homework4_3

syh

June 1, 2017

```r
# from HW4_2, we can get some features used to actually split a tree
# BsmtFinSF1   BsmtFinType1 BsmtFullBath BsmtUnfSF    CentralAir   Exterior1st
# Exterior2nd  Fireplaces   GarageArea   GrLivArea    KitchenQual  LotArea
# Neighborhood OverallCond  OverallQual  TotalBsmtSF  TotRmsAbvGrd WoodDeckSF
# X1stFlrSF    YearBuilt    YearRemodAdd
# Maybe we can use these features to build a linear regression model based on HW4
```

```r
# read all data (train + prediction) without missing value
real_all_daat <- read.csv(file = "H:/kaggle/houseprice/data/real_all_data_hybrid.csv",
                          stringsAsFactors = FALSE)[,-c(1,2)]
```

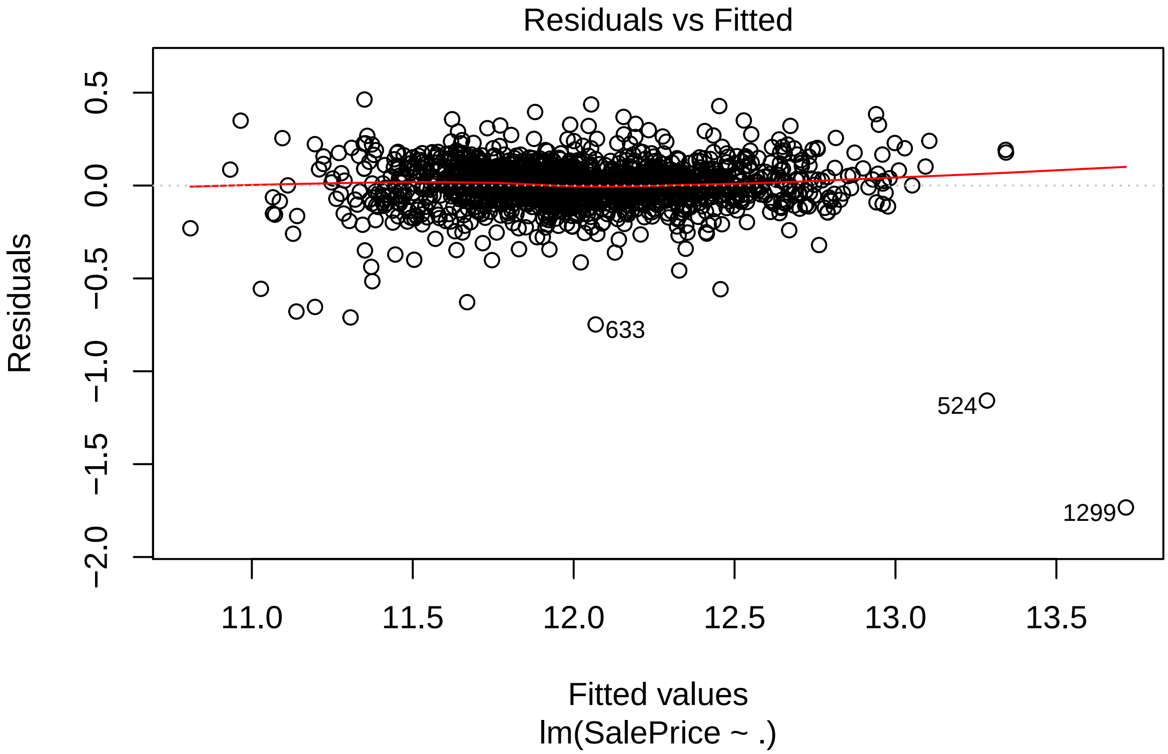
```r
# transform sale price to log sale price
real_all_daat[,"SalePrice"] <- log(real_all_daat[,"SalePrice"])
```

