getwd()

setwd("D://House\_Price")

Train\_HP <- read.csv("D://House\_Price/train.csv")

Test\_Hp <- read.csv("D://House\_Price/test.csv")

table(dim(Train\_HP))

table(dim(Test\_Hp))

summary(Train\_HP)

str(Train\_HP)

#------Check skewness of Target variable--------

plot(density(Train\_HP$SalePrice),col='Red')

apply(Train\_HP,2,function(x) length(unique(x))) #Check unique lenth of each column

Train\_HP\_Sale <- Train\_HP$SalePrice #Save target variable to Train\_HP\_Sale as new variable

Train\_HP$SalePrice <- NULL # Drop Target column from Train

table(dim(Train\_HP))

Combinedf <- merge(Train\_HP,Test\_Hp,all = TRUE) #Merge Train & Test

table(dim(Combinedf))

str(Combinedf)

table(is.na(Combinedf))

#View(Combinedf)

#---------Below variables deleted becuase they contains more than 70% NA Values------

table(is.na(Combinedf$Alley))

table(is.na(Combinedf$PoolQC))

table(is.na(Combinedf$Fence))

table(is.na(Combinedf$MiscFeature))

table(is.na(Combinedf$MiscVal))

Combinedf$Alley <- NULL

Combinedf$PoolQC <- NULL

Combinedf$Fence <- NULL

Combinedf$MiscFeature <- NULL

Combinedf$MiscVal <- NULL

colSums(is.na(Combinedf))

Combinedf$FireplaceQu <- NULL

# --------------Saperate numeric and categorical columns--------------

cat\_vars <- Combinedf[,sapply(Combinedf,is.factor)]

str(cat\_vars)

num\_vars <- Combinedf[,!sapply(Combinedf,is.factor)]

str(num\_vars)

length(num\_vars)

length(cat\_vars)

#------------Find missing values for numeric variables-----------

apply(num\_vars,2,function(x)table(is.na(x))) # Check missing values columnwise

plot(density(Combinedf$LotFrontage,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$MasVnrArea,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$BsmtFinSF1,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$BsmtFinSF2,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$BsmtUnfSF,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$TotalBsmtSF,na.rm = T),col="Red") #Skewed #--Check skewness of each numeric variable---

plot(density(Combinedf$BsmtFullBath,na.rm = T),col="Red") #Normal

plot(density(Combinedf$BsmtHalfBath,na.rm = T),col="Red") #Normal

plot(density(Combinedf$GarageYrBlt,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$GarageCars,na.rm = T),col="Red") #Normal

plot(density(Combinedf$GarageArea,na.rm = T),col="Red") #Skewed

plot(density(Combinedf$BsmtFullBath,na.rm = T),col="Red") #Normal

#-------Filling Skewed variable missing values with median-----------

Combinedf$LotFrontage[is.na(Combinedf$LotFrontage)] <- median(Combinedf$LotFrontage,na.rm = T)

Combinedf$MasVnrArea[is.na(Combinedf$MasVnrArea)] <- median(Combinedf$MasVnrArea,na.rm = T)

Combinedf$BsmtFinSF1[is.na(Combinedf$BsmtFinSF1)] <- median(Combinedf$BsmtFinSF1,na.rm = T)

Combinedf$BsmtFinSF2[is.na(Combinedf$BsmtFinSF2)] <- median(Combinedf$BsmtFinSF2,na.rm = T)

Combinedf$BsmtUnfSF[is.na(Combinedf$BsmtUnfSF)] <- median(Combinedf$BsmtUnfSF,na.rm = T)

Combinedf$TotalBsmtSF[is.na(Combinedf$TotalBsmtSF)] <- median(Combinedf$TotalBsmtSF,na.rm = T)

Combinedf$GarageYrBlt[is.na(Combinedf$GarageYrBlt)] <- median(Combinedf$GarageYrBlt,na.rm = T)

Combinedf$GarageArea[is.na(Combinedf$GarageArea)] <- median(Combinedf$GarageArea,na.rm = T)

