Data Analysis Project: AMES House Price Prediction

Dhanush Harihar, Netid: dhanush2; Viney Kharbanda, Netid:vkk2; Chien Wei Huang, Netid:cwhuang3

7/27/2019

Introduction

This document contains data analysis done on AMES House Predicton dataset. The intent of this project is to develop a multi-linear model that can predict house prices(response variable) based on the predictors(rest of the variables) we choose that optimizes the model. The linear model will be developed based on data and selected variables.

This project will incorporate the following concepts:

- Data exploration
- Data cleaning
- Variable selection
- Assumption

- Transformation
- Model building
- Model Selection
- Model evaluation
- Interpertation

Description of dataset

The Ames Housing dataset was compiled by Dean De Cock for use in data science education. The data set describes the sale of individual residential property in Ames, Iowa from 2006 to 2010.

The data set contains 1460 observations and 81 number of explanatory variables involved in assessing homevalues. The dataset will be randomly split into training data set and testing data set.

Most of the variables are the type of information that a typical home buyer would be interested in knowing about a potential property (e.g. When was it built? How big is the lot? How many bathrooms are there?How many square foot? Is there a pool?)

Below are some of the key variables in the dataset which we are interested in:

Response

- `SalePrice` : Response is the sale price and is Numeric.

Numerical Predictors

- `GrLivArea` : Above grade (ground) living area square feet
- `GarageArea` : Size of garage in square feet
- `TotalBsmtSF` : Total square feet of basement area
- `1stFlrSF` : First Floor square feet
- `OpenPorchSF` : Open porch area in square feet
- `2ndFlrSF` : Second floor square feet
- `LotArea` : Lot size in square feet

Categorical Predictors

- `OverallQual` : Overall material and finish quality
- `GarageCars` : Size of garage in car capacity
- `FullBath` : Full bathrooms above grade
- `Bedroom` : Bedrooms above grade
- `Kitchen` : Kitchens above grade
- `TotRmsAbvGrd` : Total rooms above grade (does not include bathrooms)
- `OverallCond` : Rates the overall condition of the house

Dataset Source

This project is inspired by Kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)

Business Interest

We want to explore what features influence price of the house in AMES, Iowa. Some of questions are:

- Does lot area of the house determine price of house?
- Does having a enclosed porch improves value of the house?
- Does having a larger Garage Capacity increases value of the house?

These findings can also help a homeowner decide if a certain rennovation(like adding a pool, porch etc) should be done that will eventually lead to a higher Sale Price.

Methods

Here we show the most relevant R code for analyzing this dataset. See the .Rmd files for additional R code not seen in this document.

Load required libraries

```
install.packages("dplyr")
install.packages("faraway")
install.packages("lmtest")
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
library(lmtest)
library(faraway)
library(knitr)
library(readr)
```

Functions to perform model diagnostics

```
perform_bp_test = function(model, alpha = '0.05') {
  decide = unname(bptest(model)$p.value < alpha)
  ifelse(decide,
         "Constant Variance suspect",
         "Constant Variance assumption not suspect")
}

perform_shapiro_wilk_test = function(model, alpha = '0.05') {
  decide = unname(shapiro.test(resid(model))$p.value < alpha)
  ifelse(decide,
         "Normality assumption suspect",
         "Normality assumption not suspect")
}

get_num_params = function(model) {
  length(coef(model))
}

calc_loocv_rmse = function(model, response_in_log = FALSE) {
  if (response_in_log == TRUE) {
    y_hat = exp(fitted(model))
    y = exp(model$model[,1])
  } else {
    y_hat = fitted(model)
    y = model$model[,1]
  }

  residual =  y - y_hat
  round(sqrt(mean((residual / (1 - hatvalues(model))) ^ 2)), 3)
}

calc_adj_r2 = function(model) {
  round(summary(model)$adj.r.squared, 3)
}

calc_r2 = function(model) {
  round(summary(model)$r.squared, 3)
}

calc_prediction_error = function(model, newdata, new_response, response_in_log = FALSE) {
  if (response_in_log == TRUE) {
    predicted_response = exp(predict(model, newdata = newdata))
  } else {
    predicted_response = predict(model, newdata = newdata)
  }
  round(mean((abs(predicted_response - new_response) / new_response) * 100), 3)
}

diagnostics = function(model, pcol = 'grey', lcol = 'dodgerblue',
                       alpha = '0.05', plotit = TRUE, testit = TRUE, displayit = FALSE, returnit =
  TRUE,
                       response_in_log = FALSE, newdata = NA, new_response = NA) {
```

```
  if (testit == TRUE) {
    loocv_rmse = calc_loocv_rmse(model, response_in_log)
    adj_r2 = calc_adj_r2(model)
    r2 = calc_r2(model)
    bp_decision = perform_bp_test(model, alpha)
    sw_decision = perform_shapiro_wilk_test(model, alpha)
    num_params = get_num_params(model)
    vif_result = vif(model)
    num_colinear_columns = sum(vif_result > 5)
    num_predictors = length(model$model) - 1

    if (all(is.na(newdata))) {
      test_error = "Not calculated"
    } else {
      test_error = calc_prediction_error(model, newdata, new_response, response_in_log)
    }

    if (displayit == TRUE) {
      print(paste('LOOCV RMSE :', loocv_rmse))
      print(paste('Adjusted R2:', adj_r2))
      print(paste('Test Error :', test_error))
      #print(paste('R2        :', r2))
      print(paste('Num of predictors:', num_predictors))
      print(paste('Num of paramaters:', num_params))
      #print(paste('Num of colinear columns  :', num_colinear_columns))
      print(paste('BP test decision          :', bp_decision))
      print(paste('Shapiro Wilk test decision:', sw_decision))
    }

    if (plotit == TRUE) {
      par(mfrow = c(1, 2))
      plot(fitted(model), resid(model),
           col = pcol, pch = 20, xlab = "Fitted",
           ylab = "Residuals.", main = "Fitted vs Residuals")
      abline(h = 0, col = lcol, lwd = 2)

      qqnorm(resid(model), col = pcol, main = "Normal Q-Q Plot")
      qqline(resid(model), col = lcol, lwd = 2)
    }

    if (returnit == TRUE) {
      list("loocv_rmse" = loocv_rmse, "adj_r2" = adj_r2,
           "r2" = r2, "bp_decision" = bp_decision,
           "sw_decision" = sw_decision, "num_params" = num_params,
           "num_colinear_columns" = num_colinear_columns,
           "test_error" = test_error, "num_predictors" = num_predictors
           )
    }
  }
}
```

Utility Functions

```
update_char_column_value = function(data_frame, column_name, value,
                                    update_row_indices = NA,
                                    convert_to_factor = TRUE) {

  data_frame[, column_name] = as.character(data_frame[, column_name])
  if (all(is.na(update_row_indices))) {
    data_frame[which(is.na(data_frame[, column_name])), column_name] = value
  } else {
    data_frame[update_row_indices, column_name] = value
  }
  if (convert_to_factor) {
    data_frame[, column_name] = as.factor(data_frame[, column_name])
  }
  data_frame
}
```

Plotting Functions

```
draw_boxplot = function(x = NA, formula = NA, data = NA, xlab = "",
                        ylab = "", main = "", horizontal = FALSE, notch = FALSE) {

  options(scipen = 50, digits = 5)
  if (all(is.na(x))) {
    outliers = boxplot(formula, data, xlab = xlab, ylab = ylab,
                       main = main, col = "orange", border = "black",
                       horizontal = horizontal, notch = notch)$out
  } else {
    outliers = boxplot(x = x, xlab = xlab, ylab = ylab, main = main,
                       col = "orange", border = "black", horizontal = horizontal, notch = notch)$ou
  t
    outliers
  }
}

create_table = function(data_frame, col_names, caption = "") {
  kable(data_frame, col.names = col_names, caption = caption)
}

draw_corelation_plot = function(data, main = "") {
  pairs(data, col = "dodgerblue", main=main)
}

plot_predicted_vs_actual = function(x, y, xlab = "", ylab = "", main = "") {
  options(scipen = 50, digits = 5)
  plot(x = x, y = y,
       col = 'darkorange', pch = 20, xlab = xlab,
       ylab = ylab, main = main,
       xlim = c(0,max(x)),
       ylim = c(0,max(y)))
  abline(a = 0, b = 1, col = 'dodgerblue', lwd = 2)
}
```

Load Dataset

```
housing_data = read.csv("housingdata.csv")
```

Current dimensions of dataset are 1460, 81

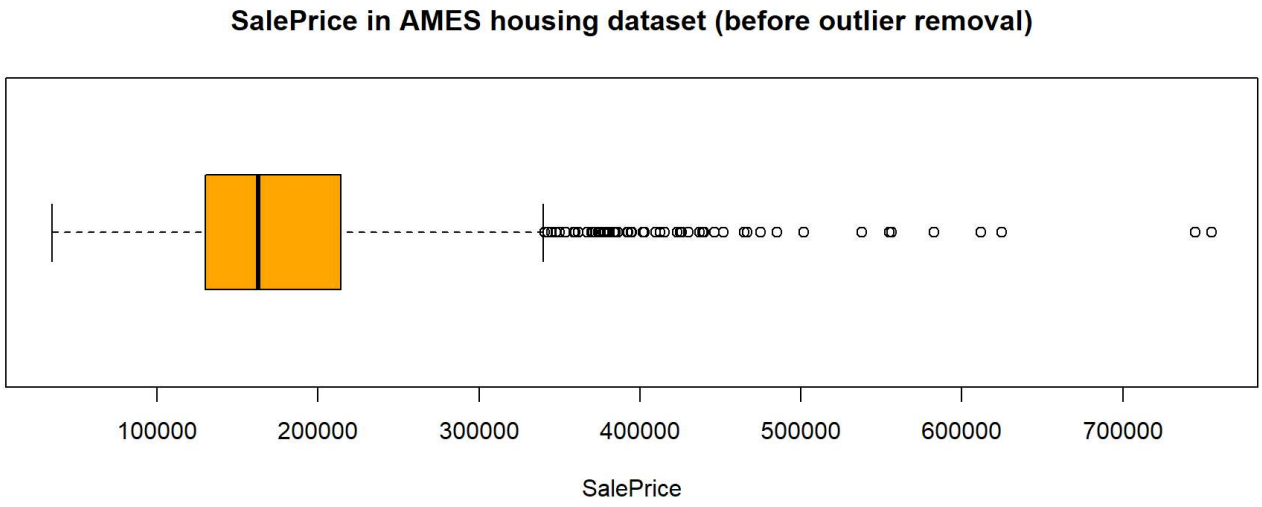
We will look at the data first and do cleaning. Then we do variable selection.

