GBM Analysis

Anonymous

2022-08-22

## Introduction

### Load packages and data

load("resit\_data.RData")  
df <- resit\_data # COPY to a global variablelibrary(caret)  
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.3

## Loading required package: lattice

library(gbm)

## Warning: package 'gbm' was built under R version 4.1.3

## Loaded gbm 2.1.8.1

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.6 v dplyr 1.0.8  
## v tidyr 1.2.0 v stringr 1.4.0  
## v readr 2.1.2 v forcats 0.5.1  
## v purrr 0.3.4

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'purrr' was built under R version 4.1.3

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()

head(df) # display the first rows of the data

## # A tibble: 6 x 8  
## meanPrice gs\_area u16 u25 u45 u65 o65 unmplyd  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 152. 2.28 5.62 20.9 45.9 26.2 1.25 18.1   
## 2 105. 4.01 3.75 14.4 56.5 17.6 7.78 9.73  
## 3 115. 3.26 5.11 12.8 64.5 14.4 3.19 6.72  
## 4 368 0 9.47 8.08 38.7 23.7 20.1 10.1   
## 5 228. 0 8.38 5.76 54.5 26.2 5.24 10.6   
## 6 554. 0 9.24 3.50 27.1 22.6 37.6 4.35

### Drop the NA values from the data for analysis

df <- df[!is.na(df)]

### Split the data to train and test

dpart <- createDataPartition(resit\_data$meanPrice, p = 0.7, list = F)  
traindata <- resit\_data[dpart, ]  
test <- resit\_data[-dpart,]  
print("Train")

## [1] "Train"

print(nrow(traindata))

## [1] 663

print("Test")

## [1] "Test"

print(nrow(test))

## [1] 281

View the GBM help

modelLookup("gbm")

## model parameter label forReg forClass probModel  
## 1 gbm n.trees # Boosting Iterations TRUE TRUE TRUE  
## 2 gbm interaction.depth Max Tree Depth TRUE TRUE TRUE  
## 3 gbm shrinkage Shrinkage TRUE TRUE TRUE  
## 4 gbm n.minobsinnode Min. Terminal Node Size TRUE TRUE TRUE

### GBM model tuning

caretGrid <- expand.grid(interaction.depth=c(1, 3, 5), n.trees = (0:50)\*50,  
shrinkage=c(0.01, 0.001),  
n.minobsinnode=10)  
metric <- "RMSE"

### Create a Train Control for model sampling

The statement below creates a 10-fold cross validation train control

trainControl <- trainControl(method="cv", number=10)

Run the method over the grid

set.seed(99)  
df <- traindata  
gbm.caret <- train(meanPrice ~ ., data=traindata, distribution="gaussian", method="gbm",  
trControl=trainControl, verbose=FALSE,  
tuneGrid=caretGrid, metric=metric, bag.fraction=0.7)

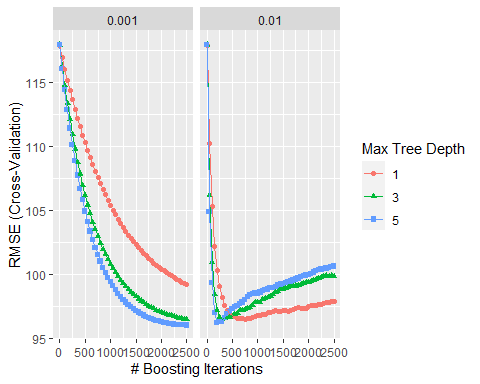
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

### Examine the results

## Examine the results  
print(gbm.caret)

## Stochastic Gradient Boosting   
##   
## 663 samples  
## 7 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 597, 596, 598, 596, 596, 598, ...   
## Resampling results across tuning parameters:  
##   
## shrinkage interaction.depth n.trees RMSE Rsquared MAE   
## 0.001 1 0 117.90409 NaN 83.22907  
## 0.001 1 50 116.93105 0.2474178 82.33075  
## 0.001 1 100 116.02294 0.2541999 81.49346  
## 0.001 1 150 115.16736 0.2636904 80.74516  
## 0.001 1 200 114.36752 0.2708587 80.05440  
## 0.001 1 250 113.62955 0.2759638 79.43556  
## 0.001 1 300 112.89612 0.2817533 78.82654  
## 0.001 1 350 112.20694 0.2863928 78.26204  
## 0.001 1 400 111.52378 0.2920089 77.71154  
## 0.001 1 450 110.88841 0.2949560 77.18876  
## 0.001 1 500 110.27384 0.2976009 76.68031  
## 0.001 1 550 109.68480 0.3010453 76.22565  
## 0.001 1 600 109.11357 0.3042141 75.80448  
## 0.001 1 650 108.58209 0.3061186 75.37179  
## 0.001 1 700 108.05145 0.3094487 74.97180  
## 0.001 1 750 107.55069 0.3117871 74.57768  
## 0.001 1 800 107.08110 0.3135450 74.20376  
## 0.001 1 850 106.63737 0.3150027 73.86236  
## 0.001 1 900 106.21485 0.3164261 73.52447  
## 0.001 1 950 105.79165 0.3180808 73.19528  
## 0.001 1 1000 105.41132 0.3190855 72.90602  
## 0.001 1 1050 105.02584 0.3204344 72.60726  
## 0.001 1 1100 104.66945 0.3214505 72.33627  
## 0.001 1 1150 104.31171 0.3228197 72.06685  
## 0.001 1 1200 103.97836 0.3238196 71.83583  
## 0.001 1 1250 103.64253 0.3252011 71.59885  
## 0.001 1 1300 103.35621 0.3259497 71.38996  
## 0.001 1 1350 103.08544 0.3266268 71.18976  
## 0.001 1 1400 102.82410 0.3273490 70.99134  
## 0.001 1 1450 102.57379 0.3281484 70.80999  
## 0.001 1 1500 102.32237 0.3289862 70.62914  
## 0.001 1 1550 102.08985 0.3296860 70.46349  
## 0.001 1 1600 101.88414 0.3300255 70.32554  
## 0.001 1 1650 101.67867 0.3304621 70.17632  
## 0.001 1 1700 101.46110 0.3313183 70.03508  
## 0.001 1 1750 101.28118 0.3316711 69.90192  
## 0.001 1 1800 101.09243 0.3322003 69.77282  
## 0.001 1 1850 100.91190 0.3327594 69.64934  
## 0.001 1 1900 100.74144 0.3333802 69.52393  
## 0.001 1 1950 100.57953 0.3340159 69.41252  
## 0.001 1 2000 100.42308 0.3345393 69.30426  
## 0.001 1 2050 100.29206 0.3347215 69.20875  
## 0.001 1 2100 100.14816 0.3351614 69.11823  
## 0.001 1 2150 100.03140 0.3352381 69.04238  
## 0.001 1 2200 99.90234 0.3355754 68.96085  
## 0.001 1 2250 99.76784 0.3361275 68.87481  
## 0.001 1 2300 99.64713 0.3366091 68.80262  
## 0.001 1 2350 99.53076 0.3370877 68.73169  
## 0.001 1 2400 99.41519 0.3375267 68.65812  
## 0.001 1 2450 99.31484 0.3376695 68.58997  
## 0.001 1 2500 99.21764 0.3379742 68.51929  
## 0.001 3 0 117.90409 NaN 83.22907  
## 0.001 3 50 116.26997 0.3284612 81.90732  
## 0.001 3 100 114.77169 0.3278187 80.70007  
## 0.001 3 150 113.37624 0.3289095 79.55467  
## 0.001 3 200 112.08121 