych statystyk:

- C_p Mallowsa

```
> lf_cp_min <-  which.min(life_exp_bs_sum$cp)
> lf_cp_min

[1] 13

> # Wybrane predyktory
> lf_model_cp <- life_exp_bs_sum$which[lf_cp_min, -1]
> lf_predictors_cp <- names(which(lf_model_cp == TRUE))
> lf_predictors_cp

 [1] "StatusDeveloping"                "Adult.Mortality"
 [3] "infant.deaths"                   "Alcohol"
 [5] "percentage.expenditure"          "BMI"
 [7] "under.five.deaths"               "Total.expenditure"
 [9] "Diphtheria"                      "HIV.AIDS"
[11] "thinness.5.9.years"              "Income.composition.of.resources"
[13] "Schooling"
```

- BIC

```
> lf_bic_min <-  which.min(life_exp_bs_sum$bic)
> lf_bic_min

[1] 9

> lf_model_bic <- life_exp_bs_sum$which[lf_bic_min, -1]
> # Wybrane predyktory
> lf_predictors_bic <- names(which(lf_model_bic == TRUE))
> lf_predictors_bic

[1] "Adult.Mortality"                 "infant.deaths"
[3] "percentage.expenditure"          "BMI"
[5] "under.five.deaths"               "Diphtheria"
[7] "HIV.AIDS"                        "Income.composition.of.resources"
[9] "Schooling"
```

- Skorygowane R^2

```
> lf_r_squared_max <- which.max(life_exp_bs_sum$adjr2)
> lf_r_squared_max

[1] 15

> lf_model_r_squared <- life_exp_bs_sum$which[lf_r_squared_max, -1]
> # Wybrane predyktory
> lf_predictors_r_squared <- names(which(lf_model_r_squared == TRUE))
> lf_predictors_r_squared

 [1] "StatusDeveloping"                "Adult.Mortality"
 [3] "infant.deaths"                   "Alcohol"
 [5] "percentage.expenditure"          "Hepatitis.B"
 [7] "BMI"                             "under.five.deaths"
 [9] "Polio"                           "Total.expenditure"
[11] "Diphtheria"                      "HIV.AIDS"
[13] "thinness.5.9.years"              "Income.composition.of.resources"
[15] "Schooling"
```

2.2 Selekcja krokowa do przodu

Używamy funkcji *regsubsets*.

```
> # Wybór najlepszego podzbioru
> lf_fwd <- regsubsets(Life.expectancy ~ ., data = lf, method = "forward",
+                      nvmax = 19)
> lf_fwd_sum <- summary(lf_fwd)
```

W celu znalezienia najlepszego podzbioru skorzystamy z optymalnych statystyk:

- C_p Mallowsa

```
> lf_fwd_cp_min <-  which.min(lf_fwd_sum$cp)
> lf_fwd_model_cp <- lf_fwd_sum$which[lf_fwd_cp_min, -1]
> # Wybrane predyktory
> lf_fwd_predictors_cp <- names(which(lf_fwd_model_cp == TRUE))
> lf_fwd_predictors_cp

 [1] "StatusDeveloping"                "Adult.Mortality"
 [3] "infant.deaths"                   "Alcohol"
 [5] "percentage.expenditure"          "BMI"
 [7] "under.five.deaths"               "Total.expenditure"
 [9] "Diphtheria"                      "HIV.AIDS"
[11] "thinness.5.9.years"              "Income.composition.of.resources"
[13] "Schooling"
```

- BIC

```
> lf_fwd_bic_min <-  which.min(lf_fwd_sum$bic)
> lf_fwd_model_bic <- lf_fwd_sum$which[lf_fwd_bic_min, -1]
> # Wybrane predyktory
> lf_fwd_predictors_bic <- names(which(lf_fwd_model_bic == TRUE))
> lf_fwd_predictors_bic

[1] "Adult.Mortality"                 "infant.deaths"
[3] "percentage.expenditure"          "BMI"
[5] "under.five.deaths"               "Diphtheria"
[7] "HIV.AIDS"                        "Income.composition.of.resources"
[9] "Schooling"
```

