Упражнение №4

Галиченко Евгений

20 03 2020

На наборе данных из своего варианта построить модели линейной регрессии с указанными Y и X. Рассмотреть модели с категориальными предикторами, включая их взаимодействие с непрерывными объясняющими переменными. Сгенерировать отчёт по структуре отчёта из практики. Включить в отчёт выводы по каждому из разделов (описание данных, модели, сравнение с kNN). Ответить на вопрос, пригодна ли построенная модель регрессии для прогнозирования и почему. Код отчёта в файле .Rmd разместить на github.com, отчёт в формате html – на rpubs.com. Ссылки на репозиторий и отчёт на rpubs выслать на почту.

# Вариант 4

* Данные: Auto{ISLR};
* Зависимая переменная: mpg;
* Объясняющие переменные:

1. Непрерывные: weight, acceleration, year;
2. Дискретные (факторы): cylinders.

# Загрузка пакетов  
library('GGally') # Графики совместного разброса переменных  
library('lmtest') # Тесты остатков регрессионных моделей  
library('FNN') # Алгоритм kNN  
library('ISLR') # Данные Auto  
  
# Загрузка данных Auto  
data('Auto')  
# Отбор необходимых данных для построения моделей  
Auto <- Auto[,c('mpg', 'weight', 'acceleration', 'year', 'cylinders'), drop=FALSE]  
  
# константы  
my.seed <- 12345  
train.percent <- 0.85  
  
Auto$cylinders <- as.factor(Auto$cylinders)  
  
# обучающая выборка  
set.seed(my.seed)  
inTrain <- sample(seq\_along(Auto$mpg),   
 nrow(Auto) \* train.percent)  
df.train <- Auto[inTrain, c(colnames(Auto)[-1], colnames(Auto)[1])]  
df.test <- Auto[-inTrain,-1]

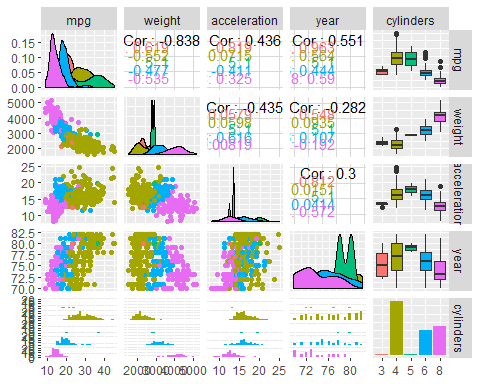
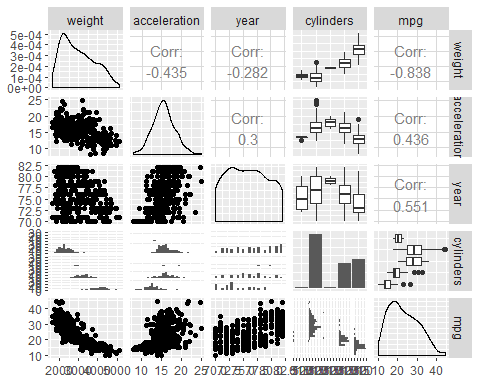
## Описание переменных

Набор данных Auto содержит переменные:

* mpg - миль на галлон;
* weight – вес автомобиля (кг.);
* acceleration – время ускорения от 0 до 60 миль в час (сек.);
* year – модельный год (по модулю 100)
* cylinders - количество цилиндров от 4 до 8

Размерность обучающей выборки: строк, объясняющих переменных. Зависимая переменная – mpg.

## weight acceleration year cylinders mpg   
## Min. :1613 Min. : 8.00 Min. :70.00 3: 4 Min. :10.00   
## 1st Qu.:2226 1st Qu.:13.70 1st Qu.:73.00 4:164 1st Qu.:17.00   
## Median :2833 Median :15.50 Median :76.00 5: 2 Median :22.00   
## Mean :2978 Mean :15.51 Mean :75.82 6: 76 Mean :23.13   
## 3rd Qu.:3613 3rd Qu.:17.00 3rd Qu.:79.00 8: 87 3rd Qu.:29.00   
## Max. :5140 Max. :24.80 Max. :82.00 Max. :44.60



Судя по коробчатой диаграмме на пересечении mpg и cylinders, средний число миль на галлон отличается в зависимости от кол-ва цилиндров в автомобиле: чем меньше кол-во цилиндров тем больше миль на галлон. Нижний правый график показывает, что доли наблюдений с различными значениями признака cylinders в наборе данных имеют следующий вид: наибольшую часть наблюдений отражает коэффициент со значением 4, 6 и 8 на равне, остальные практически отсутствуют.

