STATS Final Markdown Report

Darina Kamikazi and Chia Han Lee

2023-03-17

Table of Contents

[1.INTRODUCTION 1](#_Toc129984338)

[2. DATA AND METHODOLOGY 2](#_Toc129984339)

[a. Our Data 2](#_Toc129984340)

[Summary Statistics 2](#_Toc129984341)

[Variables of interest 3](#_Toc129984342)

[b. Statistical Method 3](#_Toc129984343)

[c. Methodology 4](#_Toc129984344)

[i. Data Cleaning 4](#_Toc129984345)

[Plot 1: Original vs. Filtered data 4](#_Toc129984346)

[Summary Statistics of Variables of Interest After Outlier Removal 4](#_Toc129984347)

[ii. Statistical Modeling 5](#_Toc129984348)

[3. RESULTS AND INTERPRETATION 5](#_Toc129984349)

[Variable importance Plot 5](#_Toc129984350)

[Barplots showing the split by dates, months and years. 7](#_Toc129984351)

[3. CONCLUSION 9](#_Toc129984352)

[fit a random forest model to our training set 9](#_Toc129984353)

[Work Cited 9](#_Toc129984354)

[APPENDIX: The Code 9](#_Toc129984355)

# 1.INTRODUCTION

For our project, my groupmate Chia Han Lee created a Denver housing price prediction model using Random Forest regression in R. Specifically, our guiding research question was “what predicts the price of a single-value house in Denver?” We were interested in this question because, as we are both international students who moved to Denver for this program, we noticed that it was hard to find affordable housing in Denver and naturally wondered why that was. However, being able to closely estimate the price of a house can be very useful not just to us internationals, but to any real estate stakeholders. Realtors and homeowners looking to sell houses could have a more accurate measure of its predicted worth, and people looking to become homeowners could use the price estimate to figure out if it would be a fair deal for them or not.

# 2. DATA AND METHODOLOGY

## a. Our Data

We obtained our dataset from [Kaggle](https://github.com/dannycrief/PythonDataMining/blob/main/data/single_family_home_values.csv). It contains 15,000 houses in the Denver area that were built between 1998 and 2016. The dataset also contains information about the physical attributes of each of those houses as well as information which help paint a picture of the perceived value of the houses on the market. In total, our dataset has 15,000 observations, and 18 different variables.

### Summary Statistics

The summary statistics are shown below:

id address city state   
 Min. : 143367 Length:11280 Length:11280 Length:11280   
 1st Qu.: 8665808 Class :character Class :character Class :character   
 Median : 21834074 Mode :character Mode :character Mode :character   
 Mean : 44996788   
 3rd Qu.: 46262688   
 Max. :251527407   
 zipcode latitude longitude bedrooms   
 Min. :80022 Min. :39.61 Min. :-105.1 Min. : 0.000   
 1st Qu.:80205 1st Qu.:39.73 1st Qu.:-105.0 1st Qu.: 2.000   
 Median :80206 Median :39.75 Median :-105.0 Median : 3.000   
 Mean :80205 Mean :39.74 Mean :-105.0 Mean : 2.684   
 3rd Qu.:80207 3rd Qu.:39.76 3rd Qu.:-104.9 3rd Qu.: 3.000   
 Max. :80209 Max. :39.85 Max. :-104.9 Max. :15.000   
 bathrooms rooms squareFootage lotSize   
 Min. : 0.000 Min. : 0.000 Min. : 350 Min. : 832   
 1st Qu.: 1.000 1st Qu.: 5.000 1st Qu.: 979 1st Qu.: 4462   
 Median : 2.000 Median : 6.000 Median : 1240 Median : 5950   
 Mean : 2.208 Mean : 6.117 Mean : 1483 Mean : 5700   
 3rd Qu.: 3.000 3rd Qu.: 7.000 3rd Qu.: 1700 3rd Qu.: 6250   
 Max. :12.000 Max. :39.000 Max. :10907 Max. :97125   
 yearBuilt lastSaleDate lastSaleAmount priorSaleDate   
 Min. :1874 Length:11280 Min. : 500 Length:11280   
 1st Qu.:1906 Class :character 1st Qu.: 224000 Class :character   
 Median :1925 Mode :character Median : 345000 Mode :character   
 Mean :1929 Mean : 424277   
 3rd Qu.:1949 3rd Qu.: 487000   
 Max. :2016 Max. :45600000   
 priorSaleAmount estimated\_value   
 Min. : 0 Min. : 147767   
 1st Qu.: 110000 1st Qu.: 401737   
 Median : 210000 Median : 516018   
 Mean : 259446 Mean : 631543   
 3rd Qu.: 330120 3rd Qu.: 679550   
 Max. :16000000 Max. :10145310