Clean Data

Analyze data and perform cleaning like remove unusual observations, remove unwanted data, and treat missing data,

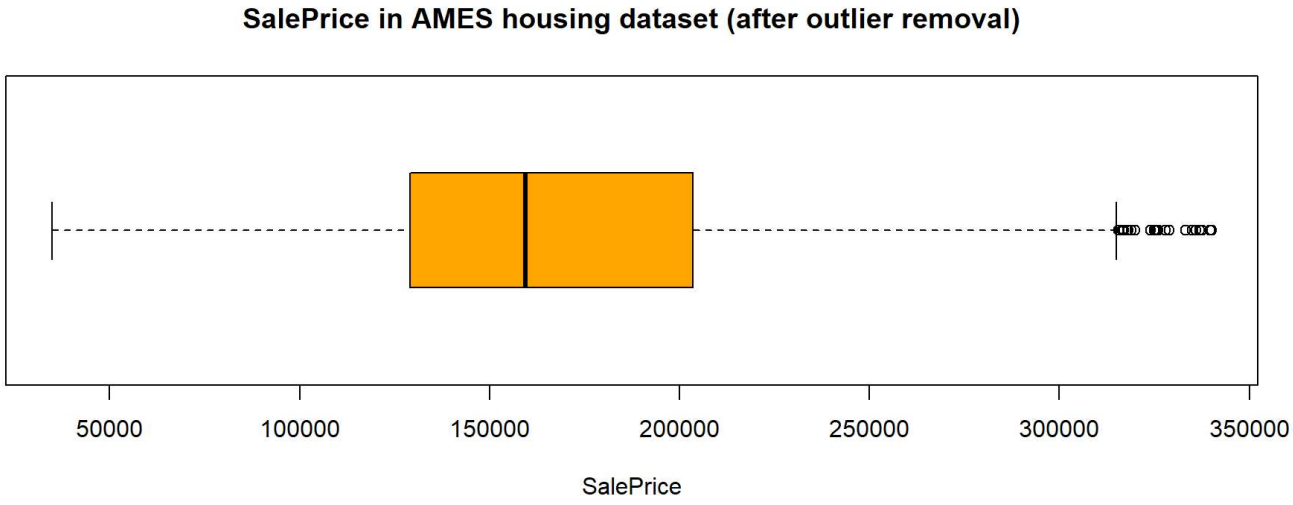
Remove outliers

Detect outliers using univariate approach on response variable `SalePrice` .



The below identified outliers in `SalePrice` are removed.

```
##  [1] 341000 342643 345000 345000 348000 350000 350000 354000 359100 360000
## [11] 361919 367294 369900 370878 372402 372500 374000 375000 377426 377500
## [21] 378500 380000 381000 383970 385000 385000 386250 392000 392500 394432
## [31] 394617 395000 395192 402000 402861 403000 410000 412500 415298 423000
## [41] 424870 426000 430000 437154 438780 440000 446261 451950 465000 466500
## [51] 475000 485000 501837 538000 555000 556581 582933 611657 625000 745000
## [61] 755000
```



Remove unwanted columns

Remove `Id` column as it is not significant for prediction.

Treat missing data

We will investiage at each column and see how we can treat missing data in those columns.

Missing Data Summary

Column Name	% of Missing data

Column Name	% of Missing data
PoolQC	99.57112
MiscFeature	96.14010
Alley	93.49535
Fence	80.20014
FireplaceQu	49.24946
LotFrontage	18.29878
GarageType	5.78985
GarageYrBlt	5.78985
GarageFinish	5.78985
GarageQual	5.78985
GarageCond	5.78985
BsmtExposure	2.71623
BsmtFinType2	2.71623
BsmtQual	2.64475
BsmtCond	2.64475
BsmtFinType1	2.64475
MasVnrType	0.50036
MasVnrArea	0.50036
Electrical	0.07148

`PoolQC` , `MiscFeature` , and `Alley` variables have most missing data.

We will make the below changes based on our investigation.

- `PoolQC` has value `NA` when `PoolArea` is 0, that means no pool. So mark it `None` .
- `MiscFeature` is empty when `MiscVal` is 0. So set `MiscFeature` to `None` when `MiscVal` is 0.
- Looking at data description `NA` in `Alley` mean no alley access. So replace `NA` with `None` in `Alley` column.
- It appears `NA` in `Fence` is no fence, so mark this as `None` also.
- `FireplaceQu` has value `NA` when `Fireplaces` is 0, that means no fireplace, so mark it `None` .
- For `LotFrontage` column, it appears that there is no relationship with other missing `NA` . So see we will replace `NA` with median.
- All the garage related columns have same percent of data missing. Perhaps there is no garage. Replace those with `None` .
- All the basement related columns have similar missing data percentage. Perhaps there is no basement. Replace those with `No Basement` .
- Missing data for `MasVnrType` and `MasVnrArea` mean those houses do not have masonry veneer walls. So marking `MasVnrType` as `None` and `MasVnrArea` as 0.

- One house does not have Electrical Info. Maybe it is not fully built yet. So remove data for this house.

Check if there are any more `NA` values.

```
sum(colSums(is.na(housing_data)) > 0)
```

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

There are none and we are done with missing data check and data treatment.

Current dimensions of dataset are 1398, 80

Transform Data

Some columns are integer type but should be factors. So convert those columns to factors.

Variable Selection

Split data into train and test

We will randomly split the data to train and test set before proceeding to next step.

```
set.seed(420)
hd_trn_idx  = sample(nrow(housing_data), size = trunc(0.80 * nrow(housing_data)))
ames_trn_data = housing_data[hd_trn_idx, ]
ames_tst_data = housing_data[-hd_trn_idx, ]
```

Look at Full additive model and Smaller model using BIC parameter selection

Before doing detailed variable selection we want to check how the model with all predictors perform.

We use a full additive model with all parameters and then do variable selection using backwards BIC.

```
model_add = lm(SalePrice ~ . ,data = ames_trn_data)
model_add_bic = step(model_add, direction = "backward", k = log(nrow(ames_trn_data)), trace = 0)
(model_add_bic_rmse = calc_loocv_rmse(model_add_bic))
```

For the model selected using `BIC` , we got LOOCV RMSE = `Inf` . This model is too complicated.

So need to do detailed variable selection by checking which predictors impact the response variable `SalePrice` the most.

Analyze and select numerical variables

We will first study numerical columns using correlation.

There are 31 numerical variables.

Lets us see their corelation with `SalePrice` .

```
##     SalePrice     GrLivArea    GarageCars    GarageArea      FullBath
##     1.0000000     0.6683954     0.6254449     0.6002089     0.5744976
##     YearBuilt   TotalBsmtSF     X1stFlrSF  TotRmsAbvGrd    Fireplaces
##     0.5636756     0.5432536     0.5285584     0.4802853     0.4552451
##   OpenPorchSF    MasVnrArea     X2ndFlrSF    BsmtFinSF1    WoodDeckSF
##     0.3540563     0.3468641     0.3170160     0.3162749     0.2884532
##   LotFrontage      HalfBath       LotArea  BsmtFullBath  BedroomAbvGr
##     0.2857157     0.2750350     0.2470137     0.2194262     0.2052221
##     BsmtUnfSF   ScreenPorch        MoSold    X3SsnPorch      PoolArea
##     0.2049408     0.1122083     0.0553998     0.0543648     0.0543421
##       MiscVal    BsmtFinSF2  BsmtHalfBath  LowQualFinSF EnclosedPorch
##     0.0012981    -0.0120948    -0.0218731    -0.0699524    -0.1186797
##  KitchenAbvGr
##    -0.1462734
```

Next we will pick top 10 variables based on corelation values and study their colinearity and relationship.

Lets us display the corelation matrix and plot for these top 10 variables.

```
##             SalePrice GrLivArea GarageCars GarageArea FullBath YearBuilt
## SalePrice      1.00000   0.66840    0.62544    0.60021  0.57450  0.563676
## GrLivArea      0.66840   1.00000    0.40702    0.41856  0.60168  0.154744
## GarageCars     0.62544   0.40702    1.00000    0.87097  0.43729  0.515455
## GarageArea     0.60021   0.41856    0.87097    1.00000  0.37030  0.448284
## FullBath       0.57450   0.60168    0.43729    0.37030  1.00000  0.448090
## YearBuilt      0.56368   0.15474    0.51545    0.44828  0.44809  1.000000
## TotalBsmtSF    0.54325   0.39505    0.36818    0.43389  0.27050  0.349955
## X1stFlrSF      0.52856   0.52231    0.37774    0.43459  0.33604  0.223338
## TotRmsAbvGrd   0.48029   0.81869    0.29754    0.27918  0.50857  0.035597
## Fireplaces     0.45525   0.44127    0.26479    0.23665  0.20128  0.113890
##              TotalBsmtSF X1stFlrSF TotRmsAbvGrd Fireplaces
## SalePrice        0.54325   0.52856     0.480285    0.45525
## GrLivArea        0.39505   0.52231     0.818692    0.44127
## GarageCars       0.36818   0.37774     0.297539    0.26479
## GarageArea       0.43389   0.43459     0.279182    0.23665
## FullBath         0.27050   0.33604     0.508567    0.20128
## YearBuilt        0.34996   0.22334     0.035597    0.11389
## TotalBsmtSF      1.00000   0.77715     0.228100    0.29901
## X1stFlrSF        0.77715   1.00000     0.373160    0.38996
## TotRmsAbvGrd     0.22810   0.37316     1.000000    0.29931
## Fireplaces       0.29901   0.38996     0.299305    1.00000
```

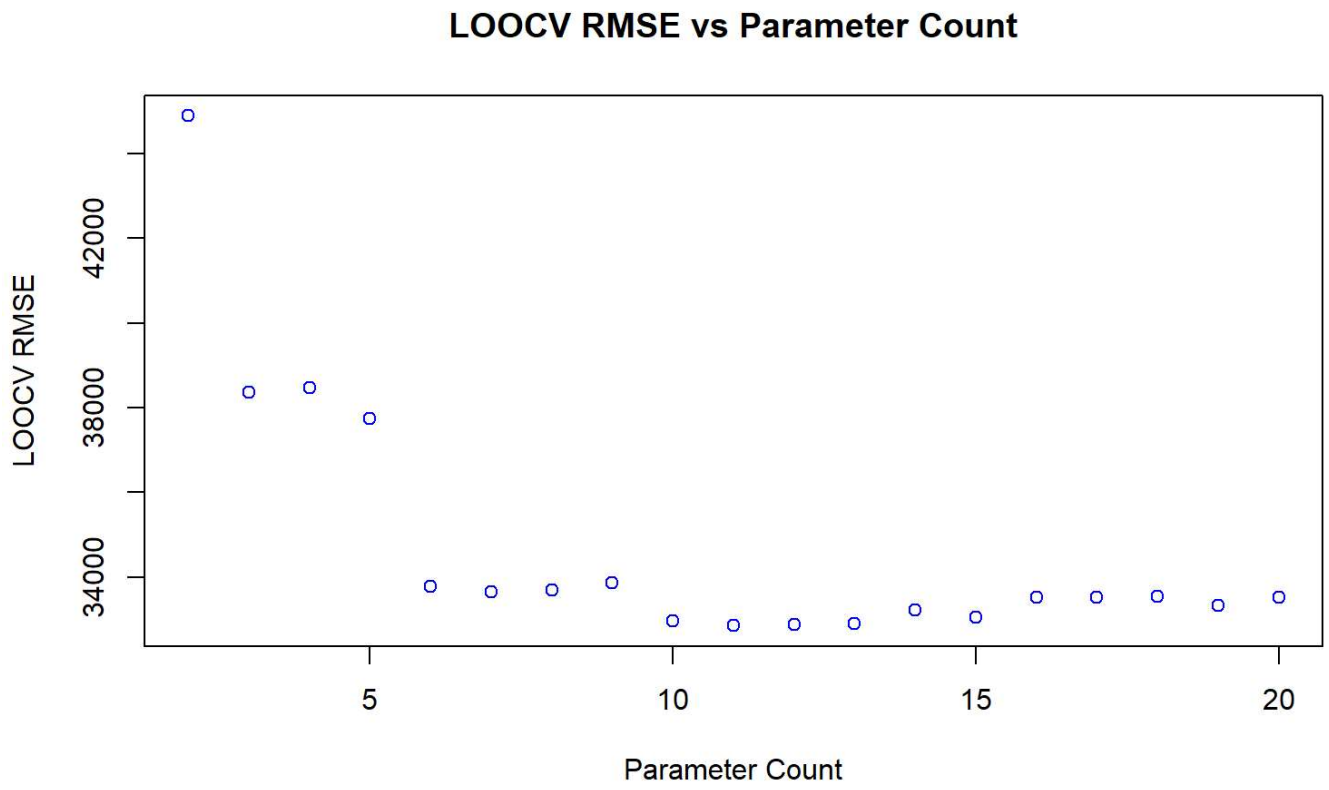
From analysis and above plot, it looks like `GarageCars` and `GarageArea` are collinear. `GrLivArea` and `TotRmsAbvGrd` are collinear. `TotalBsmtSF` and `X1stFlrSF` are collinear. Also we need to decide how many parameters to select.

