EXTRA - Work with cross validation

The below work has been done

1. Top Features have been selected
2. Some features has been transformed now (not modeled)
3. 10-fold cross validation has been implemented
4. The optimal model parameters are found
5. And data is model is ready

This CODE-INTENSIVE improvement part is READY but there is a scope of further development

In [ ]:
```
## Loading missForest package for missing value treatment
library(missForest)
library(ggplot2)

##READ
train_df <- read.csv('train.csv',sep=",")
test_df <- read.csv('test.csv',sep=",")
```

COMBINING TRAINING AND TESTING SAMPLE

In [2]:
```
fastmerge <- function(d1, d2) {
  d1.names <- names(d1)
  d2.names <- names(d2)

  # columns in d1 but not in d2
  d2.add <- setdiff(d1.names, d2.names)

  # columns in d2 but not in d1
  d1.add <- setdiff(d2.names, d1.names)

  # add blank columns to d2
  if(length(d2.add) > 0) {
    for(i in 1:length(d2.add)) {
      d2[d2.add[i]] <- NA
    }
  }

  # add blank columns to d1
  if(length(d1.add) > 0) {
    for(i in 1:length(d1.add)) {
      d1[d1.add[i]] <- NA
    }
  }

  return(rbind(d1, d2))
}
```

In [3]:
```
combined_df <- fastmerge(train_df,test_df)
```

MISSING VALUES

In [ ]:
```
#Removing >= 80% missing values
combined_df <- combined_df[,! colMeans(is.na(combined_df))>.8 ]

#Imputing the data set using pmm
data_imp <- missForest(combined_df, maxiter = 5,ntree = 50)

##combined_data imputation
combined_df <-  data_imp$ximp
```

In [9]:
```
#Imputation error
data_imp$OOBerror
```

NRMSE	0.219947612277948
PFC	0.0677595832512033

Feature Separation

In [5]:
```
#Features
cnames = colnames(combined_df)
features = cnames[!cnames %in% c('Id','SalePrice')]
length(features)

#Categorical NOMINAL Features
cat_features = c('MSSubClass', 'MSZoning', 'Street','Alley',
    'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
    'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
    'HouseStyle','OverallQual','OverallCond',
    'Foundation', 'BsmtFinType1','BsmtFinType2',
    'RoofStyle','RoofMatl','Exterior1st','Exterior2nd','MasVnrType',
    'Heating','CentralAir','Electrical','BsmtFullBath',
    'BsmtHalfBath','FullBath','HalfBath','BedroomAbvGr','KitchenAbvGr',
    'TotRmsAbvGrd','Functional','Fireplaces','GarageType',
    'GarageFinish','GarageCars','PavedDrive',
    'Fence','MiscFeature','SaleType','SaleCondition')
cat_features <- cat_features[cat_features %in% features]
length(cat_features)


#CATEGORICAL ORDINAL Features
ord_cat_features = c('ExterQual','ExterCond','BsmtQual','BsmtCond','BsmtExpo
                     'HeatingQC','KitchenQual','FireplaceQu','GarageQual','Ga
ord_cat_features <- ord_cat_features[ord_cat_features %in% features]
length(ord_cat_features)

#NUMERIC VALUED Features
num_features = c('LotFrontage','LotArea','MasVnrArea','BsmtFinSF1','BsmtFinS
    'BsmtUnfSF','TotalBsmtSF', 'LowQualFinSF','1stFlrSF','2ndFlrSF',
    'GrLivArea', 'YearRemodAdd', 'YearBuilt','GarageYrBlt','GarageArea','Woo
    'OpenPorchSF','EnclosedPorch','3SsnPorch','ScreenPorch','PoolArea','Misc
num_features <- num_features[num_features %in% features]
length(num_features)
```

75

41

10

21

Transforming functions

In [6]:
```
### Transforming neighborhood function based on 5 classes of SalePrice

transform_neighbor<- function(x){
    if(x %in% c("NoRidge", "NridgHt", "StoneBr") ){
        return (5) #Over 250k
    }
    if (x %in% c('CollgCr','Veenker','Crawfor','Somerst','Timber','ClearCr'
        return (4) #200-250
    }
    if (x %in% c('Mitchel','NWAmes','SawyerW','Gilbert','Blmngtn','SWISU',
        return (3) #150-200
    }
    if (x %in% c('OldTown','BrkSide','Sawyer','NAmes','IDOTRR','Edwards','B
        return (2) #100 - 150
    }
    if (x %in% c('MeadowV')){
        return (1)
    }

    return (9)
}

### Transforming some more ordinal values in different features
transform_others <- function(x){
    if( x %in% c('Ex')){
        return (5)
    }
    if( x %in% c('Gd')){
        return (4)
    }
    if( x %in% c('TA','Av')){
        return (3)
    }
    if( x %in% c('Fa','Mn')){
        return (2)
    }
    if( x %in% c('Po','No')){
        return (1)
    }
    return (0)
}
```

In [7]:
```
#1
combined_df['NbdClass'] <- sapply(combined_df$Neighborhood, transform_neighb
#2-#9
#c('ExterCond','ExterQual','BsmtQual','BsmtCond','BsmtExposure','HeatingQC',
combined_df['ExterQual'] <-sapply(combined_df$ExterQual, transform_others)
combined_df['ExterCond'] <-sapply(combined_df$ExterCond, transform_others)
combined_df['BsmtQual'] <-sapply(combined_df$BsmtQual, transform_others)
combined_df['BsmtCond'] <-sapply(combined_df$BsmtCond, transform_others)
combined_df['BsmtExposure'] <-sapply(combined_df$BsmtExposure, transform_oth
combined_df['HeatingQC'] <-sapply(combined_df$HeatingQC, transform_others)
combined_df['KitchenQual'] <-sapply(combined_df$KitchenQual, transform_other
combined_df['FireplaceQu'] <-sapply(combined_df$FireplaceQu, transform_other
```

