House Sales Data Analysis in King County, WA

Christopher Wong

Created: 2023-04-25
Updated: 2025-01-18

SUMMARY

Online property companies use machine learning techniques to provide house valuations. This project aims to predict house sales in King County, Washington State, USA using Multiple Linear Regression (MLR). The dataset includes historical records of houses sold between May 2014 and May 2015. This project was created by Christopher Wong, for Professor You Liang (course MTH404).

DATA

The data for this project was sourced from the Kaggle dataset titled “KC_Housesales_Data”.
The dataset can be accessed at:
https://www.kaggle.com/swathiachath/kc-housesales-data

The dataset includes 21 independent variables such as bedrooms, sqft_living, view, and grade, with the dependent variable being price. It comprises 21,597 observations.

EXPLORATORY DATA ANALYSIS

Now we modify the data slightly and add two new columns for better understanding. The price might depend on the age of the house and the number of times it has been renovated. Therefore, we extract the age and renovation count of each house from our training data.

```r
# Modify the data to add age and renovation status
date_sale <- mdy(house_data$date)
house_data$sale_date_year <- as.integer(year(date_sale))
house_data$age <- house_data$sale_date_year - house_data$yr_built

house_data$reno <- ifelse(house_data$yr_renovated == 0, 0, 1)
house_data$reno <- as.factor(house_data$reno)
```

Training and test data

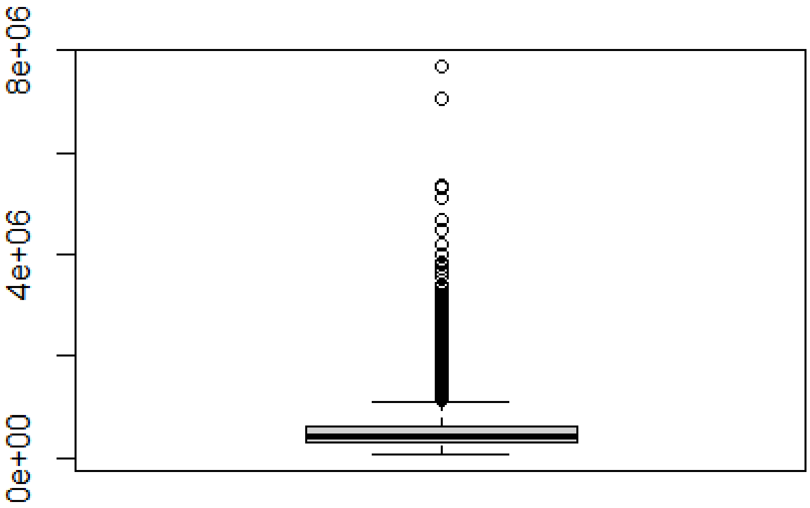
We divide the data to be the training set (80%) and test set (20%).

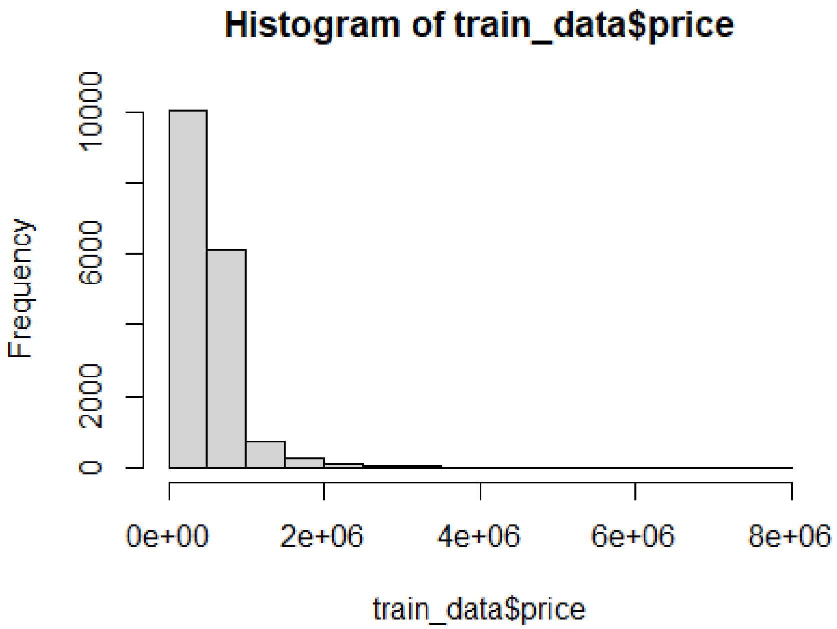
```r
set.seed(500827260)
# Get the number of rows in the dataset
n <- nrow(house_data)
# Calculate the number of test samples (20% of the data)
ntest <- trunc(0.2 * n)
# Randomly sample indices for the test set
```

```
testid <- sample(1:n, ntest)
# Split the data into training and test sets
train_data <- house_data[-testid, ]
test_data <- house_data[testid, ]
```

