Sale Price Strategy

Final project for MATH 342W at Queens College

May 25, 2021

By Jaime Lin

Abstract

A strategic war in selecting how you want your house or apartment. There are many options that you need to think of how the ideal house you want to have, and it is hard to come with option will be better. Depending on what the costumer is looking for in the house or apartment that is needed to have rooms, what kind of heater fuel or the number of floors that the person wants to live. At the end, each person has their own preference, and if it everything is there that the range of sale price is the adequate, they will take that opportunity to buy the house.

1. Introduction

In this model we are going to explore how the sale price is high or low depending on what the building or apartment has that it will determine the sale price to be sold. A predictable model is model where you try to predict how fast the change of the output that is determinate from the independent variables. This means that the output can changes depending on how many options are being adding to have the ideal place to live and much it can escalate. The unit of observation is dollars because the ideal place to live will be a starting point to see how much is added to the ideal place that our costumers are looking for to buy or rent.

2. The Data

The data type of the output sale_price is string, coop_condo is string, total_num_rooms is integer. These three variables came from the data frame called "housing_data_2016_2017" and this data frame size is 2253 observations and 55 variables with a total of 123915 observations. This data set is a good representative of population of interest. Where population of interest means that a certain amount of people has their own choices of how they want to have their ideal options of their ideal house. An outlier is a person or thing that different from others. There are outliers and the most one that show the difference are the following variables are coop_condos, garages_exits, and parking_charges that can be noticed by being outside of the house. There is no danger of extrapolation.

2.1 Featurization

I did two measurements with the people that pick cooperative or condo. Cooperative is a place where different people live in different rooms in the same apartment or house while condos are the buildings that people rent an apartment from a different floor. After the choose between cooperative or condos is time to decide how many rooms are needed so the amount of people is going to move and live there. This is mostly come from the raw data set. And there are other variables that can influence the choice of the apartment in the cooperative or condo.

The variable sale_price is a string data type. This variable will be determining the price that the house or apartment will be after the person choices what they are looking. The next variable is coop_condos is a string data type. In each observation will be determined to be cooperative (coop) or condo. The variable num_total_rooms is a string data type if it saved as document, but it is a integer data type. This will the number of rooms that the house or apartment has inside.

2.2 Errors and Missingness

I found an error that is needed to change the data type of the data set to need to use to work to build the OLS linear model. And to calculate the R^2, RMSE, oob-R^2 and oob-RMSE need to use the function rnorm to get random real numbers. Also, I found that the data set have missing data to fix it, we need to use the function RandomForest to fill the missing spaces.

3. Modeling

3.1 Regression Tree Modeling

- approx._year_built: depending on how old the structure it can show how the people took care of it or how new it is.
- coop_condo: people choose to live in condo and pick an apartment to live with family. Or choose cooperative to life with friends to reduce the cost.
- fuel_type: selecting between gas or oil for the heater.
- garage_exits: looking a place to come with garage that can be used to enter and leave.
- datet_of_sale: an opportunity to explore the building if it has all of what you are looking for.
- kitchen_type: two type of kitchen are “efficiency” is where you can cook close to living room with all the instruments needed, can eat there, and go directly to the kitchen if something is missed and bring it. And “eat in” a kitchen that you can cook and eat there.
- maintenance_cost: this is the cost to maintain the building in perfect conditions.
- floors_in_building: the number of floors that a person is looking to have in a house or if the person is looking an apartment in a specific floor.
- num_total_rooms: the number of rooms that a person or family want to have.

- parking_charges: people look for parking places that can be cheaper or no cost a lot.

3.2 Linear Modeling

The in-sample error is that the data set is in string data type and converting to integer to use in our linear model get us a no data. Therefore, using the rnorm command to give random real numbers. The in- sample elaborated has a range from -3.38 to 3.78 this show how the selection of cooperative or condo with the number of rooms that the costumer is looking for and the parking charges to park the car around the place influence the sale price of the house that is been sold.

The coefficients mean the probability of the intercept point with the closest value in the three-independent variable. Also, they show the error percentage that it has 2.15% of error is small. And the Pr (>| t |) is the one use to prove the hypothesis that is elaborated.

3.3 Random Forest Modeling

Using the random forest modeling to fill the missing data. Because the data set that we collected has missing data that can be filled with random values. This mean that the date that was there is already filled and can be used.

It is parametric because the variables have a distribution. What I gain in this model is that how the people have their own preferences of what they want to have in the house that they are buying. What I lose is that it is hard to find the right price that a costumer is going to buy the house that have all the requirement that they be looking for. If we are talking about this data set has an iterative process, then is no because the preference of each costumer varies a lot after seeing all it will depend on the sale price is in their

range, so they can buy the house. Using the method underfitting will give us an accurate of how much the customers are looking their preferences accurate that is hard to find all in one place. While using the overfitting this will work because the data set give the information of what they have in each house and how this will help to get close of what the costumers want and buy the house when everything is settled. I do not know much about it this topic I am just saying what I feel if I was the costumer is looking the necessaries of a house with a sale price that is in my range. I believe that the following variables that affect a lot the variable sale_price are coop_condo, num_total_rooms, fuel_type, and parking charges.

4. Performance Results for your Random Forest Modeling

variable	R^2	RMSE	oobR^2	oobRMSE
coop_condo = x0	0.001432	2.05531	-3.185659	2.119083
num_total_rooms = x1	0.000351	2.06429	0.0001135	2.118929
parking_charges = x2	0.000365	2.06414	0.0003097	2.118721

The oob-R^2 has a negative number this means that the error is too small and oob-RMSE has almost the same number. We can conclude that oob-estimation is close to our linear OLS model in the 3.2 in the variable x0. R^2 converting this in percentages we found that the combination of the have coop_condo, num_total_rooms and parking_charges at the same times is getting small. And the RMSE its almost the same and this demonstrate that

the options that the costumer require will be there. The validation of this model can be true in the moment to predict if you continuous adding another variable like fuel_type the probability to have it is getting smaller.

5. Discussion

Extract the data set to see what are the fundamental variables that the costumer check to have the ideal house with their own sale price. In the third step, I started saying what variables the costumer is looking and this can increase the sale price. With these variables influence the sale price to be cheaper to expensive.

Next, we start modeling a regression tree model to see the nodes of the sale_price with coop_condo these can influence how the people prefer to live before seeing what the house can offer. A OLS linear model to see how the sale price is affected with the selection of having the following variables coop_condo, num_total_rooms and parking_charges. This will estimate how the more you add the probability of finding of what you are looking for is getting smaller. After that we use the function randomForest to fill the missing data and if you try to see the information it will be long after all is filling each row in each column and it will take some time to fill. In the last one we calculate the in and out R^2 and RMSE the process to calculate both in and out is the same. First, we create our y and x then calculate x_bar and y_bar using the mean or summation of all x or y divide by the total that is n. I reduced the n to be the sample size. Second, create b_1 = (sum(x0*y)-n*x_bar0*y_bar)/(sum(x0^2)-n*x_bar0^2) and b_0 = y_bar - b_1*x_bar0. Then, find yhat=b_0+b_1*x0, then e0=y-yhat. Third, find using the following formula :SSE0=sum(e0^2), SST0=sum((y-y_bar)^2), MSE0=SSE0/(n-2), RMSE0=sqrt(MSE0), Rsq0 = 1 - SSE0 / SST0.

Acknowledgments

Jenny Zhang. She is my cousin, and she works as Real State Agent. She helped me to understand better the variables and explain how it can influence in the sale price.

