Predicting Housing Price

Intro:

In this project we use R language to build a model that predicts the price of a house based on its features (ex. No.of.bedrooms, square footage, location…)

Dataset:

Contains 271 columns, 1460 entries

We do preprocessed data:

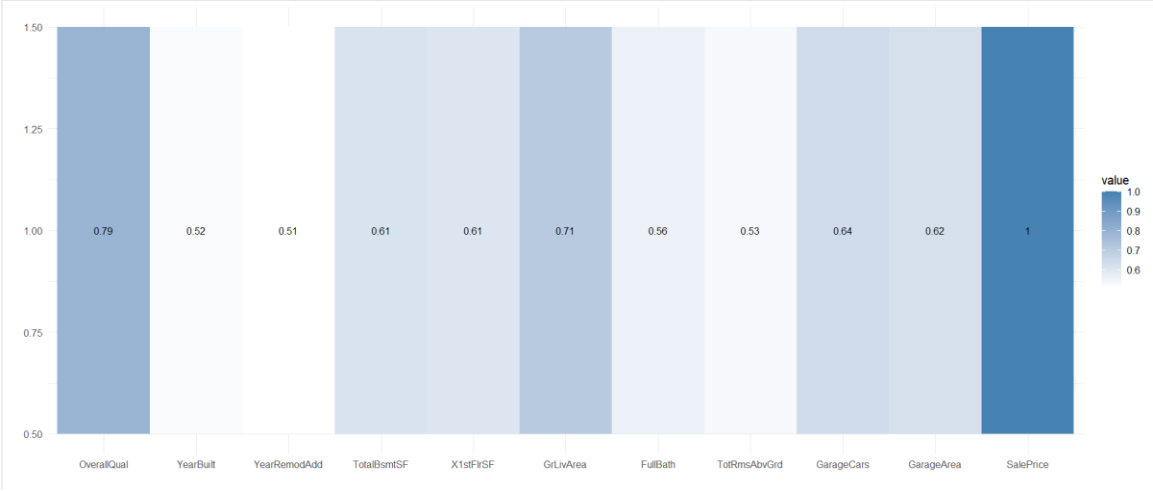
Clean it by remove any unnecessary columns or rows, drop all column that have non value or zero value, change string data into numeric values.

Data preprocessing:

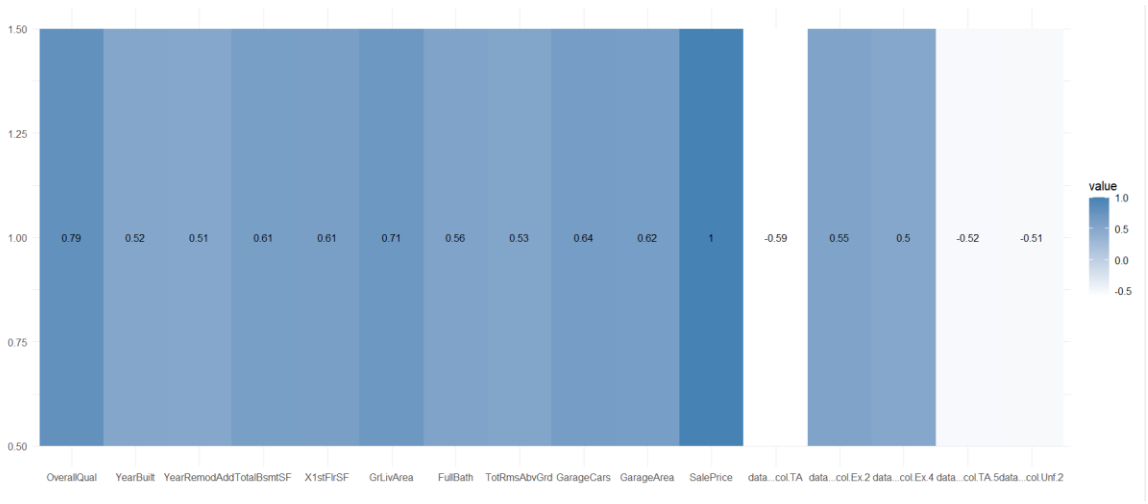
- **1st version** :
    - Filled all numeric nan values with the mean value.
    - Filled all categorical values with the mode value.
    - Dropped the following columns as the nan values in them was greater than 80% :

1. Fence
2. Alley
3. FireplaceQu
4. PoolQC
5. MiscFeature

- Dropped Id column.
- Applied one hot encoding on all categorical columns except columns which have relation between its categories ascending or descending.



- **2nd version :**
    - We get only the numeric columns.
    - Filled all nan values with mean value.
    - Dropped Id column.



(2) Models :

a) Linear regression Models :

1) 1st model :

- Used the 2nd version of dataset.
- Used all the dataset.
- 80% train , 20% test , used cross validation.
- RMSE : 37049.58
- R squared : 0.7906587
- MAE : 22083.89

```
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1313, 1315, 1314, 1313, 1314, 1314, ...
Resampling results:

  RMSE      Rsquared   MAE
  37049.58  0.7906587  22083.89
```

2) 2nd model :

- Used the 2nd version of dataset.
- Used only columns with correlation above 0.5 (-0.5).
- 80% train , 20% test , used cross validation.
- RMSE : 38064.68
- R squared : 0.7792779
- MAE : 24163.46

```
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1313, 1315, 1314, 1313, 1314, 1314, ...
Resampling results:

  RMSE      Rsquared   MAE
  38519.39  0.7745879  24125.15
```

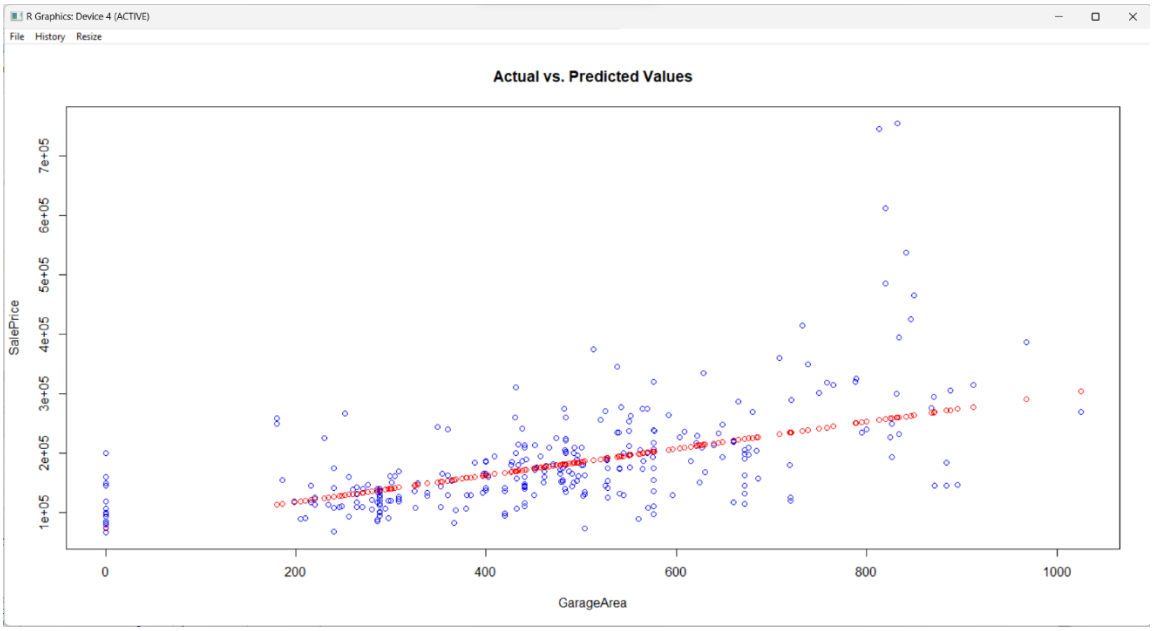
3) 3rd model :

