Упражнение 4

Маркин Артем

23 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 12

* Данные: Carseats{ISLR};
* Зависимая переменная: Sales;
* Объясняющие переменные:

1. Непрерывные: Price, Population;
2. Дискретные (факторы): US.

# загрузка пакетов  
library('GGally') # графики совместного разброса переменных  
library('lmtest') # тесты остатков регрессионных моделей  
library('FNN') # алгоритм kNN  
library('ISLR')  
  
data('Carseats') # Загрузка данных Carseats  
# отбор необходимых данных для построения моделей  
Carseats <- Carseats[,c('Sales', 'Price', 'Population', 'US'), drop = FALSE]  
  
# Ядро  
my.seed <- 12345  
# Объем обучающей выборки  
train.percent <- 0.85  
  
# Преобразуем категориальную переменную US в фактор  
Carseats$US <- as.factor(Carseats$US)  
  
# обучающая выборка  
set.seed(my.seed)  
inTrain <- sample(seq\_along(Carseats$Price),   
 nrow(Carseats) \* train.percent)  
df.train <- Carseats[inTrain, c(colnames(Carseats)[-1], colnames(Carseats)[1])]  
df.test <- Carseats[-inTrain,-1]

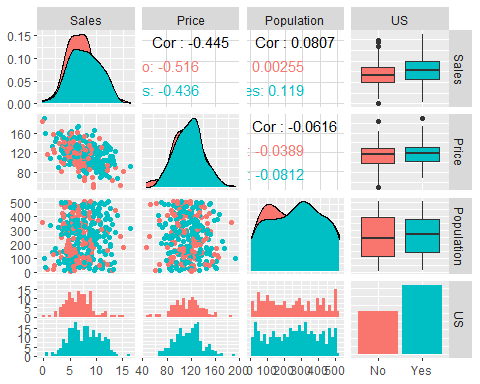
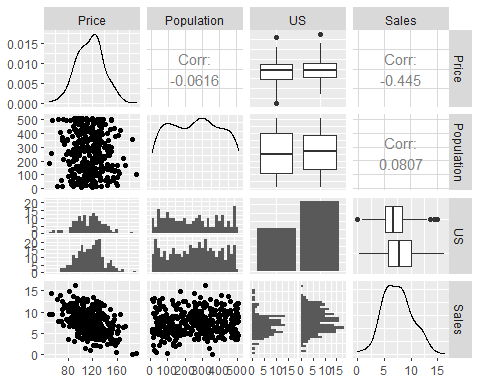
## Описание переменных

Набор данных Carseats содержит переменные:

* Sales - Объем продаж в каждом месте (в тысячах);
* Price – Взимаемая плата за автокресла на каждом участке;
* Population – Численность населения в регионе (в тысячах);
* US – Коэффициент с уровнями Нет и Да, чтобы указать, находится ли магазин в США или нет

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

## Price Population US Sales   
## Min. : 49.00 Min. : 10.0 No :130 Min. : 0.000   
## 1st Qu.: 99.75 1st Qu.:131.0 Yes:210 1st Qu.: 5.390   
## Median :117.50 Median :266.0 Median : 7.420   
## Mean :115.96 Mean :259.4 Mean : 7.453   
## 3rd Qu.:131.00 3rd Qu.:384.0 3rd Qu.: 9.160   
## Max. :191.00 Max. :509.0 Max. :16.270



Судя по коробчатой диаграмме на пересечении Sales и US, средний объем продаж отличается в зависимости от места нахождения магазина: если магазин находится в США, то больше объем продаж. В правом нижнем углу график показывает, что доли наблюдений с различными значениями признака USв наборе данных имеют следующий вид: наибольшую часть наблюдений отражает коэффициент со значением Yes.

## Модели

##   
## Call:  
## lm(formula = Sales ~ . + Price:US + Population:US, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.0481 -1.6833 -0.1061 1.5161 6.9566   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.1639519 1.1220252 11.732 < 2e-16 \*\*\*  
## Price -0.0548298 0.0090010 -6.092 3.08e-09 \*\*\*  
## Population -0.0002926 0.0014133 -0.207 0.836   
## USYes 0.6893426 1.4883987 0.463 0.644   
## Price:USYes -0.0002896 0.0117700 -0.025 0.980   
## Population:USYes 0.0019574 0.0018362 1.066 0.287   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.444 on 334 degrees of freedom  
## Multiple R-squared: 0.2438, Adjusted R-squared: 0.2325   
## F-statistic: 21.54 on 5 and 334 DF, p-value: < 2.2e-16

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

##   
## Call:  
## lm(formula = Sales ~ . + Population:US, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.0471 -1.6831 -0.1114 1.5198 6.9595   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.1835521 0.7889496 16.710 <2e-16 \*\*\*  
## Price -0.0549992 0.0057910 -9.497 <2e-16 \*\*\*  
## Population -0.0002936 0.0014106 -0.208 0.835   
## USYes 0.6552710 0.5447136 1.203 0.230   
## Population:USYes 0.0019600 0.0018306 1.071 0.285   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.44 on 335 degrees of freedom  
## Multiple R-squared: 0.2438, Adjusted R-squared: 0.2348   
## F-statistic: 27.01 on 4 and 335 DF, p-value: < 2.2e-16

Взаимодействие Population:US также исключаем.

