FinalPrjctPhs2

2023-06-29

## Subbu Subramanian

# Analysis of Homes sold Above median value in Ames, Iowa:

## Libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.0 ──  
## ✔ broom 1.0.4 ✔ rsample 1.1.1  
## ✔ dials 1.2.0 ✔ tune 1.1.1  
## ✔ infer 1.0.4 ✔ workflows 1.1.3  
## ✔ modeldata 1.1.0 ✔ workflowsets 1.0.1  
## ✔ parsnip 1.1.0 ✔ yardstick 1.2.0  
## ✔ recipes 1.0.6   
## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Use suppressPackageStartupMessages() to eliminate package startup messages

library(rpart) #for classification trees

##   
## Attaching package: 'rpart'  
##   
## The following object is masked from 'package:dials':  
##   
## prune

library(rpart.plot) #plotting trees  
library(RColorBrewer) #better visualization of classification trees  
library(rattle) #better visualization of classification trees

## 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.

library(caret) #for easy confusion matrix creation

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

## Data and Initial Analysis

Homes = read\_csv("ames\_student-1-2.csv")

## Rows: 2053 Columns: 81  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (47): MS\_SubClass, MS\_Zoning, Street, Alley, Lot\_Shape, Land\_Contour, Ut...  
## dbl (34): Lot\_Frontage, Lot\_Area, Year\_Built, Year\_Remod\_Add, Mas\_Vnr\_Area, ...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

str(Homes)

## spc\_tbl\_ [2,053 × 81] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ MS\_SubClass : chr [1:2053] "One\_Story\_1946\_and\_Newer\_All\_Styles" "One\_Story\_1946\_and\_Newer\_All\_Styles" "One\_Story\_1946\_and\_Newer\_All\_Styles" "One\_Story\_1946\_and\_Newer\_All\_Styles" ...  
## $ MS\_Zoning : chr [1:2053] "Residential\_Low\_Density" "Residential\_High\_Density" "Residential\_Low\_Density" "Residential\_Low\_Density" ...  
## $ Lot\_Frontage : num [1:2053] 141 80 81 93 74 78 43 39 0 85 ...  
## $ Lot\_Area : num [1:2053] 31770 11622 14267 11160 13830 ...  
## $ Street : chr [1:2053] "Pave" "Pave" "Pave" "Pave" ...  
## $ Alley : chr [1:2053] "No\_Alley\_Access" "No\_Alley\_Access" "No\_Alley\_Access" "No\_Alley\_Access" ...  
## $ Lot\_Shape : chr [1:2053] "Slightly\_Irregular" "Regular" "Slightly\_Irregular" "Regular" ...  
## $ Land\_Contour : chr [1:2053] "Lvl" "Lvl" "Lvl" "Lvl" ...  
## $ Utilities : chr [1:2053] "AllPub" "AllPub" "AllPub" "AllPub" ...  
## $ Lot\_Config : chr [1:2053] "Corner" "Inside" "Corner" "Corner" ...  
## $ Land\_Slope : chr [1:2053] "Gtl" "Gtl" "Gtl" "Gtl" ...  
## $ Neighborhood : chr [1:2053] "North\_Ames" "North\_Ames" "North\_Ames" "North\_Ames" ...  
## $ Condition\_1 : chr [1:2053] "Norm" "Feedr" "Norm" "Norm" ...  
## $ Condition\_2 : chr [1:2053] "Norm" "Norm" "Norm" "Norm" ...  
## $ Bldg\_Type : chr [1:2053] "OneFam" "OneFam" "OneFam" "OneFam" ...  
## $ House\_Style : chr [1:2053] "One\_Story" "One\_Story" "One\_Story" "One\_Story" ...  
## $ Overall\_Qual : chr [1:2053] "Above\_Average" "Average" "Above\_Average" "Good" ...  
## $ Overall\_Cond : chr [1:2053] "Average" "Above\_Average" "Above\_Average" "Average" ...  
## $ Year\_Built : num [1:2053] 1960 1961 1958 1968 1997 ...  
## $ Year\_Remod\_Add : num [1:2053] 1960 1961 1958 1968 1998 ...  
## $ Roof\_Style : chr [1:2053] "Hip" "Gable" "Hip" "Hip" ...  
## $ Roof\_Matl : chr [1:2053] "CompShg" "CompShg" "CompShg" "CompShg" ...  
## $ Exterior\_1st : chr [1:2053] "BrkFace" "VinylSd" "Wd Sdng" "BrkFace" ...  
## $ Exterior\_2nd : chr [1:2053] "Plywood" "VinylSd" "Wd Sdng" "BrkFace" ...  
## $ Mas\_Vnr\_Type : chr [1:2053] "Stone" "None" "BrkFace" "None" ...  
## $ Mas\_Vnr\_Area : num [1:2053] 112 0 108 0 0 20 0 0 0 0 ...  
## $ Exter\_Qual : chr [1:2053] "Typical" "Typical" "Typical" "Good" ...  
## $ Exter\_Cond : chr [1:2053] "Typical" "Typical" "Typical" "Typical" ...  
## $ Foundation : chr [1:2053] "CBlock" "CBlock" "CBlock" "CBlock" ...  
## $ Bsmt\_Qual : chr [1:2053] "Typical" "Typical" "Typical" "Typical" ...  
## $ Bsmt\_Cond : chr [1:2053] "Good" "Typical" "Typical" "Typical" ...  
## $ Bsmt\_Exposure : chr [1:2053] "Gd" "No" "No" "No" ...  
## $ BsmtFin\_Type\_1 : chr [1:2053] "BLQ" "Rec" "ALQ" "ALQ" ...  
## $ BsmtFin\_SF\_1 : num [1:2053] 2 6 1 1 3 3 1 3 1 3 ...  
## $ BsmtFin\_Type\_2 : chr [1:2053] "Unf" "LwQ" "Unf" "Unf" ...  
## $ BsmtFin\_SF\_2 : num [1:2053] 0 144 0 0 0 0 0 0 0 0 ...  
## $ Bsmt\_Unf\_SF : num [1:2053] 441 270 406 1045 137 ...  
## $ Total\_Bsmt\_SF : num [1:2053] 1080 882 1329 2110 928 ...  
## $ Heating : chr [1:2053] "GasA" "GasA" "GasA" "GasA" ...  
## $ Heating\_QC : chr [1:2053] "Fair" "Typical" "Typical" "Excellent" ...  
## $ Central\_Air : chr [1:2053] "Y" "Y" "Y" "Y" ...  
## $ Electrical : chr [1:2053] "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...  
## $ First\_Flr\_SF : num [1:2053] 1656 896 1329 2110 928 ...  
## $ Second\_Flr\_SF : num [1:2053] 0 0 0 0 701 678 0 0 0 0 ...  
## $ Low\_Qual\_Fin\_SF : num [1:2053] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Gr\_Liv\_Area : num [1:2053] 1656 896 1329 2110 1629 ...  
## $ Bsmt\_Full\_Bath : num [1:2053] 1 0 0 1 0 0 0 1 1 1 ...  
## $ Bsmt\_Half\_Bath : num [1:2053] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Full\_Bath : num [1:2053] 1 1 1 2 2 2 2 2 2 1 ...  
## $ Half\_Bath : num [1:2053] 0 0 1 1 1 1 0 0 0 1 ...  
## $ Bedroom\_AbvGr : num [1:2053] 3 2 3 3 3 3 2 2 3 2 ...  
## $ Kitchen\_AbvGr : num [1:2053] 1 1 1 1 1 1 1 1 1 1 ...  
## $ Kitchen\_Qual : chr [1:2053] "Typical" "Typical" "Good" "Excellent" ...  
## $ TotRms\_AbvGrd : num [1:2053] 7 5 6 8 6 7 5 5 6 5 ...  
## $ Functional : chr [1:2053] "Typ" "Typ" "Typ" "Typ" ...  
## $ Fireplaces : num [1:2053] 2 0 0 2 1 1 0 1 0 1 ...  
## $ Fireplace\_Qu : chr [1:2053] "Good" "No\_Fireplace" "No\_Fireplace" "Typical" ...  
## $ Garage\_Type : chr [1:2053] "Attchd" "Attchd" "Attchd" "Attchd" ...  
## $ Garage\_Finish : chr [1:2053] "Fin" "Unf" "Unf" "Fin" ...  
## $ Garage\_Cars : num [1:2053] 2 1 1 2 2 2 2 2 2 2 ...  
## $ Garage\_Area : num [1:2053] 528 730 312 522 482 470 506 608 420 506 ...  
## $ Garage\_Qual : chr [1:2053] "Typical" "Typical" "Typical" "Typical" ...  