Combinedf$BsmtFullBath[is.na(Combinedf$BsmtFullBath)] <- median(Combinedf$BsmtFullBath,na.rm = T)

#-------Filling Normal variable missing values with mean-------

Combinedf$BsmtHalfBath[is.na(Combinedf$BsmtHalfBath)] <- mean(Combinedf$BsmtHalfBath,na.rm = T)

Combinedf$GarageCars[is.na(Combinedf$GarageCars)] <- mean(Combinedf$GarageCars,na.rm = T)

table(is.na(Combinedf)) #Check a values after missing value imputation for numeric variables

#----- Missing value imputation for categoriacl data-------

#----Here Combinedf(i)means categorical variable in Combinedf---

for(i in colnames(cat\_vars))

{

for(j in colnames(Combinedf))

{

if(i==j)

{

val <- unique(Combinedf[i][!is.na(Combinedf[i])]);val

mode <- val[which.max(tabulate(match(Combinedf[i], val)))];mode #Mode of Combinedf(i)

Combinedf[i][is.na(Combinedf[i])] <- mode

}

}

}

table(is.na(Combinedf)) # Final check for missing values

#----Create dummy variables for categorical variables in Combinedf--------

#install.packages("caret",repos = "http://cran.r-project.org")

library(caret)

dmy <- dummyVars(" ~ .", data = Combinedf)

trsf <- data.frame(predict(dmy, newdata = Combinedf))

print(trsf)

str(trsf)

Combinedf <- data.frame(Combinedf,trsf) #Adding dummy variables in exising Combinedf data

str(Combinedf)

apply(trsf,2,function(x)length(unique(x))) # Check unique length after adding dummy variables

str(Combinedf)

#---------------Working for outliers-----------------

par(mfrow=c(1,4))

boxplot(Combinedf$LotArea.1);boxplot(Combinedf$LotFrontage.1);boxplot(Combinedf$MSSubClass.1);boxplot(Combinedf$OpenPorchSF)

par(mfrow=c(1,4))

boxplot(Combinedf$GarageArea);boxplot(Combinedf$GrLivArea);boxplot(Combinedf$X2ndFlrSF);boxplot(Combinedf$X1stFlrSF)

par(mfrow=c(1,4))

boxplot(Combinedf$TotalBsmtSF);boxplot(Combinedf$BsmtUnfSF);boxplot(Combinedf$BsmtFinSF1);boxplot(Combinedf$MasVnrArea)

par(mfrow=c(1,2))

boxplot(Combinedf$LotArea);boxplot(Combinedf$MSSubClass)

#----------Create function to capping the outliers with respective quantiles--------------

capOutlier <- function(x){

qnt <- quantile(x, probs=c(.25, .75), na.rm = T)

caps <- quantile(x, probs=c(.05, .98), na.rm = T)

H <- 1.5 \* IQR(x, na.rm = T)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

return(x)

}

#------Imputing outliers with capOutlier() function-----

Combinedf$LotArea.1 <- capOutlier(Combinedf$LotArea.1)

Combinedf$LotFrontage.1 <- capOutlier(Combinedf$LotFrontage.1)

Combinedf$MSSubClass.1 <- capOutlier(Combinedf$MSSubClass.1)

Combinedf$OpenPorchSF <- capOutlier(Combinedf$OpenPorchSF)

Combinedf$GarageArea <- capOutlier(Combinedf$GarageArea)

Combinedf$GrLivArea <- capOutlier(Combinedf$GrLivArea)

Combinedf$X2ndFlrSF <- capOutlier(Combinedf$X2ndFlrSF)

Combinedf$X1stFlrSF <- capOutlier(Combinedf$X1stFlrSF)

Combinedf$TotalBsmtSF <- capOutlier(Combinedf$TotalBsmtSF)

Combinedf$BsmtUnfSF <- capOutlier(Combinedf$BsmtUnfSF)

Combinedf$BsmtFinSF1 <- capOutlier(Combinedf$BsmtFinSF1)

Combinedf$MasVnrArea <- capOutlier(Combinedf$MasVnrArea)

Combinedf$LotArea <- capOutlier(Combinedf$LotArea)

