ML HW1

Runze Wang

2024-02-07

```
library(r02pro)     #INSTALL IF NECESSARY
library(tidyverse) #INSTALL IF NECESSARY
library(caret)
my_sahp <- sahp %>%
  na.omit() %>%
  select(gar_car, liv_area, kit_qual, sale_price)
my_sahp_train <- my_sahp[1:100, ]
my_sahp_test <- my_sahp[-(1:100), ]
```

##1

```
#a
lm_gar <- lm(sale_price ~ gar_car, data = my_sahp_train)
summary(lm_gar)
```

```
##
## Call:
## lm(formula = sale_price ~ gar_car, data = my_sahp_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -154.239  -43.754   -8.877   24.023  292.985
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   69.515     16.692   4.165 6.72e-05 ***
## gar_car       60.908      8.948   6.807 8.01e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.62 on 98 degrees of freedom
## Multiple R-squared:  0.321,  Adjusted R-squared:  0.3141
## F-statistic: 46.34 on 1 and 98 DF,  p-value: 8.011e-10
```

```
summary(lm_gar)$r.squared
```

```
## [1] 0.321036
```

```
lm_liv <- lm(sale_price ~ liv_area, data = my_sahp_train)
summary(lm_liv)
```

```
##
## Call:
## lm(formula = sale_price ~ liv_area, data = my_sahp_train)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -158.979  -30.101    1.136   23.119  220.986
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.89852   18.51621   0.481    0.632
## liv_area     0.11325    0.01207   9.386 2.61e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.91 on 98 degrees of freedom
## Multiple R-squared:  0.4734, Adjusted R-squared:  0.468
## F-statistic: 88.09 on 1 and 98 DF,  p-value: 2.606e-15
```

```
summary(lm_liv)$r.squared
```

```
## [1] 0.473387
```

```
lm_kit <- lm(sale_price ~ kit_qual, data = my_sahp_train)
summary(lm_kit)
```

```
##
## Call:
## lm(formula = sale_price ~ kit_qual, data = my_sahp_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.926 -27.928  -6.065  19.990 198.298
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         137.31       6.70  20.494  < 2e-16 ***
## kit_qualExcellent   209.61      17.87  11.732  < 2e-16 ***
## kit_qualFair        -19.52      29.46  -0.662    0.509
## kit_qualGood         56.61      10.94   5.174 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.69 on 96 degrees of freedom
## Multiple R-squared:  0.6068, Adjusted R-squared:  0.5945
## F-statistic: 49.37 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
summary(lm_kit)$r.squared
```

```
## [1] 0.6067548
```

gar_car: The sale price increases by 60.908 for every additional garage car capacity. The sale price is 69.515 when the garage car capacity is zero.

liv_area: The sale price increases by 0.11325 for every additional live area. The sale price is 8.89852 when the live area is zero.

kit_qual: The coefficient for kit_qual kit_qualExcellent is 209.61, so when kitchen quality is excellent, their sale price is 209.61 more than the baseline kitchen quality.

The coefficient for kit_qualFair is -19.52, so when the kitchen quality is fair, their sale price is 19.52 less than the baseline kitchen quality.

The coefficient for kit_qualGood is 56.61, so when the kitchen quality is good, their sale price is 56.61 more than the baseline kitchen quality.

The slope is 137.31, which means the expected sale price is 137.31 for the baseline kitchen quality.

The R^2 for each model is 0.321036, 0.473387, and 0.6067548 respectively. kit_qual has the largest R^2, which means it is most useful in predicting.

```
#b
#train error
pred_gar_train <- predict(lm_gar, newdata = my_sahp_train)
train_error_gar <- sum((pred_gar_train - my_sahp_train$sale_price)^2)

pred_liv_area <- predict(lm_liv, newdata = my_sahp_train)
train_error_liv <- sum((pred_liv_area - my_sahp_train$sale_price)^2)

pred_kit_qual <- predict(lm_kit, newdata = my_sahp_train)
train_error_kit <- sum((pred_kit_qual - my_sahp_train$sale_price)^2)

train_error_gar
```

```
## [1] 409245
```

```
train_error_liv
```

```
## [1] 317415.6
```

```
train_error_kit
```

```
## [1] 237028.3
```

```
#test
test_lm_gar <- lm(sale_price ~ gar_car, data = my_sahp_test)
test_lm_liv <- lm(sale_price ~ liv_area, data = my_sahp_test)
test_lm_kit <- lm(sale_price ~ kit_qual, data =my_sahp_test)

pred_gar_test <- predict(test_lm_gar, newdata = my_sahp_test)
test_error_gar <- sum((pred_gar_test - my_sahp_test$sale_price)^2)

pred_liv_test<- predict(test_lm_liv, newdata = my_sahp_test)
test_error_liv <- sum((pred_liv_test - my_sahp_test$sale_price)^2)

pred_kit_test <- predict(test_lm_kit, newdata = my_sahp_test)
test_error_kit <- sum((pred_kit_test - my_sahp_test$sale_price)^2)

test_error_gar
```

```
## [1] 280353.9
```

```
test_error_liv
```

```
## [1] 251088
```

```
test_error_kit
```

```
## [1] 266099.4
```

