

# Predicting House Prices

## Kaggle Competition

This Kaggle competition is about predicting house prices based on a set of around 80 predictor variables. Please read the brief description of the project and get familiar with the various predictors. We will have to do some initial cleaning to successfully work with these data. Overall, we (in teams) will use the provided training dataset to build a multiple linear regression model for predicting house prices. Once we have settled on a final model, we will use it with the predictors available in the testing dataset to predict house prices. The goal of the competition mentions that our predictions  $\hat{y}_i$  for the houses in the testing data are compared to the (withheld) true selling prices  $y_i^{\text{test}}$  via  $\sum_i (\log \hat{y}_i - \log y_i^{\text{test}})^2$ . Because selling prices are typically right-skewed, I think as a first step we will log-transform the selling prices of the houses in the training data to obtain a more bell-shaped distribution. However, although we will build a model for the log-prices, we will still have to submit the price of a house (and not the log-price) to Kaggle, together with the ID of the house.

## Loading and inspecting the train and test datasets

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.4      v dplyr  1.0.7
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   2.0.1      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidyr)
## Load Training Data
path_traindata <- 'https://raw.githubusercontent.com/bklingen/Price-Prediction/main/train.csv'
train <- read_csv(path_traindata)

## 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.
```

```
dim(train)
```

```
## [1] 1460 81
```

```
## Load Testing Data
```

```
path_testdata <- 'https://raw.githubusercontent.com/bklingen/Price-Prediction/main/test.csv'
test <- read_csv(path_testdata)
```

```
## 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.
```

```
dim(test)
```

```
## [1] 1459 80
```

This makes sense: We have one less column in test data because of the missing house prices.

But, are the column names the same? Let's find the "difference" between two sets: All the column names that are in the test data but not in the train data:

```
setdiff(colnames(test), colnames(train))
```

```
## character(0)
```

OK, good, and now the other way around:

```
setdiff(colnames(train), colnames(test))
```

```
## [1] "SalePrice"
```

OK, great. So no surprises there. All predictors that exist in the train data set also appear in the test dataset.

Let's see how many quantitative and how many categorical predictors we have in the training dataset, at least at face value:

```
train_quantPredictors = train %>% select(where(is.numeric)) %>% select(-SalePrice)
train_catPredictors = train %>% select(where(is.character))
dim(train_quantPredictors)
```

```
## [1] 1460 37
```

```
dim(train_catPredictors)
```

```
## [1] 1460 43
```

Let's transform the categorical predictors into factors, which should make it easier to combine categories, create a category like "other", etc.

```
train_catPredictors = train_catPredictors %>% transmute_all(as.factor)
```

First, let's see the category names and frequency for each variable:

```
for(i in 1:ncol(train_catPredictors)) {
  print(colnames(train_catPredictors)[i])
  print("----")
  print(as.data.frame(fct_count(unlist(train_catPredictors[,i]))))
  print("-----")
}
```

```
## [1] "MSZoning"
## [1] "----"
##      f      n
## 1 C (all)  10
## 2      FV   65
## 3      RH   16
## 4      RL 1151
## 5      RM  218
## [1] "-----"
## [1] "Street"
## [1] "----"
##      f      n
## 1 Grvl    6
## 2 Pave 1454
## [1] "-----"
## [1] "Alley"
## [1] "----"
##      f      n
## 1 Grvl    50
## 2 Pave    41
## 3 <NA> 1369
## [1] "-----"
## [1] "LotShape"
## [1] "----"
##      f      n
## 1 IR1  484
## 2 IR2   41
```

```

## 3 IR3 10
## 4 Reg 925
## [1] "-----"
## [1] "LandContour"
## [1] "----"
##      f      n
## 1 Bnk 63
## 2 HLS 50
## 3 Low 36
## 4 Lvl 1311
## [1] "-----"
## [1] "Utilities"
## [1] "----"
##      f      n
## 1 AllPub 1459
## 2 NoSeWa 1
## [1] "-----"
## [1] "LotConfig"
## [1] "----"
##      f      n
## 1 Corner 263
## 2 CulDSac 94
## 3 FR2 47
## 4 FR3 4
## 5 Inside 1052
## [1] "-----"
## [1] "LandSlope"
## [1] "----"
##      f      n
## 1 Gtl 1382
## 2 Mod 65
## 3 Sev 13
## [1] "-----"
## [1] "Neighborhood"
## [1] "----"
##      f      n
## 1 Blmngtn 17
## 2 Blueste 2
## 3 BrDale 16
## 4 BrkSide 58
## 5 ClearCr 28
## 6 CollgCr 150
## 7 Crawfor 51
## 8 Edwards 100
## 9 Gilbert 79
## 10 IDOTRR 37
## 11 MeadowV 17
## 12 Mitchel 49
## 13 NAmes 225
## 14 NoRidge 41
## 15 NPKvill 9
## 16 NridgHt 77
## 17 NWAmes 73
## 18 OldTown 113

```