## $ Garage\_Cond : chr [1:2053] "Typical" "Typical" "Typical" "Typical" ...  
## $ Paved\_Drive : chr [1:2053] "Partial\_Pavement" "Paved" "Paved" "Paved" ...  
## $ Wood\_Deck\_SF : num [1:2053] 210 140 393 0 212 360 0 237 483 192 ...  
## $ Open\_Porch\_SF : num [1:2053] 62 0 36 0 34 36 82 152 21 0 ...  
## $ Enclosed\_Porch : num [1:2053] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Three\_season\_porch: num [1:2053] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Screen\_Porch : num [1:2053] 0 120 0 0 0 0 144 0 0 0 ...  
## $ Pool\_Area : num [1:2053] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Pool\_QC : chr [1:2053] "No\_Pool" "No\_Pool" "No\_Pool" "No\_Pool" ...  
## $ Fence : chr [1:2053] "No\_Fence" "Minimum\_Privacy" "No\_Fence" "No\_Fence" ...  
## $ Misc\_Feature : chr [1:2053] "None" "None" "Gar2" "None" ...  
## $ Misc\_Val : num [1:2053] 0 0 12500 0 0 0 0 0 500 0 ...  
## $ Mo\_Sold : num [1:2053] 5 6 6 4 3 6 1 3 3 2 ...  
## $ Year\_Sold : num [1:2053] 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...  
## $ Sale\_Type : chr [1:2053] "WD" "WD" "WD" "WD" ...  
## $ Sale\_Condition : chr [1:2053] "Normal" "Normal" "Normal" "Normal" ...  
## $ Longitude : num [1:2053] -93.6 -93.6 -93.6 -93.6 -93.6 ...  
## $ Latitude : num [1:2053] 42.1 42.1 42.1 42.1 42.1 ...  
## $ Above\_Median : chr [1:2053] "Yes" "No" "Yes" "Yes" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. MS\_SubClass = col\_character(),  
## .. MS\_Zoning = col\_character(),  
## .. Lot\_Frontage = col\_double(),  
## .. Lot\_Area = col\_double(),  
## .. Street = col\_character(),  
## .. Alley = col\_character(),  
## .. Lot\_Shape = col\_character(),  
## .. Land\_Contour = col\_character(),  
## .. Utilities = col\_character(),  
## .. Lot\_Config = col\_character(),  
## .. Land\_Slope = col\_character(),  
## .. Neighborhood = col\_character(),  
## .. Condition\_1 = col\_character(),  
## .. Condition\_2 = col\_character(),  
## .. Bldg\_Type = col\_character(),  
## .. House\_Style = col\_character(),  
## .. Overall\_Qual = col\_character(),  
## .. Overall\_Cond = col\_character(),  
## .. Year\_Built = col\_double(),  
## .. Year\_Remod\_Add = col\_double(),  
## .. Roof\_Style = col\_character(),  
## .. Roof\_Matl = col\_character(),  
## .. Exterior\_1st = col\_character(),  
## .. Exterior\_2nd = col\_character(),  
## .. Mas\_Vnr\_Type = col\_character(),  
## .. Mas\_Vnr\_Area = col\_double(),  
## .. Exter\_Qual = col\_character(),  
## .. Exter\_Cond = col\_character(),  
## .. Foundation = col\_character(),  
## .. Bsmt\_Qual = col\_character(),  
## .. Bsmt\_Cond = col\_character(),  
## .. Bsmt\_Exposure = col\_character(),  
## .. BsmtFin\_Type\_1 = col\_character(),  
## .. BsmtFin\_SF\_1 = col\_double(),  
## .. BsmtFin\_Type\_2 = col\_character(),  
## .. BsmtFin\_SF\_2 = col\_double(),  
## .. Bsmt\_Unf\_SF = col\_double(),  
## .. Total\_Bsmt\_SF = col\_double(),  
## .. Heating = col\_character(),  
## .. Heating\_QC = col\_character(),  
## .. Central\_Air = col\_character(),  
## .. Electrical = col\_character(),  
## .. First\_Flr\_SF = col\_double(),  
## .. Second\_Flr\_SF = col\_double(),  
## .. Low\_Qual\_Fin\_SF = col\_double(),  
## .. Gr\_Liv\_Area = col\_double(),  
## .. Bsmt\_Full\_Bath = col\_double(),  
## .. Bsmt\_Half\_Bath = col\_double(),  
## .. Full\_Bath = col\_double(),  
## .. Half\_Bath = col\_double(),  
## .. Bedroom\_AbvGr = col\_double(),  
## .. Kitchen\_AbvGr = col\_double(),  
## .. Kitchen\_Qual = col\_character(),  
## .. TotRms\_AbvGrd = col\_double(),  
## .. Functional = col\_character(),  
## .. Fireplaces = col\_double(),  
## .. Fireplace\_Qu = col\_character(),  
## .. Garage\_Type = col\_character(),  
## .. Garage\_Finish = col\_character(),  
## .. Garage\_Cars = col\_double(),  
## .. Garage\_Area = col\_double(),  
## .. Garage\_Qual = col\_character(),  
## .. Garage\_Cond = col\_character(),  
## .. Paved\_Drive = col\_character(),  
## .. Wood\_Deck\_SF = col\_double(),  
## .. Open\_Porch\_SF = col\_double(),  
## .. Enclosed\_Porch = col\_double(),  
## .. Three\_season\_porch = col\_double(),  
## .. Screen\_Porch = col\_double(),  
## .. Pool\_Area = col\_double(),  
## .. Pool\_QC = col\_character(),  
## .. Fence = col\_character(),  
## .. Misc\_Feature = col\_character(),  
## .. Misc\_Val = col\_double(),  
## .. Mo\_Sold = col\_double(),  
## .. Year\_Sold = col\_double(),  
## .. Sale\_Type = col\_character(),  
## .. Sale\_Condition = col\_character(),  
## .. Longitude = col\_double(),  
## .. Latitude = col\_double(),  
## .. Above\_Median = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