The training error for gar_car, liv_area, and kit_qual are 409245, 317415.6, and 237028.3 respectively. The test errors for gar_car, liv_area, and kit_qual are 280353.9, 251088, and 266099.4 respectively. liv_area has the lowest test error and does not have the largest R^2. The test error is used to predict the accuracy and R^2

is used to test how the model fits the value.

##Q2

```
lm_all <- lm(sale_price ~ gar_car + liv_area + kit_qual, data = my_sahp_train)
summary(lm_all)
```

```
##
## Call:
## lm(formula = sale_price ~ gar_car + liv_area + kit_qual, data = my_sahp_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.872 -21.532  -0.614  17.439 120.635
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        23.810321  14.030813   1.697    0.093 .
## gar_car            27.559749   5.830365   4.727 7.98e-06 ***
## liv_area            0.057576   0.009551   6.028 3.23e-08 ***
## kit_qualExcellent 141.628826  15.556563   9.104 1.48e-14 ***
## kit_qualFair      -29.978452  22.164214  -1.353    0.179
## kit_qualGood       21.639955   9.144491   2.366    0.020 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.32 on 94 degrees of freedom
## Multiple R-squared:  0.7827, Adjusted R-squared:  0.7712
## F-statistic: 67.73 on 5 and 94 DF,  p-value: < 2.2e-16
```

```
summary(lm_all)$r.squared
```

```
## [1] 0.7827407
```

```
lm_all_test <- lm(sale_price ~ gar_car + liv_area + kit_qual, data = my_sahp_test)
pred_train <- predict(lm_all, newdata = my_sahp_train)
train_error <- sum((my_sahp_train$sale_price - pred_train)^2)
pred_test <- predict(lm_all_test, newdata = my_sahp_test)
test_error <- sum((my_sahp_test$sale_price - pred_test)^2)

train_error
```

```
## [1] 130952.9
```

```
test_error
```

```
## [1] 125715.1
```

The coefficient on gar_car is equal to 27.559749, accounting for the impact of all the variables in the model, A house with a garage car capacity has an average of 27.559749 sale price than others with no garage car capacity.

The coefficient on liv_area is equal to 0.057576, accounting for the impact of all the variables in the model, A house with a living area has an average 0.057576 sale price than others with no living area.

The coefficient on kit_qualExcellent is equal to 141.628826, accounting for the impact of all the variables in the model, so for the kitchen quality to be excellent, their sale price is 141.628826 more than the baseline kitchen quality.

The coefficient on kit_qualFair is equal to -29.978452, accounting for the impact of all the variables in the model, so for the kitchen quality is fair, their sale price is 29.978452 less than the baseline kitchen quality.

