Department of Statistics, SPPU
ST-O13 Statistical Learning and Data Mining
Internal-1 Assignment 2023-24

Name: 1)Devendra Sanjay Patil - 2234

2)Prajakta Pandurang Madane - 2231

Code: https://github.com/Dev1356/ETE_ASSINMENT_2023.git

Title : House Price Prediction

Introduction :

'Houses' dataset comprises a collection of prices of new and resale houses located in the metropolitan areas of India, the amenities provided for each house.
The analysis aims to compare the housing costs across various cities and see if the effects of other factors are similar across cities.
Cities are Mumbai,Kolkata,Hydrabad,Delhi and Chennai select any one of the data of city And do analysis on that city fit some statistical model and machine learning model and fit same models on another cities, check that the model performance in different cities.

AIM : We are going to answers the following questions

Q.1. Identify which variables are useful in predicting the prices. Justify their usefulness with the help of appropriate exploratory and/or descriptive statistical techniques.

Q.2. Try to build various supervised learning models using the variables selected from
above Use five-fold cross-validation. Compare their performances using appropriate measures.

Q.3. Do the models for predicting housing costs differ across cities? Interpret these differences (if any) in detail.

First I am considering Mumbai City data and doing all Statistical and machine learning techniques, fit all possible models on Mumbai city then

one by one take another cities and checking that the models are performing same according all cities or not

Data Cleaning:

Consider Mumbai city data,

```
D=read.csv("C:\\Users\\Shiv\\Documents\\data mining\\Assingment 1\\Houses\\Mumbai.csv")
```

Total No. of Houses given in data

```
[1] 7719
```

Dependent Variable (Response):

```
Price
```

Independent Variables(Regressors):

1)Area
2)Location
3)No. of Bedrooms
4)Resale
5)MaintenanceStaff
6)Gymnasium
7)SwimmingPool
8)LandscapedGardens
9)JoggingTrack
10)RainWaterHarvesting
11)IndoorGames
12)ShoppingMall
13)Intercom
14)SportsFacility
15)ATM
16)ClubHouse
17)School
18)24X7Security
19)PowerBackup
20)CarParking
21)StaffQuarter
22)Cafeteria
23)MultipurposeRoom
24)Hospital
25)WashingMachine
26)Gasconnection
27)AC
28)Wifi
29)Children'splayarea
30)LiftAvailable
31)BED
32)VaastuCompliant
33)Microwave
34)GolfCourse
35)TV
36)DiningTable
37)Sofa
38)Wardrobe
39)Refrigerator

Data contains 40 variables one is Price which dependent variable and other 39 are inde pendent variables.
All this variables are not of same type,

Price and Area - Continuous Variables
Location and No.of Bedrooms - Categorical Variables
All remaining - Binary Variable

Missing Values in data Data :

```
Price                          0
Area                           0
Location                       0
No..of.Bedrooms                0
Resale                         0
MaintenanceStaff            6321
Gymnasium                   6321
SwimmingPool                6321
LandscapedGardens           6321
JoggingTrack                6321
RainWaterHarvesting         6321
IndoorGames                 6321
ShoppingMall                6321
Intercom                    6321
SportsFacility              6321
ATM                         6321
ClubHouse                   6321
School                      6321
X24X7Security               6321
PowerBackup                 6321
CarParking                  6321
StaffQuarter                6321
Cafeteria                   6321
MultipurposeRoom            6321
Hospital                    6321
WashingMachine              6321
Gasconnection               6321
AC                          6321
Wifi                        6321
Children.splayarea          6321
LiftAvailable               6321
BED                         6321
VaastuCompliant             6321
Microwave                   6321
GolfCourse                  6321
TV                          6321
DiningTable                 6321
Sofa                        6321
Wardrobe                    6321
Refrigerator                6321
```

Variables like Price , Location , Area and No. of Bedrooms they don't have missing values .

Data contain too much missing values , if we drop that values then most of the information get lost from data and predictions after droping the missing values become biased.