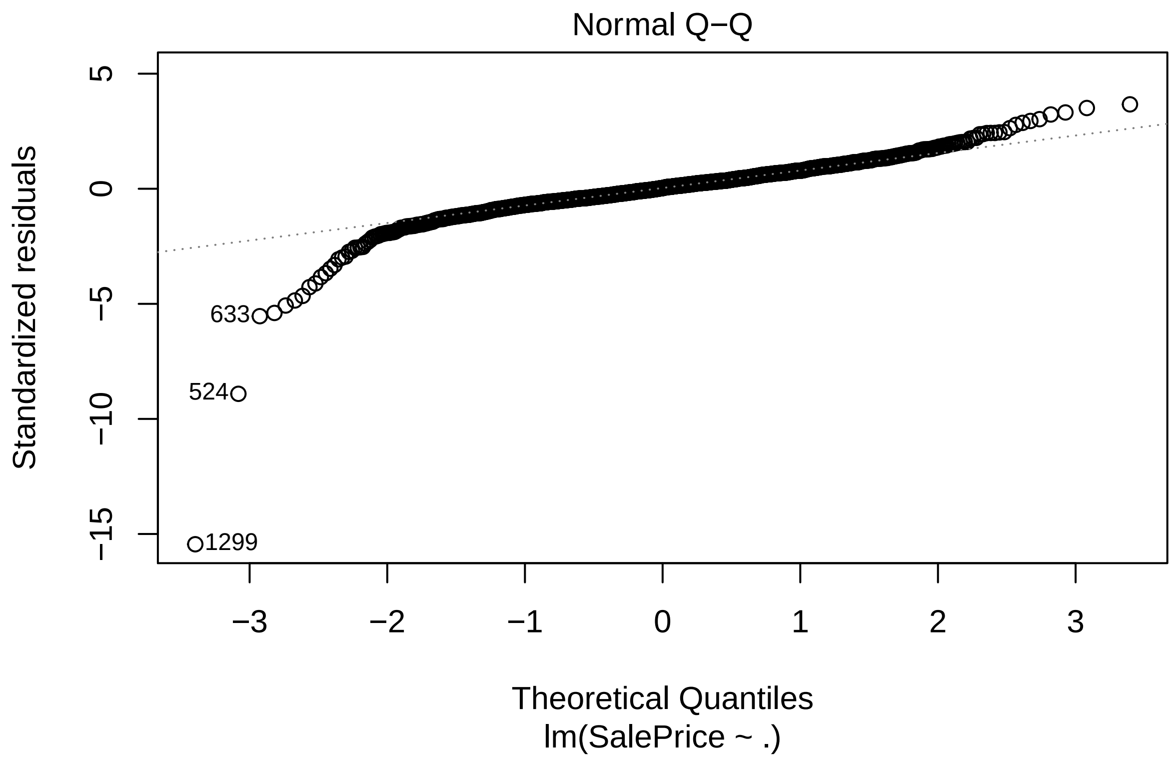
```r
# just keep features above
keep_col <- c("BsmtFinSF1", "BsmtFinType1", "BsmtFullBath", "BsmtUnfSF", "CentralAir",
              "Exterior1st", "Exterior2nd", "Fireplaces", "GarageArea", "GrLivArea",     "KitchenQual",
```

```r
real_all_daat <- real_all_daat[, keep_col]
```

