

Final Project: Housing Data

Jacob Minkin

May 19, 2021

Intro

We start this project by loading the data from GitHub.

```
rm(list = ls())
pacman::p_load(tidyverse, data.table, R.utils, stringr, missForest, YARF, mlr )

## YARF can now make use of 3 cores.
options(java.parameters = "-Xmx8000m")
housing_data = fread("https://raw.githubusercontent.com/kapelner/QC_MATH_342W_Spring_2021/master/writing/
Northeast_Queens_zip = c(11361, 11362, 11363, 11364)
North_Queens_zip = c(11354, 11355, 11356, 11357, 11358, 11359, 11360)
Central_Queens_zip = c(11365, 11366, 11367)
Jamaica_zip = c(11412, 11423, 11432, 11433, 11434, 11435, 11436)
Northwest_Queens_zip = c(11101, 11102, 11103, 11104, 11105, 11106)
West_Central_Queens_zip = c(11374, 11375, 11379, 11385)
Southeast_Queens_zip = c(11004, 11005, 11411, 11413, 11422, 11426, 11427, 11428, 11429)
Southwest_Queens_zip = c(11414, 11415, 11416, 11417, 11418, 11419, 11420, 11421)
West_Queens_zip = c(11368, 11369, 11370, 11372, 11373, 11377, 11378)
```

Remove Garbage

First we are only predicting on data that includes the sale price. We remove all data that does not include the sale price.

The important features start with 'approx_year_built'. Remove 'url', 'common_charges', 'model_type', 'date_of_sale', and 'listing_price_to_nearest_1000'. Also remove 'sq_footage', 'parking_charge' and 'total_taxes' because too much is missing.

```
no_garabge_data = housing_data %>%
  # filter(!is.na(sale_price)) %>%
  select(approx_year_built:last_col(), -url, -common_charges, -model_type, -date_of_sale, -listing_price)

table(no_garabge_data$coop_condo, exclude = NaN)

##
## co-op condo
## 1661 569
```