### Variables of interest

Since we were interested in finding what factors affect the price of a house in Denver , we chose the “$estimated\_value” as our independent variable, as it showed the respective price estimates of the houses. For our dependent variables, we chose 8 variables that contained information about both the physical value of the houses (based on their physical attributes), as well as their respective perceived value (based on their performance on the market): and respectively reveal the lot size and of the houses, and , and respectively reveal the number of bedrooms, bathrooms and rooms of the houses. Similarly, and : respectively reveal the last sale amount of a house and its sale amount before that.

## b. Statistical Method

Our statistical method of choice to predict the value of Denver houses was the random forest method in R.We built our model using our independent variables(i.e. on the perceived and physical value of a house), and evaluated its accuracy using a test set. However, to understand the random forest method, we must first understand decision trees.

[Decision trees](https://www.seldon.io/decision-trees-in-machine-learning" \l ":~:text=Decision%20trees%20are%20an%20approach,categorise%20or%20classify%20an%20object) are a common supervised machine learning method makes predictions on both categorical(ie classification trees) and numeric data(ie regression trees) in a tree-like structure that represents the decision-making process through a tree-like structure. Although they are easy to understand and straightforward, there are two main issues that arise when using decision trees: as they are trained on particular variables and very close to the original data, they are prone to both bias and overfitting, respectively.

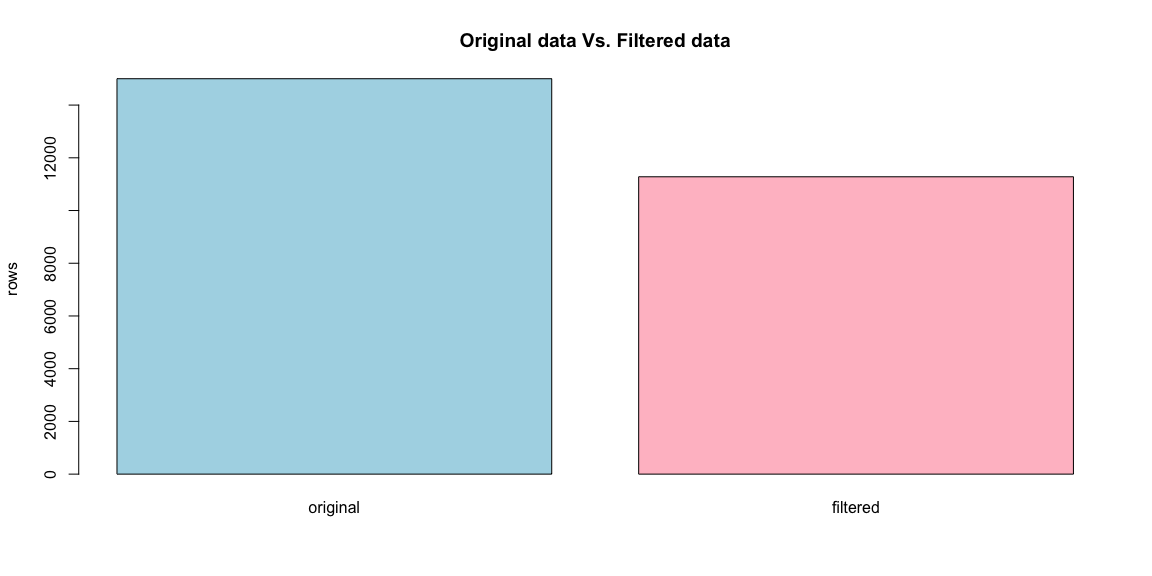
In contrast, [Random Forest](https://towardsdatascience.com/understanding-random-forest-58381e0602d2) is a statistical analysis method that aggregates the predictions of individual decision trees to make a combined classification or regression prediction. To, new datasets are created by randomly sampling(with replacement) from the original dataset(i.e. bootstrapping). From there, independent decision trees are trained on subsets of randomly sampled variables from the bootstrapped datasets. Finally, a new set of datapoints is run through the decision trees, and the final decision is made by aggregating the trees’ respective predictions(ie ensemble learning). The process of bootstrapping then aggregating trees is referred to as bagging. ([Normalized Nerd](https://www.youtube.com/watch?v=v6VJ2RO66Ag)).Randomizing the random forest building process makes it more accurate than decision trees in [two main ways](https://www.youtube.com/watch?v=gkXX4h3qYm4): randomized bootstrapping makes the model less sensitive to training data because it ensures that each tree is not relying on the same data. This, in turn, reduces overfitting. Just as well, random feature selection ensures there is low to no correlation among decision trees so, in case there is bias introduced in one,leading to a faulty prediction, it is less likely to affect the prediction of other trees. The random forest method can be applicable to various situations and across [various industries](https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/) such as finance and business, healthcare and banking.