0.3304380 78.48966  
## 0.001 3 250 110.89279 0.3312157 77.49795  
## 0.001 3 300 109.79581 0.3322950 76.59775  
## 0.001 3 350 108.77667 0.3340326 75.76006  
## 0.001 3 400 107.85137 0.3344144 74.97862  
## 0.001 3 450 106.97538 0.3356378 74.26467  
## 0.001 3 500 106.17351 0.3368439 73.62838  
## 0.001 3 550 105.43358 0.3375687 73.05205  
## 0.001 3 600 104.75388 0.3385502 72.53027  
## 0.001 3 650 104.08481 0.3402425 72.02047  
## 0.001 3 700 103.51173 0.3405793 71.59697  
## 0.001 3 750 102.94551 0.3416725 71.16612  
## 0.001 3 800 102.44287 0.3425802 70.78079  
## 0.001 3 850 101.99497 0.3430376 70.44543  
## 0.001 3 900 101.57125 0.3434203 70.12865  
## 0.001 3 950 101.18686 0.3437389 69.82642  
## 0.001 3 1000 100.81682 0.3444787 69.53482  
## 0.001 3 1050 100.49812 0.3444211 69.25386  
## 0.001 3 1100 100.15804 0.3454401 68.97896  
## 0.001 3 1150 99.84931 0.3463808 68.72433  
## 0.001 3 1200 99.56776 0.3470803 68.48046  
## 0.001 3 1250 99.30755 0.3476147 68.25183  
## 0.001 3 1300 99.07946 0.3479344 68.04155  
## 0.001 3 1350 98.85718 0.3483883 67.83062  
## 0.001 3 1400 98.63699 0.3491801 67.63205  
## 0.001 3 1450 98.43495 0.3499740 67.44309  
## 0.001 3 1500 98.27008 0.3502333 67.28842  
## 0.001 3 1550 98.10687 0.3507384 67.12983  
## 0.001 3 1600 97.94642 0.3513432 66.97436  
## 0.001 3 1650 97.80221 0.3518692 66.82942  
## 0.001 3 1700 97.66649 0.3523488 66.68652  
## 0.001 3 1750 97.53507 0.3529381 66.54671  
## 0.001 3 1800 97.44909 0.3528992 66.43680  
## 0.001 3 1850 97.33382 0.3534318 66.31549  
## 0.001 3 1900 97.23621 0.3538443 66.20004  
## 0.001 3 1950 97.14533 0.3541504 66.09656  
## 0.001 3 2000 97.05272 0.3546873 65.99718  
## 0.001 3 2050 96.97618 0.3549908 65.90681  
## 0.001 3 2100 96.89604 0.3554167 65.81222  
## 0.001 3 2150 96.83691 0.3556610 65.72506  
## 0.001 3 2200 96.78777 0.3557735 65.65434  
## 0.001 3 2250 96.72727 0.3561736 65.58504  
## 0.001 3 2300 96.67568 0.3564621 65.51832  
## 0.001 3 2350 96.60733 0.3570691 65.43450  
## 0.001 3 2400 96.56485 0.3573393 65.37485  
## 0.001 3 2450 96.51969 0.3577212 65.32037  
## 0.001 3 2500 96.48177 0.3580553 65.26820  
## 0.001 5 0 117.90409 NaN 83.22907  
## 0.001 5 50 116.05599 0.3499897 81.77332  
## 0.001 5 100 114.37598 0.3507158 80.45995  
## 0.001 5 150 112.82364 0.3520530 79.20831  
## 0.001 5 200 111.39790 0.3513319 78.04634  
## 0.001 5 250 110.09439 0.3519262 76.96089  
## 0.001 5 300 108.87787 0.3523023 75.96874  
## 0.001 5 350 107.73251 0.3534757 75.03166  
## 0.001 5 400 106.70568 0.3541178 74.16307  
## 0.001 5 450 105.79478 0.3532511 73.38562  
## 0.001 5 500 104.91907 0.3538212 72.63830  
## 0.001 5 550 104.11340 0.3541601 71.97904  
## 0.001 5 600 103.37395 0.3549964 71.40438  
## 0.001 5 650 102.71172 0.3553318 70.86895  
## 0.001 5 700 102.10989 0.3554764 70.38746  
## 0.001 5 750 101.54594 0.3562250 69.95024  
## 0.001 5 800 101.03045 0.3564934 