ML_Homework_House_Price_India

2023-08-20

```r
## load library
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 ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dplyr)
library(readxl)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
## loading data from excel
house_price_india <- read_excel("House Price India.xlsx")
glimpse(house_price_india)
```

```
## 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~
```

```
head(house_price_india)
```

```
## # A tibble: 6 x 23
##           id  Date `number of bedrooms` `number of bathrooms` `living area`
##        <dbl> <dbl>                <dbl>                 <dbl>         <dbl>
## 1 6762810145 42491                    5                  2.5           3650
## 2 6762810635 42491                    4                  2.5           2920
## 3 6762810998 42491                    5                  2.75          2910
## 4 6762812605 42491                    4                  2.5           3310
## 5 6762812919 42491                    3                  2             2710
## 6 6762813105 42491                    3                  2.5           2600
## # i 18 more variables: `lot area` <dbl>, `number of floors` <dbl>,
## #   `waterfront present` <dbl>, `number of views` <dbl>,
## #   `condition of the house` <dbl>, `grade of the house` <dbl>,
## #   `Area of the house(excluding basement)` <dbl>,
## #   `Area of the basement` <dbl>, `Built Year` <dbl>, `Renovation Year` <dbl>,
## #   `Postal Code` <dbl>, Lattitude <dbl>, Longitude <dbl>,
## #   living_area_renov <dbl>, lot_area_renov <dbl>, ...
```

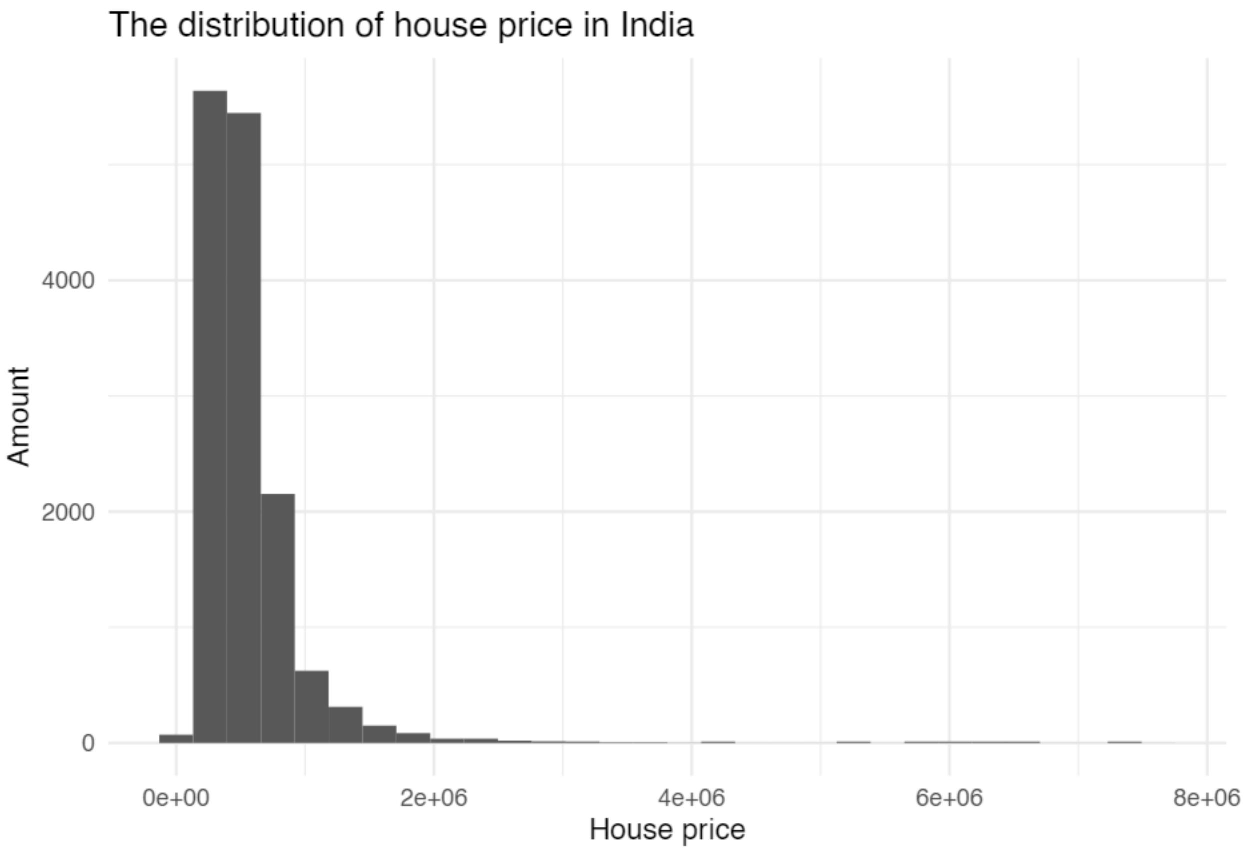
```
## subset the data
full_df <- house_price_india

## check NA
full_df %>%
  complete.cases() %>%
  mean()
```

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

```
## drop rows with NA
clean_df <- full_df %>%
  drop_na()
```

```
##check the distribution of the data
ggplot(full_df, aes(Price))+
  geom_histogram()+
  theme_minimal()+
  labs(
    title = "The distribution of house price in India",
    x = "House price",
    y = "Amount"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The distribution is right-skewed.