So, we just can't throw the missing values, To get the missing values there are so many method of imputation.

Imputation of Missing Values :

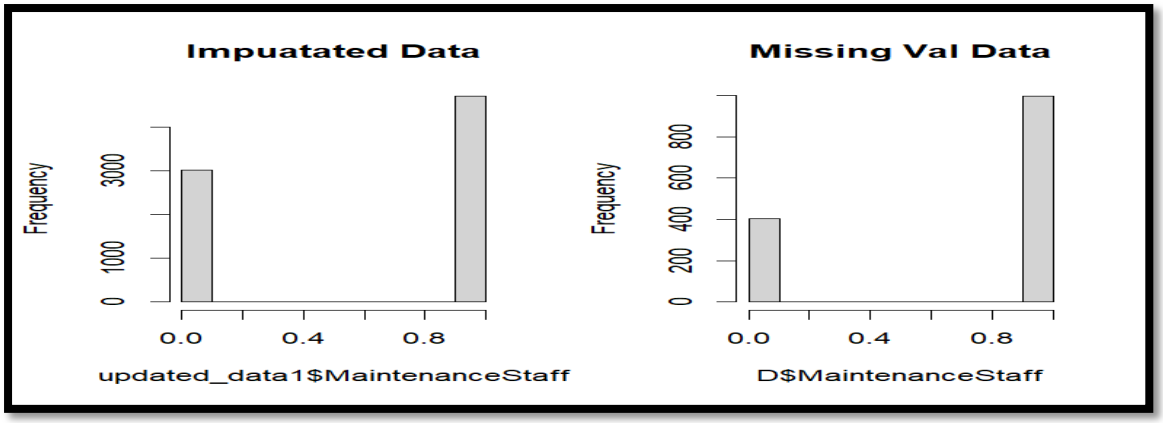
There is a in built function in r programming that is mice() function,

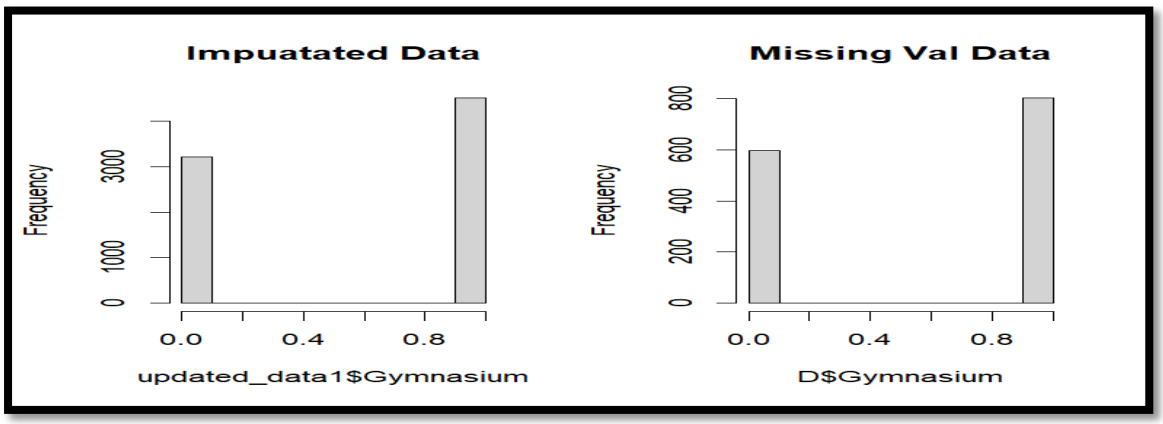
Multivariate Imputation by Chained Equations (mice)

The **mice** package implements a method to deal with missing data. The package creates multiple imputations (replacement values) for multivariate missing data. The method is based on Fully Conditional Specification, where each incomplete variable is imputed by a separate model. The MICE algorithm can impute mixes of continuous, binary, unordered categorical and ordered categorical data. In addition, MICE can impute continuous two-level data, and maintain consistency between imputations by means of passive imputation. Many diagnostic plots are implemented to inspect the quality of the imputations.