```

## 19 Sawyer 74
## 20 SawyerW 59
## 21 Somerst 86
## 22 StoneBr 25
## 23 SWISU 25
## 24 Timber 38
## 25 Veenker 11
## [1] "-----"
## [1] "Condition1"
## [1] "----"
##      f      n
## 1 Artery 48
## 2 Feedr 81
## 3 Norm 1260
## 4 PosA 8
## 5 PosN 19
## 6 RRAe 11
## 7 RRAn 26
## 8 RRNe 2
## 9 RRNn 5
## [1] "-----"
## [1] "Condition2"
## [1] "----"
##      f      n
## 1 Artery 2
## 2 Feedr 6
## 3 Norm 1445
## 4 PosA 1
## 5 PosN 2
## 6 RRAe 1
## 7 RRAn 1
## 8 RRNn 2
## [1] "-----"
## [1] "BldgType"
## [1] "----"
##      f      n
## 1 1Fam 1220
## 2 2fmCon 31
## 3 Duplex 52
## 4 Twnhs 43
## 5 TwnhsE 114
## [1] "-----"
## [1] "HouseStyle"
## [1] "----"
##      f      n
## 1 1.5Fin 154
## 2 1.5Unf 14
## 3 1Story 726
## 4 2.5Fin 8
## 5 2.5Unf 11
## 6 2Story 445
## 7 SFoyer 37
## 8 SLvl 65
## [1] "-----"

```

```

## [1] "RoofStyle"
## [1] "-----"
##      f      n
## 1   Flat   13
## 2   Gable 1141
## 3 Gambrel  11
## 4    Hip  286
## 5 Mansard   7
## 6   Shed   2
## [1] "-----"
## [1] "RoofMat1"
## [1] "-----"
##      f      n
## 1 ClyTile   1
## 2 CompShg 1434
## 3 Membran   1
## 4   Metal   1
## 5    Roll   1
## 6 Tar&Grv  11
## 7 WdShake   5
## 8 WdShngl   6
## [1] "-----"
## [1] "Exterior1st"
## [1] "-----"
##      f      n
## 1 AsbShng  20
## 2 AsphShn   1
## 3 BrkComm   2
## 4 BrkFace  50
## 5  CBlock   1
## 6 CemntBd  61
## 7 HdBoard 222
## 8 ImStucc   1
## 9 MetalSd 220
## 10 Plywood 108
## 11  Stone   2
## 12 Stucco   25
## 13 VinylSd 515
## 14 Wd Sdng 206
## 15 WdShing  26
## [1] "-----"
## [1] "Exterior2nd"
## [1] "-----"
##      f      n
## 1 AsbShng  20
## 2 AsphShn   3
## 3 Brk Cmn   7
## 4 BrkFace  25
## 5  CBlock   1
## 6 CmentBd  60
## 7 HdBoard 207
## 8 ImStucc  10
## 9 MetalSd 214
## 10 Other    1

```

```

## 11 Plywood 142
## 12 Stone 5
## 13 Stucco 26
## 14 VinylSd 504
## 15 Wd Sdng 197
## 16 Wd Shng 38
## [1] "-----"
## [1] "MasVnrType"
## [1] "-----"
## f n
## 1 BrkCmn 15
## 2 BrkFace 445
## 3 None 864
## 4 Stone 128
## 5 <NA> 8
## [1] "-----"
## [1] "ExterQual"
## [1] "-----"
## f n
## 1 Ex 52
## 2 Fa 14
## 3 Gd 488
## 4 TA 906
## [1] "-----"
## [1] "ExterCond"
## [1] "-----"
## f n
## 1 Ex 3
## 2 Fa 28
## 3 Gd 146
## 4 Po 1
## 5 TA 1282
## [1] "-----"
## [1] "Foundation"
## [1] "-----"
## f n
## 1 BrkTil 146
## 2 CBlock 634
## 3 PConc 647
## 4 Slab 24
## 5 Stone 6
## 6 Wood 3
## [1] "-----"
## [1] "BsmtQual"
## [1] "-----"
## f n
## 1 Ex 121
## 2 Fa 35
## 3 Gd 618
## 4 TA 649
## 5 <NA> 37
## [1] "-----"
## [1] "BsmtCond"
## [1] "-----"

```

