Group-5 Final Project

Harini, Likitha, Jaswanth, Nanaji

2023-12-13

###Problem Statement. Zillow's Zestimate home valuation has shaken up the U.S. real estate industry since first released 11 years ago. A home is often the largest and most expensive purchase a person makes in his or her lifetime.Ensuring homeowners have a trusted way to monitor this asset is incredibly important. The Zestimate was created to give consumers as much information as possible about homes and the housing market, marking the first-time consumers had access to this type of home value information at no cost. "Zestimates" are estimated home values based on 7.5 million statistical and machine learning models that analyze hundreds of data points on each property. And, by continually improving the median margin of error (from 14% at the onset to 5% today), Zillow has since become established as one of the largest, most trusted marketplaces for real estate information in the U.S. and a leading example of impactful machine learning. This project is the very simplified version of Zillow Prize competition. Zillow Prize was a competition with a one-million-dollar grand prize with the objective to help push the accuracy of the Zestimate even further. Winning algorithms stand to impact the home values of 110M homes across the U.S.

```
#Loading the necessary libraries.
library(stats)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ISLR)
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.3.2
```

```
library(readxl)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
#Importing the datasets.
HP_train<- read.csv("C:/R History/House_Prices.csv")

BA_pred_test <- read.csv("C:/R History/BA-Predict.csv")
```

```
#Printing first few rows of the dataset.
head(HP_train)
```

```
##   LotArea OverallQual YearBuilt YearRemodAdd BsmtFinSF1 FullBath HalfBath
## 1    8450           7      2003         2003        706        2        1
## 2    9600           6      1976         1976        978        2        0
## 3   11250           7      2001         2002        486        2        1
## 4    9550           7      1915         1970        216        1        0
## 5   14260           8      2000         2000        655        2        1
## 6   14115           5      1993         1995        732        1        1
##   BedroomAbvGr TotRmsAbvGrd Fireplaces GarageArea YrSold SalePrice
## 1            3            8          0        548   2008    208500
## 2            3            6          1        460   2007    181500
## 3            3            6          1        608   2008    223500
## 4            3            7          1        642   2006    140000
## 5            4            9          1        836   2008    250000
## 6            1            5          0        480   2009    143000
```

```
head(BA_pred_test)
```

```
##   LotArea OverallQual YearBuilt YearRemodAdd BsmtFinSF1 FullBath HalfBath
## 1    7340           4      1971         1971        322        1        0
## 2    8712           5      1957         2000        860        1        0
## 3    7875           7      2003         2003          0        2        1
## 4   14859           7      2006         2006          0        2        0
## 5    6173           5      1967         1967        599        1        0
## 6    9920           5      1954         1954        354        1        0
##   BedroomAbvGr TotRmsAbvGrd Fireplaces GarageArea YrSold SalePrice
## 1            2            4          0        684   2007    110000
## 2            2            5          0        756   2009    153000
## 3            3            8          1        393   2006    180000
## 4            3            7          1        690   2006    240000
## 5            3            6          0        288   2007    125500
## 6            3            6          0        280   2010    128000
```

```
#Shape of the datasets.
dim(HP_train)
```

```
## [1] 900  13
```

```
dim(BA_pred_test)
```

```
## [1] 90 13
```

```
#Printing the structure of the data.
str(HP_train)
```

```
## 'data.frame':    900 obs. of  13 variables:
##  $ LotArea     : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ OverallQual : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ YearBuilt   : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd: int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ BsmtFinSF1  : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ FullBath    : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath    : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr: int  3 3 3 3 4 1 3 3 2 2 ...
##  $ TotRmsAbvGrd: int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Fireplaces  : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ GarageArea  : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ YrSold      : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SalePrice   : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000
...
```