In following graph showing some examples that imputated data has also a same proportion values as that of the data with missing values

Which means that imputated data is better for the further analysis.





Outlier Detection and Removal:

Consider "Location" is categorical variable but it has 415 categories unique categories but there some Locations in Data which has less than 10 houses, just stored all such locations in one category that is in "other"

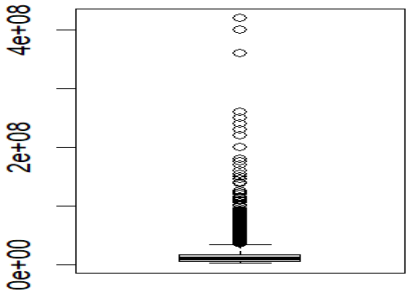
```
Table=(table(updated_data1$Location))

locations_to_replace <- names(Table[Table<11])

#Replace those locations with a new name, e.g., "Other"

updated_data1$Location[updated_data1$Location %in% locations_to_replace]=
"Other"

length(unique(updated_data1$Location))

[1] 100
```

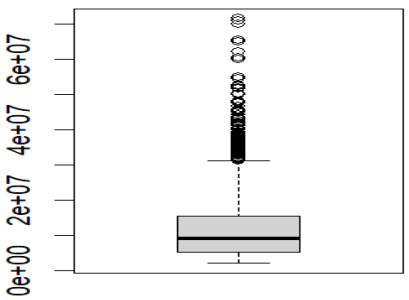
Now we just left with unique categories with 100.

Statistical Method to remove outlier:

1) R-Student

```
model_outler=lm(Price~ ., data = updated_data1)

# Detect outliers using studentized residuals

outliers= abs(rstudent(model_outler)) > 1  # Adjust the threshold as
needed

# Remove outliers from the dataset

D_clean=updated_data1[!outliers, ]

dim(D_clean)

[1] 6983   40
```

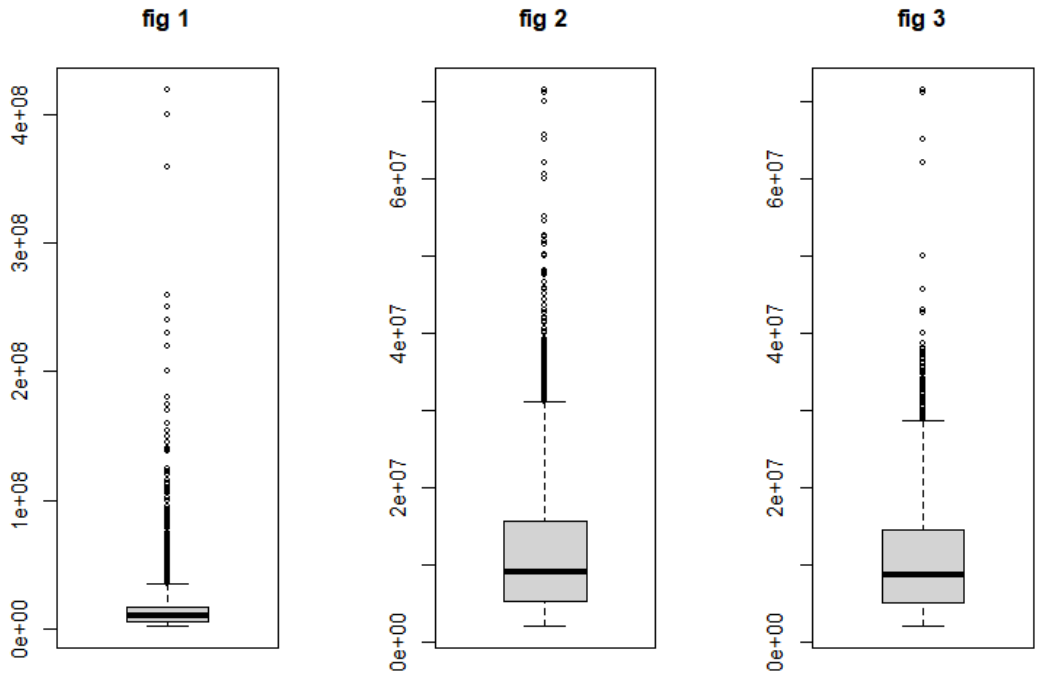




See from the Boxplot left side graph is plot before removing the outlier and right side plot shows that after using the R-Student method some outlier get removed from the data

2) Cook's Distance

```
cooksmodel=lm(Price~.,data=D_clean)
cooksd =cooks.distance(cooksmodel)
# Set a threshold for Cook's Distance
cook_threshold = 3* mean(cooksd)
# Identify and remove outliers
D_clean2 =D_clean[cooksd <= cook_threshold, ]
dim(D_clean2)
[1] 6488   40
```



Clearly we can see the difference all three boxplots, fig 3 shows the boxplot of "price" data after using method cook's distance
Now at some extent outlier get removed from the data.

One Hot Encoding:

Location variable is categorical variable with unique categories of 415 and after cleaning data there are only 100 categories are remaining , data contains so many location with frequency less than 11 and if we keeping all this location can increase the dimension of data which cause effect on the model performance.

```
Location1=updated_data2$Location
> df11 <- dummyVars("~.", data=updated_data2)
> df22 <- data.frame(predict(df11, newdata =updated_data2))
```

Feature Selection:

Data contain too many independent variables and there some variables which are not actually related to the target variable “Price” there are some techniques or statistical methods which is used to remove that unnecessary variables from data. One of the method is Stepwise Variable Selection this method has 3 type

a) Forward Selection
b) Backward elimination
c) Combination of both process

Combination of both the process can give more correct result.

```
Var_M=lm(Price~.,data=updated_data2[,-3])
Var_select1=step(Var_M,direction = "both")
Sel_cof1=names(Var_select1$coefficients)
Sel_cof1=Sel_cof1[-1]
Sel_cof1
Var_data=updated_data2[,Sel_cof1]

[1] "Area"              "No..of.Bedrooms"    "Resale"            "MaintenanceStaff"

 [5] "SwimmingPool"      "RainWaterHarvesting" "ShoppingMall"      "Intercom"

 [9] "School"            "PowerBackup"        "CarParking"        "Cafeteria"

[13] "Hospital"          "VaastuCompliant"    "DiningTable"       "Wardrobe"

[17] "Refrigerator"
```

17 Variables get selected by the process ,“Location” variable keep aside because in general we know that location is basically very important factor for the price of house, from the remaining 38 variables 17 are selected by the process.

Stepwise selection can take different model at every step and find there AIC(Akaika Information Criterion) and select the model which has lower AIC, so the model with above 17 variables has lower AIC that’s why there are get selected.

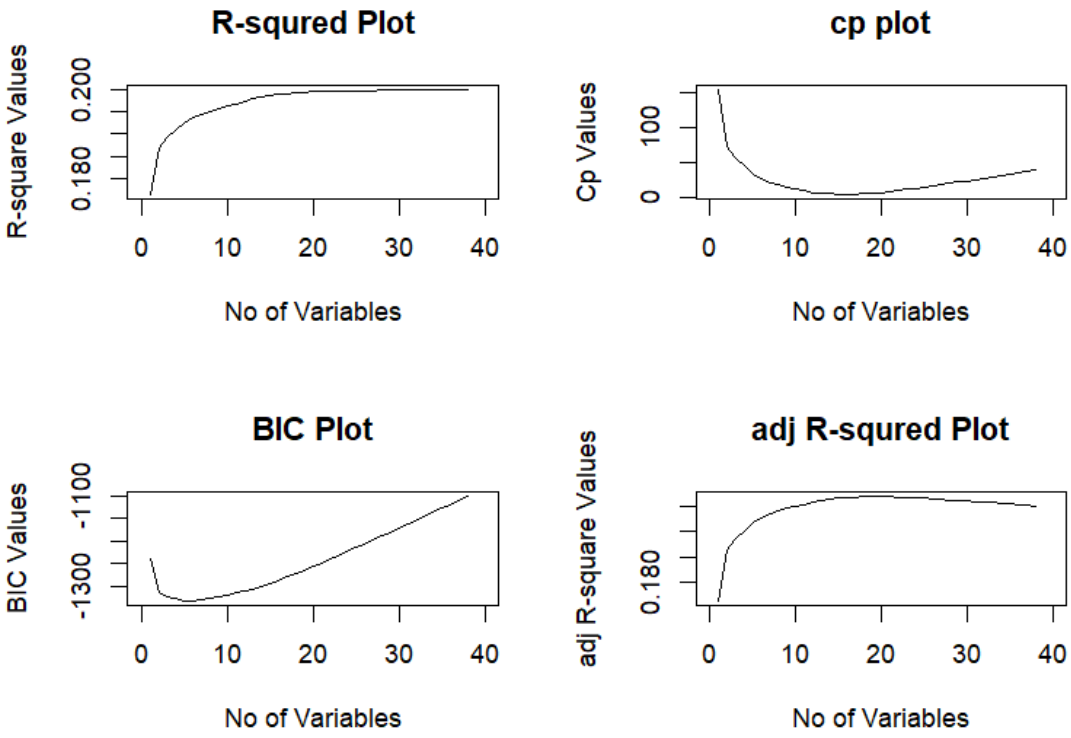
To check that the above selected variable are significant or not, there is one function called “regsubsets” in library called “leaps” .

This function is also used for variable selection ,inside this function we can see the R-squared, adj R-squared, BIC and Cp values.

See the below code and plots that which variables are explaining most of the variation

```
> breg_full = regsubsets(Price ~ .-Location, data = updated_data2,really.b
ig = T,nvmax =38 )

> breg_summary = summary(breg_full)

> breg_summary$rsq

 [1] 0.1763619 0.1867042 0.1891091 0.1906446 0.1924159 0.1936143 0.1945263
0.1951935
 [9] 0.1958515 0.1964244 0.1968730 0.1975204 0.1979471 0.1982593 0.1986274
0.1988732
[17] 0.1990609 0.1992715 0.1994051 0.1995185 0.1995890 0.1996416 0.1996949
0.1997463
[25] 0.1997772 0.1998058 0.1998301 0.1998580 0.1998865 0.1999028 0.1999110
0.1999188
[33] 0.1999248 0.1999280 0.1999288 0.1999291 0.1999293 0.1999295

> par(mfrow=c(2,2))

> plot(breg_summary$rsq,main="R-squred Plot",xlab="No of Variables",ylab="
R-square Values",type="l",xlim=c(0,40))

> plot(breg_summary$cp,main="cp plot",xlab="No of Variables",ylab="Cp Valu
es",type="l",xlim=c(0,40))

> plot(breg_summary$bic,main="BIC Plot",xlab="No of Variables",ylab="BIC V
alues",type="l",xlim=c(0,40))
> plot(breg_summary$adjr2,main="adj R-squred Plot",xlab="No of Variables",
ylab="adj R-square Values",type="l",xlim=c(0,40))
```

See the from above code the values of R-Squared there is no change in values of R-squared as much between all the variables, if we just consider “Area”,”No. of Bedrooms” and “Resale” this three variables for model building and all the variables for the model building there no change in the performance that model because most of the variation is explain by this three variable and one variable that we kept aside that is “Location”.



From this plots let’s see that after some threshold the curve for R-squared, cp and adj R-squerd plot become parallel to the x-axis and for BIC get increase from some threshold point.

Which means that only few variables are important for model building all other variable are not contributing to increase or decrease the “Price” of Houses.