So find these answers, we will iterate and see how adding more parameters in decreasing order of corelation affects $LOOCV\ RMSE$.

We do not want the model to be complex. So we start with maximum of 20 parameters selected based on their corelation with `SalePrice` .

```
loocv_rmse = rep(0, param_count)
num_params = rep(0, param_count)
cor_values_descending = sort(abs(cor(numeric_ames_trn_data)["SalePrice",]), decreasing = TRUE)
for (i in 2:param_count) {
  topx_col_names = names(cor_values_descending)[1:i]
  model = lm(SalePrice ~ ., data = numeric_ames_trn_data[, topx_col_names])
  loocv_rmse[i] = calc_loocv_rmse(model)
  num_params[i] = get_num_params(model)
}
min_index = which.min(loocv_rmse[2:param_count])
min_index = min_index + 1
```

Next we plot $LOOC\ RMSE$ Vs Parameter count we got from the above code on scatter plot.



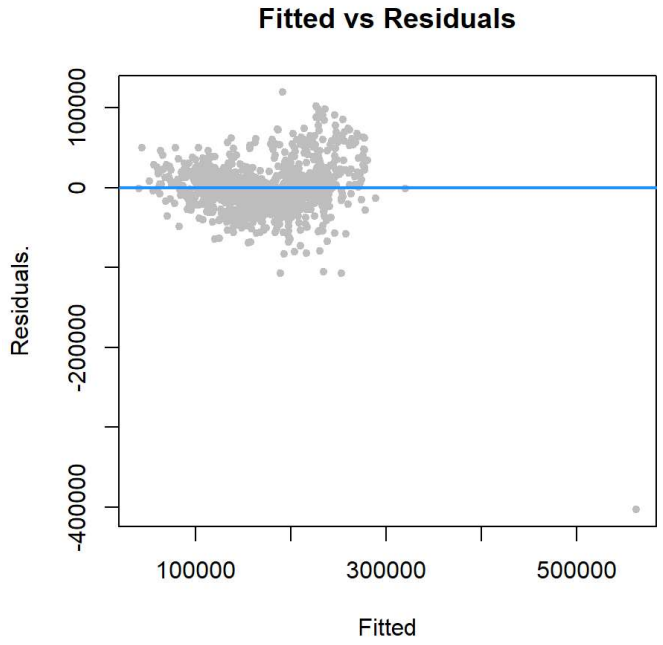
From plot above, we can see that top 11 number of correlated predictors give the best $LOOC\ RMSE$.

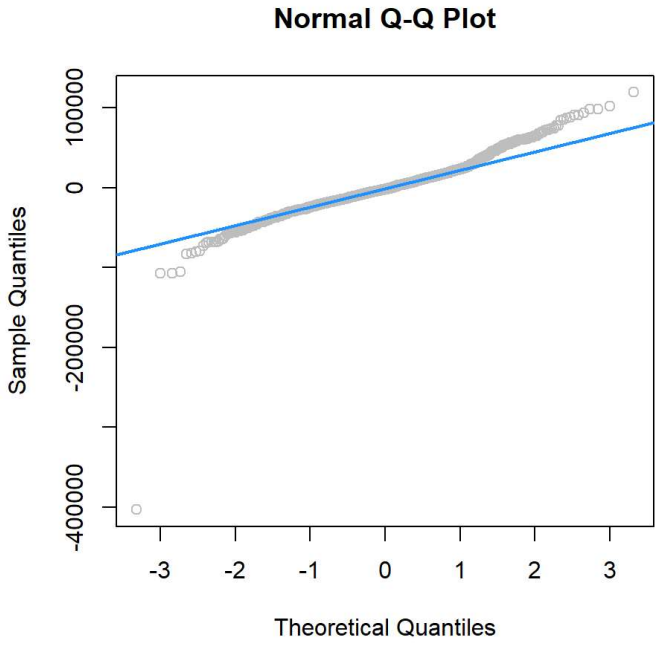
Below we will check to see how a model with these selected variables perform.

```
top_cor_col_names = names(cor_values_descending)[1:min_index]
model_numeric = lm(SalePrice ~ ., data = numeric_ames_trn_data[, top_cor_col_names])
```

Below is the result of diagnostics done on this model:

```
## [1] "LOOCV RMSE : 32856.608"
## [1] "Adjusted R2: 0.733"
## [1] "Test Error : 13.485"
## [1] "Num of predictors: 10"
## [1] "Num of paramaters: 11"
## [1] "BP test decision          : Constant Variance assumption not suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```





Now out of these 11 predictors remove collinear variables.

GarageCars vs GarageArea

Model	LOOCV RMSE	$Adjusted\ R^2$
Model (GarageCars removed)	32936	0.729
Model (GarageArea removed)	32640	0.733

We formed one model by removing `GarageCars` and another one by removing `GarageArea` and then calculate $LOOCV\ RMSE$ and $Adjusted\ R^2$. Here we chose to remove `GarageArea` since $Adjusted\ R^2$ is better when that is removed.

GrLivArea vs TotRmsAbvGrd

Model	LOOCV RMSE	$Adjusted\ R^2$
Model (GrLivArea removed)	33658	0.699
Model (TotRmsAbvGrd removed)	32446	0.733

We formed one model by removing `GrLivArea` and another one by removing `TotRmsAbvGrd` and then calculate $LOOCV\ RMSE$ and $Adjusted\ R^2$. Here we chose to remove `TotRmsAbvGrd` since $Adjusted\ R^2$ is better when that is removed.

TotalBsmtSF vs X1stFlrSF

Model	LOOCV RMSE	$Adjusted\ R^2$
Model (TotalBsmtSF removed)	32340	0.724
Model (X1stFlrSF removed)	32419	0.733

We formed one model by removing `TotalBsmtSF` and another one by removing `X1stFlrSF` and then calculate $LOOCV\ RMSE$ and $Adjusted\ R^2$. Here we chose to remove `X1stFlrSF` since $Adjusted\ R^2$ is better when that is removed.

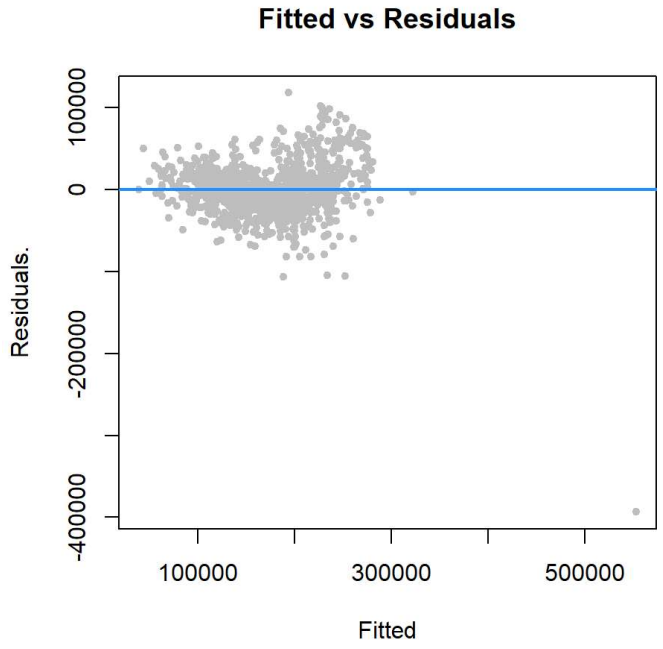
So finally we select these numerical columns
`GrLivArea, GarageCars, FullBath, YearBuilt, TotalBsmtSF, Fireplaces, OpenPorchSF .`

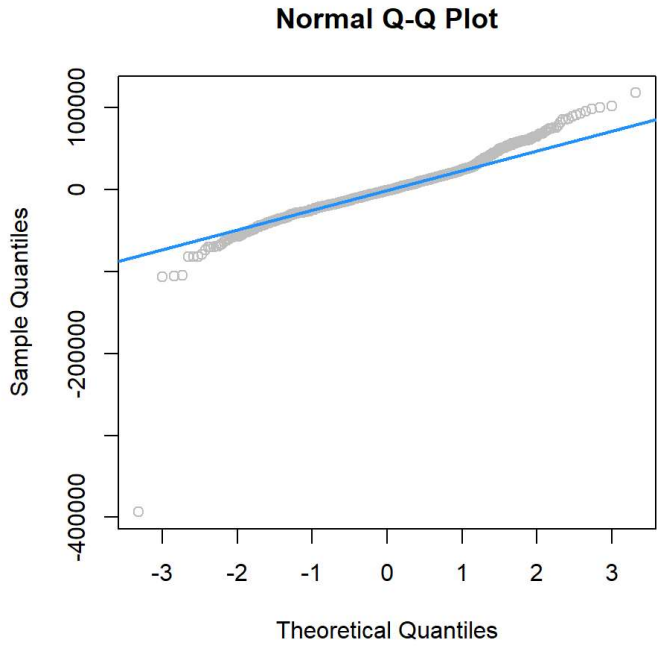
Below we will check to see how a model with these selected variables perform.

```
model_numeric_remove_colinear = lm(SalePrice ~ .,
    data = numeric_ames_trn_data[, union(selected_numerical_columns, response_column)])
```

Below is the result of diagnostics done on this model:

```
## [1] "LOOCV RMSE : 32419.358"
## [1] "Adjusted R2: 0.733"
## [1] "Test Error : 13.581"
## [1] "Num of predictors: 7"
## [1] "Num of paramaters: 8"
## [1] "BP test decision          : Constant Variance assumption not suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```



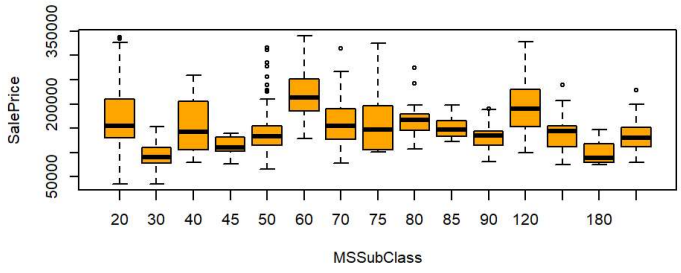


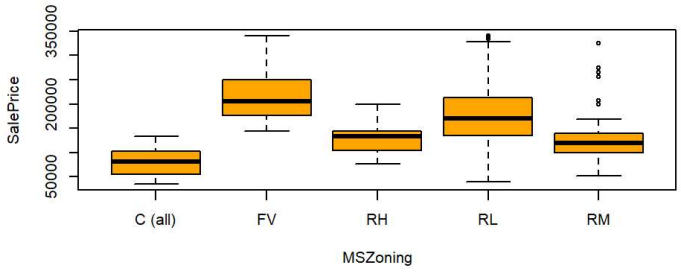
See can see that LOOCV RMSE has improved when we removed colinear variables.