- Skorygowane R^2

```
> lf_fwd_r_squared_max <-  which.max(lf_fwd_sum$adjr2)
> lf_fwd_model_r_squared <- lf_fwd_sum$which[lf_fwd_r_squared_max, -1]
> # Wybrane predyktory
> lf_fwd_predictors_r_squared <- names(which(lf_fwd_model_r_squared == TRUE))
> lf_fwd_predictors_r_squared

 [1] "StatusDeveloping"                "Adult.Mortality"
 [3] "infant.deaths"                   "Alcohol"
 [5] "percentage.expenditure"          "Hepatitis.B"
 [7] "BMI"                             "under.five.deaths"
 [9] "Polio"                           "Total.expenditure"
[11] "Diphtheria"                      "HIV.AIDS"
[13] "thinness.5.9.years"              "Income.composition.of.resources"
[15] "Schooling"
```

2.3 Selekcja krokowa wstecz

Używamy funkcji *regsubsets*.

```
> # Wybór najlepszego podzbioru
> lf_back <- regsubsets(Life.expectancy ~ ., data = lf, nvmax = 19,
+                       method = "backward")
> lf_back_sum <- summary(lf_back)
```

W celu znalezienia najlepszego podzbioru skorzystamy z optymalnych statystyk:

- C_p Mallowsa

```
> lf_back_cp_min <-  which.min(lf_back_sum$cp)
> lf_back_model_cp <- lf_back_sum$which[lf_back_cp_min, -1]
> # Wybrane predyktory
> lf_back_predictors_cp <- names(which(lf_back_model_cp == TRUE))
> lf_back_predictors_cp
```

```
 [1] "StatusDeveloping"                "Adult.Mortality"
 [3] "infant.deaths"                   "Alcohol"
 [5] "percentage.expenditure"          "BMI"
 [7] "under.five.deaths"               "Total.expenditure"
 [9] "Diphtheria"                      "HIV.AIDS"
[11] "thinness.5.9.years"              "Income.composition.of.resources"
[13] "Schooling"
```

- BIC

```
> lf_back_bic_min <-  which.min(lf_back_sum$bic)
> lf_back_model_bic <- lf_back_sum$which[lf_back_bic_min, -1]
> # Wybrane predyktory
> lf_back_predictors_bic <- names(which(lf_back_model_bic == TRUE))
> lf_back_predictors_bic

[1] "Adult.Mortality"                 "infant.deaths"
[3] "percentage.expenditure"          "BMI"
[5] "under.five.deaths"               "Diphtheria"
[7] "HIV.AIDS"                        "Income.composition.of.resources"
[9] "Schooling"
```

- Skorygowane R^2

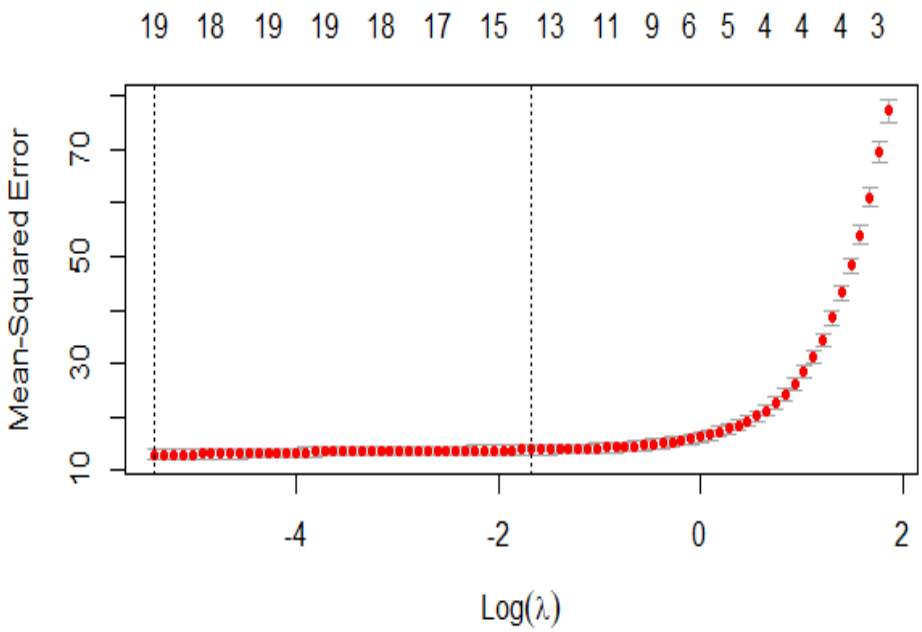
```
> lf_back_r_squared_max <-  which.max(lf_back_sum$adjr2)
> lf_back_model_r_squared <- lf_back_sum$which[lf_back_r_squared_max, -1]
> # Wybrane predyktory
> lf_back_predictors_r_squared <- names(which(lf_back_model_r_squared == TRUE))
> lf_back_predictors_r_squared

 [1] "StatusDeveloping"                "Adult.Mortality"
 [3] "infant.deaths"                   "Alcohol"
 [5] "percentage.expenditure"          "Hepatitis.B"
 [7] "BMI"                             "under.five.deaths"
 [9] "Polio"                           "Total.expenditure"
[11] "Diphtheria"                      "HIV.AIDS"
[13] "thinness.5.9.years"              "Income.composition.of.resources"
[15] "Schooling"
```