#Segmantation of the data into the numerical and categorical values is not necessary since all the variables in this dataset are numerical.

```
summary(HP_train)
```

```
##     LotArea        OverallQual      YearBuilt     YearRemodAdd 
##  Min.   :  1491   Min.   : 1.000   Min.   :1880   Min.   :1950  
##  1st Qu.:  7585   1st Qu.: 5.000   1st Qu.:1954   1st Qu.:1968  
##  Median :  9442   Median : 6.000   Median :1973   Median :1994  
##  Mean   : 10795   Mean   : 6.136   Mean   :1971   Mean   :1985  
##  3rd Qu.: 11618   3rd Qu.: 7.000   3rd Qu.:2000   3rd Qu.:2004  
##  Max.   :215245   Max.   :10.000   Max.   :2010   Max.   :2010  
##    BsmtFinSF1        FullBath        HalfBath       BedroomAbvGr 
##  Min.   :   0.0   Min.   :0.000   Min.   :0.0000   Min.   :0.000  
##  1st Qu.:   0.0   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000  
##  Median : 384.0   Median :2.000   Median :0.0000   Median :3.000  
##  Mean   : 446.5   Mean   :1.564   Mean   :0.3856   Mean   :2.843  
##  3rd Qu.: 728.8   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000  
##  Max.   :2260.0   Max.   :3.000   Max.   :2.0000   Max.   :8.000  
##   TotRmsAbvGrd      Fireplaces       GarageArea         YrSold    
##  Min.   : 2.000   Min.   :0.0000   Min.   :   0.0   Min.   :2006  
##  1st Qu.: 5.000   1st Qu.:0.0000   1st Qu.: 336.0   1st Qu.:2007  
##  Median : 6.000   Median :1.0000   Median : 480.0   Median :2008  
##  Mean   : 6.482   Mean   :0.6278   Mean   : 472.6   Mean   :2008  
##  3rd Qu.: 7.000   3rd Qu.:1.0000   3rd Qu.: 576.0   3rd Qu.:2009  
##  Max.   :14.000   Max.   :3.0000   Max.   :1390.0   Max.   :2010  
##    SalePrice     
##  Min.   : 34900  
##  1st Qu.:130000  
##  Median :163000  
##  Mean   :183108  
##  3rd Qu.:216878  
##  Max.   :755000
```

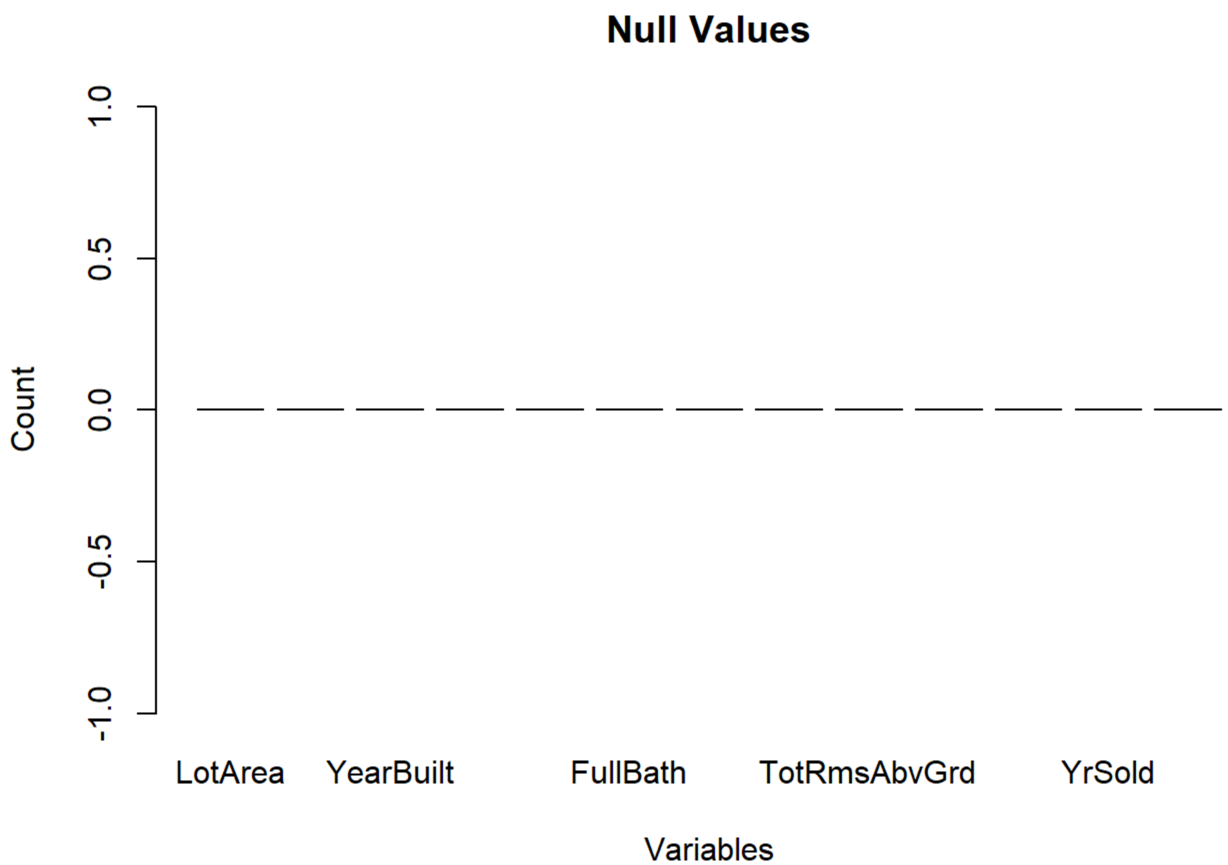
```
#Checking the missing values.
missing_values<- colSums(is.na(HP_train))
print(missing_values)
```

```
##      LotArea  OverallQual    YearBuilt YearRemodAdd   BsmtFinSF1     FullBath 
##            0            0            0            0            0            0 
##     HalfBath BedroomAbvGr TotRmsAbvGrd   Fireplaces   GarageArea       YrSold 
##            0            0            0            0            0            0 
##    SalePrice 
##            0
```

```
cat("From the above data its clear that there are no missing data in the training set")
```

```
## From the above data its clear that there are no missing data in the training set
```

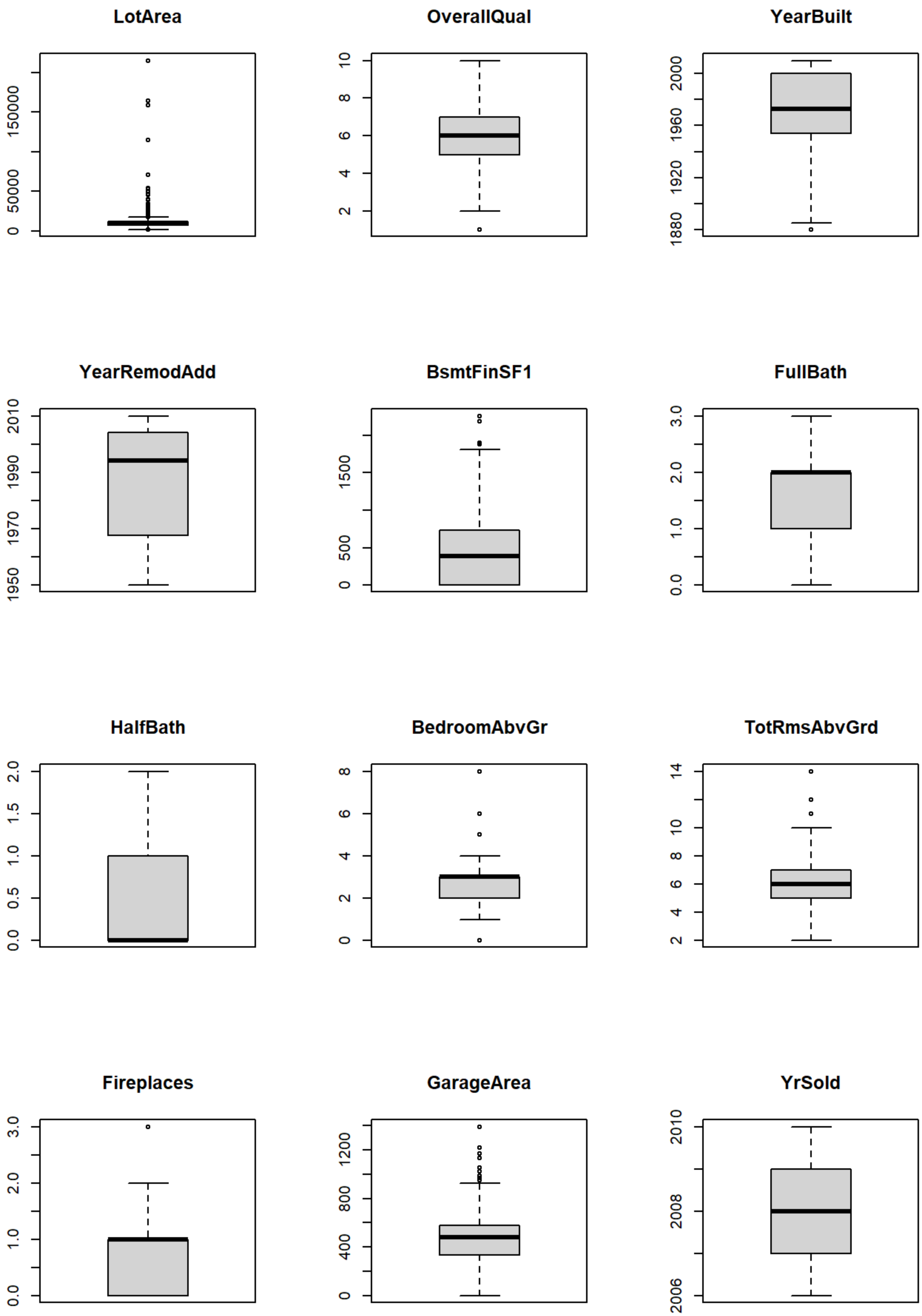
```
#Visualizing the missing values.
barplot(missing_values,main = "Null Values", xlab = "Variables", ylab = "Count")
```

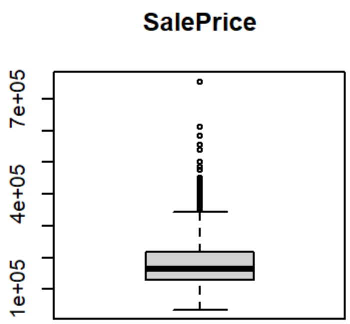



```
#Boxplots to check the outliers.
numeric_vars <- c("LotArea", "OverallQual", "YearBuilt", "YearRemodAdd", "BsmtFinSF1",
                  "FullBath", "HalfBath", "BedroomAbvGr", "TotRmsAbvGrd", "Fireplaces",
                  "GarageArea", "YrSold", "SalePrice")

par(mfrow = c(2, 3))

# Create boxplots for each numerical variable
for (var in numeric_vars) {
  # Check if the variable exists in the dataset before plotting
  if (var %in% colnames(HP_train)) {
    # Plot the boxplot if the variable exists
    boxplot(HP_train[[var]], main = var)
  } else {
    # Print a message if the variable doesn't exist in the dataset
    cat("Variable", var, "does not exist in the dataset.\n")
  }
}
```


LotArea
0
50000
150000
OverallQual
2
4
6
8
10
YearBuilt
1880
1920
1960
2000
YearRemodAdd
1950
1970
1990
2010
BsmtFinSF1
0
500
1500
FullBath
0.0
1.0
2.0
3.0
HalfBath
0.0
0.5
1.0
1.5
2.0
BedroomAbvGr
0
2
4
6
8
TotRmsAbvGrd
2
4
6
8
10
14
Fireplaces
0.0
1.0
2.0
3.0
GarageArea
0
400
800
1200
YrSold
2006
2008
2010



#Variable selection

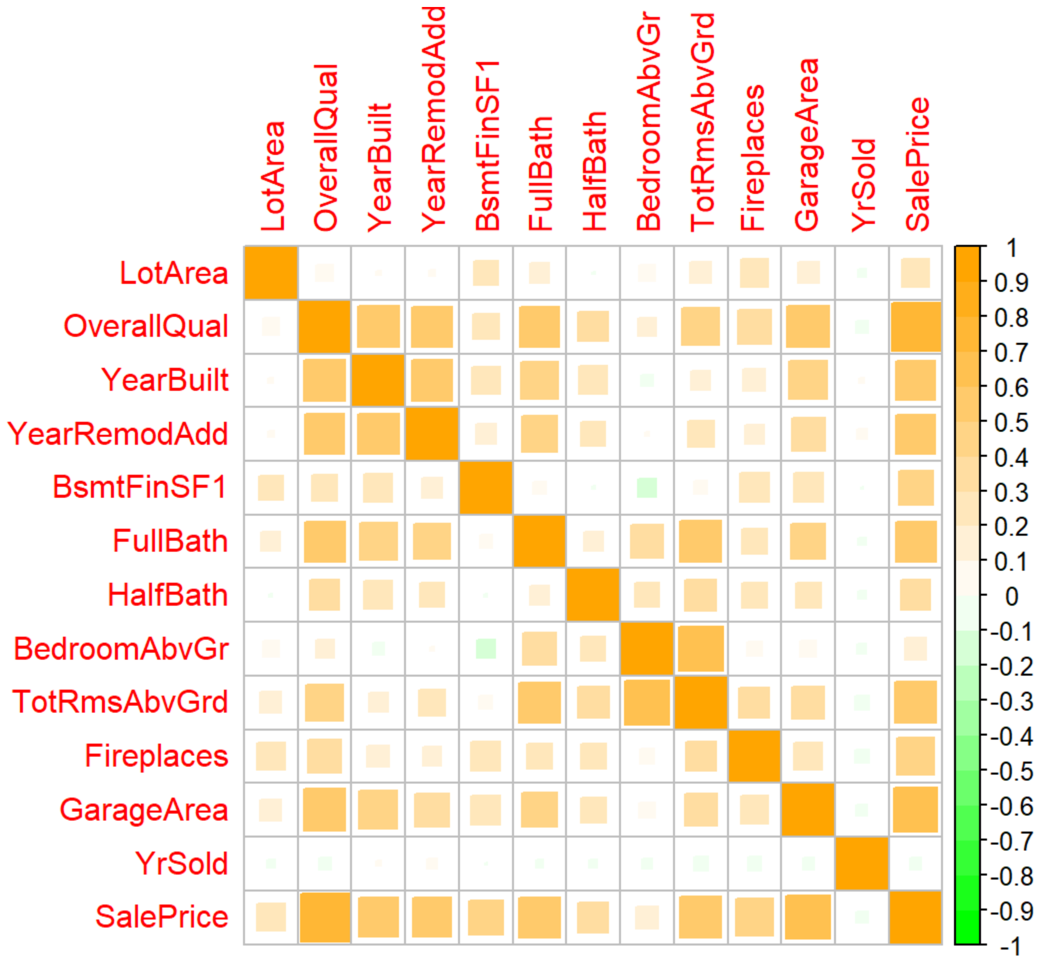
##Coreelation plots and ANOVA can effectively indicate the significance of variables concerning thei impact on the sale price.

```
# Compute correlation matrix
cor_mat <- cor(HP_train)

# Convert correlation matrix to a data frame
cor_df <- reshape2::melt(cor_mat)

# Define a custom color palette (you can choose colors as needed)
Colours <- colorRampPalette(c("green", "white", "orange"))(20)

# Visualize correlations with the specified color palette
corrplot::corrplot(cor_mat, method = "square", col = Colours)
```