## c. Methodology

### i. Data Cleaning

We relied on complete case analysis,so we filtered out observations that had missing values in any of our variables of interest. Only one of our independent variables, namely $priorSaleAmount, was significantly affected (with 3,713 missing observations). After filtering missing values, our filtered dataset contained 70% of the original dataset, which we believed, was still enough to accurately build a random forest on.

#### Plot 1: Original vs. Filtered data

Here is a plot of our original vs filtered data 

As part of our data cleaning process, we also removed outliers. As we were interested in predicting the price of single family houses in Denver, we focused on typical single-family house attributes. Houses that had attributes that were too extreme( for example those with more than 10 rooms) were filtered out.

#### Summary Statistics of Variables of Interest After Outlier Removal

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 832 4462 5950 5700 6250 97125

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.000 2.000 3.000 2.684 3.000 15.000

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.000 1.000 2.000 2.208 3.000 12.000

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.000 5.000 6.000 6.117 7.000 39.000

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 350 979 1240 1483 1700 10907

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 500 224000 345000 424277 487000 45600000

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0 110000 210000 259446 330120 16000000

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 147767 401737 516018 631543 679550 10145310

In order for a random forest to be useful, it is imperative to have variables that have some [predictive power](https://www.tandfonline.com/doi/full/10.1080/21642583.2014.956265#:~:text=4.-,Random%20forest,(2001)) and can influence our variable of interest. Our 8 independent variables each have the potential to predict the price of a house and, after our minimal data cleaning process, our dataset was ready to be run through a random forest model.

### ii. Statistical Modeling

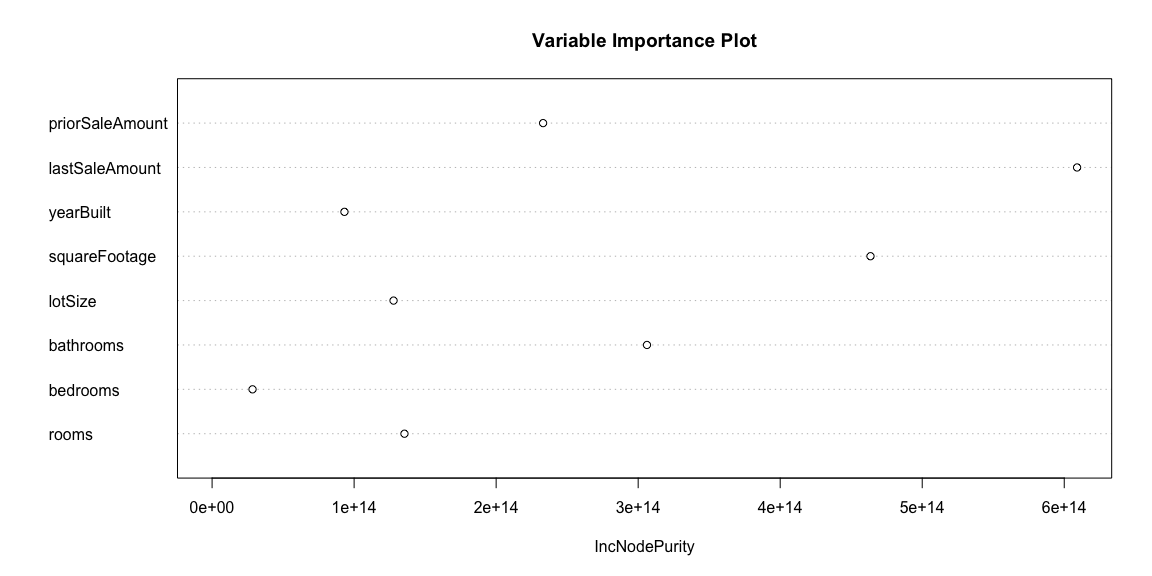
Because of random sampling with replacement, about one third of the original data usually does not make it into any of the bootstrapped datasets. This data is referred to as [an out-of-bag](https://www.edureka.co/blog/random-forest-classifier/) dataset. To replicate this in our random forest model, we split our dataset into two, where 70% was used as a training dataset and 30% was used as our out-of-bag dataset on which we could test our model’s accuracy. In building a random forest model, there are [several parameters](https://www.analyticsvidhya.com/blog/2021/03/introduction-to-random-forest-and-its-hyper-parameters/) to consider (eg: node size, number of trees, tree depths etc. For our project, we decided to go with default parameters provided in R. After building our model, we used the [Mean Average Error](https://c3.ai/glossary/data-science/mean-absolute-error/#:~:text=What%20is%20Mean%20Absolute%20Error,true%20value%20of%20that%20observation.) (MAE) estimator to test its accuracy on our out-of-bag samples. The MAE estimator calculates the average difference between the true value of an observation and the value predicted by a model. In other words, it is a value that indicates the ability of a model to make accurate predictions.

# 3. RESULTS AND INTERPRETATION

To determine the most significant predictor of our variable of interest, we used the [variable](https://medium.com/the-artificial-impostor/feature-importance-measures-for-tree-models-part-i-47f187c1a2c3#:~:text=Gini%20Importance%20or%20Mean%20Decrease%20in%20Impurity%20(MDI)%20calculates%20each,number%20of%20samples%20it%20splits.) importance estimator, and later visualized it in a variable importance plot. Variable importance is an indicator that uses the [Gini measure](https://www.baeldung.com/cs/ml-feature-importance#:~:text=Feature%20(variable)%20importance%20indicates%20how,a%20current%20model%20and%20prediction.) to determine the relevance/usefulness a given variable has on a model’s prediction. From our variable importance plot, we found that the houses’ most recent performance on the market and their square footage, respectively measured by the $lastSaleAmount and $squareFootage variables, were the most important predictors of their price.

