Untitled

Lukyanova Anastasia

16 марта 2018 г

Оценим стандартную ошибку модели для линейных регрессионных моделей а) со всеми объясняющими переменными; б) только с непрерывными объясняющими переменными Будем использовать методы: \* методом проверочной выборки с долей обучающей 50%; \* методом LOOCV; \* k-кратной кросс-валидацией с k=5 и k=10

Загрузим данные и проведем все необходимые вычисления:

model2 <- lm(crim ~ indus + age + indus:chas,  
 data = Boston)  
summary(model2)

##   
## Call:  
## lm(formula = crim ~ indus + age + indus:chas, data = Boston)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.832 -2.848 -0.602 1.084 81.220   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.00331 0.91112 -4.394 1.36e-05 \*\*\*  
## indus 0.39838 0.06616 6.022 3.34e-09 \*\*\*  
## age 0.04949 0.01609 3.076 0.00221 \*\*   
## indus:chas -0.24299 0.09782 -2.484 0.01331 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.768 on 502 degrees of freedom  
## Multiple R-squared: 0.1893, Adjusted R-squared: 0.1845   
## F-statistic: 39.08 on 3 and 502 DF, p-value: < 2.2e-16

#общее число наблюдений  
m <- nrow(Boston)  
  
# доля обучающей выборки  
train.percent <- 0.5   
  
# выбрать наблюдения в обучающую выборку  
inTrain <- sample(m, m \* train.percent)  
inTrain

## [1] 496 242 304 409 322 237 270 100 504 119 157 306 393 166 131 472 217  
## [18] 224 113 8 499 263 489 444 151 440 484 192 41 376 422 355 139 425  
## [35] 247 201 14 365 69 58 4 324 40 197 133 434 120 435 173 310 336  
## [52] 21 66 350 469 121 498 51 174 413 283 165 363 240 255 500 271 44  
## [69] 228 134 234 126 62 502 56 169 264 39 218 273 353 205 159 10 305  
## [86] 171 29 439 53 349 235 92 464 212 430 340 493 77 476 260 262 346  
## [103] 329 48 438 258 375 59 282 364 135 167 408 246 149 381 143 317 411  
## [120] 327 470 194 301 102 222 433 437 64 455 261 20 230 216 172 210 474  
## [137] 9 275 378 441 373 404 137 331 189 175 454 128 339 147 293 182 209  
## [154] 105 7 312 37 2 463 61 289 482 396 229 156 399 110 325 83 112  
## [171] 313 390 436 475 190 326 142 421 417 269 91 1 395 344 357 202 497  
## [188] 82 129 384 87 368 35 501 405 250 371 506 86 297 244 179 24 106  
## [205] 54 163 251 486 290 148 278 410 94 279 208 374 505 199 400 136 138  
## [222] 16 416 141 358 447 65 186 18 257 391 451 442 445 213 50 233 38  
## [239] 80 457 291 70 47 443 462 245 335 473 214 220 274 123 307

# Линейная модель(a) ##############################################################  
  
# присоединить таблицу с данными: названия стоблцов будут доступны напрямую  
attach(Boston)  
# подгонка линейной модели на обучающей выборке  
fit.lm.1 <- lm(crim ~ indus + age + indus:chas,   
 subset = inTrain)  
  
# считаем MSE на тестовой выборке  
mean.a.1 <- mean((crim[-inTrain] - predict(fit.lm.1,  
 Boston[-inTrain, ]))^2)  
  
# отсоединить таблицу с данными  
detach(Boston)  
  
# Квадратичная модель ##########################################################  
  
# присоединить таблицу с данными: названия стоблцов будут доступны напрямую  
attach(Boston)  
# подгонка линейной модели на обучающей выборке  
fit.lm.2 <- lm(crim ~ poly(age,2) + poly(indus, 2) + indus:chas,   
 subset = inTrain)  
summary(fit.lm.2)