Manipulating Data

Here is a list of manipulations 1. Pets allowed is a combo of cats and dogs allowed. if one pet is allowed then pets allowed is true. 2. fuel type is cleaned by combining others. Some were capitalized. 3. sq

```
manipulated_data = no_garabge_data %>%

  mutate(cats_allowed = if_else(cats_allowed == "y", "yes", cats_allowed)) %>%
  mutate(dogs_allowed = if_else(dogs_allowed == "yes89", "yes", dogs_allowed)) %>%
  mutate(pets_allowed = if_else(dogs_allowed == "no" & cats_allowed == "no", "no", "yes")) %>%

  mutate(fuel_type = if_else(fuel_type == "Other", "other", fuel_type)) %>%

  mutate(sq_footage = as.factor(case_when(sq_footage %in% 100:600 ~ "small", sq_footage %in% 600:1000 ~ "medium", sq_footage %in% 1000:1500 ~ "big", sq_footage %in% 1500:2000 ~ "big2", sq_footage %in% 2000:2500 ~ "big3", sq_footage %in% 2500:3000 ~ "big4", sq_footage %in% 3000:3500 ~ "big5", sq_footage %in% 3500:4000 ~ "big6", sq_footage %in% 4000:4500 ~ "big7", sq_footage %in% 4500:5000 ~ "big8", sq_footage %in% 5000:5500 ~ "big9", sq_footage %in% 5500:6000 ~ "big10", sq_footage %in% 6000:6500 ~ "big11", sq_footage %in% 6500:7000 ~ "big12", sq_footage %in% 7000:7500 ~ "big13", sq_footage %in% 7500:8000 ~ "big14", sq_footage %in% 8000:8500 ~ "big15", sq_footage %in% 8500:9000 ~ "big16", sq_footage %in% 9000:9500 ~ "big17", sq_footage %in% 9500:10000 ~ "big18", sq_footage %in% 10000:10500 ~ "big19", sq_footage %in% 10500:11000 ~ "big20", sq_footage %in% 11000:11500 ~ "big21", sq_footage %in% 11500:12000 ~ "big22", sq_footage %in% 12000:12500 ~ "big23", sq_footage %in% 12500:13000 ~ "big24", sq_footage %in% 13000:13500 ~ "big25", sq_footage %in% 13500:14000 ~ "big26", sq_footage %in% 14000:14500 ~ "big27", sq_footage %in% 14500:15000 ~ "big28", sq_footage %in% 15000:15500 ~ "big29", sq_footage %in% 15500:16000 ~ "big30", sq_footage %in% 16000:16500 ~ "big31", sq_footage %in% 16500:17000 ~ "big32", sq_footage %in% 17000:17500 ~ "big33", sq_footage %in% 17500:18000 ~ "big34", sq_footage %in% 18000:18500 ~ "big35", sq_footage %in% 18500:19000 ~ "big36", sq_footage %in% 19000:19500 ~ "big37", sq_footage %in% 19500:20000 ~ "big38", sq_footage %in% 20000:20500 ~ "big39", sq_footage %in% 20500:21000 ~ "big40", sq_footage %in% 21000:21500 ~ "big41", sq_footage %in% 21500:22000 ~ "big42", sq_footage %in% 22000:22500 ~ "big43", sq_footage %in% 22500:23000 ~ "big44", sq_footage %in% 23000:23500 ~ "big45", sq_footage %in% 23500:24000 ~ "big46", sq_footage %in% 24000:24500 ~ "big47", sq_footage %in% 24500:25000 ~ "big48", sq_footage %in% 25000:25500 ~ "big49", sq_footage %in% 25500:26000 ~ "big50", sq_footage %in% 26000:26500 ~ "big51", sq_footage %in% 26500:27000 ~ "big52", sq_footage %in% 27000:27500 ~ "big53", sq_footage %in% 27500:28000 ~ "big54", sq_footage %in% 28000:28500 ~ "big55", sq_footage %in% 28500:29000 ~ "big56", sq_footage %in% 29000:29500 ~ "big57", sq_footage %in% 29500:30000 ~ "big58", sq_footage %in% 30000:30500 ~ "big59", sq_footage %in% 30500:31000 ~ "big60", sq_footage %in% 31000:31500 ~ "big61", sq_footage %in% 31500:32000 ~ "big62", sq_footage %in% 32000:32500 ~ "big63", sq_footage %in% 32500:33000 ~ "big64", sq_footage %in% 33000:33500 ~ "big65", sq_footage %in% 33500:34000 ~ "big66", sq_footage %in% 34000:34500 ~ "big67", sq_footage %in% 34500:35000 ~ "big68", sq_footage %in% 35000:35500 ~ "big69", sq_footage %in% 35500:36000 ~ "big70", sq_footage %in% 36000:36500 ~ "big71", sq_footage %in% 36500:37000 ~ "big72", sq_footage %in% 37000:37500 ~ "big73", sq_footage %in% 37500:38000 ~ "big74", sq_footage %in% 38000:38500 ~ "big75", sq_footage %in% 38500:39000 ~ "big76", sq_footage %in% 39000:39500 ~ "big77", sq_footage %in% 39500:40000 ~ "big78", sq_footage %in% 40000:40500 ~ "big79", sq_footage %in% 40500:41000 ~ "big80", sq_footage %in% 41000:41500 ~ "big81", sq_footage %in% 41500:42000 ~ "big82", sq_footage %in% 42000:42500 ~ "big83", sq_footage %in% 42500:43000 ~ "big84", sq_footage %in% 43000:43500 ~ "big85", sq_footage %in% 43500:44000 ~ "big86", sq_footage %in% 44000:44500 ~ "big87", sq_footage %in% 44500:45000 ~ "big88", sq_footage %in% 45000:45500 ~ "big89", sq_footage %in% 45500:46000 ~ "big90", sq_footage %in% 46000:46500 ~ "big91", sq_footage %in% 46500:47000 ~ "big92", sq_footage %in% 47000:47500 ~ "big93", sq_footage %in% 47500:48000 ~ "big94", sq_footage %in% 48000:48500 ~ "big95", sq_footage %in% 48500:49000 ~ "big96", sq_footage %in% 49000:49500 ~ "big97", sq_footage %in% 49500:50000 ~ "big98", sq_footage %in% 50000:50500 ~ "big99", sq_footage %in% 50500:51000 ~ "big100", sq_footage %in% 51000:51500 ~ "big101", sq_footage %in% 51500:52000 ~ "big102", sq_footage %in% 52000:52500 ~ "big103", sq_footage %in% 52500:53000 ~ "big104", sq_footage %in% 53000:53500 ~ "big105", sq_footage %in% 53500:54000 ~ "big106", sq_footage %in% 54000:54500 ~ "big107", sq_footage %in% 54500:55000 ~ "big108", sq_footage %in% 55000:55500 ~ "big109", sq_footage %in% 55500:56000 ~ "big110", sq_footage %in% 56000:56500 ~ "big111", sq_footage %in% 56500:57000 ~ "big112", sq_footage %in% 57000:57500 ~ "big113", sq_footage %in% 57500:58000 ~ "big114", sq_footage %in% 58000:58500 ~ "big115", sq_footage %in% 58500:59000 ~ "big116", sq_footage %in% 59000:59500 ~ "big117", sq_footage %in% 59500:60000 ~ "big118", sq_footage %in% 60000:60500 ~ "big119", sq_footage %in% 60500:61000 ~ "big120", sq_footage %in% 61000:61500 ~ "big121", sq_footage %in% 61500:62000 ~ "big122", sq_footage %in% 62000:62500 ~ "big123", sq_footage %in% 62500:63000 ~ "big124", sq_footage %in% 63000:63500 ~ "big125", sq_footage %in% 63500:64000 ~ "big126", sq_footage %in% 64000:64500 ~ "big127", sq_footage %in% 64500:65000 ~ "big128", sq_footage %in% 65000:65500 ~ "big129", sq_footage %in% 65500:66000 ~ "big130", sq_footage %in% 66000:66500 ~ "big131", sq_footage %in% 66500:67000 ~ "big132", sq_footage %in% 67000:67500 ~ "big133", sq_footage %in% 67500:68000 ~ "big134", sq_footage %in% 68000:68500 ~ "big135", sq_footage %in% 68500:69000 ~ "big136", sq_footage %in% 69000:69500 ~ "big137", sq_footage %in% 69500:70000 ~ "big138", sq_footage %in% 70000:70500 ~ "big139", sq_footage %in% 70500:71000 ~ "big140", sq_footage %in% 71000:71500 ~ "big141", sq_footage %in% 71500:72000 ~ "big142", sq_footage %in% 72000:72500 ~ "big143", sq_footage %in% 72500:73000 ~ "big144", sq_footage %in% 73000:73500 ~ "big145", sq_footage %in% 73500:74000 ~ "big146", sq_footage %in% 74000:74500 ~ "big147", sq_footage %in% 74500:75000 ~ "big148", sq_footage %in% 75000:75500 ~ "big149", sq_footage %in% 75500:76000 ~ "big150", sq_footage %in% 76000:76500 ~ "big151", sq_footage %in% 76500:77000 ~ "big152", sq_footage %in% 77000:77500 ~ "big153", sq_footage %in% 77500:78000 ~ "big154", sq_footage %in% 78000:78500 ~ "big155", sq_footage %in% 78500:79000 ~ "big156", sq_footage %in% 79000:79500 ~ "big157", sq_footage %in% 79500:80000 ~ "big158", sq_footage %in% 80000:80500 ~ "big159", sq_footage %in% 80500:81000 ~ "big160", sq_footage %in% 81000:81500 ~ "big161", sq_footage %in% 81500:82000 ~ "big162", sq_footage %in% 820
```

```
condo_cases = manipulated_data %>%
  filter(coop_condo == "condo") %>%
  mutate(maintenance_cost = if_else(is.na(maintenance_cost), 0, maintenance_cost)) %>%
  mutate(pct_tax_deductibl = if_else(is.na(pct_tax_deductibl), 0, pct_tax_deductibl/100)) %>%
  mutate(total_taxes = parse_number(total_taxes)) %>%
  mutate(total_cost = maintenance_cost * (1-pct_tax_deductibl) + total_taxes)
```

```
coop_cases = manipulated_data %>%
  filter(coop_condo == "co-op") %>%
  mutate(pct_tax_deductibl = if_else(is.na(pct_tax_deductibl), mean(pct_tax_deductibl, na.rm = TRUE)/100, pct_tax_deductibl)) %>%
  mutate(total_taxes = if_else(is.na(total_taxes), 0, parse_number(total_taxes))) %>%
  mutate(total_cost = maintenance_cost * (1-pct_tax_deductibl)+ total_taxes)

data_with_cost = bind_rows(condo_cases, coop_cases) %>%
  select(-maintenance_cost, - pct_tax_deductibl, -total_taxes, -cats_allowed, -dogs_allowed)

#table((data_with_cost$total_cost), exclude = NaN)
```