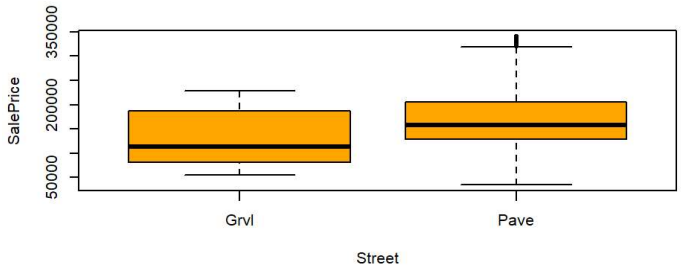
Next we will check to see if we can improve the model by including some sigificant factor variables (including categorical variables).

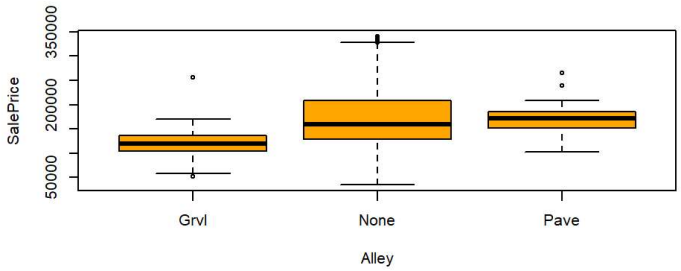
Analyze and select factor variables

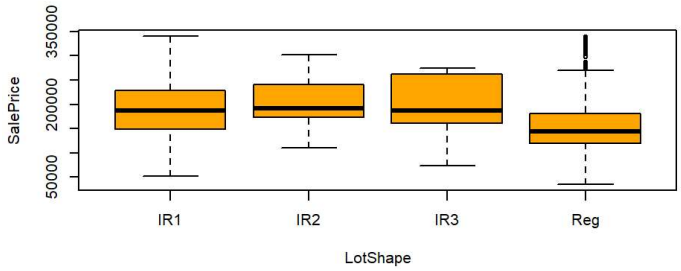
In this section we will study the factor variables by looking at boxplots for `SalePrice` vs all factor variables to see how each of them impact the response.

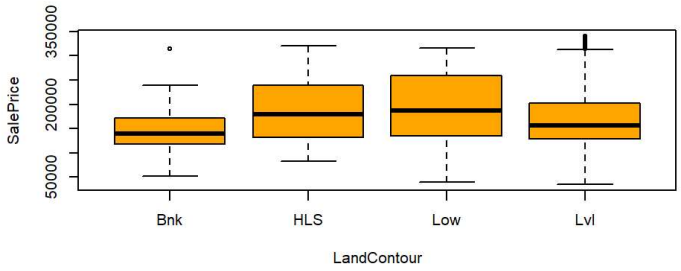


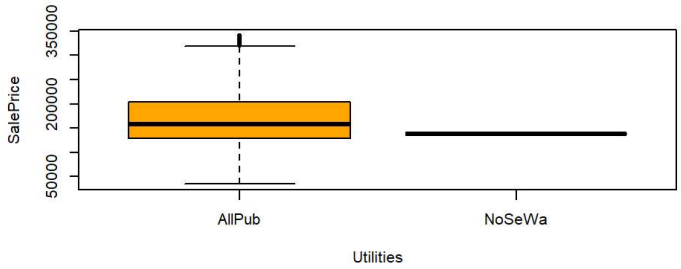


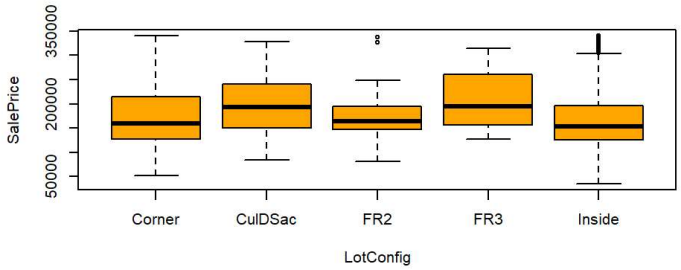


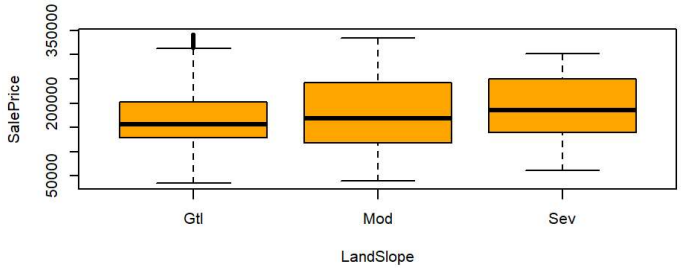


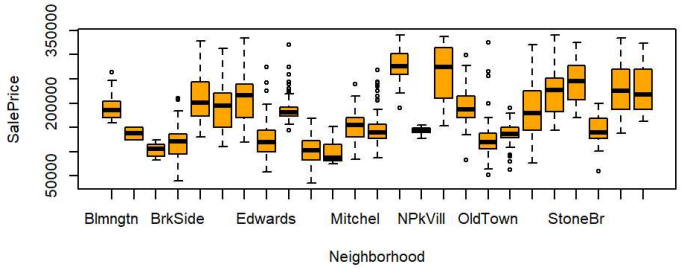


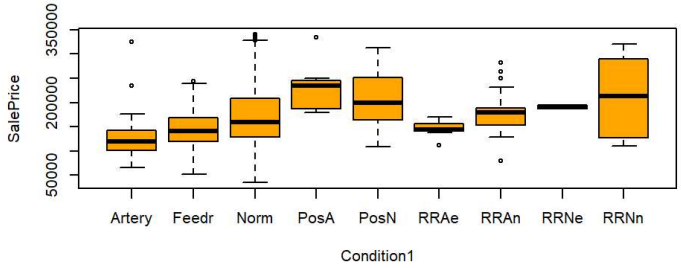


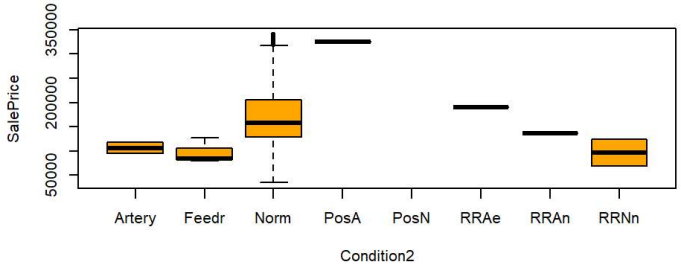

SalePrice
50000
200000
350000
AllPub
NoSeWa
Utilities

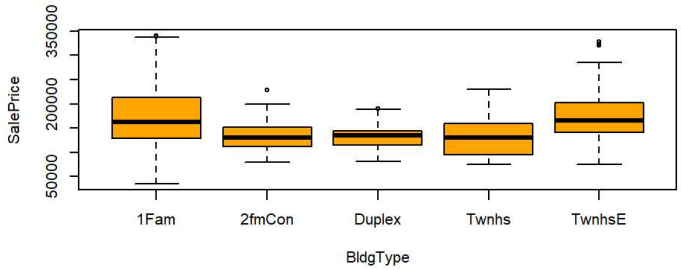

SalePrice
50000
200000
350000
Corner
CulDSac
FR2
FR3
Inside
LotConfig

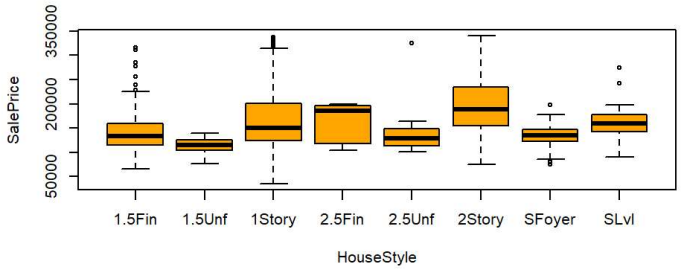

SalePrice
50000
200000
350000
Gtl
Mod
Sev
LandSlope

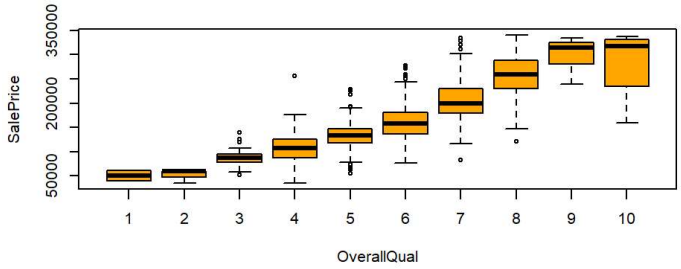

SalePrice
50000
200000
350000
Blmngtn
BrkSide
Edwards
Mitchel
NPkVill
OldTown
StoneBr
Neighborhood

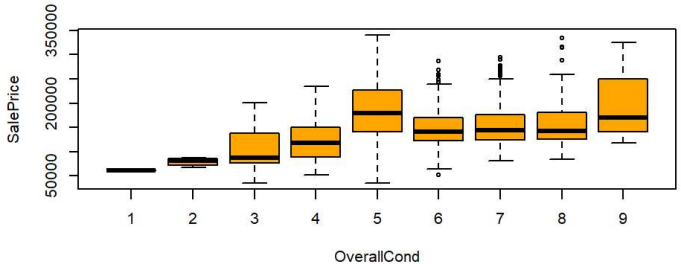

SalePrice
50000
200000
350000
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNe
RRNn
Condition1

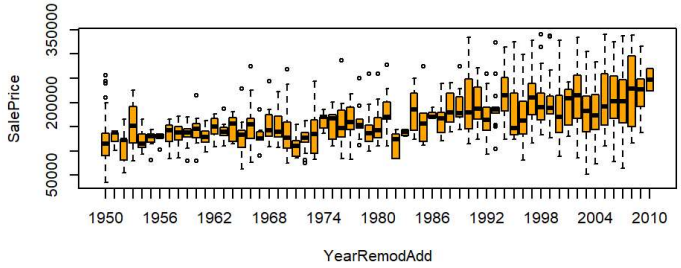

SalePrice
50000
200000
350000
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNn
Condition2

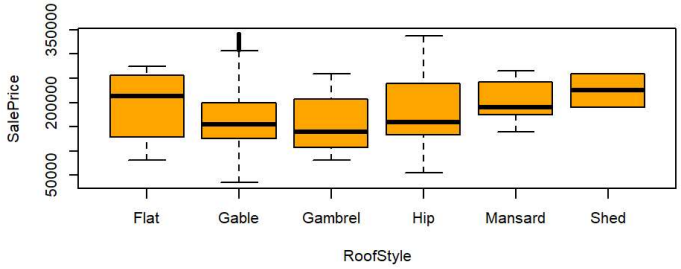

SalePrice
50000
200000
350000
1Fam
2fmCon
Duplex
Twnhs
TwnhsE
BldgType

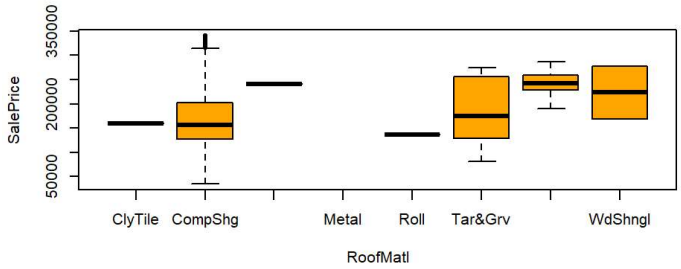

SalePrice
50000
200000
350000
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl
HouseStyle