##   
## Call:  
## lm(formula = crim ~ poly(age, 2) + poly(indus, 2) + indus:chas,   
## subset = inTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.811 -3.255 -0.754 1.332 81.291   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.6720 0.4987 7.364 2.66e-12 \*\*\*  
## poly(age, 2)1 23.2768 15.4040 1.511 0.132045   
## poly(age, 2)2 23.6279 12.1698 1.942 0.053332 .   
## poly(indus, 2)1 55.5446 15.5473 3.573 0.000425 \*\*\*  
## poly(indus, 2)2 -28.6329 11.5890 -2.471 0.014161 \*   
## indus:chas -0.2073 0.1240 -1.671 0.095903 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.609 on 247 degrees of freedom  
## Multiple R-squared: 0.2019, Adjusted R-squared: 0.1857   
## F-statistic: 12.5 on 5 and 247 DF, p-value: 7.995e-11

fit.lm.2.1 <- lm(crim ~ poly(age,2) + poly(indus, 2),   
 subset = inTrain)  
summary(fit.lm.2.1)

##   
## Call:  
## lm(formula = crim ~ poly(age, 2) + poly(indus, 2), subset = inTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.539 -3.147 -0.768 1.410 81.689   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.4409 0.4808 7.156 9.34e-12 \*\*\*  
## poly(age, 2)1 23.0828 15.4592 1.493 0.136670   
## poly(age, 2)2 24.5234 12.2019 2.010 0.045535 \*   
## poly(indus, 2)1 52.6080 15.5035 3.393 0.000804 \*\*\*  
## poly(indus, 2)2 -26.1583 11.5356 -2.268 0.024213 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.636 on 248 degrees of freedom  
## Multiple R-squared: 0.1929, Adjusted R-squared: 0.1799   
## F-statistic: 14.82 on 4 and 248 DF, p-value: 7.201e-11

# считаем MSE на тестовой выборке  
mean.a.2 <- mean((crim[-inTrain] - predict(fit.lm.2.1,  
 Boston[-inTrain, ]))^2)  
  
# отсоединить таблицу с данными  
detach(Boston)  
  
# Кубическая модель ############################################################  
  
# присоединить таблицу с данными: названия стоблцов будут доступны напрямую  
attach(Boston)  
# подгонка линейной модели на обучающей выборке  
fit.lm.3 <- lm(crim ~ poly(age,3) + poly(indus, 3) + indus:chas,   
 subset = inTrain)   
summary(fit.lm.3)

##   
## Call:  
## lm(formula = crim ~ poly(age, 3) + poly(indus, 3) + indus:chas,   
## subset = inTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.618 -2.159 -0.102 0.774 79.747   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.6989 0.4821 7.673 3.98e-13 \*\*\*  
## poly(age, 3)1 16.9407 15.0483 1.126 0.26137   
## poly(age, 3)2 12.0428 12.1117 0.994 0.32105   
## poly(age, 3)3 3.2781 11.8966 0.276 0.78312   
## poly(indus, 3)1 59.9786 15.0113 3.996 8.54e-05 \*\*\*  
## poly(indus, 3)2 -31.2120 11.1706 -2.794 0.00562 \*\*   
## poly(indus, 3)3 -52.1177 11.4481 -4.553 8.35e-06 \*\*\*  
## indus:chas -0.2395 0.1195 -2.004 0.04613 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.317 on 245 degrees of freedom  
## Multiple R-squared: 0.2679, Adjusted R-squared: 0.247   
## F-statistic: 12.81 on 7 and 245 DF, p-value: 5.231e-14

# считаем MSE на тестовой выборке  
mean.a.3 <- mean((crim[-inTrain] - predict(fit.lm.3,  
 Boston[-inTrain, ]))^2)  
  
# отсоединить таблицу с данными  
detach(Boston)  
  
  
# k-кратная перекрёстная проверка ==============================================  
  
# оценим точность полиномиальных моделей, меняя степень  
# вектор с ошибками по 10-кратной кросс-валидации  
cv.err.k.fold <- rep(0, 5)  
names(cv.err.k.fold) <- 1:5  
# цикл по степеням полиномов  
for (i in 1:5){  
 fit.glm <- glm(crim ~ poly(age,3) + poly(indus, 3) + indus:chas, data = Boston)  
 cv.err.k.fold[i] <- cv.glm(Boston, fit.glm,  
 K = 10)$delta[1]  
}  
# результат  
cv.err.k.fold

