Business Analytics Final Project

2023-12-08

Question

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.

Build a regression and decision tree model that can accurately predict the price of a house based on several predictors (you select appropriate features)

Loading the dataset and viewing the first few rows.

```
data_frame = read.csv("/Users/gauravkudeshia/Downloads/House_Prices.csv")
head(data_frame)
```

```
##   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
```

Finding the missing values in the data set, displaying the summary of the structure, and lastly displaying the summary of the data set.

```
miss_val <- colSums(is.na(data_frame))
miss_val
```

```
##      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
```

```
sum(is.na(data_frame))
```

```
## [1] 0
```

```
str(data_frame)
```

```
## '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 1
29900 118000 ...
```

```
summary(data_frame)
```

```
##     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
```

Calculating the correlation coefficient between the “SalePrice” variable and each of the other variables.

```
correlation.coefficient1 <- cor(data_frame$SalePrice, data_frame$LotArea)
cat("Correlation coefficient between Sale Price and Lot Area:", correlation.coefficient1, "\n")
```

```
## Correlation coefficient between Sale Price and Lot Area: 0.2643725
```

```
correlation.coefficient2 <- cor(data_frame$SalePrice, data_frame$OverallQual)
cat("Correlation coefficient between Sale Price and Overall Quality:", correlation.coefficient2, "\n")
```

```
## Correlation coefficient between Sale Price and Overall Quality: 0.7962135
```

```
correlation.coefficient3 <- cor(data_frame$SalePrice, data_frame$YearBuilt)
cat("Correlation coefficient between Sale Price and Build year:", correlation.coefficient3, "\n")
```

```
## Correlation coefficient between Sale Price and Build year: 0.5266341
```

```
correlation.coefficient4 <- cor(data_frame$SalePrice, data_frame$YearRemodAdd)
cat("Correlation coefficient between Sale Price and Year Remodeled:", correlation.
coefficient4, "\n")
```

```
## Correlation coefficient between Sale Price and Year Remodeled: 0.5221773
```

```
correlation.coefficient5 <- cor(data_frame$SalePrice, data_frame$BsmtFinSF1)
cat("Correlation coefficient between Sale Price and Finished Square feet:", correl
ation.coefficient5, "\n")
```

```
## Correlation coefficient between Sale Price and Finished Square feet: 0.4046632
```

```
correlation.coefficient6 <- cor(data_frame$SalePrice, data_frame$GarageArea)
cat("Correlation coefficient between Sale Price and Garage Area:", correlation.coe
fficient6, "\n")
```

```
## Correlation coefficient between Sale Price and Garage Area: 0.656042
```

```
correlation.coefficient7 <- cor(data_frame$SalePrice, data_frame$Fireplaces)
cat("Correlation coefficient between Sale Price and No. of Fireplaces:", correlati
on.coefficient7, "\n")
```

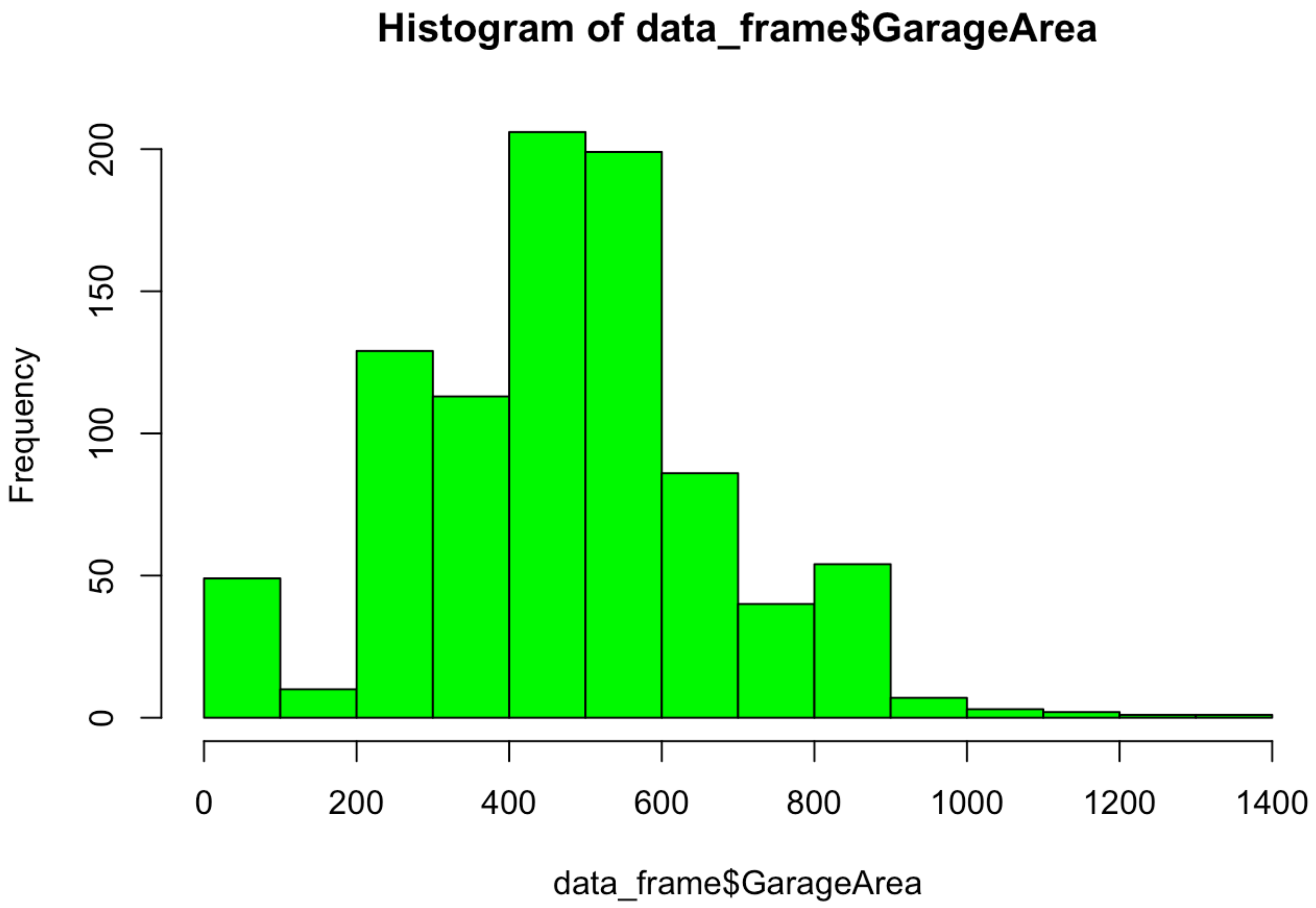
```
## Correlation coefficient between Sale Price and No. of Fireplaces: 0.4686277
```

```
correlation.coefficient8 <- cor(data_frame$SalePrice, data_frame$TotRmsAbvGrd)
cat("Correlation coefficient between Sale Price and Total rooms above ground:", co
rrelation.coefficient8, "\n")
```

```
## Correlation coefficient between Sale Price and Total rooms above ground: 0.5773
581
```