SPLITTING BACK train and test

In [8]:
```
trainX <- combined_df[1:nrow(train_df),]
testX <- combined_df[(nrow(train_df)+1):(nrow(train_df)+nrow(test_df)),]
```

In [9]:
```
use_features = c('OverallQual','YearBuilt','YearRemodAdd','MasVnrArea','GrL
           'GarageYrBlt','GarageCars','NbdClass','HouseStyle','Alley','LotSh
           'Exterior2nd','ExterQual','ExterCond','Foundation','BsmtExposure
           'FireplaceQu','GarageType','GarageFinish')
use_features <- use_features[use_features %in% features]
```

RANDOM FORESTS USING CROSS VALIDATION

In [10]:
```
ntrees = seq(100,1001,100)
maxnodes_ = seq(12,100,12)

result_data <- data.frame()
ntree_col = c()
maxnodes_col = c()
scores_ = c()
colnames=c('ntree','maxnodes','scores')
```

In [13]:
```
X = trainX[c(use_features,'SalePrice')]
Y = trainX['SalePrice']
```

In [15]:
```
#RANDOM FOREST WITH 10-FOLD CROSS VALIDATION

for(n in ntrees){
    cat("ntrees:",n,"\n")
    for( m in maxnodes_){
        run_tot = 0
        #10-fold train_test Cross Validation
        for( k in seq(0,9)){

            testXcv = X[(k*146+1):(k*146+146),]
            if(k!=0){
                trainX_left = X[(1):(k*146),]
            } else {
                trainX_left = X[(0):(k*146),]
            }
            if(k!=9){
                trainX_right= X[(((k+1)*146)+1):nrow(X),]
            } else {
                trainX_right= X[0:0,]
            }
            trainXcv = rbind(trainX_left, trainX_right)

            testYcv = Y[(k*146+1):(k*146+146),]
            ###
            #trainY_left = Y[(1):(k*146),]
            #trainY_right= Y[((k+1)*146+1):nrow(Y),]
            #trainYcv = rbind(trainY_left, trainY_right)
            ###

            #Fitting model and predicting the kth fold set
            ###rf <- randomForest(x = trainXcv, y = trainYcv['SalePrice'],nt
            cat(dim(trainXcv),"\n")
            cat(dim(trainX_left),"\n")
            cat(dim(trainX_right),"\n")
            rf <- randomForest(SalePrice~.,trainXcv,do.trace=10, ntrees = n
            preds <- predict(rf,testXcv)

            run_tot <- run_tot + sqrt(sum((preds-testYcv)**2) )
        }
        ntree_col = c(ntree_col,n)
        maxnodes_col = c(maxnodes_col,m)
        scores_ = c(scores_, run_tot/10.0)
    }
}

result_data <- data.frame(ntree_col,maxnodes_col,scores_)
```

```
ntrees: 100
1314 26
0 26
1314 26
     |      Out-of-bag   |
Tree |      MSE  %Var(y) |
  10 | 1.72e+09    26.52 |
  20 | 1.594e+09    24.57 |
  30 | 1.496e+09    23.07 |
  40 | 1.444e+09    22.26 |
  50 | 1.439e+09    22.18 |
  60 | 1.411e+09    21.75 |
  70 | 1.384e+09    21.34 |
  80 | 1.386e+09    21.37 |
  90 | 1.387e+09    21.37 |
 100 | 1.376e+09    21.20 |
 110 | 1.379e+09    21.25 |
 120 | 1.369e+09    21.11 |
 130 | 1.363e+09    21.01 |
 140 | 1.37e+09    21.11 |
```

In [16]:
```
min_score <- min(result_data$scores_)
optimal <- result_data[result_data['scores_']==min_score]
```

In [20]:
```
rf <-randomForest(SalePrice~., trainX,do.trace=10, ntrees = optimal[1],maxn
```

```
     |      Out-of-bag   |
Tree |      MSE  %Var(y) |
  10 | 1.169e+09    18.53 |
  20 | 1.093e+09    17.33 |
  30 | 9.273e+08    14.70 |
  40 | 8.71e+08    13.81 |
  50 | 8.401e+08    13.32 |
  60 | 8.376e+08    13.28 |
  70 | 8.287e+08    13.14 |
  80 | 8.178e+08    12.97 |
  90 | 8.345e+08    13.23 |
 100 | 8.289e+08    13.14 |
 110 | 8.255e+08    13.09 |
 120 | 8.203e+08    13.01 |
 130 | 8.092e+08    12.83 |
 140 | 8.032e+08    12.73 |
 150 | 8.015e+08    12.71 |
 160 | 8.026e+08    12.73 |
 170 | 8.044e+08    12.75 |
 180 | 8.029e+08    12.73 |
```

TAKES QUITE AN AMOUNT OF TIME TO TRAIN

In [ ]:
```
#testX <- combined_df[(nrow(train_df)+1):(nrow(train_df)+nrow(test_df)+1),]
```

In [21]:
```
predicted <- predict(rf, testX)
```

In [26]:
```
id <- seq(1461,2919)
```

In [27]:
```
test_result <- data.frame(id,predicted)
```

In [1]:
```
write.csv(test_result,file = "res2.csv",row.names=FALSE, col.names=FALSE)
```

FURTHER WORK CAN BE DONE BY

1. Transforming all the categorical variables
2. Creating new features
3. Ensembling some more models

----------The End----------

In [ ]: