ML_homework

Napat Teekasuk

2023-08-15

Load Library

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(readxl)
```

Import file

```
full_df <- read_excel("House Price India.xlsx", sheet = 1)
glimpse(full_df)
```

```
## Rows: 14,620
## Columns: 23
## $ id                                      <dbl> 6762810145, 6762810635, 676281~
## $ Date                                    <dbl> 42491, 42491, 42491, 42491, 42~
## $ `number of bedrooms`                    <dbl> 5, 4, 5, 4, 3, 3, 5, 3, 3, 4, ~
## $ `number of bathrooms`                   <dbl> 2.50, 2.50, 2.75, 2.50, 2.00, ~
## $ `living area`                           <dbl> 3650, 2920, 2910, 3310, 2710, ~
## $ `lot area`                              <dbl> 9050, 4000, 9480, 42998, 4500,~
## $ `number of floors`                      <dbl> 2.0, 1.5, 1.5, 2.0, 1.5, 1.0, ~
## $ `waterfront present`                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ `number of views`                       <dbl> 4, 0, 0, 0, 0, 0, 2, 0, 2, 0, ~
## $ `condition of the house`                <dbl> 5, 5, 3, 3, 4, 4, 3, 5, 4, 5, ~
## $ `grade of the house`                    <dbl> 10, 8, 8, 9, 8, 9, 10, 8, 8, 7~
## $ `Area of the house(excluding basement)` <dbl> 3370, 1910, 2910, 3310, 1880, ~
## $ `Area of the basement`                  <dbl> 280, 1010, 0, 0, 830, 900, 0, ~
## $ `Built Year`                            <dbl> 1921, 1909, 1939, 2001, 1929, ~
```

```
## $ `Renovation Year`                        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ `Postal Code`                            <dbl> 122003, 122004, 122004, 122005~
## $ Lattitude                                <dbl> 52.8645, 52.8878, 52.8852, 52.~
## $ Longitude                                <dbl> -114.557, -114.470, -114.468, ~
## $ living_area_renov                        <dbl> 2880, 2470, 2940, 3350, 2060, ~
## $ lot_area_renov                           <dbl> 5400, 4000, 6600, 42847, 4500,~
## $ `Number of schools nearby`               <dbl> 2, 2, 1, 3, 1, 1, 3, 3, 1, 2, ~
## $ `Distance from the airport`              <dbl> 58, 51, 53, 76, 51, 67, 72, 71~
## $ Price                                    <dbl> 2380000, 1400000, 1200000, 838~
```

check NA

```
full_df %>%
  complete.cases() %>%
  mean()
```

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

pre train model to find significant variable

```
pre_model <- train(Price ~ .,
                   data = full_df,
                   method = "lm")
varImp(pre_model)
```

```
## lm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                                Overall
## id                                            100.0000
## `\\`waterfront present\\``                     57.8019
## `\\`living area\\``                            50.2656
## `\\`grade of the house\\``                     43.9231
## `\\`Built Year\\``                             39.1810
## `\\`number of bedrooms\\``                     34.4984
## `\\`number of views\\``                        30.9693
## `\\`Postal Code\\``                            24.2488
## Lattitude                                      23.9589
## `\\`Area of the house(excluding basement)\\``  18.9100
## Longitude                                      14.5189
## `\\`number of bathrooms\\``                    13.9620
## `\\`number of floors\\``                       10.9579
## living_area_renov                               9.5075
## `\\`condition of the house\\``                  9.2087
## lot_area_renov                                  8.1308
## `\\`lot area\\``                                5.8154
## `\\`Renovation Year\\``                         5.2118
## `\\`Number of schools nearby\\``                0.9854
## `\\`Distance from the airport\\``               0.4768
```

prep data

```
top5_df <- full_df %>%
  select("waterfront" = "waterfront present",
         "living_area" = "living area" ,
         "grade_house" = "grade of the house",
         "built_year" = "Built Year",
         "bedrooms" = "number of bedrooms",
         "price" = Price)
```

take log price