## 1 2 3 4 5   
## 53.45825 54.01055 52.73518 52.89123 53.18092

age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio,  
rad, rm, tax, zn

Call: lm(formula = crim ~ poly(age, 2) + poly(indus, 2) + indus:chas, subset = inTrain)

Residuals: Min 1Q Median 3Q Max -8.750 -4.167 -0.516 1.500 80.284

Coefficients: Estimate Std. Error t value Pr(>|t|)  
(Intercept) 4.2824 0.5355 7.997 4.92e-14  ***poly(age, 2)1 44.4517 16.3242 2.723 0.00693***  *poly(age, 2)2 29.9490 13.0733 2.291 0.02281*   
poly(indus, 2)1 43.8684 16.1331 2.719 0.00701 \*\* poly(indus, 2)2 -34.2258 11.3467 -3.016 0.00283 \*\* indus:chas -0.3474 0.1393 -2.494 0.01328 \*  
— Signif. codes: 0 ‘***’ 0.001 ‘****’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘’ 1

Residual standard error: 8.211 on 247 degrees of freedom Multiple R-squared: 0.2253, Adjusted R-squared: 0.2096 F-statistic: 14.36 on 5 and 247 DF, p-value: 2.387e-12

Call: lm(formula = crim ~ poly(age, 2) + poly(indus, 2), subset = inTrain)

Residuals: Min 1Q Median 3Q Max -8.569 -3.976 -0.628 1.470 80.860

Coefficients: Estimate Std. Error t value Pr(>|t|)  
(Intercept) 3.9628 0.5254 7.543 8.7e-13  ***poly(age, 2)1 41.8737 16.4620 2.544 0.01158***   
**poly(age, 2)2 29.3688 13.2081 2.224 0.02708 \***  
**poly(indus, 2)1 41.2618 16.2678 2.536 0.01181 \***  
**poly(indus, 2)2 -31.5617 11.4146 -2.765 0.00612**  — Signif. codes: 0 ‘***’ 0.001 ‘****’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘’ 1

Residual standard error: 8.297 on 248 degrees of freedom Multiple R-squared: 0.2057, Adjusted R-squared: 0.1929 F-statistic: 16.06 on 4 and 248 DF, p-value: 1.043e-11

The following objects are masked from Boston (pos = 3):

age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio,  
rad, rm, tax, zn

The following objects are masked from Boston (pos = 4):

age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio,  
rad, rm, tax, zn

Call: lm(formula = crim ~ poly(age, 3) + poly(indus, 3) + indus:chas, subset = inTrain)

Residuals: Min 1Q Median 3Q Max -10.750 -2.486 -0.267 0.963 79.059

Coefficients: Estimate Std. Error t value Pr(>|t|)  
(Intercept) 4.0524 0.5157 7.858 1.23e-13  ***poly(age, 3)1 39.1007 15.7553 2.482 0.013745***   
**poly(age, 3)2 20.8278 12.7180 1.638 0.102777**  
**poly(age, 3)3 15.7821 12.2406 1.289 0.198500**  
**poly(indus, 3)1 52.7088 15.5849 3.382 0.000837**  *poly(indus, 3)2 -30.5887 11.0986 -2.756 0.006290*  ***poly(indus, 3)3 -51.5489 11.4935 -4.485 1.12e-05***  indus:chas -0.3850 0.1337 -2.879 0.004345 \*\* — Signif. codes: 0 ‘***’ 0.001 ‘****’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘’ 1

Residual standard error: 7.87 on 245 degrees of freedom Multiple R-squared: 0.294, Adjusted R-squared: 0.2739 F-statistic: 14.58 on 7 and 245 DF, p-value: 7.633e-16

1 2 3 4 5

53.35478 52.90878 52.97862 53.55968 52.93805