University of Essex

DEPARTMENT OF MATHEMATICAL SCIENCES

Group Project Assignment

Submitted as part of the requirements for:

MA321 APPLIED STATISTICS

Supervisor: DR. STELLA HADJIANTONI

DR. FANLIN MENG

Submitted by:

DANISHA.A.RAJOO -1907654

DIVYA ARORA -1901423

ROHINI SUBRAMANIAM -1908736

ROSLI, NUR F F B -1908520

SRINIDHI KARTHIKEYAN -1900637

Date of submission 27/03/2020

Word count:2400

Abstract: *This group coursework is designed to investigate the data regarding residential properties collected from 2006 to 2010 by the US Census Service.*

Contents

1 Introduction

This report is focused on many different variables and their effect on overall house condition and sale price. We have simulated the missing values that the data set contained using the mice method and analysed them using various methods to develop a logistic regression model, to model the house Condition. Using Linear Discriminant analysis we have been able to make a more precise model and from our analysis we have been able to predict house prices for the data set.

2 Preliminary Analysis

The project was conducted on residential property data set collected by US census Service during 2006 to 2010. There are 51 variables in total and 2919 values in each column. There are 23 variables with data type integer and 28 variables with datatype as Factor.

2.1 Missing Values

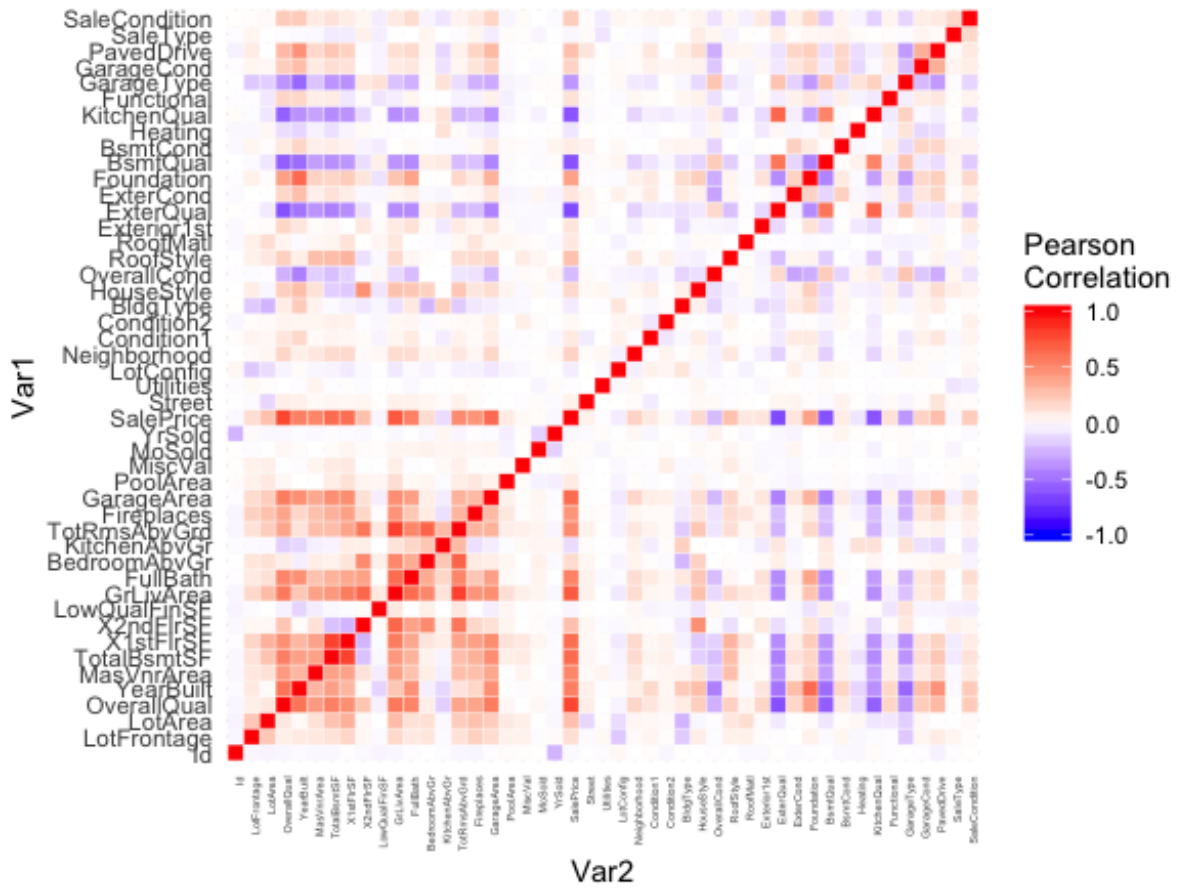
There are 11 variables with missing values. According to description table 'NA' in "Alley" represents 'No alley access', in "poolQC" variable 'NA' represents "No pool", similarly in "Fence" and "MiscFeatures" 'NA' represents "No fence" and "None" respectively, hence those variables can be removed as 'NA' represents absence of those features unlike reading error.

2.2 Imputation:

After removing the features with more than 60 percent of missing values the remaining features were imputed. Multivariate Imputation by Chained Equations(MICE) : we have assumed that the data is missing completely at random. Mice with predictive mean matching method is used for imputation as the feature "Sales price" contains 49% missing values. Imputing using the mean and median would have lead to a biased data set.

2.3 Feature Selection:

A Correlation map is constructed to find the associations between variables in order to avoid Multi-colinearity which may lead to skewed results. It was observed that variables "1stFlrSF" is highly correlated with" TotalBsmtSF" and "GrLivArea" is highly correlated with "TotRmsAbvGrd" . Hence "1stFlrSF" and "GrLivArea" are removed .