Check the response variables

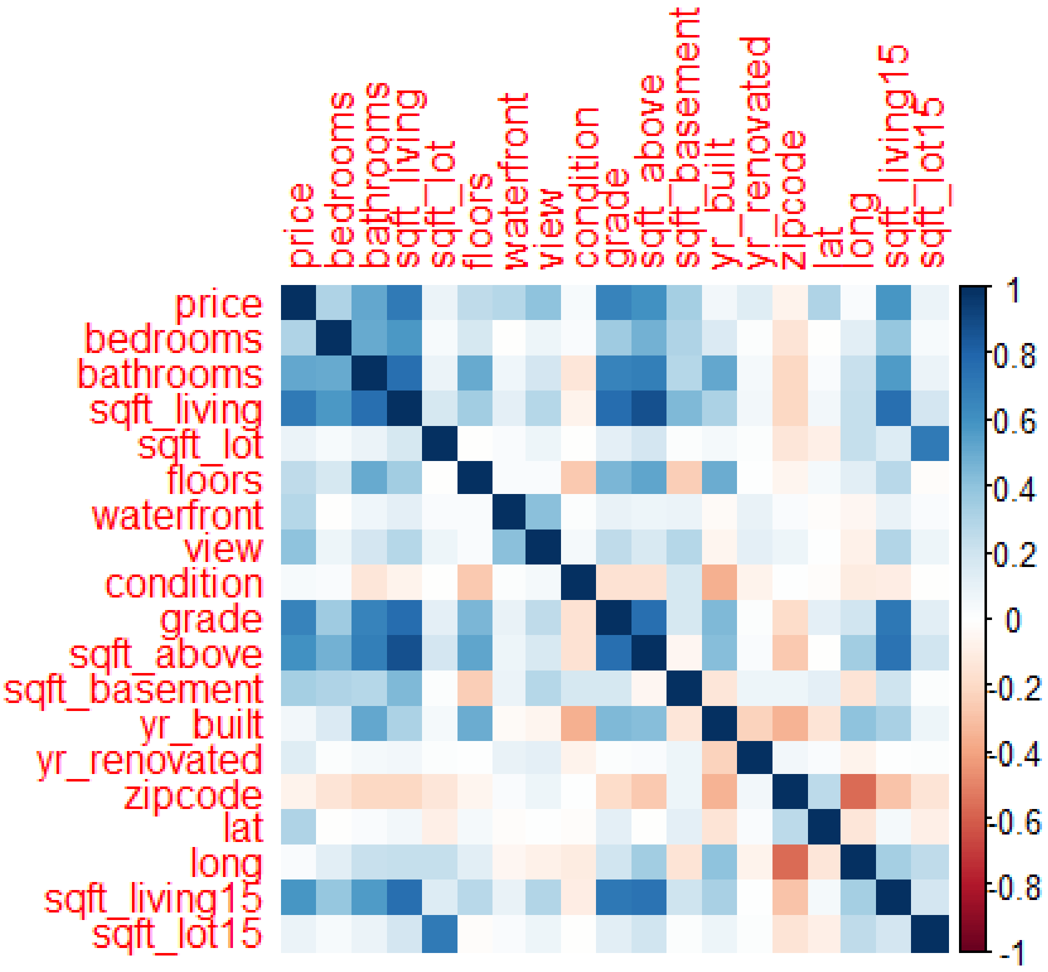
The following graphs illustrate that the price is skewed to the right with several very high prices.



