Midterm project, group 6

Ruoyuan Qian

1 Introduction

1.1 Objective

In this report, the focus is among eight methods (multiple linear regression, ridge regression, lasso regression, elastic regression, principal component regression (PCR), k-nearest neighbors algorithm (k-NN), generalized additive model (GAM), multivariate adaptive regression spline (MARS)), which one is the best to predict sale price of house for the particular dataset and which predictors are most influential for the response SalePrice.

1.2 Data cleaning

Missing value is checked for each predictor and predictors with the number of missing value greater than 500 are excluded from the dataset. At the meantime, predictors with many zeros or near-zero observations are removed as well. Finally, NA's are dropped from the remaining data.

2 Exploratory data analysis (EDA)

The distribution of response SalePrice ($) is checked (Fig. 1), as we can see, it is continuous variable with a right skewed shape. Since all methods in report do not need normal distribution assumption, so the original value of response can be used in model fitting.

Scatter plots are checked for numeric variables (Fig. 2), since I treated integers as continuous variables, so gaps are introduced in some of the scatter plots, such as "GarageCars", "MoSold", "YrSold", "BsmtFullBath". There is a non-linear trned in "GarageYrBlt", "BsmtUnfSF", "YearBuilt".

Bar plots are shown for categorical variables (Fig. 3). Some categories are not equally distributed within each predictors, like "MSZoning", "RoofStyle", "GarageType", "PavedDrive", "SaleType", "Condition1", "CentralAir", "SaleCondition", "Electrica."

Correlations between numeric predictors are visualized by heat plot (Fig. 4).

3 Models

There are 55 predictors in total including 28 numeric predictors and 27 categorical predictors. The data for all methods is the same to keep the ability of comparison. Data is scaled and standardized in model fitting. All categorical variables are transformed to factor variavles.

3.1 Linear Methods

3.1.1 Multiple linear regression (MLR)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon$$

$\beta's$ are estimated by least squared estimation:

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})$$

, where $\widehat{Y} = \widehat{\beta_0} + \widehat{\beta_1}\widehat{X_1} + \ldots + \widehat{\beta_p}\widehat{X_p}$. Although MLR requires a Gaussian error for any inference, the report is focus on the ability of prediction, so we don't have to do transformation for response variable.

3.1.2 Ridge regression

The ridge coefficient estimation is the minimium of the loss function:

$$min(RSS + \lambda \sum_{j=1}^{p} \beta_j^2)$$

All coefficients will shrink when λ increases, but none of them will shrink to zero. So ridge regression will remain all predictors in the final model. In R, $\alpha = 0$ is fixed, and a range of λ is implemented to find the best tuning parameter with the criteria of smallest MSE through cross validation (CV).

3.1.3 LASSO regression

The LASSO coefficient estimation is the minimum of the loss function:

$$min(RSS + \lambda \sum_{j=1}^{p} |\beta_j|)$$

All coefficients will shrink to zero when λ is large enough. LASSO regression will remain a subset of predictors in the final model. In R, $\alpha = 1$ is fixed, selection of best tuning parameter is similar to ridge.

3.1.4 Elastic regression

The elastic coefficient estimation is the minimum of the loss function:

$$min(RSS + \lambda_1 \sum_{i=1}^{n} \beta_j^2 + \lambda_2 \sum_{i=1}^{n} |\beta_j|)$$

All coefficients will not shrink to exact zero when λ increases. In R, α can be changed in [0,1], so combinations of a range of λ and α is implemented to find the best combination of tuning parameters with the criteria of smallest MSE through cross validation (CV).

3.1.5 Principal component regression (PCR)

It includes two steps, dimension reduction and regression.

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j, y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im}$$

The number of principal components (CP) is chosen by CV with smallest MSE.

3.2 Non-linear methods

3.2.1 K-nearest neighbors algorithm (k-NN)

The k nearest points are used to fit the line.

$$\hat{f}(x_0) = Ave(y|x \in N(x_0)) = \sum_{i=1}^{n} w(x_0, x_i) y_i$$

where $w(x, x_i) = I(x_i \in N_k(x))/K$. Tuning parameter, the number of nearset points k is chosen through CV.

3.2.2 Generalized additive model (GAM)

It allows flexible non-linearities in several variables based on their own scatter plot or degree of freedom (DF), if the points are not linear shaped or the DF is greater than 1, then a non-linear term should be considered.

$$g[E(y|X)] = \beta_0 + f_1(X_1) + ... + f_p(X_p)$$

3.2.3 Multivariate Adaptive Regression Spline (MARS)

It is a piecewise linear model while the cut points are selected by algorithm, and then the hinge functions can be written as $(h(x-c), h(c-x))$.

3.3 Results

For LASSO, plot of MSE across a sequence of λ is made (Fig. 5), and the best λ is $e^{7.035}$. For ridge model, plot of MSE across a sequence of λ (Fig. 6) shows that the best tuning parameter is $\lambda = e^{10.10}$. As for elastic model, 750 combinations of α and λ are checked (Fig. 7), the best pair is $\alpha = 0, \lambda = e^{10.10}$, which is the same as ridge, so in the model comparison, only ridge regression will be presented. For PCR, 54 principle components (PC) are tested, and the best number of principle component (PC) is 26 through MSE (Fig. 8).

For k-NN model, after testing a sequence of k from 5 to 43, the best tuning parameter is equal to 11. For GAM model, "train" function is implemented and 20 out of 54 predictors are tested for non-linear relation to response (Fig. 9). For MARS model, 10 cut points are used to fit the model (Fig. 10).

4 Conclusion

All models are compared through MSE (Tab. 1, Fig 11, Fig. 12). With differnt metrics, the best model is different. GAM model obtained the smallest median MSE while MARS model obtained the smallest average MSE. Here we use the average MSE as the final criteria. The best model is MARS. K-nn method obtain the largest MSE among the all.

There are 10 cut points in MARS model, they are in "OverallQual", "GrLivArea", "X2ndFlrSF", "YearBuilt", "BsmtFinSF1", "LotArea", "OverallCond", "X1stFlrSF", "TotalBsmtSF", "SaleCondition". The coefficients of hinge functions are shown in Tab. 2. According to Fig. 10, except "X1stFlrSF" and "SaleCondition", all predictors have increasing trends when response rises. "X1stFlrSF" has a dereaseing trend all the time when the sale price increases.

Moreover, the top 10 most important variables for the MARS model are checked in Tab.3, for a decreasing order of contribution, they are: "OverallQual", "GrLivArea", "YearBuilt", "BsmtFinSF1", "X1stFlrSF", "X2ndFlrSF", "OverallCond", "LotArea", "TotalBsmtSF", "SaleCondition".

Heat plot for the top 10 the most important variables is made (Fig. 13), "X2ndFlrSF" and "GrLivArea", "X1stFlrSF" and "TotalBsmtSF", "YearBuilt" and "OverallQual" are highly correlated, respectively.

5 Discussion

As for data interpretation, since the data from the Kaggle does not include the specific meaning for each variable, so I cannot interpret the model in a more detailed way. Besides, as for categorical data, I changed them into integer values as 1, 2, 3, . . . , otherwise, they are seemed as dummy variables in model.matrix function which include lots of zero and cause warnings in model training.