```

##      f      n
## 1   Fa    45
## 2   Gd    65
## 3   Po     2
## 4   TA 1311
## 5 <NA>    37
## [1] "-----"
## [1] "BsmtExposure"
## [1] "-----"
##      f      n
## 1   Av 221
## 2   Gd 134
## 3   Mn 114
## 4   No 953
## 5 <NA>   38
## [1] "-----"
## [1] "BsmtFinType1"
## [1] "-----"
##      f      n
## 1  ALQ 220
## 2  BLQ 148
## 3  GLQ 418
## 4  LwQ  74
## 5  Rec 133
## 6  Unf 430
## 7 <NA>   37
## [1] "-----"
## [1] "BsmtFinType2"
## [1] "-----"
##      f      n
## 1  ALQ   19
## 2  BLQ   33
## 3  GLQ   14
## 4  LwQ   46
## 5  Rec   54
## 6  Unf 1256
## 7 <NA>   38
## [1] "-----"
## [1] "Heating"
## [1] "-----"
##      f      n
## 1 Floor    1
## 2 GasA 1428
## 3 GasW   18
## 4 Grav    7
## 5 OthW    2
## 6 Wall    4
## [1] "-----"
## [1] "HeatingQC"
## [1] "-----"
##      f      n
## 1 Ex 741
## 2 Fa  49
## 3 Gd 241

```



```

## 4 Po    1
## 5 TA 428
## [1] "-----"
## [1] "CentralAir"
## [1] "-----"
##      f      n
## 1 N    95
## 2 Y 1365
## [1] "-----"
## [1] "Electrical"
## [1] "-----"
##      f      n
## 1 FuseA   94
## 2 FuseF   27
## 3 FuseP    3
## 4 Mix     1
## 5 SBrkr 1334
## 6 <NA>     1
## [1] "-----"
## [1] "KitchenQual"
## [1] "-----"
##      f      n
## 1 Ex 100
## 2 Fa  39
## 3 Gd 586
## 4 TA 735
## [1] "-----"
## [1] "Functional"
## [1] "-----"
##      f      n
## 1 Maj1   14
## 2 Maj2    5
## 3 Min1   31
## 4 Min2   34
## 5 Mod    15
## 6 Sev     1
## 7 Typ 1360
## [1] "-----"
## [1] "FireplaceQu"
## [1] "-----"
##      f      n
## 1 Ex  24
## 2 Fa  33
## 3 Gd 380
## 4 Po  20
## 5 TA 313
## 6 <NA> 690
## [1] "-----"
## [1] "GarageType"
## [1] "-----"
##      f      n
## 1 2Types    6
## 2 Attchd 870
## 3 Basment  19

```

```

## 4 BuiltIn 88
## 5 CarPort 9
## 6 Detchd 387
## 7 <NA> 81
## [1] "-----"
## [1] "GarageFinish"
## [1] "-----"
## f n
## 1 Fin 352
## 2 RFn 422
## 3 Unf 605
## 4 <NA> 81
## [1] "-----"
## [1] "GarageQual"
## [1] "-----"
## f n
## 1 Ex 3
## 2 Fa 48
## 3 Gd 14
## 4 Po 3
## 5 TA 1311
## 6 <NA> 81
## [1] "-----"
## [1] "GarageCond"
## [1] "-----"
## f n
## 1 Ex 2
## 2 Fa 35
## 3 Gd 9
## 4 Po 7
## 5 TA 1326
## 6 <NA> 81
## [1] "-----"
## [1] "PavedDrive"
## [1] "-----"
## f n
## 1 N 90
## 2 P 30
## 3 Y 1340
## [1] "-----"
## [1] "PoolQC"
## [1] "-----"
## f n
## 1 Ex 2
## 2 Fa 2
## 3 Gd 3
## 4 <NA> 1453
## [1] "-----"
## [1] "Fence"
## [1] "-----"
## f n
## 1 GdPrv 59
## 2 GdWo 54
## 3 MnPrv 157

```