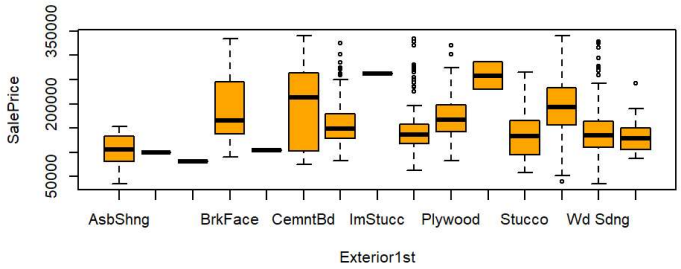

SalePrice
50000
200000
350000
1
2
3
4
5
6
7
8
9
10
OverallQual

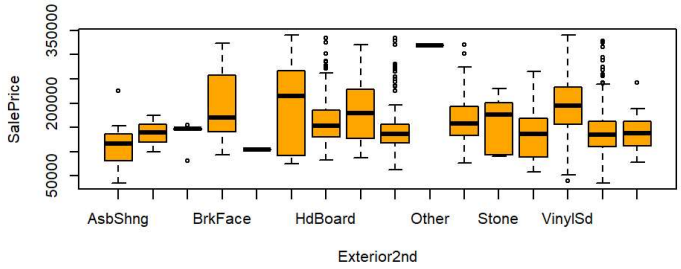

SalePrice
50000
200000
350000
1
2
3
4
5
6
7
8
9
OverallCond

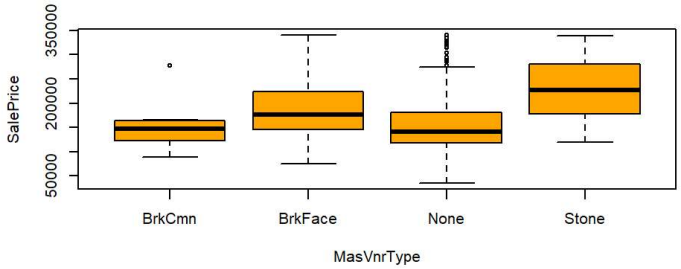

SalePrice
50000
200000
350000
1950
1956
1962
1968
1974
1980
1986
1992
1998
2004
2010
YearRemodAdd

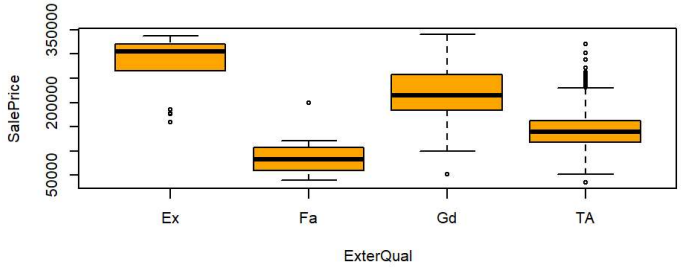

SalePrice
50000
200000
350000
Flat
Gable
Gambrel
Hip
Mansard
Shed
RoofStyle

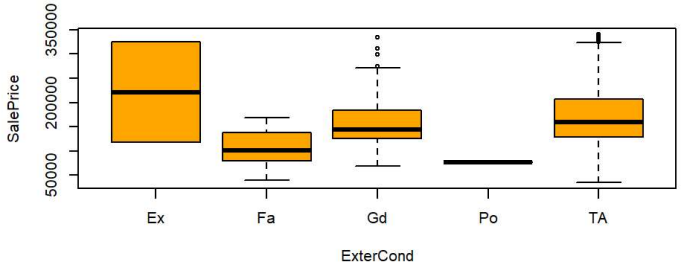

SalePrice
50000
200000
350000
ClyTile
CompShg
Metal
Roll
Tar&Grv
WdShngl
RoofMatl

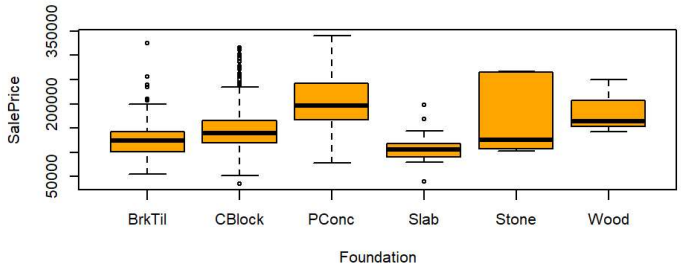

SalePrice
50000
200000
350000
AsbShng
BrkFace
CemntBd
ImStucc
Plywood
Stucco
Wd Sdng
Exterior1st

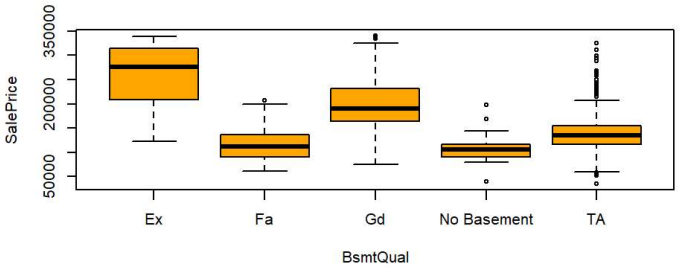

SalePrice
50000
200000
350000
AsbShng
BrkFace
HdBoard
Other
Stone
VinylSd
Exterior2nd