##   
## Call:  
## lm(formula = Sales ~ Price + Population + US, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.9863 -1.5806 -0.1325 1.5451 7.0008   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12.9036971 0.7445554 17.331 < 2e-16 \*\*\*  
## Price -0.0551254 0.0057910 -9.519 < 2e-16 \*\*\*  
## Population 0.0008686 0.0009010 0.964 0.336   
## USYes 1.1599447 0.2730679 4.248 2.8e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.441 on 336 degrees of freedom  
## Multiple R-squared: 0.2413, Adjusted R-squared: 0.2345   
## F-statistic: 35.61 on 3 and 336 DF, p-value: < 2.2e-16

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

##   
## Call:  
## lm(formula = Sales ~ Price + US, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.9220 -1.6489 -0.0711 1.5512 7.0403   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.164318 0.693677 18.978 < 2e-16 \*\*\*  
## Price -0.055482 0.005779 -9.601 < 2e-16 \*\*\*  
## USYes 1.169747 0.272850 4.287 2.37e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.44 on 337 degrees of freedom  
## Multiple R-squared: 0.2392, Adjusted R-squared: 0.2346   
## F-statistic: 52.97 on 2 and 337 DF, p-value: < 2.2e-16

В данной модели все коэффициенты оказались значимыми, но по характеристикам качества модель недостаточно хороша (). Пробуем добавить взаимодействие Price:US.

##   
## Call:  
## lm(formula = Sales ~ Price + US + Price:US, data = df.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.9267 -1.6608 -0.0731 1.5627 7.0274   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.081595 1.049221 12.468 < 2e-16 \*\*\*  
## Price -0.054757 0.008995 -6.088 3.13e-09 \*\*\*  
## USYes 1.312370 1.382914 0.949 0.343   
## Price:USYes -0.001236 0.011750 -0.105 0.916   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.444 on 336 degrees of freedom  
## Multiple R-squared: 0.2392, Adjusted R-squared: 0.2324   
## F-statistic: 35.21 on 3 and 336 DF, p-value: < 2.2e-16

Очевидно, стоит остановиться на модели без взаимодействий. Проверим её остатки.

# Проверка остатков

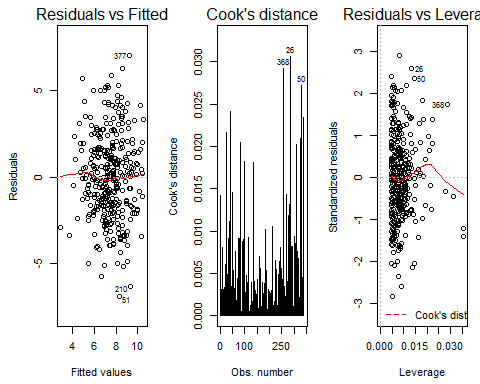
bptest(model.4) # Тест Бройша-Пагана

##   
## studentized Breusch-Pagan test  
##   
## data: model.4  
## BP = 5.3061, df = 2, p-value = 0.07044

dwtest(model.4) # Статистика Дарбина-Уотсона

##   
## Durbin-Watson test  
##   
## data: model.4  
## DW = 2.1041, p-value = 0.8312  
## 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.4, 1)  
plot(model.4, 4)  
plot(model.4, 5)

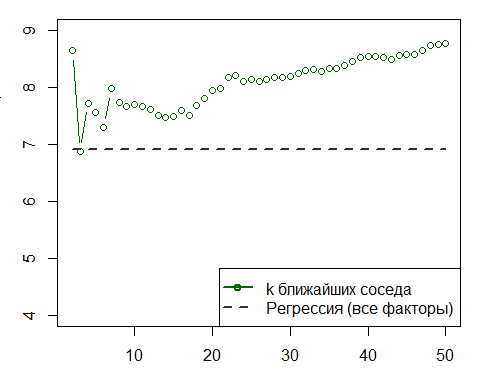


par(mfrow = c(1, 1))

Судя по графику слева, остатки случайны, и их дисперсия постоянна. В модели есть три влиятельных наблюдения: 26, 50, 368, – которые, однако, не выходят за пределы доверительных границ на третьем графике. Графики остатков не заставляют усомниться в том, что остатки удовлетворяют условиям Гаусса-Маркова.

# Сравнение с kNN

# Фактические значения y на тестовой выборке  
y.fact <- Carseats[-inTrain, 1]  
y.model.lm <- predict(model.4, 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$US <- as.numeric(df.train1$US)  
df.test1$US <- as.numeric(df.test1$US)  
  
# 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% 'Sales')],   
 y = df.train.num[, 'Sales'],   
 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))  
 }  
}  
par(mar = c(3, 3, 1, 1)) # График  
# ошибки kNN  
# ошибка регрессии  
plot(2:50, MSE.knn, ylim = c(4,9), 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 33.9% от среднего значения объясняющей переменной.  
Однако построенная модель регрессии () недостаточно пригодна для прогнозирования, так как её параметры недостатчно хорошо объясняют изменение объёма продаж ().