Appendix - Figures and Tables

Table 1 MSE of all methods through cross validation

column	mean	sd	median	min	max	range
Lm	37505.04	13940.946	33542.84	22794.36	72849.29	50054.93
LASSO	36543.47	12986.560	33204.86	22150.00	69494.01	47344.00
Ridge	36278.35	12238.432	32908.06	21886.61	66422.74	44536.14
PCR	36483.11	12114.243	32675.95	23044.04	67262.97	44218.94
Knn	38064.71	8516.354	37404.23	25491.27	58379.22	32887.94
GAM	35926.26	23224.696	27382.22	19331.35	136719.62	117388.26
MARS	33038.51	9329.652	31375.96	21072.25	58224.52	37152.27

Table 2 Hinge functions and their coefficients in MARS model

	Coefficient
(Intercept)	3.141179e+05
h(OverallQual-7)	4.077041e+04
h(7-OverallQual)	-9.109677e+03
h(2945-GrLivArea)	-5.752370e+01
h(X2ndFlrSF-1360)	3.718084e+02
h(YearBuilt-2007)	1.563304e+04
h(2007-YearBuilt)	-5.721088e+02
h(BsmtFinSF1-763)	4.820646e+01
h(763-BsmtFinSF1)	-1.379454e+01
h(LotArea-20431)	4.618026e-01
h(20431-LotArea)	-1.953780e+00
h(7-OverallCond)	-1.093203e+04
h(X1stFlrSF-2121)	-3.447610e+02
h(TotalBsmtSF-1626)	7.145797e+01
h(1626-TotalBsmtSF)	-1.678924e+01
h(SaleCondition-4)	1.813065e+04
h(4-SaleCondition)	3.657226e+03

Table 3 Top 10 most important variables in MARS model

	Overall
OverallQual	100.00000
GrLivArea	61.92524
YearBuilt	44.28836
BsmtFinSF1	32.18829
X1stFlrSF	32.18829
X2ndFlrSF	30.90316
OverallCond	23.03836
LotArea	19.41104
TotalBsmtSF	16.61787
SaleCondition	11.65620

Figure 1 Distridution of response (SalePrice)

