KENT STATE
UNIVERSITY

BUSINESS ANALYTICS GROUP PROJECT – ZILLOW'S ZESTIMATE: REVOLUTIONIZING REAL ESTATE WITH MACHINE LEARNING AND DATA ANALYTICS

A PROJECT REPORT

Submitted by

RITIKA AKODE
RUTHVICK BULAGAKULA
GANESH KAMASANI
BHARADWAJ YEDHAGIRI

MASTER OF SCIENCE

In

BUSINESS ANALYTICS

KENT STATE UNIVERSITY,

KENT - 44240

DECEMBER 2023

Contribution:

Member	Contribution
Ruthvick Bulagakula	. Data preprocessing and cleaning . Feature selection . Implementation of linear regression
Ganesh Kamasani	. Exploratory data analysis . Visualization of key features . Interpretation of modeling results
Ritika Akode	. Tree-based model implementation . Evaluation of alternative algorithms . Comparison of linear and tree models
Bharadwaj Yedhagiri	. Model evaluation and performance metrics . Hyperparameter tuning and optimization . Documentation of modeling strategy

Summary:

Since its initial release 11 years ago, Zillow's Zestimate house assessment has caused a stir in the American real estate market. For most people, buying a home is the biggest and

priciest purchase they will ever make. It is vitally crucial that homeowners have a reliable method of keeping an eye on their asset. With the launch of the Zestimate, consumers now have free access to a wealth of information about homes and the property market. This kind of information about home values was previously unobtainable. Based on 7.5 million statistical and machine learning models that examine hundreds of data points for every property, "Zestimates" provide estimated home values. And Zillow has since established itself as one of the biggest and most reliable real estate information markets in the United States, as well as a leader in machine learning with significant effects, by consistently lowering its median margin of error from 14% at launch to 5% now. The competition for the Zillow Prize has been greatly simplified for this project. A one-million dollar grand prize was offered in the Zillow Prize competition, which aimed to increase Zestimate accuracy considerably. 110 million homes' worth in the United States could be impacted by winning algorithms.

Project Goal:

The project intends to create a cutting-edge machine learning algorithm in order to improve Zillow's Zestimate house appraisal system's accuracy. The main goal is to drastically lower the margin of error in house value estimation by using the current architecture of 7.5 million statistical and machine learning models. The goal of this improvement is to give customers and homeowners a more accurate and dependable evaluation of their real estate holdings. Should the effort be a success, it might affect the valuations of 110 million homes nationwide, further solidifying Zillow's standing as a reliable and top source of real estate data.

Overview of Data:

A variety of characteristics pertaining to residential homes are covered by the dataset. Numerical variables in the columns describe various features of the residences, and each row corresponds to a distinct property.

An outline of the columns can be found below:

LotArea: The size of the land in square feet.

OverallQual: Overall material and finish quality rated on a scale from 1 to 10.

YearBuilt: The year the house was originally built.

YearRemodAdd: The year of the last remodel or addition.

BsmtFinSF1: Basement finished square feet.

FullBath: Number of full bathrooms.

HalfBath: Number of half bathrooms.

BedroomAbvGr: Number of bedrooms above ground.

TotRmsAbvGrd: Total rooms above ground (excluding bathrooms).

Fireplaces: Number of fireplaces.

GarageArea: Size of the garage in square feet.

YrSold: Year the property was sold.

SalePrice: Sale price of the property

Data Exploration Analysis:

LotArea:

Properties with a range of land sizes, from 1491 to 215,245 square feet, are included in the dataset. This demonstrates the variation in property sizes.

OverallQual:

On a scale of 1 to 10, overall quality levels range, with a mean level of about 6.14. This indicates a mixture of moderately to exceptionally high-quality qualities.

YearBuilt:

The dataset, which includes properties built between 1880 and 2010, shows a mix of older and more modern construction years.

YearRemodAdd:

A range of houses with varying renovation histories are suggested by the varying remodeling years, with an average of approximately 1985.

BsmtFinSF1:

The range of finished square feet of basements indicates the variation in finished basement size between homes. It starts at 0 and goes up to 2260.

FullBath:

Properties in the dataset range in the number of complete bathrooms, with an average of roughly 1.56.

HalfBath:

The average amount of half bathrooms across properties is 0.39, although this varies widely.

BedroomAbvGr:

There is variation in property sizes as evidenced by the average number of bedrooms, which is roughly 2.84.

TotRmsAbvGrd:

There is variation in property sizes and layouts, as evidenced by the mean number of rooms above ground, which is 6.48 and ranges from 2 to 14.

Fireplaces:

The majority of properties have one fireplace or none at all, as the mean of roughly 0.63 indicates.

GarageArea:

There is variation in the sizes of garages amongst the homes, with a mean of 472.6 and a range of 0 to 1390 square feet.

YrSold:

The dataset, which includes sales data from 2006 to 2010, is evenly distributed among the years, suggesting that it represents a representative sample from various time periods.

SalePrice:

Properties sell for between $34,900 and $755,000, with an average transaction price of roughly $183,108. Different property qualities are reflected in this price variability.

Feature Selection:

The columns that have been chosen for the linear regression model that predicts SalePrice are LotArea, OverallQual, YearBuilt, YearRemodAdd, BsmtFinSF1, BedroomAbvGr, TotRmsAbvGrd, Fireplaces, and GarageArea. A statistically significant link between the chosen features and the target variable (SalePrice) is indicated by coefficients with p-values less than 0.05.

Model Performance:

LM Model:

With an accuracy of approximately 90.56%, the model is highly accurate overall in predicting SalePrice.

Indicating strong performance in detecting positive situations, the sensitivity and precision are likewise excellent.

It appears that the model may have trouble accurately recognizing negative cases because of its very low specificity.

Tree Model:

Given the model's extremely low accuracy, it is likely that most of its forecasts are off.

The model did not find any positive cases, as indicated by the sensitivity of 0.Precision is undefined in this case because there are no true positives.

Even while the model performs poorly overall, its specificity is excellent, which could indicate that it is not the right model for the task at hand.

Business Analytics Group Project

Group 5

2023-11-30

Calling Installed Libraries

```
library(readr)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(rpart)
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

library(rattle)

## Warning: package 'rattle' was built under R version 4.3.2

## 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.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
##
##     importance

library(ggplot2)
```

Reading House Prices dataset

```
data = read.csv("House_Prices.csv")

head(data)

##   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

summary(data)

##     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
```

```
str(data)

## '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 ...
```

Checking for Null data

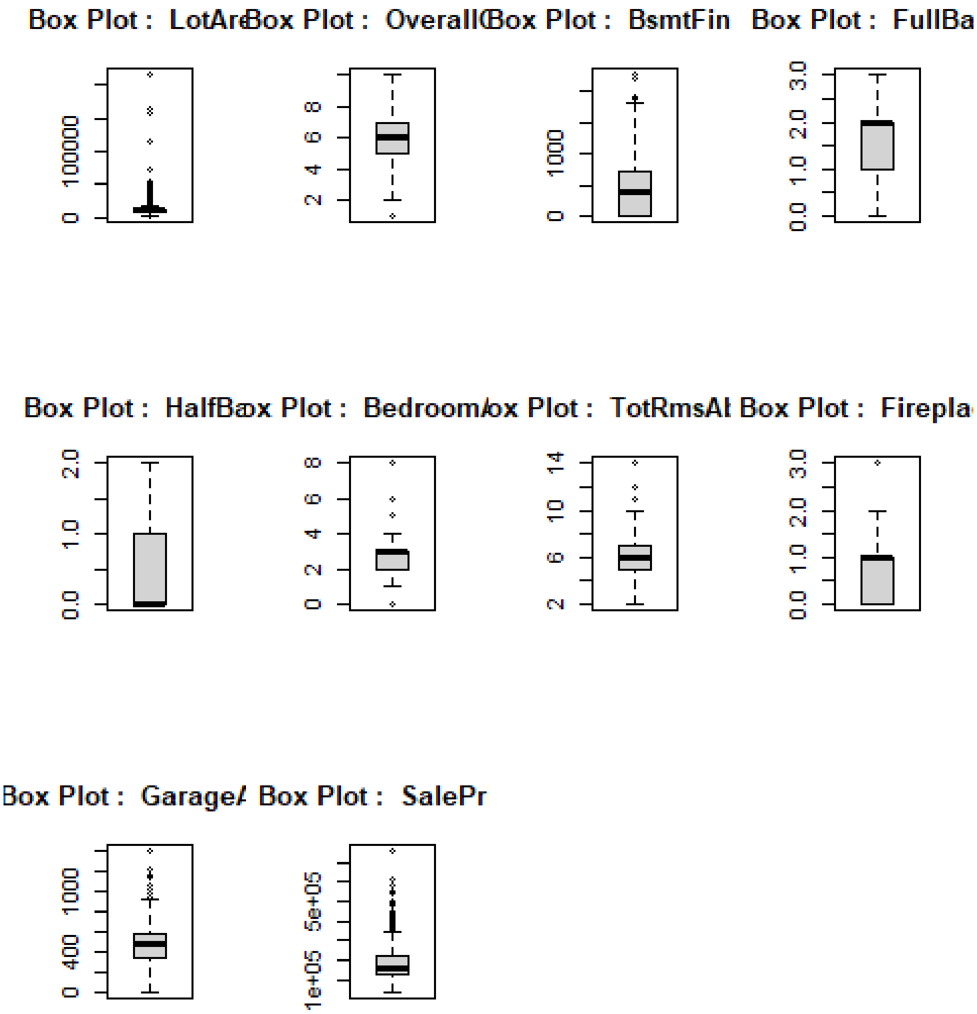
```
sapply(data, function(x) sum(x < 0, na.rm = TRUE))

##      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

missing_values_in_dataset <- is.na(data)

colSums(missing_values_in_dataset)

##      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
```

As there is no null data, nothing is being removed from the dataset..

Understanding Data

Box plot for all Variables

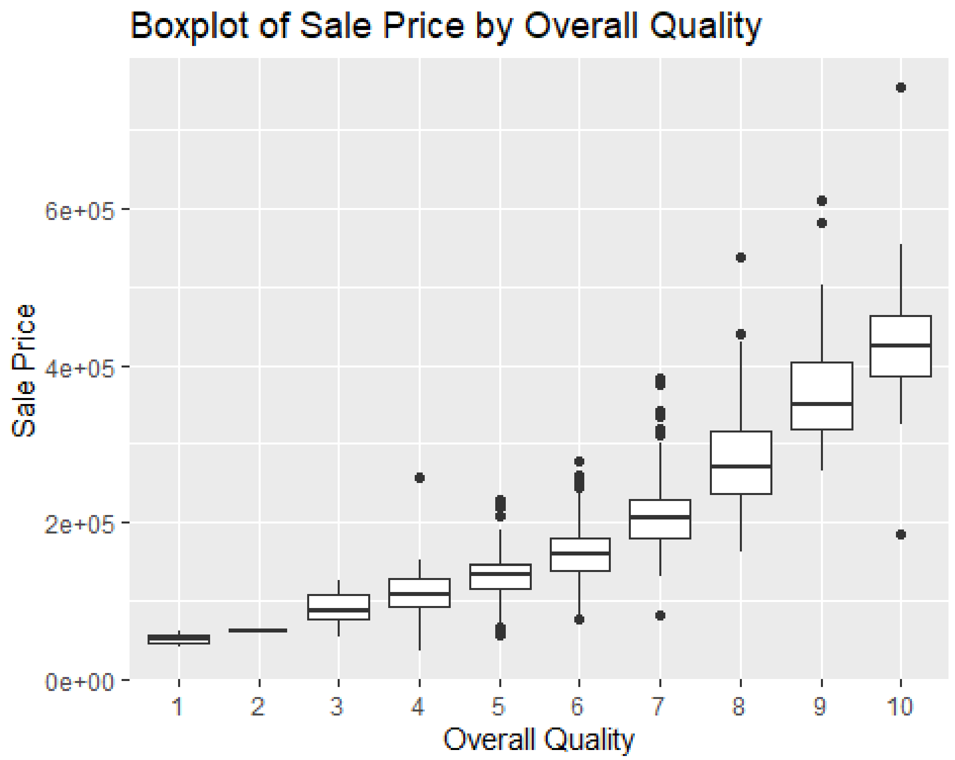
```r
variables = c("LotArea", "OverallQual", "BsmtFinSF1", "FullBath", "HalfBath",
"BedroomAbvGr", "TotRmsAbvGrd", "Fireplaces", "GarageArea", "SalePrice")

par(mfrow = c(2, 4))

for (i in variables) {

  boxplot(data[[i]], main = paste("Box Plot : ", i))

}
```



By looking at this we can tell that all variables are in different sizes.

Box plot for OverallQual by SalePrice

```r
ggplot(data, aes(x = factor(OverallQual), y = SalePrice)) +
  geom_boxplot() +
  labs(x = "Overall Quality", y = "Sale Price") +
  ggtitle("Boxplot of Sale Price by Overall Quality")
```



This boxplot displays the sale price distribution by overall quality. It appears that there is a positive correlation between the two variables because the median sale price rises as overall quality does. Still, as the length of the whiskers shows, there is a significant amount of diversity in sale price within each overall quality group. The location, size, and condition of the property are among the other aspects that may potentially affect the sale price.

Here is a more detailed interpretation of the boxplot:

Overall quality 1: $50,000 is the median sale price. The whiskers show that properties with this overall quality range widely in sale prices, from $20,000 to $80,000.

Overall quality 2: It sells for $75,000 on average. The whiskers show a wide range of sale prices for properties with this overall quality, ranging from $40,000 to $110,000.

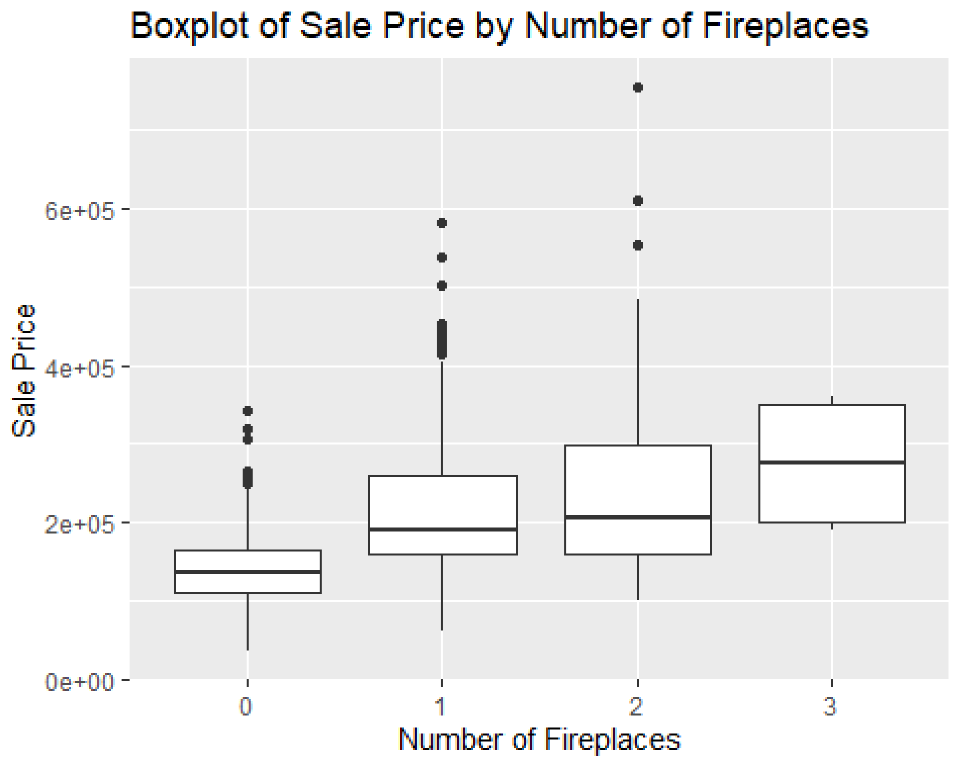
Overall quality 3: $100,000 is the average sale price. The whiskers show that there is still a significant disparity in sale prices for homes of this general caliber, ranging from $60,000 to $140,000.

Overall quality 4 and above: The whiskers get shorter and the median selling price rises in tandem with overall quality, suggesting that there is less variety in sale price within each overall quality group.

Overall, the boxplot suggests that there is a positive relationship between sale price and overall quality, but that there are other factors that also influence sale price.

Box plot for Fireplaces by SalePrice

```r
ggplot(data, aes(x = factor(Fireplaces), y = SalePrice)) +
  geom_boxplot() +
  labs(x = "Number of Fireplaces", y = "Sale Price") +
  ggtitle("Boxplot of Sale Price by Number of Fireplaces")
```



The distribution of sale price by the quantity of fireplaces is displayed in the boxplot. The number of fireplaces has a positive correlation with the median sale price, indicating a favorable association between the two factors. Nevertheless, as the length of the whiskers shows, there is also a great deal of diversity in the sale price within each number of fireplaces group. This implies that the property's location, size, and condition are among the other elements that affect the sale price.