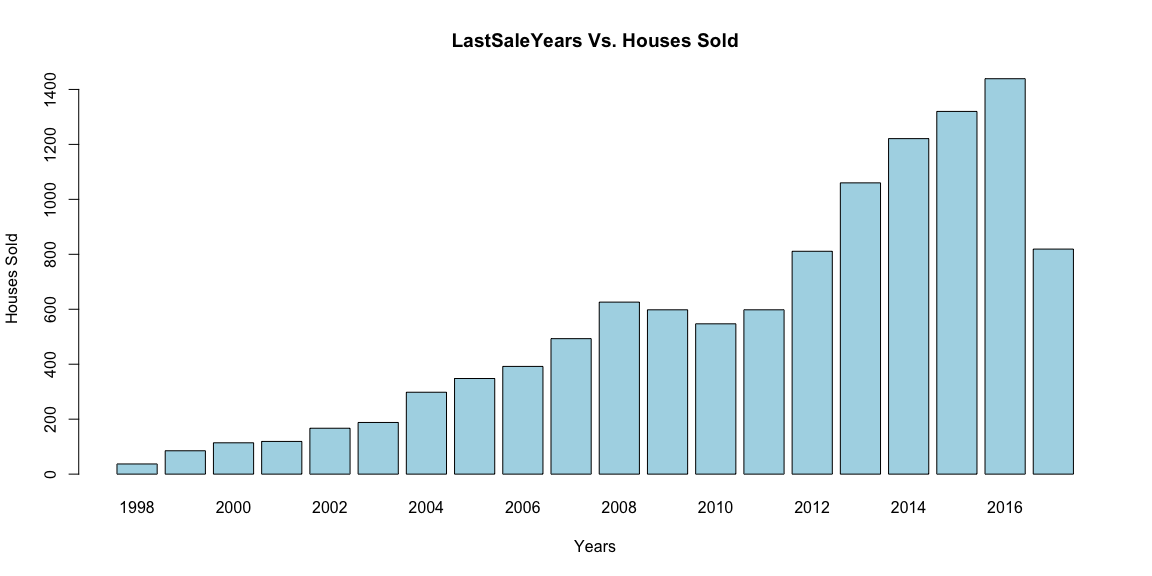
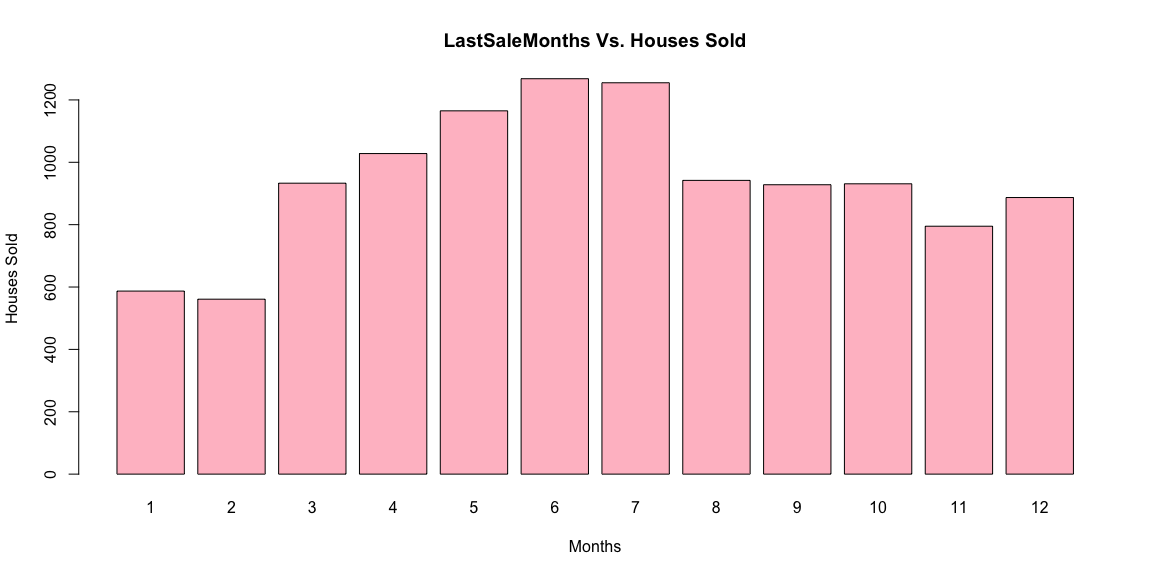
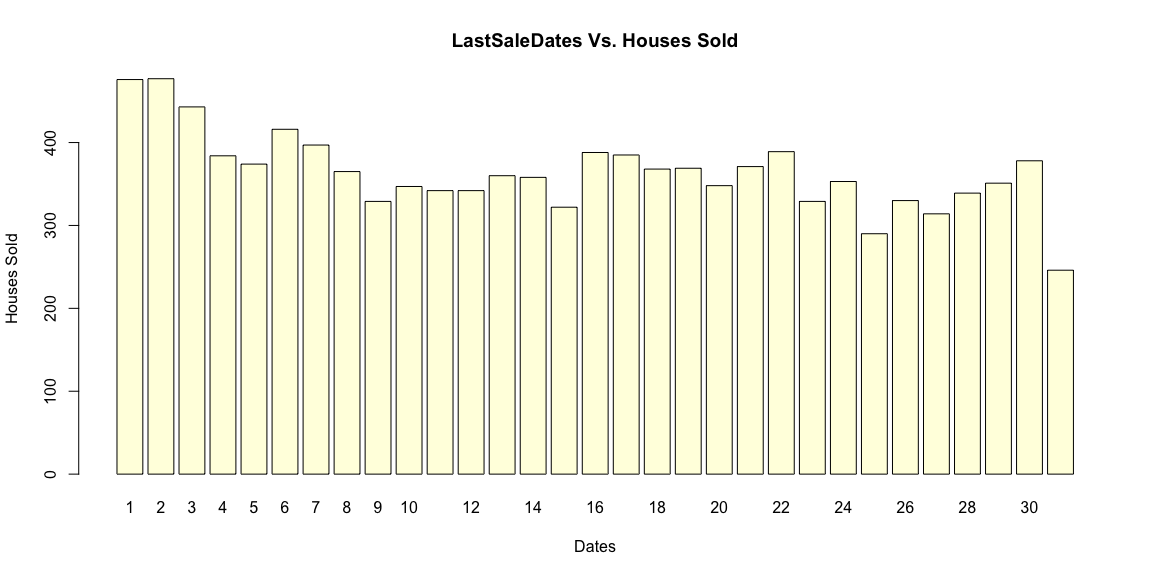
### Variable importance Plot