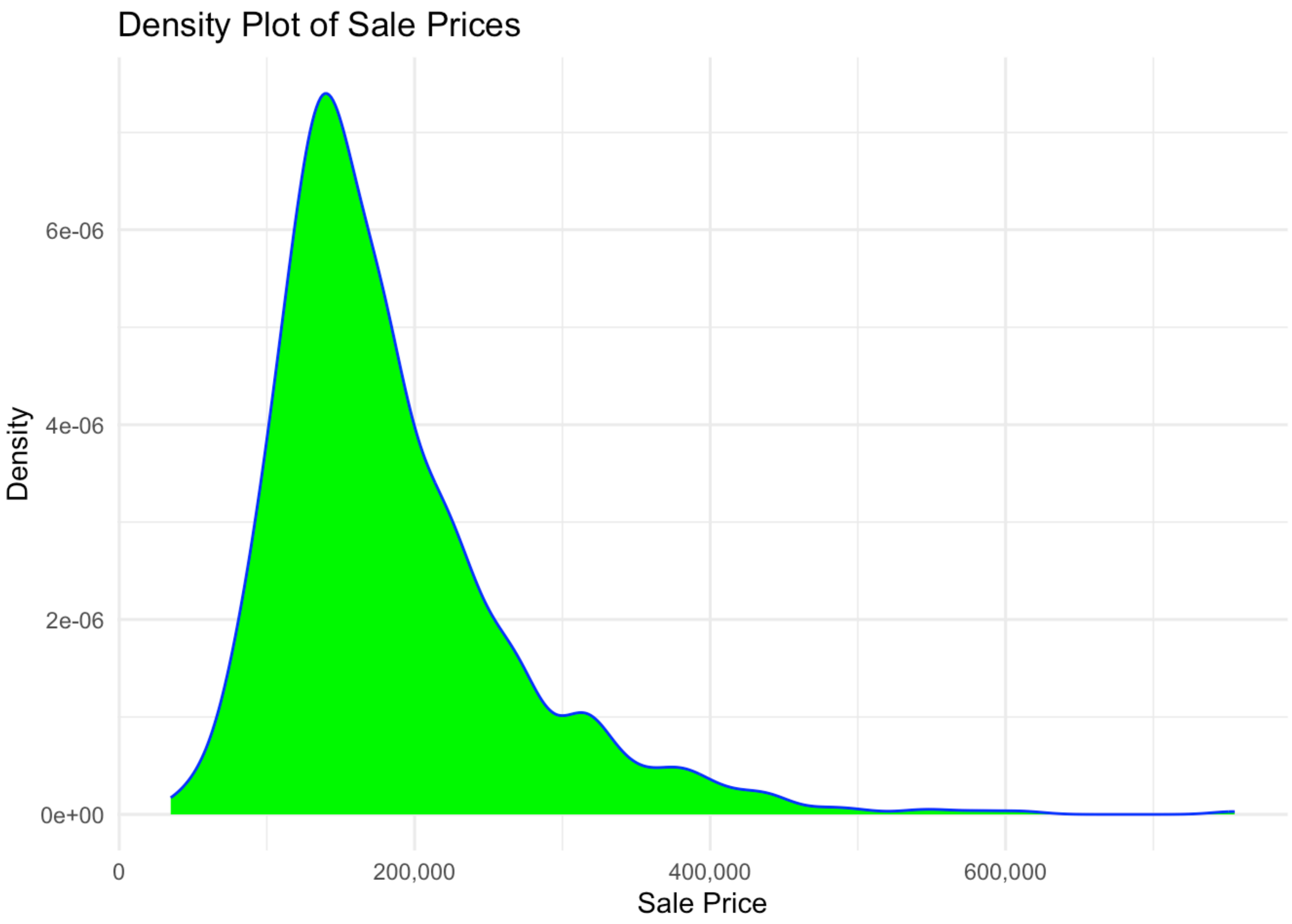
Now we will do some data exploration.

```
hist(data_frame$GarageArea, col = 'green')
```



Creating a density plot of sale prices

```
suppressMessages(library(ggplot2))

ggplot(data_frame, aes(x = SalePrice)) +
  geom_density(fill = "green", color = "blue") +
  labs(title = "Density Plot of Sale Prices", x = "Sale Price", y = "Density") + t
heme_minimal() + scale_x_continuous(labels = scales::comma)
```



Generating a bar plot showing the average sale price for each level of overall quality in our dataset.