Combinedf$MSSubClass<- capOutlier(Combinedf$MSSubClass)

#-----------------------------

#----------Split Dataset in to Train and Test--------

table(dim(Combinedf))

Train\_House <- Combinedf[1:1460,]

table(dim(Train\_House))

Test\_House <- Combinedf[1461:2919,]

table(dim(Test\_House))

Train\_House$SalePrice

#-------------- #Add original Target variable as Target variable in Train\_House.------

Train\_House$SalePrice <- Train\_HP\_Sale

table(dim(Train\_House))

#------------Remove outliers from Target vaiable-----

boxplot(Train\_House$SalePrice)

Train\_House$SalePrice <- capOutlier(Train\_House$SalePrice)

boxplot(Train\_House$SalePrice)

#------------------

#rm(list=ls()) #Clean Global Environment

#----Linear model-----

train\_linear1 <- lm(SalePrice~.,data = Train\_House)

summary(train\_linear1)

#library(car)

#vif(train\_linear1)

train\_linear\_new <- lm(Train\_House$SalePrice~Train\_House$MSZoning.RH + Train\_House$LotFrontage + Train\_House$LotArea + Train\_House$Street.Pave + Train\_House$LandSlope.Mod +

Train\_House$Neighborhood.Crawfor + Train\_House$Neighborhood.Edwards + Train\_House$Neighborhood.MeadowV + Train\_House$Neighborhood.StoneBr + Train\_House$Utilities.NoSeWa +

Train\_House$Neighborhood.Mitchel + Train\_House$Neighborhood.StoneBr + Train\_House$Condition1.Feedr + Train\_House$Condition1.Norm + Train\_House$Condition1.PosN +

Train\_House$Condition1.RRAn + Train\_House$Condition1.RRNn + Train\_House$Condition2.PosA + Train\_House$Condition2.PosN + Train\_House$OverallQual + Train\_House$OverallCond +

Train\_House$YearBuilt + Train\_House$YearRemodAdd + Train\_House$Exterior1st.HdBoard + Train\_House$MasVnrType.Stone + Train\_House$Foundation.PConc + Train\_House$Foundation.Wood + Train\_House$BsmtCond.Po +

Train\_House$BsmtExposure.Gd + Train\_House$BsmtFinSF1 + Train\_House$TotalBsmtSF + Train\_House$Heating.GasA + Train\_House$Heating.OthW + Train\_House$Heating.Wall + Train\_House$HeatingQC.Gd + Train\_House$HeatingQC.TA +

Train\_House$CentralAir.Y + Train\_House$Electrical.Mix + Train\_House$LowQualFinSF + Train\_House$GrLivArea + Train\_House$BsmtFullBath + Train\_House$FullBath + Train\_House$KitchenAbvGr + Train\_House$KitchenQual.Gd +

Train\_House$KitchenQual.TA + Train\_House$Functional.Typ + Train\_House$Fireplaces +

Train\_House$GarageType.Attchd+Train\_House$GarageType.Basment, data = Train\_House)

summary(train\_linear\_new)

vif(train\_linear\_new)

attach(Train\_House)

train\_linear\_Final <- lm(SalePrice~LotArea+LandSlope.Mod+Neighborhood.Crawfor+Neighborhood.Edwards+Neighborhood.MeadowV+

Neighborhood.StoneBr+Neighborhood.Mitchel+Condition1.Norm+

OverallQual+OverallCond+YearBuilt+YearRemodAdd+

Exterior1st.HdBoard+Foundation.PConc+BsmtCond.Po+BsmtFinSF1+TotalBsmtSF+

HeatingQC.TA+Electrical.Mix+GrLivArea+BsmtFullBath+KitchenAbvGr+

KitchenQual.Gd+KitchenQual.TA+Functional.Typ+Fireplaces, data = Train\_House)

new\_test <- data.frame(LotArea,LandSlope.Mod,Neighborhood.Crawfor,Neighborhood.Edwards,Neighborhood.MeadowV,