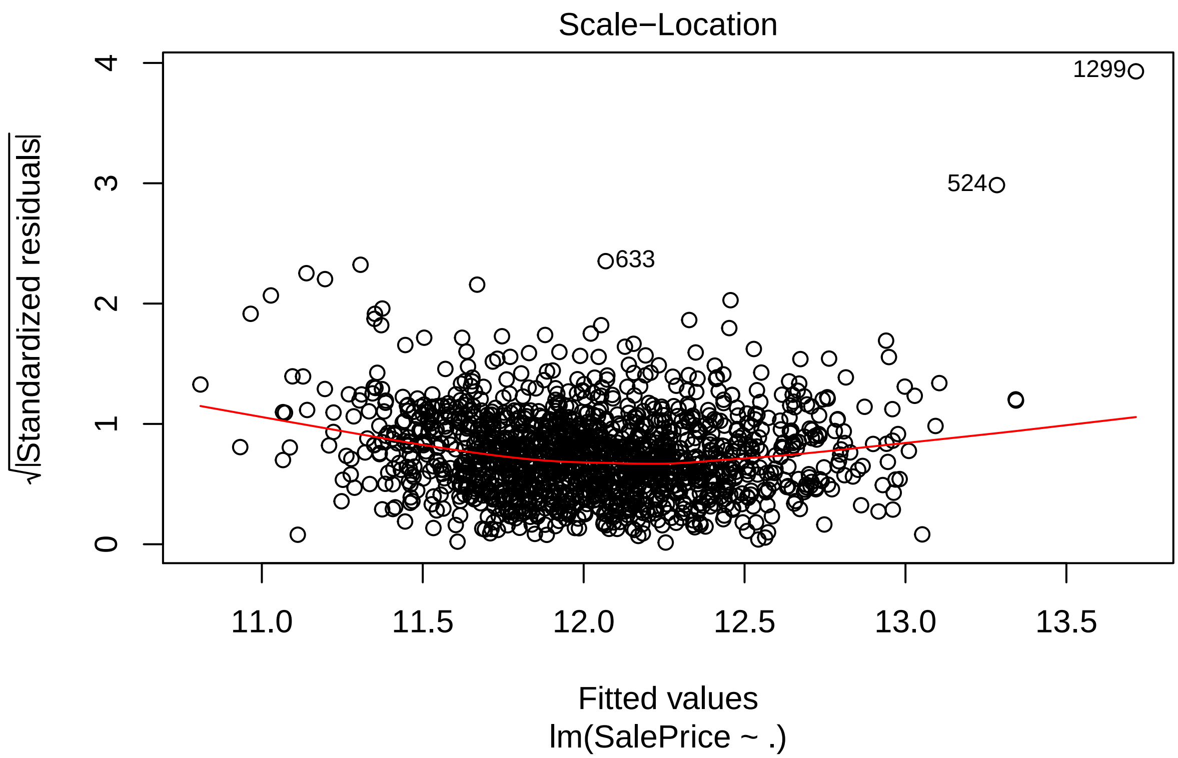
```r
# train a simple linear model by lm()
# check if it's subject to linear regression
# find some outlier and delete them
simple_lm <- lm(formula = SalePrice ~., data = real_all_daat[1:1460,])
plot(simple_lm)
```