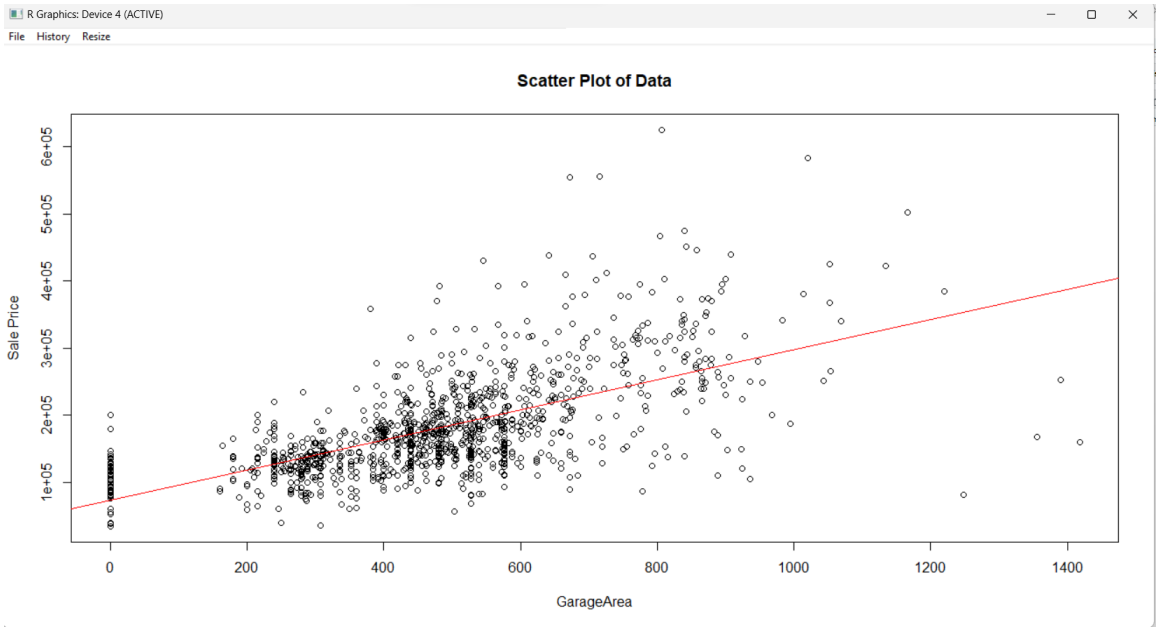
- Used the 1st version of dataset.
- Used all the dataset.
- 80% train , 20% , cross validation.
- RMSE : 35956.16
- R squared : 0.8013526
- MAE : 21954.14

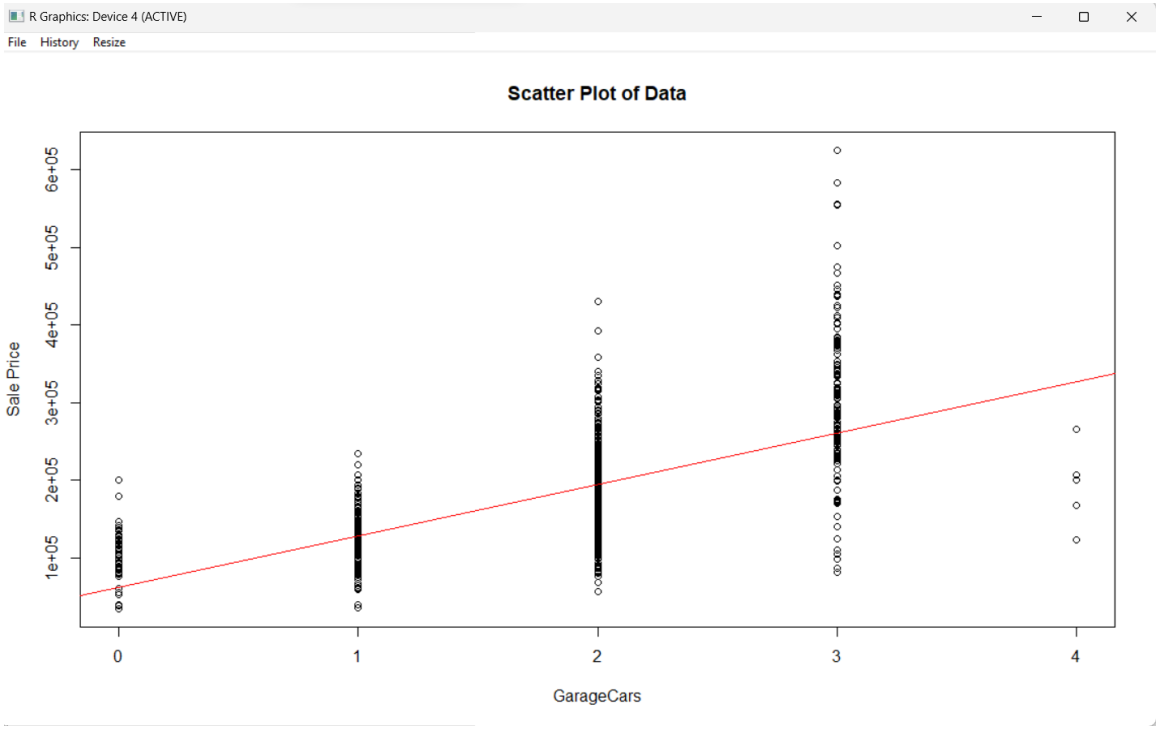
```
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1313, 1315, 1314, 1313, 1314, 1314, ...
Resampling results:

  RMSE      Rsquared   MAE
  35956.16  0.8013526  21954.14
```