69.54022  
## 0.001 5 850 100.54494 0.3573619 69.15050  
## 0.001 5 900 100.12323 0.3575033 68.80072  
## 0.001 5 950 99.75756 0.3572489 68.47607  
## 0.001 5 1000 99.41618 0.3573582 68.17514  
## 0.001 5 1050 99.09116 0.3576868 67.89212  
## 0.001 5 1100 98.78664 0.3580133 67.60814  
## 0.001 5 1150 98.51330 0.3582721 67.35539  
## 0.001 5 1200 98.25482 0.3585545 67.11207  
## 0.001 5 1250 98.02565 0.3589667 66.88308  
## 0.001 5 1300 97.83511 0.3589245 66.68410  
## 0.001 5 1350 97.64906 0.3591007 66.48657  
## 0.001 5 1400 97.46182 0.3596197 66.29222  
## 0.001 5 1450 97.31804 0.3595961 66.13303  
## 0.001 5 1500 97.17846 0.3597650 65.97176  
## 0.001 5 1550 97.06174 0.3597071 65.82889  
## 0.001 5 1600 96.93676 0.3600378 65.68637  
## 0.001 5 1650 96.82323 0.3604734 65.55552  
## 0.001 5 1700 96.73048 0.3606144 65.43088  
## 0.001 5 1750 96.64060 0.3609520 65.31227  
## 0.001 5 1800 96.56475 0.3610146 65.20382  
## 0.001 5 1850 96.49245 0.3612664 65.10471  
## 0.001 5 1900 96.41632 0.3615587 65.01821  
## 0.001 5 1950 96.37471 0.3614446 64.95028  
## 0.001 5 2000 96.31496 0.3617727 64.87641  
## 0.001 5 2050 96.26468 0.3619759 64.79912  
## 0.001 5 2100 96.21815 0.3622153 64.73097  
## 0.001 5 2150 96.17100 0.3625390 64.67666  
## 0.001 5 2200 96.14835 0.3626265 64.63236  
## 0.001 5 2250 96.11588 0.3629152 64.57661  
## 0.001 5 2300 96.08640 0.3631599 64.52356  
## 0.001 5 2350 96.06908 0.3632878 64.47920  
## 0.001 5 2400 96.05440 0.3633827 64.43972  
## 0.001 5 2450 96.05053 0.3633920 64.40762  
## 0.001 5 2500 96.04468 0.3634461 64.36177  
## 0.010 1 0 117.90409 NaN 83.22907  
## 0.010 1 50 110.25869 0.2983524 76.74313  
## 0.010 1 100 105.31127 0.3199384 72.85319  
## 0.010 1 150 102.22096 0.3293349 70.61374  
## 0.010 1 200 100.35664 0.3348526 69.24806  
## 0.010 1 250 99.11558 0.3385012 68.41723  
## 0.010 1 300 98.18827 0.3431754 67.75701  
## 0.010 1 350 97.56070 0.3468943 67.30268  
## 0.010 1 400 97.24315 0.3490900 67.01166  
## 0.010 1 450 96.99607 0.3499570 66.81136  
## 0.010 1 500 96.82504 0.3512739 66.64886  
## 0.010 1 550 96.71210 0.3523482 66.53688  
## 0.010 1 600 96.61572 0.3536674 66.46109  
## 0.010 1 650 96.60024 0.3538360 66.40019  
## 0.010 1 700 96.58064 0.3542331 66.36381  
## 0.010 1 750 96.51744 0.3552535 66.30799  
## 0.010 1 800 96.60484 0.3546171 66.30507  
## 0.010 1 850 96.60825 0.3549759 66.28103  
## 0.010 1 900 96.65049 0.3549372 66.26139  
## 0.010 1 950 96.74951 0.3541518 66.33444  
## 0.010 1 1000 96.79583 0.3536149 66.32101  
## 0.010 1 1050 96.80932 0.3537713 66.32820  
## 0.010 1 1100 96.89515 0.3530194 66.36923  
## 0.010 1 1150 96.93279 0.3529209 66.37536  
## 0.010 1 1200 97.01843 0.3520963 66.42864  
## 0.010 1 1250 97.02650 0.3523803 66.45599  
## 0.010 1 1300 97.11049 0.3516204 66.50795  
## 0.010 1 1350 97.19243 0.3508715 66.54422  
## 0.010 1 1400 97.14150 0.3513625 66.51417  
## 0.010 