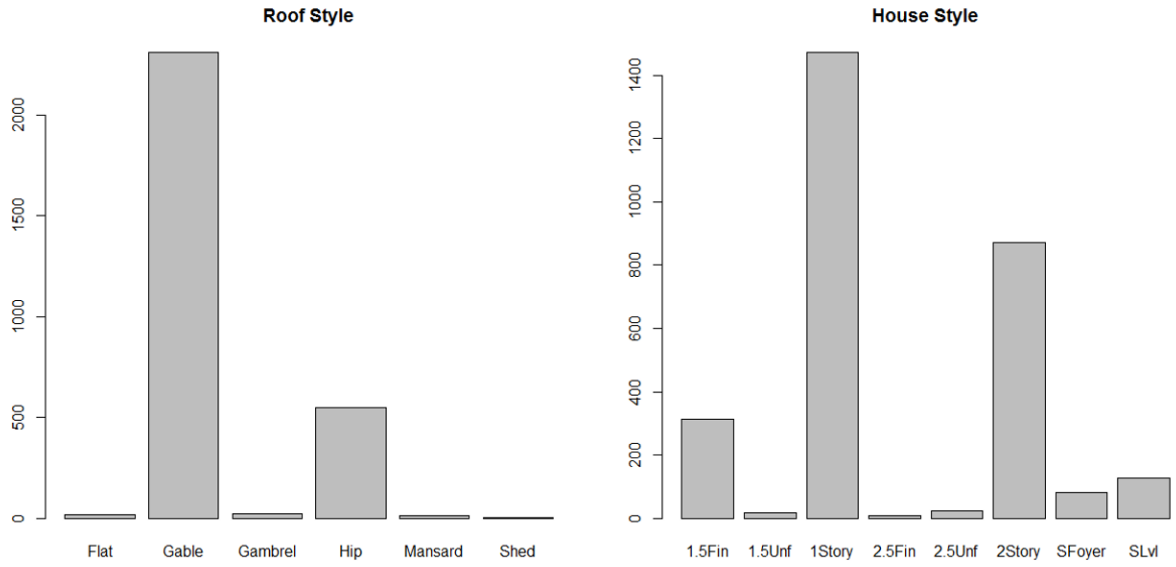


3 Analysis

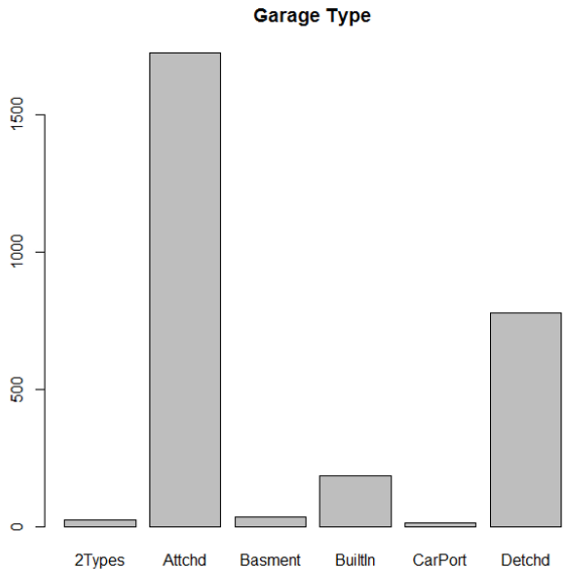
3.1 Numerical and Graphical Summaries

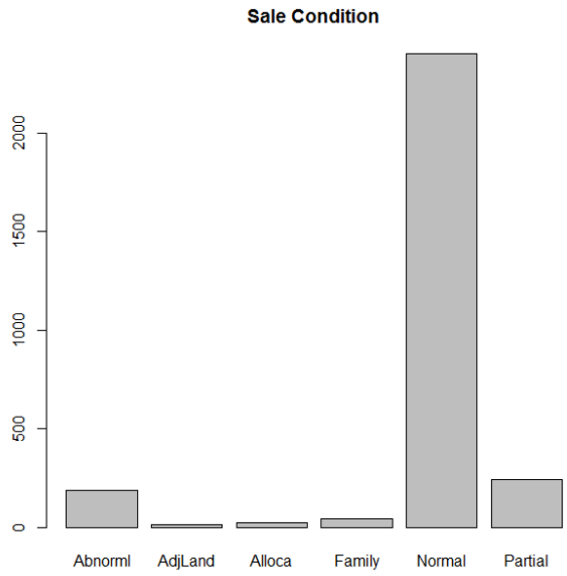
The house data consists of all the factors that can influence the sale price of house for example: garage area, basement area, number of bedrooms, number of kitchens, number of bathrooms, present condition of the house etc. The purpose of the exploratory data analysis is to find

the missing values in the data and explore the overall view and patterns among data. PoolQc, MiscFeature, Alley, fence and Sale price columns contains most of the missing values in the data set. All other variables contain no missing data. The remaining variables contain almost complete values.

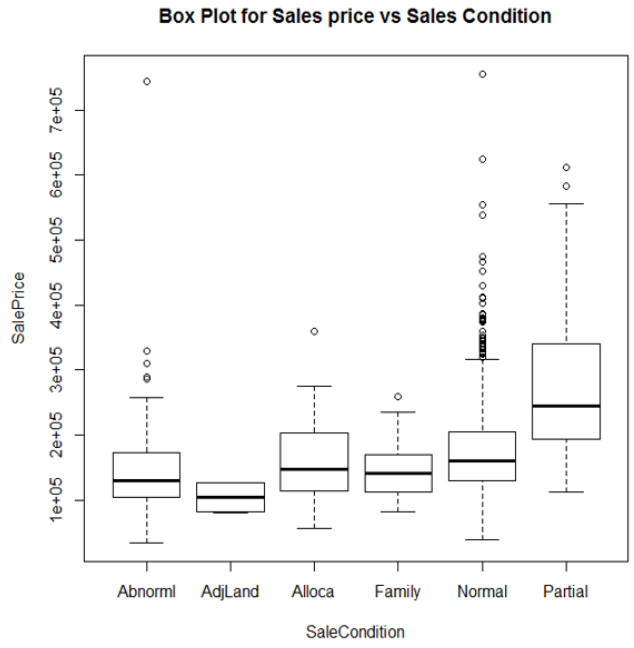


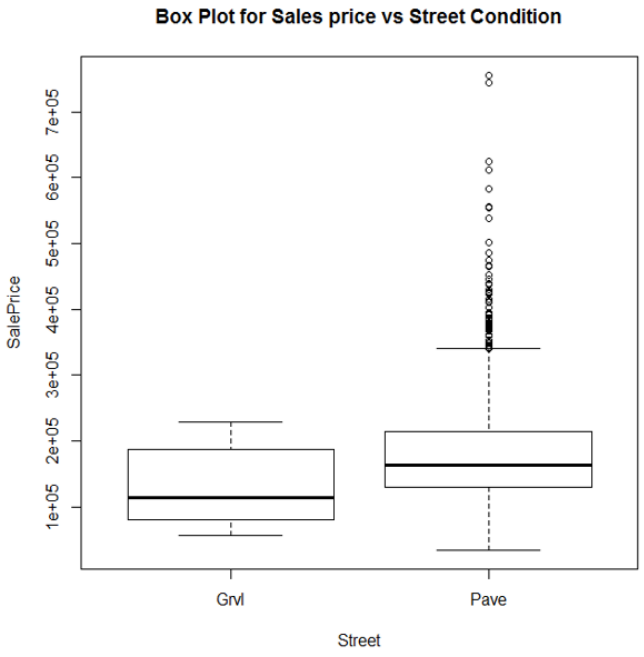
2000 houses have an inside lot configuration and 500 with corner configuration. Most of the houses (about 2500) are in normal condition when they are sold, with the Gable roof style. Most of the houses were single story as shown in bar chart, the next most popular was double story houses.