summary(Homes)

## MS\_SubClass MS\_Zoning Lot\_Frontage Lot\_Area   
## Length:2053 Length:2053 Min. : 0.00 Min. : 1300   
## Class :character Class :character 1st Qu.: 43.00 1st Qu.: 7500   
## Mode :character Mode :character Median : 62.00 Median : 9548   
## Mean : 57.38 Mean : 10258   
## 3rd Qu.: 78.00 3rd Qu.: 11600   
## Max. :313.00 Max. :215245   
## Street Alley Lot\_Shape Land\_Contour   
## Length:2053 Length:2053 Length:2053 Length:2053   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Utilities Lot\_Config Land\_Slope Neighborhood   
## Length:2053 Length:2053 Length:2053 Length:2053   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Condition\_1 Condition\_2 Bldg\_Type House\_Style   
## Length:2053 Length:2053 Length:2053 Length:2053   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Overall\_Qual Overall\_Cond Year\_Built Year\_Remod\_Add  
## Length:2053 Length:2053 Min. :1875 Min. :1950   
## Class :character Class :character 1st Qu.:1953 1st Qu.:1965   
## Mode :character Mode :character Median :1972 Median :1993   
## Mean :1971 Mean :1984   
## 3rd Qu.:2000 3rd Qu.:2004   
## Max. :2010 Max. :2010   
## Roof\_Style Roof\_Matl Exterior\_1st Exterior\_2nd   
## Length:2053 Length:2053 Length:2053 Length:2053   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## Mas\_Vnr\_Type Mas\_Vnr\_Area Exter\_Qual Exter\_Cond   
## Length:2053 Min. : 0.0 Length:2053 Length:2053   
## Class :character 1st Qu.: 0.0 Class :character Class :character   
## Mode :character Median : 0.0 Mode :character Mode :character   
## Mean : 103.8   
## 3rd Qu.: 164.0   
## Max. :1600.0   
## Foundation Bsmt\_Qual Bsmt\_Cond Bsmt\_Exposure   
## Length:2053 Length:2053 Length:2053 Length:2053   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## BsmtFin\_Type\_1 BsmtFin\_SF\_1 BsmtFin\_Type\_2 BsmtFin\_SF\_2   
## Length:2053 Min. :1.00 Length:2053 Min. : 0.00   
## Class :character 1st Qu.:3.00 Class :character 1st Qu.: 0.00   
## Mode :character Median :3.00 Mode :character Median : 0.00   
## Mean :4.21 Mean : 52.57   
## 3rd Qu.:7.00 3rd Qu.: 0.00   
## Max. :7.00 Max. :1526.00   
## Bsmt\_Unf\_SF Total\_Bsmt\_SF Heating Heating\_QC   
## Min. : 0.0 Min. : 0 Length:2053 Length:2053   
## 1st Qu.: 226.0 1st Qu.: 793 Class :character Class :character   
## Median : 460.0 Median : 988 Mode :character Mode :character   
## Mean : 561.2 Mean :1055   
## 3rd Qu.: 801.0 3rd Qu.:1304   
## Max. :2336.0 Max. :5095   
## Central\_Air Electrical First\_Flr\_SF Second\_Flr\_SF   
## Length:2053 Length:2053 Min. : 432 Min. : 0.0   
## Class :character Class :character 1st Qu.: 882 1st Qu.: 0.0   
## Mode :character Mode :character Median :1088 Median : 0.0   
## Mean :1168 Mean : 326.1   
## 3rd Qu.:1402 3rd Qu.: 701.0   
## Max. :5095 Max. :1862.0   
## Low\_Qual\_Fin\_SF Gr\_Liv\_Area Bsmt\_Full\_Bath Bsmt\_Half\_Bath   
## Min. : 0.000 Min. : 480 Min. :0.0000 Min. :0.00000   
## 1st Qu.: 0.000 1st Qu.:1137 1st Qu.:0.0000 1st Qu.:0.00000   
## Median : 0.000 Median :1447 Median :0.0000 Median :0.00000   
## Mean : 4.973 Mean :1499 Mean :0.4301 Mean :0.05796   
## 3rd Qu.: 0.000 3rd Qu.:1737 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1064.000 Max. :5095 Max. :3.0000 Max. :2.00000   
## Full\_Bath Half\_Bath Bedroom\_AbvGr Kitchen\_AbvGr   
## Min. :0.000 Min. :0.0000 Min. :0.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.:0.0000 1st Qu.:2.000 1st Qu.:1.000   
## Median :2.000 Median :0.0000 Median :3.000 Median :1.000   
## Mean :1.564 Mean :0.3751 Mean :2.855 Mean :1.047   
## 3rd Qu.:2.000 3rd Qu.:1.0000 3rd Qu.:3.000 3rd Qu.:1.000   
## Max. :4.000 Max. :2.0000 Max. :6.000 Max. :3.000   
## Kitchen\_Qual TotRms\_AbvGrd Functional Fireplaces   
## Length:2053 Min. : 3.000 Length:2053 Min. :0.000   
## Class :character 1st Qu.: 5.000 Class :character 1st Qu.:0.000   
## Mode :character Median : 6.000 Mode :character Median :1.000   
## Mean : 6.442 Mean :0.603   
## 3rd Qu.: 7.000 3rd Qu.:1.000   
## Max. :15.000 Max. :4.000   
## Fireplace\_Qu Garage\_Type Garage\_Finish Garage\_Cars   
## Length:2053 Length:2053 Length:2053 Min. :0.000   
## Class :character Class :character Class :character 1st Qu.:1.000   
## Mode :character Mode :character Mode :character Median :2.000   
## Mean :1.774   
## 3rd Qu.:2.000   
## Max. :5.000   
## Garage\_Area Garage\_Qual Garage\_Cond Paved\_Drive   
## Min. : 0 Length:2053 Length:2053 Length:2053   
## 1st Qu.: 320 Class :character Class :character Class :character   
## Median : 478 Mode :character Mode :character Mode :character   
## Mean : 472   
## 3rd Qu.: 576   
## Max. :1488   
## Wood\_Deck\_SF Open\_Porch\_SF Enclosed\_Porch Three\_season\_porch  
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.000   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.000   
## Median : 0.00 Median : 27.00 Median : 0.00 Median : 0.000   
## Mean : 93.52 Mean : 48.17 Mean : 23.02 Mean : 2.799   
## 3rd Qu.: 168.00 3rd Qu.: 72.00 3rd Qu.: 0.00 3rd Qu.: 0.000   
## Max. :1424.00 Max. :742.00 Max. :584.00 Max. :407.000   
## Screen\_Porch Pool\_Area Pool\_QC Fence   
## Min. : 0.00 Min. : 0.000 Length:2053 Length:2053   
## 1st Qu.: 0.00 1st Qu.: 0.000 Class :character Class :character   
## Median : 0.00 Median : 0.000 Mode :character Mode :character   
## Mean : 16.68 Mean : 1.339   
## 3rd Qu.: 0.00 3rd Qu.: 0.000   
## Max. :576.00 Max. :800.000   
## Misc\_Feature Misc\_Val Mo\_Sold Year\_Sold   
## Length:2053 Min. : 0.00 Min. : 1.000 Min. :2006   
## Class :character 1st Qu.: 0.00 1st Qu.: 4.000 1st Qu.:2007   
## Mode :character Median : 0.00 Median : 6.000 Median :2008   
## Mean : 60.12 Mean : 6.189 Mean :2008   
## 3rd Qu.: 0.00 3rd Qu.: 8.000 3rd Qu.:2009   
## Max. :17000.00 Max. :12.000 Max. :2010   
## Sale\_Type Sale\_Condition Longitude Latitude   
## Length:2053 Length:2053 Min. :-93.69 Min. :41.99   
## Class :character Class :character 1st Qu.:-93.66 1st Qu.:42.02   
## Mode :character Mode :character Median :-93.64 Median :42.03   
## Mean :-93.64 Mean :42.03   
## 3rd Qu.:-93.62 3rd Qu.:42.05   
## Max. :-93.58 Max. :42.06   
## Above\_Median   
## Length:2053   
## Class :character   
## Mode :character   
##   
##   
##

## Data Wrangling

You can also embed plots, for example:

Homes2 = Homes %>%  
 dplyr::select(MS\_Zoning, Lot\_Area, Neighborhood, Bldg\_Type, Year\_Built, Year\_Remod\_Add, Gr\_Liv\_Area, TotRms\_AbvGrd, Garage\_Cars,Above\_Median) %>% mutate(Neighborhood = as\_factor(Neighborhood)) %>% mutate(MS\_Zoning = as\_factor(MS\_Zoning)) %>% mutate(Bldg\_Type = as\_factor(Bldg\_Type)) %>% mutate(Above\_Median = as\_factor(Above\_Median))

## Data Split, Building a Classification Tree

str(Homes2)

## tibble [2,053 × 10] (S3: tbl\_df/tbl/data.frame)  
## $ MS\_Zoning : Factor w/ 7 levels "Residential\_Low\_Density",..: 1 2 1 1 1 1 1 1 1 1 ...  
## $ Lot\_Area : num [1:2053] 31770 11622 14267 11160 13830 ...  
## $ Neighborhood : Factor w/ 28 levels "North\_Ames","Gilbert",..: 1 1 1 1 2 2 3 3 2 2 ...  
## $ Bldg\_Type : Factor w/ 5 levels "OneFam","TwnhsE",..: 1 1 1 1 1 1 2 2 1 1 ...  
## $ Year\_Built : num [1:2053] 1960 1961 1958 1968 1997 ...  
## $ Year\_Remod\_Add: num [1:2053] 1960 1961 1958 1968 1998 ...  
## $ Gr\_Liv\_Area : num [1:2053] 1656 896 1329 2110 1629 ...  
## $ TotRms\_AbvGrd : num [1:2053] 7 5 6 8 6 7 5 5 6 5 ...  
## $ Garage\_Cars : num [1:2053] 2 1 1 2 2 2 2 2 2 2 ...  
## $ Above\_Median : Factor w/ 2 levels "Yes","No": 1 2 1 1 1 1 1 1 1 1 ...

set.seed(123)   
Homes\_split = initial\_split(Homes2, prop = 0.7, strata = Above\_Median) #70% in training  
train = training(Homes\_split)  
test = testing(Homes\_split)  
summary(train)

## MS\_Zoning Lot\_Area Neighborhood  
## Residential\_Low\_Density :1114 Min. : 1300 North\_Ames :227   
## Residential\_High\_Density : 13 1st Qu.: 7560 Old\_Town :135   
## Floating\_Village\_Residential: 62 Median : 9600 College\_Creek:119   
## Residential\_Medium\_Density : 230 Mean :10152 Edwards : 93   
## C\_all : 16 3rd Qu.:11700 Somerset : 82   
## A\_agr : 1 Max. :57200 Gilbert : 79   
## I\_all : 1 (Other) :702   
## Bldg\_Type Year\_Built Year\_Remod\_Add Gr\_Liv\_Area TotRms\_AbvGrd   
## OneFam :1203 Min. :1875 Min. :1950 Min. : 480 Min. : 3.000   
## TwnhsE : 103 1st Qu.:1951 1st Qu.:1964 1st Qu.:1138 1st Qu.: 5.000   
## Twnhs : 47 Median :1971 Median :1992 Median :1446 Median : 6.000   
## Duplex : 52 Mean :1970 Mean :1984 Mean :1505 Mean : 6.444   
## TwoFmCon: 32 3rd Qu.:2000 3rd Qu.:2004 3rd Qu.:1762 3rd Qu.: 7.000   
## Max. :2010 Max. :2010 Max. :5095 Max. :15.000   
##   
## Garage\_Cars Above\_Median  
## Min. :0.000 Yes:730   
## 1st Qu.:1.000 No :707   
## Median :2.000   
## Mean :1.777   
## 3rd Qu.:2.000   
## Max. :5.000   
##

summary(test)

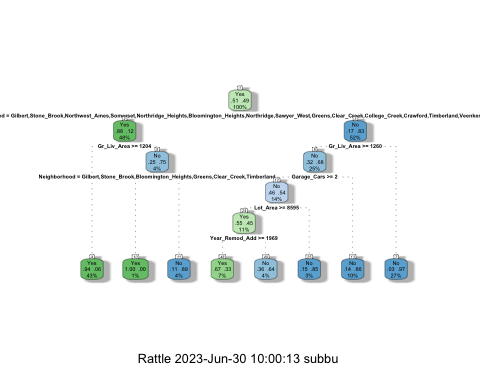
## MS\_Zoning Lot\_Area Neighborhood  
## Residential\_Low\_Density :486 Min. : 1476 North\_Ames :100   
## Residential\_High\_Density : 7 1st Qu.: 7333 College\_Creek: 64   
## Floating\_Village\_Residential: 25 Median : 9328 Old\_Town : 46   
## Residential\_Medium\_Density : 96 Mean : 10508 Somerset : 37   
## C\_all : 1 3rd Qu.: 11204 Sawyer : 37   
## A\_agr : 1 Max. :215245 Edwards : 36   
## I\_all : 0 (Other) :296   
## Bldg\_Type Year\_Built Year\_Remod\_Add Gr\_Liv\_Area TotRms\_AbvGrd   
## OneFam :503 Min. :1885 Min. :1950 Min. : 630 Min. : 3.000   
## TwnhsE : 54 1st Qu.:1956 1st Qu.:1967 1st Qu.:1136 1st Qu.: 5.000   
## Twnhs : 20 Median :1973 Median :1994 Median :1454 Median : 6.000   
## Duplex : 24 Mean :1972 Mean :1985 Mean :1482 Mean : 6.437   
## TwoFmCon: 15 3rd Qu.:2001 3rd Qu.:2004 3rd Qu.:1716 3rd Qu.: 7.000   
## Max. :2010 Max. :2010 Max. :3390 Max. :12.000   
##   
## Garage\_Cars Above\_Median  
## Min. :0.000 Yes:313   
## 1st Qu.:1.000 No :303   
## Median :2.000   
## Mean :1.766   
## 3rd Qu.:2.000   
## Max. :4.000   
##

