Comparison of Regression Models for Indian House Prices

Supreeya Srasom

1. Load libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```
library(caret)
```

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

```
library(readxl)
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(ggplot2)
library(knitr)
```

2. Import data

```
house_price <- read_excel("House Price India.xlsx")
```

```r
glimpse(house_price)
```

```
## Rows: 17,594
## 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~
```

3. Clean the dataset

```r
house_price %>%
  complete.cases() %>%
  mean()
```

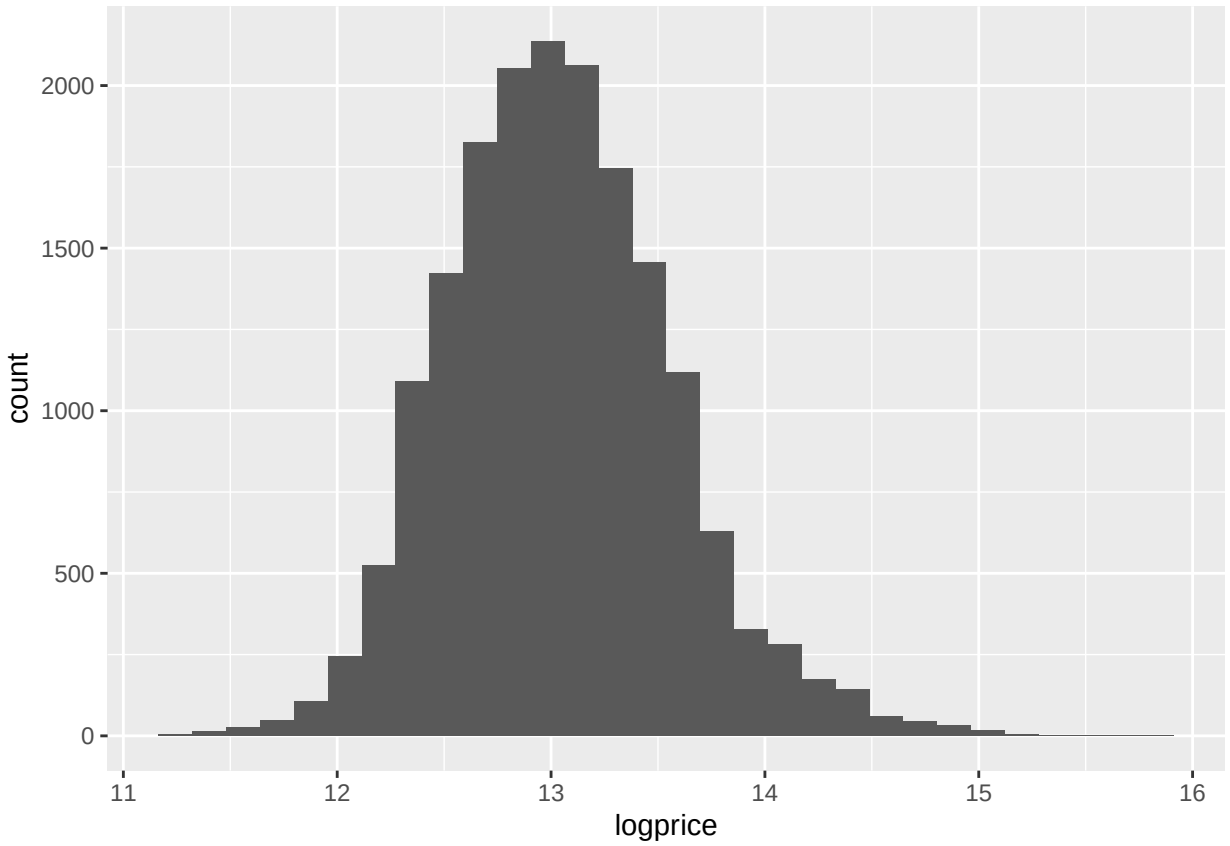
```
## [1] 1
```

4. Create new ID, changing graph by log

```r
n <- nrow(house_price)
new_house <- house_price %>%
  mutate(
    nid = 1:n,
    logprice = log(Price))

ggplot(new_house,aes(logprice))+geom_histogram()
```

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



5. Split data (80% Train / 20% Test)