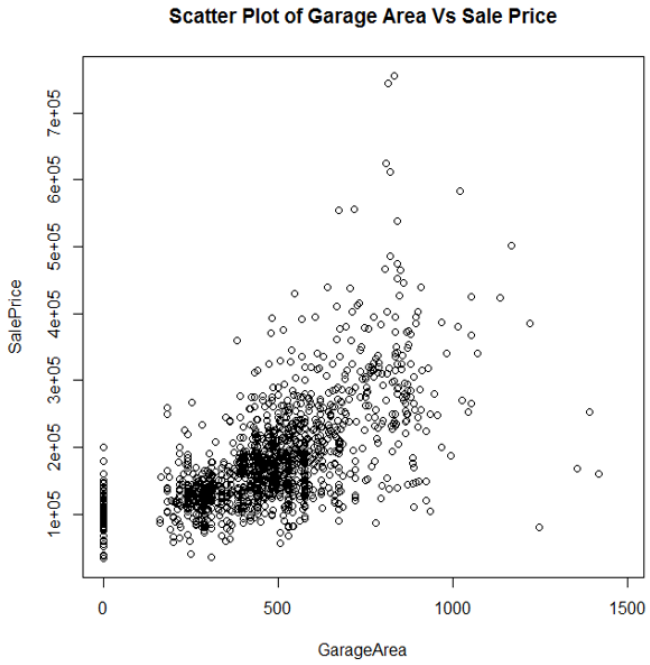
More than 1500 houses have attached garages, some are in the basement and about 800 were detached from houses. The bar chart of number of bedroom above ground floor indicates three bedrooms as the highest number in houses of that area then two bedrooms came at second place. As the number of bedrooms increases, the number of houses decreases. At the time of sale, majority of the houses were in normal position.

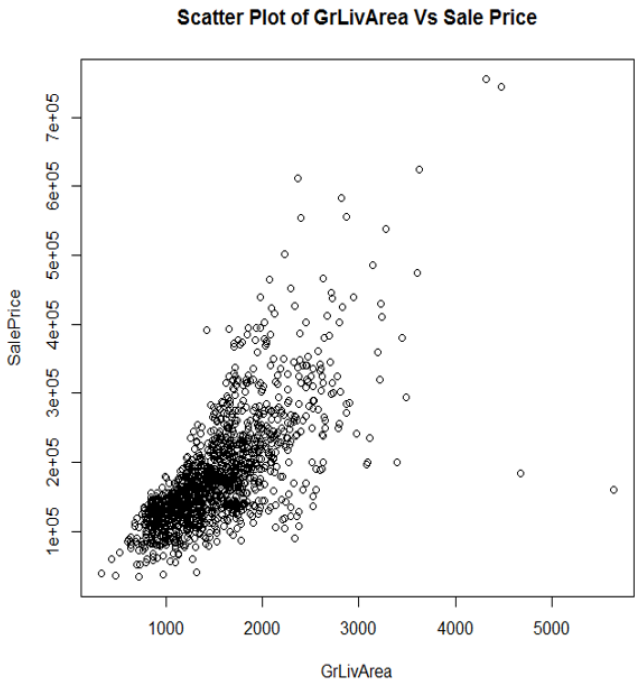




The Box plot for sale price of houses along with sale condition, indicates the houses with normal

condition have more variability in their sale prices as many observations are outside the box plot of “Normal” condition. Similarly, the paved streets shows more variability in sale prices.





The scatter plot shows the direction of relationship between two numerical variables of sales price, garage area and GrLivArea. Both scatter plots shows an increasing relationship. Increased garage area have the tendency to increase the sale price of houses.

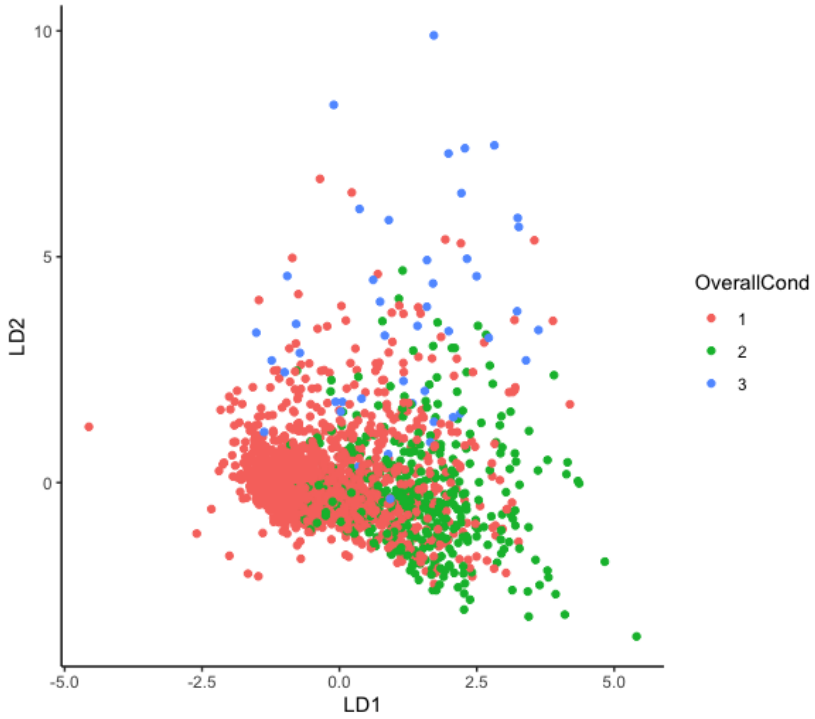
Descriptive Statistics	Sale Price	Garage Area	LotFrontage	Lot Area
Min	34900	0	21.00	1300
1st Quartile	129975	320	59.00	7478
Median	16300	480	68.00	9453
Mean	180921	472.9	69.31	10168
3rd Quartile	214000	576.0	80.00	11570
Max	755000	1488.0	313.00	215245
Missing values	1459	1	486	0

3.2 Logistic Regression Model

A multinomial Logistic Regression Model was used to predict overall condition of the house. The Target variable has values that range from 1 to 10 which is labelled as “poor”,”Average and “Good”. multinomial logistic regression is used to fit data and we get accuracy of 81.6.

