

House Prices: Advanced Regression Techniques

組員：
統計四 陳庭偉
統碩一 陳柏勳
統碩一 楊博安
統碩一 林威均
資管碩二 周平

Contents

1. Data Information
2. Data Preprocessing
3. EDA
4. Modeling
 - (1) Random Forest
 - (2) XGBoost
 - (3) Support Vector Regression
5. Reference



Data Information

What is our goal?

- Data Description and Our Goal : Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.
- With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges us to predict the final price of each home.
- We assume that we are a bank, so we predict the final price of each home because of investment.

Data Information

The data is divided into the train and test, the train dataset has 1460, the test dataset has 1459

Train dataset:

id	MSSubClass	MSZoning	LotFrontage	SaleCondition	SalePrice
1	60	RL	65	Normal	208500
2	20	RL	80	Normal	181500
.
.
.
1459	20	RL	68	Normal	142125
1460	20	RL	75	Normal	147500

Data Information

Test dataset:

id	MSSubClass	MSZoning	LotFrontage	SaleCondition
1461	20	RH	80	Normal
1462	20	RL	81	Normal
.
.
.
.
.
2918	85	RL	62	MnPrv
2919	60	RL	74	NA

Data Information

Attribute Information: Due to the large number of variables, write the details into the PDF.



Attribute-Information.pdf

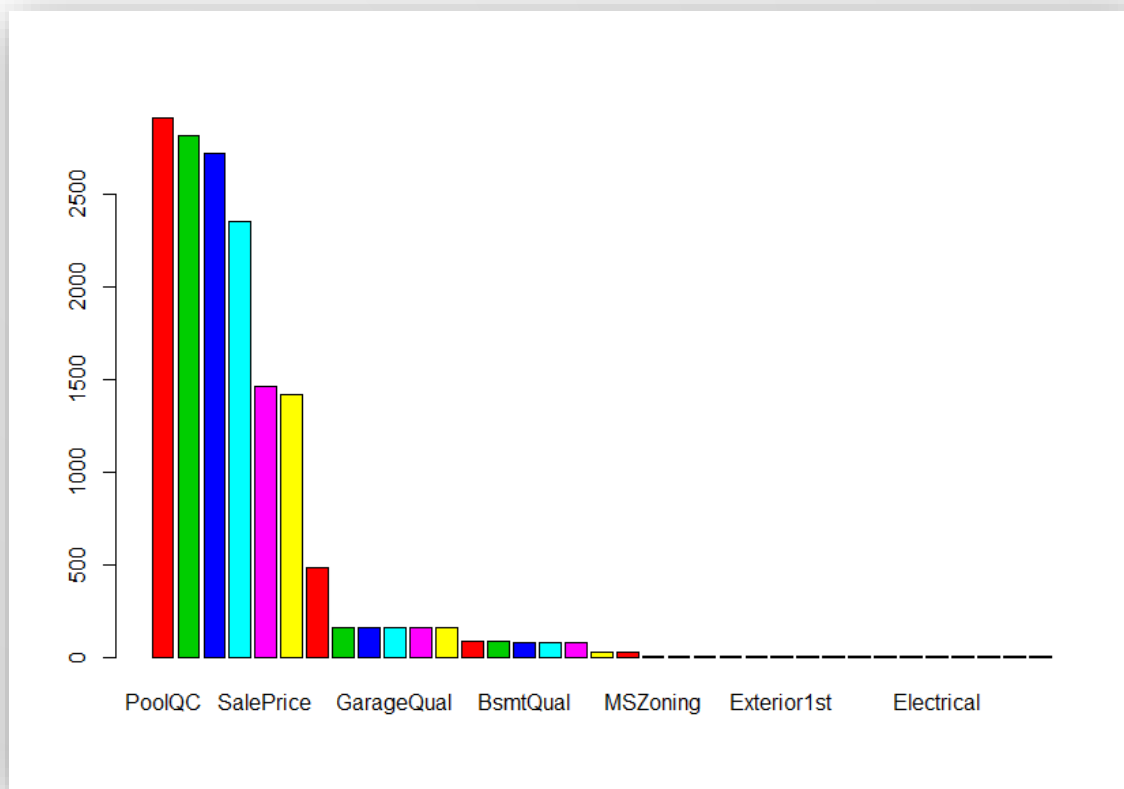
The background is a solid orange color. In the top-left corner, there are three vertical bars of varying heights, each composed of three overlapping circles. In the bottom-right corner, there are four vertical bars of increasing height, each also composed of three overlapping circles.

Data Preprocessing

Data Preprocessing

首先把訓練集(train dataset)跟測試集(test dataset)合併，所以前1460筆是訓練集的資料，後面的則是測試集。

圖為各項變數的NA數量：



Data Preprocessing: Missing Value

根據變數狀況，我們做了：

變數名稱	狀況	處理方法
PoolQC, MiscFeature, Alley, Fence, FireplaceQu	缺失值過多	刪除
Utilities	幾乎是同一個值 (AllPub)	刪除
GarageType, GarageQual, GarageCond, GarageFinish, BsmtExposure, BsmtFinType2, BsmtQual, BsmtCond, BsmtFinType1	沒有這項設施	修改成None。

此外，針對變數缺失沒有這麼嚴重的變數，我們採用了KNN的方式進行插補。

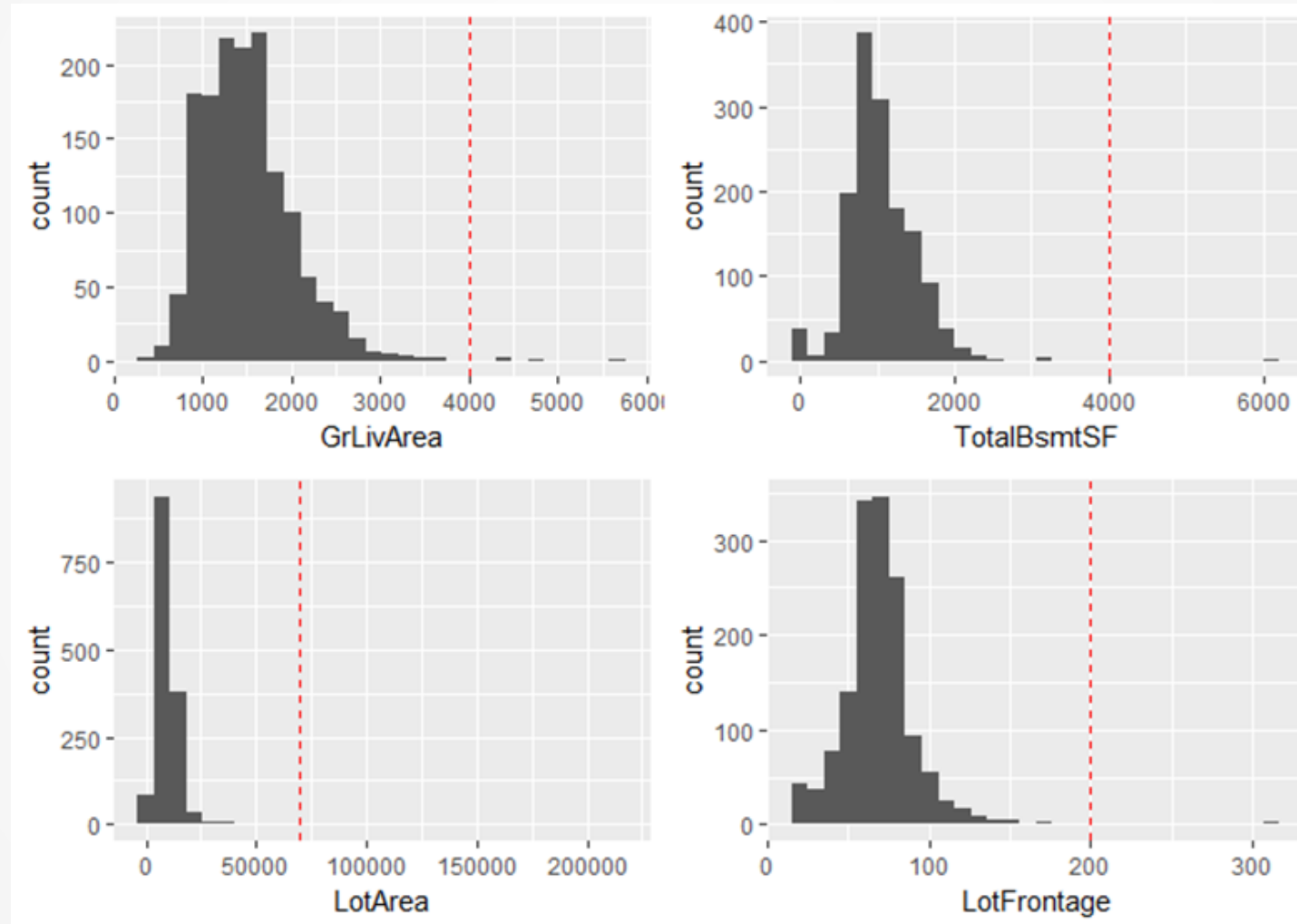
Data Preprocessing : Order Variable

針對次序(order)變數，我們做了：

變數名稱	資料變更
LotShape	IR3=0,IR2=1,IR1=2,Reg=3
LandContour	Low=0,HLS=1,Bnk=2,Lvl=3
LandSlope	Sev=0,Mod=1,Gtl=2
ExterQual,ExterCond,BsmtQual,BsmtCond,HeatingQC,KitchenQual,GarageQual,GarageCond	None=0,Po=1,Fa=2,TA=3,Gd=4,Ex=5
BsmtExposure	None=0,No=1,Mn=2,Av=3,Gd=4

變數名稱	資料變更
BsmtFinType1,BsmtFinType2	None=0,Unf=1,LwQ=2,Rec=3,BLQ=4,ALQ=5,GLQ=6
CentralAir	N=0,Y=1
Functional	Sal=0,Sev=1,Maj2=2,Maj1=3,Mod=4,Min2=5,Min1=6,Typ=7
GarageFinish	None=0,Unf=1,RFn=2,Fin=3
PavedDrive	N=0,P=1,Y=2