1 1450 97.16835 0.3513039 66.46794  
## 0.010 1 1500 97.19952 0.3510650 66.46814  
## 0.010 1 1550 97.17733 0.3509445 66.45428  
## 0.010 1 1600 97.11988 0.3518259 66.41077  
## 0.010 1 1650 97.18641 0.3512741 66.44023  
## 0.010 1 1700 97.26846 0.3505893 66.44940  
## 0.010 1 1750 97.34196 0.3497728 66.50959  
## 0.010 1 1800 97.42467 0.3490395 66.56722  
## 0.010 1 1850 97.40354 0.3494049 66.57935  
## 0.010 1 1900 97.36810 0.3496867 66.52925  
## 0.010 1 1950 97.40333 0.3494860 66.53418  
## 0.010 1 2000 97.50106 0.3488020 66.59881  
## 0.010 1 2050 97.56587 0.3480786 66.65725  
## 0.010 1 2100 97.63409 0.3474196 66.68843  
## 0.010 1 2150 97.63346 0.3473861 66.66698  
## 0.010 1 2200 97.64231 0.3473366 66.69633  
## 0.010 1 2250 97.66989 0.3468536 66.73720  
## 0.010 1 2300 97.74669 0.3460548 66.73979  
## 0.010 1 2350 97.79963 0.3460248 66.77142  
## 0.010 1 2400 97.83517 0.3452648 66.80673  
## 0.010 1 2450 97.90690 0.3447878 66.83553  
## 0.010 1 2500 97.94857 0.3446924 66.84997  
## 0.010 3 0 117.90409 NaN 83.22907  
## 0.010 3 50 106.14840 0.3347908 73.60728  
## 0.010 3 100 100.92202 0.3400756 69.50071  
## 0.010 3 150 98.44213 0.3467024 67.32576  
## 0.010 3 200 97.20551 0.3525220 66.13387  
## 0.010 3 250 96.67348 0.3561334 65.40358  
## 0.010 3 300 96.51932 0.3571919 64.94594  
## 0.010 3 350 96.57566 0.3569672 64.75637  
## 0.010 3 400 96.65708 0.3575861 64.62554  
## 0.010 3 450 96.66255 0.3597120 64.41338  
## 0.010 3 500 96.70910 0.3604948 64.29812  
## 0.010 3 550 96.88454 0.3592368 64.35577  
## 0.010 3 600 96.96754 0.3588510 64.31323  
## 0.010 3 650 97.17736 0.3574448 64.30482  
## 0.010 3 700 97.22705 0.3576220 64.28492  
## 0.010 3 750 97.26692 0.3575466 64.22781  
## 0.010 3 800 97.37462 0.3563702 64.27030  
## 0.010 3 850 97.45911 0.3558820 64.29267  
## 0.010 3 900 97.63014 0.3544201 64.37293  
## 0.010 3 950 97.80043 0.3529116 64.46219  
## 0.010 3 1000 97.81577 0.3530348 64.44573  
## 0.010 3 1050 97.85049 0.3531229 64.43545  
## 0.010 3 1100 98.05935 0.3510421 64.50627  
## 0.010 3 1150 98.16275 0.3499735 64.59681  
## 0.010 3 1200 98.21212 0.3496928 64.58191  
## 0.010 3 1250 98.28292 0.3489764 64.62218  
## 0.010 3 1300 98.44455 0.3472704 64.69494  
## 0.010 3 1350 98.60945 0.3458281 64.80256  
## 0.010 3 1400 98.68103 0.3452780 64.87929  
## 0.010 3 1450 98.84100 0.3436648 64.93348  
## 0.010 3 1500 98.83443 0.3437062 64.92753  
## 0.010 3 1550 98.91702 0.3432649 64.96822  
## 0.010 3 1600 99.00397 0.3427787 65.00246  
## 0.010 3 1650 99.13232 0.3419566 65.03507  
## 0.010 3 1700 99.11443 0.3421753 65.03599  
## 0.010 3 1750 99.11624 0.3426596 65.04437  
## 0.010 3 1800 99.17153 0.3416657 65.08531  
## 0.010 3 1850 99.23395 0.3411950 65.10633  
## 0.010 3 1900 99.27892 0.3410558 65.15974  
## 0.010 3 1950 99.35900 0.3412451 65.19551  
## 0.010 3 2000 99.41338 0.3406698 65.22040  
## 0.010 3 2050 99.49469 0.3400342 