Bowen Liu(Lec2) Stats 101A Project Report

Abstract

The purpose of this project is building a model to predict the house prices in Ames, Iowa. In this project, I am dealing with four procedures: replace missing values, pick useful predictors, improve R square and diagnostic.

As the result, R^2 of my training data is 0.9385, R^2 of the testing data is 0.93453 by using 22 predictors including interactions. My Kaggle name is Bowen Liu, and the final ranking is 28 based on the “Public and Private Score”.

Introduction

We are given a training data with 80 variables(1 Response and 79 predictors describing different aspects of a house) with 2500 observations, and my goal is making a simple and effective model to predict the response based on training data, then applying it to the testing data which has predictors only to predict the missing response value.

Methodology

1. Removing all NAs

First of all, I divide the training data into numerical parts and categorical parts.

For numerical part, I use for loop to replace NAs with the median(column wise).

And for categorical part, I replace NAs with different types of observations(Not NAs) by their proportion(column wise). Specifically, I use the function prop.table() to get the proportions of each category(known), then replaced NAs with these known observations according to their proportion randomly. In the end, I combined these two processed datasets by using cbind()to get a new and complete training data without NAs.

2. Picking the right predictors

Firstly, for numerical predictors, I use cor() to get the matrix of correlations. Since I only need the correlation between predictors and Saleprice(response), I only output the 38th column of the matrix, and select the names of predictors with high correlation.(standard is >0.3)

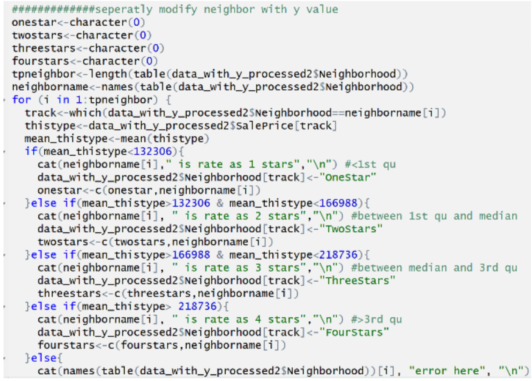
Moreover, for categorical predictors, I create a for loop, lm() Saleprice with one categorical predictor each interation, and print out the R^2, then pick the names of categorical predictors with high R^2.(standard is >0.1)

Next step is checking the VIF to see if I have to deal with the multicollinearity problem(vif>5). Fortunately, my vifs that are greater 5 are all from numerical group, thus I use the correlation matrix I created before in order to check which predictors are highly correlated with another, then delete the one with lower correlation with the response. On the other hand, I also observe the summary() of the model, predictors without significance will also be deleted in order to have a simpler model. Moreover, I also include a BIC function to simplify the number of predictors I use.

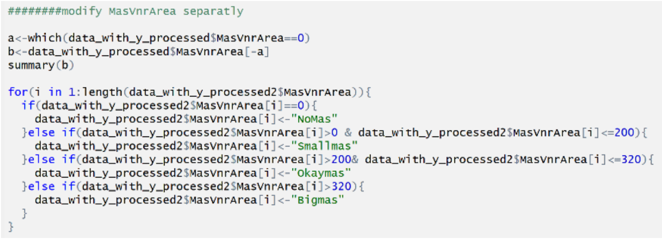
3. Improve R square

The basic idea is that among 2500 observations under each predictors, there are too many categories(like Neighborhood) or wide range of numbers(like TotalBsmtsf), making the model less effective to predict the SalePrice, thus I did some modification as follow:

For categorical predictors with many categories, I firstly summary(Saleprice), and then set up four levels as “Onestar”, “Twostar”, “Threestar” and “Fourstar” represents price below first quantile, first quantile to median, median to 3rd quantile and above 3rd quantile. Under each predictor, calculating the mean of each type, deciding which level this type should be and replace observations in this type totally with the name of this type.(example as picture below left)



```
#############seperatly modify neighbor with y value
onestar<-character(0)
twostars<-character(0)
threestars<-character(0)
fourstars<-character(0)
tpneighbor<-length(table(data_with_y_processed2$Neighborhood))
neighborname<-names(table(data_with_y_processed2$Neighborhood))
for (i in 1:tpneighbor) {
  track<-which(data_with_y_processed2$Neighborhood==neighborname[i])
  thistype<-data_with_y_processed2$SalePrice[track]
  mean_thistype<-mean(thistype)
  if(mean_thistype<132306){
    cat(neighborname[i]," is rate as 1 stars","\n") #<1st qu
    data_with_y_processed2$Neighborhood[track]<-"OneStar"
    onestar<-c(onestar,neighborname[i])
  }else if(mean_thistype>132306 & mean_thistype<166988){
    cat(neighborname[i], " is rate as 2 stars","\n") #between 1st qu and median
    data_with_y_processed2$Neighborhood[track]<-"TwoStars"
    twostars<-c(twostars,neighborname[i])
  }else if(mean_thistype>166988 & mean_thistype<218736){
    cat(neighborname[i], " is rate as 3 stars","\n") #between median and 3rd qu
    data_with_y_processed2$Neighborhood[track]<-"ThreeStars"
    threestars<-c(threestars,neighborname[i])
  }else if(mean_thistype> 218736){
    cat(neighborname[i], " is rate as 4 stars","\n") #>3rd qu
    data_with_y_processed2$Neighborhood[track]<-"FourStars"
    fourstars<-c(fourstars,neighborname[i])
  }else{
    cat(names(table(data_with_y_processed2$Neighborhood))[i], "error here", "\n")
```