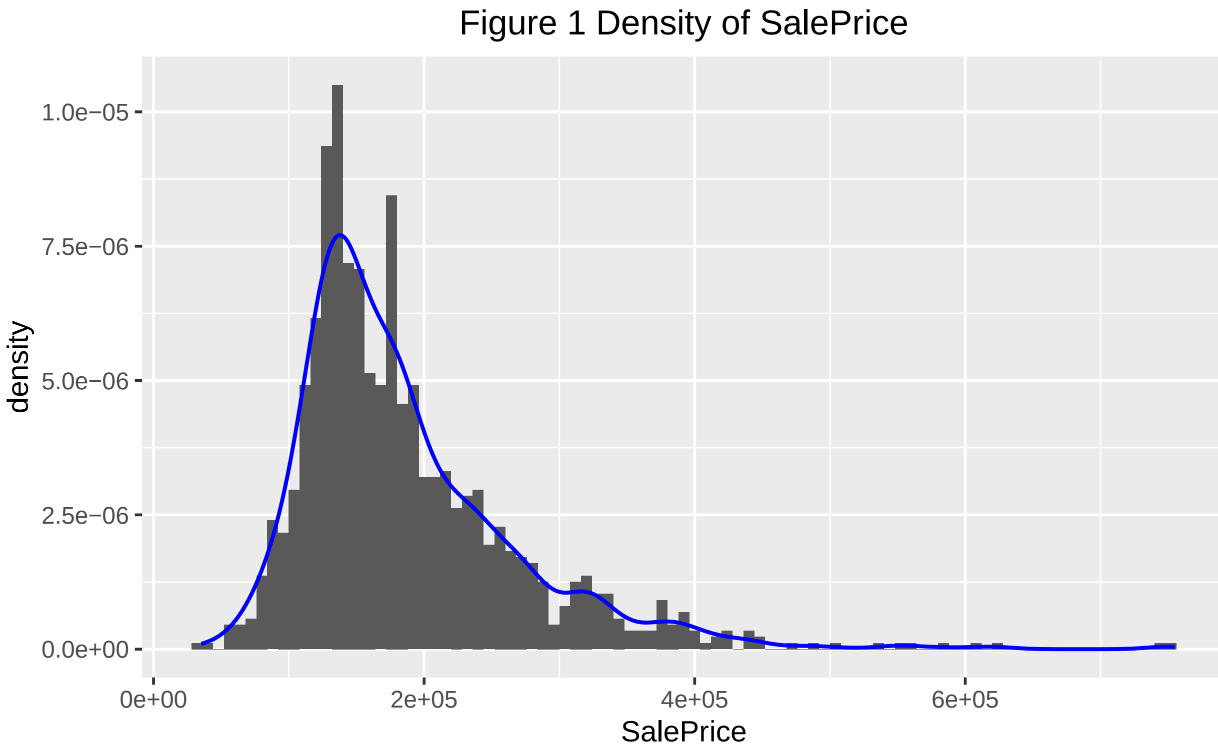


Figure 2 Scatter plots of continuous predictors against SalePrice

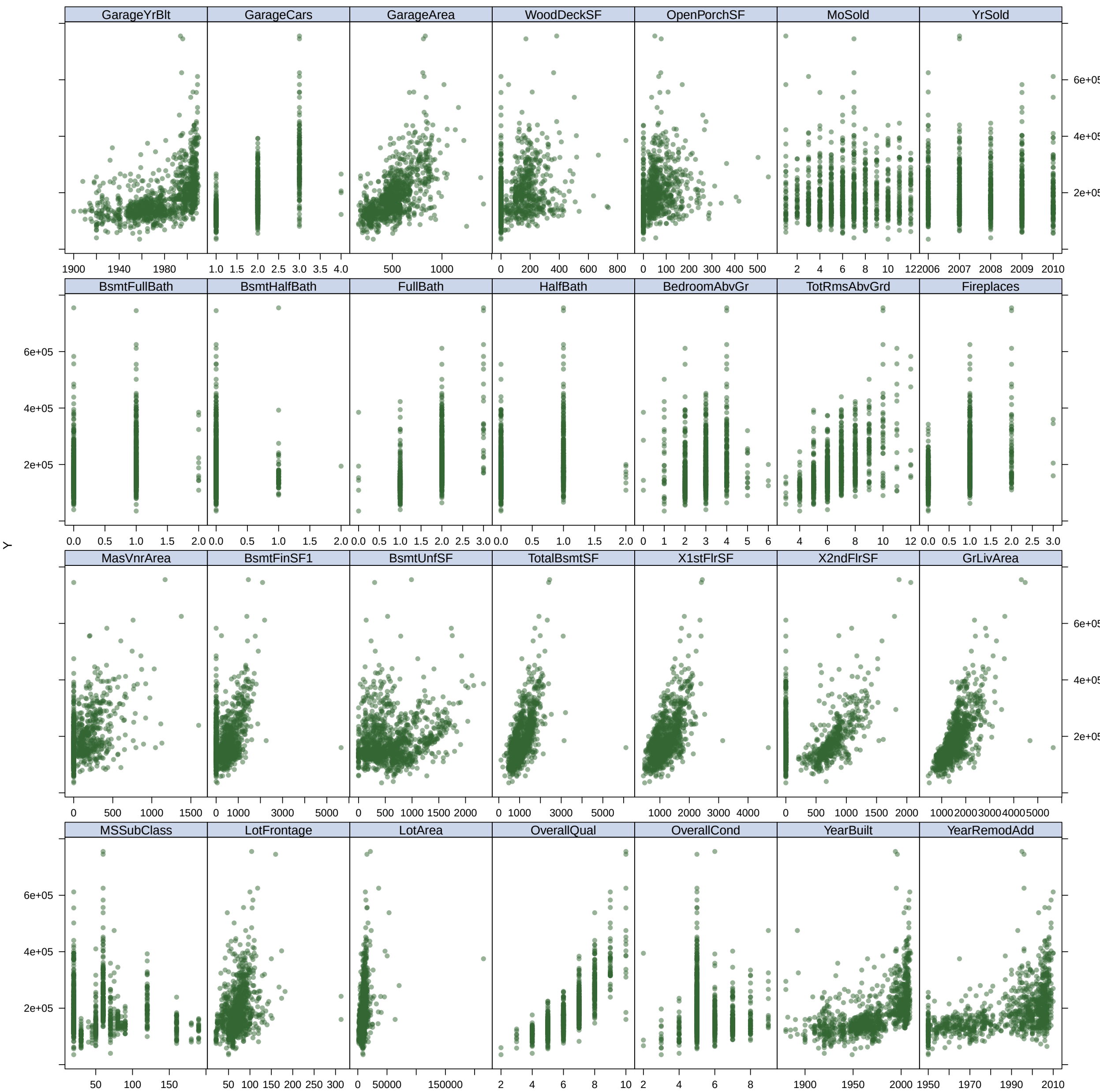


Figure 3 Bar plots of categorical predictors

Figure 4 Heat map for all predictors

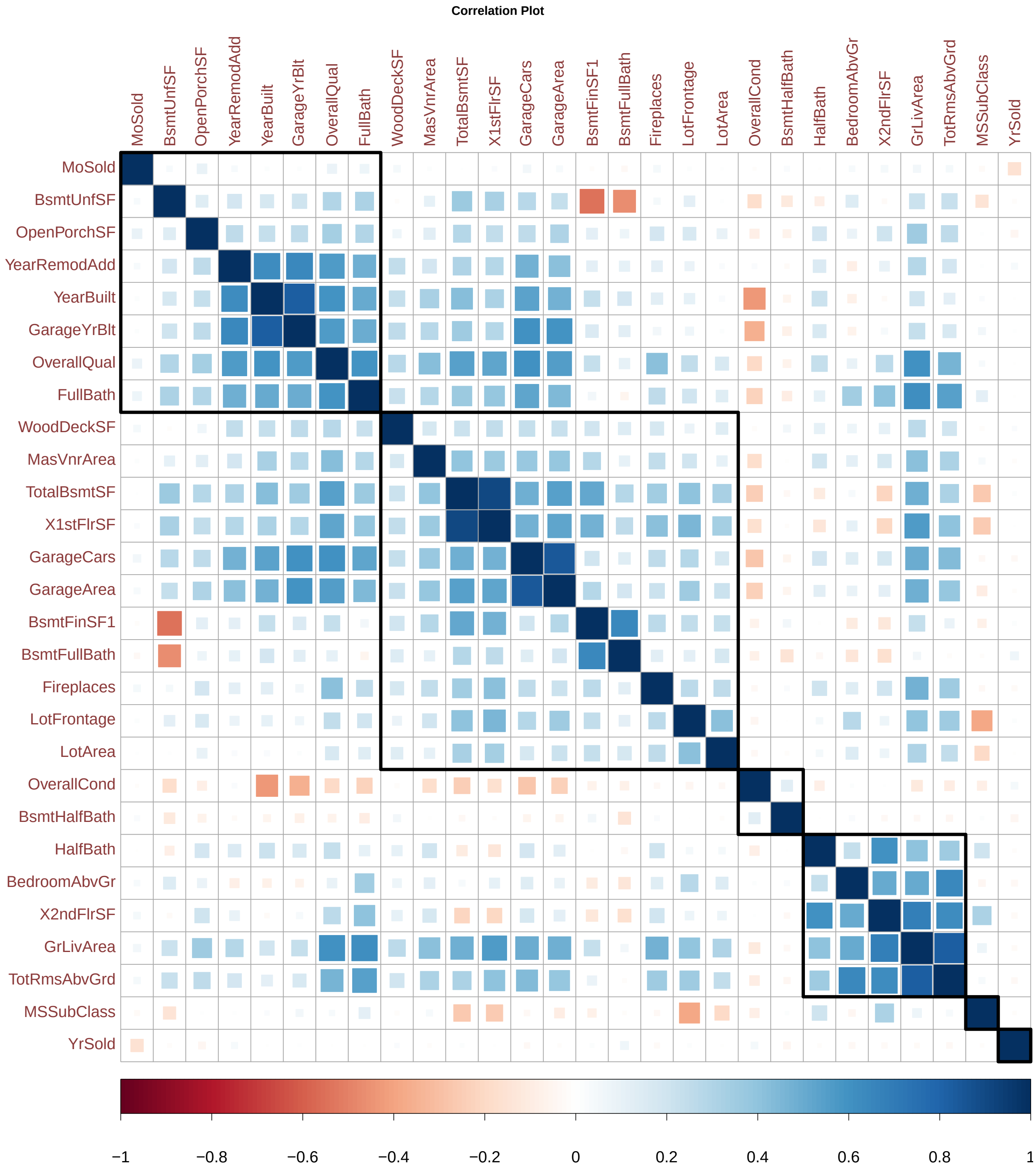


Figure 5 LASSO

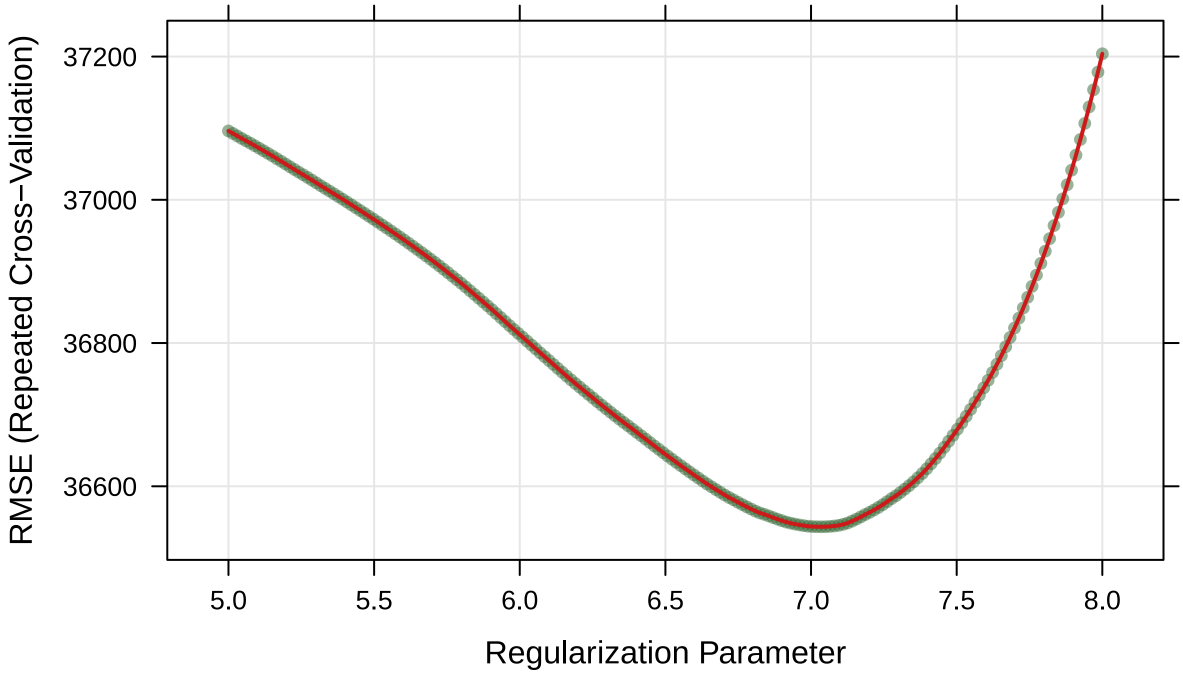


Figure 6 Ridge

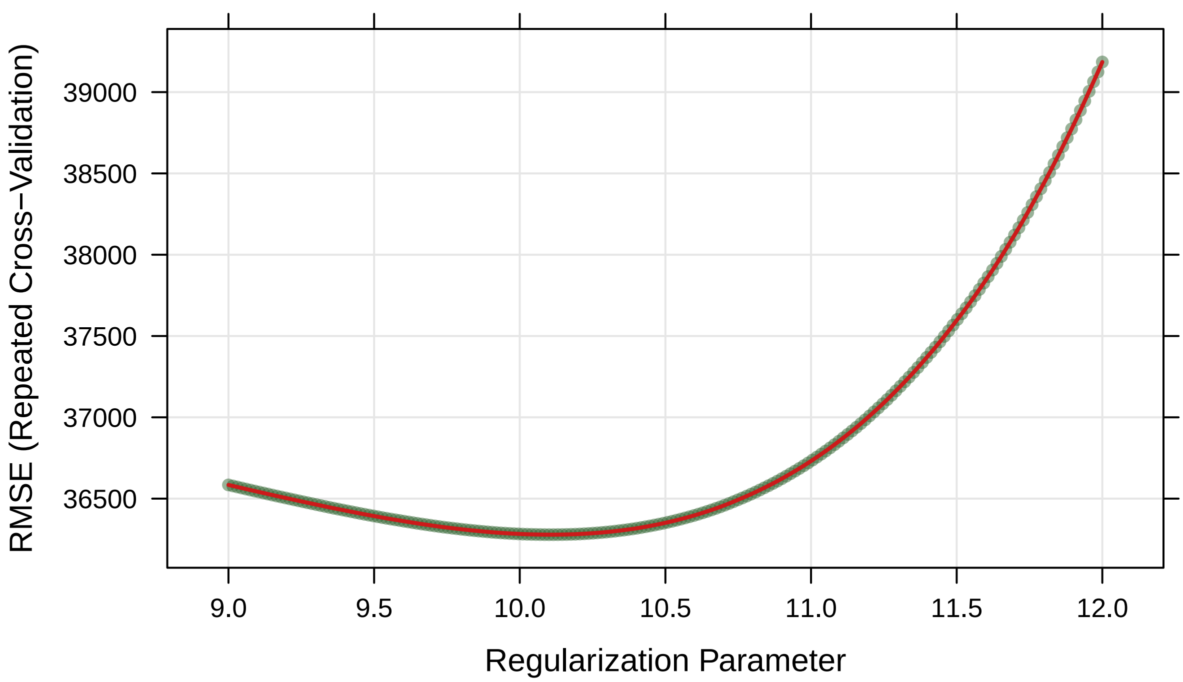


