Madison Single Family Housing

Ira Sharenow

2023-03-05

## Introduction

This document uses linear regression, random forests, and boosting to analyze Madison Wisconsin single family home sales, 1994-2023.

In this document, I used the R programming language for data cleaning and manipulation as well as for implementing the data science techniques.

The charts were created using the ggplot2 package. The tables were created using the flextable package.

I have also created a number of charts using Tableau and summary pivot tables and charts with Excel. Please visit my website for the links. <https://irasharenow.com/>

This document starts with a familiar pattern. The data is read in. Then the data is cleaned up as there are misspellings and other errors.

The data is broken down by high schools and school areas that had few sales were eliminated with the use of a user-defined R function.

First I did regression. Then random forest. Then boosting.

In each case, I created a model on the training set. Then I performed predictions on the test set.

The comparison table is at the end.

For the analyses, I broke the data up into a training set and a test set. I then performed a number of analyses. I plan on revisting this data set to do further analyses in the near future.

The document without R code is available in the alternate version of this document.

# ```{r setup, include=TRUE}
knitr::opts\_chunk$set(echo = TRUE, ft.align="left")

# Madison housing second version

# Preliminaries

library(tidyverse)
library(scales) # Graphing
library(flextable)
library(readxl) # read Excel file
library(stringr) # string manipulation
library(lubridate) # working with dates
library(leaps) # selecting best regressions
library(tree) # basic tree function
library(randomForest) # random forest
library(gbm) # gradient boosting
library(faraway) # regression techniques

# Set flextable defaults

set\_flextable\_defaults(
 font.size = 10, theme\_fun = theme\_vanilla,
 padding = 6,
 background.color = "#EFEFEF")

setwd("C:/Users/irash/Documents/Statistics/Web 2023/Madison")
load("housing.RData")

# Clean up school data -- deal with spelling issues

# housing |> select(High) |> unique() |> arrange(High)
housing$High[housing$High == 'Call School District'] = NA
housing$High[housing$High == 'East High'] = 'East'
housing$High[housing$High == 'MONONA'] = 'Monona'
housing$High[housing$High == 'Not Assigned'] = NA
housing$High[housing$High == 'Southwestern Wisconsin'] = NA
housing$High[housing$High == 'Southwestern Wisconsin'] = NA
housing$High[housing$High == 'Sun Prarie'] = 'Sun Prarie'
housing$High[housing$High == 'SunPrairie'] = 'Sun Prarie'
housing$High[housing$High == 'lafolette'] = 'Lafollette'

# housing |> select(Middle\_School) |> unique() |> arrange(Middle\_School)
housing$Middle\_School[housing$Middle\_School == 'Blackhawk'] = 'Black Hawk'
housing$Middle\_School[housing$Middle\_School == 'Call School District'] = NA
housing$Middle\_School[housing$Middle\_School == 'Blackhawk'] = 'Black Hawk'
housing$Middle\_School[housing$Middle\_School == 'Kromry'] = 'Kromrey'
housing$Middle\_School[housing$Middle\_School == 'Mc farland'] = 'McFarland'
housing$Middle\_School[housing$Middle\_School == 'Not Assigned'] = NA
housing$Middle\_School[housing$Middle\_School == 'Mc farland'] = 'McFarland'
housing$Middle\_School[housing$Middle\_School == "O'Keeffee"] = "O'Keeffe"
housing$Middle\_School[housing$Middle\_School == "Okeefe"] = "O'Keeffe"
housing$Middle\_School[housing$Middle\_School == "okeefe"] = "O'Keeffe"
housing$Middle\_School[housing$Middle\_School == 'Optional'] = NA
housing$Middle\_School[housing$Middle\_School == 'Pat. Marsh'] = 'Patrick Marsh'
housing$Middle\_School[housing$Middle\_School == 'PrairiView'] = 'Prairie View'
housing$Middle\_School[housing$Middle\_School == 'Sand hill'] = 'Sandhill'
housing$Middle\_School[housing$Middle\_School == 'VERONA'] = 'Verona'
housing$Middle\_School[housing$Middle\_School == 'Verona mid'] = 'Verona'
housing$Middle\_School[housing$Middle\_School == 'call'] = NA
housing$Middle\_School[housing$Middle\_School == 'unknown'] = NA