2.4 Selekcja cech metodą regularyzacji lasso

Przygotowujemy dane i dopasowujemy model lasso. W tym celu ustawiamy parametr $\alpha = 1$.

```
> lf_data <- model.matrix(Life.expectancy ~ ., data = lf)[, -1]
> lf_target <- lf$Life.expectancy
> lf_fit_lasso <- glmnet(lf_data, lf_target, alpha = 1)
```



Rysunek 1: Zależność MSE od $\log(\lambda)$

Narysujemy teraz wykres zależności MSE od $\log(\lambda)$.

Optymalna wartość λ to według nas e^{-2}. Wybieramy współczynniki przy pomocy funkcji *predict*.

```
> lf_pred_lasso <- predict(lf_fit_lasso, s = exp(-2),
+                          type = 'coefficients')
> lf_pred_lasso

20 x 1 sparse Matrix of class "dgCMatrix"
                                        s1
(Intercept)                   5.356224e+01
StatusDeveloping             -2.518589e-01
Adult.Mortality              -1.810627e-02
infant.deaths                 .
Alcohol                      -7.716370e-03
percentage.expenditure        3.017328e-04
Hepatitis.B                   .
Measles                       .
BMI                           3.336227e-02
```

```
under.five.deaths               -1.203872e-03
Polio                            6.023215e-03
Total.expenditure                1.621677e-02
Diphtheria                       1.406067e-02
HIV.AIDS                        -4.273719e-01
GDP                              9.506260e-06
Population                       .
thinness..1.19.years            -1.571327e-03
thinness.5.9.years              -2.132202e-02
Income.composition.of.resources  9.936449e+00
Schooling                        8.625989e-01
```

Wartości współcynników dla zmiennych, przy których widzimy symbol ".", są dokładnie równe 0.

3 Selekcja cech dla zadania klasyfikacji

Przejdziemy teraz do analizy właściwej.

3.1 Metoda najlepszego podzbioru

Używamy funkcji *regsubsets*.

```
> # Wybór najlepszego podzbioru
> titanic_bs <- regsubsets(Survived ~ .,
+                          data = titanic,
+                          nvmax = 19,
+                          really.big = TRUE)
> titanic_bs_sum <- summary(titanic_bs)
```

W celu znalezienia najlepszego podzbioru skorzystamy z optymalnych statystyk:

- C_p Mallowsa

```
> titanic_cp_min <-  which.min(titanic_bs_sum$cp)
> titanic_cp_min

[1] 7

> # Wybrane predyktory
> titanic_model_cp <- titanic_bs_sum$which[titanic_cp_min, -1]
> titanic_predictors_cp <- names(which(titanic_model_cp == TRUE))
> titanic_predictors_cp

[1] "Pclass"     "Sex"        "Age"        "Embarked"   "Has_Cabin"
[6] "FamilySize" "IsAlone"
```