Figure 7 Elastic

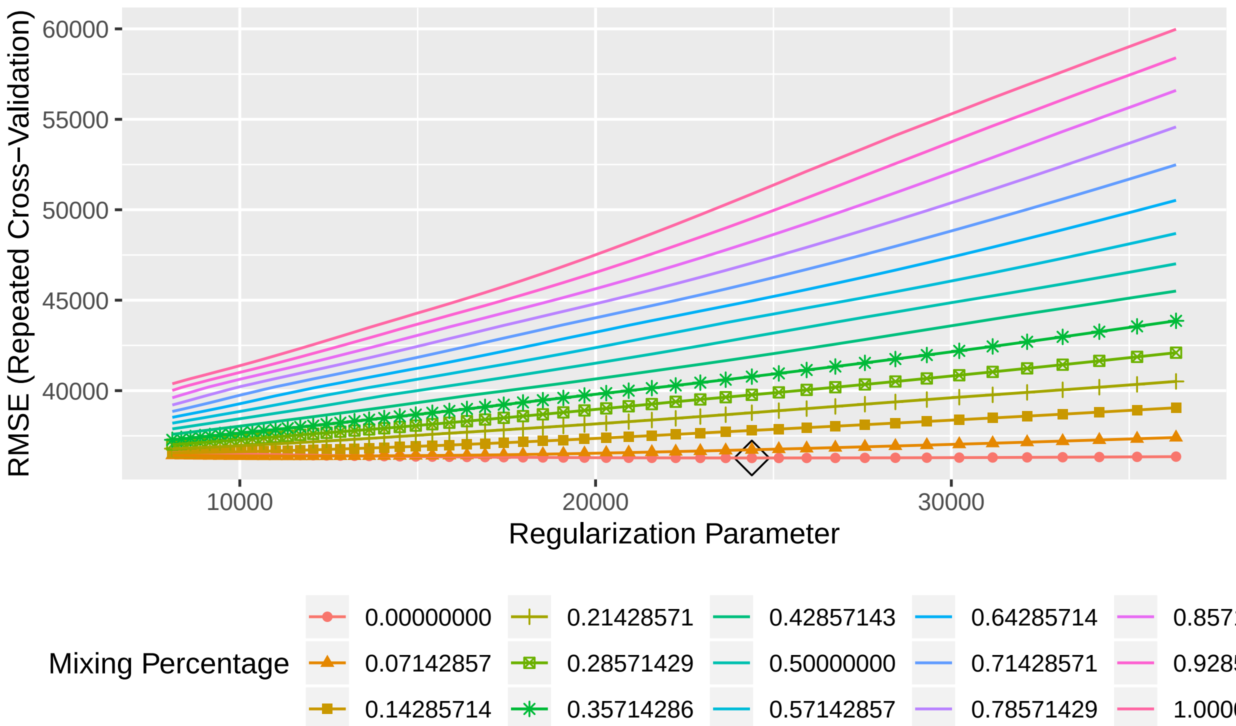


Figure 8 PCR

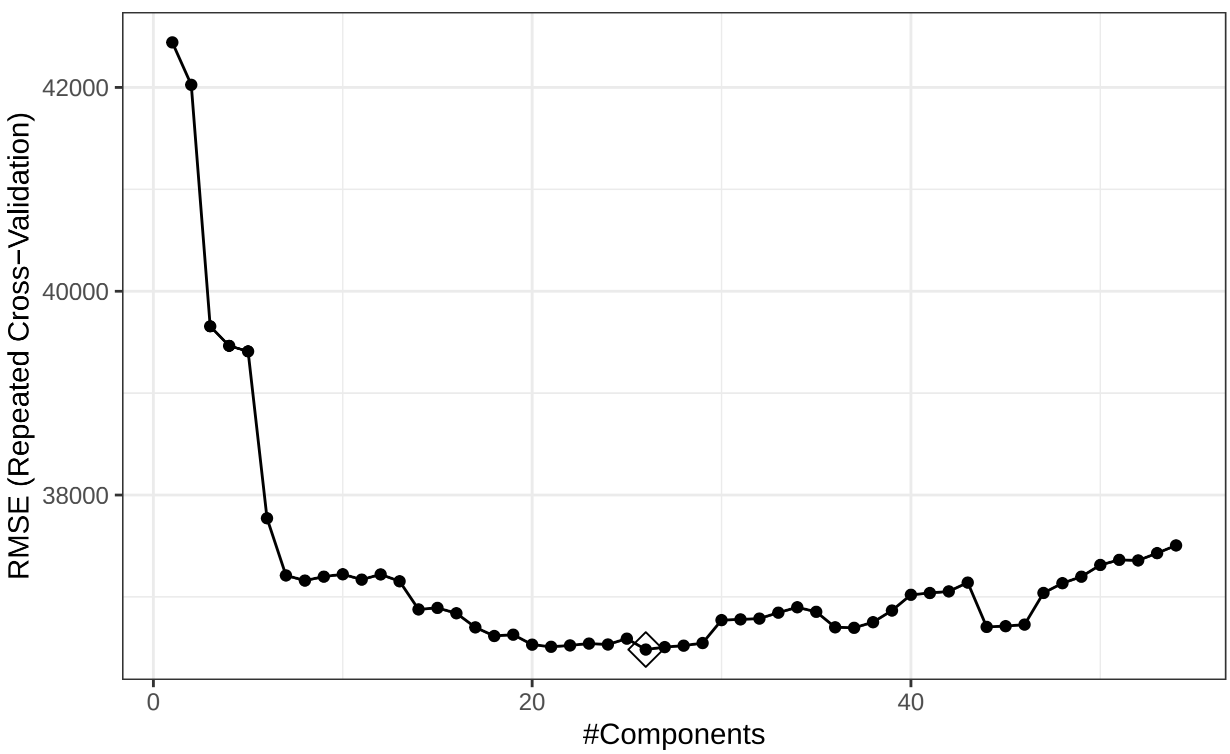
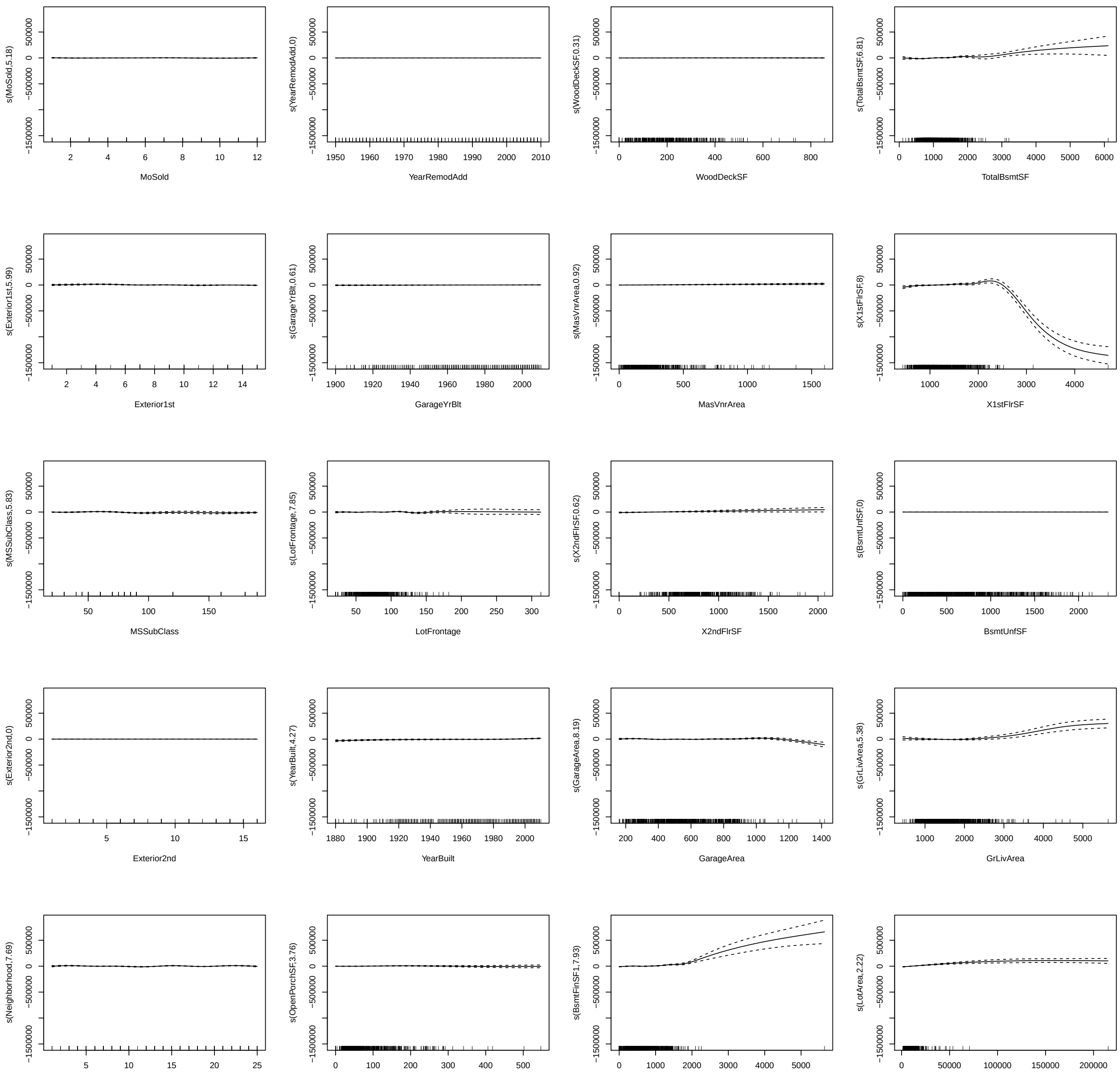


Figure 9 GAM



s(MoSold,5.18)
MoSold
s(YearRemodAdd,0)
YearRemodAdd
s(WoodDeckSF,0.31)
WoodDeckSF
s(TotalBsmtSF,6.81)
TotalBsmtSF
s(Exterior1st,5.99)
Exterior1st
s(GarageYrBlt,0.61)
GarageYrBlt
s(MasVnrArea,0.92)
MasVnrArea
s(X1stFlrSF,8)
X1stFlrSF
s(MSSubClass,5.83)
MSSubClass
s(LotFrontage,7.85)
LotFrontage
s(X2ndFlrSF,0.62)
X2ndFlrSF
s(BsmtUnfSF,0)
BsmtUnfSF
s(Exterior2nd,0)
Exterior2nd
s(YearBuilt,4.27)
YearBuilt
s(GarageArea,8.19)
GarageArea
s(GrLivArea,5.38)
GrLivArea
s(Neighborhood,7.69)
Neighborhood
s(OpenPorchSF,3.76)
OpenPorchSF
s(BsmtFinSF1,7.93)
BsmtFinSF1
s(LotArea,2.22)
LotArea

