Housing Prices Regression

Zhiwei Lin

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(caret)
```

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

```
library(moments)
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-6
```

Import data

```
train<- read_csv("train.csv")
```

```
## Rows: 1460 Columns: 81
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (43): MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConf...
## dbl (38): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
test <- read_csv("test.csv")
```

```
## Rows: 1459 Columns: 80
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (43): MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConf...
## dbl (37): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(train)
```

```
## # A tibble: 6 x 81
##      Id MSSubClass MSZoning LotFr~1 LotArea Street Alley LotSh~2 LandC~3 Utili~4
##   <dbl>      <dbl> <chr>      <dbl>   <dbl> <chr>  <chr> <chr>   <chr>   <chr>
## 1     1         60 RL            65    8450 Pave   <NA>  Reg     Lvl     AllPub
## 2     2         20 RL            80    9600 Pave   <NA>  Reg     Lvl     AllPub
## 3     3         60 RL            68   11250 Pave   <NA>  IR1     Lvl     AllPub
## 4     4         70 RL            60    9550 Pave   <NA>  IR1     Lvl     AllPub
## 5     5         60 RL            84   14260 Pave   <NA>  IR1     Lvl     AllPub
## 6     6         50 RL            85   14115 Pave   <NA>  IR1     Lvl     AllPub
## # ... with 71 more variables: LotConfig <chr>, LandSlope <chr>,
## #   Neighborhood <chr>, Condition1 <chr>, Condition2 <chr>, BldgType <chr>,
## #   HouseStyle <chr>, OverallQual <dbl>, OverallCond <dbl>, YearBuilt <dbl>,
## #   YearRemodAdd <dbl>, RoofStyle <chr>, RoofMatl <chr>, Exterior1st <chr>,
## #   Exterior2nd <chr>, MasVnrType <chr>, MasVnrArea <dbl>, ExterQual <chr>,
## #   ExterCond <chr>, Foundation <chr>, BsmtQual <chr>, BsmtCond <chr>,
## #   BsmtExposure <chr>, BsmtFinType1 <chr>, BsmtFinSF1 <dbl>, ...
```

```
test$Id <- NULL
train$Id <- NULL
test$SalePrice <- NA
```

Remove id variable in both data and add SalePrice variable to test_data

```
all <- rbind(train,test)
```

combine train and test data

```
missing_percentage <- function(df){
 colSums(is.na(df))/nrow(df)
}
missing_percentage(all)
```

```
##    MSSubClass      MSZoning   LotFrontage       LotArea        Street
##  0.0000000000  0.0013703323  0.1664953751  0.0000000000  0.0000000000
##         Alley      LotShape   LandContour     Utilities     LotConfig
##  0.9321685509  0.0000000000  0.0000000000  0.0006851662  0.0000000000
##     LandSlope  Neighborhood    Condition1    Condition2      BldgType
##  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
##    HouseStyle   OverallQual   OverallCond     YearBuilt  YearRemodAdd
##  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
##     RoofStyle      RoofMatl   Exterior1st   Exterior2nd    MasVnrType
##  0.0000000000  0.0000000000  0.0003425831  0.0003425831  0.0082219938
##    MasVnrArea     ExterQual     ExterCond    Foundation      BsmtQual
##  0.0078794108  0.0000000000  0.0000000000  0.0000000000  0.0277492292
##      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1  BsmtFinType2
##  0.0280918123  0.0280918123  0.0270640630  0.0003425831  0.0274066461
##    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating     HeatingQC
##  0.0003425831  0.0003425831  0.0003425831  0.0000000000  0.0000000000
##    CentralAir    Electrical      1stFlrSF      2ndFlrSF  LowQualFinSF
##  0.0000000000  0.0003425831  0.0000000000  0.0000000000  0.0000000000
##     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath      HalfBath
##  0.0000000000  0.0006851662  0.0006851662  0.0000000000  0.0000000000
##  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd    Functional
##  0.0000000000  0.0000000000  0.0003425831  0.0000000000  0.0006851662
##    Fireplaces   FireplaceQu    GarageType   GarageYrBlt  GarageFinish
##  0.0000000000  0.4864679685  0.0537855430  0.0544707091  0.0544707091
##    GarageCars    GarageArea    GarageQual    GarageCond    PavedDrive
##  0.0003425831  0.0003425831  0.0544707091  0.0544707091  0.0000000000
##    WoodDeckSF   OpenPorchSF EnclosedPorch     3SsnPorch   ScreenPorch
##  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
##      PoolArea        PoolQC         Fence   MiscFeature       MiscVal
##  0.0000000000  0.9965741692  0.8043850634  0.9640287770  0.0000000000
##        MoSold        YrSold      SaleType SaleCondition     SalePrice
##  0.0000000000  0.0000000000  0.0003425831  0.0000000000  0.4998287085
```

```r
all <- mutate_if(all,is.character,as.factor)
```

```r
all <- all %>% mutate_if(is.factor, ~ ifelse(is.na(.), 0, .))
# replace missing values NA with 0 for all categorical variables
all <- all %>% mutate_if(is.numeric, ~ ifelse(is.na(.), mean(., na.rm = TRUE), .))
# replcae missing values NA with mean for all numeric variables
sum(is.na(all))
```

```
## [1] 0
```

```r
# no missing values in the data anymore
```

ridge regression

```r
# Split the data into training and test sets
train_data <- all[1:nrow(train),]
test_data <- all[(nrow(train)+1):nrow(all),]
```

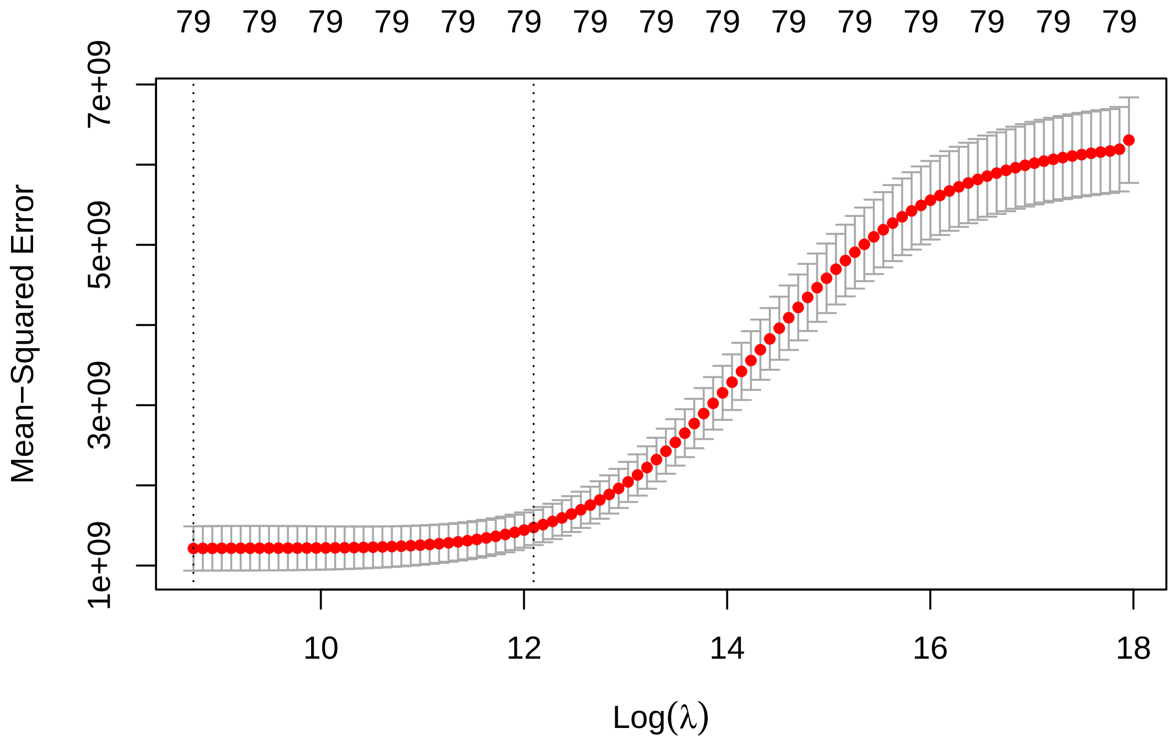
```
library(glmnet)
set.seed(123)

# Train the model
x <- model.matrix(SalePrice ~ ., data = train_data)
y <- train_data$SalePrice
model_ridge <- glmnet(x, y, alpha = 0, lambda = 1)
summary(model_ridge)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      80     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       5     -none-    call
## nobs       1     -none-    numeric
```

```
cv_model <- cv.glmnet(x, y, alpha = 0)
best_lambda <- cv_model$lambda.min
best_lambda
```

```
## [1] 6281.603
```

```
plot(cv_model)
```



```
best_model <- glmnet(x, y, alpha = 0, lambda = best_lambda)
coef(best_model)
```