Model Building:

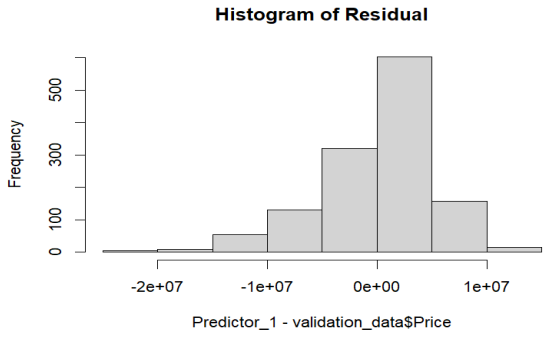
For model building there are many supervised learning model,Given data contains response and predictors, so we can used Regression technique for model building

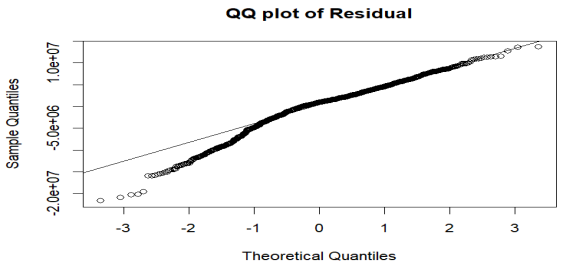
From above feature selections techniques we get know that there are only first 4 variables are contributing for price prediction.

“Area”, “Location”, ”No.of Bedrooms”, ”Resale”

1) Multiple Linear regression

```
Train_vector=Validaton_vector=c()
for(i in 1:k=5)
{
sample_data=sample(1:n,size=n,replace = FALSE)
Folds=matrix(sample_data,ncol=k)
FirstFold=Folds[,i]
validation_data=D_clean2[FirstFold,]
train_data=D_clean2[-FirstFold,]
M1=lm(Price~Area+as.factor(Location)+No..of.Bedrooms+Resale-1,data=trai
n_data)
Train_vector[i]=(anova(M1)$"Sum Sq"[5])/nrow(train_data)
Predictor_1=predict(M1,newdata=validation_data)
Validaton_vector[i]=mean((validation_data$Price-Predictor_1)^2)
}
```