```

## 4 MnWw 11
## 5 <NA> 1179
## [1] "-----"
## [1] "MiscFeature"
## [1] "----"
##      f      n
## 1 Gar2    2
## 2 Othr    2
## 3 Shed   49
## 4 TenC    1
## 5 <NA> 1406
## [1] "-----"
## [1] "SaleType"
## [1] "----"
##      f      n
## 1 COD    43
## 2 Con     2
## 3 ConLD   9
## 4 ConLI   5
## 5 ConLw   5
## 6 CWD     4
## 7 New   122
## 8 Oth     3
## 9 WD  1267
## [1] "-----"
## [1] "SaleCondition"
## [1] "----"
##      f      n
## 1 Abnorml 101
## 2 AdjLand   4
## 3 Alloca   12
## 4 Family   20
## 5 Normal 1198
## 6 Partial  125
## [1] "-----"

```

## Handle Numerical Features

### Marina: YearBuilt, GarageYrBlt

Having a look at the data, I had the feeling that YearBuilt and GarageYrBlt would be quite correlated, because a garage is usually built at the same time as the house itself. Let's check:

```
#First, check missing values in train and test set
```

```
#Null values in YearBuilt column
sum(is.na(train$YearBuilt))
```

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

```
sum(is.na(test$YearBuilt))
```

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

No missing values in YearBuilt column

```
#Null values in GarageYrBlt column  
sum(is.na(train$GarageYrBlt))
```

```
## [1] 81
```

```
sum(is.na(test$GarageYrBlt))
```

```
## [1] 78
```

We have some missing values in GarageYrBlt column in both the train and the test set. Since we want to check the correlation with another feature, we don't want to impute values or remove rows. By now we are just going to create a temporary dataframe that does not include the rows with missing values in GarageYrBlt column

```
# Make a temporary dataframe without the rows where GarageYrBlt column is NAN  
train_temp = train %>% drop_na("GarageYrBlt")  
test_temp = test %>% drop_na("GarageYrBlt")
```

```
#Check that we dont for missing values to make sure we got rid of them  
sum(is.na(train_temp$GarageYrBlt))
```

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

```
sum(is.na(test_temp$GarageYrBlt))
```

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

Now we don't have NaNs, we can check the correlation between YearBuilt, GarageYrBlt

```
# Chekcing correlations with GarageYrBlt  
cor(train_temp['GarageYrBlt'], train_temp['YearBuilt'])
```

```
##           YearBuilt  
## GarageYrBlt 0.8256675
```

```
cor(test_temp['GarageYrBlt'], test_temp['YearBuilt'])
```

```
##           YearBuilt  
## GarageYrBlt  0.84415
```

As expected, these two columns are quite correlated. Since GarageYrBlt has NaN values and YearBuilt has all the data, we are dropping GarageYrBlt from the original dataframes.

```
train = select(train, -c(GarageYrBlt))  
test = select(test, -c(GarageYrBlt))
```

**Marina: GarageCars, GarageArea**

Let's do the same with GarageCars, GarageArea which seem to be correlated.

```
#First, check missing values in train and test set
```

```
#Null values in GarageCars column  
sum(is.na(train$GarageCars))
```

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

```
sum(is.na(test$GarageCars))
```

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

```
#Null values in GarageArea column  
sum(is.na(train$GarageArea))
```

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

```
sum(is.na(test$GarageArea))
```

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

We have one missing values in GarageCars and GarageArea columns in the test set. Since we want to check the correlation with another feature, we don't want to impute values or remove rows. By now we are just going to create a temporary dataframe that does not include the rows with missing values in GarageCars and GarageArea columns

```
# Make a temporary dataframe without the rows where GarageCars and GarageArea column in NAN  
test_temp = test %>% drop_na("GarageCars", "GarageArea")  
sum(is.na(test_temp$GarageCars))
```

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

```
sum(is.na(test_temp$GarageArea))
```

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

Now we don't have NaNs, we can check the correlation between GarageCars and GarageArea

```
# Chekcing correlation between GarageCars and GarageArea  
cor(train['GarageCars'], train['GarageArea'])
```

```
##           GarageArea  
## GarageCars  0.8824754
```

```
cor(test_temp['GarageCars'], test_temp['GarageArea'])
```

```
##           GarageArea  
## GarageCars  0.8966743
```

As expected, these two columns are quite correlated, so we are dropping GarageCars (which is less descriptive) from the original dataframes.

```
train = select(train, -c(GarageCars))  
test = select(test, -c(GarageCars))
```

## Handle Categorical Features