## Модели

##   
## Call:  
## lm(formula = mpg ~ . + weight:cylinders + acceleration:cylinders +   
## year:cylinders, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.861 -1.538 0.000 1.416 9.226   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.788e+01 1.897e+02 0.252 0.801  
## weight 5.282e-03 1.202e-02 0.439 0.661  
## acceleration -3.520e+00 8.522e+00 -0.413 0.680  
## year 8.803e-02 1.376e+00 0.064 0.949  
## cylinders4 -6.943e+01 1.897e+02 -0.366 0.715  
## cylinders5 -6.861e+02 8.210e+02 -0.836 0.404  
## cylinders6 -4.816e+01 1.898e+02 -0.254 0.800  
## cylinders8 -5.473e+01 1.899e+02 -0.288 0.773  
## weight:cylinders4 -1.501e-02 1.204e-02 -1.247 0.213  
## weight:cylinders5 2.448e-01 2.977e-01 0.822 0.412  
## weight:cylinders6 -9.504e-03 1.208e-02 -0.787 0.432  
## weight:cylinders8 -7.997e-03 1.204e-02 -0.664 0.507  
## acceleration:cylinders4 3.747e+00 8.522e+00 0.440 0.660  
## acceleration:cylinders5 NA NA NA NA  
## acceleration:cylinders6 3.254e+00 8.524e+00 0.382 0.703  
## acceleration:cylinders8 3.601e+00 8.524e+00 0.422 0.673  
## year:cylinders4 8.119e-01 1.377e+00 0.590 0.556  
## year:cylinders5 NA NA NA NA  
## year:cylinders6 4.113e-01 1.379e+00 0.298 0.766  
## year:cylinders8 3.443e-01 1.381e+00 0.249 0.803  
##   
## Residual standard error: 2.635 on 315 degrees of freedom  
## Multiple R-squared: 0.8843, Adjusted R-squared: 0.8781   
## F-statistic: 141.7 on 17 and 315 DF, p-value: < 2.2e-16

Совместное влияние year:cylinders исключаем, т.к. параметры незначимы (наименее занчимы по сравнению с другими незначимыми коэффициентами).

##   
## Call:  
## lm(formula = mpg ~ . + weight:cylinders + acceleration:cylinders,   
## data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.4440 -1.6666 -0.0714 1.4746 9.7368   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.058e+01 4.550e+01 -0.892 0.373   
## weight 3.298e-04 6.344e-03 0.052 0.959   
## acceleration 2.945e-01 3.148e+00 0.094 0.926   
## year 7.475e-01 4.536e-02 16.478 <2e-16 \*\*\*  
## cylinders4 3.034e+01 4.521e+01 0.671 0.503   
## cylinders5 -3.188e+02 3.043e+02 -1.047 0.296   
## cylinders6 2.225e+01 4.528e+01 0.491 0.624   
## cylinders8 1.214e+01 4.530e+01 0.268 0.789   
## weight:cylinders4 -9.905e-03 6.371e-03 -1.555 0.121   
## weight:cylinders5 1.116e-01 1.102e-01 1.012 0.312   
## weight:cylinders6 -4.875e-03 6.455e-03 -0.755 0.451   
## weight:cylinders8 -2.634e-03 6.380e-03 -0.413 0.680   
## acceleration:cylinders4 -6.562e-02 3.149e+00 -0.021 0.983   
## acceleration:cylinders5 NA NA NA NA   
## acceleration:cylinders6 -5.454e-01 3.153e+00 -0.173 0.863   
## acceleration:cylinders8 -4.798e-01 3.154e+00 -0.152 0.879   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.712 on 318 degrees of freedom  
## Multiple R-squared: 0.8763, Adjusted R-squared: 0.8709   
## F-statistic: 160.9 on 14 and 318 DF, p-value: < 2.2e-16

Совместное влияние acceleration:cylinders исключаем, т.к. параметры незначимы (наименее занчимы по сравнению с другими незначимыми коэффициентами).

