Homework1

Phuwanate Lertsiriyothin

2023-08-25

Subject

Build the regression model to predict houses price in House Price India dataset (2017).

Load libraries

```
library(caret)
library(tidyverse)
library(ggplot2)
library(readr)
```

Prepare data

```
## Read the file.csv as data_frame
full_df <- read_csv("House 2017-Table 1.csv")

# Check missing values
full_df %>%
  complete.cases() %>%
  mean()
```

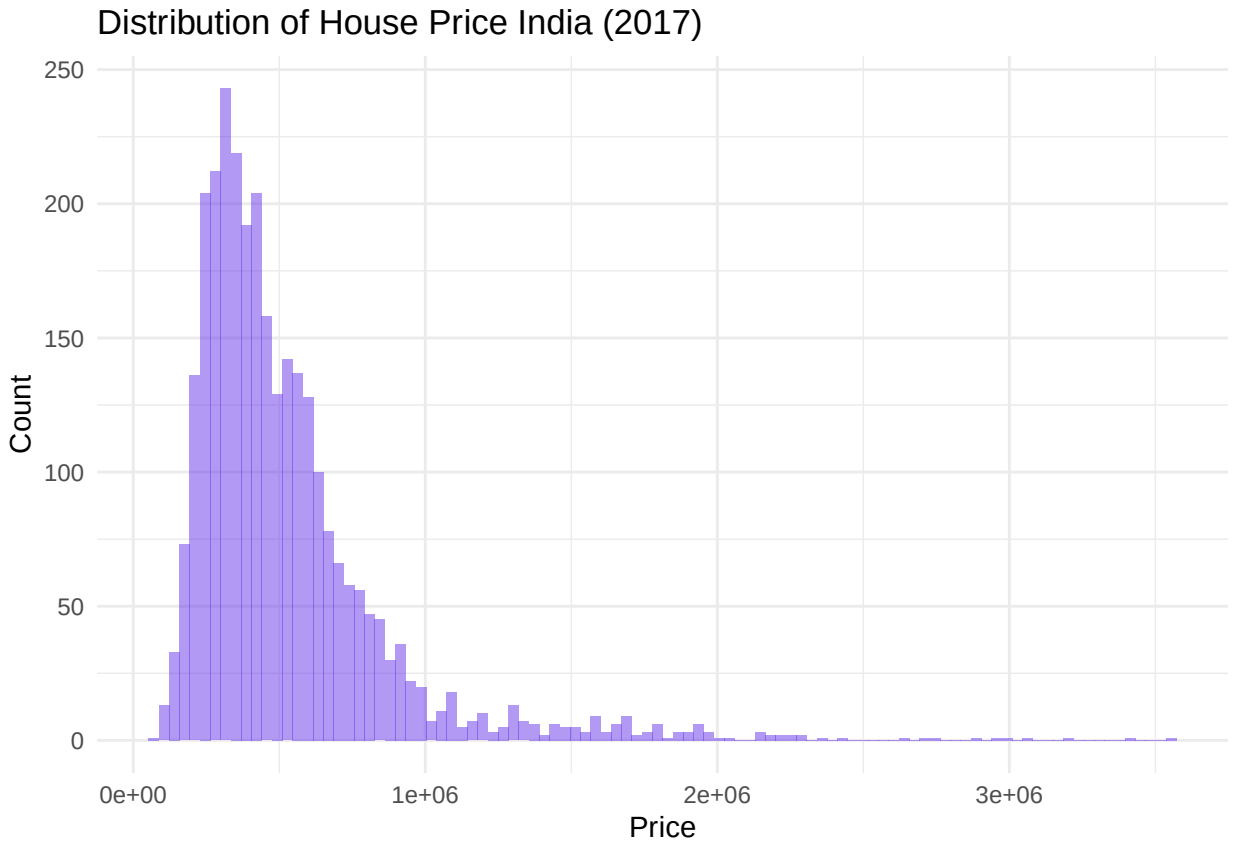
```
## [1] 1
```

Clean data

```
norm_clean_df <- full_df %>%
  subset(select = -id) #Delete unused column (id)

log_clean_df <- full_df %>%
  mutate(log_price = log(Price)) %>% #Add new column
  subset(select = -id) %>% #Delete unused column (id, Price)
  subset(select = -Price)
```

Check the distribution of prices

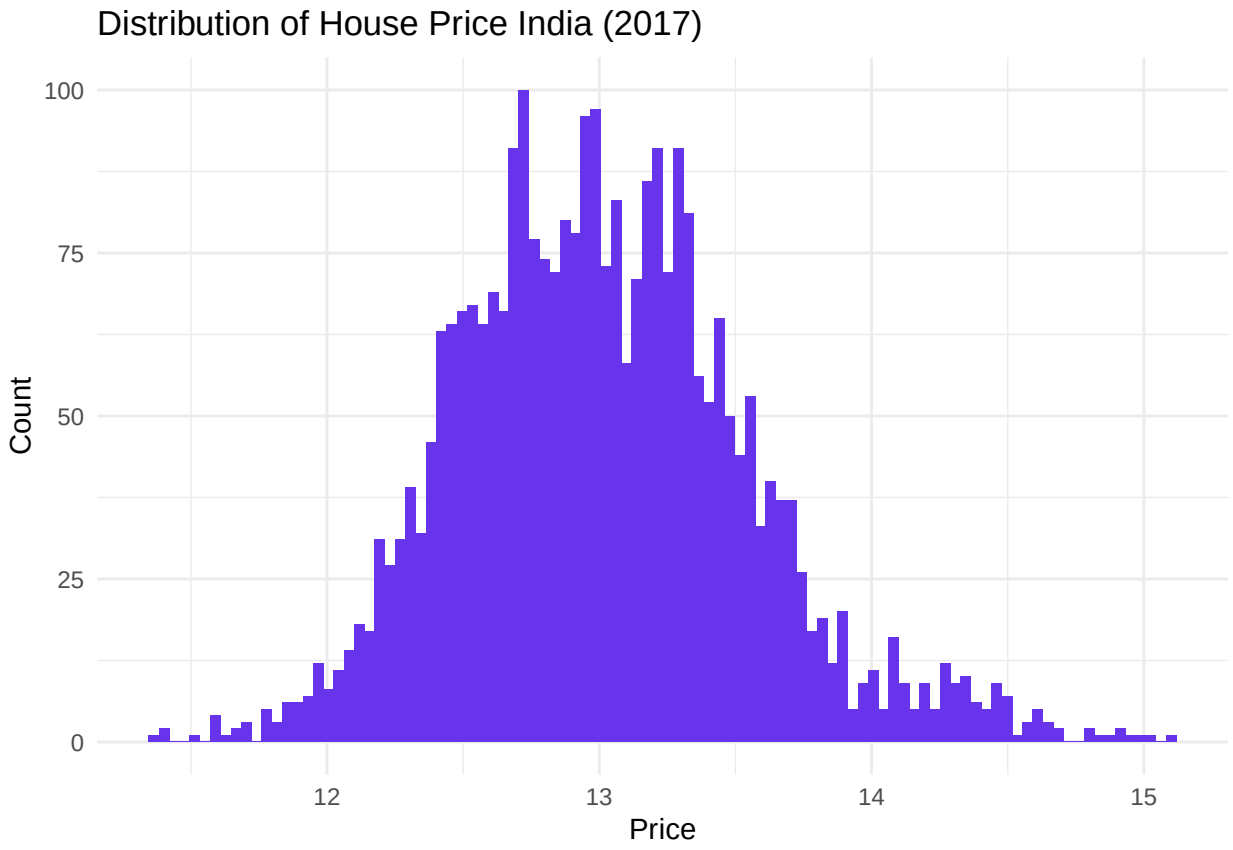
```
library(tidyverse)
library(ggplot2)
library(readr)

ggplot(data = full_df, aes(full_df$Price)) +
  geom_histogram(bins = 100,
                 fill = "#6734eb",
                 alpha = 0.5) +
  theme_minimal() +
    labs(title = "Distribution of House Price India (2017)",
         x = "Price",
         y = "Count")
```



The distribution of normal price is right skewed, for increasing the accuracy of price prediction we have to convert it to normal distribution by taking log to prices.