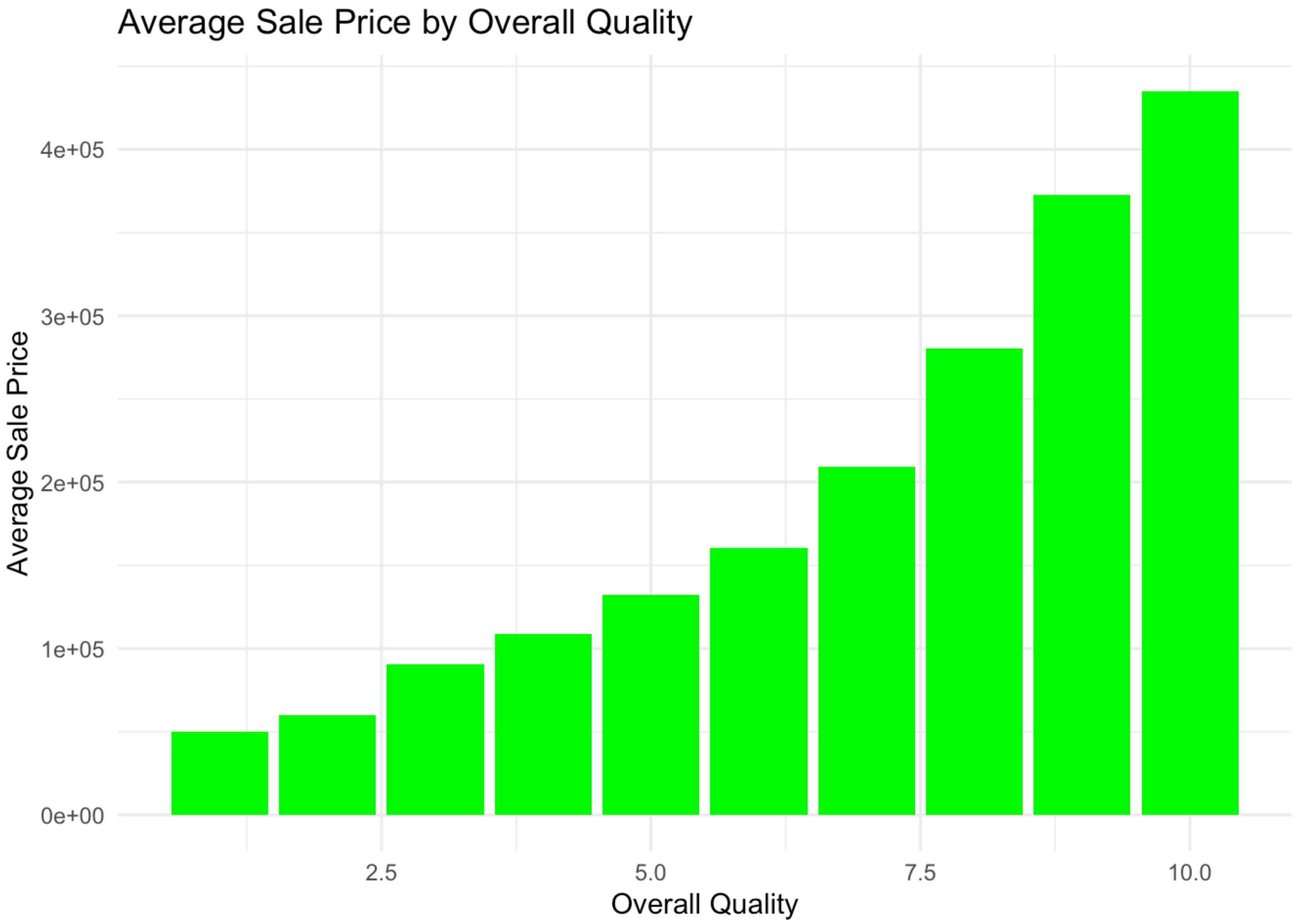
```
library(dplyr)
```

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

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

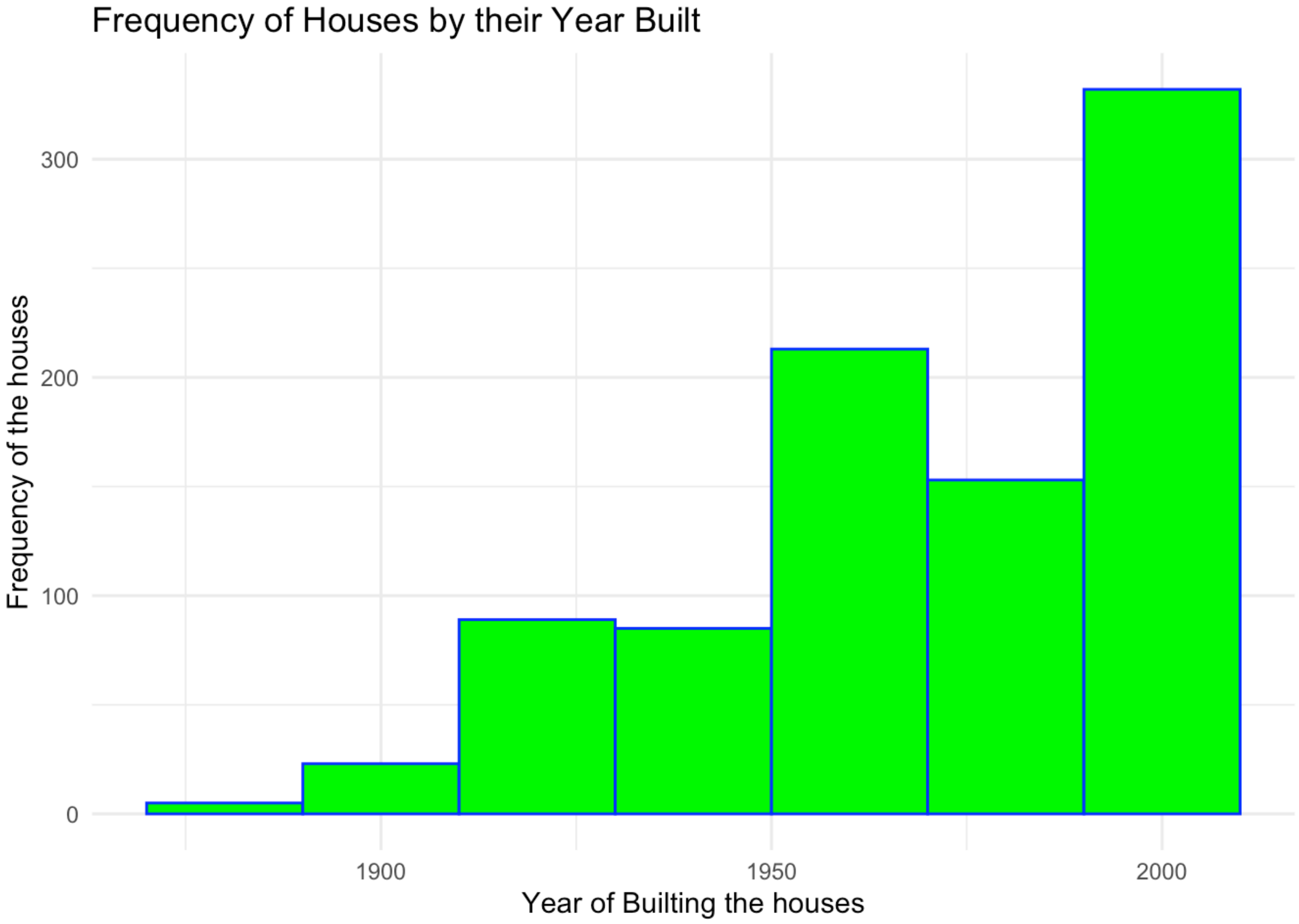
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
avgp <- data_frame %>% group_by(OverallQual) %>% summarize(avg_SalePrice = mean(Sa
lePrice))

ggplot(avgp, aes(x = OverallQual, y = avg_SalePrice)) +
  geom_bar(stat = "identity",fill = "green") +
  labs(title = "Average Sale Price by Overall Quality", x = "Overall Quality", y =
"Average Sale Price") + theme_minimal() + scale_x_continuous(labels = scales::comm
a)
```



Generating a histogram illustrating the distribution of houses based on their year of construction

```
ggplot(data_frame, aes(x = YearBuilt)) +
  geom_histogram(binwidth = 20, fill = "green", color = "blue") +
  labs(title = "Frequency of Houses by their Year Built",
       x = "Year of Builting the houses",
       y = "Frequency of the houses") +
  theme_minimal()
```



Regression

```
reg_mod = lm(SalePrice ~., data = data_frame)
summary(reg_mod)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = data_frame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -286336  -20369   -2819   16607  349565
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.378e+06  1.848e+06  -0.746   0.4561
## LotArea       7.109e-01  1.079e-01   6.586 7.73e-11 ***
## OverallQual   2.299e+04  1.418e+03  16.209  < 2e-16 ***
## YearBuilt     1.295e+02  6.085e+01   2.128   0.0336 *
## YearRemodAdd  3.855e+02  7.836e+01   4.920 1.03e-06 ***
## BsmtFinSF1    3.101e+01  3.070e+00  10.103  < 2e-16 ***
## FullBath      5.883e+03  3.235e+03   1.818   0.0694 .
## HalfBath      3.055e+03  2.792e+03   1.094   0.2743
## BedroomAbvGr -1.135e+04  2.157e+03  -5.264 1.77e-07 ***
## TotRmsAbvGrd  1.585e+04  1.338e+03  11.844  < 2e-16 ***
## Fireplaces    9.581e+03  2.170e+03   4.415 1.13e-05 ***
## GarageArea    6.106e+01  7.718e+00   7.911 7.60e-15 ***
## YrSold        1.305e+02  9.216e+02   0.142   0.8874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36270 on 887 degrees of freedom
## Multiple R-squared:  0.8066, Adjusted R-squared:  0.804
## F-statistic: 308.3 on 12 and 887 DF,  p-value: < 2.2e-16
```

As we know that low p-values (< 0.05) will make one variable as a statically insignificant. After selecting the significant variables we will again run our model.