Train and Test split First we need to filter out the the missing sale prices. Then we will find the indices of the split for train and test.

```
filtered_data = data_with_cost %>%
  filter(!is.na(sale_price)) %>%
  mutate_if(sapply(data_with_cost, is.character), as.factor)

missing_sale_price = data_with_cost %>%
  filter(is.na(sale_price)) %>%
  mutate_if(sapply(data_with_cost, is.character), as.factor)

data_summary = lapply(filtered_data, summary)
print (data_summary)
```

```
## $sale_price
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  55000  171500  259500  314957  428875  999999
##
## $community_district_num
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##       3.0   25.0   26.0   26.3   28.0   30.0     1
##
## $coop_condo
## co-op  condo
##   399   129
##
## $dining_room_type
##   combo  formal   other Unknown
##    241    118    49    120
##
## $fuel_type
## electric    gas    none    oil    other   NA's
##      11    301     3    180     9    24
##
## $garage_exists
## No Yes
## 434  94
##
## $kitchen_type
##      Combo    Eat In efficiency   NA's
##       81    209    231      7
##
## $num_bedrooms
```

```

##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      0.000   1.000   1.000   1.538   2.000   3.000
##
## $num_floors_in_building
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.      NA's
##      1.000   2.000   6.000   7.081   7.000  34.000      108
##
## $num_full_bathrooms
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.000   1.000   1.000   1.205   1.000   3.000
##
## $num_half_bathrooms
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      0.00000 0.00000 0.00000 0.05871 0.00000 2.00000
##
## $num_total_rooms
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.000   3.000   4.000   4.025   5.000   8.000
##
## $sq_footage
##      large      medium      small super large      Unknown
##      64      120      21      8      315
##
## $walk_score
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      15.0   76.0   85.0   83.1   94.0   99.0
##
## $pets_allowed
##      no yes
##      283 245
##
## $region
##      Central Queens      Jamaica      North Queens      Northeast Queens
##      34      34      113      72
##      Northwest Queens      Southeast Queens      Southwest Queens      West Central Queens
##      20      34      59      93
##      West Queens
##      69
##
## $decade_built
##      1915 - 1939      1940's      1950's      1960's      1970's      1980's
##      38      37      209      115      25      37
##      1990's      2000's      2010's      NA's
##      9      34      19      5
##
## $total_cost
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.      NA's
##      11.0   341.9   425.3   928.0   670.3   9640.0      24

