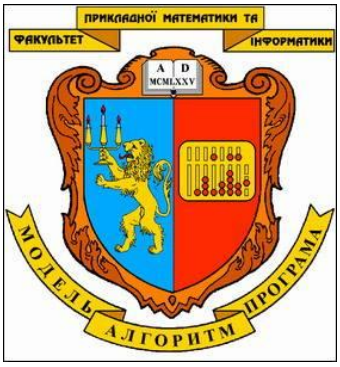
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет прикладної математики та інформатики
Кафедра програмування



Індивідуальне завдання № 5
Вибір лінійної моделі та регуляризація

Виконала:
студентка групи ПМОм-11
Кравець Ольга

Львів 2025

Хід роботи

Варіант - 3

Визначаю значення змінної variant. Встановлюю set.seed(3) та генерую redundant як випадкове ціле число з рівномірного розподілу

```
> variant=3
> variant
[1] 3
> set.seed(variant)
> redundant=floor(runif(1,5,25))
> redundant
[1] 8
```

Завдання 1.

Модифікувала дані Auto

```
> Auto_new=Auto[-sample(1:nrow(Auto), round((redundant / 100) * nrow(Auto))), ]
> fix(Auto new)
```

	row.names	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	var11	var12	var13
1	1	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu			
2	2	15	8	350	165	3693	11.5	70	1	buick skylark 320			
3	3	18	8	318	150	3436	11	70	1	plymouth satellite			
4	4	16	8	304	150	3433	12	70	1	amc rebel sst			
5	5	17	8	302	140	3449	10.5	70	1	ford torino			
6	6	15	8	429	198	4341	10	70	1	ford galaxie 500			
7	7	14	8	454	220	4354	9	70	1	chevrolet impala			
8	8	14	8	440	215	4312	8.5	70	1	plymouth fury iii			
9	9	14	8	455	225	4425	10	70	1	pontiac catalina			
10	10	15	8	390	190	3850	8.5	70	1	amc ambassador dpl			
11	11	15	8	383	170	3563	10	70	1	dodge challenger se			
12	13	15	8	400	150	3761	9.5	70	1	chevrolet monte carlo			
13	14	14	8	455	225	3086	10	70	1	buick estate wagon (sw)			
14	16	22	6	198	95	2833	15.5	70	1	plymouth duster			
15	17	18	6	199	97	2774	15.5	70	1	amc hornet			
16	18	21	6	200	85	2587	16	70	1	ford maverick			
17	19	27	4	97	88	2130	14.5	70	3	datsun pl510			
18	20	26	4	97	46	1835	20.5	70	2	volkswagen 1131 deluxe sedan			
19	21	25	4	110	87	2672	17.5	70	2	peugeot 504			
20	23	25	4	104	95	2375	17.5	70	2	saab 99e			

Розбила дані на навчальний та тестовий набори

```
> set.seed(variant)
> test_size=round((2 * redundant / 100) * nrow(Auto_new))
> test_ind=sample(1:nrow(Auto_new), test_size)
> Auto_test=Auto_new[test_ind, ]
> Auto train=Auto new[-test ind, ]
```

Викинула name. Лінійна модель на основі методу найменших квадратів

```
> Auto_train$name=NULL
> Auto_test$name=NULL
>
> lm_fit=lm(mpg ~ ., data = Auto_train)
> lm_pred=predict(lm_fit, Auto_test)
> lm_mse=mean((lm_pred - Auto_test$mpg)^2)
> lm_mse
[1] 14.12197
```

Модель гребеневої регресії

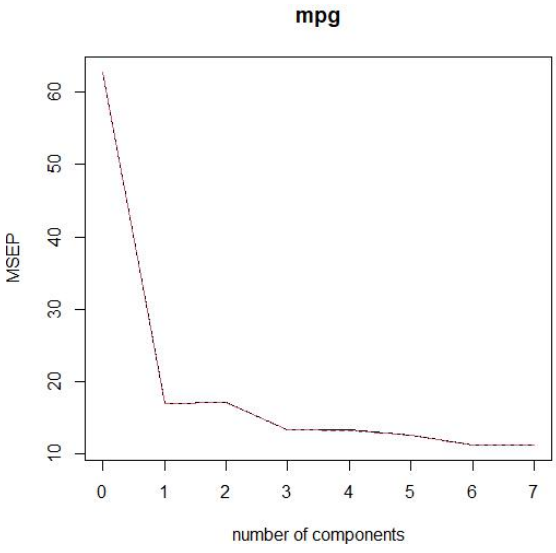
```
> x_train=model.matrix(mpg ~ ., Auto_train)[, -1]
> y_train=Auto_train$mpg
> x_test=model.matrix(mpg ~ ., Auto_test)[, -1]
> y_test=Auto_test$mpg
>
> set.seed(variant)
> cv_ridge=cv.glmnet(x_train, y_train, alpha = 0)
> ridge_pred=predict(cv_ridge, s = cv_ridge$lambda.min, newx = x_test)
> ridge_mse=mean((ridge_pred - y_test)^2)
> ridge_mse
[1] 13.77018
```