Neighborhood.StoneBr,Neighborhood.Mitchel,Condition1.Norm,

OverallQual,OverallCond,YearBuilt,YearRemodAdd,

Exterior1st.HdBoard,Foundation.PConc,BsmtCond.Po,BsmtFinSF1,TotalBsmtSF,

HeatingQC.TA,Electrical.Mix,GrLivArea,BsmtFullBath,KitchenAbvGr,

KitchenQual.Gd,KitchenQual.TA,Functional.Typ,Fireplaces)

summary(train\_linear\_Final)

#plot(train\_linear\_Final)

fittd\_val <- fitted(train\_linear\_Final)

#residuals\_val <- residuals(train\_linear\_Final)

#plot(fittd\_val,residuals\_val,type = 'o')

library(MASS)

stepAIC(train\_linear\_Final,direction = "both")

Final\_fit <- lm(SalePrice ~ LotArea + LandSlope.Mod + Neighborhood.Crawfor +

Neighborhood.Edwards + Neighborhood.StoneBr + Neighborhood.Mitchel +

Condition1.Norm + OverallQual + OverallCond + YearBuilt +

YearRemodAdd + Exterior1st.HdBoard + Foundation.PConc + BsmtFinSF1 +

TotalBsmtSF + HeatingQC.TA + GrLivArea + KitchenAbvGr + KitchenQual.Gd +

KitchenQual.TA + Functional.Typ + Fireplaces,data=Train\_House)

test\_linear <- predict(Final\_fit,newdata = new\_test)

test\_linear

#plot(test\_linear,type = "l",lty=1.8,col="blue")

#---Model validation part-------------------------------------------------------------------------

#----1st method---RMSE()

rmse <- sqrt(mean(test\_linear-Train\_House$SalePrice)^2)

rmse

#----2nd method---Min\_Max Accuray()

Min\_Max\_Accuracy <- mean(min(Train\_House$SalePrice, test\_linear)/max(Train\_House$SalePrice, test\_linear))

#---------------------------------------------------------------------------------------------------

#Submit <- data.frame("Id"=Test\_House$Id,"SalePrice"=test\_linear)

New <- write.csv(test\_linear,"D://House\_Price/New\_pred.csv")

head(test\_linear)

head(Train\_House$SalePrice)

#-------Decision Tree--------

library(rpart)

library(rpart.plot)

fit\_dt <- rpart(Train\_House$SalePrice~.,data = Train\_House,method = "anova")

prp(fit\_dt)

#----Plot the model---

plot(fit\_dt, uniform=TRUE,main="Regression Tree")

text(fit\_dt, use.n=TRUE, cex = .6)

printcp(fit\_dt)

plotcp(fit\_dt)

pred\_dt <- predict(fit\_dt,newdata = Test\_House,method="anova")

pred\_dt

#-------Check model accuracy-----

#a <- table(actual=Train\_House$SalePrice,prediction=pred\_dt);a

#sum(diag(a))/sum(a)#Check model accuracy

#library(caret)

#p1 <- predict(fit\_dt,Train\_House)

#confusionMatrix(p1,Train\_House$SalePrice)

#length(p1)

#length(Train\_House$SalePrice)

write.csv(pred\_dt,"D://House\_Price/Decision\_Tree.csv")

#--------Random forest model--------

#attach(Train\_House)

library(randomForest)

set.seed(123)

#rand\_model <- randomForest(SalePrice~.,data = Train\_House)

rand\_model <- randomForest(SalePrice~.,data = Train\_House,ntree=500,mtry=20,importance=T,proximity=T)

rand\_model

print(rand\_model)

importance(rand\_model)

varImpPlot(rand\_model)

varImp(rand\_model)

getTree(rand\_model,1)

pred\_rf <- predict(rand\_model,newdata = Test\_House)

head(pred\_rf)

attributes(rand\_model)

RMSE(pred\_rf,SalePrice)

write.csv(pred\_rf,"D://House\_Price/Random\_Forest\_New\_test2.csv")

#-------Check randomForest model accuracy using lm()----------

#totaltest <- cbind(Test\_House,pred\_rf)

#reg <- lm(SalePrice~pred\_rf,data = totaltest)

#summary(reg)

#------------------