$test  
[1] 3383 18  
  
$train  
[1] 7897 18



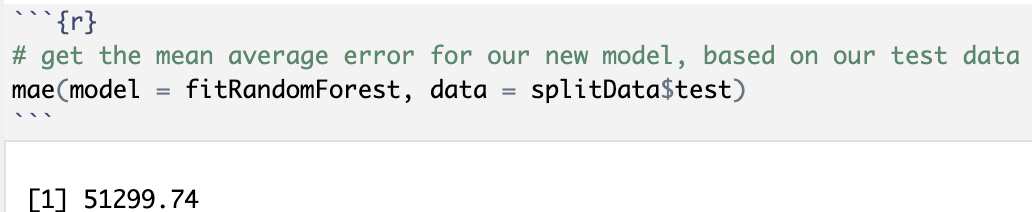
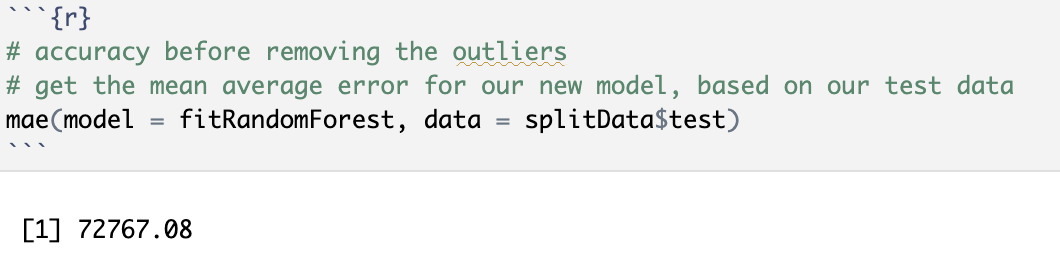
Since the variable was deemed the most important predictor, we further analyzed it and visualized the trends in the houses’ most recent sale dates,months, and years. Although we noticed a gradually positive trend in house sales throughout the years, it was hard to conclusively evaluate the true nature of the variable’s contribution to our model’s predictions.

