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("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, 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
```

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 1

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_allo
```

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)
```

Further split the data set to include only the rows with a sale_price

```
#further subset to rows with sale prices
non_NA_sale = relevant_data[!is.na(relevant_data$sale_price),]
```

Extract zip codes from full_address_or_zip_code string

```
#extract zip codes from address string
non_NA_sale$full_address_or_zip_code = str_sub(non_NA_sale$full_address_or_zip_code, start = -5)

#handle exception
non_NA_sale$full_address_or_zip_code[non_NA_sale$full_address_or_zip_code == "Share"] = "11354"

#convert to numeric
non_NA_sale$full_address_or_zip_code = as.numeric(non_NA_sale$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)

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

Dummify the zip code categorical variable

```
non_NA_sale = dummy_cols(non_NA_sale, select_columns = c("full_address_or_zip_code"), remove_firs
```

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

```
features_vec = c(1,4:6,8:11,13,14,15:23)
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, ]
```

The Linear Model

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

```
=================================================
                            Dependent variable:
                        ---------------------------
                                  sale_price
-------------------------------------------------
approx_year_built                  14,588.150
                                  (20,848.920)

coop_condo                       -216,870.800***
                                  (19,463.920)

dogs_allowed                      27,867.080***
                                   (9,321.611)

garage_exists                      11,917.370
                                  (10,930.610)

num_bedrooms                      40,605.490***
                                   (9,224.459)

num_full_bathrooms                95,350.940***
                                  (13,037.910)

num_half_bathrooms                52,838.130***
                                  (17,640.520)

num_total_rooms                    6,928.698
                                   (6,038.764)

walk_score                        1,461.956***
                                    (401.418)

total_com_maint                    152.775***
                                    (13.914)

full_address_or_zip_code_2         17,003.930
                                  (14,640.910)

full_address_or_zip_code_3        -51,196.880**
                                  (20,009.990)

full_address_or_zip_code_4       -103,405.800***
                                  (19,623.740)

full_address_or_zip_code_5        36,836.750**
                                  (16,548.090)
```

```
full_address_or_zip_code_6          -7,455.474
                                   (19,360.160)

full_address_or_zip_code_7         -84,522.640***
                                   (15,937.530)

full_address_or_zip_code_8          10,405.100
                                   (17,371.560)

full_address_or_zip_code_9         133,637.000***
                                   (24,769.710)

Constant                            32,849.430
                                   (42,414.740)

-----------------------------------------------------
Observations                              422
R2                                      0.819
Adjusted R2                             0.811
Residual Std. Error           79,266.630 (df = 403)
F Statistic                 101.538*** (df = 18; 403)
=====================================================
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... 18 total features...
Beginning YARF regression model construction...done.
Calculating OOB error...done.
```

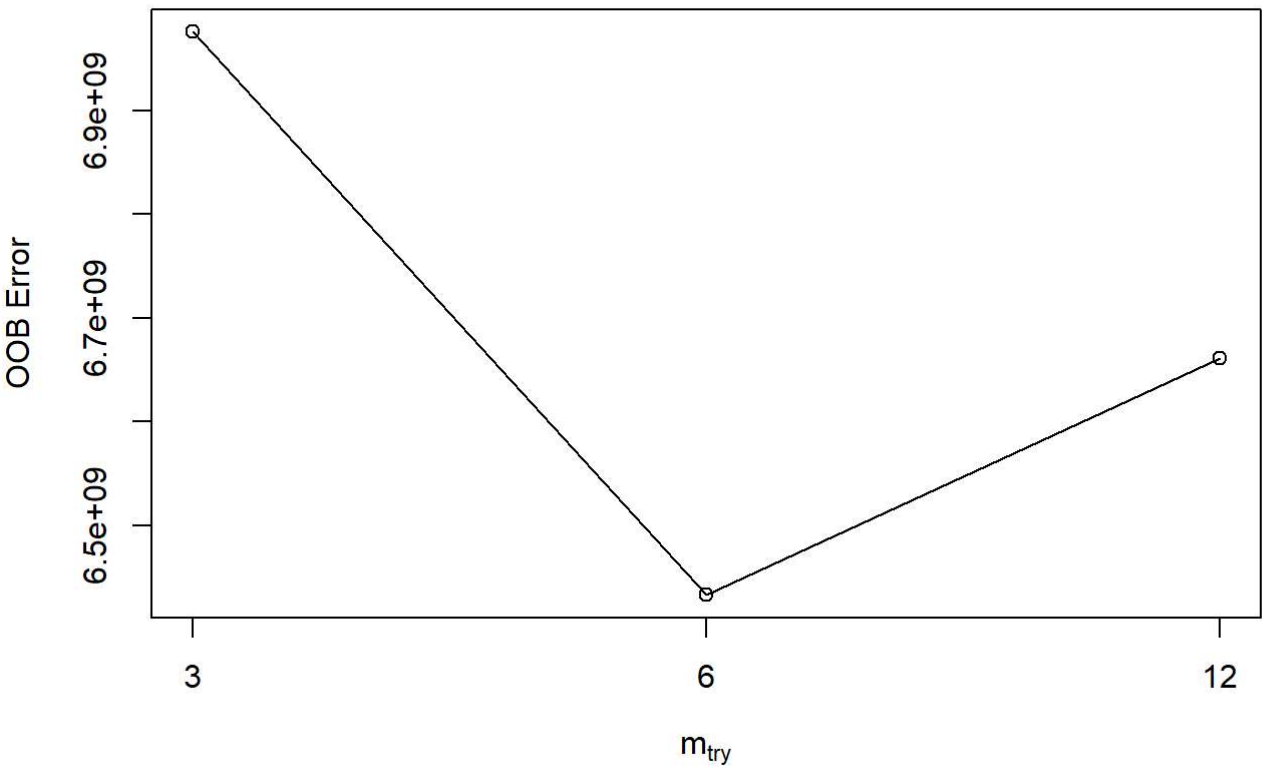
```
tree_model
```

```
YARF v1.1 for regression
Missing data feature ON.
1 trees, training data n = 422 and p = 18
Model construction completed within 0 minutes.
OOB results on 36.49% of the observations (268 missing):
  R^2: 0.8472
  RMSE: 117926.8
  MAE: 78549.36
  L2: 2.141635e+12
  L1: 12096601
```

Tune the random forest to find the best m_try parameter

```r
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 = 6433172916
Searching left ...
mtry = 12   OOB error = 6660903929
-0.03539949 0.05
Searching right ...
mtry = 3    OOB error = 6975872347
-0.08435953 0.05
```




```
   mtry   OOBError
3     3 6975872347
6     6 6433172916
12   12 6660903929
```

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... 18 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 = 18
Model construction completed within 0.01 minutes.
OOB results on all observations:
  R^2: 0.7774
  RMSE: 85983.9
  MAE: 58962.82
  L2: 3.119944e+12
  L1: 24882308
```

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

```
OLS in-sample r_sq is  0.8193381
```

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

```
OLS in-sample RMSE is  77461.65
```

```
#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  0.7217848
```

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

```
OLS out-of-sample RMSE is  90242.2
```

```
#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  0.4851503
```

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

```
tree out-of-sample RMSE is  140993.4
```

```r
#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  0.8302118
```

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

```
Forest out-of-sample RMSE is  69655.47
```