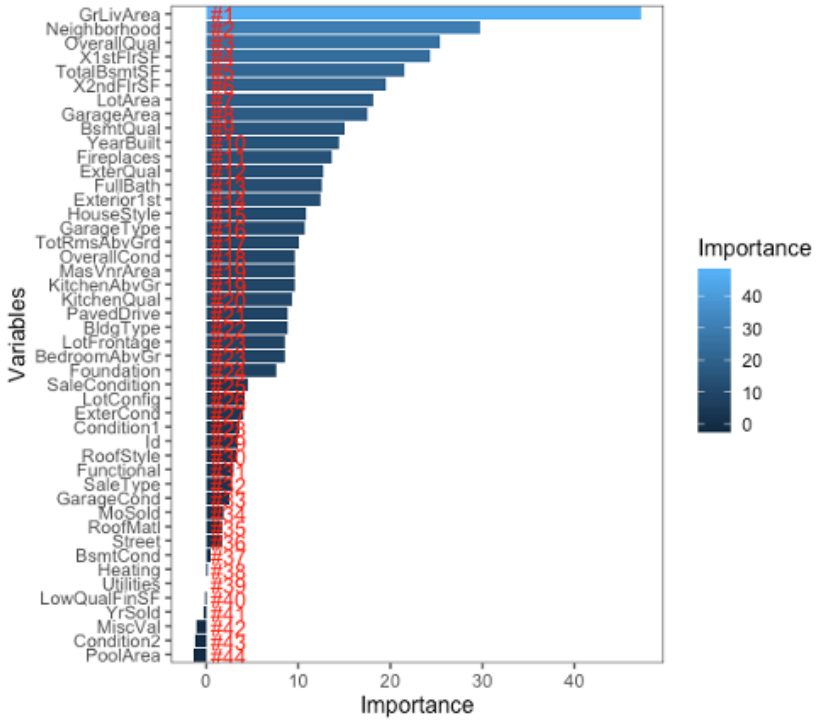
3.3 Linear Discriminant Analysis

We used multinomial logistic regression to predict the overall condition of the houses. Logistic regression is used for labels. Linear discriminant analysis is more suitable for predicting the category of an observation when the outcome variable has more than 2 classes. The LDA produces an accuracy of 81% hence here Multinomial regression and LDA produces almost same accuracy. This may be because multinomial regression is being used instead of logistic regression.



3.4 Random Forest and Boosting

Feature Importance: The importance of each features are represented as follows.

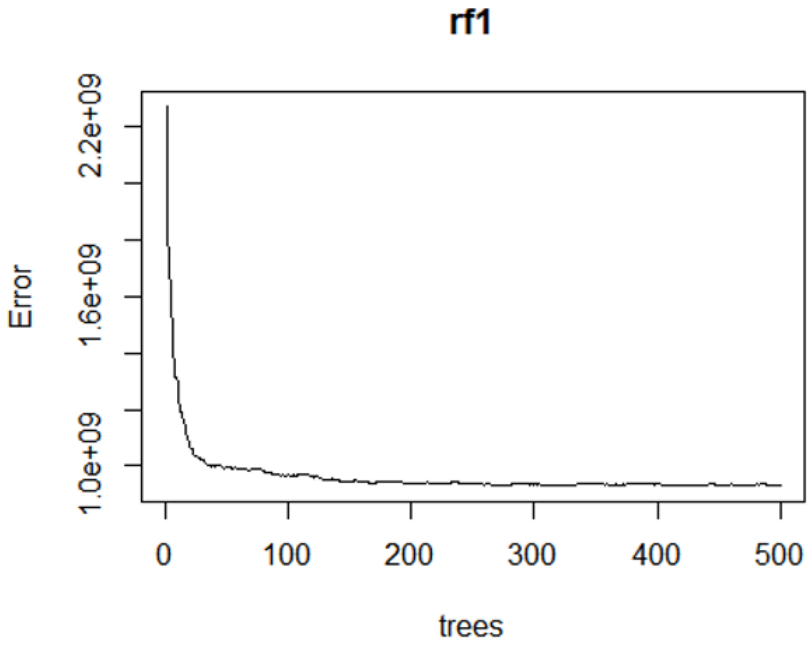


3.4.1 Random Forest

The random forest has a large number of individual decision trees that operate as an ensemble. The ensemble methods usually involve using multiple learning algorithms to obtain better predictive performance than using any of the single algorithms. Every tree in the random forest gives a class prediction and the class with maximum votes is considered the final output. The key idea of the random forest is the wisdom of crowds. That is some trees can predict wrong classes since all the tree's classes are considered and voted this method is much better than a single decision tree. To diversify each tree the following methods are used:

- Bagging - This process is carried out by choosing different samples with replacement so that the trees can be different from each other.
- Feature randomness - This means that all the features are considered while splitting a tree where the feature that has maximum separation.

The data after removing 60% of Na values and correlated features the remaining features are then used for the random forest model.



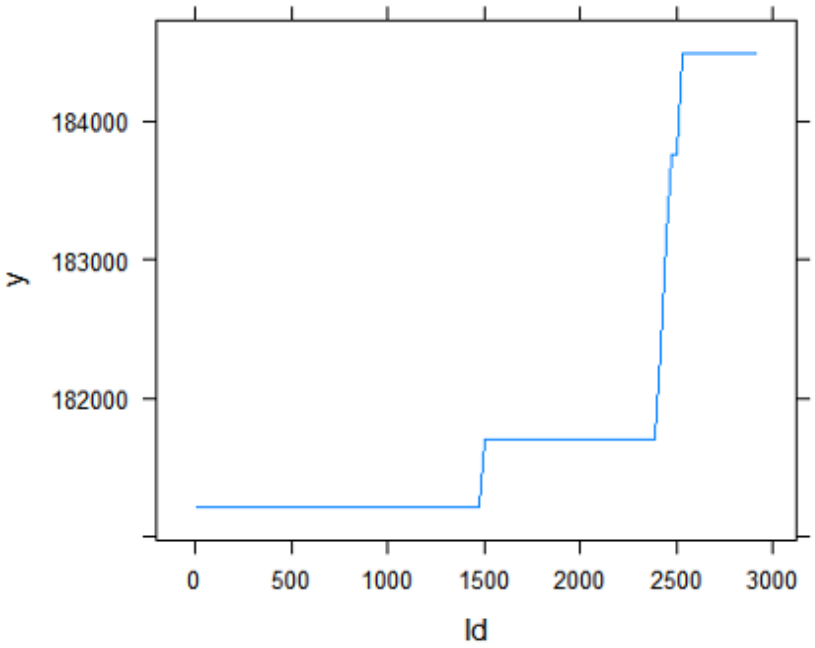
Then hyper parameter tuning for the number of trees is carried out which is represented as a table. The performance metric used is Mean Absolute Error which is given by the absolute value of the mean of the difference between the actual and predicted values

Number of Trees	MAE Score
100	18229.28
500	18105.88
1000	18170.59

From the table, we can see that MAE score is better at 500 trees, which is a mediocre value.

3.4.2 Boosting

The gradient boost method initially develops a decision tree with equal weights. Then the errors in the predicted values are then adjusted using the weights. Next the tree is constructed with the modified weights. The shortcomings are adjusted using the gradients in the loss function represented as y=ax+b+e.



The performance metric used is the same as the previous one. Mean Absolute Error which is given by the absolute value of the mean of the difference between the actual and predicted values. The distribution used for the regression model is Gaussian. Then the hyper-parameter tuning for the number of trees is carried out which is represented as a table.

Number of Trees	MAE Score
100	76253.17
500	77934.1
1000	78540.24

From the table, we can see that mae is minimum with 100 trees. So that gives better performance than others.

3.5 Validation Set Approach and Leave-One-Out Cross-Validation

3.5.1 Validation set Approach

The Validation Set Approach is a type of method that estimates a model error rate by holding out a subset of the data from the fitting process (creating a testing data set). The model is then built using the other set of observations (the training data set). Then the model result is

applied on the testing data set in which we can then calculate the error (testings data set error). In summary, this general idea allows for the model to not over fit. We randomly divide the available data into two parts, a training set, and a validation set. The model is fit on the training set, then the fitted model is used to predict the responses for the observations in the validation set. The performance of the models are measured using the mean absolute error(MAE).