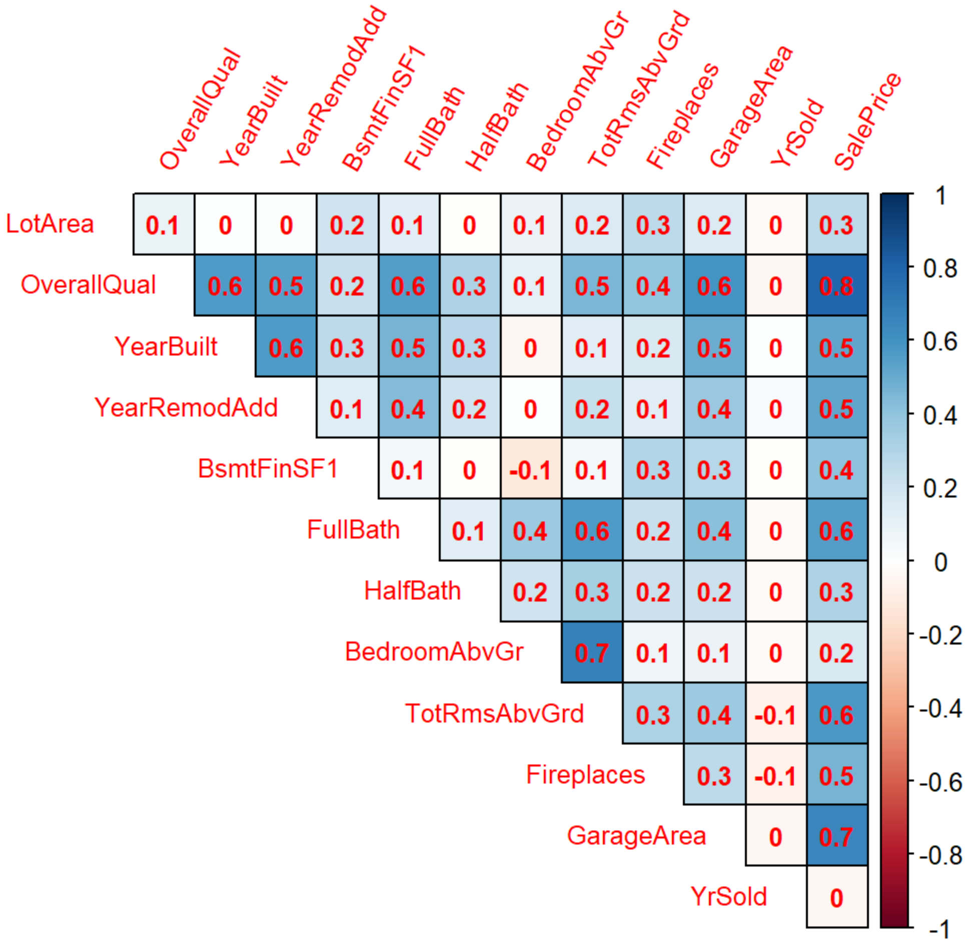



```
# Creating correlation heatmap.
corrplot(cor_mat, method = "color", type = "upper", tl.col = "red",
tl.srt = 60, tl.cex = 0.8, tl.offset = 1, cl.lim = c(-1, 1),
addCoef.col = "red", number.cex = 0.8, number.digits = 1,
diag = FALSE, outline = TRUE)
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt =
## tl.srt, : "cl.lim" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col =
## tl.col, : "cl.lim" is not a graphical parameter
```

```
## Warning in title(title, ...): "cl.lim" is not a graphical parameter
```



###INTERPRETATION: Correlation analysis reveals the relationships and strengths of associations between variables, which aids in understanding how they may influence one another or a specific target variable under investigation. Correlation values quantify the degree and direction of a linear relationship between two variables. They are numbered from -1 to 1, with 1 indicating perfect positive correlation, -1 indicating perfect negative correlation, and 0 indicating no linear relationship between the variables. BedroomAbvGr and YrSold have weak or negligible linear relationships with the objective variable, according to the plots.