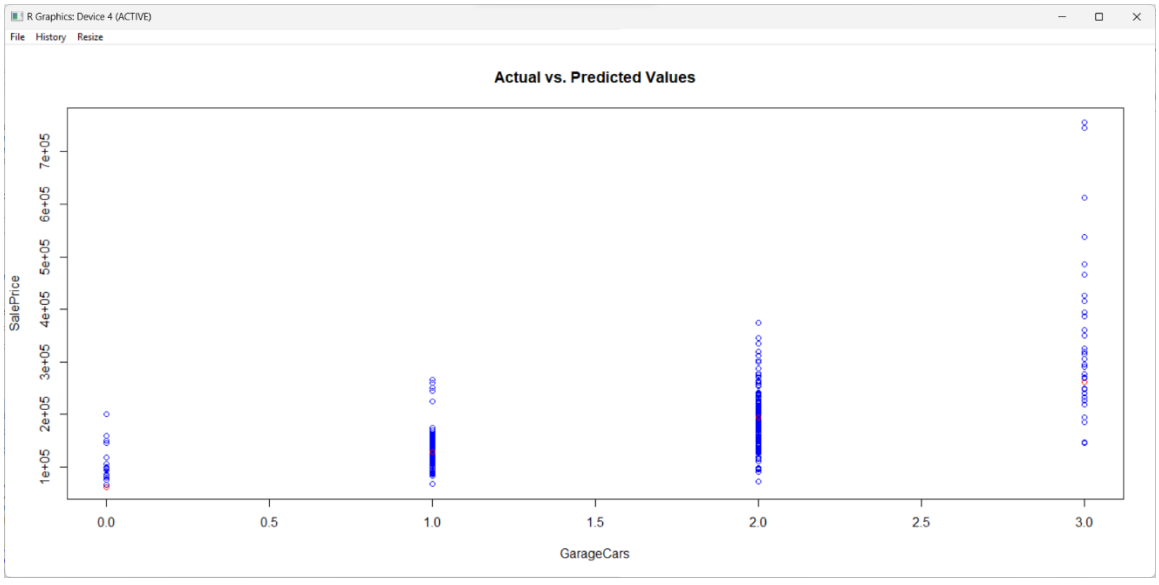
4) **4th model**

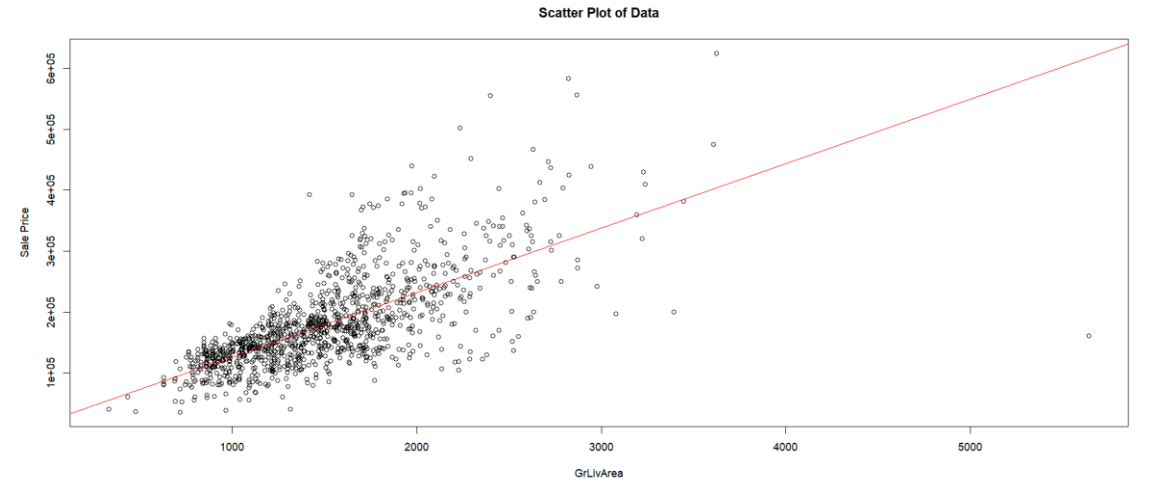
we use one feature for every fit.