# housing |> select(Elementary) |> unique() |> arrange(Elementary)
housing$Elementary[housing$Elementary == 'Call School District'] = NA
housing$Elementary[housing$Elementary == 'Country Vw'] = 'Country View'
housing$Elementary[housing$Elementary == 'Lakeview'] = 'Lake View'
housing$Elementary[housing$Elementary == 'Mid/Lin'] = 'Midvale/Lincoln'
housing$Elementary[housing$Elementary == 'Not Assigned'] = NA
housing$Elementary[housing$Elementary == 'Optional'] = NA
housing$Elementary[housing$Elementary == 'VERONA'] = 'Verona'
housing$Elementary[housing$Elementary == 'Willson'] = 'Wilson'
housing$Elementary[housing$Elementary == 'call'] = NA
housing$Elementary[housing$Elementary == 'gomph'] = NA
housing$Elementary[housing$Elementary == 'lap/marq'] = 'Lapham/Marquette'

# Now get rid of bad data and not useful data.
housing = housing |> filter(FinSqFt <= 5000) # reduce the data frame to only include FinSqFt <= 5000
housing = housing |> filter(FinSqFt >= 500) # # reduce the data frame to only include FinSqFt >= 500
housing = housing |> filter(Sold\_Price >= 30000)
housing = housing |> filter(Sold\_Price <= 1200000)
housing = housing |> filter(Total\_Baths != 0)
housing = housing |> filter(Total\_Baths < 8)
housing = housing |> filter(Beds %in% 1:7) # include numbers of bedrooms between 1 and 7 inclusive
housing = housing |> filter(YearBuilt >= 1900)

# Create some useful explanatory variables
housing$Closing\_Date\_year = year(housing$Closing\_Date)
housing$Closing\_Date\_month = month(housing$Closing\_Date)
housing$Closing\_Date\_day = day(housing$Closing\_Date)
housing$Closing\_Decade =
 ifelse(housing$Closing\_Date\_year >=1990 &
 housing$Closing\_Date\_year <= 1999, "D1990s",
 ifelse(housing$Closing\_Date\_year >= 2000 &
 housing$Closing\_Date\_year <= 2009, "D2000s",
 ifelse(housing$Closing\_Date\_year >= 2010 &
 housing$Closing\_Date\_year <= 2019, "D2010s",
 "D2020s")))
housing$Closing\_Decade = factor(housing$Closing\_Decade)
housing$YearBuilt1980\_plus =
 ifelse(housing$YearBuilt >= 1980, 1, 0)

# Divide data into three groups by size of house

housing$SqFt\_grp =
 ifelse(housing$FinSqFt < 1600, "Small",
 ifelse(housing$FinSqFt >= 1600 & housing$FinSqFt <= 2160, "Medium",
 ifelse(housing$FinSqFt > 2160, "Large", NA)))

# Only keep high schools with enough sales in 2018 to 2022
minCount = 10

# form 2-way table, school against year
sdTable <- table(housing$High, housing$Closing\_Date\_year)

# want years 2018-2022 having at least 10 rows in housing data
sdTable <- sdTable[,25:29]

# which have >= k (10) rows in all years 2018-2022
allGtEq <- function(oneRow) all(oneRow >= minCount)
whichToKeep <- which(apply(sdTable,1,allGtEq))

# whichToKeep is row numbers from the table; get the school names
whichToKeep <- names(whichToKeep)

# back to schoolData
whichOrigRowsToKeep <- which(housing$High %in% whichToKeep)
newHousing <- housing[whichOrigRowsToKeep,]

# Add small, medium, and large variables
# Then create 3 subsets based on size

# Divide housing by size into three groups

newHousingSmall = newHousing |> filter(SqFt\_grp == 'Small') # < 1600 SQFT

newHousingMedium = newHousing |> filter(SqFt\_grp == 'Medium') # < 1600 SQFT

newHousingLarge = newHousing |> filter(SqFt\_grp == 'Large') # > 2160 SQFT

# A cross tab of decade and size

res1 = data.frame(table(newHousing$Closing\_Decade, newHousing$SqFt\_grp))
res1 = res1 |>
 pivot\_wider(names\_from = Var2, values\_from = Freq) |>
 select(Var1, Small, Medium, Large) |> rename(Decade = Var1)

ft\_res1 = flextable(res1)
ft\_res1 = set\_caption(ft\_res1, "Home Sales by Decade and Size")
ft\_res1 = add\_header\_row(ft\_res1,
 colwidths = 4,
 values = "Madison Area Home Sales")
ft\_res1 <-
 add\_footer\_lines(ft\_res1, "Years: 1994-2023\nSmall: < 1600 SqFt;\nMedium: 1600-2160 SqFt;\nLarge > 2160 SqFt")
ft\_res1