###ANOVA

```
#Using ANOVA
anova_model<- aov(SalePrice~.,data = HP_train)
anova_result<- anova(anova_model)
print(anova_result)
```

```
## Analysis of Variance Table
##
## Response: SalePrice
##               Df     Sum Sq    Mean Sq   F value    Pr(>F)    
## LotArea        1 4.2155e+11 4.2155e+11  320.5296 < 2.2e-16 ***
## OverallQual    1 3.6167e+12 3.6167e+12 2750.0049 < 2.2e-16 ***
## YearBuilt      1 6.0695e+10 6.0695e+10   46.1503 2.006e-11 ***
## YearRemodAdd   1 3.9347e+10 3.9347e+10   29.9178 5.864e-08 ***
## BsmtFinSF1     1 2.0995e+11 2.0995e+11  159.6378 < 2.2e-16 ***
## FullBath       1 9.7511e+10 9.7511e+10   74.1437 < 2.2e-16 ***
## HalfBath       1 4.9694e+10 4.9694e+10   37.7854 1.192e-09 ***
## BedroomAbvGr   1 8.3559e+09 8.3559e+09    6.3535   0.01189 *  
## TotRmsAbvGrd   1 2.5570e+11 2.5570e+11  194.4266 < 2.2e-16 ***
## Fireplaces     1 2.2998e+10 2.2998e+10   17.4870 3.180e-05 ***
## GarageArea     1 8.2278e+10 8.2278e+10   62.5608 7.666e-15 ***
## YrSold         1 2.6365e+07 2.6365e+07    0.0200   0.88744    
## Residuals    887 1.1665e+12 1.3152e+09                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

###INTERPRETATION The p-value is a measure that helps determine the significance of the relationship between variables in statistical tests.

Smaller p-value suggests stronger evidence against the null hypothesis, indicating a more significant relationship or effect in the data. optimum p-value ust be less than 0.05.

From the above data BedroomAbvGr and YrSold doesnt have any significance on the response that is SalePrice. Hence the selected variables for the analysis are

1.LotArea 2.OverallQual 3.YearBuilt 4.YearRemodAdd 5.BsmtFinSF1 6.FullBath 7.HalfBath 8.TotRmsAbvGrd 9.Fireplaces 10 GarageArea

A. Build a regression and decision tree model that can accurately predict the price of a house based on several predictors. **1.** Regression Model

```
reg_model<- lm(SalePrice~
                 LotArea+OverallQual+YearBuilt+YearRemodAdd+BsmtFinSF1+FullBath+HalfBath+TotRmsA
bvGrd+Fireplaces+GarageArea,
               data= HP_train)

summary(reg_model)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + OverallQual + YearBuilt +
##     YearRemodAdd + BsmtFinSF1 + FullBath + HalfBath + TotRmsAbvGrd +
##     Fireplaces + GarageArea, data = HP_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -275039  -20705   -3043   15549  345803
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.210e+06  1.623e+05  -7.454 2.15e-13 ***
## LotArea       6.898e-01  1.094e-01   6.304 4.55e-10 ***
## OverallQual   2.411e+04  1.421e+03  16.963  < 2e-16 ***
## YearBuilt     1.237e+02  6.171e+01   2.005   0.0452 *
## YearRemodAdd  4.332e+02  7.881e+01   5.497 5.06e-08 ***
## BsmtFinSF1    3.269e+01  3.096e+00  10.559  < 2e-16 ***
## FullBath      3.967e+03  3.260e+03   1.217   0.2240
## HalfBath      2.278e+03  2.828e+03   0.805   0.4208
## TotRmsAbvGrd  1.157e+04  1.077e+03  10.740  < 2e-16 ***
## Fireplaces    1.068e+04  2.190e+03   4.878 1.27e-06 ***
## GarageArea    6.433e+01  7.802e+00   8.245 5.89e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36790 on 889 degrees of freedom
## Multiple R-squared:  0.8005, Adjusted R-squared:  0.7983
## F-statistic: 356.8 on 10 and 889 DF,  p-value: < 2.2e-16
```

In a regression model, high p-values may suggest that those components are not statistically significant in predicting the target variable.

So, take the necessary factors into account and rebuild the model. Based on the statistics presented above, the significant variables are as follows: 1.LotArea 2.OverallQual 3.YearBuilt 4.YearRemodAdd 5.BsmtFinSF1 6.TotRmsAbvGrd 7.Fireplaces 8.GarageArea

```
reg_model_rev<- lm(SalePrice~
                 LotArea+OverallQual+YearBuilt+YearRemodAdd+BsmtFinSF1+TotRmsAbvGrd+Fireplaces+G
arageArea,
               data= HP_train)

summary(reg_model_rev)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + OverallQual + YearBuilt +
##     YearRemodAdd + BsmtFinSF1 + TotRmsAbvGrd + Fireplaces + GarageArea,
##     data = HP_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -272631  -20745   -3636   15512  348816
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.289e+06  1.510e+05  -8.534  < 2e-16 ***
## LotArea       6.971e-01  1.088e-01   6.406 2.41e-10 ***
## OverallQual   2.439e+04  1.404e+03  17.366  < 2e-16 ***
## YearBuilt     1.544e+02  5.712e+01   2.704  0.00699 **
## YearRemodAdd  4.426e+02  7.839e+01   5.645 2.22e-08 ***
## BsmtFinSF1    3.187e+01  3.030e+00  10.518  < 2e-16 ***
## TotRmsAbvGrd  1.235e+04  9.008e+02  13.708  < 2e-16 ***
## Fireplaces    1.085e+04  2.172e+03   4.995 7.07e-07 ***
## GarageArea    6.433e+01  7.800e+00   8.247 5.79e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36780 on 891 degrees of freedom
## Multiple R-squared:  0.8002, Adjusted R-squared:  0.7984
## F-statistic: 445.9 on 8 and 891 DF,  p-value: < 2.2e-16
```