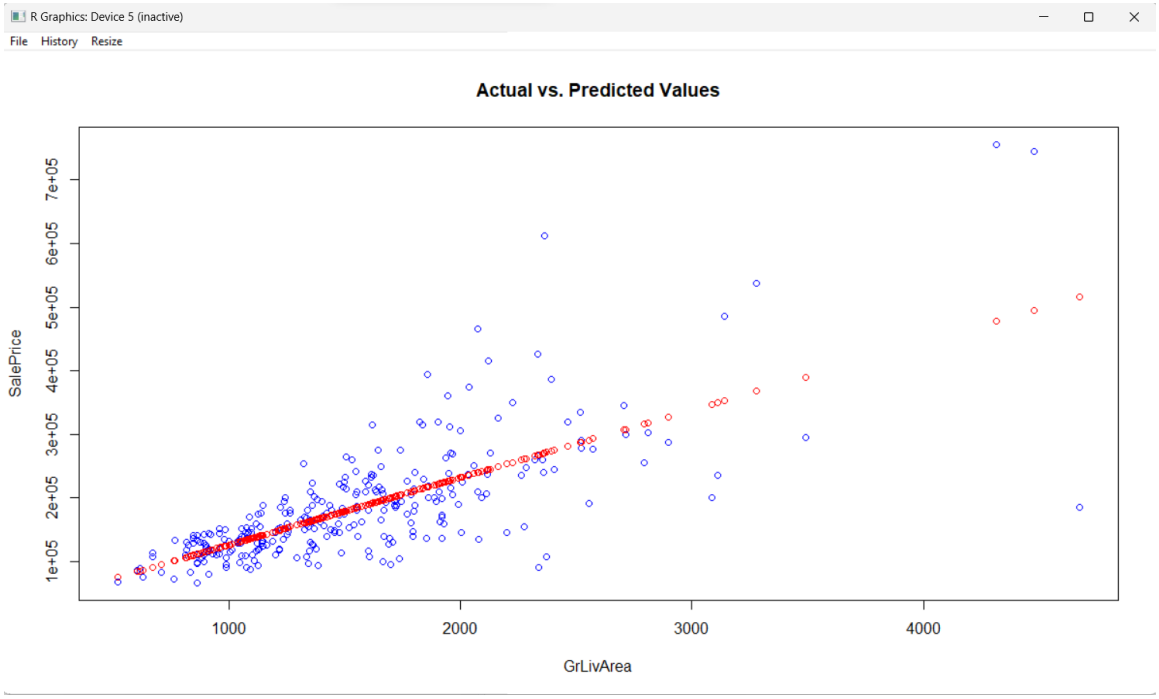


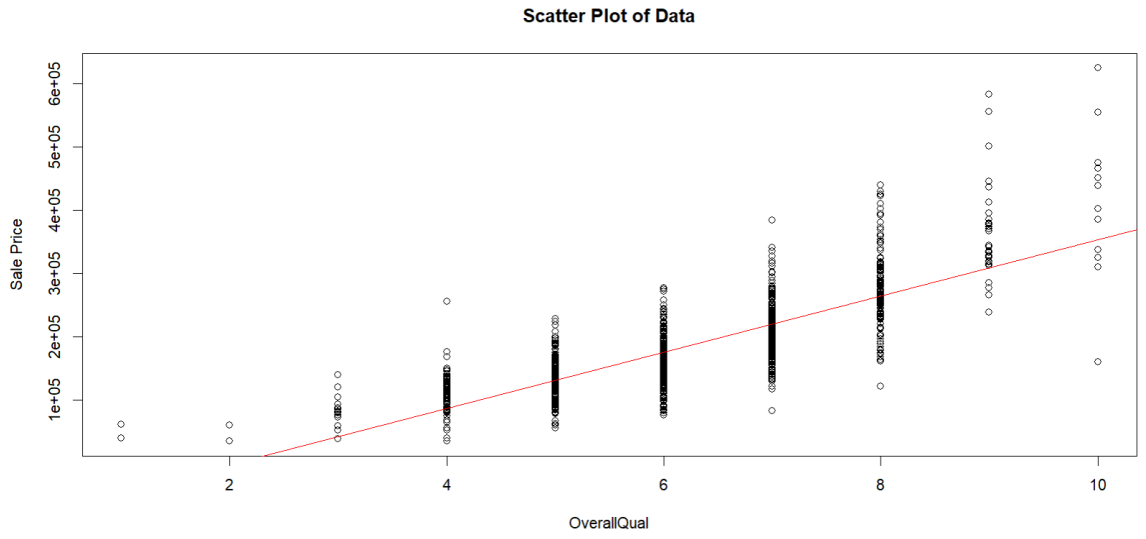


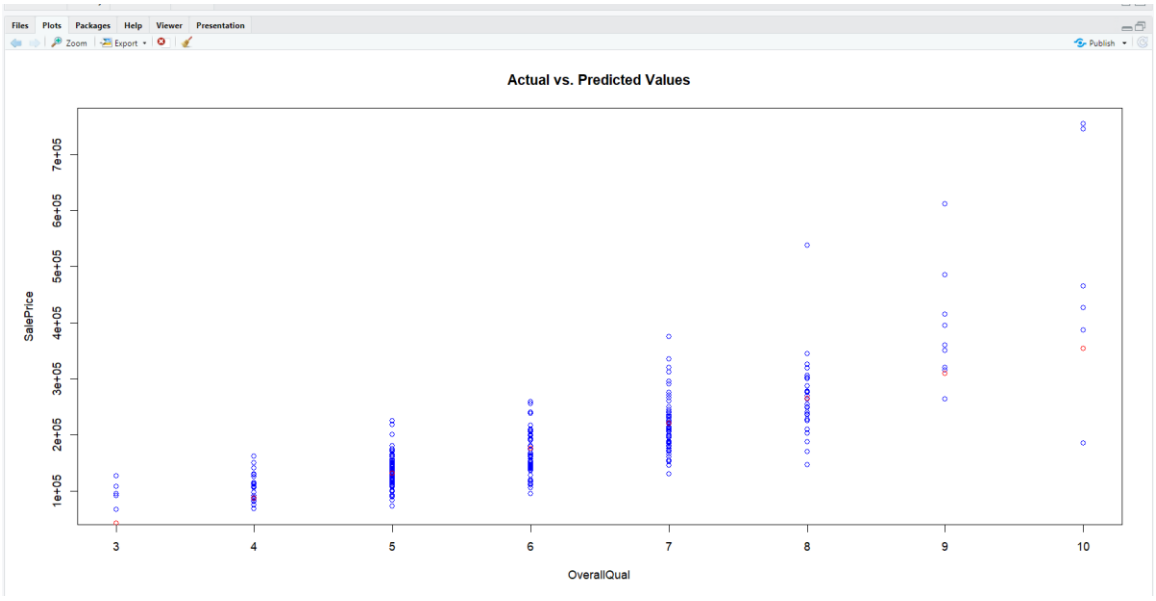

R Graphics: Device 4 (ACTIVE)
File History Resize
Scatter Plot of Data
Sale Price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
0
1
2
3
4
GarageCars


R Graphics: Device 4 (ACTIVE)
File History Resize
Actual vs. Predicted Values
SalePrice
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
0.0
0.5
1.0
1.5
2.0
2.5
3.0
GarageCars


Scatter Plot of Data
Sale Price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
1000
2000
3000
4000
5000
GrLivArea


R Graphics: Device 5 (inactive)
File History Resize
Actual vs. Predicted Values
SalePrice
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
1000
2000
3000
4000
GrLivArea


Scatter Plot of Data
Sale Price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
2
4
6
8
10
OverallQual


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Actual vs. Predicted Values
SalePrice
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
3
4
5
6
7
8
9
10
OverallQual

SVM

1st experiment:

- **dataset**: numerical data
- threshold data with correlation > 0.22
- **Parameters**:
    - trainControl(method="cv", number=15)

C	RMSE	Rsquared	MAE
0.25	34710.28	0.8207592	19097.61
0.50	32152.85	0.8427167	18220.16
1.00	30197.61	0.8587737	17658.04

2nd experiment:

- **dataset**: preprocessed data
- threshold data with correlation > 0.2
- **Parameters**:
    - trainControl(method = "cv", number = 10)

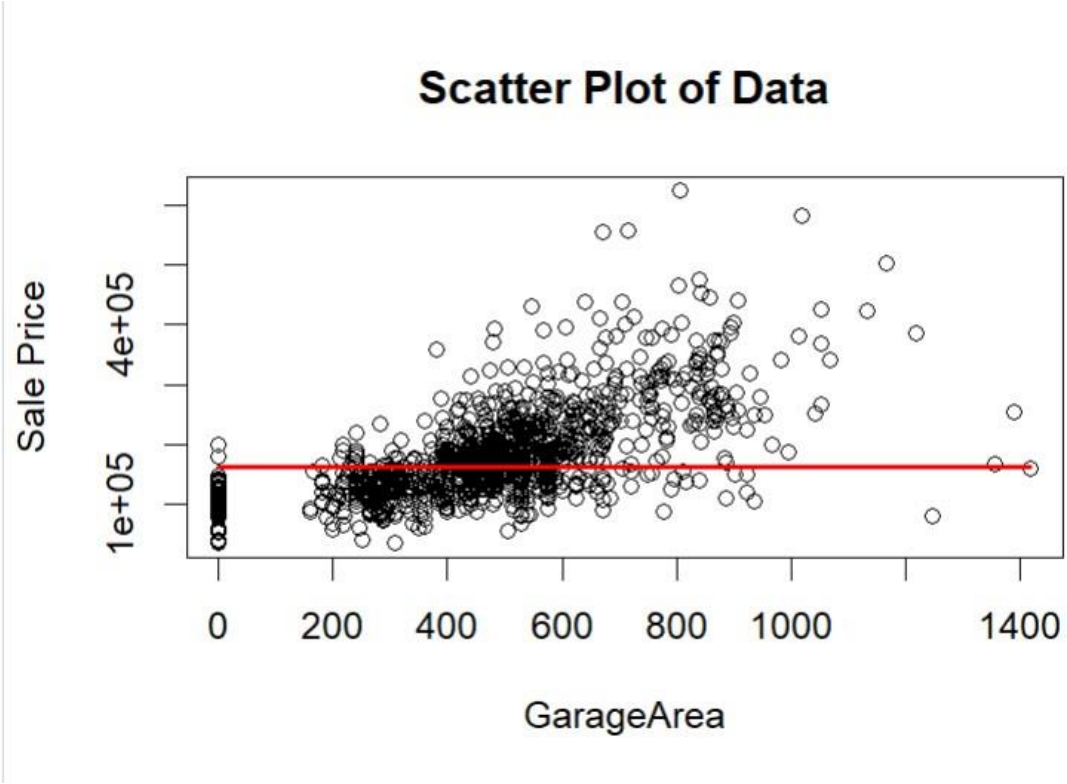
C	RMSE	Rsquared	MAE
0.25	33678.53	0.8388241	18430.15
0.50	31335.34	0.8558461	17559.09
1.00	29642.78	0.8670485	17080.64