Home Sales by Decade and Size

| **Madison Area Home Sales** |
| --- |
| **Decade** | **Small** | **Medium** | **Large** |
| D1990s | 7,495 | 5,915 | 4,509 |
| D2000s | 13,126 | 13,180 | 11,621 |
| D2010s | 11,525 | 14,168 | 15,697 |
| D2020s | 4,059 | 4,997 | 5,630 |
| Years: 1994-2023Small: < 1600 SqFt;Medium: 1600-2160 SqFt;Large > 2160 SqFt |

# Some exploratory charts

newHousing |> ggplot(mapping = aes(FinSqFt)) +
 geom\_histogram(bins = 30, fill = "red", color = "black") +
 labs(title =
 "Histogram of housing sizes (<= 5000 SQ FT)\nMadison Area Housing Sales",
 subtitle = "Sales 1994-2023") +
 theme(plot.title = element\_text(hjust = 0.5),
 plot.subtitle = element\_text(hjust = 0.5)) +
 xlab("Size of house in Square Feet") +
 annotate("text", x = 4000, y = 4000, label =
 "Homes over 5,000 square feet\nexcluded from analysis")



Boxplots of Sale Price by High School and Decade of Sale

newHousing |> filter(High %in% c("East", "West", "Memorial")) |>
 ggplot(mapping = aes(x = High, y = Sold\_Price)) +
 facet\_grid(. ~ Closing\_Decade) +
 geom\_boxplot(fill = "red", color = "black") +
 labs(title =
 "Boxplots of Sale Price by High School\nand Decade of Sale") +
 theme(plot.title = element\_text(hjust = 0.5),
 plot.subtitle = element\_text(hjust = 0.5)) +
 theme(axis.text.x = element\_text(angle = 90)) +
 xlab("High School")



newHousing |> filter(High %in% c("East", "West", "Memorial")) |>
 ggplot(mapping = aes(x = Closing\_Decade, y = Sold\_Price)) +
 facet\_grid(. ~ High) +
 geom\_boxplot(fill = "red", color = "black") +
 labs(title =
 "Boxplots of Sale Price by Decade of Sale\nand High School") +
 theme(plot.title = element\_text(hjust = 0.5),
 plot.subtitle = element\_text(hjust = 0.5)) +
 xlab("Closing Decade") +
 theme(axis.text.x = element\_text(angle = 90)) +
 scale\_y\_continuous(labels=scales::dollar\_format())



newHousing |> filter(ZIP5 %in% c(53711, 53704, 53590, 53719, 53716)) |>
 ggplot(mapping = aes(x = ZIP5, y = Sold\_Price)) +
 facet\_grid(. ~ Closing\_Decade) +
 geom\_boxplot(fill = "red", color = "black") +
 labs(title =
 "Boxplots of Sale Price by ZIP5\nand Decade of Sale") +
 theme(plot.title = element\_text(hjust = 0.5),
 plot.subtitle = element\_text(hjust = 0.5)) +
 xlab("Zip Code") +
 theme(axis.text.x = element\_text(angle = 90)) +
 scale\_y\_continuous(labels=scales::dollar\_format())



# Regression

Split data in training and test sets
Then do regression on training set
and predict on the test set

# split data into training set and test set
rowsHousing = nrow(newHousing) # 121601
halves = floor(rowsHousing/2)
set.seed(202302)
train <- sample (1: nrow(housing), halves, replace = FALSE)

housingTrain = newHousing |> slice(train)

housingTest = newHousing[-train, ]
housingTestV = housingTest[, "Sold\_Price"]
housingTestVSF = housingTest[, "Sold\_Price\_Per\_SQFT"]

lm.fit = lm(Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus +
 Beds + LandAssess +
 Total\_Full\_Garage\_Stalls + Total\_Baths + High,
 data = housingTrain)
cat("The regression formula is:lm.fit = lm(Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus + Beds + LandAssess + Total\_Full\_Garage\_Stalls + Total\_Baths + High, data = housingTrain)")