```
reg_mod_1 = lm(SalePrice ~ LotArea+ OverallQual + YearRemodAdd + BsmtFinSF1 + Bedr
oomAbvGr+ TotRmsAbvGrd + Fireplaces + GarageArea , data = data_frame)
summary(reg_mod_1)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + OverallQual + YearRemodAdd +
##     BsmtFinSF1 + BedroomAbvGr + TotRmsAbvGrd + Fireplaces + GarageArea,
##     data = data_frame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -284015  -19879   -2606   17052  351097
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.075e+06  1.407e+05  -7.640 5.61e-14 ***
## LotArea       6.986e-01  1.077e-01   6.487 1.45e-10 ***
## OverallQual   2.466e+04  1.345e+03  18.335  < 2e-16 ***
## YearRemodAdd  4.903e+02  7.254e+01   6.759 2.52e-11 ***
## BsmtFinSF1    3.130e+01  2.993e+00  10.457  < 2e-16 ***
## BedroomAbvGr -1.044e+04  2.148e+03  -4.862 1.37e-06 ***
## TotRmsAbvGrd  1.607e+04  1.230e+03  13.063  < 2e-16 ***
## Fireplaces    9.655e+03  2.161e+03   4.468 8.91e-06 ***
## GarageArea    6.675e+01  7.538e+00   8.855  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36450 on 891 degrees of freedom
## Multiple R-squared:  0.8037, Adjusted R-squared:  0.802
## F-statistic: 456.1 on 8 and 891 DF,  p-value: < 2.2e-16
```

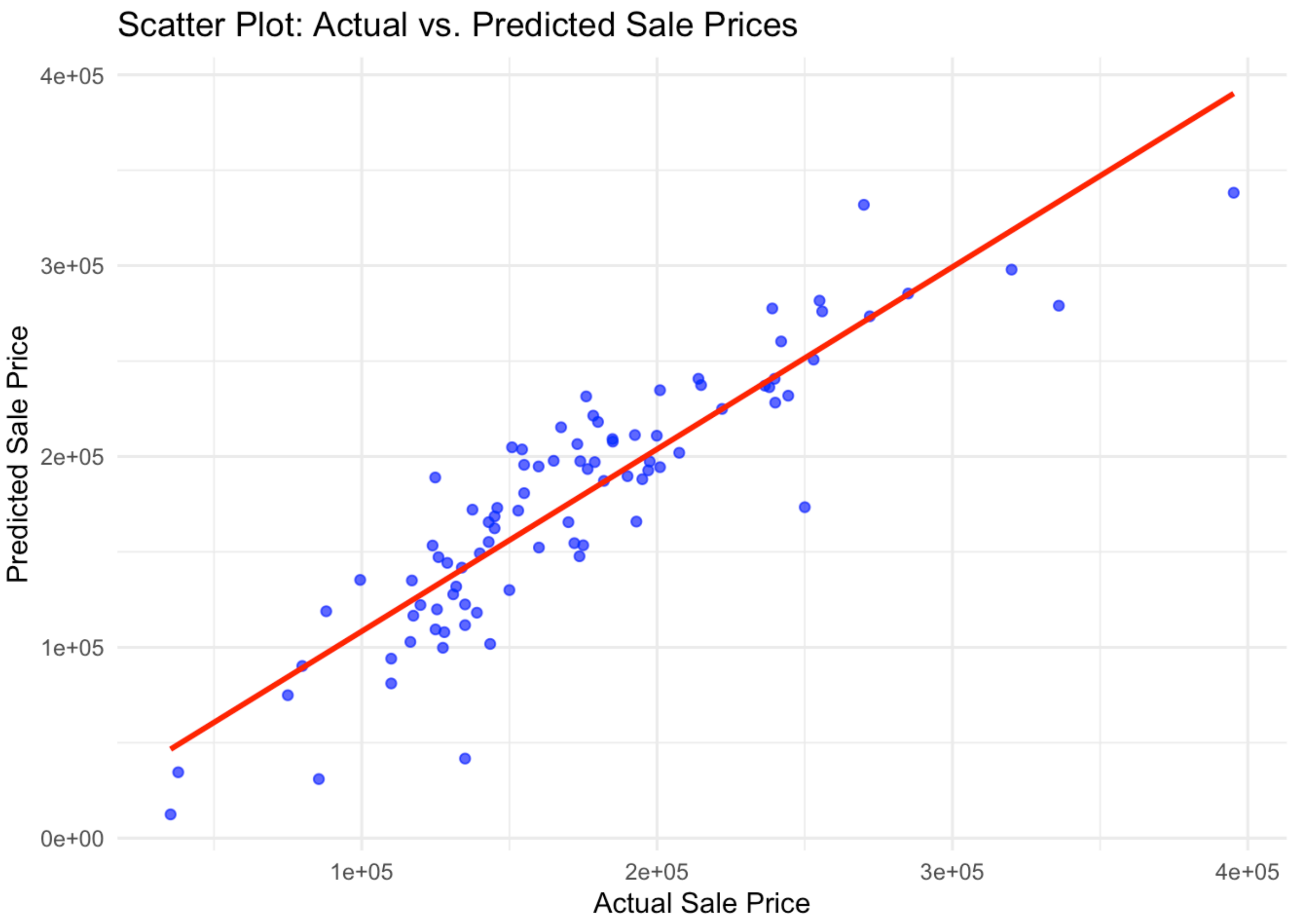
Now we will be making predictions using a trained regression model on new data and examining how well the model performs by comparing actual and predicted values.

```
# Load the readxl library for reading Excel files
library(readxl)

# Read the Excel file into a data frame
data_frame_predict = read_excel("/Users/gauravkudeshia/Desktop/Rhistory Business A
nalytics/BA-Predict-2.xlsx")

# Use the predict function to make predictions based on the linear regression mode
l (reg_mod)
Predicted_SalePrice = predict(reg_mod_1, newdata = data_frame_predict)

# Create a data frame with Actual_Price and Predicted_Price columns
SalesPrice_table = data.frame(Actual_Price = data_frame_predict$SalePrice, Predict
ed_Price = Predicted_SalePrice)

# Display the first few rows of the table
head(SalesPrice_table)
```

```
##   Actual_Price Predicted_Price
## 1       110000        94075.22
## 2       153000       171627.24
## 3       180000       218101.94
## 4       240000       228209.70
## 5       125500       119889.17
## 6       128000       107930.68
```

```
# Assuming 'SalesPrice_table' is your data frame
library(ggplot2)

