Data Modeling

library(knitr)  
library(ggplot2)  
library(plyr)  
library(dplyr)  
library(corrplot)  
library(caret)  
library(gridExtra)  
library(scales)  
library(Rmisc)  
library(ggrepel)  
library(randomForest)  
library(psych)  
library(xgboost)

##Lasso Regresion Model  
set.seed(12345678)  
Control.Train <-trainControl(method = "cv", number = 5)  
Grid.Lasso <- expand.grid(alpha = 1, lambda = seq(0.001, 0.1, by = 0.0005))  
  
Model.Lasso <- train(x = train.set, y = df.combined$SalePrice[!is.na(df.combined$SalePrice)], method = 'glmnet', trControl= Control.Train, tuneGrid = Grid.Lasso)  
Model.Lasso$bestTune

## alpha lambda  
## 5 1 0.003

min(Model.Lasso$results$RMSE)

## [1] 0.1142001

Vars.Important.Lasso <- varImp(Model.Lasso, scale = F)  
Importance.Lasso <- Vars.Important.Lasso$importance  
  
Vars.Selected.Lasso <- length(which(Importance.Lasso$Overall!= 0))  
Vars.NotSelected.Lasso <- length(which(Importance.Lasso$Overall == 0))  
  
cat('Lasso Model used', Vars.Selected.Lasso, 'variables & did not use', Vars.NotSelected.Lasso)

## Lasso Model used 85 variables & did not use 97

Prediction.Lasso <- predict(Model.Lasso, test.set)  
Prediction.Values.Lasso <- exp(Prediction.Lasso)  
  
View(Prediction.Values.Lasso)  
summary(Prediction.Values.Lasso)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 47844 126908 158759 178107 212839 685017

##XGBoost Modeling  
Grid.XGB <- expand.grid(  
 nrounds = 1000,  
 eta = c(0.1, 0.05, 0.01),  
 max\_depth = c(2, 3, 4, 5, 6),  
 gamma = 0,  
 colsample\_bytree = 1,  
 min\_child\_weight = c(1, 2, 3, 4, 5),  
 subsample = 1  
)  
  
##Find Best Hyperparameter Values  
Caret.XGB <- train(x = train.set, y = df.combined$SalePrice[!is.na(df.combined$SalePrice)], method = 'xgbTree', trControl = Control.Train, tuneGrid = Grid.XGB)  
Caret.XGB$bestTune

## nrounds max\_depth eta gamma colsample\_bytree min\_child\_weight  
## 33 1000 3 0.05 0 1 3  
## subsample  
## 33 1

##XGB Boost Train & Test Sets  
Label.Train.XGB <- df.combined$SalePrice[!is.na(df.combined$SalePrice)]  
  
##XBG Train & Test Sets in Matrix  
Train.Matrix.XGB <- xgb.DMatrix(data = as.matrix(train.set), label = Label.Train.XGB)  
Test.Matrix.XGB <- xgb.DMatrix(data = as.matrix(test.set))  
  
##XGB Parameters  
Parameters.Model.XGB <- list(  
 objective = "reg:linear",  
 booster = "gbtree",  
 eta = 0.05,  
 gamma = 0,  
 max\_depth = 3,  
 min\_child\_weight = 3,  
 subsample = 1,  
 colsample\_bytree = 1  
)  
  
##XGB Cross Validation  
Cross.Validation.XGB <- xgb.cv(params = Parameters.Model.XGB, data = Train.Matrix.XGB, nrounds = 500, nfold = 5, showsd = T, stratified = T, print\_every\_n = 40, early\_stopping\_rounds = 10, maximize = F)

## [1] train-rmse:10.955587+0.004653 test-rmse:10.955522+0.019857   
## Multiple eval metrics are present. Will use test\_rmse for early stopping.  
## Will train until test\_rmse hasn't improved in 10 rounds.  
##   
## [41] train-rmse:1.428223+0.000292 test-rmse:1.428729+0.010585   
## [81] train-rmse:0.219372+0.000623 test-rmse:0.231261+0.004968   
## [121] train-rmse:0.101754+0.001338 test-rmse:0.130095+0.009549   
## [161] train-rmse:0.089498+0.001712 test-rmse:0.123474+0.010629   
## [201] train-rmse:0.083547+0.001728 test-rmse:0.121148+0.010802   
## [241] train-rmse:0.079052+0.001580 test-rmse:0.119955+0.011079   
## [281] train-rmse:0.075318+0.001398 test-rmse:0.119487+0.010939   
## [321] train-rmse:0.072064+0.001312 test-rmse:0.119232+0.011073   
## [361] train-rmse:0.069035+0.001125 test-rmse:0.118974+0.011057   
## Stopping. Best iteration:  
## [367] train-rmse:0.068597+0.001119 test-rmse:0.118900+0.011069

##Train Model using Best Round from Cross Validation  
Model.XGB <- xgb.train(data = Train.Matrix.XGB, params = Parameters.Model.XGB, nrounds = 367)  
  
Prediction.XGB <- predict(Model.XGB, Test.Matrix.XGB)  
Prediction.Values.XGB <- exp(Prediction.XGB)  
  
head(Prediction.Values.XGB)

## [1] 118315.0 161907.8 186297.7 187056.5 193928.6 166106.4

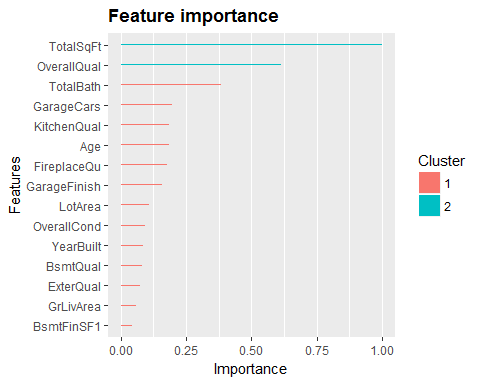
View(Prediction.Values.XGB)  
summary(Prediction.Values.XGB)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 42644 127825 156311 179084 209042 675123

##KMeans Clustering identifying Importance of Variables  
install.packages("Ckmeans.1d.dp")

## Error in install.packages : Updating loaded packages

library(Ckmeans.1d.dp)  
  
Importance.XGB <- xgb.importance(feature\_names = colnames(train.set), Model.XGB)  
xgb.ggplot.importance(importance\_matrix = Importance.XGB[1:15], rel\_to\_first = TRUE)



##Averaging Models  
Models.Average <- data.frame(Id = test.IDs, SalePrice = (Prediction.Values.XGB + Prediction.Values.Lasso)/2)  
  
head(Models.Average)

## Id SalePrice  
## 1461 1461 116905.4  
## 1462 1462 161620.0  
## 1463 1463 182991.7  
## 1464 1464 192431.8  
## 1465 1465 200390.8  
## 1466 1466 167897.1

View(Models.Average)  
summary(Models.Average$SalePrice)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46871 127287 156887 178595 209627 680070

##Export Results  
write.csv(Models.Average, file = 'Capstone Results.csv', row.names = F)