Data Preprocessing : Handling Outlier

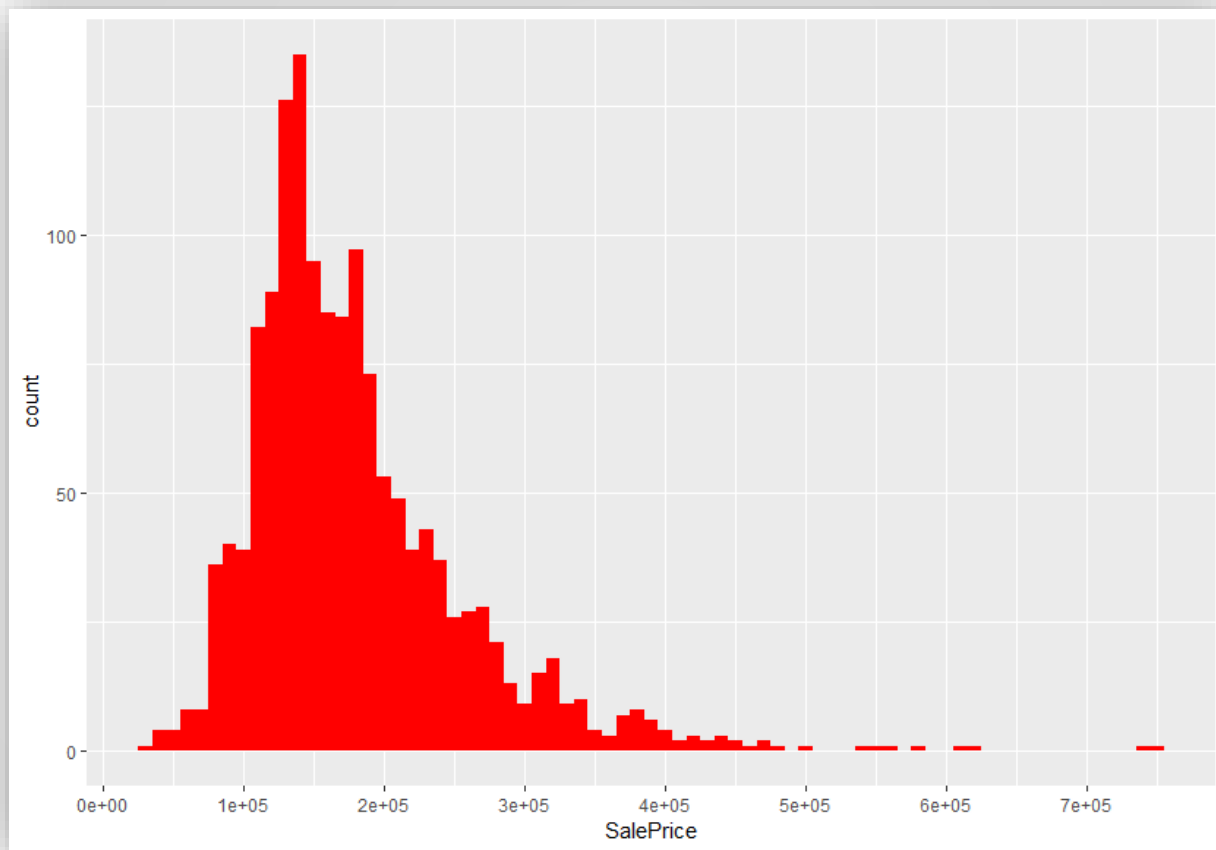




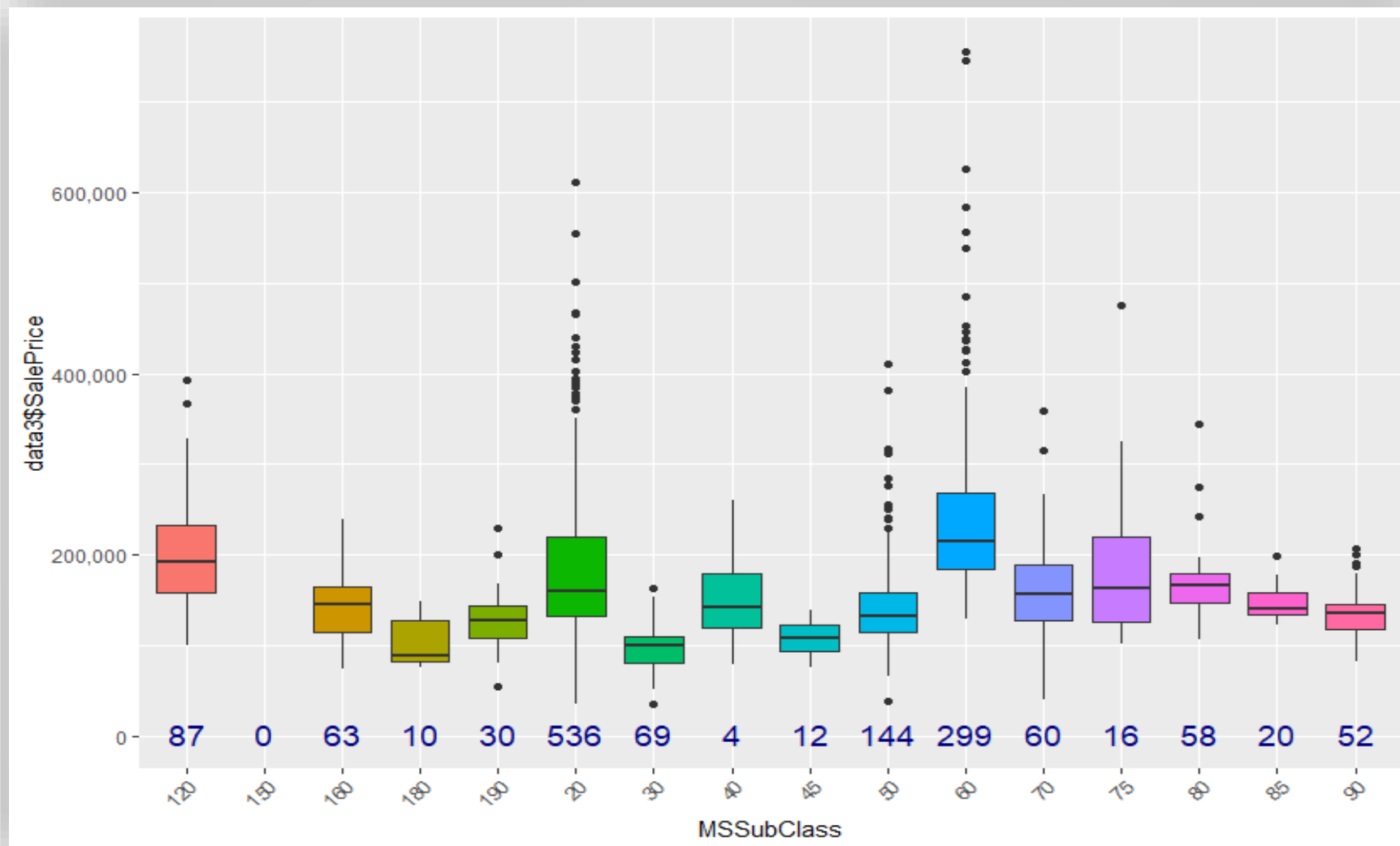
EDA

EDA

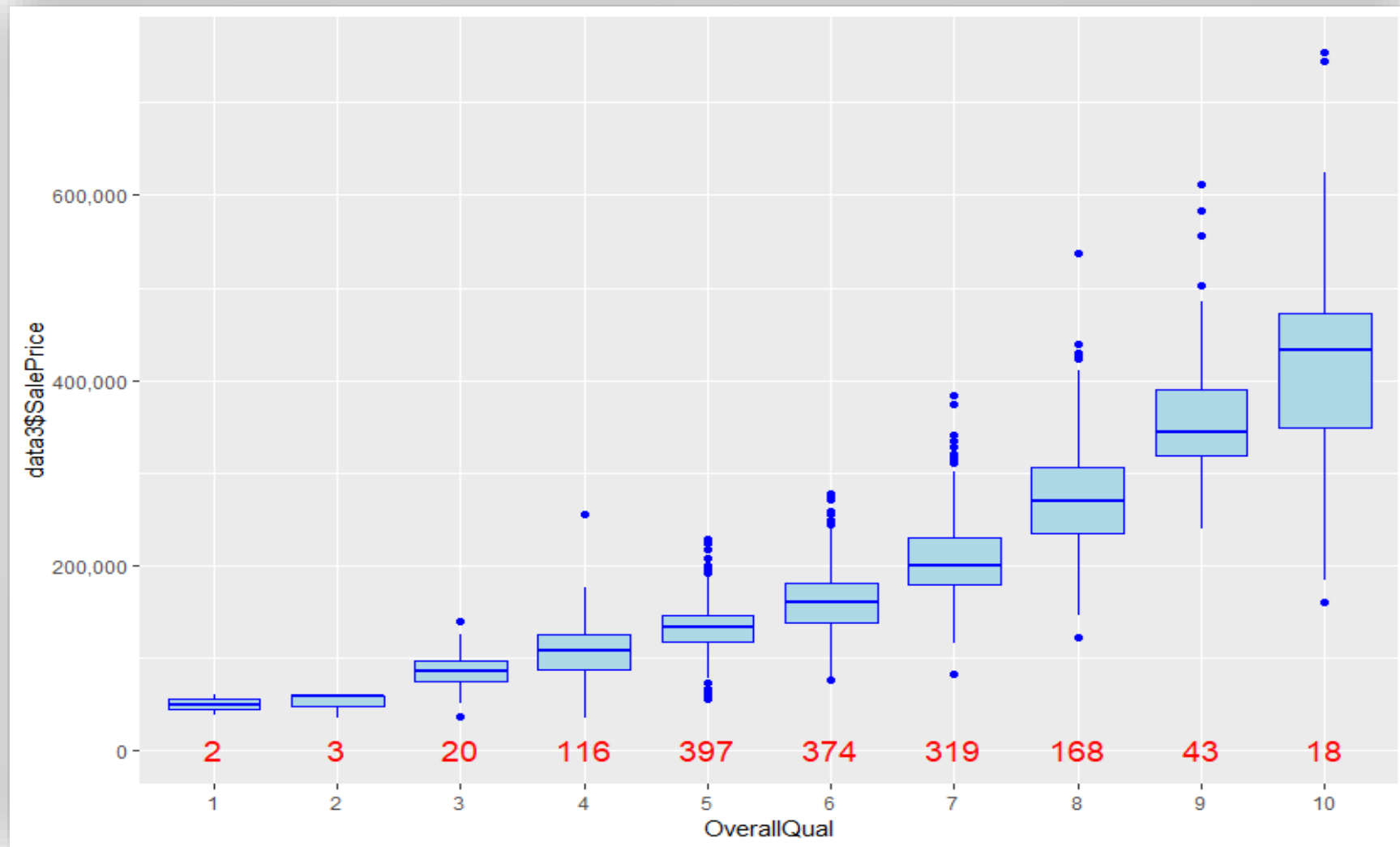
接下來看一下SalePrice的分佈，還有一些變數跟SalePrice的箱型圖：



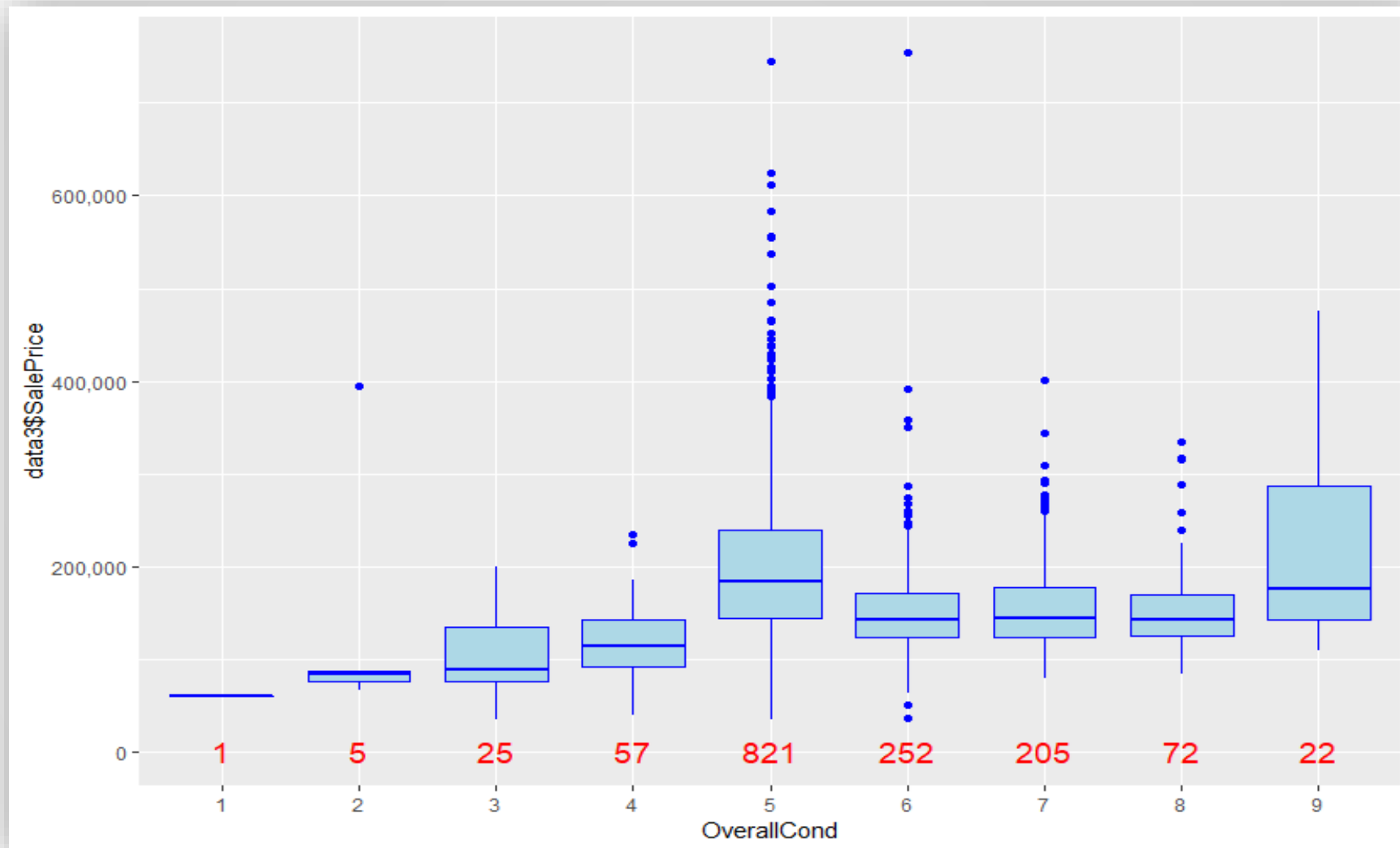
EDA



EDA

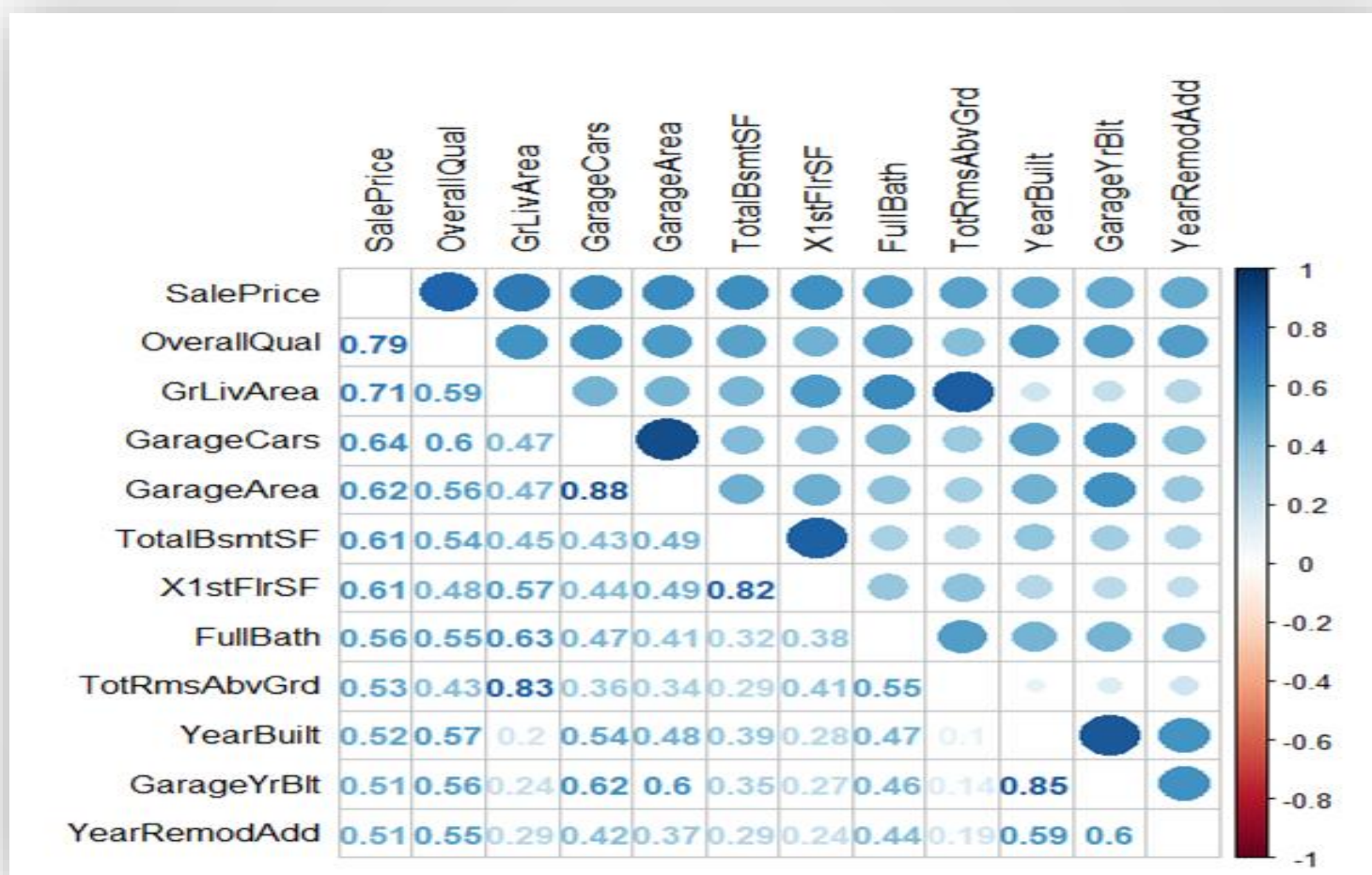


EDA



EDA

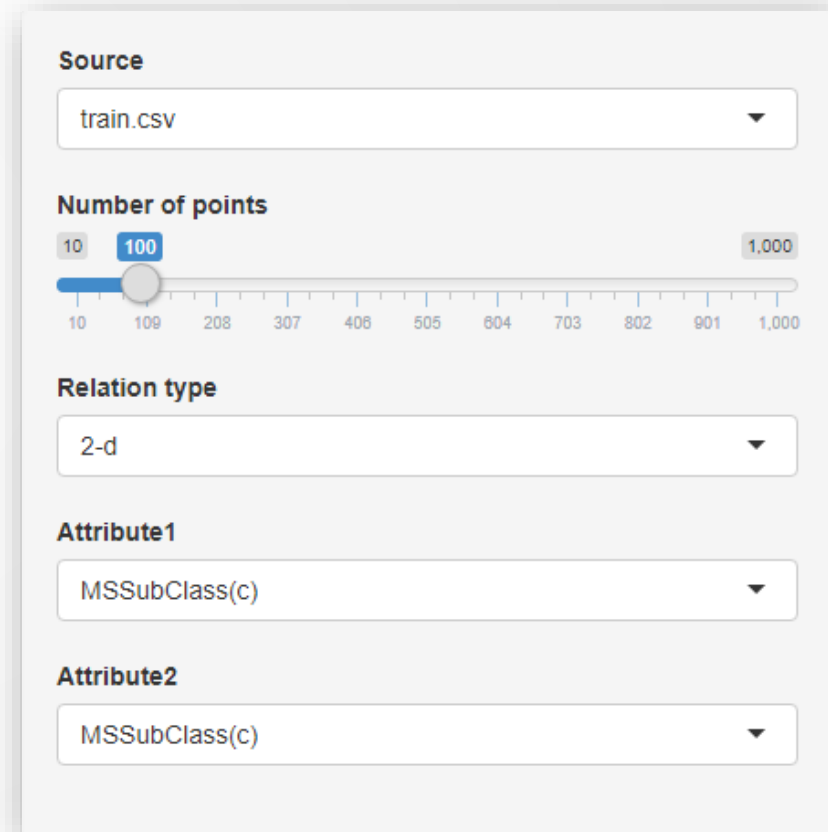
再來我們看一下相關係數大於0.5的相關矩陣，我們把它可視化。



Visualization with Shiny

Control Panel Setting:

1. Source: specify data source(train/test)
2. Sample Point: specify sample point to display
3. Relation Type: specify distribution type(1d/2d)
4. Attribute1: specify first feature to display
5. Attribute2: specify second feature(if needed)

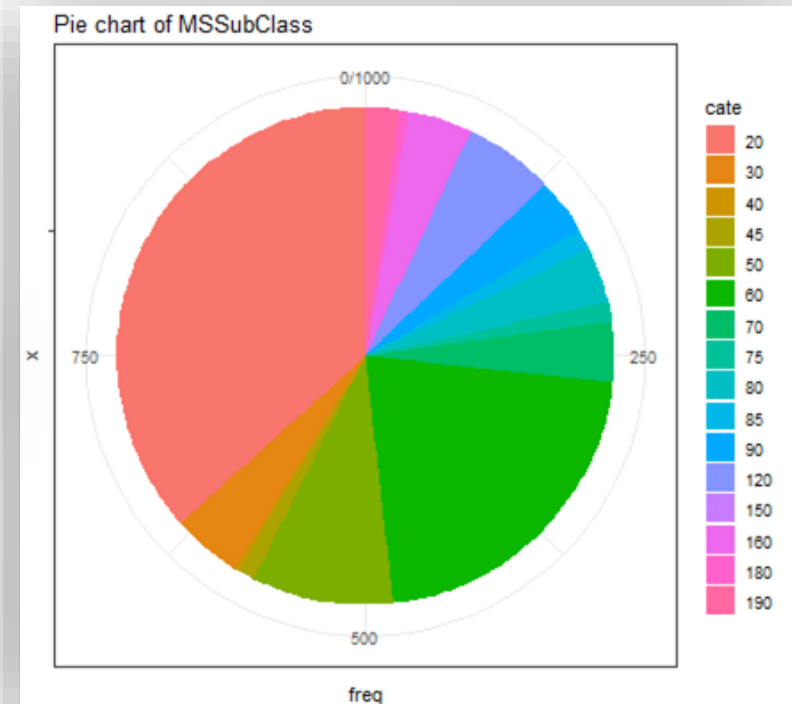
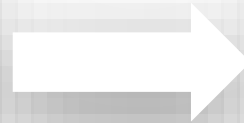
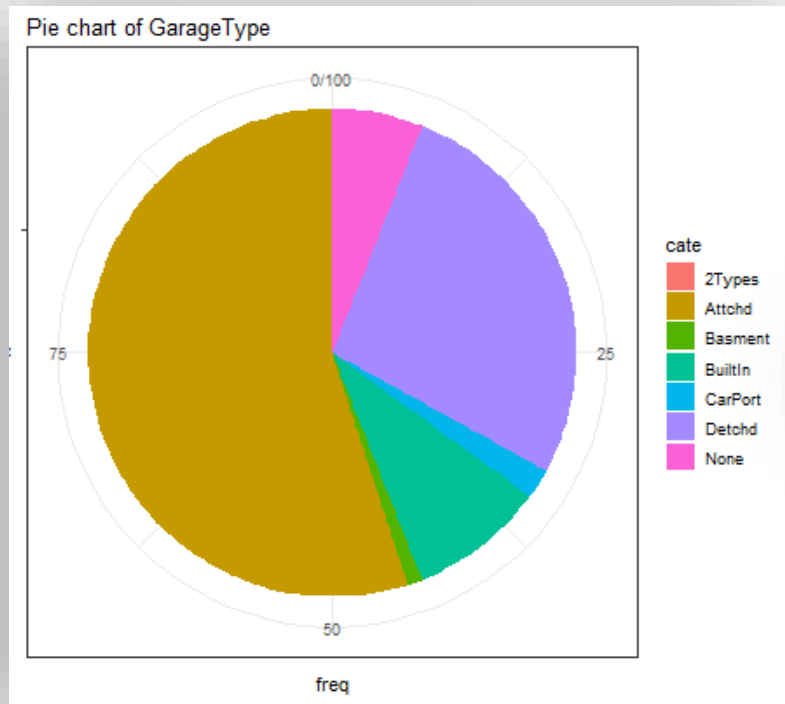


The image shows a Shiny control panel with the following settings:

- Source:** train.csv
- Number of points:** A slider set to 100, with a range from 10 to 1,000. The slider has a blue bar and a grey circle.
- Relation type:** 2-d
- Attribute1:** MSSubClass(c)
- Attribute2:** MSSubClass(c)

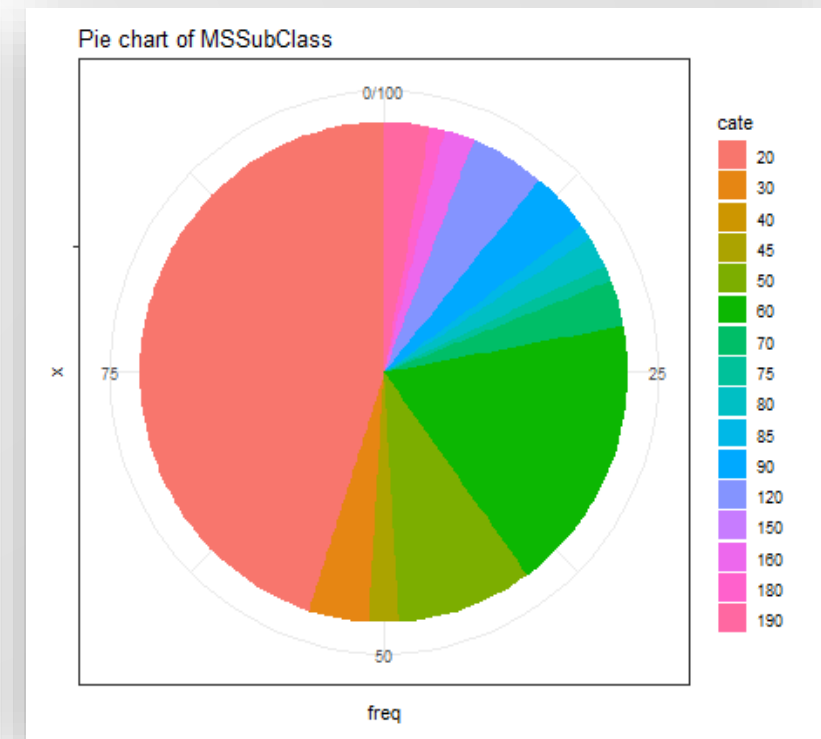
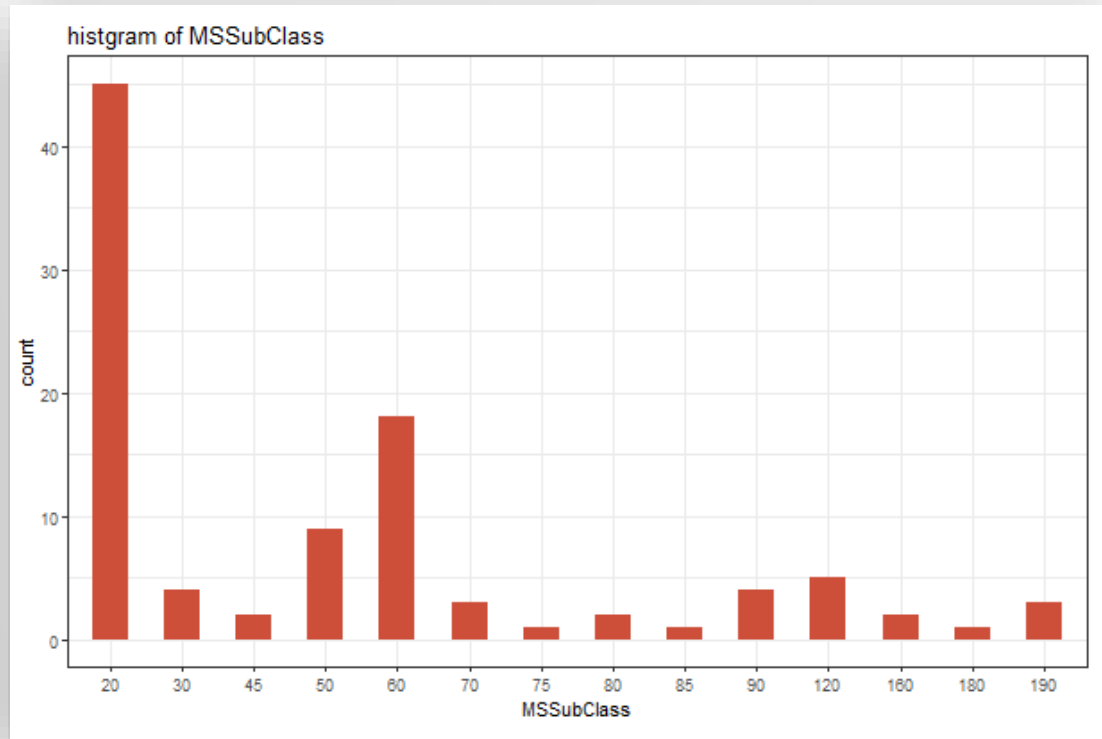
Visualization with Shiny

- Visualize Variable Distribution with Different Sample Number



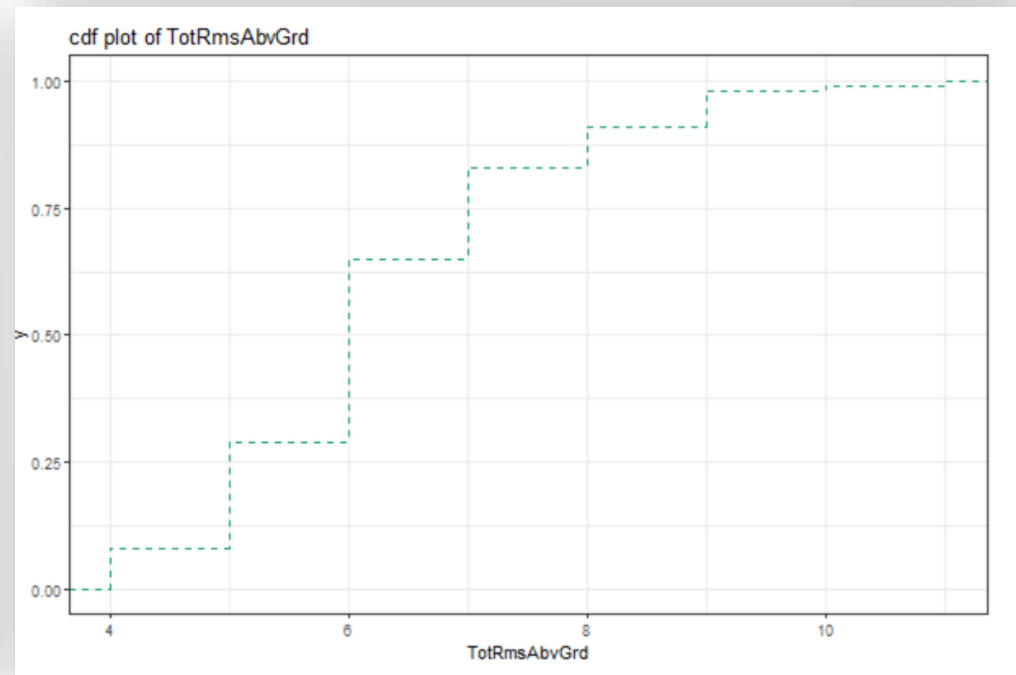
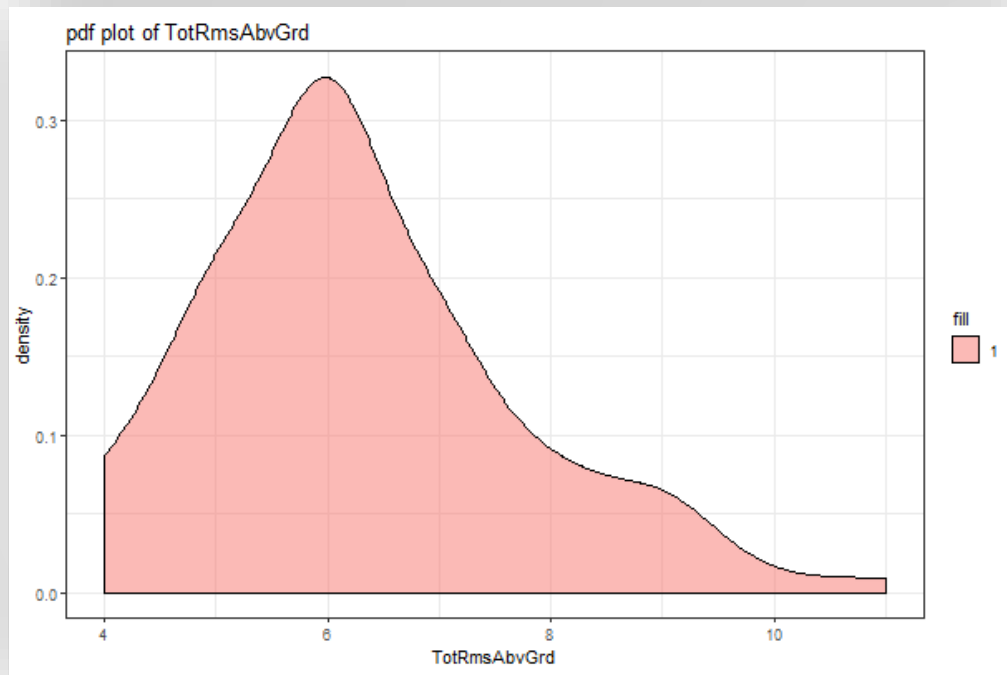
Visualization with Shiny

- 1 Dimensional Category Variable Distribution Graph:



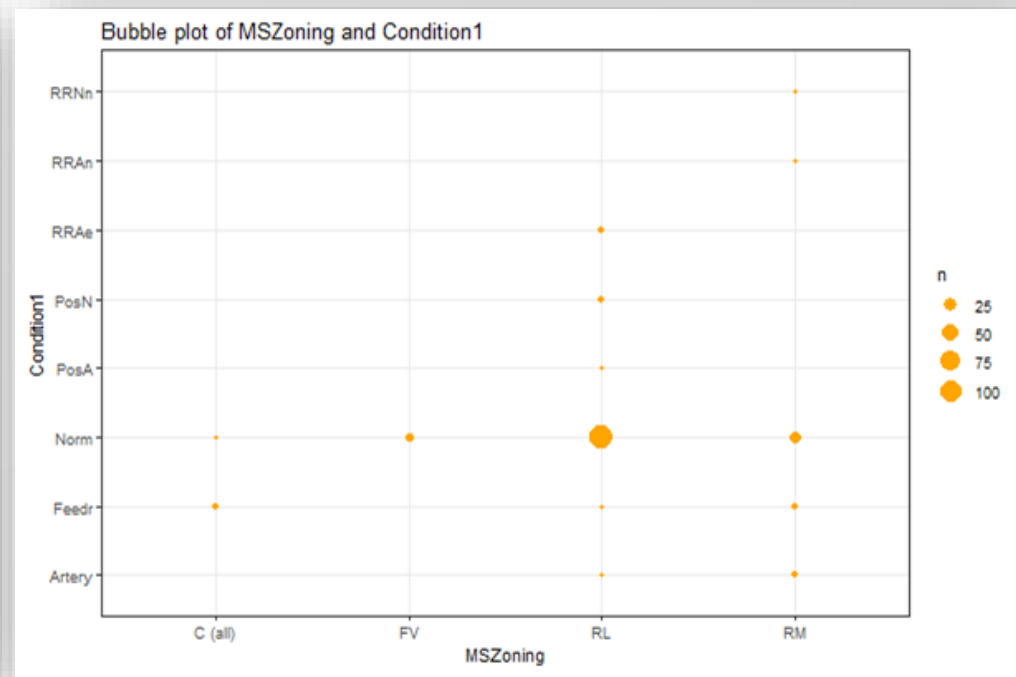
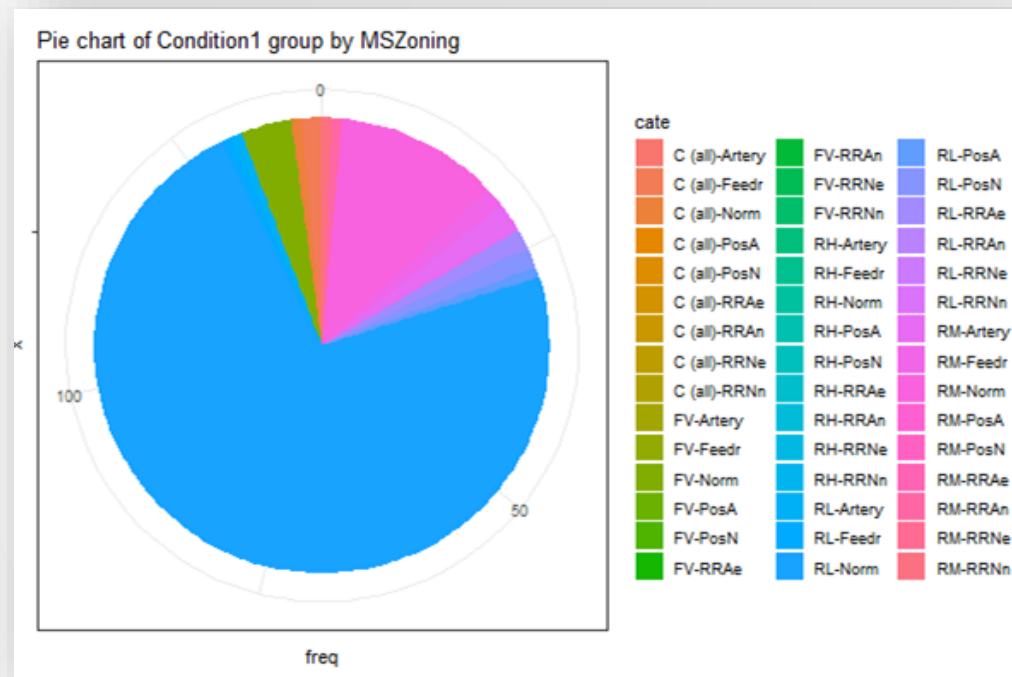
Visualization with Shiny

- 1 Dimensional Real Number Variable Distribution Graph:



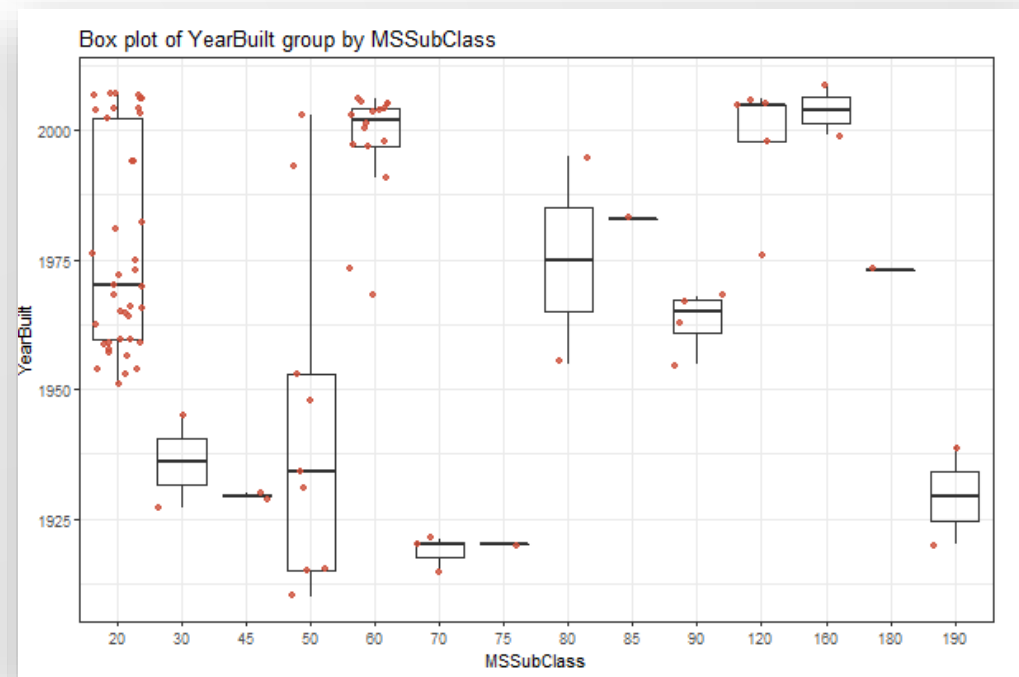
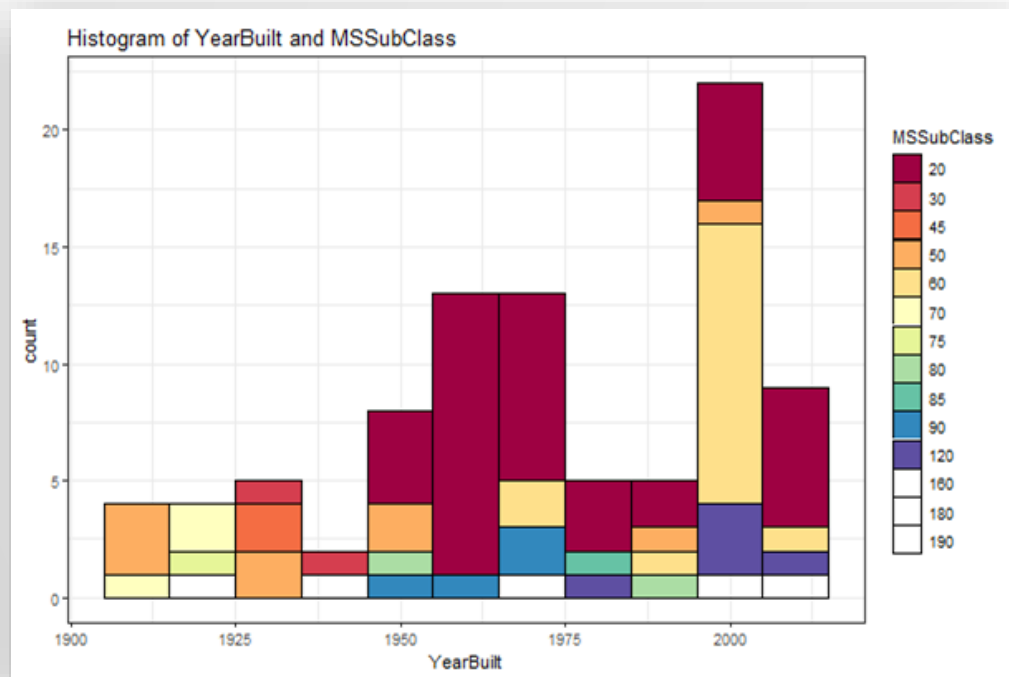
Visualization with Shiny

- 2 Dimensional Distribution Graph with 2 Category Variable:



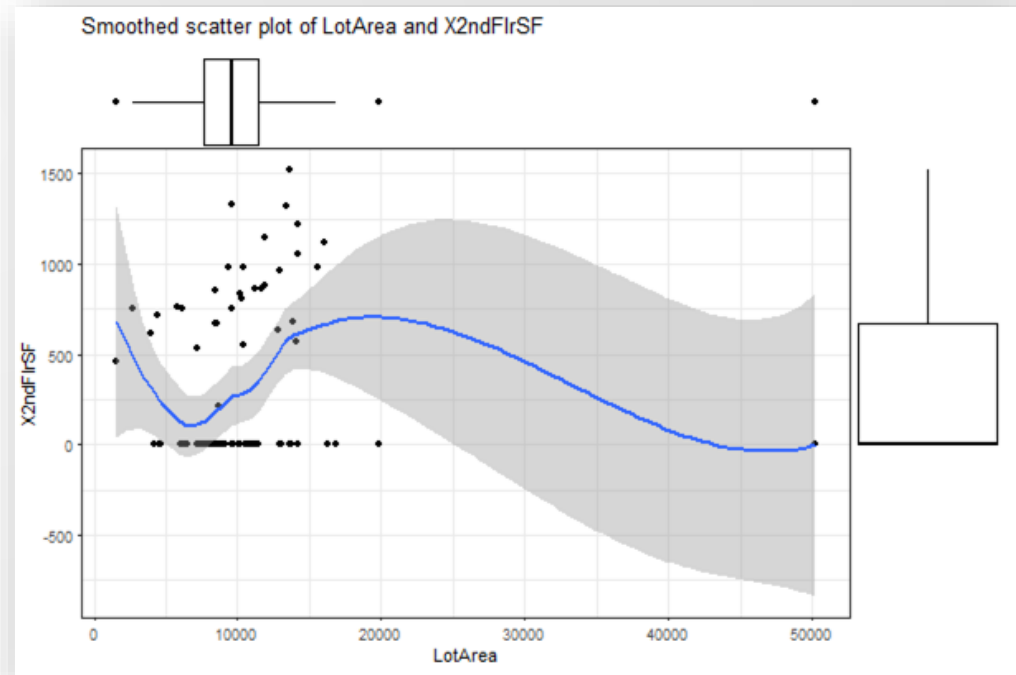
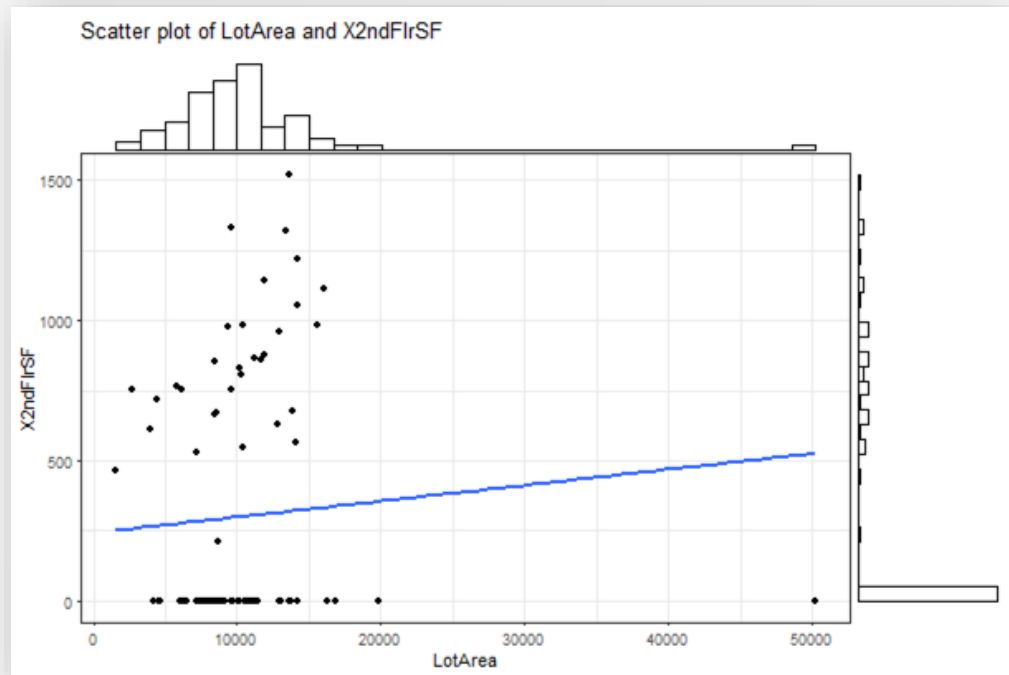
Visualization with Shiny

- 2 Dimensional Distribution Graph with 1 Cate Var 1 Number Var



Visualization with Shiny

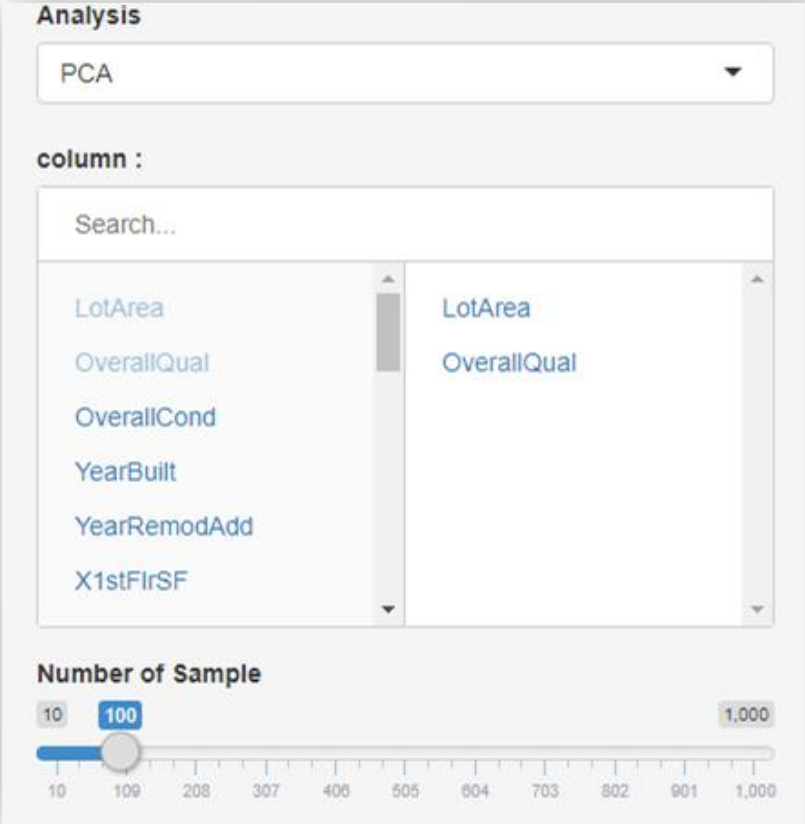
- 2 Dimensional Distribution Graph with 2 Real Number Variable



Visualization with Shiny

Analysis Panel Setting:

1. Analysis Type: PCA or CA
2. Target Column: specify the column to be analysed
3. Number of Sample: specify analysed sample number



The screenshot shows the 'Analysis' panel in a Shiny application. It features a dropdown menu for 'Analysis Type' set to 'PCA'. Below this is a 'column :' section with a search bar and two lists of available columns. The left list contains 'LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', and 'X1stFlrSF'. The right list contains 'LotArea' and 'OverallQual'. At the bottom, there is a 'Number of Sample' slider ranging from 10 to 1,000, with a current value of 100.