### Barplots showing the split by dates, months and years.



# 3. CONCLUSION

The findings of our model indicate that, overall, the prior performance of a single-family home on the Denver housing market is a more important predictor of its estimated value than any of its physical attributes. However, we acknowledge that our model had three main limitations: First, as stated in our project’s design, we filtered out big houses and houses with no prior sale amounts(ie new houses). Similarly, we relied on the default parameters that come with the random forest package in R. We predict that both of these limitations, although intentational,might negatively affect the accuracy of our model on future house price prediction. In terms of the outlier limitation,however,we found that removing outliers reduced our MAE value and improved the accuracy of our model’s predictions. Since we wanted our dataset to be more representative of the typical single-family house in Denver, so we believe that removing outliers made our model able to fit more representative data and, therefore, increased its accuracy.



In the future, our project and model could be improved by fine tuning our random forest parameters, keeping or analyzing further the missing values of our dataset. Just as well, future model analysis could benefit from exploring further the nature of each independent variable’s importance to our model’s prediction.

### 

### Work Cited

For the Research

Lee. “Feature Importance Measure for Tree Models- part 1”.Published: October 2017. Accessed: March 2023.Url: <https://medium.com/the-artificial-impostor/feature-importance-measures-for-tree-models-part-i-47f187c1a2c3#:~:text=Gini%20Importance%20or%20Mean%20Decrease%20in%20Impurity%20(MDI)%20calculates%20each,number%20of%20samples%20it%20splits>.

Ce.ai. “Mean Absolute Error(MAE)”. Accessed: March 2023. Url: <https://c3.ai/glossary/data-science/mean-absolute-error/#:~:text=What%20is%20Mean%20Absolute%20Error,true%20value%20of%20that%20observation>.

Zvornicanin. “What is Feature Importance in Machine Learning?”. Published: November 2022. Accessed: March 2023. Url: <https://www.baeldung.com/cs/ml-feature-importance#:~:text=Feature%20(variable)%20importance%20indicates%20how,a%20current%20model%20and%20prediction>.

Mbaabu. “Introduction to Random Forest in Machine Learning.” Published: December 2020. Accessed: March 2023. Url: <https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/>

Yiu. “Understanding Random Forest”. Published: June 2019. Accessed: March 2023. Url: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

Klaise. “Decision Trees in Machine Learning Explained”. Published: November 2021. Accessed: March 2023. Url: <https://www.seldon.io/decision-trees-in-machine-learning#:~:text=Decision%20trees%20are%20an%20approach,categorise%20or%20classify%20an%20object>

Lateef. “A Comprehensive Guide to Random Forest in R”. Published: November 2020. Accessed: March 2023. Url: <https://www.edureka.co/blog/random-forest-classifier/>

Singh. “Introduction to Random Forest and Its Hyper-parameters”. Published: April 2021. Accessed: March 2023. Url: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-random-forest-and-its-hyper-parameters/>

IBM Technology YouTube video. “What is Random Forest?” Published: February 2022. Accessed: March 2023. Url: <https://www.youtube.com/watch?v=gkXX4h3qYm4>

Normalized Nerd YouTube video. “Random Forest Algorithm Clearly Explained!” Published: April 2021. Accessed: March 2023. Url: <https://www.youtube.com/watch?v=gkXX4h3qYm4>

For the code

<https://www.kaggle.com/code/sivantm/melbourne-house-type-prediction-using-r>

<https://eagronin.github.io/housing-forecast-acquire/>

<https://www.kaggle.com/code/rtatman/welcome-to-data-science-in-r>

# APPENDIX: The Code

knitr::opts\_chunk$set(fig.align="center",  
 fig.height=6,  
 fig.width=12,  
 warning = FALSE,  
 message = FALSE,  
 comment = NA,  
 echo=FALSE)  
#install.packages("ISLR")  
  
library(tidyverse)# dplyr::tibble  
library(rpart) # for regression trees  
library(randomForest) # for random forests  
library(caret)  
library(dummy) # one-hot encoding   
library(nnet) # simple neural network package  
library(modelr)  
library(knitr) #allows you to create Appendix with all\_labels()  
library(tidyverse)  
library(ISLR) #Auto dataset  
theme\_set(theme\_bw()) #sets default ggplot output style  
#Load Data  
den\_data <- read.csv("/Users/dadou/Desktop/single\_family\_home\_values.csv")  
or\_data <- nrow(den\_data)  
# checking the presence of missing values  
sapply(den\_data, function(x) sum(is.na(x)))  
# dropping the missing values   
den\_data <- na.omit(den\_data)  
# checking if all the missing values were dropped  
sapply(den\_data, function(x) sum(is.na(x)))  
af\_data <- nrow(den\_data)  
compare <- c(or\_data,af\_data)  
  