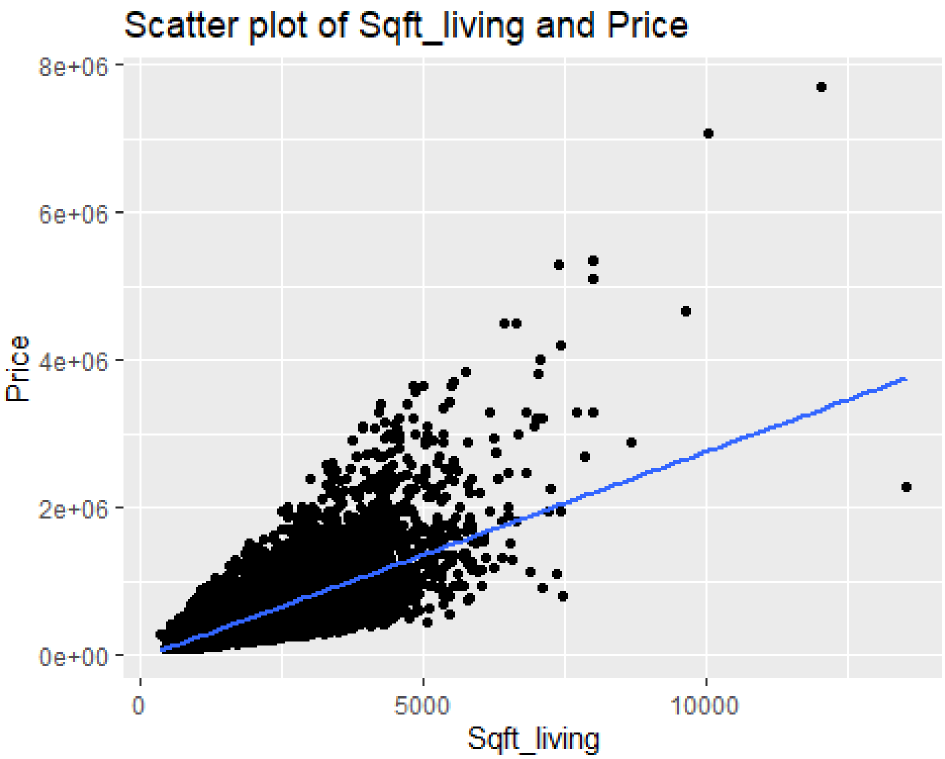


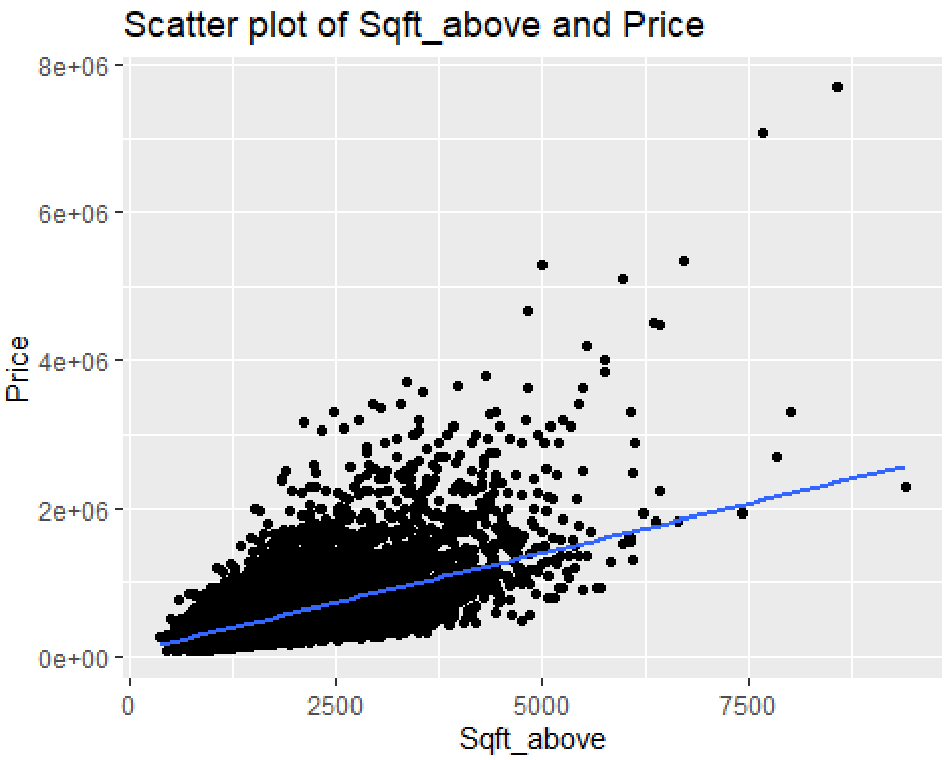
Determining the association between variables.

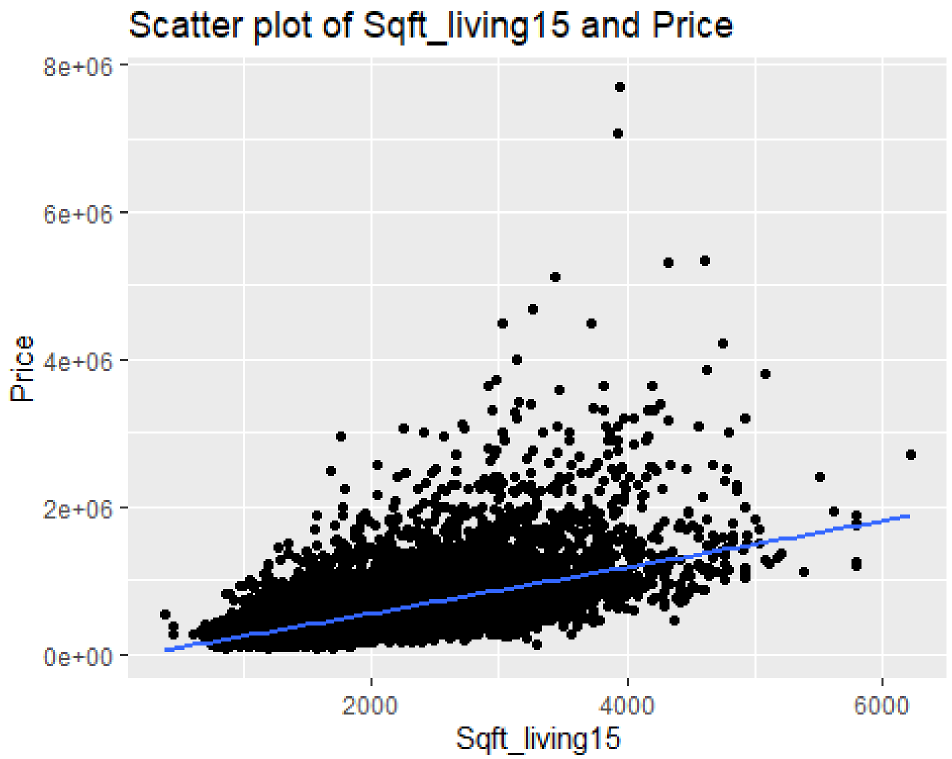
We take out the correlation plot (corrplot) to understand the association of the dependent variable (price) with the independent variables.