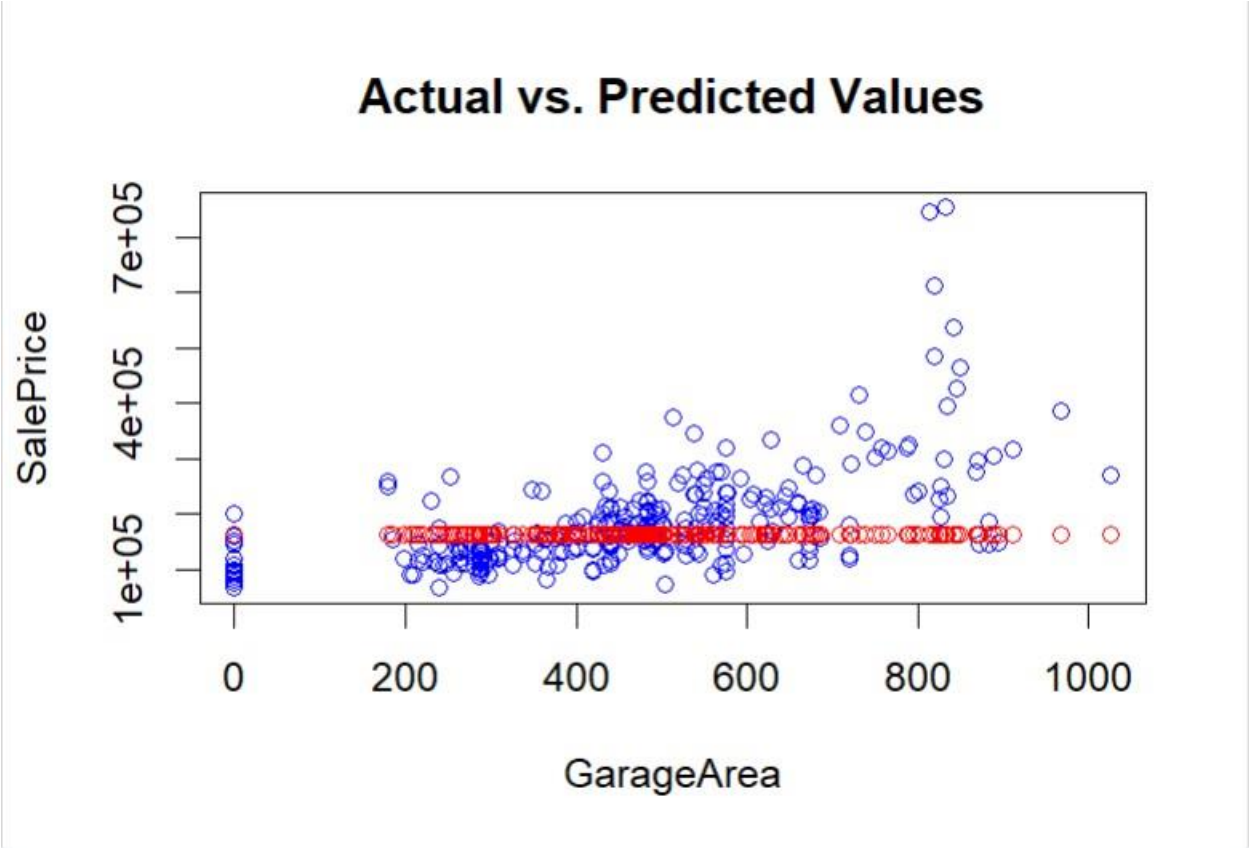
3rd experiment:

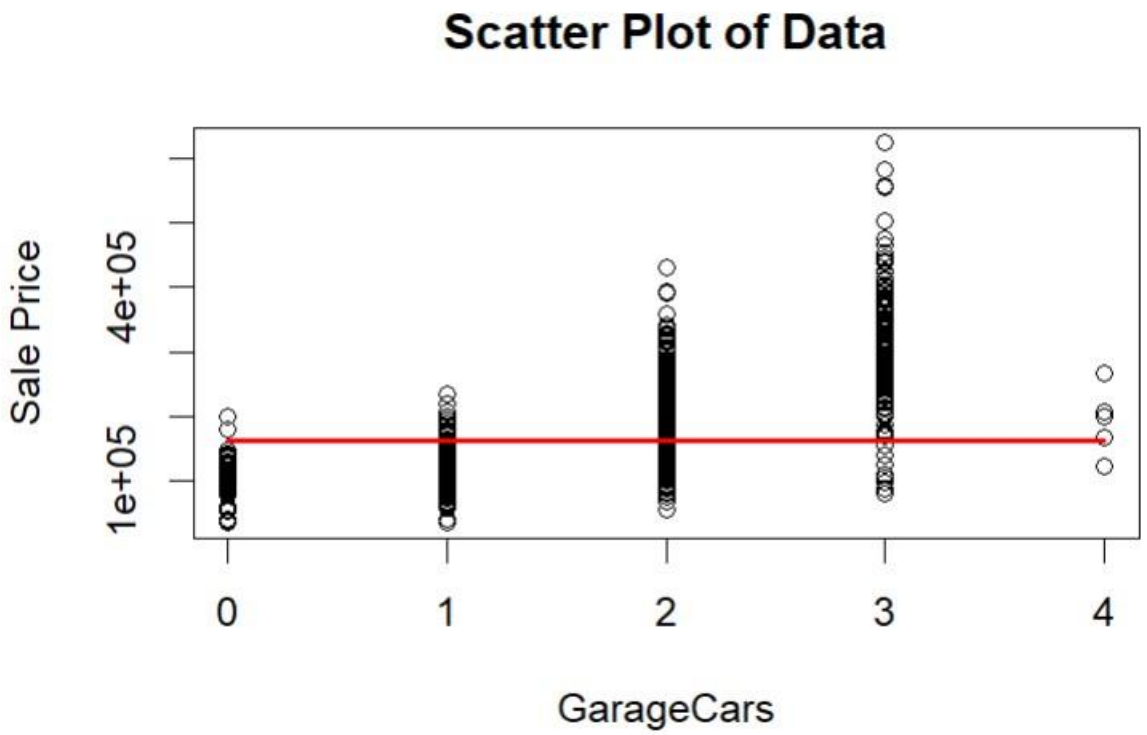
- **dataset**: preprocessed data
- threshold data with correlation > 0.2
- **Parameters**:
  - trainControl(method = "cv", number = 15)

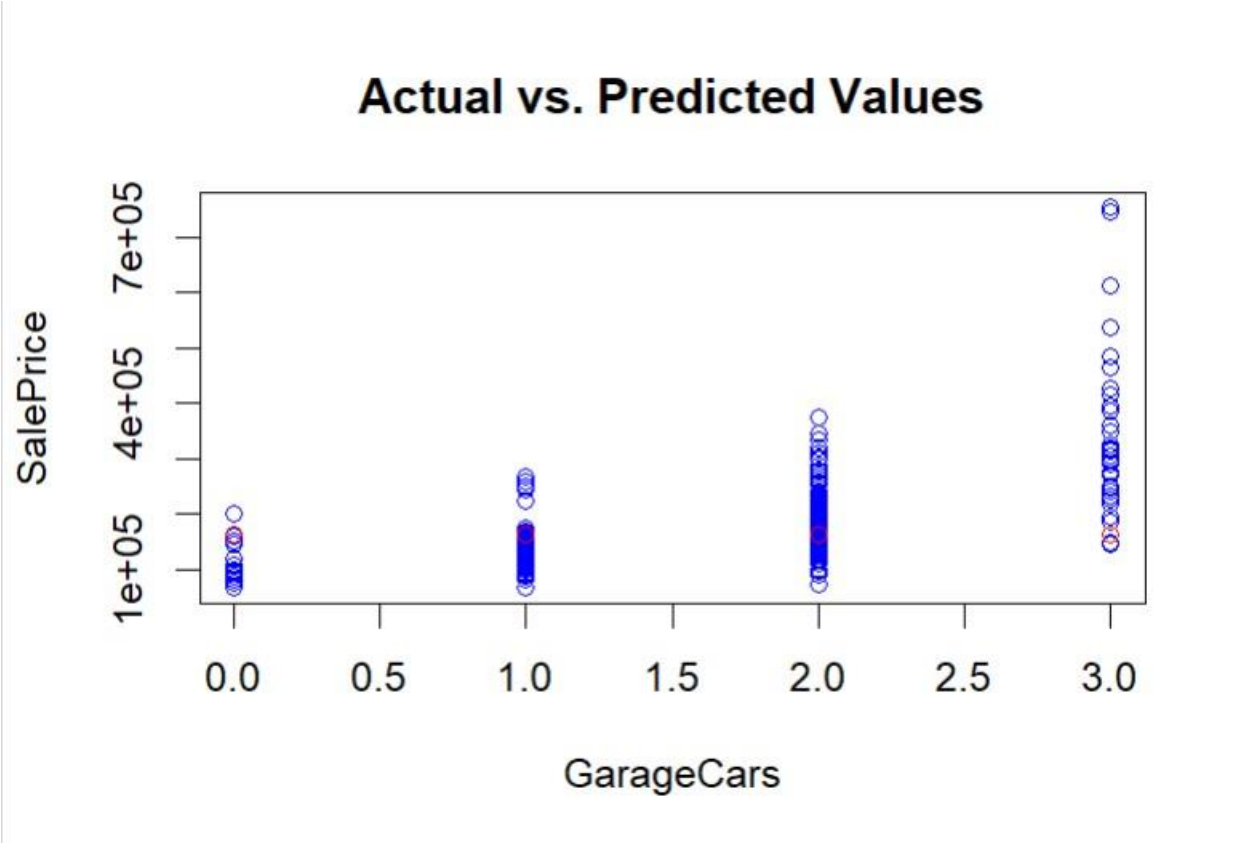
C	RMSE	Rsquared	MAE
0.25	32555.98	0.8466221	18318.22
0.50	30285.18	0.8629919	17460.63
1.00	28702.73	0.8734121	16963.40