## The regression formula is:lm.fit = lm(Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus + Beds + LandAssess + Total\_Full\_Garage\_Stalls + Total\_Baths + High, data = housingTrain)

lm.fit |> as\_flextable()

|  | **Estimate** | **Standard Error** | **t value** | **Pr(>|t|)** |  |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | -14,966,042.952 | 82,757.792 | -180.841 | 0.0000 | \*\*\* |
| FinSqFt | 90.118 | 0.799 | 112.784 | 0.0000 | \*\*\* |
| Closing\_Date\_year | 7,468.471 | 41.193 | 181.302 | 0.0000 | \*\*\* |
| YearBuilt1980\_plus | 15,737.545 | 862.532 | 18.246 | 0.0000 | \*\*\* |
| Beds | -36.748 | 659.499 | -0.056 | 0.9556 |   |
| LandAssess | 0.012 | 0.001 | 11.486 | 0.0000 | \*\*\* |
| Total\_Full\_Garage\_Stalls | 6,753.770 | 531.820 | 12.699 | 0.0000 | \*\*\* |
| Total\_Baths | 4,529.345 | 780.922 | 5.800 | 0.0000 | \*\*\* |
| HighEast | 5,041.789 | 4,371.555 | 1.153 | 0.2488 |   |
| HighLafollette | -15,727.800 | 4,351.389 | -3.614 | 0.0003 | \*\*\* |
| HighMcFarland | 3,651.282 | 4,628.138 | 0.789 | 0.4302 |   |
| HighMemorial | 4,964.116 | 4,312.594 | 1.151 | 0.2497 |   |
| HighMiddleton | 69,944.538 | 4,854.691 | 14.408 | 0.0000 | \*\*\* |
| HighMonona Grove | 180.184 | 4,524.939 | 0.040 | 0.9682 |   |
| HighOregon | 2,869.155 | 4,505.812 | 0.637 | 0.5243 |   |
| HighStoughton | -13,178.824 | 4,492.255 | -2.934 | 0.0034 |  \*\* |
| HighSun Prairie East | -22,503.858 | 4,338.327 | -5.187 | 0.0000 | \*\*\* |
| HighVerona | 6,654.108 | 4,425.759 | 1.503 | 0.1327 |   |
| HighWaunakee | 27,848.989 | 4,507.681 | 6.178 | 0.0000 | \*\*\* |
| HighWest | 59,651.492 | 4,362.233 | 13.675 | 0.0000 | \*\*\* |
| *Signif. codes: 0 <= '\*\*\*' < 0.001 < '\*\*' < 0.01 < '\*' < 0.05* |
|  |
| Residual standard error: 7.69e+04 on 52217 degrees of freedom |
| Multiple R-squared: 0.6711, Adjusted R-squared: 0.671 |
| F-statistic: 5609 on 52217 and 19 DF, p-value: 0.0000 |

lm.fit = lm(Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus +
 Beds + LandAssess +
 Total\_Full\_Garage\_Stalls + Total\_Baths + High,
 data = housingTrain)

#create residual plot
ggplot(lm.fit, aes(x = .fitted, y = .resid)) +
 geom\_point() +
 geom\_hline(yintercept = 0, linewidth = 1.2, col = "red") +
 labs(title =
 "Residuals versus Fitted") +
 theme(plot.title = element\_text(hjust = 0.5))



# Now look at the test data
yhat\_R1A <- predict(lm.fit , newdata = housingTest)
yhat\_R1A2 <- predict(lm.fit , newdata = housingTest, interval = "prediction")

cat(paste("mean square error: ", round(mean(((yhat\_R1A - housingTestV)^2), na.rm = TRUE) )))

## mean square error: 5921807030

cat(paste("mean absolute difference: ", round(mean((abs(yhat\_R1A - housingTestV)), na.rm = TRUE))))

## mean absolute difference: 51887

ht = data.frame(SqFT = housingTest$FinSqFt, Sold\_Price = housingTest$Sold\_Price)

ht |> ggplot(aes(x = SqFT, y = Sold\_Price)) + geom\_point(alpha = 0.1) +
 labs(title = "Sold Price vs. Square Feet") +
 theme(plot.title = element\_text(hjust = 0.5)) +
 xlab("Size of House in Square Feet") +
 ylab("Sold Price") +
 scale\_y\_continuous(labels=scales::dollar\_format())



