Predicting Housing Costs in Queens, NY

Final project for Math 342w Queens College

AUTHOR
Benjamin Minkin

PUBLISHED
May 26, 2024

Data Wrangling

Import the csv file

```
library("randomForest")
library("missForest")
library("YARF")
library("fastDummies")
library("stargazer")
library("dplyr")
library("stringr")
set.seed = (123)

#import the data
housing_data = read.csv("C:\\Users\\benmi\\OneDrive\\Desktop\\Math 342w\\housing_data_2016_2017.cs
```

Separate out the desired features for prediction

```
#vector of columns that are relevant for the model
cols_to_keep = c(29, 31:33,36, 38:39, 41, 43, 45:48, 50:51, 53)

#number of columns used in the model
num_cols_kept = length(cols_to_keep)

#subset of housing data that only has relevant columns
relevant_data = housing_data %>% select(all_of(cols_to_keep))
```

Dummify the garage_exists to be 1 for “yes” and 0 for NA

```
#Dummify garage 1 for yes, Yes, eys, UG, and Underground, 0 for NA
relevant_data$garage_exists = ifelse(relevant_data$garage_exists == "Yes" | relevant_data$garage_e

#set NA values to 0
relevant_data$garage_exists = ifelse(is.na(relevant_data$garage_exists), 0, 1)
```

Set NA values of num_half_bathrooms to 0

```
#set half bathrooms, NA to 0
relevant_data$num_half_bathrooms = ifelse(is.na(relevant_data$num_half_bathrooms), 0, relevant_dat
```

Cast the common_charges string to an integer, setting NA to 0

```
#turn common charge's NA values to 0
relevant_data$common_charges = ifelse(is.na(relevant_data$common_charges), 0, relevant_data$commo

#remove dollar sign and comma
relevant_data$common_charges = str_replace(relevant_data$common_charges, "\\$", "")
relevant_data$common_charges = str_replace(relevant_data$common_charges, ",", "")

#convert to numeric
relevant_data$common_charges = as.numeric(relevant_data$common_charges)
```

Cast the maintenance_cost string to an int, setting NA to 0

```
#set NA values to 0
relevant_data$maintenance_cost = ifelse(is.na(relevant_data$maintenance_cost), 0, relevant_data$ma

#remove dollar sign and comma
relevant_data$maintenance_cost = str_replace(relevant_data$maintenance_cost, "\\$", "")
relevant_data$maintenance_cost = str_replace(relevant_data$maintenance_cost, ",", "")

#set to numeric
relevant_data$maintenance_cost = as.numeric(relevant_data$maintenance_cost)
```

Create a new column that is the sum of maintenance_cost and common_charges

```
#create new col that is total maintenance and common_charges
relevant_data$total_com_maint = relevant_data$maintenance_cost + relevant_data$common_charges
```

Change exceptions of full_address_or_zip_code to their zip code

```
#manually fix zip code errors
relevant_data[relevant_data$full_address_or_zip_code == "78-07 Springfield Blvd,  Bayside NY, 113

relevant_data[relevant_data$full_address_or_zip_code == "32-42 89th St,  E. Elmhurst NY, 1136",6]

relevant_data[relevant_data$full_address_or_zip_code == "35-25 77 St,  Jackson Heights NY, 1137",

relevant_data[relevant_data$full_address_or_zip_code == "34-30 78th St,  Jackson Heights NY, 1137
```

```
relevant_data[relevant_data$full_address_or_zip_code == "61-20 Grand Central Pky,  Forest Hills NY

relevant_data[relevant_data$full_address_or_zip_code == "42-42 Colden Street,  Flushing NY, 11355

relevant_data[relevant_data$full_address_or_zip_code == "80-35 Springfield Blvd,  Queens Village N

relevant_data[relevant_data$full_address_or_zip_code == "138-35 Elder Ave,  Flushing NY, 1135",6]
```

Manually add the missing approx_year_built data

```
#add missing "year built" data
relevant_data[relevant_data$full_address_or_zip_code == "34-20 Parsons Blvd, Flushing NY, 11354",

relevant_data[relevant_data$full_address_or_zip_code == "34-41 78th Street, Jackson Heights, NY 11

relevant_data[relevant_data$full_address_or_zip_code == "92-31 57th Ave,  Elmhurst NY, 11373", ]$a

relevant_data[relevant_data$full_address_or_zip_code == "102-32 65th Ave,  Forest Hills NY, 11375

relevant_data[relevant_data$full_address_or_zip_code == "170-06 Crocheron Ave,  Flushing NY, 1135

relevant_data[relevant_data$full_address_or_zip_code == "74-63 220th Street,  Bayside NY, 11364",
```