Analysis

PCA

column :

Search...

LotArea

OverallQual

OverallCond

YearBuilt

YearRemodAdd

X1stFlrSF

LotArea

OverallQual

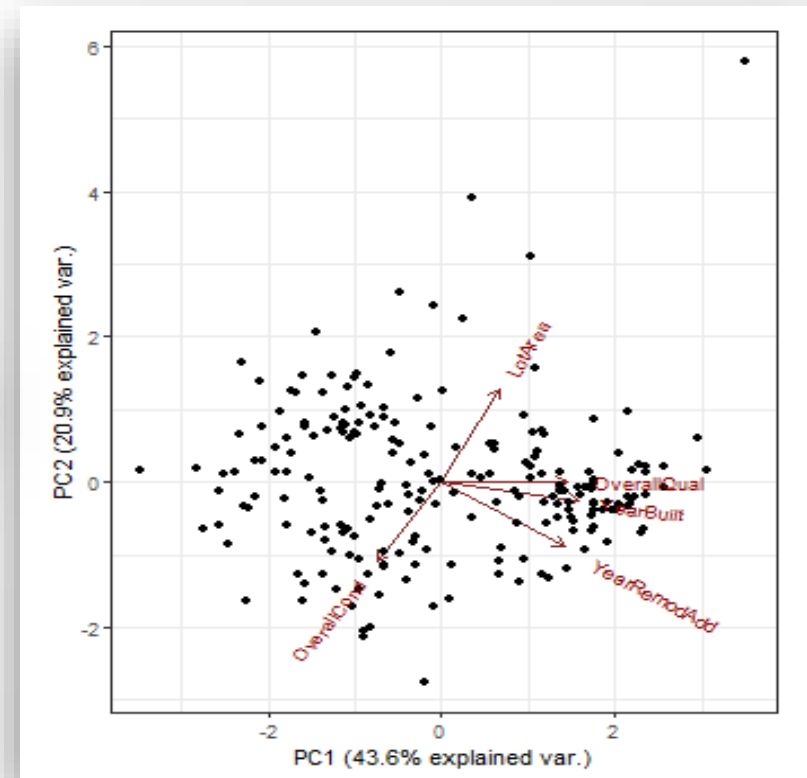
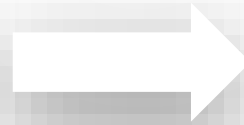
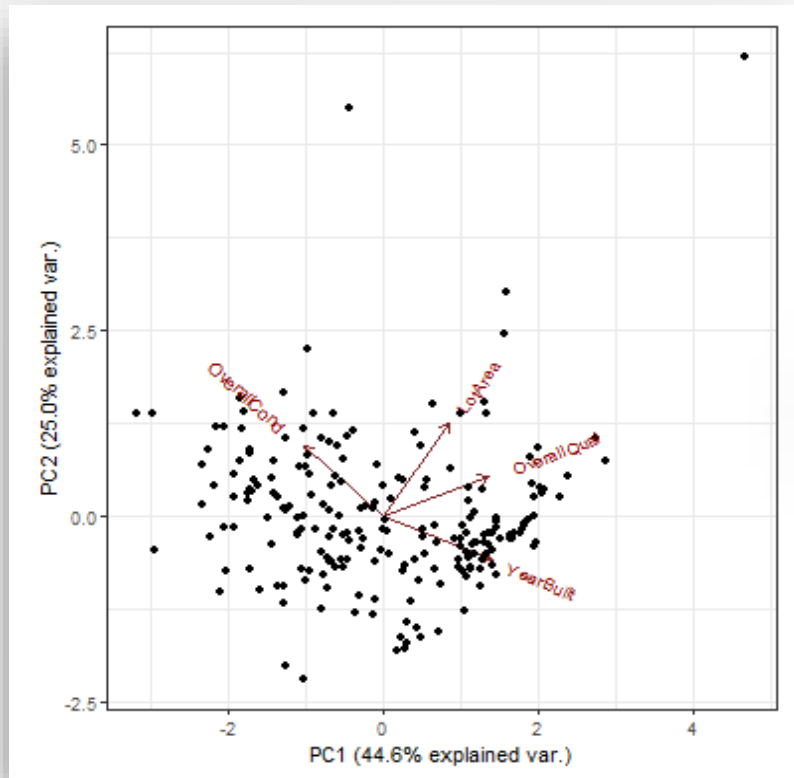
Number of Sample

10 100 1,000

10 109 208 307 406 505 604 703 802 901 1,000

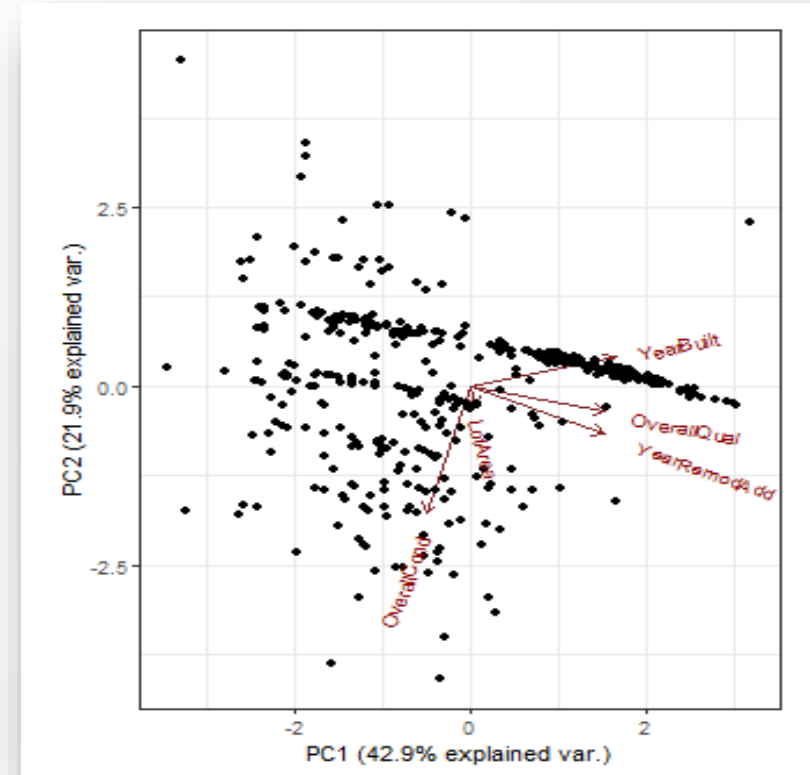
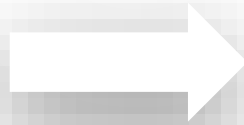
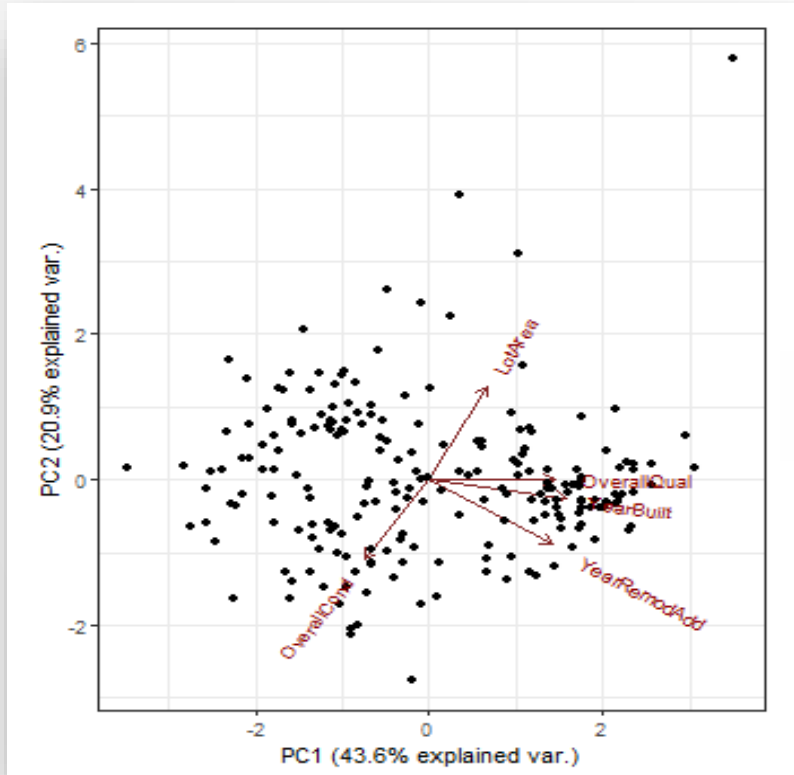
Visualization with Shiny

- Visualize Analysis Result by Different Target Column Set



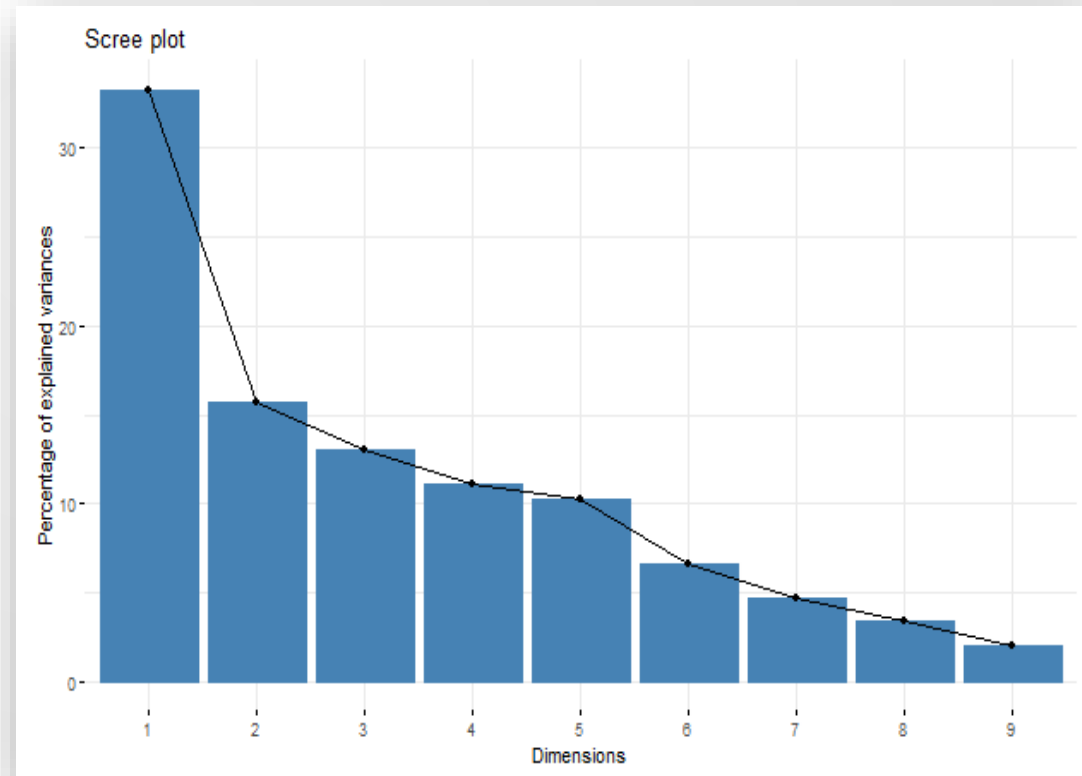
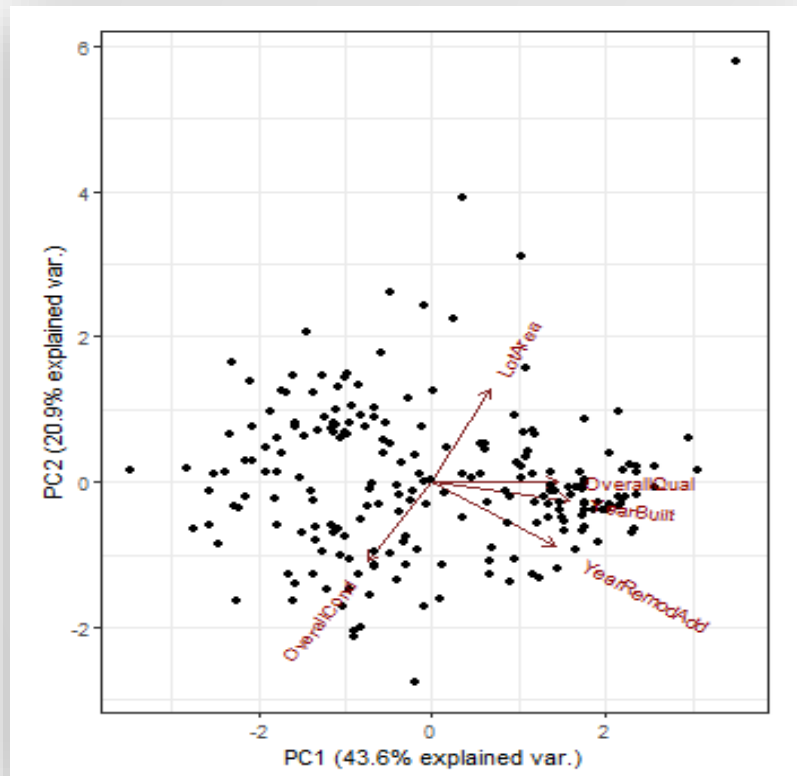
Visualization with Shiny