# Scatter plot between Actual_Price and Predicted_Price with a regression line
ggplot(SalesPrice_table, aes(x = Actual_Price, y = Predicted_Price)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +  # Add a regression line
  labs(title = "Scatter Plot: Actual vs. Predicted Sale Prices",
       x = "Actual Sale Price",
       y = "Predicted Sale Price") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



To compare models, we use criteria such as R2_adj, RMSE, and MAE

```
rsqu <- summary(reg_mod)$adj.r.squared
rsqu
```

```
## [1] 0.8039693
```

```
rmse <- sqrt(mean((data_frame_predict$SalePrice - Predicted_SalePrice)^2))
rmse
```

```
## [1] 29381.9
```

```
mae <- mean(abs(data_frame_predict$SalePrice - Predicted_SalePrice))
mae
```

```
## [1] 22715.48
```

Decision Tree

Building and summarizing a decision tree

```
# Loading the rpart library for decision tree modeling
library(rpart)

# Building a decision tree regression model
DecisionTree = rpart(SalePrice ~.,data = data_frame, method = 'anova')

# Displaying a summary of the decision tree model
summary(DecisionTree)
```

```
## Call:
## rpart(formula = SalePrice ~ ., data = data_frame, method = "anova")
##   n= 900
##
##             CP nsplit rel error    xerror       xstd
## 1  0.47785266      0 1.0000000 1.0007316 0.08988106
## 2  0.11551089      1 0.5221473 0.5248720 0.04665072
## 3  0.05814644      2 0.4066365 0.4096898 0.04421649
## 4  0.02917450      3 0.3484900 0.3631822 0.03379217
## 5  0.01958114      4 0.3193155 0.3381105 0.03360736
## 6  0.01801227      5 0.2997344 0.3287574 0.03552581
## 7  0.01429023      6 0.2817221 0.3338756 0.03828993
## 8  0.01200980      7 0.2674319 0.3223430 0.03700296
## 9  0.01183168      8 0.2554221 0.3233108 0.03739713
## 10 0.01000000      9 0.2435904 0.3177275 0.03733628
## 
## Variable importance
##  OverallQual   GarageArea    YearBuilt   BsmtFinSF1 YearRemodAdd TotRmsAbvGrd 
##           52           13           10           10            4            4 
##     FullBath BedroomAbvGr      LotArea   Fireplaces 
##            3            1            1            1 
## 
## Node number 1: 900 observations,    complexity param=0.4778527
##   mean=183107.9, MSE=6.701496e+09 
##   left son=2 (754 obs) right son=3 (146 obs)
##   Primary splits:
##       OverallQual  < 7.5     to the left,  improve=0.4778527, (0 missing)
##       YearBuilt    < 1984.5  to the left,  improve=0.3410164, (0 missing)
##       GarageArea   < 675.5   to the left,  improve=0.3352460, (0 missing)
##       FullBath     < 1.5     to the left,  improve=0.2765066, (0 missing)
##       YearRemodAdd < 1983.5  to the left,  improve=0.2410551, (0 missing)
##   Surrogate splits:
##       GarageArea   < 679     to the left,  agree=0.891, adj=0.329, (0 split)
##       YearBuilt    < 2005.5  to the left,  agree=0.863, adj=0.158, (0 split)
##       BsmtFinSF1   < 1336    to the left,  agree=0.860, adj=0.137, (0 split)
##       YearRemodAdd < 2007.5  to the left,  agree=0.850, adj=0.075, (0 split)
##       TotRmsAbvGrd < 9.5     to the left,  agree=0.844, adj=0.041, (0 split)
## 
## Node number 2: 754 observations,    complexity param=0.1155109
##   mean=158206.5, MSE=2.548301e+09 
##   left son=4 (558 obs) right son=5 (196 obs)
##   Primary splits:
##       OverallQual  < 6.5     to the left,  improve=0.3625894, (0 missing)
##       FullBath     < 1.5     to the left,  improve=0.3232482, (0 missing)
##       YearBuilt    < 1984.5  to the left,  improve=0.2933600, (0 missing)
##       GarageArea   < 387     to the left,  improve=0.2526931, (0 missing)
##       YearRemodAdd < 1983.5  to the left,  improve=0.2157413, (0 missing)
##   Surrogate splits:
##       YearBuilt    < 1985.5  to the left,  agree=0.826, adj=0.332, (0 split)
##       YearRemodAdd < 2002.5  to the left,  agree=0.765, adj=0.097, (0 split)
##       GarageArea   < 625.5   to the left,  agree=0.760, adj=0.077, (0 split)
##       BsmtFinSF1   < 1333    to the left,  agree=0.743, adj=0.010, (0 split)
##       LotArea      < 61994   to the left,  agree=0.741, adj=0.005, (0 split)
## 
## Node number 3: 146 observations,    complexity param=0.05814644
##   mean=311708.3, MSE=8.409812e+09 
##   left son=6 (104 obs) right son=7 (42 obs)
##   Primary splits:
##       OverallQual  < 8.5     to the left,  improve=0.2856263, (0 missing)
##       LotArea      < 12094.5 to the left,  improve=0.2497850, (0 missing)
##       TotRmsAbvGrd < 9.5     to the left,  improve=0.2481846, (0 missing)
##       BsmtFinSF1   < 1224.5  to the left,  improve=0.2341417, (0 missing)
##       GarageArea   < 663     to the left,  improve=0.1742764, (0 missing)
##   Surrogate splits:
```

```
##       BsmtFinSF1   < 1744    to the left,  agree=0.747, adj=0.119, (0 split)
##       TotRmsAbvGrd < 10.5    to the left,  agree=0.747, adj=0.119, (0 split)
##       YearBuilt    < 2007.5  to the left,  agree=0.740, adj=0.095, (0 split)
##       LotArea      < 12811.5 to the left,  agree=0.733, adj=0.071, (0 split)
##       YearRemodAdd < 2007.5  to the left,  agree=0.733, adj=0.071, (0 split)
##
## Node number 4: 558 observations,    complexity param=0.0291745
##   mean=140191.1, MSE=1.416245e+09
##   left son=8 (372 obs) right son=9 (186 obs)
##   Primary splits:
##       FullBath    < 1.5     to the left,  improve=0.2226614, (0 missing)
##       OverallQual < 5.5     to the left,  improve=0.2102913, (0 missing)
##       GarageArea  < 387     to the left,  improve=0.1995198, (0 missing)
##       Fireplaces  < 0.5     to the left,  improve=0.1972087, (0 missing)
##       LotArea     < 9100.5  