Figure 10 MARS

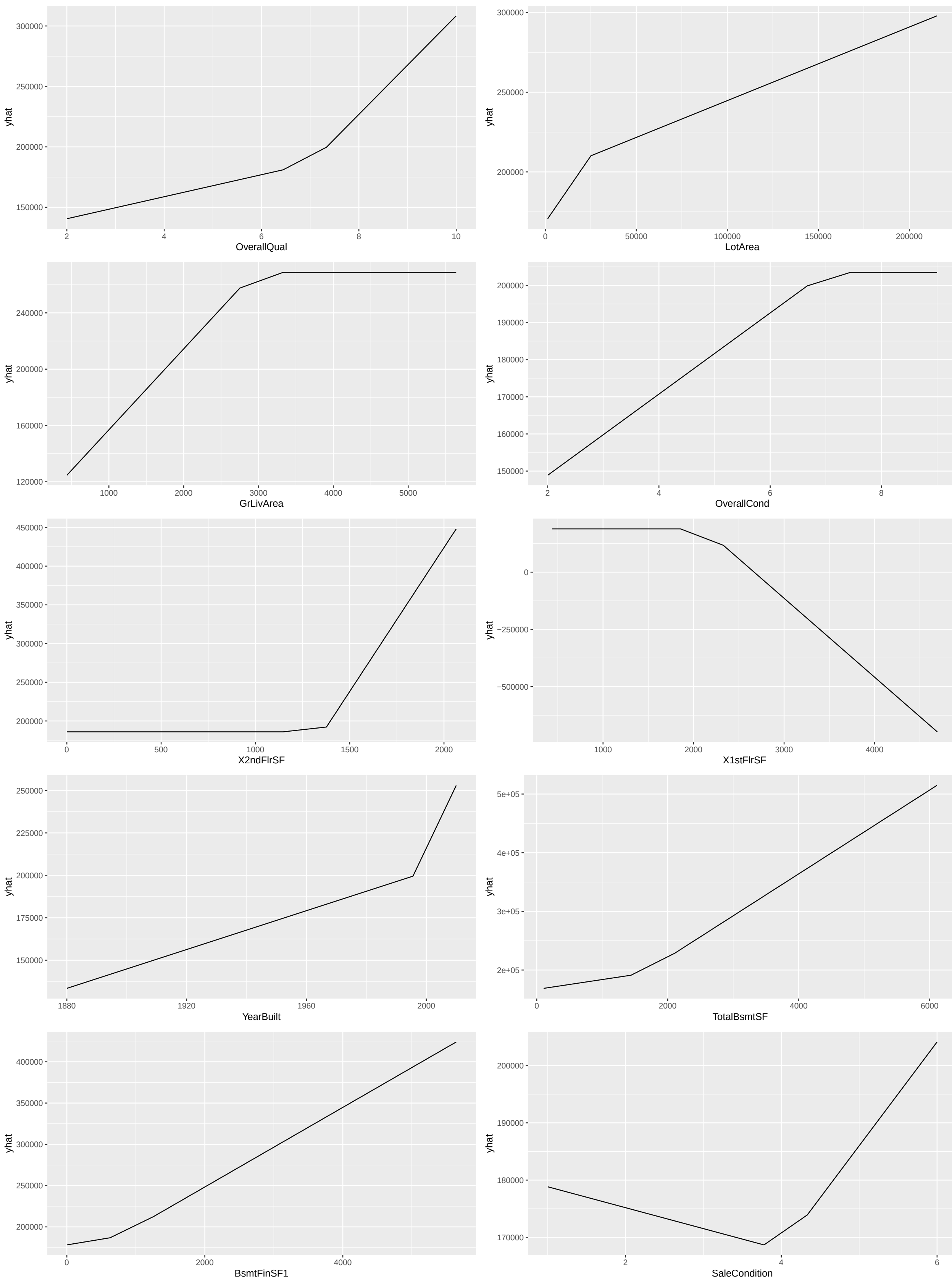


Figure 11 Box plot of all methods through CV

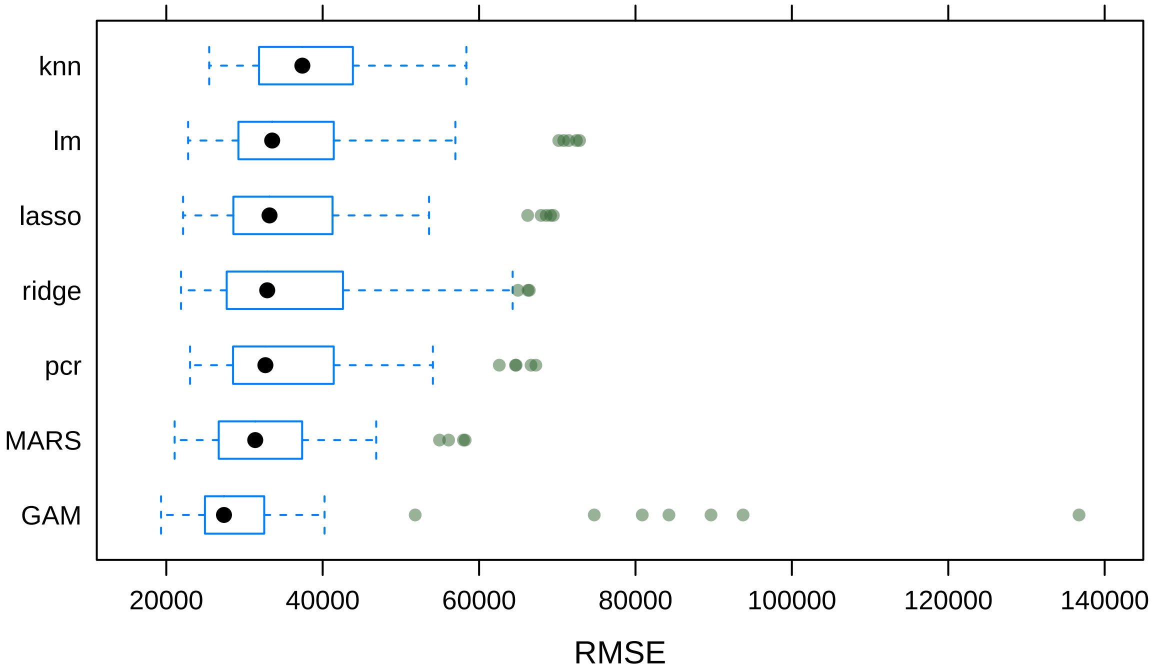


Figure 12 Box plot of all methods through CV

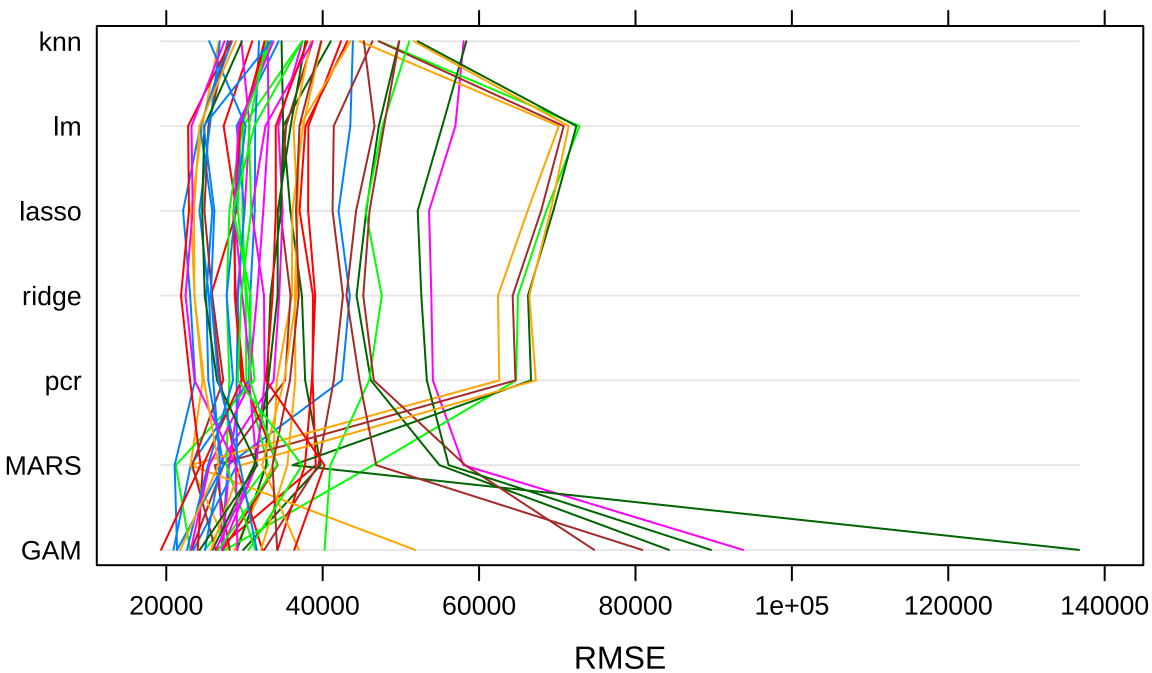
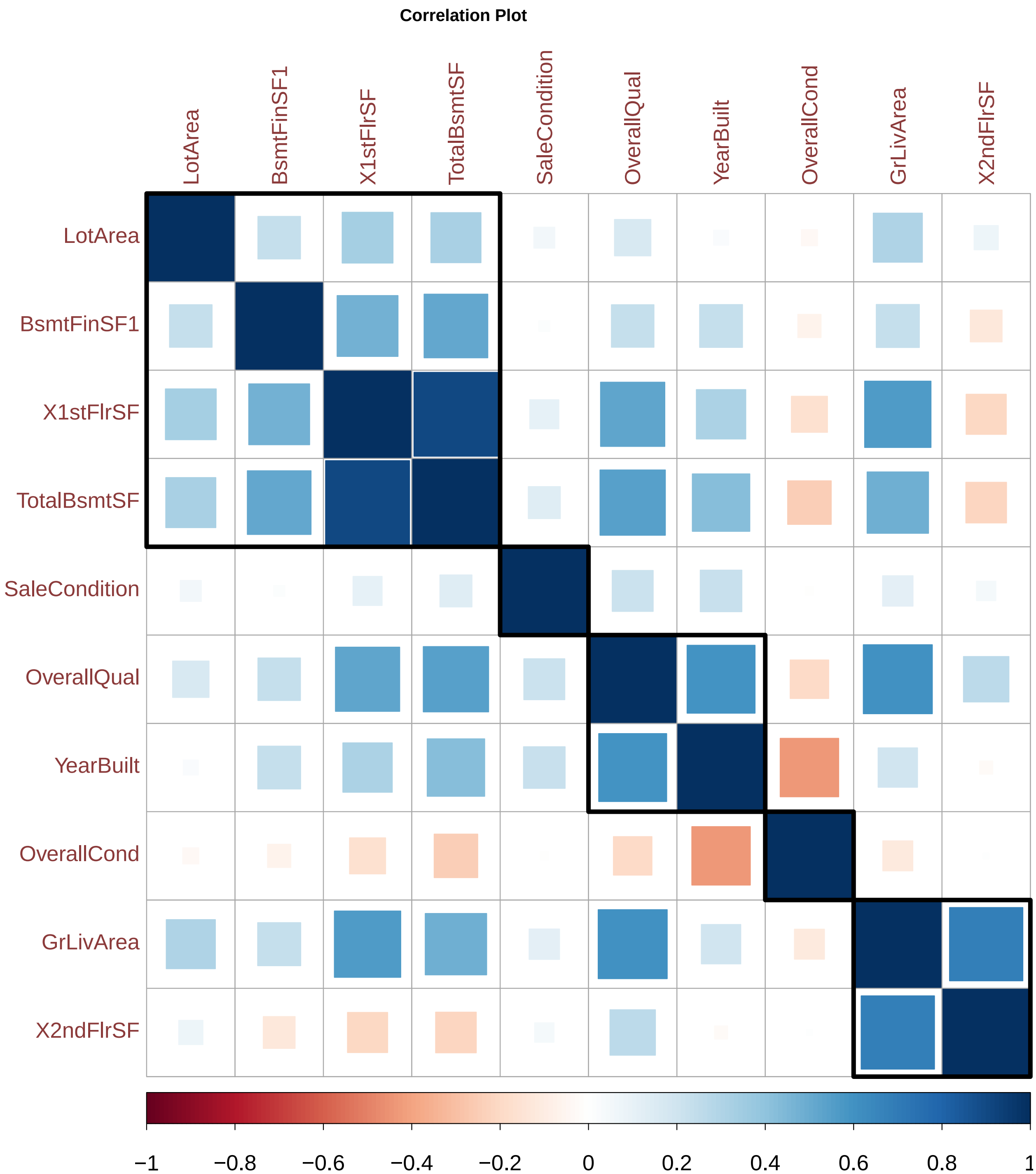


Figure 13 Heat map for top 10 most important variables



Appendix - R code

Data imput

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(caret)
library(ggplot2)
library(Rmisc)
library(corrplot)
library(FNN)
library(pdp)
library(earth)
library(sandwich)
library(stargazer)

house = read.csv(file = "train.csv")
```

Data cleaning

```
sum_na = function(x){
  sum = sum(is.na(x))
  sum}

missing_var = map(house,sum_na) %>%
  as.data.frame() %>%
  pivot_longer(
    Id : SalePrice,
    names_to = "variable",
    values_to = "value"
  ) %>%
  filter(value > 500 ) %>%
  pull(variable)

near_0_var =
  house %>%
  nearZeroVar( names = TRUE)

final_house =
  house %>%
  select(-near_0_var,-missing_var,-Id) %>%
  drop_na()
```

Visualization

```
# Distribution of response
density_sale =
ggplot(final_house, aes(x = SalePrice, ..density..)) +
```

```
  geom_histogram(binwidth = 8000) +
  geom_line(stat = 'density',size = 0.7,color = "blue")+
  ggtitle("Figure 1 Density of SalePrice") +
  xlab("SalePrice") +
  theme(plot.title = element_text(hjust = 0.5))

# Scatter plot for continuous variables
numeric_var_index =
 final_house%>%
  map(.,is.numeric) %>%
  unlist() %>%
  as.vector()

x <- model.matrix(SalePrice~.,
                  final_house[,which(numeric_var_index == TRUE)])[,-1]
y <- final_house$SalePrice

theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)
scatter_plot =
  featurePlot(x, y, plot = "scatter", labels = c("","Y"),
            type = c("p"), layout = c(7, 4))