##   
## Call:  
## lm(formula = mpg ~ . + weight:cylinders, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.2163 -1.6578 -0.1493 1.5421 10.6267   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.649e+01 1.558e+01 -2.342 0.019799 \*   
## weight 4.547e-04 6.393e-03 0.071 0.943350   
## acceleration 6.829e-02 7.156e-02 0.954 0.340629   
## year 7.290e-01 4.444e-02 16.406 < 2e-16 \*\*\*  
## cylinders4 3.014e+01 1.546e+01 1.950 0.052058 .   
## cylinders5 -3.400e+02 9.472e+01 -3.590 0.000383 \*\*\*  
## cylinders6 1.825e+01 1.571e+01 1.162 0.246274   
## cylinders8 6.316e+00 1.564e+01 0.404 0.686615   
## weight:cylinders4 -9.945e-03 6.421e-03 -1.549 0.122415   
## weight:cylinders5 1.193e-01 3.297e-02 3.618 0.000344 \*\*\*  
## weight:cylinders6 -6.219e-03 6.470e-03 -0.961 0.337219   
## weight:cylinders8 -2.792e-03 6.430e-03 -0.434 0.664440   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.737 on 321 degrees of freedom  
## Multiple R-squared: 0.8728, Adjusted R-squared: 0.8685   
## F-statistic: 200.3 on 11 and 321 DF, p-value: < 2.2e-16

Взаимодействие weigth:cylinders так же удаляем

##   
## Call:  
## lm(formula = mpg ~ weight + acceleration + year + cylinders,   
## data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.3207 -1.9249 -0.0022 1.7471 11.3827   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.869e+01 4.104e+00 -4.554 7.45e-06 \*\*\*  
## weight -6.005e-03 4.555e-04 -13.183 < 2e-16 \*\*\*  
## acceleration 8.265e-02 7.700e-02 1.073 0.283929   
## year 6.960e-01 4.891e-02 14.230 < 2e-16 \*\*\*  
## cylinders4 6.654e+00 1.560e+00 4.265 2.63e-05 \*\*\*  
## cylinders5 7.931e+00 2.661e+00 2.980 0.003096 \*\*   
## cylinders6 3.497e+00 1.608e+00 2.174 0.030393 \*   
## cylinders8 5.849e+00 1.742e+00 3.358 0.000878 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.036 on 325 degrees of freedom  
## Multiple R-squared: 0.8415, Adjusted R-squared: 0.8381   
## F-statistic: 246.6 on 7 and 325 DF, p-value: < 2.2e-16

Параметр acceleration является незначимым, поэтому его тоже исключим из уравнения регресии.

##   
## Call:  
## lm(formula = mpg ~ weight + year + cylinders, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.3222 -2.0132 -0.0525 1.7773 11.7933   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.831e+01 4.089e+00 -4.476 1.05e-05 \*\*\*  
## weight -5.933e-03 4.506e-04 -13.167 < 2e-16 \*\*\*  
## year 7.031e-01 4.848e-02 14.504 < 2e-16 \*\*\*  
## cylinders4 6.933e+00 1.539e+00 4.505 9.25e-06 \*\*\*  
## cylinders5 8.255e+00 2.645e+00 3.122 0.00196 \*\*   
## cylinders6 3.687e+00 1.599e+00 2.306 0.02171 \*   
## cylinders8 5.695e+00 1.736e+00 3.280 0.00115 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.037 on 326 degrees of freedom  
## Multiple R-squared: 0.841, Adjusted R-squared: 0.838   
## F-statistic: 287.3 on 6 and 326 DF, p-value: < 2.2e-16

В данной модели все коэффициенты оказались значимыми, (). Cтоит остановиться на модели без взаимодействий. Проверим её остатки.