65.27659  
## 0.010 3 2100 99.58150 0.3391565 65.30324  
## 0.010 3 2150 99.59523 0.3395770 65.32375  
## 0.010 3 2200 99.71776 0.3385240 65.37781  
## 0.010 3 2250 99.77414 0.3383236 65.44269  
## 0.010 3 2300 99.85345 0.3379765 65.45211  
## 0.010 3 2350 99.95695 0.3371413 65.52689  
## 0.010 3 2400 99.88968 0.3376453 65.52748  
## 0.010 3 2450 99.89504 0.3375843 65.53527  
## 0.010 3 2500 99.88953 0.3380593 65.53587  
## 0.010 5 0 117.90409 NaN 83.22907  
## 0.010 5 50 104.89004 0.3514744 72.62905  
## 0.010 5 100 99.35755 0.3592481 68.11049  
## 0.010 5 150 97.03830 0.3619015 65.72665  
## 0.010 5 200 96.25868 0.3615125 64.64040  
## 0.010 5 250 96.28587 0.3592019 64.32602  
## 0.010 5 300 96.30994 0.3598116 64.06144  
## 0.010 5 350 96.63233 0.3580703 64.02455  
## 0.010 5 400 96.91479 0.3568775 64.07542  
## 0.010 5 450 97.08497 0.3566421 63.99042  
## 0.010 5 500 97.33247 0.3550710 64.03273  
## 0.010 5 550 97.50509 0.3544746 64.04388  
## 0.010 5 600 97.53178 0.3554206 64.05257  
## 0.010 5 650 97.72706 0.3531558 64.12163  
## 0.010 5 700 97.83046 0.3525973 64.18528  
## 0.010 5 750 98.03529 0.3509729 64.30408  
## 0.010 5 800 98.24194 0.3488554 64.39173  
## 0.010 5 850 98.41245 0.3479547 64.45254  
## 0.010 5 900 98.47482 0.3478594 64.45618  
## 0.010 5 950 98.55951 0.3473370 64.48431  
## 0.010 5 1000 98.53494 0.3480131 64.50727  
## 0.010 5 1050 98.62077 0.3475304 64.50297  
## 0.010 5 1100 98.74786 0.3467012 64.56266  
## 0.010 5 1150 98.78811 0.3462551 64.59665  
## 0.010 5 1200 98.85167 0.3459449 64.68160  
## 0.010 5 1250 98.93843 0.3450517 64.70341  
## 0.010 5 1300 98.98571 0.3449642 64.71311  
## 0.010 5 1350 99.05176 0.3448602 64.73814  
## 0.010 5 1400 99.16647 0.3437620 64.79910  
## 0.010 5 1450 99.23332 0.3431447 64.85388  
## 0.010 5 1500 99.25946 0.3429448 64.90724  
## 0.010 5 1550 99.32622 0.3426782 64.98670  
## 0.010 5 1600 99.40053 0.3419860 65.02767  
## 0.010 5 1650 99.46258 0.3418274 65.08886  
## 0.010 5 1700 99.54915 0.3406649 65.18016  
## 0.010 5 1750 99.62440 0.3400492 65.25046  
## 0.010 5 1800 99.70512 0.3394756 65.29360  
## 0.010 5 1850 99.83051 0.3383950 65.39896  
## 0.010 5 1900 99.88003 0.3384134 65.41559  
## 0.010 5 1950 99.94231 0.3378514 65.52346  
## 0.010 5 2000 99.99754 0.3376288 65.54221  
## 0.010 5 2050 100.05284 0.3372925 65.55145  
## 0.010 5 2100 100.17423 0.3365876 65.62230  
## 0.010 5 2150 100.27706 0.3357798 65.75343  
## 0.010 5 2200 100.35611 0.3348892 65.80226  
## 0.010 5 2250 100.40236 0.3344585 65.84617  
## 0.010 5 2300 100.47294 0.3337382 65.91099  
## 0.010 5 2350 100.48707 0.3338152 65.96504  
## 0.010 5 2400 100.54155 0.3337440 65.99459  
## 0.010 5 2450 100.60322 0.3333717 66.07773  
## 0.010 5 2500 100.63220 0.3332372 66.11121  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were n.trees = 2500, interaction.depth =  
## 5, shrinkage = 0.001 and n.minobsinnode = 10.