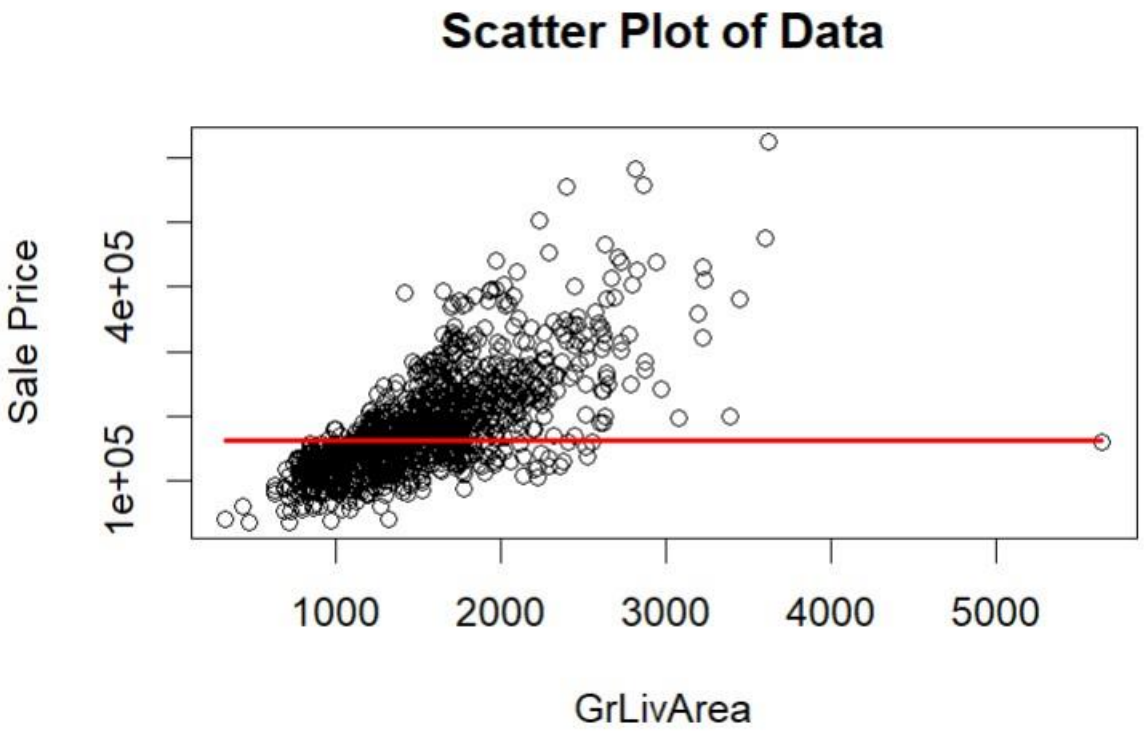
4th experiment:

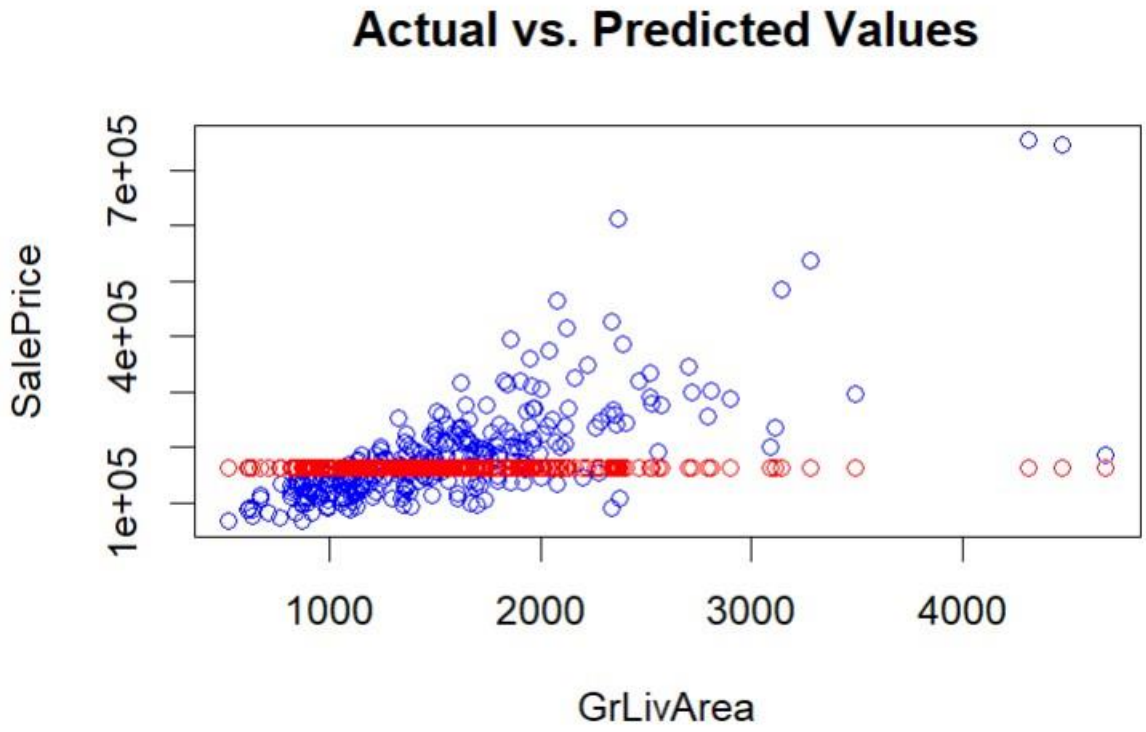


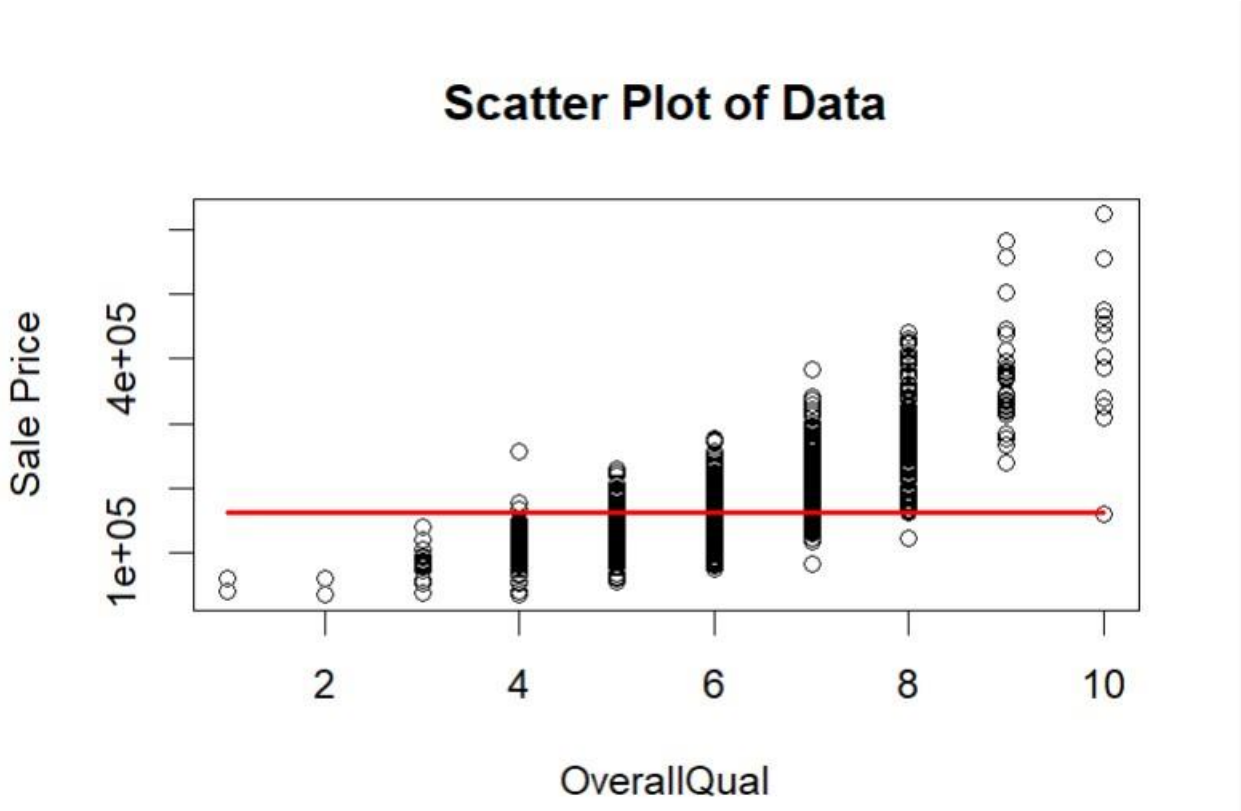


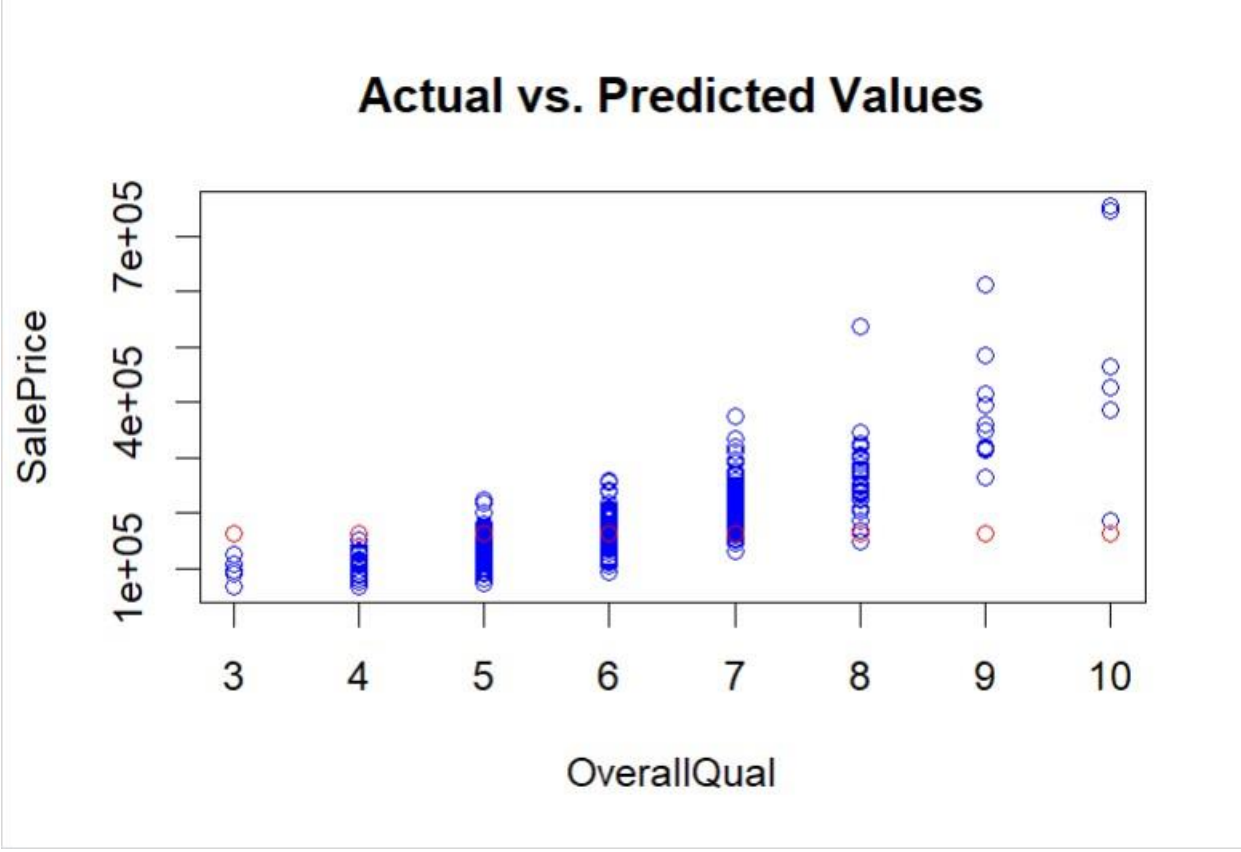

Scatter Plot of Data
Sale Price
1e+05
4e+05
0
1
2
3
4
GarageCars


Actual vs. Predicted Values
SalePrice
1e+05
4e+05
7e+05
0.0
0.5
1.0
1.5
2.0
2.5
3.0
GarageCars


Scatter Plot of Data
Sale Price
1e+05
4e+05
1000
2000
3000
4000
5000
GrLivArea


Actual vs. Predicted Values
SalePrice
1e+05
4e+05
7e+05
1000
2000
3000
4000
GrLivArea


Scatter Plot of Data
Sale Price
1e+05
4e+05
2
4
6
8
10
OverallQual


Actual vs. Predicted Values
SalePrice
1e+05
4e+05
7e+05
3
4
5
6
7
8
9
10
OverallQual

Random Forest:

Version 1

Steps:

1) Divide data into (X,Y)

2) Y is column (“SalesPrice”)

3) X is columns from [1 : 10]

4) Standard scaller for big data (column)

5) Install. Package (“caret”) to divide the data into train and test.