Homes\_recipe = recipe(Above\_Median ~., train)  
  
tree\_model = decision\_tree() %>%   
 set\_engine("rpart", model = TRUE) %>%   
 set\_mode("classification")  
  
Homes\_wflow =   
 workflow() %>%   
 add\_model(tree\_model) %>%   
 add\_recipe(Homes\_recipe)  
  
Homes\_fit = fit(Homes\_wflow, train)

#extract the tree's fit from the fit object  
tree = Homes\_fit %>%   
 pull\_workflow\_fit() %>%   
 pluck("fit")

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## ℹ Please use `extract\_fit\_parsnip()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

#plot the tree  
fancyRpartPlot(tree)



Homes\_fit$fit$fit$fit$cptable

## CP nsplit rel error xerror xstd  
## 1 0.69872702 0 1.0000000 1.0381895 0.02680266  
## 2 0.04526167 1 0.3012730 0.3097595 0.01927072  
## 3 0.01414427 2 0.2560113 0.2659123 0.01808058  
## 4 0.01166902 3 0.2418670 0.2673267 0.01812134  
## 5 0.01000000 7 0.1951909 0.2616690 0.01795727

treepred = predict(Homes\_fit, train, type = "class")  
head(treepred)

## # A tibble: 6 × 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 No   
## 3 No   
## 4 No   
## 5 No   
## 6 No

## Confusion Matrix based Prediction

confusionMatrix(treepred$.pred\_class,train$Above\_Median,positive="Yes") #predictions first then actual

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 660 68  
## No 70 639  
##   
## Accuracy : 0.904   
## 95% CI : (0.8875, 0.9187)  
## No Information Rate : 0.508   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8079   
##   
## Mcnemar's Test P-Value : 0.9322   
##   
## Sensitivity : 0.9041   
## Specificity : 0.9038   
## Pos Pred Value : 0.9066   
## Neg Pred Value : 0.9013   
## Prevalence : 0.5080   
## Detection Rate : 0.4593   
## Detection Prevalence : 0.5066   
## Balanced Accuracy : 0.9040   
##   
## 'Positive' Class : Yes   
##

## Loading needed Libraries for further Analysis

library(mice) #package for imputation

##   
## Attaching package: 'mice'

## The following object is masked from 'package:stats':  
##   
## filter

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(VIM) #visualizing missingness

## Loading required package: colorspace

## Loading required package: grid

## The legacy packages maptools, rgdal, and rgeos, underpinning this package  
## will retire shortly. Please refer to R-spatial evolution reports on  
## https://r-spatial.org/r/2023/05/15/evolution4.html for details.  
## This package is now running under evolution status 0

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:rattle':  
##   
## wine

## The following object is masked from 'package:recipes':  
##   
## prepare

## The following object is masked from 'package:datasets':  
##   
## sleep

library(ranger) #for random forests

##   
## Attaching package: 'ranger'

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

library(randomForest) #also for random forests

## randomForest 4.7-1.1

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

##   
## Attaching package: 'randomForest'

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

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

## The following object is masked from 'package:dplyr':  
##   
## combine

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

library(caret)  
library(skimr)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(gridExtra)

##   
## Attaching package: 'gridExtra'

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

## The following object is masked from 'package:dplyr':  
##   
## combine

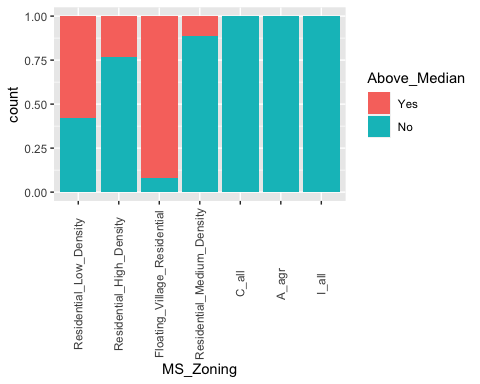
library(vip) #variable importance

##   
## Attaching package: 'vip'

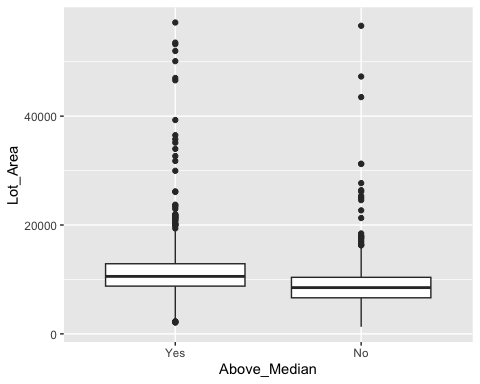
## The following object is masked from 'package:utils':  
##   
## vi

## Plotting against different Variables for further Analysis

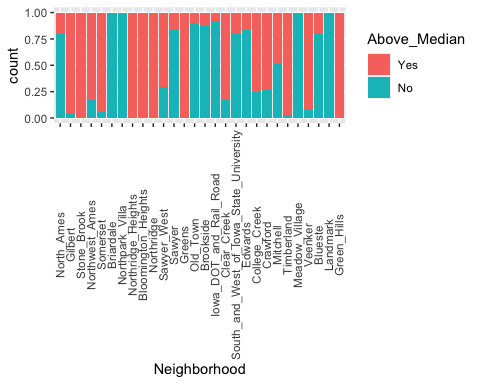
library(ggcorrplot)  
ggplot(train, aes(x = MS\_Zoning, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90,  
 vjust = 0.5,  
 hjust = 0.5))



