housing_project

2024-04-20

```
#load packages
pacman::p_load(tidyverse, ggplot2, data.table, broom, parallel, here, zoo)

#load data
train_dt = read.csv('train.csv')
test_dt = read.csv('test.csv')
```

```
#pick out necessary variables from training dataset
#create age variable (year sold - year built)
house_df = train_dt %>% transmute(
  id = Id,
  sale_price = log(SalePrice),
  age = YrSold - YearBuilt,
  remod = YrSold - YearRemodAdd,
  area = GrLivArea,
  lot_area = LotArea,
  cond = OverallCond,
  veneer = MasVnrArea,
  bsmt_sf = TotalBsmtSF,
  bath = FullBath,
  bed_abv = BedroomAbvGr,
  kit_abv = KitchenAbvGr,
  rms_abv = TotRmsAbvGrd,
  fire = Fireplaces,
  grg_age = YrSold - GarageYrBlt,
  wd_dck = WoodDeckSF,
  cl_prch = EnclosedPorch,
  pool = PoolArea
)


#create function to remove NA values from all columns
rep_NA_func = function(data) {
  for (col in names(data)) {
    data[[col]] = na.aggregate(data[[col]])
  }
  return(data)
}
#new clean DF
house_clean_df = rep_NA_func(house_df)
```

```
#first OLS
simple_lm = lm(sale_price ~ age + remod + area + lot_area + cond + veneer + bsmt_sf
               + bath + bed_abv + kit_abv + rms_abv + fire + grg_age + wd_dck +
```

```
                cl_prch + pool,
              data = house_df)
#second OLS
clean_lm = lm(sale_price ~ age + remod + area + lot_area + cond + veneer + bsmt_sf
              + bath + bed_abv + kit_abv + rms_abv + fire + grg_age + wd_dck +
                cl_prch + pool,
              data = house_clean_df)

#output model summary
summary(simple_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ age + remod + area + lot_area + cond +
##     veneer + bsmt_sf + bath + bed_abv + kit_abv + rms_abv + fire +
##     grg_age + wd_dck + cl_prch + pool, data = house_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44668 -0.07715  0.00310  0.08089  0.55452
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.130e+01  4.552e-02 248.146  < 2e-16 ***
## age         -4.187e-03  3.310e-04 -12.649  < 2e-16 ***
## remod       -1.747e-03  3.440e-04  -5.077 4.38e-07 ***
## area         2.577e-04  1.953e-05  13.193  < 2e-16 ***
## lot_area     1.939e-06  4.753e-07   4.081 4.75e-05 ***
## cond         5.803e-02  5.215e-03  11.128  < 2e-16 ***
## veneer       7.087e-05  2.815e-05   2.518 0.011929 *
## bsmt_sf      1.646e-04  1.304e-05  12.623  < 2e-16 ***
## bath         3.325e-02  1.264e-02   2.630 0.008641 **
## bed_abv     -3.353e-02  8.326e-03  -4.027 5.96e-05 ***
## kit_abv     -1.673e-01  2.573e-02  -6.504 1.10e-10 ***
## rms_abv      3.059e-02  5.994e-03   5.104 3.80e-07 ***
## fire         7.549e-02  8.389e-03   8.998  < 2e-16 ***
## grg_age     -8.803e-04  3.493e-04  -2.520 0.011845 *
## wd_dck       9.676e-05  3.820e-05   2.533 0.011435 *
## cl_prch      2.564e-04  8.237e-05   3.112 0.001895 **
## pool        -4.365e-04  1.123e-04  -3.886 0.000107 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1671 on 1354 degrees of freedom
##   (89 observations deleted due to missingness)
## Multiple R-squared:  0.809,  Adjusted R-squared:  0.8068
## F-statistic: 358.5 on 16 and 1354 DF,  p-value: < 2.2e-16
```

```
summary(clean_lm)
```

```
##
## Call:
```

```