Модель ласо

```
> cv_lasso=cv.glmnet(x_train, y_train, alpha = 1)
> lasso_pred=predict(cv_lasso, s = cv_lasso$lambda.min, newx = x_test)
> lasso_mse=mean((lasso_pred - y_test)^2)
> lasso_mse
[1] 13.93794
```

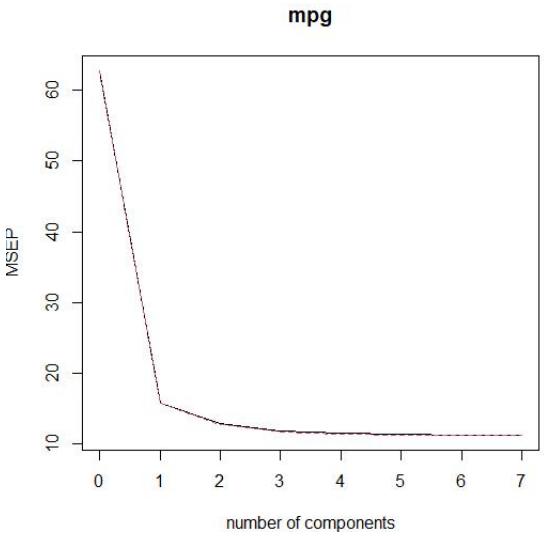
Модель PCR

```
> pcr_fit=pcr(mpg ~ ., data = Auto_train, scale = TRUE, validation = "CV")
> validationplot(pcr_fit, val.type = "MSEP")
> pcr_ncomp=which.min(pcr_fit$validation$PRESS)
> pcr_pred=predict(pcr_fit, Auto_test, ncomp = pcr_ncomp)
> pcr_mse=mean((pcr_pred - Auto_test$mpg)^2)
> pcr_mse
[1] 14.29458
```



Модель PLS

```
> set.seed(variant)
> pls_fit=plsr(mpg ~ ., data = Auto_train, scale = TRUE, validation = "CV")
> validationplot(pls_fit, val.type = "MSEP")
> pls_ncomp=which.min(pls_fit$validation$PRESS)
> pls_pred=predict(pls_fit, Auto_test, ncomp = pls_ncomp)
> pls_mse=mean((pls_pred - Auto_test$mpg)^2)
> pls_mse
[1] 14.12298
```



Для кожної з моделей оцінила тестову помилку

```
> results=data.frame(
+   Model = c("Linear", "Ridge", "Lasso", "PCR", "PLS"),
+   Test_MSE = c(lm_mse, ridge_mse, lasso_mse, pcr_mse, pls_mse)
+ )
> print(results)
   Model Test_MSE
1 Linear 14.12197
2  Ridge 13.77018
3  Lasso 13.93794
4    PCR 14.29458
5    PLS 14.12298
```

Завдання 2.

2. Встановивши попередньо seed, що дорівнює значенню змінної variant, використайте функцію rnorm() та згенеруйте предиктор X довжиною n = 100 * (1+variant %/% 10) з середнім μ = [variant/5] + 1 та середньоквадратичним відхиленням σ = [(2*variant)^(1/2)]+1), та вектор залишків ε такої ж довжини n з параметрами μ = 0 та σ = 1). Виберіть β_0, β_1, β_2 і β_3 (попередньо встановивши seed, що дорівнює значенню змінної variant) як реалізації рівномірно розподіленої випадкової величини на відрізку [-10, 10] заокруглені до найближчого цілого та обчисліть

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon.$$

```
> set.seed(variant)
> n = 100 * (1 + variant %/% 10)
> mu = (variant / 5) + 1
> sigma = sqrt(2 * variant) + 1
>
> set.seed(variant)
> x = rnorm(n, mu, sigma)
> eps = rnorm(n, 0, 1)
>
> set.seed(variant)
> b0 = round(runif(1, -10, 10))
> b1 = round(runif(1, -10, 10))
> b2 = round(runif(1, -10, 10))
> b3 = round(runif(1, -10, 10))
```

Для виконання наступних завдань, об’єднала необхідні предиктори в один датафрейм

```
> xy_dataframe = data.frame(y,
+                           x,
+                           x2 = x^2,
+                           x3 = x^3,
+                           x4 = x^4,
+                           x5 = x^5,
+                           x6 = x^6,
+                           x7 = x^7,
+                           x8 = x^8,
+                           x9 = x^9,
+                           x10 = x^10)
>
```