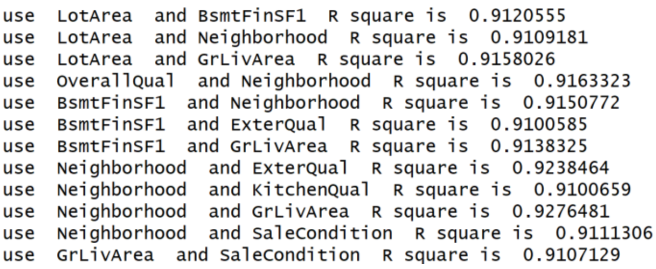
```
########modify MasVnrArea separatly

a<-which(data_with_y_processed$MasVnrArea==0)
b<-data_with_y_processed$MasVnrArea[-a]
summary(b)

for(i in 1:length(data_with_y_processed2$MasVnrArea)){
  if(data_with_y_processed2$MasVnrArea[i]==0){
    data_with_y_processed2$MasVnrArea[i]<-"NoMas"
  }else if(data_with_y_processed2$MasVnrArea[i]>0 & data_with_y_processed2$MasVnrArea[i]<=200){
    data_with_y_processed2$MasVnrArea[i]<-"Smallmas"
  }else if(data_with_y_processed2$MasVnrArea[i]>200& data_with_y_processed2$MasVnrArea[i]<=320){
    data_with_y_processed2$MasVnrArea[i]<-"Okaymas"
  }else if(data_with_y_processed2$MasVnrArea[i]>320){
    data_with_y_processed2$MasVnrArea[i]<-"Bigmas"
  }
}
```

For numerical data set(chosen predictors), firstly I will replace the 0 with “noXXX”, then summary the result numerical data(column wise), set up another three levels, 1st quantile to median, “lessXXX”, median to 3rd quantile “goodXXXX”, above 3rd quantile “greatXXXX” so that the data can be more effective to predict the price.(example as picture above right)

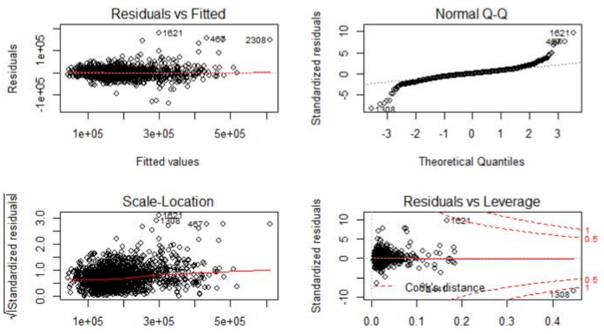
Another effective way is creating **interaction** between the predictors I just choose. Therefore, I create a for loop, and randomly combine two of the selected predictors in an interaction format, adding to the model, then run the model and print out their name & R square to see the which interaction should be add. (picture below filter only the R square >0.91)

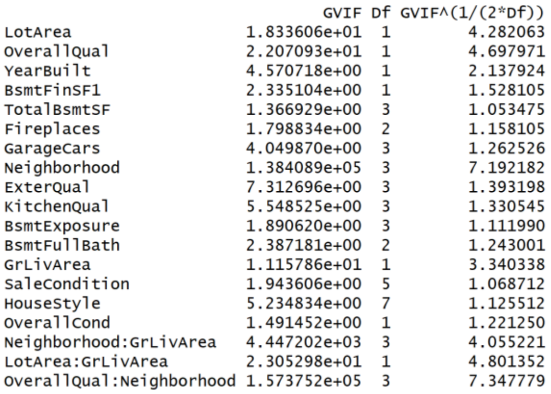


```
use  LotArea  and BsmtFinSF1  R square is  0.9120555
use  LotArea  and Neighborhood  R square is  0.9109181
use  LotArea  and GrLivArea  R square is  0.9158026
use  OverallQual  and Neighborhood  R square is  0.9163323
use  BsmtFinSF1  and Neighborhood  R square is  0.9150772
use  BsmtFinSF1  and ExterQual  R square is  0.9100585
use  BsmtFinSF1  and GrLivArea  R square is  0.9138325
use  Neighborhood  and ExterQual  R square is  0.9238464
use  Neighborhood  and KitchenQual  R square is  0.9100659
use  Neighborhood  and GrLivArea  R square is  0.9276481
use  Neighborhood  and SaleCondition  R square is  0.9111306
use  GrLivArea  and SaleCondition  R square is  0.9107129
```