to the left,  improve=0.1645839, (0 missing)
##   Surrogate splits:
##       TotRmsAbvGrd < 6.5     to the left,  agree=0.781, adj=0.344, (0 split)
##       YearBuilt    < 1983.5  to the left,  agree=0.737, adj=0.210, (0 split)
##       BedroomAbvGr < 3.5     to the left,  agree=0.728, adj=0.183, (0 split)
##       OverallQual  < 5.5     to the left,  agree=0.683, adj=0.048, (0 split)
##       BsmtFinSF1   < 1106.5  to the left,  agree=0.683, adj=0.048, (0 split)
##
## Node number 5: 196 observations,    complexity param=0.01429023
##   mean=209495.3, MSE=2.216673e+09
##   left son=10 (174 obs) right son=11 (22 obs)
##   Primary splits:
##       BsmtFinSF1   < 955.5   to the left,  improve=0.19837900, (0 missing)
##       LotArea      < 9701.5  to the left,  improve=0.18976810, (0 missing)
##       TotRmsAbvGrd < 7.5     to the left,  improve=0.18165830, (0 missing)
##       GarageArea   < 785     to the left,  improve=0.17263200, (0 missing)
##       Fireplaces   < 0.5     to the left,  improve=0.08466878, (0 missing)
##   Surrogate splits:
##       LotArea      < 92955   to the left,  agree=0.898, adj=0.091, (0 split)
##       BedroomAbvGr < 1.5     to the right, agree=0.893, adj=0.045, (0 split)
##
## Node number 6: 104 observations,    complexity param=0.01958114
##   mean=280562.4, MSE=4.17479e+09
##   left son=12 (85 obs) right son=13 (19 obs)
##   Primary splits:
##       BsmtFinSF1   < 1224.5  to the left,  improve=0.2720096, (0 missing)
##       GarageArea   < 536     to the left,  improve=0.2187127, (0 missing)
##       LotArea      < 11435.5 to the left,  improve=0.1910548, (0 missing)
##       TotRmsAbvGrd < 9.5     to the left,  improve=0.1194041, (0 missing)
##       BedroomAbvGr < 3.5     to the left,  improve=0.1085876, (0 missing)
##   Surrogate splits:
##       LotArea < 18782.5 to the left,  agree=0.837, adj=0.105, (0 split)
##
## Node number 7: 42 observations,    complexity param=0.01801227
##   mean=388831.3, MSE=1.05465e+10
##   left son=14 (27 obs) right son=15 (15 obs)
##   Primary splits:
##       TotRmsAbvGrd < 9.5     to the left,  improve=0.2452590, (0 missing)
##       Fireplaces   < 1.5     to the left,  improve=0.2196572, (0 missing)
##       GarageArea   < 797     to the left,  improve=0.1844068, (0 missing)
##       BsmtFinSF1   < 1277    to the left,  improve=0.1819313, (0 missing)
##       LotArea      < 12072   to the left,  improve=0.1793774, (0 missing)
##   Surrogate splits:
##       BedroomAbvGr < 3.5     to the left,  agree=0.810, adj=0.467, (0 split)
##       Fireplaces   < 1.5     to the left,  agree=0.786, adj=0.400, (0 split)
##       FullBath     < 2.5     to the left,  agree=0.738, adj=0.267, (0 split)
##       LotArea      < 18927   to the left,  agree=0.714, adj=0.200, (0 split)
##       HalfBath     < 0.5     to the left,  agree=0.714, adj=0.200, (0 split)
##
## Node number 8: 372 observations,    complexity param=0.0120098
##   mean=127634.4, MSE=9.157591e+08
##   left son=16 (120 obs) right son=17 (252 obs)
##   Primary splits:
##       BsmtFinSF1 < 169     to the left,  improve=0.2126306, (0 missing)
##       GarageArea < 213     to the left,  improve=0.1896401, (0 missing)
##       YearBuilt  < 1952.5  to the left,  improve=0.1737735, (0 missing)
##       Fireplaces < 0.5     to the left,  improve=0.1733798, (0 missing)
##       LotArea    < 9100.5  to the left,  improve=0.1647429, (0 missing)
##   Surrogate splits:
##       YearBuilt    < 1938.5  to the left,  agree=0.769, adj=0.283, (0 split)
##       YearRemodAdd < 1950.5  to the left,  agree=0.742, adj=0.200, (0 split)
##       LotArea      < 6411    to the left,  agree=0.702, adj=0.075, (0 split)
##       GarageArea   < 230     to the left,  agree=0.691, adj=0.042, (0 split)
##       OverallQual  < 3.5     to the left,  agree=0.685, adj=0.025, (0 split)
##
## Node number 9: 186 observations,    complexity param=0.01183168
##   mean=165304.6, MSE=1.471188e+09
##   left son=18 (64 obs) right son=19 (122 obs)
##   Primary splits:
##       OverallQual  < 5.5     to the left,  improve=0.2607831, (0 missing)
##       BsmtFinSF1   < 618     to the left,  improve=0.1998511, (0 missing)
##       YearRemodAdd < 1980.5  to the left,  improve=0.1856281, (0 missing)
##       Fireplaces   < 0.5     to the left,  improve=0.1733604, (0 missing)
##       LotArea      < 12180   to the left,  improve=0.1715189, (0 missing)
##   Surrogate splits:
##       YearRemodAdd < 1971.5  to the left,  agree=0.753, adj=0.281, (0 split)
##       YearBuilt    < 1971.5  to the left,  agree=0.737, adj=0.234, (0 split)
##       GarageArea   < 290     to the left,  agree=0.720, adj=0.188, (0 split)
##       BsmtFinSF1   < 10      to the left,  agree=0.683, adj=0.078, (0 split)
##       BedroomAbvGr < 3.5     to the right, agree=0.683, adj=0.078, (0 split)
##
## Node number 10: 174 observations
##   mean=202038.8, MSE=1.600723e+09
##
## Node number 11: 22 observations
##   mean=268469.5, MSE=3.17058e+09
```

```
##
## Node number 12: 85 observations
##   mean=264630.2, MSE=2.666789e+09
##
## Node number 13: 19 observations
##   mean=351838.2, MSE=4.705288e+09
##
## Node number 14: 27 observations
##   mean=350923.4, MSE=2.838409e+09
##
## Node number 15: 15 observations
##   mean=457065.7, MSE=1.717852e+10
##
## Node number 16: 120 observations
##   mean=107412.9, MSE=6.818746e+08
##
## Node number 17: 252 observations
##   mean=137263.7, MSE=7.396912e+08
##
## Node number 18: 64 observations
##   mean=138261, MSE=1.15381e+09
##
## Node number 19: 122 observations
##   mean=179491.4, MSE=1.052756e+09
```