6) Install. Package(‘random forest’) that support vector machine (regression)

7) Divide data into (70 :30)

8) select columns to standardize
9) Fit for random forest model
   Type of randomforest is regression

10) Prediction for model as total

11) Calculate an accuracy using mean squared error

12) **MSE** = 1388837579
13) RMSE <- sqrt(MSE)
14) **RMSE** = 37267.11
15)

Version 2

Select columns to standardize

```
cols_to_scale <- c("GarageArea", "GarageCars", "TotRmsAbvGrd",
                   "FullBath","GrLivArea","X1stFlrSF","TotalBsmtSF",
                   "OverallQual")
```

RMSE = 27280.33

Version 3

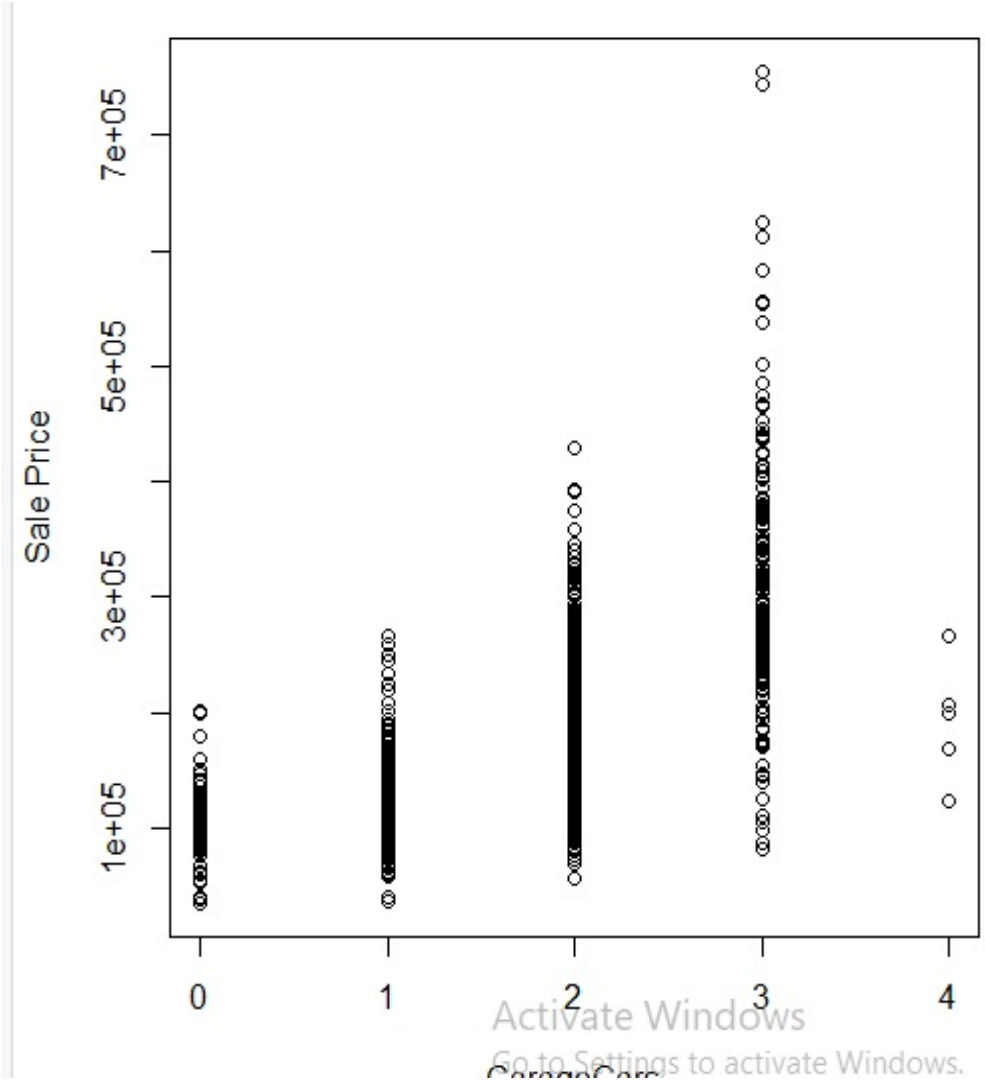
Select columns to standardize

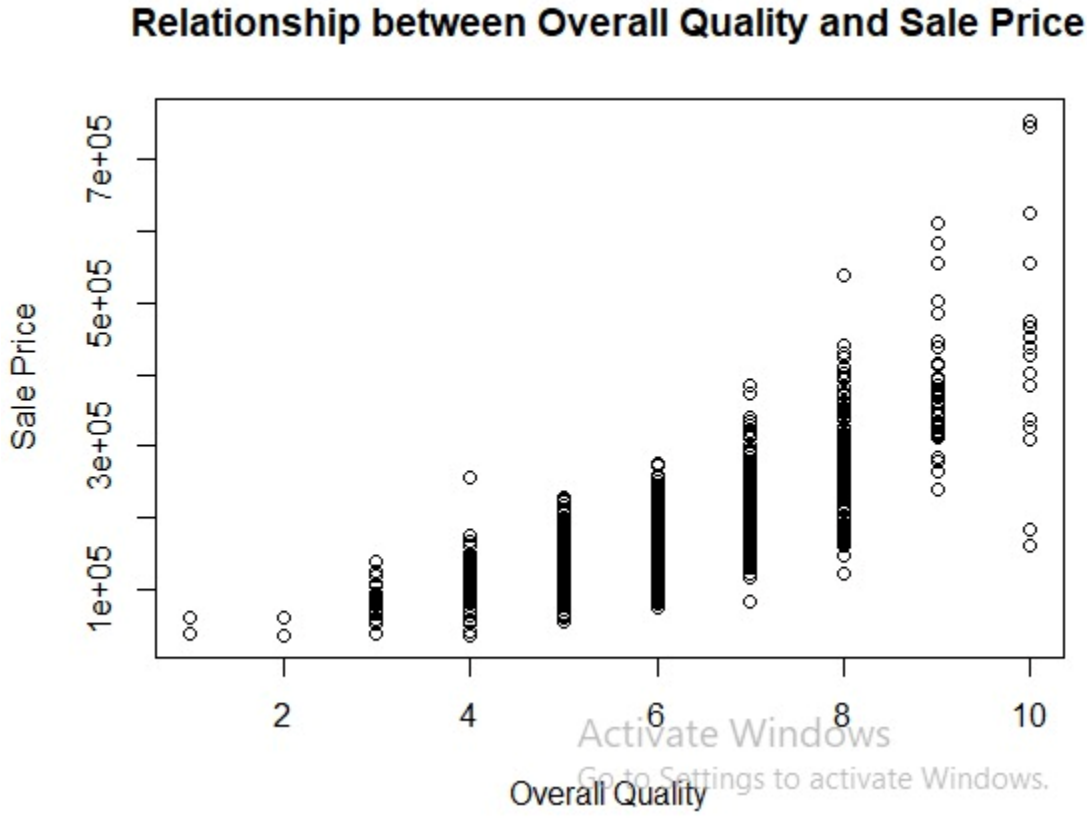
```
cols_to_scale <- c("OverallQual","GrLivArea","GarageCars",
"GarageArea", "TotalBsmtSF", "X1stFlrSF","FullBath",
"TotRmsAbvGrd","GarageYrBlt","MasVnrArea","Fireplaces","BsmtFin
SF1","LotFrontage","WoodDeckSF","OpenPorchSF","HalfBath",
"LotArea")
```

RMSE = 28568.46

Version 4

Visualization :




Relationship between Overall Quality and Sale Price
Sale Price
1e+05
3e+05
5e+05
7e+05
2
4
6
8
10
Overall Quality
Activate Windows
Go to Settings to activate Windows.