Використовуючи функцію regsubsets() вибрала найкращу модель методом вибору найкращої підмножини з множини предикторів X, X^2 , . . ., X^{10}.

```
> library(leaps)
>
> set.seed(variant)
> best_subset_model = regsubsets(y ~ ., data = xy_dataframe, nvmax = 10)
> best_subset_summary = summary(best_subset_model)
>
> par(mfrow = c(2, 2))
```

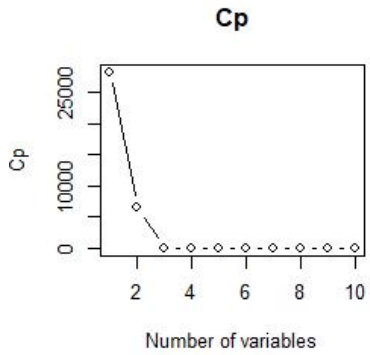
Яка модель найкраща за показниками C_p, *BIC і скорегований* R^2 *?*

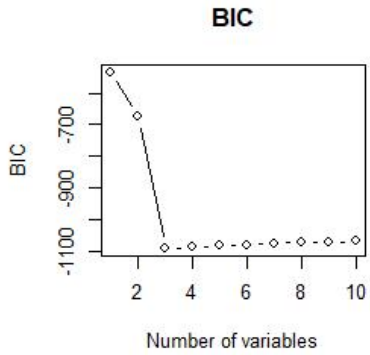
```
> set.seed(variant)
> best_subset_model = regsubsets(y ~ ., data = xy_dataframe, nvmax = 10)
> best_subset_summary = summary(best_subset_model)
>
> par(mfrow = c(2, 2))
>
> plot(best_subset_summary$cp, xlab = "Number of variables", ylab = "Cp", type = "b", main = "Cp")
> plot(best_subset_summary$bic, xlab = "Number of variables", ylab = "BIC", type = "b", main = "BIC")
> plot(best_subset_summary$adjr2, xlab = "Number of variables", ylab = "Adjusted R2", type = "b", main = "Adjusted R2")
>
> best_cp = which.min(best_subset_summary$cp)
> best_bic = which.min(best_subset_summary$bic)
> best_adjr2 = which.max(best_subset_summary$adjr2)
```

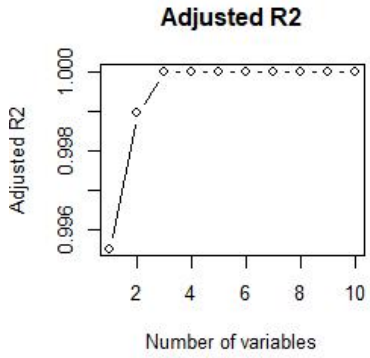
```
> cat("The best model Cp:\n")
The best model Cp:
> print(coef(best_subset_model, best_cp))
(Intercept)           x          x2          x3
  -6.973545    5.947367   -1.994704   -2.999569
>
> cat("\nThe best model BIC:\n")

The best model BIC:
> print(coef(best_subset_model, best_bic))
(Intercept)           x          x2          x3
  -6.973545    5.947367   -1.994704   -2.999569
>
> cat("\nThe best model Adjusted R2:\n")

The best model Adjusted R2:
> print(coef(best_subset_model, best_adjr2))
  (Intercept)             x            x2            x3            x5            x6            x7            x8            x9           x10
-6.948556e+00  5.372283e+00 -2.044550e+00 -2.712351e+00 -3.782885e-02  1.977547e-03  1.737674e-03 -1.470506e-04 -2.434344e-05  2.505909e-06
```







За C_p і BIC найкраща модель має 3 змінні: x, x^2, x^3.

За скорегованим R^2, модель з найбільшим R^2 включає ще більше змінних: від x до x^{10}.