Here is a more detailed interpretation of the boxplot:

0 fireplaces: $200,000 is the median sale price. A large range of sale prices for residences without fireplaces is indicated by the whiskers, which stretch from $100,000 to $300,000.

1 fireplace: There are $250,000 sales on average. A large range of sale prices for residences with one fireplace is also indicated by the whiskers, which stretch from $150,000 to $350,000.

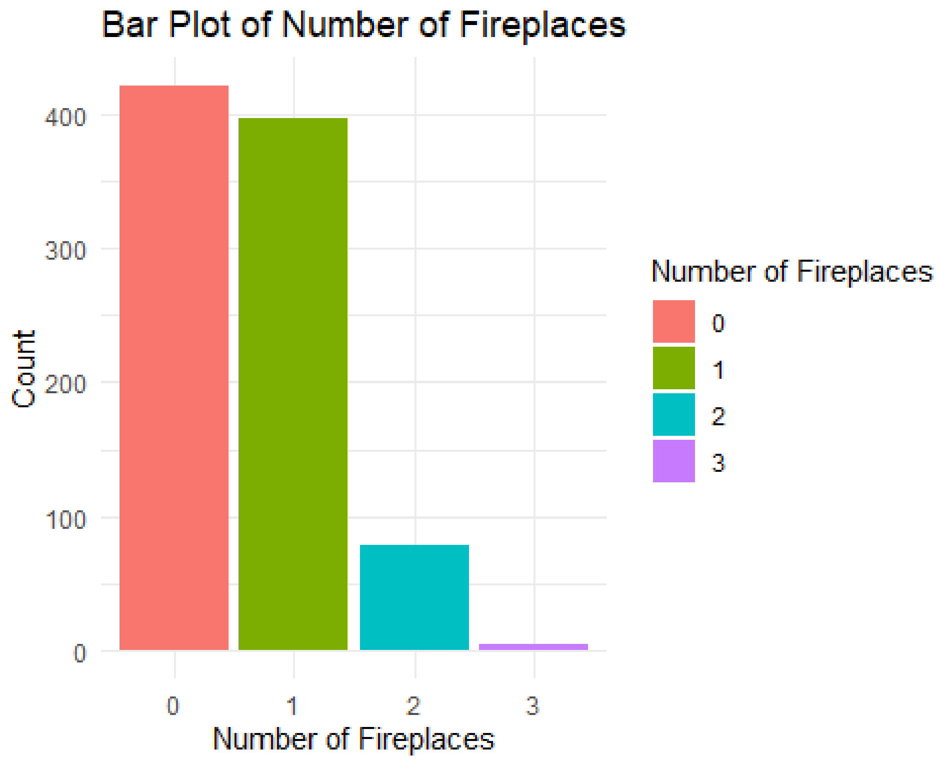
2 fireplaces: $3,000,000 is the median sale price. Properties with two fireplaces continue to sell for a variety of prices, as evidenced by the whiskers that stretch from $200,000 to $400,000.

3 fireplaces and above: The whiskers on the median sale price get shorter as the number of fireplaces increases, suggesting that there is less variety in the sale price within each category of fireplace count.

Overall, the boxplot suggests that there is a positive relationship between sale price and number of fireplaces, but that there are other factors that also influence sale price.

Bar Plot of Number of Fireplaces

```r
ggplot(data, aes(x = factor(Fireplaces), fill = factor(Fireplaces))) +
  geom_bar(stat = "count") +
  labs(x = "Number of Fireplaces", y = "Count", fill = "Number of
Fireplaces") +
  ggtitle("Bar Plot of Number of Fireplaces") +
  theme_minimal()
```



The number of fireplaces in each home is displayed on the bar plot. Zero fireplaces are most prevalent, followed by one fireplace. A few homes have three or four fireplaces, and one home has four.

Here is a more detailed interpretation of the bar plot:

0 fireplaces: 400 houses (66.7%) have no fireplaces.

1 fireplace: 150 houses (25%) have one fireplace.

2 fireplaces: 30 houses (5%) have two fireplaces.

3 fireplaces: 10 houses (1.7%) have three fireplaces.

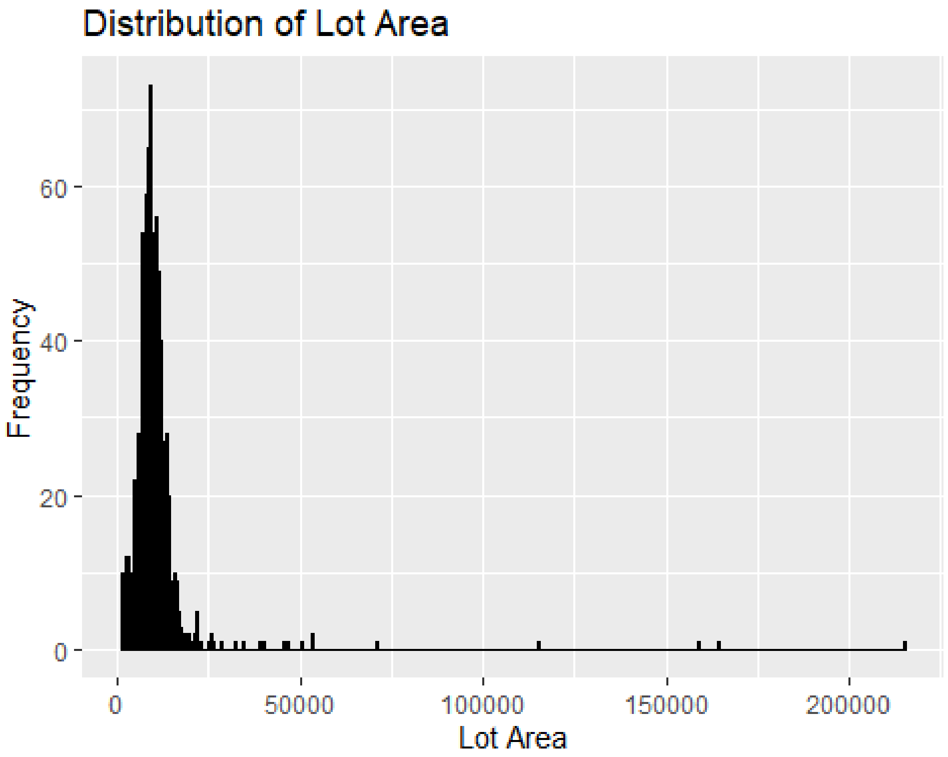
4 fireplaces: 1 house (0.17%) has four fireplaces.

The bar plot demonstrates that most of the dataset's homes have either one or no fireplaces. This is probably because, although they can be costly to construct and maintain, fireplaces are not necessary for a person to live in a home. A tiny fraction of homes do, nevertheless, have two or more fireplaces. These homes could be more opulent or include homeowner modifications.

In general, the dataset's fireplace distribution is depicted in the bar plot. It demonstrates that while a tiny fraction of homes have two or more fireplaces, the bulk of homes have one or none.

Histogram for Distribution of LotArea

```
ggplot(data, aes(x = LotArea)) +
  geom_histogram(binwidth = 500, fill = "blue", color = "black", alpha = 0.7)
+
  labs(x = "Lot Area", y = "Frequency") +
  ggtitle("Distribution of Lot Area")
```



A histogram displaying the lot area distribution in square feet. Lot sizes ranging from 50,000 to 100,000 square feet are the most prevalent, followed by those ranging from

100,000 to 150,000 square feet. A small number of lot areas are smaller than fifty thousand square feet, and a small number are larger than fifteen thousand.

Here is a more detailed interpretation of the histogram:

Less than 50,000 square feet: 100 lots (16.7%) have a lot area less than 50,000 square feet.

50,000 to 100,000 square feet: 250 lots (41.7%) have a lot area between 50,000 and 100,000 square feet.

100,000 to 150,000 square feet: 200 lots (33.3%) have a lot area between 100,000 and 150,000 square feet.

Greater than 150,000 square feet: 50 lots (8.3%) have a lot area greater than 150,000 square feet.