Dummify co-op_condo to be 1 for co-op and 0 for condo

```
relevant_data$coop_condo = ifelse(relevant_data$coop_condo == "co-op", 1, 0)
```

Dummify dogs_allowed to be 1 for “yes” and 0 for “no”

```
relevant_data$dogs_allowed = ifelse(relevant_data$dogs_allowed == "yes" | relevant_data$dogs_allow
```

Cast sale_price string to be an integer

```
#remove $ and ,
relevant_data$sale_price = str_replace(relevant_data$sale_price, "\\$", "")
relevant_data$sale_price = str_replace(relevant_data$sale_price, ",", "")

#convert to numeric
relevant_data$sale_price = as.numeric(relevant_data$sale_price)
```

Dummify approx_year_built to 0 if built before 1978, 1 if built after 1978. (When lead paint was outlawed federally)

```
relevant_data$approx_year_built = ifelse(relevant_data$approx_year_built<1978, 0, 1)
```

Extract zip codes from full_address_or_zip_code string

```
#extract zip codes from address string
relevant_data$full_address_or_zip_code = str_sub(relevant_data$full_address_or_zip_code, start = 

#handle exception
relevant_data$full_address_or_zip_code[relevant_data$full_address_or_zip_code == "Share"] = "1135

relevant_data$full_address_or_zip_code[relevant_data$full_address_or_zip_code == "1355."] = "1135

relevant_data$full_address_or_zip_code[relevant_data$full_address_or_zip_code == "1367."] = "1136

relevant_data$full_address_or_zip_code[relevant_data$full_address_or_zip_code == "17-30"] = "1136

#convert to numeric
relevant_data$full_address_or_zip_code = as.numeric(relevant_data$full_address_or_zip_code)
```

Categorize the zip codes into regions

```
Northeast = c(11361, 11362, 11363, 11364)
North = c(11354, 11355, 11356, 11357, 11358, 11359, 11360)
Central = c(11365, 11366, 11367)
Jamaica = c(11412, 11423, 11432, 11433, 11434, 11435, 11436)
Northwest = c(11101, 11102, 11103, 11104, 11105, 11106)
West_Central = c(11374, 11375, 11379, 11385)
Southeast = c(11004, 11005, 11411, 11413, 11422, 11426, 11427, 11428, 11429)
Southwest = c(11414, 11415, 11416, 11417, 11418, 11419, 11420, 11421)
West = c(11368, 11369, 11370, 11372, 11373, 11377, 11378)

relevant_data$full_address_or_zip_code = case_when(
  relevant_data$full_address_or_zip_code %in% Northeast ~ 1,
  relevant_data$full_address_or_zip_code %in% North ~ 2,
  relevant_data$full_address_or_zip_code %in% Central ~ 3,
  relevant_data$full_address_or_zip_code %in% Jamaica ~ 4,
  relevant_data$full_address_or_zip_code %in% West_Central ~ 5,
  relevant_data$full_address_or_zip_code %in% Southeast ~ 6,
  relevant_data$full_address_or_zip_code %in% Southwest ~ 7,
  relevant_data$full_address_or_zip_code %in% West ~ 8,
  relevant_data$full_address_or_zip_code %in% Northwest ~ 9)
```

Dummify the zip code categorical variable

```
relevant_data = dummy_cols(relevant_data, select_columns = c("full_address_or_zip_code"), remove_
```

Filter out columns that will not be used in the final model

```
#further subset to rows with sale prices
non_NA_sale = relevant_data[!is.na(relevant_data$sale_price),]
features_vec = c(1,4:6,8:11,13,14,15:24)
select_data = non_NA_sale %>% select(all_of(features_vec))
```

Randomly split the data into a training and testing split at an approx. 4:1 ratio respectively.

```
#randomly pick indices
split_index = sample(nrow(select_data), size = nrow(select_data), replace = FALSE)

#create subset of 80%
splitting_point = split_index[1:round(0.8*nrow(select_data), 0)]

#create training and testing sets
train_data = select_data[splitting_point, ]
test_data = select_data[-splitting_point, ]
```

Use the entire relevant data set (excluding sale price) to impute square ft. values

```
#The remaining data
NA_sale = relevant_data[is.na(relevant_data$sale_price),]

#remove unused variables
features_vec = c(1,4:6,8:11,13,14,15:24)
impute_NA_sale = NA_sale %>% select(all_of(features_vec))

impute_train = train_data
impute_test = test_data

#bind all data
impute_NA_sale = rbind(impute_NA_sale, impute_train)
impute_NA_sale = rbind(impute_NA_sale, impute_test)

#remove sale price
impute_NA_sale = impute_NA_sale %>% select(-9)
```