Validation set approach	MAE Score
Random Forrest	20012.96
Boosting	23687.20

We can see clearly from the table that the MAE score is much lower for the Random forest method

3.5.2 Leave One Out Cross Validation

Leave one out cross validation(LOOCV) involves splitting the set of observations into two parts. However, instead of creating two subsets of comparable size, a single observation is used for the validation set, and the remaining data make up the training set. This tends to give us values that are highly variable, since it is based on a single observation. So we repeat the procedure n times by alliteratively leaving one observation out, and then computing the MAE of all n test estimates. This gives us much less bias, since the training set contains n - 1 observations and there is no randomness in the training/validation sets.

Leave-one-out Cross Validation	MAE Score
Random Forrest	33475.02
Boosting	36843.11

we can see again with the leave one out cross validation method we get a lower MAE score for the Random forest algorithm.

3.6 Research Questions

The method of multivariate analysis of variance (MANOVA) is used to test the differences between vectors of means. We are interested in understanding a houses over all condition

(overallcond). By fitting MANOVA to the house data and conducting a test to check if there is an influence of over all quality (overallqual) and year built based on different overallcond: “poor”, “average” and “good”. H_0:There is no influence of overallqual and year built based on different overallcond: “poor”, “average” and “good”. H_1:There is influence of overallqual and year built based on different overallcond: “poor”, “average” and “good”.

```
            Df  Pillai approx F num Df den Df    Pr(>F)
g            2 0.23591   194.98      4   5832 < 2.2e-16 ***
Residuals 2916
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The resultant MANOVA model reports a Pillai test statistic of 2 and a p-value below 0.05, thus $H_0$is rejected and it is concluded there are significant differences in the means. This result tell us that overallqual and year built do give an impact towards overallcond.

```
Response 1 :

             Df Sum Sq Mean Sq F value    Pr(>F)
g             2  415.1 207.562  112.38 < 2.2e-16 ***
Residuals   2916 5385.7   1.847
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response 2 :
             Df  Sum Sq Mean Sq F value    Pr(>F)
g             2  565189  282594  390.12 < 2.2e-16 ***
Residuals   2916 2112284     724
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Next, we perform individual Analysis of variance (anova) tests by using the function of anova

in R to check which of the dependent variables are statistically significant. Response 1 is overallqual and response 2 is year built. Based on the result above, p-value for both variables are below 0.05. Therefore, it is concluded that there are significant differences in means for the variables.

4 Conclusion

We can conclude that most of the houses sold in the area were in normal condition as they had not had much renovate and have a higher sale prices. After the pre-processing of data that is imputing with MICE and feature selection it can be observed the logistic regression to predict overall condition has better accuracy than Linear discriminant analysis which might be due to use of multinomial logistic regression.

The random forest with a higher number of tress works better for this problem than the gradient boosting method when mean absolute error is considered the performance metric. But the efficiency of the algorithms can certainly be improved by hyper-parameter tuning. As for this assignment only the number of trees are considered whereas other parameters like varying the number of features using more feature engineering methods might works as well.

Using Both the validation set approach and Leave-one-out cross validation on the random forest and gradient boosting methods show that in both cases, the random forest algorithm performs better in this task at predicting the sales price.

Since we are interested in the relationship between overallcond and the dependent variables which are overallqual and year built. MANOVA is the right method to test the difference in mean of the two groups. By undergoing MANOVA, we get the result that shows that both of the groups do give impact towards overallcond of houses. To get in dept of knowledge of which variables give an impact towards the overallcond, we undergo the ANOVA test. Based on the result of the ANOVA test, it shows that both of the variables do give impact to the overallcond.

5 References

- Machine Learning Basics - Gradient Boosting XGBoost -https://shirinsplayground.netlify.com/2018/11/ml_basics_gbm/
- Random Forest Regression -https://towardsdatascience.com/random-forest-and-i
- Hadjiantoni, S. (2020). Lecture 5: Multivariate normal distribution and MANOVA [powerpoint slides].
- Hadjiantoni , S. (2020). Computer Lab 3 Week 20 outline solutions.