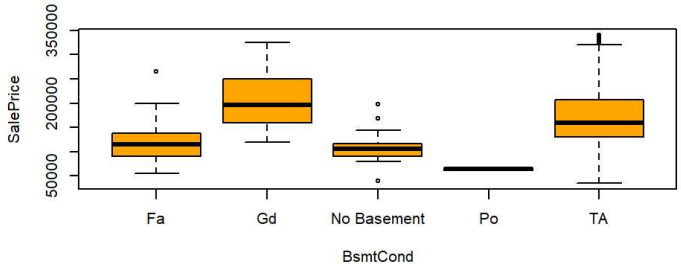

SalePrice
50000
200000
350000
BrkCmn
BrkFace
None
Stone
MasVnrType

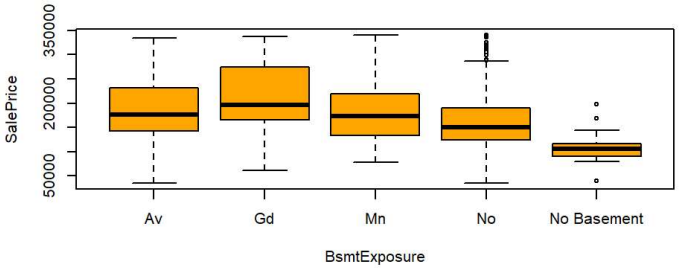

SalePrice
50000
200000
350000
Ex
Fa
Gd
TA
ExterQual

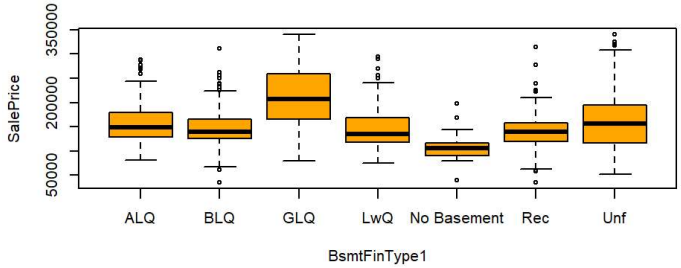

SalePrice
50000
200000
350000
Ex
Fa
Gd
Po
TA
ExterCond

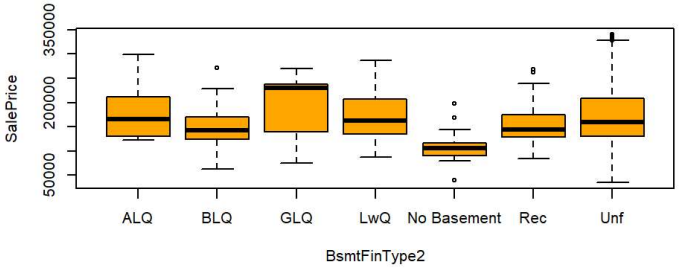

SalePrice
50000
200000
350000
BrkTil
CBlock
PConc
Slab
Stone
Wood
Foundation

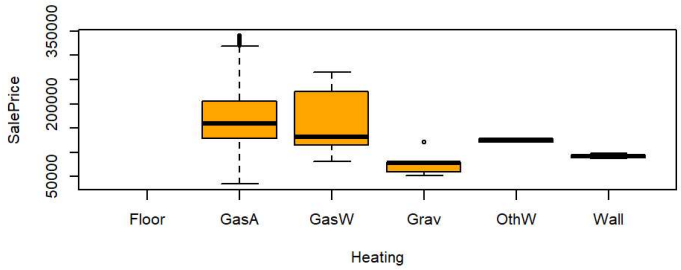

SalePrice
50000
200000
350000
Ex
Fa
Gd
No Basement
TA
BsmtQual

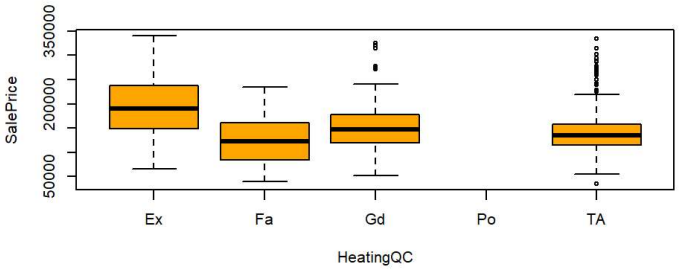

SalePrice
50000
200000
350000
Fa
Gd
No Basement
Po
TA
BsmtCond

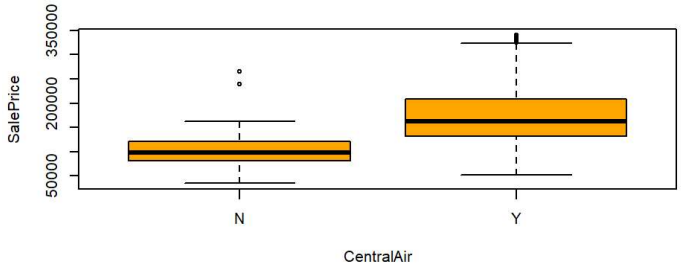

SalePrice
50000
200000
350000
Av
Gd
Mn
No
No Basement
BsmtExposure

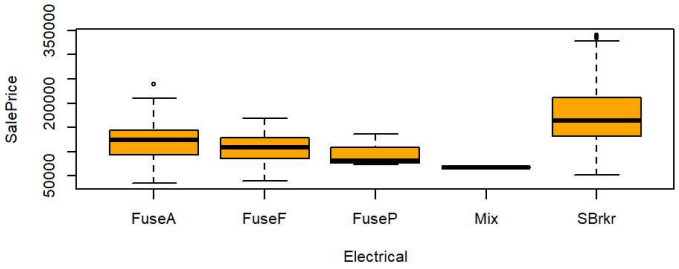

SalePrice
50000
200000
350000
ALQ
BLQ
GLQ
LwQ
No Basement
Rec
Unf
BsmtFinType1

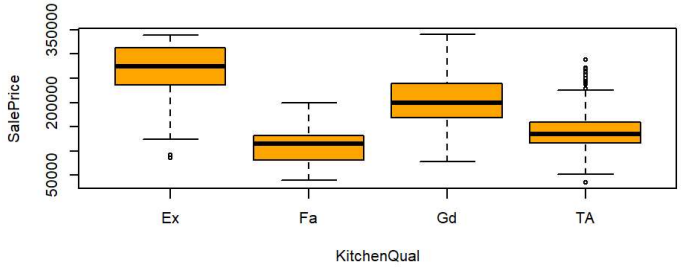

SalePrice
50000
200000
350000
ALQ
BLQ
GLQ
LwQ
No Basement
Rec
Unf
BsmtFinType2

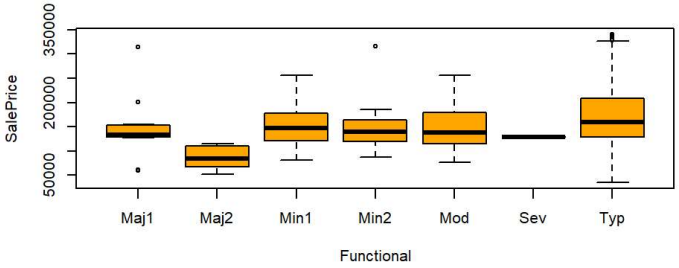

SalePrice
50000
200000
350000
Floor
GasA
GasW
Grav
OthW
Wall
Heating

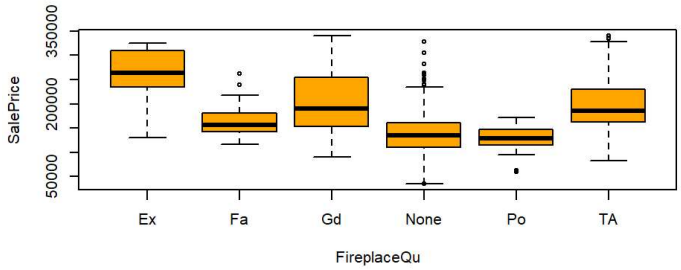

SalePrice
50000
200000
350000
Ex
Fa
Gd
Po
TA
HeatingQC

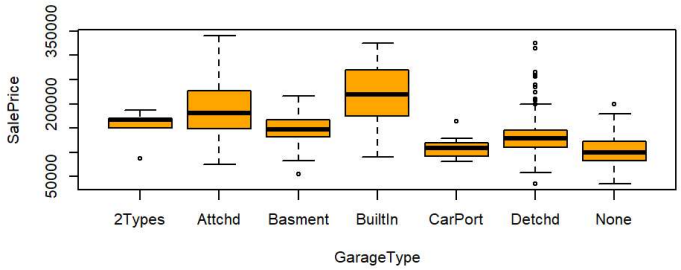

SalePrice
50000
200000
350000
N
Y
CentralAir

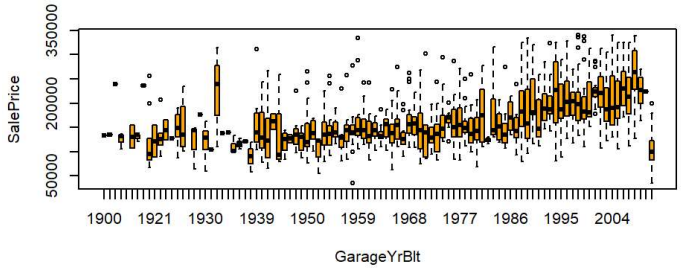

SalePrice
50000
200000
350000
FuseA
FuseF
FuseP
Mix
SBrkr
Electrical

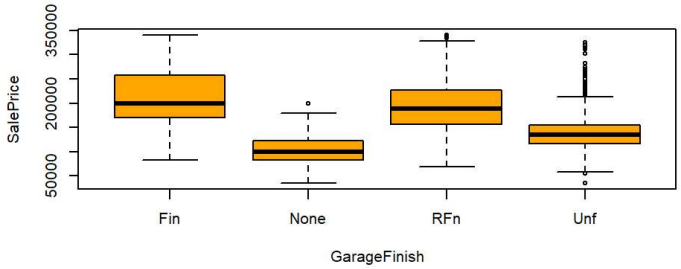

SalePrice
50000
200000
350000
Ex
Fa
Gd
TA
KitchenQual

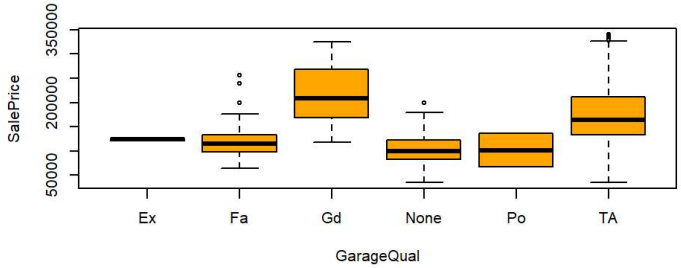

SalePrice
50000
200000
350000
Maj1
Maj2
Min1
Min2
Mod
Sev
Typ
Functional

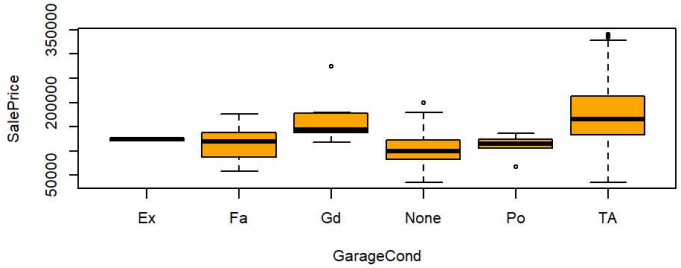

SalePrice
50000
200000
350000
Ex
Fa
Gd
None
Po
TA
FireplaceQu

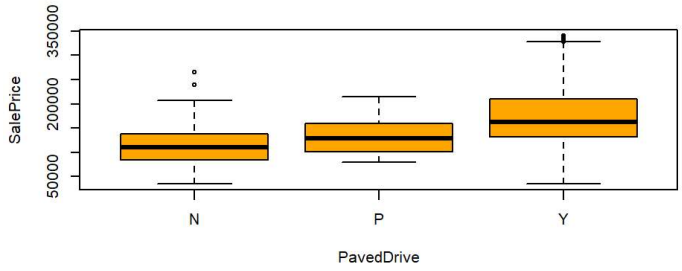

SalePrice
50000
200000
350000
2Types
Attchd
Basment
BuiltIn
CarPort
Detchd
None
GarageType

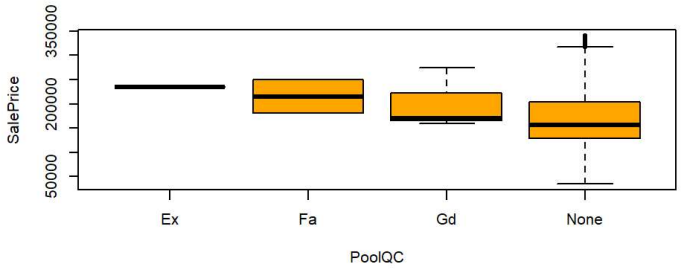

SalePrice
50000
200000
350000
1900
1921
1930
1939
1950
1959
1968
1977
1986
1995
2004
GarageYrBlt

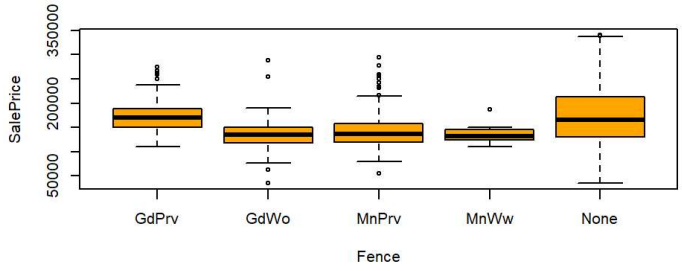

SalePrice
50000
200000
350000
Fin
None
RFn
Unf
GarageFinish

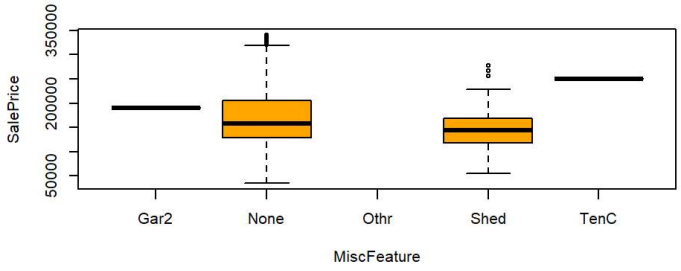

SalePrice
50000
200000
350000
Ex
Fa
Gd
None
Po
TA
GarageQual

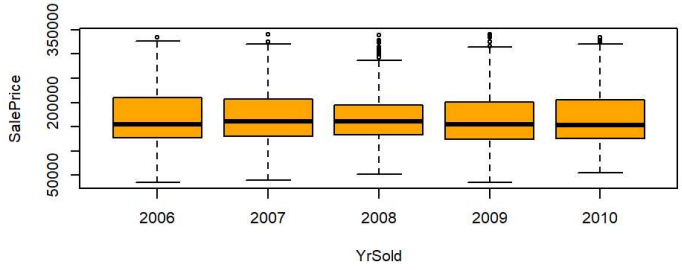

SalePrice
50000
200000
350000
Ex
Fa
Gd
None
Po
TA
GarageCond

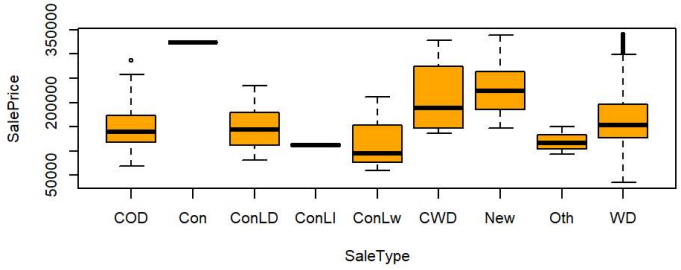












Surprisingly house price does not depend on the year house was sold from box plot above

From the above boxplots for factor variables `OverallQual` , `Neigboorhood` , and `ExterQual` impact `SalePrice` . But some of the `ExterQual` are collinear so don't include them in the model.

So finally we are done with variable selection. We select the below variables for our model:

`GrLivArea, GarageCars, FullBath, YearBuilt, TotalBsmtSF, Fireplaces, OpenPorchSF, Neigboorhood, OverallQual`

Results

Model Selection

Next we can move to model selection.

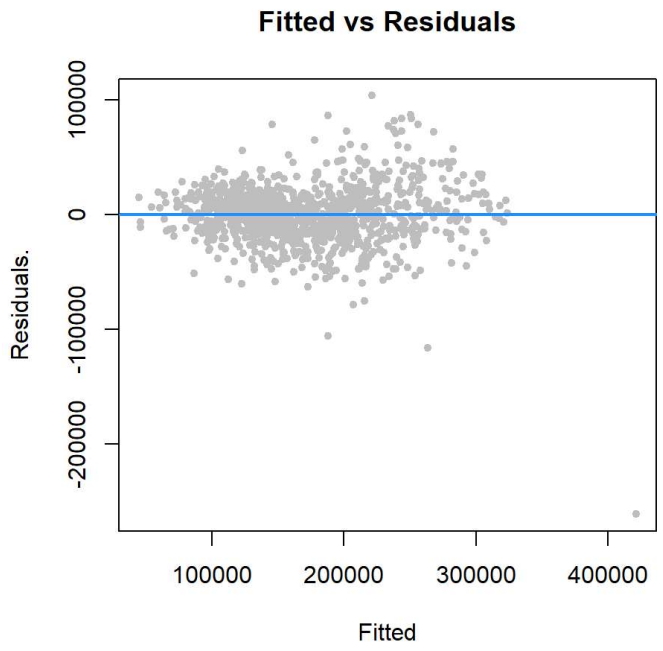
Model 1: Additive model using selected variables