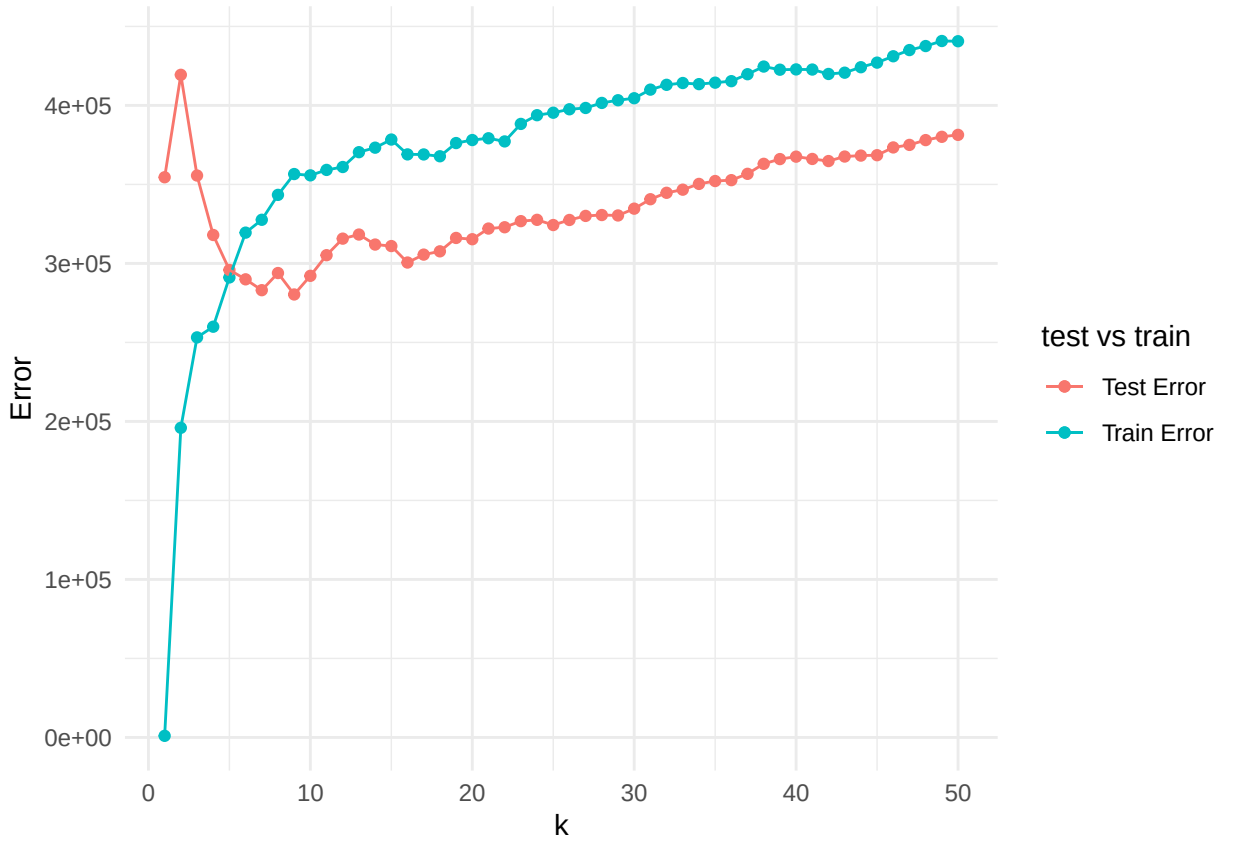
The coefficient on kit_qualGood is equal to 21.639955, accounting for the impact of all the variables in the model, so for the kitchen quality to be good, their sale price is 21.639955 more than the baseline kitchen quality.

When the gar_car, and liv_area are zero and the kitchen is in the baseline, the sale price is 23.810321.

The R^2 is 0.7827407 The training error is 130952.9 and the test error is 125715.1. The R^2 is greater than the three values we get from Q1, which means all variables in the model can be a better fit. The test errors are both less than the three values we get from Q1, which means the full model can predict data more accurately. The full model has better performance than the model with individual.

##Q3

```
k_seq <- 1:50
train_error <- numeric(length(k_seq))
test_error <- numeric(length(k_seq))
for (k in k_seq) {
  knn_model <- knnreg(sale_price ~ gar_car + liv_area + kit_qual, data = my_sahp_train, k = k)
  pred_train <- predict(knn_model, my_sahp_train)
  pred_test <- predict(knn_model, my_sahp_test)
   train_error[k] <- sum((my_sahp_train$sale_price - pred_train)^2)
  test_error[k] <- sum((my_sahp_test$sale_price - pred_test)^2)
}
error_data <- data.frame(k = k_seq, train_error = train_error, test_error = test_error)
ggplot(error_data, aes(x = k)) +
  geom_point(aes(y = train_error, colour = "Train Error")) +
  geom_point(aes(y = test_error, colour = "Test Error")) +
  geom_line(aes(y = train_error, colour = "Train Error", group = 1)) +
  geom_line(aes(y = test_error, colour = "Test Error", group = 1)) +
  labs(y = "Error", colour = "test vs train") +
  theme_minimal()
```



The training error increases as the K increases. The test error decreases as k increases, when k is greater than 9, the test error increases as k increases.

```
#b
train_error[10]
```

```
## [1] 355733
```

```
test_error[10]
```

```
## [1] 292141.5
```

From the basic concepts, the optimal k normally is the square root of the k, in this case, it should be 10. When k is equal to 10, the training error is 355733 and the test error is 292141.5. For Q2, the training error is 130952.9 and the test error is 125715.1. The regression has better performance than KNN in this case.

##Q4 we can substitute

$$\bar{x}$$

into x. so we can get

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}$$

.

$$\hat{y} = (\bar{y} - \hat{\beta}_1 \bar{x}) + \hat{\beta}_1 \bar{x}$$

$$\hat{y} = \bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 \bar{x}$$

$$\hat{y} = \bar{y}$$

so the least squares line always passes through the point $(\bar{x}, \bar{y})$.