```
#Prediction model with the test data.
prediction_reg <- predict(reg_model_rev, newdata = BA_pred_test, type = 'response')

#Evaluation metrics.
r_squared <- cor(BA_pred_test$SalePrice, prediction_reg)^2
cat("Linear Regression R-squared:\n", r_squared)
```

```
## Linear Regression R-squared:
##  0.8232827
```

```
rmse <- sqrt(mean((prediction_reg - BA_pred_test$SalePrice)^2))
cat("\nLinear Regression RMSE:\n",rmse)
```

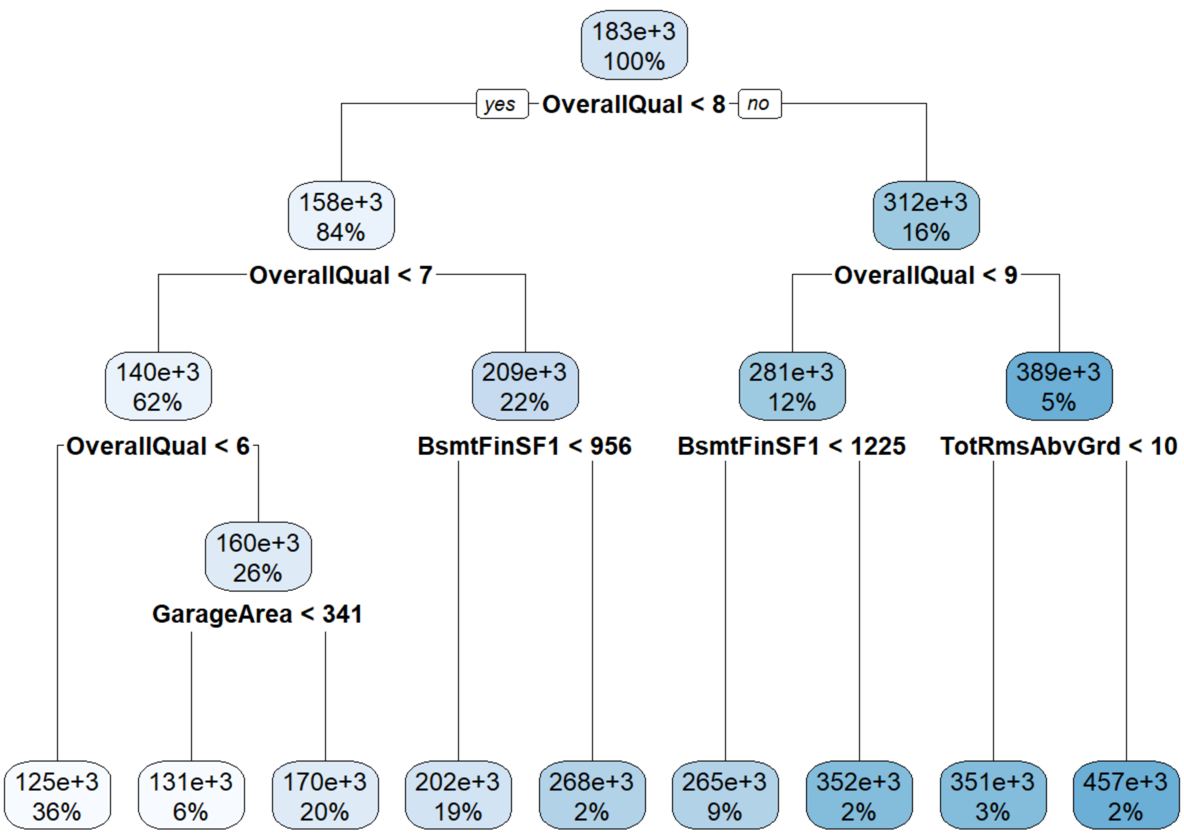
```
##
## Linear Regression RMSE:
##  28237.95
```

###INTERPRETATION The R2 is a statistic that shows how much variability in the dependent variable can be explained by the independent variables using regression models. R-squared refers to the measure of how well the predicted values correspond with the real data values and the degree of accuracy of a regression model.

In this instance, the number is 0.823, meaning that the effects of 82.3% variance from the independent variables in the regression model explain the variance of the dependent response variable. This shows how close this model to the real data is and explains about 40% of changes observed for the dependent variable.

2. Decision Tree