6 Appendix

6.1 R-Code

```
Question 1


Data<-read.csv(file="house_data.csv",header=T,sep=",") #read dataset
Data<-Data[,-1]
attach(Data)
names(Data)
str(Data)
summary(Data) ##summarise the dataset
### Convert columns into factor
OverallQual<-as.factor(OverallQual)
OverallCond <-as.factor(OverallCond )
YearBuilt<-as.factor(YearBuilt)
Fireplaces<-as.factor(Fireplaces)
FullBath<-as.factor(FullBath)
BedroomAbvGr<-as.factor(BedroomAbvGr)
KitchenAbvGr<-as.factor(KitchenAbvGr)
TotRmsAbvGrd<-as.factor(TotRmsAbvGrd)


###Missing Value Plots
### mice package implements a method to deal with missing data
library(mice)
md.pattern(Data,rotate.names=T)
###VIM package implements a method for visualization of missing
and/or imputed values
library(VIM)
aggr_plot <- aggr(Data, col=c('navyblue','red'), numbers=TRUE,
```

```
sortVars=TRUE, bars=F,
labels=names(data), cex.axis=.4, gap=1, only.miss=T,combined=T,
ylab=c("Histogram of missing data","Pattern"))

imputedData <- mice(Data,m=5,maxit=50,meth='pmm',seed=500)
summary(imputedData)
completedData <- complete(imputedData,3)
completedData
###Descriptive Statistics
summary(OverallQual)
summary(OverallCond)
summary(KitchenAbvGr)
summary(Fireplaces)
summary(SalePrice)
summary(Data)
###Bar plots
countsLotConfig<-table(LotConfig)
BarLotConfig<-barplot(countsLotConfig,main="LotConfig")
countsCondition1<-table(Condition1)
BarCondition1<-barplot(countsCondition1,main="Condition1")
countsRoofStyle<-table(RoofStyle)
BarRoofStyle<-barplot(countsRoofStyle,main="Roof Style")
countsHouseStyle<-table(HouseStyle)
BarHouseStyle<-barplot(countsHouseStyle,main="House Style")
countsGarageType<-table(GarageType)
BarGarageType<-barplot(countsGarageType,main="Garage Type")
countsBedroomAbvGr<-table(BedroomAbvGr)
BarBedroomAbvGr<-barplot(countsBedroomAbvGr,main="Bedroom Above Ground")
countsSaleCondition<-table(SaleCondition)
```

```
BarSaleCondition<-barplot(countsSaleCondition,main="Sale Condition")

###Box Plots
boxplot(SalePrice~SaleCondition,main="Box Plot for Sales price vs
Sales Condition")
boxplot(SalePrice~Street,main="Box Plot for Sales price vs Street
Condition")

###Scatter Plots
plot(GarageArea,SalePrice, main="Scatter Plot of Garage Area Vs
Sale Price")

plot(GrLivArea,SalePrice, main="Scatter Plot of GrLivArea Vs Sale
Price")

Question 2

###Divide houses based to their overall condition (overallcond)
library(ggplot2)
library(reshape2)
library(mice)
library(dplyr)
library(tidyverse)
library(caret)
theme_set(theme_classic())

data<-read.csv("house_data.csv")
names(data)
str(data)
```

```
length(names(data))
(colMeans(is.na(data)))
#data=data[-1]
names(data)
unique(data$Street)
unique(data$Alley)
unique(data$Utilities)
unique(data$LotConfig)
unique.data.frame(data)
df<-data[which(colMeans(is.na(data))<=.60)]
names(df)
(colSums(is.na(df)))

###Divide overallcond into 3 categories (Poor, Average and Good)

names(df)
for(i in 1:nrow(df)){
if((df$OverallCond[i]>=1) & (df$OverallCond[i]<=3)){
   df$OverallCond[i]="Poor"
}
  else if((df$OverallCond[i]>=4) & (df$OverallCond[i]<=6)){
    df$OverallCond[i]="average"
 }
 else{
   df$OverallCond[i]="good"
}
}
#df[sapply(df, is.character)] <- lapply(df[sapply(df,
is.character)],as.factor)
```

```
#indx <- sapply(df, is.factor)
#indx
#for(i in indx){
#as.numeric((df[i]))
#}

installed.packages("mice")
library(mice)
colnames(df)[colSums(is.na(df)) > 0]
head(df)
str(df)
#meth[c("LotFrontage")]="norm"

#meth[c("Utilities")]="logreg"
#meth[c("MasVnrArea")]="norm"
#meth[c("BsmtQual")]="polyreg"
#meth[c("BsmtCond")]="polyreg"
#meth[c("TotalBsmtSF")]="norm"
#meth[c("KitchenQual")]="polyreg"
#meth[c("Functional")]="polyreg"
#meth[c("GarageType")]="polyreg"
#meth[c("GarageArea")]="norm"
#meth[c("GarageCond")]="polyreg"
#meth[c("SaleType")]="polyreg"
#meth[c("SalePrice")]="norm"
#set.seed(2000)
#imputed = mice(df1, method=meth, predictorMatrix=predM,
m=16,mnet.MaxNWts =2000)
imputed_Data <- mice(df, m=5, maxit = 50, method = 'pmm', seed =
```

```
500)
summary(imputed_Data)
completedData <- complete(imputed_Data,3)
#

#write.csv(imputed_Data,"/Users/shankar/Desktop/Applied
statistics\\imputed_Data.csv", row.names = FALSE)
#write.csv(imputed_Data,"Path where you'd like to export the
DataFrame\\File Name.csv", row.names = FALSE)
write.csv(completedData,"/Users/shankar/Desktop/Appliedstatistics/c
ompletedData.csv", row.names = FALSE)
data2<-read.csv("completedData.csv")
#data3<-as.data.frame(unclass(data2))
#str(data3)
#M2<-sapply(data2[,must_convert],unclass)
#data3<-cbind(data2[,!must_convert],M2)
must_convert<-sapply(data2,is.factor)
M2<-sapply(data2[,must_convert],unclass)
data3<-cbind(data2[,!must_convert],M2)

#x<-as.factor(c("Poor","average","good"))
#x
#data3$OverallCond=unclass(x)
str(data3)
library("Hmisc")

library(caret)
correlationMatrix <- cor(data3)
print(correlationMatrix)
```

```
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.8)
index=highlyCorrelated
final_data=data3[-c(index)]

final_data=final_data[]
names(final_data)

final_data$OverallCond<-factor(final_data$OverallCond)
final_data$OverallCond

#final_data=select(final_data,-c(1))
names(final_data)



### Splitting the data using function from dplyr package
library(caret)

index <- final_data$OverallCond%>%
createDataPartition( p = .70, list = FALSE)
train <- final_data[index,]
test <- final_data[-index,]
final_data$OverallCond <- relevel(final_data$OverallCond, ref = '1')
require(nnet)
# ## Training the multinomial model
#multinom_model <- multinom(OverallCond ~ ., data = final_data)
multinom_model <- nnet::multinom(OverallCond~., data = final_data)
output <- summary(multinom_model)
print(output)
z <- output$coefficients/output$standard.errors
```

```
p <- (1 - pnorm(abs(z), 0, 1))*2 # we are using two-tailed z test
p
# ##Checking the model
summary(multinom_model)
exp(coef(multinom_model))
head(round(fitted(multinom_model), 2))
# ## Predicting the values for train dataset
# ## Predicting the values for train dataset
train$overallcondPredicted <- predict(multinom_model, newdata =
train, "class")

### Building classification table
tab <- table(train$OverallCond, train$overallcondPredicted)

### Calculating accuracy - sum of diagonal elements divided by
total obs
round((sum(diag(tab))/sum(tab))*100,2)
### Predicting the class for test dataset
test$overallcondPredicted <- predict(multinom_model, newdata =
test, "class")

### Building classification table
tab <- table(test$OverallCond, test$overallcondPredicted)
tab
round((sum(diag(tab))/sum(tab))*100,2)

## Linear discriminant analysis
# Estimate preprocessing parameters
library(MASS)
```

```
training.samples <- final_data$OverallQual %>%
createDataPartition(p = 0.8, list = FALSE)
train.data <- final_data[training.samples, ]
test.data <- final_data[-training.samples, ]
### Normalize the data. Categorical variables are automatically
ignored.
### Estimate preprocessing parameters
preproc.param <- train.data %>%
preProcess(method = c("center", "scale"))
### Transform the data using the estimated parameters
train.transformed <- preproc.param %>% predict(train.data)
test.transformed <- preproc.param %>% predict(test.data)
### Fit the model
model <- lda(OverallCond~., data = train.transformed)
model
### Make predictions
predictions <- model %>% predict(test.transformed)
names(predictions)

### Model accuracy
mean(predictions$class==test.transformed$OverallCond)
### Predicted classes
head(predictions$class, 6)
### Predicted probabilities of class membership.
head(predictions$posterior, 6)
### Linear discriminant analysis
head(predictions$x, 3)
### Create the LDA plot using ggplot2 as follow:
```

```
lda.data <- cbind(train.transformed, predict(model)$x)
ggplot(lda.data, aes(LD1, LD2)) +
geom_point(aes(color = OverallCond))
#Model accuracy:

mean(predictions$class==test.transformed$OverallCond)
```