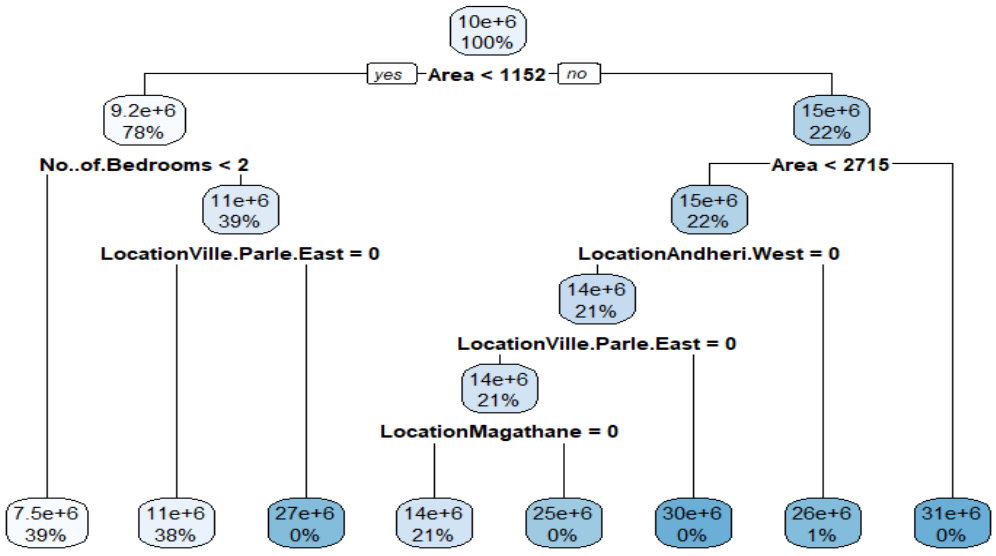


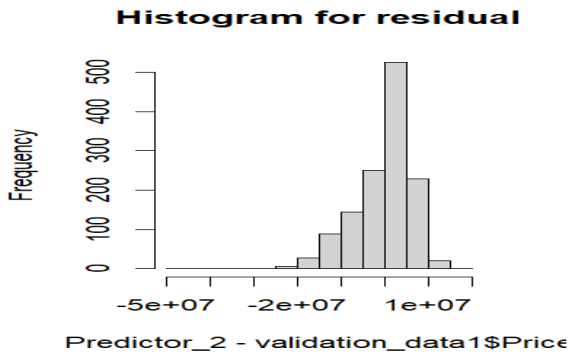


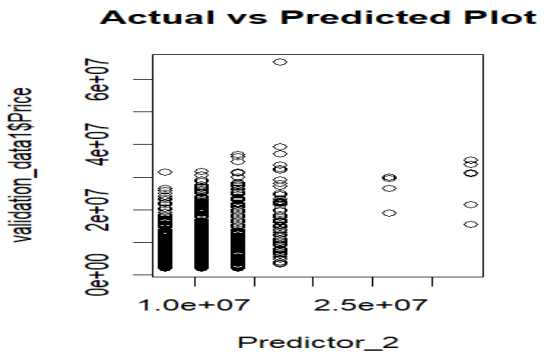
See from the above plot that histogram for Residual is kind of Normally distributed and the QQ plot and Histogram both shows that Distribution of Residuals is kind follows Normal distribution

2) Decision Trees

```
for(i in 1:k=5)
{

sample_data1=sample(1:n1,size=n1,replace = FALSE)

Folds1=matrix(sample_data1,ncol=k)

FirstFold1=Folds1[,i]

validation_data1=D_clean2[FirstFold1,]

train_data1=D_clean2[-FirstFold1,]

M2=rpart(Price~Area+as.factor(Location)+No..of.Bedrooms+Resale-1,data=trai

n_data1,method="anova",cp=0.01)

Train_vector1[i]=mean((train_data1$Price-(predict(M2,newdata=train_data1))
)^2)

Predictor_2=predict(M2,newdata=validation_data1)

Validaton_vector1[i]=mean((validation_data1$Price-Predictor_2)^2)

}
```



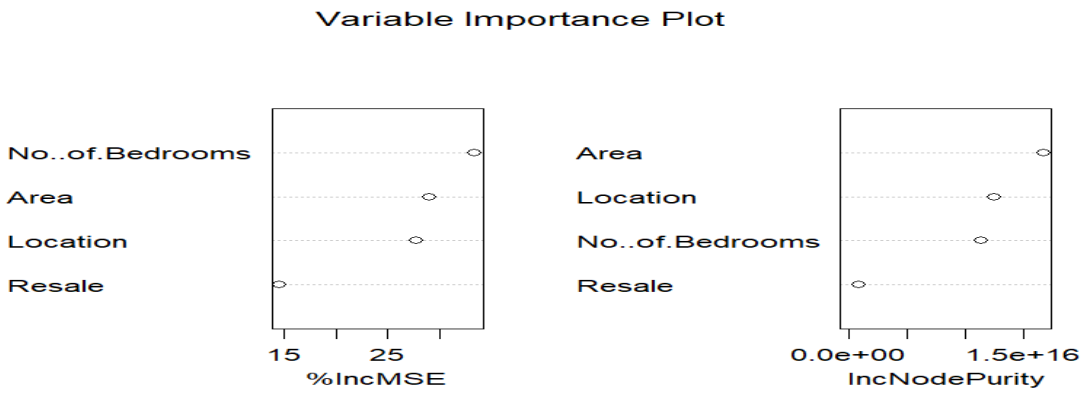




See from the above graphs histogram of Residual is kind of normal and another graph is for Actual and Predicted Values of Price.