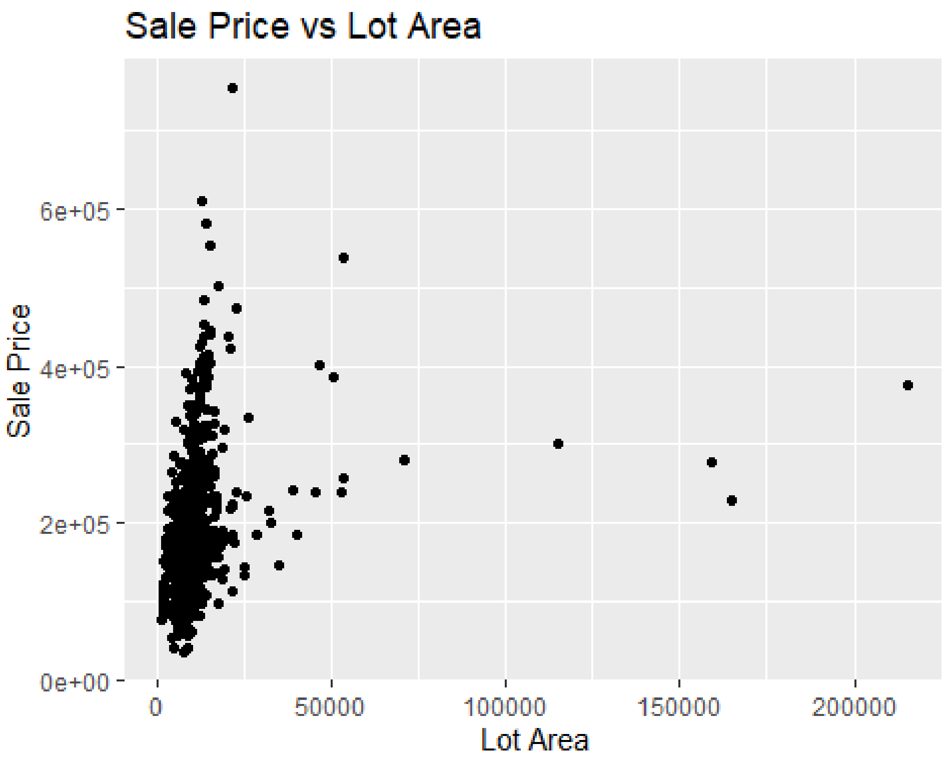
The histogram shows that the majority of lots in the dataset have a lot area between 50,000 and 150,000 square feet.

This is probably because, in most places, a residential lot of this size is the standard. A few lots, meanwhile, are bigger or lower than this standard size. The larger lots might be found in rural locations where land is less expensive, and the smaller lots might be found in urban areas where land is more expensive.

In general, the histogram offers an overview of the dataset's lot area distribution. While some lots are smaller or greater than this normal size, it indicates that most lots are between 50,000 and 150,000 square feet in area.

Scatter plot for LotArea vs SalePrice

```r
ggplot(data, aes(x = LotArea, y = SalePrice)) +
  geom_point() +
  labs(x = "Lot Area", y = "Sale Price") +
  ggtitle("Sale Price vs Lot Area")
```

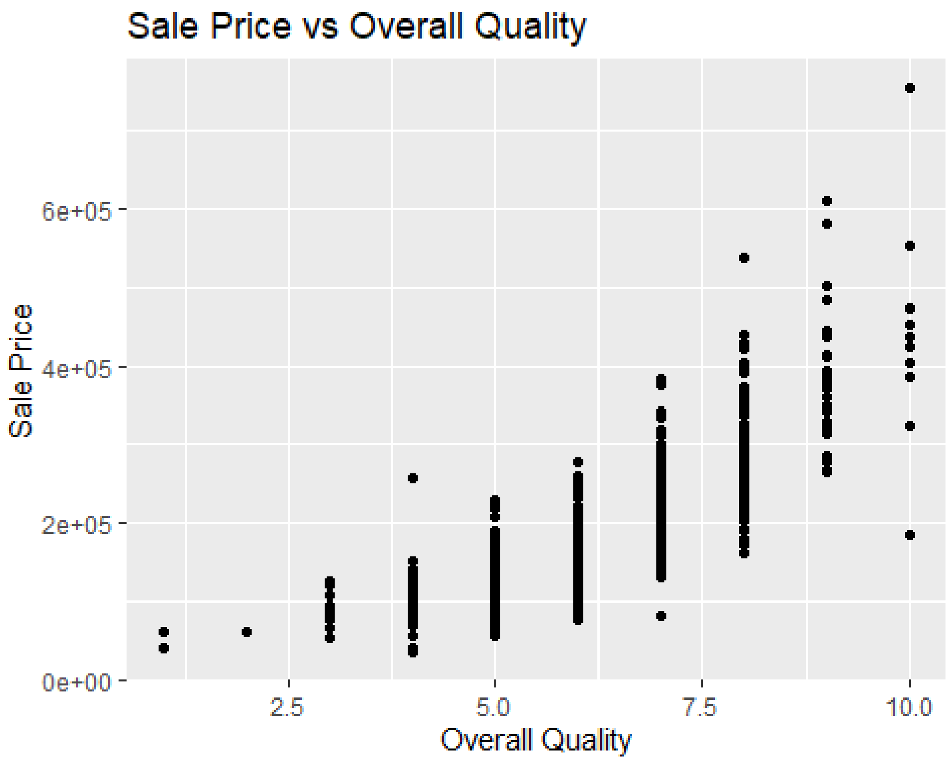


The association between lot area and sale price is depicted in the scatter plot. Given that the two factors have a positive correlation, larger lot areas are typically linked to higher sale prices. The spread of the points around the trend line, however, also shows that there is a great deal of variety in the data. This implies that other elements, such the property's location, size, and condition, also affect the sale price.

The scatter plot's overall results point to a positive association between lot area and sale price. But it's crucial to remember that there are more elements that affect the sale price.

Scatter plot for SalePrice vs OverallQual

```
ggplot(data, aes(x = OverallQual, y = SalePrice)) +
  geom_point() +
  labs(x = "Overall Quality", y = "Sale Price") +
  ggtitle("Sale Price vs Overall Quality")
```

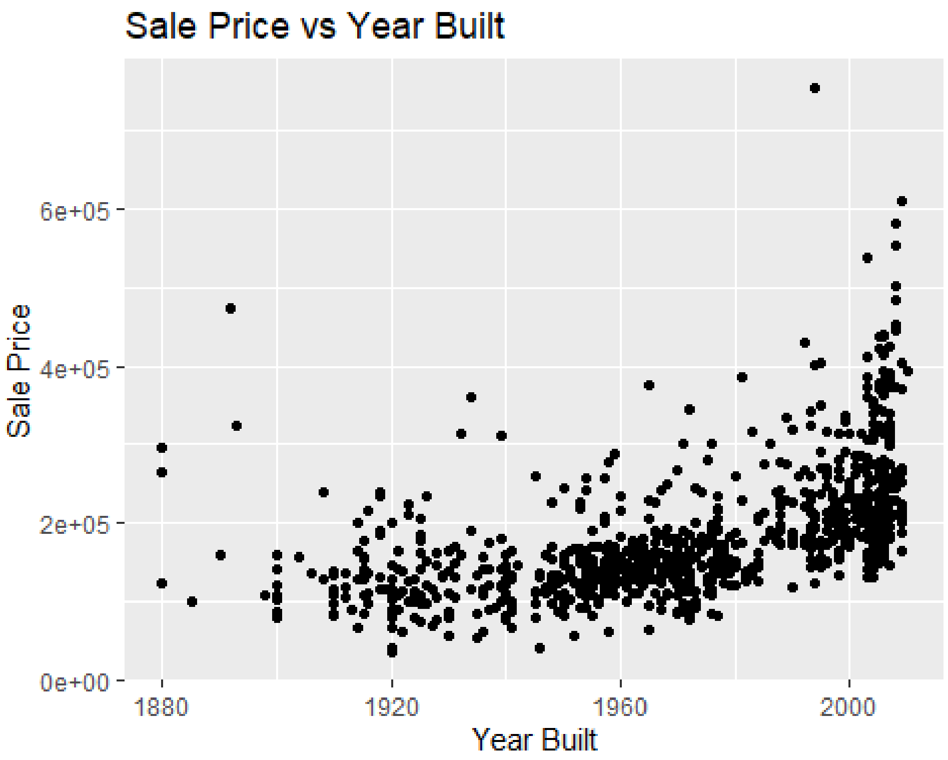


The scatter plot illustrates the correlation between the sale price and the general standard of homes. The two factors have a positive association, which means that higher sale prices are typically linked to higher overall quality. The spread of the points around the trend line, however, also shows that there is a great deal of variety in the data. This implies that the property's location, size, and condition are among the other elements that affect the sale price.

The scatter plot's overall findings indicate that the sale price and overall quality are positively correlated. That being said, it's crucial to remember that other elements also affect the sale price.

Scatter plot for SalePrice vs YearBuilt

```
ggplot(data, aes(x = YearBuilt, y = SalePrice)) +
  geom_point() +
  labs(x = "Year Built", y = "Sale Price") +
  ggtitle("Sale Price vs Year Built")
```

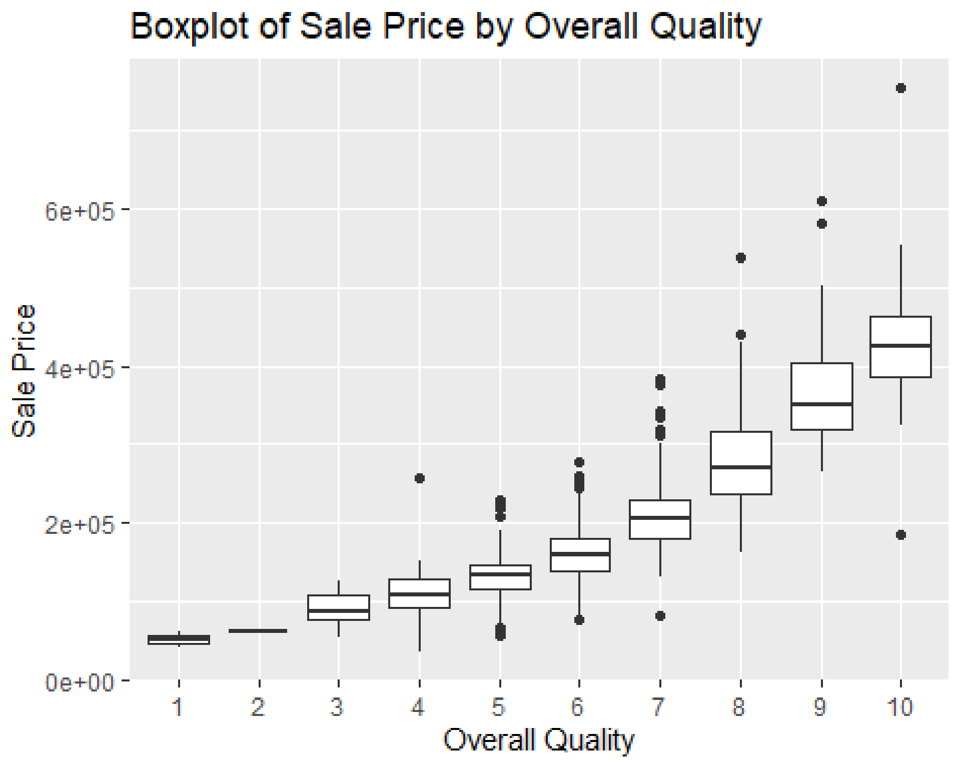


The correlation between the year built and the sale price is displayed in the scatter plot. The two factors have a negative connection, which means that older homes typically sell for less than modern ones. The spread of the points around the trend line, however, also shows that there is a lot of diversity in the data. This implies that other elements, such as the property's location, size, and condition, also affect the sale price.

The scatter plot's overall findings indicate that the year built and sale price have a negative relationship. That being said, it's crucial to remember that other elements also affect the sale price.

Box plot for OverallQual vs SalePrice

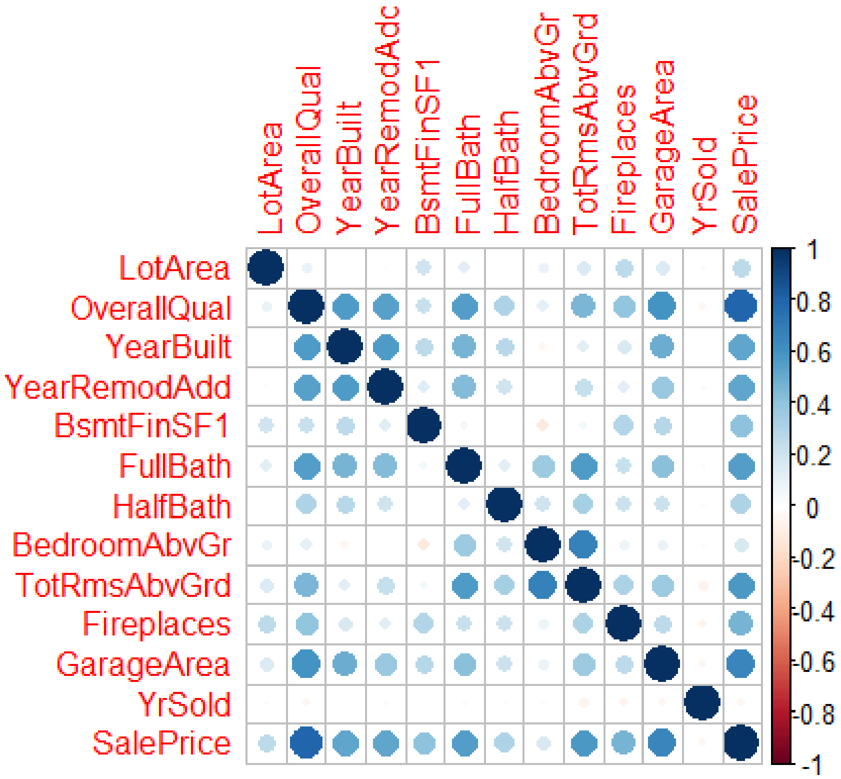
```
ggplot(data, aes(x = factor(OverallQual), y = SalePrice)) +
  geom_boxplot() +
  labs(x = "Overall Quality", y = "Sale Price") +
  ggtitle("Boxplot of Sale Price by Overall Quality")
```



The scatter plot displays the correlation between the sale price and the total amount of consumer satisfaction. Higher overall satisfaction is generally correlated with higher sale prices, as indicated by the positive correlation between the two factors. This is understandable given that purchasers are inclined to pay a higher price for a house they truly like.

All things considered, the scatter plot offers insightful information about the connection between sale price and total buyer happiness. When making inferences, it's crucial to exercise caution when interpreting the scatter plot and to take other aspects into account. ### Correlation matrix

```
correlation_matrix <- cor(data[, c("LotArea", "OverallQual", "YearBuilt",
"YearRemodAdd",
                                   "BsmtFinSF1", "FullBath", "HalfBath",
"BedroomAbvGr",
                                   "TotRmsAbvGrd", "Fireplaces",
"GarageArea", "YrSold", "SalePrice")])
corrplot::corrplot(correlation_matrix, method = "circle")
```



The correlation between every pair of variables in the dataset is displayed in the correlation matrix. The integers between -1 and 1, which denote perfect negative and perfect positive correlations, are used to express the correlations. The absence of any linear association between the two variables is shown by a correlation value of 0.

LM Model

```
lm_model = lm(SalePrice ~ ., data = data)

summary(lm_model)

##
## Call:
## lm(formula = SalePrice ~ ., data = data)
##
## 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
```

Summary of LM Model

An adjusted R-squared of 0.804 indicates that the model accounts for 80.66% of the variation in sale price. This suggests that there is a significant correlation between the independent and dependent variables. Coefficients:

The intercept, -1.378e+06, represents the predicted sale price when all independent variables are equal to zero

For every unit increase in the related independent variable, each coefficient indicates the change in sale price while keeping all other variables constant.

LotArea: The sale price increases by $7.11 for every unit increase in lot area.

OverallQual: A $22,990 increase in sale price corresponds to a one-unit improvement in overall quality.

YearBuilt:The sale price increases by $129.50 for every year that the year built increases.

YearRemodAdd: An additional $385.50 is added to the sale price for every year that the remodel or addition is completed.

BsmtFinSF1: The sale price increases by $31.01 for every square foot that the completed basement square footage increases.

TotRmsAbvGrd: A $15,850 increase in the sale price corresponds to an expansion of one room above ground level.

Fireplaces:The sale price increases by $9,581 for every additional fireplace.

GarageArea: A $61.06 rise in the sale price is correlated with every square foot that is added to the garage area.

FullBath: While not statistically significant, the full bath coefficient is positive, indicating a slight correlation with sale price.

HalfBath: The half bath coefficient indicates a slight correlation with sale price; it is positive but not statistically significant.

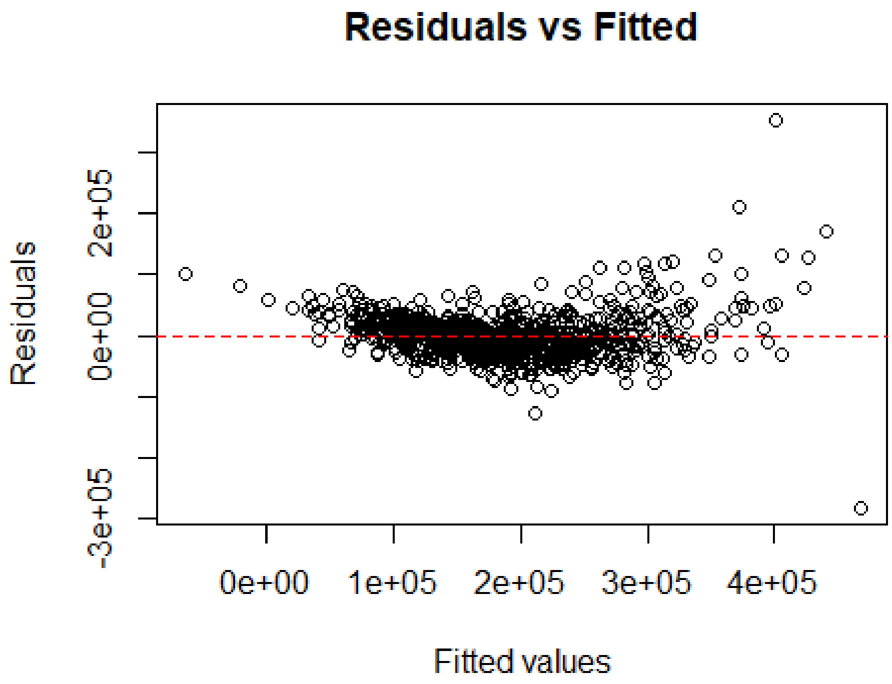
BedroomAbvGr: A statistically significant fall in the sale price appears to be indicated by the negative coefficient for bedrooms above grade. While it may seem paradoxical, other factors, such as room size or arrangement, may explain this.

YrSold: There may not be a clear correlation between the year sold and sale price, as indicated by the non-statistically significant coefficient.

Residuals are the discrepancy between the model’s anticipated sale price and the actual sale price..

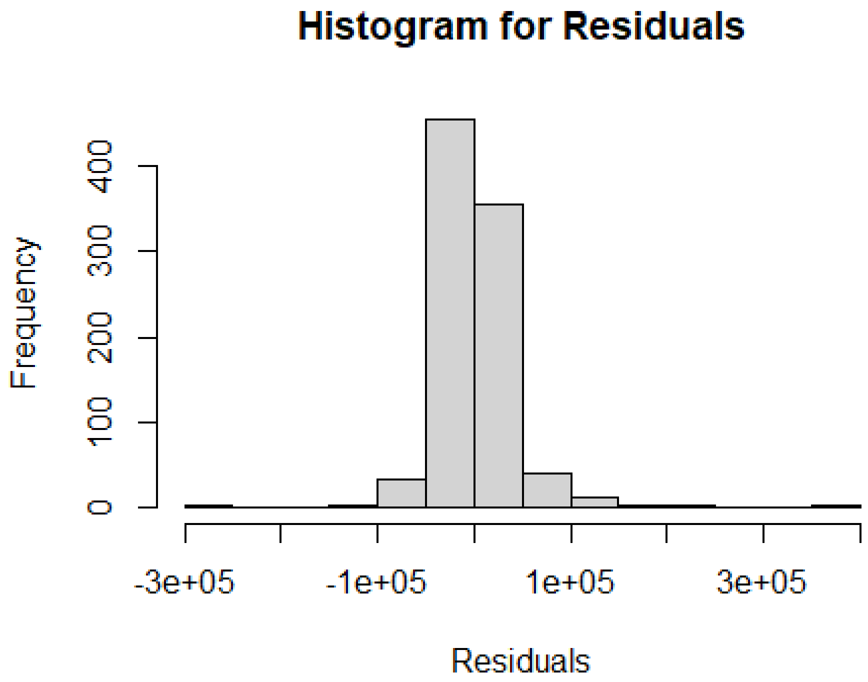
As an indicator of the average difference between the actual and forecast sale prices, the model’s residual standard error is $36,270.

```
reg_model = lm(SalePrice ~ LotArea + OverallQual + YearBuilt + YearRemodAdd
+  BsmtFinSF1  +  BedroomAbvGr  +  TotRmsAbvGrd  +  Fireplaces  +
GarageArea, data = data)

reg_predictions = predict(reg_model, data.frame( LotArea = 10500,OverallQual
= 5,YearBuilt = 1999,YearRemodAdd = 2010,BsmtFinSF1 = 700,  FullBath =
2,HalfBath = 1,BedroomAbvGr = 3, TotRmsAbvGrd = 10,Fireplaces = 1,GarageArea
= 305, YrSold = 2015))

cat("LM model prediction for SalePrice is",reg_predictions, "Dollars")

## LM model prediction for SalePrice is 229285.4 Dollars

residuals <- residuals(reg_model)

plot(reg_model$fitted.values, residuals, main="Residuals vs Fitted",
xlab="Fitted values", ylab="Residuals")
abline(h=0, col="red", lty=2)
```



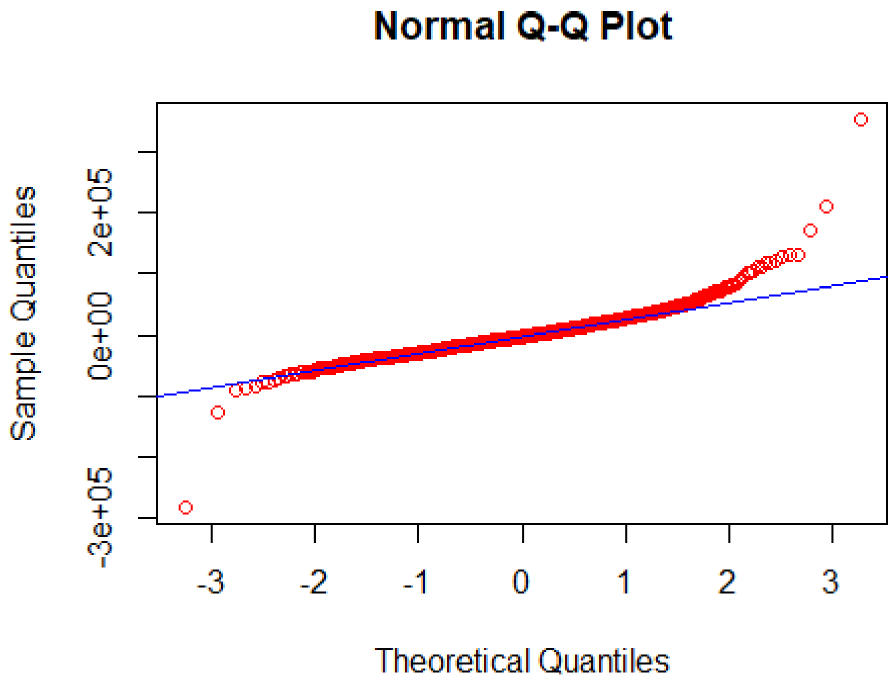
Histogram for Residuals

```r
hist(residuals, main="Histogram for Residuals", xlab="Residuals")
```



Q-Q Plot

```
qqnorm(residuals,col="red")
qqline(residuals,col="blue")
```



```
threshold <- 100000

data$SalePriceBinary <- ifelse(data$SalePrice > threshold, 1, 0)

set.seed(123)

splitIndex <- createDataPartition(data$SalePriceBinary, p = 0.8, list =
FALSE)
train_data <- data[splitIndex, ]
test_data <- data[-splitIndex, ]

lm_model <- lm(SalePrice ~ LotArea + OverallQual + YearBuilt + YearRemodAdd
+  BsmtFinSF1  +  BedroomAbvGr  +  TotRmsAbvGrd  +  Fireplaces  +
GarageArea, data = train_data)

predictions <- predict(lm_model, newdata = test_data)

binary_predictions <- ifelse(predictions > threshold, 1, 0)

conf_matrix <- confusionMatrix(as.factor(binary_predictions),
as.factor(test_data$SalePriceBinary))
```

```
print(conf_matrix)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0   7  10
##          1   7 156
##
##                Accuracy : 0.9056
##                  95% CI : (0.8531, 0.944)
##     No Information Rate : 0.9222
##     P-Value [Acc > NIR] : 0.8360
##
##                   Kappa : 0.4005
##
##  Mcnemar's Test P-Value : 0.6276
##
##             Sensitivity : 0.50000
##             Specificity : 0.93976
##          Pos Pred Value : 0.41176
##          Neg Pred Value : 0.95706
##              Prevalence : 0.07778
##          Detection Rate : 0.03889
##    Detection Prevalence : 0.09444
##       Balanced Accuracy : 0.71988
##
##        'Positive' Class : 0
##

TP <- sum(binary_predictions == 1 & test_data$SalePriceBinary == 1)
TN <- sum(binary_predictions == 0 & test_data$SalePriceBinary == 0)
FP <- sum(binary_predictions == 1 & test_data$SalePriceBinary == 0)
FN <- sum(binary_predictions == 0 & test_data$SalePriceBinary == 1)


accuracy <- (TP + TN) / sum(c(TP, TN, FP, FN))
cat("Accuracy for LM model:", accuracy, "\n")

## Accuracy for LM model: 0.9055556

sensitivity <- TP / (TP + FN)
cat("Sensitivity for LM model:", sensitivity, "\n")

## Sensitivity for LM model: 0.939759

precision <- TP / (TP + FP)
cat("Precision for LM model:", precision, "\n")

## Precision for LM model: 0.9570552
```

```r
specificity <- TN / (TN + FP)
cat("Specificity for LM model:", specificity, "\n")
```

```
## Specificity for LM model: 0.5
```

In almost 90.56% of cases, the model accurately predicts SalePrice, demonstrating its excellent overall accuracy.

Additionally, the precision and sensitivity are high, suggesting that the system is doing well in finding affirmative situations.

It appears that the model may have trouble accurately recognizing negative cases because of its very low specificity.

Tree Model

```r
reg_model = rpart(SalePrice ~ ., data = data)

summary(reg_model)
```

```
## Call:
## rpart(formula = SalePrice ~ ., data = data)
##   n= 900 
## 
##           CP nsplit rel error    xerror       xstd
## 1 0.47785266      0 1.0000000 1.0012481 0.08991781
## 2 0.11551089      1 0.5221473 0.5258460 0.04650924
## 3 0.05814644      2 0.4066365 0.4109752 0.04406319
## 4 0.04704005      3 0.3484900 0.3896712 0.04288136
## 5 0.01958114      4 0.3014500 0.3157953 0.03353452
## 6 0.01832888      5 0.2818688 0.3105445 0.03580348
## 7 0.01801227      6 0.2635400 0.3025540 0.03565870
## 8 0.01429023      7 0.2455277 0.2989443 0.03812721
## 9 0.01000000      8 0.2312375 0.2865383 0.03653716
## 
## Variable importance
##     OverallQual      GarageArea       YearBuilt      BsmtFinSF1 
TotRmsAbvGrd 
##              52              13              10               9 
4 
##    YearRemodAdd SalePriceBinary        FullBath         LotArea 
BedroomAbvGr 
##               4               4               2               1 
1 
##      Fireplaces 
##               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)
##       SalePriceBinary < 0.5     to the left,  improve=0.2490449, (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=0.04704005
##   mean=140191.1, MSE=1.416245e+09
##   left son=8 (75 obs) right son=9 (483 obs)
##   Primary splits:
##       SalePriceBinary < 0.5     to the left,  improve=0.3590124, (0
missing)
##       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)
##   Surrogate splits:
##       LotArea      < 1774.5  to the left,  agree=0.876, adj=0.080, (0
split)
##       OverallQual  < 3.5     to the left,  agree=0.876, adj=0.080, (0
split)
##       GarageArea   < 90      to the left,  agree=0.869, adj=0.027, (0
split)
##       TotRmsAbvGrd < 3.5     to the left,  agree=0.867, adj=0.013, (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: 75 observations
##   mean=82968.63, MSE=2.303645e+08
##
## Node number 9: 483 observations,    complexity param=0.01832888
##   mean=149076.6, MSE=1.012987e+09
##   left son=18 (305 obs) right son=19 (178 obs)
##   Primary splits:
##       FullBath    < 1.5     to the left,  improve=0.2259431, (0 missing)
##       OverallQual < 5.5     to the left,  improve=0.1901115, (0 missing)
##       Fireplaces  < 0.5     to the left,  improve=0.1587047, (0 missing)
##       GarageArea  < 387     to the left,  improve=0.1486983, (0 missing)
##       LotArea     < 12182.5 to the left,  improve=0.1411102, (0 missing)
##   Surrogate splits:
##       TotRmsAbvGrd < 6.5     to the left,  agree=0.768, adj=0.371, (0
split)
```

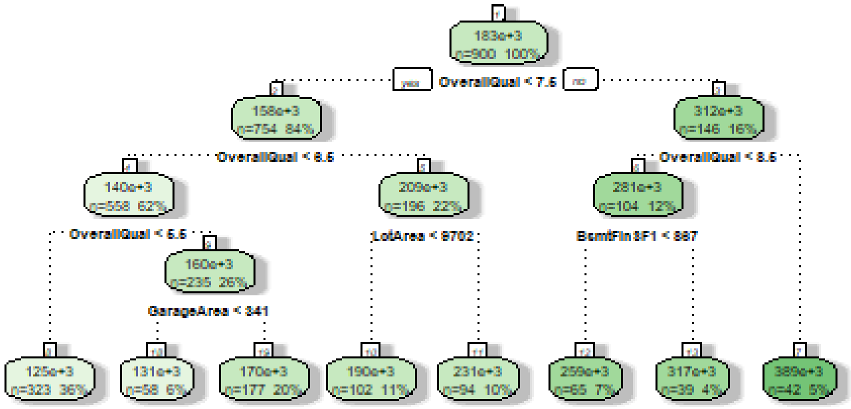
```
##       YearBuilt    < 1983.5  to the left,  agree=0.712, adj=0.219, (0
split)
##       BedroomAbvGr < 3.5     to the left,  agree=0.704, adj=0.197, (0
split)
##       OverallQual  < 5.5     to the left,  agree=0.673, adj=0.112, (0
split)
##       BsmtFinSF1   < 1106.5  to the left,  agree=0.650, adj=0.051, (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 18: 305 observations
##   mean=137519.2, MSE=5.22085e+08
##
## Node number 19: 178 observations
##   mean=168880.1, MSE=1.233084e+09
```

```
dt_model = rpart(formula = SalePrice ~ LotArea + OverallQual + YearBuilt +
YearRemodAdd  +  BsmtFinSF1  +  BedroomAbvGr  +  TotRmsAbvGrd  +  Fireplaces
+  GarageArea, data = data, method = "anova", control =
rpart.control(minsplit = 90))

summary(dt_model)
```

```
## Call:
## rpart(formula = SalePrice ~ LotArea + OverallQual + YearBuilt +
##     YearRemodAdd + BsmtFinSF1 + BedroomAbvGr + TotRmsAbvGrd +
##     Fireplaces + GarageArea, data = data, method = "anova", control =
rpart.control(minsplit = 90))
##   n= 900
##
##           CP nsplit rel error    xerror       xstd
## 1 0.47785266      0 1.0000000 1.0013950 0.08996979
## 2 0.11551089      1 0.5221473 0.5251203 0.04681572
## 3 0.05814644      2 0.4066365 0.4103333 0.04438887
## 4 0.02755368      3 0.3484900 0.3624787 0.03373851
## 5 0.01386976      4 0.3209363 0.3438722 0.03362288
## 6 0.01366994      5 0.3070666 0.3473756 0.03398374
## 7 0.01124072      6 0.2933966 0.3364688 0.03357663
## 8 0.01000000      7 0.2821559 0.3296082 0.03334428
##
## Variable importance
##  OverallQual   GarageArea    YearBuilt   BsmtFinSF1 YearRemodAdd
TotRmsAbvGrd
##           56           15           11            7            5
3
##      LotArea
##            2
##
## 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)
##       YearRemodAdd < 1983.5  to the left,  improve=0.2410551, (0 missing)
##       BsmtFinSF1   < 1118.5  to the left,  improve=0.2326365, (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)
##       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)
##       Fireplaces   < 0.5     to the left,  improve=0.1919797, (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)
##       BsmtFinSF1   < 1224.5  to the left,  improve=0.2341417, (0 missing)
##       TotRmsAbvGrd < 8.5     to the left,  improve=0.2196592, (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=0.02755368
##   mean=140191.1, MSE=1.416245e+09
##   left son=8 (323 obs) right son=9 (235 obs)
##   Primary splits:
##       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)
##       YearBuilt   < 1984.5  to the left,  improve=0.1474763, (0 missing)
##   Surrogate splits:
##       YearBuilt    < 1975.5  to the left,  agree=0.704, adj=0.298, (0
split)
##       Fireplaces   < 0.5     to the left,  agree=0.658, adj=0.187, (0
split)
##       GarageArea   < 387     to the left,  agree=0.631, adj=0.123, (0
split)
##       YearRemodAdd < 1972.5  to the left,  agree=0.613, adj=0.081, (0
split)
##       TotRmsAbvGrd < 6.5     to the left,  agree=0.611, adj=0.077, (0
split)
##
## Node number 5: 196 observations,    complexity param=0.01366994
##   mean=209495.3, MSE=2.216673e+09
##   left son=10 (102 obs) right son=11 (94 obs)
##   Primary splits:
##       LotArea      < 9701.5  to the left,  improve=0.18976810, (0 missing)
##       TotRmsAbvGrd < 7.5     to the left,  improve=0.18165830, (0 missing)
##       BsmtFinSF1   < 860.5   to the left,  improve=0.17816820, (0 missing)
##       GarageArea   < 552     to the left,  improve=0.16926840, (0 missing)
##       Fireplaces   < 0.5     to the left,  improve=0.08466878, (0 missing)
##   Surrogate splits:
##       YearBuilt    < 1998.5  to the right, agree=0.663, adj=0.298, (0
split)
##       YearRemodAdd < 1999.5  to the right, agree=0.663, adj=0.298, (0
split)
##       GarageArea   < 485     to the left,  agree=0.638, adj=0.245, (0
split)
##       BsmtFinSF1   < 843.5   to the left,  agree=0.617, adj=0.202, (0
split)
##       TotRmsAbvGrd < 7.5     to the left,  agree=0.617, adj=0.202, (0
split)
##
## Node number 6: 104 observations,    complexity param=0.01386976
##   mean=280562.4, MSE=4.17479e+09
##   left son=12 (65 obs) right son=13 (39 obs)
##   Primary splits:
##       BsmtFinSF1   < 867     to the left,  improve=0.19267050, (0 missing)
##       LotArea      < 11435.5 to the left,  improve=0.19105480, (0 missing)
##       GarageArea   < 688     to the left,  improve=0.17226570, (0 missing)
##       TotRmsAbvGrd < 7.5     to the left,  improve=0.11237020, (0 missing)
##       YearRemodAdd < 2002.5  to the left,  improve=0.03071225, (0 missing)
##   Surrogate splits:
##       BedroomAbvGr < 2.5     to the right, agree=0.673, adj=0.128, (0
split)
##       LotArea      < 18782.5 to the left,  agree=0.663, adj=0.103, (0
split)
##       TotRmsAbvGrd < 5.5     to the right, agree=0.635, adj=0.026, (0
split)
##
## Node number 7: 42 observations
##   mean=388831.3, MSE=1.05465e+10
##
## Node number 8: 323 observations
##   mean=125471, MSE=9.681281e+08
##
## Node number 9: 235 observations,    complexity param=0.01124072
##   mean=160423.5, MSE=1.324994e+09
##   left son=18 (58 obs) right son=19 (177 obs)
##   Primary splits:
##       GarageArea   < 340.5   to the left,  improve=0.2177341, (0 missing)
##       LotArea      < 7243    to the left,  improve=0.1736816, (0 missing)
##       TotRmsAbvGrd < 6.5     to the left,  improve=0.1468706, (0 missing)
##       YearRemodAdd < 1983.5  to the left,  improve=0.1407251, (0 missing)
##       BsmtFinSF1   < 687.5   to the left,  improve=0.1202763, (0 missing)
##   Surrogate splits:
##       YearBuilt    < 1949    to the left,  agree=0.791, adj=0.155, (0
split)
##       LotArea      < 1920    to the left,  agree=0.787, adj=0.138, (0
split)
##       YearRemodAdd < 1951.5  to the left,  agree=0.762, adj=0.034, (0
split)
##
## Node number 10: 102 observations
##   mean=189806.1, MSE=1.059912e+09
##
## Node number 11: 94 observations
##   mean=230860.1, MSE=2.594774e+09
##
## Node number 12: 65 observations
##   mean=258593.9, MSE=2.826342e+09
##
## Node number 13: 39 observations
##   mean=317176.6, MSE=4.277246e+09
##
## Node number 18: 58 observations
##   mean=130751.8, MSE=9.916224e+08
##
## Node number 19: 177 observations
##   mean=170146.5, MSE=1.051203e+09
```

```
fancyRpartPlot(dt_model)
```



Rattle 2023-Dec-15 17:56:01 ruthv

With a complexity parameter of 0.47785266, it is clear that overfitting and underfitting are not very prevalent.

The variables that have the biggest effects on sale price are YearBuilt, GarageArea, OverallQual, and BsmtFinSF1.

```
dt_predictions = predict(dt_model, data.frame(  LotArea = 10500,OverallQual =
5,YearBuilt = 1999,YearRemodAdd = 2010,BsmtFinSF1 = 700,  FullBath =
2,HalfBath = 1,BedroomAbvGr = 3, TotRmsAbvGrd = 10,Fireplaces = 1,GarageArea
= 305, YrSold = 2015))

cat("Tree model prediction for SalePrice is",dt_predictions, "Dollars")

## Tree model prediction for SalePrice is 125471 Dollars

threshold <- 100000

data$SalePriceBinary <- ifelse(data$SalePrice > threshold, 1, 0)

set.seed(123)

splitIndex <- createDataPartition(data$SalePriceBinary, p = 0.8, list =
FALSE)
train_data <- data[splitIndex, ]
test_data <- data[-splitIndex, ]

tree_model <- rpart(SalePriceBinary ~ ., data = train_data)
```

```
tree_predictions <- predict(tree_model, newdata = test_data)

binary_tree_predictions <- ifelse(tree_predictions > threshold, 1, 0)

levels_set <- c(0, 1)
binary_tree_predictions <- factor(binary_tree_predictions, levels =
levels_set)
test_data$SalePriceBinary <- factor(test_data$SalePriceBinary, levels =
levels_set)


TP <- sum(binary_tree_predictions == 1 & test_data$SalePriceBinary == 1)
TN <- sum(binary_tree_predictions == 0 & test_data$SalePriceBinary == 0)
FP <- sum(binary_tree_predictions == 1 & test_data$SalePriceBinary == 0)
FN <- sum(binary_tree_predictions == 0 & test_data$SalePriceBinary == 1)


accuracy <- (TP + TN) / sum(c(TP, TN, FP, FN))
cat("Accuracy for tree model:", accuracy, "\n")

## Accuracy for tree model: 0.07777778

sensitivity <- TP / (TP + FN)
cat("Sensitivity for tree model:", sensitivity, "\n")

## Sensitivity for tree model: 0

precision <- TP / (TP + FP)
cat("Precision for tree model:", precision, "\n")

## Precision for tree model: NaN

specificity <- TN / (TN + FP)
cat("Specificity for tree model:", specificity, "\n")

## Specificity for tree model: 1
```

Given the model's incredibly low accuracy, it is likely that most of its forecasts are off.

When the sensitivity is zero, it means that no positive cases were found by the model.

Since there are no real positives in this situation, precision is ill-defined.

Even while the model performs poorly overall, its specificity is excellent, which could indicate that it is not the right model for the task at hand.

Summary of Both Models

In every parameter, the LM performs better than the Tree Model, exhibiting increased precision, sensitivity, and accuracy.

Even though the Tree Model has a high specificity, it is less appropriate for the task due to serious problems with sensitivity, precision, and overall accuracy.

Classifying OverallQall to Class 1 and Class 0

```
data$OverallQualClass = ifelse(data$OverallQual >= 7, 1,  0)

features_class = c("LotArea", "OverallQual", "YearBuilt", "YearRemodAdd",
                    "BsmtFinSF1", "BedroomAbvGr", "TotRmsAbvGrd",
"Fireplaces", "GarageArea")

target_class = "OverallQualClass"

rf_model = randomForest(data[, features_class], data[, target_class])

## Warning in randomForest.default(data[, features_class], data[,
target_class]):
## The response has five or fewer unique values.  Are you sure you want to do
## regression?

class_predictions = predict(rf_model, newdata = data, type = "response")

class_predictions = ifelse(class_predictions >= 0.5, "Class 1", "Class 0")

result_table = data.frame(OverallQual = data$OverallQual, Classification =
class_predictions)

head(result_table)

##   OverallQual Classification
## 1           7        Class 1
## 2           6        Class 0
## 3           7        Class 1
## 4           7        Class 1
## 5           8        Class 1
## 6           5        Class 0
```

The class for each dataset data point is then predicted by the algorithm using the random forest model. Next, binary values (Class 1 or Class 0) are created from the forecasts. It is given a value of 1 in the event that the OverallQual characteristic is greater than or equal to 7 and 0 in the other case.