Question 3

a. Random Forest and Boosting to predict house prices

```
### Random forest works by aggregating predictions made by multiple
decision tress of varying depth
library(randomForest)
require(caTools)
dat<-read.csv("completedData.csv")
summary(dat)
dat$Street<-as.factor(dat$Street)
dat$Utilities<-as.factor(dat$Utilities)
dat$LotConfig<-as.factor(dat$LotConfig)
dat$Neighborhood<-as.factor(dat$Neighborhood)
dat$Condition1<-as.factor(dat$Condition1)
dat$Condition2<-as.factor(dat$Condition2)
dat$BldgType<-as.factor(dat$BldgType)
dat$HouseStyle<-as.factor(dat$HouseStyle)
dat$OverallCond<-as.factor(dat$OverallCond)
dat$RoofStyle<-as.factor(dat$RoofStyle)
```

```
dat$RoofMatl<-as.factor(dat$RoofMatl)
dat$Exterior1st<-as.factor(dat$Exterior1st)
dat$ExterQual<-as.factor(dat$ExterQual)
dat$ExterCond<-as.factor(dat$ExterCond)
dat$Foundation<-as.factor(dat$Foundation)
dat$BsmtQual<-as.factor(dat$BsmtQual)
dat$BsmtCond<-as.factor(dat$BsmtCond)
dat$Heating<-as.factor(dat$Heating)
dat$KitchenQual<-as.factor(dat$KitchenQual)
dat$Functional<-as.factor(dat$Functional)
dat$GarageType<-as.factor(dat$GarageType)
dat$GarageCond<-as.factor(dat$GarageCond)
dat$PavedDrive<-as.factor(dat$PavedDrive)
dat$SaleType<-as.factor(dat$SaleType)
dat$SaleCondition<-as.factor(dat$SaleCondition)
summary(dat)
sample = sample.split(dat$SalePrice, SplitRatio = .75) ###split
data into 2 dataset (test and train)
train = subset(dat, sample == TRUE)
test = subset(dat, sample == FALSE)
########
require(modelr)
rf <- randomForest(SalePrice ~ .,data=train)

pred = predict(rf, test) ### predict house prices using
randomforest
n<-c()
n<-test$SalePrice - pred
mae <- function(error)
```

```
{
  mean(abs(error))
}
mae(n)
rf1 <- randomForest(SalePrice ~ .,data=train,ntree=1000)
pred1 = predict(rf1, test)
n1 <- test$SalePrice - pred1
mae(n1)
####### Boosting:
install.packages("gbm")
library(gbm)
boost=gbm(SalePrice ~ .,data=train,
distribution="gaussian",n.trees=1000)
boost_pred=predict(boost,train,n.trees=1000) ### Predict house
prices using boosting
n2<-boost_pred-test$SalePrice
mae(n2)
sum(boost_pred-test$SalePrice)/length(boost_pred)
length(boost_pred)
head(test$SalePrice)
boost$train.error
```

b. Validation set approach and leave-one-out cross-validation to estimate the test error

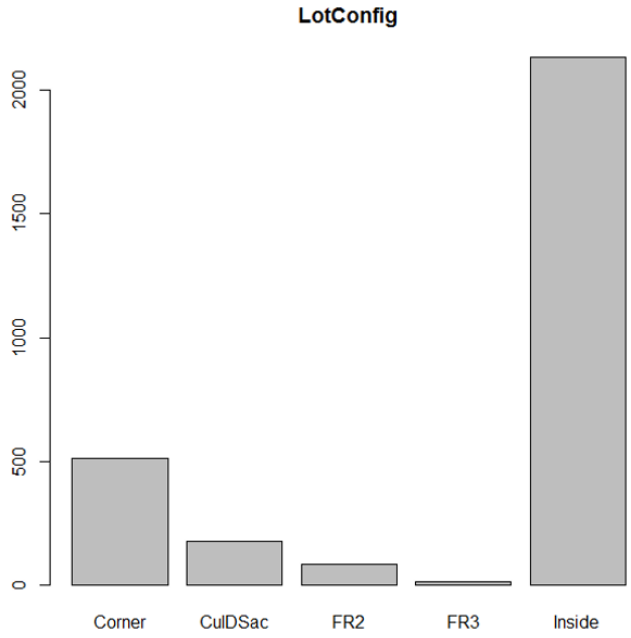
```
######## validation set approach ########

### We create a set of observations to calcualte the error on
dat1<-dat[sample(nrow(dat), 700), ]
```

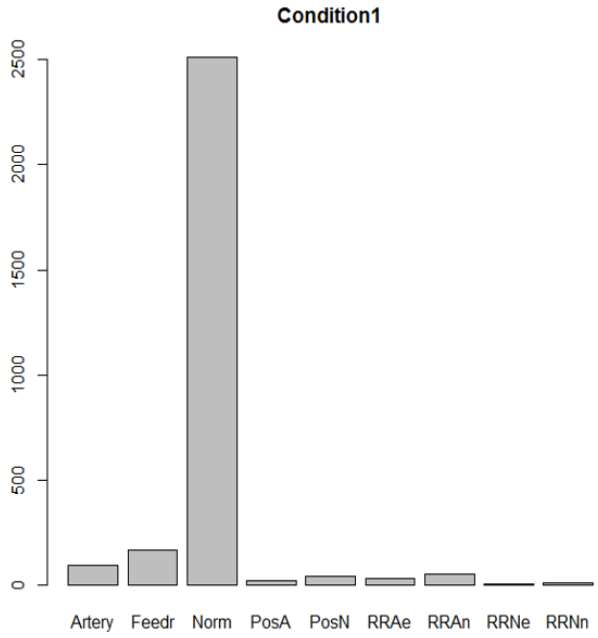
```
#split the data into 50% testing and 50% training
training.samples1 <- dat1$SalePrice %>%
createDataPartition(p = 0.5, list = FALSE)
train.val <- dat1[training.samples1, ]
test.val <- dat1[-training.samples1, ]

### Validation set approach on RF
#fit the model using the training data
rfval <- randomForest(SalePrice ~ .,data=train.val)

### Predict on the testing dataset
predval<-predict(rfval, test.val)
errorval<-(test.val$SalePrice - predval)
mae(errorval)

### Calculate Root Mean Squared Error.
RMSE<- sqrt(mae(errorval))
RMSE

### Validation set approach on Boosting
### Fit the model using traing data
boostval=gbm(SalePrice ~ .,data=train.val,
distribution="gaussian",n.trees=1000)

### Predict on the testing dataset
boostpredval=predict(boostval,test.val,n.trees=1000)
booterror<-test.val$SalePrice - boostpredval
mae(booterror)
```