Convert to Normal Distribution

```
ggplot(data = log_clean_df, aes(log_price)) +
          geom_histogram(bins = 100,
                         fill = "#6734eb") +
          theme_minimal() +
          labs(title = "Distribution of House Price India (2017)",
               x = "Price",
               y = "Count")
```



Build ML Process

1. split data

Training set 80%, Testing set 20%

```
split_data <- function(df)
{
  set.seed(42)
  n <- nrow(df)
  id <- sample(1:n, size = 0.8 * n)
  train_df <- df[id, ]
  test_df <- df[-id, ]
  return (list(training = train_df, testing = test_df))
}

prep_data <- split_data(norm_clean_df)
prep_data2 <- split_data(log_clean_df)
```

2. train model

```
# Normal Price (Price must not be dependent variable)
(lm_model <- train(Price ~ .,
                   data=prep_data$training,
                   method="lm"))
```

```
## Linear Regression
##
## 2379 samples
##   21 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2379, 2379, 2379, 2379, 2379, 2379, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   200163.7  0.6800488  128009.2
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
varImp(lm_model)
```

```
## lm variable importance
##
##                                          Overall
## Lattitude                               100.0000
## `\\`grade of the house\\``               80.4779
## `\\`number of views\\``                  50.7718
## `\\`waterfront present\\``               50.7036
## `\\`Built Year\\``                       49.2346
```

```
## `\\`living area\\``                                39.8452
## `\\`number of bathrooms\\``                        22.7417
## living_area_renov                                  22.6317
## `\\`number of bedrooms\\``                         21.4670
## `\\`condition of the house\\``                     20.9051
## Date                                               18.2291
## `\\`Area of the house(excluding basement)\\``      14.7375
## Longitude                                          13.5268
## lot_area_renov                                      6.4553
## `\\`Renovation Year\\``                             6.0634
## `\\`Number of schools nearby\\``                    4.7316
## `\\`number of floors\\``                            4.5351
## `\\`Distance from the airport\\``                   3.4407
## `\\`Postal Code\\``                                 0.6905
## `\\`lot area\\``                                    0.0000
```

```
# Take log
(lm_model_log <- train(log_price ~ .,
                  data=prep_data2$training,
                  method="lm"))
```

```
## Linear Regression
##
## 2379 samples
##   21 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2379, 2379, 2379, 2379, 2379, 2379, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.2680705  0.7464025  0.2038711
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
varImp(lm_model_log)
```

```
## lm variable importance
##
##                                                 Overall
## Lattitude                                      100.0000
## `\\`grade of the house\\``                      58.6614
## `\\`Built Year\\``                              29.6717
## living_area_renov                               23.0614
## `\\`living area\\``                             22.1514
## `\\`number of views\\``                         21.2134
## `\\`condition of the house\\``                  18.5838
## `\\`Postal Code\\``                             17.9997
## `\\`number of bathrooms\\``                     17.2004
## `\\`number of floors\\``                        13.4909
## Date                                            12.9946
## `\\`waterfront present\\``                      12.9555
## `\\`lot area\\``                                 4.7522
## lot_area_renov                                   2.8686
## `\\`Area of the house(excluding basement)\\``    1.7773
```

```
## `\\`number of bedrooms\\``                    1.6824
## `\\`Number of schools nearby\\``               1.3876
## `\\`Renovation Year\\``                        1.1328
## Longitude                                     0.9283
## `\\`Distance from the airport\\``              0.0000
```

3. score model

```
# Predict normal prices
p_norm <- predict(lm_model, newdata=prep_data$testing)

# Predict log prices
p_log <- predict(lm_model_log, newdata=prep_data2$testing)
```

4. evaluate model

```
# Normal price model Evaluation
err_norm <- p_norm - prep_data$testing$Price
mae_norm <- mean(abs(err_norm))
rmse_norm <- sqrt(mean(err_norm ** 2))

# Log price model Evaluation
err_log <- exp(p_log) - exp(prep_data2$testing$log_price)
mae_log <- mean(abs((err_log)))
rmse_log <- sqrt(mean(err_log ** 2))

cat(paste("MAE Norm_Price = ",
          round(mae_norm, 1),
          " | ",
          "MAE Log_Price = ",
          round(mae_log,1)),
          "\n")
```

```
## MAE Norm_Price =  127940  |  MAE Log_Price =  111739.7
```

```
cat(paste("RMSE Norm_Price = ",
          round(rmse_norm, 1),
          " | ",
          "RMSE Log_Price = ",
          round(rmse_log,1),
          "\n"))
```

```
## RMSE Norm_Price =  200620  |  RMSE Log_Price =  194541.6
```

Summary

By taking log to normal prices (right skewed distribution), the accuracy of price prediction was increased.