```

```
K = 4
```

```

test_indices = sample(1 : nrow(filtered_data), round(nrow(filtered_data) / K))
train_indices = setdiff(1 : nrow(filtered_data), test_indices)
test_data_miss = filtered_data[test_indices, ]
train_data_miss = filtered_data[train_indices, ]

```

```

miss_data_train_combined = bind_rows(train_data_miss, missing_sale_price) %>%
  mutate(sale_price_dummy = if_else(is.na(sale_price), 0, 1))
miss_data_train_combined = miss_data_train_combined %>%
  mutate_if(sapply(miss_data_train_combined, is.character), as.factor)

ximpMF = missForest(miss_data_train_combined)

## missForest iteration 1 in progress...done!
## missForest iteration 2 in progress...done!
## missForest iteration 3 in progress...done!
## missForest iteration 4 in progress...done!

train_data = ximpMF$ximp %>%
  filter(sale_price_dummy == 1) %>%
  select(-sale_price_dummy)

y_train = train_data$sale_price
X_train = train_data[, -1]

```

Modeling We will build 3 models. 1. Regression Tree 2. Linear Model 3. Random Forest

Regression Tree

```

tree_model = YARF(data.frame(x = X_train), y_train, num_trees = 1)

## YARF initializing with a fixed 1 trees...
## YARF factors created...
## YARF after data preprocessed... 49 total features...
## Beginning YARF regression model construction...done.
## Calculating OOB error...done.

tree_model

## YARF v1.1 for regression
## Missing data feature ON.
## 1 trees, training data n = 396 and p = 49
## Model construction completed within 0.01 minutes.
## OOB results on 35.35% of the observations (256 missing):
##   R^2: 0.8118
##   RMSE: 129691.2
##   MAE: 94116.69
##   L2: 2.354772e+12
##   L1: 13176336

```

Linear Model

```

linear_mod = lm(sale_price ~ ., train_data)
sd(y_train - linear_mod$fitted.values)

## [1] 64200.4

summary(linear_mod)

##
## Call:

```

```

## lm(formula = sale_price ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -217940  -39781   -4652   40177  296404
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -93951.766   59856.787   -1.570  0.117397
## community_district_num      4590.346    1254.245    3.660  0.000291 ***
## coop_condocondo    132270.686   20770.339    6.368  5.92e-10 ***
## dining_room_typeformal    36698.436    9805.680    3.743  0.000212 ***
## dining_room_typeother    27301.555   13025.320    2.096  0.036786 *
## dining_room_typeUnknown    15403.947    9331.861    1.651  0.099688 .
## fuel_typegas    15266.144   24323.914    0.628  0.530657
## fuel_typenone    44901.823   49158.175    0.913  0.361644
## fuel_typeoil    19037.413   24921.582    0.764  0.445439
## fuel_typeother    39106.966   38250.574    1.022  0.307293
## garage_existsYes    15019.679   10491.627    1.432  0.153142
## kitchen_typeEat In   -14703.640   11298.571   -1.301  0.193975
## kitchen_typeefficiency -26835.474   11209.957   -2.394  0.017189 *
## num_bedrooms    50060.999    8580.480    5.834  1.22e-08 ***
## num_floors_in_building    6511.477    771.884    8.436  8.42e-16 ***
## num_full_bathrooms    88886.595   13348.672    6.659  1.05e-10 ***
## num_half_bathrooms    25437.247   17586.550    1.446  0.148946
## num_total_rooms    9338.301    5983.584    1.561  0.119497
## sq_footagemedium   -41724.348   14157.433   -2.947  0.003419 **
## sq_footagesmall   -68277.156   22607.218   -3.020  0.002710 **
## sq_footagesuper large  167472.133   39710.898    4.217  3.14e-05 ***
## sq_footageUnknown   -31772.455   12537.337   -2.534  0.011699 *
## walk_score        70.127    352.561    0.199  0.842450
## pets_allowedyes    14514.972    7679.099    1.890  0.059547 .
## regionJamaica   -69712.253   20354.598   -3.425  0.000687 ***
## regionNorth Queens    48305.962   16671.068    2.898  0.003994 **
## regionNortheast Queens    32843.172   18161.936    1.808  0.071398 .
## regionNorthwest Queens   111093.042   24279.840    4.576  6.58e-06 ***
## regionSoutheast Queens   -4304.133   22967.449   -0.187  0.851453
## regionSouthwest Queens  -84046.297   18572.734   -4.525  8.24e-06 ***
## regionWest Central Queens  40252.873   17506.812    2.299  0.022070 *
## regionWest Queens    28176.842   18206.955    1.548  0.122613
## decade_built1940's   -30455.101   18638.729   -1.634  0.103152
## decade_built1950's   -55597.263   14813.967   -3.753  0.000204 ***
## decade_built1960's   -48697.445   16296.981   -2.988  0.003002 **
## decade_built1970's   -17570.583   25534.771   -0.688  0.491837
## decade_built1980's   -57266.766   25484.032   -2.247  0.025243 *
## decade_built1990's   -87600.368   33324.621   -2.629  0.008944 **
## decade_built2000's   -9939.999   28159.175   -0.353  0.724303
## decade_built2010's   108502.953   29790.419    3.642  0.000311 ***
## total_cost          9.959     5.053     1.971  0.049530 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67720 on 355 degrees of freedom
## Multiple R-squared:  0.8699, Adjusted R-squared:  0.8552

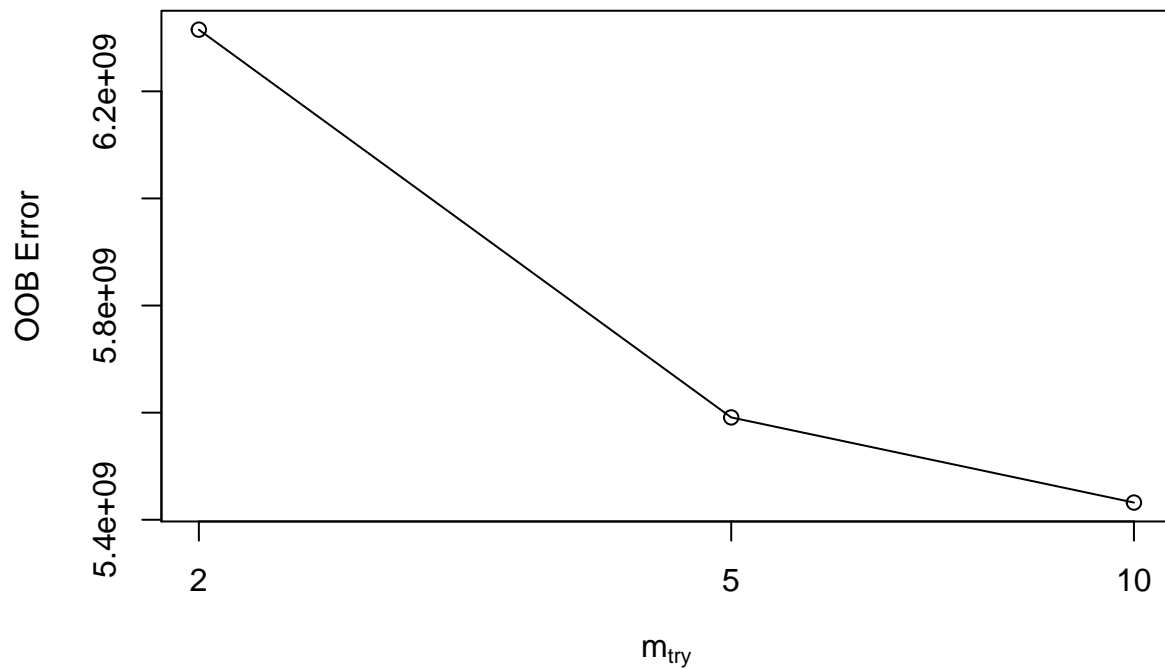
```

```
## F-statistic: 59.33 on 40 and 355 DF,  p-value: < 2.2e-16
```