```
split_data <- function(df) {
  set.seed(15)
  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(new_house)
train_df <- prep_data[[1]]
test_df <- prep_data[[2]]
```

6. Train control

```
set.seed(25)
ctrl <- trainControl(method = "cv",
                     number = 5,
                     verboseIter = TRUE)

model <- train(logprice ~ .,
               data = train_df,
```

```
               method = "lm",
               preProcess = c("center","scale"),
               trControl = ctrl)
```

```
## + Fold1: intercept=TRUE
## - Fold1: intercept=TRUE
## + Fold2: intercept=TRUE
## - Fold2: intercept=TRUE
## + Fold3: intercept=TRUE
## - Fold3: intercept=TRUE
## + Fold4: intercept=TRUE
## - Fold4: intercept=TRUE
## + Fold5: intercept=TRUE
## - Fold5: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```
summary(model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38911 -0.03016  0.00961  0.04286  0.15272
##
## Coefficients: (1 not defined because of singularities)
##                                                 Estimate Std. Error   t value
## (Intercept)                                   13.0406469  0.0006179 21105.641
## id                                            -0.3544969  0.0012932  -274.117
## Date                                          -0.0006009  0.0055921    -0.107
## `\\`number of bedrooms\\``                     0.0042235  0.0007917     5.335
## `\\`number of bathrooms\\``                    0.0067738  0.0011331     5.978
## `\\`living area\\``                           -0.0021877  0.0018553    -1.179
## `\\`lot area\\``                               0.0015406  0.0008844     1.742
## `\\`number of floors\\``                       0.0018387  0.0008855     2.076
## `\\`waterfront present\\``                     0.0004799  0.0006937     0.692
## `\\`number of views\\``                        0.0033006  0.0007476     4.415
## `\\`condition of the house\\``                 0.0063929  0.0006991     9.145
## `\\`grade of the house\\``                     0.0115080  0.0012412     9.272
## `\\`Area of the house(excluding basement)\\`` -0.0062191  0.0016433    -3.784
## `\\`Area of the basement\\``                          NA         NA        NA
## `\\`Built Year\\``                             0.0076295  0.0009929     7.684
## `\\`Renovation Year\\``                        0.0027934  0.0006660     4.194
## `\\`Postal Code\\``                            0.0006794  0.0006815     0.997
## Lattitude                                      0.0095201  0.0008181    11.637
## Longitude                                     -0.0026218  0.0007655    -3.425
## living_area_renov                              0.0032306  0.0010827     2.984
## lot_area_renov                                -0.0011073  0.0008899    -1.244
## `\\`Number of schools nearby\\``              -0.0006203  0.0006183    -1.003
## `\\`Distance from the airport\\``             -0.0003040  0.0006184    -0.492
## Price                                          0.1810544  0.0012111   149.498
## nid                                            0.0024411  0.0055928     0.436
##                                                Pr(>|t|)
```

```
## (Intercept)                                      < 2e-16 ***
## id                                               < 2e-16 ***
## Date                                            0.914425
## `\\`number of bedrooms\\``                      9.71e-08 ***
## `\\`number of bathrooms\\``                     2.31e-09 ***
## `\\`living area\\``                             0.238332
## `\\`lot area\\``                                0.081537 .
## `\\`number of floors\\``                        0.037878 *
## `\\`waterfront present\\``                      0.489095
## `\\`number of views\\``                         1.02e-05 ***
## `\\`condition of the house\\``                   < 2e-16 ***
## `\\`grade of the house\\``                       < 2e-16 ***
## `\\`Area of the house(excluding basement)\\`` 0.000155 ***
## `\\`Area of the basement\\``                          NA
## `\\`Built Year\\``                              1.64e-14 ***
## `\\`Renovation Year\\``                         2.76e-05 ***
## `\\`Postal Code\\``                             0.318836
## Lattitude                                        < 2e-16 ***
## Longitude                                       0.000616 ***
## living_area_renov                               0.002851 **
## lot_area_renov                                  0.213420
## `\\`Number of schools nearby\\``                0.315782
## `\\`Distance from the airport\\``               0.623031
## Price                                            < 2e-16 ***
## nid                                             0.662505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0733 on 14051 degrees of freedom
## Multiple R-squared:  0.9809, Adjusted R-squared:  0.9808
## F-statistic: 3.134e+04 on 23 and 14051 DF,  p-value: < 2.2e-16
```

7. Train model