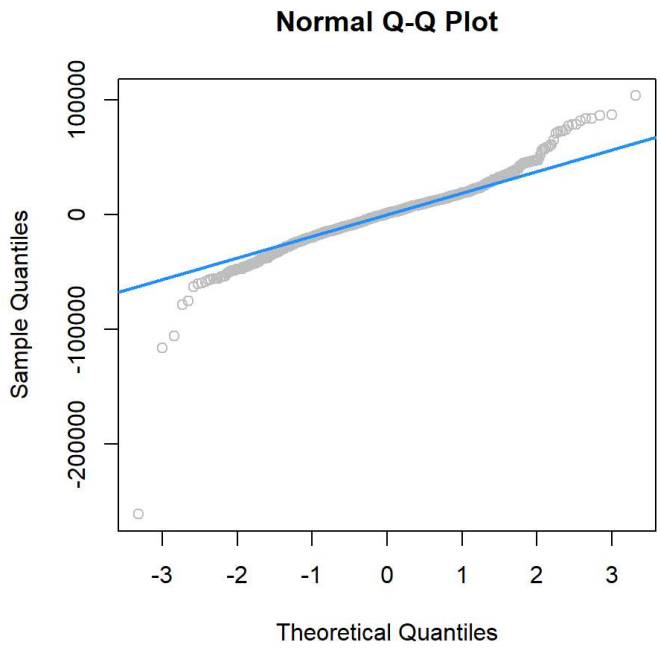
We will form an additive model from variables selected from our analysis done above.

```
model_sel_add = lm(SalePrice ~ GrLivArea + GarageCars + FullBath + YearBuilt
                               + TotalBsmtSF + Fireplaces + OpenPorchSF
                               + Neighborhood + OverallQual,
                               data = ames_trn_data)
```

Below is the result of diagnostics done on this model:

```
## [1] "LOOCV RMSE : 27707.404"
## [1] "Adjusted R2: 0.83"
## [1] "Test Error : 10.801"
## [1] "Num of predictors: 9"
## [1] "Num of paramaters: 41"
## [1] "BP test decision          : Constant Variance assumption not suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```





Both Shapiro-Wilk test and Normal Q-Q plot confirms that Normality assumption is suspected for this model.

BP test supports Constant Variance assumption for this model. But after inspecting the Fitted vs Residual plot we come to the conclusion that Constant variance assumption is suspected.

We can see that this model is doing better than the model with the numerical variables.

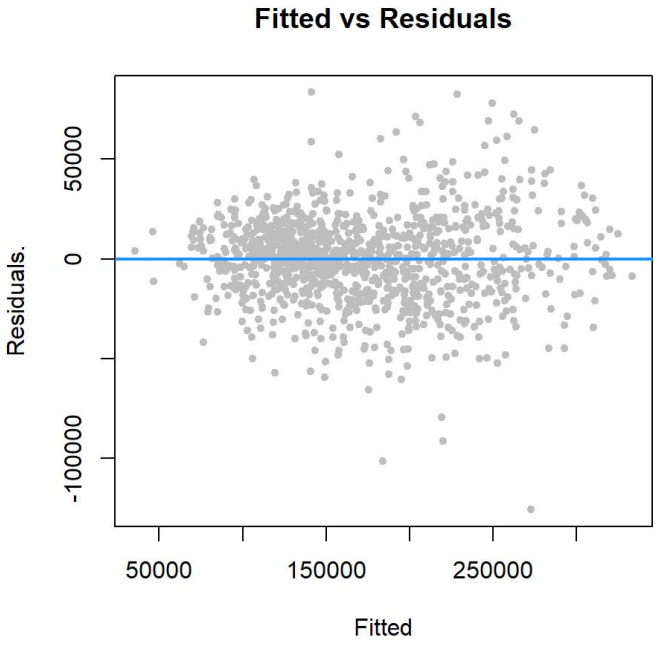
Model 2: Use BIC to select from 2-way interaction model using selected variables

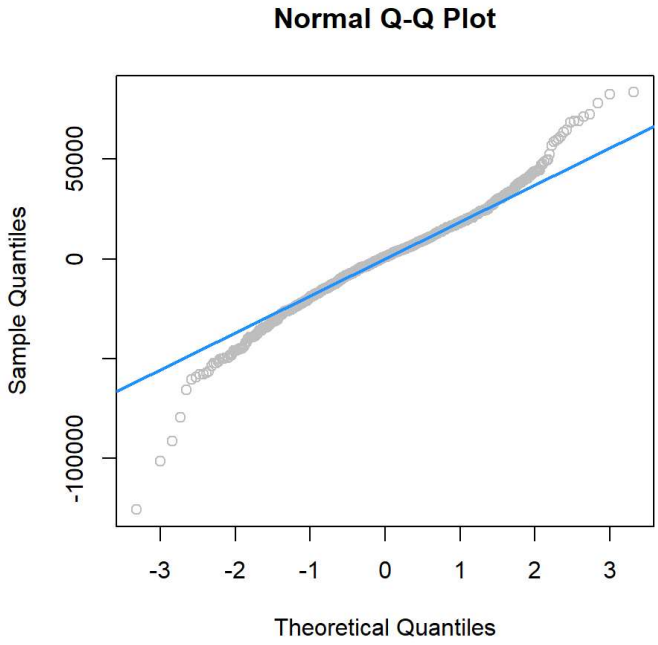
We will form 2-way interaction model using our selected variables. Then use BIC to select a smaller model.

```
model_sel_int = lm(
            SalePrice ~
            (GrLivArea + GarageCars + FullBath + YearBuilt
                        + TotalBsmtSF + Fireplaces + OpenPorchSF
                        + Neighborhood + OverallQual) ^ 2,
            data = ames_trn_data)
model_sel_int_bic = step(model_sel_int, direction = "backward", k = log(nrow(ames_trn_data)), trace
  =0)
```

Below is the result of diagnostics done on this smaller model:

```
## [1] "LOOCV RMSE : 26842.343"
## [1] "Adjusted R2: 0.865"
## [1] "Test Error : 11.142"
## [1] "Num of predictors: 9"
## [1] "Num of paramaters: 48"
## [1] "BP test decision          : Constant Variance suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```





With this model the LOOCV RMSE got better, but number of parameters increased and test error got worse.

Both Shapiro-Wilk test and Normal Q-Q plot confirms that Normality assumption is suspected for this model.

Both BP Test and Fitted vs Residual plot confirms that Constant Variance assumption is suspected for this model.

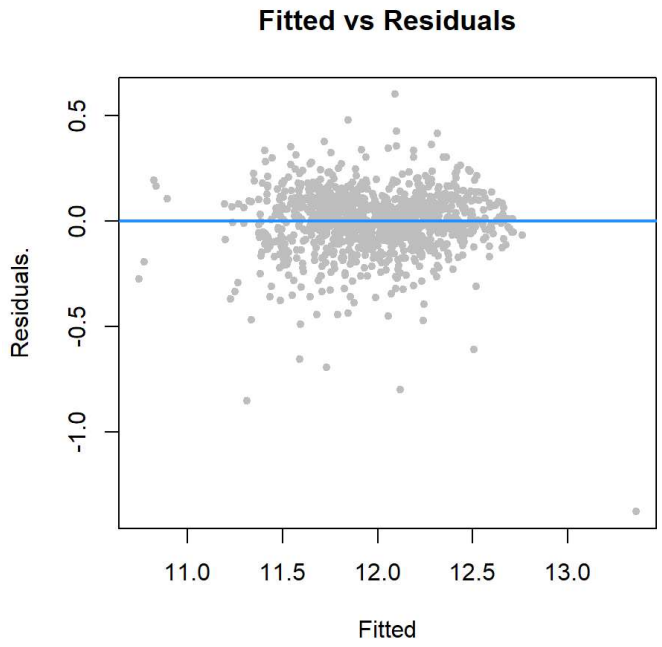
Model 3: Form a model similar to Model 1 with log transform on response

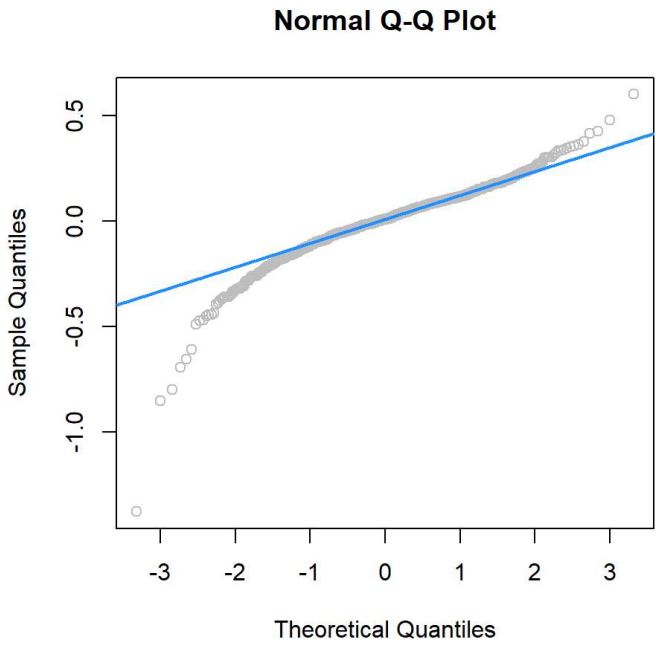
Here we form a new model to see if we can achieve constant variance and normality.

```
model_sel_add_log_resp = lm(
                  log(SalePrice) ~
                  GrLivArea + GarageCars + FullBath + YearBuilt
                            + TotalBsmtSF + Fireplaces + OpenPorchSF
                            + Neighborhood + OverallQual,
                  data=ames_trn_data)
```

Below is the result of diagnostics done on this model:

```
## [1] "LOOCV RMSE : 34282.536"
## [1] "Adjusted R2: 0.829"
## [1] "Test Error : 10.676"
## [1] "Num of predictors: 9"
## [1] "Num of paramaters: 41"
## [1] "BP test decision          : Constant Variance assumption not suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```





LOOCV RMSE of this model is worser than Model 1 and Model 2.

$Adjusted\ R^2$ is worse than Model 2 and almost same as Model 1.

Test error is better than Model 2 but comparable compared to Model 1.

Both Shapiro-Wilk test and Normal Q-Q plot confirms that Normality assumption is suspected for this model.

BP test supports Constant Variance assumption for this model. But after inspecting the Fitted vs Residual plot we come to the conclusion that Constant variance assumption is suspected.

Model 4: Use Model 3 after removing high influential and high leverage points

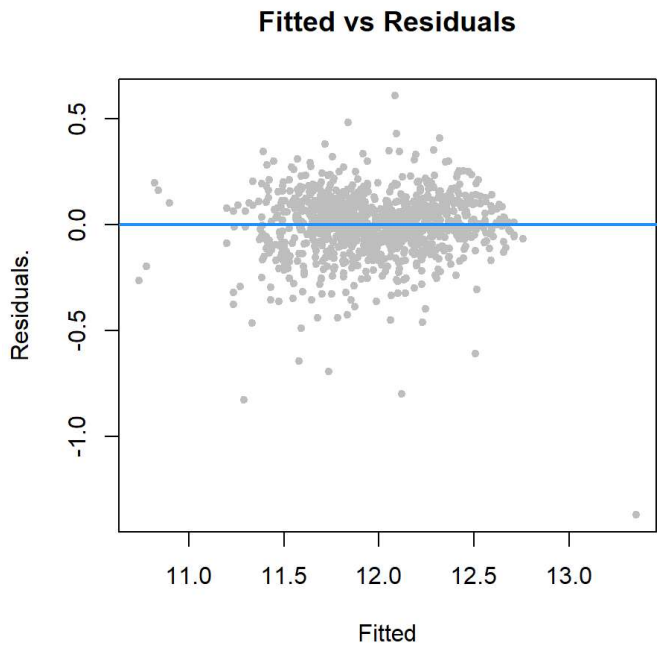
We are tring to build a model that satisfied linearity assumptions.

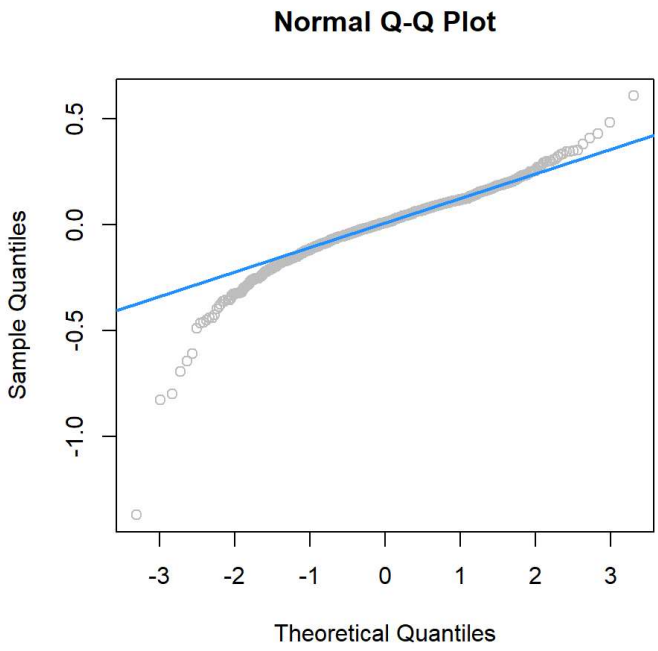
We will first remove highly influential points and test out using Model 3.