Dummify NA values of sq_footage

```
#dummify na values as NA_sq_ft where 1 is NA, 0 not
train_data$NA_sq_ft = ifelse(is.na(train_data$sq_footage),1,0)
test_data$NA_sq_ft = ifelse(is.na(test_data$sq_footage),1,0)
```

Impute square foot values

```
#impute sq_footage
ximpmf = missForest(impute_NA_sale)

#set imputed values to their respectuve indices
```

```
train_data$sq_footage = ximpmf$ximp$sq_footage[1703:2124]
test_data$sq_footage = ximpmf$ximp$sq_footage[2125:2230]
```

The Linear Model

```
ols_model = lm(sale_price ~., data = train_data)
stargazer(ols_model, type = "text")
```

```
=====================================================
                            Dependent variable:
                        ---------------------------
                                  sale_price
-----------------------------------------------------
approx_year_built                  40,535.900*
                                  (21,109.540)

coop_condo                       -197,855.900***
                                  (19,386.350)

dogs_allowed                       19,830.430**
                                   (9,516.272)

garage_exists                      19,316.880*
                                  (10,808.190)

num_bedrooms                       26,060.180**
                                  (10,450.480)

num_full_bathrooms                 64,565.280***
                                  (15,010.970)

num_half_bathrooms                  22,338.580
                                  (18,522.500)

num_total_rooms                     -3,531.935
                                   (6,526.337)

sq_footage                          161.740***
                                    (30.123)

walk_score                         1,641.243***
                                    (405.572)

total_com_maint                     119.281***
                                    (15.274)

full_address_or_zip_code_2           7,889.785
```

```
                                  (15,608.760)

full_address_or_zip_code_3        -41,436.480**
                                  (20,352.920)

full_address_or_zip_code_4       -97,702.010***
                                  (20,287.850)

full_address_or_zip_code_5         24,657.970
                                  (17,367.940)

full_address_or_zip_code_6         16,024.280
                                  (19,540.450)

full_address_or_zip_code_7       -89,101.120***
                                  (16,800.020)

full_address_or_zip_code_8         10,834.530
                                  (18,084.750)

full_address_or_zip_code_9       117,437.800***
                                  (27,589.220)

NA_sq_ft                           -5,704.729
                                   (8,837.210)

Constant                           -11,837.340
                                  (44,531.500)

-------------------------------------------------------
Observations                             422
R2                                     0.815
Adjusted R2                            0.806
Residual Std. Error           81,917.300 (df = 401)
F Statistic                 88.617*** (df = 20; 401)
=======================================================
Note:                      *p<0.1; **p<0.05; ***p<0.01
```

The Tree Model

```
tree_model = YARF(X = train_data[,-9], y = as.vector(train_data $sale_price), num_trees = 1)
```

```
YARF initializing with a fixed 1 trees...
YARF after data preprocessed... 20 total features...
Beginning YARF regression model construction...done.
Calculating OOB error...done.
```

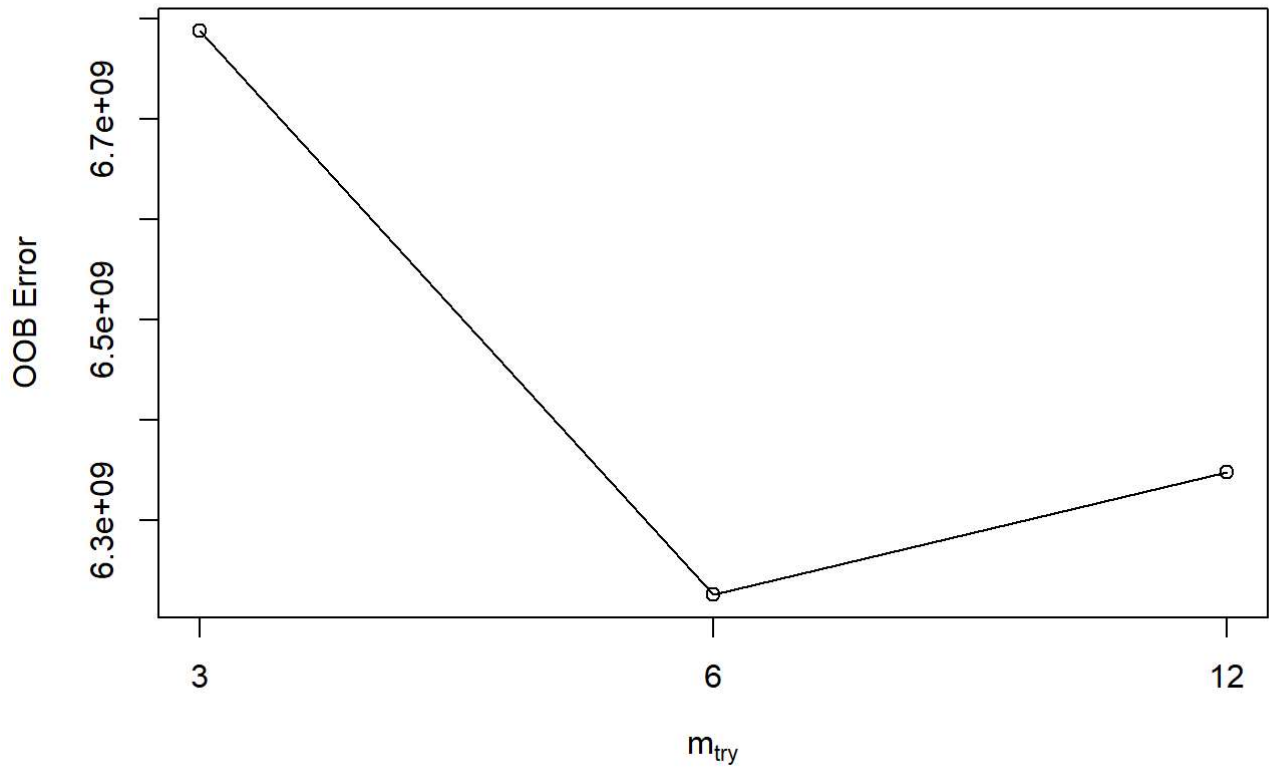
```
tree_model
```

```
YARF v1.1 for regression
Missing data feature ON.
1 trees, training data n = 422 and p = 20
Model construction completed within 0 minutes.
OOB results on 36.97% of the observations (266 missing):
  R^2: 0.79169
  RMSE: 139549.1
  MAE: 95252.85
  L2: 3.037939e+12
  L1: 14859445
```