4. Diagnostic

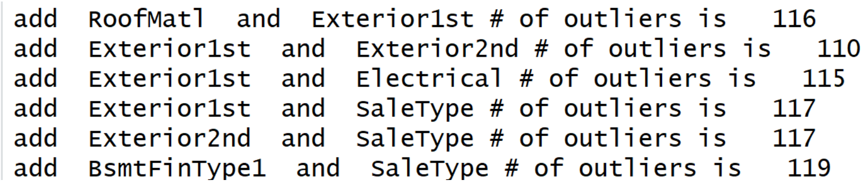
By checking the plot of the model, the assumption of linearity as well as constant variance are achieved in my model, the only problem is that the number of outlier. I tried the transformation, but it does not work well because the 3rd plot looks more curved and the number of outliers did not reduced either. (diagnostic plots on the left, vif on the right)





```
                              GVIF Df GVIF^(1/(2*Df))
LotArea               1.833606e+01  1        4.282063
OverallQual           2.207093e+01  1        4.697971
YearBuilt             4.570718e+00  1        2.137924
BsmtFinSF1            2.335104e+00  1        1.528105
TotalBsmtSF           1.366929e+00  3        1.053475
Fireplaces            1.798834e+00  2        1.158105
GarageCars            4.049870e+00  3        1.262526
Neighborhood          1.384089e+05  3        7.192182
ExterQual             7.312696e+00  3        1.393198
KitchenQual           5.548525e+00  3        1.330545
BsmtExposure          1.890620e+00  3        1.111990
BsmtFullBath          2.387181e+00  2        1.243001
GrLivArea             1.115786e+01  1        3.340338
SaleCondition         1.943606e+00  5        1.068712
HouseStyle            5.234834e+00  7        1.125512
OverallCond           1.491452e+00  1        1.221250
Neighborhood:GrLivArea 4.447202e+03  3        4.055221
LotArea:GrLivArea     2.305298e+01  1        4.801352
OverallQual:Neighborhood 1.573752e+05  3        7.347779
```

In order to reduce outliers, what I do is using a for loop again by randomly interacting two non-selected predictors and add it to the model, then calculate the number of outliers, and it turns out to be a significant drop with the “Exterior2nd*Exterior1st”, the number of outliers reduce from 147 to 110, and R^2 increases also



```
add  RoofMatl  and  Exterior1st # of outliers is   116
add  Exterior1st  and  Exterior2nd # of outliers is   110
add  Exterior1st  and  Electrical # of outliers is   115
add  Exterior1st  and  SaleType # of outliers is   117
add  Exterior2nd  and  SaleType # of outliers is   117
add  BsmtFinType1  and  SaleType # of outliers is   119
```

Result:

As the result, I used totally 22 predictors and get a R^2 of 0.9385, in addition, the plots of the model also look good.

Discussion:

One possible way to make my model better is creating new predictors by combing predictors describing similar aspects of a house based on my selected predictors, because I only modify the dataset by reducing the range of numbers(under numerical predictors) and number of categories(under categorical predictors). In this way, I may decrease the number of predictors and increase R^2 of the model. On the other hand, I think variables such as Crimes, Transport, Mall and Supermarket, which are describing the life convenience of the house should also be included in the data so that we can have a more accurate prediction.

Limitation:

There are two major limitation of the model I made, the first one is the number of outliers because the number of outlies are still around 100. Comparing the predicted SalePrice using this model with the real SalePrice in the same training data, I found that I was still underpaying some expensive houses, which leads to outliers.

The second limitation is the number of predictors(22) which a little bit more than I expected, but I could not reduce any of them since they are all statistically significant.

In a conclusion, I think my model includes nearly all possible statistically significant predictors and makes a decent R^2 predicting the SalePrice. However, modifying the data and picking all possible predictors are not enough, which needs a further work creating new datasets that have higher correlation with SalePrice to make the model simpler and better.

Reference:

Simon J. Sheather, “A Modern Approach to Regression with R”, Springer Inc, 2008, Accessed 3/23/2019.