- BIC

```
> titanic_bic_min <-  which.min(titanic_bs_sum$bic)
> titanic_bic_min

[1] 6

> titanic_model_bic <- titanic_bs_sum$which[titanic_bic_min, -1]
> # Wybrane predyktory
> titanic_predictors_bic <- names(which(titanic_model_bic == TRUE))
> titanic_predictors_bic

[1] "Pclass"     "Sex"        "Age"        "Has_Cabin"  "FamilySize"
[6] "IsAlone"
```

- Skorygowane R^2

```
> titanic_r_squared_max <- which.max(titanic_bs_sum$adjr2)
> titanic_r_squared_max

[1] 9

> titanic_model_r_squared <- titanic_bs_sum$which[titanic_r_squared_max, -1]
> # Wybrane predyktory
> titanic_predictors_r_squared <- names(which(titanic_model_r_squared == TRUE))
> titanic_predictors_r_squared

[1] "Pclass"     "Sex"        "Age"        "Fare"       "Embarked"
[6] "Has_Cabin"  "FamilySize" "IsAlone"    "Title"
```

3.2 Selekcja krokowa do przodu

Używamy funkcji *regsubsets*.

```
> # Wybór najlepszego podzbioru
> titanic_fwd <- regsubsets(Survived ~ ., data = titanic, method = "forward",
+                       nvmax = 19)
> titanic_fwd_sum <- summary(titanic_fwd)
```

W celu znalezienia najlepszego podzbioru skorzystamy z optymalnych statystyk:

- C_p Mallowsa

```
> titanic_fwd_cp_min <-  which.min(titanic_fwd_sum$cp)
> titanic_fwd_model_cp <- titanic_fwd_sum$which[titanic_fwd_cp_min, -1]
> # Wybrane predyktory
> titanic_fwd_predictors_cp <- names(which(titanic_fwd_model_cp == TRUE))
> titanic_fwd_predictors_cp
```

```
[1] "Pclass"     "Sex"        "Age"        "Embarked"   "Has_Cabin"
[6] "FamilySize" "IsAlone"
```

- BIC

```
> titanic_fwd_bic_min <-  which.min(titanic_fwd_sum$bic)
> titanic_fwd_model_bic <- titanic_fwd_sum$which[titanic_fwd_bic_min, -1]
> # Wybrane predyktory
> titanic_fwd_predictors_bic <- names(which(titanic_fwd_model_bic == TRUE))
> titanic_fwd_predictors_bic

[1] "Pclass"     "Sex"        "Age"        "Has_Cabin"  "FamilySize"
[6] "IsAlone"
```

- Skorygowane R^2

```
> titanic_fwd_r_squared_max <-  which.max(titanic_fwd_sum$adjr2)
> titanic_fwd_model_r_squared <- titanic_fwd_sum$which[titanic_fwd_r_squared_max, -1]
> # Wybrane predyktory
> titanic_fwd_predictors_r_squared <- names(which(titanic_fwd_model_r_squared == TRUE))
> titanic_fwd_predictors_r_squared

[1] "Pclass"     "Sex"        "Age"        "Fare"       "Embarked"
[6] "Has_Cabin"  "FamilySize" "IsAlone"    "Title"
```

3.3 Selekcja krokowa wstecz

Używamy funkcji *regsubsets*.

```
> # Wybór najlepszego podzbioru
> titanic_back <- regsubsets(Survived ~ ., data = titanic, nvmax = 19,
+                       method = "backward")
> titanic_back_sum <- summary(titanic_back)
```

W celu znalezienia najlepszego podzbioru skorzystamy z optymalnych statystyk:

- C_p Mallowsa

```
> titanic_back_cp_min <-  which.min(titanic_back_sum$cp)
> titanic_back_model_cp <- titanic_back_sum$which[titanic_back_cp_min, -1]
> # Wybrane predyktory
> titanic_back_predictors_cp <- names(which(titanic_back_model_cp == TRUE))
> titanic_back_predictors_cp

[1] "Pclass"     "Sex"        "Age"        "Embarked"   "Has_Cabin"
[6] "FamilySize" "IsAlone"
```