Tune the forest to find the best mtry value

```
tuneRF(x = train_data[,-9],
       y = as.vector(train_data$sale_price),
       stepFactor = 0.5,
       ntreeTry=300,
       trace=TRUE,
       improve = 0.05,
       plot = TRUE)
```

```
mtry = 6  OOB error = 6226266513
Searching left ...
mtry = 12   OOB error = 6348084628
-0.01956519 0.05
Searching right ...
mtry = 3    OOB error = 6787239590
-0.09009783 0.05
```



	mtry	OOBError
3	3	6787239590
6	6	6226266513
12	12	6348084628

The Random Forest Model

```
yarf_model = YARF(X = train_data[,-9], y = as.vector(train_data $sale_price), mtry = 6)
```

```
YARF initializing with a fixed 500 trees...
YARF after data preprocessed... 20 total features...
Beginning YARF regression model construction...done.
Calculating OOB error...done.
```

```
yarf_model
```

```
YARF v1.1 for regression
Missing data feature ON.
500 trees, training data n = 422 and p = 20
Model construction completed within 0.01 minutes.
OOB results on all observations:
  R^2: 0.78443
```

```
  RMSE: 86313.44
  MAE: 58628.68
  L2: 3.143904e+12
  L1: 24741303
```

Print in-sample OLS values and out-of-sample values for all models

```
#Test OLS model in-sample
cat("OLS in-sample r_sq is ", summary(ols_model)$r.squared, " \n")
```

```
OLS in-sample r_sq is  0.8154907
```

```
cat("OLS in-sample RMSE is ", sqrt(mean(ols_model$residuals^2)))
```

```
OLS in-sample RMSE is  79853.07
```

```
#Test OLS model oos
ols_hat = predict(ols_model, test_data[,-9])
cat("\nOLS out-of-sample r_sq is ", cor(ols_hat, test_data$sale_price)^2, " \n")
```

```
OLS out-of-sample r_sq is  0.6243565
```

```
cat("OLS out-of-sample RMSE is ", sqrt(mean((test_data$sale_price - ols_hat)^2)), " \n")
```

```
OLS out-of-sample RMSE is  107963.6
```

```
#Test tree model oos
tree_hat = predict(tree_model, test_data[,-9])
cat("tree out-of-sample r_sq is ", cor(tree_hat, test_data$sale_price)^2, " \n")
```

```
tree out-of-sample r_sq is  0.4988322
```

```
cat("tree out-of-sample RMSE is ", sqrt(mean((test_data$sale_price - tree_hat)^2)), " \n")
```

```
tree out-of-sample RMSE is  126762.4
```

```
#Test yarf model oos
forest_hat = predict(yarf_model, test_data[,-9])
cat("Forest out-of-sample r_sq is ", cor(forest_hat, test_data$sale_price)^2, " \n")
```

```
Forest out-of-sample r_sq is  0.7775858
```

```
cat("Forest out-of-sample RMSE is ", sqrt(mean((test_data$sale_price - forest_hat)^2)), " \n")
```

```
Forest out-of-sample RMSE is  70550.11
```