ggplot(gbm.caret)



# explore the results  
names(gbm.caret)

## [1] "method" "modelInfo" "modelType" "results" "pred"   
## [6] "bestTune" "call" "dots" "metric" "control"   
## [11] "finalModel" "preProcess" "trainingData" "ptype" "resample"   
## [16] "resampledCM" "perfNames" "maximize" "yLimits" "times"   
## [21] "levels" "terms" "coefnames" "xlevels"

# see best tune  
gbm.caret[6]

## $bestTune  
## n.trees interaction.depth shrinkage n.minobsinnode  
## 153 2500 5 0.001 10

### View the grid results

# see grid results  
head(data.frame(gbm.caret[4]))

## results.shrinkage results.interaction.depth results.n.minobsinnode  
## 1 0.001 1 10  
## 154 0.010 1 10  
## 52 0.001 3 10  
## 205 0.010 3 10  
## 103 0.001 5 10  
## 256 0.010 5 10  
## results.n.trees results.RMSE results.Rsquared results.MAE results.RMSESD  
## 1 0 117.9041 NaN 83.22907 27.42302  
## 154 0 117.9041 NaN 83.22907 27.42302  
## 52 0 117.9041 NaN 83.22907 27.42302  
## 205 0 117.9041 NaN 83.22907 27.42302  
## 103 0 117.9041 NaN 83.22907 27.42302  
## 256 0 117.9041 NaN 83.22907 27.42302  
## results.RsquaredSD results.MAESD  
## 1 NA 9.103217  
## 154 NA 9.103217  
## 52 NA 9.103217  
## 205 NA 9.103217  
## 103 NA 9.103217  
## 256 NA 9.103217

# check  
dim(caretGrid)

## [1] 306 4

dim(data.frame(gbm.caret[4]))

## [1] 306 10

### Examine the best result

Find the best parameter combination and put it in a dataframe

grid\_df = data.frame(gbm.caret[4])  
head(grid\_df)

## results.shrinkage results.interaction.depth results.n.minobsinnode  
## 1 0.001 1 10  
## 154 0.010 1 10  
## 52 0.001 3 10  
## 205 0.010 3 10  
## 103 0.001 5 10  
## 256 0.010 5 10  
## results.n.trees results.RMSE results.Rsquared results.MAE results.RMSESD  
## 1 0 117.9041 NaN 83.22907 27.42302  
## 154 0 117.9041 NaN 83.22907 27.42302  
## 52 0 117.9041 NaN 83.22907 27.42302  
## 205 0 117.9041 NaN 83.22907 27.42302  
## 103 0 117.9041 NaN 83.22907 27.42302  
## 256 0 117.9041 NaN 83.22907 27.42302  
## results.RsquaredSD results.MAESD  
## 1 NA 9.103217  
## 154 NA 9.103217  
## 52 NA 9.103217  
## 205 NA 9.103217  
## 103 NA 9.103217  
## 256 NA 9.103217

### confirm best model and assign to params object

grid\_df[which.min(grid\_df$results.RMSE), ]