```
## 1.split data 80% train, 20% test
split_data <- function(df){
  n <- nrow(clean_df)
  train_id <- sample(1:n, size=0.8*n)
  train_df <- clean_df[train_id, ]
  test_df <- clean_df[-train_id, ]
  return(list(training = train_df,
              testing = test_df))
}

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

## 2.train model
lm_model <- train(Price ~.,
                  data = train_df,
                  # ML algorithm
                  method = "lm")
lm_model
```

```
## Linear Regression
##
## 11696 samples
##    22 predictor
##
```

```
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 11696, 11696, 11696, 11696, 11696, 11696, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   188882.9  0.740777  104670.9
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
## 3.score_model
p <- predict(lm_model, newdata = test_df)

## 4.evaluate model
(mae <- mean(abs(p - test_df$Price)))
```

```
## [1] 105074.2
```

```r
# root sme
(rmse <- sqrt(mean((p-test_df$Price)**2)))
```

```
## [1] 179892.6
```

Rsquared is not acceptable. So, we'll take log to the price in the training model.

```r
## 2nd times: prep data
clean_df <- full_df %>%
  mutate(log_price = log(Price))

## 1.split data 80% train, 20% test
split_data <- function(df){
  n <- nrow(clean_df)
  train_id <- sample(1:n, size=0.8*n)
  train_df <- clean_df[train_id, ]
  test_df <- clean_df[-train_id, ]
  return(list(training = train_df,
              testing = test_df))
}

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


## 2.train model
lm_model_log <- train(log_price ~.,
                  data = train_df,
                  # ML algorithm
                  method = "lm")
lm_model_log
```

```
## Linear Regression
##
## 11696 samples
##    23 predictor
##
## No pre-processing
```

```
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 11696, 11696, 11696, 11696, 11696, 11696, ...
## Resampling results:
##
##   RMSE        Rsquared   MAE
##   0.07283883  0.9808864  0.04365107
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
## 3.score_model
p <- predict(lm_model, newdata = test_df)

## 4.evaluate model
(mae <- mean(abs(p - test_df$Price)))
```

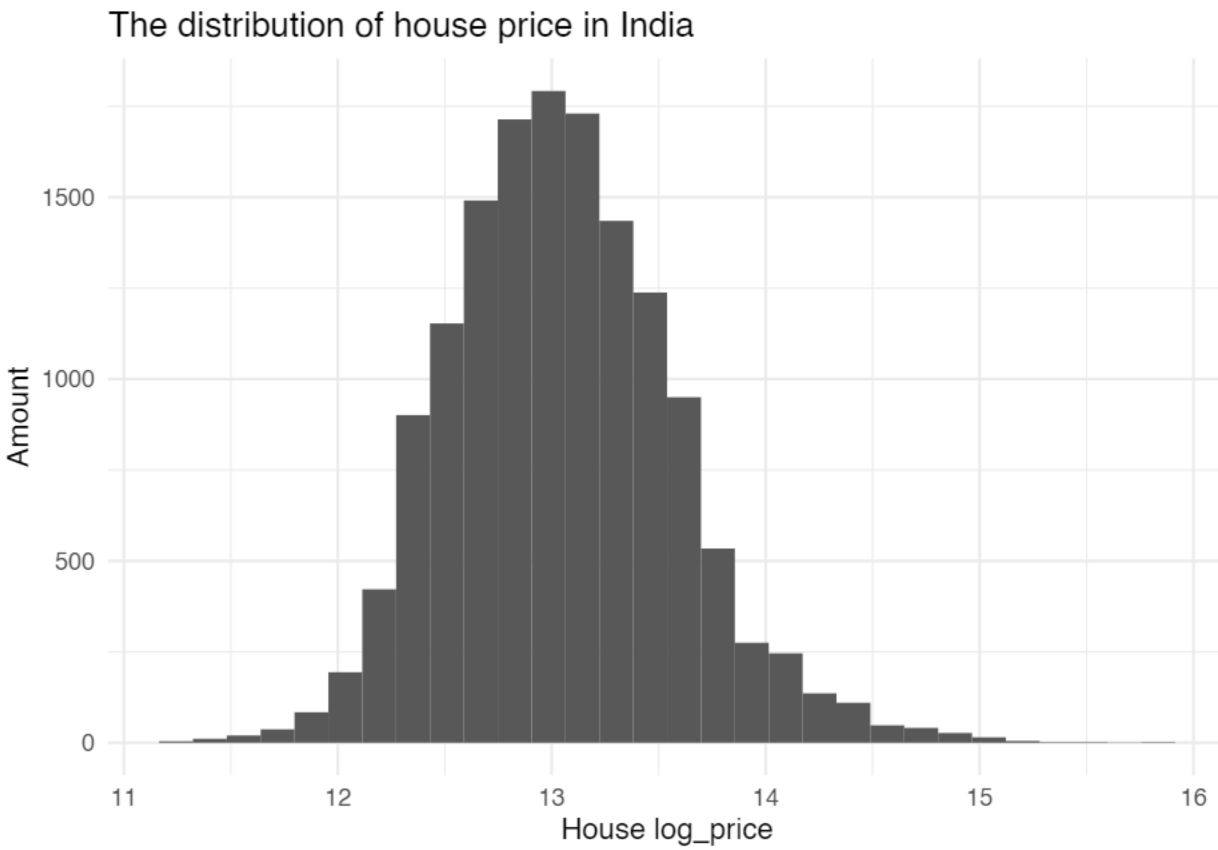
```
## [1] 105910.8
```

```r
# root sme
(rmse <- sqrt(mean((p-test_df$Price)**2)))
```

```
## [1] 180074.7
```

```r
##check the distribution of the data after take log_price
ggplot(clean_df, aes(log_price))+
  geom_histogram()+
  theme_minimal()+
  labs(
    title = "The distribution of house price in India",
    x = "House log_price",
    y = "Amount"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The distribution of the house price is now a normal distribution. And the Rsquared is acceptable.