Now, Plotting our decision tree. Comparing actual and predicted Sale Prices for the new data.

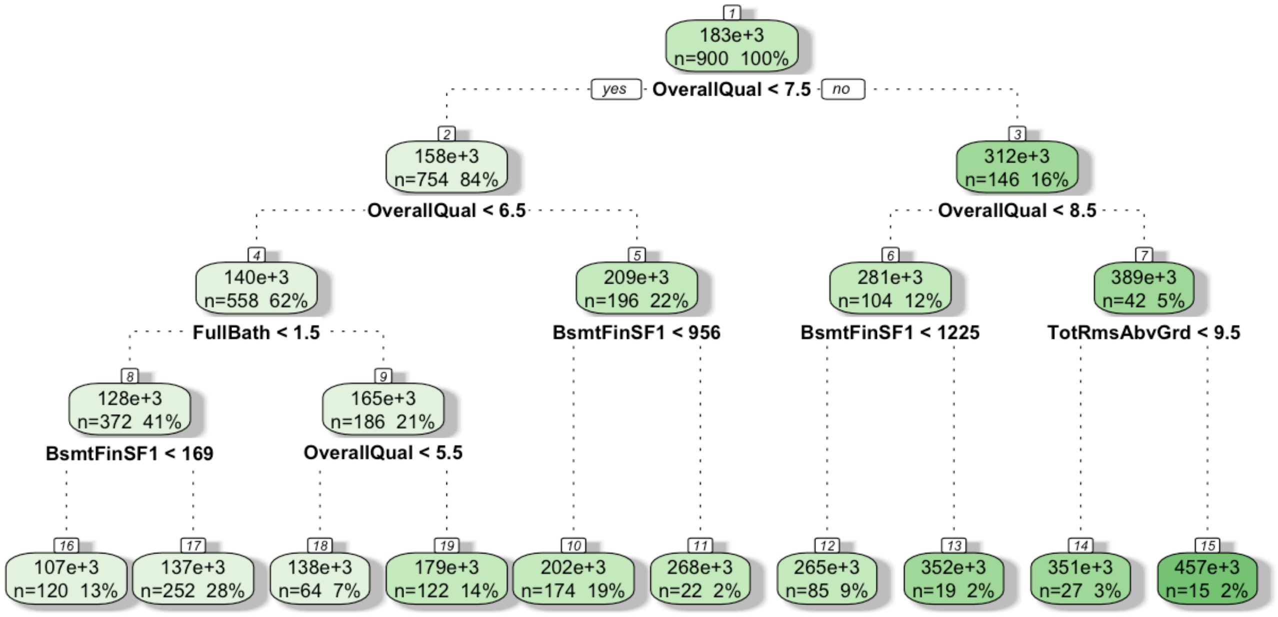
```
#install.packages("rattle")
# Loading the rattle library for decision tree visualization
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
# Visualizing the decision tree
fancyRpartPlot(DecisionTree)
```



Rattle 2024-Jan-03 23:34:08 gauravkudeshia

```
# Making predictions on new data using the decision tree model
Predicted_SalePrice_DT = predict(DecisionTree, newdata = data_frame_predict)

# Creating a data frame with actual and predicted Sale Prices
SalesPrice_table_DT = data.frame(Actual_Price = data_frame_predict$SalePrice, Pred
icted_Price = Predicted_SalePrice_DT)

# Displaying the first few rows of the comparison table
head(SalesPrice_table_DT)
```

```
##   Actual_Price Predicted_Price
## 1       110000        137263.7
## 2       153000        137263.7
## 3       180000        202038.8
## 4       240000        202038.8
## 5       125500        137263.7
## 6       128000        137263.7
```

Now, using RMSE and MAE, for the Decision Tree model to know how well the decision tree model is performing on the new data

```
# Calculate Root Mean Squared Error (RMSE)
DecisionTreea_rmse <- sqrt(mean((data_frame_predict$SalePrice - Predicted_SalePrice_DT)^2))
DecisionTreea_rmse
```

```
## [1] 35295.62
```

```
# Calculate Mean Absolute Error (MAE)
DecisionTreea_mae <- mean(abs(data_frame_predict$SalePrice - Predicted_SalePrice_DT))
DecisionTreea_mae
```

```
## [1] 27271.18
```

Use classification to model OverallQual (rating 7 and above is considered as class 1, otherwise class zero)

Classification

```
# Creating a new binary variable "label" based on the condition
data_frame$label = as.factor(ifelse(data_frame$OverallQual >= 7, 1, 0))

summary(data_frame)
```

```
##     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      label
##  Min.   : 34900   0:558
##  1st Qu.:130000   1:342
##  Median :163000
##  Mean   :183108
##  3rd Qu.:216878
##  Max.   :755000
```

Now using Logistic Regression which is a classification technique.

```
#Model
logis_ClassModel = glm(label ~ LotArea + YearBuilt + YearRemodAdd + BsmtFinSF1 + F
ullBath + HalfBath + BedroomAbvGr + TotRmsAbvGrd + Fireplaces + GarageArea + YrSol
d + SalePrice ,data = data_frame, family = "binomial")

#Displaying Summary
summary(logis_ClassModel)
```

```
## 
## Call:
## glm(formula = label ~ LotArea + YearBuilt + YearRemodAdd + BsmtFinSF1 + 
##     FullBath + HalfBath + BedroomAbvGr + TotRmsAbvGrd + Fireplaces + 
##     GarageArea + YrSold + SalePrice, family = "binomial", data = data_frame)
## 
## 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
```

```
#Model with appropriate feature
logis_ClassModel_1 = glm(label ~ LotArea + BsmtFinSF1 + SalePrice ,data = data_fra
me, family = "binomial")

#Displaying Summary
summary(logis_ClassModel_1)
```

```
##
## Call:
## glm(formula = label ~ LotArea + BsmtFinSF1 + SalePrice, family = "binomial",
##     data = data_frame)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.359e+00  6.353e-01 -14.733  < 2e-16 ***
## LotArea     -4.767e-05  8.954e-06  -5.324 1.01e-07 ***
## BsmtFinSF1  -1.879e-03  3.140e-04  -5.983 2.19e-09 ***
## SalePrice    5.620e-05  3.894e-06  14.432  < 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:  508.84  on 896  degrees of freedom
## AIC: 516.84
##
## Number of Fisher Scoring iterations: 7
```