```
Dc_Tr<- rpart(SalePrice~
                   LotArea + OverallQual + YearBuilt +
    YearRemodAdd + BsmtFinSF1  + TotRmsAbvGrd +
    Fireplaces + GarageArea,
    data = HP_train,
    method = 'anova',
    control = rpart.control(ninsplit=60),maxdepth = 3)
rpart.plot(Dc_Tr)
```




```
pred_DT<- predict(Dc_Tr, newdata = BA_pred_test)

#evaluation metrics
DT_r_squared <- cor(pred_DT, BA_pred_test$SalePrice)^2
cat("Decision Tree R-squared:\n", DT_r_squared)
```

```
## Decision Tree R-squared:
##  0.6684661
```

```
DT_rmse <- RMSE(pred_DT, BA_pred_test$SalePrice)
cat("\nDecision Tree rsme:\n", DT_rmse)
```

```
##
## Decision Tree rsme:
##  35864.1
```

B. Using classification to model OverallQual (rating 7 and above consider as class 1, otherwise class zero). **3.** Classification Model

```
classification_Model <- glm(as.factor(ifelse(OverallQual >= 7, 1, 0)) ~ ., data = HP_train, fami
ly = 'binomial')
summary(classification_Model)
```

```
##
## Call:
## glm(formula = as.factor(ifelse(OverallQual >= 7, 1, 0)) ~ .,
##     family = "binomial", data = HP_train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   8.655e+01  1.808e+02   0.479 0.632224
## LotArea      -3.361e-05  9.226e-06  -3.643 0.000269 ***
## YearBuilt     1.068e-02  6.195e-03   1.724 0.084665 .
## YearRemodAdd  1.773e-02  9.262e-03   1.914 0.055561 .
## BsmtFinSF1   -1.910e-03  3.451e-04  -5.535 3.11e-08 ***
## FullBath      3.759e-01  3.315e-01   1.134 0.256801
## HalfBath     -1.261e-01  2.593e-01  -0.486 0.626724
## BedroomAbvGr -6.622e-01  2.564e-01  -2.583 0.009795 **
## TotRmsAbvGrd  2.109e-01  1.458e-01   1.447 0.147952
## Fireplaces    1.709e-01  2.081e-01   0.821 0.411448
## GarageArea    1.958e-03  1.028e-03   1.905 0.056793 .
## YrSold       -7.529e-02  9.043e-02  -0.833 0.405071
## SalePrice     4.298e-05  5.097e-06   8.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1195.32  on 899  degrees of freedom
## Residual deviance:  471.83  on 887  degrees of freedom
## AIC: 497.83
##
## Number of Fisher Scoring iterations: 7
```

```
# Making predictions on the test data.
prob <- predict(classification_Model, newdata = BA_pred_test, type = "response")

# Assigning classes based on a threshold.
class_prediction <- as.factor(ifelse(prob >= 0.5, 1, 0))

# Creating confusion matrix on test data.
confusionMatrix(class_prediction, as.factor(ifelse(BA_pred_test$OverallQual >= 7, 1, 0)), positi
ve = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 47  7
##          1  8 28
##
##                Accuracy : 0.8333
##                  95% CI : (0.74, 0.9036)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 4.19e-06
##
##                   Kappa : 0.6512
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8000
##             Specificity : 0.8545
##          Pos Pred Value : 0.7778
##          Neg Pred Value : 0.8704
##              Prevalence : 0.3889
##          Detection Rate : 0.3111
##    Detection Prevalence : 0.4000
##       Balanced Accuracy : 0.8273
##
##        'Positive' Class : 1
##
```

###INTERPRETATION From the above analysis accuracy is 0.8333 Sensitivity : 0.8000 Specificity : 0.8545

###ANALYSIS & COMPARISION OF THREE MODELS. 1.Regression Model Linear Regression R-squared:0.8232827 Linear Regression RMSE:28237.95

2.Decision Tree Decision Tree R-squared:0.6684661 Decision Tree rsme:35864.1

3.Classification analysis Accuracy : 0.8333 Sensitivity : 0.8000 Specificity : 0.8545

The regression model has the highest R-squared value (0.823) when compared to the decision tree model (R-squared: 0.668), indicating stronger explanatory power. The regression model, on the other hand, has a lower error (RMSE: 28237.95) than the decision tree model. The classification analysis achieved an accuracy of 83.33%, showing the model’s ability to correctly classify instances. It also demonstrates good sensitivity (80.00%) and specificity (85.45%), indicating its capability to accurately identify positive and negative cases. Finally, the regression model has the most explanatory power, although the classification analysis has good accuracy and a decent mix of sensitivity and specificity. Despite its weaker performance measurements, the decision tree approach may nonetheless provide insights into nonlinear relationships in data.