```
set.seed(42)
lm_model <- train(logprice ~ .
             -( `grade of the house`+`Area of the house(excluding basement)`+ `Area of the basement`),
                  data = train_df,
                  method = "lm",
                  preProcess = c("center","scale"),
                  trControl = ctrl)
```

```
## + Fold1: intercept=TRUE
## - Fold1: intercept=TRUE
## + Fold2: intercept=TRUE
## - Fold2: intercept=TRUE
## + Fold3: intercept=TRUE
## - Fold3: intercept=TRUE
## + Fold4: intercept=TRUE
## - Fold4: intercept=TRUE
## + Fold5: intercept=TRUE
## - Fold5: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```
summary(lm_model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41322 -0.03034  0.00948  0.04331  0.14121
##
## Coefficients:
##                                    Estimate Std. Error   t value Pr(>|t|)
## (Intercept)                       1.304e+01  6.199e-04 21037.902  < 2e-16 ***
## id                               -3.575e-01  1.259e-03  -283.914  < 2e-16 ***
## Date                             -8.928e-04  5.610e-03    -0.159  0.87355
## `\\`number of bedrooms\\``        3.546e-03  7.906e-04     4.485 7.35e-06 ***
## `\\`number of bathrooms\\``       7.568e-03  1.123e-03     6.738 1.67e-11 ***
## `\\`living area\\``              -3.712e-03  1.422e-03    -2.610  0.00908 **
## `\\`lot area\\``                  1.472e-03  8.868e-04     1.659  0.09704 .
## `\\`number of floors\\``          1.185e-03  7.880e-04     1.504  0.13249
## `\\`waterfront present\\``       -3.979e-05  6.938e-04    -0.057  0.95427
## `\\`number of views\\``           3.834e-03  7.365e-04     5.206 1.96e-07 ***
## `\\`condition of the house\\``    6.330e-03  6.985e-04     9.062  < 2e-16 ***
## `\\`Built Year\\``                1.027e-02  9.509e-04    10.799  < 2e-16 ***
## `\\`Renovation Year\\``           2.874e-03  6.681e-04     4.302 1.70e-05 ***
## `\\`Postal Code\\``               5.270e-04  6.834e-04     0.771  0.44064
## Lattitude                         9.062e-03  8.137e-04    11.137  < 2e-16 ***
## Longitude                        -3.972e-03  7.472e-04    -5.316 1.07e-07 ***
## living_area_renov                 4.900e-03  1.039e-03     4.714 2.45e-06 ***
## lot_area_renov                   -1.227e-03  8.927e-04    -1.374  0.16933
## `\\`Number of schools nearby\\``  -6.404e-04  6.203e-04    -1.032  0.30195
## `\\`Distance from the airport\\`` -2.551e-04  6.203e-04    -0.411  0.68088
## Price                             1.827e-01  1.190e-03   153.468  < 2e-16 ***
## nid                               2.616e-03  5.610e-03     0.466  0.64105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07354 on 14053 degrees of freedom
## Multiple R-squared:  0.9808, Adjusted R-squared:  0.9807
## F-statistic: 3.41e+04 on 21 and 14053 DF,  p-value: < 2.2e-16
```

```
lm_rsquared_train <- round(summary(lm_model)$r.squared,4)
cat(paste("Linear Regression Train Rsquared: ",lm_rsquared_train))
```

```
## Linear Regression Train Rsquared:  0.9808
```

```
glm_model <- train(logprice ~ .
             -( `grade of the house`+`Area of the house(excluding basement)`+ `Area of the basement`),
                   data = train_df,
                   method = "glmnet",
                   metric = "Rsquared",
                   preProcess = c("center","scale"),
                   trControl = ctrl)
```