# Bar plot for categorical variables
variable_name = names(final_house %>%
  select(which(numeric_var_index == FALSE)))

dataframe =
  final_house %>%
  select(which(numeric_var_index == FALSE))

plots = function(dataframe){
variable_name = names(dataframe)

summary =
final_house %>%
  select(which(numeric_var_index == FALSE)) %>%
  pivot_longer(
   everything(),
   names_to = "variable",
   values_to = "category"
  ) %>%
  group_by(variable,category)%>%
  count() %>%
  mutate(n = freq) %>%
  select(-freq)

plot_tem =
```

```r
  summary %>%
  filter(variable == variable_name) %>%
  ggplot(mapping = aes(x = category,
                       y = n,fill = category)) +
   geom_bar(stat = 'identity',position = 'dodge') +
 scale_fill_hue(c = 80)+
 ggtitle(paste("Bar plot of",variable_name))+
  labs(x = variable_name) +
 theme(plot.title = element_text(hjust = 0.5),
       legend.position="right")
plot_tem
  }


plot_name = NULL
for(i in 1: length(dataframe)){
  plot_name_tem = paste("plots(dataframe %>% select(",i,"))",",")
  plot_name = c(plot_name,plot_name_tem)
}


bar_plot = multiplot(
 plots(dataframe %>% select( 1 )) ,
plots(dataframe %>% select( 2 )) ,
plots(dataframe %>% select( 3 )) ,
plots(dataframe %>% select( 4 )) ,
plots(dataframe %>% select( 5 )) ,
plots(dataframe %>% select( 6 )) ,
plots(dataframe %>% select( 7 )) ,
plots(dataframe %>% select( 8 )) ,
plots(dataframe %>% select( 9 )) ,
plots(dataframe %>% select( 10 )) ,
plots(dataframe %>% select( 11 )) ,
plots(dataframe %>% select( 12 )) ,
plots(dataframe %>% select( 13 )) ,
plots(dataframe %>% select( 14 )) ,
plots(dataframe %>% select( 15 )) ,
plots(dataframe %>% select( 16 )) ,
plots(dataframe %>% select( 17 )) ,
plots(dataframe %>% select( 18 )) ,
plots(dataframe %>% select( 19 )) ,
plots(dataframe %>% select( 20 )) ,
plots(dataframe %>% select( 21 )) ,
plots(dataframe %>% select( 22 )) ,
plots(dataframe %>% select( 23 )) ,
plots(dataframe %>% select( 24 )) ,
plots(dataframe %>% select( 25 )) ,
plots(dataframe %>% select( 26 )) ,
          cols=4)

# heat plot for all variables
reg_data = as.data.frame(map(final_house,as.numeric))
```

```r
corrplot(cor(reg_data %>% select(-SalePrice)),title = "Correlation Plot", method = "square", addgrid.co
```

Training models

Multiple linear regression

```r
set.seed(1)
lm.fit <- train(x, y,
                method = "lm",
                trControl = ctrl1,
                preProcess = c("center", "scale"))

tibble("MSE" = lm.fit $ results $ RMSE)%>%
knitr::kable()
```

LASSO

```r
set.seed(1)
lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(5,8, length=200))),
                   preProc = c("center", "scale"),
                   trControl = ctrl1)

LASSO_plot = plot(lasso.fit, xTrans = function(x) log(x))

coe = coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)

as.data.frame(lasso.fit$ results ) %>%
  select(lambda,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  mutate("Number of non-zero coefficient" = length(which(coe[-1] != 0))) %>%
  knitr::kable()
```

Ridge

```r
set.seed(1)
ridge.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(9, 12, length=200))),
                   preProc = c("center", "scale"),
                   trControl = ctrl1)


ridge_plot = plot(ridge.fit, xTrans = function(x) log(x))
ridge_plot

as.data.frame(ridge.fit$ results ) %>%
  select(lambda,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

Elastic

```r
set.seed(1)
enet.fit <- train(x, y,
                  method = "glmnet",
                  tuneGrid =
                    expand.grid(alpha = seq(0, 1, length = 15),
                                lambda = exp(seq(9,10.5, length = 50))),
                  preProc = c("center", "scale"),
                  trControl = ctrl1)
enet.fit$bestTune

elastic_plot = ggplot(enet.fit, highlight = TRUE) +
  theme(legend.position="bottom")
elastic_plot

as.data.frame(enet.fit$ results ) %>%
  select(lambda,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

PCA

```r
set.seed(1)

pcr.fit <- train(x, y,
                  method = "pcr",
                  tuneGrid = data.frame(ncomp = 1:54),
                  trControl = ctrl1,
                  scale = TRUE)

as.data.frame(pcr.fit$ results ) %>%
  select(ncomp,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()

pcr_plot = ggplot(pcr.fit, highlight = TRUE) + theme_bw()
pcr_plot
```

K-nn

```r
set.seed(1)
knnFit <- train(x, y,
                method = "knn",
                trControl = ctrl1,
                preProcess = c("center", "scale"),
                tuneLength = 20)
               #  tuneGrid = tibble(n = 7:10))


as.data.frame(knnFit$ results ) %>%
  select(k,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

GAM

```
set.seed(1)
gam.fit <- train(x, y,
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
                 trControl = ctrl1)

summary(gam.fit$finalModel)
```

MARS

```
mars_grid <- expand.grid(degree = 1:2,
                         nprune = 2:56)

set.seed(1)
mars.fit <- train(x, y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)

ggplot(mars.fit)

mars.fit$bestTune

summary(mars.fit$finalModel)

mars_table = mars.fit$finalModel$coefficients %>%
  as.data.frame()%>%
  select("Coefficient" = y) %>%
  knitr::kable()

plot_mars = function(name){
  p1 <- partial(mars.fit, pred.var = c(name), grid.resolution = 10) %>% autoplot()
  p1
}

cut_name = c("OverallQual","GrLivArea","X2ndFlrSF","YearBuilt","BsmtFinSF1",
             "LotArea","OverallCond","X1stFlrSF","TotalBsmtSF","SaleCondition")

plot_list = map(cut_name,plot_mars)

mars_plot = multiplot(plotlist = plot_list, cols=2)
```

Model comparison

```
resamp <- resamples(list(lm = lm.fit,
                         lasso = lasso.fit,
                         ridge = ridge.fit,
                         pcr = pcr.fit,
                         knn = knnFit,
                         GAM = gam.fit,
                         MARS = mars.fit))
```

```r
summary(resamp)

names(resamp)

str(resamp)

table_resample = resamp$values %>%
  as.data.frame() %>%
  janitor::clean_names() %>%
  select(Lm = lm_rmse, LASSO = lasso_rmse, Ridge = ridge_rmse,
         PCR = pcr_rmse, Knn = knn_rmse, GAM = gam_rmse, MARS = mars_rmse
         ) %>% broom::tidy() %>%
  select(-n,-trimmed,-mad,-skew,-kurtosis,-se)%>%
  knitr::kable()

bwplot(resamp, metric = "RMSE")
parallelplot(resamp, metric = "RMSE")
```

Exploration of final model

```r
im_var = varImp(mars.fit)
im_var$importance %>%
  head(10) %>%
  knitr::kable()

corrplot(cor(reg_data %>% select(-SalePrice) %>% select(cut_name)),title = "Correlation Plot", method =
```