Predicting Real Estate Prices in the U.S.

11/26/2024

Payton Chen, Wilson Huang, Danny Nguyen, Lynna Nguyen, Richie Nguyen

Group 15

Introduction

Our project focuses on analyzing real estate data to understand the factors that influence housing prices and to build a model that can predict them. To understand and determine what factors contribute most to USA housing prices, we will analyze data from the "USA House Prices" provided on Kaggle. The output, or response variable, is the price of the property. The dataset includes predictor variables of date, bedrooms, bathrooms, sqft living, sqft lot, floors, waterfront, view, condition, sqft above, sqft basement, yr built, yr renovated, street, city, statezip, and country. For the purposes of our analysis, qualitative variables of street, city, statezip, and country will not be used; additionally, sqft basement will not be used since its values appear as non-numerical in the dataset. The following variables, along with their descriptions, will be used:

bedrooms: number of bedrooms in the property
bathrooms: number of bathrooms in the property
sqft_living: square footage of living area
sqft_lot: square footage of the lot size
floors: number of floors in the property
waterfront: binary indicator of whether the property has a waterfront view
view: an index from 0-4 representing the quality of the property's view
condition: an index from 1-5 representing the property's condition
sqft_above: square footage of the property above the basement
yr_built: year the property was built
yr_renovated: year the property was last renovated

The main question we aim to answer is: "What factors determine the price of a house, and can we predict real estate prices accurately based on these factors?" This project allows us to explore how housing prices are influenced and prepare for future decisions in real estate.

Methods & Results

- **Linear Regression Model** (Payton, Lynna)

  Given that our predictor is a quantitative variable, a linear regression model will be used in this section to predict the price outcome of houses. Linear regression models are used to determine a direct relationship between the predictor, housing price, and the other response variables. This model is advantageous given its simplicity and ease to interpret; this helps to determine how price changes with an increase or decrease in a single unit of each predictor. However, linear regression models assume that a direct, multicollinear relationship exists between the predictor and response variables and can be sensitive to outliers. Additionally, linear regression models are often prone to overfitting. We will select the most significant variables, remove outlier points as necessary, and perform cross-validation to optimize the accuracy of the linear regression model and obtain the lowest test error possible.

We begin by fitting an initial linear regression model on the dataset using all 11 predictors to compare against price. The initial linear regression model is follows:

In this equation, Y represents the predictor variable, price, represents the model intercept, $_i$ represents the intercept for each predictor, and X_i for i = 1, 2, …, 11 represents each of the predictor variables of bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, sqft_above, yr_built, and yr_renovated.

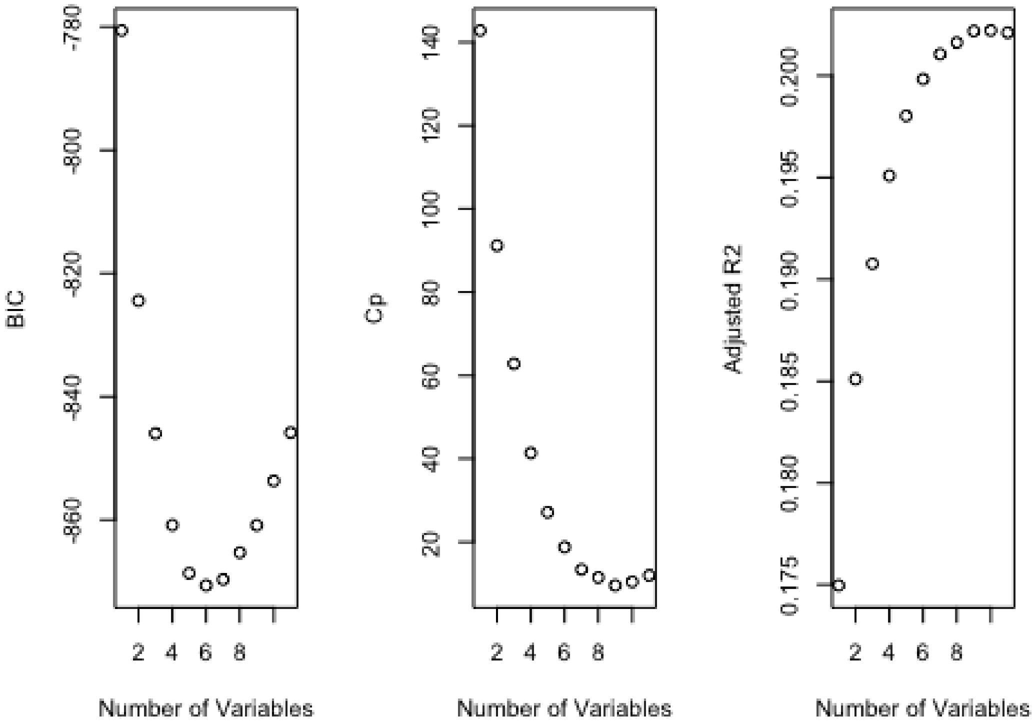
```
initial.lm = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
floors + waterfront + view + condition + sqft_above + yr_built +
yr_renovated, data = housing_data)
summary(initial.lm)
```

```
Call:
lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
    floors + waterfront + view + condition + sqft_above + yr_built +
    yr_renovated, data = housing_data)

Residuals:
     Min       1Q   Median       3Q      Max
-2172984  -129121   -16354    89761 26332337

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.433e+06  7.551e+05   5.871 4.67e-09 ***
bedrooms     -6.145e+04  1.155e+04  -5.320 1.09e-07 ***
bathrooms     5.915e+04  1.870e+04   3.164 0.001570 **
sqft_living   2.398e+02  2.396e+01  10.006  < 2e-16 ***
sqft_lot     -6.376e-01  2.333e-01  -2.733 0.006306 **
floors        3.364e+04  2.076e+04   1.620 0.105215
waterfront    3.225e+05  1.010e+05   3.192 0.001421 **
view          4.248e+04  1.188e+04   3.577 0.000351 ***
condition     3.095e+04  1.434e+04   2.158 0.030956 *
sqft_above    2.540e+01  2.376e+01   1.069 0.285017
yr_built     -2.297e+03  3.766e+02  -6.099 1.17e-09 ***
```

```
yr_renovated  6.542e+00  9.514e+00   0.688 0.491721
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 521400 on 4128 degrees of freedom
Multiple R-squared:  0.2042,    Adjusted R-squared:  0.2021
F-statistic: 96.31 on 11 and 4128 DF,  p-value: < 2.2e-16
```

A summary of details using the linear regression model with all 11 predictors was generated to reveal information about F-statistic p-value, F-statistic, t-statistic p-values, and multiple R^2. The F-statistic p-value of the initial linear regression model is less than $2.2x10^{-16}$, so we can reject the null hypothesis that . This means that at least one of the 11 predictors has a statistically significant relationship with price in this model. Additionally, the F-statistic is 96.31. The predictors bedrooms, bathrooms, sqft_living, sqft_lot, waterfront, view, condition, and yr_built all have t-statistic p-values of less than 0.05, indicating that they are statistically significant and their relationship to price is likely not due to chance. However, the predictors floors, sqft_above, and yr_renovated have p-values greater than this threshold, suggesting that these variables are not statistically significant and that their relationship to price may be due to chance. There is little evidence that floors, sqft_above, and yr_renovated are significant to predicting price if the other predictors are considered in the model. The multiple R^2 is 0.2042, which means that 20.42% of the variance can be explained by this full initial model. This is a relatively poor fit, so the three variables that were not statistically significant can be excluded from the model to improve simplicity, reduce multicollinearity, and avoid overfitting. A best subset selection will be performed to confirm the optimal number of variables to use for the best linear regression model.

```
regfit.full = regsubsets(price ~ bedrooms + bathrooms + sqft_living +
sqft_lot + floors + waterfront + view + condition + sqft_above + yr_built +
yr_renovated,
                            data = housing_data, nvmax = 11)
reg.summary = summary(regfit.full)
par(mfrow=c(1,3))

# subset regression selection plots
plot(reg.summary$bic, xlab="Number of Variables", ylab="BIC")
plot(reg.summary$cp, xlab="Number of Variables", ylab="Cp")
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="Adjusted R2")
```



```
reg.summary$outmat
```

```
          bedrooms bathrooms sqft_living sqft_lot floors waterfront view
1  ( 1 )  " "      " "       "*"         " "      " "    " "        " " 
2  ( 1 )  " "      " "       "*"         " "      " "    " "        " " 
3  ( 1 )  " "      " "       "*"         " "      " "    " "        "*" 
4  ( 1 )  "*"      " "       "*"         " "      " "    " "        "*" 
5  ( 1 )  "*"      "*"       "*"         " "      " "    " "        "*" 
6  ( 1 )  "*"      "*"       "*"         " "      " "    "*"        "*" 
7  ( 1 )  "*"      "*"       "*"         "*"      " "    "*"        "*" 
8  ( 1 )  "*"      "*"       "*"         "*"      "*"    "*"        "*" 
9  ( 1 )  "*"      "*"       "*"         "*"      "*"    "*"        "*" 
10  ( 1 ) "*"      "*"       "*"         "*"      "*"    "*"        "*" 
11  ( 1 ) "*"      "*"       "*"         "*"      "*"    "*"        "*" 
          condition sqft_above yr_built yr_renovated
1  ( 1 )  " "       " "        " "      " "         
2  ( 1 )  " "       " "        "*"      " "         
3  ( 1 )  " "       " "        "*"      " "         
4  ( 1 )  " "       " "        "*"      " "         
5  ( 1 )  " "       " "        "*"      " "         
6  ( 1 )  " "       " "        "*"      " "         
7  ( 1 )  " "       " "        "*"      " "         
8  ( 1 )  " "       " "        "*"      " "         
9  ( 1 )  "*"       " "        "*"      " "         
```

```
10  ( 1 ) "*"       "*"        "*"       " "
11  ( 1 ) "*"       "*"        "*"       "*"
```

To determine the best model that would provide the lowest test error, we can estimate using the BIC, C_p, and adjusted R^2 statistics for each number of variables used. These statistics were plotted against each possible number of variables used in subset models to visualize the model with the lowest BIC, lowest C_p, and highest adjusted R^2. Based on these plots, the models with either 7 or 8 predictors appear to be similar in these statistics, so exact BIC, C_p, and adjusted R^2 values were calculated to determine if a model using 7 or 8 predictors would be best.

```
bic_values = reg.summary$bic
print(bic_values)
```

```
 [1] -780.5944 -824.4208 -845.9551 -860.8391 -868.6496 -870.6273 -869.6672
 [8] -865.2820 -860.8594 -853.6670 -845.8127
```

```
cp_values = reg.summary$cp
print(cp_values)
```

```
 [1] 142.760093  91.195307  62.827661  41.373672  27.144051  18.813021
 [7]  13.442735  11.503732   9.605848  10.472842  12.000000
```

```
adjr2_values = reg.summary$adjr2
print(adjr2_values)
```

```
 [1] 0.1749664 0.1850979 0.1907592 0.1950892 0.1980269 0.1998273 0.2010569
 [8] 0.2016243 0.2021840 0.2022097 0.2021079
```

By comparing the exact BIC, C_p, and adjusted R^2 values, we determined that the best model to yield the lowest test error is the 7-variable model. A new model was fitted using bedrooms, bathrooms, sqft_living, sqft_lot, waterfront, view, and yr_built as the predictors.

```
price.lm = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
waterfront + view + yr_built, data = housing_data)
summary(price.lm)
```

```

Call:
lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
    waterfront + view + yr_built, data = housing_data)

Residuals:
     Min       1Q   Median       3Q      Max
-2171538  -133017   -18445    93794 26331765

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.708e+06  6.173e+05   7.626 2.99e-14 ***
bedrooms    -6.288e+04  1.148e+04  -5.479 4.54e-08 ***
```

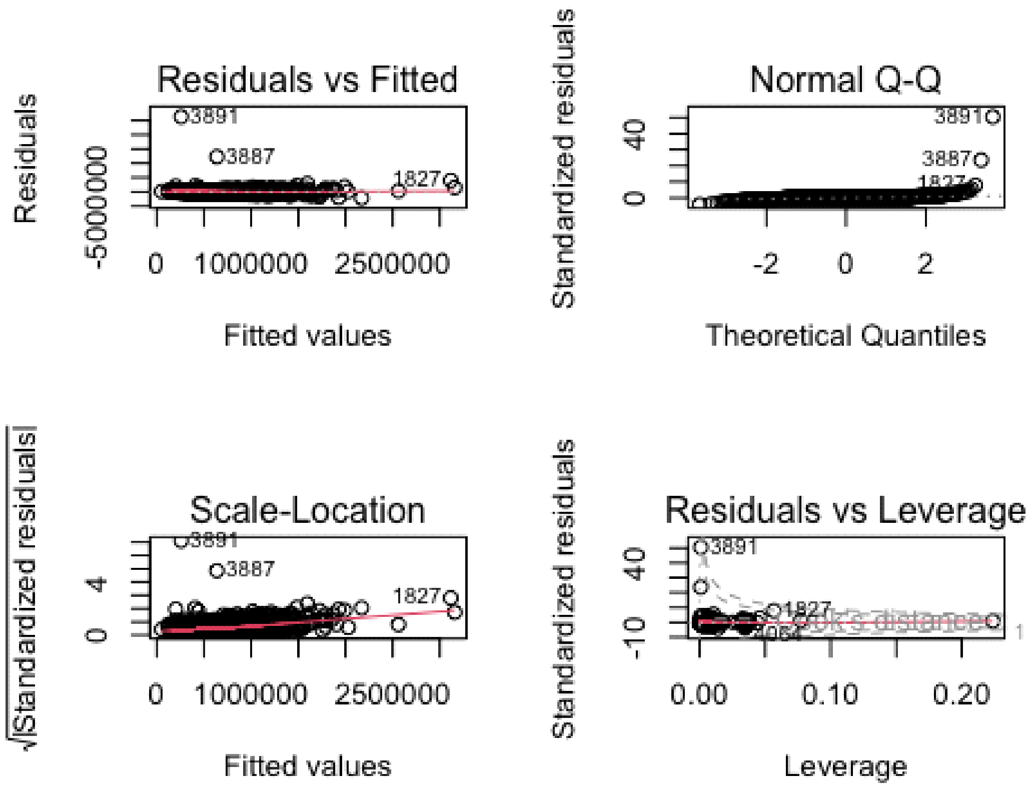
```
bathrooms    6.762e+04  1.778e+04   3.803 0.000145 ***
sqft_living  2.605e+02  1.451e+01  17.954  < 2e-16 ***
sqft_lot    -6.280e-01  2.315e-01  -2.713 0.006695 ** 
waterfront   3.207e+05  1.010e+05   3.174 0.001515 ** 
view         4.031e+04  1.167e+04   3.455 0.000555 ***
yr_built    -2.359e+03  3.177e+02  -7.424 1.37e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 521700 on 4132 degrees of freedom
Multiple R-squared:  0.2024,    Adjusted R-squared:  0.2011 
F-statistic: 149.8 on 7 and 4132 DF,  p-value: < 2.2e-16
```

From this summary, we obtain a fitted model with 7 variables that are all statistically significant to price and can specify the linear regression model formula as follows:

Looking at this 7-variable model, we can see that waterfront and bathrooms have the greatest positive relationship with housing price while bedrooms have the greatest negative relationship with price. Given that other predictors remain fixed, a single unit increase in each variable will result in the following changes to price: bedrooms decreases price by \$6.288x10^{4}, bathrooms increases price by \$6.762x10^{4}, sqft_living increases price by \$260.5, sqft_lot decreases price by \$0.628, waterfront increases price by \$3.207x10^{5}, view increases price by \$4.031x10^{4}, and yr_built decreases price by \$2.359x10^{3}.

Linear model plots were then generated for the 7-variable model to visualize linearity, normality, variance, and outliers.

```
par(mfrow=c(2,2))
plot(price.lm)
```



Looking at the Residual vs Fitted, Normal Q-Q, and Scale-Location plots, the assumption of linearity, normal distribution, and equal variance was somewhat consistent to the linear model. The Residual vs Fitted plot contains outlier observations 3891 and 3887 that are not consistent with the linear graph and could highly influence the model. Looking at the rest of the graphs, observations 3891, 3887, and 1827 appear as outliers and stand out in the majority of the plots, so they will be excluded from the dataset to increase model accuracy.

```
housing.new = housing_data[c(-3891, -3887, -1827),]
pricenew.lm = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
waterfront + view + yr_built, data = housing.new)
summary(pricenew.lm)
```

```

Call:
lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot + 
    waterfront + view + yr_built, data = housing.new)

Residuals:
     Min       1Q   Median       3Q      Max 
-2132968  -123303   -10030   103540  2914019 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  4.700e+06  2.996e+05  15.686  < 2e-16 ***
bedrooms    -5.688e+04  5.571e+03 -10.209  < 2e-16 ***
bathrooms    6.182e+04  8.636e+03   7.158 9.63e-13 ***
sqft_living  2.582e+02  7.112e+00  36.302  < 2e-16 ***
sqft_lot    -5.948e-01  1.123e-01  -5.297 1.24e-07 ***
waterfront   1.797e+05  4.976e+04   3.611 0.000309 ***
view         4.796e+04  5.674e+03   8.452  < 2e-16 ***
yr_built    -2.362e+03  1.542e+02 -15.319  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 253000 on 4129 degrees of freedom
Multiple R-squared:  0.5077,    Adjusted R-squared:  0.5069 
F-statistic: 608.3 on 7 and 4129 DF,  p-value: < 2.2e-16
```

In this revised 7-variable model with omitted outliers, we obtain a better fitted formula as follows:

The F-statistic p-value is still less $2.2\text{x}10^{-16}$, so at least one of the 7 included predictors is statistically significant to price in this model. The new F-statistic is 608.3, which is a very significant increase from both the full initial model and the 7-variable model that still included outliers; this indicates the 7-variable model with omitted outliers is performing better at explaining the variance in the predictors relative to price. The revised multiple R^2 is 0.5077, which means 50.77% of the variance can be explained by this model. This is an over 2-fold increase from both the full initial and 7-variable models, indicating that removing statistically insignificant variables and outliers resulted in a more accurate model that will yield a lower test error.

Even though R^2 is not directly proportional to the training error, they are relative to each other when analyzing datasets and their models. In this modified model, we observed a multiple R^2 value that was 0.3035 units higher than the original model and showed a lower training error, which can be expected due to the exclusion of these 3 outliers that heavily influenced the model and utilizing the C_p, BIC, and adjusted R^2. However, training error is not a good representation of test error due to overfitting or underfitting and bias-variance tradeoff that can happen. To get a better estimate of test error and its prediction accuracy, we proceeded to use pricenew.lm and used the cross-validation technique to get a better estimate. First, we split our modified data set in a randomized 80% training set and 20% test set. We cross-validated 10 times and then calculated the average mean squared error (MSE) across all 10 folds.

```
MSE = rep(0,10)
for (i in 1:10) {
  set.seed(i)
  train = sample(1:nrow(housing.new), 0.8*nrow(housing.new))
  test = housing.new[-train,]
  price.lm = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
waterfront + view + yr_built, data = housing.new,
                subset = train)
  yhat = predict(price.lm, newdata = test)
  MSE[i] = mean((yhat-test$price)^2)
}
MSE
```

```
 [1] 64231060268 66262433128 67081336441 57768140581 61791863437 64999875271
 [7] 68124206795 51907916458 66694604325 62553853429
```

```
mean(MSE)
```

```
[1] 63141529013
```

The average test MSE we obtained was 63,141,529,013 across all 10 folds of training and testing. MSE is the mean squared error that is used to evaluate the average squared difference between the predicted values in our model and the actual values in our testing subset. Usually, a more accurate model has a lower MSE that is closer to 0, yet the test MSE we obtained from the cross validation is on average 63,141,529,013 with it ranging from 51,907,916,458 to 68,124,206,795. The RMSE is a better metric to interpret the data with our RMSE being 251,279.78, meaning that the model's prediction for housing price on average is off by $251,279.78. Overall, the model, pricenew.lm, is not very accurate in predicting the price of houses and linear regression is not the best model to use. This is expected considering the many factors involved in housing price and the likelihood that an entirely multicollinear relationship does not exist between the predictors in real life, which would and did result in a poor linear model.

- **Regression Tree Model** (Wilson, Danny, Richie)

  Regression trees are a helpful tool for predicting outcomes, especially when there are nonlinear relationships or interactions between variables. They work by splitting the data into smaller groups based on feature values, forming a tree-like structure that is easy to understand and visualize. One great thing about regression trees is that they don't need much data preparation—they can handle both numbers and categories without scaling or special formatting. The biggest advantages of regression trees are their simplicity and interpretability. You can easily see how the model makes decisions by looking at the tree. They also handle nonlinear patterns well, making them more flexible than linear models. Additionally, they are less affected by outliers, as splits focus on minimizing the variance within groups. However, regression trees have some downsides. They can overfit the data if the tree becomes too detailed, which makes them less accurate on new data. Pruning can help reduce overfitting but adds extra work. They are also usually less accurate than more advanced methods like random forest.

Model Formula:
price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors + waterfront + view + condition + sqft_above + yr_built + yr_renovated

Our thought process while fitting the model was understanding the dataset's structure and cleaning the data. We made sure to check for outliers in the dataset, primarily in important variables such as price, square foot living, and bedrooms since these variables will be extremely helpful in predicting the housing market. We also analyze how numerical predictors and price correlates to see what predictors are strong. While looking at the dataset some variables we thought to exclude were street, city, country, and state zip from the model because they are not variables that will directly contribute to the house price prediction in the upcoming years, additionally we thought that the variables sqft living, sqft above, and sqft basement would cause some overlapping when it came to the information, causing us to consider whether using all three would be a good idea.

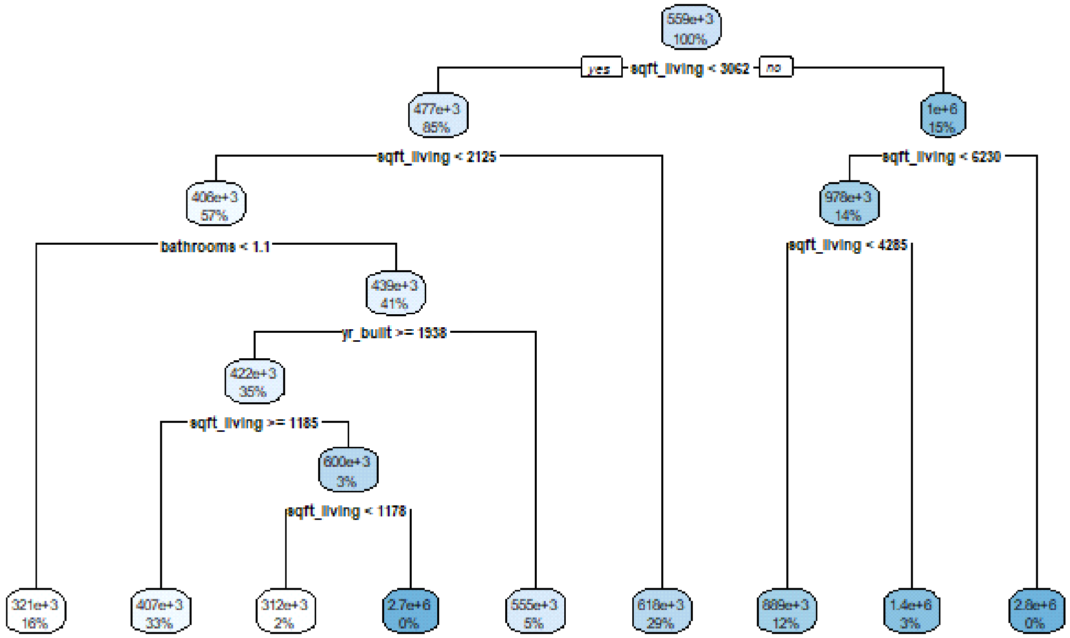
Firstly, we decided to split the data into two parts for training and testing.

```
set.seed(1)
n <- nrow(housing_data)
train_indices <- sample(1:n, size = 0.8 * n)
train_data <- housing_data[train_indices, ]
test_data <- housing_data[-train_indices, ]
```

We then fit the regression tree model on the training with prune control.

```
reg_tree <- rpart(price ~ bedrooms + bathrooms + sqft_living + sqft_lot
+
                 waterfront + view + yr_built,
                  data = train_data, method = "anova",
                  control = rpart.control(minsplit = 20, cp = 0.01,
maxdepth = 10))
```

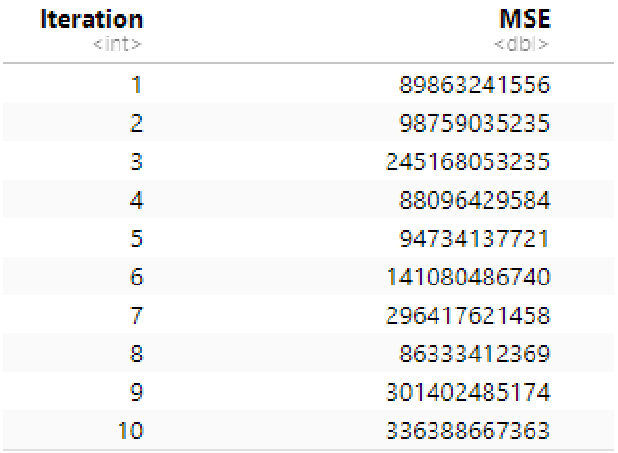
Decision Tree Diagram



From a single iteration of training our model using the test set, we found that it had an MSE of 89,863,241,556.

```
y_pred <- predict(reg_tree, test_data)
mse <- mean((y_pred - test_data$price)^2)
```

After repeating this method over 10 different iterations, we obtained these MSE values:



Iteration <int>	MSE <dbl>
1	89863241556
2	98759035235
3	245168053235
4	88096429584
5	94734137721
6	141080486740
7	296417621458
8	86333412369
9	301402485174
10	336388667363

From these values, the mean MSE is 177,824,357,043, and the RMSE is 133,345. This shows us that the regression tree model's prediction for housing price on average is off by $133,345.

```
for (i in 1:10) {
  set.seed(i)
  train_indices <- sample(1:n, size = 0.8 * n)
  train_data <- housing_data[train_indices, ]
```

```r
  test_data <- housing_data[-train_indices, ]

  reg_tree <- rpart(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
                    waterfront + view + yr_built,
                    data = train_data, method = "anova",
                    control = rpart.control(minsplit = 20, cp = 0.01,
maxdepth = 10))

  y_pred <- predict(reg_tree, test_data)

  mse_values[i] <- mean((y_pred - test_data$price)^2)
}
```

- **Random Forest and Boosting**

The original plan was to compare the regression tree with the linear regression model to assess whether a tree-based method could better capture the complexities of predicting house prices. However, linear regression assumes a strictly linear relationship between predictors and the target variable, which might not fully represent the dataset's structure. To further demonstrate that linear regression is not always the best choice, we also tested random forest and boosting, which are ensemble methods that combine multiple trees for more robust predictions.

Random forest builds a collection of decision trees by using different subsets of the data and averaging their predictions, reducing overfitting and improving stability. Boosting, on the other hand, sequentially trains trees, each focusing on correcting the errors of the previous ones, allowing it to model complex patterns and reduce bias. By including these additional models, we can highlight the strengths of ensemble methods over simpler approaches like linear regression and individual regression trees, ensuring a thorough evaluation of the dataset. The following sample code was created for boosting:

```r
boost_model <- gbm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
                   waterfront + view + yr_built,
                   data = train_data,
                   distribution = "gaussian",
                   n.trees = 1000,
                   interaction.depth = 4,
                   shrinkage = 0.01,
                   n.minobsinnode = 10,                    cv.folds = 5)
```

Boosting showed competitive performance, closely rivaling random forest. It captured complex relationships in the data but had a slightly higher RMSE of about 125,739 than the random forest model. The following sample code was created for random forest:

```
rf_model <- randomForest(price ~ bedrooms + bathrooms + sqft_living +
sqft_lot + waterfront + view + yr_built,
data = train_data, ntree = 500, mtry = 3)
```

Random forest significantly outperformed the regression tree by reducing variability and improving accuracy due to its ensemble structure. It produced the lowest RMSE of about 122,082, indicating its strong predictive capability.

Conclusion

The regression tree outperformed the linear regression model, achieving an RMSE of 133,345 compared to Linear Regression's RMSE of 251,276. These values indicated that the linear regression model was off, on average by $251,276 for housing prices, while the regression tree was only off by $133,345 on average. This demonstrates that the tree-based model can better capture non-linear relationships in the data, providing improved predictive accuracy. However, the regression tree showed higher variability across iterations and is prone to overfitting, making it less robust overall. Linear regression, while consistent and interpretable, struggled due to its strict assumption of linearity, which may not fully represent the complex interactions in the dataset. Other methods, such as random forest and boosting, can offer even greater accuracy and robustness by leveraging ensemble techniques, making them valuable options for future consideration.

References

Özcan, F. (2024, July 21). *USA House Prices*. Kaggle. https://www.kaggle.com/datasets/fratzcan/usa-house-prices