```
## + Fold1: alpha=0.10, lambda=0.1022
```

```
## - Fold1: alpha=0.10, lambda=0.1022
## + Fold1: alpha=0.55, lambda=0.1022
## - Fold1: alpha=0.55, lambda=0.1022
## + Fold1: alpha=1.00, lambda=0.1022
## - Fold1: alpha=1.00, lambda=0.1022
## + Fold2: alpha=0.10, lambda=0.1022
## - Fold2: alpha=0.10, lambda=0.1022
## + Fold2: alpha=0.55, lambda=0.1022
## - Fold2: alpha=0.55, lambda=0.1022
## + Fold2: alpha=1.00, lambda=0.1022
## - Fold2: alpha=1.00, lambda=0.1022
## + Fold3: alpha=0.10, lambda=0.1022
## - Fold3: alpha=0.10, lambda=0.1022
## + Fold3: alpha=0.55, lambda=0.1022
## - Fold3: alpha=0.55, lambda=0.1022
## + Fold3: alpha=1.00, lambda=0.1022
## - Fold3: alpha=1.00, lambda=0.1022
## + Fold4: alpha=0.10, lambda=0.1022
## - Fold4: alpha=0.10, lambda=0.1022
## + Fold4: alpha=0.55, lambda=0.1022
## - Fold4: alpha=0.55, lambda=0.1022
## + Fold4: alpha=1.00, lambda=0.1022
## - Fold4: alpha=1.00, lambda=0.1022
## + Fold5: alpha=0.10, lambda=0.1022
## - Fold5: alpha=0.10, lambda=0.1022
## + Fold5: alpha=0.55, lambda=0.1022
## - Fold5: alpha=0.55, lambda=0.1022
## + Fold5: alpha=1.00, lambda=0.1022
## - Fold5: alpha=1.00, lambda=0.1022
## Aggregating results
## Selecting tuning parameters
## Fitting alpha = 0.55, lambda = 0.00102 on full training set
```

```
glm_model
```

```
## glmnet
##
## 14075 samples
##    24 predictor
##
## Pre-processing: centered (21), scaled (21)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11260, 11260, 11260, 11260, 11260
## Resampling results across tuning parameters:
##
##   alpha  lambda       RMSE        Rsquared   MAE
##   0.10   0.001021679  0.07423357  0.9804541  0.04385376
##   0.10   0.010216787  0.07486427  0.9802274  0.04249209
##   0.10   0.102167873  0.09994555  0.9726032  0.06063987
##   0.55   0.001021679  0.07420357  0.9804591  0.04432456
##   0.55   0.010216787  0.07516544  0.9801981  0.04229676
##   0.55   0.102167873  0.11171533  0.9795398  0.06146077
##   1.00   0.001021679  0.07420483  0.9804546  0.04460942
##   1.00   0.010216787  0.07573589  0.9800310  0.04284607
##   1.00   0.102167873  0.13177267  0.9789371  0.07188410
```

```
##
## Rsquared was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.55 and lambda = 0.001021679.
```

```
glm_rsquared_train <- round(c(head(glm_model$results$Rsquared,5),
                              tail(glm_model$results$Rsquared,5))[1],4)
cat(paste("Regularized Regression Train Rsquared: ",glm_rsquared_train))
```

```
## Regularized Regression Train Rsquared:  0.9805
```

8. Score model

```
# Predict
p_lm <- predict(lm_model, newdata=test_df)
p_glm <- predict(glm_model, newdata=test_df)
```

```
# Evaluate
error_lm <- test_df$logprice - p_lm
error_glm <- test_df$logprice - p_glm

ssr_lm <- sum(error_lm**2)
ssr_glm <- sum(error_glm**2)
sst <- sum((test_df$logprice - mean(test_df$logprice))**2)

lm_rsquared_test <- round(1 - ssr_lm/sst,4)
glm_rsquared_test<- round(1 - ssr_glm/sst,4)
```

9. Comparing model

```
result <- data.frame(
  c(lm_rsquared_train,glm_rsquared_train),
  c(lm_rsquared_test,glm_rsquared_test)
  )

colnames(result) <- c("Train Rsquared", "Test Rsquared")
row.names(result) <- c("Linear Regression", "Regularized Regression")

kable(result,caption = "Rsquared Comparison")
```

Table 1: Rsquared Comparison

	Train Rsquared	Test Rsquared
Linear Regression	0.9808	0.9835
Regularized Regression	0.9805	0.9835

The test results in the table show that both Linear Regression and Regularized Regression achieve the same R-squared value (98.35%), suggesting similar performance on this task.