Now making prediction of prices and plotting an ROC curve

```
# Creating a binary variable "label" based on the condition for prediction
data_frame_predict$label = as.factor(ifelse(data_frame_predict$OverallQual >= 7,
1, 0))

# Predicting the probabilities using the logistic regression model
OverallQualityPrediction = predict(logis_ClassModel, newdata = data_frame_predict,
type='response')

# Loading the pROC library
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

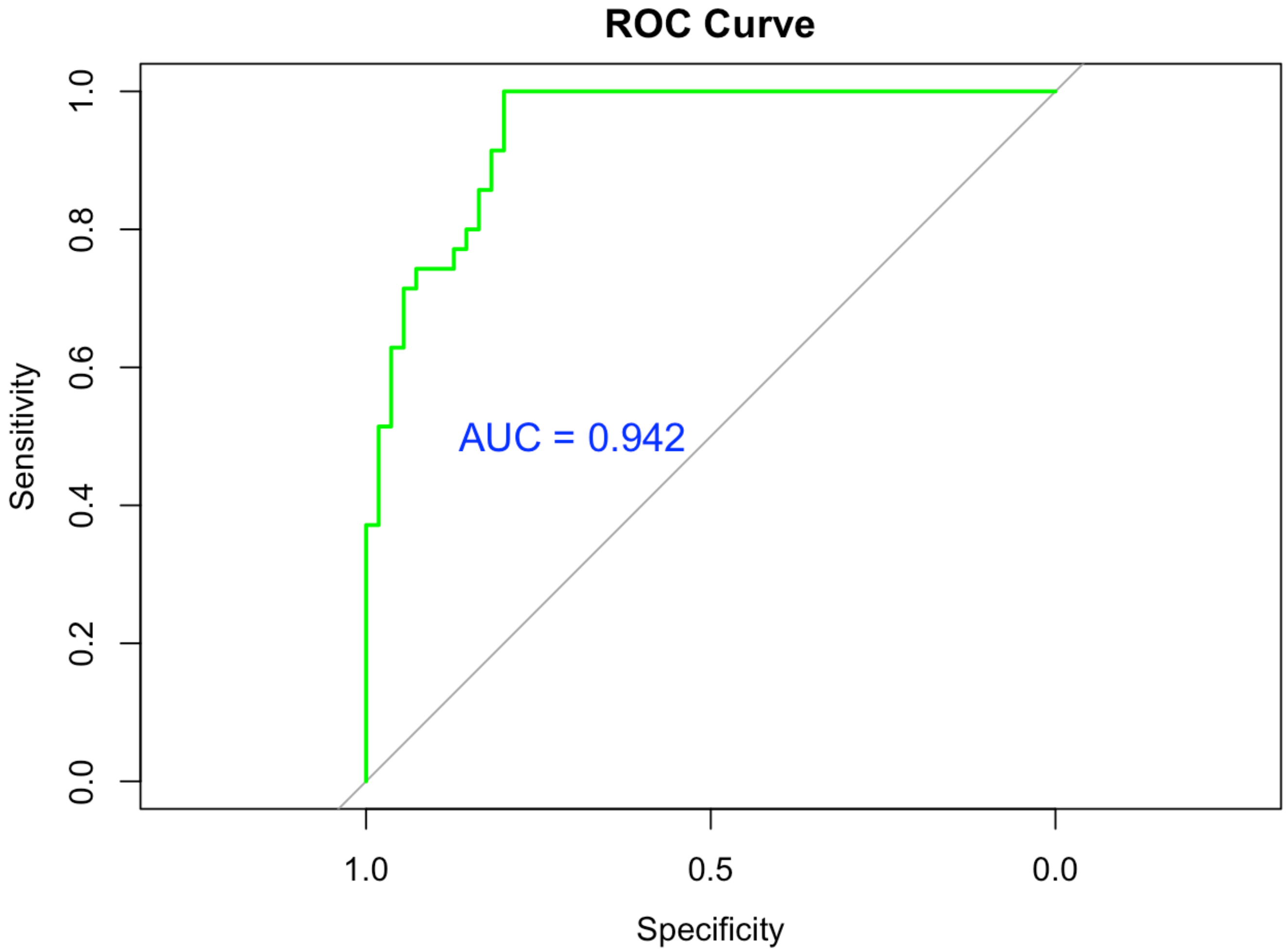
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Creating an ROC curve
roc_curve <- roc(data_frame_predict$label, OverallQualityPrediction)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Ploting the ROC curve
plot(roc_curve, main = "ROC Curve", col = "green", lwd = 2)

# Adding AUC (Area Under the Curve) to the plot
auc_value <- auc(roc_curve)
text(0.7, 0.5, paste("AUC =", round(auc_value, 3)), col = "blue", cex = 1.2)
```



Confusion Matrix

```
suppressMessages(library(caret))

# Converting predicted probabilities to binary predictions based on a threshold of
0.5
Predicted = as.factor(ifelse(OverallQualityPrediction > 0.5,1,0))

# Creating a confusion matrix
ConfuMatrix  = confusionMatrix(Predicted,data_frame_predict$label)

ConfuMatrix
```

```
## 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 : 0.8545
##             Specificity : 0.8000
##          Pos Pred Value : 0.8704
##          Neg Pred Value : 0.7778
##              Prevalence : 0.6111
##          Detection Rate : 0.5222
##    Detection Prevalence : 0.6000
##       Balanced Accuracy : 0.8273
## 
##        'Positive' Class : 0
## 
```

Conclusion

After carefully observing the different parameters responsible for determining the Sales Price of a house and their descriptive statistics as shown below:

• LotArea: - Range: The minimum lot area is 1491 square feet, and the maximum is 215,245 square feet. - Distribution: The mean lot area is approximately 10,795 square feet, with a median (50th percentile) of 9,442 square feet.

- OverallQual (Overall Quality): - Range: The overall quality ranges from 1 (lowest) to 10 (highest). - Distribution: The mean overall quality is approximately 6.136, with a median of 6.000.

- YearBuilt: - Range: Houses in the dataset were built between 1880 and 2010. - Distribution: The mean year of construction is around 1971, with a median of 1973.

- YearRemodAdd (Year of Remodeling or Addition): - Range: Remodeling or additions occurred between 1950 and 2010. - Distribution: The mean year of remodeling or addition is approximately 1985, with a median of 1994.

- BsmtFinSF1 (Basement Finished Square Feet): - Range: Basement finished square feet range from 0 to 2260. - Distribution: The mean finished square feet is 446.5, and the median is 384.0.

- FullBath, HalfBath: - Range: Full baths range from 0 to 3, while half baths range from 0 to 2. - Distribution: On average, houses have around 1.564 full baths and 0.386 half baths. • BedroomAbvGr (Bedrooms Above Ground): - Range: The number of bedrooms above ground ranges from 0 to 8. - Distribution: The mean is approximately 2.843 bedrooms, with a median of 3.000.

- TotRmsAbvGrd (Total Rooms Above Ground): - Range: Total rooms above ground range from 2 to 14. - Distribution: The mean is approximately 6.482 rooms, with a median of 6.000.

- Fireplaces: - Range: The number of fireplaces ranges from 0 to 3. - Distribution: On average, houses have around 0.628 fireplaces.

- GarageArea: - Range: Garage areas range from 0 to 1390 square feet. - Distribution: The mean garage area is 472.6 square feet, with a median of 480.0.

- YrSold: - Range: Houses were sold between 2006 and 2010. - Distribution: The mean year of sale is approximately 2008.

- SalePrice: - Range: Sale prices range from $34,900 to $755,000. - Distribution: The mean sale price is $183,108, with a median of $163,000

 From these parameters and our algorithm results we can ascertain that the significant factors out of all the given parameters are: LotArea Overall Quality, ‘YearRemodAdd’ that is Remodel Date Year, ‘BsmtFinSF1’ that is Finished square feet, ‘BedroomAbvGr’ that is Number of Bedrooms above the ground, ‘TotRmsAbvGrd’ Number of rooms above the ground, Number of fireplaces and Size of garage in square feet.

 We are considering these factors as the significant ones because the the p-value for all these factors is less than 0.05((p-value) < 0.05). This result is obtained from Regression model using the lm function.

 The results also describe that the value of SalesPrice is increasing with the increase in value of LotArea Overall Quality, ‘YearRemodAdd’ that is Remodel Date Year, ‘BsmtFinSF1’ that is Finished square feet, ‘BedroomAbvGr’ that is Number of Bedrooms above the ground, ‘TotRmsAbvGrd’ Number of rooms above the ground, Number of fireplaces and Size of garage in square feet which can be noted with the help of the correlation coefficients stating a positive and non zero value.

 The maximum increase is observed with the increase in Overall Quality with the correlation coefficient value of 0.7962135.