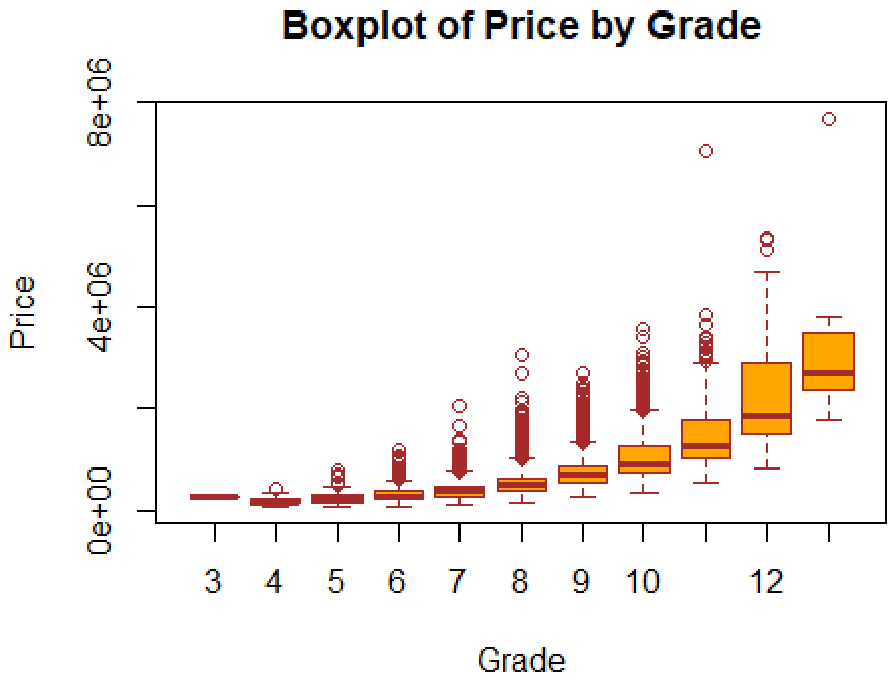
Price is strongly positively correlated with bathroom, Sqft_living, grade, sqft_above, sqft_living15. We will use scatterplot and boxplot to visualize the relationship between price and some predictors in the next pages.


Scatter plot of Sqft_living and Price
8e+06
6e+06
4e+06
2e+06
0e+00
Price
0
5000
10000
Sqft_living

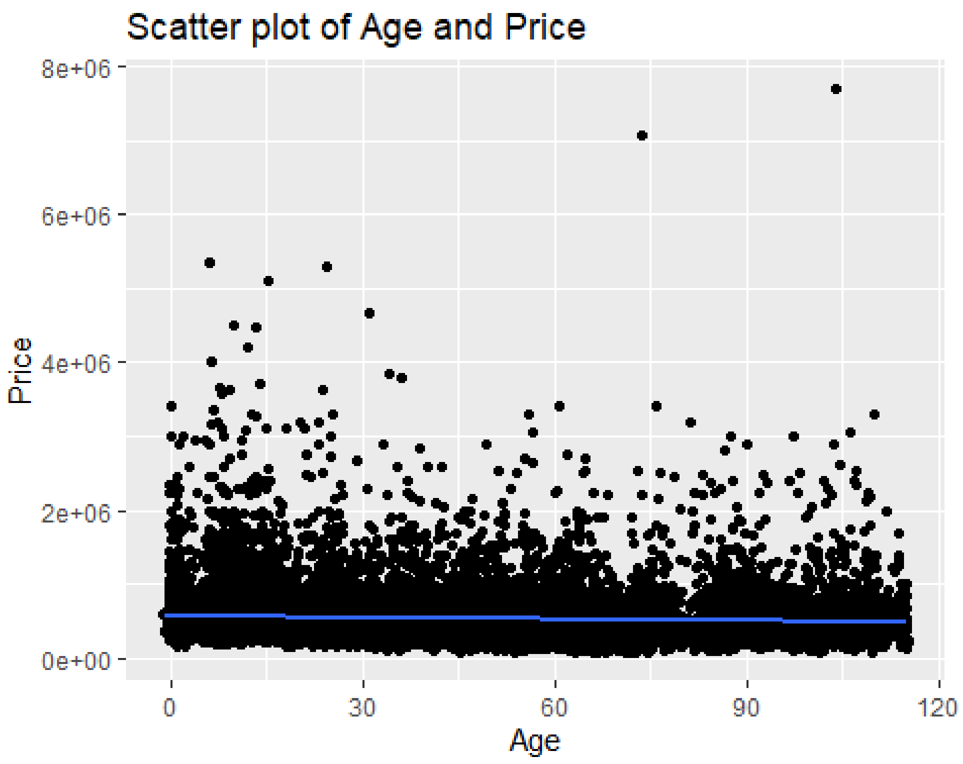

Scatter plot of Sqft_above and Price
8e+06
6e+06
4e+06
2e+06
0e+00
Price
0
2500
5000
7500
Sqft_above

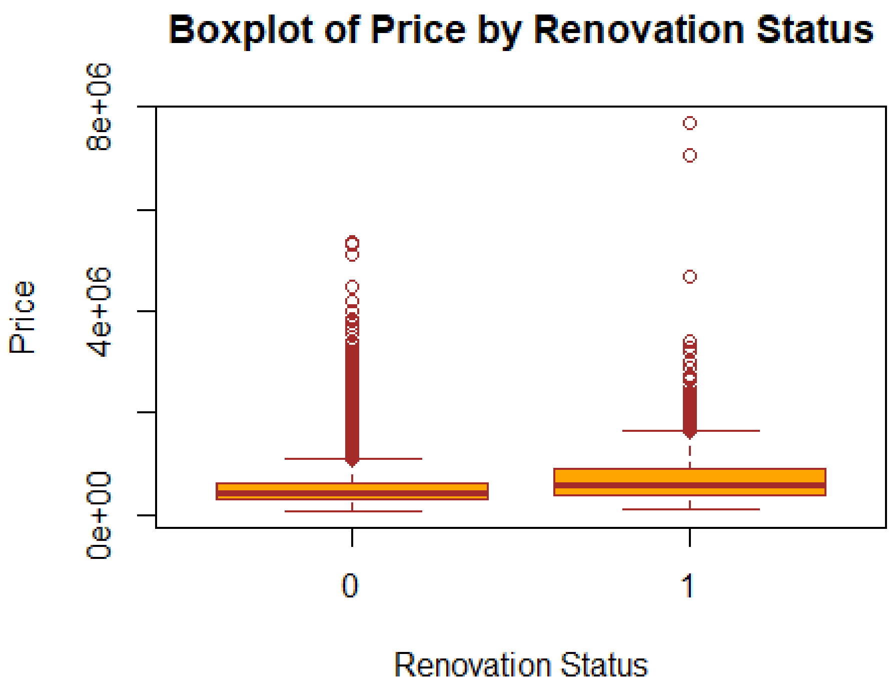


Scatter plot of Sqft_living15 and Price



Boxplot of Price by Grade


Scatter plot of Age and Price
8e+06
6e+06
4e+06
2e+06
0e+00
Price
0
30
60
90
120
Age