3) RandForest

```
randomForest_sample=sample(1:n1,size=n1/3,replace = FALSE)

validation_data_forest=D_clean2[randomForest_sample,]

train_data_forest=D_clean2[-randomForest_sample,]

M3=randomForest(Price~Area+Location+No..of.Bedrooms+Resale,data=train_d
ata_forest,importance=TRUE)

MSE_Train=mean((train_data_forest$Price-(predict(M3,newdata=train_data_for
est)))^2)

Predictor_forest=predict(M3,newdata=validation_data_forest)

MSE_Validation=mean((validation_data_forest$Price-Predictor_forest)^2)
```

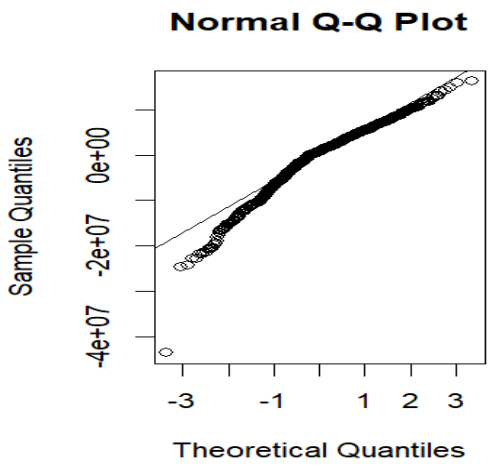


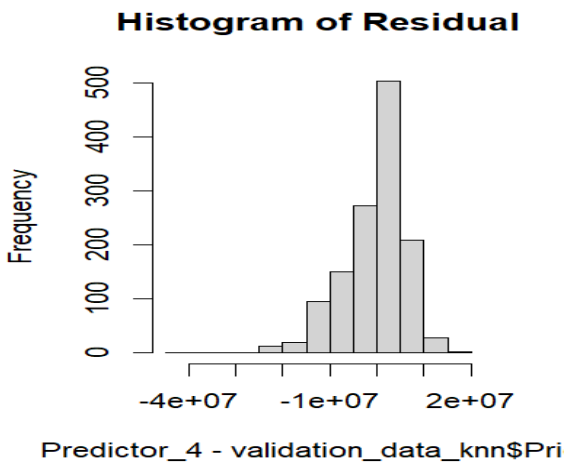
From this 4 variable No. of bedrooms are important as compare to MSE and Area is important by comparing there Node Purity.

4) K- Nearest Neighbour –KNN

```
MSE_Train_knn=MSE_val_knn=c()

for(i in 1:k)

{

sample_data_knn=sample(1:n2,size=n2,replace = FALSE)

Folds_knn=matrix(sample_data1,ncol=k)

FirstFoldknn=Folds_knn[,i]

validation_data_knn=updated_data3[FirstFoldknn,]

train_data_knn=updated_data3[-FirstFoldknn,]
```

```
ctrl <- trainControl(method = "cv", number = 10)

model_knn <- train(Price~., data = train_data_knn, method = "knn", trContr
ol = ctrl)

MSE_Train_knn[i]=mean((train_data_knn$Price-(predict(model_knn,newdata=tra
in_data_knn)))^2)

Predictor_4=predict(model_knn,newdata=validation_data_knn)

MSE_val_knn[i]=mean((validation_data_knn$Price-Predictor_4)^2)

}
```





See from the both plot that residuals are kind normally distributed

Performance Metrics:

To checking to performance of each model by using their Mean Squred Error and correlation between actual Price and predicted Price

```
outputmatrix
              MSE.TestData   Cor(y,yhat)
MLR           2.775014e+13   0.6641747
DecisionTree  4.065457e+13   0.4861484
RandomForest  3.898409e+13   0.5205430
KNN           4.145188e+13   0.4418054
```

See from above table that MSE is less and Cor(y,yhat) become high in Multiple linear regression model ,which shows that MLR fitted well in comaparsion of other models.