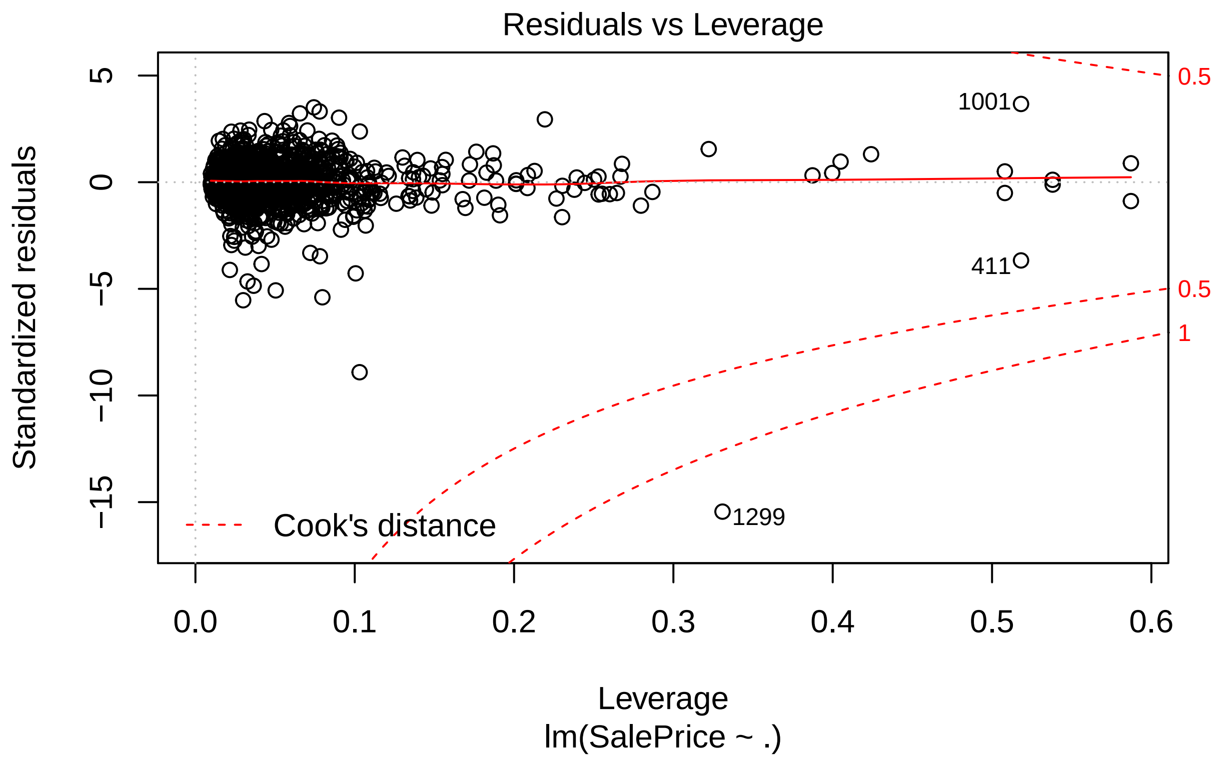
```
## Warning: not plotting observations with leverage one:
##   596, 1012, 1188, 1371
```


Residuals vs Fitted
Residuals
0.5
0.0
−0.5
−1.0
−1.5
−2.0
633
524
1299
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
lm(SalePrice ~ .)




```
## Warning: not plotting observations with leverage one:
##   596, 1012, 1188, 1371
```


Scale−Location
√|Standardized residuals|
0
1
2
3
4
11.0
11.5
12.0
12.5
13.0
13.5
1299
524
633
Fitted values
lm(SalePrice ~ .)




```
# from above, we know basically it can suject to linear regression
# there are some outliers: 524, 1299, 633
# get rid of them now after converting data
```

```
# we would like to train a linear regression model with regulation
# 1. convert categorical ones to factors
for(i in 1:dim(real_all_daat)[2]){
  if(is.character(real_all_daat[,i])){
    real_all_daat[,i] <- as.factor(real_all_daat[,i])
  }
}
```

```
# 2. convert x to matrix
all_data_x_matrix <- model.matrix(~.-SalePrice, data = real_all_daat)
```

```
# 1. split all data into train and prediction
model_x <- all_data_x_matrix[1:1460,]
model_y <- real_all_daat[1:1460,"SalePrice"]

pre_x <- all_data_x_matrix[-c(1:1460),]

# 2. get rid of outlier in train data
model_x <- model_x[-c(524, 1299, 633),]
model_y <- model_y[-c(524, 1299, 633)]

# 3. split model data into train and test
set.seed(1000)
```

```r
train_ind <- sample(1:dim(model_x)[1], size = dim(model_x)[1] * 0.7)
# train_ind <- read.csv(file = "H:/kaggle/houseprice/data/train_index.csv")
# train_ind <- train_ind[,1]

train_x <- model_x[train_ind,]
train_y <- model_y[train_ind]

test_x <- model_x[-train_ind,]
test_y <- model_y[-train_ind]
```

```r
# use cross validation to find optimal alpha in penality
library(glmnet)
```