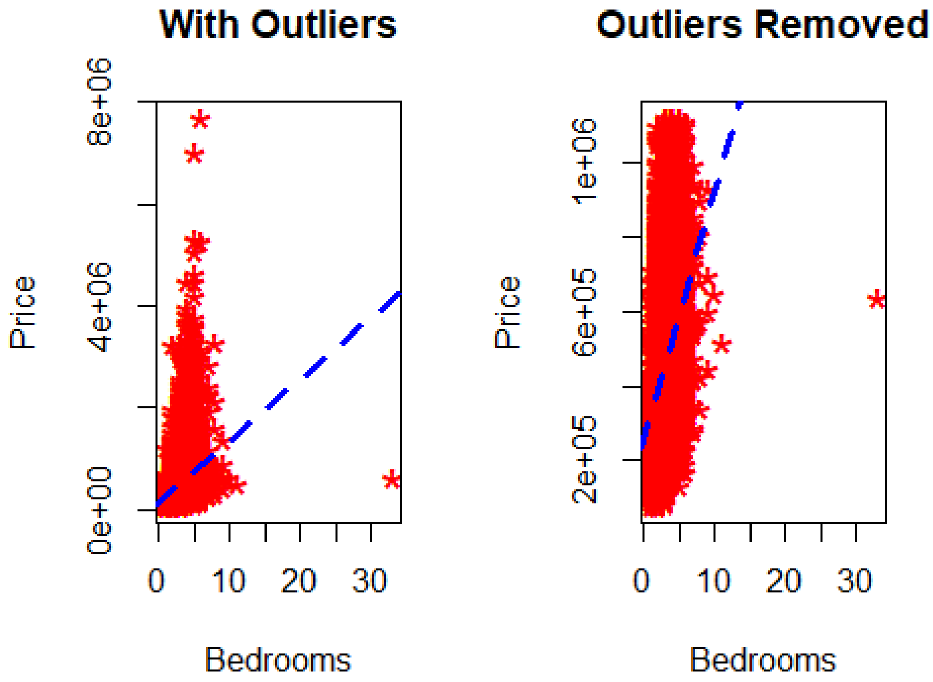

Boxplot of Price by Renovation Status
8e+06
4e+06
0e+00
Price
0
1
Renovation Status

Removing outliers could be optional

We see that we have a significantly large number of outliers.

Treating or altering the outlier/extreme values in genuine observations is not a standard operating procedure. However, it is essential to understand their impact on our predictive models.

To better understand the implications of outliers better, we should compare the fit of a simple linear regression model on the dataset with and without outliers. For this we first extract outliers from the data and then obtain the data without the outliers.