```
top5_df  <- top5_df  %>%
  mutate(log_price = log(price))
```

Train Test split

```
split_data <- function(df){
  set.seed(8)
  n <- nrow(df)
  train_id <- sample(1:n, size = 0.8*n)
  train_df <- df[train_id, ]
  test_df <- df[-train_id, ]
  return(list(training = train_df,
              testing = test_df))
}

prep_data <- split_data(top5_df)
train_df <- prep_data[[1]]
test_df <- prep_data[[2]]
```

train model with price

```
set.seed(8)
lm_model <- train(price ~ waterfront + living_area + grade_house +
                          built_year + bedrooms,
                        data = train_df,
                        method = "lm")
lm_model
```

```
## Linear Regression
##
## 11696 samples
##     5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 11696, 11696, 11696, 11696, 11696, 11696, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   216164.5  0.6405045  139377.5
##
```

```
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

train model with log_price

```
set.seed(8)
lm_model_log <- train(log_price ~ waterfront + living_area + grade_house +
                        built_year + bedrooms,
                      data = train_df,
                      method = "lm")
lm_model_log
```

```
## Linear Regression
##
## 11696 samples
##     5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 11696, 11696, 11696, 11696, 11696, 11696, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.3165443  0.6336925  0.2519876
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

score model

- score model with price (test_df)

```
p <- predict(lm_model, newdata = test_df)
```

- score model with log_price (test_df)

```
p_log <- predict(lm_model_log, newdata = test_df)
```

evaluate model

- mean absolute error with price

```
mae <- mean(abs(p - test_df$price))
```

- root mean square error with price

```
rmse <- sqrt(mean((p - test_df$price)**2))
```

RMSE and MAE with price

```
cat("Test_mae:", mae ,
    "\nTest_rmse:", rmse)
```

```
## Test_mae: 142036.3
## Test_rmse: 238276.2
```

- mean absolute error with log_price

```
mae_log_test = mean(abs(exp(p_log) - exp(test_df$log_price)))
```

- root mean square error with log_price

```
rmse_log_test = sqrt( mean((exp(p_log) - exp(test_df$log_price))**2))
```

RMSE and MAE with log_price

```
cat("Test_mae_with_log_price:", mae_log_test ,
    "\nTest_rmse_with_log_price:", rmse_log_test)
```

```
## Test_mae_with_log_price: 139733.3
## Test_rmse_with_log_price: 295273.1
```

evaluate model with log_price by Train Data

```
p_train <- predict(lm_model_log, newdata=train_df)
```

- mean absolute error by Train Data

```
mae_log_train = mean(abs(exp(p_train) - exp(train_df$log_price)))
```

- root mean square error by Train Data

```
rmse_log_train = sqrt( mean((exp(p_train) - exp(train_df$log_price))**2))
```

RMSE and MAE with log_price by Train Data

```
cat("Train_mae_with_log_price:", mae_log_train ,
    "\nTrain_rmse_with_log_price:", rmse_log_train)
```

```
## Train_mae_with_log_price: 133785.7
## Train_rmse_with_log_price: 231729.8
```

Summary

- RMSE and MAE with price

```
cat("Train_mae:", lm_model$results[[4]] ,
    "\nTrain_rmse:", lm_model$results[[2]],
    "\nTest_mae:", mae ,
    "\nTest_rmse:", rmse)
```

```
## Train_mae: 139377.5
## Train_rmse: 216164.5
## Test_mae: 142036.3
## Test_rmse: 238276.2
```

- RMSE and MAE with log_price

```
cat("Train_mae_with_log_price:", mae_log_train ,
    "\nTrain_rmse_with_log_price:", rmse_log_train,
    "\nTest_mae_with_log_price:", mae_log_test ,
    "\nTest_rmse_with_log_price:", rmse_log_test)
```

```
## Train_mae_with_log_price: 133785.7
## Train_rmse_with_log_price: 231729.8
## Test_mae_with_log_price: 139733.3
## Test_rmse_with_log_price: 295273.1
```