homework4_5

syh

June 2, 2017

```r
# in this section, we would like to use xgboost to predict sale price
# attention: only matrix and factor are supported in xgboost
```

```r
# read all data (train + prediction) without missing value
real_all_data <- 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_data[,"SalePrice"] <- log(real_all_data[,"SalePrice"])
```

```r
# 1. convert categorical ones to factors
for(i in 1:dim(real_all_data)[2]){
  if(is.character(real_all_data[,i])){
    real_all_data[,i] <- as.factor(real_all_data[,i])
  }
}
```

```r
# 2. convert independent features into matrix
fea_mat <- model.matrix(~., data = subset(real_all_data, select = -SalePrice))
```

```r
# 1. split all data into train and prediction

model_x <- fea_mat[1:1460,]
model_y <- real_all_data[1:1460, "SalePrice"]

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

# 2. split model data into train and test
# seed <- sample.int(n = 10000, size = 1)
# set.seed(seed)
set.seed(2)
train_ind <- sample(1:dim(model_x)[1], size = dim(model_x)[1] * 0.7)

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

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

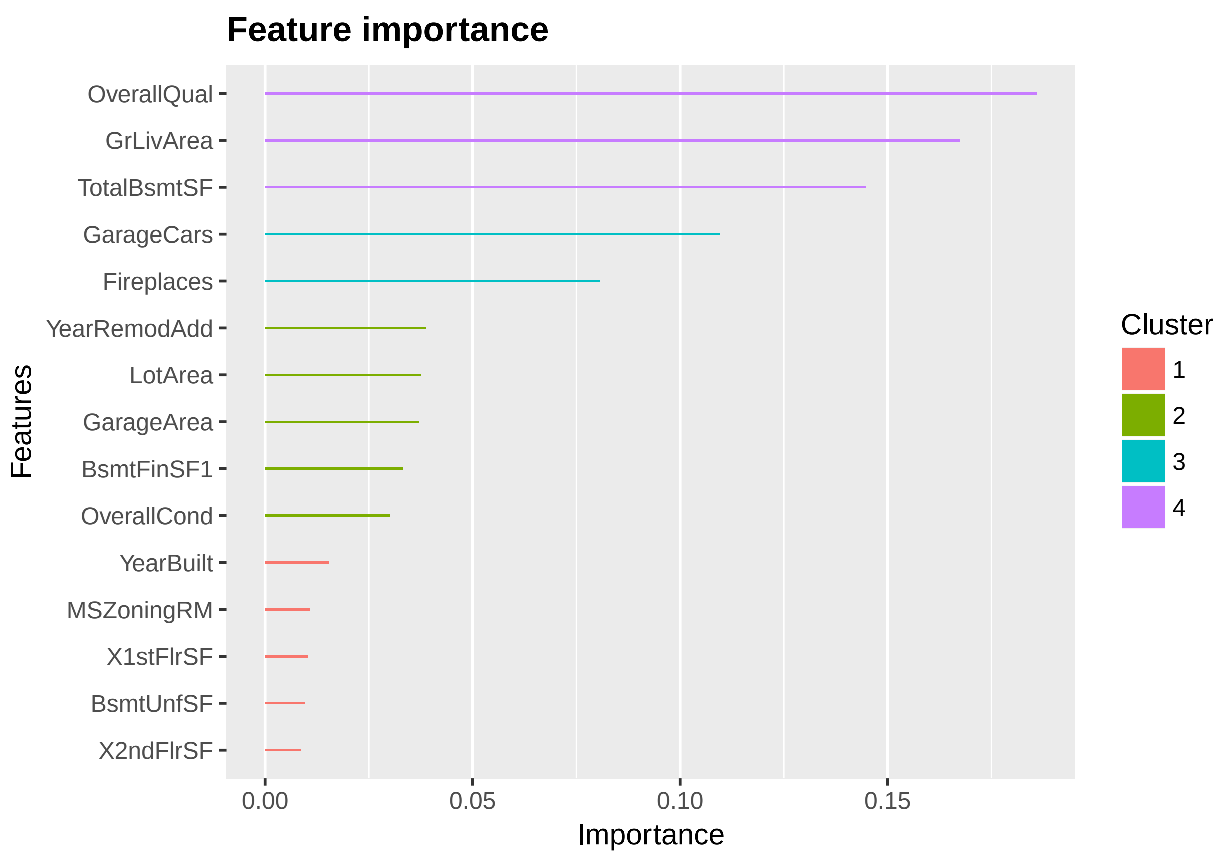
```r
# at first we use common parameter to train a simple xgboost model
library(xgboost)

set.seed(2)
xgb <- xgboost(data = train_x, label = train_y, nrounds = 20,
               verbose = 0, nthread = 3, max_depth = 8, object = "reg:linear"
               )
```