MODELING

We first use the entire data.

```r
# Prepare the training data by removing unnecessary columns
# and converting some columns to factors
train_data.m <- train_data[, -c(1, 2, 15, 16, 17, 22)] %>%
  mutate(waterfront = as.factor(waterfront),
         view = as.factor(view),
         condition = as.factor(condition),
         reno = as.factor(reno))

# Check the structure of the data
str(train_data.m)
```

```
## tibble [17,278 × 18] (S3: tbl_df/tbl/data.frame)
##  $ price        : num [1:17278] 221900 538000 180000 510000 1230000 ...
##  $ bedrooms     : num [1:17278] 3 3 2 3 4 3 3 3 3 2 ...
##  $ bathrooms    : num [1:17278] 1 2.25 1 2 4.5 2.25 1 2.5 2.5 1 ...
##  $ sqft_living  : num [1:17278] 1180 2570 770 1680 5420 ...
##  $ sqft_lot     : num [1:17278] 5650 7242 10000 8080 101930 ...
##  $ floors       : num [1:17278] 1 2 1 1 1 2 1 2 1 1 ...
##  $ waterfront   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ view         : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ condition    : Factor w/ 5 levels "1","2","3","4",..: 3 3 3 3 3 3 3 3 3
4 ...
##  $ grade        : num [1:17278] 7 7 6 8 11 7 7 7 8 7 ...
##  $ sqft_above   : num [1:17278] 1180 2170 770 1680 3890 ...
##  $ sqft_basement: num [1:17278] 0 400 0 0 1530 0 730 0 1700 300 ...
##  $ lat          : num [1:17278] 47.5 47.7 47.7 47.6 47.7 ...
##  $ long         : num [1:17278] -122 -122 -122 -122 -122 ...
##  $ sqft_living15: num [1:17278] 1340 1690 2720 1800 4760 ...
##  $ sqft_lot15   : num [1:17278] 5650 7639 8062 7503 101930 ...
##  $ age          : num [1:17278] 59 63 82 28 13 19 55 12 50 72 ...
##  $ reno         : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
```

```
# Check the number of columns
ncol(train_data.m)
```

```
## [1] 18
```

```
# Fit the model
model.full <- lm (formula = price ~ ., data = train_data.m)
# Display the model summary
summary(model.full)
```

```
##
## Call:
## lm(formula = price ~ ., data = train_data.m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1205541   -99237    -9262    76784  4393501
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.013e+07  1.711e+06 -23.447  < 2e-16 ***
## bedrooms      -3.262e+04  2.114e+03 -15.428  < 2e-16 ***
## bathrooms      3.989e+04  3.661e+03  10.896  < 2e-16 ***
## sqft_living    1.474e+02  4.916e+00  29.980  < 2e-16 ***
## sqft_lot       1.192e-01  5.223e-02   2.283   0.0224 *
## floors         2.982e+03  4.033e+03   0.739   0.4596
## waterfront1    5.268e+05  2.182e+04  24.143  < 2e-16 ***
## view1          1.158e+05  1.247e+04   9.286  < 2e-16 ***
## view2          6.193e+04  7.669e+03   8.075 7.22e-16 ***
## view3          1.261e+05  1.048e+04  12.035  < 2e-16 ***
## view4          2.918e+05  1.656e+04  17.619  < 2e-16 ***
## condition2    -1.272e+04  4.554e+04  -0.279   0.7799    
## condition3    -2.483e+04  4.238e+04  -0.586   0.5580    
## condition4     6.739e+03  4.238e+04   0.159   0.8737    
## condition5     4.148e+04  4.263e+04   0.973   0.3306    
## grade          9.994e+04  2.427e+03  41.177  < 2e-16 ***
## sqft_above     2.643e+01  4.911e+00   5.381 7.50e-08 ***
## sqft_basement         NA         NA      NA       NA    
## lat            5.518e+05  1.177e+04  46.888  < 2e-16 ***
## long          -1.077e+05  1.337e+04  -8.059 8.20e-16 ***
## sqft_living15  2.660e+01  3.898e+00   6.824 9.13e-12 ***
## sqft_lot15    -3.861e-01  8.005e-02  -4.824 1.42e-06 ***
## age            2.409e+03  8.185e+01  29.432  < 2e-16 ***
## reno1          5.513e+04  8.226e+03   6.702 2.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 202100 on 17255 degrees of freedom
## Multiple R-squared:  0.6967, Adjusted R-squared:  0.6963 
## F-statistic:  1802 on 22 and 17255 DF,  p-value: < 2.2e-16
```

```
# Get the model coefficients
models <- regsubsets(price ~ ., data = train_data.m, nvmax = 23)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in =
## force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
summary(models)
```

```
## Subset selection object
## Call: regsubsets.formula(price ~ ., data = train_data.m, nvmax = 23)
## 23 Variables  (and intercept)
##               Forced in Forced out
## bedrooms          FALSE      FALSE
## bathrooms         FALSE      FALSE
## sqft_living       FALSE      FALSE
## sqft_lot          FALSE      FALSE
## floors            FALSE      FALSE
## waterfront1       FALSE      FALSE
## view1             FALSE      FALSE
## view2             FALSE      FALSE
## view3             FALSE      FALSE
## view4             FALSE      FALSE
## condition2        FALSE      FALSE
## condition3        FALSE      FALSE
## condition4        FALSE      FALSE
## condition5        FALSE      FALSE
## grade             FALSE      FALSE
## sqft_above        FALSE      FALSE
## lat               FALSE      FALSE
## long              FALSE      FALSE
## sqft_living15     FALSE      FALSE
## sqft_lot15        FALSE      FALSE
## age               FALSE      FALSE
## reno1             FALSE      FALSE
## sqft_basement     FALSE      FALSE
## 1 subsets of each size up to 22
## Selection Algorithm: exhaustive
##           bedrooms bathrooms sqft_living sqft_lot floors waterfront1 view1
## 1  ( 1 )  " "      " "       "*"         " "      " "    " "         " "  
## 2  ( 1 )  " "      " "       "*"         " "      " "    " "         " "  
## 3  ( 1 )  " "      " "       "*"         " "      " "    "*"         " "  
## 4  ( 1 )  " "      " "       "*"         " "      " "    " "         " "  
## 5  ( 1 )  " "      " "       "*"         " "      " "    "*"         " "  
## 6  ( 1 )  " "      " "       "*"         " "      " "    "*"         " "  
## 7  ( 1 )  "*"      " "       "*"         " "      " "    "*"         " "  
## 8  ( 1 )  "*"      "*"       "*"         " "      " "    "*"         " "  
## 9  ( 1 )  "*"      "*"       "*"         " "      " "    "*"         " "  
## 10  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"  
## 11  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"  
## 12  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"  
## 13  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"  
## 14  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"  
## 15  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"  
## 16  ( 1 ) "*"      "*"       " "         " "      " "    "*"         "*"  
## 17  ( 1 ) "*"      "*"       " "         " "      " "    "*"         "*"  
## 18  ( 1 ) "*"      "*"       " "         " "      " "    "*"         "*"  
## 19  ( 1 ) "*"      "*"       "*"         "*"      " "    "*"         "*"  
## 20  ( 1 ) "*"      "*"       "*"         "*"      " "    "*"         "*"  
## 21  ( 1 ) "*"      "*"       "*"         "*"      "*"    "*"         "*"  
## 22  ( 1 ) "*"      "*"       "*"         "*"      "*"    "*"         "*"  
##           view2 view3 view4 condition2 condition3 condition4 condition5
grade
## 1  ( 1 )  " "   " "   " "   " "        " "        " "        " "        "
"
## 2  ( 1 )  " "   " "   " "   " "        " "        " "        " "        "
"
## 3  ( 1 )  " "   " "   " "   " "        " "        " "        " "        "
"
## 4  ( 1 )  " "   " "   " "   " "        " "        " "        " "
"*"
## 5  ( 1 )  " "   " "   " "   " "        " "        " "        " "
"*"
## 6  ( 1 )  " "   " "   "*"   " "        " "        " "        " "
"*"
## 7  ( 1 )  " "   " "   "*"   " "        " "        " "        " "
"*"
```

```
## 8  ( 1 )  " "   " "   "*"   " "        " "        " "        " "
"*"
## 9  ( 1 )  " "   "*"   "*"   " "        " "        " "        " "
"*"
## 10  ( 1 ) " "   "*"   "*"   " "        " "        " "        " "
"*"
## 11  ( 1 ) "*"   "*"   "*"   " "        " "        " "        " "
"*"
## 12  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 13  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 14  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 15  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 16  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 17  ( 1 ) "*"   "*"   "*"   " "        " "        "*"        "*"
"*"
## 18  ( 1 ) "*"   "*"   "*"   " "        " "        "*"        "*"
"*"
## 19  ( 1 ) "*"   "*"   "*"   " "        " "        "*"        "*"
"*"
## 20  ( 1 ) "*"   "*"   "*"   "*"        "*"        " "        "*"
"*"
## 21  ( 1 ) "*"   "*"   "*"   "*"        "*"        " "        "*"
"*"
## 22  ( 1 ) "*"   "*"   "*"   "*"        "*"        "*"        "*"
"*"
##           sqft_above sqft_basement lat long sqft_living15 sqft_lot15 age
reno1
## 1  ( 1 )  " "        " "           " " " "  " "           " "        " " "
"
## 2  ( 1 )  " "        " "           "*" " "  " "           " "        " " "
"
## 3  ( 1 )  " "        " "           "*" " "  " "           " "        " " "
"
## 4  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
## 5  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
## 6  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
## 7  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
## 8  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
## 9  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
```

```
## 10  ( 1 ) " "        " "           "*" " "  " "           " "        "*" "
"
## 11  ( 1 ) " "        " "           "*" " "  " "           " "        "*" "
"
## 12  ( 1 ) " "        " "           "*" " "  " "           " "        "*" "
"
## 13  ( 1 ) " "        " "           "*" "*"  " "           " "        "*" "
"
## 14  ( 1 ) "*"        " "           "*" "*"  " "           " "        "*" "
"
## 15  ( 1 ) " "        " "           "*" "*"  "*"           " "        "*"
"*"
## 16  ( 1 ) "*"        "*"           "*" "*"  "*"           " "        "*"
"*"
## 17  ( 1 ) "*"        "*"           "*" "*"  "*"           " "        "*"
"*"
## 18  ( 1 ) "*"        "*"           "*" "*"  "*"           "*"        "*"
"*"
## 19  ( 1 ) " "        "*"           "*" "*"  "*"           "*"        "*"
"*"
## 20  ( 1 ) "*"        " "           "*" "*"  "*"           "*"        "*"
"*"
## 21  ( 1 ) "*"        " "           "*" "*"  "*"           "*"        "*"
"*"
## 22  ( 1 ) "*"        " "           "*" "*"  "*"           "*"        "*"
"*"
```

```r
# Get the best model based on Adjusted R-squared, Cp, and BIC
res.sum <- summary(models)
data.frame(
  Adj.R2 = which.max(res.sum$adjr2),
  CP = which.min(res.sum$cp),
  BIC = which.min(res.sum$bic)
)
```

```
##   Adj.R2 CP BIC
## 1     19 19  18
```

```r
# id: model id
# object: regsubsets object
# data: data used to fit regsubsets
# outcome: outcome variable
get_model_formula <- function(id, object, outcome) {
  # get models data
  models <- summary(object)$which[id, -1]
  # Get model predictors
  predictors <- names(which(models == TRUE))
  predictors <- paste(predictors, collapse = "+")
  # Build model formula
  as.formula(paste0(outcome, "~", predictors))
}

get_model_formula(21, models, "Price")

## Price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors +
##     waterfront1 + view1 + view2 + view3 + view4 + condition2 +
##     condition3 + condition5 + grade + sqft_above + lat + long +
##     sqft_living15 + sqft_lot15 + age + reno1
## <environment: 0x000001915b045e48>

# # Fit the linear regression model 1 with selected predictors
model1 <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors +
            waterfront + view + condition + grade + sqft_basement + lat +