## lm(formula = sale_price ~ age + remod + area + lot_area + cond +
##     veneer + bsmt_sf + bath + bed_abv + kit_abv + rms_abv + fire +
##     grg_age + wd_dck + cl_prch + pool, data = house_clean_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50463 -0.07782  0.00699  0.08500  0.56374
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.121e+01  4.297e-02 260.969  < 2e-16 ***
## age         -4.992e-03  2.984e-04 -16.728  < 2e-16 ***
## remod       -1.801e-03  3.347e-04  -5.380 8.69e-08 ***
## area         2.719e-04  1.957e-05  13.893  < 2e-16 ***
## lot_area     1.914e-06  4.912e-07   3.897 0.000102 ***
## cond         6.283e-02  5.005e-03  12.553  < 2e-16 ***
## veneer       6.412e-05  2.897e-05   2.213 0.027044 *
## bsmt_sf      1.720e-04  1.311e-05  13.117  < 2e-16 ***
## bath         3.388e-02  1.263e-02   2.682 0.007407 **
## bed_abv     -2.988e-02  8.094e-03  -3.692 0.000231 ***
## kit_abv     -1.443e-01  2.317e-02  -6.227 6.22e-10 ***
## rms_abv      2.852e-02  6.044e-03   4.718 2.61e-06 ***
## fire         7.615e-02  8.521e-03   8.936  < 2e-16 ***
## grg_age     -8.385e-05  3.314e-04  -0.253 0.800260
## wd_dck       1.180e-04  3.894e-05   3.030 0.002486 **
## cl_prch      2.851e-04  8.211e-05   3.472 0.000533 ***
## pool        -4.766e-04  1.164e-04  -4.095 4.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1735 on 1443 degrees of freedom
## Multiple R-squared:  0.8135, Adjusted R-squared:  0.8114
## F-statistic: 393.4 on 16 and 1443 DF,  p-value: < 2.2e-16
```

```
#finding RMSE for simple model
simple_predictions = predict(simple_lm)

simple_residuals = simple_predictions - house_df$sale_price
```

```
## Warning in simple_predictions - house_df$sale_price: longer object length is
## not a multiple of shorter object length
```

```
simple_mse = mean(simple_residuals^2)

simple_rmse = sqrt(simple_mse)

print(simple_rmse)
```

```
## [1] 0.5227765
```

```
#finding RMSE for clean model
clean_predictions = predict(clean_lm)
```

```r
clean_residuals = clean_predictions - house_clean_df$sale_price

clean_mse = mean(clean_residuals^2)

clean_rmse = sqrt(clean_mse)

print(clean_rmse)
```

```
## [1] 0.1724514
```

RMSE for clean model is clearly better so I will use that for my prediction.

```r
#first clean up test data for prediction
#create age variable (year sold - year built)
predict_df = test_dt %>% transmute(
  id = Id,
  age = YrSold - YearBuilt,
  remod = YrSold - YearRemodAdd,
  area = GrLivArea,
  lot_area = LotArea,
  cond = OverallCond,
  veneer = MasVnrArea,
  bsmt_sf = TotalBsmtSF,
  bath = FullBath,
  bed_abv = BedroomAbvGr,
  kit_abv = KitchenAbvGr,
  rms_abv = TotRmsAbvGrd,
  fire = Fireplaces,
  grg_age = YrSold - GarageYrBlt,
  wd_dck = WoodDeckSF,
  cl_prch = EnclosedPorch,
  pool = PoolArea
)
```

```r
#replace all NA values
test_clean_df = rep_NA_func(predict_df)

# Predicting sale price in log scale
pred_log = predict(object = clean_lm, newdata = test_clean_df)

# Convert predicted sale prices from log scale to original scale
pred_original = exp(pred_log)

# View the predictions
head(pred_original)
```

```
##        1        2        3        4        5        6
## 115941.8 143672.8 194066.2 214615.8 160931.0 187556.6
```