#View summary statistics  
summary(den\_data)  
af\_data <- nrow(den\_data)  
compare <- c(or\_data,af\_data)  
barplot(compare, main="Original data Vs. Filtered data", col=c("lightblue","pink"),names.arg = c("original","filtered"), ylab="rows")  
#View Plot  
#Lot size  
summary(den\_data$lotSize)  
  
#Bedrooms  
summary(den\_data$bedrooms)  
  
#Bathrooms  
summary(den\_data$bathrooms)  
  
#Rooms   
summary(den\_data$rooms)  
  
#Square footage  
summary(den\_data$squareFootage)  
#Last Sale Amount  
summary(den\_data$lastSaleAmount)  
  
#Prior Sale Amount  
summary(den\_data$priorSaleAmount)  
  
#Estimated Value  
summary(den\_data$estimated\_value)  
# split our data so that 30% is in the test set and 70% is in the training set  
splitData <- resample\_partition(den\_data, c(test = 0.3, train = 0.7))  
  
# how many cases are in test & training set?   
lapply(splitData, dim)  
fitRandomForest <- randomForest(estimated\_value ~ rooms + bedrooms + bathrooms + lotSize + squareFootage +  
 yearBuilt + lastSaleAmount + priorSaleAmount, data = splitData$train)  
  
randomForest::varImpPlot(fitRandomForest,  
 sort=FALSE,  
 main="Variable Importance Plot")  
# checking the format of the date  
prior\_date = den\_data$priorSaleDate  
last\_date = den\_data$lastSaleDate  
#head(prior\_date)  
#head(last\_date)  
# make the priorsaledate into year month and date independently  
datetxt <- den\_data$priorSaleDate  
datetxt <- as.Date(datetxt,format="%Y-%m-%d")  
df <- data.frame(pr\_day = as.numeric(format(datetxt, format = "%d")),  
 pr\_month = as.numeric(format(datetxt, format = "%m")),  
 pr\_year = as.numeric(format(datetxt, format = "%Y")))  
#head(df)  
den\_data = cbind(df,subset(den\_data,select = -c(priorSaleDate)))  
#head(den\_data)  
# make the lastsaledate into year month and date independently  
datetxt <- den\_data$lastSaleDate  
datetxt <- as.Date(datetxt,format="%Y-%m-%d")  
df <- data.frame(ls\_day = as.numeric(format(datetxt, format = "%d")),  
 ls\_month = as.numeric(format(datetxt, format = "%m")),  
 ls\_year = as.numeric(format(datetxt, format = "%Y")))  
#head(df)  
den\_data = cbind(df,subset(den\_data,select = -c(lastSaleDate)))  
#head(den\_data)  
#By day  
barplot(table(den\_data$ls\_day), main="LastSaleDates Vs. Houses Sold", col="lightyellow",xlab = "Dates", ylab="Houses Sold")  
#By month  
barplot(table(den\_data$ls\_month), main="LastSaleMonths Vs. Houses Sold", col="pink",xlab = "Months", ylab="Houses Sold")  
#By year  
barplot(table(den\_data$ls\_year), main="LastSaleYears Vs. Houses Sold", col="lightblue",xlab = "Years", ylab="Houses Sold")  
# split our new data so that 30% is in the test set and 70% is in the training set  
splitData<- resample\_partition(den\_data, c(test = 0.3, train = 0.7))  
  
# how many cases are in test & training set?   
#lapply(splitData, dim)  
fitRandomForest <- randomForest(estimated\_value ~ rooms + bedrooms + bathrooms + lotSize + squareFootage +  
 yearBuilt + lastSaleAmount + priorSaleAmount, data = splitData$train)  
  
randomForest::varImpPlot(fitRandomForest,  
 sort=FALSE,  
 main="Variable Importance Plot")