# Проверка остатков

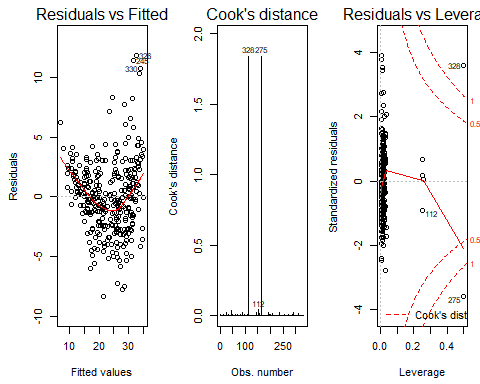
# тест Бройша-Пагана  
bptest(model.5)

##   
## studentized Breusch-Pagan test  
##   
## data: model.5  
## BP = 35.312, df = 6, p-value = 3.749e-06

# статистика Дарбина-Уотсона  
dwtest(model.5)

##   
## Durbin-Watson test  
##   
## data: model.5  
## DW = 2.0536, p-value = 0.6896  
## alternative hypothesis: true autocorrelation is greater than 0

# графики остатков  
par(mar = c(4.5, 4.5, 2, 1))  
par(mfrow = c(1, 3))  
plot(model.5, 1)  
plot(model.5, 4)  
plot(model.5, 5)

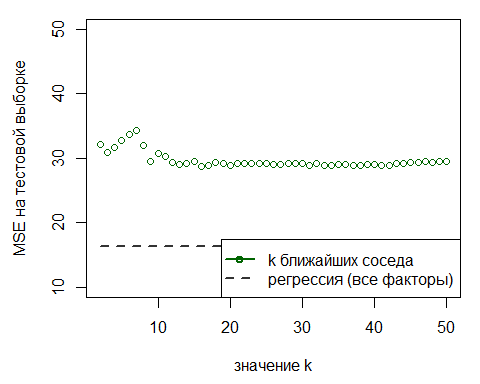


par(mfrow = c(1, 1))

Судя по графику слева, остатки случайны, и их дисперсия постоянна. В модели есть три влиятельных наблюдения: 112, 275, 328. Графики остатков не заставляют усомниться в том, что остатки удовлетворяют условиям Гаусса-Маркова.

# Сравнение с kNN

# фактические значения y на тестовой выборке  
y.fact <- Auto[-inTrain, 1]  
y.model.lm <- predict(model.5, df.test)  
MSE.lm <- sum((y.model.lm - y.fact)^2) / length(y.model.lm)  
  
df.train1 <- df.train  
df.test1 <- df.test  
df.train1$cylinders <- as.numeric(df.train1$cylinders)  
df.test1$cylinders <- as.numeric(df.test1$cylinders)  
  
# kNN требует на вход только числовые переменные  
df.train.num <- as.data.frame(apply(df.train1, 2, as.numeric))  
df.test.num <- as.data.frame(apply(df.test1, 2, as.numeric))  
  
for (i in 2:50){  
 model.knn <- knn.reg(train = df.train.num[, !(colnames(df.train.num) %in% 'mpg')],   
 y = df.train.num[, 'mpg'],   
 test = df.test.num, k = i)  
 y.model.knn <- model.knn$pred  
 if (i == 2){  
 MSE.knn <- sum((y.model.knn - y.fact)^2) / length(y.model.knn)  
 } else {  
 MSE.knn <- c(MSE.knn,   
 sum((y.model.knn - y.fact)^2) / length(y.model.knn))  
 }  
}  
  
# график  
par(mar = c(4.5, 4.5, 1, 1))  
# ошибки kNN  
# ошибка регрессии  
plot(2:50, MSE.knn, ylim = c(10,50), type = 'b', col = 'darkgreen',  
 xlab = 'значение k', ylab = 'MSE на тестовой выборке')  
lines(2:50, rep(MSE.lm, 49), lwd = 2, col = grey(0.2), lty = 2)  
legend('bottomright', lty = c(1, 2), pch = c(1, NA),   
 col = c('darkgreen', grey(0.2)),   
 legend = c('k ближайших соседа', 'регрессия (все факторы)'),   
 lwd = rep(2, 2))



Как можно видеть по графику, ошибка регрессии на тестовой выборке меньше, чем ошибка метода k ближайших соседей с k от 2 до 50. Ошибка регрессионной модели на тестовой выборке не очень велика и составляет от среднего значения зависимой переменной. У лучшей модели kNN точность ещё хуже: она ошибается на 21.3% от среднего значения объясняющей переменной.