            long + sqft_living15 + sqft_lot15 + age + reno,
            data = train_data.m)
summary(model1)

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     lat + long + sqft_living15 + sqft_lot15 + age + reno, data =
train_data.m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1205541   -99237    -9262    76784  4393501
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.013e+07  1.711e+06 -23.447  < 2e-16 ***
## bedrooms      -3.262e+04  2.114e+03 -15.428  < 2e-16 ***
## bathrooms      3.989e+04  3.661e+03  10.896  < 2e-16 ***
## sqft_living    1.738e+02  4.137e+00  42.019  < 2e-16 ***
## sqft_lot       1.192e-01  5.223e-02   2.283   0.0224 *
## floors         2.982e+03  4.033e+03   0.739   0.4596
## waterfront1    5.268e+05  2.182e+04  24.143  < 2e-16 ***
## view1          1.158e+05  1.247e+04   9.286  < 2e-16 ***
## view2          6.193e+04  7.669e+03   8.075 7.22e-16 ***
## view3          1.261e+05  1.048e+04  12.035  < 2e-16 ***
## view4          2.918e+05  1.656e+04  17.619  < 2e-16 ***
## condition2    -1.272e+04  4.554e+04  -0.279   0.7799
## condition3    -2.483e+04  4.238e+04  -0.586   0.5580
## condition4     6.739e+03  4.238e+04   0.159   0.8737
## condition5     4.148e+04  4.263e+04   0.973   0.3306
## grade          9.994e+04  2.427e+03  41.177  < 2e-16 ***
## sqft_basement -2.643e+01  4.911e+00  -5.381 7.50e-08 ***
## lat            5.518e+05  1.177e+04  46.888  < 2e-16 ***
## long          -1.077e+05  1.337e+04  -8.059 8.20e-16 ***
## sqft_living15  2.660e+01  3.898e+00   6.824 9.13e-12 ***
## sqft_lot15    -3.861e-01  8.005e-02  -4.824 1.42e-06 ***
## age            2.409e+03  8.185e+01  29.432  < 2e-16 ***
## reno1          5.513e+04  8.226e+03   6.702 2.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202100 on 17255 degrees of freedom
## Multiple R-squared:  0.6967, Adjusted R-squared:  0.6963
## F-statistic:  1802 on 22 and 17255 DF,  p-value: < 2.2e-16
```

```
# Fit the linear regression model 2 with selected predictors
model2 <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors +
             waterfront + view + condition + grade + sqft_basement + lat +
             long + sqft_living15 + sqft_lot15 + age + reno +
             bathrooms * grade + grade * sqft_living15 + grade * sqft_lot15 +
             lat * long,
             data = train_data.m)
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     lat + long + sqft_living15 + sqft_lot15 + age + reno + bathrooms *
##     grade + grade * sqft_living15 + grade * sqft_lot15 + lat *
##     long, data = train_data.m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1891128   -91109    -7415    73422  3912013
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.985e+09  4.713e+08  12.700  < 2e-16 ***
## bedrooms              -1.393e+04  2.041e+03  -6.825 9.08e-12 ***
## bathrooms             -4.107e+05  1.346e+04 -30.505  < 2e-16 ***
## sqft_living            1.236e+02  4.106e+00  30.092  < 2e-16 ***
## sqft_lot               7.988e-02  4.940e-02   1.617 0.105924
## floors                 1.466e+04  3.858e+03   3.800 0.000145 ***
## waterfront1            5.409e+05  2.061e+04  26.242  < 2e-16 ***
## view1                  1.115e+05  1.177e+04   9.475  < 2e-16 ***
## view2                  5.465e+04  7.238e+03   7.549 4.59e-14 ***
## view3                  1.105e+05  9.890e+03  11.175  < 2e-16 ***
## view4                  2.417e+05  1.565e+04  15.440  < 2e-16 ***
## condition2             1.975e+04  4.295e+04   0.460 0.645615
## condition3             4.129e+04  4.000e+04   1.032 0.301965
## condition4             8.319e+04  4.001e+04   2.080 0.037582 *
## condition5             1.205e+05  4.025e+04   2.994 0.002762 **
## grade                 -3.041e+04  4.151e+03  -7.326 2.48e-13 ***
## sqft_basement          5.339e-01  4.708e+00   0.113 0.909713
## lat                    -1.261e+08  9.913e+06 -12.725  < 2e-16 ***
## long                    4.919e+07  3.856e+06  12.757  < 2e-16 ***
## sqft_living15           1.727e+01  1.600e+01   1.080 0.280371    
## sqft_lot15              2.157e+00  2.938e-01   7.342 2.20e-13 ***
## age                     1.743e+03  7.886e+01  22.101  < 2e-16 ***
## reno1                   7.797e+04  7.780e+03  10.023  < 2e-16 ***
## bathrooms:grade         5.804e+04  1.655e+03  35.063  < 2e-16 ***
## grade:sqft_living15     2.633e+00  1.870e+00   1.408 0.159251    
## grade:sqft_lot15       -3.024e-01  3.458e-02  -8.747  < 2e-16 ***
## lat:long               -1.037e+06  8.112e+04 -12.782  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 190500 on 17251 degrees of freedom
## Multiple R-squared:  0.7307, Adjusted R-squared:  0.7303 
## F-statistic:  1800 on 26 and 17251 DF,  p-value: < 2.2e-16
```

PREDICTION ON THE TEST DATA

```
# Preparing the test data
test_data.m <- test_data[, -c(1, 2, 15, 16, 17, 22)] %>%
  mutate(waterfront = as.factor(waterfront),
         view = as.factor(view),
         condition = as.factor(condition),
         reno = as.factor(reno))

# Predict the house prices on the test data using model2
pred_test <- predict(newdata = test_data.m, model2)

# Create a data frame to compare actual and predicted prices
tally_table_1 <- data.frame(actual = test_data.m$price, predicted =
pred_test)

mean(abs(test_data.m$price - pred_test))
```

```
## [1] 118663.6
```

Compare with a one-layer forward neural network

```
# Create the model matrix for the predictors and scale the data
x <- model.matrix(price ~ . - 1, data = train_data.m) %>% scale()
y <- train_data.m$price
```

The neural network has a similar test error to the multiple linear regression model. However, with additional time, we could tune the parameters to improve the performance of the neural network.