```
ames_trn_data_cd = cooks.distance(model_sel_add_log_resp)
ames_trn_data_removed_infl = ames_trn_data[!row.names(ames_trn_data)
                            %in% which(ames_trn_data_cd > 4 / length(ames_trn_data_cd)),]
model_log_resp_no_infl_points = lm(
                    log(SalePrice) ~
                    GrLivArea + GarageCars + FullBath + YearBuilt
                              + TotalBsmtSF + Fireplaces + OpenPorchSF
                              + Neighborhood + OverallQual,
                    data = ames_trn_data_removed_infl)
```

Below is the result of diagnostics done on this model:

```
## [1] "LOOCV RMSE : 34804.483"
## [1] "Adjusted R2: 0.829"
## [1] "Test Error : 10.693"
## [1] "Num of predictors: 9"
## [1] "Num of paramaters: 41"
## [1] "BP test decision          : Constant Variance assumption not suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```





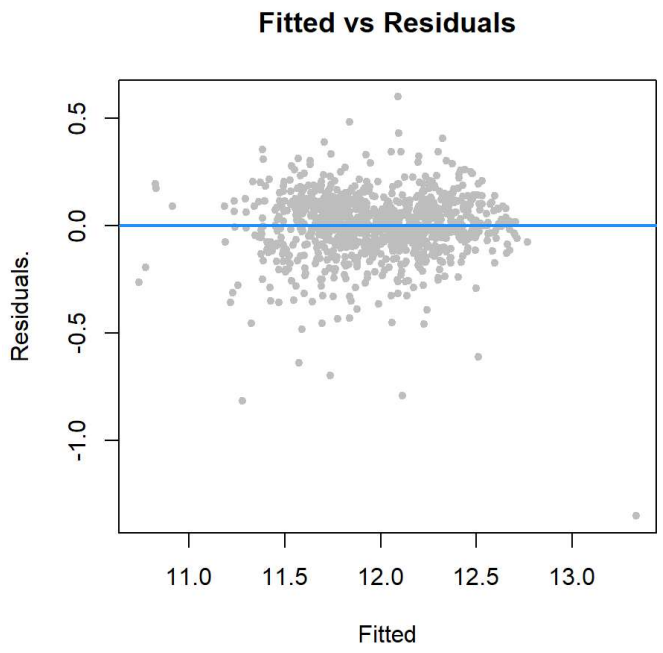
We can see that LOOCV RMSE of this model got worse.

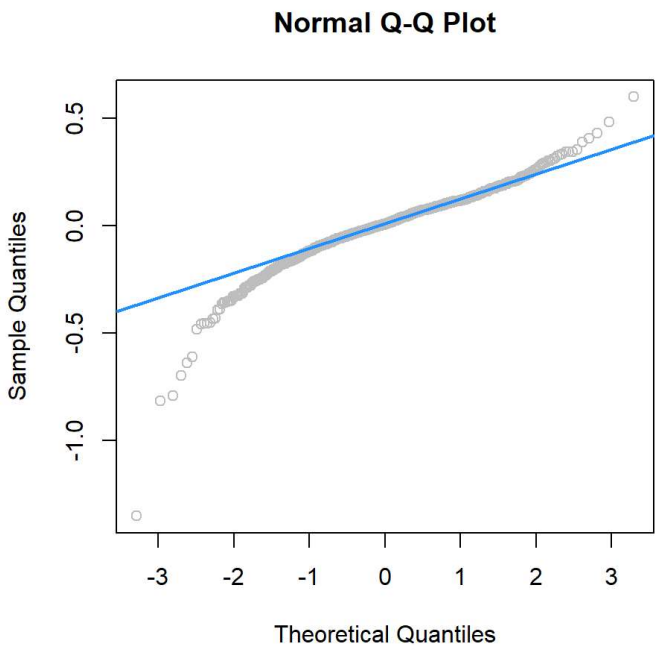
Lets remove high leverage points as well and test out using Model 3.

```
ames_trn_data_hatvals = hatvalues(model_log_resp_no_infl_points)
ames_trn_data_removed_infl_levr = ames_trn_data_removed_infl[!row.names(ames_trn_data_removed_infl)
                                  %in% which(ames_trn_data_hatvals > 2 * mean(ames_trn_data_hatval
  s)),]
model_no_infl_levr_points = lm(
                      log(SalePrice) ~
                      GrLivArea + GarageCars + FullBath + YearBuilt
                              + TotalBsmtSF + Fireplaces + OpenPorchSF
                              + Neighborhood + OverallQual,
                      data = ames_trn_data_removed_infl_levr)
```

Below is the result of diagnostics done on this model:

```
## [1] "LOOCV RMSE : 35348.641"
## [1] "Adjusted R2: 0.827"
## [1] "Test Error : 10.674"
## [1] "Num of predictors: 9"
## [1] "Num of paramaters: 41"
## [1] "BP test decision          : Constant Variance assumption not suspect"
## [1] "Shapiro Wilk test decision: Normality assumption suspect"
```





After removing high influenetial and leverage points Model 4 still does not satisy linearity assumptions.

Let us compare all our models in the below table.

Model Comparison

Model	Training LOOCV RMSE	Training Adjusted R^2	Test Error	Num of parameters	SW decision	BP decision

Model	Training LOOCV RMSE	Training Adjusted R^2	Test Error	Num of parameters	SW decision	BP decision
Model#1: Additive model	27707	0.830	10.801	41	Normality assumption suspect	Constant Variance assumption not suspect
Model#2: Selected using BIC from 2-way interaction model	26842	0.865	11.142	48	Normality assumption suspect	Constant Variance suspect
Model#3: Additive model with Log Transform of Response	34283	0.829	10.676	41	Normality assumption suspect	Constant Variance assumption not suspect
Model#4: High influential and high leverage points removed from Model 3	35349	0.827	10.674	41	Normality assumption suspect	Constant Variance assumption not suspect

Based on the above comparison we feel that `Model 1` is the best model which less complex, with better $LOOCV\ RMSE$ and $Adjusted\ R^2$ and reasonable test error value.

Lets plot Predicted Vs Actual for the best model.

Predicted Vs Actual plot indicates that this model does a good job in estimating prices.

Discussion

The model we picked for the best is the lowest in LOOCV RMSE and the highest in $adjusted\ R^2$ so that it does not have any collinearity and gets one of the lowest prediction error for test sets. Following would be our final model:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \epsilon,$$

where

- Y is `SalePrice`
- x_1 is `GrLivArea`
- x_2 is `GarageCars`
- x_3 is `FullBath`
- x_4 is `YearBuilt`
- x_5 is `TotalBsmtSF`
- x_6 is `Fireplaces`
- x_7 is `OpenPorchSF`
- x_8 is `Neighborhood`
- x_9 is `OverallQual`.

These are the coefficients of the model selected:

```
##         (Intercept)           GrLivArea          GarageCars
##         -603590.893              35.363           10151.680
##            FullBath           YearBuilt         TotalBsmtSF
##            1562.315             320.805              10.956
##          Fireplaces         OpenPorchSF NeighborhoodBlueste
##            8366.858              43.270          -12984.798
##  NeighborhoodBrDale NeighborhoodBrkSide NeighborhoodClearCr
##          -15670.153           11760.369           39173.386
## NeighborhoodCollgCr NeighborhoodCrawfor NeighborhoodEdwards
##           17965.801           45288.317            2024.119
## NeighborhoodGilbert  NeighborhoodIDOTRR NeighborhoodMeadowV
##            9901.467           -1621.617          -13782.314
## NeighborhoodMitchel   NeighborhoodNAmes NeighborhoodNoRidge
##            8677.243           10428.063           43844.182
## NeighborhoodNPkVill NeighborhoodNridgHt  NeighborhoodNWAmes
##           -5669.000           35885.299           12122.234
## NeighborhoodOldTown  NeighborhoodSawyer NeighborhoodSawyerW
##            -471.749            8981.976           12314.186
## NeighborhoodSomerst NeighborhoodStoneBr   NeighborhoodSWISU
##           21015.951           29178.574            8157.530
##  NeighborhoodTimber NeighborhoodVeenker        OverallQual2
##           24565.020           47407.116           -6120.329
##        OverallQual3        OverallQual4        OverallQual5
##            6854.242           18282.772           27031.375
##        OverallQual6        OverallQual7        OverallQual8
##           31630.536           48687.727           75444.893
##        OverallQual9       OverallQual10
##          110606.401           51286.324
```

Evaluating coefficient of the selected model we see the change in price for `overalQual` of 1 to `overallQual` of 9 is whooping 110606.4. Hence has most influence on price sold. Also, it appears Nieghborhood `Crawfor` is possibly a affluent neighborhood, owning a house there relative to a house in `Blmgtn` increases sold house price on average by 45288.32

From AMES IOWA there were some interesting observation observed. From the analysis above it could be seen in AMES IOWA price of the house is determined primarily with factors `GrLivArea` , `GarageCars` , `FullBath` , `YearBuilt` , `TotalBsmtSF` , `Fireplaces` , `OpenPorchSF` , `Neighborhood` and `OverallQual` . Suprisingly, it was observed house prices were not influnenced the by the year house was sold, this information maybe useful for an investor who maybe interested in growth.

The reason why we reject other models and choose to lean towards additive model `Model 1` are: Linear assumptions are violated in the model even with log transformation of response. However, `Model 1` and `Model 3` are good for prediction. As shown earlier ~10.801% errors is in `Model 1` and ~10.676% in `Model 3` . `Model 2` is ignored due to high paramters with no major improvements in other criteria.

Future implementation?

As in the previous context we talked about how we find our best model, since we have some knowledges towards the machine learning implementation. We could use different algorithms to train the model, such as random forest, gradient boost, xgboost and even deep learning libraries that train neurons for forward propogation and backward propogation to optimize residuals of each layer. Thus, there are multiple algorithms out there for us to discover. However, it would be our responsibility to actually look into each of them, figure out the accuracy and the mechanism for that algorithm, then polish, compare and discuss about those implementations. How can we actually make this solution better, and what is the best approach to it. That is our journey towards an optimize resolution.

Appendix

This project is inspired by Kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)