Використала методи покрокового вибору вперед та назад та порівняла результати з результатами вибору найкращої підмножини.

```
> step_forward=step(lm(y ~ 1, data = xy_dataframe), scope = ~., direction = "forward")
Start:  AIC=1127.69
y ~ 1
```

```
> step_backward=step(lm(y ~ ., data = xy_dataframe), direction = "backward")
Start:  AIC=29.05
y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

       Df Sum of Sq    RSS     AIC
- x4    1      0.15 107.46  27.195
- x6    1      0.87 108.18  27.864
<none>              107.31  29.052
- x9    1      7.10 114.40  33.456
- x7    1      7.57 114.88  33.868
- x5    1      8.00 115.31  34.242
- x10   1      8.47 115.78  34.651
- x8    1      8.52 115.83  34.696
- x2    1    145.13 252.44 112.599
- x     1    374.85 482.16 177.310
- x3    1    592.32 699.62 214.537

Step:  AIC=27.2
y ~ x + x2 + x3 + x5 + x6 + x7 + x8 + x9 + x10

       Df Sum of Sq     RSS     AIC
<none>               107.46  27.195
- x6    1      7.64  115.10  32.063
- x8    1      8.81  116.27  33.076
- x5    1      8.88  116.34  33.136
- x10   1      8.94  116.40  33.188
- x9    1      9.27  116.73  33.470
- x7    1      9.32  116.78  33.513
- x     1    381.27  488.73 176.664
- x3    1    647.08  754.54 220.094
- x2    1   1188.34 1295.81 274.172
> cat("Cp:\n", coef(best_subset_model, which.min(best_subset_summary$cp)), "\n")
Cp:
 -6.973545 5.947367 -1.994704 -2.999569
> cat("BIC:\n", coef(best_subset_model, which.min(best_subset_summary$bic)), "\n")
BIC:
 -6.973545 5.947367 -1.994704 -2.999569
> cat("R^2:\n", coef(best_subset_model, which.max(best_subset_summary$adjr2)), "\n")
R^2:
 -6.948556 5.372283 -2.04455 -2.712351 -0.03782885 0.001977547 0.001737674 -0.0001470506 -2.434344e-05 2.505909e-06
> print(summary(step_forward))

Call:
lm(formula = y ~ 1, data = xy_dataframe)

Residuals:
     Min       1Q   Median       3Q      Max
-1235.79   -81.42   132.57   143.17   749.40

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -150.51      27.96  -5.383 4.94e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 279.6 on 99 degrees of freedom
```

```
> print(summary(step_backward))

Call:
lm(formula = y ~ x + x2 + x3 + x5 + x6 + x7 + x8 + x9 + x10,
    data = xy_dataframe)

Residuals:
     Min       1Q   Median       3Q      Max
-2.08356 -0.72361 -0.08313  0.70165  2.78144

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.949e+00  2.121e-01 -32.767  < 2e-16 ***
x            5.372e+00  3.006e-01  17.870  < 2e-16 ***
x2          -2.045e+00  6.481e-02 -31.548  < 2e-16 ***
x3          -2.712e+00  1.165e-01 -23.280  < 2e-16 ***
x5          -3.783e-02  1.387e-02  -2.727  0.00768 **
x6           1.978e-03  7.818e-04   2.529  0.01316 *
x7           1.738e-03  6.219e-04   2.794  0.00636 **
x8          -1.471e-04  5.413e-05  -2.716  0.00791 **
x9          -2.434e-05  8.737e-06  -2.786  0.00650 **
x10          2.506e-06  9.157e-07   2.737  0.00748 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.093 on 90 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 7.203e+05 on 9 and 90 DF,  p-value: < 2.2e-16
```

Моделі після покрокового вибору (вперед і назад) дають одну і ту ж фінальну модель y ~ $x+ x^2 + x^3+ x^5+ x^6+ x^7+ x^8+ x^9+ x^{10}$.

Моделі за допомогою найкращої підмножини дають різні набори змінних залежно від обраного критерію. У випадку C_p та BIC модель обирає лише x, x^2, x^3, тоді як R^2 включає всі 10 змінних.

Пристосувала модель ласо до згенерованих даних, використовуючи X, X^2, . . ., X^{10} як предиктори.

```
> x_lasso = model.matrix(y ~ ., data = xy_dataframe)[, -1]
> y_lasso = xy_dataframe$y
> set.seed(variant)
> cv_lasso_gen = cv.glmnet(x_lasso, y_lasso, alpha = 1)
> lasso_pred_gen = predict(cv_lasso_gen, s = cv_lasso_gen$lambda.min, newx = x_lasso)
> lasso_mse_gen = mean((lasso_pred_gen - y_lasso)^2)
> lasso_mse_gen
[1] 73.71023
```

Отримані оцінки коефіцієнтів моделі

```
> lasso_coef = coef(cv_lasso_gen, s = cv_lasso_gen$lambda.min)
> print(lasso_coef)
11 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept) -10.529006889
x             .
x2           -1.509607325
x3           -2.649972149
x4           -0.008811359
x5           -0.003397106
x6            .
x7            .
x8            .
x9            .
x10           .
```

Значущими залишилися лише x^2, x^3, x^4, x^5, що свідчить про наявність нелінійної залежності.

Найкращою моделлю, як нам показує модель ласо, буде:

-1.51 * x^2 – 2.65 * x^3 – 0.0088 * x^4 – 0.0034 * x^5 – 10.53.