Code Appendix

Data

```
pacman::p_load(dplyr, magrittr, data.table)
x = read.csv("C:\\Users\\jaime\\OneDrive\\Desktop\\housing_data_2016_2017.csv
", header = TRUE)
xhat = x %>%
  select(coop_condo, full_address_or_zip_code, approx_year_built, common_char
ges, maintenance_cost, parking_charges, listing_price_to_nearest_1000, total_
taxes, date_of_sale, fuel_type, garage_exists, kitchen_type, num_total_rooms,
 num_bedrooms, num_floors_in_building, sale_price)
```

Regression Tree Modeling

```
pacman::p_load(rpart)
rpart(sale_price~coop_condo, data = xhat)

## n=528 (1702 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 528 517 $155,000  (0.0019 0.0019 0.0019 0.0019 0.0038 0.0019 0.001
9 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019 0.0095 0.0019 0.0038 0.007
6 0.0019 0.0019 0.0019 0.0019 0.0019 0.013 0.0038 0.0057 0.0019 0.0038 0.0038
 0.0019 0.0019 0.0019 0.0019 0.0038 0.0038 0.0019 0.011 0.0038 0.0019 0.0095
0.0019 0.0019 0.0019 0.0019 0.0019 0.021 0.0019 0.0038 0.0057 0.0038 0.0019
0.0076 0.0038 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0038
0.0057 0.0095 0.0019 0.0019 0.0038 0.019 0.0019 0.0019 0.0019 0.0019 0.0038
0.0095 0.0019 0.0019 0.0095 0.0019 0.0019 0.0038 0.0057 0.0019 0.0019 0.0019
0.0019 0.0019 0.0019 0.0076 0.0019 0.0038 0.0019 0.0095 0.0038 0.0038 0.0019
0.0019 0.0019 0.0019 0.0038 0.013 0.0038 0.0019 0.0019 0.0076 0.0019 0.0038
0.0057 0.0019 0.0038 0.0019 0.013 0.0019 0.0019 0.0019 0.0019 0.0019 0.0076
0.0019 0.0019 0.0057 0.0019 0.0038 0.0019 0.0019 0.0076 0.0019 0.0019 0.0057
0.0019 0.0019 0.0019 0.0019 0.0057 0.0038 0.0019 0.0076 0.0019 0.0019 0.0019
0.0057 0.0019 0.0019 0.011 0.0019 0.0019 0.0095 0.0019 0.0019 0.0076 0.0019
0.0019 0.0019 0.0019 0.0076 0.0019 0.0019 0.0019 0.0019 0.0038 0.0038 0.0019
0.0038 0.0019 0.0076 0.0057 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0038
0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0076 0.0019 0.0038 0.0019 0.0057
0.0038 0.0019 0.0038 0.0038 0.0076 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019
0.0019 0.0019 0.0038 0.0019 0.0019 0.0038 0.0019 0.0057 0.0057 0.0019 0.0076
0.0038 0.0019 0.0019 0.0057 0.0019 0.0038 0.0019 0.0038 0.0019 0.0038 0.0038
0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0038 0.0019
0.0057 0.0019 0.0019 0.0057 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019
0.0057 0.0038 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019 0.0019
0.0038 0.0038 0.0019 0.0038 0.0038 0.0038 0.0038 0.0019 0.0019 0.0019 0.0038
0.0019 0.0038 0.0038 0.0019 0.0019 0.0019 0.0019 0.0057 0.0019 0.0019 0.0019
```

