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 pricel. 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_pric

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 footage was factorized into small medium large super large and unknown. 4. Garage exists takes all the na's and makes them no. All others are yes. 5. Missing half bathrooms are set to zero. 6. In the dining_room_type, we assume 'dining area' is considered 'formal'. Set 'NA' as "unknown". 7. In the 'kitchen_type' feature, there was a observation from 'approx_year_built'. Added that data back and cleaned the rest of 'kitchen_type' 8. Converted 'maintenance_cost' and 'parking_charges' into numerics by removing the dollar sign and the comma. 7. Get zip_code from 'full_address_or_zip_code' and sort it into regions. 8. Made 'decade_built' from approx_year_built 9. Finally put 'sale_price' in front and remove 'zip_code', 'full_address_or_zip_code' and 'approx_year_built'.

```
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% 601:998 ~

  mutate(garage_exists = if_else(is.na(garage_exists), "No", "Yes")) %>%

  mutate(num_half_bathrooms = replace_na(num_half_bathrooms, 0)) %>%

  mutate(dining_room_type = if_else(dining_room_type == "dining area", "formal", dining_room_type)) %>%
  mutate(dining_room_type = ifelse(is.na(dining_room_type), "Unknown", dining_room_type)) %>%

  mutate(approx_year_built = if_else(kitchen_type == "1955", as.integer(1955), approx_year_built, missi
  mutate(kitchen_type = as.factor( case_when(kitchen_type == "combo" ~ "Combo", kitchen_type == "Combo"

  mutate(maintenance_cost =  as.numeric(gsub("[\\$,\\,]", "", maintenance_cost))) %>%
  mutate(sale_price =  as.numeric(gsub("[\\$,\\,]", "", sale_price))) %>%

  mutate(zip_code = str_sub(full_address_or_zip_code, start= -5), zip_code = if_else(zip_code == "Share
  mutate(region = as.factor( case_when(zip_code %in% Northeast_Queens_zip  ~ "Northeast Queens", zip_co

  mutate(decade_built = as.factor(case_when(approx_year_built %in% 1915:1939 ~ "1915 - 1939", approx_ye



  select(sale_price, everything(), -full_address_or_zip_code, -zip_code, -approx_year_built)
```

```
## Warning in replace_with(out, !condition, false, fmt_args(~false), glue("length
## of {fmt_args(~condition)}")): NAs introduced by coercion
```

```
 # table(manipulated_data$dining_room_type, exclude = NaN)
```

##co-op Condo cost There is a big difference between co-op and condo. we will seperate them into 2 data frames and find the total cost. we will then combine the cases.

```
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)/10
  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 indicies 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:
```

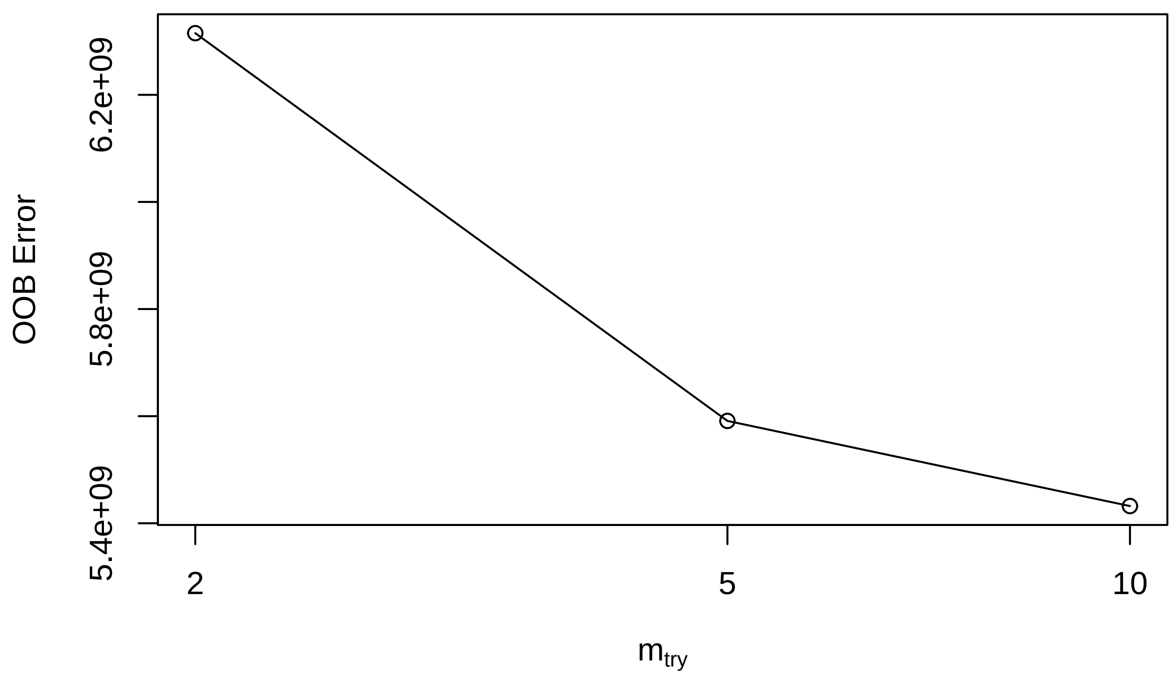
```
## lm(formula = sale_price ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -217940  -39781   -4652   40177  296404
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -93951.766  59856.787  -1.570 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

```r
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
```




```r
print(mtry_mlr)
```

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

```r
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.
```

```r
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

```r
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!
```

```r
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
```

```r
sd(y_test - y_hat)
```

```
## [1] 93058.01
```