Random Forest

```
mtry_mlr= tuneRF(X_train,
                  y_train,
                  stepFactor=0.5,
                  plot=TRUE,
                  ntreeTry=300,
                  trace=TRUE,
                  improve = 0.05)
```

```
## mtry = 5   OOB error = 5591104297
## Searching left ...
## mtry = 10   OOB error = 5432153980
## 0.02842915 0.05
## Searching right ...
## mtry = 2     OOB error = 6315139277
## -0.1294977 0.05
```



```
print(mtry_mlr)
```

```
##      mtry  OOBError
## 2         2 6315139277
## 5         5 5591104297
## 10        10 5432153980
```

```

mod_rf = YARF(X_train, y_train, mtry = 10)

## YARF initializing with a fixed 500 trees...
## YARF factors created...
## YARF after data preprocessed... 49 total features...
## Beginning YARF regression model construction...done.
## Calculating OOB error...done.

mod_rf

## YARF v1.1 for regression
## Missing data feature ON.
## 500 trees, training data n = 396 and p = 49
## Model construction completed within 0.02 minutes.
## OOB results on all observations:
##   R^2: 0.79065
##   RMSE: 81329.65
##   MAE: 56159.71
##   L2: 2.619347e+12
##   L1: 22239246

##Performance of Random Forest

y_test = test_data_miss$sale_price
x_test_miss = test_data_miss %>%
  mutate(sale_price = -1)
miss_data_test_combined = bind_rows(train_data_miss, missing_sale_price, x_test_miss) %>%
  mutate(sale_price_dummy = if_else(sale_price == -1, 1, 0)) %>%
  mutate(sale_price_dummy = if_else(is.na(sale_price_dummy), 0, sale_price_dummy)) %>%
  mutate(sale_price = na_if(sale_price, -1))
miss_data_test_combined = miss_data_test_combined %>%
  mutate_if(sapply(miss_data_test_combined, is.character), as.factor)

test_imputed = missForest(miss_data_test_combined)

## missForest iteration 1 in progress...done!
## missForest iteration 2 in progress...done!
## missForest iteration 3 in progress...done!
## missForest iteration 4 in progress...done!
## missForest iteration 5 in progress...done!

X_test = test_imputed$ximp %>%
  filter(sale_price_dummy == 1) %>%
  select(-sale_price, -sale_price_dummy)

y_hat = predict(mod_rf, X_test)

y_bar = mean(y_test)
SSR = sum((y_hat-y_bar)^2)
SST = sum((y_test-y_bar)^2)
rsq = (SSR/SST)
rsq

## [1] 0.5155823

sd(y_test - y_hat)

```


[1] 93058.01