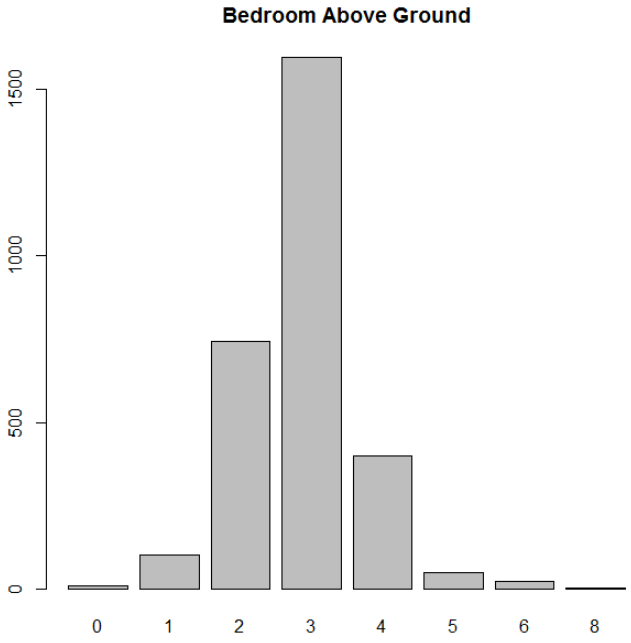
```
### Calculate Root Mean Squared Error.
RMSE_boost<- sqrt(mae(booterror))
RMSE_boost


library(MASS)
library(ipred)
library(tree)
library(ada)

##### Leave one out - cross validation method ####

## LOO - cross validation- RF ###

Saleprice.rf <- randomForest(SalePrice ~ ., data=train.val)
Saleprice.rf

mypredict.randomForest <- function(object, newdata)
predict(object, newdata = newdata, type = c("response"))

rf_cv<- errorest(SalePrice ~ ., data= test.val, model=randomForest,
estimator = "cv", est.para = control.errorest(k = 349),
predict= mypredict.randomForest)
# plot loss function as a result of n trees added to the ensemble
rf.perf(rf_cv, method = "cv")


## LOO - cross validation- Boosting
library(adabag)
```

```
saleprice_cv<- gbm(
formula = SalePrice ~ .,
distribution = "gaussian",
data = train.val,
n.trees = 10000,
 interaction.depth = 1,
shrinkage = 0.001,
cv.folds = 351,
n.cores = NULL, # will use all cores by default
verbose = FALSE)

print(saleprice_cv)

# get MSE and compute RMSE
sqrt(min(saleprice_cv$cv.error))

### Plot loss function as a result of n trees added to the ensemble
gbm.perf(saleprice_cv, method = "cv")



Question 4

### Manova and anova to test if there is an influence of year built
and overallqual towards overallcond

mydata<-house_data

cond<- c()### if else statement for clustering condition into 3
```

```
categories
for (i in 1:length(mydata$OverallCond)){
  if(mydata$OverallCond[i] <= 3){
    cond<-append(cond, 'poor')
  }
  else if(mydata$OverallCond[i] >= 4 & mydata$OverallCond[i] <= 6){
    cond<-append(cond, 'average')
  }
  else{
    cond<-append(cond, 'good')
  }
}
cond

table(cond)
x<-as.matrix(cbind(mydata$OverallQual,mydata$YearBuilt))###
Dependent variables formatted as matrix
g<-as.vector(cond)
m<-manova(x~g)### Manova for dependent variables with response
variable
summary(m)

anova<-(aov(x~g))### Anova for each dependent variable with
response variable
summary(anova)
```

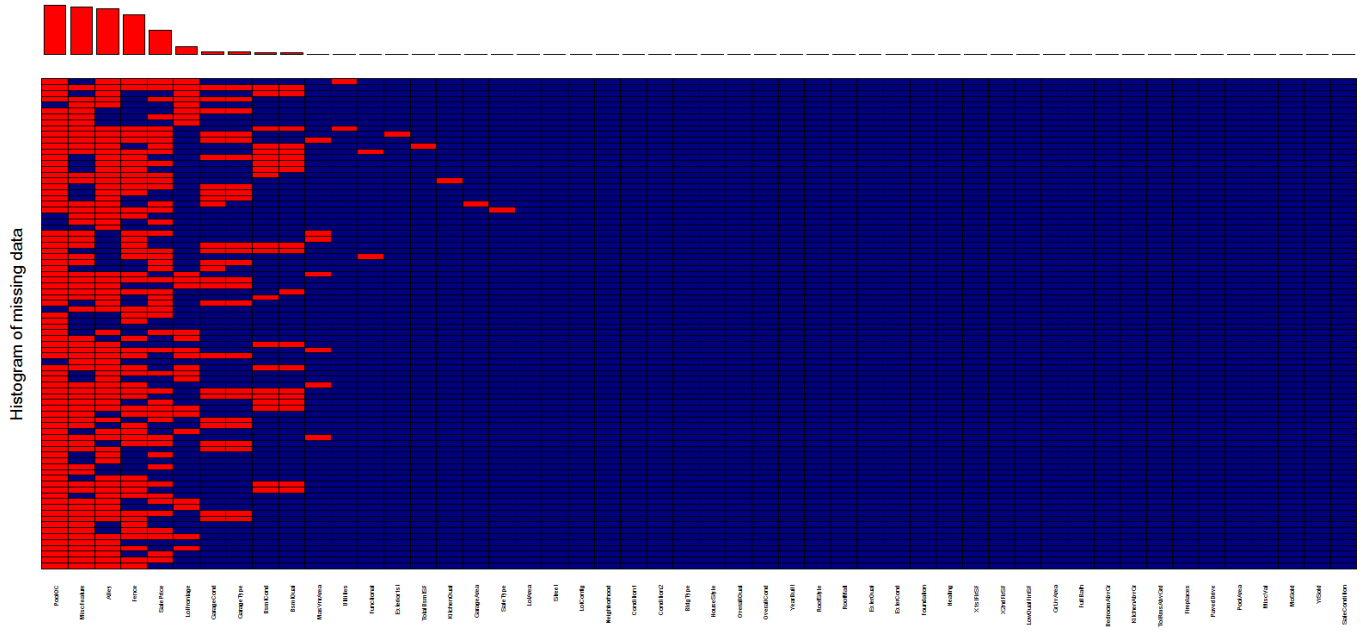

LotConfig
0
500
1000
1500
2000
Corner
CulDSac
FR2
FR3
Inside


Condition1
0
500
1000
1500
2000
2500
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNe
RRNn


Bedroom Above Ground
0
500
1000
1500
0
1
2
3
4
5
6
8

Over All Quality Rating	Counts
1	4
2	13
3	40
4	226
5	825
6	731
7	600
8	342
9	107
10	31

Over All Condition	Counts
1	7
2	10
3	50
4	101
5	1645
6	531
7	390
8	144
9	41



Missing Values Exploration

6.2 Contributions

A meeting was held once the project details were provided, and It was decided that each person will proceed with the analysis and to have frequent meeting to discuss on their progress and methodology used and proceed with the method that is more appropriate. We all worked on all four questions and then together choose the best code to proceed with at each stage resulting in

the following breakdown:

Rohini Subramaniam 1908736: Preliminary analysis, Feature selection, Logistic Regression, LDA, Random Forest, conclusions

Srinidhi Karthikeyan 1900637: Random Forest, Boosting, conclusion

Divya Arora 1901423: Latex Formatting, Title page, Contents page, Analysis - Numerical and Graphical summaries, Conclusion

Danisha Rajoo 1907654: Latex Formatting - report compilation, Introduction, Validation Set approach and Leave one out cross validation, conclusion

Rosli, Nur FFB 1908520: Question 4 - Research Questions, conclusion, Appendix