## Random Forest

#################
# Random Forest #
#################

set.seed(202302)

housingTrainRF = complete.cases(housingTrain)
rf.Dane4 <-
 randomForest(Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus +
 Beds + Full\_Baths + YearBuilt + Total\_Baths + ZIP5+ High
 ,
 data = housingTrain,
 mtry = 4, importance = TRUE, ntree = 200
)

rf.Dane4

##
## Call:
## randomForest(formula = Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus + Beds + Full\_Baths + YearBuilt + Total\_Baths + ZIP5 + High, data = housingTrain, mtry = 4, importance = TRUE, ntree = 200)
## Type of random forest: regression
## Number of trees: 200
## No. of variables tried at each split: 4
##
## Mean of squared residuals: 3273089717
## % Var explained: 82.05

yhat.rf <- predict(rf.Dane4, newdata = newHousing[-train , ])
cat(paste("RF MSE: ", mean ((yhat.rf - housingTestV)^2)))

## RF MSE: 3199413497.22394

cat(paste("mean abs error: ", mean(abs(yhat.rf - housingTestV))))

## mean abs error: 32273.7294074767

importance(rf.Dane4)

## %IncMSE IncNodePurity
## FinSqFt 77.36642 3.452654e+14
## Closing\_Date\_year 327.82376 2.650597e+14
## YearBuilt1980\_plus 26.50031 4.475491e+12
## Beds 28.77647 1.947189e+13
## Full\_Baths 12.07724 2.218624e+13
## YearBuilt 134.38530 1.246365e+14
## Total\_Baths 19.23591 1.114328e+14
## ZIP5 71.91995 3.139464e+13
## High 96.52026 4.368963e+13

varImpPlot(rf.Dane4)



## Boosting

############
# Boosting #
############

# Need factors for boosting
housingTrain$ZIP5F = factor(housingTrain$ZIP5)
housingTrain$HighF = factor(housingTrain$High)

newHousingTest = newHousing[-train , ]

newHousingTest$ZIP5F = factor(newHousingTest$ZIP5)
newHousingTest$HighF = factor(newHousingTest$High)

boost.Dane <- gbm(Sold\_Price ~ FinSqFt + Closing\_Date\_year + YearBuilt1980\_plus +
 Beds + Full\_Baths + YearBuilt +
 Total\_Baths + ZIP5F + HighF,
 data = housingTrain,
 distribution = "gaussian", n.trees = 5000,
 interaction.depth = 4)

summary(boost.Dane)



## var rel.inf
## FinSqFt FinSqFt 42.87730738
## Closing\_Date\_year Closing\_Date\_year 26.43764245
## YearBuilt YearBuilt 11.48942343
## ZIP5F ZIP5F 10.06617681
## HighF HighF 5.03125775
## Total\_Baths Total\_Baths 2.91438632
## Beds Beds 0.79407083
## Full\_Baths Full\_Baths 0.36061158
## YearBuilt1980\_plus YearBuilt1980\_plus 0.02912345

yhat.boost <- predict(boost.Dane ,
 newdata = newHousingTest, n.trees = 5000)
cat(paste("MSE Boosting: ", mean(( yhat.boost - housingTestV)^2)))

## MSE Boosting: 3061540248.45942

cat(paste("AMAD Boosting: ", mean(abs(yhat.boost - housingTestV))))

## AMAD Boosting: 31876.3971474589

summaryDF =
 data.frame(
 Technique = c("Regression", "RF", "Boosting"),
 VarExpl = c(0.671, 0.821, NA),
 MeanAbsError = c(51887, 32773, 31876))
summaryDF

## Technique VarExpl MeanAbsError
## 1 Regression 0.671 51887
## 2 RF 0.821 32773
## 3 Boosting NA 31876

ft\_summary = flextable(summaryDF)
ft\_summary = set\_table\_properties(ft\_summary, width = .3, layout = "autofit")
ft\_summary = set\_table\_properties(ft\_summary, align = "left", layout = "autofit")
ft\_summary = set\_caption(ft\_summary, "Performance of the 3 Data Science Techniques")
ft\_summary = add\_header\_row(ft\_summary,
 colwidths = 3,
 values = "Analysis of Madison Area Home Sales")

ft\_summary

Performance of the 3 Data Science Techniques

| **Analysis of Madison Area Home Sales** |
| --- |
| **Technique** | **VarExpl** | **MeanAbsError** |
| Regression | 0.671 | 51,887 |
| RF | 0.821 | 32,773 |
| Boosting |  | 31,876 |