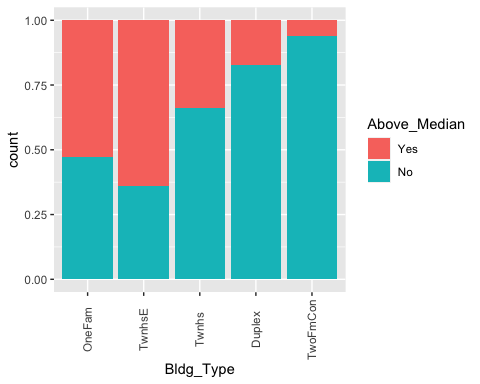
ggplot(train, aes(y = Lot\_Area, x = Above\_Median)) + geom\_boxplot()



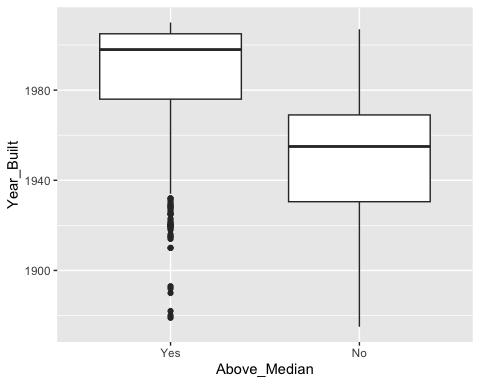
ggplot(train, aes(x = Neighborhood, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90,  
 vjust = 0.5,  
 hjust = 0.5))



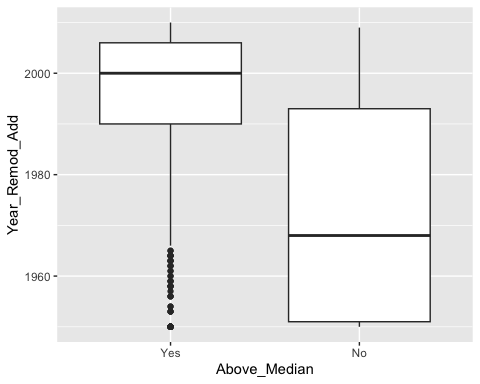
ggplot(train, aes(x = Bldg\_Type, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90,  
 vjust = 0.5,  
 hjust = 0.5))



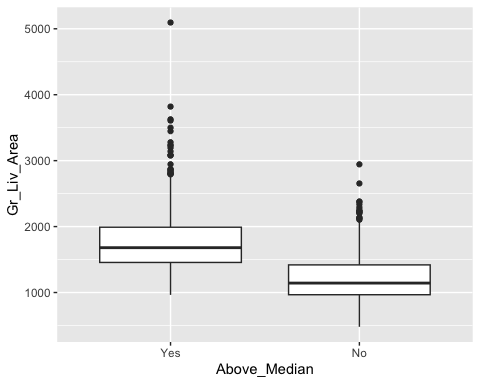
ggplot(train, aes(y = Year\_Built, x = Above\_Median)) + geom\_boxplot()



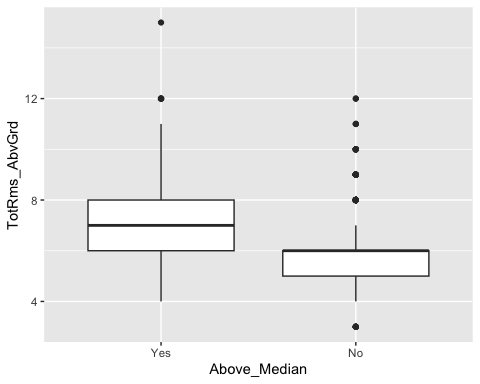
ggplot(train, aes(y = Year\_Remod\_Add, x = Above\_Median)) + geom\_boxplot()



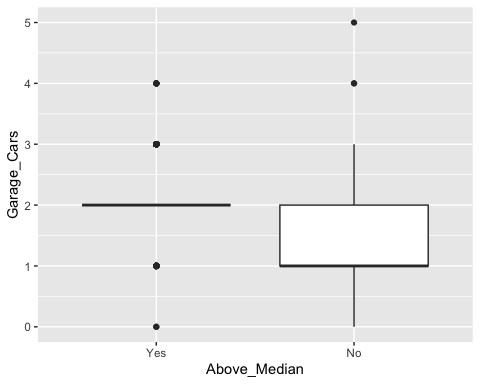
ggplot(train, aes(y = Gr\_Liv\_Area, x = Above\_Median)) + geom\_boxplot()



ggplot(train, aes(y = TotRms\_AbvGrd, x = Above\_Median)) + geom\_boxplot()



ggplot(train, aes(y = Garage\_Cars, x = Above\_Median)) + geom\_boxplot()

 ## Folds

set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

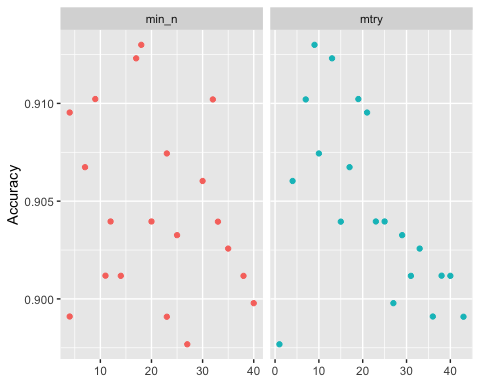
## Tuning the Model

Homes\_recipe2 = recipe(Above\_Median ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>%   
 set\_engine("ranger", importance = "permutation") %>%   
 set\_mode("classification")  
  
Homes\_wflow2 =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(Homes\_recipe2)  
  
set.seed(123)  
rf\_res = tune\_grid(  
 Homes\_wflow2,  
 resamples = rf\_folds,  
 grid = 20   
)