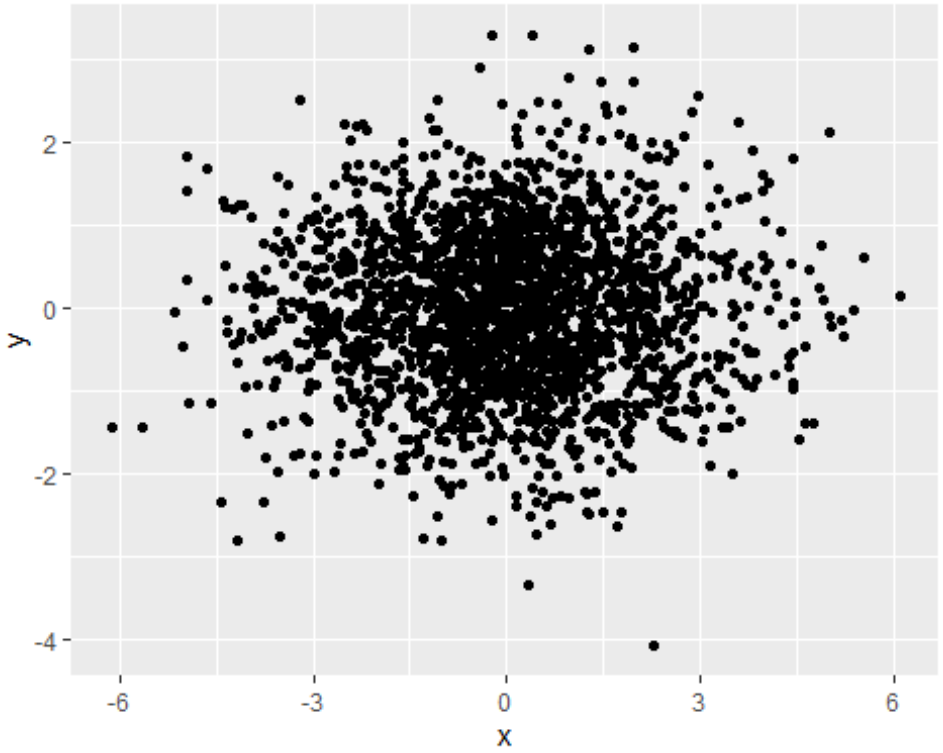
```
0.0019 0.0019 0.0057 0.0038 0.0038 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019
0.0019 0.0019 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0038 0.0019 0.0019
0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019
0.0019 0.0019 0.0019 0.0019 0.0038 0.0019 0.0019 0.0019 0.0019 0.0019 0.0019)
 *
```

Linear Modeling

```
y = rnorm(xhat$sale_price)
x0 = rnorm(xhat$coop_condo)
x1 = rnorm(xhat$num_total_rooms)
x2 = rnorm(xhat$parking_charges)
data_set = lm(y ~ x0 + x1 + x2)
summary(data_set)

##
## Call:
## lm(formula = y ~ x0 + x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0746 -0.6439  0.0062  0.6375  3.3461
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.034478   0.020185  -1.708   0.0878 .
## x0           0.003352   0.020036   0.167   0.8671
## x1           0.014030   0.019827   0.708   0.4792
## x2          -0.011799   0.020244  -0.583   0.5601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9529 on 2226 degrees of freedom
## Multiple R-squared:  0.0003774,  Adjusted R-squared:  -0.0009698
## F-statistic: 0.2801 on 3 and 2226 DF,  p-value: 0.8398

pacman::p_load(ggplot2)
ggplot(data.frame(x = x0 + x1 + x2, y = y)) +
  geom_point(aes(x = x, y = y))
```



Random Forest

```
pacman::p_load(randomForest)

n = 500
sigma = 0.5
x_min = min(y)
x_max = max(y)
f_x = function(x){sin(x)}
y_x = function(x, sigma){f_x(x) + rnorm(n, 0, sigma)}
x_train = runif(n, x_min, x_max)
y_train = y_x(x_train, sigma)
x_test = runif(n, x_min, x_max)
y_test = y_x(x_test, sigma)
node_sizes = 1:500

se_by_node_sizes = array(NA, length(node_sizes))
for (i in 1:length(node_sizes)) {
  rf_mod = randomForest(x = data.frame(x = x_train), y = y_train, ntree = 1,
replace = FALSE, sampsize = n, nodesize = node_sizes[i])
  y_hat_test = predict(rf_mod,data.frame(x = x_test))
  se_by_node_sizes[i] = sd(y_test - y_hat_test)
}
```