## results.shrinkage results.interaction.depth results.n.minobsinnode  
## 153 0.001 5 10  
## results.n.trees results.RMSE results.Rsquared results.MAE results.RMSESD  
## 153 2500 96.04468 0.3634461 64.36177 28.28309  
## results.RsquaredSD results.MAESD  
## 153 0.1471327 11.54779

### Assign the parameters and inspect

params = grid\_df[which.min(grid\_df$results.RMSE), 1:4 ]  
params

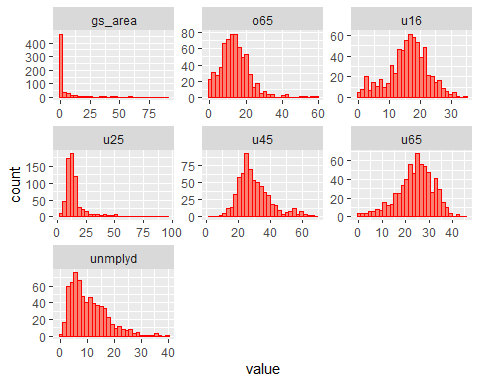
## results.shrinkage results.interaction.depth results.n.minobsinnode  
## 153 0.001 5 10  
## results.n.trees  
## 153 2500

### Examine the Data

The figures below show the distribution of the numeric values using histograms

df %>% mutate(ID = 1:n()) %>% as\_tibble() %>%  
select( -meanPrice) %>%  
pivot\_longer(-ID) %>%  
ggplot(aes(x = value)) + geom\_histogram(col = "red", fill = "salmon") +  
facet\_wrap(~name, scales = "free")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Train our model

## Create final model  
# because parameters are known, model can be fit without parameter tuning  
fitControl <- trainControl(method = "none", classProbs = FALSE)  
# extract the values from params  
gbmFit <- train(meanPrice ~ ., data=df, distribution="gaussian", method = "gbm",  
trControl = fitControl,  
verbose = FALSE,  
## only a single model is passed to the  
tuneGrid = data.frame(interaction.depth = 3,  
n.trees = 750,  
shrinkage = .01,  
n.minobsinnode = 10),  
metric = metric)

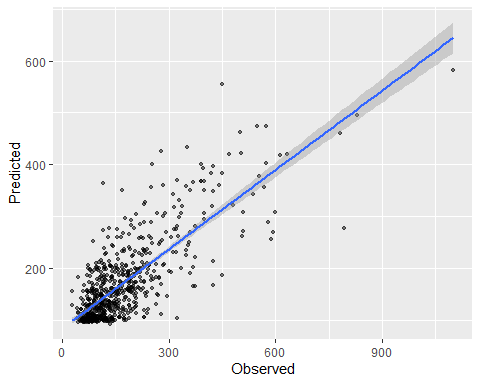
### Predict the Mean price

## Create final model  
# because parameters are known, model can be fit without parameter tuning  
fitControl <- trainControl(method = "none", classProbs = FALSE)  
# extract the values from params  
gbmFit <- train(meanPrice ~ ., data=df, distribution="gaussian", method = "gbm",  
trControl = fitControl,  
verbose = FALSE,  
## only a single model is passed to the  
tuneGrid = data.frame(interaction.depth = 3,  
n.trees = 750,  
shrinkage = .01,  
n.minobsinnode = 10),  
metric = metric)

## Prediction and Model evaluation

# generate predictions  
pred = predict(gbmFit, newdata = df)  
# plot these against observed  
data.frame(Predicted = pred, Observed = df$meanPrice) %>%  
ggplot(aes(x = Observed, y = Predicted))+ geom\_point(size = 1, alpha = 0.5)+  
geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'

 ### Get the prediction Accuracy

# generate some prediction accuracy measures  
postResample(pred = pred, obs = test$meanPrice)

## RMSE Rsquared MAE   
## NA 0.0007076505 NA

### Examine the Important Variables

varImp(gbmFit, scale = FALSE)

## gbm variable importance  
##   
## Overall  
## unmplyd 71633683  
## u45 51171511  
## u65 47156732  
## o65 22982348  
## u25 21155569  
## u16 19400772  
## gs\_area 10698052