```r
# let's look at importance of each featues
importance <- xgb.importance(feature_names = colnames(train_x), model = xgb)
```

```r
importance
```

```
##                Feature         Gain        Cover   Frequency
##   1:         OverallQual 1.859170e-01 7.430550e-02 0.049655172
##   2:           GrLivArea 1.674253e-01 1.191554e-01 0.055172414
##   3:         TotalBsmtSF 1.448037e-01 5.378285e-02 0.033103448
##   4:          GarageCars 1.096569e-01 9.932702e-03 0.004137931
##   5:          Fireplaces 8.066602e-02 1.023818e-02 0.002758621
##  ---
## 136: Exterior1stMetalSd 1.109187e-05 1.666250e-04 0.002758621
## 137:          MSZoningRH 3.194426e-06 4.628473e-05 0.001379310
## 138: Exterior2ndImStucc 1.507100e-06 2.777084e-05 0.001379310
## 139:         LotShapeIR2 4.508379e-07 8.331251e-05 0.001379310
## 140:    HouseStyle2Story 3.136915e-07 3.702778e-05 0.001379310
```

```r
# visually look at first 15 most important features
library(ggplot2)
library(Ckmeans.1d.dp)
xgb.ggplot.importance(importance_matrix = importance, top_n = 15)
```




```r
# let's calculate the sse of this model on test data
est_y <- predict(object = xgb, newdata = test_x)
sqrt(mean((est_y - test_y) ^ 2))
```

```
## [1] 0.1556872
```

```r
# try to use caret, cross validation to find optimal combination of parameters.
# set possible combinations of parameters
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
paraGrid <- expand.grid(
            nrounds = c(200, 250, 300),
            max_depth = c(3, 5, 10),
            eta = c(0.05,0.1, 0.2),
            gamma = 0,  # first 0
            min_child_weight = c(10, 15, 20),
            subsample = c(0.6, 0.7, 0.8, 0.9),
            colsample_bytree = c(0.6, 0.7, 0.8)
            )
```

```r
# set train control
xgb_Control <- trainControl(method = "repeatedcv", verboseIter = F,
                            number = 5,repeats = 2,
                            returnData = FALSE, allowParallel = TRUE
                          )
```

```r
# trian our model using cross validation for grid search for parameters

gbm_train <- train(x = train_x, y = train_y, method = "xgbTree",
                   trControl = xgb_Control, tuneGrid = paraGrid)
```

```
## Loading required package: plyr
```

```r
rmse_order <- order(gbm_train$results$RMSE)
head(gbm_train$result[rmse_order,])
```

```
##      eta max_depth gamma colsample_bytree min_child_weight subsample
## 123 0.05         5     0              0.6               15       0.6
## 126 0.05         5     0              0.6               15       0.7
## 213 0.05         5     0              0.8               20       0.8
## 177 0.05         5     0              0.7               20       0.8
## 168 0.05         5     0              0.7               15       0.9
## 195 0.05         5     0              0.8               15       0.6
##     nrounds      RMSE  Rsquared     RMSESD RsquaredSD
## 123     300 0.1226106 0.9050017 0.01311464 0.01995468
## 126     300 0.1226407 0.9054051 0.01208968 0.01899921
## 213     300 0.1227658 0.9052387 0.01199691 0.01751308
## 177     300 0.1229347 0.9049472 0.01183679 0.01725657
## 168     300 0.1229729 0.9048490 0.01172866 0.01701256
## 195     300 0.1230281 0.9047249 0.01239297 0.01860893
```

```r
# for now best parameters
gbm_train$best
```

```
##     nrounds max_depth  eta gamma colsample_bytree min_child_weight
## 123     300         5 0.05     0              0.6               15
##     subsample
## 123       0.6
```

```r
# let's make a estimate on test data
est_y <- predict(object = gbm_train, newdata = test_x)
sqrt(mean((est_y - test_y) ^ 2))
```

```
## [1] 0.132788
```

```r
# make prediction
pre_sale_price <- predict(object = gbm_train, newdata = pre_x)
result <- data.frame(Id = c(1461:2919), SalePrice = exp(pre_sale_price))
write.csv(x = result, file = "H:/kaggle/houseprice/data/submission_7.csv",
          row.names = FALSE)
# the score on kaggle is about 0.1349, not better than before.
# I'm pretty sad now ...
```