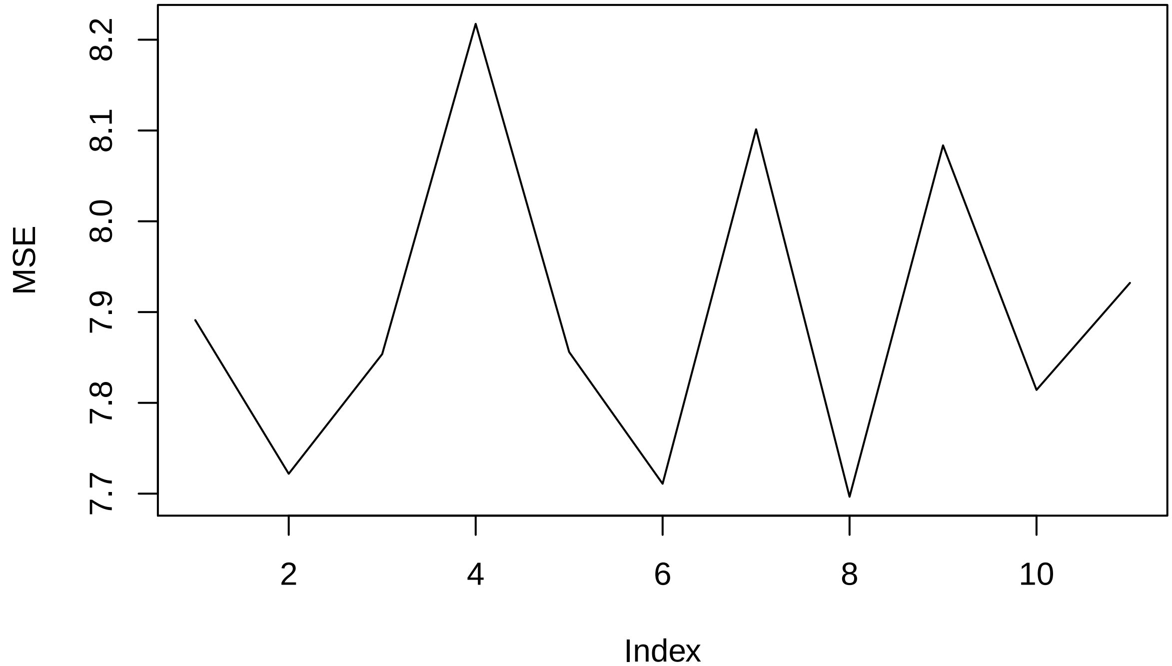
```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-10
```

```r
for(i in 0:10){
  assign(paste("fit", i, sep = ""),
         cv.glmnet(x = train_x, y = train_y, alpha = i/10,
            type.measure = "mse", family = "gaussian"))
}
```

```r
# find the optimal fit with least sse on test data
fit <- list(fit0, fit1, fit2, fit3, fit4, fit5, fit6, fit7, fit8, fit9, fit10)
mse <- NULL
for(i in 1:11){
  pre_y <- predict.cv.glmnet(object = fit[[i]], newx = test_x,s = fit[[i]]$lambda.1se)
  tmp_mse <- sum((pre_y - test_y) ^ 2)
  mse <- c(mse, tmp_mse)
}
plot(mse,type = 'l', ylab = "MSE")
```




```
# let's use this model to make prediction
opt_ind <- which.min(mse)
log_pre_y <- predict.cv.glmnet(object = fit[[opt_ind]],
                               newx = pre_x,s = fit[[opt_ind]]$lambda.1se)

# create submission
result <- data.frame(Id = c(1461:2919), SalePrice = exp(log_pre_y))
names(result) <- c("Id", "SalePrice")
write.csv(x = result, file = "H:/kaggle/houseprice/data/submission_5.csv",
          row.names = FALSE)
# Your submission scored 0.13634
# although score on kaggle doesn't improve, but from graph above,
# the range of MSE has decresed, I think this model is better.
```