## i Creating pre-processing data to finalize unknown parameter: mtry

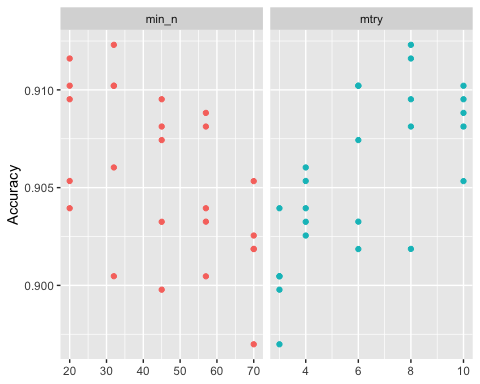
## Parameter Performance

rf\_res %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 dplyr::select(mean, min\_n, mtry) %>%  
 pivot\_longer(min\_n:mtry,  
 values\_to = "value",  
 names\_to = "parameter"  
 ) %>%  
 ggplot(aes(value, mean, color = parameter)) +  
 geom\_point(show.legend = FALSE) +  
 facet\_wrap(~parameter, scales = "free\_x") +  
 labs(x = NULL, y = "Accuracy")

 ## Refining Parameters

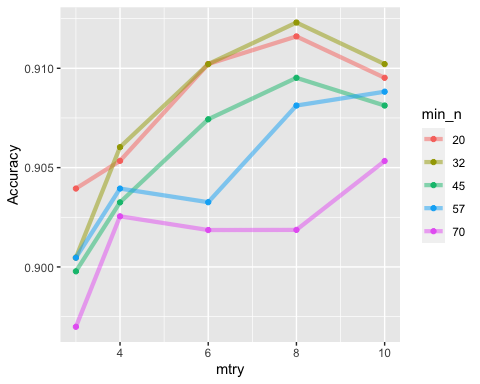
Homes\_recipe2 = recipe(Above\_Median ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>% #add tuning of mtry and min\_n parameters  
 #setting trees to 100 here should also speed things up a bit, but more trees might be better  
 set\_engine("ranger", importance = "permutation") %>% #added importance metric  
 set\_mode("classification")  
  
Homes\_wflow2 =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(Homes\_recipe2)  
  
rf\_grid = grid\_regular(  
 mtry(range = c(3, 10)), #these values determined through significant trial and error  
 min\_n(range = c(20, 70)), #these values determined through significant trial and error  
 levels = 5  
)  
  
set.seed(123)  
rf\_res\_tuned = tune\_grid(  
 Homes\_wflow2,  
 resamples = rf\_folds,  
 grid = rf\_grid #use the tuning grid  
)

rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 dplyr::select(mean, min\_n, mtry) %>%  
 pivot\_longer(min\_n:mtry,  
 values\_to = "value",  
 names\_to = "parameter"  
 ) %>%  
 ggplot(aes(value, mean, color = parameter)) +  
 geom\_point(show.legend = FALSE) +  
 facet\_wrap(~parameter, scales = "free\_x") +  
 labs(x = NULL, y = "Accuracy")

 ## Alternate View of the Parameters

rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 mutate(min\_n = factor(min\_n)) %>%  
 ggplot(aes(mtry, mean, color = min\_n)) +  
 geom\_line(alpha = 0.5, size = 1.5) +  
 geom\_point() +  
 labs(y = "Accuracy")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 Homes\_wflow2,  
 best\_rf  
)  
  
final\_rf

## ══ Workflow ════════════════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 1 Recipe Step  
##   
## • step\_dummy()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 8  
## trees = 100  
## min\_n = 32  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

#fit the finalized workflow to our training data  
final\_rf\_fit = fit(final\_rf, train)

## Checking Variables Importance

final\_rf\_fit %>% pull\_workflow\_fit() %>% vip(geom = "point")

 ## Predictions

trainpredrf = predict(final\_rf\_fit, train)  
head(trainpredrf)

## # A tibble: 6 × 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 No   
## 3 No   
## 4 No   
## 5 No   
## 6 No

## Confusion Matrix

confusionMatrix(trainpredrf$.pred\_class, train$Above\_Median,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 686 37  
## No 44 670  
##   
## Accuracy : 0.9436   
## 95% CI : (0.9304, 0.955)  
## No Information Rate : 0.508   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8873   
##   
## Mcnemar's Test P-Value : 0.505   
##   
## Sensitivity : 0.9397   
## Specificity : 0.9477   
## Pos Pred Value : 0.9488   
## Neg Pred Value : 0.9384   
## Prevalence : 0.5080   
## Detection Rate : 0.4774   
## Detection Prevalence : 0.5031   
## Balanced Accuracy : 0.9437   
##   
## 'Positive' Class : Yes   
##

## Predictions on Test Data

treepred\_test = predict(Homes\_fit, test, type = "class")  
head(treepred\_test)

## # A tibble: 6 × 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 No   
## 3 Yes   
## 4 Yes   
## 5 Yes   
## 6 Yes

confusionMatrix(treepred\_test$.pred\_class,test$Above\_Median,positive="Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 270 29  
## No 43 274  
##   
## Accuracy : 0.8831   
## 95% CI : (0.8551, 0.9074)  
## No Information Rate : 0.5081   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.7663   
##   
## Mcnemar's Test P-Value : 0.1255   
##   
## Sensitivity : 0.8626   
## Specificity : 0.9043   
## Pos Pred Value : 0.9030   
## Neg Pred Value : 0.8644   
## Prevalence : 0.5081   
## Detection Rate : 0.4383   
## Detection Prevalence : 0.4854   
## Balanced Accuracy : 0.8835   
##   
## 'Positive' Class : Yes   
##