- Visualize Analysis Result by Different Sample Point



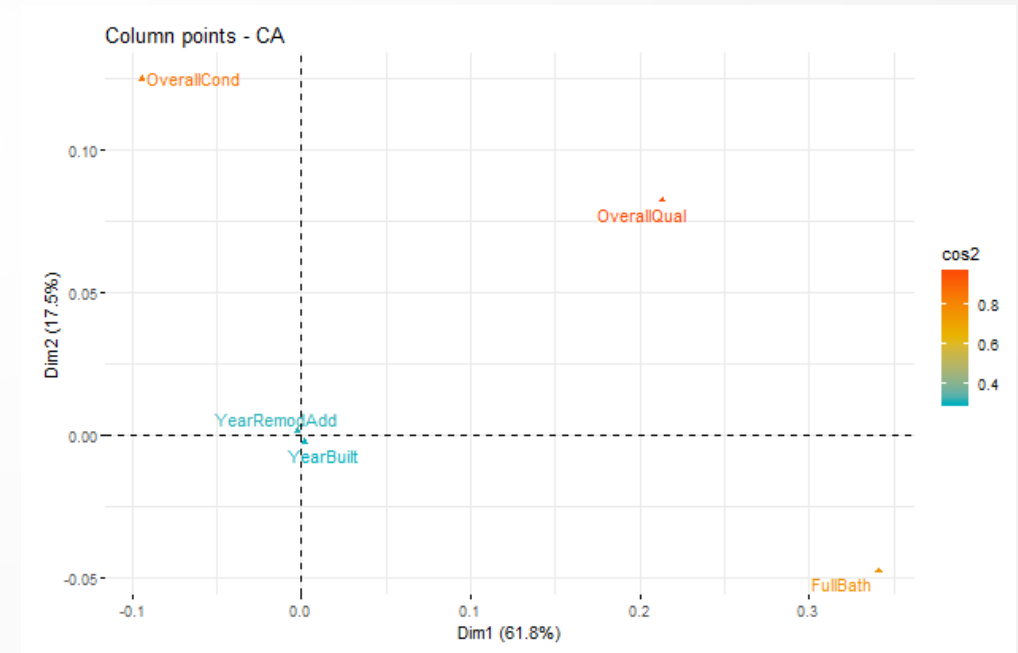
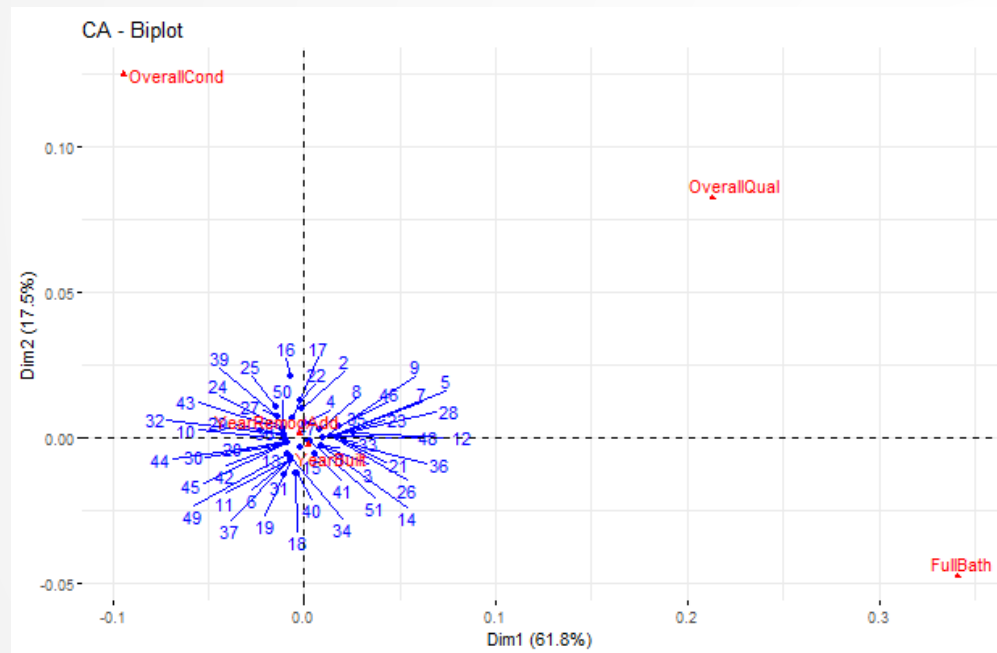
Visualization with Shiny

- Principal Component Analysis(PCA)



Visualization with Shiny

- Correspondence Analysis(CA)





Modeling



Feature Engineering

- Add new variables:
 - `Old <- YrSold - YearBuilt`
 - `OverallGrd <- OverallQual * OverallCond`
 - `GarageScore <- GarageArea * GarageQual`
- Delete variables

Model Evaluation & Comparison

- 此為Kaggle競賽資料，所以原本就有測試資料集被切出，因此我們的訓練資料(即前1460筆資料)僅切割成訓練集和驗證集，採用的方式為10-fold，另外，為了公平比較模型，四個模型的訓練集與驗證集皆相同。
- 房價為連續型資料，所以我們針對四個模型分別計算訓練集與驗證集的RMSE與RMSLE，並且將Kaggle所給的測試資料預測後上傳，看測試資料的RMSLE，之後針對三個資料集的四個模型預測結果進行比較。



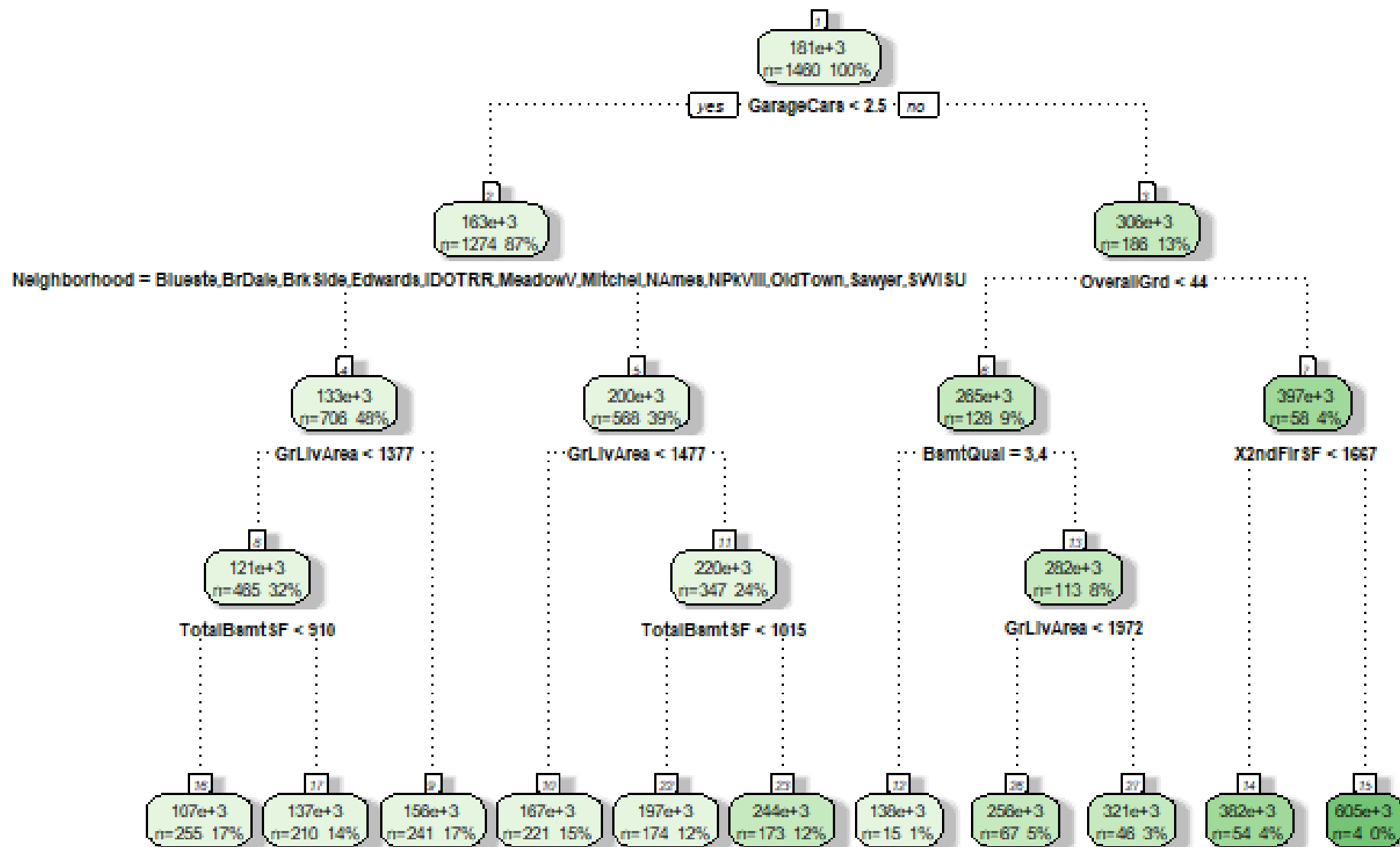
Null Model: Decision Tree



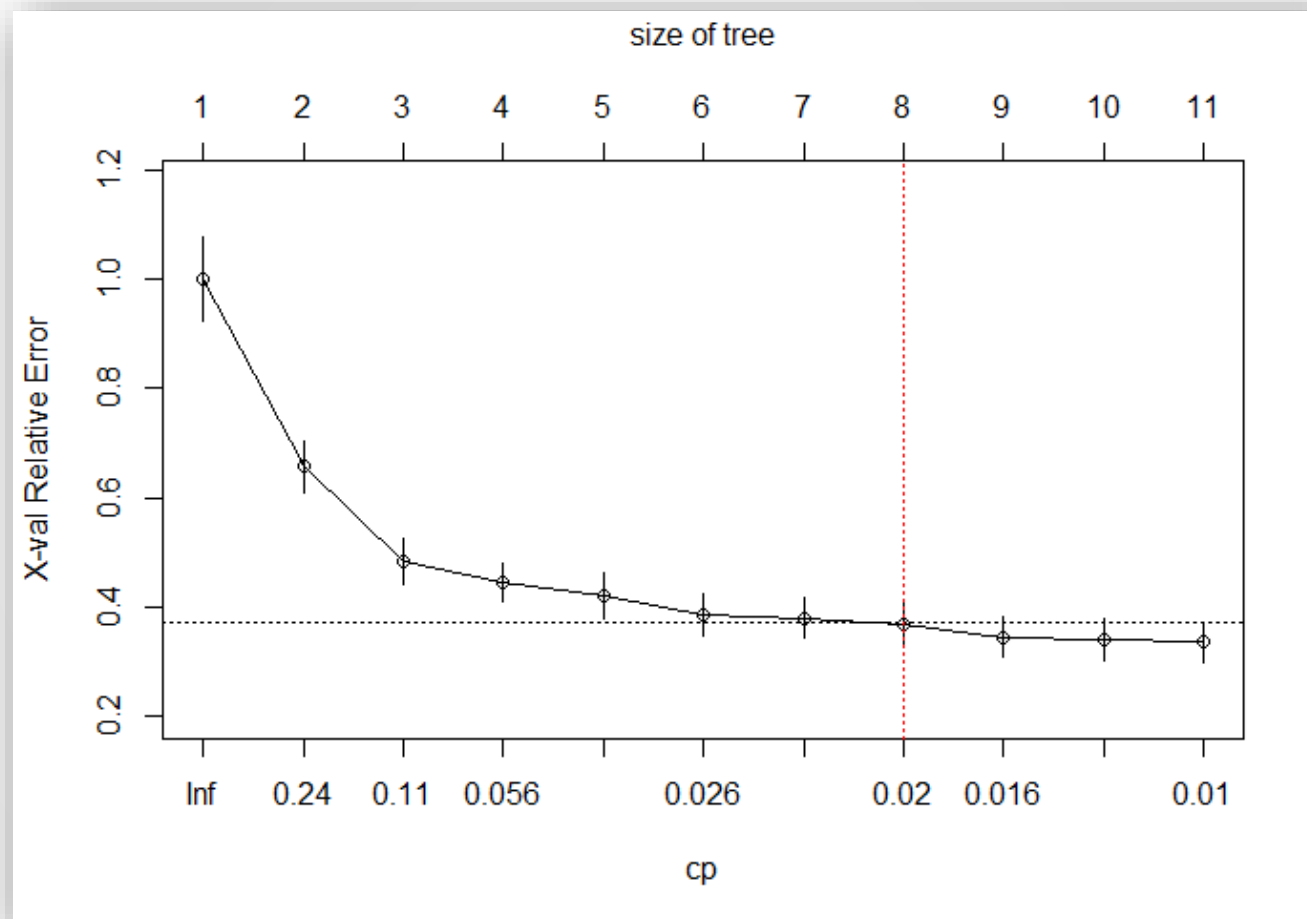
Modeling – Decision tree

- Regression tree:
 - Starts by searching for every distinct values of all its predictors, and splits the value of a predictor that minimizes some statistic. Ex: SSE
 - Suppose we have p variables, each has n observations, we go through every p and n to split the space until the specific p and n minimizes the loss function.
 - $\min_{p,n} [\min_{c_1} Loss(y_i, c_1) + \min_{c_2} Loss(y_i, c_2)]$, where $c_m = mean(y_i | x_i \in R_m)$, R_m is the m unit of space.
 - To avoid overfitting, use cost-complexity $C_\alpha(Tree) = SSE(TREE) + \alpha * Size(Tree)$ to prune the tree.