```
## 81 x 1 sparse Matrix of class "dgCMatrix"
##                            s0
## (Intercept)      1.588595e+06
## (Intercept)      .
## MSSubClass      -7.580948e+01
## MSZoning        -1.855773e+03
## LotFrontage     -9.808454e+01
## LotArea          3.739227e-01
## Street           3.040306e+04
## Alley           -2.807203e+03
## LotShape        -9.781491e+02
## LandContour      2.072183e+03
## Utilities       -4.202446e+04
## LotConfig       -7.700931e+01
## LandSlope        4.856669e+03
## Neighborhood     2.658600e+02
## Condition1      -5.693669e+02
## Condition2      -8.215755e+03
## BldgType        -2.331943e+03
## HouseStyle      -6.530477e+02
## OverallQual      1.004865e+04
## OverallCond      4.434593e+03
## YearBuilt      1.341136e+02
## YearRemodAdd   7.784967e+01
## RoofStyle      1.852196e+03
## RoofMatl       4.716268e+03
## Exterior1st   -6.881609e+02
## Exterior2nd    2.518150e+02
## MasVnrType     3.619757e+03
## MasVnrArea     3.004510e+01
## ExterQual     -9.823521e+03
## ExterCond      6.563518e+02
## Foundation     8.333589e+02
## BsmtQual      -6.800604e+03
## BsmtCond       2.841490e+03
## BsmtExposure  -2.717737e+03
## BsmtFinType1  -4.240115e+02
## BsmtFinSF1     9.731094e+00
## BsmtFinType2   1.842868e+03
## BsmtFinSF2     1.322509e+01
## BsmtUnfSF      5.560221e-01
## TotalBsmtSF    1.297241e+01
## Heating       -1.751328e+03
## HeatingQC     -7.890244e+02
## CentralAir     2.500883e+03
## Electrical    -4.657167e+02
## `1stFlrSF`     1.567057e+01
## `2ndFlrSF`     1.789557e+01
## LowQualFinSF  -1.521050e+01
## GrLivArea      2.078322e+01
## BsmtFullBath   6.361188e+03
## BsmtHalfBath   3.582740e+02
## FullBath       4.835354e+03
## HalfBath       1.824746e+03
## BedroomAbvGr  -2.699794e+03
## KitchenAbvGr  -1.634595e+04
## KitchenQual   -8.131974e+03
## TotRmsAbvGrd   3.675393e+03
## Functional     3.526636e+03
## Fireplaces     7.317767e+03
## FireplaceQu   -8.020173e+02
## GarageType     1.576947e+02
## GarageYrBlt   -2.684399e+01
## GarageFinish  -2.241061e+03
## GarageCars     9.120546e+03
## GarageArea     1.242116e+01
## GarageQual    -1.294795e+03
## GarageCond    -9.910798e+01
## PavedDrive     2.756969e+03
## WoodDeckSF     2.088454e+01
## OpenPorchSF   -4.704388e+00
## EnclosedPorch  3.632602e+00
## `3SsnPorch`    2.467738e+01
## ScreenPorch    4.671957e+01
## PoolArea       2.900141e+02
## PoolQC        -9.106570e+04
## Fence          2.907605e+02
## MiscFeature   -1.674022e+03
## MiscVal        2.675883e-01
## MoSold        -9.743064e+01
## YrSold        -9.651580e+02
## SaleType      -5.222981e+02
## SaleCondition  2.840561e+03
```

```
# Make predictions on the test data
x_test <- model.matrix(SalePrice ~ ., data = test_data)
y_pred <- predict(model_ridge, s=best_lambda, newx = x_test)
head(y_pred)
```

```
##         s1
## 1 113788.0
## 2 164188.7
## 3 169775.9
## 4 187647.6
## 5 189031.2
## 6 171278.4
```

Lasso regression

```
best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
coef(best_model)
```

```
## 81 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)   -1.899347e+05
## (Intercept)    .
## MSSubClass    -9.833075e+00
## MSZoning       .
## LotFrontage    .
## LotArea        1.167164e-01
## Street         .
## Alley          .
## LotShape       .
## LandContour    .
## Utilities      .
## LotConfig      .
## LandSlope      .
## Neighborhood   .
## Condition1     .
## Condition2     .
## BldgType       .
## HouseStyle     .
## OverallQual    1.700244e+04
## OverallCond    .
## YearBuilt      6.516485e+01
## YearRemodAdd   5.016657e+01
## RoofStyle      .
## RoofMatl       .
## Exterior1st    .
## Exterior2nd    .
## MasVnrType     .
## MasVnrArea     8.627788e+00
## ExterQual     -8.348907e+03
## ExterCond      .
## Foundation     .
## BsmtQual      -2.282964e+03
## BsmtCond       .
## BsmtExposure   .
## BsmtFinType1   .
## BsmtFinSF1     1.048287e+01
## BsmtFinType2   .
## BsmtFinSF2     .
## BsmtUnfSF      .
## TotalBsmtSF    1.250509e+01
## Heating        .
## HeatingQC      .
## CentralAir     .
## Electrical     .
## `1stFlrSF`     2.662357e+00
## `2ndFlrSF`     .
## LowQualFinSF   .
## GrLivArea      3.797923e+01
## BsmtFullBath   .
## BsmtHalfBath   .
## FullBath       .
## HalfBath       .
## BedroomAbvGr   .
## KitchenAbvGr   .
## KitchenQual   -7.904662e+03
## TotRmsAbvGrd   .
## Functional     .
## Fireplaces     2.925052e+03
## FireplaceQu    .
## GarageType     .
## GarageYrBlt    .
## GarageFinish   .
## GarageCars     1.072251e+04
## GarageArea     2.214134e+00
## GarageQual     .
## GarageCond     .
## PavedDrive     .
## WoodDeckSF     4.822403e+00
## OpenPorchSF    .
## EnclosedPorch  .
## `3SsnPorch`    .
## ScreenPorch    .
## PoolArea       .
## PoolQC         .
## Fence          .
## MiscFeature    .
## MiscVal        .
## MoSold         .
## YrSold         .
## SaleType       .
## SaleCondition  .
```

```
#use lasso regression model to predict response value
head(predict(best_model, s = best_lambda, newx = x_test))
```

```
##         s1
## 1 113620.4
## 2 167673.8
## 3 165300.6
## 4 185439.0
## 5 212969.7
## 6 171229.4
```