- BIC

```
> titanic_back_bic_min <-  which.min(titanic_back_sum$bic)
> titanic_back_model_bic <- titanic_back_sum$which[titanic_back_bic_min, -1]
> # Wybrane predyktory
> titanic_back_predictors_bic <- names(which(titanic_back_model_bic == TRUE))
> titanic_back_predictors_bic

[1] "Pclass"     "Sex"        "Age"        "Has_Cabin"  "FamilySize"
[6] "IsAlone"
```

- Skorygowane R^2

```
> titanic_back_r_squared_max <-  which.max(titanic_back_sum$adjr2)
> titanic_back_model_r_squared <- titanic_back_sum$which[titanic_back_r_squared_max, -1
> # Wybrane predyktory
> titanic_back_predictors_r_squared <- names(which(titanic_back_model_r_squared == TRUE
> titanic_back_predictors_r_squared

[1] "Pclass"     "Sex"        "Age"        "Fare"       "Embarked"
[6] "Has_Cabin"  "FamilySize" "IsAlone"    "Title"
```

3.4 Selekcja cech metodą regularyzacji lasso

Przygotowujemy dane i dopasowujemy model lasso. W tym celu ustawiamy parametr $\alpha = 1$. Jest to zadanie klasyfikacji, zatem ustawiamy także parametr family na 'binomial'.

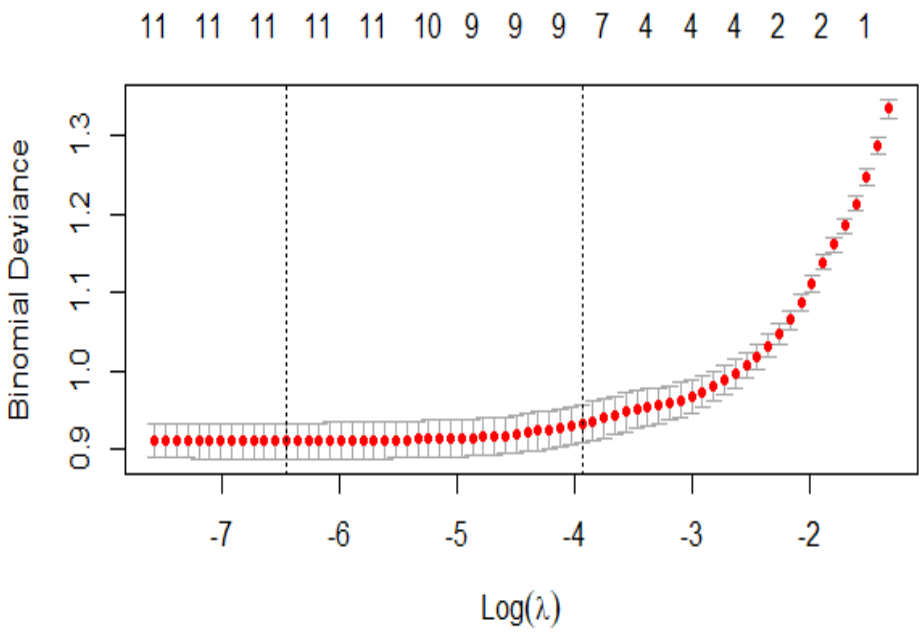
```
> titanic_data <- model.matrix(Survived ~ ., data = titanic)[, -1]
> titanic_target <- titanic$Survived
> titanic_fit_lasso <- glmnet(titanic_data, titanic_target, alpha = 1,
+                             family = 'binomial')
```

Narysujemy teraz wykres zależności Binomial Deviance od $\log(\lambda)$.

Optymalna wartość λ to według nas e^{-4}. Wybieramy współczynniki przy pomocy funkcji *predict*.

```
> titanic_pred_lasso <- predict(titanic_fit_lasso, s = exp(-4),
+                           type = 'coefficients')
> titanic_pred_lasso

12 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept)  2.37629697
X            .
Pclass      -0.64493571
Sex         -2.12576018
Age         -0.19148240
Parch        .
Fare         0.01221780
```



Rysunek 2: Zależność Binomial Deviance od $\log(\lambda)$

```
Embarked     0.09457447
Has_Cabin    0.53083235
FamilySize  -0.08093536
IsAlone     -0.04574183
Title        0.08050636
```

Wartości współcynników dla zmiennych, przy których widzimy symbol ".", są dokładnie równe 0.