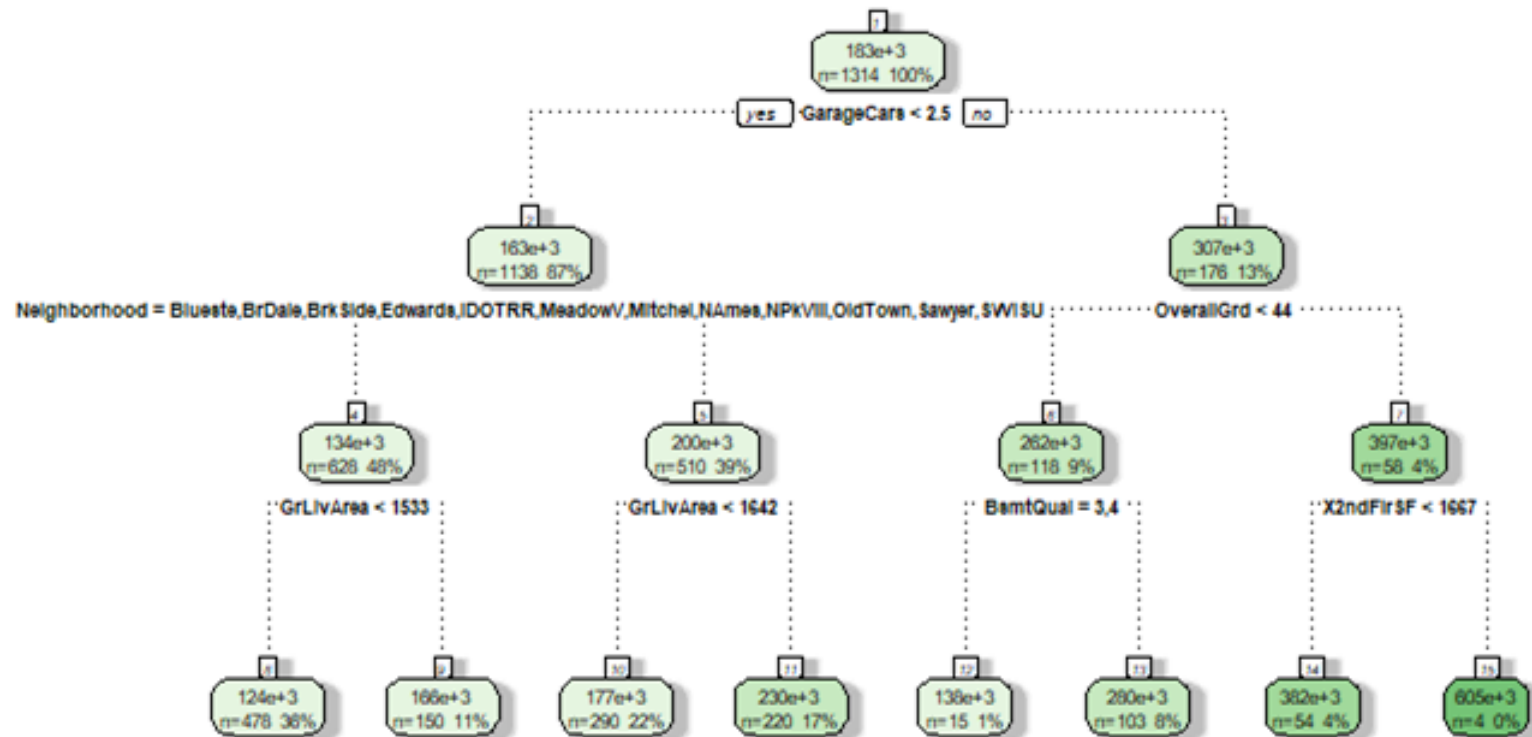
Tree



CP



Prune tree



Result

	RMSE	RMSLE
Train	42103.03	0.32294
Validation	48190.42	0.50644
Test		0.26675

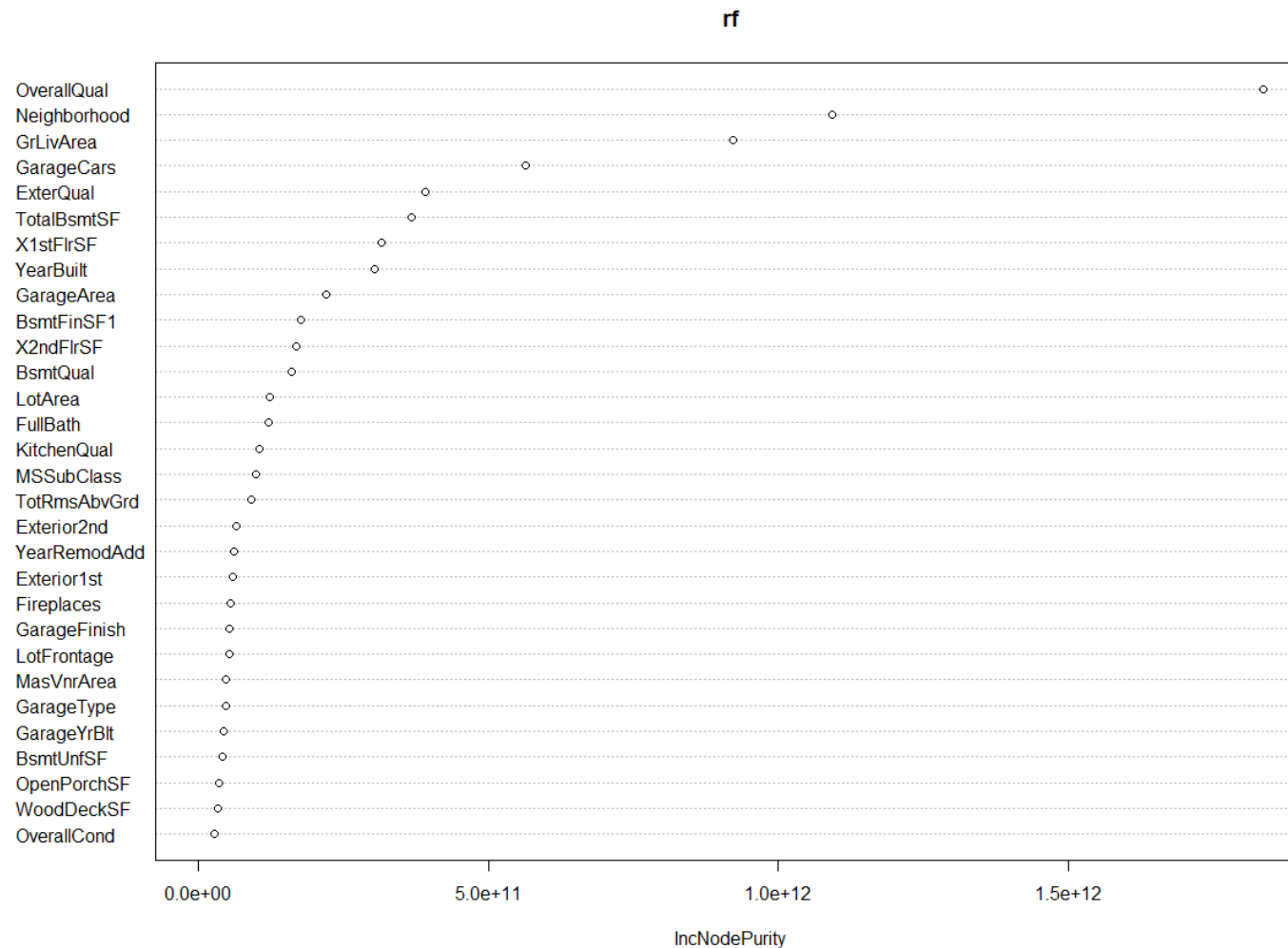


Random Forest

Modeling – Random Forest

- An ensemble learning method, which is based on decision tree.
- Random Forest is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.
- Use resampling method through row and column to grow many decision trees.
- $g(x) = f_0(x) + f_1(x) + f_2(x) + \dots$
- The random forest model is very good at handling tabular data with numerical variables, or categorical variables with fewer than hundreds of categories.

Importance



The total decrease in node impurities from splitting on the variable, averaged over all trees. For regression, it is measured by residual sum of squares.

Result

	RMSE	RMSLE
Train	27830.70	0.14063
Validation	27577.55	0.13999
Test		0.14814



XGBoost

Modeling – XGBoost

- Boosted tree 為基於cart樹的一種ensemble方法。
- 基本概念是將每次cart樹的預測分數 f 累加得到我們要的預測結果，每一輪都選取能使目標函數下降最多的cart樹的結果加入前一輪的結果。
- Cart樹對資料點的預測分數，取決於資料點被劃到哪個葉子上，而該葉子會加減多少分稱之為權重 w 。
- 資料點如何分到特定葉子則稱為 q (樹的結構)
- 定義複雜度函數 Ω 為 W 與 T (葉子個數)的函數

Modeling – XGBoost

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant \end{aligned}$$

想找到最佳的 f_t

- 目標函數可視為真實值 y 與預測值 \hat{y} 的loss function與複雜度的函數
- 在給定樹的結構下，我們能計算出能使object function下降最多的樹的權重
- 為了找到最佳的 f_t ，每次迭代我們會一次列舉很多不同結構的樹，並利用上述obj找到對應該結構的最佳權重，接著再比較不同樹對obj下降的表現，找到最優表現的樹加入結果。

Modeling – XGBoost

- 每次迭代列舉的樹都是把之前已分割的節點繼續切下去，由於有複雜度懲罰項的關係，一直切下去不一定會比較好，若是下降幅度太小會減掉這次樹的分枝。
- XGBoost是在目標函數中的模型複雜度中，使用L2的正規化項(權重 w 的平方和乘上給定的 λ)，並導入學習速率 η ，讓我們能自己調整每次迭代影響的大小。

Choosing parameters

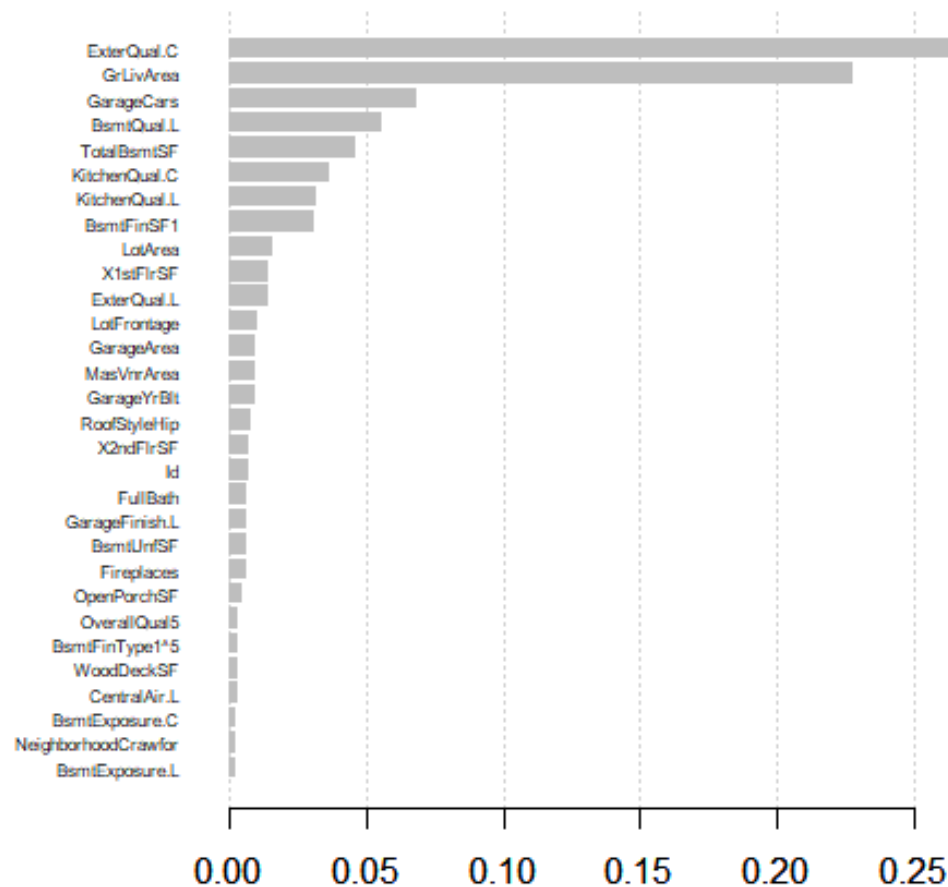
對於模型參數

- `nrounds` = 總迭代次數
- `max_depth` = 單一CART樹最大深度
- `eta` = 學習率
- `subsample` = 生成每棵樹使用的樣本比率

我們使用`caret`套件幫助我們對參數進行grid search，經過嘗試後，使用

- `nrounds=100`
- `eta = 0.075,`
- `max_depth = 8,`
- `subsample = 0.5`
- 而L2正規項係數的`lambda`經過嘗試後決定使用3

Feature importance



變數重要性如下，其中x軸為gain，為目標函數下降的貢獻比例(由每次迭代的樹加總)，這裡僅列出前30重要的特徵。

Result

	RMSE	RMSLE
Train	9709.454	0.05482
Validation	26980.14	0.13752
Test		0.13882

The background is a solid orange color. In the top-left corner, there are three vertical bars of varying heights, each composed of three overlapping circles. In the bottom-right corner, there are four vertical bars of varying heights, each composed of three overlapping circles.

Support Vector Regression

Introduction

- The basic idea of SVM :

Find a hyperplane to separate different categories and maximize margin

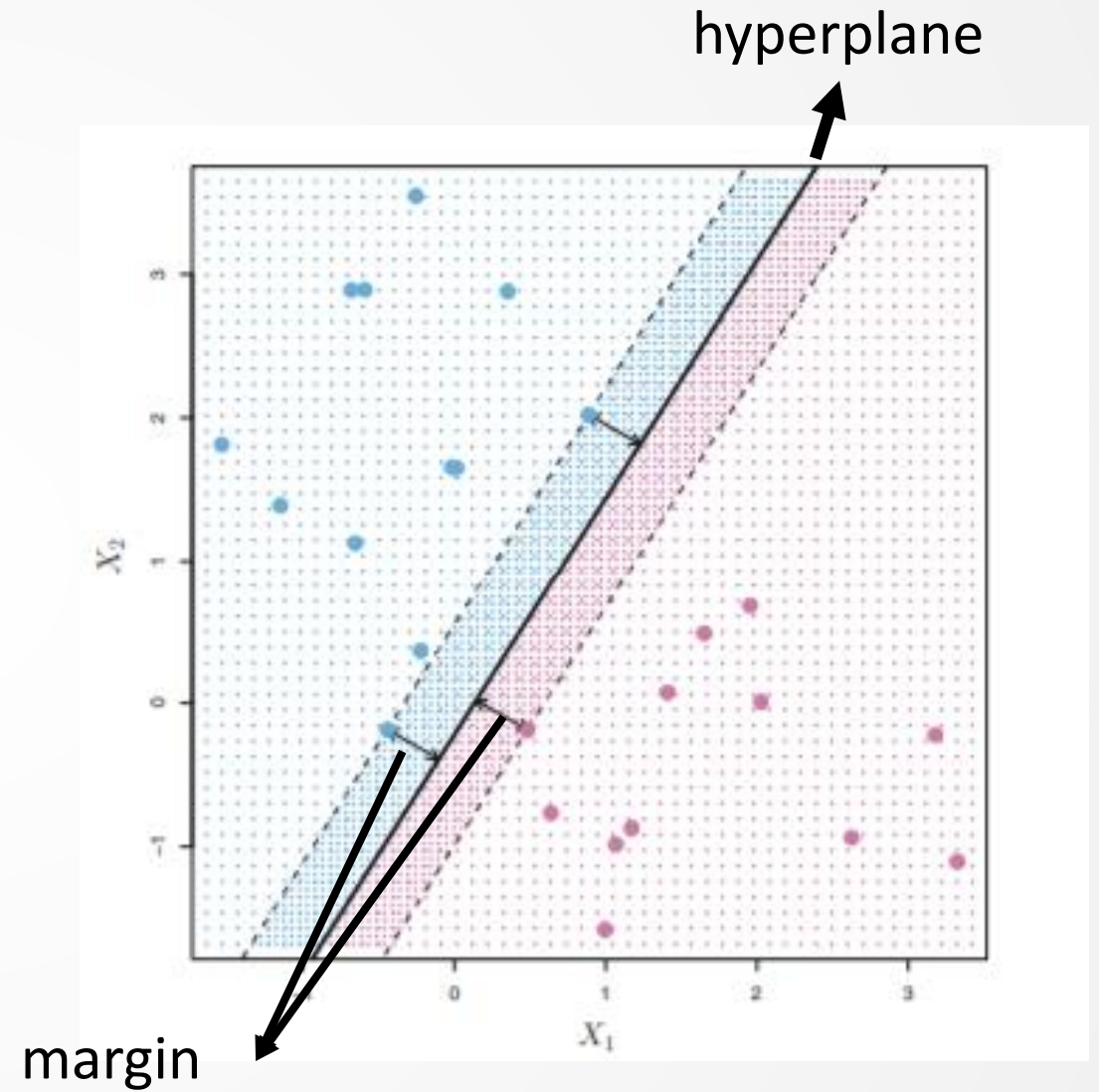


Figure 9.3 An Introduction to Statistical Learning by Gareth James Daniela Witten Trevor Hastie Robert Tibshirani

Introduction

- The basic idea of support vector regression:

Find a hyperplane to predict y and minimize the ε -insensitive loss

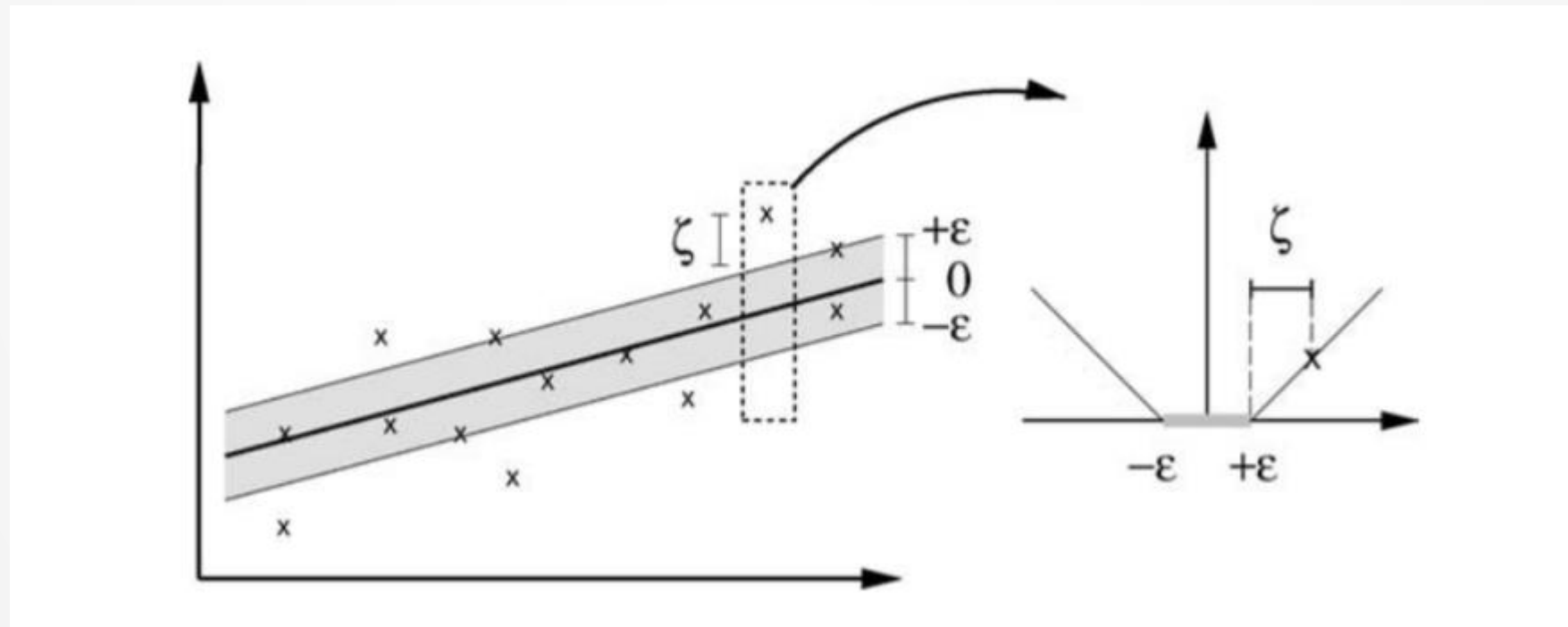
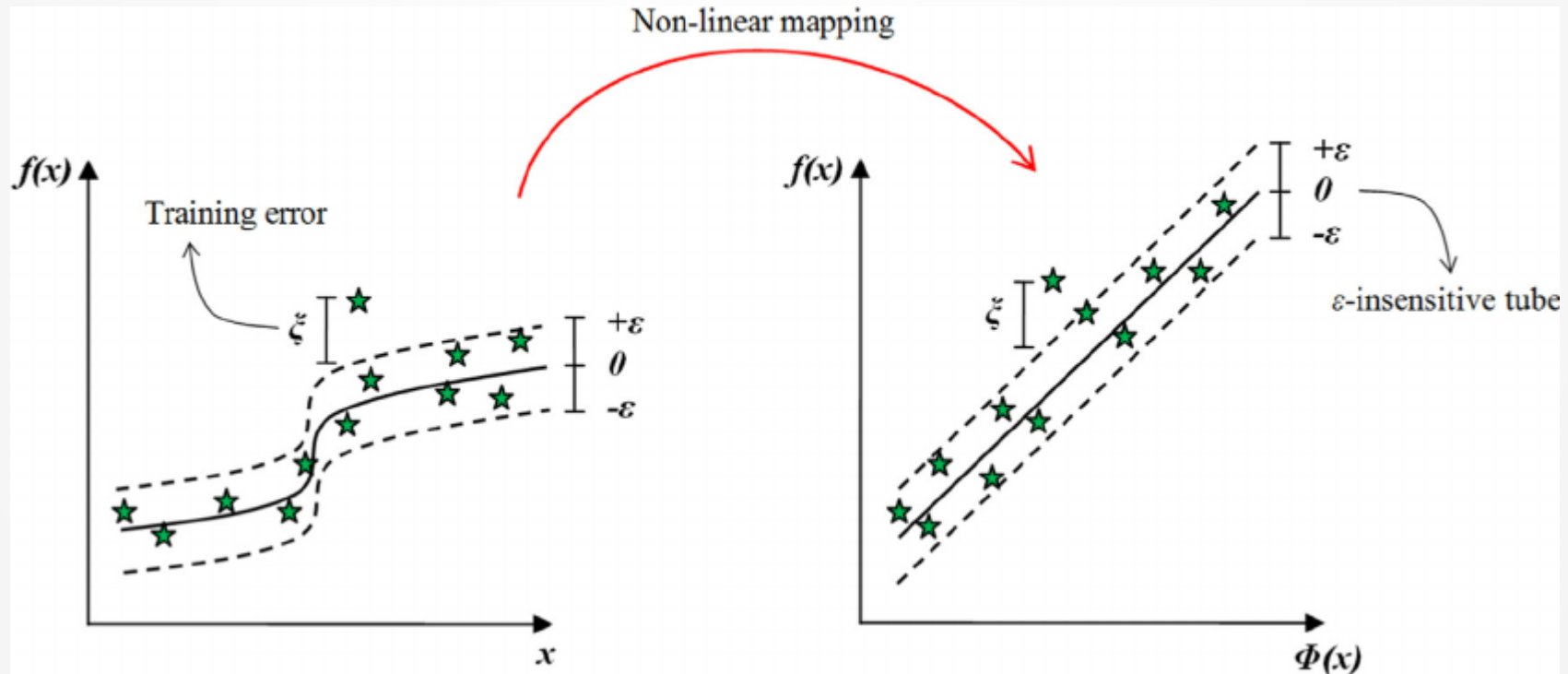


Figure 1 A tutorial on support vector regression, ALEX J. SMOLA and BERNHARD SCHOLKOPF, Statistics and Computing 14: 199–222, 2004

Introduction

- The purpose of kernel:



Introduction

- The equations of SVR need to solve:

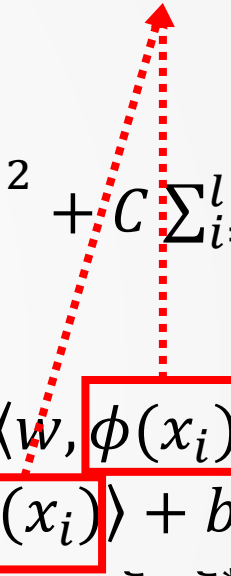
$$\text{minimize} \quad \frac{1}{2} ||w||^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Linear

差異在是否對自變數做非線性轉換

$$\text{minimize} \quad \frac{1}{2} ||w||^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, \phi(x_i) \rangle - b \leq \epsilon + \xi_i \\ \langle w, \phi(x_i) \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$


Nonlinear

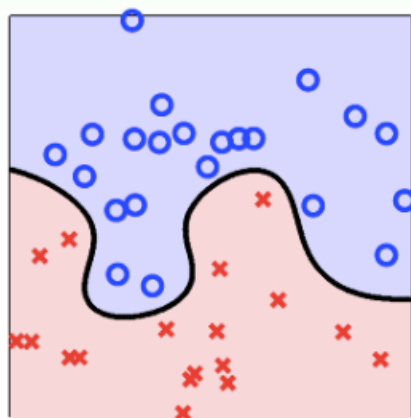
R Package Tune Parameter

- 使用的套件：e1071
- 可調參數對照：

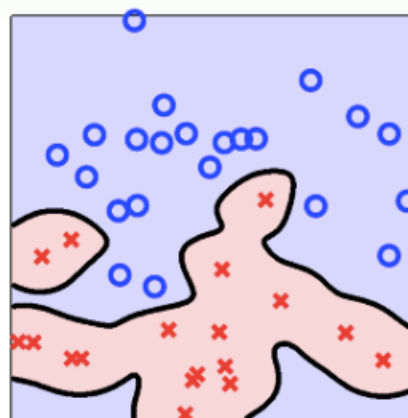
C: cost – 與容錯程度相關，越大則容錯程度越小

kernel: $\phi(x)$ – 非線性轉換的函數

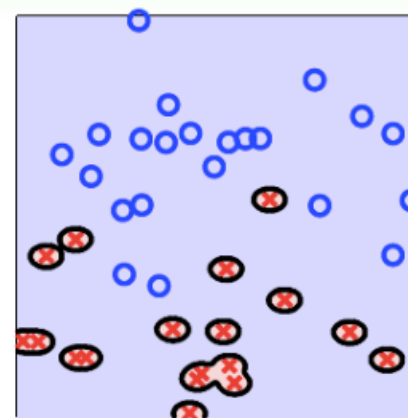
gamma: 使用 Gaussian Kernel 才會用到的參數，越大越會導致過度配適



$$\exp(-1\|\mathbf{x} - \mathbf{x}'\|^2)$$



$$\exp(-10\|\mathbf{x} - \mathbf{x}'\|^2)$$

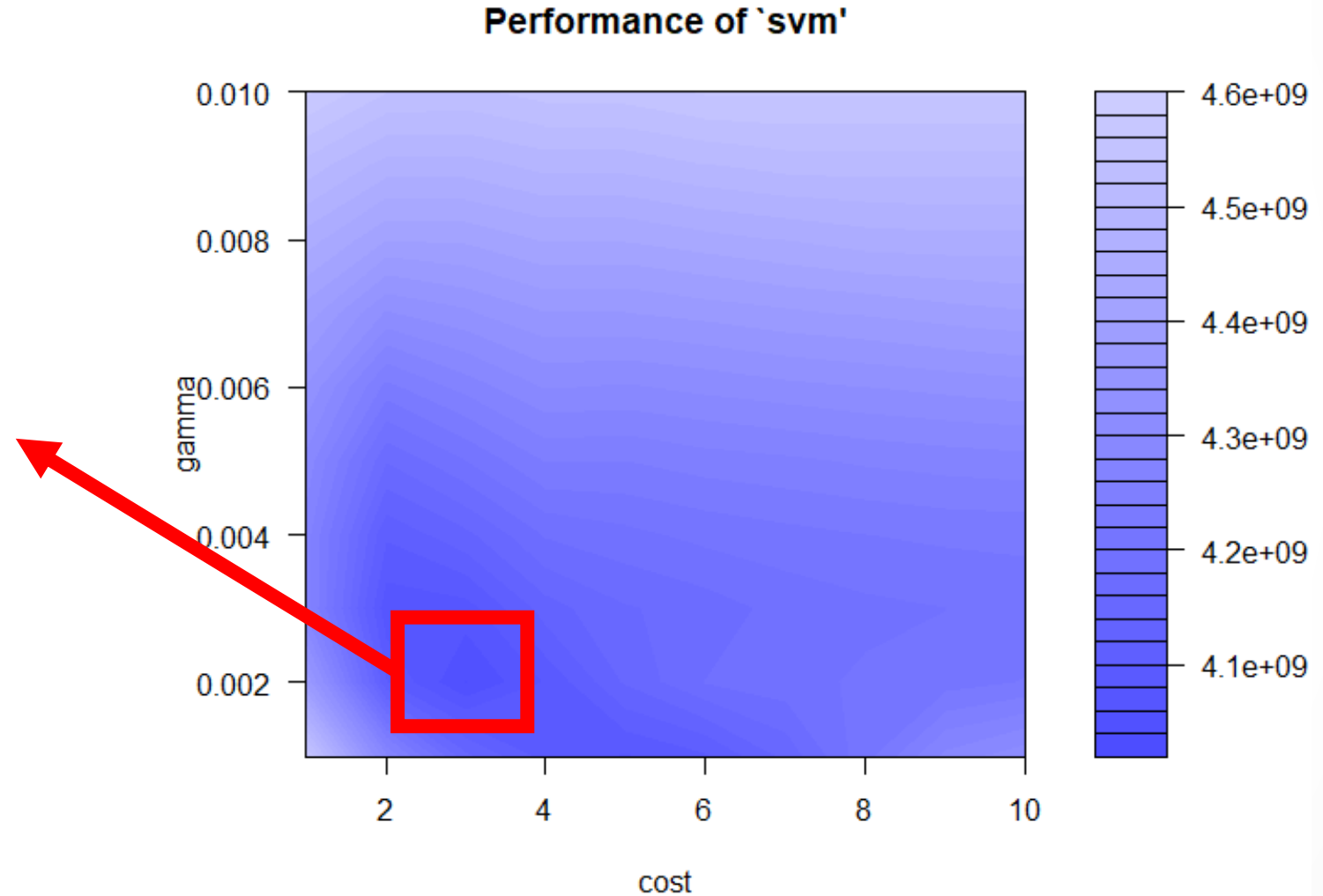


$$\exp(-100\|\mathbf{x} - \mathbf{x}'\|^2)$$

gamma分別為1, 10, 100下的SVM配適情況，擷取自林軒田教授機器學習技法

Tune Result

cost : 3.4
gamma : 0.002



Model Evaluation

	RMSE	RMSLE
Train	22990.07	0.10166
Validation	27321.55	0.12306
Test		0.11798

Comparison Result

RMSE of Four Model

	Decision Tree	Random Forest	XGBoost	SVR
Train	42103.03	27830.70	9709.454	22990.07
Validation	48190.42	27577.55	26980.14	27321.55

Comparison Result

RMSLE of Four Model

	Decision Tree	Random Forest	XGBoost	SVR
Train	0.32294	0.14063	0.05482	0.10166
Validation	0.50644	0.13999	0.13752	0.12306
Test	0.26675	0.14814	0.13882	0.11798

Reference

Packages:

- ggplot2
- ggpubr
- rpart
- rpart.plot
- rattle
- randomForest
- e1071
- xgboost

Reference

- <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>
- <https://www.kaggle.com/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda>
- <https://zhuanlan.zhihu.com/p/51586879>
- <https://zhuanlan.zhihu.com/p/49049535>
- <https://cloud.tencent.com/developer/article/1005033>
- <https://github.com/topepo/caret/issues/336>
- <https://zhuanlan.zhihu.com/p/24577989> <https://stackoverflow.com/questions/39371738/r-xgboost-importance-plot-with-many-features>
- <http://steve-chen.tw/?p=369>
- <http://www.dehong.space/XGBoost>
- <https://www.rdocumentation.org/packages/xgboost/versions/0.6.4.1/topics/xgb.importance>
- A tutorial on support vector regression, ALEX J. SMOLA and BERNHARD SCHOLKOPF, Statistics and Computing 14: 199–222, 2004
- A support vector regression model for predicting tunnel boring machine penetration rates, Satar Mahdevari, Kourosh Shahriar, Saffet Yagiz and Mohsen Akbarpour Shirazi, International Journal of Rock Mechanics & Mining Sciences 72 (2014) 214–229
- An Introduction to Statistical Learning, Gareth James Daniela Witten Trevor Hastie Robert Tibshirani
- Support Vector Machine Regression, <http://kernelsvm.tripod.com/>, 2019/06/24
- 林軒田教授機器學習技法，<https://www.csie.ntu.edu.tw/~htlin/mooc/>