Finding R^2 and RMSE

```
  # x0 = coop_condo
  x_bar0 = sum(x0)/n
  y_bar = sum(y)/n
  b_1 = (sum(x0*y)-n*x_bar0*y_bar) / (sum(x0^2)-n*x_bar0^2)
  b_0 = y_bar - b_1*x_bar0
  yhat = b_0 + b_1 *x0
  e0 = y -yhat
  SSE0 = sum(e0^2)
  SST0 = sum((y-y_bar)^2)
  MSE0 = SSE0 /(n-2)
  RMSE0 = sqrt(MSE0)
  Rsq0 = 1 - SSE0 / SST0

  # x1 = num_total_rooms
  x_bar1 = sum(x1)/n
  b_1 = (sum(x1*y)-n*x_bar1*y_bar) / (sum(x1^2)-n*x_bar1^2)
  b_0 = y_bar - b_1*x_bar1
  yhat = b_0 + b_1 *x1
  e1 = y -yhat
  SSE1 = sum(e1^2)
  SST1 = sum((y-y_bar)^2)
  MSE1 = SSE1 /(n-2)
  RMSE1 = sqrt(MSE1)
  Rsq1 = 1 - SSE1 / SST1

  # x2 = parking_charges
  x_bar2 = sum(x2)/n
  b_1 = (sum(x2*y)-n*x_bar2*y_bar) / (sum(x2^2)-n*x_bar2^2)
  b_0 = y_bar - b_1*x_bar2
  yhat = b_0 + b_1 *x2
  e2 = y -yhat
  SSE2 = sum(e2^2)
  SST2 = sum((y-y_bar)^2)
  MSE2 = SSE2 /(n-2)
  RMSE2 = sqrt(MSE2)
  Rsq2 = 1 - SSE2 / SST2
```

oobRsq and oobRMSE

```
nstar = 500
ystar = rnorm(xhat$sale_price)
x0_star = rnorm(xhat$coop_condo)
x1_xtar = rnorm(xhat$num_total_rooms)
x2_xtar = rnorm(xhat$parking_charges)
# x0 = coop_condo
  x_bar0 = sum(x0_star)/n
  y_bar = sum(ystar)/n
  b_1 = (sum(x0_star*ystar)-n*x_bar0*y_bar) / (sum(x0_star^2)-n*x_bar0^2)
  b_0 = y_bar - b_1*x_bar0
  yhat = b_0 + b_1 *x0_star
```

```
oobe0 = ystar -yhat
oobSSE0 = sum(oobe0^2)
oobSST0 = sum((ystar-y_bar)^2)
oobMSE0 = oobSSE0 /(n-2)
oobRMSE0 = sqrt(oobMSE0)
oobRsq0 = 1 - oobSSE0 / oobSST0

# x1 = num_total_rooms
x_bar1 = sum(x1_xtar)/n
b_1 = (sum(x1_xtar*ystar)-n*x_bar1*y_bar) / (sum(x1_xtar^2)-n*x_bar1^2)
b_0 = y_bar - b_1*x_bar1
yhat = b_0 + b_1 *x1_xtar
oobe1 = ystar -yhat
oobSSE1 = sum(oobe1^2)
oobSST1 = sum((ystar-y_bar)^2)
oobMSE1 = oobSSE1 /(n-2)
oobRMSE1 = sqrt(oobMSE1)
oobRsq1 = 1 - oobSSE1 / oobSST1

# x2 = parking_charges
x_bar2 = sum(x2_xtar)/n
b_1 = (sum(x2_xtar*ystar)-n*x_bar2*y_bar) / (sum(x2_xtar^2)-n*x_bar2^2)
b_0 = y_bar - b_1*x_bar2
yhat = b_0 + b_1 *x2_xtar
oobe2 = ystar -yhat
oobSSE2 = sum(oobe2^2)
oobSST2 = sum((ystar-y_bar)^2)
oobMSE2 = oobSSE2 /(n-2)
oobRMSE2 = sqrt(oobMSE2)
oobRsq2 = 1 - oobSSE2 / oobSST2
```